GigaDevice Semiconductor Inc.

GD32E53x
Arm® Cortex®-M33 32-bit MCU

适用于 **GD32E533xx, GD32E535xx, GD32E537xx, GD32E53Pxx** 系列

用户手册

1.0 版本

(2026 年 11 月)

目录

目录 2

图索引 29

表索引 38

1. 系统及存储器架构 42

1.1. Arm® Cortex®-M33 处理器 42

1.2. 系统架构 43

1.3. 存储器映射 45

1.3.1. 片上 SRAM 存储器 50
1.3.2. 片上 FLASH 存储器概述 51

1.4. 引导配置 51

1.5. 系统配置寄存器(SYSCFG) 53

1.5.1. 配置寄存器 0 (SYSCFG_CFG0) 53
1.5.2. 配置寄存器 1(SYSCFG_CFG1) 53
1.5.3. 锁定控制寄存器(SYSCFG_LKCTL) 54
1.5.4. 总线超时寄存器(SYSCFG_BUSTO) 55
1.5.5. SRAM_ECC 错误状态寄存器 0(SYSCFG_ECCS0) 56
1.5.6. FPU 中断使能寄存器(SYSCFG_FPUINTEN) 57
1.5.7. SRAM 写保护寄存器 0 (SYSCFG_SRAMWP0) 58
1.5.8. SRAM ECC 错误状态寄存器 1 (SYSCFG_ECCS1) 59
1.5.9. ECC 配置寄存器 0 (SYSCFG_ECCCFG0) 60
1.5.10. SRAM ECC 配置寄存器 1 (SYSCFG_ECCCFG1) 61
1.5.11. SRAM ECC 配置寄存器 2 (SYSCFG_ECCCFG2) 62
1.5.12. SRAM ECC 配置寄存器 3 (SYSCFG_ECCCFG3) 63
1.5.13. 总线超时状态寄存器 (SYSCFG_BUSTOSTAT) 63
1.5.14. SRAM 写保护寄存器 1 (SYSCFG_SRAMWP1) 64
1.5.15. 定时器输入选择寄存器 (SYSCFG_TIMERCISEL) 65
1.5.16. SRAM ECC 配置寄存器 4 (SYSCFG_ECCCFG4) 67

1.5.17. SRAM ECC 配置寄存器 5(SYSCFG_ECCCFG5) 67
1.5.18. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=0, 7, 19) 68
1.5.19. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=0, 7, 19) 70
1.5.20. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=0, 7, 19) 70
1.5.21. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=1, 2, 3, 4) 71
1.5.22. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=1, 2, 3, 4) 73
1.5.23. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=1, 2, 3, 4) 74
1.5.24. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=9, 10, 12, 13) 75
1.5.25. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=9, 10, 12, 13) 77
1.5.26. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=9, 10, 12, 13) 78
1.5.27. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=14, 40-44) 78
1.5.28. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=14, 40-44) 80
1.5.29. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=14, 40-44) 81

1.6. 设备电子签名 82

1.6.1. 存储容量信息 82
1.6.2. 设备唯一 ID(96 位/位域) 82

2. 闪存控制器(FMC) 84

2.1. 概述 84

2.2. 特性 84

2.3. 功能概述 84

2.3.1. Flash 存储器架构 84
2.3.2. 错误检查与纠正(ECC) 85
2.3.3. 读操作 86
2.3.4. FMC_CTL / FMC_OBCTL 寄存器解锁 88
2.3.5. 页擦除 88
2.3.6. 全片擦除 89
2.3.7. 主闪存编程 90
2.3.8. OTP 编程 92
2.3.9. 选项字节 96
2.3.10. 页擦除/编程保护(写保护) 101
2.3.11. 安全保护 101
2.3.12. 安全用户区域(SCR) 102

2.4. 寄存器定义 103

2.4.1. 等待状态寄存器(FMC_WS) 103
2.4.2. 解锁寄存器(FMC_KEY) 104

2.4.3. 选项字节解锁寄存器(FMC_OBKEY) 104
2.4.4. 状态寄存器(FMC_STAT) 104
2.4.5. 控制寄存器(FMC_CTL) 106
2.4.6. 地址寄存器 (FMC_ADDR) 107
2.4.7. ECC 控制寄存器 (FMC_ECCCTL) 108
2.4.8. 偏移区域寄存器 (FMC_OFRG) 108
2.4.9. 偏移值寄存器 (FMC_OFVR) 109
2.4.10. SCR 控制寄存器 (FMC_SCRCTL) 109
2.4.11. 选项字节控制寄存器 (FMC_OBCTL) 110
2.4.12. SCR 写保护寄存器 (FMC_SCRWLK) 112
2.4.13. 擦除/编程保护寄存器 (FMC_WPx) (x = 0…3) 112
2.4.14. OTP DATA1 低块锁定寄存器(FMC_OTPDATA1_LLK) 113
2.4.15. OTP DATA1 高块锁定寄存器 (FMC_OTPDATA1_HLK) 113
2.4.16. OTP DATA2 低块锁定寄存器 (FMC_OTPDATA2_LLK) 114
2.4.17. OTP DATA2 高块锁定寄存器 (FMC_OTPDATA2_HLK) 114
2.4.18. OTP DATA2 低读取控制寄存器 (FMC_OTPDATA2_LR_CTL) 115
2.4.19. OTP DATA2 高读取控制寄存器 (FMC_OTPDATA2_HR_CTL) 115
2.4.20. ECC 控制和状态寄存器 (FMC_ECCCS) 116
2.4.21. ECC 地址寄存器 (FMC_ECCADDR) 117
2.4.22. 产品 ID 寄存器 0 (FMC_PID0) 117
2.4.23. 产品 ID 寄存器 1 (FMC_PID1) 118

3. 指令缓存(ICACHE) 119

3.1. 简介 119

3.2. 主要特性 119

3.3. ICACHE 功能描述 119

3.3.1. ICACHE 初始化 120
3.3.2. 双主机缓存 120
3.3.3. ICACHE TAG 存储 121
3.3.4. 地址重映射 121
3.3.5. 可缓冲和非缓冲访问 122
3.3.6. ICACHE 性能监测 124
3.3.7. ICACHE 错误管理和中断 124

3.4. ICACHE 寄存器 125

3.4.1. 控制寄存器(ICACHE_CTL) 125
3.4.2. 状态寄存器(ICACHE_STAT) 126
3.4.3. 中断使能寄存器(ICACHE_INTEN) 126

3.4.4. 标志清除寄存器(ICACHE_FC) 127
3.4.5. 命中监测计数寄存器(ICACHE_HMC) 127
3.4.6. 缺失监测计数寄存器(ICACHE_MMC) 128
3.4.7. 配置寄存器(ICACHE_CFGx)(x = 0...3) 128

4. 数据缓存(DCACHE) 130

4.1. 简介 130

4.2. 特性 130

4.3. 功能概述 130

4.3.1. DCACHE 初始化 131
4.3.2. DCACHE TAG 存储 132
4.3.3. 可缓存和不可缓存访问 133
4.3.4. 缓存维护操作 136
4.3.5. DCACHE 性能监控 137

4.4. 寄存器定义 138

4.4.1. 控制寄存器(DCACHE_CTL) 138
4.4.2. 状态寄存器(DCACHE_STAT) 139
4.4.3. 中断使能寄存器(DCACHE_INTEN) 140
4.4.4. 标志清除寄存器(DCACHE_FC) 141
4.4.5. 读命中监视计数寄存器(DCACHE_RHMC) 142
4.4.6. 读未命中监视计数寄存器(DCACHE_RMMC) 142
4.4.7. 写命中监视计数寄存器(DCACHE_WHMC) 142
4.4.8. 写未命中监视计数寄存器(DCACHE_WMMC) 143
4.4.9. 维护操作起始地址寄存器(DCACHE_MOSADDR) 143
4.4.10. 维护操作结束地址寄存器 (DCACHE_MOEADDR) 144

5. 电源管理单元(PMU) 145

5.1. 简介 145

5.2. 主要特性 145

5.3. 功能描述 146

5.3.1. 电池备份域 147
5.3.2. 备份 SRAM 148
5.3.3. V_{DD} / V_{DDA} 电源域 148

5.3.4. V_{CORE} 电源域 152
5.3.5. 省电模式 153

5.4. PMU 寄存器 157

5.4.1. 控制寄存器 0(PMU_CTL0) 157
5.4.2. 电源控制和状态寄存器 0(PMU_CS0) 159
5.4.3. 控制寄存器 1(PMU_CTL1) 162
5.4.4. 电源控制和状态寄存器 1(PMU_CS1) 164
5.4.5. 控制寄存器 2(PMU_CTL2) 164
5.4.6. 控制寄存器 3(PMU_CTL3) 165
5.4.7. 参数寄存器(PMU_PAR) 167

6. 复位和时钟单元(RCU) 168

6.1. 复位控制单元(RCTL) 168

6.1.1. 简介 168
6.1.2. 功能说明 168

6.2. 时钟控制单元(CCTL) 169

6.2.1. 简介 169
6.2.2. 主要特征 172
6.2.3. 功能说明 172

6.3. 寄存器定义 177

6.3.1. 控制寄存器(RCU_CTL) 177
6.3.2. 时钟配置寄存器 0(RCU_CFG0) 179
6.3.3. 时钟中断寄存器(RCU_INT) 182
6.3.4. APB2 复位寄存器(RCU_APB2RST) 185
6.3.5. APB1 复位寄存器(RCU_APB1RST) 188
6.3.6. AHB 使能寄存器(RCU_AHBEN) 191
6.3.7. APB2 使能寄存器(RCU_APB2EN) 193
6.3.8. APB1 使能寄存器(RCU_APB1EN) 197
6.3.9. 备份域控制寄存器(RCU_BDCTL) 200
6.3.10. 复位源/时钟寄存器(RCU_RSTSCK) 202
6.3.11. AHB 复位寄存器(RCU_AHBRST) 203
6.3.12. 时钟配置寄存器 1(RCU_CFG1) 205
6.3.13. 深睡眠模式电压寄存器(RCU_DSV) 207
6.3.14. 时钟频率监测配置寄存器 0(RCU_CKFMCFG0) 208
6.3.15. 时钟频率监测配置寄存器 1(RCU_CKFMCFG1) 209
6.3.16. 时钟频率监测配置寄存器 2(RCU_CKFMCFG2) 210
6.3.17. 时钟频率监测配置寄存器 3(RCU_CKFMCFG3) 211

6.3.18. 时钟频率监测配置寄存器 4(RCU_CKFMCFG4) 212
6.3.19. 附加时钟控制寄存器(RCU_ADDCTL) 213
6.3.20. 附加时钟中断寄存器(RCU_ADDINT) 214
6.3.21. PLL 时钟展频控制寄存器(RCU_PLLSSCTL) 215
6.3.22. 时钟配置寄存器 2(RCU_CFG2) 216
6.3.23. APB1 附加复位寄存器(RCU_ADDAPB1RST) 217
6.3.24. APB1 附加使能寄存器(RCU_ADDAPB1EN) 217
6.3.25. 锁定寄存器(RCU_LOCK) 218

7. 时钟校准控制器(CTC) 219

7.1. 简介 219

7.2. 主要特征 219

7.3. 功能说明 219

7.3.1. REF 同步脉冲发生器 220
7.3.2. CTC 校准计数器 220
7.3.3. 频率评估和自动校准过程 221
7.3.4. 软件编程指南 222

7.4. CTC 寄存器 223

7.4.1. 控制寄存器 0(CTC_CTL0) 223
7.4.2. 控制寄存器 1(CTC_CTL1) 224
7.4.3. 状态寄存器(CTC_STAT) 225
7.4.4. 中断清除寄存器(CTC_INTC) 227

8. 中断/事件控制器(EXTI) 229

8.1. 简介 229

8.2. 主要特征 229

8.3. 中断功能描述 229

8.4. 外部中断及事件(EXTI)框图 233

8.5. 外部中断及事件功能概述 233

8.5.1. 硬件触发 234

8.5.2. 软件触发 234

8.6. EXTI 寄存器 236

8.6.1. 中断使能寄存器(EXTI_INTEN) 236
8.6.2. 事件使能寄存器(EXTI_EVEN) 236
8.6.3. 上升沿触发使能寄存器(EXTI_RTEN) 237
8.6.4. 下降沿触发使能寄存器(EXTI_FTEN) 237
8.6.5. 软件中断事件寄存器(EXTI_SWIEV) 237
8.6.6. 挂起寄存器(EXTI_PD) 238

9. 触发选择控制器(TRIGSEL) 239

9.1. 简介 239

9.2. 主要特征 239

9.3. 功能说明 239

9.4. 内部连接 240

9.5. TRIGSEL 寄存器 249

9.5.1. EXTOUT 触发选择寄存器 0(TRIGSEL_EXTOUT_0) 249
9.5.2. EXTOUT 触发选择寄存器 1(TRIGSEL_EXTOUT_1) 249
9.5.3. EXTOUT 触发选择寄存器 2(TRIGSEL_EXTOUT_2) 250
9.5.4. EXTOUT 触发选择寄存器 3(TRIGSEL_EXTOUT_3) 251
9.5.5. TIMER0_ITI 触发选择寄存器(TRIGSEL_TIMER0ITI) 251
9.5.6. TIMER1_ITI 触发选择寄存器(TRIGSEL_TIMER1ITI) 252
9.5.7. TIMER2_ITI 触发选择寄存器(TRIGSEL_TIMER2ITI) 253
9.5.8. TIMER3_ITI 触发选择寄存器(TRIGSEL_TIMER3ITI) 253
9.5.9. TIMER4_ITI 触发选择寄存器(TRIGSEL_TIMER4ITI) 254
9.5.10. TIMER7_ITI 触发选择寄存器(TRIGSEL_TIMER7ITI) 254
9.5.11. TIMER14_ITI 触发选择寄存器(TRIGSEL_TIMER14ITI) 255
9.5.12. TIMER19_ITI 触发选择寄存器(TRIGSEL_TIMER19ITI) 256
9.5.13. TIMER40_ITI 触发选择寄存器(TRIGSEL_TIMER40ITI) 256
9.5.14. TIMER41_ITI 触发选择寄存器(TRIGSEL_TIMER41ITI) 257
9.5.15. TIMER42_ITI 触发选择寄存器(TRIGSEL_TIMER42ITI) 257
9.5.16. TIMER43_ITI 触发选择寄存器(TRIGSEL_TIMER43ITI) 258
9.5.17. TIMER44_ITI 触发选择寄存器(TRIGSEL_TIMER44ITI) 259
9.5.18. DAC 触发选择寄存器 0(TRIGSEL_DAC0OUT0) 259
9.5.19. DAC 触发选择寄存器 1(TRIGSEL_DAC0OUT1) 260

9.5.20. ADC0_ROUTRG 触发选择寄存器(TRIGSEL_ADC0_ROUTRG) 260
9.5.21. ADC0_INSTRG 触发选择寄存器(TRIGSEL_ADC0_INSTRG) 261
9.5.22. ADC1_ROUTRG 触发选择寄存器(TRIGSEL_ADC1_ROUTRG) 262
9.5.23. ADC1_INSTRG 触发选择寄存器(TRIGSEL_ADC1_INSTRG) 262
9.5.24. ADC2_ROUTRG 触发选择寄存器(TRIGSEL_ADC2_ROUTRG) 263
9.5.25. ADC2_INSTRG 触发选择寄存器(TRIGSEL_ADC2_INSTRG) 263
9.5.26. TIMER0_BRKIN 触发选择寄存器(TRIGSEL_TIMER0BRKIN) 264
9.5.27. TIMER0_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER0CHBRKIN) 265
9.5.28. TIMER7_BRKIN 触发选择寄存器(TRIGSEL_TIMER7BRKIN) 265
9.5.29. TIMER7_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER7CHBRKIN) 266
9.5.30. TIMER19_BRKIN 触发选择寄存器(TRIGSEL_TIMER19BRKIN) 267
9.5.31. TIMER19_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER19CHBRKIN) 267
9.5.32. TIMER14_BRKIN 触发选择寄存器(TRIGSEL_TIMER14BRKIN) 268
9.5.33. TIMER40_BRKIN 触发选择寄存器(TRIGSEL_TIMER40BRKIN) 269
9.5.34. TIMER41_BRKIN 触发选择寄存器(TRIGSEL_TIMER41BRKIN) 269
9.5.35. TIMER42_BRKIN 触发选择寄存器(TRIGSEL_TIMER42BRKIN) 270
9.5.36. TIMER43_BRKIN 触发选择寄存器(TRIGSEL_TIMER43BRKIN) 270
9.5.37. TIMER44_BRKIN 触发选择寄存器(TRIGSEL_TIMER44BRKIN) 271
9.5.38. TIMER9_ITI 触发选择寄存器(TRIGSEL_TIMER9ITI) 272
9.5.39. TIMER10_ITI 触发选择寄存器(TRIGSEL_TIMER10ITI) 272
9.5.40. TIMER12_ITI 触发选择寄存器(TRIGSEL_TIMER12ITI) 273
9.5.41. TIMER13_ITI 触发选择寄存器(TRIGSEL_TIMER13ITI) 274

10. 通用和备用输入/输出接口(GPIO 和 AFIO) 275

10.1. 简介 275

10.2. 主要特征 275

10.3. 功能说明 275

10.3.1. GPIO 引脚配置 277
10.3.2. 外部中断/事件线 277
10.3.3. 备用功能(AF) 277
10.3.4. 输入配置 277
10.3.5. 输出配置 278
10.3.6. 模拟配置 279
10.3.7. 备用功能(AF)配置 279
10.3.8. GPIO 锁定功能 280
10.3.9. GPIO I/O 补偿单元 280
10.3.10. 输入滤波 280
10.3.11. CLK 引脚 AF 重映射 283

10.4. GPIO 寄存器 284
10.4.1. 端口控制寄存器 (GPIOx_CTL, x=A...E) 284
10.4.2. 端口输出模式寄存器 (GPIOx_OMODE x=A...E) 286
10.4.3. 端口输出速度寄存器 (GPIOx_OSPD x=A...E) 287
10.4.4. 端口上拉/下拉寄存器(GPIOx_PUD, x=A...E) 289
10.4.5. 端口输入状态寄存器 (GPIOx_ISTAT, x=A...E) 291
10.4.6. 端口输出控制寄存器 (GPIOx_OCTL, x=A...E) 291
10.4.7. 端口位操作寄存器 (GPIOx_BOP, x=A...E) 292
10.4.8. 端口配置锁定寄存器 (GPIOx_LOCK, x=A...E) 292
10.4.9. AFIO 选择寄存器 0 (GPIOx_AFSEL0, x=A...E) 293
10.4.10. AFIO 选择寄存器 1(GPIOx_AFSEL1, x=A...E) 294
10.4.11. 位清除寄存器 (GPIOx_BC, x=A...E) 295
10.4.12. 输入滤波寄存器(GPIOx_IFL, x=A...E) 296
10.4.13. 输入滤波类型寄存器(GPIOx_IFTP, x=A...E) 296
10.4.14. EXTI 源选择寄存器 0 寄存器 (AFIO_EXTISS0) 298
10.4.15. EXTI 源选择寄存器 1 寄存器 (AFIO_EXTISS1) 299
10.4.16. EXTI 源选择寄存器 2 寄存器 (AFIO_EXTISS2) 300
10.4.17. EXTI 源选择寄存器 3 寄存器 (AFIO_EXTISS3) 301
10.4.18. IO 补偿控制寄存器 (AFIO_CPSCTL) 302
10.4.19. PD01 重映射寄存器 (AFIO_PD01) 303

11. 循环冗余校验计算单元(CRC) 304

11.1. 简介 304

11.2. 主要特性 304

11.3. 功能描述 305

11.4. CRC 寄存器 306
11.4.1. 数据寄存器(CRC_DATA) 306
11.4.2. 独立数据寄存器(CRC_FDATA) 306
11.4.3. 控制寄存器(CRC_CTL) 307
11.4.4. 初值寄存器(CRC_IDATA) 307
11.4.5. 多项式寄存器(CRC_POLY) 308

12. 真随机数生成器(TRNG) 309

12.1. 简介 309

12.2. 主要特征 309
12.3. 功能说明 309
12.3.1. 操作流程 310
12.3.2. 错误标志 310
12.4. TRNG 寄存器 311
12.4.1. 控制寄存器(TRNG_CTL) 311
12.4.2. 状态寄存器(TRNG_STAT) 311
12.4.3. 数据寄存器(TRNG_DATA) 312
13. 加密处理器(CAU) 314
13.1. 简介 314
13.2. 主要特征 314
13.3. CAU 数据类型和初始化向量 314
13.3.1. 数据类型 314
13.4. 加密处理器流程 316
13.4.1. AES 加密处理流程 316
13.5. 操作模式 318
13.6. CAU DMA 接口 318
13.7. CAU 中断 319
13.8. CAU 寄存器 320
13.8.1. 控制寄存器(CAU_CTL) 320
13.8.2. 状态寄存器 0 (CAU_STAT0) 321
13.8.3. 数据输入寄存器 (CAU_DI) 322
13.8.4. 数据输出寄存器 (CAU_DO) 322
13.8.5. DMA 使能寄存器 (CAU_DMAEN) 323
13.8.6. 中断使能寄存器 (CAU_INTEN) 323
13.8.7. 状态寄存器 1 (CAU_STAT1) 324
13.8.8. 中断标志寄存器 (CAU_INTF) 324

13.8.9. 密钥寄存器 (CAU_KEY0..3(H/L)) 325
13.8.10. 错误标志寄存器 (CAU_ERROR_FLAG) 328

14. 哈希处理器(HAU) 329

14.1. 简介 329

14.2. 主要特征 329

14.3. HAU 数据类型 329

14.4. HAU 内核 331

14.4.1. 自动数据填充 331
14.4.2. 摘要计算 332
14.4.3. 哈希模式 332

14.5. HAU 中断 333

14.5.1. 输入 FIFO 中断 333
14.5.2. 计算完成中断 333

14.6. HAU 寄存器 334

14.6.1. 控制寄存器(HAU_CTL) 334
14.6.2. 数据输入寄存器(HAU_DI) 335
14.6.3. 配置寄存器 (HAU_CFG) 335
14.6.4. 数据输出寄存器 (HAU_DO0..7) 336
14.6.5. 中断使能寄存器 (HAU_INTEN) 338
14.6.6. 状态与标志寄存器 (HAU_STAT) 339

15. 直接存储器访问控制器(DMA) 341

15.1. 简介 341

15.2. 主要特征 341

15.3. 结构框图 342

15.4. 功能说明 342

15.4.1. DMA 操作 342
15.4.2. 链表描述符 344

15.4.3. 数据传输模式 345
15.4.4. 重复区域访问 345
15.4.5. 外设握手 346
15.4.6. 仲裁 347
15.4.7. 地址生成 347
15.4.8. 循环模式 347
15.4.9. 存储器到存储器模式 347
15.4.10. 通道配置 348
15.4.11. 中断 348
15.4.12. DMA 请求映射 349

15.5. 寄存器定义 350

15.5.1. 中断标志寄存器 0(DMA_INTF0) 350
15.5.2. 中断标志寄存器 1(DMA_INTF1) 351
15.5.3. 中断标志清除寄存器 0(DMA_INTC0) 352
15.5.4. 中断标志清除寄存器 1(DMA_INTC1) 352
15.5.5. 通道 x 控制寄存器 0(DMA_CHxCTL0) 353
15.5.6. 通道 x 计数器寄存器(DMA_CHxCNT) 355
15.5.7. 通道 x 外设基地址寄存器(DMA_CHxPADDR) 356
15.5.8. 通道 x 存储器基地址寄存器(DMA_CHxMADDR) 356
15.5.9. 通道 0 外设地址偏移寄存器(DMA_CH0PAOFS) 357
15.5.10. 通道 0 存储器地址偏移寄存器(DMA_CH0MAOFS) 357
15.5.11. 通道 x 链表地址寄存器(DMA_CHxLADDR) 358
15.5.12. 通道 x 块计数寄存器(DMA_CHxBCNT) 358
15.5.13. 通道 x 外设块偏移寄存器(DMA_CHxPABOFS) 359
15.5.14. 通道 x 存储器块偏移寄存器(DMA_CHxMABOFS) 359
15.5.15. 通道 x 控制 1 寄存器(DMA_CHxCTL1) 360

16. DMA 请求多路复用器(DMAMUX) 363

16.1. 简介 363

16.2. 主要特征 363

16.3. 结构框图 364

16.4. 功能说明 364

16.4.1. DMAMUX 信号 365
16.4.2. DMAMUX 请求路由器 365
16.4.3. DMAMUX 请求生成器 367
16.4.4. 通道配置 368

16.4.5. 中断 368
16.4.6. DMAMUX 映射 369

16.5. DMAMUX 寄存器 375

16.5.1. 请求路由通道 x 配置寄存器(DMAMUX_RM_CHxCFG) 375
16.5.2. 请求路由通道中断标志位寄存器(DMAMUX_RM_INTF) 376
16.5.3. 请求路由通道中断标志位清除寄存器(DMAMUX_RM_INTC) 376
16.5.4. 请求生成通道 x 配置寄存器(DMAMUX_RG_CHxCFG) 377
16.5.5. 请求生成通道中断标志位寄存器(DMAMUX_RG_INTF) 378
16.5.6. 请求生成通道中断标志位清除寄存器(DMAMUX_RG_INTC) 378

17. 调试(DBG) 379

17.1. 简介 379

17.2. JTAG/SW 功能描述 379

17.2.1. 切换 JTAG/SW 接口 379
17.2.2. 引脚分配 379
17.2.3. JTAG 链状结构 380
17.2.4. 安全 JTAG 380
17.2.5. 调试复位 382
17.2.6. JEDEC-106 ID 代码 382

17.3. 调试保持功能描述 382

17.3.1. 低功耗模式调试支持 382
17.3.2. TIMER、I2C、CAN、WWDGT、FWDGT 和 RTC 外设调试支持 382

17.4. DBG 寄存器 383

17.4.1. ID 寄存器(DBG_ID) 383
17.4.2. 控制寄存器 0(DBG_CTL0) 383
17.4.3. 控制寄存器 1(DBG_CTL1) 386
17.4.4. 控制寄存器 2(DBG_CTL2) 388

18. 模数转换器(ADC) 389

18.1. 简介 389

18.2. 主要特征 389

18.3. 引脚和内部信号 390

18.4. 功能描述 391

18.4.1. ADC 时钟 391
18.4.2. ADC 使能 391
18.4.3. 常规序列和注入序列 391
18.4.4. 运行模式 392
18.4.5. 注入序列管理 396
18.4.6. 转换数据锁存 396
18.4.7. 转换结果阈值监测功能 398
18.4.8. 数据存储模式 398
18.4.9. 采样时间配置 399
18.4.10. 外部触发 400
18.4.11. DMA 请求 401
18.4.12. 溢出检测 401
18.4.13. ADC 内部通道 402
18.4.14. 电池电压检测 402
18.4.15. 可编程分辨率(DRES) 402
18.4.16. 片上硬件过采样 403

18.5. ADC 同步模式 405

18.5.1. 独立模式 406
18.5.2. 常规并行模式 406
18.5.3. 注入并行模式 407
18.5.4. 常规快速交叉模式 407
18.5.5. 常规慢速交叉模式 408
18.5.6. 注入交替触发模式 409
18.5.7. 常规并行和注入并行组合模式 410
18.5.8. 常规并行和注入交替触发组合模式 410
18.5.9. 注入并行和交叉组合模式 411

18.6. 中断 412

18.7. ADC 寄存器 413

18.7.1. 状态寄存器(ADC_STAT) 413
18.7.2. 控制寄存器 0(ADC_CTL0) 415
18.7.3. 控制寄存器 1(ADC_CTL1) 418
18.7.4. 采样时间寄存器 0(ADC_SAMPT0) 421
18.7.5. 采样时间寄存器 1(ADC_SAMPT1) 422
18.7.6. 注入通道数据偏移寄存器 x(ADC_IOFFx)(x=0..3) 423
18.7.7. 看门狗 0 高阈值寄存器(ADC_WD0HT) 423
18.7.8. 看门狗 0 低阈值寄存器(ADC_WD0LT) 424
18.7.9. 常规序列寄存器 0(ADC_RSQ0) 424

18.7.10. 常规序列寄存器 1(ADC_RSQ1) 425
18.7.11. 常规序列寄存器 2(ADC_RSQ2) 425
18.7.12. 注入序列寄存器(ADC_ISQ) 426
18.7.13. 锁存数据寄存器 x(ADC_LDATAx)(x= 0..3) 426
18.7.14. 常规数据寄存器(ADC_RDATA) 427
18.7.15. 注入数据寄存器(ADC_IDATA) 427
18.7.16. 锁存数据控制寄存器(ADC_LDCTL) 428
18.7.17. 看门狗 1 高阈值寄存器(ADC_WD1HT) 429
18.7.18. 看门狗 1 低阈值寄存器(ADC_WD1LT) 429
18.7.19. 看门狗 2 高阈值寄存器(ADC_WD2HT) 430
18.7.20. 看门狗 2 低阈值寄存器(ADC_WD2LT) 430
18.7.21. 看门狗 1 通道选择寄存器(ADC_WD1SR) 431
18.7.22. 看门狗 2 通道选择寄存器(ADC_WD2SR) 431
18.7.23. 过采样控制寄存器(ADC_OVSAMPCTL) 432
18.7.24. 采样时间寄存器 2(ADC_SAMPT2) 433

19. 数模转换器(DAC) 435

19.1. 简介 435

19.2. 主要特征 435

19.3. 功能描述 436

19.3.1. DAC 使能 436
19.3.2. DAC 输出缓冲 437
19.3.3. DAC 数据配置 437
19.3.4. DAC 触发 437
19.3.5. DAC 转换 437
19.3.6. DAC 噪声波 437
19.3.7. DAC 输出电压 438
19.3.8. DMA 请求 439
19.3.9. DAC 并发转换 439
19.3.10. DAC 输出缓冲区校准 439

19.4. DAC 寄存器 441

19.4.1. DACx 控制寄存器 (DAC_CTL0) 441
19.4.2. DACx 软件触发寄存器 (DAC_SWT) 444
19.4.3. DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH) 444
19.4.4. DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH) 445
19.4.5. DACx_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH) 445
19.4.6. DACx_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH) 445

19.4.7. DACx_OUT1 12 位左对齐数据保持寄存器（DAC_OUT1_L12DH) 446
19.4.8. DACx_OUT1 8 位右对齐数据保持寄存器（DAC_OUT1_R8DH) 446
19.4.9. DACx 并发模式 12 位右对齐数据保持寄存器（DACC_R12DH) 447
19.4.10. DACx 并发模式 12 位左对齐数据保持寄存器（DACC_L12DH) 447
19.4.11. DACx 并发模式 8 位右对齐数据保持寄存器（DACC_R8DH) 448
19.4.12. DACx_OUT0 数据输出寄存器（DAC_OUT0_DO) 448
19.4.13. DACx_OUT1 数据输出寄存器（DAC_OUT1_DO) 449
19.4.14. DAC 状态寄存器 0 (DAC_STAT0) 449
19.4.15. DACx 校准寄存器（DAC_CALR) 450

20. 比较器(CMP) 452

20.1. 简介 452

20.2. 主要特征 452

20.3. 功能描述 452

20.3.1. 比较器时钟 453
20.3.2. 比较器的 I/O 配置 453
20.3.3. 比较器迟滞 454
20.3.4. 比较器寄存器写保护 455
20.3.5. 比较器输出消隐 455
20.3.6. 比较器数字滤波 456
20.3.7. 电压定标器功能 456
20.3.8. 比较器中断 456

20.4. 比较器寄存器 457

20.4.1. 状态寄存器(CMP_STAT) 457
20.4.2. 中断标志位清除寄存器(CMP_IFC) 458
20.4.3. CMP0 控制状态寄存器(CMP0_CS) 458
20.4.4. CMP1 控制状态寄存器(CMP1_CS) 460

21. 看门狗定时器(WDGT) 463

21.1. 独立看门狗定时器(FWDGT) 463

21.1.1. 简介 463
21.1.2. 主要特性 463
21.1.3. 功能描述 463
21.1.4. FWDGT 寄存器 466

21.2. 窗口看门狗定时器(WWDGT) 470

21.2.1. 简介 470
21.2.2. 主要特性 470
21.2.3. 功能描述 470
21.2.4. WWDGT 寄存器 473

22. 实时时钟(RTC) 475

22.1. 简介 475

22.2. 主要特征 475

22.3. 功能描述 476

22.3.1. 结构框图 476
22.3.2. 时钟源和预分频 477
22.3.3. 影子寄存器 477
22.3.4. 位域可屏蔽可配置的闹钟 477
22.3.5. RTC 初始化和配置 478
22.3.6. 可配置周期的自动唤醒定时器 479
22.3.7. 读取日历 480
22.3.8. RTC 复位 481
22.3.9. RTC 移位功能 481
22.3.10. RTC 参考时钟检测 482
22.3.11. RTC 数字平滑校准 482
22.3.12. RTC 高精度校准 484
22.3.13. 时间戳功能 486
22.3.14. 侵入检测 487
22.3.15. 校准时钟输出 488
22.3.16. 闹钟输出 488
22.3.17. RTC 引脚配置 488
22.3.18. RTC 省电模式管理 489
22.3.19. RTC 中断 489

22.4. RTC 寄存器 491

22.4.1. 时间寄存器(RTC_TIME) 491
22.4.2. 日期寄存器(RTC_DATE) 491
22.4.3. 控制寄存器(RTC_CTL) 492
22.4.4. 状态寄存器(RTC_STAT) 495
22.4.5. 预分频寄存器(RTC_PSC) 497
22.4.6. 唤醒定时器寄存器(RTC_WUT) 498
22.4.7. 闹钟 0 时间日期寄存器(RTC_ALRM0TD) 499
22.4.8. 闹钟 1 时间日期寄存器(RTC_ALRM1TD) 500

22.4.9. 写保护钥匙寄存器(RTC_WPK) 501
22.4.10. 亚秒寄存器(RTC_SS) 501
22.4.11. 移位控制寄存器(RTC_SHIFTCTL) 502
22.4.12. 时间戳 0 上升沿时间寄存器(RTC_TTS0R) 503
22.4.13. 时间戳 0 上升沿日期寄存器(RTC_DTS0R) 503
22.4.14. 时间戳 0 上升沿亚秒寄存器(RTC_SSTS0R) 504
22.4.15. 高精度频率补偿寄存器(RTC_HRFC) 505
22.4.16. 侵入寄存器(RTC_TAMP) 506
22.4.17. 闹钟 0 亚秒寄存器(RTC_ALRM0SS) 507
22.4.18. 闹钟 1 亚秒寄存器(RTC_ALRM1SS) 508
22.4.19. 配置寄存器(RTC_CFG) 509
22.4.20. 备份寄存器(RTC_BKPx)(x=0..3) 510
22.4.21. 时间戳 0 下降沿时间寄存器(RTC_TTS0F) 511
22.4.22. 时间戳 0 下降沿日期寄存器(RTC_DTS0F) 511
22.4.23. 时间戳 0 下降沿亚秒寄存器(RTC_SSTS0F) 512
22.4.24. 时间戳 1 上升沿时间寄存器(RTC_TTS1R) 513
22.4.25. 时间戳 1 上升沿日期寄存器(RTC_DTS1R) 513
22.4.26. 时间戳 1 上升沿亚秒寄存器(RTC_SSTS1R) 514
22.4.27. 时间戳 1 下降沿时间寄存器(RTC_TTS1F) 515
22.4.28. 时间戳 1 下降沿日期寄存器(RTC_DTS1F) 516
22.4.29. 时间戳 1 下降沿亚秒寄存器(RTC_SSTS1F) 516
22.4.30. 高精度时钟频率校准寄存器(RTC_HRCFC) 517
22.4.31. 高精度校准寄存器(RTC_HRCR) 517

23. 定时器(TIMER) 519

23.1. 高级定时器(TIMERx, x=0, 7, 19) 521

23.1.1. 简介 521
23.1.2. 主要特性 521
23.1.3. 结构框图 521
23.1.4. 功能描述 522
23.1.5. TIMERx 寄存器(x=0,7,19) 561

23.2. 通用定时器 L0(TIMERx, x=1,2,3,4) 603

23.2.1. 简介 603
23.2.2. 主要特性 603
23.2.3. 结构框图 603
23.2.4. 功能描述 604
23.2.5. TIMERx 寄存器(x=1/2/3/4) 623

23.3. 通用定时器 L2(TIMERx, x=9,10,12,13) 647

23.3.1. 简介 647

23.3.2. 主要特性 647
23.3.3. 结构框图 647
23.3.4. 功能描述 648
23.3.5. TIMERx 寄存器(x=9,10,12,13) 655

23.4. 通用定时器 L3(TIMERx, x=14,40,41,42,43,44) 665

23.4.1. 简介 665
23.4.2. 主要特性 665
23.4.3. 结构框图 665
23.4.4. 功能描述 666
23.4.5. TIMERx 寄存器(x=14,40,41,42,43,44) 694

23.5. 基本定时器(TIMERx, x=5,6,50,51) 726

23.5.1. 简介 726
23.5.2. 主要特性 726
23.5.3. 结构框图 726
23.5.4. 功能描述 726
23.5.5. TIMERx 寄存器(x=5/6/50/51) 730

24. 通用同步异步收发器(USART) 735

24.1. 通用同步异步收发器(USARTx/UARTx, x=0..4) 735

24.1.1. 简介 735
24.1.2. 主要特性 735
24.1.3. 功能描述 736
24.1.4. USART 寄存器 751

24.2. 通用同步异步收发器(USARTx, x=5) 766

24.2.1. 简介 766
24.2.2. 主要特性 766
24.2.3. 功能描述 767
24.2.4. USART 寄存器 783

25. 内部集成电路总线接口(I2C) 805

25.1. 简介 805

25.2. 主要特征 805

25.3. 功能说明 805

25.3.1. 时钟要求 806
25.3.2. I2C 通讯流程 807
25.3.3. 噪声滤波器 809
25.3.4. I2C 时序配置 810
25.3.5. I2C 复位 811
25.3.6. 数据传输 812
25.3.7. I2C 从机模式 814
25.3.8. I2C 主机模式 819
25.3.9. SMBus 支持 824
25.3.10. SMBus 模式 827
25.3.11. 从省电模式唤醒 828
25.3.12. DMA 模式下数据传输 828
25.3.13. I2C 错误和中断 829
25.3.14. I2C 调试模式 829

25.4. I2C 寄存器 830

25.4.1. 控制寄存器 0(I2C_CTL0) 830
25.4.2. 控制寄存器 1(I2C_CTL1) 832
25.4.3. 从机地址寄存器 0(I2C_SADDR0) 834
25.4.4. 从机地址寄存器 1(I2C_SADDR1) 835
25.4.5. 时序寄存器(I2C_TIMING) 836
25.4.6. 超时寄存器(I2C_TIMEOUT) 837
25.4.7. 状态寄存器(I2C_STAT) 838
25.4.8. 状态清除寄存器(I2C_STATC) 840
25.4.9. PEC 寄存器(I2C_PEC) 841
25.4.10. 接收数据寄存器(I2C_RDATA) 842
25.4.11. 发送数据寄存器(I2C_TDATA) 842
25.4.12. 控制寄存器 2(I2C_CTL2) 842

26. 串行外设接口/片上音频接口(SPI/I2S) 844

26.1. 简介 844

26.2. 主要特性 844

26.2.1. SPI 主要特性 844
26.2.2. I2S 主要特性 844

26.3. SPI 功能说明 845

26.3.1. SPI 结构框图 845
26.3.2. SPI 信号线描述 845
26.3.3. SPI 时序和数据帧格式 846

26.3.4. NSS 功能 847
26.3.5. SPI 运行模式 848
26.3.6. DMA 功能 856
26.3.7. CRC 功能 856
26.3.8. SPI 中断 857

26.4. I2S 功能说明 858

26.4.1. I2S 结构框图 858
26.4.2. I2S 信号线描述 859
26.4.3. I2S 音频标准 859
26.4.4. I2S 时钟 866
26.4.5. 运行 867
26.4.6. DMA 功能 872
26.4.7. I2S 中断 872

26.5. SPI/I2S 寄存器 874

26.5.1. 控制寄存器 0(SPI_CTL0) 874
26.5.2. 控制寄存器 1(SPI_CTL1) 876
26.5.3. 状态寄存器(SPI_STAT) 877
26.5.4. 数据寄存器(SPI_DATA) 878
26.5.5. CRC 多项式寄存器(SPI_CRCPOLY) 879
26.5.6. 接收 CRC 寄存器(SPI_RCRC) 879
26.5.7. 发送 CRC 寄存器(SPI_TCRC) 880
26.5.8. I2S 控制寄存器(SPI_I2SCTL) 881
26.5.9. I2S 时钟预分频寄存器(SPI_I2SPSC) 882
26.5.10. SPI0 四路 SPI 控制寄存器(SPI_QCTL) 883

27. 八线 SPI 接口(OSPI) 884

27.1. 简介 884

27.2. 主要特征 884

27.3. OSPI 功能描述 884

27.3.1. OSPI 结构框图 884
27.3.2. OSPI 常规命令模式 888
27.3.3. HyperBusTM 模式 891
27.3.4. 回卷支持 894
27.3.5. 片选边界 894
27.3.6. 刷新支持 894

27.4. 操作模式 895
27.4.1. 间接模式 895
27.4.2. 状态轮询模式 896
27.4.3. 内存映射模式 896
27.5. OSPI 配置 897
27.5.1. OSPI 系统配置 897
27.5.2. OSPI 器件配置 897
27.5.3. OSPI 常规命令模式配置 897
27.5.4. OSPI HyperBusTM 模式配置 898
27.6. OSPI 数据采样移位 899
27.7. OSPI 只发送一次指令 899
27.8. 繁忙状态 899
27.9. 错误管理 900
27.10. OSPI 中断 900
27.11. OSPI 寄存器 901
27.11.1. 控制寄存器(OSPI_CTL) 901
27.11.2. 设备配置寄存器 0(OSPI_DCFG0) 904
27.11.3. 设备配置寄存器 1(OSPI_DCFG1) 905
27.11.4. 设备配置寄存器 2(OSPI_DCFG2) 907
27.11.5. 设备配置寄存器 3(OSPI_DCFG3) 907
27.11.6. 状态寄存器(OSPI_STAT) 908
27.11.7. 状态清除寄存器(OSPI_STATC) 909
27.11.8. 数据长度寄存器(OSPI_DTLEN) 909
27.11.9. 地址寄存器(OSPI_ADDR) 910
27.11.10. 数据寄存器(OSPI_DATA) 910
27.11.11. 状态屏蔽寄存器(OSPI_STATMK) 911
27.11.12. 状态匹配寄存器(OSPI_STATMATCH) 911
27.11.13. 间隔寄存器(OSPI_INTERVAL) 912
27.11.14. 传输配置寄存器 0(OSPI_TCFG0) 912
27.11.15. 时序配置寄存器 0(OSPI_TIMCFG0) 915
27.11.16. 指令寄存器 0(OSPI_INS0) 915
27.11.17. 交替字节寄存器 0(OSPI_ALTE0) 916

27.11.18. 低功耗超时寄存器(OSPI_LPOUT) 916
27.11.19. 回卷传输配置寄存器 0(OSPI_WPTCFG0) 917
27.11.20. 回卷时序配置寄存器 0(OSPI_WPTIMCFG0) 919
27.11.21. 回卷指令寄存器 0(OSPI_WPINS0) 920
27.11.22. 回卷交替字节寄存器 0(OSPI_WPALTE0) 920
27.11.23. 写入传输配置寄存器 0(OSPI_WTCFG0) 920
27.11.24. 写入时序配置寄存器 0(OSPI_WTIMCFG0) 923
27.11.25. 写入指令寄存器 0(OSPI_WINS0) 923
27.11.26. 写入交替字节寄存器 0(OSPI_WALTE0) 924
27.11.27. HyperBusTM 延迟配置寄存器(OSPI_HBLCFG) 924
27.11.28. 输配置寄存器 1(OSPI_TCFG1) 925
27.11.29. 时序配置寄存器 1(OSPI_TIMCFG1) 927
27.11.30. 指令寄存器 1(OSPI_INS1) 928
27.11.31. 交替字节寄存器 1(OSPI_ALTE1) 928
27.11.32. 回卷传输配置寄存器 1(OSPI_WPTCFG1) 929
27.11.33. 回卷时序配置寄存器 1(OSPI_WPTIMCFG1) 931
27.11.34. 回卷指令寄存器 1(OSPI_WPINS1) 931
27.11.35. 回卷交替字节寄存器 1(OSPI_WPALTE1) 932
27.11.36. 写入传输配置寄存器 1(OSPI_WTCFG1) 932
27.11.37. 写入时序配置寄存器 1(OSPI_WTIMCFG1) 934
27.11.38. 写入指令寄存器 1(OSPI_WINS1) 935
27.11.39. 写入交替字节寄存器 1(OSPI_WALTE1) 935

28. 外部存储器控制器(EXMC) 937

28.1. 简介 937

28.2. 主要特性 937

28.3. 功能描述 937

28.3.1. 结构框图 937
28.3.2. EXMC 访问基本规范 938
28.3.3. 外部设备地址映射 939
28.3.4. NOR/PSRAM 控制器 941
28.3.5. NAND Flash 控制器 950

28.4. EXMC 寄存器 954

28.4.1. NOR/PSRAM 控制器寄存器 954
28.4.2. NAND Flash 控制器寄存器 957

29. 控制器局域网络(CAN) 963

29.1. 简介 963

29.2. 主要特征 963

29.3. 功能说明 964

29.3.1. 工作模式 964
29.3.2. 通信模式 965
29.3.3. 数据发送 966
29.3.4. 数据接收 968
29.3.5. 过滤功能 969
29.3.6. 时间触发通信 973
29.3.7. 通信参数 973
29.3.8. CAN-FD 操作 975
29.3.9. 传输延迟补偿 975
29.3.10. 错误标志 976
29.3.11. 中断 977

29.4. CAN 寄存器 979

29.4.1. 控制寄存器(CAN_CTL) 979
29.4.2. 状态寄存器(CAN_STAT) 980
29.4.3. 发送状态寄存器(CAN_TSTAT) 982
29.4.4. 接收 FIFO0 寄存器(CAN_RFIFO0) 984
29.4.5. 接收 FIFO1 寄存器(CAN_RFIFO1) 985
29.4.6. 中断使能寄存器(CAN_INTEN) 986
29.4.7. 错误寄存器(CAN_ERR) 988
29.4.8. 位时序寄存器(CAN_BT) 989
29.4.9. FD 控制寄存器(CAN_FDCTL) 990
29.4.10. FD 状态寄存器(CAN_FDSTAT) 991
29.4.11. FD 传输延迟补偿寄存器(CAN_FDTDC) 991
29.4.12. 数据位时序寄存器(CAN_DBT) 992
29.4.13. 发送邮箱标识符寄存器(CAN_TMIx)(x = 0…2) 992
29.4.14. 发送邮箱属性寄存器(CAN_TMPx)(x = 0...2) 993
29.4.15. 发送邮箱 data0 寄存器(CAN_TMDATA0x)(x = 0...2) 994
29.4.16. 发送邮箱 data1 寄存器(CAN_TMDATA1x)(x = 0...2) 994
29.4.17. 接收 FIFO 邮箱标识符寄存器(CAN_RFIFOMIx)(x = 0,1) 995
29.4.18. 接收 FIFO 邮箱属性寄存器(CAN_RFIFOMPx)(x = 0,1) 996
29.4.19. 接收 FIFO 邮箱 data0 寄存器(CAN_RFIFOMDATA0x)(x = 0,1) 996
29.4.20. 接收 FIFO 邮箱 data1 寄存器(CAN_RFIFOMDATA1x)(x = 0,1) 997
29.4.21. 过滤器控制寄存器(CAN_FCTL) 997
29.4.22. 过滤器模式配置寄存器(CAN_FMCFG) 998
29.4.23. 过滤器位宽配置寄存器(CAN_FSCFG) 999

29.4.24. 过滤器关联 FIFO 寄存器(CAN_FAFIFO) 1000
29.4.25. 过滤器激活寄存器(CAN_FW) 1001
29.4.26. 过滤器(x)数据(y)寄存器(CAN_FxDATAy)(x = 0…27, y = 0,1) 1001

30. SDIO 接口(SDIO) 1003

30.1. 简介 1003

30.2. 主要特性 1003

30.3. SDIO 总线拓扑 1003

30.4. SDIO 功能描述 1005

30.4.1. SDIO 适配器 1006
30.4.2. APB2 接口 1009

30.5. 卡功能描述 1011

30.5.1. 卡寄存器 1011
30.5.2. 命令 1012
30.5.3. 响应 1020
30.5.4. 数据包格式 1024
30.5.5. 卡的两种状态 1025

30.6. 编程序列 1031

30.6.1. 卡识别 1031
30.6.2. 无数据命令 1033
30.6.3. 单个数据块或多个数据块写 1033
30.6.4. 单个数据块或多个数据块读 1034
30.6.5. 数据流写和数据流读(仅适用于 MMC) 1035
30.6.6. 擦除 1036
30.6.7. 总线宽度选择 1037
30.6.8. 保护管理 1037
30.6.9. 卡上锁/解锁操作 1037

30.7. 特定操作 1039

30.7.1. SD I/O 特定操作 1039
30.7.2. CE-ATA 特定操作 1042

30.8. SDIO 寄存器 1044

30.8.1. 电源控制寄存器(SDIO_PWRCTL) 1044

30.8.2. 时钟控制寄存器(SDIO_CLKCTL) 1044
30.8.3. 命令参数寄存器(SDIO_CMDAGMT) 1046
30.8.4. 命令控制寄存器(SDIO_CMDCTL) 1046
30.8.5. 命令索引响应寄存器(SDIO_RSPCMDIDX) 1047
30.8.6. 响应寄存器(SDIO_RESPx)(x=0,...,3) 1048
30.8.7. 数据超时寄存器(SDIO_DATATO) 1048
30.8.8. 数据长度寄存器(SDIO_DATALEN) 1049
30.8.9. 数据控制寄存器(SDIO_DATACTL) 1049
30.8.10. 数据计数寄存器(SDIO_DATACNT) 1051
30.8.11. 状态寄存器(SDIO_STAT) 1051
30.8.12. 中断清除寄存器(SDIO_INTC) 1053
30.8.13. 中断使能寄存器(SDIO_INTEN) 1054
30.8.14. FIFO 计数寄存器(SDIO_FIFOCNT) 1056
30.8.15. FIFO 数据寄存器(SDIO_FIFO) 1057

31. 通用串行总线全速接口(USBFS) 1058

31.1. 概述 1058

31.2. 主要特性 1058

31.3. 结构框图 1059

31.4. 信号线描述 1059

31.5. 功能描述 1059

31.5.1. USBFS 时钟及工作模式 1059
31.5.2. USB 主机功能 1060
31.5.3. USB 设备功能 1062
31.5.4. 数据 FIFO 1063
31.5.5. 操作手册 1065

31.6. 中断 1068

31.7. USBFS 寄存器 1070

31.7.1. 全局控制与状态寄存器组 1070
31.7.2. 主机控制和状态寄存器组 1087
31.7.3. 设备控制和状态寄存器组 1098
31.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL) 1119

32. 附录 1121

32.1. 寄存器表中使用的缩写列表 1121

32.2. 术语表 1121

32.3. 可用外设 1121

33. 版本历史 1122

图索引

图 1-1. Cortex®-M33 处理器结构框图 43
图 1-2. GD32E53x 系列器件的系统架构示意图 45
图 1-3. ECC 解码器示意图 50
图 2-1. 页擦除操作流程 89
图 2-2. 整片擦除操作流程 90
图 2-3. 四字编程操作流程 92
图 3-1. ICACHE 结构框图 120
图 3-2. ICACHE 重映射区域大小 122
图 4-1. DCACHE 结构框图 131
图 4-2. DCACHE 缓存框图 131
图 4-3. 缓存行 132
图 4-4. 缓存路 133
图 5-1. 电源域概览 146
图 5-2. V_{CORE} 域概览 147
图 5-3. 上电/掉电复位波形图 149
图 5-4. LVD 阈值波形图 150
图 5-5. OVD 阈值波形图 150
图 5-6. BOR 波形图 151
图 5-7. VAUVD 阈值监测波形图 151
图 5-8. VAOVDF 阈值监测波形图 152
图 5-9. VOVD 阈值监测波形图 153
图 5-10. VUVD 阈值监测波形图 153
图 6-1.系统复位电路 169
图 6-2. GD32E533 时钟树 170
图 6-3. GD32E535 和 GD32E53P 时钟树 171
图 6-4. HXTAL 时钟源 173
图 6-5. HXTAL 时钟源在旁路模式下 173
图 7-1. CTC 简介 220
图 7-2. CTC 校准计数器 221
图 8-1. EXTI 框图 233
图 9-1. TRIGSEL 主要组成示例 240
图 10-1. 标准 I/O 端口位的基本结构 276
图 10-2. 输入配置 278
图 10-3. 输出配置 278
图 10-4. 模拟配置 279
图 10-5. 备用功能配置 280
图 10-6. 使用采样窗口进行过滤 281
图 10-7. 输入滤波时钟周期 283
图 11-1. CRC 计算单元框图 304

图 12-1. TRNG 模块框图 309
图 13-1. DATAM 不交换/半字交换 315
图 13-2. DATATM 字节交换/位交换 315
图 13-3. CAU 框图 316
图 13-4. AES ECB 加密 317
图 13-5. AES ECB 解密 317
图 14-1. DATAM 不交换/半字交换 330
图 14-2. DATAM 字节交换/位交换 330
图 14-3. HAU 结构框图 331
图 15-1. DMA 框图 342
图 15-2. 块传输模式下重复地址访问示例。 346
图 15-3. 握手机制 346
图 15-4. DMA 中断逻辑 349
图 16-1. DMAMUX 结构框图 364
图 16-2. 同步模式 366
图 16-3. 通道事件输出 367
图 18-1. ADC 模块框图 391
图 18-2. 单次运行模式 392
图 18-3. 连续运行模式 393
图 18-4. 扫描运行模式，且连续运行模式禁能 394
图 18-5. 扫描运行模式，连续运行模式使能 394
图 18-6. 间断运行模式 395
图 18-7. 自动注入，CTN=1 396
图 18-8. 触发注入 396
图 18-9. 数据锁存 397
图 18-10. ADC_LDATAx 读取序列数据 397
图 18-11. 12 位数据存储模式 399
图 18-12. 6 位数据存储模式 399
图 18-13.在常规序列转换过程中常规触发出现(ITGRECEN = 0, RTGRECEN = 0) 400
图 18-14. 在常规序列转换过程中常规触发出现(RTGRECEN = 1) 400
图 18-15. 20 位到 16 位的结果截断 404
图 18-16. 右移 5 位和取整的数例 404
图 18-17. ADC 同步框图 406
图 18-18. 基于 16 个通道的常规并行模式 407
图 18-19. 4 个通道的注入并行模式 407
图 18-20. 常规序列上的快速交叉模式(两个 ADC 的 CTN=1) 408
图 18-21. 常规序列上的慢速交叉模式 409
图 18-22. 注入交替触发模式：DISIC=0, IL=1 409
图 18-23. 注入交替触发模式：DISIC=1, IL=1 410
图 18-24. 常规并行和交替触发组合模式 411
图 18-25. 在注入序列转换过程中注入触发出现 411
图 18-26. 单通道交叉转换被注入序列中断 411
图 19-1. DAC 结构框图 436

图 19-2. DAC LFSR 算法 438
图 19-3. DAC 三角噪声模式生成的波形 438
图 20-1.比较器框图 453
图 20-2. 比较器迟滞 455
图 20-3. 比较器的输出消隐 455
图 20-4. 比较器噪声滤波和中断操作 456
图 21-1. 独立看门狗定时器框图 464
图 21-2. 窗口看门狗定时器框图 470
图 21-3. 窗口看门狗定时器时序图 471
图 22-1. RTC 结构框图 476
图 23-1. 高级定时器结构框图 522
图 23-2. 内部时钟分频为 1 时正常模式下的控制电路 523
图 23-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图 524
图 23-4. 向上计数时序图，PSC=0/2 525
图 23-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 525
图 23-6. 向下计数时序图，PSC=0/2 526
图 23-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 527
图 23-8. 中央计数模式计数器时序图 528
图 23-9. 中央计数模式下计数器重复时序图 529
图 23-10. 在向上计数模式下计数器重复时序图 529
图 23-11. 在向下计数模式下计数器重复时序图 530
图 23-12. 通道 0 输入捕获逻辑 530
图 23-13. 输出比较逻辑 532
图 23-14. 三种输出比较模式 533
图 23-15. EAPWM 时序图 534
图 23-16. CAPWM 时序图 534
图 23-17. 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD) 536
图 23-18. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD) 536
图 23-19. 通道 x 输出 PWM(CHxVAL > CHxCOMVAL_ADD) 537
图 23-20. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL) 537
图 23-21. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 538
图 23-22. 复合 PWM 模式下四通道输出 538
图 23.23 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL) 540
图 23.24. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD) 541
图 23.25. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL) 541
图 23-26. 边沿对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00) 543
图 23-27. 中央对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00) 543
图 23-28. 带死区时间的互补输出 546
图 23-29. 不同死区时间的互补输出(DTDIFEN=1) 547
图 23-30. BREAK 中止功能逻辑图 547
图 23-31. 通道响应 BREAK 中止输入(高电平有效)时，输出信号的行为(IOS=1) 548
图 23-32. 通道中止功能逻辑图 549
图 23-33. 译码器接口模式下计数器运行例子 550

图 23-34. CI0FE0 极性反相的译码器接口模式下的例子 550
图 23-35. 霍尔传感器用在 BLDC 电机控制中 551
图 23-36. 两个定时器之间的霍尔传感器时序图 552
图 23-37. 复位模式 553
图 23-38. 暂停模式下的控制电路 553
图 23-39. 事件模式 554
图 23-40. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 554
图 23-41. 用定时器 2 的使能信号触发定时器 0 555
图 23-42. 用定时器 2 的更新事件来触发定时器 0 556
图 23-43. 用定时器 2 的使能来选通定时器 0 557
图 23-44. 用定时器 2 的 O0CPRE 信号选通定时器 0 557
图 23-45. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 558
图 23.46. 用主计数器的 trigger 信号触发 4 个计数器 559
图 23.47. CINITDIR 为 0 或 1 时的计数方向 559
图 23-48. 通用定时器 L0 结构框图 604
图 23-49. 内部时钟分频为 1 时正常模式下的控制电路 605
图 23-50. 当预分频器的参数从 1 变到 2 时，计数器的时序图 606
图 23-51. 向上计数时序图，PSC=0/2 607
图 23-52. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 607
图 23-53. 向下计数时序图，PSC=0/2 608
图 23-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 609
图 23-55. 中央计数模式计数器时序图 610
图 23-56. 输入捕获逻辑 611
图 23-57. 输出比较逻辑(x=0,1,2,3) 612
图 23-58. 三种输出比较模式 613
图 23-59. EAPWM 时序图 614
图 23-60. CAPWM 时序图 614
图 23-61. 译码器接口模式下计数器运行例子 616
图 23-62. CI0FE0 极性反相的译码器接口模式下的例子 616
图 23-63. 正交译码器信号断线检测框图 617
图 23-64. 译码器模式 0/1 计数器运行实例(CH1P=0) 618
图 23-65. 译码器模式 2 / 3 计数器运行实例(CH0P / CH1P=0) 618
图 23-66. 复位模式 620
图 23-67. 暂停模式 620
图 23-68. 事件模式 621
图 23-69. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 621
图 23-70. 通用定时器 L2 结构框图 647
图 23-71. 内部时钟分频为 1 时，计数器的时序图 648
图 23-72. 当 PSC 数值从 0 变到 2 时，计数器的时序图 649
图 23-73. 向上计数时序图，PSC=0/2 650
图 23-74. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 650
图 23-75. 通道输入捕获原理 651
图 23-76. 通道输出比较原理 652

图 23-77. 三种输出比较模式 653
图 23-78. 通用定时器 L3 结构框图 666
图 23-79. 内部时钟分频为 1 时正常模式下的控制电路 667
图 23-80. 当预分频器的参数从 1 变到 2 时，计数器的时序图 668
图 23-81. 向上计数时序图，PSC=0/2 669
图 23-82. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 669
图 23-83. 向下计数时序图，PSC=0/2 670
图 23-84. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 671
图 23-85. 中央计数模式计数器时序图 672
图 23-86. 中央计数模式下计数器重复时序图 673
图 23-87. 在向上计数模式下计数器重复时序图 673
图 23-88. 在向下计数模式下计数器重复时序图 674
图 23-89. 通道 0 输入捕获逻辑 675
图 23-90. 多模式通道 0 输入捕获逻辑 675
图 23-91. 输出比较逻辑(当 MCHxMSEL = 2'00 时，x=0) 676
图 23-92. 输出比较逻辑(当 MCHxMSEL = 2'11 时，x=0) 677
图 23-93. 输出比较逻辑(x=1) 677
图 23-94. 三种输出比较模式 679
图 23-95. EAPWM 时序图 680
图 23-96. CAPWM 时序图 680
图 23-97. 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD) 682
图 23-98. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD) 682
图 23-99. 通道 x 输出 PWM(CHxVAL > CHxCOMVAL_ADD) 683
图 23-100. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL) 683
图 23-101. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变 684
图 23-102. 边沿对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00) 685
图 23-103. 中央对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00) 685
图 23-104. 带死区时间的互补输出 688
图 23-105. BREAK 中止功能逻辑图 689
图 23-106. 通道响应 BREAK 中止输入(高电平有效)时，输出信号的行为(IOS=1) 690
图 23-107. 复位模式下的控制电路 691
图 23-108. 暂停模式下的控制电路 691
图 23-109. 事件模式下的控制电路 691
图 23-110. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 692
图 23-111. 基本定时器结构框图 726
图 23-112. 内部时钟分频为 1 时正常模式下的控制电路 727
图 23-113. 当预分频器的参数从 1 变到 2 时，计数器的时序图 727
图 23-114. 向上计数时序图，PSC=0/2(TIMERx, x=5,6) 728
图 23-115. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值(TIMERx, x=5,6) 729
图 24-1. USART 模块内部框图 737
图 24-2. USART 字符帧（8 数据位和 1 停止位) 737
图 24-3. USART 发送步骤 739
图 24-4. 过采样方式接收一个数据位(OSB=0) 740

图 24-5. 采用 DMA 方式实现 USART 数据发送配置步骤 741
图 24-6. 采用 DMA 方式实现 USART 数据接收配置步骤 742
图 24-7. 两个 USART 之间的硬件流控制 742
图 24-8. 硬件流控制 743
图 24-9. 空闲状态下检测断开帧 744
图 24-10. 数据传输过程中检测断开帧 744
图 24-11. 同步模式下的 USART 示例 745
图 24-12. 8-bit 格式的 USART 同步通信波形(CLEN=1) 745
图 24-13. IrDA SIR ENDEC 模块 746
图 24-14. IrDA 数据调制 746
图 24-15. ISO7816-3 数据帧格式 747
图 24-16. USART 中断映射框图 749
图 24-17.红外调制波形示意图 750
图 24-18. USART 模块内部框图 768
图 24-19. USART 字符帧(8 数据位和 1 停止位) 768
图 24-20. USART 发送步骤 770
图 24-21. 过采样方式接收一个数据位(OSB=0) 771
图 24-22. 采用 DMA 方式实现 USART 数据发送配置步骤 772
图 24-23. 采用 DMA 方式实现 USART 数据接收配置步骤 773
图 24-24. 两个 USART 之间的硬件流控制 773
图 24-25. 硬件流控制 774
图 24-26. 空闲状态下检测断开帧 775
图 24-27. 数据传输过程中检测断开帧 776
图 24-28. 同步模式下的 USART 示例 776
图 24-29. 8-bit 格式的 USART 同步通信波形(CLEN=1) 776
图 24-30. IrDA SIR ENDEC 模块 777
图 24-31. IrDA 数据调制 777
图 24-32. ISO7816-3 数据帧格式 778
图 24-33. USART 接收 FIFO 结构 780
图 24-34. USART 发送 FIFO 结构 781
图 24-35. USART 中断映射框图 782
图 25-1. I2C 模块框图 806
图 25-2. 数据有效性 807
图 25-3. 开始和停止信号 808
图 25-4. 10 位地址的 I2C 通讯流程(主机发送) 808
图 25-5. 7 位地址的 I2C 通讯流程(主机发送) 809
图 25-6. 7 位地址的 I2C 通讯流程(主机接收) 809
图 25-7. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=0) 809
图 25-8. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1) 809
图 25-9. 数据保持时间 810
图 25-10. 数据建立时间 810
图 25-11. 数据发送 812
图 25-12. 数据接收 813

图 25-13. I2C 从机初始化 816
图 25-14. I2C 从机发送编程模型(SS=0) 817
图 25-15. I2C 从机发送编程模型(SS=1) 818
图 25-16. I2C 从机接收编程模型 819
图 25-17. I2C 主机初始化 820
图 25-18. I2C 主机发送编程模型(N<=255) 821
图 25-19. I2C 主机发送编程模型(N>255) 822
图 25-20. I2C 主机接收编程模型(N<=255) 823
图 25-21. I2C 主机接收编程模型(N>255) 824
图 25-22. SMBus 主机发送器和从机接收器通信流程 827
图 25-23. SMBus 主机接收器和从机发送器通信流程 828
图 26-1. SPI 结构框图 845
图 26-2. 常规模式下的 SPI 时序图 846
图 26-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0) 847
图 26-4. 典型的全双工模式连接 849
图 26-5. 典型的单工模式连接(主机：接收，从机：发送) 850
图 26-6. 典型的单工模式连接(主机：只发送，从机：接收) 850
图 26-7. 典型的双向线连接 850
图 26-8. 主机 TI 模式在不连续发送时的时序图 852
图 26-9. 主机 TI 模式在连续发送时的时序图 852
图 26-10. 从机 TI 模式时序图 852
图 26-11. NSS 脉冲模式时序图(主机连续发送) 853
图 26-12. SPI 四线模式四线写操作时序图 854
图 26-13. SPI 四路模式四路读操作时序图 855
图 26-14. I2S 结构框图 858
图 26-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0) 859
图 26-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1) 859
图 26-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0) 860
图 26-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1) 860
图 26-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 860
图 26-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 860
图 26-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 861
图 26-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 861
图 26-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0) 861
图 26-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1) 861
图 26-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0) 861
图 26-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1) 862
图 26-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 862
图 26-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 862
图 26-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 862
图 26-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 862
图 26-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 863
图 26-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 863

图 26-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 863
图 26-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 863
图 26-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) 864
图 26-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) 864
图 26-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) 864
图 26-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) 864
图 26-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) 864
图 26-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) 864
图 26-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) 865
图 26-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) 865
图 26-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) 865
图 26-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) 865
图 26-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) 865
图 26-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) 865
图 26-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) 866
图 26-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) 866
图 26-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) 866
图 26-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) 866
图 26-51. I2S 时钟生成结构框图 866
图 26-52. I2S 初始化流程 869
图 26-53. I2S 主机接收禁能流程 871
图 27-1. OSPI 八线通信模式结构框图 886
图 27-2. OSPI 四线通信模式结构框图 887
图 27-3. OSPI 双片选通信模式结构框图 887
图 27-4. 八线模式 OSPI 命令格式 888
图 27-5. 四线模式 OSPI 命令格式 888
图 27-6. 八线模式下使用 DQS 的 DTR 读时序 890
图 27-7. CSN 和 SCK 的行为 891
图 27-8. 具有初始延迟的 HyperBus™ 读取操作 892
图 27-9. 具有额外延迟的 HyperBus™ 读取操作 893
图 27-10. 超出页界限的读取操作 893
图 27-11. 具有初始延迟的 HyperBus™ 写入操作 893
图 27-12. 具有额外延迟的 HyperBus™ 写入操作 894
图 27-13. 无延迟的 HyperBus™ 写入操作 894
图 28-1. 系统架构 938
图 28-2. EXMC Bank 划分 939
图 28-3. Bank0 地址映射 940
图 28-4. NAND 地址映射 940
图 28-5. Bank1 通用空间 941
图 28-6. 复用模式读时序 944
图 28-7. 复用模式写时序 944
图 28-8. 异步等待有效时的读时序 946
图 28-9. 异步等待有效时的写时序 946

图 28-10. 同步复用突发传输读时序 948
图 28-11. 同步复用突发传输写时序 949
图 28-12. NAND flash 通用空间操作时序 952
图 28-13. NCE 敏感 NAND Flash 访问时序 953
图 29-1. CAN 模块结构框图 964
图 29-2. 发送寄存器 966
图 29-3. 发送邮箱状态转换 967
图 29-4. 接收寄存器 968
图 29-5. 32-bit 位宽过滤器 970
图 29-6. 16-bit 位宽过滤器 970
图 29-7. 32-bit 位宽掩码模式过滤器 970
图 29-8. 16-bit 位宽掩码模式过滤器 970
图 29-9. 32-bit 位宽列表模式过滤器 971
图 29-10. 16-bit 位宽列表模式过滤器 971
图 29-11. 位时序 974
图 29-12 传输延迟测量 976
图 30-1. SDIO "无响应" 和 "无数据" 操作 1004
图 30-2. SDIO 多块读操作 1004
图 30-3. SDIO 多块写操作 1005
图 30-4. SDIO 数据流读操作 1005
图 30-5. SDIO 数据流写操作 1005
图 30-6. SDIO 框图 1006
图 30-7. 命令标记格式 1012
图 30-8. 响应令牌格式 1021
图 30-9. 1 位数据总线宽度 1024
图 30-10. 4 位数据总线宽度 1024
图 30-11. 8 位数据总线宽度 1025
图 30-12. 通过停止 SDIO_CLK 的读等待操作 1040
图 30-13. 使用 SDIO_DAT[2]信号线的读等待操作 1040
图 30-14. 在功能 1 的多块读周期期间插入功能 2 读周期 1041
图 30-15. 读中断周期时序 1041
图 30-16. 写中断周期时序 1042
图 30-17. 4 位模式下多块读中断周期时序 1042
图 30-18. 4 位模式下多块写中断周期时序 1042
图 30-19. 命令完成信号关闭操作 1043
图 31-1. USBFS 结构框图 1059
图 31-2. 在主机或设备模式下连接示意图 1060
图 31-3. 主机端口状态转移图 1060
图 31-4. 主机模式 FIFO 空间 1063
图 31-5. 主机模式 FIFO 访问寄存器映射表 1064
图 31-6. 设备模式 FIFO 空间 1064
图 31-7. 设备模式 FIFO 访问寄存器映射表 1065

表索引

表 1-1. AHB 互联矩阵的互联关系列表 43
表 1-2. GD32E53x 系列器件的存储器映射表 46
表 1-3. 引导模式 52
表 2-1. 闪存基地址和构成 84
表 2-2. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E533) 86
表 2-3. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E535 和 GD32E53P) 86
表 2-4. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E537) 86
表 2-5. OTP 字节 93
表 2-6. 选项字节 96
表 2-7. OB_WP 位用于受保护页面 101
表 2-8. SCR/WP 擦除保护 102
表 3-1. 两路组相连模式下 TAG 存储标示尺寸参数 121
表 3-2. ICACHE 重映射区域大小 122
表 3-3. ICACHE 缓存事务特性 123
表 3-4. 存储配置 123
表 3-5. ICACHE 中断 124
表 4-1. 两路组相连模式下 TAG 存储标示尺寸参数 133
表 4-2. DCACHE 事务的缓存能力 134
表 4-3. 读操作 135
表 4-4. 写回 135
表 4-5. 写通 136
表 5-1. 节电模式总结 155
表 6-1.时钟输出源选择 175
表 8-1. Cortex®-M33 中的 NVIC 异常类型 229
表 8-2. 中断向量表 230
表 8-3. EXTI 触发源 234
表 9-1.触发输入位域选择 240
表 9-2. TRIGSEL 输入输出映射关系 243
表 10-1. GPIO 配置表 275
表 15-1. DMA 传输操作 343
表 15-2. DMA 传输操作 345
表 15-3. 中断事件 348
表 16-1. DMAMUX 信号 365
表 16-2. 中断事件 369
表 16-3. DMAMUX 请求路由输入信号映射 369
表 16-4. 触发输入信号映射 373
表 16-5. 同步输入信号映射 374
表 17-1. 引脚分配 380
表 17-2. 调试模式与 OTP 配置 380

表 18-1. ADC 内部信号 390
表 18-2. ADC 引脚定义 390
表 18-3. 外部触发模式和触发类型 400
表 18-4. 不同分辨率对应的 t_{CONV} 时间 403
表 18-5. N 和 M 的最大输出值(灰色部分表示截断) 404
表 18-6. ADC 同步模式表 405
表 19-1. DAC 引脚 436
表 19-2. DAC 触发与输出 436
表 19-3. DAC 外部触发 437
表 20-1. 输入和输出汇总 454
表 21-1. 独立看门狗定时器在 40kHz(IRC40K)时的最小/最大超时周期 464
表 21-2. 在 100MHz(f_{PCLK1})时的最大/最小超时值 471
表 22-1. RTC 引脚配置和功能 488
表 22-2. 省电模式管理 489
表 22-3. RTC 中断控制 490
表 23-1. 定时器(TIMERx)分为 4 种类型 519
表 23-2 复合 PWM 脉冲宽度 535
表 23-3 非对称 PWM 脉冲宽度 539
表 23-4. 由参数控制的互补输出表 544
表 23-5. BREAK 输入信号时，TIMER 互补通道输出情况(break 输入高电平有效) 548
表 23-6. 不同译码器模式下的计数方向 549
表 23-7. 从模式例子列表 552
表 23-8. 不同译码器模式下的计数方向 615
表 23-9. 译码器模式 1 的计数情况 617
表 23-10. 译码器模式 2 / 3 下的计数方向 618
表 23-11. 从模式列表和举例(通用定时器 L0) 619
表 23-12. 复合 PWM 脉冲宽度 681
表 23-13. 由参数控制的互补输出表(MCHxMSEL=2'b11) 686
表 23-14. 从模式例子列表 690
表 24-1. USART 重要引脚描述 736
表 24-2. 停止位配置 737
表 24-3. USART 中断请求 748
表 24-4. USART 重要引脚描述 767
表 24-5. 停止位配置 768
表 24-6. USART 中断请求 781
表 25-1. I2C 总线术语说明(参考飞利浦 I2C 规范) 806
表 25-2. 数据建立时间和数据保持时间 811
表 25-3. 可关闭通信模式 813
表 25-4. I2C 错误标志 829
表 25-5. I2C 中断事件 829
表 26-1. SPI 信号描述 845
表 26-2. SPI 四线信号描述 846
表 26-3. 从机模式 NSS 功能 847

表 26-4. 主机模式 NSS 功能 848
表 26-5. SPI 运行模式 848
表 26-6. SPI 中断请求 858
表 26-7. I2S 比特率计算公式 867
表 26-8. 音频采样频率计算公式 867
表 26-9. 各种运行模式下 I2S 接口信号的方向 867
表 26-10. I2S 中断 873
表 27-1. OSPI 信号线描述 884
表 27-2. HyperBus™ 模式下信号线描述 891
表 27-3. HyperBus™ 模式下指令/地址阶段说明 892
表 27-4. SPI 中断请求 900
表 28-1. NOR Flash 接口信号描述 942
表 28-2. PSRAM 复用接口信号描述 942
表 28-3. EXMC 的 Bank0 支持的访问模式 942
表 28-4. NOR/PSRAM 控制时序参数 943
表 28-5. EXMC 时序模型 943
表 28-6. 复用模式相关寄存器配置 944
表 28-7. 同步复用模式读时序配置 948
表 28-8. 同步复用模式写时序配置 949
表 28-9. 8 位/16 位 NAND 接口信号描述 950
表 28-10. Bank1 支持的访问模式 950
表 28-11. NADN Flash 可编程参数 951
表 29-1. 32-bit 过滤序号 971
表 29-2. 过滤索引 972
表 29-3. CAN 事件/中断标志 978
表 30-1. SDIO I/O 定义 1006
表 30-2. 命令格式 1012
表 30-3. 卡命令类 (CCCs) 1013
表 30-4. 基本命令(class 0) 1015
表 30-5. 面向块的读命令(class 2) 1016
表 30-6. 流读取命令(class 1)和流写入命令(class 3) 1016
表 30-7. 面向块的写命令(class 4) 1017
表 30-8. 擦除命令(class 5) 1017
表 30-9. 面向块的写保护命令(class 6) 1018
表 30-10. 锁卡命令(class 7) 1018
表 30-11. 特定应用命令(class 8) 1018
表 30-12. I/O 模式命令(class 9) 1019
表 30-13. 切换功能命令(class 10) 1020
表 30-14. R1 响应 1021
表 30-15. R2 响应 1022
表 30-16. R3 响应 1022
表 30-17. R4 响应(MMC) 1022
表 30-18. R4 响应(SD I/O) 1022

表 30-19. R5 响应(MMC) 1023
表 30-20. R5 响应(SD I/O) 1023
表 30-21. R6 响应 1023
表 30-22. R7 响应 1024
表 30-23. 卡状态 1026
表 30-24. SD 状态 1027
表 30-25. 移动性能字段 1029
表 30-26. AU_SIZE 字段 1029
表 30-27. 最大 AU 大小 1030
表 30-28. 擦除大小字段 1030
表 30-29. 擦除超时字段 1030
表 30-30. 擦除偏移字段 1031
表 30-31. 上锁/解锁数据结构 1038
表 30-32. 不同响应类型对应的 SDIO_RESPx 寄存器 1048
表 31-1. USBFS 信号线描述 1059
表 31-2. USBFS 全局中断 1068
表 32-1. 寄存器功能位访问属性 1121
表 32-2. 术语 1121
表 33-1. 版本历史 1122

1. 系统及存储器架构

GD32E53x 系列器件是基于 Arm® Cortex®-M33 处理器的 32 位通用微控制器。Arm® Cortex®-M33 处理器包括数据总线和系统总线这两条 AHB 总线。Cortex®-M33 处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这两条 AHB 总线上执行。存储器的组织采用了哈佛结构，预先定义的存储器映射和高达 4GB 的存储空间，充分保证了系统的灵活性和可扩展性。

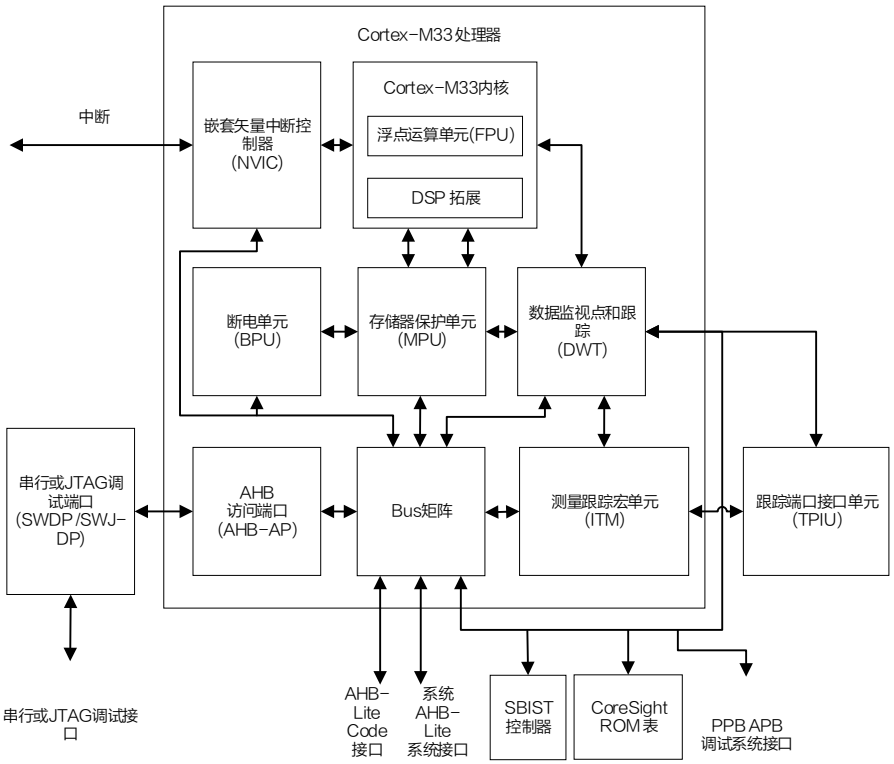
1.1. Arm® Cortex®-M33 处理器

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

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

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

图 1-1. Cortex®-M33 处理器结构框图



1.2. 系统架构

GD32E53x 系列器件采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、一个 AHB 总线和两个 APB 总线。AHB 互联矩阵的互联关系接下来将进行说明。在*表 1-1. AHB 互联矩阵的互联关系列表*中，"1" 表示相应的主机可以通过 AHB 互联矩阵访问对应的从机，空白的单元格表示相应的主机不可以通过 AHB 互联矩阵访问对应的从机。

表 1-1. AHB 互联矩阵的互联关系列表

	CBUS-F	CBUS-S	SBUS-F	SBUS-S	DMA0	DMA1
FMC					1	1
SRAM0	1		1		1	1
SRAM1	1		1		1	1
AHB			1		1	1
EXMC		1		1	1	1
OSPI		1		1	1	1

如*表 1-1. AHB 互联矩阵的互联关系列表*所示，AHB 互联矩阵共连接多个主机，分别为：

CBUS(CBUS-S \ CBUS-F)、SBUS(SBUS-S \ SBUS-F)、DMA0、DMA1。CBUS 是 Cortex®-M33 内核的数据总线，用于从代码区域中取指令和向量。SBUS 是 Cortex®-M33 内核的系统总线，用于数据加载和存储以及系统区域的调试访问。系统区域包括内部 SRAM 区域和外设区域。DMA0 和 DMA1 分别是 DMA0 和 DMA1 的总线。

GD32E53x 系列芯片内置了主机端口超时监测机制，用于检测系统总线传输超时情况。

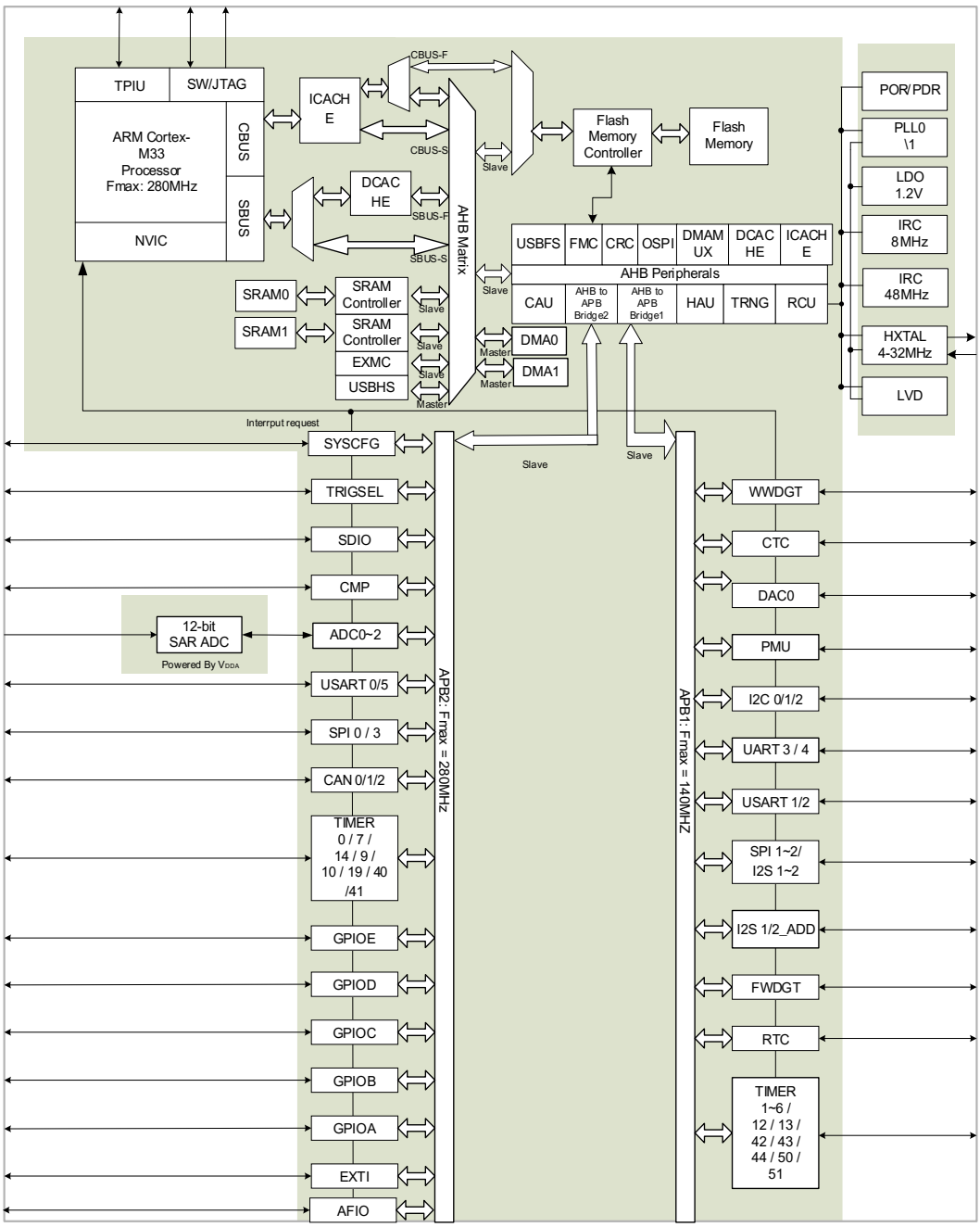
系统对 6 个 AHB 主机端口(CBUS-F、CBUS-S、SBUS-F、SBUS-S、DMA0、DMA1)进行超时监测。当访问异常导致长时间卡住总线访问时，硬件自动产生总线错误反馈给相应的 master，同时 SYSCFG 模块设置相应的超时状态标志位并触发 NMI 中断进行异常处理。

通过总线超时寄存器(SYSCFG_BUSTO)使能对应的主机端口超时监测功能，通过总线超时状态寄存器(SYSCFG_BUSTOSTAT)查看各主机端口的超时状态标志位。

AHB 互联矩阵也连接了多个从机，分别为：FMC、SRAM0\1、EXMC、AHB 和 OSPI。FMC 是闪存存储器控制器。SRAM 是片上静态随机存取存储器。EXMC 是外部存储器控制器。AHB 连接所有 AHB 从机的 AHB 总线，而 APB1 和 APB2 是连接所有 APB 从机的两条 APB 总线。APB1 最高可达 140MHz，APB2 可以全速运行(最高可到 280MHz)。

GD32E53x 系列器件的系统架构如下图所示*图 1-2. GD32E53x 系列器件的系统架构示意图*。

图 1-2. GD32E53x 系列器件的系统架构示意图



1.3. 存储器映射

Arm® Cortex®-M33 处理器采用哈佛架构，可以使用单独的总线来提取指令和加载/存储数据。程序存储器，数据存储器，寄存器和 I/O 端口组织在同一线性 4GB 地址空间内，这是 Cortex®-M33 的最大地址范围，因为总线地址宽度为 32 位。此外，Cortex®-M33 处理器提供了预定义的内存映射，以减少不同客户在相同应用时的软件复杂度。在内存映射表中，Arm® Cortex®-M33 系统外围设备使用的某些区域无法修改。但是，其他区域可供供应商使用。*表 1-2. GD32E53x 系列器件的存储器映射表*显示了 GD32E53x 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. GD32E53x 系列器件的存储器映射表

预先定义的地址空间	总线	地址	外设
外部设备	AHB3	0xC000 0000 - 0xDFFF FFFF	保留
		0xB000 0000 - 0xBFFF FFFF	OSPI(MEM)
		0xA000 1400 - 0xAFFF FFFF	保留
		0xA000 1000 - 0xA000 13FF	保留
		0xA000 0000 - 0xA000 0FFF	EXMC - SWREG
外部 RAM		0x9000 0000 - 0x9FFF FFFF	保留
		0x7000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SRAM
外设	AHB1	0x5004 0000 - 0x5FFF FFFF	保留
		0x5000 0000 - 0x5003 FFFF	USBFS
		0x4008 0400 - 0x4FFF FFFF	保留
		0x4008 0000 - 0x4008 03FF	保留
		0x4004 0000 - 0x4007 FFFF	保留
		0x4002 BC00 - 0x4003 FFFF	保留
		0x4002 B000 - 0x4002 BBFF	保留
		0x4002 A000 - 0x4002 AFFF	保留
		0x4002 8000 - 0x4002 9FFF	保留
		0x4002 6800 - 0x4002 7FFF	保留
		0x4002 6400 - 0x4002 67FF	DCACHE
		0x4002 6000 - 0x4002 63FF	ICACHE
		0x4002 5000 - 0x4002 5FFF	OSPI(REG)
		0x4002 4000 - 0x4002 4FFF	BKPSRAM
		0x4002 3C00 - 0x4002 3FFF	TRNG
		0x4002 3800 - 0x4002 3BFF	HAU
		0x4002 3400 - 0x4002 37FF	CAU
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2C00 - 0x4002 2FFF	保留
		0x4002 2800 - 0x4002 2BFF	保留
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX

预先定义的地址空间	总线	地址	外设
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 9C00 - 0x4001 FFFF	保留
	APB2	0x4001 9800 - 0x4001 9BFF	SYSCFG
		0x4001 9400 - 0x4001 97FF	Shared APB/CAN2 SRAM
		0x4001 9000 - 0x4001 93FF	TRIGSEL
		0x4001 8D00 - 0x4001 8FFF	保留
		0x4001 8C00 - 0x4001 8CFF	Shared APB/CAN SRAM (256 bytes)
		0x4001 8800 - 0x4001 8BFF	Shared APB/CAN SRAM
		0x4001 8400 - 0x4001 87FF	SPI3
		0x4001 8000 - 0x4001 83FF	SDIO
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	CAN2
		0x4001 7000 - 0x4001 73FF	USART5
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	保留
		0x4001 5C00 - 0x4001 67FF	保留
		0x4001 5800 - 0x4001 5BFF	保留
		0x4001 5400 - 0x4001 57FF	TIMER10
		0x4001 5000 - 0x4001 53FF	TIMER9
		0x4001 4C00 - 0x4001 4FFF	TIMER40
		0x4001 4800 - 0x4001 4BFF	TIMER19
		0x4001 4400 - 0x4001 47FF	TIMER41
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	ADC2
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	CAN1
		0x4001 1C00 - 0x4001 1FFF	CAN0
		0x4001 1800 - 0x4001 1BFF	GPIOE
		0x4001 1400 - 0x4001 17FF	GPIOD
		0x4001 1000 - 0x4001 13FF	GPIOC

预先定义的地址空间	总线	地址	外设
		0x4001 0C00 - 0x4001 0FFF	GPIOB
		0x4001 0800 - 0x4001 0BFF	GPIOA
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	AFIO
	APB1	0x4000 F400 - 0x4000 FFFF	保留
		0x4000 F000 - 0x4000 F3FF	TIMER51
		0x4000 EC00 - 0x4000 EFFF	TIMER50
		0x4000 E800 - 0x4000 EBFF	TIMER44
		0x4000 E400 - 0x4000 E7FF	TIMER43
		0x4000 E000 - 0x4000 E3FF	TIMER42
		0x4000 D000 - 0x4000 DFFF	保留
		0x4000 CC00 - 0x4000 CFFF	CTC1
		0x4000 C800 - 0x4000 CBFF	CTC0
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	保留
		0x4000 6800 - 0x4000 6BFF	保留
		0x4000 6400 - 0x4000 67FF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	I2S2_ADD
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	I2S1_ADD
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13

预先定义的地址空间	总线	地址	外设
		0x4000 1C00 - 0x4000 1FFF	TIMER12
		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	AHB	0x2007 0000 - 0x3FFF FFFF	保留
		0x2006 0000 - 0x2006 FFFF	保留
		0x2003 0000 - 0x2005 FFFF	保留
		0x2002 0000 - 0x2002 FFFF	SRAM1
		0x2000 0000 - 0x2001FFFF	SRAM0
Code	AHB	0x1FFF F850 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F84F	Option Bytes
		0x1FFF F000 - 0x1FFF F7FF	Boot loader
		0x1FFF C010 - 0x1FFF EFFF	
		0x1FFF C000 - 0x1FFF C00F	
		0x1FFF B000 - 0x1FFF BFFF	
		0x1FFF 7800 - 0x1FFF AFFF	保留
		0x1FFF 6800 - 0x1FFF 77FF	OTP
		0x1FFF 0000 - 0x1FFF 67FF	保留
		0x1FFE C010 - 0x1FFE FFFF	External memories
		0x1FFE C000 - 0x1FFE C00F	
		0x1001 0000 - 0x1FFE BFFF	
		0x1000 0000 - 0x1000 FFFF	
		0x0F00 0000 - 0x0FFF FFFF	保留
		0x0E03 0000 - 0x0EFF FFFF	保留
		0x0E02 0000 - 0x0E02 FFFF	SRAM1
		0x0E00 0000 - 0x0E01 FFFF	SRAM0
		0x083C 0000 - 0x0DFF FFFF	保留
		0x0830 0000 - 0x083B FFFF	保留
		0x0808 0000 - 0x082F FFFF	保留
		0x0800 0000 - 0x0807 FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	保留
		0x0008 0000 - 0x000F FFFF	保留
		0x0002 0000 - 0x0007 FFFF	External memories
		0x0000 0000 - 0x0001 FFFF	

1.3.1. 片上 SRAM 存储器

GD32E53x 系列微控制器含有高达 128KB 的片上 SRAM0 和 64KB 的片上 SRAM1。支持字节、半字(16 比特)和整字(32 比特)访问。4KB 的 BKPSRAM(备份 SRAM)应用于备份域，即使当 VDD 供电电源掉电时或处于待机模式，该 SRAM 仍可保持其内容。

SRAM 写保护

SRAM0 支持前 64KB 的写保护，并以 1KB 的页面粒度进行写保护。SRAM 页面写保护功能可以通过设置写保护寄存器(SYSCFG_SRAMWP0 、 SYSCFG_SRAMWP1) 中的 SRAM_PxWP(x=0..31)和 SRAM_PxWP(x=32..63)位来启用，写保护功能只能通过系统复位来移除或清除。当 SRAM 设置为写保护时，对其进行操作将会触发 Hardfault。

ECC

当读写前 64KB SRAM0 时，SRAM0 支持 7 比特的 ECC 功能。可纠错 1 比特，发现多比特(两比特)错误。

在读取 SRAM0 之前必须先进行写操作，否则可能会导致 ECC 错误。非对齐的读操作将按照 32 位读操作执行。非对齐的写操作会产生读-改-写过程。例如，当向 SRAM0 写入 16 位数据时，首先会从 SRAM0 中读出另一组 16 位数据，然后将需要写入的 16 位数据与读出的 16 位数据组合成 32 位数据，最后将该 32 位数据一并写入 SRAM。因此，在初始化 SRAM0 时，只能以 32 位宽度进行写操作。

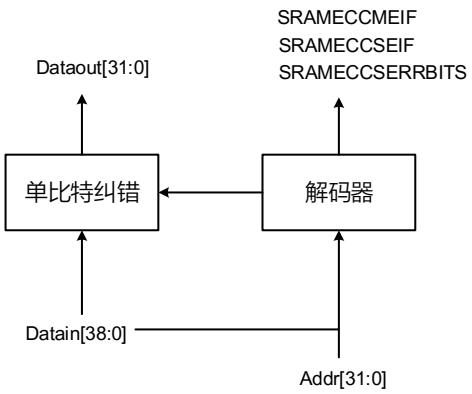
ECC 模块由编码器和解码器两部分构成：

编码器：在进行 SRAM0 写操作时，会产生一个 7 比特的 ECC 码，和数据一起写入 SRAM0。

解码器：在进行 SRAM0 读操作时，使用与编码器相同的算法，解码生成一个 7 比特的 ECC 码。ECC 码包括 ECC 错误状态和 32 位数据中哪位存在单比特位错误的信息。

解码器如下*图 1-3. ECC 解码器示意图*所示：

图 1-3. ECC 解码器示意图



EEIC

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

单比特可纠错事件

当检测到发生了单比特可纠错的事件时，EEIC 模块有如下配置：

(1) 在 SYSCFG_ECCS1 寄存器中的 SRAM0ECCSEIF 位将被置位。软件可以通过写入 1 来清除它。

(2) SYSCFG_ECCCFG0 记录发生单比特校正错误事件的地址。

多比特(两比特)不可纠错事件

当检测到发生了多比特(两比特)不可纠错的事件时，EEIC 模块有如下配置：

(1) 在 SYSCFG_ECCS1 寄存器中的 SRAM0ECCMEIF 位将被置位。软件可以通过写入 1 来清除它。

(2) SYSCFG_ECCCFG0 记录发生多比特非校正错误事件的地址。

单比特可纠错中断

在 SYSCFG_ECCCFG0 寄存器中设置 SRAM0ECCSEIE 位，当检测到一个单比特可纠正错误事件时，将产生 SRAM ECC 中断。

多比特(两比特)不可纠错中断

在 SYSCFG_ECCCFG0 寄存器中设置 SRAM0ECCMEIE 位。当检测到一个多比特(两比特)不可纠错错误事件时，将产生 NMI 和 SRAM ECC 中断。

1.3.2. 片上 FLASH 存储器概述

GD32E53x 系列微控制器可以提供高密度片上 FLASH 存储器，按以下分类进行组织：

- 高达512KB主FLASH存储器；
- 高达18KB引导装载程序(boot loader)信息块存储器；
- 器件配置的选项字节。

更多详细说明请参考*闪存控制器(FMC)*章节。

1.4. 引导配置

GD32E53x系列微控制器提供了三种引导源，可以通过BOOT0引脚和用户选项字节中的引导模式配置位(BOOTLK比特位，nBOOT1比特位，SWBT0比特位，nBOOT0比特位)来进行选择，详细说明*表1-3. 引导模式*。当SWBT0置1时，BOOT0引脚的电平状态会在复位后的第四个CK_SYS(系统时钟)的上升沿进行锁存。上电复位或系统复位后，由用户设置引导模式配置，

选择所需的启动源。当SWBT0置1时，一旦这个引脚电平被采样，它可以被释放并用于其他用途。

表 1-3. 引导模式

引导源选择	启动模式配置				
	BOOTLK	nBOOT1 位	BOOT0 引脚	SWBT0 位	nBOOT0 位
主 Flash 存储器	1	x	x	0	1
System 存储器	1	1	x	0	0
片上 SRAM	1	0	x	0	0
主 Flash 存储器	1	x	0	1	x
System 存储器	1	1	1	1	x
片上 SRAM	1	0	1	1	x
主 Flash 存储器	0	x	x	x	x

上电序列或系统复位后，Arm® Cortex®-M33处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选的引导源，片上闪存的主存(开始于0x0800 0000的存储空间)或系统存储器(开始于0x1FFF B000的存储空间)会被映射到引导空间，即从0x0000 0000开始的地址空间。如果片上SRAM(开始于0x2000 0000的存储空间)被选为引导源，用户必须在应用程序初始化代码中通过修改NVIC异常向量表和偏移寄存器将向量表重置到SRAM中。

嵌入式启动加载程序位于系统内存中，用于重新编程闪存。GD32E53x MCU 嵌入式启动加载程序支持多种接口来更新闪存。在 GD32E53x 系列产品上可以使用 USART0、USART1 端口，以及标准 USB 端口。

无论其他模式如何配置，都可以通过配置BOOTLK位从主Flash存储器的唯一入口进行强制引导。

- 重映射配置

  当选择启动模式配置时，软件可以通过内存重映射配置寄存器(FMC_OBCTL)中的 BIT 位配置内存重映射功能。存储器包括主 Flash 存储器，System 存储器，片上 SRAM 可以重新映射。

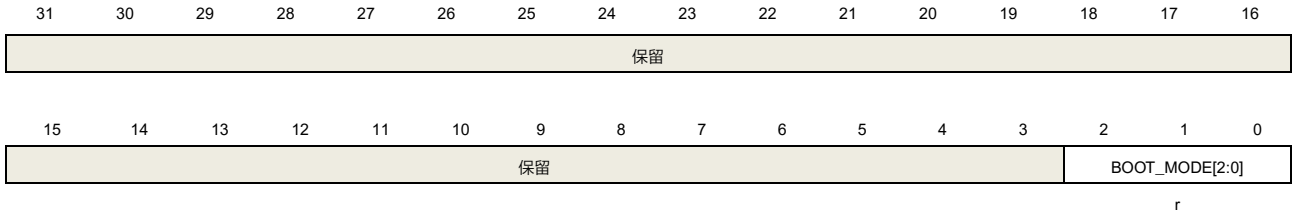
1.5. 系统配置寄存器(SYSCFG)

SYSCFG 基地址：0x4001 9800

1.5.1. 配置寄存器 0 (SYSCFG_CFG0)

地址偏移：0x00
复位值：0x0000 000X (根据 BOOT0 引脚的状态和选项字节中的引导模式配置位(BOOTLK、nBOOT1、SWBT0、nBOOT0)，X 表示 BOOT_MODE[2:0]，可能为任意值)

该寄存器只能按字(32 位)访问。

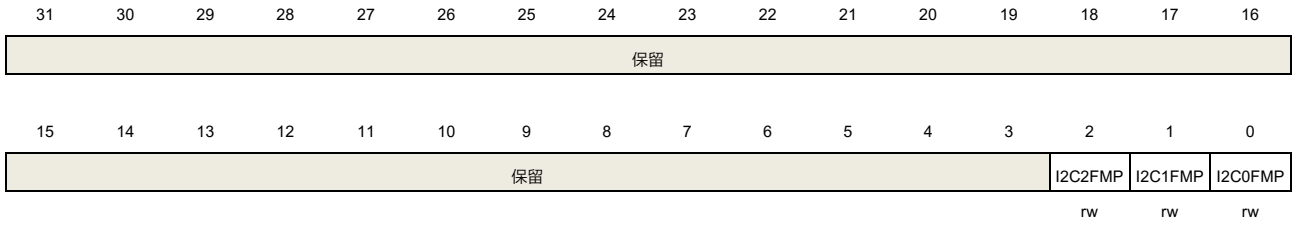


位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	BOOT_MODE [2:0]	引导模式(详细请参考*表1-3. 引导模式*)。

1.5.2. 配置寄存器 1(SYSCFG_CFG1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



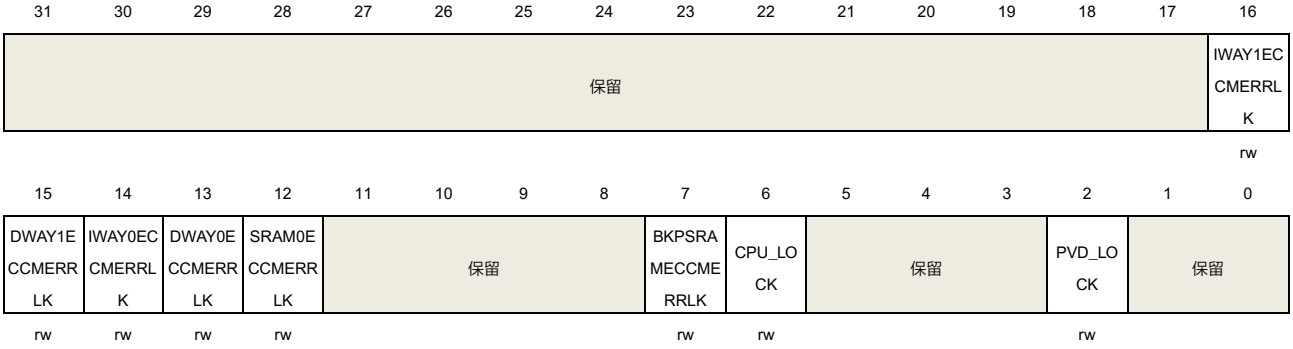
位/位域	名称	描述
31:3	保留	必须保持复位值。
2	I2C2FMP	在 I2C2 上使能 Fast mode+ 0：失能 1：使能
1	I2C1FMP	在 I2C1 上使能 Fast mode+

		0：失能 1：使能
0	I2C0FMP	在 I2C0 上使能 Fast mode+ 0：失能 1：使能

1.5.3. 锁定控制寄存器(SYSCFG_LKCTL)

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
31:17	保留	必须保持复位值。
16	IWAY1ECCMERRLK	ICACHE way1 SRAM ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: ICACHE way1 SRAM ECC多比特不可纠错信号断开与TIMER0/7/14/19/40-44的中止输入 1: ICACHE way1 SRAM ECC 多比特不可纠错信号连接到 TIMER0/7/14/19/40-44 的中止输入
15	DWAY1ECCMERRLK	DCACHE way1 SRAM ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: DCACHE way1 SRAM ECC多比特不可纠错信号断开连接到TIMER0/7/14/19/40-44的中止输入 1: DCACHE way1 SRAM ECC 多比特不可纠错信号连接到 TIMER0/7/14/19/40-44 的中止输入
14	IWAY0ECCMERRLK	ICACHE way0 SRAM ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: ICACHE way0 SRAM ECC多比特不可纠错信号断开连接到TIMER0/7/14/19/40-44的中止输入 1: ICACHE way0 SRAM ECC 多比特不可纠错信号连接到 TIMER0/7/14/19/40-44 的

		中止输入
13	DWAY0ECCMERRLK	DCACHE way0 SRAM ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: DCACHE way0 SRAM ECC多比特不可纠错信号断开连接到TIMER0/7/14/19/40-44的中止输入 1: DCACHE way0 SRAM ECC多比特不可纠错信号连接到TIMER0/7/14/19/40-44的中止输入
12	SRAM0ECCMERRLK	SRAM0 ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: SRAM ECC多比特不可纠错信号与TIMER0/7/14/19/40-44的中止输入断开连接 1: SRAM ECC多比特不可纠错信号与TIMER0/7/14/19/40-44的中止输入连接
11:8	保留	必须保持复位值。
7	BKPSRAMECCMERRLK	BKPSRAM ECC多比特不可纠错锁定位 该位由软件置位，并通过系统复位清除。 0: BKPSRAM ECC多比特不可纠错信号与TIMER0/7/14/19/40-44的中止输入断开连接 1: BKPSRAM ECC 多比特不可纠错信号与 TIMER0/7/14/19/40-44 的中止输入连接
6	CPU_LOCK	CPU 锁定位 该位由软件置位，由系统复位清零。 0：CPU 锁定信号从 TIMER0/7/14/19/40-44 中止输入端断开 1：CPU 锁定信号与 TIMER0/7/14/19/40-44 中止输入端连接
5:3	保留	必须保持复位值。
2	PVD_LOCK	PVD 锁定位 该位由软件置位，由系统复位清零。 0：PVD 信号从 TIMER0/7/14/19/40-44 中止输入端断开 1：PVD 信号与 TIMER0/7/14/19/40-44 中止输入端连接
1:0	保留	必须保持复位值。

1.5.4. 总线超时寄存器(SYSCFG_BUSTO)

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										DMA1BU	DMA0BU	CPUSBU	CPUSBU	CPUCBU	CPUCBU

	STO	STO	SSTO	SFTO	SSTO	SFTO
	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	DMA1BUSTO	DMA1总线超时使能位 0：禁用 1：启用
4	DMA0BUSTO	DMA0总线超时使能位 0：禁用 1：启用
3	CPUSBUSSTO	CPU Sbus-Slow超时使能位 0：禁用 1：启用
2	CPUSBUSFTO	CPU Sbus-Fast超时使能位 0：禁用 1：启用
1	CPUCBUSSTO	CPU Cbus-Slow超时使能位 0：禁用 1：启用
0	CPUCBUSFTO	CPU Cbus-Fast超时使能位 0：禁用 1：启用

1.5.5. SRAM_ECC 错误状态寄存器 0(SYSCFG_ECCS0)

地址偏移：0x20
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DWAY1ECCERRBITS[5:0]						DWAY0ECCERRBITS[5:0]						IWAY1ECCERRBITS[5:2]			
r						r						r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWAY1ECCERRBITS[1:0]		IWAY0ECCERRBITS[5:0]						保留							
r		r													

位/位域	名称	描述
31:26	DWAY1ECCERRBITS[	指示DCACHE way1的错误位

	5:0]	哪个位具有 DCACHE way1 的单比特可纠错错误 000000: 无错误 000001: 第 0 位 … 100111: 第38位
25:20	DWAY0ECCERRBITS[5:0]	指示DCACHE way0的错误位 哪个位具有 DCACHE way0 的单比特可纠错错误。 000000: 无错误 000001: 第 0 位 … 100111: 第 38 位
19:14	IWAY1ECCERRBITS[5:0]	指示ICACHE way1的错误位 哪个位具有 ICACHE way1 的单比特可纠错错误。 000000: 无错误 000001: 第 0 位 … 100111: 第38位
13:8	IWAY0ECCERRBITS[5:0]	指示ICACHE way0的错误位 指出 ICACHE way0 中发生单比特可纠错错误的位。 000000: 无错误 000001: 第 0 位 … 100111: 第38位
7:0	保留	必须保持复位值。

1.5.6. FPU 中断使能寄存器(SYSCFG_FPUINTEN)

地址偏移：0x24

复位值：0x0000 001F

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										IXIE	IDIE	OVFIE	UFIE	DZIE	IOPIE
										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	IXIE	不精确中断使能位 0：不精确中断禁用 1：不精确中断启用
4	IDIE	输入异常中断使能位 0：输入异常中断禁用 1：输入异常中断启用
3	OVFIE	溢出中断使能位 0：溢出中断禁用 1：溢出中断启用
2	UFIE	下溢中断使能位 0：下溢中断禁用 1：下溢中断启用
1	DZIE	除以0中断使能位 0：除以0中断禁用 1：除以0中断启用
0	IOPIE	无效操作中断使能位 0：无效操作中断禁用 1：无效操作中断启用

1.5.7. SRAM 写保护寄存器 0 (SYSCFG_SRAMWP0)

地址偏移：0x28

复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_P31WP	SRAM_P30WP	SRAM_P29WP	SRAM_P28WP	SRAM_P27WP	SRAM_P26WP	SRAM_P25WP	SRAM_P24WP	SRAM_P23WP	SRAM_P22WP	SRAM_P21WP	SRAM_P20WP	SRAM_P19WP	SRAM_P18WP	SRAM_P17WP	SRAM_P16WP
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
SRAM_P15WP	SRAM_P14WP	SRAM_P13WP	SRAM_P12WP	SRAM_P11WP	SRAM_P10WP	SRAM_P9WP	SRAM_P8WP	SRAM_P7WP	SRAM_P6WP	SRAM_P5WP	SRAM_P4WP	SRAM_P3WP	SRAM_P2WP	SRAM_P1WP	SRAM_P0WP
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	SRAM_PxWP(x=0…31)	SRAM页x写保护 这些位由软件设置，仅通过系统复位清除。 0：SRAM页x的写保护已禁用 1：SRAM 页 x 的写保护已启用

1.5.8. SRAM ECC 错误状态寄存器 1 (SYSCFG_ECCS1)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DWAY1ECCSEIF	DWAY1ECCMEIF	DWAY0ECCSEIF	DWAY0ECCMEIF	IWAY1ECCSEIF	IWAY1ECCMEIF	IWAY0ECCSEIF	IWAY0ECCMEIF	BKPECCSEIF	BKPECCMEIF	SRAM0ECCSEIF	SRAM0ECCMEIF
				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

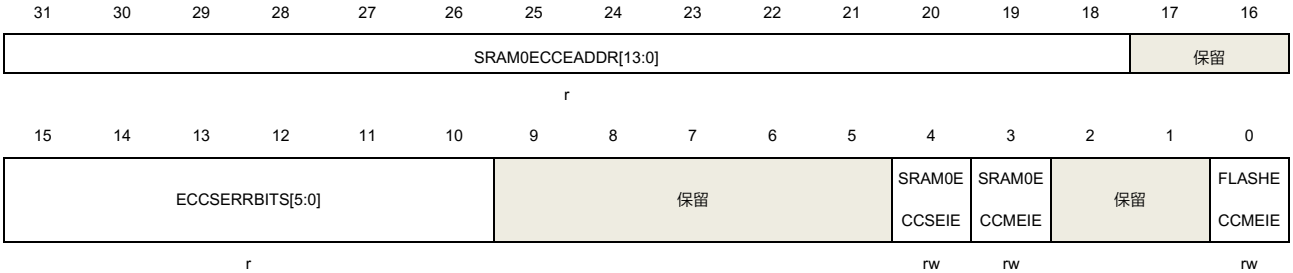
位/位域	名称	描述
31:12	保留	必须保持复位值。
11	DWAY1ECCSEIF	DCACHE way1 SRAM单比特可纠错事件标志 软件可通过写1清除。 0: 未检测到DCACHE way1 SRAM单比特可纠错。 1: 检测到DCACHE way1 SRAM单比特可纠错。
10	DWAY1ECCMEIF	DCACHE way1 SRAM多比特不可纠错事件标志 软件可通过写1清除。 0: 未检测到DCACHE way1 SRAM多比特不可纠错。 1: 检测到DCACHE way1 SRAM多比特不可纠错。
9	DWAY0ECCSEIF	DCACHE way0 SRAM单比特可纠错事件标志 软件可通过写1清除。 0: 未检测到DCACHE way0 SRAM单比特可纠错。 1: 检测到DCACHE way0 SRAM单比特可纠错。
8	DWAY0ECCMEIF	DCACHE way0 SRAM多比特不可纠错事件标志 软件可通过写1清除。 0: 未检测到DCACHE way0 SRAM不可纠错。 1: 检测到DCACHE way0 SRAM不可纠错。
7	IWAY1ECCSEIF	ICACHE way1 SRAM单比特可纠错事件标志 软件可通过写1清除。 0: 未检测到ICACHE way1 SRAM单比特可纠错。 1: 检测到ICACHE way1 SRAM单比特可纠错。
6	IWAY1ECCMEIF	ICACHE way1 SRAM多比特不可纠错事件标志 软件可通过写1清除。 0: 未检测到ICACHE way1 SRAM多比特不可纠错。 1: 检测到ICACHE way1 SRAM多比特不可纠错。

5	IWAY0ECCSEIF	ICACHE way0 SRAM单比特可纠错事件标志 软件可通过写1清除。 0: 未检测到ICACHE way0 SRAM单比特可纠错。 1: 检测到ICACHE way0 SRAM单比特可纠错。
4	IWAY0ECCMEIF	ICACHE way0 SRAM多比特不可纠错事件标志 软件可通过写1清除。 0: 未检测到ICACHE way0 SRAM不可纠错事件。 1: 检测到ICACHE way0 SRAM不可纠错事件。
3	BKPECCSEIF	BKPSRAM单比特可纠错事件标志 软件可通过写1清除。 0: 未检测到BKPSRAM单比特可纠错。 1: 检测到BKPSRAM单比特可纠错。
2	BKPECCMEIF	BKPSRAM多比特不可纠错标志 软件可通过写1清除。 0: 未检测到BKPSRAM不可纠错事件。 1: 检测到BKPSRAM不可纠错事件。
1	SRAM0ECCSEIF	SRAM0 单比特可纠错事件标志 软件可通过写 1 清除。 0: 未检测到 SRAM0 单比特可纠错。 1: 检测到SRAM0单比特可纠错。
0	SRAM0ECCMEIF	SRAM0 多比特不可纠错事件标志 软件可通过写 1 清除。 0: 未检测到 SRAM0 不可纠错事件。 1: 检测到SRAM0不可纠错事件。

1.5.9. ECC 配置寄存器 0 (SYSCFG_ECCCFG0)

地址偏移：0x30
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。



位/位域	名称	描述

31:18	SRAM0ECCEADDR[13:0]	记录上一次 SRAM0 发生 SRAM ECC 事件时的故障系统地址，实际地址 = SRAM0 基地址(0x2000 0000) + SRAM0ECCEADDR × 4。
17:16	保留	必须保持复位值。
15:10	ECCSERRBITS[5:0]	指示错误位 指示哪个位在 SRAM0 中发生了 ECC 单比特可纠错错误。 000000: 无错误 000001: 第 0 位 … 100111: 第 38 位
9:5	保留	必须保持复位值。
4	SRAM0ECCSEIE	SRAM0 单比特可纠错错误中断使能 0：禁能 SRAM0 单比特可纠错错误中断。 1：使能SRAM0单比特可纠错错误中断。
3	SRAM0ECCMEIE	SRAM0 多比特不可纠错错误 NMI 中断使能 0：禁能 SRAM0 多比特不可纠错错误 NMI 中断。 1：使能SRAM0多比特不可纠错错误NMI中断。
2:1	保留	必须保持复位值。
0	FLASHECCMEIE	Flash多比特不可纠错错误NMI中断使能。 0: 禁能Flash多比特不可纠错错误NMI中断。 1: 使能Flash多比特不可纠错错误NMI中断。

1.5.10. SRAM ECC 配置寄存器 1 (SYSCFG_ECCCFG1)

地址偏移：0x34
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。



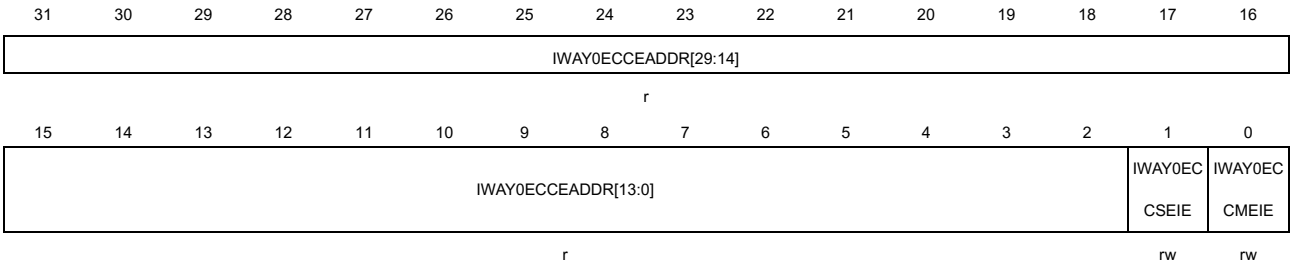
位/位域	名称	描述
31:29	保留	必须保持复位值。
28:19	BKPECCEADDR[9:0]	记录上一次发生 ECC 事件的 BKPSRAM 故障系统地址。 BKPSRAM基地址 = 0x4002 4000

		物理 BKPSRAM ECC 字地址 = BKPSRAM 基地址 + BKPECCEADDR * 4
18:16	保留	必须保持复位值。
15:10	BKPECCERRBITS[5:0]	指示BKPSRAM的错误位。 哪个位具有 BKPSRAM ECC 单比特可纠错错误。 000000: 无错误 000001: 第 0 位 … 100111: 第 38 位
9:5	保留	必须保持复位值。
4	BKPECCSEIE	BKPSRAM 单比特可纠错错误中断使能 0：禁能 BKPSRAM 单比特可纠错错误中断。 1：使能 BKPSRAM 单比特可纠错错误中断。
3	BKPECCMEIE	BKPSRAM 多比特不可纠正错误 NMI 中断使能 0：禁能 BKPSRAM 多比特不可纠错错误 NMI 中断。 1：使能 BKPSRAM 多比特不可纠错错误 NMI 中断。
2:0	保留	必须保持复位值。

1.5.11. SRAM ECC 配置寄存器 2 (SYSCFG_ECCCFG2)

地址偏移：0x38
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。



位/位域	名称	描述
31:2	IWAY0ECCEADDR[29:0]	记录上一次发生 ECC 事件的 ICACHE way0 SRAM 故障系统地址。
1	IWAY0ECCSEIE	ICACHE way0 单比特可纠错错误中断使能 0: 禁能ICACHE way0 SRAM单比特可纠错错误中断。 1: 使能 ICACHE way0 SRAM 单比特可纠错错误中断。
0	IWAY0ECCMEIE	ICACHE way0多比特不可纠错错误NMI中断使能 0: 禁能ICACHE way0 SRAM不可纠错错误NMI中断。

1: 使能 ICACHE way0 SRAM 不可纠错错误 NMI 中断。

1.5.12. SRAM ECC 配置寄存器 3 (SYSCFG_ECCCFG3)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IWAY1ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWAY1ECCEADDR[13:0]														IWAY1ECCSEIE	IWAY1ECCMEIE
r														rw	rw

位/位域	名称	描述
31:2	IWAY1ECCEADDR[29:0]	记录上一次发生 ECC 事件的 ICACHE way1 SRAM 故障系统地址。
1	IWAY1ECCSEIE	ICACHE way1 单比特可纠错错误中断使能 0: 禁能ICACHE way1 SRAM单比特可纠错错误中断。 1: 使能 ICACHE way1 SRAM 单比特可纠错错误中断。
0	IWAY1ECCMEIE	ICACHE way1多比特(两位)不可纠错错误NMI中断使能 0: 禁能ICACHE way1 SRAM多比特(两位)不可纠错错误NMI中断。 1: 使能 ICACHE way1 SRAM 多比特(两位)不可纠错错误 NMI 中断。

1.5.13. 总线超时状态寄存器 (SYSCFG_BUSTOSTAT)

地址偏移：0x40
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										DMA1BUSTOF	DMA0BUSTOF	CPUSBUSSTOF	CPUSBUSFTOF	CPUCBUSSTOF	CPUCBUSFTOF
										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	DMA1BUSTOF	DMA1 总线超时标志 0：DMA1 总线未超时 1：DMA1总线超时
4	DMA0BUSTOF	DMA0 总线超时标志 0：DMA0 总线未超时 1：DMA0总线超时
3	CPUSBUSSTOF	CPU Sbus-Slow 超时标志。 0: CPU Sbus-S 未超时。 1: CPU Sbus-S 超时。
2	CPUSBUSFTOF	CPU Sbus-Fast 超时标志。 0: CPU Sbus-F 未超时。 1: CPU Sbus-F 超时。
1	CPUCBUSSTOF	CPU Cbus-Slow 超时标志。 0: CPU Cbus-S 未超时。 1: CPU Cbus-S超时。
0	CPUCBUSFTOF	CPU Cbus-Fast 超时标志。 0: CPU Cbus-F 未超时 1: CPU Cbus-F超时

1.5.14. SRAM 写保护寄存器 1 (SYSCFG_SRAMWP1)

地址偏移：0x44
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_P 63WP	SRAM_P 62WP	SRAM_P 61WP	SRAM_P 60WP	SRAM_P 59WP	SRAM_P 58WP	SRAM_P 57WP	SRAM_P 56WP	SRAM_P 55WP	SRAM_P 54WP	SRAM_P 53WP	SRAM_P 52WP	SRAM_P 51WP	SRAM_P 50WP	SRAM_P 49WP	SRAM_P 48WP
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
SRAM_P 47WP	SRAM_P 46WP	SRAM_P 45WP	SRAM_P 44WP	SRAM_P 43WP	SRAM_P 42WP	SRAM_P 41WP	SRAM_P 40WP	SRAM_P 39WP	SRAM_P 38WP	SRAM_P 37WP	SRAM_P 36WP	SRAM_P 35WP	SRAM_P 34WP	SRAM_P 33WP	SRAM_P 32WP
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	SRAM_PxWP (x=32…63)	SRAM页x写保护 这些位由软件设置，仅通过系统复位清除。 0：SRAM页x的写保护已禁用 1：SRAM页x的写保护已启用

1.5.15. 定时器输入选择寄存器 (SYSCFG_TIMERCISEL)

地址偏移：0x48

复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		TIMER42_CI1_SEL[1:0]		保留	TIMER42_CI0_SEL[2:0]			保留		TIMER41_CI1_SEL[1:0]		保留	TIMER41_CI0_SEL[2:0]		
		rw			rw					rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TIMER40_CI1_SEL[1:0]		保留	TIMER40_CI0_SEL[2:0]			保留		TIMER14_CI1_SEL[1:0]		保留	TIMER14_CI0_SEL[2:0]		
		rw			rw					rw			rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	TIMER42_CI1_SEL[1:0]	选择TIMER42_CI1输入 00: TIMER42_CH1输入 01: TIMER4_CH1输入 10: 保留 11: 保留
27	保留	必须保持复位值。
26:24	TIMER42_CI0_SEL[2:0]	选择TIMER42_CI0输入 000: TIMER42_CH0输入 001: TIMER4_CH0输入 010: 保留 011: 保留 100: LXTAL 101: 保留 110: MCO 111: 保留
23:22	保留	必须保持复位值。
21:20	TIMER41_CI1_SEL[1:0]	选择 TIMER41_CI1 输入 00: TIMER41_CH1 输入 01: TIMER3_CH1 输入 10: TIMER4_CH1 输入 11: 保留
19	保留	必须保持复位值。
18:16	TIMER41_CI0_SEL[2	选择 TIMER41_CI0 输入

	:0]	000: TIMER41_CH0 输入 001: TIMER3_CH0 输入 010: TIMER4_CH0 输入 011: 保留 100: LXTAL 101: 保留 110: MCO 111: 保留
15:14	保留	必须保持复位值。
13:12	TIMER40_CI1_SEL[1 :0]	选择 TIMER40_CI1 输入 00: TIMER40_CH1 输入 01: TIMER2_CH1 输入 10: TIMER3_CH1 输入 11: TIMER4_CH1 输入
11	保留	必须保持复位值。
10:8	TIMER40_CI0_SEL[2 :0]	选择 TIMER40_CI0 输入 000: TIMER40_CH0 输入 001: TIMER2_CH0 输入 010: TIMER3_CH0 输入 011: TIMER4_CH0 输入 100: LXTAL 101: 保留 110: MCO 111: 保留
7:6	保留	必须保持复位值。
5:4	TIMER14_CI1_SEL[1 :0]	选择 TIMER14_CI1 输入 00: TIMER14_CH1 输入 01: TIMER1_CH1 输入 10: TIMER2_CH1 输入 11: TIMER3_CH1 输入
3	保留	必须保持复位值。
2:0	TIMER14_CI0_SEL[2 :0]	选择 TIMER14_CI0 输入 000: TIMER14_CH0 输入 001: TIMER1_CH0 输入 010: TIMER2_CH0 输入 011: TIMER3_CH0 输入 100: LXTAL 101: 保留 110: MCO

111: 保留

1.5.16. SRAM ECC 配置寄存器 4 (SYSCFG_ECCCFG4)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DWAY0ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWAY0ECCEADDR[13:0]														DWAY0ECCSEIE	DWAY0ECCMEIE
r														rw	rw

位/位域	名称	描述
31:2	DWAY0ECCEADDR[29:0]	记录上一次发生 ECC 事件的 DCACHE way0 SRAM 故障系统地址。
1	DWAY0ECCSEIE	DCACHE way0 单比特可纠错错误中断使能 0: 禁能DCACHE way0 SRAM单比特可纠错错误中断。 1: 使能 DCACHE way0 SRAM 单比特可纠错错误中断。
0	DWAY0ECCMEIE	DCACHE way0多比特不可纠错错误NMI中断使能 0: 禁能DCACHE way0 SRAM多比特不可纠错错误NMI中断。 1: 使能 DCACHE way0 SRAM 多比特不可纠错错误 NMI 中断。

1.5.17. SRAM ECC 配置寄存器 5(SYSCFG_ECCCFG5)

地址偏移：0x50
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DWAY1ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWAY1ECCEADDR[13:0]														DWAY1ECCSEIE	DWAY1ECCMEIE
r														rw	rw

位/位域	名称	描述
31:2	DWAY1ECCEADDR[29:0]	记录上一次发生 ECC 事件的 DCACHE way1 SRAM 故障系统地址。

1	DWAY1ECCSEIE	DCACHE way1 单比特可纠错错误中断使能 0: 禁能DCACHE way1 SRAM单比特可纠错错误中断。 1: 使能 DCACHE way1 SRAM 单比特可纠错错误中断。
0	DWAY1ECCMEIE	DCACHE way1多比特不可纠错错误NMI中断使能 0: 禁能DCACHE way1 SRAM不可纠错错误NMI中断。 1: 使能 DCACHE way1 SRAM 不可纠错错误 NMI 中断。

1.5.18. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=0, 7, 19)

地址偏移：0x100 for TIMER0

地址偏移：0x13C for TIMER7

地址偏移：0x184 for TIMER19
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[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) 其他值：保留
25:21	TSCFG4[4:0]	暂停模式配置

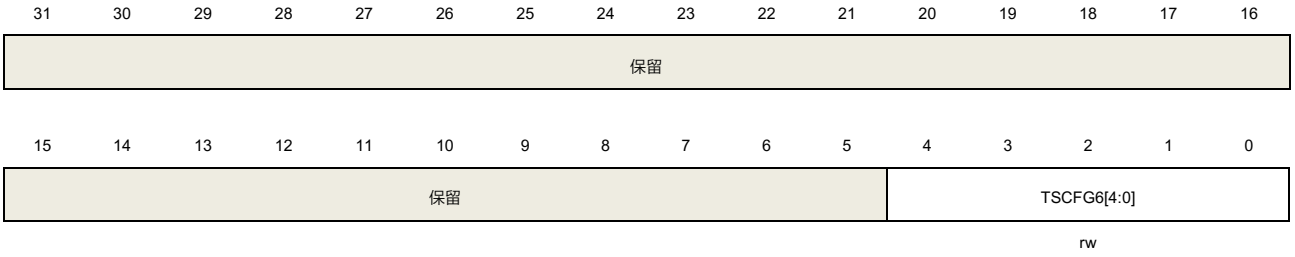
		当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止。 00000：暂停模式禁能 00001：内部触发输入 0(ITI0) 00010：内部触发输入 1(ITI1) 00011：内部触发输入 2(ITI2) 00100：内部触发输入 3(ITI3) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 00111：滤波后的通道 1 输入(CI1FE1) 01000：滤波后的外部触发输入(ETIFP) 01001：滤波后的通道 2 输入(CI2FE2) 01010：滤波后的通道 3 输入(CI3FE3) 其他：保留
20:16	TSCFG3[4:0]	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) 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交编码器模式 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]	正交编码器模式 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]	正交编码器模式 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。

1.5.19. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=0, 7, 19)

地址偏移：0x104 for TIMER0
地址偏移：0x140 for TIMER7
地址偏移：0x188 for TIMER19
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[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
31:5	保留	必须保持复位值。
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) 其他：保留

1.5.20. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=0, 7, 19)

地址偏移：0x108 for TIMER0
地址偏移：0x144 for TIMER7
地址偏移：0x18C for TIMER19
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[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：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) 其他：保留 **注意：**使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0](y=0..6)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。

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

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
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]，TSCFG9[4:0]..TSCFG15[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) 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：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) 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) 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交编码器模式 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]	正交编码器模式 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]	正交编码器模式 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。

1.5.22. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=1, 2, 3, 4)

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
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]，TSCFG9[4:0]..TSCFG15[4:0]之间相互互斥，不能同时配置。

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG11[4:0]	编码器模式 2 配置 00000：禁能 其他值：计数器将在 CI0FE0 和 CI1FE1 信号的上升沿和下降沿计数，计数方向由 CH0P 位和 CH1P 位决定。

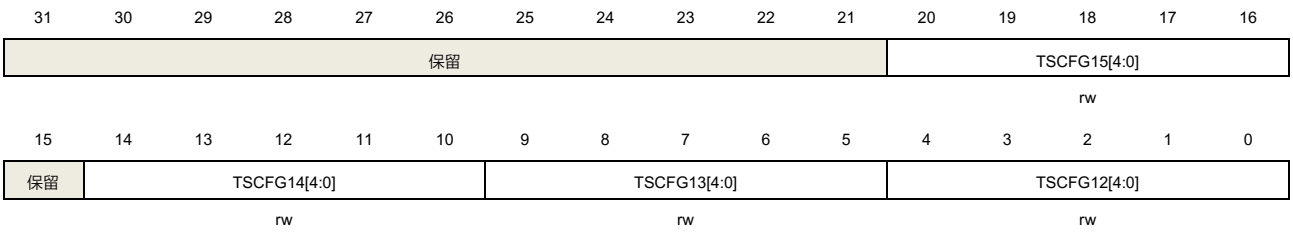
25:21	TSCFG10[4:0]	编码器模式 1 配置 00000：禁能 其他值：CI0FE0 作为计数方向信号，CI1FE1 信号作为计数脉冲。计数器在 CI1FE1 的边沿开始计数，该边沿由 CH1P 设置。
20:16	TSCFG9[4:0]	编码器模式 0 配置 00000：禁能 其他值：CI0FE0作为计数方向信号，CI1FE1信号作为计数脉冲。计数器在CI1FE1的上升沿和下降沿开始计数。
15:5	保留	必须保持复位值。
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 选中的触发输入的上升沿驱动计数器。 00000：外部时钟模式 0 禁能 00001：内部触发输入 0(ITI0) 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) 其他：保留

1.5.23. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=1, 2, 3, 4)

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
复位值：0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]，TSCFG9[4:0]..TSCFG15[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	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：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) 其他：保留 **注意：**使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0](y=0..6, 9..15)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	保留	必须保持复位值。
14:10	TSCFG14[4:0]	正交编码器模式 4 配置 00000：禁能 其他：根据 CI1FE0 的电平，计数器仅在 CI1FE1 的边沿开始计数。
9:5	TSCFG13[4:0]	正交编码器模式 3 配置 00000：禁能 其他值：根据 CI0FE1 的电平，计数器仅在 CI0FE0 的边沿开始计数。
4:0	TSCFG12[4:0]	编码器模式3配置 00000：禁能 其他：计数器计数 CI0FE0 和 CI1FE1 信号的上升沿或下降沿。当 CHxP=0 时，计数器将在高电平或下降沿或 CIxFEx 信号上开始计数；当 CHxP=1 时，计数器将在低电平或上升沿或 CIxFEx 信号上开始计数。

1.5.24. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=9, 10, 12, 13)

地址偏移：0x148 for TIMER9
地址偏移：0x154 for TIMER10
地址偏移：0x160 for TIMER12
地址偏移：0x16C for TIMER13
复位值：0x0000 0000

该寄存器只能按字(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

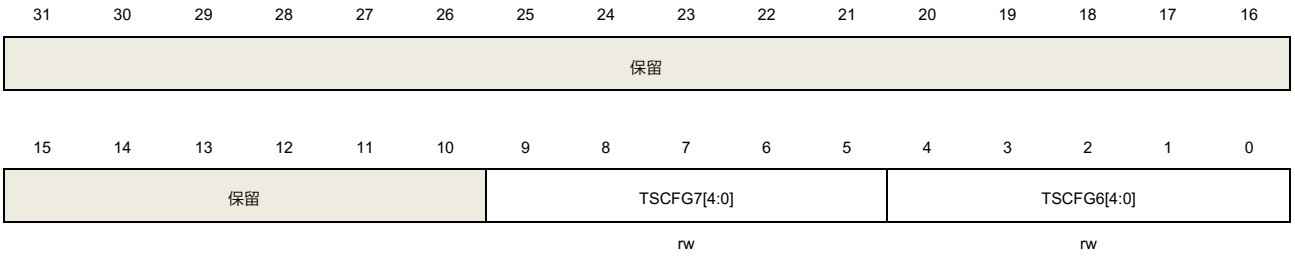
位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000：事件模式禁能 00001：内部触发输入 0(ITI0) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 10011：内部触发输入 14(ITI14) 其他：保留
25:21	TSCFG4[4:0]	暂停模式配置 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止。 00000：暂停模式禁能 00001：内部触发输入 0(ITI0) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 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) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 10011：内部触发输入 14(ITI14) 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交编码器模式 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]	正交编码器模式 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]	正交编码器模式 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。

1.5.25. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=9, 10, 12, 13)

地址偏移：0x14C for TIMER9
地址偏移：0x158 for TIMER10
地址偏移：0x164 for TIMER12
地址偏移：0x170 for TIMER13
复位值：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。
9:5	TSCFG7[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) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 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) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 10011：内部触发输入 14(ITI14) 其他：保留

1.5.26. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=9, 10, 12, 13)

地址偏移：0x150 for TIMER9
地址偏移：0x15C for TIMER10
地址偏移：0x168 for TIMER12
地址偏移：0x174 for TIMER13
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

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) 10011：内部触发输入 14(ITI14) 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG14[4:0]	正交编码器模式 2 配置 00000：禁能 其他：根据 CI0FE1 的电平，计数器仅在 CI1FE1 的边沿开始计数。
9:5	TSCFG13[4:0]	正交编码器模式 1 配置 00000：禁能 其他值：根据 CI0FE0 的电平，计数器仅在 CI1FE0 的边沿开始计数。
4:0	TSCFG12[4:0]	编码器模式2配置 00000：禁能 其他：计数器计数 CI0FE0 和 CI1FE1 信号的上升沿或下降沿。当 CHxP=0 时，计数器将在高电平或下降沿或 CIxFEx 信号上开始计数；当 CHxP=1 时，计数器将在低电平或上升沿或 CIxFEx 信号上开始计数。

1.5.27. TIMERx 配置寄存器 0(SYSCFG_TIMERxCFG0, x=14, 40-44)

地址偏移： 0x178 for TIMER14
地址偏移： 0x190 for TIMER40

地址偏移： 0x19C for TIMER41
地址偏移： 0x1A8 for TIMER42
地址偏移： 0x1B4 for TIMER43
地址偏移： 0x1C0 for TIMER44
复位值：0x0000 0000

该寄存器只能按字(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) 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：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) 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

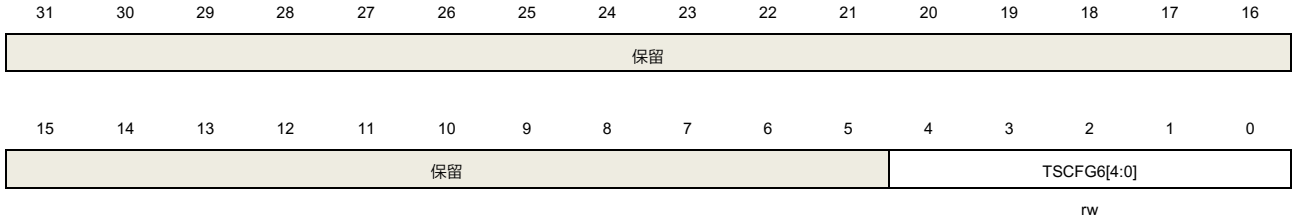
20:16	TSCFG3[4:0]	复位模式配置 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 00000：复位模式禁能 00001：内部触发输入 0(ITI0) 00010：内部触发输入 1(ITI1) 00011：内部触发输入 2(ITI2) 00100：内部触发输入 3(ITI3) 00101：CI0 的边沿标志位(CI0F_ED) 00110：滤波后的通道 0 输入(CI0FE0) 00111：滤波后的通道 1 输入(CI1FE1) 01011：滤波后的多模式通道 0 输入(MCI0FEM0) 10011：内部触发输入 14(ITI14) 其他：保留
15:0	保留	必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

1.5.28. TIMERx 配置寄存器 1(SYSCFG_TIMERxCFG1, x=14, 40-44)

地址偏移： 0x17C for TIMER14
地址偏移： 0x194 for TIMER40
地址偏移： 0x1A0 for TIMER41
地址偏移： 0x1AC for TIMER42
地址偏移： 0x1B8 for TIMER43
地址偏移： 0x1C4 for TIMER44
复位值：0x0000 0000

TSCFG3[4:0]..TSCFG6[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。



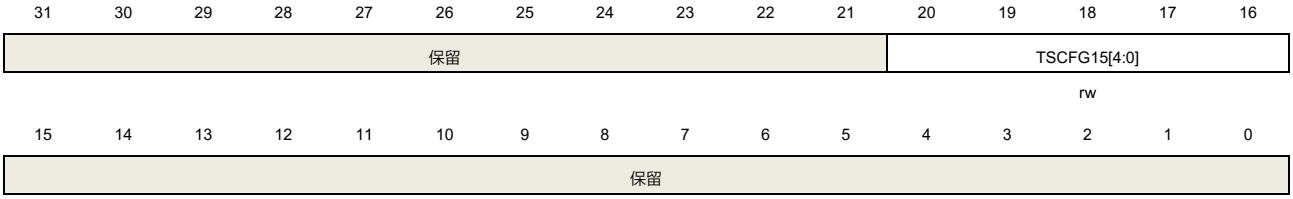
位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 选中的触发输入的上升沿驱动计数器。 00000：外部时钟模式 0 禁能 00001：内部触发输入 0(ITI0) 00010：内部触发输入 1(ITI1) 00011：内部触发输入 2(ITI2)

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)
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

1.5.29. TIMERx 配置寄存器 2(SYSCFG_TIMERxCFG2, x=14, 40-44)

地址偏移： 0x180 for TIMER14
地址偏移： 0x198 for TIMER40
地址偏移： 0x1A4 for TIMER41
地址偏移： 0x1B0 for TIMER42
地址偏移： 0x1BC for TIMER43
地址偏移： 0x1C8 for TIMER44
复位值：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	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：滤波后的通道 0 输入(CI0FE0) 00111：滤波后的通道 1 输入(CI1FE1) 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 **注意：**使用 TSCFG15[4:0]时，需保证 TSCFGy[4:0](y=3..6)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。

1.6. 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。

1.6.1. 存储容量信息

基地址：0x1FFF F7E0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字(32 位)访问

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

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

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

1.6.2. 设备唯一 ID(96 位/位域)

基地址：0x1FFF F7E8

该值是原厂设定的，不能由用户修改。

该寄存器只能按字(32 位)访问

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

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

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

基地址：0x1FFF F7EC

该值是原厂设定的，不能由用户修改。

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[47:32]															
r															

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

基地址：0x1FFF F7F0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															
r															

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

2. 闪存控制器(FMC)

2.1. 概述

闪存控制器(FMC)，提供了片上闪存需要的所有功能。在闪存的 512K 字节空间内，CPU 执行指令需要插入等待状态。FMC 也提供了页擦除、整片擦除以及编程等操作。

2.2. 特性

- 高达512KB的片上闪存可用于存储指令或数据；最多支持18KB引导加载程序；最多支持2KB用户OTP；2KB安全OTP；80B选项字节；
- 在闪存的512KB空间内，CPU执行指令需0~6个等待状态(GD32E533为0~5个)；
- 闪存支持128位宽数据读取
- 预取缓冲区用于提高读操作速度；
- 每页大小为2KB；
- 支持四字编程、页擦除和整片擦除操作；
- 提供安全用户区(SCR)；
- 具备ECC功能，支持单比特可纠错和双比特错误检测ECC功能；
- 2KB OTP(一次性编程)，用于存储用户数据；
- 80B选项字节块用于用户应用需求；
- 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；
- 提供安全 OTP 保护；
- 提供读取地址偏移功能以支持OTA；
- 具有页擦除/写入保护以防止意外操作。

2.3. 功能概述

2.3.1. Flash 存储器架构

闪存存储器包括一个高达 512 KB 的主 Flash，分为 256 个页，每个页面容量为 2KB，另包含一个 18KB 的用于引导装载程序的信息块。主 Flash 存储器包含最多 256 页，每页可单独擦除。闪存存储器的基地址和大小如*表 2-1. 闪存基地址和构成* 所示。

表 2-1. 闪存基地址和构成

闪存块	名称	地址	大小(字节)
主Flash块	第0页	0x0800 0000 - 0x0800 07FF	2KB
	第1页	0x0800 0800 - 0x0800 0FFF	2KB
	第2页	0x0800 1000 - 0x0800 17FF	2KB

闪存块	名称	地址	大小(字节)
	第255页	0x0807 F800 - 0x0807 FFFF	2KB
信息块	引导装载程序	0x1FFF B000 - 0x1FFF F7FF	18KB
一次性可编程块	安全OTP	0x1FFF_6800~0x1FFF_6FFF	2KB
	用户OTP	0x1FFF_7000~0x1FFF_77FF	2KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F84F	80B

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

2.3.2. 错误检查与纠正(ECC)

ECC 机制支持：

- 单比特错误检测和纠错
- 双比特错误检测
- 支持Flash的128位数据和9位ECC码
- 支持在ECC功能后通过硬件EDC检查ECC逻辑
- 1比特错误状态和错误地址寄存器

当检测到单比特错误并纠正时：

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCCOR位将置位。如果设置了FMC_ECCCS寄存器中的ECCCORIE位，则会产生对应中断。ECCADDR表示各个存储空间出现ECC错误的地址。向FMC_ECCCS寄存器ECCCOR位写1可以清除ECCCOR位。

当检测到双比特错误或错误检测码(EDC)时：

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCDET位将置位，数据返回全F。如果设置了SYSCFG_ECCCFG0寄存器中的FLASHECCMEIE位，则会产生中断。ECCADDR位表示各个存储空间出现ECC错误的地址。向FMC_ECCCS寄存器ECCDET位写1可以清除ECCDET位。

注意：

1. 当启用ECC功能时，闪存中的数据是按128位存储的，每4字(128位)后增加了9位纠错码。但增加的9位纠错码由硬件自动计算，用户不可访问。
2. 对于初始数据" 1"(128位)，其在Flash中存储的ECC校验码为0x1FF。
3. 当发生新的ECC错误时，仅记录首次发生错误(单比特或双比特)的地址。
4. 当上报ECC错误时，如果数据仍然存在于当前的缓冲区或预取缓冲区中，即使清除ECCCOR和ECCDET，重新读取该错误地址也不会产生ECC错误。
5. 支持通过设置FMC_ECCCS寄存器中的ECCINJEN和ECCINJDATA[1:0]位进行ECC错误注入

2.3.3. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据可以使用 CPU 的 AHB BUS 总线。

等待状态添加：

根据 AHB 时钟频率正确配置 FMC_WS 寄存器中 WSCNT 位。WSCNT 位域与 AHB 时钟频率之间的关系如*表 2-2. WSCNT 与 AHB 时钟频率的关系*所示。

表 2-2. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E533)

AHB时钟频率	WSCNT配置
<= 38MHz	0(添加0个等待状态)
<= 72MHz	1(添加1个等待状态)
<= 114MHz	2(添加2个等待状态)
<= 152MHz	3(添加3个等待状态)
<= 190MHz	4(添加4个等待状态)
<= 200MHz	5(添加5个等待状态) (适用于GD32E533)

表 2-3. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E535 和 GD32E53P)

AHB时钟频率	WSCNT配置
<= 38MHz	0(添加0个等待状态)
<= 72MHz	1(添加1个等待状态)
<= 114MHz	2(添加2个等待状态)
<= 152MHz	3(添加3个等待状态)
<= 190MHz	4(添加4个等待状态)
<= 220MHz	5(添加5个等待状态) (适用于GD32E535和 GD32E53P)
<= 252MHz	6(添加6个等待状态) (适用于GD32E535和 GD32E53P)

如果发生系统复位，AHB 时钟频率为 8MHz，WSCNT 的值为 0。

表 2-4. WSCNT 与 AHB 时钟频率的关系(适用于 GD32E537)

AHB时钟频率	WSCNT配置
<= 38MHz	0(添加0个等待状态)
<= 72MHz	1(添加1个等待状态)
<= 114MHz	2(添加2个等待状态)
<= 152MHz	3(添加3个等待状态)
<= 190MHz	4(添加4个等待状态)
<= 220MHz	5(添加5个等待状态)

	(适用于GD32E537)
<= 280MHz	6(添加6个等待状态) (适用于GD32E537)

如果发生系统复位，AHB 时钟频率为 8MHz，WSCNT 的值为 0。

注意：

1. 如果需要提高 AHB 的时钟频率，首先，参考***表 2-2. WSCNT 与 AHB 时钟频率的关系***，根据目标 AHB 时钟频率配置 WSCNT 位域。然后将 AHB 时钟频率提高到目标时钟频率。在配置 WSCNT 位域之前提高 AHB 时钟频率是不可行的。

2. 如果需要降低 AHB 的时钟频率，首先，将 AHB 时钟频率降低到目标频率，然后参考***表 2-2. WSCNT 与 AHB 时钟频率的关系***，根据目标 AHB 时钟频率配置 WSCNT 位域。在降低 AHB 时钟频率之前配置 WSCNT 位域是不可行的。

由于添加了等待状态，读效率非常低(如 252MHz/280MHz 时添加六个等待状态)。为了加速读操作，需要用到以下功能。

当前缓冲区

当前缓冲区总是被使能的。每次对闪存存储器进行读操作，将获得一个 128 位的数据并存储在当前缓冲区中。而每一个读操作，CPU 只读取 32 位或 16 位数据。所以在连续代码的情况下，下一个数据能从当前缓冲区中获取到而不需要重复地从存储器获取。

预取缓冲区

当 FMC_WS 寄存器中 PFEN 位被置位时预取缓冲区被使能。预取缓冲区仅在 IBUS(指令总线)上执行。在连续代码的情况下，当 CPU 处理当前缓冲区中数据(128 位)时，32 位操作至少需要 4 个时钟，16 位操作至少需要 8 个时钟。在这种情况下，在闪存存储器的下一个 128 位(四字)地址预取数据并存到预取缓冲区。当 CPU 完成当前缓冲区并需要处理下一个数据，预取缓冲区被命中。

读偏移功能：

将总线初始地址增加一个偏移后再从 FMC 中读取。通过配置 FMC_OFRG 寄存器和 FMC_OFVR 寄存器设置读取的偏移值和应用的区域后，读取源地址的值等同于读取添加偏移后的地址的值。此功能可用于支持 OTA。

读偏移功能生效的条件：

1. 地址在 OF_SPAGE 和 OF_EPAGE 指定的页范围内。

2. OF_SPAGE 必须超出 SCR 区域。

注意：

偏移功能仅支持读操作，不支持编程操作和擦除操作。

OF_SPAGE、OF_EPAGE 和 OF_VALUE 均以页(2KB)为单位。

为保护 SCR 区域，如果 OF_SPAGE 偏移范围与 SCR 重叠，则设置 OFERR 标志且偏移功能失能。

2.3.4. 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 寄存器和 FMC_CTL 中的 OBSTART 位在 FMC_CTL 被解锁后仍然处于被保护状态。解锁过程为两次写操作，向 FMC_OBKEY 寄存器先后写入 0x45670123 和 0xCDEF89AB，然后硬件将 FMC_CTL 寄存器中的 OBLK 位清零。软件可以将 FMC_CTL 的 OBLK 位置 1 来锁定 FMC_OBCTL 寄存器、FMC_CTL 中的 OBSTART 位。

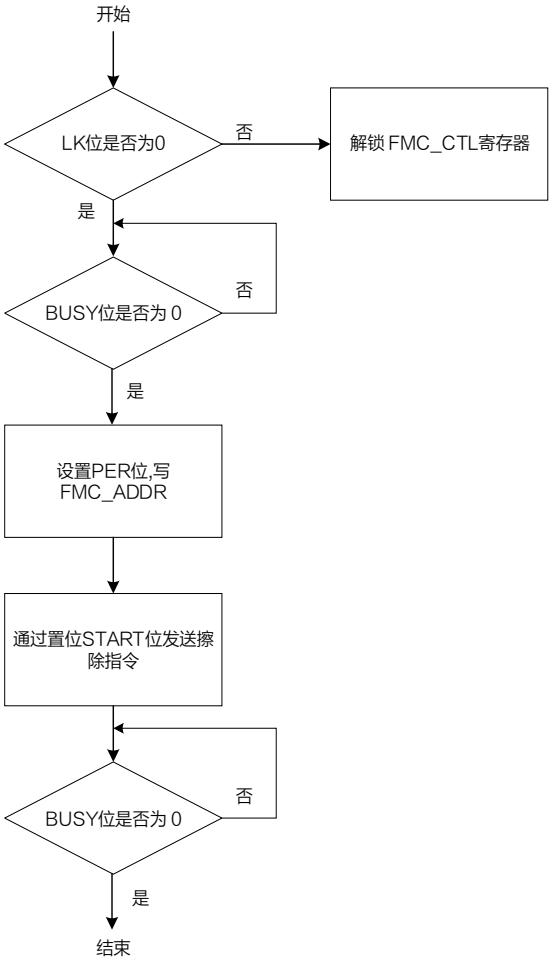
2.3.5. 页擦除

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

- 确保 FMC_CTL 寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 设置FMC_CTL寄存器中的PER位。
- 将页的绝对地址(0x08XX XXXX)写入FMC_ADDR寄存器。
- 通过设置FMC_CTL寄存器中的START位向FMC发送页擦除命令。
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，读取并校验该页是否擦除成功。

当页擦除操作成功执行时，FMC_STAT 寄存器中的 ENDF 将被置位。如果 FMC_CTL 寄存器中的 ENDIE 位被置位，FMC 将触发中断。需要注意的是，用户需确保写入的是正确的擦除页，否则当待擦除页的地址被用来取指令或访问数据时，软件将会"跑飞"。该情况下，FMC 不会有任何出错提示。另一方面，对擦写保护的页进行擦除操作将无效。此时如果 FMC_CTL 寄存器中的 ERRIE 位置位，则 FMC 将触发闪存操作错误中断。软件可以在中断处理程序中检查 FMC_STAT 寄存器中的 WPERR 位来检测这种情况。页擦除操作流程如*图 2-1. 页擦除操作流程* 所示。

图 2-1. 页擦除操作流程



2.3.6. 全片擦除

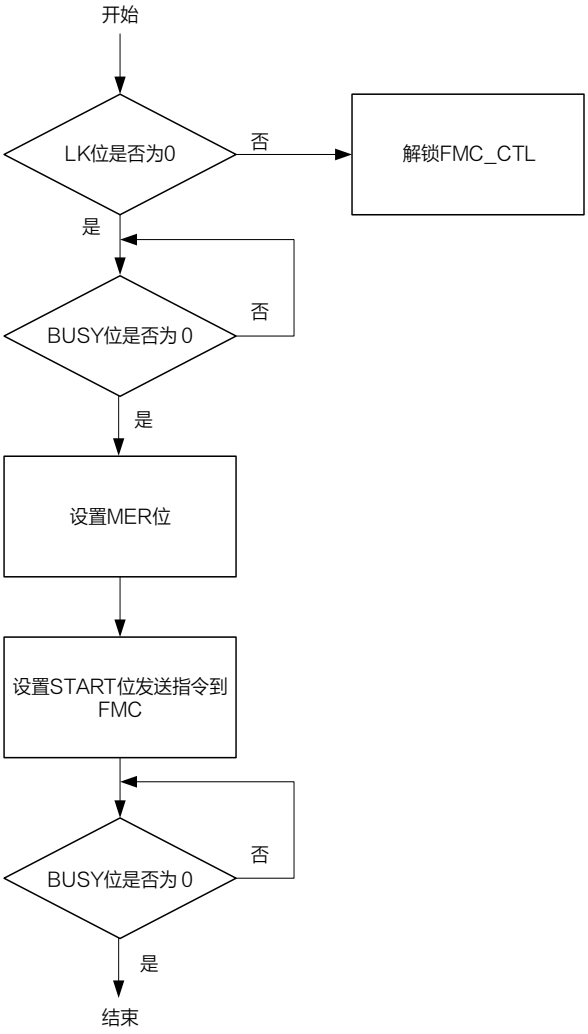
FMC 提供了整片擦除功能可以初始化主存储闪存块的内容。整片擦除操作，寄存器设置具体步骤如下：

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

当整片擦除成功执行，FMC_STAT 寄存器的 ENDF 位置位。若 FMC_CTL 寄存器的 ENDIE 位被置 1，FMC 将触发一个中断。所有的闪存数据都将被复位为 0xFFFF FFFF。整片擦除操作应通过运行在 SRAM 中的程序或使用调试工具直接访问 FMC 寄存器来执行(擦除期间不能从 Flash 取指)。此外，如果任意页被擦除/编程保护，全片擦除操作将被忽略。在此情况下，如果

FMC_CTL 寄存器中的 ERRIE 位被使能，FMC 将触发闪存操作错误中断。软件可以在中断处理程序中检查 FMC_STAT 寄存器中的 WPERR 位来检测此情况。整片擦除操作流程如*图 2-2. 整片擦除操作流程*所示

图 2-2. 整片擦除操作流程



2.3.7. 主闪存编程

FMC 提供 128 位(4 字)编程功能，用于修改主闪存内容。下面的步骤显示了编程操作的寄存器设置顺序。

- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 置位FMC_CTL寄存器的PG位。
- 将要编程的数据写入指定绝对地址(0x08XX XXXX)。通过DBUS写入形成128位数据，然后将128位数据编程到闪存。

要编程的数据必须4字对齐。

- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，读取并校验是否编程成功。

当主存储块编程执行成功时，FMC_STAT 寄存器中的 ENDF 置位，如果 FMC_CTL 寄存器中的 ENDIE 位置位，FMC 将触发中断。请注意，编程操作之前需要检查目的地址是否已经被擦除。如果该地址没有被擦除， FMC_STAT 寄存器的 PGERR 位也将被置 1，除非编程 0x0，。每个 128 位(四字)在一次擦除后只能编程一次，下一次擦除之前无法重复编程。需注意，编程操作前必须设置 PG 位。此外，对擦除/编程保护页面的编程操作会被忽略，并且 FMC_STAT 寄存器中的 WPERR 位会被置位。

在以下情况下，FMC_STAT 寄存器中的 PGAERR 位会被置位。

- 程序使用字节写入(非32位写入)。
- 编程大小与之前编程大小不匹配。不允许混合使用32位和16位写入。
- 写入未对齐。
- 在设置PG位之前发生写入操作。
- PER、MER、PG和OBSTART中的两个及以上同时被置位。

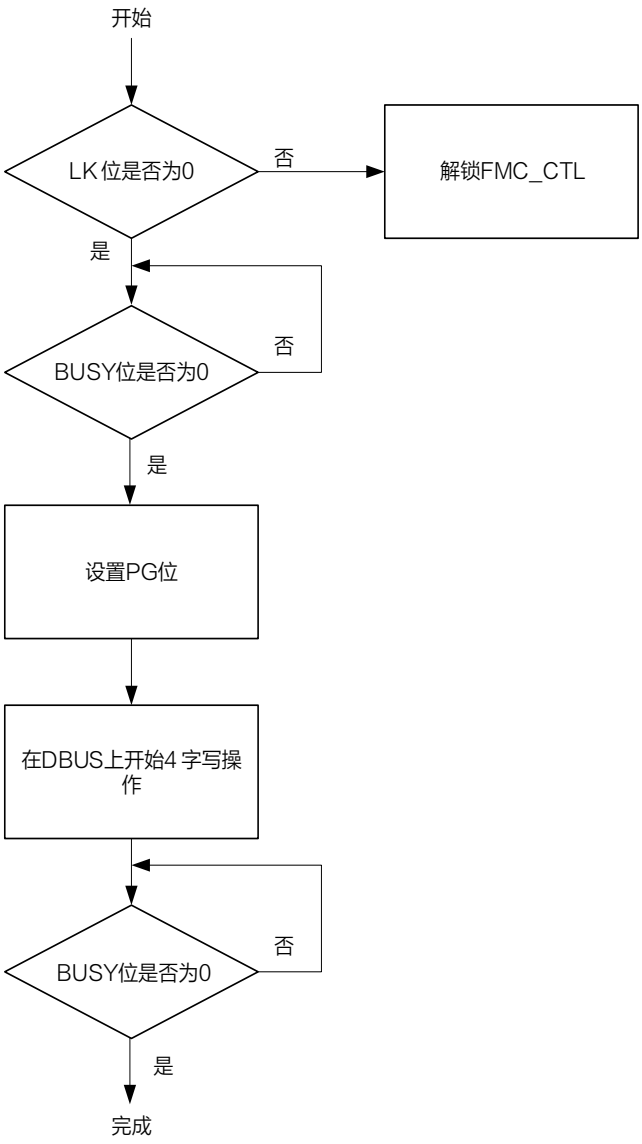
注意：

如果程序未写满总计 128 位，数据将不会被写入 Flash 存储器，且不会有任何提示。

在这些情况下，如果 FMC_CTL 寄存器中的 ERRIE 位被设置，FMC 将触发 Flash 操作错误中断。软件可以通过检查 FMC_STAT 寄存器中的 PGERR 位、PGAERR、PGSERR 位或 WPERR 位来检测中断处理程序中发生的具体情况。

*图 2-3. 四字编程操作流程*显示了主存储块编程操作流程。

图 2-3. 四字编程操作流程



注意：

1. 当程序/擦除操作正在进行时，应避免读取 Flash。同时，如果 CPU 进入省电模式，访问 Flash 存储器将会失败。

如果程序/擦除操作被掉电、复位等中断，Flash 中的内容将无法保证，并处于不确定状态。因此，应采取适当措施以避免程序/擦除操作中断导致的数据丢失。

2.3.8. OTP 编程

OTP 编程方法与主闪存编程相同。OTP 块只能被编程一次并且不能被擦除。

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

意外中断，闪存中的数据有很小可能性会出错。

表 2-5. OTP 字节

地址	名称	属性	描述			出厂默认值
			位/位域	名称	描述	
OTP_SCRCTL						
0x1fff_6800	OTP_SCRCTL	rw	[31:16]	SCR_CTL_LK	OTP 锁定位。一旦锁定，它将变为只读。锁定值为 0xA55A，其他值表示解锁状态。	0xFFFF
			[15:8]	保留		0x7F
			[7:3]	SCR_BLOCK_CNT[4:0]	配置安全用户区块计数。 0x00：无安全用户区 0x01：1 块安全用户区 0x02：2 块安全用户区 … 0x10：16 块安全用户区 0x11-0x1E：16 块安全用户区 0x1F：无安全用户区域	0x1F
			[2:1]	保留	-	0x3
			[0]	BOOTLK	启动地址锁定 1：根据引脚和选项字节选择启动地址 0：锁定启动地址为主闪存	0x1
0x1fff_6804	保留	-	-	-	-	-
0x1fff_6808	保留	-	-	-	-	-
0x1fff_680C	保留	-	-	-	-	-
0x1fff_6810	DBGPSWD_DATA0[31:0]	w	[31:0]		安全 JTAG 密钥 0	0xFFFFFFFF
0x1fff_6814	DBGPSWD_DATA1[31:0]	w	[31:0]		安全 JTAG 密钥 1	0xFFFFFFFF
0x1fff_6818	DBGPSWD_DATA2[31:0]	w	[31:0]		安全 JTAG 密钥 2	0xFFFFFFFF
0x1fff_681C	DBGPSWD_DATA3[31:0]	w	[31:0]		安全 JTAG 密钥 3	0xFFFFFFFF
0x1fff_6820	OTP_DBG_CTL	rw	[31:16]	DBG_CTL_LK	锁定与 DBG 相关的字段(DBGPSWD_DATA0-3 和 OTP_DBG_CTL)。一旦锁定，它们将无法被写。锁定值为 0x3CC3，其他值表示未锁定。	0xFFFF
			[15:3]	保留		

			[2:1]	NDBG	配置调试行为： 0x：无调试 10：安全调试 11：正常调试	11
			[0]	JTAGNSW	切换 SW/JTAG 调试 1：使能 SW 调试 0：使能 JTAG 调试	0x1
0x1fff_6824	保留	-	-	-	-	-
0x1fff_6828	保留	-	-	-	-	-
0x1fff_682C	保留	-	-	-	-	-
AES 密钥区域						
0x1fff_6830	OTP_KEY_0	w	[31:0]	OTP_KEY_0_0L	AES 密钥 0	0xFFFFFFFF
0x1fff_6834			[31:0]	OTP_KEY_0_0H		0xFFFFFFFF
0x1fff_6838			[31:0]	OTP_KEY_0_1L		0xFFFFFFFF
0x1fff_683C			[31:0]	OTP_KEY_0_1H		0xFFFFFFFF
0x1fff_6840			[31:0]	OTP_KEY_0_2L		0xFFFFFFFF
0x1fff_6844			[31:0]	OTP_KEY_0_2H		0xFFFFFFFF
0x1fff_6848			[31:0]	OTP_KEY_0_3L		0xFFFFFFFF
0x1fff_684C			[31:0]	OTP_KEY_0_3H		0xFFFFFFFF
0x1fff_6850	OTP_KEY_1	w	[31:0]	OTP_KEY_1_0L	AES 密钥 1	0xFFFFFFFF
0x1fff_6854			[31:0]	OTP_KEY_1_0H		0xFFFFFFFF
0x1fff_6858			[31:0]	OTP_KEY_1_1L		0xFFFFFFFF
0x1fff_685C			[31:0]	OTP_KEY_1_1H		0xFFFFFFFF
0x1fff_6860			[31:0]	OTP_KEY_1_2L		0xFFFFFFFF
0x1fff_6864			[31:0]	OTP_KEY_1_2H		0xFFFFFFFF
0x1fff_6868			[31:0]	OTP_KEY_1_3L		0xFFFFFFFF

0x1fff_686C			[31:0]	OTP_KEY_1_3H		0xFFFF FFFF
0x1fff_6870	OTP_KEY_2	w	[31:0]	OTP_KEY_2_0L	AES 密钥 2	0xFFFF FFFF
0x1fff_6874			[31:0]	OTP_KEY_2_0H		0xFFFF FFFF
0x1fff_6878			[31:0]	OTP_KEY_2_1L		0xFFFF FFFF
0x1fff_687C			[31:0]	OTP_KEY_2_1H		0xFFFF FFFF
0x1fff_6880			[31:0]	OTP_KEY_2_2L		0xFFFF FFFF
0x1fff_6884			[31:0]	OTP_KEY_2_2H		0xFFFF FFFF
0x1fff_6888			[31:0]	OTP_KEY_2_3L		0xFFFF FFFF
0x1fff_688C			[31:0]	OTP_KEY_2_3H		0xFFFF FFFF
0x1fff_6890	OTP_KEY_3	w	[31:0]	OTP_KEY_3_0L	AES 密钥 3	0xFFFF FFFF
0x1fff_6894			[31:0]	OTP_KEY_3_0H		0xFFFF FFFF
0x1fff_6898			[31:0]	OTP_KEY_3_1L		0xFFFF FFFF
0x1fff_689C			[31:0]	OTP_KEY_3_1H		0xFFFF FFFF
0x1fff_68A0			[31:0]	OTP_KEY_3_2L		0xFFFF FFFF
0x1fff_68A4			[31:0]	OTP_KEY_3_2H		0xFFFF FFFF
0x1fff_68A8			[31:0]	OTP_KEY_3_3L		0xFFFF FFFF
0x1fff_68AC			[31:0]	OTP_KEY_3_3H		0xFFFF FFFF
OTP 用户数据区域						
0x1fff_68B0-0x1fff_6FFF	USER_DATA0	rw				全 F
0x1fff_7000-0x1fff_73FF	USER_DATA1	rw			总共有 64 个块，每个块包含 16 字节。写保护由 FMC_OTPDATA1_LLK 和 FMC_OTPDATA1_HLK 寄存器控制，每个位对应一个块。"0"表示该块已写保护(应用：如用户状态、生命周期等)。	全 F

0x1fff_7400-0x1fff_77FF	USER_DATA2	rw			总共有 64 个块，每个块包含 16 字节。写锁由 FMC_OTPDATA2_LLK 和 FMC_OTPDATA2_HLK 寄存器控制，每个位控制一个块。"0"表示该块已写锁。读锁由 FMC_OTPDATA2_LR_CTL 和 FMC_OTPDATA2_HR_CTL 寄存器控制，每个位控制一个块。"0"表示该块已读锁。(应用：例如存储公钥哈希值)	全 F

强制从 Flash 存储器启动

为了提升安全性并建立信任链，OTP_SCRCTL 寄存器中的 BOOTLK 位允许强制系统从主 Flash 存储器启动，而忽略其他启动选项。

2.3.9. 选项字节

选项字节描述

选项字节块在每次系统复位后重新加载到 FMC_OBCTL 寄存器中，然后选项字节生效。补充选项字节与选项字节相反。当选项字节被重新加载时，如果补充选项字节与选项字节不匹配，FMC_STAT 寄存器中的 OBERR 位将被置位，并且选项字节将设置为 0xFF。如果选项字节及其补充字节均为 0xFF，则 OBERR 位不会被置位。以下表格展示了选项字节的详细信息。

表 2-6. 选项字节

地址	名称	说明
0x1fff f800	SPC	选项字节安全保护值 0xA5：未保护状态 除0xA5和0xCC外的任何值：保护级别低 0xCC：保护级别高
0x1fff f801	SPC_N	SPC补码字节
0x1fff f802	USER[7:0]	[7:5]：BOR_TH(欠压复位阈值选择) 000：3.0V(上升), 2.9V(下降) 001：2.9V(上升), 2.8V(下降) 010：2.7V(上升), 2.6V(下降) 011：2.6V(上升), 2.5V(下降) 100：2.5V(上升), 2.4V(下降) 101：2.3V(上升), 2.2V(下降) 110：2.1V(上升), 2.0V(下降) 111：1.85V(上升), 1.75V(下降) [4]：BOR_EN(欠压复位使能) 0：使能BOR 1：失能BOR

地址	名称	说明
		[3]：SRAM_ECCEN(SRAM ECC使能位) 0：禁能SRAM ECC 1：使能SRAM ECC [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式 1：设置待机模式时进入待机模式而不产生复位 [1]：nRST_DPSLP 0：设置深度睡眠模式时产生复位而不进入深度睡眠模式 1：设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]：nFWDG_HW 0：硬件使能独立看门狗功能 1：软件使能独立看门狗功能
0x1fff f803	USER_N[7:0]	USER补码字节
0x1fff f804	USER[15:8]	[7:5]：BOREF_TH(Flash电压欠压阈值选择) 000：1.61V(上升), 1.57V(下降) 001：1.66V(上升), 1.62V(下降) 010：1.71V(上升), 1.67V(下降) 011：1.76V(上升), 1.72V(下降) 100：1.81V(上升), 1.77V(下降) 101~111：1.86V(上升), 1.82V(下降) [4]：FWDGTSPD_DPSLP 0：FWDGT深度睡眠模式冻结选项状态位 1：FWDGT在深度睡眠模式下保持运行 [3]：FWDGTSPD_STANDBY 0：FWDGT在待机模式下不运行 1：FWDGT在待机模式下保持运行 [2]：nBOOT0 0：BOOT0 为 1 1：BOOT0 为 0 [1]：nBOOT1 0：BOOT1 为 1 1：BOOT1 为 0 它与 BOOT0 引脚共同决定 boot 模式 [0]：nSWBT0 (默认值为0) 1：BOOT0取决于PB11/BOOT0引脚 0：BOOT0取决于nBOOT0选项位
0x1fff f805	USER_N[15:8]	USER补码字节
0x1fff f806	USER[23:16]	[2]：DCACHE_ECCEN (默认值为1) 0：失能DCACHE ECC 1：使能DCACHE ECC [1]：ICACHE_ECCEN

地址	名称	说明
		(默认值为1) 0：失能ICACHE ECC 1：使能ICACHE ECC [0]：SRAM_RST 0：电源复位后初始化SRAM 1：电源复位后SRAM不初始化
0x1fff f807	USER_N[23:16]	USER补码字节
0x1fff f808	OB_SCR_WLK[7:0]	SCRWLK[x]表示SCR数据块x已被写保护锁定
0x1fff f809	OB_SCR_WLK_N[7:0]	OB_SCR_WLK[7:0]补码字节
0x1fff f80a	OB_SCR_WLK[15:8]	SCRWLK[x]表示SCR数据块x已被写保护锁定
0x1fff f80b	OB_SCR_WLK_N[15:8]	OB_SCR_WLK[15:8]补码字节
0x1fff f80c- 0x1fff f80f	保留	
0x1fff f810	WP0_SADDR[7:0]	WP0_SADDR包含写保护区域0的第一页地址
0x1fff f811	WP0_SADDR_N[7:0]	WP0_SADDR补码字节
0x1fff f812	保留	
0x1fff f813	保留	
0x1fff f814	WP0_EADDR[7:0]	WP0_EADDR包含写保护区域0的最后一页地址
0x1fff f815	WP0_EADDR_N[7:0]	WP0_EADDR补码字节
0x1fff f816	保留	
0x1fff f817	保留	
0x1fff f818	WP1_SADDR[7:0]	WP1_SADDR包含写保护区域1的第一页地址
0x1fff f819	WP1_SADDR_N[7:0]	WP1_SADDR补码字节
0x1fff f81A	保留	
0x1fff f81B	保留	
0x1fff f81C	WP1_EADDR[7:0]	WP1_EADDR包含写保护区域1的最后一页地址
0x1fff f81D	WP1_EADDR_N[7:0]	WP1_EADDR补码字节
0x1fff f81E	保留	
0x1fff f81F	保留	
0x1fff f820	WP2_SADDR[7:0]	WP2_SADDR包含写保护区域2的第一页地址
0x1fff f821	WP2_SADDR_N[7:0]	WP2_SADDR补码字节
0x1fff f822	保留	
0x1fff f823	保留	
0x1fff f824	WP2_EADDR[7:0]	WP2_EADDR包含写保护区域2的最后一页地址
0x1fff f825	WP2_EADDR_N[7:0]	WP2_EADDR补码字节
0x1fff f826	保留	
0x1fff f827	保留	
0x1fff f828	WP3_SADDR[7:0]	WP3_SADDR包含写保护区域3的第一页地址
0x1fff f829	WP3_SADDR_N[7:0]	WP3_SADDR补码字节
0x1fff f82A	保留	
0x1fff f82B	保留	

地址	名称	说明
0x1fff f82C	WP3_EADDR [7:0]	WP3_EADDR包含写保护区域3的最后一页地址
0x1fff f82D	WP3_EADDR_N[7:0]	WP3_EADDR补码字节
0x1fff f82E	保留	
0x1fff f82F	保留	
0x1fff f830	FMC_OTPDATA1_LK[7:0]	OTP USER_DATA1 block(7-0)锁定
0x1fff f831	FMC_OTPDATA1_LK_N[7:0]	FMC_OTPDATA1_LK[7:0]补码字节
0x1fff f832	FMC_OTPDATA1_LK[15:8]	OTP USER_DATA1 block(15-8)锁定
0x1fff f833	FMC_OTPDATA1_LK_N[15:8]	FMC_OTPDATA1_LK[15:8]补码字节
0x1fff f834	FMC_OTPDATA1_LK[23:16]	OTP USER_DATA1 block(23-16)锁定
0x1fff f835	FMC_OTPDATA1_LK_N[23:16]	FMC_OTPDATA1_LK[23:16]补码字节
0x1fff f836	FMC_OTPDATA1_LK[31:24]	OTP USER_DATA1 block(31-24)锁定
0x1fff f837	FMC_OTPDATA1_LK_N[31:24]	FMC_OTPDATA1_LK[31:24]补码字节
0x1fff f838	FMC_OTPDATA1_LK[39:32]	OTP USER_DATA1 block(39-32)锁定
0x1fff f839	FMC_OTPDATA1_LK_N[39:32]	FMC_OTPDATA1_LK[39:32]补码字节
0x1fff f83a	FMC_OTPDATA1_LK[47:40]	OTP USER_DATA1 block(47-40)锁定
0x1fff f83b	FMC_OTPDATA1_LK_N[47:40]	FMC_OTPDATA1_LK[47:40]补码字节
0x1fff f83c	FMC_OTPDATA1_LK[55:48]	OTP USER_DATA1 block(55-48)锁定
0x1fff f83d	FMC_OTPDATA1_LK_N[55:48]	FMC_OTPDATA1_LK[55:48]补码字节
0x1fff f83e	FMC_OTPDATA1_LK[63:56]	OTP USER_DATA1 block(63-56)锁定
0x1fff f83f	FMC_OTPDATA1_LK_N[63:56]	FMC_OTPDATA1_LK[63:56]补码字节
0x1fff f840	FMC_OTPDATA2_LK[7:0]	OTP USER_DATA2 block(7-0)锁定
0x1fff f841	FMC_OTPDATA2_LK_N[7:0]	FMC_OTPDATA2_LK[7:0]补码字节
0x1fff f842	FMC_OTPDATA2_LK[1	OTP USER_DATA2 block(15-8)锁定

地址	名称	说明
	5:8]	
0x1fff f843	FMC_OTPDATA2_LK_N[15:8]	FMC_OTPDATA2_LK[15:8]补码字节
0x1fff f844	FMC_OTPDATA2_LK[23:16]	OTP USER_DATA2 block(23-16)锁定
0x1fff f845	FMC_OTPDATA2_LK_N[23:16]	FMC_OTPDATA2_LK[23:16]补码字节
0x1fff f846	FMC_OTPDATA2_LK[31:24]	OTP USER_DATA2 block(31-24)锁定
0x1fff f847	FMC_OTPDATA2_LK_N[31:24]	FMC_OTPDATA2_LK[31:24]补码字节
0x1fff f848	FMC_OTPDATA2_LK[39:32]	OTP USER_DATA2 block(39-32)锁定
0x1fff f849	FMC_OTPDATA2_LK_N[39:32]	FMC_OTPDATA2_LK[39:32]补码字节
0x1fff f84a	FMC_OTPDATA2_LK[47:40]	OTP USER_DATA2 block(47-40)锁定
0x1fff f84b	FMC_OTPDATA2_LK_N[47:40]	FMC_OTPDATA2_LK[47:40]补码字节
0x1fff f84c	FMC_OTPDATA2_LK[55:48]	OTP USER_DATA2 block(55-48)锁定
0x1fff f84d	FMC_OTPDATA2_LK_N[55:48]	FMC_OTPDATA2_LK[55:48]补码字节
0x1fff f84e	FMC_OTPDATA2_LK[63:56]	OTP USER_DATA2 block(63-56)锁定
0x1fff f84f	FMC_OTPDATA2_LK_N[63:56]	FMC_OTPDATA2_LK[63:56]补码字节

选项字节修改

以下步骤显示了修改顺序。

- 确保FMC_OBCTL寄存器不处于锁定状态；
- 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；
- 通过向FMC_OBKEY寄存器写入正确的序列，解锁FMC_CTL寄存器中的OBLK位。
- 等待FMC_CTL寄存器中的OBLK位被清除。
- 将所需的选项字节写入目标选项字节寄存器。
- 设置FMC_CTL寄存器中的OBSTART位以发送选项字节更改命令。
- 通过检查FMC_STAT寄存器中BUSY位的值，等待所有操作完成。

注意：

在修改过程中，禁止断电或复位，否则可能导致选项字节异常，导致程序无法运行。代码和选项字节需要重新编程。

2.3.10. 页擦除/编程保护(写保护)

FMC 的页擦除/编程保护功能可以阻止对闪存的意外操作。当 FMC 对被保护页进行页擦除或编程操作时，操作本身无效且 FMC_STAT 寄存器的 WPERR 位将被置 1。若任一区域启用写保护功能则软件将无法执行整片擦除操作。WP 区域由起始地址偏移量和结束地址偏移量定义。通过配置 WP 地址寄存器 FMC_WPx (x = 0,1,2,3)，可以单独使能页面保护功能。

可以定义四个 WP 区域，粒度为 2 Kbytes。WP 区域由以下方式定义：

[(WPx _EADDR - WPx _SADDR) + 1] x 2Kbytes

从基地址 + [(WPx_SADDR[7:0]) x 0x800](包含)到基地址 + [(WPx_EADDR[7:0] + 1) x 0x800](不包含)。

例如，要保护地址从 0x0800 2000(包含)到 0x0800 4FFF(包含)，FMC_WPx 寄存器必须设置如下：

WPx_SADDR[7:0] = 0x04

WPx_EADDR[7:0] = 0x09

表 2-7. OB_WP 位用于受保护页面

FMC_WPx 寄存器值 (x=0,1,2,3)	WP 区域
WPx_SADDR = WPx_EADDR	页 WPx_SADDR 被保护
WPx_SADDR > WPx_EADDR	无 WP 区域
WPx_SADDR < WPx_EADDR	从 WPx_SADDR 到 WPx_EADDR 的页被保护

2.3.11. 安全保护

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

无保护状态：当将 SPC 字节及其补字节被设置为 0xA55A，系统复位以后，闪存将处于非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

低等级保护：当设置 SPC 字节及其补字节值为任何除 0xA55A 和 0xCC33 外的值，系统复位以后，低安全保护状态生效。注意，如果在调试器保持连接到 JTAG/SWD 设备时执行了 SPC 修改，则应该进行电源重置，而不是系统重置。在低等级保护下，主 flash 只能通过用户代码访问。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读操作，将会产生一个总线错误。在调试模式或者从 SRAM 中启动时或者从 bootloader 模式启动，如果对主存储块执行编程/擦除操作，FMC_STAT 寄存器中的 WPERR 位会被置位。

通过 DMA 和 CPU 写入 Flash 编程不会成功也不会触发 WPERR 置位，会触发总线错误，DMA

不会处理该错误，而 CPU 侧软件可检测到 HardFault。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。通过将 SPC 字节及其补码值设置为 0xA55A，返回到无保护级别，然后自动触发一次整片擦除操作。

高等级保护：将 SPC 字节及其补字节设置为 0xCC33 时，激活高等级安全保护。当编程选择该保护等级时，调试模式，从 SRAM 中启动，或者从 boot loader 启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC 字节及其补字节禁止再次编程。所以，如果高等级保护被激活，将不能再降回到低等级保护或未保护状态。

2.3.12. 安全用户区域(SCR)

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

安全用户区域可以将安全代码与应用程序非安全代码隔离开来。安全用户区域可用于保护自定义安全引导库、固件更新代码或第三方安全库。当安全用户区被使能时，对 SCR 区域进行擦除操作会设置 WPERR 位。当与数据块对应的 SCRREN 位为 0 时，该数据块受到读取和执行保护。读取该区域将返回 0 并触发总线错误。当与数据块对应的 SCRWLK 位为 0 时，对该数据块执行写操作将设置 WPERR 标志。

安全用户区的大小由 OTP_SCRCTL 的 SCR_BLOCK_CNT[4:0]位定义。只有在 SPC 无保护级别时才能修改其大小。在 SPC 低级别保护切换到无保护时，除被配置了安全保护区外的其他内容会被擦除。

安全用户区定义如下：

- 从基地址(0x0800 0000)到基地址 + (SCR_BLOCK_CNT[4:0] * 0x2000)(不包含此地址)。SCR_BLOCK_CNT[4:0]为 0x10~0x1E 时按 16 块计算，每块大小为 8KB(0x2000)；为 0x00 和 0x1F 时无安全用户区。

表 2-8. SCR/WP 擦除保护

擦除类型	SCR_BLOCK_CNT[4:0]	写保护	行为	WPERR
页擦除	0x00 或 0x1F(无安全用户区)	无	页可正常擦除	不置位
	0x00 或 0x1F(无安全用户区)	有	写保护页不可擦除	置位
	0x01-0x1E	-	页被保护无法擦除	置位
全片擦除	0x00 或 0x1F(无安全用户区)	无	全片正常擦除	不置位
	0x00 或 0x1F(无安全用户区)	有	不可全片擦除	置位
	0x01-0x1E	-	不可全片擦除	置位
低等级保护或高等级保护	0x00 或 0x1F(无安全用户区)	无	全片正常擦除	不置位
	0x00 或 0x1F(无安全用户区)	有	不可正常擦除	不置位
	0x01-0x1E	-	除安全保护区外页正常擦除	不置位

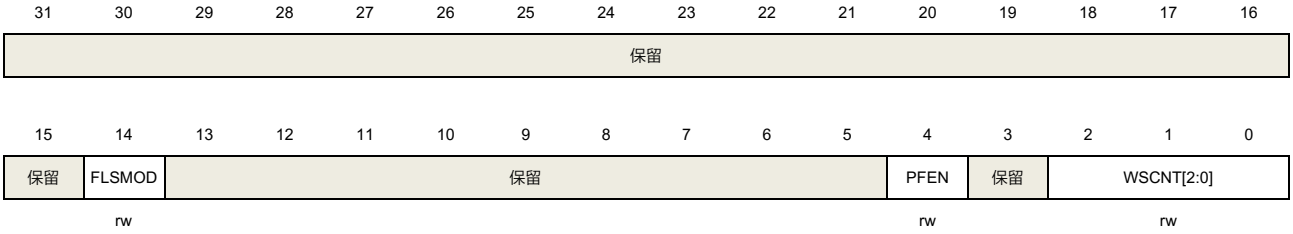
2.4. 寄存器定义

FMC 基地址：0x4002 2000

2.4.1. 等待状态寄存器(FMC_WS)

地址偏移：0x00
复位值：0x0000 0010

该寄存器必须以字(32 位)访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14	FLSMOD	当 MCU 进入深度睡眠模式时，Flash 进入睡眠模式或掉电模式。 0：Flash 进入睡眠模式。 1：Flash进入掉电模式。
13:5	保留	必须保持复位值。
4	PFEN	预取使能 0：预取失能 1：预取使能
3	保留	必须保持复位值。
2:0	WSCNT[2:0]	等待状态计数寄存器 软件置 1 和清 0。 000：不增加等待状态 001：增加 1 个等待状态 010：增加 2 个等待状态 011：增加 3 个等待状态 100：增加 4 个等待状态 101：增加 5 个等待状态 110：增加 6 个等待状态 111：增加 7 个等待状态

2.4.2. 解锁寄存器(FMC_KEY)

地址偏移：0x04
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

2.4.3. 选项字节解锁寄存器(FMC_OBKEY)

地址偏移：0x08

复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

2.4.4. 状态寄存器(FMC_STAT)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31	FLSPWR	Flash电源就绪 0：Flash 电源未就绪 1：Flash电源就绪
30:9	保留	必须保持复位值。
8	OFERR	偏移错误标志位 当 SCR 区域中的 OF_SPAGE 发生错误时，硬件设置该位。通过写入 1 可清除此位。 0：未发生偏移错误 1：发生偏移错误
7	保留	必须保持复位值。
6	OBERR	选项字节读取错误标志位 当选项字节与其补充字节不匹配时，该位由硬件置位，并且该选项字节强制设置为 0xFF。
5	ENDF	操作结束标志位 当闪存编程或擦除操作已完成并成功时，此位将被硬件置 1。 该位可以由软件写 1 清 0。
4	WPERR	擦除/编程保护错误标志位 当擦除/编程保护错误发生时，该位由硬件置位，软件写 1 清 0。 0：未发生擦除/编程保护错误。 1：发生擦除/编程保护错误。
3	PGAERR	编程对齐错误标志位 当写入的数据非128位对齐时，即要编程的数据不能包含在同一128位闪存行中，该位将由硬件置位。 该位可以由软件写 1 清 0。
2	PGERR	编程错误标志位 向未擦除的地址编程时，该位由硬件置 1。软件可通过写入 1 将其清除。
1	PGSERR	编程顺序错误标志位 当编程操作没有将PG位预先置1就编程时，该位将由硬件置位。由于先前的编程错误导致PGERR，PGAERR或WPERR置1时，该位也置1。 该位可以由软件写 1 清 0。 0：未发生编程顺序错误。

1：发生编程顺序错误。

0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置 1。当操作结束或者出错，此位被清 0。

2.4.5. 控制寄存器(FMC_CTL)

地址偏移：0x10
复位值：0xC000 0000

该寄存器必须以字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFERRIE	保留	OBERRIE	ENDIE	保留	ERRIE	保留							MER	PER	PG
rw		rw	rw		rw								rw	rw	rw

位/位域	名称	描述
31	LK	FMC_CTL 寄存器锁定标志位 当正确的序列写入到 FMC_KEY 寄存器时，该位将由硬件清 0。该位可以由软件置 1。
30	OBLK	FMC_OBCTL 寄存器锁定标志位 如果该位被置1，FMC_OBCTL寄存器中关于用户选项字节的所有位以及选项字节页都将被锁定。当正确的序列被写入到FMC_OBKEY寄存器时，该位被硬件清0。该位可由软件置1。
29:18	保留	必须保持复位值。
17	OBSTART	发送选项字节更改命令位 该位由软件置1，只有当OBLK被设置为0时，才向FMC发送选项字节更改命令。当BUSY位被清除时，该位被硬件清除。
16	START	向 FMC 发送擦除命令位 该位用于向 FMC 发送擦除命令。当 BUSY 位被清除时，该位被硬件清除。
15	OFERRIE	偏移误差中断使能 0：硬件不产生中断 1：偏移误差中断使能
14	保留	必须保持复位值。
13	OBERRIE	选项字节错误中断使能 0：硬件不产生中断

		1：选项字节错误中断使能
12	ENDIE	操作结束中断使能位 该位可以由软件置 1 或清 0。 0：失能操作结束中断。 1：使能操作结束中断。
11	保留	必须保持复位值。
10	ERRIE	操作错误中断使能位 该位可以由软件置 1 或清 0。 0：失能操作失败错误中断。 1：使能操作失败错误中断。
9:3	保留	必须保持复位值。
2	MER	Flash全片擦除指令位 该位由软件设置或清除。 0：无影响 1：Flash 全片擦除指令
1	PER	主闪存页擦除指令位 该位可以由软件置 1 或清 0 0：无作用。 1：主闪存页擦除指令。
0	PG	主闪存编程命令位 该位可以由软件置 1 或清 0 0：失能主存储块编程命令。 1：使能主存储块编程命令。

2.4.6. 地址寄存器 (FMC_ADDR)

地址偏移：0x14
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

位/位域	名称	描述
31:0	ADDR[31:0]	闪存命令地址位

软件置位

ADDR 位是闪存擦除/编程命令的地址。

2.4.7. ECC 控制寄存器 (FMC_ECCCTL)

地址偏移：0x18

复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												DCACHE_ECCEN	ICACHE_ECCEN	SRAM_ECCEN	SRAM_ECCSEL
												rw	rw	rw	rw

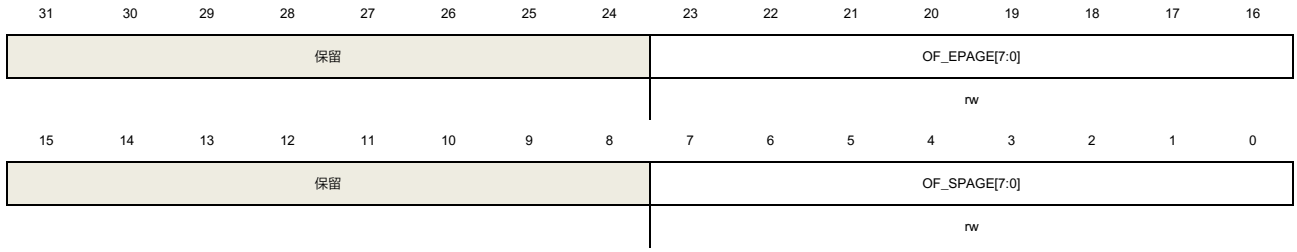
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	DCACHE_ECCEN	DCACHE ECC 使能 0：DCACHE ECC 失能 1：DCACHE ECC 使能 当SRAM_ECCSEL=1时，DCACHE_ECCEN控制DCACHE ECC的使能。
2	ICACHE_ECCEN	ICACHE ECC 使能 0：ICACHE ECC 失能 1：ICACHE ECC 使能 当SRAM_ECCSEL=1时，ICACHE_ECCEN控制ICACHE ECC的使能。
1	SRAM_ECCEN	SRAM ECC 使能 0：SRAM ECC 失能 1：SRAM ECC 使能 当 SRAM_ECCSEL=1 时，SRAM_ECCEN 控制 SRAM ECC 的使能。
0	SRAM_ECCSEL	SRAM ECC 使能控制选择 0：SRAM ECC 功能通过选项字节控制 1：SRAM ECC 功能通过寄存器位使能

2.4.8. 偏移区域寄存器 (FMC_OFRG)

地址偏移：0x1C

复位值：0x0000 00FF

该寄存器必须以字(32 位)访问。

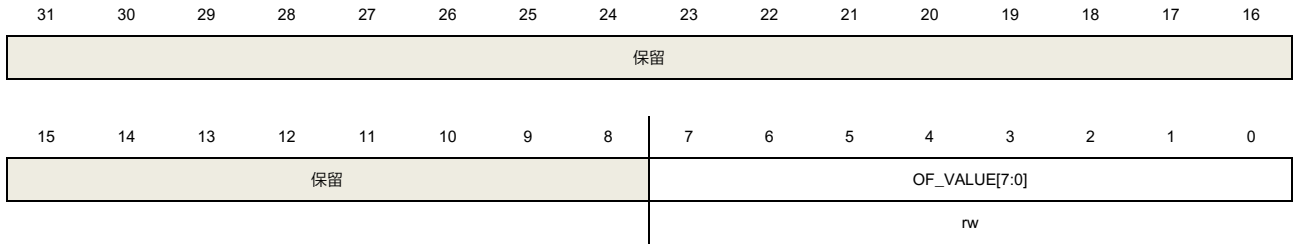


位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	OF_EPAGE[7:0]	偏移区域结束页
15:8	保留	必须保持复位值。
7:0	OF_SPAGE[7:0]	偏移区域起始页

2.4.9. 偏移值寄存器 (FMC_OFVR)

地址偏移：0x20
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	OF_VALUE[7:0]	页偏移值

2.4.10. SCR 控制寄存器 (FMC_SCRCTL)

地址偏移：0x24
复位值：0x0000 FFFF

该寄存器必须以字(32 位)访问。



SCRREN[15:0]
rc_w0

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SCRREN[15:0]	使能SCR区域读取 SCRREN[n]表示SCR数据块n可读且可执行，n=0..15。 默认值为1，软件可写入0，并在此复位周期内锁定。 SCRREN中有效位的数量由SCR_BLOCK_CNT的值决定。如果安全用户区未使能，则此位无效。 0：数据块不可读 1：数据块可读

2.4.11. 选项字节控制寄存器 (FMC_OBCTL)

地址偏移：0x28
复位值：0x0XXX XXXX

该寄存器需以字(32 位)方式访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					DCACHE_ECCEN	ICACHE_ECCEN	SRAM_RST	BOREF_TH			FWDGSP D_DPSLP	FWDGSP D_STDBY	nBOOT0	nBOOT1	nSWBT0
					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
BOR_TH			BOR_EN	SRAM_ECCEN	nRST_STDBY	nRST_DPSLP	nFWDG_HW	SPC[7:0]							
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	DCACHE_ECCEN	使能DCACHE ECC 0：失能DCACHE ECC 1：使能 DCACHE ECC
25	ICACHE_ECCEN	使能ICACHE ECC 0：失能ICACHE ECC 1：使能ICACHE ECC
24	SRAM_RST	SRAM初始化 0：上电复位后初始化SRAM 1：上电复位后不初始化 SRAM **注：**当通过写 OBSTART 位手动触发 Option Bytes 重新加载时，芯片将产生一次系统复位。若本次重新加载使 Option Bytes 中的 SRAM_RST 位由加

		载前"1"改为加载后"0"时，则在该系统复位完成后，SRAM 将执行一次硬件初始化。
23:21	BOREF_TH	BOR eFlash阈值 000：1.61V上升阈值，1.57V下降阈值 001：1.66V上升阈值，1.62V下降阈值 010：1.71V上升阈值，1.67V下降阈值 011：1.76V上升阈值，1.72V下降阈值 100：1.81V上升阈值，1.77V下降阈值 101~111: 1.86V上升阈值，1.82V下降阈值
20	FWDGSPD_DPSLP	FWDGT 在深睡眠模式配置中的挂起选项 0：FWDGT 在系统深睡眠模式下挂起 1：FWDGT 在系统深睡眠模式下运行。
19	FWDGSPD_STDBY	FWDGT 在待机模式配置中的挂起选项 0：FWDGT 在系统待机模式下挂起 1：FWDGT 在系统待机模式下运行
18	nBOOT0	Boot0 配置位 0：BOOT0 为 1 1：BOOT0 为 0
17	nBOOT1	Boot1 配置位 0：BOOT1 为 1 1：BOOT1 为 0 与 BOOT0 引脚一起，该位选择启动模式
16	nSWBT0	软件 BOOT0 失能 0：BOOT0 取决于选项位 nBOOT0 1：BOOT0 取决于 PB11/BOOT0 引脚
15:13	BOR_TH	BOR_TH电平选择 000：3.0V上升阈值，2.9V下降阈值 001：2.9V上升阈值，2.8V下降阈值 010：2.7V上升阈值，2.6V下降阈值 011：2.6V上升阈值，2.5V下降阈值 100：2.5V上升阈值，2.4V下降阈值 101：2.3V上升阈值，2.2V下降阈值 110：2.1V上升阈值，2.0V下降阈值 111：1.85V上升阈值，1.75V下降阈值
12	BOR_EN	BOR_EN 0：BOR使能 1：BOR失能
11	SRAM_ECCEN	SRAM ECC 使能 0：SRAM ECC 失能

		1：SRAM ECC 使能
10	nRST_STDBY	选项字节待机复位值 0：进入待机模式时生成复位 1：进入待机模式时不复位
9	nRST_DPSLP	选项字节深度睡眠复位值 0：进入深度睡眠模式时生成复位 1：进入深度睡眠模式时不复位
8	nFWDG_HW	独立看门狗配置位 0：硬件独立看门狗 1：软件独立看门狗
7:0	SPC[7:0]	系统复位后存储选项字节块的 SPC。

2.4.12. SCR 写保护寄存器 (FMC_SCRWLK)

地址偏移：0x2C
复位值：0x0000 FFFF

该寄存器必须以字(32 位)访问。

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

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

rc_w0

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SCRWLK[15:0]	使能 SCR 写锁。 SCRWLK[n]指示SCR数据块n已写锁，n=0..15。 0：数据块锁定 1：数据块未锁定 当一个数据块被锁定时，无法移除写锁。 SCRWLK中有效位的数量由SCR_BLOCK_CNT的值决定。如果安全用户区未使能，则该位无效。

2.4.13. 擦除/编程保护寄存器 (FMC_WPx) (x = 0…3)

地址偏移：0x30 + 0x04 * x
复位值：0x0000 00FF

该寄存器需以字(32 位)方式访问。

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	WPx_EADDR[7:0]	WP 区域结束页
15:8	保留	必须保持复位值。
7:0	WPx_SADDR[7:0]	WP区域起始页

2.4.14. OTP DATA1 低块锁定寄存器(FMC_OTPDATA1_LLK)

地址偏移：0x44
复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTPDATA1_LWLK[31:16]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA1_LWLK[15:0]															
rc_w0															

位/位域	名称	描述
31:0	OTPDATA1_LWLK[31:0]	OTP USER_DATA1 块写锁 块 n 对应 OTP USER_DATA1 的 128 位 * n。 LWLK[n] = 0：表示 OTP 块 n 中所有 OTP USER_DATA1 128 位已锁定，尝试编程将导致 WPERR。 LWLK[n] = 1：表示 OTP 块 n 中所有 OTP USER_DATA1 128 位未锁定。 当一个块被锁定时，无法移除写保护。

2.4.15. OTP DATA1 高块锁定寄存器 (FMC_OTPDATA1_HLK)

地址偏移：0x48
复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTPDATA1_HWLK[63:48]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA1_HWLK[47:32]															
rc_w0															

位/位域	名称	描述
31:0	OTPDATA1_HWLK[63:32]	OTP DATA1 块写锁 块 n 对应 OTP DATA1 128 位 * n HWLK[n] = 0：表示 OTP 块 n 中的所有 OTP DATA1 128 位已被锁定，尝试对其编程将导致 WPERR。 HWLK[n] = 1：表示 OTP 块 n 中的所有 OTP DATA1 128 位未被锁定。 当一个块被锁定时，无法移除写保护。

2.4.16. OTP DATA2 低块锁定寄存器 (FMC_OTPDATA2_LLK)

地址偏移：0x4C
复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTPDATA2_LWLK[31:16]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_LWLK[15:0]															
rc_w0															

位/位域	名称	描述
31:0	OTPDATA2_LWLK[31:0]	OTP USER_DATA2块写保护锁定 块n对应于OTP USER_DATA2 128-bits * n。 LWLK[n] = 0：表示OTP块n中的所有OTP USER_DATA2 128位已被锁定，尝试对其编程将导致WPERR。 LWLK[n] = 1：表示OTP块n中的所有 OTP USER_DATA2 128位未被锁定。 当一个块被锁定时，无法移除写保护。

2.4.17. OTP DATA2 高块锁定寄存器 (FMC_OTPDATA2_HLK)

地址偏移：0x50
复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTPDATA2_HWLK[63:48]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_HWLK[47:32]															
rc_w0															

位/位域	名称	描述
31:0	OTPDATA2_HWLK[63:32]	OTP USER_DATA2块写保护锁定 块n对应于OTP USER_DATA2 128-bits * n。 HWLK[n] = 0：表示OTP块n中的所有OTP USER_DATA2 128位已被锁定，尝试对其编程将导致WPERR。 HWLK[n] = 1：表示OTP块n中的所有OTP USER_DATA2 128位未被锁定。 当一个块被锁定时，无法移除写保护。

2.4.18. OTP DATA2 低读取控制寄存器 (FMC_OTPDATA2_LR_CTL)

地址偏移：0x54

复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OTPDATA2_LREN[31:16]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_LREN[15:0]															
rc_w0															

位/位域	名称	描述
31:0	OTPDATA2_LREN[31:0]	使能OTP USER_DATA2块读取。 块n对应于OTP USER_DATA2 128-bits * n。 LREN[n] = 1：表示OTP块n中的所有OTP USER_DATA2 128-bits可读。 LREN[n] = 0：表示OTP块n中的所有OTP USER_DATA2 128-bits不可读。 软件可写0，但在本次复位周期内不可修改。硬件在复位后置1。

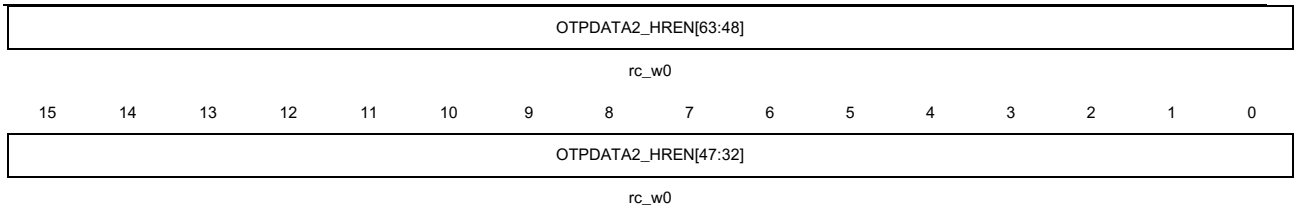
2.4.19. OTP DATA2 高读取控制寄存器 (FMC_OTPDATA2_HR_CTL)

地址偏移：0x58

复位值：0xFFFF FFFF

该寄存器必须以字(32 位)访问。

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



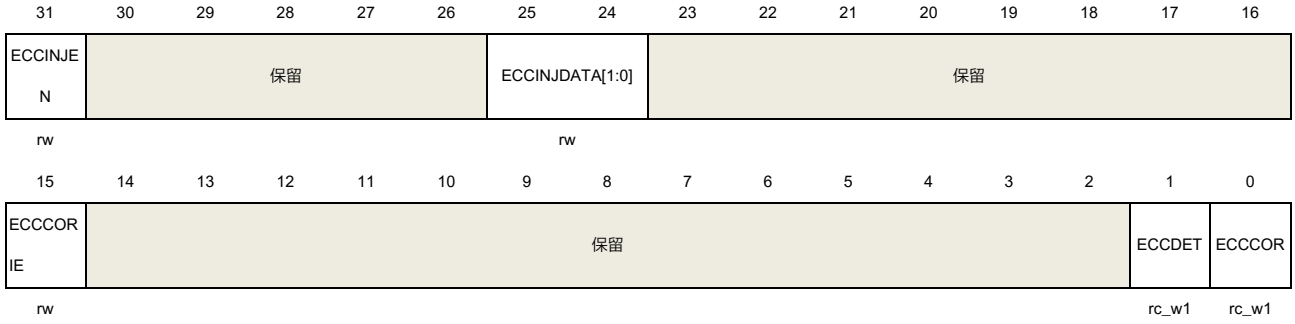
位/位域	名称	描述
31:0	OTPDATA2_HREN[63:32]	使能 OTP USER_DATA2 块读取。 块 n 对应于 OTP USER_DATA2 128-bits * n。 HREN[n] = 1：表示 OTP 块 n 中的所有 OTP USER_DATA2 128-bits 可读。 HREN[n] = 0：表示 OTP 块 n 中的所有 OTP USER_DATA2 128-bits 不可读。 软件可写 0，但在本次复位周期内不可修改。硬件在复位后置 1。

2.4.20. ECC 控制和状态寄存器 (FMC_ECCCS)

地址偏移：0x60

复位值：0x0000 0000

该寄存器需以字(32 位)方式访问。



位/位域	名称	描述
31	ECCINJEN	使能 ECC 故障注入。 0：失能 ECC 故障注入。 1：使能ECC故障注入。
30:26	保留	必须保持复位值。
25:24	ECCINJDATA[1:0]	当 ECCINJEN 被置 1 时，这些位字段可用于控制向 Flash 存储器注入一位或两位错误故障。 00：不注入错误故障。 01：单比特错误故障注入(bit0) 10：单比特错误故障注入(bit1) 11：双比特错误故障注入(bit0 和 bit1) **注意：**这些位字段仅可用于将错误注入到 Flash 存储器数据的 bit0 和 bit1 中。
23:16	保留	必须保持复位值。

15	ECCCORIE	单比特可纠错错误中断使能。 0：单比特可纠错错误中断失能。 1：单比特可纠错错误中断使能。
14:2	保留	必须保持复位值。
1	ECCDET	双比特错误检测标志。当检测到双比特错误时，该位被置位。 该位通过写 1 进行清除。 0：未检测到双比特错误。 1：检测到双比特错误。
0	ECCCOR	单比特可纠错错误检测并校正标志。 该位通过写 1 进行清除。 0：未检测到可纠错单比特错误。 1：检测到可纠错单比特错误。

2.4.21. ECC 地址寄存器 (FMC_ECCADDR)

地址偏移：0x64
复位值：0x0000 0000

该寄存器需以字(32 位)方式访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCEAR[31:16]															
rc_w0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCEAR[15:0]															
rc_w0															

位/位域	名称	描述
31:0	ECCEAR[31:0]	ECC 错误逻辑地址 这些位可通过写入 0 清除。 如果读取地址偏移功能已正确使能，则ECCADDR地址将是偏移后的目标地址。

2.4.22. 产品 ID 寄存器 0 (FMC_PID0)

地址偏移：0x100
复位值：0xXXXX XXXX

该寄存器需以字(32 位)方式访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PID0[31:16]															
r															

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

PID0[15:0]

r

位/位域	名称	描述
31:0	PID0[31:0]	产品保留 ID 代码寄存器 这些位仅供软件读取。 这些位在上电后保持不变。这些位在芯片生产时一次性编程。

2.4.23. 产品 ID 寄存器 1 (FMC_PID1)

地址偏移：0x104
复位值：0xXXXX XXXX

该寄存器需以字(32 位)方式访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PID1[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID1[15:0]															
r															

位/位域	名称	描述
31:0	PID1[31:0]	产品保留 ID 代码寄存器 这些位仅供软件读取。 这些位在上电后保持不变。这些位在芯片生产时一次性编程。

3. 指令缓存(ICACHE)

3.1. 简介

指令缓存(ICACHE)是由 Cortex®-M33 内核的 CBUS 代码总线引入的，用以提升从内部和外部存储介质读取指令和数据时的性能。

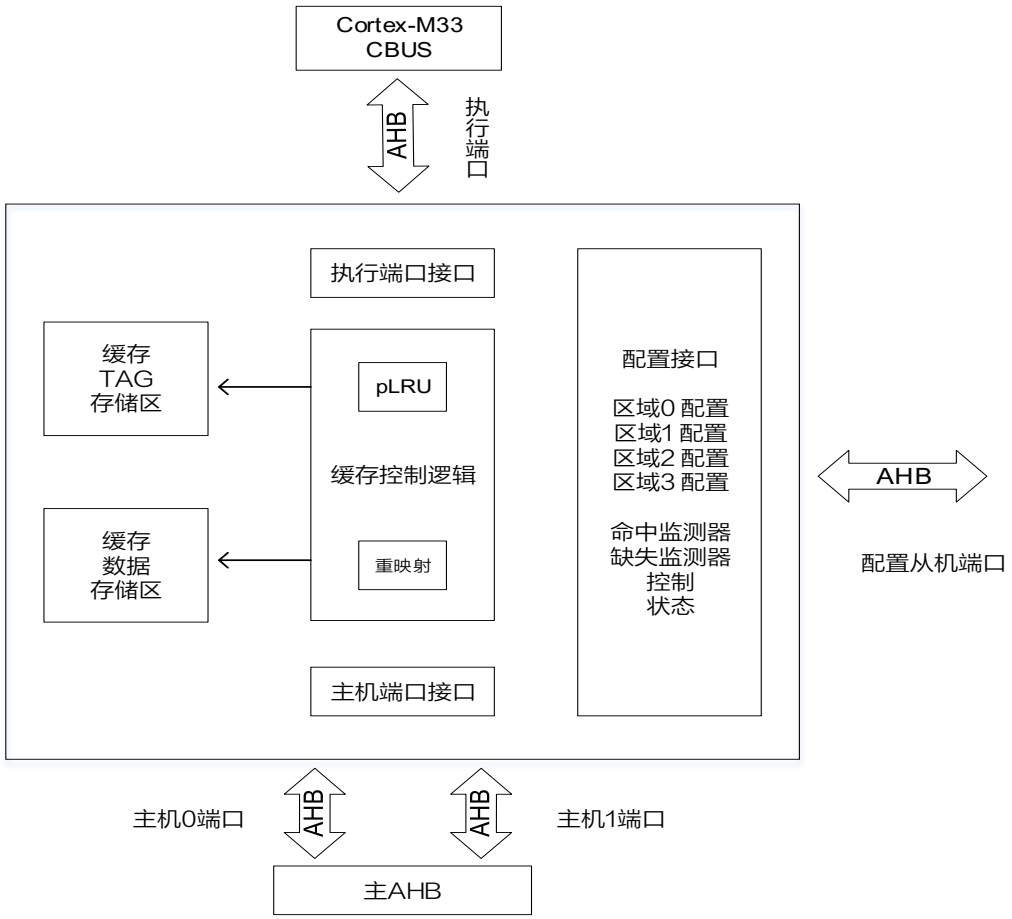
3.2. 主要特性

- 支持 2 路缓存，共计 8KB，每路 256 根缓存线，每根缓存线 16B；
- 支持缓存命中时取址零等待；
- 支持两路 32 位从端口、一路 32 位主端口和一路 128 位主端口：一路从端口是执行端口，另一路是供寄存器访问。一路主端口输出至快总线，而另一路输出至慢总线；
- 缓存访问支持缺失命中和双主机访问特性；
- 支持存储地址重映射；
- 支持 pLRU 替换策略和关键字优先填充策略；
- 支持重映射区域 AHB 事务突发类型配置；
- 实现两个性能计数器：32 位命中监测计数器和 16 位缺失监测计数器；
- 操作管理：缓存失效和可选的中断处理；

3.3. ICACHE 功能描述

指令缓存的目的是缓存指令取值或指令存储加载，ICACHE 仅负责处理读事务，而不负责写事务。对于错误管理，假如写缓存事务出现，ICACHE 置位错误标志，如果中断使能置位，触发一个错误中断。

图 3-1. ICACHE 结构框图



3.3.1. ICACHE 初始化

寄存器 ICACHE_CTL 的位 EN 置 1 时，ICACHE 在相同时钟周期内被使能。如果 ICACHE 被失能，即意味着 ICACHE 被忽略，ICACHE 在启动时的默认状态是失能。

当硬件复位信号释放，缓存失效操作自动发生，每个 TAG 有效位置 0，INVAL 位自动被清除，ICACHE_STAT 寄存器的 BUSY 标志被置位。当该步骤完成时，所有的缓存线有效位被清零，BUSY 标志被清除并且 END 标志被置位。

为了确保其性能，在使能 ICACHE 前，有必要检查缓存失效操作是否完成。在使能 ICACHE 之前，软件必须检测 BUSY 和 END 标志，此外，如果 ICACHE 在无效操作结束前被使能，在 BUSY 被置位期间，任意缓存访问都被视为不可缓冲的，其访问特性取决于主存储的访问时间。

3.3.2. 双主机缓存

ICACHE 支持双端口 AHB 主机：主机 0 和主机 1 端口，这使得 ICACHE 分类数据访问至不同目的存储。主机 0 端口(快总线)访问内部 Flash 和内部 SRAM，主机 1 端口(慢总线)通过 OSPI 接口访问外部 Flash。主机 0 端口用以访问内部存储，而主机 1 端口用以负责至外部存储的重

映射数据流。

双主机特性使得处理器在提取不同存储介质数据时拥有备份方法，这使得到内部 Flash 和到内部 SRAM(如果内部 SRAM 被重映射)的数据流可以被分开，从而降低在 ICACHE 没有命中时的处理器阻塞。

3.3.3. ICACHE TAG 存储

ICACHE TAG 存储包含有效位和地址标记，地址标记用以指示某个数据被包含在缓存数据存储中。

每个缓存线有一个有效位，当缓存线被填充时，该有效位被置位。在下述情景中，有效位被复位：复位 ICACHE、失能缓存和执行 ICACHE 失效操作。

当可缓冲事务在输入执行端口被接收，它的 AHB 地址(AHB_ADDR_in，32 位)由下述域构成：Bits [3:0]、Bits [11:4]和 Bits [31:12]。Bits [3:0]是地址字节偏移，Bits [11:4]是地址缓存线地址，[31:12]是标记地址，AHB 地址的三个域分别指示缓存线内字节偏移，某路内缓存线索引，以及与 TAG 标记地址比较以确认所请求的数据是否在缓存中有效。

TAG 存储的关于两路组相连模式的主要参数详见*表 3-1. 两路组相连模式下 TAG 存储标示尺寸参数*。

表 3-1. 两路组相连模式下 TAG 存储标示尺寸参数

参数	数值
缓存大小	8KB
缓存路数	2
缓存线大小	128-bit
缓存线数目	256 per way
地址字节偏移大小	4-bit
地址路索引大小	8-bit
TAG地址大小	20-bit

3.3.4. 地址重映射

对于外部存储区域，在代码区域定义别名地址。地址重映射被应用在代码别名地址，转换别名地址为目的外部物理地址。重映射机制的根本就是输入 AHB 地址和特定代码区域基地址的映射。

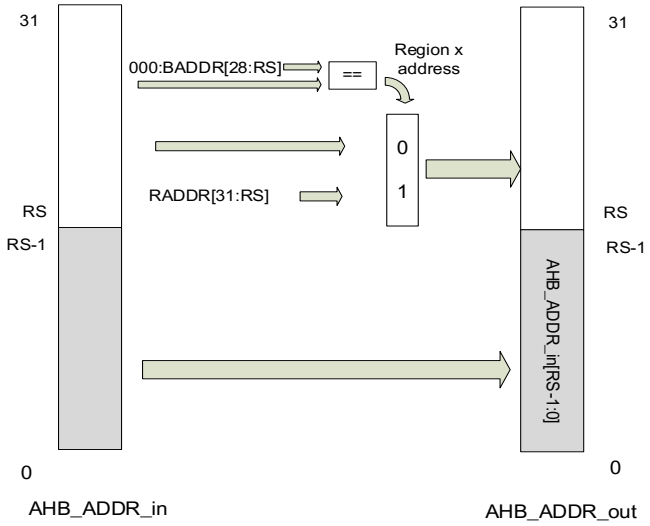
注意：重映射后的别名地址不支持写入操作。

如果 AHB_ADDR_in[31:RS]等于 000：BADDR[28:RS], AHB_ADDR_in 属于区域 x，BADDR 是代码子区域基地址，由 ICACHE_CFGx 寄存器的 BADDR 位定义。RS 定义了有效位的数目，RS 等于 log2(区域大小)，其最小值为 21，对应区域大小是 2MB，其最大值为 27，对应区域大小为 128MB。

如果区域 x 是使能的，主机端口输出 AHB 地址(AHB_ADDR_out)是由 MSBs(ICACHE_CFGx

寄存器的 RADDR[31:RS])和 LSBs(AHB_ADDR_in[RS-1:0])所构成的。

图 3-2. ICACHE 重映射区域大小



当编写 ICACHE_CFGx 寄存器的 BADDR 位和 RADDR 位时，如果所编写的数据比预期数值(MSBs)要大，其没有必要的额外 LSBs 会被忽略。

表 3-2. ICACHE 重映射区域大小

区域大小(MB)	基地址大小(MSBs)	重映射地址(MSBs)
2	8	11
4	7	10
8	6	9
16	5	8
32	4	7
64	3	6
128	2	5

重映射操作可以通过编程寄存器 ICACHE_CFGx 实现软件配置，ICACHE_CFGx 编程仅在 ICACHE 失能时可以实行。每个区域可以独立地通过 ICACHE_CFGx 寄存器被使能，一旦使能后，即使 ICACHE 被失能或事务变为不可缓冲，重映射操作依然被生成。每个区域的大小可以由编写寄存器 ICACHE_CFGx 的位 SIZE 来设置。每个区域的大小是范围粒度(2MB)的二的幂次倍，其区域大小最小为 2MB，最大为 128MB。

软件可以通过设置 ICACHE_CFGx 寄存器的 OBT 位编写 AHB 突发类型 INCR4。

INCR4 突发模式应用于通过 OSPI 接口访问外部存储的情景，INCR4 突发大小等于缓存线大小，INCR4 突发起始地址等于包含请求字的缓存线边界对齐的地址。

3.3.5. 可缓冲和非缓冲访问

为了确保外部存储地址区域可缓冲，ICACHE 支持存储地址重映射特性。它能以缓冲的方式访问多达三个外部存储区域。这些外部存储区域(地址范围[0x6000 0000, 0x6FFF FFFF], [0x7000

0000，0x7FFF FFFF]，[0xB000 0000，0xBFFF FFFF])在代码区域(地址范围[0x0000 0000:0x07FF FFFF]和[0x1000 0000:0x1FFF FFFF])都拥有别名。通过它们的代码别名可以寻址这些外部存储区域，从而实现存储请求被路由至 CBUS 总线并且被 ICACHE 管理。

根据*表 3-3. ICACHE 缓存事务特性*所示存储请求事务查询属性，向 ICACHE 的请求被定义为可缓冲的或不可缓冲的。

表 3-3. ICACHE 缓存事务特性

AHB查询属性	可缓冲性
1	可缓冲的
0	不可缓冲的

如果当前代码区域有存储的请求，ICACHE 首先实现地址重映射功能，如果存储地址需要定义别名，首先重映射地址，再对该地址进行缓冲，重映射之后的目的地物理地址不再需要进一步操作。重映射功能也可用于非缓冲数据流和缓存被失能的情况。

在非缓冲访问的情景下，ICACHE 被忽视，即 AHB 事务被未经改变地传播给主机输出端口，根据地址重映射特性，只有该事务地址可能被改变。忽视 ICACHE 的操作对访问目标存储地址没有影响。

可缓冲的存储区域被用户在 MPU 中定义和编程，MPU 负责针对任意事务寻址给定区域的 AHB 属性信号的产生

表 3-4. 存储配置

存储	可缓(MPU编程)	重映射(ICACHE_CFGx编程)
Flash	是/否	不支持
SRAM	不推荐	不支持
外部存储(OSPI 接口)	是/否	需要

当 ICACHE 接收到来自内核的可缓冲事务时，有两种结果：缓存命中和缓存缺失。

缓存命中：TAG 存储中含有目标地址，缓存线有效，即缓存命中，数据从缓存中读取，并且在相同周期内提供给处理器。

然而，根据缺失命中特性，如果 TAG 存储中不含有目标地址，但属于正在进行的来自主存储的填充突发，也可以达成缓存命中，即使在缓存线填充的期间，只要数据在其主机接口可用，ICACHE 可以快速提供所需要的数据，从而避免缓存缺失。

缓存缺失：TAG 存储中不含有目标地址，即缓存缺失，数据从主存储中读取并提供给处理器，然后执行缓存线填充。关键字优先策略可优化等待周期，该策略允许处理器先读取所请求的数据，然后再执行该请求数据所在缓存线的填充。假如有地址重映射发生，主机端口所产生的突发的类型取决于 ICACHE_CFGx 寄存器的 OBT 位。

