GigaDevice Semiconductor Inc.

GD32F20x
Arm® Cortex®-M3 32-bit MCU

用户手册

2.7 版本

（2022 年 6 月）

目 录

目 录 2
图索引 20
表索引 28
1. 系统及存储器架构 32
1.1. Arm Cortex-M3 内核 32
1.2. 系统架构 33
1.3. 存储器映射 34
1.3.1. 位带操作 38
1.3.2. 片上 SRAM 39
1.3.3. 片上 Flash 39
1.4. 启动(Boot)选项配置 39
1.5. 设备电子签名 40
1.5.1. 产品容量信息 40
1.5.2. 器件唯一 ID（96 位） 41
1.6. 系统配置寄存器 42
2. 闪存控制器（FMC） 43
2.1. 简介 43
2.2. 主要特征 43
2.3. 功能说明 43
2.3.1. 闪存结构 43
2.3.2. 读操作 44
2.3.3. FMC_CTLx 寄存器解锁 44
2.3.4. 页擦除 44
2.3.5. 整片擦除 45
2.3.6. 主存储闪存块编程 47
2.3.7. 选项字节擦除 48
2.3.8. 选项字节块编程 48
2.3.9. 选项字节块说明 48
2.3.10. 页擦除/编程保护 50
2.3.11. 安全保护 50
2.4. FMC 寄存器 51
2.4.1. 等待状态寄存器（FMC_WS） 51
2.4.2. 解锁寄存器 0（FMC_KEY0） 51
2.4.3. 选项字节操作解锁寄存器（FMC_OBKEY） 52
2.4.4. 状态寄存器 0（FMC_STAT0） 52

2.4.5. 控制寄存器 0（FMC_CTL0）.....53
2.4.6. 地址寄存器 0（FMC_ADDR0）.....54
2.4.7. 选项字节状态寄存器（FMC_OBSTAT）.....54
2.4.8. 擦除/编程保护寄存器（FMC_WP）.....55
2.4.9. 解锁寄存器 1（FMC_KEY1）.....55
2.4.10. 状态寄存器 1（FMC_STAT1）.....56
2.4.11. 控制寄存器 1（FMC_CTL1）.....57
2.4.12. 地址寄存器 1（FMC_ADDR1）.....58
2.4.13. 等待状态使能寄存器（FMC_WSEN）.....58
2.4.14. 产品 ID 寄存器（FMC_PID）.....59

3. 电源管理单元（PMU）.....60

3.1. 简介.....60

3.2. 主要特征.....60

3.3. 功能说明.....60

3.3.1. 电池备份域.....61
3.3.2. VDD / VDDA 电源域.....62
3.3.3. 1.2V 电源域.....63
3.3.4. 省电模式.....64

3.4. PMU 寄存器.....66

3.4.1. 控制寄存器（PMU_CTL）.....66
3.4.2. 电源控制和状态寄存器（PMU_CS）.....67

4. 备份寄存器（BKP）.....69

4.1. 简介.....69

4.2. 主要特征.....69

4.3. 功能说明.....69

4.3.1. RTC 时钟校准.....69
4.3.2. 侵入检测 0.....69
4.3.3. 侵入检测 1.....70
4.3.4. 方波检测.....70

4.4. BKP 寄存器.....71

4.4.1. 备份数据寄存器（BKP_DATAx）（x= 0..41）.....71
4.4.2. RTC 信号输出控制寄存器（BKP_OCTL）.....71
4.4.3. 侵入引脚控制寄存器 0（BKP_TPCTL0）.....72
4.4.4. 侵入控制状态寄存器（BKP_TPCS）.....73
4.4.5. 侵入引脚控制寄存器 1（BKP_TPCTL1）.....74

5. 复位和时钟单元（RCU）.....76

5.1. 复位控制单元（RCTL）.....76

5.1.1. 简介.....76
5.1.2. 功能说明.....76

5.2. 时钟控制单元（CCTL） 77
5.2.1. 简介 77
5.2.2. 主要特征 79
5.2.3. 功能说明 79
5.3. RCU 寄存器 84
5.3.1. 控制寄存器（RCU_CTL） 84
5.3.2. 配置寄存器 0（RCU_CFG0） 86
5.3.3. 时钟中断寄存器（RCU_INT） 89
5.3.4. APB2 复位寄存器（RCU_APB2RST） 92
5.3.5. APB1 复位寄存器（RCU_APB1RST） 94
5.3.6. AHB1 使能寄存器（RCU_AHB1EN） 97
5.3.7. APB2 使能寄存器（RCU_APB2EN） 99
5.3.8. APB1 使能寄存器（RCU_APB1EN） 101
5.3.9. 备份域控制寄存器（RCU_BDCTL） 104
5.3.10. 复位源/时钟寄存器（RCU_RSTSCK） 106
5.3.11. AHB1 复位寄存器（RCU_AHB1RST） 107
5.3.12. 配置寄存器 1（RCU_CFG1） 108
5.3.13. 深度睡眠模式电压寄存器（RCU_DSV） 110
5.3.14. AHB2 使能寄存器（RCU_AHB2EN） 111
5.3.15. APB2 附加使能寄存器（RCU_ADDAPB2EN） 112
5.3.16. APB1 附加使能寄存器（RCU_ADDAPB1EN） 112
5.3.17. AHB2 复位寄存器（RCU_AHB2RST） 113
5.3.18. APB2 附加复位寄存器（RCU_ADDAPB2RST） 114
5.3.19. APB1 附加复位寄存器（RCU_ADDAPB1RST） 115
5.3.20. 配置寄存器 2（RCU_CFG2） 116
5.3.21. PLLT 控制寄存器（RCU_PLLTCTL） 117
5.3.22. PLLT 中断寄存器（RCU_PLLTINT） 117
5.3.23. PLLT 配置寄存器（RCU_PLLTCFG） 118
6. 中断/事件控制器（EXTI） 121
6.1. 简介 121
6.2. 主要特性 121
6.3. 中断功能描述 121
6.4. 外部中断及事件(EXTI)框图 125
6.5. 外部中断及事件功能概述 125
6.6. EXTI 寄存器 127
6.6.1. 中断使能寄存器 (EXTI_INTEN) 127
6.6.2. 事件使能寄存器 (EXTI_EVEN) 127
6.6.3. 上升沿触发使能寄存器 (EXTI_RTEN) 128
6.6.4. 下降沿触发使能寄存器 (EXTI_FTEN) 128
6.6.5. 软件中断事件寄存器 (EXTI_SWIEV) 128
6.6.6. 挂起寄存器 (EXTI_PD) 129

7. 通用和备用输入/输出接口（GPIO 和 AFIO）..... 130

7.1. 简介 130

7.2. 主要特征 130

7.3. 功能说明 130

7.3.1. GPIO 引脚配置 131
7.3.2. 外部中断/事件线 132
7.3.3. 备用功能（AF） 132
7.3.4. 输入配置 132
7.3.5. 输出配置 133
7.3.6. 模拟配置 133
7.3.7. 备用功能（AF）配置 134
7.3.8. IO 引脚功能选择 135
7.3.9. GPIO 锁定功能 135

7.4. I/O 重映射功能和调试配置 136

7.4.1. 介绍 136
7.4.2. 主要特性 136
7.4.3. JTAG/SWD 备用功能重映射 136
7.4.4. ADC AF 重映射 137
7.4.5. TIMER AF 重映射 138
7.4.6. USART AF 重映射 139
7.4.7. I2C 备用功能重映射 140
7.4.8. SPI 备用功能重映射 140
7.4.9. CAN 备用功能重映射 141
7.4.10. ENET 备用功能重映射 142
7.4.11. DCI 备用功能重映射 143
7.4.12. TLI 备用功能重映射 143
7.4.13. CLK 引脚 AF 重映射 144

7.5. GPIO 寄存器 146

7.5.1. 端口控制寄存器 0（GPIOx_CTL0，x=A..I） 146
7.5.2. 端口控制寄存器 1（GPIOx_CTL1，x=A..I） 148
7.5.3. 端口输入状态寄存器（GPIOx_ISTAT，x=A..I） 149
7.5.4. 端口输出控制寄存器（GPIOx_OCTL，x=A..I） 150
7.5.5. 端口位操作寄存器（GPIOx_BOP，x=A..I） 150
7.5.6. 位清除寄存器（GPIOx_BC，x=A..I） 151
7.5.7. 端口配置锁定寄存器（GPIOx_LOCK，x=A..I） 152
7.5.8. 事件控制寄存器（AFIO_EC） 152
7.5.9. AFIO 端口配置寄存器 0（AFIO_PCF0） 153
7.5.10. EXTI 源选择寄存器 0 寄存器（AFIO_EXTISS0） 157
7.5.11. EXTI 源选择寄存器 1 寄存器（AFIO_EXTISS1） 158
7.5.12. EXTI 源选择寄存器 2 寄存器（AFIO_EXTISS2） 160
7.5.13. EXTI 源选择寄存器 3 寄存器（AFIO_EXTISS3） 161
7.5.14. AFIO 端口配置寄存器 1（AFIO_PCF1） 163

7.5.15. AFIO 端口配置寄存器 2（AFIO_PCF2）.....164
7.5.16. AFIO 端口配置寄存器 3（AFIO_PCF3）.....167
7.5.17. AFIO 端口配置寄存器 4（AFIO_PCF4）.....170
7.5.18. AFIO 端口配置寄存器 5（AFIO_PCF5）.....173

8. 循环冗余校验管理单元（CRC）.....178

8.1. 简介.....178

8.2. 主要特征.....178

8.3. 功能说明.....179

8.4. CRC 寄存器.....180

8.4.1. 数据寄存器（CRC_DATA）.....180
8.4.2. 独立数据寄存器（CRC_FDATA）.....180
8.4.3. 控制寄存器 (CRC_CTL).....181

9. 真随机数生成器（TRNG）.....182

9.1. 简介.....182

9.2. 主要特征.....182

9.3. 功能说明.....182

9.3.1. 操作流程.....183
9.3.2. 错误标志.....183

9.4. TRNG 寄存器.....184

9.4.1. 控制寄存器（TRNG_CTL）.....184
9.4.2. 状态寄存器（TRNG_STAT）.....184
9.4.3. 数据寄存器（TRNG_DATA）.....185

10. 加密处理器（CAU）.....187

10.1. 简介.....187

10.2. 主要特征.....187

10.3. CAU 数据类型和初始化向量.....188

10.3.1. 数据类型.....188
10.3.2. 初始化向量.....189

10.4. 加密处理器流程.....189

10.4.1. DES/TDES 加密处理流程.....190
10.4.2. AES 加密处理流程.....194

10.5. 操作模式.....198

10.6. CAU DMA 接口.....199

10.7. CAU 中断.....199

10.8. CAU 挂起模式.....200

10.9. CAU 寄存器.....201

10.9.1. CAU 控制寄存器 (CAU_CTL) 201
10.9.2. CAU 状态寄存器 0 (CAU_STAT0) 202
10.9.3. CAU 数据输入寄存器 (CAU_DI) 203
10.9.4. CAU 数据输出寄存器 (CAU_DO) 203
10.9.5. CAU DMA 使能寄存器 (CAU_DMAEN) 204
10.9.6. CAU 中断使能寄存器 (CAU_INTEN) 204
10.9.7. CAU 状态寄存器 1 (CAU_STAT1) 205
10.9.8. CAU 中断标志寄存器 (CAU_INTF) 205
10.9.9. CAU 密钥寄存器 (CAU_KEY0..3(H/L)) 206
10.9.10. CAU 初始化向量寄存器 (CAU_IV0..1(H/L)) 209

11. 哈希处理器（HAU） 211

11.1. 简介 211

11.2. 主要特征 211

11.3. 数据类型 211

11.4. HAU 内核 213

11.4.1. 自动数据填充 213
11.4.2. 摘要计算 214
11.4.3. 哈希模式 214
11.4.4. HMAC 模式 215

11.5. HAU 中断 215

11.6. HAU 寄存器 216

11.6.1. HAU 控制寄存器 (HAU_CTL) 216
11.6.2. HAU 数据输入寄存器 (HAU_DI) 217
11.6.3. HAU 配置寄存器 (HAU_CFG) 218
11.6.4. HAU 数据输出寄存器 (HAU_DO0..7) 218
11.6.5. HAU 中断使能寄存器 (HAU_INTEN) 220
11.6.6. HAU 状态与中断标志寄存器 (HAU_STAT) 221

12. 直接存储器访问控制器（DMA） 223

12.1. 简介 223

12.2. 主要特征 223

12.3. 结构框图 224

12.4. 功能说明 224

12.4.1. DMA 操作 224
12.4.2. 外设握手 226
12.4.3. 仲裁 227
12.4.4. 地址生成 227
12.4.5. 循环模式 227
12.4.6. 存储器到存储器模式 227
12.4.7. 通道配置 228

12.4.8. 中断 228
12.4.9. DMA 请求映射 229
12.5. DMA 寄存器 233
12.5.1. 中断标志位寄存器（DMA_INTF） 233
12.5.2. 中断标志位清除寄存器（DMA_INTC） 233
12.5.3. 通道 x 控制寄存器（DMA_CHxCTL，x = 0…6） 234
12.5.4. 通道 x 计数寄存器（DMA_CHxCNT，x = 0…6） 236
12.5.5. 通道 x 外设基地址寄存器（DMA_CHxPADDR，x = 0…6） 237
12.5.6. 通道 x 存储器基地址寄存器（DMA_CHxMADDR，x = 0…6） 237
12.5.7. DMA 附加配置寄存器（DMA_ACFG） 237
13. 调试（DBG） 239
13.1. 简介 239
13.2. JTAG/SW 功能描述 239
13.2.1. 切换 JTAG/ SW 接口 239
13.2.2. 引脚分配 239
13.2.3. JTAG 链状结构 240
13.2.4. 调试复位 240
13.2.5. JEDEC-106 ID code 240
13.3. 调试保持功能描述 240
13.3.1. 低功耗模式调试支持 240
13.3.2. TIMER，I2C，WWDGT，FWDGT 和 CAN 外设调试支持 241
13.4. DBG 寄存器 242
13.4.1. ID 寄存器（DBG_ID） 242
13.4.2. 控制寄存器（DBG_CTL） 242
14. 模数转换器（ADC） 246
14.1. 简介 246
14.2. 主要特征 246
14.3. 引脚和内部信号 247
14.4. 功能说明 248
14.4.1. 前置校准功能 248
14.4.2. ADC 时钟 249
14.4.3. ADCON 使能 249
14.4.4. 常规序列 249
14.4.5. 运行模式 249
14.4.6. 转换结果阈值监测功能 252
14.4.7. 数据存储模式 252
14.4.8. 采样时间配置 253
14.4.9. 外部触发配置 253
14.4.10. DMA 请求 254

14.4.11. ADC 内部通道 254
14.4.12. 可编程分辨率(DRES) 255
14.4.13. 片上硬件过采样 255

14.5. ADC 同步模式 257
14.5.1. 独立模式 258
14.5.2. 常规并行模式 258
14.5.3. 常规快速交叉模式 259
14.5.4. 常规慢速交叉模式 259

14.6. 中断 260

14.7. ADC 寄存器 261
14.7.1. 状态寄存器（ADC_STAT） 261
14.7.2. 控制寄存器 0（ADC_CTL0） 261
14.7.3. 控制寄存器 1（ADC_CTL1） 263
14.7.4. 采样时间寄存器 0（ADC_SAMPT0） 265
14.7.5. 采样时间寄存器 1（ADC_SAMPT1） 266
14.7.6. 看门狗高阈值寄存器（ADC_WDHT） 267
14.7.7. 看门狗低阈值寄存器（ADC_WDLT） 267
14.7.8. 常规序列寄存器 0（ADC_RSQ0） 268
14.7.9. 常规序列寄存器 1（ADC_RSQ1） 268
14.7.10. 常规序列寄存器 2（ADC_RSQ2） 269
14.7.11. 常规数据寄存器（ADC_RDATA） 269
14.7.12. 过采样控制寄存器（ADC_OVSAMPCTL） 270

15. 数模转换器（DAC） 272

15.1. 简介 272

15.2. 主要特征 272

15.3. 功能说明 273
15.3.1. DAC 使能 273
15.3.2. DAC 输出缓冲 273
15.3.3. DAC 数据配置 273
15.3.4. DAC 触发 273
15.3.5. DAC 工作流程 274
15.3.6. DAC 噪声波 274
15.3.7. DAC 输出计算 275
15.3.8. DMA 功能 275
15.3.9. DAC 并发转换 275

15.4. DAC 寄存器 276
15.4.1. 控制寄存器 (DAC_CTL) 276
15.4.2. 软件触发寄存器 (DAC_SWT) 278
15.4.3. DAC0 12 位右对齐数据保持寄存器 (DAC0_R12DH) 279
15.4.4. DAC0 12 位左对齐数据保持寄存器 (DAC0_L12DH) 279
15.4.5. DAC0 8 位右对齐数据保持寄存器 (DAC0_R8DH) 280

15.4.6. DAC1 12 位右对齐数据保持寄存器 (DAC1_R12DH) 280
15.4.7. DAC1 12 位左对齐数据保持寄存器 (DAC1_L12DH) 281
15.4.8. DAC1 8 位右对齐数据保持寄存器 (DAC1_R8DH) 281
15.4.9. DAC 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH) 281
15.4.10. DAC 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH) 282
15.4.11. DAC 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH) 282
15.4.12. DAC0 数据输出寄存器 (DAC0_DO) 283
15.4.13. DAC1 数据输出寄存器 (DAC1_DO) 283

16. 看门狗定时器（WDGT） 285

16.1. 独立看门狗定时器（FWDGT） 285
16.1.1. 简介 285
16.1.2. 主要特性 285
16.1.3. 功能描述 285
16.1.4. FWDGT 寄存器 287

16.2. 窗口看门狗定时器（WWDGT） 290
16.2.1. 简介 290
16.2.2. 主要特性 290
16.2.3. 功能描述 290
16.2.4. WWDGT 寄存器 292

17. 实时时钟（RTC） 294

17.1. 简介 294

17.2. 主要特性 294

17.3. 功能描述 294
17.3.1. RTC 复位 295
17.3.2. RTC 读取 295
17.3.3. RTC 配置 295
17.3.4. RTC 标志位 296

17.4. RTC 寄存器 297
17.4.1. RTC 中断使能寄存器（RTC_INTEN） 297
17.4.2. RTC 控制寄存器（RTC_CTL） 297
17.4.3. RTC 预分频寄存器高位（RTC_PSCH） 298
17.4.4. RTC 预分频寄存器低位（RTC_PSCL） 299
17.4.5. RTC 分频器高位（RTC_DIVH） 299
17.4.6. RTC 分频器低位（RTC_DIVL） 299
17.4.7. RTC 计数寄存器高位（RTC_CNTH） 300
17.4.8. RTC 计数寄存器低位（RTC_CNTL） 300
17.4.9. RTC 闹钟寄存器高位（RTC_ALRMH） 301
17.4.10. RTC 闹钟寄存器低位（RTC_ALRML） 301

18. 定时器（TIMER） 302

18.1. 高级定时器（TIMERx, x=0,7） 303

18.1.1. 简介 303
18.1.2. 主要特征 303
18.1.3. 结构框图 304
18.1.4. 功能说明 304
18.1.5. TIMERx 寄存器(x=0,7) 330

18.2. 通用定时器 L0（TIMERx, x=1,2,3,4） 355
18.2.1. 简介 355
18.2.2. 主要特征 355
18.2.3. 结构框图 356
18.2.4. 功能说明 356
18.2.5. TIMERx 寄存器(x=1,2,3,4) 372

18.3. 通用定时器 L1（TIMERx, x=8,11） 393
18.3.1. 简介 393
18.3.2. 主要特征 393
18.3.3. 结构框图 394
18.3.4. 功能说明 394
18.3.5. TIMERx 寄存器(x=8,11) 408

18.4. 通用定时器 L2（TIMERx, x=9,10,12,13） 420
18.4.1. 简介 420
18.4.2. 主要特征 420
18.4.3. 结构框图 420
18.4.4. 功能说明 421
18.4.5. TIMERx 寄存器(x=9,10,12,13) 430

18.5. 基本定时器（TIMERx, x=5,6） 440
18.5.1. 简介 440
18.5.2. 主要特征 440
18.5.3. 结构框图 440
18.5.4. 功能说明 440
18.5.5. TIMERx 寄存器(x=5,6) 444

19. 通用同步异步收发器（USART） 449

19.1. 简介 449

19.2. 主要特征 449

19.3. 功能说明 450
19.3.1. USART 帧格式 451
19.3.2. 波特率发生 452
19.3.3. USART 发送器 452
19.3.4. USART 接收器 453
19.3.5. DMA 方式访问数据缓冲区 454
19.3.6. 硬件流控制 456
19.3.7. 多处理器通信 457
19.3.8. LIN 模式 457

19.3.9. 同步通信模式 458
19.3.10. 串行红外（IrDA SIR）编解码功能模块 459
19.3.11. 半双工通信模式 460
19.3.12. 智能卡（ISO7816）模式 460
19.3.13. USART 中断 462

19.4. USART 寄存器 464
19.4.1. 状态寄存器 0（USART_STAT0） 464
19.4.2. 数据寄存器（USART_DATA） 466
19.4.3. 波特率寄存器（USART_BAUD） 466
19.4.4. 控制寄存器 0（USART_CTL0） 467
19.4.5. 控制寄存器 1（USART_CTL1） 469
19.4.6. 控制寄存器 2（USART_CTL2） 470
19.4.7. 保护时间和预分频器寄存器（USART_GP） 472
19.4.8. 控制寄存器 3（USART_CTL3） 473
19.4.9. 接收超时寄存器（USART_RT） 474
19.4.10. 状态寄存器 1（USART_STAT1） 475

20. 内部集成电路总线接口（I2C） 477

20.1. 简介 477

20.2. 主要特征 477

20.3. 功能描述 478
20.3.1. SDA 线和 SCL 线 478
20.3.2. 数据有效性 479
20.3.3. 开始和停止信号 479
20.3.4. 时钟同步 479
20.3.5. 仲裁 480
20.3.6. I2C 通讯流程 480
20.3.7. 软件编程模型 481
20.3.8. SCL 线控制 489
20.3.9. DMA 模式下数据传输 490
20.3.10. 报文错误校验 490
20.3.11. SMBus 支持 490
20.3.12. 状态、错误和中断 492

20.4. I2C 寄存器 493
20.4.1. 控制寄存器 0（I2C_CTL0） 493
20.4.2. 控制寄存器 1（I2C_CTL1） 495
20.4.3. 从机地址寄存器 0（I2C_SADDR0） 496
20.4.4. 从机地址寄存器 1（I2C_SADDR1） 496
20.4.5. 传输缓冲区寄存器（I2C_DATA） 497
20.4.6. 传输状态寄存器 0（I2C_STAT0） 497
20.4.7. 传输状态寄存器 1（I2C_STAT1） 499
20.4.8. 时钟配置寄存器（I2C_CKCFG） 501

20.4.9. 上升时间寄存器（I2C_RT） 501

21. 串行外设接口/片上音频接口（SPI/I2S） 503

21.1. 简介 503

21.2. 主要特征 503

21.2.1. SPI 主要特征 503

21.2.2. I2S 主要特征 503

21.3. SPI 功能说明 504

21.3.1. SPI 结构框图 504

21.3.2. SPI 信号线描述 504

21.3.3. SPI 时序和数据帧格式 505

21.3.4. NSS 功能 506

21.3.5. SPI 运行模式 507

21.3.6. DMA 功能 513

21.3.7. CRC 功能 513

21.3.8. SPI 中断 513

21.4. I2S 功能说明 515

21.4.1. I2S 结构框图 515

21.4.2. I2S 信号线描述 515

21.4.3. I2S 音频标准 515

21.4.4. I2S 时钟 523

21.4.5. 运行 524

21.4.6. DMA 功能 528

21.4.7. I2S 中断 528

21.5. SPI/I2S 寄存器 530

21.5.1. 控制寄存器 0（SPI_CTL0） 530

21.5.2. 控制寄存器 1（SPI_CTL1） 532

21.5.3. 状态寄存器（SPI_STAT） 532

21.5.4. 数据寄存器（SPI_DATA） 534

21.5.5. CRC 多项式寄存器（SPI_CRCPOLY） 534

21.5.6. 接收 CRC 寄存器（SPI_RCRC） 535

21.5.7. 发送 CRC 寄存器（SPI_TCRC） 535

21.5.8. I2S 控制寄存器（SPI_I2SCTL） 536

21.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC） 537

21.5.10. SPI0 四线 SPI 控制寄存器（SPI_QCTL） 538

22. 数字摄像头接口（DCI） 540

22.1. 简介 540

22.2. 主要特征 540

22.3. 结构框图 540

22.4. 信号描述 541

22.5. 功能说明 541
22.5.1. DCI 硬件同步模式 541
22.5.2. 内嵌码同步模式 542
22.5.3. 用快照或连续捕获模式捕获数据 542
22.5.4. 窗口功能 542
22.5.5. 像素格式，数据填充和 DMA 接口 543
22.6. 中断 543
22.7. DCI 寄存器 545
22.7.1. 控制寄存器（DCI_CTL） 545
22.7.2. 状态寄存器 0（DCI_STAT0） 546
22.7.3. 状态寄存器 1（DCI_STAT1） 547
22.7.4. 中断使能寄存器（DCI_INTEN） 548
22.7.5. 中断标志寄存器（DCI_INTF） 548
22.7.6. 中断标志清除寄存器（DCI_INTC） 549
22.7.7. 同步码寄存器（DCI_SC） 550
22.7.8. 同步码屏蔽寄存器（DCI_SCUMSK） 550
22.7.9. 剪裁窗口开始位置寄存器（DCI_CWSPOS） 551
22.7.10. 剪裁窗口大小寄存器（DCI_CWSZ） 551
22.7.11. 数据寄存器（DCI_DATA） 552
23. TFT-LCD 接口（TLI） 553
23.1. 简介 553
23.2. 主要特征 553
23.3. 结构框图 553
23.4. 信号线描述 554
23.5. 功能说明 554
23.5.1. LCD 显示时序 554
23.5.2. 像素 DMA 功能 555
23.5.3. 像素格式 556
23.5.4. 层窗口和混合功能 556
23.5.5. Layer 配置重载 557
23.5.6. 抖动 557
23.5.7. 中断 557
23.6. TLI 寄存器 559
23.6.1. 同步脉冲宽度寄存器（TLI_SPSZ） 559
23.6.2. 后沿宽度寄存器（TLI_BPSZ） 559
23.6.3. 有效宽度寄存器（TLI_ASZ） 560
23.6.4. 总宽度寄存器(TLI_TSZ） 560
23.6.5. 控制寄存器（TLI_CTL） 561
23.6.6. 重载层配置寄存器（TLI_RL） 562
23.6.7. 背景色配置寄存器（TLI_BGC） 563

23.6.8. 中断使能寄存器（TLI_INTEN）.....563
23.6.9. 中断标志寄存器（TLI_INTF）.....564
23.6.10. 中断标志清除寄存器（TLI_INTC）.....565
23.6.11. 行标记寄存器（TLI_LM）.....565
23.6.12. 当前像素位置寄存器（TLI_CPPOS）.....566
23.6.13. 状态寄存器（TLI_STAT）.....566
23.6.14. 第 x 层控制寄存器（TLI_LxCTL）.....567
23.6.15. 第 x 层水平位置参数寄存器（TLI_LxHPOS）.....568
23.6.16. 第 x 层垂直位置参数寄存器（TLI_LxVPOS）.....568
23.6.17. 第 x 层色键值寄存器（TLI_LxCKEY）.....569
23.6.18. 第 x 层像素格式寄存器（TLI_LxPPF）.....569
23.6.19. 第 x 层恒定 Alpha 寄存器（TLI_LxSA）.....570
23.6.20. 第 x 层默认颜色寄存器（TLI_LxDC）.....570
23.6.21. 第 x 层混合寄存器（TLI_LxBLEND）.....571
23.6.22. 第 x 层帧基地址寄存器（TLI_LxFBADDR）.....571
23.6.23. 第 x 层行长度寄存器（TLI_LxFLLEN）.....572
23.6.24. 第 x 层总行数寄存器（TLI_LxFTLN）.....572
23.6.25. 第 x 层颜色查找表寄存器（TLI_LxLUT）.....573

24. SDIO 接口（SDIO）.....574

24.1. 简介.....574

24.2. 主要特征.....574

24.3. SDIO 总线拓扑.....574

24.4. SDIO 功能描述.....576
24.4.1. SDIO 适配器.....577
24.4.2. AHB 接口.....580

24.5. 卡功能描述.....582
24.5.1. 卡寄存器.....582
24.5.2. 命令.....583
24.5.3. 响应.....592
24.5.4. 数据包格式.....595
24.5.5. 卡的两种状态.....596

24.6. 编程序列.....602
24.6.1. 卡识别.....602
24.6.2. 无数据命令.....603
24.6.3. 单个数据块或多个数据块写.....604
24.6.4. 单个数据块或多个数据块读.....605
24.6.5. 数据流写和数据流读 (仅适用于 MMC).....606
24.6.6. 擦除.....607
24.6.7. 总线宽度选择.....608
24.6.8. 保护管理.....608
24.6.9. 卡上锁/解锁操作.....608

24.7. 特定操作 610
24.7.1. SD I/O 特定操作 610
24.7.2. CE-ATA 特定操作 613
24.8. SDIO 寄存器 615
24.8.1. 电源控制寄存器（SDIO_PWRCTL） 615
24.8.2. 时钟控制寄存器（SDIO_CLKCTL） 615
24.8.3. 命令参数寄存器（SDIO_CMDAGMT） 616
24.8.4. 命令控制寄存器（SDIO_CMDCTL） 617
24.8.5. 命令索引响应寄存器（SDIO_RSPCMDIDX） 618
24.8.6. 响应寄存器（SDIO_RESPx x=0..3） 619
24.8.7. 数据超时寄存器（SDIO_DATATO） 619
24.8.8. 数据长度寄存器（SDIO_DATALEN） 620
24.8.9. 数据控制寄存器（SDIO_DATACTL） 620
24.8.10. 数据计数寄存器（SDIO_DATACNT） 622
24.8.11. 状态寄存器（SDIO_STAT） 622
24.8.12. 中断清除寄存器（SDIO_INTC） 624
24.8.13. 中断使能寄存器（SDIO_INTEN） 625
24.8.14. FIFO 计数寄存器（SDIO_FIFOCNT） 627
24.8.15. FIFO 数据寄存器（SDIO_FIFO） 627
25. 外部存储器控制器（EXMC） 629
25.1. 简介 629
25.2. 主要特征 629
25.3. 功能说明 629
25.3.1. 结构框图 629
25.3.2. EXMC 访问基本规范 630
25.3.3. 外部设备地址映射 631
25.3.4. NOR/PSRAM 控制器 634
25.3.5. NAND Flash 或 PC Card 控制器 654
25.3.6. SDRAM 控制器 658
25.4. EXMC 寄存器 669
25.4.1. NOR/PSRAM 控制器寄存器 669
25.4.2. NAND Flash/PC Card 控制器寄存器 673
25.4.3. SDRAM 控制器寄存器 679
25.4.4. SQPI-PSRAM 控制器寄存器 686
26. 控制器局域网络（CAN） 690
26.1. 简介 690
26.2. 主要特征 690
26.3. 功能说明 690
26.3.1. 工作模式 691
26.3.2. 通信模式 692

26.3.3. 数据发送 693
26.3.4. 数据接收 694
26.3.5. 过滤功能 695
26.3.6. 时间触发通信 698
26.3.7. 通信参数 699
26.3.8. 错误标志 700
26.3.9. 中断 701

26.4. CAN 寄存器 702
26.4.1. 控制寄存器（CAN_CTL） 702
26.4.2. 状态寄存器（CAN_STAT） 703
26.4.3. 发送状态寄存器（CAN_TSTAT） 705
26.4.4. 接收 FIFO0 寄存器（CAN_RFIFO0） 707
26.4.5. 接收 FIFO1 寄存器（CAN_RFIFO1） 708
26.4.6. 中断使能寄存器（CAN_INTEN） 709
26.4.7. 错误寄存器（CAN_ERR） 710
26.4.8. 位时序寄存器（CAN_BT） 711
26.4.9. 发送邮箱标识符寄存器（CAN_TMIx）（x=0..2） 712
26.4.10. 发送邮箱属性寄存器（CAN_TMPx）（x=0..2） 713
26.4.11. 发送邮箱 data0 寄存器（CAN_TMDATA0x）（x=0..2） 713
26.4.12. 发送邮箱 data1 寄存器（CAN_TMDATA1x）（x=0..2） 714
26.4.13. 接收 FIFO 邮箱标识符寄存器（CAN_RFIFOMIx）（x=0,1） 714
26.4.14. 接收 FIFO 邮箱属性寄存器（CAN_RFIFOMPx）（x=0,1） 715
26.4.15. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x=0,1） 716
26.4.16. 接收 FIFO 邮箱 data1 寄存器（CAN_RFIFOMDATA1x）（x=0,1） 716
26.4.17. 过滤器控制寄存器（CAN_FCTL）（仅 CAN0 可用） 716
26.4.18. 过滤器模式配置寄存器（CAN_FMCFG）（仅 CAN0 可用） 717
26.4.19. 过滤器位宽配置寄存器（CAN_FSCFG）（仅 CAN0 可用） 718
26.4.20. 过滤器关联 FIFO 寄存器（CAN_FAFIFO）（仅 CAN0 可用） 718
26.4.21. 过滤器激活寄存器（CAN_FW）（仅 CAN0 可用） 718
26.4.22. 过滤器（x）数据（y）寄存器（CAN_FxDATAy）（x=0..27, y=0,1）（仅 CAN0 可用） 719

27. 以太网（ENET） 720

27.1. 简介 720

27.2. 主要特性 720
27.2.1. 模块框图 721
27.2.2. MAC 802.3 以太网数据包描述 722
27.2.3. 以太网信号描述 722

27.3. 功能说明 723
27.3.1. 接口配置 723
27.3.2. MAC 功能简介 727
27.3.3. MAC 统计计数器：MSC 737
27.3.4. 唤醒管理：WUM 738
27.3.5. 精确时间协议：PTP 740

27.3.6. DMA 控制器描述 744
27.3.7. 典型的以太网配置流程示例 761
27.3.8. 以太网中断 762

27.4. ENET 寄存器 765
27.4.1. MAC 配置寄存器（ENET_MAC_CFG） 765
27.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF） 767
27.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH） 769
27.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL） 769
27.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL） 769
27.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA） 770
27.4.7. MAC 流控寄存器（ENET_MAC_FCTL） 771
27.4.8. MAC 流控阈值寄存器（ENET_MAC_FCTH） 772
27.4.9. MAC VLAN 标签寄存器（ENET_MAC_VLT） 773
27.4.10. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF） 774
27.4.11. MAC 唤醒管理寄存器（ENET_MAC_WUM） 774
27.4.12. MAC 中断状态寄存器（ENET_MAC_INTF） 775
27.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK） 776
27.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H） 777
27.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L） 777
27.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H） 778
27.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L） 779
27.4.18. MAC 地址 2 高寄存器（ENET_MAC_ADDR2H） 779
27.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L） 780
27.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H） 780
27.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L） 781
27.4.22. MSC 控制寄存器（ENET_MSC_CTL） 781
27.4.23. MSC 接收中断状态寄存器（ENET_MSC_RINTF） 782
27.4.24. MSC 发送中断状态寄存器（ENET_MSC_TINTF） 783
27.4.25. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK） 783
27.4.26. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK） 784
27.4.27. MSC 1 次冲突后发送”好”帧的计数器寄存器（ENET_MSC_SCCNT） 785
27.4.28. MSC 1 次以上冲突后发送”好”帧的计数器寄存器（ENET_MSC_MSCCNT） 785
27.4.29. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT） 786
27.4.30. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT） 786
27.4.31. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT） 787
27.4.32. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT） 787
27.4.33. PTP 时间戳控制寄存器（ENET_PTP_TSCTL） 787
27.4.34. PTP 亚秒递增寄存器（ENET_PTP_SSINC） 789
27.4.35. PTP 时间戳高寄存器（ENET_PTP_TSH） 789
27.4.36. PTP 时间戳低寄存器（ENET_PTP_TSL） 789
27.4.37. PTP 时间戳高更新寄存器（ENET_PTP_TSUH） 790
27.4.38. PTP 时间戳低更新寄存器（ENET_PTP_TSUL） 790
27.4.39. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND） 791

27.4.40. PTP 期望时间高寄存器（ENET_PTP_ETH）.....791
27.4.41. PTP 期望时间低寄存器（ENET_PTP_ETL）.....792
27.4.42. DMA 总线控制寄存器（ENET_DMA_BCTL）.....792
27.4.43. DMA 发送查询使能寄存器（ENET_DMA_TPEN）.....794
27.4.44. DMA 接收查询使能寄存器（ENET_DMA_RPEN）.....795
27.4.45. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR）.....795
27.4.46. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR）.....796
27.4.47. DMA 状态寄存器（ENET_DMA_STAT）.....796
27.4.48. DMA 控制寄存器（ENET_DMA_CTL）.....799
27.4.49. DMA 中断使能寄存器（ENET_DMA_INTEN）.....802
27.4.50. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT）.....804
27.4.51. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR）.....804
27.4.52. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR）.....805
27.4.53. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR）.....805
27.4.54. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR）.....806

28. 通用串行总线全速接口（USBFS）.....807

28.1. 简介.....807

28.2. 主要特征.....807

28.3. 结构框图.....808

28.4. 信号线描述.....808

28.5. 功能说明.....808

28.5.1. USBFS 时钟及工作模式.....808
28.5.2. USB 主机功能.....810
28.5.3. USB 设备功能.....812
28.5.4. OTG 功能概述.....813
28.5.5. 数据 FIFO.....814
28.5.6. 操作手册.....816

28.6. 中断.....819

28.7. USBFS 寄存器.....821

28.7.1. 全局控制与状态寄存器组.....821
28.7.2. 主机控制和状态寄存器.....841
28.7.3. 设备控制和状态寄存器.....853
28.7.4. 电源和时钟控制寄存器 （USBFS_PWRCLKCTL）.....875

29. 版本历史.....877

图索引

图 1-1. Cortex®-M3 结构图 33
图 1-2. GD32F20x Connectivity line 系列系统架构 34
图 2-1. 页擦除操作流程 45
图 2-2. 整片擦除操作流程 46
图 2-3. 字编程操作流程 47
图 3-1. 电源域概览 61
图 3-2. 上电/掉电复位波形图 62
图 3-3. LVD 阈值波形图 63
图 5-1. 系统复位电路 77
图 5-2. 时钟树 78
图 5-3. HXTAL 时钟源 80
图 5-4. 旁路模式下 HXTAL 时钟源 80
图 6-1. EXTI 框图 125
图 7-1. 标准 I/O 端口和 5V 兼容 I/O 口的基本结构 131
图 7-2. 输入配置 133
图 7-3. 输出配置 133
图 7-4. 模拟配置 134
图 7-5. 备用功能配置 135
图 8-1. CRC 管理单元框图 178
图 9-1. TRNG 模块框图 182
图 10-1. DATAM 不交换/半字交换 188
图 10-2. DATATM 字节交换/位交换 189
图 10-3. CAU 框图 190
图 10-4. DES/TDES ECB 加密 191
图 10-5. DES/TDES ECB 解密 192
图 10-6. DES/TDES CBC 加密 193
图 10-7. DES/TDES CBC 解密 194
图 10-8. AES ECB 加密 195
图 10-9. AES ECB 解密 195
图 10-10. AES CBC 加密 196
图 10-11. AES CBC 解密 197
图 10-12. 计数器块结构 197
图 10-13. AES CTR 加密/解密 198
图 11-1. DATAM 不交换/半字交换 212
图 11-2. DATAM 字节交换/位交换 212
图 11-3. HAU 结构框图 213
图 11-4. DMA 结构框图 224
图 11-5. 握手机制 227
图 11-6. DMA 中断逻辑图 229
图 11-7. DMA0 请求映射 230
图 11-8. DMA1 请求映射 231

图 14-1. ADC 模块框图 248
图 14-2. 单次运行模式 249
图 14-3. 连续运行模式 250
图 14-4. 扫描运行模式，且连续运行模式失能 251
图 14-5. 扫描运行模式，连续运行模式使能 251
图 14-6. 间断运行模式 251
图 14-7. 12 位数据存储模式 252
图 14-8. 10 位数据存储模式 252
图 14-9. 8 位数据存储模式 253
图 14-10. 6 位数据存储模式 253
图 14-11. 20 位到 16 位的结果截断 256
图 14-12. 右移 5 位和取整的数例 256
图 14-13. ADC 同步框图 258
图 14-14. 基于 10 个通道的常规并行模式 259
图 14-15. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1） 259
图 14-16. 常规序列通道上的慢速交叉模式 260
图 15-1. DAC 结构框图 272
图 15-2. DAC LFSR 算法 274
图 15-3. DAC 三角噪声模式波形 275
图 16-1. 独立看门狗定时器框图 286
图 16-2. 窗口看门狗定时器框图 290
图 16-3. 窗口看门狗定时器时序图 291
图 17-1. RTC 框图 295
图 17-2. RTC 秒信号及闹钟信号的波形（RTC_PSC = 3,RTC_ALRM = 2） 296
图 17-3. RTC 秒信号及溢出信号的波形（RTC_PSC = 3） 296
图 18-1. 高级定时器结构框图 304
图 18-2. 内部时钟分频为 1 时，计数器的时序图 305
图 18-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 306
图 18-4. 向上计数时序图，PSC=0/2 307
图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 307
图 18-6. 向下计数时序图，PSC=0/2 308
图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 309
图 18-8. 中央对齐计数模式计数器时序图 310
图 18-9. 中央对齐计数模式下计数器重复时序图 311
图 18-10. 在向上计数模式下计数器重复时序图 311
图 18-11. 在向下计数模式下计数器重复时序图 312
图 18-12. 通道输入捕获原理 313
图 18-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2） 314
图 18-14. 通道输出比较原理 314
图 18-15. 三种输出比较模式 315
图 18-16. EAPWM 时序图 316
图 18-17. CAPWM 时序图 317
图 18-18. 带死区时间的通道互补输出 319
图 18-19. 通道响应中止输入（高电平有效）时，输出信号的行为 320

图 18-20. 在编码器模式 2 且 CI0FE0 极性不反相时计数器行为 321
图 18-21. 在编码器模式 2 且 CI0FE0 极性反相时计数器行为 321
图 18-22. 霍尔传感器用在 BLDC 电机控制中 322
图 18-23. 两个定时器之间的霍尔传感器时序图 323
图 18-24. 复位模式下的控制电路 324
图 18-25. 暂停模式下的控制电路 324
图 18-26. 事件模式下的控制电路 325
图 18-27. 单脉冲模式，TIMERx_CHxCV = 4，TIMERx_CAR=99 325
图 18-28. 定时器 0 主/从模式的例子 326
图 18-29. 用定时器 2 的使能信号启动定时器 0 327
图 18-30. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2 328
图 18-31. 通用定时器 L0 结构框图 356
图 18-32. 内部时钟分频为 1 时，计数器的时序图 357
图 18-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图 358
图 18-34. 向上计数时序图，PSC=0/2 359
图 18-35. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 359
图 18-36. 向下计数时序图，PSC=0/2 360
图 18-37. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 361
图 18-38. 中央对齐计数模式计数器时序图 362
图 18-39. 通道输入捕获原理 363
图 18-40. 通道输出比较原理（x=0,1,2,3） 364
图 18-41. 三种输出比较模式 365
图 18-42. EAPWM 时序图 366
图 18-43. CAPWM 时序图 366
图 18-44. 复位模式下的控制电路 368
图 18-45. 暂停模式下的控制电路 368
图 18-46. 事件模式下的控制电路 369
图 18-47. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 369
图 18-48. 通用定时器 L1 结构框图 394
图 18-49. 内部时钟分频为 1 时，计数器的时序图 395
图 18-50. 当 PSC 数值从 0 变到 2 时，计数器的时序图 396
图 18-51. 向上计数时序图，PSC=0/2 397
图 18-52. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 397
图 18-53. 向下计数时序图，PSC=0/2 398
图 18-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 399
图 18-55. 中央对齐计数模式计数器时序图 400
图 18-56. 通道输入捕获原理 401
图 18-57. 通道输出比较原理（x=0,1） 402
图 18-58. 三种输出比较模式 403
图 18-59. EAPWM 时序图 404
图 18-60. CAPWM 时序图 404
图 18-61. 复位模式下的控制电路 405
图 18-62. 暂停模式下的控制电路 406
图 18-63. 事件模式下的控制电路 406

图 18-64. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 407
图 18-65. 通用定时器 L2 结构框图 420
图 18-66. 内部时钟分频为 1 时，计数器的时序图 421
图 18-67. 当 PSC 数值从 0 变到 2 时，计数器的时序图 422
图 18-68. 向上计数时序图，PSC=0/2 423
图 18-69. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 423
图 18-70. 向下计数时序图，PSC=0/2 424
图 18-71. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 425
图 18-72. 中央对齐计数模式计数器时序图 426
图 18-73. 通道输入捕获原理 427
图 18-74. 三种输出比较模式 428
图 18-75. 基本定时器结构框图 440
图 18-76. 内部时钟分频为 1 时，计数器的时序图 441
图 18-77. 当 PSC 数值从 0 变到 2 时，计数器的时序图 441
图 18-78. 向上计数时序图，PSC=0/2 442
图 18-79. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 443
图 19-1. USART 模块内部框图 451
图 19-2. USART 字符帧（8 数据位和 1 停止位）） 451
图 19-3. USART 发送步骤 453
图 19-4. 过采样方式接收一个数据位 454
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤 455
图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤 456
图 19-7. 两个 USART 之间的硬件流控制 456
图 19-8. 硬件流控制 457
图 19-9. 空闲状态下检测断开帧 458
图 19-10. 数据传输过程中检测断开帧 458
图 19-11. 同步模式下的 USART 示例 459
图 19-12. 8-bit 格式的 USART 同步通信波形（CLEN=1） 459
图 19-13. IrDA SIR ENDEC 模块 459
图 19-14. IrDA 数据调制 460
图 19-15. ISO7816-3 数据帧格式 461
图 19-16. USART 中断映射框图 463
图 20-1. I2C 模块框图 478
图 20-2. 数据有效性 479
图 20-3. 开始和停止信号 479
图 20-4. 时钟同步 479
图 20-5. SDA 线仲裁 480
图 20-6. 7 位地址的 I2C 通讯流程 480
图 20-7. 10 位地址的 I2C 通讯流程（主机发送） 480
图 20-8. 10 位地址的 I2C 通讯流程（主机接收） 481
图 20-9. 从机发送模式（10 位地址模式） 483
图 20-10. 从机接收模式（10 位地址模式） 484
图 20-11. 主机发送模式（10 位地址模式） 485
图 20-12. 主机接收使用方案 A 模式（10 位地址模式） 487

图 20-13. 主机接收使用方案 B 模式（10 位地址模式） 489
图 21-1. SPI 结构框图 504
图 21-2. 常规模式下的 SPI 时序图 505
图 21-3. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0） 506
图 21-4. 典型的全双工模式连接 508
图 21-5. 典型的单工模式连接（主机：接收，从机：发送） 509
图 21-6. 典型的单工模式连接（主机：只发送，从机：接收） 509
图 21-7. 典型的双向线连接 509
图 21-8. SPI 四线模式写操作时序图 511
图 21-9. SPI 四线模式读操作时序图 512
图 21-10. I2S 结构框图 515
图 21-11. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 516
图 21-12. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 516
图 21-13. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 516
图 21-14. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 516
图 21-15. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 517
图 21-16. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 517
图 21-17. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 517
图 21-18. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 517
图 21-19. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 518
图 21-20. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 518
图 21-21. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 518
图 21-22. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 518
图 21-23. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 518
图 21-24. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 518
图 21-25. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 519
图 21-26. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 519
图 21-27. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 519
图 21-28. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 519
图 21-29. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 520
图 21-30. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 520
图 21-31. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 520
图 21-32. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 520
图 21-33. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 520
图 21-34. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 521
图 21-35. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 521
图 21-36. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 521
图 21-37. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 521
图 21-38. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 521
图 21-39. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0） 521
图 21-40. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1） 522
图 21-41. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0） 522
图 21-42. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1） 522
图 21-43. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0） 522

图 21-44. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1） 522
图 21-45. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0） 522
图 21-46. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1） 523
图 21-47. I2S 时钟生成结构框图 523
图 21-48. I2S 初始化流程 525
图 21-49. I2S 主机接收禁能流程 527
图 22-1. DCI 模块示意图 540
图 22-2. 硬件同步模式 541
图 22-3. 硬件同步模式之 JPEG 子模式 542
图 23-1. TLI 模块框图 554
图 23-2. 显示时序图 555
图 23-3. 混合过程框图 557
图 24-1. SDIO “无响应” 和 “无数据” 操作 575
图 24-2. SDIO 多块读操作 575
图 24-3. SDIO 多块写操作 576
图 24-4. SDIO 数据流读操作 576
图 24-5. SDIO 数据流写操作 576
图 24-6. SDIO 框图 577
图 24-7. 命令标记格式 583
图 24-8. 响应令牌格式 592
图 24-9. 1 位数据总线宽度 595
图 24-10. 4 位数据总线宽度 595
图 24-11. 8 位数据总线宽度 596
图 24-12. 通过停止 SDIO_CLK 的读等待操作 611
图 24-13. 使用 SDIO_DAT[2]信号线的读等待操作 611
图 24-14. 在功能 1 的多块读周期期间插入功能 2 读周期 612
图 24-15. 读中断周期时序 612
图 24-16. 写中断周期时序 613
图 24-17. 4 位模式下多块读中断周期时序 613
图 24-18. 4 位模式下多块写中断周期时序 613
图 24-19. 命令完成信号关闭操作 614
图 25-1. 系统架构 630
图 25-2. EXMC Bank 划分 631
图 25-3. Bank0 地址映射 632
图 25-4. NAND/PC Card 地址映射 633
图 25-5. Bank1 通用空间 633
图 25-6. SDRAM 地址映射 634
图 25-7. 模式 1 读时序 638
图 25-8. 模式 1 写时序 638
图 25-9. 模式 A 读时序 639
图 25-10. 模式 A 写时序 639
图 25-11. 模式 2/B 读时序 641
图 25-12. 模式 2 写时序 641
图 25-13. 模式 B 写时序 641

图 25-14. 模式 C 读时序 643
图 25-15. 模式 C 写时序 643
图 25-16. 模式 D 读时序 644
图 25-17. 模式 D 写时序 645
图 25-18. 复用模式读时序 646
图 25-19. 复用模式写时序 646
图 25-20. 异步等待有效时的读时序 647
图 25-21. 异步等待有效时的写时序 648
图 25-22. 同步复用突发传输读时序 650
图 25-23. 同步复用突发传输写时序 651
图 25-24. SPI-PSRAM 访问时序 653
图 25-25. SQPI-PSRAM 访问时序 653
图 25-26. QPI-PSRAM 访问时序 654
图 25-27. NAND/PC Card 通用空间操作时序 656
图 25-28. NCE 敏感 NAND Flash 访问时序 657
图 25-29. SDRAM 系统架构 660
图 25-30. 突发读操作 663
图 25-31. 数据采样时钟延迟模块 663
图 25-32. 突发写操作 664
图 25-33. FIFO 未命中时的读访问（BRSTRD=1，CL=2，SDCLK=2，PIPED=2） 665
图 25-34. FIFO 命中时的读访问（BRSTRD=1） 665
图 25-35. 跨边界读操作 666
图 25-36. 跨边界写操作 666
图 25-37. 自刷新模式进入和退出的处理 667
图 25-38. 掉电模式进入和退出的处理 667
图 26-1. CAN 模块结构框图 691
图 26-2. 发送寄存器 693
图 26-3. 发送邮箱状态转换 693
图 26-4. 接收寄存器 695
图 26-5. 32-bit 位宽过滤器 696
图 26-6. 16-bit 位宽过滤器 696
图 26-7. 32-bit 位宽掩码模式过滤器 696
图 26-8. 16-bit 位宽掩码模式过滤器 696
图 26-9. 32-bit 位宽列表模式过滤器 697
图 26-10. 16-bit 位宽列表模式过滤器 697
图 26-11. 位时序 699
图 27-1. 以太网模块框图 721
图 27-2. MAC/带标签的 MAC 帧格式 722
图 27-3. 站点管理接口信号 724
图 27-4. 媒体独立接口(MII)信号线 725
图 27-5. 精简媒体独立接口(RMII)信号线 727
图 27-6. 唤醒帧过滤器寄存器 738
图 27-7. 系统时钟精细校准方法 741
图 27-8. 描述符的环结构和链结构 745

图 27-9. 发送描述符 749
图 27-10. 接收描述符 756
图 27-11. MAC 控制器中断示意图 763
图 27-12. 以太网中断示意图 764
图 27-13. 远程唤醒帧过滤器寄存器 774
图 28-1. USBFS 结构框图 808
图 28-2. 在主机或设备模式下连接示意图 809
图 28-3. OTG 模式下连接示意图 810
图 28-4. 主机端口状态转移图 810
图 28-5. 主机模式 FIFO 空间 814
图 28-6. 主机模式 FIFO 访问寄存器映射表 815
图 28-7. 设备模式 FIFO 空间 815
图 28-8. 设备模式 FIFO 访问寄存器映射表 816

表索引

表 1-1 GD32F20x 系列器件的存储映射表 35
表 1-2. SRAM 中各模块 39
表 1-3. 引导模式 39
表 1-4. 引导程序支持外设 40
表 2-1. GD32F20x_CL 43
表 2-2. 选项字节 49
表 3-1. 节电模式总结 65
表 5-1. 时钟输出 0 的时钟源选择 82
表 5-2. 时钟输出 1 的时钟源选择 83
表 5-3. 深度睡眠模式下 1.2V 域电压选择 83
表 6-1. Cortex-M3 中的 NVIC 异常类型 121
表 6-2. 中断向量表 122
表 6-3. EXTI 触发源 126
表 7-1. GPIO 配置表 131
表 7-2. 调试接口信号 136
表 7-3. 调试端口映射 137
表 7-4. ADC0/1 常规转换外部触发备用功能重映射[1] 137
表 7-5. TIMER 备用功能重映射 138
表 7-6. TMER4 备用功能重映射[1] 139
表 7-7. USART0/1 备用功能重映射 139
表 7-8. I2C0/I2C1/I2C2 备用功能重映射 140
表 7-9. SPI0/SPI1/I2S1/SPI2/I2S2 备用功能重映射 140
表 7-10. CAN0/1 备用功能重映射 141
表 7-11. ENET 备用功能重映射 142
表 7-12. DCI 备用功能重映射 143
表 7-13. TLI 备用功能重映射 143
表 7-14. OSC32 引脚配置 144
表 7-15. OSC 引脚配置 1 145
表 7-16. OSC 引脚配置 2 145
表 11-1. DMA 传输操作（普通模式） 225
表 11-2. DMA 传输操作（完整数据传输模式） 226
表 11-3. 中断事件 228
表 11-4. DMA0 各通道请求表 230
表 11-5. DMA1 各通道请求表 232
表 13-1. 调试引脚分配 240
表 14-1. ADC 内部输入信号 247
表 14-2. ADC 输入引脚定义 247
表 14-3. ADC0 和 ADC1 的外部触发源 253
表 14-4. ADC2 的外部触发源 254
表 14-5. 不同分辨率对应的 t_{CONV} 时间 255
表 14-6. N 和 M 的最大输出值（灰色部分表示截断） 256

表 14-7. ADC 同步模式表 257
表 15-1. DAC I/O 描述 273
表 15-2. DAC 外部触发 273
表 16-1. 独立看门狗定时器在 40kHz（IRC40K）时的最小/最大超时周期 286
表 16-2. 在 60MHz（f_{PCLK1} 时的最大/最小超时值 291
表 18-1. 定时器（TIMERx）分为五种类型 302
表 18-2. 由参数控制的互补输出表 318
表 18-3. 不同编码器模式下的计数方向 320
表 18-4. 从模式例子列表 323
表 18-5. 定时器 0 和定时器 7 内部触发 326
表 18-6. 定时器 0 和定时器 7 触发输出 326
表 18-7. 从模式列表和举例 367
表 18-8. 定时器 x 输入触发（x=1,2,3,4） 370
表 18-9. 定时器 x 输出触发（x=1,2,3,4） 370
表 18-10. 从模式列表和举例 405
表 18-11. 定时器 x 输入触发（x=8,11） 407
表 19-1. USART 重要引脚描述 450
表 19-2. 停止位配置 451
表 19-3. USART 中断请求 462
表 20-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 478
表 20-2. 事件状态标志位 492
表 20-3. I2C 错误标志位 492
表 21-1. SPI 信号描述 504
表 21-2. SPI 四线信号描述 505
表 21-3. 从机模式 NSS 功能 506
表 21-4. 主机模式 NSS 功能 507
表 21-5. SPI 运行模式 507
表 21-6. SPI 中断请求 514
表 21-7. I2S 比特率计算公式 523
表 21-8. 音频采样频率计算公式 523
表 21-9. 各种运行模式下 I2S 接口信号的方向 524
表 21-10. I2S 中断 529
表 22-1. DCI 引脚 541
表 22-2. 字节填充模式下的存储视图 543
表 22-3. 半字填充模式下的存储视图 543
表 22-4. 状态/错误标志 544
表 23-1. TLI 提供的显示接口的引脚 554
表 23-2. 八种像素格式 556
表 23-3. 状态标志 557
表 23-4. 错误标志 558
表 24-1. SDIO I/O 定义 577
表 24-2. 命令格式 583
表 24-3. 卡命令类 (CCCs) 584
表 24-4. 基本命令(class 0) 586