在缓存填充的情景，ICACHE 选择缓存线索引。在两路组相连模式中，缓存线索引由每路中的地址索引指定，基于 pLRU 替换算法，一路被当前使用，另一路应用于下一次填充。在两路组相连模式下，如果缓存线被选为填充数据，该缓存线当前必须是有效的，且缓存线必须在填充前进行无效操作。

通过使用 ICACHE 可以减少功耗，大多数时间下可从内部 ICACHE 提取指令，而不是从更大

且功耗更大的主存储，假如所缓存的主存储是外部的，功耗减少的效果更为明显。

3.3.6. ICACHE 性能监测

为了分析性能，ICACHE 提供两个监测器：一个 32 位命中监测器和一个 16 位缺失监测器，默认监视器是失能状态。

命中监视器在 ICACHE 输入端(执行端口)统计 AHB 事务，ICACHE 输出端(主机 0 和主机 1 端口)不再生成事务。考虑所有的访问的地址出现在 TAG 存储或填充缓存中。

缺失监视器在 ICACHE 输入端(执行端口)统计 AHB 事务，ICACHE 输出端(主机 0 和主机 1 端口)生成事务，考虑所有的访问的地址不出现在 TAG 存储和填充缓存中。

命中和缺失监视器可以被使能和复位，软件可以执行下述操作：

- 通过配置 ICACHE_CTL 寄存器的 HMEN 位开启/关闭命中监视器；
- 通过置位 ICACHE_CTL 寄存器的 HMRST 位复位命中监视器；
- 通过配置 ICACHE_CTL 寄存器的 MMEN 位开启/关闭缺失监视器；
- 通过置位 ICACHE_CTL 寄存器的 MMRST 位复位缺失监视器。

3.3.7. ICACHE 错误管理和中断

当不支持的缓冲写请求被检测到时，ICACHE 通过置位 ICACHE_STAT 寄存器的 ERR 标志生成一个错误事件，在此情景下，如果对应的中断使能被打开(ICACHE_INTEN 寄存器的 ERRIE 位使能)，触发错误中断。

另一个可能的中断生成是在缓存无效操作结束时，ICACHE_STAT 寄存器的 END 标志置位，在此情景下，如果对应的中断使能被打开(ICACHE_INTEN 寄存器的 ENDIE 位使能)，触发完成中断，然后缓存再次变为可用。

表 3-5. ICACHE 中断

ICACHE	错误	完成
中断事件	功能错误	操作完成
事件标志	ERR	END
使能控制位	ERRIE	ENDIE
清中断位	ERRC	ENDC

3.4. ICACHE 寄存器

ICACHE基地址：0x4002 6000

3.4.1. 控制寄存器(ICACHE_CTL)

地址偏移：0x00
复位值：0x0000 000C

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												MMRST	HMRST	MMEN	HMEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												BSTT	AMSEL	INVAL	EN
												rw	rw	w	rw

位/位域	名称	描述
31:20	保留	必须保持复位值
19	MMRST	缺失监测器复位 0：没有影响 1：复位缓存缺失监测器
18	HMRST	命中监测器复位 0：没有影响 1：复位缓存命中监测器
17	MMEN	缺失监测器使能 0：关闭缓存缺失监测器，停止监测并不复位 1：缓存缺失监测器使能
16	HMEN	命中监测器使能 0：关闭缓存命中监测器，停止监测并不复位 1：缓存命中监测器使能
15:4	保留	必须保持复位值
3	BSTT	快总线的突发类型 0：WRAP4 1：INCR4
2	AMSEL	缓存组相连模式选择 设置缓存组相连模式，仅当缓存失效时，可软件写 0：没有影响 1：两路组相连缓存(复位值)

1	INVAL	缓存失效，软件置 1，硬件清 0(BUSY 标志置位时) 0：没有影响 1：失效所有缓存
0	EN	使能 0：缓存失能 1：缓存使能

3.4.2. 状态寄存器(ICACHE_STAT)

地址偏移：0x04
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													ERR	END	BUSY
													r	r	r

位/位域	名称	描述
31:3	保留	必须保持复位值
2	ERR	缓存错误标志 0：无错误 1：操作期间错误发生
1	END	完成标志 0：缓存失操作未完成 1：缓存失效操作完成
0	BUSY	忙碌标志 0：缓存不在执行失效操作 1：缓存在执行失效操作

3.4.3. 中断使能寄存器(ICACHE_INTEN)

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字(32 位)访问。

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

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

位/位域	名称	描述
31:3	保留	必须保持复位值
2	ERRIE	缓存错误中断使能 0：错误中断失能 1：错误中断使能
1	ENDIE	缓存操作完成中断使能 0：操作完成中断失能 1：操作完成中断使能
0	保留	必须保持复位值

3.4.4. 标志清除寄存器(ICACHE_FC)

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													ERRC	ENDC	保留
													w	w	

位/位域	名称	描述
31:3	保留	必须保持复位值
2	ERRC	软件清除缓存错误标志 0：无影响 1：清除寄存器 ICACHE_STAT 的 ERR 标志
1	ENDC	软件清除缓存操作完成标志 0：无影响 1：清除寄存器 ICACHE_STAT 的 END 标志
0	保留	必须保持复位值

3.4.5. 命中监测计数寄存器(ICACHE_HMC)

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
HMC[31:16]															
r															

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

位/位域	名称	描述
31:0	HMC[31:0]	缓存命中监测器计数值

3.4.6. 缺失监测计数寄存器(ICACHE_MMC)

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
MMC [15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MMC[15:0]	缓存缺失监测器计数值

3.4.7. 配置寄存器(ICACHE_CFGx)(x = 0...3)

地址偏移：0x20 + 0x04 * x

复位值：0x0000 0200

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
OBT	保留				RADDR										
rw					rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	保留			SIZE			保留	BADDR							
rw				rw				rw							

位/位域	名称	描述
31	OBT	区域 x 的输出突发类型 0：WRAP4 1：INCR4
30:27	保留	必须保持复位值
26:16	RADDR	区域 x 重映射地址，该域替换域 BADDR 所定义的别名地址
15	EN	区域 x 使能 0：失能 1：使能
14:12	保留	必须保持复位值
11:9	SIZE	区域 x 大小 000：保留 001：2 MB 010：4 MB 011：8 MB 100：16 MB 101：32 MB 110：64 MB 111：128 MB
8	保留	必须保持复位值
7:0	BADDR	区域 x 基地址

4. 数据缓存(DCACHE)

4.1. 简介

数据缓存(DCACHE)作为 Cortex®-M33 处理器 S-AHB 系统总线的重要组成部分，旨在提升外部存储器数据传输的整体性能。

DCACHE 采用了多种先进技术，包括命中/未命中并行处理机制和关键字优先填充策略，有效提高了外部存储器的数据访问效率。

4.2. 特性

- 支持4KB缓存，具有2路、每路128个缓存行，每行16B。
- 支持在缓存命中时无等待状态取值操作。
- DCACHE接口支持一个32位从端口、一个32位主端口，以及一个用于寄存器访问的32位从端口。
- 缓存访问支持命中未命中功能。
- 支持写回和写通策略。
- 支持字节、半字和字写入。
- 支持伪随机替换策略和关键字优先填充策略。
- 错误管理和可选中断处理。
- 维护操作：缓存失效、缓存范围失效、缓存范围清除、缓存范围清除并失效。
- 支持四个性能计数器：两个命中监视器计数器(32 位)，计数读取命中和写入命中事件和两个未命中监视器计数器(16 位)，计数读取未命中和写未命中事件。

4.3. 功能概述

数据缓存(DCACHE)是处理器与外部存储器之间的重要高速缓冲层，其主要作用是缓存处理器在执行程序过程中产生的外部存储器访问数据，包括读和写操作。当处理器需要访问外部存储器时，DCACHE 会首先检查目标数据是否已存在于缓存中，从而避免频繁访问较慢的外部存储器，提高整体运算效率。此外，DCACHE 也可对外部地址空间中的指令取值请求进行缓存管理，以进一步优化系统性能。DCACHE 模块负责协调和管理所有读写事务，确保数据一致性与访问高效性。

图 4-1. DCACHE 结构框图

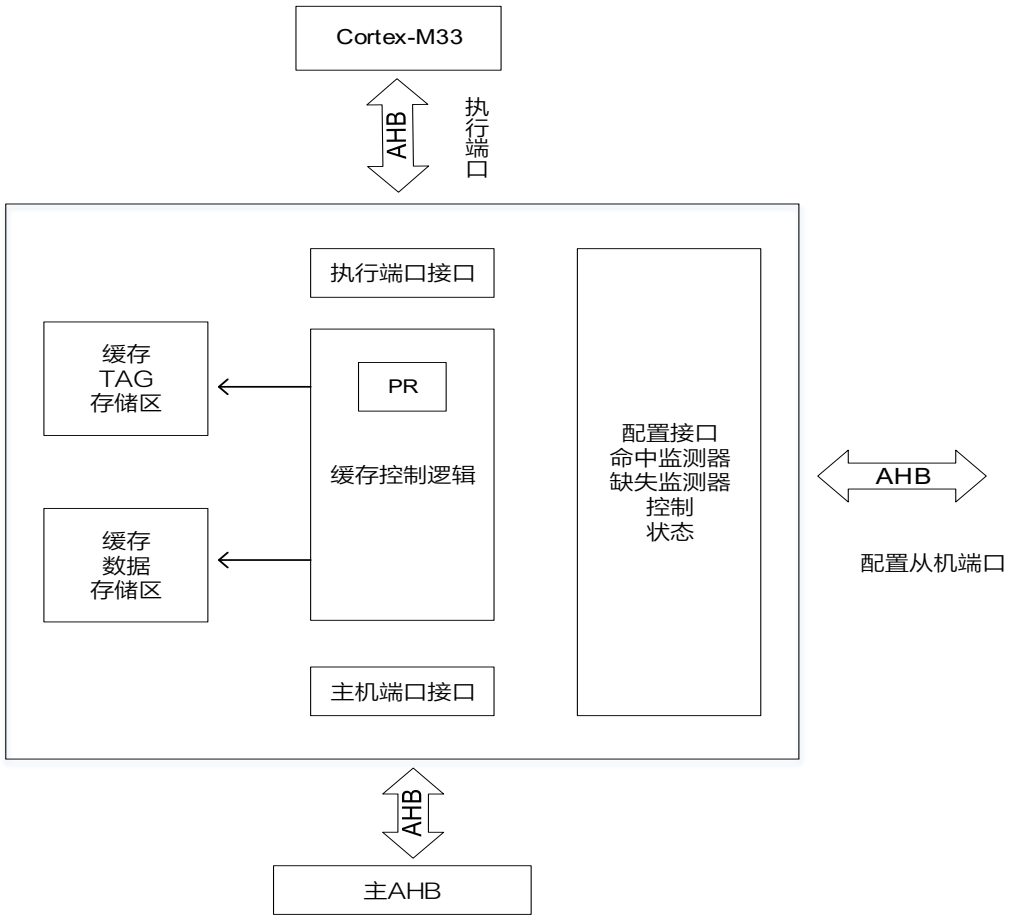
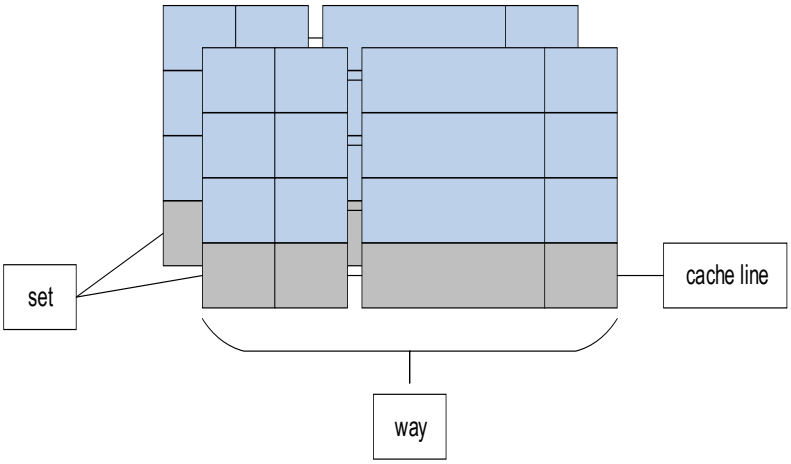


图 4-2. DCACHE 缓存框图



4.3.1. DCACHE 初始化

设置 DCACHE_CTL 寄存器的 DCACHEEN 位即可使能 DCACHE 功能，并在同一时钟周期内

生效。而失能 DCACHE 则意味着忽略 DCACHE，DCACHE 在启动时的默认状态为失能。当复位信号被释放后，缓存失效(Cache Invalidate)操作会自动启动。此时，每个缓存标记(TAG)的有效位将被置为 0，并且 DCACHE_STAT 寄存器中的 MOBUSYF 标志会被置位。当缓存失效操作结束后，所有缓存行的有效位和脏位都会被清除，MOBUSYF 标志将复位，同时 MOENDF 标志会被置位，表示操作完成。

为了确保其性能，在使能 DCACHE 前，有必要检查缓存失效操作是否完成。在使能 DCACHE 之前，软件必须检测 MOBUSYF 和 MOENDF 标志，此外，如果 DCACHE 在无效操作结束前被使能，在 MOBUSYF 被置位期间，任意缓存访问都被视为不可缓冲的，其访问特性取决于主存储的访问时间。

4.3.2. DCACHE TAG 存储

DCACHE TAG 存储器包括脏位、有效位和地址标签位，用于指示缓存数据存储器中包含哪些数据。

图 4-3. 缓存行



TAG 位：存储需要匹配的地址

有效位：每个缓存行都有一个对应的有效位，当该缓存行被重新填充时，有效位会被置位。在某些情况下有效位会被复位，例如复位 DCACHE、失能 DCACHE 以及执行 DCACHE 失效操作。

脏位：脏位指示缓存中的数据是否与主存储器中的数据一致，用于判断是否需要更新主存储器。脏位复位的核心条件是缓存数据已同步到主存或缓存行被废弃，如：

在 DCACHE 中，脏位(Dirty bit)复位的情况包括：

1. 缓存行被清除(Clean)操作

执行 Clean 指令时，脏数据被写回主存储器；

写回完成后，脏位复位(置 0)，表示缓存与主存已同步；

2. 缓存行被无效化(Invalidate)操作

执行 Invalidate 指令时，缓存行被标记为无效；

脏位同时复位，但脏数据会丢失(未写回主存)；

3. 缓存行被清除并无效化(Clean & Invalidate)

先将脏数据写回主存(Clean)；

再将缓存行无效化，脏位复位；

4. 缓存行被替换(Eviction)

发生缓存缺失时，需要替换某个缓存行；

如果被替换行的脏位为 1，先执行写回操作；

写回完成后，脏位复位，新数据填入该行；

5. 系统复位或缓存禁用

硬件复位时，所有脏位清零；

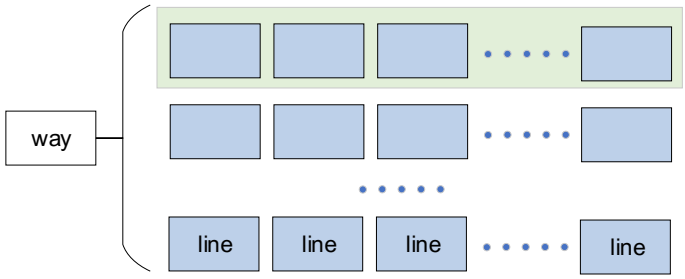
禁用 DCACHE 时，所有脏位复位；

6. 写穿透模式(Write-Through)

在写穿透模式下，写操作同时写入缓存和主存；

脏位始终为 0，不需要复位操作。

图 4-4. 缓存路



DCACHE 默认双路组关联的 TAG 存储器主要参数详见*表 4-1. 两路组相连模式下 TAG 存储标示尺寸参数*。

表 4-1. 两路组相连模式下 TAG 存储标示尺寸参数

参数	值
缓存大小	4KB
缓存路数数量	2
缓存行大小	128-bit
缓存行数量	每路128
地址字节偏移大小	4-bits
地址路索引大小	7-bits
TAG地址大小	21-bits

4.3.3. 可缓存和不可缓存访问

根据内存请求 AHB 事务的内存查找属性，DCache 的写事务可以总结为写通(Write Through)和写回(Write Back)。写通表示所有对 DCache 的写操作会同时在其下一级内存中执行。而写回表示数据写入 DCache 后，相应的缓存行会标记为 Dirty，只有在该缓存行被替换时数据才会写入下一级内存。在需要时，可以强制使数据缓存采用直写(Write-Through)策略。启用强制直写后，缓存将在所有可缓存的访问或传输中始终使用直写方式进行数据更新。要激活该功能，需在 DCACHE_CTL 控制寄存器中将 WTENFP 位设置为 1。

可缓冲的内存区域由用户在 MPU 中定义和配置，MPU 负责根据事务的地址为指定区域生成 AHB 属性信号。*表 4-2. DCACHE 事务的缓存能力*，请求 DCACHE 时定义为可缓存或不可缓存。

可缓冲内存区域由用户在 MPU 中定义和配置，MPU 根据任何事务的地址负责为指定区域生成 AHB 属性信号。

表 4-2. DCACHE 事务的缓存能力

AHB Lookup属性	AHB 缓存属性	缓存策略说明
0	X	读写均不可缓存(Non-cacheable)该类型的访问不会经过数据缓存(DCACHE)，所有读写操作均直接访问主存。
1	0	可缓存读取(Read Cacheable) 写通(Write-Through)策略 当AHB Lookup属性为1且bufferable属性为0时，读取操作可缓存，写入操作采用直写方式，即数据同时写入缓存与主存。
1	1	可缓存读取(Read Cacheable) 回写(Write-Back)策略 当AHB Lookup属性为1且bufferable属性为1时，读取操作可缓存，写入操作采用回写方式，即数据先写入缓存，稍后再回写主存。

当 DCACHE 接收到来自内核的可缓冲事务时，可能出现两种情况：缓存命中和缓存未命中。

读事务：

缓存命中：TAG 包含目标地址，缓存行有效(valid=1)，缓存命中。数据从缓存中读取并在同一周期内提供给处理器。

但是，根据误命中特性，如果 TAG 存储不包含目标地址，但属于来自主内存的持续填充突发，则仍然可以实现缓存命中。在缓存行填充过程中，只要数据在其主机接口上可用，DCACHE 就可以快速提供所需的数据，从而避免缓存未命中。

缓存未命中：TAG 存储不包含目标地址，导致缓存未命中。数据从主存储器读取并提供给处理器，然后进行缓存行填充，是否填充对应的缓存行由 AHB 事务的内存分配属性(memory allocate attribute)决定。关键词优先策略可以优化等待周期，允许处理器首先读取请求的数据，然后继续填充包含请求数据的缓存行。

表 4-3. 读操作

事务	情形	执行操作
读	TAG 存储中含有目标地址，缓存线有效(valid=1)	数据从缓存中读取并在同一周期内提供给处理器
	TAG 存储中不含有目标地址，正在进行的来自主存储的填充突发	数据从缓存中读取并在同一周期内提供给处理器
	TAG 存储中不含有目标地址	从主存储器读取数据并提供给处理器

写回事务：

缓存命中：TAG 存储包含目标地址且缓存行有效(valid=1)，缓存命中。数据写入缓存。

但是，根据误命中特性，如果 TAG 存储不包含目标地址，但属于来自主内存的持续填充突发，则仍然可以实现缓存命中。在缓存行填充过程中，只要数据在其主机接口上可用，DCACHE 就可以快速写入所需的数据，从而避免缓存未命中。

缓存未命中：TAG 存储不包含目标地址，导致缓存未命中。数据从主存储器读取并提供给处理器，然后进行缓存行填充。关键词优先策略可以优化等待周期，允许处理器首先读取请求的数据，然后继续填充包含请求数据的缓存行。

表 4-4. 写回

事务	情形	执行操作
写回	TAG 存储中含有目标地址，缓存线有效(valid=1)	数据写入缓存
	TAG 存储中不含有目标地址，正在进行的来自主存储的填充突发	执行行分配：从主存储器读取整行数据重填缓存。重填完成后，将初始数据写入该缓存行(覆盖刚重填的对应数据部分)。
	TAG 存储中不含有目标地址	直接将数据写入正在重填的缓存行中，无需等待重填完成或再次访问主存储器。

写通事务：

缓存命中：TAG 包含目标地址，缓存行有效(valid=1)，导致缓存命中。数据同时写入缓存和主存储器。

缓存未命中：TAG 不包含目标地址，缓存未命中(valid=0)。来自从端口的数据仅写入主存储器。

表 4-5. 写通

事务	情形	执行操作
写通	TAG 存储中含有目标地址，缓存线有效(valid=1)	数据同时写入缓存和主内存
	TAG 存储中不含有目标地址	从端口传入的数据仅写入主内存

4.3.4. 缓存维护操作

DCACHE 具有多种维护操作，软件可以在控制寄存器中进行编程 DCACHE_CTL 这些操作：

dcache 失效有两种方法：

完全失效：使整个缓存范围失效

软件将 INVALFULL 设置 DCACHE_CTL 为 1 以启用完全失效。启用后，系统会 DCACHE_STAT 设置 FINVLBUSY 标志，指示完全失效作的开始。在执行完全失效期间，每个缓存行中的有效标志将设置为 0，并且每个脏标志将被重置。同时，INVALFULL 会自动清除。完全失效完成后，硬件会清除 FINVLBUSY 标志，并设置 DCACHE_STAT 中的 FINVLENDF 标志。如果启用了 DCACHE_INTEN 中的 BSYENDIE 完成中断，则将触发 DCACHE 中断。

完全失效过程不能中断，缓存在执行过程中不处理任何请求。

指定范围失效：指定地址范围的缓存失效

软件首先设置失效范围(起始地址为 MOSADDR，结束地址为 MOEADDR)，然后通过 MOM 选择范围失效模式 3b010，最后将 MOSTART 设置为 1，使能对指定地址范围失效，同时将设置 DCACHE_STAT 中的 MOBUSYF 标志。启动执行指定地址范围失效后 MOSTART 也将会被清除。操作完成后，DCACHE 会将 MOBUSYF 清除并将 MOENDF 置 1。若使能 MOENDIE 中断位，则 DCACHE 中断将会被触发。在执行此范围失效操作期间，新输入请求将处于挂起状态。

有两种 Dcache 清理方法：

清理指定范围：

清理缓存行的目的是为了确保主存中的内容与缓存中的数据保持同步。清理操作将会把 dirty=1 的缓存行写回主存。

软件首先设置清理范围(起始地址为 MOSADDR，结束地址为 MOEADDR)，然后通过 MOM 选择范围清理模式 3b001，最后将 MOSTART 设置为 1，使能对指定地址范围清理，同时将设置 DCACHE_STAT 中的 MOBUSYF 标志。启动执行指定地址范围清理后 MOSTART 也将会被清除。操作完成后，DCACHE 会将 MOBUSYF 清除并将 MOENDF 置 1。若使能 MOENDIE 中断位，则 DCACHE 中断将会被触发。在执行此范围失效操作期间，新输入请求将处于挂起状态。

清理并失效指定范围：

清理操作将指定范围内 dirty=1 的缓存行写回主存，失效操作会把指定范围内的缓存行失效。

软件首先设置清理并失效的范围(起始地址为 MOSADDR，结束地址为 MOEADDR)，然后通过 MOM 选择范围清理并失效模式 3b011，最后将 MOSTART 设置为 1，使能对指定地址范围清理并失效，同时将设置 DCACHE_STAT 中的 MOBUSYF 标志。启动执行指定地址范围清理并失效后 MOSTART 也将会被清除。操作完成后，DCACHE 会将 MOBUSYF 清除并将 MOENDF 置 1。若使能 MOENDIE 中断位，则 DCACHE 中断将会被触发。在执行此范围失效操作期间，新输入请求将处于挂起状态。

4.3.5. DCACHE 性能监控

为了分析性能，DCACHE 提供了两组监视器：

命中监视器：32 位读取命中监视器和 32 位写入命中监视器;

未命中监视器：16 位读未命中监视器和 16 位写未命中监视器。

默认情况下，监视器处于禁用状态。可以通过 DCACHE_CTL 中的相关 bit 位进行软件开启或者关闭命中或者未命中监视器。

4.4. 寄存器定义

DCACHE 基地址：0x4002 6400

4.4.1. 控制寄存器(DCACHE_CTL)

地址偏移：0x00
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								WMMRST	WHMRST	WMMEN	WHMEN	RMMRST	RHMRST	RMMEN	RHMEN
								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				STARTMO	MOM[2:0]			保留					WTENFP	INVALFULL	DCACHEEN
				rw	rw								rw	w	rw

位/位域	名称	描述
31:24	保留	必须保持在重置值
23	WMMRST	写未命中监视器复位 0：无效果 1：重置写未命中监控
22	WHMRST	写命中监视器复位 0：无效果 1：重置写中监视器
21	WMMEN	写未命中监视器启用 0：关闭写未命中监视器，但不复位。 1：启用写未命中监控
20	WHMEN	写命中监视器启用 0：关闭写命中监视器，但不复位。 1：启用写命中击监视器
19	RMMRST	读未命中监视器复位 0：无效果 1：重置读未命中监视器
18	RHMRST	读命中监视器复位 0：无效果 1：重置读命中监视器

17	RMMEN	读未命中监视器启用 0：关闭读未命中监视器，但不复位。 1：启用读未命中监视器
16	RHMEN	读命中监视器启用 0：关闭读命中监视器，但不重置。 1：启用读命中监视器
15:12	保留	必须保持在重置值
11	STARTMO	启动缓存维护操作。 软件可在以下条件下专门编程：DCACHEEN=1、MOBUSYF=0、FINVLBUSY=0 和 MOM=0b001、0b010 或 0b011。在 MOBUSYF 标志断言时进行硬件重置(在整个缓存服务过程中)。写 0 不会产生任何效果。 0：维护操作完成 1：执行缓存维护操作
10:8	MOM[2:0]	维护操作模式 仅当没有正在进行的维护命令时，才可通过软件设置和清除(MOBUSYF=0)。 000：无操作 001：范围清理 010：范围失效 011：范围清理并失效 100-111：保留
7:3	保留	必须保持在重置值
2	WTENFP	使用强制策略的直写 0：禁用强制策略的直写 1：启用"使用强制策略的直写"
1	INVALFULL	使全部缓存失效 由软件设置，完全失效作时由硬件清除。 0：无效果 1：使整个缓存失效
0	DCACHEEN	启用 Dcache 0：已禁用 Dcache 1：启用 Dcache

4.4.2. 状态寄存器(DCACHE_STAT)

地址偏移：0x04
复位值：0x0000 0001

该寄存器必须以字(32 位)访问。

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											MOENDF	MOBUSYF	ERRF	FINVLENDF	FINVLBUSY
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持在重置值
4	MOENDF	维护操作结束标志 0：缓存繁忙或空闲 1：维护操作业结束
3	MOBUSYF	维护操作繁忙标志 0：维护操作业不忙 1：维护操作业繁忙
2	ERRF	dcache 错误标志 0：无错误 1：作过程中发生错误
1	FINVLENDF	完全无效作结束标志 0：缓存繁忙 1：完全无效作结束
0	FINVLBUSY	完全无效作繁忙标志 0：缓存未执行完全失效作 1：缓存正在执行完全失效作

4.4.3. 中断使能寄存器(DCACHE_INTEN)

地址偏移：0x08
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											MOENDIE	保留	ERRIE	BUSYENDIE	保留
											rw		rw	rw	

位/位域	名称	描述

31:5	保留	必须保持在重置值
4	MOENDIE	维护操作结束中断使能 0：禁用维护操作结束中断 1：启用维护操作结束中断
3	保留	必须保持在重置值
2	ERRIE	缓存错误中断启用 0：禁用缓存错误中断 1：启用缓存错误中断
1	BUSYENDIE	忙端中断使能 0：禁用忙端中断 1：启用忙端中断
0	保留	必须保持在复位值。

4.4.4. 标志清除寄存器(DCACHE_FC)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	MOENDFC	软件清除维护操作结束标志 0：无效果 1：清除 DCACHE_STAT 中的 MOENDF 标志。
3	保留	必须保持在重置值
2	ERRFC	软件清除缓存错误标志 0：无效果 1：清除 DCACHE_STAT 中的 ERRF 标志
1	FINVLENDFC	软件清除完全无效作结束标志 0：无效果

1：清除 DCACHE_STAT 中的 FINVLENDF 标志。

0	保留	必须保持复位值。

4.4.5. 读命中监视计数寄存器(DCACHE_RHMC)

地址偏移：0x10
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31:0	RHMC[31:0]	读命中监视计数器

4.4.6. 读未命中监视计数寄存器(DCACHE_RMMC)

地址偏移：0x14
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RMMC[15:0]	读未命中监视计数器

4.4.7. 写命中监视计数寄存器(DCACHE_WHMC)

地址偏移：0x20
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31:0	WHMC[31:0]	写命中监视计数器

4.4.8. 写未命中监视计数寄存器(DCACHE_WMMC)

地址偏移：0x24
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WMMC[15:0]	写未命中监视计数器

4.4.9. 维护操作起始地址寄存器(DCACHE_MOSADDR)

地址偏移：0x28
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOSADDR[28:13]															
rw															

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

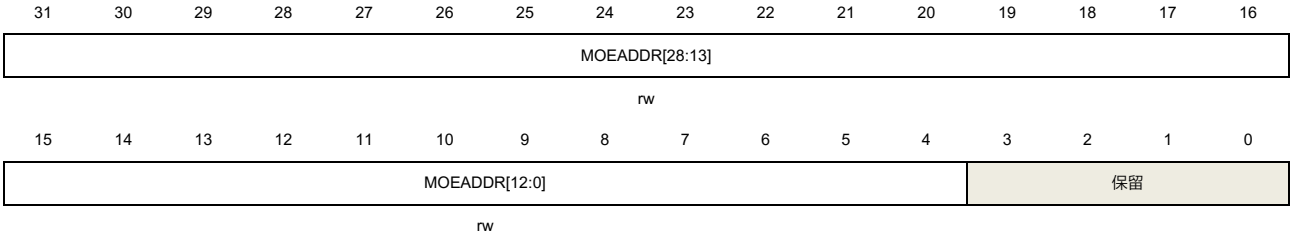
位/位域	名称	描述
31:4	MOSADDR[28:0]	维护操作起始地址

3:0	保留	必须保持复位值。

4.4.10. 维护操作结束地址寄存器 (DCACHE_MOEADDR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器必须以字(32 位)访问。



位/位域	名称	描述
31:4	MOEADDR[28:0]	维护操作结束地址
3:0	保留	必须保持复位值。

5. 电源管理单元(PMU)

5.1. 简介

功耗设计是本系列产品比较注重的问题之一。电源管理单元提供了四种省电模式，包括睡眠模式，深度睡眠模式，深度睡眠模式 1 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。如所示，本系列设备有三个电源域，包括 V_{DD} / V_{DDA} 域，V_{CORE} 域和备份域。V_{DD} / V_{DDA} 域由电源直接供电。在 V_{DD} / V_{DDA} 域中嵌入了一个 LDO，用来为 V_{CORE} 域供电。在备份域中有一个电源切换器，当 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。

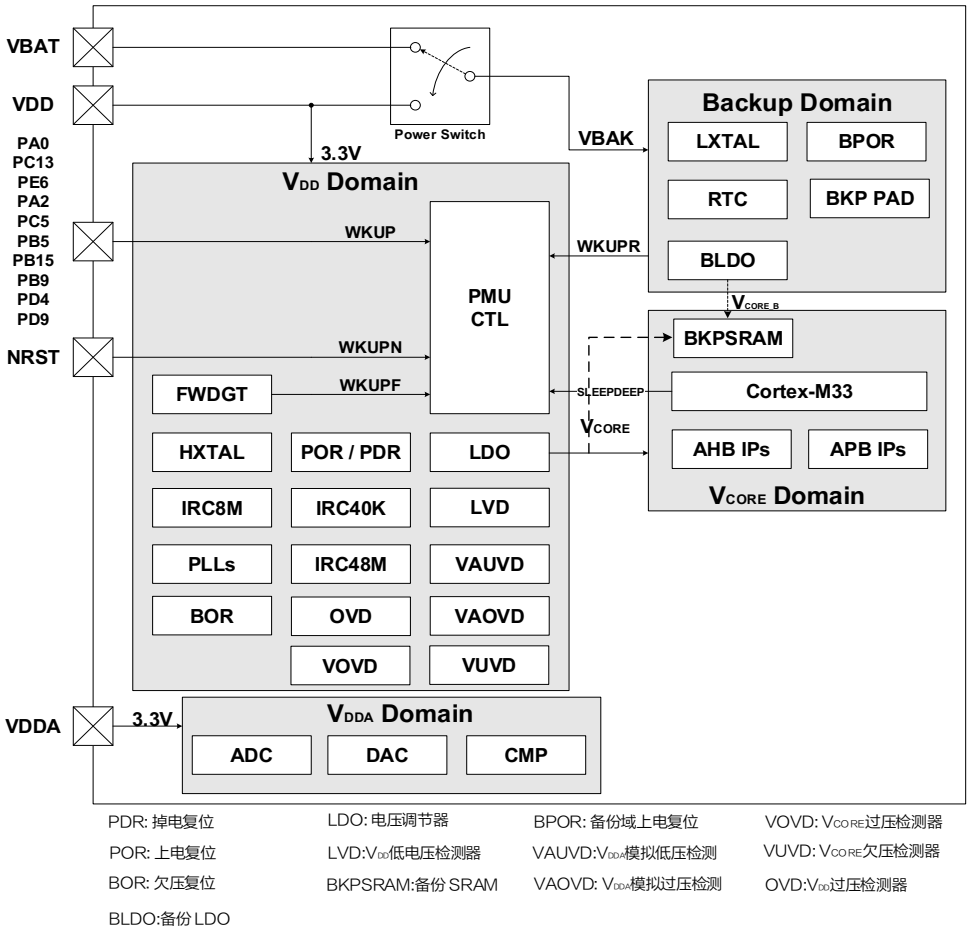
5.2. 主要特性

- 三个电源域：备份域、V_{DD} / V_{DDA}域和V_{CORE}电源域；
- 四种省电模式：睡眠模式、深度睡眠模式，深度睡眠模式1和待机模式；
- 内部电压调节器(LDO)为V_{CORE}电源域提供V_{CORE}电源和备份SRAM电压调节器(BLDO)专用于备份SRAM；
- 提供低电压检测器，当电压低于所设定的阈值时能发出中断或事件；
- 当V_{DD}供电关闭时，由V_{BAT}(电池)为备份域供电；
- 由V_{CORE}(源于V_{DD}或V_{BAK})电源供电的4K字节的SRAM，保证在VDD电源断开的情况下，用户数据也不会丢失。
- LDO输出电压用于节约能耗；
- 供电监控：POR / PDR监控、LVD监控、OVD监控、VAUVD监控、VAOVD监控、VOVD监控、VUVD监控。

5.3. 功能描述

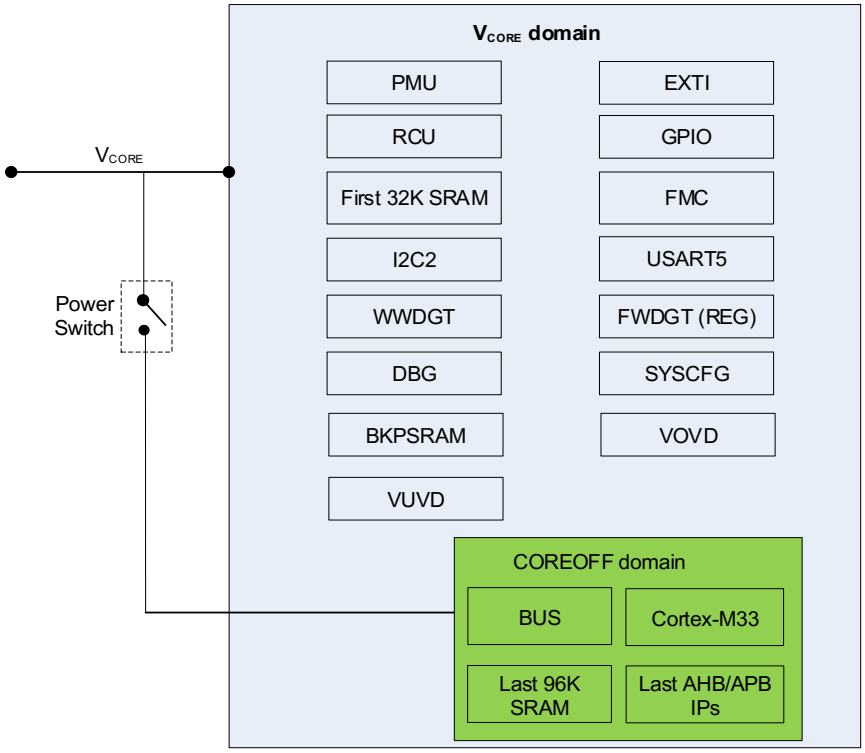
*图 5-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

图 5-1. 电源域概览



PDR: 掉电复位
POR: 上电复位
BOR: 欠压复位
BLDO:备份 LDO
LDO: 电压调节器
LVD:V_{DD}低电压检测器
BKPSRAM:备份 SRAM
BPOR: 备份域上电复位
VAUVD:V_{DDA}模拟低压检测
VAOVD: V_{DDA}模拟过压检测
VOVD: V_{CORE}过压检测器
VUVD: V_{CORE}欠压检测器
OVD:V_{DD}过压检测器

图 5-2. V_{CORE} 域概览



如图*图 5-2. VCORE 域概览*所示，V_{CORE} 域包含两个部分：

1. COREOFF 域包含：
   - 剩余AHB/APB IPs：除了FMC，PMU，RCU，EXTI，GPIO，DBG，FWDGT(仅寄存器)，WWDGT，USART5，I2C2，SYSCFG；
   - 剩余96K SRAM：SRAM(除前32K)；
   - Cortex®-M33 CPU；
   - 总线：包括一个AHB互联矩阵、两个AHB总线和两个APB总线；
2. 深度睡眠保持供电模块：FMC，PMU，RCU，EXTI，GPIO，DBG，FWDGT(仅寄存器)，WWDGT，USART5，I2C2，SYSCFG，BKPSRAM，VOVD，VUVD 以及前 32K SRAM。

电源切换器用于省电模式的供电控制。

5.3.1. 电池备份域

备份域由内部电源切换器来选择 V_{DD} 供电或 V_{BAT}(电池)供电，然后由 V_{BAK} 为备份域供电，该备份域包含 RTC(实时时钟)、LXTAL(低速外部晶体振荡器)、BPOR(备份域上电复位)，BLDO(备份 LDO)以及 PC13 至 PC15 共 3 个 PAD。为了确保备份域中寄存器的内容及 RTC 正常工作，当 V_{DD} 关闭时，VBAT 引脚可以连接至电池或其他电源等备份源供电。电源切换器是由 V_{DD} / V_{DDA} 域掉电复位电路控制的。对于没有外部电池的应用，建议将 VBAT 引脚通过 100nF 的外部陶瓷去耦电容连接到 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 RC 振荡器(IRC40K)或低速外部晶体振荡器(LXTAL)，或高速外部晶体振荡器(HXTAL)时钟 128 分频。当 V_{DD} 被关闭时，RTC 只能选择 LXTAL 作为时钟源。在通过 WFI/WFE 指令进入省电模式之前，Cortex®-M33 需要通过 RTC 寄存器设置预期的闹钟时间并启用闹钟功能，通过 EXTI 线获取 RTC 闹钟事件。进入省电模式一定时间之后，当经过的时间与预设的闹钟时间匹配时，RTC 将唤醒设备。RTC 的配置和操作的细节将在*实时时钟(RTC)*来描述。

当备份域由 V_{DD} 供电(VBAK 连接至 VDD)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(VBAK 连接至 VBAT)，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)。

5.3.2. 备份 SRAM

在 V_{CORE} 电源域内有 4K 字节备份 SRAM。当 PMU_CTL1 寄存器中 BLDOON 被置位时，备份 SRAM 用于在待机模式或 VDD 断开时，存储应用数据。在这些情况下，备份 SRAM 由来自 V_{BAK} 的备份电压调节器 BLDO 供电。BLDO 电压电平由 RTC_CTL 寄存器中的 VCSCAL[1:0]位选择。在其他模式(非待机模式或 VDD 断开)下，系统总线访问备份 SRAM 与访问普通 SRAM 的方式一样，由 V_{DD} 电压调节器供电。

备份 SRAM 只能在 FMC 处于低级安全保护模式时被用户代码访问，以防止不合法的程序或数据访问。如果 FMC 退出低级安全保护模式进入无保护模式，备份 SRAM 被擦除，所有数据丢失。置位 RCU_BDCTL 寄存器的 BKPRST 位，不会复位备份 SRAM 中存储的数据。

5.3.3. V_{DD} / V_{DDA} 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)、LDO(电压调节器)、LVD(低电压检测器)、OVD(过压检测器)、POR / PDR(上电/掉电复位)、FWDGT(独立看门狗定时器)、IRC8M(内部 8M RC 振荡器)、IRC48M(内部 48M RC 振荡器)、VAUVD(模拟电压检测)、VAOVD(模拟过压检测)、IRC40K(内部 40KHz RC 振荡器)和 PLLs(锁相环)和除 PC13、PC14 和 PC15 之外的所有 PAD 等。VDDA 域包括 ADC / DAC(AD / DA 转换器)、CMP(比较器)等。

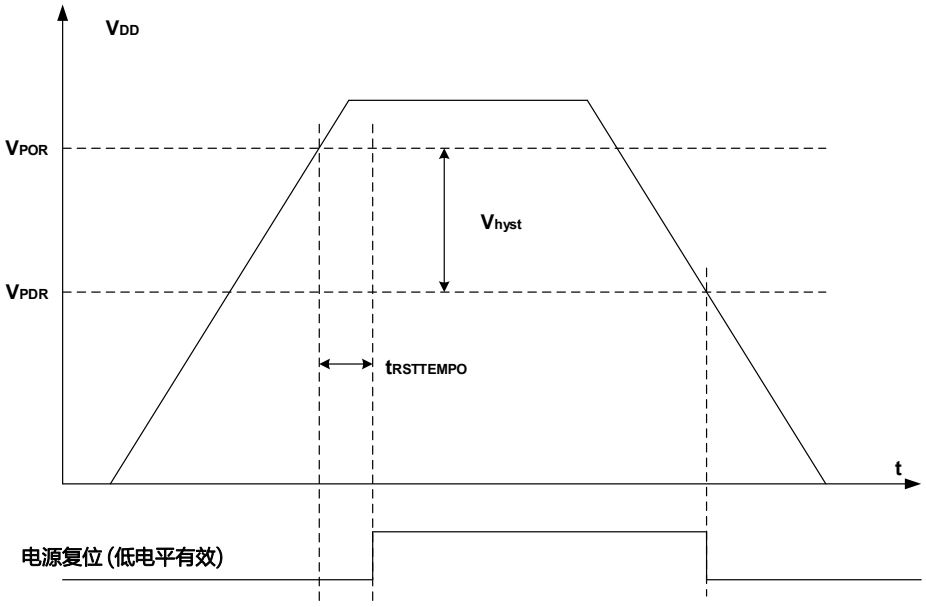
V_{DD} 域

为 V_{CORE} 域供电的 LDO(电压调节器)，其复位后保持使能。可以被配置为三种不同的工作状态：

包括睡眠模式(全供电状态)、深度睡眠模式 / 深度睡眠模式 1(全供电或低功耗状态)和待机模式(关闭状态)。

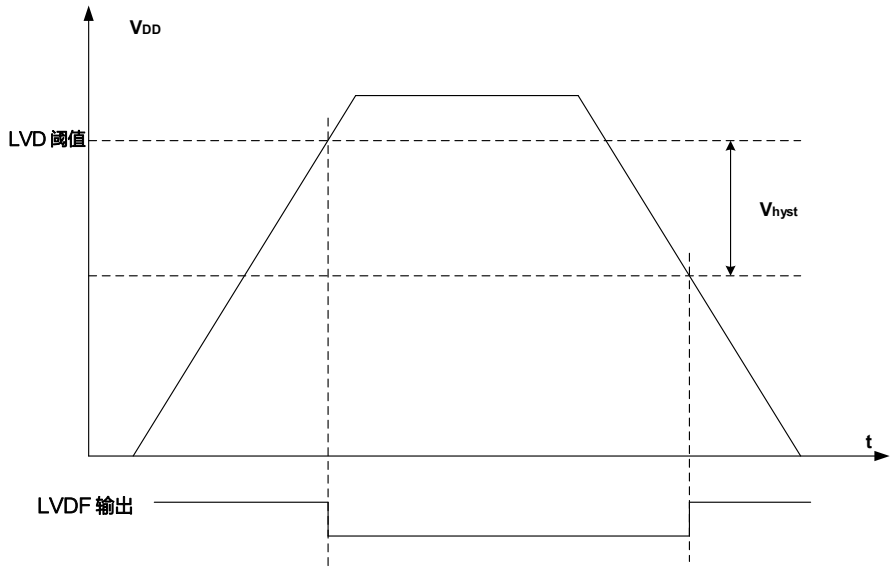
POR / PDR(上电/掉电复位)电路检测 V_{DD} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 5-3. 上电/掉电复位波形图*显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压，V_{PDR} 表示掉电复位的阈值电压，V_{hyst} 表示迟滞电压，具体数值可参考数据手册。

图 5-3. 上电/掉电复位波形图



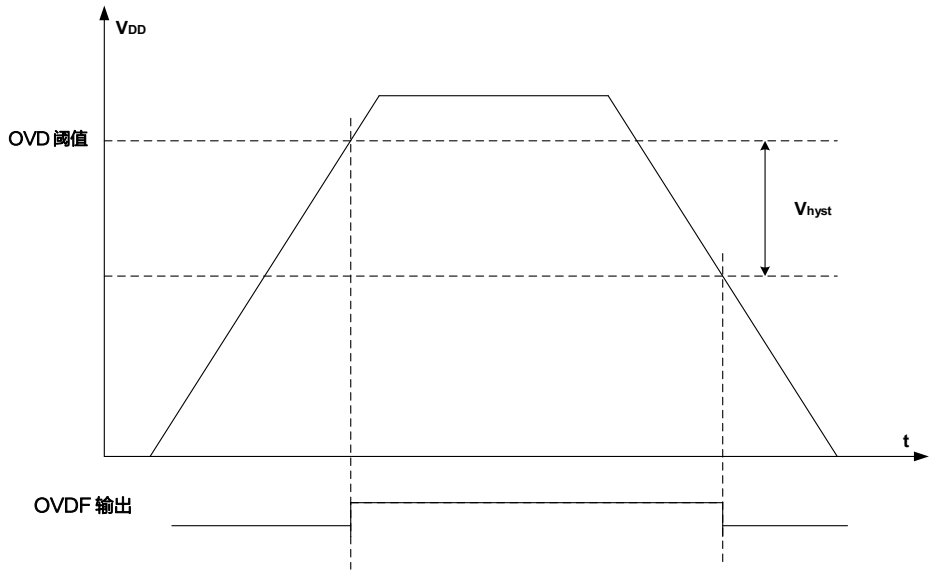
LVD 的功能是检测 V_{DD} 供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器(PMU_CTL0)中的 LVDT[3:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器(PMU_CS0)中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 5-4. LVD 阈值波形图*显示了 V_{DD} 供电电压和 LVD 输出信号的关系。迟滞电压 V_{hyst} 值参考数据手册。

图 5-4. LVD 阈值波形图



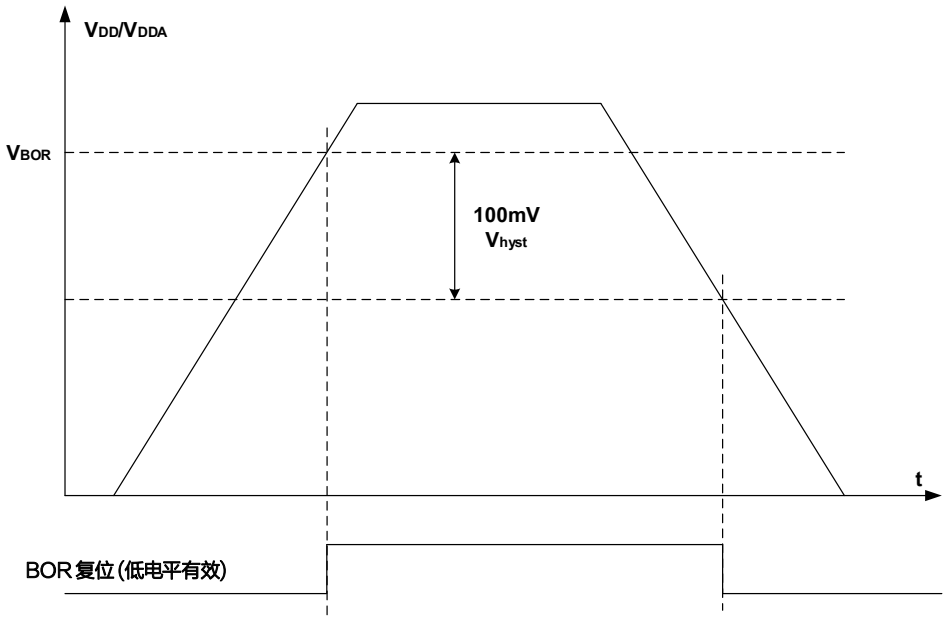
OVD 的功能是检测 V_{DD} 供电电压是否高于过压检测阈值，该阈值由电源控制寄存器(PMU_CTL3)中的 OVDT[1:0]位进行配置。OVD 通过 OVDEN 置位使能，位于电源状态寄存器(PMU_CS0)中的 OVDF 位表示过压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 5-5. OVD 阈值波形图*显示了 V_{DD} 供电电压和 OVD 输出信号的关系。迟滞电压 V_{hyst} 值参考数据手册。

图 5-5. OVD 阈值波形图



BOR 电路检测 V_{DD} 并在电压低于选项字节的 BOR_TH 定义的阈值且选项字节 BOR_EN 位使能时会触发系统复位，复位除备份域和 LDO 之外的整个芯片。*图 5-6. BOR 波形图*显示了供电电压和 BOR 复位信号之间的关系。V_{BOR} 表示 BOR 复位的阈值电压，该值在选项字节 BOR_TH 中定义。迟滞电压 V_{hyst} 值请参考数据手册。

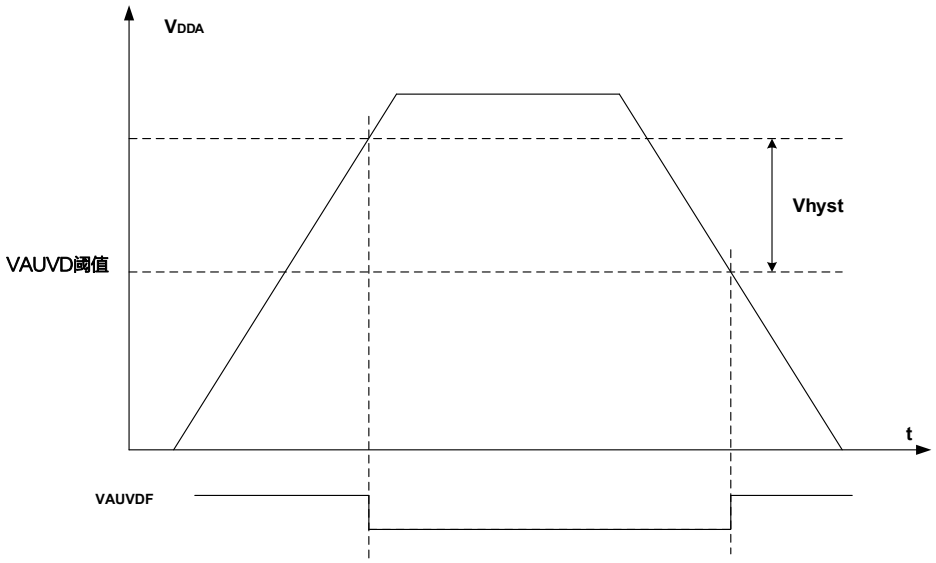
图 5-6. BOR 波形图



V_{DDA} 域

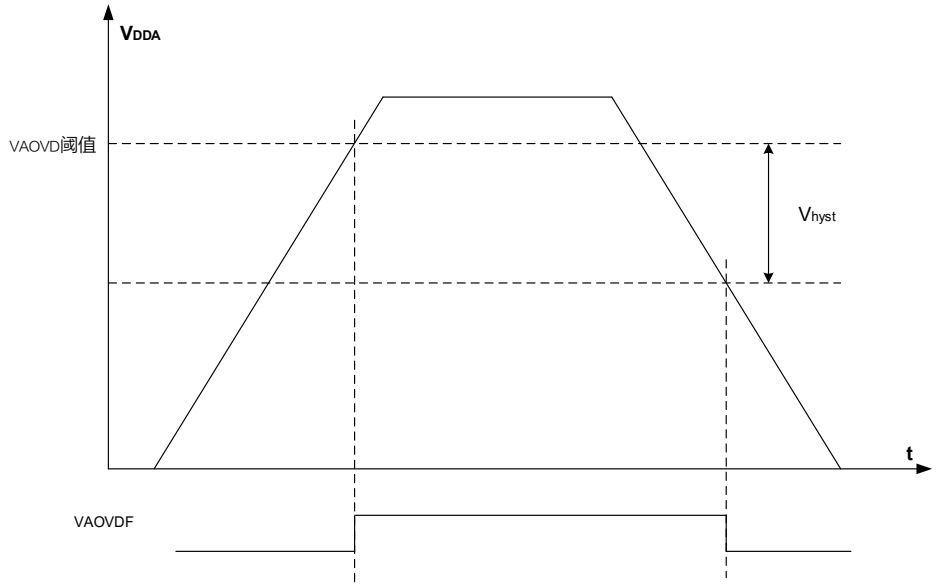
V_{DD} 模拟电压检测器(VAUVD)用于检测 V_{DDA} 电源电压是否低于电源控制寄存器(PMU_CTL0)中 VAUVDVC[3:0]位域选择的编程阈值。通过置位 VAUVDEN 位能够使能 VAUVD，PMU_CS0 寄存器中的 VAUVDF 位指示 V_{DDA} 高于或低于指定的 VAUVD 阈值。该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 5-5. OVD 阈值波形图图 5-7. VAUVD 阈值监测波形图*显示了 VAUVD 门限与 VAUVDF 之间的关系。迟滞电压 V_{hyst} 值参考数据手册。

图 5-7. VAUVD 阈值监测波形图



V_{DDA} 模拟过压检测器(VAOVD)用于检测 V_{DDA} 电源电压是否超过电源控制寄存器(PMU_CTL3)中 VAOVDVC[1:0]位域选择的编程阈值。通过置位 VAOVDEN 位能够使能 VAOVD，PMU_CS0 寄存器中的 VAOVDF 位指示 V_{DDA} 高于或低于指定的 VAOVDVC[1:0]阈值。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. OVD 阈值波形图图 5-8. VAOVDF 阈值监测波形图*显示了 VAOVD 门限与 VAOVDF 之间的关系。迟滞电压 V_{hyst} 值参考数据手册。

图 5-8. VAOVDF 阈值监测波形图



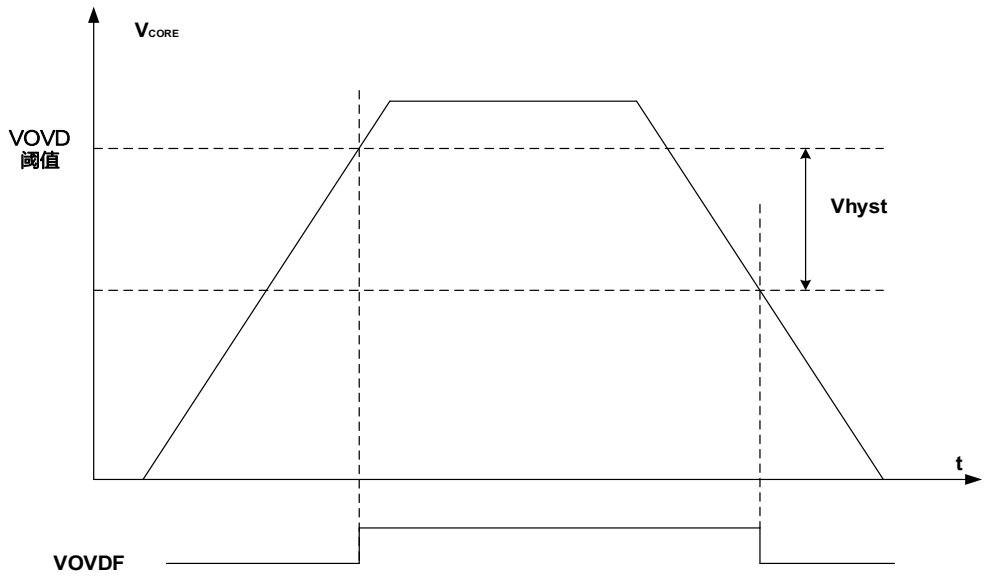
一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，VDDA 通过外部滤波电路连接至 VDD，相应的 VSSA 通过特定电路连接至 VSS。否则，当 V_{DD} 和 V_{DDA} 不是同一个电源提供时，在上电和运行过程中 V_{DD} 与 V_{DDA} 差值不超过 0.3V。

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 VDDA 引脚，或者外部参考电压，外部参考电压的范围请参考*表 18-2. ADC 引脚定义*和*表 19-1. DAC 引脚*。VREFN 须被连接至 VSSA 引脚，VREFP 引脚仅存在于不小于 100-pin 封装上，而在 64-pin 或更少引脚封装不存在，因其内部已经连接至 VDDA 和 VSSA。VREFN 仅存在于不小于 100-pin 封装上，在其他封装里，其内部连接至 VSSA。

5.3.4. V_{CORE} 电源域

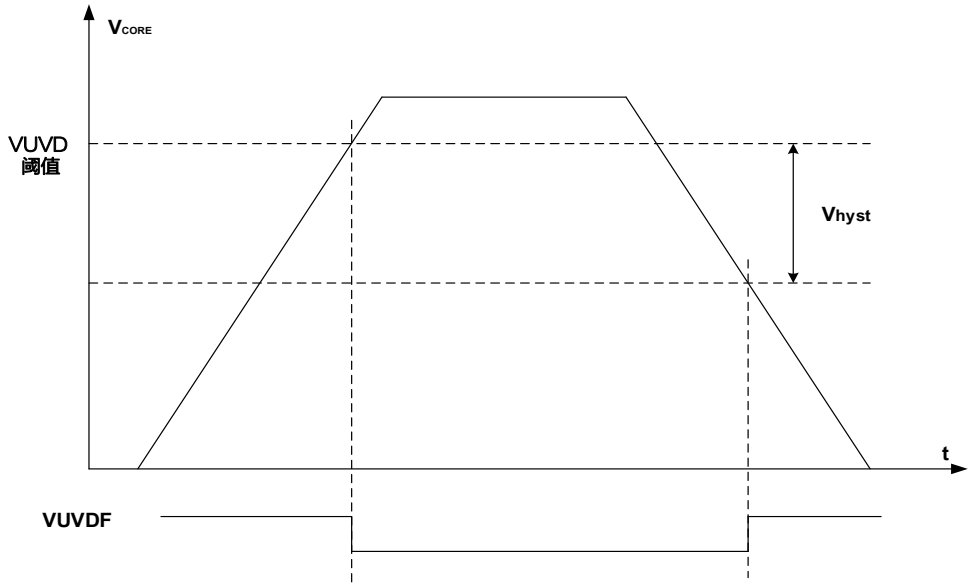
芯片内部有一个 V_{CORE} 电源域电压监测器，当 VOVDEN 为 0b1，将使能 V_{CORE} 电源域电压检测器。一旦 V_{CORE} 电源域超过电源控制寄存器(PMU_CTL0)中 VOVDVC[1:0]位选择的编程阈值，在模拟两个触发器同步后，VOVDF 将立即被置位。通过配置 PMU_CTL2 寄存器中的 VOVDO_DNF[7:0]位，可以使用数字滤波后的 VOVDF。允许抑制峰值的可编程长度为 1024* T_{pclk1} 的 1 到 255。VOVDF 中断与 IRQ 内部连接。*图 5-9. VOVD 阈值监测波形图*显示了 VOVD 门限与 VOVDF 之间的关系,迟滞电压 V_{hyst} 值参考数据手册。

图 5-9. VOVD 阈值监测波形图



芯片内部有一个 V_{CORE} 欠压检测器，当 VUVDEN 为 0b1，将使能 V_{CORE} 电源域电压检测器。一旦 V_{CORE} 电源域低于电源控制寄存器(PMU_CTL0)中 VUVDVC[1:0]位选择的编程阈值，在模拟两个触发器同步后，VUVDF0 将立即被置位。通过配置 PMU_CTL2 寄存器中的 VUVDO_DNF[7:0]位，可以使用数字滤波后的 VUVDF1。允许抑制峰值的可编程长度为 1024* T_{pclk1} 的 1 到 255。VUVDF 中断与 IRQ 内部连接。迟滞电压 V_{hyst} 值参考数据手册。

图 5-10. VUVD 阈值监测波形图



5.3.5. 省电模式

系统复位或电源复位后，MCU 处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有三种：减慢系统时钟(HCLK，PCLK1，PCLK2)，关闭未使用的外设的时钟。此外，

四种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式，深度睡眠模式 1 和待机模式。

睡眠模式

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

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

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

深度睡眠模式

深度睡眠模式与 Cortex®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，V_{CORE} 域中的所有时钟全部关闭，IRC8M、IRC48M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL0 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒(如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M33 技术手册)。刚退出深度睡眠模式时，IRC8M 被选中作为系统时钟。注意，如果 LDO 在低功耗模式下工作，则会产生额外的唤醒延迟。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态(在 EXTI_PD 寄存器中)和和相关外设标志位必须被复位，参考*表 8-3. EXTI 触发源*。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

深度睡眠模式 1

深度睡眠模式 1 与 Cortex®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式 1 下，V_{CORE} 域中的所有时钟全部关闭，IRC8M、IRC48M、HXTAL 及 PLLs 也全部被禁用。COREOFF 域停止供电。SRAM(除前 32K)和 COREOFF 域外设寄存器中的内容全部丢失。根据 PMU_CTL0 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式 1 之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，再置位 PMU_CTL1 寄存器的 DPMOD1 位，然后执行 WFI 或 WFE 指令即

可进入深度睡眠模式 1。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式 1 中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式 1 中唤醒(如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M33 技术手册)。刚退出深度睡眠模式 1 时，IRC8M 被选中作为系统时钟。从深度睡眠模式 1 唤醒需要额外的延时时间用于给 COREOFF 域上电。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

注意：为了顺利进入深度睡眠模式 1，所有 EXTI 线上的挂起状态(在 EXTI_PD 寄存器中)和相关外设标志位必须被复位。否则，程序将直接跳过深度睡眠模式 1 进入过程而继续执行下面的程序。

待机模式

待机模式是基于 Cortex®-M33 的 SLEEPDEEP 模式实现的。在待机模式下，整个 V_{CORE} 域全部停止供电，同时 LDO 和包括 IRC8M、IRC48M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 STBMOD 位置 1，再清除 PMU_CS0 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS0 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有五个唤醒源，包括来自 NRST 引脚的外部复位，RTC 事件，FWDGT 复位，WKUP 引脚的上升沿和 LXTAL 时钟监控，可以参考*表 22-2. 省电模式管理*。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(除了备份 SRAM，当 BLDOON 置位时)的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex®-M33 将从 0x00000000 地址开始执行指令代码。

表 5-1. 节电模式总结

模式	睡眠	深度睡眠	深度睡眠 1	待机
描述	仅关闭 CPU 时钟	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL	1、关闭 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL 3、关闭 CPU、SRAM(除前 32K)、COREOFF 域外设的供电	1、关闭 V_{CORE} 电源域的供电 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL
LDO 状态	开启 (正常功耗模式)	开启 (正常功耗模式或者低功耗模式)	开启 (正常功耗模式或者低功耗模式)	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 0 DPMOD1 = 1	SLEEPDEEP = 1 STBMOD = 1, WURST = 1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤	若通过 WFI 进入，来自 EXTI 的任何中断可	若通过 WFI 进入，来自 EXTI 的任何中断可唤	1、NRST 引脚 2、WKUP 引脚

模式	睡眠	深度睡眠	深度睡眠 1	待机
	醒； 若通过 WFE 进入，则任何事件(或 SEVONPEND=1 时的中断)均可唤醒	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	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	3、FWDGT 复位 4、RTC 5、LCKM
唤醒延迟	无	1、IRC8M 唤醒时间 2、如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	1、IRC8M 唤醒时间 2、COREOFF 域上电时间 3、如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	上电序列

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

5.4. PMU 寄存器

PMU 基地址：0x4000 7000

5.4.1. 控制寄存器 0(PMU_CTL0)

地址偏移：0x00
复位值：0x0000 C000(从待机模式唤醒后复位)

该寄存器可以按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNLOCK	保留		VUVDVC[1:0]		VOVDVC[1:0]		VUVDEN	VOVDEN	VAUVDVC[3:2]		VAUVDEN	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				VAUVDVC[1:0]		LVDT[3]	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
				rw		rw	rw	rw			rw	rc_w1	rc_w1	rw	rw

位/位域	名称	描述
31	UNLOCK	寄存器 lock 0：其他寄存器和位都不可写 1：其他寄存器和位都可写
30:29	保留	必须保持复位值
28:27	VUVDVC[1:0]	V_{CORE}欠压检测器电压等级配置位(VDD复位) 这些位由软件置位和清除。 00：配置V_{CORE}欠压检测器电压等级为1.075V 01：配置V_{CORE}欠压检测器电压等级为0.975V 1x：配置 V_{CORE} 欠压检测器电压等级为 0.875V
26:25	VOVDVC[3:2]	V_{CORE}过压检测器电压等级配置位 这些位由软件置位和清除。 00：配置V_{CORE}过压检测器电压等级为1.25V 01：配置V_{CORE}过压检测器电压等级为1.30V 10：配置V_{CORE}过压检测器电压等级为1.35V 11：配置V_{CORE}过压检测器电压等级为1.40V
24	VUVDEN	V_{CORE}欠压检测器使能位 该位由软件置位和清除。 0：失能V_{CORE}欠压检测器 1：使能V_{CORE}欠压检测器
23	VOVDEN	V_{CORE}过压检测器使能位 该位由软件置位和清除。

		0：失能V_{CORE}过压检测器 1：使能V_{CORE}过压检测器
22:21	VAUVDVC[3:2]	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。 与11:10位组合使用。
20	VAUVDEN	V_{DDA}模拟电压检测器使能位 该位由软件置位和清除。 0：失能V_{DDA}模拟电压检测器 1：使能V_{DDA}模拟电压检测器
19:12	保留	必须保持复位值
11:10	VAUVDVC[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。 0000：配置V_{DDA}模拟电压检测器电压等级为1.8V 0001：配置V_{DDA}模拟电压检测器电压等级为1.9V 0010：配置V_{DDA}模拟电压检测器电压等级为2.0V 0011：配置V_{DDA}模拟电压检测器电压等级为2.1V 0100：配置V_{DDA}模拟电压检测器电压等级为2.2V 0101：配置V_{DDA}模拟电压检测器电压等级为2.3V 0110：配置V_{DDA}模拟电压检测器电压等级为2.4V 0111：配置V_{DDA}模拟电压检测器电压等级为2.5V 1000：配置V_{DDA}模拟电压检测器电压等级为2.6V 1001：配置V_{DDA}模拟电压检测器电压等级为2.7V 1010：配置V_{DDA}模拟电压检测器电压等级为2.8V 1011：配置V_{DDA}模拟电压检测器电压等级为2.9V 1100：配置V_{DDA}模拟电压检测器电压等级为3.0V 1101：配置V_{DDA}模拟电压检测器电压等级为3.1V 1110：配置V_{DDA}模拟电压检测器电压等级为3.2V 1111：配置 V_{DDA} 模拟电压检测器电压等级为 3.3V
9	LVDT[3]	低电压检测器阈值 与 7:5 位组合使用。
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 0000：1.8V 0001：1.9V 0010：2.0V 0011：2.1V

		0100：2.2V 0101：2.3V 0110：2.4V 0111：2.5V 1000：2.6V 1001：2.7V 1010：2.8V 1011：2.9V 1100：3.0V 1101：3.1V 1110：3.2V 1111：3.3V
4	LVDEN	低电压检测器使能 0：关闭低电压检测器 1：开启低电压检测器
3	STBRST	待机标志复位 0：无影响 1：复位待机标志 读该位，始终返回 0
2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回 0
1	STBMOD	待机模式 0：当 Cortex®-M33 进入 SLEEPDEEP 模式时，系统进入深度睡眠模式 / 深度睡眠模式 1 1：当 Cortex®-M33 进入 SLEEPDEEP 模式时，系统进入待机模式
0	LDOLP	LDO 低功耗模式 0：当系统进入深度睡眠模式 / 深度睡眠模式 1 时，LDO 仍正常工作 1：当系统进入深度睡眠模式 / 深度睡眠模式 1 时，LDO 进入低功耗模式 **注意：**在深度睡眠 / 深度睡眠模式 1 模式下，个别外设可能会开启 IRC8M 时钟来做一些工作。在这种情况下，如果 LDO 正处于低功耗模式，LDO 会自动从低功耗模式切换到正常工作模式，并保持正常工作模式，直到外设工作完毕。

5.4.2. 电源控制和状态寄存器 0(PMU_CS0)

地址偏移：0x04

复位值：0x0000 0000 (从待机模式唤醒后不复位)

注意：受到极性、上下拉、初始输入电平影响，软件使能 WKUP 引脚唤醒功能的瞬间，可能会导致 WUF 置 1，建议软件使能 WUPENx 后，先清除 WUF，再进入低功耗模式。

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							VUVDF1	VOVDF	保留					OVDF	VAOVDF
							r	r						r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUPEN7	LDOVSRF	WUPEN5	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	WUPEN6	WUPEN8	WUPEN9	VUVDF0	VAUVDF	LVDF	STBF	WUF
rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rc_w1	r	r	r	r

位/位域	名称	描述
31:25	保留	必须保持复位值
24	VUVDF1	数字滤波后V_{CORE}欠压检测器标志 位由硬件设置和清除，仅当VUVDEN使能时有效 0：V_{CORE}大于VUVDVC[1:0]阈值。 1：V_{CORE} 小于等于 VUVDVC[1:0 阈值。
23	VOVDF	数字滤波后V_{CORE}过电压检测器标志 位由硬件设置和清除，仅当VOVDEN使能时有效 0：V_{CORE}小于VOVDVC[1:0]阈值。 1：V_{CORE} 大于等于 VOVDVC[1:0]阈值。
22:18	保留	必须保持复位值
17	OVDF	过压状态标志 位由硬件设置和清除，仅当OVDEN使能时有效 0：过压事件没出现(V_{DD} 低于设定的 OVDT[1:0]阈值) 1：过压事件出现(V_{DD} 等于或高于 OVDT[1:0]阈值)
16	VAOVDF	V_{DDA}模拟电压检测器过压状态标志 位由硬件设置和清除，仅当VAOVDEN使能时有效 0：V_{DDA}电压低于VAOVDVC[1:0]位配置阈值。 1：V_{DDA} 电压等于或高于 VAOVDVC[1:0]位配置阈值。
15	WUPEN7	WKUP 引脚 7(PB9)唤醒使能 0：关闭 WKUP 引脚 7 唤醒功能 1：开启 WKUP 引脚 7 唤醒功能 如果 WUPEN7 在进入省电模式之前置 1，WKUP 引脚 7 的上升沿或下降沿会将系统从省电模式唤醒。
14	LDOVSRF	LDO 电压选择就绪标志 0：LDO 电压选择未就绪 1：LDO 电压选择就绪
13	WUPEN5	WKUP 引脚 5(PB5)唤醒使能 0：关闭 WKUP 引脚 5 唤醒功能 1：开启 WKUP 引脚 5 唤醒功能

		如果 WUPEN5 在进入省电模式之前置 1，WKUP 引脚 5 的上升沿或下降沿会将系统从省电模式唤醒。
12	WUPEN4	WKUP 引脚 4(PC5)唤醒使能 0：关闭 WKUP 引脚 4 唤醒功能 1：开启 WKUP 引脚 4 唤醒功能 如果 WUPEN4 在进入省电模式之前置 1，WKUP 引脚 4 的上升沿会将系统从省电模式唤醒。
11	WUPEN3	WKUP 引脚 3(PA2)唤醒使能 0：关闭 WKUP 引脚 3 唤醒功能 1：开启 WKUP 引脚 3 唤醒功能 如果 WUPEN3 在进入省电模式之前置 1，WKUP 引脚 3 的上升沿或下降沿会将系统从省电模式唤醒。
10	WUPEN2	WKUP 引脚 2(PE6)唤醒使能 0：关闭 WKUP 引脚 2 唤醒功能 1：开启 WKUP 引脚 2 唤醒功能 如果 WUPEN2 在进入省电模式之前置 1，WKUP 引脚 2 的上升沿或下降沿会将系统从省电模式唤醒。
9	WUPEN1	WKUP 引脚 1(PC13)唤醒使能 0：关闭 WKUP 引脚 1 唤醒功能 1：开启 WKUP 引脚 1 唤醒功能 如果 WUPEN1 在进入省电模式之前置 1，WKUP 引脚 1 的上升沿或下降沿会将系统从省电模式唤醒。
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 的上升沿或下降沿会将系统从省电模式唤醒。
7	WUPEN6	WKUP 引脚 6(PB15)唤醒使能 0：关闭 WKUP 引脚 6 唤醒功能 1：开启 WKUP 引脚 6 唤醒功能 如果 WUPEN6 在进入省电模式之前置 1，WKUP 引脚 6 的上升沿或下降沿会将系统从省电模式唤醒。
6	WUPEN8	WKUP 引脚 8(PD4)唤醒使能 0：关闭 WKUP 引脚 8 唤醒功能 1：开启 WKUP 引脚 8 唤醒功能 如果 WUPEN8 在进入省电模式之前置 1，WKUP 引脚 8 的上升沿或下降沿会将系统从省电模式唤醒。
5	WUPEN9	WKUP 引脚 9(PD9)唤醒使能 0：关闭 WKUP 引脚 9 唤醒功能

		1：开启 WKUP 引脚 9 唤醒功能 如果 WUPEN9 在进入省电模式之前置 1，WKUP 引脚 9 的上升沿或下降沿会将系统从省电模式唤醒。
4	VUVDF0	记录 V_{CORE} 是否发生欠压复位 0：没有发生欠压复位 1：发生了欠压复位
3	VAUVDF	V_{DDA}模拟电压检测器输出标志 位由硬件设置和清除，仅当VAUVDEN使能时有效 0：V_{DDA}大于等于由VAUVDVC[3:0]位配置的阈值。 1：V_{DDA}小于由VAUVDVC[3:0]位配置的阈值。
2	LVDF	低电压状态标志 0：低电压事件没出现(V_{DD} 高于设定的 LVDT[3:0]阈值) 1：低电压事件出现(V_{DD} 等于或低于 LVDT[3:0]阈值) **注意**：LVD 功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式 1：设备曾进入过待机模式 该位只能由 POR/PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件 1：收到来自 WKUP 引脚，LCKM 或 RTC 事件 该位只能由 POR/PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUPP9	WUPP8	WUPP7	WUPP6	WUPP5	WUPP4	WUPP3	WUPP2	WUPP1	WUPP0	保留			BLDORF	BLDOON	DPMOD1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值
15	WUPP9	WKUP 引脚(PD9)极性选择 0：不反转

		1：反转
14	WUPP8	WKUP 引脚(PD4)极性选择 0：不反转 1：反转
13	WUPP7	WKUP 引脚(PB9)极性选择 0：不反转 1：反转
12	WUPP6	WKUP 引脚(PB15)极性选择 0：不反转 1：反转
11	WUPP5	WKUP 引脚(PB5)极性选择 0：不反转 1：反转
10	WUPP4	WKUP 引脚(PC5)极性选择 0：不反转 1：反转
9	WUPP3	WKUP 引脚(PA2)极性选择 0：不反转 1：反转
8	WUPP2	WKUP 引脚(PE6)极性选择 0：不反转 1：反转
7	WUPP1	WKUP 引脚(PC13)极性选择 0：不反转 1：反转
6	WUPP0	WKUP 引脚(PA0)极性选择 0：不反转 1：反转
5:3	保留	必须保持复位值
2	BLDORF	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
1	BLDOON	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)。 没有置位时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 VBAT 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
0	DPMOD1	深度睡眠模式 1 使能 0：无作用 1：当设置了 SLEEPDEEP 位且 STBMOD 位清除后，进入 Deep-sleep 1 模式。

5.4.4. 电源控制和状态寄存器 1(PMU_CS1)

地址偏移：0x0C
复位值：0x0000 0000(从待机模式唤醒后不复位)

该寄存器可以按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															DPF1
															rc_w0

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

5.4.5. 控制寄存器 2(PMU_CTL2)

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
VUVDO_DNF[7:0]								VOVDO_DNF[7:0]							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:8	VUVDO_DNF[7:0]	VUVD 模拟输出数字噪声滤波器 这些位用于配置 VUVD 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 VUVDO_DNF[7:0] * 1024 * T_{PCLK1} 0：关闭数字滤波器 1：开启数字滤波器，滤波峰值长度高达 1024 * T_{PCLK1} … 255：开启数字滤波器，滤波峰值长度高达 255 * 1024 * T_{PCLK1}
7:0	VOVDO_DNF[7:0]	VOVD 模拟输出数字噪声滤波器 这些位用于配置 VOVD 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 VOVDO_DNF[7:0] * 1024 * T_{PCLK1} 0：关闭数字滤波器 1：开启数字滤波器，滤波峰值长度高达1024 * T_{PCLK1} … 255：开启数字滤波器，滤波峰值长度高达255 * 1024 * T_{PCLK1}

5.4.6. 控制寄存器 3(PMU_CTL3)

地址偏移：0x14
复位值：0x0000 003C

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
WUPPU9	WUPPU8	WUPPU7	WUPPU6	WUPPU5	WUPPU4	WUPPU3	WUPPU2	WUPPU1	WUPPU0	VAOVDVC[1:0]		OVDT[1:0]		VAOVDEN	OVDEN
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	WUPPU9	WKUP 引脚(PD9)上下拉选择 0：下拉 1：上拉
14	WUPPU8	WKUP 引脚(PD4)上下拉选择 0：下拉 1：上拉
13	WUPPU7	WKUP 引脚(PB9)上下拉选择 0：下拉 1：上拉
12	WUPPU6	WKUP 引脚(PB15)上下拉选择

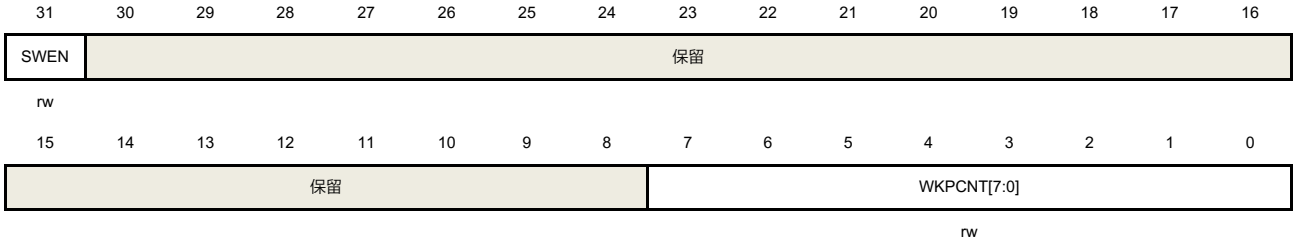
		0：下拉 1：上拉
11	WUPPU5	WKUP 引脚(PB5)上下拉选择 0：下拉 1：上拉
10	WUPPU4	WKUP 引脚(PC5)上下拉选择 0：下拉 1：上拉
9	WUPPU3	WKUP 引脚(PA2)上下拉选择 0：下拉 1：上拉
8	WUPPU2	WKUP 引脚(PE6)上下拉选择 0：下拉 1：上拉
7	WUPPU1	WKUP 引脚(PC13)上下拉选择 0：下拉 1：上拉
6	WUPPU0	WKUP 引脚(PA0)上下拉选择 0：下拉 1：上拉
5:4	VAOVDVC[1:0]	V_{DDA} 过压阈值选择信号，监测 VDDA 电压 00：3.65V 上升阈值 01：3.75V 上升阈值 10：3.85V 上升阈值 11：3.95V 上升阈值
3:2	OVDT[1:0]	V_{DD} 过压阈值选择信号，监测 V_{DD} 电压 00：3.65V 上升阈值 01：3.75V 上升阈值 10：3.85V 上升阈值 11：3.95V 上升阈值
1	VAOVDEN	V_{DDA} 过压检测使能信号 0：过压检测功能关闭 1：过压检测功能开启
0	OVDEN	V_{DD} 过压检测使能信号 0：过压检测功能关闭 1：过压检测功能开启

5.4.7. 参数寄存器(PMU_PAR)

地址偏移：0x18

复位值：0x000A 0040

该寄存器可以按字(32 位)访问。



位/位域	名称	描述
31	SWEN	唤醒深度睡眠模式 1 时是否使用户软件设置的值 0：使用硬件默认值。 1：使用软件配置值，该值由 WKPCNT[7:0]位设置。
30:8	保留	必须保持复位值。
7:0	WKPCNT[7:0]	深度睡眠模式 1 的唤醒时间。2 时钟步长，最大值为 PCLK1*512。

6. 复位和时钟单元(RCU)

6.1. 复位控制单元(RCTL)

6.1.1. 简介

GD32E53x 的复位控制包括三种复位的控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，会重置除了备份域的整个系统。系统复位会复位除了 SW-DP 控制器和备份域的其余部分，包括处理器内核和外设 IP。备份域复位会重置备份域。复位能够被外部信号、内部事件和复位发生器触发。关于这些复位的更多信息将在后续章节中描述。

6.1.2. 功能说明

电源复位

当发生以下任一事件时，产生电源复位：上电/掉电复位(POR/PDR 复位)，从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部 LDO 电源基准准备好提供 V_{CORE} 域电压时，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

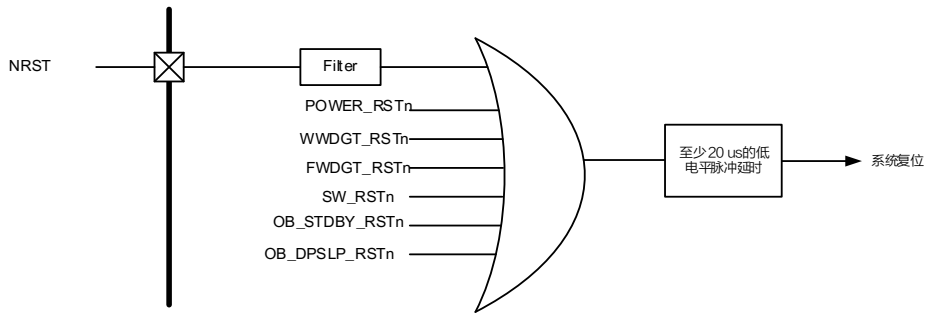
当发生以下任一事件时，产生一个系统复位：

- 电源复位(POWER_RSTn)。
- 外部引脚复位(NRST)。
- 窗口看门狗定时器复位(WWDGT_RSTn)。
- 独立看门狗定时器复位(FWDGT_RSTn)。
- Cortex®-M33的中断应用和复位控制寄存器中的SYSRESETREQ位置'1'(SW_RSTn)。
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位(OB_STDBY_RSTn)；
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位(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

当以下任一事件发生时，将生成备份域复位：

1、备份域控制寄存器中的 BKPRST 位被设置为 1。

2、备份域上电复位发生：在 VDD 和 VBAT 都断电后，VDD 或 VBAT 再次上电。

6.2. 时钟控制单元(CCTL)

6.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部 8M RC 振荡器时钟(IRC8M)、一个内部 48M RC 振荡器时钟(IRC48M)、一个外部高速晶体振荡器时钟(HXTAL)、一个内部 40K RC 振荡器时钟(IRC40K)、一个外部低速晶体振荡器时钟(LXTAL)、两个锁相环(PLLs)、一个 HXTAL 时钟监视器、时钟频率监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB 和 Cortex®-M33 的时钟源自系统时钟(CK_SYS)，系统时钟可以由 IRC8M、HXTAL 或 PLL 提供。系统时钟(CK_SYS)的最大工作频率可达到 252 MHz。

图 6-2. GD32E533 时钟树

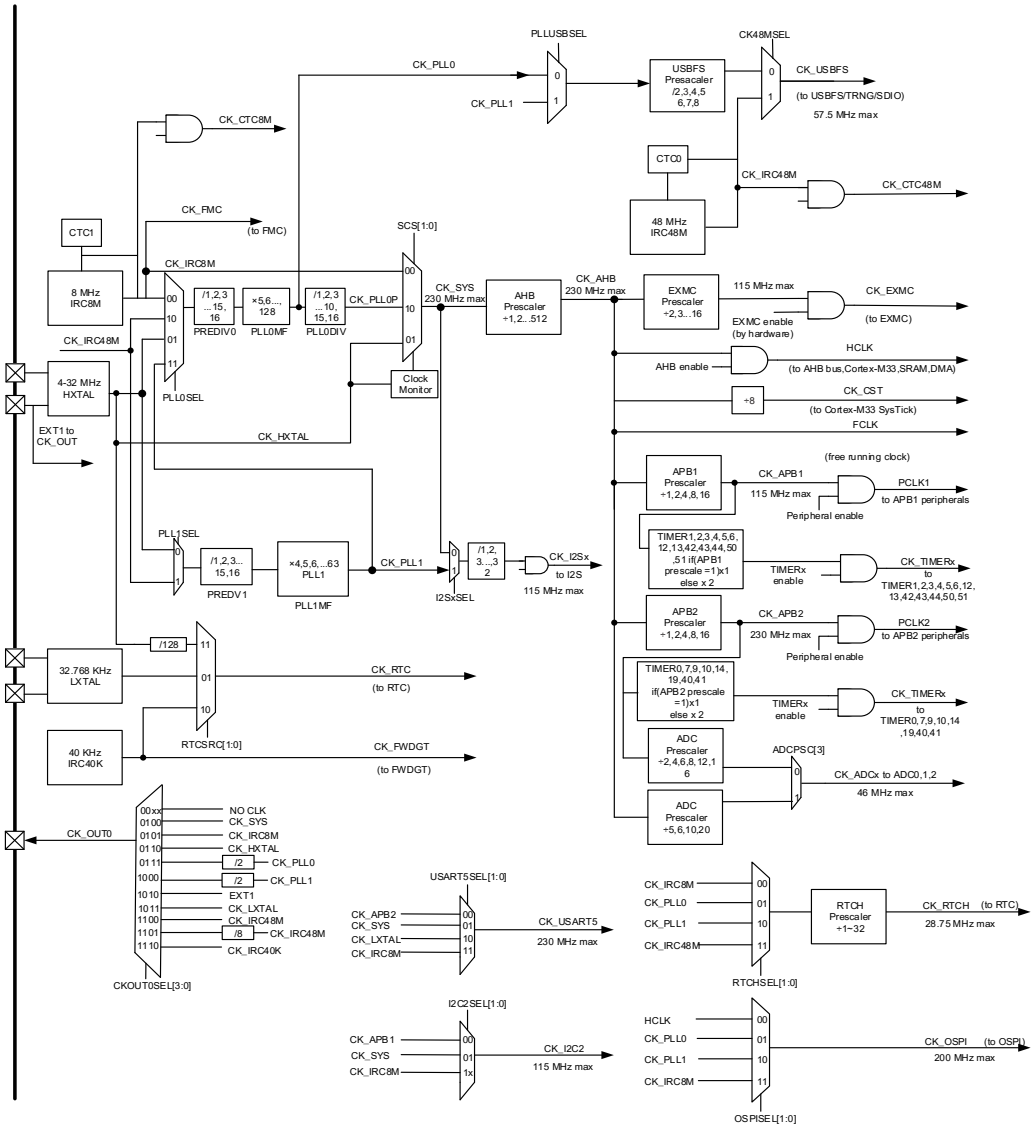
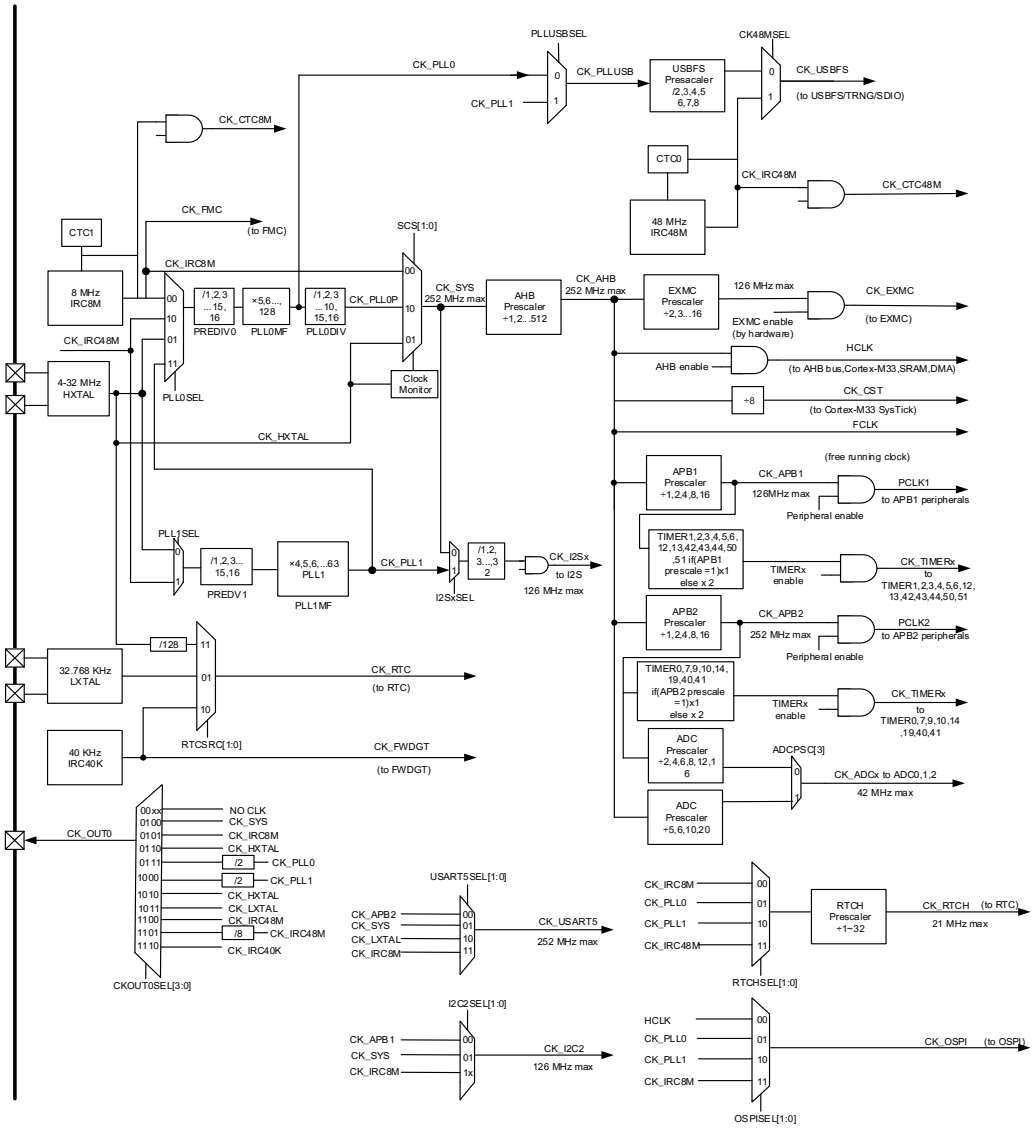


图 6-3. GD32E535 和 GD32E53P 时钟树



AHB、APB2 和 APB1 域的频率可以通过各自的分频器进行配置。AHB、APB2 和 APB1 域的最大频率分别为 252 MHz、252 MHz 和 126 MHz。Cortex®系统定时器(SysTick)的外部时钟由 AHB 时钟(HCLK)除以 8 得到。SysTick 可以工作在该时钟或 AHB 时钟(HCLK)下，这可以在 SysTick 控制和状态寄存器中配置。

模数转换器(ADCs)的时钟由 APB2 时钟除以 2、4、6、8、12，或由 AHB 时钟除以 5、6、10、20 得到，这些值由 RCU_CFG0 和 RCU_CFG1 寄存器中的 ADCPSC 定义。

USART5 由 IRC8M 时钟、LXTAL 时钟、CK_SYS 或 APB2 时钟提供时钟，由 RCU_CFG2 寄存器的 USART5SEL 位选择。

I2C2 由 IRC8M 时钟、CK_SYS 或 APB1 时钟提供时钟，由 RCU_CFG2 寄存器的 I2C2SEL 位选择。

EXMC 的时钟由 CK_AHB 提供。

USBFS、TRNG 和 SDIO 的时钟由 CK_USBFS 提供。

TIMERs 由从 CK_APB2 和 CK_APB1 分频得到的时钟驱动。定时器时钟的频率等于 CK_APBx(如果 APB 分频器为 1)，是 CK_APBx 的两倍(如果 APB 分频器不为 1)。

CTC0 由 IRC48M 时钟驱动。IRC48M 可以由 CTC0 单元自动校准。

CTC1 由 IRC8M 时钟驱动。IRC8M 可以由 CTC1 单元自动校准。

I2S 由 CK_SYS 或 CLK_PLL1 时钟驱动。

RTC 由 LXTAL 时钟或 IRC40K 时钟或 HXTAL 时钟除以 128(由 RCU_BDCTL 寄存器的 RTCSRC 位定义选择)。在 RTC 选择 HXTAL 时钟除以 128 后，当 V_{CORE} 内核域断电时，时钟消失。在 RTC 选择 IRC40K 后，当 VDD 断电时，时钟消失。在 RTC 选择 LXTAL 后，当 VDD 和 VBAT 断电时，时钟消失。

当 FWDGT 启动时，FWDGT 时钟被强制选择由 IRC40K 时钟做为时钟源。

当 FMC 启动时，IRC8M 时钟被强制选择由 FMC 时钟做为时钟源。

6.2.2. 主要特征

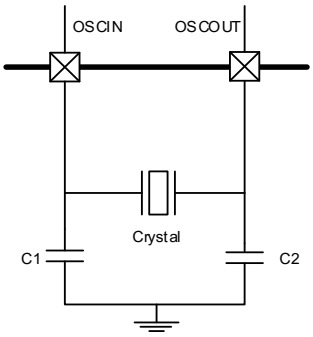
- 4至32 MHz高速晶振(HXTAL)。
- 内部8 MHz RC振荡器(IRC8M)。
- 内部48 MHz RC振荡器(IRC48M)。
- 32,768 Hz低速晶振(LXTAL)。
- 内部40 kHz RC振荡器(IRC40K)。
- PLL0时钟源可以是HXTAL、IRC8M、IRC48M或PLL1。
- HXTAL时钟监测。

6.2.3. 功能说明

高速晶振(HXTAL)

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

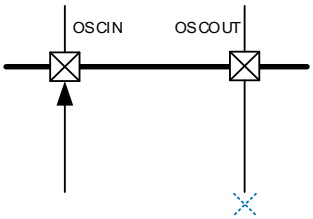
图 6-4. HXTAL 时钟源



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 6-5. HXTAL 时钟源在旁路模式下*所示。HXTAL 时钟源在旁路模式下。CK_HXTAL 等于驱动 OSCIN 引脚的外部时钟。

图 6-5. HXTAL 时钟源在旁路模式下



内部 8M RC 振荡器(IRC8M)

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

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

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

内部 48M RC 振荡器(IRC48M)

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

工厂会校准 IRC48M 时钟频率的精度，但是它的精度仍然不够精准。因为 USB 模块需要的时钟频率必须满足 48MHz(500ppm)。CTC0 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。

相位锁定环(PLL)

有两个内部相位锁定环，PLL0 和 PLL1。

PLL0 可以通过使用 RCU_CTL 寄存器中的 PLL0EN 位来开启或关闭。RCU_CTL 寄存器中的 PLL0STB 标志将指示 PLL0 时钟是否稳定。当 PLL0 稳定时，如果 RCU_INT 寄存器中的相关中断使能位 PLL0STBIE 被设置，可以产生一个中断。

PLL1 可以通过使用 RCU_CTL 寄存器中的 PLL1EN 位来开启或关闭。RCU_CTL 寄存器中的 PLL1STB 标志将指示 PLL1 时钟是否稳定。当 PLL1 稳定时，如果 RCU_INT 寄存器中的相关中断使能位 PLL1STBIE 被设置，可以产生一个中断。

当进入深睡眠/待机模式或在使用 HXTAL 作为 PLLs 的源时钟时 HXTAL 监控失败时，两个 PLL 将被硬件关闭。

低速晶体振荡器(LXTAL)

低速外部晶体或陶瓷谐振器振荡器，其频率为 32,768 Hz，为实时时钟电路产生一个低功耗但高精度的时钟源。LXTAL 振荡器可以通过备份域控制寄存器(RCU_BDCTL)中的 LXTALEN 位来开启或关闭。备份域控制寄存器(RCU_BDCTL)中的 LXTALSTB 标志将指示 LXTAL 时钟是否稳定。当 LXTAL 稳定时，如果中断寄存器 RCU_INT 中的相关中断使能位 LXTALSTBIE 被设置，可以产生一个中断。

通过在备份域控制寄存器(RCU_BDCTL)中设置 LXTALBPS 和 LXTALEN 位来选择外部时钟旁路模式。CK_LXTAL 等于驱动 OSC32IN 引脚的外部时钟。

内部 40K RC 振荡器(IRC40K)

内部 RC 振荡器的频率约为 40 kHz，是实时时钟电路或独立看门狗定时器的低功耗时钟源。IRC40K 提供了一种低成本的时钟源，因为不需要外部组件。IRC40K RC 振荡器可以通过使用复位源/时钟寄存器(RCU_RSTSCK)中的 IRC40KEN 位来开启或关闭。复位源/时钟寄存器

RCU_RSTSCK 中的 IRC40KSTB 标志将指示 IRC40K 时钟是否稳定。当 IRC40K 变得稳定时，如果设置了时钟中断寄存器(RCU_INT)中的相关中断使能位 IRC40KSTBIE，可以生成一个中断。

系统时钟(CK_SYS)选择

系统复位后，默认的 CK_SYS 源将是 IRC8M，可以通过改变时钟配置寄存器 0(RCU_CFG0)中的系统时钟切换位 SCS，切换到 HXTAL 或 CK_PLL0P。当 SCS 值改变时，CK_SYS 将继续使用原始时钟源，直到目标时钟源稳定。当一个时钟源直接或间接(通过 PLL)用作 CK_SYS 时，不可能停止它。

HXTAL 时钟监控(CKM)

HXTAL 时钟监控功能由控制寄存器(RCU_CTL)中的 HXTAL 时钟监控使能位 CKMEN 启用。该功能应在 HXTAL 启动延迟后启用，并在 HXTAL 停止时禁用。一旦检测到 HXTAL 故障，HXTAL 将自动被禁用。时钟中断寄存器(RCU_INT)中的 HXTAL 时钟卡住中断标志 CKMIF 将被设置，并生成 HXTAL 故障事件。此故障中断连接到 Cortex®-M33 的不可屏蔽中断(NMI)。如果 HXTAL 被选为 CK_SYS、PLL 和 CK_RTC 的时钟源，HXTAL 故障将强制 CK_SYS 源切换到 IRC8M，PLL 将自动禁用。如果 HXTAL 被选为 RTC 的时钟源，HXTAL 故障将重置 RTC 时钟选择。

时钟输出能力

可以通过时钟配置寄存器 0(RCU_CFG0)中的 CK_OUT 时钟源选择位 CKOUTSEL 选择多个时钟信号。相应的 GPIO 引脚应配置在正确的复用功能 I/O(AFIO)模式下，以输出所选的时钟信号。

表 6-1.时钟输出源选择

时钟输出源选择位	时钟源
0100	CK_SYS
0101	CK_IRC8M
0110	CK_HXTAL
0111	CK_PLL0/2
1000	CK_PLL1/2
1010	CK_HXTAL
1011	CK_LXTAL
1100	CK_IRC48M
1101	CK_IRC48M/8
1110	CK_IRC40K

电压控制

深度睡眠模式下 V_{CORE} 域电压可以通过深度睡眠模式电压寄存器(RCU_DSV)中的DSLPVS[2:0]位来控制。

时钟频率监控(CKFM)

时钟频率监控可以使用 IRC48M 来监控 IRC8M、HXTAL、PLL0P、PLL1 和 SYSCLK 的时钟频率范围。对于IRC8M和HXTAL,使用1000个IRC48M时钟周期作为监控窗口。对于PLL0P、PLL1 和 SYSCLK，使用 100 个 IRC48M 时钟周期作为监控窗口。用户可以通过配置RCU_CKFMCFGx(x = 0, 1, 2, 3, 4)寄存器来配置时钟频率的监控范围。如果相应的中断被启用，并且时钟频率故障标志被置位，将发生中断。

当 IRC48M 时钟被禁用或丢失时，时钟频率监控无效。

深度睡眠唤醒

深度睡眠唤醒提供两种模式：快速唤醒和慢速唤醒，可通过 RCU_DSV 寄存器中的DSLPWKMD 位选择。

慢速唤醒：当 DSLPVSMD = 1(用户配置的内核电压)时，DSLPVS[2:0]字段决定了深度睡眠期间的内核电压。如果在深度睡眠中将内核电压设置为 0.85V 或 0.9V，则在唤醒后需要额外时间让 LDO 升至 1.2V。在慢速唤醒中，首先使能 IRC40K；等待由 IRC40KCNT[4:0]定义的周期后，使能 IRC8M 并将系统时钟切换到 IRC8M。

快速唤醒：此模式仅支持深度睡眠内核电压为 1.2V。无需额外等待；IRC8M 立即被使能，系统时钟切换到 IRC8M。I2C/USART 唤醒仅支持快速唤醒模式。

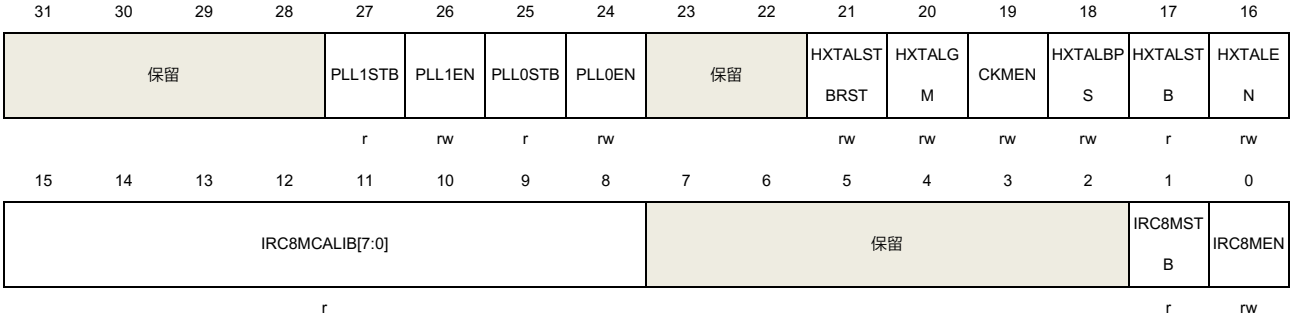
6.3. 寄存器定义

RCU 基地址：0x40021000

6.3.1. 控制寄存器(RCU_CTL)

地址偏移：0x00
复位值：0x0000xx03，其中 x 是未定义的。

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	PLL1STB	PLL1 时钟稳定标志 硬件设置，用于指示 PLL1 输出时钟是否稳定并准备就绪。 0：PLL1 不稳定 1：PLL1 稳定
26	PLL1EN	PLL1 使能 软件设置或复位，当 PLL 时钟用作系统时钟且 PLL1 用作 PLL 的输入源时，此位不能被复位。 进入深睡眠或待机模式时，由硬件复位。 0：PLL1 关闭 1：PLL1 开启
25	PLL0STB	PLL0 时钟稳定标志 硬件设置，用于指示 PLL0 输出时钟是否稳定并准备就绪。 0：PLL0 未稳定 1：PLL0 已稳定
24	PLL0EN	PLL0 使能 由软件设置和复位。如果 PLL0 时钟被用作系统时钟，则此位不能被复位。在进入深睡眠或待机模式时，由硬件复位。 0：PLL0 关闭 1：PLL0 开启

23:22	保留	必须保持复位值。
21	HXTALSTBRST	复位 HXTALSTB 位。 通过软件设置或复位，当 HXTAL 就绪不能清除为 0 时，使用软件控制。 0：无复位 1：强制复位
20	HXTALGM	HXTAL 增益控制 0：HXTAL 振荡频率为 4M - 10M 1：HXTAL 振荡频率为 10M - 32M
19	CKMEN	HXTAL 时钟监视使能 0：失能高速 4~32MHz 晶体振荡器(HXTAL)时钟监控 1：使能高速 4~32MHz 晶体振荡器(HXTAL)时钟监控 当硬件检测到 HXTAL 时钟卡在低电平或高电平时，内部硬件会将系统时钟切换为内部高速 IRC8M RC 时钟。恢复原始系统时钟的方法是：通过外部复位、上电复位或软件清除 CKMIF。
18	HXTALBPS	高速晶体振荡器(HXTAL)时钟旁路模式使能 只有当 HXTALEN 为 0 时，才能写入 HXTALBPS 位。 0：失能 HXTAL 旁路模式 1：使能 HXTAL 旁路模式，在此模式下 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 硬件设置，用于指示 HXTAL 振荡器是否稳定且准备就绪。 0：HXTAL 振荡器不稳定 1：HXTAL 振荡器稳定
16	HXTALEN	高速晶体振荡器(HXTAL)使能 由软件设置和复位。如果 HXTAL 时钟被用作系统时钟或 PLL0 输入时钟，并且当 PLL0 时钟被选择为系统时钟时，此位不能被复位。 进入深睡眠或待机模式时，由硬件复位。 0：高速 4~32MHz 晶体振荡器失能 1：高速 4~32 MHz 晶体振荡器使能
15:8	IRC8MCALIB[7:0]	内部 8MHz RC 振荡器校准值寄存器 这些位在上电时自动加载。
7:2	保留	必须保持复位值。
1	IRC8MSTB	IRC8M 内部 8MHz RC 振荡器稳定标志 硬件设置，用于指示 IRC8M 振荡器是否稳定并准备就绪。 0：IRC8M 振荡器不稳定 1：IRC8M 振荡器稳定
0	IRC8MEN	内部 8MHz RC 振荡器使能 由软件设置和复位。如果 IRC8M 时钟被用作系统时钟，则此位不能被复位。 当退出深睡眠或待机模式，或者在 CKMEN 设置时 HXTAL 时钟卡在低或高状态

时，由硬件设置。

0：内部 8 MHz RC 振荡器失能

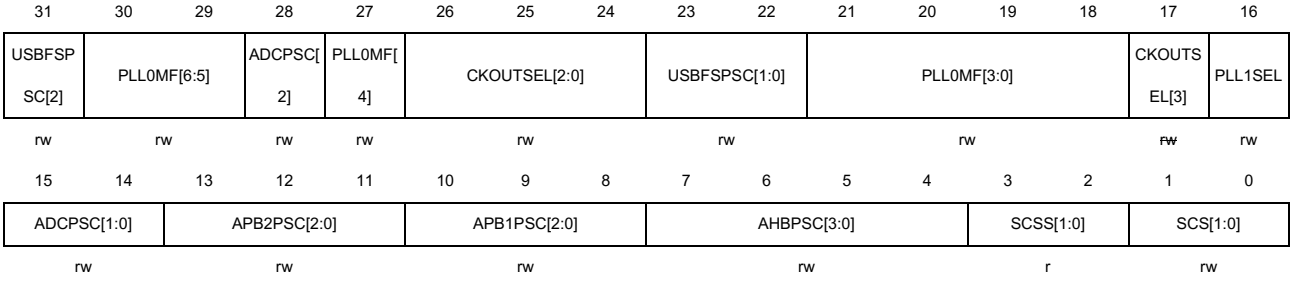
1：内部 8 MHz RC 振荡器使能

6.3.2. 时钟配置寄存器 0(RCU_CFG0)

地址偏移：0x04

复位值：0x4000 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	USBFSPSC[2]	USBFSPSC的第2位 参见 RCU_CFG0 的第 23 位和第 22 位
30:29	PLL0MF[6:5]	PLL0MF 的第 6 位和第 5 位 参见 RCU_CFG0 的位 21:18。
28	ADCPSC[2]	ADCPSC的第2位 参见 RCU_CFG0 的位 15:14。
27	PLL0MF[4]	PLL0MF 的第 4 位 参见 RCU_CFG0 的位 21:18。
26:24	CKOUTSEL[2:0]	CKOUT 时钟源选择 由软件设置和复位。 CKOUTSEL[3:0]: 00xx：无时钟信号 0100：CK_SYS 0101：CK_IRC8M 0110：CK_HXTAL 0111：CK_PLL0 / 2 1000：CK_PLL1 / 2 1010：CK_HXTAL 1011：CK_LXTAL 1100：CK_IRC48M 1101：CK_IRC48M / 8

		1110：CK_IRC40K
23:22	USBFSPSC[1:0]	USBFS 时钟分频选择 通过软件设置和复位来控制USBFS时钟分频值。USBFS时钟必须为48MHz。如果USBFS时钟已使能，则这些位不能被复位。 000：CK_USBFS = CK_PLL0 / 3 001：CK_USBFS = CK_PLL0 / 2 010：CK_USBFS = CK_PLL0 / 5 011：CK_USBFS = CK_PLL0 / 4 100：CK_USBFS = CK_PLL0 / 6 101：CK_USBFS = CK_PLL0 / 7 11x：CK_USBFS = CK_PLL0 / 8
21:18	PLL0MF[3:0]	PLL0 时钟倍频因子 RCU_CFG0 的位 30、位 29、位 27 由软件写入，用以定义 PLL0 的倍频因子。由软件设置和复位。 **注意：** PLL 输出时钟频率范围是 32~480MHz。 VCO 输出时钟频率 = PLL0 环路输入时钟频率 × PLLN(PLLN 是乘法因子，其值 = PLL0MF[6:0] + 1)，并且必须满足以下约束条件： 当使能展频时，必须满足 PLL0MF[6:0] >= 11(因为 MUL 配置的最小值是 4，PLL0MF 的最小值是 4。在展频算法中，PLL0MF[6:0]可能需要减少 7，因此最小值是 11)。 当使能展频时，必须满足 122 >= PLL0MF(因为 PLL0MF 配置值大于 122 可能会导致展频算法中的溢出问题，最大值是 122)。 PLL0MF[6:0]: 0x00：PLL0 源时钟 × 5 0x01：PLL0 源时钟 × 5 0x02：PLL0 源时钟 × 5 0x03：PLL0 源时钟 × 5 0x04：PLL0 源时钟 × 5 0x05：PLL0 源时钟 × 6 0x06：PLL0 源时钟乘以 7 0x07：PLL0 源时钟乘以 8 …… 0x7E：PLL0 源时钟乘以 127 0x7F：PLL0 源时钟乘以 128
17	CKOUTSEL[3]	CKOUTSEL 的第 3 位 参见 RCU_CFG0 的第 24 至 26 位
16	PLL1SEL	PLL1时钟源选择 通过软件设置和复位以控制PLL1时钟源。 00：选择HXTAL时钟作为PLL1的源时钟 01：选择IRC48M时钟作为PLL1的源时钟
15:14	ADCPSC[1:0]	ADC 时钟预分频选择

		这些位，RCU_CFG0 的第 28 位和 RCU_CFG1 的第 29 位，由软件写入以定义 ADC 预分频因子。 由软件设置和清除。 ADCPSC[3:0]: 0000：CK_ADC = CK_APB2 / 2 0001：CK_ADC = CK_APB2 / 4 0010：CK_ADC = CK_APB2 / 6 0011：CK_ADC = CK_APB2 / 8 0100：CK_ADC = CK_APB2 / 2 0101：CK_ADC = CK_APB2 / 12 0110：CK_ADC = CK_APB2 / 8 0111：CK_ADC = CK_APB2 / 16 1x00：CK_ADC = CK_AHB / 5 1x01：CK_ADC = CK_AHB / 6 1x10：CK_ADC = CK_AHB / 10 1x11：CK_ADC = CK_AHB / 20
13:11	APB2PSC[2:0]	APB2 分频器选择 通过软件设置和复位来控制 APB2 时钟分频比。 0xx：选择 CK_AHB 100：选择 (CK_AHB / 2) 101：选择 (CK_AHB / 4) 110：选择 (CK_AHB / 8) 111：选择 (CK_AHB / 16)
10:8	APB1PSC[2:0]	APB1 分频器选择 通过软件设置和复位来控制 APB1 时钟分频比。 0xx：选择 CK_AHB 100：选择 (CK_AHB / 2) 101：选择 (CK_AHB / 4) 110：选择 (CK_AHB / 8) 111：选择 (CK_AHB / 16)
7:4	AHBPSC[3:0]	AHB 时钟分频选择 通过软件设置和复位以控制 AHB 时钟分频比 0xxx：选择 CK_SYS 1000：选择 (CK_SYS / 2) 1001：选择 (CK_SYS / 4) 1010：选择 (CK_SYS / 8) 1011：选择 (CK_SYS / 16) 1100：选择 (CK_SYS / 64) 1101：选择 (CK_SYS / 128) 1110：选择 (CK_SYS / 256) 1111：选择 (CK_SYS / 512)

3:2	SCSS[1:0]	系统时钟切换状态 由硬件设置和复位，以指示系统时钟的时钟源。 00：选择 CK_IRC8M 作为系统时钟源(CK_SYS)。 01：选择 CK_HXTAL 作为系统时钟源(CK_SYS)。 10：选择 CK_PLL0P 作为系统时钟源(CK_SYS)。 11：保留
1:0	SCS[1:0]	系统时钟切换 由软件设置以选择 CK_SYS 的时钟源。由于 CK_SYS 的变化具有固有的延迟，软件应该读取 SCSS 来确认切换是否完成。当离开深睡眠和待机模式或在 HXTAL 被直接或间接选择作为 CK_SYS 的时钟源时，检测到 HXTAL 失效，切换将被强制到 IRC8M。 00：选择 CK_IRC8M 作为系统时钟源(CK_SYS)。 01：选择 CK_HXTAL 作为系统时钟源(CK_SYS)。 10：选择 CK_PLL0P 作为系统时钟源(CK_SYS)。 11：保留

6.3.3. 时钟中断寄存器(RCU_INT)

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CKMIC	LCKMIC	保留	PLL0 STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
								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	LCKMIE	PLL1 STBIE	PLL0 STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	LCKMIF	PLL1 STBIF	PLL0 STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL 时钟卡死中断清除 软件写 1 以复位 CKMIF 标志。 0：不复位 CKMIF 标志 1：复位 CKMIF 标志。
22	LCKMIC	LXTAL 时钟卡死中断清除。 软件写 1 以复位 LCKMIF 标志。 0：不复位 LCKMIF 标志。 1：复位 LCKMIF 标志。

21	保留	必须保持复位值。
20	PLL0STBIC	PLL0 稳定化中断清除。 软件写 1 以复位 PLL0STBIF 标志。 0：不复位 PLL0STBIF 标志。 1：复位 PLL0STBIF 标志。
19	HXTALSTBIC	外部高速晶振稳定中断清除。 软件写 1 以复位 HXTALSTBIF 标志。 0：不复位 HXTALSTBIF 标志。 1：复位 HXTALSTBIF 标志。
18	IRC8MSTBIC	内部 8MHz 振荡器稳定中断清除。 软件写 1 以复位 IRC8MSTBIF 标志。 0：不复位 IRC8MSTBIF 标志。 1：复位 IRC8MSTBIF 标志。
17	LXTALSTBIC	外部低速晶振稳定中断清除。 软件写 1 以复位 LXTALSTBIF 标志。 0：不复位 LXTALSTBIF 标志。 1：复位 LXTALSTBIF 标志。
16	IRC40KSTBIC	IRC40K 稳定化中断清除 通过软件写入 1 来复位 IRC40KSTBIF 标志。 0：不复位 IRC40KSTBIF 标志。 1：复位 IRC40KSTBIF 标志。
15	保留	必须保持复位值。
14	LCKMIE	LXTAL 时钟卡死中断使能。 通过软件设置和复位以使能/失能 LXTAL 时钟卡死中断。 0：失能 LXTAL 时钟卡死中断。 1：使能 LXTAL 时钟卡住中断
13	PLL1STBIE	PLL1 稳定化中断使能 通过软件设置和复位以使能/失能 PLL1 稳定化中断。 0：失能 PLL1 稳定化中断 1：使能 PLL1 稳定化中断
12	PLL0STBIE	PLL0 稳定化中断使能 通过软件设置和复位以使能/失能 PLL0 稳定化中断。 0：失能 PLL0 稳定化中断。 1：使能 PLL0 稳定化中断。
11	HXTALSTBIE	HXTAL 稳定化中断使能 通过软件设置和复位以使能/失能 HXTAL 稳定化中断。 0：失能 HXTAL 稳定化中断。 1：使能 HXTAL 稳定化中断。

10	IRC8MSTBIE	IRC8M 稳定化中断使能 通过软件设置和复位以使能/失能 IRC8M 稳定化中断。 0：失能 IRC8M 稳定化中断 1：使能 IRC8M 稳定化中断
9	LXTALSTBIE	LXTAL 稳定化中断使能 LXTAL 稳定化中断使能/失能控制 0：失能 LXTAL 稳定化中断 1：使能 LXTAL 稳定化中断
8	IRC40KSTBIE	IRC40K 稳定化中断使能 IRC40K 稳定化中断使能/失能控制 0：失能 IRC40K 稳定化中断 1：使能 IRC40K 稳定化中断
7	CKMIF	HXTAL 时钟卡住中断标志 当 HXTAL 时钟卡住时，由硬件设置。 通过软件设置 CKMIC 位时复位。 0：时钟正常工作 1：HXTAL 时钟卡住
6	LCKMIF	LXTAL 时钟卡住中断标志 当 LXTAL 时钟卡住时，由硬件设置。 通过软件设置 LCKMIC 位时复位。 0：时钟正常工作 1：LXTAL 时钟卡住
5	PLL1STBIF	PLL1 稳定化中断标志 当 PLL1 稳定且 PLL1STBIE 位被设置时，由硬件设置。 通过软件设置 PLL1STBIC 位时复位。 0：未产生 PLL1 稳定化中断 1：PLL1 稳定中断生成
4	PLL0STBIF	PLL0 稳定中断标志 当 PLL0 稳定且 PLL0STBIE 位被设置时，由硬件设置。 当软件设置 PLL0STBIC 位时复位。 0：未生成 PLL0 稳定中断 1：PLL0 稳定中断已生成
3	HXTALSTBIF	HXTAL 稳定中断标志 当高速 4~32MHz 晶体振荡器时钟稳定且 HXTALSTBIE 位被设置时，由硬件设置。 通过软件设置 HXTALSTBIC 位时复位。 0：未生成 HXTAL 稳定中断 1：生成 HXTAL 稳定中断
2	IRC8MSTBIF	IRC8M 稳定化中断标志 当内部 8MHz RC 振荡器时钟稳定且 IRC8MSTBIE 位被设置时，由硬件设置。

通过软件设置 IRC8MSTBIC 位时复位。
0：未生成 IRC8M 稳定中断
1：生成 IRC8M 稳定中断

1	LXTALSTBIF	LXTAL 稳定中断标志 当 LXTAL 稳定且 LXTALSTBIE 位被设置时，由硬件设置。 通过软件设置 LXTALSTBIC 位时复位。 0：未生成 LXTAL 稳定中断 1：生成 LXTAL 稳定中断
0	IRC40KSTBIF	IRC40K 稳定中断标志 当内部 40kHz RC 振荡器时钟稳定且 IRC40KSTBIE 位被设置时，由硬件设置。 通过软件设置 IRC40KSTBIC 位时复位。 0：未生成 IRC40K 稳定时钟就绪中断 1：生成 IRC40K 稳定中断

6.3.4. APB2 复位寄存器(RCU_APB2RST)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMPRST	保留	CAN2RST	USART5RST	CAN1RST	TIMER19RST	TIMER41RST	TIMER14RST	SYSCFGRST	CAN0RST	TIMER10RST	TIMER9RST	TIMER40RST	TRIGSELRST	SDIORST	SPI3RST
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
ADC2RST	USART0RST	TIMER7RST	SPI0RST	TIMER0RST	ADC1RST	ADC0RST	保留		PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST
rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31	CMPRST	比较器复位 此位由软件设置和复位。 0：无复位 1：复位比较器
30	保留	必须保持复位值。
29	CAN2RST	CAN2 复位 此位由软件设置和复位。 0：无复位 1：复位 CAN2
28	USART5RST	USART5 复位

此位由软件设置和复位。
0：无复位
1：复位 USART5

27 CAN1RST CAN1 复位
此位由软件设置和复位。
0：无复位
1：复位 CAN1

26 TIMER19RST TIMER19 复位
此位由软件设置和复位。
0：无复位
1：复位 TIMER19

25 TIMER41RST TIMER41 复位
此位由软件设置和复位。
0：无复位
1：复位 TIMER41

24 TIMER14RST TIMER14 复位
此位由软件设置和复位。
0：无复位
1：复位 TIMER14

23 SYSCFGRST SYSCFG 复位
此位由软件设置和复位。
0：无复位
1：复位 SYSCFG

22 CAN0RST CAN0 复位
此位由软件设置和复位。
0：无复位
1：复位 CAN0

21 TIMER10RST TIMER10 复位
此位由软件设置和复位。
0：无复位
1：复位 TIMER10

20 TIMER9RST TIMER9 复位
此位由软件设置和复位。
0：无复位
1：复位 TIMER9

19 TIMER40RST TIMER40 复位
此位由软件设置和复位。
0：无复位

		1：复位 TIMER40
18	TRIGSELRST	TRIGSEL 复位 此位由软件设置和复位。 0：无复位 1：复位 TRIGSEL
17	SDIORST	SDIO 复位 此位由软件设置和复位。 0：无复位 1：复位 SDIO
16	SPI3RST	SPI3 复位 此位由软件设置和复位。 0：无复位 1：复位 SPI3
15	ADC2RST	ADC2 复位 此位由软件设置和复位。 0：无复位 1：复位 ADC2
14	USART0RST	USART0 复位 此位由软件设置和复位。 0：无复位 1：复位 USART0
13	TIMER7RST	TIMER7 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER7
12	SPI0RST	SPI0 复位 此位由软件设置和复位。 0：无复位 1：复位 SPI0
11	TIMER0RST	TIMER0 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER0
10	ADC1RST	ADC1 复位 此位由软件设置和复位。 0：无复位 1：复位 ADC1
9	ADC0RST	ADC0 复位

		此位由软件设置和复位。 0：无复位 1：复位 ADC0
8:7	保留	必须保持复位值。
6	PERST	GPIOE 复位 此位由软件设置和复位。 0：无复位 1：复位 GPIOE
5	PDRST	复位 GPIOD 此位由软件设置和复位。 0：无复位 1：复位 GPIOD
4	PCRST	GPIOC 复位 此位由软件设置和复位。 0：无复位 1：复位 GPIOC
3	PBRST	GPIOB 复位 此位由软件设置和复位。 0：无复位 1：复位 GPIOB
2	PARST	GPIOA 复位 此位由软件设置和复位。 0：无复位 1：复位 GPIOA
1	保留	必须保持复位值。
0	AFRST	AFIO 复位 此位由软件设置和复位。 0：无复位 1：复位 AFIO

6.3.5. APB1 复位寄存器(RCU_APB1RST)

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DACRST	PMURST	保留			I2C2RST	保留	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	保留

		rw	rw				rw		rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	TIMER51RST	TIMER50RST	WWDGTRST	TIMER44RST	TIMER43RST	TIMER13RST	TIMER12RST	TIMER42RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DACRST	DAC 复位 此位由软件设置和复位。 0：无复位 1：复位 DAC
28	PMURST	PMU 复位 此位由软件设置和复位。 0：无复位 1：复位 PMU
27:25	保留	必须保持复位值。
24	I2C2RST	I2C2 复位 此位由软件设置和复位。 0：无复位 1：复位 I2C2
23	保留	必须保持复位值。
22	I2C1RST	I2C1 复位 此位由软件设置和复位。 0：无复位 1：复位 I2C1
21	I2C0RST	I2C0 复位 此位由软件设置和复位。 0：无复位 1：复位 I2C0
20	UART4RST	UART4 复位 此位由软件设置和复位。 0：无复位 1：复位 UART4
19	UART3RST	UART3 复位 此位由软件设置和复位。 0：无复位 1：复位 UART3

18	USART2RST	USART2 复位 此位由软件设置和复位。 0：无复位 1：复位 USART2
17	USART1RST	USART1 复位 此位由软件设置和复位。 0：无复位 1：复位 USART1
16	保留	必须保持复位值。
15	SPI2RST	SPI2 复位 此位由软件设置和复位。 0：无复位 1：复位 SPI2
14	SPI1RST	SPI1 复位 此位由软件设置和复位。 0：无复位 1：复位 SPI1
13	TIMER51RST	TIMER51 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER51
12	TIMER50RST	TIMER50 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER50
11	WWDGTRST	窗口看门狗复位 此位由软件设置和复位。 0：无复位 1：复位窗口看门狗定时器(WWDGT)。
10	TIMER44RST	定时器 44 复位。 此位由软件设置和复位。 0：无复位 1：复位定时器 44。
9	TIMER43RST	定时器 43 复位。 此位由软件设置和复位。 0：无复位 1：复位 TIMER43
8	TIMER13RST	TIMER13 复位

		此位由软件设置和复位。 0：无复位 1：复位 TIMER13
7	TIMER12RST	TIMER12 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER12
6	TIMER42RST	TIMER42 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER42
5	TIMER6RST	TIMER6 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER6
4	TIMER5RST	TIMER5 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER5
3	TIMER4RST	TIMER4 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER4
2	TIMER3RST	TIMER3 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER3
1	TIMER2RST	TIMER2 复位 此位由软件设置和复位。 0：无复位 1：复位定时器 TIMER2
0	TIMER1RST	定时器 TIMER1 复位 此位由软件设置和复位。 0：无复位 1：复位 TIMER1

6.3.6. AHB 使能寄存器(RCU_AHBEN)

地址偏移：0x14

复位值：0x0000 001C

注意：BIT4 不能清 0。

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNGEN	HAUEN	CAUEN	USBFSEN	DMAMUXEN	保留		EXMCEN	保留	CRCEN	保留		BKSRAMSPEN	SRAMSPEN	DMA1EN	DMA0EN
rw	rw	rw	rw	rw			rw		rw			rw	rw	rw	rw

位/位域	名称	描述
31	OSPIEN	OSPI 时钟使能 此位由软件设置和复位。 0：失能 OSPI 时钟 1：使能 OSPI 时钟
30:16	保留	必须保持复位值。
15	TRNGEN	TRNG 时钟使能 此位由软件设置和复位。 0：失能 TRNG 时钟 1：使能 TRNG 时钟
14	HAUEN	HAU 时钟使能 此位由软件设置和复位。 0：失能 HAU 时钟 1：使能 HAU 时钟
13	CAUEN	CAU 时钟使能 此位由软件设置和复位。 0：失能 CAU 时钟 1：使能 CAU 时钟
12	USBFSEN	USBFS 时钟使能 此位由软件设置和复位。 0：失能 USBFS 时钟 1：使能 USBFS 时钟
11	DMAMUXEN	DMAMUX 时钟使能 此位由软件设置和复位。 0：失能 DMAMUX 时钟 1：使能 DMAMUX 时钟
10:9	保留	必须保持复位值。

8	EXMCEN	EXMC 时钟使能 此位由软件设置和复位。 0：失能 EXMC 时钟 1：使能 EXMC 时钟
7	保留	必须保持复位值。
6	CRCEN	CRC 时钟使能 此位由软件设置和复位。 0：失能 CRC 时钟 1：使能 CRC 时钟
5:4	保留	必须保持复位值。
3	BKSRAMSPEN	睡眠模式下备份 SRAM 时钟使能 此位由软件设置和复位。 0：睡眠模式下失能备份 SRAM 时钟 1：睡眠模式下使能备份 SRAM 时钟
2	SRAMSPEN	睡眠模式下使能 SRAM 接口时钟 这个位由软件设置和复位，以在睡眠模式下使能/失能 SRAM 接口时钟。 0：睡眠模式下失能 SRAM 接口时钟。 1：睡眠模式下使能 SRAM 接口时钟。
1	DMA1EN	DMA1 时钟使能 此位由软件设置和复位。 0：失能 DMA1 时钟 1：使能 DMA1 时钟
0	DMA0EN	DMA0 时钟使能 此位由软件设置和复位。 0：失能 DMA0 时钟 1：使能 DMA0 时钟

6.3.7. APB2 使能寄存器(RCU_APB2EN)

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMPEN	保留	CAN2EN	USART5EN	CAN1EN	TIMER19EN	TIMER41EN	TIMER14EN	SYSCFGEN	CAN0EN	TIMER10EN	TIMER9EN	TIMER40EN	TRIGSELEN	SDIOEN	SPI3EN
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
ADC2EN	USART0EN	TIMER7EN	SPI0EN	TIMER0EN	ADC1EN	ADC0EN	保留		PEEN	PDEN	PCEN	PBEN	PAEN	保留	AFEN

rw rw rw rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31	CMPEN	CMP 时钟使能 此位由软件设置和复位。 0：失能 CMP 时钟 1：使能 CMP 时钟
30	保留	必须保持复位值。
29	CAN2EN	CAN2 时钟使能 此位由软件设置和复位。 0：失能 CAN2 时钟 1：使能 CAN2 时钟
28	USART5EN	USART5 时钟使能 此位由软件设置和复位。 0：失能 USART5 时钟 1：使能 USART5 时钟
27	CAN1EN	CAN1 时钟使能 此位由软件设置和复位。 0：失能 CAN1 时钟 1：使能 CAN1 时钟
26	TIMER19EN	TIMER19 时钟使能 此位由软件设置和复位。 0：失能 TIMER19 时钟 1：使能 TIMER19 时钟
25	TIMER41EN	TIMER41 时钟使能 此位由软件设置和复位。 0：失能 TIMER41 时钟 1：使能 TIMER41 时钟
24	TIMER14EN	TIMER14 时钟使能 此位由软件设置和复位。 0：失能 TIMER14 时钟 1：使能 TIMER14 时钟
23	SYSCFGEN	SYSCFG 时钟使能 此位由软件设置和复位。 0：失能 SYSCFG 时钟 1：使能 SYSCFG 时钟
22	CAN0EN	CAN0 时钟使能 此位由软件设置和复位。 0：失能 CAN0 时钟

		1：使能 CAN0 时钟
21	TIMER10EN	TIMER10 时钟使能 此位由软件设置和复位。 0：失能 TIMER10 时钟 1：使能 TIMER10 时钟
20	TIMER9EN	TIMER9 时钟使能 此位由软件设置和复位。 0：失能 TIMER9 时钟 1：使能 TIMER9 时钟
19	TIMER40EN	TIMER40 时钟使能 此位由软件设置和复位。 0：失能 TIMER40 时钟 1：使能 TIMER40 时钟
18	TRIGSELEN	TRIGSEL 时钟使能 此位由软件设置和复位。 0：失能 TRIGSEL 时钟 1：使能 TRIGSEL 时钟
17	SDIOEN	SDIO 时钟使能 此位由软件设置和复位。 0：失能 SDIO 时钟 1：使能 SDIO 时钟
16	SPI3EN	SPI3 时钟使能 此位由软件设置和复位。 0：失能 SPI3 时钟 1：使能 SPI3 时钟
15	ADC2EN	ADC2 时钟使能 此位由软件设置和复位。 0：失能 ADC2 时钟 1：使能 ADC2 时钟
14	USART0EN	USART0 时钟使能 此位由软件设置和复位。 0：失能 USART0 时钟 1：使能 USART0 时钟
13	TIMER7EN	TIMER7 时钟使能 此位由软件设置和复位。 0：失能 TIMER7 时钟 1：使能 TIMER7 时钟
12	SPI0EN	SPI0 时钟使能

		此位由软件设置和复位。 0：失能 SPI0 时钟 1：使能 SPI0 时钟
11	TIMER0EN	TIMER0 时钟使能 此位由软件设置和复位。 0：失能 TIMER0 时钟 1：使能 TIMER0 时钟
10	ADC1EN	ADC1 时钟使能 此位由软件设置和复位。 0：失能 ADC1 时钟 1：使能 ADC1 时钟
9	ADC0EN	ADC0 时钟使能 此位由软件设置和复位。 0：失能 ADC0 时钟 1：使能 ADC0 时钟
8:7	保留	必须保持复位值。
6	PEEN	GPIO 端口 E 时钟使能 此位由软件设置和复位。 0：失能 GPIO 端口 E 时钟 1：使能 GPIO 端口 E 时钟
5	PDEN	GPIO 端口 D 时钟使能 此位由软件设置和复位。 0：失能 GPIO 端口 D 时钟 1：使能 GPIO 端口 D 时钟
4	PCEN	GPIO 端口 C 时钟使能 此位由软件设置和复位。 0：失能 GPIO 端口 C 时钟 1：使能 GPIO 端口 C 时钟
3	PBEN	GPIO 端口 B 时钟使能 此位由软件设置和复位。 0：失能 GPIO 端口 B 时钟 1：使能 GPIO 端口 B 时钟
2	PAEN	GPIO 端口 A 时钟使能 此位由软件设置和复位。 0：失能 GPIO 端口 A 时钟 1：使能 GPIO 端口 A 时钟
1	保留	必须保持复位值。
0	AFEN	复用功能 I/O 时钟使能

此位由软件设置和复位。
0：失能复用功能 I/O 时钟
1：使能复用功能 I/O 时钟

6.3.8. APB1 使能寄存器(RCU_APB1EN)

地址偏移：0x1C
复位值：0x1000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DACEN	PMUEN	保留			I2C2EN	保留	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	保留
		rw	rw				rw		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2EN	SPI1EN	TIMER51EN	TIMER50EN	WWDGTEN	TIMER44EN	TIMER43EN	TIMER13EN	TIMER12EN	TIMER42EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DACEN	DAC 时钟使能 此位由软件设置和复位。 0：失能 DAC 时钟 1：使能 DAC 时钟
28	PMUEN	PMU 时钟使能 此位由软件设置和复位。 0：失能 PMU 时钟 1：使能 PMU 时钟
27:25	保留	必须保持复位值。
24	I2C2EN	I2C2 时钟使能 此位由软件设置和复位。 0：失能 I2C2 时钟 1：使能 I2C2 时钟
~~23~~	保留	必须保持复位值。
22	I2C1EN	I2C1 时钟使能 此位由软件设置和复位。 0：失能 I2C1 时钟 1：使能 I2C1 时钟

21	I2C0EN	I2C0 时钟使能 此位由软件设置和复位。 0：失能 I2C0 时钟 1：使能 I2C0 时钟
20	UART4EN	UART4 时钟使能 此位由软件设置和复位。 0：失能 UART4 时钟 1：使能 UART4 时钟
19	UART3EN	UART3 时钟使能 此位由软件设置和复位。 0：失能 UART3 时钟 1：使能 UART3 时钟
18	USART2EN	USART2 时钟使能 此位由软件设置和复位。 0：失能 USART2 时钟 1：使能 USART2 时钟
17	USART1EN	USART1 时钟使能 此位由软件设置和复位。 0：失能 USART1 时钟 1：使能 USART1 时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2 时钟使能 此位由软件设置和复位。 0：失能 SPI2 时钟 1：使能 SPI2 时钟
14	SPI1EN	SPI1 时钟使能 此位由软件设置和复位。 0：失能 SPI1 时钟 1：使能 SPI1 时钟
13	TIMER51EN	TIMER51 时钟使能 此位由软件设置和复位。 0：失能 TIMER51 时钟 1：使能 TIMER51 时钟
12	TIMER50EN	TIMER50 时钟使能 此位由软件设置和复位。 0：失能 TIMER50 时钟 1：使能 TIMER50 时钟
11	WWDGTEN	窗口看门狗时钟使能

此位由软件设置和复位。
0：失能窗口看门狗时钟
1：使能窗口看门狗时钟

10	TIMER44EN	TIMER44 时钟使能 此位由软件设置和复位。 0：失能 TIMER44 时钟 1：使能 TIMER44 时钟
9	TIMER43EN	TIMER43 时钟使能 此位由软件设置和复位。 0：失能 TIMER43 时钟 1：使能 TIMER43 时钟
8	TIMER13EN	TIMER13 时钟使能 此位由软件设置和复位。 0：失能 TIMER13 时钟 1：使能 TIMER13 时钟
7	TIMER12EN	TIMER12 时钟使能 此位由软件设置和复位。 0：失能 TIMER12 时钟 1：使能 TIMER12 时钟
6	TIMER42EN	TIMER42 时钟使能 此位由软件设置和复位。 0：失能 TIMER42 时钟 1：使能 TIMER42 时钟
5	TIMER6EN	TIMER6 时钟使能 此位由软件设置和复位。 0：失能 TIMER6 时钟 1：使能 TIMER6 时钟
4	TIMER5EN	TIMER5 时钟使能 此位由软件设置和复位。 0：失能 TIMER5 时钟 1：使能 TIMER5 时钟
3	TIMER4EN	TIMER4 时钟使能 此位由软件设置和复位。 0：失能 TIMER4 时钟 1：使能 TIMER4 时钟
2	TIMER3EN	TIMER3 时钟使能 此位由软件设置和复位。 0：失能 TIMER3 时钟

		1：使能 TIMER3 时钟
1	TIMER2EN	TIMER2 时钟使能 此位由软件设置和复位。 0：失能 TIMER2 时钟 1：使能 TIMER2 时钟
0	TIMER1EN	TIMER1 时钟使能 此位由软件设置和复位。 0：失能 TIMER1 时钟 1：使能 TIMER1 时钟

6.3.9. 备份域控制寄存器(RCU_BDCTL)

地址偏移：0x20
复位值：0x0000 0018

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)中的 LXTALEN、LXTALRDYRST、LXTALSTARTON、LCKMEN、LXTALDRI、LXTALBPS、RTCSRC 和 RTCEN 位只有在备份域复位后才会复位。这些位只有在电源控制寄存器(PMU_CTL)中的 BKPWEN 位被设置时才能被修改。

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			RTCHENF	LXTALSTARTON	RTCSRC[1:0]		LXTALRDYRST	LCKMD	LCKMEN	LXTALDRI[1:0]		LXTALBPS	LXTALSTB	LXTALEN
rw				r	rw	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 此位由软件设置和复位。 0：无复位 1：复位备份域
15	RTCEN	RTC 时钟使能 此位由软件设置和复位。 0：RTC 时钟失能 1：RTC 时钟使能
14:12	保留	必须保持复位值。
11	RTCHENF	RTC 高频时钟使能标志

		该位指示 RTC_HRCR 的 HRCE 位状态。 0：RTCH 时钟失能 1：RTCH 时钟使能
10	LXTALSTARTON	LXTAL 启动电路控制 0：当 LXTAL 就绪时关闭启动电路 1：当 LXTAL 就绪时保持启动电路
9:8	RTCSRC[1:0]	RTC 时钟输入选择 由软件设置和复位以控制 RTC 时钟源。一旦选择了 RTC 时钟源，除非备份域被复位，否则无法再更改。 00：未选择时钟 01：选择 CK_LXTAL 作为 RTC 源时钟 10：选择 CK_IRC40K 作为 RTC 源时钟 11：选择 (CK_HXTAL / 128) 作为 RTC 源时钟
7	LXTALRDYRST	LXTAL 就绪复位 0：LXTAL 就绪无复位 1：LXTAL 就绪复位
6	LCKMD	LXTAL 时钟失效检测 由硬件设置，用于指示时钟安全系统检测到故障时。 系统依赖外部 32 kHz 振荡器(LXTAL)。可以通过失能 LCKMEN 或失能 LXTALEN 来清除。 0：LXTAL(32 kHz 振荡器)未检测到故障。 1：检测到 LXTAL(32 kHz 振荡器)故障。
5	LCKMEN	LXTAL 时钟监视使能。 0：失能 LXTAL 时钟监视。 1：使能 LXTAL 时钟监视。 由软件设置以使能 LXTAL(32 kHz 振荡器)上的时钟安全系统。只有在 LXTAL 使能(LXTALEN 位使能)并且就绪(LXTALSTB 标志由硬件设置)时，才应使能 LCKMEN。 **注意：**一旦设置了 LCKMEN 位，这个位可以通过备份域复位或在检测到 LXTAL 时钟故障后(LCKMD = 1)复位这个位来重置。
4:3	LXTALDRI[1:0]	LXTAL 驱动能力。 由软件设置和复位。备份域复位会重置这个值。 00：较低驱动能力 01：中低驱动能力 10：中高驱动能力 11：较高驱动能力(复位值) **注意：**LXTALDRI 不在旁路模式。
2	LXTALBPS	使能 LXTAL 旁路模式 由软件设置和复位。 0：失能 LXTAL 旁路模式

		1：使能 LXTAL 旁路模式
1	LXTALSTB	低速晶振稳定标志 硬件设置，用于指示 LXTAL 输出时钟是否稳定且准备就绪。 0：LXTAL 不稳定 1：LXTAL 稳定
0	LXTALEN	使能 LXTAL 由软件设置和复位。 0：失能 LXTAL 1：使能 LXTAL

6.3.10. 复位源/时钟寄存器(RCU_RSTSCK)

地址偏移：0x24
复位值：0x0800 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
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														IRC40KSTB	IRC40KEN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志 当产生深睡眠/待机复位时，由硬件设置。 通过写入 1 到 RSTFC 位来复位。 0：未产生低功耗管理复位 1：产生低功耗管理复位
30	WWDGTRSTF	窗口看门狗定时器复位标志 当窗口看门狗定时器生成复位时，由硬件设置。 通过写入 1 到 RSTFC 位来复位。 0：未生成窗口看门狗复位 1：已生成窗口看门狗复位
29	FWDGTRSTF	独立看门狗定时器复位标志 由硬件设置，当产生一个独立看门狗定时器复位时。 通过写入 1 到 RSTFC 位来复位。 0：未产生独立看门狗定时器复位

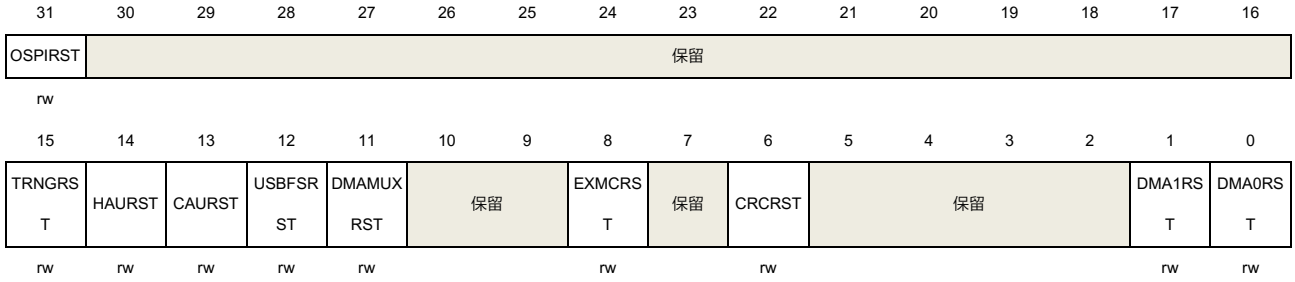
		1：产生独立看门狗定时器复位
28	SWRSTF	软件复位标志 当产生软件复位时，由硬件设置。 通过写入 1 到 RSTFC 位来复位。 0：未生成软件复位 1：已生成软件复位
27	PORRSTF	上电复位标志 由硬件在产生电源复位时设置。 通过写入 1 到 RSTFC 位来复位。 0：未产生电源复位 1：已产生电源复位
26	EPRSTF	外部引脚复位标志 当产生外部引脚复位时，由硬件设置。 通过写入 1 到 RSTFC 位来复位。 0：未产生外部引脚复位 1：已产生外部引脚复位
25	BORRSTF	BOR 复位标志 当产生 BOR 复位时，由硬件设置。 通过写入 1 到 RSTFC 位来复位。 0：未产生 BOR 复位 1：产生了 BOR 复位
24	RSTFC	复位标志清除 此位由软件设置以清除所有复位标志。 0：不清除复位标志 1：清除复位标志
23:2	保留	必须保持复位值。
1	IRC40KSTB	IRC40K 稳定标志 由硬件设置，以指示 IRC40K 输出时钟是否稳定且准备就绪。 0：IRC40K 不稳定 1：IRC40K 已稳定
0	IRC40KEN	IRC40K 使能 由软件设置和复位。 0：失能 IRC40K 1：使能 IRC40K

6.3.11. AHB 复位寄存器(RCU_AHBRST)

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。



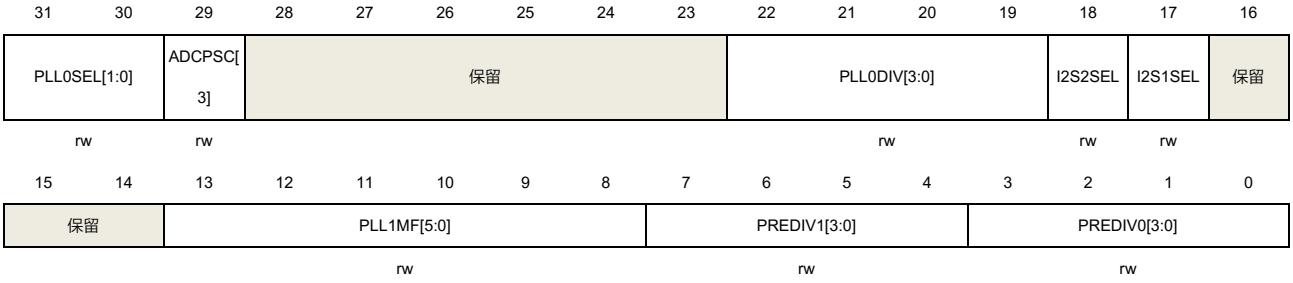
位/位域	名称	描述
31	OSPIRST	OSPI 复位 此位由软件设置和复位。 0：无复位 1：复位 OSPI
30:16	保留	必须保持复位值。
15	TRNGRST	TRNG 复位 此位由软件设置和复位。 0：无复位 1：复位 TRNG
14	HAURST	HAU 复位 此位由软件设置和复位。 0：无复位 1：复位 HAU
13	CAURST	复位 CAU 此位由软件设置和复位。 0：无复位 1：复位 CAU
12	USBFSRST	USBFS 复位 此位由软件设置和复位。 0：无复位 1：复位 USBFS
11	DMAMUXRST	DMAMUX 复位 此位由软件设置和复位。 0：无复位 1：复位 DMAMUX
10:9	保留	必须保持复位值。
8	EXMCRST	EXMC 复位 此位由软件设置和复位。 0：无复位

		1：复位 EXMC
7	保留	必须保持复位值。
6	CRCRST	CRC 复位 此位由软件设置和复位。 0：无复位 1：复位 CRC
5:2	保留	必须保持复位值。
1	DMA1RST	DMA0 复位 此位由软件设置和复位。 0：无复位 1：复位 DMA0
0	DMA0RST	DMA1 复位 此位由软件设置和复位。 0：无复位 1：复位 DMA1

6.3.12. 时钟配置寄存器 1(RCU_CFG1)

地址偏移：0x2C
复位值：0x0000 2000

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	PLL0SEL[1:0]	PLL0时钟源选择 由软件设置和复位以控制PLL0时钟源。 00：选择IRC8M时钟作为PLL0的源时钟 01：选择IRC48M时钟作为PLL0的源时钟 10：选择HXTAL时钟作为PLL0的源时钟 11：选择 CK_PLL1 时钟作为 PLL0 的源时钟
29	ADCPSC[3]	ADCPSC的第4位 参见 RCU_CFG0 的第 15 位和第 14 位以及 RCU_CFG0 的第 28 位。

28:23	保留	必须保持复位值。
22:19	PLL0DIV[3:0]	PLL0分频器 分频器是 PLL0DIV +1。
18	I2S2SEL	I2S2 时钟源选择 通过软件设置和复位来控制I2S2时钟源。 0：选择系统时钟作为I2S2源时钟。 1：选择 CK_PLL1 作为 I2S2 源时钟。
17	I2S1SEL	I2S1 时钟源选择。 通过软件设置和复位来控制I2S1时钟源。 0：选择系统时钟作为I2S1源时钟。 1：选择 CK_PLL1 作为 I2S1 源时钟。
16:14	保留	必须保持复位值。
13:8	PLL1MF[5:0]	PLL1 时钟倍频因子。 由软件设置和复位。 VCO 输出时钟经过倍频后的频率范围是 32~240MHz。 000000：PLL1 源时钟乘以 4。 000001：PLL1 源时钟乘以 4。 000010：PLL1 源时钟乘以 4。 000011：PLL1 源时钟乘以 4 000100：PLL1 源时钟乘以 4 000101：PLL1 源时钟乘以 5 000110：PLL1 源时钟乘以 6 ..... 111110：PLL1 源时钟乘以 62 111111：PLL1 源时钟乘以 63
7:4	PREDIV1[3:0]	PREDIV1 分频因子。 PLL1 环路输入时钟分频后的频率范围是 4~16MHz。 该位由软件设置和复位。当 PLL1 失能时，这些位可以被写入。 0000：PREDIV1 输入源时钟不进行分频 0001：PREDIV1 输入源时钟除以 2 0010：PREDIV1 输入源时钟除以 3 ..... 1110：PREDIV1 输入源时钟除以 15 1111：PREDIV1 输入源时钟除以 16
3:0	PREDIV0[3:0]	PREDIV0 分频因子。 PLL0 环路输入时钟分频后的频率范围是 4~16MHz。 这个位由软件设置和复位。当 PLL0 失能时，这些位可以被写入。 0000：PREDIV0 输入源时钟未分频。 0001：PREDIV0 输入源时钟分频 2 倍。 0010：PREDIV0 输入源时钟分频 3 倍。

.....

1110：PREDIV0 输入源时钟分频 15 倍。

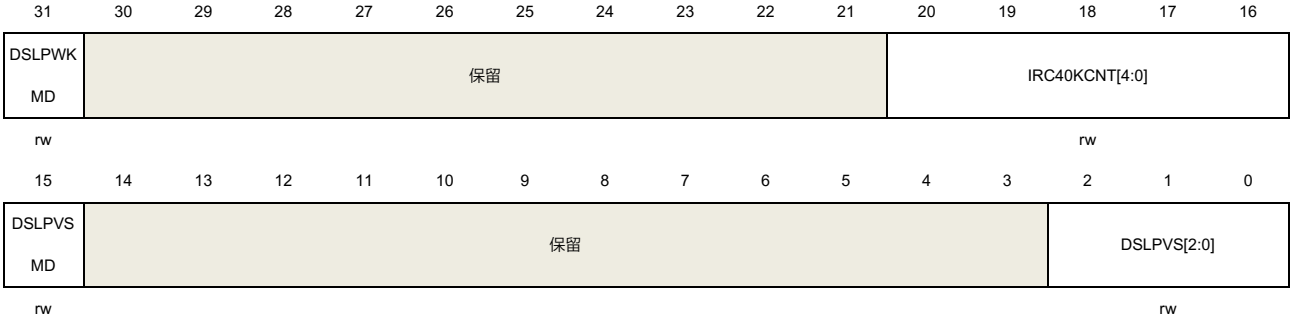
1111：PREDIV0 输入源时钟分频 16 倍。

6.3.13. 深睡眠模式电压寄存器(RCU_DSV)

地址偏移：0x34

复位值：0x8003 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	DSLPWKMD	深度睡眠唤醒方法。 0：慢速唤醒模式。选择此模式将等待(IRC40KCNT - 1)个 IRC40K 时钟周期，并支持在 DSLPVS 下从 0.9V 到 1.2V 的电压增强。 1：快速唤醒模式。此模式仅支持配置为 1.2V 内核电压的 DSLPVS 以实现快速唤醒。 I2C/USART 的唤醒方法仅支持快速唤醒模式。
30:21	保留	必须保持复位值。
20:16	IRC40KCNT[4:0]	深度睡眠唤醒后，如果启动等待时间小于 3，则强制配置为 3。启动时间计算为(IRC40KCNT - 1) * 25us，配置为 3 时等于 50us。
15	DSLPVSMD	深度睡眠模式电压选择模式 0：选择系统默认值 1：选择 DSLPVS[2:0]的值
14:3	保留	必须保持复位值。
2:0	DSLPVS[2:0]	深度睡眠模式电压选择 这些位由软件设置和清除。 000：深度睡眠模式下内核电压为 1.2V 001：深度睡眠模式下内核电压为 1.15V 010：深度睡眠模式下内核电压为 1.1V 011：深度睡眠模式下内核电压为 1.05V 100：深睡眠模式下内核电压为 1.00V

101：深睡眠模式下内核电压为 0.95V
110：深睡眠模式下内核电压为 0.9V
111：深睡眠模式下内核电压为 0.85V

6.3.14. 时钟频率监测配置寄存器 0(RCU_CKFMCFG0)

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)和字(32 位)访问。



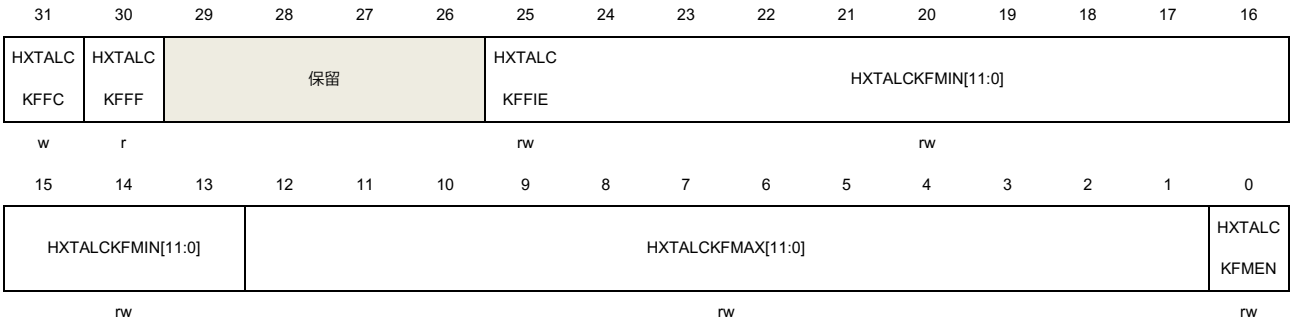
位/位域	名称	描述
31:6	保留	必须保持复位值。
5	IRC8MCKFFIE	IRC8M时钟频率失效中断使能 此位由软件设置和复位。 0：失能IRC8M时钟频率失效中断 1：使能 IRC8M 时钟频率失效中断
4	IRC8MCKFFC	IRC8M时钟频率失效标志清除 软件写1以复位IRC8MCKFFF标志，硬件在IRC8MCKFFF清除后自动复位该位。 0：不复位IRC8MCKFFF标志 1：复位IRC8MCKFFF标志
3	IRC8MCKFFF	IRC8M时钟频率失效标志 该位指示IRC8M时钟频率是否超出监控范围。 0：IRC8M时钟频率未超出监控范围 1：IRC8M 时钟频率超出监控范围
2:1	IRC8MCKFMC[1:0]	IRC8M 时钟频率监测配置 此位由软件设置和复位。 00：时钟频率监测范围是 8MHz +/- 5% 01：时钟频率监测范围是 8MHz +/- 10% 10：时钟频率监测范围是 8MHz +/- 15% 11：时钟频率监测范围是 8MHz +/- 20%
0	IRC8MCKFMEN	IRC8M 时钟频率监测使能 此位由软件设置和复位。

0：失能内部 8M RC 振荡器(IRC8M)的时钟频率监测
1：使能内部 8M RC 振荡器(IRC8M)的时钟频率监测

6.3.15. 时钟频率监测配置寄存器 1(RCU_CKFMCFG1)

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)和字(32 位)访问。



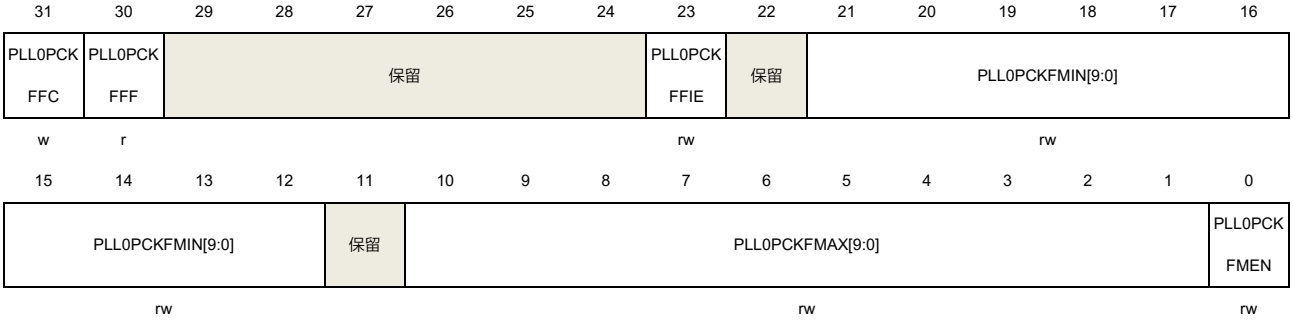
位/位域	名称	描述
31	HXTALCKFFC	HXTAL 时钟频率失效标志清除 软件写入1以复位 HXTALCKFFF 标志，硬件在 HXTALCKFFF 清除后自动复位此位。 0：不复位 HXTALCKFFF 标志 1：复位 HXTALCKFFF 标志
30	HXTALCKFFF	HXTAL时钟频率失效标志 该位表示HXTAL时钟频率是否超出监控范围。 0：HXTAL时钟频率不在监控范围之外 1：HXTAL时钟频率超出监控范围
29:26	保留	必须保持复位值。
25	HXTALCKFFIE	HXTAL时钟频率失效中断使能 此位由软件设置和复位。 0：失能HXTAL时钟频率失效中断 1：使能 HXTAL 时钟频率失效中断
24:13	HXTALCKFMIN[11:0]	HXTAL时钟频率监测配置最小值 此位由软件设置和复位。 参考时钟为IRC48M，监控窗口为1000个IRC48M时钟周期。
12:1	HXTALCKFMAX[11:0]	HXTAL时钟频率监测配置最大值 此位由软件设置和复位。 参考时钟为IRC48M，监控窗口为1000个IRC48M时钟周期。
0	HXTALCKFMEN	HXTAL时钟频率监测使能

此位由软件设置和复位。
0：失能HXTAL时钟频率监测
1：使能HXTAL时钟频率监测

6.3.16. 时钟频率监测配置寄存器 2(RCU_CKFMCFG2)

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)和字(32 位)访问。



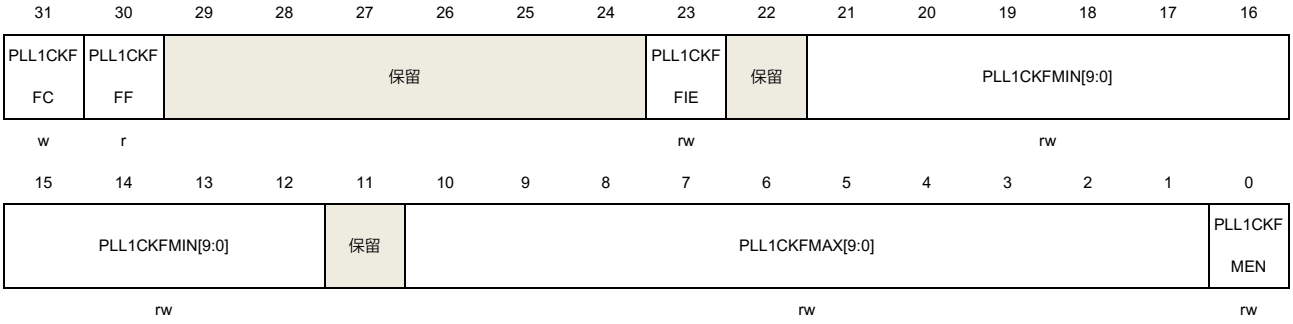
位/位域	名称	描述
31	PLL0PCKFFC	PLL0P时钟频率失败标志清除 软件写入1以复位PLL0PCKFFF标志，且在PLL0PCKFFF被清除后，该位由硬件自动复位。 0：未复位PLL0PCKFFF标志 1：复位PLL0PCKFFF标志
30	PLL0PCKFFF	PLL0P时钟频率失效标志 该位表示PLL0P时钟频率是否超出监测范围。 0：PLL0P时钟频率未超出监测范围。 1：PLL0P时钟频率超出监测范围。
29:24	保留	必须保持复位值。
23	PLL0PCKFFIE	PLL0P时钟频率失效中断使能。 此位由软件设置和复位。 0：失能PLL0P时钟频率失效中断。 1：使能 PLL0P 时钟频率失效中断。
22	保留	必须保持复位值。
21:12	PLL0PCKFMIN[9:0]	PLL0P时钟频率监测配置最小值。 此位由软件设置和复位。 参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
11	保留	必须保持复位值。

10:1	PLL0PCKFMAX[9:0]	PLL0P时钟频率监测配置最大值 此位由软件设置和复位。 参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
0	PLL0PCKFMEN	PLL0P时钟频率监测使能 此位由软件设置和复位。 0：失能PLL0P时钟频率监测 1：使能PLL0P时钟频率监测

6.3.17. 时钟频率监测配置寄存器 3(RCU_CKFMCFG3)

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)和字(32 位)访问。



位/位域	名称	描述
31	PLL1CKFFC	PLL1时钟频率故障标志清除 软件写1以复位PLL1CKFFF标志，当PLL1CKFFF被清除后，该位由硬件自动复位。 0：未复位PLL1CKFFF标志 1：复位PLL1CKFFF标志
30	PLL1CKFFF	PLL1时钟频率故障标志 该位指示PLL1时钟频率是否超出监控范围。 0：PLL1时钟频率未超出监测范围。 1：PLL1时钟频率超出监测范围。
29:24	保留	必须保持复位值。
23	PLL1CKFFIE	PLL1时钟频率故障中断使能。 此位由软件设置和复位。 0：失能PLL1时钟频率故障中断。 1：使能 PLL1 时钟频率故障中断。
22	保留	必须保持复位值。
21:12	PLL1CKFMIN[9:0]	PLL1时钟频率监测配置最小值。 此位由软件设置和复位。

		参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
11	保留	必须保持复位值。
10:1	PLL1CKFMAX[9:0]	PLL1时钟频率监测配置最大值 此位由软件设置和复位。 参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
0	PLL1CKFMEN	PLL1时钟频率监测使能 此位由软件设置和复位。 0：失能PLL1时钟频率监测 1：使能PLL1时钟频率监测

6.3.18. 时钟频率监测配置寄存器 4(RCU_CKFMCFG4)

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)和字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYSCKFFC	SYSCKFFF	保留						SYSCKFFIE	保留	SYSCKFMIN[9:0]					
w	r							rw		rw					

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

位/位域	名称	描述
31	SYSCKFFC	系统时钟频率失效标志清除 软件写1复位SYSCKFFF标志，当SYSCKFFF清除后，该位由硬件自动复位。 0：不复位SYSCKFFF标志 1：复位SYSCKFFF标志
30	SYSCKFFF	系统时钟频率失效标志 该位指示PLL1时钟频率是否超出监控范围。 0：系统时钟频率未超出监控范围。 1：系统时钟频率超出监控范围。
29:24	保留	必须保持复位值。
23	SYSCKFFIE	系统时钟频率失效中断使能。 此位由软件设置和复位。 0：失能系统时钟频率失效中断。 1：使能系统时钟频率失效中断。

22	保留	必须保持复位值。
21:12	SYSCKFMIN[9:0]	系统时钟频率监测配置最小值。 此位由软件设置和复位。 参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
11	保留	必须保持复位值。
10:1	SYSCKFMAX[9:0]	系统时钟频率监测配置最大值。 此位由软件设置和复位。 参考时钟是IRC48M，监测窗口是100个IRC48M时钟周期。
0	SYSCKFMEN	系统时钟频率监测使能 此位由软件设置和复位。 0：失能SYSCLK时钟频率监视器 1：使能SYSCLK时钟频率监视器

6.3.19. 附加时钟控制寄存器(RCU_ADDCTL)

地址偏移：0xC0
复位值：0x8000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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														PLLUSBSEL	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 这些位在上电时自动加载。
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 硬件设置，以指示 IRC48M 振荡器是否稳定且准备就绪。 0：IRC48M 不稳定 1：IRC48M 稳定
16	IRC48MEN	内部 48MHz RC 振荡器使能 由软件设置和复位。进入深睡眠或待机模式时由硬件复位。 0：IRC48M 失能

		1：IRC48M 使能
15:2	保留	必须保持复位值。
1	PLLUSBSEL	PLLUSB 时钟源选择 0：CK_PLL0 1：CK_PLL1
0	CK48MSEL	48MHz 时钟选择 由软件设置和复位。此位用于生成 CK48M 时钟，选择 IRC48M 时钟或 PLL48M 时钟。 0：不选择 IRC48M 时钟(使用 CK_PLLUSB 时钟除以 USBFSPSC) 1：选择 IRC48M 时钟

6.3.20. 附加时钟中断寄存器(RCU_ADDINT)

地址偏移：0xCC
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	IRC48MSTBIC	内部 48 MHz RC 振荡器稳定中断清除 软件写 1 以复位 IRC48MSTBIF 标志。 0：不复位 IRC48MSTBIF 标志。 1：复位 IRC48MSTBIF 标志。
21:15	保留	必须保持复位值。
14	IRC48MSTBIE	内部 48 MHz RC 振荡器稳定中断使能。 软件设置和复位以使能/失能 IRC48M 稳定中断。 0：失能 IRC48M 稳定中断。 1：使能 IRC48M 稳定中断。
13:7	保留	必须保持复位值。
6	IRC48MSTBIF	IRC48M 稳定中断标志。

		当内部 48 MHz RC 振荡器时钟稳定且 IRC48MSTBIE 位被设置时，由硬件设置。 当设置 IRC48MSTBIC 位时，由软件复位。 0：未生成 IRC48M 稳定中断。 1：生成 IRC48M 稳定中断。
5:0	保留	必须保持复位值。

6.3.21. PLL 时钟展频控制寄存器(RCU_PLLSSCTL)

地址偏移：0xD0
复位值：0x0000 0000

展频调制仅适用于主 PLL0 时钟。

RCU_PLLSSCTL 寄存器必须在主 PLL0 失能时写入。

该寄存器用于根据以下公式配置 PLL0 展频时钟生成：

MODCNT = round(f_{PLL0IN}/4/fmod)

MODSTEP = round(mdamp*PLL0N*2^{14}/(MODCNT*100))

其中 fPLL0IN 表示 PLL0 输入时钟频率，fmod 表示展频调制频率，mdamp 表示以百分比表示的展频调制幅度，PLL0N 表示 PLL0 时钟频率乘法因子。

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31	SSCGON	PLL0 展频调制使能 0：展频调制失能 1：展频调制使能
30	SS_TYPE	PLL0 展频调制类型选择 0：中心展频选择 1：下降展频选择
29:28	保留	必须保持复位值。
27:13	MODSTEP[14:0]	这些位用于配置 PLL0 展频调制的幅度和频率。必须满足以下条件： MODSTEP*MODCNT <= 2^15 - 1。
12:0	MODCNT[12:0]	这些位用于配置 PLL0 展频调制的幅度和频率。必须满足以下条件：

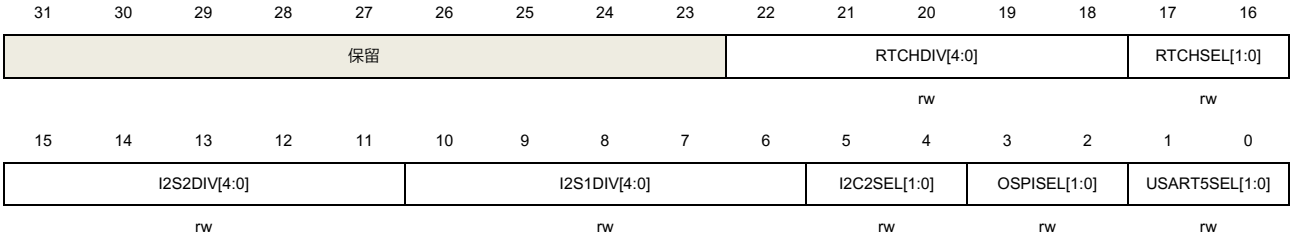
MODSTEP*MODCNT <= 2^15 - 1。

6.3.22. 时钟配置寄存器 2(RCU_CFG2)

地址偏移：0XD4

复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22:18	RTCHDIV[4:0]	RTCH分频器 分频值是RTCHDIV + 1。
17:16	RTCHSEL[1:0]	RTCH 时钟源选择 通过软件设置和复位以控制 RTCH 时钟源。 00：选择 CK_IRC8M 作为 RTCH 源时钟 01：选择 CK_PLL0 作为 RTCH 源时钟 10：选择 CK_PLL1 作为 RTCH 源时钟 11：选择CK_IRC48M作为RTCH源时钟
15:11	I2S2DIV[4:0]	I2S2DIV分频器 分频器是I2S2DIV +1。
10:6	I2S1DIV[4:0]	I2S1DIV分频器 分频器是I2S1DIV +1。
5:4	I2C2SEL[1:0]	I2C2 时钟源选择 通过软件设置和复位来控制 I2C2 时钟源。 00：选择 APB1 时钟作为 I2C2 源时钟 01：选择系统时钟作为 I2C2 源时钟 1x：选择CK_IRC8M时钟作为I2C2源时钟
3:2	OSPISEL[1:0]	OSPI 时钟源选择 通过软件设置和复位来控制 OSPI 时钟源。 00：选择 HCLK 作为 OSPI 源时钟 01：选择 CK_PLL0 作为 OSPI 源时钟 10：选择 CK_PLL1 作为 OSPI 源时钟

		11：选择CK_IRC8M作为OSPI源时钟
1:0	USART5SEL[1:0]	USART5 时钟源选择 通过软件设置和复位以控制 USART5 的时钟源。 00：选择 CK_APB2 作为 USART5 的源时钟。 01：选择 CK_SYS 作为 USART5 的源时钟。 10：选择 CK_LXTAL 作为 USART5 的源时钟。 11：选择CK_IRC8M作为USART5的源时钟。

6.3.23. APB1 附加复位寄存器(RCU_ADDAPB1RST)

地址偏移：0xE0
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			CTC1RST	CTC0RST	保留										
			rw	rw											

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	CTC1RST	CTC1 复位 此位由软件设置和复位。 0：无复位 1：复位 CTC1
27	CTC0RST	CTC0 复位 此位由软件设置和复位。 0：无复位 1：复位 CTC0
26:0	保留	必须保持复位值。

6.3.24. APB1 附加使能寄存器(RCU_ADDAPB1EN)

地址偏移：0xE4
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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

保留			CTC1EN	CTC0EN	保留										
			rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	CTC1EN	CTC1 时钟使能 此位由软件设置和复位。 0：CTC1 时钟失能 1：使能 CTC1 时钟
27	CTC0EN	CTC0 时钟使能 此位由软件设置和复位。 0：CTC0 时钟失能 1：使能 CTC0 时钟
26:0	保留	必须保持复位值。

6.3.25. 锁定寄存器(RCU_LOCK)

地址偏移：0x100
复位值：0x0000 0000

该寄存器可以按字节(8 位)、半字(16 位)或字(32 位)访问。

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

位/位域	名称	描述
31	LOCK	RCU 寄存器锁定位 此位由软件设置和复位。 0：RCU 寄存器可以被配置 1：RCU 寄存器不能被配置但可以读取 **注意：**当 LOCK = 1 且写入 RCU 寄存器时，将产生 HardFault 异常。
30:0	保留	必须保持复位值。

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/8MHz RC 晶振(IRC48M / IRC8M)。CTC 模块基于外部高精度的参考信号源来校准 IRC48M / IRC8M 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/ IRC8M 时钟。

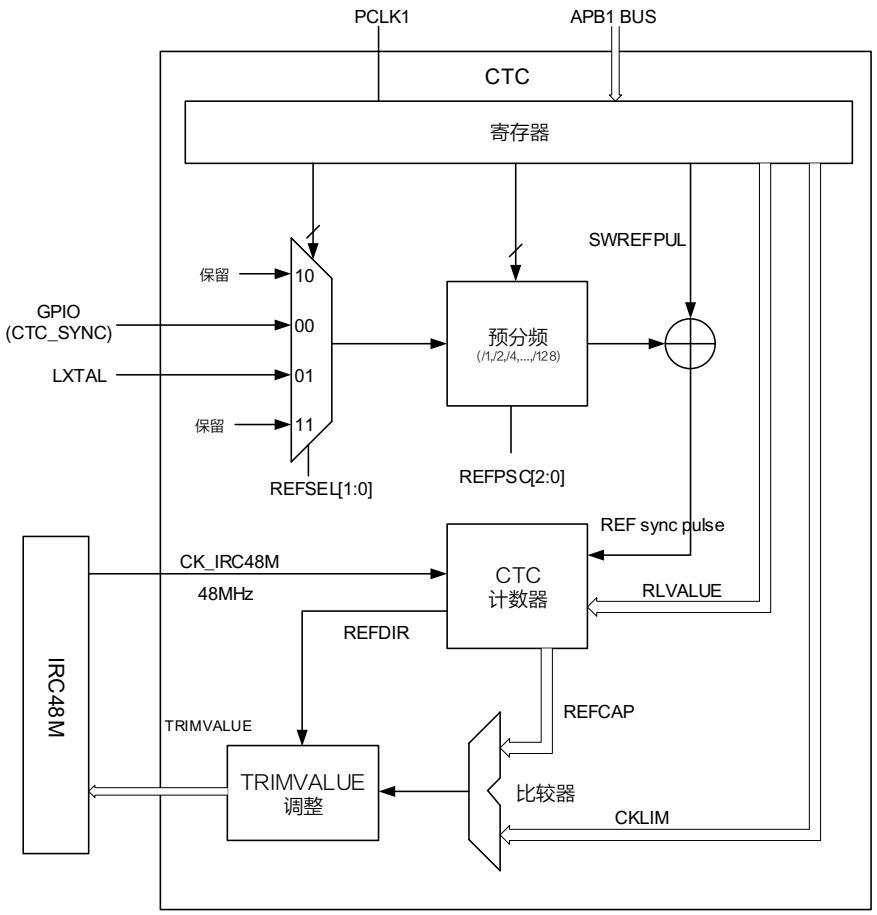
7.2. 主要特征

- 两个外部参考信号源：GPIO(CTC_SYNC)，LXTAL时钟；
- 提供软件参考同步脉冲；
- 硬件自动校准，无需软件操作；
- 具有参考信号源捕获和重载功能的16 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. 功能说明

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)或者 LXTAL 时钟输出。

然后，可以通过设置 CTC_CTL1 寄存器中的 REFPOL 位来配置参考信号源同步时的信号极性，通过设置 CTC_CTL1 寄存器中的 REFPSC 位来产生一个合适的同步时钟频率信号。

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

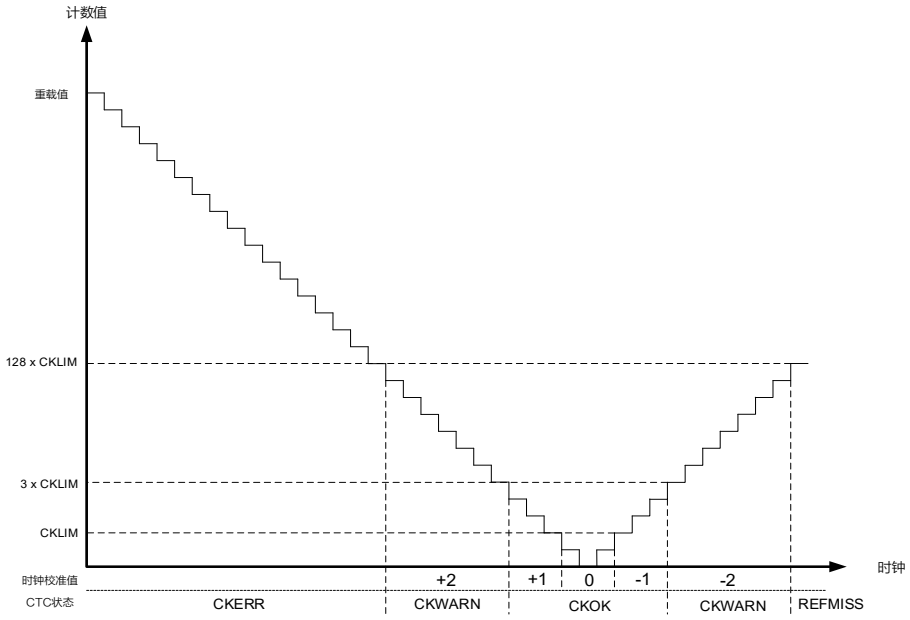
注意：参考信号脉冲宽度必须大于被校准时钟的宽度的 1.5 倍。

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/8M)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，IRC8M：8 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：

$$RLVALUE=(F_{clock}/ F_{REF})-1 \quad (7\text{-}1)$$

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：

$$CKLIM=(F_{clock}/ F_{REF}) \times 0.13\% /2 \quad (7\text{-}2)$$

$$CKLIM=(F_{clock}/ F_{REF}) \times 0.2\% /2 \quad (7\text{-}3)$$

典型的步长值是 0.13%(IRC48M))and 0.2%(IRC8M)，F_{clock}是期望时钟的频率(IRC48M/8M)，F_{REF}是 REF 同步脉冲信号的频率。

7.4. CTC 寄存器

CTC0 基地址：0x4000 C800

CTC1 基地址：0x4000 CC00

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。

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

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

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	TRIMVALUE[5:0]	IRC48M / IRC8M 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。 IRC48M 的调整单位为 0.13%。IRC8M 的调整单位为 0.2%。 TRIMVALUE 的中间值是 32，当 TRIMVALUE 值加 1 时，IRC48M / IRC8M 时钟频率增加大约 0.13%/0.2%。当 TRIMVALUE 值减 1 时，IRC48M / IRC8M 时钟频率的减少大约 0.13%/0.2%。
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 / IRC8M 的时钟频率达到 48MHz/8MHz。 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

CTC0 复位值：0x2022 BB7F

CTC1 复位值：0x2009 1F3F

该寄存器只能按字(32 位)访问。

注意：当 CNTEN 为 1 时，不能修改该寄存器的值。

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

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

位/位域	名称	描述
31	REFPOL	参考信号源极性 该位由软件置位或清除，用于选择参考信号源的同步极性

		0：选择上升沿 1：选择下降沿
30	保留	必须保持复位值。
29:28	REFSEL[1:0]	参考信号源选择 该位由软件置位或清除，用于选择参考信号源 00：选择 GPIO(CTC_SYNC)输入信号 01：选择 LXTAL 时钟 10：保留 11：保留
27	保留	必须保持复位值。
26:24	REFPSC[2:0]	参考信号源预分频 该位由软件置位或清除 000：参考信号不分频 001：参考信号 2 分频 010：参考信号 4 分频 011：参考信号 8 分频 100：参考信号 16 分频 101：参考信号 32 分频 110：参考信号 64 分频 111：参考信号 128 分频
23:16	CKLIM[7:0]	时钟校准时基限值 该位由软件置位或清除，用于定义时钟校准时基限值。该位用于频率评估和自动校准过程，详细情况请参考*频率评估和自动校准过程*。
15:0	RLVALUE[15:0]	CTC 计数器重载值 该位由软件置位或清除，用于定义 CTC 计数器的重载值，当检测到一个同步参考脉冲时，该值将重载到 CTC 校准计数器中。

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。

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

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

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	EREFIC	EREFIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 EREFIF 位，写 0 没影响。

2	ERRIC	ERRIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 ERRIF 位，TRIMERR 位，REFMISS 位和 CKERR 位，写 0 没影响。
1	CKWARNIC	CKWARNIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKWARNIF 位，写 0 没影响。
0	CKOKIC	CKOKIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKOKIF 位，写 0 没影响。

8. 中断/事件控制器(EXTI)

8.1. 简介

Cortex®-M33 集成了嵌套式矢量型中断控制器(Nested Vectored Interrupt Controller(NVIC))来实现高效的异常和中断处理。NVIC 实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于 NVIC 的说明请参考《 Cortex®-M33 技术参考手册 》。

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

8.2. 主要特征

- Cortex®-M33 系统异常；
- 多达 115 种可屏蔽的外设中断；
- 4 位中断优先级配置位，可配置 16 个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI 中有多达 25 个相互独立的边沿检测电路；
- 3 种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

8.3. 中断功能描述

Arm Cortex®-M33 处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序(ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。*表 8-1. Cortex®-M33 中的 NVIC 异常类型*和*表 8-2. 中断向量表*列出了 Cortex®-M33 中的 NVIC 异常类型。

表 8-1. Cortex®-M33 中的 NVIC 异常类型

异常类型	向量编号	优先级(a)	向量地址	描述
-	0	-	0x0000_0000	保留
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断

异常类型	向量编号	优先级(a)	向量地址	描述
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障
存储器管理	4	可编程设置	0x0000_0010	存储器管理
总线故障	5	可编程设置	0x0000_0014	预取指故障，存储器访问故障
用法故障	6	可编程设置	0x0000_0018	未定义的指令或非法状态
-	7-10	-	0x0000_001C - 0x0000_002B	保留
SVCall 服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统服务调用
调试监控	12	可编程设置	0x0000_0030	调试监视器
-	13	-	0x0000_0034	保留
PendSV 挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
SysTick	15	可编程设置	0x0000_003C	系统节拍定时器

表 8-2. 中断向量表

中断编号	向量编号	中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	来自 EXTI16 的 VAUVD/VAOVD/LVD/OVD 检测中断	0x0000_0044
IRQ 2	18	来自 EXTI24 的 RTC 侵入/时间戳中断	0x0000_0048
IRQ 3	19	来自 EXTI19 的 RTC 唤醒中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 与 CTC 中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	0x0000_0060
IRQ 9	25	EXTI 线 3 中断	0x0000_0064
IRQ 10	26	EXTI 线 4 中断	0x0000_0068
IRQ 11	27	DMA0 通道 0 全局中断	0x0000_006C
IRQ 12	28	DMA0 通道 1 全局中断	0x0000_0070
IRQ 13	29	DMA0 通道 2 全局中断	0x0000_0074
IRQ 14	30	DMA0 通道 3 全局中断	0x0000_0078
IRQ 15	31	DMA0 通道 4 全局中断	0x0000_007C
IRQ 16	32	DMA0 通道 5 全局中断	0x0000_0080
IRQ 17	33	DMA0 通道 6 全局中断	0x0000_0084
IRQ 18	34	ADC0 和 ADC1 全局中断	0x0000_0088
IRQ 19	35	CAN0 发送中断	0x0000_008C
IRQ 20	36	CAN0 接收 0 中断	0x0000_0090
IRQ 21	37	CAN0 接收 1 中断	0x0000_0094
IRQ 22	38	CAN0 EWMC 中断	0x0000_0098
IRQ 23	39	EXTI[9:5]中断	0x0000_009C

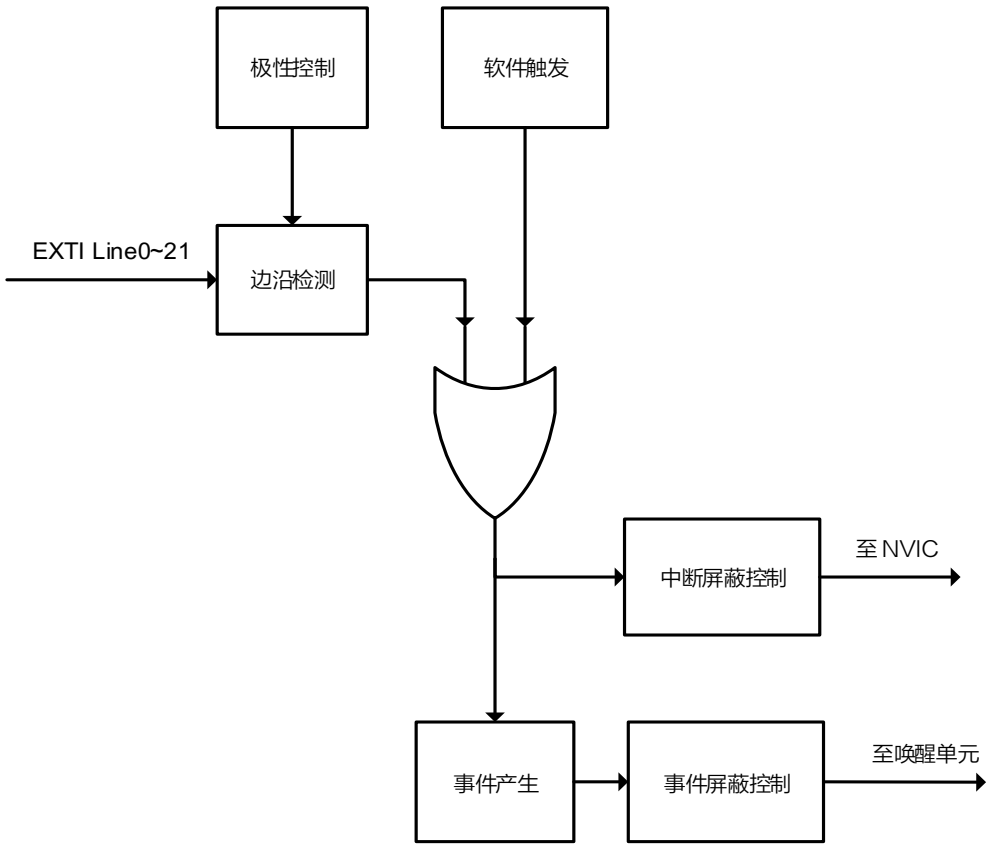
中断编号	向量编号	中断描述	向量地址
IRQ 24	40	TIMER0 中止中断和 TIMER14 全局中断	0x0000_00A0
IRQ 25	41	TIMER0 更新中断和 TIMER9 全局中断	0x0000_00A4
IRQ 26	42	TIMER0 触发与通道换相中断和 TIMER10 全局中断	0x0000_00A8
IRQ 27	43	TIMER0 通道捕获/比较中断	0x0000_00AC
IRQ 28	44	TIMER1 全局中断	0x0000_00B0
IRQ 29	45	TIMER2 全局中断	0x0000_00B4
IRQ 30	46	TIMER3 全局中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 或 I2S1ADD 全局中断	0x0000_00D0
IRQ 37	53	USART0 全局中断	0x0000_00D4
IRQ 38	54	USART1 全局中断	0x0000_00D8
IRQ 39	55	USART2 全局中断	0x0000_00DC
IRQ 40	56	EXTI[15:10]中断请求	0x0000_00E0
IRQ 41	57	来自 EXTI17 的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	来自 EXTI18 的 USBFS 唤醒中断	0x0000_00E8
IRQ 43	59	定时器 7 中止中断	0x0000_00EC
IRQ 44	60	TIMER7 更新中断和 TIMER12 全局中断	0x0000_00F0
IRQ 45	61	TIMER7 触发与通道换相中断和 TIMER13 全局中断	0x0000_00F4
IRQ 46	62	TIMER7 通道捕获/比较中断	0x0000_00F8
IRQ 47	63	ADC2 全局中断	0x0000_00FC
IRQ 48	64	保留	0x0000_0100
IRQ 49	65	SDIO 全局中断	0x0000_0104
IRQ50	66	TIMER4 全局中断	0x0000_0108
IRQ51	67	SPI2 或 I2S2ADD 全局中断	0x0000_010C
IRQ52	68	UART3 全局中断	0x0000_0110
IRQ53	69	UART4 全局中断	0x0000_0114
IRQ54	70	TIMER5 或 DAC 全局中断	0x0000_0118
IRQ55	71	TIMER6 全局中断	0x0000_011C
IRQ56	72	DMA1 通道 0 全局中断	0x0000_0120
IRQ57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ61	77	DMAMUX 全局中断	0x0000_0134
IRQ62	78	保留	0x0000_0138
IRQ63	79	CAN1 发送中断	0x0000_013C
IRQ64	80	CAN1 接收 0 中断	0x0000_0140

中断编号	向量编号	中断描述	向量地址
IRQ65	81	CAN1 接收 1 中断	0x0000_0144
IRQ66	82	CAN1 EWMC 中断	0x0000_0148
IRQ67	83	USBFS 全局中断	0x0000_014C
IRQ68	84	保留	0x0000_0150
IRQ69	85	保留	0x0000_0154
IRQ70	86	保留	0x0000_0158
IRQ71	87	保留	0x0000_015C
IRQ72	88	保留	0x0000_0160
IRQ73	89	保留	0x0000_0164
IRQ74	90	保留	0x0000_0168
IRQ75	91	保留	0x0000_016C
IRQ76	92	保留	0x0000_0170
IRQ77	93	保留	0x0000_0174
IRQ78	94	CAN2 发送中断	0x0000_0178
IRQ79	95	CAN2 接收 0 中断	0x0000_017C
IRQ80	96	CAN2 接收 1 中断	0x0000_0180
IRQ81	97	CAN2 EWMC 中断	0x0000_0184
IRQ82	98	I2C2 事件中断	0x0000_0188
IRQ83	99	I2C2 错误中断	0x0000_018C
IRQ84	100	USART5 全局中断	0x0000_0190
IRQ85	101	来自 EXTI20 的 I2C2 唤醒中断	0x0000_0194
IRQ86	102	来自 EXTI21 的 USART5 唤醒中断	0x0000_0198
IRQ87	103	FPU 中断	0x0000_019C
IRQ88	104	TRNG 全局中断	0x0000_01A0
IRQ89	105	CAU 全局中断	0x0000_01A4
IRQ90	106	TIMER40 全局中断	0x0000_01A8
IRQ91	107	TIMER41 全局中断	0x0000_01AC
IRQ92	108	TIMER42 全局中断	0x0000_01B0
IRQ93	109	TIMER43 全局中断	0x0000_01B4
IRQ94	110	TIMER44 全局中断	0x0000_01B8
IRQ95	111	TIMER50 全局中断	0x0000_01BC
IRQ96	112	TIMER51 全局中断	0x0000_01C0
IRQ97	113	ICACHE 全局中断	0x0000_01C4
IRQ98	114	DCACHE 全局中断	0x0000_01C8
IRQ99	115	HAU 中断	0x0000_01CC
IRQ100	116	OSPI 中断	0x0000_01D0
IRQ101	117	SPI3 中断	0x0000_01D4
IRQ102	118	VOVD/VUVD 中断	0x0000_01D8
IRQ103	119	TIMER0 通道刹车中断	0x0000_01DC
IRQ104	120	TIMER7 通道刹车中断	0x0000_01E0
IRQ105	121	SYSCFG 中断	0x0000_01E4

中断编号	向量编号	中断描述	向量地址
IRQ106	122	RCU 时钟频率监控中断	0x0000_01E8
IRQ107	123	TIMER19 中止中断	0x0000_01EC
IRQ108	124	TIMER19 更新中断	0x0000_01F0
IRQ109	125	TIMER19 触发与通道换相中断	0x0000_01F4
IRQ110	126	TIMER19 通道捕获比较中断	0x0000_01F8
IRQ111	127	TIMER19 通道刹车中断	0x0000_01FC
IRQ112	128	CMP 中断	0x0000_0200
IRQ113	129	来自 EXTI22 的 CMP0 唤醒中断	0x0000_0204
IRQ114	130	来自 EXTI23 的 CMP1 唤醒中断	0x0000_0208

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 25 个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI 提供 3 种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI 中每个边沿检测电路

都可以分别予以配置或屏蔽。

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 9 根线，具体细节参考*表 8-3. EXTI 触发源*。通过配置 GPIO 模块的 AFIO_EXTISSx 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源，具体细节请参考*通用和备用输入/输出接口(GPIO 和 AFIO)*。

除了中断，EXTI 还可以向处理器提供事件信号。The Cortex®-M33 内核完全支持等待中断(WFI)，等待事件(WFE)和发送事件(SEV)指令。芯片内部有一个唤醒中断控制器(WIC)，用户可以放心的让处理器和 NVIC 进入功耗极低的省电模式，由 WIC 来识别中断和事件以及判断优先级。当某些预期的事件发生时，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作，EXTI 能唤醒处理器及整个系统。

8.5.1. 硬件触发

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

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

8.5.2. 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 8-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PE0
1	PA1 / PB1 / PC1 / PD1 / PE1
2	PA2 / PB2 / PC2 / PD2 / PE2
3	PA3 / PB3 / PC3 / PD3 / PE3
4	PA4 / PB4 / PC4 / PD4 / PE4
5	PA5 / PB5 / PC5 / PD5 / PE5
6	PA6 / PB6 / PC6 / PD6 / PE6
7	PA7 / PB7 / PC7 / PD7 / PE7

EXTI 线编号	触发源
8	PA8 / PB8 / PC8 / PD8 / PE8
9	PA9 / PB9 / PC9 / PD9 / PE9
10	PA10 / PB10 / PC10 / PD10 / PE10
11	PA11 / PC11 / PD11 / PE11
12	PA12 / PB12 / PC12 / PD12 / PE12
13	PA13 / PB13 / PC13 / PD13 / PE13
14	PA14 / PB14 / PC14 / PD14 / PE14
15	PA15 / PB15 / PC15 / PD15 / PE15
16	VAUVD /VAOVD / LVD / OVD
17	RTC 闹钟
18	USB 唤醒
19	RTC 唤醒
20	I2C2 唤醒
21	USART5 唤醒
22	CMP0 唤醒
23	CMP1 唤醒
24	RTC 侵入/时间戳

8.6. EXTI 寄存器

EXTI 基地址：0x4001 0400

8.6.1. 中断使能寄存器(EXTI_INTEN)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	INTENx	中断使能位 x(x = 0…24) 0：第 x 线中断被禁用 1：第 x 线中断被使能

8.6.2. 事件使能寄存器(EXTI_EVEN)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	EVENx	事件使能位 x(x = 0…24) 0：第 x 线事件被禁用 1：第 x 线事件被使能

8.6.3. 上升沿触发使能寄存器(EXTI_RTEN)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							rw	rw	w	rw	w	rw	rw	rw	rw

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	RTENx	上升沿触发使能x(x = 0…24) 0：第x线上升沿触发无效 1：第x线上升沿触发有效(中断/事件请求)

8.6.4. 下降沿触发使能寄存器(EXTI_FTEN)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	FTENx	下降沿触发使能 x(x = 0…24) 0：第 x 线下降沿触发无效 1：第 x 线下降沿触发有效(中断/事件请求)

8.6.5. 软件中断事件寄存器(EXTI_SWIEV)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	SWIEVx	中断/事件软件触发 x(x = 0…24) 0：禁用 EXTI 线 x 软件中断/事件请求 1：激活 EXTI 线 x 软件中断/事件请求

8.6.6. 挂起寄存器(EXTI_PD)

地址偏移：0x14

复位值：0Xxxxx XXXX

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	PDx	中断挂起状态 x(x = 0…24) 0：EXTI 线 x 没有被触发 1：EXTI 线 x 被触发 对这些位写 1，可将其清 0。

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。

使用 TRIGSEL，每个外设最多可以配置 4 路 TRIGSEL 输出作为该外设的触发输入信号。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

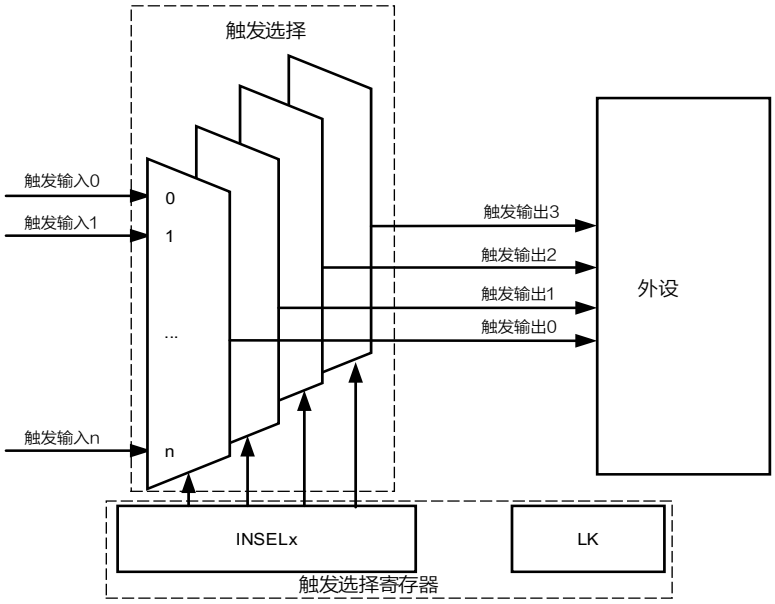
- 支持多达125个触发输入信号；
- 每个外设都有专用的触发信号选择寄存器；
- TRIGSEL寄存器最多可以配置4个输出到外围设备；
- 触发选择控制器的输入信号可来源于外部输入或外设输出；
- 触发选择控制器的输出信号可输出到外部输出或者到外设输入。

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 4 路输出，这些输出连接到外设的触发输入。每路输出均可从不同的触发输入源中选择。

*图 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	V_{SS}
	0x01	V_{DD}
	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	TIMER0_TRGO
	0x0B	TIMER0_CH0
	0x0C	TIMER0_CH1
	0x0D	TIMER0_CH2
	0x0E	TIMER0_CH3
	0x0F	TIMER7_TRGO
	0x10	TIMER7_CH0

位域名称	位域值	触发输入选择
	0x11	TIMER7_CH1
	0x12	TIMER7_CH2
	0x13	TIMER7_CH3
	0x14	TIMER19_TRGO
	0x15	TIMER19_CH0
	0x16	TIMER19_CH1
	0x17	TIMER19_CH2
	0x18	TIMER19_CH3
	0x19	TIMER5_TRGO
	0x1A	TIMER6_TRGO
	0x1B	TIMER50_TRGO
	0x1C	TIMER51_TRGO
	0x1D	TIMER1_TRGO
	0x1E	TIMER1_CH0
	0x1F	TIMER1_CH1
	0x20	TIMER1_CH2
	0x21	TIMER1_CH3
	0x22	TIMER2_TRGO
	0x23	TIMER2_CH0
	0x24	TIMER2_CH1
	0x25	TIMER2_CH2
	0x26	TIMER2_CH3
	0x27	TIMER3_TRGO
	0x28	TIMER3_CH0
	0x29	TIMER3_CH1
	0x2A	TIMER3_CH2
	0x2B	TIMER3_CH3
	0x2C	TIMER4_TRGO
	0x2D	TIMER4_CH0
	0x2E	TIMER4_CH1
	0x2F	TIMER4_CH2
	0x30	TIMER4_CH3
	0x31	TIMER9_TRGO
	0x32	TIMER9_CH0
	0x33	TIMER10_TRGO
	0x34	TIMER10_CH0
	0x35	TIMER12_TRGO
	0x36	TIMER12_CH0
	0x37	TIMER13_TRGO
	0x38	TIMER13_CH0

位域名称	位域值	触发输入选择
	0x39	TIMER14_TRGO
	0x3A	TIMER14_CH0
	0x3B	TIMER14_CH1
	0x3C	TIMER14_MCH0
	0x3D	TIMER40_TRGO
	0x3E	TIMER40_CH0
	0x3F	TIMER40_CH1
	0x40	TIMER40_MCH0
	0x41	TIMER41_TRGO
	0x42	TIMER41_CH0
	0x43	TIMER41_CH1
	0x44	TIMER41_MCH0
	0x45	TIMER42_TRGO
	0x46	TIMER42_CH0
	0x47	TIMER42_CH1
	0x48	TIMER42_MCH0
	0x49	TIMER43_TRGO
	0x4A	TIMER43_CH0
	0x4B	TIMER43_CH1
	0x4C	TIMER43_MCH0
	0x4D	TIMER44_TRGO
	0x4E	TIMER44_CH0
	0x4F	TIMER44_CH1
	0x50	TIMER44_MCH0
	0x51	ADC0_WD0_OUT_H
	0x52	ADC0_WD0_OUT_L
	0x53	ADC0_WD1_OUT_H
	0x54	ADC0_WD1_OUT_L
	0x55	ADC0_WD2_OUT_H
	0x56	ADC0_WD2_OUT_L
	0x57	ADC1_WD0_OUT_H
	0x58	ADC1_WD0_OUT_L
	0x59	ADC1_WD1_OUT_H
	0x5A	ADC1_WD1_OUT_L
	0x5B	ADC1_WD2_OUT_H
	0x5C	ADC1_WD2_OUT_L
	0x5D	ADC2_WD0_OUT_H
	0x5E	ADC2_WD0_OUT_L
	0x5F	ADC2_WD1_OUT_H
	0x60	ADC2_WD1_OUT_L

位域名称	位域值	触发输入选择
	0x61	ADC2_WD2_OUT_H
	0x62	ADC2_WD2_OUT_L
	0x63	CMP0_OUT
	0x64	CMP1_OUT
	0x65	CK_OUT
	0x66	TIMER0_BRKIN
	0x67	TIMER0_CH0BRKIN
	0x68	TIMER0_CH1BRKIN
	0x69	TIMER0_CH2BRKIN
	0x6A	TIMER7_BRKIN
	0x6B	TIMER7_CH0BRKIN
	0x6C	TIMER7_CH1BRKIN
	0x6D	TIMER7_CH2BRKIN
	0x6E	TIMER19_BRKIN
	0x6F	TIMER19_CH0BRKIN
	0x70	TIMER19_CH1BRKIN
	0x71	TIMER19_CH2BRKIN
	0x72	TIMER14_BRKIN
	0x73	TIMER40_BRKIN
	0x74	TIMER41_BRKIN
	0x75	TIMER42_BRKIN
	0x76	TIMER43_BRKIN
	0x77	TIMER44_BRKIN
	0x78	EXTI9
	0x79	EXTI11
	0x7A	EXTI15
	0x7B	RTC_Alarm
	0x7C	RTC_TPTS

*表 9-2. TRIGSEL 输入输出映射关系*给出了 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 寄存器可以配置 4 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
VSS VDD TRIGSEL_IN0 TRIGSEL_IN1	INSELx[7:0]	TRIGSEL_EXTOUT_0	输出 0 输出 1	TRIGSEL_OUT0 TRIGSEL_OUT1

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5 TRIGSEL_IN6 TRIGSEL_IN7 TIMER0_TRGO TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER7_TRGO TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER19_TRGO TIMER19_CH0 TIMER19_CH1 TIMER19_CH2 TIMER19_CH3 TIMER5_TRGO TIMER6_TRGO TIMER50_TRGO TIMER51_TRGO TIMER1_TRGO TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3 TIMER2_TRGO TIMER2_CH0 TIMER2_CH1 TIMER2_CH2 TIMER2_CH3 TIMER3_TRGO TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3		TRIGSEL_EXTOUT_1	输出 0 输出 1	TRIGSEL_OUT2 TRIGSEL_OUT3
		TRIGSEL_EXTOUT_2	输出 0 输出 1	TRIGSEL_OUT4 TRIGSEL_OUT5
		TRIGSEL_EXTOUT_3	输出 0 输出 1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_TIMER0ITI	输出 0	TIMER0_ITI
		TRIGSEL_TIMER1ITI	输出 0	TIMER1_ITI
		TRIGSEL_TIMER2ITI	输出 0	TIMER2_ITI
		TRIGSEL_TIMER3ITI	输出 0	TIMER3_ITI
		TRIGSEL_TIMER4ITI	输出 0	TIMER4_ITI
		TRIGSEL_TIMER7ITI	输出 0	TIMER7_ITI
		TRIGSEL_TIMER14ITI	输出 0	TIMER14_ITI

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER4_TRGO TIMER4_CH0 TIMER4_CH1 TIMER4_CH2 TIMER4_CH3 TIMER9_TRGO TIMER9_CH0 TIMER10_TRGO TIMER10_CH0 TIMER12_TRGO TIMER12_CH0 TIMER13_TRGO TIMER13_CH0 TIMER14_TRGO TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER40_TRGO TIMER40_CH0 TIMER40_CH1 TIMER40_MCH0 TIMER41_TRGO TIMER41_CH0 TIMER41_CH1 TIMER41_MCH0 TIMER42_TRGO TIMER42_CH0 TIMER42_CH1 TIMER42_MCH0 TIMER43_TRGO TIMER43_CH0 TIMER43_CH1 TIMER43_MCH0 TIMER44_TRGO TIMER44_CH0 TIMER44_CH1 TIMER44_MCH0 ADC0_WD0_OUT_H ADC0_WD0_OUT_L ADC0_WD1_OUT_H		TRIGSEL_TIMER19ITI	输出 0	TIMER19_ITI
		TRIGSEL_TIMER40ITI	输出 0	TIMER40_ITI
		TRIGSEL_TIMER41ITI	输出 0	TIMER41_ITI
		TRIGSEL_TIMER42ITI	输出 0	TIMER42_ITI
		TRIGSEL_TIMER43ITI	输出 0	TIMER43_ITI
		TRIGSEL_TIMER44ITI	输出 0	TIMER44_ITI
		TRIGSEL_DAC0OUT0	输出 0	DAC0_OUT0_EXTRG
		TRIGSEL_DAC0OUT1	输出 0	DAC0_OUT1_EXTRG
		TRIGSEL_ADC0_ROUTRG	输出 0	ADC0_ROUTRG
		TRIGSEL_ADC0_INSTRG	输出 0	ADC0_INSTRG

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
ADC0_WD1_OUT_L ADC0_WD2_OUT_H ADC0_WD2_OUT_L ADC1_WD0_OUT_H ADC1_WD0_OUT_L ADC1_WD1_OUT_H ADC1_WD1_OUT_L ADC1_WD2_OUT_H ADC1_WD2_OUT_L ADC2_WD0_OUT_H ADC2_WD0_OUT_L ADC2_WD1_OUT_H ADC2_WD1_OUT_L ADC2_WD2_OUT_H ADC2_WD2_OUT_L CMP0_OUT CMP1_OUT CK_OUT TIMER0_BRKIN TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN TIMER7_BRKIN TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN TIMER19_BRKIN TIMER19_CH0BRKIN TIMER19_CH1BRKIN TIMER19_CH2BRKIN TIMER14_BRKIN TIMER40_BRKIN TIMER41_BRKIN TIMER42_BRKIN TIMER43_BRKIN TIMER44_BRKIN EXTI9 EXTI11 EXTI15 RTC_Alarm		TRIGSEL_ADC1_ROUTRG	输出 0	ADC1_ROUTRG
		TRIGSEL_ADC1_INSTRG	输出 0	ADC1_INSTRG
		TRIGSEL_ADC2_ROUTRG	输出 0	ADC2_ROUTRG
		TRIGSEL_ADC2_INSTRG	输出 0	ADC2_INSTRG
		TRIGSEL_TIMER0BRKIN	输出 0	TIMER0_BRKIN
		TRIGSEL_TIMER0CHBRKIN	输出 0 输出 1 输出 2	TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN
		TRIGSEL_TIMER7BRKIN	输出 0	TIMER7_BRKIN
		TRIGSEL_TIMER7CHBRKIN	输出 0 输出 1 输出 2	TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
RTC_TPTS		TRIGSEL_TIMER19BRKIN	输出 0	TIMER19_BRKIN
		TRIGSEL_TIMER19CHBRKIN	输出 0 输出 1 输出 2	TIMER19_CH0BRKIN TIMER19_CH1BRKIN TIMER19_CH2BRKIN
		TRIGSEL_TIMER14BRKIN	输出 0	TIMER14_BRKIN
		TRIGSEL_TIMER40BRKIN	输出 0	TIMER40_BRKIN
		TRIGSEL_TIMER41BRKIN	输出 0	TIMER41_BRKIN
		TRIGSEL_TIMER42BRKIN	输出 0	TIMER42_BRKIN
		TRIGSEL_TIMER43BRKIN	输出 0	TIMER43_BRKIN
		TRIGSEL_TIMER44BRKIN	输出 0	TIMER44_BRKIN
		TRIGSEL_TIMER9ITI	输出 0 输出 1	TIMER9_ITI2 TIMER9_ITI13
		TRIGSEL_TIMER10ITI	输出 0 输出 1	TIMER10_ITI2 TIMER10_ITI13

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
		TRIGSEL_TIMER12ITI	输出 0 输出 1	TIMER12_ITI2 TIMER12_ITI13
		TRIGSEL_TIMER13ITI	输出 0 输出 1	TIMER13_ITI2 TIMER13_ITI13

注意：

1. TRIGSEL_OUTx (x=0,…,7) 不能选择任何TRIGSEL_INy (y=0,…,7)作为触发源。
2. 所有TIMERx_ITI (x=0…4,7,9,10,12,13,14,19,40…44)不能选择来自其自身模块的信号作为触发输入源。
3. 除1、2所述之外，所有输出都可以选择所有输入作为触发源。

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。当选择了非法触发输入（如 INSELx[7:0]位域值大于 0x7C，或违反上述注意事项 1、2 所述的限制）时，其输出将被强制选择为 0。

9.5. TRIGSEL 寄存器

TRIGSEL 基地址：0x4001 9000

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 寄存器可读可写 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 寄存器可读可写 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 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_2将不能被修改。 0：TRIGSEL_EXTOUT_2 寄存器可读可写 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 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_3将不能被修改。 0：TRIGSEL_EXTOUT_3 寄存器可读可写 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. TIMER0_ITI 触发选择寄存器(TRIGSEL_TIMER0ITI)

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
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_TIMER0ITI将不能被修改。 0：TRIGSEL_TIMER0ITI 寄存器可读可写 1：TRIGSEL_TIMER0ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_ITI 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. TIMER1_ITI 触发选择寄存器(TRIGSEL_TIMER1ITI)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

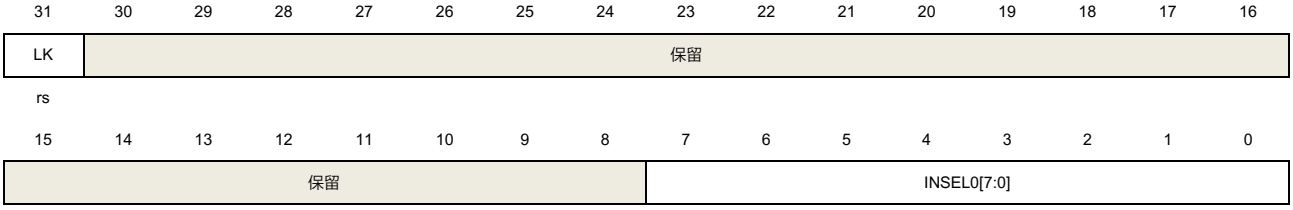
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_TIMER1ITI将不能被修改。 0：TRIGSEL_TIMER1ITI 寄存器可读可写 1：TRIGSEL_TIMER1ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER1_ITI 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. TIMER2_ITI 触发选择寄存器(TRIGSEL_TIMER2ITI)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

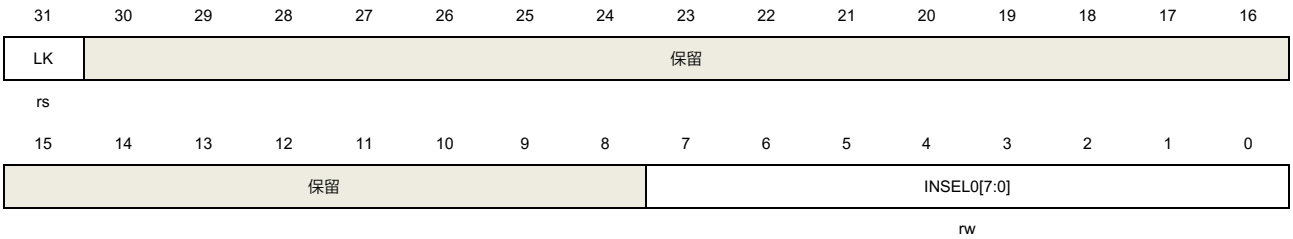


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER2ITI将不能被修改。 0：TRIGSEL_TIMER2ITI 寄存器可读可写 1：TRIGSEL_TIMER2ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER2_ITI 的触发源。 关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.8. TIMER3_ITI 触发选择寄存器(TRIGSEL_TIMER3ITI)

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	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_TIMER3ITI将不能被修改。 0：TRIGSEL_TIMER3ITI 寄存器可读可写

		1：TRIGSEL_TIMER3ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER3_ITI 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. TIMER4_ITI 触发选择寄存器(TRIGSEL_TIMER4ITI)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	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_TIMER4ITI将不能被修改。 0：TRIGSEL_TIMER4ITI 寄存器可读可写 1：TRIGSEL_TIMER4ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER4_ITI 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. TIMER7_ITI 触发选择寄存器(TRIGSEL_TIMER7ITI)

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
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_TIMER7ITI将不能被修改。 0：TRIGSEL_TIMER7ITI 寄存器可读可写 1：TRIGSEL_TIMER7ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_ITI 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. TIMER14_ITI 触发选择寄存器(TRIGSEL_TIMER14ITI)

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
LK	保留														
rs															

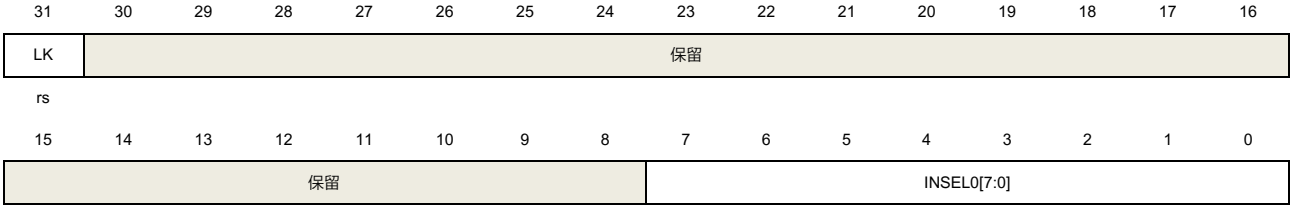
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_TIMER14ITI将不能被修改。 0：TRIGSEL_TIMER14ITI 寄存器可读可写 1：TRIGSEL_TIMER14ITI 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_ITI 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. TIMER19_ITI 触发选择寄存器(TRIGSEL_TIMER19ITI)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

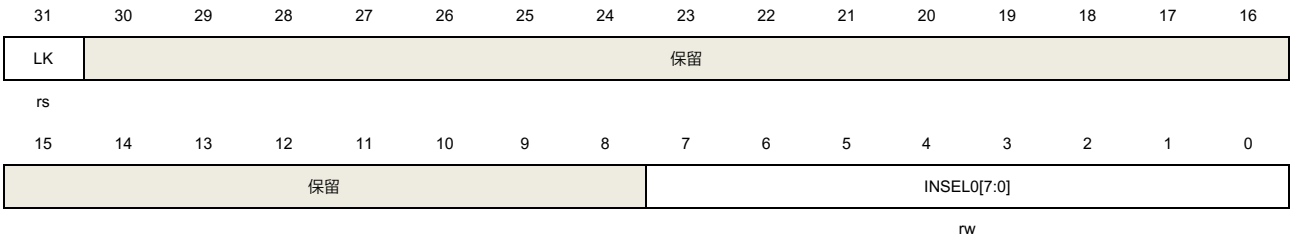


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19ITI将不能被修改。 0：TRIGSEL_TIMER19ITI 寄存器可读可写 1：TRIGSEL_TIMER19ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19_ITI 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.13. TIMER40_ITI 触发选择寄存器(TRIGSEL_TIMER40ITI)

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	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_TIMER40ITI将不能被修改。 0：TRIGSEL_TIMER40ITI 寄存器可读可写

		1：TRIGSEL_TIMER40ITI 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_ITI 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. TIMER41_ITI 触发选择寄存器(TRIGSEL_TIMER41ITI)

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
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_TIMER41ITI将不能被修改。 0：TRIGSEL_TIMER41ITI 寄存器可读可写 1：TRIGSEL_TIMER41ITI 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_ITI 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. TIMER42_ITI 触发选择寄存器(TRIGSEL_TIMER42ITI)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	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_TIMER42ITI将不能被修改。 0：TRIGSEL_TIMER42ITI 寄存器可读可写 1：TRIGSEL_TIMER42ITI 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_ITI 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. TIMER43_ITI 触发选择寄存器(TRIGSEL_TIMER43ITI)

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
LK	保留														
rs															

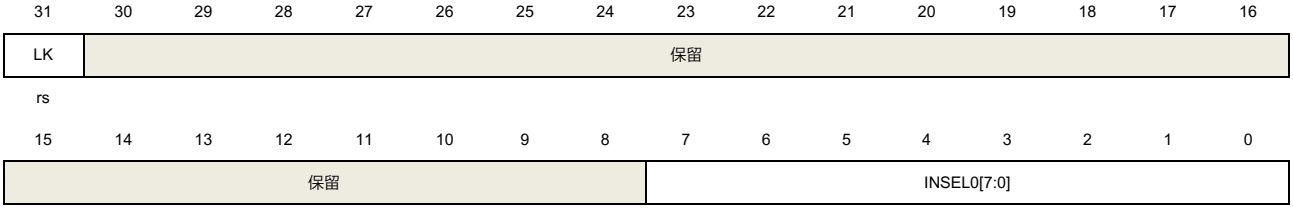
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_TIMER43ITI将不能被修改。 0：TRIGSEL_TIMER43ITI 寄存器可读可写 1：TRIGSEL_TIMER43ITI 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_ITI 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. TIMER44_ITI 触发选择寄存器(TRIGSEL_TIMER44ITI)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

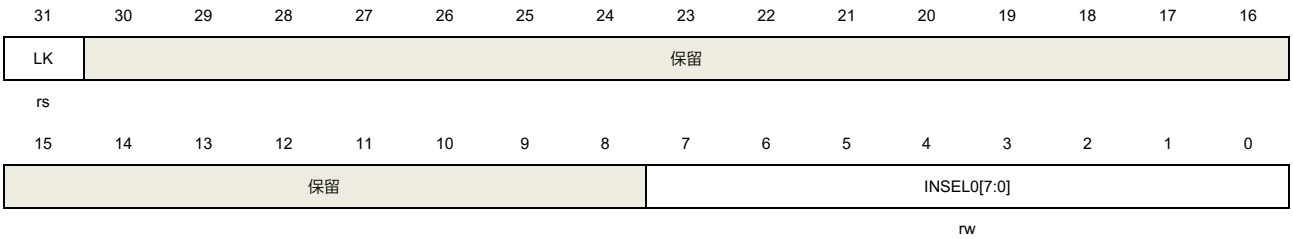


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER44ITI将不能被修改。 0：TRIGSEL_TIMER44ITI 寄存器可读可写 1：TRIGSEL_TIMER44ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER44_ITI 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.18. DAC 触发选择寄存器 0(TRIGSEL_DAC0OUT0)

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	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_EXTRG (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*。

9.5.19. DAC 触发选择寄存器 1(TRIGSEL_DAC0OUT1)

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
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_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_EXTRG (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*。

9.5.20. ADC0_ROUTRG 触发选择寄存器(TRIGSEL_ADC0_ROUTRG)

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
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_ROUTRG 将不能被修改。 0：TRIGSEL_ADC0_ROUTRG 寄存器可读可写 1：TRIGSEL_ADC0_ROUTRG 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 ADC0_ROUTRG 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. ADC0_INSTRG 触发选择寄存器(TRIGSEL_ADC0_INSTRG)

地址偏移：0x50
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

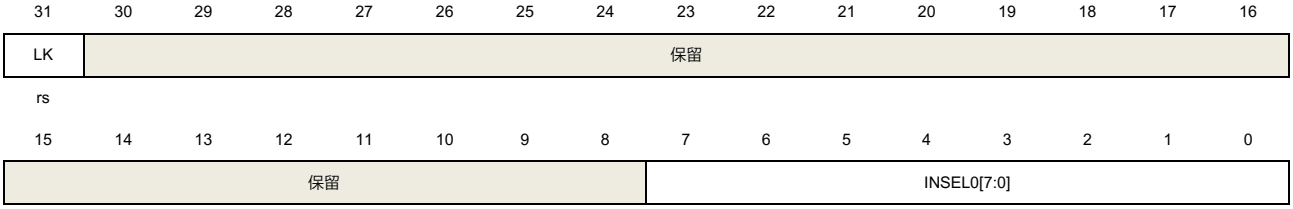
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
								rw							

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_INSTRG将不能被修改。 0：TRIGSEL_ADC0_INSTRG 寄存器可读可写 1：TRIGSEL_ADC0_INSTRG 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 ADC0_INSTRG 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. ADC1_ROUTRG 触发选择寄存器(TRIGSEL_ADC1_ROUTRG)

地址偏移：0x54
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

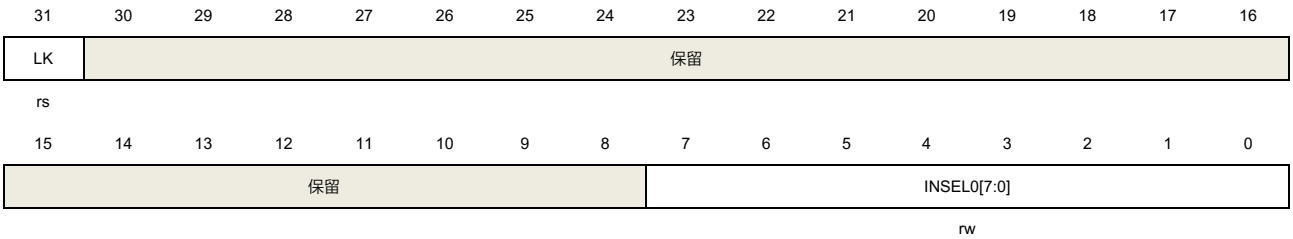


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC1_ROUTRG 将不能被修改。 0：TRIGSEL_ADC1_ROUTRG 寄存器可读可写 1：TRIGSEL_ADC1_ROUTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC1_ROUTRG 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.23. ADC1_INSTRG 触发选择寄存器(TRIGSEL_ADC1_INSTRG)

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	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_INSTRG将不能被修改。 0：TRIGSEL_ADC1_INSTRG 寄存器可读可写

		1：TRIGSEL_ADC1_INSTRG 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 ADC1_INSTRG 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. ADC2_ROUTRG 触发选择寄存器(TRIGSEL_ADC2_ROUTRG)

地址偏移：0x5C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC2_ROUTRG 将不能被修改。 0：TRIGSEL_ADC2_ROUTRG 寄存器可读可写 1：TRIGSEL_ADC2_ROUTRG 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 ADC2_ROUTRG 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. ADC2_INSTRG 触发选择寄存器(TRIGSEL_ADC2_INSTRG)

地址偏移：0x60
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

保留	INSEL0[7:0]
	rw

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC2_INSTRG将不能被修改。 0：TRIGSEL_ADC2_INSTRG 寄存器可读可写 1：TRIGSEL_ADC2_INSTRG 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 ADC2_INSTRG 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. TIMER0_BRKIN 触发选择寄存器(TRIGSEL_TIMER0BRKIN)

地址偏移：0x64
复位值：0x0000 0066

该寄存器只能按字(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							

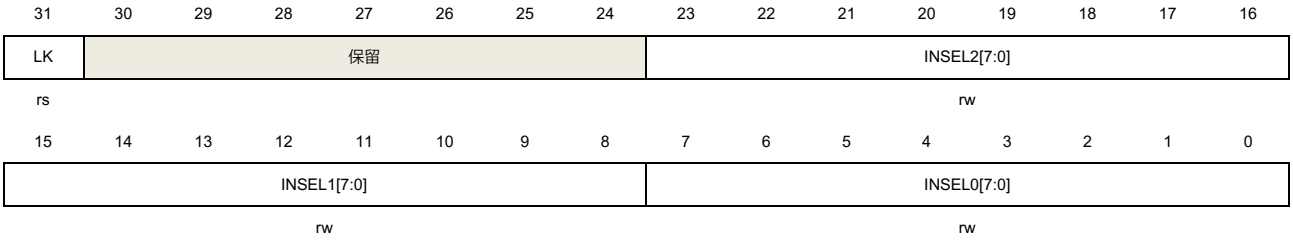
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0BRKIN将不能被修改。 0：TRIGSEL_TIMER0BRKIN 寄存器可读可写 1：TRIGSEL_TIMER0BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.27. TIMER0_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER0CHBRKIN)

地址偏移：0x68

复位值：0x0069 6867

该寄存器只能按字(32 位)访问。



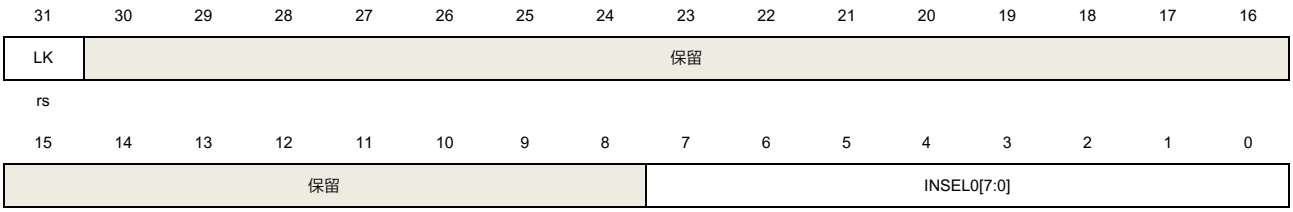
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0CHBRKIN 将不能被修改。 0：TRIGSEL_TIMER0CHBRKIN 寄存器可读可写 1：TRIGSEL_TIMER0CHBRKIN 寄存器只读
30:24	保留	必须保持复位值。
23:16	INSEL2[7:0]	触发输出 2 的输入源选择 这些位用来选择连接到输出 2 的触发输入信号，输出 2 作为 TIMER0_CH2BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。
15:8	INSEL1[7:0]	触发输出 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 TIMER0_CH1BRKIN 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_CH0BRKIN 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. TIMER7_BRKIN 触发选择寄存器(TRIGSEL_TIMER7BRKIN)

地址偏移：0x6C

复位值：0x0000 006A

该寄存器只能按字(32 位)访问。



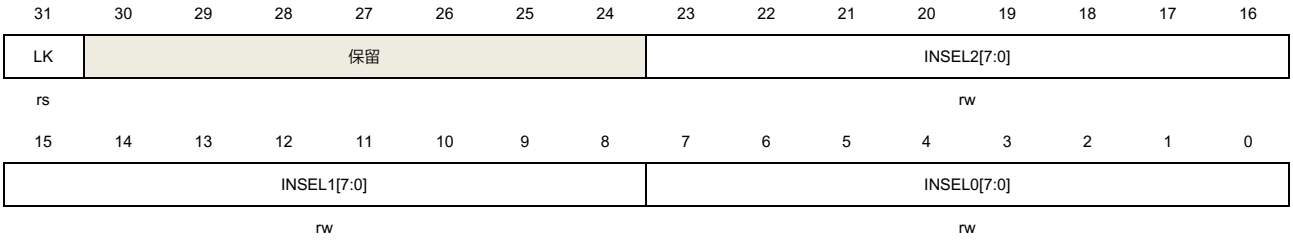
rw

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
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_BRKIN 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_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER7CHBRKIN)

地址偏移：0x70
复位值：0x006D 6C6B

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_TIMER7CHBRKIN将不能被修改。 0：TRIGSEL_TIMER7CHBRKIN 寄存器可读可写 1：TRIGSEL_TIMER7CHBRKIN 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_CH2BRKIN 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_CH1BRKIN 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

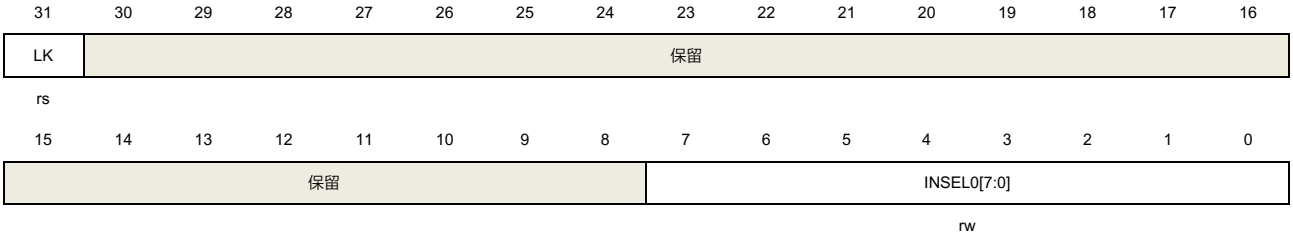
这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_CH0BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

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

地址偏移：0x74

复位值：0x0000 006E

该寄存器只能按字(32 位)访问。



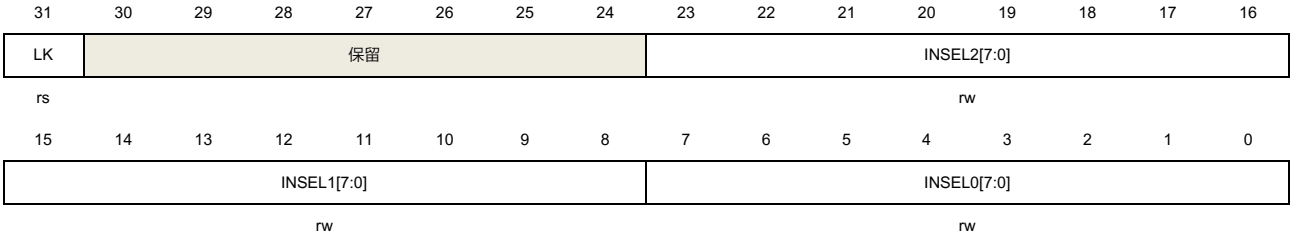
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER19BRKIN将不能被修改。 0：TRIGSEL_TIMER19BRKIN 寄存器可读可写 1：TRIGSEL_TIMER19BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER19_BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.31. TIMER19_CHBRKIN 触发选择寄存器(TRIGSEL_TIMER19CHBRKIN)

地址偏移：0x78

复位值：0x0071 706F

该寄存器只能按字(32 位)访问。



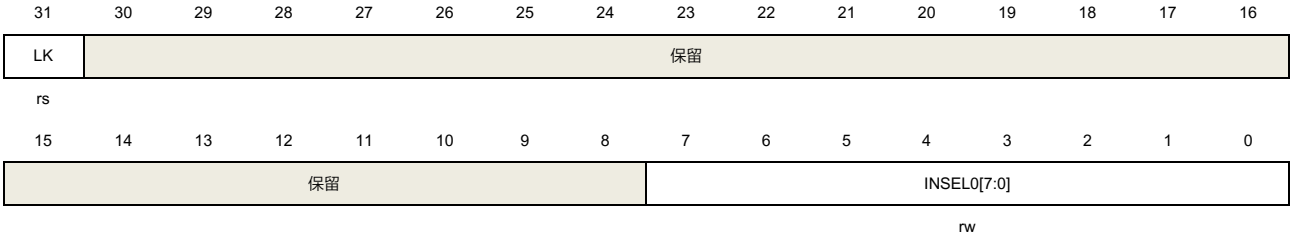
位/位域	名称	描述

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_TIMER19CHBRKIN 将不能被修改。 0：TRIGSEL_TIMER19CHBRKIN 寄存器可读可写 1：TRIGSEL_TIMER19CHBRKIN 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 TIMER19_CH2BRKIN 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 TIMER19_CH1BRKIN 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 TIMER19_CH0BRKIN 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. TIMER14_BRKIN 触发选择寄存器(TRIGSEL_TIMER14BRKIN)

地址偏移：0x7C
复位值：0x0000 0072

该寄存器只能按字(32 位)访问。

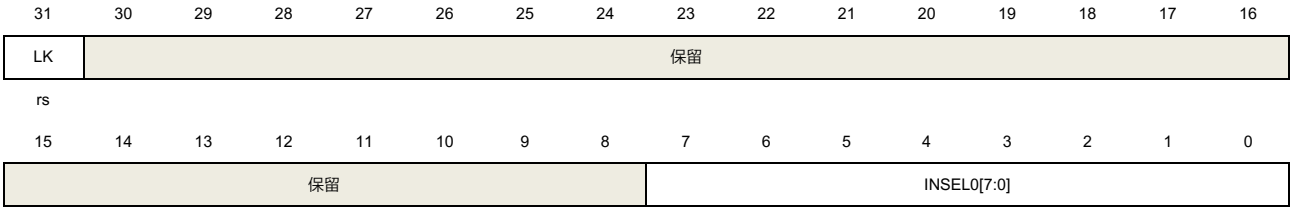


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER14BRKIN将不能被修改。 0：TRIGSEL_TIMER14BRKIN 寄存器可读可写 1：TRIGSEL_TIMER14BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER14_BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.33. TIMER40_BRKIN 触发选择寄存器(TRIGSEL_TIMER40BRKIN)

地址偏移：0x80
复位值：0x0000 0073

该寄存器只能按字(32 位)访问。

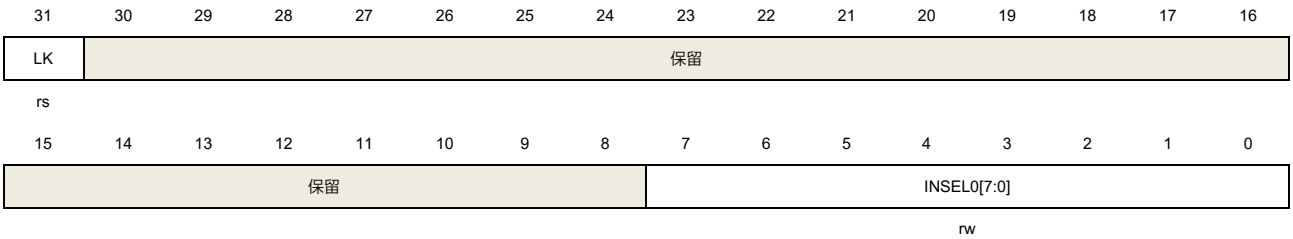


位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER40BRKIN将不能被修改。 0：TRIGSEL_TIMER40BRKIN 寄存器可读可写 1：TRIGSEL_TIMER40BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER40_BRKIN 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。

9.5.34. TIMER41_BRKIN 触发选择寄存器(TRIGSEL_TIMER41BRKIN)

地址偏移：0x84
复位值：0x0000 0074

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_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_BRKIN 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. TIMER42_BRKIN 触发选择寄存器(TRIGSEL_TIMER42BRKIN)

地址偏移：0x88
复位值：0x0000 0075

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_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_BRKIN 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. TIMER43_BRKIN 触发选择寄存器(TRIGSEL_TIMER43BRKIN)

地址偏移：0x8C
复位值：0x0000 0076

该寄存器只能按字(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_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_BRKIN 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. TIMER44_BRKIN 触发选择寄存器(TRIGSEL_TIMER44BRKIN)

地址偏移：0x90
复位值：0x0000 0077

该寄存器只能按字(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_BRKIN 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. TIMER9_ITI 触发选择寄存器(TRIGSEL_TIMER9ITI)

地址偏移：0x94

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



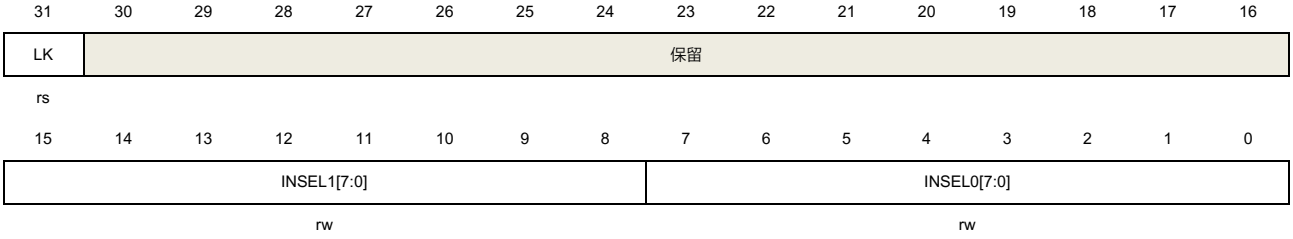
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER9ITI将不能被修改。 0：TRIGSEL_TIMER9ITI 寄存器可读可写 1：TRIGSEL_TIMER9ITI 寄存器只读
30:16	保留	必须保持复位值。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TIMER9_ITI13 的触发源。关于具体配置请参考*表 9-1.触发输入位域选择*。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER9_ITI2 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. TIMER10_ITI 触发选择寄存器(TRIGSEL_TIMER10ITI)

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_TIMER10ITI将不能被修改。 0：TRIGSEL_TIMER10ITI 寄存器可读可写 1：TRIGSEL_TIMER10ITI 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 TIMER10_ITI13 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 TIMER10_ITI2 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. TIMER12_ITI 触发选择寄存器(TRIGSEL_TIMER12ITI)

地址偏移：0x9C
复位值：0x0000 0000

该寄存器只能按字(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_TIMER12ITI将不能被修改。 0：TRIGSEL_TIMER12ITI 寄存器可读可写 1：TRIGSEL_TIMER12ITI 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 TIMER12_ITI13 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 TIMER12_ITI2 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. TIMER13_ITI 触发选择寄存器(TRIGSEL_TIMER13ITI)

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_TIMER13ITI将不能被修改。 0：TRIGSEL_TIMER13ITI 寄存器可读可写 1：TRIGSEL_TIMER13ITI 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 TIMER13_ITI13 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 TIMER13_ITI2 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

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

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

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

10.2. 主要特征

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

10.3. 功能说明

每个通用 I/O 端口可以通过 GPIO 32 位配置寄存器(GPIOx_CTL)配置为 GPIO 输入、GPIO 输出、AF 功能或模拟模式。当选择 AF 功能时，引脚的输入或输出由所选 AF 功能输出使能决定。当端口为输出(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。并且每个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 上拉/下拉寄存器(GPIOx_PUD)配置为浮空(无上拉和下拉)、上拉或下拉功能。*表 10-1. GPIO 配置表*显示了详细信息。

表 10-1. GPIO 配置表

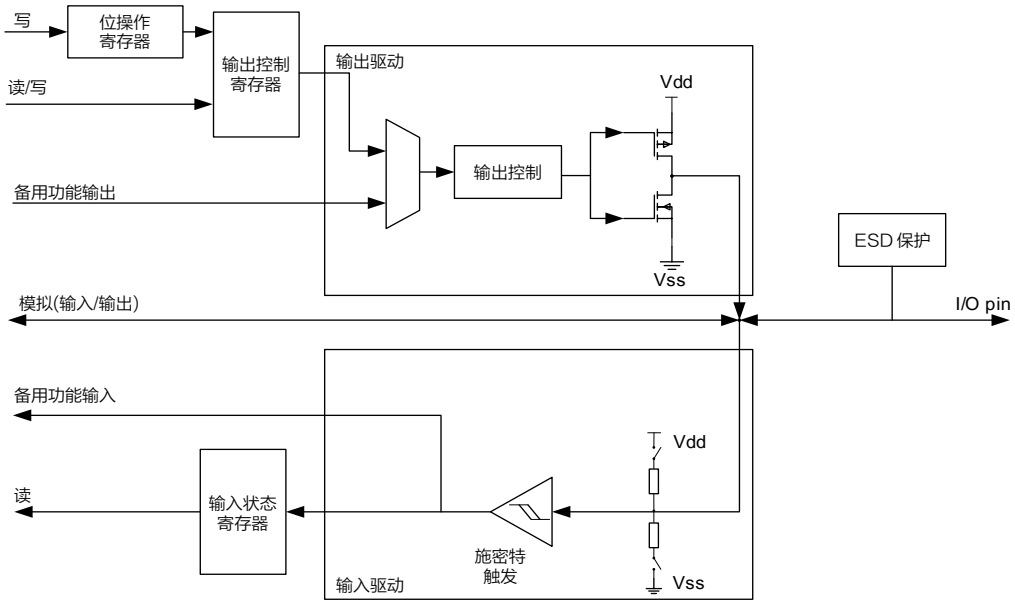
PAD类型			CTLy	PUDy
GPIO	X	悬空	00	00

PAD类型			CTLy	PUDy
输入		上拉		01
		下拉		10
GPIO 输出	推挽	悬空	01	00
		上拉		01
		下拉		10
	开漏	悬空		00
		上拉		01
		下拉		10
AFIO 输入	X	悬空	10	00
		上拉		01
		下拉		10
AFIO 输出	推挽	悬空	10	00
		上拉		01
		下拉		10
	开漏	悬空		00
		上拉		01
		下拉		10
ANALOG	X	X	11	XX

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

*图 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，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试端口(JTAG/Serial-Wired Debug pins)为输入 PU/PD 模式：

PA15：JTDI 为上拉模式；

PA14：JTCK / SWCLK 为下拉模式；

PA13：JTMS / SWDIO 为上拉模式；

PB4：NJTRST 为上拉模式。

PB3：JTDO 为浮空模式。

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

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

当对 GPIOx_OCTL 进行位操作时，不需要先读再写，用户可以通过写'1'到位操作寄存器(GPIOx_BOP，或用于清 0 的 GPIOx_BC)修改一位或几位，该过程仅需要一个最小的 APB2 写访问周期，而其他位不受影响。

10.3.2. 外部中断/事件线

所有的端口都有外部中断能力，为了使用外部中断线，端口必须配置为输入模式。

10.3.3. 备用功能(AF)

当端口配置为 AFIO(设置 GPIOx_CTL 寄存器值为"0b10")时，该端口用作外设备用功能，每个端口有 16 个 AF，需通过 GPIOx_AFSELz(z= 0,1)寄存器配置。端口备用功能分配的详细介绍见芯片数据手册。

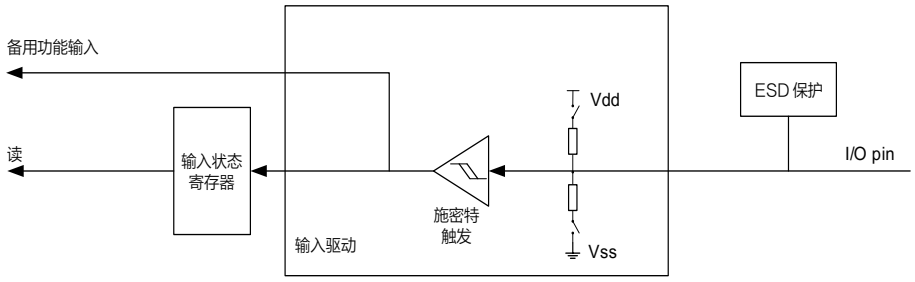
10.3.4. 输入配置

当 GPIO 引脚配置为输入时：

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

*图 10-2. 输入配置*显示 I/O 引脚的输入配置。

图 10-2. 输入配置



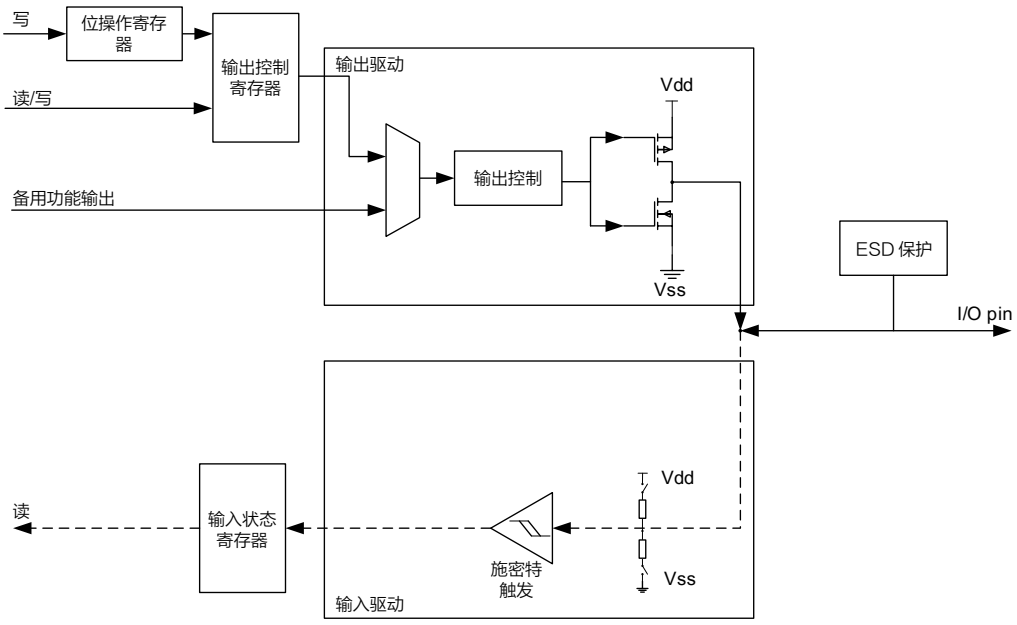
10.3.5. 输出配置

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

*图 10-3. 输出配置*是 I/O 端口的输出配置。

图 10-3. 输出配置



10.3.6. 模拟配置

当 GPIO 引脚用于模拟模式时：

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

*图 10-4. 模拟配置*是 I/O 端口的模拟模式配置。

图 10-4. 模拟配置



10.3.7. 备用功能(AF)配置

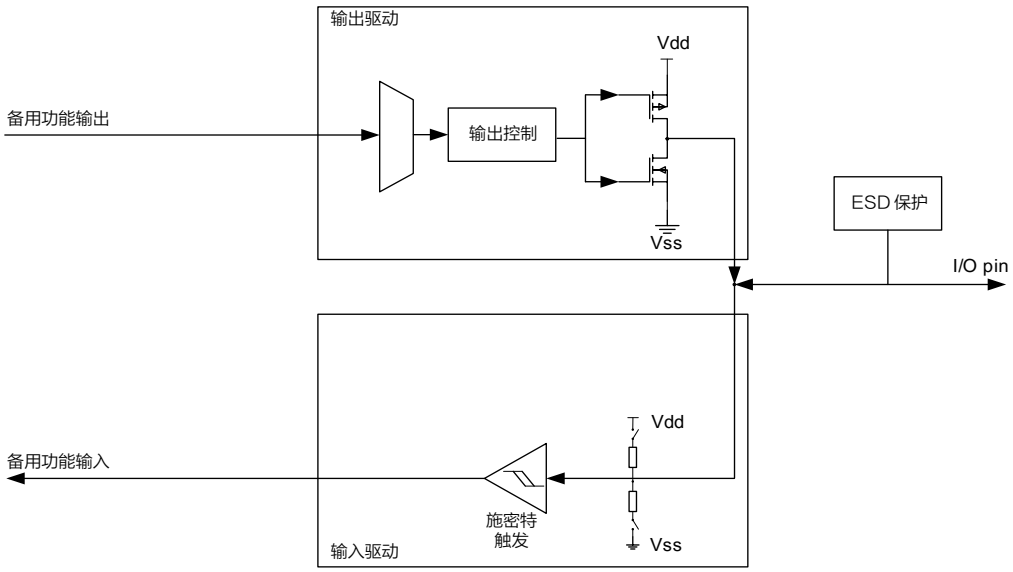
为了适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

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

*图 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



10.3.8. GPIO 锁定功能

GPIO 的锁定机制可以保护 I/O 端口的配置。

被保护的寄存器有 GPIOx_CTL、GPIOx_OMODE、GPIOx_OSPD、GPIOx_PUD 和 GPIOx_AFSELz (z=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。通过特定的锁定序列配置 GPIOx_LOCK 中的 LKK 位和 LKy 位，相应的端口位被锁定，直到下一个复位前，相应端口位的配置都不能修改。建议在电源驱动模块的配置中使用锁定功能。

10.3.9. GPIO I/O 补偿单元

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

在使能 I/O 补偿单元后，将产生一个准备完成标志位 CPS_RDY，用于指示补偿单元已经准备好，可以使用。

10.3.10. 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_APB2 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_APB2 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_APB2。

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。

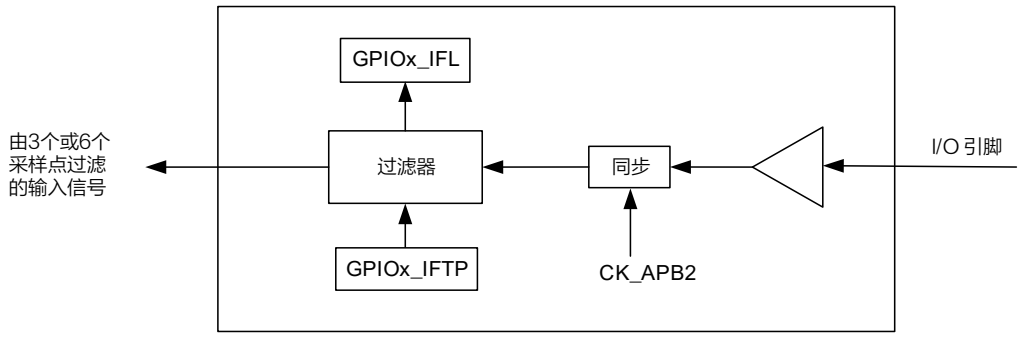
仅同步到 CK_APB2

在这种模式下，输入信号仅与 CK_APB2 同步。由于输入信号是异步的，它可能需要一个 CK_APB2 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_APB2 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 10-6. 使用采样窗口进行过滤



采样周期

为了过滤，对输入信号以一个固定的周期进行采样。采样周期由用户指定，并确定采样之间的持续时间，或相对于 CK_APB2 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_APB2}。例如，如果f_{CK_APB2}=100MHz，信号将以 100MHz 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_APB2}×1/(2×FLPRDx)。例如，如果f_{CK_APB2}=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 10-7. 输入滤波时钟周期*所示。通过使用采样周期和采样次数的进行计算，可以确定窗口的总宽度。

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

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

注意：外部信号变化与采样周期和 CK_APB2 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_APB2}。

采样窗口示例

如*图 10-7. 输入滤波时钟周期*所示的例子，输入滤波配置如下：

- GPIOx_IFTP寄存器的IFTP0[1:0]=10，表示有6个采样点；
- GPIOx_IFL寄存器的FLPRD0=1，采样周期$T_{SP}=2\times FLPRD0\times T_{CK_APB2}=2\times T_{CK_APB2}$；

配置结果如下：

- 采样宽度为：$T_{SW}=6\times T_{SP}=6\times 2\times FLPRD0\times T_{CK_APB2}=6\times 2\times T_{CK_APB2}$；
- 如果T_{CK_APB2}=10ns，则采样窗口的持续时间为：

$T_{SP}=2\times T_{CK_APB2}=2\times 10ns=20ns$

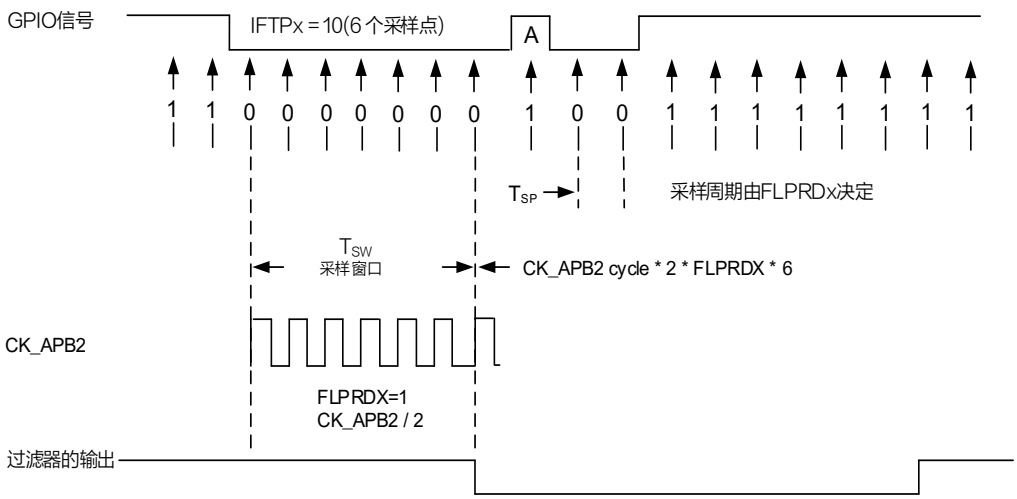
$T_{SW}=6\times 2\times T_{CK_APB2}=6\times 20ns=120ns$

- 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_APB2周期来检测输入信号的变化。

$T_{SW}+T_{CK_APB2}=120+10=130ns$

- 在*图10-7. 输入滤波时钟周期*，干扰(A)比总采样窗口短，所以将会被过滤。

图 10-7. 输入滤波时钟周期



10.3.11. CLK 引脚 AF 重映射

当 LXTAL 振荡器关闭时，LXTAL 振荡器引脚 OSC32IN 和 OSC32OUT 可以分别用作通用 I/O PC14 和 PC15。LXTAL 优先于 GPIOs 功能。

注意:

1、但是，当 V_{DD} 电源域断电(通过进入待机模式)或当备份电源域由 VBAT 供电(VDD 不再供电)时，PC14/PC15 GPIO 功能将丧失，并会被设置为模拟模式。

2、参考第 5.3.1 节中关于 I/O 使用限制的说明。

表 103. OSC32 引脚配置

备用功能	LXTAL= ON	LXTAL= OFF
PC14	OSC32IN	PC14
PC15	OSC32OUT	PC15

HXTAL 振荡器引脚 OSCIN/OSCOUT 可以被用作通用 I/O PD0/PD1。

表 104. OSC 引脚配置

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

当配置为 AF8 并使用事件输出时，如果 OSCIN/OSCOUT 和 PD0/PD1 都存在，重映射后 PD0/PD1 将输出 0。

对于 100 引脚封装，可使用单独的 PD0/PD1 引脚而不是 OSC-PD0/PD1 引脚。

10.4. GPIO 寄存器

GPIOA 基地址：0x4001 0800

GPIOB 基地址：0x4001 0C00

GPIOC 基地址：0x4001 1000

GPIOD 基地址：0x4001 1400

GPIOE 基地址：0x4001 1800

AFIO 基地址：0x4001 0000

10.4.1. 端口控制寄存器 (GPIOx_CTL, x=A...E)

地址偏移：0x00

复位值：端口 A 为 0xABFF FFFF；端口 B 为 0xFF3F FEBF；其他端口为 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
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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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 参考 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: AFIO 模式 11:模拟模式 (输入/输出) (复位状态)

10.4.2. 端口输出模式寄存器 (GPIOx_OMODE x=A...E)

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
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	OM15	Pin 15 输出模式 该位由软件置位和清除 参考 OM0 描述
14	OM14	Pin 14 输出模式 该位由软件置位和清除 参考 OM0 描述
13	OM13	Pin 13 输出模式 该位由软件置位和清除 参考 OM0 描述
12	OM12	Pin 12 输出模式 该位由软件置位和清除 参考 OM0 描述
11	OM11	Pin 11 输出模式 该位由软件置位和清除 参考 OM0 描述
10	OM10	Pin 10 输出模式 该位由软件置位和清除 参考 OM0 描述
9	OM9	Pin 9 输出模式 该位由软件置位和清除 参考 OM0 描述
8	OM8	Pin 8 输出模式 该位由软件置位和清除 参考 OM0 描述

7	OM7	Pin 7 输出模式 该位由软件置位和清除 参考 OM0 描述
6	OM6	Pin 6 输出模式 该位由软件置位和清除 参考 OM0 描述
5	OM5	Pin 5 输出模式 该位由软件置位和清除 参考 OM0 描述
4	OM4	Pin 4 输出模式 该位由软件置位和清除 参考 OM0 描述
3	OM3	Pin 3 输出模式 该位由软件置位和清除 参考 OM0 描述
2	OM2	Pin 2 输出模式 该位由软件置位和清除 参考 OM0 描述
1	OM1	Pin 1 输出模式 该位由软件置位和清除 参考 OM0 描述
0	OM0	Pin 0 输出模式位 0: 推挽输出 (复位状态) 1: 开漏输出

10.4.3. 端口输出速度寄存器 (GPIOx_OSPD x=A...E)

地址偏移：0x08
复位值：端口 A 为 0x0C00 0000；端口 B 为 0x0000 00C0；其他端口为 0x0000 0000

该寄存器只能按字(32 位)访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31:30	OSPD15[1:0]	Pin 15 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
29:28	OSPD14[1:0]	Pin 14 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
27:26	OSPD13[1:0]	Pin 13 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
25:24	OSPD12[1:0]	Pin 12 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
23:22	OSPD11[1:0]	Pin 11 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
21:20	OSPD10[1:0]	Pin 10 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
19:18	OSPD9[1:0]	Pin 9 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
17:16	OSPD8[1:0]	Pin 8 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
15:14	OSPD7[1:0]	Pin 7 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
13:12	OSPD6[1:0]	Pin 6 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
11:10	OSPD5[1:0]	Pin 5 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
9:8	OSPD4[1:0]	Pin 4 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
7:6	OSPD3[1:0]	Pin 3 输出速度位

		该位由软件置位和清除。 参考 OSPD0[1:0] 描述
5:4	OSPD2[1:0]	Pin 2 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
3:2	OSPD1[1:0]	Pin 1 输出速度位 该位由软件置位和清除。 参考 OSPD0[1:0] 描述
1:0	OSPD0[1:0]	Pin 0 输出速度位 该位由软件置位和清除。 00: 输出模式，最大速度 2 MHz 01: 输出模式，最大速度 10MHz 10: 输出模式，最大速度 50 MHz 11: 输出模式，最大速度 100MHz

10.4.4. 端口上拉/下拉寄存器(GPIOx_PUD, x=A...E)

地址偏移：0x0C

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

该寄存器只能按字(32 位)访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUD7[1:0	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
29:28	PUD14[1:0]	Pin 14 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
27:26	PUD13[1:0]	Pin 13 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
25:24	PUD12[1:0]	Pin 12 上拉模式/下拉模式 位

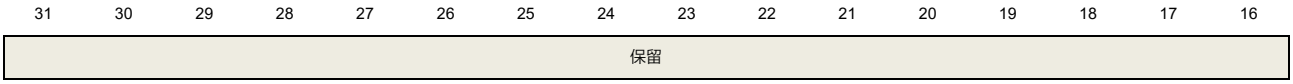
		该位由软件置位和清除。 参考 PUD0[1:0] 描述
23:22	PUD11[1:0]	Pin 11 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
21:20	PUD10[1:0]	Pin 10 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
19:18	PUD9[1:0]	Pin 9 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
17:16	PUD8[1:0]	Pin 8 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
15:14	PUD7[1:0]	Pin 7 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
13:12	PUD6[1:0]	Pin 6 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
11:10	PUD5[1:0]	Pin 5 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
9:8	PUD4[1:0]	Pin 4 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
7:6	PUD3[1:0]	Pin 3 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
5:4	PUD2[1:0]	Pin 2 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
3:2	PUD1[1:0]	Pin 1 上拉模式/下拉模式 位 该位由软件置位和清除。 参考 PUD0[1:0] 描述
1:0	PUD0[1:0]	Pin 0 上拉模式/下拉模式 位 该位由软件置位和清除。

00: 浮空输入 (复位状态)
01: 上拉模式
10: 下拉模式
11: 保留

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

地址偏移：0x10
复位值：0x0000 XXXX

该寄存器只能按字(32 位)访问。



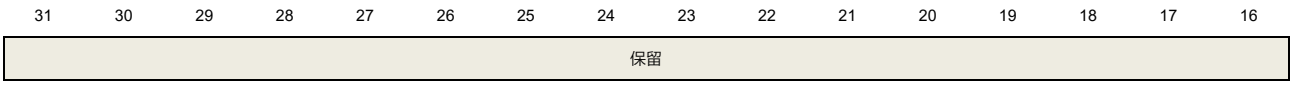


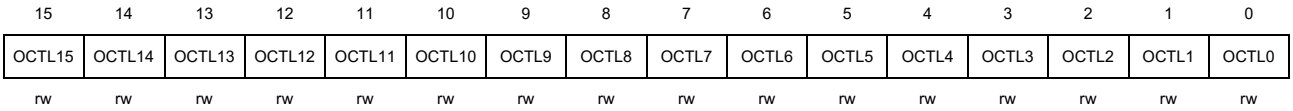
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ISTATy	端口 y 输入状态位(y=0..15) 这些位由软件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

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

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。





位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OCTLy	端口输出控制位(y=0..15) 这些位由软件置位和清除。

0：引脚输出低电平
1：引脚输出高电平

10.4.7. 端口位操作寄存器 (GPIOx_BOP, x=A...E)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	CRy	端口清除位 y(y=0..15) 这些位由软件置位和清除。 0：相应的 OCTLy 位没有改变 1：清除相应的 OCTLy 位为 0
15:0	BOPy	端口置位位 y(y=0..15) 这些位由软件置位和清除。 0：相应的 OCTLy 位没有改变 1：设置相应的 OCTLy 位为 1

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

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

10.4.9. AFIO 选择寄存器 0 (GPIOx_AFSEL0, x=A...E)

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
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 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
27:24	SEL6[3:0]	Pin 6 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
23:20	SEL5[3:0]	Pin 5 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
19:16	SEL4[3:0]	Pin 4 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
15:12	SEL3[3:0]	Pin 3 AFIO 选择 这些位由软件置位和清除。

		参考 SEL0 [3:0] 描述
11:8	SEL2[3:0]	Pin 2 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
7:4	SEL1[3:0]	Pin 1 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
3:0	SEL0[3:0]	Pin 0 AFIO 选择 这些位由软件置位和清除。 0000: 选择 AF0 (复位状态) 0001: 选择 AF1 0010: 选择 AF2 0011: 选择 AF3 … 1111: 选择 AF15

10.4.10. AFIO 选择寄存器 1(GPIOx_AFSEL1, x=A...E)

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
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				Rw			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
27:24	SEL14[3:0]	Pin 14 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
23:20	SEL13[3:0]	Pin 13 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
19:16	SEL12[3:0]	Pin 12 AFIO 选择

		这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
15:12	SEL11[3:0]	Pin 11 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
11:8	SEL10[3:0]	Pin 10 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
7:4	SEL9[3:0]	Pin 9 AFIO 选择 这些位由软件置位和清除。 参考 SEL0 [3:0] 描述
3:0	SEL8[3:0]	Pin 8 AFIO 选择 这些位由软件置位和清除。 0000: 选择 AF0 (复位状态) 0001: 选择 AF1 0010: 选择 AF2 0011: 选择 AF3 … 1111: 选择 AF15

10.4.11. 位清除寄存器 (GPIOx_BC, x=A...E)

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
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRy	端口清除位 y(y=0..15) 这些位由软件置位和清除。 0：相应 OCTLy 位没有改变 1：清除相应的 OCTLy 位

10.4.12. 输入滤波寄存器(GPIOx_IFL, x=A...E)

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLPRD1								FLPRD0							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	FLPRD1	GPIO8 到 GPIO15 的过滤采样频率： 00：FLPRDx = CK_APB2 01：FLPRDx = CK_APB2 / 2 02：FLPRDx = CK_APB2 / 4 .... FF：FLPRDx = CK_APB2 / 510
7:0	FLPRD0	GPIO0 到 GPIO7 的过滤采样频率： 00：FLPRDx = CK_APB2 01：FLPRDx = CK_APB2 / 2 02：FLPRDx = CK_APB2 / 4 .... FF：FLPRDx = CK_APB2 / 510

10.4.13. 输入滤波类型寄存器(GPIOx_IFTP, x=A...E)

地址偏移：0x34

复位值：0xFFFF FFFF

该寄存器可以按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IFTP15[1:0]		IFTP14[1:0]		IFTP13[1:0]		IFTP12[1:0]		IFTP11[1:0]		IFTP10[1:0]		IFTP9[1:0]		IFTP8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFTP7[1:0]		IFTP6[1:0]		IFTP5[1:0]		IFTP4[1:0]		IFTP3[1:0]		IFTP2[1:0]		IFTP1[1:0]		IFTP0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述

31:30	IFTP15[1:0]	Pin 15 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
29:28	IFTP14[1:0]	Pin 14 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
27:26	IFTP13[1:0]	Pin 13 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
25:24	IFTP12[1:0]	Pin 12 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
23:22	IFTP11[1:0]	Pin 11 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
21:20	IFTP10[1:0]	Pin 10 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
19:18	IFTP9[1:0]	Pin 9 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
17:16	IFTP8[1:0]	Pin 8 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
15:14	IFTP7[1:0]	Pin 7 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
13:12	IFTP6[1:0]	Pin 6 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
11:10	IFTP5[1:0]	Pin 5 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
9:8	IFTP4[1:0]	Pin 4 的过滤输入类型 该位由软件置位和清除。 参照 IFTP0[1:0]的描述
7:6	IFTP3[1:0]	Pin 3 的过滤输入类型

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

10.4.14. EXTI 源选择寄存器 0 寄存器 (AFIO_EXTISS0)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

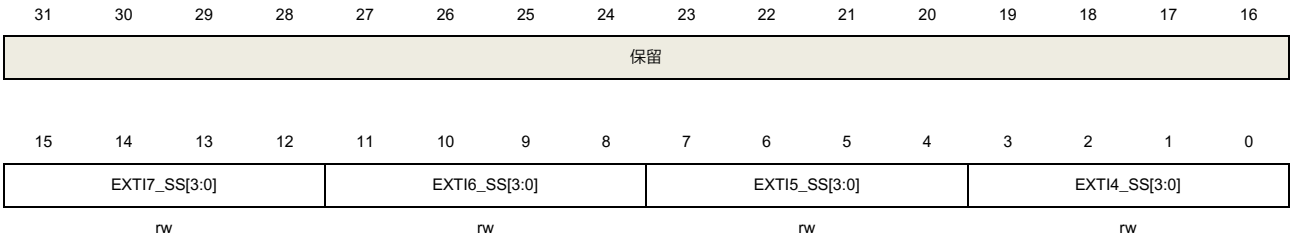
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI3 源选择 0000：PA3 引脚 0001：PB3 引脚 0010：PC3 引脚 0011：PD3 引脚 0100：PE3 引脚 其他配置保留。
11:8	EXTI2_SS[3:0]	EXTI2 源选择 0000：PA2 引脚 0001：PB2 引脚

		0010：PC2 引脚 0011：PD2 引脚 0100：PE2 引脚 其他配置保留。
7:4	EXTI1_SS[3:0]	EXTI1 源选择 0000：PA1 引脚 0001：PB1 引脚 0010：PC1 引脚 0011：PD1 引脚 0100：PE1 引脚 其他配置保留。
3:0	EXTI0_SS[3:0]	EXTI0 源选择 0000：PA0 引脚 0001：PB0 引脚 0010：PC0 引脚 0011：PD0 引脚 0100：PE0 引脚 其他配置保留。

10.4.15. EXTI 源选择寄存器 1 寄存器 (AFIO_EXTISS1)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI7 源选择 0000：PA7 引脚 0001：PB7 引脚 0010：PC7 引脚 0011：PD7 引脚 0100：PE7 引脚 其他配置保留。

11:8	EXTI6_SS[3:0]	EXTI 6 源选择 0000：PA6 引脚 0001：PB6 引脚 0010：PC6 引脚 0011：PD6 引脚 0100：PE6 引脚 其他配置保留。
7:4	EXTI5_SS[3:0]	EXTI5 源选择 0000：PA5 引脚 0001：PB5 引脚 0010：PC5 引脚 0011：PD5 引脚 0100：PE5 引脚 其他配置保留。
3:0	EXTI4_SS[3:0]	EXTI4 源选择 0000：PA4 引脚 0001：PB4 引脚 0010：PC4 引脚 0011：PD4 引脚 0100：PE4 引脚 其他配置保留。

10.4.16. EXTI 源选择寄存器 2 寄存器 (AFIO_EXTISS2)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS[3:0]				EXTI10_SS[3:0]				EXTI9_SS[3:0]				EXTI8_SS[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI11 源选择 0000：PA11 引脚 0001：PB11 引脚 0010：PC11 引脚

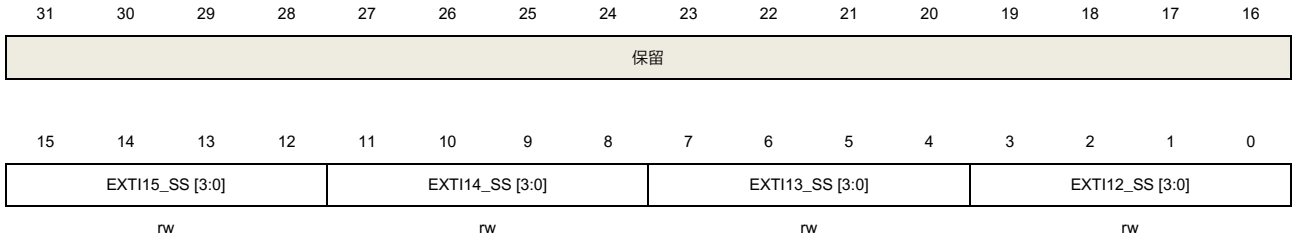
		0011：PD11 引脚 0100：PE11 引脚 其他配置保留。
11:8	EXTI10_SS[3:0]	EXTI10 源选择 0000：PA10 引脚 0001：PB10 引脚 0010：PC10 引脚 0011：PD10 引脚 0100：PE10 引脚 其他配置保留。
7:4	EXTI9_SS[3:0]	EXTI9 源选择 0000：PA9 引脚 0001：PB9 引脚 0010：PC9 引脚 0011：PD9 引脚 0100：PE9 引脚 其他配置保留。
3:0	EXTI8_SS[3:0]	EXTI8 源选择 0000：PA8 引脚 0001：PB8 引脚 0010：PC8 引脚 0011：PD8 引脚 0100：PE8 引脚 其他配置保留。

10.4.17. EXTI 源选择寄存器 3 寄存器 (AFIO_EXTISS3)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI15 源选择

0000：PA15 引脚
0001：PB15 引脚
0010：PC15 引脚
0011：PD15 引脚
0100：PE15 引脚
其他配置保留。

11:8	EXTI14_SS[3:0]	EXTI14 源选择 0000：PA14 引脚 0001：PB14 引脚 0010：PC14 引脚 0011：PD14 引脚 0100：PE14 引脚 其他配置保留。
7:4	EXTI13_SS[3:0]	EXTI13 源选择 0000：PA13 引脚 0001：PB13 引脚 0010：PC13 引脚 0011：PD13 引脚 0100：PE13 引脚 其他配置保留。
3:0	EXTI12_SS[3:0]	EXTI12 源选择 0000：PA12 引脚 0001：PB12 引脚 0010：PC12 引脚 0011：PD12 引脚 0100：PE12 引脚 其他配置保留。

10.4.18. IO 补偿控制寄存器 (AFIO_CPSCTL)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

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

10.4.19. PD01 重映射寄存器 (AFIO_PD01)

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
31:1	保留	必须保持复位值。
0	PD01_REMAPEN	PD0/PD1 重映射使能寄存器 当 HXTAL 关闭时，可通过配置该寄存器使 PD0/PD1 仍作普通 IO。 0：OSCIN 和 OSCOUT 1：PD0 和 PD1

11. 循环冗余校验计算单元(CRC)

11.1. 简介

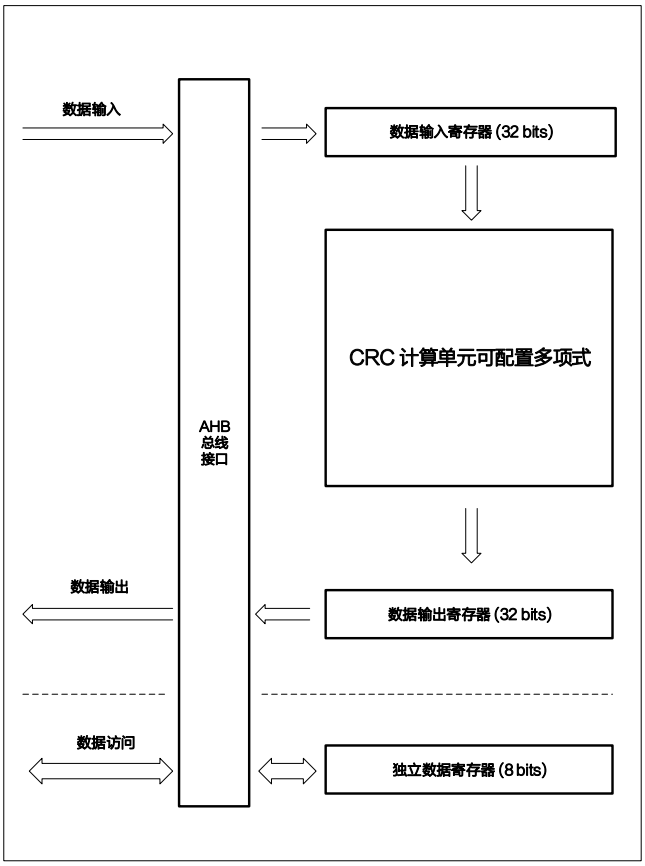
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

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8 / 16 / 32位数据输入；
- 对于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(7 / 8 / 16 / 32)；
- 用户可以配置CRC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 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。当 PS[1:0]或多项式改变后，需要复位 CRC。

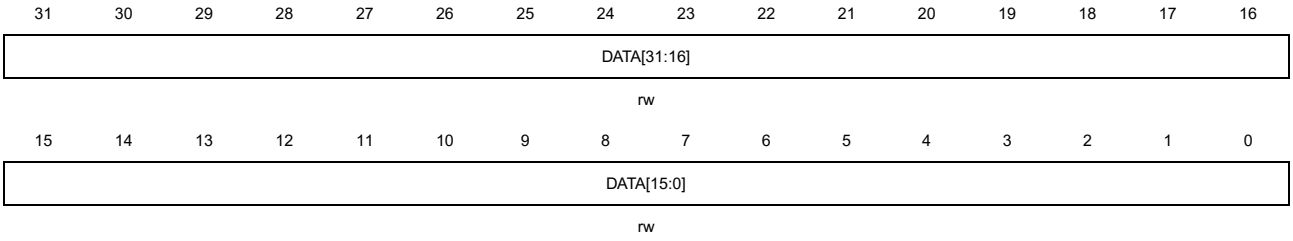
11.4. CRC 寄存器

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。

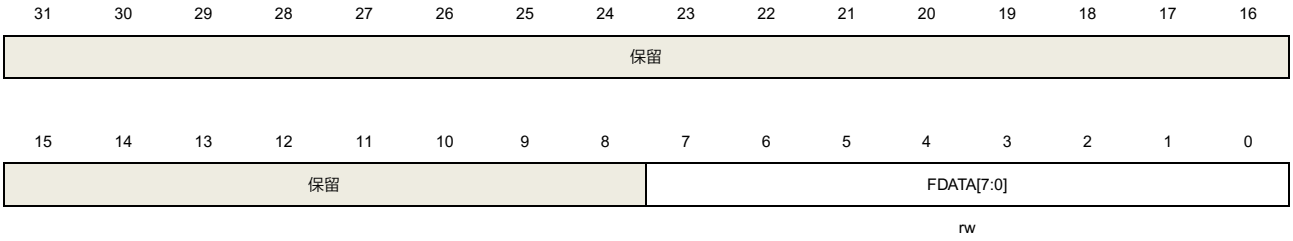


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

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。



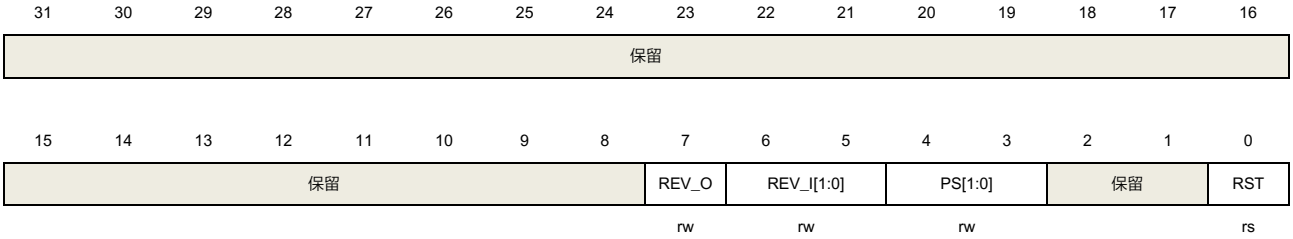
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受

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。

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

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

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

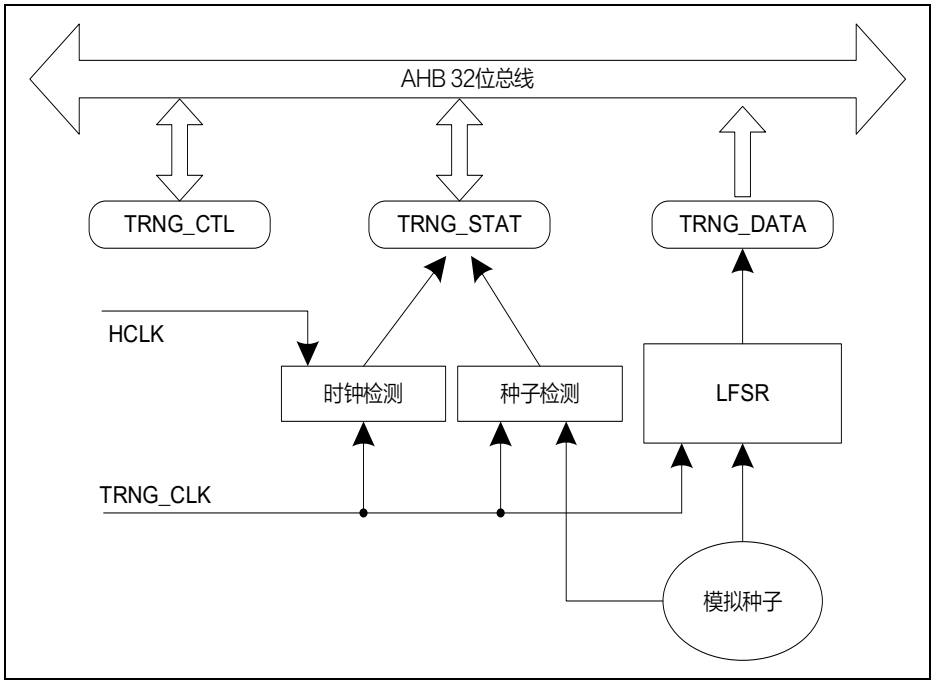
真随机数发生器模块(TRNG)能够通过连续模拟噪声生成一个 32 位的随机数值。

12.2. 主要特征

- 两个连续随机数的间隔大约为40个TRNG_CLK时钟周期；
- 禁用TRNG模块将显著降低芯片的功耗；
- 32位随机数的种子是由模拟噪声产生的，因此该随机数是一个真随机数值。

12.3. 功能说明

图 12-1. TRNG 模块框图



随机数种子由模拟电路实现。模拟种子信号输出到一个线性反馈移位寄存器(LFSR)之后在该寄存器中转化成一个 32 位宽度的随机数。

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** 相关章节)驱动，因此随机数质量仅与 TRNG_CLK 时钟有关，与 HCLK 频率无关。

当有足够数量的种子被输入 LFSR 之后，LFSR 会输出 32 位数据到 TRNG_DATA 寄存器。

同时，系统会监视模拟种子和 TRNG_CLK 时钟。一旦模拟种子发生错误或者时钟产生错误，

TRNG_STAT 寄存器的相关状态位将被置 1，如果 TRNG_CTL 寄存器的 TRNGIE 位同时被置 1 还将产生中断。

12.3.1. 操作流程

以下步骤为 TRNG 模块的推荐操作流程：

1)根据需要使能中断，这样当随机数或错误产生时，将会触发一个中断；

2)使能 TRNGEN 位；

3)等待中断发生，检测 TRNG_STAT 寄存器，如果 SEIF = 0，CEIF = 0 并且 DRDY = 1 那么数据寄存器中的随机值可以被读取。

按照 FIPS PUB 140-2 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。

12.3.2. 错误标志

■ 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。

■ 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。此时，数据寄存器中的随机数值不应当被使用，并且软件需要清除 SEIF 位。之后将 TRNGEN 位清零并置 1 以便重新启动 TRNG 模块。

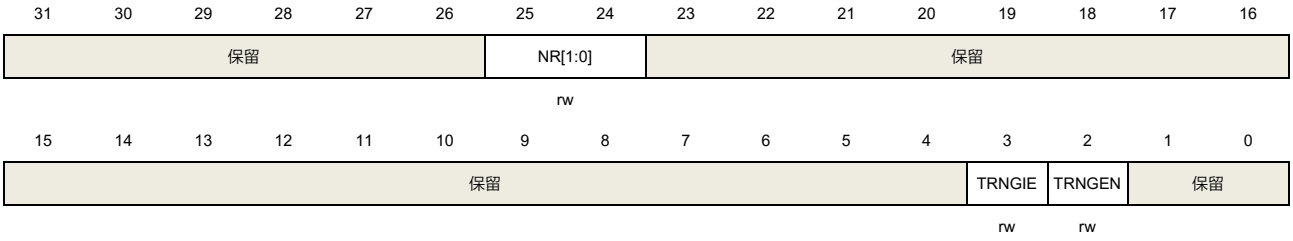
12.4. TRNG 寄存器

TRNG 基地址：0x4002 3C00

12.4.1. 控制寄存器(TRNG_CTL)

地址偏移：0x00
复位值：0x3000 0000

该寄存器可以按字(32 位)访问。

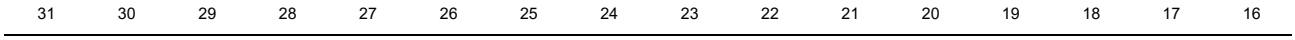


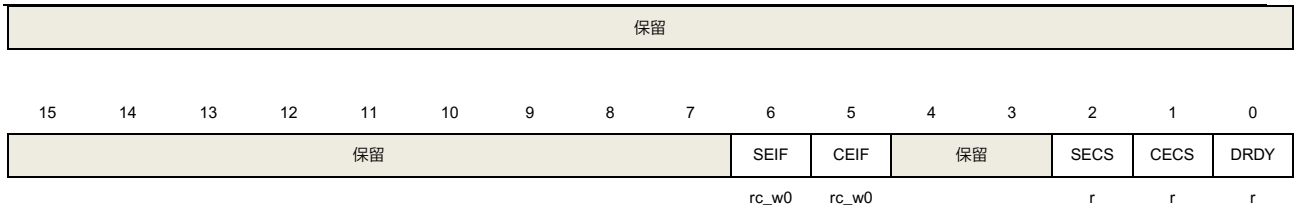
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	NR[1:0]	模拟 rng 电源模式。 00：超低；可能导致弱随机性 01：低 10：中等 11：高
23:4	保留	必须保持复位值。
3	TRNGIE	中断使能位，当 DRDY，SEIF 或 CEIF 位被置位时该位控制生成一个中断。 0：禁止 TRNG 中断 1：使能 TRNG 中断
2	TRNGEN	TRNG 使能位 0：禁止 TRNG 模块 1：使能 TRNG 模块
1:0	保留	必须保持复位值。

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	SEIF	种子错误中断标志位 如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到则此位将置 1。 0：未检测到错误 1：检测到种子错误。写 0 将清除该位
5	CEIF	时钟错误中断标志位 如果 TRNG_CLK 时钟频率低于 HCLK 频率的 1 / 16 时该位被置位。 0：未检测到错误 1：检测到时钟错误。写 0 将清除该位
4:3	保留	必须保持复位值。
2	SECS	种子错误当前状态 0：当前未检测到种子错误。如果 SEIF = 1 和 SECS = 0，说明之前已经检测到种子错误但现在已恢复正常。 1：当前检测到种子错误。如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到时，该位置 1。
1	CECS	时钟错误当前状态 0：当前未检测到时钟错误。如果 CEIF = 1 和 CECS = 0，则意味着之前已检测到时钟错误但现在已恢复正常。 1：当前检测到时钟错误。此时 TRNG_CLK 时钟频率低于 1 / 16 HCLK 频率。
0	DRDY	随机数准备状态位 读 TRNG_DATA 寄存器会清零该位，当一个新的随机数产生时被置位。 0：TRNG 数据寄存器的内容无效 1：TRNG 数据寄存器的内容有效

12.4.3. 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。

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

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

位/位域	名称	描述
31:0	TRNGDATA[31:0]	32 位随机数据

13. 加密处理器(CAU)

13.1. 简介

加密处理单元支持处理 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 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AES加密解密算法；
- 支持电子密码本(ECB)算法；
- 输入与输出支持DMA传输；

AES

- 支持ECB链算法；
- 支持128位、192位或256位密钥；
- 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

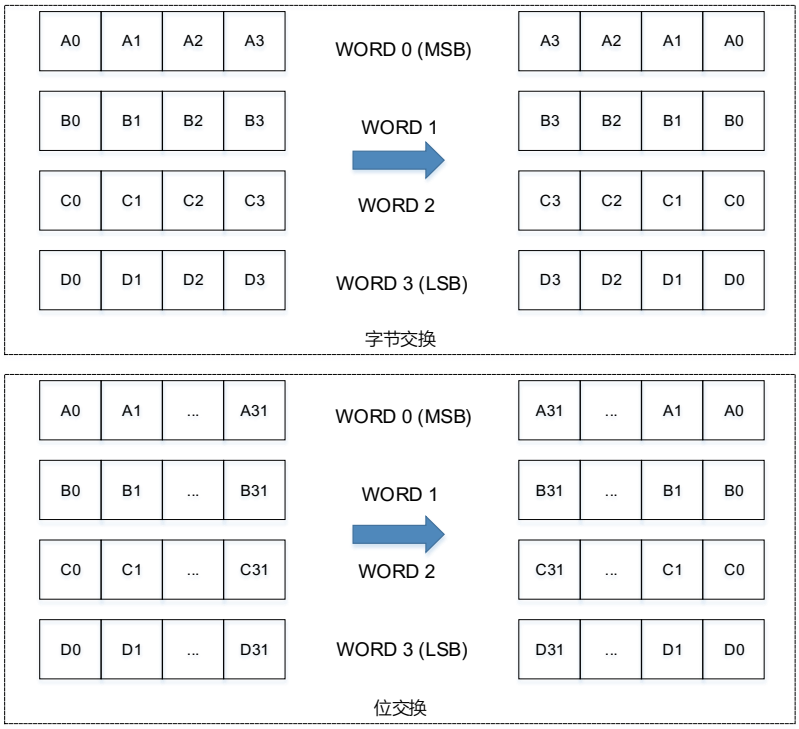
CAU 处理器一次输入 32 位(字)数据， AES 每 128 位对数据流进行处理。对于每个数据块，在其进入 CAU 处理器之前，可对这些数据执行位、字节、半字交换或不交换操作(取决于要加密的数据类型)。在 CAU 数据写入 OUT FIFO 之前，需要对其执行同样的交换操作。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。

*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。

图 13-1. DATAM 不交换/半字交换



图 13-2. DATATM 字节交换/位交换

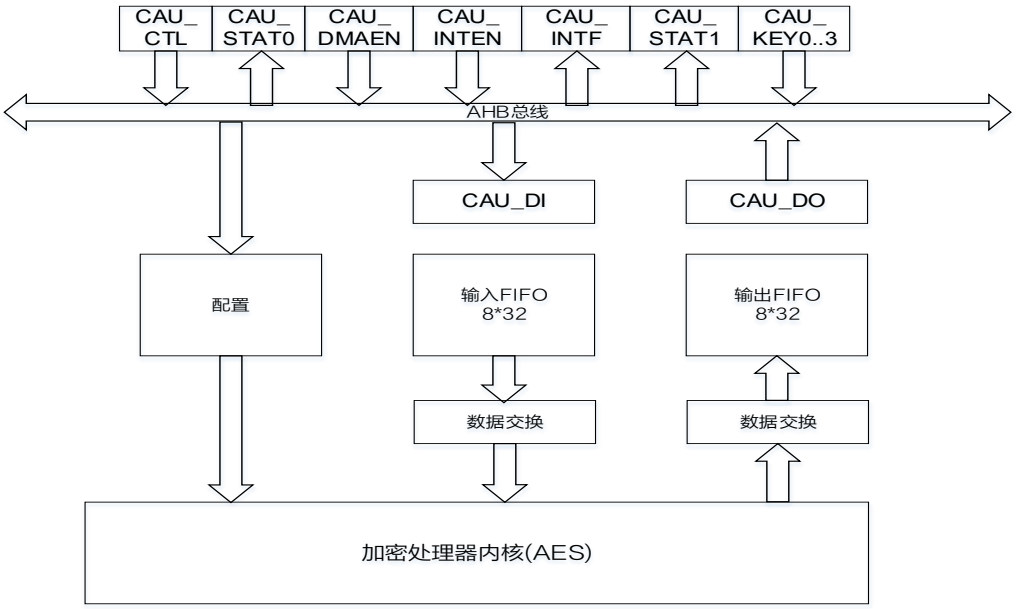


13.4. 加密处理器流程

加密处理器关于 DES 和 AES 加密处理的实现具体请参考章节 *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. 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。

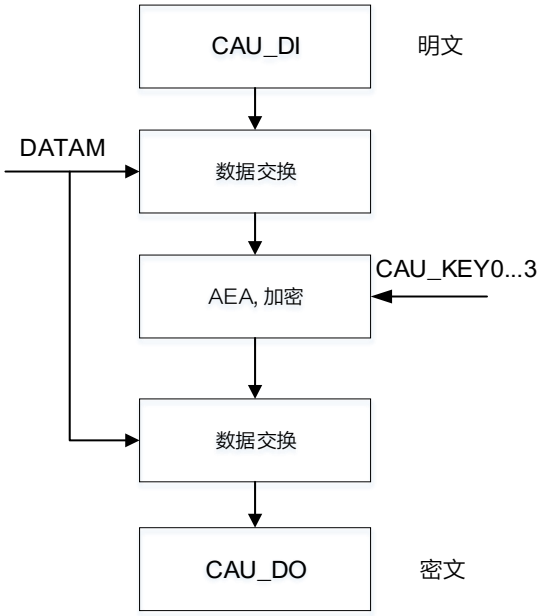
使用 128 位密钥时 AES 密钥为[KEY3 KEY2]，使用 192 位密钥时 AES 密钥为[KEY3 KEY2 KEY1]，使用 256 位密钥时 AES 密钥为[KEY3 KEY2 KEY1 KEY0]。

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 13-4. AES ECB 加密*。

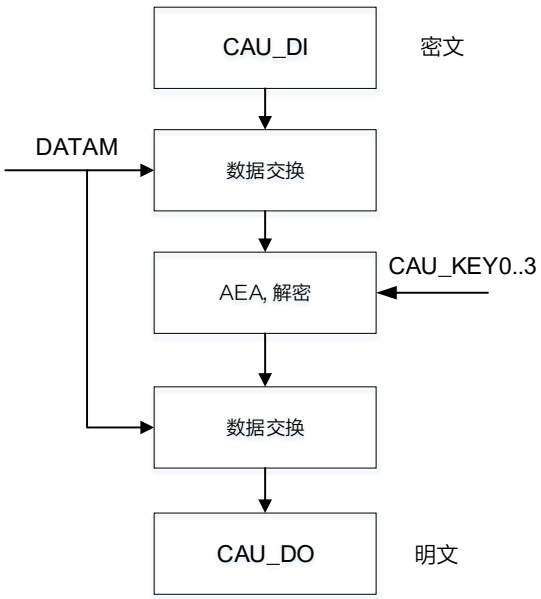
图 13-4. AES ECB 加密



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 13-5. AES ECB 解密*。

图 13-5. AES ECB 解密



13.5. 操作模式

加密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 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. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
4. 设置CAU_CTL寄存器的ALGM[2:0]位，配置算法(AES)和模式(ECB)；
5. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作；
6. 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；
7. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
8. 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；
9. 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；
10. 重复步骤8和步骤9，直到所有的数据块都完成加密。

解密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 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. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
4. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
5. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好；
6. 设置CAU_CTL寄存器的ALGM[2:0]位，配置算法(AES)和模式(ECB)；
7. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作；
8. 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；
9. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
10. 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；
11. 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；
12. 重复步骤10和步骤11，直到所有的数据块都完成解密。

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。

注意：CAU 输出数据 DMA 传输启动后，传输过程中不得清除 CAU_DMAEN 寄存器的 DMAOEN 位或 CAU_CTL 寄存器的 CAUEN 位。若上述任一位被意外清除，DMA 传输将中断；即使随后重新置位相关控制位，传输也无法从中断处继续，需重新完成 CAU 与 DMA 配置后重新启动传输。

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 从而使能了输入 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。

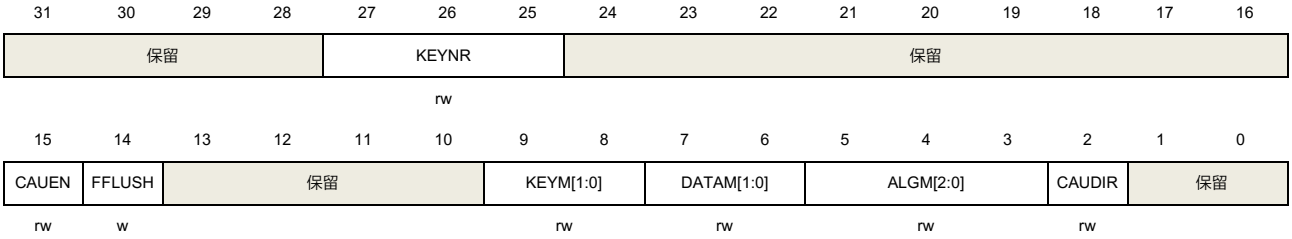
13.8. CAU 寄存器

CAU 基地址：0x4002 3400

13.8.1. 控制寄存器(CAU_CTL)

偏移地址：0x00
复位值：0x0800 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。
27:25	KEYNR	该位字段定义了所选操作的内部密钥编号。 000：选择 k0(来自闪存)作为 AES 密钥 001：选择 k1(来自闪存)作为 AES 密钥 010：选择 k2(来自闪存)作为 AES 密钥 011：选择 k3(来自闪存)作为 AES 密钥 其他：选择 k4(来自寄存器)作为 AES 密钥
24:16	Reserved	必须保持复位值。
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=111b)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 2 100：AES-ECB(AES 电子密码本)，使用 CAU_KEY0, 1, 2, 3. 111：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。 注意：当选择不支持的模式时，将不会执行加密操作。
2	CAUDIR	CAU 算法方向，必须在 BUSY=0 时才可配置 0：加密 1：解密
1:0	保留	必须保持复位值。

13.8.2. 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。

31	30	29	28	27	26	25	24	23	22	21	20	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	OFU	ONE	INF	IEM
											r	r	r	r	r

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

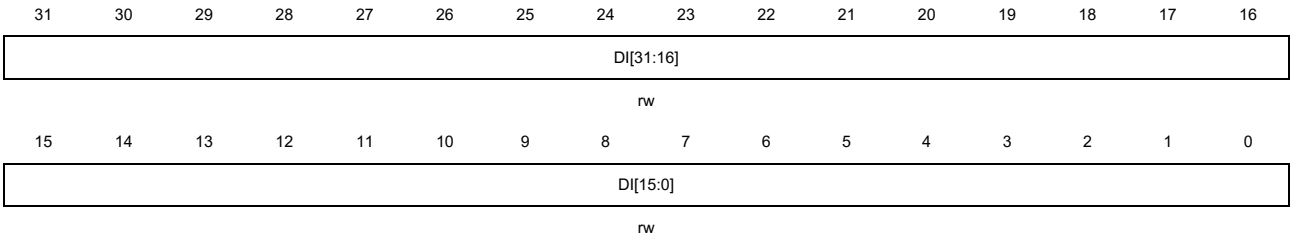
2	ONE	输出 FIFO 非空 0：输出 FIFO 为空 1：输出 FIFO 非空
1	INF	输入 FIFO 未满 0：输入 FIFO 满 1：输入 FIFO 未满
0	IEM	输入 FIFO 空 0：输入 FIFO 非空 1：输入 FIFO 空

13.8.3. 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。



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。

13.8.4. 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

13.8.5. 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

13.8.6. 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

13.8.7. 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

13.8.8. 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

13.8.9. 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 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															

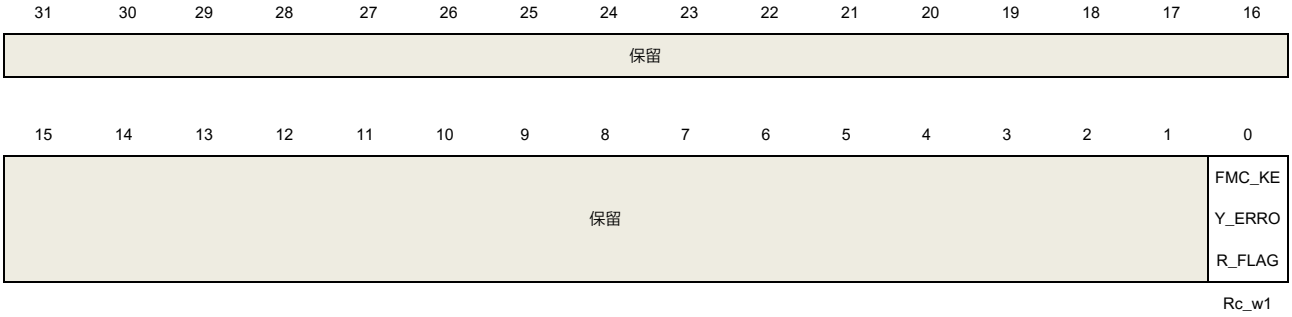
位/位域	名称	描述
31:0	KEY0...3(H/L)	用于 DES 或 TDES 或 AES 的密钥

13.8.10. 错误标志寄存器 (CAU_ERROR_FLAG)

Address offset: 0x100

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:1	保留	必须保持复位值。
0	FMC_KEY_ERROR_FLAG	KEYNR = 000 1)Keysize ==10，256 位全为 0 或全为 FF，使用此密钥并启用 CAUEN 2)Keysize ==01，k0[191:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN 3)Keysize ==00，k0[127:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN KEYNR = 001 1)Keysize ==10，256 位全为 0 或全为 FF，使用此密钥并启用 CAUEN 2)Keysize ==01，k1[191:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN 3)Keysize ==00，k1[127:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN KEYNR = 010 1)Keysize ==10，256 位全为 0 或全为 FF，使用此密钥并启用 CAUEN 2)Keysize ==01，k2[191:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN 3)Keysize ==00，k2[127:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN KEYNR = 011 1)Keysize ==10，256 位全为 0 或全为 FF，使用此密钥并启用 CAUEN 2)Keysize ==01，k3[191:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN 3)Keysize ==00，k3[127:0]全为 0 或全为 FF，使用此密钥并启用 CAUEN

注意：FMC_KEY_ERROR_FLAG 仅在 AES 算法下使用

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-256)。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-256 为 256 位。

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(SHA256)；

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；

14.3. HAU 数据类型

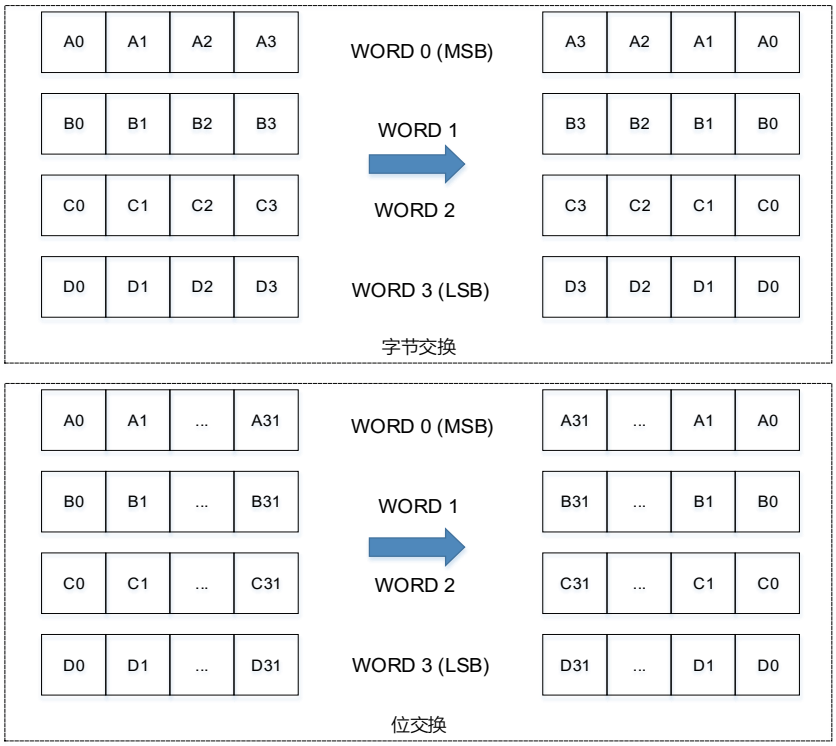
哈希处理器每次接收 32 位字，但每次计算处理一个 512 位块。对每个输入字，在送入哈希内核之前都会根据数据类型进行位/字节/半字/不交换。同样在数据输出之前也要进行相同的数据交换。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。SHA-256 的计算均为大端模式。

*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

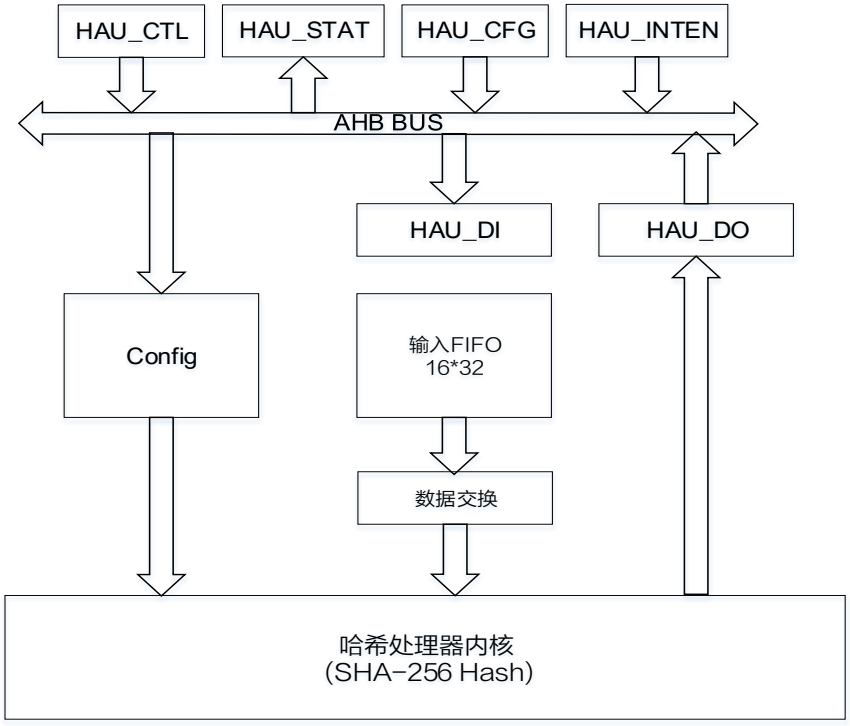


14.4. HAU 内核

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-256 为 256 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

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。

14.5. 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。

14.5.1. 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。

14.5.2. 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。

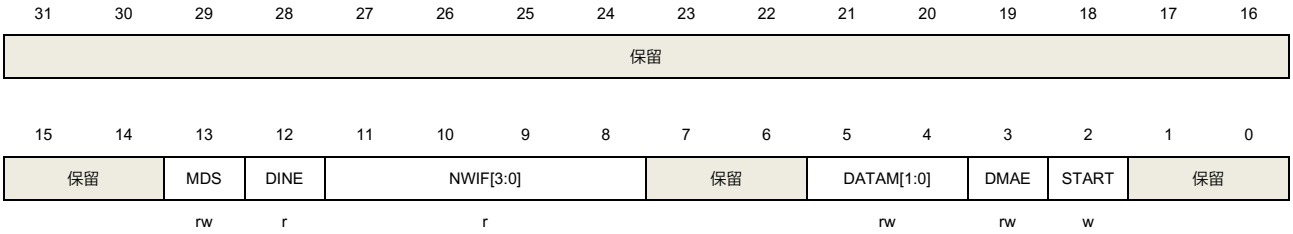
14.6. HAU 寄存器

HAU 基地址：0x4002 3800

14.6.1. 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
31:14	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
7:6	保留	必须保持复位值。
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。

14.6.2. 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。

14.6.3. 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。

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

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

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。

14.6.4. 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。

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

14.6.5. 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

14.6.6. 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)

15. 直接存储器访问控制器(DMA)

15.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。数据可以通过 DMA 在外设和存储器之间以及存储器和存储器之间快速移动，而无需任何 CPU 操作。DMA 控制器中有 12 个通道。每个通道都专门用于管理来自一个或多个外设的存储器访问请求。内部实现了一个仲裁器来处理 DMA 请求之间的优先级。

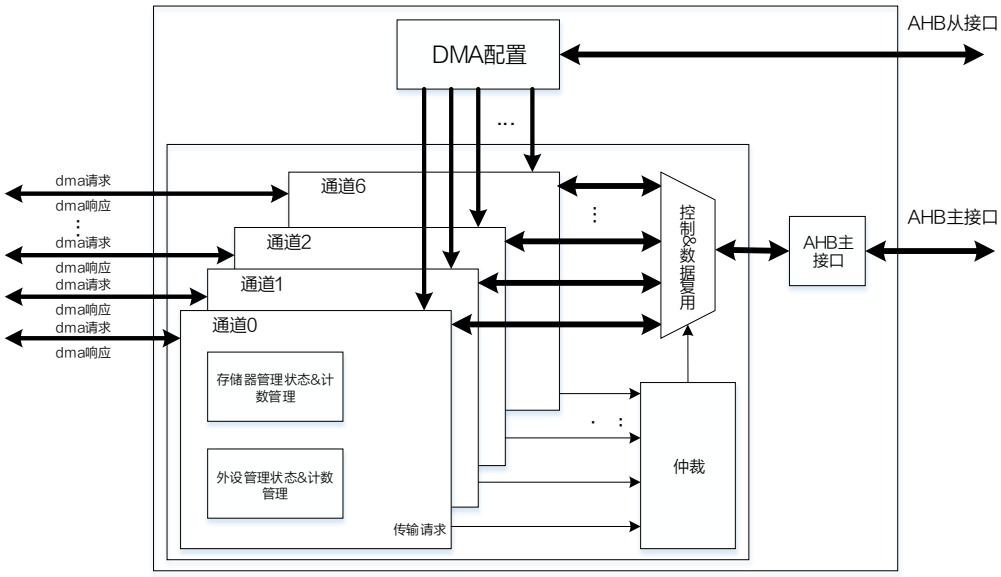
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

15.2. 主要特征

- 传输数据长度可编程配置，最大可达65535。
- 12个通道(DMA0有7个，DMA1有5个)，每个通道都是可配置的。
- AHB和APB外设、FLASH、SRAM可以作为源和目标被访问。
- 软件DMA通道优先级(低、中、高、超高)和硬件DMA通道优先级(DMA通道0具有最高优先级，DMA通道6具有最低优先级)。
- 支持独立的8位、16位、32位存储器和外设传输。
- 支持独立的固定、递增、递减和偏移(仅限通道0)地址生成算法的存储器和外设。
- 支持循环传输模式。
- 支持通道抢占，当当前通道被抢占时，会被设置为待处理状态，并在下一次其优先级最高时运行。
- 支持四种传输模式：
  - 每个请求传输单个数据元素
  - 每个请求一个块传输
  - 每个请求按链表描述符进行传输
  - 每个请求所有链表传输
- 支持重复区域访问
- 支持外设到存储器、存储器到外设以及存储器到存储器的数据传输。
- 每个通道一个独立的中断，共有四种类型的事件标志。
- 支持中断使能和清除。

15.3. 结构框图

图 15-1. DMA 框图



如*图 15-1. DMA 框图*所示，DMA 控制器由四个主要部分组成：

- 通过AHB从站接口配置DMA。
- 通过一个AHB主接口进行数据传输，用于存储器访问和外设访问。
- 内部有一个仲裁器来管理同时到来的多个外设请求。
- 通道管理以控制地址/数据选择和数据计数。

15.4. 功能说明

15.4.1. DMA 操作

每个 DMA 传输由两个操作组成，包括从源加载数据和将加载的数据存储到目标。DMA 控制器根据 DMA_CHxPADDR、DMA_CHxMADDR 和 DMA_CHxCTL1 寄存器中设置的值计算源和目标地址。DMA_CHxCNT 寄存器控制通道上要传输的传输次数。DMA_CHxCTL1 寄存器中的 PWIDTH 和 MWIDTH 位确定一次传输中要传输的字节数。

假设 DMA_CHxCNT 为 4，并且 PNAGA 和 MNAGA 都置位。对于 PWIDTH 和 MWIDTH 的每种组合，DMA 传输操作如下*表 15-1. DMA 传输操作*所示。

表 15-1. DMA 传输操作

传输尺寸		数据传输操作	
源	目标	源	目标
32位	32位	1：读取 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC	1：写入 B3B2B1B0[31:0] @0x0 2：写入 B7B6B5B4[31:0] @0x4 3：写入 BBBAB9B8[31:0] @0x8 4：写入 BFBEBDBC[31:0] @0xC
32位	16位	1：读取 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC	1：写入B1B0[15:0] @0x0 2：写入B5B4[15:0] @0x2 3：写入B9B8[15:0] @0x4 4：写入 BDBC[15:0] @0x6
32位	8位	1：读取 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC	1：写入B0[7:0] @0x0 2：写入B4[7:0] @0x1 3：写入B8[7:0] @0x2 4：写入BC[7:0] @0x3
16位	32位	1：读取B1B0[15:0] @0x0 2：读取B3B2[15:0] @0x2 3：读取B5B4[15:0] @0x4 4：读取 B7B6[15:0] @0x6	1：写入0000B1B0[31:0] @0x0 2：写入0000B3B2[31:0] @0x4 3：写入0000B5B4[31:0] @0x8 4：写入0000B7B6[31:0] @0xC
16位	16位	1：读取B1B0[15:0] @0x0 2：读取B3B2[15:0] @0x2 3：读取B5B4[15:0] @0x4 4：读取B7B6[15:0] @0x6	1：写入B1B0[15:0] @0x0 2：写入B3B2[15:0] @0x2 3：写入B5B4[15:0] @0x4 4：写入B7B6[15:0] @0x6
16位	8位	1：读取B1B0[15:0] @0x0 2：读取B3B2[15:0] @0x2 3：读取B5B4[15:0] @0x4 4：读取B7B6[15:0] @0x6	1：写入B0[7:0] @0x0 2：写入B2[7:0] @0x1 3：写入B4[7:0] @0x2 4：写入B6[7:0] @0x3
8位	32位	1：读取B0[7:0] @0x0 2：读取B1[7:0] @0x1 3：读取B2[7:0] @0x2 4：读取B3[7:0] @0x3	1：写入000000B0[31:0] @0x0 2：写入 000000B1[31:0] @0x4 3：写入000000B2[31:0] @0x8 4：写入000000B3[31:0] @0xC
8位	16位	1：读取B0[7:0] @0x0 2：读取B1[7:0] @0x1 3：读取B2[7:0] @0x2 4：读取B3[7:0] @0x3	1：写入00B0[15:0] @0x0 2：写入00B1[15:0] @0x2 3：写入00B2[15:0] @0x4 4：写入00B3[15:0] @0x6
8位	8位	1：读取B0[7:0] @0x0 2：读取B1[7:0] @0x1 3：读取B2[7:0] @0x2 4：读取B3[7:0] @0x3	1：写入B0[7:0] @0x0 2：写入B1[7:0] @0x1 3：写入B2[7:0] @0x2 4：写入B3[7:0] @0x3

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

通过清除 DMA_CHxCTL0 寄存器中的 CHEN 位来停止 DMA 传输。

清除 CHEN 有两种可能的情况：软件清除 CHEN 和硬件在检测到地址错误后自动清除 CHEN。

- 当软件清除对应 DMA 传输的 CHEN 位时，传输将被禁用。每种传输模式的停止时间如下：
  - TRANSMOD=00：传输在完成单次传输后停止，并置位 FTFIFx 标志。
  - TRANSMOD=01：传输在完成块传输后停止，并置位 FTFIFx 标志。
  - TRANSMOD=10/11：传输在完成链表传输并加载下一个链表后停止，并置位 FTFIFx 标志。
  - 如果清除 CHEN 位时当前通道未完成且没有传输正在进行，FTFIFx 标志将直接置位。
  - 在清除 CHEN 位后，软件可以使用 FTFIFx 标志来确定通道是否已停止传输。只有在传输停止后才能进行配置修改。
- 如果 DMA 传输在要传输的数据中检测到地址错误，它将不会访问错误的地址。硬件清除相应的 CHEN 位以停止传输，并置位 ERRIFx 标志。
- 如果在清除 CHEN 位时 DMA 传输未完成，重新启动此 DMA 通道时可能会出现以下四种情况：
  - 如果在重启 DMA 通道之前没有对通道进行寄存器配置操作，DMA 将继续完成剩余的传输。
  - TRANSMOD=00：如果对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行了任何配置操作，DMA 将重新开始新的传输。
  - TRANSMOD=01：如果对相应通道的 DMA_CHxCNT、DMA_CHxPADDR、DMA_CHxMADDR 或 DMA_CHxBCNT 寄存器进行了任何配置操作，DMA 将重新开始新的传输。
  - TRANSMOD=10/11：如果在重启 DMA 通道之前修改了寄存器配置，DMA 将根据修改后的配置进行传输。需要注意的是，所有相关的配置都需要被修改。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

15.4.2. 链表描述符

当 TRANSMOD = 10 且 ALINK = 1 时，DMA 进入链表传输模式。

在此模式下，当当前描述符定义的传输完成后，DMA 会从存储在 DMA_LADDR 中的链接地址 LADDR 处重新加载通道的配置寄存器。以下寄存器将被重新加载：

DMA_CHxCTL1、DMA_CHxCNT、DMA_CHxPADDR、DMA_CHxMADDR、DMA_CH0PAOFS、DMA_CH0MAOFS。

在重新加载完成之前，通道的 ACK 信号保持非活动状态，因此该通道不会被其他请求抢占。

注意：

- 在使用 LADDR 重新加载 DMA 传输操作寄存器时，软件必须确保 TRANSMOD 模式不被更改。
- 用作不同通道 LADDR(描述符存储地址)的存储器区域不得重叠。
- DMA_CH0PAOFS 和 DMA_CH0MAOFS 寄存器仅适用于通道 0。

表 15-2. DMA 传输操作

寄存器	加载地址
DMA_CHxCTL1	LADDRx[31:0] + 0x00
DMA_CHxCNT	LADDRx[31:0] + 0x04
DMA_CHxPADDR	LADDRx[31:0] + 0x08
DMA_CHxMADDR	LADDRx[31:0] + 0x0C
DMA_CHxLADDR	LADDRx[31:0] + 0x10
DMA_CH0PAOFS	LADDRx[31:0] + 0x14
DMA_CH0MAOFS	LADDRx[31:0] + 0x18

15.4.3. 数据传输模式

DMA 支持由 DMA_CHxCTL1 中的 TRANSMOD 字段设置的四种传输模式。

1. 当 TRANSMOD 设置为 00 时，DMA 在每个 DMA 请求中将单个数据元素从源地址传输到目标地址。如果需要连续传输多个数据元素，则必须生成多个请求。传输的总数量(数据元素)在 DMA_CHxCNT 寄存器中定义。

2. 当 TRANSMOD 设置为 01 时，DMA 在接收到一个请求后，会根据 DMA_CHxCNT 寄存器中定义的值传输全部数据。在此模式下，也可以通过 DMA_CHxBCNT 寄存器启用基于块的传输。DMA_CHxBCNT 寄存器设置位(Number_of_Blocks - 1)，因此实际传输的块数等于 DMA_CHxBCNT + 1，每个块传输完成一个 DMA 握手。在块传输期间，如果当前通道未重新配置，则地址更新遵循 PNAGA 和 MNAGA 的配置规则，而其他配置字段保持初始配置值。当块传输完成时，会触发一个中断。如果 DMA_CHxCTL0 寄存器中的 BOFSMEN 或 BOFSPEN 位被启用，则块间地址偏移由 DMA_CHxPABOFS 或 DMA_CHxMABOFS 确定。

3. TRANSMOD=10 总是与 ALINK 一起使用; 在每个请求上传输 DMA_CHxCNT 指定的全部数据量，如果 ALINK 为 1，DMA 将自动切换到下一个链表描述符，并用一个新的请求继续传输。在此模式下，每个请求完成一个链表描述符传输，当该描述符传输完成时，DMA 发出应答(ACK)，当描述符中的 ALINK 为 0 时，整体传输序列终止。

4. TRANSMOD=11 也与 ALINK 一起使用; 在一次请求中传输 DMA_CHxCNT 指定的全部数据量，如果 ALINK 为 1，DMA 将自动切换到下一个链表描述符，并在不需要新请求的情况下继续传输。在此模式下，一个请求完成所有链表描述符传输，当整个链式传输完成时，DMA 发出应答(ACK)，当描述符中的 ALINK 为 0 时，传输序列终止。

15.4.4. 重复区域访问

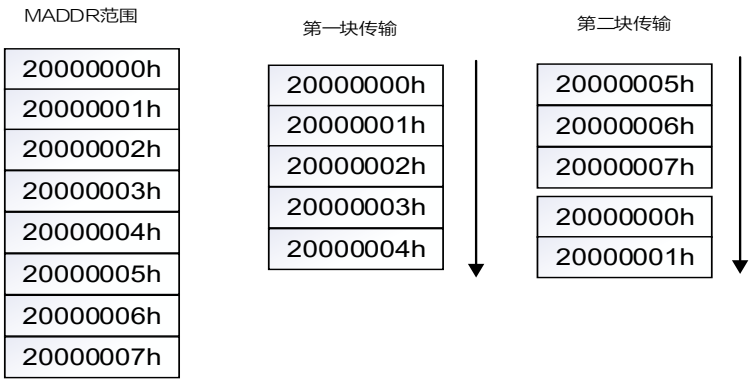
DMA 支持由 RPTAREA 寄存器定义的重复区域访问。为了在保持传输地址寄存器的高位固定

的同时，反复从某个区域读取或写入，用户可以配置 DMA_CHxCNT 寄存器的 RAPA 或 RAPA 位。当地址达到边界时，DMA 会自动返回到起始地址以继续传输。重复区域的大小支持 2 的幂(以字节为单位)。

注意：在块模式下，不能同时使能重复区域和块偏移功能。如果两者都设置为 1，那么只有块偏移功能会被执行。

以下图示展示了重复区域访问的一个例子。TRANSMOD 设置为 2'b01，MARA[4:0]设置为 5'b00011，MWIDTH 设置为 2'b00，BCNT 设置为 16'd1。

图 15-2. 块传输模式下重复地址访问示例。



15.4.5. 外设握手

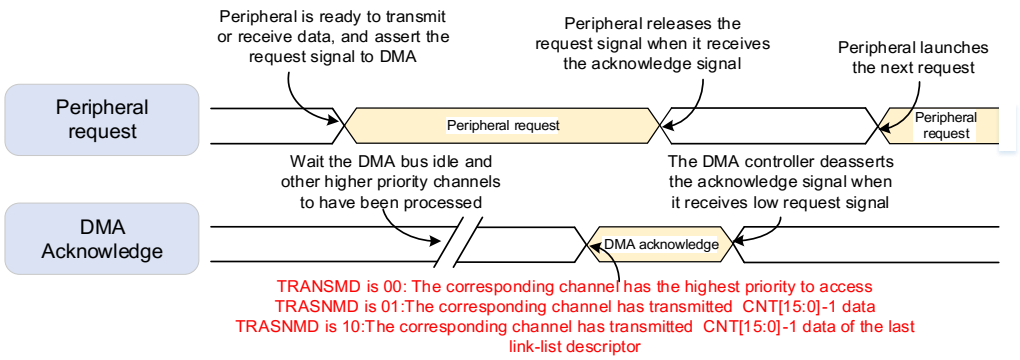
为确保数据有序和高效传输，DMA 与外设之间引入握手机制，包括请求信号和应答信号：

- 外设向 DMA 控制器发出的请求信号，表示外设已准备好发送或接收数据。
- DMA 向外设响应的应答信号，表示 DMA 控制器已启动 AHB 命令以访问外设。

注意：当外设发出请求以触发DMA时，软件必须首先使能相应的DMA通道(通道使能)，然后才能在外设寄存器中使能相关的DMA功能位(例如，SPI)。

*图 15-3. 握手机制*展示了 DMA 控制器与外设之间握手机制的工作方式。

图 15-3. 握手机制



15.4.6. 仲裁

当同时收到两个或更多请求时，仲裁器根据通道的优先级来决定服务哪个请求。存在两级优先级，包括软件优先级和硬件优先级。仲裁器根据以下优先级规则来确定选择哪个通道来响应：

- 软件优先级：通过配置DMA_CHxCTL1寄存器中的PRIO位，包括低、中、高和极高四个等级。
- 对于软件优先级相同的通道，优先级赋予通道号较低的通道。

抢占是可编程的，每个通道都可以通过 DMA_CHxCTL1 寄存器中的 PREN 位来设置。如果使能，高优先级请求将在当前传输完成之前进行仲裁。被抢占的通道将被设置为待处理状态，并将在下一次其优先级最高时运行。如果未能，当前传输不能被中断，直到完成。

15.4.7. 地址生成

四种地址生成算法独立地应用于存储器和外设，包括固定模式、递增模式、递减模式和偏移模式。DMA_CHxCTL1 寄存器中的 PNAGA 和 MNAGA 位用于配置外设和存储器的下一个地址生成算法。

在固定模式下，下一个地址总是等于在基地址寄存器(DMA_CHxPADDR, DMA_CHxMADDR)中配置的基地址。

在递增模式下，下一个地址等于当前地址加上 1、2 或 4，这取决于传输数据的宽度。

在递减模式下，下一个地址等于当前地址减去 1、2 或 4，这取决于传输数据的宽度。

在偏移模式下(仅通道 0 可用)，下一个地址等于当前地址加上 DMA_CH0PAOFS 或 DMA_CH0MAOFS 中的偏移值，该值以二进制补码形式表示，支持正负偏移。

15.4.8. 循环模式

循环模式用于处理连续的外设请求(例如，ADC 扫描模式)。通过设置 DMA_CHxCTL0 寄存器中的 CMEN 位来使能循环模式。

在循环模式下，CNT 位会自动重新加载预设值，并且在每次 DMA 传输结束时断言完全传输完成标志。直到 DMA_CHxCTL0 寄存器中的 CHEN 位被清除，DMA 始终能响应外设请求。

注意：循环模式仅在 TRANSMOD 设置为 00 时有效。

15.4.9. 存储器到存储器模式

存储器到存储器模式通过设置 DMA_CHxCTL1 寄存器中的 M2M 位来启用。在这种模式下，DMA 通道也可以在没有外设请求触发的情况下工作。一旦通过设置 DMA_CHxCTL0 寄存器中的 CHEN 位来使能 DMA 通道，它就会开始传输，并在 DMA_CHxCNT 寄存器达到零时完成。

注意：当 TRANSMOD = 2'b11 时，所有描述符中的 M2M 位必须保持一致(设置为相同的值)。

15.4.10. 通道配置

当开始一个新的 DMA 传输时，建议遵循以下步骤：

1. 读取 CHEN 位并判断通道是否已使能。如果通道已使能，通过软件清除 CHEN 位。当 FTFIFx 位读为'1'时，允许配置和启动新的 DMA 传输。
2. 在 DMA_CHxCTL1 寄存器中配置 TRANSMOD 位、M2M 位和 DIR 位以设置传输模式。
3. 在 DMA_CHxCTL0 寄存器中配置 CMEN 位以使能/失能循环模式。
4. 在 DMA_CHxCTL1 寄存器中配置 PRIO 位以设置通道软件优先级。
5. 在 DMA_CHxCTL1 寄存器中配置 MWIDTH、PWIDTH、MNAGA 和 PNAGA，以设置存储器和外设传输宽度以及存储器和外设地址生成算法。如果选择了偏移地址模式，则配置 DMA_CH0PAOFS 和 DMA_CH0MAOFS 以设置地址偏移。
6. 在 DMA_CHxCTL0 寄存器中配置 FTFIE、HTFIE、ERRIE 和 BTFIE，以启用完全传输完成中断、半传输完成中断、传输错误中断和块传输完成中断的使能位。
7. 配置 DMA_CHxPADDR 寄存器以设置外设基地址。
8. 配置 DMA_CHxMADDR 寄存器以设置存储器基地址。
9. 配置 DMA_CHxCNT 寄存器中的 CNT 位、RASEL 位、RAPA 位和 RAMA 位，以设置块传输数据数量和重复区域。
10. 如果 TRANSMOD 为 01，通过设置 DMA_CHxCTL0 寄存器中的 BOFSMEN 和 BOFSPEN 使能块地址偏移，并在 DMA_CHxPABOFS 和 DMA_CHxMABOFS 寄存器中配置块地址偏移值。
11. 如果 TRANSMOD 为 10/11，配置 DMA_CHxCTL1 寄存器中的 ALINK 位和 DMA_CHxLADDR 寄存器，以设置下一个描述符地址并使能自动获取。
12. 将 DMA_CHxCTL0 寄存器中的 CHEN 位配置为'1'以使能该通道。

15.4.11. 中断

每个 DMA 通道都有一个专用的中断。中断事件有四种类型，包括完整传输完成、半传输完成、传输错误和块传输完成。

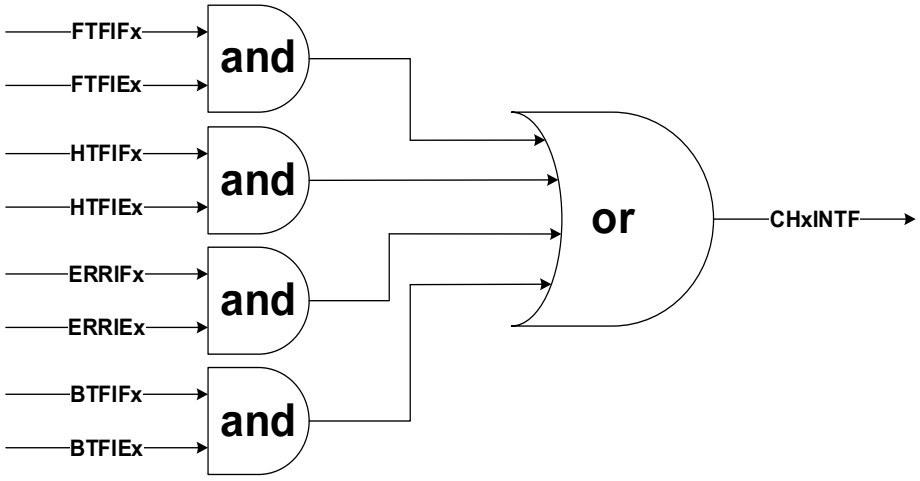
每个中断事件在 DMA_INTF 寄存器中都有一个专用的标志位，在 DMA_INTC 寄存器中都有一个专用的清除位，以及在 DMA_CHxCTL0 寄存器中都有一个专用的使能位。这些关系在下文中进行了描述。

表 15-3. 中断事件

中断事件	标志位	清除位	使能位	模式
	DMA_INTF	DMA_INTC	DMA_CHxCTL0	TRANSMOD
完整传输完成	FTFIF	FTFIFC	FTFIE	00/01/10/11
半传输完成	HTFIF	HTFIFC	HTFIE	00/01
传输错误	ERRIF	ERRIFC	ERRIE	00/01/10/11
块传输完成	BTFIF	BTFIFC	BTFIE	01/10/11

*图 15-4. DMA 中断逻辑*显示了 DMA 中断逻辑。当通道上发生任何类型的中断事件并被使能时，可以产生中断。

图 15-4. DMA 中断逻辑



注意："x"表示通道编号(DMA0 对应于 x=0...6；DMA1 对应于 x=0...4)。

15.4.12. DMA 请求映射

一个通道的 DMA 请求来自 AHB/APB 外设，通过 DMAMUX 请求复用器的相应通道输出，参考 ***DMA 请求多路复用器(DMAMUX)***。

15.5. 寄存器定义

DMA0 基地址：0x4002 0000

DMA1 基地址：0x4002 0400

DMA 寄存器必须通过字(32 位)来访问。

注意：对于拥有5个通道的DMA1，以下寄存器中所有与通道5和通道6相关的位不适用于DMA1。

15.5.1. 中断标志寄存器 0(DMA_INTF0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		BTFIF5	ERRIF5	HTFIF5	FTFIF5	GIF5	BTFIF4	ERRIF4	HTFIF4	FTFIF4	GIF4	BTFIF3	ERRIF3	HTFIF3	FTFIF3
		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
GIF3	BTFIF2	ERRIF2	HTFIF2	FTFIF2	GIF2	BTFIF1	ERRIF1	HTFIF1	FTFIF1	GIF1	BTFIF0	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:30	保留	必须保持复位值。
4+5 *x	BTFIFx	通道x(x=0…5)的块传输完成标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0: 通道x上的块传输尚未完成 1: 通道x上的块传输已完成
3+5*x	ERRIFx	通道x(x=0…5)的错误标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0: 通道x上未发生传输错误 1: 通道x上发生了传输错误。
2+5*x	HTFIFx	通道x(x=0…5)的半传输完成标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上的传输数量未完成一半。 1: 通道x上的传输数量的一半已完成
1+5*x	FTFIFx	通道x(x=0…5)的全传输完成标志。 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上的传输尚未完成。 1：通道x上的传输已完成。

5*x	GIFx	通道x(x=0…5)的全局中断标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上没有发生ERRIF、HTFIF或FTFIF。 1：至少ERRIF、HTFIF或FTFIF中的一个在通道x上发生。

15.5.2. 中断标志寄存器 1(DMA_INTF1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											BTFIF6	ERRIF6	HTFIF6	FTFIF6	GIF6
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	BTFIFx	通道x(x=6)的块传输完成标志。 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上的块传输尚未完成 1：通道x上的块传输已完成
3	ERRIFx	通道x(x=6)的错误标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上未发生传输错误 1：通道x上发生了传输错误。
2	HTFIFx	通道x(x=6)的半传输完成标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上的传输数量未完成一半。 1：通道x上的传输数量的一半已完成
1	FTFIFx	通道x(x=6)的全传输完成标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上的传输尚未完成。 1：通道x上的传输已完成。
0	GIFx	通道x(x=6)的全局中断标志 由硬件设置并通过配置DMA_INTC寄存器由软件清除。 0：通道x上没有发生ERRIF、HTFIF或FTFIF。

1：至少ERRIF、HTFIF或FTFIF中的一个在通道x上发生。

15.5.3. 中断标志清除寄存器 0(DMA_INTC0)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		BTFIFC5	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	BTFIFC4	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4	BTFIFC3	ERRIFC3	HTFIFC3	FTFIFC3
		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
GIFC3	BTFIFC2	ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	BTFIFC1	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	BTFIFC0	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:30	保留	必须保持复位值。
4+5 *x	BTFIFCx	清除通道x(x=0…5)的块传输完成标志位。 0：无影响 1：清除块传输结束标志
3+5*x	ERRIFCx	清除通道x(x=0…5)的错误标志位 0：无影响 1：清除错误标志
2+5*x	HTFIFCx	清除通道x(x=0…5)的半传输完成标志位 0：无影响 1：清除半传输完成标志
1+5*x	FTFIFCx	清除通道x(x=0…5)的完整传输完成标志位 0：无影响 1：清除完整传输完成标志
5*x	GIFCx	清除通道x(x=0…5)的全局中断标志 0：无影响 1：清除DMA_INTF寄存器中的GIFx、ERRIFx、HTFIFx、FTFIFx和BTFIFx位

15.5.4. 中断标志清除寄存器 1(DMA_INTC1)

地址偏移：0x10
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											BTFIFC6	ERRIFC6	HTFIFC6	FTFIFC6	GIFC6
											w	w	w	w	w

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	BTFIFCx	清除通道x(x=6)的块传输完成标志位 0：无影响 1：清除块传输完成标志
3	ERRIFCx	清除通道x(x=6)的错误标志位 0：无影响 1：清除错误标志
2	HTFIFCx	清除通道x(x=6)的半传输完成标志位 0：无影响 1：清除半传输完成标志
1	FTFIFCx	清除通道x(x=6)的完整传输完成标志位 0：无影响 1：清除完整传输完成标志
0	GIFCx	清除通道x(x=6)的全局中断标志 0：无影响 1：清除DMA_INTF寄存器中的GIFx、ERRIFx、HTFIFx、FTFIFx和BTFIFx位

15.5.5. 通道 x 控制寄存器 0(DMA_CHxCTL0)

x = 0...6，其中 x 是通道号。
地址偏移：0x18 + 0x10 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTFIE	保留									CMEN	保留	ERRIE	HTFIE	FTFIE	CHEN
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	BOFSMEN	使能存储器块偏移 当CHEN为'1'时，该位不能被写入。 0: 失能 1: 使能
24	BOFSPEN	使能外设偏移块 当CHEN为'1'时，该位不能被写入。 0: 失能 1: 使能
23:16	保留	必须保持复位值。
15	BTFIE	块传输完成中断的使能位 软件设置和清除 0：失能通道块传输完成中断 1：使能通道块传输完成中断
14:6	保留	必须保持复位值。
5	CMEN	使能循环模式 软件设置和清除 0: 失能循环模式 1: 使能循环模式 当CHEN为'1'时，该位不能被写入。
4	保留	必须保持复位值。
3	ERRIE	使能通道错误中断的位 软件设置和清除 0: 失能通道错误中断 1: 使能通道错误中断
2	HTFIE	使能通道半传输完成中断的位 软件设置和清除 0: 失能通道半传输完成中断 1: 使能通道半传输完成中断
1	FTFIE	使能通道全传输完成中断的位 软件设置和清除 0: 失能通道全传输完成中断 1: 使能通道全传输完成中断
0	CHEN	通道使能 软件设置和清除 0：失能通道 1：使能通道

15.5.6. 通道 x 计数器寄存器(DMA_CHxCNT)

x = 0...6，其中 x 是通道号。

地址偏移：0x1C + 0x10 × x

复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:26	RASEL[1:0]	重复区域选择 00：无重复区域 01：外设重复区域访问 10：存储器重复区域访问 11: 外设和存储器重复区域访问 当CHEN为'1'时，这些位不能被写入。
25:21	RAPA[4:0]	外设地址的重复区域 5'b00000：无重复区域 5'b00001：由PADDR的最低1位指定的2字节作为扩展重复区域 5'b00010：由PADDR的最低2位指定的4字节作为扩展重复区域。 5'b00011：由PADDR的低3位指定的8字节扩展重复区域 5'b00100：由PADDR的低4位指定的16字节扩展重复区域。 …… 5'b11010：由PADDR的低26位指定的64M字节扩展重复区域。 5'b11011：由PADDR的低27位指定的128M字节扩展重复区域。 当CHEN为'1'时，这些位不能被写入。
20:16	RAMA[4:0]	重复的存储器地址区域 5'b00000：无重复区域 5'b00001：由MADDR的最低1位指定的2字节扩展重复区域 5'b00010：由MADDR的最低2位指定的4字节扩展重复区域 5'b00011：由MADDR的最低3位指定的8字节扩展重复区域 5'b00100：由MADDR的最低4位指定的16字节扩展重复区域 …… 5'b11010：由MADDR的最低26位指定的64M字节扩展重复区域 5'b11011：由MADDR的最低27位指定的128M字节扩展重复区域。

当CHEN为'1'时，这些位不能被写入。

15:0	CNT[15:0]	传输计数器 当CHEN为'1'时，这些位不能被写入。 这个寄存器指示剩余的传输次数。一旦通道被使能，它就是只读的，并且在每次DMA传输后递减。如果寄存器值被写零，硬件会将其置为1，实际的传输计数为1。一旦通道的传输完成，如果通道被配置在循环模式下，寄存器可以自动重新加载为之前编程的值。

15.5.7. 通道 x 外设基地址寄存器(DMA_CHxPADDR)

x = 0...6，其中 x 是通道号。
地址偏移：0x20 + 0x10 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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 当CHEN为'1'时，这些位不能被写入。 当PWIDTH为01(16位)时，这些位的最低位(LSB)被忽略。访问自动对齐到半字地址。 当PWIDTH为10(32位)时，这些位的两个最低有效位被忽略。访问自动对齐到字地址。

15.5.8. 通道 x 存储器基地址寄存器(DMA_CHxMADDR)

x = 0...6，其中 x 是通道号。
地址偏移：0x24 + 0x10 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

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

位/位域	名称	描述

31:0	MADDR[31:0]	存储器基地址 当CHEN为'1'时，这些位不能被写入。 当MWIDTH为01(16位)时，这些位的最低有效位(LSB)被忽略。访问自动对齐到半字地址。 当MWIDTH为10(32位)时，这些位的两个最低有效位(LSBs)被忽略。访问自动对齐到字地址。

15.5.9. 通道 0 外设地址偏移寄存器(DMA_CH0PAOFS)

地址偏移：0x118
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

位/位域	名称	描述
31:0	PAOFS[31:0]	当PNAGA为11时的Paddr偏移值。 0000 0000h 至 00FF FFFFh(0字节至16M-1字节) FF00 0000h 到 FFFF FFFFh(-16M字节到-1字节) 当CHEN为'1'时，这些位不能被写入。

15.5.10. 通道 0 存储器地址偏移寄存器(DMA_CH0MAOFS)

地址偏移：0x11C
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

位/位域	名称	描述
31:0	MAOFS[31:0]	当MNAGA为11时的Maddr偏移值。 0000 0000h 至 00FF FFFFh(0字节至16M-1字节) FF00 0000h 到 FFFF FFFFh(-16M字节到-1字节)

当CHEN为'1'时，这些位不能被写入。

15.5.11. 通道 x 链表地址寄存器(DMA_CHxLADDR)

x = 0...6，其中 x 是通道号。
地址偏移：0x120 + 0x08 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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]	链表模式下的描述符地址 当CHEN为'1'时，这些位不能被写入。

15.5.12. 通道 x 块计数寄存器(DMA_CHxBCNT)

x = 0...6，其中 x 是通道号。
地址偏移：0x124 + 0x08 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	BCNT[15:0]	DMA传输的块编号是BCNT[15:0] + 1。 当CHEN为'1'时，这些位不能被写入。

15.5.13. 通道 x 外设块偏移寄存器(DMA_CHxPABOFS)

y = 0...6，其中 y 是支持块偏移功能的通道数量。
地址偏移：0x1D8 + 0x08 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

位/位域	名称	描述
31:0	PABOFS[31:0]	当BOFSPEN为1时，块之间的PADDR偏移值。 0000 0000h 到 00FF FFFFh(0字节到(16M - 1)字节) FF00 0000h 到 FFFF FFFFh(-16M字节到-1字节) 当CHEN为'1'时，这些位不能被写入。

15.5.14. 通道 x 存储器块偏移寄存器(DMA_CHxMABOFS)

y = 0...6，其中 y 是支持块偏移功能的通道数量。
地址偏移：0x1DC + 0x08 × y
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

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

位/位域	名称	描述
31:0	MABOFS[31:0]	当BOFSMEN为1'b1时，块之间的MADDR偏移值。 0000 0000h 到 00FF FFFFh(0字节到(16M - 1)字节) FF00 0000h 到 FFFF FFFFh(-16M字节到-1字节) 当CHEN为'1'时，这些位不能被写入。

15.5.15. 通道 x 控制 1 寄存器(DMA_CHxCTL1)

x = 0...6，其中 x 是通道号。
地址偏移：0x258 + 0x04 × x
复位值：0x0000 0000

该寄存器必须通过字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									ALINK	TRANSMOD[1:0]		MNAGA[1:0]		PNAGA[1:0]	
									rw	rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		PREN	保留		DIR	保留			
	rw	rw		rw		rw		rw			rw				

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	ALINK	自动翻转到下一个描述符使能 当CHEN为'1'时，该位不能被写入。 0: 失能 1: 使能
21:20	TRANSMOD[1:0]	数据传输模式 00: 一次请求触发一次数据传输 01: 一次请求触发DMA_CHxCNT上所有数据传输 10: 一次请求触发一个链表描述符传输 11: 一次请求触发所有链表描述符传输 当CHEN为'1'时，该位不能被写入。
19:18	MNAGA[1:0]	存储器的下一个地址生成算法 软件设置和清除 00：固定地址模式 01：递增地址模式 10：递减地址模式 11：偏移地址模式 当CHEN为'1'时，该位不能被写入。
17:16	PNAGA[1:0]	外设的下一个地址生成算法 软件设置和清除 00：固定地址模式 01：递增地址模式 10：递减地址模式 11：偏移地址模式 当CHEN为'1'时，该位不能被写入。

15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件设置和清除 0：失能存储器到存储器模式 1：使能存储器到存储器模式 当CHEN为'1'时，该位不能被写入。
13:12	PRIO[1:0]	优先级 软件设置和清除 00: 低 01: 中等 10: 高 11: 超高 当CHEN为'1'时，这些位不能被写入。
11:10	MWIDTH[1:0]	存储器传输数据大小 软件设置和清除 00: 8位 01: 16位 10: 32位 11: 保留 当CHEN为'1'时，这些位不能被写入。
9:8	PWIDTH[1:0]	外设传输数据大小 软件设置和清除 00: 8位 01: 16位 10: 32位 11: 保留 当CHEN为'1'时，这些位不能被写入。
7	PREN	抢占使能 0: 在数据传输完成前禁用抢占 1: 当有更高优先级请求到达时使能抢占 当CHEN为'1'时，这些位不能被写入。
6:5	保留	必须保持复位值。
4	DIR	传输方向 软件设置和清除 0: 从外设读取并写入存储器 1: 从存储器读取并写入外设 当CHEN为'1'时，该位不能被写入。
3:0	保留	必须保持复位值。

16. DMA 请求多路复用器(DMAMUX)

16.1. 简介

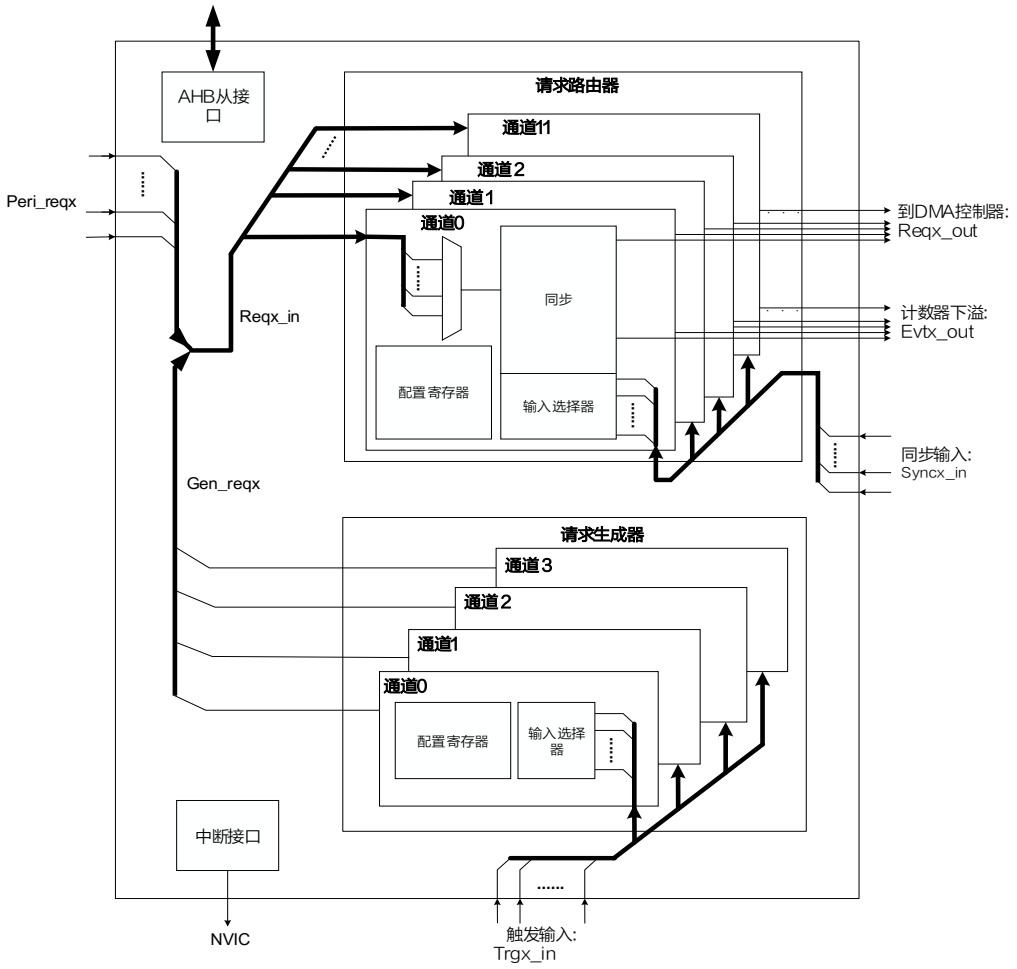
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号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

16.2. 主要特征

- 12 个可配置的 DMAMUX 请求路由输出通道；
- 4 个 DMAMUX 请求生成通道；
- 20 路触发输入信号到 DMAMUX 请求生成器；
- 20 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
  - 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 127 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
  - 一个指示被选中的同步输入信号的事件溢出标志

16.3. 结构框图

图 16-1. DMAMUX 结构框图



16.4. 功能说明

如*图 16-1. 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。

16.4.1. DMAMUX 信号

表 16-1. DMAMUX 信号

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

16.4.2. 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 16-3. 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

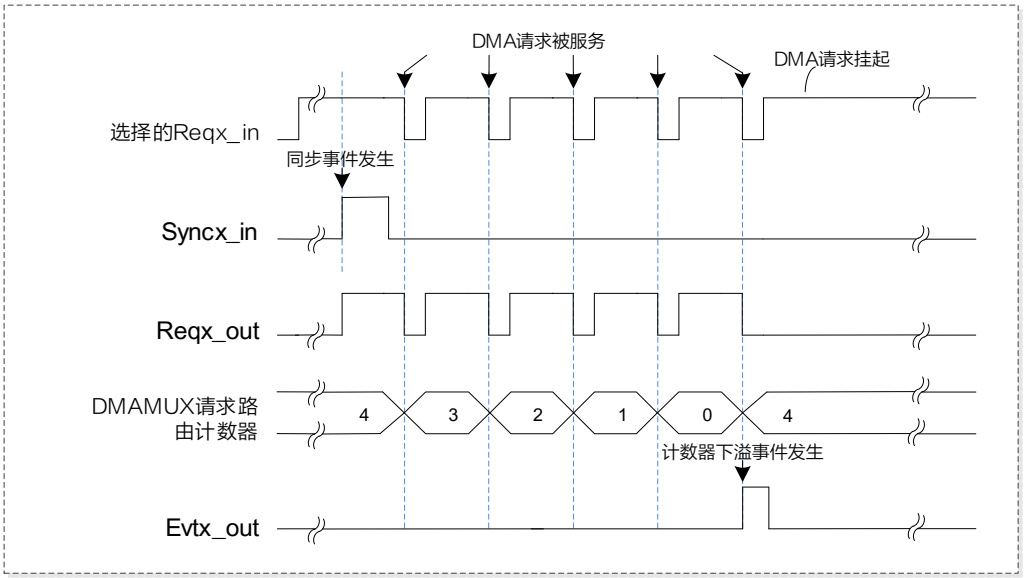
EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

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 16-2. 同步模式*为当 NBR[4:0]=4，SYNCEN=1，EVGEN=1，SYNCP[1:0]=0b01 时的举例。

图 16-2. 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 16-5. 同步输入信号映射*。同步输入信号的有效边沿由 DMAMUX_RM_CHxCFG 寄存器的 SYNCP[1:0]位域来配置。

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

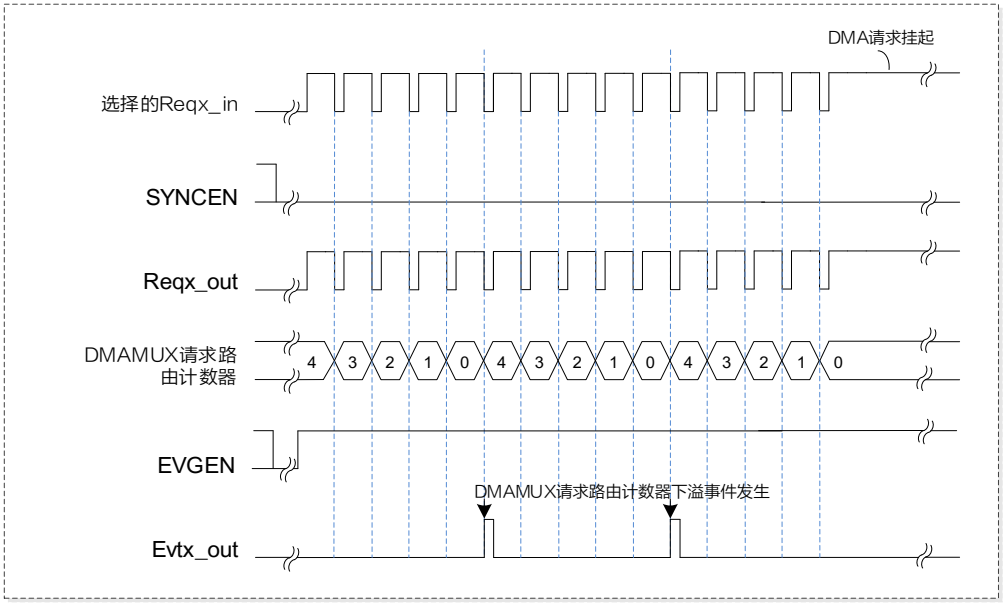
通道事件输出

每个 DMAMUX 请求路由通道都有一个通道事件输出信号 Evtx_out，用于 DMAMUX 请求路由计数器的下溢事件输出。Evt0_out ~ Evt3_out 信号可用于 DMA 请求级联。如果通过置位 DMAMUX_RM_CHxCFG 寄存器的 EVGEN 位来使能 DMAMUX 请求路由通道 x 的通道事件

输出，当 DMAMUX 请求路由计数器自动重装载为 NBR[4:0]值时，发生一个通道事件，输出为一个 AHB 时钟周期脉冲。

*图 16-3. 通道事件输出*为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 16-3. 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。

16.4.3. DMAMUX 请求生成器

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

触发输入信号的有效边沿由 DMAMUX_RG_CHxCFG 寄存器的 RGTP[1:0]位域来配置。DMAMUX 请求生成通道 x 的触发输入信号由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 16-4. 触发输入信号映射*。置位 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位来使能 DMAMUX 请求生成通道 x。

DMAMUX 请求生成通道

当发生触发输入事件时，对应的 DMAMUX 请求生成通道 x 开始产生 DMA 请求到通道的输出上，通道输出连到 DMAMUX 请求路由器的输入上。每当 DMAMUX 生成的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求生成计数器将减 1。当 DMAMUX 请求生成计数器发生下溢时，DMAMUX 请求生成通道将停止产生 DMA 请求，在下一个触发输入事件发生时，DMAMUX 请求生成计数器将自动重装载为 DMAMUX_RG_CHxCFG 寄存器的 NBRG[4:0]位域值。

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

触发溢出

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

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

16.4.4. 通道配置

根据以下步骤来配置 DMAMUX 的通道 y 和对应的 DMA 通道 x：

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

16.4.5. 中断

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

每个中断事件都有一个专用的标志位，专用的清除位和专用的使能位。*表 16-2. 中断事件*描述了其对应关系。

表 16-2. 中断事件

中断事件	标志位	清除位	使能位
DMAMUX 请求路由通道 x 上的同步溢出事件	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。

16.4.6. DMAMUX 映射

DMAMUX 与 DMA0 和 DMA1 配合使用。DMAMUX 的通道 0 到 6 与 DMA0 的通道 0 到 6 相连，DMAMUX 的通道 7 到 11 与 DMA1 的通道 0 到 4 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 16-3. DMAMUX 请求路由输入信号映射*，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[7:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 16-3. DMAMUX 请求路由输入信号映射

请求路由通道输入标识 MUXID[7:0]	来源
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC0_ROUTINE
6	ADC0_INSERTED
7	DAC0_CH0

请求路由通道输入标识 MUXID[7:0]	来源
8	TIMER5_UP
9	TIMER6_UP
10	SPI0_RX
11	SPI0_TX
12	SPI1_RX
13	SPI1_TX
14	SPI2_RX
15	SPI2_TX
16	I2C0_RX
17	I2C0_TX
18	I2C1_RX
19	I2C1_TX
20	USART0_RX
21	USART0_TX
22	USART1_RX
23	USART1_TX
24	USART2_RX
25	USART2_TX
26	UART3_RX
27	UART3_TX
28	TIMER0_CH0
29	TIMER0_CH1
30	TIMER0_CH2
31	TIMER0_CH3
32	TIMER0_UP
33	TIMER0_TI
34	TIMER0_CMT
35	TIMER7_CH0
36	TIMER7_CH1
37	TIMER7_CH2
38	TIMER7_CH3
39	TIMER7_UP
40	TIMER7_TI
41	TIMER7_CMT
42	TIMER1_CH0
43	TIMER1_CH1
44	TIMER1_CH2
45	TIMER1_CH3
46	TIMER1_UP
47	TIMER1_TI

请求路由通道输入标识 MUXID[7:0]	来源
48	TIMER2_CH0
49	TIMER2_CH1
50	TIMER2_CH2
51	TIMER2_CH3
52	TIMER2_UP
53	TIMER2_TI
54	TIMER3_CH0
55	TIMER3_CH1
56	TIMER3_CH2
57	TIMER3_CH3
58	TIMER3_UP
59	TIMER3_TI
60	TIMER4_CH0
61	TIMER4_CH1
62	TIMER4_CH2
63	TIMER4_CH3
64	TIMER4_UP
65	TIMER4_TI
66	ADC1_ROUTINE
67	ADC1_INSERTED
68	ADC2_ROUTINE
69	ADC2_INSERTED
70	Reserved
71	TIMER19_CH0
72	TIMER19_CH1
73	TIMER19_CH2
74	TIMER19_CH3
75	TIMER19_UP
76	TIMER19_TI
77	TIMER19_CMT
78	TIMER14_CH0
79	TIMER14_CH1
80	TIMER14_CH0N
81	TIMER14_UP
82	TIMER14_TI
83	TIMER14_CMT
84	TIMER40_CH0
85	TIMER40_CH1
86	TIMER40_CH0N
87	TIMER40_UP

请求路由通道输入标识 MUXID[7:0]	来源
88	TIMER40_TI
89	TIMER40_CMT
90	TIMER41_CH0
91	TIMER41_CH1
92	TIMER41_CH0N
93	TIMER41_UP
94	TIMER41_TI
95	TIMER41_CMT
96	TIMER42_CH0
97	TIMER42_CH1
98	TIMER42_CH0N
99	TIMER42_UP
100	TIMER42_TI
101	TIMER42_CMT
102	TIMER43_CH0
103	TIMER43_CH1
104	TIMER43_CH0N
105	TIMER43_UP
106	TIMER43_TI
107	TIMER43_CMT
108	TIMER44_CH0
109	TIMER44_CH1
110	TIMER44_CH0N
111	TIMER44_UP
112	TIMER44_TI
113	TIMER44_CMT
114	TIMER50_UP
115	TIMER51_UP
116	CAU_IN
117	CAU_OUT
118	HAU_IN
119	SPI3_RX
120	SPI3_TX
121	USART5_RX
122	USART5_TX
123	OSPI_TC
124	OSPI_FT
125	SDIO
126	DAC0_CH1
127	I2C2_RX

请求路由通道输入标识 MUXID[7:0]	来源
128	I2C2_TX
129	I2S1_RX
130	I2S1_TX
131	I2S2_RX
132	I2S2_TX

触发输入映射

DMAMUX 请求生成通道 x 的触发输入可由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 16-4. 触发输入信号映射*。

表 16-4. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT

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 16-5. 同步输入信号映射*。

表 16-5. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT

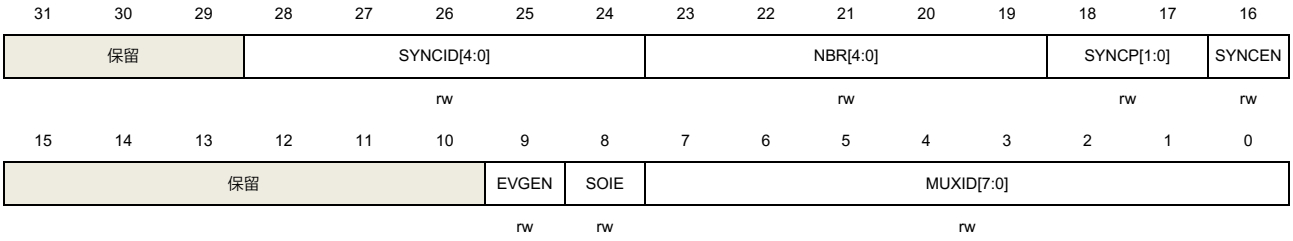
16.5. DMAMUX 寄存器

DMAMUX 基地址：0x4002 0800

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

x = 0...11，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。

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

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFx	请求路由通道x的同步溢出事件标志位(x=0..11) 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。

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

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

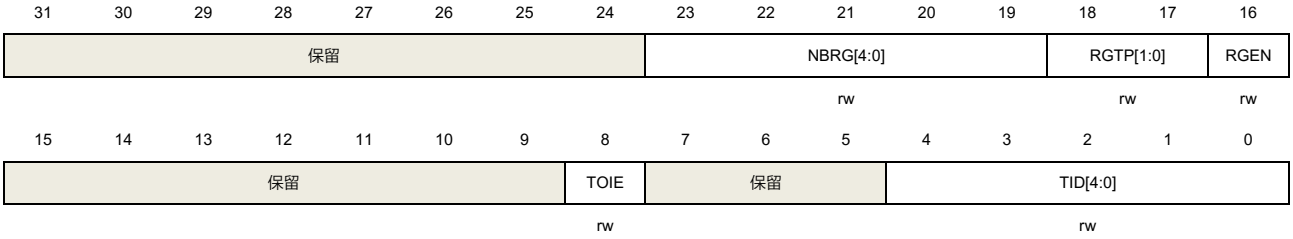
位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFCx	请求路由通道x的同步溢出事件标志清除位(x=0..11)

写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIFx。

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

x = 0...3，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
7:5	保留	必须保持复位值。
4:0	TID[4:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

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

地址偏移：0x140
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TOIFx	DMAMUX请求生成通道x的触发溢出标志位(x=0..3) 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。

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

地址偏移：0x144
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

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

17. 调试(DBG)

17.1. 简介

GD32E53x 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 控制器实现。调试和跟踪功能集成在 Arm® Cortex®-M33 内核中。调试系统支持串行(SW)调试和跟踪功能，也支持 JTAG 调试。调试和跟踪功能请参考下列文档：

- Cortex®-M33技术参考手册；
- Arm调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试一些外设。当相应的位被置 1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、CAN、WWDGT、FWDGT 和 RTC。

17.2. 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。

17.2.1. 切换 JTAG/SW 接口

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

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

切换 SW 调试到 JTAG 调试的软件序列：

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

17.2.2. 引脚分配

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

JTCK 复用。

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

调试引脚分配：

表 17-1. 引脚分配

引脚	调试接口
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB4	NJTRST
PB3	JTDO/TRACESWO

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

17.2.3. JTAG 链状结构

Cortex®-M33 内核的 JTAG TAP 和 MCU JTAG TAP 串行连接。MCU JTAG TAP 的 IR(指令寄存器)是 5 位，而 Cortex®-M33 内核的 JTAG 的 IR(指令寄存器)是 4 位。所以当 JTAG 进行 IR 移位输入时，首先移位 5 位 BYPASS 指令给 MCU JTAG，然后移位 4 位标准指令给 Cortex®-M33 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为 MCU JTAG 已处在 BYPASS 模式。

MCU JTAG ID 代码是 0x790007A3。

17.2.4. 安全 JTAG

1. 安全 JTAG 只支持 JTAG，不支持 SW
2. OTP 配置

OTP 相关位：JTAGNSW、NDBG[1:0]、DBGPSWD_DATA[127:0]

表 17-2. 调试模式与 OTP 配置

模式	寄存器配置
无调试	NDBG[1:0] = 2'b0X JTAGNSW：不关心 DBGPSWD_DATA[127:0]：不关心
SW	NDBG[1:0] = 2'b10 JTAGNSW = 1'b1

	DBGPSWD_DATA[127:0]：不关心
普通JTAG	NDBG[1:0] = 2'b11 JTAGNSW = 1'b0 DBGPSWD_DATA[127:0]：不关心
安全JTAG	NDBG[1:0] = 2'b10 JTAGNSW = 1'b0 DBGPSWD_DATA[127:0]：OTP中调试密码字段值

3. 安全 JTAG 的使用

a) 配置 OTP 为安全 JTAG：首先配置调试密码 DBGPSWD_DATA[127:0]，再配置 JTAGNSW=1'b0，NDBG[1:0]=2'b10。

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 4 个密码写入 MCU JTAG 以解除安全模式。输入正确密码后，secure_jtag 为 0，可通过 JTAG 对 CPU 进行操作。

- IR：写入 5'b10011，DR：写入 DBGPSWD_DATA[31:0]。
- IR：写入 5'b10100，DR：写入 DBGPSWD_DATA[63:32]。
- IR：写入 5'b10101，DR：写入 DBGPSWD_DATA[95:64]。
- IR：写入 5'b10110，DR：写入 DBGPSWD_DATA[127:96]。

注意：输入密码错误或输入序列出现任何错误后，需电源复位才能重新解密。

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. 输入正确密码打开调试。

d) 读取写入值和 JTAG 状态。

- IR：写入 5'b10111，DR：可读出 IR 为 5'b10011 写入的值，检查写入值是否正确。
- IR：写入 5'b11000，DR：可读出 IR 为 5'b10100 写入的值，检查写入值是否正确。
- IR：写入 5'b11001，DR：可读出 IR 为 5'b10101 写入的值，检查写入值是否正确。
- IR：写入 5'b11010，DR：可读出 IR 为 5'b10110 写入的值，检查写入值是否正确。
- IR：写入 5'b11011，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。

17.2.5. 调试复位

JTAG-DP 和 SW-DP 寄存器位于上电复位域。系统复位初始化了 Cortex®-M33 的绝大部分组件，除了 NVIC 和调试逻辑。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。

17.2.6. JEDEC-106 ID 代码

Cortex®-M33 集成了 JEDEC-106 ID 代码。位于 ROM 表中，映射地址为 0xE00FF000~0xE00FFFFF。

17.3. 调试保持功能描述

17.3.1. 低功耗模式调试支持

当控制寄存器 0(DBG_CTL0)的 STB_HOLD 位置 1 并且进入待机模式，AHB 总线时钟和系统时钟由 CK_IRC8M 提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当控制寄存器 0(DBG_CTL0)的 DSLP_HOLD 位置 1 并且进入深度睡眠模式，AHB 总线时钟和系统时钟由 CK_IRC8M 提供，可以在深度睡眠模式下调试。

当控制寄存器 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。

17.3.2. TIMER、I2C、CAN、WWDGT、FWDGT 和 RTC 外设调试支持

当内核停止，并且控制寄存器 0/1(DBG_CTL0/DBG_CTL1)中的相应位置 1 时，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 CAN 外设，接收寄存器停止计数并进行调试；
- 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，保持 RTC 计数器时钟并进行调试。

17.4. DBG 寄存器

DBG 基地址：0xE005 4000

17.4.1. 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。

17.4.2. 控制寄存器 0(DBG_CTL0)

地址偏移：0x04
复位值：0x0000 0000，仅上电复位

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TIMER10_HOLD	TIMER9_HOLD	保留	TIMER13_HOLD	TIMER12_HOLD	保留			I2C2_HOLD	CAN1_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER7_HOLD	I2C1_HOLD
	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	CAN0_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	保留		TRACE_IOEN	保留	CAN2_HOLD	STB_HOLD	DSLP_HOLD	SLP_HOLD
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	TIMER10_HOLD	TIMER10 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 TIMER10 计数器不变，用于调试。
29	TIMER9_HOLD	TIMER9 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 TIMER9 计数器不变，用于调试。
28	保留	必须保持复位值。
27	TIMER13_HOLD	TIMER13 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 TIMER13 计数器不变，用于调试。
26	TIMER12_HOLD	TIMER12 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 TIMER12 计数器不变，用于调试。
25:23	保留	必须保持复位值。
22	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。
21	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。
20	TIMER6_HOLD	TIMER6 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER6 计数器不变，用于调试。
19	TIMER5_HOLD	TIMER5 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER5 计数器不变，用于调试。
18	TIMER4_HOLD	TIMER4 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER4 计数器不变，用于调试。
17	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 TIMER7 计数器不变，用于调试。
16	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。
15	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。
14	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。
13	TIMER3_HOLD	TIMER3 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER3 计数器不变，用于调试。
12	TIMER2_HOLD	TIMER2 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER2 计数器不变，用于调试。
11	TIMER1_HOLD	TIMER1 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 TIMER1 计数器不变，用于调试。
10	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 TIMER0 计数器不变，用于调试。
9	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。
8	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。
7:6	保留	必须保持复位值。
5	TRACE_IOEN	跟踪引脚分配使能 该位由软件置位和复位。

		0：禁用跟踪引脚分配 1：使能跟踪引脚分配
4	保留	必须保持复位值。
3	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。
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，系统时钟和 AHB 时钟由 CK_IRC8M 提供，当退出待机模式时，产生系统复位。
1	DSLP_HOLD	深度睡眠模式保持位 该位由软件置位和复位。 0：无影响 1：在深度睡眠模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供。
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位。 0：无影响 1：在睡眠模式下, AHB 时钟继续运行。

17.4.3. 控制寄存器 1(DBG_CTL1)

地址偏移：0x08
复位值：0x0000 0000，仅上电复位

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TIMER44_HOLD	TIMER43_HOLD	TIMER42_HOLD	TIMER41_HOLD	TIMER40_HOLD	保留		TIMER14_HOLD
								rw	rw	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	TIMER20_HOLD	RTC_HOLD	保留							
				rw	rw	rw	rw								

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	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 TIMER44 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 TIMER43 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 TIMER42 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 TIMER41 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 TIMER40 计数器不变，用于调试。
18:17	保留	必须保持复位值。
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 TIMER14 计数器不变，用于调试。
15:12	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 TIMER51 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 TIMER50 计数器不变，用于调试。
9	TIMER20_HOLD	TIMER20 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 TIMER20 计数器不变，用于调试。
8	RTC_HOLD	RTC 保持位 该位由软件置位和复位。 0：无影响

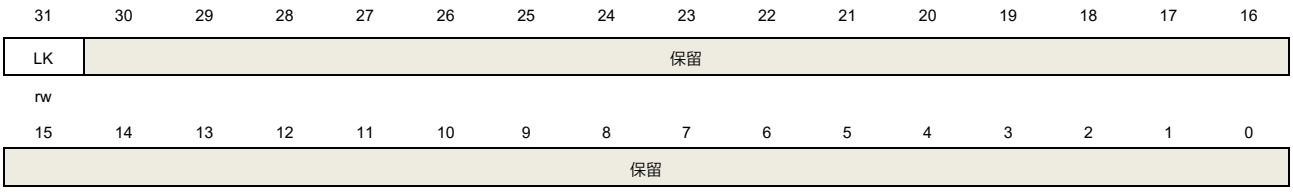
1：当内核停止时保持 RTC 计数器时钟，用于调试。

7:0	保留	必须保持复位值。

17.4.4. 控制寄存器 2(DBG_CTL2)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31	LK	DBG 寄存器解锁 0：DBG 寄存器锁定，寄存器只读 1：DBG寄存器解锁，寄存器可配置
30:0	保留	必须保持复位值。

18. 模数转换器(ADC)

18.1. 简介

MCU 片上集成了 12 位逐次逼近式模数转换器模块(ADC)，可以采样来自于外部输入通道和内部通道的模拟信号。ADC0 有 16 个外部通道和 5 个内部通道(温度传感器 V_{SENSE}，内部参考电压 V_{REFINT}，外部电池供电 V_{BAT}，正参考电压 V_{REFP}，负参考电压 V_{REFN})，ADC1 有 18 个外部通道和 3 个内部通道(内部参考电压 V_{REFINT}，正参考电压 V_{REFP}，负参考电压 V_{REFN})，ADC2 有 18 个外部通道和 3 个内部通道(内部参考电压 V_{REFINT}，正参考电压 V_{REFP}，负参考电压 V_{REFN})。

模拟看门狗允许应用程序来检测输入电压是否超出用户设定的高低阈值。

所有的 ADC 采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。

片上的硬件过采样机制可以通过减少来自 MCU 的相关计算负担来提高性能。

18.2. 主要特征

- 高性能：
  - ADC采样分辨率：12位、10位、8位、或者6位分辨率；
  - 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；
  - 常规和注入序列都支持 DMA 请求。
- 模拟输入通道：
  - ADC0有16个外部模拟输入通道，ADC1有18个外部模拟输入通道，ADC2有18个外部模拟输入通道；
  - 内部通道(ADC0_CH16)连接到温度传感器输出电压(V_{SENSE})；
  - 内部通道(ADC0_CH17, ADC1_CH18, ADC2_CH17)连接到内部参考电压(V_{REFINT})；
  - 内部通道(ADC0_CH18)连接到外部电池供电(V_{BAT})；
  - 内部通道(ADC0_CH19, ADC1_CH19, ADC2_CH19)连接到正基准电压(V_{REFP})；
  - 内部通道(ADC0_CH20, ADC1_CH20, ADC2_CH20)连接到负基准电压(V_{REFN})；
- 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)。
- 转换结果阈值监测功能：模拟看门狗。

- 中断产生：
  - 序列转换结束；
  - 模拟看门狗事件；
  - 溢出事件；
  - 触发错误事件；
- 过采样：
  - ADC0/1/2支持16位的数据寄存器；
  - 可调整的过采样率：ADC0/1/2的范围从2x到256x；
  - 可编程数据移位：ADC0/1/2最高可达8位。
- ADC输入范围：V_{REFN} <=V_{IN} <=V_{REFP}。

18.3. 引脚和内部信号

图 18-1. ADC 模块框图 给出了 ADC 模块框图。*表 18-1. ADC 内部信号*给出了 ADC 内部信号。*表 18-2. ADC 引脚定义*给出了 ADC 引脚说明。

表 18-1. ADC 内部信号

内部信号名称	说明
V_{SENSE}	内部温度传感器输出电压
V_{REFINT}	内部参考输出电压
V_{BAT}	外部电池供电电压
V_{REFP}	ADC正参考电压
V_{REFN}	ADC负参考电压

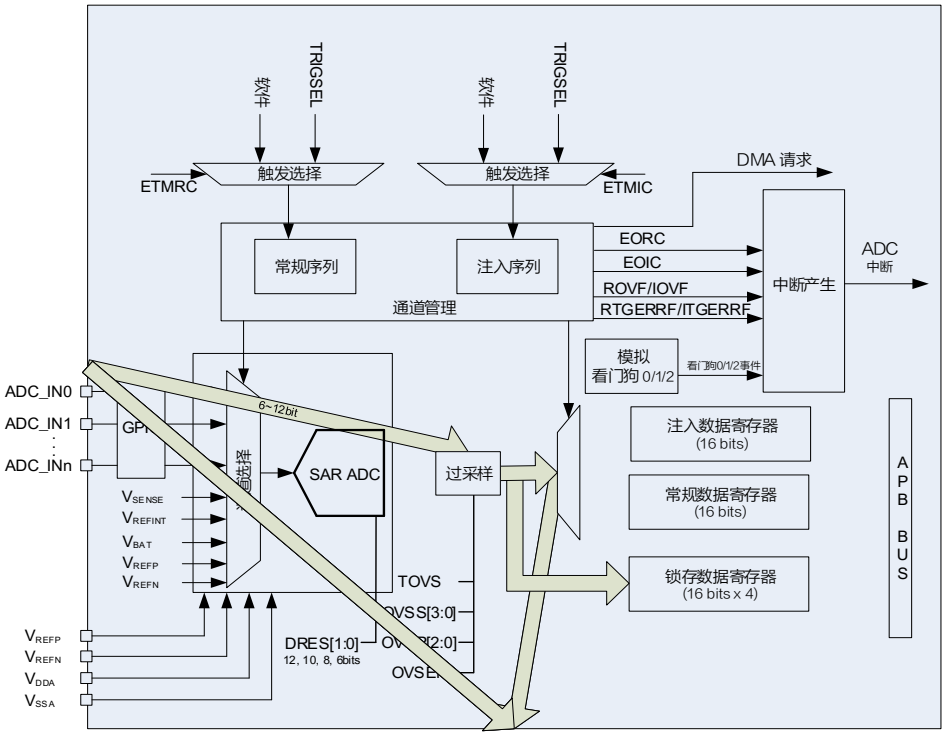
表 18-2. ADC 引脚定义

名称	注释
V_{DDA}	模拟电源输入,等于V_{DD}
V_{SSA}	模拟地，等于V_{SS}
V_{REFP}	ADC正参考电压
V_{REFN}	ADC负参考电压
ADC0_INn(n=0…15)	ADC0多达16路外部通道
ADC1_INn(n=0…17)	ADC1多达18路外部通道
ADC2_INn(n=0…16,18)	ADC2多达18路外部通道

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

18.4. 功能描述

图 18-1. ADC 模块框图



18.4.1. ADC 时钟

ADC 最大工作时钟频率为 42MHz。CK_ADC 时钟是由时钟控制器提供的，它和 AHB、APB2 时钟保持同步。ADC 时钟可以在 RCU 时钟控制器中进行分频和配置。

想要更多 ADC 时钟产生的信息，可以参考 RCU 章节内容。

18.4.2. ADC 使能

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

18.4.3. 常规序列和注入序列

通道管理电路把采样通道组织成两个序列：一个常规序列和一个注入序列。

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

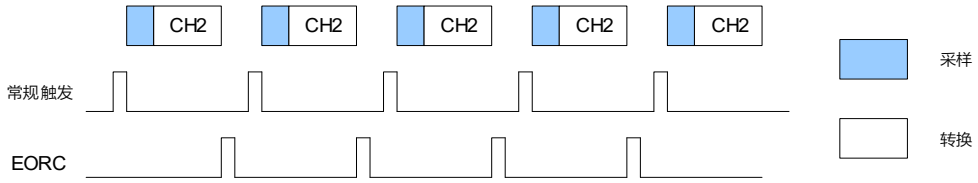
注入序列支持高达 4 个通道，每个通道称为注入通道。ADC_ISQ 寄存器的 IL[1:0]位规定了整个注入序列的长度。ADC_ISQ 寄存器规定了注入序列的通道选择。

18.4.4. 运行模式

单次运行模式

该模式能够运行在常规序列和注入序列。单次运行模式下，ADC_RSQ2 寄存器的 RSQ0[4:0]位或者 ADC_ISQ 寄存器的 ISQ3[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 18-2. 单次运行模式



常规序列的通道单次转换结束后，转换数据将被存放于 ADC_RDATA 寄存器中，EORC 将会置 1。如果 EORCIE 位被置 1，将产生一个中断。

注入序列的通道单次转换结束后，转换数据将被存放于 ADC_IDATA 寄存器中，EOIC 位将会置 1。如果 EOICIE 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ADC_RSQ2寄存器的RSQ0[4:0]位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 设置SWRCST位，或者为常规序列产生一个TRIGSEL触发信号；
6. 等到EORC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EORC标志位。

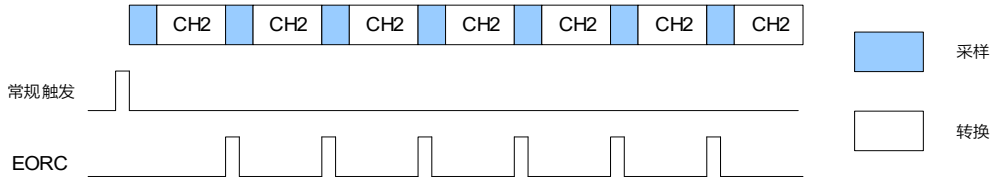
注入序列单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISIC和SM位为0；
2. 用模拟通道编号来配置ADC_ISQ寄存器的ISQ3[4:0]位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMIC[1:0]位域；
5. 设置SWICST位，或者为注入序列产生一个TRIGSEL触发信号；
6. 等到EOIC置1；
7. 从ADC_IDATA寄存器中读ADC转换结果；
8. 写0清除EOIC标志位。

连续运行模式

该模式能够运行在常规序列。将 ADC_CTL1 寄存器的 CTN 位置 1 可以使能连续运行模式。在此模式下，ADC 执行由 RSQ0[4:0]规定的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者 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 18-3. 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ADC_RSQ2寄存器的RSQ0[4:0]；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 设置SWRCST位，或者给常规序列产生一个TRIGSEL触发信号；
6. 等待EORC标志位置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EORC标志位；
9. 只要还需要进行连续转换，重复步骤6~8。

可以使用 DMA 来传输转换数据，不需循环查询 EORC 标志位，软件流程如下：

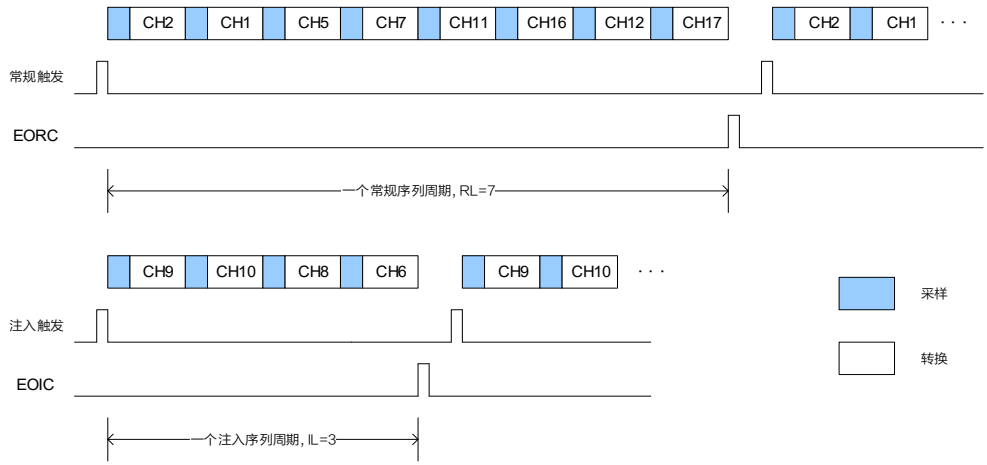
1. 设置ADC_CTL1寄存器的CTN和RDMA位为1；
2. 根据模拟通道编号配置ADC_RSQ2寄存器的RSQ0[4:0]；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个TRIGSEL触发。

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_RSQ2 寄存器或 ADC_ISQ 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或注入序列通道。转换数据存储在 ADC_RDATA 或 ADC_IDATA 寄存器中。常规序列或注入序列转换结束后，EORC 或者 EOIC 位将被置 1。如果 EORCIE 或 EOICIE 位被置 1，将产生中断。当常规或者注入序列工作在扫描模式下时，可以通过 ADC_CTL1 寄存器的 RDMA 或者 IDMA 位置 1 使能相应的 DMA 功能。

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 18-4. 扫描运行模式，且连续运行模式禁能



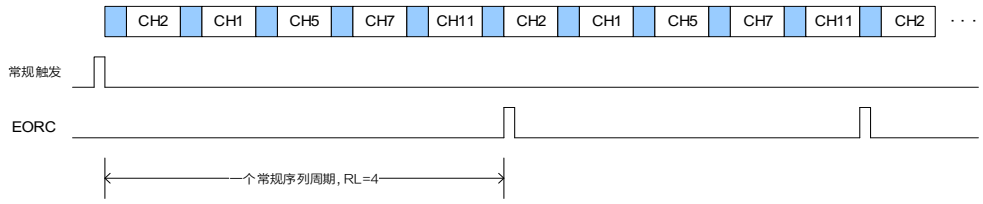
常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 RDMA 位为 1；
2. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMRC[1:0]位；
4. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据；
5. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；

注入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 IDMA 位为 1；
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMIC[1:0]位；
4. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；
5. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；

图 18-5. 扫描运行模式，连续运行模式使能



间断运行模式

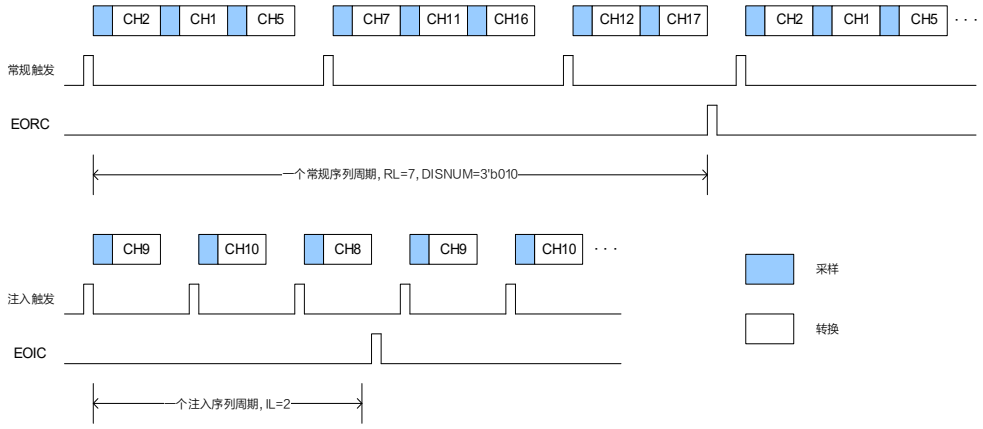
对于常规序列，当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列间断运行模式使能。该模式下可以执行一次 n 个通道的短序列转换(n<=8)，这个短序列是 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISNUM[2:0]位给出。当相应的软件触发或 TRIGSEL 触发发生，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所选择

通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EORC 位将置 1。如果 EORCIE 位置 1 将产生一个中断。

对于注入序列，当 ADC_CTL0 寄存器的 DISIC 位置 1 时，注入序列间断运行模式使能。该模式下可以执行 ADC_ISQ 寄存器所选择的转换序列的一个通道进行转换。当相应的软件触发或 TRIGSEL 触发发生，ADC 就会采样和转换 ADC_ISQ 寄存器中所选择通道的下一个通道，直到注入序列中所有通道转换完成。每个注入序列转换周期结束后，EOIC 位将被置 1。如果 EOICIE 位被置 1 将产生一个中断。

常规序列和注入序列不能同时工作在间断模式，同一时刻只能有一个序列被设置成间断模式。

图 18-6. 间断运行模式



常规序列间断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 RDMA 位为 1；
2. 配置 ADC_CTL0 寄存器的 DISNUM[2:0]位；
3. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 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；
7. 如果需要，重复步骤 6；

注入序列间断模式的软件流程：

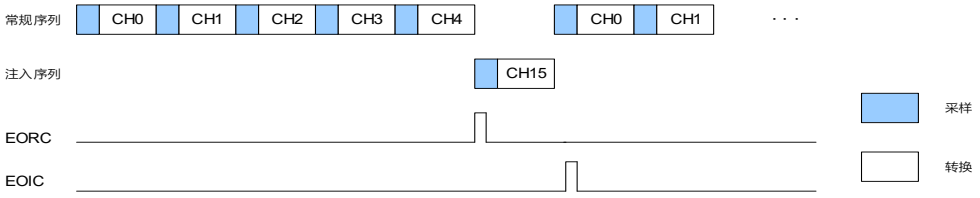
1. 设置 ADC_CTL0 寄存器的 DISIC 位和 ADC_CTL1 寄存器的 IDMA 位为 1；
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMIC[1:0]位；
4. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；
5. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；
6. 如果需要，重复步骤 5；

18.4.5. 注入序列管理

自动注入

如果将 ADC_CTL0 寄存器的 ICA 位置 1，在常规序列之后，注入序列被自动转换。该模式下注入序列的外部触发不能被使能。该模式可以转换 ADC_RSQ0~ADC_RSQ2 和 ADC_ISQ 寄存器中设置的多至 20 个通道。除了 ICA 位之外，如果 CTN 位也被置 1，常规序列将在注入序列之后被自动转换。

图 18-7. 自动注入，CTN=1

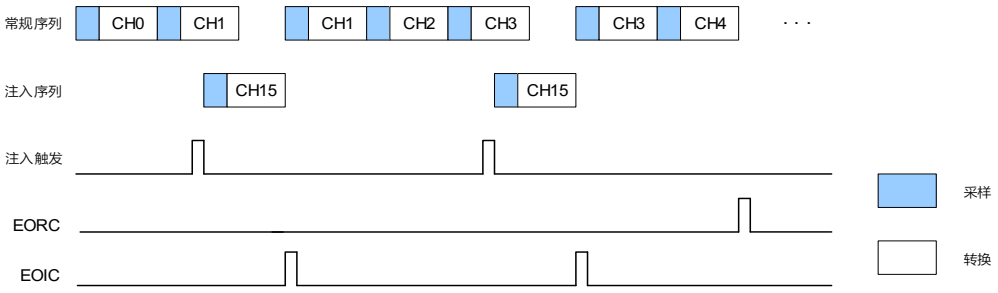


不能同时使用自动注入和间断模式。

触发注入

清除 ICA 位，在常规序列转换期间如果注入序列的软件触发或者 TRIGSEL 触发发生，则启动注入序列转换。这种情况下，ADC 取消常规序列中当前正在转换的通道，注入序列进行转换。注入序列转换结束后，ADC 从上次被取消的常规序列通道处重新开始转换。

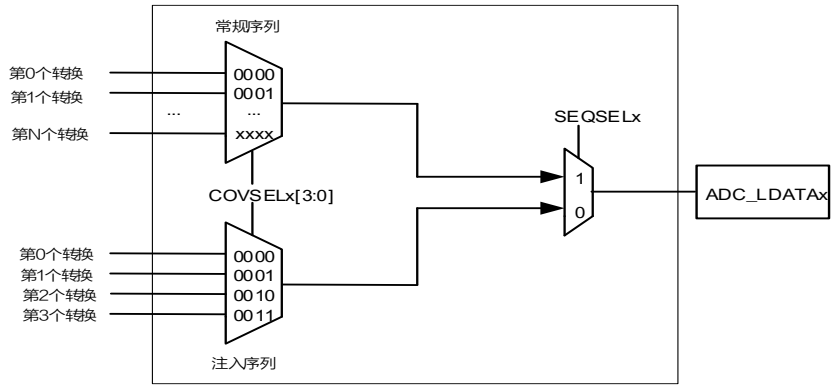
图 18-8. 触发注入



18.4.6. 转换数据锁存

ADC 拥有 4 个锁存数据寄存器，ADC_LDATAx(x=0…3)，这些寄存器可以在常规或注入序列通道转换完成后锁存数据。锁存数据寄存器的控制逻辑如*图 18-9. 数据锁存*。

图 18-9. 数据锁存

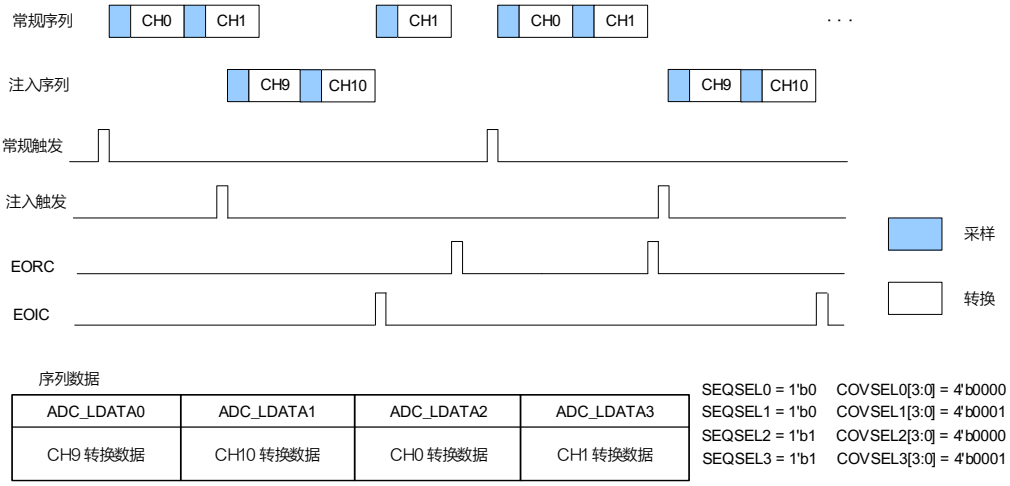


是从常规序列还是从注入序列中选择要锁存的数据，可以通过 ADC_LDCTL 寄存器中的 SEQSELx(x=0…3)位来决定，而 COVSELx[3:0]位则选择要锁存序列中的哪个转换结果。

设置 ADC_LDATAx(x=0…3)寄存器来存储注入序列或常规序列的转换结果，默认存储注入序列的第 x 次转换结果。

因此，除了使用 DMA 从 ADC_RDATA 或 ADC_IDATA 寄存器传输转换数据外，还可以直接从锁存数据寄存器读取数据，这为序列的连续转换提供了更大的灵活性。

图 18-10. ADC_LDATAx 读取序列数据



触发注入模式下从 ADC_LDATAx 读取序列数据的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位；
2. 配置 ADC_RSQx，ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 配置 ADC_LDCTL 寄存器的 SEQSEL2，COVSEL2[3:0]，SEQSEL3 和 COVSEL3[3:0]位域用于常规序列；
4. 配置 ADC_LDCTL 寄存器的 SEQSEL0，COVSEL0[3:0]，SEQSEL1 和 COVSEL1[3:0]位域用于注入序列；

5. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMRC[1:0]和 ETMIC[1:0]位域；

6. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；

7. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；

8. 等待 EORC 或 EOIC 标志位置 1；

9. 如果 EORC 标志位置 1，从 ADC_LDATA2 和 ADC_LDATA3 寄存器中读常规序列 ADC 转换结果；

10. 写 0 清除 EORC 标志位。

11. 如果 EOIC 标志位置 1，从 ADC_LDATA0 和 ADC_LDATA1 寄存器中读注入序列 ADC 转换结果；

12. 写 0 清除 EOIC 标志位。

18.4.7. 转换结果阈值监测功能

模拟看门狗 0

ADC_CTL0 寄存器的 RWD0EN 和 IWD0EN 位置 1 将分别使能常规序列和注入序列的模拟看门狗 0 功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值，ADC_STAT 状态寄存器的 WD0LE 或 WD0HE 位将被置 1。如果 WD0LEIE 或 WD0HEIE 位被置 1，将产生中断。ADC_WD0HT 和 ADC_WD0LT 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，因此阈值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWD0EN，IWD0EN，WD0SC 和 WD0CHSEL[4:0]位可以用来选择模拟看门狗 0 监控单一通道或者多通道。

模拟看门狗 1/2

模拟看门狗 1/2 更加的灵活，可以进行单个或多个通道的看门狗功能配置。

通过配置 ADC_WD1SR 寄存器中的 AWD1CS[20:0]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_WD1LT，ADC_WD1HT，ADC_WD2LT 和 ADC_WD2HT 寄存器中进行配置。

注： 如果 OVSEN=1，模拟看门狗 0/1/2 可以将转换的模拟电压(过采样后)与低阈值或高阈值进行比较。如果 OVSEN=0，模拟看门狗 0/1/2 可以将转换的模拟电压(过采样前)与低阈值或高阈值进行比较。

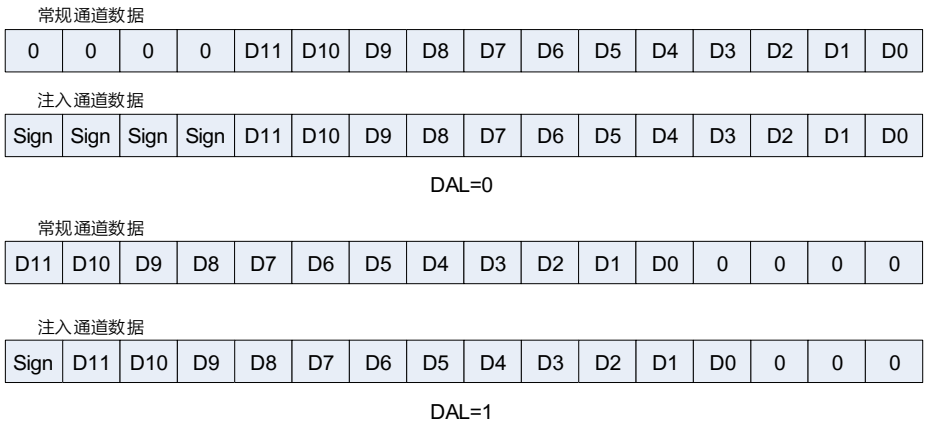
18.4.8. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

注入序列通道转换的数据值已经减去了在 ADC_IOFFx 寄存器中定义的偏移量，因此结果可能

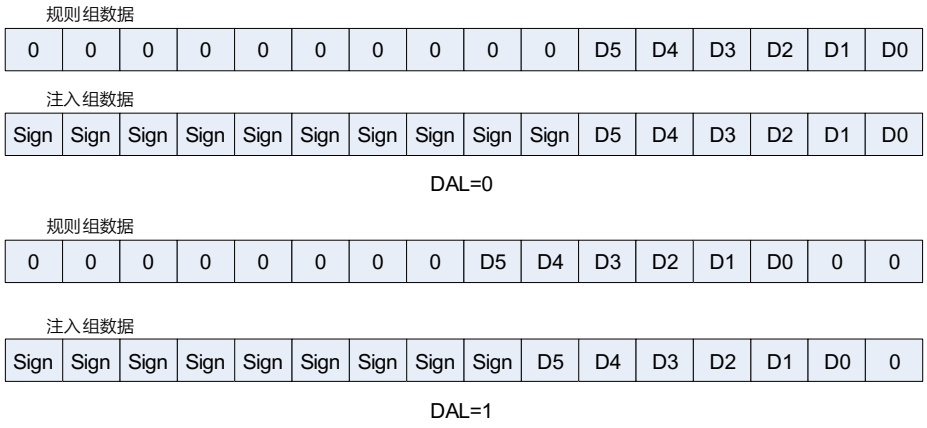
是一个负值。符号值是一个扩展值。

图 18-11. 12 位数据存储模式



6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如*图 18-12. 6 位数据存储模式*。

图 18-12. 6 位数据存储模式



18.4.9. 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_SAMPTx(x=0…2)寄存器的 SPTn[2:0]位更改。每个通道可以用不同的时间采样。在 12 位分辨率的情况下，总采样转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 42MHz，采样时间为 1.5 个周期，那么总的转换时间为："1.5+12.5"个 CK_ADC 周期，即 0.333us。

18.4.10. 外部触发

TRIGSEL 触发输入的上升沿或者软件触发可以触发常规序列或注入序列的转换。ADC_CTL1 寄存器的 ETMRC[1:0]和 ETMIC[1:0]分别控制常规序列或注入序列的触发模式。常规序列和注入序列的外部触发源均由 TRIGSEL 模块控制 。

表 18-3. 外部触发模式和触发类型

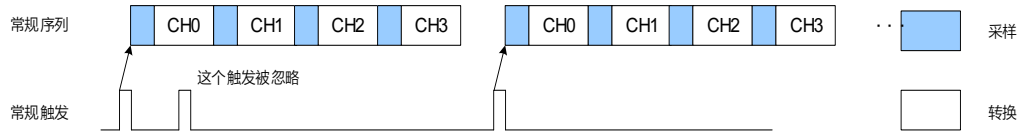
ETMRC[1:0]/ETMIC[1:0]	触发模式	触发类型
00	外部触发信号上升沿触发使能	硬件触发：TRIGSEL
01		
10		
11	外部触发禁能	软件触发：SWRCST/SWICST

外部触发的选择可以动态更改，同时不会因该更改而发生触发事件。

■ ITGRECEN = 0, RTGRECEN = 0

在注入序列转换期间，若发生新的注入序列或常规序列触发请求，或者在常规序列转换期间，发生新的常规序列触发请求，这些触发请求将会被忽略，如*图 18-13.在常规序列转换过程中常规触发出现(ITGRECEN = 0, RTGRECEN = 0)*所示。

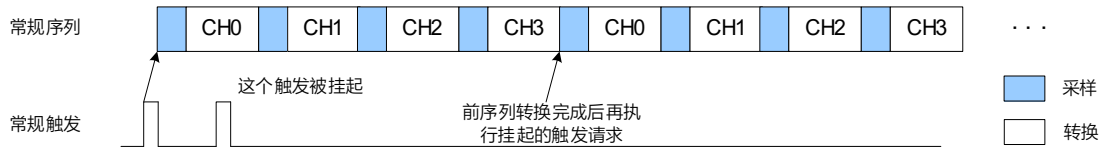
图 18-13.在常规序列转换过程中常规触发出现(ITGRECEN = 0, RTGRECEN = 0)



■ RTGRECEN = 1

在常规序列转换期间，若发生新的常规序列触发请求，该触发请求将被挂起，待当前序列转换完成后再执行挂起的触发请求，如*图 18-14. 在常规序列转换过程中常规触发出现(RTGRECEN = 1)*所示。

图 18-14. 在常规序列转换过程中常规触发出现(RTGRECEN = 1)



■ ITGRECEN = 1, RTGRECEN = 1

在注入序列转换期间，若发生新的注入序列或常规序列触发请求，这些触发请求将被挂起，待当前转换完成后再执行挂起的触发请求。当注入序列和常规序列的触发请求均被挂起时，ADC 先执行注入序列转换，注入序列转换完成后，再执行常规序列转换。

每个序列(注入序列或常规序列)的触发请求最多只能挂起一个，后续产生的触发请求将被忽略。

注意：ADC 同步模式中，触发挂起功能仅在 3 种模式下允许配置：规则并行，注入并行，规则并行 + 注入并行模式。

ADC_CTL1 寄存器的 RTGERREN 和 ITGERREN 位置 1 将分别使能常规序列和注入序列的触发错误检测功能。

如果在常规序列转换期间产生了一个常规序列触发信号，则 ADC_STAT 寄存器中的 RTGERRF 位将被置 1。如果 RTGERRIE 位被置 1，则产生中断。如果在注入序列转换期间产生了一个注入序列触发信号，则 ADC_STAT 寄存器中的 ITGERRF 位将被置 1。如果 ITGERRIE 位被置 1，则产生中断。

18.4.11. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 RDMA 或者 IDMA 位来使能，它用于传输常规或者注入序列多个通道的转换结果。ADC 在常规或者注入序列的一个通道转换结束后产生一个 DMA 请求，DMA 接受到请求后可以将转换的数据从 ADC_RDATA 或者 ADC_IDATA 寄存器传输到用户指定的目的地址。

18.4.12. 溢出检测

当 DMA 使能时，可以使能溢出检测。如果一个常规或者注入通道转换在上一个常规或者注入通道在上一个常规或者注入通道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 或者 IOVF 标志位会置位。如果 ADC_CTL0 寄存器的 ROVFIE 或者 IOVFIE 置位，溢出中断产生。

为了使得 ADC 从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，以保证常规序列转换数据正确的传输。当检测到溢出，ADC 转换将会停止，直到溢出标志位被清零。

注意：如果 ADC_CTL1 寄存器的 ACST=1，当产生溢出事件时，ADC 转换停止。

ADC 从溢出状态恢复的软件流程如下：

1. 将 ADC_CTL1 寄存器的 RDMA 或 IDMA 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 或 IOVF 位清 0，EORC 或 EOIC 位清 0；
5. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1；
6. 将 ADC_CTL1 寄存器的 RDMA 或 IDMA 位置 1；
7. 将 ADC_CTL1 的 ADCON 位置 1；
8. 等待 t_{su}；
9. 通过软件或触发开始 ADC 转换。

18.4.13. ADC 内部通道

将 ADC_CTL1 寄存器的 INREFEN 位置 1，可以使能内部参考电压 V_{REFINT} 通道(ADC0_CH17，ADC1_CH18，ADC2_CH17)。内部电压参考(V_{REFINT})提供了一个稳定的(带隙基准)电压输出给 ADC 和比较器。V_{REFINT} 内部连接到(ADC0_CH17，ADC1_CH18，ADC2_CH17)输入通道。

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

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

使用温度传感器：

1. 配置温度传感器通道(ADC0_IN16)的转换序列和采样时间大于t_{s_temp} μs；
2. 置位ADC_CTL1寄存器的TSVEN位，使能温度传感器；
3. 置位ADC_CTL1寄存器的ADCON位，或者由TRIGSEL触发启动ADC转换；
4. 从ADC数据寄存器中读取并计算温度传感器输出电压V_{SENSE}，并由下面公式计算出实际温度：

$$温度(°C) = \{(V_{25} - V_{SENSE}) / Avg_Slope\} + 25 \quad (18\text{-}1)$$

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)。

18.4.14. 电池电压检测

V_{BAT} 通道用于监测从 V_{BAT} 引脚过来的备份电池电压。当 ADC_CTL1 寄存器中的 VBATEN 位置 1 时，使能 V_{BAT} 通道(ADC0_CH18)，同时一个集成在 V_{BAT} 引脚上的 4 分压桥也随之自动被使能。由于 V_{BAT} 可能比 V_{DDA} 高，所以使用这个 4 分压桥用来确保 ADC 正确操作。它将 ADC0_CH18 输入通道连接到 V_{BAT}/4，所以，ADC0_CH18 输入通道转换的值是 V_{BAT}/4。为了防止不必要的电池能量消耗，推荐仅在需要时才使能 4 分压桥。

18.4.15. 可编程分辨率(DRES)

对寄存器 ADC_OVSAMPCTL 中的 DRES[1:0]位进行编程即可配置分辨率为 6、8、10、12 位。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 位为 0 时，才能修改 DRES[1:0]的值。ADC 转换的结果只有 12 位，其余没有被用到

的低位读出来都是为 0。较低的分辨率能够减少逐次逼近步骤所需的转换时间，如*表 18-4. 不同分辨率对应的 tCONV 时间*所示。

表 18-4. 不同分辨率对应的 t_{CONV} 时间

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=42MHz	t_s(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=42MHz
12	12.5	298 ns	1.5	14	333 ns
10	10.5	250 ns	1.5	12	286 ns
8	8.5	202 ns	1.5	10	238 ns
6	6.5	155 ns	1.5	8	190 ns

18.4.16. 片上硬件过采样

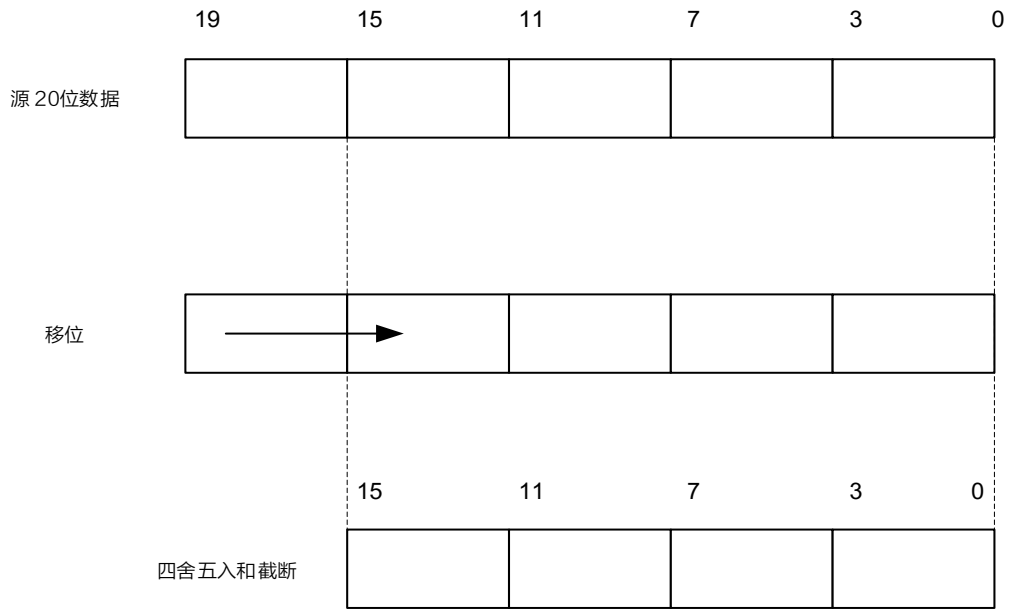
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，增加数据宽度，最高可达 16 位。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整。过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSEN 位来使能。Dout(n)是指 ADC 输出的第 n 个数字信号：

$$Result=\frac{1}{M}*\sum_{n=0}^{N-1}D_{out}(n) \quad (18\text{-}2)$$

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

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

图 18-15. 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 18-16. 右移 5 位和取整的数例*描述一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 18-16. 右移 5 位和取整的数例

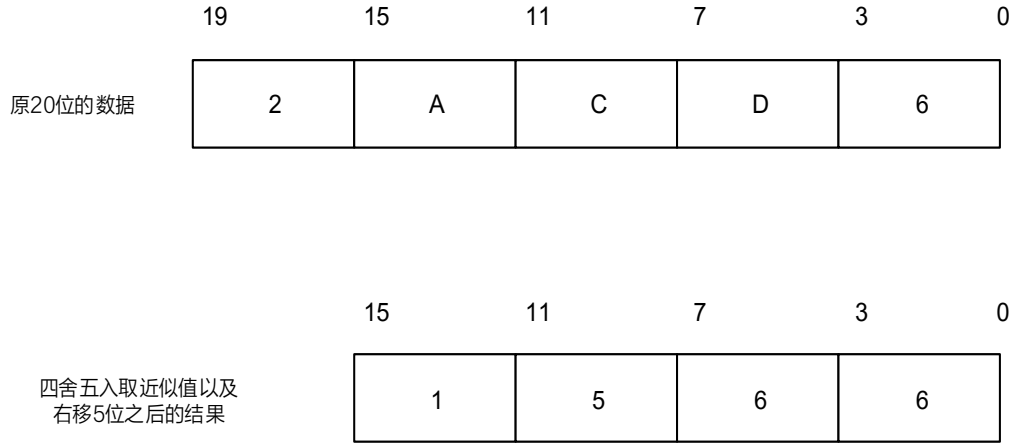


*表 18-5. N 和 M 的最大输出值(灰色部分表示截断)*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

表 18-5. N 和 M 的最大输出值(灰色部分表示截断)

过采样率	最大原始数据	无右移 OVSS=0000	右移 1 位 OVSS=0001	右移 2 位 OVSS=0010	右移 3 位 OVSS=0011	右移 4 位 OVSS=0100	右移 5 位 OVSS=0101	右移 6 位 OVSS=0110	右移 7 位 OVSS=0111	右移 8 位 OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F

4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

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

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (18\text{-}3)$$

18.5. ADC 同步模式

在具有多个 ADC 的设备上，可以使用 ADC 同步模式。

在 ADC 同步模式中，通过 ADC0 的触发器来交替或者同步 ADC1 的转换。根据 ADC0_CTL0 寄存器的 SYNCM[3:0]位域来选择 ADC 的同步模式。

在 ADC 同步模式中，当转换配置成外部事件触发时，ADC1 的外部触发必须禁能。

ADC 同步模式如*表 18-6. ADC 同步模式表*所示。

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

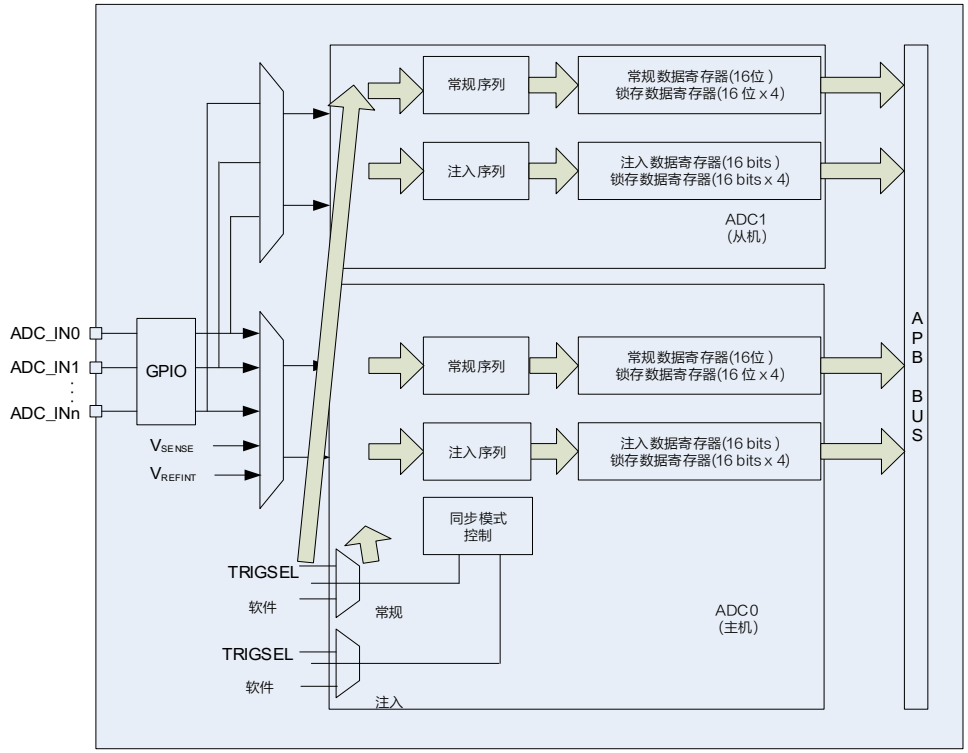
表 18-6. ADC 同步模式表

SYNCM[3: 0]	模式
0000	独立模式。所有的ADC都独立工作。
0001	ADC0和ADC1工作在常规并行和注入并行的组合模式。
0010	ADC0和ADC1工作在常规并行和注入交替触发的组合模式。
0011	ADC0和ADC1工作在注入并行和常规快速跟随的组合模式。
0100	ADC0和ADC1工作在注入并行和常规慢速跟随的组合模式。
0101	ADC0和ADC1工作在注入并行模式。
0110	ADC0和ADC1工作在常规并行模式。
0111	ADC0和ADC1工作在常规快速跟随模式。
1000	ADC0和ADC1工作在常规慢速跟随模式。
1001	ADC0和ADC1工作在注入交替触发模式。

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。注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

ADC 同步框图如*图 18-17. ADC 同步框图*所示。

图 18-17. ADC 同步框图



18.5.1. 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。

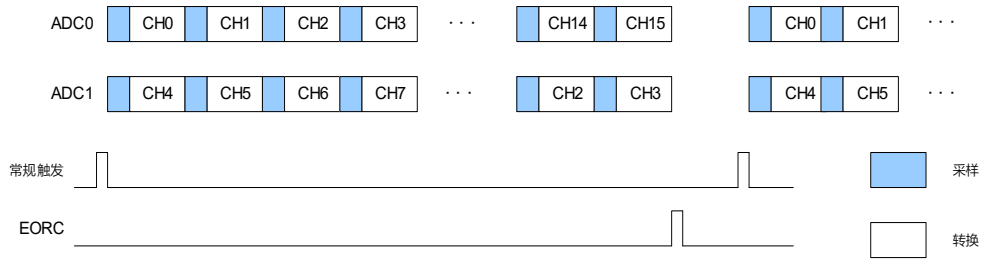
18.5.2. 常规并行模式

设置 ADC0_CTL0 寄存器中 SYNCM[3:0]位为 4'b0110，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，ADC0 和 ADC1 并行的转换常规通道。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0]位进行配置。

在转换结束时产生 EORC 中断(如果 ADC 使能了该中断)。常规并行模式的行为如*图 18-18. 基于 16 个通道的常规并行模式*所示。

32 位 ADC0_RDATA 寄存器包含上半字(由 ADC1 转换的数据构成)和下半字(由 ADC0 转换的数据构成)，32 位的 DMA 被用来将 ADC0_RDATA 中的数据传送到 SRAM。

图 18-18. 基于 16 个通道的常规并行模式



注意:

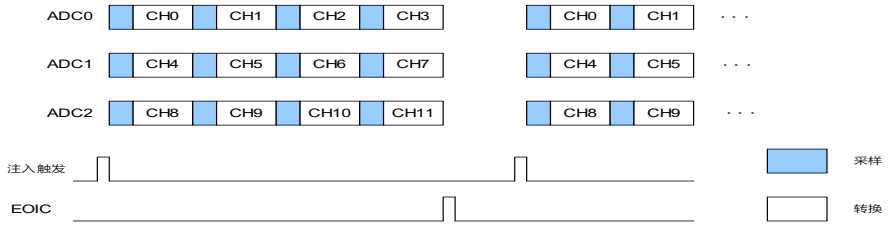
1. 两个 ADC 不能同时转换同一个通道。(当两个 ADC 转换同一通道时，采样时间不可重叠)
2. 在并行模式下，ADC0 和 ADC1 并行采样的两个通道需要设置为相同的采样时间。
3. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。

18.5.3. 注入并行模式

设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 0101，使能注入并行模式。在注入并行模式中，根据 ADC0 中选择的外部触发，ADC0 和 ADC1 并行的转换注入通道。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMIC[1:0]位进行配置。

在注入序列转换结束时产生 EOIC 中断(如果 ADC 使能了该中断)。转换的数据都存储在每个 ADC 的 ADC_IDATA 寄存器中。注入并行模式的行为如*图 18-19. 4 个通道的注入并行模式*所示。

图 18-19. 4 个通道的注入并行模式



注意:

1. 两个 ADC 不能同时转换同一个通道。(当两个 ADC 转换同一通道时，采样时间不可重叠)
2. 在并行模式下，ADC0 和 ADC1 并行采样的两个通道需要设置为相同的采样时间。
3. 注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

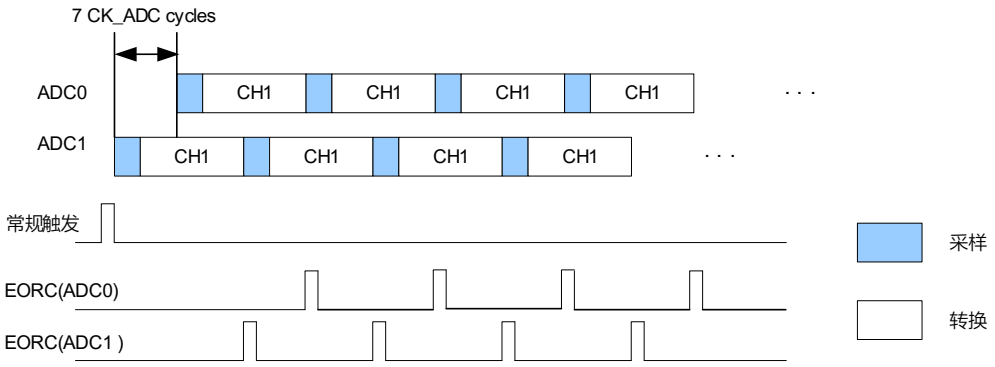
18.5.4. 常规快速交叉模式

此模式应用于 ADC 的常规序列(通常采样同一个通道)，设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 0111，使能快速交叉模式。ADC0 中选择的外部触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0]位进行配置。

如果 ADC0 和 ADC1 的 CTN 位被置位,所选的常规序列通道将在两个 ADC 中被不停的转换。如*图 18-20. 常规序列上的快速交叉模式(两个 ADC 的 CTN=1)* 所示。

32 位 ADC0_RDATA 寄存器包含上半字(由 ADC1 转换的数据构成)和下半字(由 ADC0 转换的数据构成)，在 ADC0 产生 EORC 中断后(如果 ADC 使能了该中断)，可通过 32 位 DMA 将 ADC0_RDATA 中数据传送到 SRAM。

图 18-20. 常规序列上的快速交叉模式(两个 ADC 的 CTN=1)



注意:

1. 可允许的最大采样时间必须小于 7 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在采样相同通道时出现采样时间重叠。
2. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。

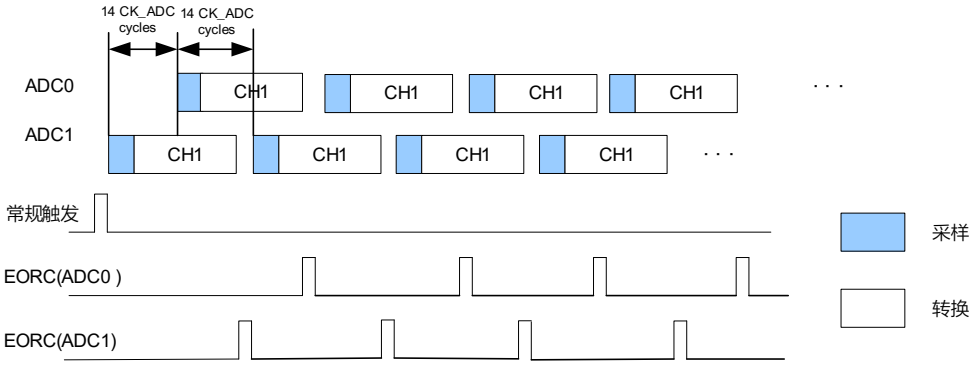
18.5.5. 常规慢速交叉模式

此模式应用于 ADC 的常规序列(通常采样同一个通道),设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 1000，使能慢速交叉模式。ADC0 中选择的外部触发产生时，ADC1 立刻启动，而 ADC0 在 14 个 ADC 时钟周期后启动，在 ADC0 启动后的 14 个时钟周期，ADC1 再次启动。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0]位进行配置。

在这种模式下，不能使用连续运行模式，因为在这种模式下所选的常规序列通道在两个 ADC 中被不停的转换，如*图 18-21. 常规序列上的慢速交叉模式*所示。

32 位 ADC0_RDATA 寄存器包含上半字(由 ADC1 转换的数据构成)和下半字(由 ADC0 转换的数据构成)，在 ADC0 产生 EORC 中断后(如果 ADC 使能了该中断)，可通过 32 位 DMA 将 ADC0_RDATA 中数据传送到 SRAM。

图 18-21. 常规序列上的慢速交叉模式



注意：

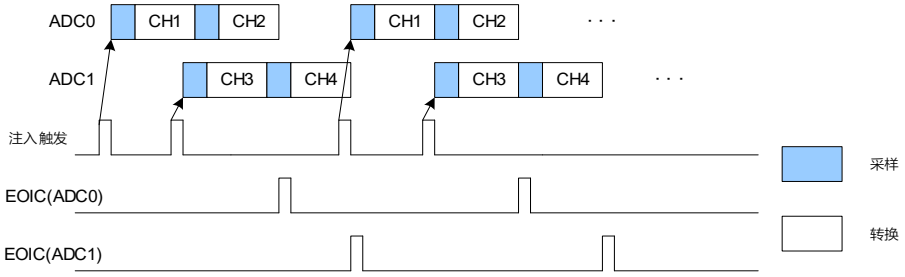
1. 可允许的最大采样时间必须小于 14 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时间重叠。
2. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。

18.5.6. 注入交替触发模式

此模式应用于注入序列，设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 1001，使能注入交替触发模式。如果 ADC0 和 ADC1 注入序列间断运行模式禁能，ADC0 中选择的外部触发第一次产生时，ADC0 注入序列的所有通道被转换，当第二次触发产生，ADC1 注入序列的所有通道被转换。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMIC[1:0]位进行配置。

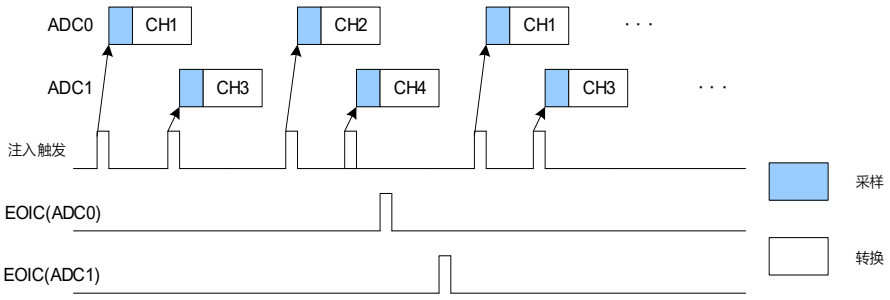
在注入序列转换结束时产生 EOIC 中断(如果 ADC 使能了该中断)。注入交替触发模式在 DISIC=0 时的行为如*图 18-22. 注入交替触发模式：DISIC=0, IL=1*所示。

图 18-22. 注入交替触发模式：DISIC=0, IL=1



如果 ADC0 和 ADC1 注入序列间断运行模式使能，当第一次触发发生，ADC0 转换第一个注入通道。当第二次触发发生，ADC1 转换第一个注入通道。然后，ADC0 转换第二个注入通道，ADC1 转换第二个通道，以此类推。注入交替触发模式在 DISIC=1 时的行为如*图 18-23. 注入交替触发模式：DISIC=1, IL=1*所示。

图 18-23. 注入交替触发模式：DISIC=1, IL=1



注意：

1. 两个 ADC 不能同时转换同一个通道。(当两个 ADC 转换同一通道时，采样时间不可重叠)
2. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。
3. 注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

18.5.7. 常规并行和注入并行组合模式

设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 0001，使能常规并行和注入并行组合模式。在常规并行和注入并行组合同步模式中，注入序列的并行转换可以中断常规并行转换。

在常规序列转换结束时产生 EORC 中断(如果 ADC 使能了该中断)。

在注入序列转换结束时产生 EOIC 中断(如果 ADC 使能了该中断)。

注意：

1. 在常规并行和注入并行组合模式下，ADC0 和 ADC1 并行采样的两个通道需要设置为相同的采样时间。
2. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。
3. 注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

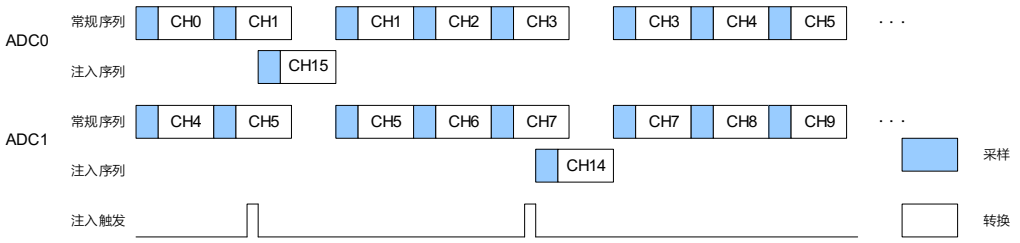
18.5.8. 常规并行和注入交替触发组合模式

设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 0010，使能常规并行和交替触发组合模式。在常规并行和注入交替触发组合模式中，注入序列的交替触发转换可以中断常规并行转换。当注入序列触发出现时，ADC 的注入序列转换立即开始。当常规序列转换被中断，ADC 的常规序列转换会停止在注入序列触发的时刻，并且在注入序列转换结束时，从被中断的位置恢复到并行模式。常规序列转换被注入交替触发中断的行为如*图 18-24. 常规并行和交替触发组合模式*所示。

在常规序列转换结束时产生 EORC 中断(如果 ADC 使能了该中断)。

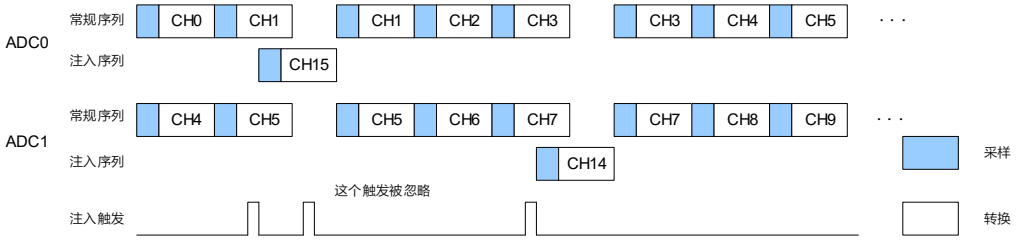
在注入序列转换结束时产生 EOIC 中断(如果 ADC 使能了该中断)。

图 18-24. 常规并行和交替触发组合模式



如果在一个注入序列转换期间，另一个注入触发出现，那么后面的这个触发将会被忽略，如*图 18-25. 在注入序列转换过程中注入触发出现*所示。

图 18-25. 在注入序列转换过程中注入触发出现



注意：

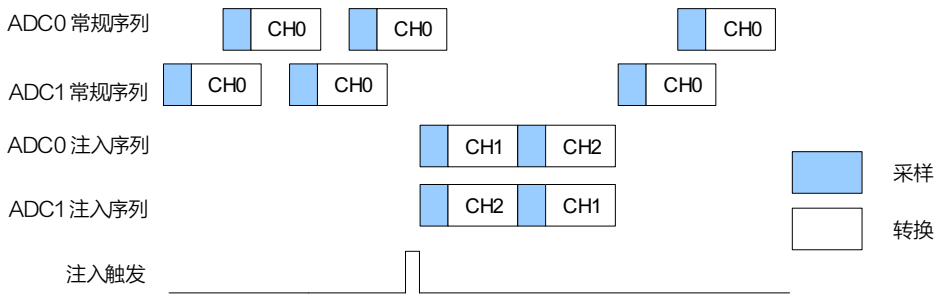
1. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。
2. 注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

18.5.9. 注入并行和交叉组合模式

设置 ADC_CTL0 寄存器中 SYNCM[3:0]位为 0011 或 0100，使能注入并行和快速(慢速)交叉组合模式。

注入并行转换可以中断(快速和慢速)交叉转换，当注入触发发生时，交叉转换被中断，注入序列转换启动。在注入序列转换完成后，交叉转换恢复，如*图 18-26. 单通道交叉转换被注入序列中断*所示。

图 18-26. 单通道交叉转换被注入序列中断



注意：

1. 确保设置 ADCON 位为 1 前先进行 ADC 模块复位。
2. 注入数据寄存器(ADC_IDATA)不能用于 DMA 传输。

18.6. 中断

以下任一个事件发生都可以产生中断：

- 常规序列和注入序列转换结束；
- 模拟看门狗事件；

单独的中断使能位可使得使用更灵活。ADC0、ADC1 被映射到同一个中断向量 IRQ18。ADC2 都被映射到中断向量 IRQ47。

18.7. ADC 寄存器

ADC0 基地址：0x4001 2400

ADC1 基地址：0x4001 2800

ADC2 基地址：0x4001 3C00

18.7.1. 状态寄存器(ADC_STAT)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		RTGERRF	ITGERRF	IOVF	ROVF	WD2HE	WD2LE	WD1HE	WD1LE	WD0LE	STRC	STIC	EOIC	EORC	WD0HE
		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	说明
31:14	保留	必须保持复位值。
13	RTGERRF	常规序列触发错误标志 0：无触发错误 1：发生触发错误 当检测到常规序列触发错误时，该位由硬件置位。常规序列触发错误检测仅在常规序列工作期间有效。 软件写0清除。
12	ITGERRF	注入序列触发错误标志 0：无触发错误 1：发生触发错误 当检测到注入序列触发错误时，该位由硬件置位。注入序列触发错误检测仅在注入序列工作期间有效。 软件写0清除。
11	IOVF	注入数据寄存器溢出 0：注入数据寄存器没有溢出 1：注入数据寄存器溢出 当注入数据寄存器溢出时，该位由硬件置位。只有在DMA使能(IDMA=1)时，这个标志位才会置位。 如果该位置位，则最近的注入数据会被丢失。

		软件写0清除。
10	ROVF	常规数据寄存器溢出 0：常规数据寄存器没有溢出 1：常规数据寄存器溢出 当常规数据寄存器溢出时，该位由硬件置位。只有在DMA使能(RDMA=1)时，这个标志位才会置位。 如果该位置位，则最近的常规数据会被丢失。 软件写0清除。
9	WD2HE	模拟看门狗2高阈值事件标志 0：没有模拟看门狗2高阈值事件 1：产生模拟看门狗2高阈值事件 转换电压超过ADC_WD2HT寄存器设定的阈值时由硬件置1。 软件写0清除。
8	WD2LE	模拟看门狗2低阈值事件标志 0：没有模拟看门狗2低阈值事件 1：产生模拟看门狗2低阈值事件 转换电压超过ADC_WD2LT寄存器设定的阈值时由硬件置1。 软件写0清除。
7	WD1HE	模拟看门狗1高阈值事件标志 0：没有模拟看门狗1高阈值事件 1：产生模拟看门狗1高阈值事件 转换电压超过ADC_WD1HT寄存器设定的阈值时由硬件置1。 软件写0清除。
6	WD1LE	模拟看门狗1低阈值事件标志 0：没有模拟看门狗1低阈值事件 1：产生模拟看门狗1低阈值事件 转换电压超过ADC_WD1LT寄存器设定的阈值时由硬件置1。 软件写0清除。
5	WD0LE	模拟看门狗0低阈值事件标志 0：没有模拟看门狗0低阈值事件 1：产生模拟看门狗0低阈值事件 转换电压超过ADC_WD0LT寄存器设定的阈值时由硬件置1。 软件写0清除。
4	STRC	常规序列转换开始标志 0：常规序列转换没有开始 1：常规序列转换开始 常规序列转换开始时硬件置位，软件写0清除。
3	STIC	注入序列转换开始标志 0：注入序列转换没有开始

		1：注入序列转换开始 注入序列转换开始时硬件置位，软件写 0 清除。
2	EOIC	注入序列转换结束标志 0：注入序列转换没有结束 1：注入序列转换结束 注入序列所有的通道转换结束时硬件置位，软件写0清除。
1	EORC	常规序列转换结束标志 0：常规序列转换没有结束 1：常规序列转换结束 常规序列转换结束时硬件置位，软件写0或读ADC_RDATA寄存器清除。
0	WD0HE	模拟看门狗0高阈值事件标志 0：没有模拟看门狗0高阈值事件 1：产生模拟看门狗0高阈值事件 转换电压超过ADC_WD0HT寄存器设定的阈值时由硬件置1。 软件写 0 清除。

18.7.2. 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
IOVFIE	ROVFIE	WD2HEIE	WD2LEIE	WD1HEIE	WD1LEIE	WD0HEIE	WD0LEIE	RWD0EN	IWD0EN	ITGERRIE	RTGERRIE	SYNCM[3:0]			
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
DISNUM[2:0]			DISIC	DISRC	ICA	WD0SC	SM	EOICIE	保留	EORCIE	WD0CHSEL[4:0]				
rw			rw	rw	rw	rw	rw	rw		rw	rw				

位/位域	名称	说明
31	IOVFIE	IOVF 中断使能 0：IOVF 中断禁能 1：IOVF 中断使能
30	ROVFIE	ROVF 中断使能 0：ROVF 中断禁能 1：ROVF 中断使能
29	WD2HEIE	WD2HE 中断使能 0：WD2HE 中断禁能 1：WD2HE 中断使能

28	WD2LEIE	WD2LE 中断使能 0：WD2LE 中断禁能 1：WD2LE 中断使能
27	WD1HEIE	WD1HE 中断使能 0：WD1HE 中断禁能 1：WD1HE 中断使能
26	WD1LEIE	WD1LE 中断使能 0：WD1LE 中断禁能 1：WD1LE 中断使能
25	WD0HEIE	WD0HE 中断使能 0：WD0HE 中断禁能 1：WD0HE 中断使能
24	WD0LEIE	WD0LE 中断使能 0：WD0LE 中断禁能 1：WD0LE 中断使能
23	RWD0EN	常规序列模拟看门狗 0 使能 0：常规序列模拟看门狗 0 禁能 1：常规序列模拟看门狗 0 使能
22	IWD0EN	注入序列模拟看门狗 0 使能 0：注入序列模拟看门狗 0 禁能 1：注入序列模拟看门狗 0 使能
21	ITGERRIE	ITGERRF 中断使能 0：ITGERRF 中断禁能 1：ITGERRF 中断使能
20	RTGERRIE	RTGERRF 中断使能 0：RTGERRF 中断禁能 1：RTGERRF 中断使能
19:16	SYNCM[3: 0]	同步模式选择 这些位用于运行模式选择 0000：独立模式 0001：常规并行和注入并行组合模式 0010：常规并行和注入交替触发组合模式 0011：注入并行和常规快速交叉组合模式 0100：注入并行和常规慢速交叉组合模式 0101：注入并行模式 0110：常规并行模式 0111：常规快速交叉模式 1000：常规慢速交叉模式 1001：注入交替触发模式

注意： 1)这些位只用于 ADC0；2)建议用户在任何配置之前关闭同步模式。

15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成 DISNUM[2:0]+1
12	DISIC	注入序列间断模式 0：注入序列间断运行模式禁能 1：注入序列间断运行模式使能
11	DISRC	常规序列间断模式 0：常规序列间断运行模式禁能 1：常规序列间断运行模式使能
10	ICA	注入序列自动转换 0：注入序列自动转换禁能 1：注入序列自动转换使能
9	WD0SC	扫描模式下，模拟看门狗 0 在通道配置 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：扫描运行模式禁能 1：扫描运行模式使能
7	EOICIE	EOIC 中断使能 0：EOIC 中断禁能 1：EOIC 中断使能
6	保留	必须保持复位值
5	EORCIE	EORC 中断使能 0：EORC 中断禁能 1：EORC 中断使能
4:0	WD0CHSEL[4:0]	模拟看门狗 0 通道选择 00000: ADC 通道 0 00001: ADC 通道 1 00010: ADC 通道 2 00011: ADC 通道 3 00100: ADC 通道 4 00101: ADC 通道 5 00110: ADC 通道 6 00111: ADC 通道 7 01000: ADC 通道 8 01001: ADC 通道 9 01010: ADC 通道 10 01011: ADC 通道 11

01100: ADC 通道 12
01101: ADC 通道 13
01110: ADC 通道 14
01111: ADC 通道 15
10000: ADC 通道 16
10001: ADC 通道 17
10010: ADC 通道 18
10011: ADC 通道 19
10100: ADC 通道 20
其他值保留。

注意:

1. ADC0 模拟输入通道 16、17、18、19、20 内部分别连接至温度传感器、V_{REFINT}、V_{BAT}、V_{REFP}、V_{REFN}。
2. ADC1 模拟输入通道 18、19、20 内部分别连接至 V_{REFINT}、V_{REFP}、V_{REFN}。
3. ADC2 模拟输入通道 17、19、20 内部分别连接至 V_{REFINT}、V_{REFP}、V_{REFN}。

18.7.3. 控制寄存器 1(ADC_CTL1)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		ITGERREN	RTGERREN	ITGRECEN	RTGRECEN	VBATEN	TSVEN	INREFEN	SWRCST	SWICST	保留	ETMRC[1:0]		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
保留	ETMIC[1:0]		保留	DAL	保留		RDMA	IDMA	保留				ACST	CTN	ADCON
	rw			rw			rw	rw					rw	rw	rw

位/位域	名称	说明
31:30	保留	必须保持复位值。
29	ITGERREN	注入序列触发错误检测使能 0：注入序列触发错误检测禁能 1：注入序列触发错误检测使能 **注意:** 1.注入序列触发错误检测仅在注入序列工作期间有效 2. ADC 同步模式中，只需要 ADC1 的 ITGERREN=1，ADC2 的 ITGERREN=0 即可。发生触发错误后，ADC1 的 ITGERRF 置 1，ADC2 的 ITGERRF 为 0。
28	RTGERREN	常规序列触发错误检测使能 0：常规序列触发错误检测禁能 1：常规序列触发错误检测使能

		注意： 1.常规序列触发错误检测仅在常规序列工作期间有效 2.ADC 同步模式中，只需要 ADC1 的 RTGREREN=1，ADC2 的 RTGREREN=0 即可。发生触发错误后，ADC1 的 RTGERRF 置 1，ADC2 的 RTGERRF 为 0。
27	ITGRECEN	注入序列触发恢复使能 0：注入序列触发将不会被存储和恢复 1：注入序列触发将被存储和恢复。(最多存储一个触发) **注意：** 1.注入序列触发恢复仅在注入序列工作期间有效 2.只能存储一个触发 3.ADC 同步模式中，触发挂起功能仅在 3 种模式下允许配置：规则并行，注入并行，规则并行 + 注入并行模式
26	RTGRECEN	常规序列触发恢复使能 0：常规序列触发将不会被存储和恢复 1：常规序列触发将被存储和恢复。(最多存储一个触发) **注意：** 1.常规序列触发恢复仅在常规序列工作期间有效 2.只能存储一个触发 3.ADC 同步模式中，触发挂起功能仅在 3 种模式下允许配置：常规并行，注入并行，常规并行 + 注入并行模式。 4.连续模式下(CTN=1)，该位必须为 0，不支持常规序列的触发挂起功能，并且要求 RTGERREN=0。
25	VBATEN	在 ADC0 中，该位可由软件置位或清零。 ADC0 的通道 18(电池供电电压 V_{BAT})使能 0：电池供电电压通道禁能 1：电池供电电压通道使能
24	TSVEN	在 ADC0 中，该位可由软件置位或清零。 ADC0 的通道 16(温度传感器)使能 0：温度传感器通道禁能 1：温度传感器通道使能
23	INREFEN	在 ADC0 中，该位可由软件置位或清零。 ADC0 的通道 17(内部参考电压 V_{REFINT})使能 ADC1 的通道 18(内部参考电压 V_{REFINT})使能 ADC2 的通道 17(内部参考电压 V_{REFINT})使能 0：内部参考电压通道禁能 1：内部参考电压通道使能
22	SWRCST	常规序列软件启动转换 如果 ETSRC 是 111，该位置'1'开启常规序列转换。软件置位，软件清零，或转换开始后，由硬件清零。
21	SWICST	注入序列软件启动转换

		如果 ETSIC 是 111，该位置'1'开启注入序列转换。软件置位，软件清零，或转换开始后，由硬件清零。
20	保留	必须保持复位值。
19:18	ETMRC[1:0]	常规序列外部触发模式 00，01，10：常规序列外部触发上升沿使能 11：常规序列外部触发禁能
17:15	保留	必须保持复位值。
14:13	ETMIC[1:0]	注入序列外部触发模式 00，01，10：注入序列外部触发上升沿使能 11：注入序列外部触发禁能
12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10:9	保留	必须保持复位值。
8	RDMA	常规序列的DMA请求使能 0：DMA请求禁止 1：DMA 请求使能
7	IDMA	注入序列的DMA请求使能 0：DMA请求禁止 1：DMA 请求使能 注意：同步模式下，该位保持 0，注入序列的 DMA 请求禁止。
6:3	保留	必须保持复位值。
2	ACST	当注入或常规数据寄存器溢出时，ADC 转换停止。 0：当注入或常规数据寄存器溢出时，ADC 转换不停止 1：当注入或常规数据寄存器溢出时，ADC 转换停止 **注意：** 1. 如果 ACST = 0，用户可以在溢出中断中处理溢出事件。 2. ACST = 1 时，ADC 转换停止后，如果溢出标志位 ROVF 或 IOVF 不清除，后续所有触发都被屏蔽。清除 ROVF 或 IOVF 后，ADC 继续被触发。 3. 同步模式下，若常规序列发生溢出(ADC1_STAT 寄存器的 ROVF=1)且 ACST=0，不影响 ADC 转换。 4. 同步模式下，若常规序列发生溢出(ADC1_STAT 寄存器的 ROVF=1)且 ACST=1，则 ADC0 和 ADC1 都停止转换。 5. 同步模式下，若注入序列发生溢出(ADC1_STAT 寄存器的 IOVF=1)且 ACST=0，不影响 ADC 转换。 6. 同步模式下，若注入序列发生溢出(ADC1_STAT 寄存器的 IOVF=1)且 ACST=1，