表 24-5. 面向块的读命令(class 2) 587
表 24-6. 流读取命令(class 1)和流写入命令(class 3) 588
表 24-7. 面向块的写命令(class 4) 588
表 24-8. 擦除命令(class 5) 589
表 24-9. 面向块的写保护命令(class 6) 589
表 24-10. 锁卡命令(class 7) 589
表 24-11. 特定应用命令(class 8) 590
表 24-12. I/O 模式命令(class 9) 590
表 24-13. 切换功能命令(class 10) 591
表 24-14. R1 响应 592
表 24-15. R2 响应 593
表 24-16. R3 响应 593
表 24-17. R4 响应(MMC) 593
表 24-18. R4 响应(SD I/O) 594
表 24-19. R5 响应(MMC) 594
表 24-20. R5 响应(SD I/O) 594
表 24-21. R6 响应 594
表 24-22. R7 响应 595
表 24-23. 卡状态 597
表 24-24. SD 状态 599
表 24-25. 移动性能字段 600
表 24-26. AU_SIZE 字段 600
表 24-27. 最大 AU 大小 601
表 24-28. 擦除大小字段 601
表 24-29. 擦除超时字段 601
表 24-30. 擦除偏移字段 602
表 24-31. 上锁/解锁数据结构 609
表 24-32. 不同响应类型对应的 SDIO_RESPx 寄存器 619
表 25-1. SDRAM 地址映射 634
表 25-2. NOR Flash 接口信号描述 635
表 25-3. PSRAM 非复用接口信号描述 635
表 25-4. SQPI-PSRAM 接口信号描述 635
表 25-5. EXMC 对 NOR，PSRAM 和 SRAM 支持的访问模式 636
表 25-6. NOR/PSRAM 控制时序参数 637
表 25-7. EXMC 时序模型 637
表 25-8. 模式 1 相关寄存器配置 638
表 25-9. 模式 A 相关寄存器配置 640
表 25-10. 模式 2/B 相关寄存器配置 642
表 25-11. 模式 C 相关寄存器配置 643
表 25-12. 模式 D 相关寄存器配置 645
表 25-13. 复用模式相关寄存器配置 646
表 25-14. 同步复用模式读时序配置 650
表 25-15. 同步复用模式写时序配置 651
表 25-16. SPI/QPI 接口 652

表 25-17. 8 位/16 位 NAND 接口信号描述 654
表 25-18. 16 位 PC Card 接口信号描述 654
表 25-19. Bank1/2/3 支持的访问模式 655
表 25-20. NADN/PC Card 可编程参数 655
表 25-21. SDRAM 命令真值表 660
表 25-22. SDRAM IO 定义 661
表 26-1. 32-bit 过滤序号 697
表 26-2. 过滤索引 698
表 27-1. 以太网模块引脚配置 722
表 27-2. 时钟范围 725
表 27-3. 接收接口信号编码 726
表 27-4. 目标地址过滤器结果列表 734
表 27-5. 源地址过滤器结果列表 734
表 27-6. 接收描述符 0 错误状态描述 759
表 28-1. USBFS 信号线描述 808
表 28-2. USBFS 全局中断 819
表 29-1. 版本历史 877

1. 系统及存储器架构

GD32F20x 系列器件是基于 Arm® Cortex®-M3 处理器的 32 位通用控制器。Arm® Cortex®-M3 处理器包括三条 AHB 总线分别称为 I-CODE 总线、D-CODE 总线和系统总线。Cortex®-M3 处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这三条总线上执行。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。

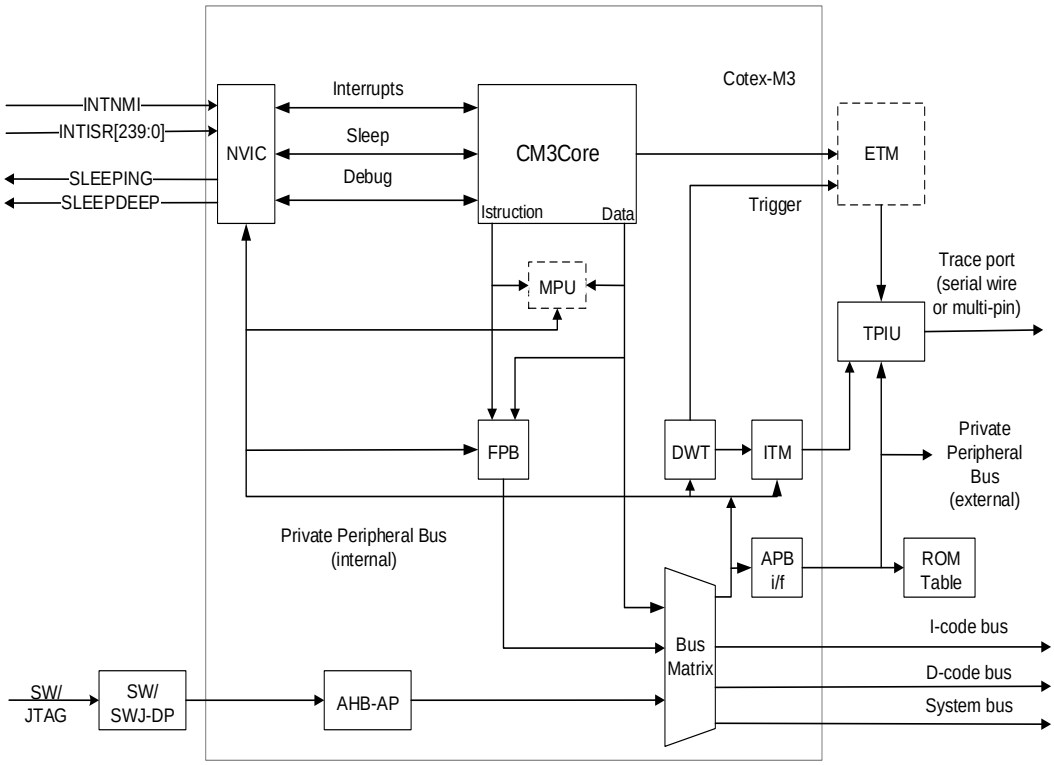
1.1. Arm Cortex-M3 内核

Arm® Cortex®-M3 处理器是一个具有低中断延迟时间和低成本调试特性的 32 位处理器。高集成度和增强的特性使 Cortex®-M3 处理器适合于需要高性能和低功耗微控制器的市场领域。Cortex®-M3 处理器基于 Armv7 架构，并且支持一个强大且可扩展的指令集，包括通用数据处理 I/O 控制任务和增强的数据处理位域操作。下面列出由 Cortex®-M3 提供的一些系统外设：

- 内部总线矩阵，用于实现 I-Code 总线、D-Code 总线、系统总线、专用总线(PPB)以及调试专用总线（AHB-AP）的互联；
- 嵌套向量中断控制器（NVIC）；
- 闪存地址重载及断点单元（FPB）；
- 数据观测点及跟踪单元（DWT）；
- 指令跟踪宏单元（ITM）；
- 串行线 JTAG 调试端口（SWJ-DP）；
- 跟踪端口接口单元（TPIU）；
- 嵌入式跟踪宏单元（ETM）；

*图 1-1. Cortex®-M3 结构图*如下，关于它的更详细信息，可参考 Arm® Cortex®-M3 技术参考手册。

图 1-1. Cortex®-M3 结构图

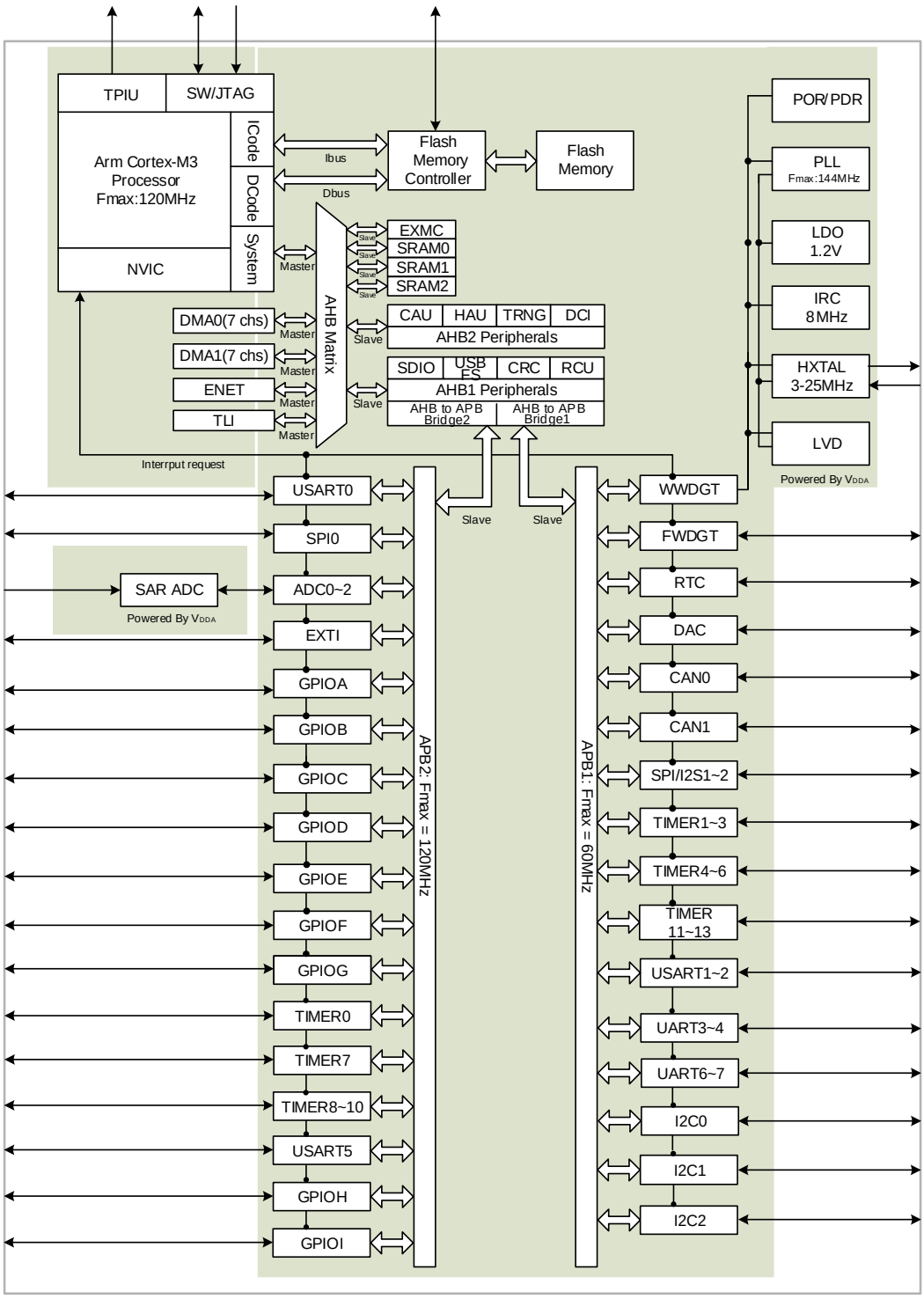


1.2. 系统架构

*图 1-2. GD32F20x Connectivity line 系列系统架构*如下图所示。AHB 总线矩阵是基于 AMBA 3.0 AHB-LITE 技术的，这是一种多层级的 AHB，它实现了系统中多主机和多从机之间的并行通信。AHB 总线矩阵包含 Cortex®-M3 内核的 ICode 总线、DCode 总线、系统总线、DMA0、DMA1、Ethernet，以及 TLDI 共 7 个主设备。ICode 总线即指令总线，同时也可以从代码存储区（0x0000 0000~0x1FFF FFFF）取出向量送至 Cortex®-M3 内核。DCode 总线即数据总线，主要用于读写数据，同时也可用于代码存储区的调试访问。与之类似，系统总线用于指令和向量获取、数据加载和存储以及系统区域的调试访问。系统存储区可划分为片内 SRAM 区、外部存储映射区以及外设映射区。AHB 总线矩阵包含 flash 存储器的 ICode 总线接口和 DCode 总线接口、片内 SRAM0、SRAM1、SRAM2、片外存储器控制器，以及 AHB1 和 AHB2 共 8 个从设备。

AHB1 总线与几乎所有的 AHB 外设相连，此外还包含两条 AHB-APB 总线桥，这样可以在 AHB1 总线以及两个 APB 总线之间实现完全同步连接。两条 APB 总线则与所有的 APB 外设相连。APB1 总线的最高速度限制为 60MHz，APB2 总线则可全速运行（依器件型号而定，最高可达 120MHz）。

图 1-2. GD32F20x Connectivity line 系列系统架构



1.3. 存储器映射

Arm® Cortex®-M3 采用哈佛架构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。Cortex®-M3 内核总线宽度

为 32 位，最大寻址空间为 4GB。程序存储器、数据存储器、寄存器以及 I/O 端口均整合这个 4GB 线性寻址空间内。Cortex®-M3 提供预定义的存储器映射，这样可以降低不同器件供应商芯片的软件差异程度，减少重复性的基础工作。同时 Cortex®-M3 的系统级外设占用了一些特定区域。下表是*表 1-1 GD32F20x 系列器件的存储映射表*，包括代码、SRAM、外设以及其它预定义存储段。任何系列的每个外设被分配有 1KB 空间，这容许为每个外设简化地址代码。APB1 外设位于 0x4000 0000~0x4000 FFFF 存储段内，而 APB2 外设位于 0x4001 0000~0x4001 7FFF 存储段内。AHB1 外设位于 0x4001 8000~0x5003 FFFF 存储段内，以及 AHB2 外设位于 0x5004 0000~0x5FFF FFFF 存储段内。

表1-1 GD32F20x系列器件的存储映射表

预定义的区域	总线	地址范围	外设
外部 RAM	AHB	0xC000 0000 – 0xDFFF FFFF	EXMC - SDRAM
		0xA000 0000 - 0xA000 0FFF	保留
		0x9000 0000 - 0x9FFF FFFF	EXMC - PC CARD
		0x7000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM/SQPI-PSRAM
外设	AHB	0x5006 0C00 – 0x5FFF FFFF	保留
		0x5006 0800 – 0x5006 0BFF	TRNG
		0x5006 0400 – 0x5006 07FF	HAU
		0x5006 0000 – 0x5006 03FF	CAU
		0x5005 0400 – 0x5005 FFFF	保留
		0x5005 0000 -0x5005 03FF	DCI
		0x5004 0000 - 0x5004 FFFF	保留
		0x5000 0000 - 0x5003 FFFF	USBFS
		0x4008 0000 - 0x4FFF FFFF	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	ENET
		0x4002 6800 - 0x4002 7FFF	保留
		0x4002 6400 - 0x4002 67FF	保留
		0x4002 6000 - 0x4002 63FF	保留
		0x4002 5000 - 0x4002 5FFF	保留
		0x4002 4000 - 0x4002 4FFF	保留
		0x4002 3C00 - 0x4002 3FFF	保留
		0x4002 3800 - 0x4002 3BFF	保留
		0x4002 3400 - 0x4002 37FF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2C00 - 0x4002 2FFF	保留

预定义的区域	总线	地址范围	外设
		0x4002 2800 - 0x4002 2BFF	保留
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	保留
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 8400 - 0x4001 FFFF	保留
		0x4001 8000 - 0x4001 83FF	SDIO
	APB2	0x4001 7C00 - 0x4001 7FFF	保留
		0x4001 7800 - 0x4001 7BFF	GPIOI
		0x4001 7400 - 0x4001 77FF	GPIOH
		0x4001 7000 - 0x4001 73FF	USART5
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	TLI
		0x4001 5C00 - 0x4001 67FF	保留
		0x4001 5800 - 0x4001 5BFF	保留
		0x4001 5400 - 0x4001 57FF	TIMER10
		0x4001 5000 - 0x4001 53FF	TIMER9
		0x4001 4C00 - 0x4001 4FFF	TIMER8
		0x4001 4800 - 0x4001 4BFF	保留
		0x4001 4400 - 0x4001 47FF	保留
		0x4001 4000 - 0x4001 43FF	保留
		0x4001 3C00 - 0x4001 3FFF	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	GPIOG
		0x4001 1C00 - 0x4001 1FFF	GPIOF
		0x4001 1800 - 0x4001 1BFF	GPIOE
		0x4001 1400 - 0x4001 17FF	GPIOD
		0x4001 1000 - 0x4001 13FF	GPIOC
		0x4001 0C00 - 0x4001 0FFF	GPIOB

预定义的区域	总线	地址范围	外设
		0x4001 0800 - 0x4001 0BFF	GPIOA
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	AFIO
	APB1	0x4000 CC00 - 0x4000 FFFF	保留
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	UART7
		0x4000 7800 - 0x4000 7BFF	UART6
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	CAN1
		0x4000 6400 - 0x4000 67FF	CAN0
		0x4000 5C00 - 0x4000 63FF	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	保留
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1C00 - 0x4000 1FFF	TIMER12
		0x4000 1800 - 0x4000 1BFF	TIMER11
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM	AHB	0x2007 0000 - 0x3FFF FFFF	保留

预定义的区域	总线	地址范围	外设
		0x2006 0000 - 0x2006 FFFF	保留
		0x2002 0000 -0x2005 FFFF	SRAM2
		0x2001 C000 -0x2001 FFFF	SRAM1
		0x2000 0000 - 0x2001 BFFF	SRAM0
Code	AHB	0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option Bytes
		0x1FFF F000 - 0x1FFF F7FF	Boot loader
		0x1FFF E000 - 0x1FFF EFFF	
		0x1FFF B000 - 0x1FFF DFFF	
		0x1FFF 7A10 - 0x1FFF AFFF	保留
		0x1FFF 7800 - 0x1FFF 7A0F	保留
		0x1FFF 0000 - 0x1FFF 77FF	保留
		0x1FFE C010 - 0x1FFE FFFF	保留
		0x1FFE C000 - 0x1FFE C00F	保留
		0x1001 0000 - 0x1FFE BFFF	保留
		0x1000 0000 - 0x1000 FFFF	保留
		0x083C 0000 - 0x0FFF FFFF	保留
		0x0830 0000 - 0x083B FFFF	保留
		0x0800 0000 - 0x082F FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	Aliased to Main Flash or Boot loader
		0x0010 0000 - 0x002F FFFF	
		0x0002 0000 - 0x000F FFFF	
		0x0000 0000 - 0x0001 FFFF	

1.3.1. 位带操作

为减少“读-修改-写”操作所消耗的时间，Cortex®-M3 处理器提供了位带功能(bit-banding)，可按位为单位进行原子级的访问。存储器映射表中包含了两个位带区，分别位于 SRAM 和外设中。位带别名区把每个比特扩展成一个 32 位的字。

根据如下的映射计算公式，即可体现出位别名区的每个字如何对应于位带区的每个位。计算公式为：

$$bit_word_addr = bit_band_base + (byte_offset \times 32) + (bit_number \times 4) \quad (1\text{-}1)$$

其中：

- bit_word_addr 指的是位带区目标比特对应在位带别名区的地址；
- bit_band_base 指的是位带别名区的起始地址；
- byte_offset 指的是位带区目标比特所在的字节的字节地址偏移量；
- bit_number 指的是目标比特在对应字节中的位置(0-7)。

举例来说，要想对 0x2000 0200 的第 7 位进行原子级操作，其对应的位别名地址为：

$$bit_word_addr = 0x2200\ 0000 + (0x200 * 32) + (7 * 4) = 0x2200\ 401C \quad (1\text{-}2)$$

如果对 0x2200 401C 进行写操作，那么 0x2000 0200 的第 7 位将会相应变化；如果对 0x2200 401C 进行读操作，那么视 0x2000 0200 的第 7 位状态将会分别返回 0x0000 0001 或 0x0000 0000。

1.3.2. 片上 SRAM

GD32F20x 系列产品的片上 SRAM 起始地址是 0x2000 0000，最大容量可达 384KB，可支持字节、半字(16bits)和字(32bits)访问。片上 SRAM 被分为 SRAM0、SRAM1 和 SRAM2 等三个模块，且每个模块都有一个与 AHB 总线矩阵连接的专用接口，这意味着它们可以被同时访问。它们的位置与容量如*表 1-2. SRAM 中各模块*所示。

表1-2. SRAM中各模块

模块	容量	地址范围
SRAM1	112KB	0x2000 0000 ~ 0x2001 BFFF
SRAM2	16KB	0x2001 C000 ~ 0x2001 FFFF
SRAM3	256KB	0x2002 0000 ~ 0x2005 FFFF

1.3.3. 片上 Flash

GD32F20x 系列产品的片上 Flash 起始地址时 0x0800 0000，最大容量可达 3072KB。读操作为 0 等待，可支持字节、半字(16 bits)和字(32 bits)访问。Flash 编程以半字(16 bits)或字(32 bits)为单位。擦除可以以页(page)为单位，也可以进行全片擦除（information blocks 除外）。

1.4. 启动(Boot)选项配置

GD32F20x 系列产品提供 3 种启动(Boot)方式，可通过 BOOT1 和 BOOT0 引脚进行配置。用户可以设置 BOOT1 和 BOOT0 引脚后，进行上电复位或系统复位操作。在复位操作后的第 4 个上升沿，BOOT 引脚将被锁存，从而确定启动选项。详细说明见*表 1-3. 引导模式*。

表1-3. 引导模式

Boot模式	Boot引脚	
	Boot1	Boot0
主FLASH存储器	x	0
系统存储器	0	1
片上SRAM	1	1

上电序列或系统复位后，Arm®Cortex®-M3 处理器先从 0x0000 0000 地址获取栈顶值，再从 0x0000 0004 地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选择的引导源，主 FLASH 存储器（开始于 0x0800 0000 的原始存储空间）或系统存储器（开始于 0x1FFF B000 的原始存储空间）被映射到引导存储空间（起始于 0x0000 0000）。片上 SRAM 存储空间的起始地址是 0x2000 0000，当它被选择为引导源时，在应用初始化代

码中，你必须使用 NVIC 异常表和偏移寄存器来将向量表重定向到 SRAM 中。

Bootloader 存放在系统(System)存储内，用于对 FLASH 存储器进行重新编程。在 GD32F20x 设备中，Bootloader 可以通过 USART 或 USB 接口和外界交互，如*表 1-4. 引导程序支持外设*所示。

表1-4. 引导程序支持外设

产品线	产品	支持串行外设
互联型	GD32F205xx	USART0(PA9 PA10) USART1(PD5 PD6) USB(PA9 PA10 PA11 PA12)
	GD32F207xx	USART0(PA9 PA10) USART1(PD5 PD6) USB(PA9 PA10 PA11 PA12)

1.5. 设备电子签名

互联型产品(GD32F20X_CL)是指闪存容量在256K至3072K之间的GD32F205xx和GD32F207xx微控制器。

设备的电子签名中包含的存储容量信息和96位的唯一设备ID，它位于Flash存储器的信息块中。96位器件唯一ID对任何器件来说都是独一无二的，可以用来作为序列号，或者密钥的一部分。

1.5.1. 产品容量信息

基地址：0x1FFF F7E0
其值出厂已设置，不可由用户更改

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容量 该值表示器件的片上SRAM容量，以Kbytes为单位。 比如：0x0020表示32 Kbytes。
15:0	FLASH_DENSITY[15:0]	FLASH存储器容量 该值表示器件的FLASH存储器容量，以Kbytes为单位。 比如：0x0020表示32 Kbytes。

1.5.2. 器件唯一 ID（96 位）

基地址：0x1FFF F7E8
其值出厂已设置，不可由用户更改

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
31:16	UNIQUE_ID[31:16]	器件唯一ID，此域可保留为其他功能
15:0	UNIQUE_ID[15:0]	器件唯一ID

地址偏移：0x04
其值出厂已设置，不可由用户更改

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]	器件唯一ID

地址偏移：0x08
其值出厂已设置，不可由用户更改

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]	器件唯一ID

1.6. 系统配置寄存器

基地址：0x4002 103C

其值出厂已设置，不可由用户更改

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	CEE	代码执行效率 0：默认的代码执行效率。 1：增强的代码执行效率。
6:0	保留	必须保持复位值。

注意：只有bit[7]可以被读-改-写，其他的位都不允许。

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在闪存的前384K字节空间内，CPU执行指令零等待（在闪存大小等于256K或512K时，闪存全片执行指令零等待）。FMC也提供了页擦除，整片擦除，以及32位整字或16位半字编程闪存等操作。

2.2. 主要特征

- 高达3072KB的片上闪存可用于存储指令或数据；
- 在闪存的前384K字节空间内，CPU执行指令零等待（在闪存大小等于256K或512K时，闪存全片执行指令零等待），在此范围外，CPU提取指令存在较长延时；
- 对于闪存容量大于512KB的GD32F20x_CL，使用了两片闪存；前512KB容量在第一片闪存（bank0）中，后续的容量在第二片闪存（bank1）中；
- 对于主存储闪存容量不多于512KB的GD32F20x_CL，只使用了bank0；
- 对于闪存容量大于512KB的GD32F20x_CL，bank0的闪存页大小为2KB，bank1的闪存页大小为4KB；
- 支持32位整字或16位半字编程，页擦除和整片擦除操作；
- 大小为16字节的选项字节可根据用户需求配置；
- 当系统复位时，选项字节被上载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作。

2.3. 功能说明

2.3.1. 闪存结构

对于主存储闪存容量不多于512KB的GD32F20x_CL，闪存页大小为2KB。对于主存储闪存容量不少于768KB的GD32F20x_CL，使用了两片闪存；前512KB容量在第一片闪存（bank0）中，后续的容量在第二片闪存（bank1）中。其中bank0的闪存页大小为2KB，bank1的闪存页大小为4KB。主存储闪存的每页都可以单独擦除。闪存结构见*表2-1. GD32F20x_CL*。

表 2-1. GD32F20x_CL

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第0页	0x0800 0000 - 0x0800 07FF	2KB
	第1页	0x0800 0800 - 0x0800 0FFF	2KB
	第2页	0x0800 1000 - 0x0800 17FF	2KB

闪存块	名称	地址范围	大小（字节）
	第255页	0x0807 F800 - 0x0807 FFFF	2KB
	第256页	0x0808 0000 - 0x0808 0FFF	4KB
	第257页	0x0808 1000 - 0x0808 1FFF	4KB
	第895页	0x082F F000 - 0x082F FFFF	4KB
信息块	引导装载程序	0x1FFF B000- 0x1FFF F7FF	18KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B

注意：信息块存储了boot loader，不能被用户编程或擦除。

2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据分别使用CPU的IBUS或DBUS总线。

2.3.3. FMC_CTLx 寄存器解锁

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

FMC_CTL0寄存器的OBPG位和OBER位在FMC_CTL0寄存器解锁后，仍然被保护。向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后硬件将FMC_CTL0寄存器的OBWEN位置1。软件可以将FMC_CTL0的OBWEN位清0来锁定FMC_CTL0的OBPG位和OBER位。

对于闪存大于512KB的GD32F20x_CL，FMC_CTL0寄存器用来设置对bank0和选项字节块的操作，FMC_CTL1寄存器用来设置对bank1的擦写操作。FMC_CTL1的解锁和锁定机制和FMC_CTL0类似。对FMC_KEY1写解锁序列可解除FMC_CTL1的锁定。

2.3.4. 页擦除

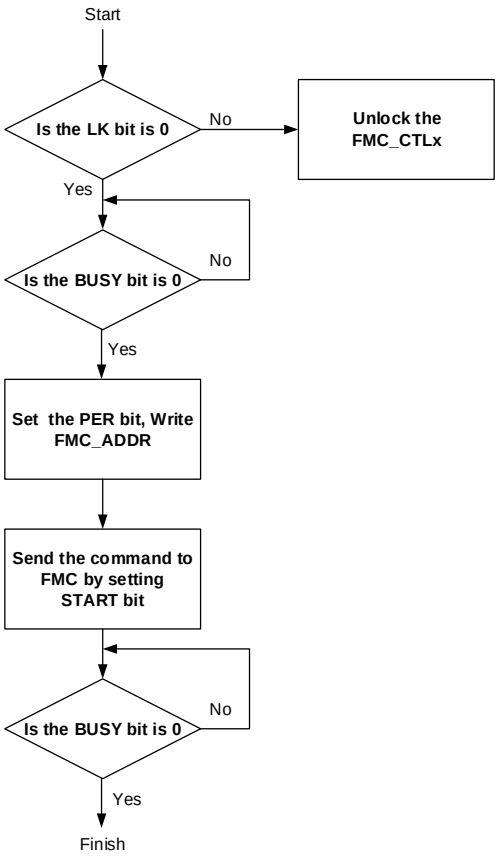
FMC的页擦除功能使得主存储闪存的页内容初始化为高电平。 每一页都可以被独立擦除，而不影响其他页内容。页擦除操作，寄存器设置具体步骤如下：

1. 确保FMC_CTLx寄存器不处于锁定状态；
2. 检查FMC_STATx寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
3. 置位FMC_CTLx寄存器的PER位；

4. 将待擦除页的绝对地址（0x08XX XXXX）写到FMC_ADDRx寄存器；
5. 通过将FMC_CTLx寄存器的START位置1来发送页擦除命令到FMC；
6. 等待擦除指令执行完毕，FMC_STATx寄存器的BUSY位清0；
7. 如果需要，使用DBUS读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STATx寄存器的ENDF位将置位。若FMC_CTLx寄存器的ENDIE位被置1，则FMC将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除地址。否则当待擦除页的地址被用来取指令或访问数据时，软件将会跑飞。该情况下，FMC不会提供任何出错通知。另一方面，对擦写保护的页进行擦除操作将无效。如果FMC_CTLx寄存器的ERRIE位被置位，该操作将触发操作出错中断。中断服务程序可通过检测FMC_STATx寄存器的WPERR位来判断该中断是否发生。*图2-1. 页擦除操作流程*显示了页擦除操作流程。

图 2-1. 页擦除操作流程



对于闪存大于512KB的GD32F20x_CL，FMC_STAT0寄存器反应对bank0和选项字节块的操作状态，FMC_STAT1反应对bank1的操作状态。对bank1的页擦除操作与对bank0的页擦除操作类似。需要注意的是，在安全保护状态下，对bank1的页擦除，需将地址同时写至FMC_ADDR1和FMC_ADDR0寄存器。

2.3.5. 整片擦除

FMC提供了整片擦除功能可以初始化主存储闪存块的内容。当FMC_CTL0寄存器中MER被置1时，擦除过程仅作用于bank0，当FMC_CTL1寄存器中MER被置1时，擦除过程仅作用于bank1，

当FMC_CTL0和FMC_CTL1寄存器中MER都被置1时，擦除过程作用于整片闪存。整片擦除操作，寄存器设置具体步骤如下：

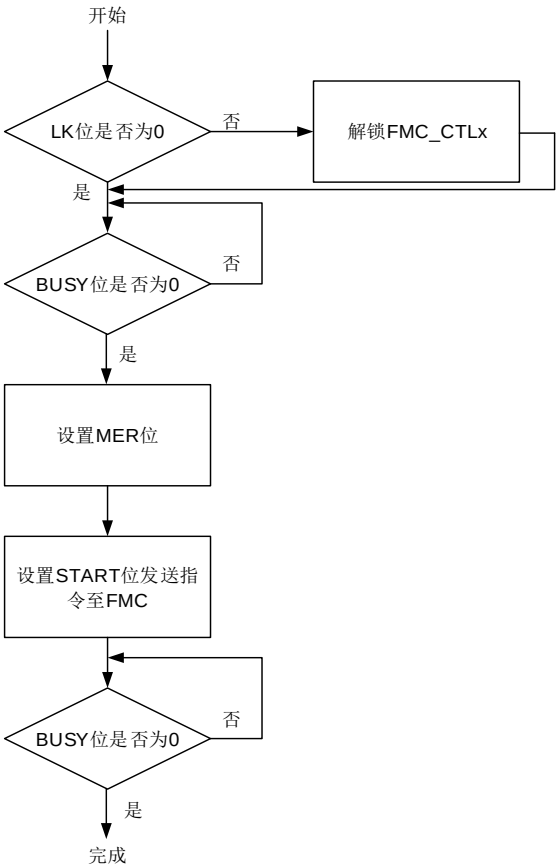
1. 确保FMC_CTLx寄存器不处于锁定状态；
2. 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
3. 如果单独擦除Bank0，置位FMC_CTL0寄存器的MER位。如果单独擦除Bank1，置位FMC_CTL1寄存器的MER位。如果整片擦除闪存，同时置位FMC_CTL0和FMC_CTL1寄存器的MER位；
4. 通过将FMC_CTLx寄存器的START位置1来发送整片擦除命令到FMC；
5. 等待擦除指令执行完毕，FMC_STATx寄存器的BUSY位清0；
6. 如果需要，使用DBUS读并验证是否擦除成功。

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

对于闪存大于512KB的GD32F20x_CL，对bank1的整片擦除操作与对bank0的整片擦除操作类似。

*图2-2. 整片擦除操作流程*显示了整片擦除操作流程。

图 2-2. 整片擦除操作流程



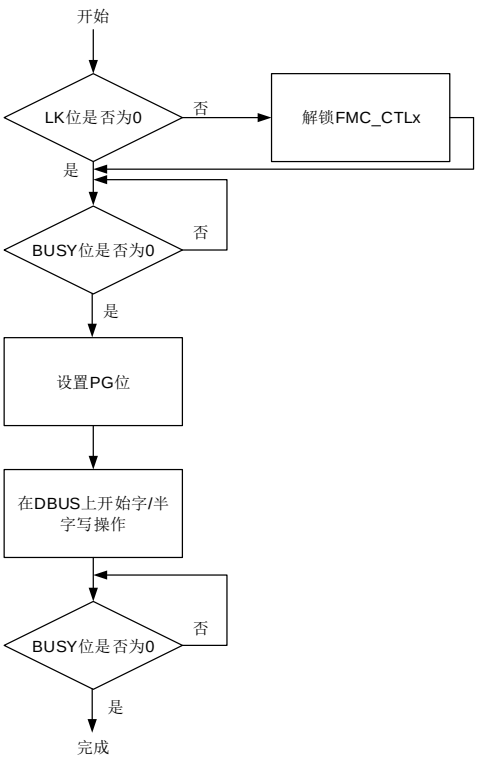
2.3.6. 主存储闪存块编程

FMC提供了一个32位整字/16位半字编程功能，用来修改主存储闪存块内容。编程操作使用各寄存器流程如下。

1. 确保FMC_CTLx寄存器不处于锁定状态；
2. 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
3. 置位FMC_CTLx寄存器的PG位；
4. DBUS写一个32位整字/16位半字到目的绝对地址（0x08XX XXXX）；
5. 等待编程指令执行完毕，FMC_STATx寄存器的BUSY位清0；
6. 如果需要，使用DBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STATx寄存器的ENDF位置位。若FMC_CTLx寄存器的ENDIE位被置1，FMC将触发一个中断。需要注意的是，执行整字/半字编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除，对该地址写一个非0x0值，FMC_STATx寄存器的PGERR位将被置1。注意在执行整字/半字编程操作前必须将PG位置1。另一方面，如果目的地址在一个处于擦除和编程保护的页中，编程不会成功且FMC_STATx寄存器的WPERR位将会置位。在这两种情形下，如果FMC_CTLx寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。在中断服务程序中，可以检查FMC_STATx寄存器的PGERR位和WPERR位来判断哪一种错误发生了。*图2-3. 字编程操作流程*显示了主存储块编程操作流程。

图 2-3. 字编程操作流程



对于闪存大于512KB的GD32F20x_CL，对bank1的编程操作与对bank0的编程操作类似。

注意：避免在同一个bank中既进行读操作，又进行擦除或编程操作。当CPU进入省电模式时，

对闪存的操作将失败。

2.3.7. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节块。选项块擦除过程如下所示。

1. 确保FMC_CTL0寄存器不处于锁定状态；
2. 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
3. 解锁FMC_CTL0寄存器的选项字节操作位；
4. 等待FMC_CTL0寄存器的OBWEN位置1；
5. 置位FMC_CTL0寄存器的OBER位；
6. 通过将FMC_CTL0寄存器的START位置1来发送可选字节块擦除命令到FMC；
7. 等待擦除指令执行完毕，FMC_STAT0寄存器的BUSY位清0；
8. 如果需要，使用DBUS读并验证是否擦除成功。

当选项字节块擦除成功执行，FMC_STAT0寄存器的ENDF位置位。若FMC_CTL0寄存器的ENDIE位被置1，FMC将触发一个中断。

2.3.8. 选项字节块编程

FMC提供了一个32位整字/16位半字编程功能，可用来修改选项字节块内容。选项字节块共有8对选项字节。每对选项字节的高字节是低字节的补码。当低字节被修改时，FMC自动生成该选项字节的高字节。擦除操作过程如下。

1. 确保FMC_CTL0寄存器不处于锁定状态；
2. 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
3. 解锁FMC_CTL0寄存器的可选字节操作位；
4. 等待FMC_CTL0寄存器的OBWEN位置1；
5. 置位FMC_CTL0寄存器的OBPG位；
6. DBUS写一个32位整字/16位半字到目的地址；
7. 等待编程指令执行完毕，FMC_STAT0寄存器的BUSY位清0；
8. 如果需要，使用DBUS读并验证是否编程成功。

当选项字节块编程成功执行，FMC_STAT0寄存器的ENDF位置位。若FMC_CTL0寄存器的ENDIE位被置1，FMC将触发一个中断。需要注意的是，执行整字/半字编程操作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，对该地址写一个非0x0值，FMC_STAT0寄存器的PGERR位将被置1。

被修改的选项字节在系统复位后生效。

2.3.9. 选项字节块说明

每次系统复位后，闪存的选项字节块被重加载到FMC_OBSTAT和FMC_WP寄存器，可选字节生效。选项字节的补字节具体为选项字节取反。当选项字节被重装载时，如果选项字节的补字

节和选项字节不匹配，FMC_OBSTAT寄存器的OBERR位将被置1，选项字节被强制设置为0xFF。若选项字节和其补字节同为0xFF，则OBERR位不置位。选项字节详情见*表2-2. 选项字节*。

表 2-2. 选项字节

地址	名称	说明
0x1fff f800	OB_SPC	选项字节安全保护值 0xA5：未保护状态 除0xA5外的任何值：已保护状态
0x1fff f801	OB_SPC_N	OB_SPC补字节
0x1fff f802	OB_USER	[7:4]：保留 [3]：BB 0：当配置为从主存储块启动时，若bank1有启动程序，从bank1启动，否则从bank0启动。 1：当配置为从主存储块启动时，从bank0启动。 [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式。 1：设置待机模式时进入待机模式而不产生复位。 [1]：nRST_DPSLP 0：设置深度睡眠模式时产生复位而不进入深度睡眠模式 1：设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]：nWDG_HW 0：硬件使能独立看门狗功能 1：软件使能独立看门狗功能
0x1fff f803	OB_USER_N	OB_USER补字节值
0x1fff f804	OB_DATA[7:0]	用户定义数据7到0位
0x1fff f805	OB_DATA_N[7:0]	OB_DATA补字节值的7到0位
0x1fff f806	OB_DATA[15:8]	用户定义数据15到8位
0x1fff f807	OB_DATA_N[15:8]	OB_DATA补字节值的15到8位
0x1fff f808	OB_WP[7:0]	页擦除/编程保护值的7到0位 0：保护生效 1：未保护
0x1fff f809	OB_WP_N[7:0]	OB_WP补字节值的7到0位
0x1fff f80a	OB_WP[15:8]	页擦除/编程的保护值的15到8位
0x1fff f80b	OB_WP_N[15:8]	OB_WP补字节值的15到8位
0x1fff f80c	OB_WP[23:16]	页擦除/编程的保护值的23到16位
0x1fff f80d	OB_WP_N[23:16]	OB_WP补字节值的23到16位
0x1fff f80e	OB_WP[31:24]	页擦除/编程的保护值的31到24位 OB_WP[30:24]：每个bit可设置4KB闪存的保护状态，对于GD32F20x_CL是2页闪存。第0位设置前4KB闪存的保护状态，以此类推。这31位总计可设置前124KB的闪存保护状态。 OB_WP[31]：第31位可设置闪存剩下部分的保护状态。
0x1fff f80f	OB_WP_N[31:24]	OB_WP补字节值的31到24位

2.3.10. 页擦除/编程保护

FMC的页擦除/编程保护功能可以阻止对闪存的意外操作。当FMC对被保护页进行页擦除或编程操作时，操作本身无效且FMC_STATx寄存器的WPERR位将被置1。如果WPERR位被置1且FMC_CTLx寄存器的ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作出错中断，等待CPU处理。配置选项字节块的OB_WP[31:0]某位为0可以单独使能某几页的保护功能。如果在选项字节块执行了擦除操作，所有的闪存页擦除和编程保护功能都将失效。当选项字节的OB_WP被改变时，需要系统复位使之生效。

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。

未保护状态：当将OB_SPC字节和它的补字节被设置为0x5AA5，系统复位以后，闪存将处于非安全保护状态。主存储块和选项字节块可以被所有操作模式访问。

已保护状态：当设置OB_SPC字节和它的补字节值为任何除0x5AA5外的值，系统复位以后，安全保护状态生效。需要注意的是，若该修改过程中，MCU的调试模块依然和外部JTAG/SWD设备相连，需要用上电复位代替系统复位以使得修改后的保护状态生效。在安全保护状态下，主存储闪存块仅能被用户代码访问且前4KB的闪存自动处于页擦除/编程保护状态下。在调试模式下，或从SRAM中启动时，以及从boot loader区启动时，这些模式下对主存储块的操作都被禁止。如果在这些模式下读主存储块，将产生总线错误。如果在这些模式下，对主存储块进行编程或擦除操作，FMC_STAT寄存器的WPERR位将被置1。但这些模式下都可以对选项字节块进行操作，从而可以通过该方式失能安全保护功能。如果将OB_SPC字节和它的补字节设置为0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

注意：在读保护情况下，flash前4k不能被擦写。

2.4. FMC 寄存器

FMC基地址：0x4002 2000

2.4.1. 等待状态寄存器（FMC_WS）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:3	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。FMC_WSEN 寄存器的 WSEN 位被置 1 时 WSCNT 位有效。 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 ~ 111：保留

2.4.2. 解锁寄存器 0（FMC_KEY0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL0 解锁寄存器 这些位仅能被软件写。

写解锁值到 KEY[31:0]可以解锁 FMC_CTL0 寄存器。

2.4.3. 选项字节操作解锁寄存器（FMC_OBKEY）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_CTL0 选项字节操作解锁寄存器 这些位仅能被软件写 写解锁值到 OBKEY[31:0]解锁 FMC_CTL0 寄存器的选项字节命令。

2.4.4. 状态寄存器 0（FMC_STAT0）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置 1。软件写 1 清 0。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置 1。软件写 1 清 0。
3	保留	必须保持复位值。
2	PGERR	编程错误标志位

		当被编程区域状态不为 0xFFFF 时，对闪存编程，此位被硬件置 1。软件写 1 清 0。
1	保留	必须保持复位值。
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置 1。当操作结束或者出错，此位被清 0。

2.4.5. 控制寄存器 0（FMC_CTL0）

地址偏移：0x10

复位值：0x0000 0080

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ENDIE	操作结束中断使能位 软件置1和清0 0：无硬件中断产生 1：使能操作结束中断
11	保留	必须保持复位值。
10	ERRIE	出错中断使能位 软件置1和清0 0：无硬件中断产生 1：使能出错中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入FMC_OBKEY寄存器，此位由硬件置1。此位可以被软件清0。
8	保留	必须保持复位值。
7	LK	FMC_CTL0寄存器锁定标志位 当正确的序列写入FMC_KEY0寄存器，此位由硬件清0。此位可以由软件置1。
6	START	向FMC发送擦除命令位 软件置1可以发送擦除命令到FMC。当BUSY位被清0时，此位由硬件清0。
5	OBER	选项字节擦除命令位 软件置1和清0 0：无作用

		1：选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置1和清0 0：无作用 1：选项字节编程命令
3	保留	必须保持复位值。
2	MER	主存储块整片擦除命令位 软件置1和清0 0：无作用 1：主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置1和清0 0：无作用 1：主存储块页擦除命令
0	PG	主存储块编程命令位 软件置1和清0 0：无作用 1：主存储块编程命令

注意：当相应闪存操作完成后，该寄存器需处于复位状态。

2.4.6. 地址寄存器 0（FMC_ADDR0）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

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

2.4.7. 选项字节状态寄存器（FMC_OBSTAT）

地址偏移：0x1C

复位值：0x0XXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						DATA[15:6]									
						r									

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

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:10	DATA[15:0]	系统复位后保存选项字节块的 OB_DATA[15:0]部分
9:2	USER[7:0]	系统复位后保存选项字节块的 OB_USER 字节
1	SPC	安全保护状态 0：未保护 1：已保护
0	OBERR	选项字节读错误位 当选项字节和它的补字节不匹配时此位由硬件置 1，选项字节被强制设置为 0xFF。

2.4.8. 擦除/编程保护寄存器（FMC_WP）

地址偏移：0x20
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	WP[31:0]	系统复位后保存选项字节块的 OB_WP[31:0]部分

2.4.9. 解锁寄存器 1（FMC_KEY1）

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
KEY[31:16]															
w															

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

位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL1 解锁寄存器 这些位仅能被软件写 写解锁值到 KEY[31:0]可以解锁 FMC_CTL1 寄存器

2.4.10. 状态寄存器 1（FMC_STAT1）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置 1。软件写 1 清 0。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置 1。软件写 1 清 0。
3	保留	必须保持复位值。
2	PGERR	编程错误标志位 当被编程区域状态不为 0xFFFF 时，对闪存编程，此位被硬件置 1。软件写 1 清 0。
1	保留	必须保持复位值。
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置 1。当操作结束或者出错，此位被清 0。

2.4.11. 控制寄存器 1（FMC_CTL1）

地址偏移：0x50

复位值：0x0000 0080

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	ENDIE	操作结束中断使能位 软件置 1 和清 0 0：无硬件中断产生 1：使能操作结束中断
11	保留	必须保持复位值。
10	ERRIE	出错中断使能位 软件置 1 和清 0 0：无硬件中断产生 1：使能出错中断
9:8	保留	必须保持复位值。
7	LK	FMC_CTL1 寄存器锁定标志位 当正确的序列写入 FMC_KEY1 寄存器，此位由硬件清 0。此位可以由软件置 1。
6	START	发送擦除命令到 FMC 位 软件置 1 可以发送擦除命令到 FMC。当 BUSY 位被清 0 时，此位由硬件清 0。
5:3	保留	必须保持复位值。
2	MER	主存储块整片擦除命令位 软件置 1 和清 0 0：无作用 1：主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置 1 和清 0 0：无作用 1：主存储块页擦除命令
0	PG	主存储块编程命令位

软件置 1 和清 0

0：无作用

1：主存储块编程命令

注意：当相应闪存操作完成后，该寄存器需处于复位状态。

2.4.12. 地址寄存器 1（FMC_ADDR1）

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

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

2.4.13. 等待状态使能寄存器（FMC_WSEN）

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															WSEN
															rw

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	WSEN	FMC 等待状态使能寄存器 此位由软件置 1 和清 0。此位也被 FMC_KEYx 寄存器保护。需要写 0x45670123 和 0xCDEF89AB 到 FMC_KEYx 寄存器。 0：从闪存取指无等待状态 1：从闪存取指增加等待状态

2.4.14. 产品 ID 寄存器（FMC_PID）

地址偏移：0x100
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

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

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

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

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32F20x系列产品比较注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式，深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如*图3-1. 电源域概览*所示，GD32F20x系列设备有三个电源域，包括V_{DD} / V_{DDA}域，1.2V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1.2V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。

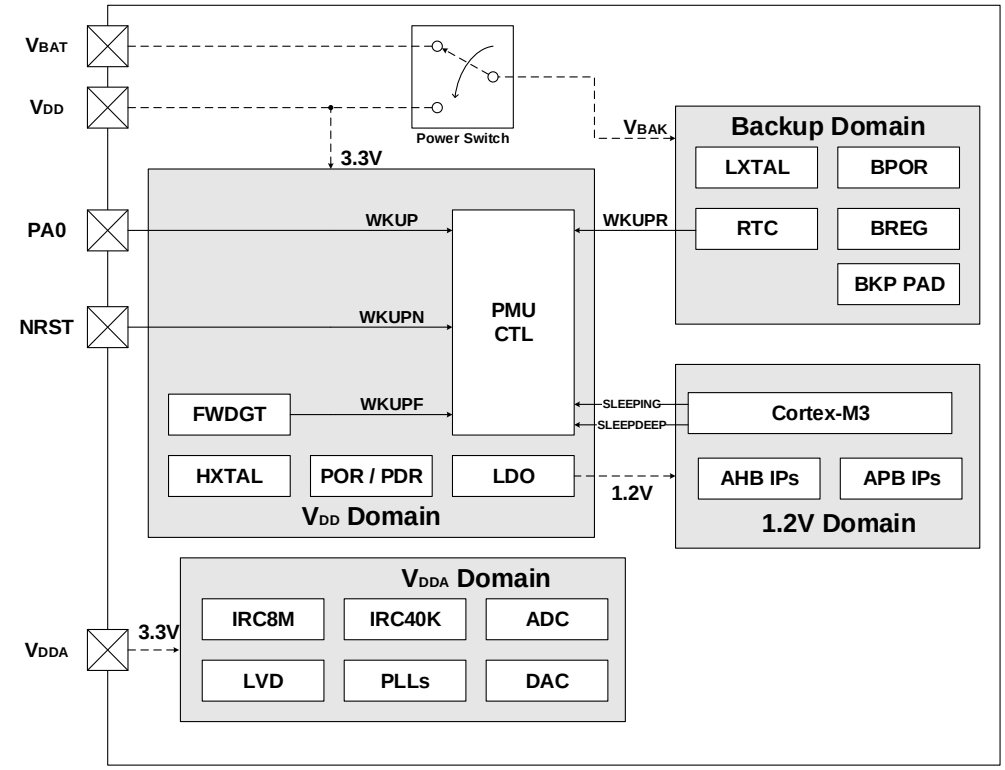
3.2. 主要特征

- 三个电源域：备份域、V_{DD} / V_{DDA}域和1.2V电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器（LDO）提供1.2V电源；
- 提供低电压检测器（LVD），当电压低于所设定的阈值时能发出中断或事件；
- 当V_{DD}供电关闭时，由V_{BAT}（电池）为备份域供电。

3.3. 功能说明

*图3-1. 电源域概览*提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



LVD：低压检测器　　LDO：电压调节器　　BPOR：备份域上电复位
POR：上电复位　　PDR：掉电复位　　BREG：备份寄存器

3.3.1. 电池备份域

电池备份域由内部电源切换器来选择V_{DD}供电或V_{BAT}（电池）供电，然后由V_{BAK}为备份域供电，该备份域包含RTC（实时时钟）、LXTAL（低速外部晶体振荡器）、BPOR（备份域上电复位）、BREG（备份寄存器），以及PI8、PC13至PC15共4个BKP PAD。为了确保备份域中寄存器的内容及RTC正常工作，当V_{DD}关闭时，V_{BAT}引脚可以连接至电池或其他备份电源供电。电源切换器是由V_{DD} / V_{DDA}域掉电复位电路控制的。对于没有外部电池的应用，建议将V_{BAT}引脚通过100nF的外部陶瓷去耦电容连接到V_{DD}引脚上。

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

RTC的时钟源可以是低速内部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®-M3需要通过RTC寄存器设置预期的唤醒时间并启用唤醒功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在*实时时钟（RTC）*章节描述。

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

- PC13可以作为通用I / O口或RTC功能引脚（参见*实时时钟（RTC）*）；

- PC14和PC15可以作为通用I / O口或LXTAL晶振引脚；
- PI8可以作为通用I / O口或RTC功能引脚（参见*实时时钟（RTC）*）。

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

- PC13仅可以作为RTC功能引脚（参见*实时时钟（RTC）*）；
- PC14和PC15仅可作为LXTAL晶振引脚；
- PI8仅可以作为RTC功能引脚（参见*实时时钟（RTC）*）。

注意：由于PC13、PC14、PC15和PI8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到PC14和PI8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）。

3.3.2. VDD / VDDA 电源域

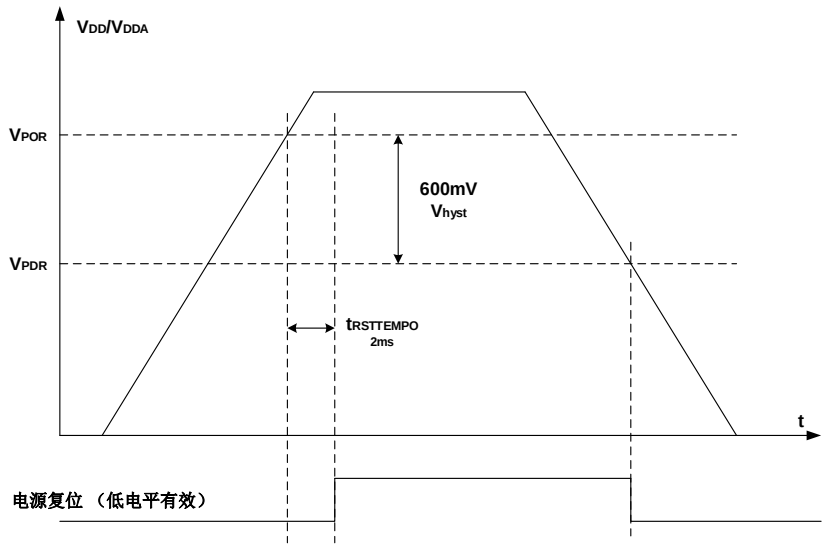
V_{DD} / V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。V_{DD} 域包括 HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、POR / PDR（上电/掉电复位）、FWDGT（独立看门狗定时器）和除 PC13、PC14、PC15 和 PI8 之外的所有 PAD 等等。V_{DDA} 域包括 ADC / DAC（AD / DA 转换器）、IRC8M（内部 8M RC 振荡器）、IRC40K（内部 40KHz RC 振荡器）、PLLs（锁相环）和 LVD（低电压检测器）等等。

V_{DD} 域

为 1.2V 域供电的 LDO（电压调节器），其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式（全供电状态）、深度睡眠模式（全供电或低功耗状态）和待机模式（关闭状态）。

POR / PDR（上电/掉电复位）电路检测 V_{DD} / V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 3-2. 上电/掉电复位波形图*显示了供电电压和电源复位信号之间的关系。VPOR 表示上电复位的阈值电压，典型值约为 2.40V，V_{PDR} 表示掉电复位的阈值电压，典型值约为 1.80V。迟滞电压 Vhyst 值约为 600mV。

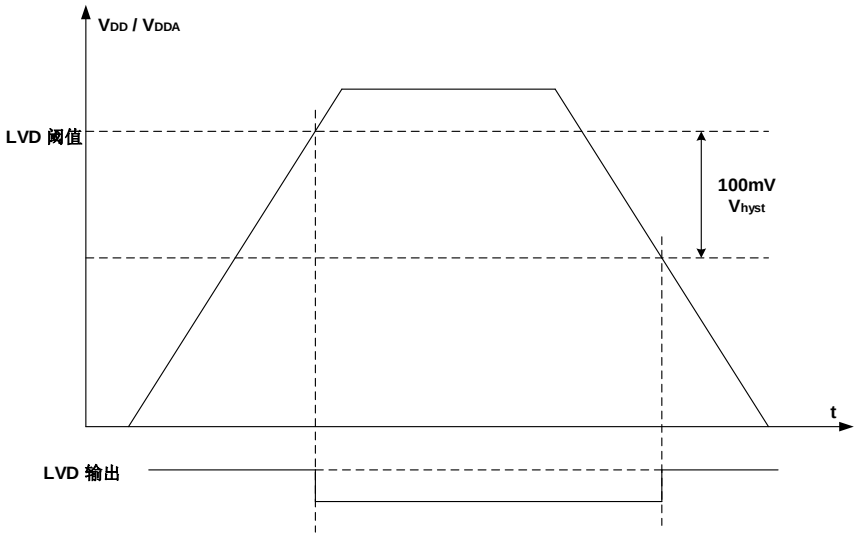
图3-2. 上电/掉电复位波形图



V_{DDA} 域

LVD 的功能是检测 V_{DD} / V_{DDA} 供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器（PMU_CTL）中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器（PMU_CS）中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 3-3. LVD 阈值波形图*显示了 V_{DD} / V_{DDA} 供电电压和 LVD 输出信号的关系。（LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置）。迟滞电压 Vhyst 值为 100mV。