		则 ADC0 和 ADC1 都停止转换。
1	CTN	连续模式 0：禁能连续运行模式 1：使能连续运行模式
0	ADCON	开启 ADC 该位从'0'变成'1'将在稳定时间结束后唤醒 ADC。 0：禁能 ADC 关闭电源 1：使能 ADC

18.7.4. 采样时间寄存器 0(ADC_SAMPT0)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
					rw			rw			rw			rw	

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

位/位域	名称	说明
31:27	保留	必须保持复位值。
26:24	SPT18[2:0]	参考 SPT10[2:0]的描述
23:21	SPT17[2:0]	参考 SPT10[2:0]的描述
20:18	SPT16[2:0]	参考 SPT10[2:0]的描述
17:15	SPT15[2:0]	参考 SPT10[2:0]的描述
14:12	SPT14[2:0]	参考 SPT10[2:0]的描述
11:9	SPT13[2:0]	参考 SPT10[2:0]的描述
8:6	SPT12[2:0]	参考 SPT10[2:0]的描述
5:3	SPT11[2:0]	参考 SPT10[2:0]的描述
2:0	SPT10[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期

100：通道采样时间为41.5周期
101：通道采样时间为55.5周期
110：通道采样时间为71.5周期
111：通道采样时间为 239.5 周期

18.7.5. 采样时间寄存器 1(ADC_SAMPT1)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

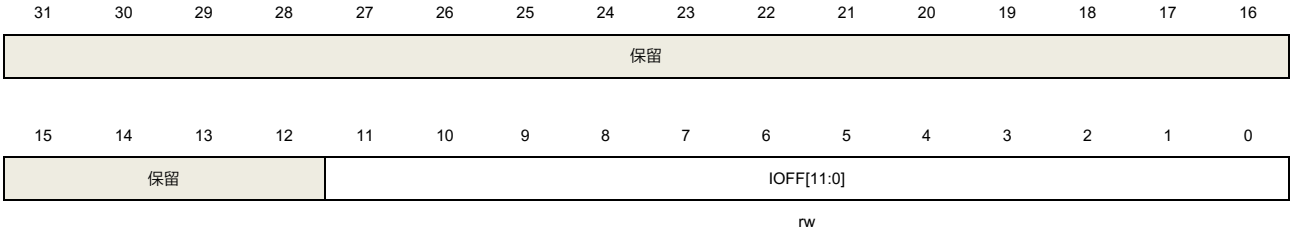
位/位域	名称	说明
31:30	保留	必须保持复位值。
29:27	SPT9[2:0]	参考 SPT0[2:0]的描述
26:24	SPT8[2:0]	参考 SPT0[2:0]的描述
23:21	SPT7[2:0]	参考 SPT0[2:0]的描述
20:18	SPT6[2:0]	参考 SPT0[2:0]的描述
17:15	SPT5[2:0]	参考 SPT0[2:0]的描述
14:12	SPT4[2:0]	参考 SPT0[2:0]的描述
11:9	SPT3[2:0]	参考 SPT0[2:0]的描述
8:6	SPT2[2:0]	参考 SPT0[2:0]的描述
5:3	SPT1[2:0]	参考 SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期

111：通道采样时间为 239.5 周期

18.7.6. 注入通道数据偏移寄存器 x(ADC_IOFFx)(x=0..3)

地址偏移：0x14-0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

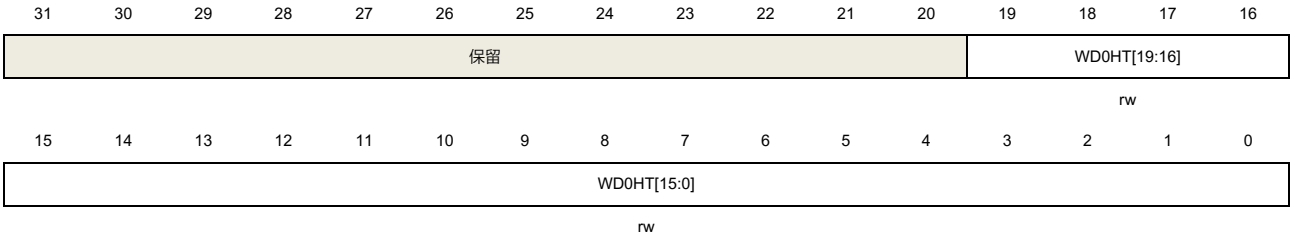


位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	IOFF[11:0]	注入通道 x 的数据偏移 当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在 ADC_IDATA 寄存器中读出。

18.7.7. 看门狗 0 高阈值寄存器(ADC_WD0HT)

地址偏移：0x24
复位值：0x000F FFFF

该寄存器只能按字(32 位)访问

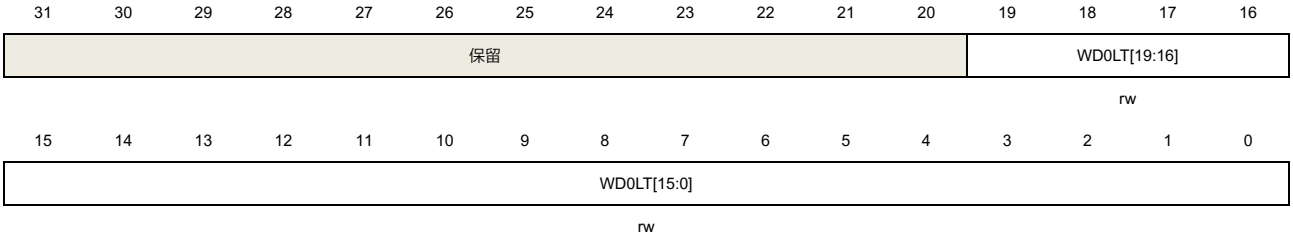


位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD0HT[19:0]	模拟看门狗 0 高侧阈值 这些位定义了模拟看门狗 0 的高阈值。

18.7.8. 看门狗 0 低阈值寄存器(ADC_WD0LT)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

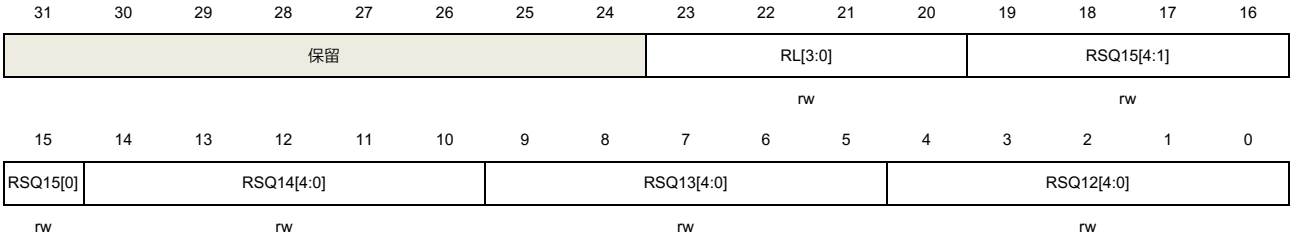


位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD0LT[19:0]	模拟看门狗 0 低侧阈值 这些位定义了模拟看门狗 0 的低阈值。

18.7.9. 常规序列寄存器 0(ADC_RSQ0)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

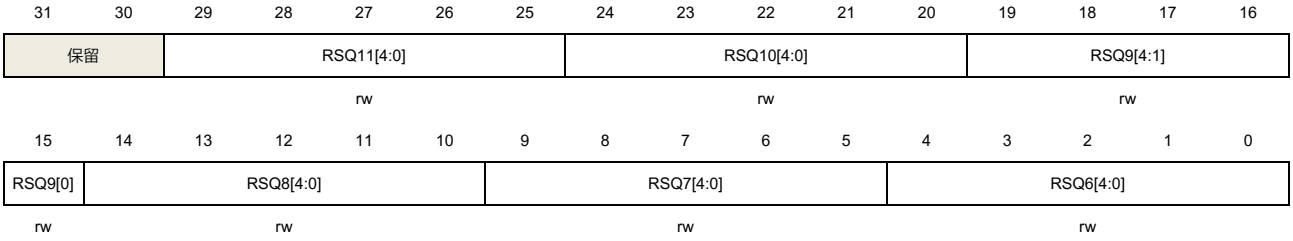


位/位域	名称	说明
31:24	保留	必须保持复位值。
23:20	RL[3:0]	常规序列长度 常规转换序列中的总的通道数目为 RL[3:0]+1。
19:15	RSQ15[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考 RSQ0[4:0]的描述

18.7.10. 常规序列寄存器 1(ADC_RSQ1)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

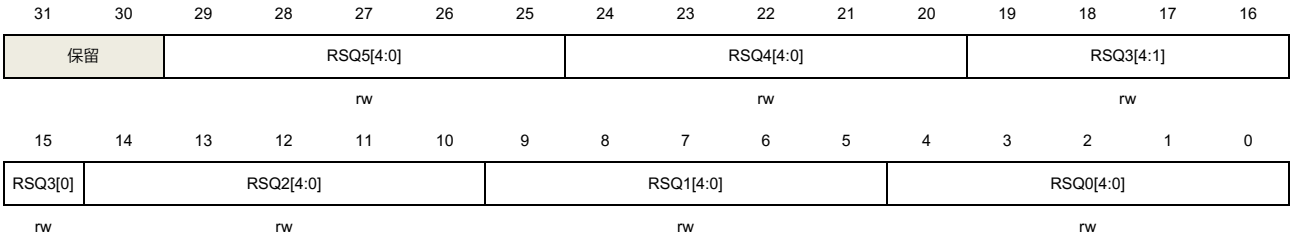


位/位域	名称	说明
31:30	保留	必须保持复位值。
29:25	RSQ11[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考 RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考 RSQ0[4:0]的描述

18.7.11. 常规序列寄存器 2(ADC_RSQ2)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



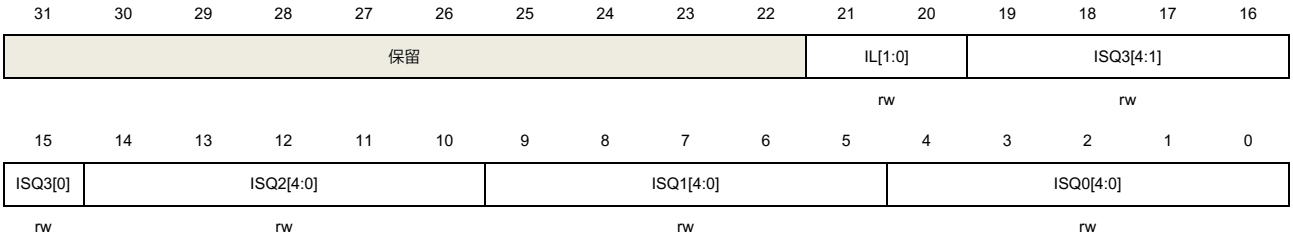
位/位域	名称	说明
31:30	保留	必须保持复位值。
29:25	RSQ5[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考 RSQ0[4:0]的描述

19:15	RSQ3[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ0[4:0]	通道编号(0..20)写入这些位来选择常规序列的第 n 个转换的通道

18.7.12. 注入序列寄存器(ADC_ISQ)

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	说明
31:22	保留	必须保持复位值。
21:20	IL[1:0]	注入序列长度 注入序列总的通道数目为 IL[1:0]+1。
19:15	ISQ3[4:0]	参考 ISQ0[4:0]的描述
14:10	ISQ2[4:0]	参考 ISQ0[4:0]的描述
9:5	ISQ1[4:0]	参考 ISQ0[4:0]的描述
4:0	ISQ0[4:0]	通道编号(0..20)写入这些位来选择注入序列的第 n 个转换的通道 和常规序列不同的是，如果 IL[1:0]长度不足 4，注入通道转换从(4-IL[1:0]-1) 开始。 IL 注入通道转换顺序 3 ISQ0 >> ISQ1 >> ISQ2 >> ISQ3 2 ISQ1 >> ISQ2 >> ISQ3 1 ISQ2 >> ISQ3 0 ISQ3

18.7.13. 锁存数据寄存器 x(ADC_LDATAx)(x= 0..3)

地址偏移：0x3C + 0x04 * x (x=0..3)
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	LDATAn[15:0]	注入或者常规序列第 n 个转换数据 这些位包含了注入或者常规序列第 n 个转换数据，只读。

18.7.14. 常规数据寄存器(ADC_RDATA)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC1RDATA[15:0]															
r															

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

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

18.7.15. 注入数据寄存器(ADC_IDATA)

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
IDATA[15:0]															
r															

位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	IDATA[15:0]	注入通道转换的数据 这些位包含了注入通道的转换结果，只读。

18.7.16. 锁存数据控制寄存器(ADC_LDCTL)

地址偏移：0x54
复位值：0x0001 0203

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEQSEL0	保留			COVSEL0[3:0]				SEQSEL1	保留			COVSEL1[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQSEL2	保留			COVSEL2[3:0]				SEQSEL3	保留			COVSEL3[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31	SEQSEL0	ADC_LDATA0寄存器序列源选择 0：选择注入序列 1：选择常规序列 **注意：**只有在ADC禁能(ADCON=0)时，才能软件写这些位。
30:28	保留	必须保持复位值。
27:24	COVSEL0[3:0]	ADC_LDATA0寄存器转换源选择 当ADC_LDATA0寄存器用于序列转换时，可以选择将注入序列(SEQSEL0 = 0)或者常规序列(SEQSEL0 = 1)的第n个转换的数据存储在ADC_LDATA0寄存器中。 0000：序列第0个转换 0001：序列第1个转换 0010：序列第2个转换 … 1111：序列第15个转换 其他值保留。 **注意：**在注入序列(SEQSEL0 = 0)中，只能为4'b0000、4'b0001、4'b0010和4'b0011，默认值为4'b0000。只有在ADC禁能(ADCON=0)时，才能软件写这些位。

23	SEQSEL1	参考SEQSEL0描述。
22:20	保留	必须保持复位值。
19:16	COVSEL1[3:0]	参考COVSEL0[3:0]描述。
15	SEQSEL2	参考SEQSEL0描述。
14:12	保留	必须保持复位值。
11:8	COVSEL2[3:0]	参考COVSEL0[3:0]描述。
7	SEQSEL3	参考SEQSEL0描述。
6:4	保留	必须保持复位值。
3:0	COVSEL3[3:0]	参考COVSEL0[3:0]描述。

18.7.17. 看门狗 1 高阈值寄存器(ADC_WD1HT)

地址偏移：0x60
复位值：0x000F 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												WD1HT[19:16]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD1HT[15:0]															
rw															

位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD1HT[19:0]	模拟看门狗 1 高侧阈值 这些位定义了模拟看门狗 1 的高阈值。

18.7.18. 看门狗 1 低阈值寄存器(ADC_WD1LT)

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												WD1LT[19:16]			
												rw			

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

WD1LT[15:0]
rw

位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD1LT[19:0]	模拟看门狗 1 低侧阈值 这些位定义了模拟看门狗 1 的低阈值。

18.7.19. 看门狗 2 高阈值寄存器(ADC_WD2HT)

地址偏移：0x68
复位值：0x000F 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												WD2HT[19:16]			
												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD2HT[15:0]															
rw															

位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD2HT[19:0]	模拟看门狗 2 高侧阈值 这些位定义了模拟看门狗 2 的高阈值。

18.7.20. 看门狗 2 低阈值寄存器(ADC_WD2LT)

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												WD2LT[19:16]			
												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD2LT[15:0]															
rw															

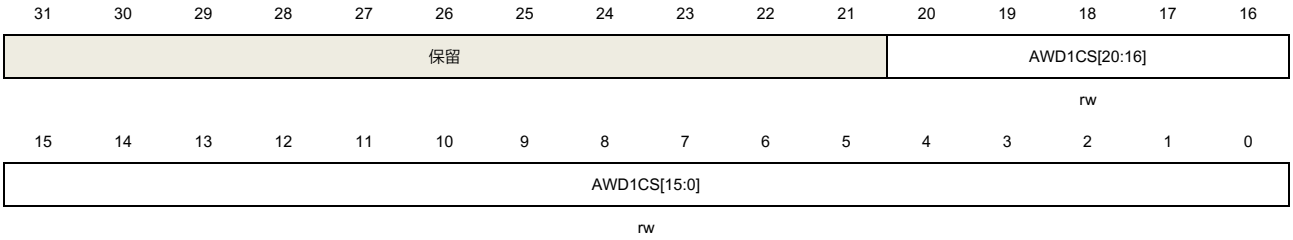
位/位域	名称	说明
31:20	保留	必须保持复位值。

19:0	WD2LT[19:0]	模拟看门狗 2 低侧阈值 这些位定义了模拟看门狗 2 的低阈值。

18.7.21. 看门狗 1 通道选择寄存器(ADC_WD1SR)

地址偏移：0x70
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

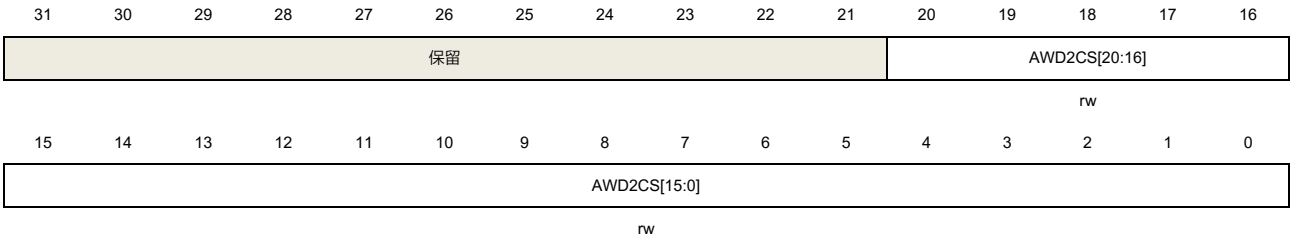


位/位域	名称	描述
31:21	保留	必须保持复位值。
20:0	AWD1CS[20:0]	模拟看门狗1通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗1保护的输入通道。 AWD1CS[n] = 0：ADC模拟输入通道n不由模拟看门狗1保护。 AWD1CS[n] = 1：ADC模拟输入通道n由模拟看门狗1保护。 当AWD1CS[20:0] = 000..0，模拟看门狗1禁能。 **注意:** 1)通过AWD1CS位域配置的模拟看门狗1功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道; 2)只有在ADC禁能(ADCON=0)时，才能软件写这些位。

18.7.22. 看门狗 2 通道选择寄存器(ADC_WD2SR)

地址偏移：0x74
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述

31:21	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AWD2CS[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。

18.7.23. 过采样控制寄存器(ADC_OVSAMPCTL)

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

位/位域	名称	说明
31:14	保留	必须保持复位值。
13:12	DRES[1：0]	ADC 分辨率 00：12 位 01：10 位 10：8 位 11：6 位
11:10	保留	必须保持复位值。
9	TOVS	过采样触发 该位通过软件置位和清除。 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[2:0])决定。 **注意：**只有在 ADCON=0 的时候才允许通过软件对该位进行写操作(确保没有转换正在执行)。
8:5	OVSS[3:0]	过采样移位 该位通过软件置位和清除。

		0000：不移位 0001：移 1 位 0010：移 2 位 0011：移 3 位 0100：移 4 位 0101：移 5 位 0110：移 6 位 0111：移 7 位 1000：移 8 位 其余值都保留 **注意：**只有在 ADCON=0 的时候才允许通过软件对该位进行写操作(确保没有转换正在执行)。
4:2	OVSR[2:0]	过采样率 这些位定义了过采样率的大小. 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x **注意：**只有在 ADCON=0 的时候才允许通过软件对该位进行写操作(确保没有转换正在执行)。
1	保留	必须保持复位值。
0	OVSEN	过采样使能 该位通过软件置位和清除。 0：过采样禁能 1：过采样使能 **注意：**只有在 ADCON=0 的时候才允许通过软件对该位进行写(确保没有转换正在执行)。

18.7.24. 采样时间寄存器 2(ADC_SAMPT2)

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

保留	SPT20[2:0]	SPT19[2:0]
	rw	rw

位/位域	名称	说明
31:6	保留	必须保持复位值。
5:3	SPT20 [2:0]	参考 SPT19[2:0]的描述
2:0	SPT19[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为 239.5 周期

19. 数模转换器(DAC)

19.1. 简介

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

在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

每个 DAC 的两个通道可以独立或并发工作。

19.2. 主要特征

DAC 的主要特征如下：

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

*图 19-1. DAC 结构框图*为 DAC 的结构框图，*表 19-1. DAC 引脚*给出了引脚描述。

图 19-1. DAC 结构框图

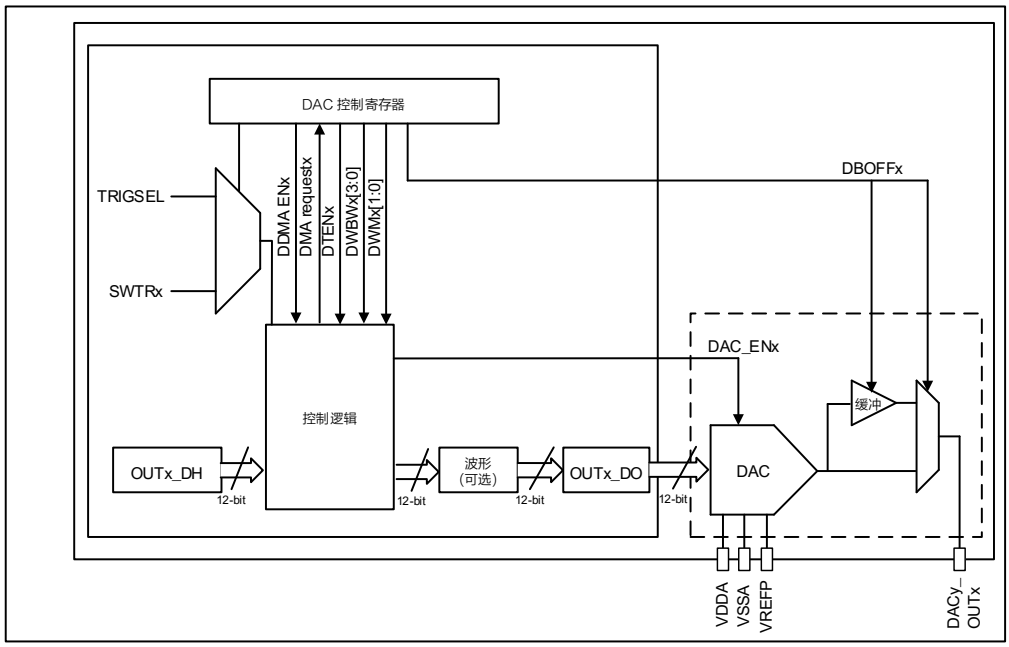


表 19-1. DAC 引脚

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 19-2. 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 触发信号	●	
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。

19.3. 功能描述

19.3.1. 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。

19.3.2. 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。

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

19.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据(OUTx_DH)，可以通过对 DAC_OUTx_R12DH、DAC_OUTx_L12DH 和 DAC_OUTx_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 DAC_OUTx_R8DH 寄存器时，只有 8 位最高有效位是可配置，4 位最低有效位被强制置为 4'b0000。

19.3.4. 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 19-3. DAC 外部触发*所示。

表 19-3. DAC 外部触发

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

外部触发器由 TRIGSEL 生成，软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

注意：只能在 DAC 模块使能之后(DAC_CTL0 寄存器中 DENx=1)，再去触发 DAC 模块。

19.3.5. DAC 转换

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

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

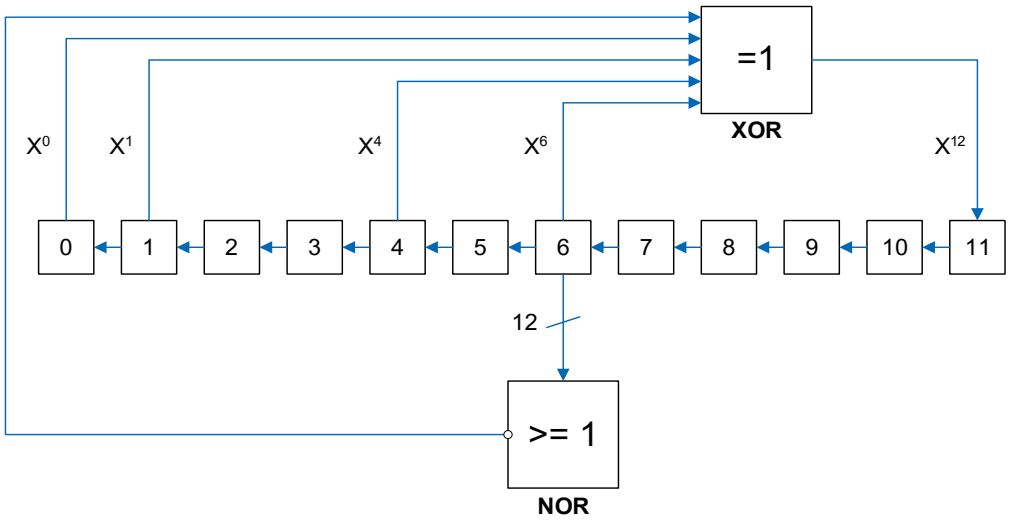
19.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通

过 DAC_CTL0 寄存器的 DWMx 位来进行选择。噪声的幅值可以通过配置 DAC_CTL0 寄存器的 DAC 噪声波位宽(DWBWx)位来进行设置。

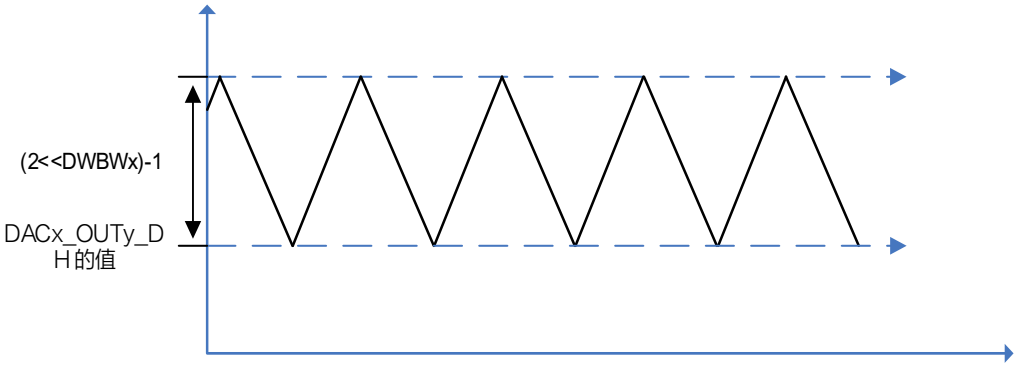
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器(LFSR)。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器(DAC_OUTx_DO)。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 19-2. DAC LFSR 算法



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

图 19-3. DAC 三角噪声模式生成的波形



19.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (19\text{-}1)$$

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}。

19.3.8. 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。

19.3.9. 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，与此同时需要任意 DDMAENx 置位。

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。

19.3.10. 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：

$$V_{out}=(D/2^N)*G*V_{REFP}+V_{of} \quad (19\text{-}2)$$

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：

- 缓冲区与片上外设断开连接并进入三态。
- 缓冲区将用作比较器，并通过内部电桥配置V_{REFP}*255/4095和V_{REFP}*3584/4095作为输入，DAC_CALR 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 OTVx_P[4:0]和 OTVx_N[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。

通过使用逐次逼近或二分法技术，可以更快地计算 OTVx_P[4:0]和 OTVx_N[4:0]的值。以下过程是逐次逼近方法：

- DAC_CTL0 寄存器中 DENx 位写 0 以禁能 DAC 输出。
- DAC_CTL0 寄存器中 DBOFFx 位写 0 以使能 DAC 输出缓冲区。
- DAC_CALR 寄存器中 CALENx 位置 1 使能 DAC 校准。
- 执行校准算法
  - 写入 0xFF(255)到 DAC_OUTx_R12DH 寄存器的 OUTx_DH[11:0]。
  - 从 0'b00000 开始写入 OTVx_P[4:0]。
  - 等待t_{TRIM}时间。
  - 检查 DAC_CALR 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 P。
  - 写入 0xE00(3584)到 DAC_OUTx_R12DH 寄存器的 OUTx_DH[11:0]。
  - 从 0'b00000 开始写入 OTVx_N[4:0]。
  - 等待t_{TRIM}时间。
  - 检查 DAC_CALR 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 N。

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。

19.4. DAC 寄存器

DAC0 基地址：0x4000 7400

19.4.1. 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	DCSEL1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		保留	DTSEL1[1:0]		DTEN1	DBOFF1	DEN1
	rw	rw	rw	rw				rw			rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DCSEL0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		保留	DTSEL0[1:0]		DTEN0	DBOFF0	DEN0
	rw	rw	rw	rw				rw			rw		rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值
30	DCSEL1	DACx_OUT1 与 GPIO 的连接选择 0：DACx_OUT1与GPIO连接 1：DACx_OUT1 不与 GPIO 连接
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

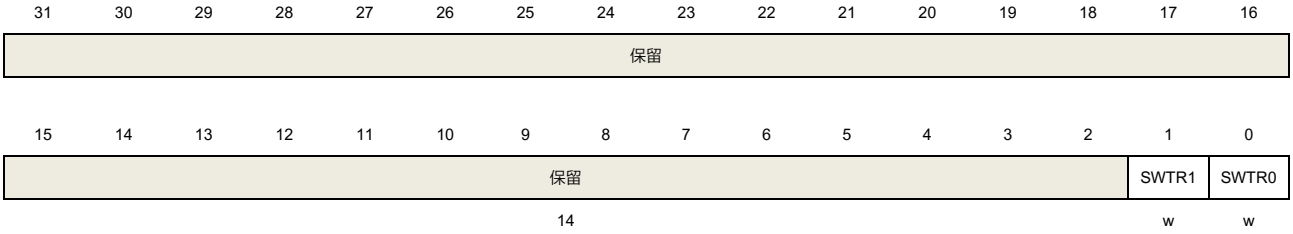
		0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 >=1011：波形信号的位宽为12 **注意**：当 DAC 使能(DEN1=1)时，该位域不能写入。
23:22	DWM1[1:0]	DACx_OUT1 噪声波模式 这些位指定了在 DACx_OUT1 外部触发使能(DTEN1=1)的情况下，DACx_OUT1 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
21	保留	必须保持复位值
20:19	DTSEL1[1:0]	DACx_OUT1 触发选择 这些位仅在 DTEN1=1 并选择用于触发 DAC 的外部事件时使用。 00：TRIGSEL 11：软件触发 其他所有值：保留
18	DTEN1	DACx_OUT1 触发使能 0：DACx_OUT1 触发禁能 1：DACx_OUT1 触发使能
17	DBOFF1	DACx_OUT1输出缓冲区关闭 0：DACx_OUT1输出缓冲区打开，以降低输出阻抗，提高驱动能力 1：DACx_OUT1输出缓冲区关闭
16	DEN1	DACx_OUT1 使能 0：DACx_OUT1 禁能 1：DACx_OUT1 使能
15	保留	必须保持复位值
14	DCSEL0	DACx_OUT0 与 GPIO 的连接选择 0：DACx_OUT0与GPIO连接 1：DACx_OUT0不与GPIO连接
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 **注意**：当DAC使能(DEN0=1)时，该位域不能写入。
7:6	DWM0[1:0]	DACx_OUT0 噪声波模式 这些位指定了在 DACx_OUT0 外部触发使能(DTEN0=1)的情况下，DACx_OUT0 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
5	保留	必须保持复位值
4:3	DTSEL0[1:0]	DACx_OUT0 触发选择 这些位仅在 DTEN0=1 并选择用于触发 DAC 的外部事件时使用。 00：TRIGSEL 11：软件触发 其他所有值：保留
2	DTEN0	DACx_OUT0 触发使能 0：DACx_OUT0 触发禁能 1：DACx_OUT0触发使能
1	DBOFF0	DACx_OUT0输出缓冲区关闭 0：DACx_OUT0输出缓冲区打开，以降低输出阻抗，提高驱动能力 1：DACx_OUT0输出缓冲区关闭
0	DEN0	DACx_OUT0 使能 0：DACx_OUT0 禁能 1：DACx_OUT0使能

19.4.2. 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。

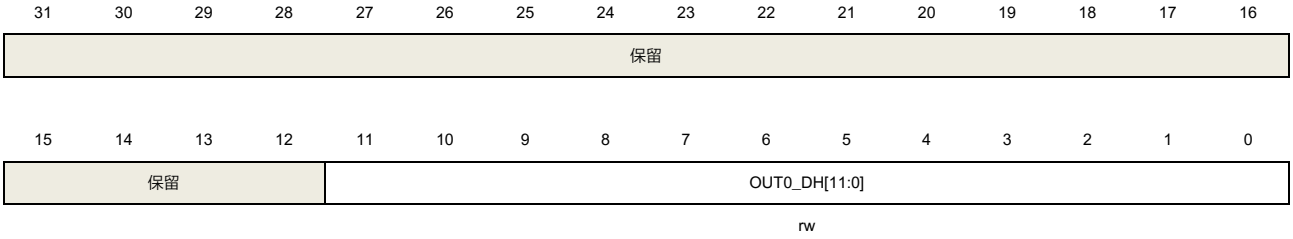


位/位域	名称	描述
31:2	保留	必须保持复位值
1	SWTR1	DACx_OUT1 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能
0	SWTR0	DACx_OUT0 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

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

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



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

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

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

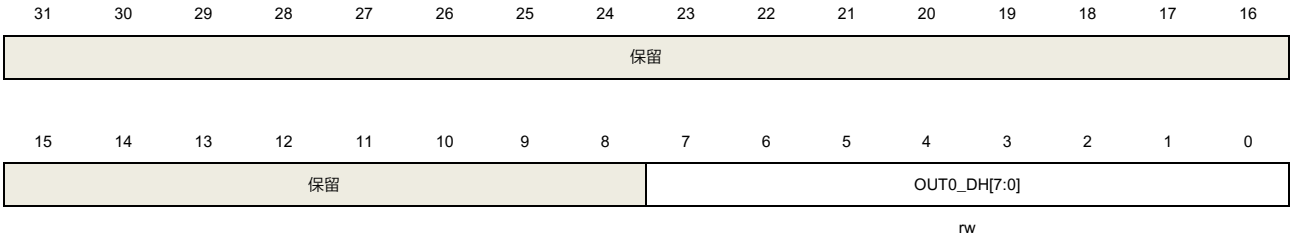


位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

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

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

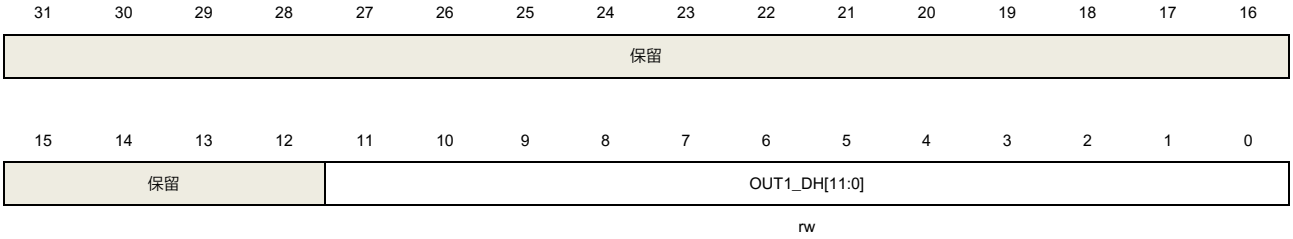


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

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

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

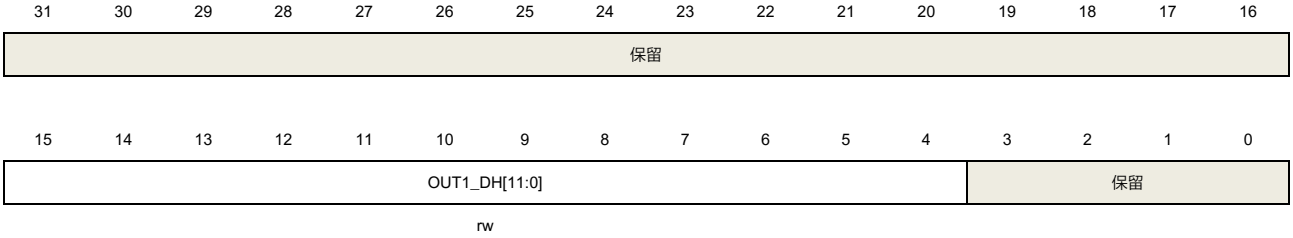


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

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

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



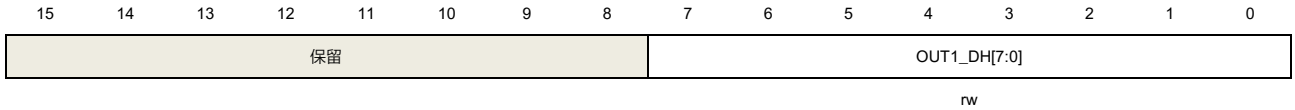
位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
3:0	保留	必须保持复位值

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

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



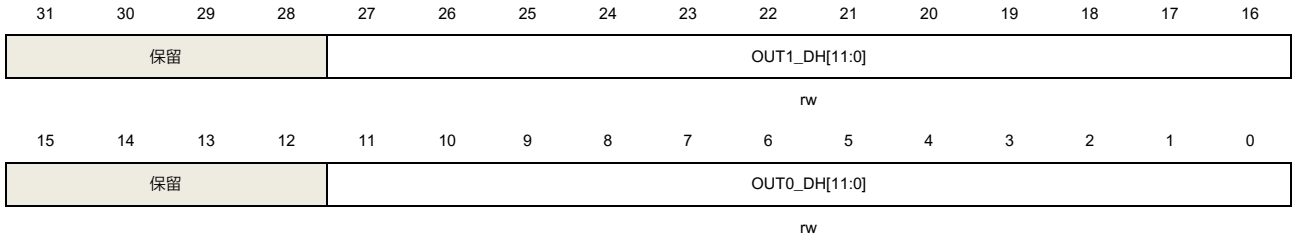


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的最高 8 位有效位。

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

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

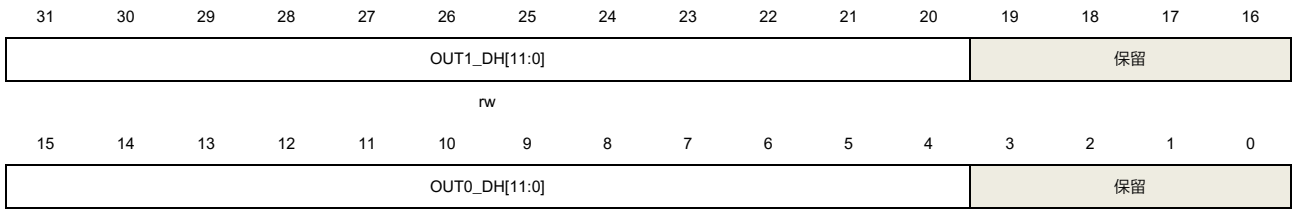


位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
15:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

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

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



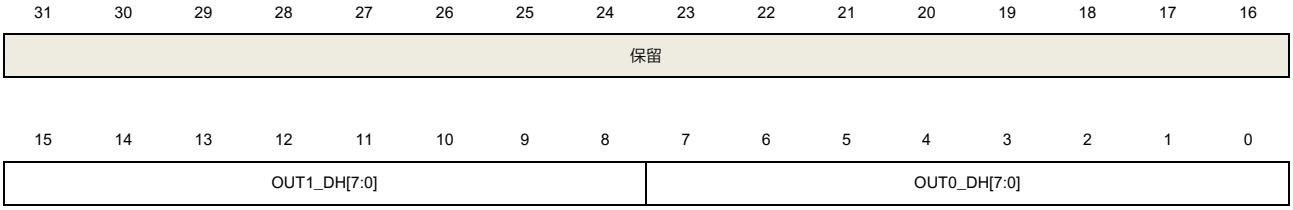
rw

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

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

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

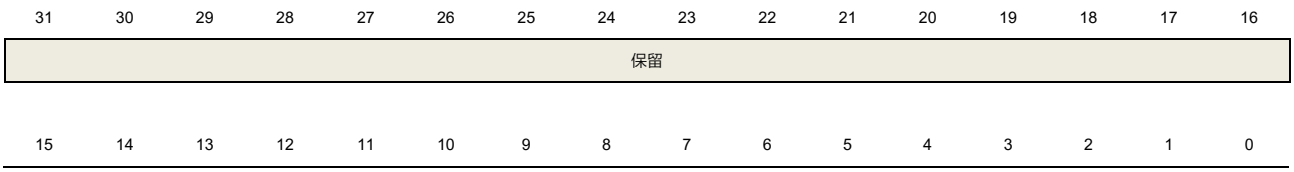


位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的 8 位最高有效位。

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

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



保留	OUT0_DO [11:0]
	r

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

19.4.13. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				OUT1_DO [11:0]											
				r											

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

19.4.14. 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
保留		DDUDR1	保留												
		rc_w1													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DDUDR0	保留												
		rc_w1													

位/位域	名称	描述
31:30	保留	必须保持复位值

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

19.4.15. DACx 校准寄存器 (DAC_CALR)

地址偏移：0x38
复位值：0x0210 0210

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CALEN1	CALF1	OTV1_P[4:0]					OTV1_N[4:0]				
				rw	r	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CALEN0	CALF0	OTV0_P[4:0]					OTV0_N[4:0]				
				rw	r	rw					rw				

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CALEN1	DACx_OUT1 校准使能 0：DACx_OUT1 校准模式禁能 1：DACx_OUT1 校准模式使能 注意：只有 DAC_CTL0 寄存器的 DEN1=0 时，才可对 CALEN1 写 1。
26	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
25:21	OTV1_P[4:0]	DACx_OUT1 偏移校准值 P 默认是工厂校准值 如果配置此寄存器，则为用户校准模式，如下 00000：偏移校准值为 0 00001：偏移校准值为 1 …… 11111：偏移校准值为 31

20:16	OTV1_N[4:0]	DACx_OUT1 偏移校准值 N 默认是工厂校准值 如果配置此寄存器，则为用户校准模式，如下 00000：偏移校准值为 0 00001：偏移校准值为 1 …… 11111：偏移校准值为 31
15:12	保留	必须保持复位值
11	CALEN0	DACx_OUT0 校准使能 0：DACx_OUT0 校准模式禁能 1：DACx_OUT0 校准模式使能 注意：只有 DAC_CTL0 寄存器的 DEN0=0 时，才可对 CALEN0 写 1。
10	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
9:5	OTV0_P[4:0]	DACx_OUT0 偏移校准值 P 默认是工厂校准值 如果配置此寄存器，则为用户校准模式，如下 00000：偏移校准值为 0 00001：偏移校准值为 1 …… 11111：偏移校准值为 31
4:0	OTV0_N[4:0]	DACx_OUT0 偏移校准值 N 默认是工厂校准值 如果配置此寄存器，则为用户校准模式，如下 00000：偏移校准值为 0 00001：偏移校准值为 1 …… 11111：偏移校准值为 31

20. 比较器(CMP)

20.1. 简介

通用比较器 CMP 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 和定时器的 PWM 输出，可以实现电流控制。

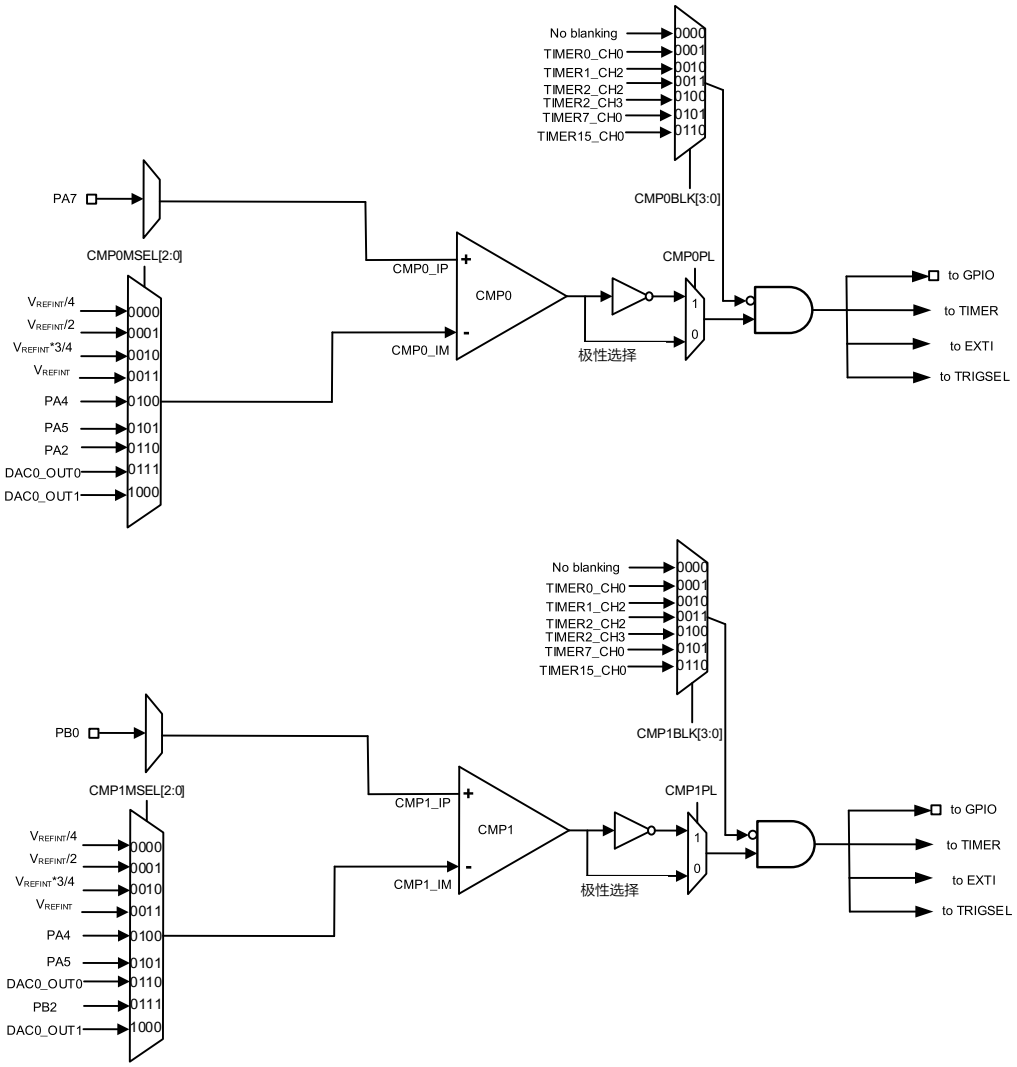
20.2. 主要特征

- 轨对轨比较器；
- 迟滞可配置；
- 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；
  - 内部参考电压的整体或分数倍值；
- 比较器输出消隐；
- 输出到I/O口；
- 作为触发源输出到定时器；
- 输出到EXTI；
- 输出到NVIC；
- 输出到TRIGSEL。

20.3. 功能描述

*图 20-1.比较器框图*展示如下：

图 20-1.比较器框图



注意：V_{REFINT} 是 1.2V。

20.3.1. 比较器时钟

比较器的时钟与 PCLK 同步，它有专用的复位和时钟使能位。

20.3.2. 比较器的 I/O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

比较器的输出可同时实现内部和外部输出。

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

比较器输出内部连接到定时器，他们的连接关系如下：

■ CMP 输出连接到定时器中止功能。

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 20-1. 输入和输出汇总*详细描述了 CMP 的输入和输出。

表 20-1. 输入和输出汇总

	CMP0	CMP1
CMP 同相输入连接到 I/O	PA7	PB0
CMP 反相输入连接到 I/O	PA4 PA5 PA2	PA4 PA5 PB2
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	PA12 PA2 PB9 PE13 PE8	PA10 PB1 PE12 PE9
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 TIMER break	•	•

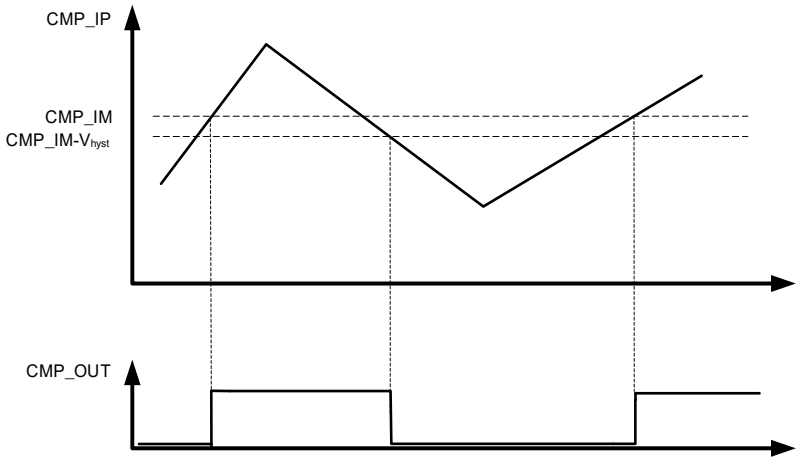
注意：

1. 有关CMP输出连接到TIMER break的详细信息，请参阅TIMER模块；

20.3.3. 比较器迟滞

为了避免噪声信号所引起的假输出，电路设计了可编程的迟滞功能，通过配置 CMPx_CS 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 20-2. 比较器迟滞



20.3.4. 比较器寄存器写保护

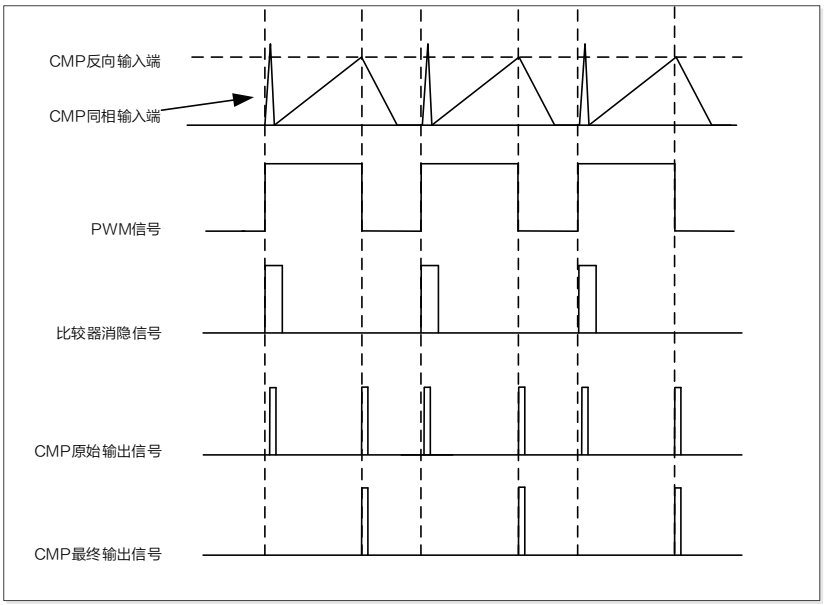
比较器的控制状态寄存器(CMPx_CS)可通过设置 CMP0LK 位为 1 来进行保护，CMPx_CS 寄存器，包含 CMPxLK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

20.3.5. 比较器输出消隐

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

*图 20-3. 比较器的输出消隐*显示了比较器的输出消隐功能。

图 20-3. 比较器的输出消隐



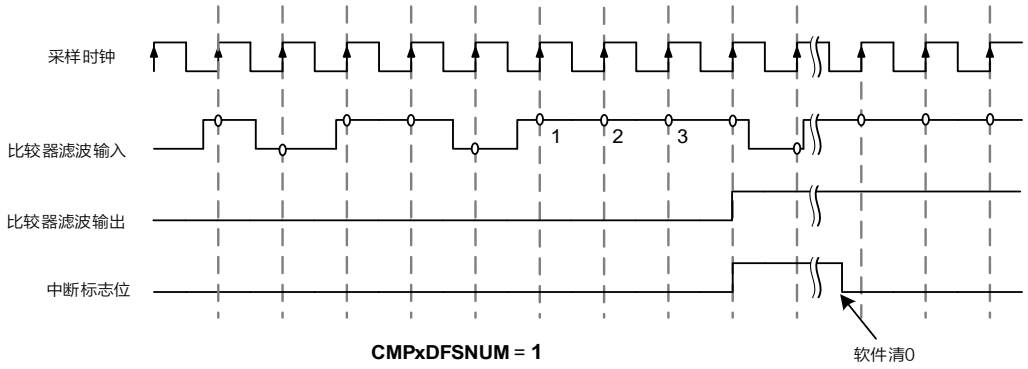
20.3.6. 比较器数字滤波

数字滤波器的采样时钟可由 CMPxDFSCDIV[1:0]位选择，匹配次数可由 CMPx_CS 寄存器的 CMPxDFSNUM 位选择。数字滤波器函数的作用如下：

- 可选择三种采样周期之一。
- 也可以关闭过滤功能。
- 经过噪声滤波的信号可用于产生中断请求输出、POE 源输出、GPT 内部触发源输出和比较结果输出。

每个采样时钟对比较器输出进行采样，如果对相同的值进行二次或三次采样，则将该值确定为下一个采样时钟的噪声滤波器输出。数字噪声滤波器的细节如图*图 20-4. 比较器噪声滤波和中断操作*所示。

图 20-4. 比较器噪声滤波和中断操作



注意:

上面的操作示例是使能数字噪声滤波，适用于 CMPxDFSCDIV[1:0]位为 2'b01, 2'b10 和 2'b11，CMPxDFSNUM 位为 1 的情况。通过设置 CMPxDFSCDIV[1:0]位的值来选择采样时间，可以改变消噪宽度。

20.3.7. 电压定标器功能

电压定标器功能可为 CMP 输入提供 1 / 4、1 / 2、3 / 4 参考电压供选择。它由 CMPx_CS 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 VREFINT 电压输出和分压电路，以产生所选择的电压

20.3.8. 比较器中断

CMP 输出连接到 EXTI，EXTI 线对每个 CMP 都是独占的。通过这个功能，可以产生中断或者事件，用于退出省电模式或者睡眠模式。

CMP 还可以产生对 NVIC 的中断。它是一个序列逻辑信号，因此需要 PCLK。

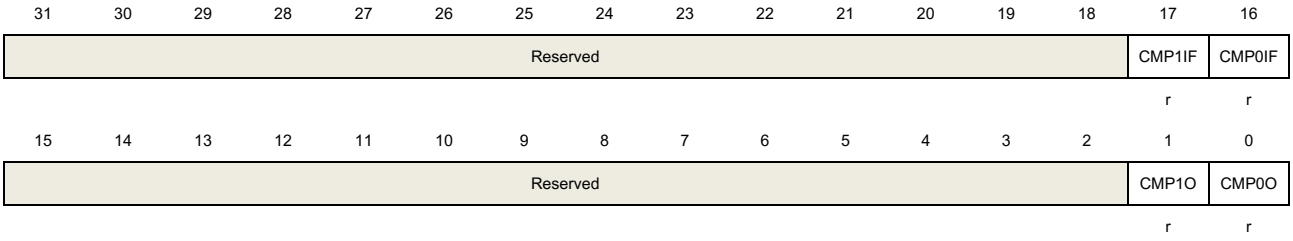
20.4. 比较器寄存器

CMP 基地址：0x4001 7C00

20.4.1. 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。

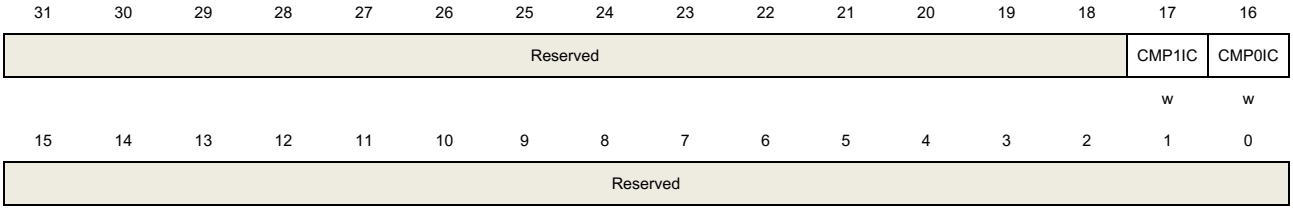


位/位域	名称	描述
31:18	保留	必须保持复位值。
17	CMP1IF	CMP1中断标志位 0：无CMP1中断发生 1：CMP1中断发生 当CMP1输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP1IC位清0。
16	CMP0IF	CMP0中断标志位 0：无CMP0中断发生 1：CMP0中断发生 当CMP0输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP0IC位清0。
15:2	保留	必须保持复位值。
1	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平
0	CMP0O	CMP0 输出 该位反映 CMP0 输出状态，是只读位 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平

20.4.2. 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。

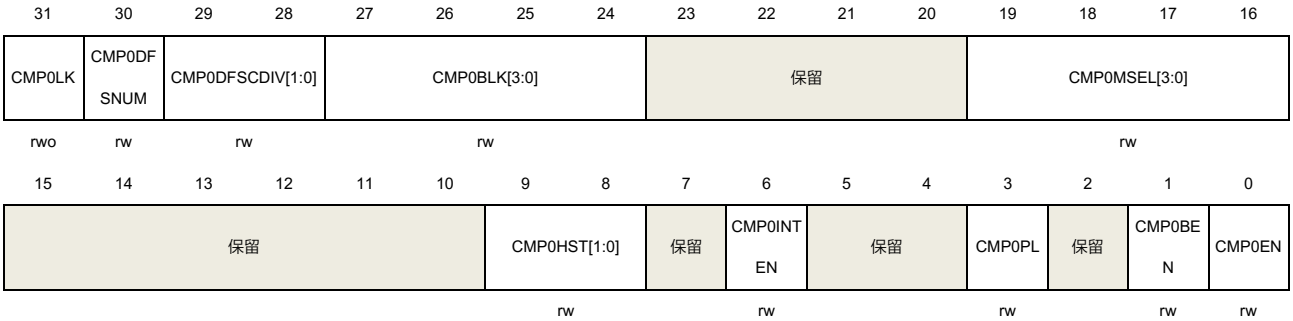


位/位域	名称	描述
31:18	保留	必须保持复位值。
17	CMP1IC	CMP1中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
16	CMP0IC	CMP0中断标志位清除 0：无清除中断标志位发生 1：清除中断标志位
15:0	保留	必须保持复位值。

20.4.3. CMP0 控制状态寄存器(CMP0_CS)

地址偏移：0x08
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	CMP0LK	CMP0 写保护 该位可将 CMP0 的各控制位设为只读。该位只能写入一次。一旦由软件写入后，只能通过系统复位来清除。

		0：CMP0_CS 是可读可写位 1：CMP0_CS 是只读位
30	CMP0DFSNUM	噪声滤波模式选择 0：同一值采样 2 次，输出发生变化。 1：同一值采样 3 次，输出发生变化
29:28	CMP0DFSCDIV[1:0]	CMP0 数字噪声采样选择。 00：未使用噪声滤波器 01：采样频率为 PCLK / 8。 10：采样频率为 PCLK / 16。 11：采样频率为 PCLK / 32。
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。
23:20	保留	必须保持复位值。
19:16	CMP0MSEL[3:0]	CMP0_IM 输入选择 该位域用来选择 CMP0 的 CMP0_IM 输入源选择。 0000：V_{REFINT} / 4 0001：V_{REFINT} / 2 0010：V_{REFINT}*3 / 4 0011：V_{REFINT} 0100: PA4 0101: PA5 0110: PA2 0111: DAC0_OUT0 1000: DAC0_OUT1 其它：保留。
15:10	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 该位是用来控制 CMP0 输出极性 0：输出是正相的 1：输出是反相的
2	保留	必须保持复位值。
1	CMP0BEN	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	CMP0EN	CMP0 使能 0：CMP0 禁能 1：CMP0 使能

20.4.4. CMP1 控制状态寄存器(CMP1_CS)

地址偏移：0x10
复位值：0x8000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP1LK	CMP1DFSNUM	CMP1DFSCDIV[1:0]		CMP1BLK[3:0]				保留				CMP1MSEL[3:0]			
rwo	rw	rw		rw								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CMP1HST[1:0]		保留	CMP1INTEN	保留		CMP1PL	保留	CMP1BEN	CMP1EN
						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。该位只能写入一次。一旦由软件写入后，只能通过系统复位来清除。 0：CMP1_CS 是可读可写位 1：CMP1_CS 是只读位

30	CMP1DFSNUM	噪声滤波模式选择 0：同一值采样 2 次，输出发生变化。 1：同一值采样 3 次，输出发生变化
29:28	CMP1DFSCDIV[1:0]	CMP1 数字噪声采样选择。 00：未使用噪声滤波器 01：采样频率为 PCLK / 8。 10：采样频率为 PCLK / 16。 11：采样频率为 PCLK / 32。
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 1 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。
23:20	保留	必须保持复位值。
19:16	CMP1MSEL[3:0]	CMP1_IM 输入选择 该位域用来选择 CMP1 的 CMP1_IM 输入源选择。 0000：V_{REFINT} / 4 0001：V_{REFINT} / 2 0010：V_{REFINT}*3 / 4 0011：V_{REFINT} 0100: PA4 0101: PA5 0110: DAC0_OUT0 0111: PB2 1000: DAC0_OUT1 其它：保留。
15:10	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
5:4	保留	必须保持复位值。
3	CMP1PL	CMP1 输出极性 该位是用来控制 CMP1 输出极性 0：输出是正相的 1：输出是反相的
2	保留	必须保持复位值。
1	CMP1BEN	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	CMP1EN	CMP1 使能 0：CMP1 禁能 1：CMP1 使能

21. 看门狗定时器(WDGT)

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

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

21.1. 独立看门狗定时器(FWDGT)

21.1.1. 简介

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

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

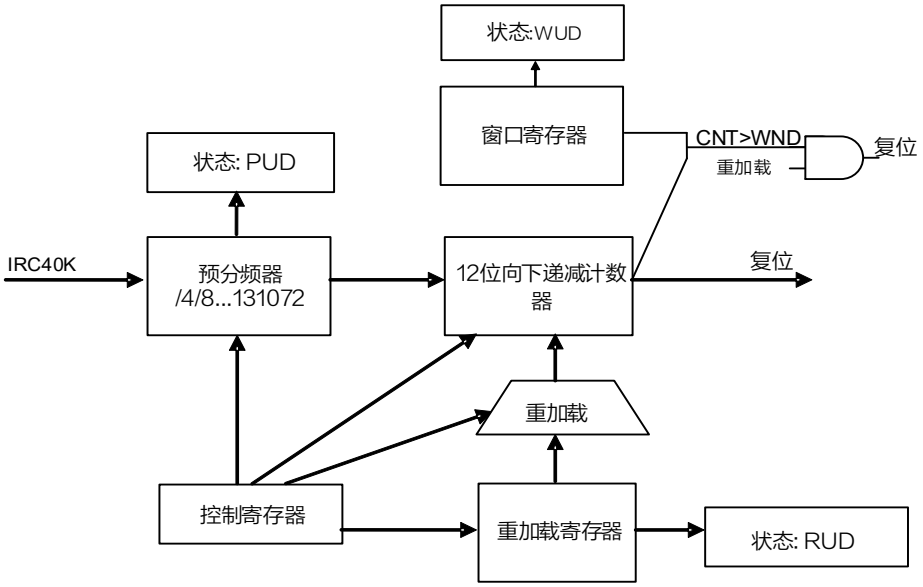
21.1.2. 主要特性

- 自由运行的12位向下计数器；
- 使能独立看门狗定时器，以下两种情况下将产生复位：
  - 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，深度睡眠模式1和深度睡眠模式下)时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

21.1.3. 功能描述

独立看门狗定时器带有一个 8 级预分频器和一个 12 位的向下递减计数器。*图 21-1. 独立看门狗定时器框图*为独立看门狗模块的功能框图。

图 21-1. 独立看门狗定时器框图



向控制寄存器(FWDGT_CTL)中写 0xCCCC 可以开启独立看门狗定时器，计数器开始向下计数。当计数器记到 0x000，产生一次复位。在任何时候 FWDGT_CTL 中写 0xAAAA 都可以重装载计数器，重装载值来源于重装载寄存器(FWDGT_RLD)。软件在计数器计数值达到 0x000 之前可以通过重装载计数器来阻止看门狗定时器产生系统复位。

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

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

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

如果 DBG 控制寄存器 0(DBG_CTL0)中的 FWDGT_HOLD 位被清 0，即使 Cortex®-M33 内核停止(调试模式下)独立看门狗定时器依然工作。如果 FWDGT_HOLD 位被置 1，独立看门狗定时器将在调试模式下停止工作。

表 21-1. 独立看门狗定时器在 40kHz(IRC40K)时的最小/最大超时周期

预分频系数	PSC[3:0] 位	最小超时(ms) RLD[11:0]=0x000	最大超时(ms) RLD[11:0]=0xFFF
1 / 4	0000	0.1	409.6
1 / 8	0001	0.2	819.2

预分频系数	PSC[3:0] 位	最小超时(ms) RLD[11:0]=0x000	最大超时(ms) RLD[11:0]=0xFFF
1 / 16	0010	0.4	1638.4
1 / 32	0011	0.8	3276.8
1 / 64	0100	1.6	6553.6
1 / 128	0101	3.2	13107.2
1 / 256	0110	6.4	26214.4
1 / 512	0111	12.8	52428.8
1 / 1024	1000	25.6	104857.6
1 / 2048	1001	51.2	209715.2
1 / 4096	1010	102.4	419430.4
1 / 8192	1011	204.8	838860.8
1 / 16384	1100	409.6	1677721.6
1 / 32768	1101	819.2	3355443.2
1 / 65536	1110	1638.4	6710886.4
1 / 131072	1111	3276.8	13421772.8

通过校准 IRC40K 可以使独立看门狗定时器超时更加精确。

注意:

1.当执行完喂狗 reload 操作之后，如需要立即进入深度睡眠 / 深度睡眠 1 / 待机模式时，必须通过软件设置，在 reload 命令及深度睡眠 / 深度睡眠 1 / 待机模式命令中间插入 3 个以上 IRC40K 时钟间隔。

2.两次连续喂狗之间需插入 7 个及以上 IRC40K 时间间隔。

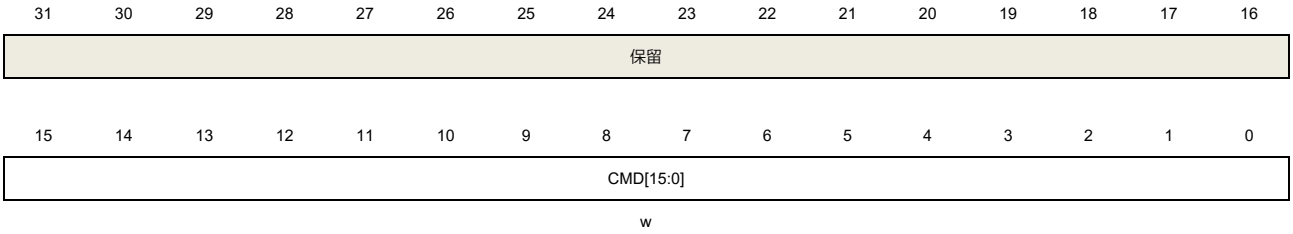
21.1.4. FWDGT 寄存器

FWDGT 基地址：0x4000 3000

控制寄存器(FWDGT_CTL)

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字(32 位)访问。

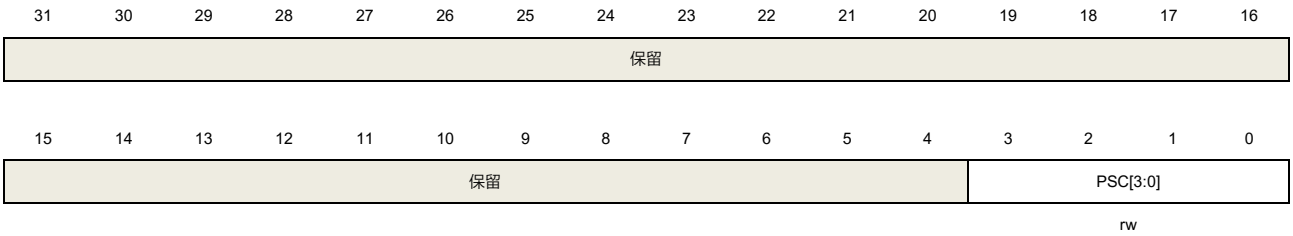


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC、FWDGT_RLD和FWDGT_WND的写保护 0xCCCC：开启独立看门狗定时器定时计数器。计数减到0时产生复位 0xAAAA：重装载计数器

预分频寄存器(FWDGT_PSC)

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字(32 位)访问。



位/位域	名称	说明
31:4	保留	必须保持复位值。
3:0	PSC[3:0]	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。

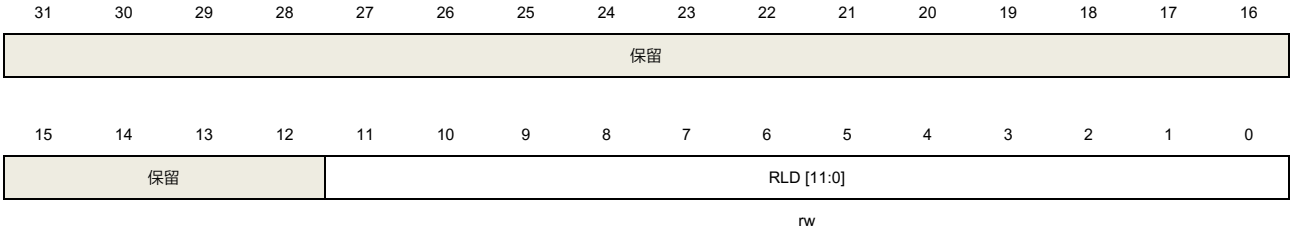
000：1 / 4
001：1 / 8
010：1 / 16
011：1 / 32
100：1 / 64
101：1 / 128
110：1 / 256
111：1 / 256
……
1100: 1/16384
1101: 1/32768
1110: 1/65536
1111: 1/131072

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

重装载寄存器(FWDGT_RLD)

地址偏移：0x08
复位值：0x0000 0FFF

该寄存器可以按字(32 位)访问。

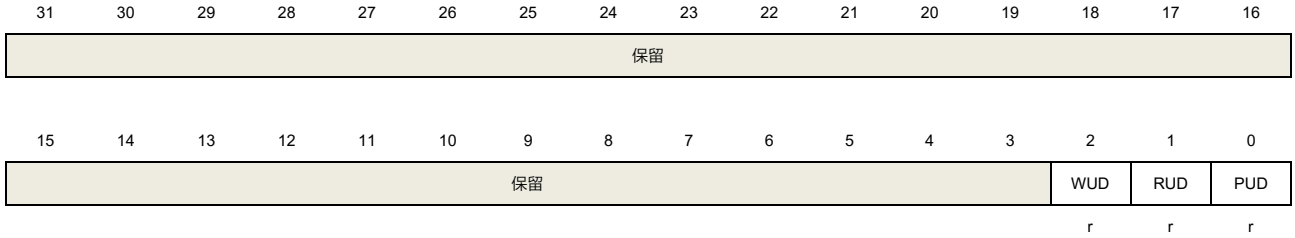


位/位域	名称	说明
31:12	保留	必须保持复位值
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值。向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到独立看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载值后，在代码持续执行之前不必等待RUD值被清零(在进入省电模式前需等待RUD值清零)。

状态寄存器(FWDGT_STAT)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字(32 位)访问。

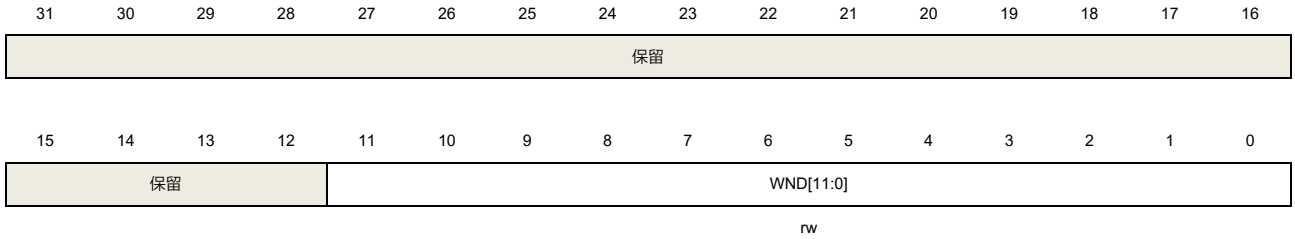


位/位域	名称	说明
31:2	保留	必须保持复位值。
2	WUD	独立看门狗定时器计数器窗口值更新 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。
0	PUD	独立看门狗定时器预分频值更新 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。

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[11:0]	独立看门狗定时器计数器窗口值。这些位用来将窗口值的上限值与向下递减计数器进

行比较。当计数值大于WND[11:0]中值，重装载操作会引起复位。若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。

这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。

如果应用需要使用几个窗口值，改变窗口值之前必须等到WUD位被清0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待WUD值被清零。

21.2. 窗口看门狗定时器(WWDGT)

21.2.1. 简介

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

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

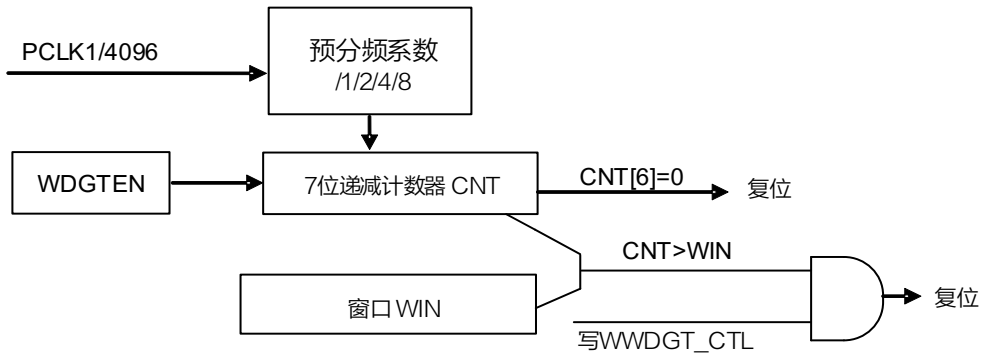
21.2.2. 主要特性

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

21.2.3. 功能描述

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系统复位。

图 21-2. 窗口看门狗定时器框图



上电复位之后窗口看门狗定时器总是关闭的。软件可以向 WWDGT_CTL 的 WDGTEN 写 1 开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该

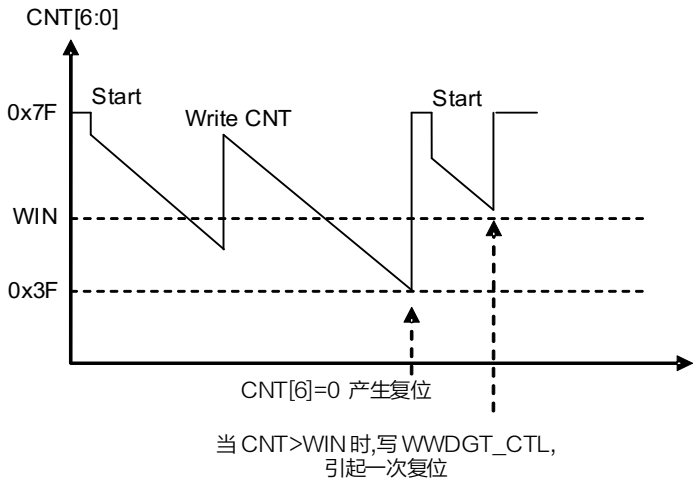
大于 0x3F，也就是说 CNT[6]位应该被置 1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于 APB1 时钟和预分频器(WWDGT_CFG 寄存器的 PSC[4:0]位)。

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 中断标志。

图 21-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (21\text{-}1)$$

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK1}：APB1 以 ms 为单位的时钟周期

下表展示 PCLK1 在 100MHZ 时 t_{WWDGT} 的最大值和最小值。

表 21-2. 在 100MHz(f_{PCLK1})时的最大/最小超时值

预分频系数	PSC[4:0]	最小超时(ms) CNT[5:0] =0x00	最大超时(ms) CNT[5:0]=0x3F
1/1	00000	0.041	2.621
1/2	00001	0.082	5.243
1/4	00010	0.164	10.486

1/8	00011	0.328	20.972
1/16	00100	0.655	41.943
1/32	00101	1.311	83.886
1/64	00110	2.621	167.772
1/128	00111	5.243	335.544
1/256	01000	10.486	671.089
1/512	01001	20.972	1342.177
1/1024	01010	41.943	2684.355
1/2048	01011	83.886	5368.709
1/4096	01100	167.772	10737.418
1/8192	01101	335.544	21474.836
1/16384	01110	671.089	42949.673
1/32768	01111	1342.177	85899.346
1/65536	10000	2684.355	171798.692
1/131072	10001	5368.709	343597.384
1/262144	>=10010	10737.418	687194.767

如果 MCU 调试模块中的 WWDGT_HOLD 位被清 0，即使 Cortex®-M33 内核停止工作(调试模式下)，窗口看门狗定时器也可以继续工作。当 WWDGT_HOLD 位被置 1 时，窗口看门狗定时器在调试模式下停止工作。

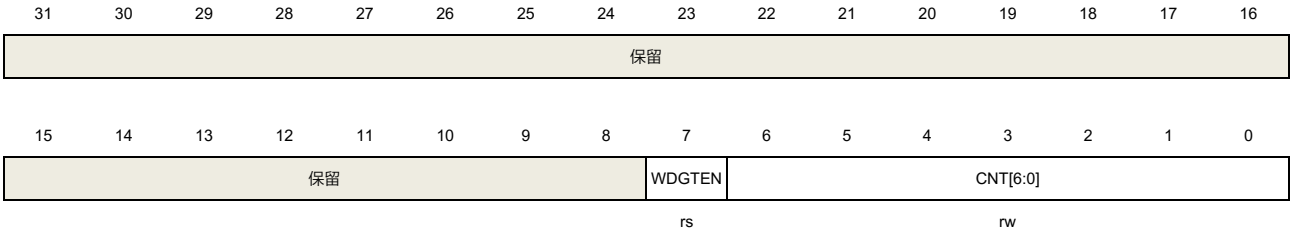
21.2.4. WWDGT 寄存器

WWDGT 基地址：0x4000 2C00

控制寄存器(WWDGT_CTL)

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按字(32 位)访问。

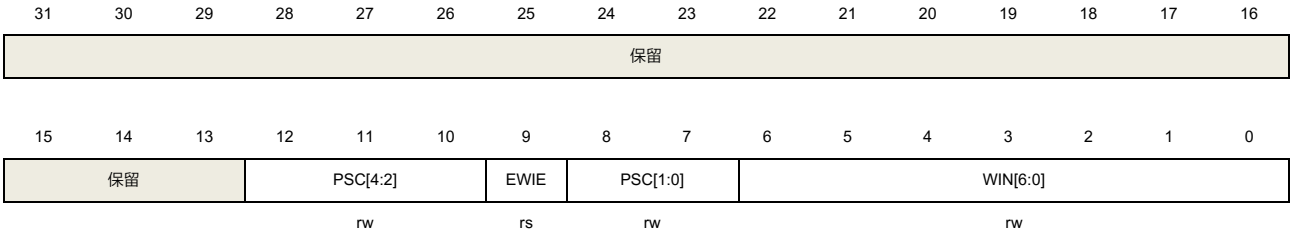


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，可以通过硬件复位或软件时钟复位来(参考***APB1使能寄存器(RCU_APB1EN)***)清0。写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器系统复位。

配置寄存器(WWDGT_CFG)

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按字(32 位)访问。



位/位域	名称	说明
31:10	保留	必须保持复位值。

12:10	PSC[4:2]	预分频器，看门狗定时器计数器的时间基准 00000：PCLK1 / 4096 / 1 00001：PCLK1 / 4096 / 2 00010：PCLK1 / 4096 / 4 00011：PCLK1 / 4096 / 8 …… 10010: (PCLK1 / 4096) / 262144 >=10010: (PCLK1 / 4096) / 262144
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，或通过置位RCU模块的WWDGTRST位进行软件复位。写0没有任何作用。
8:7	PSC[1:0]	预分频值低位。 PSC[4:0]由高位值PSC[4:2]和低位值PSC[1:0]组成。 PSC[4:0]是看门狗计数器的时基。
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生系统复位。

状态寄存器(WWDGT_STAT)

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字(32 位)访问。

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

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

位/位域	名称	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，即使中断没有被使能(WWDGT_CFG中的EWIE位为0)该位也会被硬件置1。这个位可以通过写0清零，写1无效。

22. 实时时钟(RTC)

22.1. 简介

RTC 模块提供了一个包含日期(年/月/日)和时间(时/分/秒/亚秒)的日历功能。除亚秒用二进制码显示外，时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下，并通过软件配置来智能唤醒。RTC 支持外接更高精度的低频时钟，用以达到更高的日历精度。

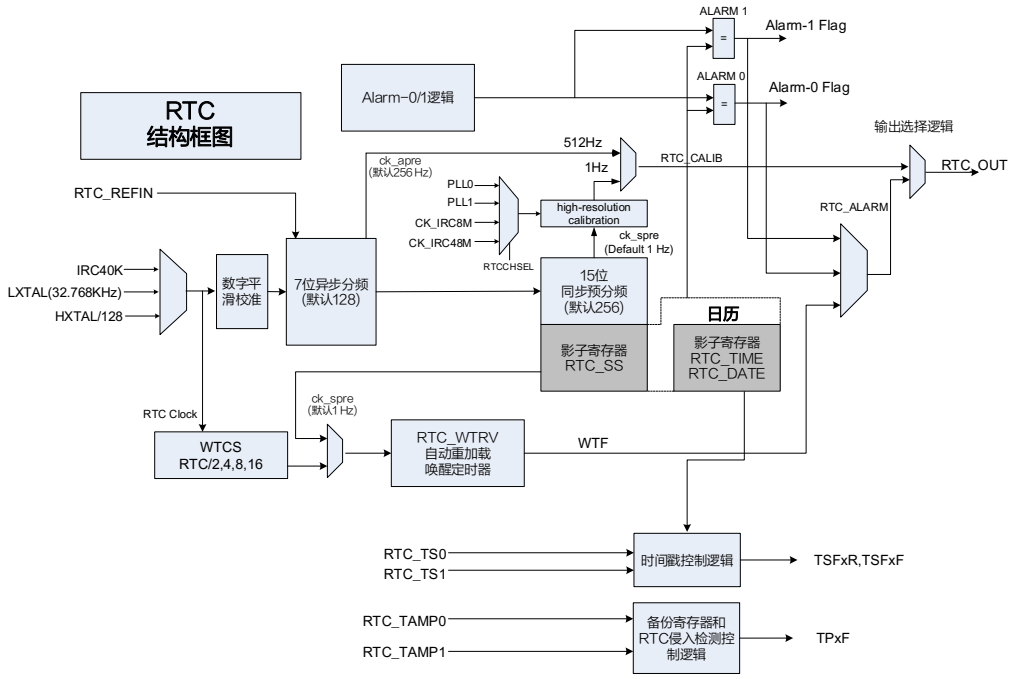
22.2. 主要特征

- 支持日历功能，可支持年、月、日(月的天数)、日(星期几)、时、分、秒和亚秒。
- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能：通过外接更高精度的低频率时钟源(50Hz或60Hz)来提高日历精度。
- 数字校准功能：通过调整最小时间单位(平滑数字校准精度为0.95ppm，高精度校准精度为0.0625PPM)来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 两个独立的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
- 4个32位(共128字节)通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

22.3. 功能描述

22.3.1. 结构框图

图 22-1. RTC 结构框图



RTC 单元包括：

- 两个闹钟事件/中断，支持两个来自I/O引脚的闹钟事件和中断。
- 侵入事件/中断。
- 当检测到侵入事件时将擦除备份寄存器。
- 当电源切换到V_{BAT}时将产生一个时间戳事件。
- 4个32位备份寄存器。
- 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 0 事件检测：RTC_TS0(PC13)；
  - 时间戳 1 事件检测：RTC_TS1(PD4)；
  - 侵入事件检测 0：RTC_TAMP0(PC13)；
  - 侵入事件检测 1：RTC_TAMP1(PD4)；
  - 参考时钟输入：RTC_REFIN(PB15 或 PB13)。

22.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源：LXTAL、IRC40K 和 HXTAL 经过 128 分频后的时钟。

在 RTC 单元，有两个预分频器用来实现日历功能和其他功能。一个分频器是 7 位异步预分频器，另一个是 15 位同步预分频器。异步分频器主要用来降低功率消耗。如果两个分频器都被使用，建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下：

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \tag{22-1}$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1)*(FACTOR_S + 1)} \tag{22-2}$$

ck_apre 用于为 RTC_SS 亚秒寄存器自减计数器提供时钟，该寄存器值为二进制，表示到达下一秒时间，该寄存器自减到 0 时，自动加载 FACTOR_S[14:0]的值。ck_spre 用于为日历寄存器提供时钟，每个时钟增加一秒。

注：当 RTC 采用 HXTAL 的分频时钟作为工作时钟源时，系统复位会导致 HXTAL 时钟使能复位，RTC 无法正常工作，需要重新配置时钟后才能正常工作。

22.3.3. 影子寄存器

当 APB 总线访问 RTC 日历寄存器 RTC_DATE、RTC_TIME 和 RTC_SS 时，BPSHAD 位决定是访问影子寄存器还是真实日历寄存器。默认情况下 BPSHAD 为 0，APB 总线访问影子日历寄存器。每两个 RTC 时钟，影子日历寄存器值会更新为真实日历寄存器的值，与此同时 RSYNF 位也会再次置位。在 Deep-sleep 和 Standby 模式下，影子寄存器不会更新。退出这两种模式时，软件必须清除 RSYNF 位。如果想要在 BPSHAD=0 的情况下读日历寄存器的值，须等待 RSYNF 置 1(最大的等待时间是 2 个 RTC 时钟周期)。

注意：在 BPSHAD=0 下，读日历寄存器(RTC_SS，RTC_TIME，RTC_DATE)的 APB 时钟的频率(f_{apb})必须至少是 RTC 时钟频率(f_{rtcclk})的七倍。

系统复位将复位影子寄存器。

22.3.4. 位域可屏蔽可配置的闹钟

RTC 闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC 闹钟功能的使能由 RTC_CTL 寄存器中的 ALRMxEN(x=0,1)位控制。当 ALRMxEN=1(x=0,1)并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF(x=0,1)标志位将会置位。

注意：当秒字段未被屏蔽时(RTC_ALRMxTD 寄存器的 MSKS=0)，为确保正常运行，RTC_PSC 寄存器的同步预分频系数(FACTOR_S[14:0])应大于等于 3。

如果一个位域被屏蔽，这个位域被认为在逻辑上匹配的。如果所有的位域被屏蔽，在 ALRMxEN

位被置位 3 个 RTC 时钟周期后，ALRMxF 位将置位。

22.3.5. 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。

注意：AM/PM(RTC_TIME 寄存器中的 PM 位)可以在 24 小时模式下配置，但退出初始化模式后，实际日历寄存器不会加载 AM/PM 的值。退出初始化模式后，AM/PM 的值将在两个 RTC 时钟周期后被重置，并且 RTC_STAT 寄存器中的 RSYNF 位将被设置。

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。

注意：24 小时格式为 0:00~1:00,12 小时格式为 12:00~1:00 AM(PM=0)时不能执行 S1H 操作。

此时执行的操作将不生效。

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器 RTC_CTL 的 ALRMxEN(x=0,1)位，禁用闹钟；
2. 设置 Alarm 寄存器(RTC_ALRMxTD/RTC_ALRMxSS(x=0,1))；
3. 设置寄存器 RTC_CTL 的 ALRMxEN(x=0,1)位，使能闹钟功能。

22.3.6. 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，精度低至 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，精度低至 1s。

- WTCS[2:1] = 0b10，唤醒中断周期在 1s 到 18 h。
- WTCS[2:1] = 0b11，唤醒中断周期在 18h 到 36 h。

该功能使能后，计数器自动递减。当计数器到 0 时，WTF 标志位置 1，唤醒计数器自动重载 RTC_WUT 的值。

当 WTF 置 1 后，必须软件清除该标志。

如果 WTIE 被置位，计数器到 0 时，会产生唤醒中断，从而使系统退出省电模式。系统复位对该功能没有影响。

WTF 标志可以从 RTC_ALARM 通道输出到 RTC_OUT。

22.3.7. 读取日历

当 BPSHAD=0 时，读日历寄存器

当 BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB 总线时钟频率必须大于或等于 RTC 时钟频率的 7 倍。在任何情况下 APB 总线时钟的频率都不能低于 RTC 的时钟频率。

当 APB 总线时钟频率低于 7 倍 RTC 时钟频率时，日历的读取应该遵守以下流程：

1. 读取两次日历时间和日期寄存器；
2. 如果两次的值相等，那么这个值就是正确的；
3. 如果这两次的值不相等，应该再读一次；
4. 第三次的值可以认为是正确的。

RSYNF 每 2 个 RTC 时钟周期被置位一次。在这时，影子日历寄存器会更新为真实的日历时间和日期。

为了确保这 3 个值(RTC_SS，RTC_TIME，RTC_DATE)为同一时间，硬件上采取了如下一致性机制：

1. 读 RTC_SS 锁定 RTC_TIME 和 RTC_DATE 的更新；
2. 读 RTC_TIME 锁定 RTC_DATE 的更新；
3. 读 RTC_DATE 解锁 RTC_TIME 和 RTC_DATE 的更新。

如果想在一个很短的时间间隔内(少于 2 个 RTCCLK)读取日历，应先清除 RSYNF 位并等待其置位后再读取。

下面几种情况，软件须等待 RSYNF 置位后才能读日历寄存器(RTC_SS，RTC_TIME，RTC_DATE)：

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

特别是从低功耗模式唤醒后，软件必须清除 RSYNF 位并等待 RSYNF 再次置位后才能读取日历寄存器。

当 BPSHAD=1 时，读日历寄存器

当 BPSHAD=1，RSYNF 位会被硬件清 0，读日历寄存器不需考虑 RSYNF 位。当前真实的日历寄存器值会被直接读取。如此配置的好处是当从低功耗模式(Deep-sleep/Standby 模式)唤醒后，软件可以立即获取当前日历寄存器的值而无需加入任何等待延迟(此延迟最大为 2 个 RTC

时钟周期)。

由于没有 RSYNF 位周期性的置位，如果两次读日历寄存器之间出现 ck_apre 时钟边沿，不同寄存器(RTC_SS / RTC_TIME / RTC_DATE)的值可能并非同一时刻。

另外，如果日历寄存器的值正在发生变化的时刻被 APB 总线读取，那么有可能 APB 总线读取的值是不准确的。

为了确保日历值的正确性和一致性，读取时软件须如下操作：连续读取所有日历寄存器的值两次，如果上两次的值是一样的，那么这个值就是一致的且准确的。

22.3.8. RTC 复位

在 RTC 单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效时，日历影子寄存器和 RTC_STAT 寄存器的某些位将要复位到默认值。

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 真实的日历寄存器；
- RTC 控制寄存器(RTC_CTL)；
- RTC 预分频寄存器(RTC_PSC)；
- RTC 唤醒定时器寄存器(RTC_WUT)；
- RTC 高精度频率补偿寄存器(RTC_HRFC)；
- RTC 移位控制寄存器(RTC_SHIFTCTL)；
- RTC 时间戳寄存器(RTC_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR)；
- RTC 侵入寄存器(RTC_TAMP)；
- RTC 备份寄存器(RTC_BKPx，RTC_CFG)；
- RTC 闹钟寄存器(RTC_ALRMxSS/RTC_ALRMxTD(x=0，1))。

当系统复位或者进入省电模式的时候，RTC 单元将会继续运行。但是如果备份域复位，RTC 将会停止计数并且所有的寄存器会复位。

22.3.9. RTC 移位功能

当用户有一个高精度的远程时钟而且 RTC 1Hz 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[14:0]的值。FACTOR_S[14:0]越大，调整的精度也就越高。

因为 1Hz 的时钟(ck_spre)由 FACTOR_A[6:0]和 FACTOR_S[14:0]共同产生，越高的

FACTOR_S[14:0]值就意味着越低的 FACTOR_A[6:0]值，同时越低的 FACTOR_A[6:0]意味着越高的功耗。

注意：在使用移位功能之前，软件必须检查 RTC_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。

写 RTC_SHIFTCTL 寄存器之后，RTC_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。

当 REFEN=0 时，移位操作才能正确的工作。

如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

22.3.10. RTC 参考时钟检测

RTC 参考时钟是另外一种提高 RTC 秒级精度的方法。为了使能这项功能，需要有一个相对于 LXTAL 有更高精度的外部参考时钟源(50Hz 或 60Hz)。

使能这项功能之后(REFEN=1)，每一个秒更新的时钟(1Hz)边沿将与最近的 RTC_REFIN 参考时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于 LXTAL 准确度的原因没有对齐的时候，RTC 参考时钟的检测功能会偏移 1Hz 时钟沿一点相位，使得下一个 1Hz 时钟沿和参考时钟沿对齐。

当 REFEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态，时间窗时长也不同。

当检测状态处于检测第一个参考时钟边沿时，使用 7 个 ck_apre 时长的时间窗，当检测状态处于边沿对齐操作时，使用 3 个 ck_apre 时长的时间窗。

无论使用哪一种时间窗，当参考时钟在时间窗中被检测到的时候，同步预分频计数器会被强制重载。当两个时钟(ck_spre 和参考时钟)边沿是对齐的，这个重载操作对 1Hz 日历更新没有任何影响。但是当两个时钟边沿没有对齐时，这个重载操作将会移动 ck_spre 时钟边沿，以使得 ck_spre(1Hz)时钟边沿和参考时钟边沿对齐。

当参考检测功能正在运行中但外部参考时钟消失(在 3 个 ck_apre 时长时间窗内没有发现参考时钟边沿)，日历也能通过 LXTAL 继续自动更新。如果这个参考时钟重新恢复，参考时钟检测功能会先用 7 个 ck_apre 时长时间窗口去检测参考时钟，然后用 3 个 ck_apre 时长时间窗口去调节 ck_spre(1Hz)时钟边沿。

注意：使能参考时钟检测功能之前(REFEN=1)，软件必须配置 FACTOR_A[6:0]为 0x7F，FACTOR_S[14:0]为 0xFF。

待机模式下，参考时钟检测功能不可用。

22.3.11. RTC 数字平滑校准

RTC 平滑校准是一种用于校准 RTC 频率的方法，该方法通过调整校准周期内的 RTC 时钟脉冲个数的方式来实现校准。

完成一次这种校准相当于在一次校准周期内，RTC 时钟的脉冲个数增加或者减少了一定的数目。这种校准的精度大约为 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 精度的情况下，调整范围为从-487.1ppm 到+488.5ppm。

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal}=f_{rtcclk} \times(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (22\text{-}3)$$

注意： N=20 /19 /18 (32 / 16 / 8 秒)校准时间周期。

当 FACTOR_A < 3 时校准：

当异步预分频器值(FACTOR_A[6:0])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[6:0]小于 3 时，FACTOR_S[14:0]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[14:0]应该按下面所示设置：

FACTOR_A = 2：FACTOR_S[14:0]减少 2(8189)

FACTOR_A = 1：FACTOR_S[14:0]减少 4(16379)

FACTOR_A = 0：FACTOR_S[14:0]减少 8(32759)

当 FACTOR_A[6:0]小于 3，CMSK 为 0x100，校准频率公式如下：

$$f_{cal}=f_{rtcclk} \times(1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (22\text{-}4)$$

注意： N=20 /19 /18 (32/16/8 秒)校准时间周期。

验证 RTC 校准

提供 1Hz 校准时钟的输出用于协助软件测量并验证 RTC 的精度。

在有限的测量周期内测量 RTC 的频率，最高可能发生 2 个 RTCCLK 的测量误差。

为了消除这一测量误差，测量周期应该和校准周期一致。

■ 校准周期设为32秒(默认配置)

用准确的 32 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 0.477ppm(在 32 秒周期内 0.5 个 RTCCLK)之内。

■ 校准周期设为16秒(通过设置CWND16位)

使用此配置，CMSK[0]被硬件置 0。

用准确的 16 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 0.954ppm(在 16 秒周期内 0.5 个 RTCCLK)之内。

■ 校准周期设为8秒(通过设置CWND8位)

使用此配置，CMSK[1：0]被硬件置 0。

用准确的 8 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 1.907ppm(在 8 秒周期内 0.5 个 RTCCLK)之内。

运行中重校准

当 INITF 位是 0，用下面的步骤，软件可以更新 RTC_HRFC：

1). 等待 SCPF 位置 0；

2). 写一个新的值到 RTC_HRFC 寄存器；

3). 3 个 ck_apre 时钟周期之后，新的校准设置开始生效。

多校准窗口

软件可以选择单窗口模式或多窗口模式进行校准。如果软件设置 RTC_HRFC 中的 MWE 为 0 时，仅一个校准窗口会激活。如果软件设置 RTC_HRFC 中的 MWE 为 1 时，将激活多窗口校准，当前校准窗口完成工作时，下一个校准窗口将立即启动。

软件将 MWE 置为 0 可关闭多窗口模式。当前正在运行的校准窗口不会受到关闭操作的影响。而且，在写入 MWE 为 0 时，不允许同时更改平滑校准窗口的配置和校准值。

22.3.12. RTC 高精度校准

通过使用高频时钟信号，可以对 RTC 分频时钟进行高精度校准。校准精度取决于高频时钟的频率范围(8MHz 至 31MHz)。建议使用 16MHz，此时可以实现 0.0625 PPM 的校准步长(校准精度为 0.03125 PPM)。

高精度时钟频率校准寄存器(RTC_HRCFC)和高精度校准寄存器(RTC_HRCR)共同决定了每个 ck_spre 周期内校准的 RTCCLK 时钟周期数和高频时钟周期数。

通过配置 HRC_MASK[14:0]，可以在每个 ck_spre 周期内增加 HRC_MASK[14:0]个 RTCCLK 周期(最多可达 FACTOR_S[14:0])。通过配置 RTC_HRCR 寄存器中的 HRCP 位，可以在每个

ck_spre 周期内减少 FACTOR_S+1 个 RTCCLK 周期。通过配置 HRCN[9:0]，可以根据 HRCADD 的值在每个 ck_spre 周期内增加或减少 HRCN 个高频时钟周期。高精度校准在配置 RTC_HRCR 寄存器后两个 RTCCLK 周期后生效，其状态可以通过 RTC_STAT 寄存器中的 HRCPF 位进行检查。

高精度校准是通过使用高频时钟模拟每个 ck_spre 周期的最后一个 RTCCLK 来实现的，这需要电路知道所采用的高频时钟频率与 RTCCLK 频率之间的比率。考虑到高频时钟频率可以通过系统时钟分频得到，该频率并非固定值，且系统时钟可能存在一定误差，提供了两种方法来计算高频时钟频率与 RTCCLK 频率之间的比率：

硬件计算：

用户通过 RTC_HRCFC 寄存器中的 HRCF[14:0]位直接将高频时钟频率值写入电路(单位：KHz)。硬件电路通过公式 HRCF[14:0]/32768 计算频率比率。

软件计算：

使用定时器或外部设备精确测量高频时钟与 RTCCLK 的频率比率，将比率值写入 RTC_HRCFC 寄存器中的 HRCR[9:0]位，并启用 RTC_HRCFC 中的 HRCRV 位。当高精度补偿激活时，电路从寄存器加载时钟频率比率。

注意：在启用高精度校准之前，必须完成 HRCF[14:0] / HRCR[9:0] / HRCRV 的配置。
启用高精度校准所需寄存器值的计算公式如下：

$$PREDIV=(FACTOR_A+1)\times(FACTOR_S+1) \tag{22-5}$$

$$RTCCAL=HRC_MASK-(FACTOR_S+1)\times HRCP \tag{22-6}$$

$$HRCADD`=\begin{cases}-1, & HRCADD=0\\ 1, & HRCADD=1\end{cases} \tag{22-7}$$

$$HIGHCAL=\frac{HRCADD`\times HRCN}{HRCR`} \tag{22-8}$$

$$f_{hcal}=f_{ck_spre}\times\frac{PREDIV}{PREDIV+RTCCAL+HIGHCAL} \tag{22-9}$$

公式解释:

f_{hcal}：高精度校准后的 ck_spre 频率

f_{ck_spre}：校准前的 ck_spre 频率

HRCR`：电路中高频时钟与 RTCCLK 之间的最终有效频率比率

注意：RTC_HRCR 寄存器中 HRC_MASK[14:0]位的实际配置值不得超过同步分频系数 FACTOR_S。RTC_HRCR 寄存器中 HRCN[9:0]位的配置值应小于 HRCR`。如果配置值超过 HRCR`，则有效的 HRCN[9:0]值将等于 HRCR`-1，但 RTC_HRCR 寄存器中的 HRCN[9:0]位仍会读取原始配置值。

高精度校准验证

高精度校准后的 ck_spre 可以通过 PC13 引脚输出进行验证。验证方法如下：

启用高精度补偿并检查 RTC 状态寄存器 RTC_STAT 中的 HRCPF 位。一旦该标志位被设置，从 ck_spre 的第一个上升沿开始测量 ck_spre 的周期长度和频率。

由于高精度校准中涉及的高频时钟可能存在误差，强烈建议通过软件计算高频校准时钟与 RTCCLK 时钟之间的频率比。

高精度校准的动态修改

高精度校准支持在校准过程中进行动态修改。当芯片的运行环境发生变化，例如运行温度或外部晶振波动时，软件可以在不禁用高精度校准的情况下修改校准寄存器。在修改过程中，必须先完成 RTC_HRCFC 寄存器的配置，然后再配置 RTC_HRCR 寄存器。如果当前动态修改仅涉及 RTC_HRCFC 寄存器，则 RTC_HRCR 寄存器也需要重新配置，并且配置值必须与之前的 RTC_HRCR 寄存器配置值相同。动态修改还需要两个 RTCCLK 周期才能生效。

注意：启用高精度校准时，PREDIV_A > 4 是必需的。

注意：高精度校准不能与平滑数字校准同时使用。

22.3.13. 时间戳功能

时间戳功能在 RTC_TS0 和 RTC_TS1 引脚上执行，由控制位 TSxRE / TSxFE 使能。它也可以通过控制位 ITSEN 使能。

当 RTC_TS0 或 RTC_TS1 管脚检测到时间戳事件(上升沿或下降沿)发生时(TSxRE = 1 / TSxFE = 1)，会将日历的值保存在时间戳寄存器中(RTC_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR)，同时时间戳标志(TSF0R / TSF0F / TSF1R / TSF1F)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_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR)，同时时间戳标志(TSF0R / TSF0F / TSF1R / TSF1F)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记录日历时间，而当时间戳标志位为 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。日历值保存在时间戳 0 上升沿寄存器中。

注意：因为同步机制的原因，当时间戳事件发生时，时间戳标志位会延迟 2 个 ck_apre 周期置位。

22.3.14. 侵入检测

RTC_TAMPx 管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置滤波功能的电平检测模式。

入侵检测的配置可以用于以下目的：

- 默认配置会擦除RTC备份寄存器和BKPSRAM寄存器。
- 可以从深度睡眠模式，深度睡眠模式1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[1:0]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[1:0]=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，时间戳标志位也将会被置位，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[1:0]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：侵入检测功能使能的情况下，即使 V_{DD} 电源被关掉该功能也依然可以运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当 FLT[1:0]位没有被设置成 0x0 时，侵入检测被设置成电平检测模式，FLT[1:0]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[1:0]位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

22.3.15. 校准时钟输出

如果 RTC_CTL 寄存器中的 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[6:0])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：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (12\text{-}10)$$

若 RTCCLK 为 32.768KHz，如果预分频器是默认值，那么 RTC_CALIB 对应的输出是 1Hz。

22.3.16. 闹钟输出

当 OS[1:0]控制位不为 0x00 时，RTC_ALARM 复用输出功能被启用。这个功能将直接输出 RTC_STAT 寄存器的闹钟标志 ALRMxF 或者自动唤醒标志 WTF。

RTC_CTL 寄存器中的 OPOL 位可以配置 ALRMxF 标志或者 WTF 标志输出时候的极性，因此 RTC_ALARM 的输出电平有可能与相应的位值相反。

22.3.17. RTC 引脚配置

RTC_OUT，RTC_TS0 和 RTC_TAMP0 都使用同一个 PC13 引脚。无论 PC13 的 GPIO 是什么配置，PC13 的功能由 RTC 控制。PC13 的 RTC 功能可以用于所以低功耗模式和 Vbat 模式。

PC13 的输出优先级如表*表 22-1. RTC 引脚配置*。

表 22-1. RTC 引脚配置和功能

功能配置和引脚功能	OS[1:0] (输出选择)	COEN (校准输出)	TP0EN (侵入检测0使能)	TS0REN / TS0FEN	ALRMOUTTYPE(闹钟输出类型)

				(时间戳0使能)	
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位来使能。
- RTC_TAMP0：侵入事件检测。
- RTC_TS0：时间戳0事件检测。

RTC_CFG 中的 ALRMOUTTYPE 位用于选择 RTC_ALARM 输出是推挽模式还是开漏模式。

RTC_CFG 中的 OUT2EN 位可以选择 RTC_OUT 输出到 PB2 或 PC13 引脚上。该输出在 VBAT / Standby 模式下不可用。

22.3.18. RTC 省电模式管理

表 22-2. 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC中断：闹钟0 / 闹钟1 / 侵入事件0 / 侵入事件1 / 时间戳事件 / 唤醒
深度睡眠模式 / 深度睡眠模式1	是：当时钟源是LXTAL或IRC40K时可以工作	RTC中断：闹钟0 / 闹钟1 / 侵入事件0 / 侵入事件1 / 时间戳事件 / 唤醒
待机模式	是：当时钟源是LXTAL或IRC40K时可以工作	RTC中断：闹钟0 / 闹钟1 / 侵入事件0 / 侵入事件1 / 时间戳事件 / 唤醒

注意：在 VBAT 供电模式下，PD4 引脚的时间戳事件无法唤醒待机模式。

22.3.19. RTC 中断

所有的 RTC 中断都被连接到 EXTI 控制器。

如果想使用 RTC 闹钟/侵入事件/时间戳中断/自动唤醒中断，应按下面步骤操作：

1. 设置并使能对应的 EXTI 中连接到 RTC 闹钟 / 侵入事件 / 时间戳 / 自动唤醒的中断线，

然后配置该线为上升沿触发模式；

2. 配置并使能 RTC 闹钟 / 侵入事件/ 时间戳 / 自动唤醒的全局中断；
3. 配置并使能 RTC 闹钟 / 侵入事件 / 时间戳功能。

表 22-3. RTC 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式和待机模式
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	TSF0R / TSF0F / TSF1R / TSF1F	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 IRC40K 时有效。

22.4. RTC 寄存器

RTC 基地址：0x4000 2800

22.4.1. 时间寄存器(RTC_TIME)

地址偏移：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。

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]			
									rw	rw		rw			

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]			
	rw			rw					rw			rw			

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.2. 日期寄存器(RTC_DATE)

地址偏移：0x04

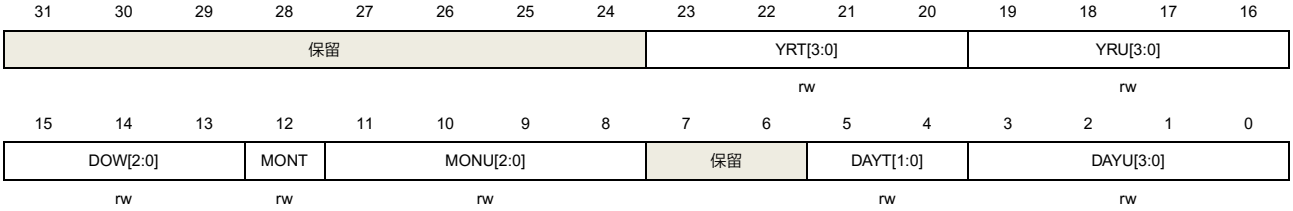
系统复位值：当 BPSHAD = 0，0x0000 2101

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。

注意：非闰年的 2 月不要配置为 29 号。

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	YRT[3:0]	年份十位值，以 BCD 码形式存储
19:16	YRU[3:0]	年份个位值，以 BCD 码形式存储
15:13	DOW[2:0]	星期 0x0：保留 0x1：星期一 … 0x7：星期日
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[2:0]	月份个位值，以 BCD 码形式存储
7:6	保留	必须保持复位值。
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

22.4.3. 控制寄存器(RTC_CTL)

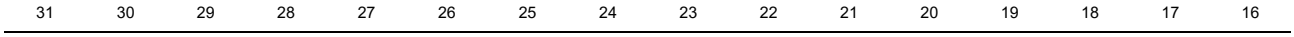
地址偏移：0x08

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。



保留		VCSCAL[1:0]		保留	TS1FEN	TS1REN	ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
		rw			rw	rw	rw	rw	rw		rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WTIE	ALRM1IE	ALRM0IE	TS0FEN	WTEN	ALRM1EN	ALRM0EN	保留	CS	BPSHAD	REFEN	TS0REN	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	VCSCAL[1:0]	BLDO 电压等级选择 00: 0.9V 01: 1.0V 10: 1.1V 11: 1.2V
27	保留	必须保持复位值。
26	TS1FEN	RTC_TS1 时间戳事件下降沿使能 0：关闭 RTC_TS1 时间戳事件下降沿 1：使能 RTC_TS1 时间戳事件下降沿
25	TS1REN	RTC_TS1 时间戳事件上升沿使能 0：关闭 RTC_TS1 时间戳事件上升沿 1：使能 RTC_TS1 时间戳事件上升沿
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	TS0FEN	RTC_TS0 时间戳事件下降沿使能 0：关闭 RTC_TS0 时间戳事件下降沿 1：使能 RTC_TS0 时间戳事件下降沿
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	TS0REN	RTC_TS0 时间戳事件上升沿使能 0：关闭 RTC_TS0 时间戳事件上升沿 1：使能 RTC_TS0 时间戳事件上升沿
2:0	WTCS[2:0]	自动唤醒定时器时钟选择 0x0：RTC 时钟的 16 分频 0x1：RTC 时钟的 8 分频 0x2：RTC 时钟的 4 分频 0x3：RTC 时钟的 2 分频 0x4，0x5：ck_spre (默认 1Hz)时钟 0x6，0x7：ck_spre (默认 1Hz)时钟并且将唤醒计数器值增加2^{16}

22.4.4. 状态寄存器(RTC_STAT)

地址偏移：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

除了 RTC_STAT[13:8]位之外，这个寄存器的其他位是写保护的。

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										HRCPF	TSF1F	TSF1R	TSF0F	ITSF	SCPF
										r	rc_w0	rc_w0	rc_w0	rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP1F	保留	TP0F	TSOVRF	TSF0R	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
31:22	保留	必须保持复位值。
21	HRCPF	高精度校准待定标志

		当软件未进入初始化模式写 RTC_HRCR 时，由硬件设置为 1， 高精度校准配置生效时，由硬件设置为 0。
20	TSF1F	时间戳 1 下降沿时间标志 当检测到时间戳 1 事件下降沿时由硬件设置。 软件写入清除 0。
19	TSF1R	时间戳 1 上升沿时间标志 当检测到时间戳 1 事件下降沿时由硬件设置。 软件写入清除 0。
18	TSF0F	时间戳 0 下降沿时间标志 当检测到时间戳 0 事件下降沿时由硬件设置。 软件写入清除 0。
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 TSF0R 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	侵入 1 事件标志 当在侵入 1 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	侵入 0 事件标志 当在时间戳 0 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 TSF0R, TSF0F, TSF1R, TSF1F 或 ITSF 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	TSF0R	时间戳 0 上升沿时间标志 当检测到时间戳 0 事件升沿沿时由硬件设置。 软件写入清除 0。
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	闹钟 1 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	闹钟 0 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	闹钟 1 配置可写标志 硬件置位和清零。ALRM1EN=0 时，标记闹钟是否可写。 0：不允许修改闹钟寄存器设置 1：允许修改闹钟寄存器设置
0	ALRM0WF	闹钟 0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记闹钟是否可写。 0：不允许修改闹钟寄存器设置 1：允许修改闹钟寄存器设置

22.4.5. 预分频寄存器(RTC_PSC)

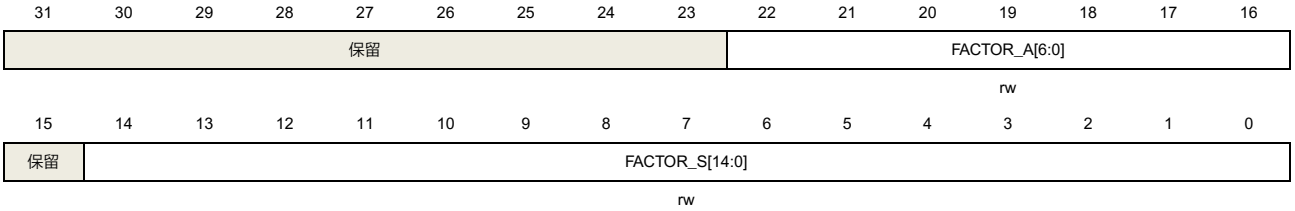
地址偏移：0x10

系统复位：无影响

备份域复位值：0x007F 00FF

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22:16	FACTOR_A[6:0]	异步预分频系数 ck_apre 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值。
14:0	FACTOR_S[14:0]	同步预分频系数 ck_spre 频率 = ck_apre 频率/(FACTOR_S+1)

22.4.6. 唤醒定时器寄存器(RTC_WUT)

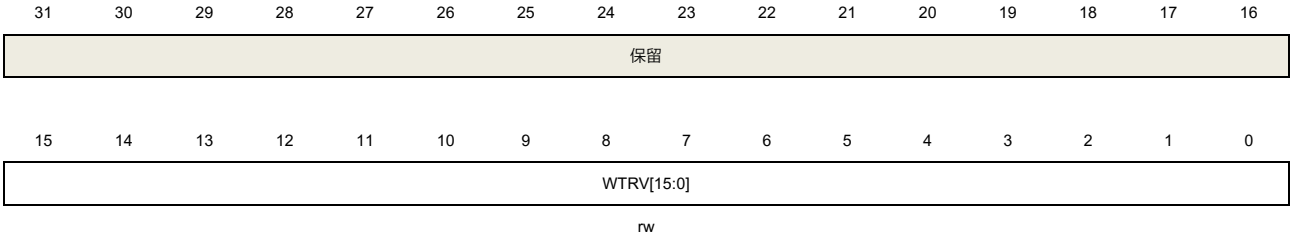
地址偏移：0x14

系统复位：无影响

备份域复位值：0x0000 FFFF

写保护寄存器

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTRV[15:0]	自动唤醒定时器重载值 当 WTEN 置 1 时，每隔 (WTRV[15:0]+1) 个 ck_wut 周期，WTF 置 1 一次。 ck_wut 通过 WTCS[2:0] 位选择

注意：禁止在 WTCS[2:0]=0b 011 时配置 WTRV=0x0000。

该寄存器仅在 WTWF=1 时才能写操作。

22.4.7. 闹钟 0 时间日期寄存器(RTC_ALRM0TD)

地址偏移：0x1C

系统复位：无影响

备份域复位值：0x0000 0000

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0]代表日期个位值 1：此时 DAYU[3:0]代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域

14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.8. 闹钟 1 时间日期寄存器(RTC_ALRM1TD)

地址偏移：0x20

系统复位：无影响

备份域复位值：0x0000 0000

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

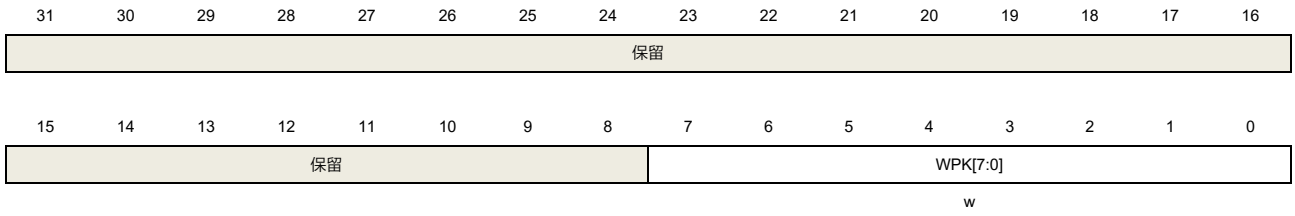
位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0] 代表日期个位值 1：此时 DAYU[3:0] 代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制

		1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.9. 写保护钥匙寄存器(RTC_WPK)

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	WPK[7:0]	写保护的解锁值

22.4.10. 亚秒寄存器(RTC_SS)

地址偏移：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。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值 该位值是同步预分频计数器的值。秒的小数部分由下面公式给出： 秒的小数部分 = (FACTOR_S - SSC)/(FACTOR_S + 1)

22.4.11. 移位控制寄存器(RTC_SHIFTCTL)

地址偏移：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。

注意：写入此寄存器会导致 RSYNF 位被清 0。

该寄存器只能按字(32 位)访问。

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

22.4.12. 时间戳 0 上升沿时间寄存器(RTC_TTS0R)

地址偏移：0x30

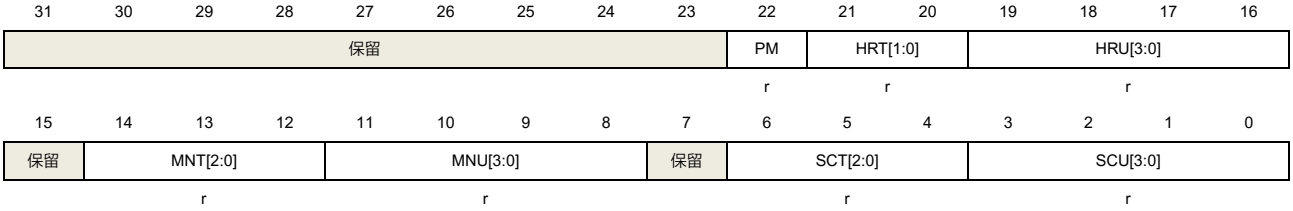
备份域复位值：0x0000 0000

系统复位：无影响

当 TSF0R 被置 1，该位用来记录日历时间。

清除 TSF0R 位也会清除此寄存器。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

22.4.13. 时间戳 0 上升沿日期寄存器(RTC_DTS0R)

地址偏移：0x34

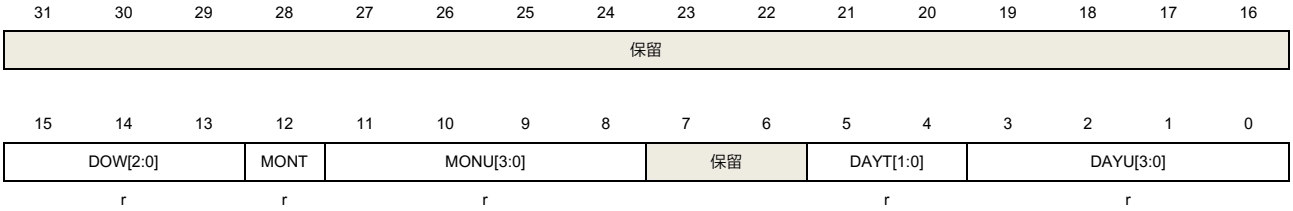
备份域复位值：0x0000 0000

系统复位：无影响

当 TSF0R 被置 1，该位用来记录日历日期。

清除 TSF0R 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。
6:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

22.4.14. 时间戳 0 上升沿亚秒寄存器(RTC_SSTS0R)

地址偏移：0x38

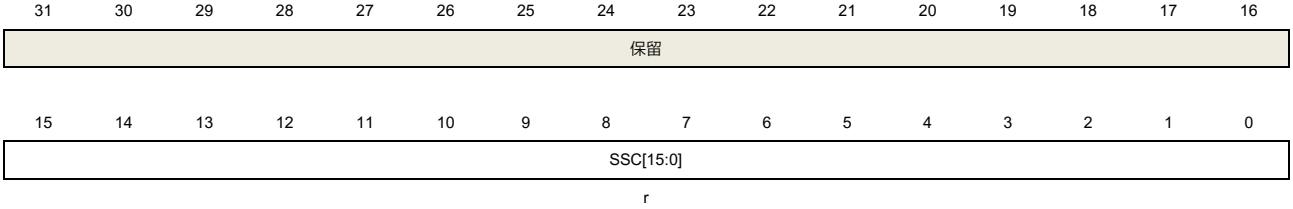
备份域复位：0x0000 0000

系统复位：无影响

当 TSF0R 被置 1，该位用来记录时间戳亚秒。

清除 TSF0R 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 TSF0R 置 1 时记录当时的同步预分频计数器的值。

22.4.15. 高精度频率补偿寄存器(RTC_HRFC)

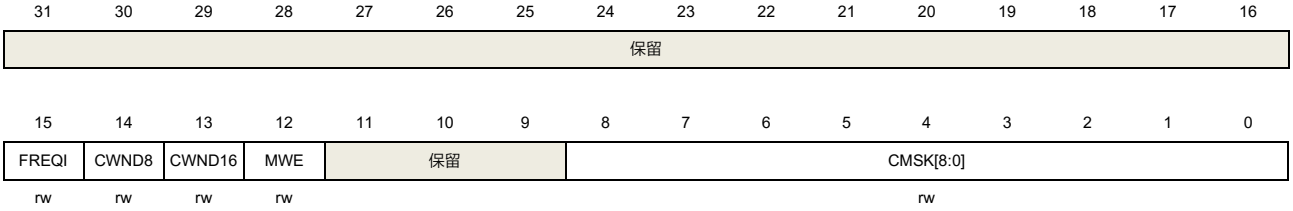
地址偏移：0x3C

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	MWE	多窗口校准使能: 0：单窗口模式 1：多窗口模式 在多窗口模式下，当前校准窗口结束后，校准将再次启动。 在连续平滑校准过程中，关闭平滑标准需要将 MWE 写为 0。当向 MWE 写入 0

		时，RTC_HRFC 寄存器中的其他位字段必须保持不变。
11:9	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 的精度来降低日历频率

22.4.16. 侵入寄存器(RTC_TAMP)

地址偏移：0x40

备份域复位：0x0000 0000

系统复位：无影响

注意：强烈建议在改变侵入检测配置之前，应该复位 TPxEN 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
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 TS0REN=0，TSF0R 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

22.4.17. 闹钟 0 亚秒寄存器(RTC_ALRM0SS)

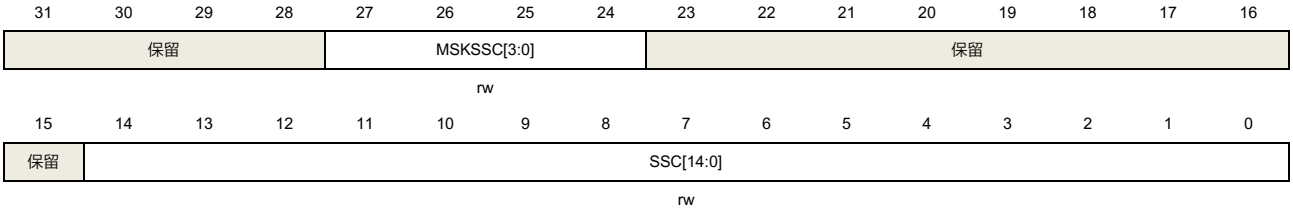
地址偏移：0x44

备份域复位：0x0000 0000

系统复位：无影响

写保护寄存器，仅当 ALRM0EN=0 或 INITM=1，可以进行写操作。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1:0]位用于时间匹配，其他位被忽略。 0x3：SSC[2:0]位用于时间匹配，其他位被忽略。 0x4：SSC[3:0]位用于时间匹配，其他位被忽略。 0x5：SSC[4:0]位用于时间匹配，其他位被忽略。 0x6：SSC[5:0]位用于时间匹配，其他位被忽略。 0x7：SSC[6:0]位用于时间匹配，其他位被忽略。 0x8：SSC[7:0]位用于时间匹配，其他位被忽略。 0x9：SSC[8:0]位用于时间匹配，其他位被忽略。 0xA：SSC[9:0]位用于时间匹配，其他位被忽略。 0xB：SSC[10:0]位用于时间匹配，其他位被忽略。 0xC：SSC[11:0]位用于时间匹配，其他位被忽略。 0xD：SSC[12:0]位用于时间匹配，其他位被忽略。 0xE：SSC[13:0]位用于时间匹配，其他位被忽略。 0xF：SSC[14:0]位用于时间匹配，其他位被忽略。 **注意**：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。
23:15	保留	必须保持复位值。
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

22.4.18. 闹钟 1 亚秒寄存器(RTC_ALRM1SS)

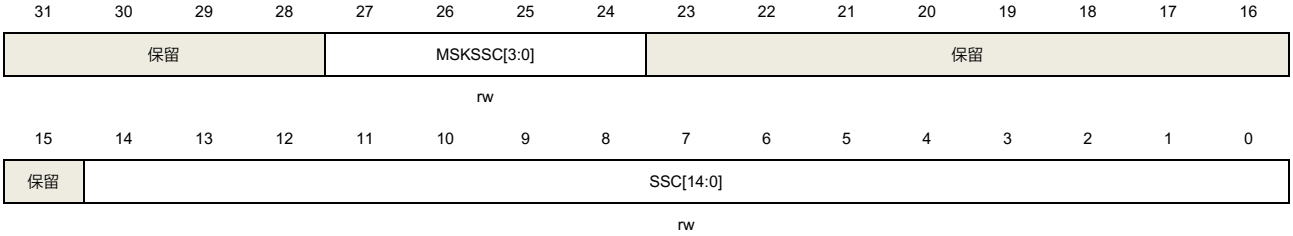
地址偏移：0x48

备份域复位：0x0000 0000

系统复位：无影响

写保护寄存器，仅当 ALRM1EN=0 或 INITM=1，可以进行写操作。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1:0]位用于时间匹配，其他位被忽略。 0x3：SSC[2:0]位用于时间匹配，其他位被忽略。 0x4：SSC[3:0]位用于时间匹配，其他位被忽略。 0x5：SSC[4:0]位用于时间匹配，其他位被忽略。 0x6：SSC[5:0]位用于时间匹配，其他位被忽略。 0x7：SSC[6:0]位用于时间匹配，其他位被忽略。 0x8：SSC[7:0]位用于时间匹配，其他位被忽略。 0x9：SSC[8:0]位用于时间匹配，其他位被忽略。 0xA：SSC[9:0]位用于时间匹配，其他位被忽略。 0xB：SSC[10:0]位用于时间匹配，其他位被忽略。 0xC：SSC[11:0]位用于时间匹配，其他位被忽略。 0xD：SSC[12:0]位用于时间匹配，其他位被忽略。 0xE：SSC[13:0]位用于时间匹配，其他位被忽略。 0xF：SSC[14:0]位用于时间匹配，其他位被忽略。 **注意**：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。
23:15	保留	必须保持复位值。
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

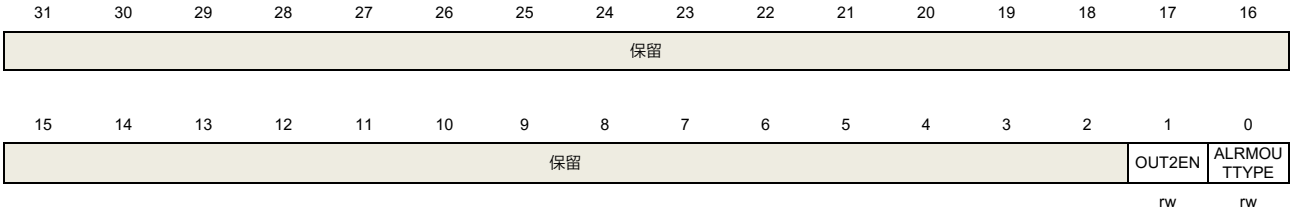
22.4.19. 配置寄存器(RTC_CFG)

地址偏移：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。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OUT2EN	RTC_OUT 引脚选择 0：RTC_OUT 输出到 PC13 1：RTC_OUT 输出到 PB2
0	ALRMOUTTYPE	RTC_ALARM 输出类型 0：开漏输出 1：推挽输出

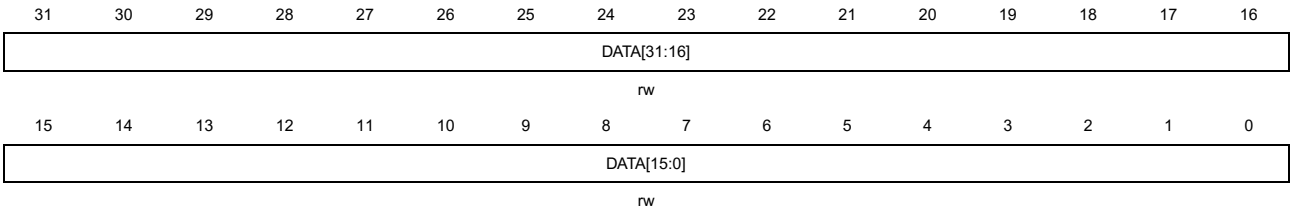
22.4.20. 备份寄存器(RTC_BKPx)(x=0..3)

地址偏移：0x50 到 0x5C

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。



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。

22.4.21. 时间戳 0 下降沿时间寄存器(RTC_TTS0F)

地址偏移：0xD0

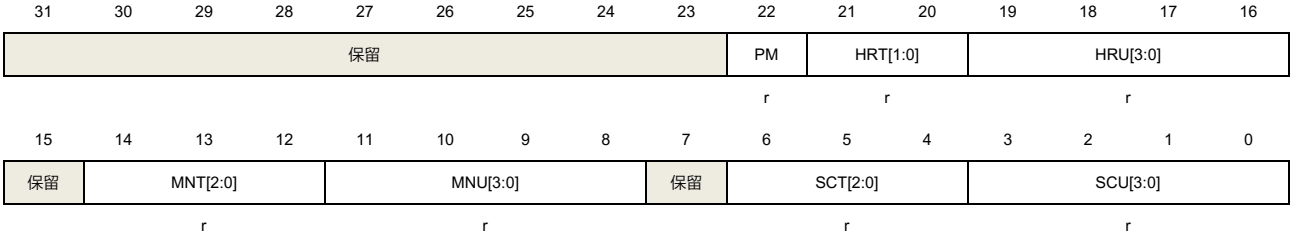
备份域复位：0x0000 0000

系统复位：无影响

当 TSF0F 设置为 1 时，这个寄存器将记录日历时间。

重置 TSF0F 位也将清除这个寄存器。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储。
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储。
15	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储。
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储。
7	保留	必须保持复位值
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储。
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储。

22.4.22. 时间戳 0 下降沿日期寄存器(RTC_DTS0F)

地址偏移：0xD4

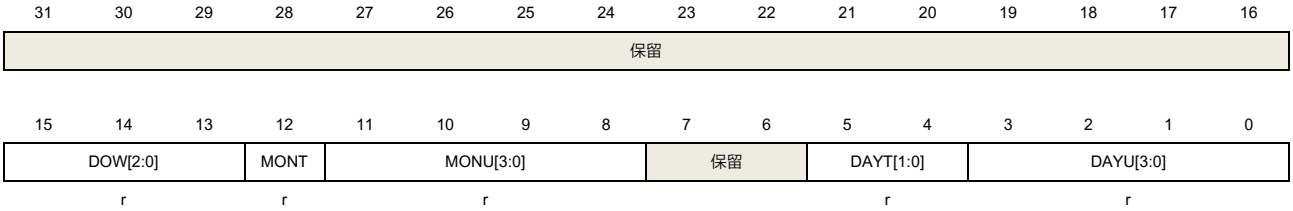
备份域复位：0x0000 0000

系统复位：无影响

当 TSF0F 设置为 1 时，这个寄存器将记录日历日期。

重置 TSF0F 位也将清除这个寄存器。

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。
7:6	保留	必须保持复位值
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储。
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储。

22.4.23. 时间戳 0 下降沿亚秒寄存器(RTC_SSTS0F)

地址偏移：0xD8

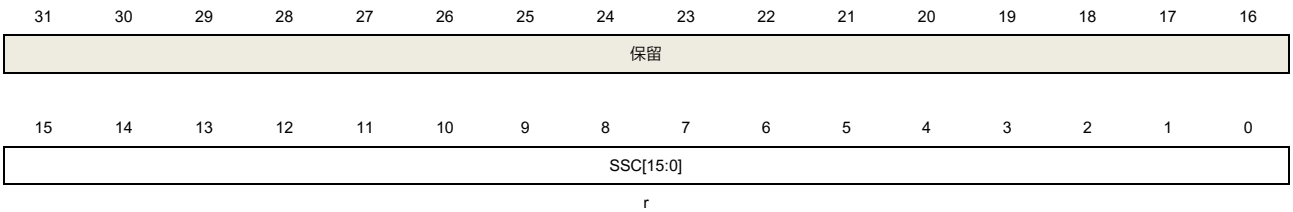
备份域复位：0x0000 0000

系统复位：无影响

当 TSF0F 设置为 1 时，这个寄存器将记录时间戳 0 的亚秒。

重置 TSF0F 位也将清除这个寄存器。

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

TSF0F 置 1 时记录当时的同步预分频计数器的值。

22.4.24. 时间戳 1 上升沿时间寄存器(RTC_TTS1R)

地址偏移：0xDC

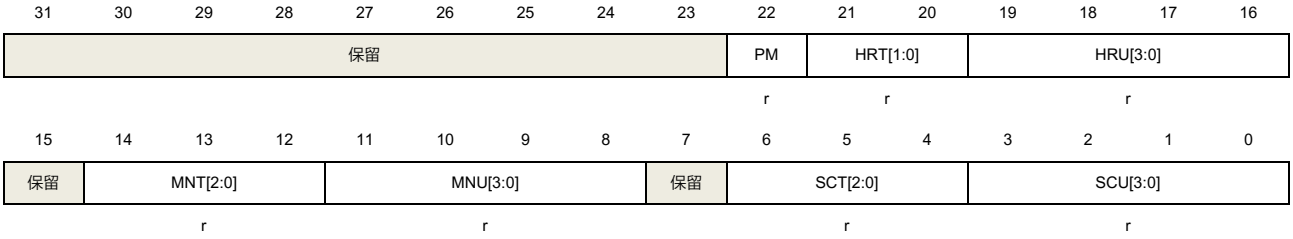
备份域复位：0x0000 0000

系统复位：无影响

当 TSF1R 设置为 1 时，这个寄存器将记录日历时间。

重置 TSF1R 位也会清除这个寄存器。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储。
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储。
7	保留	必须保持复位值
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储。
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储。

22.4.25. 时间戳 1 上升沿日期寄存器(RTC_DTS1R)

地址偏移：0xE0

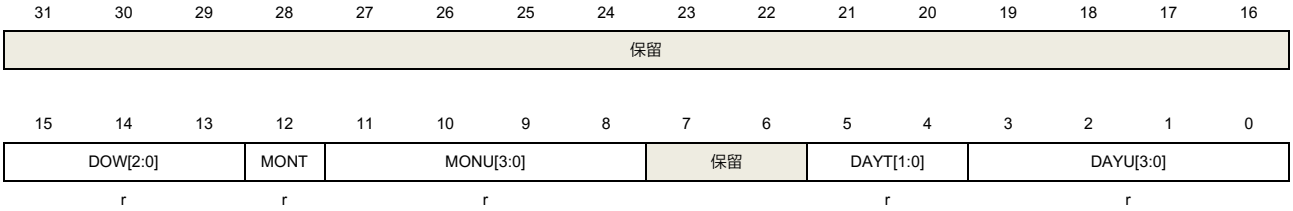
备份域复位：0x0000 0000

系统复位：无影响

当 TSF1R 设置为 1 时，这个寄存器将记录日历日期。

重置 TSF1R 位也会清除这个寄存器。

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。
7:6	保留	必须保持复位值
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储。
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储。

22.4.26. 时间戳 1 上升沿亚秒寄存器(RTC_SSTS1R)

地址偏移：0xE4

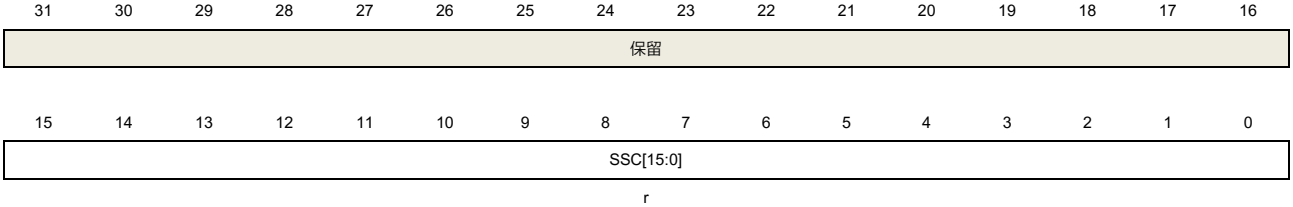
备份域复位：0x0000 0000

系统复位：无影响

当 TSF1R 设置为 1 时，这个寄存器将记录时间戳 1 的亚秒。

重置 TSF1R 位也会清除这个寄存器。

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 TSF1R 置 1 时记录当时的同步预分频计数器的值。

22.4.27. 时间戳 1 下降沿时间寄存器(RTC_TTS1F)

地址偏移：0xE8

备份域复位：0x0000 0000

系统复位：无影响

当 TSF1F 设置为 1 时，这个寄存器将记录日历时间。

重置 TSF1F 位也会清除这个寄存器。

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。

22.4.28. 时间戳 1 下降沿日期寄存器(RTC_DTS1F)

地址偏移：0xEC

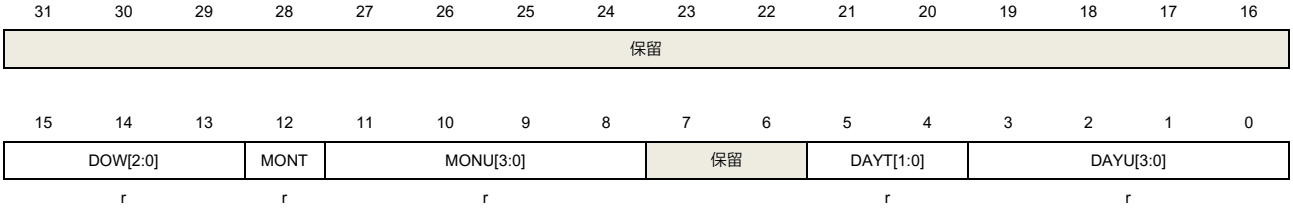
备份域复位：0x0000 0000

系统复位：无影响

当 TSF1F 设置为 1 时，这个寄存器将记录日历日期。

重置 TSF1F 位也会清除这个寄存器。

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。
7:6	保留	必须保持复位值
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储。
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储。

22.4.29. 时间戳 1 下降沿亚秒寄存器(RTC_SSTS1F)

地址偏移：0xF0

备份域复位：0x0000 0000

系统复位：无影响

当 TSF1F 设置为 1 时，这个寄存器将记录时间戳 1 的亚秒。

重置 TSF1F 位也会清除这个寄存器。

该寄存器只能按字(32 位)访问。



SSC[15:0]
r

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 TSF1F 置 1 时记录当时的同步预分频计数器的值。

22.4.30. 高精度时钟频率校准寄存器(RTC_HRCFC)

地址偏移：0xF4

备份域复位：0x0000 3E80

系统复位：无影响

写保护寄存器。

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	HRCRV	HRCR[9:0)有效 0：由硬件根据公式 HRCF[14:0]/32768 计算比值值。 1：通过软件配置
27:26	保留	必须保持复位值。
25:16	HRCR[9:0]	高分辨率校准时钟频率与 RTC 时钟频率之比。 如果设置了 HRCRV 位，则 HRCR[9:0]有效。
15	保留	必须保持复位值。
14:0	HRCF[14:0]	高分辨率校准时钟频率。单位是 KHZ。 在配置 RTC_HRFC 寄存器之前，必须将 HRCF[14:0]写入正确的值。

22.4.31. 高精度校准寄存器(RTC_HRCR)

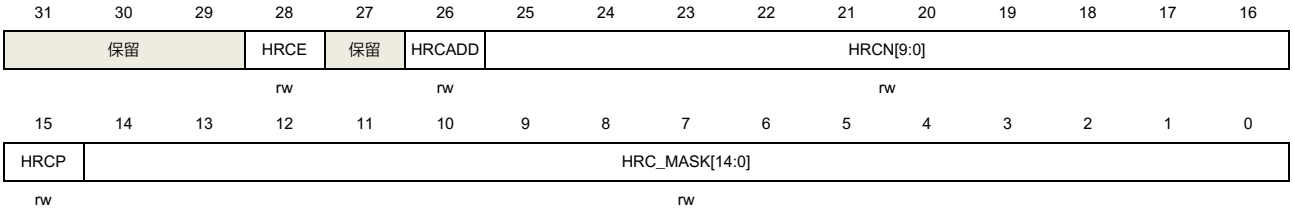
地址偏移：0xF8

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
31:29	保留	必须保持复位值。
28	HRCE	高分辨率校准功能 0：关闭高分辨率校准 1：开启高分辨率校准
27	保留	必须保持复位值。
26	HRCADD	RTC 预分频时钟 ck_spre 的 HRCN[9:0]高频时钟周期调整方向。 0：每个 ck_spre 减去 HRCN[9:0]高频时钟周期 1：每个 ck_spre 增加 HRCN[9:0]高频时钟周期
25:16	HRCN[9:0]	用于高分辨率校准调整的高频时钟周期数
15	HRCP	预分频时钟 ck_spre 每亚秒减少一个 RTC 时钟周期。
14:0	HRC_MASK[14:0]	预分频时钟 ck_spre 增加 HRC_MASK[14:0] RTC 时钟周期。 HRC_MASK[14:0]在 RTC_PSC 寄存器中不应该写入大于 FACTOR_S[14:0]位的值。

23. 定时器(TIMER)

表 23-1. 定时器(TIMERx)分为 4 种类型

定时器		定时器 0/7/19	定时器 1/2/3/4	定时器 9/10/12/13	定时器 15/40/41/42/43/44	定时器 5/6/50/51
类型		高级	通用 L0	通用 L2	通用 L3	基本
预分频器		16 位	16 位	16 位	16 位	16 位
计数器		16 位	16 位 (TIMER2/3/4) 32 位 (TIMER1)	16 位	16 位	16 位
计数模式		向上，向下，中央对齐	向上，向下，中央对齐	向上	向上，向下，中央对齐	只有向上
可重复性		●	×	×	●	×
捕获/比较通道数		4	4	1	3	0
复合 PWM 模式		●	×	×	●	×
非对称 PWM 模式		●	×	×	×	×
输出匹配脉冲选择		●	×	×	●	×
互补和死区时间		●	×	×	●	×
中止功能	BREAK	●	×	×	●	×
	通道中止	●	×	×	×	×
单脉冲		●	●	●	●	●
正交译码器		●	●	×	×	×
译码器		×	●	×	×	×
主-从管理		●	●	●	●	×
内部连接		● (1)	● (2)	●	● (3)	TRGO TO DAC
同步和初始值、计数方向刷新		●	×	×	●	×
DMA		●	●	×	●	● (4)
Debug 模式		●	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI14
(1)	TIMER0	TIMER4_TRGO	TIMER1_TRGO	TIMER2_TRGO	TRIGSEL	
	TIMER7	TIMER0_TRGO	TIMER1_TRGO	TIMER3_TRGO		
	TIMER19	TIMER0_TRGO	TIMER1_TRGO	TIMER4_TRGO		
(2)	TIMER1	TIMER0_TRGO	TIMER7_TRGO	TIMER2_TRGO		
	TIMER2	TIMER0_TRGO	TIMER1_TRGO	TIMER4_TRGO		
	TIMER3	TIMER0_TRGO	TIMER1_TRGO	TIMER2_TRGO		
	TIMER4	TIMER1_TRGO	TIMER2_TRGO	TIMER3_TRGO		
(3)	TIMER9	TRIGSEL				TRIGSEL
	TIMER10	TRIGSEL				
	TIMER12	TRIGSEL				
	TIMER13	TRIGSEL				
(4)	TIMER14	TIMER0_TRGO	TIMER2_TRGO	TIMER12_TRGO	TIMER40_CH0	
	TIMER40	TIMER0_TRGO	TIMER2_TRGO	TIMER13_TRGO	TIMER14_CH0	
	TIMER41	TIMER0_TRGO	TIMER2_TRGO	TIMER9_TRGO	TIMER13_TRGO	
	TIMER42	TIMER0_TRGO	TIMER2_TRGO	TIMER10_TRGO	TIMER13_TRGO	
	TIMER43	TIMER0_TRGO	TIMER2_TRGO	TIMER10_TRGO	TIMER13_TRGO	
	TIMER44	TIMER0_TRGO	TIMER2_TRGO	TIMER12_TRGO	TIMER13_TRGO	
(5)	只有更新事件可以产生 DMA 请求。但是定时器 5/6 中没有 DMA 配置 DMAS(DMA 请求源选择)					

23.1. 高级定时器(TIMERx, x=0, 7, 19)

23.1.1. 简介

高级定时器(TIMER0/7/19)是四通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。高级定时器含有一个 16 位无符号计数器。

高级定时器是可编程的，可以用于计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

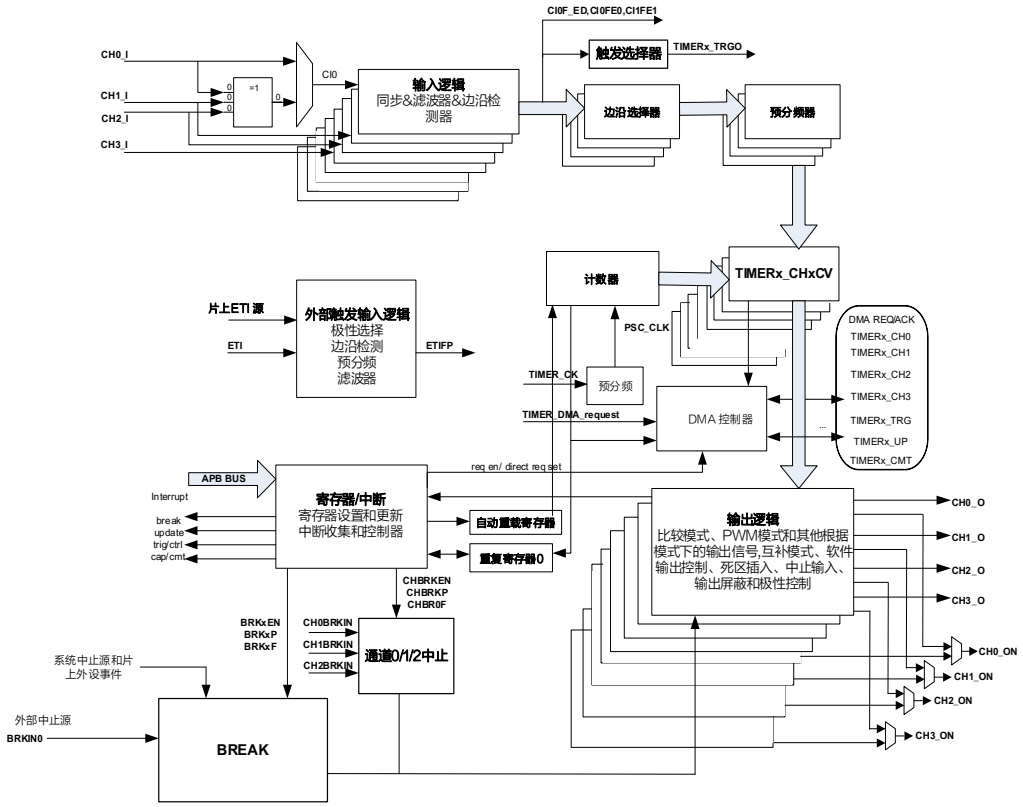
23.1.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：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：BREAK和通道0/1/2中止；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

23.1.3. 结构框图

*图 23-1. 高级定时器结构框图*提供了高级定时器的内部配置细节。

图 23-1. 高级定时器结构框图



23.1.4. 功能描述

时钟源选择

高级定时器可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x=0,7,19)寄存器中的 TSCFGy[4:0] (y=0...6)位域控制的复用时钟源驱动。

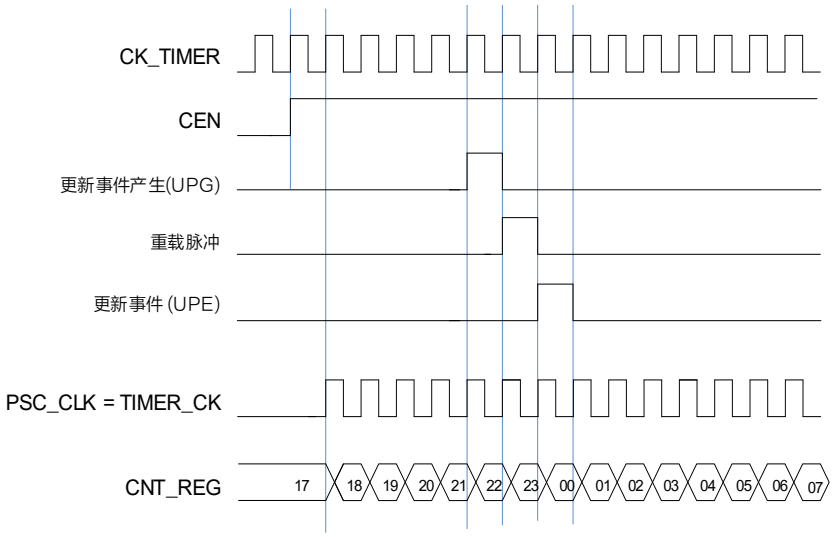
- 当SYSCFG_TIMERxCFG(x=0,7,19)寄存器中的TSCFGy[4:0]=5'b00000(y=0...6)时，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

如果 SYSCFG_TIMERxCFG(x=0,7)寄存器中的 TSCFGy[4:0] =5'b00000(y=0...6)，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器(预分频值由 TIMERx_PSC 寄存器确定)产生 PSC_CLK。

这种模式下，驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER。

如果 SYSCFG_TIMERxCFG(x=0,7,19)寄存器中的 TSCFGy[4:0] (y=0..2,6)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)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-2. 内部时钟分频为 1 时正常模式下的控制电路



■ TSCFG6[4:0] !=5'b00000(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在 TIMERx_CHn(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 0x9/0xA 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1~0x4 来选择。

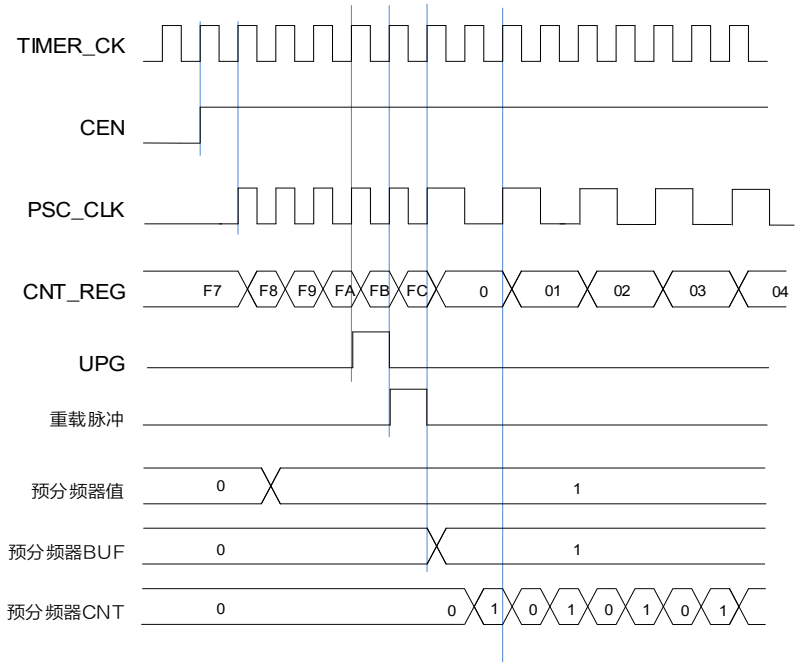
■ SMC1=1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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。

时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 23-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数。如果设置了重复计数器，在(TIMERx_CREP0+1)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

*图 23-4. 向上计数时序图,PSC=0/2*和*图 23-5. 向上计数时序图,在运行时改变 TIMERx_CAR 寄存器的值*给出一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 23-4. 向上计数时序图，PSC=0/2

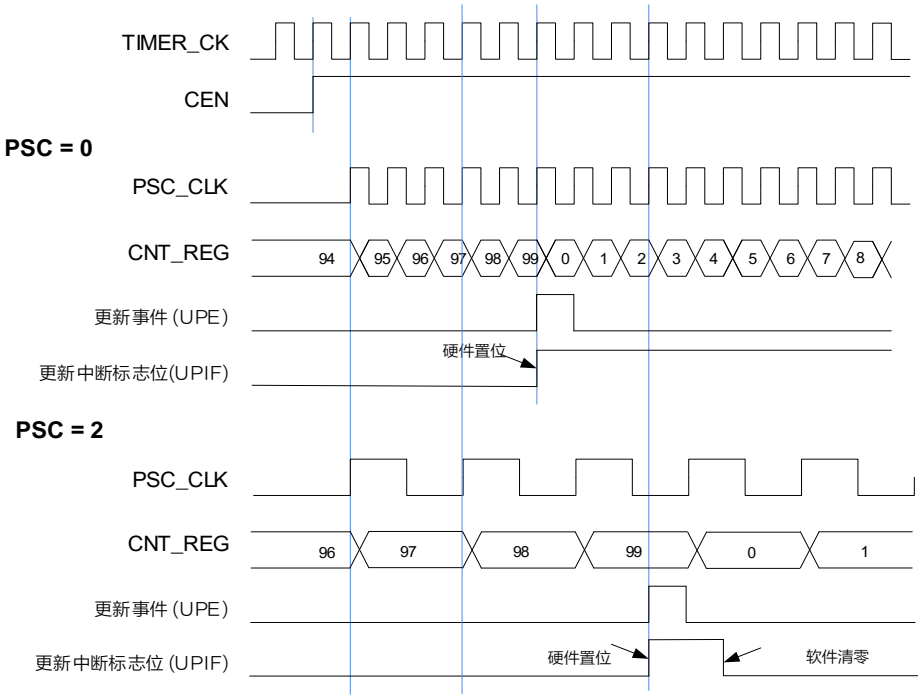
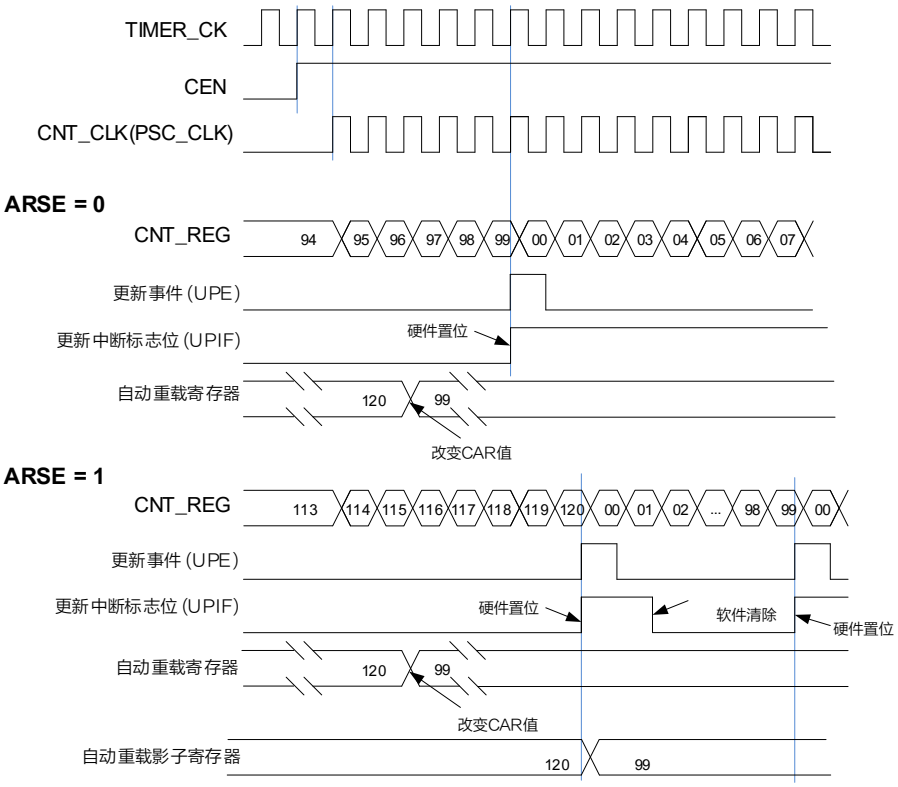


图 23-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值(定义在 TIMERx_CAR 寄存器中)向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数。如果设置了重复计数器，在(TIMERx_CREP0+1)次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

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

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

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

*图 23-6. 向下计数时序图，PSC=0/2*和*图 23-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 23-6. 向下计数时序图，PSC=0/2

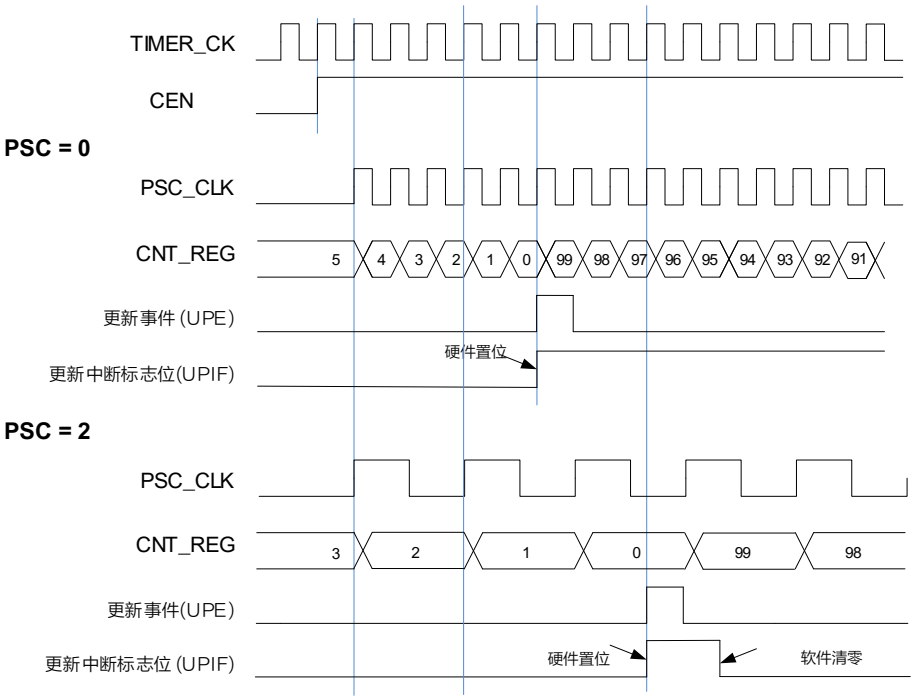
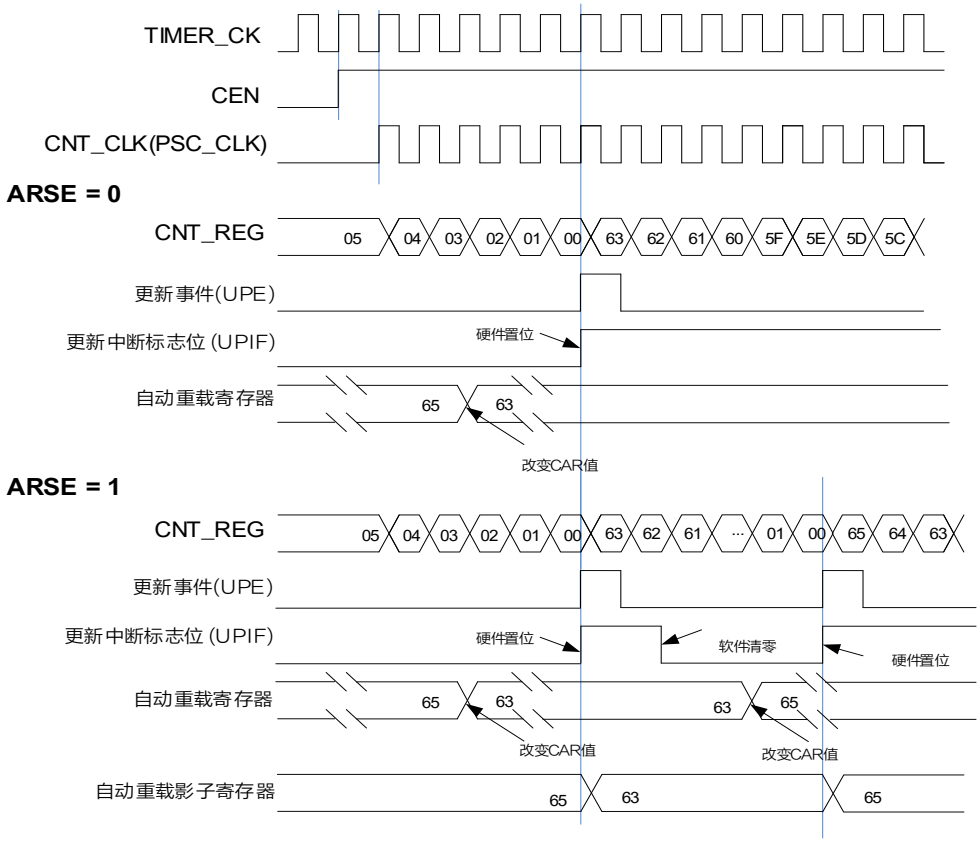


图 23-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到(TIMERx_CAR-1)产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。计数方向被硬件自动更新。

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

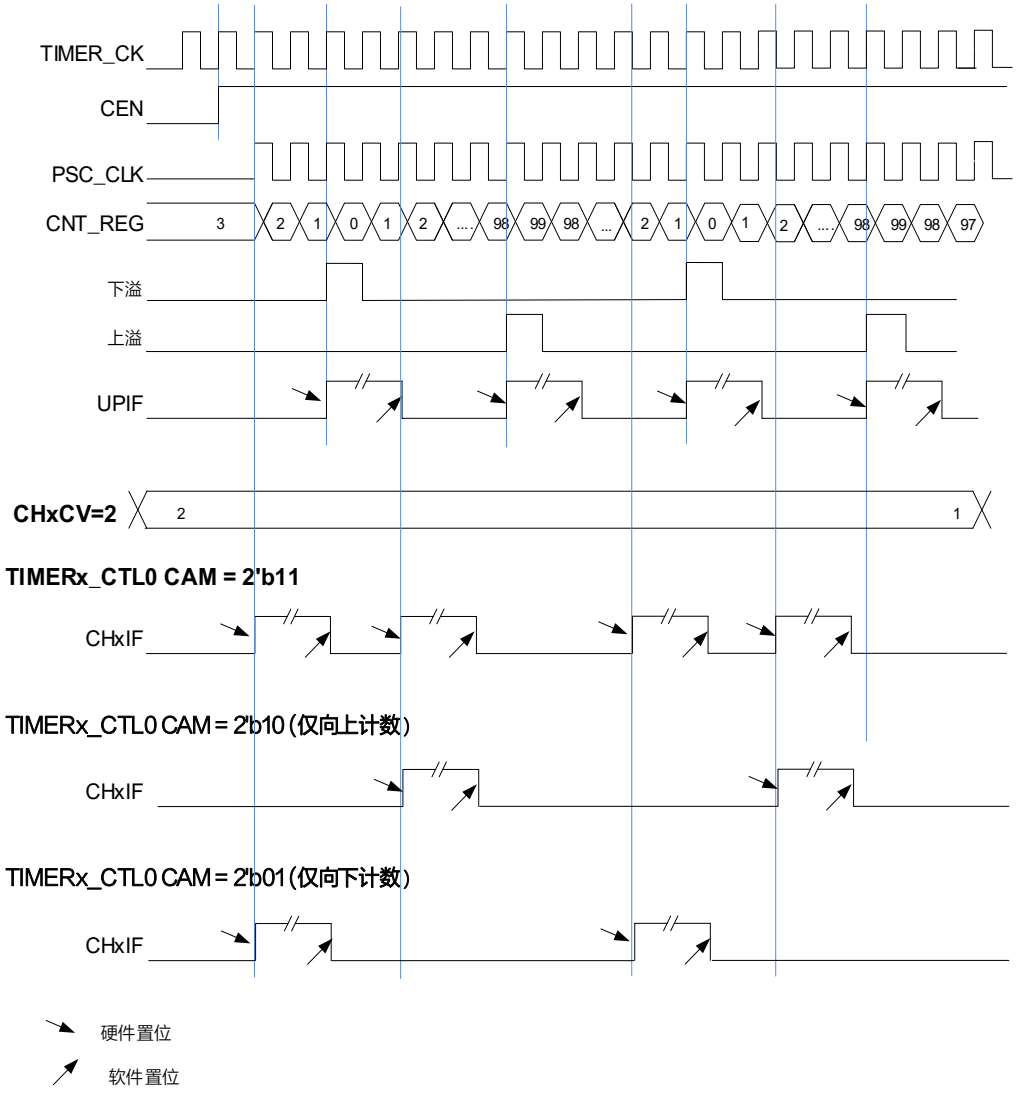
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 23-8. 中央计数模式计数器时序图*。

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

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

*图 23-8. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为。

图 23-8. 中央计数模式计数器时序图



重复计数器

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 寄存器的 CREP0 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 寄存器中 CREP0 的值并产生一个更新事件。

新写入的 CREP0 值将在下一次更新事件到来时生效。当 CREP0 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 值何时生效。如果在写入奇数到CREP0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 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事

件。

图 23-9. 中央计数模式下计数器重复时序图

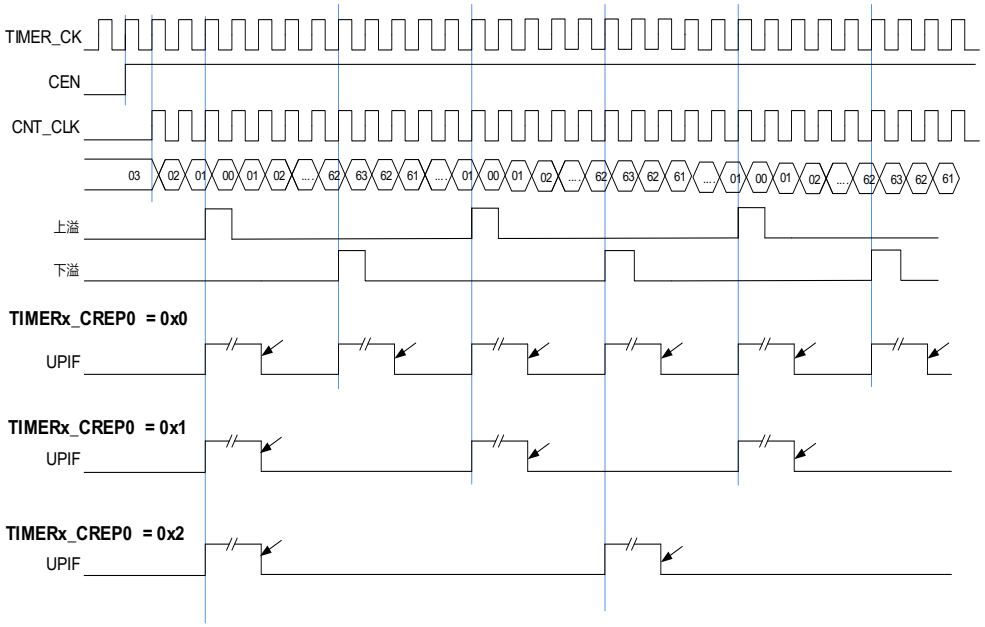


图 23-10. 在向上计数模式下计数器重复时序图

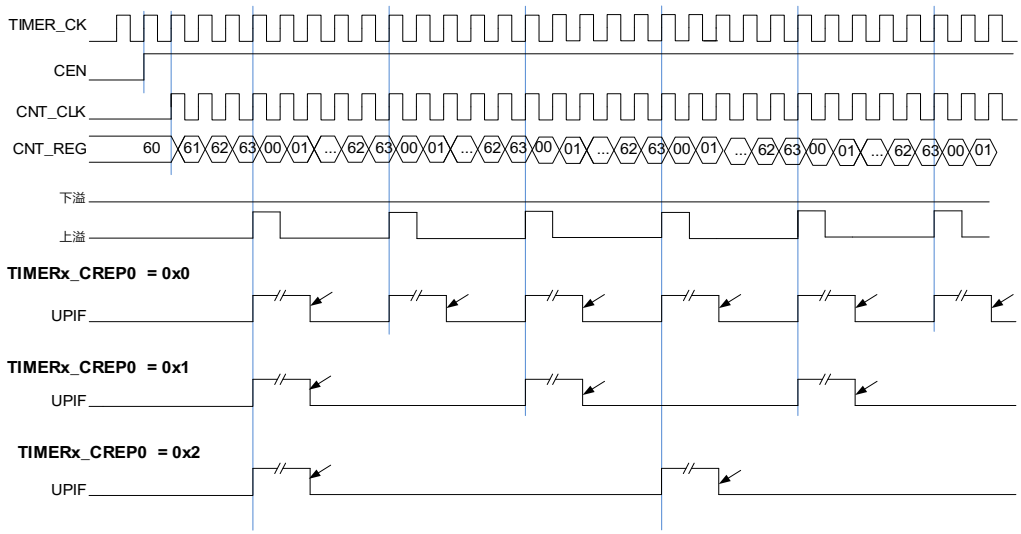
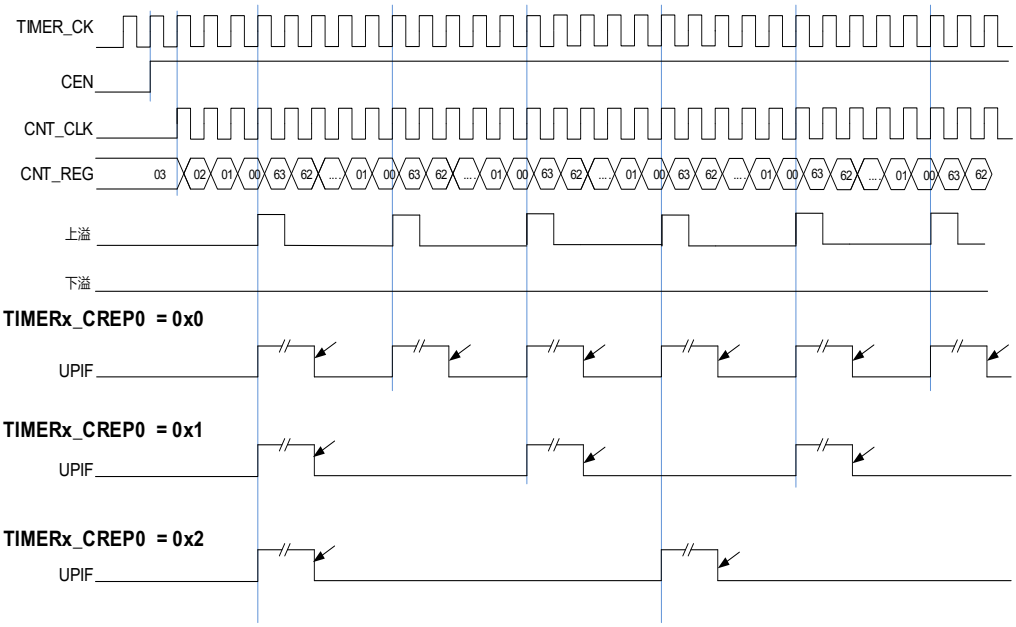


图 23-11. 在向下计数模式下计数器重复时序图



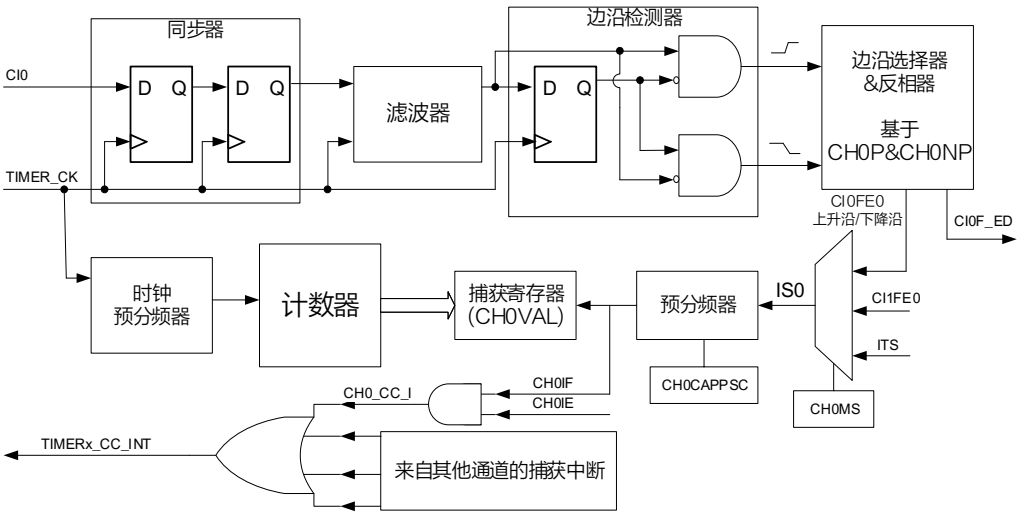
捕获/比较通道

高级定时器拥有 4 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(x=0..3)寄存器会捕获计数器当前的值，同时 CHxIF(x=0..3)位置 1，如果 CHxIE =1(x=0..3)，则产生相应的通道中断。

图 23-12. 通道 0 输入捕获逻辑



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/ CHxNP 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 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)：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT 位。

第二步：边沿选择(TIMERx_CHCTL2 寄存器中 CHxP 和 CHxNP 位)：

配置 CHxP 和 CHxNP 位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!=0x000)，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、CHxDEN 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)。

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 位置 1。如果 CHxIF 位已经为 1，则 CHxOF 位置 1。根据 TIMERx_DMAINTEN 寄存器中 CHxIE、CHxDEN 位的配置，相应的中断和 DMA 请求会被提出。

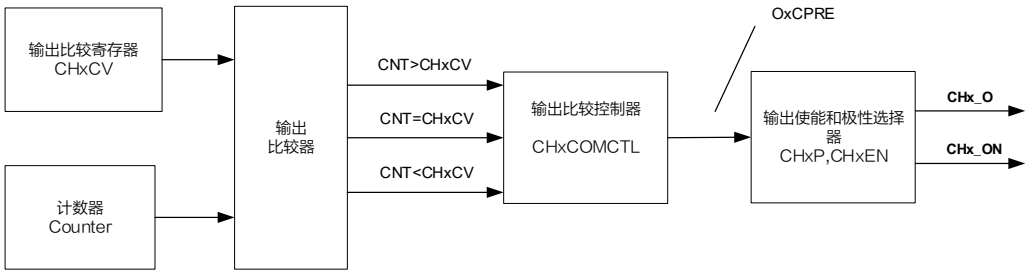
直接产生：软件设置 CHxG 位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 3'b001，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 3'b010，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 输出比较模式

*图 23-13. 输出比较逻辑*给出了通道的输出比较逻辑。

图 23-13. 输出比较逻辑



在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的TIMERx_CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者翻转。当计数器的值与 TIMERx_CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN =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/CHxNP位来选择有效电平的极性；
- 设置CHxEN/CHxNEN使能输出。

第三步：通过 CHxIE/ CHxDEN 位配置中断/DMA 请求使能。

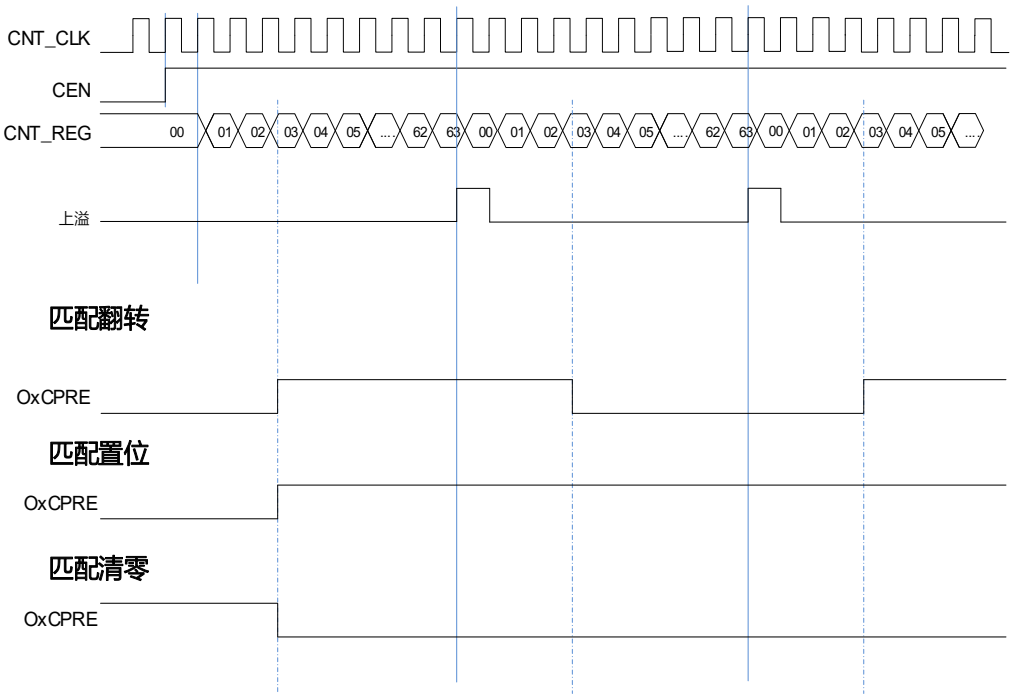
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

TIMERx_CHxCV 可以在运行时根据你所期望的波形而改变。

第五步：设置 CEN 位使能定时器。

*图 23-14. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 23-14. 三种输出比较模式



PWM 模式

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL 为 4'b0111),通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 23-15. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由(2*TIMERx_CAR 寄存器值)决定，占空比由(2*TIMERx_CHxCV 寄存器值)决定。*图 23-16. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下(CHxCOMCTL =4'b0110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平；PWM1 模式下(CHxCOMCTL =4'b0111)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为无效电平。

图 23-15. EAPWM 时序图

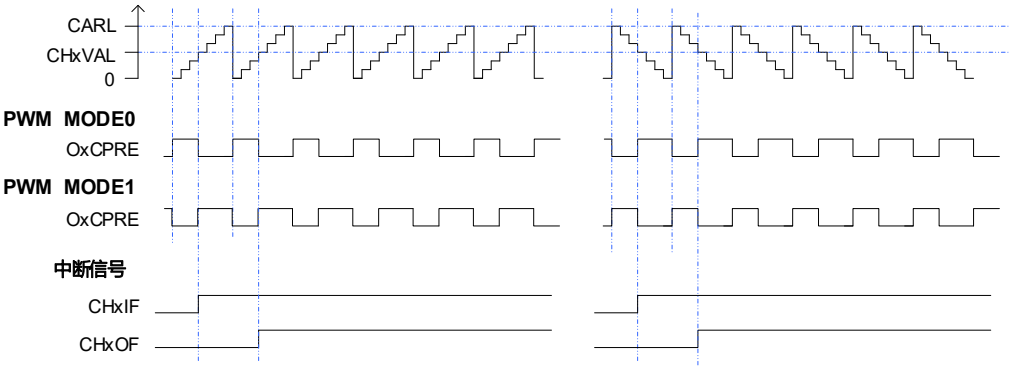
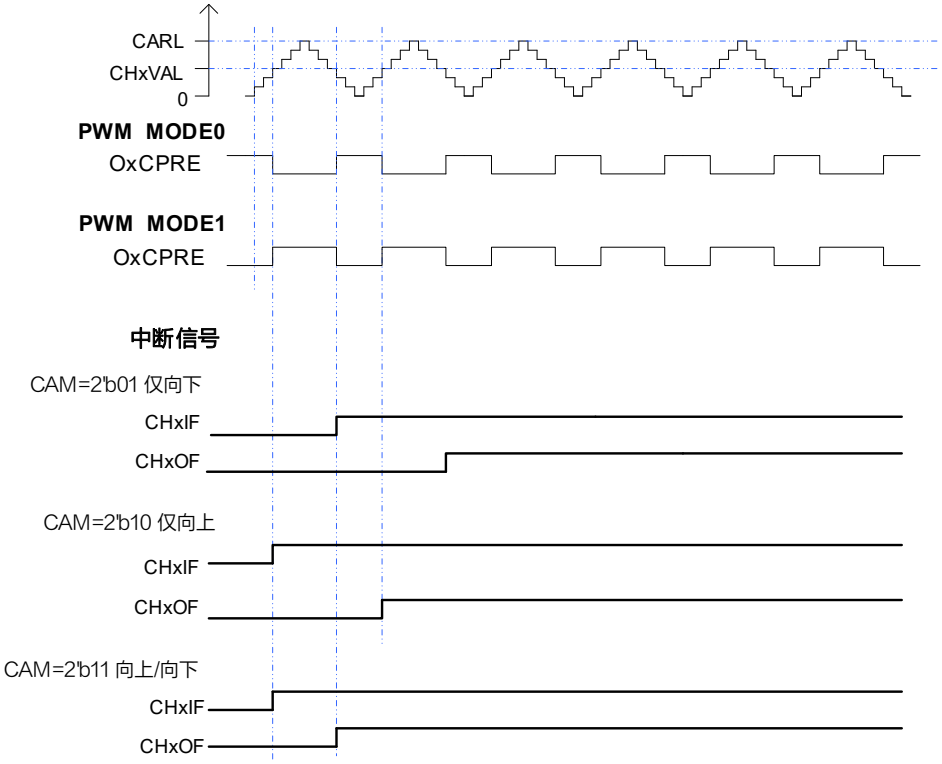


图 23-16. CAPWM 时序图



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。

当 CHxVAL 或 CHxCOMVAL_ADD = 0 / CARL，通道输出被进行了特殊处理，通过 TIMERx_CHCTL2 寄存器的 CHxPERFOREN 位置位，强制 OxCPRE 输出高电平或低电平(根据所选复合 PWM 模式确定)。

PWM 的周期取决于(CARL + 0x0001)，PWM 脉冲宽度可以下*表 23-2 复合 PWM 脉冲宽度*计算。

表 23-2 复合 PWM 脉冲宽度

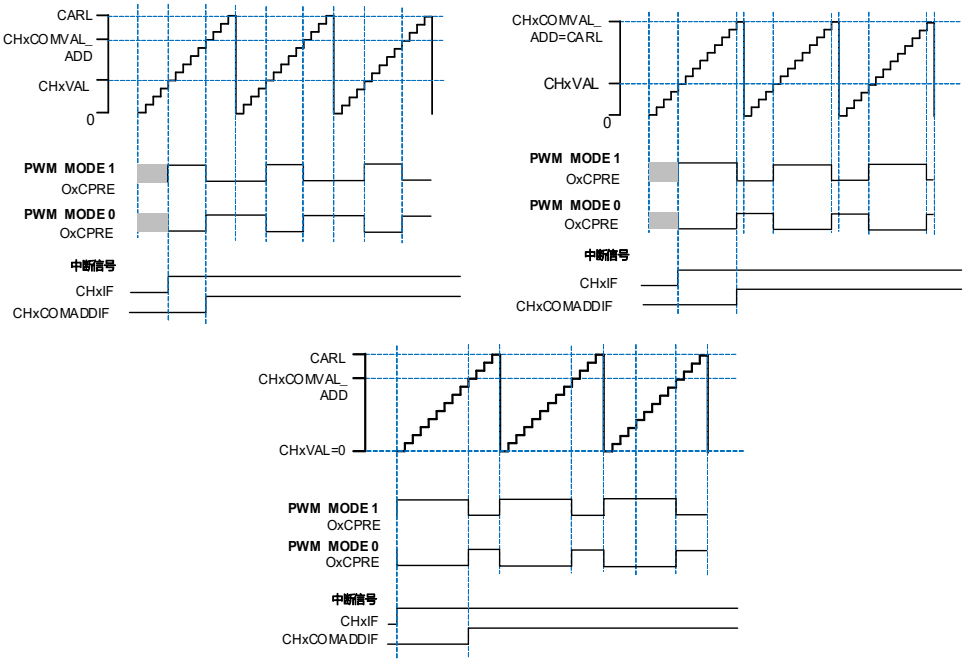
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD <= CARL	PWM 模式 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	PWM 模式 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL <= CARL	PWM 模式 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM 模式 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD <= CARL)或 (CHxVAL > CARL > 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：

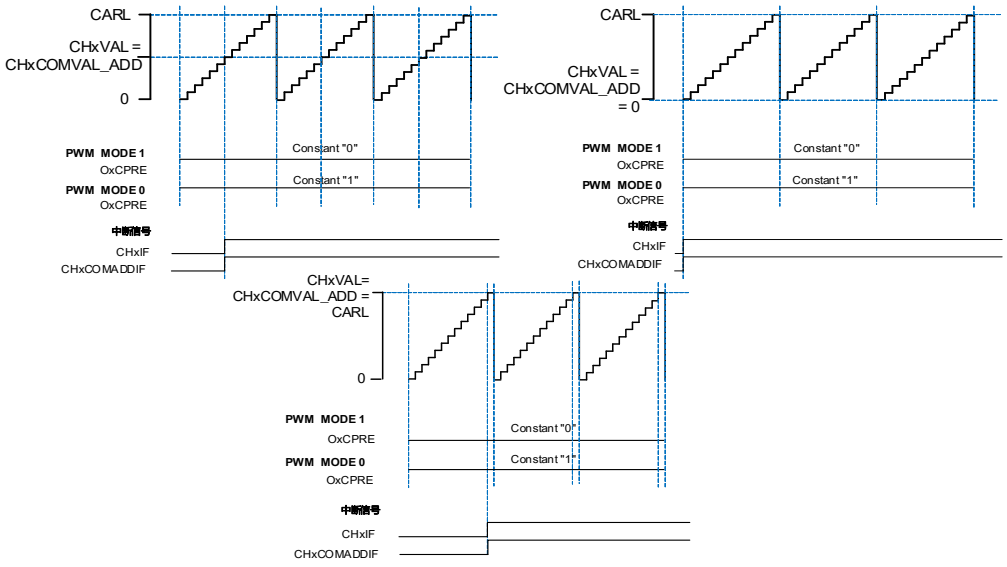
1） CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-17. 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD)



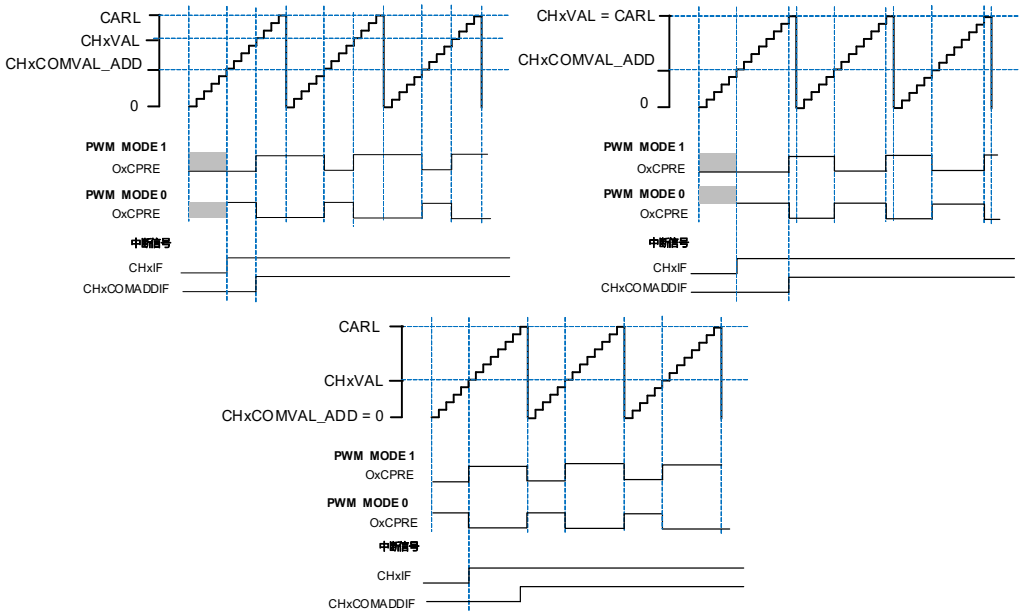
2） CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-18. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD)



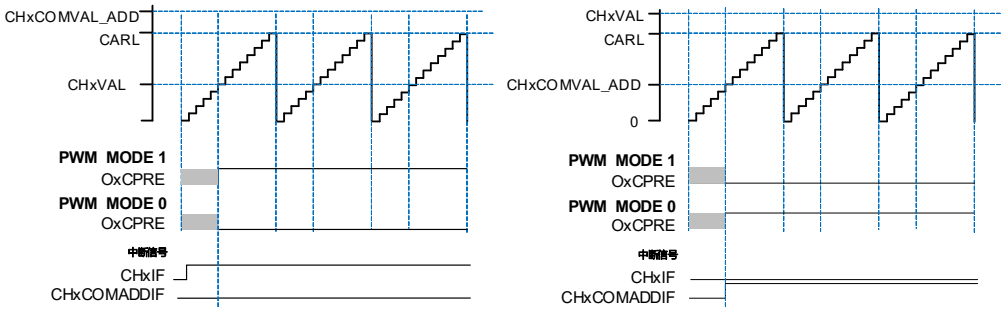
3） CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-19. 通道 x 输出 PWM(CHxVAL > CHxCOMVAL_ADD)



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

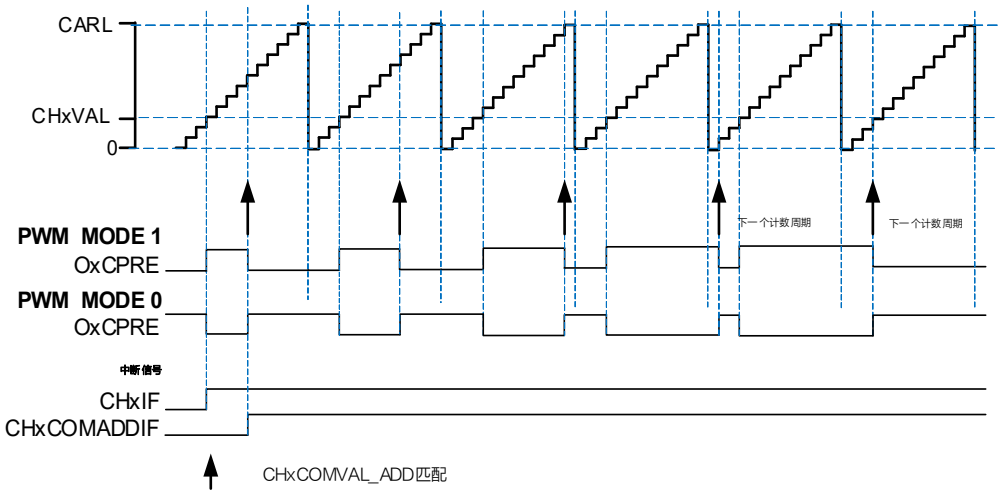
图 23-20. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL)



复合 PWM 模式支持不修改周期只修改占空比的 PWM 信号的生成。*图 23-21. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变*显示 PWM 输出和中断波形。

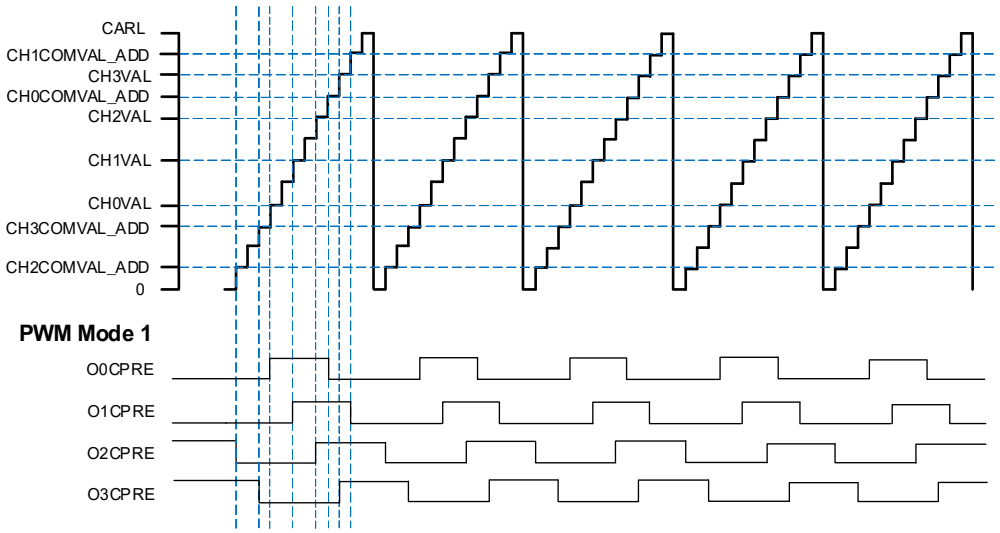
在某些情况下，CHxCOMVAL_ADD 的匹配事件可以发生在下一个计数周期(CHxCOMVAL_ADD 值在计数器到达 CHxVAL 值之后被写入，且 CHxCOMVAL_ADD 值小于或者等于 CHxVAL 值)。

图 23-21. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变



如果多个通道配置为复合 PWM 模式，可以为每对通道 x 的匹配边沿设定一个偏移量(相对于其它通道)。这种特性在产生照明 PWM 控制信号时非常有用，因为在这种情况下，希望彼此边缘不重合，以消除噪声的产生。CHxVAL 寄存器值是 PWM 脉冲相对于计数器周期开始的偏移。

图 23-22. 复合 PWM 模式下四通道输出



非对称 PWM 模式

非对称 PWM 波允许两个中央对齐的 PWM 波之间产生一个可编程的相位偏移。在非对称 PWM 波模式下(CHxCOMCTL[3:0]= 4'b1010/4'b1011 and CHxMS[2:0] = 3'b000)，在通道上输出的 PWM 波信号是由 CHxVAL 和 CHxCOMVAL_ADD 位共同控制的，非对称 PWM 波只在中央对齐模式下有效。

如果 CHxCOMCTL = 4'b1010(非对称 PWM 波模式 0)，向上计数模式(DIR = 1'b0)，当计数值达到 CHxVAL 的值时，输出强制为低电平。

如果 CHxCOMCTL = 4'b1010(非对称 PWM 波模式 0)，向下计数模式(DIR = 1'b1)，当计数值达到 CHxCOMVAL_ADD 的值时，输出强制为高电平。

如果 CHxCOMCTL = 4'b1011(非对称 PWM 波模式 1)，向上计数模式(DIR = 1'b0)，当计数值达到 CHxVAL 的值时，输出强制为高电平。

如果 CHxCOMCTL = 4'b1011(非对称 PWM 波模式 1)，向下计数模式(DIR = 1'b1)，当计数值达到 CHxCOMVAL_ADD 的值时，输出强制为低电平。

与复合 PWM 模式相比，非对称 PWM 模式只在中央对齐模式下有效，同时 CHxVAL 只在向上计数模式时有效，CHxCOMVAL_ADD 只在向下计数模式时有效。当 CHxVAL 或 CHxCOMVAL_ADD = 0 / CARL，通道输出被进行了特殊处理，通过 TIMERx_CHCTL2 寄存器的 CHxPERFOREN 位置位，强制 OxCPRE 输出高电平或低电平(根据所选非对称 PWM 模式确定)。

当 CHxVAL = 0，在非对称 PWM0 模式下，通道输出被强制为低电平，在非对称 PWM1 模式下，通道输出被强制为高电平。

当 CHxCOMVAL_ADD = 0，在非对称 PWM0 模式下，通道输出被强制为高电平，在非对称 PWM1 模式下，通道输出被强制为低电平。

当 CHxVAL = CARL，在非对称 PWM0 模式下，通道输出被强制为低电平，在非对称 PWM1 模式下，通道输出被强制为高电平。

当 CHxCOMVAL_ADD = CARL，在非对称 PWM0 模式下，通道输出被强制为高电平，在非对称 PWM1 模式下，通道输出被强制为低电平。

PWM 的周期取决于 2*CARL，PWM 脉冲宽度可以下*表 23-3 非对称 PWM 脉冲宽度*计算。

表 23-3 非对称 PWM 脉冲宽度

条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD <= CARL 或者 CHxCOMVAL_ADD < CHxVAL <= CARL 或者 CHxVAL = CHxCOMVAL_ADD	非对称 PWM 模式 0	CHxCOMVAL_ADD + CHxVAL
	非对称 PWM 模式 1	2*CARL - CHxCOMVAL_ADD - CHxVAL
CHxVAL > CARL > CHxCOMVAL_ADD	非对称 PWM 模式 0	100%
	非对称 PWM 模式 1	0%
CHxCOMVAL_ADD > CARL > CHxVAL	非对称 PWM 模式 0	0%
	非对称 PWM 模式 1	100%
(CHxVAL>CARL)且 (CHxCOMVAL_ADD > CARL)	-	CHx_O 输出保持

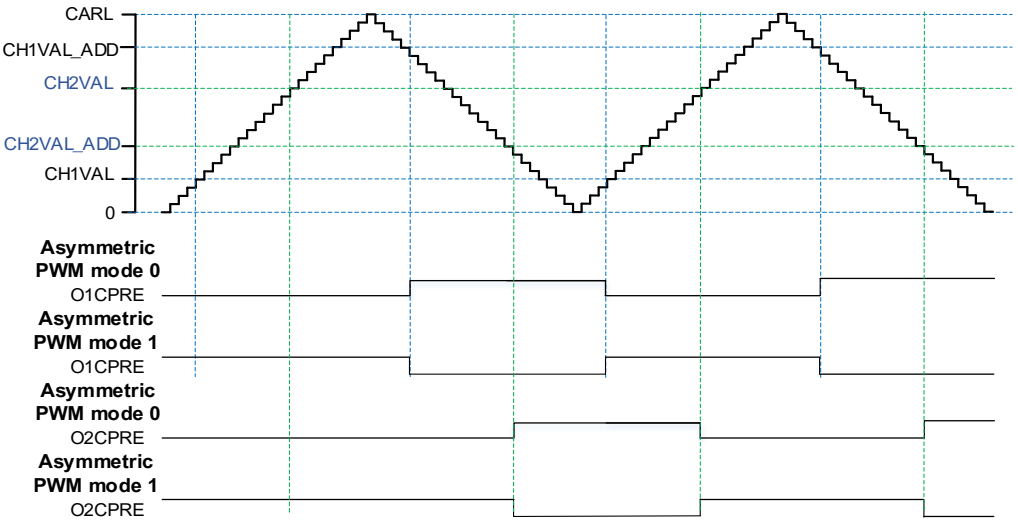
当计数器计数到 CHxVAL，CHxIF 位置 1 且如果 CHxIE=1 通道 x 产生中断，如果 CHxDEN=1，则产生 DMA 请求。当计数器计数到 CHxCOMVAL_ADD 时，CHxCOMADDIF 位置 1(该中断标志位只在复合 PWM 模式和非对称 PWM 模式有效，当 CHxCOMCTL[3:2]= 2'b10)，如果 CHxCOMADDIE = 1 通道 x 附加比较中断产生(只有中断产生，没有 DMA 请求响应)。

根据 CHxVAL，CHxCOMVAL_ADD 和 CARL 之间的关系，可以分为四种情况：

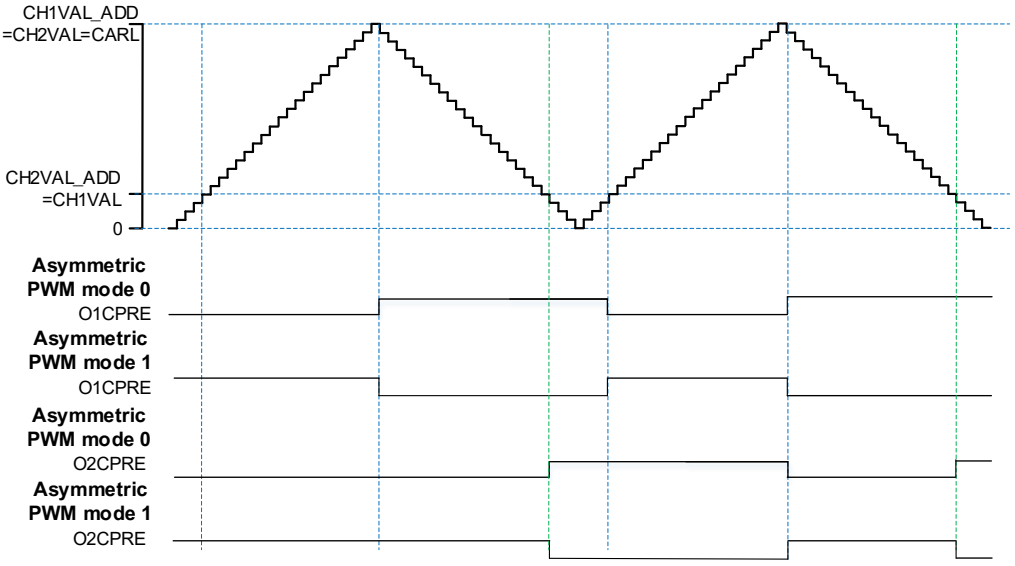
1）CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23.23 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)

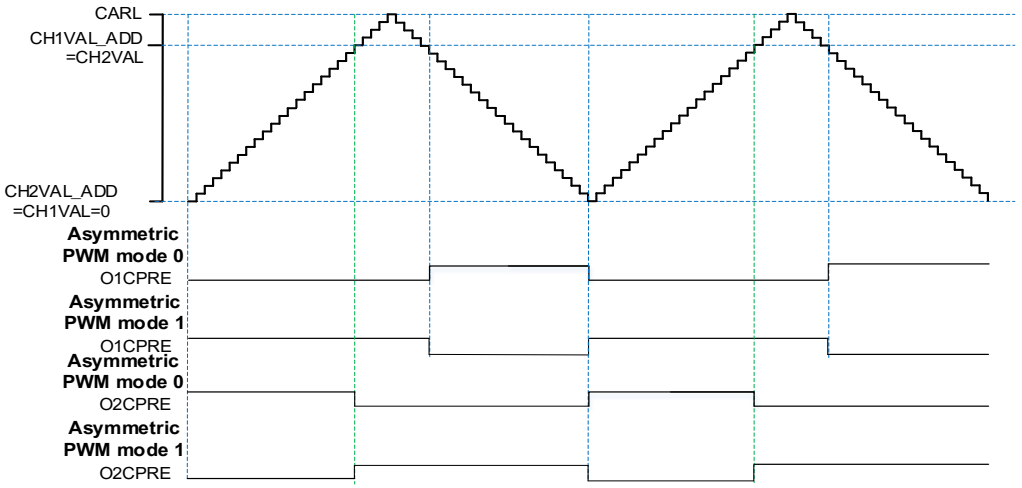
a. 0 < CH1VAL < CH1VAL_ADD < CARL / 0 < CH2VAL_ADD < CH2VAL < CARL



b. 0 < CH1VAL < CH1VAL_ADD = CARL / 0 < CH2VAL_ADD < CH2VAL = CARL，同时 CH1VAL = CH2VAL_ADD



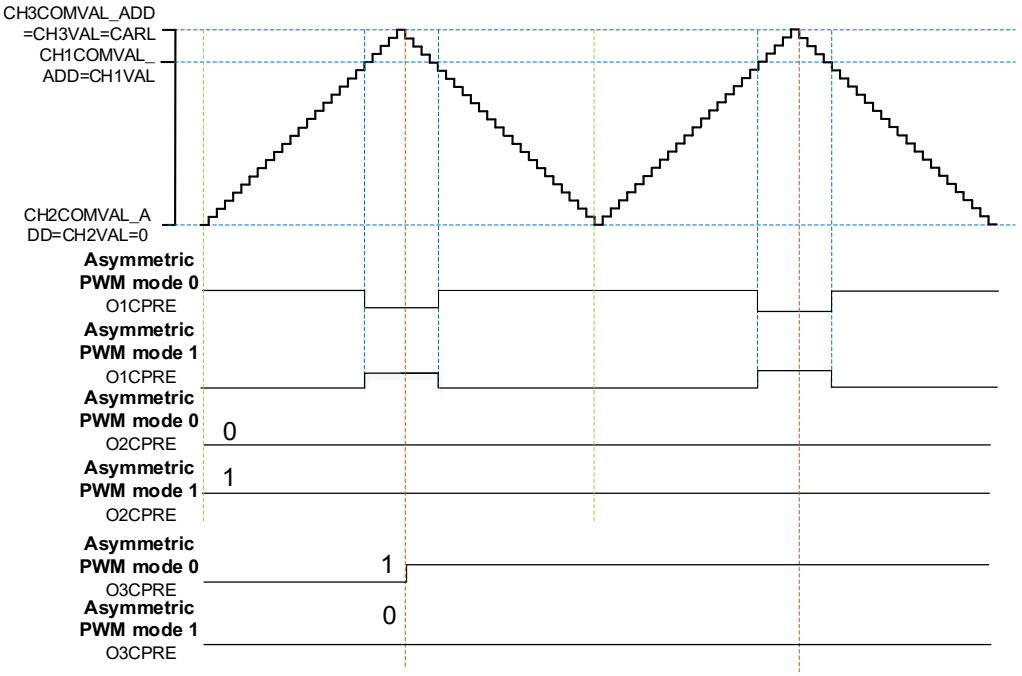
c. 0 = CH1VAL < CH1VAL_ADD < CARL / 0 = CH2VAL_ADD < CH2VAL < CARL，同时CH1VAL_ADD = CH2VAL



2）CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23.24. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD)

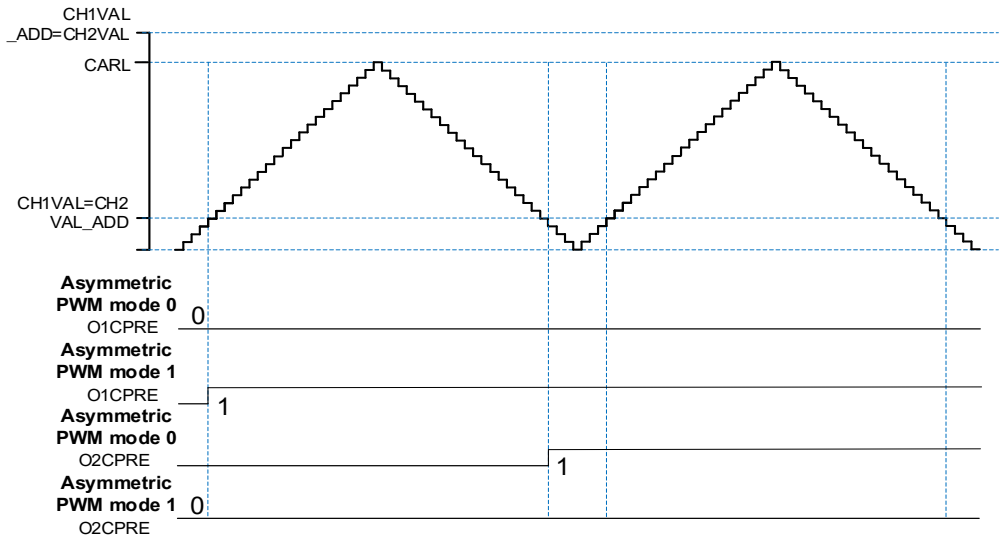
0 < CH1VAL = CH1VAL_ADD < CARL / CH2VAL = CH2VAL_ADD = 0 / CH3VAL = CH3VAL_ADD = CARL



3）CHxVAL或CHxCOMVAL_ADD值大于CARL。

图 23.25. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL)

0 < CH1VAL < CARL < CH1VAL_ADD / 0 < CH2VAL_ADD < CARL < CH2VAL



输出匹配脉冲选择

当发生匹配事件时，CHx_O(x=0..3)的输出由 CHxCOMCTL[3:0](x=0..3)位设置，通过配置 CHxOMPSEL[1:0](x=0..3)位，可选择 CHx_O(x=0..3)的输出信号正常或者脉冲。

当匹配事件发生时，CHxOMPSEL[1:0](x=0..3)用于选择 OxCPRE 信号输出(驱动 CHx_O)：

- CHxOMPSEL = 2'b00，OxCPRE信号根据CHxCOMCTL[3:0]位的配置正常输出；
- CHxOMPSEL = 2'b01，只有在计数器向上计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b10，只有在计数器向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b11，无论计数器向上计数还是向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；

图 23-26. 边沿对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00)

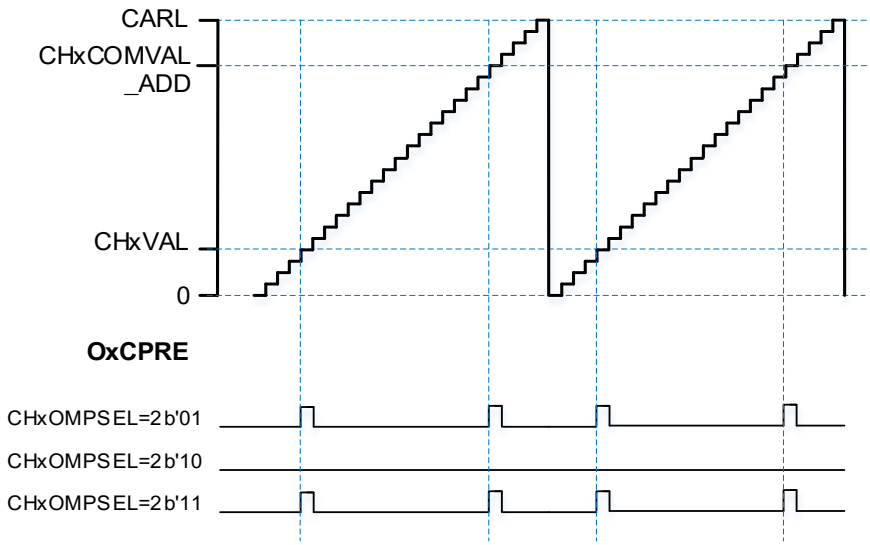
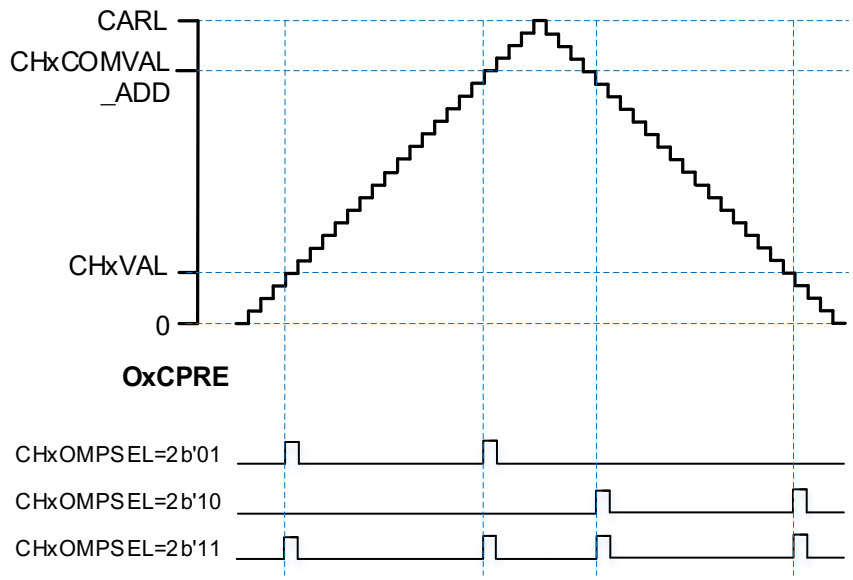


图 23-27. 中央对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00)



通道输出参考信号

如*图 23-13. 输出比较逻辑*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 信号(通道 x 参考信号)。

OxCPRE 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。

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(OCPRECIVM = 1)或者上溢/下溢事件(OCPRECIVM = 0)到来时，OxCPRE 信号才会回到有效电平状态。

互补输出

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有 4 对通道，所有 4 对通道都具有此功能。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN&CHPOENx(x=0...2)、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

当 CHx_O 和 CHx_ON 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考*表 23-4. 由参数控制的互补输出表*。

表 23-4. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN &CHP OENx</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td>0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td rowspan="3">1</td><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2):
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP)；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态:
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP)；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP
CHx_ON输出使能</td></tr>
</table>

互补参数					输出状态	
POEN &CHPOENx	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁能
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON =(!OxCPRE)(5)⊕CHxNP CHx_ON输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON =OxCPRE⊕CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON =(!OxCPRE)⊕CHxNP CHx_ON输出使能

注意：

（1） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平(CHx_O = 0⊕CHxP = CHxP)；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

死区时间插入

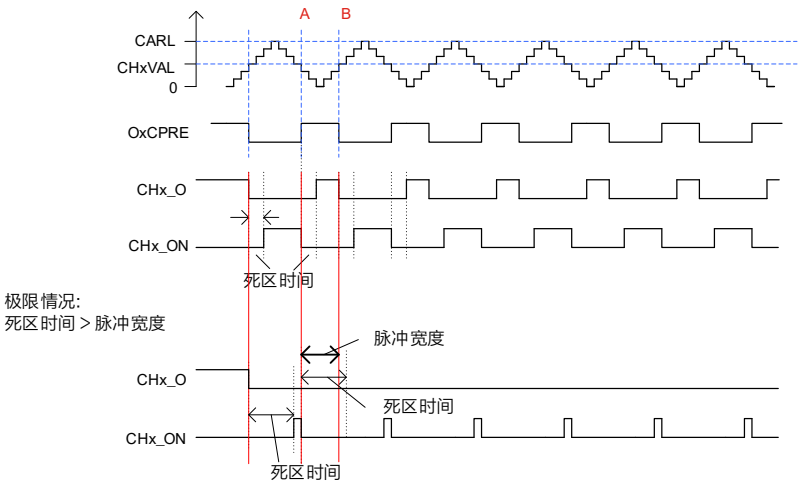
设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN&CHPOENx(x=0...2)，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考*互补通道保护寄存器 0(TIMERx_CCHP0)*。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时(TIMERx 计数器=TIMERx_CHxCV)，OxCPRE 翻转。在*图 23-28. 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 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配(TIMERx 计数器=TIMERx_CHxCV)，OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O 信号一直为无效值。如*图 23-28. 带死区时间的互补输出*所示。

图 23-28. 带死区时间的互补输出



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 CHx_ON 将不会插入死区。

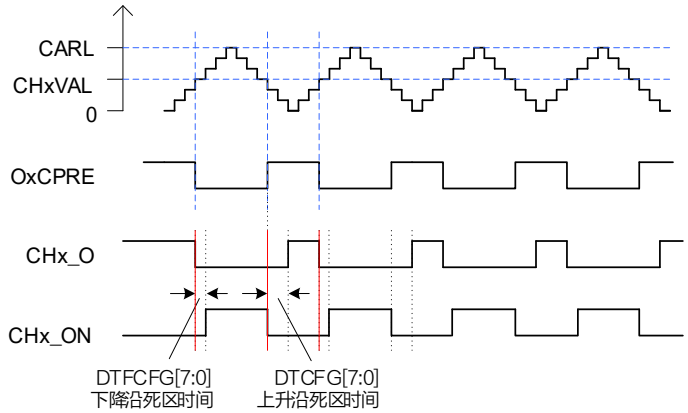
不同的死区时间插入

当 DTDIFEN 位(在 TIMERx_CCHP1 寄存器中)设置为 1 时，CHx_O 和 CHx_ON 信号可以输出不同的死区时间，具体如*图 23-29. 不同死区时间的互补输出(DTDIFEN=1)*所示。

通道输出准备信号 OxCPRE 上升沿的死区时间由 TIMERx_CCHP0 寄存器中的 DTCFG[7:0]位域配置。OxCPRE 信号的下降沿的死区时间由 TIMERx_CCHP1 寄存器中的 DTFCFG[7:0]位域配置。

DTDIFEN 位必须在使能计数器之前写入，且当 CEN=1 时不能修改。

图 23-29. 不同死区时间的互补输出(DTDIFEN=1)



中止功能

使用中止功能时，CHx_O 和 CHx_ON 信号的输出电平由以下位控制：TIMERx_CCHP0 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。在这种情况下，CHx_O 和 CHx_ON 信号不能同时设置为有效电平。

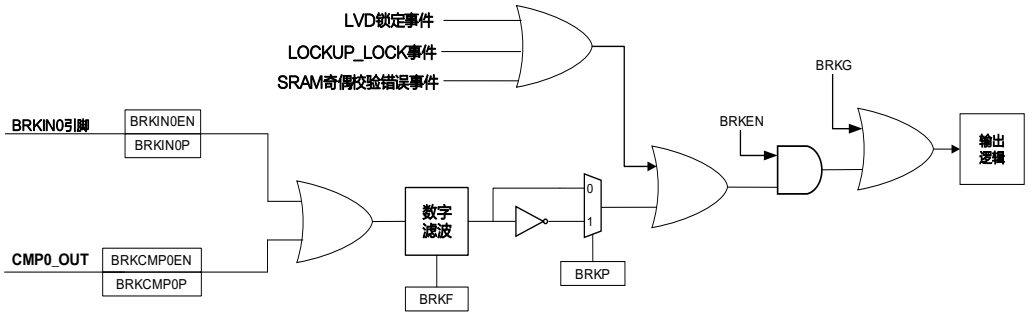
中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKIN0输入；
- 系统源： LVD锁定事件，Cortex®-M33锁定事件或SRAM奇偶校验错误事件；
- 片上外设源：比较器输出。

中止事件也可以由软件置位 TIMERx_SWEVG 寄存器中的 BRKG 位产生。

BREAK 中止功能逻辑如*图 23-30. BREAK 中止功能逻辑图*所示，其中 BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚。

图 23-30. BREAK 中止功能逻辑图



BREAK 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 BREAK 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；

当发生 BREAK 中止事件时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 的输出由 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 位确定。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。当 IOS=1 时，通道输出情况如*图 23-31. 通道响应 BREAK*

*中止输入(高电平有效)时，输出信号的行为(IOS=1)*所示，首先通道互补输出为复位状态，然后死区时间发生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

图 23-31. 通道响应 BREAK 中止输入(高电平有效)时，输出信号的行为(IOS=1)

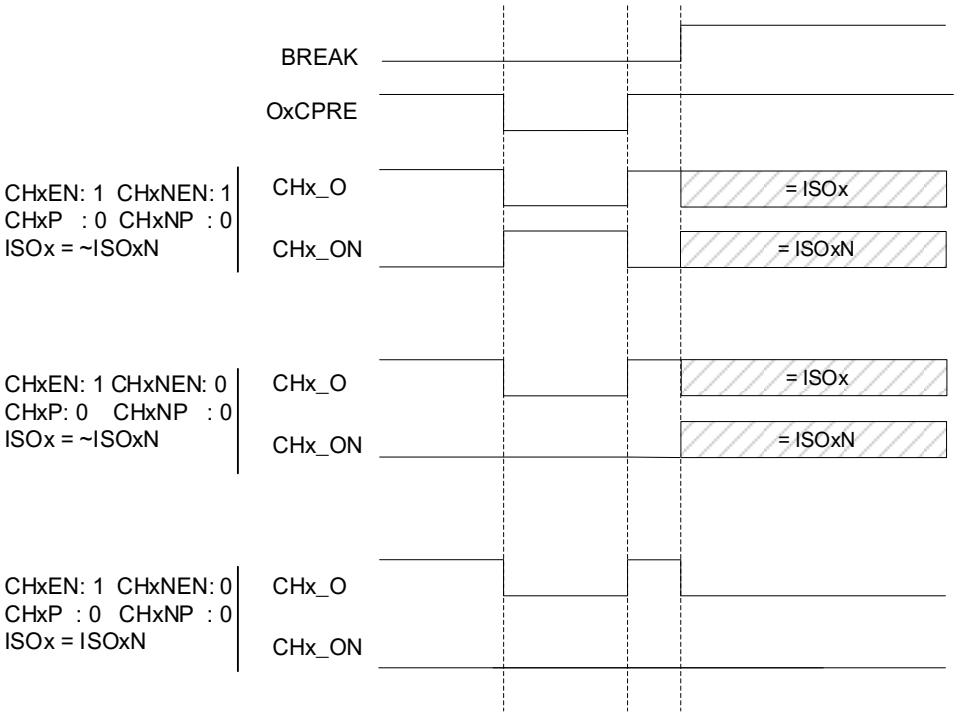


表 23-5. BREAK 输入信号时，TIMER 互补通道输出情况(break 输入高电平有效)

BREAK 输入	Output Status	
	CHx_O	CHx_ON
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：CHx_ON 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：CHx_ON 输出禁能(无效)
低电平	CHx_O 输出禁能(无效)	CHx_ON 输出禁能(无效)

发生中止事件时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

通过配置 TIMERx_CTL2 寄存器中的 BRKENCHx(x = 0..3)位，可实现对每对通道的中止功能进行独立控制。当 BRKENCHx(x = 0..3)位为"0"且发生中止事件时，相应的通道 CHx_O 和 CHx_ON 输出保持不变。

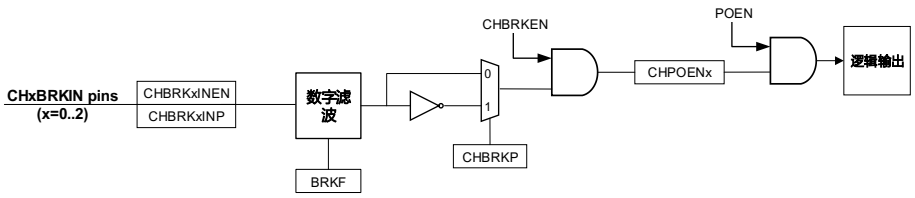
为了增强中止功能的灵活性，对于高级定时器，每对通道都有单独的中止输入 CHxBRKIN(x=0...2)，当相应的 CHxBRKIN(x=0...2)输入有效电平时，可以异步对相应的 CHPOENx(x=0...2)清 0，可实现单独的中止输入控制对应的每对通道中止后输出。通过在 TIMERx_CCHP0 寄存器中设置 CHBRKEN 位，可以开启通道中止。通过在 TIMERx_CCHP0 寄存器中设置 CHBRKP 位，可以配置通道中止的极性。

此外，如果启用了通道的中止功能，CHx_O 和 CHx_ON(x=0...2)不仅受 POEN 控制，还受

CHPOENx(x=0...2)控制。CHx 和 CHx_ON(x=0...2)在被中止后的输出与 BREAK 功能相同。每个通道中止输入都有自己的使能位 CHBRKxINEN(x=0...2)和极性控制位 CHBRKxINP(x=0...2)，并与 BREAK 有相同的滤波配置位域 BRKF[3:0]。请参考*图 23-32. 通道中止功能逻辑图*。

当通道 x(x=0...2)的 BREAK 输入有效电平时，相应的通道中止标志位 CHxBRKIF(x=0...2)会被设置为 1。如果通道 x(x=0...2)的 BREAK 输入在多个周期内保持有效状态，标志位 CHxBRKMIF(x=0...2)将被设置为 1。周期计数可以通过配置 TIMERx_CHBRKPER 寄存器来确定。

图 23-32. 通道中止功能逻辑图



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, 1, 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 23-6. 不同译码器模式下的计数方向*所示。其中，CI0FE0、CI1FE1 是经过滤波和极性选择后的 CI0、CI1 信号。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置 TIMERx_CAR 寄存器。

表 23-6. 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 23-33. 译码器接口模式下计数器运行例子

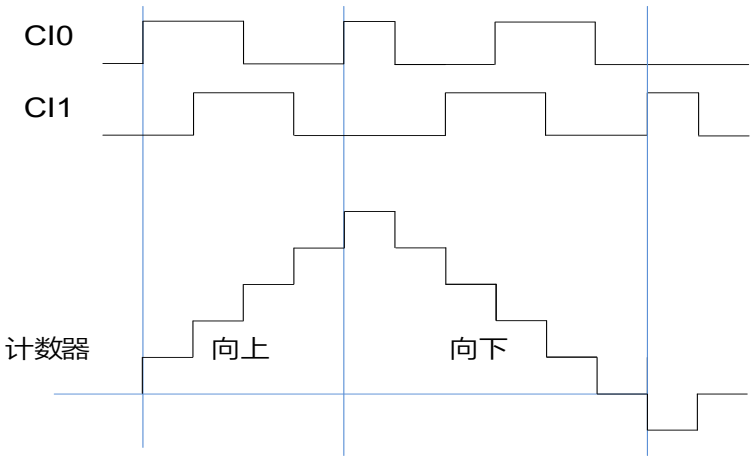
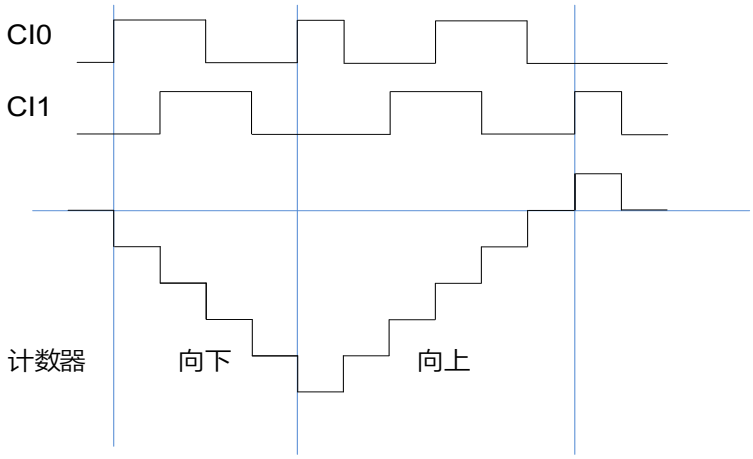


图 23-34. CI0FE0 极性反相的译码器接口模式下的例子



正交译码器和译码器的时钟输出

定时器可以通过 TRGO 输出译码器时钟输出信号。该功能仅用于正交译码器模式 0~2，通过将 TIMERx_CTL1 寄存器中 MMC[3:0]位域设置为 4'b1000 来使能。

霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

*图 23-35. 霍尔传感器用在 BLDC 电机控制中*是定时器和电机的连接示意图。众所周知，我们要两个定时器。TIMER_in 定时器(可以是高级定时器或者通用 L0 定时器)接收来自电机霍尔传感器的三路信号，这三路信号是电机转子的位置信号。

三个霍尔传感器与 TIMER_in 定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接，例如 TRGO-ITIx，TIMER_in 定时器和 TIMER_out 定时器可以连接在

一起。TIMER_out 定时器根据 ITIx 触发信号输出 PWM 波，驱动 BLDC 电机，控制 BLDC 电机的速度。这样，TIMER_in 定时器和 TIMER_out 定时器的连接形成了一个反馈电路，可以根据需求改变配置。

TIMER_in 定时器需要具备输入异或功能，所以可以选择高级定时器和通用 L0 定时器。

TIMER_out 定时器需要具备互补输出和死区插入功能，所以可以选择高级定时器。

另外，可以通过 TRIGSEL 模块，选择互连的定时器，例如：

TIMER_in(TIMER0) -> TIMER_out(TIMER7 ITI0)

TIMER_in(TIMER1) -> TIMER_out(TIMER0 ITI1)

选择好合适的互连定时器，定时器和 BLDC 的线路也已经连接好，我们就可以配置定时器了。有以下关键配置：

- 设置TI0S，使能异或功能。三路输入信号的任何一路发生变化，CI0都会翻转，CH0VAL此时会捕获计数器的当前值。
- 设置CCUC和CCSE，使能ITIx直接连接到换相功能。
- 根据需求配置PWM参数。

图 23-35. 霍尔传感器用在 BLDC 电机控制中

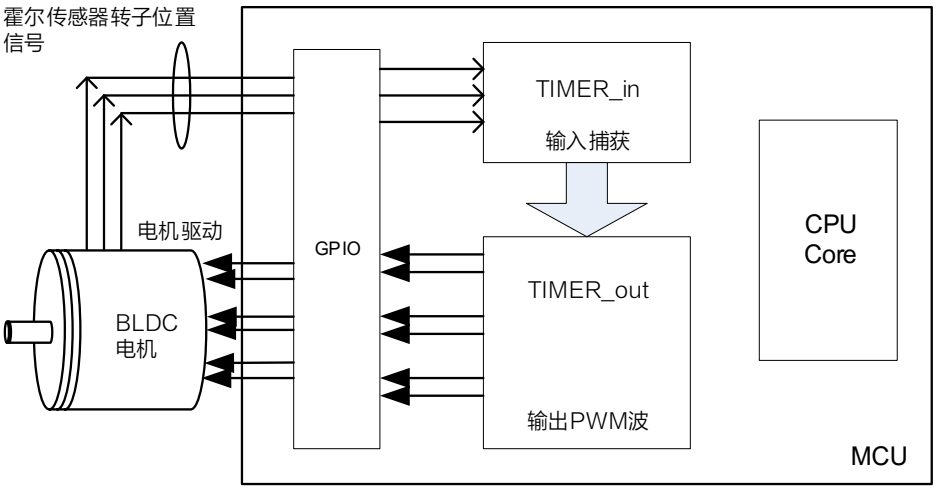
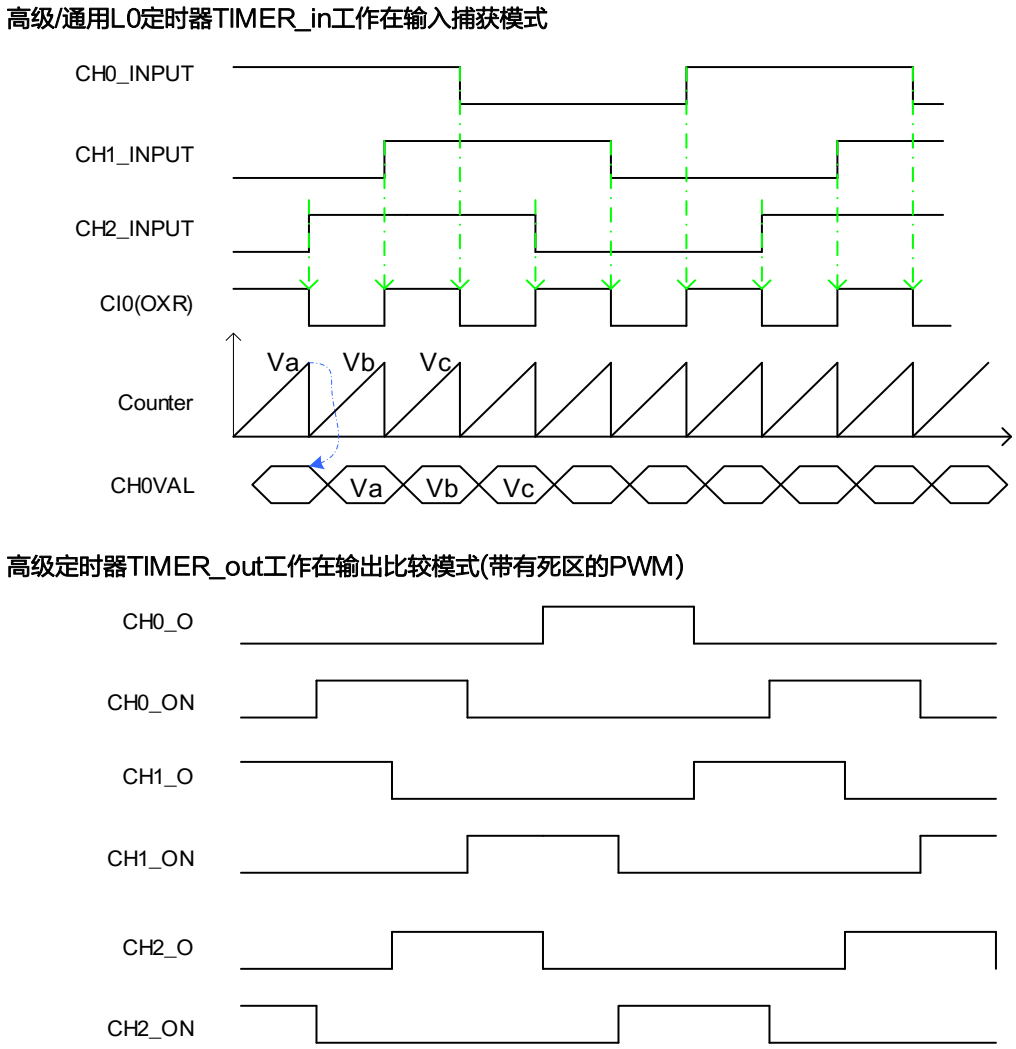


图 23-36. 两个定时器之间的霍尔传感器时序图

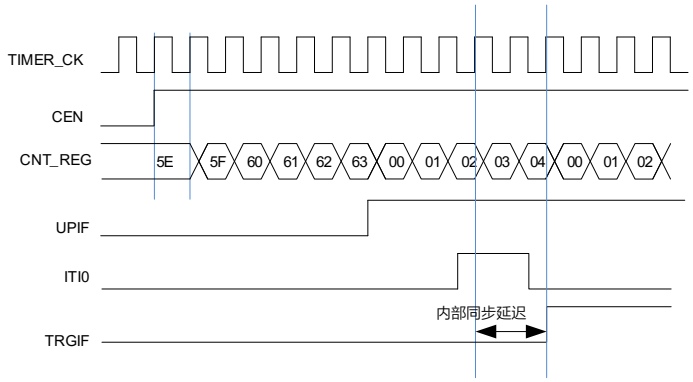
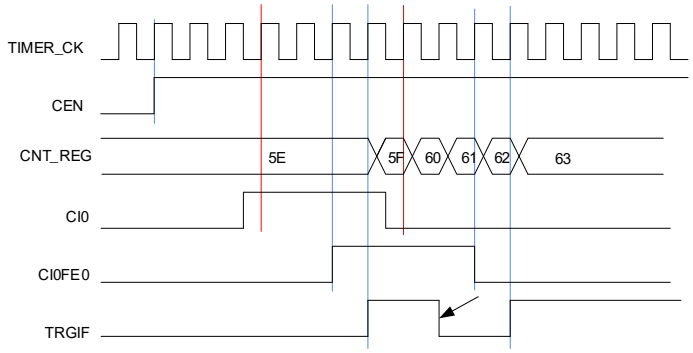


主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式等，可以通过设置 SYSCFG_TIMERxCFG(x=0,7)寄存器中的 TSCFGy[4:0] (y=3..6)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6)位域值来选择。

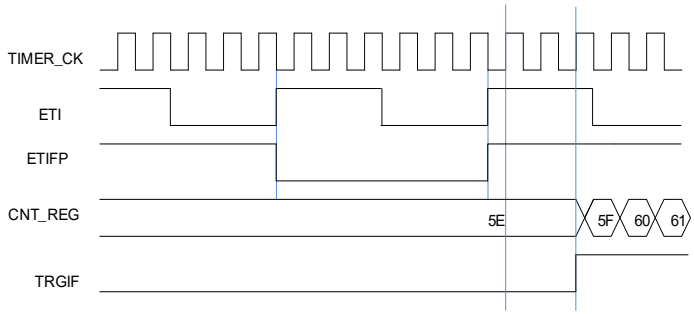
表 23-7. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y=3：复位模式 y=4：暂停模式 y=5：事件模式 y=6：外部时钟模式0	TSCFGy[4:0] 00000：Mode disable 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3	如果触发源是CIxFEx(x=0..3)，配置CHxP、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	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，配置

<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>00101：CI0F_ED
00110：CI0FE0
00111：CI1FE1
01000：ETIFP(1)
01001：CI2FE2
01010：CI3FE3</td><td>入)，配置ETP选择极性和反相</td><td>ETFC设置滤波，配置ETPSC设置预分频</td></tr>
<tr><td rowspan="2">例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TSCFG3[4:0]
5'b00001，选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 23-37. 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] 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 23-38. 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 23-39. 事件模式



单脉冲模式

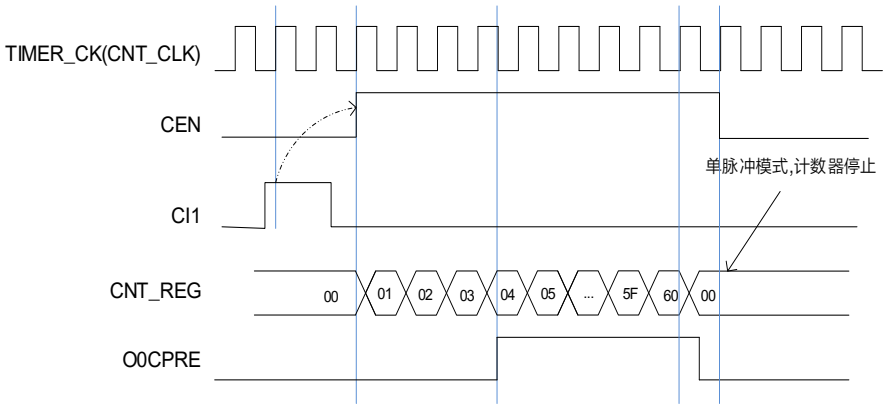
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，可使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。单脉冲模式下，触发上升沿产生之后，OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。

单脉冲模式也同样适用于复合 PWM 模式(CHxCPWMEN = 1'b1 和 CHxMS[2:0] = 3'b000)。

图 23-40. 单脉冲模式，TIMERx_CHxCV = 0x04 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MMC=4'b0010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期(TIMER2_CAR寄存器)；
3. 配置定时器0工作在外部时钟模式0，定时器0输入触发源为定时器2(SYSCFG_TIMER0CFG1寄存器中的TSCFG6[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)。

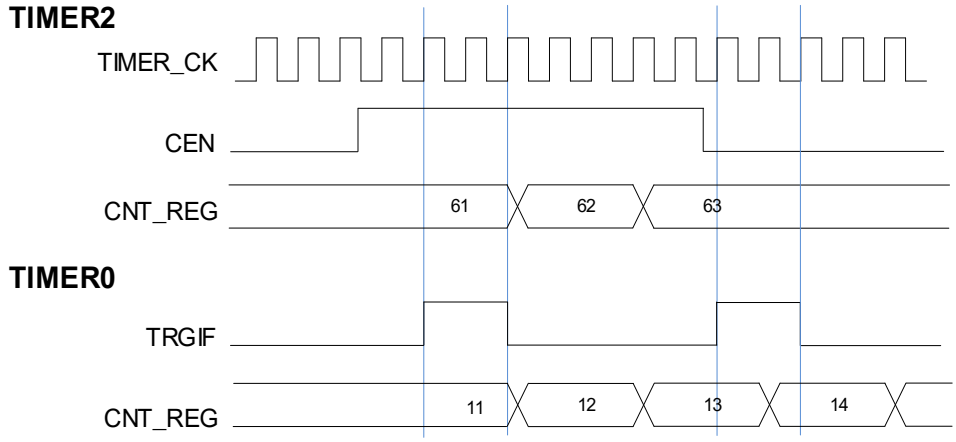
■ 用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见*图 23-41. 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MMC=4'b0001)；
2. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2(SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00011)；
3. 写1到CEN来开启定时器2(TIMER2_CTL0寄存器)。

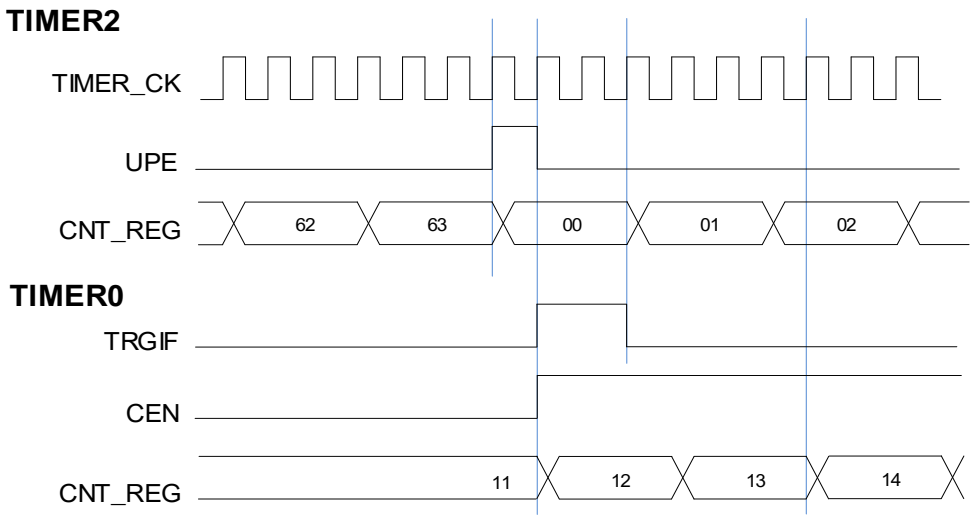
图 23-41. 用定时器 2 的使能信号触发定时器 0



在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。见*图 23-42. 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MMC=4'b0010)；
2. 配置定时器2的周期(TIMER2_CARL寄存器)；
3. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2(SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00011)
4. 写1到CEN来开启定时器2(TIMER2_CTL0寄存器)。

图 23-42. 用定时器 2 的更新事件来触发定时器 0

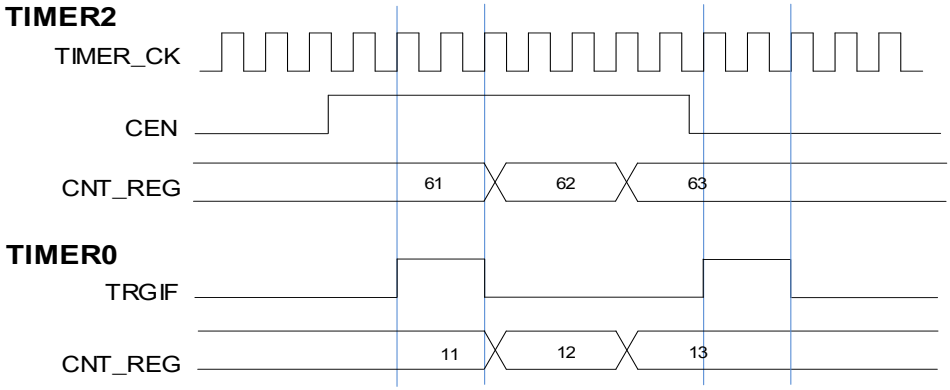


■ 使用定时器2的使能/O0CPRE参考信号来使能定时器0计数。

在这个例子中，我们使用定时器 2 的使能输出来控制定时器 0 的使能。如*图 23-43. 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MMC=4'b0001)；
2. 配置定时器0工作在暂停模式，定时器0输入触发源为定时器2(SYSCFG_TIMER0CFG0寄存器中的TSCFG4[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)。

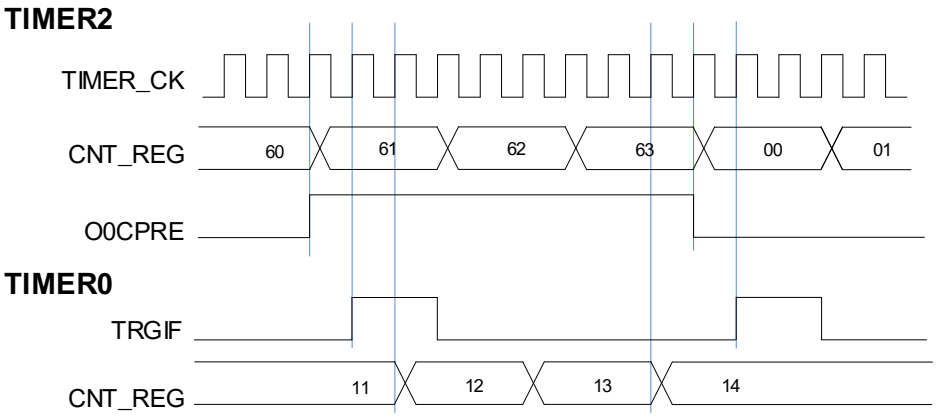
图 23-43. 用定时器 2 的使能来选通定时器 0



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TSCFG4[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 23-44. 用定时器 2 的 O0CPRE 信号选通定时器 0



■ 使用一个外部触发来同步两个定时器

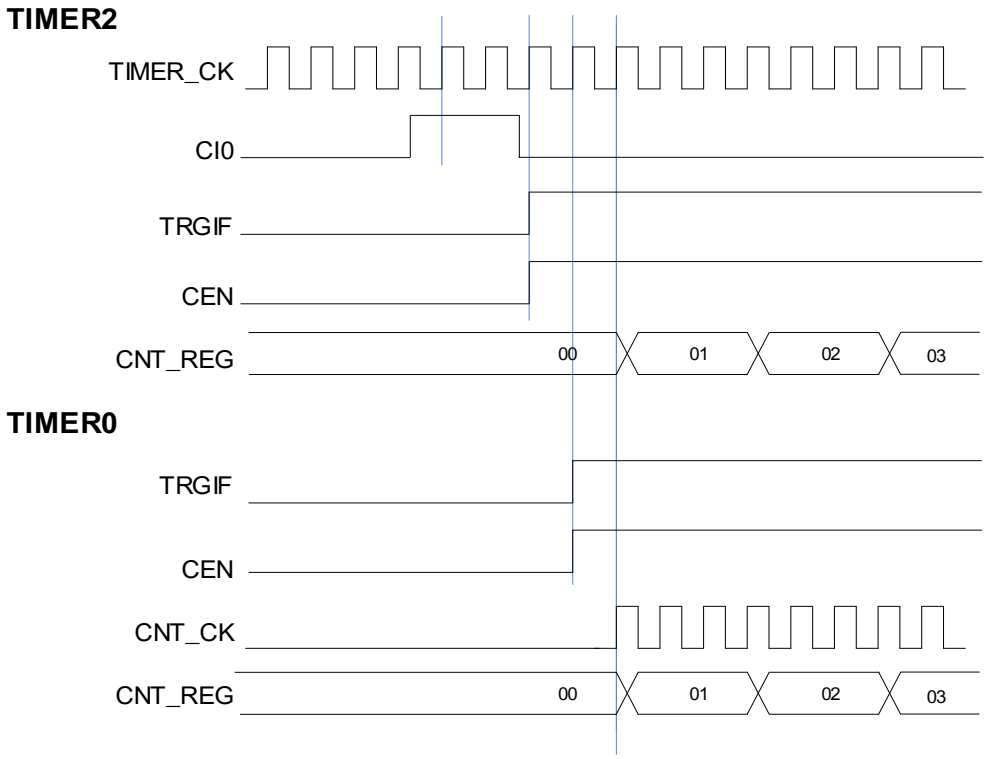
配置定时器 2 的使能信号触发定时器 0 的开启，配置定时器 2 的 CI0 输入信号上升沿来触发定时器 2。为了确保两个定时器同步开启，定时器 2 必须配置在主/从模式。步骤如下：

1. 配置定时器2工作在事件模式，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TSCFG5[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TSCFG5[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 23-45. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



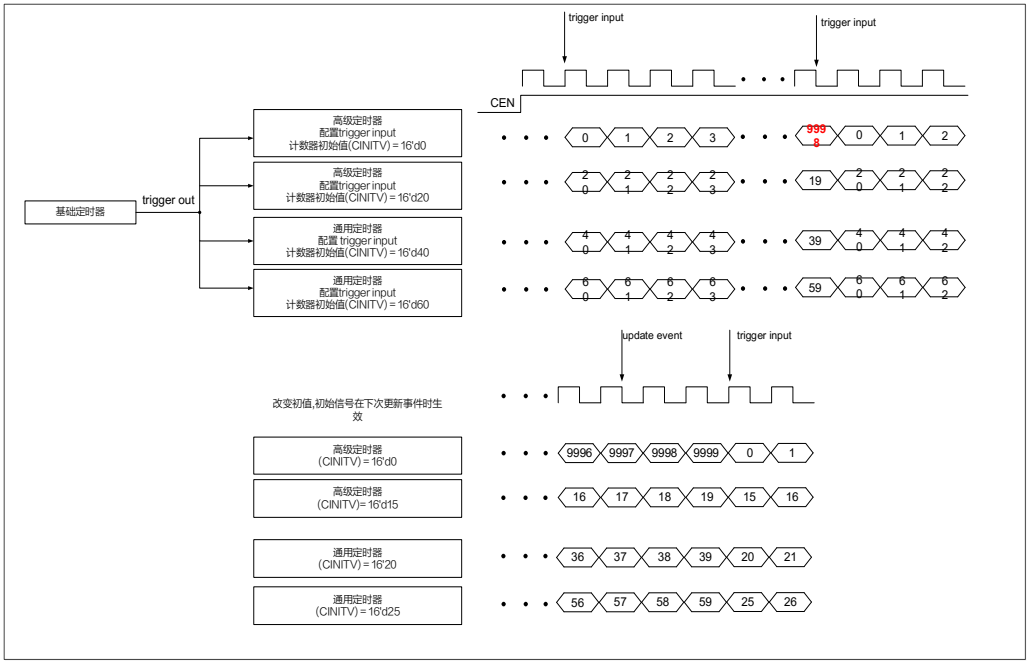
计数器同步、初始方向和值刷新

在一些菊链的配置下，多个计数器被触发并且被同步在同一个时刻开始计数。由于不同的计数器之间在长时间计数之后可能会产生相位偏移，因此通过外部触发定时地刷新同步计数器来消除硬件上产生的多个计数器之间产生的相位偏移。此外，通过软件可配的计数器初始值配置(TIMERx_CINITV),可以控制多个被触发的计数器之间的相位偏移关系。

4 个从计数器被配置存在同一触发输入源，通过使能 TIMERx_CINITCTL 寄存器的 CINITVEN 位域,可以使能计数器的初值载入功能，4 个从计数器的载入初值分别位 0，20，40，60，因此之间产生了 20 个 CK_TIMER 的等差相移。

下图中的 9998 计数值，表明了在长时间计数之后，不同计数器的之间产生了相位偏移，通过基础定时器的触发输出，刷新同步从定时器的计数值，避免了相位偏移的累积。

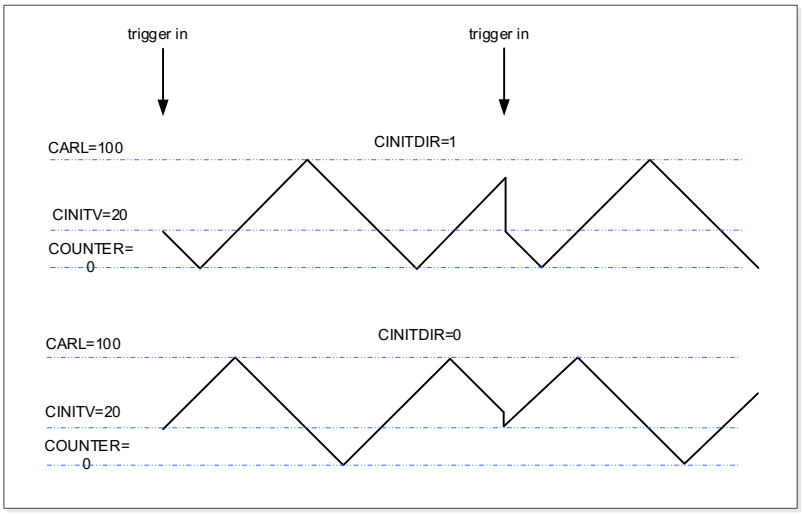
图 23.46. 用主计数器的 trigger 信号触发 4 个计数器



当计数器处于中央对齐的计数方式时，可以通过配置 TIMERx_CINITCTL 寄存器的 CINITDIR 位域来配置计数器复位之后的计数方向。CINITDIR 只有当使能了 CINITVEN 时才会生效，另外当计数器处于向上/向下计数模式下时，CINITDIR 的配置也不会生效。

当 CINITDIR 为 0 时，复位计数器之后的计数方向为向上；当 CINITDIR 为 1 时，复位计数器之后的计数方向为向下。如下图所示：

图 23.47. CINITDIR 为 0 或 1 时的计数方向



计数器的初始计数方向和初始值也可以通过软件同步事件刷新。将 TIMERx_CINITCTL 寄存器中的 SWSYNCG 位置 1，可以产生一个软件同步事件，用于刷新 TIMERx 计数器的初始计数方向和初始值。

当高级定时器使用主模式下来同步其他高级定时器时，在 TIMERx_CTL1 寄存器中 MMC[3:0]

位域需要设置为 4'b1001。此时，TIMERx 的软件同步事件(设置 SWSYNCG 位为 1 产生)可以作为 TRGO 信号输出，用于同步其他高级定时器。

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4、DMATA+0x8、DMATA+0xC 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL 寄存器中的 TIMERx_HOLD 位置 1 时，定时器的计数器停止计数。

23.1.5. TIMERx 寄存器(x=0,7,19)

TIMER0 基地址：0x4001 2C00

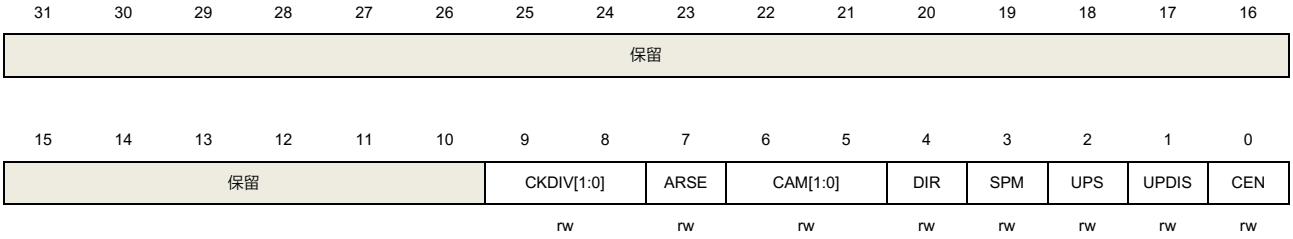
TIMER7 基地址：0x4001 3400

TIMER19 基地址：0x4001 4800

控制寄存器 0(TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置 CKDIV，规定定时器时钟(CK_TIMER)与死区时间和采样时钟(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=3'b000)，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=3'b000)，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=3'b000)，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				MMC[3]	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	MMC[2:0]			DMAS	CCUC[0]	保留	CCSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:26	保留	必须保持复位值。
25	MMC[3]	主模式控制 请参考 MMC[2:0]的描述。
24:16	保留	必须保持复位值
15	ISO3N	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	通道 2 的互补通道空闲状态输出 参考 ISO0N 位。
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位。
11	ISO1N	通道 1 的互补通道空闲状态输出 参考 ISO0N 位。
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位。
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN&CHPOENx(x=0...2)复位，CH0_ON 输出低电平 1：当 POEN&CHPOENx(x=0...2)复位，CH0_ON 输出高电平 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0：当 POEN&CHPOENx(x=0...2)复位，CH0_O 输出低电平 1：当 POEN&CHPOENx(x=0...2)复位，CH0_O 输出高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	TI0S	通道 0 触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0，CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 该位域控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功

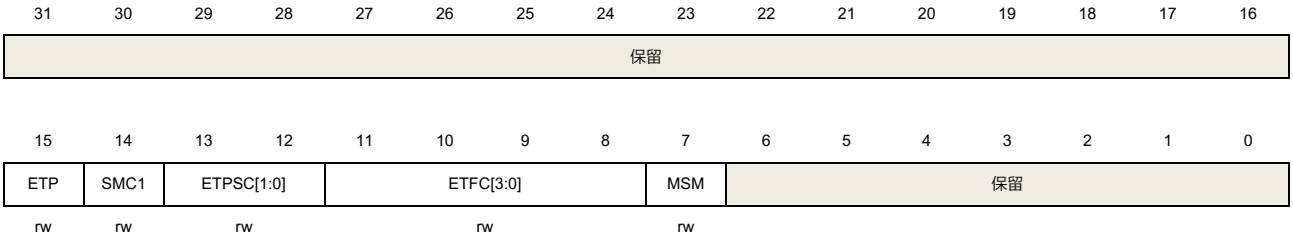
		能。 0000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲，后一种情况下，TRGO 上的信号相对实际的复位会有一个延迟。 0001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和 TRGO 上会有一个延迟，除非选择了主/ 从模式。 0010：更新。主模式控制器选择更新事件作为 TRGO。 0011：捕获/比较脉冲。通道 0 在发生一次捕获或一次比较成功时，主模式控制器产生一个 TRGO 脉冲。 0100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO。 0101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO。 0110：比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO。 0111：比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO。 1000：译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出TRGO。此值仅用于正交译码器模式0~2。 1001：在这种模式下主模式控制器选择软同步事件信号(设置SWSYNCG位为1产生)作为触发输出TRGO。 1010~1111：保留
3	DMAS	DMA 请求源选择 0：当通道捕获/比较事件发生时，发送通道 CHx 的 DMA 请求 1：当更新事件发生，发送通道 CHx 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 TRGI 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、CHxNEN 和 CHxCOMCTL 位禁能

1：影子寄存器 CHxEN、CHxNEN 和 CHxCOMCTL 位使能
如果这些位已经被写入了，换相事件到来时这些位才被更新。
当通道没有互补输出时，此位无效。

从模式配置寄存器(TIMERx_SMCFG)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



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 TRGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:0	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											
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。
27:15	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	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		SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	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。
27:14	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	保留	必须保持复位值
7	BRKIF	BREAK 中断标志位 一旦 BREAK 输入有效，由硬件对该位置'1'。如果 BREAK 输入无效，则该位可由软件清'0'。 0：无 BREAK 事件产生 1：BREAK 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											
w	w	w	w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31	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。
27:8	保留	必须保持复位值
7	BRKG	产生 BREAK 事件 该位由软件置 1，用于产生一个 BREAK 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生 BREAK 事件 1：产生 BREAK 事件
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0。当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0。当此位被置 1，通道捕获/比较控制寄存器(CHxEN、CHxNEN 和 CHxCOMCTL 位)的互补输出被更新。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和

DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。
0：不产生通道 0 捕获或比较事件
1：发生通道 0 捕获或比较事件

0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0(TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	保留	CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留					保留								保留
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 附加比较输出影子寄存器 1：使能通道 0 附加比较输出影子寄存器

		仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27	CH1ADDUPS	通道1附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH1COMV_ADD寄存器 1：在发生计数器计数值匹配CH1VAL值时，更新TIMERx_CH1COMV_ADD寄存器
26	CH0ADDUPS	通道0附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH0COMV_ADD寄存器 1：在发生计数器计数值匹配 CH0VAL 值时，更新 TIMERx_CH0COMV_ADD 寄存器
25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域定义了通道的方向和输入信号的选择。只有当通道关闭(当 TIMERx_CHCTL2 寄存器的 CH1EN 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)。 100~111：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 和 CH0_ON 的值。O0CPRE 高电平有效，而 CH0_O 和 CH0_ON 的有效电平取决于 CH0P 和 CH0NP 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：保留。 1001：保留。 1010：非对称 PWM 模式 0，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为无效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为有效电平。 1011：非对称 PWM 模式 1，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为有效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为无效电平。 1100~1111：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_O 和 CH0_ON 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000(比较模式)时，此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器

		仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为 3 个时钟周期
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(当 TIMERx_CHCTL2 寄存器的 CH0EN 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)。 100~111：保留

输入捕获模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 模式选择 与输出模式相同。
30	CH0MS[2]	通道 0 模式选择 与输出模式相同。
29:16	保留	必须保持复位值.
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述。
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述。
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 CI0 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，$f_{SAMP}= f_{DTS}$，N=1

0001：$f_{SAMP} = f_{CK_TIMER}$，N=2
0010：$f_{SAMP} = f_{CK_TIMER}$，N=4
0011：$f_{SAMP} = f_{CK_TIMER}$，N=8
0100：$f_{SAMP}=f_{DTS}/2$，N=6
0101：$f_{SAMP}=f_{DTS}/2$，N=8
0110：$f_{SAMP}=f_{DTS}/4$，N=6
0111：$f_{SAMP}=f_{DTS}/4$，N=8
1000：$f_{SAMP}=f_{DTS}/8$，N=6
1001：$f_{SAMP}=f_{DTS}/8$，N=8
1010：$f_{SAMP}=f_{DTS}/16$，N=5
1011：$f_{SAMP}=f_{DTS}/16$，N=6
1100：$f_{SAMP}=f_{DTS}/16$，N=8
1101：$f_{SAMP}=f_{DTS}/32$，N=5
1110：$f_{SAMP}=f_{DTS}/32$，N=6
1111：$f_{SAMP}=f_{DTS}/32$，N=8

3:2	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同。

通道控制寄存器 1(TIMERx_CHCTL1)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	CH3ADDUPS	CH2ADDUPS	保留	CH3COMCTL[3]	保留							CH2COMCTL[3]
		保留	保留	保留	保留		保留								保留
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29	CH3COMADDSEN	通道 3 附加输出比较影子寄存器使能 参考 CH2COMADDSEN 描述。
28	CH2COMADDSEN	通道 2 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2COMV_ADD 寄存器的影子寄存器使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 附加输出/比较影子寄存器 1：使能通道 2 附加输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
27	CH3ADDUPS	通道3附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH3COMV_ADD寄存器 1：在发生计数器计数值匹配 CH3VAL 值时，更新 TIMERx_CH3COMV_ADD 寄存器
26	CH2ADDUPS	通道2附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH2COMV_ADD寄存器 1：在发生计数器计数值匹配 CH2VAL 值时，更新 TIMERx_CH2COMV_ADD 寄存器
25	保留	必须保持复位值.
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 CH2COMCTL[2:0]描述
23:17	保留	必须保持复位值.
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 CH2COMCTL[2:0]描述
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较控制 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH3COMFEN	通道3输出比较快速使能

		参见 CH2COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(当 TIMERx_CHCTL2 寄存器的 CH3EN 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)。 100~111：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 和 CH2_ON 的值。O2CPRE 高电平有效，而 CH2_O 和 CH2_ON 的有效电平取决于 CH2P 和 CH2NP 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：保留。 1001：保留。 1010：非对称 PWM 模式 0，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为无效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为有效电平。 1011：非对称 PWM 模式 1，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为有效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为无效电平。

		1100~1111：保留 **注意**：在复合 PWM 模式下(CH2CPWMEN = 1'b1 和 CH2MS = 3'b000)，通道 2 的 PWM 输出信号由 TIMERx_CH2CV 和 TIMERx_CH2COMV_ADD 寄存器共同确定。详细信息请参考***复合 PWM 模式***。 在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O2CPRE 电平才改变。 当 CH2_O 和 CH2_ON 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH2MS =000(比较模式)时此位不能被改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
2	CH2COMFEN	通道2输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。 0：通道2输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为 3 个时钟周期
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(当 TIMERx_CHCTL2 寄存器的 CH2EN 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)。 100~111：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

该寄存器可以按半字(16 位)或字(32 位)访问。

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

CH3PERFOREN	CH2PERFOREN	CH1PERFOREN	CH0PERFOREN	保留
rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	CH3NEN	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH3PERFOREN	Channel 3在复合PWM或者非对称PWM模式下，周期点匹配时刻O3CPRE电平值设置。 参考CH0PERFOREN描述。
30	CH2PERFOREN	Channel 2在复合PWM或者非对称PWM模式下，周期点匹配时刻O2CPRE电平值设置。 参考CH0PERFOREN描述。
29	CH1PERFOREN	Channel 1在复合PWM或者非对称PWM模式下，周期点匹配时刻O1CPRE电平值设置。 参考CH0PERFOREN描述。
28	CH0PERFOREN	Channel 0在复合PWM或者非对称PWM模式下，周期点匹配时刻O0CPRE电平值设置。 0：禁能 1：复合PWM0或者非对称PWM0模式下，周期点匹配时刻电平值强制拉高，复合PWM1或者非对称PWM1模式下，周期点匹配时刻电平值强制拉低。 注意：中央对齐模式下，周期点为计数器下溢时刻；边沿对齐模式下，周期点为计数器溢出时刻。
27:16	保留	必须保持复位值
15	CH3NP	通道 3 互补捕获/比较极性 参考 CH0NP 描述。
14	CH3NEN	通道 3 互补捕获/比较使能 参考 CH0NEN 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	CH2NP	通道 2 互补捕获/比较极性 参考 CH0NP 描述。
10	CH2NEN	通道 2 互补捕获/比较使能 参考 CH0NEN 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	CH1NP	通道 1 互补捕获/比较极性 参考 CH0NP 描述。
6	CH1NEN	通道 1 互补捕获/比较使能 参考 CH0NEN 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	CH0NP	通道 0 互补捕获/比较极性 当通道 0 配置为输出模式，此位定义了通道 0 输出信号 CH0_ON 的极性。 0：通道 0 互补输出高电平有效 1：通道 0 互补输出低电平有效 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	CH0NEN	通道 0 互补捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_ON 信号有效。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
1	CH0P	通道 0 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。[CH0NP, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。 10：保留。 11：把通道 0 输入信号的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为

输入模式时，将此位置 1 使能通道 0 上的捕获事件。
0：禁止通道 0
1：使能通道 0

计数器寄存器(TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器(TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

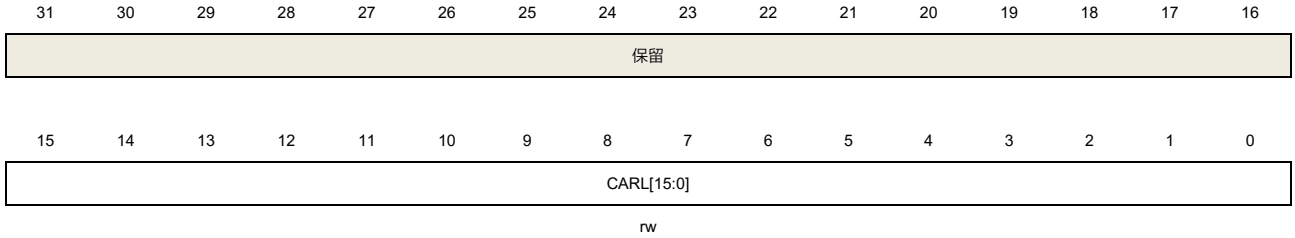
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

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。

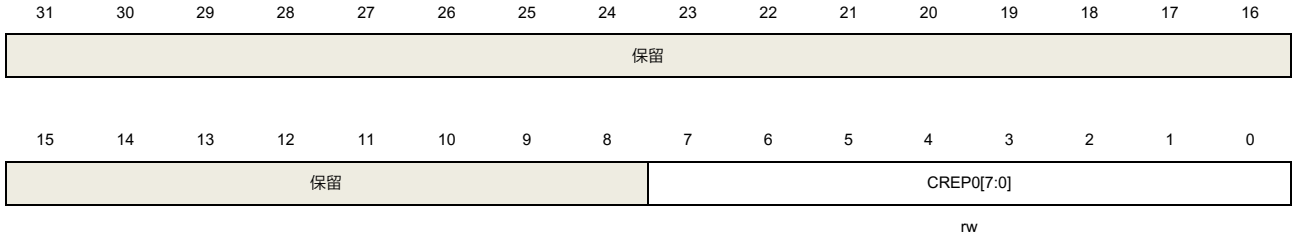


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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。

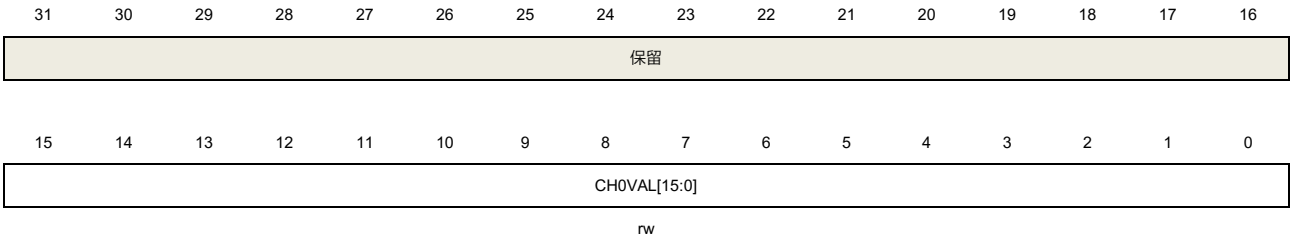


位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP0[7:0]	重复计数器的值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

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。

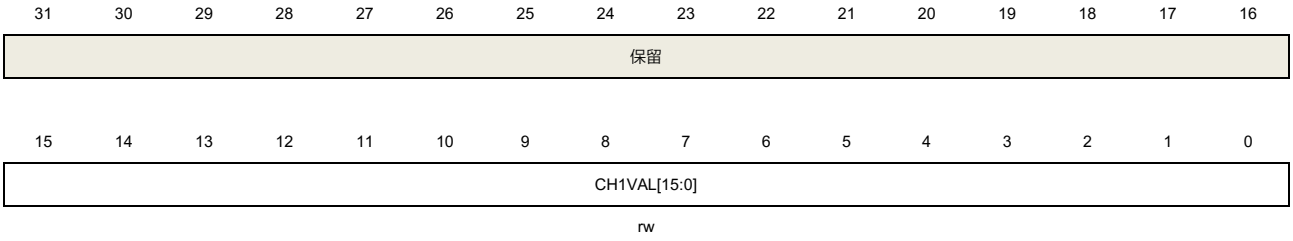


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



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 0(TIMERx_CCHP0)

地址偏移：0x44
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31	CHBRKP	通道中止输入信号极性 该位用于配置通道中止输入信号的极性。 0：通道中止输入信号低电平有效 1：通道中止输入信号高电平有效 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
30	CHBRKEN	通道中止输入信号使能 该位置1时，使能通道中止输入信号。 0：通道中止输入禁能 1：通道中止输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
29:20	保留	必须保持复位值
19:16	BRKF[3:0]	BREAK和通道中止输入信号滤波 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 BREAK 和通道中止输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK 和通道中止异步有效，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$, N=2 0010：$f_{SAMP} = f_{CK_TIMER}$, N=4 0011：$f_{SAMP} = f_{CK_TIMER}$, N=8 0100：$f_{SAMP} = f_{DTS}/2$, N=6 0101：$f_{SAMP} = f_{DTS}/2$, N=8 0110：$f_{SAMP} = f_{DTS}/4$, N=6 0111：$f_{SAMP} = f_{DTS}/4$, N=8 1000：$f_{SAMP} = f_{DTS}/8$, N=6 1001：$f_{SAMP} = f_{DTS}/8$, N=8 1010：$f_{SAMP} = f_{DTS}/16$, N=5 1011：$f_{SAMP} = f_{DTS}/16$, N=6 1100：$f_{SAMP} = f_{DTS}/16$, N=8 1101：$f_{SAMP} = f_{DTS}/32$, N=5 1110：$f_{SAMP} = f_{DTS}/32$, N=6 1111：$f_{SAMP} = f_{DTS}/32$, N=8 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。

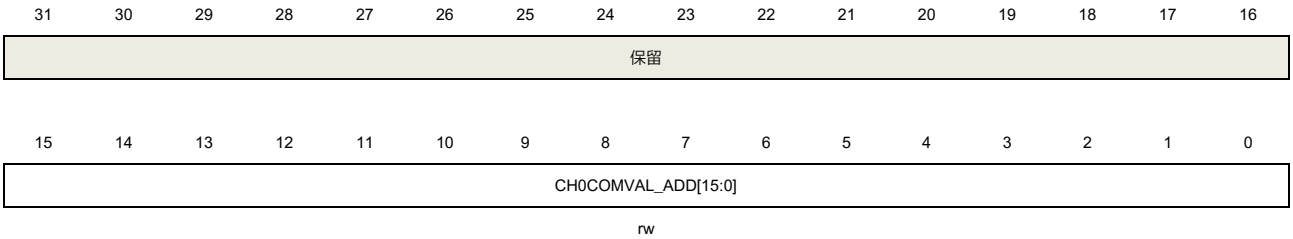
15	POEN	所有的通道输出使能 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHCTL2 寄存器的 CHxEN 位，CHxNEN 位)，则使能 CHx_O 和 CHx_ON 得输出。 0：禁止通道输出或强制为空闲状态 1：通道输出使能
14	OAEN	自动输出使能 此位定义了 POEN、CHPOENx(x=0…2)位是否可以被硬件自动置 1。 0：POEN、CHPOENx(x=0…2)位不能被硬件置 1 1：如果中止输入无效，下一次更新事件发生时，POEN、CHPOENx(x=0…2)位能被硬件自动置 1 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	BREAK 输入信号极性 此位定义了 BREAK 输入的极性。 0：BREAK 输入低电平有效 1：BREAK 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
12	BRKEN	BREAK 输入信号使能 此位置 1 使能 BREAK 输入信号。 0：BREAK 输入禁能 1：BREAK 输入使能 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下"关闭状态"使能 当 POEN&CHPOENx(x=0...2)位被置 1(运行模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 23-4. 由参数控制的互补输出表*。 0：输出"关闭状态"禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"禁能状态"。 1：输出"关闭状态"使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出"关闭状态"。 此位在 TIMERx_CCHP0 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下"关闭状态"使能 当 POEN&CHPOENx(x=0...2)位被清 0(空闲模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见*表 23-4. 由参数控制的互补输出表*。 0：输出"关闭状态"禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出"禁能状态"。 1：输出"关闭状态"使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出"关闭状态"。

		此位在 TIMERx_CCHP0 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00：禁能保护模式。无写保护。 01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP0 寄存器中 BRKEN/ BRKP/ OAEN/ DTCFG 位写保护。 10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/ CHxNP 位(如果相应通道配置为输出模式)，TIMERx_CCHP0 寄存器中 ROS/IOS 位。 11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0/ 1 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN 位(如果相关通道配置为输出模式)写保护。 系统复位后这两位只能被写一次，一旦 TIMERx_CCHP0 寄存器被写入，这两位被写保护。
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3'b0xx：DTvalue =DTCFG [7:0]x t_{DT}, $t_{DT}=t_{DTS}$. DTCFG [7:5] =3'b 10x：DTvalue =(64+DTCFG [5:0])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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

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。



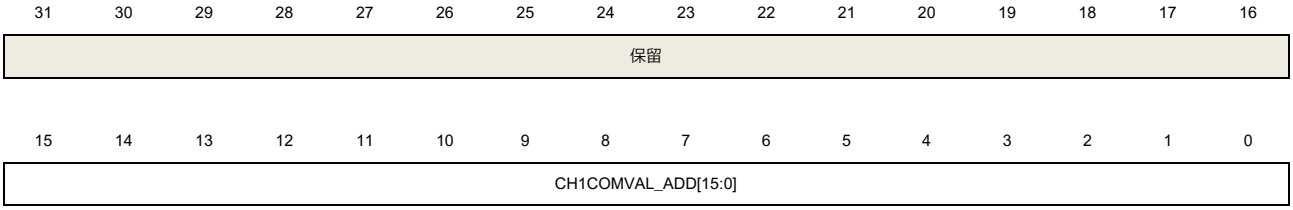
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。

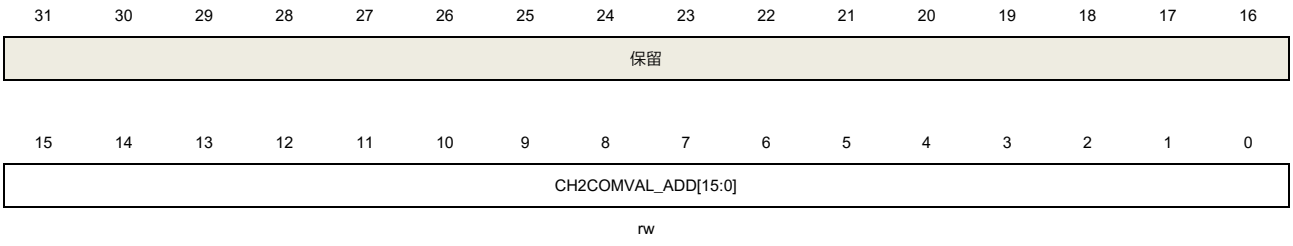


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 **注意**：该寄存器仅用于复合PWM模式(当CH1CPWMEN=1时)。

通道 2 附加比较寄存器(TIMERx_CH2COMV_ADD)

地址偏移：0x6C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2COMVAL_ADD [15:0]	通道2附加比较值 当通道2附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 **注意**：该寄存器仅用于复合PWM模式(当CH2CPWMEN=1时)。

通道 3 附加比较寄存器(TIMERx_CH3COMV_ADD)

地址偏移：0x70
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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CH3CPWMEN=1时)。

控制寄存器 2(TIMERx_CTL2)

地址偏移：0x74
复位值：0x0000 00FF

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3C PWMEN	CH2C PWMEN	CH1C PWMEN	CH0C PWMEN	保留											
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		BRKEN CH3	BRKEN CH2	BRKEN CH1	BRKEN CH0	DTIEN CH3	DTIEN CH2	DTIEN CH1	DTIEN CH0
rw		rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	CH3CPWMEN	通道 3 复合 PWM 模式使能 0：通道 3 复合 PWM 模式禁能 1：通道 3 复合 PWM 模式使能
30	CH2CPWMEN	通道 2 复合 PWM 模式使能 0：通道 2 复合 PWM 模式禁能 1：通道 2 复合 PWM 模式使能

29	CH1CPWMEN	通道 1 复合 PWM 模式使能 0：通道 1 复合 PWM 模式禁能 1：通道 1 复合 PWM 模式使能
28	CH0CPWMEN	通道 0 复合 PWM 模式使能 0：通道 0 复合 PWM 模式禁能 1：通道 0 复合 PWM 模式使能
27:16	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 CH3_ON 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 CH2_ON 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 CH1_ON 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 CH0_ON 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

备用功能控制寄存器 0(TIMERx_AFCTL0)

地址偏移：0x8C

复位值：0x0000 0001

该寄存器只能按字(32 位)访问。

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

保留	BRKCMP0P	保留	BRKIN0P
	rw		rw

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

保留	BRKCMP0EN	保留	BRKIN0EN
	rw		rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	BRKCMP0P	BREAK CMP0输入极性 0：CMP0输入信号不反相(BRKP =0，输入信号低有效；BRKP =1，输入信号高有效) 1：CMP输入信号反相(BRKP =0，输入信号高有效；BRKP =1，输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
24:17	保留	必须保持复位值。
16	BRKIN0P	BREAK 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BRKP位共同确定。 0：BRKIN0输入信号不反相(BRKP =0，输入信号低有效；BRKP =1，输入信号高有效) 1：BRKIN0输入信号反相(BRKP =0，输入信号高有效；BRKP =1，输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
15:10	保留	必须保持复位值。
9	BRKCMP0EN	BREAK CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
8:1	保留	必须保持复位值。
0	BRKIN0EN	BREAK BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

互补通道保护寄存器 1(TIMERx_CCHP1)

地址偏移：0x9C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

保留	DTDIFEN
	rw

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DTDIFEN	死区时间不同配置使能 0：上升沿和下降沿的死区时间(在 TIMERx_CCHP0 寄存器的 DTCFG[7:0]位域配置)相同。 1：上升沿的死区时间由 TIMERx_CCHP0 寄存器的 DTCFG[7:0]位域配置；下降沿的死区时间由 TIMERx_CCHP1 寄存器的 DTFCFG[7:0]位域配置。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15:8	Reserved	Must be kept at reset value.
7:0	DTFCFG[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。DTFCFG 值和死区时间的关系如下： DTFCFG [7:5] = 3'b0xx：DTvalue = DTFCFG [7:0] x t_{DT}, t_{DT} = t_{DTS} DTFCFG [7:5] = 3'b 10x：DTvalue =(64+DTFCFG [5:0])x t_{DT}，t_{DT} = t_{DTS} * 2 DTFCFG [7:5] = 3'b 110：DTvalue =(32+DTFCFG [4:0])x t_{DT}，t_{DT}= t_{DTS} * 8 DTFCFG [7:5] = 3'b 111：DTvalue =(32+DTFCFG [4:0])x t_{DT}，t_{DT} = t_{DTS} * 16 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。

计数器初值控制寄存器(TIMERx_CINITCTL)

地址偏移：0xA4
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													SWSYNCG	CINITDIR	CINITVEN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SWSYNCG	软件同步事件产生

		该位由软件置位，硬件自动清零，对该位进行读操作，读取值均为0。 0：无效 1：产生软件更新事件
1	CINITDIR	计数器初始计数方向 0：同步事件发生时，计数器向上计数 1：同步事件发生时，计数器向下计数 当同步事件发生，且计数器从TIMERx_CINITV寄存器加载计数器初始值后，该位表明了此时计数器的计数初始方向。 **注意**：该位仅用于CAM[1:0] != 00的情况。
0	CINITVEN	计数器初值寄存器使能 0：计数器初值寄存器禁能。计数器寄存器不从计数器初值寄存器加载初值。 1：计数器初值寄存器使能。当同步事件发生(通过软件将SWSYNCG位置1产生软件同步事件，输入引脚输入硬件同步事件)时，计数器寄存器可以从计数器初值寄存器加载初值。

计数器初值寄存器(TIMERx_CINITV)

地址偏移：0xA8
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINITVAL[15:0]	计数器初始值 该位域用于表示计数器的初始值。 CINITVEN位为0时，计数器的寄存器不能加载在CINITVAL[15:0]位域中设置的初始值。 CINITVEN位为1时，当同步事件发生时，计数器的寄存器可以加载在CINITVAL[15:0]位域中设置的初始值。

通道中止控制寄存器(TIMERx_CHBRKCTL)

地址偏移：0xAC
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									CHPOEN2	CHPOEN1	CHPOEN0	保留			
									rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH2BRKINEN	CH1BRKINEN	CH0BRKINEN	保留	CH2BRKIE	CH1BRKIE	CH0BRKIE	保留	CH2BRKINP	CH1BRKINP	CH0BRKINP	保留	CH2BRKMIE	CH1BRKMIE	CH0BRKMIE
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

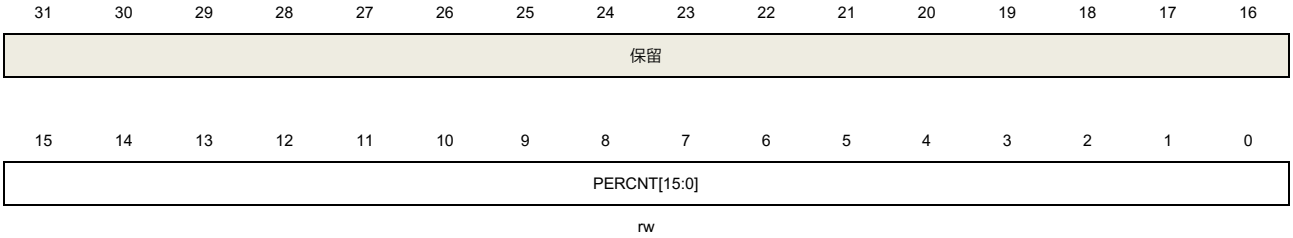
位/位域	名称	描述
31:23	保留	必须保持复位值。
22	CHPOEN2	通道输出使能 2 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 2 中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位(CH2EN 位，CH2NEN 位)，则使能 CH2_O 和 CH2_ON 输出。 0：禁止 CH2_O/CH2_ON 输出或强制为空闲状态 1：CH2_O/CH2_ON 输出使能
21	CHPOEN1	通道输出使能 1 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 1 中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位(CH1EN 位，CH1NEN 位)，则使能 CH1_O 和 CH1_ON 输出。 0：禁止 CH1_O/CH1_ON 输出或强制为空闲状态 1：CH1_O/CH1_ON 输出使能
20	CHPOEN0	通道输出使能 0 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 0 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(CH0EN 位，CH0NEN 位)，则使能 CH0_O 和 CH0_ON 输出。 0：禁止 CH0_O/CH0_ON 输出或强制为空闲状态 1：CH0_O/CH0_ON 输出使能
19:15	保留	必须保持复位值。
14	CH2BRKINEN	通道2中止输入信号使能 该位置1时，使能通道2中止输入信号。 0：通道2中止输入禁能 1：通道 2 中止输入使能
13	CH1BRKINEN	通道1中止输入信号使能 该位置1时，使能通道1中止输入信号。 0：通道1中止输入禁能 1：通道 1 中止输入使能

12	CH0BRKINEN	通道0中止输入信号使能 该位置1时，使能通道0中止输入信号。 0：通道0中止输入禁能 1：通道0中止输入使能
11	保留	必须保持复位值。
10	CH2BRKIE	通道2中止中断使能 0：禁能 1：使能
9	CH1BRKIE	通道1中止中断使能 0：禁能 1：使能
8	CH0BRKIE	通道0中止中断使能 0：禁能 1：使能
7	保留	必须保持复位值。
6	CH2BRKINP	通道2中止输入信号极性 该位用于配置通道2中止输入信号的极性 0：通道2中止输入信号不反相 1：通道 2 中止输入信号反相
5	CH1BRKINP	通道1中止输入信号极性 该位用于配置通道1中止输入信号的极性 0：通道1中止输入信号不反相 1：通道 1 中止输入信号反相
4	CH0BRKINP	通道0中止输入信号极性 该位用于配置通道0中止输入信号的极性 0：通道0中止输入信号不反相 1：通道 0 中止输入信号反相
3	保留	必须保持复位值。
2	CH2BRKMIE	通道2中止多周期中断使能 0：禁能 1：使能
1	CH1BRKMIE	通道1中止多周期中断使能 0：禁能 1：使能
0	CH0BRKMIE	通道0中止多周期中断使能 0：禁能 1：使能

通道中止周期寄存器(TIMERx_CHBRKPER)

地址偏移：0xB0
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

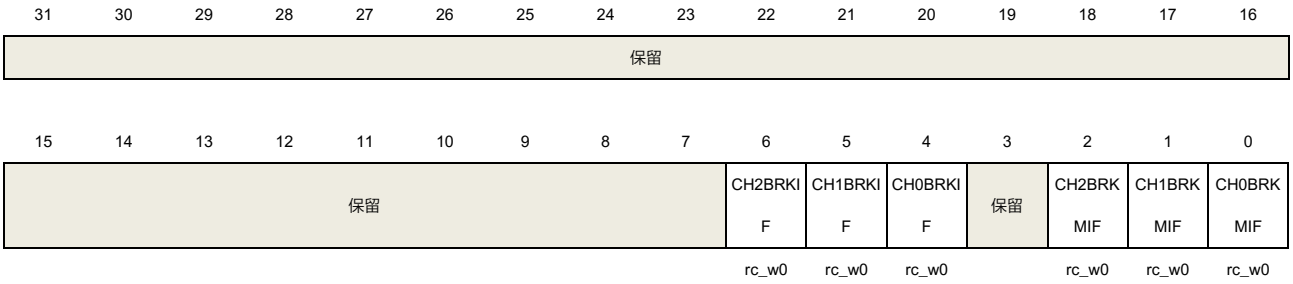


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PERCNT[15:0]	通道中止周期数 在边沿对齐模式下，该位域代表 PERCNT 个 PWM 周期。在中央对齐模式下，该位域代表 PERCNT/2 个 PWM 周期。

通道中止中断标志寄存器(TIMERx_CHBRKINTF)

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
31:7	保留	必须保持复位值
6	CH2BRKIF	通道 2 中止中断标志位 一旦通道 2 中止输入有效，由硬件对该位置'1'。如果通道 2 中止输入无效，则该位可由软件清'0'。 0：通道 2 中止输入未检测到有效电平 1：通道 2 中止输入检测到有效电平 注意：当 CH2BRKIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。
5	CH1BRKIF	通道 1 中止中断标志位

		一旦通道 1 中止输入有效，由硬件对该位置'1'。如果通道 1 中止输入无效，则该位可由软件清'0'。 0：通道 1 中止输入未检测到有效电平 1：通道 1 中止输入检测到有效电平 注意：当 CH1BRKIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。
4	CH0BRKIF	通道 0 中止中断标志位 一旦通道 0 中止输入有效，由硬件对该位置'1'。如果通道 0 中止输入无效，则该位可由软件清'0'。 0：通道 0 中止输入未检测到有效电平 1：通道 0 中止输入检测到有效电平 注意：当 CH0BRKIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。
3	保留	必须保持复位值
2	CH2BRKMIF	通道 2 中止多周期中断标志位 当通道2中止输入在多个连续周期内(由TIMERx_CHBRKPER寄存器决定)处于有效状态，硬件会该位置'1'，如果通道2中止输入不在有效状态，则软件可清除此标志。 0：通道2中止输入未检测到多个连续周期有效电平 1：通道2中止输入检测到多个连续周期有效电平 注意：当 CH2BRKMIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。
1	CH1BRKMIF	通道 1 中止多周期中断标志位 当通道1中止输入在多个连续周期内(由TIMERx_CHBRKPER寄存器决定)处于有效状态，硬件会该位置'1'，如果通道1中止输入不在有效状态，则软件可清除此标志。 0：通道1中止输入未检测到多个连续周期有效电平 1：通道1中止输入检测到多个连续周期有效电平 注意：当 CH1BRKMIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。
0	CH0BRKMIF	通道 0 中止多周期中断标志位 当通道0中止输入在多个连续周期内(由TIMERx_CHBRKPER寄存器决定)处于有效状态，硬件会该位置'1'，如果通道0中止输入不在有效状态，则软件可清除此标志。 0：通道0中止输入未检测到多个连续周期有效电平 1：通道0中止输入检测到多个连续周期有效电平 注意：当 CH0BRKMIE =1 和 BRKIE =1 时，如果该位被置 1，则会产生中断。

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'b111111：传输 63 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 … 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											OCPRECIVM	CCUSEL	保留	CHVSEL	OUTSEL
											rw	rw		rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值
4	OCPRECIVM	输出准备信号清除无效模式选择 这个位决定了硬件何时将输出准备信号清除设置为无效。 0：下个计数器上溢或者下溢事件 1：下个计数器更新事件
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	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1：如果 POEN 位与 IOS 位均为 0，则输出无效 0：无影响

23.2. 通用定时器 L0(TIMERx, x=1,2,3,4)

23.2.1. 简介

通用定时器 L0(TIMER1/2/3/4)是 4 通道定时器，支持输入捕获，输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L0 的计数器是 16 位或 32 位无符号计数器。

通用定时器 L0 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

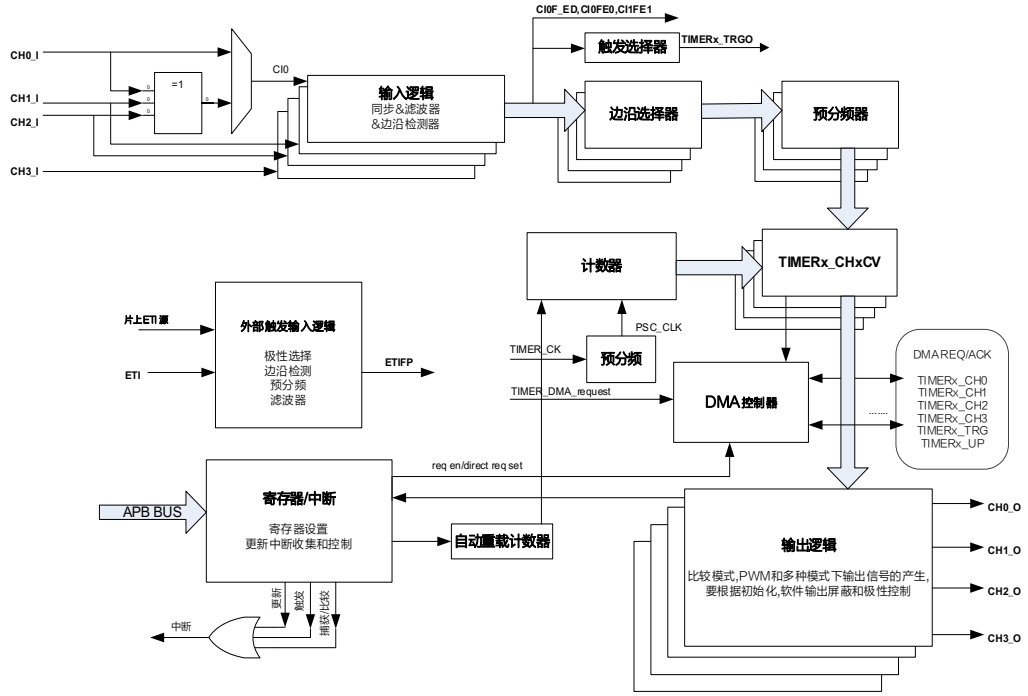
23.2.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位(TIMER2/3/4)和32位(TIMER1)；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

23.2.3. 结构框图

*图 23-48. 通用定时器 L0 结构框图*提供了通用定时器 L0 的内部细节

图 23-48. 通用定时器 L0 结构框图



23.2.4. 功能描述

时钟源选择

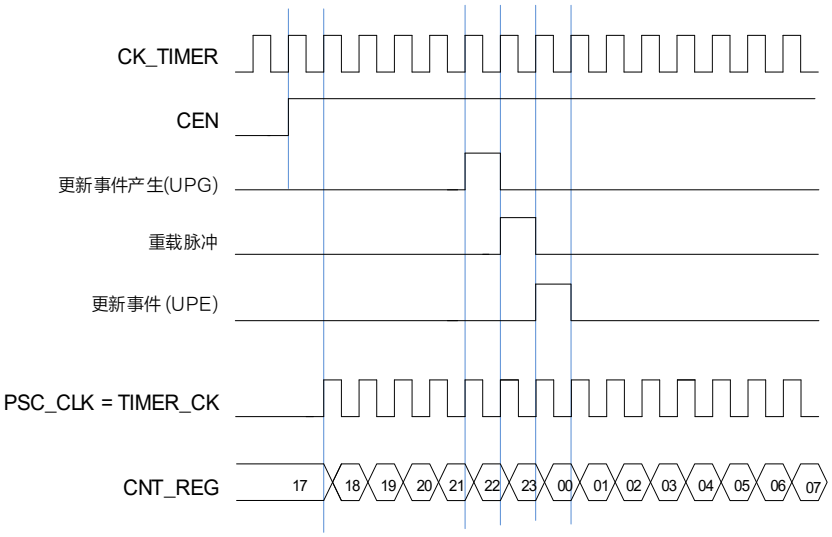
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] (y=0..6,9..15)位域控制的复用时钟源驱动。

- 当SYSCFG_TIMERxCFG(x=1..4)寄存器中的TSCFGy[4:0]=5'b00000(y= y=0..6,9..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)寄存器中的 TSCFGy[4:0] =5'b00000 (y=0..6,9..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)寄存器中的 TSCFGy[4:0] (y=0..2,6,9..14)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)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-49. 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~ITI3 的上升沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x1~0x4 来选择。

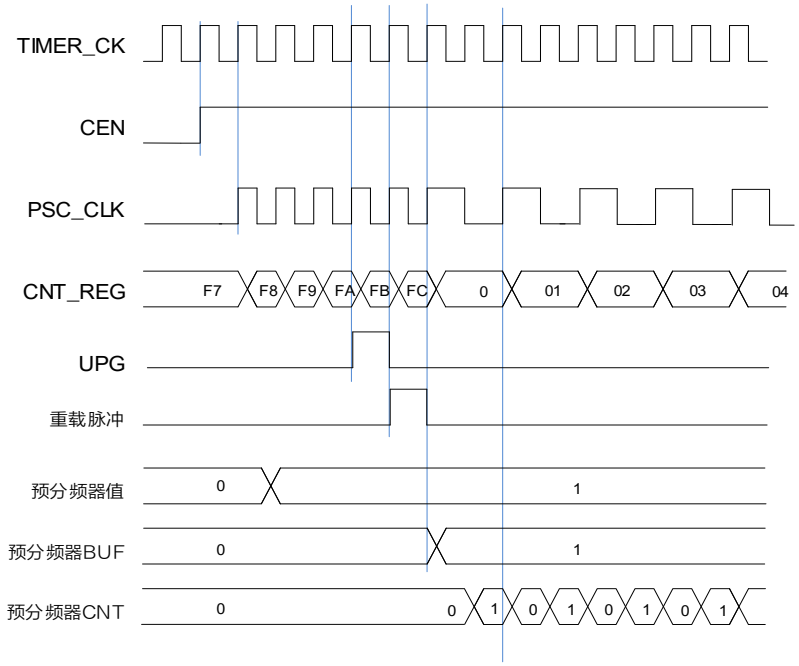
■ SMC1= 1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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。

时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 23-50. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

图 23-51. 向上计数时序图，PSC=0/2 和 *图 23-52. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 23-51. 向上计数时序图，PSC=0/2

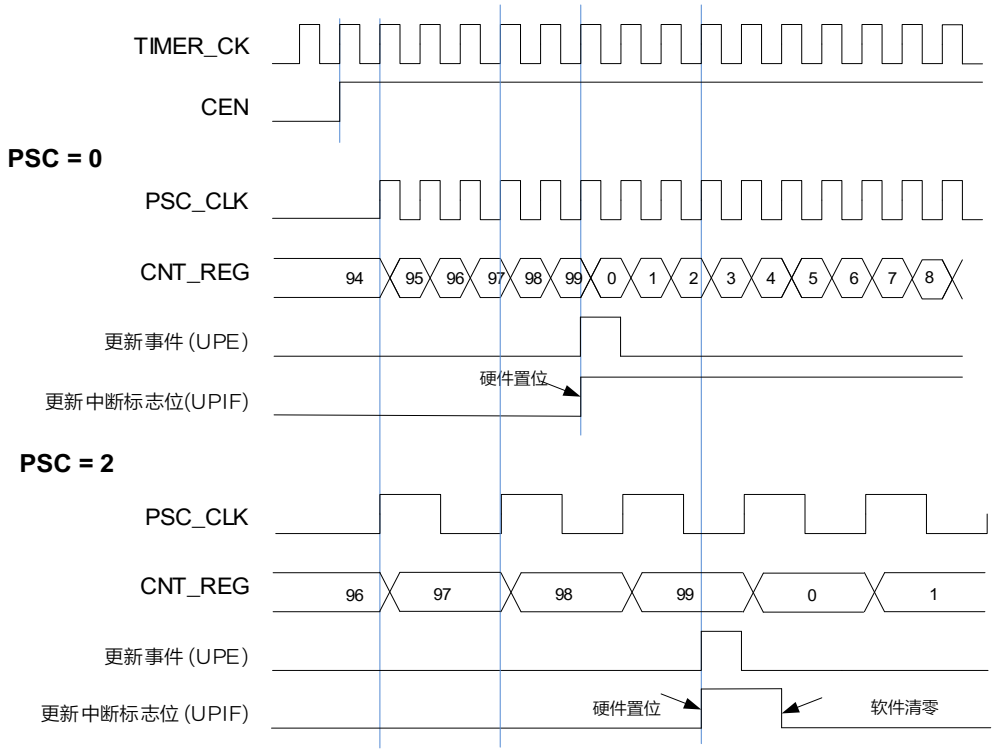
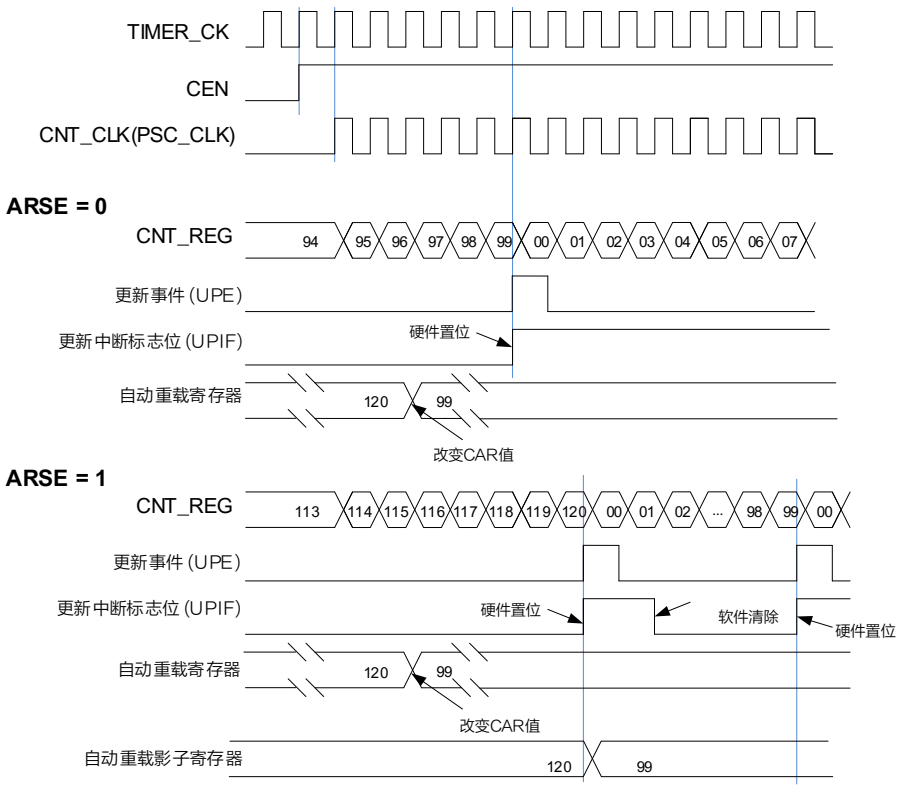


图 23-52. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值(定义在 TIMERx_CAR 寄存器中)向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

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

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

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

图 23-53. 向下计数时序图，PSC=0/2 和 *图 23-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 23-53. 向下计数时序图，PSC=0/2

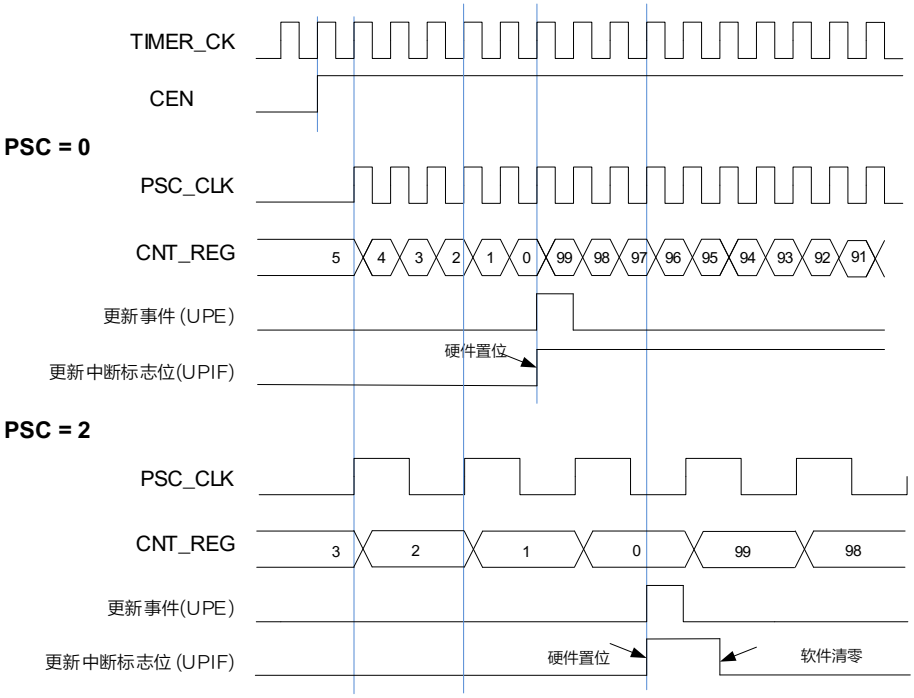
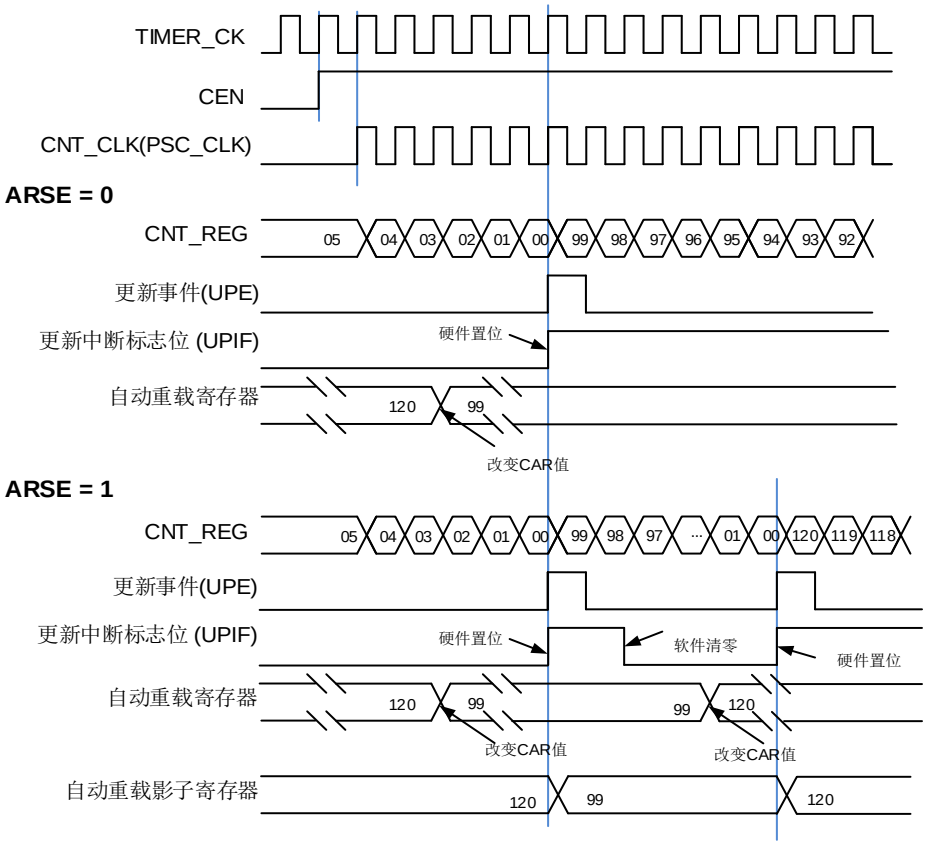


图 23-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



.

中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到(TIMERx_CAR-1)产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。计数方向被硬件自动更新。

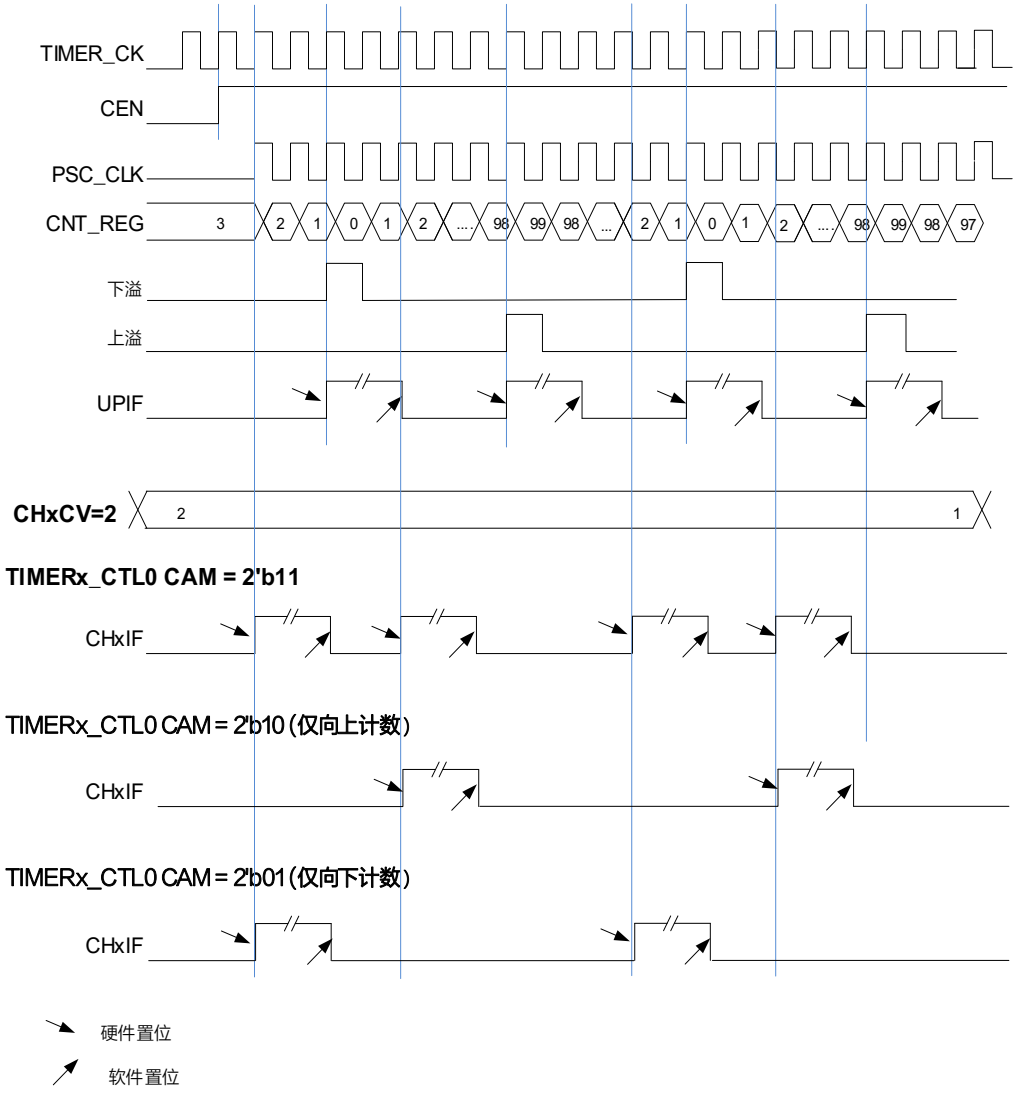
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 23-55. 中央计数模式计数器时序图*。如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

*图 23-55. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为

图 23-55. 中央计数模式计数器时序图



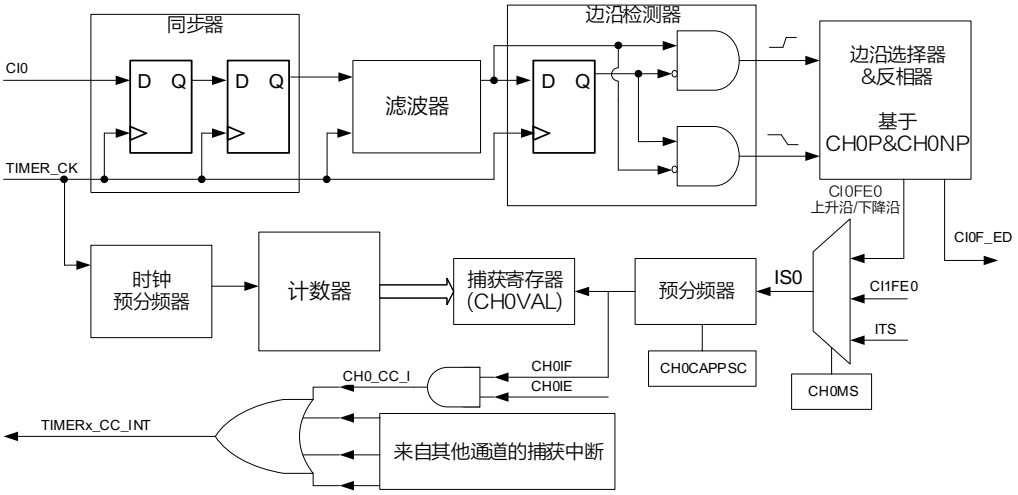
捕获/比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 23-56. 输入捕获逻辑



通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号(仅限于 CI0)。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置(TIMERx_CHCTL0 寄存器中 CHxCAPFLT)：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT。

第二步：边沿选择(TIMERx_CHCTL2 寄存器中 CHxP)：

配置 CHxP 选择上升沿或者下降沿。

第三步：捕获源选择(TIMERx_CHCTL0 寄存器中 CHxMS)：

一旦通过配置 CHxMS 选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且 TIMERx_CHxCV 寄存器不能再被写。

第四步：中断使能(TIMERx_DMAINTEN 寄存器中 CHxIE 和 CHxDEN)：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能(TIMERx_CHCTL2 寄存器中 CHxEN)。

结果：当期望的输入信号发生时，TIMERx_CHxCV 被设置成当前计数器的值，CHxIF 为置 1。如果 CHxIF 位已经为 1，则 CHxOF 位置 1。根据 TIMERx_DMAINTEN 寄存器中 CHxIE 和 CHxDEN 的配置，相应的中断和 DMA 请求会被提出。

直接产生：软件设置 CHxG 位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连

接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 输出比较模式

图 23-57. 输出比较逻辑(x=0,1,2,3)

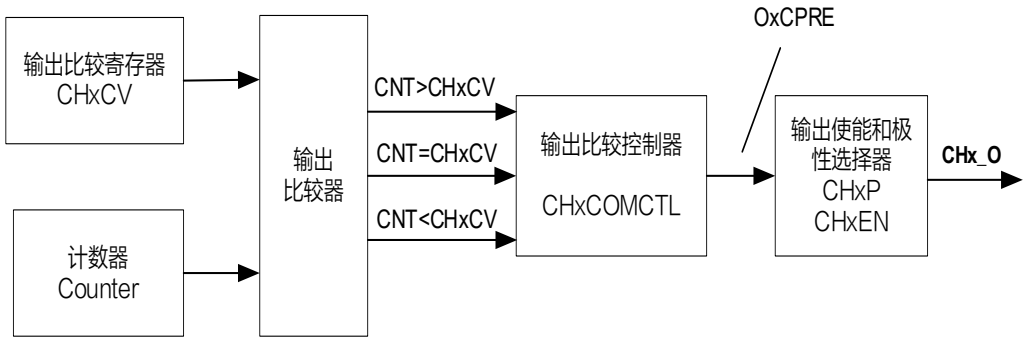


*图 23-57. 输出比较逻辑(x=0,1,2,3)*给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号(详情请见*通道输出参考信号*)的关系描述如下：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关(具体情况请见 TIMERx_CHCTL2 寄存器中的描述)。例如，当设置 CHxP=0(CHx_O 高电平有效，与 OxCPRE 输出极性相同)、CHxEN=1(CHx_O 输出使能)时：

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。

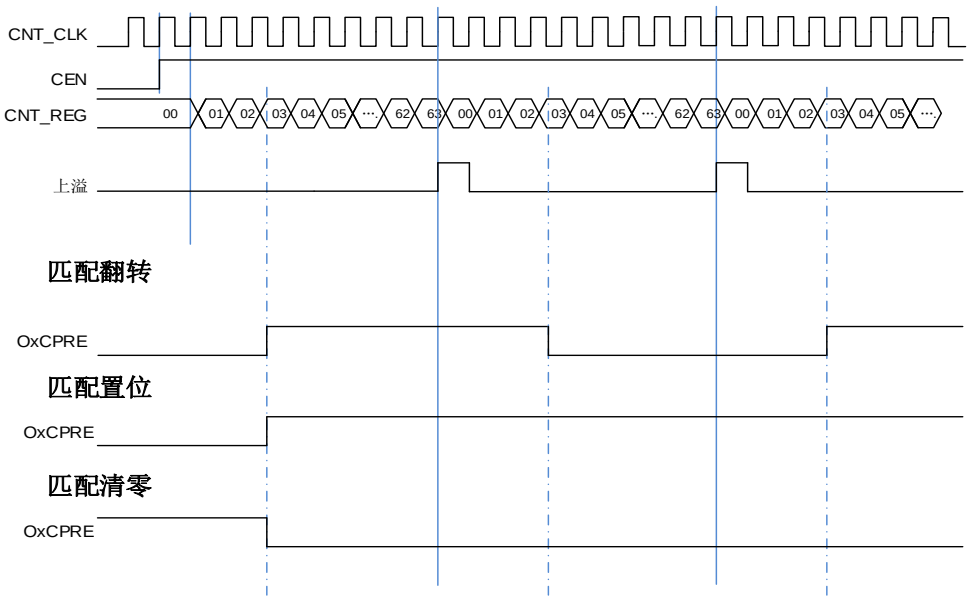
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 23-58. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 23-58. 三种输出比较模式



PWM 模式

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL 为 4'b0111),通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 23-59. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由(2*TIMERx_CAR 寄存器值)决定，占空比由(2*TIMERx_CHxCV 寄存器值)决定。*图 23-60. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下(CHxCOMCTL =4'b0110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平；PWM1 模式下(CHxCOMCTL =4'b0111)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的

值，通道输出一直为无效电平。

图 23-59. EAPWM 时序图

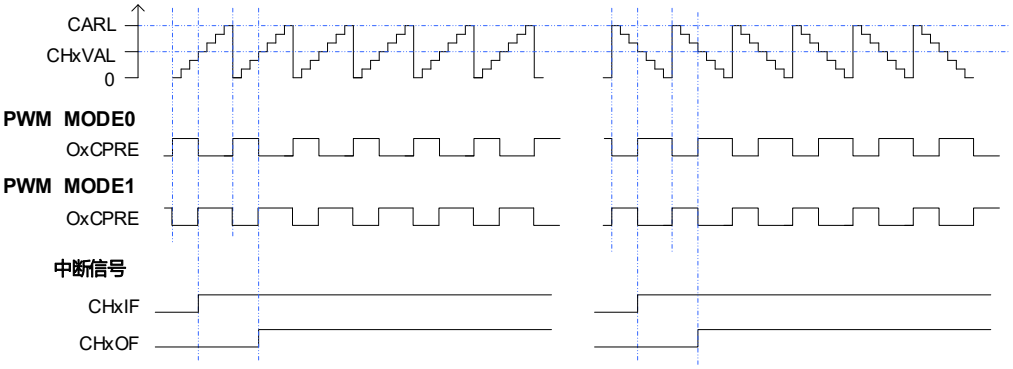
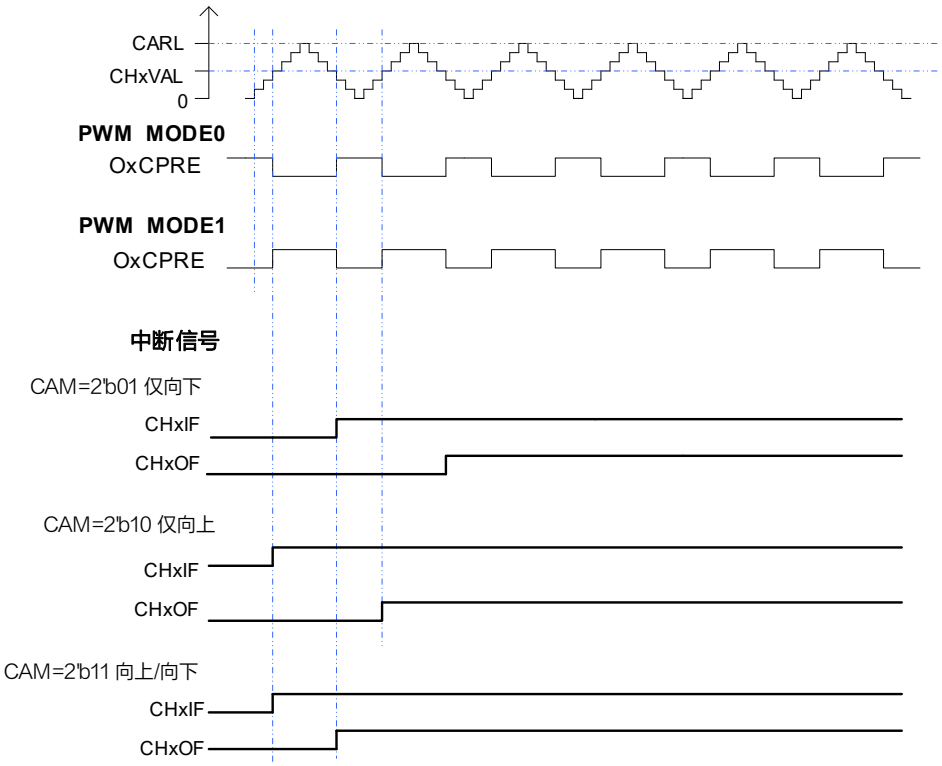


图 23-60. CAPWM 时序图



通道输出参考信号

根据*图 23-57. 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。

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, 1, 2, 13, 14) != 5'b00000 来选择使用哪种模式。计数器计数方向改变的机制如*表 23-6. 不同译码器模式下的计数方向*所示。其中，CI0FE0、CI1FE1 是经过滤波和极性选择后的 CI0、CI1 信号。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置 TIMERx_CAR 寄存器。

表 23-8. 不同译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式0 TSCFG0[4:0]!= 5'b00000	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 TSCFG1[4:0]!= 5'b00000	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 TSCFG2[4:0]!= 5'b00000	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上
正交译码器模式3 TSCFG13[4:0]!= 5'b00000	CI0FE1=1	向下	向上	-	-
	CI0FE1=0	-	-	-	-
正交译码器模式4 TSCFG14[4:0]!= 5'b00000	CI1FE0=1	-	-	向上	向下
	CI1FE0=0	-	-	-	-

注意： "-"意思是"无计数"； "X" 意思是不可能。"0"意思是低电平，"1"意思是高电平。

图 23-61. 译码器接口模式下计数器运行例子

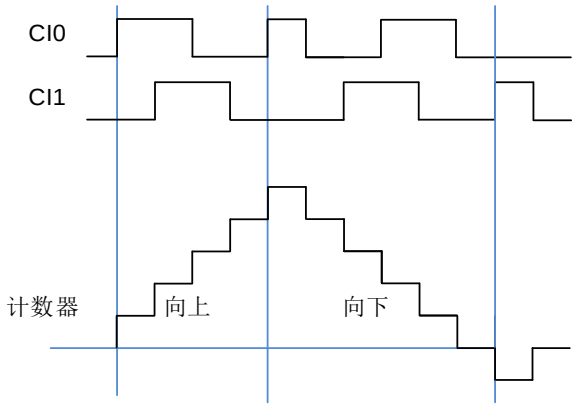
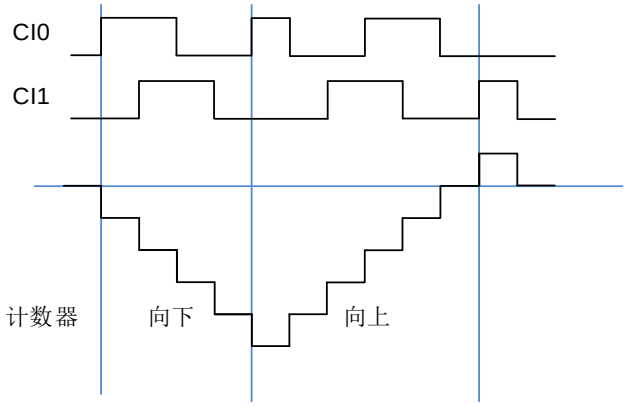


图 23-62. CI0FE0 极性反相的译码器接口模式下的例子



当正交译码器模式下计数器计数方向发生变化时，TIMERx_CTL0 寄存器中的 DIR 位发生改变。

正交译码器信号检测

支持两种正交译码器信号检测：信号跳变检测和断线检测。

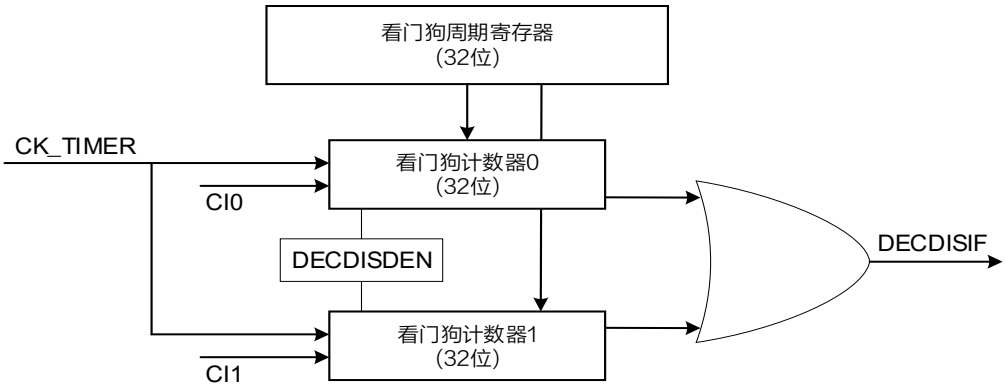
正交译码器信号跳变检测功能可用于检测两个正交译码器输入信号 CI0、CI1 的电平跳变沿(上升沿或下降沿)是否同时发生，可通过将 TIMERx_CTL2 寄存器中的 DECJDEN 位置 1 来使能。当 DECJDEN=1 时，若两个正交信号 CI0 和 CI1 的电平跳变同时发生，则中断标志位 DECJIF 置位。若 DECJIE=1，则相应的中断产生。

正交译码器信号断线检测功能可用于检测正交译码器输入信号 CI0、CI1 是否正常，可通过将 TIMERx_CTL2 寄存器中的 DECDISDEN 位置 1 来使能。正交译码器信号检测模块包括 2 个 32 位的看门狗计数器和 1 个周期寄存器，具体如*图 23-63. 正交译码器信号断线检测框图*所示，CI0FE0、CI1FE1 信号分别用于复位 2 个看门狗计数器。

当 DECDISDEN=1 时，2 个看门狗计数器同时开始向上计数，若看门狗计数器计数到看门狗

周期值(该值由 TIMERx_WDGPER 寄存器中的 WDGPER[31:0]位域确定)，则看门狗计数器计数超时，中断标志位 DECDISIF 置位。若 DECDISIE=1，则相应的中断产生。

图 23-63. 正交译码器信号断线检测框图



译码器

译码器功能有 4 种模式：译码器模式 0~3，通过设置 TSCFGy[4:0](y = 9, 10, 11, 12) != 5'b00000 来选择。这 4 种计数模式下的输入源有两个：CI0 和 CI1，其中，CI0FE0、CI1FE1 是经过滤波和极性选择后的 CI0、CI1 信号。

使用译码器模式 0/1 时，CI0FE0 作为计数方向信号，CI1FE1 作为计数脉冲。

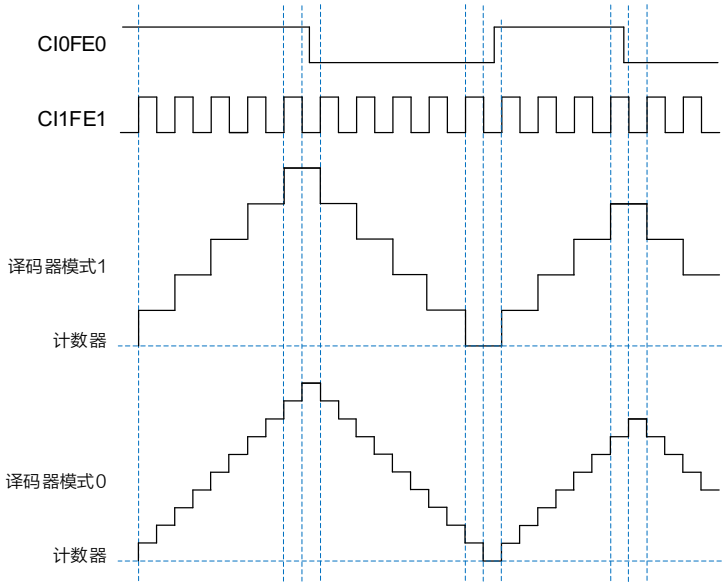
其中，CH0P 用于计数方向选择：当 CH0P=0 时，CI0FE0 为高电平时向上计数，CI0FE0 为低电平时向下计数；当 CH0P=1 时，CI0FE0 为高电平时向下计数，CI0FE0 为低电平时向上计数。

CH1P 用于选择 CI1FE1 信号的计数边沿：译码器模式 0 时，计数器在 CI1FE1 信号的上升沿和下降沿进行计数；译码器模式 1 时，当 CH1P=0 时，在 CI1FE1 信号的上升沿计数，当 CH1P=1 时，在 CI1FE1 信号的下降沿计数。更多译码器模式 1 的细节如*表 23-9. 译码器模式 1 的计数情况*和*图 23-64. 译码器模式 0/1 计数器运行实例(CH1P=0)*所示。

表 23-9. 译码器模式 1 的计数情况

CH1P	电平	计数器计数情况
0	CI0FE0为高电平	计数器在CI1FE1信号的上升沿向上计数
	CI0FE0为低电平	计数器在CI1FE1信号的上升沿向下计数
1	CI0FE0为高电平	计数器在CI1FE1信号的下降沿向上计数
	CI0FE0为低电平	计数器在CI1FE1信号的下降沿向下计数

图 23-64. 译码器模式 0/1 计数器运行实例(CH1P=0)



译码器模式 2/3 由 CI0FE0、CI1FE1 信号各自相互作用产生计数值，DIR 位被硬件自动改变。

译码器模式 2 时，计数器在 CI0FE0、CI1FE1 信号的上升沿和下降沿进行计数，计数方向由 CH0P 和 CH1P 确定；译码器模式 3，计数器在 CI0FE0、CI1FE1 信号的上升沿或下降沿进行计数，当 CHxP=0 时，信号为高电平时计数，或在信号的下降沿计数；当 CHxP=1 时，信号为低电平时计数，或在信号上升沿计数。具体情况请见*图 23-65. 译码器模式 2 / 3 计数器运行实例(CH0P / CH1P=0)*和*表 23-10. 译码器模式 2 / 3 下的计数方向*。

图 23-65. 译码器模式 2 / 3 计数器运行实例(CH0P / CH1P=0)

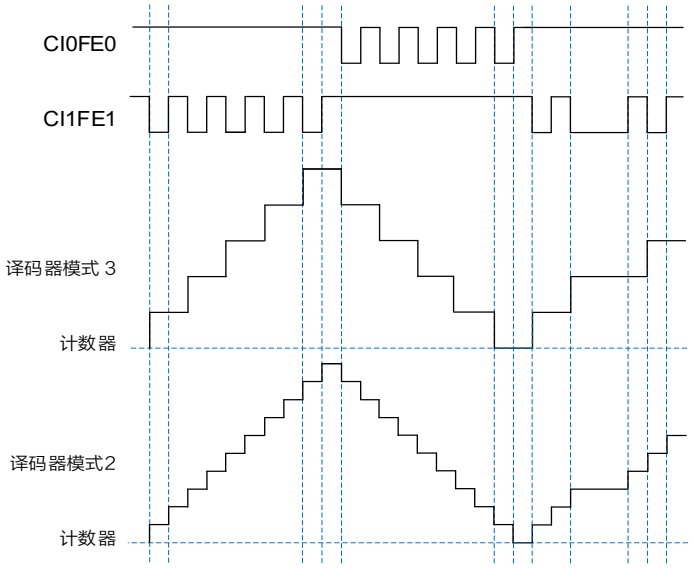


表 23-10. 译码器模式 2 / 3 下的计数方向

计数模式	极性	电平	CI0FE0		CI1FE1	
			上升	下降	上升	下降
译码器模式2	CHxP=0	CI1FE1=1	向下	向下	x	x

计数模式	极性	电平	CI0FE0		CI1FE1	
			上升	下降	上升	下降
TSCFG11[4:0]!= 5'b00000	(x = 0, 1)	CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	向上	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	向下	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	向上
译码器模式3 TSCFG12[4:0]!= 5'b00000	CHxP=0 (x = 0, 1)	CI1FE1=1	-	向下	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	-	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	-	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	-

当译码器模式下计数器计数方向发生变化时，TIMERx_CTL0 寄存器中的 DIR 位改变。

正交译码器和译码器的时钟输出

定时器可以通过 TRGO 输出译码器时钟输出信号。该功能仅用于正交译码器模式 0~4 和译码器模式 0~3，通过将 TIMERx_CTL1 寄存器中 MMC[3:0]位域设置为 4'b1000 来使能。

霍尔传感器接口功能

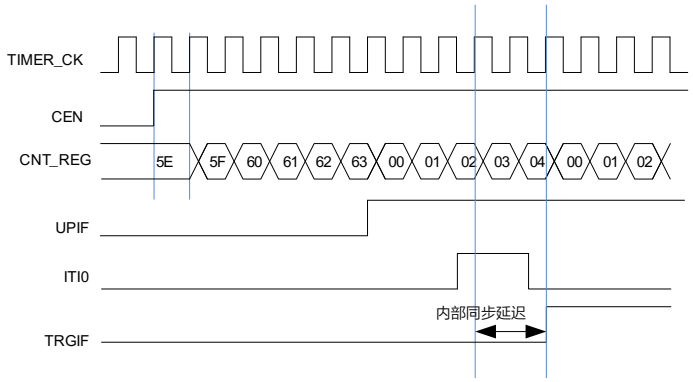
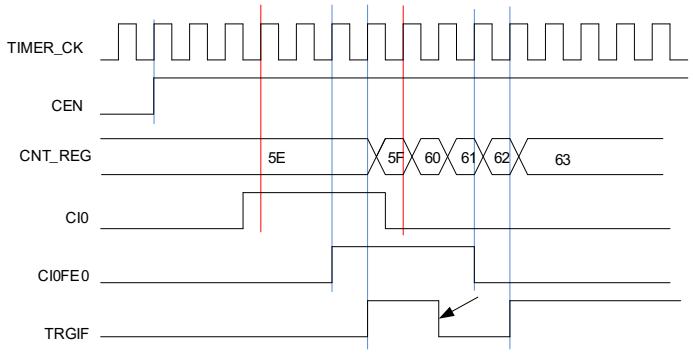
参考*高级定时器(TIMERx, x=0, 7, 19)霍尔传感器接口功能*。

主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式等，可以通过设置 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] (y=3..6)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6)位域值来选择。

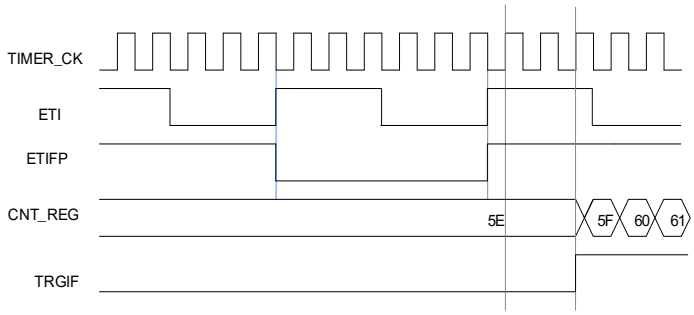
表 23-11. 从模式列表和举例(通用定时器 L0)

	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	TSCFGy[4:0] 00000：Mode disable 00001：ITI0 00010：ITI1 00011：ITI2 00100：ITI3	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	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，配置

<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>00101：CI0F_ED
00110：CI0FE0
00111：CI1FE1
01000：ETIFP(1)
01001: CI2FE2
01010: CI3FE3</td><td>入)，配置ETP选择极性和反相</td><td>ETFC设置滤波，配置ETPSC设置预分频</td></tr>
<tr><td>例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TSCFG3[4:0]
5'b00001，选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 23-66. 复位模式

</td></tr>
<tr><td>例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TSCFG4[4:0]
=5'b00110，
选择CI0FE0为触发源</td><td>TI0S=0(非异或)
[CH0NP=0， CH0P=0]
不反相，在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 23-67. 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 23-68. 事件模式



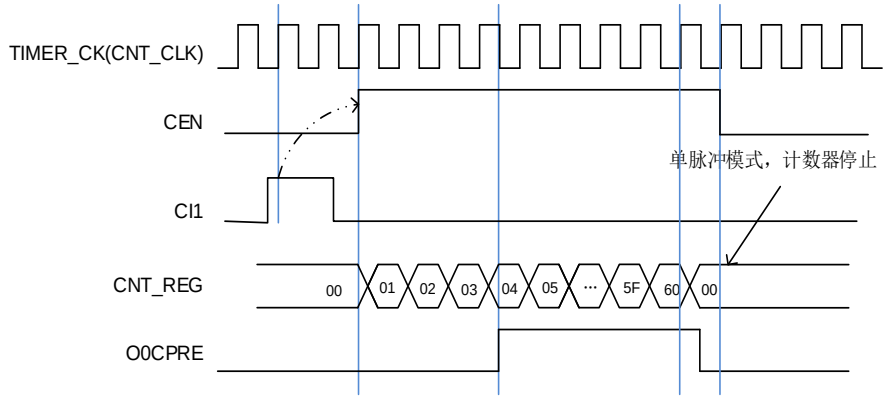
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。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 23-69. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考*高级定时器(TIMERx, x=0, 7, 19)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。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.2.5. TIMERx 寄存器(x=1/2/3/4)

TIMER1 基地址：0x4000 0000

TIMER2 基地址：0x4000 0400

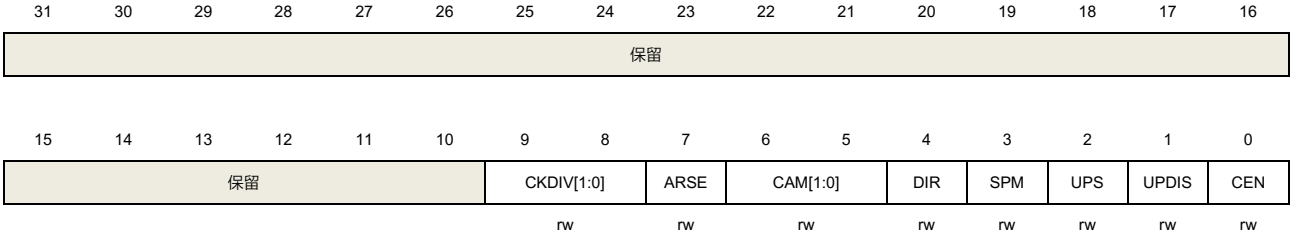
TIMER3 基地址：0x4000 0800

TIMER4 基地址：0x4000 0C00

控制寄存器 0(TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:10	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=3'b000)，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=3'b000)，只有在向上计数时，通道的比较中断标志置 1

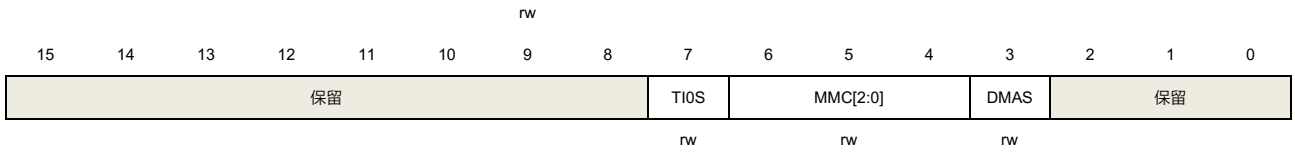
		11：中央对齐上下计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式(TIMERx_CHCTL0 寄存器中 CHxMS=3'b000)，在向上和向下计数时，通道的比较中断标志都会置 1 当计数器使能以后，该位不能从 0x00 切换到非 0x00 状态。
4	DIR	方向 0：向上计数 1：向下计数 当计数器配置为中央对齐模式或译码器模式时，该位为只读。
3	SPM	单脉冲模式 0：更新事件发生后，计数器继续计数 1：在下一次更新事件发生时，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0:使能后，下述任一事件产生更新中断或 DMA 请求： - 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						MMC[3]	保留								



位/位域	名称	描述
31:26	保留	必须保持复位值
25	MMC[3]	主模式控制 请参考 MMC[2:0]的描述。
24:8	保留	必须保持复位值
7	TI0S	通道 0 触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 0000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲，后一种情况下，TRGO 上的信号相对实际的复位会有一个延迟。 0001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。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 TRGO 上会有一个延迟，除非选择了主/ 从模式。 0010：更新。主模式控制器选择更新事件作为 TRGO。 0011：捕获/比较脉冲。通道 0 在发生一次捕获或一次比较成功时，主模式控制器产生一个 TRGO 脉冲 0100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO 0101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO 0110：比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO 0111：比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO 1000：译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出TRGO。此值仅用于正交译码器模式0~4和译码器模式0~3。 1001~1111：保留
3	DMAS	DMA 请求源选择 0：当通道捕获/比较事件发生时，发送通道 CHx 的 DMA 请求 1：当更新事件发生，发送通道 CHx 的 DMA 请求

2:0	保留	必须保持复位值

从模式配置寄存器(TIMERx_SMCFG)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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 TRGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:0	保留	必须保持复位值

DMA 和中断使能寄存器(TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														DECDISIE	DECJIE
														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
31:18	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
保留														DECDISIF	DECJIF
														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
31:18	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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGG	保留	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	保留	必须保持复位值.
4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述。
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述。
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0(TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	保留					CH1COMCTL[3]	保留							CH0COMCTL[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
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29:25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述。
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 000：通道 1 配置为输出 001：通道 1 配置为输入，IS1 映射在 CI1FE1 上 010：通道 1 配置为输入，IS1 映射在 CI0FE1 上 011：通道 1 配置为输入，IS1 映射在 ITS 上，此模式仅工作在内部触发器输入被选

		中时(由 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择)。 100~111：保留
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O0CPRE 参考信号被清 0。 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 CH0COMCTL[3]和 CH0COMCTL[2:0]位域定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O 的值。O0CPRE 高电平有效，而 CH0_O 的有效电平取决于 CH0P 位。 0000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 0001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 0010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 0011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 0100：强制为低。强制 O0CPRE 为低电平 0101：强制为高。强制 O0CPRE 为高电平 0110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。 0111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。 1000~1111：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，O0CPRE 电平才改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的

		最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为 3 个时钟周期
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时，CH0MS[2:0]位才可以写。 000：通道 0 配置为输出 001：通道 0 配置为输入，IS0 映射在 CI0FE0 上 010：通道 0 配置为输入，IS0 映射在 CI1FE0 上 011：通道 0 配置为输入，IS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时(由 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择)。 100~111：保留

输入捕获模式：

位/位域	名称	描述
31:16	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]	保留					CH3COMCTL[3]	保留							CH2COMCTL[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
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29:25	保留	必须保持复位值.
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 CH2COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 CH2COMCTL[2:0]描述

15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH3COMFEN	通道3输出比较快速使能 参见 CH2COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 000：通道 3 配置为输出 001：通道 3 配置为输入，IS3 映射在 CI3FE3 上 010：通道 3 配置为输入，IS3 映射在 CI2FE3 上 011：通道 3 配置为输入，IS3 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时(由 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择)。 100~111：保留
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIF 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O 的值。O2CPRE 高电平有效，而 CH2_O 的有效电平取决于 CH2P 位。 0000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 0001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 0010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。 0011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 0100：强制为低。强制 O2CPRE 为低电平 0101：强制为高。强制 O2CPRE 为高电平 0110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为无效电平，否则为有效电平。 0111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为有效电平，否则为无效电平。

		在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O2CPRE 电平才改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。
2	CH2COMFEN	通道2输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。 0：通道2输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为 3 个时钟周期
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。 000：通道 2 配置为输出 001：通道 2 配置为输入，IS2 映射在 CI2FE2 上 010：通道 2 配置为输入，IS2 映射在 CI3FE2 上 011：通道 2 配置为输入，IS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时(由 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择)。 100~111：保留

输入捕获模式：

位/位域	名称	描述
31:16	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

该寄存器可以按半字(16 位)或字(32 位)访问。

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

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

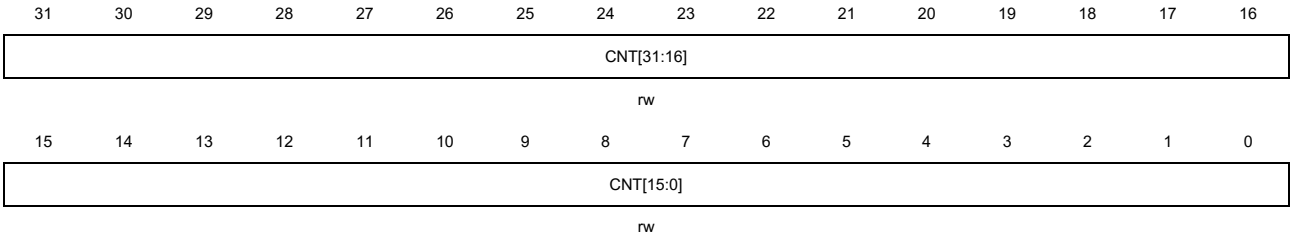
		参考 CH0NP 描述。
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述。
12	CH3EN	通道 3 使能 参考 CH0EN 描述。
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述。
10	保留	必须保持复位值
9	CH2P	通道 2 极性 参考 CH0P 描述。
8	CH2EN	通道 2 使能 参考 CH0EN 描述。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述。
6	保留	必须保持复位值
5	CH1P	通道 1 极性 参考 CH0P 描述。
4	CH1EN	通道 1 使能 参考 CH0EN 描述。
3	CH0NP	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)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

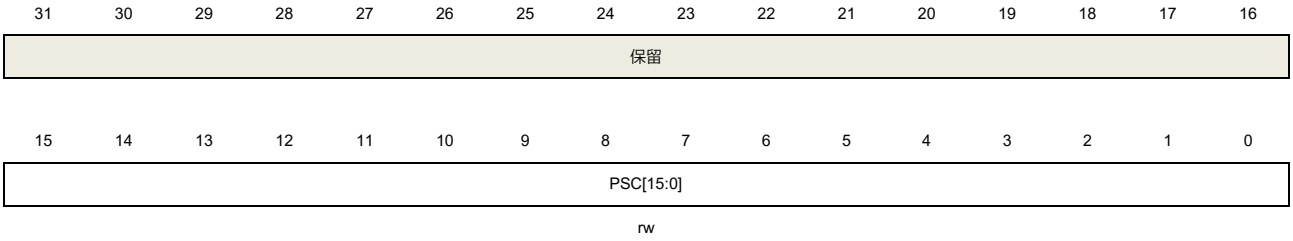


位/位域	名称	描述
31:16	CNT[31:16]	这些位是当前的计数值。写操作能改变计数器值。(bit 16 到 bit 31) 该位域仅用于 TIMER1。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器(TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值

15:0	PSC[15:0]	计数器时钟预分频值 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。
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。
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。
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。
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。
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。

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

位/位域	名称	描述

31:20	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
17:0	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'b111111：传输 63 次
7:6	保留	必须保持复位值
5:0	DMATA [5:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6'b000000：TIMERx_CTL0 6'b000001：TIMERx_CTL1 … 总之：起始地址 = TIMERx_CTL0 + DMATA*4。

DMA 发送缓冲区寄存器(TIMERx_DMATB)

地址偏移：0xE4
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

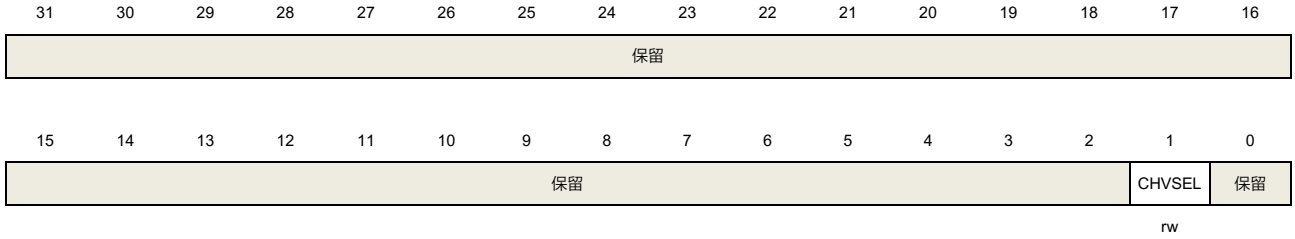
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

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

配置寄存器(TIMERx_CFG)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	保留	必须保持复位值

23.3. 通用定时器 L2(TIMERx, x=9,10,12,13)

23.3.1. 简介

通用定时器 L2 (TIMERx, x=9, 10, 12, 13)是单通道定时器，支持输入捕获和输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L2 含有一个 16 位无符号计数器。

通用定时器 L2 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

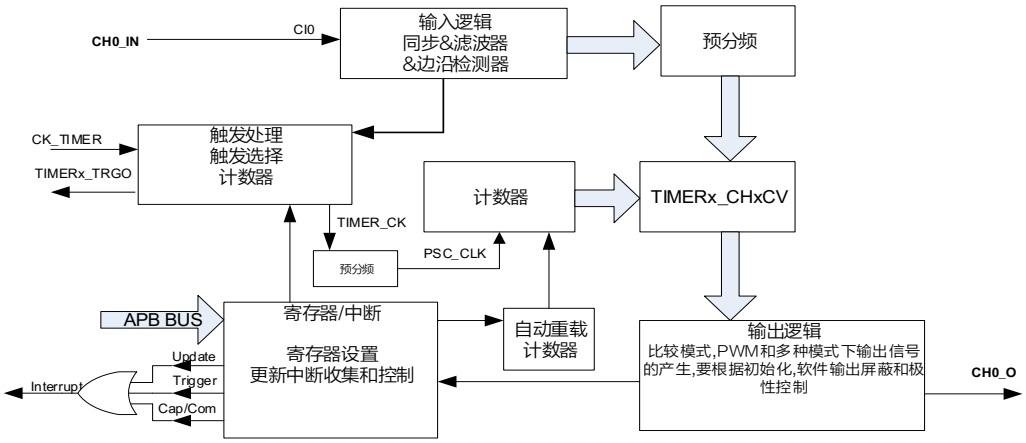
23.3.2. 主要特性

- 总通道数：1
- 计数器宽度：16位
- 时钟源：内部时钟
- 计数模式：向上计数，向下计数和中央计数
- 可编程的预分频器：16位，运行时可以被改变
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式
- 自动重装载功能.
- 中断输出：更新事件，比较/捕获事件

23.3.3. 结构框图

*图 23-70. 通用定时器 L2 结构框图*提供了通用定时器 L2 的内部配置细节。

图 23-70. 通用定时器 L2 结构框图



23.3.4. 功能描述

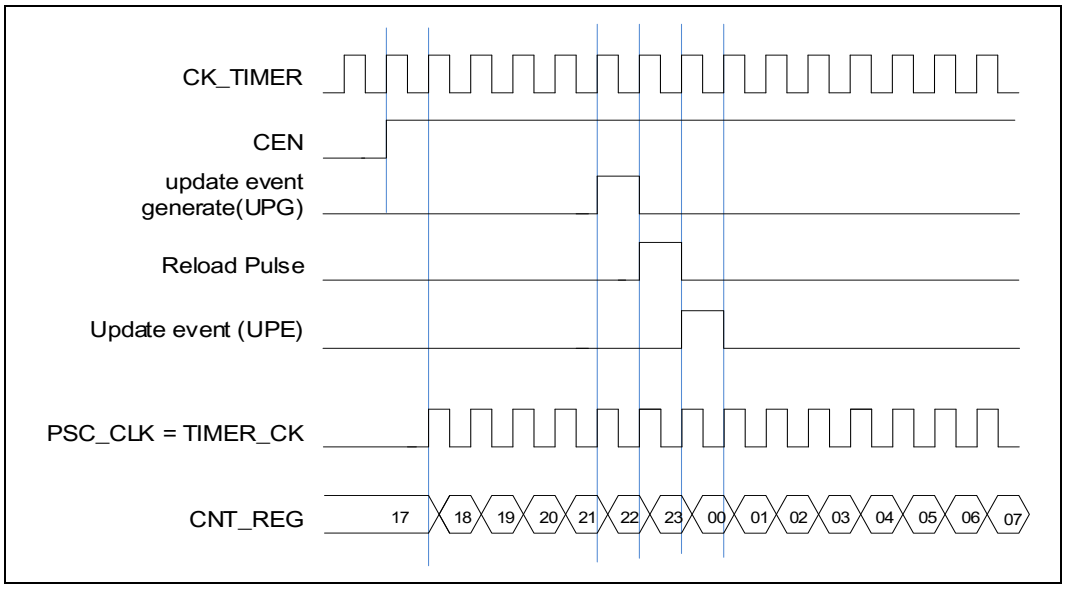
时钟源配置

通用定时器 L2 由内部时钟源 CK_TIMER 驱动

■ 定时器时钟TIMER_CK连接到RCU模块的CK_TIMER

通用定时器 L2 仅有一个时钟源 CK_TIMER，用来驱动计数器预分频器。当 CEN 置位，CK_TIMER 经过预分频器(预分频值由 TIMERx_PSC 寄存器确定)产生 PSC_CLK。

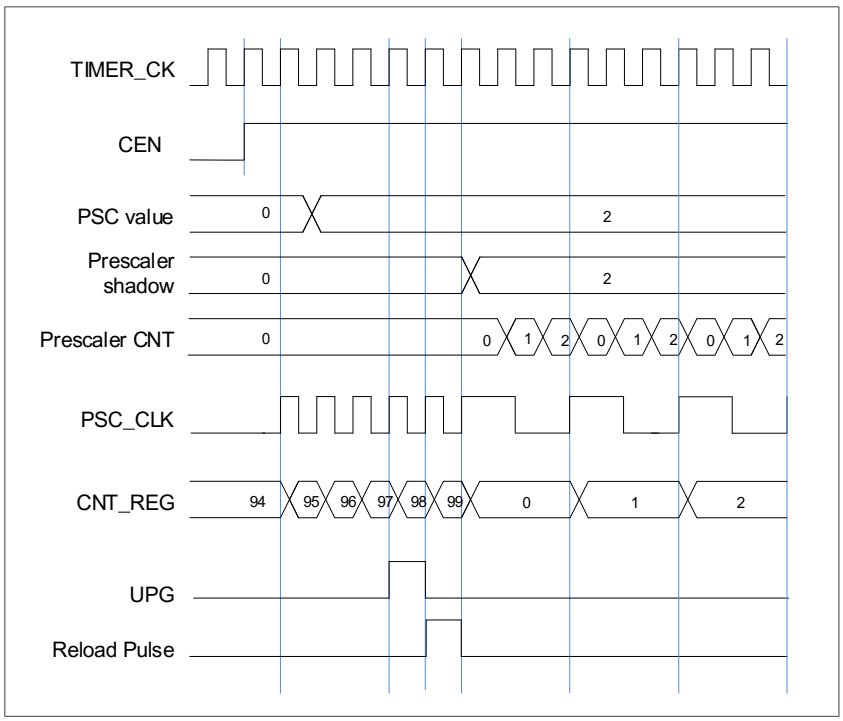
图 23-71. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 23-72. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

图 23-73. 向上计数时序图，PSC=0/2 和 *图 23-74. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 23-73. 向上计数时序图，PSC=0/2

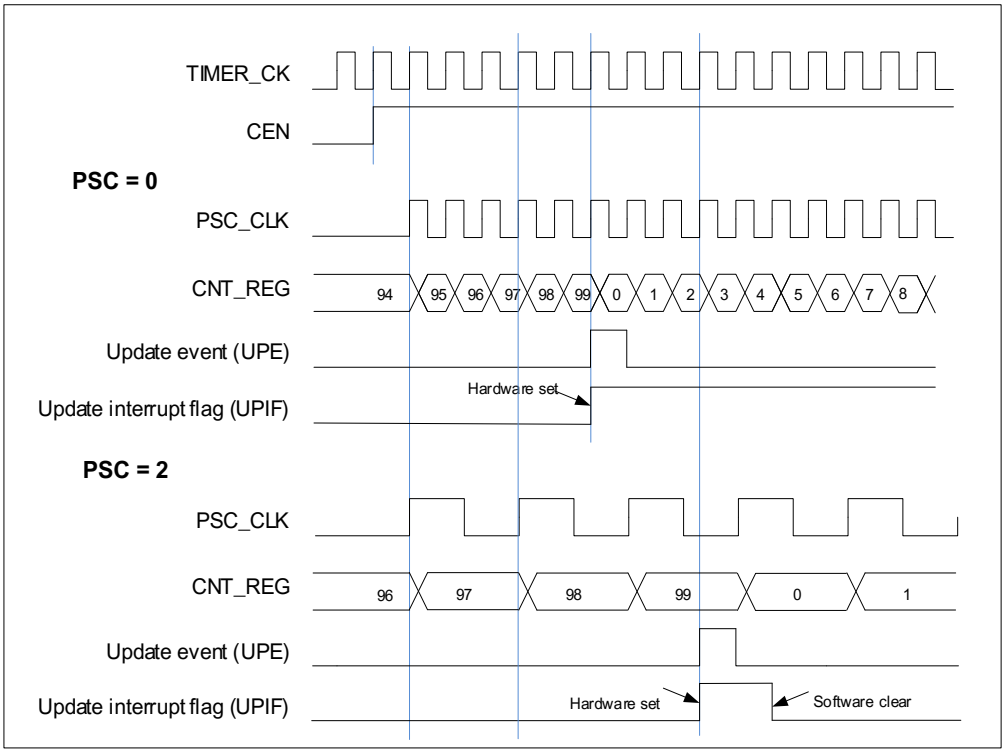
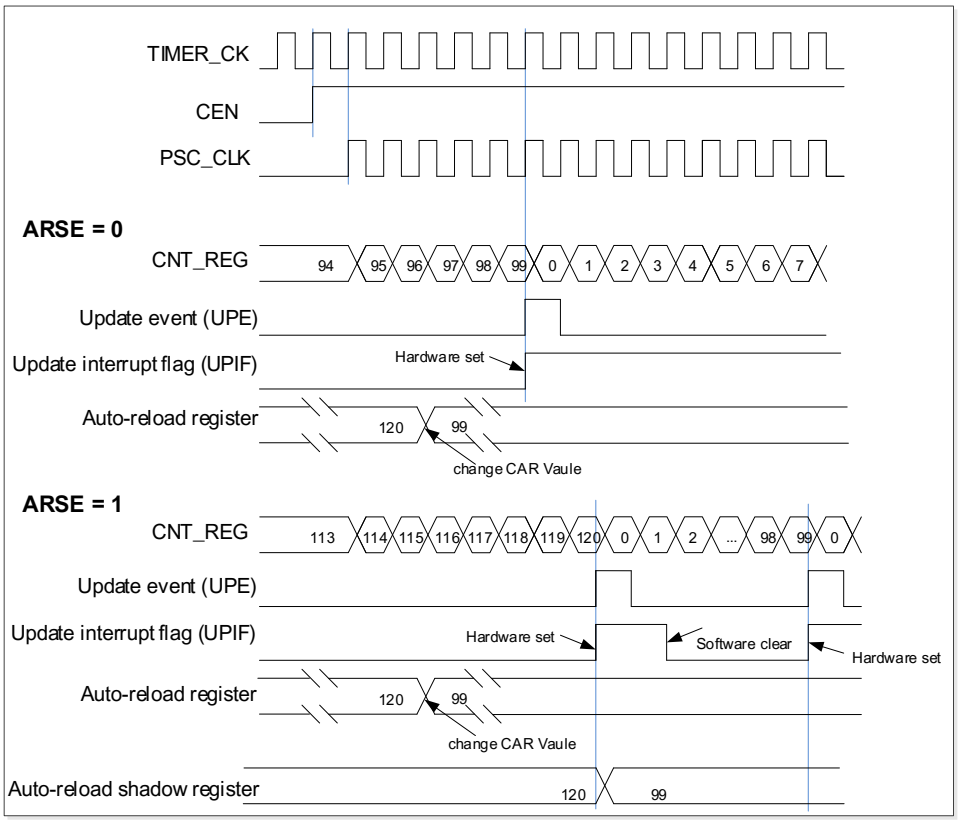


图 23-74. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



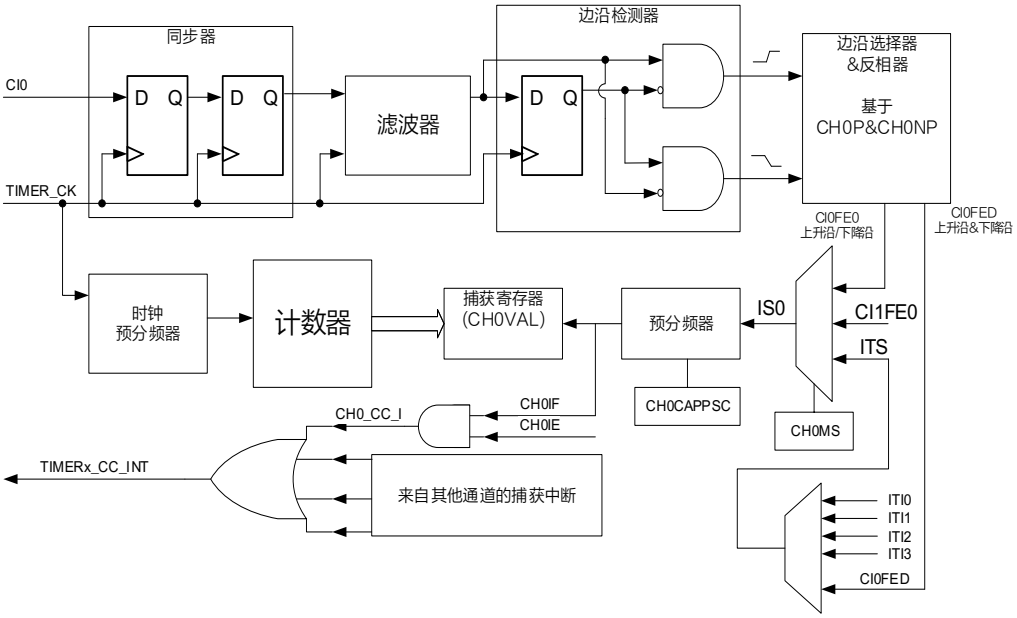
输入捕获和输出比较通道

通用定时器 L2 只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 23-75. 通道输入捕获原理



通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，TIMERx_CHxCV 存储计数器的值。

配置步骤如下：

第一步：滤波器配置(TIMERx_CHCTL0 寄存器中 CHxCAPFLT)：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT。

第二步：边沿选择(TIMERx_CHCTL2 寄存器中 CHxP/CHxNP)：

配置 CHxP/CHxNP 选择上升沿或者下降沿。

第三步：捕获源选择(TIMERx_CHCTL0 寄存器中 CHxMS)：

一旦通过配置 CHxMS 选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x0)，而且 TIMERx_CHxCV 寄存器不能再被写。

第四步：中断使能(TIMERx_DMAINTEN 寄存器中 CHxIE)：

使能相应中断，可以获得中断。

第五步：捕获使能(TIMERx_CHCTL2 寄存器中 CHxEN)。

结果：当期望的输入信号发生时，TIMERx_CHxCV 被设置成当前计数器的值，CHxIF 为置 1。如果 CHxIF 位已经为 1，则 CHxOF 位置 1。根据 TIMERx_DMAINTEN 寄存器中 CHxIE 的配置，相应的中断会被提出。

直接产生：软件设置 CHxG 位，会直接产生中断。

通道输入捕获功能也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 通道输出比较功能

图 23-76. 通道输出比较原理

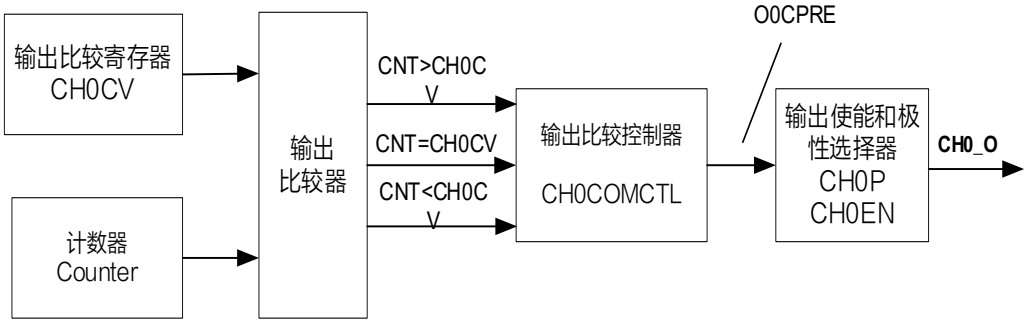


*图 23-76. 通道输出比较原理*给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号(详情请见*通道输出准备信号*)的关系描述：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关(具体情况请见 TIMERx_CHCTL2 寄存器中的描述)。例如，当设置 CHxP=0(CHx_O 高电平有效，与 OxCPRE 输出极性相同)、CHxEN=1(CHx_O 输出使能)时：

若 OxCPRE 输出有效(高)电平，则 CHx_O 输出有效(高)电平；

若 OxCPRE 输出无效(低)电平，则 CHx_O 输出无效(低)电平。

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 TIMERx_CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置 CHxCOMSEN 位来配置输出比较影子寄存器；

设置 CHxCOMCTL 位来配置输出模式(置高电平/置低电平/反转)；

设置 CHxP/CHxNP 位来选择有效电平的极性；

设置 CHxEN 使能输出。

第三步：通过 CHxIE 位配置中断使能。

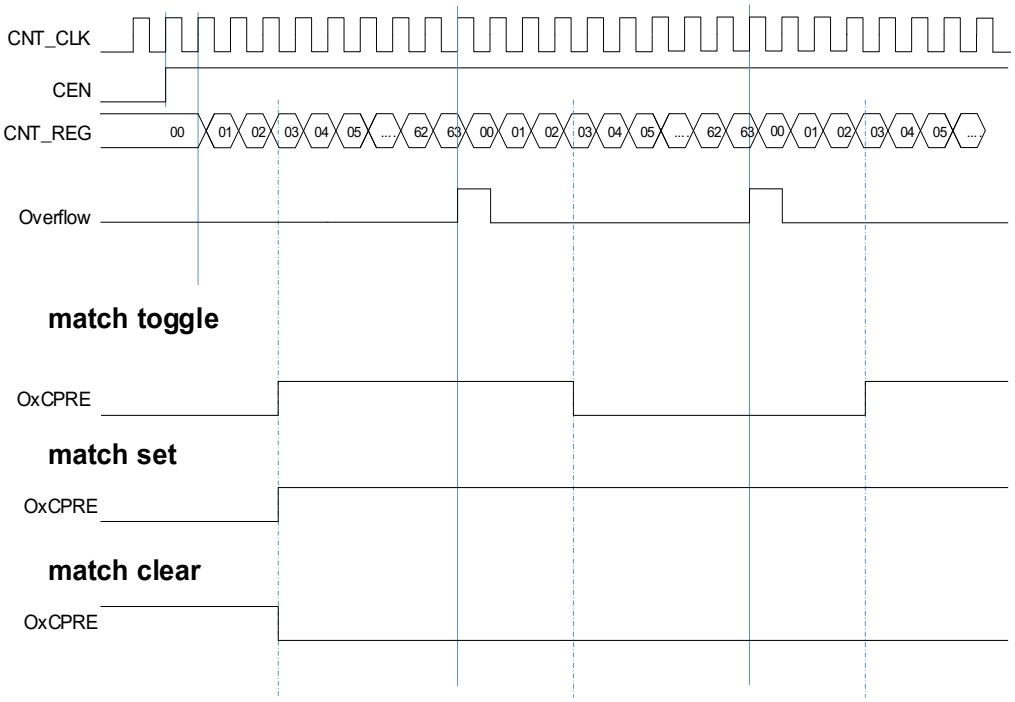
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

TIMERx_CHxCV 可以在运行时根据你所期望的波形而改变。

第五步：设置 CEN 位使能定时器。

*图 23-77. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 23-77. 三种输出比较模式



通道输出准备信号

根据*图 23-76. 通道输出比较原理*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号(通道 x 输出准备信号)。设置 CHxCOMCTL 位可以

定义 OxCPRE 信号类型。当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 输出准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域为 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL=0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.3.5. TIMERx 寄存器(x=9,10,12,13)

TIMER9 基地址：0x4001 5000

TIMER10 基地址：0x4001 5400

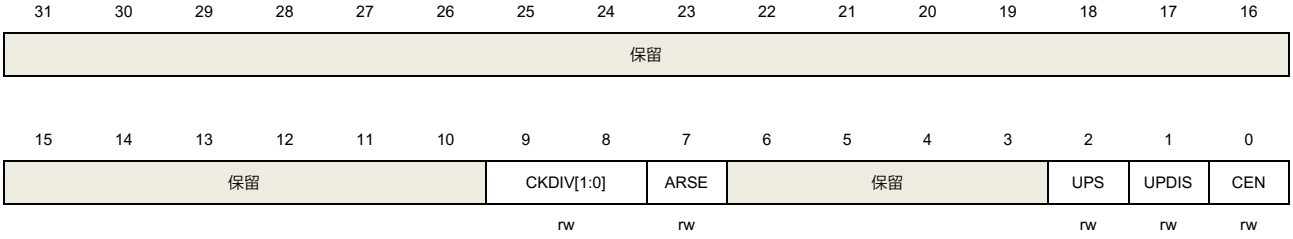
TIMER12 基地址：0x4000 1C00

TIMER13 基地址：0x4000 2000

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:3	保留	必须保持复位值
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新

		1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

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

		UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：保留 110：保留 111：保留
3:0	保留	必须保持复位值

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0IE	UPIE
														rw	rw

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

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0OF	保留							CH0IF	UPIF

rc_w0 rc_w0 rc_w0

位/位域	名称	描述
31:10	保留	必须保持复位值
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:2	保留	必须保持复位值
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0G	UPG
														w	w

位/位域	名称	描述
31:2	保留	必须保持复位值.
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。

0：不产生通道 0 捕获或比较事件
1：发生通道 0 捕获或比较事件

0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
									rw			rw		rw	

输出比较模式：

位/位域	名称	描述
31:7	保留	必须保持复位值.
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。

111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00(比较模式)时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：保留 11：保留

输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值.
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4'b0000	无滤波器	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	

		4'b0100	6	$f_{DTS}/2$
		4'b0101	8	
		4'b0110	6	$f_{DTS}/4$
		4'b0111	8	
		4'b1000	6	$f_{DTS}/8$
		4'b1001	8	
		4'b1010	5	
		4'b1011	6	$f_{DTS}/16$
		4'b1100	8	
		4'b1101	5	
		4'b1110	6	$f_{DTS}/32$
		4'b1111	8	

3:2	CH0CAPPSC[1:0]	通道 0 输入捕获预分频器 这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH0MS[1:0]	通道 0 模式选择 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CH0NP	保留	CH0P	CH0EN
												rw		rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值.
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选

		择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值.
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：保留。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改保留。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移： 0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

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

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

配置寄存器 (TIMERx_CFG)

地址偏移：0xFC

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	保留	必须保持复位值

23.4. 通用定时器 L3(TIMERx, x=14,40,41,42,43,44)

23.4.1. 简介

通用定时器 L3(TIMER14/40/41/42/43/44)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。

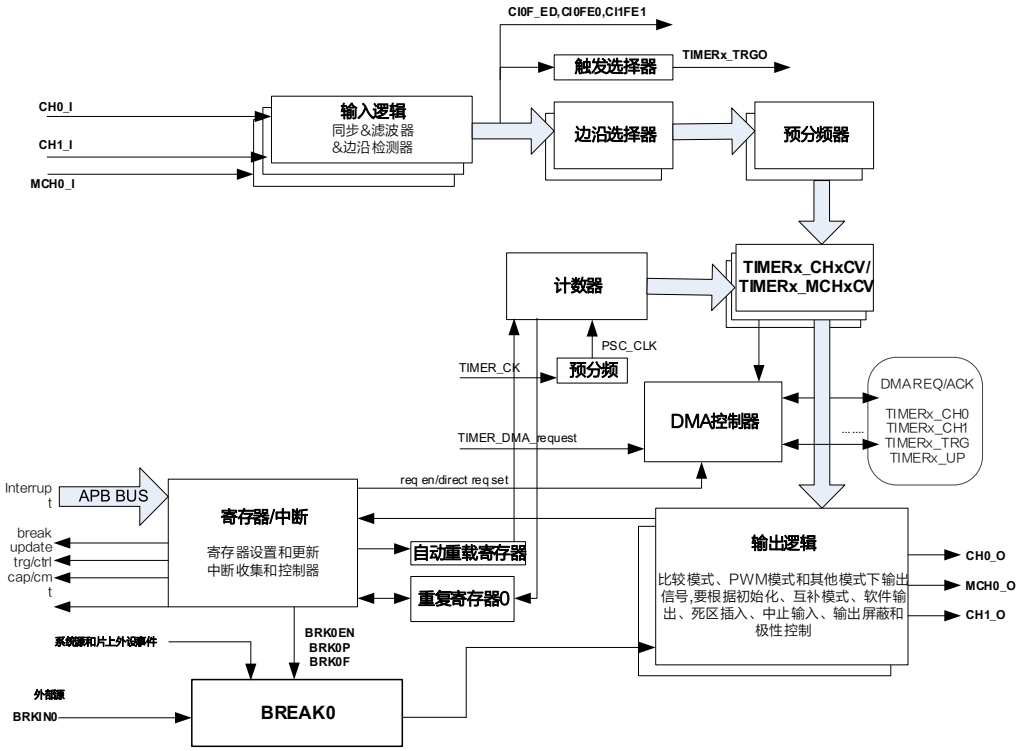
23.4.2. 主要特性

- 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；
- 多种计数模式：向上计数，向下计数和中央计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 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：BREAK；
- 中断输出和DMA请求：更新事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

23.4.3. 结构框图

*图 23-78. 通用定时器 L3 结构框图*提供了通用定时器 L3 的内部配置细节

图 23-78. 通用定时器 L3 结构框图



23.4.4. 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..6,15)位域。

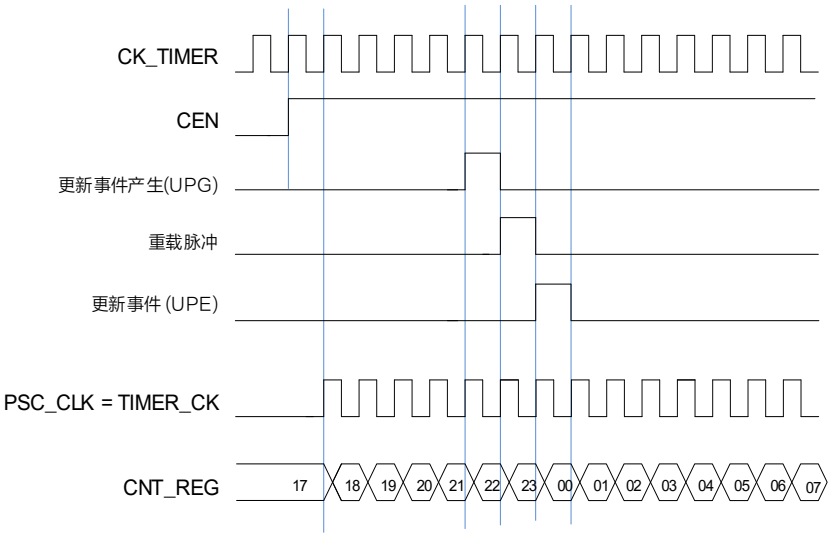
- 当 SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) 寄存器中的 TSCFGy[4:0]=5'b00000(y=3..6,15)，定时器选择内部时钟源(连接到 RCU 模块的 CK_TIMER)

如果 SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) 寄存器中的 TSCFGy[4:0] =5'b00000 (y=3..6,15)，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)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 23-79. 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

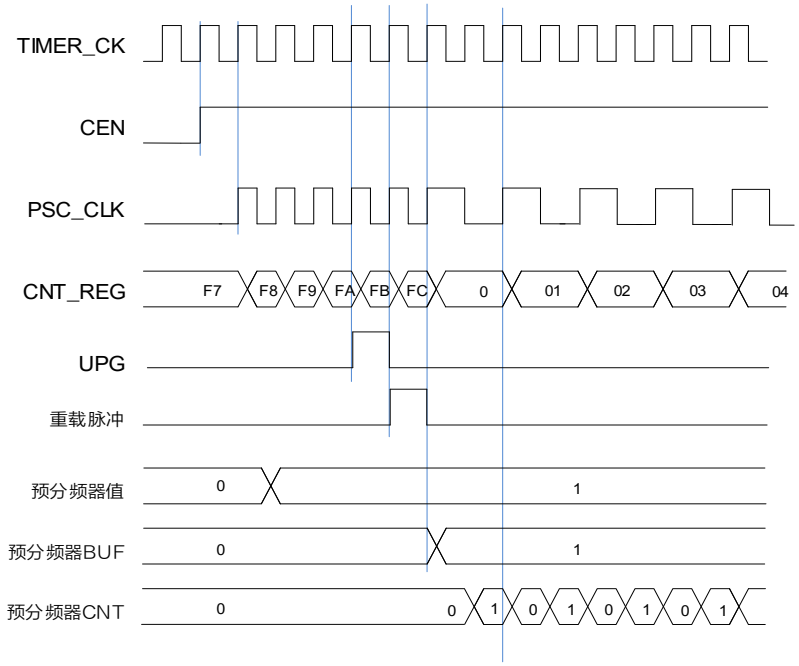
计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 TSCFG6[4:0]为 0x5~0x7 和 0xB。

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 23-80. 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)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

图 23-81. 向上计数时序图，PSC=0/2 和 *图 23-82. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 23-81. 向上计数时序图，PSC=0/2

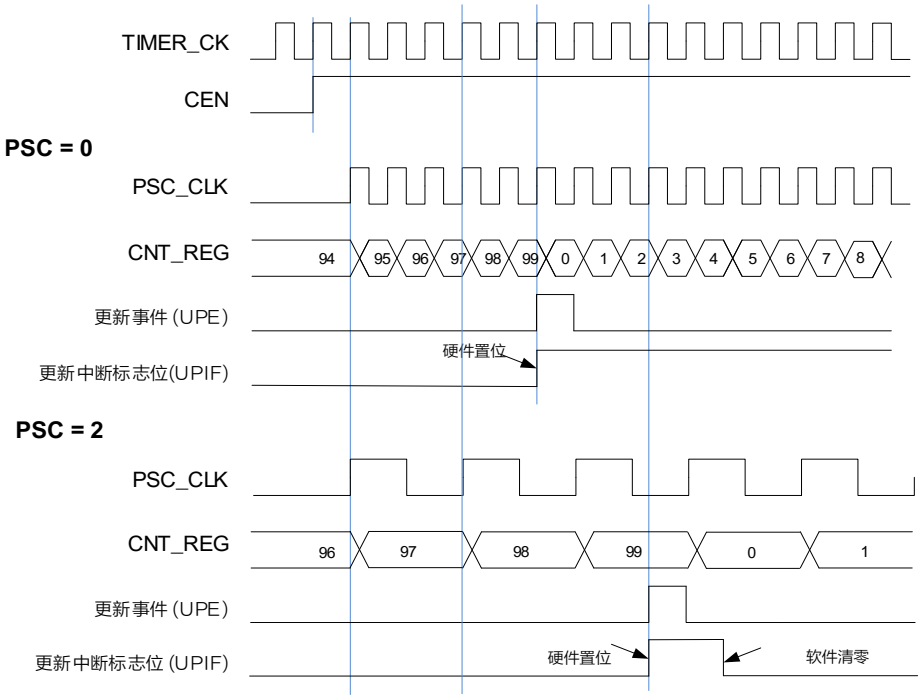
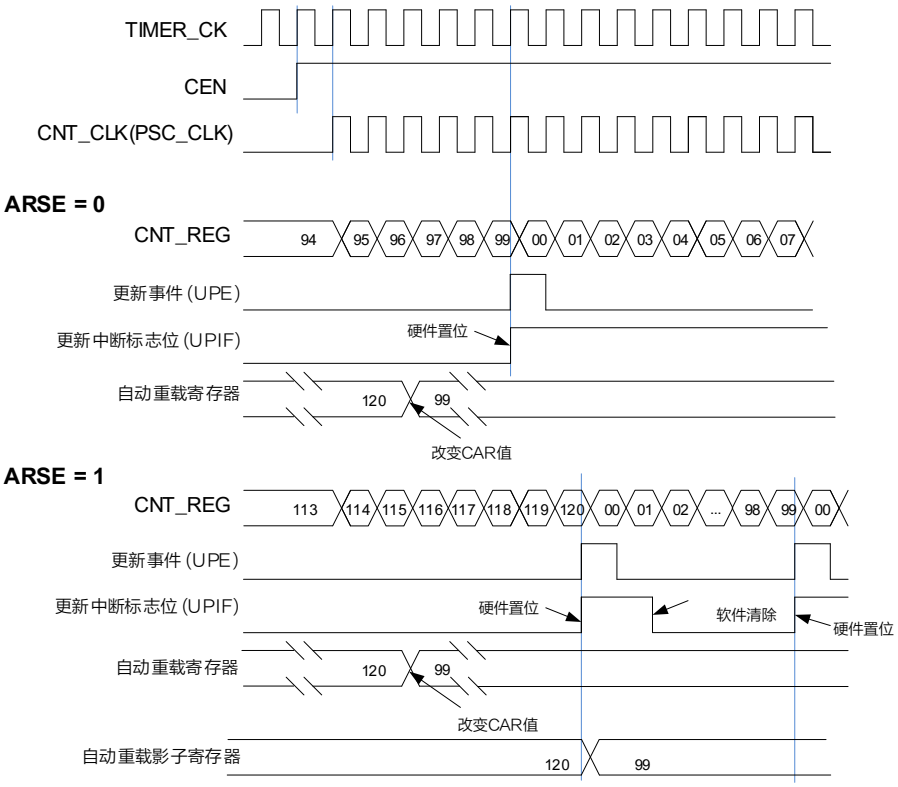


图 23-82. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值(定义在 TIMERx_CAR 寄存器中)向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

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

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

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

图 23-53. 向下计数时序图，PSC=0/2 和 *图 23-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 23-83. 向下计数时序图，PSC=0/2

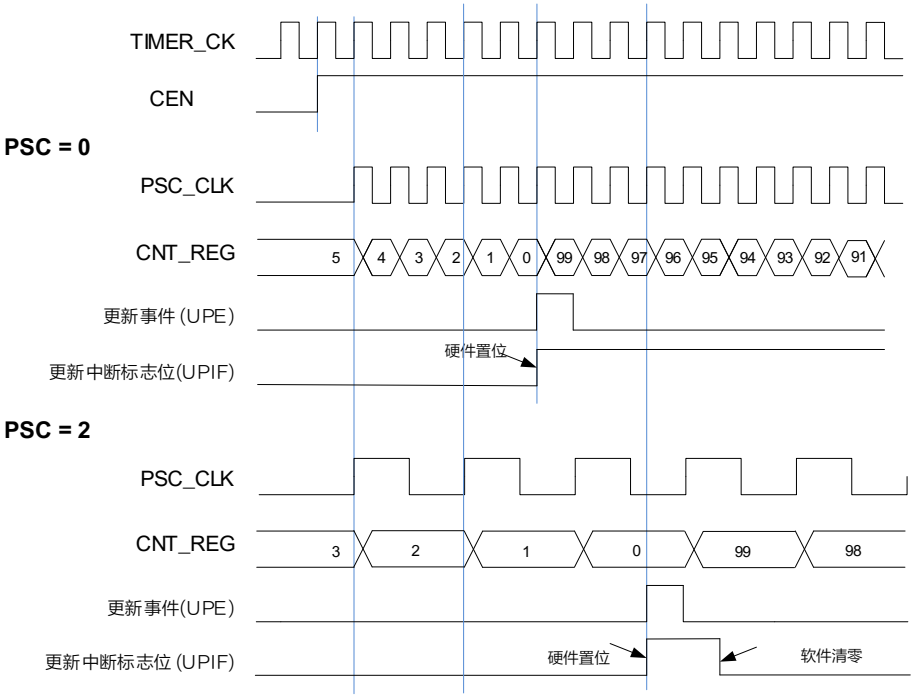
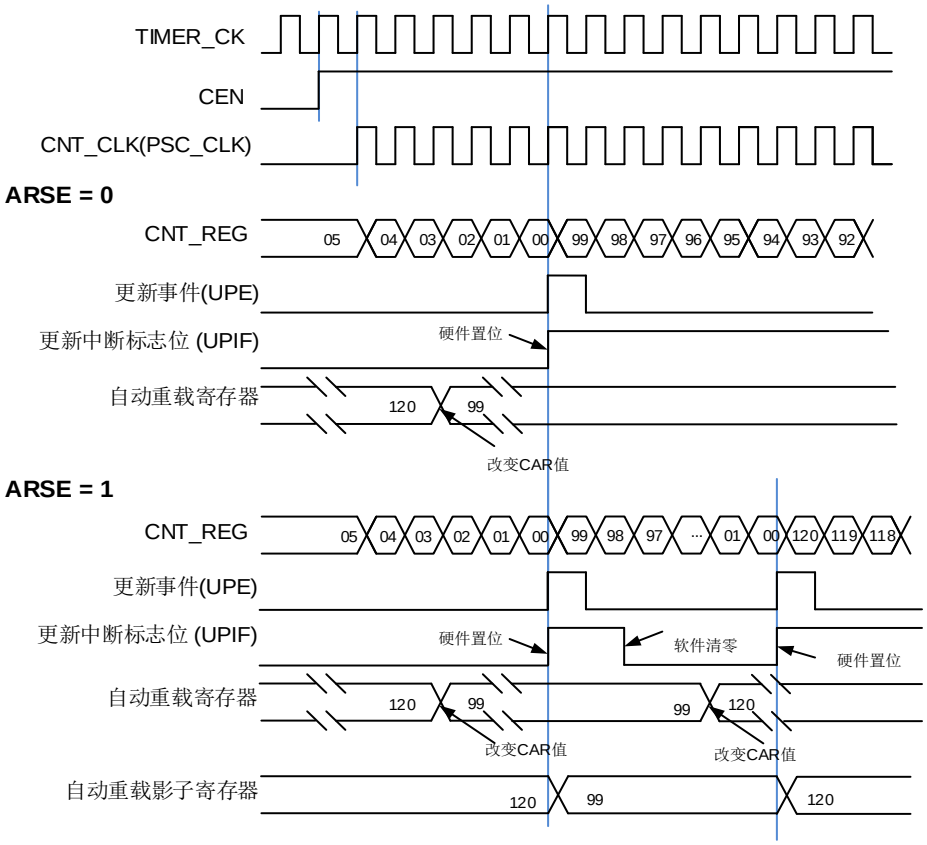


图 23-84. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



.

中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到(TIMERx_CAR-1)产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。计数方向被硬件自动更新。

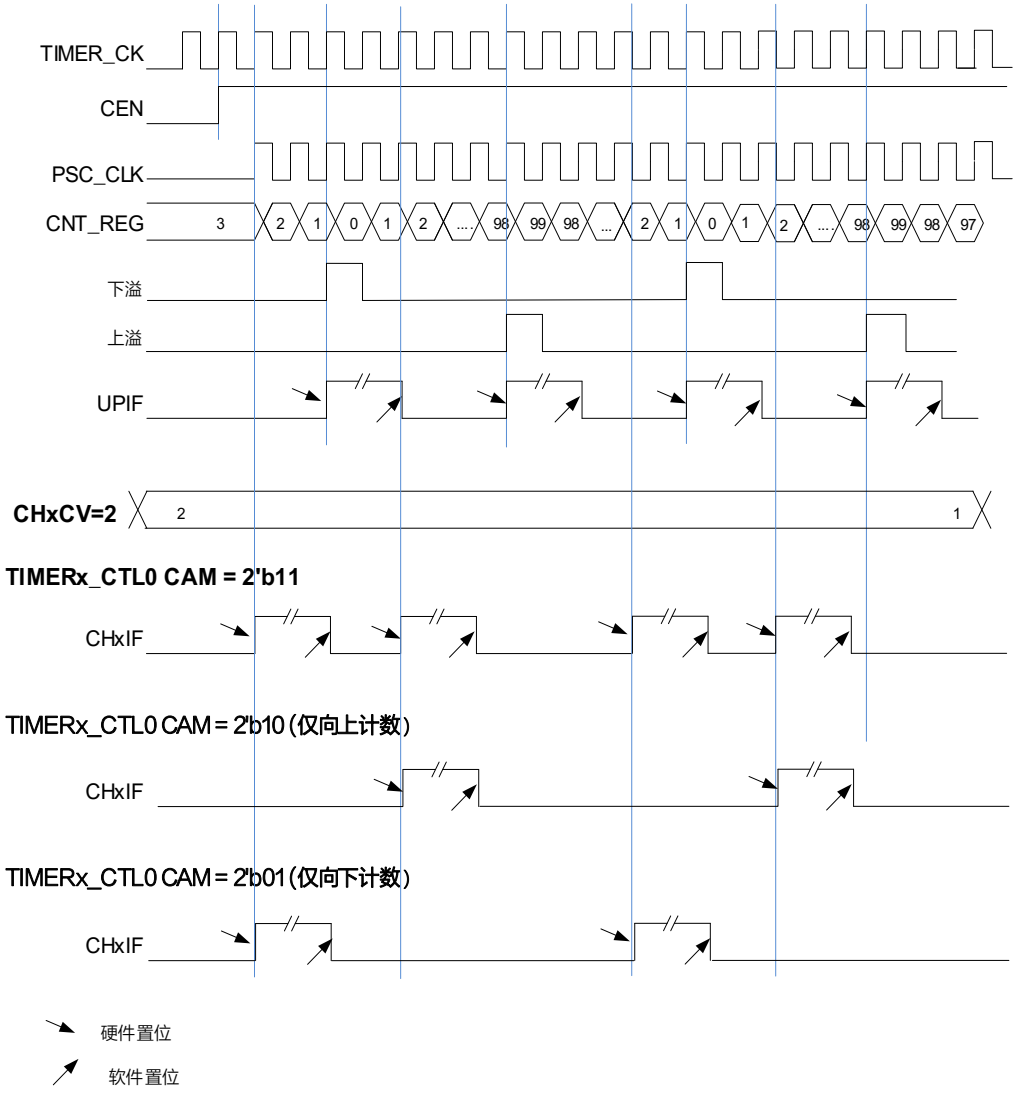
将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 23-55. 中央计数模式计数器时序图*。如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

*图 23-55. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为

图 23-85. 中央计数模式计数器时序图



重复计数器

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 寄存器的 CREP0 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 寄存器中 CREP0 的值并产生一个更新事件。

新写入的 CREP0 值将在下一次更新事件到来时生效。当 CREP0 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 值何时生效。如果在写入奇数到CREP0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 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事

件。

图 23-86. 中央计数模式下计数器重复时序图

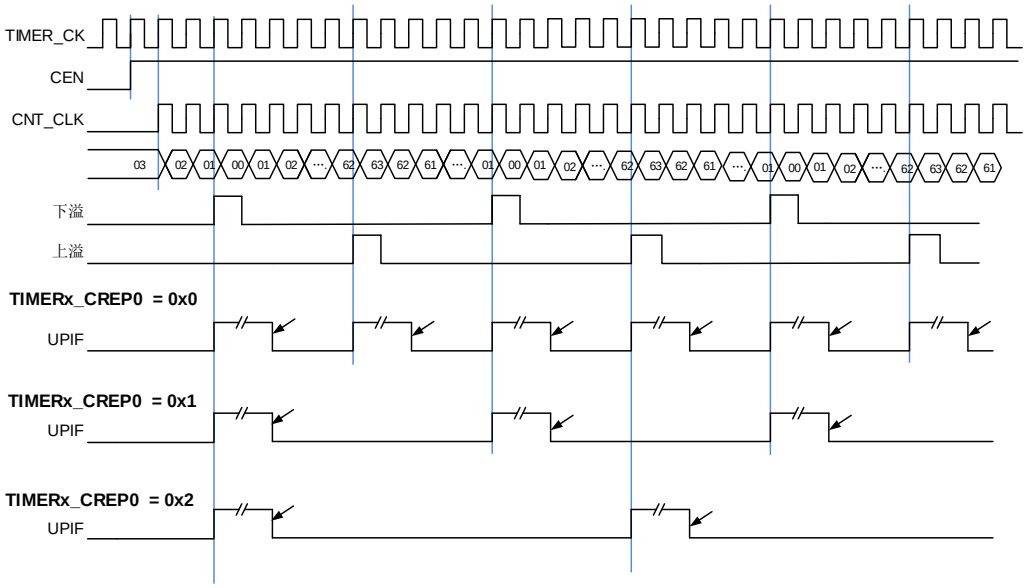


图 23-87. 在向上计数模式下计数器重复时序图

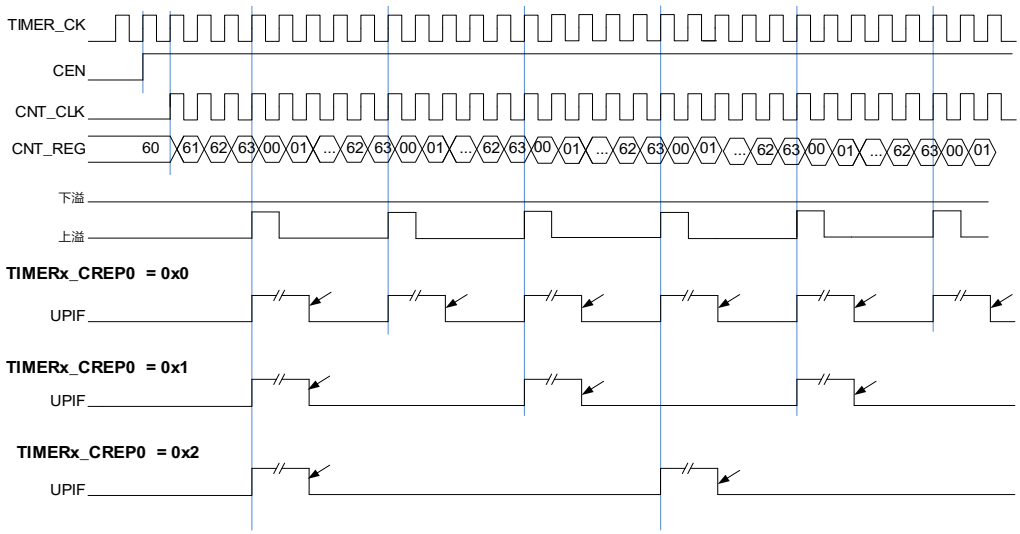
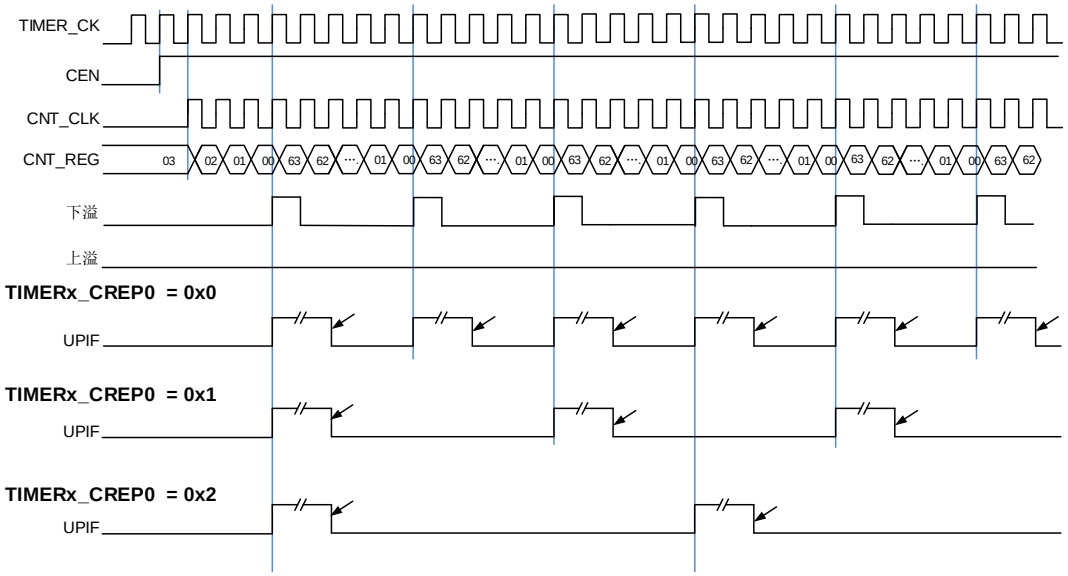


图 23-88. 在向下计数模式下计数器重复时序图



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_MCH0CV(x=0,1)寄存器会捕获计数器当前的值，同时 CHxIF/ MCH0IF(x=0,1)位置 1，如果 CHxIE/ MCH0IE =1(x=0,1)，则产生相应的通道中断。

图 23-89. 通道 0 输入捕获逻辑

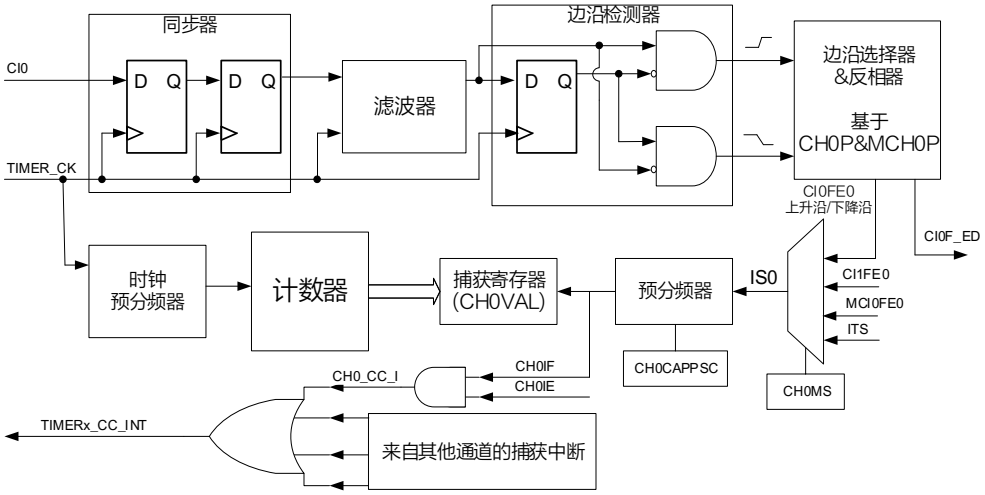
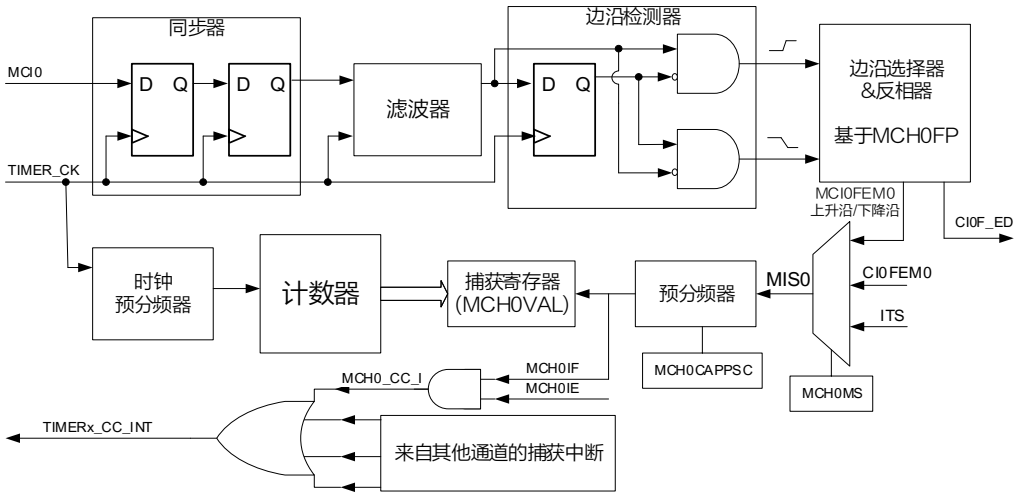


图 23-90. 多模式通道 0 输入捕获逻辑



通道输入信号 CIx/ MCIx 有两种选择，一种是 TIMERx_CHx/ TIMERx_MCHxCV 信号，另一种是 TIMERx_CH0，TIMERx_CH1 和 TIMERx_CH2 异或之后的信号(仅限于 CI0)。

通道输入信号 CIx/ MCIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP/ MCHxP、MCHxFP 选择使用上升沿或者下降沿。配置 CHxMS/ MCHxMS，可以选择其他通道的输入信号或内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，TIMERx_CHxCV/ TIMERx_MCHxCV 存储计数器的值。

配置步骤如下：

第一步：滤波器配置(TIMERx_CHCTL0 寄存器中 CHxCAPFLT 位和 TIMERx_MCHCTL0 寄存器中 CHxMCAPFLT)：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT/ CHxMCAPFLT 位。

第二步：边沿选择(TIMERx_CHCTL2 寄存器中 CHxP 和 MCHxP 位，TIMERx_MCHCTL2 寄

存器中 MCHxFP[1:0]位域):

配置 CHxP 和 MCHxP 位或 MCHxFP 位域选择上升沿或者下降沿。

第三步: 捕获源选择(TIMERx_CHCTL0 寄存器中 CHxMS、TIMERx_MCHCTL0 寄存器中 MCHxMS):

一旦通过配置 CHxMS/ MCHxMS 选择输入捕获源，必须确保通道配置在输入模式(CHxMS!=0x000 或 MCHxMS!=0x000)，而且 TIMERx_CHxCV/TIMERx_MCHxCV 寄存器不能再被写。

第四步: 中断使能(TIMERx_DMAINTEN 寄存器中 CHxIE、CHxDEN 位和 MCHxIE、MCHxDEN 位):

使能相应中断，可以获得中断和 DMA 请求。

第五步: 捕获使能(TIMERx_CHCTL2 寄存器中 CHxEN/ MCHxEN)。

结果: 当期望的输入信号发生时，TIMERx_CHxCV/ TIMERx_MCHxCV 被设置成当前计数器的值，CHxIF/ MCHxIF 位置 1。如果 CHxIF/ MCHxIF 位已经为 1，则 CHxOF/ MCHxOF 位置 1。根据 TIMERx_DMAINTEN 寄存器中 CHxIE、CHxDEN 位和 MCHxIE、MCHxDEN 位的配置，相应的中断和 DMA 请求会被提出。

直接产生: 软件设置 CHxG 位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量 TIMERx_CHx 和 TIMERx_MCHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 3'b001，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 3'b010，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

■ 输出比较模式

图 23-91. 输出比较逻辑(当 MCHxMSEL = 2'00 时，x=0)，*图 23-92. 输出比较逻辑(当 MCHxMSEL = 2'11 时，x=0)*和*图 23-93. 输出比较逻辑(x=1)*给出了通道的输出比较逻辑。

图 23-91. 输出比较逻辑(当 MCHxMSEL = 2'00 时，x=0)

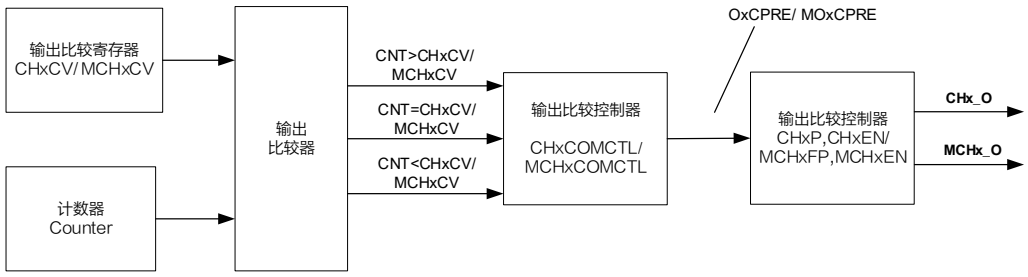


图 23-92. 输出比较逻辑(当 MCHxMSEL = 2'11 时，x=0)

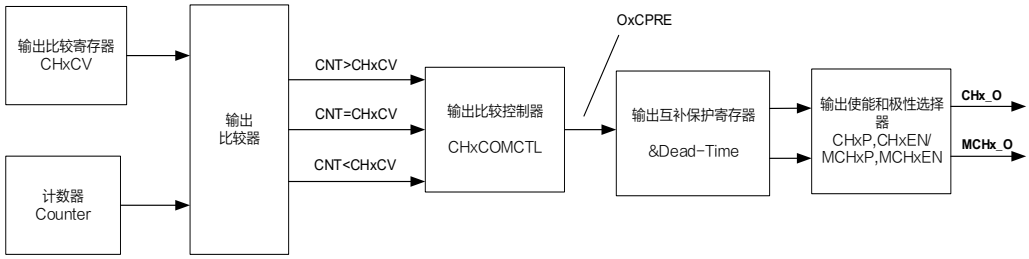
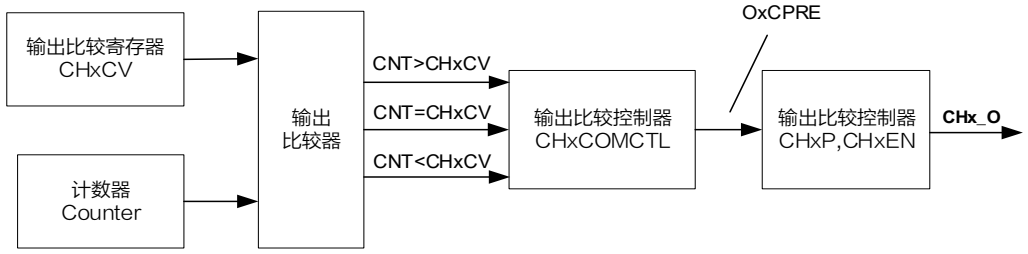


图 23-93. 输出比较逻辑(x=1)



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 23-91. 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23-92. 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_CCHP0 寄存器中的相关位(ROS、IOS、POE 和 DTCFG 等位)有关。详情请见*互补输出*。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL/ MCHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者翻转。当计数器的值与 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值匹配时，CHxIF/ MCHxIF 位被置 1，如果 CHxIE/ MCHxIE = 1 则会产生中断，如果 CHxDEN/ MCHxDEN =1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

- 设置CHxCOMSEN/ MCHxCOMSEN位来配置输出比较影子寄存器；
- 设置CHxCOMCTL/ MCHxCOMCTL位来配置输出模式(置高电平/置低电平/翻转)；
- 设置CHxP/ MCHxP/ MCHxFP位来选择有效电平的极性；
- 设置CHxEN/MCHxEN使能输出。

第三步：通过 CHxIE/ MCHxIE/ CHxDEN/ MCHxDEN 位配置中断/DMA 请求使能。

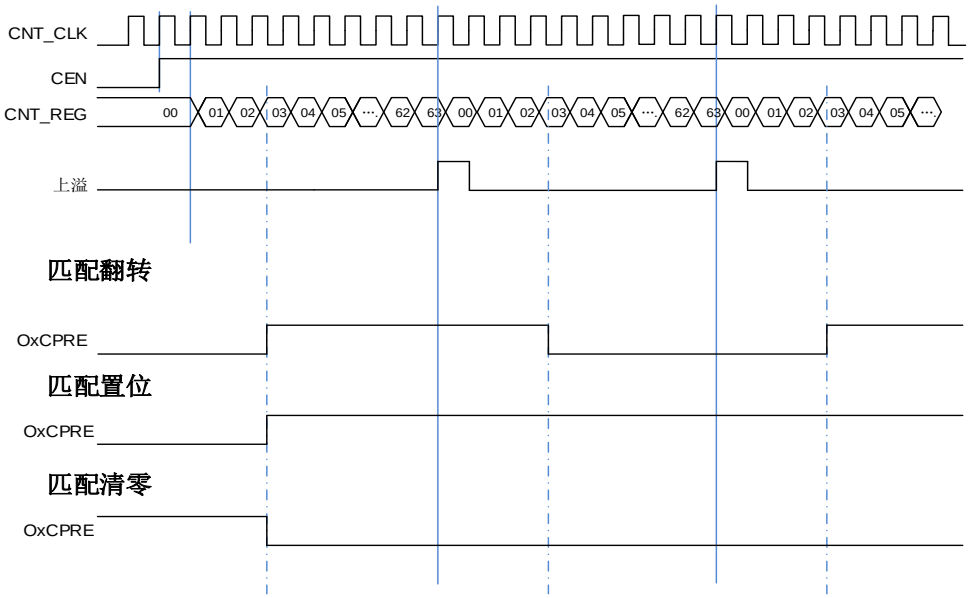
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

TIMERx_CHxCV/ TIMERx_MCHxCV 可以在运行时根据你所期望的波形而改变。

第五步：设置 CEN 位使能定时器。

*图 23-94. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 23-94. 三种输出比较模式



PWM 模式

在 PWM 输出模式下(PWM 模式 0 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0111)，通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器值决定。*图 23-15. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由(2*TIMERx_CAR 寄存器值)决定，占空比由(2*TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器值)决定。*图 23-16. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下(CHxCOMCTL/ MCHxCOMCTL =4'b0110)，如果 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平；PWM1 模式下(CHxCOMCTL/ MCHxCOMCTL=4'b0111)，如果 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为无效电平。

图 23-95. EAPWM 时序图

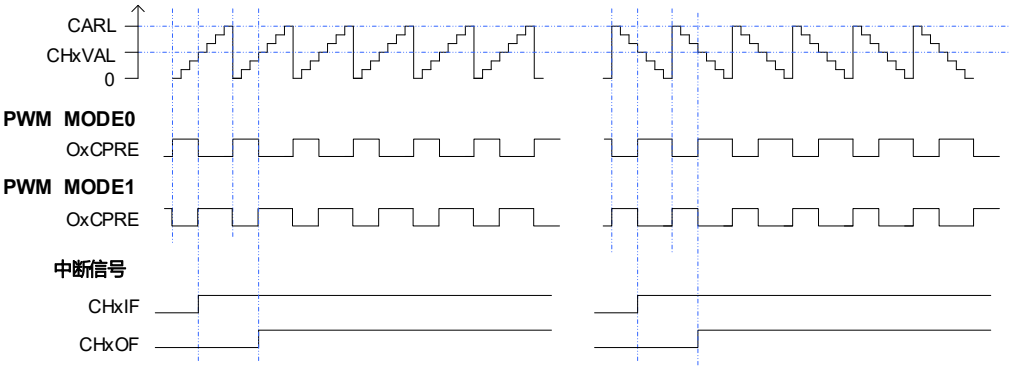
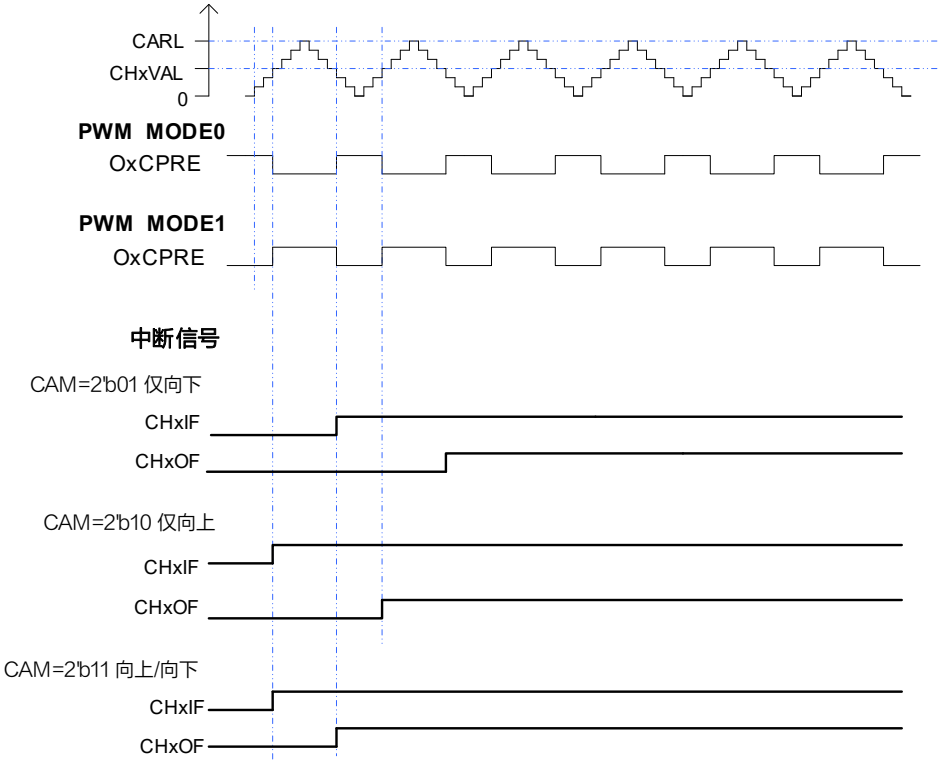


图 23-96. CAPWM 时序图



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..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。

当 CHxVAL 或 CHxCOMVAL_ADD = 0 / CARL，通道输出被进行了特殊处理，通过 TIMERx_CHCTL2 寄存器的 CHxPERFOREN 位置位，强制 OxCPRE 输出高电平或低电平(根据所选复合 PWM 模式确定)。

PWM 的周期取决于(CARL + 0x0001)，PWM 脉冲宽度可以下*表 23-12. 复合 PWM 脉冲宽度*计算。

表 23-12. 复合 PWM 脉冲宽度

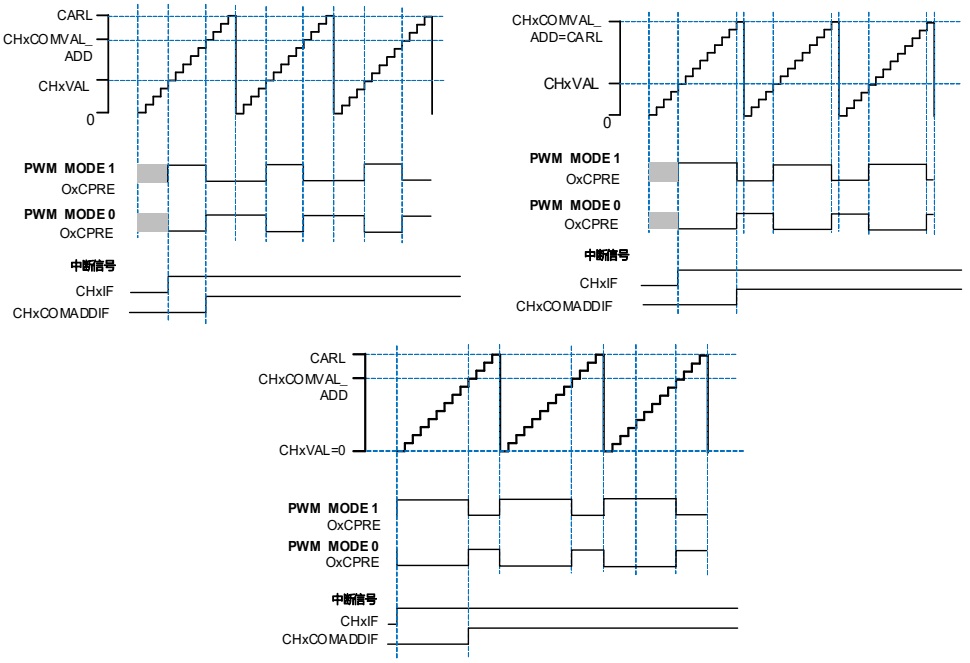
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD <= CARL	PWM 模式 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	PWM 模式 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL <= CARL	PWM 模式 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM 模式 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD <= CARL)或 (CHxVAL > CARL > 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：

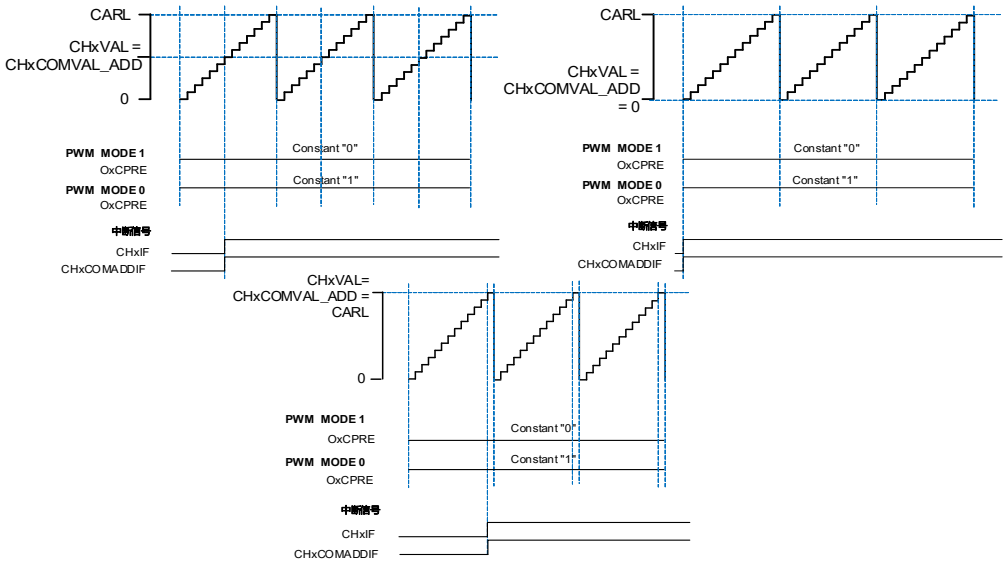
1）CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-97. 通道 x 输出 PWM(CHxVAL < CHxCOMVAL_ADD)



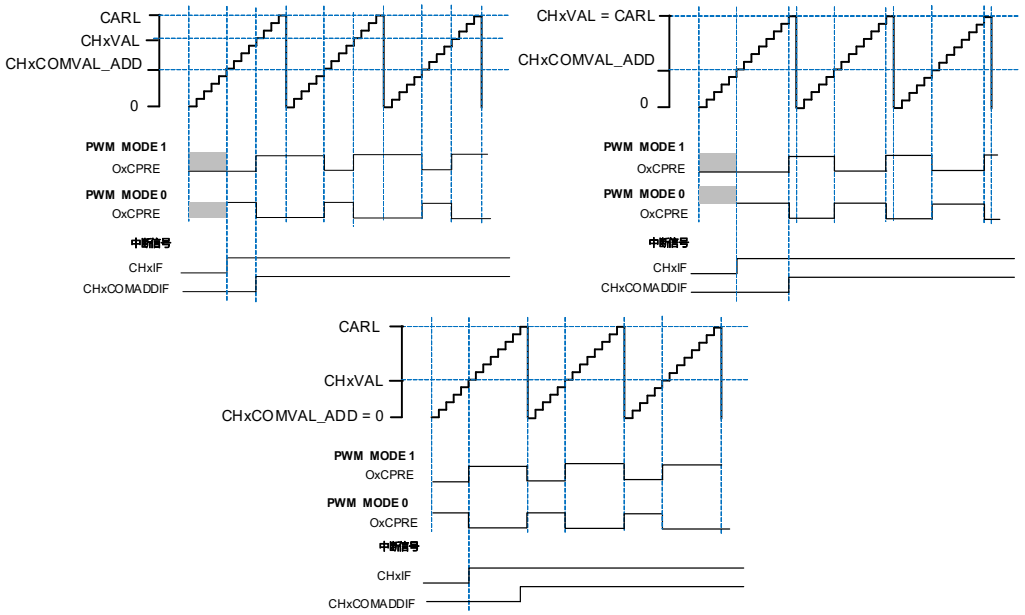
2） CHxVAL = CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-98. 通道 x 输出 PWM(CHxVAL = CHxCOMVAL_ADD)



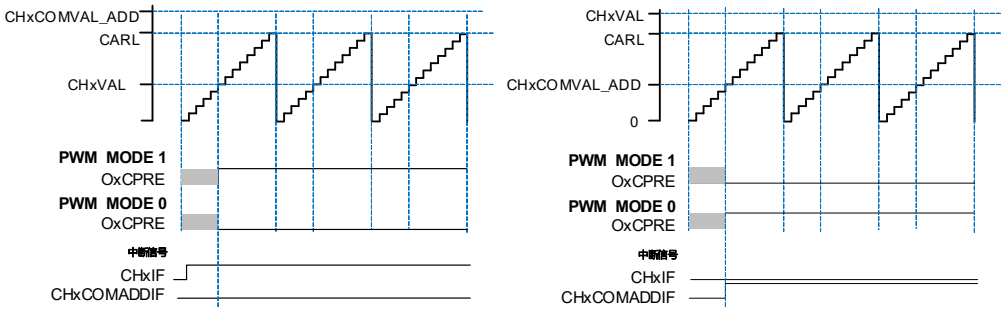
3） CHxVAL > CHxCOMVAL_ADD, CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 23-99. 通道 x 输出 PWM(CHxVAL > CHxCOMVAL_ADD)



4） CHxVAL或CHxCOMVAL_ADD值大于CARL。

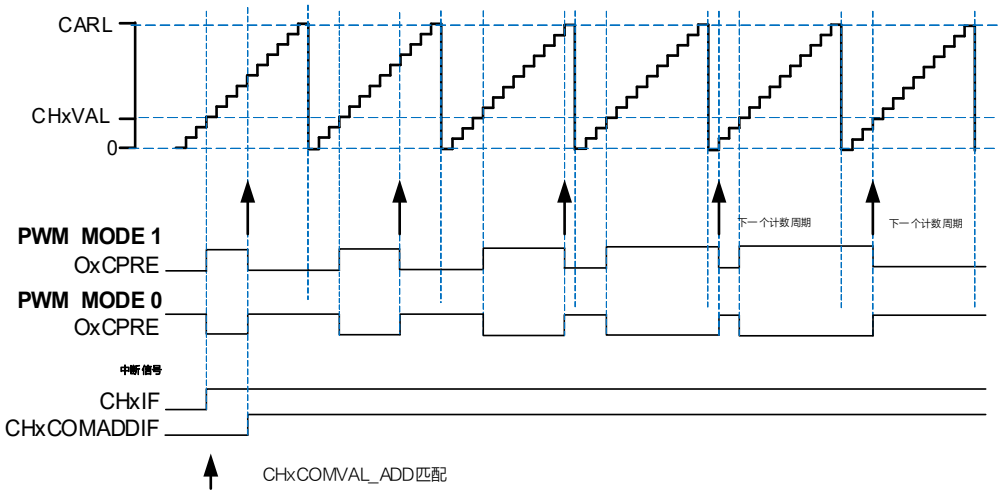
图 23-100. 通道 x 输出 PWM(CHxVAL 或 CHxCOMVAL_ADD > CARL)



复合 PWM 模式支持不修改周期只修改占空比的 PWM 信号的生成。*图 23-101. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变*显示 PWM 输出和中断波形。

在某些情况下，CHxCOMVAL_ADD 的匹配事件可以发生在下一个计数周期(CHxCOMVAL_ADD 值在计数器到达 CHxVAL 值之后被写入，且 CHxCOMVAL_ADD 值小于或者等于 CHxVAL 值)。

图 23-101. 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；
- CHxOMPSEL = 2'b10，只有在计数器向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b11，无论计数器向上计数还是向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；

图 23-102. 边沿对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00)

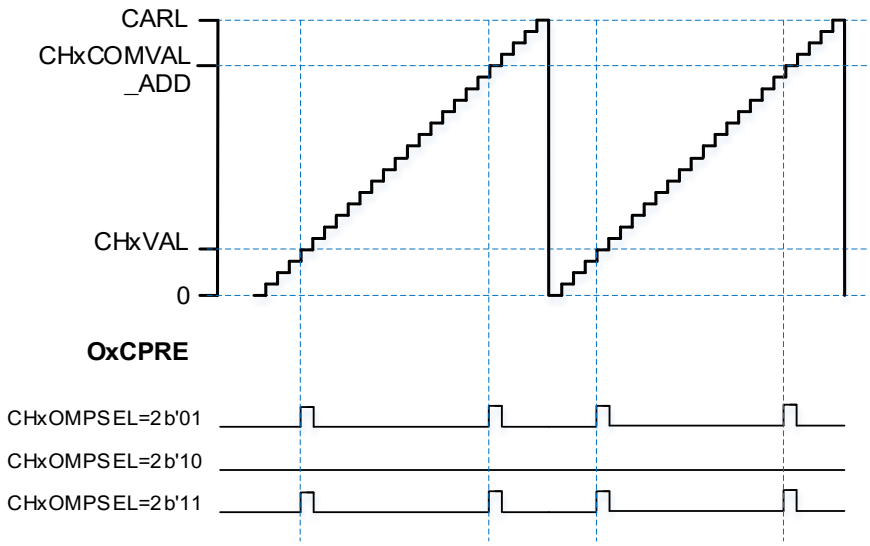
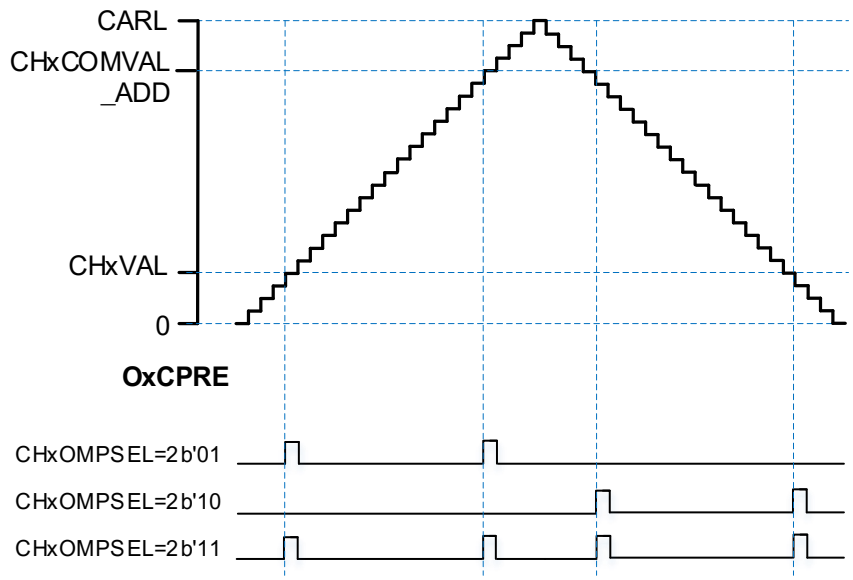


图 23-103. 中央对齐模式下 CHx_O 输出脉冲(CHxOMPSEL!=2'b00)



通道输出参考信号

如*图 23-91. 输出比较逻辑(当 MCHxMSEL = 2'00 时，x=0)*，*图 23-92. 输出比较逻辑(当 MCHxMSEL = 2'11 时，x=0)*和*图 23-93. 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 CHxMOMCTL位配置。

当 CHx_O 和 MCHx_O 输出互补时，这两个信号不能同时有效。TIMERx 的 1 对通道具有此功能。互补信号 CHx_O 和 MCHx_O 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 MCHxEN 位，TIMERx_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

当 CHx_O 和 MCHx_O 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考*表 23-13. 由参数控制的互补输出表(MCHxMSEL=2'b11)*。

表 23-13. 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="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / MCHx_O = LOW
CHx_O / MCHx_O 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/ MCHx_O输出关闭状态(2):
通道先输出无效电平：CHx_O = CHxP，MCHx_O = MCHxP)；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，MCHx_O = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/MCHx_O输出关闭状态:
通道先输出无效电平：CHx_O = CHxP，MCHx_O = MCHxP)；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，MCHx_O = ISOxN</td></tr>
<tr><td>1</td><td>0</td><td>0/1</td><td>0</td><td>0</td><td colspan="2">CHx_O/MCHx_O = LOW</td></tr>
</table>

互补参数					输出状态	
POEN	POEN	POEN	POEN	POEN	CHx_O	MCHx_O
					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
			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。

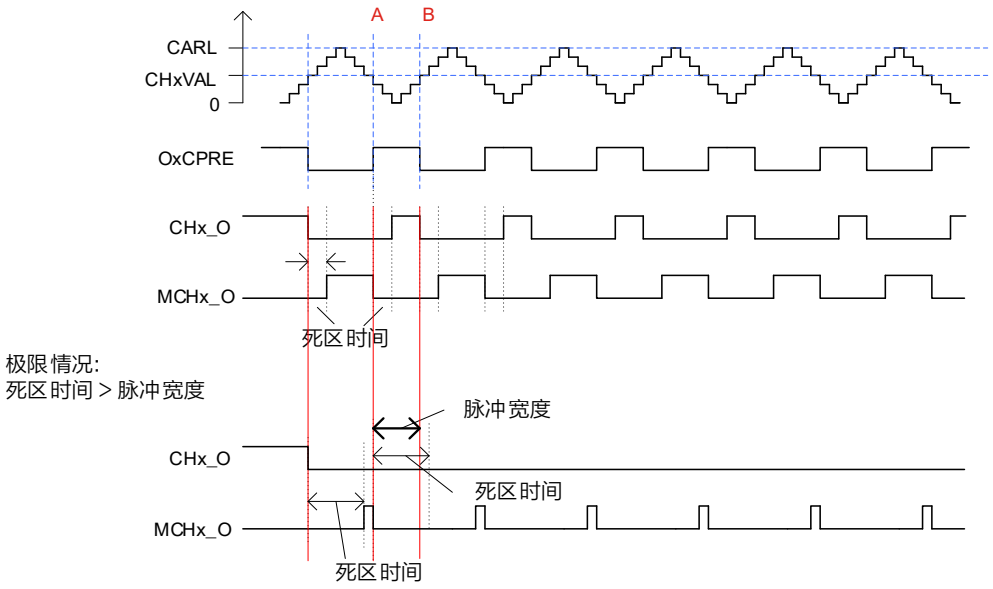
死区时间插入

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 0(TIMERx_CCHP0)*。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时(TIMERx 计数器=TIMERx_CHxCV)，OxCPRE 翻转。在*图 23-104. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 MCHx_O 信号立刻变为低电平。同样，在 B 点，计数器再次匹配(TIMERx 计数器=TIMERx_CHxCV)，OxCPRE 信号被清 0，CHx_O 信号被立即清零，MCHx_O 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。有时会有一些死角事件发生，例如：如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O 信号一直为无效值，如*图 23-104. 带死区时间的互补输出*所示。

图 23-104. 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 CHxMOMCTL 位配置)时，MCHx_O 输出与 CHx_O 输出互补。在这种情况下，CHx_O 和 MCHx_O 信号不能同时设置为有效电平。

通用 L3 定时器具有 BREAK 中止功能。可以通过将 TIMERx_CCHP0 寄存器中的 BRKEN 位置 1 来使能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRKP 位配置，电平有效。

使用 BREAK 功能时，CHx_O 和 MCHx_O 信号的输出电平由以下位控制：TIMERx_CCHP0

寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

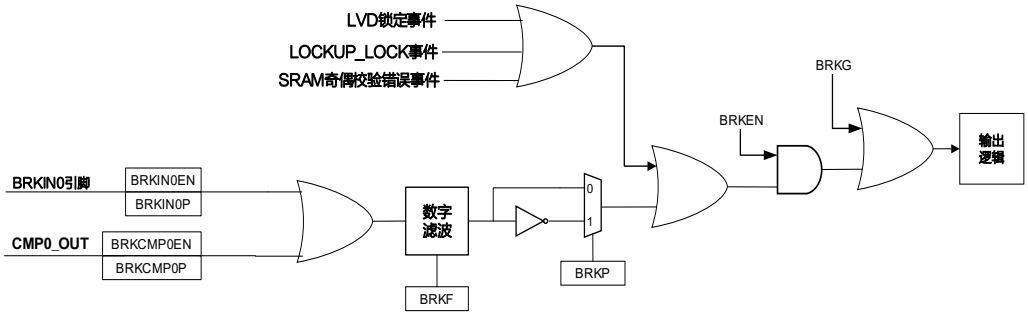
中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKIN0输入；
- 系统源： LVD锁定事件，Cortex®-M33锁定事件或SRAM奇偶校验错误事件；
- 片上外设源：比较器输出。

BREAK 中止事件也可以由软件置位 TIMERx_SWEVG 寄存器中的 BRKG 位产生。

如*图 23-105. BREAK 中止功能逻辑图*所示，BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚。

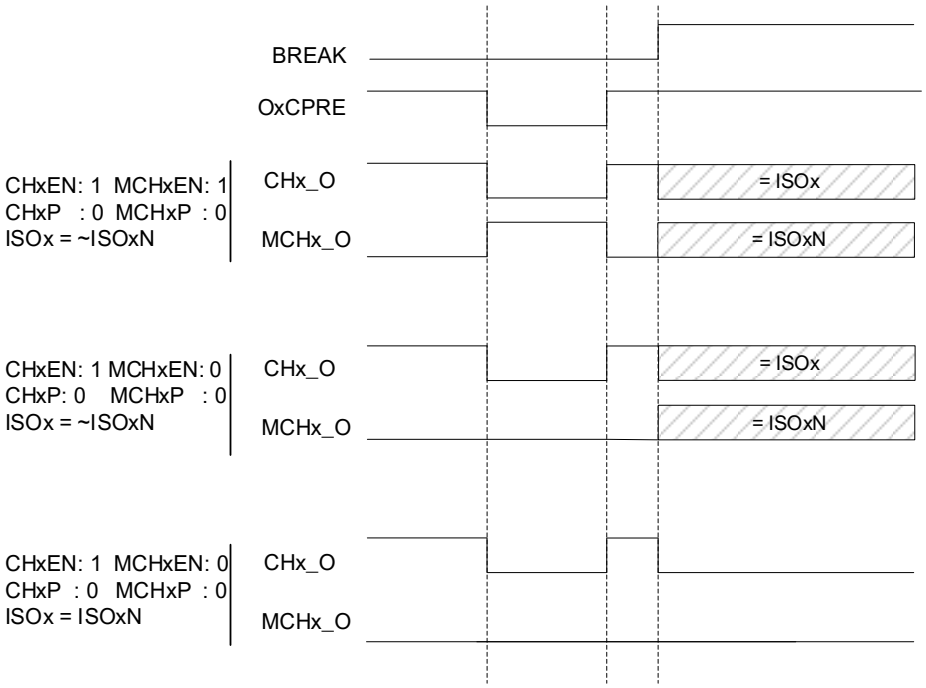
图 23-105. BREAK 中止功能逻辑图



BREAK 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 BREAK 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 BREAK 中止事件时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 MCHx_O 的输出由 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 位确定。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。当 IOS=1 时，通道输出情况如*图 23-106. 通道响应 BREAK 中止输入(高电平有效)时，输出信号的行为(IOS=1)*所示，首先通道互补输出为复位状态，然后死区时间发生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

图 23-106. 通道响应 BREAK 中止输入(高电平有效)时，输出信号的行为(IOS=1)



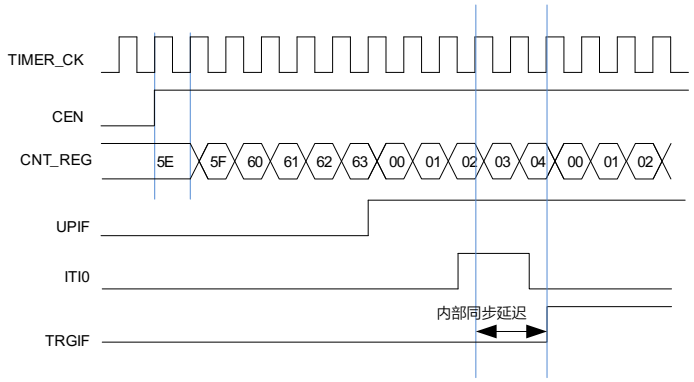
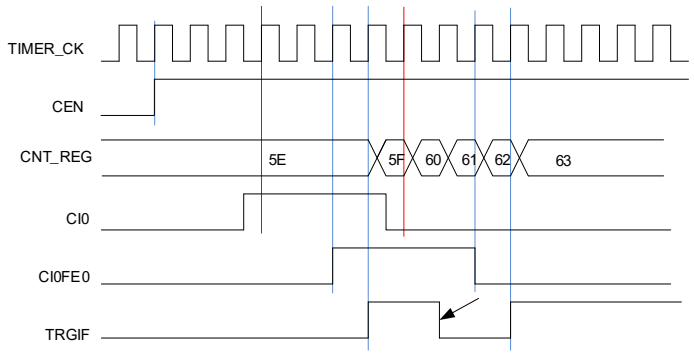
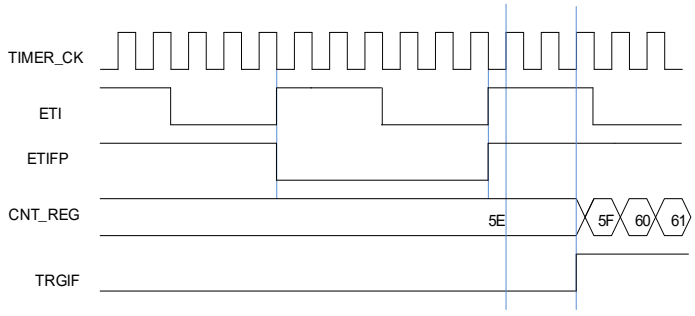
发生中止事件时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

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=15,16)寄存器中的 TSCFGy[4:0] (y=3..6)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6)位域的值来选择。

表 23-14. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y=3：复位模式 y=4：暂停模式 y=5：事件模式 y=6：外部时钟模式0	TSCFGy[4:0] 00000：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)，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
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

	模式选择	触发源选择	极性选择	滤波和预分频
	图 23-107. 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	TI0S=0(非异或)[MCH0P=0,CH0P=0] CI0FE0不反相，在上升沿捕获	在这个例子中滤波被旁路
	图 23-108. 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
	图 23-109. 事件模式下的控制电路  			

单脉冲模式

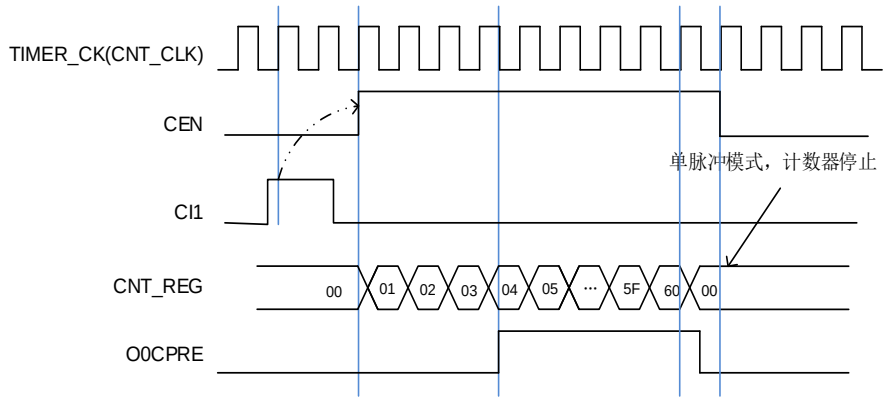
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，可使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL/ MCHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 23-110. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考*高级定时器(TIMERx, x=0, 7, 19)定时器互连*。

计数器同步、初始方向和值刷新

参考*高级定时器(TIMERx, x=0, 7, 19)计数器同步、初始方向和值刷新*。

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成

M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.4.5. TIMERx 寄存器(x=14,40,41,42,43,44)

TIMER14 基地址：0x4001 4000

TIMER40 基地址：0x4001 4C00

TIMER41 基地址：0x4001 4400

TIMER42 基地址：0x4000 E000

TIMER43 基地址：0x4000 E400

TIMER44 基地址：0x4000 E800

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						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置 CKDIV，规定定时器时钟(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
6:5	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=3'b000)，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=3'b000)，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 - 计数器上溢或者下溢 - 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 - 计数器上溢或者下溢 - 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				MMC[3]	保留								
rw						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC[0]	保留	CCSE
					rw	rw	rw	rw	rw			rw	rw		rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:26	保留	必须保持复位值
25	MMC[3]	主模式控制 请参考 MMC[2:0]的描述。
24:11	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	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 输出低电平 1：当 POEN 复位，CH0_ON 输出高电平 此位只有在 TIMERx_CCHP0 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 输出低电平 1：当 POEN 复位，CH0_O 输出高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP0 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
7	TI0S	通道 0 触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0，CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 0000：复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲，后一种情况下，TRGO 上的信号相对实际的复位会有一个延迟。 0001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。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 TRGO 上会有一个延迟，除非选择了主/从模式。

		0010：更新。主模式控制器选择更新事件作为 TRGO。 0011：捕获/比较脉冲.通道 0 在发生一次捕获或一次比较成功时，主模式控制器产生一个 TRGO 脉冲 0100：比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO 0101：比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO 1000：保留 1001：在这种模式下主模式控制器选择软同步事件信号(设置SWSYNCG位为1产生)作为触发输出TRGO。 1010~1111：保留
3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道 CHx/MCHx 的 DMA 请求 . 1：当更新事件发生，发送通道 CHx/MCHx 的 DMA 请求
2	CCUC[0]	换相控制影子寄存器更新控制 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 TRGI 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 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:0	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	BRKIF	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	BRKIF	BREAK 中断标志位 一旦 BREAK 输入有效，由硬件对该位置'1'。如果 BREAK 输入无效，则该位可由软件清'0'。 0：无 BREAK 事件产生 1：BREAK 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								BRKG	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	BRKG	产生 BREAK 事件 该位由软件置 1，用于产生一个 BREAK 事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生 BREAK 事件 1：产生 BREAK 事件
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器 (CHxEN、MCHxEN 和 CHxCOMCTL) 的互补输出被更新(根据 TIMERx_CTL1 中 CCSE 值)。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:3	保留	必须保持复位值.
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0(TIMERx_CHCTL0)