图3-3. LVD阈值波形图



一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，V_{DDA} 通过外部滤波电路连接至 V_{DD}，相应的 V_{SSA} 通过特定电路连接至 V_{SS}。否则，如果 V_{DDA} 和 V_{DD} 不同时，V_{DDA} 须高于 V_{DD}，但压差不超过 0.3V。

为提高 ADC 和 DAC 的精度，可将独立的外部参考电压连接至 ADC / DAC 引脚 VREF+ / VREF-。根据不同的封装，VREF+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 14-2. ADC 输入引脚定义*和*表 15-1. DAC I/O 描述*，V_{REF-}须被连接至 V_{SSA} 引脚。V_{REF+}引脚仅存在于不小于 100-pin 的封装上，而在 64-pin 或更少引脚的封装上不存在，因其内部已经连接至 V_{DDA}，V_{REF-}内部则直接连接至 V_{SSA}。

3.3.3. 1.2V 电源域

1.2V电源域为Cortex®-M3内核逻辑、AHB / APB外设、备份域和V_{DD} / V_{DDA}域的APB接口等供电。当1.2V电压上电后，POR将在1.2V域中产生一个复位序列，复位完成后，如果要进入指定的省电模式，须先配置相关的控制位，之后一旦执行WFI或WFE指令，设备便进入该省电模式。关于这方面的详细内容，将在以下章节予以说明。

3.3.4. 省电模式

系统复位或电源复位后，GD32F20x MCU处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有：减慢系统时钟（HCLK，PCLK1，PCLK2），关闭未使用的外设的时钟。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

睡眠模式

睡眠模式与Cortex®-M3的SLEEPING模式相对应。在睡眠模式下，仅关闭Cortex®-M3的时钟。如需进入睡眠模式，只要清除Cortex®-M3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®-M3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®-M3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®-M3 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，1.2V 域中的所有时钟全部关闭，IRC8M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M3 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何中断都可以唤醒系统，请参考 Cortex®-M3 技术手册）。刚退出深度睡眠模式时，IRC8M 被选中作为系统时钟。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考*表 6-3. EXTI 触发源*。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

待机模式

待机模式是基于 Cortex®-M3 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.2V 域全部停止供电，同时 LDO 和包括 IRC8M、HXTAL 和 PLLs 也会被关闭。进入待机模式前，先将 Cortex®-M3 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL 寄存器的 STBMOD 位置 1，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有四个唤醒源，包

括来自 NRST 引脚的外部复位，RTC 报警，FWDGT 复位，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 1.2V 电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex®-M3 将从 0x00000000 地址开始执行指令代码。

表3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1. 关闭 1.2V 电源域的所有时钟 2. 关闭 IRC8M、HXTAL 和 PLL	1. 关闭 1.2V 电源域的供电 2. 关闭 IRC8M、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）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1， STBMOD = 0	SLEEPDEEP = 1， STBMOD = 1，WURST = 1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND = 1 时的中断）均可唤醒。	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND = 1 时的中断）可唤醒。	1. NRST 引脚 2. WKUP 引脚 3. FWDGT 复位 4. RTC
唤醒延迟	无	IRC8M 唤醒时间 如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间。	上电序列

注意：在待机模式下，除了 NRST 引脚，配置为 RTC 功能的 PC13 和 PI8，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUP 引脚，其他所有 I / O 都处于高阻态。

3.4. PMU 寄存器

PMU 基地址：0x4000 7000

3.4.1. 控制寄存器（PMU_CTL）

地址偏移：0x00
复位值：0x0000 0000（从待机模式唤醒后复位）

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	BKPWEN	备份域写使能 0：禁止对备份域寄存器的写访问 1：允许对备份域寄存器的写访问 复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000：2.2V 001：2.3V 010：2.4V 011：2.5V 100：2.6V 101：2.7V 110：2.8V 111：2.9V
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®-M3进入SLEEPDEEP模式时，系统进入深度睡眠模式 1：当Cortex®-M3进入SLEEPDEEP模式时，系统进入待机模式
0	LDOLP	LDO低功耗模式 0：当系统进入深度睡眠模式时，LDO仍正常工作 1：当系统进入深度睡眠模式时，LDO进入低功耗模式

3.4.2. 电源控制和状态寄存器（PMU_CS）

地址偏移：0x04

复位值：0x0000 0000（从待机模式唤醒后不复位）

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	WUPEN	WKUP引脚唤醒使能 0：关闭WKUP引脚唤醒功能 1：开启WKUP引脚唤醒功能 如果WUPEN在进入待机模式之前置1，WKUP引脚的上升沿会将系统从待机模式唤醒。由于WKUP引脚为高电平有效，WKUP引脚内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
7:3	保留	必须保持复位值。
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD}高于设定的LVD阈值） 1：低电压事件出现（V_{DD}等于或低于LVD阈值） **注意：**LVD功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式 1：设备曾进入过待机模式 该位只能由POR / PDR或通过设置PMU_CTL寄存器的STBRST位来清零。

0	WUF	唤醒标志 0：没有收到唤醒事件 1：收到来自WKUP引脚或RTC闹钟事件 该位只能由POR / PDR或通过设置PMU_CTL寄存器的WURST位来清零。

4. 备份寄存器（BKP）

4.1. 简介

位于备份域中的备份寄存器可在 V_{DD} 电源关闭时由 V_{BAT} 供电，备份寄存器有 42 个 16 位（84 字节）寄存器可用来存储并保护用户应用数据，从待机模式唤醒或系统复位也不会对这些寄存器造成影响。

此外，BKP 寄存器也可实现侵入检测，RTC 校准功能和方波检测。

在复位之后,任何对备份域寄存器的写操作都将被禁止，也就是说，备份寄存器和 RTC 不允许写访问。为使能对备份寄存器和 RTC 的写访问，首先通过设置 RCU_APB1EN 寄存器的 PMUEN 和 BKPEN 位来打开电源和备份接口时钟，然后再通过设置 PMU_CTL 寄存器的 BKPWEN 位来使能对备份域中寄存器的写访问。

4.2. 主要特征

- 84字节备份寄存器用来在省电模式下保护数据。如果侵入事件发生，备份寄存器会被复位；
- 侵入源（PC13和PI8）的有效电平可配置；
- RTC时钟校准寄存器可提供RTC闹钟或秒输出选择，及设置校准值的功能；
- 侵入控制状态寄存器（BKP_TPCS）可实现侵入检测中断或事件的控制；
- 两种方波检测：PC13->PI8或者PC14->PC15。

4.3. 功能说明

4.3.1. RTC 时钟校准

为提高RTC时钟精度，MCU提供时钟输出校准功能。RTC时钟或者RTC时钟经64分频后作为输出至PC13。通过设置BKP_OCTL寄存器中的COEN位来使能此功能。

校准值通过BKP_OCTL寄存器中的RCCV[6:0]设置，校准功能可实现以1000000/2^20ppm的比例减慢RTC时钟。

4.3.2. 侵入检测 0

MCU提供侵入检测功能以保护重要的用户数据，可通过设置BKP_TPCTL寄存器中的TPEN0位来使能TAMPER0引脚（PC13）对应的功能。为防止侵入事件的丢失，边沿检测信号与TPEN0位的逻辑与作为侵入检测信号的输入，因此在TAMPER0引脚使能之前，侵入检测应该被配置。当侵入事件被检测到，对应的BKP_TPCS寄存器中的TEF0位被置位。如果侵入中断被使能，侵入事件可以产生一个中断。任何侵入事件将会复位所有备份数据寄存器。

注意：当TPAL0 = 0/1，如果TAMPER0引脚在使能（通过设置TPEN0位）之前已经为高/低，

尽管TAMPER0引脚上没有上升/下降沿信号，一个额外的侵入事件将会发生。

4.3.3. 侵入检测 1

MCU提供侵入检测功能以保护重要的用户数据，可通过设置BKP_TPCTL寄存器中的TPEN1位来使能TAMPER1引脚（PI8）对应的功能。为防止侵入事件的丢失，边沿检测信号与TPEN1位的逻辑与作为侵入检测信号的输入，因此在TAMPER1引脚使能之前，侵入检测应该被配置。当侵入事件被检测到，对应的BKP_TPCS寄存器中的TEF1位被置位。如果侵入中断被使能，侵入事件可以产生一个中断。任何侵入事件将会复位所有备份数据寄存器。

注意：当TPAL1 = 0/1，如果TAMPER1引脚在使能（通过设置TPEN1位）之前已经为高/低，尽管TAMPER1引脚上没有上升/下降沿信号，一个额外的侵入事件将会发生。

4.3.4. 方波检测

MCU提供有两个方波检测。如果置位TPM1，通过在PC13发送一个方波信号，通过在PI8接收和检查这个方波信号。如果置位TPM2，通过在PC14发送一个方波信号，通过在PC15接收和检查这个方波信号。当检查错误时，对应的BKP_TPCS寄存器中的TEF0/TEF1会被置位。如果侵入中断被使能，侵入事件可以产生一个中断。任何侵入事件将会复位所有备份数据寄存器。

注意：如果使能LXTAL，置位TPM2无效。如果置位TPM1，必须设置TPEN0和TPEN1为0。

4.4. BKP 寄存器

BKP 基地址：0x4000 6C00

4.4.1. 备份数据寄存器（BKP_DATAx）（x= 0..41）

地址偏移：0x04到0x28，0x40到0xBC
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

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

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

4.4.2. RTC 信号输出控制寄存器（BKP_OCTL）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALDIR	CCOSEL	保留				ROSEL	ASOEN	COEN	RCCV[6:0]						
rw	rw					rw	rw	rw	rw						

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CALDIR	RTC时钟校方向 0：变慢 1：变快
14	CCOSEL	RTC时钟输出选择

		0：RTC时钟64分频 1：RTC时钟
13:10	保留	必须保持复位值。
9	ROSEL	RTC输出选择 0：RTC输出为闹钟脉冲 1：RTC输出为秒脉冲 该位只能被备份域复位清除。
8	ASOEN	RTC闹钟或秒脉冲输出使能 0：RTC闹钟或秒脉冲输出禁止 1：RTC闹钟或秒脉冲输出使能 使能后，TAMPER0引脚可作为RTC输出。 该位只能被备份域复位清除。
7	COEN	RTC时钟校准输出使能 0：RTC时钟校准输出禁止 1：RTC时钟校准输出使能 使能后，TAMPER0引脚输出RTC时钟或RTC时钟的64分频。ASOEN位优先于COEN位，当ASOEN位置位时，不管COEN置位与否，TAMPER0引脚作为RTC闹钟或秒脉冲输出。 该位只能被上电/掉电复位（POR/PDR）清除。
6:0	RCCV[6:0]	RTC时钟校准值 该值表示在每2^20个时钟脉冲内将有多少个时钟脉冲被忽略。

4.4.3. 侵入引脚控制寄存器 0（BKP_TPCTL0）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														TPAL0	TPEN0
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	TPAL0	TAMPER0引脚有效电平 0：TAMPER0引脚高电平有效 1：TAMPER0引脚低电平有效

0	TPEN0	TAMPER0引脚使能 0：TAMPER0引脚作为GPIO口使用 1：TAMPER0引脚可实现备份复位功能。TAMPER0引脚上的有效电平将复位BKP_DATAx寄存器中所有数据。

4.4.4. 侵入控制状态寄存器（BKP_TPCS）

地址偏移：0x34

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIF1	TEF1	保留				TIF0	TEF0	TPIE1	TIR1	TER1	保留		TPIE0	TIR0	TER0
r	r					r	r	rw	w	w			rw	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	TIF1	侵入中断1标志 0：没有侵入中断1发生 1：有侵入中断1发生 该位可通过TIR1位置1或TPIE1位置0来清零。
14	TEF1	侵入事件1标志 0：没有侵入事件1发生 1：有侵入事件1发生 该位可通过对TER1位写1来清零。
13:10	保留	必须保持复位值。
9	TIF0	侵入中断0标志 0：没有侵入中断0发生 1：有侵入中断0发生 该位可通过TIR0位置1或TPIE0位置0来清零。
8	TEF0	侵入事件0标志 0：没有侵入事件0发生 1：有侵入事件0发生 该位可通过对TER0位写1来清零。
7	TPIE1	侵入中断1使能 0：侵入中断1禁用 1：侵入中断1使能 该位仅可通过系统复位或待机模式唤醒后复位。

6	TIR1	侵入中断1复位 0：不影响 1：复位TIF1位 该位一直读为0。
5	TER1	侵入事件1复位 0：不影响 1：复位TEF1位 该位一直读为0。
4:3	保留	必须保持复位值。
2	TPIE0	侵入中断0使能 0：侵入中断0禁用 1：侵入中断0使能 该位仅可通过系统复位或待机模式唤醒后复位。
1	TIR0	侵入中断0复位 0：不影响 1：复位TIF0位 该位一直读为0。
0	TER0	侵入事件0复位 0：不影响 1：复位TEF0位 该位一直读为0。

4.4.5. 侵入引脚控制寄存器 1（BKP_TPCTL1）

地址偏移：0x38

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
TPM1	TPM2	保留				TPAL1	TPEN1	保留							
w	w					rw	rw								

位/位域	名称	描述
31:10	保留	必须保持复位值。
15	TPM1	方波1检测使能 0：不影响 1：检测RTC时钟64分频方波，需配置CCOSEL，TPEN0和TPEN1均为0

		PC13 -> PI8
14	TPM2	方波2检测使能 0：不影响 1：检测RTC时钟64分频方波，需配置CCOSEL，TPE_1和TPE_2均为0 PC14 -> PC15
13:10	保留	必须保持复位值。
9	TPAL1	TAMPER1引脚有效电平 0：TAMPER1引脚高电平有效 1：TAMPER1引脚低电平有效
8	TPEN1	TAMPER1引脚使能 0：TAMPER1引脚作为GPIO口使用 1：TAMPER1引脚可实现备份复位功能。TAMPER1引脚上的有效电平将复位BKP_DATAx寄存器中所有数据。
7:0	保留	必须保持复位值。

5. 复位和时钟单元（RCU）

5.1. 复位控制单元（RCTL）

5.1.1. 简介

GD32F20x复位控制包括三种控制方式：电源复位、系统复位和备份域复位。电源复位又称为冷复位，将复位除了备份域以外的所有系统。系统复位将复位除了SW-DP控制器和备份域之外的其余部分，包括处理器内核和外设IP。备份域复位将复位备份区域。复位能够被外部信号、内部事件和复位发生器触发。后续章节将介绍关于这些复位的详细信息

5.1.2. 功能说明

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1.2V电压时，电源复位电平将变为无效。复位入口向量被固定在存储器映射的地址0x0000 0004。

系统复位

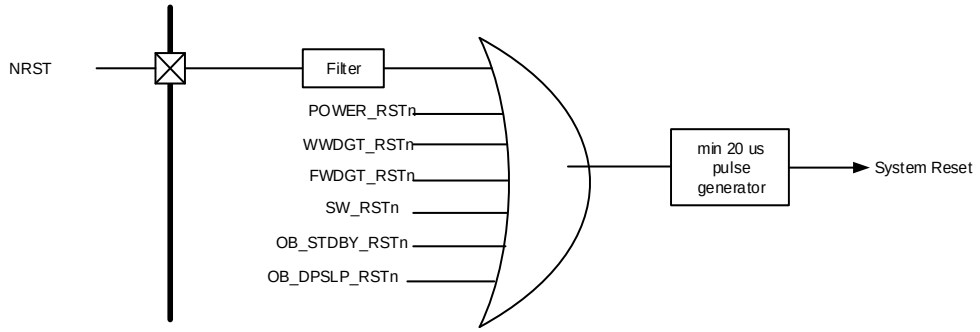
当发生以下任一事件时，产生一个系统复位：

- 上电复位（POWER_RSTn）；
- 外部引脚复位（NRST）；
- 窗口看门狗计数终止（WWDGT_RSTn）；
- 自由看门狗计数终止（FWDGT_RSTn）；
- Cortex®-M3的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（ SW_RSTn）；
- 用户选项字节中nRST_STDBY位设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）；
- 用户选项字节中nRST_DPSLP位设置为0，并且进入深度睡眠模式时产生复位（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5-1. 系统复位电路



备份域复位

以下事件之一发生时，产生备份域复位：

1、设置备份域控制寄存器中的BKPRST位为‘1’；
2、备份域电源上电复位（在V_{DD}和V_{BAT}两者都掉电的前提下，V_{DD}或V_{BAT}上电）。

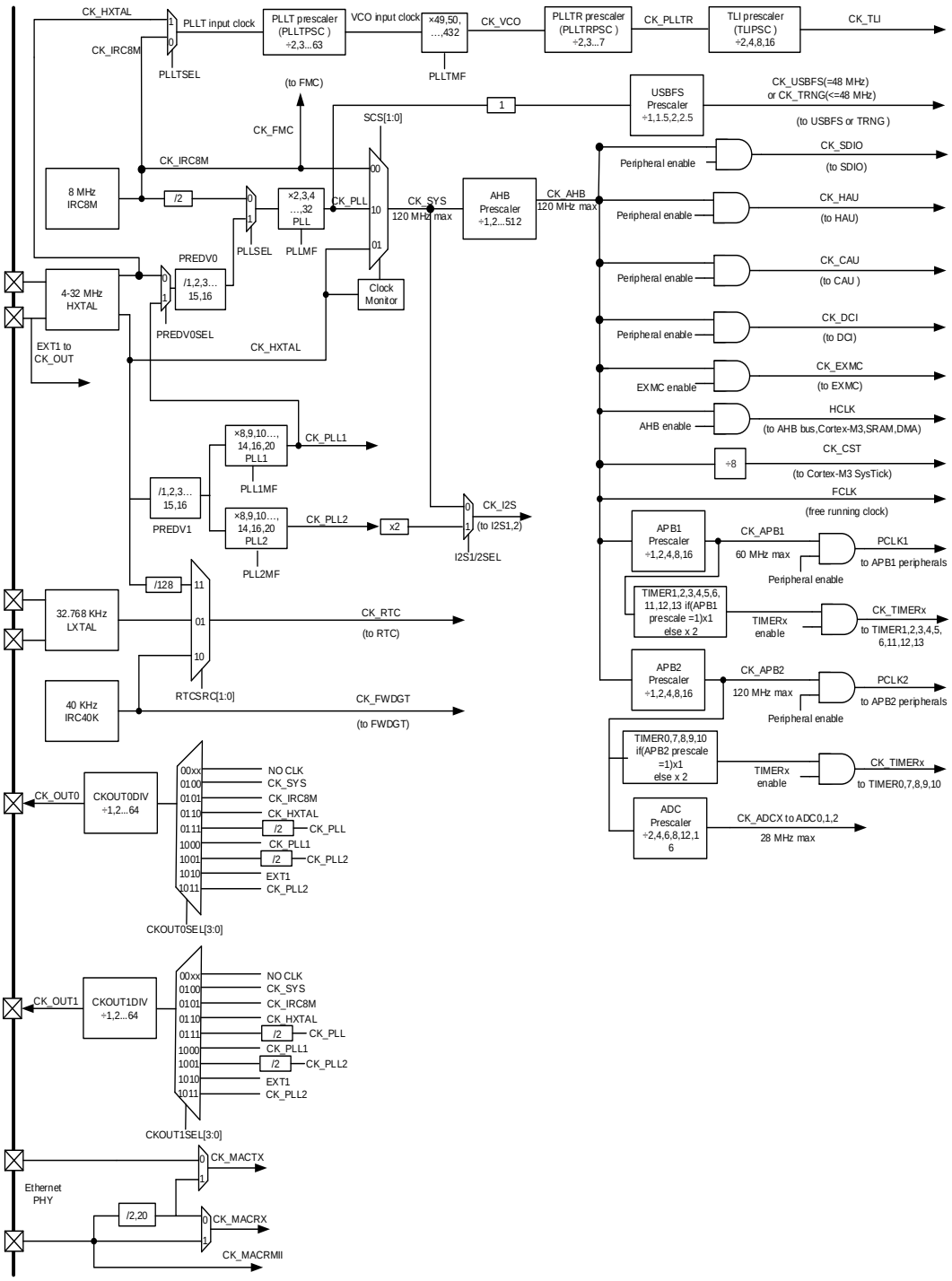
5.2. 时钟控制单元（CCTL）

5.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部8M RC振荡器时钟（IRC8M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速40K RC振荡器时钟（IRC40K）、一个外部低速晶体振荡器时钟（LXTAL）、三个锁相环（PLL、PLL1、PLL2）、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB和Cortex®-M3时钟都源自系统时钟（CK_SYS），系统时钟的时钟源可以选择IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到120MHz。

图5-2. 时钟树



互联型产品的RCU控制器有三个PLLs（PLL、PLL1和PLL2），能够提供丰富的时钟频率配置，以满足控制器的不同时钟频率需求。

预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2、APB1域的最高时钟频率分别为120MHz、120MHz、60MHz。RCU通过AHB时钟（HCLK）8分频后作为Cortex系统定时器（SysTick）的外部时钟。通过对SysTick控制和状态寄存器的设置，可选择上述时钟或AHB（HCLK）时钟作为SysTick时钟。

ADC时钟由APB2时钟经2、4、6、8、12、16分频获得，它们是通过设置RCU_CFG0的ADCPSC位来选择。

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

USBFS时钟来源于PLL，经过1、1.5、2、2.5分频后所得，通过设置配置寄存器0（RCU_CFG0）的USBFSPSC位来选择。USBFS时钟频率必须是48MHz。这些比特位同时也控制随机数生成器（TRNG）的时钟（≤48MHz）。TRNG的时钟也是由PLL提供，通过USBFSPSC位选择进行1、1.5、2、2.5分频而获得。

I2S的时钟由CK_SYS或PLL2*2提供，通过配置RCU_CFG1寄存器的I2SxSEL来选择。

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

以太网MAC的时钟来源于外部的PHY。如果启用以太网ENET模块，必须保证AHB的时钟频率至少在25MHz。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC40K时钟或HXTAL时钟的128分频提供。

当FWDGT启动时，FWDGT时钟被强制选择由IRC40K时钟做为时钟源。

5.2.2. 主要特征

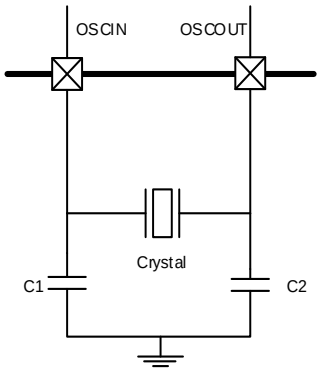
- 4到32MHz外部高速晶体振荡器（HXTAL）
- 内部8MHz RC振荡器（IRC8M）
- 32.768 KHz外部低速晶体振荡器（LXTAL）
- 内部40KHz RC振荡器（IRC40K）
- PLL时钟源可选HXTAL、IRC8M或PLL1
- HXTAL时钟监视器

5.2.3. 功能说明

外部高速晶体振荡时钟（HXTAL）

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

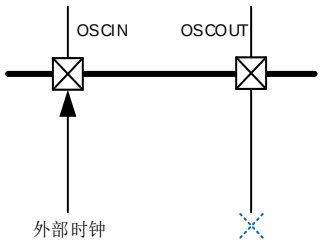
图5-3. HXTAL时钟源



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

将控制寄存器RCU_CTL的HXTALBPS和HXTAKEN位置‘1’来使能外部旁路模式。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如*图5-4. 旁路模式下HXTAL时钟源*所示。此时，CK_HXTAL等于驱动OSCIN管脚的外部时钟。

图5-4. 旁路模式下HXTAL时钟源



内部 8M RC 振荡器时钟（IRC8M）

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

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

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

锁相环（PLL）

CL系列的芯片，内部有三个锁相环，PLL，PLL1，PLL2。

内部锁相环PLL通过对输入参考频率为4~32MHz的时钟基准2~32倍频，可以提供16~120MHz的时钟输出。

PLL的输入时钟源有三个：IRC8M二分频，HXTAL以及由HXTAL产生的PLL1，选择其中的一个作为PLL时钟源。

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

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

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

当进入Deepsleep/Standby模式或者HXTAL监视器检测到时钟阻塞时（HXTAL做为锁相环的输入时钟），三个PLL将被关闭。

PLL1和PLL2的时钟通过HXTAL时钟的分频和倍频所得，具体可以通过设置配置寄存器1（RCU_CFGR1）的PLL2MF[3:0]、PLL1MF[3:0]和PREDV1[3:0]所得。

外部低速晶体振荡器时钟（LXTAL）

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

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

内部 40K RC 振荡器时钟（IRC40K）

IRC40K内部RC振荡器时钟担当一个低功耗时钟源的角色，不需要外部器件，它的时钟频率大约40KHz，为自由看门狗和实时时钟电路提供时钟。IRC40K RC振荡器可以通过设置复位源/时钟寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。复位源/时钟寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果复位源/时钟寄存器RCU_RSTSCK中的相应中断使能位IRC40KSTBIE被置‘1’，在IRC40K稳定以后，将产生一个中断。

TIMER4_CH3可以捕获IRC40K的时钟，进而对RTC和FWDGT的计数器进行校准，详细的信息

可以参考AFIO_PCF0寄存器的位TIMER4CH3_IREMAP。

系统时钟（CK_SYS）选择

系统复位后，IRC8M时钟默认做为CK_SYS的时钟源，改变配置寄存器0（RCU_CFG0）中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或CK_PLL。当SCS的值被改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置控制寄存器RCU_CTL中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，中断寄存器RCU_INT中的HXTAL时钟阻塞中断标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex-M3的不可屏蔽中断NMI相连。如果HXTAL被选作PLL的时钟源，HXTAL故障将促使PLL被自动禁止。

时钟输出功能

时钟输出功能输出从30KHz到60MHz的时钟。

CK_OUT0

通过设置时钟配置寄存器0（RCU_CFG0）中的CK_OUT0时钟源选择位域CKOUT0SEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O（AFIO）模式来输出选择的时钟信号。

表5-1. 时钟输出0的时钟源选择

时钟输出 0 的时钟源选择位域	时钟源
00xx	NO CLK
0100	CK_SYS
0101	CK_IRC8M
0110	CK_HXTAL
0111	CK_PLL/2
1000	CK_PLL1
1001	CK_PLL2/2
1010	EXT1
1011	CK_PLL2

通过配置时钟配置寄存器2（RCU_CFG2）的CKOUT0DIV[5:0]位，可以将输出时钟按比例分频，进而降低CKOUT0频率。

CK_OUT1

通过设置时钟配置寄存器2（RCU_CFG2）中的CK_OUT1时钟源选择位域CKOUT1SEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O（AFIO）模式来输出选择的时钟信号。

表5-2. 时钟输出1的时钟源选择

时钟输出 1 的时钟源选择位域	时钟源
00xx	NO CLK
0100	CK_SYS
0101	CK_IRC8M
0110	CK_HXTAL
0111	CK_PLL/2
1000	CK_PLL1
1001	CK_PLL2/2
1010	EXT1
1011	CK_PLL2

通过配置时钟配置寄存器2（RCU_CFG2）的CKOUT1DIV[5:0]位，可以将输出时钟按比例分频，进而降低CKOUT1频率

电压控制

深度睡眠模式电压寄存器（RCU_DSV）中的DSLPVS[2:0]位域可以控制1.2V域在深度睡眠模式下的电压。

表5-3. 深度睡眠模式下1.2V域电压选择

DSLPVS[2:0]	深度睡眠模式电压（V）
000	1.2
001	1.1
010	1.0
011	0.9

5.3. RCU 寄存器

RCU 基地址：0x4002 1000

5.3.1. 控制寄存器（RCU_CTL）

地址偏移：0x00

复位值：0x0000 xx83， x表示未定义

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PLL2STB	PLL2EN	PLL1STB	PLL1EN	PLLSTB	PLLEN	保留				CKMEN	HXTALBPS	HXTALSTB	HXTALEN
		r	rw	r	rw	r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC8MCALIB[7:0]								IRC8MADJ[4:0]					保留	IRC8MSTB	IRC8MEN
r								rw						r	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	PLL2STB	PLL2时钟稳定标志位 硬件置1来表示PLL2输出时钟是否稳定待用 0：PLL2未稳定 1：PLL2已稳定
28	PLL2EN	PLL2使能 软件置位或复位。当进入深度睡眠或待机模式时由硬件复位 0：PLL2被关闭 1：PLL2被打开
27	PLL1STB	PLL1时钟稳定标志位 硬件置1来表示PLL1输出时钟是否稳定待用 0：PLL1未稳定 1：PLL1已稳定
26	PLL1EN	PLL1使能 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位 0：PLL1被关闭 1：PLL1被打开
25	PLLSTB	PLL时钟稳定标志位 硬件置1来表示PLL输出时钟是否稳定待用 0：PLL未稳定 1：PLL已稳定
24	PLLEN	PLL使能

		软件置位或复位，当PLL时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位 0：PLL被关闭 1：PLL被打开
23:20	保留	必须保持复位值。
19	CKMEN	HXTAL时钟监视器使能 0：禁止高速4 ~ 32 MHz晶体振荡器（HXTAL）时钟监视器 1：使能高速4 ~ 32 MHz晶体振荡器（HXTAL）时钟监视器 当硬件检测到HXTAL时钟被阻塞在低或高状态时，内部硬件自动切换系统时钟到IRC8M时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清CKMIF位。 **注意：**使能HXTAL时钟监视器以后，硬件无视控制位IRC8MEN的状态，自动使能IRC8M时钟。
18	HXTALBPS	高速晶体振荡器（HXTAL）时钟旁路模式使能 只有在HXTALEN位为0时HXTALBPS位才可写 0：禁止HXTAL旁路模式 1：使能HXTAL旁路模式 HXTAL输出时钟等于输入时钟
17	HXTALSTB	高速晶体振荡器（HXTAL）时钟稳定标志位 硬件置‘1’来指示HXTAL振荡器时钟是否稳定待用 0：HXTAL振荡器未稳定 1：HXTAL振荡器已稳定
16	HXTALEN	高速晶体振荡器（HXTAL）使能 软件置位或复位，如果HXTAL时钟作为系统时钟或者当PLL时钟做为系统时钟时，其做为PLL的输入时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位 0：高速4 ~ 32 MHz晶体振荡器被关闭 1：高速4 ~ 32 MHz晶体振荡器被打开
15:8	IRC8MCALIB[7:0]	内部8MHz RC振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	内部8MHz RC振荡器时钟调整值 这些位由软件置位，最终调整值为IRC8MADJ[4:0]位域的当前值加上IRC8MCALIB[7:0]位域的值。最终调整值应该调整IRC8M到8MHz ± 1%
2	保留	必须保持复位值。
1	IRC8MSTB	IRC8M内部8MHz RC振荡器稳定标志位 硬件置‘1’来指示IRC8M振荡器时钟是否稳定待用 0：IRC8M振荡器未稳定 1：IRC8M振荡器已稳定
0	IRC8MEN	内部8MHz RC振荡器使能 软件置位或复位，如果IRC8M时钟做为系统时钟时，该位不能被复位。当从深度睡眠或待机模式返回，或当CKMEN置位同时用作系统时钟的HXTAL振荡器发生故障时，

该位由硬件置1来启动IRC8M振荡器。

0：内部8MHz RC振荡器被关闭

1：内部8MHz RC振荡器被打开

5.3.2. 配置寄存器 0（RCU_CFG0）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PLLMF[4]	ADCPSC[2]	CKOUT0SEL[3:0]				USBFSPSC[1:0]		PLLMF[3:0]				PREDV0_LSB	PLLSEL
		rw	rw	rw				rw		rw				rw	rw

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	PLLMF[4]	PLLMF的第4位 参考寄存器RCU_CFG0的18到21位
28	ADCPSC[2]	ADCPSC的第2位 参考寄存器RCU_CFG0的14到15位
27:24	CKOUT0SEL[3:0]	CKOUT0时钟源选择 由软件置位或清零 00xx：无时钟输出 0100：选择系统时钟CK_SYS 0101：选择内部8M RC振荡器时钟 0110：选择高速晶体振荡器时钟 （HXTAL） 0111：选择（CK_PLL / 2）时钟 1000：选择CK_PLL1时钟 1001：选择（CK_PLL2 / 2）时钟 1010：选择提供给ENET的EXT1时钟 1011：选择CK_PLL2时钟
23:22	USBFSPSC[1:0]	USBFS和TRNG的时钟分频系数 由软件置位或清零。USBFS的时钟必须为48MHz，当USBFS时钟使能的时候，这些位无法修改。这些位也控制TRNG的时钟（≤48 MHz）。 00：CK_PLL / 1.5 01：CK_PLL / 1 10：CK_PLL / 2.5

		11：CK_PLL / 2
21:18	PLLMF[3:0]	PLL时钟倍频因子 与寄存器RCU_CFG0的29位共同构成倍频因子，由软件置位或清零 **注意**：PLL输出时钟频率不能超过120MHz 00000：（PLL源时钟 x 2） 00001：（PLL源时钟 x 3） 00010：（PLL源时钟 x 4） 00011：（PLL源时钟 x 5） 00100：（PLL源时钟 x 6） 00101：（PLL源时钟 x 7） 00110：（PLL源时钟 x 8） 00111：（PLL源时钟 x 9） 01000：（PLL源时钟 x 10） 01001：（PLL源时钟 x 11） 01010：（PLL源时钟 x 12） 01011：（PLL源时钟 x 13） 01100：（PLL源时钟 x 14） 01101：（PLL源时钟 x 6.5） 01110：（PLL源时钟 x 16） 01111：（PLL源时钟 x 16） 10000：（PLL源时钟 x 17） 10001：（PLL源时钟 x 18） 10010：（PLL源时钟 x 19） 10011：（PLL源时钟 x 20） 10100：（PLL源时钟 x 21） 10101：（PLL源时钟 x 22） 10110：（PLL源时钟 x 23） 10111：（PLL源时钟 x 24） 11000：（PLL源时钟 x 25） 11001：（PLL源时钟 x 26） 11010：（PLL源时钟 x 27） 11011：（PLL源时钟 x 28） 11100：（PLL源时钟 x 29） 11101：（PLL源时钟 x 30） 11110：（PLL源时钟 x 31） 11111：（PLL源时钟 x 32）
17	PREDV0_LSB	PREDV0分频因子的最低位 与寄存器RCU_CFG1位PREDV0第0位相同，通过寄存器RCU_CFG1来改变PREDV0的值，此位也会一同改。当PREDV0的第1到3位未修改时，此位决定PREDV0的输入时钟是否二分频。
16	PLLSEL	PLL时钟源选择 由软件置位或复位，控制PLL时钟源

		0: （IRC8M / 2）被选择为PLL时钟的时钟源 1: PREDV0输出时钟被选择为PLL时钟的时钟源
15:14	ADCPSC[1:0]	ADC的时钟分频系数 与寄存器RCU_CFG0的28位共同构成分频因子，由软件置位或清零。 000：CK_ADC = CK_APB2 / 2 001：CK_ADC = CK_APB2 / 4 010：CK_ADC = CK_APB2 / 6 011：CK_ADC = CK_APB2 / 8 100：CK_ADC = CK_APB2 / 2 101：CK_ADC = CK_APB2 / 12 110：CK_ADC = CK_APB2 / 8 111：CK_ADC = CK_APB2 / 16
13:11	APB2PSC[2:0]	APB2预分频选择 由软件置位或清零，控制APB2时钟分频因子 0xx：CK_APB2 = CK_AHB 100：CK_APB2 = CK_AHB / 2 101：CK_APB2 = CK_AHB / 4 110：CK_APB2 = CK_AHB / 8 111：CK_APB2 = CK_AHB / 16
10:8	APB1PSC[2:0]	APB1预分频选择 由软件置位或清零，控制APB1时钟分频因子。 **注意**：CK_APB1输出频率不超过60MHz。 0xx：CK_APB1 = CK_AHB 100：CK_APB1 = CK_AHB / 2 101：CK_APB1 = CK_AHB / 4 110：CK_APB1 = CK_AHB / 8 111：CK_APB1 = CK_AHB / 16
7:4	AHBPSC[3:0]	AHB预分频选择 由软件置位或清零，控制AHB时钟分频因子. 0xxx：CK_AHB = CK_SYS 1000：CK_AHB = CK_SYS / 2 1001：CK_AHB = CK_SYS / 4 1010：CK_AHB = CK_SYS / 8 1011：CK_AHB = CK_SYS / 16 1100：CK_AHB = CK_SYS / 64 1101：CK_AHB = CK_SYS / 128 1110：CK_AHB = CK_SYS / 256 1111：CK_AHB = CK_SYS / 512
3:2	SCSS[1:0]	系统时钟选择状态 由硬件置位或清零，标识当前系统时钟的时钟源 00：选择CK_IRC8M时钟作为CK_SYS时钟源

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

5.3.3. 时钟中断寄存器（RCU_INT）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CKMIC	PLL2 STBIC	PLL1 STBIC	PLL STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
								w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL2 STBIE	PLL1 STBIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	PLL2 STBIF	PLL1 STBIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL时钟阻塞中断清零 软件写1复位CKMIF标志位. 0：不复位CKMIF标志位 1：复位CKMIF标志位
22	PLL2STBIC	PLL2时钟稳定中断清零 软件写1复位PLL2STBIF标志位 0：不复位PLL2STBIF标志位 1：复位PLL2STBIF标志位
21	PLL1STBIC	PLL1时钟稳定中断清零 软件写1复位PLL1STBIF标志位 0：不复位PLL1STBIF标志位

		1：复位PLL1STBIF标志位
20	PLLSTBIC	PLL时钟稳定中断清零 软件写1复位PLLSTBIF标志位 0：不复位PLLSTBIF标志位 1：复位PLLSTBIF标志位
19	HXTALSTBIC	HXTAL时钟稳定中断清零 软件写1复位HXTALSTBIF标志位 0：不复位HXTALSTBIF标志位 1：复位HXTALSTBIF标志位
18	IRC8MSTBIC	IRC8M时钟稳定中断清零 软件写1复位IRC8MSTBIF标志位 0：不复位IRC8MSTBIF标志位 1：复位IRC8MSTBIF标志位
17	LXTALSTBIC	LXTAL时钟稳定中断清零 软件写1复位LXTALSTBIF标志位 0：不复位LXTALSTBIF标志位 1：复位LXTALSTBIF标志位
16	IRC40KSTBIC	IRC40K时钟稳定中断清零 软件写1复位IRC40KSTBIF标志位 0：不复位IRC40KSTBIF标志位 1：复位IRC40KSTBIF标志位
15	保留	必须保持复位值。
14	PLL2STBIE	PLL2时钟稳定中断使能 软件置位和复位来使能/禁止PLL2时钟稳定中断 0：禁止PLL2时钟稳定中断 1：使能PLL2时钟稳定中断
13	PLL1STBIE	PLL1时钟稳定中断使能 软件置位和复位来使能/禁止PLL1时钟稳定中断 0：禁止PLL1时钟稳定中断 1：使能PLL1时钟稳定中断
12	PLLSTBIE	PLL时钟稳定中断使能 软件置位和复位来使能/禁止PLL时钟稳定中断 0：禁止PLL时钟稳定中断 1：使能PLL时钟稳定中断
11	HXTALSTBIE	HXTAL时钟稳定中断使能 软件置位和复位来使能/禁止HXTAL时钟稳定中断 0：禁止HXTAL时钟稳定中断 1：使能HXTAL时钟稳定中断

10	IRC8MSTBIE	IRC8M时钟稳定中断使能 软件置位和复位来使能/禁止IRC8M时钟稳定中断 0：禁止IRC8M时钟稳定中断 1：使能IRC8M时钟稳定中断
9	LXTALSTBIE	LXTAL时钟稳定中断使能 软件置位和复位来使能/禁止LXTAL时钟稳定中断 0：禁止LXTAL时钟稳定中断 1：使能LXTAL时钟稳定中断
8	IRC40KSTBIE	IRC40K时钟稳定中断使能 软件置位和复位来使能/禁止IRC40K时钟稳定中断 0：禁止IRC40K时钟稳定中断 1：使能IRC40K时钟稳定中断
7	CKMIF	HXTAL时钟阻塞中断标志位 当HXTAL时钟被阻塞时由硬件置位. 软件置位CKMIC位时清除该位 0：时钟正常运行 1：HXTAL时钟阻塞
6	PLL2STBIF	PLL2时钟稳定中断标志位 当PLL时钟稳定且PLL2STBIE位被置1时由硬件置1 软件置位PLL2STBIC位时清除该位 0：无PLL2时钟稳定中断产生 1：产生PLL2时钟稳定中断
5	PLL1STBIF	PLL1时钟稳定中断标志位 当PLL时钟稳定且PLL1STBIE位被置1时由硬件置1 软件置位PLL1STBIC位时清除该位 0：无PLL1时钟稳定中断产生 1：产生PLL1时钟稳定中断
4	PLLSTBIF	PLL时钟稳定中断标志位 当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1 软件置位PLLSTBIC位时清除该位 0：无PLL时钟稳定中断产生 1：产生PLL时钟稳定中断
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当高速4~32 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1 软件置位HXTALSTBIC位时清除该位 0：无HXTAL时钟稳定中断产生 1：产生HXTAL时钟稳定中断
2	IRC8MSTBIF	IRC8M时钟稳定中断标志位 当内部8MHz RC振荡器时钟稳定且IRC8MSTBIE位被置1时由硬件置1 软件置位IRC8MSTBIC位时清除该位

		0：无IRC8M时钟稳定中断产生 1：产生IRC8M时钟稳定中断
1	LXTALSTBIF	LXTAL时钟稳定中断标志位 当低速晶体振荡器时钟稳定且LXTALSTBIE位被置1时由硬件置1 软件置位LXTALSTBIC位时清除该位 0：无LXTAL时钟稳定中断产生 1：产生LXTAL时钟稳定中断
0	IRC40KSTBIF	IRC40K时钟稳定中断标志位 当内部40kHz RC振荡器时钟稳定且IRC40KSTBIE位被置1时由硬件置1 软件置位IRC40KSTBIC位时清除该位 0：无IRC40K时钟稳定中断产生 1：产生IRC40K时钟稳定中断

5.3.4. APB2 复位寄存器（RCU_APB2RST）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										TIMER10 RST	TIMER9 RST	TIMER8 RST	保留		
										rw	rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2 RST	USART0 RST	TIMER7 RST	SPI0RST	TIMER0 RST	ADC1 RST	ADC0 RST	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	TIMER10RST	TIMER10复位 由软件置位或复位 0：无作用 1：复位TIMER10
20	TIMER9RST	TIMER9复位 由软件置位或复位 0：无作用 1：复位TIMER9
19	TIMER8RST	TIMER8复位 由软件置位或复位 0：无作用

		1：复位TIMER8
18:16	保留	必须保持复位值。
15	ADC2RST	ADC2复位 由软件置位或复位 0：无作用 1：复位所有ADC2
14	USART0RST	USART0复位 由软件置位或复位 0：无作用 1：复位USART0
13	TIMER7RST	TIMER7复位 由软件置位或复位 0：无作用 1：复位TIMER7
12	SPI0RST	SPI0复位 由软件置位或复位 0：无作用 1：复位SPI0
11	TIMER0RST	TIMER0复位 由软件置位或复位 0：无作用 1：复位TIMER0
10	ADC1RST	ADC1复位 由软件置位或复位 0：无作用 1：复位所有ADC1
9	ADC0RST	ADC0复位 由软件置位或复位 0：无作用 1：复位所有ADC0
8	PGRST	GPIO端口G复位 由软件置位或复位 0：无作用 1：复位GPIO端口G
7	PFRST	GPIO端口F复位 由软件置位或复位 0：无作用 1：复位GPIO端口F

6	PERST	GPIO端口E复位 由软件置位或复位 0：无作用 1：复位GPIO端口E
5	PDRST	GPIO端口D复位 由软件置位或复位 0：无作用 1：复位GPIO端口D
4	PCRST	GPIO端口C复位 由软件置位或复位 0：无作用 1：复位GPIO端口C
3	PBRST	GPIO端口B复位 由软件置位或复位 0：无作用 1：复位GPIO端口B
2	PARST	GPIO端口A复位 由软件置位或复位 0：无作用 1：复位GPIO端口A
1	保留	必须保持复位值。
0	AFRST	复用功能I/O复位 由软件置位或复位 0：无作用 1：复位复用功能I/O

5.3.5. APB1 复位寄存器（RCU_APB1RST）

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DACRST	PMURST	BKPIRST	CAN1 RST	CAN0 RST	保留		I2C1RST	I2C0RST	UART4 RST	UART3 RST	USART2 RST	USART1 RST	保留
		rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	保留		WWDGT RST	保留		TIMER13 RST	TIMER12 RST	TIMER11 RST	TIMER6 RST	TIMER5 RST	TIMER4 RST	TIMER3 RST	TIMER2 RST	TIMER1 RST
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DACRST	DAC复位 由软件置位或复位 0：无作用 1：复位DAC
28	PMURST	PMU复位 由软件置位或复位 0：无作用 1：复位PMU
27	BKPIRST	BKP接口复位 由软件置位或复位 0：无作用 1：复位BKP
26	CAN1RST	CAN1复位 由软件置位或复位 0：无作用 1：复位CAN1
25	CAN0RST	CAN0复位 由软件置位或复位 0：无作用 1：复位CAN0
24:23	保留	必须保持复位值。
22	I2C1RST	I2C1复位 由软件置位或复位 0：无作用 1：复位I2C1
21	I2C0RST	I2C0复位 由软件置位或复位 0：无作用 1：复位I2C0
20	UART4RST	UART4复位 由软件置位或复位 0：无作用 1：复位UART4
19	UART3RST	UART3复位 由软件置位或复位

		0：无作用 1：复位UART3
18	USART2RST	USART2复位 由软件置位或复位 0：无作用 1：复位USART2
17	USART1RST	USART1复位 由软件置位或复位 0：无作用 1：复位USART1
16	保留	必须保持复位值。
15	SPI2RST	SPI2复位 由软件置位或复位 0：无作用 1：复位SPI2
14	SPI1RST	SPI1复位 由软件置位或复位 0：无作用 1：复位SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	WWDGT复位 由软件置位或复位 0：无作用 1：复位WWDGT
10:9	保留	必须保持复位值。
8	TIMER13RST	TIMER13复位 由软件置位或复位 0：无作用 1：复位TIMER13
7	TIMER12RST	TIMER12复位 由软件置位或复位 0：无作用 1：复位TIMER12
6	TIMER11RST	TIMER11复位 由软件置位或复位 0：无作用 1：复位TIMER11
5	TIMER6RST	TIMER6复位

		由软件置位或复位 0：无作用 1：复位TIMER6
4	TIMER5RST	TIMER5复位 由软件置位或复位 0：无作用 1：复位TIMER5
3	TIMER4RST	TIMER4复位 由软件置位或复位 0：无作用 1：复位TIMER4
2	TIMER3RST	TIMER3复位 由软件置位或复位 0：无作用 1：复位TIMER3
1	TIMER2RST	TIMER2复位 由软件置位或复位 0：无作用 1：复位TIMER2
0	TIMER1RST	TIMER1复位 由软件置位或复位 0：无作用 1：复位TIMER1

5.3.6. AHB1 使能寄存器（RCU_AHB1EN）

地址偏移：0x14

复位值：0x0000 0014

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENETTXEN	ENETEN	保留	USBFSEN	保留	SDIOEN	保留	EXMCEN	保留	CRCEN	保留	FMCSPEN	保留	SRAMSPEN	DMA1EN	DMA0EN
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	ENETRXEN	以太网RX时钟使能 由软件置位或复位 0：关闭以太网RX时钟 1：开启以太网RX时钟
15	ENETTXEN	以太网TX时钟使能 由软件置位或复位 0：关闭以太网TX时钟 1：开启以太网TX时钟
14	ENETEN	以太网时钟使能 由软件置位或复位 0：关闭以太网时钟 1：开启以太网时钟
13	保留	必须保持复位值。
12	USBFSEN	USBFS时钟使能 由软件置位或复位 0：关闭USBFS时钟 1：开启USBFS时钟
11	保留	必须保持复位值。
10	SDIOEN	SDIO时钟使能 由软件置位或复位 0：关闭SDIO时钟 1：开启SDIO时钟
9	保留	必须保持复位值。
8	EXMCEN	EXMC时钟使能 由软件置位或复位 0：关闭EXMC时钟 1：开启EXMC时钟
7	保留	必须保持复位值。
6	CRCEN	CRC时钟使能 由软件置位或复位 0：关闭CRC时钟 1：开启CRC时钟
5	保留	必须保持复位值。
4	FMCSPEN	在睡眠模式下FMC时钟使能 由软件置位或复位 0：在睡眠模式下关闭FMC时钟

		1：在睡眠模式下开启FMC时钟
3	保留	必须保持复位值。
2	SRAMSPEN	在睡眠模式下SRAM时钟使能 由软件置位或复位 0：在睡眠模式下关闭SRAM时钟 1：在睡眠模式下开启SRAM时钟
1	DMA1EN	DMA1时钟使能 由软件置位或复位 0：关闭DMA1时钟 1：开启DMA1时钟
0	DMA0EN	DMA0时钟使能 由软件置位或复位 0：关闭DMA0时钟 1：开启DMA0时钟

5.3.7. APB2 使能寄存器（RCU_APB2EN）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										TIMER10EN	TIMER9EN	TIMER8EN	保留		
										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	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	保留	AFEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	TIMER10EN	TIMER10时钟使能 由软件置位或复位 0：关闭TIMER10时钟 1：开启TIMER10时钟
20	TIMER9EN	TIMER9时钟使能 由软件置位或复位 0：关闭TIMER9时钟 1：开启TIMER9时钟

19	TIMER8EN	TIMER8时钟使能 由软件置位或复位 0：关闭TIMER8时钟 1：开启TIMER8时钟
18:16	保留	必须保持复位值。
15	ADC2EN	ADC2时钟使能 由软件置位或复位 0：关闭ADC2时钟 1：开启ADC2时钟
14	USART0EN	USART0时钟使能 由软件置位或复位 0：关闭USART0时钟 1：开启USART0时钟
13	TIMER7EN	TIMER7复位 由软件置位或复位 0：无作用 1：复位TIMER7
12	SPI0EN	SPI0复位 由软件置位或复位 0：无作用 1：复位SPI0
11	TIMER0EN	TIMER0复位 由软件置位或复位 0：无作用 1：复位TIMER0
10	ADC1EN	ADC1时钟使能 由软件置位或复位 0：关闭ADC1时钟 1：开启ADC1时钟
9	ADC0EN	ADC0时钟使能 由软件置位或复位 0：关闭ADC0时钟 1：开启ADC0时钟
8	PGEN	GPIO端口G时钟使能 由软件置位或复位 0：关闭GPIO端口G时钟 1：开启GPIO端口G时钟
7	PFEN	GPIO端口F时钟使能 由软件置位或复位

		0：关闭GPIO端口F时钟 1：开启GPIO端口F时钟
6	PEEN	GPIO端口E时钟使能 由软件置位或复位 0：关闭GPIO端口E时钟 1：开启GPIO端口E时钟
5	PDEN	GPIO端口D时钟使能 由软件置位或复位 0：关闭GPIO端口D时钟 1：开启GPIO端口D时钟
4	PCEN	GPIO端口C时钟使能 由软件置位或复位 0：关闭GPIO端口C时钟 1：开启GPIO端口C时钟
3	PBEN	GPIO端口B时钟使能 由软件置位或复位 0：关闭GPIO端口B时钟 1：开启GPIO端口B时钟
2	PAEN	GPIO端口A时钟使能 由软件置位或复位 0：关闭GPIO端口A时钟 1：开启GPIO端口A时钟
1	保留	必须保持复位值。
0	AFEN	复用功能IO时钟使能 由软件置位或复位 0：关闭复用功能IO时钟 1：开启复用功能IO时钟

5.3.8. APB1 使能寄存器（RCU_APB1EN）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		DACEN	PMUEN	BKPIEN	CAN1EN	CAN0EN	保留		I2C1EN	I2C0EN	UART4 EN	UART3 EN	USART2 EN	USART1 EN	保留
		rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2EN	SPI1EN	保留		WWDGT	保留		TIMER13	TIMER12	TIMER11	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1

			EN		EN	EN	EN	EN	EN	EN	EN	EN	EN
rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DACEN	DAC时钟使能 由软件置位或复位 0：关闭DAC时钟 1：开启DAC时钟
28	PMUEN	PMU时钟使能 由软件置位或复位 0：关闭PMU时钟 1：开启PMU时钟
27	BKPIEN	BKP接口时钟使能 由软件置位或复位 0：关闭BKP接口时钟 1：开启BKP接口时钟
26	CAN1EN	CAN1时钟使能 由软件置位或复位 0：关闭CAN1时钟 1：开启CAN1时钟
25	CAN0EN	CAN0时钟使能 由软件置位或复位 0：关闭CAN0时钟 1：开启CAN0时钟
24:23	保留	必须保持复位值。
22	I2C1EN	I2C1时钟使能 由软件置位或复位 0：关闭I2C1时钟 1：开启I2C1时钟
21	I2C0EN	I2C0时钟使能 由软件置位或复位 0：关闭I2C0时钟 1：开启I2C0时钟
20	UART4EN	UART4时钟使能 由软件置位或复位 0：关闭UART4时钟 1：开启UART4时钟

19	UART3EN	UART3时钟使能 由软件置位或复位 0：关闭UART3时钟 1：开启UART3时钟
18	USART2EN	USART2时钟使能 由软件置位或复位 0：关闭USART2时钟 1：开启USART2时钟
17	USART1EN	USART1时钟使能 由软件置位或复位 0：关闭USART1时钟 1：开启USART1时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2时钟使能 由软件置位或复位 0：关闭SPI2时钟 1：开启SPI2时钟
14	SPI1EN	SPI1时钟使能 由软件置位或复位 0：关闭SPI1时钟 1：开启SPI1时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	WWDGT时钟使能 由软件置位或复位 0：关闭WWDGT时钟 1：开启WWDGT时钟
10:9	保留	必须保持复位值。
8	TIMER13EN	TIMER13时钟使能 由软件置位或复位 0：关闭TIMER13时钟 1：开启TIMER13时钟
7	TIMER12EN	TIMER12时钟使能 由软件置位或复位 0：关闭TIMER12时钟 1：开启TIMER12时钟
6	TIMER11EN	TIMER11时钟使能 由软件置位或复位 0：关闭TIMER11时钟

		1：开启TIMER11时钟
5	TIMER6EN	TIMER6时钟使能 由软件置位或复位 0：关闭TIMER6时钟 1：开启TIMER6时钟
4	TIMER5EN	TIMER5时钟使能 由软件置位或复位 0：关闭TIMER5时钟 1：开启TIMER5时钟
3	TIMER4EN	TIMER4时钟使能 由软件置位或复位 0：关闭TIMER4时钟 1：开启TIMER4时钟
2	TIMER3EN	TIMER3时钟使能 由软件置位或复位 0：关闭TIMER3时钟 1：开启TIMER3时钟
1	TIMER2EN	TIMER2时钟使能 由软件置位或复位 0：关闭TIMER2时钟 1：开启TIMER2时钟
0	TIMER1EN	TIMER1时钟使能 由软件置位或复位 0：关闭TIMER1时钟 1：开启TIMER1时钟

5.3.9. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x20
复位值：0x0000 0018，只能由备份域复位进行复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

					BPS	STB	
rw		rw		rw	rw	r	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置位或复位 0：无作用 1：复位备份域
15	RTCEN	RTC时钟使能 由软件置位或复位 0：关闭RTC时钟 1：开启RTC时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC时钟源选择 由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00：没有时钟 01：选择CK_LXTAL时钟作为RTC的时钟源 10：选择CK_IRC40K时钟作为RTC的时钟源 11：选择CK_HXTAL / 128时钟作为RTC的时钟源
7:5	保留	必须保持复位值。
4:3	LXTALDRI[1:0]	LXTAL驱动能力 由软件置位或复位。当备份域复位时将复位该值 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力 （复位后的缺省值） **注意：**LXTALDRI位在旁路模式下无效
2	LXTALBPS	LXTAL旁路模式使能 由软件置位或复位 0：禁止LXTAL旁路模式 1：使能LXTAL旁路模式
1	LXTALSTB	低速晶体振荡器稳定标志位 硬件置‘1’来指示LXTAL振荡器时钟是否稳定待用 0：LXTAL未稳定 1：LXTAL已稳定
0	LXTALEN	LXTAL时钟使能 由软件置位或复位

0：关闭LXTAL时钟

1：使能LXTAL时钟

5.3.10. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x24

复位值：0x0C00 0000，所有复位标志位仅在电源复位时被清零，RSTFC/IRC40KEN在系统复位时被清零。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC40K STB	IRC40K EN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置位 向RSTFC位写1来清除该位 0：无低功耗管理复位发生 1：发生低功耗管理复位
30	WWDGTRSTF	窗口看门狗定时器复位标志位 窗口看门狗定时器复位发生时由硬件置1 向RSTFC位写1来清除该位 0：无窗口看门狗复位发生 1：发生窗口看门狗复位
29	FWDGTRSTF	自由看门狗定时器复位标志位 自由看门狗复位发生时由硬件置1 向RSTFC位写1来清除该位 0：无自由看门狗定时器复位发生 1：发生自由看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置1 向RSTFC位写1来清除该位 0：无软件复位发生 1：发生软件复位
27	PORRSTF	电源复位标志位