地址偏移：0x18

复位值：0x0000 0000

该寄存器通过字访问(32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	CH1ADDUPS	CH0ADDUPS	保留	CH1COMCTL[3]	保留							CH0COMCTL[3]
		保留					保留								保留
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	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		保留	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：通道 0 附加比较输出影子寄存器禁能 1：通道 0 附加比较输出影子寄存器使能 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27	CH1ADDUPS	通道1附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH1COMV_ADD寄存器 1：在发生计数器计数值匹配 CH1VAL 值时，更新 TIMERx_CH1COMV_ADD 寄存器
26	CH0ADDUPS	通道0附加寄存器更新源 0：在发生更新事件时，更新TIMERx_CH0COMV_ADD寄存器 1：在发生计数器计数值匹配 CH0VAL 值时，更新 TIMERx_CH0COMV_ADD 寄存

		器
25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	保留	必须保持复位值
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 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=15,16)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 和 MCH0_O 的值。O0CPRE 高电平有效，而 CH0_O 和 MCH0_O 的有效电平取决于 CH0P 和 MCH0P 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**：在复合 PWM 模式下(CH0CPWMEN = 1'b1 和 CH0MS = 3'b000)，通道 0 的 PWM 输出信号由 TIMERx_CH0CV 和 TIMERx_CH0COMV_ADD 寄存器共同确定。详细信息请参考***复合 PWM 模式***。 在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000(比较模式)时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM模式0或者PWM模式1，会加快捕获 / 比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为 3 个时钟周期
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(当 MCH0MSEL[1:0] = 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=15,16)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

该寄存器可以按半字(16 位)或字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留.		CH1PERFOREN	CH0PERFOREN	保留.											
		rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.								MCH1P	保留	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
								rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1PERFOREN	Channel 1在复合PWM模式下，周期点匹配时刻O1CPRE电平值设置。 参考 CH0PERFOREN 描述。
28	CH0PERFOREN	Channel 0在复合PWM模式下，周期点匹配时刻O0CPRE电平值设置。 0：禁能 1：复合PWM0模式下，周期点匹配时刻电平值强制拉高，复合PWM1模式下，周期点匹配时刻电平值强制拉低。 注意：中央对齐模式下，周期点为计数器下溢时刻；边沿对齐模式下，周期点为计数器溢出时刻。
27:8	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	保留	必须保持复位值
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_CCHP0 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。 10：保留。 11：把通道 0 输入信号的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不翻转。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器(TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器通过字访问(32 位)。

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

保留

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器(TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器通过字访问(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															

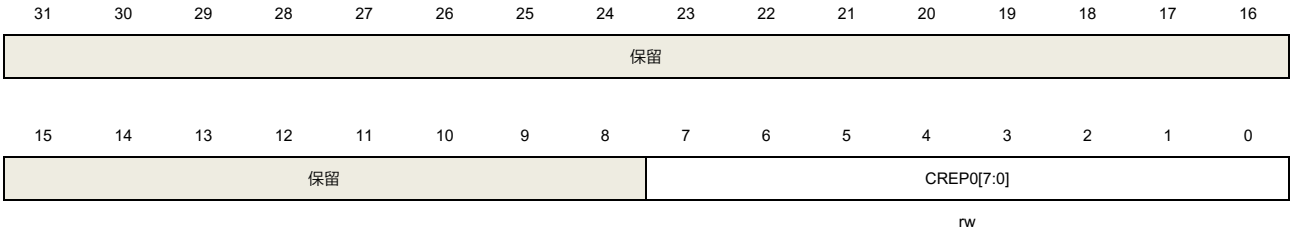
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

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)。

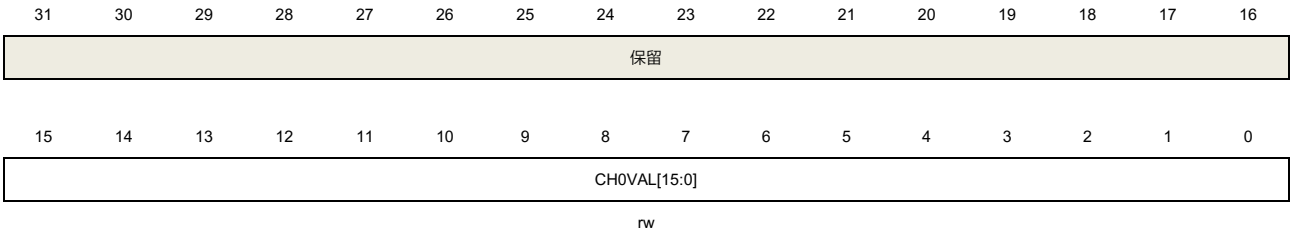


位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP0[7:0]	重复计数器的值 0 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。 **注意**：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

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)。

31	30	29	28	27	26	25	24	23	22	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。

互补通道保护寄存器 0(TIMERx_CCHP0)

地址偏移：0x44
复位值：0x0000 0000

该寄存器通过字访问(32 位)。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	BRKF[3:0]	BREAK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 BREAK 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，BREAK 异步有效，N=1 0001：$f_{SAMP} = f_{CK_TIMER}$, N=2 0010：$f_{SAMP} = f_{CK_TIMER}$, N=4 0011：$f_{SAMP} = f_{CK_TIMER}$, N=8 0100：$f_{SAMP} = f_{DTS}/2$, N=6 0101：$f_{SAMP} = f_{DTS}/2$, N=8 0110：$f_{SAMP} = f_{DTS}/4$, N=6 0111：$f_{SAMP} = f_{DTS}/4$, N=8 1000：$f_{SAMP} = f_{DTS}/8$, N=6 1001：$f_{SAMP} = f_{DTS}/8$, N=8 1010：$f_{SAMP} = f_{DTS}/16$, N=5 1011：$f_{SAMP} = f_{DTS}/16$, N=6 1100：$f_{SAMP} = f_{DTS}/16$, N=8 1101：$f_{SAMP} = f_{DTS}/32$, N=5 1110：$f_{SAMP} = f_{DTS}/32$, N=6 1111：$f_{SAMP} = f_{DTS}/32$, N=8 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
15	POEN	所有的通道输出使能 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHCTL2 寄存器的 CHxEN, MCHxEN 位)，则开启 CHx_O 和 MCHx_O 输出。 0：禁止通道输出或强制为空闲状态 1：使能通道输出
14	OAEN	自动输出使能 此位定义了 POEN 位是否可以被硬件自动置 1。 0：POEN 位不能被硬件置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位能被硬件自动置 1 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	BREAK 输入信号极性 此位定义了 BREAK 输入的极性。 0：BREAK 输入低电平有效 1：BREAK 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
12	BRKEN	BREAK 输入信号使能 此位置 1 使能 BREAK 输入信号。 0：BREAK 输入禁能 1：BREAK 输入使能 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下"关闭状态"使能

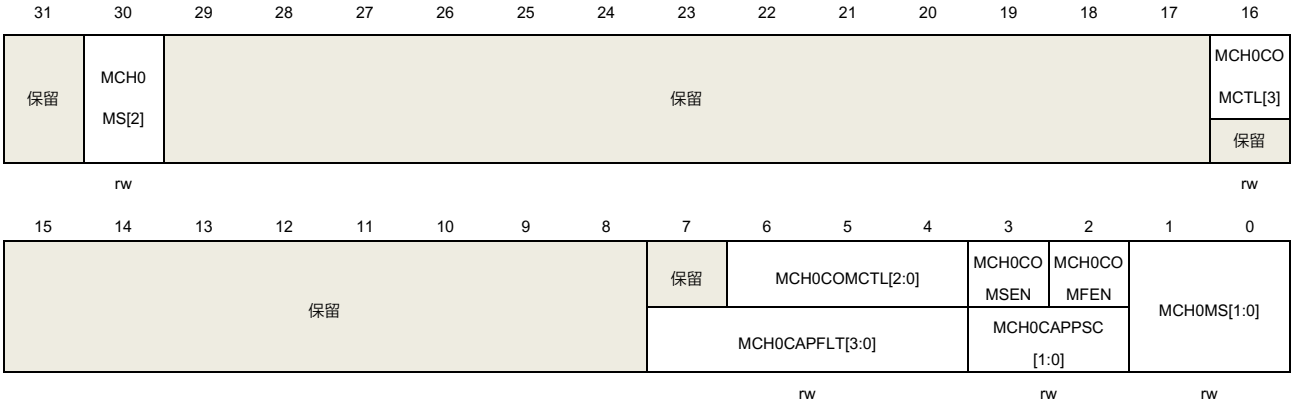
		当 POEN 位被置 1(运行模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见 ***表 23-13. 由参数控制的互补输出表(MCHxMSEL=2'b11)***。 0：输出"关闭状态"禁能。当 CHxEN 或者 MCHxEN 位被清零，对应通道为输出"禁能状态"。 1：输出"关闭状态"使能。当 CHxEN 或者 MCHxEN 位被清零，对应通道为输出"关闭状态"。 此位在 TIMERx_CCHP0 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下"关闭状态"使能 当 POEN 位被清 0(空闲模式)，此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出"关闭状态"。参见 ***表 23-13. 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 MCHxEN 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 MCHxEN 位的值，对应通道为输出"关闭状态"。 此位在 TIMERx_CCHP0 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00：禁能保护模式。无写保护. 01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP0 寄存器中 BRKEN /BRKP/OAEN/DTCFG 位写保护 10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/MCHxP 位(如果相应通道配置为输出模式)，TIMERx_CCHP0 寄存器中 ROS/IOS 位。 11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位(如果相关通道配置为输出模式)写保护。 11：PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0 及 TIMERx_MCHCTL0 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN 位(如果相关通道配置为输出模式)写保护。 系统复位后这两位只能被写一次，一旦 TIMERx_CCHP0 寄存器被写入，这两位被写保护
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3'b0xx：DTvalue =DTCFG [7:0]x 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_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

多模式通道控制寄存器 0(TIMERx_MCHCTL0)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



输出比较模式：

位/位域	名称	描述
31	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:17	保留	必须保持复位值。
16	MCH0COMCTL[3]	多模式通道 0 输出比较控制 请参考MCH0COMCTL[2:0]描述。
15:7	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
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，MO0CPRE 电平才改变。
当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。
当TIMERx_CCHP0寄存器的PROT[1:0]=11且MCH0MS =000(比较模式)时此位不能被改变。

3	MCH0COMSEN	多模式通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_MCH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止多模式通道 0 输出/比较影子寄存器 1：使能多模式通道 0 输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当TIMERx_CCHP0寄存器的PROT [1:0]=11且CH0MS=00时此位不能被改变。
2	MCH0COMFEN	多模式通道0输出比较快速使能 当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH0_O被设置为比较电平而与比较结果无关。 0：禁止多模式通道0输出比较快速. 当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为5个时钟周期 1：使能多模式通道0输出比较快速。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为3个时钟周期
1:0	MCH0MS[1:0]	多模式通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 MCH0EN 位清 0)时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出 001：多模式通道 0 配置为输入，MIS0 映射在 MCI0FEM0 上 010：保留 011：多模式通道 0 配置为输入，MIS0 映射在 ITS 上。此模式仅工作在内部触发输

入被选中时(由 SYSCFG_TIMERxCFG2(x=14,40,41,42,43,44)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_CCHP0 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器(TIMERx_MCH0CV)

地址偏移：0x54
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

MCH0VAL[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 0 附加比较寄存器(TIMERx_CH0COMV_ADD)

地址偏移：0x64
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0COMVAL_ADD[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0COMVAL_ADD[15:0]	通道0附加比较值 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 **注意**：该寄存器仅用于复合PWM模式(当CH0CPWMEN=1时)。

通道 1 附加比较寄存器(TIMERx_CH1COMV_ADD)

地址偏移：0x68
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1COMVAL_ADD[15:0]	通道1附加比较值 当通道1附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 **注意**：该寄存器仅用于复合PWM模式(当CH0CPWMEN=1时)。

控制寄存器 2(TIMERx_CTL2)

地址偏移：0x74
复位值：0x0030 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CH1CPWMEN	CH0CPWMEN	保留						MCH0MSEL[1:0]		保留			
		rw	rw							rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		保留							
				rw		rw									

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1CPWMEN	通道 1 复合 PWM 模式使能 0：通道 1 复合 PWM 模式禁能 1：通道 1 复合 PWM 模式使能
28	CH0CPWMEN	通道 0 复合 PWM 模式使能 0：通道 0 复合 PWM 模式禁能 1：通道 0 复合 PWM 模式使能
27:22	保留	必须保持复位值
21:20	MCH0MSEL[1:0]	多模式通道 0 模式选择 00：独立模式，MCH0 独立于 CH0 01：保留 10：保留 11：互补模式，只有 CH0 可用于输入，MCH0 输出与 CH0 输出互补
19:12	保留	必须保持复位值

11:10	CH1OMPSEL[1:0]	通道 1 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O1CPRE(用来驱动 CH1_O 信号)。 00：O1CPRE 信号根据 CH1COMCTL[2:0]位的配置正常输出。 01：只有在计数器向上计数，匹配事件发生时，O1CPRE 信号输出一个脉冲，且脉冲宽度是一个 CK_TIMER 时钟周期。 10：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，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，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

TIMERx 备用功能控制寄存器 0(TIMERx_AFCTL0)

地址偏移：0x8C
复位值：0x0000 0001

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						BRKCMP0P	保留								BRKIN0P
						rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						BRKCMP0EN	保留								BRKIN0EN
						rw				rw					rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	BRKCMP0P	BREAK CMP0输入极性 0：CMP0输入信号不反相(BRKP =0，输入信号低有效；BRKP =1，输入信号高有效) 1：CMP输入信号反相(BRKP =0，输入信号高有效；BRKP =1，输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

24:17	保留	必须保持复位值。
16	BRKIN0P	BREAK 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BRKP位共同确定。 0：BRKIN0输入信号不反相(BRKP =0，输入信号低有效；BRKP =1，输入信号高有效) 1：BRKIN0输入信号反相(BRKP =0，输入信号高有效；BRKP =1，输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
15:10	保留	必须保持复位值。
9	BRKCMP0EN	BREAK CMP0输入使能 0：CMP0输入禁能 1：CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
8:1	保留	必须保持复位值。
0	BRKIN0EN	BREAK BRKIN0备用输入使能 0：BRKIN0输入禁能 1：BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

计数器初值控制寄存器(TIMERx_CINITCTL)

地址偏移：0xA4
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													SWSYNCG	CINITDIR	CINITVEN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SWSYNCG	软件同步事件产生 该位由软件置位，硬件自动清零，对该位进行读操作，读取值均为0。 0：无效 1：产生软件更新事件
1	CINITDIR	计数器初始计数方向

		0：同步事件发生时，计数器向下计数 1：同步事件发生时，计数器向上计数 当同步事件发生，且计数器从TIMERx_CINITV寄存器加载计数器初始值后，该位表明了此时计数器的计数初始方向。 **注意**：该位仅用于CAM[1:0] != 00的情况。
0	CINITVEN	计数器初值寄存器使能 0：计数器初值寄存器禁能。计数器寄存器不从计数器初值寄存器加载初值。 1：计数器初值寄存器使能。当同步事件发生(通过软件将SWSYNCG位置1产生软件同步事件，输入引脚输入硬件同步事件)时，计数器寄存器可以从计数器初值寄存器加载初值。

计数器初值寄存器(TIMERx_CINITV)

地址偏移：0xA8
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINITVAL[15:0]	计数器初始值 该位域用于表示计数器的初始值。 CINITVEN位为0时，计数器的寄存器不能加载在CINITVAL[15:0]位域中设置的初始值。 CINITVEN位为1时，当同步事件发生时，计数器的寄存器可以加载在CINITVAL[15:0]位域中设置的初始值。

DMA 配置寄存器(TIMERx_DMACFG)

地址偏移：0xE0
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问(读/写)TIMERx_DMATB 寄存器的次数。 6'b000000：传输1次 6'b000001：传输2次 … 6'b111111：传输 63 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。



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	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1：如果 POEN 位与 IOS 位均为 0，则输出无效 0：无影响

23.5. 基本定时器(TIMERx, x=5,6,50,51)

23.5.1. 简介

基本定时器(TIMER5/6/50/51)包含一个无符号 16 位计数器。可以被用作通用定时器和为DAC(数字到模拟转换器)提供时钟。基本定时器可以配置产生 DMA 请求，TRGO 触发连接到DAC。

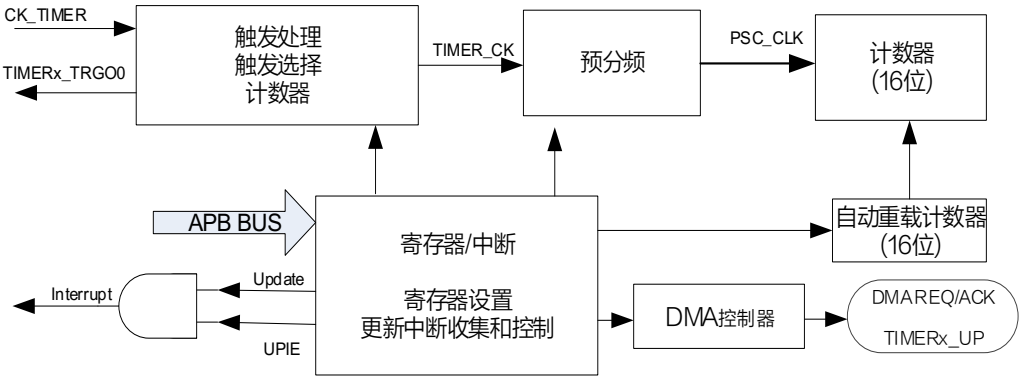
23.5.2. 主要特性

- 计数器宽度：16位(TIMER5/6/50/51)
- 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

23.5.3. 结构框图

*图 23-111. 基本定时器结构框图*提供了基本定时器内部配置的细节。

图 23-111. 基本定时器结构框图



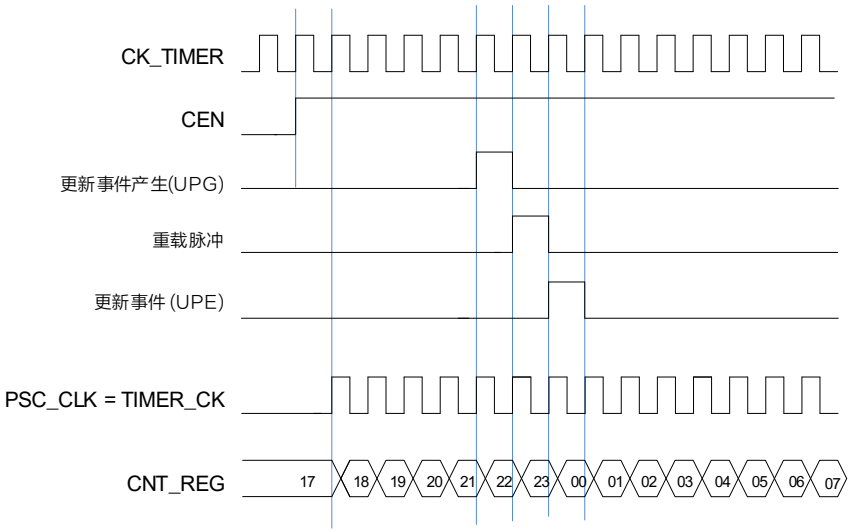
23.5.4. 功能描述

时钟源选择

基本定时器只能由内部时钟源 CK_TIMER 驱动(来自 RCU 模块)。

TIMER_CK 用来驱动计数器预分频器。当 CEN 置位，TIMER_CK 经过预分频器(预分频值由TIMERx_PSC 寄存器确定)产生 PSC_CLK。

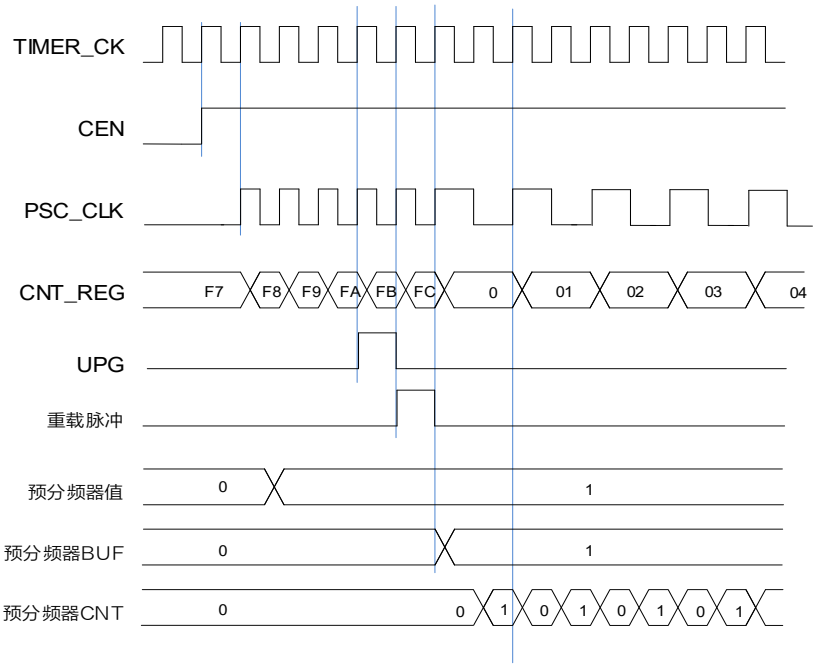
图 23-112. 内部时钟分频为 1 时正常模式下的控制电路



预分频

预分频器可以将定时器的时钟(TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 23-113. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值(定义在 TIMERx_CAR 寄存器中)，一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当 TIMERx_CAR=0x99(TIMERx, x=5,6)时，计数器在不同预分频因子下的行为。

图 23-114. 向上计数时序图，PSC=0/2(TIMERx, x=5,6)

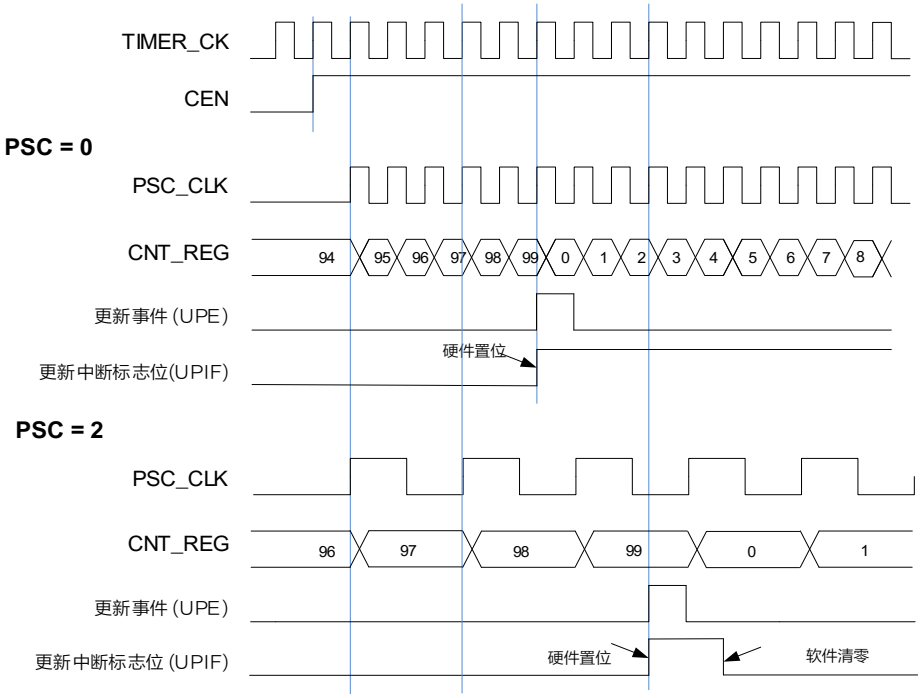
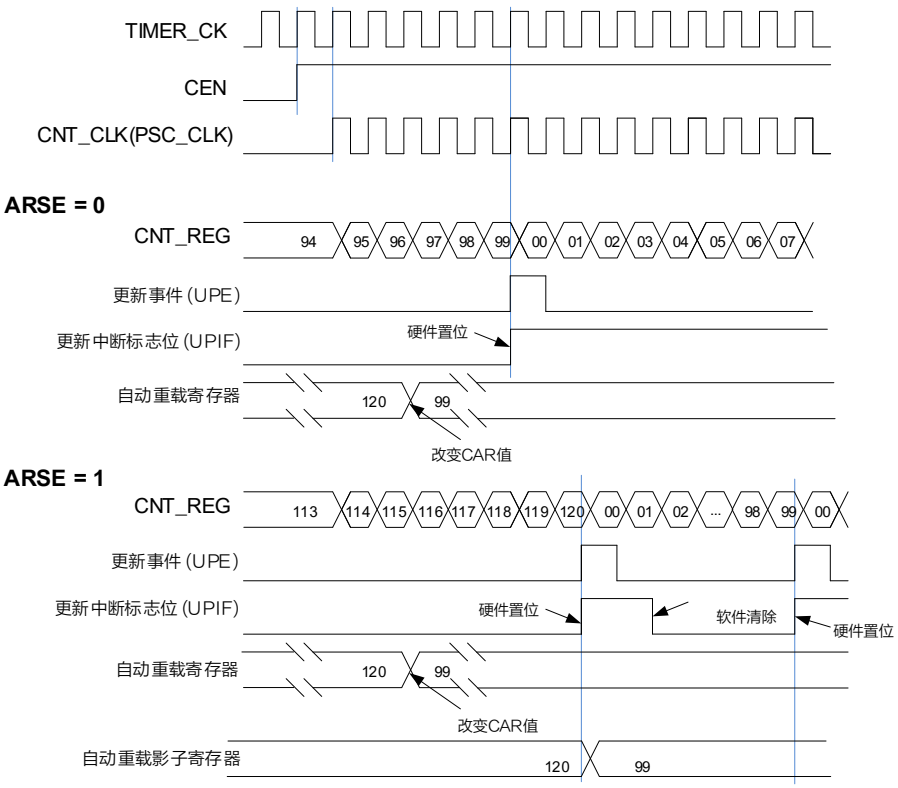


图 23-115. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值(TIMERx, x=5,6)



定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

23.5.5. TIMERx 寄存器(x=5/6/50/51)

TIMER5 基地址：0x4000 1000

TIMER6 基地址：0x4000 1400

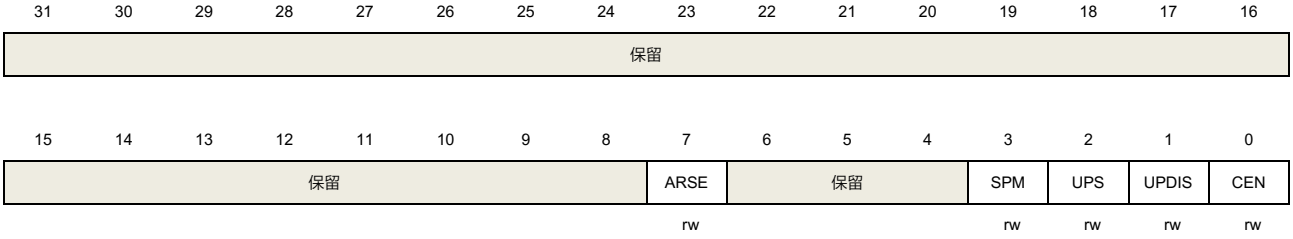
TIMER50 基地址：0x4000 EC00

TIMER51 基地址：0x4000 F000

控制寄存器 0(TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值
3	SPM	单脉冲模式 0：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。

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

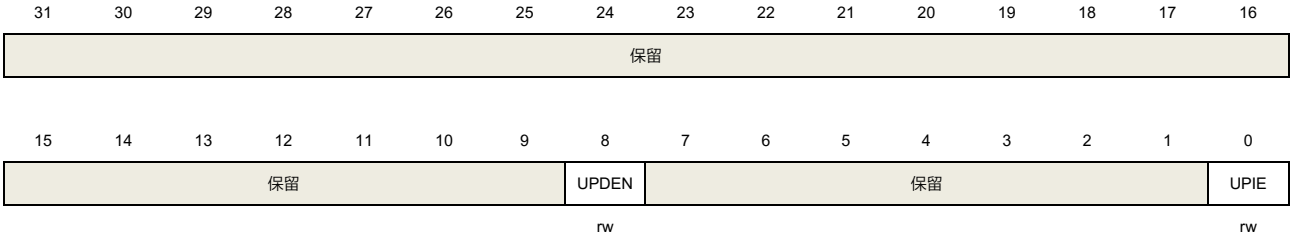
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									MMC[2:0]			保留			
									rw						

位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：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 TRGO 脉冲，后一种情况下，TRGO 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 TRGO。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 TRGO 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 TRGO。 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。

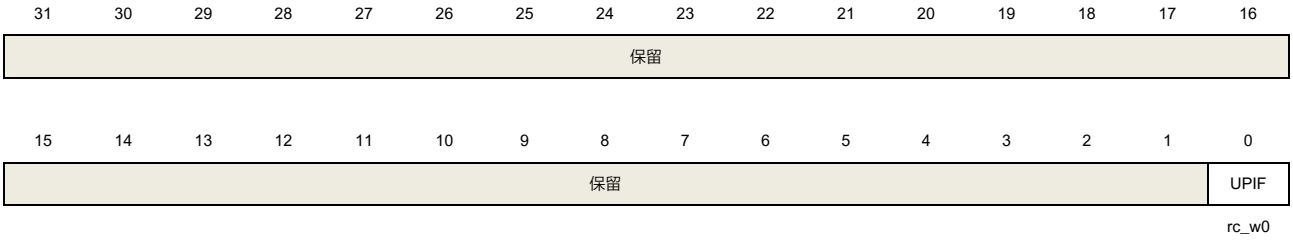


位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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。



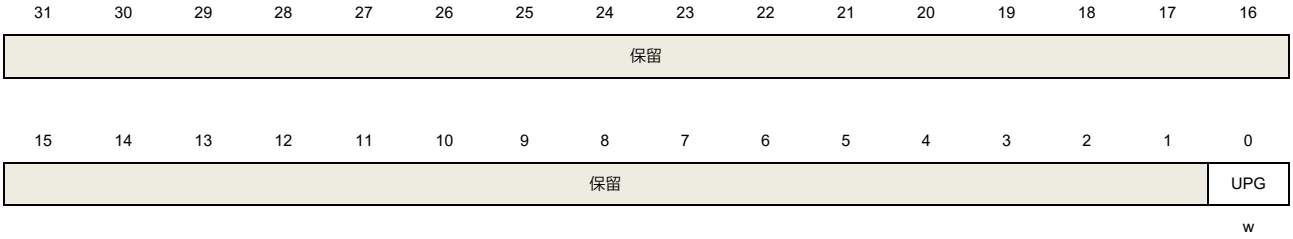
位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生

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。

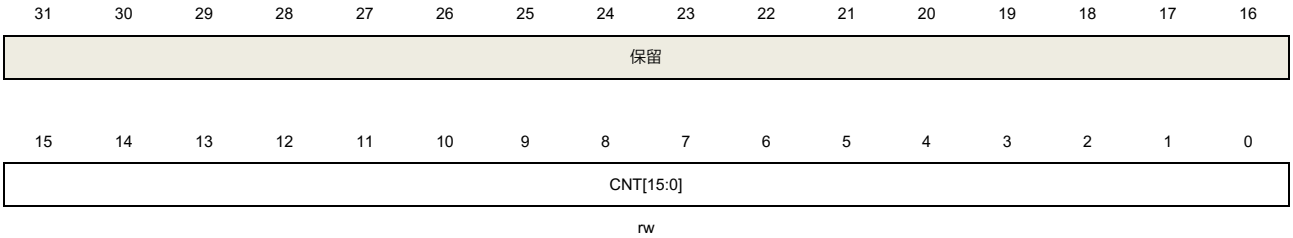


位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

计数器寄存器(TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器(TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

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

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

计数器自动重载寄存器(TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 FFFF

该寄存器只能按字(32 位)访问。

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

24. 通用同步异步收发器(USART)

24.1. 通用同步异步收发器(USARTx/UARTx, x=0..4)

24.1.1. 简介

通用同步异步收发器(USART)提供了一个灵活方便的串行数据交换接口，数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART 提供了可编程的波特率发生器，能对UCLK(CK_APB1，CK_APB2)进行分频产生 USART 发送和接收所需的特定频率。

USART 不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，以及同步单双工模式。它还支持多处理器通信和 Modem 流控操作(CTS/RTS)。数据帧支持从 LSB 或者 MSB 开始传输。数据位的极性和 TX/RX 引脚都可以灵活配置。

所有 USART/UART 都支持 DMA 功能，以实现高速率的数据通信(除 UART4 以外)。

24.1.2. 主要特性

- NRZ标准格式(Mark/Space)；
- 全双工异步通信；
- 半双工单线通信；
- 可编程的波特率产生器：
  - 由外设时钟分频产生，其中USART0由CK_APB2分频得到，USART1/2和UART3/4由CK_APB1分频得到；
  - 8或16倍过采样；
  - 当时钟频率为280M，过采样为8，最高速度可到35MBits/s；
- 完全可编程的串口特性：
  - 偶校验位，奇校验位，无校验位的生成/检测；
  - 数据位(8或9位)；
  - 产生0.5，1，1.5或者2个停止位；
- 发送器和接收器可分别使能；
- 支持硬件Modem流控操作(CTS/RTS)；
- DMA访问数据缓冲区；
- LIN断开帧的产生和检测；
- 支持红外数据协议(IrDA)；
- 串口TX可配置为红外调制输出，支持可调节的调制信号占空比和可选择的调制极性(仅USART2支持)；
- 同步传输模式以及为同步传输输出发送时钟；
- 支持兼容ISO7816-3的智能卡接口：

- 字节模式(T=0)；
- 块模式(T=1)；
- 直接和反向转换；

■ 多处理器通信：
- 如果地址不匹配，则进入静默模式；
- 通过线路空闲检测或者地址匹配检测从静默模式唤醒；

■ 多种状态标志：
- 传输检测标志：接收缓冲区不为空(RBNE)，发送缓冲区为空(TBE)，传输完成(TC)，忙(BSY)；
- 错误检测标志：过载错误(ORERR)，噪声错误(NERR)，帧格式错误(FERR)，奇偶校验错误(PERR)；
- 硬件流控操作标志：CTS变化(CTSF)；
- LIN模式标志：LIN断开检测(LBDF)；
- 多处理器通信模式标志：IDLE帧检测(IDLEF)；
- 智能卡模式标志：块结束(EBF)和接收超时(RTF)；
- 若相应的中断使能，这些事件发生将会触发中断。

USART0/1/2完全实现上述功能，但是UART3/4只实现了上面所介绍功能的部分，下面这些功能在UART3/4中没有实现：

■ 智能卡模式；
■ 同步模式；
■ 硬件流操作(CTS/RTS)；
■ 设置数据极性。

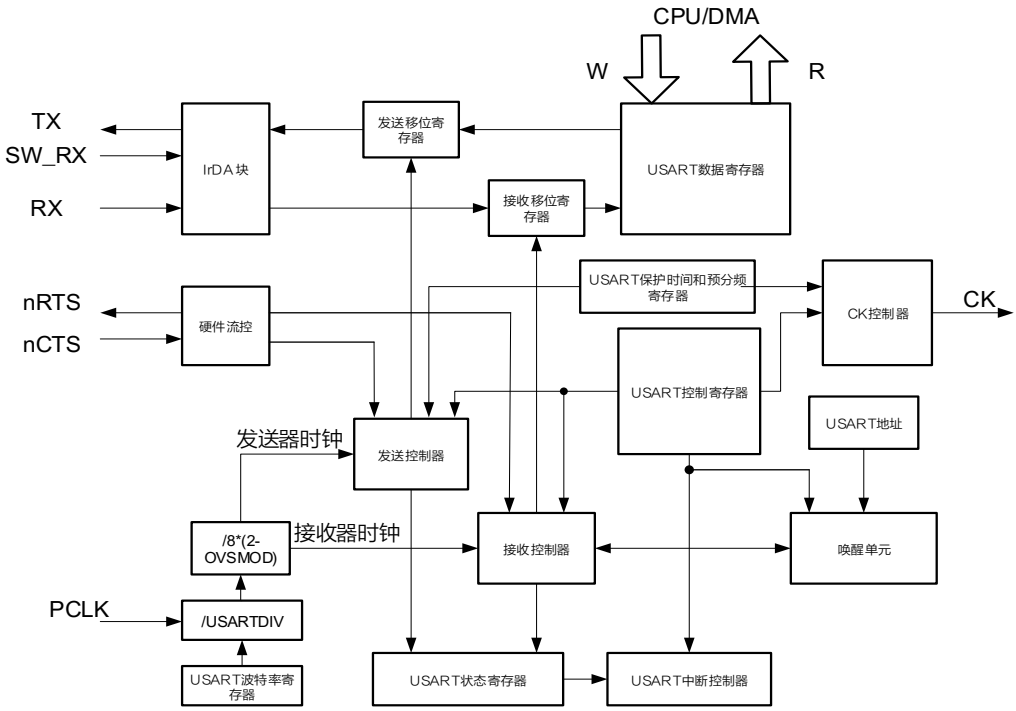
24.1.3. 功能描述

USART 接口通过*表 24-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 24-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O (单线模式/智能卡模式)	发送数据。当 USART 使能后，若无数据发送，默认为高电平
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

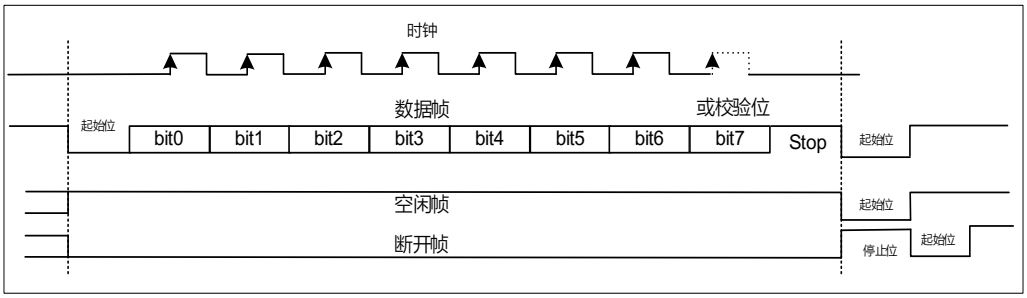
图 24-1. USART 模块内部框图



USART 帧格式

USART 数据帧开始于起始位，结束于停止位。USART_CTL0 寄存器中 WL 位可以设置数据长度。将 USART_CTL0 寄存器中 PCEN 置位，最后一个数据位可以用作校验位。若 WL 位为 0，第七位为校验位。若 WL 位置 1，第八位为校验位。USART_CTL0 寄存器中 PM 位用于选择校验位的计算方法。

图 24-2. USART 字符帧（8 数据位和 1 停止位)



在发送和接收中，停止位可以由 USART_CTL1 寄存器中 STB[1:0]位域配置。

表 24-2. 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

STB[1:0]	停止位长度(位)	功能描述
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，所有位都为 1。数据帧长度与正常 USART 数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART 数据帧的传输速度由 UCLK 时钟频率，波特率发生器的配置，以及过采样模式共同决定。

波特率发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数(USARTDIV)与系统时钟 UCLK 具有如下关系：

如果过采样率是 16，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \tag{24-1}$$

当过采样率是 8 时，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \tag{24-2}$$

例如，当过采样是 16：

1. 由USART_BAUD寄存器的值得到USARTDIV：

假设 USART_BAUD=0x21D，则 INTDIV=33(0x21)，FRADIV=13(0xD)。

UASRTDIV=33+13/16=33.81。

2. 由USARTDIV得到USART_BAUD寄存器的值：

假设要求 UASRTDIV=30.37，INTDIV=30(0x1E)

16*0.37=5.92，接近整数 6，所以 FRADIV=6(0x6)

USART_BAUD=0x1E6。

注意：若取整后 FRADIV=16(溢出)，则进位必须加到整数部分。

USART 发送器

如果 USART_CTL0 寄存器的发送使能位(TEN)被置位，当发送数据缓冲区不为空时，发送器将会通过 TX 引脚发送数据帧。TX 引脚的极性可以通过 USART_CTL3 寄存器中 TINV 位来配置。时钟脉冲通过 CK 引脚输出。

TEN 置位后发送器会发出一个空闲帧。TEN 位在数据发送过程中是不可以被复位的。

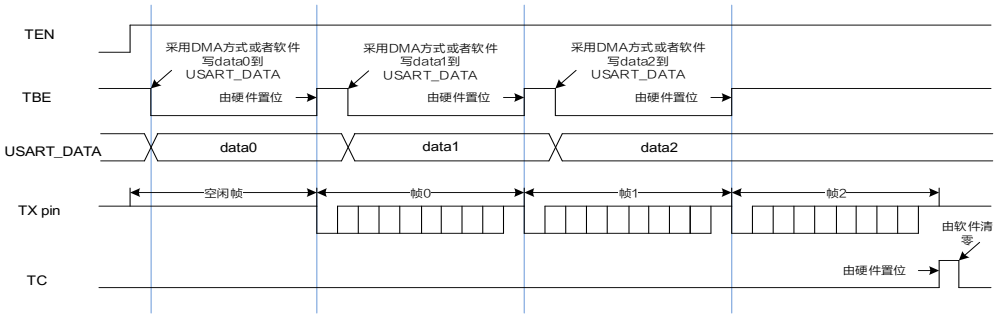
系统上电后，TBE 默认为高电平。在 USART_STAT0 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_DATA 寄存器。当数据写入 USART_DATA 寄存器，TBE 位将被清 0。在数据由 USART_DATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_DATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_DATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

假如一帧数据已经发送出去，并且 TBE 位已经置位，那么 USART_STAT0 寄存器中 TC 位将被置 1。如果 USART_CTL0 寄存器中的中断使能位(TCIE)为 1，将会产生中断。

*图 24-3. USART 发送步骤*给出了 USART 发送步骤。软件操作按以下流程进行：

1. 在 USART_CTL0 寄存器中置位 UEN 位，使能 USART；
2. 通过 USART_CTL0 寄存器的 WL 设置字长；
3. 在 USART_CTL1 寄存器中写 STB[1:0]位来设置停止位的长度；
4. 如果选择了多级缓存通信方式，应该在 USART_CTL2 寄存器中使能 DMA(DENT 位)；
5. 在 USART_BAUD 寄存器中设置波特率；
6. 在 USART_CTL0 寄存器中设置 TEN 位；
7. 等待 TBE 置位；
8. 向 USART_DATA 寄存器写数据；
9. 若 DMA 未使能，每发送一个字节都需重复步骤 7-8；
10. 等待 TC=1，发送完成。

图 24-3. USART 发送步骤



在禁用 USART 或进入低功耗状态之前，必须等待 TC 置位。先读 USART_STAT0 然后再写 USART_DATA 可将 TC 位清 0。在多级缓存通信方式(DENT=1)下，直接向 TC 写 0，也能清 TC。

USART 接收器

上电后，USART 接收器使能按以下步骤进行：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 写USART_CTL0寄存器的WL去设置字长；
3. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；

4. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA(DENR位)；
5. 在USART_BAUD寄存器中设置波特率；
6. 在USART_CTL0中设置REN位。

接收器在使能后若检测到一个有效的起始脉冲便开始接收码流。在接收一个数据帧的过程中会检测噪声错误，奇偶校验错误，帧错误和过载错误。

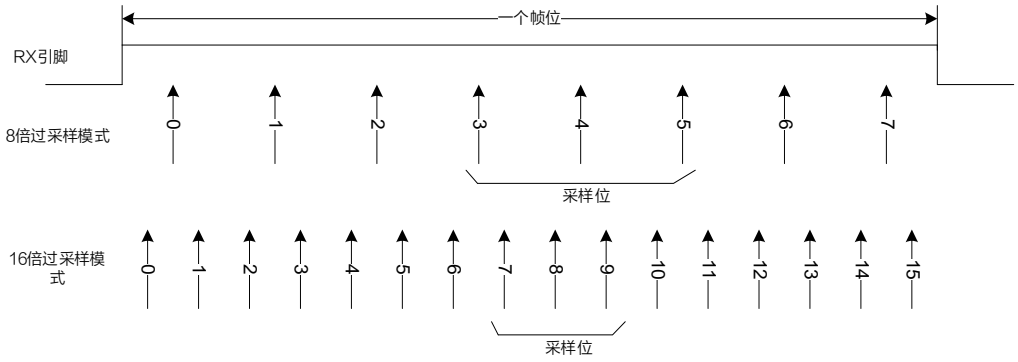
当接收到一个数据帧，USART_STAT0 寄存器中的 RBNE 置位，如果设置了 USART_CTL0 寄存器中相应的中断使能位 RBNEIE，将会产生中断。在 USART_STAT0 寄存器中可以观察接收状态标志。

软件可以通过读 USART_DATA 寄存器或者 DMA 方式获取接收到的数据。不管是直接读寄存器还是通过 DMA，只要是对 USART_DATA 寄存器的一个读操作都可以清除 RBNE 位。

在接收过程中，需使能 REN 位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是 8 倍过采样模式，选择第 3、4、5 个采样点；如果是 16 倍过采样模式，选择第 7、8、9 个采样点。如果在 3 个采样点中有 2 个或 3 个为 0，该数据位被视为 0，否则为 1。如果 3 个采样点中有一个采样点的值与其他两个不同，不管是数据位，奇偶校验位或者停止位，都将产生噪声错误(NERR)。如果使能 DMA，并置位 USART_CTL2 寄存器中 ERRIE，将会产生中断。如果在 USART_CTL2 中置位 OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 24-4. 过采样方式接收一个数据位(OSB=0)



通过置位 USART_CTL0 寄存器中的 PCEN 位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT0 寄存器中 PERR 被置位。如果设置了 USART_CTL0 寄存器中的 PERRIE 位，将产生中断。

如果在停止位传输过程中 RX 引脚为 0，将产生帧错误，USART_STAT0 寄存器中 FERR 置位。如果使能 DMA 并置位 USART_CTL2 寄存器中 ERRIE 位，将产生中断。

当接收到一帧数据，而 RBNE 位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART_STAT0 寄存器中的溢出错误标志位 ORERR 将置位。如果使能 DMA 并置位 USART_CTL2 寄存器中 ERRIE 位或者置位 RBNEIE，将产生中断。

若接收过程中，产生了噪声错误(NERR)、校验错误(PERR)、帧错误(FERR)或溢出错误(ORERR)，则 NERR、PERR、FERR 或 ORERR 将和 RBNE 同时置位。如果没有使能 DMA，

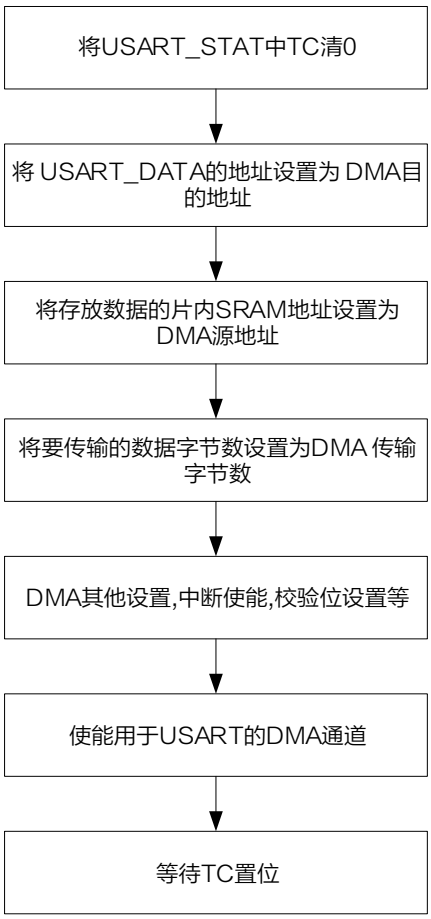
RBNE 中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用 DMA 访问发送缓冲区或者接收缓冲区。置位 USART_CTL2 寄存器中 DENT 位可以使能 DMA 发送，置位 USART_CTL2 寄存器中 DENR 位可以使能 DMA 接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如*图 24-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

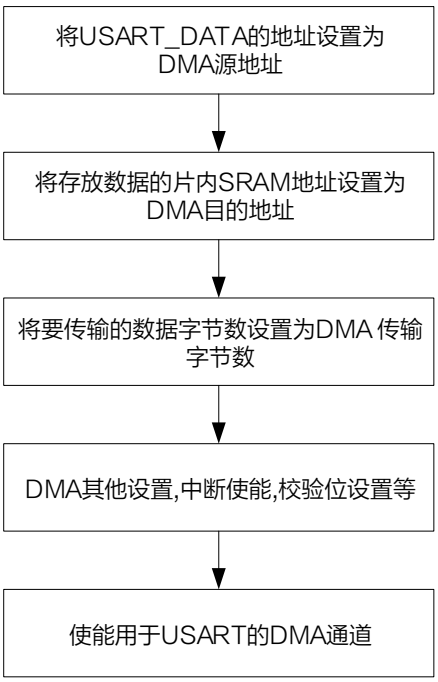
图 24-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_STAT0 寄存器中 TC 位置 1。如果 USART_CTL0 寄存器中 TCIE 置位，将产生中断。

当 DMA 用于 USART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如*图 24-6. 采用 DMA 方式实现 USART 数据接收配置步骤*所示。如果将 USART_CTL2 寄存器中 ERRIE 位置 1，USART_STAT0 寄存器中的错误标志位(FERR、ORERR 和 NERR)被置位时将产生中断。

图 24-6. 采用 DMA 方式实现 USART 数据接收配置步骤

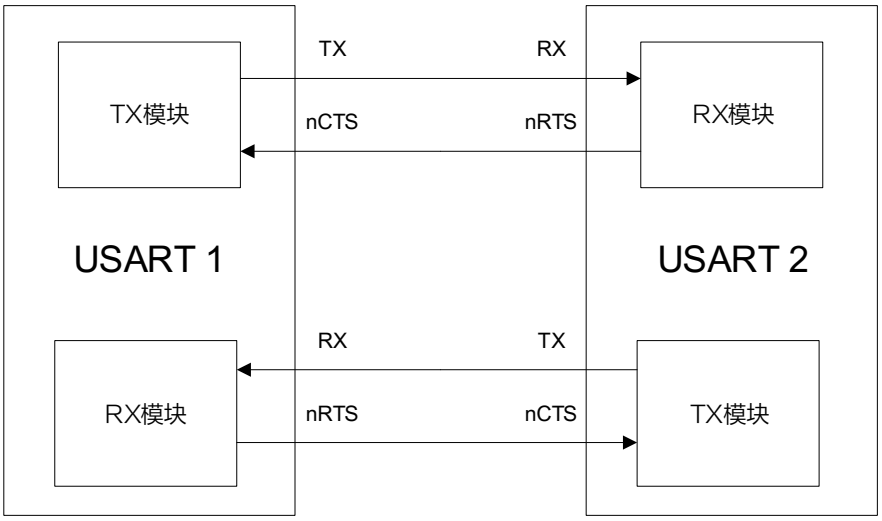


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

硬件流控制

硬件流控制功能通过 nCTS 和 nRTS 引脚来实现。通过将 USART_CTL2 寄存器中 RTSEN 位置 1 来使能 RTS 流控，将 USART_CTL2 寄存器中 CTSEN 位置 1 来使能 CTS 流控。

图 24-7. 两个 USART 之间的硬件流控制



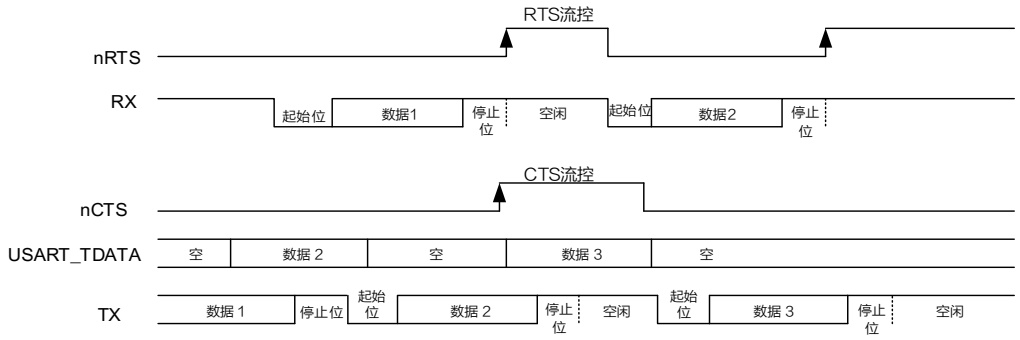
RTS 流控

USART 接收器输出 nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS 变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS 保持高电平，可以通过读 USART_DATA 寄存器来清零。

CTS 流控

USART 发送器监视 nCTS 输入引脚来决定数据帧是否可以发送。如果 USART_STAT0 寄存器中 TBE 位是 0 且 nCTS 为低电平，发送器发送数据帧。在发送期间，若 nCTS 信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 24-8. 硬件流控制



如果 CTS 流控制被使能，在 nCTS 引脚信号发生变化时，USART_STAT0 寄存器中 CTSF 位会置 1。如果 USART_CTL2 寄存器中的 CTSIE 位被置位，将会产生中断。

多处理器通信

在多处理器通信中，多个 USART 被连接成一个网络。对于一个设备来说，监视所有来自 RX 引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将 USART_CTL0 寄存器中 RWU 位置 1 使一个 USART 进入静默模式。

如果 USART 处于静默模式，所有的接收状态标志位将不会被置位。软件可以通过对 RWU 清零来唤醒 USART。

此外，USART 可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒 USART。当在 RX 引脚检测到空闲帧时，硬件会将 RWU 清零，从而退出静默模式，但 USART_STAT0 寄存器中 IDLEF 位不会被置 1。

当 USART_CTL0 寄存器中 WM 被置位，数据最高位会被认为是地址标志位。如果地址标志位为 1，该字节被认为是地址字节。如果地址字节的低 4 位与 USART_CTL1 寄存器中的 ADDR[3:0] 相同，硬件会将 RWU 清零，并退出静默模式。接收到将 USART 唤醒的数据帧，RBNE 将置位。状态标志可以从 USART_STAT0 寄存器中获取。如果地址字节的低 4 位与 USART_CTL1 寄存器中的 ADDR[3:0]不相同，硬件会置位 RWU 并进入静默模式。在这种情况下，RBNE 不会被置位。

如果采用地址掩码检测，默认情况下，接收器对地址字节不做奇偶校验。如果 USART_CHC 寄

存器中 PCEN 位被置位，地址字节最高位被视为校验位，其余位被视为地址。

LIN 模式

将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互联网络模式。

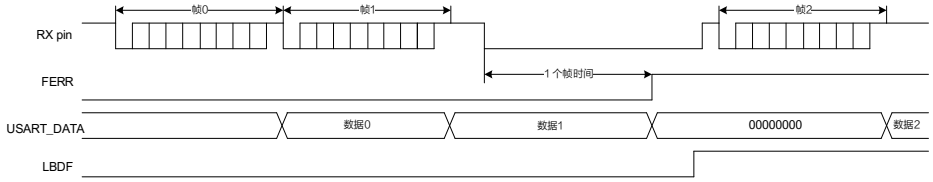
在 LIN 模式下，USART_CTL1 寄存器中 CKEN，WL，STB[1:0]以及 USART_CTL2 的 SCEN，HDEN，IREN 位都应该被清 0。

在发送一个普通数据帧时，LIN 发送过程与普通发送过程相同。当 USART_CTL0 寄存器中 SBKCMD 置位时，USART 在发送完一个停止位后会连续发送 13 个 0。

断开检测功能完全独立于普通 USART 接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART_CTL1 寄存器中 LBLEN 位可以选择断开帧长度。如果在 RX 引脚检测到大于或等于与预期断开帧长度相等数量的 0(LBLEN=0 时，10 个 0；LBLEN=1 时，11 个 0)，USART_STAT0 寄存器中 LBDF 置位。如果 USART_CTL1 寄存器中 LBDIE 被置位，将产生中断。

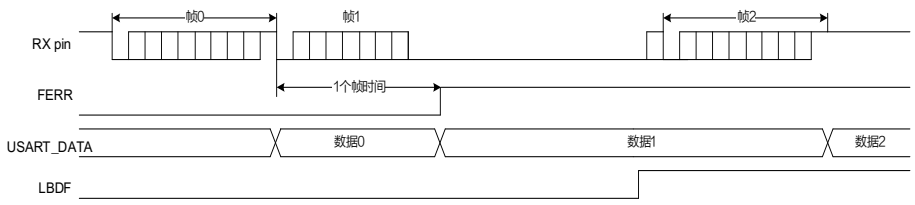
如*图 24-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 24-9. 空闲状态下检测断开帧



如*图 24-10. 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 24-10. 数据传输过程中检测断开帧



同步通信模式

USART 支持主机模式下的全双工同步串行通信，可以通过置位 USART_CTL1 的 CKEN 位来使能。在同步模式下，USART_CTL1 的 LMEN 和 USART_CTL2 的 SCEN，HDEN，IREN 位应该被清 0。CK 引脚作为 USART 同步发送器的时钟输出，仅仅当 TEN 位被使能时，它才被激活。在起始位和停止位传送期间，不会从 CK 引脚输出时钟脉冲。USART_CTL1 的 CLEN 位用来决定在最低位(地址索引位)发送期间是否有时钟信号输出。USART_CTL1 的 CPH 位用

来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1 的 CPL 位用来决定在 USART 同步模式空闲状态下，时钟引脚的电平。

CK 引脚输出波形由 USART_CTL1 寄存器中 CPL，CPH，CLEN 位决定。软件仅在 USART 禁用(UEN=0)时才可以改变它们的值。

如果 USART_CTL0 寄存器中 REN 置位，接收器的工作方式与普通模式下接收方式是不同的。接收器在时钟捕获沿采样数据，并无任何过采样。

图 24-11. 同步模式下的 USART 示例

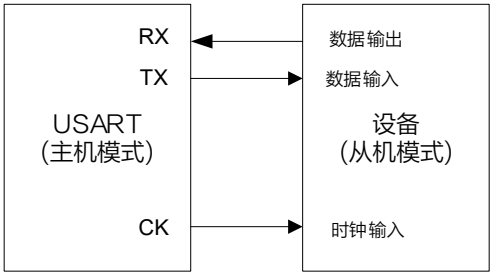
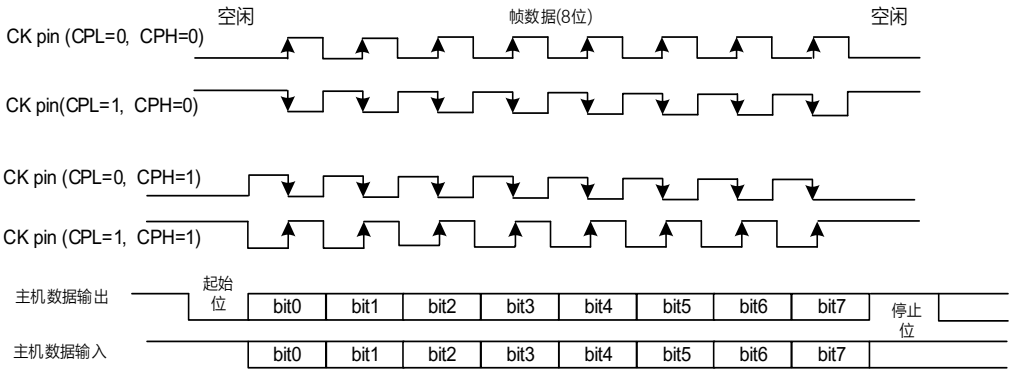


图 24-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)

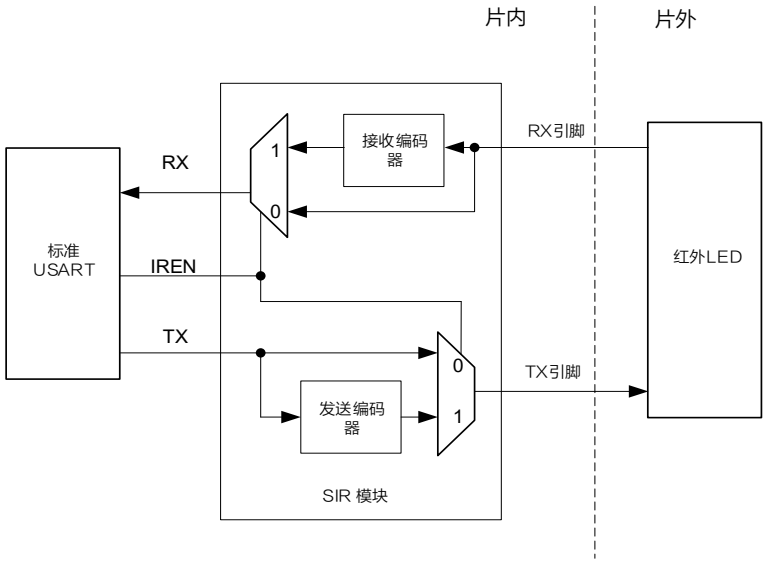


串行红外(IrDA SIR)编解码功能模块

串行红外编解码功能通过置位 USART_CTL2 寄存器中 IREN 使能。在 IrDA 模式下，USART_CTL1 寄存器的 LMEN，STB[1:0]，CKEN 位和 USART_CTL2 寄存器的 HDEN，SCEN 位将被清 0。

在 IrDA 模式下，USART 数据帧由 SIR 发送正交译码器进行调制，调制后的信号经由红外 LED 进行发送，经解调后将数据发送至 USART 接收器。对于正交译码器而言，波特率应小于 115200。

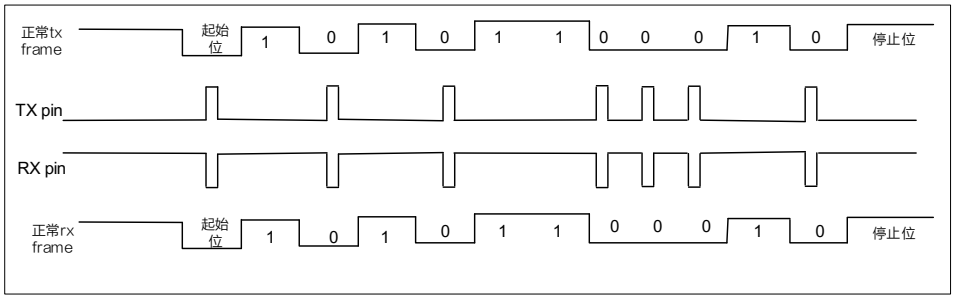
图 24-13. IrDA SIR ENDEC 模块



在 IrDA 模式下，TX 引脚电平与 RX 引脚不同。TX 引脚通常为低电平，RX 引脚通常为高电平。IrDA 引脚电平保持稳定代表逻辑'1'，红外光源脉冲(RTZ 信号)代表逻辑'0'。其脉冲宽度通常占一个位时间的 3/16。IrDA 无法检测到宽度小于一个 1 个 PSC 时钟的脉冲。如果脉冲宽度大于 1 但是小于 2 倍 PSC 时钟，IrDA 则无法可靠的检测到。

由于 IrDA 是一种半双工协议，因此在 IrDA SIR ENDEC 模块中，发送和接收不得同时进行。

图 24-14. IrDA 数据调制



将 USART_CTL2 寄存器中 IRLP 置位可以使 SIR 子模块工作在低功耗模式下。发送编码器由 PCLK 分频得到的低速时钟来驱动。分频系数在 USART_GP 寄存器中 PSC[7:0]位配置，USART_BAUD 寄存器需配置为 16*PSC[7:0]。TX 引脚脉冲宽度可以为低功耗波特率的 3 倍。接收器解码器工作模式与正常 IrDA 模式相同。

半双工通信模式

通过设置 USART_CTL2 寄存器的 HDEN 位，可以使能半双工模式。

在半双工通信模式下，USART_CTL1 寄存器的 LMEN，CKEN 位和 USART_CTL2 寄存器的 SCEN，IREN 位清零。

半双工模式下仅用单线通信，TX 引脚和 RX 引脚从内部连接到一起，RX 引脚不再使用。TX 引脚应被配置为开漏模式，通信冲突由软件处理。

智能卡(ISO7816-3)模式

智能卡模式是一种异步通信模式，支持 ISO7816-3 协议。支持字节模式(T=0)和块模式(T=1)。将 USART_CTL2 寄存器的 SCEN 位置 1，即可使能智能卡模式。在智能卡模式下，USART_CTL1 寄存器的 LMEN 位和 USART_CTL2 的 HDEN，IREN 位应该清 0。

如果 CKEN 位被置位，USART 通过 CK 引脚向智能卡提供一个由 PCLK 分频得到的时钟。分频系数可在 USART_GP 寄存器中 PSC[4:0]配置。CK 引脚只为智能卡提供时钟源。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX 引脚需要被设置成开漏模式，外接上拉电阻，这个引脚将会与智能卡驱动同一条双向连线。智能卡模式下的帧格式为：1 起始位+9 数据位(包括 1 奇偶校验位)+1.5 停止位。其中 0.5 个停止位被配置为接收器的停止位。

图 24-15. ISO7816-3 数据帧格式



字节模式(T=0)

相较于正常操作模式下的时序，从发送移位寄存器到 TX 引脚的传递时间延迟了半个波特率时钟，并且 TC 标志的置位将根据 USART_GP 寄存器的 GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为 ISO7816-3 协议的 CGT 减 12。在保护时间寄存器向上计数这段时间 TC 将被强制拉低，当计数达到设定值时，TC 被置位。

在 USART 发送期间，如果检测到有奇偶校验错误，TX 引脚在停止位最后一个位时间内被拉低，智能卡发送一个 NACK 信号。根据协议，USART 会自动重发 SCRTNUM 次。在重发数据帧前面会插入 2 位的帧间隔。最后一次重发字节后，TC 会立即被置位。如果在最大重发次数后仍然收到 NACK 信号，USART 将会停止发送，帧错误标志被置位。USART 不会将 NACK 信号作为起始位。

在 USART 接收期间，如果在当前数据帧检测到校验错误，TX 引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到 NACK 信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE 中断和接收 DMA 请求都不会被激活。根据协议，智能卡将要重新发送数据。如果在最大的重新发送次数后(这个次数的具体值在 SCRTNUM 位域)，接收到的字符仍然是错误的，USART 停止发送 NACK 信号和标注这个错误为奇偶校验错误。将 USART_CTL2 寄存器中的 NKEN 置位可以使能 NACK 信号。

空闲帧和断开帧在智能卡模式下不适用。

块模式(T=1)

在 T=1(块模式)下，USART_CTL2 寄存器的 NKEN 位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将 USART_RT 寄存器设置成 BWT(块等待)-11 的值并将 RBNEIE 置位。这个超时时间体现在波特时间单元。如果这个时间到了，还没有从智能卡收到应答，USART_STAT1 寄存器中 RTF 位被置位。如果设置了 USART_CTL3 寄存器中 RTIE 位，将会产生中断。如果在超时之前收到了第一个字节，则会引起 RBNE 中断。如果用 DMA 从智能卡读取数据，也只能在第一个字节接收好后再去使能 DMA。

第一个字节接收到后，RT[23:0]的值设置成 CWT(字节等待时间)-11 来使能两个连续字节间最大帧间隔自动校验。如果在 RT[23:0]周期内智能卡停止发送字节，USART_STAT1 寄存器中 RTF 将被置位。

USART 用一个块长度计数器统计收到的字节数，这个计数器在 USART 开始发送的时候自动清 0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)，这个值必须写入 USART_RT 寄存器 BL[7:0]。块长度计数器从 0 开始计数到最大值 BL[7:0]+4。在块计数器计数到最大值时，USART_STAT1 寄存器中块结束状态标志位 EBF 置位。如果设置了 USART_CTL3 寄存器中的 EBIE 位，将会产生中断。如果块长度发生错误，RTF 置位。

当使用 DMA 模式接收时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以从接收缓冲区读取第三个字节作为块长度。

如果接收时不使用 DMA 方式，为避免产生 EBF 状态标志，BL[7:0]需首先配置为最大值 0xFF。在收到第三个字节后，真正的块长度值可以重新写入到 BL[7:0]。

直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX 引脚高电平代表逻辑'1'，偶校验。在这种情况下，USART_CTL3 寄存器中 MSBF 位和 DINV 位都为 0(默认值)。

如果选择反向转换，从数据帧的最高位开始传输，TX 引脚高电平代表逻辑'0'，偶校验。在这种情况下，USART_CTL3 寄存器中 MSBF 位和 DINV 位都为 1。

USART 中断

USART 中断事件和标志如*表 24-3. USART 中断请求*所示：

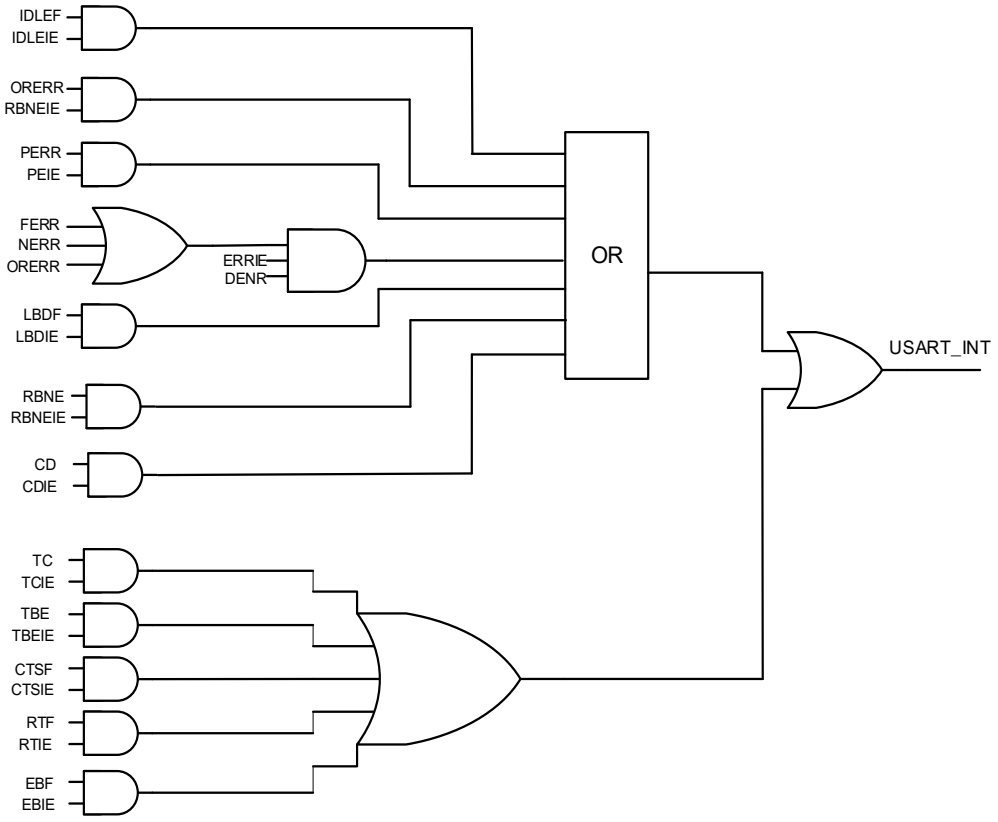
表 24-3. USART 中断请求

中断事件	事件标志	控制寄存器	使能控制位
发送数据寄存器空	TBE	USART_CTL0	TBEIE
CTS 标志	CTSF	USART_CTL2	CTSIE
发送结束	TC	USART_CTL0	TCIE
接收到的数据可以读取	RBNE	USART_CTL0	RBNEIE
检测到过载错误	ORERR		

中断事件	事件标志	控制寄存器	使能控制位
检测到线路空闲	IDLEF	USART_CTL0	IDLEIE
奇偶校验错误	PERR	USART_CTL0	PERRIE
LIN 模式下，检测到断开标志	LBDF	USART_CTL1	LBDIE
接收超时错误	RTF	USART_CTL3	RTIE
发现块尾	EBF	USART_CTL3	EBIE
当 DMA 接收使能时，接收错误(噪声错误、溢出错误、帧错误)	NERR 或 ORERR 或 FERR	USART_CTL2	ERRIE

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 24-16. USART 中断映射框图



红外调制

红外调制是一种通过将数据信号调制到载波上实现红外传输的技术。USART2 的 TX 端支持红外调制输出功能，可以通过配置 IMEN(红外调制使能)位来激活该功能。在使用 USART2 的红外调制功能时，可以灵活选择调制信号的极性，并根据实际需求调整红外调制信号的占空比。

注意：当采用 8 倍过采样时，波特率配置必须满足：

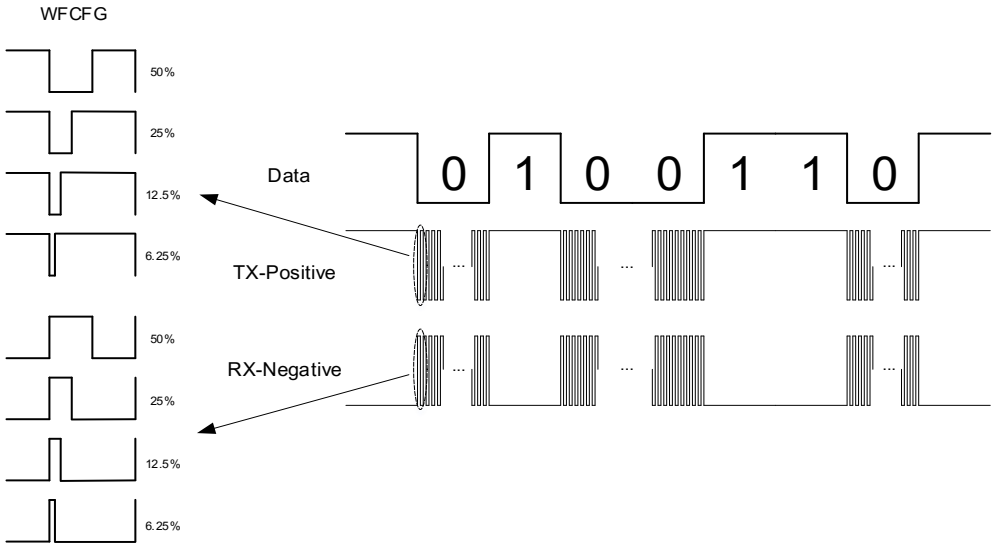
$$USART_BAUD >= CFDIV * 16 * 4 \quad (24\text{-}3)$$

当采用 16 倍过采样时：

$$USART_BAUD >= CFDIV * 16 * 2 \quad (24\text{-}4)$$

否则可能无法在一个载波周期内输出有效数量。

图 24-17.红外调制波形示意图



24.1.4. USART 寄存器

USART0 基地址：0x4001 3800

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

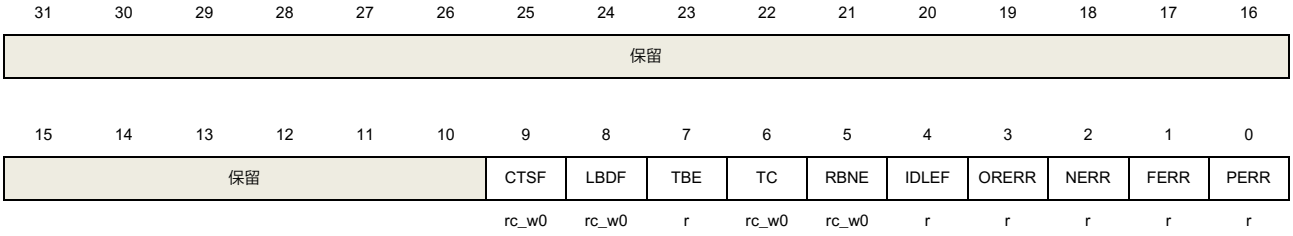
UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

状态寄存器 0 (USART_STAT0)

地址偏移：0x00
复位值：0x0000 00C0

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CTSF	CTS 变化标志。 如果设置了 USART_CTL2 寄存器中 CTSEN 位，当 nCTS 输入变化时，该位由硬件置位。如果设置了 USART_CTL2 寄存器中 CTSIE 位，将产生中断。 该位由软件清 0。 0：nCTS 状态线没有变化。 1：nCTS 状态线发生变化。 该位对 UART3/4 无效。
8	LBDF	LIN 断开检测标志。 寄存器 USART_CTL1 寄存器中 LMEN 置位，说明检测到 LIN 断开。如果 USART_CTL1 寄存器中 LBDIE 被置位时，将产生中断。 该位由软件清 0。 0：没有检测到 LIN 断开字符。 1：检测到 LIN 断开字符。
7	TBE	发送数据缓冲区空 上电复位或待发送数据已发送至移位寄存器后，该位置 1。USART_CTL0 寄存器中 TBEIE 被置位将产生中断。 该位在软件将待发送数据写入 USART_DATA 时被清 0。

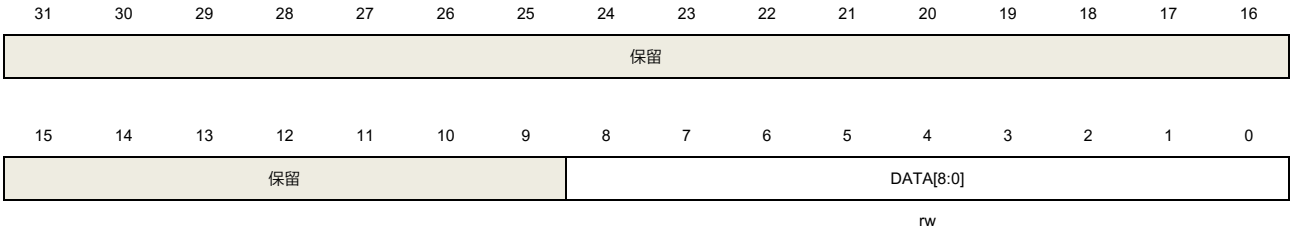
		0：发送数据缓冲区不为空。 1：发送数据缓冲区空。
6	TC	发送完成 上电复位后，该位被置 1。如果 TBE 置位，在当前数据发送完成时该位置 1。USART_CTL0 寄存器中 TCIE 被置位将产生中断。 该位由软件清 0。 0：发送没有完成。 1：发送完成。
5	RBNE	读数据缓冲区非空。 当读数据缓冲区接收到来自移位寄存器的数据时，该位置 1。当寄存器 USART_CTL0 的 RBNEIE 位被置位，将会有中断产生。 软件可以通过对该位写 0 或读 USART_DATA 寄存器来将该位清 0。 0：读数据缓冲区为空。 1：读数据缓冲区不为空。
4	IDLEF	空闲线检测标志。 在一个帧时间内，在 RX 引脚检测到空闲状态，该位置 1。当寄存器 USART_CTL0 的 IDLEIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：未检测到空闲帧。 1：检测到空闲帧。
3	ORERR	溢出错误。 在 RBNE 置位的情况下，如果 USART_DATA 寄存器接收到来自移位寄存器的数据，该位置 1。当寄存器 USART_CTL2 的 ERRIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：没有检测到溢出错误。 1：检测到溢出错误。
2	NERR	噪声错误标志。 将 USART_CTL2 寄存器中 OSB 清 0，在接收数据时，如果在 RX 引脚检测到噪声，该位被置位。当寄存器 USART_CTL2 的 ERRIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：没检测到噪声错误。 1：检测到噪声错误。
1	FERR	帧错误。 接收数据期间，在停止位传输过程中，RX 引脚检测到低电平，该位被置位。当寄存器 USART_CTL2 的 ERRIE 位被置位，将会有中断产生。 软件先读 USART_STAT0，再读 USART_DATA 可清除该位。 0：未检测到帧错误。 1：检测到帧错误。
0	PERR	校验错误。 当接收到的数据帧校验位与预期校验值不同时，该位置位。

软件先读 USART_STAT0，再读或者写 USART_DATA 可清除该位。
0：没检测到校验错误。
1：检测到校验错误。

数据寄存器 (USART_DATA)

地址偏移：0x04
复位值：未定义

该寄存器只能按字(32 位)访问。



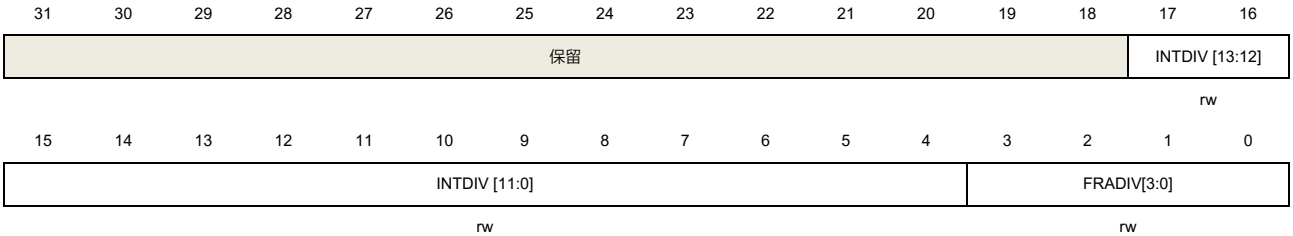
位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	DATA[8:0]	发送或接收的数据值。 软件可以通过写这些位来改变要发送的数据，或读这些位的值来获取接收到的数据。 如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位(第 7 位或第 8 位取决于 USART_CTL0 寄存器的 WL 位)将被校验位取代。

波特率寄存器(USART_BAUD)

地址偏移：0x08
复位值：0x0000 0000

使能 USART(UEN=1)时，不能写该寄存器。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17:4	INTDIV[13:0]	波特率分频器的整数部分。

注意：在 IrDA 低功耗模式下，INTDIV[13:0]需要配置为 PSC 的 16 倍。

3:0	FRADIV [3:0]	波特率分频器的小数部分。 如果启用了 8 倍过采样模式，软件必须保持 FRADIV[3]位为复位状态。

控制寄存器 0 (USART_CTL0)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	保留	UEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	RWU	SBKCMD
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OVSMOD	采样模式 0：16 倍采样 1：8 倍采样 如果 SCEN=1，IREN=1 或者 LMEN=1，OVSMOD 由硬件强制为 0。
14	保留	必须保持复位值。
13	UEN	USART 使能 0：USART 禁用 1：USART 使能
12	WL	字长 0：8 数据位 1：9 数据位
11	WM	从静默模式唤醒方法。 0：空闲线 1：地址掩码
10	PCEN	校验控制使能。 0：校验控制禁用。 1：校验控制使能。
9	PM	校验模式 0：偶校验 1：奇校验

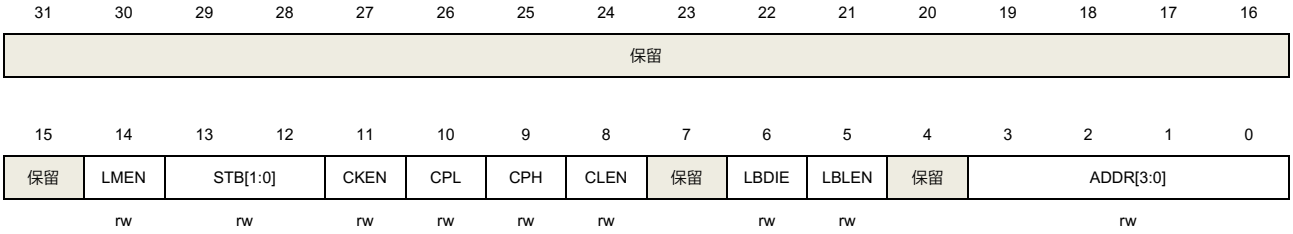
8	PERRIE	校验错误中断使能。 如果该位置 1，USART_STAT0 寄存器中 PERR 被置位时产生中断。 0：校验错误中断禁用。 1：校验错误中断使能。
7	TBEIE	发送缓冲区空中断使能。 如果该位置 1，USART_STAT0 寄存器中 TBE 被置位时产生中断。 0：发送缓冲区空中断禁止。 1：发送缓冲区空中断使能。
6	TCIE	发送完成中断使能。 如果该位置 1，USART_STAT0 寄存器中 TC 被置位时产生中断。 0：发送完成中断禁用。 1：发送完成中断使能。
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能。 如果该位置 1，USART_STAT0 寄存器中 RBNE 或 ORERR 被置位时产生中断。 0：读数据缓冲区非空中断和过载错误中断禁用。 1：读数据缓冲区非空中断和过载错误中断使能。
4	IDLEIE	IDLE 线检测中断使能。 如果该位置 1，USART_STAT0 寄存器中 IDLEF 被置位时产生中断。 0：IDLE 线检测中断禁用。 1：IDLE 线检测中断使能。
3	TEN	发送器使能。 0：发送器禁用。 1：发送器使能。
2	REN	接收器使能。 0：接收器禁用。 1：接收器使能。
1	RWU	接收器从静默模式中唤醒。 软件可以通过将该位置 1 使得 USART 进入静默模式，将该位清 0 唤醒 USART。 空闲帧唤醒模式下(WM=0)，当检测到空闲帧时，该位由硬件清 0。地址掩码模式下(WM=1)，当接收到一个地址匹配帧时，该位由硬件清 0；或接收到一个地址非匹配帧时，由硬件置 1。 0：接收器处于正常工作模式。 1：接收器处于静默模式。
0	SBKCMD	发送断开帧。 软件通过发送断开帧将该位置 1。 断开帧传输结束由硬件清 0。 0：没有发送断开帧。 1：发送断开帧。

控制寄存器 1 (USART_CTL1)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14	LMEN	LIN 模式使能。 0：LIN 模式禁用。 1：LIN 模式使能。
13:12	STB[1:0]	STOP 位长 00：1 停止位 01：0.5 停止位 10：2 停止位 11：1.5 停止位 对于 UART3/4，只有 1 位停止位和两位停止位是有效的。
11	CKEN	CK 引脚使能 0：CK 引脚禁用 1：CK 引脚使能 该位对于 UART3/4 无效。
10	CPL	时钟极性 该位用来设定在同步模式下 CK 引脚的极性。 0：CK 引脚不对外发送时保持为低电平。 1：CK 引脚不对外发送时保持为高电。 该位对于 UART3/4 无效。
9	CPH	时钟相位 该位用来设定在同步模式下 CK 引脚的相位。 0：在首个时钟边沿采样第一个数据 1：在第二个时钟边沿采样第一个数据 该位对于 UART3/4 无效。
8	CLEN	CK 信号长度。 该位用来设定在同步模式下 CK 信号的长度。

		0：8 位数据帧中有 7 个 CK 脉冲，9 位数据帧中有 8 个 CK 脉冲。 1：8 位数据帧中有 8 个 CK 脉冲，9 位数据帧中有 9 个 CK 脉冲。 该位对于 UART3/4 无效。
7	保留	必须保持复位值。
6	LBDIE	LIN 断开信号检测中断使能。 如果该位置 1，当 USART_STAT0 寄存器中 LBDF 被置位时将产生中断。 0：断开信号检测中断禁用。 1：断开信号检测中断使能。
5	LBLEN	LIN 断开帧长度 该位用来设定在断开帧长度。 0：10 位 1：11 位
4	保留	必须保持复位值。
3:0	ADDR[3:0]	USART 地址 地址掩码唤醒模式下(WM=1)，如果接收到的数据帧低四位与 ADDR[3:0]值不相等，USART 就会进入静默模式；如果接收到的数据帧低四位与 ADDR[3:0]值相等，USART 就会被唤醒。

控制寄存器 2 (USART_CTL2)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	OSB	一个数据位采样一次的方法。 该位用于选择采样方法。当该位被置位时，USART 对一个数据位取一个采样点，而不是一个数据位取三个采样点。当使用该方法时，噪声错误标志(NERR)需禁用。 0：三次采样方法。 1：一次采样方法。
10	CTSIE	CTS 中断使能。 如果该位置 1，当 USART_STAT0 寄存器中 CTSF 被置位时将产生中断。

		0：CTS 中断禁用。 1：CTS 中断使能。 该位对于 UART3/4 无效。
9	CTSEN	CTS 使能 该位用于使能 CTS 硬件流控制功能。 0：CTS 硬件流控制禁用。 1：CTS 硬件流控制使能。 该位对于 UART3/4 无效。
8	RTSEN	RTS 使能 该位用于使能 RTS 硬件流控制功能。 0：RTS 硬件流控制禁用。 1：RTS 硬件流控制使能。 该位对于 UART3/4 无效。
7	DENT	DMA 发送使能 0：DMA 发送模式禁用。 1：DMA 发送模式使能。
6	DENR	DMA 接收使能 0：DMA 接收模式禁用。 1：DMA 接收模式使能。
5	SCEN	智能卡模式使能 该位用于使能智能卡模式。 0：智能卡模式禁用。 1：智能卡模式使能。 该位对于 UART3/4 无效。
4	NKEN	在智能卡模式 NACK 使能。 该位用于智能卡模式在奇偶校验错误发生时使能 NACK 发送。 0：当出现校验错误时不发送 NACK。 1：当出现校验错误时发送 NACK。 该位对于 UART3/4 无效。
3	HDEN	半双工使能 该位用于使能半双工模式。 0：半双工模式禁用 1：半双工模式使能
2	IRLP	IrDA 低功耗模式。 该位用于为 IrDA 模式选择低功耗模式。 0：正常模式 1：低功耗模式
1	IREN	IrDA 模式使能 0：IrDA 禁用

1：IrDA 使能

0	ERRIE	错误中断使能。 当 DMA 接收模式(DENR=1)使能时，如果该位被置 1，USART_STAT0 寄存器中 FERR，ORERR，NERR 被置位将产生中断。 0：错误中断禁用。 1：错误中断使能。

保护时间和预分频器寄存器 (USART_GP)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



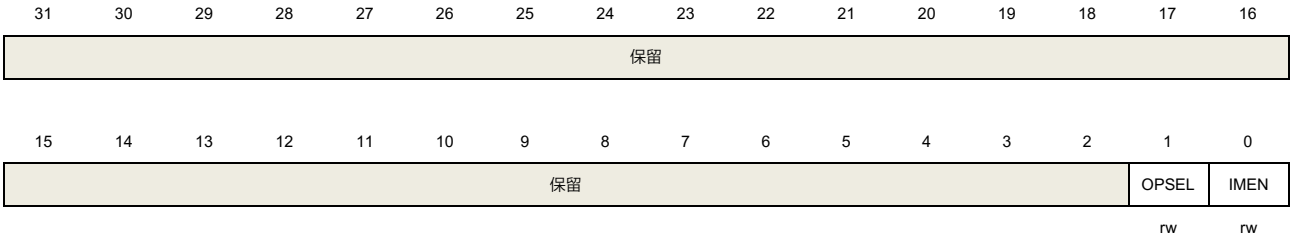
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	智能卡模式下的保护时间值。 TC 标志置位时间延时 GUAT[7:0]个波特时钟周期。 该位对于 UART3/4 无效。
7:0	PSC[7:0]	使能 USART IrDA 低功耗模式，这些位用来设定将外设时钟(CK_APB1/CK_APB2)分频产生低功耗频率的分频系数。 00000000：保留 - 不要写入该值。 00000001：对源时钟 1 分频。 … 11111111：对源时钟 255 分频。 在 IrDA 正常模式下，PSC 只能设置成 00000001。 在智能卡模式下，PSC[4:0]用于设定外设时钟(APB1/APB2)生成智能卡时钟的分频系数。实际的分频系数为 PSC[4:0]设定值的两倍。 00000：保留 - 不要写入该值。 00001：对源时钟 2 分频。 00010：对源时钟 4 分频。 … 11111：对源时钟 62 分频。 在智能卡模式下，PSC[7:5]保留。

红外调制控制寄存器 (USART_IMCTL)

地址偏移：0x20
复位值：0x0000 0000

此寄存器仅适用于 USART2。

该寄存器只能按字(32 位)访问。



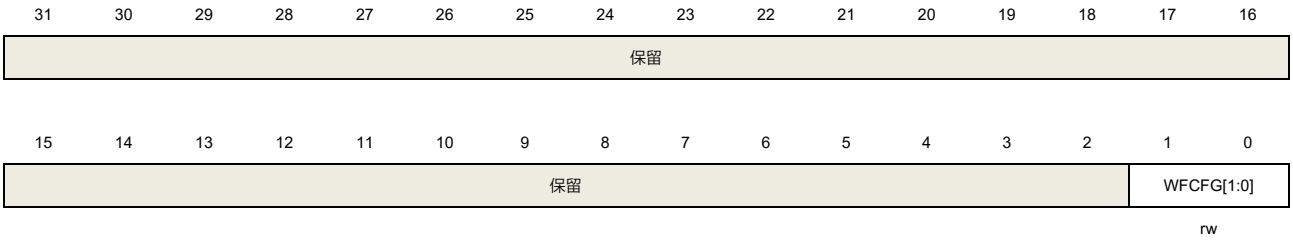
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OPSEL	红外调制输出极性选择 0：正极性 1：负极性
0	IMEN	红外调制功能使能 0：禁用 TX 输出的红外调制功能。 1：启用 TX 输出的红外调制功能。

红外调制脉宽寄存器 (USART_IMPW)

地址偏移：0x24
复位值：0x0000 0000

此寄存器仅适用于 USART2。

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	WFCFG[1:0]	红外调制波形占空比配置 00：50% 01：25% 10：12.5% 11：6.25%

红外调制预分频寄存器 (USART_IMPRE)

地址偏移：0x28
复位值：0x0000 0000

此寄存器仅适用于 USART2。

该寄存器只能按字(32 位)访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CFDIV[7:0]	红外调制时钟频率分频配置 红外调制信号频率为： $F = \frac{UCLK}{(CFDIV[7:0] + 1) * 16}$

控制寄存器 32(USART_CTL3)

偏移地址：0x80
复位值：0x0000 0000

UART3/4 未使用该寄存器

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				MSBF	DINV	TINV	RINV	保留		EBIE	RTIE	SCRTNUM[2:0]			RTEN
				rw	rw	rw	rw			rw	rw	rw			rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	MSBF	高位在前 该位用于设定数据在发送或接收时的顺序。 0：数据发送/接收，采用低位在前。 1：数据发送/接收，采用高位在前。 USART 被使能(UEN=1)时，这一位不能被改写。
10	DINV	数据位反转 该位用于设定在发送或接收时数据位的极性。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART 被使能(UEN=1)时，这一位不能被改写。
9	TINV	TX 引脚电平反转。 该位用于设定 TX 引脚极性。 0：TX 引脚信号值没有反转。 1：TX 引脚信号值被反转。 USART 被使能(UEN=1)时，这一位不能被改写。
8	RINV	RX 引脚电平反转。 该位用于设定 RX 引脚极性。 0：RX 引脚信号值没有反转。 1：RX 引脚信号值被反转。 USART 被使能(UEN=1)时，这一位不能被改写。
7:6	保留	必须保持复位值。
5	EBIE	块结束标志中断使能位 如果该位置 1，USART_STAT1 寄存器中 EBF 被置位时产生中断。 0：块中断禁用。 1：块中断使能。
4	RTIE	接收超时标志中断使能位。 如果该位置 1，USART_STAT1 寄存器中 RTF 被置位时产生中断。 0：接收超时中断禁用。 1：接收超时中断使能。
3:1	SCRTNUM[2:0]	智能卡自动重试次数寄存器。 在智能卡模式下，这些位用来设定在发送和接收时重试的次数。 在发送模式下，一帧数据可以重发 SCRTNUM 次。如果一帧数据发送失败 SCRTNUM+1 次，FERR 被置位。 在接收模式下，USART 接收一个数据帧可以执行 SCRTNUM+1 次。如果一个数据帧校验位不匹配事件产生 SCRTNUM+1 次，RBNE 位和 PERR 位被置位。 当这些位被设置为 0x0 时，在发送模式下这些位将不会自动发送。
0	RTEN	接收器超时使能。

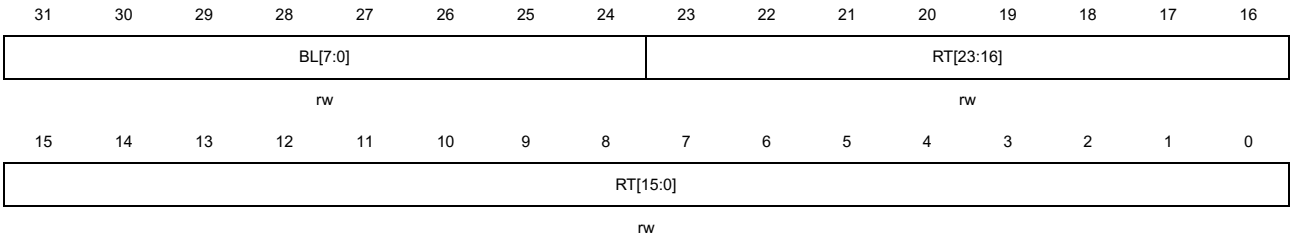
该位用于使能 USART 接收超时。
0：接收器超时检测功能禁用。
1：接收器超时检测功能使能。

接收超时寄存器 (USART_RT)

偏移地址：0x84
复位值：0x0000 0000

UART3/4 未使用该寄存器。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位用于设定智能卡 T=1 的接收时，块的长度。它的值等于信息字节的长度+结束部分的长度(1-LEC/2-CRC) - 1。 这个值可以在块接收开始去设置(用于需要从块的序言提取块的长度的情形)，这个值在每一个接收时钟周期只能设置一次。在智能卡模式下，当 TBE=0 时，块的长度计数器被清 0。 在其他模式下，当 REN=0 (禁用接收器)或者当 USART_STAT1 寄存器的 EBF 位被写 0 时，块的长度计数器被清 0。
23:0	RT[23:0]	接收器超时阈值。 该位域用于指定接收超时值，单位是波特时钟的时长。 标准模式下，如果在最后一个字节接收后，在 RT 规定的时长内，没有检测到新的起始位，USART_STAT1 寄存器中 RTF 标志被置位。 在智能卡模式，这个值被用来实现 CWT 和 BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始算的。 这些位可以在工作时改写。假如一个新数据到来的时间比 RT 规定的晚，RTF 标志会被置位。对于每个接收字符，这个值只能改写一次。

状态寄存器 1 (USART_STAT1)

偏移地址：0x88
复位值：0x0000 0000

UART3/4 未使用该寄存器

该寄存器只能按字(32 位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBF	RTF	保留										
			w0	w0											

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BSY	忙标志 USART 接收一帧数据时被置位。 0：USART 接收通道空闲。 1：USART 接收通道忙。
15:13	保留	必须保持复位值。
12	EBF	块结束标志。 该位在接收字节数(从块开始开始计数，包含序言)等于或者大于 BLEN+4 时被置位。 USART_CTL3 寄存器中 EBIE 被置位将产生中断。 软件可以通过写 0 清除该位。 0：块结束事件没有发生。 1：块结束事件发生。
11	RTF	接收超时标志。 该位在 RX 引脚空闲时间已经超过 RT 值时被置位。USART_CTL3 寄存器中 RTIE 被置位将产生中断。 软件可以通过写 0 清除该位。 0：接收器超时事件没有发生。 1：接收器超时事件发生。
10:0	保留	必须保持复位值。

相干性控制寄存器 (USART_CHC)

偏移地址：0xC0

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

保留	EPERR	保留	BCM	PCM	HCM
	w0c		rw	rw	rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 该标志将在RBNE置位前，校验位被检测到时置位。 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
7:3	保留	必须保持复位值。
2	BCM	断开帧兼容模式 0：当CTS硬件流控制使能时，发送器在发送断开帧之前不会检测nCTS的输入状态 1：当CTS硬件流控制使能时，发送器在发送断开帧之前检测nCTS的输入状态 该位对于UART3/4无效。
1	PCM	校验兼容模式 0：在地址匹配模式唤醒情况下，数据的最高位被用来识别地址，而不是用作校验位。接收器不会对地址数据进行校验(PERR将不会被置位除非发生校验错误) 1：在地址匹配模式唤醒情况下，接收器对地址进行校验，当检测到校验错误时PERR被置位
0	HCM	硬件流控制兼容模式 0：nRTS信号与USART_STAT0寄存器中RBNE位相同 1：nRTS信号在最后一个数据位被采样后被置位 该位对于UART3/4无效。

24.2. 通用同步异步收发器(USARTx, x=5)

24.2.1. 简介

通用同步/异步收发器(USART)提供了一个灵活方便的串行数据交换接口。数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART 提供了可编程的波特率发生器，能对UCLK(CK_USART5)进行分频产生 USART 发送和接收所需的特定频率。

USART 不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡模式，LIN，以及同步和单双工模式。它还支持多处理器通信。数据帧支持从 LSB 或者 MSB 开始传输。数据位的极性和 TX/RX 引脚都可以灵活配置。

所有 USART 都支持 DMA 功能，以实现高速率的数据通信。

24.2.2. 主要特性

- NRZ标准格式；
- 全双工异步通信；
- 半双工单线通信；
- 接收/发送FIFO功能；
- 双时钟域：
  - 互为异步关系的PCLK时钟和USART时钟；
  - 不依赖PCLK设置的波特率设置。
- 可编程的波特率产生器，当时钟频率为280MHz，过采样为8，最高速度可达35MBits/s；
- 完全可编程的串口特性：
  - 数据位(7、8、9或10位)低位或高位在前；
  - 偶校验位，奇校验位，无校验位的生成或检测；
  - 产生0.5，1，1.5或者2个停止位。
- 可互换的Tx/Rx引脚；
- 可配置的数据极性；
- 支持硬件 Modem 流控操作(CTS / RTS)和 RS485 驱动使能；
- 可配置的多级缓存通信DMA访问数据缓冲区；
- 发送器和接收器可分别使能；
- 奇偶校验位控制：
  - 发送奇偶校验位；
  - 检测接收的数据字节的奇偶校验位。
- LIN断开帧的产生和检测；
- 支持红外数据协议(IrDA)；
- 同步传输模式以及为同步传输输出发送时钟；
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式(T=0)；
  - 块模式(T=1)；
  - 直接和反向转换。

- 多处理器通信：
  - 如果地址不匹配，则进入静默模式；
  - 通过线路空闲检测或者地址匹配检测从静默模式唤醒。
- 支持ModBus通信：
  - 超时功能；
  - CR/LF字符识别。
- 从深度睡眠模式唤醒：
  - 通过标准的RBNE中断；
  - 通过WUF中断。
- 多种状态标志：
  - 传输检测标志：接收缓冲区不为空(RBNE)，接收FIFO满(RFF)，接收FIFO为空(RFE)，接收FIFO达到阈值(RFT)，接收FIFO非空(RFNE)，发送缓冲区为空(TBE)，传输完成(TC)，发送FIFO未满(TFNF)，发送FIFO达到阈值(TFT)，发送FIFO为空(TFE)，发送FIFO到达阈值(TFTIF)；
  - 错误检测标志：过载错误(ORERR)，噪声错误(NERR)，帧格式错误(FERR)，奇偶校验错误(PERR)；
  - 硬件流控操作标志：CTS变化(CTSF)；
  - LIN模式标志：LIN断开检测(LBDF)和断开信号发送标识(SBF)；
  - 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(AMF0 / AMF1)，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(WUF)；
  - 接收器从静默模式唤醒(RWU)；
  - 若相应的中断使能，这些事件发生将会触发中断。
- 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。

24.2.3. 功能描述

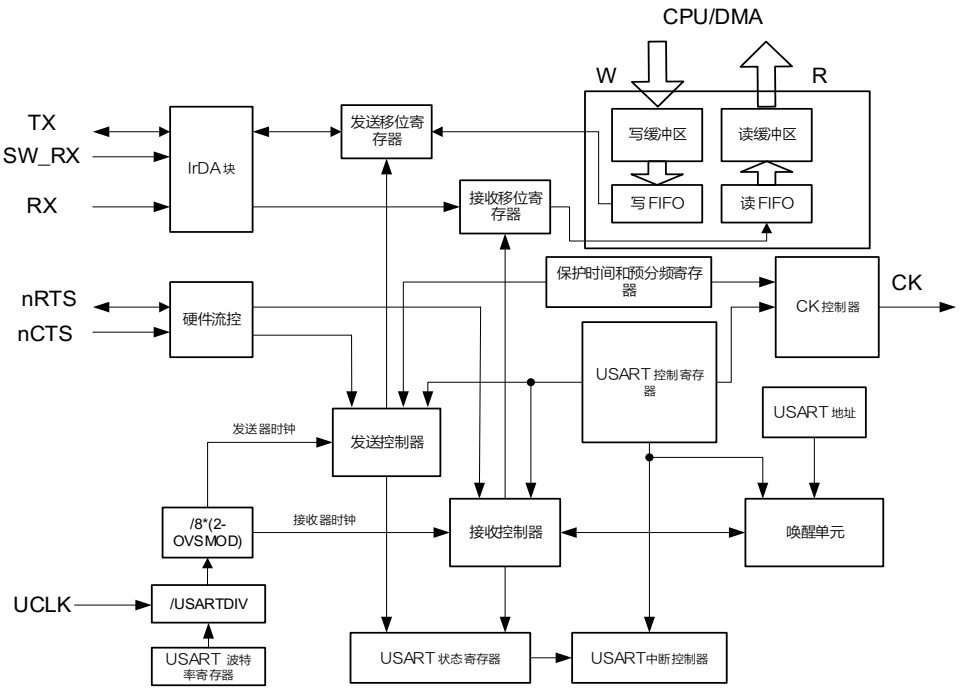
USART 接口通过*表 24-4. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 24-4. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O(单线模式/智能卡模式)	发送数据。当 USART 使能后，若无数据发送，默认为高电平
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号

引脚	类型	描述
nRTS	输出	硬件流控模式发送请求信号

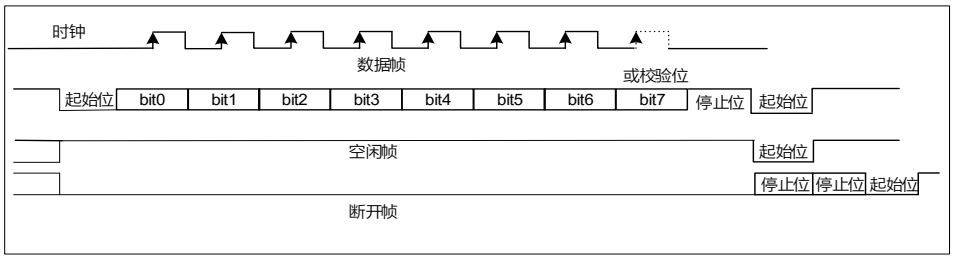
图 24-18. USART 模块内部框图



USART 帧格式

USART 数据帧开始于起始位，结束于停止位。USART_CTL0 寄存器中 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 24-19. 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 24-5. 停止位配置

STB[1:0]	停止位长度(位)	功能描述
00	1	默认值

STB[1:0]	停止位长度(位)	功能描述
01	0.5	智能卡模式接收
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，所有位都为 1。数据帧长度与正常 USART 数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART 数据帧的传输速度由 UCLK 时钟频率，波特率发生器的配置，以及过采样模式共同决定。

波特率发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数(USARTDIV)与 UCLK 具有如下关系：

如果过采样率是 16，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \tag{24-5}$$

如果过采样是 8，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \tag{24-6}$$

例如，当过采样是 16：

1. 由USART_BAUD寄存器的值得到USARTDIV：

假设 USART_BAUD=0x21D，则 INTDIV=33(0x21)，FRADIV=13(0xD)。

UASRTDIV=33+13/16=33.81。

2. 由USARTDIV得到USART_BAUD寄存器的值：

假设要求 UASRTDIV=30.37，INTDIV=30(0x1E)

16*0.37=5.92，接近整数 6，所以 FRADIV=6(0x6)

USART_BAUD=0x1E6。

注意：若取整后 FRADIV=16(溢出)，则进位必须加到整数部分。

USART 发送器

如果 USART_CTL0 寄存器的发送使能位(TEN)被置位，当发送数据缓冲区不为空时，发送器将会通过 TX 引脚发送数据帧。TX 引脚的极性可以通过 USART_CTL1 寄存器中 TINV 位来配置。时钟脉冲通过 CK 引脚输出。

TEN 置位后发送器会发出一个空闲帧。TEN 位在数据发送过程中是不可以被复位的。

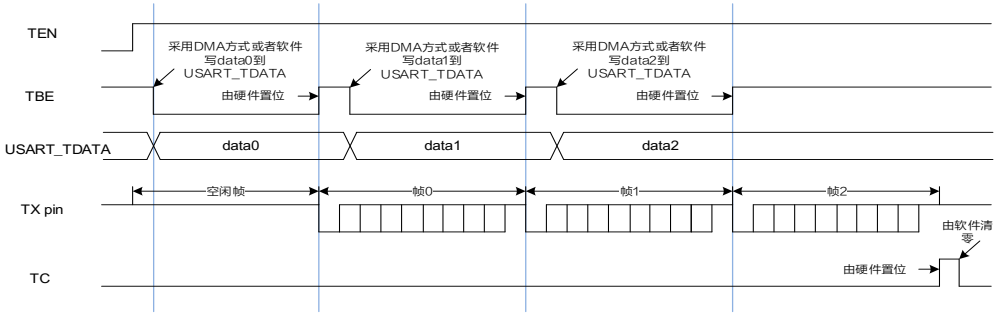
系统上电后，TBE 默认为高电平。在 USART_STAT 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_TDATA 寄存器。当数据写入 USART_TDATA 寄存器，TBE 位将被清 0。在数据由 USART_TDATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_TDATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_TDATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

假如一帧数据已经被发送出去，并且 TBE 位已被置位，那么 USART_STAT 寄存器中 TC 位将被置 1。如果 USART_CTL0 寄存器中的中断使能位(TCIE)为 1，将会产生中断。

*图 24-20. 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. 如果未启用FIFO，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 24-20. 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。

注意：当启用 FIFO 模式时，即使 TXFIFO 已满，发送断字符也优先于发送数据。

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。

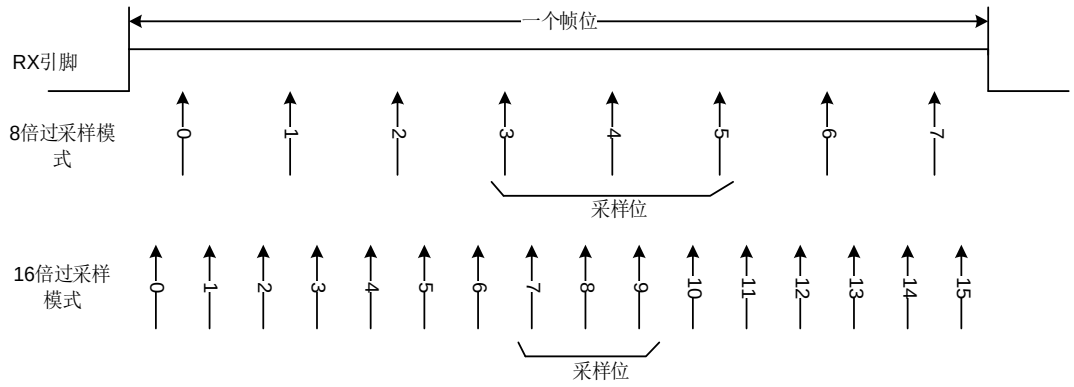
当接收到一个数据帧，USART_STAT 寄存器中的 RBNE 置位，如果设置了 USART_CTL0 寄存器中相应的中断使能位 RBNEIE，将会产生中断。在 USART_STAT 寄存器中可以观察接收状态标志。

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 24-21. 过采样方式接收一个数据位(OSB=0)



通过置位 USART_CTL0 寄存器中的 PCEN 位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT 寄存器中 PERR 被置位。如果置位了 USART_CTL0 寄存器中的 PERRIE 位，将产生中断。

如果在停止位传输过程中 RX 引脚为 0，将产生帧错误，USART_STAT 寄存器中 FERR 置位。如果使能 DMA 并置位 USART_CTL2 寄存器中 ERRIE 位，将产生中断。

当接收到一帧数据，而 RBNE 位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART_STAT 寄存器中的溢出错误标志位 ORERR 将置位。如果使能 DMA 并置位

USART_CTL2 寄存器中 ERRIE 位或者置位 RBNEIE，将产生中断。

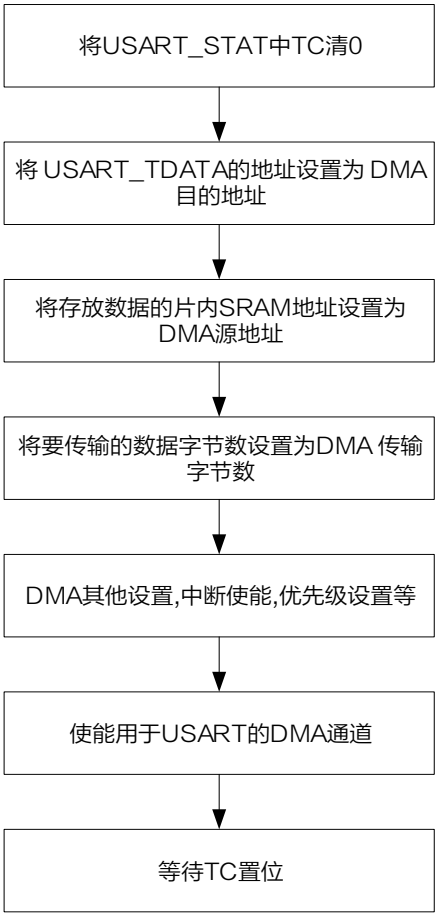
若接收过程中，产生了噪声错误(NERR)、校验错误(PERR)、帧错误(FERR)或溢出错误(ORERR)，则 NERR、PERR、FERR 或 ORERR 将和 RBNE 同时置位。如果没有使能 DMA，RBNE 中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用 DMA 访问发送缓冲区或者接收缓冲区。置位 USART_CTL2 寄存器中 DENT 位可以使能 DMA 发送，置位 USART_CTL2 寄存器中 DENR 位可以使能 DMA 接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如*图 24-22. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

图 24-22. 采用 DMA 方式实现 USART 数据发送配置步骤

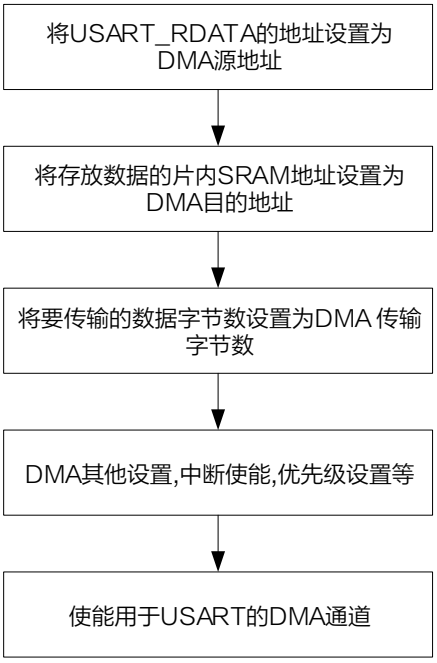


所有数据帧都传输完成后，USART_STAT 寄存器中 TC 位置 1。如果 USART_CTL0 寄存器中 TCIE 置位，将产生中断。

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*

*24-23. 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 24-23. 采用 DMA 方式实现 USART 数据接收配置步骤

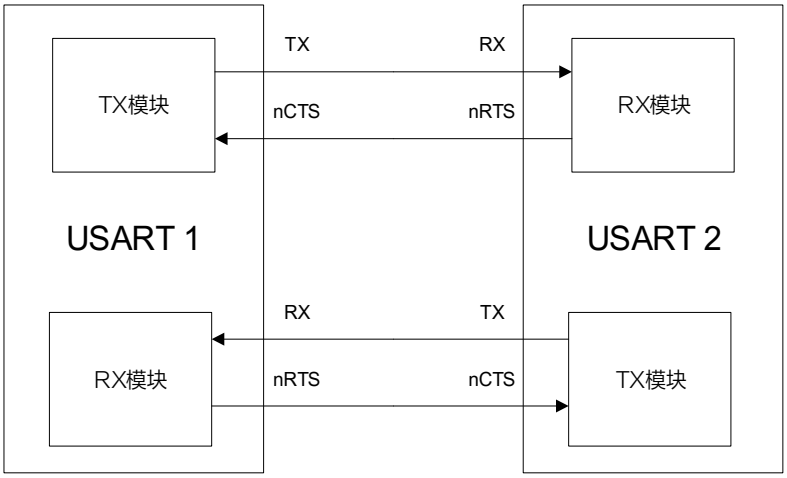


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

硬件流控制

硬件流控制功能通过 nCTS 和 nRTS 引脚来实现。通过将 USART_CTL2 寄存器中 RTSEN 位置 1 来使能 RTS 流控，将 USART_CTL2 寄存器中 CTSEN 位置 1 来使能 CTS 流控。

图 24-24. 两个 USART 之间的硬件流控制



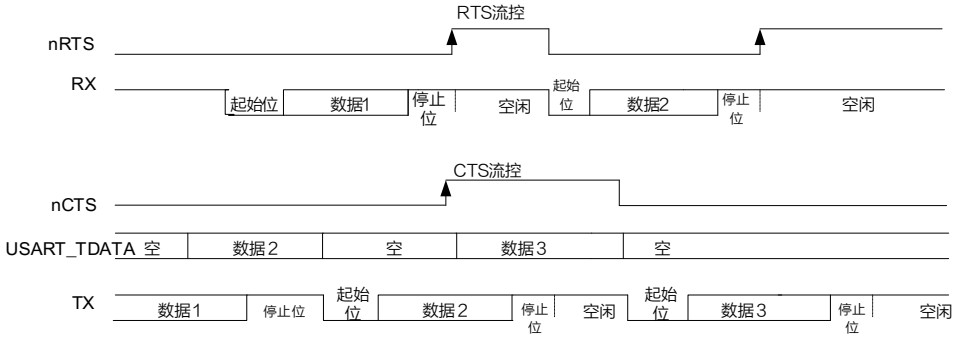
RTS 流控

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 24-25. 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。

注意： DE 信号与 RTS 共用一个引脚。

多处理器通信

在多处理器通信中，多个 USART 被连接成一个网络。对于一个设备来说，监视所有来自 RX 引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将 USART_CMD 寄存器中 MMCMD 位置 1 使 USART 进入静默模式。

如果 USART 处于静默模式，所有的接收状态标志位将不会被置位。此外，USART 可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址标记检测。

设备默认使用空闲总线检测方法唤醒 USART。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。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位相同，硬件会将 RWU 清零，并退出静默模式。接收到将 USART 唤醒的数据帧，RBNE 将置位。状态标志可以从 USART_STAT 寄存器中获取。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位不相同，硬件会置位 RWU 并自动进入静默模式。在这种情况下，RBNE 不会被置位。

如果 USART_CTL0 寄存器中 PCEN 位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果 ADDM 位被置位，且接收帧为 7 位的数据，其中最低的 6 位将与 ADDR[5:0]比较。如果 ADDM 位被置位，且接收帧为 9 位的数据，其中低 8 位将与 ADDR[7:0]进行比较。

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。

LIN 模式

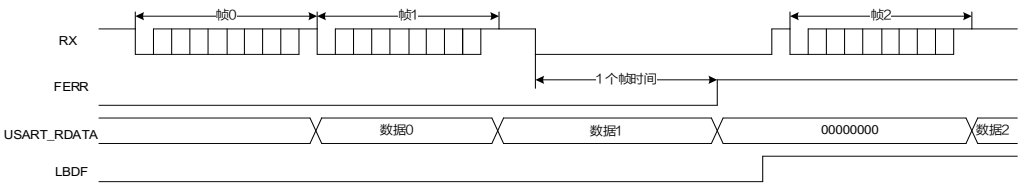
将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互联网络模式。在 LIN 模式下，USART_CTL1 寄存器中 CKEN，STB[1:0]和 USART_CTL2 的 SCEN，HDEN，IREN 位都应该被清 0。

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。

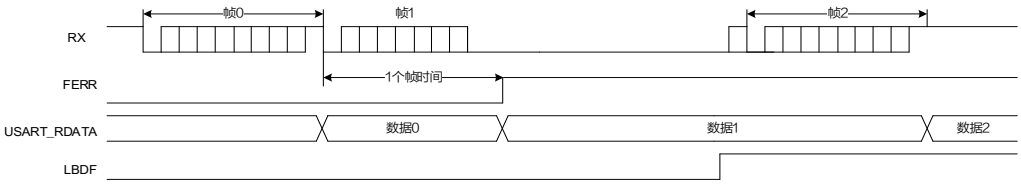
如*图 24-26. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 24-26. 空闲状态下检测断开帧



如*图 24-27. 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 24-27. 数据传输过程中检测断开帧



同步通信模式

USART 支持主机模式下的全双工同步串行通信，可以通过置位 USART_CTL1 的 CKEN 位来使能。在同步模式下，USART_CTL1 的 LMEN 和 USART_CTL2 的 SCEN，HDEN，IREN 位应被清 0。CK 引脚作为 USART 同步发送器的时钟输出，仅当 TEN 位被使能时，它才被激活。在起始位和停止位传送期间，不会从 CK 引脚输出时钟脉冲。USART_CTL1 的 CLEN 位用来决定在最低位(地址索引位)发送期间是否有时钟信号输出。在空闲状态和断开帧的发送过程中，也不会有时钟信号产生。USART_CTL1 的 CPH 位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1 的 CPL 位用来决定在 USART 同步模式空闲状态下，时钟引脚的电平。

CK 引脚输出波形由 USART_CTL1 寄存器中 CPL，CPH，CLEN 位决定。软件仅在 USART 禁用(UEN=0)时才可以改变它们的值。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，并无任何过采样。

图 24-28. 同步模式下的 USART 示例

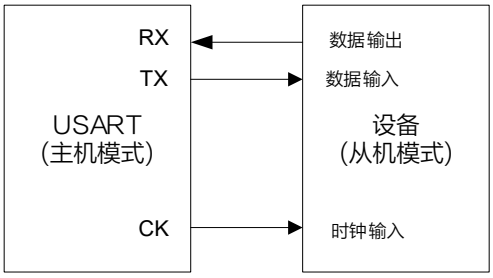
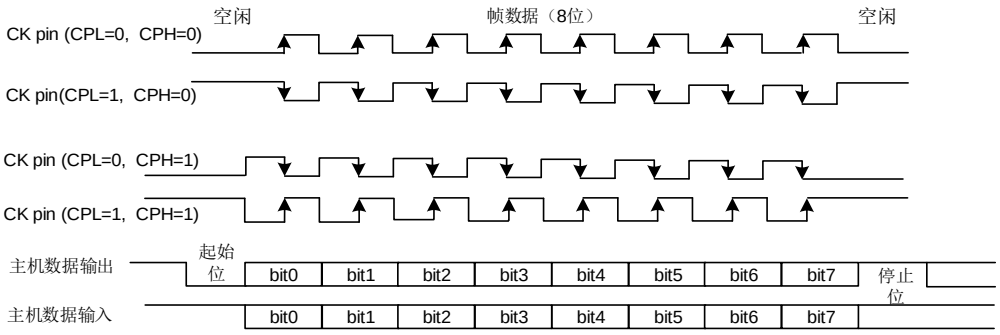


图 24-29. 8-bit 格式的 USART 同步通信波形(CLEN=1)

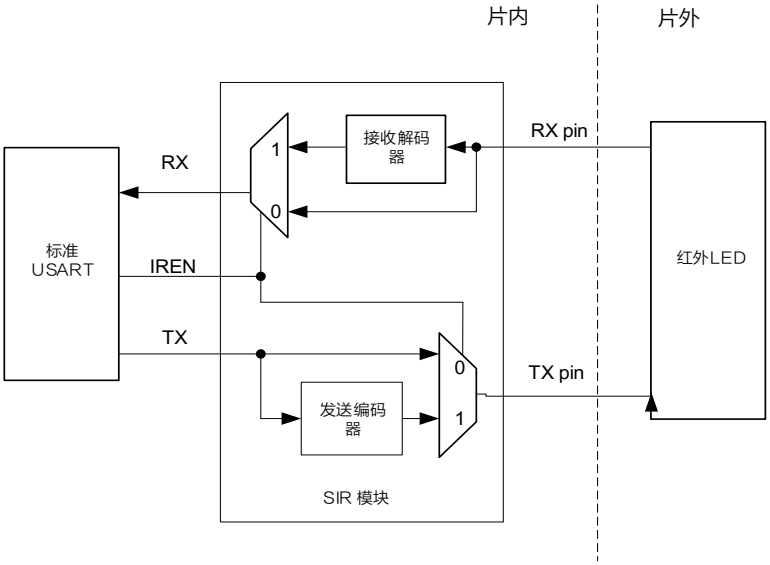


串行红外(IrDA SIR)编解码功能模块

串行红外编解码功能通过置位 USART_CTL2 寄存器中 IREN 使能。在 IrDA 模式下，USART_CTL1 寄存器的 LMEN，STB[1:0]，CKEN 位和 USART_CTL2 寄存器的 HDEN，SCEN 位应被清 0。

在 IrDA 模式下，USART 数据帧由 SIR 发送编码器进行调制，调制后的信号经由红外 LED 进行发送，经解调后将数据发送至 USART 接收器。对于编码器而言，波特率应小于 115200。

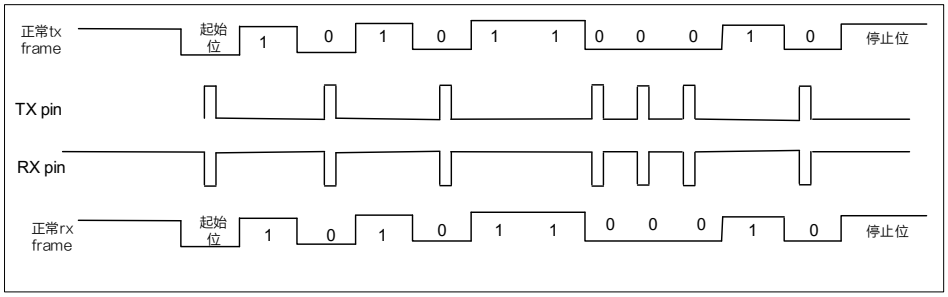
图 24-30. IrDA SIR ENDEC 模块



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 24-31. IrDA 数据调制



将 USART_CTL2 寄存器中 IRLP 置位可以使 SIR 子模块工作在低功耗模式下。发送编码器由 PCLK 分频得到的低速时钟来驱动。分频系数在 USART_GP 寄存器中 PSC[7:0]位配置。TX 引脚脉冲宽度可以为低功耗波特率的 3 倍。接收解码器工作模式与正常 IrDA 模式相同。

半双工通信模式

通过设置 USART_CTL2 寄存器的 HDEN 位，可以使能半双工模式。在半双工通信模式下，USART_CTL1 寄存器的 LMEN，CKEN 位和 USART_CTL2 寄存器的 SCEN，IREN 位应被清零。

半双工模式下仅用单线通信，TX 引脚和 RX 引脚从内部连接到一起，RX 引脚不再使用。TX 引脚应被配置为开漏模式，通信冲突由软件处理。当 TEN 被置位时，在数据寄存器中的数据将会被发送。

智能卡(ISO7816-3)模式

智能卡模式是一种异步通信模式，支持 ISO7816-3 协议。支持字节模式(T=0)和块模式(T=1)。将 USART_CTL2 寄存器的 SCEN 位置 1，即可使能智能卡模式。在智能卡模式下，USART_CTL1 寄存器的 LMEN 位和 USART_CTL2 的 HDEN，IREN 位应该清 0。

如果 CKEN 位被置位，USART 将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1 起始位+9 数据位(包括 1 个奇偶校验位)+1.5 停止位。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX 引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

图 24-32. ISO7816-3 数据帧格式



字节模式(T=0)

相较于正常操作模式下的时序，从发送移位寄存器到 TX 引脚的传递时间延迟了半个波特率时钟，并且 TC 标志的置位将根据 USART_GP 寄存器的 GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为 ISO7816-3 协议的 CGT 减 12。在保护时间寄存器向上计数这段时间 TC 将被强制拉低，当计数达到设定值时，TC 被置位。

在 USART 发送期间，如果检测到有奇偶校验错误，TX 引脚在停止位最后一个位时间内被拉低，智能卡发送一个 NACK 信号。根据协议，USART 会自动重发 SCRTNUM 次。在重发数据帧前面会插入 2 位的帧间隔。最后一次重发字节后，TC 会立即被置位。如果在最大重发次数后仍然收到 NACK 信号，USART 将会停止发送，帧错误标志被置位。USART 不会将 NACK 信号作为起始位。

在 USART 接收期间，如果在当前数据帧检测到校验错误，TX 引脚在停止位的最后一个位时

间内会被拉低。智能卡会接收到 NACK 信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE 中断和接收 DMA 请求都不会被激活。根据协议，智能卡将重新发送数据。如果在最大的重新发送次数后(这个次数的具体值在 SCRTNUM 位域)，接收到的字符仍然是错误的，USART 停止发送 NACK 信号和标注这个错误为奇偶校验错误。将 USART_CTL2 寄存器中的 NKEN 置位可以使能 NACK 信号。

空闲帧和断开帧在智能卡模式下不适用。

块模式(T=1)

在 T=1(块模式)下，USART_CTL2 寄存器的 NKEN 位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将 USART_RT 寄存器的 RT[23:0]位域设置成 BWT(块等待时间)-11 的值，并将 RBNEIE 置位。如果到了这个时间，还没有从智能卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起 RBNE 中断。块模式下，如果用 DMA 从智能卡读取数据，也只能在第一个字节接收完后再去使能 DMA。

在接收到第一个字节之后(RBNE 中断)必须将 USART_RT 寄存器设置为 CWT(字节等待时间)-11 之间的某个值(这个时间以波特时间作为单位)，这是为了自动检测两个连续字符之间的最大等待时间。如果智能卡在前一个字符发送结束后到设定的 CWT 周期之间没有发送字符，USART 会通过 RTF 标志提醒软件，当 RTIE 被置位时，会引起中断。

USART 用一个块长度计数器统计收到的字节数，这个计数器在 USART 开始发送的时候自动清 0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)。这个值必须写入 USART_RT 寄存器的 BL[7:0]。当使用 DMA 模式时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以从接收缓冲区读取第三个字节作为块长度。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置 BL 的值得到。但是在块开始之前，BL(0xFF)可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度(包括序言区，收尾区和信息区)等于 BL+4。块尾通过 EBF 标志和相应中断提醒给软件(当 EBIE 位置 1 时)。如果块长度出错，将会引起一个 RT 中断。

直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX 引脚高电平代表逻辑'1'，偶校验。在这种情况下，MSBF 位和 DINV 位都应设置为 0(默认值)。

如果选择反向转换，从数据帧的最高位开始传输，TX 引脚低电平代表逻辑'1'，偶校验。在这种情况下，MSBF 位和 DINV 位都应设置为 1。

ModBus 通信

通过实现块尾检测功能，USART 提供实现 ModBus/RTU 和 ModBus/ASCII 协议的基本支持。

在 ModBus/RTU 模式下，通过一个超过 2 个字符长度的空闲状态来识别块尾。这个功能是通过一个可编程的超时检测功能来实现的。

为了检测空闲状态，必须置位 USART_CTL1 寄存器的 RTEN 位和 USART_CTL0 寄存器的 RTIE 位。USART_RT 寄存器必须被设置成与 2 个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

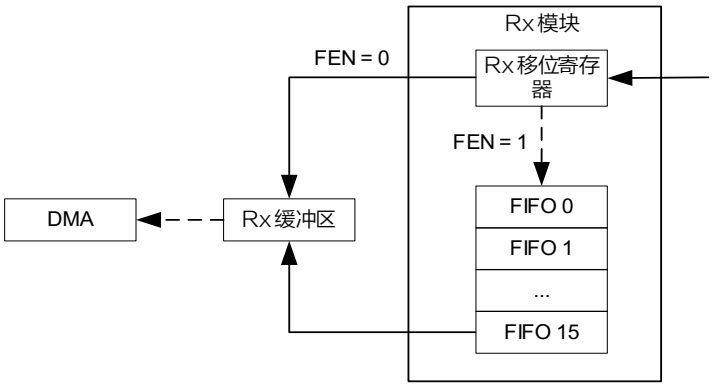
在 ModBus/ASCII 模式下，块尾被认为是一个特定的字符(CR/LF)串。USART 用字符匹配机制实现这个功能。具体是通过将 LF 的 ASCII 码配置到 ADDR 区域并激活地址匹配中断(AMIE=1)来实现。软件将在收到 LF 或可以在 DMA 缓存中查找到 CR/LF 时得到提示。

接收/发送 FIFO

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 24-33. USART 接收 FIFO 结构

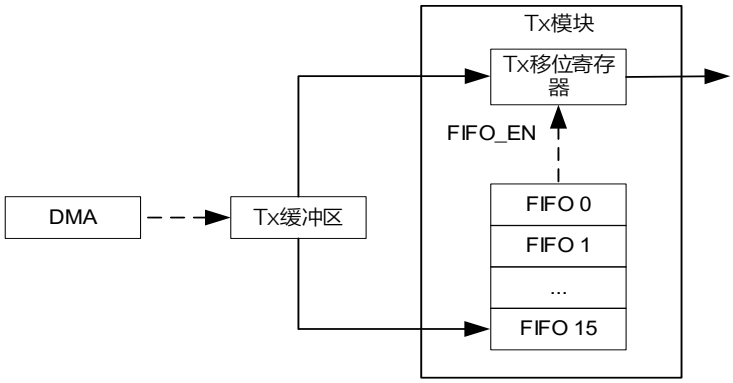


如果软件在响应 RBNE 中断时读数据接收缓冲区，在响应开始时，RBNEIE 位应清 0。当所有接收的数据被读出后，RBNEIE 位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF 都应被清 0。

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 17 帧的数据。若发送 FIFO 空，TFE 将被置位，如果 TFEIE 位被置位，将产生中断。若发送 FIFO 中的可用空间达到 TFTCFG[2:0]位域配置的阈值，TFTIF 将被置位(或者 TFT 将被清除)，如果 TFTIE 被置位，将产生中断。若发送 FIFO 非满，TFNF 位被置位，如果 TFNFIE 被置位，将产生中断。

图 24-34. USART 发送 FIFO 结构



从 Deepsleep 模式唤醒

通过标准 RBNE 中断或 WUM 中断 USART 能从深度睡眠模式和深度睡眠模式 1 唤醒 MCU。

UESM 位必须置 1 并且 USART 时钟必须设置为 IRC8M 或 LXTAL(请参考 RCU 部分)。

当使用 RBNE 标准中断时,必须在进入深度睡眠模式和深度睡眠模式 1 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和深度睡眠模式 1 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。

USART 中断

USART 中断事件和标志如*表 24-6. USART 中断请求*所示:

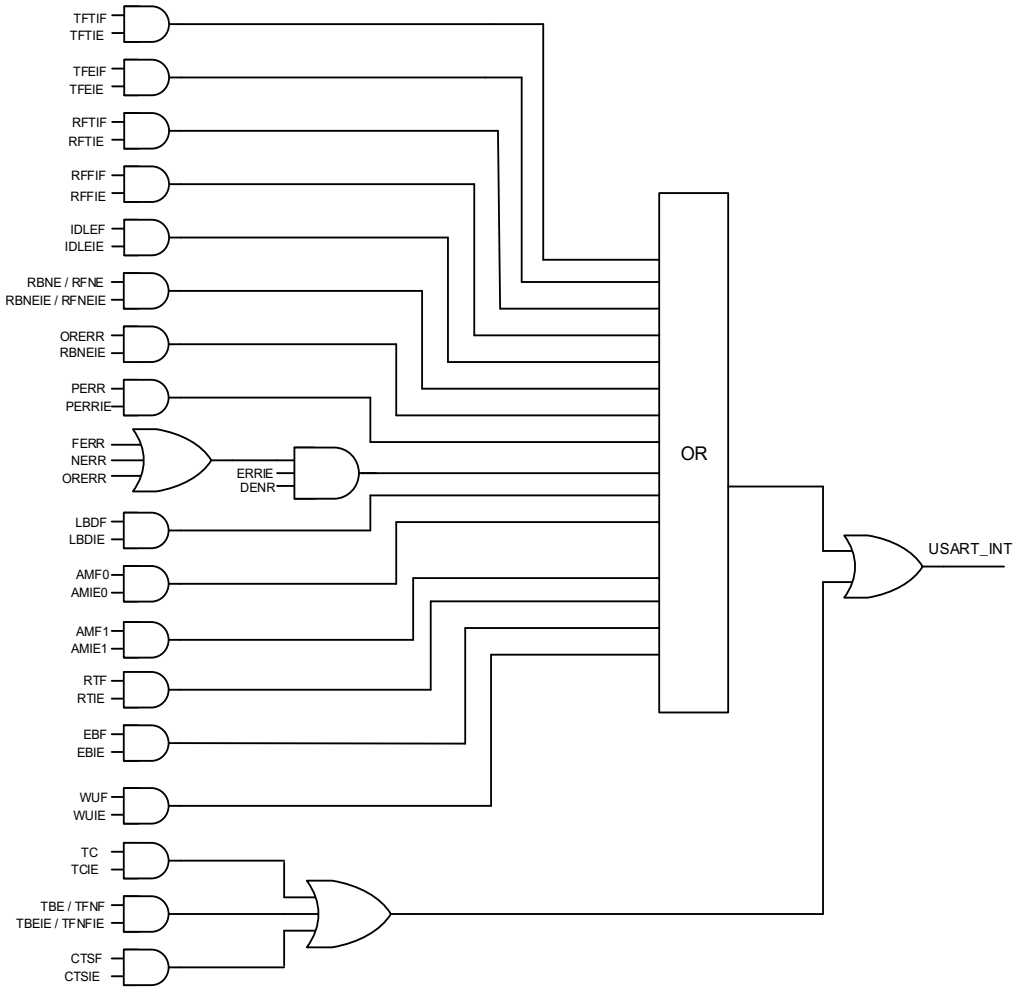
表 24-6. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空或发送FIFO非满	TBE / TFNF	TBEIE / TFNFIE
CTS标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取或接收FIFO非空	RBNE / RFNE	RBNEIE / RFNEIE
检测到过载错误	ORERR	
接收FIFO满	RFFIF	RFFIE
接收FIFO达到阈值	RFTIF	RFTIE
发送FIFO空	TFEIF	TFEIE

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 24-35. USART 中断映射框图



24.2.4. USART 寄存器

USART5基地址：0x4001 7000

USART 控制寄存器 0 (USART_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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。 **注意:** 1. 在7位数据长度模式下，不支持智能卡模式和LIN主模式。 2. 在 10 位数据长度模式下，不支持地址标记功能。
27	EBIE	块尾中断使能 0：中断禁止 1：中断使能
26	RTIE	接收超时中断使能 0：中断禁止 1：中断使能
25:21	DEA[4:0]	驱动使能置位时间 这些数字用来定义DE(驱动使能)信号的置位与第一个字节的起始位之间的时间间

		隔。它以采样时间为单位(1/8或1/16位时间)，可以通过OVSMOD位来配置。 当 USART 被使能(UEN = 1)时，该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低DE(驱动使能)信号之间的时间间隔。它以采样时间为单位(1/8或1/16位时间)，可以通过OVSMOD位来配置。 当 USART 被使能(UEN = 1)时，该位域不能被改写。
15	OVSMOD	过采样模式 0：16倍过采样 1：8倍过采样 在LIN，IrDA和智能卡模式，该位保持清0。 当 USART 被使能(UEN = 1)时，该位域不能被改写。
14	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。 当 USART 被使能(UEN = 1)时，该位域不能被改写。
11	WM	从静默模式唤醒方法 0：空闲线 1：地址标记 当 USART 被使能(UEN = 1)时，该位域不能被改写。
10	PCEN	校验控制使能 0：校验控制禁用 1：校验控制被使能 当 USART 被使能(UEN = 1)时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当 USART 被使能(UEN = 1)时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0：校验错误中断禁用 1：当 USART_STAT 寄存器的 PERR 位置位时，将触发中断。
7	TBEIE	当FIFO禁用: 发送寄存器空中断使能 0：中断禁止

		1：当 USART_STAT 寄存器的 TBE 位置位时，将触发中断。
	TFNFIE	当FIFO使能： 发送FIFO非满中断使能 0：中断禁止 1：当 USART_STAT 寄存器的 TFNF 位置位时，将触发中断。
6	TCIE	发送完成中断使能 如果该位置1，USART_STAT寄存器中TC被置位时产生中断。 0：发送完成中断禁用 1：发送完成中断使能
5	RBNEIE	当FIFO禁用： 读数据缓冲区非空中断和过载错误中断使能 0：读数据缓冲区非空中断和过载错误中断禁用 1：当 USART_STAT 寄存器的 ORERR 或 RBNE 位置位时，将触发中断。
	RFNEIE	当FIFO使能： 接收FIFO非空中断使能和过载错误中断使能 0：接收FIFO非空中断和过载错误中断禁用 1：当 USART_STAT 寄存器的 ORERR 或 RFNE 位置位时，将触发中断。
4	IDLEIE	IDLE线检测中断使能 0：IDLE线检测中断禁用 1：当 USART_STAT 寄存器的 IDLEF 位置位时，将触发中断。
3	TEN	发送器使能 0：发送器关闭 1：发送器打开
2	REN	接收器使能 0：接收器关闭 1：接收器打开并且开始搜索起始位。
1	UESM	USART在深度睡眠模式下使能 0：USART不能从深度睡眠模式唤醒MCU 1:USART 能从深度睡眠模式唤醒 MCU。条件是 USART 的时钟源必须是 CK_IRC8M 或 CK_LXTAL。
0	UEN	USART使能 0：USART预分频器和输出禁用 1：USART 预分频器和输出被使能

USART 控制寄存器 1 (USART_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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		LBEN	AMEN0
rw	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。
22:20	保留	必须保持复位值。
19	MSBF	高位在前 0：数据发送/接收，采用低位在前。 1：数据发送/接收，采用高位在前。 USART 被使能(UEN=1)时，该位域不能被改写。
18	DINV	数据位反转。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
17	TINV	TX 管脚电平反转。 0：TX 管脚信号值没有反转。 1：TX 管脚信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
16	RINV	RX 管脚电平反转。 0：RX 管脚信号值没有反转。 1：RX 管脚信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
15	STRP	交换 TX/RX 管脚。 0：TX 和 RX 管脚功能不被交换。

		1：TX 和 RX 管脚功能被交换 。 当 USART 被使能(UEN=1)时，该位域不能改写。
14	LMEN	LIN 模式使能。 0：LIN 模式关闭。 1：LIN 模式开启。 USART 被使能(UEN=1)时，该位域不能被改写。
13:12	STB[1:0]	STOP 位长 00：1 停止位 01：0.5 停止位 10：2 停止位 11：1.5 停止位 USART 被使能(UEN=1)时，该位域不能被改写。
11	CKEN	CK 管脚使能。 0：CK 管脚禁用。 1：CK 管脚被使能。 USART 被使能(UEN=1)时，该位域不能被改写。
10	CPL	时钟极性 0：在同步模式下，CK 管脚不对外发送时保持为低电平。 1：在同步模式下，CK 管脚不对外发送时保持为高电平。 USART 被使能(UEN=1)时，该位域不能被改写。
9	CPH	时钟相位 0：在同步模式下，在首个时钟边沿采样第一个数据。 1：在同步模式下，在第二个时钟边沿采样第一个数据。 USART 被使能(UEN=1)时，该位域不能被改写。
8	CLEN	CK 长度 0：在同步模式下，最后一位(MSB)的时钟脉冲不输出到 CK 管脚。 1：在同步模式下，最后一位(MSB)的时钟脉冲输出到 CK 管脚。 USART 被使能(UEN=1)时，该位域不能被改写。
7	保留	必须保持复位值。
6	LBDIE	LIN 断开信号检测中断使能。 0：断开信号检测中断禁用。 1：当 USART_STAT 的 LBDF 位置位，将产生中断。
5	LBDL	LIN 断开帧长度。 0：检测 10 位断开帧。 1：检测 11 位断开帧。 USART 被使能(UEN=1)时，该位域不能被改写。
4	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。
3:2	保留	必须保持复位值。
1	LBEN	Loopback模式 0：失能Loopback模式 1：使能Loopback模式 USART 被使能(UEN = 1)时，该位域不能被改写。
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

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
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位USART_STAT寄存器中的WUF(从深度睡眠唤醒标志)标志。 00：WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01：保留 10：WUF在检测到起始位时置位 11：WUF在检测到RBNE时置位 USART被使能(UEN = 1)时，该位域不能被改写。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误(FERR位置位)之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误(RBNE位和PERR位置位)之前自动重试的接收次数。 当这些位被设置为0x0时，在发送模式下这些位将不会自动发送。 USART被使能(UEN = 1)时，该位域被清零，并停止重发。
16	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 USART被使能(UEN = 1)时，该位域不能被改写。
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。
4	NKEN	智能卡模式NACK使能 0：当出现校验错误时不发送NACK 1：当出现校验错误时发送NACK USART被使能(UEN = 1)时，该位域不能被改写。
3	HDEN	半双工使能 0：禁用半双工模式

		1：开启半双工模式 USART被使能(UEN = 1)时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0：正常模式 1：低功耗模式 USART被使能(UEN = 1)时，该位域不能被改写。
1	IREN	IrDA模式使能 0：IrDA禁用 1：IrDA被使能 USART被使能(UEN = 1)时，该位域不能被改写。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断 1：在多级缓存通信时，当USART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

USART 波特率寄存器 (USART_BAUD)

地址偏移：0x0C
复位值：0x0000 0000

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。

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

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

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:4	BRR[19:4]	波特率分频系数的整数部分。 INTDIV = BRR[19:4]
3:0	BRR[3:0]	波特率分频系数的小数部分。 如果 OVSMOD = 0，FRADIV = BRR [3:0]。 如果 OVSMOD = 1，FRADIV = BRR [2:0]，BRR [3]必须被置 0。

USART 保护时间和预分频器寄存器 (USART_GP)

地址偏移：0x10

复位值：0x0000 0000

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。



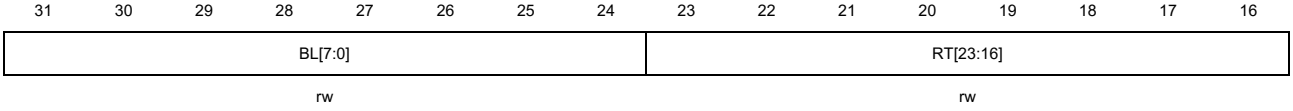
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值。 USART 被使能(UEN=1)时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000：保留 - 不设置这个值 00000001：1 分频 00000010：2 分频 ... 在 IrDA 正常模式下的分频值 00000001：仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于 PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000：保留 -不设置这个值 00001：2 分频 00010：4 分频 00011：6 分频 ... USART 被使能(UEN=1)时，该位域不能被改写。

USART 接收超时寄存器 (USART_RT)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



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

位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位给出了智能卡 T=1 的接收时块的长度。它的值等于信息字节的长度+结束部分的长度(1-LEC/2-CRC) - 1。 这个值可以在块接收开始时设置(用于需要从块的序言提取块的长度的情形)，这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当 TBE=0 时，块的长度计数器被清 0。 在其他模式下，当 REN=0 (禁用接收器)并且/或者当 EBC 位被写 1 时块的长度计数器被清 0。
23:0	RT[23:0]	接收器超时门限。 该位域指定接收超时值，单位是波特时钟的时长。 标准模式下，如果在最后一个字节接收后，在 RT 规定的时长内，没有检测到新的起始位，RTF 标志被置位。 在智能卡模式，这个值被用来实现 CWT 和 BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始。 这些位可以在工作时改写。假如一个新数据到来的时间比 RT 规定的晚，RTF 标志会被置位。对于每个接收字符，这个值只能改写一次。

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。 当 FIFO 被禁用时，向该位写入 1 会设置 TBE 标志，以丢弃发送数据。 当 FIFO 被启用时，向该位写入 1 会清空整个发送 FIFO，并在 USART_RFCS 中设置 TFE 标志。 **注意:** 当 FIFO 被启用时，在执行清空命令期间，TFNF 标志会被复位，直到发送 FIFO

		为空。
3	RXFCMD	接收数据清空请求。 当 FIFO 被禁用时，向该位写入 1 会清除 RBNE 标志，以丢弃未读取的接收数据。 当 FIFO 被启用时，向该位写入 1 会清空接收 FIFO，并在 USART_RFCS 中设置 RFNE 标志。
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。

USART 状态寄存器(USART_STAT)

地址偏移：0x1C
复位值：0x0000 00C0

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									REA	TEA	WUF	RWU	SBF	AMF0	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		AMF1	EBF	RTF	CTS	CTSF	LBDF	TBE TFNF	TC	RBNE RFNE	IDLEF	ORERR	NERR	FERR	PERR
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志 这位反映了USART核心逻辑的接收使能状态，该位可以通过硬件设置。 0：USART核心接收逻辑禁用 1：USART核心接收逻辑被使能
21	TEA	发送使能通知标志 该位反映了USART核心逻辑的发送使能状态，该位可以通过硬件设置。 0：USART核心发送逻辑禁用 1：USART核心发送逻辑被使能
20	WUF	从深度睡眠模式和深度睡眠模式1唤醒标志 0：没有从深度睡眠模式和深度睡眠模式1唤醒 1：已从深度睡眠模式和深度睡眠模式1唤醒，如果在USART_CTL2寄存器的WUFIE = 1并且MCU处于深度睡眠模式和深度睡眠模式1，将引发一个中断。

		当检测到一个唤醒事件时，该位通过硬件置位，这个事件在WUM位域被定义。 向USART_INTC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。
19	RWU	接收器从静默模式唤醒 该位表示USART处于静默模式。 0：接收器在工作状态 1：接收器在静默状态 当在唤醒和静默模式切换时，它通过硬件清0或者置1。静默模式控制(地址帧还是空闲帧)是用通过USART_CTL0寄存器的WM位选择。 如果选择空闲信号唤醒，只能通过向USART_CMD寄存器的MMCMD位写1来将该位置位。 **注意：**当 UEN=1 时，RWU 仅有效；当 UEN=0 时，此位保持为 0。
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。
11	RTF	接收超时标志 0：尚未超时 1：已经超时，如果USART_CTL1寄存器的RTIE被置位，将会引发中断。 如果空闲的时间已经超过了在USART_RT寄存器中设定的RT值，通过硬件置1。 通过向USART_INTC寄存器的RTC位写1清0。 在智能卡模式，这个超时相当于CWT或BWT计时。
10	CTS	CTS电平 这个值等于nCTS输入引脚电平的反向拷贝。 0：nCTS输入引脚高电平 1：nCTS输入引脚低电平
9	CTSF	CTS变化标志 0：nCTS状态线没有变化 1：nCTS状态线发生变化 如果USART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向USART_INTC寄存器的CTSC位写1，清零该位。
8	LBDF	LIN断开检测标志 0：没有检测到LIN断开字符 1：检测到LIN断开字符。当USART_CTL1寄存器的LBDIE位被置位时，将会有中断产生。 当LIN断开帧被检测到的时候，硬件置位。 通过向USART_INTC寄存器的LBDC位写1，清零该位。
7	TBE	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0。 **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。 通过读取USART_RDATA或向USART_CMD寄存器的RXFCMD位写入1来清除。
	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。

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

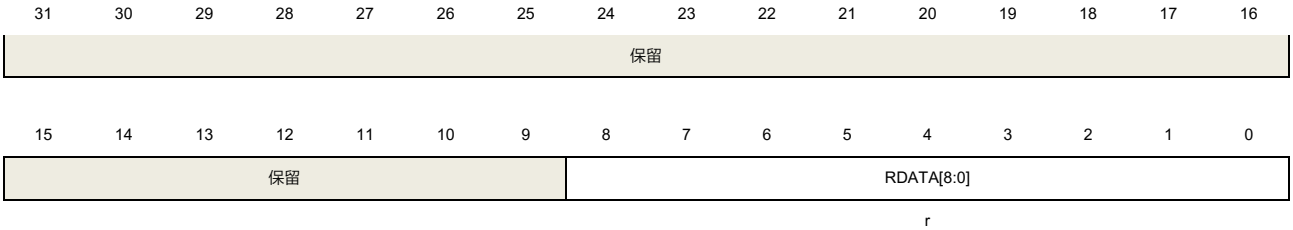
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除USART_STAT寄存器的WUF位。
19:18	保留	必须保持复位值。
17	AMC0	ADDR0中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF0位。
16	AMC1	ADDR1中字符匹配标志清除 向该位写1清除USART_STAT寄存器的AMF1位。
15:13	保留	必须保持复位值。
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。
11	RTC	接收超时标志清除

		向该位写1清除USART_STAT寄存器的RTF标志。
10	保留	必须保持复位值。
9	CTSC	CTS变化标志清除 向该位写1清除USART_STAT寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清除USART_STAT寄存器的LBDF标志位。
7	保留	必须保持复位值。
6	TCC	发送完成标志清除 向该位写1清除USART_STAT寄存器的TC位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除 向该位写1清除USART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除USART_STAT寄存器的ORERR位。
2	NEC	噪声检测清除 向该位写1清除USART_STAT寄存器的NERR位。
1	FEC	帧格式错误标志清除 向该位写1清除USART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除USART_STAT寄存器的PERR位。

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。



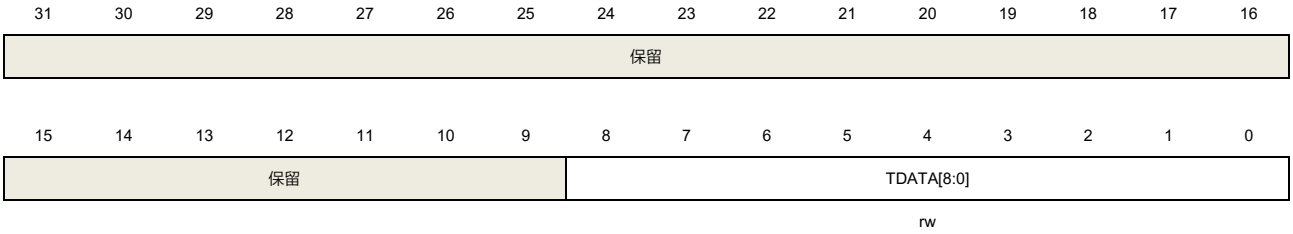
位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值。

包含接收到的数据字节。
如果接收到的数据打开了奇偶校验位(USART_CTL0 寄存器的 PCEN 置 1)，那么接收到的数据的最高位(第 7 位或 8 位，取决于数据的长度)是奇偶校验位。

USART 数据发送寄存器 (USART_TDATA)

地址偏移：0x28
复位值：未定义

该寄存器只能按字(32 位)访问。

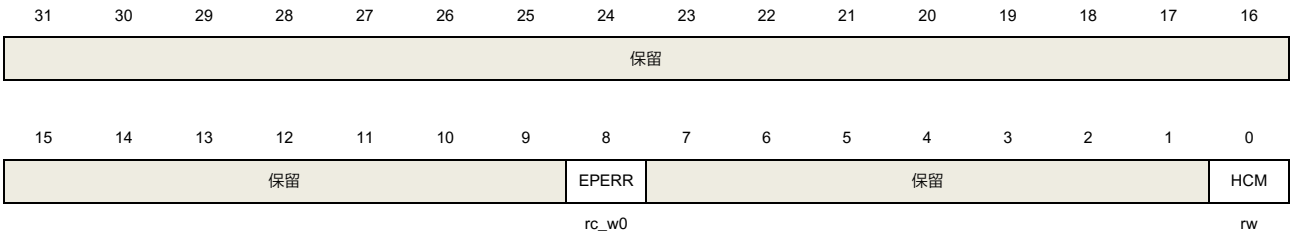


位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值。 包含发送的数据字节。 如果发送到的数据打开了奇偶校验位(USART_CTL0 寄存器的 PCEN 置 1)，那么发送的数据的最高位(第 7 位或 8 位取决于数据的长度)将会被奇偶校验位替代。 只有当 USART_STAT 寄存器的 TBE 位被置位时，这个寄存器才可以改写。

USART 兼容性控制寄存器 (USART_CHC)

地址偏移：0xC0
复位值： 0x0000 0000

该寄存器只能按字(32 位)访问。



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

USART FIFO 控制和状态寄存器 (USART_FCS)

地址偏移：0xD0
复位值：0x0000 0400

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFEIE	保留	TFTIE	保留	RFTIE	TFEC	TFTIF	TFEIF	保留	RFTIF	TFTCFG[2:0]			RFTCFG[2:0]		
rw		rw		rw	rw	r	r		rc_w0	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFIF	RFCNT[2:0]			RFF	RFE	RFFIE	FEN	TFF	TFE	TFT	RFT	保留	RFCNT[4:3]		ELNACK
rc_w0	r			r	r	rw	rw	r	r	r	r		r		rw

位/位域	名称	描述
31	TFEIE	发送FIFO空中断使能 如果该位置位，当TFE位置位时，中断发生。 0：禁止发送FIFO空中断 1：使能发送FIFO空中断
30	保留	必须保持复位值。
29	TFTIE	发送FIFO到达阈值中断使能 如果该位置位，当发送FIFO中的可用空间到达TFTCFG[2:0]配置的阈值时，中断发生。 0：禁止发送FIFO到达阈值中断 1：使能发送FIFO到达阈值中断
28	保留	必须保持复位值。
27	RFTIE	接收FIFO到达阈值中断使能 如果该位置位，当发送FIFO到达RFTCFG[2:0]配置的阈值时，中断发生。 0：禁止接收FIFO到达阈值中断 1：使能接收FIFO到达阈值中断
26	TFEC	发送FIFO空标志清除 写1清除TFE标志

25	TFTIF	发送FIFO到达阈值中断标志 当TFTIE位置位时，该位有效。 0：发送FIFO中的可用空间未到达可编程阈值 1：发送FIFO中的可用空间到达可编程阈值中断标志 当发送FIFO中的可用空间到达TFTCFG[2:0]配置的阈值时，由硬件置位。
24	TFEIF	发送FIFO空中断标志 当TFEIE置位时，该位有效。 0：发送FIF非空 1：发送FIFO为空中断标志
23	保留	必须保持复位值。
22	RFTIF	接收FIFO到达阈值中断标志 当RFTIE位置位时，该位有效。 0：接收FIFO到达可编程阈值 1：接收FIFO到达可编程阈值中断标志
21:19	TFTCFG[2:0]	发送FIFO阈值配置 000：发送FIFO到达FIFO深度的1/8 001：发送FIFO到达FIFO深度的1/4 010：发送FIFO到达FIFO深度的1/2 011：发送FIFO到达FIFO深度的3/4 100：发送FIFO到达FIFO深度的7/8 101：发送FIFO为空 11x：保留
18:16	RFTCFG[2:0]	接收FIFO阈值配置 000：接收FIFO到达FIFO深度的1/8 001：接收FIFO到达FIFO深度的1/4 010：接收FIFO到达FIFO深度的1/2 011：接收FIFO到达FIFO深度的3/4 100：接收FIFO到达FIFO深度的7/8 101：接收FIFO为满 11x：保留
15	RFFIF	接收FIFO满中断标志 当RFFIE置位时，该位有效。 0：接收FIFO非满 1：接收FIFO满中断标志
14:12	RFCNT[2:0]	接收FIFO计数值 该位域与RFCNT[4:3]位域决定接收FIFO计数值。
11	RFF	接收FIFO满标志 0：接收FIFO不为满 1：接收FIFO满。当RFFIE置位时，中断发生。

		当接收数据个数为RXFIFO 大小加1时，由硬件置1。
10	RFE	接收FIFO空标志 0：接收FIFO不为空 1：接收FIFO空
9	RFFIE	接收FIFO满中断使能 如果该位置位，当RFF位置位时，中断发生。 0：禁止接收FIFO满中断 1：使能接收FIFO满中断
8	FEN	FIFO使能 0：禁止使用FIFO 1：使能FIFO 当USART被使能(UEN = 1)时，该位域不能被改写。 **注意：**当接收或发送数据未完成时，不要改变该位。当UEN位清零且不改变该位，在重配UEN位时，如果之前FIFO的值不在需要，需要先刷新FIFO。
7	TFF	发送FIFO满标志 0：发送FIFO不为满 1：发送FIFO满。
6	TFE	发送FIFO空标志 0：发送FIFO不为空 1：发送FIFO空。当TFEIE置位时，中断发生。 当发送FIFO为空时，由硬件置位。当发送FIFO中只少有一个数据时，由硬件清0。向USART_CMD寄存器的TXFCMD位写1时，由硬件置位
5	TFT	发送FIFO阈值标志 0：发送FIFO中的可用空间到达可编程阈值。当TFTIE置位时，中断发生。 1：发送FIFO的可用空间未到达可编程阈值。 当发送FIFO中的可用空间到达TFTCFG[2:0]配置的阈值时，由硬件复位。
4	RFT	接收FIFO阈值标志 0：接收FIFO未到达可编程阈值 1：接收FIFO到达可编程阈值。当RFTIE置位时，中断发生。 当接收FIFO到达RFTCFG[2:0]配置的阈值时，由硬件置1。这意味着接收FIFO中有RFTCFG[2:0] -1个数据，USART_RDATA寄存器中有一个数据。 **注意：**当RTFCFG[2:0]=0b101且接收到16个数据时，RFT被置位。
3	保留	必须保持复位值。
2:1	RFCNT[4:3]	接收FIFO计数值 该位域与RFCNT[2:0]位域决定接收FIFO计数值。
0	ELNACK	若选择了智能卡模式，提前NACK 如果检测到校验位错误，NACK脉冲提前1/16位的时间。 0：若选择了智能卡模式，禁止提前NACK

1：若选择了智能卡模式，使能提前NACK

25. 内部集成电路总线接口(I2C)

25.1. 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(Fast mode plus，Fm+)，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)和 PMBus(电源管理总线)。此外，I2C 接口模块还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

25.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口；
- 同一接口既可实现主机功能又可实现从机功能；
- 主从机之间的双向数据传输；
- 支持 7 位和 10 位的地址模式和广播寻址；
- 多个 7 位从机地址(两个地址，其中一个可配置地址位屏蔽)；
- 可编程的建立时间和保持时间；
- 支持 I2C 多主机模式；
- 支持标速(最高 100 kHz)，快速(最高 400 kHz)和 Fm+模式(最高 1MHz，必须在 SYSCFG_CFG1 中使能 Fm+)；
- 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。

25.3. 功能说明

I2C 接口的内部结构如*图 25-1. I2C 模块框图*所示。

图 25-1. I2C 模块框图

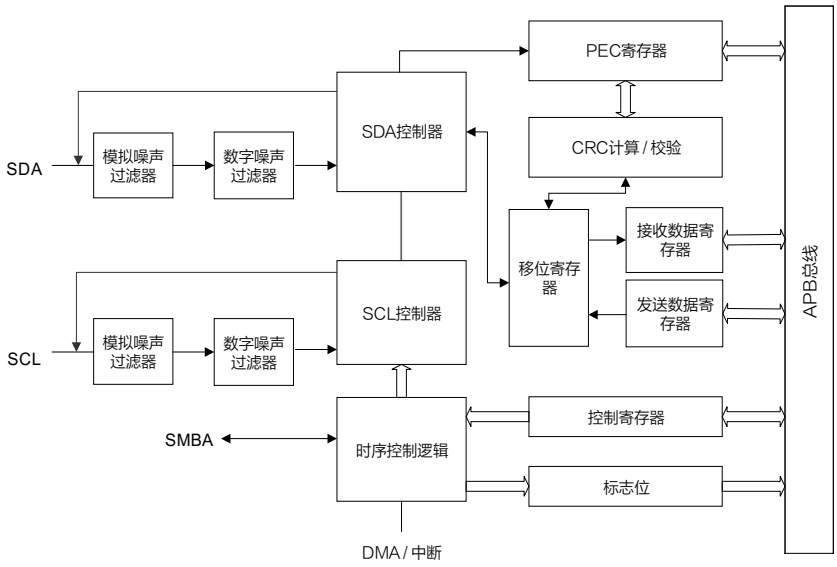


表 25-1. I2C 总线术语说明(参考飞利浦 I2C 规范)

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

25.3.1. 时钟要求

I2C2 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C2 时钟(I2C2CLK)可以从以下三个时钟源中选择：

- APB1 时钟 CK_AKB1(默认值)
- IRC8M 时钟
- 系统时钟 SYSCLK

I2C 时钟周期t_{I2CCLK}必须满足以下条件：

- $t_{I2CCLK}<(t_{LOW}-t_{filters})/4$
- $t_{I2CCLK}<t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间

t_{HIGH}：SCL 高电平时间

$t_{filters}$：在使能滤波器时，表示模拟滤波器和数字滤波器产生的延时总和。模拟滤波器产生的延时最大值为 150ns，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。

25.3.2. I2C 通讯流程

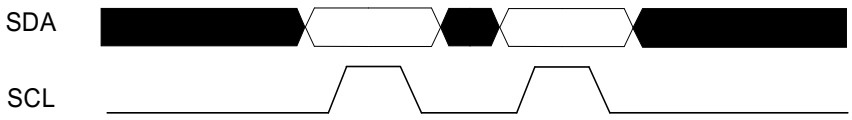
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 25-2. 数据有效性*)。每个数据比特传输需要一个时钟脉冲。

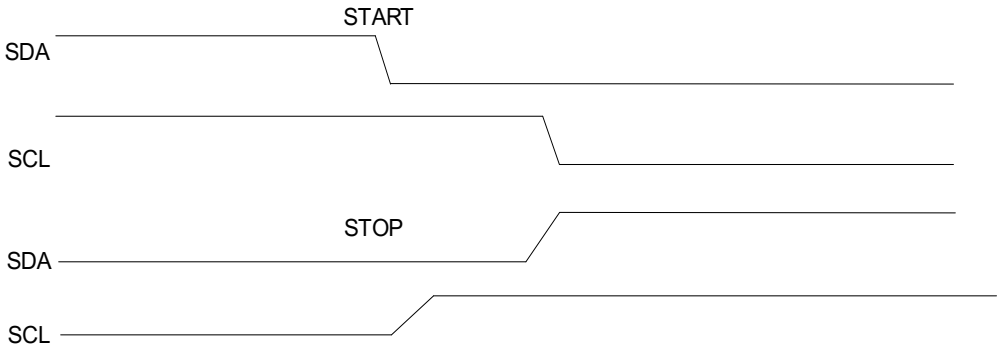
图 25-2. 数据有效性



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 25-3. 开始和停止信号*)。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 25-3. 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。

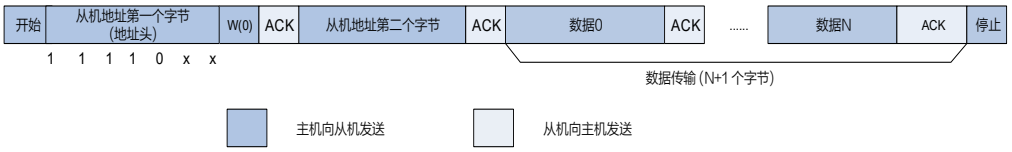
数据和地址都是 8 位传输，高位在前。START 信号之后的字节(在 7 位地址模式下是一个字节，10 位地址模式下是两个字节)是主机发送的从机地址。

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 25-4. 10 位地址的 I2C 通讯流程(主机发送)



注意： 在 10 位地址主机发送场景下，一个数据包发送完毕后，主机必须先发送一个停止信号，然后再发送一个开始信号来发起一个新的数据包的传输。

图 25-5. 7 位地址的 I2C 通讯流程(主机发送)

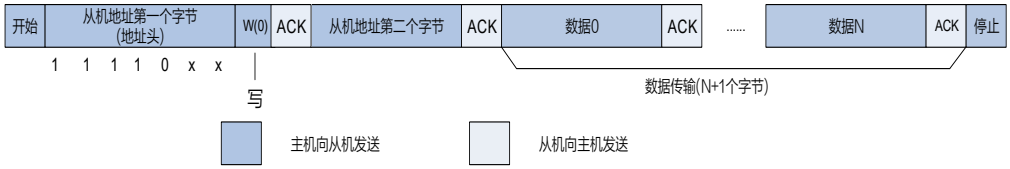
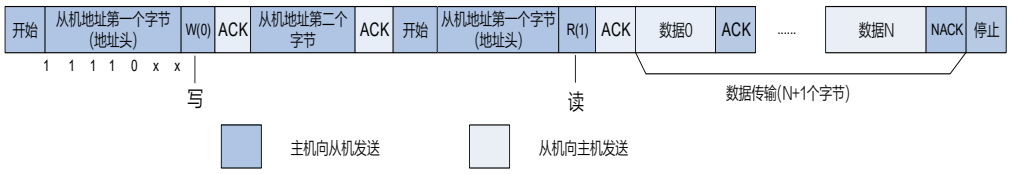


图 25-6. 7 位地址的 I2C 通讯流程(主机接收)



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 25-7. 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，当 HEAD10R=1 时，I2C 传输序列限制为主机发送数据完毕后紧接着主机接收数据。该传输序列如*图 25-8. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1)*所示。

图 25-7. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=0)

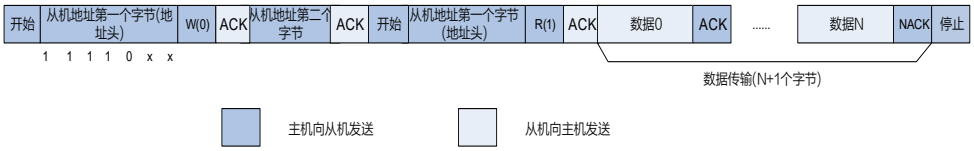
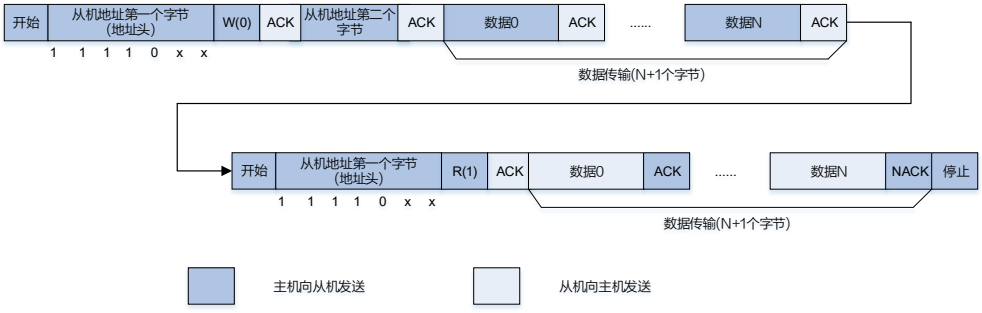


图 25-8. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1)



25.3.3. 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，模拟滤波器可以抑制脉冲宽度小于 50ns 的噪声。

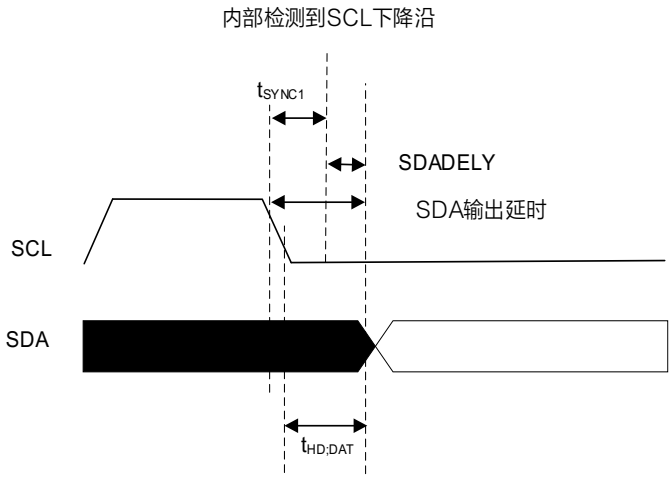
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

25.3.4. 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。

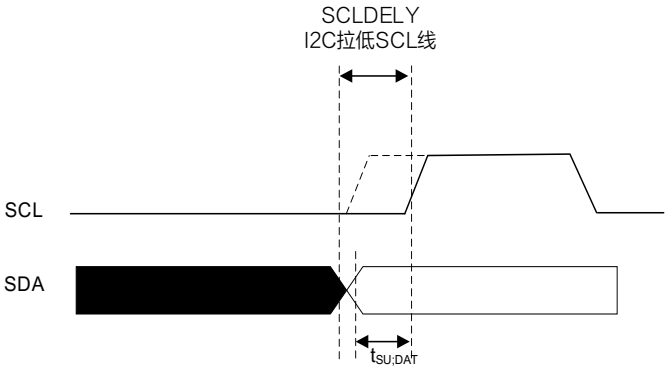
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 25-9. 数据保持时间*所示。

图 25-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如*图 25-10. 数据建立时间*所示。

图 25-10. 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+1)*(PSC+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 25-2. 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	

25.3.5. 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：

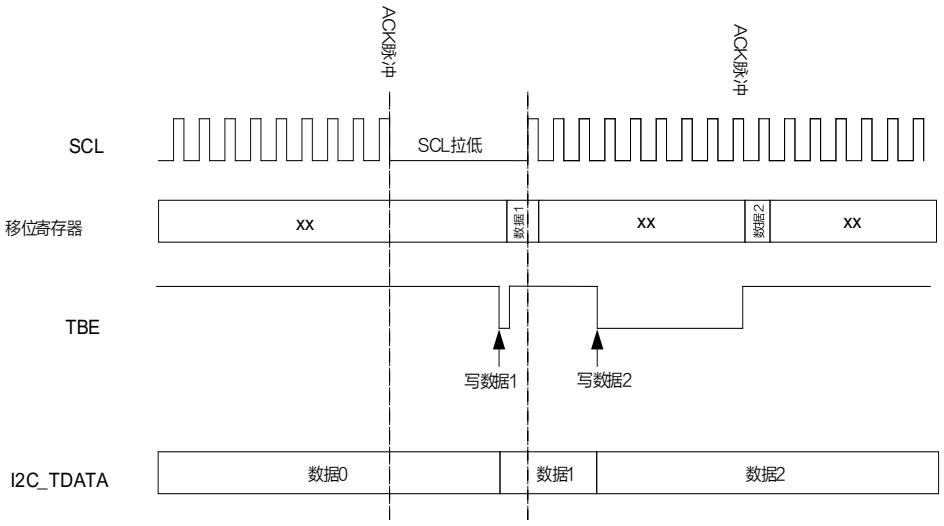
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

25.3.6. 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。

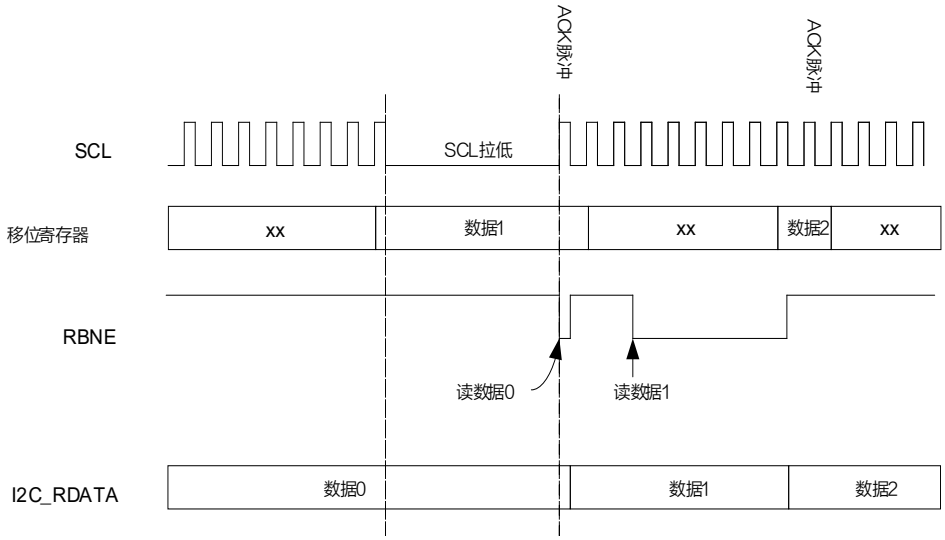
图 25-11. 数据发送



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 25-12. 数据接收



重载和自动结束模式

为了管理字节传输和中断如*表 25-3. 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 25-3. 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。

25.3.7. 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。

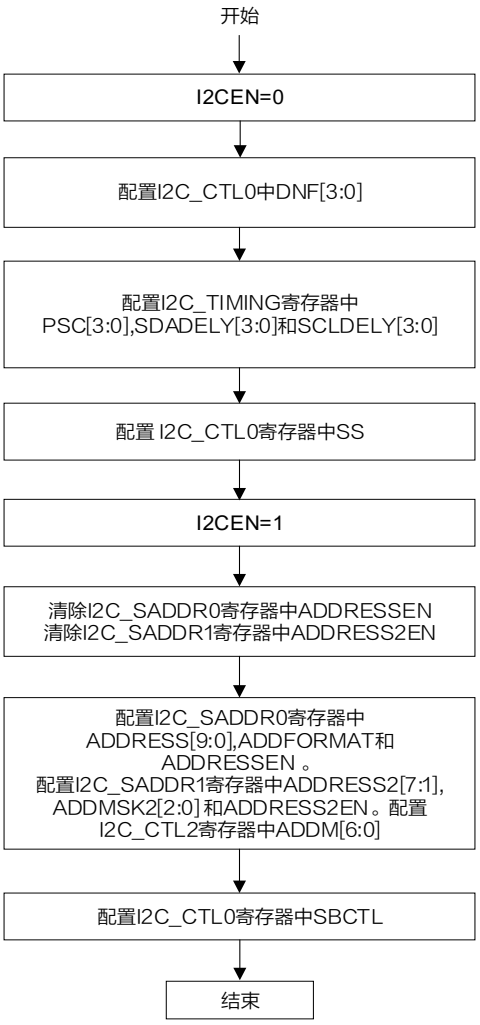
当 BYTENUM[7:0]大于 0x1 时，在 BYTENUM[7:0]数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

- I2CEN=0。
- 从机还未被寻址。
- ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 25-13. 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，但当作为 SMBus 从机发送器并启用 PEC 功能时，BYTENUM 需要在 START 置位之前配置。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 25-14. I2C 从机发送编程模型(SS=0)

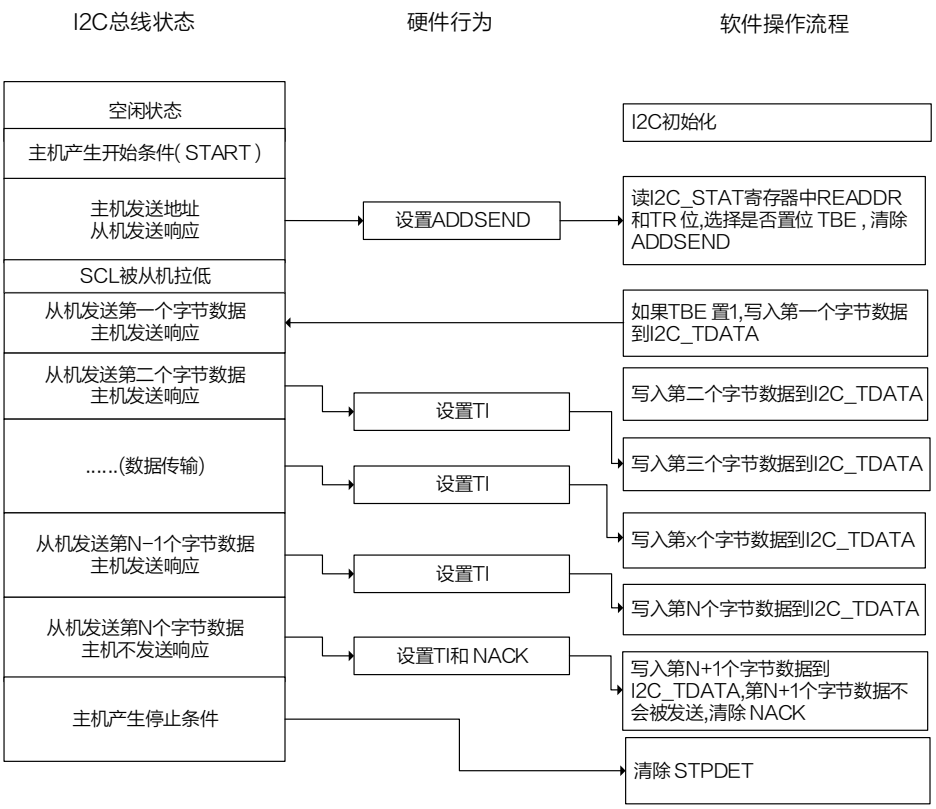


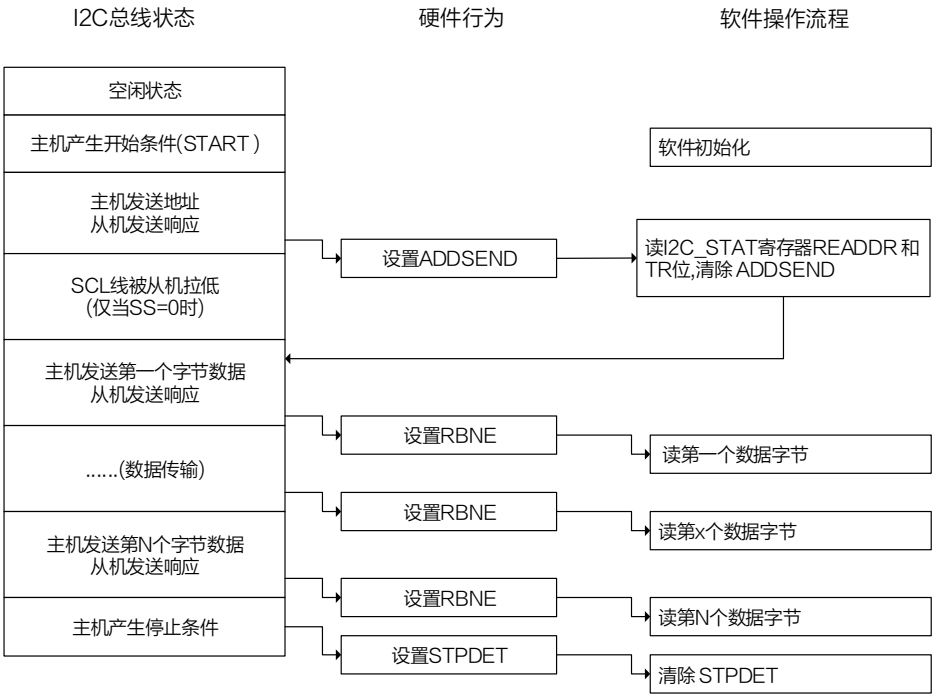
图 25-15. I2C 从机发送编程模型(SS=1)



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 25-16. I2C 从机接收编程模型



25.3.8. 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

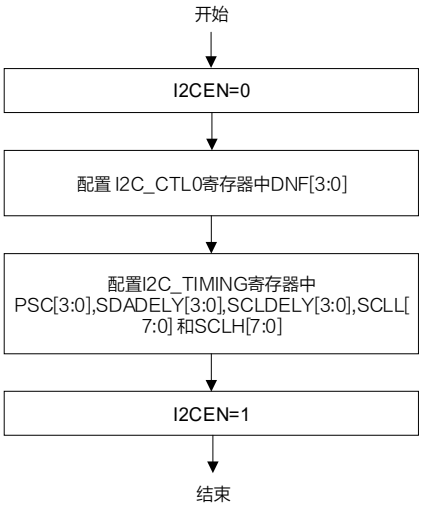
号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，主机切换成从机模式，START 位由硬件清零。当从机地址发送完成时，START 位由硬件清零。

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 25-17. 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 25-18. I2C 主机发送编程模型(N<=255)

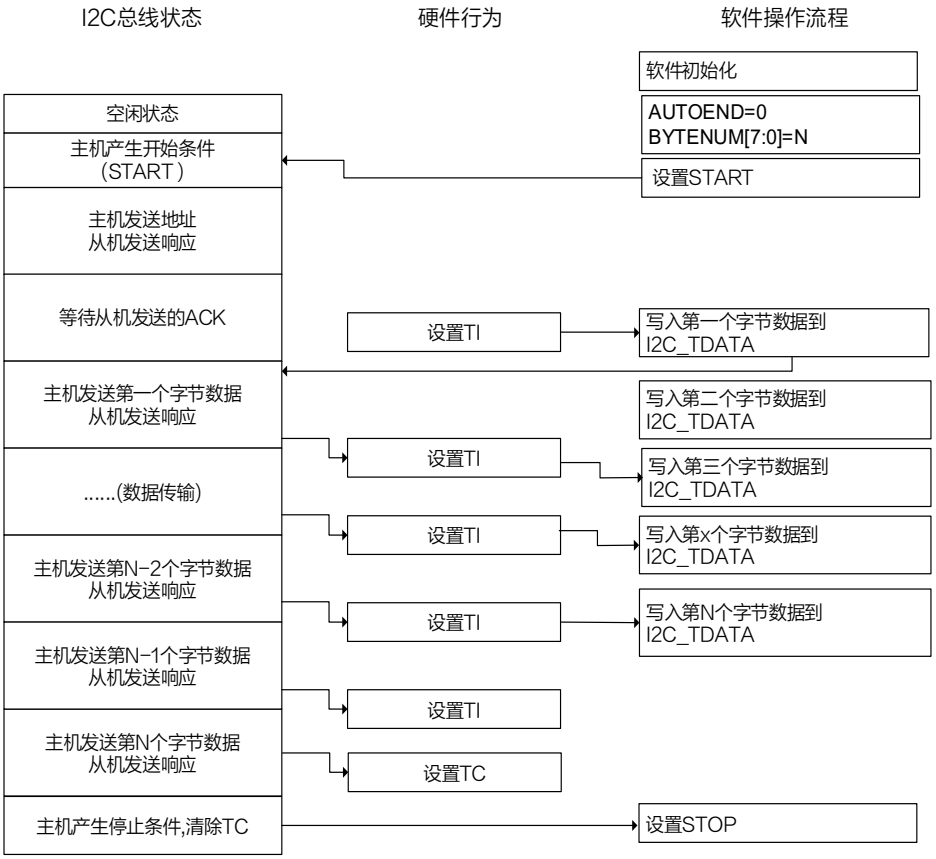


图 25-19. I2C 主机发送编程模型(N>255)



主机接收模式下的软件流程

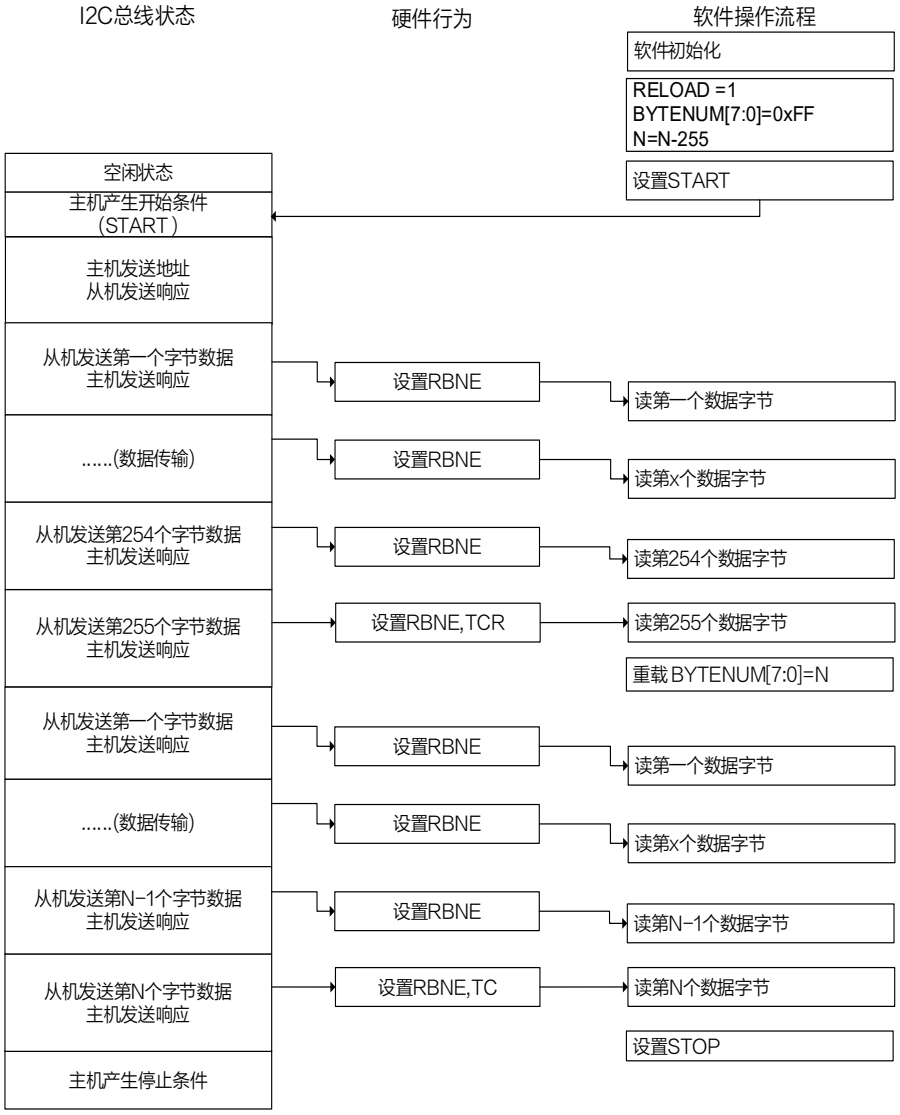
在主机接收模式下，当接收到一个字节时，I2C_STAT 寄存器中 RBNE 位置 1。如果 I2C_CTL0 寄存器中 RBNEIE 置 1，将产生一个中断。如果待接收字节数大于 255，必须将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1，在 BYTENUM[7:0]中写入一个非零值之前，SCL 被拉低。

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 25-20. I2C 主机接收编程模型(N<=255)



图 25-21. I2C 主机接收编程模型(N>255)



25.3.9. SMBus 支持

系统管理总线(System Management Bus，简写为 SMBus 或 SMB)是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信。SMBus 是 I2C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统(参见 Smart Battery Data)。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不

符合这些协议的 I2C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus 采用了 I2C 硬件以及 I2C 的硬件寻址方式，但在 I2C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。在此协议中，系统中的 host 与设备之间有一个重要的区别，即 host 具有分配地址的功能。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

超时特性

SMBus 有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为 10kHz--为了防止长时间锁死总线。I2C 在本质上可以视为一个"直流"总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续 I2C 通信。I2C 总线协议中并没有限制这个延时的上限，但在 SMBus 系统中，这个时间被限定为 25~35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种(问题)状态。这样就并不允许从设备将时钟拉低太长时间。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查$t_{TIMEOUT}$参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0] 和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于(BUSTOA+1)*2048*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOB[11:0]为从机校验$t_{LOW:SEXT}$，为主机校验$t_{LOW:MEXT}$。通过将 I2C_TIMEOUT 寄存器中 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于(BUSTOB+1)*2048*t_{I2CCLK}，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验。一个 PEC 字节(PEC 错误码)附加在每次传输结束。PEC 的计算方式是对所有消息字节(包含地址和读/写位)使用 CRC-8 计算校验和。CRC-8 多项式位x^8+x^2+x+1(CRC-8-ATM HEC 算法，初始化为 0)。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在 BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT#设备。如果 SMBALERT#电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备(SMBHAEN=0)时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备(SMBHAEN=1)，且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN=0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于$t_{HIGH,MAX}$，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能t_{IDLE}检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取t_{IDLE}参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于(BUSTOA+1)*4*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址(0b1100 001)。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址(0b0001 000)。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警

响应地址(0b0001 100)。

25.3.10. SMBus 模式

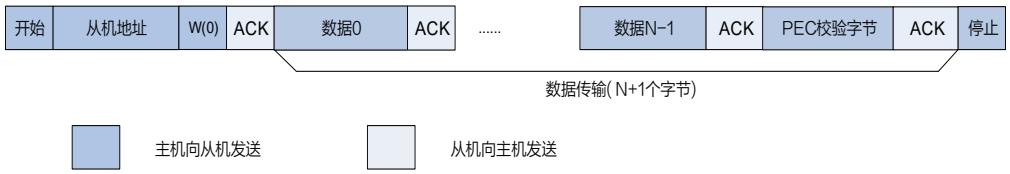
SMBus 主机发送器和从机接收器

当 SMBus 主机发送 PEC 时，必须在 START 位置 1 前，将 PECTRANS 位置 1 并在 BYTENUM[7:0]位域中配置字节数。在这种情况下，总 TI 中断数为 BYTENUM-1。因此，如果 BYTENUM=0x1 且 PECTRANS 位置 1，则 I2C_PEC 寄存器的数据将自动发送。如果 AUTOEND 为 1，SMBus 主机在 PEC 字节发送完成之后将自动发送 STOP 信号。如果 AUTOEND 为 0，SMBus 主机可以在 PEC 字节发送完成之后发送 RESTART 信号。I2C_PEC 寄存器中的数据将在 BYTENUM -1 个字节发送完成后发送，PEC 字节发送完成后 TC 位将置 1。SCL 线被拉低。RESTART 位必须在 TC 中断服务程序中置 1。

SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，且 PEC 值不匹配，PECERR 将会置 1 并产生中断。如果无须使用 ACK 控制，PECTRANS 可以设置为 1，BYTENUM 可以根据待接收字节数来配置。

注意： 在 RELAOD 位置 1 之后，PECTRANS 不可以被修改。

图 25-22. 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。

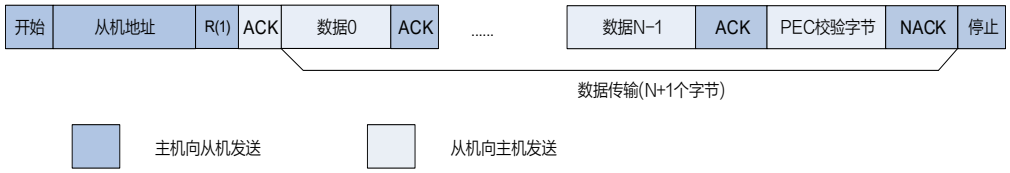
如果 SMBus 主机需要在接收到 PEC 字节之后产生 RESTART 信号，需要禁能自动结束模式。在 START 信号发送之前，PECTRANS 位必须置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在 PEC 字节发送完成之后 TC 位将置 1，SCL 线被拉低。在 TC 中断服务程序中可将 RESTART 位置 1。

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：

1. PECTRANS 位在 RELOAD 置 1 之后不能被修改。
2. SMBus 作为从机发送器,并且使用 PEC 功能时,必须在检测到 START 信号前配置 BYTENUM[7:0]。

图 25-23. SMBus 主机接收器和从机发送器通信流程



25.3.11. 从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从深度睡眠模式和深度睡眠模式 1 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 CK_IRC8M。在深度睡眠模式和深度睡眠模式 1 下，CK_IRC8M 关闭。当 I2C 检测到 START 信号时，CK_IRC8M 打开，I2C 会将 SCL 拉低直到 CK_IRC8M 被唤醒。在接收地址期间，IRC8M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，CK_IRC8M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断(ADDMIE = 1)能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN = 0，CK_IRC8M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒(WUEN = 0)，则在进入省电模式之前必须禁能 I2C 外设(I2CEN = 0)。

注意：只有 I2C2 的地址匹配才能从深度睡眠模式和深度睡眠模式 1 唤醒 MCU。

25.3.12. 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。

25.3.13. I2C 错误和中断

I2C 错误标志如*表 25-4. I2C 错误标志*所示。

表 25-4. 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 25-5. I2C 中断事件*所示。

表 25-5. 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	

25.3.14. I2C 调试模式

当控制器进入调试模式(Cortex®-M33 内核停止)，SMBus 超时定时器会根据 DBG 模块中的 I2Cx_HOLD 配置位选择继续正常工作还是停止工作。

25.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

I2C2 基地址：0x4000 C000

25.4.1. 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，包含深度睡眠模式和深度睡眠模式 1。

		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。

25.4.2. 控制寄存器 1(I2C_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者 ADDSEND 置 1 或者检测到 STOP 信号或者 I2CEN=0。 0：不传输 PEC 值。 1：传输的 PEC 值。

		注意：当 RELOAD=1 或者从机模式下 SBCTL=0 时，该位无效。
25	AUTOEND	主机模式下自动结束模式 0：当 BYTENUM[7:0]个字节传输完成后时，TC 位置 1。 1：当 BYTENUM[7:0]个字节传输完成后时，自动发送 STOP 信号。 **注意：**该位仅在 RELOAD=0 时有效。该位由软件置 1 和清 0。
24	RELOAD	重载模式使能 0：当 BYTENUM[7:0]个字节传输完成后时，传输结束。 1：当 BYTENUM[7:0]个字节传输完成后时，传输未结束，重载新的 BYTENUM[7:0]。每次 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1。 该位由软件置 1 和清 0。
23:16	BYTENUM[7:0]	待传输的字节数 这些用来编程待传输的字节数。当 SBCTL=0 时，这些位无效。 **注意：**当 START 位置 1 时，这些位不能被修改，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。

25.4.3. 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。

25.4.4. 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。

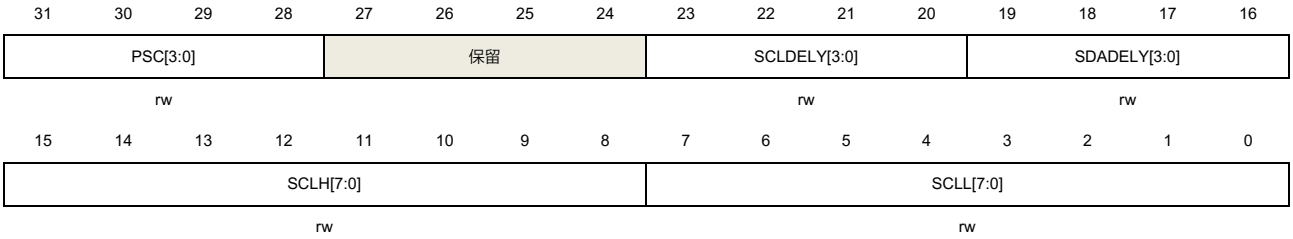
0	保留	必须保持复位值。

25.4.5. 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。

25.4.6. 超时寄存器(I2C_TIMEOUT)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTOEN	保留			BUSTOB[11:0]											
rw				rw											

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

位/位域	名称	描述
31	EXTOEN	时钟信号延展超时使能 当 SCL 累计拉低时间大于 $t_{LOW:EXT}$ 时，将会产生超时错误，$t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 0：时钟信号延展超时检测禁能。 1：时钟信号延展超时检测使能。
30:28	保留	必须保持复位值。
27:16	BUSTOB[11:0]	总线超时 B 配置累积时钟延展超时。在主机模式下，检测主机累计时钟低电平延展时间$t_{LOW:MEXT}$。从机模式下，检测从机累计时钟低电平延展时间$t_{LOW:SEXT}$。 $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 **注意：**该位域仅在 EXTOEN = 0 时可以被修改。
15	TOEN	时钟超时使能 当 TOIDLE = 0，SCL 拉低时间大于$t_{TIMEOUT}$或当 TOIDLE = 1，SCL 拉低时间大于t_{IDLE}，将检测到超时错误。 0：SCL 超时检测禁能 1：SCL 超时检测使能
14:13	保留	必须保持复位值。
12	TOIDLE	空闲时钟超时检测 0：BUSTOA 用于检测 SCL 低电平超时。 1：BUSTOA 用于检测 SCL 和 SDA 高电平超时(总线空闲条件)。 **注意：**该位域仅在 TOEN = 0 时可以被改写。
11:0	BUSTOA[11:0]	总线超时 A 当 TOIDLE=0 时，$t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK} 当 TOIDLE=1 时，t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK}

注意：该位域仅在 TOEN = 0 时可以被改写。

25.4.7. 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 引脚上未检测到 SMBALERT 事件。 1：在 SMBA 引脚上检测到 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。

25.4.8. 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			

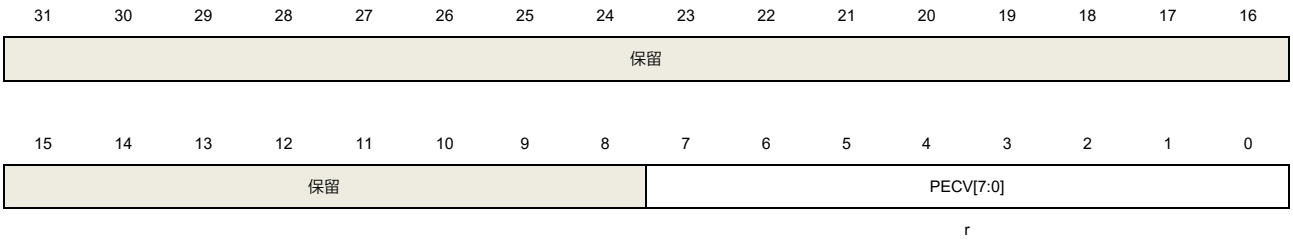
位/位域	名称	描述

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。

25.4.9. 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。

25.4.10. 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

25.4.11. 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。

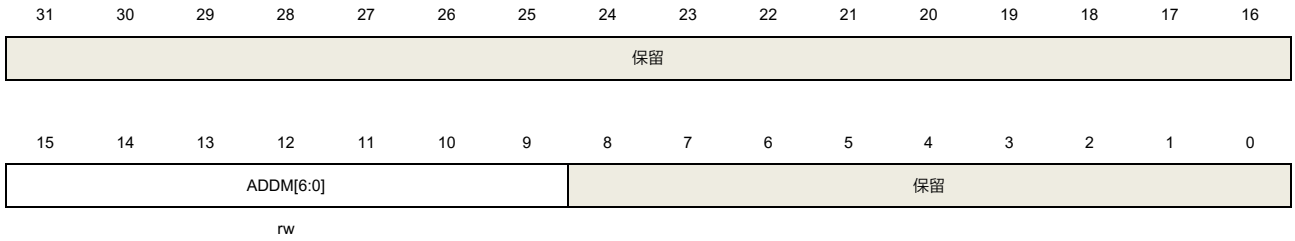


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

25.4.12. 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。

26. 串行外设接口/片上音频接口(SPI/I2S)

26.1. 简介

SPI/I2S 模块可以通过 SPI 协议或 I2S 音频协议与外部设备进行通信。

串行外设接口(Serial Peripheral Interface，缩写为 SPI)提供了基于 SPI 协议的数据发送和接收功能，可以工作于主机或从机模式。SPI 接口支持具有硬件 CRC 计算和校验的全双工和单工模式。SPI0 和 SPI3 还支持 SPI 四线主机模式。

片上音频接口(Inter-IC Sound，缩写为 I2S)支持四种音频标准，分别是 I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。通过使用附加的 I2S 模块：I2S1_ADD 和 I2S2_ADD，SPI1 和 SPI2 支持 I2S 全双工模式。

26.2. 主要特性

26.2.1. SPI 主要特性

- 具有全双工和单工模式的主从操作；
- 16位宽度，独立的发送和接收缓冲区；
- 8位或16位数据帧格式；
- 低位在前或高位在前的数据位顺序；
- 软件和硬件NSS管理；
- 硬件CRC计算、发送和校验；
- 发送和接收支持DMA模式；
- 支持SPI TI模式；
- 支持SPI NSS脉冲模式
- 支持SPI四线功能的主机模式(仅在SPI0和SPI3中)。

26.2.2. I2S 主要特性

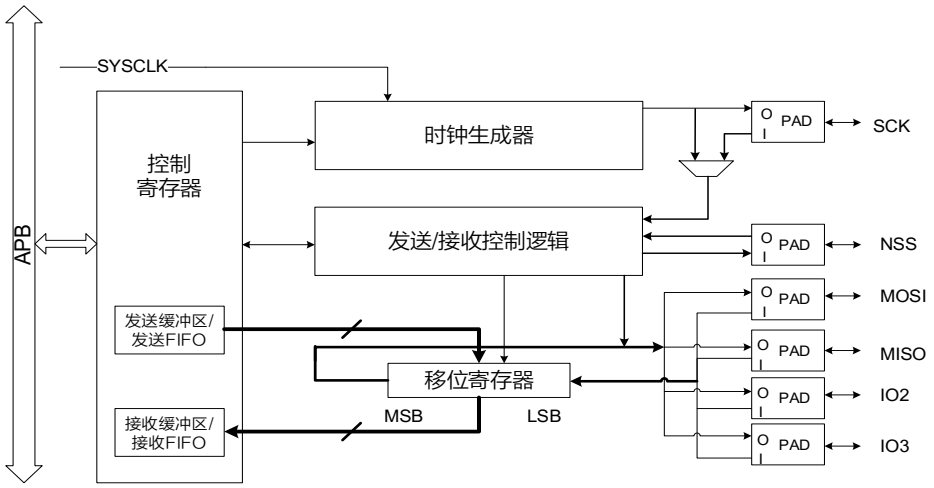
- 具有发送和接收功能的主从操作；
- 主机或从机操作支持全双工模式(仅适用于SPI1和SPI2)
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 16位缓冲区用于发送和接收；
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟(MCK)；

■ 发送和接收支持DMA功能。

26.3. SPI 功能说明

26.3.1. SPI 结构框图

图 26-1. SPI 结构框图



26.3.2. SPI 信号线描述

常规配置(非 SPI 四线模式)

表 26-1. 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

引脚名称	方向	描述
		式。 从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

SPI 四线配置

SPI 默认配置为单路模式，当 SPI_QCTL 中的 QMOD 位置 1 时，配置为 SPI 四线模式(只适用于 SPI0 和 SPI3)。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 26-2. 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 线
MISO	I / O	发送或接收数据 1 线
IO2	I / O	发送或接收数据 2 线
IO3	I / O	发送或接收数据 3 线
NSS	O	NSS 输出

26.3.3. SPI 时序和数据帧格式

SPI_CTL0 寄存器中的 CKPL 位和 CKPH 位决定了 SPI 时钟和数据信号的时序。CKPL 位决定了空闲状态时 SCK 的电平，CKPH 位决定了第一个或第二个时钟跳变沿为有效采样边沿。在 TI 模式下，这两位没有意义。

图 26-2. 常规模式下的 SPI 时序图

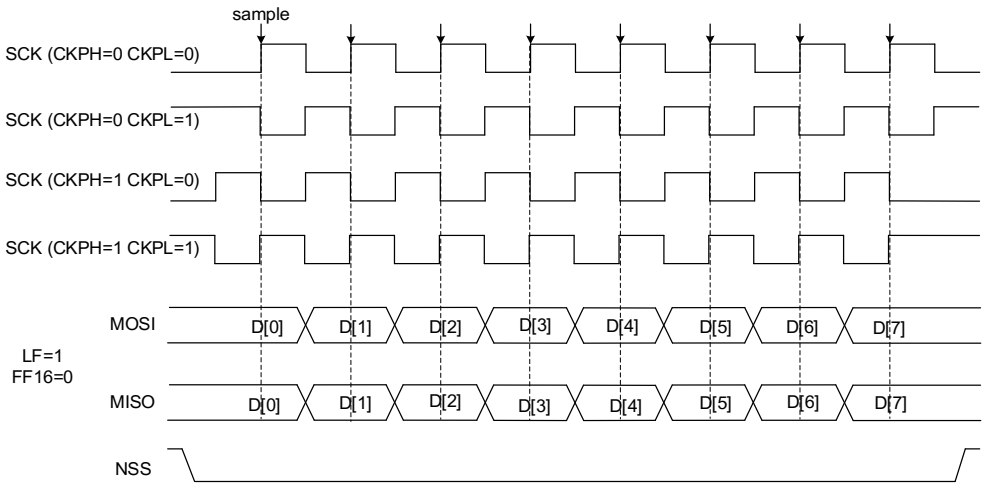
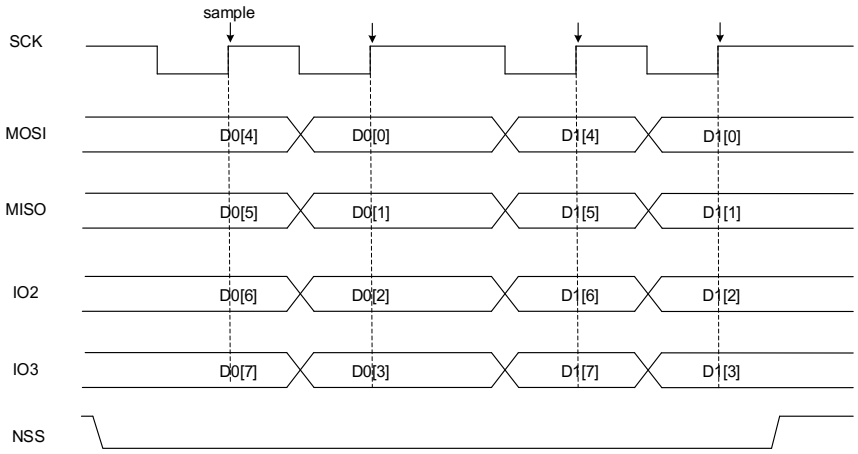


图 26-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



在常规模式中，通过 SPI_CTL0 中的 FF16 位配置数据长度，当 FF16=1 时，数据长度为 16 位，否则为 8 位。在 SPI 四线模式下，数据帧长度固定为 8 位。

通过设置 SPI_CTL0 中的 LF 位可以配置数据顺序，当 LF=1 时，SPI 先发送 LSB 位，当 LF=0 时，则先发送 MSB 位。在 TI 模式中，数据顺序固定为先发 MSB 位。

26.3.4. NSS 功能

从机模式

当配置为从机模式(MSTMOD=0)时，在硬件 NSS 模式(SWNSSEN = 0)下，SPI 从 NSS 引脚获取 NSS 电平，在软件 NSS 模式(SWNSSEN = 1)下，SPI 根据 SWNSS 位得到 NSS 电平。只有当 NSS 为低电平时，才能发送或接收数据。在软件 NSS 模式下，不使用 NSS 引脚。

表 26-3. 从机模式 NSS 功能

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0：NSS 电平为低 SWNSS = 1：NSS 电平为高

主机模式

在主机模式(MSTMOD=1)下，如果应用程序使用多主机连接方式，NSS 可以配置为硬件输入模式(SWNSSEN=0, NSSDRV=0)或者软件模式(SWNSSEN=1)。一旦 NSS 引脚(在硬件 NSS 模式下)或 SWNSS 位(在软件 NSS 模式下)被拉低，SPI 将自动进入从机模式，并且产生主机

配置错误，CONFERR 位置 1。

如果应用程序希望使用 NSS 引脚控制 SPI 从设备，NSS 应该配置为硬件输出模式(SWNSSEN=0, NSSDRV=1)。使能 SPI 后，NSS 变为低电平。

应用程序可以使用一个通用 I/O 口作为 NSS 引脚，以实现更加灵活的 NSS 应用。

表 26-4. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV：不要求	适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

26.3.5. SPI 运行模式

表 26-5. SPI 运行模式

模式	描述	寄存器配置	使用的数据引脚
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	MOSI：发送 MISO：不使用

模式	描述	寄存器配置	使用的数据引脚
		BDEN = 1 BDOEN = 1	
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 BDOEN：不要求	MOSI：不使用 MISO：发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI：不使用 MISO：发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI：不使用 MISO：接收

图 26-4. 典型的全双工模式连接

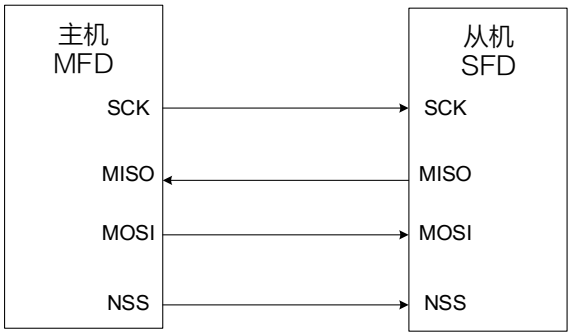


图 26-5. 典型的单工模式连接(主机：接收，从机：发送)

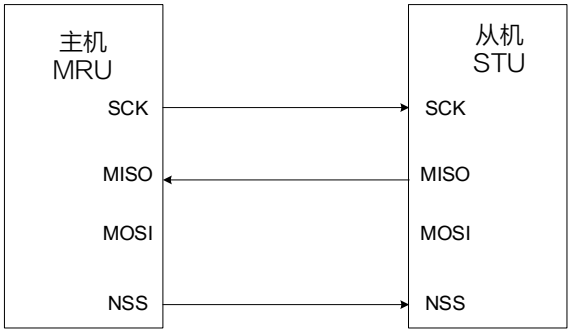


图 26-6. 典型的单工模式连接(主机：只发送，从机：接收)

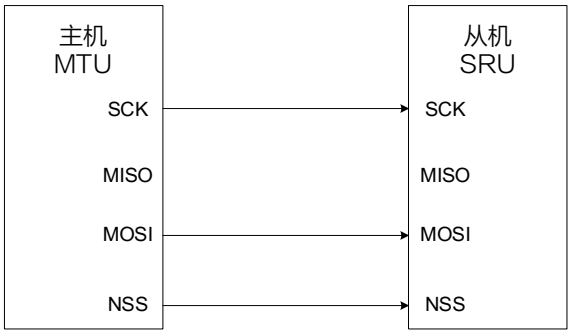
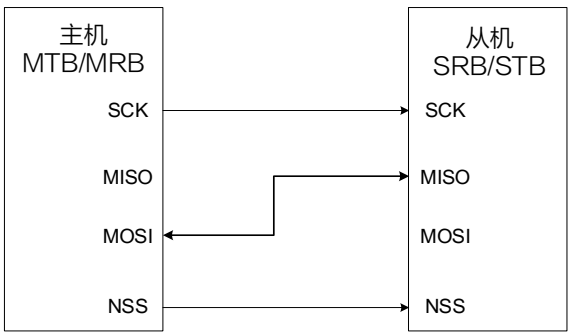


图 26-7. 典型的双向线连接



SPI 初始化流程

在发送或接收数据之前，应用程序应遵循如下的 SPI 初始化流程：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置数据格式(SPI_CTL0中的FF16位)。
3. 配置时钟时序(SPI_CTL0中的CKPL位和CKPH位)。
4. 配置帧格式(SPI_CTL0中的LF位)。
5. 按照上文***NSS功能***的描述，根据应用程序的需求，配置NSS模式(SPI_CTL0中的SWNSSEN位和NSSDRV位)。

6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1。否则，忽略此步骤。
7. 如果工作在NSSP模式，需要将SPI_CTL1中的NSSP位置1，否则，忽略此步骤。
8. 根据上面描述的运行模式，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1。如果不是，则忽略此步骤。
10. 使能SPI(将SPIEN位置1)。

注意：在通信过程中，不应更改 CKPH、CKPL、MSTMOD、PSC[2:0]和 LF 位。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI 模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当 SCK 引脚上的 SCK 信号开始翻转，且 NSS 引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当 SPI 开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE(发送缓冲区空)位置 1。TBE 标志位置 1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写 SPI_DATA 寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入 SPI_DATA 寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且 RBNE(接收缓冲区非空)位置 1。软件通过读 SPI_DATA 寄存器获得接收的数据，此操作会自动清除 RBNE 标志位。在 MRU 和 MRB 模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式(MFD)中，当发送缓冲区非空时，硬件只接收下一个数据帧。

SPI 不同模式下的操作流程(非 SPI 四线模式，TI 模式或 NSSP 模式)

在全双工模式下，无论是 MFD 模式或者 SFD 模式，应用程序都应该监视 RBNE 标志位和 TBE 标志位，并且遵循上文描述的操作流程。

除了忽略 RBNE 位和 RXORERR 位，且只执行上述的发送流程之外，发送模式(MTU，MTB，STU 和 STB)与全双工模式类似。

在主机接收模式(MRU 或 MRB)下，全双工模式和发送模式是不同的。在 MRU 模式或 MRB 模式下，在 SPI 使能后，SPI 产生连续的 SCK 信号，直到 SPI 停止。所以，软件应该忽略 TBE 标志位，并且在 RBNE 位置 1 后及时读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略 TBE 标志位，且只执行上述的接收流程之外，从机接收模式(SRU 或 SRB)与全双工模式类似。

SPI TI 模式

SPI TI 模式将 NSS 作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式(MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB 和 SRB)都支持 TI 模式。但是，在 TI 模式中，SPI_CTL0 中的 CKPL 位和 CKPH 位是没有意义的，SCK 信号的采样边沿为下降沿。

图 26-8. 主机 TI 模式在不连续发送时的时序图

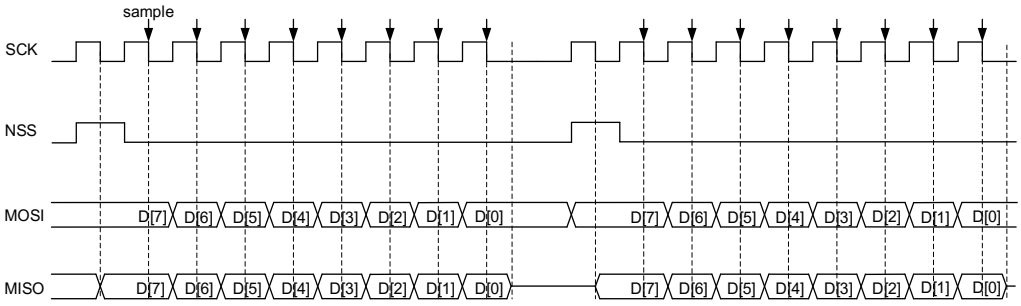
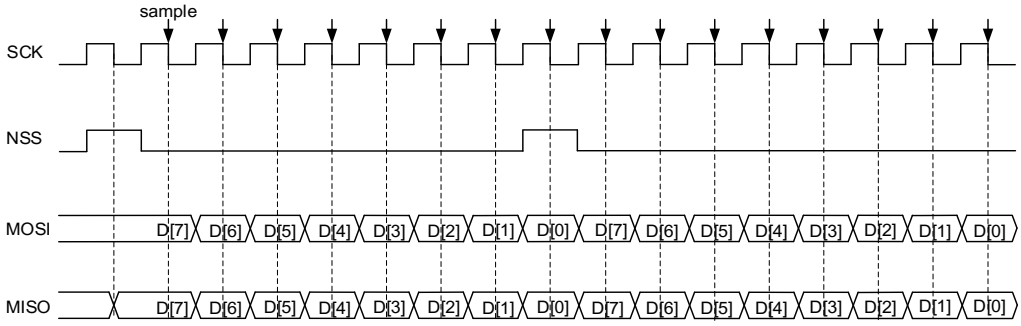
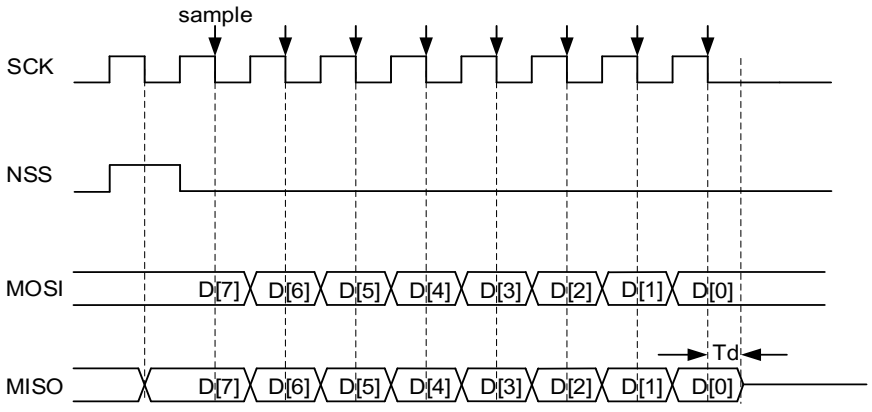


图 26-9. 主机 TI 模式在连续发送时的时序图



在主机 TI 模式下，SPI 模块可实现连续传输或者不连续传输。如果主机写 SPI_DATA 的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。但是在连续传输中，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 26-10. 从机 TI 模式时序图



在从机 TI 模式中，在 SCK 信号的最后一个上升沿，从机开始发送最后一个字节的 LSB 位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在 SCK 信号的下降沿之后继续驱动该位一段时间，这段时间称为T_d，T_d通过 SPI_CTL0 寄存器中的 PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (26\text{-}1)$$

例如，如果 PSC[2:0] = 010，那么T_d数值为 9*Tpclk。

在从机模式下，从机需要监视 NSS 信号，如果检测到错误的 NSS 信号，将会置位 FERR 标志位。例如，NSS 信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

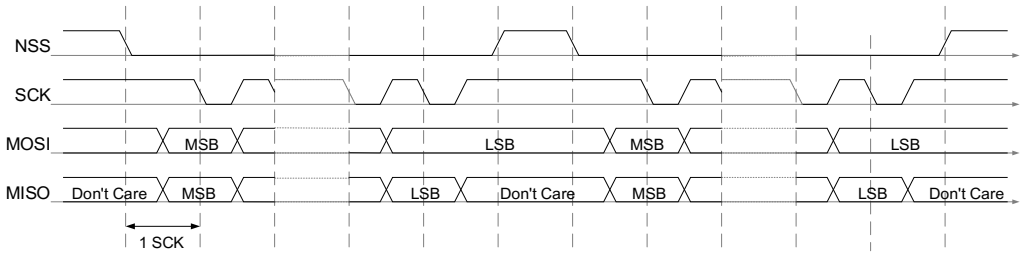
配置 SPI_CTL1 寄存器中的 NSSP 位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通 SPI 协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：NSSP = 1， MSTMOD = 1，CKPH = 0。

当使用 NSS 脉冲模式时，根据内部数据发送缓冲区的状态，NSS 脉冲会在两个连续的数据帧之间产生，且持续时间至少为 1 个 SCK 时钟周期。如果数据发送缓冲区保持为空，可能会持续多个 SCK 时钟周期。NSS 脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了 NSS 脉冲模式在主机连续发送时的时序图。

图 26-11. NSS 脉冲模式时序图(主机连续发送)



SPI 四线模式操作流程

SPI 四线模式用于控制四线 SPI flash 外设。

要配置成 SPI 四线模式，首先要确认 TBE 位置 1，且 TRANS 位清零，然后将 SPI_QCTL 寄存器中的 QMOD 位置 1。在 SPI 四线模式，SPI_CTL0 寄存器中 BDEN 位、BDOEN 位、CRCEN 位、CRCNT 位、FF16 位、RO 位和 LF 位保持清零，且 MSTMOD 位置 1，以保证 SPI 工作于主机模式。SPIEN 位、PSC 位、CKPL 位和 CKPH 位根据需要进行配置。

SPI 四线模式有两种运行模式：四线写模式和四线读模式，通过 SPI_QCTL 寄存器中的 QRD 位进行配置。

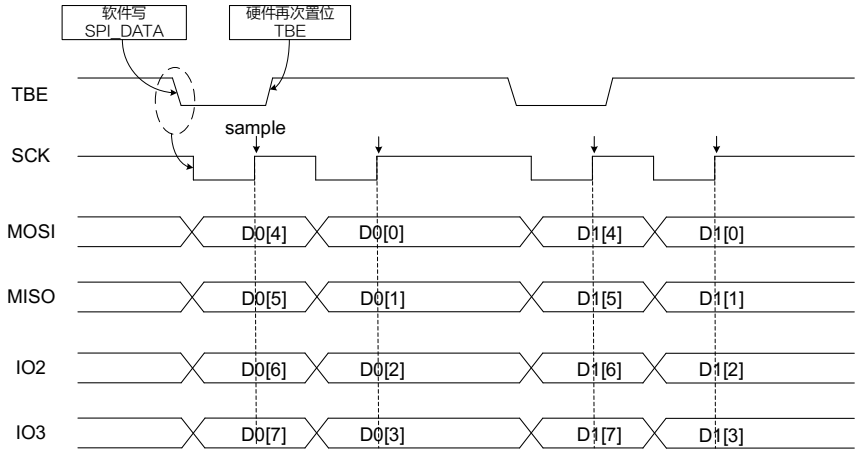
四线写模式

当 SPI_QCTL 寄存器中的 QMOD 位置 1 且 QRD 位清零时，SPI 工作在四路写模式。在四路写模式中，MOSI、MISO、IO2 和 IO3 都用作输出引脚。在 SCK 产生时钟信号后，一旦数据写入 SPI_DATA 寄存器(TBE 位清零)且 SPIEN 位置 1 时，SPI 将会通过这四个引脚发送写入的数据。一旦 SPI 开始数据传输，它总是在数据帧结束时检查 TBE 标志位的状态，若不能满足条件则停止传输。

四路模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 26-12. SPI 四线模式四线写操作时序图



四线读模式

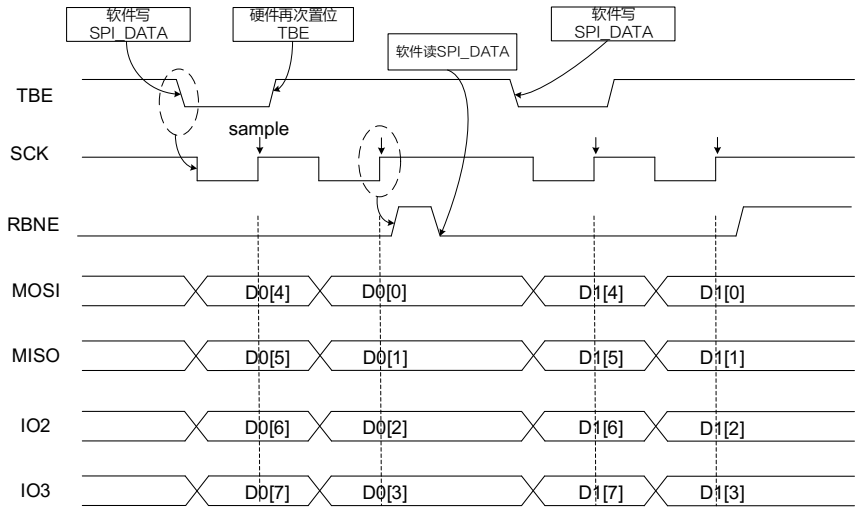
当 SPI_QCTL 寄存器中的 QMOD 位和 QRD 位都置 1 时，SPI 工作在四路读模式。在四路读模式中，MOSI、MISO、IO2 和 IO3 都用作输入引脚。当数据写入 SPI_DATA 寄存器(此时 TBE 位被清零)且 SPIEN 位置 1 时，SPI 开始在 SCK 信号线上产生时钟信号。写数据到 SPI_DATA 寄存器只是为了产生 SCK 时钟信号，所以可以写入任何数据。SPI 开始数据传输之后，每发送一个数据帧都要检测 SPIEN 位和 TBE 位，若条件不满足则停止传输。所以软件需要一直向 SPI_DATA 写空闲数据，以产生 SCK 时钟信号。

四路模式下接收操作流程：

1. 根据应用需求，配置 SPI_CTL0 和 SPI_CTL1 中时钟预分频、时钟极性、相位等参数；
2. 将 SPI_QCTL 中的 QMOD 位和 QRD 位置 1，然后将 SPI_CTL0 中的 SPIEN 位置 1 来使能 SPI 功能；
3. 写任意数据(例如 0xFF)到 SPI_DATA 寄存器；
4. 等待 RBNE 位置 1，然后读 SPI_DATA 寄存器来获取接收的数据；

5. 写任意数据(例如 0xFF)到 SPI_DATA 寄存器，以接收下一个字节数据。

图 26-13. SPI 四路模式四路读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止 SPI 功能。

MFD SFD

等待最后一个 RBNE 位并接收最后一个数据，等待 TBE=1 和 TRANS=0。最后，通过清零 SPIEN 位关闭 SPI。

MTU MTB STU STB

将最后一个数据写入 SPI_DATA 寄存器，等待 TBE 位置 1。然后等待 TRANS 位清零，通过清零 SPIEN 位关闭 SPI。

MRU MRB

等待倒数第二个 RBNE 位置 1，从 SPI_DATA 寄存器读数据，等待一个 SCK 时钟周期，然后通过清零 SPIEN 位关闭 SPI。等待最后一个 RBNE 位置 1，并从 SPI_DATA 读数据。

SRU SRB

当应用程序不想接收数据时，可以禁用 SPI，然后等待 TRANS=0 以确保正在进行的传输完成。

TI 模式

TI 模式的停止流程与上面描述过程相同。

NSS 脉冲模式

NSS 脉冲模式的停止流程与上面描述过程相同。

SPI 四路模式

在禁用 SPI 四路模式或者关闭 SPI 功能之前，软件应该先检查：TBE 位置 1，TRANS 位清零，SPI_QCTL 中的 QMOD 位和 SPI_CTL0 中的 SPIEN 位清零。

26.3.6. DMA 功能

DMA 功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位 SPI_CTL1 寄存器中的 DMATEN 位和 DMAREN 位，使能 SPI 模式的 DMA 功能。为了使用 DMA 功能，软件首先应当正确配置 DMA 模块，然后通过初始化流程配置 SPI 模块，最后使能 SPI。

SPI 使能后，如果 DMATEN 位置 1，每当 TBE=1 时，SPI 将会发出一个 DMA 请求，然后 DMA 应答该请求，并自动写数据到 SPI_DATA 寄存器。如果 DMAREN 位置 1，每当 RBNE=1 时，SPI 将会发出一个 DMA 请求，然后 DMA 应答该请求，并自动从 SPI_DATA 寄存器读取数据。

26.3.7. CRC 功能

SPI 模块包含两个 CRC 计算单元：分别用于发送数据和接收数据。CRC 计算单元使用 SPI_CRCPOLY 寄存器中定义的多项式。

通过配置 SPI_CTL0 中的 CRCEN 位使能 CRC 功能。对于数据线上每个发送和接收的数据，CRC 单元逐位计算 CRC 值，计算得到的 CRC 值可以从 SPI_TCRC 寄存器和 SPI_RCRC 寄存器中读取。

为了传输计算得到的 CRC 值，应用程序需要在最后一个数据写入发送缓冲区之后，设置 SPI_CTL0 中的 CRCNT 位。在全双工模式(MFD 或 SFD)，当 SPI 发送一个 CRC 值并且准备校验接收到的 CRC 值时，会将最新接收到的数据当作 CRC 值。在接收模式(MRB，MRU，SRU 和 SRB)下，在倒数第二个数据帧被接收后，软件应该把 CRCNT 位置 1。在 CRC 校验失败时，CRCERR 错误标志位将会置 1。

如果使能了 DMA 功能，软件不需要设置 CRCNT 位，硬件将会自动处理 CRC 传输和校验。

注意：当 SPI 处于从机模式且 CRC 功能使能时，无论 SPI 是否使能，CRC 计算器都对输入 SCK 时钟敏感。只有当时钟稳定时，软件才能启用 CRC，以避免错误的 CRC 计算。当 SPI 作为从机工作时，在数据阶段和 CRC 阶段之间，内部 NSS 信号需要保持低电平。

当配置 SPI 为从模式并且使用 CRC 功能时，即使 NSS 引脚为高时仍然会执行 CRC 的计算(当 NSS 信号为高时，只要 SCK 引脚上有时钟脉冲，则 CRC 计算会继续执行)。当主设备交替地与多个从设备进行通信时，将会出现这种情况，此时建议在 NSS 信号为低时重启 CRC 功能。当从设备未选中(NSS 信号为高)转换到被选中为一个新的从设备(NSS 信号为低)的时候，为了保持主从设备端下次 CRC 计算结果的同步，应该清除主从两端的 CRC 数值。建议按照下述步骤清除 CRC 数值：

1. 关闭 SPI 模块(SPIEN=0)；
2. 清除 CRCEN 位(CRCEN=0)；

3. 设置 CRCEN 位(CRCEN=1)；

4. 使能 SPI 模块(SPIEN=1)。

26.3.8. SPI 中断

状态标志位

■ **发送缓冲区空标志位(TBE)**

当发送缓冲区为空时，TBE 置位。软件可以通过写 SPI_DATA 寄存器将下一个待发送数据写入发送缓冲区。

■ **接收缓冲区非空标志位(RBNE)**

当接收缓冲区非空时，RBNE 置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读 SPI_DATA 寄存器来读取此数据。

■ **SPI通信进行中标志位(TRANS)**

TRANS 位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

■ **配置错误标志(CONFERR)**

在主机模式中，CONFERR 位是一个错误标志位。在硬件 NSS 模式中，如果 NSSDRV 没有使能，当 NSS 被拉低时，CONFERR 位被置 1。在软件 NSS 模式中，当 SWNSS 位为 0 时，CONFERR 位置 1。当 CONFERR 位置 1 时，SPIEN 位和 MSTMOD 位由硬件清除，SPI 关闭，设备强制进入从机模式。

在 CONFERR 位清零之前，SPIEN 位和 MSTMOD 位保持写保护，从机的 CONFERR 位不能置 1。在多主机配置中，设备可以在 CONFERR 位置 1 时进入从机模式，这意味着发生了系统控制的多主冲突。

■ **接收过载错误(RXORERR)**

在 RBNE 位为 1 时，如果再有数据被接收，RXORERR 位将会置 1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **帧错误(FERR)**

在 TI 从机模式下，从机也要监视 NSS 信号，如果检测到错误的 NSS 信号，将会置位 FERR 标志位。例如，NSS 信号在一个字节的中间位发生翻转。

■ **CRC错误(CRCERR)**

当 CRCEN 位置 1 时，SPI_RCRC 寄存器中接收到的 CRC 值将会和紧随着最后一帧数据接收到的 CRC 值进行比较。当两者不同时，CRCERR 位将会置 1。

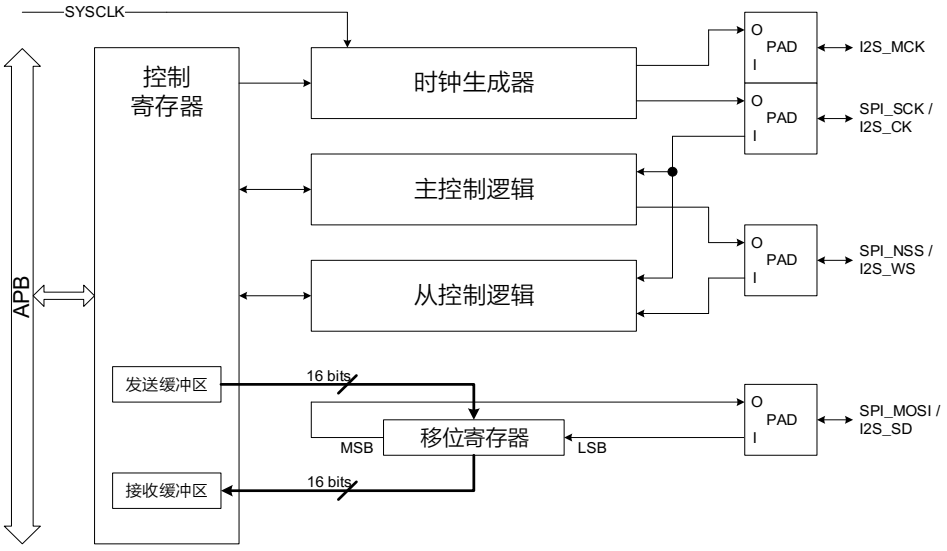
表 26-6. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE
RXORERR	接收过载错误	读SPI_DATA寄存器，然后读 SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

26.4. I2S 功能说明

26.4.1. I2S 结构框图

图 26-14. I2S 结构框图



I2S 功能有 5 个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成 I2S 通信时钟。主机控制逻辑用来在主机模式下生成 I2S_WS 信号并控制通信。从机控制逻辑根据接收到的 I2SCK 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。

26.4.2. I2S 信号线描述

I2S 接口有 4 个引脚，分别是 I2S_CK、I2S_WS、I2S_SD 和 I2S_MCK。I2S_CK 是串行时钟信号，与 SPI_SCK 共享引脚。I2S_WS 是数据帧控制信号，与 SPI_NSS 共享引脚。I2S_SD 是串行数据信号，与 SPI_MOSI 共享引脚。I2S_MCK 是主时钟信号，它提供了一个 256 倍于 Fs 的时钟频率，其中 Fs 是音频采样率。

26.4.3. I2S 音频标准

I2S 音频标准是通过设置 SPI_I2SCTL 寄存器中的 I2SSTD 位来选择的，可以选择四种音频标准：I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。除 PCM 之外的所有标准都是两个通道(左通道和右通道)的音频数据分时复用 I2S 接口的，并通过 I2S_WS 信号来区分当前数据属于哪个通道。对于 PCM 标准，I2S_WS 信号表示帧同步信息。

数据长度和通道长度可以通过 SPI_I2SCTL 寄存器中的 DTLEN 位和 CHLEN 位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16 位数据打包成 16 位数据帧格式，16 位数据打包成 32 位数据帧格式，24 位数据打包成 32 位数据帧格式，32 位数据打包成 32 位数据帧格式。用于发送和接收的数据缓冲区都是 16 位宽度。所以，要完成数据长度为 24 位或 32 位的数据帧传输，SPI_DATA 寄存器需要被访问 2 次；而要完成数据长度为 16 位的数据帧传输，SPI_DATA 寄存器只需被访问 1 次。如需将 16 位数据打包成 32 位数据帧，硬件会自动插入 16 位 0 将 16 位数据扩展为 32 位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于 I2S 飞利浦标准，I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化，I2S_WS 在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 26-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

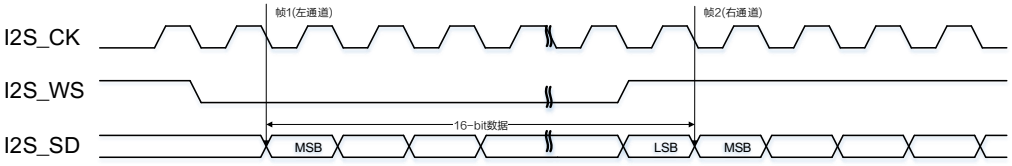
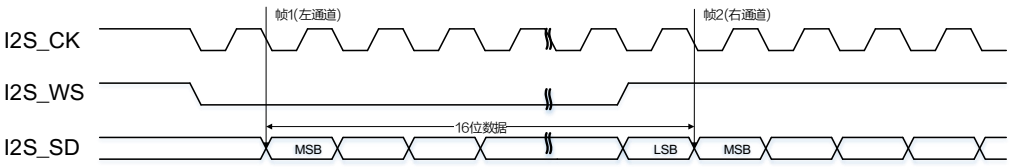


图 26-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一

次。

图 26-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

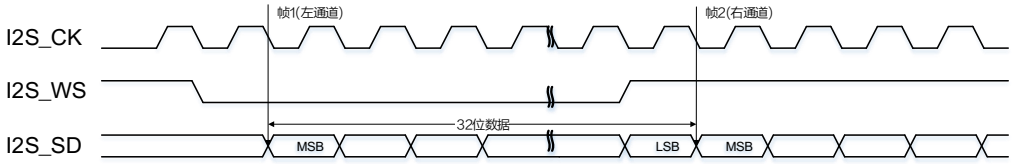
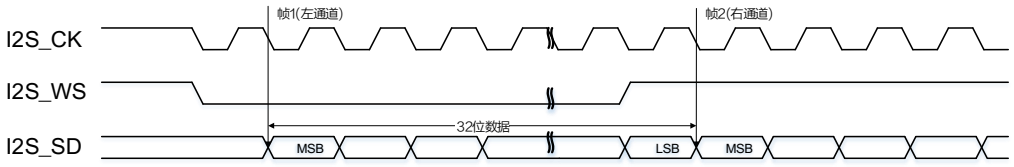


图 26-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



当 32 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要访问 SPI_DATA 寄存器 2 次。在发送模式下，如果要发送一个 32 位数据，第一个写入 SPI_DATA 寄存器的数据应该是高 16 位数据，第二个数据应该是低 16 位数据。在接收模式下，如果要接收一个 32 位数据，第一个从 SPI_DATA 寄存器读到的数据应该是高 16 位数据，第二个数据应该是低 16 位数据。

图 26-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

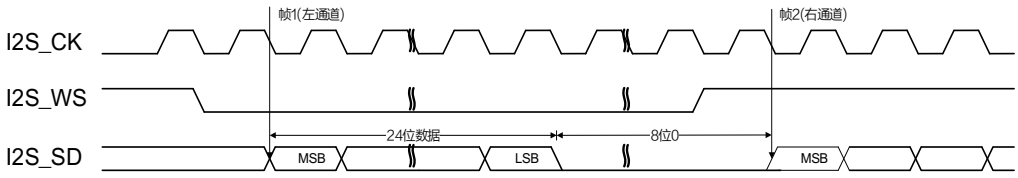
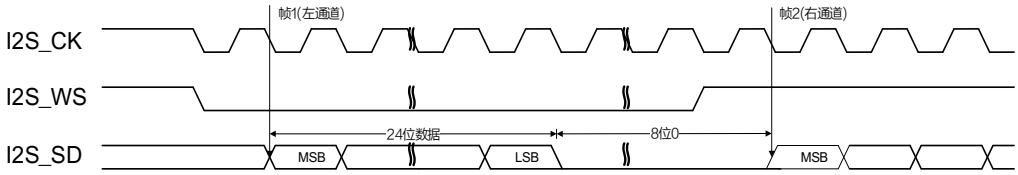


图 26-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当 24 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要访问 SPI_DATA 寄存器 2 次。在发送模式下，如果要发送一个 24 位数据 D[23:0]，第一个写入 SPI_DATA 寄存器的数据应该是高 16 位数据 D[23:8]，第二个数据应该是一个 16 位数据，该 16 位数据的高 8 位是 D[7:0]，低 8 位数据可以是任意值。在接收模式下，如果要接收一个 24 位数据 D[23:0]，第一个从 SPI_DATA 寄存器读到的数据应该是高 16 位数据 D[23:8]，第二个数据应该是一个 16 位数据，该 16 位数据的高 8 位是 D[7:0]，低 8 位数据全是 0。

图 26-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

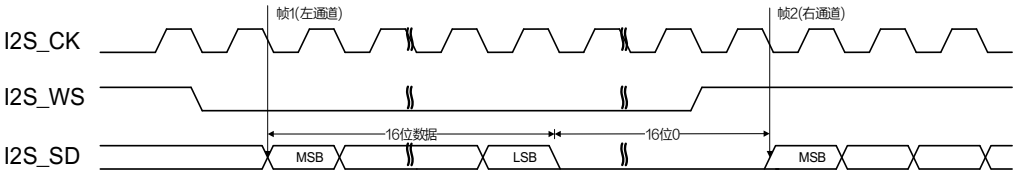
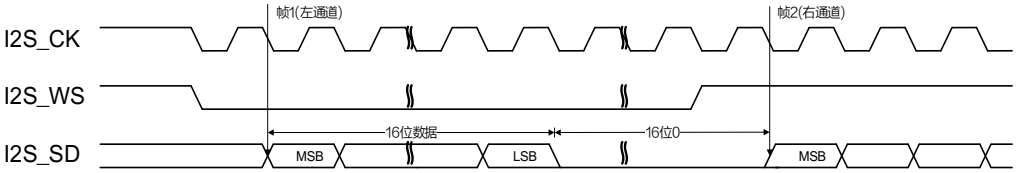


图 26-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



当 16 位数据打包成 32 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。为了将该 16 位数据扩展成 32 位数据，剩下的 16 位被硬件强制填充为 0x0000。

MSB 对齐标准

对于 MSB 对齐标准，I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化。SPI_DATA 寄存器的处理方式与 I2S 飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 26-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

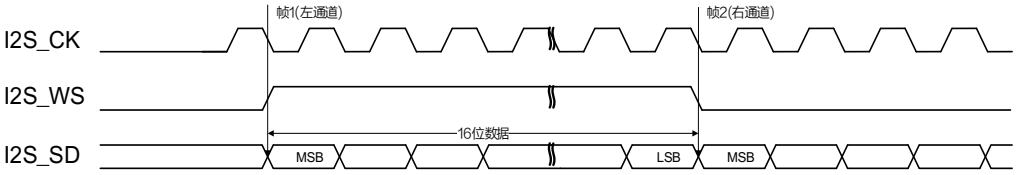


图 26-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

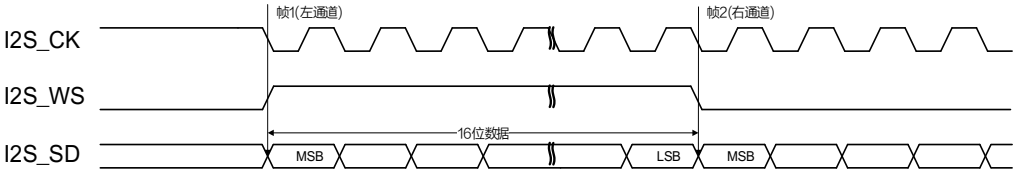


图 26-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

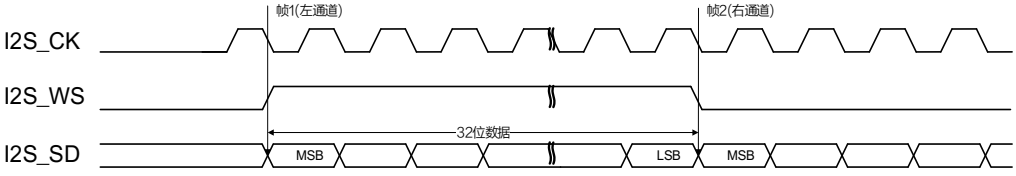


图 26-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)

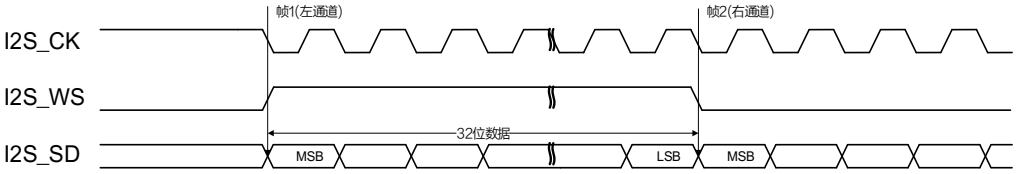


图 26-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

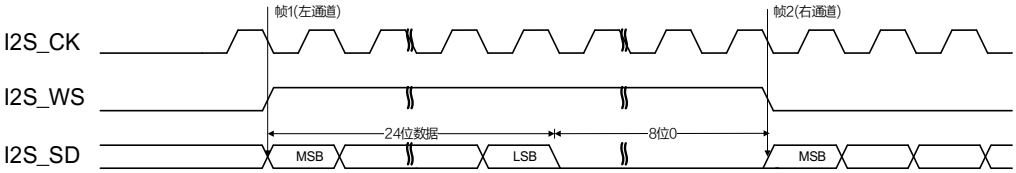


图 26-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

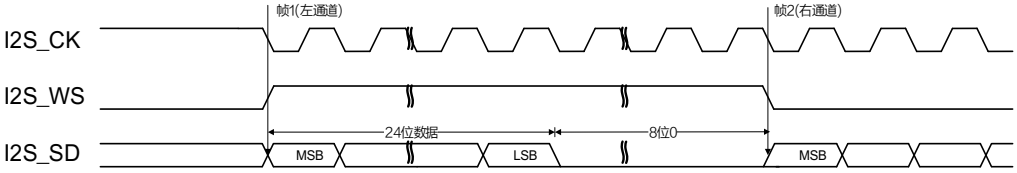


图 26-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

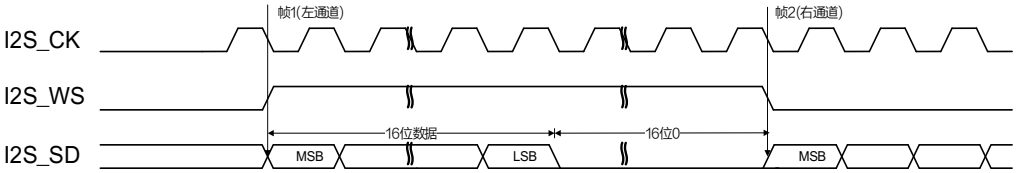
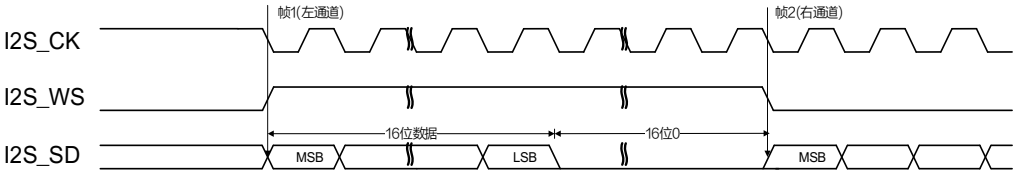


图 26-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



LSB 对齐标准

对于 LSB 对齐标准，I2S_WS 和 I2S_SD 在 I2S_CK 的下降沿变化。在通道长度与数据长度相同的情况下，LSB 对齐标准和 MSB 对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB 对齐标准的有效数据与最低位对齐，而 MSB 对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 26-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

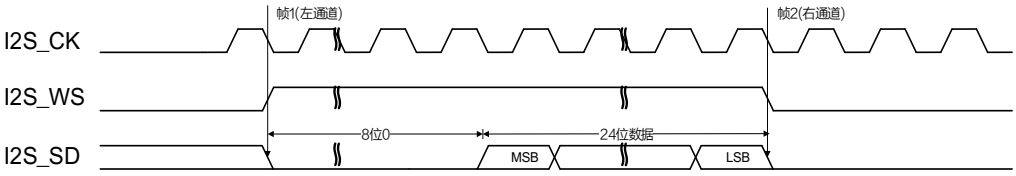
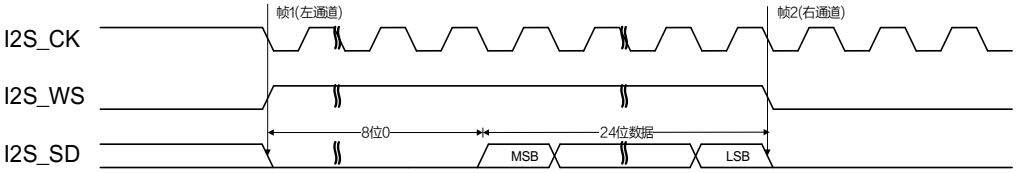


图 26-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当 24 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要访问 SPI_DATA 寄存器 2 次。在发送模式下，如果要发送一个 24 位数据 D[23:0]，第一个写入 SPI_DATA 寄存器的数据应该是一个 16 位数据，该 16 位数据的高 8 位可以是任意值，低 8 位是 D[23:16]，第二个数据应该是低 16 位数据 D[15:0]。在接收模式下，如果要接收一个 24 位数据 D[23:0]，第一个从 SPI_DATA 寄存器读到的数据应该是一个 16 位数据，该 16 位数据的高 8 位是 0，低 8 位是 D[23:16]，第二个数据应该是低 16 位数据 D[15:0]。

图 26-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

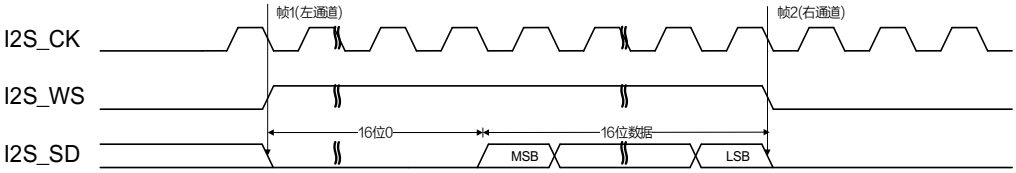
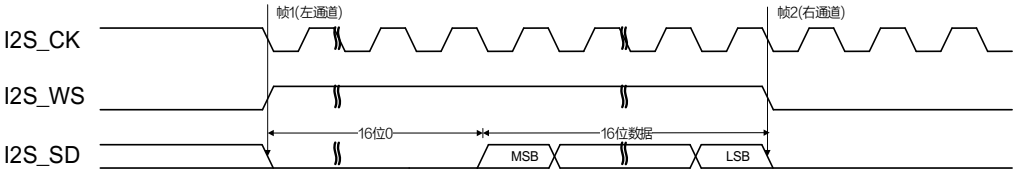


图 26-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



当 16 位数据打包成 32 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。为了将该 16 位数据扩展成 32 位数据，剩下的 16 位被硬件强制填充为 0x0000。

PCM 标准

对于 PCM 标准，I2S_WS 和 I2S_SD 在 I2S_CK 的上升沿变化，I2S_WS 信号表示帧同步信息。可以通过 SPI_I2SCTL 寄存器的 PCMSMOD 位来选择短帧同步模式和长帧同步模式。SPI_DATA 寄存器的处理方式与 I2S 飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 26-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

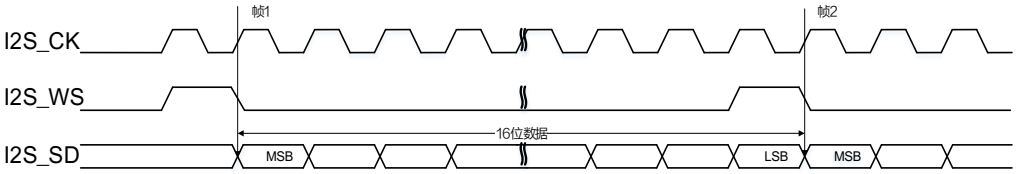


图 26-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

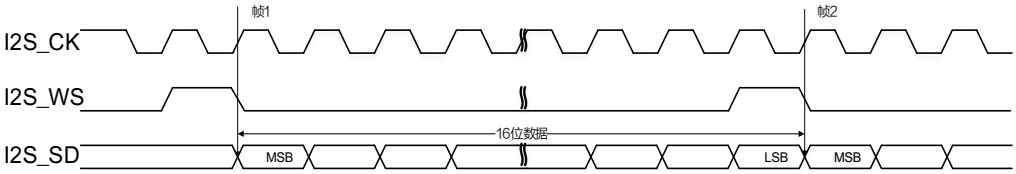


图 26-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

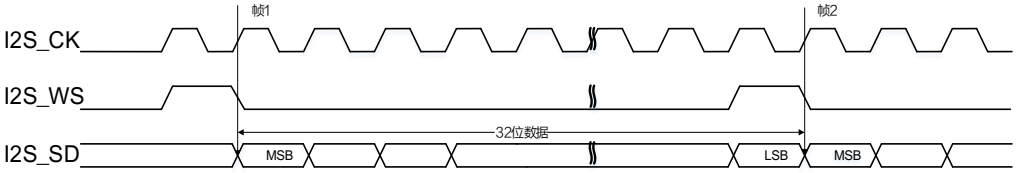


图 26-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

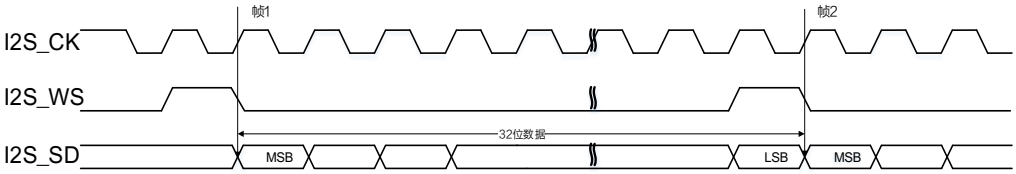


图 26-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

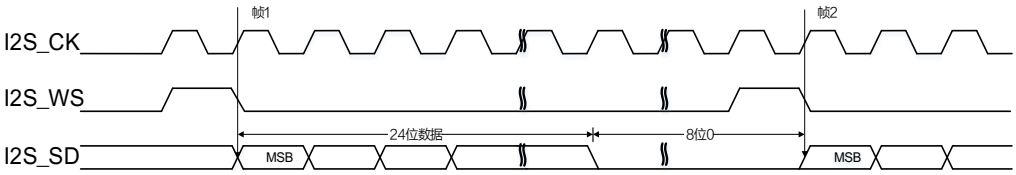


图 26-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

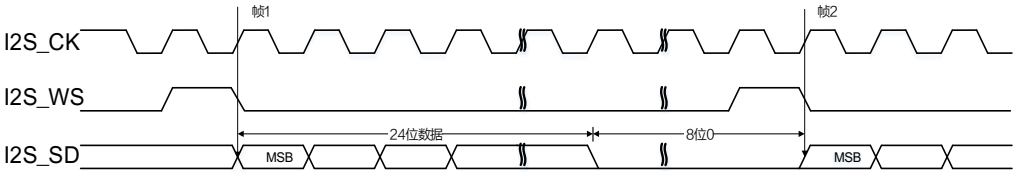


图 26-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

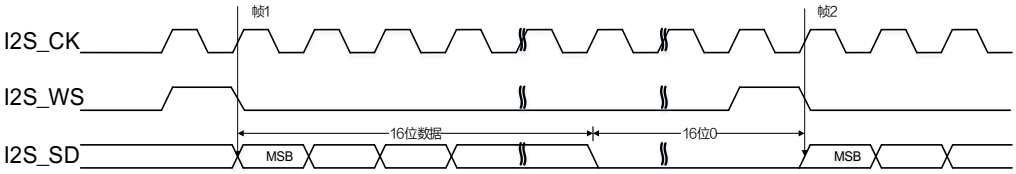
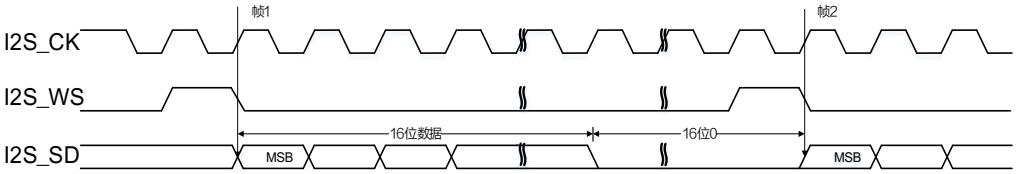


图 26-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 26-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

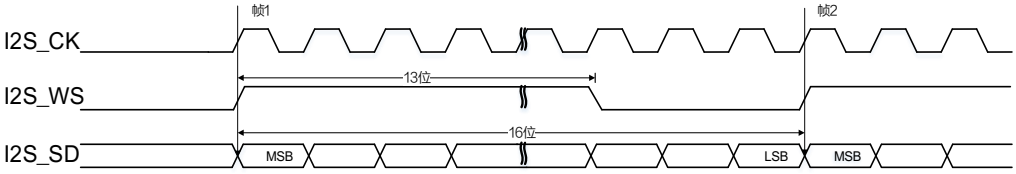


图 26-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

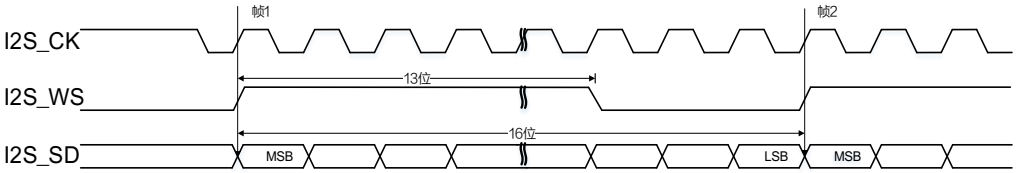


图 26-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

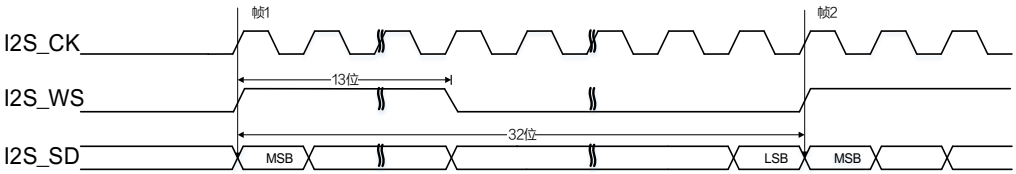


图 26-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

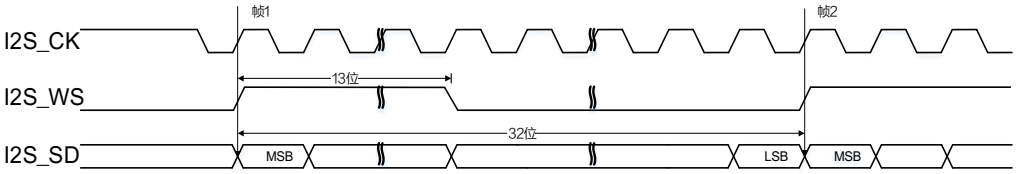


图 26-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

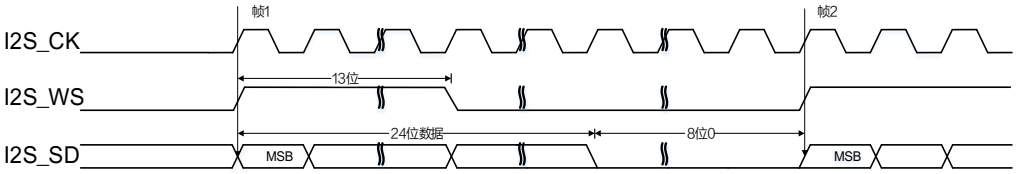


图 26-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

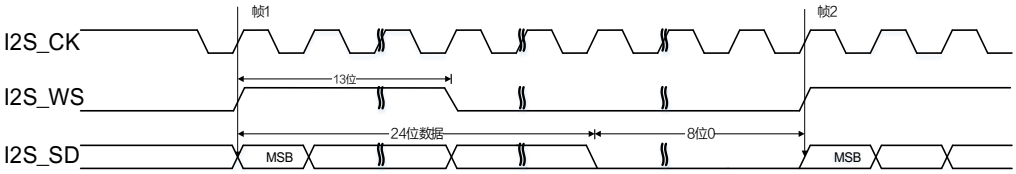


图 26-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

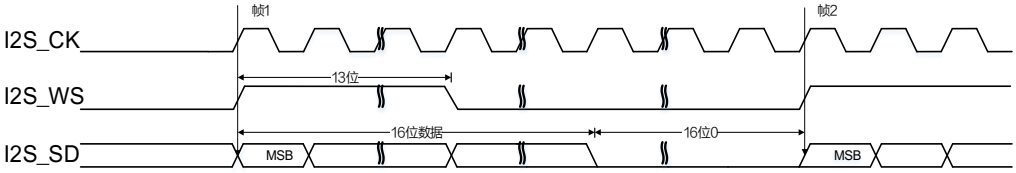
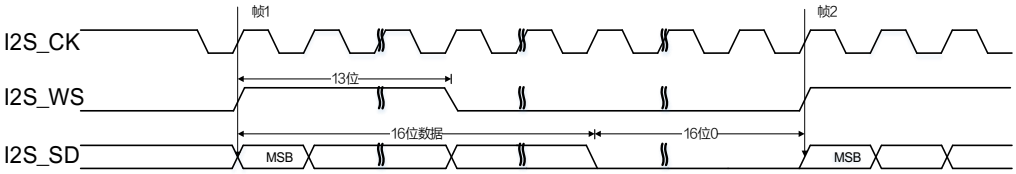
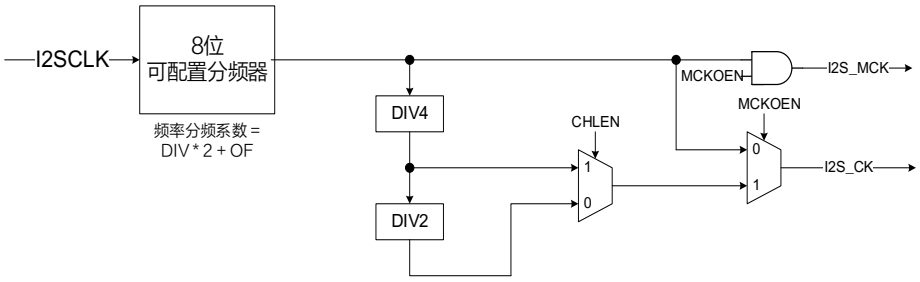


图 26-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



26.4.4. I2S 时钟

图 26-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图 26-51. I2S 时钟生成结构框图*所示。I2S 接口时钟是通过 SPI_I2SPSC 寄存器的 DIV 位，OF 位和 MCKOEN 位以及 SPI_I2SCTL 寄存器的 CHLEN 位来配置的。时钟源是系统时钟或 CK_PLL1(CK_SYS/ CK_PLL1)。比特率可以通过*表 26-7. I2S 比特率计算公式*所示的公式计算。

注意： I2S 串行时钟的配置值需设置为低于相应 PCLK 时钟的 1/6 倍以下(不包含 1/6)。

表 26-7. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

音频采样率(Fs)和 I2S 比特率的关系由如下公式定义：

Fs = I2S 比特率 / (通道长度 * 通道数)

所以，为了得到期望的音频采样率，时钟生成器需要按*表 26-8. 音频采样频率计算公式*所列的公式进行配置。

表 26-8. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

I2S 时钟的来源可以是 PLLI2S 或外部 I2S_CKIN 引脚，并且可以通过 RCU 进行编程配置。软件应仔细计算 I2S 和 PLLI2S 的因子，以获得最精确的音频采样频率。如果 PLLI2S 无法满足应用的精度需求，可以通过 I2S_CKIN 引脚引入外部精确的 I2S 时钟。

26.4.5. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下 I2S 接口信号的方向如*表 26-9. 各种运行模式下 I2S 接口信号的方向*所示。

表 26-9. 各种运行模式下 I2S 接口信号的方向

运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD	I2S_ADD_SD(2)
主机发送	输出或 NU(1)	输出	输出	输出	NU(1)
主机接收	输出或 NU(1)	输出	输出	输入	NU(1)
从机发送	输入或 NU(1)	输入	输入	输出	NU(1)
从机接收	输入或 NU(1)	输入	输入	输入	NU(1)
全双工	输出或 NU(1)	输出	输出	输出或输入	输出或输入

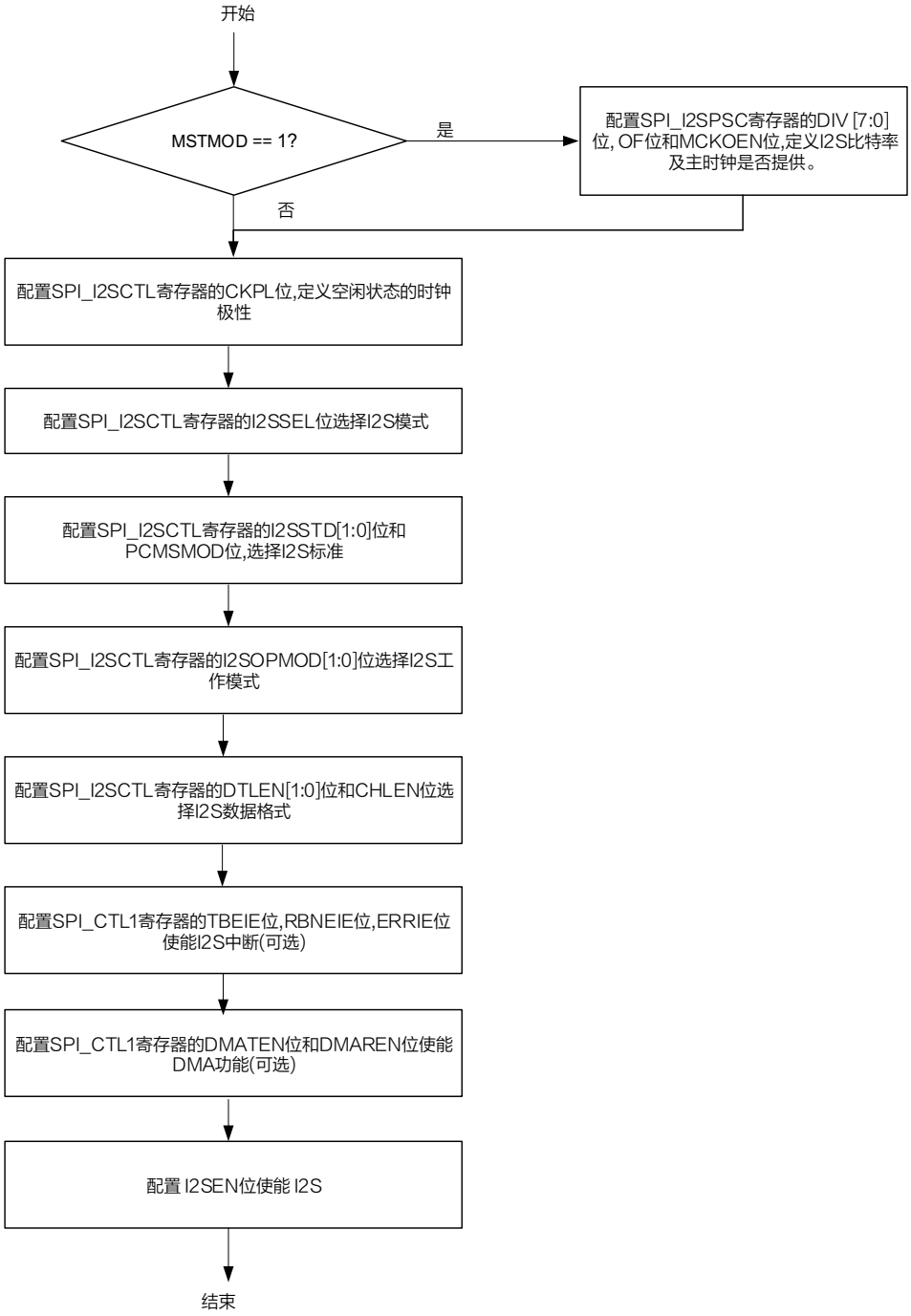
1. NU表示该引脚没有被I2S使用，可以用于其他功能。

2. I2S1 和 I2S2 为了支持全双工运行模式，需要两个额外的片上 I2S 模块：I2S_ADD1 和 I2S_ADD2。I2S_ADD_SD 引脚是 I2S_ADD 模块的数据引脚，在后面的章节将详细介绍全双工模式。

I2S 初始化流程

I2S 初始化过程如*图 26-52. I2S 初始化流程*所示。

图 26-52. I2S 初始化流程



I2S 主机发送流程

TBE 标志位被用来控制发送流程。如前文所述，TBE 标志位表示发送缓冲区空，此时，如果 SPI_CTL1 寄存器的 TBEIE 位为 1，将产生中断。首先，发送缓冲区为空(TBE 为 1)，且移位寄存器中没有发送序列。当 16 位数据被写入 SPI_DATA 寄存器时(TBE 变为 0)，数据立即从发送缓冲区装载到移位寄存器中(TBE 变为 1)。此时，发送序列开始。

数据是并行地装载到 16 位移位寄存器中的，然后串行地从 I2S_SD 引脚发出(高位先发)。下一个数据应该在 TBE 为 1 时写入 SPI_DATA 寄存器。数据写入 SPI_DATA 寄存器之后，TBE 变为 0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后 TBE 标志变回 1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入 SPI_DATA 寄存器。

对于除 PCM 标准外的所有标准，I2SCH 标志用来区别当前传输数据所属的通道。I2SCH 标志在每次 TBE 标志由 0 变 1 的时候更新。刚开始 I2SCH 标志为 0，表示左通道的数据应该被写入 SPI_DATA 寄存器。

为了关闭 I2S，I2SEN 位必须在 TBE 标志为 1 且 TRANS 标志为 0 之后清零。

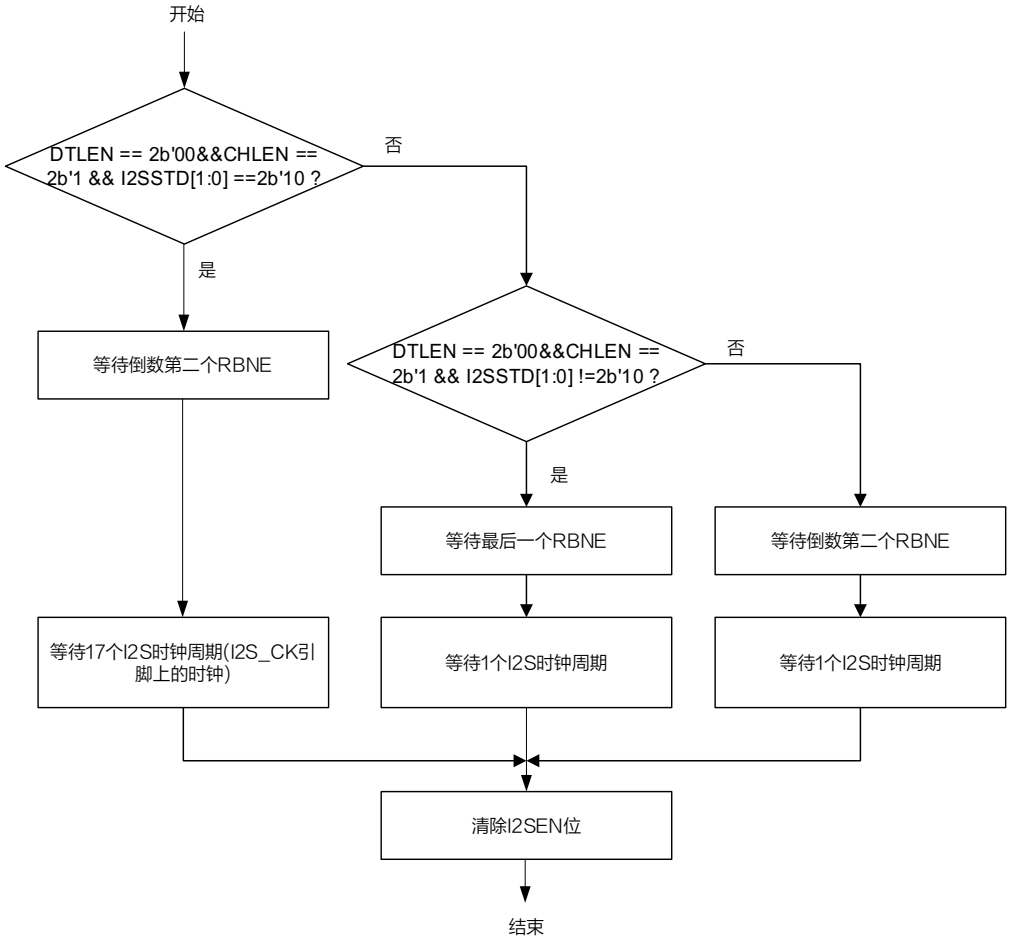
I2S 主机接收流程

RBNE 标志被用来控制接收序列。如前文所述，RBNE 标志表示接收缓冲区非空，如果 SPI_CTL1 寄存器的 RBNEIE 位为 1，将产生中断。当 SPI_I2SCTL 寄存器的 I2SEN 位被置 1 时，接收流程立即开始。首先，接收缓冲区为空(RBNE 为 0)。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区(RBNE 变为 1)。当 RBNE 为 1 时，用户应该将数据从 SPI_DATA 寄存器中读走。读操作完成后，RBNE 变为 0。必须在下一次接收结束之前读走 SPI_DATA 寄存器中的数据，否则将发生接收过载错误。此时 RXORERR 标志位会被置 1，如果 SPI_CTL1 寄存器的 ERRIE 位为 1，将会产生中断。这种情况下，必须先关闭 I2S 再打开 I2S，然后再恢复通讯。

对于除 PCM 之外的所有标准来说，I2SCH 标志用来区分当前传输数据所属的通道。I2SCH 标志在每次 RBNE 标志由 0 变 1 时更新。

为了关闭 I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作步骤如*图 26-53. I2S 主机接收禁能流程*所示。

图 26-53. I2S 主机接收禁能流程



I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且 I2S_WS 信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入 SPI_DATA 寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入 SPI_DATA 寄存器，否则会产生发送欠载错误。此时 TXURERR 标志会置 1，如果 SPI_CTL1 寄存器的 ERRIE 位为 1，将会产生中断。这种情况下，必须先关闭 I2S 再打开 I2S 来恢复通讯。从机模式下，I2SCH 标志是根据外部主机发送的 I2S_WS 信号而变化的。

为了关闭 I2S，必须在 TBE 标志变为 1 且 TRANS 标志变为 0 之后，才能清除 I2SEN 位。

I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且 I2S_WS 信号指示数据开始时，接收流程开始。从机模式下，I2SCH 标志是根据外部主机发送的 I2S_WS 信号而变化的。

为了关闭 I2S，必须在收到最后一个 RBNE 之后立即清除 I2SEN 位。

I2S 全双工模式

单个的 I2S 模块只支持单向传输：发送模式或接收模式，通过一个附加的 I2S 模块(I2S_ADD 模块)可以实现 I2S 的全双工模式。I2S_ADD 模块与 I2S 模块功能一样，但只工作在从模式。一共有两个 I2S_ADD 模块：I2S_ADD1 和 I2S_ADD2，所以只有 I2S1 和 I2S2 支持全双工模式。I2S_ADD 模块的 I2S_CK 和 I2S_WS 引脚分别与对应的 I2S 模块的相应引脚内部连接，I2S_ADD 模块的 I2S_SD 引脚映射到对应的 I2S 模块的 SPI_MISO 引脚。

为了工作在全双工模式，需要使能 I2S 模块和相应的 I2S_ADD 模块，I2S 模块支持两种全双工模式：主机模式和从机模式。

在主机全双工模式下，软件必须设置 I2S 为主机，I2S_ADD 为从机，I2S_ADD 模块的 WS 和 SCK 信号都由主机 I2S 模块提供。

在从机全双工模式下，软件必须设置 I2S 和 I2S_ADD 都为从机，I2S 模块和 I2S_ADD 模块的 WS 和 SCK 信号都由外部信号提供。

应用程序可以配置 I2S 模块为发送或接收模式，然后配置 I2S_ADD 为相反的模式。在发送过程中，软件同时操作 I2S 模块和 I2S_ADD 模块的寄存器和中断来实现全双工模式发送。

26.4.6. DMA 功能

DMA 功能与 SPI 模式完全一样，唯一不同的地方就是 I2S 模式不支持 CRC 功能。

26.4.7. I2S 中断

状态标志位

SPI_STAT 寄存器中有 4 个可用的标志位，分别是 TBE、RBNE、TRANS 和 I2SCH，用户通过这些标志位可以全面监视 I2S 总线的状态。

■ 发生缓冲区空标志(TBE)：

当发送缓冲区为空时，TBE 置位。软件可以通过写 SPI_DATA 寄存器将下一个数据写入发送缓冲区。

■ 接收缓冲区非空标志(RBNE)：

接收缓冲区非空时，RBNE 置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读 SPI_DATA 寄存器来读取此数据。

■ I2S通信进行中标志(TRANS):

TRANS 是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法通过软件进行操作。该标志位不会产生如何中断。

■ I2S通道标志(I2SCH):

I2SCH 用来表明当前传输数据的通道信息，对 PCM 音频标准来说没有意义。在发送模式下，I2SCH 标志在每次 TBE 由 0 变 1 时更新，在接收模式下，I2SCH 标志在每次 RBNE 由 0 变 1 时更新。该标志位不会产生任何中断。

错误标志

有三个错误标志:

■ 发送欠载错误标志(TXURERR):

在从发送模式下，当有效的 SCK 信号开始发送时，如果发送缓冲区为空时，发送欠载错误标志 TXURERR 将会置位。

■ 接收过载错误标志(RXORERR):

当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志 RXORERR 置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。

■ 帧错误(FERR):

在从 I2S 模式下，I2S 模块监视 I2S_WS 信号，如果 I2S_WS 信号在一个错误的位置发生翻转，将会置位 FERR 帧错误标志位。

*表 26-10. I2S 中断*总结了 I2S 中断事件和相应的使能位。

表 26-10. I2S 中断

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

26.5. SPI/I2S 寄存器

SPI0 基地址：0x4001 3000

SPI1/I2S1 基地址：0x4000 3800

SPI2/I2S2 基地址：0x4000 3C00

SPI3 基地址：0x4001 8400

I2S1_ADD 基地址：0x4000 3400

I2S2_ADD 基地址：0x4000 4000

26.5.1. 控制寄存器 0(SPI_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

该寄存器在 I2S 模式下没有意义。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式 0：2 线单向传输模式 1：1 线双向传输模式。数据在主机的 MOSI 引脚和从机的 MISO 引脚之间传输。
14	BDOEN	双向传输输出使能 当 BDEN 置位时，该位决定了数据的传输方向。 0：工作在只接收模式 1：工作在只发送模式
13	CRCEN	CRC 计算使能 0：CRC 计算禁止 1：CRC 计算使能
12	CRCNT	下一次传输 CRC 0：下一次传输值为数据

		1：下一次传输值为 CRC 值(TCR) 当数据传输由 DMA 管理时，CRC 值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入 SPI_DATA 寄存器后应将该位置 1。在只接收模式下，在接收完倒数第二个数据后应将该位置 1。
11	FF16	数据帧格式 0：8 位数据帧格式 1：16 位数据帧格式
10	RO	只接收模式 当 BDEN 清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS 软件模式选择 0：NSS 硬件模式，NSS 电平取决于 NSS 引脚 1：NSS 软件模式，NSS 电平取决于 SWNSS 位 该位在 SPI TI 模式下没有意义。
8	SWNSS	NSS 软件模式下 NSS 引脚选择 0：NSS 引脚拉低 1：NSS 引脚拉高 只有在 SWNSSEN 置位时，该位有效。 该位在 SPI TI 模式下没有意义。
7	LF	最低有效位先发模式 0：先发送最高有效位 1：先发送最低有效位 该位在 SPI TI 模式下没有意义。
6	SPIEN	SPI 使能 0：SPI 设备禁止 1：SPI 设备使能
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2　100：PCLK/32 001：PCLK/4　101：PCLK/64 010：PCLK/8　110：PCLK/128 011：PCLK/16　111：PCLK/256 当使用 SPI0 和 SPI3 时，PCLK=PCLK2，当使用 SPI1 和 SPI2 时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0：从机模式 1：主机模式
1	CKPL	时钟极性选择 0：SPI 为空闲状态时，CLK 引脚拉低

		1：SPI 为空闲状态时，CLK 引脚拉高
0	CKPH	时钟相位选择 0：在第一个时钟跳变沿采集第一个数据 1：在第二个时钟跳变沿时钟跳变沿采集第一个数据

26.5.2. 控制寄存器 1(SPI_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	TBEIE	发送缓冲区空中断使能 0：TBE 中断禁止 1：TBE 中断使能。当 TBE 置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：RBNE 中断禁止. 1：RBNE 中断使能。当 RBNE 置位时，产生中断。
5	ERRIE	错误中断使能 0：错误中断禁止 1：错误中断使能。当 CRCERR 位，CONFERR 位，RXORERR 位或者 TXURERR 位置 1 时，产生中断。
4	TMOD	SPI TI 模式使能 0：SPI TI 模式禁止 1：SPI TI 模式使能
3	NSSP	SPI NSS 脉冲模式使能 0：SPI NSS 脉冲模式禁止 1：SPI NSS 脉冲模式使能
2	NSSDRV	NSS 输出使能 0：NSS 输出禁止 1：NSS 输出使能。

		当 SPI 使能时，如果 NSS 引脚配置为输出模式，NSS 引脚在主模式时被拉低。如果 NSS 引脚配置为输入模式，NSS 引脚在主模式时被拉高，此时该位无效。
1	DMATEN	发送缓冲区 DMA 使能 0：发送缓冲区 DMA 禁止 1：发送缓冲区 DMA 使能。当 SPI_STAT 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。
0	DMAREN	接收缓冲区 DMA 使能 0：接收缓冲区 DMA 禁止 1：接收缓冲区 DMA 使能。当 SPI_STAT 中的 RBNE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。

26.5.3. 状态寄存器(SPI_STAT)

地址偏移：0x08
复位值：0x0002

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
							rc_w0	r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	FERR	帧错误 SPI TI 模式： 0：没有 TI 模式帧错误发生 1：TI 模式帧错误发生 I2S 模式： 0：没有 I2S 帧错误发生 1：I2S 帧错误发生 该位由硬件置位，可以通过写 0 清除。
7	TRANS	通信进行中标志 0：SPI 或 I2S 空闲 1：SPI 或 I2S 当前正在发送或接收数据 该位由硬件置位和清除。
6	RXORERR	接收过载错误标志

		0：没有接收过载错误发生 1：接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读 SPI_DATA 寄存器，然后读 SPI_STAT 寄存器。
5	CONFERR	SPI 配置错误 0：无配置错误发生 1：配置错误发生(主机模式下，在硬件 NSS 模式时 NSS 引脚被拉低，或者软件 NSS 模式时 SWNSS 位为 0，都会产生 CONFERR 错误) 该位由硬件置位，软件序列清零。软件序列为：先读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器。 I2S 模式下不使用该位。
4	CRCERR	SPI CRC 错误标志 0：SPI_RCRC 值等于最后接收到的 CRC 值 1：SPI_RCRC 值不等于最后接收到的 CRC 值 该位由硬件置位，可以通过写 0 清除。 I2S 模式下不使用该位。
3	TXURERR	发送欠载错误标志 0：无发送欠载错误发生 1：发送欠载错误发生 该位由硬件置位，通过读 SPI_STAT 寄存器清除。 SPI 模式下不使用该位。
2	I2SCH	I2S 通道标志 0：下一个将要发送或接收的数据属于左通道 1：下一个要发送或接收的数据属于右通道 该位由硬件置位和清除。 SPI 模式下该位无用，I2S PCM 模式下该位没有意义。
1	TBE	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RBNE	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

26.5.4. 数据寄存器(SPI_DATA)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

保留

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向 SPI_DATA 写数据将会把数据存入发送缓冲区，从 SPI_DATA 读数据，将从接收缓冲区获得数据。 当数据帧格式为 8 位时，SPI_DATA[15:8]强制为 0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是 8 位。如果数据帧格式为 16 位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是 16 位。

26.5.5. CRC 多项式寄存器(SPI_CRCPOLY)

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY [15:0]	CRC 多项式寄存器值 该值包含了 CRC 多项式，用于 CRC 计算，默认值为 0007h。

26.5.6. 接收 CRC 寄存器(SPI_RCRC)

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	接收 CRC 寄存器值 当 SPI_CTL0 中的 CRCEN 置位时，硬件计算接收数据的 CRC 值，并保存到 RCR 寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 RCRC[7:0]。如果是 16 位数据帧格式，CRC 计算基于 CRC16 标准进行，保存数据到 RCRC [15:0]。 硬件在接收到每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。 当 SPI_CTL0 寄存器中的 CRCEN 位和 SPIEN 位清零时，该寄存器复位。

26.5.7. 发送 CRC 寄存器(SPI_TCRC)

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	发送 CRC 寄存器值 当 SPI_CTL0 中的 CRCEN 置位时，硬件计算发送数据的 CRC 值，并保存到 TCR 寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 TCRC[7:0]。如果是 16 位数据帧格式，CRC 计算基于 CRC16 标准进行，保存数据到 TCRC[15:0]。 硬件在发送出每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。不同的数据帧格式(SPI_CTL0 中的 LF 位决定)将会得到不同的 CRC 值。 当 SPI_CTL0 寄存器中的 CRCEN 位和 SPIEN 位清零时，该寄存器复位。

26.5.8. I2S 控制寄存器(SPI_I2SCTL)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

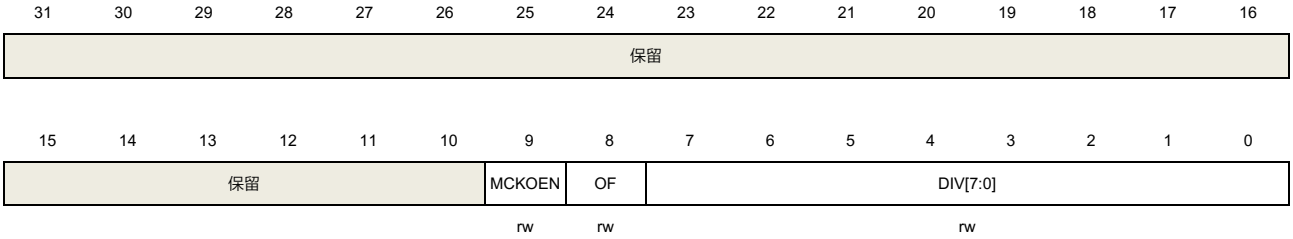
位/位域	名称	描述
31:12	保留	必须保持复位值。
11	I2SSEL	I2S 模式选择 0：SPI 模式 1：I2S 模式 当 SPI 模式或 I2S 模式关闭时配置该位。
10	I2SEN	I2S 使能 0：I2S 禁止 1：I2S 使能 SPI 模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S 运行模式 00：从机发送模式 01：从机接收模式 10：主机发送模式 11：主机接收模式 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
7	PCMSMOD	PCM 帧同步模式 0：短帧同步 1：长帧同步 只有在 PCM 标准下，该位才有意义。 当 I2S 模式关闭时配置该位。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 模式关闭时配置该位。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 模式关闭时配置该位。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 模式关闭时配置该位。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 模式关闭时配置该位。SPI 模式不使用该位。

26.5.9. I2S 时钟预分频寄存器(SPI_I2SPSC)

地址偏移：0x20
复位值：0x0000 0002

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	MCKOEN	I2S_MCK 输出使能 0：I2S_MCK 输出禁止 1：I2S_MCK 输出使能 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。
8	OF	预分频器的奇系数 0：实际分频系数为 DIV * 2

		1：实际分频系数为 DIV * 2 + 1 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是 DIV * 2 + OF。 DIV 不能为 0 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。

26.5.10. SPI0 四路 SPI 控制寄存器(SPI_QCTL)

地址偏移：0x80

复位值：0x0000 0000

该寄存器可以按字节(8 位)或半字(16 位)或字(32 位)访问。

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QRD	四路 SPI 模式读选择 0：SPI 四路模式写操作 1：SPI 四路模式读操作 该位仅能在 SPI 未通信时配置(TRANS 位清零)。 该位仅适用于 SPI0 和 SPI3。
0	QMOD	四路 SPI 模式使能 0：SPI 工作在单路模式 1：SPI 工作在四路模式 该位仅能在 SPI 未通信时配置(TRANS 位清零)。 该位仅适用于 SPI0 和 SPI3。

27. 八线 SPI 接口(OSPI)

27.1. 简介

OSPI 是一种专用于和外部存储器通信的接口，如串行 PSRAMS，串行 NAND，串行 NOR Flash，HypterRAM™ 和 HypterFlash™ 存储器，支持单线，双线，四线，双片选和八线 SPI 模式。

27.2. 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(OSPI：0xB000 0000 - 0xBFFF 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写；
- 支持单线、双线、四线、双片选和八线通信；
- 支持DQS
- 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(双倍数据速率)
- 支持两种帧格式
  - 常规帧格式
  - HyperBus™帧格式
- 常规帧格式和HyperBus™帧格式都支持回卷
- 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、超时、传输完成、访问错误中断。

27.3. OSPI 功能描述

27.3.1. OSPI 结构框图

OSPI 引脚描述在*表 27-1. OSPI 信号线描述*中。

表 27-1. OSPI 信号线描述

引脚	方向	描述
CSN0/1	O	片选输出(低电平有效)
SCK	O	时钟输出
nSCK	O	反向时钟信号(仅在 1.8V HyperBus 协议下使用)

引脚	方向	描述
DQS	I/O	数据选通信号
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。 双片四线模式：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。 双片四线模式：数据输入或输出。 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。 双片四线模式：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。 双片四线模式：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"。 双片四线模式：数据输入或输出。 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"。 双片四线模式：数据输入或输出。 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"。 双片四线模式：数据输入或输出。 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"。

引脚	方向	描述
		四线模式：输出模式并强制置"0"。 双片四线模式：数据输入或输出。 八线模式：数据输入或输出。

图 27-1. OSPI 八线通信模式结构框图

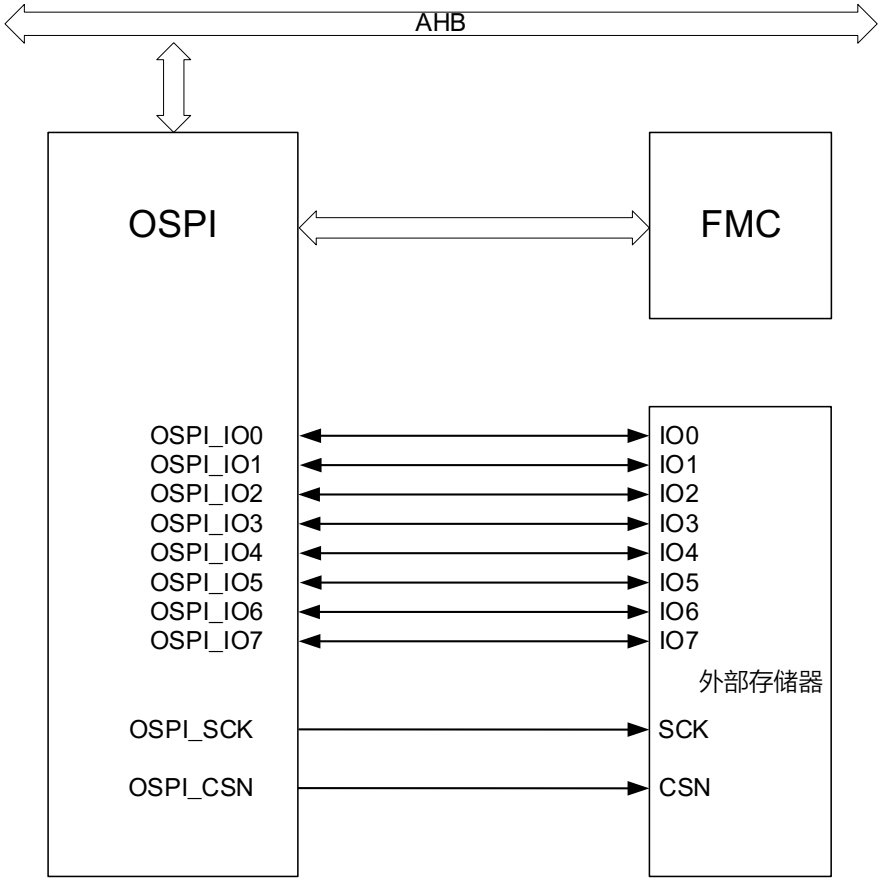


图 27-2. OSPI 四线通信模式结构框图

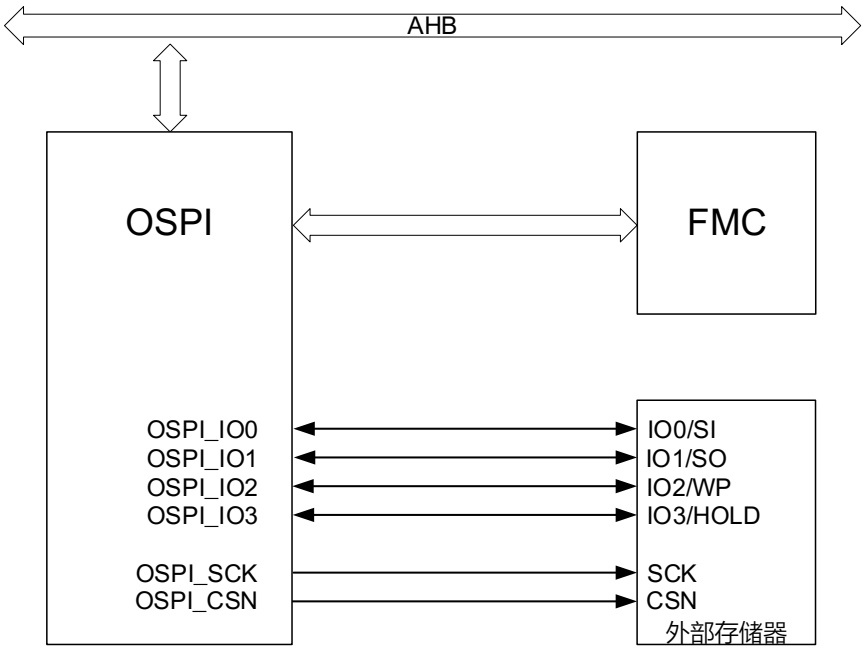
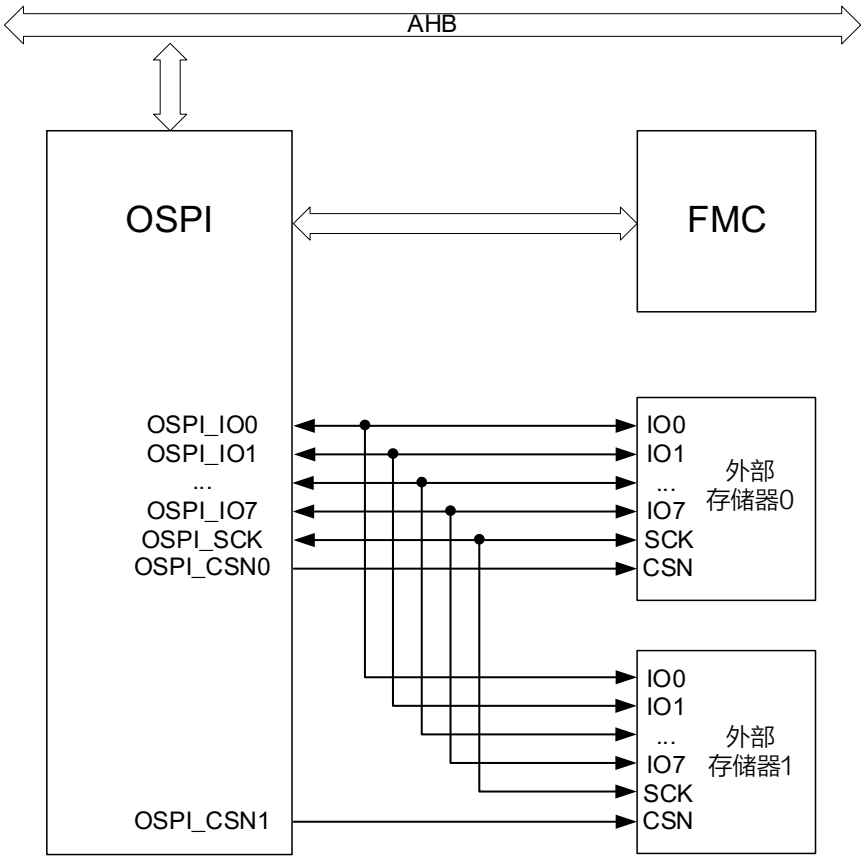


图 27-3. OSPI 双片选通信模式结构框图



27.3.2. 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 27-4. 八线模式 OSPI 命令格式

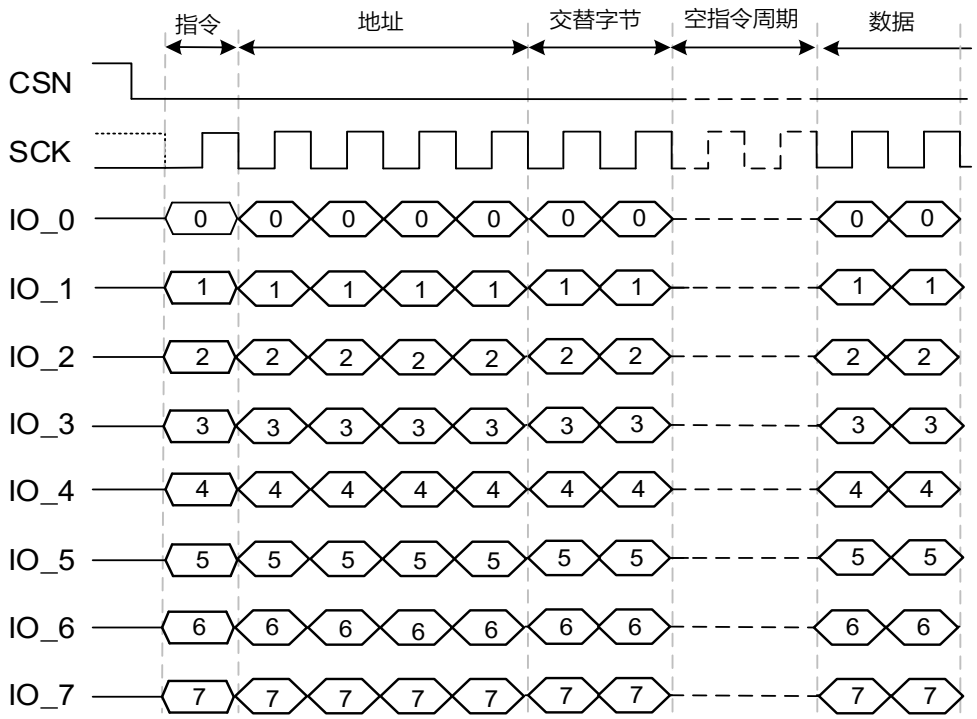
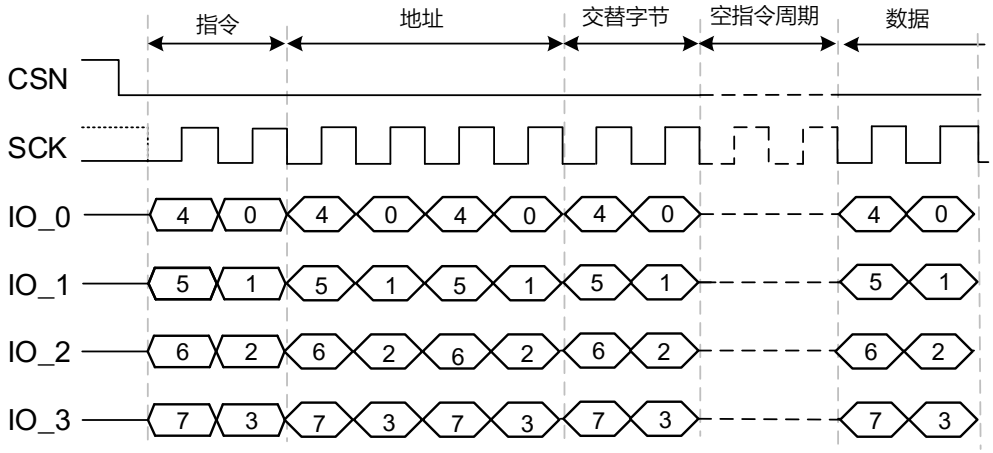


图 27-5. 四线模式 OSPI 命令格式



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)。将 OSPI_TCFG 寄存器的 INSDTR 位置 1 可使能 DTR 模式，则指令在时钟的每个边沿被发送。

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，模式以及是否开启 DTR)。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。在间接模式和状态轮询模式下，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 AHB 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指令大小(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 ABDTR 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。
2. 使用激活了DQS的外部存储器时，至少设置5个空指令周期。

数据阶段

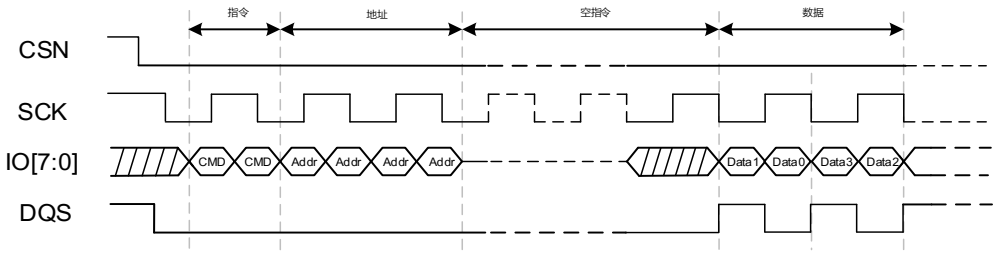
数据阶段，任意数量的字节可以在外部存储器和 OSPI 之间传输。OSPI_TCFG 寄存器的DATAMOD[2:0]位域定义了数据阶段的模式(无数据，1 线，2 线，4 线或者 8 线，其中无数据仅可用于间接写入模式)。将 OSPI_TCFG 寄存器的 DADTR 位置 1 可使能 DTR 模式，则数据在时钟的每个边沿被发送或接收。在间接模式下，OSPI_DTLEN 寄存器定义了待接收或者待发送数据的字节数。在写操作时，需要发送的数据写入 OSPI_DATA 寄存器，在读操作时，接收数据从 OSPI_DATA 寄存器中获取。在状态轮询模式下，OSPI_DTLEN 寄存器定义了待接收的字节数，接收数据从 OSPI_DATA 寄存器中获取。

在内存映射模式下，数据的传输通过 AHB 直接发送或接收。传输的字节个数确定 AHB 总线的访问操作，可以 8 位，16 位或者 32 位读写访问，相应传输 1 个，2 个，4 个字节。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。

DQS 信号

DQS 信号通常用于高速应用，以指示控制器何时准备好读取数据。当启用 DQS 功能时，DQS 信号的切换频率与 SCK 相同。对于 STR 读操作，数据只在 DQS 信号的上升沿被锁存。对于 DTR 读操作，数据在 DQS 信号的上升沿和下降沿被锁存。

图 27-6. 八线模式下使用 DQS 的 DTR 读时序



双片选模式

当双片选模式启用(DCSNEN = 1)时，请参考*图 27-3. OSPI 双片选通信模式结构框图*，存储器映射模式要求映射大小(MESZ)配置为最大值 256M。存储空间随后被分为两部分，每部分占用 128M。具体如下:

- CSN0对应地址范围B000_0000 ~ B7FF_FFFF
- CSN1对应地址范围B800_0000 ~ BFFF_FFFF

当总线访问地址范围 B000_0000～B7FF_FFFF 时，CSN0 被拉低；否则，CSN1 被拉低。此设计允许同时连接两个存储器设备，共享同一组 IO，通过片选信号(CSN)区分它们。

在间接模式下，启用 CSN1EN 允许选择固定的 CSN 输出。通过寄存器配置，可以独立设置两个存储器设备的参数，从而实现灵活高效的操作。

注意：

1. 使用 CSN1 配置为存储器映射模式工作时，为保证访问正常，需将 CSN0 关联的 TCFG0 寄存器中的 ADDRMOD 位域配置为非 0。

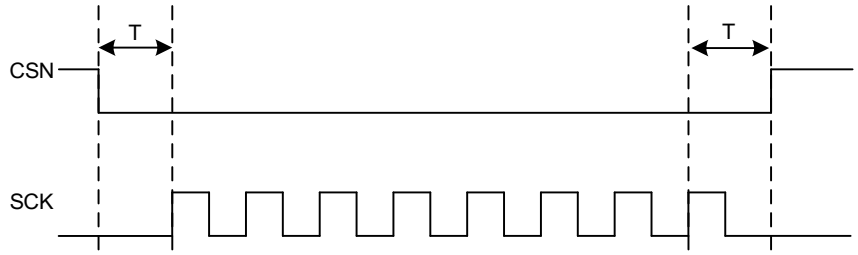
2. PE7/PE8/PE9/PE10/PE12/PE13/PE14/PE15，PD9/PD10 无法用作 CSN1 对应的存储器映射模式引脚。

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 为高电平时，SCK 必须保持低电平。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 27-7. CSN 和 SCK 的行为*所示：

图 27-7. CSN 和 SCK 的行为



27.3.3. HyperBusTM 模式

通过配置 OSPI_DCFG0 寄存器中的 DTYSEL[2:0]位域可使 OSPI 工作在 HyperBus™ 模式下，在此模式下 OSPI 可以通过 HyperBus™ 协议与外部存储器进行通信。HyperBus™ 帧由指令/地址阶段和数据阶段两个阶段组成。HyperBus™ 模式下使用的引脚定义如*表 27-2. HyperBusTM 模式下信号线描述*所示：

表 27-2. HyperBus™ 模式下信号线描述

引脚	方向	描述
CSN	O	片选输出(低电平有效)
SCK，nSCK	O	差分时钟信号，nSCK 仅在 1.8V HyperBus™ 模式下使用。
RWDS	I	读写数据选通和延迟插入
IO[7:0]	I/O	数据输入/输出

指令/地址阶段

在指令/地址阶段，OSPI 通过 IO[7:0]发送 48 位数据来指定 OSPI 与设备之间的操作，例如地址空间、突发类型、读取或写入，参考*表 27-3. HyperBusTM 模式下指令/地址阶段说明*:

表 27-3. HyperBus™ 模式下指令/地址阶段说明

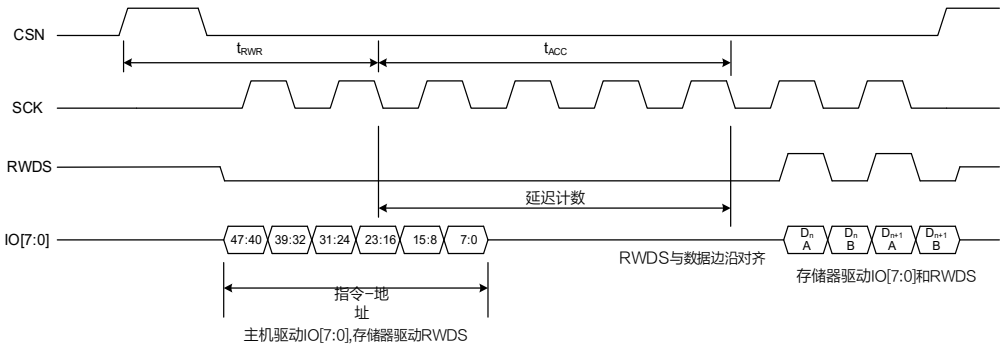
CA 位	位名	描述
47	R/W#	表示访问为数据读操作或写操作，R/W# = 1 表示读操作，R/W# = 0 表示写操作。
46	目标空间	表示访问的目标空间是存储器空间或寄存器空间，数值 0 表示对存储器空间进行操作，数值 1 表示对寄存器空间进行操作。
45	突发类型	表示突发序列是线性还是回卷，数值 0 表示回卷突发，数值 1 表示线性突发。
44:16	行与高位列地址	表示行地址和高位列地址。
15:3	保留	-
2:0	低位列地址	表示目标地址的低位列地址，即半页内的起始 16 位字。

目标空间的类型通过 OSPI_DCFG0 寄存器中的 DTYSEL[2:0]位域进行配置，器件的总大小在 OSPI_DCFG0 寄存器中的 MESZ[4:0]位域进行配置。

读取操作

当 OSPI 对 HyperBus™ 设备进行读操作时，首先是由 OSPI 向设备发送 48 位的 CA 信息，然后再进行一定周期的延迟计数后开始读取数据。当数据开始传输，主机读取数据在 RWDS 信号的边沿进行采样。如*图 27-8. 具有初始延迟的 HyperBusTM 读取操作*所示:

图 27-8. 具有初始延迟的 HyperBus™ 读取操作



其中，t_{RWR} 为器件最短读写恢复时间，由 OSPI_HBLCFG 寄存器的 RWRTM[7:0]位域定义，t_{ACC} 为器件的初始访问时间，由 OSPI_HBLCFG 寄存器的 ACCTM[7:0]位域定义。

当外部的 HyperBus™ 设备需要一段额外的延迟，则 OSPI 会增加一段额外的 t_{ACC} 时间，如*图 27-9. 具有额外延迟的 HyperBusTM 读取操作*所示。而对于越过页的访问，在越过页时需要添加延迟，如*图 27-10. 超出页界限的读取操作*所示。

图 27-9. 具有额外延迟的 HyperBus™ 读取操作

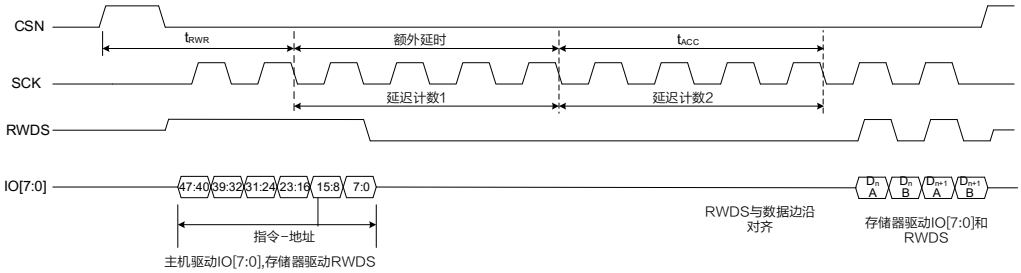
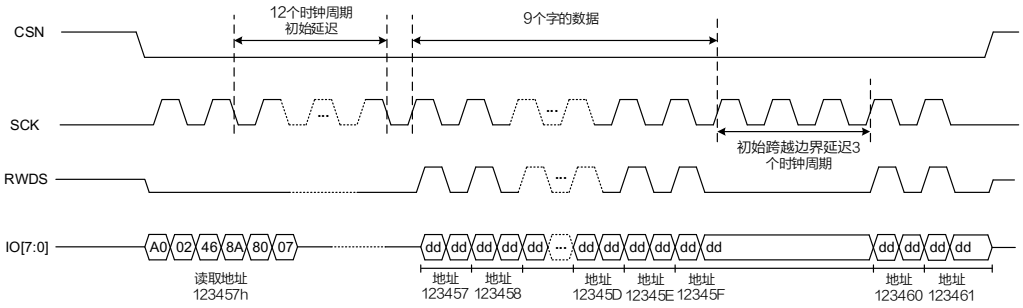


图 27-10. 超出页界限的读取操作



写入操作

当主机对 HyperBus™ 设备进行写入操作，RWDS 在指令地址阶段请求额外延迟和在数据阶段进行写入数据屏蔽，具有初始延迟的写入操作如*图 27-11. 具有初始延迟的 HyperBusTM 写入操作*所示。

当外部的 HyperBus™ 设备需要一段额外的延迟，则 OSPI 会增加一段额外的 t_{ACC} 时间，如*图 27-12. 具有额外延迟的 HyperBusTM 写入操作*所示。对于一些要求写入操作延迟为零的器件，通过配置 OSPI_HBLCFG 寄存器的 WZLAT 位，可以将延迟时间设置为零，如*图 27-13. 无延迟的 HyperBusTM 写入操作*所示。

对于一些要求写入操作有固定延迟时间的器件，通过配置 OSPI_HBLCFG 寄存器的 LMOD 位，可以将延迟时间设置为固定住 $2xt_{ACC}$。在固定延迟时间模式下，OSPI 不考虑 RWDS 信号。

注： 当 Hyper 存在额外延迟时，为了避免走线延迟对数据接收的干扰，需配置 OSPI_TIMCFG 的 SSAMPLE。

图 27-11. 具有初始延迟的 HyperBus™ 写入操作

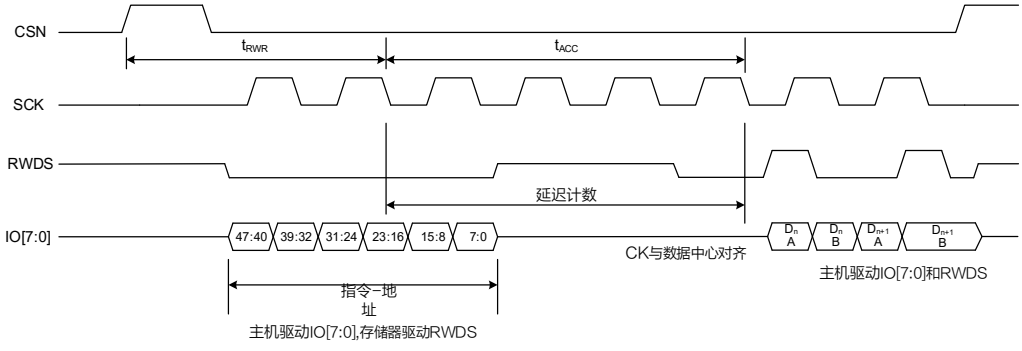


图 27-12. 具有额外延迟的 HyperBus™ 写入操作

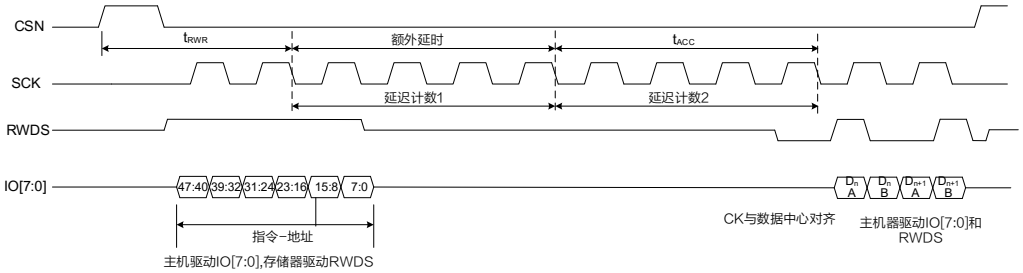
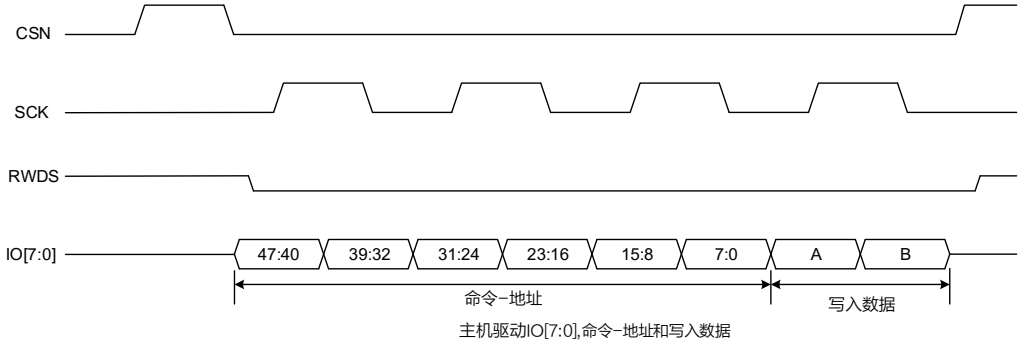


图 27-13. 无延迟的 HyperBus™ 写入操作



27.3.4. 回卷支持

仅在开启 cache 时，读模式下支持回卷，并且当数据大小为 AHB 的 4 字节时适用。要启用 cache，必须根据回卷读访问特性编程 OSPI_WPTCFGx、OSPI_WPTIMCFGx、OSPI_WPINSx 和 OSPI_WPALTEx。

27.3.5. 片选边界

片选(CS)边界功能将事务限制在特定边界范围内对齐的地址。该边界大小由 OSPI_DCFG2 寄存器中的 CS 边界字段 CSBND[4:0]定义，其中字段值表示 2^{CSBND}。当最低有效地址达到 0xF 时，nCS 信号将停用并过渡到下一个数据地址以进行后续事务。如果 CSBND[4:0] = 0，则该功能被禁用，建议将最小值设置为 3。

注意： 片选边界功能不适用于 Flash 存储器的写模式。写操作需要额外的命令来完成下一页的编程。

27.3.6. 刷新支持

标准帧格式和 HyperBus™ 帧格式都支持刷新功能，可以将事务的持续时间限制为 OSPI_DCFG3 寄存器的 REFRATE[31:0]字段中定义的值。持续时间以周期数表示，允许外部 RAM 设备定期执行内部刷新操作。刷新值必须大于以周期数计算的，包括指令/地址/交替/空指令阶段在内的最小事务大小。

27.4. 操作模式

27.4.1. 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。

HyperBus™ 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。

在间接读取模式下，在访问 OSPI_ADDR 寄存器之后立即开始命令序列。

在间接写入模式下，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。

设置的阈值大于 20 个字节时，必须读够足够的 16 个字节，否则 OSPI 不会继续从外部设备取数，FT 标志也不会置位。

27.4.2. 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，会产生一个中断，SMIE 置位。

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。否则，周期性读序列会一直持续直到发生中止或者 OSPI 被关闭。

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。

对于 HyperBus，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。

27.4.3. 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，AHB 会产生错误。错误的影响取决于 AHB 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，读操作支持字节、半字、字和突发访问，写操作支持字、双字和突发访问。

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。如果超时计数器被使能，当保持状态时间到达 TMOUT 位域定义的 SCK 时钟周期数，CSN 会被拉高。

在开始传输时，在 CSN 下降之前 BUSY 位会变为高电平，在发生超时、终止或者禁用 OSPI

之后变为低电平。

27.5. OSPI 配置

27.5.1. 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状态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. 若OSPI工作在内存映射模式，配置TMOUTEN位选择是否使能超时计数器，设置OSPI_LPTOUT寄存器的TIMEOUT[15:0]位域配置超时值。
6. 如需使用中断，则可设置相关中断的使能位。

27.5.2. 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。

27.5.3. 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. 设置OSPI_DTLEN寄存器，配置数据长度。
2. 设置OSPI_TIMCFG寄存器，配置帧时序。
3. 设置OSPI_TCFG寄存器，配置帧格式。
4. 设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
5. 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. 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 OSPI 工作在状态标识轮询模式时，具体配置如下：

1. 设置OSPI_STATMK寄存器，指定对接收的状态字节进行屏蔽。
2. 设置OSPI_STATMATCH寄存器，指定与OSPI_STATMK寄存器比较的值。
3. 设置OSPI_INTERVAL寄存器，指定读取操作之间的时钟周期数。
4. 设置OSPI_DTLEN寄存器，配置数据长度。
5. 设置OSPI_TIMCFG寄存器，配置帧时序。
6. 设置OSPI_TCFG寄存器，配置帧格式。
7. 设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
8. 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. 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. 设置OSPI_TIMCFG寄存器，配置读取操作帧时序。
2. 设置OSPI_TCFG寄存器，配置读取操作帧格式。
3. 设置OSPI_INS寄存器，指定要发送到外部存储器的指令。
4. 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. 设置OSPI_WTIMCFG寄存器，配置写操作帧时序。
6. 设置OSPI_WTCFG寄存器，配置写操作帧格式。
7. 设置OSPI_WINS寄存器，指定要发送到外部存储器的指令。
8. 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。

27.5.4. OSPI HyperBusTM 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. 设置OSPI_DTLEN寄存器，配置数据长度。
2. 设置OSPI_ADDR寄存器，指定要发送到外部存储器的地址。
3. 通过OSPI_DATA寄存器读取/写入数据。

状态标志轮询模式

当 OSPI 工作在状态标志轮询模式时，具体配置如下：

1. 设置OSPI_STATMK寄存器，指定对接收的状态字节进行屏蔽。

2. 设置OSPI_STATMATCH寄存器，指定与OSPI_STATMK寄存器比较的值。
3. 设置OSPI_INTERVAL寄存器，指定读取操作之间的时钟周期数。
4. 设置OSPI_DTLEN寄存器，配置数据长度。
5. 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。

27.6. OSPI 数据采样移位

OSPI 会在复位后由外部存储器驱动数据 1/2 个时钟之后对数据进行采样。当用内部时钟采样而非外部设备的 DQS 时，通过在 OSPI_TIMCFG 寄存器中配置 SSAMPLE 位，可以将数据采样移位 1/2 个时钟周期，或者配合 DCFG0 的 RXSFT[2:0]，可以最多移位 7.5 个周期。当 DQSEN = 1 时，可以对 DQS 信号进行偏移以弥补数据传播延迟。该功能可以在内部执行和控制。通过软件设置 DCFG0 的 RSCKDEN 位可以绕过延迟模块。

27.7. OSPI 只发送一次指令

只发送一次指令由 SIOO 位设置，该功能对所有模式有效。如果 SIOO 置位，在 OSPI_TCFG 访问后该指令只发送一次，后续命令序列会跳过指令阶段，直到 OSPI_TCFG 再次被访问。在 IMOD=00 时，该功能无效。

27.8. 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，BUSY 位清 0，如果在间接模式读操作时，需要 FIFO 为空时，BUSY 位才会清 0。在状态标志轮询模式下，只有在发生匹配或中止，BUSY 位才会清 0。在内存映射模式下进行第一次访问后，只有在发生中止，BUSY 位才会清 0。一旦中止操作完成，BUSY 和 ABORT 都将被重置，同时 FIFO 将被清空。在将 ABORT 设置为 1 之前，软

件必须确保所有当前事务都通过同步屏障完成。如果对状态寄存器的写操作被中止，某些设备可能会表现出异常行为。

27.9. 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，AHB超出范围的访问会返回AHB总线错误。
- 当AHB主机访问内存映射空间时，内存映射模式没有被使能，会返回AHB总线错误。

OSPI 在下面条件下会产生 AHB 总线错误：

- 内存映射模式不使能时，AHB读请求响应。
- 读或者写超过了外部 memory 的大小。
- 在读或写的过程中，中止发生。
- 在读或写的过程中，OSPI 被禁止。
- 写回卷突发。

27.10. OSPI 中断

表 27-4. 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
TMOUT	超时	设置 OSPI_STATC 寄存器 TMOUTC 位为 1	TMOUTIE
SM	状态匹配	设置 OSPI_STATC 寄存器 SMC 位为 1	SMIE

27.11. OSPI 寄存器

OSPI 基地址：0x4002 5000

27.11.1. 控制寄存器(OSPI_CTL)

偏移地址：0x00
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改(位 0 和位 1 除外)。

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
DCSNEN	CSN1EN	FMOD[1:0]		RCKDV[3:0]				SPMOD	SPS	保留	TMOUTIE	SMIE	FTIE	TCIE	TERRIE
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				FTL[4:0]				DEVS	保留	DQSOEN	保留	TMOUTEN	DMAEN	ABORT	OSPIEN
				rw				rw		rw		rw	rw	rw	rw

位/位域	名称	描述
31	DCSNEN	启用双片选模式 0: 禁用双片选模式 1: 启用双片选模式
30	CSN1EN	选择固定特定的CSN输出(当不处于内存映射模式时) 0: 固定CSN0输出 1: 固定CSN1输出
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。如果在DMAEN = 1的情况下错误更新FMODE[1:0]和FTHRES[4:0]，DMA请求信号会自动变为非活动状态。
27:24	RCKDV[3:0]	接收 SCK 延迟值 0000：延迟 1 个延迟单元 0001：延迟 2 个延迟单元 …… 1111：延迟16个延迟单元
23	SPMOD	状态轮询匹配模式

		该位指示在状态轮询模式下必须使用哪种方法来确定匹配。 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：在中止或者禁能OSPI模块时状态轮询停止。 1：在产生匹配后状态轮询停止。
21	保留	必须保持复位值。
20	TMOUTIE	超时中断使能 0：禁用超时中断。 1：使能超时中断。
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阈值标志被置位。 间接模式写操作时(FMOD = 00)： 0：FT会被置位，如果有1个或者更多字节可以有效写入FIFO。 1：FT会被置位，如果有2个或者更多字节可以有效写入FIFO。 ... 31：FT会被置位，如果有32个字节可以有效写入FIFO。 间接模式读操作时(FMOD = 01)： 0：FT会被置位，如果有1个或者更多有效数据能从FIFO中读取。 1：FT会被置位，如果有2个或者更多有效数据能从FIFO中读取。 ... 31：FT会被置位，如果有32个有效数据能从FIFO中读取。

		如果DMAEN为1，在改变FTL之前，DMA控制器的相应通道必须是禁用的。
7	DEVS	设备选择 该位选择在单片存储器模式下单线/双线/四线模式中要寻址的设备。 0：选择设备0，通过IO[3:0]传输数据。 1：选择设备1，通过IO[7:4]传输数据。 当OSPI工作在八线模式下，忽略该位。
6	保留	必须保持复位值。
5	DQSOEN	DQS输出使能 将该位设置为1将在写操作期间将DQS信号输出为SCK。 0：失能DQS输出 1：使能DQS输出
4	保留	必须保持复位值。
3	TMOUTEN	超时计数器使能 仅当在内存映射模式下有效。如果没有访问外部存储器，激活该位会在TMOUT[15:0]定义的时间之后释放CSN。 0：超时计数器失能，在内存映射模式下访问后片选保持激活。 1：超时计数器使能，在内存映射模式下，外部存储器不活动时间超过TMOUT[15:0]个周期后释放片选。
2	DMAEN	DMA使能 间接模式下，可以使用DMA通过OSP_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。 **注意：**在DMA传输进行时重置DMAEN位会破坏与DMA的握手。在DMA操作期间请勿写入该位。
1	ABORT	中止请求 该位停止当前命令，中止请求完成后会自动清除。 状态轮询模式或者内存映射模式，该位会复位SPS位或者DMAEN位。 0：无终止请求。 1：中止请求。 **注意：**该位始终读为0。
0	OSPIEN	使能OSPI 0：禁用OSPI。 1：使能OSPI。 **注意：**如果在操作期间清除了OSPIEN位，则可以在未收到ACK的情况下中止DMA请求。 如果在DMA传输期间将该位设置为0，则发送给DMA的REQ信号会返回到非活动状态，而无需等待DMA的ACK信号变为活动状态。

27.11.2. 设备配置寄存器 0(OSPI_DCFG0)

偏移地址：0x08
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31	RCKADD	在接收过程中发送一个额外的时钟，但接收到的数据保持不变。为了防止DQS错过一个边缘，选择DQS进行接收时需要进行配置。 0：无额外时钟 1：一个额外时钟
30:27	保留	必须保持复位值。
29:27	DTYSEL1[2:0]	当启用双CSN模式时，与CSN1对应的设备类型 位域功能与DTYSEL0[2:0]相同
26:24	DTYSEL0[2:0]	选择设备类型 **注意：**这里是默认值，对于非美光的存储器，需谨慎修改DTYSEL0[2:0]。 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：Micronix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。 100：HyperBus存储器模式，协议遵循HyperBus规范。必须选择8位数据模式。 101：HyperBus寄存器模式，寻址寄存器空间。该模式下的内存映射访问必须是不可缓存的，或者必须使用间接读取/写入模式。 其它值：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 = $2^{[MESZ[4:0]+1]}$。

		MESZ[4:0]+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	保留	必须保持复位值。
6:4	RXSFT[2:0]	数据延迟超过0.5个周期时，接收移位步长 配置该位域时，需要保证配置CSHC的时间大于该位域至少3个周期 000：不延迟 001：1个周期 010：2个周期 011：3个周期 100：4个周期 101：5个周期 110：6个周期 111：7个周期
3	RSCKDEN	采样延迟使能 0：失能采样延时 1：使能采样延时
2	保留	必须保持复位值。
1	FRCK	自由运行时钟使能 0：禁用SCK自由运行。 1：使能SCK自由运行。
0	保留	必须保持复位值。

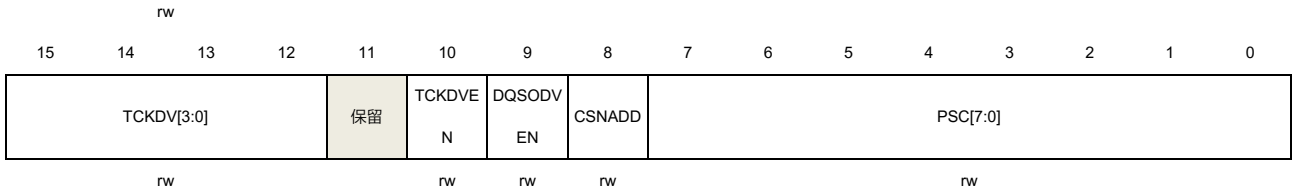
27.11.3. 设备配置寄存器 1(OSPI_DCFG1)

偏移地址：0x0C
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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



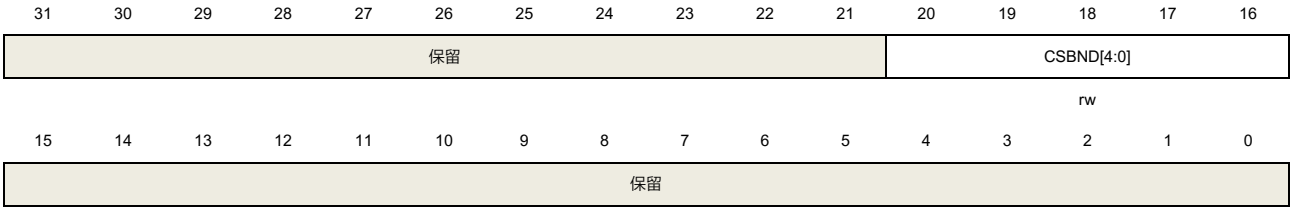
位/位域	名称	描述
31:28	DQSODV[3:0]	DQS 输出传输 SCK 延迟值 0000：延迟 1 个时钟单元 0001：延迟 2 个时钟单元 …… 1111：延迟16个时钟单元
27:16	保留	必须保持复位值。
15:12	TCKDV[3:0]	SCK 传输延迟值 0000：延迟 1 个时钟单元 0001：延迟 2 个时钟单元 …… 1111：延迟16个时钟单元
11	保留	必须保持复位值。
10	TCKDVEN	SCK传输延迟使能 0：失能 SCK 传输延迟 1：使能SCK传输延迟
9	DQSODVEN	DQS 输出传输 SCK 延迟使能 0：失能 DQS 传输延迟 1：使能DQS传输延迟
8	CSNADD	CSN增加1/2个周期 0：不增加 1：增加1/2周期
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：$F_{CLK} = F_{KERNEL}$。这种情况下不支持DTR模式。 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%，时钟信号保持高电平时间要比高电平时间多一个周期。

27.11.4. 设备配置寄存器 2(OSPI_DCFG2)

偏移地址：0x10
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



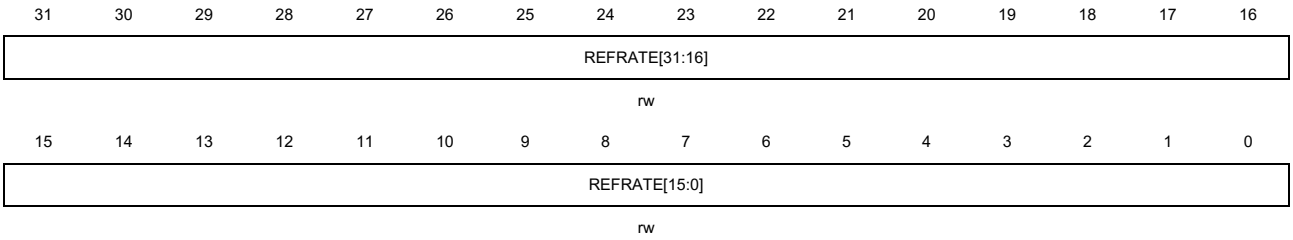
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	CSBND[4:0]	片选边界 此字段用于启用事务边界功能。启用时，建议将最小值设置为3。 在每2^{CSBND}字节的边界处，nCS信号将被释放。 0：禁用片选边界功能 其他值：将片选边界设置为2^{CSBND}字节
15:0	保留	必须保持复位值。

27.11.5. 设备配置寄存器 3(OSPI_DCFG3)

偏移地址：0x14
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



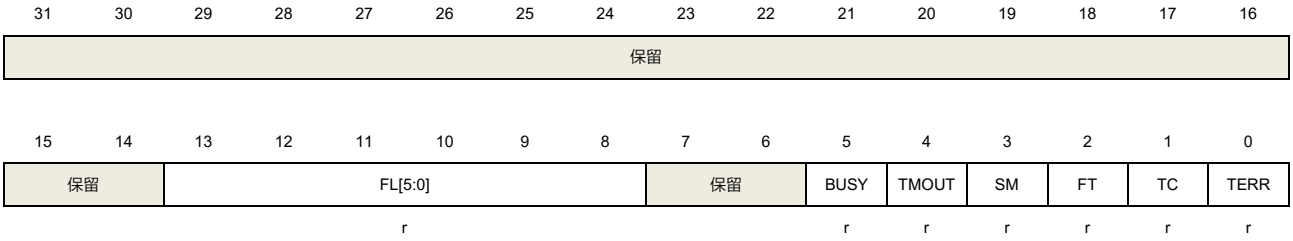
位/位域	名称	描述
31:0	REFRATE[31:0]	刷新速率 仅支持内存映射模式下8线读 0：禁用刷新。

其它值：最大通信长度为REFRATE+1个时钟周期

27.11.6. 状态寄存器(OSPI_STAT)

偏移地址：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
31:14	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	TMOUT	超时标志 超时发生时该位置1，向TMOUTC位写1清0。当这一位置起后，内存映射模式下读不再保证数据正确性，在下一次传输前，需要清零这一位。
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，或在任何其它模式下发生中止时，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。

27.11.7. 状态清除寄存器(OSPI_STATC)

偏移地址：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											TMOUTC	SMC	保留	TCC	TERRC
											w	w		w	w

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TMOUTC	清除超时标志 写1清除状态寄存器的TMOUT标志。
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。

27.11.8. 数据长度寄存器(OSPI_DTLEN)

偏移地址：0x40
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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[4:0]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[4:0]定义)，如果MESZ[4:0]为0x1F，无限读数据。 内存映射模式下，该位无影响。 **注意：**DTR模式不支持奇数字节，如果配置成奇数字节可能导致非预期错误。

27.11.9. 地址寄存器(OSPI_ADDR)

偏移地址：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。 在HyperBus™模式下，写入该寄存器的值必须为偶数。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。

27.11.10. 数据寄存器(OSPI_DATA)

偏移地址：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。

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

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

位/位域	名称	描述
31:0	DATA[31:0]	将要与存储器交互的数据。 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。

27.11.11. 状态屏蔽寄存器(OSPI_STATMK)

偏移地址：0x80
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:0	MASK[31:0]	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。

27.11.12. 状态匹配寄存器(OSPI_STATMATCH)

偏移地址：0x88
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.13. 间隔寄存器(OSPI_INTERVAL)

偏移地址：0x90
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.14. 传输配置寄存器 0(OSPI_TCFG0)

偏移地址：0x100
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	保留	DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
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		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			保留		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31	SIOO	只发送一次指令模式 IMOD = 00时该位没有影响。 0：每次命令序列都发送指令。 1：命令序列第一次时发送指令。
30	保留	必须保持复位值。
29	DQSEN	DQS使能 0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。 其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	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	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。
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。

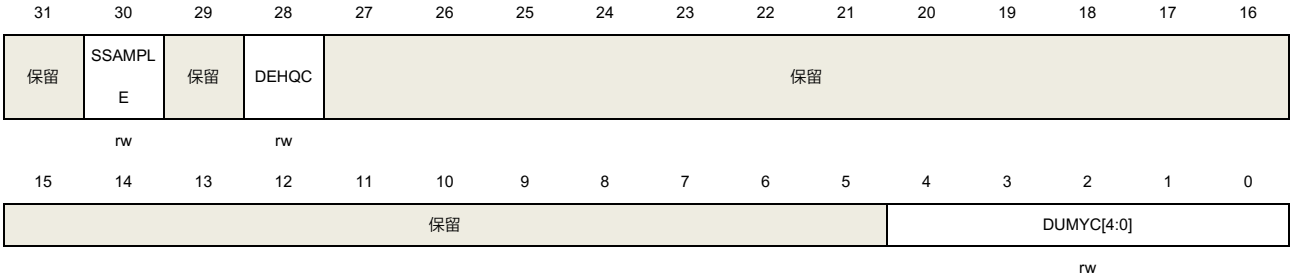
100：八线传输指令。
101：保留。
110：保留。
111：保留。

27.11.15. 时序配置寄存器 0(OSPI_TIMCFG0)

偏移地址：0x108
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

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	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。
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。 **注意：**为保证数据采样正确，该位必须设置为1
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，当使用DQS时，至少6个空指令周期。

27.11.16. 指令寄存器 0(OSPI_INS0)

偏移地址：0x110

复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.17. 交替字节寄存器 0(OSPI_ALTE0)

偏移地址：0x120
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.18. 低功耗超时寄存器(OSPI_LPOUT)

偏移地址：0x130
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

保留

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TIMEOUT	超时周期 当OSPI工作在内存映射模式下，访问结束后，该位域表明在时钟变为无效后，OSPI片选保持低电平的SCK周期数，将外部存储器设置为低功耗状态。

27.11.19. 回卷传输配置寄存器 0(OSPI_WPTCFG0)

偏移地址：0x140
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
		rw		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]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DQSEN	DQS使能 0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。

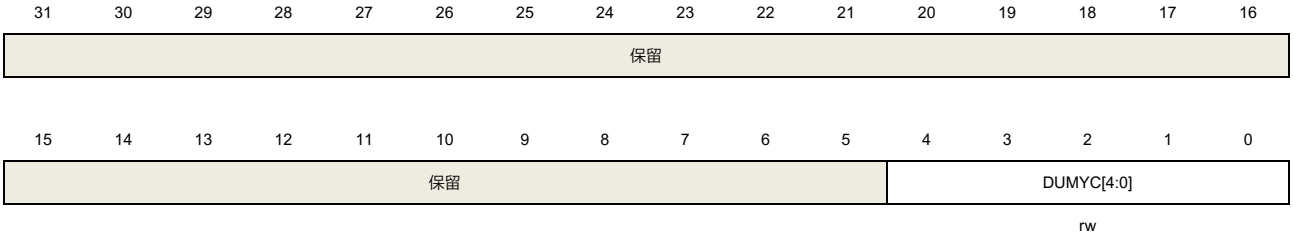
		其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	指令大小 该位域定义指令大小。 00：8位指令。 01：16位指令。 10：24位指令。 11：32位指令。
3	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。
2:0	IMOD[2:0]	命令模式 该位定义指令阶段的操作模式： 000：无指令。 001：单线传输指令。 010：双线传输指令。 011：四线传输指令。 100：八线传输指令。 其他：保留。

27.11.20. 回卷时序配置寄存器 0(OSPI_WPTIMCFG0)

偏移地址：0x148
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

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。
4:0	DUMYC[4:0]	空指令周期数 该位域定义空指令阶段持续时间，当使用DQS时，至少5个空指令周期。

27.11.21. 回卷指令寄存器 0(OSPI_WPINS0)

偏移地址：0x150
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.22. 回卷交替字节寄存器 0(OSPI_WPALTE0)

偏移地址：0x160
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.23. 写入传输配置寄存器 0(OSPI_WTCFG0)

偏移地址：0x180
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
		rw		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]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DQSEN	DQS使能 0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。 其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	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	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。
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。

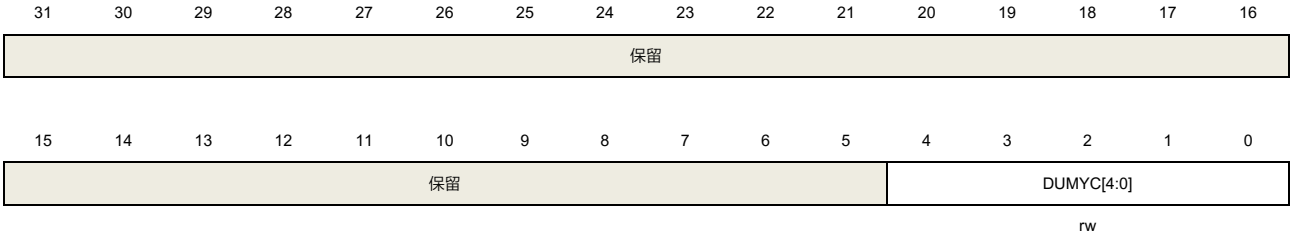
010：双线传输指令。
011：四线传输指令。
100：八线传输指令。
其他：保留。

27.11.24. 写入时序配置寄存器 0(OSPI_WTIMCFG0)

偏移地址：0x188
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



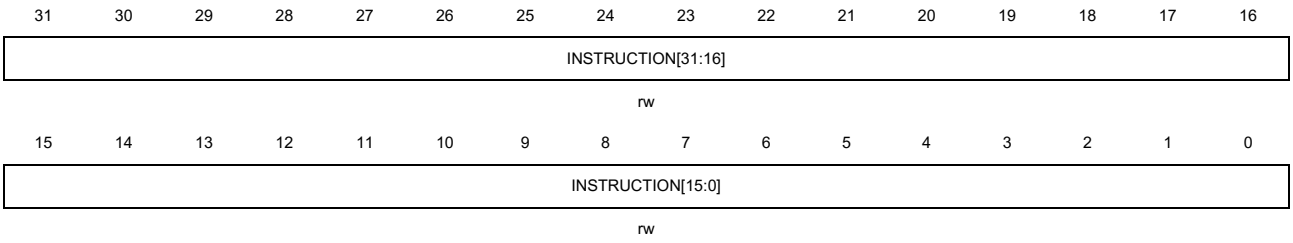
位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	DUMYC[4:0]	空指令周期数 该位域定义空指令阶段持续时间，当使用DQS时，至少5个空指令周期。

27.11.25. 写入指令寄存器 0(OSPI_WINS0)

偏移地址：0x190
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:0	INSTRUCTION[31:0]	指令

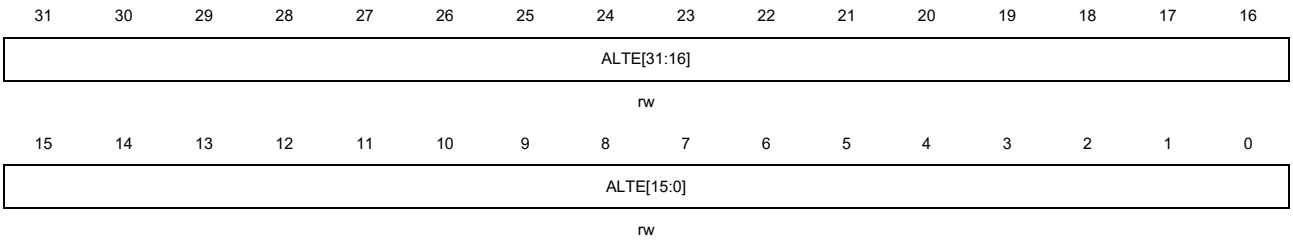
发送到外部存储器的命令信息。

27.11.26. 写入交替字节寄存器 0(OSPI_WALTE0)

偏移地址：0x1A0
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。



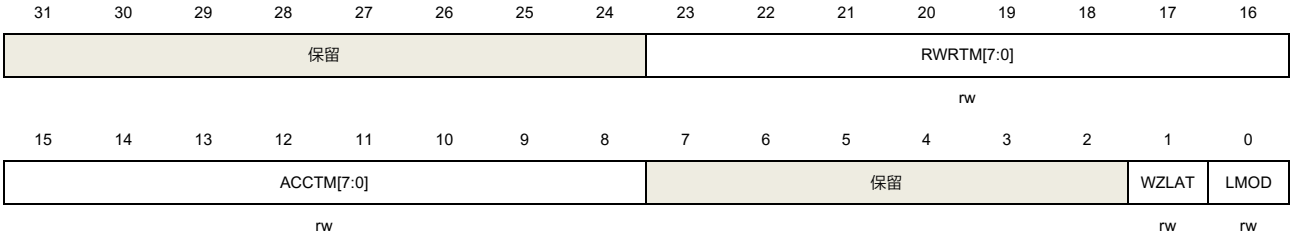
位/位域	名称	描述
31:0	ALTE[31:0]	交替字节 发送给外部存储器的可选数据。

27.11.27. HyperBusTM 延迟配置寄存器(OSPI_HBLCFG)

偏移地址：0x200
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

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	RWRTIM[7:0]	读写恢复时间 读写恢复时间用SCK周期数表示。
15:8	ACCTM[7:0]	访问时间 访问时间用SCK周期数表示。

7:2	保留	必须保持复位值。
1	WZLAT	写入零延迟使能 0：禁用写入操作零延迟。 1：使能写入操作零延迟。
0	LMOD	延迟模式 0：可变初始延迟。 1：固定延迟。

27.11.28. 输配置寄存器 1(OSPI_TCFG1)

偏移地址：0x300
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	保留	DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
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		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			保留		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31	SIOO	只发送一次指令模式 IMOD = 00时该位没有影响。 0：每次命令序列都发送指令。 1：命令序列第一次时发送指令。
30	保留	必须保持复位值。
29	DQSEN	DQS使能 0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。

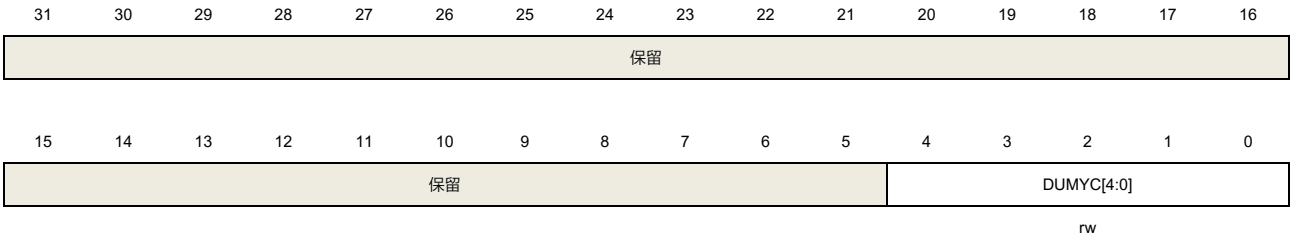
		100：八线传输地址。 其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	指令大小 该位域定义指令大小。 00：8位指令。 01：16位指令。 10：24位指令。 11：32位指令。
3	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。
2:0	IMOD[2:0]	命令模式 该位定义指令阶段的操作模式： 000：无指令。 001：单线传输指令。 010：双线传输指令。 011：四线传输指令。 100：八线传输指令。 101：保留。 110：保留。 111：保留。

27.11.29. 时序配置寄存器 1(OSPI_TIMCFG1)

偏移地址：0x304
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

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。
4:0	DUMYC[4:0]	空指令周期数

该位域定义空指令阶段持续时间，当使用DQS时，至少6个空指令周期。

27.11.30. 指令寄存器 1(OSPI_INS1)

偏移地址：0x308
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.31. 交替字节寄存器 1(OSPI_ALTE1)

偏移地址：0x30C
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.32. 回卷传输配置寄存器 1(OSPI_WPTCFG1)

偏移地址：0x310
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
		rw		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]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DQSEN	DQS使能 0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。 其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	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	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。

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。 其他：保留。

27.11.33. 回卷时序配置寄存器 1(OSPI_WPTIMCFG1)

偏移地址：0x314
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

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
31:5	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，当使用DQS时，至少5个空指令周期。

27.11.34. 回卷指令寄存器 1(OSPI_WPINS1)

偏移地址：0x318
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.35. 回卷交替字节寄存器 1(OSPI_WPALTE1)

偏移地址：0x31C
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.36. 写入传输配置寄存器 1(OSPI_WTCFG1)

偏移地址：0x320
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DQSEN	保留	DADTR	DATAMOD[2:0]			保留		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
		rw		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]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DQSEN	DQS使能

		0：禁用DQS。 1：使能DQS。
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。
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。 其他：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。
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。 其他：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：地址阶段能是能DTR模式。
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。 其他：保留。
7:6	保留	必须保持复位值。
5:4	INSSZ[5:4]	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	INSDTR	指令双倍传输速率 0：指令阶段禁用DTR。 1：指令阶段使能DTR。
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。 其他：保留。

27.11.37. 写入时序配置寄存器 1(OSPI_WTIMCFG1)

偏移地址：0x324
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

保留

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
31:5	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，当使用DQS时，至少5个空指令周期。

27.11.38. 写入指令寄存器 1(OSPI_WINS1)

偏移地址：0x328
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

27.11.39. 写入交替字节寄存器 1(OSPI_WALTE1)

偏移地址：0x32C
复位值：0x0000 0000

当 BUSY 位为 1，该寄存器不可修改。

该寄存器只能按字(32 位)访问。

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

28. 外部存储器控制器(EXMC)

28.1. 简介

外部存储器控制器 EXMC，用来访问各种片外存储器，通过配置寄存器，EXMC 可以把 AMBA 协议转换为专用的片外存储器通信协议，包括 PSRAM，NOR Flash，NAND Flash。用户还可以调整配置寄存器中的时间参数来提高通信效率。EXMC 的访问空间被划分为许多个块(Bank)，每个块支持特定的存储器类型，用户可以通过对 Bank 的控制寄存器配置来控制外部存储器。

28.2. 主要特性

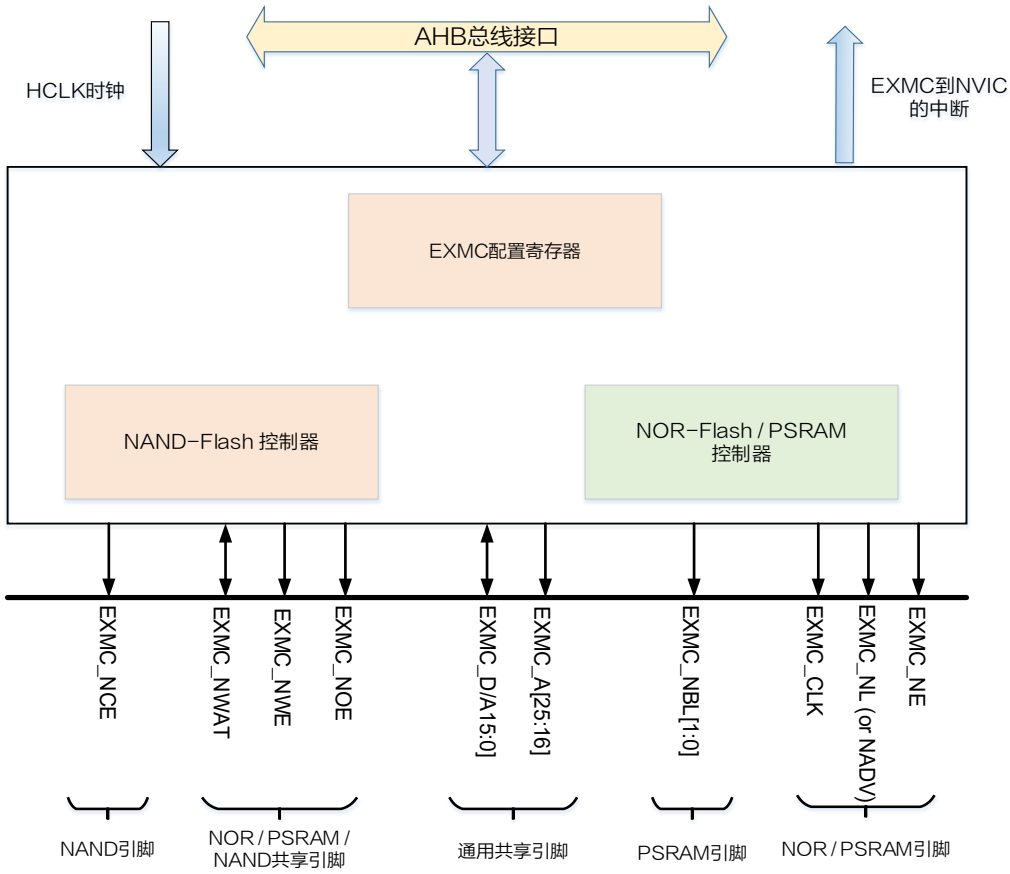
- 支持片外存储器类型：
  - PSRAM；
  - NOR Flash；
  - 8 位或 16 位 NAND Flash；
- AMBA 协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 每个 Bank 有独立的片选信号；
- 对于 NAND Flash 内置硬件 ECC；
- 支持 8 位或 16 位总线带宽；
- NOR Flash 和 PSRAM 支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当 AMBA 总线宽度与外部存储器数据宽度不同时，会自动分割操作。

28.3. 功能描述

28.3.1. 结构框图

EXMC 由 5 个模块组成：AHB 总线接口，EXMC 配置寄存器，NOR/PSRAM 控制器，NAND 控制器和外部设备接口。AHB 时钟(HCLK)是参考时钟。

图 28-1. 系统架构



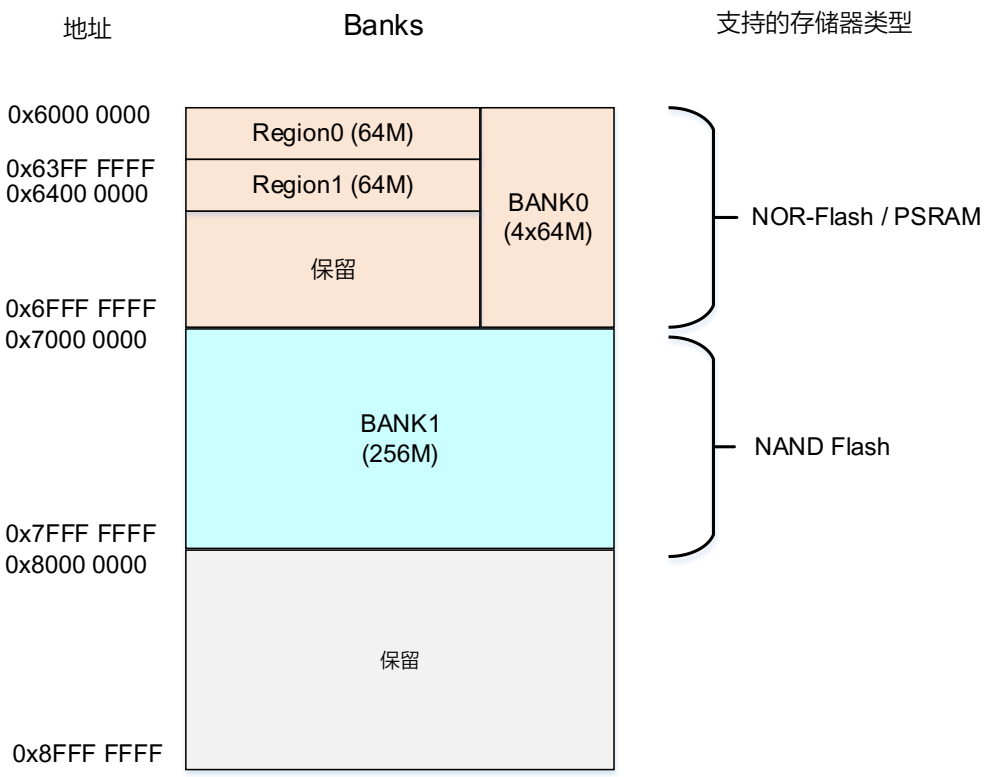
28.3.2. EXMC 访问基本规范

EXMC 是 AHB 总线至外部设备协议的转换接口。32 位的 AHB 读写操作可以转化为几个连续的 8 位或 16 位读写操作。在数据传输的过程中，AHB 数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，EXMC 读写访问需要遵从以下规范：

- AHB 访问宽度等于存储器宽度，则没有问题；
- AHB 访问宽度大于存储器宽度，则自动将 AHB 访问分割成几个连续的存储器数据宽度的传输；
- AHB 访问宽度小于存储器宽度。如果外部存储设备具有字节选择功能，如 SRAM，ROM，PSRAM，则可通过它的字节通道 EXMC_NBL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

28.3.3. 外部设备地址映射

图 28-2. EXMC Bank 划分



EXMC 将外部存储器分成多个 Bank，每个 Bank 占 256M 字节，其中 Bank0 又分为 4 个 Region，每个 Region 占 64M 字节。Bank1 被分成 2 个 Section，分别是属性存储空间和通用存储空间。

每个 Bank 或 Region 都有独立的片选控制信号，也都能进行独立的配置。

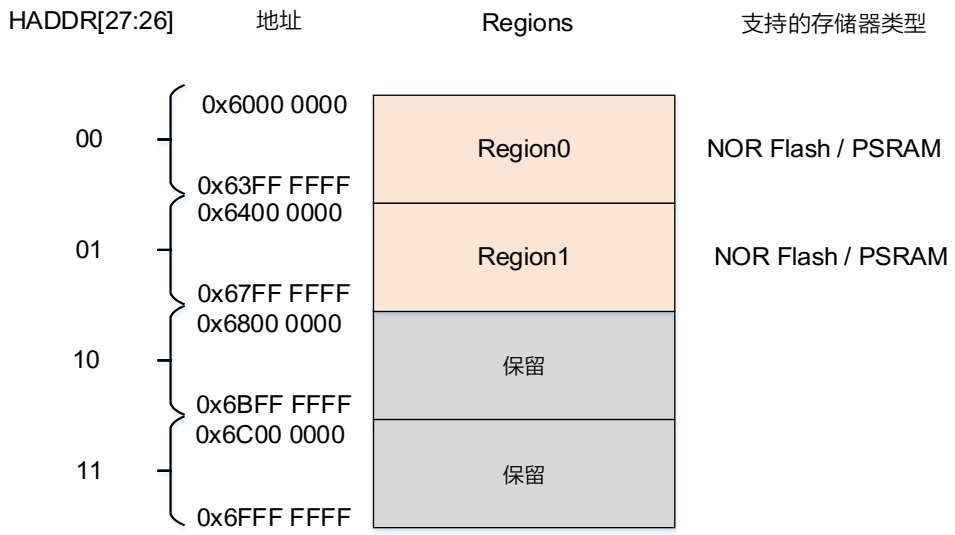
Bank0 用于访问 NOR-Flash、PSRAM 设备。

Bank1 用于连接 NAND Flash。

NOR 和 PSRAM 的地址映射

*图 28-3. Bank0 地址映射*是 Bank0 四个 Region 的地址映射。AHB 地址线 HADDR[27:26]作为四个 Region 的片选信号。

图 28-3. Bank0 地址映射



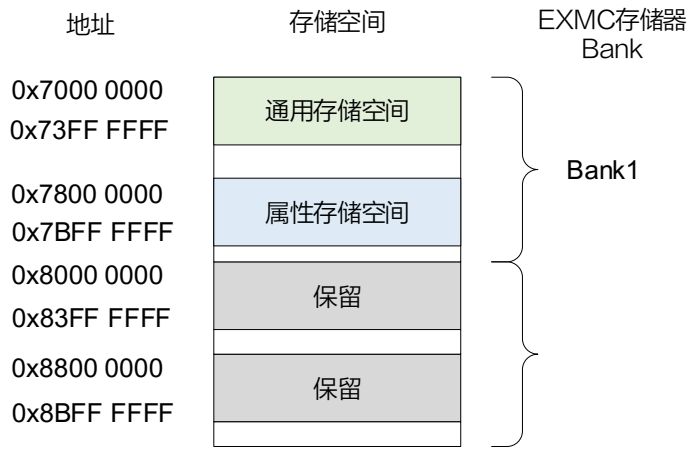
由于 HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但 EXMC 能实现对 HADDR 的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是 8 位按字节对齐，EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；
- 如果外部存储器的数据宽度是 16 位按半字对齐，就需要将 HADDR 的字节地址转化为半字地址之后再连接外存储器，EXMC 内部将 HADDR[25:1]与 EXMC_A[24:0]相连，然后 EXMC_A[24:0]与外部存储器的地址线相连。

NAND 地址映射

Bank1 用来访问 NAND Flash，Bank1 如*图 28-4. NAND 地址映射*被分为多个存储空间。

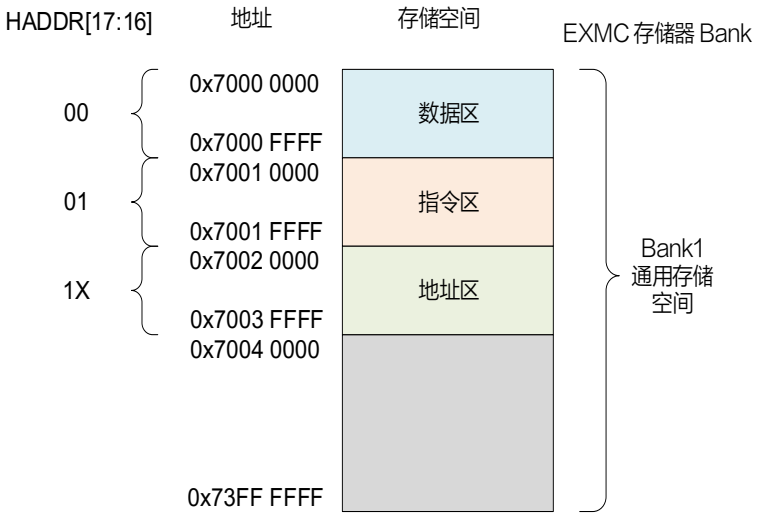
图 28-4. NAND 地址映射



对于 NAND Flash，通用和属性空间又可以细划分为 3 个区域。*图 28-5. Bank1 通用空间*为

Bank1 通用存储空间的数据区域，指令区域和地址区域的划分。

图 28-5. Bank1 通用空间



AHB 利用 HADDR[17:16]来实现对以上三个区的选择：

- HADDR[17:16]=00，即选择数据区。
- HADDR[17:16]=01，即选择命令区。
- 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：

1. 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。
2. 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]。
3. 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]。

28.3.4. NOR/PSRAM 控制器

EXMC 模块的 NOR/PSRAM 控制器控制 Bank0，它可以支持 NOR Flash、PSRAM。EXMC 对 Bank0 每个 Region 输出一个唯一的片选信号，EXMC_NE[x](x=0,1)，用于 Region 的片选，所有其他的信号都是共享的。Region 有专门的寄存器控制。

注意：

在异步模式下，所有控制器输出信号在内部 AHB 总线时钟(HCLK)的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟(EXMC_CLK)的下降沿改变。

NOR/PSRAM 接口描述

表 28-1. NOR Flash 接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
复用 EXMC_A[25:16]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步 (复用)	地址/数据总线
EXMC_NE[x]	输出	异步/同步	片选 x=0/1
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 28-2. PSRAM 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
复用 EXMC_A[25:16]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步	地址/数据总线
EXMC_NE[x]	输出	异步/同步	片选 x=0/1
EXMC_NOE	输出	异步/同步	读使能
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL(NADV)	输出	异步/同步	地址锁存信号(地址有效使能，NADV)
EXMC_NBL[1]	输出	异步/同步	高字节使能
EXMC_NBL[0]	输出	异步/同步	低字节使能

支持的存储器访问模式

*表 28-3. EXMC 的 Bank0 支持的访问模式*列出了 EXMC 对 NOR 和 PSRAM 支持的访问模式。

表 28-3. EXMC 的 Bank0 支持的访问模式

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
NOR Flash	异步	R	8	16	
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
	同步	R	16	16	
	同步	R	32	16	
PSRAM	异步	R	8	16	
	异步	W	8	16	使用字节信号 NBL[1:0]
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问
	同步	R	16	16	
	同步	R	32	16	
	同步	W	8	16	使用字节信号 NBL[1:0]
	同步	W	16	16	
	同步	W	32	16	

NOR Flash/PSRAM 控制时序

EXMC 为 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 28-4. NOR/PSRAM 控制时序参数

参数	功能	访问模式	单位	最小值	最大值
CKDIV	同步时钟分频比	同步	HCLK	2	16
DLAT	数据延迟	异步	EXMC_CLK	2	17
BUSLAT	总线延迟	异步读写	HCLK	1	16
DSET	数据建立时间	异步	HCLK	2	256
AHLD	地址保持时间	异步(复用)	HCLK	2	16
ASET	地址建立时间	异步	HCLK	1	16

表 28-5. EXMC 时序模型

时序模型		模式描述	写时序参数	读时序参数
异步	模式 AM	NOR Flash 数据 / 地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 SM	NOR Flash(读)/PSRAM(CRAM) 数据 / 地址复用	DLAT CKDIV	DLAT CKDIV

如*表 28-5. EXMC 时序模型*所示，EXMC 模块 NOR Flash/PSRAM 控制器可以提供多种时序模型。用户可以通过修改*表 28-4. NOR/PSRAM 控制时序参数*中列出的参数来使之适合不同

类型外部存储器的时序以及满足用户的要求。

异步访问时序

模式 AM - NOR Flash 地址/数据总线复用

图 28-6. 复用模式读时序

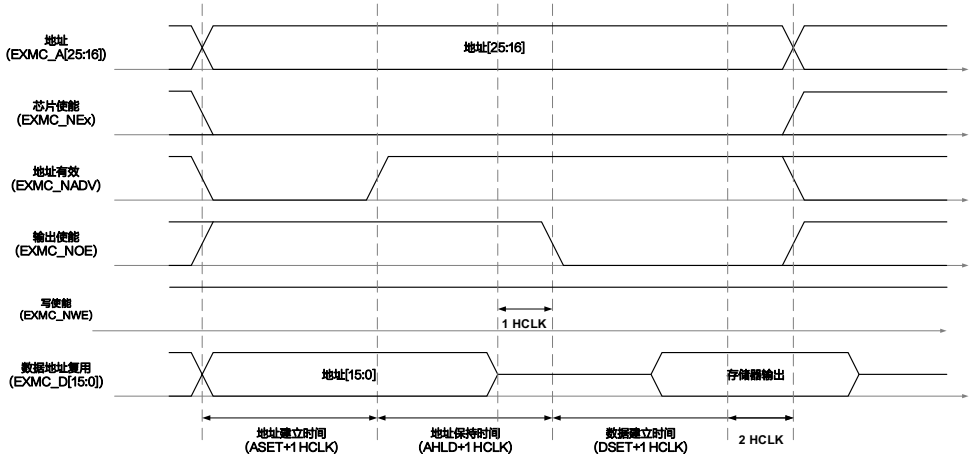


图 28-7. 复用模式写时序

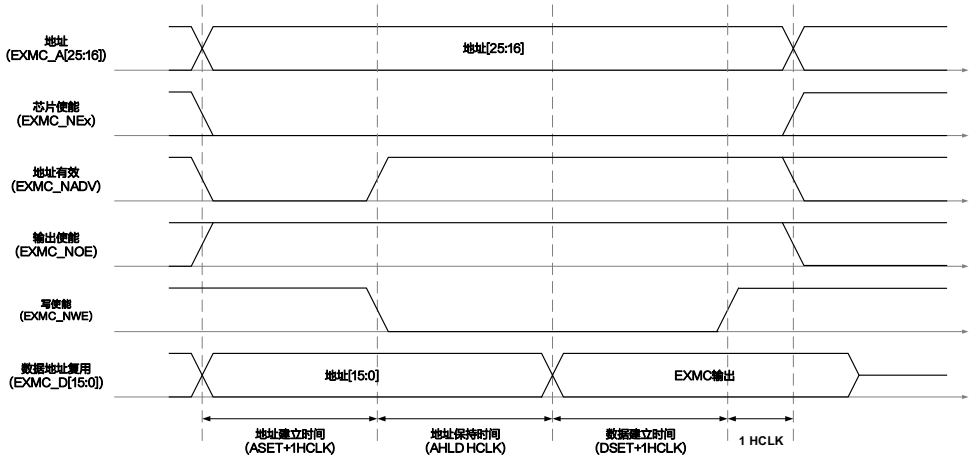


表 28-6. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_PNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	保留	0x0
13	NRWTEN	0x0
12	WEN	取决于存储器
11	NRWTCFG	无影响

位域/位	名称	参考设定值
EXMC_PNCTLx		
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	0x2：NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_PNTCFGx		
31-28	保留	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+2HCLK 时钟周期，读操作为 DSET+3HCLK 时钟周期)
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

异步通信的等待时间：

等待功能由寄存器 EXMC_PNCTLx 位 ASYNCWAIT 控制。在访问外部存储器期间，若使能异步等待功能(ASYNCWAIT=1)，数据建立时间将会自动延长。延长时间的计算如下：

若存储器等待信号与 EXMC_NOE / EXMC_NWE 信号对齐：

$$T_{DATA_SETUP} >= maxT_{WAIT_ASSERTION} + 4HCLK \quad ((28\text{-}1)$$

若存储器等待信号与 EXMC_NE[x]信号对齐：

如果

$$maxT_{WAIT_ASSERTION} >= T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (28\text{-}2)$$

则

$$T_{DATA_SETUP} >= (maxT_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (28\text{-}3)$$

否则

$$T_{DATA_SETUP} >= 4HCLK \quad (28\text{-}4)$$

图 28-8. 异步等待有效时的读时序

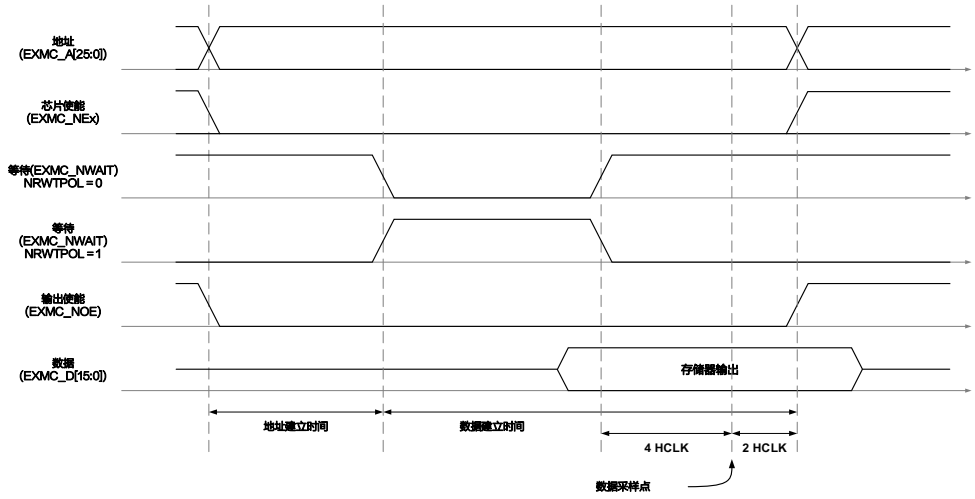
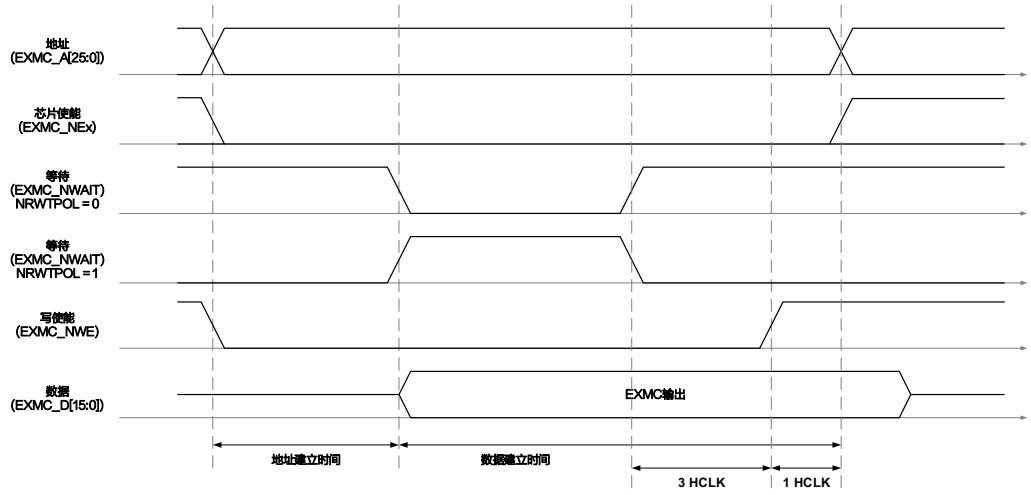


图 28-9. 异步等待有效时的写时序



同步访问时序

同步访问模式中，存储器时钟(EXMC_CLK)与系统时钟(HCLK)关系如下：

$$EXMC_CLK=\frac{HCLK}{CKDIV+1} \tag{28-5}$$

其中 CKDIV 是同步时钟分频比，通过配置寄存器 EXMC_PNTCFGx 中的 CKDIV 位来设置不同的值。

1. 数据延迟与 NOR Flash 延迟

数据延迟 DLAT 是指在采样数据之前需要等待的 EXMC_CLK 周期数。它和 NOR 闪存延迟的关系如下：

NOR 闪存延迟不包含 NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+2 (28-6)

NOR 闪存延迟包含 NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+3 (28-7)

2. 数据等待

用户需要保证 EXMC_NWAIT 信号与外部设备一致。该信号通过寄存器 EXMC_PNCTLx 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_PNCTLx 位 NRWTEN 置 1，在数据延迟之后后检测到 EXMC_NWAIT 信号。如果 EXMC_NWAIT 有效，在 EXMC_NWAIT 无效之前会一直插入等待时钟。

■ EXMC_NWAIT 有效极性：

NRWTPOL = 1，EXMC_NWAIT 高电平有效

NRWTPOL = 0，EXMC_NWAIT 低电平有效

■ 在同步突发模式中，EXMC_NWAIT 信号有两种配置：

NRWTCFG = 1，EXMC_NWAIT 信号有效时，当前时钟周期数据无效

NRWTCFG = 0，EXMC_NWAIT 信号有效时，下一个时钟周期数据无效，这是复位后的默认配置。

在 EXMC_NWAIT 信号有效的等待周期内，EXMC 会持续的给存储器发送时钟信号，保持片选和输出使能有效，并且忽视总线上的无效数据。

3. CRAM 页边界突发传输的自动分组

CRAM1.5 中禁止突发传输跨越页边界，EXMC 遇到边界会进行传输的自动分组。为了保证正确的突发分组操作，用户需要在寄存器 EXMC_PNCTLx 位 CPS 中需要设定 CRAM 的页大小。

4. 模式 SM - 单次突发传输

对于同步突发传输，如果 AHB 需要的数据为 16 位，则 EXMC 会执行一次长度为 1 的成组传输；如果 AHB 需要的数据为 32 位，则 EXMC 会把这次传输分成 2 次 16 位的传输，即执行一次长度为 2 的突发传输。

对于其他的配置，请参考*表 28-3. EXMC 的 Bank0 支持的访问模式*。

同步复用突发读时序 - NOR,PSRAM(CRAM)

图 28-10. 同步复用突发传输读时序

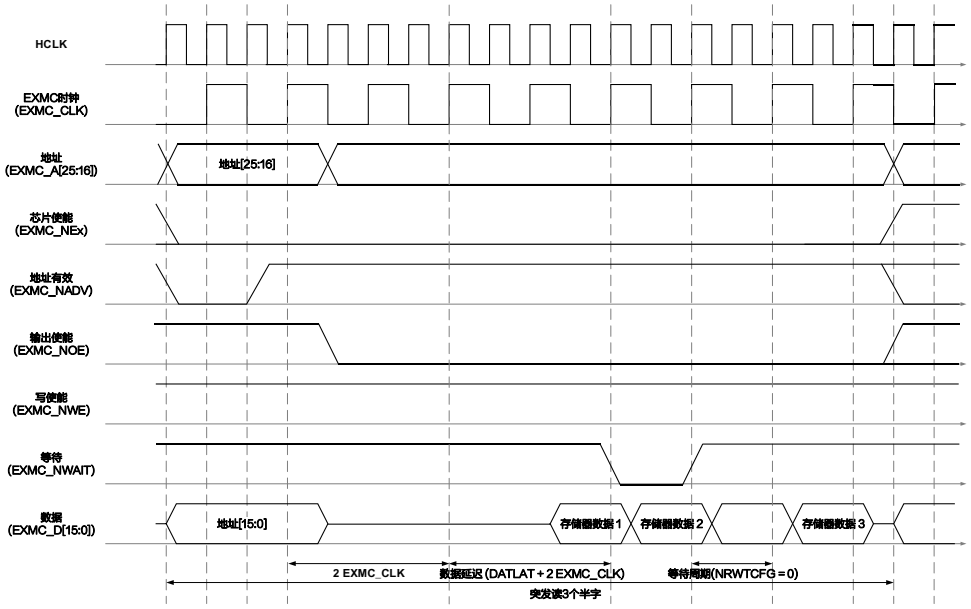


表 28-7. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_PNCTLx		
31-20	保留	0x000
19	SYNCWR	无影响
18-16	CPS	0x0
15	ASYNCWTEN	0x0
14	保留	0x0
13	NRWTEN	取决于存储器
12	WEN	无影响
11	NRWTCFG	取决于存储器
10	WRAPEN	0x0
9	NRWTPOL	取决于存储器
8	SBRSTEN	0x1，突发读使能
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	取决于存储器，0x1/0x2
1	NRMUX	0x1，取决于存储器与用户
0	NRBKEN	0x1
EXMC_PNTCFGx(Read)		
31-28	保留	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	上图设置：0x1,EXMC_CLK=2HCLK
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间

位域/位	名称	参考设定值
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

同步复用突发写时序 - NOR,PSRAM(CRAM)

图 28-11. 同步复用突发传输写时序

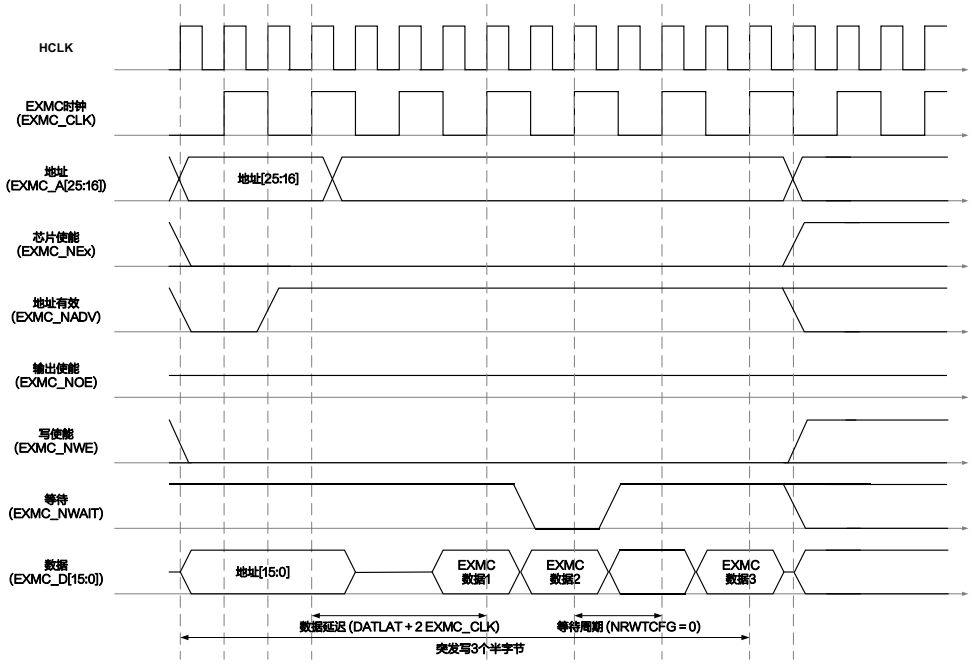


表 28-8. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_PNCTLx		
31-20	保留	0x000
19	SYNCWR	0x1，同步写使能
18-16	CPS	0x0
15	AYSNCWAIT	0x0
14	保留	0x0
13	NRWTEN	取决于存储器
12	WREN	0x1
11	NRWTCFG	0x0(这里必须为 0)
10	WRAPEN	0x0
9	NTWTPOL	取决于存储器
8	SBRSTEN	无影响
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	0x1

位域/位	名称	参考设定值
1	NRMUX	0x1，取决于用户
0	NRBKEN	0x1
EXMC_PNTCFGx(写)		
31-28	保留	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	上图设置：0x1，EXMC_CLK=2HCLK
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

28.3.5. NAND Flash 控制器

EXMC 模块 Bank1 支持 NAND Flash。对于 Bank1，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 28-9. 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	输入	NAND Flash 就绪/忙输入信号 EXMC

支持的存储器访问模式

表 28-10. Bank1 支持的访问模式

存储器	模式	读/写	AHB 传输宽度	注释
8 位 NAND	异步	R	8	
	异步	W	8	
	异步	R	16	分成 2 次 EXMC 访问
	异步	W	16	
	异步	R	32	分成 4 次 EXMC 访问

	异步	W	32	
16 位 NAND	异步	R	8	
	异步	W	8	不支持此操作
	异步	R	16	
	异步	W	16	
	异步	R	32	分成 2 次 EXMC 访问
	异步	W	32	

NAND Flash 的控制时序

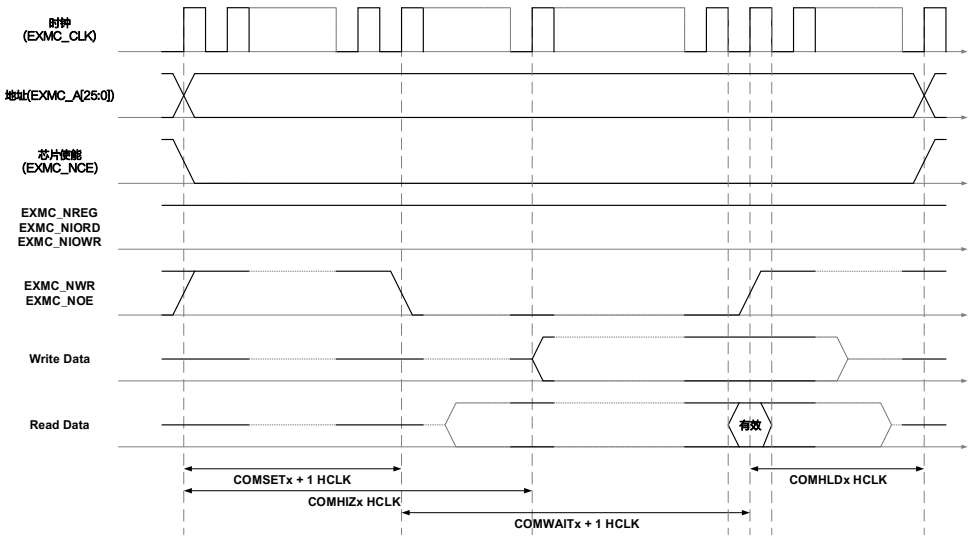
EXMC 能够为 NAND Flash 设备产生合适的时序信号。每个 Bank 都有相应的寄存器来对外部存储器进行管理和控制，EXMC_NCTL、EXMC_NSTAT、EXMC_NCTCFG、EXMC_NATCFG 和 EXMC_NECC，其中寄存器 EXMC_NCTCFG、EXMC_NATCFG 都可以配置 4 个时序参数，可以根据用户需求和外部存储器的特性来进行相应的配置。

表 28-11. NADN 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	HCLK	启动写操作之后保持数据总线为高阻态的时间	0	255
存储器保持时间(HLD)	W/R	HCLK	在发送命令结束后保持地址的(HCLK)时钟周期数目，写操作时也是数据的保持时间	1	254
存储器等待时间(WAIT)	W/R	HCLK	发出命令的最短持续时间(HCLK)时钟周期数目	2	256
存储器建立时间(SET)	W/R	HCLK	发出命令之前建立地址的(HCLK)时钟周期数目	1	255

*图 28-12. NAND flash 通用空间操作时序*显示了在公共存储空间操作中定义的可编程参数。属性存储空间的可编程参数也已定义。

图 28-12. NAND flash 通用空间操作时序



NAND Flash 操作

EXMC 在对 NAND Flash 发送命令或地址时，需要利用其命令锁存信号(EXMC_A[16])或地址锁存信号(EXMC_A[17])这两条地址线，即 CPU 需要在特定的地址进行写操作。

示例：NAND Flash 读操作步骤：

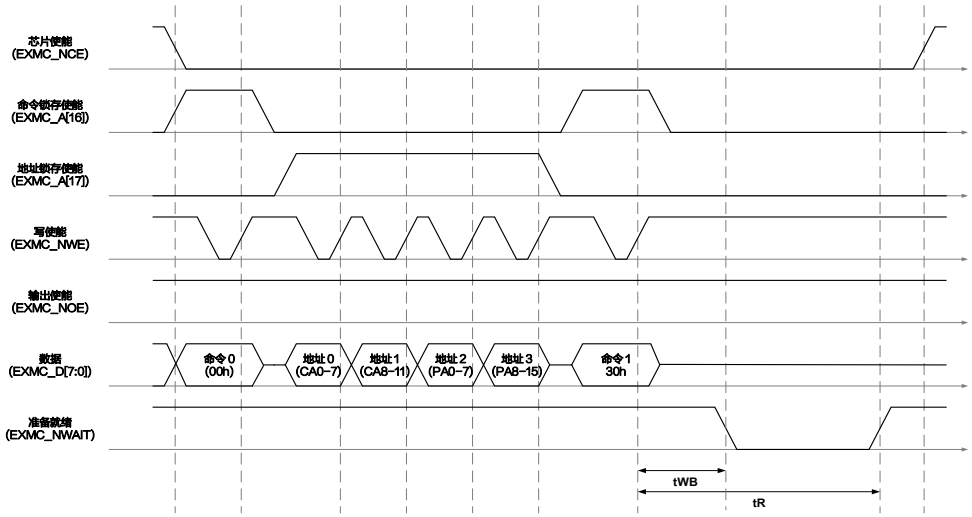
1. 配置 EXMC_NCTL 和 EXMC_NCTCFG，若需要预等待功能，还需配置 EXMC_NATCFG；
2. 往通用空间写入 NAND Flash 读数据命令，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_CLE(EXMC_A[16])变为有效电平(高)，则被 NAND 认为写入命令；
3. 往通用空间写入读操作的起始地址，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_ALE(EXMC_A[17])变为有效电平(高)，则被 NAND 认为写入地址；
4. 等待 NAND 就绪信号，NAND 控制器会在这期间将和 EXMC_NCE 一直保持有效；
5. 从通用空间的数据区逐字节的读出数据；
6. 在不写入新的命令和地址，可以自动读出 NAND 下一页数据；或转到步骤 3)写入新的地址进行下一页的读取；或转到步骤 2)写入新的命令和地址。

NAND Flash 预等待功能

某些 NAND Flash 要求在输入最后一个地址字节后，控制器等待 NAND Flash 就绪，并且还有一些对 EXMC_NCE 敏感型的 NAND Flash 还要求在其就绪前 NCE 必须保持有效。

下面以 TOSHIBA128M*8bit NADN Flash 为例：

图 28-13. 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_NWAIT 信号，ATTHLD 要大于 t_{WB}(EXMC_NWE 高到 EXMC_NWAIT 低)。

对于那些对片选信号 EXMC_NCE 敏感的 NAND Flash，在地址字节之后的第一个地址字节输入后，EXMC_NCE 必须保持低电平，直到 EXMC_NWAIT 从低电平变为高电平。可以在 EXMC_NATCFG 寄存器中设置属性空间的 ATTHLD 值，以满足 t_{WB} 的时序要求。这样 CPU 只有在地址字节之后写入第一个命令字节时才使用属性存储空间的时序，而在其他时候都使用通用存储空间的时序。

NAND Flash 的 ECC 计数模块

EXMC 模块中的 Bank1 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 计算可以矫正 1 个 bit 的错误并且能检测 2 个 bit 的错误。

当 NAND 存储器块使能，ECC 模块就会检测 D[15:0]以及 EXMC_NCE、EXMC_NWE 信号。当已经完成 ECCSZ 大小字节的读写操作时，软件必须读出 EXMC_NECC 中的结果值。如果需要再次开始 ECC 计算，软件需要先将 EXMC_NCTL 中 ECCEN 清 0 来清除 EXMC_NECC 中的值，再将 ECCEN 置 1 来重新启动 ECC 计算。

28.4. EXMC 寄存器

EXMC 基地址：0xA000 0000

28.4.1. NOR/PSRAM 控制器寄存器

PSRAM/NOR Flash 控制寄存器 (EXMC_PNCTLx) (x=0, 1)

地址偏移：0x00 + 8 * x, (x = 0, 1)
复位值：0x0000 30DX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												SYNCWR	CPS[2:0]		
												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	保留	NRWTEN	WREN	NRWTCFG	WRAPEN	NRWTPOL	SBRSTEN	保留	NREN	NRW[1:0]		NRTP[1:0]		NRMUX	NRBKEN
rw		rw	rw	rw	rw	rw	rw		rw	rw		rw		rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	SYNCWR	选择写操作模式 0：异步写操作 1：同步写操作
18:16	CPS[2:0]	CRAM页大小 000：页边界自动突发分割 001：128字节 010：256字节 011：512字节 100：1024字节 其他：保留
15	ASYNCWAIT	异步等待功能使能位 0：禁用异步等待功能 1：使能异步等待功能
14	保留	必须保持复位值。
13	NRWTEN	NWAIT信号使能 对于存储器的突发模式访问，该位使能/禁用等待状态插入NWAIT信号功能 0：禁用NWAIT信号

		1：使能NWAIT信号
12	WREN	写操作使能 0：禁止EXMC对外部存储器的写操作，否则产生一个AHB错误 1：允许EXMC对外部存储器的写操作(复位缺省值)
11	NRWTCFG	NWAIT信号配置，只在同步模式有效 0：NWAIT信号在等待状态前的一个数据周期有效 1：NWAIT信号在等待状态期间有效
10	WRAPEN	非对齐成组模式使能 0：禁止非对齐成组操作 1：允许非对齐成组操作
9	NRWTPOL	NWAIT信号极性 0：NWAIT低电平有效 1：NWAIT高电平有效
8	SBRSTEN	同步突发模式使能 0：禁止同步突发模式 1：使能同步突发模式
7	保留	必须保持复位值。
6	NREN	NOR闪存访问使能 0：禁止NOR Flash访问 1：允许NOR Flash访问
5:4	NRW[1:0]	存储器数据宽度 00：8位 01：16位(复位缺省值) 10/11：保留
3:2	NRTP[1:0]	存储器类型 00：保留 01：PSRAM(CRAM) 10：NOR Flash(region0复位之后的默认值) 11：保留
1	NRMUX	数据线/地址线复用 0：禁用地址/数据复用功能 1：允许地址/数据复用功能
0	NRBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块

PSRAM/NOR Flash 时序寄存器 (EXMC_PNTCFGx) (x=0, 1)

地址偏移：0x04 + 8 * 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				DLAT[3:0]				CKDIV[3:0]				BUSLAT[3:0]			
				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
31:28	保留	必须保持复位值。
27:24	DLAT[3:0]	NOR Flash数据延时，仅在同步模式有效 0x0：第一数据的保持时间为2个EXMC_CLK时钟周期 0x1：第一数据的保持时间为3个EXMC_CLK时钟周期 …… 0xF：第一数据的保持时间为17个EXMC_CLK时钟周期
23:20	CKDIV[3:0]	同步模式时钟分频比，仅在同步模式有效 0x0：保留 0x1：EXMC_CLK周期=2个HCLK周期 …… 0xF：EXMC_CLK周期=16个HCLK周期
19:16	BUSLAT[3:0]	总线延迟时间 在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最小时间 0x0：总线延迟=1个HCLK周期 0x1：总线延迟=2个HCLK周期 …… 0xF：总线延迟=16个HCLK周期
15:8	DSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=2个HCLK周期 …… 0xFF：数据建立时间=256个HCLK周期
7:4	AHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在复用模式有效 0x0：保留 0x1：地址建立时间=2个HCLK

		…… 0xF：地址建立时间=16个HCLK
3:0	ASET[3:0]	异步地址建立时间 该位域设置地址建立时间 **注意：**该位域仅在NOR Flash的异步模式有效 0x0：地址建立时间= 1个HCLK …… 0xF：地址建立时间= 16个HCLK

28.4.2. NAND Flash 控制器寄存器

NAND Flash 控制寄存器 (EXMC_NCTL)

地址偏移：0x60
复位值：0x0000 0018

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												ECCSZ[2:0]			ATR[3]
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATR[2:0]			CTR[3:0]				保留		ECCEN	NDW[1:0]		NDTP	NDBKEN	NDWTEN	保留
rw			rw						rw	rw		rw	rw	rw	

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:17	ECCSZ[2:0]	ECC块大小 000：256字节 001：512字节 010：1024字节 011：2048字节 100：4096字节 101：8192字节
16:13	ATR[3:0]	ALE至RE的延迟 0x0：1个HCLK …… 0xF：16个HCLK
12:9	CTR[3:0]	CLE至RE的延迟 0x0：1个HCLK 0x1：2个HCLK

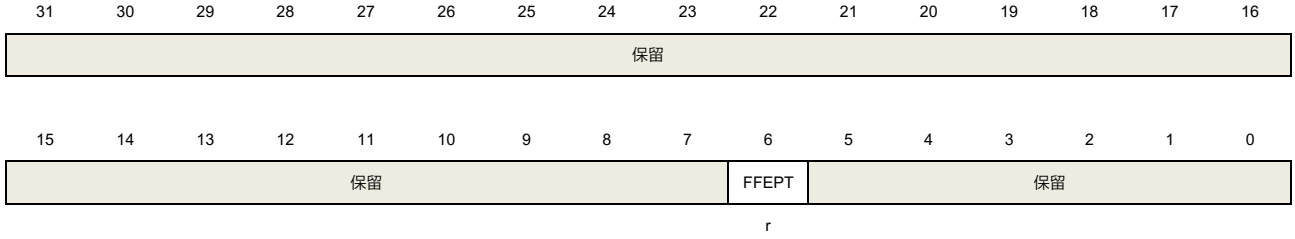
		…… 0xF：16个HCLK
8:7	保留	必须保持复位值。
6	ECCEN	ECC使能 0：关闭ECC，并复位EXMC_NECC寄存器。 1：打开ECC
5:4	NDW[1:0]	外部存储器数据宽度 00：8位 01：16位 其他：保留
3	NDTP	外部存储器的类型 0：保留 1：NAND Flash
2	NDBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块
1	NDWTEN	NWAIT信号使能位 0：关闭等待功能 1：使能等待功能
0	保留	必须保持复位值。

NAND Flash 状态寄存器 (EXMC_NSTAT)

地址偏移：0x64
复位值：0x0000 0042(对于 bank1)

该寄存器包含一个 FIFO 空状态位，该位主要用于 ECC。当写外部存储器时，FIFO 可以容纳来自 AHB 访问的 2 个字，使得 AHB 总线可以暂时被释放而用于其他外设。ECC 计算是基于从 FIFO 传递的数据。为了得到正确的 ECC 值，用户应该在 FIFO 空状态标志位为 1 时才去读 ECC 寄存器。

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	FFEPT	FIFO空标志位 0：FIFO非空 1：FIFO空
5:0	保留	必须保持复位值。

NAND Flash 通用空间时序寄存器 (EXMC_NCTCFG)

地址偏移：0x68
复位值：0xFCFC FCFC

这些操作适用于以下类型的外部存储器的通用存储空间的 NAND Flash。

该寄存器只能按字(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 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留
23:16	COMHLD[7:0]	通用空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：1个HCLK …… 0xFE：254个HCLK 0xFF：保留
15:8	COMWAIT[7:0]	通用空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2个HCLK (加上NWAIT时钟周期) …… 0xFE：255个HCLK (加上NWAIT时钟周期)

		0xFF：保留
7:0	COMSET[7:0]	通用空间的建立时间 定义地址信号的建立时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留

NAND Flash 属性空间时序寄存器 (EXMC_NATCFG)

地址偏移：0x6C
复位值：0xFCFC FCFC

这些参数适用于在 NAND 闪存的最后一个地址写操作的时序与其它操作不同的时候(可参考 ***NAND Flash 预等待功能***)。

该寄存器只能按字(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 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留
23:16	ATTHLD[7:0]	属性空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：1个HCLK …… 0xFE：254个HCLK 0xFF：保留
15:8	ATTWAIT[7:0]	属性空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2个HCLK (加上NWAIT时钟周期)

		…… 0xFE：255个HCLK (加上NWAIT时钟周期) 0xFF：保留
7:0	ATTSET[7:0]	属性空间的建立时间 定义地址信号的建立时间 0x00：1个HCLK …… 0xFE：255个HCLK 0xFF：保留

NAND Flash ECC 结果寄存器 (EXMC_NECC)

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
ECC[31:16]															
r															

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

位/位域	名称	描述
31:0	ECC[31:0]	ECC计算结果

ECCSZ[2:0]	NAND Flash 页大小 (字节)	ECC 位
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

29. 控制器局域网络(CAN)

29.1. 简介

CAN(Controller Area Network)总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。

CAN 网络接口遵循 CAN 总线协议 2.0A、2.0B 规范，还遵循 ISO11898-1:2015 和 BOSCH CAN-FD 规范。CAN 总线控制器可以处理总线上的数据收发，CAN0 和 CAN1 共享 28 个过滤器,而 CAN2 拥有 14 个独立的过滤器。用户可以通过 3 个发送邮箱将待发送数据传输至总线，邮箱发送的顺序由发送调度器决定。并通过 2 个深度为 3 的接收 FIFO 获取总线上的数据，接收 FIFO 的管理完全由硬件控制。同时 CAN 总线控制器硬件支持时间触发通信(Time-trigger communication)功能。

29.2. 主要特征

- 支持 CAN 总线协议 2.0A 和 2.0B；
- 常规帧：通信波特率最大为 1Mbit/s；
- 支持时间触发通信(Time-triggered communication)；
- 中断使能和清除；
- 支持最多 64 个数据字节的 CAN FD 帧(ISO11898-1 和博世 CAN FD 规范 V1.0)
- CAN-FD 帧：通信波特率最大为 8Mbit/s；
- 支持传输延迟补偿。

发送功能

- 3 个发送邮箱；
- 支持发送优先级；
- 支持发送时间戳。

接收功能

- 2 个深度为 3 的接收 FIFO；
- CAN0 和 CAN1 具有 28 个标识符过滤器；
- CAN2 拥有 14 个独立的标识符过滤器；
- FIFO 锁定功能。

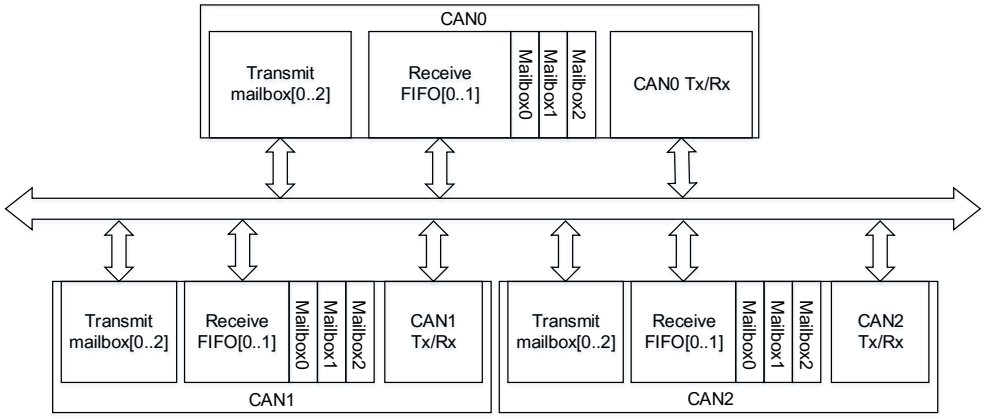
时间触发通信

- 在时间触发通信模式下禁用自动重传；
- 16 位定时器；
- 接收时间戳；
- 发送时间戳。

29.3. 功能说明

CAN 模块结构框图如*图 29-1. CAN 模块结构框图*所示。

图 29-1. CAN 模块结构框图



29.3.1. 工作模式

CAN 总线控制器有 3 种工作模式：

- 睡眠工作模式；
- 初始化工作模式；
- 正常工作模式。

睡眠工作模式

芯片复位后，CAN 总线控制器处于睡眠工作模式。该模式下 CAN 总线控制器的时钟停止工作并处于一种低功耗状态。

将 CAN_CTL 寄存器的 SLPWMOD 位置 1，可以使 CAN 总线控制器进入睡眠工作模式。当进入睡眠工作模式后，CAN_STAT 寄存器的 SLPWS 位将被硬件置 1。

将 CAN_CTL 寄存器的 AWU 位置 1，并当 CAN 检测到总线活动时，CAN 总线控制器将自动退出睡眠工作模式。将 CAN_CTL 寄存器的 SLPWMOD 位清 0，也可以退出睡眠工作模式。

由睡眠模式进入初始化工作模式：将 CAN_CTL 寄存器的 IWMOD 位置 1，SLPWMOD 位清 0。

由睡眠模式进入正常工作模式：将 CAN_CTL 寄存器的 IWMOD 位和 SLPWMOD 位清 0。

初始化工作模式

如果需要配置 CAN 总线通信参数，CAN 总线控制器必须进入初始化工作模式。

将 CAN_CTL 寄存器的 IWMOD 位置 1，使 CAN 总线控制器进入初始化工作模式，将其清 0 则离开初始化工作模式。在进入初始化工作模式后，CAN_STAT 寄存器的 IWS 位将被硬件置 1。

由初始化模式进入睡眠模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，IWMOD 位清 0。

由初始化模式进入正常工作模式：CAN_CTL 寄存器的 SLPWMOD 位和 IWMOD 位清 0。

正常工作模式

在初始化工作模式中配置完 CAN 总线通信参数后，将 CAN_CTL 寄存器的 IWMOD 位和 SLPWMOD 位清 0 可以进入正常工作模式并与 CAN 总线网络中的节点进行正常通信。

由正常工作模式进入睡眠工作模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，并等待当前数据收发过程结束。

由正常工作模式初始化工作模式：CAN_CTL 寄存器的 IWMOD 位置 1，并等待当前数据收发过程结束。

29.3.2. 通信模式

CAN 总线控制器有 4 种通信模式：

- 静默(Silent)通信模式；
- 回环(Loopback)通信模式；
- 回环静默(Loopback and Silent)通信模式；
- 正常(Normal)通信模式。

静默(Silent)通信模式

在静默通信模式下，可以从 CAN 总线接收数据，但不向总线发送任何数据。将 CAN_BT 寄存器中的 SCMOD 位置 1，使 CAN 总线控制器进入静默通信模式，将其清 0 可以退出静默通信模式。

静默通信模式可以用来监控 CAN 网络上的数据传输。

回环(Loopback)通信模式

在回环通信模式下，由 CAN 总线控制器发送的数据可以被自己接收并存入接收 FIFO，同时这些发送数据也送至 CAN 网络。将 CAN_BT 寄存器中的 LCMOD 位置 1，使 CAN 总线控制器进入回环通信模式，将其清 0 可以退出回环通信模式。

回环通信模式通常用来进行 CAN 通信自测。

回环静默(Loopback and Silent)通信模式

在回环静默通信模式下，CAN 的 RX 和 TX 引脚与 CAN 网络断开。CAN 总线控制器既不从 CAN 网络接收数据，也不向 CAN 网络发送数据，其发送的数据仅可以被自己接收。

将 CAN_BT 寄存器中的 LCMOD 位和 SCMOD 位置 1，使 CAN 总线控制器进入回环静默通信模式，将它们清 0 可以退出回环静默通信模式。

回环静默通信模式通常用来进行 CAN 通信自测。对外 TX 引脚保持隐性状态(逻辑 1)，RX 引脚保持高阻态。

正常(Normal)通信模式

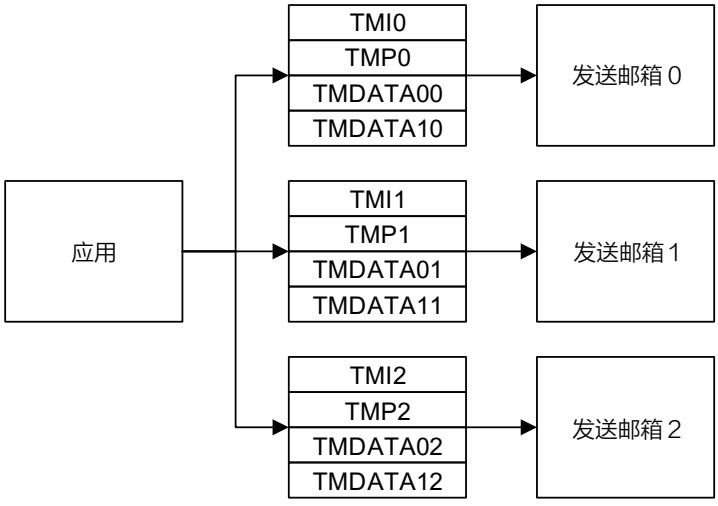
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN_BT 寄存器的 LCMOD 位和 SCMOD 位清 0。

29.3.3. 数据发送

发送寄存器

数据发送通过 3 个发送邮箱进行，可以通过寄存器 CAN_TMIx，CAN_TMPx，CAN_TMDATA0x 和 CAN_TMDATA1x 对发送邮箱进行配置。如*图 29-2. 发送寄存器*所示。

图 29-2. 发送寄存器

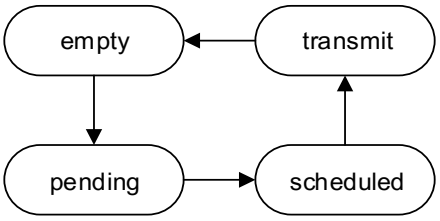


如果想要发送 CAN-FD 帧，使用发送邮箱 x(x 等于 0 到 2)时，仅需要写对应的 TMDATA0x(x 等于 0 到 2)寄存器。例如软件想要使用发送邮箱 0 发送 64 字节数据，需要通过写 TMDATA00 寄存器 16 次将待发送数据写入内部专用 SRAM 区。

发送邮箱状态转换

当发送邮箱处于 **empty** 状态时，应用程序才可以对邮箱进行配置。当邮箱被配置完成后，可以将 CAN_TMIx 寄存器的 TEN 位置 1，从而向 CAN 总线控制器提交发送请求，这时发送邮箱处于 **pending** 状态。当超过 1 个邮箱处于 **pending** 状态时，需要对多个邮箱进行调度，这时发送邮箱处于 **scheduled** 状态。当调度完成后，发送邮箱中的数据开始向 CAN 总线上发送，这时发送邮箱处于 **transmit** 状态。当数据发送完成，邮箱变为空闲，可以再次交给应用程序使用，这时发送邮箱重新变为 **empty** 状态。如*图 29-3. 发送邮箱状态转换*所示。

图 29-3. 发送邮箱状态转换



发送状态和错误信息

CAN_TSTAT 寄存器中的 MTFx，MTFNERRx，MALx 和 MTEx 位用来说明发送状态和错误信息。

- MTFx：发送完成标志位。当数据发送完成时，MTFx 置 1。
- MTFNERRx：无错误发送完成标志位。当数据发送完成且没有错误时，MTFNERRx 置 1。
- MALx：仲裁失败标志位。当发送数据过程中出现仲裁失败时，MALx置1。
- MTEx：发送错误标志位。当发送过程中检测到总线错误时，MTEx置1。

数据发送步骤

数据发送步骤如下：

第 1 步：选择一个空闲发送邮箱；

第 2 步：根据应用程序要求，配置 4 个发送寄存器；

第 3 步：将 CAN_TMIx 寄存器的 TEN 置 1；

第 4 步：检测发送状态和错误信息。典型情况是检测到 MTFx 和 MTFNERRx 置 1，说明数据被成功发送。

发送选项

中止数据发送

将 CAN_TSTAT 寄存器的 MSTx 置 1，可以中止数据发送。

当发送邮箱处于 **pending** 和 **scheduled** 状态，CAN_TSTAT 寄存器的 MSTx 置 1 可以立即中止数据发送。

当发送邮箱处于 **transmit** 状态，则面临两种情况。一种情况是数据发送被成功地完成，MTFx 和 MTFNERRx 为 1，这时发送邮箱将转换为 **empty** 状态。相对的，如果数据发送过程中出现了问题，这时发送邮箱将转换为 **scheduled** 状态，这时数据发送被中止。

发送优先级

当有 2 个及其以上发送邮箱等待发送时，寄存器 CAN_CTL 的 TFO 位的值可以决定发送顺序。

当 TFO 为 1，所有等待发送的邮箱按照先来先发送(FIFO)的顺序进行。

当 TFO 为 0，具有最小标识符(Identifier)的邮箱最先发送。如果所有的标识符(Identifier)相等，具有最小邮箱编号的邮箱最先发送。

29.3.4. 数据接收

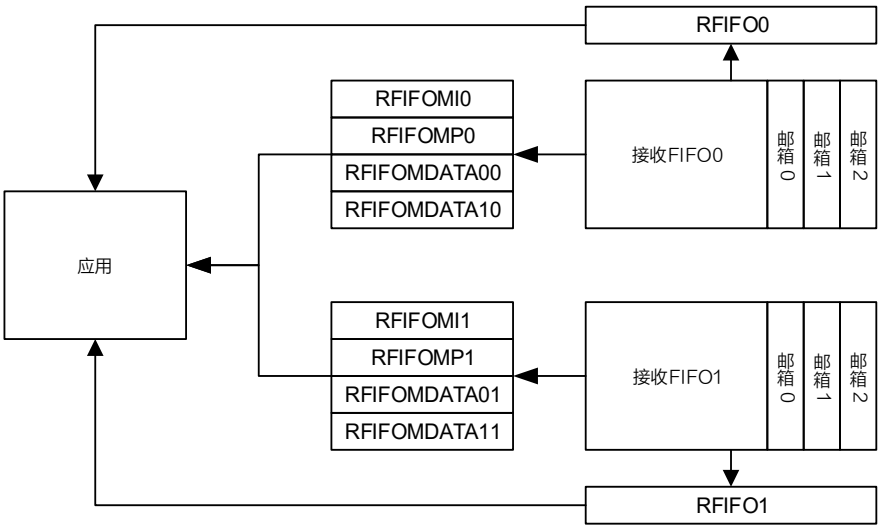
接收寄存器

应用程序通过 2 个深度为 3 的 FIFO 接收来自 CAN 网络的数据。

Rx FIFO 由五个寄存器管理：CAN_RFIFOx，CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x。FIFO 的状态和操作可以通过 CAN_RFIFOx 寄存器来处理。寄存器 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 用于接收数据帧。

如*图 29-4. 接收寄存器*所示。

图 29-4. 接收寄存器



接收 FIFO

每个接收 FIFO 包含 3 个接收邮箱，用来接收存储数据帧。这些邮箱按照先进先出方式进行组织，最早从 CAN 网络接收的数据，最早被应用程序处理。

寄存器 CAN_RFIFOx 包含 FIFO 状态信息和帧的数量。当 FIFO 中包含数据时，可以通过寄存器 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 读取数据，之后将寄存器 CAN_RFIFOx 的 RFD 位置 1 释放邮箱，并且等待其由硬件自动清 0。

如果接收到 CAN-FD 帧，数据将被存储到内部专用 SRAM 中，并通过多次读 CAN_RFIFOMDATA0x 寄存器将数据取出。FIFO0 使用 CAN_RFIFOMDATA00，FIFO1 使用 CAN_RFIFOMDATA01 寄存器。例如，如果软件想要从 FIFO0 中读 64 字节数据，需要通过读 CAN_RFIFOMDATA00 寄存器 16 次将数据全部读出。

接收 FIFO 状态信息

接收 FIFO 状态信息包含在寄存器 CAN_RFIFOx 中。

RFLx[1:0]：FIFO 中包含的帧数量。FIFO 为空时，RFL 为 0；FIFO 为满时，RFL 为 3。

RFFx：FIFO 满状态标志位。这时 RFL 为 3。

RFOx：FIFO 溢出标志位。当 FIFO 已经包含了 3 个数据帧时，新的数据帧到来使 FIFO 发生溢出。如果 CAN_CTL 寄存器的 RFOD 位被置 1，新的数据帧将丢弃。如果该位被清 0，新的数据帧将覆盖接收 FIFO 中最后一帧数据。

当 FIFO 状态过满后又有新的帧到达时，CAN_RFIFOx 寄存器中的 RFOVRNx 位会被设置，表示 FIFO 发生过满，此时 FIFO 中的数据不正确。

数据接收步骤

第 1 步：查看 FIFO 中帧的数量。

第 2 步：通过 CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x 读取数据。

第 3 步：将寄存器 CAN_RFIFOx 的 RFDx 置 1 释放邮箱，并且等待其由硬件自动清 0。

29.3.5. 过滤功能

一个待接收的数据帧会根据其标识符(Identifier)进行过滤：硬件会将通过过滤的帧送至接收 FIFO，并丢弃没有通过过滤的帧。

过滤器位宽

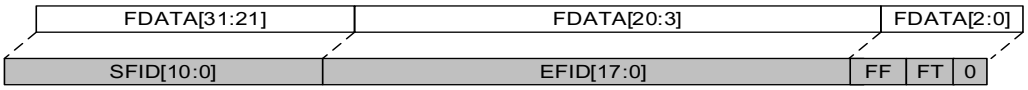
CAN0 和 CAN1 的过滤器包含 28 个单元，它们是 bank0 到 bank27。CAN2 的过滤器包含 14

个独立的单元。

每一个过滤器单元有 2 个寄存器 CAN_FxDATA0 和 CAN_ FxDATA1，它们可以配置为 2 种位宽：32-bit 位宽和 16-bit 位宽。

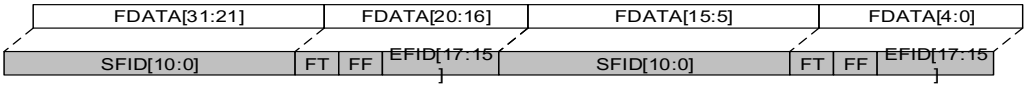
32-bit 位宽：CAN_FxDATAy 包含字段 SFID[10:0]，EFID[17:0]，FF 和 FT。如*图 29-5. 32-bit 位宽过滤器*所示。

图 29-5. 32-bit 位宽过滤器



16-bit 位宽：CAN_FxDATAy 包含字段：SFID[10:0]，FT，FF 和 EFID[17:15]。如*图 29-6. 16-bit 位宽过滤器*所示。

图 29-6. 16-bit 位宽过滤器



掩码模式

对于一个待过滤的数据帧的标识符(Identifier)，掩码模式用来指定哪些位必须与预设的标识符相同，哪些位无需判断。

一个 32-bit 位宽掩码模式过滤器如*图 29-7. 32-bit 位宽掩码模式过滤器*所示。

图 29-7. 32-bit 位宽掩码模式过滤器

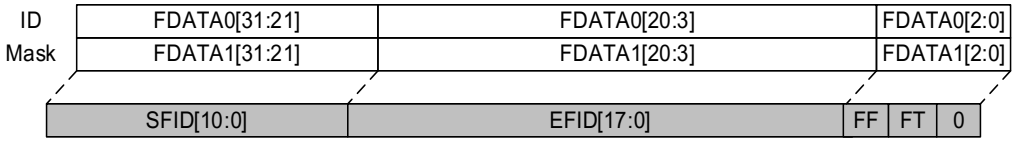
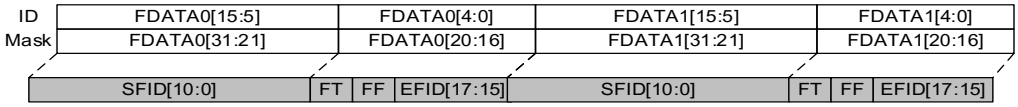


图 29-8. 16-bit 位宽掩码模式过滤器



列表模式

对于一个待过滤的数据帧的标识符(Identifier)，列表模式用来表示与预设的标识符列表中能够匹配则通过，否则丢弃。

一个 32-bit 位宽列表模式过滤器如*图 29-9. 32-bit 位宽列表模式过滤器*所示。

图 29-9. 32-bit 位宽列表模式过滤器

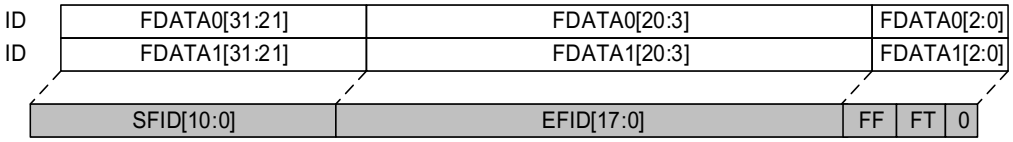
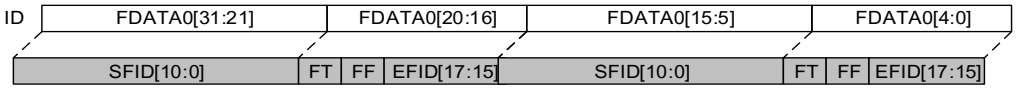


图 29-10. 16-bit 位宽列表模式过滤器



过滤序号

过滤器由若干过滤单元(Bank)组成，每个过滤单元因为位宽和模式的选择不同，而具有不同的过滤效果。例如*表 29-1. 32-bit 过滤序号*所示的 2 个过滤单元，Bank0 是 32-bit 位宽掩码模式，Bank1 是 32-bit 位宽列表模式。

表 29-1. 32-bit 过滤序号

过滤单元	过滤器数据寄存器	过滤序号
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1
	F1DATA1-32bit-ID	2

过滤器关联的 FIFO

28 个过滤单元均可以关联接收 FIFO0 或接收 FIFO1。一旦一个过滤单元关联到接收 FIFO，只有通过这个过滤单元的帧才会被传送到接收 FIFO 中存储。

过滤器激活控制

一个过滤单元如果被应用程序用到，就必须激活。通过 CAN_FW 寄存器可以进行配置。

过滤索引

一个包含过滤序号(Fiter Number)N 的过滤单元通过了某个帧，则该帧数据的过滤索引(Filtering Index)为 N。这时 CAN_RFIFOMPx 中 FI 的值为 N。

在*表 29-2. 过滤索引*中，如果一个帧通过了 FIFO0 中过滤序号 10(Filter Number=10)的过滤单元，那么该帧的过滤索引为 10。这时 CAN_RFIFOMPx 中 FI 的值为 10。

过滤序号不关心对应的过滤单元(Bank)是否处于工作状态。例如 Bank3 被关联到 FIFO0，且为"不激活"状态，但它仍然包含过滤序号 3 和 4。

表 29-2. 过滤索引

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
0	F0DATA0-32bits-ID	是	0	2	F2DATA0[15:0]-16bits-ID	是	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	是	1		F2DATA1[15:0]-16bits-ID		1
	F1DATA1-32bits-ID		2		F2DATA1[31:16]-16bits-Mask		
3	F3DATA0[15:0]-16bits-ID	否	3	4	F4DATA0-32bits-ID	否	2
	F3DATA0[31:16]-16bits-Mask				F4DATA1-32bits-Mask		
	F3DATA1[15:0]-16bits-ID		4	5	F5DATA0-32bits-ID	否	3
	F3DATA1[31:16]-16bits-Mask				F5DATA1-32bits-ID		4
7	F7DATA0[15:0]-16bits-ID	否	5	6	F6DATA0[15:0]-16bits-ID	是	5
	F7DATA0[31:16]-16bits- ID		6		F6DATA0[31:16]-16bits- ID		6
	F7DATA1[15:0]-16bits-ID		7		F6DATA1[15:0]-16bits-ID		7
	F7DATA1[31:16]-16bits- ID		8		F6DATA1[31:16]-16bits- ID		8
8	F8DATA0[15:0]-16bits-ID	是	9	10	F10DATA0[15:0]-16bits-ID	否	9
	F8DATA0[31:16]-16bits- ID		10		F10DATA0[31:16]-16bits-Mask		
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID		10
	F8DATA1[31:16]-16bits- ID		12		F10DATA1[31:16]-16bits-Mask		
9	F9DATA0[15:0]-16bits-ID	是	13	11	F11DATA0[15:0]-16bits-ID	否	11
	F9DATA0[31:16]-16bits-Mask				F11DATA0[31:16]-16bits-ID		12
	F9DATA1[15:0]-16bits-ID		14		F11DATA1[15:0]-16bits-ID		13
	F9DATA1[31:16]-16bits-Mask				F11DATA1[31:16]-16bits-ID		14
12	F12DATA0-32bits-ID	是	15	13	F13DATA0-32bits-ID	是	15

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
	F12DATA1-32bits-Mask				F13DATA1-32bits- ID		16

优先级

过滤器优先级规则如下：

1、32-bits 位宽模式高于 16-bits 位宽模式；

2、列表模式高于掩码模式；

3、较小的过滤序号(Filter Number)具有较高的优先级。

29.3.6. 时间触发通信

时间触发通信是 CAN 数据链路层应用协议。CAN 网络中的所有节点都按照一个预先设定的时间序列进行通信，尤其适合于时间周期性应用和时间确定性应用。

在这种通信模式下，一个内部的 16-bit 计数器开始工作，在每一个 CAN 位时间(Bit time)增 1。这个内部计数器为数据发送和数据接收提供时间戳，这些时间戳存放在寄存器 CAN_RFIFOMPx 和 CAN_TMPx 中。

在这种通信模式下，自动重发功能是禁止的。

29.3.7. 通信参数

自动重发禁止模式

在时间触发通信模式下，要求自动重发必须是禁止的，可以通过将 CAN_CTL 寄存器的 ARD 位置 1 满足要求。

在这种模式下，数据只会被发送一次，如果因为仲裁失败或者总线错误而导致发送失败，CAN 总线控制器不会像通常那样进行数据自动重发。

发送结束时，寄存器 CAN_TSTAT 的 MTFx 位被硬件置 1，而发送状态信息可以通过 MTFNERRx，MALx 和 MTEx 获得。

位时序(Bit time)

CAN 协议采用位同步传输方式。这种方式不仅增大了传输容量，而且意味着需要一种复杂的位同步方法。面向字节传输的位同步方式适用于接收在每个字节前都有起始位的情况，而同步传输协议只要求数据帧的最开始有一个起始位。为保证接收器能正确读取信息，需要不断地进行重新同步。因此，在相位缓冲段采样点前面和后面都应该插入一个帧间隔。

可以通过位操作仲裁方式访问 CAN 总线。信号从发送器到接收器，再回到发送器必须在一个位时间内完成。为了达到同步的目的，除了相位缓冲段外，还需要一个传输延时段。在信号传输过程中，传输延时段被视为发送或接收延时。

CAN 总线控制器将位时间分为 3 个部分。

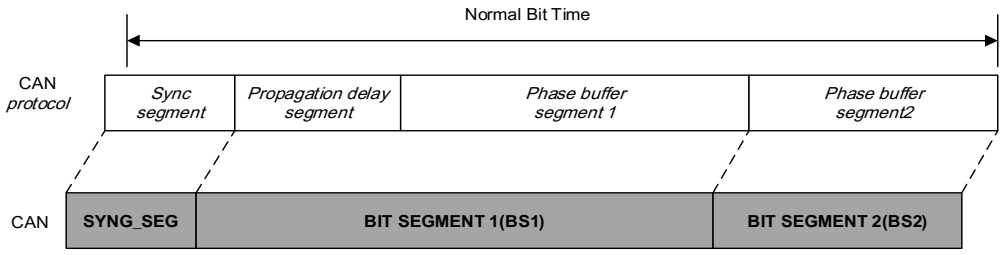
同步段(Synchronization segment)，记为 SYNC_SEG。期望在该段检测到有效跳变沿.该段占用 1 个时间单元($1 \times t_q$)。

位段 1(Bit segment 1)，记为 BS1。该段占用 1 到 16 个时间单元。相对于 CAN 协议而言，BS1 相当于传播时间段(Propagation delay segment)和相位缓冲段 1(Phase buffer segment 1)。

位段 2(Bit segment 2)，记为 BS2。该段占用 1 到 8 个时间单元。相对于 CAN 协议而言，BS2 相当于相位缓冲段 2(Phase buffer segment 2)。

对比与 CAN 协议，位时序如*图 29-11. 位时序*所示。

图 29-11. 位时序



再同步补偿宽度 SJW(resynchronization Jump Width)：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_DBT 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。

如果有效跳变在 BS1 期间被检测到，而不是 SYNC_SEG 期间，BS1 将会最多被延长 SJW，因此采样点延时。

相反，如果有效跳变在 BS2 期间被检测到，而不是 SYNC_SEG 期间，BS2 将会最多被缩短 SJW，因此采样点提前。

注意：有关硬同步和再同步的详细说明，请参考 ISO 11898 标准。

波特率

波特率计算公式如下：

$$BaudRate = \frac{1}{Normal\ Bit\ Time} \quad (29\text{-}1)$$

$$Normal\ Bit\ Time = t_{SYNC_SEG} + t_{BS1} + t_{BS2} \quad (29\text{-}2)$$

其中：

$$t_{SYNC_SEG}=1\times t_q \quad (29\text{-}3)$$

$$t_{BS1}=(1+BT.BS1)\times t_q \quad (29\text{-}4)$$

$$t_{BS2}=(1+BT.BS2)\times t_q \quad (29\text{-}5)$$

$$t_q=(1+BT.BAUDPSC)\times t_{PCLK2} \quad (29\text{-}6)$$

29.3.8. CAN-FD 操作

通过将 CAN_FDCTL 寄存器的 FDEN 位置 1，可以使能 CAN FD (CAN with Flexible Data rate)功能。如果 FDEN 位被清 0，CAN 总线控制器仅支持常规帧(标准帧和扩展帧)的收发，若 FDEN 位被置 1，则 CAN 总线控制器同时支持常规帧(标准帧和扩展帧)以及 FD 帧的收发。根据协议，当前帧是否为 FD 帧是通过帧的 FDF 位来判断(在常规帧中该位为保留位)。如果 FDF 位为隐性，表示是 CAN FD 帧；如果为显性，表示是常规帧。

通过配置 CAN_FDCTL 寄存器的 NISO 位，可以选择 CAN-FD 功能支持 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

在 CAN-FD 帧的帧结构中，FDF 位之后是保留位和 BRS 位。BRS 位决定数据位速率，当 BRS 位为显性时，表示不能通过配置 CAN_DBT 寄存器来切换数据位速率。当 BRS 位为隐性时，可以通过配置 CAN_DBT 寄存器使得数据段的位速率高于仲裁段的位速率。数据阶段应从检测到 BRS 位为隐性时的采样点开始，在到达 CRC 分隔符的第一个采样点时结束，或者当 CAN 节点检测到导致错误帧启动的错误条件时结束。数据的比特率可以在 BRS 位和 CRC 分隔符位的采样点处切换。(详情请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0)。

通过将 CAN_FDCTL 寄存器的 PRED 位清 0，可以使能协议异常处理功能。此时，在接收帧数据过程中检测到隐性的保留位时，该功能将使操作状态转变为 IDLE 并在下一个采样点中止当前帧。反之，将 PRED 位置 1，该功能将被禁止，隐性的保留位将被视为格式错误，并当做错误帧来进行处理，同时 CAN_FDSTAT 寄存器的 PRE 位将被置 1。

ISO11898-1 或 BOSCH CAN FD 规范 V1.0 规定的发送 ESI 位(该位位于 CAN FD 帧的 DLC 位域之前)功能通过 CAN_FDCTL 寄存器的 ESIMOD 位和 CAN_TMPx 寄存器的 ESI 位来实现。如果将 ESIMOD 位清 0，当 CAN 总线控制器处于被动错误状态时，该位为隐性；当处于主动错误状态时，该位为显性。若将 ESIMOD 位置 1，将根据 CAN_TMPx 寄存器的 ESI 位的值来决定该位为显性还是隐性。

发送帧 FDF 位和 BRS 位的总线电平逻辑由 CAN_TMPx 寄存器的 FDF 位和 BRS 位的值决定。

29.3.9. 传输延迟补偿

CAN-FD 协议支持传输延迟补偿机制。由于 CAN 收发器存在回路延迟，因此当发送 CAN-FD 帧的高速数据段的位时间长度小于收发器内部回路延迟的限定值时，该机制可以避免当采样点到来时发送节点还没有收到其自己发出的位，从而报错的情况发生。关于传输延迟补偿的具体

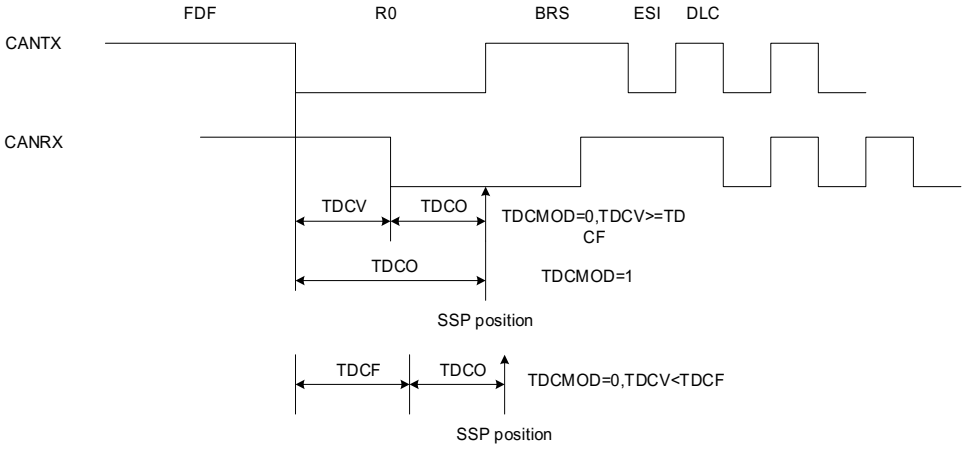
定义，请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

将 CAN_FDCTL 寄存器的 TDCEN 位置 1 将使能传输延迟补偿。

传输延迟补偿机制可以调节次级采样点(SSP)的位置。SSP_Delay 被定义为 CANTX 上的信号到 SSP 采样点的延迟。如果 CAN_FDCTL 寄存器的 TDCMOD 位被置 1，SSP_Delay 的值由 CAN_FDTDC 寄存器的 TDCO[7:0]位域软件配置决定。如果 TDCMOD 位被清 0，硬件将自动计算位速率转换之前的 FDF 位到 RES0 位的下降沿在 CAN_TX 与 CAN_RX 上出现的延迟，并将计算值存入 CAN_FDSTAT 寄存器的 TDCV[6:0]位域。由于存在信号毛刺，可能导致硬件自动计算的 SSP 位置比预期的提前，为了避免 TDCV[6:0]的值过小，可以使用 CAN_FDTDC 寄存器的 TDCF[7:0]位域。如果 TDCV[6:0]的值大于 TDCF[7:0]，SSP_Delay 的值被定义为 TDCO[7:0]加上 TDCV[6:0]，否则 SSP_Delay 的值被定义为 TDCO[7:0]加上 TDCF[7:0]。

SSP_Delay 的值不能大于 3 个数据位时间。

图 29-12 传输延迟测量



29.3.10. 错误标志

CAN 总线的状态可以通过 CAN_ERR 寄存器的发送错误计数值(Transmit Error Counter，TECNT[7:0])和接收错误计数值(Receive Error Counter，RECNT[7:0])反映，其值会根据错误的情况由硬件增加或减少，软件可以通过这些值判断 CAN 网络的稳定性。关于错误计数值的详细信息请参考 CAN 协议相关章节。

通过使能 CAN_INTEN 寄存器中的相应位(ERRIE 等)，软件可以在检测到错误时产生相应中断。

离线恢复

当 TECNT[7:0]大于 255 时，CAN 总线控制器进入离线状态，这时寄存器 CAN_ERR 中的 BOERR 置 1，并且发送和接收失效。

根据寄存器 CAN_CTL 中的 ABOR 配置，离线恢复(变为主动错误状态)有 2 种方式。这两种方式都要求处于离线状态的 CAN 总线控制器检测到 CAN 协议所定义的离线恢复序列(在

CAN_RX 检测到 128 次连续 11 个位的隐性位)时，才会自动恢复。

如果 ABOR 为 1，将在检测到离线恢复序列后自动恢复。

如果 ABOR 为 0，则必须先将 CAN_CTL 中的 IWMOD 置 1 进入初始化工作模式，然后进入正常工作模式并在检测到离线恢复序列后恢复。

29.3.11. 中断

CAN 总线控制器占用 4 个中断向量，通过寄存器 CAN_INTEN 进行控制。

这 4 个中断向量对应 4 类中断源：

- 发送中断；
- FIFO0 中断；
- FIFO1 中断；
- 错误和状态改变中断。

发送中断

发送中断包括：

- 寄存器CAN_TSTAT中的MTF0置1：发送邮箱0变为空闲。
- 寄存器CAN_TSTAT中的MTF1置1：发送邮箱1变为空闲。
- 寄存器CAN_TSTAT中的MTF2置1：发送邮箱2变为空闲。

FIFO0 中断

FIFO0 中断包括：

- FIFO0中包含待接收数据：寄存器CAN_RFIFO0中的RFL0不为0，CAN_INTEN寄存器中RFNEIE0被置位；
- FIFO0满：寄存器CAN_RFIFO0中的RFF0为1，CAN_INTEN寄存器中RFFIE0被置位；
- FIFO0溢出：寄存器CAN_RFIFO0中的RFO0为1，CAN_INTEN寄存器中RFOIE0被置位。

FIFO1 中断

FIFO1 中断包括：

- FIFO1中包含待接收数据：寄存器CAN_RFIFO1中的RFL1不为0，CAN_INTEN寄存器中RFNEIE1被置位；
- FIFO1满：寄存器CAN_RFIFO1中的RFF1为1，CAN_INTEN寄存器中RFFIE1被置位；
- FIFO1溢出：寄存器CAN_RFIFO1中的RFO1为1，CAN_INTEN寄存器中RFOIE1被置位。

错误和工作模式改变(EWMC)中断

错误和工作模式改变中断可由以下条件触发：

- 错误：CAN_STAT寄存器的ERRIF和CAN_INTEN寄存器的ERRIE被置位，请参考CAN_STAT寄存器中ERRIF位描述；
- 唤醒：CAN_STAT寄存器中的WUIF和CAN_INTEN寄存器的WIE被置位；
- 进入睡眠模式：CAN_STAT寄存器中的SLPIF和CAN_INTEN寄存器的SLPWIE被置

CAN 总线控制器的中断产生条件可参考*表 29-3. CAN 事件/中断标志*。

表 29-3. CAN 事件/中断标志

中断事件	事件/中断标志		使能控制位	
发送中断	发送邮箱 0 空闲标志 MTF0		TMEIE	
	发送邮箱 1 空闲标志 MTF1			
	发送邮箱 2 空闲标志 MTF2			
FIFO0 中断	接收 FIFO0 中帧的数量 RFL0[1:0]		RFNEIE0	
	接收 FIFO0 满 RFF0		RFFIE0	
	接收 FIFO0 溢出 RFO0		RFOIE0	
FIFO1 中断	接收 FIFO1 中帧的数量 RFL1[1:0]		RFNEIE1	
	接收 FIFO1 满 RFF1		RFFIE1	
	接收 FIFO1 溢出 RFO1		RFOIE1	
EWMC 中断	警告错误 WERR	错误中断标志 ERRIF	WERRIE	ERRIE
	被动错误 PERR		PERRIE	
	离线错误 BOERR		BOIE	
	错误种类 1<= ERRN[2:0] <= 6		ERRNIE	
	从睡眠工作模式唤醒的状态改变中断标志WUIF		WIE	
	进入睡眠工作模式的状态改变中断标志SLPIF		SLPWIE	

29.4. CAN 寄存器

CAN0 基地址：0x4001 1C00

CAN1 基地址：0x4001 2000

CAN2 基地址：0x4001 7400

29.4.1. 控制寄存器(CAN_CTL)

地址偏移：0x00
复位值：0x0001 0002

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	DFZ	调试冻结 如果DBG_CTL寄存器中CANx_HOLD被置位，该位用来定义CAN控制器工作在调试冻结或正常工作状态。如果DBG_CTL寄存器中CANx_HOLD被清零，该位无效。 0：处于Debug时，CAN接收和发送正常工作 1：处于Debug时，CAN接收和发送停止
15	SWRST	软件复位 0：正常操作 1：复位CAN并进入睡眠工作模式。该位会自动清0。
14:8	保留	必须保持复位值。
7	TTC	时间触发通信 0：禁用时间触发通信 1：使能时间触发通信
6	ABOR	自动离线恢复 0：通过软件手动地从离线状态恢复 1：通过硬件自动的从离线状态恢复
5	AWU	自动唤醒 一旦自动唤醒后，CAN_CTL寄存器的SLPWMOD位将自动被清0。

		0：通过软件手动的从睡眠工作模式唤醒 1：通过硬件自动的从睡眠工作模式唤醒
4	ARD	自动重发禁止 0：使能自动重发 1：禁用自动重发
3	RFOD	禁用接收FIFO满时覆盖 0：使能接收FIFO满时覆盖。当接收FIFO满时，FIFO中的数据被新来的数据覆盖 1：禁用接收FIFO满时覆盖。当接收FIFO满时，新来的数据被丢弃，FIFO中的数据保持不变，不会被覆盖
2	TFO	发送FIFO顺序 0：标识符(Identifier)较小的帧先发送 1：所有等待发送的邮箱按照先进先出(FIFO)的顺序发送
1	SLPWMOD	睡眠工作模式 如果软件将该位置1，CAN将会在当前发送或接收完成时进入睡眠工作模式。该位可由软件或者硬件清0。如果CAN_CTL寄存器中AWU被置位，当检测到CAN总线工作时，该位被清0。 0：禁用睡眠工作模式 1：使能睡眠工作模式
0	IWMOD	初始化工作模式 0：禁用初始化工作模式 1：使能初始化工作模式

29.4.2. 状态寄存器(CAN_STAT)