		电源复位发生时由硬件置1 向RSTFC位写1来清除该位 0：无电源复位发生 1：发生电源复位
26	EPRSTF	外部引脚复位标志位 外部引脚复位发生时由硬件置1 向RSTFC位写1来清除该位 0：无外部引脚复位发生 1：发生外部引脚复位
25	保留	必须保持复位值。
24	RSTFC	清除复位标志位 由软件置1来清除所有复位标志位 0：无作用 1：清除所有复位标志位
23:2	保留	必须保持复位值。
1	IRC40KSTB	IRC40K时钟稳定标志位 该位由硬件置1指示IRC40K输出时钟是否稳定待用 0：IRC40K时钟未稳定 1：IRC40K已稳定
0	IRC40KEN	IRC40K使能 由软件置位和复位 0：关闭IRC40K时钟 1：开启IRC40K时钟

5.3.11. AHB1 复位寄存器（RCU_AHB1RST）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:15	保留	必须保持复位值。

14	ENETRST	ENET复位 由软件置位或复位 0：无作用 1：复位ENET
13	保留	必须保持复位值。
12	USBFSRST	USBFS复位 由软件置位或复位 0：无作用 1：复位USBFS
11:0	保留	必须保持复位值。

5.3.12. 配置寄存器 1（RCU_CFG1）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													I2S2SEL	I2S1SEL	PREDV0SEL
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL2MF[3:0]				PLL1MF[3:0]				PREDV1[3:0]				PREDV0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	I2S2SEL	I2S2时钟源选择 由软件置位或复位，控制I2S2时钟源 0：系统时钟被选择为I2S2时钟的时钟源 1：（CK_PLL2 x 2） 被选择为I2S2时钟的时钟源
17	I2S1SEL	I2S1时钟源选择 由软件置位或复位，控制I2S1时钟源 0：系统时钟被选择为I2S1时钟的时钟源 1：（CK_PLL2 x 2） 被选择为I2S1时钟的时钟源
16	PREDV0SEL	PREDV0时钟源选择 由软件置位或复位 0：HXTAL被选择为PREDV0的时钟源 1：CK_PLL1被选择为PREDV0的时钟源
15:12	PLL2MF[3:0]	PLL2时钟倍频因子

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

		1101：PREDV2输入源时钟14分频 1110：PREDV2输入源时钟15分频 1111：PREDV2输入源时钟16分频
3:0	PREDV0[3:0]	PREDV0分频因子 由软件置位或清零，PLL未使能时，可以修改这些位 **注意**：PREDV0的第0位与RCU_CFG0寄存器的17位相同，修改RCU_CFG0寄存器的17位，PREDV0的第0位也会进行相同的修改 0000：PREDV0输入源时钟未分频 0001：PREDV0输入源时钟2分频 0010：PREDV0输入源时钟3分频 0011：PREDV0输入源时钟4分频 0100：PREDV0输入源时钟5分频 0101：PREDV0输入源时钟6分频 0110：PREDV0输入源时钟7分频 0111：PREDV0输入源时钟8分频 1000：PREDV0输入源时钟9分频 1001：PREDV0输入源时钟10分频 1010：PREDV0输入源时钟11分频 1011：PREDV0输入源时钟12分频 1100：PREDV0输入源时钟13分频 1101：PREDV0输入源时钟14分频 1110：PREDV0输入源时钟15分频 1111：PREDV0输入源时钟16分频

5.3.13. 深度睡眠模式电压寄存器（RCU_DSV）

地址偏移：0x34
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	DSLPVS[2:0]	深度睡眠模式电压选择 由软件置位和清零这些位 000：在深度睡眠模式下内核电压为1.2V

001：在深度睡眠模式下内核电压为1.1V

010：在深度睡眠模式下内核电压为1.0V

011：在深度睡眠模式下内核电压为0.9V

1xx：保留

5.3.14. AHB2 使能寄存器（RCU_AHB2EN）

地址偏移：0x60

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRNGEN	HAUEN	CAUEN	保留			DCIEN
									rw	rw	rw				rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TRNGEN	TRNG时钟使能 由软件置位或复位 0：关闭TRNG时钟 1：开启TRNG时钟
5	HAUEN	HAU时钟使能 由软件置位或复位 0：关闭HAU时钟 1：开启HAU时钟
4	CAUEN	CAU时钟使能 由软件置位或复位 0：关闭CAU时钟 1：开启CAU时钟
3:1	保留	必须保持复位值。
0	DCIEN	DCI时钟使能 由软件置位或复位 0：关闭DCI时钟 1：开启DCI时钟

5.3.15. APB2 附加使能寄存器（RCU_ADDAPB2EN）

地址偏移：0x64

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
PIEN	PHEN	保留			TLIEN	保留	USART5EN	保留							
rw	rw				rw		rw								

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

位/位域	名称	描述
31	PIEN	GPIO端口I时钟使能 由软件置位或复位 0：关闭GPIO端口I时钟 1：开启GPIO端口I时钟
30	PHEN	GPIO端口H时钟使能 由软件置位或复位 0：关闭GPIO端口H时钟 1：开启GPIO端口H时钟
29:27	保留	必须保持复位值。
26	TLIEN	TLI时钟使能 由软件置位或复位 0：关闭TLI时钟 1：开启TLI时钟
25	保留	必须保持复位值。
24	USART5EN	USART5时钟使能 由软件置位或复位 0：关闭USART5时钟 1：开启USART5时钟
23:0	保留	必须保持复位值。

5.3.16. APB1 附加使能寄存器（RCU_ADDAPB1EN）

地址偏移：0x68

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART7EN	UART6EN	保留						I2C2EN	保留						
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	UART7EN	UART7时钟使能 由软件置位或复位 0：关闭UART7时钟 1：开启UART7时钟
30	UART6EN	UART6时钟使能 由软件置位或复位 0：关闭UART6时钟 1：开启UART6时钟
29:24	保留	必须保持复位值。
23	I2C2EN	I2C2时钟使能 由软件置位或复位 0：关闭I2C2时钟 1：开启I2C2时钟
22:0	保留	必须保持复位值。

5.3.17. AHB2 复位寄存器（RCU_AHB2RST）

地址偏移：0x70

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRNGRST	HAURST	CAURST	保留			DCIRST
									rw	rw	rw				rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TRNGRST	TRNG复位

		由软件置位或复位 0：无作用 1：复位TRNG
5	HAURST	HAU复位 由软件置位或复位 0：无作用 1：复位HAU
4	CAURST	CAU复位 由软件置位或复位 0：无作用 1：复位CAU
3:1	保留	必须保持复位值。
0	DCIRST	DCI复位 由软件置位或复位 0：无作用 1：复位DCI

5.3.18. APB2 附加复位寄存器（RCU_ADDAPB2RST）

地址偏移：0x74

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
PIRST	PHRST	保留			TLIRST	保留	USART5RST	保留							
rw	rw				rw		rw								

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

位/位域	名称	描述
31	PIRST	GPIO端口I复位 由软件置位或复位 0：无作用 1：复位GPIO端口I
30	PHRST	GPIO端口H复位 由软件置位或复位 0：无作用 1：复位GPIO端口H

29:27	保留	必须保持复位值。
26	TLIRST	TLI复位 由软件置位或复位 0：无作用 1：复位TLI
25	保留	必须保持复位值。
24	USART5RST	USART5复位 由软件置位或复位 0：无作用 1：复位USART5
23:0	保留	必须保持复位值。

5.3.19. APB1 附加复位寄存器（RCU_ADDAPB1RST）

地址偏移：0x78

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART7RST	UART6RST	保留						I2C2RST	保留						
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	UART7RST	UART7复位 由软件置位或复位 0：无作用 1：复位UART7
30	UART6RST	UART6复位 由软件置位或复位 0：无作用 1：复位UART6
29:24	保留	必须保持复位值。
23	I2C2RST	I2C2复位 由软件置位或复位 0：无作用

		1：复位I2C2
22:0	保留	必须保持复位值。

5.3.20. 配置寄存器 2（RCU_CFG2）

地址偏移：0x80

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												CKOUT1SEL[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CKOUT1DIV[5:0]					保留			CKOUT0DIV[5:0]					
		rw								rw					

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	CKOUT1SEL[3:0]	CKOUT1时钟源选择 由软件置位或清零 00xx：无时钟输出 0100：选择系统时钟CK_SYS 0101：选择内部8M RC振荡器时钟 0110：选择高速晶体振荡器时钟（HXTAL） 0111：选择（CK_PLL / 2）时钟 1000：选择CK_PLL1时钟 1001：选择（CK_PLL2 / 2）时钟 1010：选择EXT1时钟 1011：选择CK_PLL2时钟
15:14	保留	必须保持复位值。
13:8	CKOUT1DIV[5:0]	CK_OUT1分频器，来降低CK_OUT1频率 由软件置位或清零 000000：CK_OUT1被1分频 000001：CK_OUT1被2分频 000010：CK_OUT1被3分频 … 111111：CK_OUT1被64分频
7:6	保留	必须保持复位值。
5:0	CKOUT0DIV[5:0]	CK_OUT0分频器，来降低CK_OUT0频率 由软件置位或清零

000000：CK_OUT0被1分频
000001：CK_OUT0被2分频
000010：CK_OUT0被3分频
…
111111：CK_OUT0被64分频

5.3.21. PLLT 控制寄存器（RCU_PLLTCTL）

地址偏移：0x90
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
保留.		PLLTSTB	PLLTEN	保留.											
		r	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
31:30	保留	必须保持复位值。
29	PLLTSTB	PLLT时钟稳定标志位 硬件置1来表示PLLT输出时钟是否稳定待用 0：PLLT未稳定 1：PLLT已稳定
28	PLLTEN	PLLT使能 软件置位或复位。当进入深度睡眠或待机模式时由硬件复位 0：PLLT被关闭 1：PLLT被打开
27:0	保留	必须保持复位值。

5.3.22. PLLT 中断寄存器（RCU_PLLTINT）

地址偏移：0x94
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.	PLLTSTBIE	保留							PLLTSTBIF	保留					
	rw								r						

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PLLTSTBIC	PLLT时钟稳定中断清零 软件写1复位PLLTSTBIF标志位 0：不复位PLLTSTBIF标志位 1：复位PLLTSTBIF标志位
21:15	保留	必须保持复位值。
14	PLLTSTBIE	PLLT时钟稳定中断使能 软件置位和复位来使能/禁止PLLT时钟稳定中断 0：禁止PLLT时钟稳定中断 1：使能PLLT时钟稳定中断
13:7	保留	必须保持复位值。
6	PLLTSTBIF	PLLT时钟稳定中断标志位 当PLPLLTL时钟稳定且PLLTSTBIE位被置1时由硬件置1 软件置位PLLTSTBIC位时清除该位 0：无PLLT时钟稳定中断产生 1：产生PLLT时钟稳定中断
5:0	保留	必须保持复位值。

5.3.23. PLLT 配置寄存器（RCU_PLLTCFG）

地址偏移：0x98

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLTSEL	PLLTRPSC[2:0]			保留										TLIPSC[1:0]	
rw	rw													rw	

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

位/位域	名称	描述
31	PLLTSEL	PLLT时钟源选择 只有当PLLT失能时才可以对这个位进行写操作

		0：CK_IRC8M被选择为PLLT时钟的时钟源 1：CK_HXTAL被选择为PLLT时钟的时钟源
30:28	PLLTRPSC[2:0]	PLLTR预分频因子 由软件置位或清零来控制TLI时钟频率。 当PLLT失能时才可以写这些位 000：PLLTRPSC = 0，错误配置 001：PLLTRPSC = 1，错误配置 010：PLLTRPSC = 2 ... 111：PLLTRPSC = 7
27:18	保留	必须保持复位值。
17:16	TLIPSC[1:0]	TLI预分频因子 由软件置位或清零来控制CK_TLI的频率 CK_TLI的频率 = f（PLLTR） / TLIPSC，其中2≤TLIPSC≤16 00：TLIPSC = 2 01：TLIPSC = 4 10：TLIPSC = 8 11：TLIPSC = 16
15	保留	必须保持复位值。
14:6	PLLTMF[8:0]	PLLT的倍频因子，用于倍频VCO的输入时钟源 软件设置和清除来控制VCO的输入时钟源的倍频因子. 当PLLT失能的时候对这些位进行写。 对这些位只能进行半字或字的写入操作。 VCO输出时钟频率= VCO输入时钟频率 x PLLTMF，并且49 ≤ PLLTMF≤432 000000000：PLLTMF = 0，错误配置 000000001：PLLTMF = 1，错误配置 ..... 000110000：PLLTMF = 48，错误配置 000110001：PLLTMF = 49 ... 011000000：PLLTMF = 192 011000001：PLLTMF = 193 ... 110110000：PLLTMF = 432 110110000：PLLTMF = 433，错误配置 ... 111111111：PLLTMF = 511，错误配置
5:0	PLLTPSC[5:0]	PLLT预分频选择 当PLLT失能时才能对这些位进行写操作。 **注意**：软件必须正确设置这些位，以确保VCO的输入时钟频率在1 MHz到2MHz之间，推荐设置VCO的输入时钟频率为 2 MHz。

VCO输入时钟频率 = PLLT输入时钟频率 / PLLTPSC，并且 2 ≤PLLTPSC≤ 63

000000：PLLTPSC = 0，错误配置

000001：PLLTPSC = 1，错误配置

000010：PLLTPSC = 2

000011：PLLTPSC = 3

000100：PLLTPSC = 4

...

111110：PLLTPSC = 62

111111：PLLTPSC = 63

6. 中断/事件控制器（EXTI）

6.1. 简介

Cortex-M3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®-M3技术参考手册》。

EXTI（中断/事件控制器）包括20个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

6.2. 主要特性

- Cortex-M3 系统异常；
- 多达 90 种可屏蔽的外设中断；
- 4 位中断优先级配置位——16 个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI 中有多达 20 个相互独立的边沿检测电路；
- 3 种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

6.3. 中断功能描述

Arm Cortex-M3处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序(ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。*表6-1. Cortex-M3中的NVIC异常类型*列出了Cotrex-M3中的NVIC异常类型。

表 6-1. Cortex-M3 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

SysTick校准值设为15000，SysTick时钟频率配置为HCLK/8，此时若HCLK时钟被配置为120MHz，则SysTick中断会1ms响应一次。

表 6-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000 0040
IRQ 1	17	连至 EXTI 线的 LVD 中断	0x0000 0044
IRQ 2	18	侵入检测中断	0x0000 0048
IRQ 3	19	RTC 全局中断	0x0000 004C
IRQ 4	20	FMC 全局中断	0x0000 0050
IRQ 5	21	RCU 全局中断	0x0000 0054
IRQ 6	22	EXTI 线 0 中断	0x0000 0058
IRQ 7	23	EXTI 线 1 中断	0x0000 005C
IRQ 8	24	EXTI 线 2 中断	0x0000 0060
IRQ 9	25	EXTI 线 3 中断	0x0000 0064
IRQ 10	26	EXTI 线 4 中断	0x0000 0068
IRQ 11	27	DMA0 通道 0 全局中断	0x0000 006C
IRQ 12	28	DMA0 通道 1 全局中断	0x0000 0070
IRQ 13	29	DMA0 通道 2 全局中断	0x0000 0074
IRQ 14	30	DMA0 通道 3 全局中断	0x0000 0078
IRQ 15	31	DMA0 通道 4 全局中断	0x0000 007C
IRQ 16	32	DMA0 通道 5 全局中断	0x0000 0080
IRQ 17	33	DMA0 通道 6 全局中断	0x0000 0084
IRQ 18	34	ADC0 和 ADC1 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

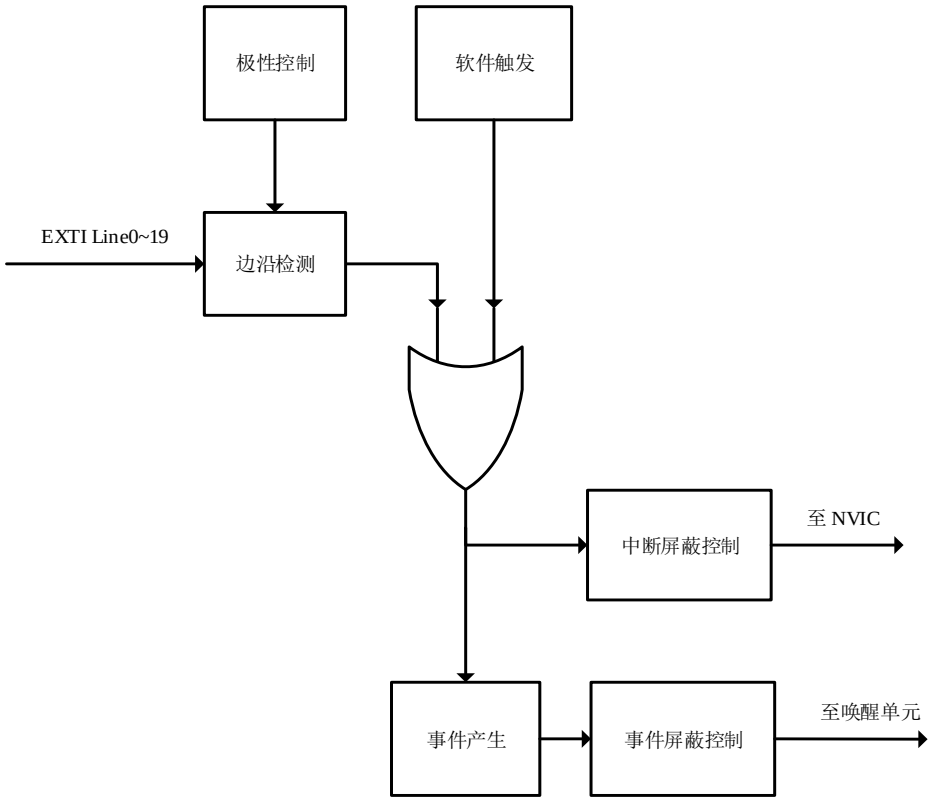
中断编号	向量编号	外设中断描述	向量地址
IRQ 21	37	CAN0 接收 1 中断	0x0000 0094
IRQ 22	38	CAN0 的 EWMC 中断	0x0000 0098
IRQ 23	39	EXTI 线[9:5]中断	0x0000 009C
IRQ 24	40	TIMER0 中止中断和 TIMER8 全局中断	0x0000 00A0
IRQ 25	41	TIMER0 更新中断和 TIMER9 全局中断	0x0000 00A4
IRQ 26	42	TIMER0 触发和通信中断和 TIMER10 全局中断	0x0000 00A8
IRQ 27	43	TIMER0 通道捕获比较中断	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 全局中断	0x0000 00D0
IRQ 37	53	USART0 全局中断	0x0000 00D4
IRQ 38	54	USART1 全局中断	0x0000 00D8
IRQ 39	55	USART2 全局中断	0x0000 00DC
IRQ 40	56	EXTI 线[15:10]中断	0x0000 00E0
IRQ 41	57	连至 EXTI 线的 RTC 闹钟中断	0x0000 00E4
IRQ 42	58	连至 EXTI 线的 USBFS 唤醒中断	0x0000 00E8
IRQ 43	59	TIMER7 中止中断和 TIMER11 全局中断	0x0000 00EC
IRQ 44	60	TIMER7 更新中断和 TIMER12 全局中断	0x0000 00F0
IRQ 45	61	TIMER7 触发和通信中断和 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	EXMC 全局中断	0x0000 0100
IRQ 49	65	SDIO 全局中断	0x0000 0104
IRQ 50	66	TIMER4 全局中断	0x0000 0108
IRQ 51	67	SPI2 全局中断	0x0000 010C
IRQ 52	68	UART3 全局中断	0x0000 0110
IRQ 53	69	UART4 全局中断	0x0000 0114
IRQ 54	70	TIMER5 全局中断	0x0000 0118
IRQ 55	71	TIMER6 全局中断	0x0000 011C
IRQ 56	72	DMA1 通道 0 全局中断	0x0000 0120
IRQ 57	73	DMA1 通道 1 全局中断	0x0000 0124

中断编号	向量编号	外设中断描述	向量地址
IRQ 58	74	DMA1 通道 2 全局中断	0x0000 0128
IRQ 59	75	DMA1 通道 3 全局中断	0x0000 012C
IRQ 60	76	DMA1 通道 4 全局中断	0x0000 0130
IRQ 61	77	以太网全局中断	0x0000 0134
IRQ 62	78	连至 EXTI 线的以太网唤醒中断	0x0000 0138
IRQ 63	79	CAN1 发送中断	0x0000 013C
IRQ 64	80	CAN1 接收 0 中断	0x0000 0140
IRQ 65	81	CAN1 接收 1 中断	0x0000 0144
IRQ 66	82	CAN1 的 EWMC 中断	0x0000 0148
IRQ 67	83	USBFS 全局中断	0x0000 014C
IRQ 68	84	保留	保留
IRQ 69	85	DMA1 通道 5 中断请求	0x0000 0154
IRQ 70	86	DMA1 通道 6 中断请求	0x0000 0158
IRQ 71	87	USART5 全局中断	0x0000 015C
IRQ 72	88	I2C2 事件中断	0x0000 0160
IRQ 73	89	I2C2 错误中断	0x0000 0164
IRQ 74 – IRQ 77	90 - 93	保留	保留
IRQ 78	94	DCI 全局中断	0x0000 0178
IRQ 79	95	CAU 全局中断	0x0000 017C
IRQ 80	96	HAU 或 TRNG 全局中断	0x0000 0180
IRQ 81	97	保留	保留
IRQ 82	98	UART6 全局中断	0x0000 0188
IRQ 83	99	UART7 全局中断	0x0000 018C
IRQ 84 – IRQ 87	100 - 103	保留	保留
IRQ 88	104	TLI 全局中断	0x0000 01A0
IRQ 89	105	TLI 全局错误中断	0x0000 01A4

注意：IRQ61与IRQ62只用于GD32F207xx系列

6.4. 外部中断及事件(EXTI)框图

图 6-1. EXTI 框图



6.5. 外部中断及事件功能概述

EXTI包含多达20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4根线。(包括LVD、RTC闹钟、USB唤醒、以太网唤醒)。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。Cortex-M3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，EXTI能唤醒处理器及整个系统，例如一个特定的I/O管脚电平翻转或者RTC闹钟动作。

表 6-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0/PB0/PC0/PD0/PE0/PF0/PG0/PH0/PI0
1	PA1/PB1/PC1/PD1/PE1/PF1/PG1/PH1/PI1
2	PA2/PB2/PC2/PD2/PE2/PF2/PG2/PH2/PI2
3	PA3/PB3/PC3/PD3/PE3/PF3/PG3/PH3/PI3
4	PA4/PB4/PC4/PD4/PE4/PF4/PG4/PH4/PI4
5	PA5/PB5/PC5/PD5/PE5/PF5/PG5/PH5/PI5
6	PA6/PB6/PC6/PD6/PE6/PF6/PG6/PH6/PI6
7	PA7/PB7/PC7/PD7/PE7/PF7/PG7/PH7/PI7
8	PA8/PB8/PC8/PD8/PE8/PF8/PG8/PH8/PI8
9	PA9/PB9/PC90/PD90/PE9/PF9/PG9/PH9/PI9
10	PA10/PB10/PC10/PD10/PE10/PF10/PG10/PH10/PI10
11	PA11/PB11/PC11/PD11/PE11/PF11/PG11/PH11/PI11
12	PA12/PB12/PC12/PD12/PE12/PF12/PG12/PH12
13	PA13/PB13/PC13/PD13/PE13/PF13/PG13/PH13
14	PA14/PB14/PC14/PD14/PE14/PF14/PG14/PH14
15	PA15/PB15/PC15/PD15/PE15/PF15/PG15/PH15
16	LVD
17	RTC 闹钟
18	USB 唤醒
19	Ethernet 唤醒

注意：EXTI19只用于GD32F207xx系列

6.6. EXTI 寄存器

EXTI 基地址：0x4001 0400

6.6.1. 中断使能寄存器 (EXTI_INTEN)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												INTEN19	INTEN18	INTEN17	INTEN16
												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
31:20	保留	必须保持复位值。
19:0	INTENx	中断使能位 0：第x线中断被禁用 1：第x线中断被使能

6.6.2. 事件使能寄存器 (EXTI_EVEN)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												EVEN19	EVEN18	EVEN17	EVEN16
												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
31:20	保留	必须保持复位值。
19:0	EVENx	事件使能位 0：第x线事件被禁用 1：第x线事件被使能

6.6.3. 上升沿触发使能寄存器 (EXTI_RTEN)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												RTEN19	RTEN18	RTEN17	RTEN16
												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
31:20	保留	必须保持复位值。
19:0	RTENx	上升沿触发使能 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件请求）

6.6.4. 下降沿触发使能寄存器 (EXTI_FTEN)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												FTEN19	FTEN18	FTEN17	FTEN16
												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: 20	保留	必须保持复位值。
19: 0	FTENx	下降沿触发使能 0：第x线下降沿触发无效 1：第x线下降沿触发有效（中断/事件请求）

6.6.5. 软件中断事件寄存器 (EXTI_SWIEV)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												SWIEV19	SWIEV18	SWIEV17	SWIEV16
												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
31:20	保留	必须保持复位值。
19: 0	SWIEVx	中断/事件软件触发 0：禁用EXTI线x软件中断/事件请求 1：激活EXTI线x软件中断/事件请求

6.6.6. 挂起寄存器 (EXTI_PD)

地址偏移：0x14
复位值：0xXXXX XXXX

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												PD19	PD18	PD17	PD16
												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: 20	保留	必须保持复位值。
19: 0	PDx	中断挂起状态 0：EXTI线x没有被触发 1：EXTI线x被触发，对这些位写1，可将其清0。

7. 通用和备用输入/输出接口（GPIO 和 AFIO）

7.1. 简介

最多可支持140个通用I/O引脚（GPIO），分别为PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD15，PE0 ~ PE15，PF0 ~ PF15，PG0 ~ PG15，PH0 ~ PH15和PI0 ~ PI11，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引脚，备用功能输入/输出都可。

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

7.2. 主要特征

- 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；

7.3. 功能说明

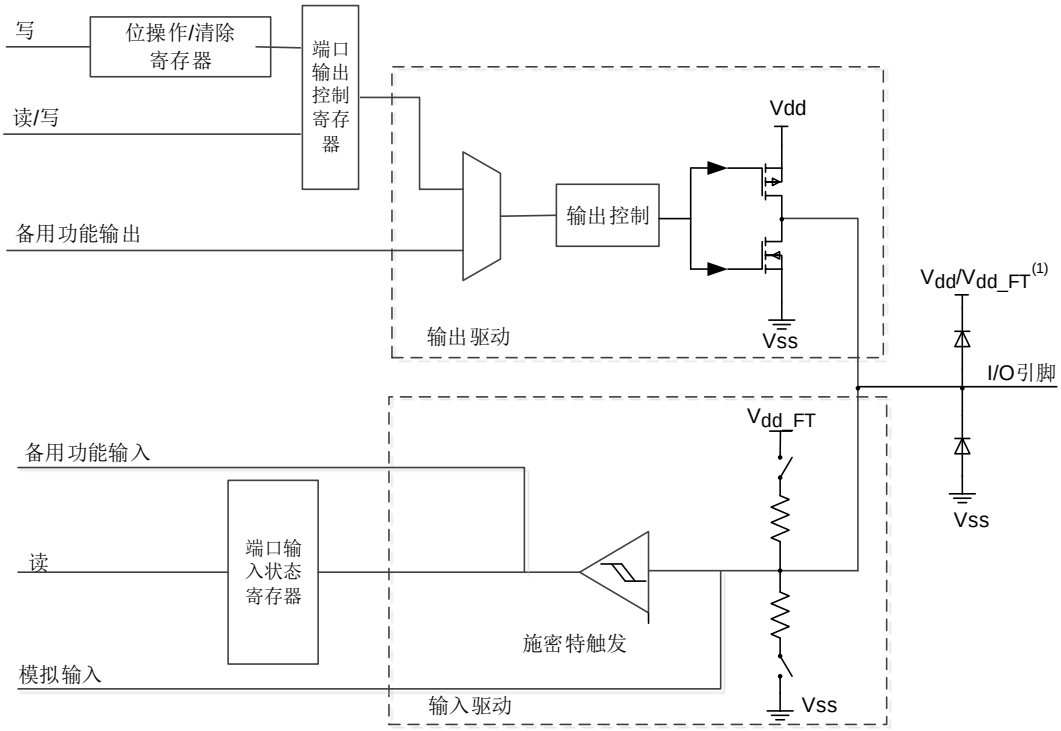
每个通用I/O端口都可以通过两个32位的控制寄存器（GPIOx_CTL0/ GPIOx_CTL1）和两个32位的数据寄存器（GPIOx_ISTAT，GPIOx_OCTL）配置为8种模式：模拟输入，浮空输入，上拉输入，下拉输入，GPIO推挽输出，GPIO开漏输出，AFIO推挽输出和AFIO开漏输出。详情请见*表7-1. GPIO配置表*。

表 7-1. GPIO 配置表

配置模式		CTL[1:0]	MD[1:0]	OCTL
输入	模拟	00	00	不使用
	浮空输入	01		不使用
	下拉输入	10		0
	上拉输入	10		1
普通输出（GPIO）	推挽	00	00：未用 01：最大速度到10MHz 10：最大速度到2MHz 11：最大速度到50MHz	0 或 1
	开漏	01		0 或 1
备用功能输出（AFIO）	推挽	10		不使用
	开漏	11		不使用

*图 7-1. 标准 I/O 端口和 5V 兼容 I/O 口的基本结构*为标准 I/O 端口的基本结构图。

图 7-1. 标准 I/O 端口和 5V 兼容 I/O 口的基本结构



1. V_{dd_FT} 对于 5V 兼容的 I/Os 是特殊的，它不同于 V_{dd}。

7.3.1. GPIO 引脚配置

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

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。

7.3.2. 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。

7.3.3. 备用功能（AF）

当端口配置为AFIO（设置GPIOx_CTL0/GPIOx_CTL1寄存器中的CTLy值为“0b10”或“0b11”，MDy位值为“0b01”，“0b10”或“0b11”）时，该端口用作外设备用功能。端口备用功能分配的详细介绍见芯片数据手册。

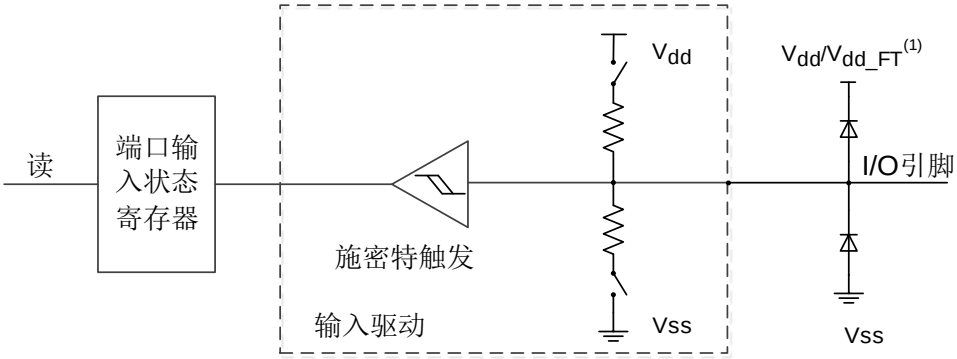
7.3.4. 输入配置

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 7-2. 输入配置*显示 I/O 引脚的输入配置。

图 7-2. 输入配置



1. V_{dd_FT} 对于 5V 兼容的 I/Os 是特殊的，它不同于 V_{dd}。

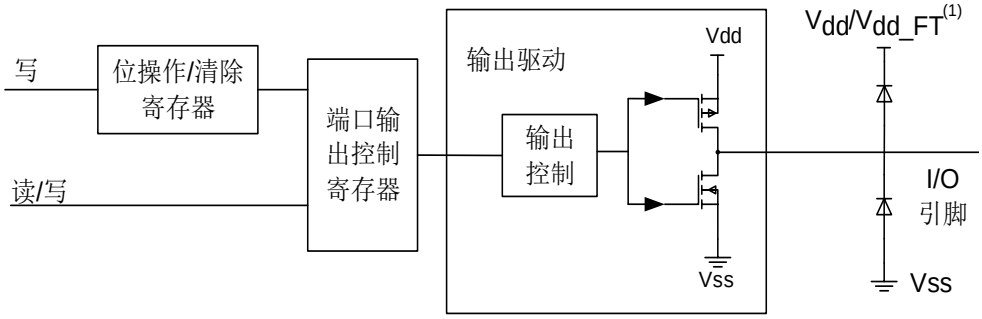
7.3.5. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 弱上拉和下拉电阻禁用；
- 输出缓冲器使能；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值；
- 对端口输入状态寄存器进行读操作，将获得当前I/O口的状态。

*图 7-3. 输出配置*是 I/O 端口的输出配置。

图 7-3. 输出配置



1. V_{dd_FT} 对于 5V 兼容的 I/Os 是特殊的，它不同于 V_{dd}。

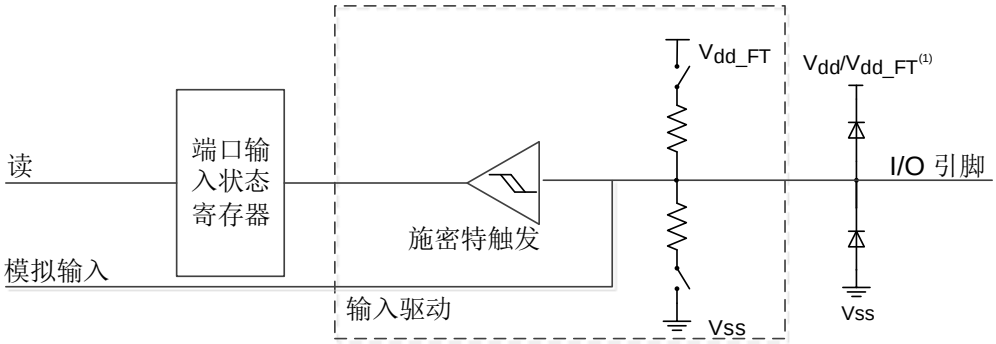
7.3.6. 模拟配置

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 7-4. 模拟配置*是 I/O 端口的模拟模式配置。

图 7-4. 模拟配置



1. V_{dd_FT} 对于 5V 兼容的 I/Os 是特殊的，它不同于 V_{dd}。

7.3.7. 备用功能（AF）配置

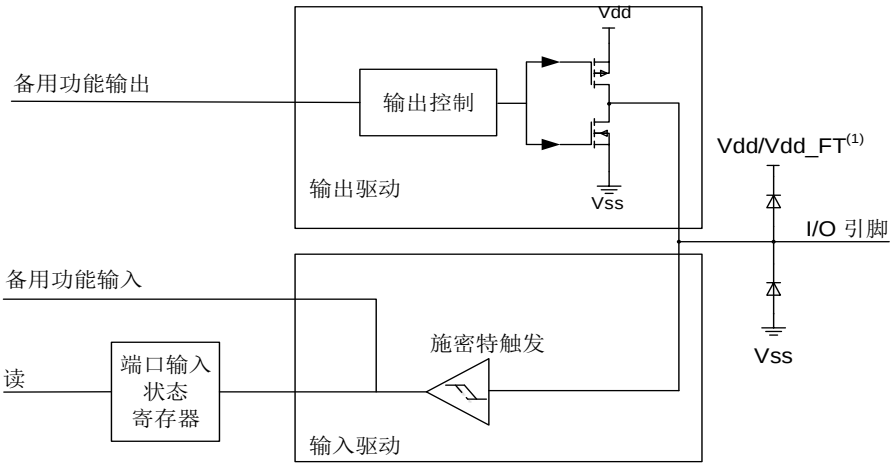
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

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；
- 弱上拉/下拉电阻禁止；
- 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 7-5. 备用功能配置*是 I/O 端口备用功能配置图。

图 7-5. 备用功能配置



1. V_{dd_FT} 对于 5V 兼容的 I/Os 是特殊的，它不同于 V_{dd}。

7.3.8. IO 引脚功能选择

每个I/O引脚都可以配置很多功能，每个功能都是通过对GPIO寄存器配置实现的。

GPIO

通过配置GPIOx_CTL0/ GPIOx_CTL1中的相应MDy位为0b00，可以配置I/O引脚为输入功能。通过配置相应MDy位为0b01，0b10或0b11，CTLy位为0b00（GPIO推挽输出）或0b01（GPIO开漏输出），将I/O引脚配置为输出功能。

备用功能

通过配置GPIOx_CTL0/ GPIOx_CTL1中的相应MDy位为0b00，可以配置I/O引脚为备用输入功能。通过配置GPIOx_CTL0/ GPIOx_CTL1中的相应MDy位为0b01，0b10或0b11，相应CTLy位为0b00（AF推挽输出）或0b01（AF开漏输出），每个I/O引脚可以配置为备用输出功能。

7.3.9. GPIO 锁定功能

GPIO的锁定机制可以保护I/O端口的配置。

被保护的寄存器有GPIOx_CTL0和GPIOx_CTL1。通过配置32位锁定寄存器（GPIOx_LOCK）可以锁定I/O端口的配置。通过特定的锁定序列配置GPIOx_LOCK中的LKK位和LKy位，相应的端口位被锁定，直到下一个复位前，相应端口位的配置都不能修改。建议在电源驱动模块的配置中使用锁定功能。

7.4. I/O 重映射功能和调试配置

7.4.1. 介绍

为了扩展GPIO的灵活性或外设功能使用，通过配置AFIO端口配置寄存器（AFIO_PCFx，x=0..5），每个I/O引脚都可以配置多达4种不同的功能。通过使用外设IO的重映射功能可以选择合适的引脚位置。另外，通过配置相应的EXTI源选择寄存器（AFIO_EXTISSx）选择触发中断或事件，GPIO引脚可以用作EXTI中断线。

7.4.2. 主要特性

- APB 从接口寄存器的访问
- EXTI 源选择
- 每个引脚具有多达 4 种备用功能的配置

7.4.3. JTAG/SWD 备用功能重映射

调试接口信号映射在GPIO端口的情况如*表7-2. 调试接口信号*所示。

表 7-2. 调试接口信号

备用功能	GPIO 端口
JTMS / SWDIO	PA13
JTCK / SWCLK	PA14
JTDI	PA15
JTDO / TRACESWO	PB3
NJTRST	PB4
TRACECK	PE2
TRACECK0	PE3
TRACECK1	PE4
TRACECK2	PE5
TRACECK3	PE6

为了减少用于调试的GPIO端口，用户可以配置AFIO_PCF0寄存器中的SWJ_CFG [2:0]位为不同的值。具体情况参照*表7-3. 调试端口映射*。

表 7-3. 调试端口映射

SWJ _CFG [2:0]	可用的调试端口	SWJ I/O 引脚分配				
		PA13/ JTMS/ SWDIO	PA14/ JTCK/S WCLK	PA15/ JTDI	PB3/ JTDO/ TRACE SWO	PB4/ NJTRST
000	完全 SWJ（JTAG-DP + SW-DP）（复位状态）	●	●	●	●	●
001	完全 SWJ（JTAG-DP + SW-DP）但没有 NJTRST	●	●	●	●	X
010	JTAG-DP 关闭 SW-DP 开启	●	●	X	X(1)	X
100	JTAG-DP 关闭 SW-DP 关闭	X	X	X	X	X
其他	禁用					

1. 只有在不使用异步跟踪时，I/O才能使用。

7.4.4. ADC AF 重映射

参考AFIO端口配置寄存器0（AFIO_PCF0）。

表 7-4. ADC0/1 常规转换外部触发备用功能重映射[1]

Register	ADC0	ADC1
ADC0_ETRGINS_REMAP = 0	ADC0 注入转换外部触发与 EXTI15 相连	-
ADC0_ETRGINS_REMAP = 1	ADC0 注入转换外部触发与 TIMER7_CH3 相连	-
ADC0_ETRGRER_REMAP = 0	ADC0 常规转换外部触发与 EXTI11 相连	-
ADC0_ETRGRER_REMAP = 1	ADC0 常规转换外部触发与 TIMER7_TRGO 相连	-
ADC1_ETRGINS_REMAP = 0	-	ADC1 注入转换外部触发与 EXTI15 相连
ADC1_ETRGINS_REMAP = 1	-	ADC1 注入转换外部触发与 TIMER7_CH3 相连
ADC1_ETRGRER_REMAP = 0	-	ADC1 常规转换外部触发与 EXTI11 相连
ADC1_ETRGRER_REMAP = 1	-	ADC1 常规转换外部触发与 TIMER7_TRGO 相连

1. 重映射仅仅适用于高密度和超高密度的产品。

7.4.5. TIMER AF 重映射

表 7-5. TIMER 备用功能重映射

备用功能	TIMERx_REMAP [1:0](x = 0，1，2)			
	TIMERx_REMAP(x = 8，9，10，12，13)		-	
	“0” /“00” (没有映射)	“1” /“01” (部分映射)	“10” (部分映射)	“11” (全映射)
TIMER0_ETI	PA12		-	PE7
TIMER0_CH0	PA8		-	PE9
TIMER0_CH1	PA9		-	PE11
TIMER0_CH2	PA10		-	PE13
TIMER0_CH3	PA11		-	PE14
TIMER0_BKIN	PB12(2)	PA6	-	PE15
TIMER0_CH0_ON	PB13(2)	PA7	-	PE8
TIMER0_CH1_ON	PB14(2)	PB0	-	PE10
TIMER0_CH2_ON	PB15(2)	PB1	-	PE12
TIMER1_CH0/TIMER1_ETI (1)	PA0	PA15	PA0	PA15
TIMER1_CH1	PA1	PB3	PA1	PB3
TIMER1_CH2	PA2		PB10	
TIMER1_CH3	PA3		PB11	
TIMER2_CH0	PA6	-	PA15	PC6
TIMER2_CH1	PA7	-	PB3	PC7
TIMER2_CH2	PB0	-	PB0	PC8
TIMER2_CH3	PB1	-	PB1	PC9
TIMER3_CH0	PB6	PD12	-	-
TIMER3_CH1	PB7	PD13	-	-
TIMER3_CH2	PB8	PD14	-	-
TIMER3_CH3	PB9	PD15	-	-
TIMER8_CH0	PA2	PE5	-	-
TIMER8_CH1	PA3	PE6	-	-
TIMER9_CH0	PB8	PF6	-	-
TIMER10_CH0	PB9	PF7	-	-
TIMER12_CH0	PA6	PF8	-	-
TIMER13_CH0	PA7	PF9	-	-

1. TIMER0 重映射仅仅适用于100引脚和144引脚的封装。
2. TIMER0 重映射不适用于36引脚封装。
3. TIMER1_CH0 and TIMER1_ETI 共用一个引脚，但不能同时使用。
4. TIMER1 重映射不适用于36引脚封装。
5. TIMER2 重映射仅仅适用于64引脚，100引脚和144引脚的封装。
6. TIMER3 重映射仅仅适用于100引脚和144引脚的封装。
7. TIMER8/9/10/12/13 参考备用功能映射和调试I/O配置寄存器1（AFIO_PCF1）。

表 7-6. TMER4 备用功能重映射[1]

备用功能	TIMER4CH3_IREMAP = 0	TIMER4CH3_IREMAP = 1
TIMER4_CH3	TIMER4_CH3 与 PA3 相连	IRC40K 内部时钟与 TIMER4_CH3 输入相连，用于校正

1. 重映射适用于高密度、超高密度和互联型的产品。

7.4.6. USART AF 重映射

参考 AFIO 端口配置寄存器 0 （AFIO_PCF0）。

表 7-7. USART0/1 备用功能重映射

寄存器	USART0	USART1	USART2
USART0_REMAP = 0	PA9(USART0_TX) PA10(USART0_RX)		-
USART0_REMAP = 1	PB6(USART0_TX) PB7(USART0_RX)		-
USART1_REMAP = 0	-	PA0(USART1_CTS) PA1(USART1_RTS) PA2(USART1_TX) PA3(USART1_RX) PA4(USART1_CK)	-
USART1_REMAP = 1 (1)	-	PD3(USART1_CTS) PD4(USART1_RTS) PD5(USART1_TX) PD6(USART1_RX) PD7(USART1_CK)	-
USART2_REMAP [1:0] = “00” （没有映射）	-	-	PB10(USART2_TX) PB11(USART2_RX) PB12(USART2_CK) PB13(USART2_CTS) PB14(USART2_RTS)
USART2_REMAP [1:0] =“01” （部分映射） (2)	-	-	PC10(USART2_TX) PC11(USART2_RX) PC12(USART2_CK) PB13(USART2_CTS)

寄存器	USART0	USART1	USART2
			PB14(USART2_RTS)
USART2_REMAP [1:0] ="11" （全映射） (3)	-	-	PD8(USART2_TX) PD9(USART2_RX) PD10(USART2_CK) PD11(USART2_CTS) PD12(USART2_RTS)

1. 重映射仅仅适用于 100 引脚和 144 引脚的封装。
2. 重映射仅仅适用于 64 引脚，100 引脚和 144 引脚的封装。
3. 重映射仅仅适用于 100 引脚和 144 引脚的封装。

7.4.7. I2C 备用功能重映射

参考 AFIO 端口配置寄存器 0 （AFIO_PCF0）。

表 7-8. I2C0/I2C1/I2C2 备用功能重映射

Register	I2Cx_SCL	I2Cx_SDA	I2Cx_SMBA
I2C0_REMAP = 0	PB6	PB7	
I2C0_REMAP = 1	PB8	PB9	
I2C1_REMAP[1:0] = 00/01	PB10	PB11	PB12
I2C1_REMAP[1:0] = 10	PH4	PH5	PH6
I2C1_REMAP[1:0] = 11	PF0	PF1	PF2
I2C2_REMAP1 = 1	PA8	PC9	PA9
I2C2_REMAP2 = 1	PH7	PH8	PH9

7.4.8. SPI 备用功能重映射

参考 AFIO 端口配置寄存器 0 （AFIO_PCF0）。

表 7-9. SPI0/SPI1/I2S1/SPI2/I2S2 备用功能重映射

Register	SPI0	SPI1/I2S1	SPI2/I2S2
SPI0_REMAP = 1	PA4(SPI0_NSS) PA5(SPI0_SCK) PA6(SPI0_MISO) PA7(SPI0_MOSI) PA2(SPI0_IO2) PA3(SPI0_IO3)		-
SPI0_REMAP = 1	PA15(SPI0_NSS) PB3(SPI0_SCK) PB4(SPI0_MISO) PB5(SPI0_MOSI) PB6(SPI0_IO2) PB7(SPI0_IO3)		=

Register	SPI0	SPI1/I2S1	SPI2/I2S2
SPI1_NSCK /IO _REMAP = 00/01		PB12(SPI1_NSS/ I2S1_WS) PB13(SPI1_SCK/ I2S1_CK) PB14(SPI1_MISO) PB15(SPI1_MOSI/I2S2_SD) PC6(I2S1_MCK)	
SPI1_NSCK /IO _REMAP = 10		PI0(SPI1_NSS/ I2S1_WS) PI1(SPI1_SCK/ I2S1_CK) PI2(SPI1_MISO) PI3(SPI1_MOSI/I2S2_SD)	
SPI1_NSCK /IO _REMAP = 11		PB9(SPI1_NSS/ I2S1_WS) PB10(SPI1_SCK/ I2S1_CK) PC2(SPI1_MISO) PC3(SPI1_MOSI/I2S2_SD)	
SPI1_SCK _REMAP = 1		PD3(SPI1_SCK/ I2S1_CK)	
SPI2_REMAP = 0	-		PA15(SPI2_NSS/ I2S2_WS) PB3(SPI2_SCK/ I2S2_CK) PB4(SPI2_MISO) PB5(SPI2_MOSI/I2S2_SD)
SPI2_REMAP = 1	-		PA4(SPI2_NSS/ I2S2_WS) PC10(SPI2_SCK/ I2S2_CK) PC11(SPI2_MISO) PC12(SPI2_MOSI/I2S2_SD)
SPI2_MOSI_REMAP = 1			PD6(SPI2_MOSI/I2S2_SD)

7.4.9. CAN 备用功能重映射

如下表所示，CAN0的信号引脚可以映射到端口A，端口B或端口D。对于端口D，重映射不适用与64引脚的封装中。

表 7-10. CAN0/1 备用功能重映射

寄存器 (1)	CAN0	CAN1
CAN0_REMAP[1:0] =“00”	PA11(CAN0_RX) PA12(CAN0_TX)	-
CAN0_REMAPI[1:0] =“10”	PB8(CAN0_RX) PB9(CAN0_TX)	-
CAN0_REMAP[1:0]	PD0(CAN0_RX)	-

寄存器 (1)	CAN0	CAN1
="11"(2)	PD1(CAN0_TX)	
CAN1_REMAP = "0"	-	PB12(CAN1_RX) PB13(CAN1_TX)
CAN1_REMAP = "1"	-	PB5(CAN1_RX) PB6(CAN1_TX)

1. CAN0_RX 和 CAN0_TX 用于互联型产品中; CAN_RX 在 CAN_TX 用于其他具有单个 CAN 接口的产品线中。
2. 重映射不适用于 36 引脚的封装。

7.4.10. ENET 备用功能重映射

表 7-11. ENET 备用功能重映射

寄存器	ENET
ENET_REMAP = "0"	PA7(RX_DV-CRS_DV) PC4(RXD0) PC5(RXD1) PB0(RXD2) PB1(RXD3)
ENET_REMAP = "1"	PD8(RX_DV-CRS_DV) PD9(RXD0) PD10(RXD1) PD11(RXD2) PD12(RXD3)
ENET_RX_HI_REMAP/ ENET_CRSCOL_REMAP/ ENET_TXD01_REMAP	PA0(MII_CRS) PA3(MII_COL) refer to ENET_REMAP(MII_RXD2) refer to ENET_REMAP(MII_RXD3) PB8(MII_TXD3) PB10(MII_RX_ER) PB11(MII_TX_EN-RMII_TX_EN) PB12(MII_TXD0- RMII_TXD0) PB13(MII_TXD1- RMII_TXD1)
ENET_RX_HI_REMAP/ ENET_CRSCOL_REMAP/ ENET_TXD01_REMAP	PH2(MII_CRS) PH3(MII_COL) PH6(MII_RXD2) PH7(MII_RXD3) PE2(MII_TXD3) PI10(MII_RX_ER) PG11(MII_TX_EN- RMII_TX_EN) PG13(MII_TXD0- RMII_TXD0) PG14(MII_TXD1- RMII_TXD1)

寄存器	ENET
PTP_PPS_REMAP = 1	PB5(PPS_OUT)
PPS_HI_REMAP = 1	PG8(PPS_OUT)

7.4.11. DCI 备用功能重映射

表 7-12. DCI 备用功能重映射

备用功能	DCI_Dx_ REMAP = "00"	DCI_Dx_ REMAP = "01"	DCI_Dx_ REMAP = "10"	DCI_Dx_ REMAP = "11"
DCI_D0	PA9	PC6		PH9
DCI_D1	PA10	PC7		PH10
DCI_D2	PC8	PE0	PG10	PH11
DCI_D3	PC9	PE1	PG11	PH12
DCI_D4	PC11	PE4		PH14
DCI_D5	PB6	PD3		PI4
DCI_D6	PB8	PE5		PI6
DCI_D7	PB9	PE6		PI7
DCI_D8	PC10	PH6		PI1
DCI_D9	PC12	PH7		PI2
DCI_D10	PB5	PD6		PI3
DCI_D11	PD2	PF10		PH15
DCI_D12	PF11	PG6		
DCI_D13	PG7	PG15		PI0
DCI_HSYNC	PA4	PH8		
DCI_VSYNC	PB7	PG9		PI5

7.4.12. TLI 备用功能重映射

表 7-13. TLI 备用功能重映射

备用功能	AFIO_PCF3	AFIO_PCF4
	TLI_xx_Pn_REMAP = 1(1)	TLI_xx_Pn_REMAP = 1(2)
TLI_R0		PH2 / PH4
TLI_R1		PH3 / PI3
TLI_R2	PC10	PH8
TLI_R3	PB0	PH9

备用功能	AFIO_PCF3	AFIO_PCF4
	TLI_xx_Pn_REMAP = 1[1]	TLI_xx_Pn_REMAP = 1[2]
TLI_R4	PA11	PH10
TLI_R5	PA12	PH11
TLI_R6	PA8 / PB1	PH12
TLI_R7	PE15 / PG6	
TLI_G0	PE5	
TLI_G1	PE6	
TLI_G2	PA6	PH13
TLI_G3	PG10 / PE11	PH14
TLI_G4	PB10	PH15
TLI_G5	PB11	PI0
TLI_G6	PC7	PI1
TLI_G7	PD3	PI2
TLI_B0	PE4	
TLI_B1		PG12
TLI_B2	PD6 / PG10	
TLI_B3	PD10 / PG11	
TLI_B4	PE12	PI4 / PG12
TLI_B5	PA3	PI5
TLI_B6	PB8	PI6
TLI_B7	PB9	PI7
TLI_DE	PE13 / PF10	
TLI_CLK	PE14 / PG7	
TLI_VSYNC	PA4	PI9
TLI_HSYNC	PC6	PI10

1. 参考AFIO端口配置寄存器3 （AFIO_PCF3）。
2. 参考AFIO端口配置寄存器4 （AFIO_PCF4）。

7.4.13. CLK 引脚 AF 重映射

当LXTAL关闭的时候，OSC32_IN和OSC32_OUT分别可以用做普通的I/O端口PC14和PC15。HXTAL的优先级比其他普通IO功能高。

注意：

1. 当1.8V区域关掉（进入待机模式）或备份区域由VBAT供电（不使用VDD供电），PC14/PC15不能用于普通IO功能，将会被设置为模拟模式。
2. 参考*电池备份域*章节中的IO口用法。

表 7-14. OSC32 引脚配置

备用功能	LXTAL= ON	LXTAL= OFF

PC14	OSC32_IN	PC14
PC15	OSC32_OUT	PC15

HXTAL晶振引脚OSC_IN/OSC_OUT可以用做普通的I/O端口PD0/PD1。

表 7-15. OSC 引脚配置 1

备用功能	PD01_REMAP = 0	PD01_REMAP = 1
PD0	PD0	OSC_IN
PD1	PD1	OSC_OUT

表 7-16. OSC 引脚配置 2

备用功能	PH01_REMAP = 0	PH01_REMAP = 1(1)
PH0		OSC_IN
PH1		OSC_OUT

1. 仅对于176引脚芯片，PH0/PH1默认为 OSC_IN/OSC_OUT，当PH01_REMAP =1，PH0/PH1为普通的IO端口。

7.5. GPIO 寄存器

GPIOA 基地址：0x4001 0800

GPIOB 基地址：0x4001 0C00

GPIOC 基地址：0x4001 1000

GPIOD 基地址：0x4001 1400

GPIOE 基地址：0x4001 1800

GPIOF 基地址：0x4001 1C00

GPIOG 基地址：0x4001 2000

GPIOH 基地址：0x4001 7400

GPIOI 基地址：0x4001 7800

AFIO 基地址：0x4001 0000

7.5.1. 端口控制寄存器 0（GPIOx_CTL0，x=A..I）

地址偏移：0x00
复位值：0x4444 4444

该寄存器只能按字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL7[1:0]	MD7[1:0]	CTL6[1:0]	MD6[1:0]	CTL5[1:0]	MD5[1:0]	CTL4[1:0]	MD4[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CTL3[1:0]	MD3[1:0]	CTL2[1:0]	MD2[1:0]	CTL1[1:0]	MD1[1:0]	CTL0[1:0]	MD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTL7[1:0]	Port 7配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
29:28	MD7[1:0]	Port 7模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
27:26	CTL6[1:0]	Port 6配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
25:24	MD6[1:0]	Port 6模式位

		该位由软件置位和清除。 参考MD0[1:0]的描述
23:22	CTL5[1:0]	Port 5配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
21:20	MD5[1:0]	Port 5模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
19:18	CTL4[1:0]	Port 4配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
17:16	MD4[1:0]	Port 4模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
15:14	CTL3[1:0]	Port 3配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
13:12	MD3[1:0]	Port 3模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
11:10	CTL2[1:0]	Port 2配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
9:8	MD2[1:0]	Port 2模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
7:6	CTL1[1:0]	Port 1配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
5:4	MD1[1:0]	Port 1模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
3:2	CTL0[1:0]	Pin 0配置位 该位由软件置位和清除。 输入模式（MD[1:0] =00） 00：模拟输入 01：浮空输入 10：上拉输入/下拉输入

		11：保留 输出模式（MD[1:0]>00） 00：GPIO推挽输出 01：GPIO开漏输出 10：AFIO推挽输出 11：AFIO开漏输出
1:0	MD0[1:0]	Port 0模式位 该位由软件置位和清除。 00：输入模式（复位状态） 01：输出模式，最大速度10MHz 10：输出模式，最大速度2MHz 11：输出模式，最大速度50MHz

7.5.2. 端口控制寄存器 1（GPIOx_CTL1，x=A..I）

地址偏移：0x04
复位值：0x4444 4444

该寄存器只能按字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	MD15[1:0]	CTL14[1:0]	MD14[1:0]	CTL13[1:0]	MD13[1:0]	CTL12[1:0]	MD12[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CTL11[1:0]	MD11[1:0]	CTL10[1:0]	MD10[1:0]	CTL9[1:0]	MD9[1:0]	CTL8[1:0]	MD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTL15[1:0]	Port 15配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
29:28	MD15[1:0]	Port 15模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
27:26	CTL14[1:0]	Port 14配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
25:24	MD14[1:0]	Port 14模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
23:22	CTL13[1:0]	Port13配置位 该位由软件置位和清除。

		参考CTL0[1:0]的描述
21:20	MD13[1:0]	Port 13模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
19:18	CTL12[1:0]	Port 12配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
17:16	MD12[1:0]	Port 12模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
15:14	CTL11[1:0]	Port 11配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
13:12	MD11[1:0]	Port 11模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
11:10	CTL10[1:0]	Port 10配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
9:8	MD10[1:0]	Port 10模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
7:6	CTL9[1:0]	Port 9配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
5:4	MD9[1:0]	Port 9模式位 该位由软件置位和清除。 参考MD0[1:0]的描述
3:2	CTL8[1:0]	Pin 8配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
1:0	MD8[1:0]	Port 8模式位 该位由软件置位和清除。 参考MD0[1:0]的描述

7.5.3. 端口输入状态寄存器（GPIOx_ISTAT，x=A..I）

地址偏移：0x08

复位值：0x0000 XXXX

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ISTATy	端口输入状态位（y=0..15） 这些位由软件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

7.5.4. 端口输出控制寄存器（GPIOx_OCTL，x=A..I）

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
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OCTLy	端口输出控制位（y=0..15） 这些位由软件置位和清除。 0：引脚输出低电平 1：引脚输出高电平

7.5.5. 端口位操作寄存器（GPIOx_BOP，x=A..I）

地址偏移：0x10
复位值：0x00000000

该寄存器只能按字（32 位）访问。

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

注意：如果CRy和BOPy同时设置，BOPy有优先级。

7.5.6. 位清除寄存器（GPIOx_BC，x=A..I）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRy	端口清除位y（y=0..15） 这些位由软件置位和清除。 0：相应OCTLy位没有改变 1：清除相应的OCTLy位

7.5.7. 端口配置锁定寄存器（GPIOx_LOCK，x=A..I）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK1	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

7.5.8. 事件控制寄存器（AFIO_EC）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	EOE	事件输出使能 该位由软件置位和清除。当设置该位后，Cortex的EVENTOUT输出将连接到由PORT[2:0]和PIN[3:0]位选择的I/O口。
6:4	PORT[2:0]	事件输出端口选择 这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的端口。 000：选择端口A 001：选择端口B 010：选择端口C 011：选择端口D 100：选择端口E
3:0	PIN[3:0]	事件输出引脚选择 这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的引脚。 0000：选择引脚0 0001：选择引脚1 0010：选择引脚2 … 1111：选择引脚15

7.5.9. AFIO 端口配置寄存器 0（AFIO_PCF0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	PTP_PPS_REMAP	TIMER1ITI1_REMAP	SPI2_REMAP	保留	SWJ_CFG[2:0]			ENET_PHY_SEL	CAN1_REMAP	ENET_REMAP	ADC1_ETRGREG_REMAP	ADC1_ETRGINS_REMAP	ADC0_ETRGREG_REMAP	ADC0_ETRGINS_REMAP	TIMER4CH3_IREMAP
	rw	rw	rw			w		rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01_REMAP	CAN0_REMAP [1:0]		TIMER3_REMAP	TIMER2_REMAP [1:0]		TIMER1_REMAP [1:0]		TIMER0_REMAP [1:0]		USART2_REMAP[1:0]		USART1_REMAP	USART0_REMAP	I2C0_REMAP	SPI0_REMAP
rw	rw		rw	rw		rw		rw		rw		rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	PTP_PPS_REMAP	以太网PTP PPS重映射 该位由软件置位和清除，使用以太网MAC_PPS输出到PB5引脚。

		0：PPT_PPS没有输出到PB5引脚 1：PPT_PPS输出到PB5引脚
29	TIMER1ITI1_REMAP	TIMER1内部触发0重映射 该位由软件置位和清除，用于控制TMER1_ITI1内部重映射。 0：TIMER1_ITI1内部连接到以太网PTP输出，用于校准 1：TIMER1_ITI1内部连接到USB OTG SOF（起始帧）输出，用于校准
28	SPI2_REMAP	SPI2/I2S2重映射 该位由软件置位和清除。 0：没有重映射（SPI2_NSS-I2S2_WS/PA15，SPI2_SCK-I2S2_CK/PB3，SPI2_MISO/PB4，SPI2_MOSI-I2S_SD/PB5） 1：完全重映射（SPI2_NSS-I2S2_WS/PA4，SPI2_SCK-I2S2_CK/PC10，SPI2_MISO/PC11，SPI2_MOSI-I2S_SD/PC12）
27	保留	必须保持复位值。
26:24	SWJ_CFG[2:0]	串行线JTAG配置 这些位只写（读这些位，将返回未定义值）。用于配置SWJ和跟踪复用功能的I/O口。SWJ（串行线JTAG）支持JTAG或SWD访问Cortex调试端口。系统复位后的默认状态是启用SWJ但没有跟踪功能，这种状态下，可以通过在JTMS/JTCK引脚上的发送特定的信号使能JTAG或SW（串行线）模式。 000：完全SWJ（JTAG-DP +SW-DP）：复位状态 001：完全SWJ（JTAG-DP +SW-DP）：没有NJTRST 010：JTAG-DP禁用和SW-DP使能 100：JTAG-DP禁用和SW-DP禁用 其他组合：无作用
23	ENET_PHY_SEL	以太网MII或RMII PHY选择 该位由软件置位和清除，它配置以太网内部MAC使用外部MII或RMII PHY。 0：配置以太网MAC使用外部MII PHY 1：配置以太网MAC使用外部RMII PHY
22	CAN1_REMAP	CAN1 I/O重映射 该位由软件置位和清除，控制着CAN1_TX和CAN1_RX引脚。 0：没有重映射（CAN1_RX/PB12，CAN_TX/PB13） 1：重映射（CAN1_RX/PB5，CAN_TX/PB6）
21	ENET_REMAP	以太网MAC I/O重映射 该位由软件置位和清除，控制着以太网MAC连接到PHY。 0：没有重映射（RX_DV-CRS_DV/PA7，RXD0/PC4，RXD1/PC5，RXD2/PB0，RXD3/PB1） 1：重映射（RX_DV-CRS_DV/PD8，RXD0/PD9，RXD1/PD10，RXD2/PD11， RXD3/PD12）
20	ADC1_ETRGREG _REMAP	ADC1常规转换外部触发重映射 该位由软件置位和清除。该位控制着触发输入与ADC1常规转换外部触发连

		接。当该位复位时，ADC1常规转换外部触发与EXTI11相连。当该位置位时，ADC1常规转换外部触发与TIM7_TRGO相连。
19	ADC1_ETRGINS_REMAP	ADC1注入转换外部触发重映射 该位由软件置位和清除。该位控制着触发输入与ADC1注入转换外部触发连接。当该位复位时，ADC1注入转换外部触发与EXTI15相连。当该位置位时，ADC1注入转换外部触发与TIM7_CH3相连。
18	ADC0_ETRGREG_REMAP	ADC0常规转换外部触发重映射 该位由软件置位和清除。该位控制着触发输入与ADC0常规转换外部触发连接。当该位复位时，ADC0常规转换外部触发与EXTI11相连。当该位置位时，ADC0常规转换外部触发与TIM7_TRGO相连。
17	ADC0_ETRGINS_REMAP	ADC0注入转换外部触发重映射 该位由软件置位和清除。该位控制着触发输入与ADC0注入转换外部触发连接。当该位复位时，ADC0注入转换外部触发与EXTI15相连。当该位置位时，ADC0注入转换外部触发与TIM7_CH3相连。
16	TIMER4CH3_IREMAP	TIMER4通道3内部重映射 该位由软件置位和清除，控制着TIMER4_CH3的内部重映射.当该位复位时TIMER4_CH3与PA3相连。当该位置位时，IRC40K内部振荡器与TIMER4_CH3相连，用于对IRC40K校准。 0：没有重映射 1：重映射
15	PD01_REMAP	OSC_IN/OSC_OUT重映射到Port D0/Port D1 该位由软件置位和清除。 0：没有重映射 1：OSC_IN重映射到PD0，OSC_OUT重映射到PD1
14:13	CAN0_REMAP[1:0]	CAN0备用功能接口重映射 这些位由软件置位和清除。 00：没有重映射（CAN0_RX/PA11，CAN0_TX/PA12） 01：没有使用 10：部分重映射（CAN0_RX/PB8，CAN0_TX/PB9） 11：完全重映射（CAN0_RX/PD0，CAN0_TX/PD1）
12	TIMER3_REMAP	TIMER3重映射 该位由软件置位和清除。 0：没有重映射（TIMER3_CH0/PB6，TIMER3_CH1/PB7，TIMER3_CH2/PB8，TIMER3_CH3/PB9） 1：完全重映射（TIMER3_CH0/PD12，TIMER3_CH1/PD13，TIMER3_CH2/PD14，TIMER3_CH3/PD15）
11:10	TIMER2_REMAP[1:0]	TIMER2重映射 这些位由软件置位和清除。 00：没有重映射（TIMER2_CH0/PA6，TIMER2_CH1/PA7，TIMER2_CH2/PB0，TIMER2_CH3/PB1）

		01：没有使用 10：部分重映射（TIMER2_CH0/PB4，TIMER2_CH1/PB5，TIMER2_CH2/PB0，TIMER2_CH3/PB1） 11：完全重映射（TIMER2_CH0/PC6，TIMER2_CH1/PC7，TIMER2_CH2/PC8，TIMER2_CH3/PC9）
9:8	TIMER1_REMAP[1:0]	TIMER1重映射 这些位由软件置位和清除。 00：没有重映射（TIMER1_CH0-TIMER1_ETI/PA0，TIMER1_CH1/PA1，TIMER1_CH2/PA2，TIMER1_CH3/PA3） 01：部分重映射0（TIMER1_CH0-TIMER1_ETI/PA15，TIMER1_CH1/PB3，TIMER1_CH2/PA2，TIMER1_CH3/PA3） 10：部分重映射1（TIMER1_CH0-TIMER1_ETI/PA0，TIMER1_CH1/PA1，TIMER1_CH2/PB10，TIMER1_CH3/PB11） 11：完全重映射（TIMER1_CH0-TIMER1_ETI/PA15，TIMER1_CH1/PB3，TIMER1_CH2/PB10，TIMER1_CH3/PB11）
7:6	TIMER0_REMAP[1:0]	TIMER0重映射 这些位由软件置位和清除。 00：没有重映射（TIMER0_ETI/PA12，TIMER0_CH0/PA8，TIMER0_CH1/PA9， TIMER0_CH2/PA10，TIMER0_CH3/PA11，TIMER0_BKIN/PB12，TIMER0_CH0_ON/PB13，TIMER0_CH1_ON/PB14，TIMER0_CH2_ON/PB15） 01：部分重映射（TIMER0_ETI/PA12，TIMER0_CH0/ PA8，TIMER0_CH1/PA9，TIMER0_CH2/PA10，TIMER0_CH3/PA11，TIMER0_BKIN/PA6，TIMER0_CH0_ON/PA7，TIMER0_CH1_ON/PB0，TIMER0_CH2_ON/PB1） 10：没有使用 11：完全重映射（TIMER0_ETI/PE7，TIMER0_CH0/ PE9，TIMER0_CH1/PE11，TIMER0_CH2/PE13，TIMER0_CH3/PE14，TIMER0_BKIN/PE15，TIMER0_CH0_ON/PE8，TIMER0_CH1_ON/PE10，TIMER0_CH2_ON/PE12）
5:4	USART2_REMAP[1:0]	USART2重映射 这些位由软件置位和清除。 00：没有重映射（USART2_TX/PB10，USART2_RX /PB11，USART2_CK/PB12，USART2_CTS/PB13，USART2_RTS/PB14） 01：部分重映射（USART2_TX/PC10，USART2_RX /PC11，USART2_CK/PC12，USART2_CTS/PB13，USART2_RTS/PB14） 10：没有使用 11：完全重映射（USART2_TX/PD9，USART2_RX /PD10，USART2_CK/PD11，USART2_CTS/PD12，USART2_RTS/PD13）
3	USART1_REMAP	USART1重映射 该位由软件置位和清除。

		0：没有重映射（USART1_CTS/PA0，USART1_RTS/PA1，USART1_TX/PA2，USART1_RX /PA3，USART1_CK/PA4） 1：重映射（USART1_CTS/PD3，USART1_RTS/PD4，USART1_TX/PD5，USART1_RX /PD6，USART1_CK/PD7）
2	USART0_REMAP	USART0重映射 该位由软件置位和清除。 0：没有重映射（USART0_TX/PA9，USART0_RX /PA10） 1：重映射（USART0_TX/PB6，USART0_RX /PB7）
1	I2C0_REMAP	I2C0重映射 该位由软件置位和清除。 0：没有重映射（I2C0_SCL/PB6，I2C0_SDA /PB7） 1：重映射（I2C0_SCL/PB8，I2C0_SDA /PB9）
0	SPI0_REMAP	SPI0重映射 该位由软件置位和清除。 0：没有重映射（SPI0_NSS/PA4，SPI0_SCK /PA5，SPI0_MISO /PA6，SPI0_MOSI /PA7，SPI0_IO2 /PA2，SPI0_IO3 /PA3） 1：重映射（SPI0_NSS/PA15，SPI0_SCK /PB3，SPI0_MISO /PB4，SPI0_MOSI /PB5，SPI0_IO2 /PB6，SPI0_IO3 /PB7）

7.5.10. EXTI 源选择寄存器 0 寄存器（AFIO_EXTISS0）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI 3源选择 0000：PA3引脚 0001：PB3引脚 0010：PC3引脚 0011：PD3引脚 0100：PE3引脚 0101：PF3引脚

		0110：PG3引脚 0111：PH3引脚 1000：PI3引脚 其他配置保留。
11:8	EXTI2_SS[3:0]	EXTI 2源选择 0000：PA2引脚 0001：PB2引脚 0010：PC2引脚 0011：PD2引脚 0100：PE2引脚 0101：PF2引脚 0110：PG2引脚 0111：PH2引脚 1000：PI2引脚 其他配置保留。
7:4	EXTI1_SS[3:0]	EXTI 1源选择 0000：PA1引脚 0001：PB1引脚 0010：PC1引脚 0011：PD1引脚 0100：PE1引脚 0101：PF1引脚 0110：PG1引脚 0111：PH1引脚 1000：PI1引脚 其他配置保留。
3:0	EXTI0_SS[3:0]	EXTI 0源选择 0000：PA0引脚 0001：PB0引脚 0010：PC0引脚 0011：PD0引脚 0100：PE0引脚 0101：PF0引脚 0110：PG0引脚 0111：PH0引脚 1000：PI0引脚 其他配置保留。

7.5.11. EXTI 源选择寄存器 1 寄存器（AFIO_EXTISS1）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7源选择 0000：PA7引脚 0001：PB7引脚 0010：PC7引脚 0011：PD7引脚 0100：PE7引脚 0101：PF7引脚 0110：PG7引脚 0111：PH7引脚 1000：PI7引脚 其他配置保留。
11:8	EXTI6_SS[3:0]	EXTI 6源选择 0000：PA6引脚 0001：PB6引脚 0010：PC6引脚 0011：PD6引脚 0100：PE6引脚 0101：PF6引脚 0110：PG6引脚 0111：PH6引脚 1000：PI6引脚 其他配置保留。
7:4	EXTI5_SS[3:0]	EXTI 5源选择 0000：PA5引脚 0001：PB5引脚 0010：PC5引脚 0011：PD5引脚 0100：PE5引脚 0101：PF5引脚 0110：PG5引脚 0111：PH5引脚 1000：PI5引脚

		其他配置保留。
3:0	EXTI4_SS[3:0]	EXTI 4源选择 0000：PA4引脚 0001：PB4引脚 0010：PC4引脚 0011：PD4引脚 0100：PE4引脚 0101：PF4引脚 0110：PG4引脚 0111：PH4引脚 1000：PI4引脚 其他配置保留。

7.5.12. EXTI 源选择寄存器 2 寄存器（AFIO_EXTISS2）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11源选择 0000：PA11引脚 0001：PB11引脚 0010：PC11引脚 0011：PD11引脚 0100：PE11引脚 0101：PF11引脚 0110：PG11引脚 0111：PH11引脚 1000：PI11引脚 其他配置保留。
11:8	EXTI10_SS[3:0]	EXTI 10源选择 0000：PA10引脚 0001：PB10引脚

		0010：PC10引脚 0011：PD10引脚 0100：PE10引脚 0101：PF10引脚 0110：PG10引脚 0111：PH10引脚 1000：PI10引脚 其他配置保留。
7:4	EXTI9_SS[3:0]	EXTI 9源选择 0000：PA9引脚 0001：PB9引脚 0010：PC9引脚 0011：PD9引脚 0100：PE9引脚 0101：PF9引脚 0110：PG9引脚 0111：PH9引脚 1000：PI9引脚 其他配置保留。
3:0	EXTI8_SS[3:0]	EXTI 8源选择 0000：PA8引脚 0001：PB8引脚 0010：PC8引脚 0011：PD8引脚 0100：PE8引脚 0101：PF8引脚 0110：PG8引脚 0111：PH8引脚 1000：PI8引脚 其他配置保留。

7.5.13. 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。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			

rw rw rw rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI 15源选择 0000：PA15引脚 0001：PB15引脚 0010：PC15引脚 0011：PD15引脚 0100：PE15引脚 0101：PF15引脚 0110：PG15引脚 其他配置保留。
11:8	EXTI14_SS[3:0]	EXTI 14源选择 0000：PA14引脚 0001：PB14引脚 0010：PC14引脚 0011：PD14引脚 0100：PE14引脚 0101：PF14引脚 0110：PG14引脚 其他配置保留。
7:4	EXTI13_SS[3:0]	EXTI 13源选择 0000：PA13引脚 0001：PB13引脚 0010：PC13引脚 0011：PD13引脚 0100：PE13引脚 0101：PF13引脚 0110：PG13引脚 其他配置保留。
3:0	EXTI12_SS[3:0]	EXTI 12源选择 0000：PA12引脚 0001：PB12引脚 0010：PC12引脚 0011：PD12引脚 0100：PE12引脚 0101：PF12引脚 0110：PG12引脚 其他配置保留。

7.5.14. AFIO 端口配置寄存器 1（AFIO_PCF1）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4.	3	2	1	0
保留					EXMC_NADV	TIMER13_REMAP	TIMER12_REMAP	TIMER10_REMAP	TIMER9_REMAP	TIMER8_REMAP	保留				
					rw	rw	rw	rw	rw	rw					

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	EXMC_NADV	EXMC_NADV连接/不连接 该位由软件置位和清除，控制着可选的EXMC_NADV信号 0：NADV信号连接到输出（默认值） 1：NADV信号没有连接，I/O引脚可以用于其他外设。
9	TIMER13_REMAP	TIMER13重映射 该位由软件置位和清除，控制着将TIMER13_CH0备用功能重映射到GPIO端口。 0：没有重映射（PA7） 1：重映射（PF9）
8	TIMER12_REMAP	TIMER12重映射 该位由软件置位和清除，控制着将TIMER12_CH0备用功能重映射到GPIO端口。 0：没有重映射（PA6） 1：重映射（PF8）
7	TIMER10_REMAP	TIMER10重映射 该位由软件置位和清除，控制着将TIMER10_CH0备用功能重映射到GPIO端口。 0：没有重映射（PB9） 1：重映射（PF7）
6	TIMER9_REMAP	TIMER9重映射 该位由软件置位和清除，控制着将TIMER9_CH0备用功能重映射到GPIO端口。 0：没有重映射（PB8） 1：重映射（PF6）
5	TIMER8_REMAP	TIMER8重映射 该位由软件置位和清除，控制着将TIMER8_CH0和TIMER8_CH1备用功能重映射到GPIO端口。 0：没有重映射（TIMER8_CH0连接到PA2和TIMER8_CH1连接到PA3）

		1：重映射（TIMER8_CH0重映射到PE5和TIMER8_CH1重映射到PE6）
4:0	保留	必须保持复位值。

7.5.15. AFIO 端口配置寄存器 2（AFIO_PCF2）

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
PH01_REMAP	保留	DCI_HSYNC_REMAP	DCI_D13_REMAP [1:0]		DCI_D12_REMAP	DCI_D11_REMAP [1:0]		DCI_D10_REMAP [1:0]		DCI_D9_REMAP [1:0]		DCI_D8_REMAP [1:0]		DCI_D7_REMAP [1:0]	
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
DCI_D6_REMAP [1:0]		DCI_D5_REMAP [1:0]		DCI_D4_REMAP [1:0]		DCI_D3_REMAP [1:0]		DCI_D2_REMAP [1:0]		DCI_D1_REMAP [1:0]		DCI_D0_REMAP [1:0]		DCI_VSYNC_REMAP [1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31	PH01_REMAP	PH0/PH1重映射 该位由软件置位和清除。 0：没有重映射（无PH0/PH1）（用作OSC_IN/OSC_OUT） 1：PH0/PH1，176引脚芯片中PH0/PH1重映射到OSC_IN/OSC_OUT
30	保留	必须保持复位值。
29	DCI_HSYNC_REMAP	DCI_HSYNC重映射 该位由软件置位和清除。 0：没有重映射（PA4） 1：DCI_HSYNC重映射到PH8
28:27	DCI_D13_REMAP[1:0]	DCI_D13重映射 该位由软件置位和清除。 00：没有重映射（PG7） 01：DCI_D13重映射到PG15 10：保留 11：DCI_D13重映射到PI0
26	DCI_D12_REMAP	DCI_D12重映射 该位由软件置位和清除。 0：没有重映射（PF11） 1：DCI_D12重映射到PG6
25:24	DCI_D11_	DCI_D11重映射

	REMAP[1:0]	该位由软件置位和清除。 00：没有重映射（PD2） 01：DCI_D11重映射到PF10 10：保留 11：DCI_D11重映射到PH15
23:22	DCI_D10_ REMAP[1:0]	DCI_D10重映射 该位由软件置位和清除。 00：没有重映射（PB5） 01：DCI_D10重映射到PD6 10：保留 11：DCI_D10重映射到PI3
21:20	DCI_D9_ REMAP[1:0]	DCI_D9重映射 该位由软件置位和清除。 00：没有重映射（PC12） 01：DCI_D9重映射到PH7 10：保留 11：DCI_D9重映射到PI2
19:18	DCI_D8_ REMAP[1:0]	DCI_D8重映射 该位由软件置位和清除。 00：没有重映射（PC10） 01：DCI_D8重映射到PH6 10：保留 11：DCI_D8重映射到PI1
17:16	DCI_D7_ REMAP[1:0]	DCI_D7重映射 该位由软件置位和清除。 00：没有重映射（PB9） 01：DCI_D7重映射到PE6 10：保留 11：DCI_D7重映射到PI7
15:14	DCI_D6_ REMAP[1:0]	DCI_D6重映射 该位由软件置位和清除。 00：没有重映射（PB8） 01：DCI_D6重映射到PE5 10：保留 11：DCI_D6重映射到PI6
13:12	DCI_D5_ REMAP[1:0]	DCI_D5重映射 该位由软件置位和清除。 00：没有重映射（PB6） 01：DCI_D5重映射到PD3 10：保留

		11：DCI_D5重映射到PI4
11:10	DCI_D4_ REMAP[1:0]	DCI_D4重映射 该位由软件置位和清除。 00：没有重映射（PC11） 01：DCI_D4重映射到PE4 10：保留 11：DCI_D4重映射到PH14
9:8	DCI_D3_ REMAP[1:0]	DCI_D3重映射 该位由软件置位和清除。 00：没有重映射（PC9） 01：DCI_D3重映射到PE1 10：DCI_D3重映射到PG11 11：DCI_D3重映射到PH12
7:6	DCI_D2_ REMAP[1:0]	DCI_D2重映射 该位由软件置位和清除。 00：没有重映射（PC8） 01：DCI_D2重映射到PE0 10：DCI_D2重映射到PG10 11：DCI_D2重映射到PH11
5:4	DCI_D1_ REMAP[1:0]	DCI_D1重映射 该位由软件置位和清除。 00：没有重映射（PA10） 01：DCI_D1重映射到PC7 10：保留 11：DCI_D1重映射到PH10
3:2	DCI_D0_ REMAP[1:0]	DCI_D0重映射 该位由软件置位和清除。 00：没有重映射（PA9） 01：DCI_D0重映射到PC6 10：保留 11：DCI_D0重映射到PH9
1:0	DCI_VSYNC_ REMAP[1:0]	DCI_VSYNC重映射 该位由软件置位和清除。 00：没有重映射（PB7） 01：DCI_VSYNC重映射到PG9 10：保留 11：DCI_VSYNC重映射到PI5

7.5.16. AFIO 端口配置寄存器 3（AFIO_PCF3）

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
TLI_B3_PG11_REMAP	TLI_B2_PG10_REMAP	TLI_G3_PG10_REMAP	TLI_CLK_PG7_REMAP	TLI_R7_PG6_REMAP	TLI_DE_PF10_REMAP	TLI_R7_PE15_REMAP	TLI_CLK_PE14_REMAP	TLI_DE_PE13_REMAP	TLI_B4_PE12_REMAP	TLI_G3_PE11_REMAP	TLI_G1_PE6_REMAP	TLI_G0_PE5_REMAP	TLI_B0_PE4_REMAP	TLI_B3_PD10_REMAP	TLI_B2_PD6_REMAP
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
TLI_G7_PD3_REMAP	TLI_R2_PC10_REMAP	TLI_G6_PC7_REMAP	TLI_HSYNC_PC6_REMAP	TLI_G5_PB11_REMAP	TLI_G4_PB10_REMAP	TLI_B7_PB9_REMAP	TLI_B6_PB8_REMAP	TLI_R6_PB1_REMAP	TLI_R3_PB0_REMAP	TLI_R5_PA12_REMAP	TLI_R4_PA11_REMAP	TLI_R6_PA8_REMAP	TLI_G2_PA6_REMAP	TLI_VSYNC_PA4_REMAP	TLI_B5_PA3_REMAP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	TLI_B3_PG11_REMAP	TLI_B3_PG11重映射 该位由软件置位和清除。 0：TLI_B3没有重映射到PG11 1：TLI_B3重映射到PG11
30	TLI_B2_PG10_REMAP	TLI_B2_PG10重映射 该位由软件置位和清除。 0：TLI_B2没有重映射到PG10 1：TLI_B2重映射到PG10
29	TLI_G3_PG10_REMAP	TLI_G3_PG10重映射 该位由软件置位和清除。 0：TLI_G3没有重映射到PG10 1：TLI_G3重映射到PG10
28	TLI_CLK_PG7_REMAP	TLI_CLK_PG7重映射 该位由软件置位和清除。 0：TLI_CLK没有重映射到PG7 1：TLI_CLK重映射到PG7
27	TLI_R7_PG6_REMAP	TLI_R7_PG6重映射 该位由软件置位和清除。 0：TLI_R7没有重映射到PG6 1：TLI_R7重映射到PG6
26	TLI_DE_PF10_REMAP	TLI_DE_PF10重映射 该位由软件置位和清除。

		0：TLI_DE没有重映射到PF10 1：TLI_DE重映射到PF10
25	TLI_R7_PE15 _REMAP	TLI_R7_PE15重映射 该位由软件置位和清除。 0：TLI_R7没有重映射到PE15 1：TLI_R7重映射到PE15
24	TLI_CLK_PE14 _REMAP	TLI_CLK_PE14重映射 该位由软件置位和清除。 0：TLI_CLK没有重映射到PE14 1：TLI_CLK重映射到PE14
23	TLI_DE_PE13 _REMAP	TLI_DE_PE13重映射 该位由软件置位和清除。 0：TLI_DE没有重映射到PE13 1：TLI_DE重映射到PE13
22	TLI_B4_PE12 _REMAP	TLI_B4_PE12重映射 该位由软件置位和清除。 0：TLI_B4没有重映射到PE12 1：TLI_B4重映射到PE12
21	TLI_G3_PE11 _REMAP	TLI_G3_PE11重映射 该位由软件置位和清除。 0：TLI_G3没有重映射到PE11 1：TLI_G3重映射到PE11
20	TLI_G1_PE6 _REMAP	TLI_G1_PE6重映射 该位由软件置位和清除。 0：TLI_G1没有重映射到PE6 1：TLI_G1重映射到PE6
19	TLI_G0_PE5 _REMAP	TLI_G0_PE5重映射 该位由软件置位和清除。 0：TLI_G0没有重映射到PE5 1：TLI_G0重映射到PE5
18	TLI_B0_PE4 _REMAP	TLI_B0_PE4重映射 该位由软件置位和清除。 0：TLI_B0没有重映射到PE4 1：TLI_B0重映射到PE4
17	TLI_B3_PD10 _REMAP	TLI_B3_PD10重映射 该位由软件置位和清除。 0：TLI_B3重映射到PD10 1：TLI_B3没有重映射到PD10
16	TLI_B2_PD6	TLI_B2_PD6重映射

	_REMAP	该位由软件置位和清除。 0：TLI_B2没有重映射到PD6 1：TLI_B2重映射到PD6
15	TLI_G7_PD3 _REMAP	TLI_G7_PD3重映射 该位由软件置位和清除。 0：TLI_G7没有重映射到PD3 1：TLI_G7重映射到PD3
14	TLI_R2_PC10 _REMAP	TLI_R2_PC10重映射 该位由软件置位和清除。 0：TLI_R2没有重映射到PC10 1：TLI_R2重映射到PC10
13	TLI_G6_PC7 _REMAP	TLI_G6_PC7重映射 该位由软件置位和清除。 0：TLI_G6没有重映射到PC7 1：TLI_G6重映射到PC7
12	TLI_HSYNC_PC6 _REMAP	TLI_HSYNC_PC6重映射 该位由软件置位和清除。 0：TLI_HSYNC没有重映射到PC6 1：TLI_HSYNC重映射到PC6
11	TLI_G5_PB11 _REMAP	TLI_G5_PB11重映射 该位由软件置位和清除。 0：TLI_G5没有重映射到PB11 1：TLI_G5重映射到PB11
10	TLI_G4_PB10 _REMAP	TLI_G4_PB10重映射 该位由软件置位和清除。 0：TLI_G4没有重映射到PB10 1：TLI_G4重映射到PB10
9	TLI_B7_PB9 _REMAP	TLI_B7_PB9重映射 该位由软件置位和清除。 0：TLI_B7没有重映射到PB9 1：TLI_B7重映射到PB9
8	TLI_B6_PB8 _REMAP	TLI_B6_PB8重映射 该位由软件置位和清除。 0：TLI_B6没有重映射到PB8 1：TLI_B6重映射到PB8
7	TLI_R6_PB1 _REMAP	TLI_R6_PB1重映射 该位由软件置位和清除。 0：TLI_R6没有重映射到PB1 1：TLI_R6重映射到PB1

6	TLI_R3_PB0 _REMAP	TLI_R3_PB0重映射 该位由软件置位和清除。 0：TLI_R3没有重映射到PB0 1：TLI_R3没有重映射到PB0
5	TLI_R5_PA12 _REMAP	TLI_R5_PA12重映射 该位由软件置位和清除。 0：TLI_R5没有重映射到PA12 1：TLI_R5重映射到PA12
4	TLI_R4_PA11 _REMAP	TLI_R4_PA11重映射 该位由软件置位和清除。 0：TLI_R4没有重映射到PA11 1：TLI_R4重映射到PA11
3	TLI_R6_PA8 _REMAP	TLI_R6_PA8重映射 该位由软件置位和清除。 0：TLI_R6没有重映射到PA8 1：TLI_R6重映射到PA8
2	TLI_G2_PA6 _REMAP	TLI_G2_PA6重映射 该位由软件置位和清除。 0：TLI_G2没有重映射到PA6 1：TLI_G2重映射到PA6
1	TLI_VSYNC_PA4 _REMAP	TLI_VSYNC_PA4 重映射 该位由软件置位和清除。 0：TLI_VSYNC没有重映射到PA4 1：TLI_VSYNC重映射到PA4
0	TLI_B5_PA3 _REMAP	TLI_B5_PA3重映射 该位由软件置位和清除。 0：TLI_B5没有重映射到PA3 1：TLI_B5重映射到PA3

7.5.17. AFIO 端口配置寄存器 4（AFIO_PCF4）

地址偏移：0x44
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							SPI2_MOSI_REMAP	SPI1_SCK_REMAP	TLI_R1_PI3_REMAP	TLI_R0_PH4_REMAP	TLI_HSYNC_PI10_REMAP	TLI_VSYNC_PI9_REMAP	TLI_B7_PI7_REMAP	TLI_B6_PI6_REMAP	TLI_B5_PI5_REMAP
							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
TLI_B4_PI4_REMAP	TLI_G7_PI2_REMAP	TLI_G6_PI1_REMAP	TLI_G5_PI0_REMAP	TLI_G4_PH15_REMAP	TLI_G3_PH14_REMAP	TLI_G2_PH13_REMAP	TLI_R6_PH12_REMAP	TLI_R5_PH11_REMAP	TLI_R4_PH10_REMAP	TLI_R3_PH9_REMAP	TLI_R2_PH8_REMAP	TLI_R1_PH3_REMAP	TLI_R0_PH2_REMAP	TLI_B1_PG12_REMAP	TLI_B4_PG12_REMAP
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	SPI2_MOSI_REMAP	SPI2_MOSI重映射 该位由软件置位和清除。 0：没有影响，参考SPI2_REMAP 1：SPI2_MOSI重映射到PD6
23	SPI1_SCK_REMAP	SPI1_SCK重映射 该位由软件置位和清除。 0：没有影响，参考SPI1_NSCK_REMAP 1：SPI1_SCK重映射到PD3
22	TLI_R1_PI3_REMAP	TLI_R1_PI3重映射 该位由软件置位和清除。 0：TLI_R1没有重映射到PI3 1：TLI_R1重映射到PI3
21	TLI_R0_PH4_REMAP	TLI_R0_PH4重映射 该位由软件置位和清除。 0：TLI_R0没有重映射到PH4 1：TLI_R0重映射到PH4
20	TLI_HSYNC_PI10_REMAP	TLI_HSYNC_PI10重映射 该位由软件置位和清除。 0：TLI_HSYNC没有重映射到PI10 1：TLI_HSYNC重映射到PI10
19	TLI_VSYNC_PI9_REMAP	TLI_VSYNC_PI9重映射 该位由软件置位和清除。 0：TLI_VSYNC没有重映射到PI9 1：TLI_VSYNC重映射到PI9
18	TLI_B7_PI7_REMAP	TLI_B7_PI7重映射 该位由软件置位和清除。 0：TLI_B7没有重映射到PI7 1：TLI_B7重映射到PI7
17	TLI_B6_PI6_REMAP	TLI_B6_PI6重映射 该位由软件置位和清除。 0：TLI_B6没有重映射到PI6

		1：TLI_B6重映射到PI6
16	TLI_B5_PI5_REMAP	TLI_B5_PI5重映射 该位由软件置位和清除。 0：TLI_B5没有重映射到PI5 1：TLI_B5重映射到PI5
15	TLI_B4_PI4_REMAP	TLI_B4_PI4重映射 该位由软件置位和清除。 0：TLI_B4没有重映射到PI4 1：TLI_B4重映射到PI4
14	TLI_G7_PI2_REMAP	TLI_G7_PI2重映射 该位由软件置位和清除。 0：TLI_G7没有重映射到PI2 1：TLI_G7重映射到PI2
13	TLI_G6_PI1_REMAP	TLI_G6_PI1重映射 该位由软件置位和清除。 0：TLI_G6没有重映射到PI1 1：TLI_G6重映射到PI1
12	TLI_G5_PI0_REMAP	TLI_G5_PI0重映射 该位由软件置位和清除。 0：TLI_G5没有重映射到PI0 1：TLI_G5重映射到PI0
11	TLI_G4_PH15_REMAP	TLI_G4_PH15重映射 该位由软件置位和清除。 0：TLI_G4没有重映射到PH15 1：TLI_G4重映射到PH15
10	TLI_G3_PH14_REMAP	TLI_G3_PH14重映射 该位由软件置位和清除。 0：TLI_G3没有重映射到PH14 1：TLI_G3重映射到PH14
9	TLI_G2_PH13_REMAP	TLI_G2_PH13重映射 该位由软件置位和清除。 0：TLI_G2没有重映射到PH13 1：TLI_G2重映射到PH13
8	TLI_R6_PH12_REMAP	TLI_R6_PH12重映射 该位由软件置位和清除。 0：TLI_R6没有重映射到PH12 1：TLI_R6重映射到PH12
7	TLI_R5_PH11_REMAP	TLI_R5_PH11重映射 该位由软件置位和清除。

		0：TLI_R5没有重映射到PH11 1：TLI_R5重映射到PH11
6	TLI_R4_PH10_REMAP	TLI_R4_PH10重映射 该位由软件置位和清除。 0：TLI_R4没有重映射到PH10 1：TLI_R4重映射到PH10
5	TLI_R3_PH9_REMAP	TLI_R3_PH9重映射 该位由软件置位和清除。 0：TLI_R3没有重映射到PH9 1：TLI_R3重映射到PH9
4	TLI_R2_PH8_REMAP	TLI_R2_PH8重映射 该位由软件置位和清除。 0：TLI_R2没有重映射到PH8 1：TLI_R2重映射到PH8
3	TLI_R1_PH3_REMAP	TLI_R1_PH3重映射 该位由软件置位和清除。 0：TLI_R1没有重映射到PH3 1：TLI_R1重映射到PH3
2	TLI_R0_PH2_REMAP	TLI_R0_PH2重映射 该位由软件置位和清除。 0：TLI_R0没有重映射到PH2 1：TLI_R0重映射到PH2
1	TLI_B1_PG12_REMAP	TLI_B1_PG12 重映射 该位由软件置位和清除。 0：TLI_B1没有重映射到PG12 1：TLI_B1重映射到PG12
0	TLI_B4_PG12_REMAP	TLI_B4_PG12重映射 该位由软件置位和清除。 0：TLI_B4没有重映射到PG12 1：TLI_B4重映射到PG12

7.5.18. AFIO 端口配置寄存器 5（AFIO_PCF5）

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

EXMC_SDNE1_REMAP	EXMC_SDNE0_REMAP	EXMC_SDCKE1_REMAP	EXMC_SDCKE0_REMAP	EXMC_SDNWE_REMAP	USART5_RX_REMAP	USART5_TX_REMAP	USART5_CTS_REMAP	USART5_RTS_REMAP	USART5_CK_REMAP	UART6_REMAP	ENET_RX_HI_REMAP	ENET_CRSCOL_REMAP	ENET_TXD01_REMAP	PPS_HI_REMAP	ENET_TXD3_REMAP
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

CAN0_ADD_REMAP	TIMER11_REMAP	UART3_REMAP	SPI1_IO_REMAP[1:0]	SPI1_NSCK_REMAP[1:0]	I2C1_REMAP[1:0]	TIMER7_CH_REMAP	TIMER7_CHON_REMAP[1:0]	TIMER4_REMAP	TIMER1_CH0_REMAP	I2C2_REMAP1	I2C2_REMAP0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	EXMC_SDNE1_REMAP	EXMC_SDNE1重映射 该位由软件置位和清除。 0：没有重映射（PH6） 1：EXMC_SDNE1重映射到PB6
30	EXMC_SDNE0_REMAP	EXMC_SDNE0重映射 该位由软件置位和清除。 0：没有重映射（PH3） 1：EXMC_SDNE0重映射到PC2
29	EXMC_SDCKE1_REMAP	EXMC_SDCKE1重映射 该位由软件置位和清除。 0：没有重映射（PH7） 1：EXMC_SDCKE1重映射到PB5
28	EXMC_SDCKE0_REMAP	EXMC_SDCKE0重映射 该位由软件置位和清除。 0：没有重映射（PH2） 1：EXMC_SDCKE0重映射到PC3
27	EXMC_SDNWE_REMAP	EXMC_SDNWE重映射 该位由软件置位和清除。 0：没有重映射（PH5） 1：EXMC_SDNWE重映射到PC0
26	USART5_RX_REMAP	USART5_RX重映射 该位由软件置位和清除。 0：没有重映射（PC7） 1：USART5_RX重映射到PG9
25	USART5_TX_REMAP	USART5_TX重映射 该位由软件置位和清除。 0：没有重映射（PC6） 1：USART5_TX重映射到PG14
24	USART5_CTS_REM	USART5_CTS重映射

	AP	该位由软件置位和清除。 0：没有重映射（PG15） 1：USART5_CTS重映射到PG13
23	USART5_RTS_REMAP	USART5_RTS重映射 该位由软件置位和清除。 0：没有重映射（PG8） 1：USART5_RTS重映射到PG12
22	USART5_CK_REMAP	USART5_CK重映射 该位由软件置位和清除。 0：没有重映射（PC8） 1：USART5_CK重映射到PG7
21	UART6_REMAP	UART6重映射 该位由软件置位和清除。 0：没有重映射（UART6_RX/UART6_TX映射到PE7/PE8） 1：UART6_RX/UART6_TX重映射到PF6/PF7
20	ENET_RX_HI_REMAP	ETH_RXD2/ETH_RXD3/ETH_RX_ER重映射 该位由软件置位和清除。 0：没有影响，ETH_RXD2/ETH_RXD3参考ENET_REMAP，ETH_RX_ER映射到PB10。 1：ETH_RXD2/ETH_RXD3/ETH_RX_ER重映射到PH6/PH7/PI10
19	ENET_CRSCOL_REMAP	ETH_MII_CRS/ETH_MII_COL重映射 该位由软件置位和清除。 0：没有重映射，ETH_MII_CRS/ETH_MII_COL映射到PA0/PA3 1：ETH_MII_CRS/ETH_MII_COL重映射到PH2/PH3
18	ENET_TXD01_REMAP	ETH_TX_EN/ETH_TXD0/ETH_TXD1重映射 该位由软件置位和清除。 0：没有重映射，ETH_TX_EN/ETH_TXD0/ETH_TXD1映射到PB11/PB12/PB13 1：ETH_TX_EN/ETH_TXD0/ETH_TXD1重映射到PG11/PG13/PG14
17	PPS_HI_REMAP	ETH_PPS_OUT重映射 该位由软件置位和清除。 0：ETH_PPS_OUT没有重映射到PG8 1：ETH_PPS_OUT重映射到PG8
16	ENET_TXD3_REMAP	ETH_TXD3重映射 该位由软件置位和清除。 0：没有重映射，ETH_TXD3映射到PB8 1：ETH_TXD3重映射到PE2
15	CAN0_ADD_REMAP	CAN0附加重映射 该位由软件置位和清除。

		0：没有影响，参考CAN0_REMAP 1：CAN0_TX/CAN0_RX重映射到PH13/PI9
14	TIMER11_REMAP	TIMER11重映射 该位由软件置位和清除。 0：没有重映射，TIMER11_CH0/TIMER11_CH1映射到PB14/PB15 1：TIMER11_CH0/TIMER11_CH1重映射到PH6/PH9
13	UART3_REMAP	UART3重映射 该位由软件置位和清除。 0：没有重映射，UART3_TX/UART3_RX映射到PC10/PC11 1：UART3_TX/UART3_RX重映射到PA0/PA1
12:11	SPI1_IO_REMAP[1:0]	SPI1_MISO/SPI1_MOSI重映射 该位由软件置位和清除。 00/01：没有重映射，SPI1_MISO/SPI1_MOSI映射到PB14/PB15 10：SPI1_MISO/SPI1_MOSI重映射到PI2/PI3 11：SPI1_MISO/SPI1_MOSI重映射到PC2/PC3
10:9	SPI1_NSCK_REMAP[1:0]	SPI1_NSS/SPI1_SCK重映射 该位由软件置位和清除。 00/01：没有重映射，SPI1_NSS/SPI1_SCK映射到PB12/PB13 10：SPI1_NSS/SPI1_SCK重映射到PI0/PI1 11：SPI1_NSS/SPI1_SCK重映射到PB9/PB10
8:7	I2C1_REMAP[1:0]	I2C1重映射 该位由软件置位和清除。 00/01：没有重映射，I2C1_SCL/I2C1_SDA/I2C1_SMBA映射到PB10/PB11/PB12 10：I2C1_SCL/I2C1_SDA/I2C1_SMBA重映射到PH4/PH5/PH6 11：I2C1_SCL/I2C1_SDA/I2C1_SMBA重映射到PF0/PF1/PF2
6	TIMER7_CH_REMAP	TIMER7_CH0/TIMER7_CH1/TIMER7_CH2/TIMER7_CH3/TIMER7_ETI/TIMER7_BKIN重映射 该位由软件置位和清除。 0：没有重映射，TIMER7_CH0/TIMER7_CH1/TIMER7_CH2/ TIMER7_CH3/ TIMER7_ ETI / TIMER7_BKIN 映射到PC6/PC7/PC8/PC9/PA0/PA6 1：TIMER7_CH0/ TIMER7_CH1/ TIMER7_CH2 / TIMER7_CH3/ TIMER7_ ETI / TIMER7_BKIN重映射到PI5/PI6/PI7/PI2/PI3/PI4
5:4	TIMER7_CHON_REMAP[1:0]	TIMER7_CH0N/TIMER7_CH1N/TIMER7_CH2N重映射 该位由软件置位和清除。 00/01：没有重映射，TIMER7_CH0_ON / TIMER7_CH1_ON / TIMER7_CH2_ON映射到PA7/PB0/PB1 10：TIMER7_CH0_ON / TIMER7_CH1_ON / TIMER7_CH2_ON重映射到PA5/PB14/PB15

		11：TIMER7_CH0_ON / TIMER7_CH1_ON / TIMER7_CH2_ON重映射到PH13/PH14/PH15
3	TIMER4_REMAP	TIMER4 重映射 该位由软件置位和清除。 0：没有重映射，TIMER4_CH0/ TIMER4_CH1/ TIMER4_CH2/ TIMER4_CH3映射到PA0/PA1/PA2/PA3 1：TIMER4_CH0/ TIMER4_CH1/ TIMER4_CH2/ TIMER4_CH3重映射到PH10/PH11/PH12/PI0
2	TIMER1_CH0_REMAP	TIMER1_CH0重映射 该位由软件置位和清除。 0：没有影响，参考TIMER1_ REMAP 1：TIMER1_CH0/ TIMER1_ETR重映射到PA5
1	I2C2_REMAP1	I2C2 重映射1 该位由软件置位和清除。 0：没有重映射 1：I2C2_SCL/I2C2_SDA/I2C2_SMBA重映射到PH7/PH8/PH9
0	I2C2_REMAP0	I2C2 重映射0 该位由软件置位和清除。 0：没有重映射 1：I2C2_SCL/I2C2_SDA/I2C2_SMBA重映射到PA8/ PC9/ PA9

8. 循环冗余校验管理单元（CRC）

8.1. 简介

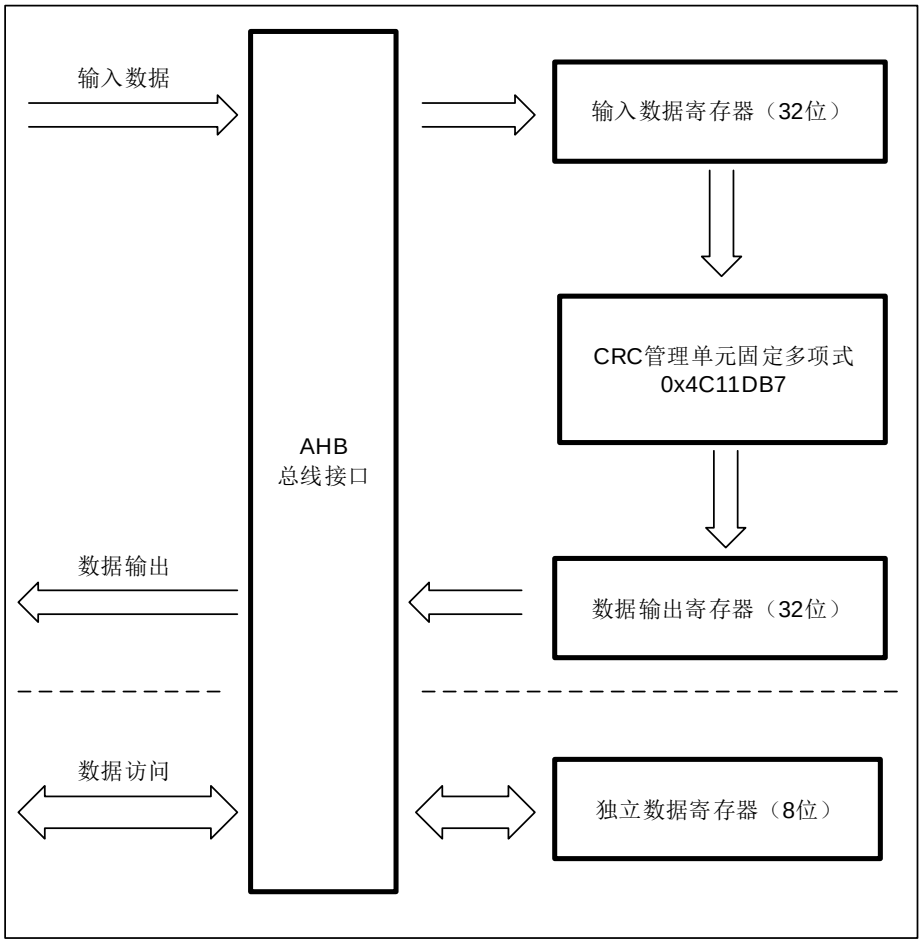
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 管理单元能用固定的多项式来计算 32 位的 CRC 校验码。

8.2. 主要特征

- 32位数据输入/输出寄存器。从输入数据到得出计算结果，对于32位输入数据，计算周期为4个AHB时钟周期。
- 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。
- 固定的计算多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$
  该32位CRC多项式是一个与以太网计算多项式相同的多项式。

图 8-1. CRC 管理单元框图



8.3. 功能说明

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

  如果不通过软件设置CRC_CTL 寄存器的方式来清除CRC_DATA寄存器，新输入的原始数据将会基于前一次CRC_DATA寄存器中的结果进行计算。

  对于32位的数据长度的CRC计算，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。

8.4. CRC 寄存器

CRC 基地址：0x4002 3000

8.4.1. 数据寄存器（CRC_DATA）

地址偏移：0x00
复位值：0xFFFF FFFF

该寄存器只能按字(32位）访问

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

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

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

8.4.2. 独立数据寄存器（CRC_FDATA）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位）访问

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

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

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

8.4.3. 控制寄存器 (CRC_CTL)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	RST	该位用来复位CRC_DATA寄存器，设置其值为0xFFFFFFFF，然后该位被硬件自动清零。该位对CRC_FDATA寄存器没有影响。 软件可读可写

9. 真随机数生成器（TRNG）

9.1. 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。

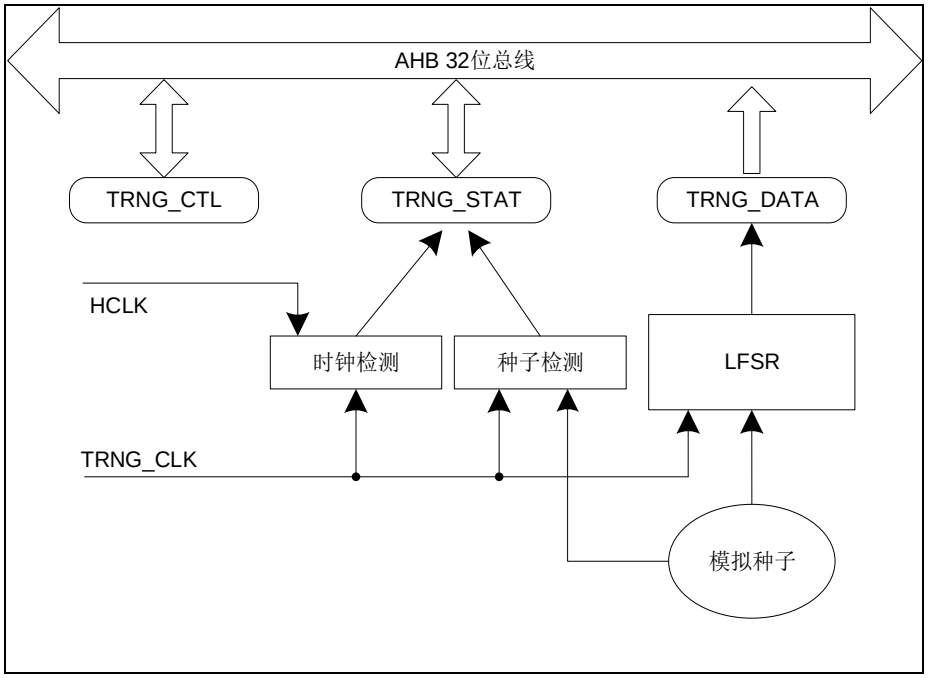
GD32F207xx系列微控制器具有TRNG外设，而GD32F205xx系列微控制器不具有TRNG外设。

9.2. 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；
- 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。

9.3. 功能说明

图 9-1. TRNG 模块框图



随机数种子来自于模拟电路。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*复位和时钟单元（RCU）*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位被置1将产生

中断。

9.3.1. 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）当有中断产生，检测TRGN_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。每一个连续的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。

9.3.2. 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。时钟错误对产生的上一个随机数没有影响。

种子错误

当模拟种子的值在64个TRNG_CLK时钟周期间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。

9.4. TRNG 寄存器

TRNG 基地址：0x5006 0800

9.4.1. 控制寄存器（TRNG_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:4	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。

9.4.2. 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。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									SEIF	CEIF	保留		SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

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，说明已检测到种子错误但已恢复正常。 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

9.4.3. 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

10. 加密处理器（CAU）

10.1. 简介

加密处理单元支持处理DES，三重DES或AES（128，192或256）算法，对数据进行加密或解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物“FIPS PUB 46-3，1999年10月25日”规定的数据加密标准（DES）和三重DES（TDES）。它遵循美国国家标准协会（ANSI）X9.52标准；
- 联邦信息处理标准出版物（FIPS PUB 197，2001年11月26日）规定的高级加密标准（AES）。

CAU 处理器可在多种模式下使用 DES/三重 DES/多种长度密钥的 AES 算法执行数据加密和解密。

CAU外设为32位AHB外设，它支持对输入FIFO和输出FIFO的DMA传输。

GD32F207xx微控制器具有CAU外设，而GD32F205xx微控制器不具有CAU外设。

10.2. 主要特征

- 支持DES，三重DES和AES加密解密算法
- 支持DES，三重DES和AES下的多种模式，包括电子密码本（ECB）、加密分组链接（CBC）模式或计数器模式（CTR）
- 输入与输出FIFO支持DMA传输

DES/三重DES

- 支持电子密码本（ECB）或加密分组链接（CBC）模式
- 支持在CBC模式下使用2×32位初始化向量（IV）
- 输入FIFO和输出FIFO可存储8x32位数据
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输

AES

- 支持支持电子密码本（ECB）、加密分组链接（CBC）模式或计数器模式（CTR）
- 支持128位、192位或256位密钥
- 支持在CBC、CTR模式下使用4×32位初始化向量(IV)
- 输入和输出FIFO各8字深
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输

10.3. CAU 数据类型和初始化向量

10.3.1. 数据类型

CAU处理器一次输入32 位（字）数据，DES每64位对数据流进行处理，AES每128位对数据流进行处理。对于每个数据块，在其进入CAU处理器之前，可对这些数据执行位、字节、半字交换或不交换操作（取决于要加密的数据类型）。在CAU数据写入OUT FIFO之前，需要对其执行同样的交换操作。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。

*图10-1. DATAM不交换/半字交换*和*图10-2. DATATM字节交换/位交换*介绍了128位AES块在不同数据类型下的数据交换。（对于DES，数据块大小为2个32位字，请参考图中前两个字的数据交换）。

图 10-1. DATAM 不交换/半字交换

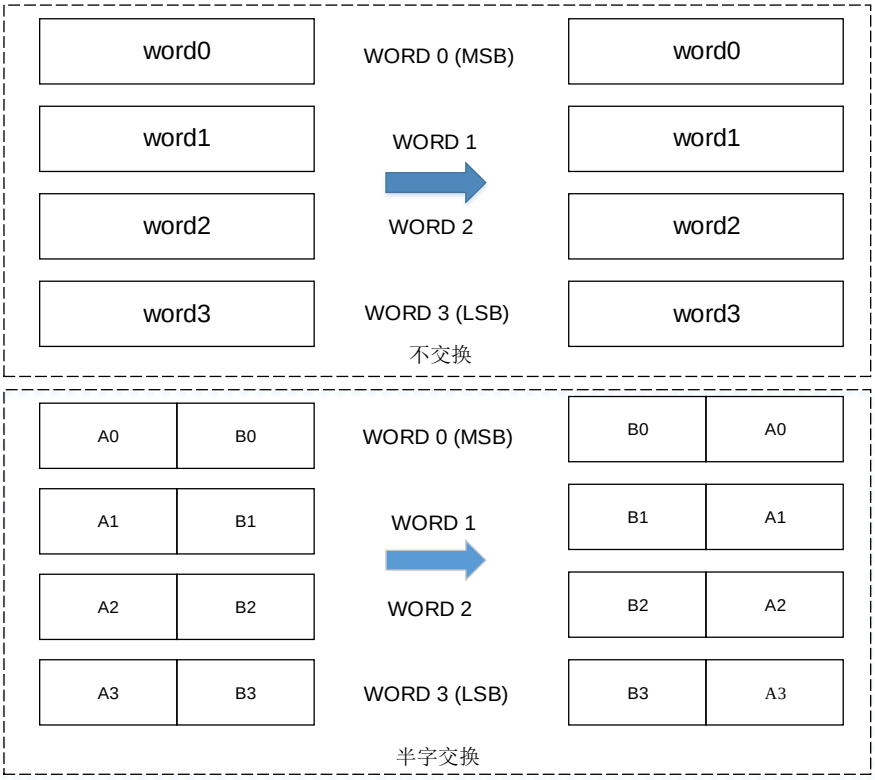
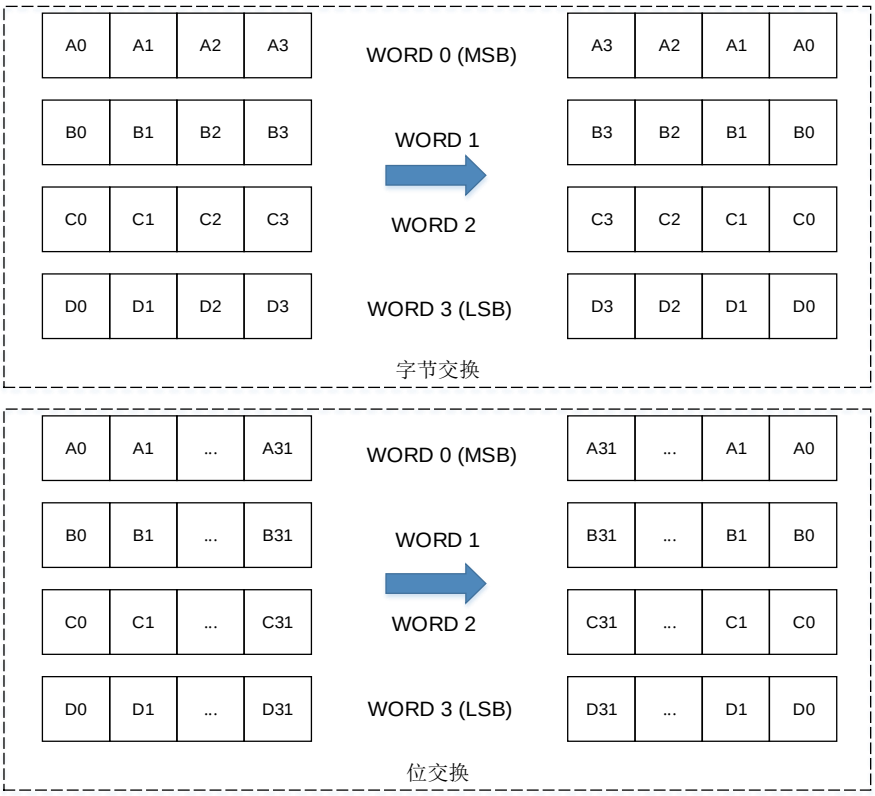


图 10-2. DATATM 字节交换/位交换



10.3.2. 初始化向量

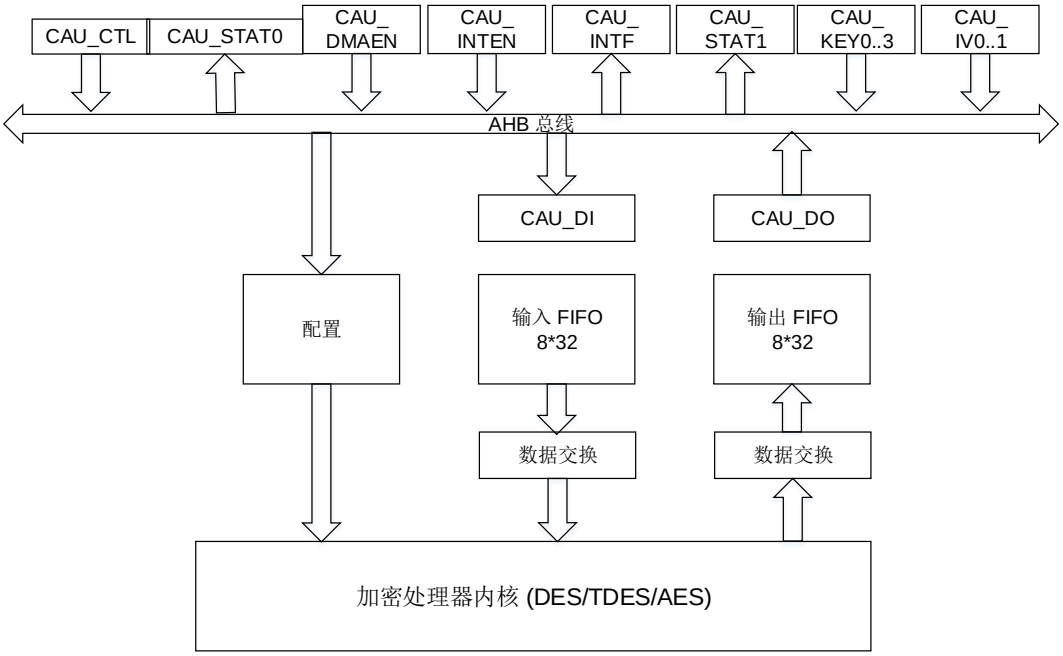
初始化向量用于在CBC和CTR模式下与数据块进行异或。初始化向量与明文或密码数据无关，而且它们不受DATAM值的影响。注意初始化向量寄存器CAU_IV0..1（H/L）只有在BUSY位（CAU_STAT0寄存器位4）为0时才能被修改，否则写操作都是无效的。

10.4. 加密处理器流程

加密处理器关于DES和AES加密处理的实现具体请参考章节***DES/TDES加密处理流程***和***AES加密处理流程***。

图10-3. CAU框图为加密处理器的模块框图。

图 10-3. CAU 框图



10.4.1. DES/TDES 加密处理流程

DES/三重DES加密处理器由DES算法（DEA），密钥（DES算法使用1个密钥，TDES算法使用3个密钥），以及在CBC模式下使用的初始化向量组成。

DES/TDES 密钥

DES模式密钥为[KEY1]，TDES模式密钥为[KEY3 KEY2 KEY1]。当配置使用TDES算法，支持以下三种密钥选项：

■ 三个相同密钥

三个密钥KEY3、KEY2和KEY1是相同的，即KEY3=KEY2=KEY1。该选项详见FIPS PUB 46-3-1999（以及ANSI X9.52-1998）。这种模式下实际上与DES是等同的。

■ 两个独立密钥

这个选项中，KEY2与KEY1不同，KEY3与KEY1相同，即KEY1与KEY2独立，而KEY3=KEY1。该选项详见FIPS PUB 46-3-1999（以及ANSI X9.52-1998）。

■ 三个独立密钥

这个选项中，KEY1，KEY2与KEY3都是独立的。详见FIPS PUB 46-3 -1999（以及ANSI X9.52-1998）。

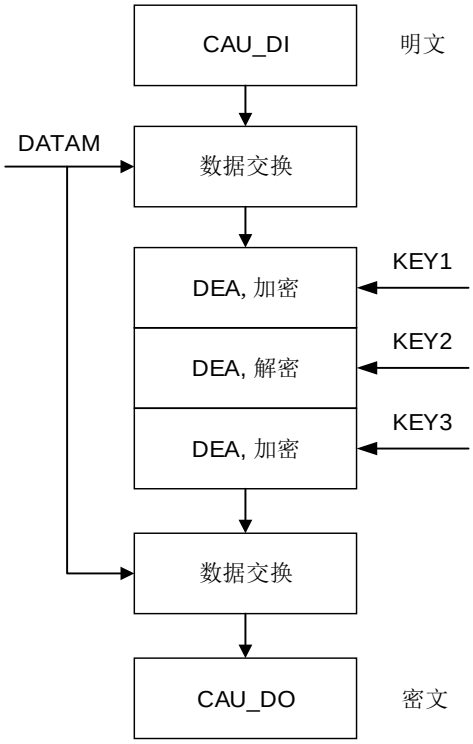
FIPS PUB 46-3（以及ANSI X9.52-1998）对DES/TDES中密钥的使用进行了详尽的解释，在本手册中不进行赘述。

DES/TDES 电子密码本（ECB）加密

64位输入明文数据首先经过根据数据类型值进行数据交换后作为输入数据块。若配置使用的是TDES算法，则输入数据块通过DEA使用KEY1进行加密处理。处理结果输出直接反馈到DEA，

使用KEY2进行解密处理。之后处理结果输出直接反馈到最后的DEA，使用KEY3进行加密处理。上述的处理过程的输出需要再次根据数据类型值进行数据交换，生成一个64位密文输出数据块。若配置使用的是DES算法，在通过DEA使用KEY1进行加密处理后的结果直接根据数据类型值进行数据交换，生成一个64位密文输出数据块。DES/TDES电子密码本加密流程图见*图10-4. DES/TDES ECB加密*。

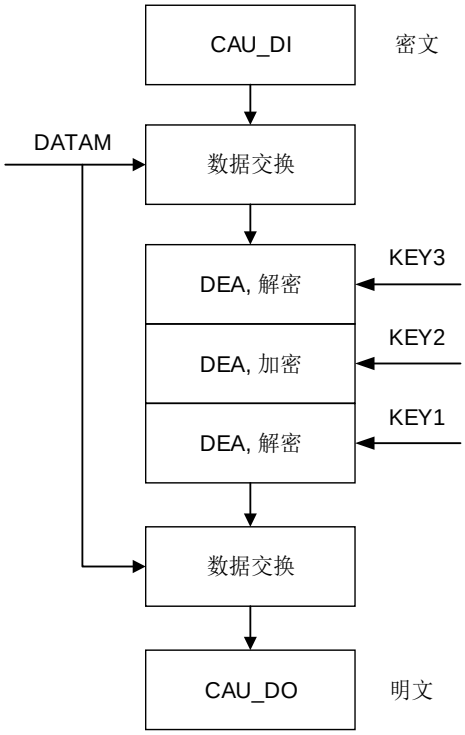
图 10-4. DES/TDES ECB 加密



DES/TDES 电子密码本（ECB）解密

根据数据类型进行数据交换后，首先得到64位的输入密文。若配置使用的是TDES算法，将在DEA中读取输入数据块并使用KEY3进行解密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行加密处理。之后处理结果输出直接反馈到最后的DEA，使用KEY1进行解密处理。上述的处理过程的输出需要再次根据数据类型进行数据交换，生成一个64位明文输出数据块。若配置使用的是DES算法，在通过DEA使用KEY1进行解密处理后的结果直接根据数据类型值进行数据交换，生成一个64位明文输出数据块。DES/TDES电子密码本解密流程图见*图10-5. DES/TDES ECB解密*。

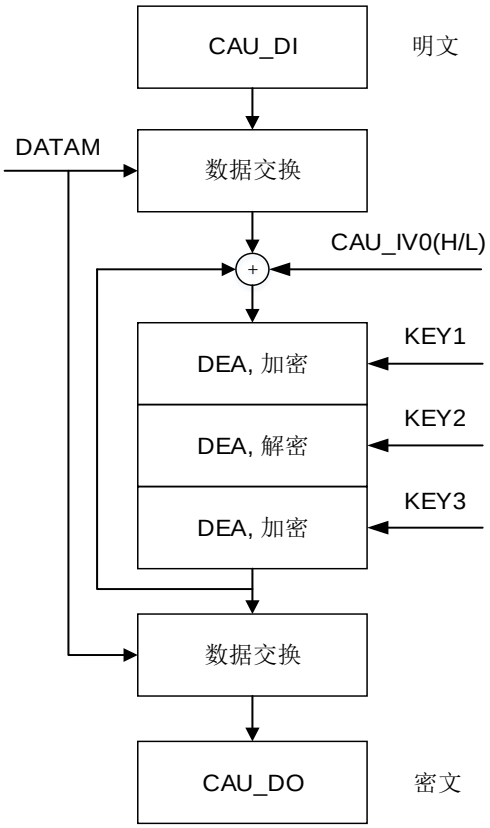
图 10-5. DES/TDES ECB 解密



DES/TDES 加密分组链接（CBC）加密

CBC模式下DEA块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。若配置使用的是TDES算法，第一个数据交换后的输入明文数据块与64位初始化向量CAU_IV0..1进行异或运算，结果在DEA中读取并使用KEY1进行加密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行解密处理。之后处理结果输出直接反馈到最后的DEA，使用KEY3进行加密处理。上述的处理过程的输出作为下一个初始化向量，并与下一个明文数据块进行异或运算，进行下一轮的加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成密文输出数据块。若配置使用的是DES算法，则在上述的步骤操作中忽略第二次和第三次DEA的运算处理。DES/TDES加密分组链接加密流程图见*图10-6. DES/TDES CBC加密*。

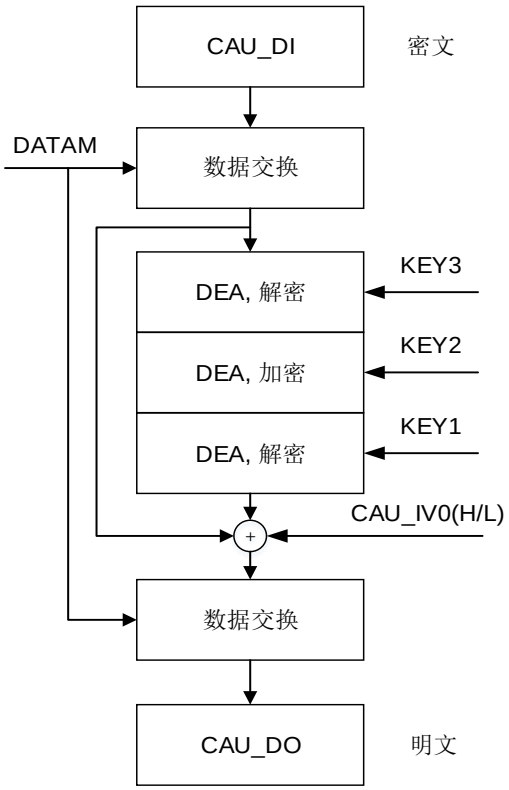
图 10-6. DES/TDES CBC 加密



DES/TDES 密码块链接（CBC）解密

使用DES/TDES CBC模式解密，若配置使用的是TDES算法，第一个数据交换后的输入密文数据块，通过DEA读取并使用KEY3进行解密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行加密处理。之后处理结果输出直接反馈到最后的DEA，使用KEY1进行解密处理。上述的处理过程的输出再与64位初始化向量CAU_IV0..1进行异或运算。之后，第一个输入密文数据块作为下一个初始化向量，并与后续的DEA解密处理后的输出结果进行异或运算。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。若配置使用的是DES算法，则在上述的步骤操作中忽略第二次和第三次DEA的运算处理。DES/TDES加密分组链接解密流程图见*图 10-7. DES/TDES CBC解密*。

图 10-7. DES/TDES CBC 解密



10.4.2. AES 加密处理流程

AES加密处理器由AES算法（AEA），多个密钥，以及初始化向量或随机数三部分组成。

AES支持三种长度的密钥：128、192和256位密钥，根据操作模式的不同使用不同数目的初始化向量或随机数。

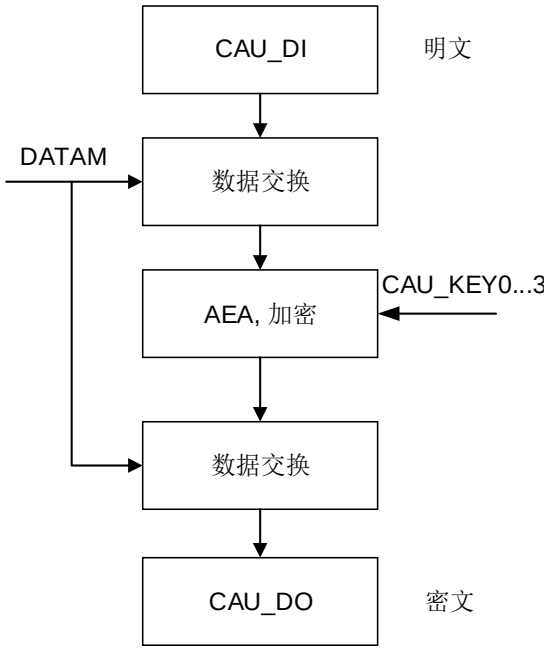
使用128位密钥时AES密钥为[KEY3 KEY2]，使用192位密钥时AES密钥为[KEY3 KEY2 KEY1]，使用256位密钥时AES密钥为[KEY3 KEY2 KEY1 KEY0]。

FIPS PUB 197（2001年11月26日）中对AES中使用的密钥进行了详细的解释，本手册不再进行赘述。

AES 电子密码本（ECB）加密

根据数据类型进行数据交换后，首先得到128位输入明文数据块。输入数据块通过AEA使用128位，或192位，或256位密钥进行加密处理。处理结果再根据数据类型进行数据交换，生成一个128位密文输出数据块，并存储在输出FIFO中。AES电子密码本加密流程图见*图10-8. AES ECB加密*。

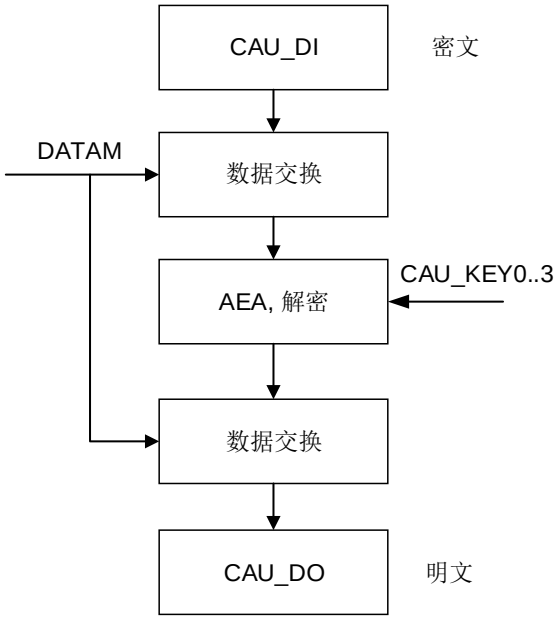
图 10-8. AES ECB 加密



AES 电子密码本（ECB）解密

首先需要准备密钥，以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换得到128位输入密文数据块。输入数据块在AEA中读取并使用上面准备的密钥进行解密处理。处理结果输出再根据数据类型值进行数据交换，生成一个128位明文输出数据块。AES电子密码本解密流程图见*图10-9. AES ECB解密*。

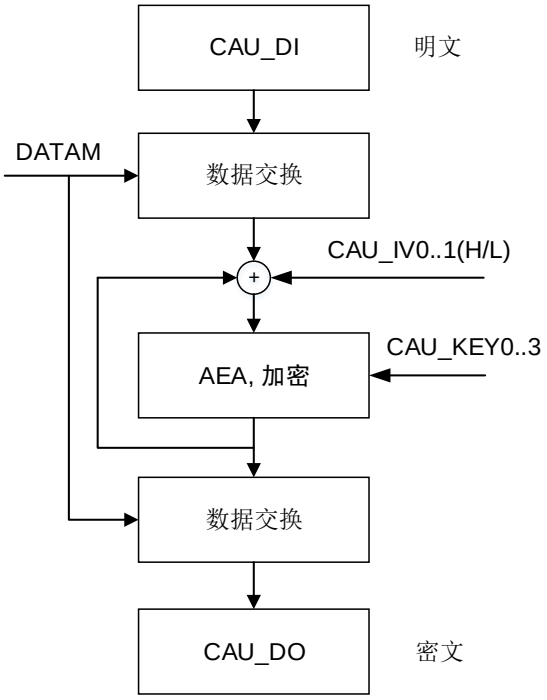
图 10-9. AES ECB 解密



AES 加密分组链接（CBC）加密

CBC模式下AEA块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。数据交换后的输入明文数据块与128位初始化向量CAU_IV0..1进行异或运算，结果再通过AEA使用128位，或192位，或256位密钥进行加密处理。处理结果作为下一个初始化向量，并与下一个输入明文数据块进行异或运算，进行下一轮加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型值进行数据交换，生成密文输出数据块。AES加密分组链接加密流程图见*图10-10. AES CBC加密*。

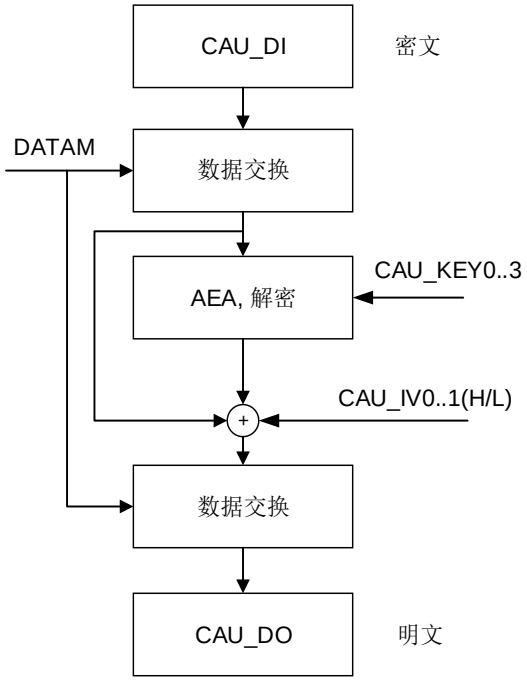
图 10-10. AES CBC 加密



AES 加密分组链接（CBC）解密

与AES电子密码本（ECB）模式解密类似，首先需要准备密钥以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换，得到128位输入密文数据块，输入数据块在AEA中读取并使用准备的密钥进行解密处理。之后，第一个输入密文数据块作为下一个初始化向量，并与下一个AEA解密处理结果进行异或运算（第一次的初始化向量为输入CAU_IV0..1的初值）。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。AES加密分组链接解密流程图见*图10-11. AES CBC解密*。

图 10-11. AES CBC 解密

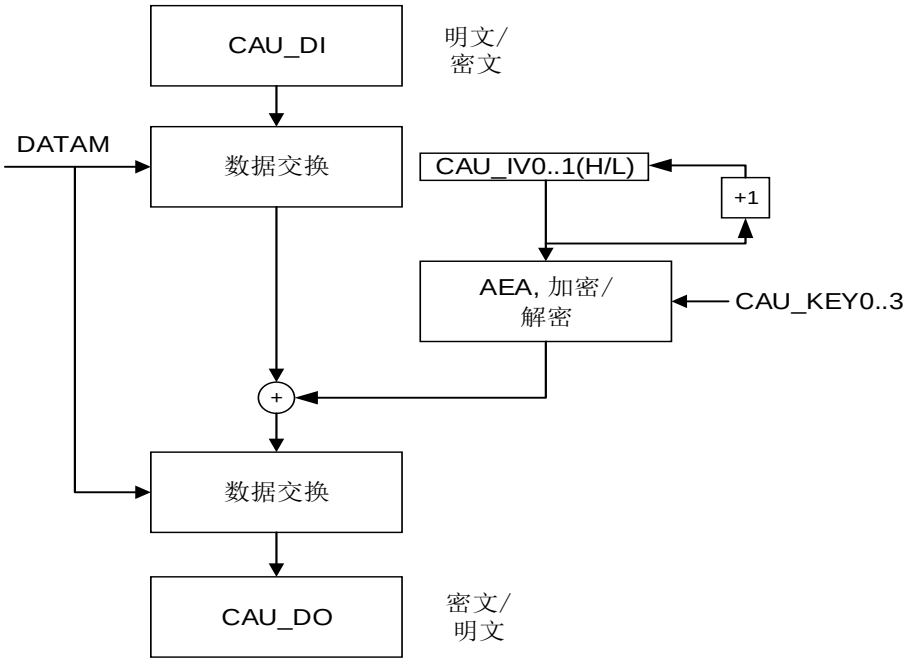


AES 计数器（CTR）模式

在计数器模式下，随机数与计数器的组合会作为AEA计算单元的输入来进行运算，运算结果会与输入的明文或密文进行异或，来求得最终加密或者解密的结果。由于加密和解密处理的计数器值是由相同的初始值进行递增的，因此加密和解密处理用的密钥序列是相同的。解密处理的操作与加密操作的流程完全相同。128位初始化向量的低32位表示为计数器值，这意味着其余96位在操作过程中保持不变，并且计数器的初始值应当设置为1。随机数是一个32位一次性值，应当更新到每个通信块。64位的初始化向量应确保每个给定值只用于一个给定密钥。计数器块框图结构见*图10-12. 计数器块结构*，AES计数器加密/解密流程图见*图10-13. AES CTR加密/解密*。

图 10-12. 计数器块结构

图 10-13. AES CTR 加密/解密



10.5. 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. 根据算法配置CAU_KEY0..3(H/L)寄存器。
4. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型。
5. 设置CAU_CTL寄存器的ALGM位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR）。
6. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作。
7. 设置CAU_IV0..1(H/L)寄存器，配置初始化向量。
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. 根据算法配置CAU_KEY0..3（H/L）寄存器。
4. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型。

5. 设置CAU_CTL寄存器的ALGM位为“111”，配置准备密钥用于解密。
6. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU。
7. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好。
8. 设置CAU_CTL寄存器的ALGM位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR）。
9. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作。
10. 设置CAU_IV0..1（H/L）寄存器，配置初始化向量。
11. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO。
12. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU。
13. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输。
14. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输。
15. 重复步骤13和步骤14，直到所有的数据块都完成解密。

10.6. 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。

支持单次传输和突发传输，以确保传输长度不为突发长度的整数倍的情况下能正确完成数据传输。注意DMA控制器需要配置突发长度为4个字或更小，以确保不会丢失数据。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。

10.7. 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。

10.8. CAU 挂起模式

当CAU中待处理的新的数据块优先级高于正在处理的数据块，则正在处理的数据块可能被挂起。按照下列的步骤来完成被挂起数据块的加密/解密处理。

当使用 DMA 进行数据传输：

1. 停止当前输入数据传输。将CAU_DMAEN寄存器的DMAIEN位清零。
2. 若为DES或AES算法，则需等待直到输入和输出FIFO均为空，如果检查到输入FIFO不为空即IEM位为0，则写入一个字的数据，再检查IEM位，直到IEM位为1，则停止写入数据，再等待BUSY位为0，以确保下一个数据块不会被上一个数据块影响。若为TDES算法，则与AES算法相似，但不需要等待输入FIFO为空。
3. 将CAU_DMAEN寄存器中的DMAOEN位清零，停止输出数据传输。并将CAU_CTL寄存器中的CAUEN位清零，禁用CAU。
4. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，以及密钥值。若为CBC或CTR模式，则还需要保存初始化向量。
5. 配置并处理新数据块。
6. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量。再将CAU_CTL寄存器的CAUEN位置位以使能CAU。

当使用 CPU 来传输数据到 CAU_DI 和 CAU_DO：

1. 当使用CPU来进行数据传输，则需要等待第四次读CAU_DO寄存器，并在写CAU_DI之前，以确保一个数据块处理结束的时候再挂起消息处理。
2. 将CAU_CTL寄存器的CAUEN位清零，禁用CAU。
3. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，以及密钥值。若为CBC或CTR模式，则还需要保存初始化向量。
4. 配置并处理新数据块。
5. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量。再将CAU_CTL寄存器的CAUEN位置位以使能CAU。

10.9. CAU 寄存器

CAU 基地址：0x5006 0000

10.9.1. CAU 控制寄存器（CAU_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUEN	FFLUSH	保留				KEYM[1:0]		DATAM[1:0]		ALGM[2:0]			CAUDIR	保留	
rw	w					rw		rw		rw			rw		

位/位域	名称	描述
31:16	保留	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，必须在BUSY=0时才可配置 000：TDES-ECB（三重DES电子密码本），使用CAU_KEY1，2，3.

		不使用初始化向量（CAU_IV0..1） 001：TDES-CBC（三重DES加密分组链接），使用CAU_KEY1，2， 3. 使用初始化向量（CAU_IV0）与数据块进行异或 010：DES-ECB（DES电子密码本），仅使用CAU_KEY1 不使用初始化向量（CAU_IV0..1） 011：DES-CBC（DES加密分组链接），仅使用CAU_KEY1 使用初始化向量（CAU_IV0）与数据块进行异或 100：AES-ECB（AES电子密码本），使用CAU_KEY0, 1, 2, 3. 不使用初始化向量（CAU_IV0..1） 101：AES-CBC（AES加密分组链接），使用CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 110：AES-CTR（AES计数器模式），使用CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 该模式下，加密与解密处理相同，忽略CAUDIR位 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。
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。

10.9.2. CAU 状态寄存器 0 (CAU_STAT0)

地址偏移：0x04

复位值：0x0000 0003

该寄存器只能按字（32 位）访问。

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

10.9.3. CAU 数据输入寄存器 (CAU_DI)

地址偏移：0x08
复位值：0x0000 0000

数据输入寄存器用于传输明文或密文数据块到输入FIFO中进行处理。首先写入FIFO的是数据块的MSB，最后才是LSB。当CAUEN位为0，并且输入FIFO非空时，读取该寄存器时返回FIFO中的首个字。当CAUEN位为1时，读取该寄存器返回一个不确定的值。一旦执行了读操作，则必须要刷新FIFO以处理新数据块。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	DI[31:0]	数据输入 写这些位，数据会写入输入FIFO。当CAUEN位为0时，读这些位将返回输入FIFO中的值，否则将返回不确定的值。

10.9.4. CAU 数据输出寄存器 (CAU_DO)

地址偏移：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。

10.9.5. CAU DMA 使能寄存器 (CAU_DMAEN)

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														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

10.9.6. CAU 中断使能寄存器 (CAU_INTEN)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

10.9.7. CAU 状态寄存器 1 (CAU_STAT1)

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															

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

10.9.8. CAU 中断标志寄存器 (CAU_INTF)

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														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

10.9.9. CAU 密钥寄存器 (CAU_KEY0..3(H/L))

地址偏移：0x20~0x3C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问，必须在 BUSY 位为 0 时写这些寄存器。
在DES模式下，仅使用CAU_KEY1。

在TDES模式下，使用CAU_KEY1，CAU_KEY2和CAU_KEY3。

在AES-128模式下，KEY2H[31:0]和KEY2L[31:0]分别对应于AES_KEY[0:63]的高32位与低32位，而KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[64:127]的高32位与低32位。

在AES-192模式下，KEY1H[31:0]和KEY1L[31:0]分别对应于AES_KEY[0:63]的高32位与低32位，KEY2H[31:0]和KEY2L[31:0]分别对应于AES_KEY[64:127]的高32位与低32位，KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[128:191]的高32位与低32位。

在AES-256模式下，KEY0H[31:0]和KEY0L[31:0]分别对应于AES_KEY[0:63]的高32位与低32位，KEY1H[31:0]和KEY1L[31:0]分别对应于AES_KEY[64:127]的高32位与低32位，KEY2H[31:0]和KEY2L[31:0]分别对应于AES_KEY[128:191]的高32位与低32位，KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[192:255]的高32位与低32位。

CAU_KEY0H

地址偏移：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

地址偏移：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

地址偏移：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

地址偏移：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

地址偏移：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

地址偏移：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

地址偏移：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

地址偏移：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

10.9.10. CAU 初始化向量寄存器 (CAU_IV0..1(H/L))

地址偏移：0x40~0x4C
复位值：0x0000 0000

该寄存器只能按字（32位）访问，必须在BUSY位为0时写这些寄存器。

在DES/TDES模式下，IV0H和IV0L分别对应于初始化向量的高32位和低32位。

在AES模式下，IV0H和IV1H分别对应于128位初始化向量的最高32位和最低32位。

CAU_IV0H

地址偏移：0x40
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															
rw															

CAU_IV0L

地址偏移：0x44
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															
rw															

CAU_IV1H

地址偏移：0x48
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															
rw															

CAU_IV1L

地址偏移：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															
rw															

位/位域	名称	描述
31:0	IV0...1(H/L)	用于DES或TDES或AES的初始化向量

11. 哈希处理器（HAU）

11.1. 简介

哈希处理器应用于信息安全。支持应用于多种场合的安全哈希算法（SHA-1，SHA-224和SHA-256），消息摘要算法（MD5）和哈希运算消息认证码（HAMC）。对长达（2^{64} - 1）位的消息，哈希处理器计算消息摘要长度对应于SHA-1，SHA-224，SHA-256，和MD5算法分别为160位，224位，256位，128位。而在HMAC算法中，SHA-1、SHA-224、SHA-256或MD5将作为哈希函数被调用两次，来产生验证消息。

哈希处理器完全兼容下列标准：

- 联邦信息处理标准出版物180-2（FIPS PUB 180-2）；
- 安全散列标准规范（SHA-1，SHA-224，SHA256）；
- 互联网工程任务组征求意见文档编号 1321（IETF RFC 1321）规范（MD5）。

GD32F207xx 系列微控制器具有 HAU 外设，而 GD32F205xx 系列微控制器不具有系列 HAU 外设。

11.2. 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。

11.3. 数据类型

哈希处理器每次接收32位字，但每次计算处理一个512位块。对每个输入字，在送入哈希内核之前都会根据数据类型进行位/字节/半字/不交换。同样在数据输出之前也要进行相同的数据交换。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。SHA-1，SHA-224，SHA-256的计算均为大端模式。

*图11-1. DATAM 不交换/半字交换*和*图11-2. 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 11-1. DATAM 不交换/半字交换

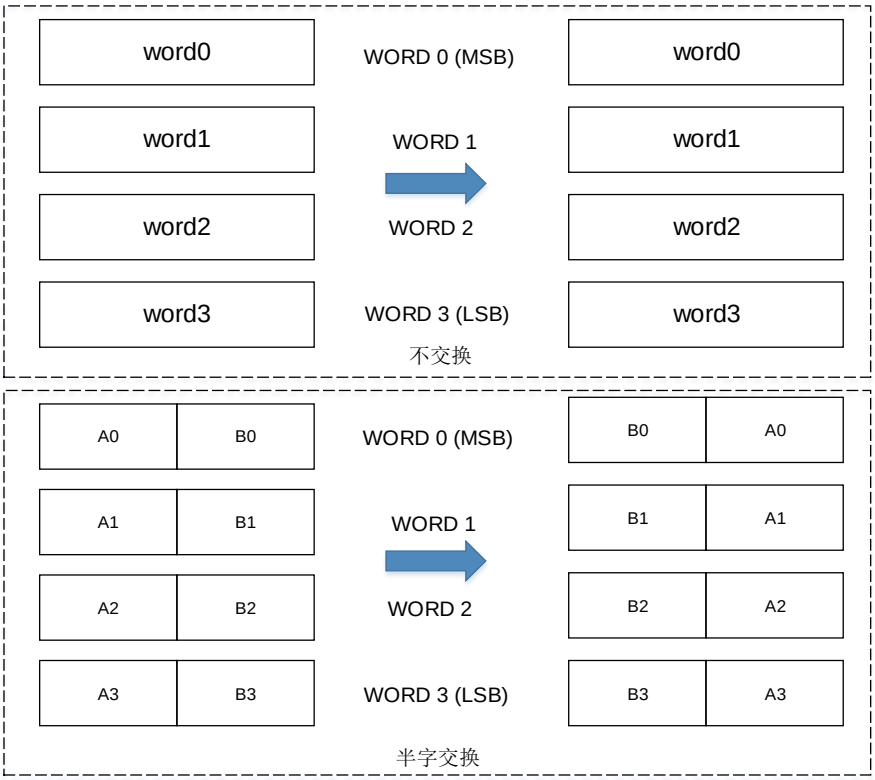
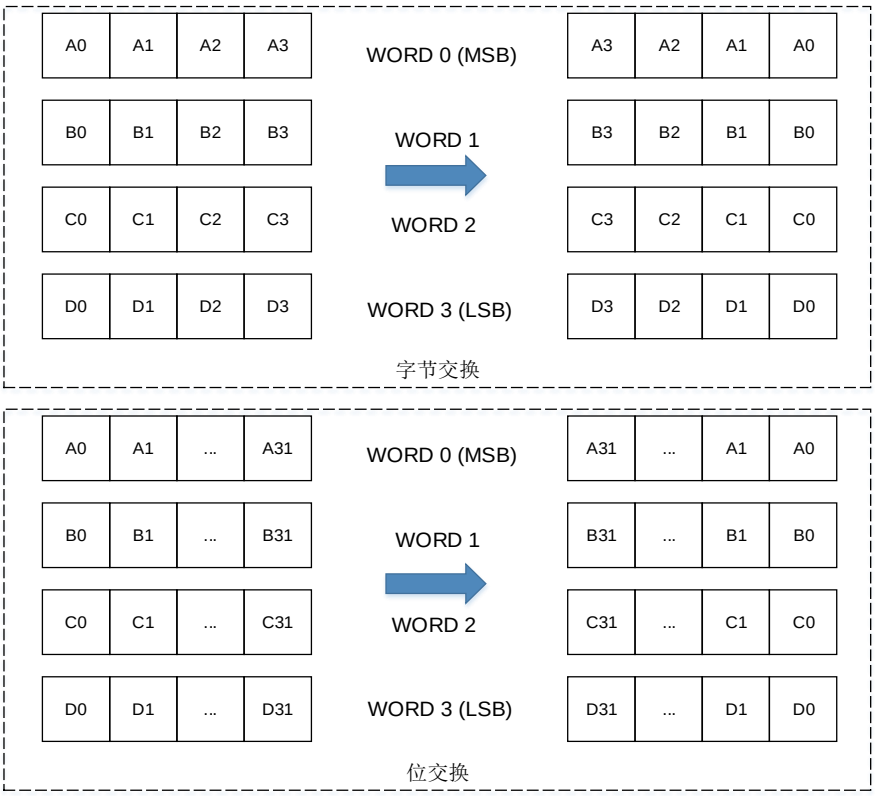


图 11-2. DATAM 字节交换/位交换

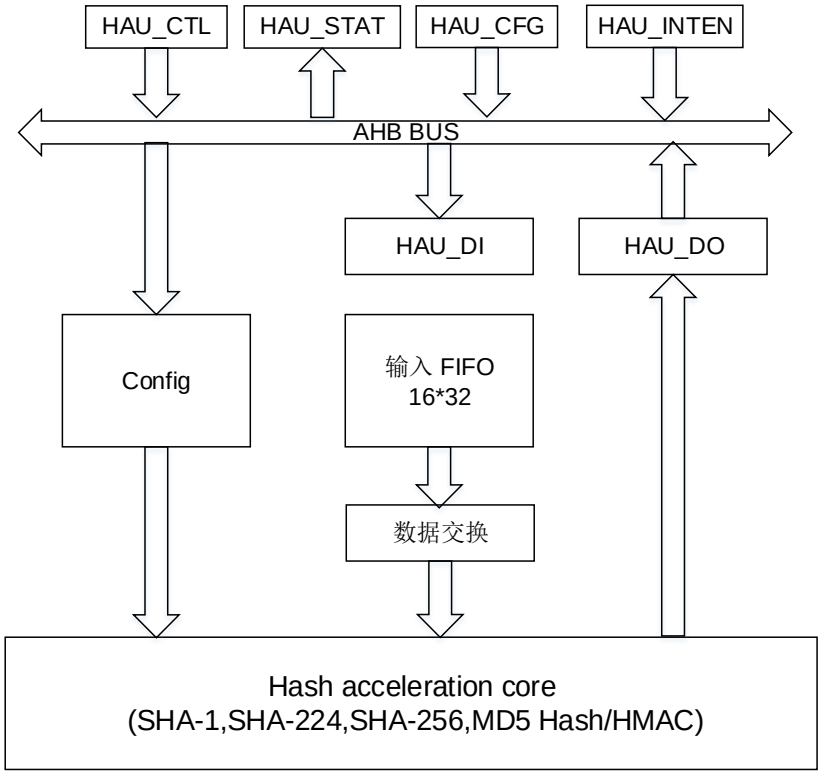


11.4. HAU 内核

哈希处理器使用安全哈希算法对输入消息进行信息压缩计算。对长度为（2^{64} - 1）位的消息摘要计算结果的长度对应于SHA-1，SHA-224，SHA-256，和MD5算法分别为160位，224位，256位，128位。哈希处理器可用于生成和验证消息签名，并由于摘要远远小于消息的大小而具有更高的效率。

要由哈希处理器处理的消息应视为位串。消息长度为消息的位数。哈希处理器可以确保信息的安全，因为根据某个给定消息摘要来寻找原对应的消息在计算层面是无法实现的，而在原输入消息上任何的改动都将导致生成完全不同的消息摘要。

图 11-3. HAU 结构框图



11.4.1. 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

11.4.2. 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。

使用 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。

使用 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。

11.4.3. 哈希模式

将HAU_CTL寄存器的HMS位设为0，选择为哈希模式。则当HAU_CTL寄存器的START位为1时，将根据ALGM位域的配置选择SHA-1，SHA-224，SHA-256或MD5算法进行计算。

当从HAU_DI寄存器和接收FIFO中接收到512位的消息块时，将根据DMA和CALEN位状态开始摘要的计算。

最终的计算结果可以从HAU_DO0...7寄存器中读取。

11.4.4. HMAC 模式

HMAC模式根据用户所选的密钥来进行消息验证。更多关于HMAC规范的信息请参阅“HMAC：密钥散列消息认证，H. Krawczyk，M. Bellare，R. Canetti，1997年2月”。

HMAC算法表示如下：

HMAC(input) = HASH[((key | opad) XOR 0x5c) | HASH(((key | ipad) XOR 0x36) | input)]

其中“ipad”和“opad”用于将密钥用数个“0”进行填充扩展到512位，“|”为连接符。

HMAC模式需要四个不同阶段：

1. 将HAU_CTL寄存器的HMS位置1，并根据期望的算法设置ALGM位域。若密钥“key”长度超过64个字节，则还需配置HAU_CTL寄存器的KLM位。之后，将START位置位以启动HAU内核；
2. 密钥“key”作为输入消息来进行哈希模式下的计算；
3. 当输入了最后一个字并开始计算之后，HAU生成新的密钥“key”；
4. 在第一次的哈希计算后，外部的哈希函数使用新的密钥“key”。当输入了密钥的最后一个字，则开始进行计算，计算结果可从HAU_DO寄存器中读取。

11.5. 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。这两个中断用于指示输入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。

输入 FIFO 中断

当输入FIFO中有足够的空间时，输入FIFO中断标志位DINT置位。注意当DIIE位为0，禁用了输入FIFO中断时，DINT将保持为0。

计算完成中断

当摘要计算完成时，计算完成中断位CINT将置位。注意当CCIE位为0，禁用了计算完成中断时，CINT将保持为0。

11.6. HAU 寄存器

HAU 基地址：0x5006 0400

11.6.1. HAU 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													ALGM[1]	保留	KLM
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		MDS	DINE	NWIF[3:0]				ALGM[0]	HMS	DATAM[1:0]		DMAE	START	保留	
		rw	r	r				rw	rw	rw		rw	w		

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	ALGM[1]	算法选择位1
17	保留	必须保持复位值。
16	KLM	密钥长度模式 0：密钥长度 ≤ 64 字节 1：密钥长度 > 64 字节 **注意：**必须在非计算期间修改该位。
15:14	保留	必须保持复位值。
13	MDS	多DMA选择 如果哈希消息为大型文件需要多个DMA传输时，将此位置1 0：仅需要单次DMA传输，在DMA传输结束时硬件自动将CALEN位置1 1：需要多次DMA传输，在DMA传输结束时硬件不自动将CALEN位置1
12	DINE	DI寄存器非空 0：输入FIFO空 1：输入FIFO非空 **注意：**当START位或CALEN位为1时此位会清零。
11:8	NWIF[3:0]	输入FIFO中的字数 **注意：**当START位置位时，或开始进行摘要计算时（CALEN位置位，或者DMA传输结束），该位域清零。
7	ALGM[0]	算法选择位0 该位和CTL寄存器的位18用于选择SHA-1，SHA-224，SHA256或MD5 算法：

		00：选择SHA-1算法 01：选择MD5算法 10：选择SHA224算法 11：选择SHA256算法
6	HMS	HAU模式选择，必须在非计算期间修改该位 0：选择HASH模式 1：选择HMAC模式。如果密钥长度大于64字节，则还需配置KLM位。
5:4	DATAM[1:0]	数据交换类型 定义输入到HAU_DI寄存器中的数据格式 00：不交换，写入到HAU_DI寄存器的数据将直接送入FIFO，不进行交换 01：半字交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行半字交换 10：字节交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行字节交换 11：位交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行位交换
3	DMAE	DMA使能 0：禁止DMA传输 1：使能DMA传输 **注意：**1.当DMA传输消息的最后一个数据时，将由硬件清零该位。当START置位时，不会清零该位。 2.如果DMA正在传输数据，将该位写入0不会中止当前的传输，而直到当前传输结束或START位置为1之后，才会禁止传输。
2	START	开始摘要计算 1：开始新消息的摘要计算 0：没有影响 **注意：**读取该位将始终返回0。
1:0	保留	必须保持复位值。

11.6.2. HAU 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。

11.6.3. HAU 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。

11.6.4. HAU 数据输出寄存器 (HAU_DO0..7)

复位值：0x0000 0000
该寄存器只能按字（32位）访问

数据输出寄存器为只读寄存器，用于从输出FIFO中接收计算结果。置位START位将复位该寄存器。当正在进行摘要计算时，所有对该寄存器的新的读访问将被延迟，直到计算完成。
在SHA-1模式中，使用HAU_DO0…4

在MD5模式中，使用HAU_DO0…3
在SHA-224模式中，使用HAU_DO0…6
在SHA-256模式中，使用HAU_DO0…7

HAU_DO0

地址偏移：0x0C 和 0x310

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

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

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

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

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

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

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

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

11.6.5. HAU 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

11.6.6. HAU 状态与中断标志寄存器 (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	CINT	DINT
												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	CINT	计算完成中断状态标志 0：计算未完成 1：所有消息摘要计算完成

		注意：当CCIE=0时该位会被清零
0	DINT	数据输入中断状态标志 0：输入FIFO的空闲位置少于16个 1：输入FIFO有16个空闲位置，可以将一个新块输入到输入FIFO中 **注意：**当DIIE=0时该位会被清零。

12. 直接存储器访问控制器（DMA）

12.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 14 个通道（DMA0 有 7 个通道，DMA1 有 7 个通道）。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

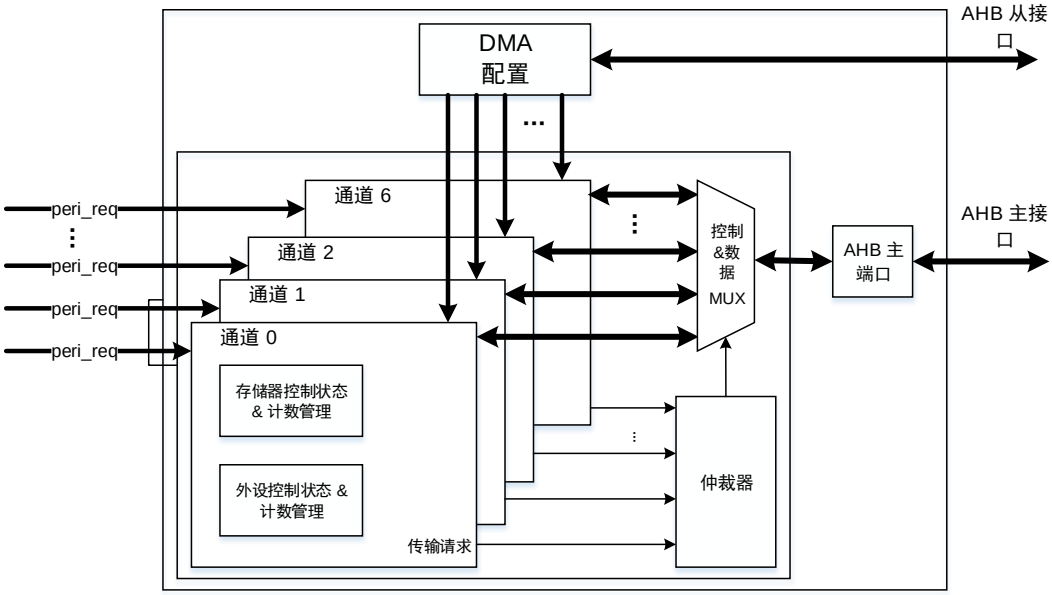
DMA 控制器和 Cortex®-M3 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

12.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 14 个通道，并且每个通道都可配置（DMA0 有 7 个通道，DMA1 有 7 个通道）；
- AHB 和 APB 外设，片上闪存和 SRAM 都可以作为访问的源端和目的端；
- 每个通道连接固定的硬件 DMA 请求；
- 支持软件优先级（低、中、高、极高）和硬件优先级（通道号越低，优先级越高）；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输；
- 每个通道有 3 种类型的事件标志和独立的中断，支持中断的使能和清除。
- 支持完整数据传输模式：当源端的传输数据宽度与目的端的传输数据宽度不相等时，并且源端传输数据宽度为 32-bit 时，DMA 能自动的根据目的端的传输数据宽度，优化匹配传输数据的次数，以保证数据全部传输完成。该模式只适用于 DMA1 的第 5 通道。

12.3. 结构框图

图 11-4. DMA 结构框图



由*图 11-4. DMA 结构框图*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA；
- AHB 主接口进行数据传输；
- 仲裁器进行 DMA 请求的优先级管理；
- 数据处理和计数。

12.4. 功能说明

12.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDTH 和 MWIDTH 位域决定每次发送和接收的字节数（字节/半字/字）。通过配置 DMA_ACFG 寄存器的 FD_CH5EN 位切换普通传输模式或完整数据传输模式。

DMA_CHxCNT寄存器的CNT位域必须在CHEN位置位前被配置，其控制传输的次数。在传输过程中，CNT位域的值表示还有多少次数据传输将被执行。

将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
  - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输；

- 在重新使能 DMA 通道前，对任意相关寄存器进行了操作，则 DMA 将开始一次新的传输。

■ 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

普通传输模式

假设DMA_CHxCNT寄存器的值为4，并且PNAGA和MNAGA位均置位。结合PWIDTH和MWIDTH的各种配置，DMA传输的操作详见*表11-1. DMA传输操作（普通模式）*。

表 11-1. DMA 传输操作（普通模式）

传输宽度		传输操作	
源	目标	源	目标
32 位	32 位	1：读 B3B2B1B0[31:0] @0x0 2：读 B7B6B5B4[31:0] @0x4 3：读 BBBAB9B8[31:0] @0x8 4：读 BFBEBDBC[31:0] @0xC	1：写 B3B2B1B0[31:0] @0x0 2：写 B7B6B5B4[31:0] @0x4 3：写 BBBAB9B8[31:0] @0x8 4：写 BFBEBDBC[31:0] @0xC
32 位	16 位	1：读 B3B2B1B0[31:0] @0x0 2：读 B7B6B5B4[31:0] @0x4 3：读 BBBAB9B8[31:0] @0x8 4：读 BFBEBDBC[31:0] @0xC	1：写 B1B0[7:0] @0x0 2：写 B5B4[7:0] @0x2 3：写 B9B8[7:0] @0x4 4：写 BDBC[7:0] @0x6
32 位	8 位	1：读 B3B2B1B0[31:0] @0x0 2：读 B7B6B5B4[31:0] @0x4 3：读 BBBAB9B8[31:0] @0x8 4：读 BFBEBDBC[31:0] @0xC	1：写 B0[7:0] @0x0 2：写 B4[7:0] @0x1 3：写 B8[7:0] @0x2 4：写 BC[7:0] @0x3
16 位	32 位	1：读 B1B0[15:0] @0x0 2：读 B3B2[15:0] @0x2 3：读 B5B4[15:0] @0x4 4：读 B7B6[15:0] @0x6	1：写 0000B1B0[31:0] @0x0 2：写 0000B3B2[31:0] @0x4 3：写 0000B5B4[31:0] @0x8 4：写 0000B7B6[31:0] @0xC
16 位	16 位	1：读 B1B0[15:0] @0x0 2：读 B3B2[15:0] @0x2 3：读 B5B4[15:0] @0x4 4：读 B7B6[15:0] @0x6	1：写 B1B0[15:0] @0x0 2：写 B3B2[15:0] @0x2 3：写 B5B4[15:0] @0x4 4：写 B7B6[15:0] @0x6
16 位	8 位	1：读 B1B0[15:0] @0x0 2：读 B3B2[15:0] @0x2 3：读 B5B4[15:0] @0x4 4：读 B7B6[15:0] @0x6	1：写 B0[7:0] @0x0 2：写 B2[7:0] @0x1 3：写 B4[7:0] @0x2 4：写 B6[7:0] @0x3
8 位	32 位	1：读 B0[7:0] @0x0 2：读 B1[7:0] @0x1 3：读 B2[7:0] @0x2 4：读 B3[7:0] @0x3	1：写 000000B0[31:0] @0x0 2：写 000000B1[31:0] @0x4 3：写 000000B2[31:0] @0x8 4：写 000000B3[31:0] @0xC
8 位	16 位	1：读 B0[7:0] @0x0 2：读 B1[7:0] @0x1 3：读 B2[7:0] @0x2	1：写 00B0[15:0] @0x0 2：写 00B1[15:0] @0x2 3：写 00B2[15:0] @0x4

传输宽度		传输操作	
源	目标	源	目标
		4：读 B3[7:0] @0x3	4：写 00B3[15:0] @0x6
8 位	8 位	1：读 B0[7:0] @0x0 2：读 B1[7:0] @0x1 3：读 B2[7:0] @0x2 4：读 B3[7:0] @0x3	1：写 B0[7:0] @0x0 2：写 B1[7:0] @0x1 3：写 B2[7:0] @0x2 4：写 B3[7:0] @0x3

完整数据传输模式

在完整数据传输模式中，源端的传输数据尺寸与目的端的传输数据尺寸必须不相等，并且源端的数据传输尺寸必须为32-bit。如果目的端的传输数据尺寸16-bit，每次DMA传输会在源端操作一次32-bit数据，在目的端操作两次16-bit数据；如果目的端的传输数据尺寸8-bit，每次DMA传输会在源端操作一次32-bit数据，在目的端操作四次8-bit数据；设置为假设DMA_CHxCNT寄存器的值为2，并且PNAGA和MNAGA位均置位。结合PWIDTH和MWIDTH的各种配置，DMA传输的操作详见*表11-2. DMA传输操作（完整数据传输模式）*。

表 11-2. DMA 传输操作（完整数据传输模式）

传输宽度		传输操作	
源	目标	源	目标
32 位	16 位	1：读 B3B2B1B0[31:0] @0x0	1：写 B1B0[16:0] @0x0 2：写 B3B2[16:0] @0x2
		2：读 B7B6B5B4[31:0] @0x4	1：写 B5B4[16:0] @0x4 2：写 B7B6[16:0] @0x6
32 位	8 位	1：读 B3B2B1B0[31:0] @0x0	1：写 B0[7:0] @0x0 2：写 B1[7:0] @0x1 3：写 B2[7:0] @0x2 4：写 B3[7:0] @0x3
		2：读 B7B6B5B4[31:0] @0x4	1：写 B4[7:0] @0x4 2：写 B5[7:0] @0x5 3：写 B6[7:0] @0x6 4：写 B7[7:0] @0x7

注意：完整数据传输模式只适用于 DMA1 的第 5 通道。

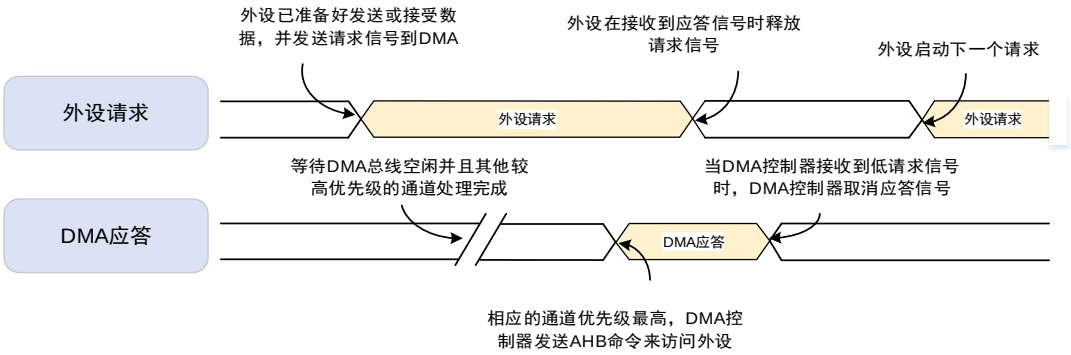
12.4.2. 外设握手

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

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

*图11-5. 握手机制*中详细描述了DMA控制器与外设之间的握手机制。

图 11-5. 握手机制



12.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA_CHxCTL的PRIO位域来配置。
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

12.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA_CHxCTL的PNAGA和MNAGA位用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址（DMA_CHxPADDR，DMA_CHxMADDR）。

在增量模式中，下一次传输数据的地址是当前地址加1（或者2，4），这个值取决于数据传输宽度。

12.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA_CHxCTL寄存器的CMEN位置位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位（DMA_CHxCTL寄存器的CHEN位）被清0。

12.4.6. 存储器到存储器模式

将DMA_CHxCTL寄存器的M2M位置位可以使能存储器到存储器模式。在此模式下，DMA通道传输数据时不依赖外设的请求信号。一旦DMA_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA_CHxCNT寄存器达到0，DMA通道才会停止。

12.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，如果为 1（通道已使能），清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA 开始新的传输。
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式。
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式。
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级。
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法。
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位。
7. 通过 DMA_CHxPADDR 寄存器配置外设基地址。
8. 通过 DMA_CHxMADDR 寄存器配置存储器基地址。
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量。
10. 按需要配置 DMA_ACFG 寄存器设置 DMA1 通道 5 的传输模式。
11. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1，使能 DMA 通道。

12.4.8. 中断

每个DMA通道都有一个专用的中断。中断事件有三种类型：传输完成，半传输完成和传输错误。

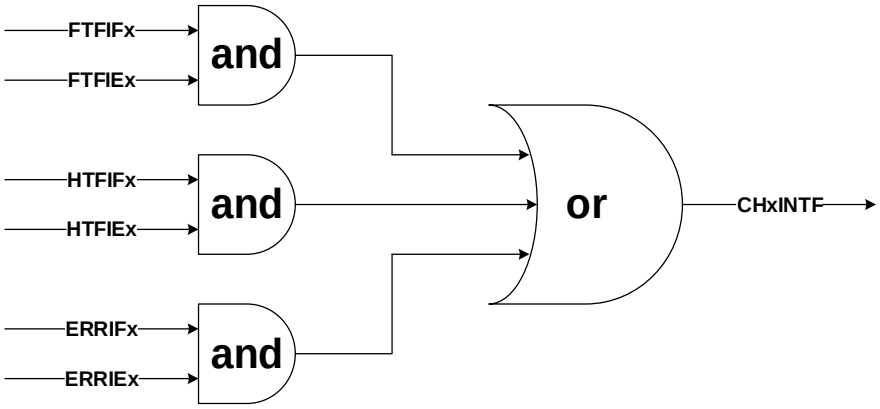
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表11-3. 中断事件*描述了其对应关系。

表 11-3. 中断事件

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成	FTFIF	FTFIFC	FTFIE
半传输完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如*表11-4. DMA0各通道请求表*所示，任何类型中断使能时，在产生了相应中断事件时均会产生中断。

图 11-6. DMA 中断逻辑图



注意: “x”表示通道数（x=0…6）。

12.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道。这些请求信号在经过逻辑或后进入 DMA。详情可见*图 11-7. DMA0 请求映射*和*图 11-8. DMA1 请求映射*。通过配置对应外设的寄存器，每个外设的请求均可以独立的开启或关闭。用户必须确保同一时间，在同一个通道上仅有一个外设的请求被开启。*表 11-4. DMA0 各通道请求表*列举了 DMA0 的每个通道所支持的外设请求，*表 11-5. DMA1 各通道请求表*列举了 DMA1 的每个通道所支持的外设请求。

图 11-7. DMA0 请求映射

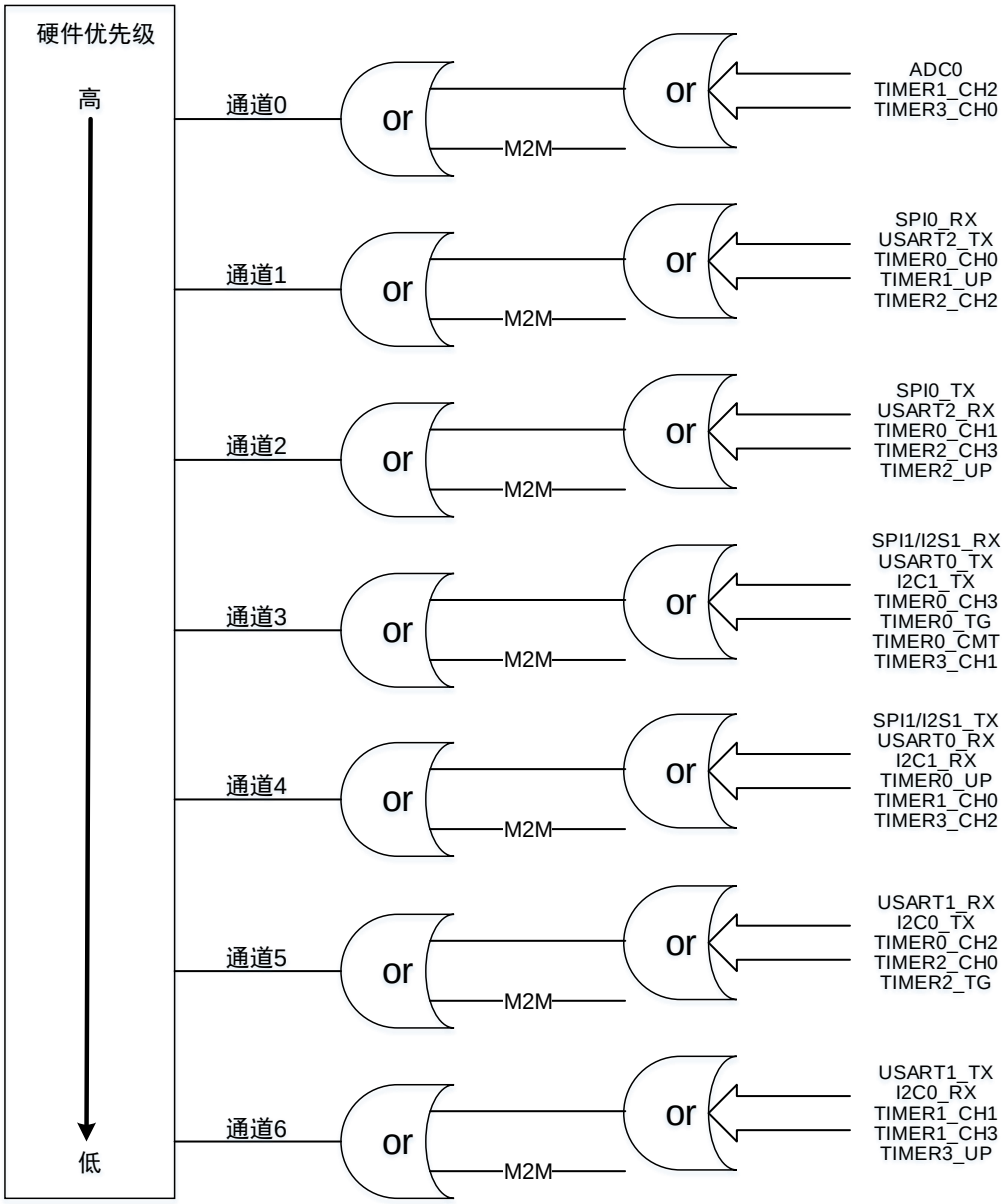


表11-4. DMA0各通道请求表

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

外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
SPI/I2S	•	SPI0_RX	SPI0_TX	SPI1/I2S1_RX	SPI1/I2S1_TX	•	•
USART	•	USART2_TX	USART2_RX	USART0_TX	USART0_RX	USART1_RX	USART1_TX
I2C	•	•	•	I2C1_TX	I2C1_RX	I2C0_TX	I2C0_RX

图11-8. DMA1请求映射

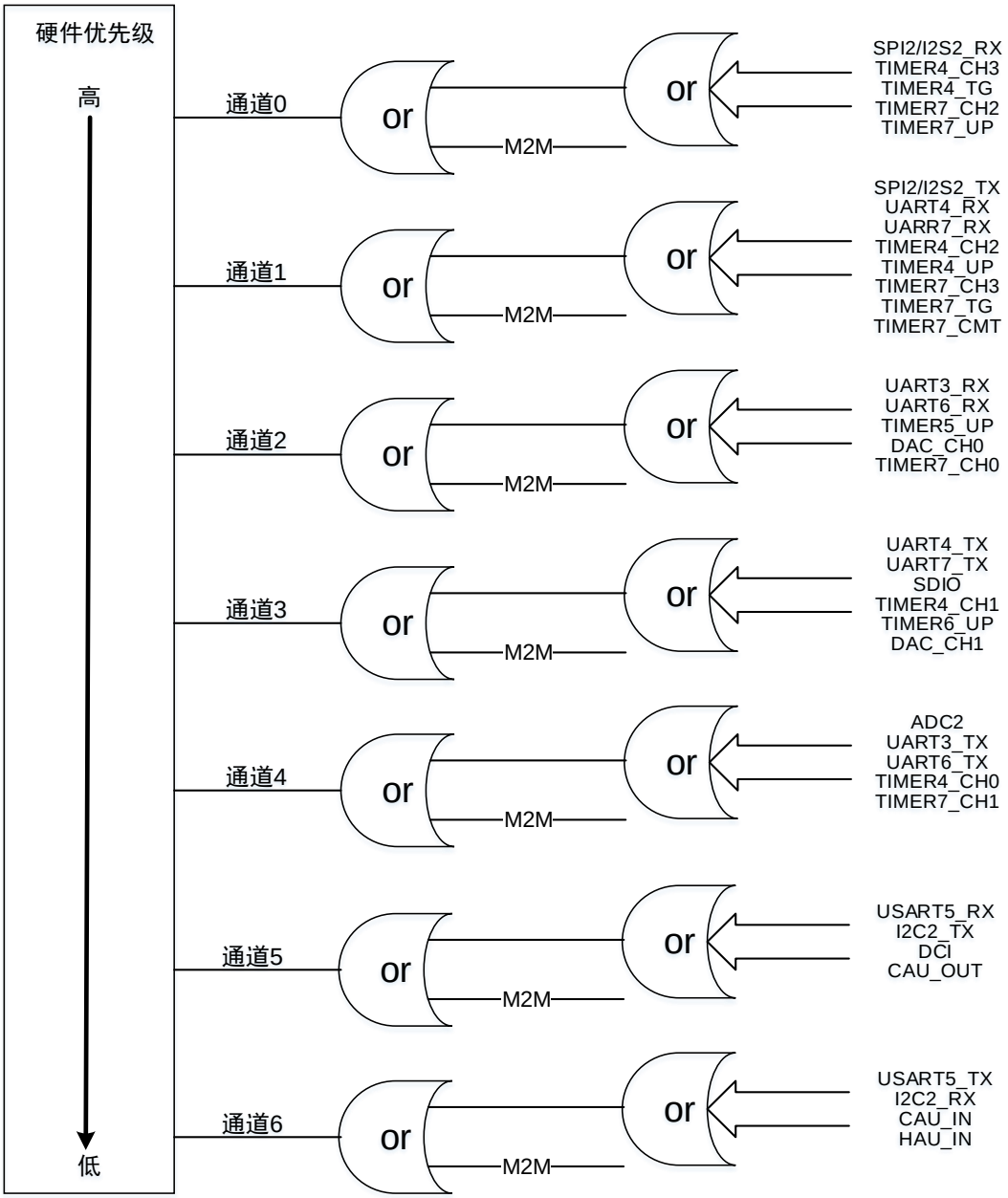


表11-5. DMA1各通道请求表

外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
TIMER4	TIMER4_CH3 TIMER4_TG	TIMER4_CH2 TIMER4_UP	•	TIMER4_CH1	TIMER4_CH0	•	•
TIMER5/ DAC_ CH0	•	•	TIMER5_UP/ DAC_CH0	•	•	•	•
TIMER6/ DAC_ CH1	•	•	•	TIMER6_UP/ DAC_CH1	•	•	•
TIMER7	TIMER7_CH2 TIMER7_UP	TMER7_CH3 TMER7_TG TMER7_CMT	TMER7_CH0	•	TMER7_CH1	•	•
ADC2	•	•	•	•	ADC2	•	•
SPI/I2S	SPI2/I2S2_RX	SPI2/I2S2_TX	•	•	•	•	•
USART	•	UART4_RX UART7_RX	UART3_RX UART6_RX	UART4_TX UART7_TX	UART3_TX UART6_TX	USART5_RX	USART5_TX
SDIO	•	•	•	SDIO	•	•	•
I2C2	•	•	•	•	•	I2C2_TX	I2C2_RX
DCI	•	•	•	•	•	DCI	•
CAU	•	•	•	•	•	CAU_OUT	CAU_IN
HAU	•	•	•	•	•	•	HAU_IN

12.5. DMA 寄存器

DMA0 基地址：0x4002 0000

DMA1 基地址：0x4002 0400

12.5.1. 中断标志位寄存器（DMA_INTF）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

12.5.2. 中断标志位清除寄存器（DMA_INTC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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

12.5.3. 通道 x 控制寄存器（DMA_CHxCTL，x = 0…6）

地址偏移：0x08 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件置位和清零 0：禁止存储器到存储器模式

		1：使能存储器到存储器模式 CHEN位为1时，该位不能被配置
13:12	PRIO[1:0]	软件优先级 软件置位和清零 00：低 01：中 10：高 11：极高 CHEN位为1时，该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 CHEN位为1时，该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN位为1时，该位不能被配置
5	CMEN	循环模式使能 软件置位和清零 0：禁止循环模式 1：使能循环模式 CHEN位为1时，该位不能被配置
4	DIR	传输方向 软件置位和清零

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

12.5.4. 通道 x 计数寄存器（DMA_CHxCNT，x = 0…6）

地址偏移：0x0C + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	传输计数 CHEN位为1时，该位域不能被配置 该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个DMA传输之后值减1。如果该寄存器的值为0，无论通道开启与否，都不会有数据传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初始设置值。

12.5.5. 通道 x 外设基地址寄存器（DMA_CHxPADDR，x = 0…6）

地址偏移：0x10 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	PADDR[31:0]	外设基地址 CHEN位为1时，该位域不能被配置 当PWIDTH位域的值为01 (16-bit)，PADDR[0]被忽略，访问自动与16位地址对齐。 当PWIDTH位域的值为10 (32-bit)，PADDR [1:0]被忽略，访问自动与32位地址对齐。

12.5.6. 通道 x 存储器基地址寄存器（DMA_CHxMADDR，x = 0…6）

地址偏移：0x14 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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

位/位域	名称	描述
31:0	MADDR[31:0]	存储器基地址 CHEN位为1时，该位域不能被配置 当MWIDTH位域的值为01 (16-bit)时，MADDR [0]被忽略，访问自动与16位地址对齐。 当MWIDTH位域的值为10 (32-bit)时，MADDR [1:0]被忽略，访问自动与32位地址对齐。

12.5.7. DMA 附加配置寄存器（DMA_ACFG）

地址偏移：0x0300

复位值：0x0000 0000

注意：该寄存器不适用于 DMA0。

该寄存器只能按字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										FD_CH5EN	保留				
										rw					

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	FD_CH5EN	DMA1通道5工作模式配置位 CHEN位为1时，该位不能被配置 0：关闭DMA1通道5完整数据传输模式 1：开启DMA1通道5完整数据传输模式
4:0	保留	必须保持复位值。

13. 调试（DBG）

13.1. 简介

GD32F20x系列产品提供了各种各样的调试，跟踪和测试功能。这些功能通过ARM CoreSight组件的标准配置和链状连接的TAP控制器来实现的。调试和跟踪功能集成在ARM Cortex-M3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-M3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或者一些外设调试。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、WWDGT、FWDGT、I2C和CAN。

13.2. 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。

13.2.1. 切换 JTAG/ SW 接口

默认使用JTAG调试接口，可以通过下列软件序列从JTAG调试切换到SW调试：

- 发送50个以上TCK周期的TMS = 1信号；
- 发送16位TMS = 1110011110011110（0xE79E LSB）信号；
- 发送50个以上TCK周期的TMS = 1信号。

切换SW调试到JTAG调试的软件序列：

- 发送50个以上TCK周期的TMS=1信号；
- 发送16位TMS = 1110011100111100（0xE73C LSB）信号；
- 发送50个以上TCK周期的TMS = 1信号。

13.2.2. 引脚分配

JTAG调试提供五个引脚的接口：JTAG时钟引脚（JTCK），JTAG模式选择引脚（JTMS），JTAG数据输入引脚（JTDI），JTAG数据输出引脚（JTDO），JTAG复位引脚（NJTRST，低电平有效）。串行调试（SWD）提供两个引脚的接口：数据输入输出引脚（SWDIO）和时钟引脚（SWCLK）。SW调试接口的两个引脚与JTAG调试接口的两个引脚复用，SWDIO和JTMS复用，SWCLK和JTCK复用。

当异步跟踪功能开启时，JTDO引脚也用作异步跟踪数据输出（TRACESWO）。

表 13-1. 调试引脚分配

引脚	调试接口
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB4	NJTRST
PB3	JTDO

默认复位后使用五个引脚的JTAG调试，用户可以在不使用NJTRST引脚情况下正常使用JTAG功能，此时PB4可以用作普通GPIO功能（NJTRST硬件拉高）。如果切换到SW调试模式，PA15/PB4/PB3释放作为普通GPIO功能。如果JTAG和SW调试功能都没有使用，这五个引脚都释放作为普通GPIO功能。五个引脚具体配置请参考*通用和备用输入/输出接口（GPIO和AFIO）*。

13.2.3. JTAG 链状结构

Cortex-M3内核的JTAG TAP和边界扫描（BSD）TAP串行连接。边界扫描（BSD）JTAG的IR（指令寄存器）是5位，而Cortex-M3内核的JTAG的IR（指令寄存器）是4位。所以当JTAG进行IR移位输入时，首先移位5位BYPASS指令给BSD JTAG，然后移位4位标准指令给Cortext-M3 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为BSD JTAG已处在BYPASS模式。

BSD JTAG ID代码是0x790007A3。

13.2.4. 调试复位

JTAG-DP和SW-DP寄存器位于上电复位域。系统复位初始化了Cortex-M3的绝大部分组件，除了NVIC，调试逻辑（FPB，DWT，ITM）。NJTRST能复位JTAG TAP控制器。所以，可以在系统复位下实现调试功能。例如：复位后停止，用户在系统复位后配置相应停止位，系统复位释放后处理器会立即停止。

13.2.5. JEDEC-106 ID code

Cortex-M3集成了JEDEC-106 ID代码。位于ROM表中，映射地址为0xE00FF000_0xE00FFFFF。

13.3. 调试保持功能描述

13.3.1. 低功耗模式调试支持

当DBG控制寄存器（DBG_CTL）的STB_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器（DBG_CTL）的DSLP_HOLD位置1并且进入深度睡眠模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器（DBG_CTL）的SLP_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

13.3.2. TIMER，I2C，WWDGT，FWDGT 和 CAN 外设调试支持

当内核停止，并且DBG控制寄存器（DBG_CTL）中的相应位置1。对于不同外设，有不同动作：

对于TIMER外设，TIMER计数器停止并进行调试；

对于I2C外设，SMBUS保持状态并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试；

对于CAN外设，接收寄存器停止计数并进行调试。

13.4. DBG 寄存器

DBG基地址：0xE004 2000

13.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00
只读寄存器

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数。

13.4.2. 控制寄存器（DBG_CTL）

地址偏移：0x04
复位值：0x0000 0000，仅上电复位

该寄存器只能按字（32 位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2C2_HOLD	TIMER10_HOLD	TIMER9_HOLD	TIMER8_HOLD	TIMER13_HOLD	TIMER12_HOLD	TIMER11_HOLD	保留.			CAN1_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER7_HOLD	I2C1_HOLD
rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw

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	TRACE_MODE[1:0]		TRACE_IOEN	保留.		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	I2C2_HOLD	I2C2 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C2 的 SMBUS 状态不变，用于调试。
30	TIMER10_HOLD	TIMER10 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 10 计数器不变，用于调试。

29	TIMER9_HOLD	TIMER9 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 9 计数器不变，用于调试。
28	TIMER8_HOLD	TIMER8 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 8 计数器不变，用于调试。
27	TIMER13_HOLD	TIMER13 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 13 计数器不变，用于调试。
26	TIMER12_HOLD	TIMER12 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 13 计数器不变，用于调试。
25	TIMER11_HOLD	TIMER11 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 13 计数器不变，用于调试。
24:22	保留	必须保持复位值。
21	CAN1_HOLD	CAN1 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时 CAN1 接收寄存器停止接收数据。
20	TIMER6_HOLD	TIMER6 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 6 计数器不变，用于调试。
19	TIMER5_HOLD	TIMER5 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 5 计数器不变，用于调试。
18	TIMER4_HOLD	TIMER4 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 4 计数器不变，用于调试。
17	TIMER7_HOLD	TIMER7 保持寄存器 该位由软件置位和复位

		0：无影响 1：当内核停止时保持定时器 7 计数器不变，用于调试。
16	I2C1_HOLD	I2C1 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C1 的 SMBUS 状态不变，用于调试。
15	I2C0_HOLD	I2C0 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C0 的 SMBUS 状态不变，用于调试。
14	CAN0_HOLD	CAN0 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时 CAN0 接收寄存器停止接收数据。
13	TIMER3_HOLD	TIMER3 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 3 计数器不变，用于调试。
12	TIMER2_HOLD	TIMER2 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 2 计数器不变，用于调试。
11	TIMER1_HOLD	TIMER1 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 1 计数器不变，用于调试
10	TIMER0_HOLD	TIMER0 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持定时器 0 计数器不变，用于调试。
9	WWDGT_HOLD	WWDG 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持WWDGT计数器时钟，用于调试。
8	FWDGT_HOLD	FWDGT 保持寄存器 该位由软件置位和复位 0：无影响 1：当内核停止时保持FWDGT计数器时钟，用于调试。
7:6	TRACE_MODE[1:0]	跟踪引脚分配模式

		该位由软件置位和复位 00：跟踪引脚用于异步模式 01：跟踪引脚用于同步模式且数据长度为 1 10：跟踪引脚用于同步模式且数据长度为 2 11：跟踪引脚用于同步模式且数据长度为 4
5	TRACE_IOEN	跟踪引脚分配使能 该位由软件置位和复位 0：跟踪引脚分配禁用 1：跟踪引脚分配使能
4:3	保留	必须保持复位值。
2	STB_HOLD	待机模式保持寄存器 该位由软件置位和复位 0：无影响 1：在待机模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供，当退出待机模式时，产生系统复位。
1	DSLP_HOLD	深度睡眠模式保持寄存器 该位由软件置位和复位 0：无影响 1：在深度睡眠模式下，系统时钟和 AHB 时钟由 CK_IRC8M 提供。
0	SLP_HOLD	睡眠模式保持寄存器 该位由软件置位和复位 0：无影响 1：在睡眠模式下，AHB 时钟继续运行。

14. 模数转换器（ADC）

14.1. 简介

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

14.2. 主要特征

- 高性能：
  - ADC采样分辨率：12位、10位、8位、或者6位分辨率；
  - ADC采样率：12位分辨率为2MSPs。分辨率越低，转换越快；
  - 前置校准功能；
  - 可编程采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA请求。
- 模拟输入通道：
  - 16个外部模拟输入通道；
  - 1个内部温度传感通道(V_{SENSE})；
  - 1个内部参考电压输入通道(V_{REFINT})。
- 转换开始的发起：
  - 软件；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一序列的通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式；
  - 同步模式（适用于具有两个或多个ADC的设备）。
- 转换结果阈值监测器功能：模拟看门狗。
- 中断的产生：
  - 常规序列转换结束；
  - 模拟看门狗事件。
- 过采样：
  - 16位的数据寄存器；
  - 可调整的过采样率，从2x到256x；
  - 高达8位的可编程数据移位。
- 模块供电要求：2.6V到3.6V，一般电源电压为3.3V。
- 通道输入范围：$V_{REFN} \leq V_{IN} \leq V_{REFP}$。

14.3. 引脚和内部信号

*图14-1. ADC模块框图*给出了ADC框图。*表14-1. ADC内部输入信号*给出了ADC内部信号。*表14-2. ADC输入引脚定义*给出了ADC引脚定义说明。

表 14-1. ADC 内部输入信号

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

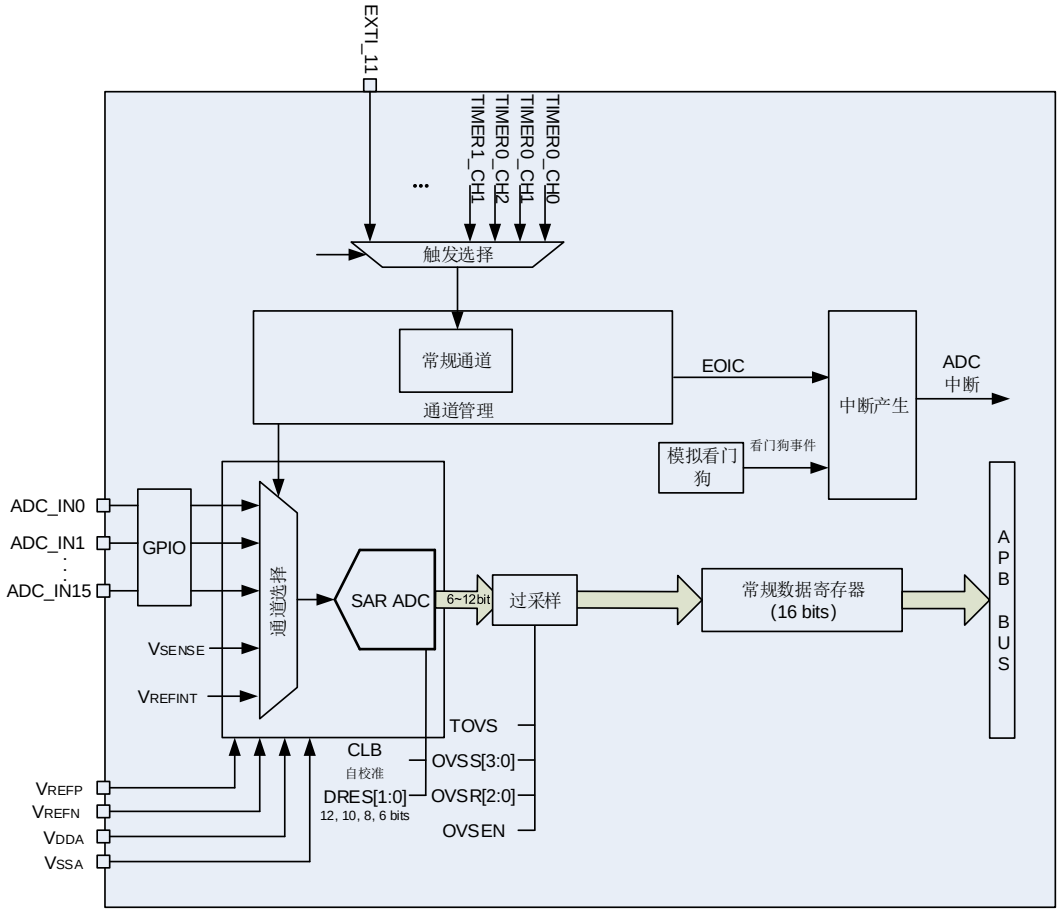
表 14-2. ADC 输入引脚定义

名称	注释
V_{DDA}	模拟电源输入等于 V_{DD}，2.6V≤V_{DDA}≤3.6V
V_{SSA}	模拟地，等于 V_{SS}
V_{REFP}	ADC 正参考电压，2.6 V ≤ V_{REFP} ≤ V_{DDA}
V_{REFN}	ADC 负参考电压，V_{REFN} = V_{SSA}
ADCx_IN[15:0]	多达 16 路外部通道

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

14.4. 功能说明

图 14-1. ADC 模块框图



14.4.1. 前置校准功能

在前置校准期间，ADC 计算一个校准系数，这个系数是应用于 ADC 内部的，它直到 ADC 下次掉电才无效。在校准期间，应用不能使用 ADC，它必须等到校准完成。在 A/D 转换前应执行校准操作。通过软件设置 CLB=1 来对校准进行初始化，在校准期间 CLB 位会一直保持 1，直到校准完成，该位由硬件清 0。

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

内部的模拟校准通过设置 ADC_CTL1 寄存器的 RSTCLB 位来重置。

软件校准过程：

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

14.4.2. ADC 时钟

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 最大的时钟频率为 28MHz。ADC 时钟可以在 RCU 时钟控制器中进行分配和配置。

14.4.3. ADCON 使能

ADC_CTL1 寄存器中的 ADCON 位是 ADC 模块的使能开关。如果该位为 0，则 ADC 模块保持复位状态。为了省电，当 ADCON 位为 0 时，ADC 模拟子模块将会进入掉电模式。ADC 使能后需要等待 t_{su} 时间后才能采样，t_{su} 数值详见芯片数据手册。

14.4.4. 常规序列

通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多16个通道，每个通道称为常规通道。

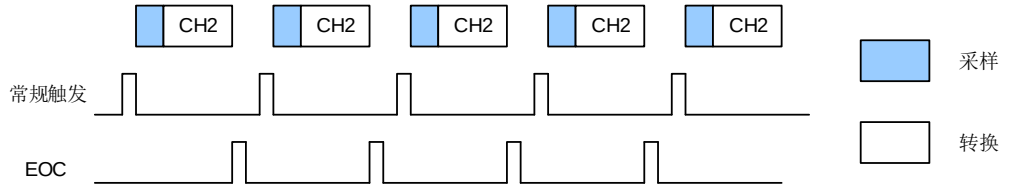
ADC_RSQ0寄存器的RL[3:0]位规定了整个常规序列转列的长度。ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

14.4.5. 运行模式

单次运行模式

单次运行模式下，ADC_RSQ2 寄存器的 RSQ0[4:0]位规定了 ADC 的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者外部触发发生，ADC 就会采样和转换一个通道。

图 14-2. 单次运行模式



常规通道单次转换结束后，转换数据将被存放于 ADC_RDATA 寄存器中，EOC 将会置 1。如果 EOCIE 位被置 1，将产生一个中断。

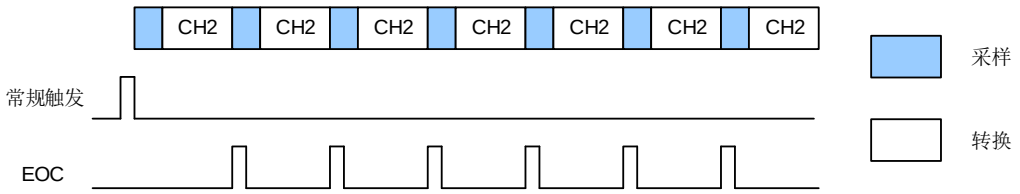
常规序列单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC和SM位以及ADC_CTL1寄存器的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等待EOC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

连续运行模式

对 ADC_CTL1 寄存器的 CTN 位置 1 可以使能连续运行模式。在此模式下，ADC 执行由 RSQ0[4:0]规定的转换通道。当 ADCON 位被置 1，一旦相应软件触发或者外部触发产生，ADC 就会采样和转换规定的通道。转换数据保存在 ADC_RDATA 寄存器中。

图 14-3. 连续运行模式



常规序列连续运行模式的软件流程：

1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者给常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位；
9. 只要还需要进行连续转换，重复步骤6~8。

由于要循环查询EOC标志位，DMA可以被用来传输转换数据，软件流程如下：

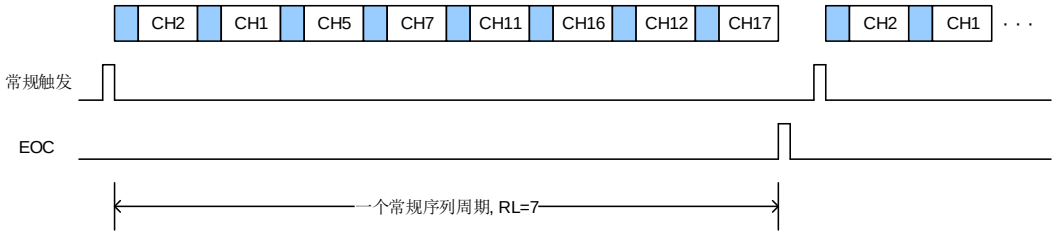
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

扫描运行模式

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

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

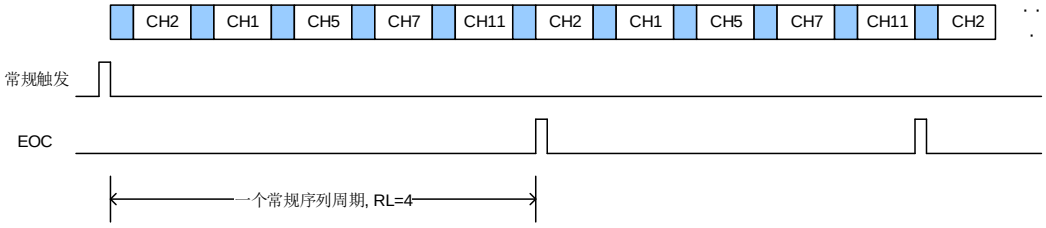
图 14-4. 扫描运行模式，且连续运行模式失能



常规序列扫描运行模式的软件流程：

1. 设置ADC_CTL0寄存器的SM位和ADC_CTL1寄存器的DMA位为1；
2. 配置ADC_RSQx和ADC_SAMPTx寄存器；
3. 如果有需要，配置ADC_CTL1寄存器中的ETERC和ETSRC位；
4. 准备DMA模块，用于传输来自ADC_RDATA的数据；
5. 设置SWRCST位，或者给常规序列产生一个外部触发；
6. 等待EOC标志位置1；
7. 写0清除EOC标志位。

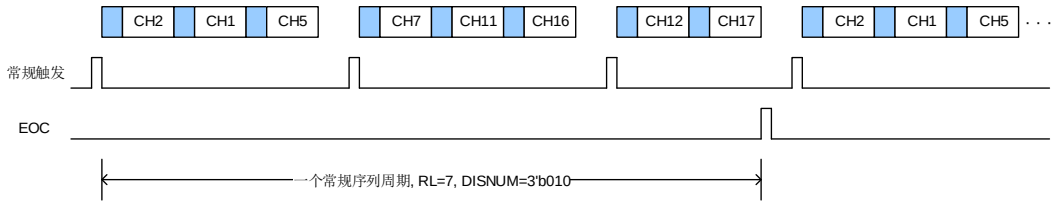
图 14-5. 扫描运行模式，连续运行模式使能



间断运行模式

对于常规序列，当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列使能间断运行模式。该模式下可以执行一次 n 个通道的短序列转换(n 不超过 8)，该序列是 ADC_RSQ0~RSQ2 寄存器所选择的序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位配置。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所配置通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EOC 位将被置 1。如果 EOCIE 位被置 1 将产生一个中断。

图 14-6. 间断运行模式



常规序列间断运行模式的软件流程：

1. 设置ADC_CTL0寄存器的DISRC位和ADC_CTL1寄存器的DMA位为1；
2. 配置ADC_CTL0寄存器的DISNUM[2:0]位；
3. 配置ADC_RSQx和ADC_SAMPTx寄存器；

4. 如果有需要，配置ADC_CTL1寄存器中的ETERC和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发；
7. 如果需要，重复步骤6；
8. 等待EOC标志位置1；
9. 写0清除EOC标志位。

14.4.6. 转换结果阈值监测功能

配置 ADC_CTL0 寄存器的 RWDEN 位置 1，可使能常规序列的模拟看门狗功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN，WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或者多通道。

14.4.7. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

在最高有效位对齐中，12/10/8/6 位数据按半字方式对齐，如下*图 14-7. 12 位数据存储模式*，*图 14-8. 10 位数据存储模式*，*图 14-9. 8 位数据存储模式*和*图 14-10. 6 位数据存储模式*所示。

图 14-7. 12 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

图 14-8. 10 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

图 14-9. 8 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

图 14-10. 6 位数据存储模式

常规通道数据

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

DAL=0

常规通道数据

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

DAL=1

14.4.8. 采样时间配置

ADC 使用多个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和 ADC_SAMPT1 寄存器的 SPTn[2:0]位配置。每个通道可以用不同的采样时间。在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 14MHz，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 1us。

14.4.9. 外部触发配置

外部触发输入的上升沿可以触发常规序列的转换。常规序列的外部触发源由 ADC_CTL1 寄存器的 ETSRC[2:0]位控制。

表 14-3. ADC0 和 ADC1 的外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER0_CH0	硬件触发
001	TIMER0_CH1	
010	TIMER0_CH2	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER3_CH3	
110	EXTI11/ TIMER7_TRGO	
111	SWRCST	软件触发

表 14-4. ADC2 的外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER2_CH0	硬件触发
001	TIMER1_CH2	
010	TIMER0_CH2	
011	TIMER7_CH0	
100	TIMER7_TRGO	
101	TIMER4_CH0	
110	TIMER4_CH2	
111	SWRCST	软件触发

14.4.10. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能，它用于传输常规序列多个通道的转换结果。ADC 在常规序列一个通道转换结束后产生一个 DMA 请求，DMA 接受到请求后可以将转换的数据从 ADC_RDATA 寄存器传输到用户指定的目的地址。

注意：只有 ADC0 和 ADC2 有 DMA 功能，ADC1 转换的数据可以在 ADC 同步模式下传输。

14.4.11. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVREN 位置 1 可以使能温度传感器通道(ADC0_CH16)和 V_{REFINT} 通道(ADC0_CH17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议温度传感器的采样时间至少设置为 ts_temp μs。温度传感器不用时，复位 TSVREN 位可以将其置于掉电模式。

温度传感器的输出电压随温度会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏差在芯片间会有不同(最多相差 45°C)。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

内部参考电压 (V_{REFINT})提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。V_{REFINT} 内部连接到 ADC0_CH17 输入通道。

使用温度传感器的软件流程：

1. 配置温度传感器通道（ADC_IN16）的转换序列和采样时间为ts_temp us
2. 置位ADC_CTL1寄存器中的TSVREN位，使能温度传感器
3. 置位ADC_CTL1寄存器的ADCON位，或者由外部触发ADC转换
4. 从ADC数据寄存器中读取并计算温度传感器数据Vtemperature，并由下面公式计算出实际温度：

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

V_{25}：内部温度传感器在 25°C 下的电压，典型值请参考数据手册。

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

14.4.12. 可编程分辨率(DRES)

ADC 分辨率可以通过寄存器 ADC_OVSAMPCTL 中的 DRES[1:0]位进行配置。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 位为 0 时，才能修改 DRES[1:0]的值。较低的分辨率能够减少转换时间。如*表 14-5. 不同分辨率对应的 tCONV 时间*所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 t_{ADC}。

表 14-5. 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}= 28MHz	t_{SMPL}(min)(ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(us) at f_{ADC}= 28MHz
12	12.5	446 ns	1.5	14	500 ns
10	10.5	375 ns	1.5	12	429 ns
8	8.5	304 ns	1.5	10	357 ns
6	6.5	232 ns	1.5	8	286 ns

14.4.13. 片上硬件过采样

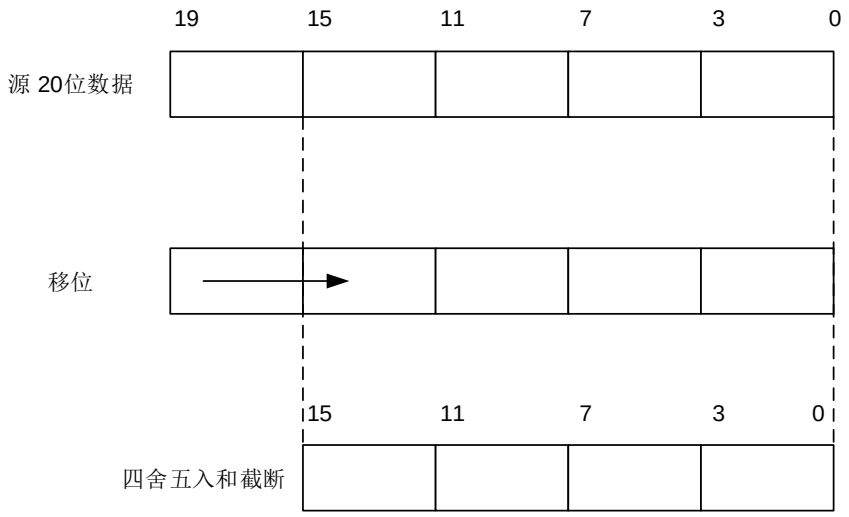
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果值根据如下公式计算得出，其中 N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSEN 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \tag{12-1}$$

片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC_OVSAMPCTL 寄存器的 OVSR[2:0]位定义，它的取值范围为 2x 到 256x。除法系数 M 定义一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

求和单元能够生成一个多达 20 位(256*12 位)的值。首先，将这个值要进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位会被截断，仅保留最低 16 位有效位作为最终值传入对应的数据寄存器中。

图 14-11. 20 位到 16 位的结果截断



注意：如果移位后的中间结果还是超过 16 位，那么该结果的高位就会被直接截掉。

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

图 14-12. 右移 5 位和取整的数例

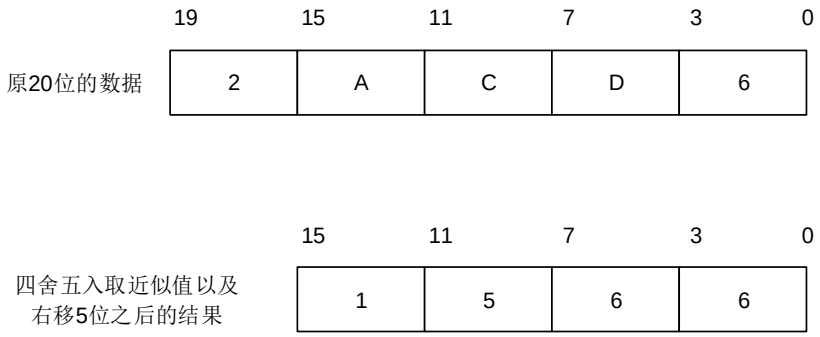


*表 14-6. N 和 M 的最大输出值（灰色部分表示截断）*给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFF。

表 14-6. N 和 M 的最大输出值（灰色部分表示截断）

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
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

和标准的转换模式相比，过采样模式的转换时间不会改变：在整个过采样序列的过程中采样时间仍然保持相等。每 N 个转换就会产生一个新的数据，一个等价的延迟为：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (14\text{-}2)$$

14.5. ADC 同步模式

在有多个ADC模块的产品中，可以使用ADC同步模式。在ADC同步模式下，根据ADC_CTL0寄存器中SYNCM[3:0]位所选的模式，转换的启动可以是ADC0和ADC1的交替触发或同步触发。

在同步模式下，当配置由外部事件触发的转换时，ADC0必须通过软件来配置触发来，从而避免错误的触发引起不必要的转换。此外，对于ADC0和ADC1的外部触发必须被使能。

ADC同步模式如*表14-7. ADC同步模式表*所示。

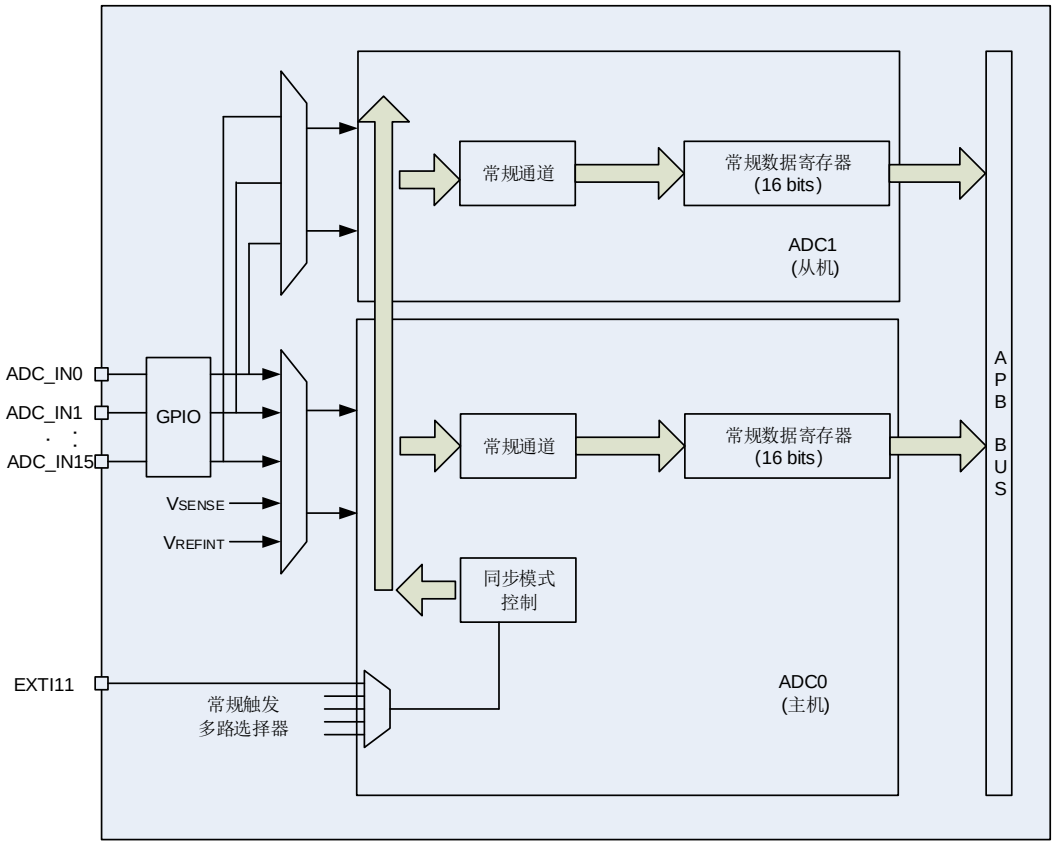
表 14-7. ADC 同步模式表

SYNCM[3: 0]	模式
0000	独立模式
0110	常规并行模式
0111	常规快速交叉模式
1000	常规慢速交叉模式

在ADC同步模式下，即使DMA不用，也要将DMA置位，ADC1的转换数据可以通过ADC0数据寄存器读取。

ADC 同步框图如*图 14-13. ADC 同步框图*所示。

图 14-13. ADC 同步框图



14.5.1. 独立模式

在这种模式下，每个 ADC 都独立工作，互不干扰。

14.5.2. 常规并行模式

此模式可并行转换常规序列，外部触发来源于 ADC0 常规序列（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定），ADC1 常规序列通道配置为软件触发模式。

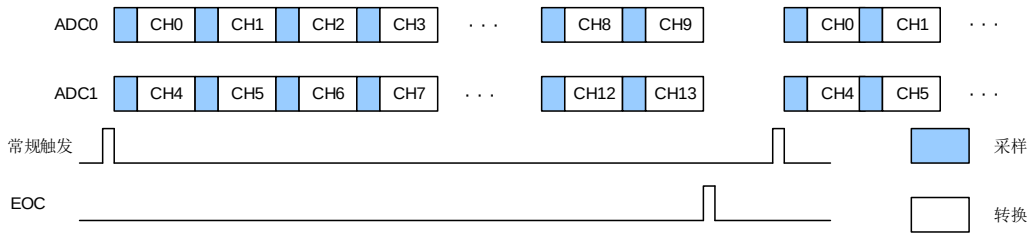
在 ADC0 或 ADC1 的转换事件结束时，即 ADC0 或 ADC1 的常规序列转换完毕，会产生一个 EOC 中断（如果某个 ADC 中断使能）。常规并行模式请参考*图 14-14. 基于 10 个通道的常规并行模式*。

32 位 ADC_RDATA 寄存器（ [15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据），32 位的 DMA 被用来将 ADC_RDATA 中的数据传送到 SRAM。

注意：

1. 若两个 ADC 模块使用了相同的采样通道，应保证不在同一时间使用该通道。
2. 两个 ADC 在同一时刻采样的两个通道，应该配置相同的采样时间。

图 14-14. 基于 10 个通道的常规并行模式



14.5.3. 常规快速交叉模式

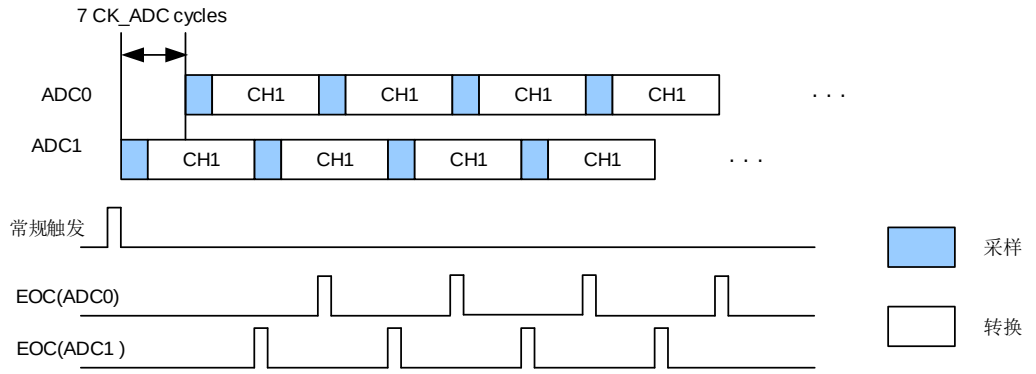
快速交叉模式适用于两个 ADC 的常规序列采样同一个通道，，外部触发来源于 ADC0 常规序列（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。

如果 ADC0 和 ADC1 的 CTN 位被置位，所选的常规序列在两个 ADC 中被不停的转换。如*图 14-15. 常规序列上的快速交叉模式*所示。

32 位 ADC_RDATA 寄存器（[15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据）。在 ADC0 产生 EOC 中断后(可通过置位 EOCIE 位)，可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

注意：两个 ADC 模块常规通道的采样时间都应小于 7 个 ADC 时钟周期。

图 14-15. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1）



14.5.4. 常规慢速交叉模式

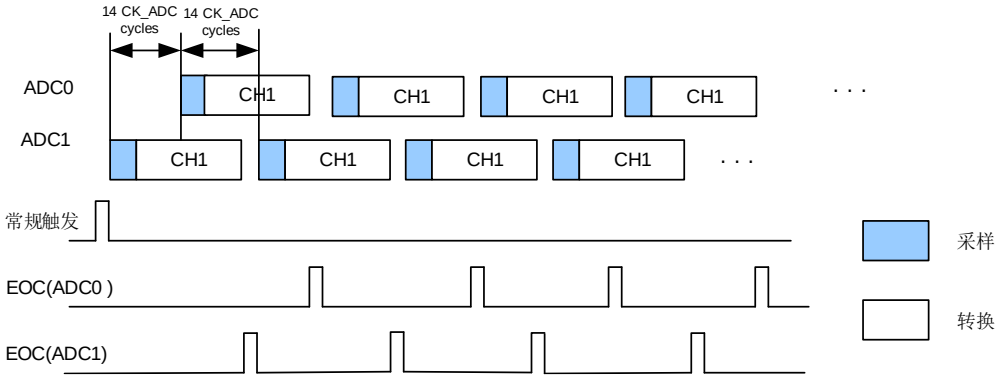
此模式应用于两个 ADC 的常规序列（通常一个常规通道），外部触发来源于 ADC0 常规序列（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 14 个 ADC 时钟周期后启动，在 ADC0 启动后的 14 个时钟周期，ADC1 再次启动。

在这种模式下，不能使用连续转换模式，因为在这种模式下所选的常规通道在两个 ADC 中被不停的转换，如*图 14-16. 常规序列通道上的慢速交叉模式*所示。

32 位 ADC_RDATA 寄存器（[15: 0]位域用于保存 ADC0 常规通道采样数据，[31: 16]位域用于保存 ADC1 常规通道采样数据）。在 ADC0 产生 EOC 中断后(可通过置位 EOCIE 位)，可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

注意：可允许的最大采样时间必须小于 14 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时钟重叠。

图 14-16. 常规序列通道上的慢速交叉模式



14.6. 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件。

单独的中断使能位可使得使用更灵活。

ADC0和ADC1的中断映射到相同的中断矢量，而ADC2独立映射到一个独立的中断矢量。

14.7. ADC 寄存器

ADC0 基地址：0x4001 2400

ADC1 基地址：0x4001 2800

ADC2 基地址：0x4001 3C00

14.7.1. 状态寄存器（ADC_STAT）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	说明
31:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0：转换没有开始 1：转换开始 常规序列转换开始时硬件置位，软件写 0 清除。
3:2	保留	必须保持复位值。
1	EOC	常规序列转换结束标志 0：转换没有结束 1：转换结束 常规序列转换结束时硬件置位，软件写 0 或读 ADC_RDATA 寄存器清除。
0	WDE	模拟看门狗事件标志 0：没有模拟看门狗事件 1：产生模拟看门狗事件 转换电压超过 ADC_WDLT 和 ADC_WDHT 寄存器设定的阈值时由硬件置 1，软件写 0 清除。

14.7.2. 控制寄存器 0（ADC_CTL0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			保留	DISRC	保留	WDSC	SM	保留	WDEIE	EOCIE	WDCHSEL[4:0]				
rw				rw		rw	rw		rw	rw	rw				

位/位域	名称	说明
31:24	保留	必须保持复位值。
23	RWDEN	常规序列看门狗使能 0：常规序列看门狗禁止 1：常规序列看门狗使能
22:20	保留	必须保持复位值。
19：16	SYNCM[3: 0]	同步模式选择 这些位用于运行模式选择 0000：独立模式 0110：常规并行模式 0111：常规快速交叉模式 1000：常规慢速交叉模式 **注意：**1）这些位只用于 ADC0；2）建议用户在任何配置之前关闭同步模式。
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成 DISNUM[2:0]+1
12	保留	必须保持复位值。
11	DISRC	常规序列间断模式 0：间断运行模式禁止 1：间断运行模式使能
10	保留	必须保持复位值。
9	WDSC	扫描模式下，模拟看门狗在通道配置 0：模拟看门狗在所有通道有效 1：模拟看门狗在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDEIE	WDE 中断使能 0：中断禁止

		1：中断使能
5	EOCIE	EOC 中断使能 0：中断禁止 1：中断使能
4:0	WDCHSEL[4:0]	模拟看门狗通道选择 00000：ADC 通道 0 00001：ADC 通道 1 00010：ADC 通道 2 00011：ADC 通道 3 00100：ADC 通道 4 00101：ADC 通道 5 00110：ADC 通道 6 00111：ADC 通道 7 01000：ADC 通道 8 01001：ADC 通道 9 01010：ADC 通道 10 01011：ADC 通道 11 01100：ADC 通道 12 01101：ADC 通道 13 01110：ADC 通道 14 01111：ADC 通道 15 10000：ADC 通道 16 10001：ADC 通道 17 其他值保留。 注意：ADC0 的模拟输入通道 16 和通道 17 分别连接到温度传感器和 V_{REFINT}。ADC1 的模拟输入通道 16 和通道 17 内部都连接到 V_{SSA}。 ADC2 的模拟输入通道 16 和通道 17 内部都连接到 V_{SSA}。

14.7.3. 控制寄存器 1（ADC_CTL1）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								TSVREN	SWRCST	保留	ETERC	ETSRC[2：0]			保留.
								rw	rw		rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAL	保留.		DMA	保留				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

位/位域	名称	说明

31：24	保留	必须保持复位值。
23	TSVREN	ADC0 的通道 16 和 17 使能 0：ADC0 的通道 16 和 17 禁止 1：ADC0 的通道 16 和 17 使能
22	SWRCST	常规序列转换开始。 如果 ETSRC 是 111，该位置‘1’开启常规序列转换。软件置位，软件清零，或转换开始后，由硬件清零。
21	保留	必须保持复位值。
20	ETERC	常规序列外部触发使能 0：常规序列外部触发禁止 1：常规序列外部触发使能
19：17	ETSRC[2：0]	常规序列外部触发选择 对于 ADC0 与 ADC1： 000：定时器 0 CH0 001：定时器 0 CH1 010：定时器 0 CH2 011：定时器 1 CH1 100：定时器 2 TRGO 101：定时器 3 CH3 110：中断线 11/定时器 7 TRGO 111：软件触发 对于 ADC2： 000：定时器 2 CH0 001：定时器 1 CH2 010：定时器 0 CH2 011：定时器 7 CH0 100：定时器 7 TRGO 101：定时器 4 CH0 110：定时器 4 CH2 111：软件触发
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10：9	保留	必须保持复位值。
8	DMA	DMA 请求使能 0：DMA 请求禁止 1：DMA 请求使能

7：4	保留	必须保持复位值。
3	RSTCLB	校准复位 软件置位，在校准寄存器初始化后该位硬件清零。 0：校准寄存器初始化结束 1：校准寄存器初始化开始
2	CLB	ADC 校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：禁止连续运行模式 1：使能连续运行模式
0	ADCON	开启 ADC。该位从‘0’变成‘1’将在稳定时间结束后唤醒 ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写‘1’，将开启转换。 0：禁止 ADC 关闭电源 1：使能 ADC

14.7.4. 采样时间寄存器 0（ADC_SAMPT0）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
									rw			rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		
rw	rw			rw			rw			rw			rw		

位/位域	名称	说明
31:24	保留	必须保持复位值。
23:21	SPT17[2:0]	参考 SPT10[2:0]的描述
20:18	SPT16[2:0]	参考 SPT10[2:0]的描述
17:15	SPT15[2:0]	参考 SPT10[2:0]的描述
14:12	SPT14[2:0]	参考 SPT10[2:0]的描述
11:9	SPT13[2:0]	参考 SPT10[2:0]的描述
8:6	SPT12[2:0]	参考 SPT10[2:0]的描述

5:3	SPT11[2:0]	参考 SPT10[2:0]的描述
2:0	SPT10[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为 239.5 周期

14.7.5. 采样时间寄存器 1（ADC_SAMPT1）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]	SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]		
rw	rw			rw			rw			rw			rw		

位/位域	名称	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	SPT2[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

14.7.6. 看门狗高阈值寄存器（ADC_WDHT）

地址偏移：0x24
复位值：0x0000 0FFF

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WDHT[11:0]	模拟看门狗高侧阈值 这些位定义了模拟看门狗的高侧阈值。

14.7.7. 看门狗低阈值寄存器（ADC_WDLT）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

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

14.7.8. 常规序列寄存器 0（ADC_RSQ0）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	说明
31:24	保留	必须保持复位值。
23:20	RL[3:0]	常规序列长度 常规通道转换序列中的总的通道数目为 RL[3:0]+1。
19:15	RSQ15[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考 RSQ0[4:0]的描述

14.7.9. 常规序列寄存器 1（ADC_RSQ1）

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	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

14.7.10. 常规序列寄存器 2（ADC_RSQ2）

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ5[4:0]					RSQ4[4:0]					RSQ3[4:1]			
				rw					rw					rw	

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

位/位域	名称	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..17)写入这些位来选择常规通道的第 n 个转换的通道

14.7.11. 常规数据寄存器（ADC_RDATA）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

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

14.7.12. 过采样控制寄存器（ADC_OVSAMPCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	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	触发过采样 该位通过软件设置和清除。 0：所有的过采样连续转换完成一个触发后 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：当 ADCON= 0 时软件才允许写该位(确定没有转换正在进行)
8:5	OVSS[3:0]	过采样移位 该位通过软件设置和清除。 0000：不移位 0001：移 1 位 0010：移 2 位 0011：移 3 位

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

15. 数模转换器（DAC）

15.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式，左对齐或右对齐模式。当使能了外部触发，DMA 可被用于更新输入端数字数据。在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

两个 DAC 可以独立或并发工作。

15.2. 主要特征

DAC 的主要特征如下：

- 12 位分辨率，数据右对齐或左对齐；；
- 支持 DMA 功能；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 外部参考电压，VREF+；
- 噪声波形(LSFR 噪声模式和三角噪声模式)；
- 双 DAC 并发模式。

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

图 15-1. DAC 结构框图

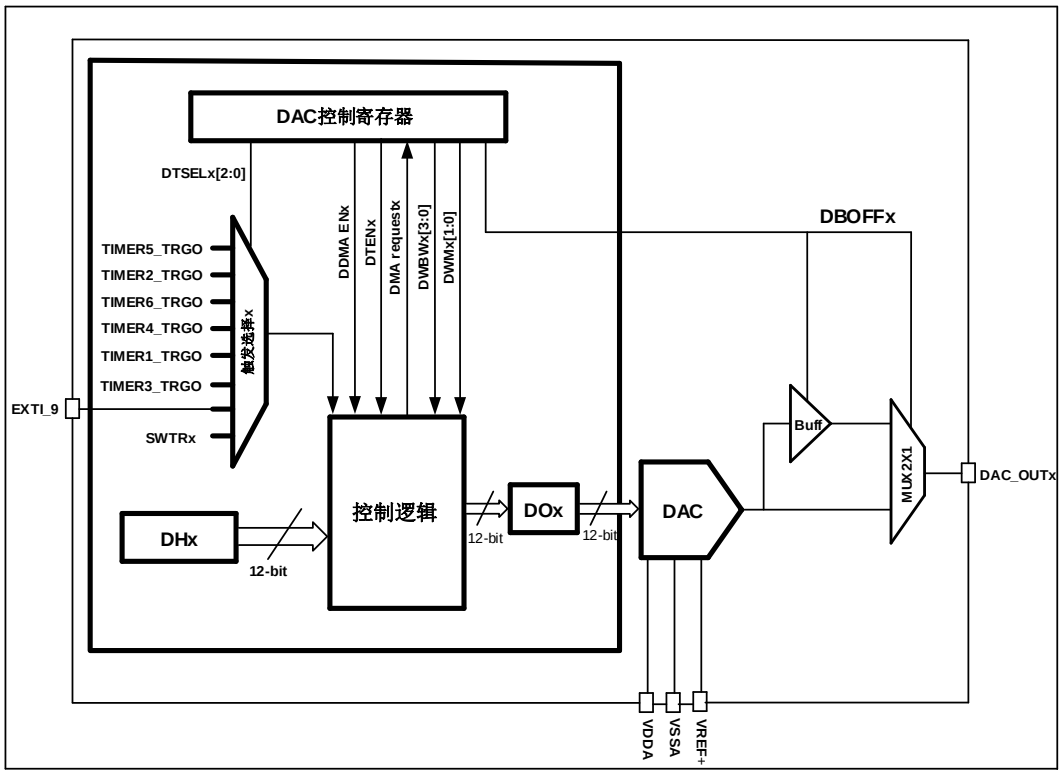


表 15-1. DAC I/O 描述

名称	描述	信号类型
V_{DDA}	模拟电源	电源
V_{SSA}	模拟电源地	电源
V_{REF+}	参考电压	模拟输入
DAC_OUTx	DACx 模拟输出	模拟输出

在使能 DAC 模块前，GPIO 口（PA4 对应 DAC0，PA5 对应 DAC1）应配置为模拟模式。

15.3. 功能说明

15.3.1. DAC 使能

将 DAC_CTL 寄存器中的 DENx 位置 1 可以给 DAC 上电，DAC 子模块完全启动需要等待 t_{WAKEUP}。

15.3.2. DAC 输出缓冲

为了降低输出阻抗并驱动外部负载，每个 DAC 模块内各集成了一个输出缓冲区。

这个输出缓冲区默认情况是开启的，可以通过设置 DAC_CTL 寄存器的 DBOFFx 位来开启或者关闭缓冲区。

15.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据（DACx_DH），可以通过对 DACx_R12DH、DACx_L12DH 和 DACx_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 DACx_R8DH 寄存器时，只有 8 位最高有效位是可被配置的，4 位最低有效位被强制置为 0。

15.3.4. DAC 触发

通过设置 DAC_CTL 寄存器中 DTENx 位来使能 DAC 外部触发。触发源可以通过 DAC_CTL 寄存器中 DTSELx 位来进行选择。如*表 15-2. DAC 外部触发*所示。

表 15-2. DAC 外部触发

DTSELx[2:0]	触发源	触发类型
3b’000	TIMER5_TRGO	硬件触发
3b’001	TIMER2_TRGO	
3b’010	TIMER6_TRGO	
3b’011	TIMER4_TRGO	
3b’100	TIMER1_TRGO	
3b’101	TIMER3_TRGO	
3b’110	EXTI_9	外部信号
3b’111	SWTRIG	软件触发

TIMERx_TRGO 信号是由定时器生成的，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

15.3.5. DAC 工作流程

如果使能了外部触发（通过设置 DAC_CTL 寄存器的 DTENx 位），根据已经选择的触发事件，DAC 保持数据（DACx_DH）会被转移到 DAC 数据输出寄存器（DACx_DO）。否则，在外部触发没有使能的情况下，DAC 保持数据（DACx_DH）会被自动转移到 DAC 数据输出寄存器（DACx_DO）。

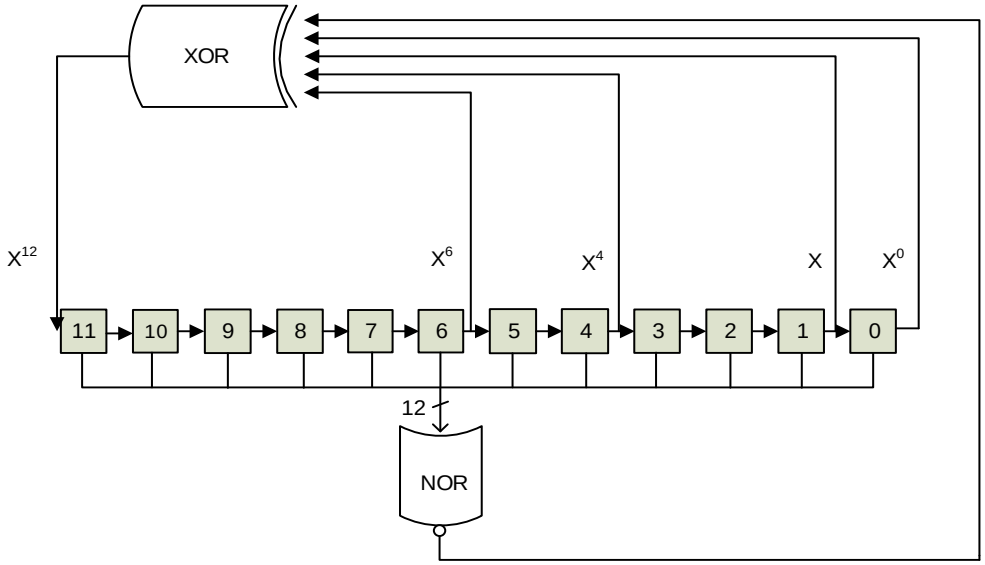
当 DAC 保持数据（DACx_DH）加载到 DACx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

15.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通过 DAC_CTL 寄存器的 DWMx 位来进行选择。噪声的幅值可以通过配置 DAC_CTL 寄存器的 DAC 噪声波位宽（DWBWx）位来进行设置。

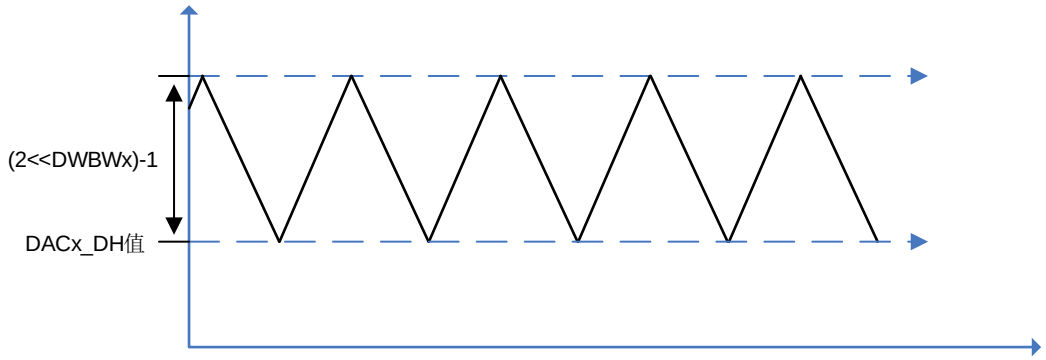
在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在 LFSR 噪声模式下，LFSR 的值与 DACx_DH 值相加后，被写入到 DAC 数据输出寄存器（DACx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，DWBWx 位决定了不屏蔽 LFSR 的哪些位。

图 15-2. DAC LFSR 算法



在三角噪声模式下，三角波信号与 DACx_DH 值相加后，被写入到 DAC 数据输出寄存器（DACx_DO）。三角波幅值的最小值为 0，最大值为(2<<DWBWx)-1。

图 15-3. DAC 三角噪声模式波形



15.3.7. DAC 输出计算

DAC 引脚上的输出电压取决于下面的等式：

$$V_{DACx_out}=V_{REF+}*DAC_DO/4095 \quad (式\ 15\text{-}1)$$

数字输入被线性地转换成模拟输出电压，输出范围为 0 到 V_{REF+}。

15.3.8. DMA 功能

在外部触发使能的情况下，通过设置 DAC_CTL 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），DAC 产生一个 DMA 请求。

15.3.9. DAC 并发转换

当两个 DAC 同时工作时，为了在特定应用中最大限度利用总线带宽，两个 DAC 可以被配置为并发模式。在并发模式中，DACx_DH 和 DACx_DO 的值将被同时更新。

有 3 个寄存器可被用于加载 DACx_DH 的值，分别是：DACC_R8DH、DACC_R12DH 和 DACC_L12DH。只需访问特定的寄存器来实现同时驱动两个 DAC。

当使能了外部触发时，两个 DAC 模块的 DTENx 位都应被置位。DTSEL0 和 DTSEL1 位应被配置为相同的值。

当使能了 DMA 功能时，任一 DAC 的 DDMAENx 位被置位即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

15.4. DAC 寄存器

DAC 基地址：0x4000 7400

15.4.1. 控制寄存器 (DAC_CTL)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			DDMAEN1	DWBW1[3:0]				DWM1[1:0]		DTSEL1[2:0]			DTEN1	DBOFF1	DEN1
			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			DDMAEN0	DWBW0[3:0]				DWM0[1:0]		DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
			rw	rw				rw		rw			rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	DDMAEN1	DAC1 DMA 使能 0：DAC1 DMA 模式失能 1：DAC1 DMA 模式使能
27:24	DWBW1[3:0]	DAC1 噪声波位宽 这些位指定了 DAC1 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为((2<<(n-1))-1)。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 ≥1011：波形信号的位宽为 12
23:22	DWM1[1:0]	DAC1 噪声波模式 这些位指定了在 DAC1 外部触发使能(DTEN1=1)的情况下，DAC1 的噪声波模式的选择。 00：波形生成失能

		01：LFSR 噪声模式 1x：三角噪声模式
21:19	DTSEL1[2:0]	DAC1 触发选择 这些位用于在 DAC1 外部触发使能(DTEN1=1)的情况下，DAC1 外部触发的选择。 000：Timer 5 TRGO 001：Timer 2 TRGO 010：Timer 6 TRGO 011：Timer 4 TRGO 100：Timer 1 TRGO 101：Timer 3 TRGO 110：外部中断线 9 111：软件触发
18	DTEN1	DAC1 触发使能 0：DAC1 触发失能 1：DAC1 触发使能
17	DBOFF1	DAC1 输出缓冲区关闭 0：DAC1 输出缓冲区打开以降低输出阻抗，增大驱动能力 1：DAC1 输出缓冲区关闭
16	DEN1	DAC1 使能 0：DAC1 失能 1：DAC1 使能
15:13	保留	必须保持复位值。
12	DDMAEN0	DAC0 DMA 使能 0：DAC0 DMA 模式失能 1：DAC0 DMA 模式使能
11:8	DWBW0[3:0]	DAC0 噪声波位宽 这些位指定了 DAC0 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为((2<<(n-1))-1)。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11

		≥1011：波形信号的位宽为 12
7:6	DWM0[1:0]	DAC0 噪声波模式 这些位指定了在 DAC0 外部触发使能(DTEN0=1)的情况下，DAC0 的噪声波模式的选择。 00：波形生成失能 01：LFSR 噪声模式 1x：三角噪声模式
5:3	DTSEL0[2:0]	DAC0 触发选择 这些位用于在 DAC0 外部触发使能(DTEN0=1)的情况下，DAC0 外部触发的选择。 000：Timer 5 TRGO 001：Timer 2 TRGO 010：Timer 6 TRGO 011：Timer 4 TRGO 100：Timer 1 TRGO 101：Timer 3 TRGO 110：外部中断线 9 111：软件触发
2	DTEN0	DAC0 触发使能 0：DAC0 触发失能 1：DAC0 触发使能
1	DBOFF0	DAC0 输出缓冲区关闭 0：DAC0 输出缓冲区打开以降低输出阻抗，增大驱动能力 1：DAC0 输出缓冲区关闭
0	DEN0	DAC0 使能 0：DAC0 失能 1：DAC0 使能

15.4.2. 软件触发寄存器 (DAC_SWT)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														SWTR1	SWTR0
														w	w

位/位域	名称	描述

31:2	保留	必须保持复位值。
1	SWTR1	DAC1 软件触发，由硬件清除 0: 软件触发失能 1: 软件触发使能
0	SWTR0	DAC0 软件触发，由硬件清除 0: 软件触发失能 1: 软件触发使能

15.4.3. DAC0 12 位右对齐数据保持寄存器 (DAC0_R12DH)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAC0_DH[11:0]											
				rw											

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DAC0_DH[11:0]	DAC0 12 位右对齐数据 这些位指定了数据将由 DAC0 转换。

15.4.4. DAC0 12 位左对齐数据保持寄存器 (DAC0_L12DH)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC0_DH[11:0]												保留			
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:4	DAC0_DH[11:0]	DAC0 12 位左对齐数据 这些位指定了数据将由 DAC0 转换。
3:0	保留	必须保持复位值。

15.4.5. DAC0 8 位右对齐数据保持寄存器 (DAC0_R8DH)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DAC0_DH[7:0]							
											rw				

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	DAC0_DH[7:0]	DAC0 8 位右对齐数据 这些位指定了数据的 8 位最高有效位将由 DAC0 转换。

15.4.6. DAC1 12 位右对齐数据保持寄存器 (DAC1_R12DH)

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				DAC1_DH[11:0]											
										rw					

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DAC1_DH[11:0]	DAC1 12 位右对齐数据 这些位指定了数据将由 DAC1 转换。

15.4.7. DAC1 12 位左对齐数据保持寄存器 (DAC1_L12DH)

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
DAC1_DH[11:0]												保留			
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	DAC1_DH[11:0]	DAC1 12 位左对齐数据 这些位指定了数据将由 DAC1 转换。
3:0	保留	必须保持复位值。

15.4.8. DAC1 8 位右对齐数据保持寄存器 (DAC1_R8DH)

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								DAC1_DH[7:0]							
								rw							

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	DAC1_DH[7:0]	DAC1 8 位右对齐数据 这些位指定了数据的 8 位最高有效位将由 DAC1 转换。

15.4.9. DAC 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DAC1_DH[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAC0_DH[11:0]											
				rw											

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	DAC1_DH[11:0]	DAC1 12 位右对齐数据 这些位指定了数据将由 DAC1 转换。
15:12	保留	必须保持复位值。
11:0	DAC0_DH[11:0]	DAC0 12 位右对齐数据 这些位指定了数据将由 DAC0 转换。

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

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAC1_DH[11:0]												保留			
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC0_DH[11:0]												保留			
rw															

位/位域	名称	描述
31:20	DAC1_DH[11:0]	DAC1 12 位左对齐数据 这些位指定了数据将由 DAC1 转换。
19:16	保留	必须保持复位值。
15:4	DAC0_DH[11:0]	DAC0 12 位左对齐数据 这些位指定了数据将由 DAC0 转换。
3:0	保留	必须保持复位值。

15.4.11. DAC 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC1_DH [7:0]								DAC0_DH [7:0]							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	DAC1_DH[7:0]	DAC1 8 位右对齐数据 这些位指定了数据的 8 位最高有效位将由 DAC1 转换。
7:0	DAC0_DH[7:0]	DAC0 8 位右对齐数据 这些位指定了数据的 8 位最高有效位将由 DAC0 转换。

15.4.12. DAC0 数据输出寄存器 (DAC0_DO)

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
保留				DAC0_DO [11:0]											
				r											

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DAC0_DO [11:0]	DAC0 数据输出 这些位为只读类型，存储由 DAC0 转换的数据。

15.4.13. DAC1 数据输出寄存器 (DAC1_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				DAC1_DO [11:0]											
				r											

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DAC1_DO [11:0]	DAC1 数据输出 这些位为只读类型，存储由 DAC1 转换的数据。

16. 看门狗定时器（WDGT）

看门狗定时器（WDGT）是一个硬件计时电路，用来监测由软件故障导致的系统故障。片上有两个看门狗定时器外设，独立看门狗定时器（FWDGT）和窗口看门狗定时器（WWDGT）。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到了预设的门限的时候，会触发一个复位。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

16.1. 独立看门狗定时器（FWDGT）

16.1.1. 简介

独立看门狗定时器（FWDGT）有独立的时钟源（IRC40K）。即使主时钟失效，FWDGT依然能保持正常工作状态，适用于需要独立环境且对计时精度要求不高的场合。

当内部向下计数器的计数值达到0，独立看门狗会产生一个系统复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

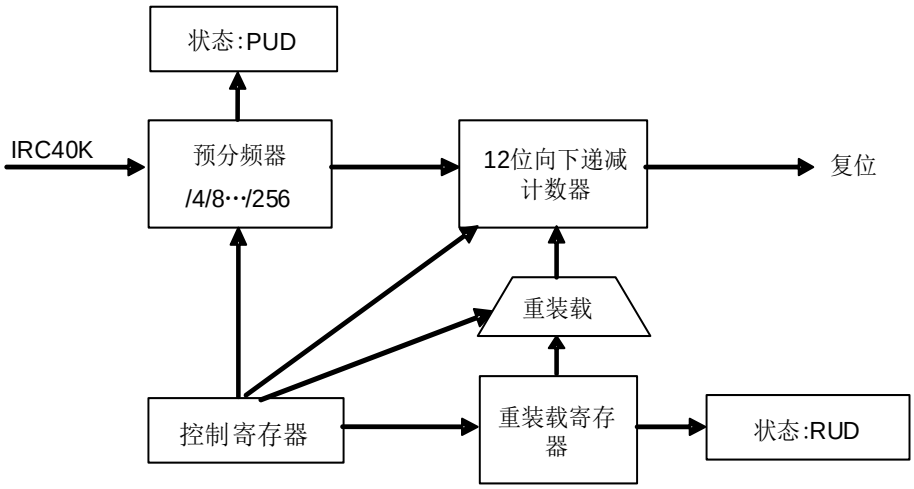
16.1.2. 主要特性

- 独立运行的12位向下计数器；
- 使能看门狗定时器，当向下计数器的值达到0时产生系统复位；
- 独立时钟源，独立看门狗定时器在主时钟故障（例如待机和深度睡眠模式下）时仍能工作；
- 选项字节中的独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

16.1.3. 功能描述

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。参考*图16-1. 独立看门狗定时器框图*为独立看门狗定时器的功能模块。

图 16-1. 独立看门狗定时器框图



向控制寄存器（FWDGT_CTL）中写0xCCCC可以开启独立看门狗定时器，计数器开始向下计数。当计数器记到0x000，产生一次系统复位。

在任何时候向FWDGT_CTL中写0xAAAA都可以重装载计数器，重装载值来源于，重装载寄存器（FWDGT_RLD）。软件可以在计数器计数值达到0x000之前可以通过重装载计数器来阻止看门狗定时器产生系统复位。

如果在选项字节中打开了“硬件看门狗定时器”功能，那么在上电的时候看门狗定时器就被自动打开。为了避免系统复位，软件应该在计数器达到0x000之前重装载计数器。

预分频寄存器（FWDGT_PSC）和FWDGT_RLD寄存器都有写保护功能。在写数据到这些寄存器之前，需要写0x5555到FWDGT_CTL中。写其他任何值到FWDGT_CTL中将会再次启动对这些寄存器的写保护。当FWDGT_PSC或者FWDGT_RLD更新时，FWDGT_STAT寄存器的相应状态位会被置1。

如果DBG控制寄存器0（DBG_CTL0）中的FWDGT_HOLD位被清0，即使Cortex®-M3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 16-1. 独立看门狗定时器在 40kHz（IRC40K）时的最小/最大超时周期

预分频系数	PSC[2:0]位	最小超时(ms) RLD[11:0]=0x000	最大超时(ms) RLD[11:0]=0xFFF
1 / 4	000	0.025	409.525
1 / 8	001	0.025	819.025
1 / 16	010	0.025	1638.025
1 / 32	011	0.025	3276.025
1 / 64	100	0.025	6552.025
1 / 128	101	0.025	13104.025
1 / 256	110 or 111	0.025	26208.025

通过校准IRC40K可以使独立看门狗定时器超时更加精确。

注意：当执行完喂狗reload操作之后，如需要立即进入deepsleep / standby模式时，必须通过

软件设置，在reload命令及deepsleep / standby模式命令中间插入3个以上IRC40K时钟间隔。

16.1.4. FWDGT 寄存器

FWDGT基地址：0x4000 3000

控制寄存器（FWDGT_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

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

预分频寄存器（FWDGT_PSC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	说明
31:3	保留	必须保持复位值。
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。

000：1 / 4
001：1 / 8
010：1 / 16
011：1 / 32
100：1 / 64
101：1 / 128
110：1 / 256
111：1 / 256

如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零（进入省电模式前需要等待PUD值清零）。

重装载寄存器（FWDGT_RLD）

地址偏移：0x08
复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

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

状态寄存器（FWDGT_STAT）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	说明
31:2	保留	必须保持复位值。
1	RUD	独立看门狗定时器计数器重装载值更新。 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。
0	PUD	独立看门狗定时器预分频值更新。 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。

16.2. 窗口看门狗定时器（WWDGT）

16.2.1. 简介

窗口看门狗定时器（WWDGT）用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，7位向下递减计数器值逐渐减小。计数值达到0x3F时会产生系统复位（CNT[6]位被清0）。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生系统复位。因此软件需要在给定的区间内更新计数器。窗口看门狗定时器在计数器计数值达到0x40，会产生一个提前唤醒标志，如果使能中断将会产生提前唤醒中断。

窗口看门狗定时器时钟是由APB1时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

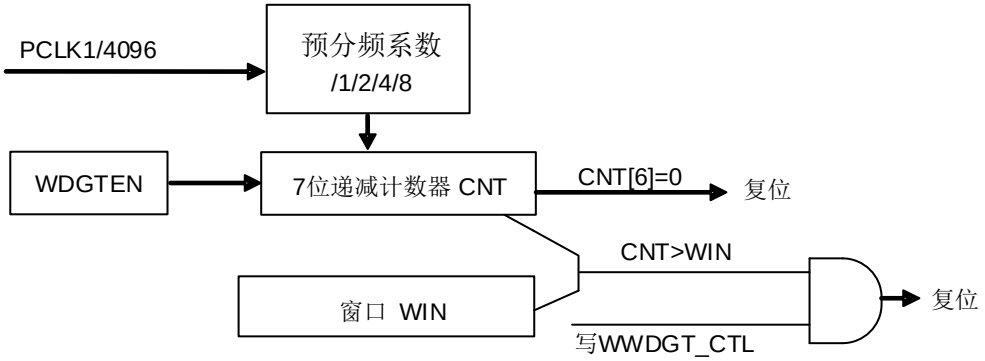
16.2.2. 主要特性

- 可编程的7位向下递减计数器；
- 当窗口看门狗使能后，有以下两种情况会产生复位：
  - 当计数器达到0x3F时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位；
- 提前唤醒中断（EWI）：看门狗定时器打开，中断使能，计数值达到0x40时会产生中断；
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

16.2.3. 功能描述

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的时候产生系统复位（CNT[6]位被清0），或是在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 16-2. 窗口看门狗定时器框图



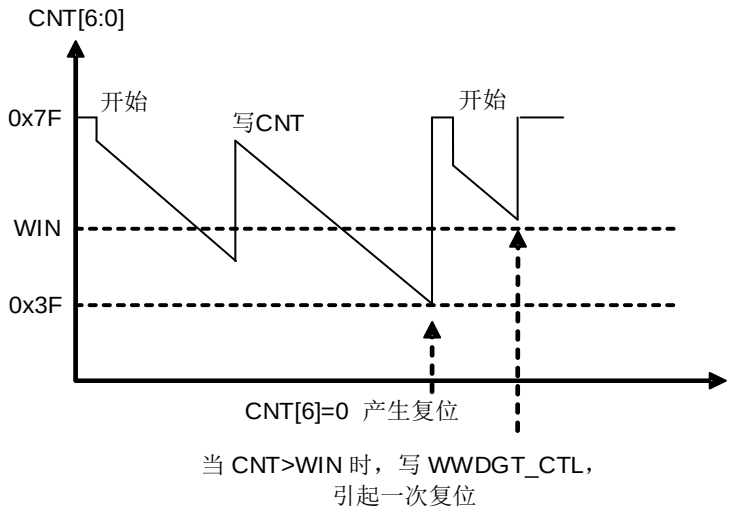
上电复位之后窗口看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB1时钟和预分频器（WWDGT_CFG寄存器的PSC[1:0]位）。

配置寄存器（WWDGT_CFG）中的WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，且大于0x3F的时候，重装载向下计数器可以避免复位，否则在其他时候进行重加载就会引起复位。

对WWDGT_CFG寄存器的EWIE位置1可以使能提前唤醒中断（EWI），当计数值达到0x40的时候该中断产生。同时可以用相应的中断服务程序（ISR）来触发特定的行为（例如通信或数据记录），来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，在ISR中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将WWDGT_STAT寄存器的EWIF位写0可以清除EWI中断。

图 16-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

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

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK1}：APB1以ms为单位的时钟周期

t_{WWDGT}的最大值和最小值请参考*表16-2. 在60MHz（fPCLK1时的最大/最小超时值*。

表 16-2. 在 60MHz（f_{PCLK1} 时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[6:0] =0x40	最大超时 CNT[6:0]=0x7F
1/1	00	68.2 μs	4.3ms
1/2	01	136.4 μs	8.6 ms
1/4	10	272.8μs	17.2 ms
1/8	11	545.6 μs	34.4 ms

调试模式下，如果DBG中的WWDGT_HOLD位被清0，即使Cortex®-M3内核停止工作，窗口看门狗定时器也可以继续工作。当WWDGT_HOLD位被置1时，窗口看门狗定时器会随着内核停止工作而停止计数。

16.2.4. WWDGT 寄存器

WWDGT基地址：0x4000 2C00

控制寄存器（WWDGT_CTL）

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

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

配置寄存器（WWDGT_CFG）

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。

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

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

位/位域	名称	说明
31:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，或通过RCU模块的WWDGTRST位进行软件复位。写0没有任何作用。

8:7	PSC[1:0]	预分频器，看门狗定时器计数器的时间基准 00：PCLK1 / 4096 / 1 01：PCLK1 / 4096 / 2 10：PCLK1 / 4096 / 4 11：PCLK1 / 4096 / 8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生系统复位。

状态寄存器（WWDGT_STAT）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	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bit可以通过写0清零，写1无效。

17. 实时时钟（RTC）

17.1. 简介

实时时钟RTC通常被用作时钟日历。RTC电路分属于两个电源域：备份域和VDD电源域。位于备份域中的RTC电路，包含一个32位的累加计数器、一个闹钟、一个预分频器、一个分频器以及RTC时钟配置寄存器，这意味着当系统复位或者从待机模式唤醒时，RTC的设置和时间都保持不变。位于VDD电源域的RTC电路只包括APB接口以及一组控制寄存器。在本章接下来的部分，将详细介绍RTC的功能。

17.2. 主要特性

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

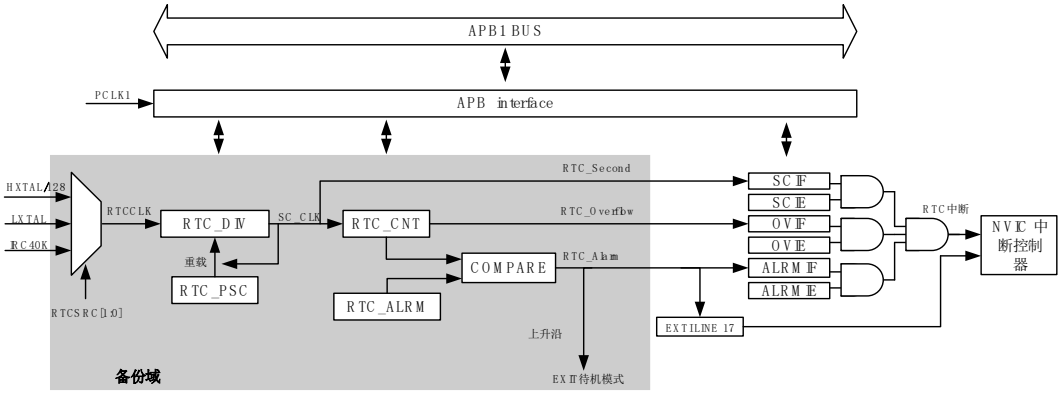
17.3. 功能描述

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

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

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

图 17-1. RTC 框图



17.3.1. RTC 复位

APB接口和RTC_INTEN寄存器会随着系统复位进行复位。RTC内核（预分频器、分频器、计数器以及闹钟）只会随备份域复位进行复位。

通过下面的步骤，可以在复位后访问备份域寄存器以及RTC寄存器：

1. 通过对RCU_APB1EN寄存器中的PMUEN和BKPIEN位进行置位，以使能电源以及备份接口时钟。
2. 通过对PMU_CTL中的BKPWEN位进行置位，以使能对备份域寄存器和RTC的访问。

17.3.2. RTC 读取

APB接口和RTC内核分属于两个不同电源域。

在RTC内核中，只有计数器和分频器寄存器为可读寄存器。这两个寄存器的值以及RTC标志会在每个RTC时钟的上升沿进行内部更新，并与APB1时钟进行重新同步。

当APB接口从禁用状态使能后，建议不要立即进行读操作，因为这些寄存器的首次内部更新可能尚未完成。这表明，在系统复位、电源复位、从待机/深度睡眠模式下唤醒时，APB接口是被禁用的，但是RTC内核仍然保持运行。在这类情况下，正确的读操作应该先将RTC_CTL寄存器的RSYNF清零并等待其被硬件置位。此时，WFI和WFE指令对于RTC的APB接口没有影响。

17.3.3. RTC 配置

RTC内核中的RTC_PSC、RTC_CNT和RTC_ALRM寄存器都是可写的。只有在外设进入配置模式后，这些寄存器的值才能进行设置。通过查询RTC_CTL寄存器的CMF位，可以检测配置模式的状态。只有在外设退出配置模式后，之前对这些寄存器的写操作才能生效，且至少需要三个RTCCLK周期才能完成。在写操作完成后，RTC_CTL寄存器中的LWOFF位的值变为‘1’。下一个写操作必须等待上次写操作完成之后才能进行。

配置过程如下：

1. 等待RTC_CTL寄存器中的LWOFF位的值变为1；
2. 通过将RTC_CTL寄存器中的CMF置位来进入配置模式；

3. 对RTC寄存器进行写操作；
4. 通过将RTC_CTL寄存器中的CMF清零来退出配置模式；
5. 等待RTC_CTL寄存器中的LWOFF位的值变为1。

17.3.4. RTC 标志位

RTC秒中断标志（SCIF）在RTC计数器更新之前的RTCCLK周期进行置位。

RTC闹钟中断标志（ALRMIF）在计数器达到存储于闹钟寄存器中的RTC闹钟值加1的前一个RTCCLK周期进行置位。

RTC溢出中断标志（OVIF）在计数器值达到0x00的前一个RTCCLK周期置位。

RTC闹钟的写操作需要按照下列方法之一来进行，以达到和秒中断标志保持同步：

1. 启用RTC闹钟中断，在RTC内部中断服务程序内更新RTC闹钟寄存器与/或RTC计数器寄存器的数值。
2. RTC闹钟寄存器及/或RTC计数器寄存器的内容必须等待RTC控制寄存器中SCIF置位后才能更新。

图 17-2. RTC 秒信号及闹钟信号的波形（RTC_PSC = 3,RTC_ALRM = 2）

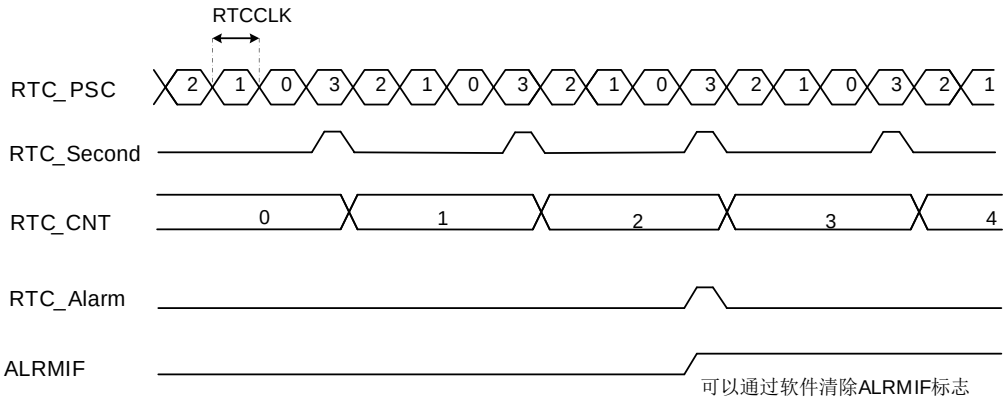
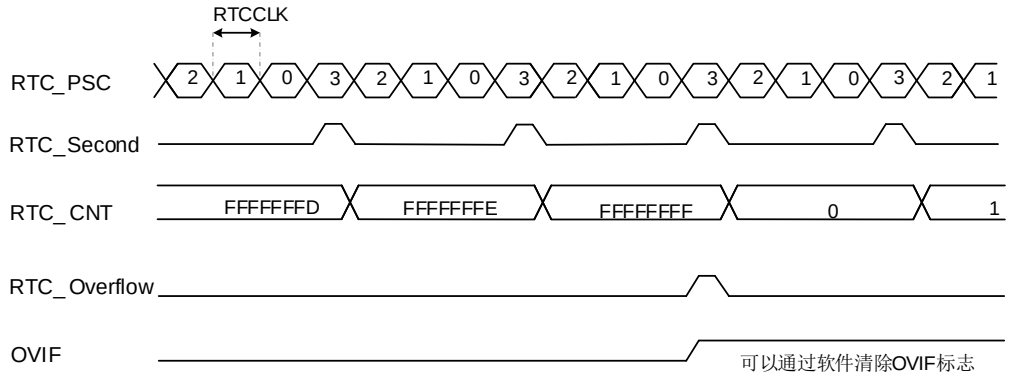


图 17-3. RTC 秒信号及溢出信号的波形（RTC_PSC = 3）



17.4. RTC 寄存器

RTC 基地址：0x4000 2800

17.4.1. RTC 中断使能寄存器（RTC_INTEN）

偏移地址：0x00
复位值：0x0000

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

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

17.4.2. RTC 控制寄存器（RTC_CTL）

偏移地址：0x04
复位值：0x0020

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	LWOFF	上次对RTC寄存器写操作标志 0：上次对RTC寄存器写操作没有完成 1：上次对RTC寄存器写操作已经完成
4	CMF	配置模式标志 0：退出配置模式 1：进入配置模式
3	RSYNF	寄存器同步标志 0：寄存器没有与APB1时钟同步 1：寄存器已经与APB1时钟同步
2	OVIF	溢出中断标志 0：没有检测到溢出事件 1：检测到溢出事件。当RTC_INTEN寄存器的OVIE位被置1，中断发生。
1	ALRMIF	闹钟中断标志 0：没有检测到闹钟事件 1：检测到闹钟事件。当RTC_INTEN寄存器的ALRMIE位被置1，RTC全局中断发生。并且当EXTI17被使能中断模式，发生RTC闹钟中断。
0	SCIF	秒中断标志 0：没有检测到秒事件 1：检测到秒事件。当RTC_INTEN寄存器的SCIE位被置1，中断发生。 当分频器重加载RTC_PSC值时，硬件将该位置1，从而累加RTC计数器。

17.4.3. RTC 预分频寄存器高位（RTC_PSCH）

偏移地址：0x08
复位值：0x0000

该寄存器可以按半字（16 位）或字（32 位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												PSC[19:16]			
														w	

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

17.4.4. RTC 预分频寄存器低位（RTC_PSCL）

偏移地址：0x0C
复位值：0x8000

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

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

17.4.5. RTC 分频器高位（RTC_DIVH）

偏移地址：0x10
复位值：0x0000

该寄存器可以按半字（16 位）或字（32 位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												DIV[19:16]			
														r	

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

17.4.6. RTC 分频器低位（RTC_DIVL）

偏移地址：0x14
复位值：0x8000

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DIV[15:0]	RTC分频器低位 当RTC预分频寄存器或者RTC计数寄存器更新时，RTC分频器寄存器会由硬件自动加载

17.4.7. RTC 计数寄存器高位（RTC_CNTH）

偏移地址：0x18
复位值：0x0000

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

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

17.4.8. RTC 计数寄存器低位（RTC_CNTL）

偏移地址：0x1C
复位值：0x0000

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

位/位域	名称	描述

31:16	保留	必须保持复位值
15:0	CNT[15:0]	RTC计数寄存器低位

17.4.9. RTC 闹钟寄存器高位（RTC_ALRMH）

偏移地址：0x20
复位值：0xFFFF

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

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

17.4.10. RTC 闹钟寄存器低位（RTC_ALRML）

偏移地址：0x24
复位值：0xFFFF

该寄存器可以按半字（16 位）或字（32 位）访问

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

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

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

18. 定时器（TIMER）

表 18-1. 定时器（TIMERx）分为五种类型

定时器	定时器 0/7	定时器 1/2/3/4	定时器 8/11	定时器 9/10/12/13	定时器 5/6
类型	高级	通用（L0）	通用（L1）	通用（L2）	基本
预分频器	16 位	16 位	16 位	16 位	16 位
计数器	16 位	16 位	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，向下，中央对齐	向上，向下，中央对齐	只有向上
可重复性	●	×	×	×	×
捕获/比较通道数	4	4	2	1	0
互补和死区时间	●	×	×	×	×
中止输入	●	×	×	×	×
单脉冲	●	●	●	×	●
正交译码器	●	●	×	×	×
主-从管理	●	●	●	×	×
内部连接	●(1)	●(2)	●(3)	×	TRGO TO DAC
DMA	●	●	×	×	●(4)
Debug 模式	●	●	●	●	●

(1) TIMER0 **ITI0:**TIMER4_TRGO **ITI1：**IMER1_TRGO **ITI2：**TIMER2_TRGO **ITI3：**TIMER3_TRGO

TIMER7 **ITI0：**TIMER0_TRGO **ITI1：**IMER1_TRGO **ITI2：**TIMER3_TRGO **ITI3：**TIMER4_TRGO

(2) TIMER1 **ITI0：**TIMER0_TRGO **ITI1：**参考注释(5) **ITI2：**TIMER2_TRGO **ITI3：**TIMER3_TRGO

TIMER2 **ITI0：**TIMER0_TRGO **ITI1：**TIMER1_TRGO **ITI2：**TIMER4_TRGO **ITI3：**TIMER3_TRGO

TIMER3 **ITI0：**TIMER0_TRGO **ITI1：**TIMER1_TRGO **ITI2：**TIMER2_TRGO **ITI3：**TIMER7_TRGO

TIMER4 **ITI0：**TIMER1_TRGO **ITI1：**TIMER2_TRGO **ITI2：**TIMER3_TRGO **ITI3：**TIMER7_TRGO

(3) TIMER8 **ITI0：**TIMER1_TRGO **ITI1：**TIMER2_TRGO **ITI2：**TIMER9_TRGO **ITI3：**TIMER10_TRGO

TIMER11 **ITI0:** TIMER3_TRGO **ITI1：**TIMER4_TRGO **ITI2：**TIMER12_TRGO **ITI3：**TIMER13_TRGO

(4) 只有更新事件可以产生 DMA 请求，定时器 5 和定时器 6 中没有 DMA 配置寄存器。

(5) TIMER1 的 ITI1 由 ***AFIO 端口配置寄存器 0（AFIO_PCF0）*** 中的 TIMER1ITI1_REMAP 位来决定内部连接的信号源；

18.1. 高级定时器（TIMERx, x=0,7）

18.1.1. 简介

高级定时器（TIMER0/7）是四通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。高级定时器含有一个16位无符号计数器。

高级定时器是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是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。

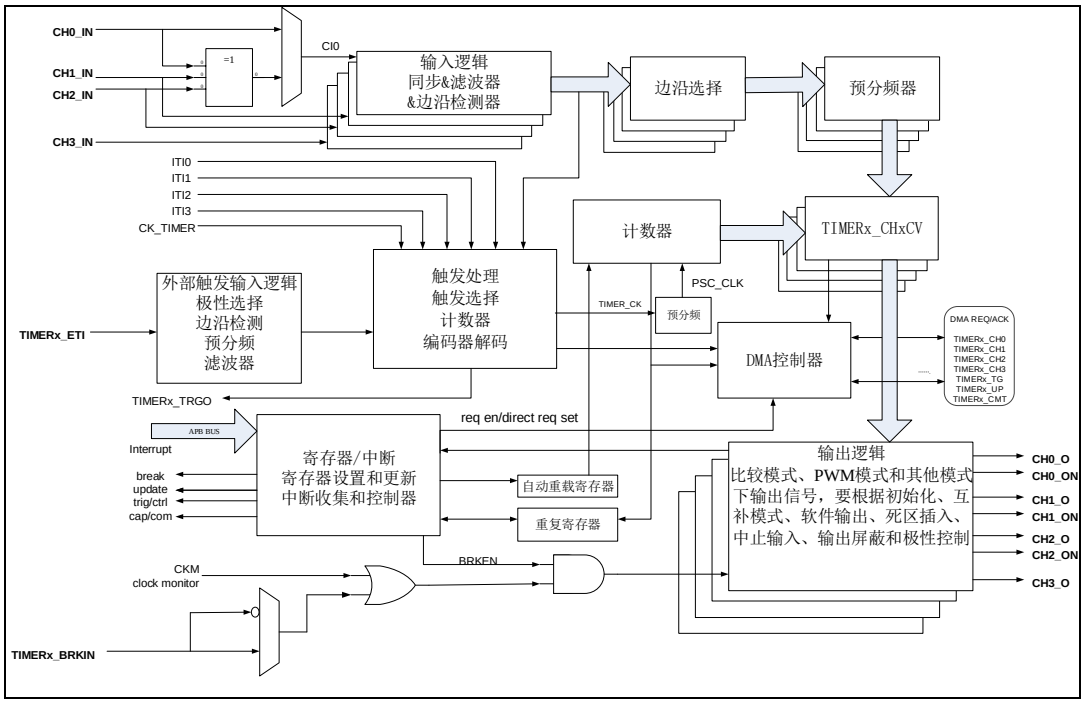
18.1.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央对齐计数；
- 正交编码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能使定时器按照用户的配置输出；
- 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；
- 多个定时器的菊型链接使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

18.1.3. 结构框图

*图 18-1. 高级定时器结构框图*提供了高级定时器的内部配置细节

图 18-1. 高级定时器结构框图



18.1.4. 功能说明

时钟源配置

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

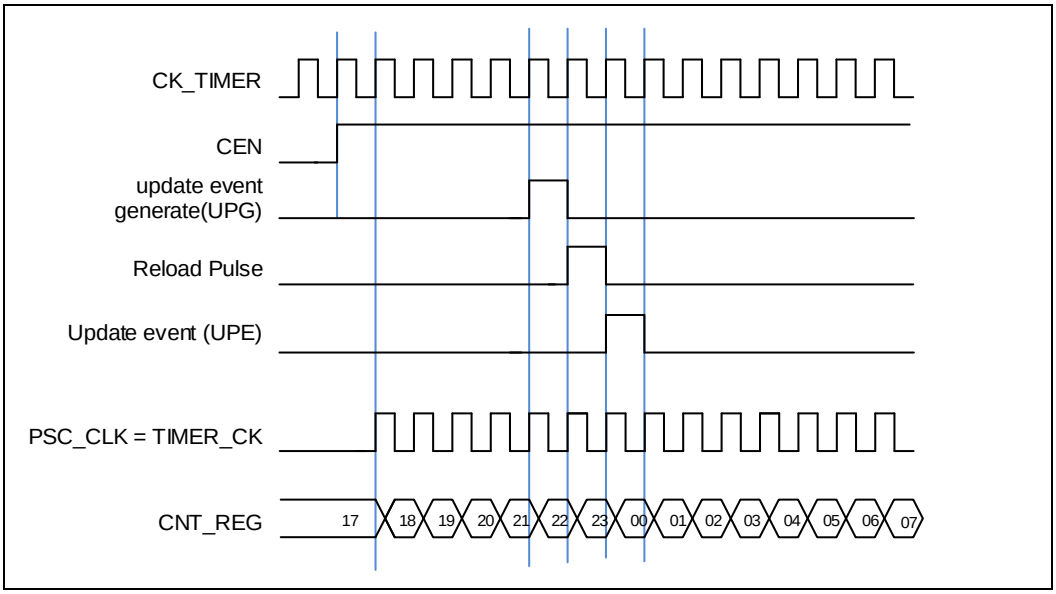
- SMC[2:0]==3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源CK_TIMER。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

这种模式下，驱动预分频器计数的TIMER_CK等于来自于RCU模块的CK_TIMER。

如果将TIMERx_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7，预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动，在下文说明。当SMC位被设置为0x4、0x5和0x6，计数器预分频器时钟源由内部时钟CK_TIMER驱动。

图 18-2. 内部时钟分频为 1 时，计数器的时序图



■ SMC[2:0]==3’b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在TIMERx_CI0/ TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7的同时设置TRGS [2:0]为0x4、0x5或0x6来选择。

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

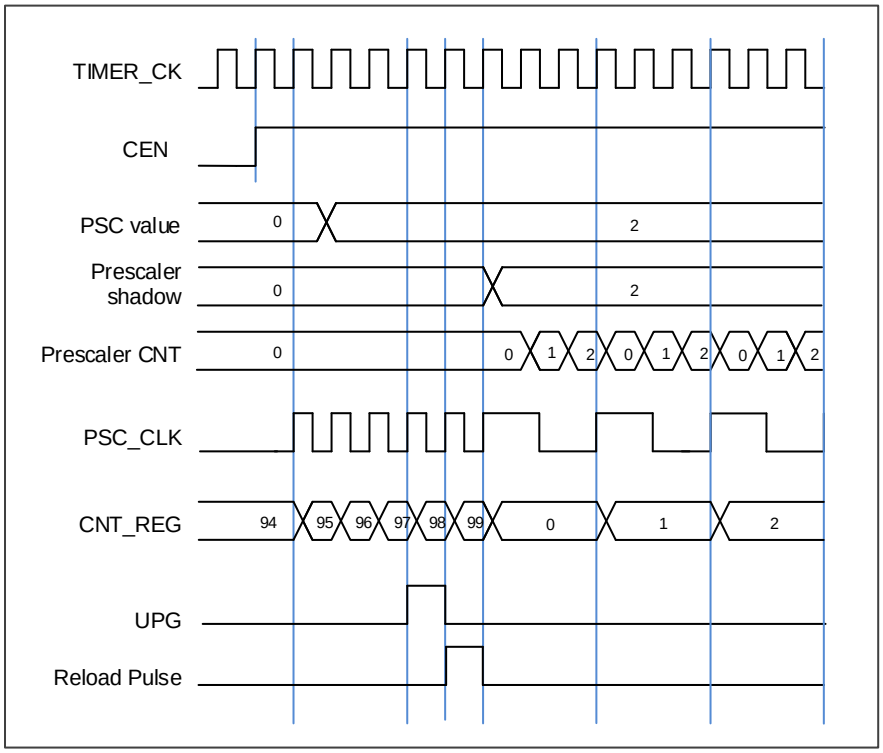
■ SMC1==1’b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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

时钟预分频器

预分频器可以将定时器的时钟(TIMER_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 18-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到计数器自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数，并且产生上溢事件。另外，在(TIMERx_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-4. 向上计数时序图，PSC=0/2*和*图18-5. 向上计数时序图，在运行时改变TIMERx_CAR寄存器的值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同预分频因子下的行为。

图 18-4. 向上计数时序图，PSC=0/2

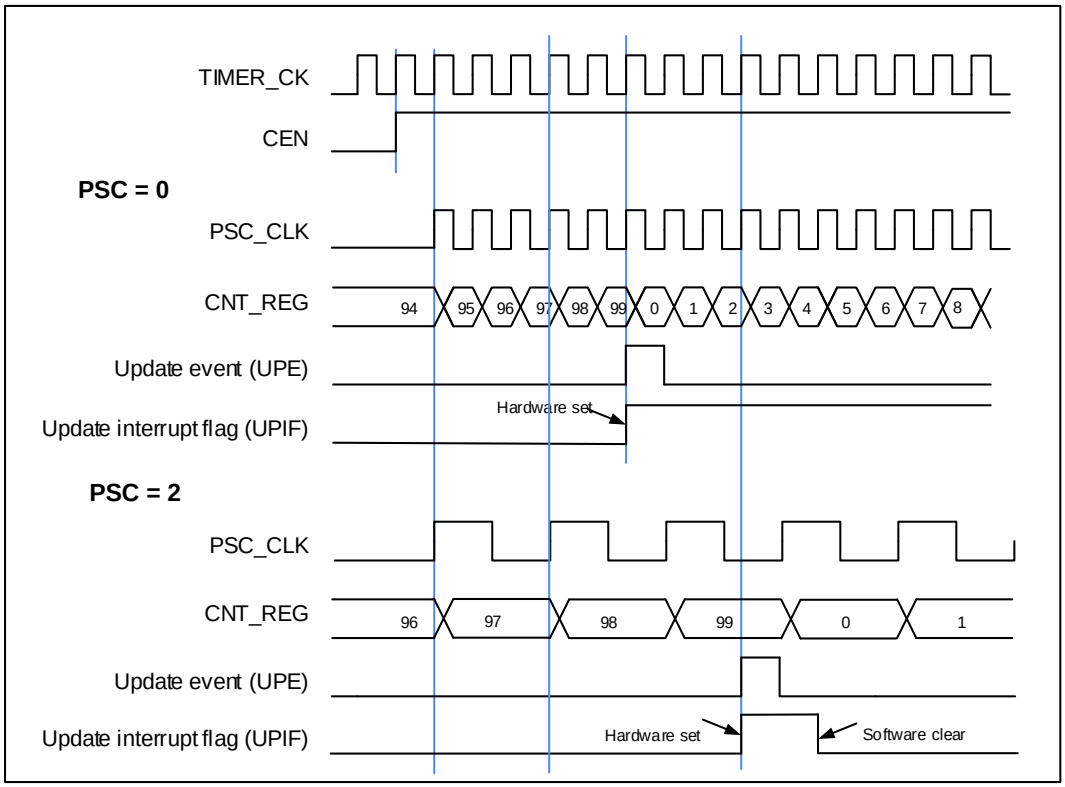
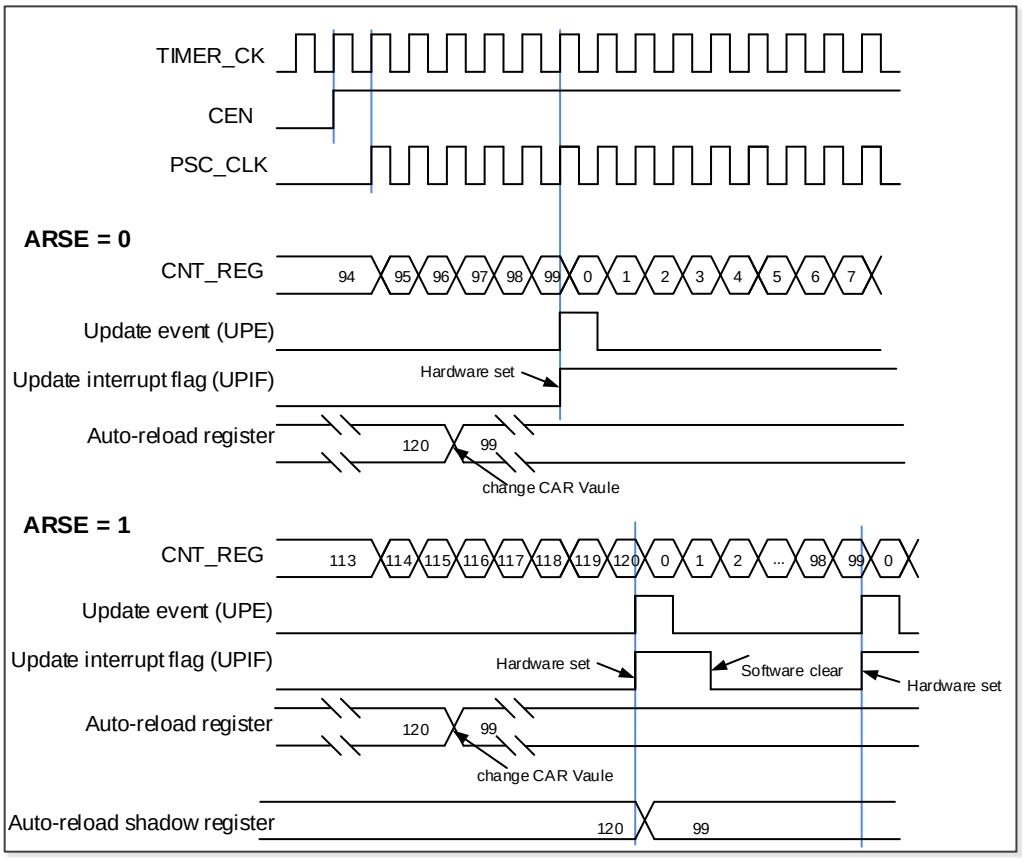


图 18-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数并且产生下溢事件。另外，在(TIMERx_CREP+1)次下溢后将会产生更新事件。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

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

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-6. 向下计数时序图，PSC=0/2*和*图18-7. 向下计数时序图，在运行时改变TIMERx_CAR寄存器值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同时钟频率下的行为。

图 18-6. 向下计数时序图，PSC=0/2