地址偏移：0x04
复位值：0x0000 0C02

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RXL	LASTRX	RS	TS	保留			SLPIF	WUIF	ERRIF	SLPWS	IWS
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	RXL	RX引脚电平
10	LASTRX	RX引脚最近一次的采样值

9	RS	接收状态 0：CAN当前不是接收器 1：CAN当前是接收器
8	TS	发送状态 0：CAN当前不是发送器 1：CAN当前是发送器
7:5	保留	必须保持复位值。
4	SLPIF	进入睡眠工作模式的状态改变中断标志 该位在进入睡眠工作模式时由硬件置位。当CAN不再处于睡眠工作模式时由硬件清零。该位也可以由软件写1清0。 0：CAN没有进入睡眠工作模式 1：CAN进入睡眠工作模式。如果相应的中断使能位为1，则发生中断
3	WUIF	从睡眠工作模式唤醒的状态改变中断标志 该位在睡眠工作模式时检测到CAN总线上的活动时由硬件置位。该位由软件写1清0。 0：没有检测到唤醒信号 1：发现唤醒信号。如果相应的中断使能位为1，则发生中断。
2	ERRIF	错误中断标志 该位由以下事件置位。CAN_ERR寄存器中BOERR位和CAN_INTEN寄存器中BOIE位都置位。或CAN_ERR寄存器中PERR位和CAN_INTEN寄存器中PERRIE位都置位。或CAN_ERR寄存器中WERR位和CAN_INTEN寄存器中WERRIE位都置位。或CAN_ERR寄存器中ERRN位域的值不为0且CAN_INTEN寄存器中ERRNIE位置位。该位由软件写1清零。 0：没有错误 1：发生错误。如果相应的中断使能位为1，则发生中断
1	SLPWS	睡眠工作状态 将CAN_CTL寄存器中SLPWMOD位置位进入睡眠工作模式后该位由硬件置位。当CAN由正常通信模式切换到睡眠工作模式，需等待当前发送过程或者接收过程完成。当CAN离开睡眠工作模式(清除CAN_CTL寄存器中SLPWMOD位或是在CAN_CTL寄存器中AWU置位时检测到CAN总线上的活动)时，该位由硬件清零。如果由睡眠工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。 0：CAN没有处于睡眠工作状态 1：CAN处于睡眠工作状态
0	IWS	初始化工作状态 将CAN_CTL寄存器中IWMOD位置位进入初始化工作模式后该位由硬件置位。当CAN由正常通信模式切换到初始化工作模式，需等待当前发送过程或者接收过程完成。在清除CAN_CTL寄存器中IWMOD位离开初始化模式后，该位由硬件清0。如果由初始化工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。 0：CAN没有处于初始化工作状态

1：CAN处于初始化工作状态

29.4.3. 发送状态寄存器(CAN_TSTAT)

地址偏移：0x08

复位值：0x1C00 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]		MST2	保留			MTE2	MAL2	MTFNERR2	MTF2
r	r	r	r	r	r	r		rs				rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	保留			MTE1	MAL1	MTFNERR1	MTF1	MST0	保留			MTE0	MAL0	MTFNERR0	MTF0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31	TMLS2	在发送FIFO中邮箱2最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱2具有最后的发送顺序。
30	TMLS1	在发送FIFO中邮箱1最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱1具有最后的发送顺序。
29	TMLS0	在发送FIFO中邮箱0最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱0具有最后的发送顺序。
28	TME2	发送邮箱2空 0：发送邮箱2不为空 1：发送邮箱2空
27	TME1	发送邮箱1空 0：发送邮箱1不为空 1：发送邮箱1空
26	TME0	发送邮箱0空 0：发送邮箱0不为空 1：发送邮箱0空
25:24	NUM[1:0]	当发送FIFO不满时，NUM表示下一个将要发送的邮箱号。 当发送FIFO满时，NUM表示最后一个将要发送的邮箱号。
23	MST2	邮箱2停止发送 将其置1，将停止邮箱2的发送过程。 当邮箱2变为**emtpy**状态时，该位被硬件自动清0。

22:20	保留	必须保持复位值。
19	MTE2	邮箱2发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。 当发生错误时该位被置1。
18	MAL2	邮箱2仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。
17	MTFNERR2	邮箱2无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
16	MTF2	邮箱2发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI2寄存器的TEN被置位时清0。 0：发送邮箱2正在发送 1：发送邮箱2完成发送
15	MST1	邮箱1停止发送 将其置1，将停止邮箱1的发送过程。 当邮箱1变为**emtpy**状态时，该位被硬件自动清0。
14:12	保留	必须保持复位值。
11	MTE1	邮箱1发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。 当发生错误时该位被置1。
10	MAL1	邮箱1仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。 当发生仲裁失败时该位被置1。
9	MTFNERR1	邮箱1无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
8	MTF1	邮箱1发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI1寄存器的TEN被置位时清0。

		0：发送邮箱1正在发送 1：发送邮箱1完成发送
7	MST0	邮箱0停止发送 将其置1，将停止邮箱0的发送过程。 当邮箱0变为**emtpy**状态时，该位被硬件自动清0。
6:4	保留	必须保持复位值。
3	MTE0	邮箱0发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。 当发生错误时该位被置1。
2	MAL0	邮箱0仲裁失败 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。 当发生仲裁失败时该位被置1。
1	MTFNERR0	邮箱0无错发送完成 当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在无错传输结束时由硬件清0。 0：传输结束时发生了错误 1：传输结束且没有错误
0	MTF0	邮箱0发送完成 当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TMI0寄存器的TEN被置位时清0。 0：发送邮箱0正在发送 1：发送邮箱0完成发送

29.4.4. 接收 FIFO0 寄存器(CAN_RFIFO0)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							RFOVRN	保留		RFD0	RFO0	RFF0	保留	RFL0[1:0]	
							0								
							r			rs	rc_w1	rc_w1		r	

位/位域	名称	描述

31:9	保留	必须保持复位值。
8	RFOVRN0	Rx FIFO0过满 当启用Rx FIFO覆盖并且Rx FIFO过满时，该位由硬件设置。 0：Rx FIFO0没有过满 1：Rx FIFO0过满 **注意**：当设置此位时，不要从FIFO读取数据。
7:6	保留	必须保持复位值。
5	RFD0	释放一次FIFO0中的数据 该位被置1，将释放FIFO0中的一帧数据。 FIFO释放相应的数据空间后，该位被清0。
4	RFO0	接收FIFO0溢出 当接收FIFO0溢出时被置位，由软件写1清0。 0：接收FIFO0没有溢出 1：接收FIFO0溢出
3	RFF0	接收FIFO0满 当接收FIFO0满时被置位，由软件写1清0。 0：接收FIFO0不满 1：接收FIFO0满
2	保留	必须保持复位值。
1:0	RFL0[1:0]	接收FIFO0中帧的数量

29.4.5. 接收 FIFO1 寄存器(CAN_RFIFO1)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							RFOVRN1	保留		RFD1	RFO1	RFF1	保留	RFL1[1:0]	
							r			rs	rc_w1	rc_w1		r	

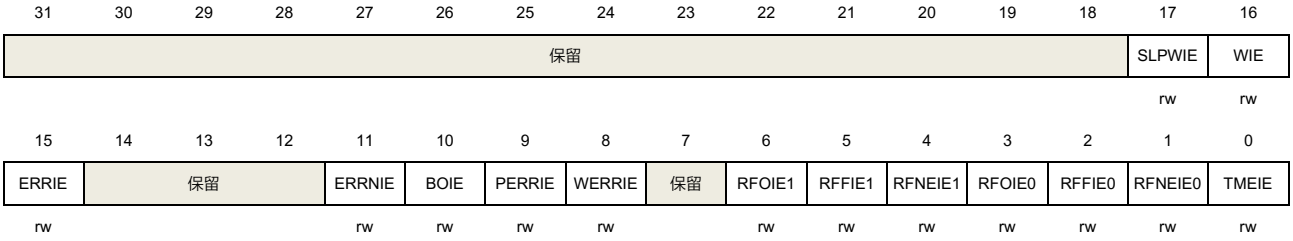
位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RFOVRN1	Rx FIFO1过满 当启用Rx FIFO覆盖并且Rx FIFO过满时，该位由硬件设置。

		0：Rx FIFO1没有过满 1：Rx FIFO1过满 **注意**：当设置此位时，不要从FIFO读取数据。
7:6	保留	必须保持复位值。
5	RFD1	释放一次FIFO1中的数据 该位被置1， 将释放FIFO1中的一帧数据。 FIFO释放相应的数据空间后，该位被清0。
4	RFO1	接收FIFO1溢出 当接收FIFO1溢出时被置位，由软件写1清0。 0：接收FIFO1没有溢出 1：接收FIFO1溢出
3	RFF1	接收FIFO1满 当接收FIFO1满时被置位，由软件写1清0。 0：接收FIFO1不满 1：接收FIFO1满
2	保留	必须保持复位值。
1:0	RFL1[1:0]	接收FIFO1中帧的数量

29.4.6. 中断使能寄存器(CAN_INTEN)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17	SLPWIE	睡眠中断使能 0：禁用睡眠中断 1：使能睡眠中断
16	WIE	唤醒中断使能 0：禁用唤醒中断

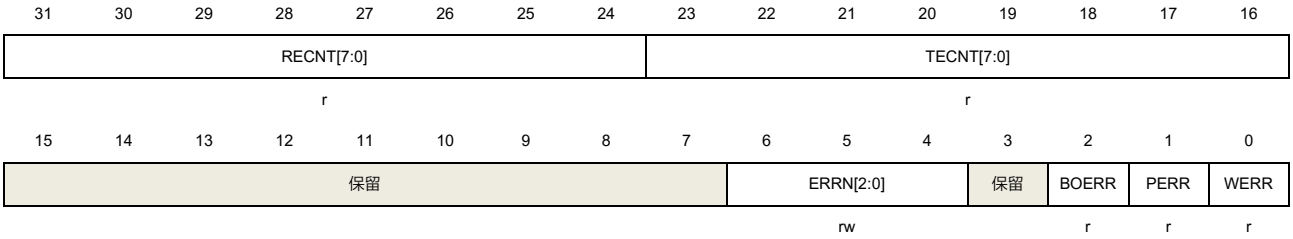
		1：使能唤醒中断
15	ERRIE	错误中断使能 0：禁用错误中断 1：使能错误中断
14:12	保留	必须保持复位值。
11	ERRNIE	错误种类中断使能 0：禁用错误种类中断 1：使能错误种类中断
10	BOIE	离线中断使能 0：禁用离线中断 1：使能离线中断
9	PERRIE	被动错误中断使能 0：禁用被动错误 1：使能被动错误
8	WERRIE	警告错误中断使能 0：禁用警告错误中断 1：使能警告错误中断
7	保留	必须保持复位值。
6	RFOIE1	接收FIFO1溢出中断使能 0：禁用接收FIFO1溢出中断 1：使能接收FIFO1溢出中断
5	RFFIE1	接收FIFO1满中断使能 0：禁用接收FIFO1满中断 1：使能接收FIFO1满中断
4	RFNEIE1	接收FIFO1非空中断使能 0：禁用接收FIFO1非空中断 1：使能接收FIFO1非空中断
3	RFOIE0	接收FIFO0溢出中断使能 0：禁用接收FIFO0溢出中断 1：使能接收FIFO0溢出中断
2	RFFIE0	接收FIFO0满中断使能 0：禁用接收FIFO0满中断 1：使能接收FIFO0满中断
1	RFNEIE0	接收FIFO0非空中断使能 0：禁用接收FIFO0非空中断 1：使能接收FIFO0非空中断

0	TMEIE	发送邮箱空中断使能 0：禁用发送邮箱空中断 1：使能发送邮箱空中断

29.4.7. 错误寄存器(CAN_ERR)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:24	RECNT[7:0]	接收错误计数值
23:16	TECNT[7:0]	发送错误计数值
15:7	保留	必须保持复位值。
6:4	ERRN[2:0]	错误种类 ERRN由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ERRN为0。软件可以设置ERRN为0b111。 000：无错误 001：填充错误 010：格式错误 011：ACK错误 100：位隐性错 101：位显性错误 110：CRC错误 111：软件设置值
3	保留	必须保持复位值。
2	BOERR	离线错误 当TECNT上溢(超过 255)时，CAN总线控制器进入离线状态，该位被置1。
1	PERR	被动错误 当TECNT或者RECNT大于127时，该位由硬件置1。
0	WERR	警告错误 当TECNT或RECNT大于等于96时，该位由硬件置1。

29.4.8. 位时序寄存器(CAN_BT)

地址偏移：0x1C

复位值：0x0123 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCMOD	LCMOD	保留	SJW[4:0]					保留	BS2[2:0]			BS1[3:0]			
rw	rw		rw						rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BS2[4:3]		BS1[6:4]					BAUDPSC[9:0]							
	rw		rw					rw							

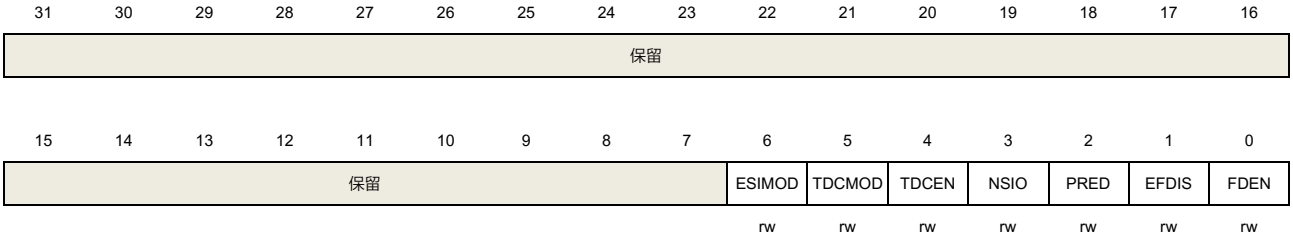
位/位域	名称	描述
31	SCMOD	静默通信模式 0：禁用静默通信模式 1：使能静默通信模式
30	LCMOD	回环通信模式 0：禁用回环通信模式 1：使能回环通信模式
29	保留	必须保持复位值。
28:24	SJW[4:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[4:0]+1
23	保留	必须保持复位值。
22:20	BS2[2:0]	位段2的2:0位 位段2占用的时间单元数量=BS2[4:0]+1
19:16	BS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[6:0]+1
15:10	保留	必须保持复位值。
14:13	BS2[4:3]	位段2的4:3位 参考CAN_BT寄存器的22:20位 当CAN_FDCTL寄存器中的FDEN=1时，此位域才能被修改。
12:10	BS1[6:4]	位段1的6:4位 参考CAN_BT寄存器的19:16位 当CAN_FDCTL寄存器中的FDEN=1时，此位域才能被修改。
9:0	BAUDPSC[9:0]	波特率分频系数

29.4.9. FD 控制寄存器(CAN_FDCTL)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

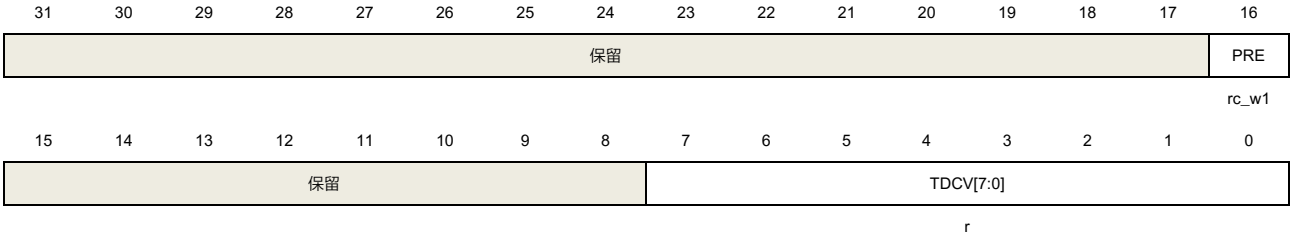


位/位域	名称	描述
31:7	保留	必须保持复位值。
6	ESIMOD	错误状态指示模式 0：发送帧的ESI位始终显示节点错误状态。主动错误状态节点发送显性位，被动错误状态节点发送隐性位 1：当节点处于主动错误状态时，通过CAN_TMPx寄存器的ESI位值决定发送帧的ESI位显示的错误状态。当节点处于被动错误状态时，发送帧该位发送隐性位。
5	TDCMOD	传输延迟补偿模式 0：测量值加软件偏移 1：仅使用软件偏移
4	TDCEN	传输延迟补偿使能 0：传输延迟补偿禁止 1：传输延迟补偿使能
3	NSIO	ISO/BOSCH 0：ISO 1：BOSCH
2	PRED	协议异常事件检测禁用 0：协议异常事件检测使能(当接收到隐性保留位时转为IDLE状态) 1：协议异常事件检测禁用(当接收到隐性保留位时，将此视为帧格式错误)
1	EFDIS	边沿过滤禁能 0：使能边沿过滤 1：禁能边沿过滤
0	FDEN	FD功能使能 0：CAN-FD功能禁止 1：CAN-FD功能使能

29.4.10. FD 状态寄存器(CAN_FDSTAT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

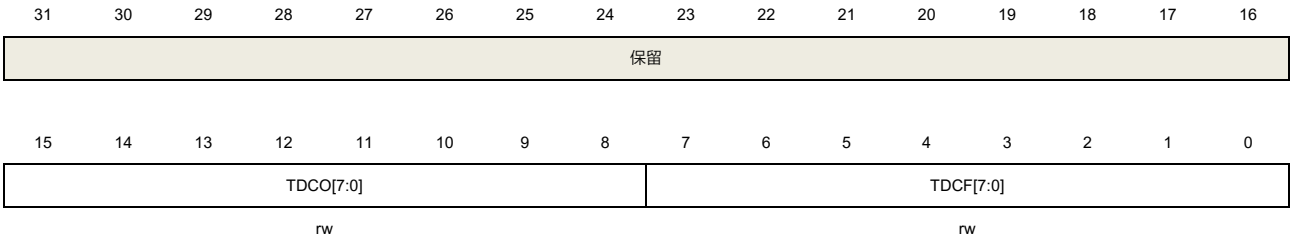


位/位域	名称	描述
31:17	保留	必须保持复位值。
16	PRE	协议异常事件 当协议异常事件被检测到时该位由硬件置1，软件写1可清0该位。
15:8	保留	必须保持复位值。
7:0	TDCV[7:0]	传输延迟补偿值 这些位由硬件控制，显示硬件自动计算的传输延迟补偿值。

29.4.11. FD 传输延迟补偿寄存器(CAN_FDTDC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	TDCO[7:0]	传输延迟补偿偏移 这些位被用于设置次级采样点(SSP)基于测量的补偿时间的偏移，测量的补偿时间是由硬件计算信号从CAN_TX发出到从CAN_RX接收到的过程的延迟时间得出的。
7:0	TDCF[7:0]	传输延迟补偿过滤器 这些位定义了次级采样点(SSP)的最小值。当配置的TDCF值大于TDCV时，这些位的

值用于替代TDCV，以过滤脉冲毛刺引起的测量误差导致的SSP采样点过早的情况发生。

29.4.12. 数据位时序寄存器(CAN_DBT)

地址偏移：0x2C
复位值：0x0123 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	DSJW[2:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[2:0]+1
23	保留	必须保持复位值。
22:20	DBS2[2:0]	位段2的2:0位 位段2占用的时间单元数量=BS2[2:0]+1
19:16	DBS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[3:0]+1
15:10	保留	必须保持复位值。
9:0	DBAUDPSC[9:0]	波特率分频系数

29.4.13. 发送邮箱标识符寄存器(CAN_TMIx)(x = 0…2)

地址偏移：0x180 + 0x10 * x
复位值：0xXXXX XXXX (bit0 = 0)

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]：标准格式帧标识符 EFID[28:18]：扩展格式帧标识符
20:3	EFID[17:0]	标识符 EFID[17:0]：扩展格式帧标识符
2	FF	帧格式 0：标准格式帧 1：扩展格式帧
1	FT	帧种类 0：数据帧 1：遥控帧
0	TEN	发送使能 当应用程序想要发送数据时，该位被置1将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清0。 0：禁用发送 1：使能发送

29.4.14. 发送邮箱属性寄存器(CAN_TMPx)(x = 0...2)

地址偏移：0x184 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:16	TS[15:0]	时间戳 发送时间戳
15:9	保留	必须保持复位值。
8	TSEN	时间戳使能 0：禁用时间戳 1：使能时间戳。时间戳TS[15:0]将放在寄存器CAN_TMDATA1的DATA6和DATA7中

		只有当寄存器CAN_CTL中的TTC为1时，该位才有效。
7	FDF	CAN-FD帧标志位 0：常规帧(标准帧或扩展帧) 1：CAN-FD帧
6	保留	必须保持复位值。
5	BRS	位速率转换开关 0：不转换速率 1：转换可变速率
4	ESI	错误状态指示 仅当CAN_FDCTL寄存器的ESIMOD位为1时，该位有效 0：发送帧中的ESI位将发送显性位 1：发送帧中的ESI位将发送隐性位
3:0	DLENC[3:0]	数据长度，DLENC[3:0]表示帧内数据长度。

29.4.15. 发送邮箱 data0 寄存器(CAN_TMDATA0x)(x = 0...2)

地址偏移：0x188 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB3[7:0]								DB2[7:0]							
rw								rw							

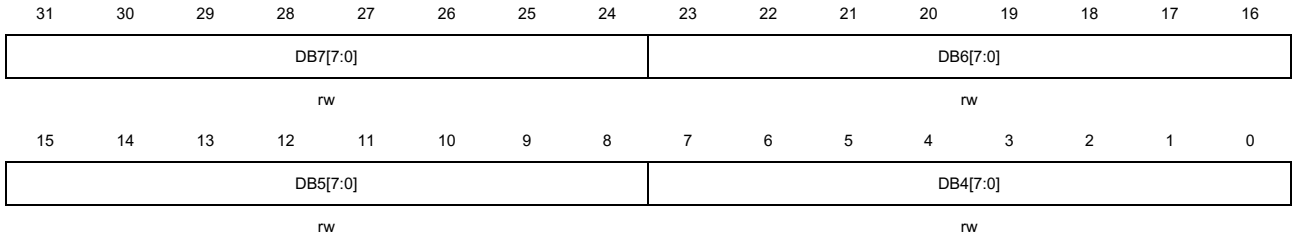
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
rw								rw							

位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

29.4.16. 发送邮箱 data1 寄存器(CAN_TMDATA1x)(x = 0...2)

地址偏移：0x18C + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

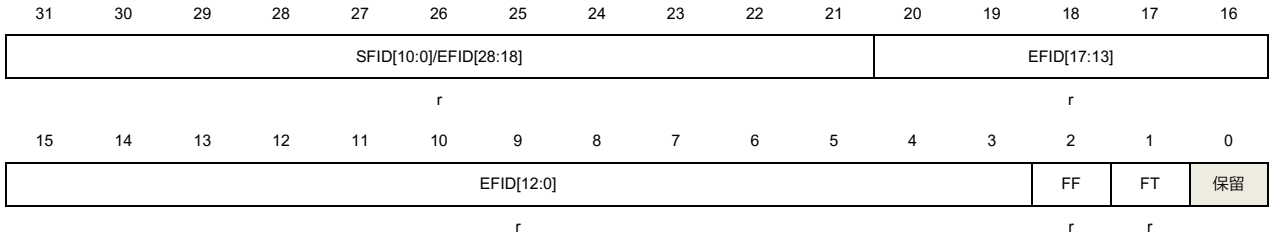


位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

29.4.17. 接收 FIFO 邮箱标识符寄存器(CAN_RFIFOMIx)(x = 0,1)

地址偏移：0x1B0 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。



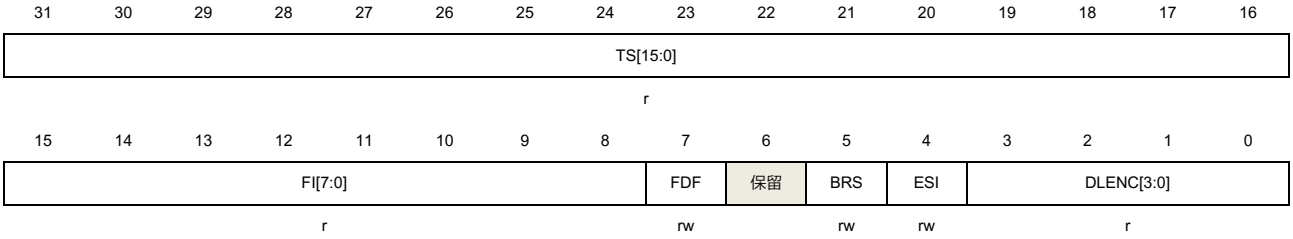
位	区域	说明
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]：标准格式帧标识符 EFID[28:18]：扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]：扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]：扩展格式帧标识符
2	FF	帧格式 0：标准格式帧 1：扩展格式帧
1	FT	帧种类 0：数据帧

		1：遥控帧
0	保留	必须保持复位值。

29.4.18. 接收 FIFO 邮箱属性寄存器(CAN_RFIFOMPx)(x = 0,1)

地址偏移：0x1B4 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	TS[15:0]	时间戳 接收时间戳
15:8	FI[7:0]	过滤索引 帧通过过滤器时的过滤序号
7	FDF	CAN-FD帧标志位 0：常规帧(标准帧或扩展帧) 1：CAN-FD帧
6	保留	必须保持复位值。
5	BRS	位速率转换开关 0：不转换速率 1：转换可变速率
4	ESI	错误状态指示 仅当CAN_FDCTL寄存器的ESIMOD位为1时，该位有效 0：发送帧中的ESI位将发送显性位 1：发送帧中的ESI位将发送隐性位
3:0	DLENC[3:0]	数据长度 DLENC[3:0]表示帧内数据长度。

29.4.19. 接收 FIFO 邮箱 data0 寄存器(CAN_RFIFOMDATA0x)(x = 0,1)

地址偏移：0x1B8 + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB3[7:0]								DB2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
r								r							

位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

29.4.20. 接收 FIFO 邮箱 data1 寄存器(CAN_RFIFOMDATA1x)(x = 0,1)

地址偏移：0x1BC + 0x10 * x
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

29.4.21. 过滤器控制寄存器(CAN_FCTL)

地址偏移：0x200
复位值：0x2A1C 0E01

该寄存器只能按字(32 位)访问。

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		HBC1F[5:0]						保留							FLD
		rw													rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	HBC1F[5:0]	CAN1过滤器单元起始位置 这些位用来定义CAN1过滤器起始位置。CAN0可以用编号为0~HBC1F-1过滤器，CAN1可以用编号为HBC1F~27过滤器。当这些位的值为0，CAN0将没有过滤器可以使用。当这些位的值为28时，CAN1将没有过滤器可以使用。 此位域在CAN2中为保留位。
7:1	保留	必须保持复位值。
0	FLD	过滤器锁禁用 0：使能过滤器锁 1：禁用过滤器锁

29.4.22. 过滤器模式配置寄存器(CAN_FMCFG)

地址偏移：0x204
复位值：0x0000 0000

在 CAN0 和 CAN1 中的过滤器模式配置寄存器：

该寄存器只能按字(32 位)访问。修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FMOD27	FMOD26	FMOD25	FMOD24	FMOD23	FMOD22	FMOD21	FMOD20	FMOD19	FMOD18	FMOD17	FMOD16
				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
FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FMODx	过滤器模式 0：掩码模式 1：列表模式

在 CAN2 中的过滤器模式配置寄存器：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	FMODx	过滤器模式 0：掩码模式 1：列表模式

29.4.23. 过滤器位宽配置寄存器(CAN_FSCFG)

地址偏移：0x20C
复位值：0x0000 0000

在 CAN0 和 CAN1 中的过滤器位宽配置寄存器。

该寄存器只能按字(32 位)访问，修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FSx	过滤器位宽 0：16-bit位宽 1：32-bit位宽

CAN2 中的过滤器位宽配置寄存器：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	FSx	过滤器位宽 0：16-bit位宽 1：32-bit位宽

29.4.24. 过滤器关联 FIFO 寄存器(CAN_FAFIFO)

地址偏移：0x214
复位值：0x0000 0000

在 CAN0 和 CAN1 中与 FIFO 寄存器相关联的过滤器。

该寄存器只能按字(32 位)访问，修改此寄存器的值，需要将 CAN_FCTL 寄存器的 FLD 位置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FAFx	过滤器关联FIFO 0：关联FIFO0 1：关联FIFO1

在 CAN2 中与 FIFO 寄存器相关联的过滤器：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	FAFx	过滤器关联FIFO 0：关联FIFO0

1：关联FIFO1

29.4.25. 过滤器激活寄存器(CAN_FW)

地址偏移：0x21C
复位值：0x0000 0000

在 CAN0 和 CAN1 中的过滤器激活寄存器。

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				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FWx	过滤器激活 0：没有激活 1：激活工作

CAN2 中的过滤器激活寄存器：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	FWx	过滤器激活 0：没有激活 1：激活工作

29.4.26. 过滤器(x)数据(y)寄存器(CAN_FxDATAy)(x = 0…27, y = 0,1)

地址偏移：0x240+8 * x+4 * y, (在 CAN2 中：x = 0…13, y = 0,1)
复位值：0xXXXX XXXX

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	FDx	过滤器数据 掩码模式下： 0：标识符的Bit(x)不需参与比较 1：标识符的Bit(x)需要参与比较 列表模式下： 0：标识符的Bit(x)必须为0 1：标识符的Bit(x)必须为1

30. SDIO 接口(SDIO)

30.1. 简介

安全的数字输入/输出接口(SDIO)定义了 SD 卡、SD I/O 卡、多媒体卡(MMC)和 CE-ATA 卡主机接口，提供 APB2 系统总线与 SD 存储卡、SD I/O 卡、MMC 和 CE-ATA 设备之间的数据传输。

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。

所支持的多媒体卡(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。

所支持的 CE-ATA 系统规格书可以通过 CE-ATA 工作组网站(www.ce-ata.org)获取。

30.2. 主要特性

SDIO 的主要特征如下：

- **MMC：**与多媒体卡系统规格书 V4.2 及之前的版本全兼容。有三种不同的数据总线模式：1 位(默认)、4 位和 8 位；
- **SD 卡：**与 SD 存储卡规格版本 2.0 全兼容；
- **SD I/O：**与 SD I/O 卡规格版本 2.0 全兼容，有两种不同的数据总线模式：1 位(默认)和 4 位；
- **CE-ATA：**与 CE-ATA 数字协议版本 1.1 全兼容；
- 48MHz 数据传输频率和 8 位数据传输模式；
- 中断和 DMA 请求；
- 完成信号使能和失能(CE-ATA)。

注意：SDIO 在同一时间仅支持一个 SD、SD I/O、MMC4.2 或 CE-ATA 设备，但可支持多个 MMC4.1 或以前版本的卡。

30.3. 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。

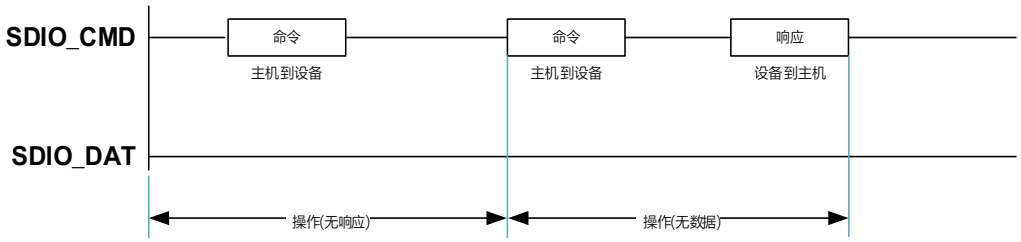
有几种不同类型的操作。一般操作总是包含一个命令和响应。此外，一些操作还有一个数据令牌。还有一些其他操作直接将他们的信息包含在命令或响应结构中。在这种情况下，操作没有数据令牌。在 DAT0-DAT7 和 CMD 信号线上的比特位根据主机时钟同步传输。

两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 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。与连续读相同，当 CMD 信号线上出现停止命令时，多块传输终止。

总线上的基本操作是命令/响应操作(参考*图 30-1. 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 30-1. 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 30-2. SDIO 多块读操作*所示，多个块的写操作如*图 30-3. 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。CE-ATA 设备在准备接收数据之前有一个可选的忙信号。

图 30-2. SDIO 多块读操作

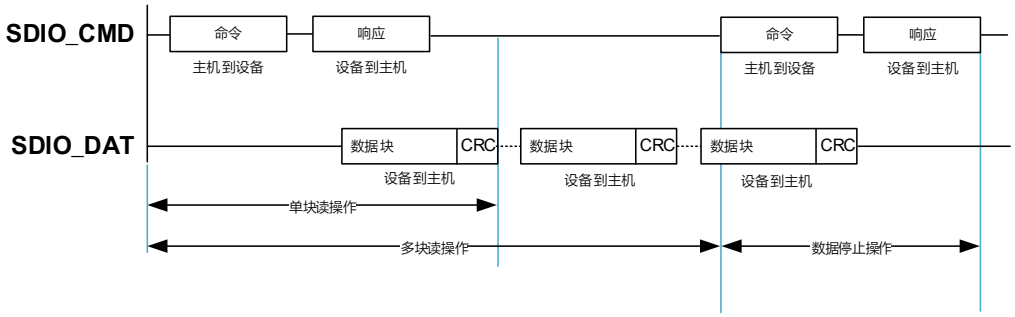
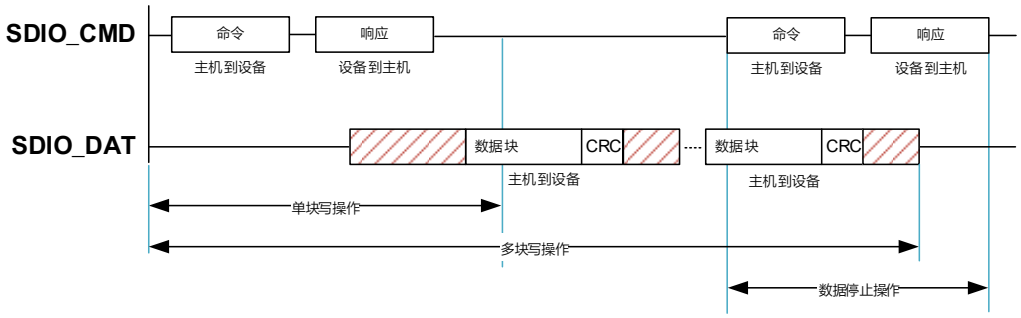


图 30-3. SDIO 多块写操作



SD 存储卡、SD I/O 卡(包括仅 IO 卡和组合卡)和 CE-ATA 设备直接的数据传输是以数据块的方式完成的。MMC 卡以数据块或数据流方式进行数据传输。*图 30-4. SDIO 数据流读操作*和*图 30-5. SDIO 数据流写操作*分别是数据流的读和写操作。

图 30-4. SDIO 数据流读操作

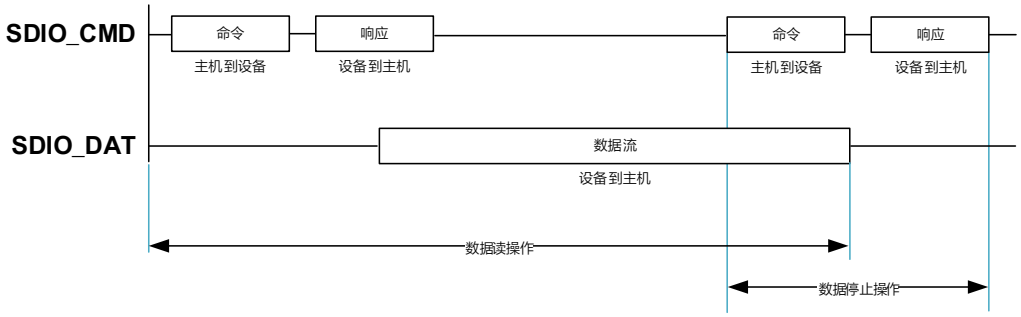
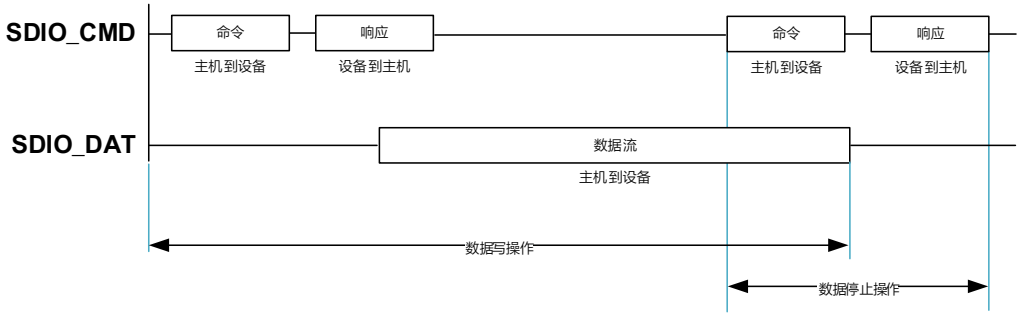


图 30-5. SDIO 数据流写操作

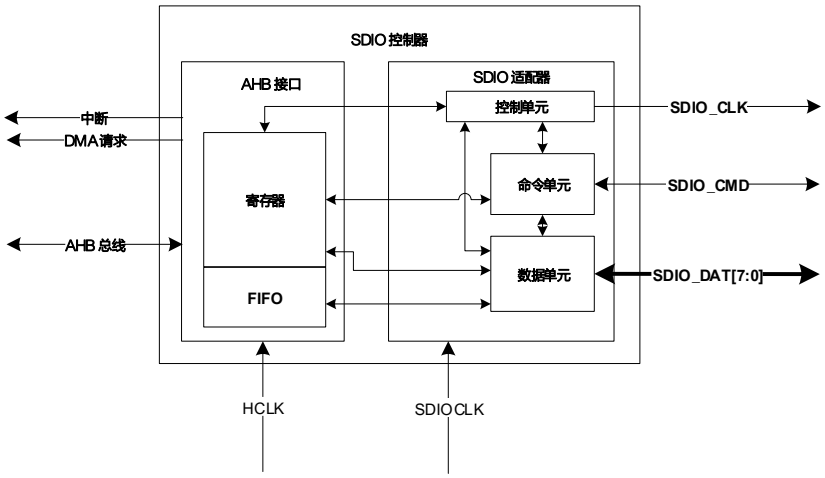


30.4. SDIO 功能描述

*图 30-6. SDIO 框图*显示了 SDIO 的结构框图，主要有两大部分：

- 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。
- APB2 接口：包括通过 APB2 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。

图 30-6. SDIO 框图



30.4.1. 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)和所有的数据线(SDIO_DAT)上直接发送一位命令或数据。对于 MMC 卡 V3.31 版本，SDIO_CLK 频率可以在 0 MHz 到 20 MHz 之间，对于 MMC 卡 V4.2 版本可以在 0 MHz 到 48 MHz 之间，对于 SD 或 SD I/O 卡可以在 0 MHz 到 25 MHz。

SDIO 使用两个时钟信号：SDIO 适配器时钟(SDIOCLK <= 48MHz)和 APB2 总线时钟(CK_APB2)。

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 MMC 卡 V3.31 及之前版本)和用于命令传送的推挽模式(SD 卡/SD I/O 卡和 MMC 卡 4.2 版本初始化时也是用推挽模式)。

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 MMC V4.2)。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 和 DAT2 的内部上拉(DAT3 内部上拉保持不变是由于 SPI 模式下 CS 片选的使用)。相应地，在进入 8 位模式后，断开 DAT1，DAT2 和 DAT4-DAT7 的内部上拉。

表 30-1. SDIO I/O 定义

引脚功能	信号	方向	描述
SDIO_CLK	SDIO_CLK	O	SD/SD I/O /MMC 时钟
SDIO_CMD	SDIO_CMD	I/O	命令的输入/输出
SDIO_D[7:0]	SDIO_DAT[7:0]	I/O	数据线 DAT[7:0]的数据输入/输出

SDIO 适配器是 SD/SD I/O /MMC/CE-ATA 的接口，它由 3 个子单元组成：

控制单元

控制单元包含电源管理功能和时钟管理功能用于存储卡时钟。电源管理是由 SDIO_PWRCTL 寄存器控制的，实现电源的掉电和上电。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。时钟管理向卡生成 SDIO_CLK 时钟信号。当 SDIO_CLKCTL 寄存器的 CLKBYP 位为 0 时，SDIO_CLK 由 SDIOCLK 分频得到；当 SDIO_CLKCTL 寄存器的 CLKBYP 位为 1 时，SDIO_CLK 直接为 SDIOCLK。

通过设置 SDIO_CLKCTL 寄存器的 HWCLKEN 位使能硬件时钟控制。该功能用于避免 FIFO 下溢和上溢错误，硬件根据系统总线是否繁忙，控制 SDIO_CLK 的开关。当 FIFO 不能接收或发送数据，主机将会关闭 SDIO_CLK 并冻结 SDIO 状态机来避免相关错误。只有状态机能被冻结，但 APB2 接口仍在工作。所以，FIFO 可以通过 APB2 总线访问。

命令单元

命令单元实现向卡发送和接收命令。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 CMDIDX 位不为 0b00 或 0b10 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	->	CS_Send
3.CSM 被关闭	->	CS_Idle
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

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.在 CE-ATA 模式下收到响应，失能 CE-ATA 中断并且等待 CE-ATA 设备命令完成信号使能	->	CS_Waitcompl
2.在 CE-ATA 模式下收到响应，失能 CE-ATA 中断并且等待 CE-ATA 设备命令完成信号失能	->	CS_Pend
3.CSM 被关闭	->	CS_Idle
4.收到响应	->	CS_Idle
5.命令 CRC 检测失败	->	CS_Idle

CS_Waitcompl	等待 CE-ATA 设备命令完成信号	
1.收到 CE-ATA 命令完成信号	->	CS_Idle
2.CSM 被关闭	->	CS_Idle
3.命令 CRC 检测失败	->	CS_Idle

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 使能并且数据传输方向为主机到卡	->	DS_WaitS
2.DSM 使能并且数据传输方向为卡到主机	->	DS_WaitR
3.DSM 使能并且读等待已经开始并且使能 SD I/O 模式	->	DS_Readwait

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	->	DS_Idle
3.数据 FIFO 为空标志无效	->	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_Idle
3.数据 FIFO 下溢错误发生	->	DS_Idle
4.内部 CRC 错误	->	DS_Idle

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	->	DS_Idle
3.DSM 被关闭	->	DS_Idle
4.数据超时发生	->	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.数据超时	->	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	->	DS_WaitR
2.数据传输结束	->	DS_WaitR
3.数据 FIFO 下溢发送	->	DS_Idle
4.数据已经接收并且读等待开始并且使能 SD I/O 模式	->	DS_Readwait
5.DSM 被关闭或 CRC 错误	->	DS_Idle

DS_Readwait	等待"读等待停止"指令	
1."读等待停止"使能	->	DS_WaitR
2.DSM 被关闭	->	DS_Idle

30.4.2. APB2 接口

APB2 接口实现了访问 SDIO 寄存器、数据 FIFO 和生成中断和 DMA 请求。它包括数据 FIFO 单元、寄存器单元和中断/DMA 逻辑。

至少一个已经被选中的状态标志为高时，中断逻辑产生中断。中断使能寄存器允许中断逻辑产生相应的中断。

DMA 接口提供一种方法，可以快速地在 SDIO 数据 FIFO 和存储器直接进行数据传输。下面的例子描述了如何实现这种方法：

1. 完成卡识别的过程。
2. 提高 SDIO_CLK 时钟频率。
3. 发送 CMD7 用于选择卡并配置总线宽度。
4. DMA1 的配置过程如下：

打开 DMA1 控制器并清除任何中断标志。用存储器基地址来配置 DMA1 通道 3 的源地址寄存器，用 SDIO_FIFO 寄存器的地址来配置 DMA1 通道 3 的目的地址寄存器。配置 DMA1 通道 3 的控制寄存器(存储器地址指针递增，外设地址指针固定，存储器和外设的数据宽度为字，M2M 失能)。

5. 写数据块(CMD24)到卡的过程如下：

以字节的形式将数据大小写入到 SDIO_DATALEN 寄存器中。以字节的形式将块大小(BLKSZ)写入到 SDIO_DATACTL 寄存器中，然后主机以每个块 BLKSZ 大小发送数据。向 SDIO_CMDAGMT 中写入数据的地址，该地址是卡中需要传输的数据地址。配置 SDIO 命令控制寄存器(SDIO_CMDCTL)：CMDIDX 置为 24，CMDRESP 置为 1(SDIO 卡主机等待短响应)，CSMEN 置为 1(发送命令使能)。其他字段为其复位值。

当 CMDRECV 标志被置位，配置 SDIO 数据控制寄存器(SDIO_DATACTL)：DATAEN 置为 1(发送数据使能)，DATADIR 置为 0(传输方向从控制器到卡)，TRANSMOD 置为 0(块传输)，DMAEN 置为 1(DMA 使能)，BLKSZ 置为 0x9(512 字节)。其他字段不用设置。

等待 DTBLKEND 标志位置位。通过轮询 DMA 中断标志寄存器，检查没有通道处于使能状态。

它还包括下面两个子单元：

寄存器单元

寄存器单元包含所有的系统寄存器，生成信号用于控制卡与控制器之间的通信。

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。发送 FIFO 被用在当需要写数据到卡上并且 SDIO_STAT 寄存器的 TXRUN 位为 1 时。待传输的数据通过 AHB 总线写入到发送 FIFO 中，SDIO 适配器中的数据单元从发送 FIFO 中读取数据，然后发送到卡上。接收 FIFO 被用在当需要从卡中读取数据并且 SDIO_STAT 寄存器的 RXRUN 为 1 时。从卡读取数据，然后将待传输的数据写入到接收 FIFO。在需要的时候，通过 AHB 总线读取接收 FIFO 中的数据。这个单元同样可以生成不同的 FIFO 标志(在 SDIO_STAT 寄存器中定义)。

30.5. 卡功能描述

30.5.1. 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(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 MMC 和 SD 卡之间有一点不同。主机可以使用 CMD1(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 MMC4.2 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 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(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，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。

30.5.2. 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 30-7. 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 30-7. 命令标记格式



表 30-2. 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 30-3. 卡命令类 (CCCs)*)。每类支持一组卡的功能。*表 30-3. 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 MMC 卡，类别为 0 的命令是强制性的，应被 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，设置在每个卡的卡特定数据(CSD)寄存器，提供给主机如何访问该卡信息。

对于 CE-ATA 设备,设备必须支持 MMC 命令,这些命令需要在设备初始化阶段完成传输状态。其它接口配置的设置，如总线宽度，可能需要额外的 MMC 命令来支持，具体请参考 MMC 引用。CE-ATA 利用以下的 MMC 命令：CMD0 - GO_IDLE_STATE，CMD12 - STOP_TRANSMISSION,CMD39 - FAST_IO,CMD60 - RW_MULTIPLE_REGISTER,CMD61 - RW_MULTIPLE_BLOCK。GO_IDLE_STATE(CMD0)，STOP_TRANSMISSION(CMD12)和 FAST_IO(CMD39) 由 MMC 引用定义。 RW_MULTIPLE_REGISTER(CMD60) 和 RW_MULTIPLE_BLOCK(CMD61)是 CE-ATA 协议定义的 MMC 命令。

表 30-3. 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	O										+		
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				+								
CMD23	M			+		+							
CMD24	M					+							

CMD25	M					+							
CMD26	M					+							
CMD27	M					+							
CMD28	M							+					
CMD29	M							+					
CMD30	M							+					
CMD32	M						+						
CMD33	M						+						
CMD34	O											+	
CMD35	O											+	
CMD36	O											+	
CMD37	O											+	
CMD38	M						+						
CMD39											+		
CMD40											+		
CMD42									+				
CMD50	O											+	
CMD52	O										+		
CMD53	O										+		
CMD55	M									+			
CMD56	M									+			
CMD57	O											+	
CMD60	M									+			
CMD61	M									+			
ACMD6	M									+			
ACMD13	M									+			
ACMD22	M									+			
ACMD23	M									+			
ACMD41	M									+			
ACMD42	M									+			
ACMD51	M									+			

注意： 1. CMD1, CMD11, CMD14, CMD19, CMD20, CMD23, CMD26, CMD39 和 CMD40 仅用于 MMC 卡。CMD5, CMD32-34, CMD50, CMD52, CMD53, CMD57 和 ACMDx 仅用于 SD 存储卡。CMD60, CMD61 仅用于 CE-ATA 设备。

2. 在使用 ACMD 命令之前发送 APP_CMD 命令(CMD55)。

3. CMD8 对于 MMC 卡和 SD 卡有不同的含义。

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 30-4. 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] 填充位	-	GO_IDLE_STATE	复位所有的卡到空闲状态。
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。
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 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 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)。
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)。
CMD12	ac	[31:0] 填充位	R1b	STOP TRANSMISSION	强制卡停止传输。

命令索引	类型	参数	响应格式	简称	描述
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 30-5. 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 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 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 READ_SINGLE_BLOCK 命令是一样的。
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 30-6. 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_STO	从卡中读取数据流，起始于给定的地

命令索引	类型	参数	响应格式	简称	描述
				P	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 30-7. 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 30-5. 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。
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 30-8. 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)

命令索引	类型	参数	响应格式	简称	描述
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。(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。(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 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 30-9. 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 30-10. 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 30-5. 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 30-11. 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	发送给主机容量支持信息(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

命令索引	类型	参数	响应格式	简称	描述
					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。
CMD60	adtc	[31] WR [23:18] 地址 [7:2] 字节数 其他位为保留位	R1(read)/ R1b(write)	RW_MULTIPLE_REGISTER	在地址范围内，读或写寄存器。
CMD61	adtc	[31] WR [15:0] 数据单元数 其他位为保留位	R1(read)/ R1b(write)	RW_MULTIPLE_BLOCK	在地址范围内，读或写寄存器。
注意： 1. ACMDx 是针对 SD 存储卡的特定应用命令 2. CMD60, CMD61 针对 CE-ATA 设备的特定应用命令					

表 30-12. 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 [7:0] 写数据/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		IO_RW_EXTENDED	该命令允许用一个简单命令读取

命令索引	类型	参数	响应格式	简称	描述
		[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			或写入大量的 I/ O 寄存器。
注意： 1.CMD39, CMD40 仅用于 MMC 卡 2. CMD52, CMD53 仅用于 SD I/O 卡					

表 30-13. 切换功能命令(class 10)

命令索引	类型	参数	响应格式	简称	描述
CMD6	adtc	[31] 模式 0：检测功能 1：切换功能 [30:24] 保留 [23:20] 为功能组 6 保留(0h 或 Fh) [19:16] 为功能组 5 保留(0h 或 Fh) [15:12] 为功能组 4 保留(0h 或 Fh) [11:8] 为功能组 3 保留(0h 或 Fh) [7:4] 功能组 2 命令系统 [3:0] 功能组 1 访问模式	R1	SWITCH_FUNC	仅用于 SD 存储卡和 SD I/O 卡。 检测可切换功能(模式 0)和切换卡功能(模式 1)。

30.5.3. 响应

所有的响应都是通过 CMD 信号线发送。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

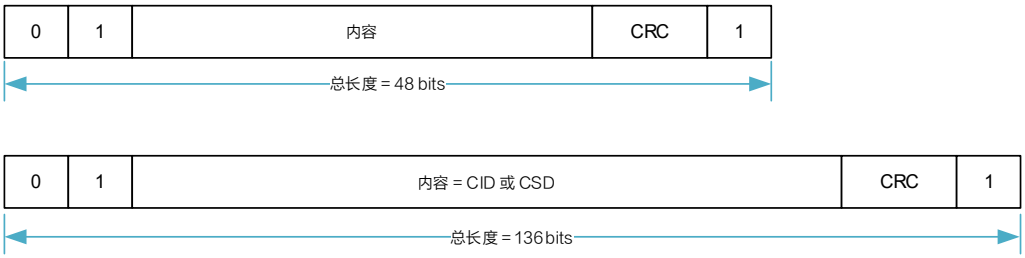
- **R5：** 中断请求
- **R6：** 发布的 RCA 响应
- **R7：** 卡接口条件

SD 存储卡支持其中的五种响应，R1 / R1b, R2, R3, R6, R7。SD I/O 卡和 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 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 30-8. 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 30-8. 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*数据包格式*中描述。

表 30-14. 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 30-15. 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(MMC)的响应被发送。不同卡的响应可能有一点不同。

表 30-16. 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 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 30-17. R4 响应(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 30-18. 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	当前存储	填充位	S18A	I/O OCR	保留	结束位

R5 (中断请求)

仅适用于 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 30-19. 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 30-20. 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 30-21. 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 30-22. 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	回送检查模式	CRC7	结束位

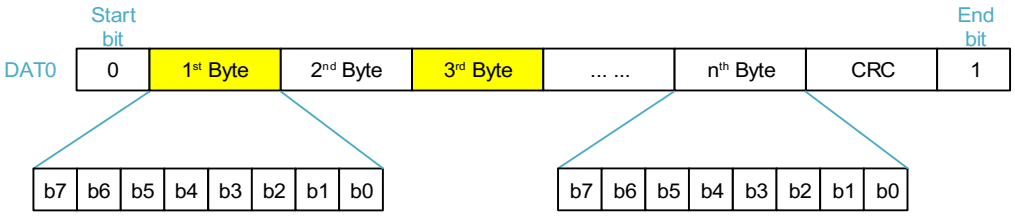
30.5.4. 数据包格式

数据总线模式有三种，1 位、4 位和 8 位宽度。1 位模式是强制的，4 位和 8 位模式是可选的。虽然使用 1 位模式，当卡复位和初始化时，DAT3 还需要通知卡当前的工作模式是 SDIO 或 SPI。

1 位数据包格式

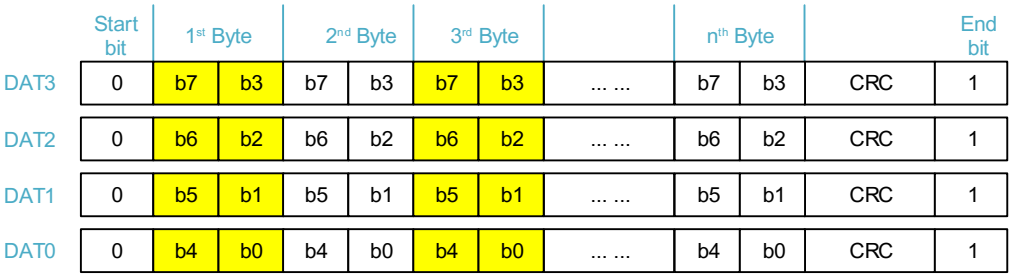
卡复位和初始化之后，只有 DAT0 被用于传输数据。其他引脚可以用于其他用处。*图 30-9. 1 位数据总线宽度*，*图 30-10. 4 位数据总线宽度*和*图 30-11. 8 位数据总线宽度*显示了数据宽度是 1 位，4 位和 8 位时的数据包格式。

图 30-9. 1 位数据总线宽度



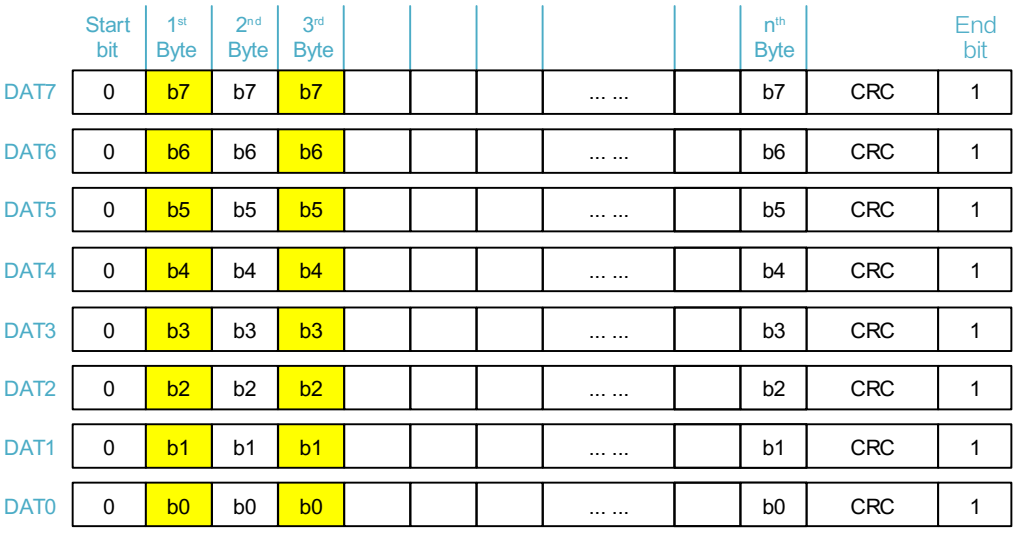
4 位数据包格式

图 30-10. 4 位数据总线宽度



8 位数据包格式

图 30-11. 8 位数据总线宽度



30.5.5. 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(有一个命令的延迟)。

•C: 读可清除。

表 30-23. 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_FAILED	ERX	'0'= 无错误 '1'= 错误	在上锁/解锁中有命令的顺序错误或检测到密码错误时置位。	C
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 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 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	执行命令时未使用内部 ECC。	A

位	标识符	类型	数值	说明	清除条件
			'1'= 失能		
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 9-14 = 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。这四个位按十进制数 0 至 15 解释。	B
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 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 状态寄存器是大小是一个数据块 512 比特。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 30-24. SD 状态

位	标识符	类型	数值	描述	清除条件
[511:51	DAT_BUS_WIDTH	SR	'00'= 1(默认)	由 SET_BUS_WIDTH 命令显示当	A

位	标识符	类型	数值	描述	清除条件
0]			'01'= 保留 '10'= 4 位宽 '11'= 保留	前定义的数据总线宽度	
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
[479:448]	SIZE_OF_PROTECTED_AREA	SR	受保护区域的大小。	(见下面描述)	A
[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 30-25. 移动性能字段*中定义。

表 30-25. 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 30-26. 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

AU_SIZE	数值定义
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 30-26. AU_SIZE 字段*中定义。卡可以任意的设置 AU 大小(由*表 30-27. 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 30-27. 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 30-28. 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：

$$\text{Erase timeout of X AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad (30\text{-}1)$$

表 30-29. 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

ERASE_TIMEOUT	数值定义
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 TOFFSET，可以选择如*表 30-30. 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 30-30. 擦除偏移字段

ERASE_OFFSET	数值定义
0h	0 秒
1h	1 秒
2h	2 秒
3h	3 秒

30.6. 编程序列

30.6.1. 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时钟速率 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 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 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 MMC、CE-ATA、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、MMC(CE-ATA)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 MMC 或者 CE-ATA 设备。

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；

- MMC/CE-ATA - 发送 CMD0，CMD1，CMD2，CMD3。

4. 识别 MMC/CE-ATA 设备。

- CPU 应该通过发送 CMD8 查询 EXT_CSD 寄存器的 504 字节(S_CMD_SET)。如果第 4 位被设置为 1，则该设备支持 ATA 模式；

- 如果支持 ATA 模式，CPU 应通过设置 EXT_CSD 寄存器的 191 字节(CMD_SET)的(第 4 位)ATA 位选择 ATA 模式，以激活使用 ATA 命令集。 CPU 使用 SWITCH(CMD6)命令选择命令集；

- 如果 CE-ATA 设备存在，FAST_IO(CMD39)和 RW_MULTIPLE_REGISTER(CMD60)命令将

会成功，并且返回的数据将会是 CE-ATA 复位签名。

30.6.2. 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。

30.6.3. 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 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。对于 CE-ATA，先用 CMD60 写 ATA 任务文件，然后使用 CMD61 命令写入数据。在写 CMD 寄存器之后，主机开始执行一个命令，当该命令被发送到总线时，CMDRECV 标志被设置。

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。

30.6.4. 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 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。对于 CE-ATA，先用 CMD60 写 ATA 任务文件，然后使用 CMD61 来读取数据。设置 CMD 寄存器之后，主机开始执行该命令，当该命令被发送到总线时，CMDRECV 标志被设置。

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。

30.6.5. 数据流写和数据流读(仅适用于 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：

$$\text{max write frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (30\text{-}2)$$

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：

$$\text{max read frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (30\text{-}3)$$

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。

30.6.6. 擦除

MMC/ SD 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。

30.6.7. 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 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。

30.6.8. 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/解锁操作

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。

30.6.9. 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 30-31. 上锁/解锁数据结构*为上锁/解锁命令的结构。

表 30-31. 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(SET_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。

30.7. 特定操作

30.7.1. SD I/O 特定操作

SD I/O 卡(包括仅 IO 卡和组合卡)支持这些特定操作：

读等待操作

暂停/恢复操作

中断

只有在 SDIO_DATACTL[11]位被设置时，SD I/O 才支持这些操作，但暂停读操作除外，因为它不需要特定的硬件实现。

SD I/O 读等待操作

读等待(RW)操作是可选择的，仅用于 SD I/O 的 1 位和 4 位模式。读等待操作允许一个主机给卡在执行一个读多个块(CMD53)操作时发信号，以暂时停止数据传输，同时允许主机发送命令到 SD I/O 卡内任何功能函数。如果要判断一个卡是否支持读等待协议，主机应测试 CCCR 的卡功能字节的 SRW 功能位。读等待时序是基于中断周期的。如果卡不支持读等待协议，只能表明主机在读取多个命令控制 SDIO_CLK 时已经暂停(不中止)数据。这种方法的局限是，随着时钟停止，主机不能发出任何命令，所以在延迟期间不能执行其他操作。支持读等待的卡是强制性支持暂停和恢复的。*图 30-12. 通过停止 SDIO_CLK 的读等待操作*和*图 30-13. 使用 SDIO_DAT[2]信号线的读等待操作*所示为通过停止 SDIO_CLK 和使用 SDIO_DAT[2]读等待模式。

图 30-12. 通过停止 SDIO_CLK 的读等待操作

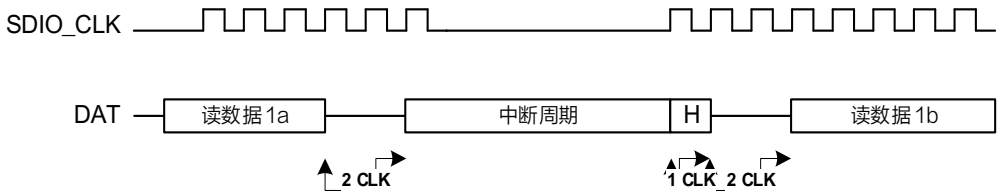
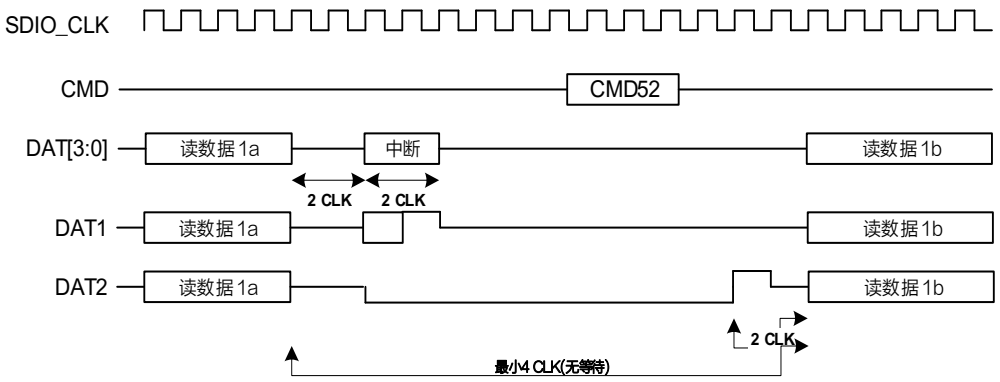


图 30-13. 使用 SDIO_DAT[2]信号线的读等待操作



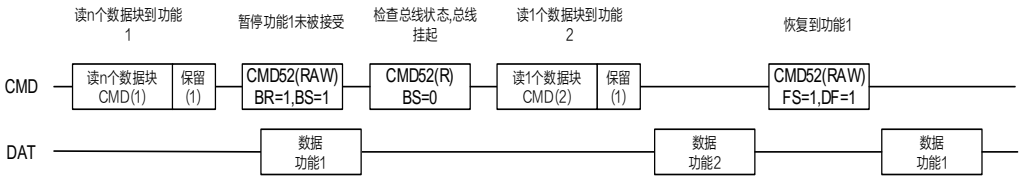
在接收到数据块之前就可以开始读等待：当数据单元使能(设置 SDIO_DATACTL[0]位)，SD I/O 特定操作使能(设置 SDIO_DATACTL[11]位)，开始读等待(SDIO_DATACTL[10] = 0 并且 SDIO_DATACTL[8] = 1)，数据方向为从卡到 SD I/O 主机 (SDIO_DATACTL[1] = 1)，DSM 直接从空闲状态到读等待状态。在读等待时，2 个 SDIO_CLK 时钟周期后，DSM 驱动 SDIO_DAT[2]为 0。在这种状态下，当设置了 RWSTOP 位(SDIO_DATACTL[9])时，DSM 会在等待状态多停留 2 个 SDIO_CLK 时钟周期，并在一个时钟周期中驱动 SDIO_DAT[2]为 1。然后 DSM 再次开始等待直到从卡里接收到数据。在接收数据块时，即使设置了开始读等待，DSM 也不会开始一个读等待间隔，读等待将在收到 CRC 后开始。必须清除 RWSTOP 才能开始新的读等待操作。在读等待期间，SDIO 主机可以在 SDIO_DAT[1]上监测 SD I/O 中断。

SD I/O 暂停/恢复操作

对于多功能 SD I/O 或组合卡，它们有多个设备(I/O 和存储)共享 SD 总线。为了允许主机同时访问多个设备，SD I/O 和组合卡可以实现可选的暂停/恢复操作。如果卡支持暂停/恢复，为了给其他的功能或者存储器提供更高优先级的传输而释放总线，主机可以暂停某个功能或者存储器的数据传输。一旦高优先级的传输完成后，原来的传输在暂停处重新开始。

*图 30-14. 在功能 1 的多块读周期期间插入功能 2 读周期*显示第一次暂停请求没有立即接受的条件。然后主机检查一个读请求的状态，并确定该总线已被释放(BS = 0)。此时，功能 2 的读操作被启动。一旦读取单个块完成，恢复发送功能，从而恢复数据传输(DF = 1)。

图 30-14. 在功能 1 的多块读周期期间插入功能 2 读周期



当主机向卡发送数据时，主机可以暂停写操作。设置 SDIO_CMDCTL[11]位并指示 CSM 当前的命令是一个暂停命令。CSM 分析响应，当从卡收到响应时(暂停被接受)，它确认 DSM 在收到当前数据块的 CRC 后进入空闲状态。

为了暂停读操作，DSM 在 WaitR 状态等待，在停止数据传输之前，当功能被挂起时一个完整的数据包。随后应用程序继续读出接收 FIFO 直到 FIFO 为空，最后 DSM 自动地进入空闲状态。

中断

为了允许 SD I/O 卡中断主机，SD 接口增加了一个中断功能的引脚。在 4 位模式下，引脚 8 被用作 SDIO_DAT[1]，它被用于卡到主机的中断信号。对于每张卡中断的功能是可选的。SD I/O 中断"电平敏感"，即中断线应保持有效(低)直到卡要么被主机认可并采取行动，要么或者由于中断周期结束而解除有效状态。一旦主机服务中断，通过函数的唯一 I/O 操作清除中断。

当设置 SDIO_DATACTL[11]位，SD I/O 中断可以在 SDIO_DAT[1]信号线上检测到。

*图 30-15. 读中断周期时序*显示单个数据读周期的中断周期时序。

图 30-15. 读中断周期时序

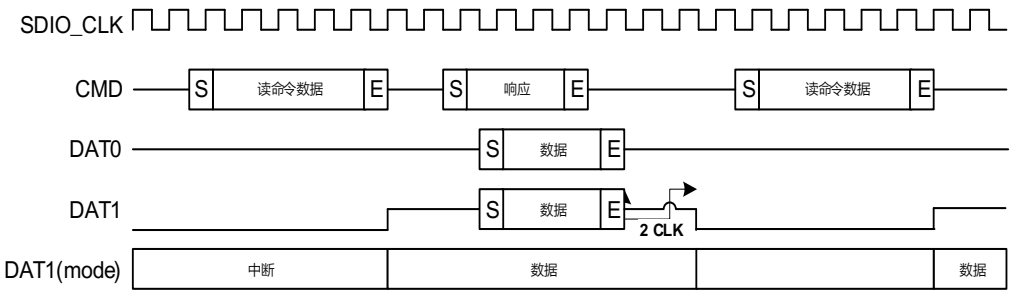
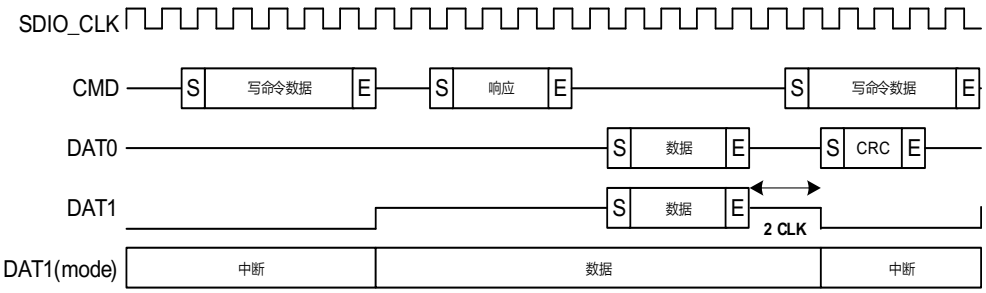


图 30-16. 写中断周期时序



当在 4 位 SD 模式传送数据的多个块时，需要中断周期的特定的定义。为了运行通信的最高速度，中断周期限制在 2 个时钟周期。卡如果想向主机发送一个中断信号，应该在第一个时钟周期设置 DAT1 为低，第二个时钟周期设置 DAT1 为高。然后卡应释放 DAT1 进入 Hi-Z 状态。*图 30-17. 4 位模式下多块读中断周期时序*显示了 4 位的多块读取时中断操作，*图 30-18. 4 位模式下多块写中断周期时序*显示了 4 位的多块写入时的中断操作。

图 30-17. 4 位模式下多块读中断周期时序

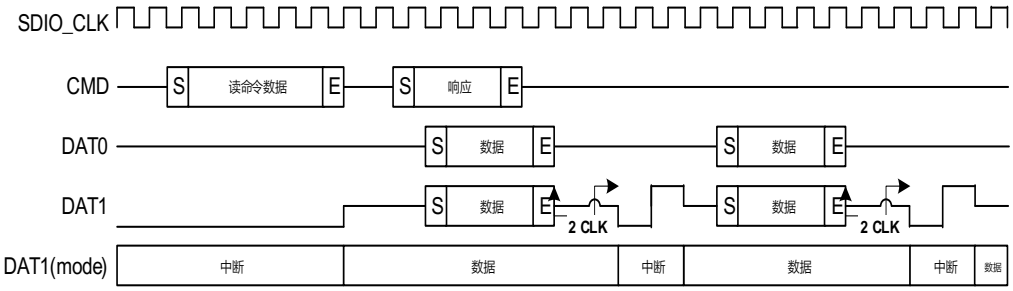
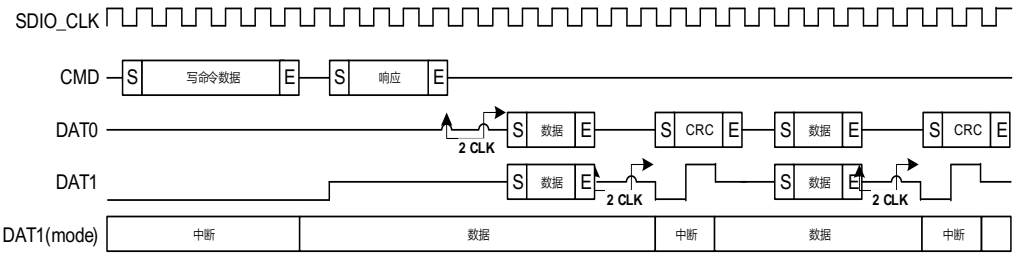


图 30-18. 4 位模式下多块写中断周期时序



30.7.2. CE-ATA 特定操作

CE-ATA 设备支持下述特定操作：

接收命令完成信号

发送命令完成关闭信号

只有当设置了 SDIO_CMDCTL[14]位时，SDIO 才支持这些操作。

命令完成信号

CE-ATA 定义了命令完成信号，设备使用该信号通知主机正常 ATA 命令完成或者由于设备遇到一个错误条件，ATA 命令终止。

如果"启用 CMD 完成"位 SDIO_CMDCTL[12]被设置并且"不中断使能'位 SDIO_CMDCTL[13]被设置，CSM 等待在 Waitcompl 状态的命令完成信号。

当在 CMD 线上接收到起始位，CSM 进入空闲状态。在 7 位周期之内不能发送新的命令。然后，在 5 个时钟周期内，把 CMD 信号变为 1(推挽模式)。

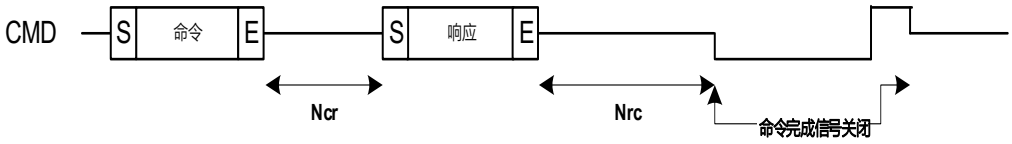
在主机从设备检测到一个命令完成信号之后，应该发送 FAST_IO(CMD39)命令来读取 ATA 状态寄存器以确定 ATA 命令的结束状态。

命令完成关闭信号

主机可以通过发送命令完成关闭信号来取消设备返回命令完成信号的功能。只有当主机在发送 RW_MULTIPLE_BLOCK (CMD61)之后接收到 R1b 响应后才能发送命令完成关闭信号。

如果未设置 SDIO_CMDCTL[12]中的"使能命令完成信号"并且重置了 SDIO_CMDCTL[13]中的"非中断使能位"，则在收到一个短响应后的 8 位周期之后，发出命令完成关闭信号。

图 30-19. 命令完成信号关闭操作



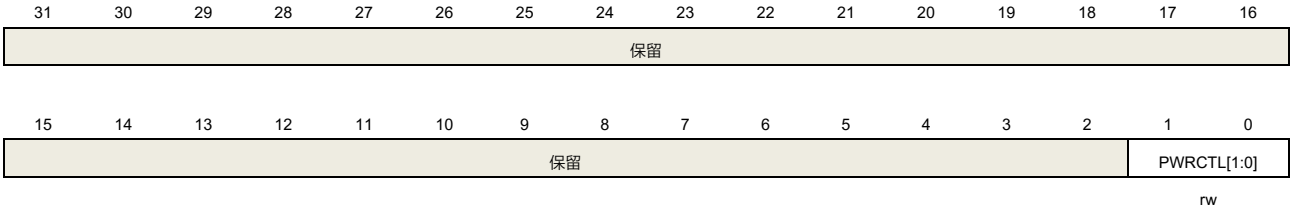
30.8. SDIO 寄存器

SDIO 基地址：0x4001 8000

30.8.1. 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
31:2	保留	必须保持复位值。
1:0	PWRCTL[1:0]	SDIO 电源控制位 这些位控制 SDIO 状态，卡输入或输出。 00：SDIO 电源关闭：SDIO CSM/DSM 复位到 IDLE，卡的时钟停止，没有命令/数据输出到卡 01：保留 10：保留 11：SDIO 上电

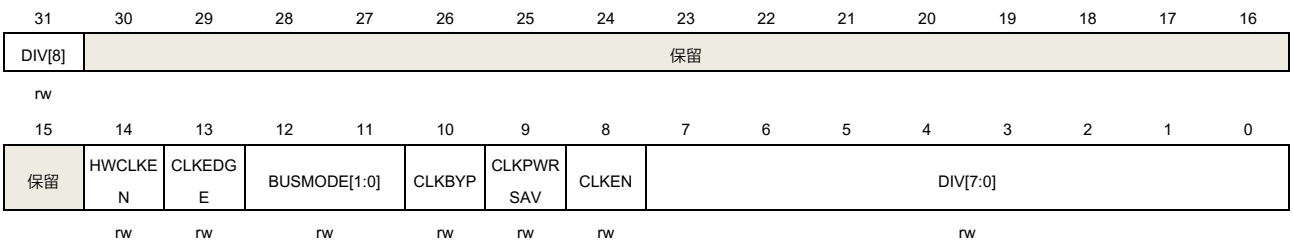
注意：两次对该寄存器写访问之间，需要至少 3 个 SDIOCLK 和 2 个 PCLK2 时钟周期，用于同步寄存器到 SDIOCLK 时钟域。

30.8.2. 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	DIV[8]	时钟分频系数的最高位 这个域定义了输入时钟(SDIOCLK)与输出时钟间的分频系数的最高位，参考 SDIO_CLKCTL 寄存器的 0 到 7 位。
30:15	保留	必须保持复位值。
14	HWCLKEN	硬件时钟控制使能位 如果该位置位，根据系统总线是否非常忙，硬件控制 SDIO_CLK 开/关。由于硬件可以在快要下溢/上溢时关闭 SDIO_CLK，所以当该位被置位时不会有下溢/上溢错误。 0：关闭硬件时钟控制 1：开启硬件时钟控制
13	CLKEDGE	SDIO_CLK 时钟边沿选择位 0：选择 SDIOCLK 的上升沿产生 SDIO_CLK 1：选择 SDIOCLK 的下降沿产生 SDIO_CLK
12:11	BUSMODE[1:0]	SDIO 卡总线模式控制位 00：1 位 SDIO 卡总线模式 01：4 位 SDIO 卡总线模式 10：8 位 SDIO 卡总线模式
10	CLKBYP	旁路时钟使能位 该位定义了 SDIO_CLK 直接来自于 SDIOCLK 或是 SDIOCLK 分频。 0：无旁路，SDIO_CLK 时钟参考 SDIO_CLKCTL 寄存器的 DIV 位域 1：旁路时钟，SDIO_CLK 时钟直接为 SDIOCLK (SDIOCLK/1)
9	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。 0：SDIO_CLK 时钟总是开启 1：SDIO_CLK 时钟在总线空闲时关闭
8	CLKEN	SDIO_CLK 时钟输出使能位 0：关闭 SDIO_CLK 1：开启 SDIO_CLK
7:0	DIV[7:0]	时钟分频 该个域和 DIV[8]位定义了分频因子来向卡产生 SDIO_CLK 时钟。如果 CLKBYP 位为 0，SDIO_CLK 是由 SDIOCLK 分频得到，并且 SDIO_CLK 频率= SDIOCLK / (DIV[8:0] + 2)。

注意：两次对该寄存器写访问之间，需要至少 3 个 SDIOCLK 和 2 个 PCLK2 时钟周期，用于同步寄存器到 SDIOCLK 时钟域。

30.8.3. 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，这些参数将被用作于命令的一部分(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 这个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。

30.8.4. 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ATAEN	NINTEN	ENCMDC	SUSPEND	CSMEN	WAITDEND	INTWAIT	CMDRESP[1:0]		CMDIDX[5:0]					
	rw	rw	rw	rw	rw	rw	rw	rw		rw					

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	ATAEN	CE-ATA 命令使能(仅用于 CE-ATA) 如果该位置位，主机进入 CE-ATA 模式，并且 CSM 传输 CMD61。 0：CE-ATA 失能 1：CE-ATA 使能
13	NINTEN	无 CE-ATA 中断(仅用于 CE-ATA) 该位定义了有无 CE-ATA 中断。该位仅用于 CE-ATA 卡的情况。

		0：CE-ATA 中断使能 1：CE_ATA 中断失能
12	ENCMDC	使能命令完成信号(仅用于 CE-ATA) 该位定义了在 CE-ATA 上有无命令完成信号。 0：无命令完成信号 1：有命令完成信号
11	SUSPEND	SD I/O 暂停命令(仅用于 SD I/O) 该位定义了 CSM 是否发送了暂停命令。该位仅用于 SDIO 卡。 0：无影响 1：暂停命令
10	CSMEN	命令状态机(CSM)使能位 0：命令状态机失能(停留在 CS_Idle) 1：命令状态机使能
9	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
8	INTWAIT	中断等待超时 该位定义了命令状态机在 CS_Wait 状态等待卡中断。如果该位被置位，无命令等待超时生成。 0：无等待中断 1：等待中断
7:6	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。 00：无响应 01：短响应 10：无响应 11：长响应
5:0	CMDIDX[5:0]	命令索引 这个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，需要至少 3 个 SDIOCLK 和 2 个 PCLK2 时钟周期，用于同步寄存器到 SDIOCLK 时钟域。

30.8.5. 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

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

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

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不未定义的。

30.8.6. 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

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

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

位/位域	名称	描述
31:0	RESPx[31:0]	卡状态。响应内容由*表 30-32. 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 30-32. 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

30.8.7. 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。

注意: 当需要数据传输时，数据定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。

30.8.8. 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，数据计数器加载到这个寄存器并开始递减。

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，数据定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。

30.8.9. 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				IOEN	RWTYPE	RWSTOP	RWEN	BLKSZ[3:0]				DMAEN	TRANSMOD	DATADIR	DATAEN
				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	IOEN	SD I/O 特定功能使能(仅用于 SD I/O) 0：未使能 SD I/O 特定功能 1：使能 SD I/O 特定功能
10	RWTYPE	读等待类型(仅用于 SD I/O) 0：使用 SDIO_DAT[2]控制读等待 1：通过停止 SDIO_CLK 控制读等待
9	RWSTOP	读等待停止(仅用于 SD I/O) 0：无影响 1：如果 RWEN 位被置位，停止读等待过程
8	RWEN	读等待模式使能(仅用于 SD I/O) 0：读等待模式失能 1：读等待模式使能
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。 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

3	DMAEN	DMA 使能位 0：DMA 失能 1：DMA 使能
2	TRANSMOD	数据传输模式 0：块传输模式 1：流传输或 SDIO 多字节传输模式
1	DATADIR	数据传输方向 0：写数据到卡上 1：从卡中读取数据
0	DATAEN	数据传输使能位 写 1 到该位开启数据传输不管该位为 0 或 1。如果 RWEN 置位，DSM 进入到读等待状态，或者根据 DATADIR 位 DSM 进入 WaitS 或 WaitR 状态。 开始一个新的数据传输，不需要清该位为 0。

注意：两次对该寄存器写访问之间，需要至少 3 个 SDIOCLK 和 2 个 PCLK2 时钟周期，用于同步寄存器到 SDIOCLK 时钟域。

30.8.10. 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 WaitR 或者 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。

30.8.11. 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[23:22, 10:0]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[21:11]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								ATAEND	SDIOINT	RXDTVAL	TXDTVAL	RFE	TFE	RFF	TFF
								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	RXRUN	TXRUN	CMDRUN	DTBLKEND	STBITE	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
31:24	保留	必须保持复位值。
23	ATAEND	CE-ATA 命令完成信号已接收(仅用于 CMD61)
22	SDIOINT	SD I/O 中断已接收
21	RXDTVAL	接收 FIFO 中的数据有效
20	TXDTVAL	发送 FIFO 中的数据有效
19	RFE	接收 FIFO 为空
18	TFE	发送 FIFO 为空，当硬件流控制使能，并且 FIFO 中包含 2 个字时，TFE 信号变得有效。
17	RFF	接收 FIFO 为满，当硬件流控制使能，RFF 信号在 FIFO 差 2 个字就满时变得有效。
16	TFF	发送 FIFO 为满
15	RFH	接收 FIFO 半满：FIFO 中至少还有 8 个字可被读取
14	TFH	发送 FIFO 半空：至少还有 8 个字可被写入到 FIFO 中
13	RXRUN	正在接收数据
12	TXRUN	正在传输数据
11	CMDRUN	正在传输命令
10	DTBLKEND	数据块已发送/已接收(CRC 检测通过)
9	STBITE	总线上起始位错误
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)

30.8.12. 中断清除寄存器(SDIO_INTC)

地址偏移：0x38
复位值：0x0000 0000

该寄存器为只读。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								ATAENDC	SDIOINTC	保留					
								w	w						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					DTBLKENDC	STBITEC	DTENDC	CMDSENDC	CMDRECVC	RXORECC	TXURECC	DTTMOUTC	CMDTMOUTC	DTCRCERRC	CCRCERRC
					w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	ATAENDC	ATAEND 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。
21:11	保留	必须保持复位值。
10	DTBLKENDC	DTBLKEND 标志清除位 写 1 清除标志。
9	STBITEC	STBITE 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。

30.8.13. 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								ATAENDIE	SDIOINTIE	RXDTVALIE	TXDTVALIE	RFEIE	TFEIE	RFFIE	TFFIE
								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	RXRUNIE	TXRUNIE	CMDRUNIE	DTBLKENDIE	STBITEIE	DTENDIE	CMDSENDIE	CMDRECVIE	RXOREIE	TXUREIE	DTTMOUTIE	CMDTMOUTIE	DTCRCERRIE	CCRCERRIE
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	保留	必须保持复位值。
23	ATAENDIE	CE-ATA 命令完成信号已接收中断使能 0：失能 CE-ATA 命令完成信号接收中断 1：使能 CE-ATA 命令完成信号接收中断

22	SDIOINTIE	SD I/O 中断已接收中断使能 0：失能 SD I/O 中断接收中断。 1：使能 SD I/O 中断接收中断。
21	RXDTVALIE	接收 FIFO 中的数据有效中断使能 0：失能接收 FIFO 数据有效中断。 1：使能接收 FIFO 数据有效中断。
20	TXDTVALIE	发送 FIFO 中的数据有效中断使能 0：失能发送 FIFO 数据有效中断。 1：使能发送 FIFO 数据有效中断。
19	RFEIE	接收 FIFO 空中断使能 0：失能接收 FIFO 空中断。 1：使能接收 FIFO 空中断。
18	TFEIE	发送 FIFO 空中断使能 0：失能发送 FIFO 空中断。 1：使能发送 FIFO 空中断。
17	RFFIE	接收 FIFO 满中断使能 0：失能接收 FIFO 满中断。 1：使能接收 FIFO 满中断。
16	TFFIE	发送 FIFO 满中断使能 0：失能发送 FIFO 满中断。 1：使能发送 FIFO 满中断。
15	RFHIE	接收 FIFO 半满中断使能 0：失能接收 FIFO 半满中断。 1：使能接收 FIFO 半满中断。
14	TFHIE	发送 FIFO 半满中断使能 0：失能发送 FIFO 半满中断。 1：使能发送 FIFO 半满中断。
13	RXRUNIE	正在接收数据中断使能 0：失能数据接收中断。 1：使能数据接收中断。
12	TXRUNIE	正在传输数据中断使能 0：失能数据发送中断。 1：使能数据发送中断。
11	CMDRUNIE	正在传输命令中断使能 0：失能命令发送中断。 1：使能命令发送中断。
10	DTBLKENDIE	数据块已发送/已接收中断使能

		0：失能数据块结束中断。 1：使能数据块结束中断。
9	STBITEIE	起始位错误中断使能 0：失能起始位错误中断。 1：使能起始位错误中断。
8	DTENDIE	数据结束中断使能 0：失能数据结束中断。 1：使能数据结束中断。
7	CMDSENDIE	命令发送完成中断使能 0：失能命令发送完成中断。 1：使能命令发送完成中断。
6	CMDRECVIE	命令响应已接收中断使能 0：失能命令响应接收中断。 1：使能命令响应接收中断。
5	RXOREIE	接收 FIFO 上溢错误中断使能 0：失能接收 FIFO 上溢错误中断。 1：使能接收 FIFO 上溢错误中断。
4	TXUREIE	发送 FIFO 下溢错误中断使能 0：失能发送 FIFO 下溢错误中断。 1：使能发送 FIFO 下溢错误中断。
3	DTTMOUTIE	数据超时中断使能 0：失能数据超时中断。 1：使能数据超时中断。
2	CMDTMOUTIE	命令响应超时中断使能 0：失能命令响应超时中断。 1：使能命令响应超时中断。
1	DTCRCERRIE	数据 CRC 错误中断使能 0：失能数据 CRC 错误中断。 1：使能数据 CRC 错误中断。
0	CCRCERRIE	命令响应 CRC 错误中断使能 0：失能命令响应 CRC 错误中断。 1：使能命令响应 CRC 错误中断。

30.8.14. FIFO 计数寄存器(SDIO_FIFOCNT)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字(32 位)访问

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

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	FIFOCNT[23:0]	FIFO 计数器 这些位定义了从 FIFO 中读取或写入到 FIFO 剩余的字数。当 DATAEN 置位时，它加载数据长度寄存器的值(如果 SDIO_DATALEN 是字对齐时，该值为 SDIO_DATALEN[24:2]；如果 SDIO_DATALEN 不是字对齐，该值为 SDIO_DATALEN[24:2]+1)，然后当写一个字到 FIFO 或从 FIFO 中读取一个字时，开始递减计数。

30.8.15. 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 32 个 32 位的字，地址偏移从 0x80 到 0xFC。

该寄存器只能按字(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。

31. 通用串行总线全速接口(USBFS)

31.1. 概述

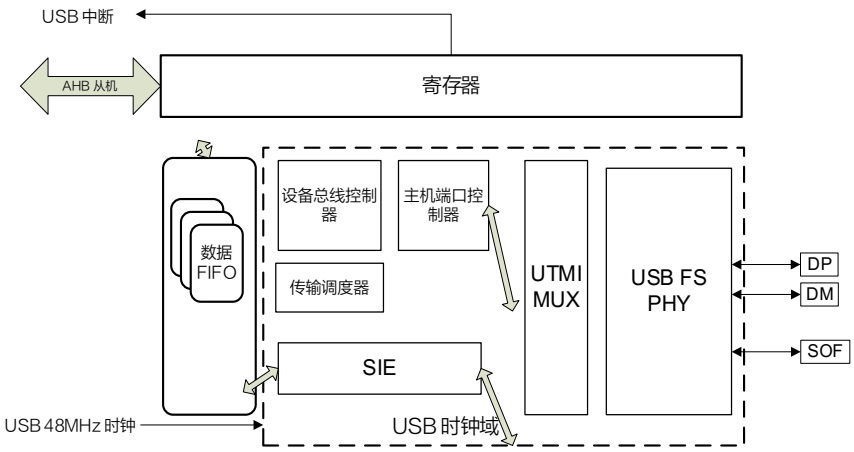
USB 全速(USBFS)控制器为便携式设备提供了一套 USB 互联解决方案。USBFS 支持主机模式和设备模式。USBFS 包含了一个内部的全速 USB PHY，并且不再需要外部 PHY 芯片。USBFS 可以支持 USB 2.0 协议所定义的所有四种传输方式(控制传输、批量传输、中断传输和同步传输)。

31.2. 主要特性

- 支持USB 2.0全速(12Mb/s)/低速(1.5Mb/s)主机模式；
- 支持USB 2.0全速(12Mb/s)设备模式；
- 支持所有的4种传输方式：控制传输、批量传输、中断传输和同步传输；
- 在主机模式下，包含USB事务调度器，用于有效地处理USB事务请求；
- 包含一个1.25KB的FIFO RAM；
- 在主机模式下，支持12个通道；
- 在主机模式下，包含2个发送FIFO(周期性发送FIFO和非周期性发送FIFO)和1个接收FIFO(由所有的通道共享)；
- 在设备模式下，包含6个发送FIFO(每个IN端点一个发送FIFO)和1个接收FIFO(由所有的OUT端点共享)；
- 在设备模式下，支持6个OUT端点和6个IN端点；
- 在设备模式下，支持远程唤醒功能；
- 包含一个支持USB协议的全速USB PHY；
- 在主机模式下，SOF的时间间隔可动态调节；
- 可将SOF脉冲输出到PAD；
- 在主机模式下，需要外部部件为连接的USB设备提供电源。

31.3. 结构框图

图 31-1. USBFS 结构框图



31.4. 信号线描述

表 31-1. USBFS 信号线描述

I/O 端口	类型	描述
DM	输入/输出	差分信号 D-端口
DP	输入/输出	差分信号 D+端口
SOF	输出	SOF 信号输出

31.5. 功能描述

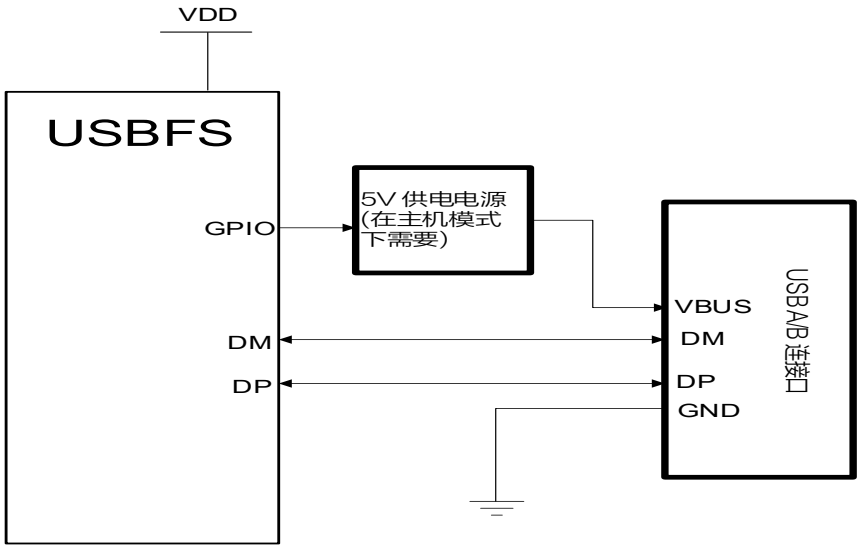
31.5.1. USBFS 时钟及工作模式

USBFS 可以作为一个主机、一个设备，并且包含一个内部全速 PHY。USBFS 可支持的最大速率为全速。

内部 PHY 支持全速和低速的主机模式、全速的设备模式。USBFS 所使用的 USB 时钟需要配置为 48MHz。该 48MHz USB 时钟从系统内部时钟产生，并且其时钟源和分频器需要在 RCU 模块中配置。

上拉或下拉电阻已经集成在内部全速 PHY 的内部，并且 USBFS 可根据当前模式(主机、设备)和连接状态进行自动控制。一个利用内部全速 PHY 的典型连接示意图如*图 31-2. 在主机或设备模式下连接示意图*所示。

图 31-2. 在主机或设备模式下连接示意图



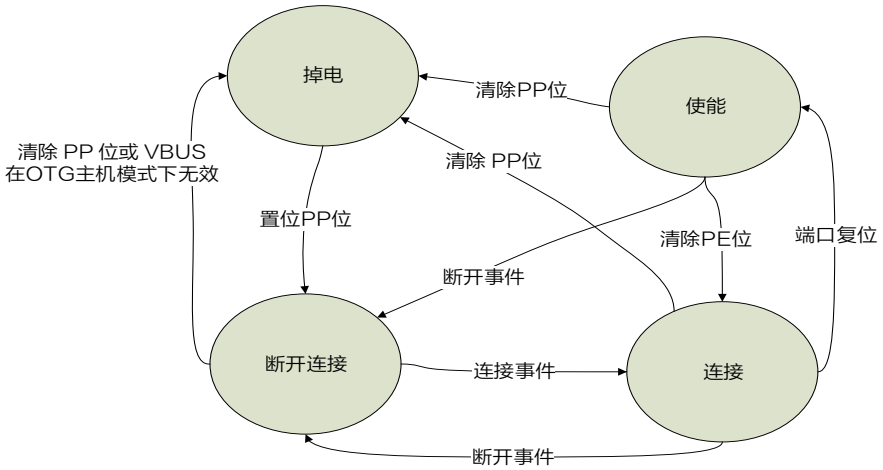
当 USBFS 工作在主机模式下时(FHM 控制位置位、FDM 控制位清除)，VBUS 为 USB 协议所定义的 5V 电源检测引脚。内部 PHY 不能提供 5V VBUS 电源。所以，如果应用需要提供 VBUS 电源，那么则需要一个外部的供电电源 IC。

31.5.2. USB 主机功能

USB主机端口状态

主机应用可以通过 USBFS_HPCS 寄存器控制 USB 端口状态。系统初始化之后，USB 端口保持掉电状态。通过软件置位 PP 控制位后，内部 USB PHY 将被上电，并且 USB 端口变为断开状态。检测到连接后，USB 端口变为连接状态。在 USB 总线上产生一个复位后，USB 端口将变为使能状态。

图 31-3. 主机端口状态转移图



连接、复位和速度识别

作为 USB 主机，在检测到一个连接事件后，USBFS 会为应用触发一个连接标志；同样，若检测到一个断开事件后，将会触发一个断开标志。

PRST 控制位用于实现 USB 复位序列。应用可以置位该控制位以启动一个 USB 复位序列，或者清除该控制位以结束 USB 复位序列。仅当端口在连接或使能状态时，该控制位有效。

USBFS 在对设备连接和复位时执行速度检测，并且速度检测的结果会反馈在 USBFS_HPCS 寄存器的 PS 位域中。USBFS 以 DM 或 DP 的电平状态确定设备速度，如 USB 协议所描述，全速设备上拉 DP 信号线，而低速设备上拉 DM 信号线。

挂起和复位

USBFS 支持挂起和复位状态，当 USBFS 端口在使能状态时，向 USBFS_HPCS 寄存器的 PSP 控制位写 1，USBFS 会进入到挂起状态。在挂起状态下，USBFS 停止在 USB 总线上发送 SOF，并且这样会让所连接的 USB 设备在 3ms 后进入挂起状态。应用程序能够置位 USBFS_HPCS 寄存器中的 PREM 控制位以启动一个恢复序列，从而唤醒挂起的设备，当清除该控制位时，则可以停止恢复序列。如果主机在挂起状态下检测到一个远程唤醒信号，将会置位 USBFS_GINTF 寄存器的 WKUPIF 标志位，并且触发 USBFS 唤醒中断。

SOF产生器

在主机模式下，USBFS 向 USB 总线发送 SOF 令牌包。如 USB 2.0 协议所描述，全速连接下，每毫秒产生一次 SOF 令牌包(由主机控制器或者 HUB 事务转换器产生)。

每当 USBFS 进入到使能状态后，它将会按照 USB2.0 所定义的周期发送 SOF 令牌包。然而，应用程序可以通过写 USBFS_HFT 寄存器中的 FRI 位来调整一帧的间隔。FRI 位定义了在一帧中的 USB 时钟周期个数，并且应用程序应该基于 USBFS 所使用的 USB 时钟频率计算该值。FRT 位显示当前帧剩余的时钟周期个数，并且在挂起状态时，该值将停止改变。

USBFS 能够在每个 SOF 令牌包中产生一个脉冲信号，并且将其输出至一个引脚。该脉冲信号长度为 12 个 HCLK 周期。如果应用程序希望使用该功能，需要置位 USBFS_GCCFG 寄存器的 SOFOEN 控制位，并且配置相应的引脚寄存器为 GPIO 功能。

USB通道和事务

USBFS 在主机模式下包含 12 个独立的通道。每个通道能够与一个 USB 设备端点通信。通道的传输类型、方向、数据包长和其他信息都在通道相应的寄存器中配置，例如 USBFS_HCHxCTL 和 USBFS_HCHxLEN 寄存器。

USBFS 支持所有的四种传输类型：控制、批量、中断和同步。USB 2.0 协议将这些传输类型划分为两类：非周期性传输(控制和批量)和周期性传输(中断和同步)。基于此，为了有效地进行事务调度，USBFS 包含两种请求队列：周期性请求队列和非周期性请求队列。在上述请求队列中的请求条目可能代表一个 USB 事务请求或者一个通道操作请求。

如果应用程序想要在 USB 总线上启动一个 OUT 事务，需要通过 AHB 寄存器接口向数据 FIFO 中写入数据包。USBFS 硬件会在整包数据写完后，自动产生一个事务请求并进入请求队列。

请求队列中的请求条目通过事务控制模块按顺序处理。USBFS 通常首先尝试处理周期性请求队列，然后处理非周期性请求队列。

帧起始后，USBFS 首先开始处理周期性队列，直到队列为空抑或当前周期性请求队列所需时间不够，然后处理非周期性队列。这种做法保证了一帧中周期性传输的带宽。每次 USBFS 从请求队列中读取并取出一个请求条目。如果取出的是通道禁用请求，这将直接禁用通道并准备处理下个条目。

如果当前请求是一个事务请求并且 USB 总线时间能够处理这个请求，USBFS 会使用 SIE 在 USB 总线上产生该事务。

在当前帧内，当前请求所需的总线时间不足时，如果当前请求为周期性请求，USBFS 停止处理该周期性请求队列，并启动处理非周期性请求。如果当前请求为非周期性请求，USBFS 会停止处理任何队列，并等待直到当前帧结束。

31.5.3. USB 设备功能

USB设备连接

在设备模式下，USBFS 在初始化后保持掉电状态。利用 VBUS 引脚上的 5V 电源连接 USB 主机后，USBFS 将进入供电状态。USBFS 首先打开 DP 信号线上的上拉电阻，之后主机将会检测到一个连接事件。

复位和速度识别

USB 主机在检测到设备连接之后，总是会启动一个 USB 复位序列，并且在设备模式下，检测到 USB 总线复位事件后，USBFS 会为软件触发一个复位中断。

在复位序列后，USBFS 将会触发 USBFS_GINTF 寄存器中的 ENUMF 中断，并且利用 USBFS_DSTAT 寄存器内的 ES 标志位指示当前枚举设备速度，该位总是为 0b '11'(全速)。

如 USB 2.0 协议所描述，USBFS 在外设模式下不支持低速。

挂起和唤醒

USB 总线保持 IDLE 状态并且数据线 3 毫秒无变化，USB 设备将会进入挂起状态。当 USB 设备在挂起状态时，软件能够关闭大部分的时钟以节省电能。USB 主机可以通过在 USB 总线上产生恢复信号，来唤醒挂起的设备。USBFS 检测到恢复信号后，将置位 USBFS_GINTF 寄存器的 WKUPIF 标志位并且触发 USBFS 唤醒中断。

在挂起设备模式，USBFS 也能够远程唤醒 USB 总线。软件可以通过置位 USBFS_DCTL 寄存器的 RWKUP 控制位来发送一个远程唤醒信号，并且如果 USB 主机支持远程唤醒，主机会在 USB 总线上启动发送一个恢复信号。

软件断开

USBFS 支持软件断开。设备进入到供电状态后，USBFS 会打开 DP 信号线的上拉电阻，并且这样主机会检测到设备连接。然后，软件可以通过置位 USBFS_DCTL 寄存器中 SD 控制位进行强制断开。在 SD 控制位被置位后，USBFS 将会直接关闭上拉电阻。这样，USB 主机将会在 USB 总线上检测到设备断开。

SOF跟踪

当 USBFS 在 USB 总线上接收到一个 SOF 令牌包时，将触发一个 SOF 中断，并且开始利用本地 USB 时钟计算总线时间。当前帧的帧号将会反应在 USBFS_DSTAT 寄存器的 FNRSOF 位域中。当 USB 总线时间达到 EOF1 或 EOF2 点(帧结束，在 USB 2.0 协议中描述)，USBFS 会触发 USBFS_GINTF 寄存器中的 EOPFIF 中断。软件能够使用这些标志位和寄存器以获得当前总线时间和位置信息。

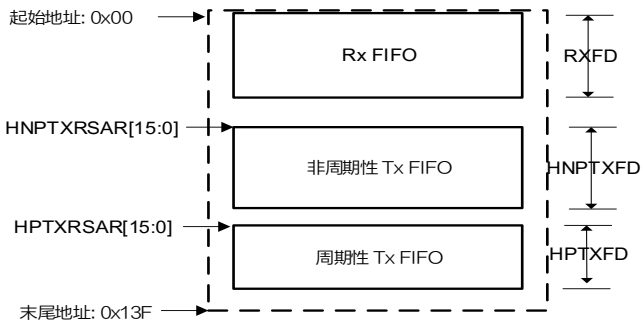
31.5.4. 数据 FIFO

USBFS 中采用 1.25K 字节数据 FIFO 存储包数据，数据 FIFO 是通过 USBFS 的内部 SRAM 实现的。

主机模式

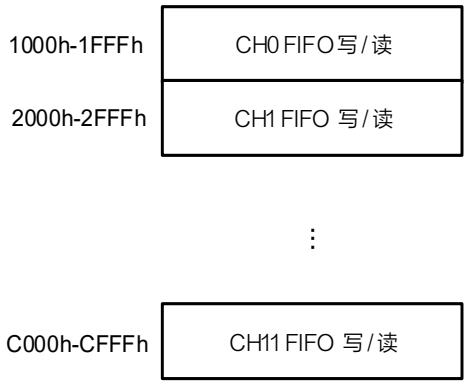
主机模式下，数据 FIFO 空间分为三个部分，分别是：用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据，所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBFS_GRFLEN、USBFS_HNPTFLEN 和 USBFS_HPTFLEN，软件可以配置以上数据 FIFO 的大小和起始偏移地址。*图 31-4. 主机模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是按照 32 位为单位写的。

图 31-4. 主机模式 FIFO 空间



USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 31-5. 主机模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。尽管所有的非周期通道共享相同的 FIFO 以及所有的周期通道共享相同的 FIFO，每个通道都拥有它们的 FIFO 访问寄存器空间。对 USBFS 而言，获知当前压入数据包的通道号是非常重要的，通过寄存器 USBFS_GRXTATR/USBFS_GRSTATP 来访问数据包所从属的 Rx FIFO。

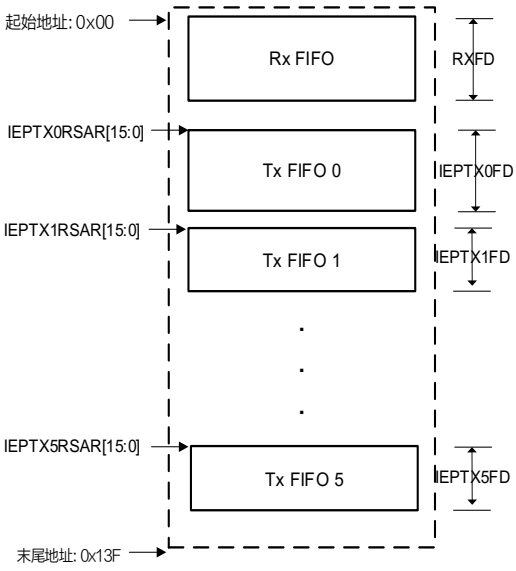
图 31-5. 主机模式 FIFO 访问寄存器映射表



设备模式

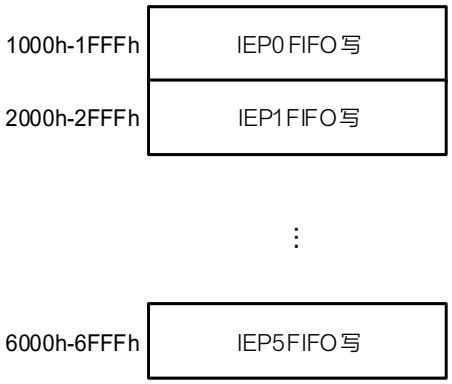
在设备模式下，数据 FIFO 分为多个部分，其中包含 1 个 Rx FIFO 和 6 个 Tx FIFO，每个 Tx FIFO 对应着一个 IN 端点，所有的 OUT 端点通过共享 Rx FIFO 接收数据包。通过寄存器 USBFS_GRFLEN 和 USBFS_DIEPxTFLEN (x=0…5)，程序可配置数据 FIFO 的大小和起始偏移地址。*图 31-6. 设备模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是以按照 32 位写的。

图 31-6. 设备模式 FIFO 空间



USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 31-7. 设备模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 所访问的寄存器存储空间，图中的数值是以字节为单位寻址。每个端点都拥有它们的 FIFO 访问寄存器空间。通过寄存器 USBFS_GRXTATR / USBFS_GRSTATP 来访问 Rx FIFO。

图 31-7. 设备模式 FIFO 访问寄存器映射表



31.5.5. 操作手册

该部分描述的是 USBFS 的操作手册。

主机模式

全局寄存器初始化顺序：

1. 根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
2. 根据应用的需求，如操作模式(主机、设备)，设置寄存器USBFS_GUSBCS；
3. 根据应用的需求，设置寄存器USBFS_GCCFG；
4. 根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
5. 通过设置寄存器USBFS_GINTEN使能模式错误和主机端口中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
6. 设置寄存器USBFS_HPCS，置位PP位；
7. 等待设备连接，当设备连接后，触发寄存器USBFS_HPCS的PCD位，然后置位PRST位，执行一次端口复位，等待至少10毫秒后，清除PRST位；
8. 等待USBFS_HPCS寄存器的PEDC中断，然后读取PE位以确认端口被成功地使能，读取PS位以获取连接的设备速度，之后，如果软件需要改变SOF间隔，设置USBFS_HFT寄存器。

通道初始化和使能顺序：

1. 根据期望的传输类型、方向、包大小等信息，设置寄存器USBFS_HCHxCTL，在设置期间，要保证位CEN和CDIS保持清除；
2. 设置寄存器USBFS_HCHxINTEN，设置期望的中断使能位；
3. 设置寄存器USBFS_HCHxLEN，PCNT表示一次传输中的包数，TLEN表示一次传输中发送或接收的包数据的总字节数；

对于 OUT 通道，如果 PCNT 为 1，单包的大小等于 TLEN。如果 PCNT 大于 1，前 PCNT-1 个包被认定为最大包长度的包，其大小是由寄存器 USBFS_HCHxCTL 的位 MPL 所定义。最

后一包的大小可通过 PCNT、TLEN 和 MPL 计算得到。如果程序想要发出一个零长度的包，应该设定 TLEN 为 0，PCNT 位 1；

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；

4. 置位寄存器USBFS_HCHxCTL中的CEN位以使能通道。

通道除能顺序：

程序可以通过同时置位 CEN 和 CDIS 除能通道。在寄存器操作后，USBFS 将在请求队列中产生一个通道除能请求条目。当这个请求条目到达请求队列的顶部时，USBFS 立即进行处理。

对于 OUT 通道而言，特定的通道将被立即除能。然后，会产生 CH 标志，USBFS 将清除 CEN 和 CDIS 位。

对于 IN 通道而言，USBFS 将通道除能状态条目压入 Rx FIFO，然后，程序应该处理 Rx FIFO 非空事件：读和取出该状态条目，然后会产生 CH 标志，USBFS 将清除 CEN 和 CDIS 位。

IN传输操作顺序：

1. 初始化USBFS全局寄存器；
2. 初始化相应的通道；
3. 使能相应的通道；
4. 通过软件使能IN通道后，USBFS在相应请求队列中生成一个Rx请求条目；
5. 当Rx请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。对于由请求条目所指示的事务而言，如果总线时间足够，USBFS在USB总线上开始IN事务；
6. 当IN事务结束时(收到ACK握手包)，USBFS将接收到的数据包压入Rx FIFO，ACK标志位被触发，否则，状态标志(NAK)会指示事务结果；
7. 如果步骤5所描述的IN事务完成后，步骤2的PCNT的数值比1大，程序将会返回步骤3，继续接收剩下的数据包。如果步骤5中描述的IN事务没有成功完成，程序将会返回步骤3来再次发送该数据包；
8. 在所有的传输中的所有事务都被成功接收后，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束；
9. 除能通道，当通道处于空闲状态，即可为其他传输做准备。

OUT传输操作顺序：

4. 初始化USBFS全局寄存器；
5. 初始化及使能相应通道；
6. 将数据包写入通道的Tx FIFO(周期性Tx FIFO或非周期性Tx FIFO)。在所有的数据包都被写入FIFO后，USBFS在相应的请求队列中产生一个Tx请求条目，并且将USBFS_HCHxTLEN中的TLEN值减少，减少的数值等于已写的包大小；
7. 当请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。如果请求条目对应的事务的总线时间足够，USBFS在USB总线上开展OUT事务；
8. 当由请求条目所指示的OUT事务结束时，寄存器USBFS_HCHnTLEN的位PCNT减1。如果该事务完成(收到ACK握手包)，ACK标志位被触发，否则，状态标志(NAK)会指示事务

结果；

9. 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；
10. 在所有的传输中的所有事务都被成功送达后，USBFS生成TF标志来指示传输成功结束；
11. 除能通道，当通道处于空闲状态，即可为其他传输做准备。

注意：开始写发送 FIFO 之前，需要保证发送 FIFO 以及请求队列存在剩余空间。

设备模式

全局寄存器初始化顺序：

1. 根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
2. 根据应用的需求，如操作模式(主机、设备)，设置寄存器USBFS_GUSBCS；
3. 根据应用的需求，设置寄存器USBFS_GCCFG；
4. 根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
5. 通过设置寄存器USBFS_GINTEN使能模式错误、挂起、SOF、枚举完成和USB复位中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
6. 根据应用的需求，如设备的地址等，设置寄存器USBFS_DCFG；
7. 在设备连接上主机上后，主机在USB总线上执行端口复位，触发寄存器USBFS_GINTF的RST中断；
8. 等待寄存器USBFS_GINTF的ENUMF中断。

端点初始化和使能顺序：

1. 根据预期的传输类型、包大小等信息，设置寄存器 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL；
2. 设定寄存器 USBFS_DIEPINTEN 或 USBFS_DOEPINTEN，置位相应中断使能位；
3. 设定寄存器 USBFS_DIEPxLEN 或 USBFS_DOEPxLEN，PCNT 表示一次传输中的包数，TLEN 表示一次传输中发送或接收的数据包的总字节数；

对于 IN 端点，如果 PCNT 等于 1，单数据包的大小等于 TLEN。如果 PCNT 大于 1，前 PCNT-1 个包被认定为最大包长度的包，其大小是由寄存器 USBFS_DIEPxCTL 的位 MPL 所定义。最后一包的大小可通过 PCNT、TLEN 和 MPL 计算得到。如果程序想要发出一个零长度的包，应该设定 TLEN 为 0，PCNT 位 1；

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；

4. 置位 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL 寄存器 EPEN 位使能端点。

端点除能顺序:

当 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL 寄存器的 EPEN 位被清除时，程序可以在任何时候除能端点

IN传输操作顺序：

1. 初始化USBFS全局寄存器；
2. 初始化和使能IN端点；
3. 将数据包写入端点的Tx FIFO，每当数据包写入FIFO，USBFS减少USBFS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的数据包大小；
4. 当IN令牌接收后，USBFS发送数据包，在USB总线上的事务完成后，USBFS_DIEPxLEN寄存器的PCNT值减1。如果事务成功完成(接收到ACK握手包)，ACK标志被触发，或者，其他状态标志表示事务的结果；
5. 在一次传输的所有数据包都被成功发送，USBFS生成一个TF标志位以表明传输成功结束，除能相应IN端点。

OUT传输操作顺序：

1. 初始化USBFS全局寄存器；
2. 初始化和使能端点；
3. 当OUT令牌接收后，USBFS接收数据包或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成(USBFS接收并保存数据到Rx FIFO，发送ACK握手包)，USBFS_DOEPxLEN寄存器的PCNT值减1。如果事务成功完成(接收到ACK握手包)，ACK标志被触发，或者，其他状态标志表示事务的结果；
4. 在一次传输的所有数据包都被成功接收，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束。USBFS生成一个TF标志位以表明传输成功结束，除能相应OUT端点。

31.6. 中断

USBFS 有两种中断：全局中断、唤醒中断。

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTF 寄存器读取，列举在*表 31-2. USBFS 全局中断*。

表 31-2. USBFS 全局中断

中断标志	描述	运行模式
DISCIF	断开连接中断标志	主机模式
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
PXNCIF/ ISOONCIF	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

中断标志	描述	运行模式
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
MFIF	模式错误中断标志	主机或设备模式

唤醒中断可以在 USBFS 处于挂起状态时触发，即使 USBFS 的时钟停止。寄存器 USBFS_GINTF 的位 WKUPIF 是唤醒源。

31.7. USBFS 寄存器

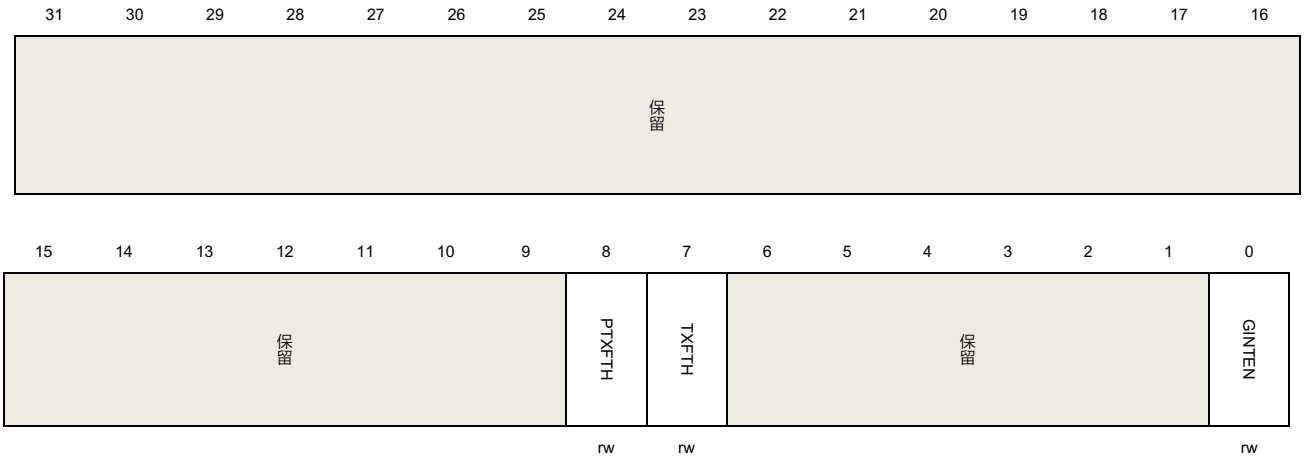
USBFS 基地址：0x5000 0000

31.7.1. 全局控制与状态寄存器组

全局 AHB 控制和状态寄存器 (USBFS_GAHBCS)

地址偏移：0x0008
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	PTXFTH	周期性Tx FIFO阈值 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
6:1	保留	必须保持复位值。
0	GINTEN	全局中断使能

0：全局中断不使能

1：全局中断使能

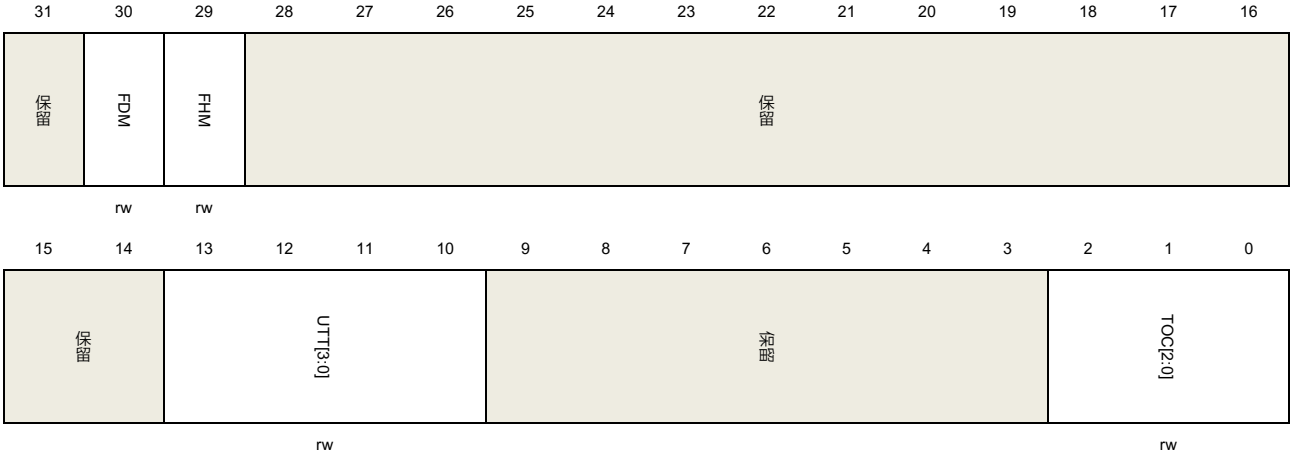
注意：在主机和设备模式下，均可访问

全局 USB 控制和状态寄存器 (USBFS_GUSBCS)

地址偏移：0x000C

复位值：0x0000 0880

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USBFS 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USBFS 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
28:14	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

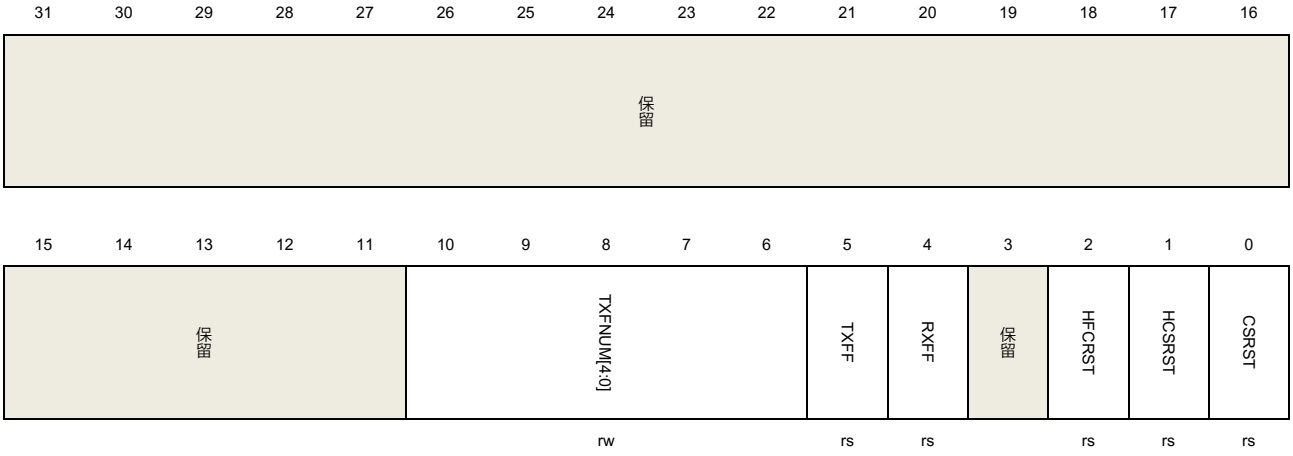
9:3	保留	必须保持复位值。
2:0	TOC[2:0]	超时校准 当等待一个包时，USBFS需要使用USB2.0协议中需要的超时数值。应用可以使用TOC[2:0]增加该数值(以PHY时钟为单位)。PHY时钟频率为48MHz。

全局复位控制寄存器 (USBFS_GRSTCTL)

地址偏移：0x0010
复位值：0x8000 0000

应用通过该寄存器来复位内核的不同硬件特性。

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:11	保留	必须保持复位值。
10:6	TXFNUM[4:0]	Tx FIFO数目 当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被冲刷 主机模式： 00000：仅非周期性Tx FIFO被冲刷 00001：仅周期性Tx FIFO被冲刷 1xxxx：周期性和非周期性Tx FIFO均被冲刷 其他：没有数据被冲刷 设备模式： 00000：仅Tx FIFO0被冲刷 00001：仅Tx FIFO1被冲刷 … 00101：仅Tx FIFO5被冲刷 1XXXX：所有的Tx FIFO均被冲刷

其他：没有数据被冲刷

5	TXFF	Tx FIFO冲刷控制位 应用通过置位该控制位来冲刷Tx FIFO数据，并且TXFNUM[4:0]决定冲刷的FIFO数目。当冲刷完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS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USBFS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USBFS内的帧计数器。该控制位置位后，接下来SOF的帧计数器将变为0。当复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。 **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USBFS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 (USBFS_GINTF)

地址偏移：0x0014

复位值：0x0000 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	保留	DISCIF	保留		PTXFEIF	HCIF	HPIF	保留		PXNCIF/ISOONCIF	ISOINCIF	OEPIF	IEPIF	保留	
rc_w1		rc_w1			r	r	r			rc_w1	rc_w1	r	r		
15		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	保留	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1		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	保留	必须保持复位值。
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
28:27	保留	必须保持复位值。
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USBFS_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，USBFS将置位该标志位。软件应该首先读取USBFS_HACHINT寄存器以获取通道号，然后读取相应的USBFS_HCHxINTF寄存器以获取产生中断的通道标志位。当产生通道中断的独立通道标志位被清除后，该中断标志位将自动清除。 **注意**：仅在主机模式下访问
24	HPIF	主机端口中断标志位 当USBFS在主机模式下检测到端口状态改变时，USB内核将置位该标志位。软件应该读取USBFS_HPCS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，USBFS将置位该标志位(主机模式)。
	ISOONCIF	同步OUT传输未完成中断标志位 在周期性帧结束时(由USBFS_DCFG寄存器的EOPFT控制位定义)，如果仍有同步OUT端点未完成传输，USBFS将置位该标志位(设备模式)。

20	ISOINCIF	同步IN传输未完成中断标志位 在周期性帧结束时(由USBFS_DCFG寄存器的EOPFT控制位定义)，如果仍有同步IN端点未完成传输，USBFS将置位该标志位(设备模式)。 **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，USBFS将置位该中断标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_ DOEPxINTF寄存器以获取产生中断的端点标志位。当产生中断的相应端点标志位被清除后，该中断标志位被自动清除。 **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，USBFS将置位该标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_DIEPxINTF寄存器以获取产生中断的端点标志位。当相应产生中断的端点标志位被清除后，该中断标志位被自动清除。
17:16	保留	必须保持复位值。
15	EOPFIF	周期性帧结束中断标志位 当一帧内USB总线时间已经达到USBFS_DCFG寄存器中EOPFT控制位所定义的数值时，USBFS将置位该中断标志位。 **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USBFS接收到一个同步OUT包，但是Rx FIFO没有足够的空间来接收该OUT包，USBFS将置位该标志位。 **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，USBFS将置位该中断标志位。软件能够读取USBFS_DSTAT寄存器以获取当前设备速度。 **注意**：仅在设备模式下访问
12	RST	USB复位中断标志位 当USBFS在USB总线上检测到一个USB复位信号后，USBFS将置位该中断标志位。 **注意**：仅在设备模式下访问
11	SP	USB挂起中断标志位 当USBFS检测到USB总线空闲3ms并且进入挂起状态，USBFS将置位该中断标志位。 **注意**：仅在设备模式下访问
10	ESP	早期挂起中断标志位 当USBFS检测到USB总线空闲3ms时，USBFS将置位该中断标志位。

9:8	保留	必须保持复位值。
7	GONAK	全局OUT NAK有效标志位 软件能够向USBFS_DCTL寄存器的SGONAK控制位写1，并且USBFS将会在SGONAK写入有效后，置位GONAK标志位。 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USBFS_DCTL寄存器中的SGINAK控制位写1，并且USBFS将会在SGINAK写入有效后，置位GNPINAK标志位 **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USBFS_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，USBFS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，USBFS将置位该中断标志位。软件可以通过写1清除该中断标志位。 设备模式： 当USBFS接收到一个SOF令牌包后，USBFS置位该标志位。应用可以读取设备状态寄存器以获取当前帧号。软件可以通过写1清除该中断标志位。 **注意**：在设备和主机模式下，均可访问
2	保留	必须保持复位值。
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，USBFS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 (USBFS_GINTEN)

地址偏移：0x0018
复位值：0x0000 0000

这个寄存器同全局中断标志寄存器(USBFS_GINTF)一起工作来中断应用程序。当中断使能位被禁止后，相应的中断就不会产生。然而，相应的全局中断标志位依然会被置位。

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIE	保留	DISCIE	保留		PTXFEIE	HCIE	HPIE	保留		PXNCIE/ ISOONCIE	ISOINCIE	OEPIE	IEPIE	保留	
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	保留	MFIE	保留
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	保留	必须保持复位值。
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
28:27	保留	必须保持复位值。
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	保留	必须保持复位值。
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(USBFS_GRSTATR / USBFS_GRSTATP)

读地址偏移：0x001C

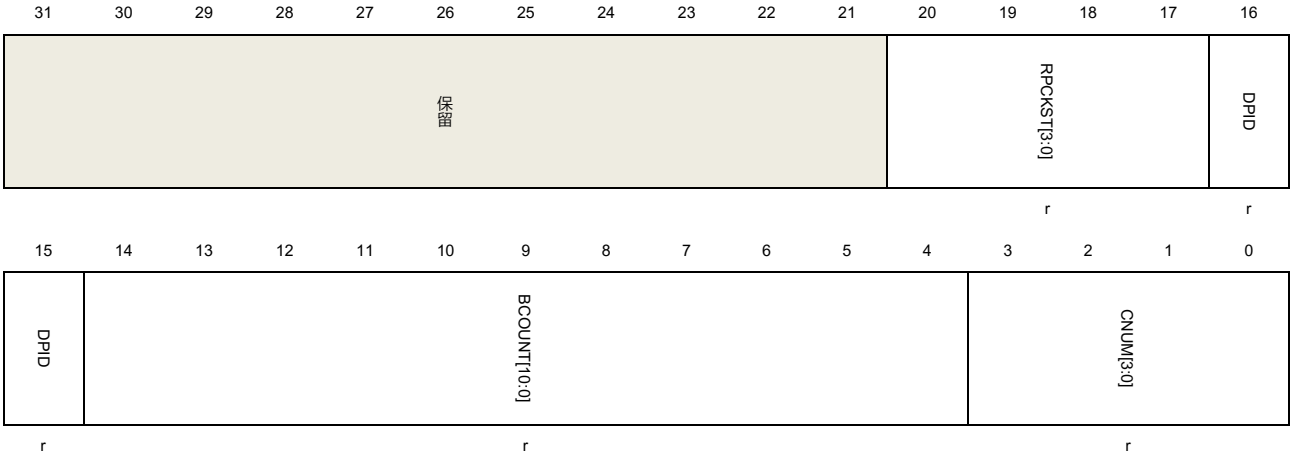
弹出地址偏移：0x0020
复位值：0x0000 0000

对接收状态读寄存器的读操作，将返回接收 FIFO 中顶部的条目。对接收状态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(USBFS_GINTF)中的接收 FIFO 非空中断标志位(RXFNEIF)置位后，软件应该读取该寄存器。

该寄存器只能按字(32 位)访问

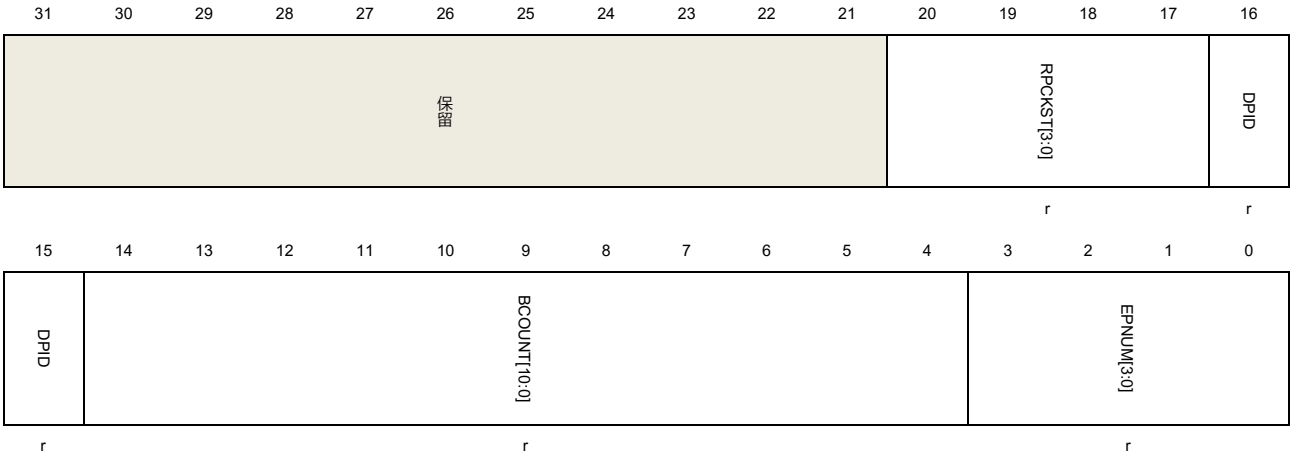
主机模式:



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:17	RPCKST[3:0]	接收包状态 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 其他：保留

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:



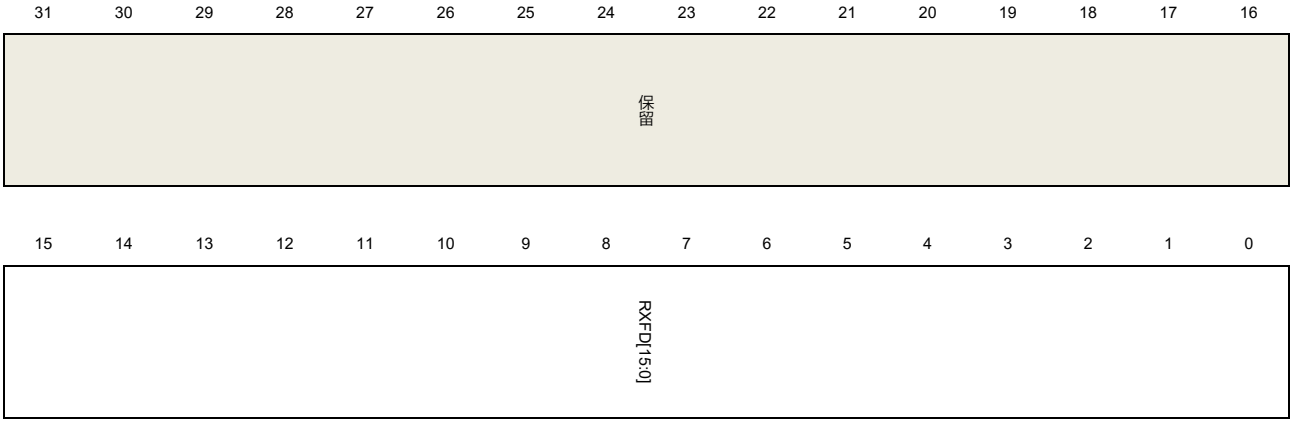
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:17	RPCKST[3:0]	接收包状态 0001：全局OUT NAK(产生一个中断) 0010：接收到OUT数据包 0011：OUT传输完成(产生一个中断) 0100：SETUP传输完成(产生一个中断) 0110：接收到SETUP数据包 其他：保留
16:15	DPID[1:0]	数据PID 接收到OUT数据包的数据PID 00：DATA0 10：DATA1 其他：保留
14:4	BCOUNT[10:0]	字节数 接收数据包的字节数
3:0	EPNUM[3:0]	端点号 当前接收包所属端点编号

全局接收 FIFO 长度寄存器 (USBFS_GRFLEN)

地址偏移：0x0024

复位值：0x0000 0080

该寄存器只能按字(32 位)访问



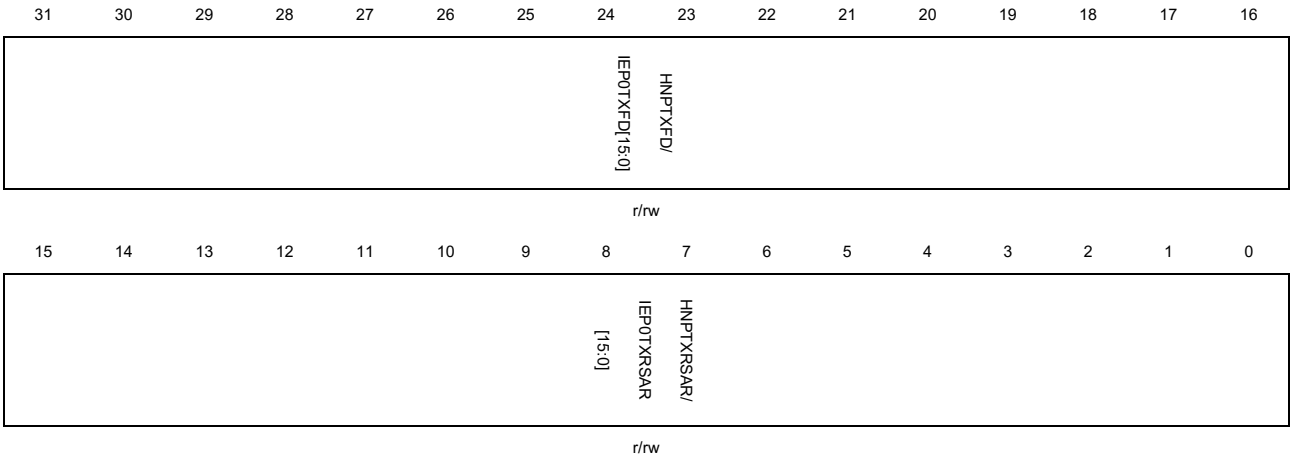
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RXFD[15:0]	Rx FIFO 深度 以32位字计数 1<=RXFD<=1024

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 (USBFS_HNPTFLEN_DIEP0TFLEN)

地址偏移：0x0028

复位值：0x0080 0080

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 (USBFS_HNPTFQSTAT)

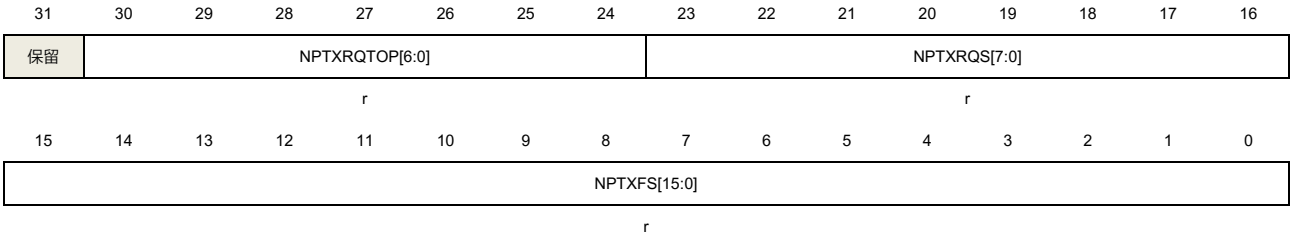
地址偏移：0x002C
复位值：0x0008 0080

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。

该寄存器只能按字(32 位)访问



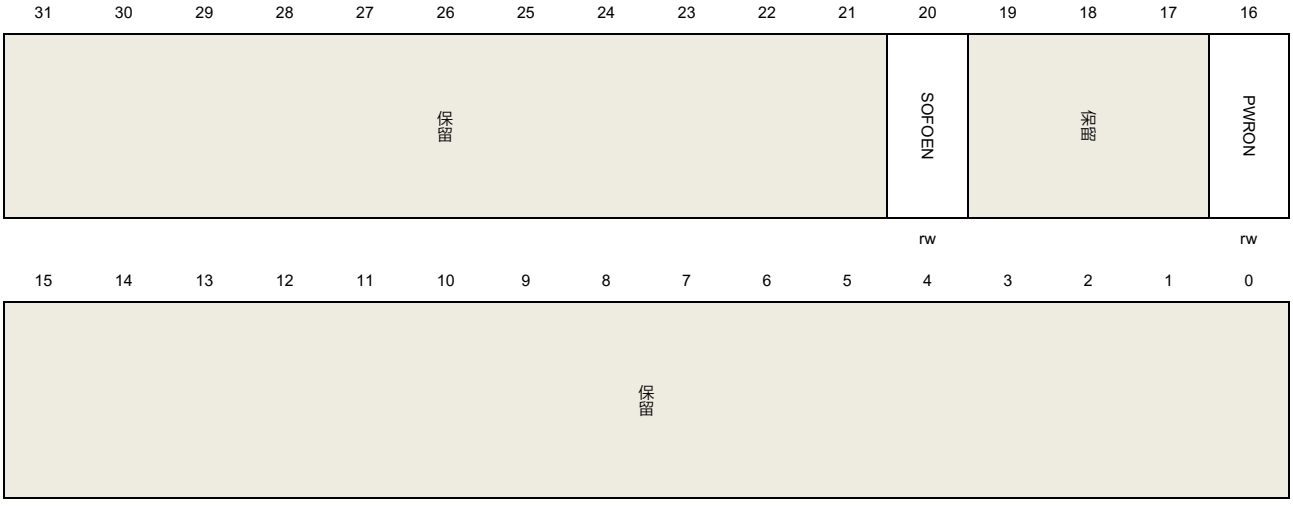
位/位域	名称	描述
31	保留	必须保持复位值。
30:24	NPTXRQTOP[6:0]	非周期性发送请求队列的顶部条目 在非周期性传输请求队列中的条目。 位30:27：通道号 位26:25： - 00：IN/OUT令牌 - 01：0长度OUT包 - 11：通道中止请求 位24：结束标志位，表明所选通道的最后一个条目

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 (USBFS_GCCFG)

地址偏移：0x0038
复位值：0x000C 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	SOFOEN	SOF输出使能 0：SOF脉冲输出禁止 1：SOF脉冲输出使能

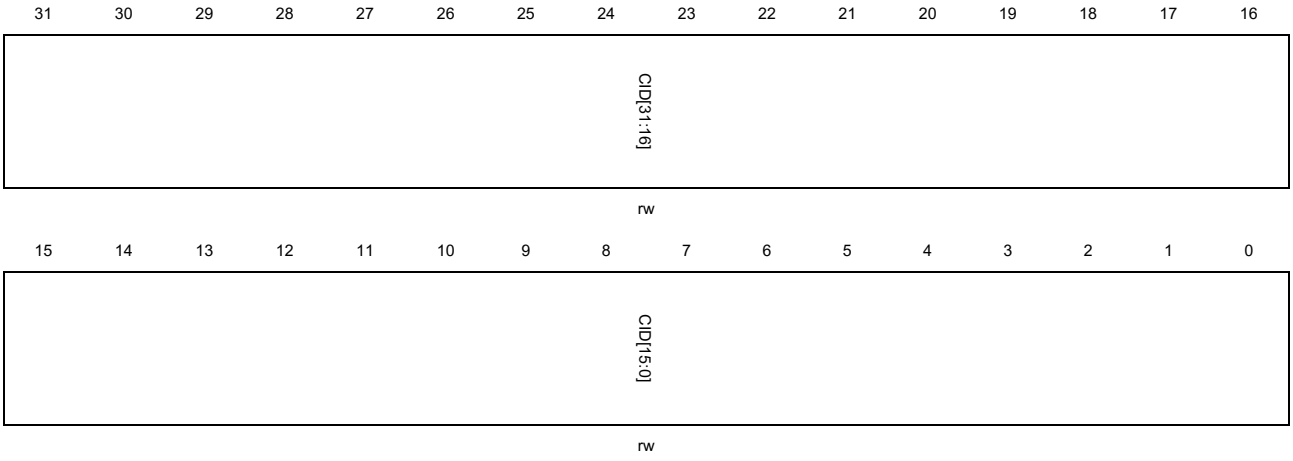
19:17	保留	必须保持复位值。
16	PWRON	上电 该控制位为内部嵌入式全速PHY的电源开关 0：嵌入式全速PHY掉电 1：嵌入式全速PHY上电
15:0	保留	必须保持复位值。

内核 ID 寄存器 (USBFS_CID)

地址偏移：0x003C
复位值：0x0000 1000

该寄存器包含产品 ID

该寄存器只能按字(32 位)访问

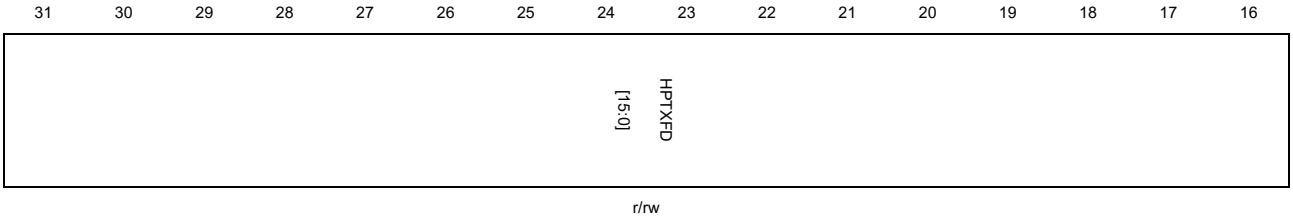


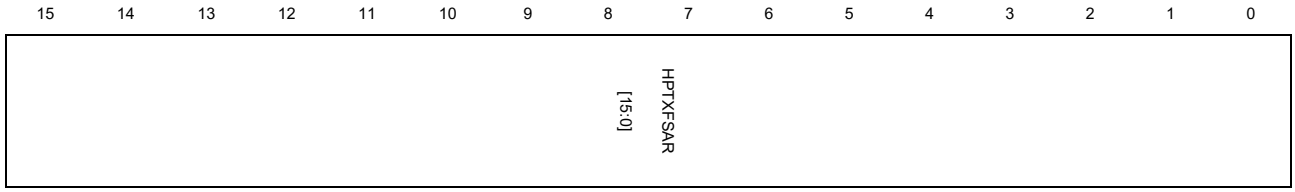
位/位域	名称	描述
31:0	CID[31:0]	内核ID 软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

主机周期性发送 FIFO 长度寄存器 (USBFS_HPTFLEN)

地址偏移：0x0100
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



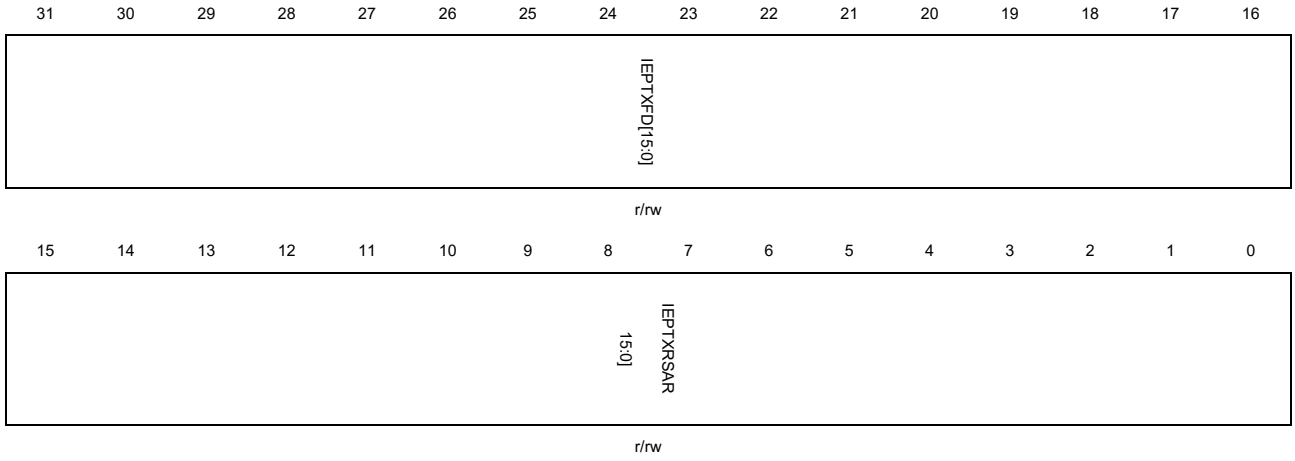


位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 1<=HPTXFD<=1024
15:0	HPTXFSAR[15:0]	主机周期性Tx RAM起始地址 主机周期性发送FIFO RAM起始地址

设备 IN 端点发送 FIFO 长度寄存器 (USBFS_DIEPxTFLEN) (x = 1...5，其中 x 是 FIFO 编号)

地址偏移：0x0104 + (FIFO 编号-1) × 0x04
复位值：0x0000 00000

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

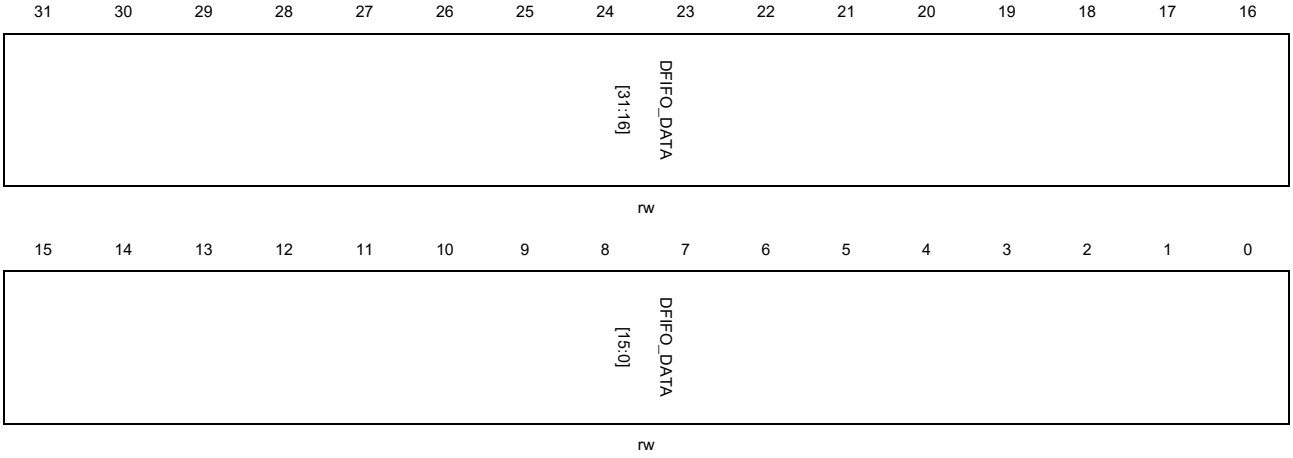
数据 FIFO 寄存器(USBFS_DFIFO)

地址偏移：0x1000

主机模式下：写地址范围：[0x1000,0x8FFF]，读地址范围：[0x1000,0xFFFF]

设备模式下：写地址范围：[0x1000,0x4FFF]，读地址范围：[0x1000,0xFFFF]

复位值：0x0000 0000



位/位域	名称	描述
31:0	DFIFO_DATA [31:0]	写该地址会将data push到相应的data FIFO中 读该地址会从相应的RX FIFO中pop出data值，读溢出的地址会读到上次pop的最新值。

31.7.2. 主机控制和状态寄存器组

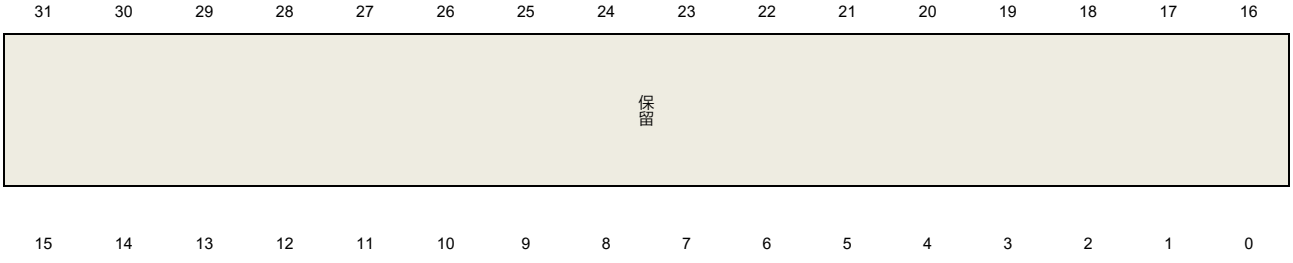
主机帧间隔寄存器 (USBFS_HFT)

地址偏移：0x0404

复位值：0x0000 BB80

当 USBFS 控制器正在枚举中时，该寄存器为当前枚举速度设置帧间隔。帧间隔的修改将在下一帧生效。

该寄存器只能按字(32 位)访问



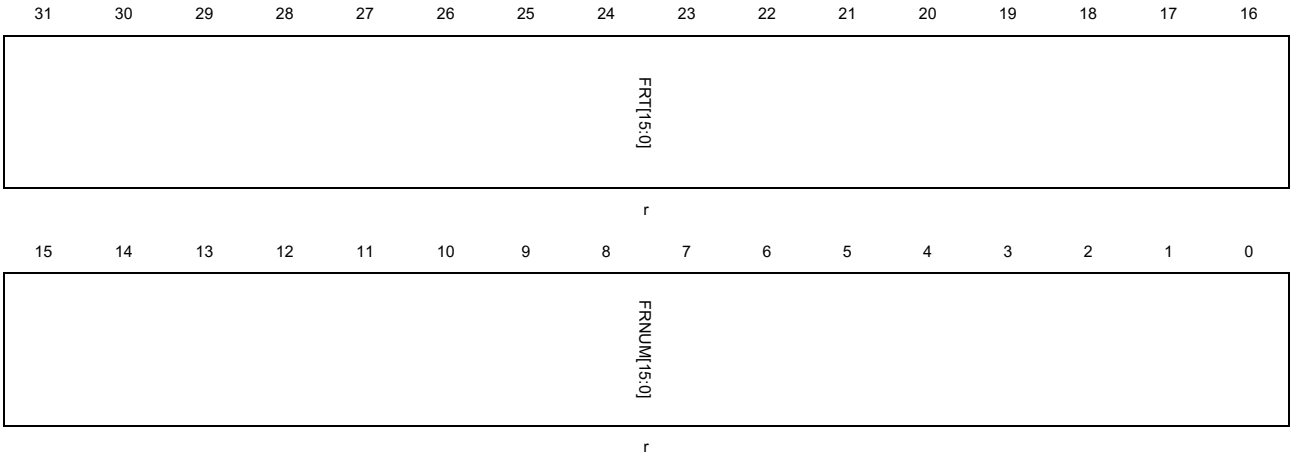


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	FRI[15:0]	帧间隔 该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBFS根据当前速度，采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算： 全速：48MHz 低速：6MHz

主机帧信息保持寄存器 (USBFS_HFINFR)

地址偏移：0x0408
复位值：0xBB80 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:16	FRT[15:0]	帧剩余时间 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。

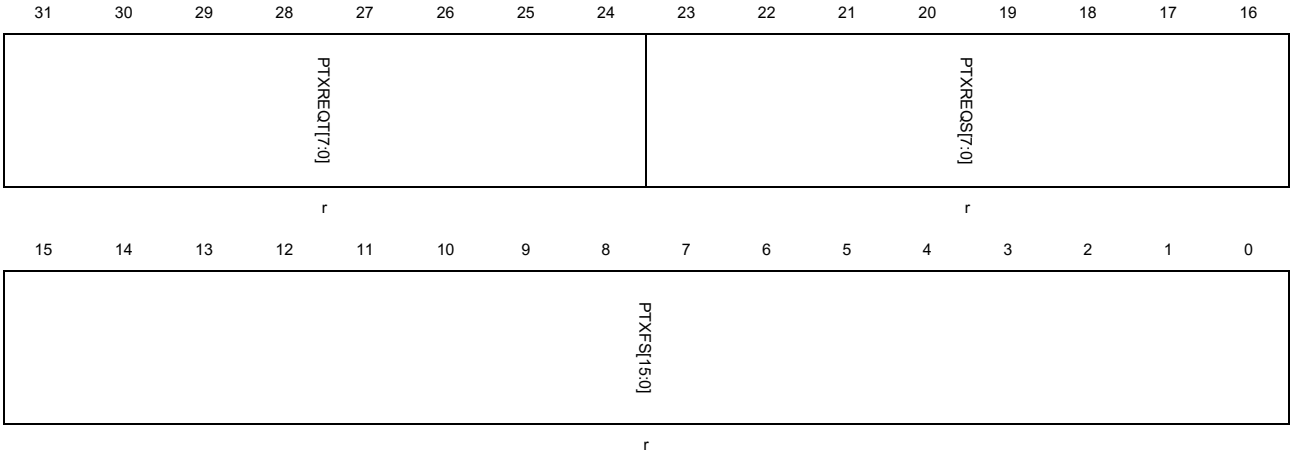
主机周期性发送 FIFO/队列状态寄存器 (USBFS_HPTFQSTAT)

地址偏移：0x0410

复位值：0x0008 0000

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))

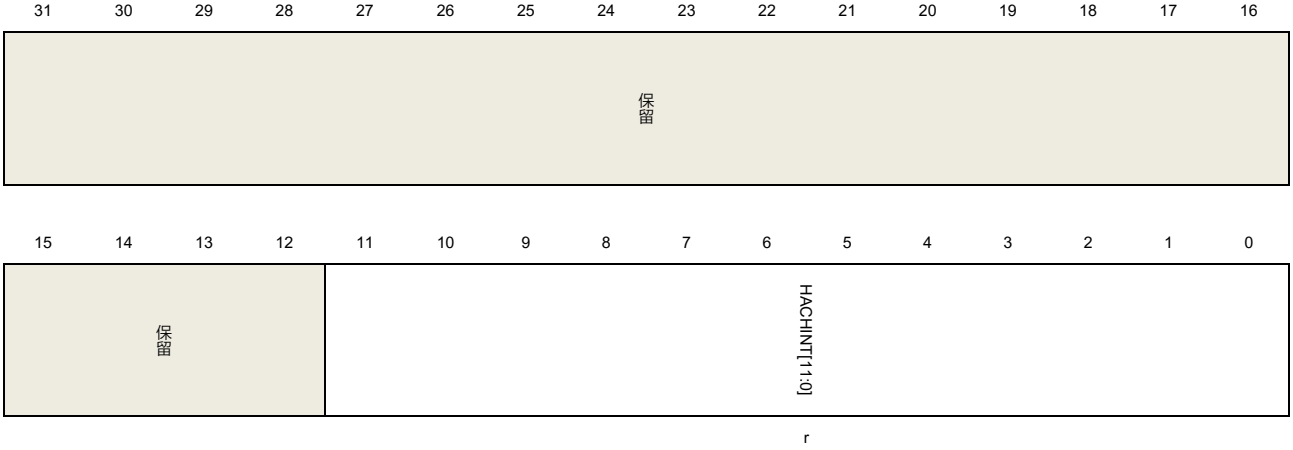
其他：保留

主机所有通道中断寄存器 (USBFS_HACHINT)

地址偏移：0x0414
复位值：0x0000 0000

当触发一个通道中断时，USBFS 在该寄存器中置位相应的位，并且软件可以读取该寄存器以获取产生中断的通道。

该寄存器只能按字(32 位)访问



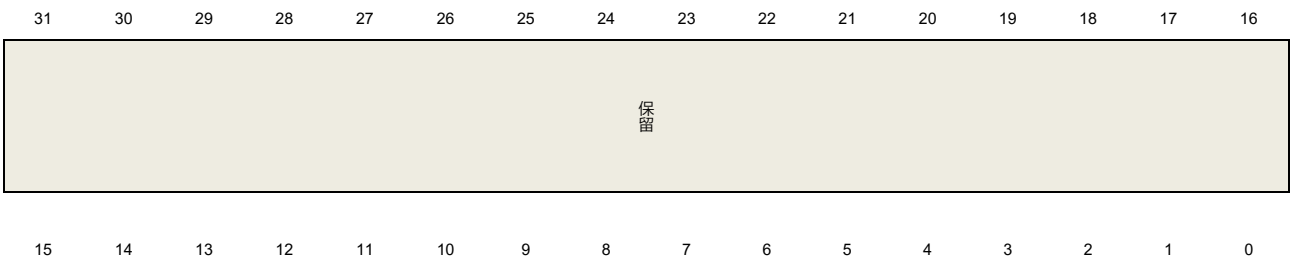
位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	HACHINT[11:0]	主机所有通道中断 每一位表示一个通道：位0代表通道0，位11表示通道11

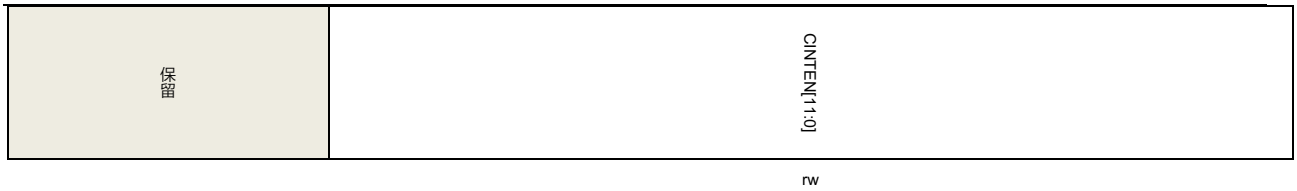
主机所有通道中断使能寄存器 (USBFS_HACHINTEN)

地址偏移：0x0418
复位值：0x0000 0000

软件可以使用该寄存器使能或禁用一个通道的中断。只有该寄存器中相应通道的中断使能控制位被置位，USBFS_GINTF 寄存器中的通道中断标志位 HCIF 标志位才可产生。

该寄存器只能按字(32 位)访问





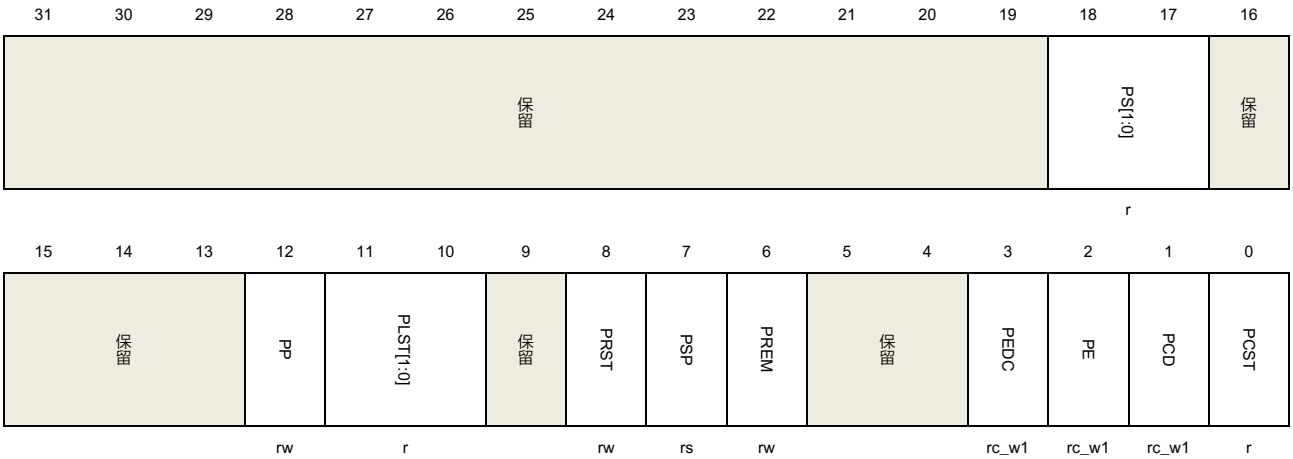
位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	CINTEN	通道中断使能 0：禁用通道n中断 1：使能通道n中断 每一位表示一个通道：位0代表通道0，位11代表通道11

主机端口控制和状态寄存器 (USBFS_HPCS)

地址偏移：0x0440
复位值：0x0000 0000

该寄存器控制端口行为，并且也包含一些反映端口状态的标志位。如果本寄存器中的 PRST、PEDC 和 PCD 标志位被 USBFS 置位的话，USBFS_GINTF 寄存器中的 HPIF 标志位会被置位。

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:19	保留	必须保持复位值。
18:17	PS[1:0]	端口速度 反映连接到该端口的设备的枚举速度。 01：全速 10：低速 其他：保留

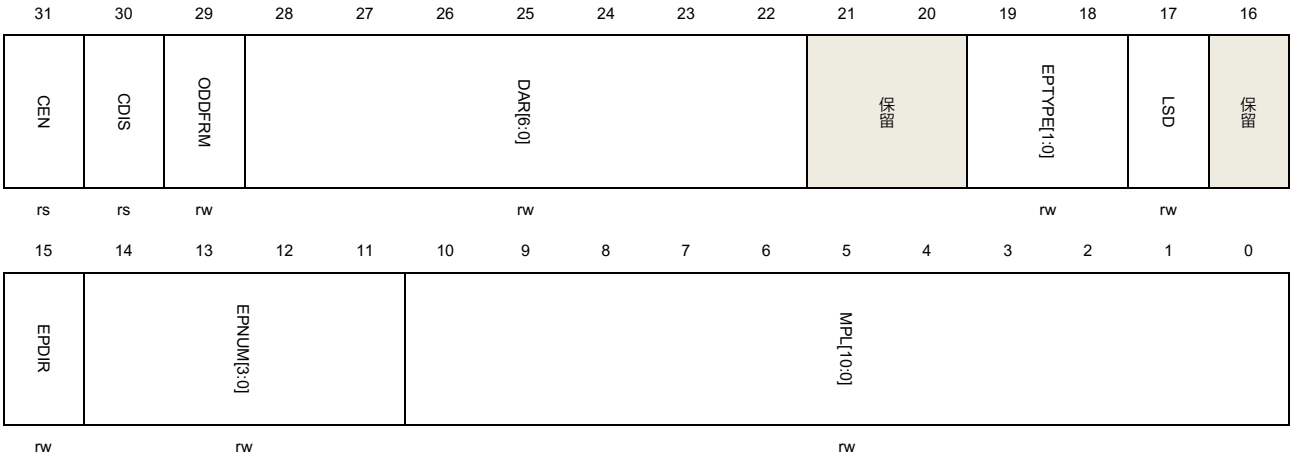
16:13	保留	必须保持复位值。
12	PP	端口供电 在端口被使用后，该控制位应该被置位。由于USBFS不具有电源供应能力，它只能使用该控制位以获取该端口是否在供电状态。软件应该在设置该控制位之前，保证在VBUS引脚上具有电源供应。 0：端口掉电 1：端口供电
11:10	PLST[1:0]	端口线状态 反映USB数据线当前状态 位10：DP线状态 位11：DM线状态
9	保留	必须保持复位值。
8	PRST	端口复位 应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0：端口不在复位状态 1：端口处于复位状态
7	PSP	端口挂起 应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
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，USBFS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，USBFS置位该标志位。可通过向该位写1清除该标志位。
0	PCST	端口连接状态 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 (USBFS_HCHxCTL) (x = 0…11，其中 x 是通道号)

地址偏移：0x0500 +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31	CEN	通道使能 由应用设置，并且由USBFS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[1:0]	设备地址 与该通道通信的USB设备地址。

21:20	保留	必须保持复位值。
19:18	EPTYPE	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 中断标志寄存器 (USBFS_HCHxINTF) (x = 0…11，其中 x 是通道号)

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	保留	ACK	NAK	STALL	保留	CH	TF
					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USBFS_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	保留	必须保持复位值。
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	保留	必须保持复位值。
1	CH	通道中止 通道被当前请求所禁用，在当前请求处理的过程中，并不响应其他请求处理。
0	TF	发送完成 该通道所有的事务成功完成并且无错误发生。 对于IN通道，在USBFS_HCHxLEN寄存器的PCNT位减到0后，该标志位被置位。 对于OUT通道，当软件从RxFIFO中读取和取出一个TF状态条目时，该标志位被置位。

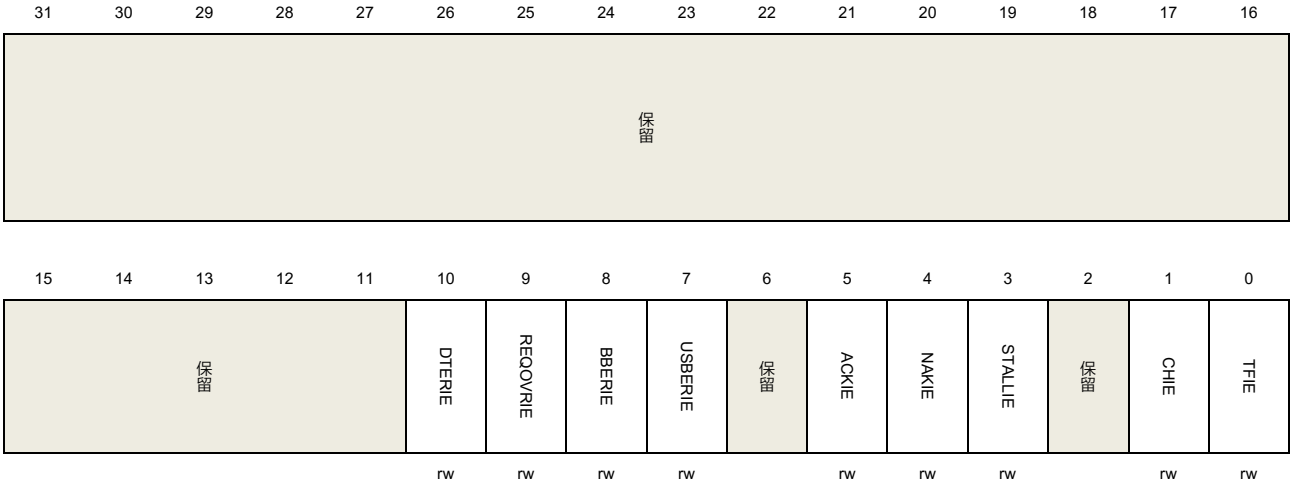
主机通道 x 中断使能寄存器 (USBFS_HCHxINTEN) (x = 0…11，其中 x 是通道号)

地址偏移：0x050C +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器包含 USBFS_HCHxINTF 寄存器内中断标志位的中断使能位。如果该寄存器的某位被软件置位，USBFS_HCHxINTF 寄存器内的相应位能够触发一个通道中断。该寄存器内的位

可由软件置位和清除。

该寄存器只能按字(32 位)访问



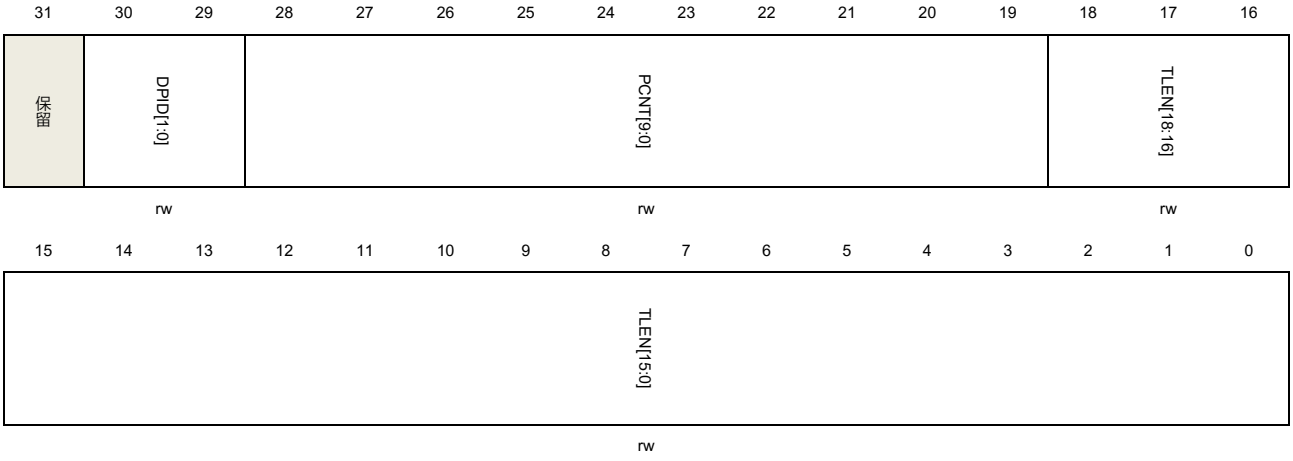
位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DTERIE	数据切换错误中断使能 0：禁用数据切换错误中断 1：使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0：禁用请求队列上溢中断 1：使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0：禁用串扰错误中断 1：使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0：禁用USB总线错误中断 1：使能USB总线错误中断
6	保留	必须保持复位值。
5	ACKIE	ACK中断使能 0：禁用ACK中断 1：使能ACK中断
4	NAKIE	NAK中断使能 0：禁用NAK中断 1：使能NAK中断
3	STALLIE	STALL中断使能 0：禁用STALL中断

		1：使能STALL中断
2	保留	必须保持复位值。
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 (USBFS_HCHxLEN) (x = 0…11，其中 x 是通道号)

地址偏移：0x0510 +(通道号 × 0x20)
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31	保留	必须保持复位值。
30:29	DPID[1:0]	数据PID 软件应该在传输起始之前写该段位域。对于OUT传输，该位域包含第一个传输包的数据PID。对于IN传输，该位域包含第一个接收包的数据PID，并且如果数据PID不匹配的话，将会触发DTER标志位。在传输开始之后，USBFS遵循USB协议自动改变和切换该位域。 00：DATA0 10：DATA1 11：SETUP(仅对于控制传输) 01：保留
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USBFS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。

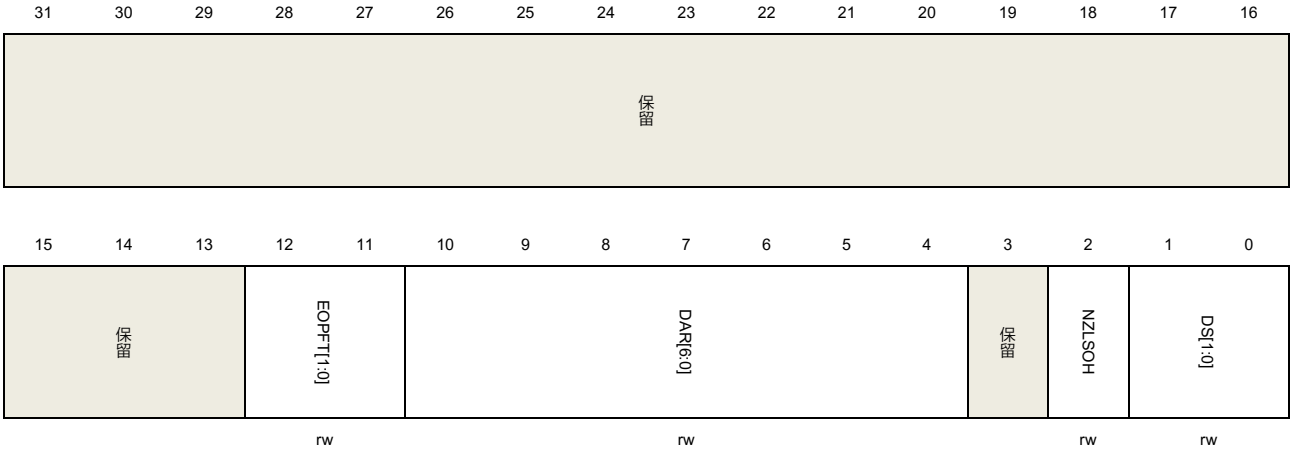
31.7.3. 设备控制和状态寄存器组

设备配置寄存器 (USBFS_DCFG)

地址偏移：0x0800
复位值：0x0000 0003

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



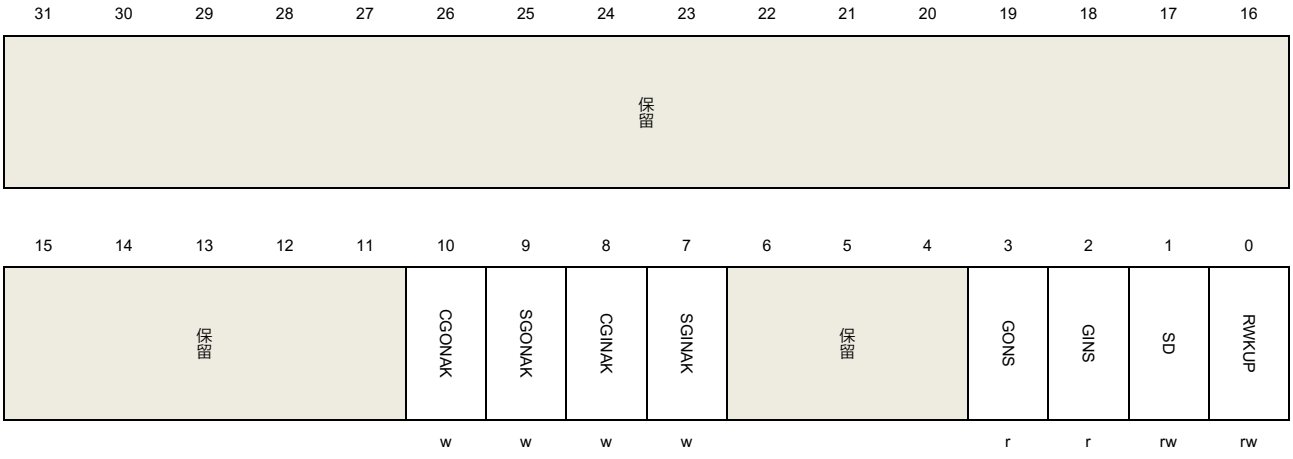
位/位域	名称	描述
31:13	保留	必须保留为复位值。
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00：80%的帧时间 01：85%的帧时间 10：90%的帧时间 11：95%的帧时间

10:4	DAR[6:0]	设备地址 该位定义USB设备地址，USBFS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域
3	保留	必须保留为复位值。
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USBFS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 11：全速 其他：保留

设备控制寄存器 (USBFS_DCTL)

地址偏移：0x0804
复位值：0x0000 0000

该寄存器采用字(32 位)访问



位/位域	名称	描述
31:11	保留	必须保留为复位值。
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USBFS_GINTF寄存器的GONAK标志触发，软件

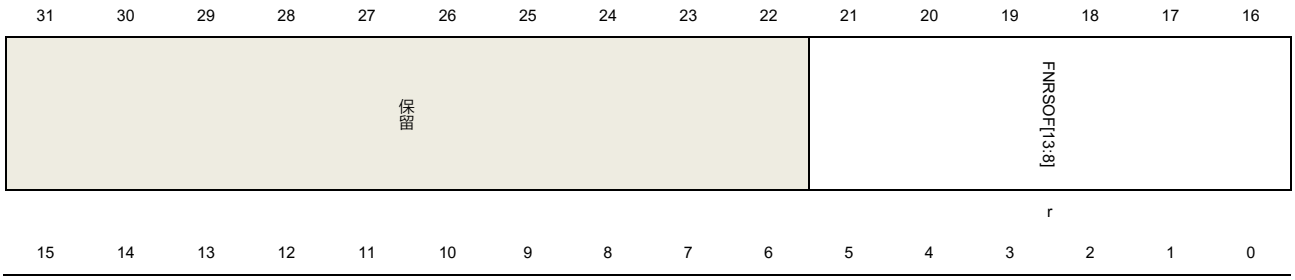
		应该在再写该位前清除GONAK标志。
8	CGINAK	清零全局IN NAK 软件设置该位从而清零该寄存器的GINS位
7	SGINAK	设置全局IN NAK 软件设置该位从而实现该寄存器的位GINS置位 当GINS位为零，设置该位会引起USBFS_GINTF寄存器的GINAK标志触发，软件应该在再写该位前清除GINAK标志。
6:4	保留	必须保留为复位值。
3	GONS	全局OUT NAK状态 0：USBFS回复OUT事务的握手信号以及是否保存OUT数据包由Rx FIFO状态、端点的NAKS、STALL位确定。 1：USBFS回复OUT事务NAK握手信号，不保存接收的OUT数据包。
2	GINS	全局IN NAK状态 0：USBFS回复IN事务的握手信号由Tx FIFO状态、端点的NAKS、STALL位确定。 1：USBFS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，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

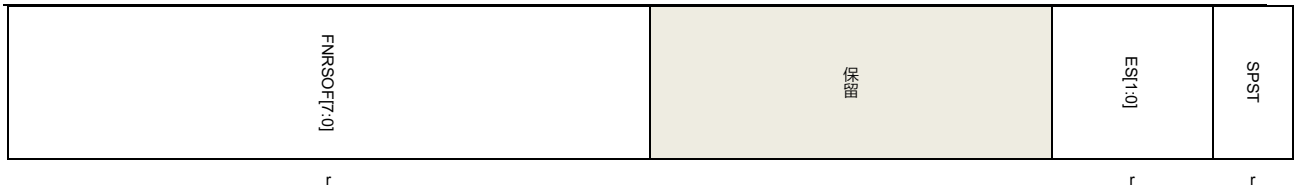
设备状态寄存器 (USBFS_DSTAT)

地址偏移：0x0808
复位值：0x0000 0000

该寄存器包含设备模式下的 USBFS 的状态和信息。

该寄存器采用字(32 位)访问





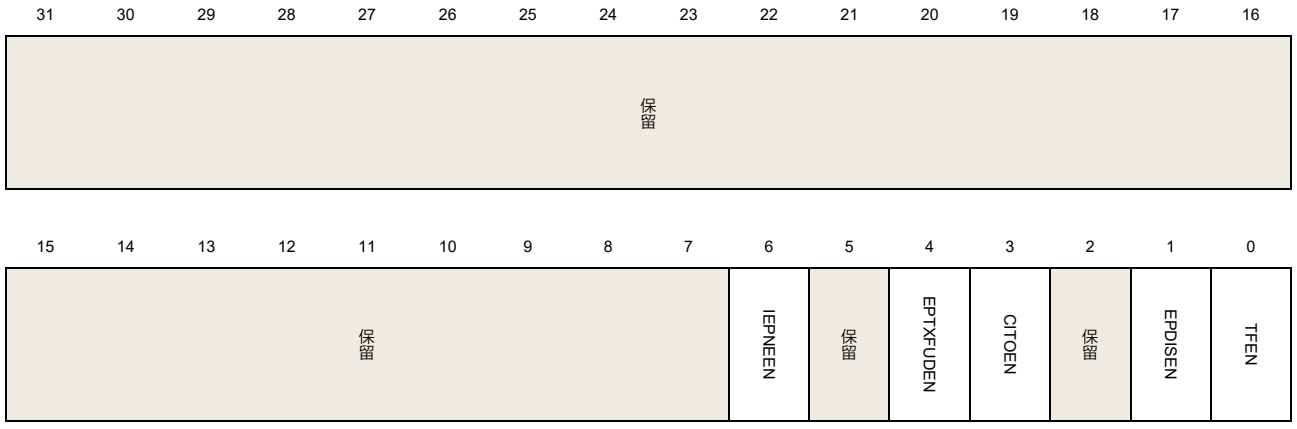
位/位域	名称	描述
31:22	保留	必须保留为复位值。
21:8	FNRSOF[13:0]	所接收的SOF帧编号 USBFS会在接收到一个SOF令牌后更新该域。
7:3	保留	必须保留为复位值。
2:1	ES[1:0]	枚举速度 该域指示所枚举的设备速度，在寄存器USBFS_GINTF的ENUMF标志触发后，软件可以读取该域。 01：全速 其他：保留
0	SPST	挂起状态 该位指示设备是否处于挂起状态。 0：设备不在挂起状态 1：设备在挂起状态

设备 IN 端点通用中断使能寄存器 (USBFS_DIEPINTEN)

地址偏移：0x810
复位值：0x0000 0000

该寄存器包含寄存器 USBFS_DIEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DIEPxINTF 中对应的位可以触发一个寄存器 USBFS_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



rw rw rw rw rw

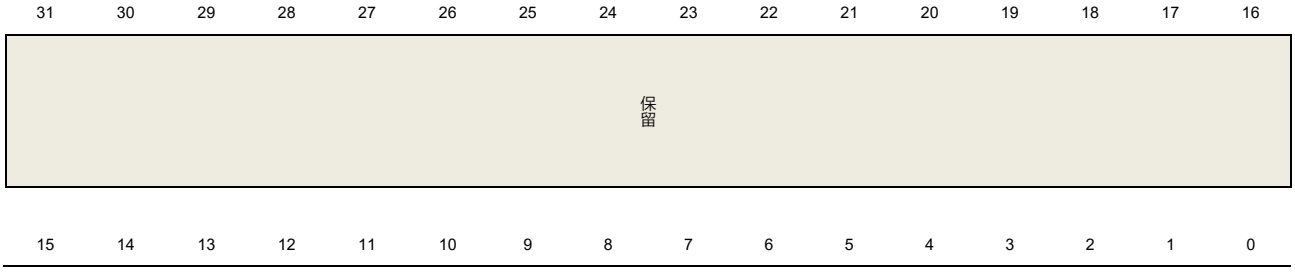
位/位域	名称	描述
31:7	保留	必须保留为复位值。
6	IEPNEEN	IN端点NAK有效中断使能位 0：除能中断 1：使能中断
5	保留	必须保留为复位值。
4	EPTXFUDEN	端点Tx FIFO下溢中断使能位 0：除能中断 1：使能中断
3	CITOEN	控制IN事务超时中断使能位 0：除能中断 1：使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

设备 OUT 端点通用中断使能寄存器 (USBFS_DOEPINTEN)

地址偏移：0x0814
复位值：0x0000 0000

该寄存器包含寄存器 USBFS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DOEPxINTF 中对应的位可以触发一个寄存器 USBFS_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



保留	BTBSTPEN	保留	EPRXFOVREN	STPFEN	保留	EPDISEN	TFEN
	rw		rw	rw		rw	rw

位/位域	名称	描述
31:7	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	OTEPDEN	端点失能时接收到OUT令牌中断使能位(仅适用于控制OUT端点) 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

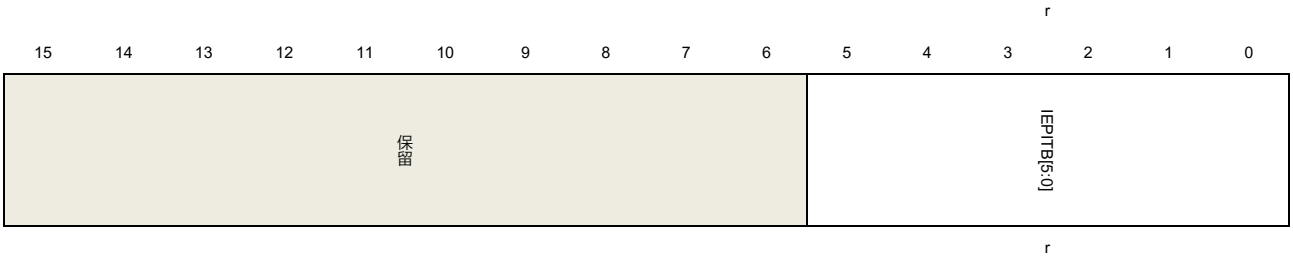
设备端点中断寄存器 (USBFS_DAEPINT)

地址偏移：0x0818
复位值：0x0000 0000

当一个端点的中断被触发，USBFS 置 1 该寄存器的相应位，软件可通过该寄存器知道在本次中断中的端点号。

该寄存器采用字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										OEPITB[5:0]					



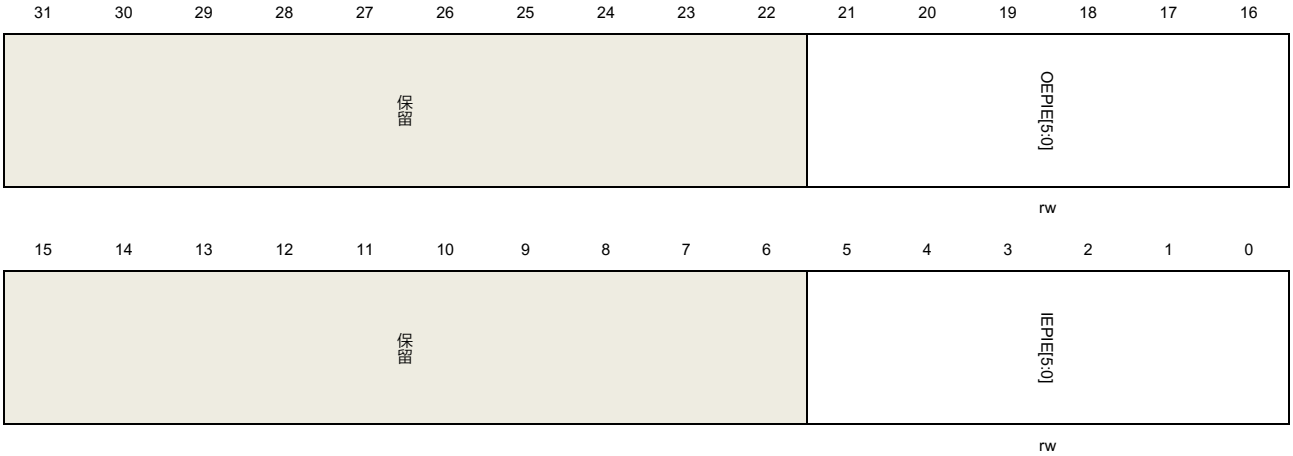
位/位域	名称	描述
31:22	保留	必须保留为复位值。
21:16	OEPITB[5:0]	设备OUT端点中断位 每个位代表一个OUT端点：Bit16代表OUT端点0，Bit21代表OUT端点5
15:6	保留	必须保留为复位值。
5:0	IEPITB[5:0]	设备IN端点中断位 每个位代表一个IN端点：Bit0代表IN端点0，Bit5代表IN端点5

设备端点中断使能寄存器 (USBFS_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 USBFS_GINTF 的端点中断标志 OEPIF 或 IEPIF。

该寄存器采用字(32 位)访问



位/位域	名称	描述
31:22	保留	必须保留为复位值。
21:16	OEPIE[5:0]	OUT端点中断使能位 0：除能OUT端点n中断

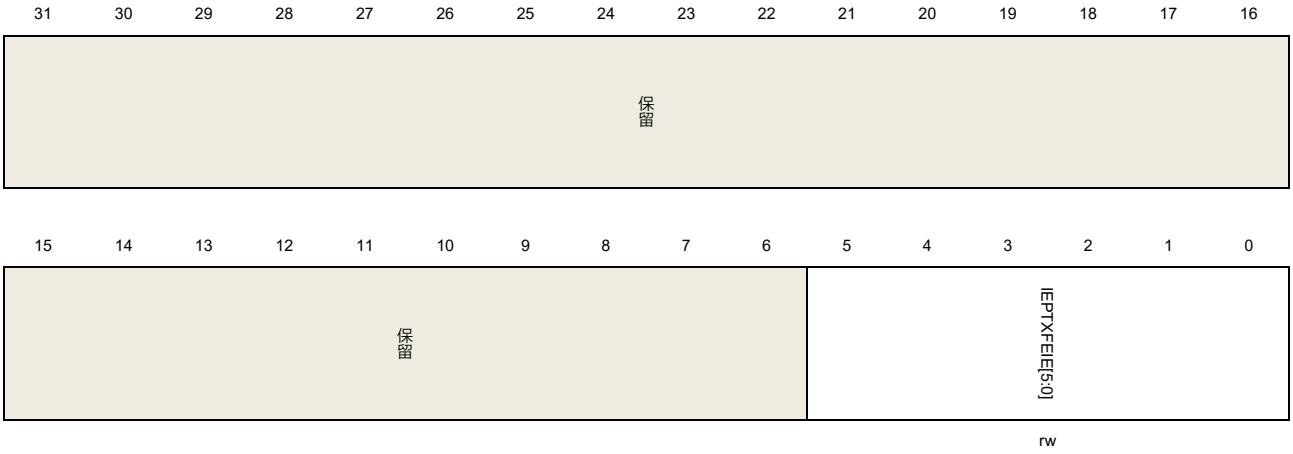
		1：使能OUT端点n中断 每个位代表一个OUT端点：Bit16对应OUT端点0，Bit21对应OUT端点5
15:6	保留	必须保留为复位值。
5:0	IEPIE[5:0]	IN端点中断使能位 0：除能IN端点n中断 1：使能IN端点n中断 每个位代表一个IN端点：Bit0对应IN端点0，Bit5对应IN端点5

设备 IN 端点 FIFO 空中断使能寄存器（USBFS_DIEPFEINTEN)

地址偏移：0x0834
复位值：0x0000 0000

该寄存器包含 IN 端点 Tx FIFO 空中断的使能位

寄存器采用字(32 位)访问

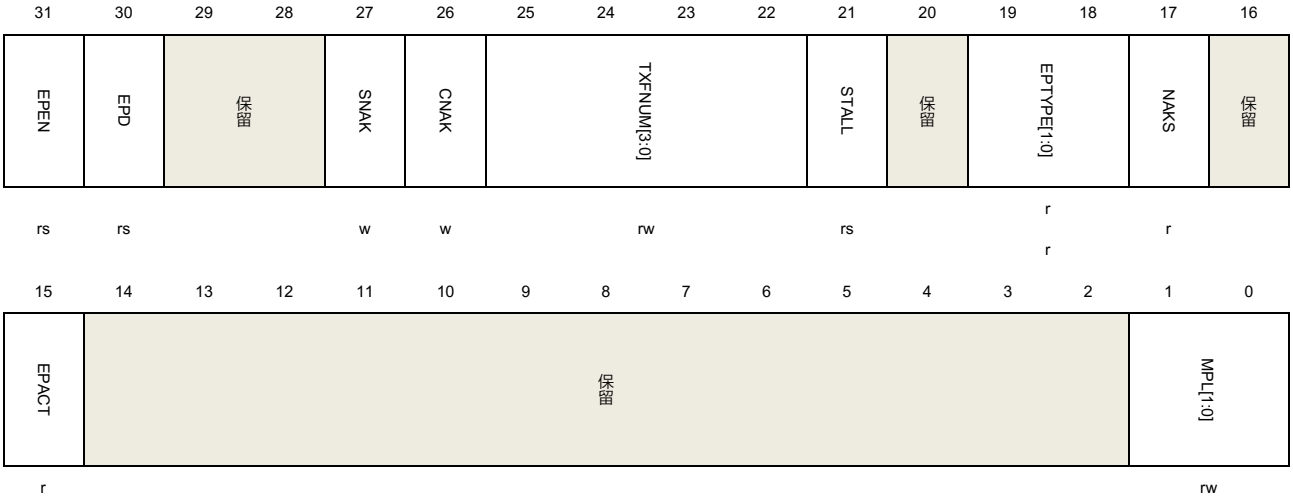


位/位域	名称	描述
31:6	保留	必须保留为复位值。
5:0	IEPTXFEIE[5:0]	IN端点Tx FIFO空中断的使能位 该域控制着USBFS_DIEPxINTF寄存器的TXFE位能否生成一个寄存器USBFS_DAEPINT的端点中断位 Bit0对应IN端点0，Bit5对应IN端点5 0：除能FIFO空中断 1：使能FIFO空中断

设备 IN 端点 0 控制寄存器（USBFS_DIEP0CTL)

地址偏移：0x0900
复位值：0x0000 8000

该寄存器采用字(32 位)访问



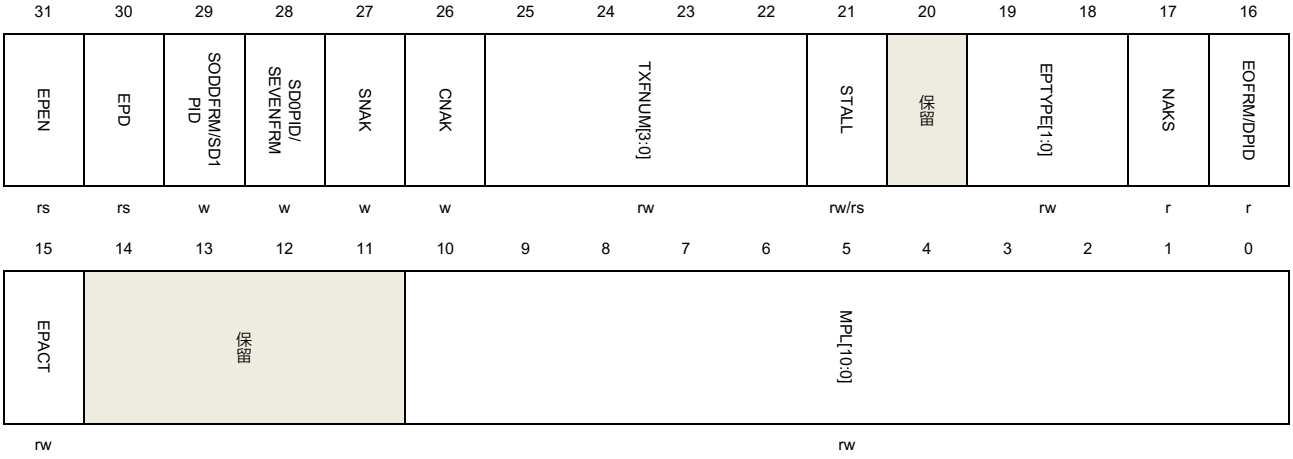
位/位域	名称	描述
31	EPEN	端点使能 软件置位、USBFS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29:28	保留	必须保留为复位值。
27	SNAK	置位NAK 软件置位该位来设置该寄存器的NAKS位
26	CNAK	清零NAK 软件置位该位来清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 定义IN端点0的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。
20	保留	必须保留为复位值。
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USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态。 0：根据端点Tx FIFO的状态，USBFS发送数据或握手包 1：USBFS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 (USBFS_DIEPxCTL) (x = 1…5, 其中 x 是端点编号)

地址偏移：0x0900 + (端点编号 x 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS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DATA1(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USBFS_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USBFS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USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态： 0：根据端点Tx FIFO的状态，USBFS发送数据或握手包

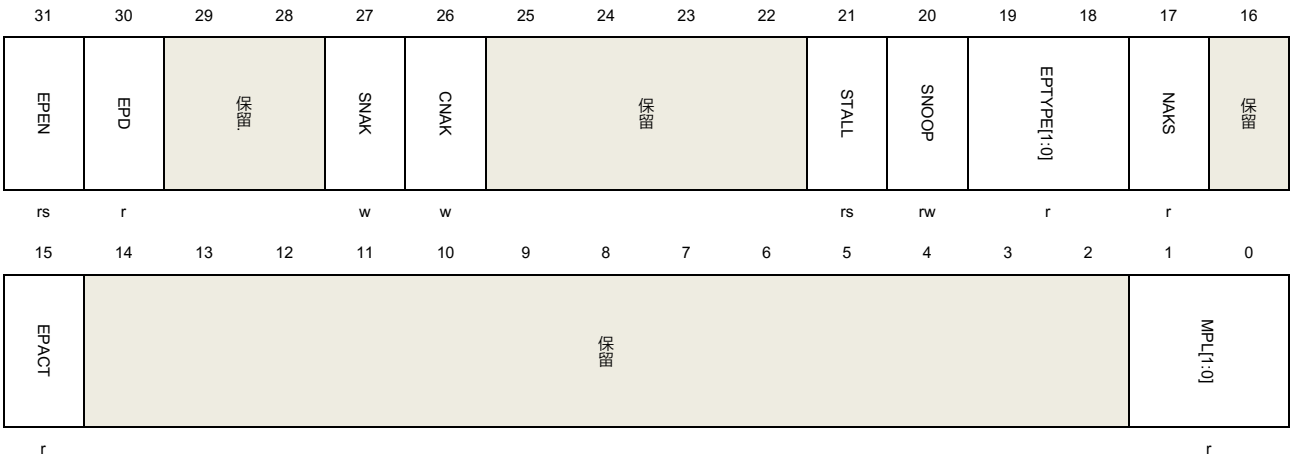
		1：USBFS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	EOFRM	奇偶帧(适用于同步IN端点) 对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBFS回复一个零长度的包： 0：只在偶数帧发送数据 1：只在奇数帧发送数据
	DPID	端点数据PID(适用于中断或大容量IN端点) 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。 0：数据包的PID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 (USBFS_DOEP0CTL)

地址偏移: 0x0B00

复位值: 0x0000 8000

该寄存器采用字(32 位)访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS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，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，即如果STALL和NAKS位都被置位，STALL位生效。
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值 0：调查模式除能 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USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态： 0：根据端点Rx FIFO的状态，USBFS发送数据或握手包 1：USBFS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USBFS_DIEP0CTL的位MPL： 00：64字节 01：32字节

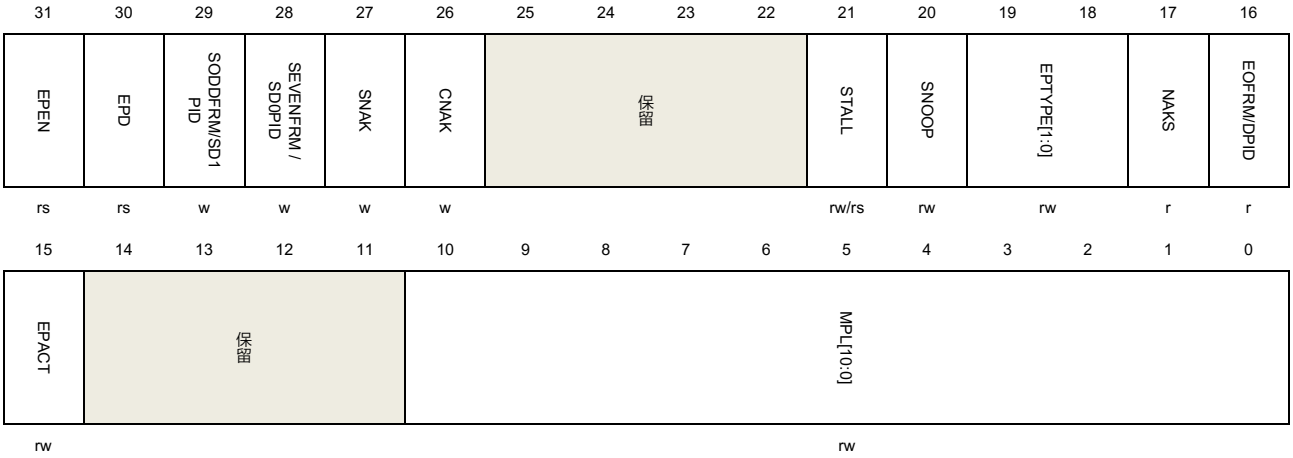
10：16字节
11：8字节

设备 OUT 端点 x 控制寄存器 (USBFS_DOEPxCTL) (x= 1…5，其中 x 是端点编号)

地址偏移：0x0B00 + (端点编号 x 0x20)
复位值：0x0000 0000

软件用该寄存器控制 OUT 端点 0 以外的每个逻辑 OUT 端点

该寄存器采用字(32 位)访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS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USBFS_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USBFS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，USBFS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USBFS_DCTL的位GONS被清零，该位控制USBFS的NAK状态： 0：根据端点的Rx FIFO的状态，发送握手包 1：USBFS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USBFS只在奇数帧或偶数帧发送数据包给OUT事务，如果当前帧号的奇偶性不匹配该位，USBFS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，USBFS在传输过程中保持该位。

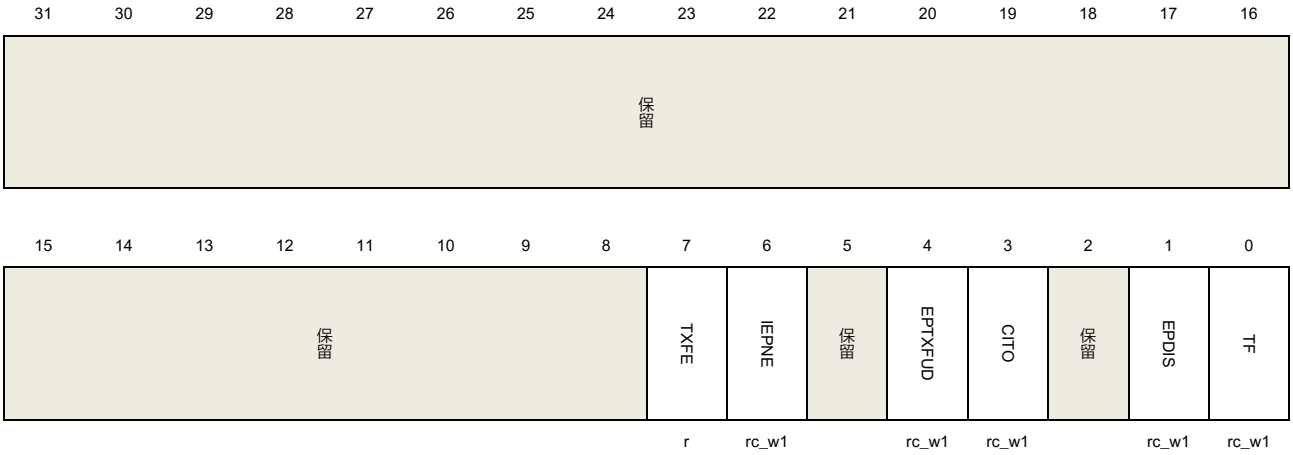
		0：数据包PID是DATA0 1：数据包PID是DATA1
15	EPACT	端点激活 位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复
14:11	保留	必须保留为复位值。
10:0	MPL[10:0]	该位定义最大包长

设备 IN 端点 x 中断标志寄存器 (USBFS_DIEPxINTF) (x = 0…5, 其中 x 是端点编号)

地址偏移：0x0908 + (端点编号 x 0x20)
复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当获得一个 IN 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，除了 TXFE 位，各位写 1 清零。

该寄存器采用字(32 位)访问



位/位域	名称	描述
31:8	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USBFS_GAHBCS的位TXFTH定义的空阈值。
6	IEPNE	IN端点NAK有效 寄存器USBFS_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

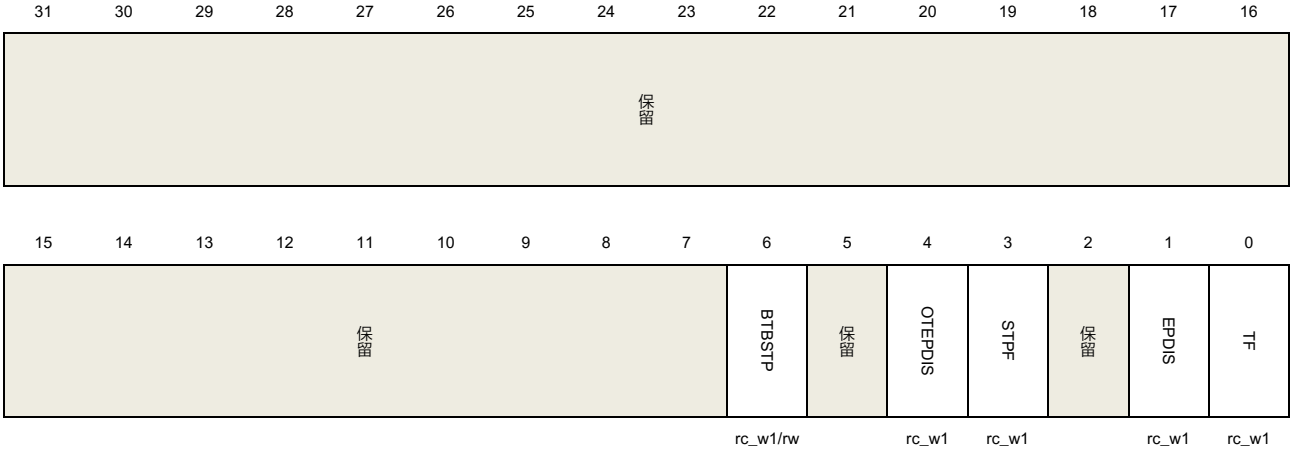
		在控制IN事务中，如果设备等待的握手包超时，该标志位被触发
2	保留	必须保留为复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

设备 OUT 端点 x 中断标志寄存器 (USBFS_DOEPxINTF) (x = 0…5, 其中 x 是端点编号)

地址偏移：0x0B08 + (端点编号 x 0x20)
复位值：0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当获得一个 OUT 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，各位写 1 清零。

该寄存器采用字(32 位)访问



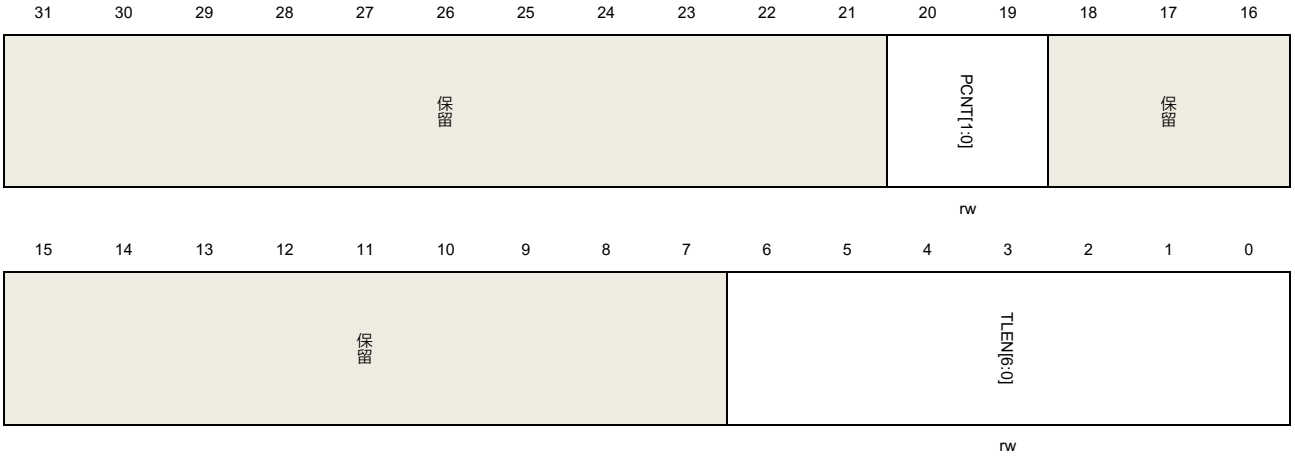
位/位域	名称	描述
31:7	保留	必须保留为复位值。
6	BTBSTP	连续SETUP包(适用于控制OUT端点) 当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保留为复位值。
4	OTEPD	端点失能时接收到OUT令牌(适用于控制OUT端点) 当端点未使能时，如果接收到OUT令牌，就会触发一个中断。
3	STPF	SETUP阶段完成(适用于控制OUT端点) 当一个SETUP阶段完成，也就是USBFS在一个setup令牌后接收了一个IN或OUT令

		牌，该位被置位。
2	保留	必须保留为复位值。
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 (USBFS_DIEP0LEN)

地址偏移：0x0910
复位值：0x0000 0000

该寄存器采用字(32 位)访问

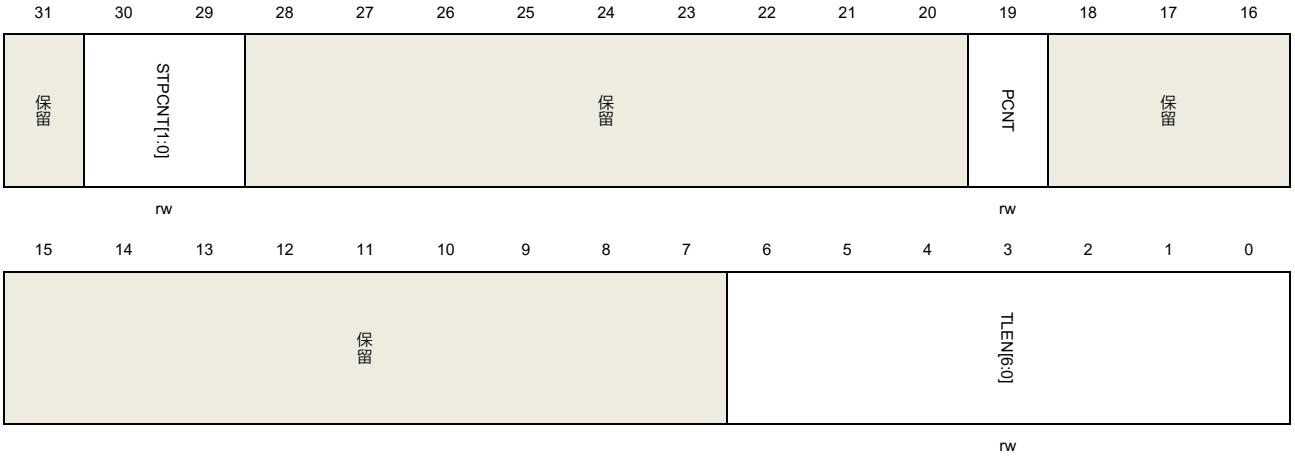


位/位域	名称	描述
31:21	保留	必须保留为复位值。
20:19	PCNT[1:0]	包数 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	传输长度 一次传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 0 传输长度寄存器 (USBFS_DOEP0LEN)

地址偏移：0x0B10
复位值：0x0000 0000

该寄存器采用字(32 位)访问

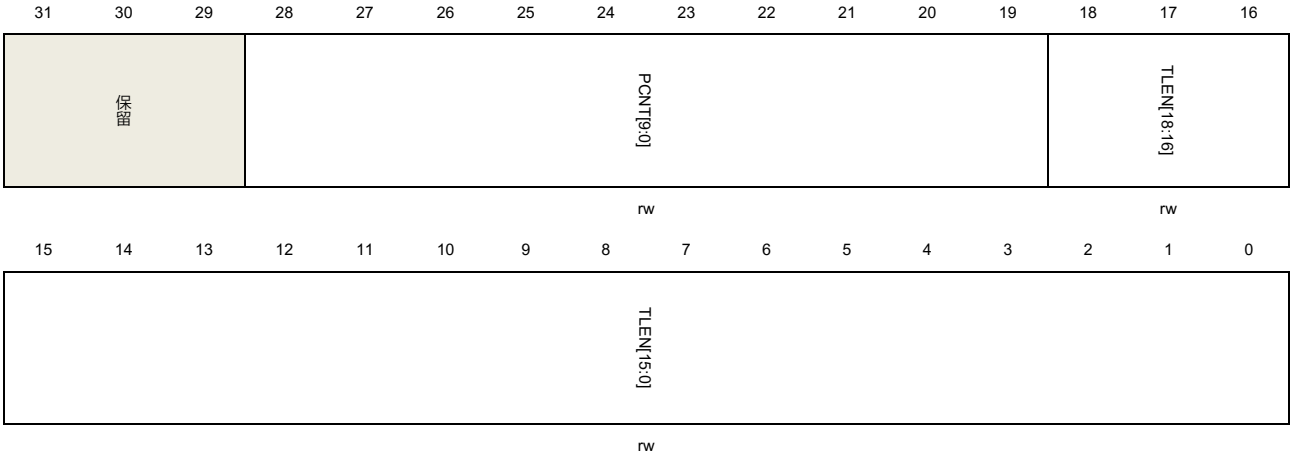


位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	STPCNT[1:0]	SETUP包计数 该域定义端点可以接受的最大连续SETUP包数量 在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTP标志被触发。 00：0个包 01：1个包 10：2个包 11：3个包
28:20	保留	必须保留为复位值。
19	PCNT	包计数 一次传输中应该接收到包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	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 (USBFS_DIEPxLEN) (x = 1…5, 其中 x 是端点编号)

地址偏移：0x910 + (端点编号 x 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问

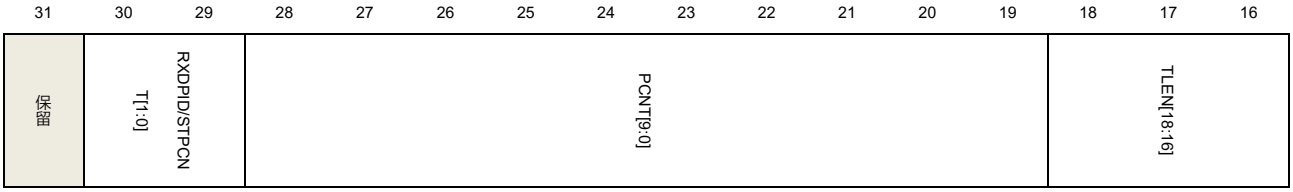


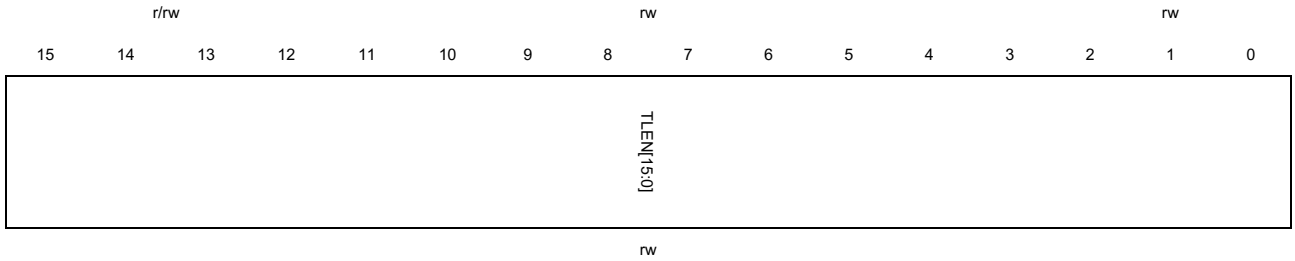
位/位域	名称	描述
31:29	保留	必须保留为复位值。
28:19	PCNT[9:0]	包数量 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 x 传输长度寄存器 (USBFS_DOEPxLEN) (x = 1…5, x 是端点编号)

地址偏移：0x0B10 + (端点编号 x 0x20)
复位值：0x0000 0000

该寄存器采用字(32 位)访问





位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	RXDPID[1:0]	接收数据PID(适用于同步OUT端点) 该域保存该端点该数据包所接受的最后一个数据包的PID 00：DATA0 10：DATA1 其他：保留 SETUP包数(适用于控制OUT端点)
	STPCNT[1:0]	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USBFS_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 发送 FIFO 状态寄存器 (USBFS_DIEPxTFSTAT) (x = 0…5, x 是端点编号)

地址偏移：0x0918 + (端点编号 x 0x20)
复位值：0x0000 0080

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字(32 位)访问



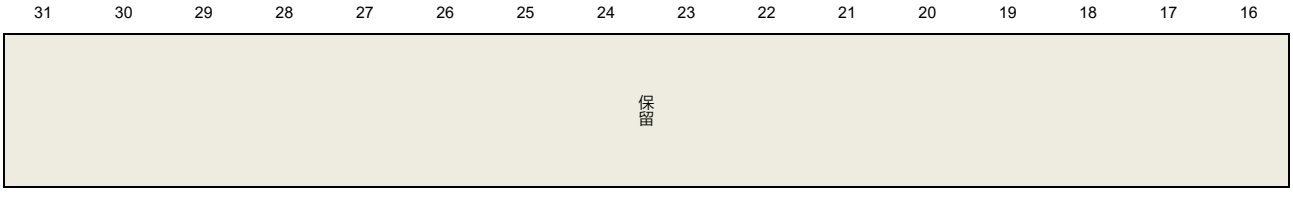


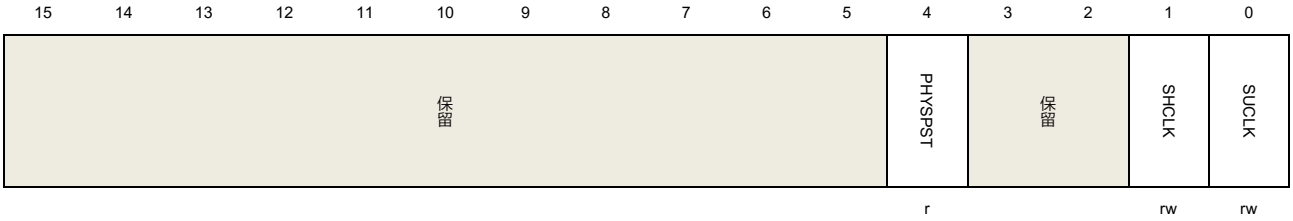
位/位域	名称	描述
31:16	保留	必须保留为复位值。
15:0	IEPTFS[15:0]	IN端点的Tx FIFO可用空间 IN端点的Tx FIFO可用空间用32位字为单位 0：FIFO是满的 1：1字可用 … n：n字可用 **注意：**该复位值和Tx FIFO编号有关，这里的复位情况是Tx FIFO0。

31.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL)

地址偏移：0x0E00
复位值：0x0000 0000

该寄存器采用字(32 位)访问





位/位域	名称	描述

31:5	保留	必须保留为复位值。
4	PHYSPST	挂起状态 该位表明PHY是否处于挂起状态 0：PHY不是挂起状态 1：PHY是挂起状态
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

32. 附录

32.1. 寄存器表中使用的缩写列表

表 32-1. 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。

32.2. 术语表

表 32-2. 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

32.3. 可用外设

对于各个 MCU 系列的外设及其数量，请参考相应型号的数据手册。

33. 版本历史

表 33-1. 版本历史

版本号	描述	日期
1.0	初始发布	2026 年 11 月 17 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. the Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, aeronautic or aerospace applications, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); and/or (iii) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

While the Company has implemented advanced security features, the Product may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on Customer's applications and products, and to the maximum extent permitted by applicable law, the Company accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved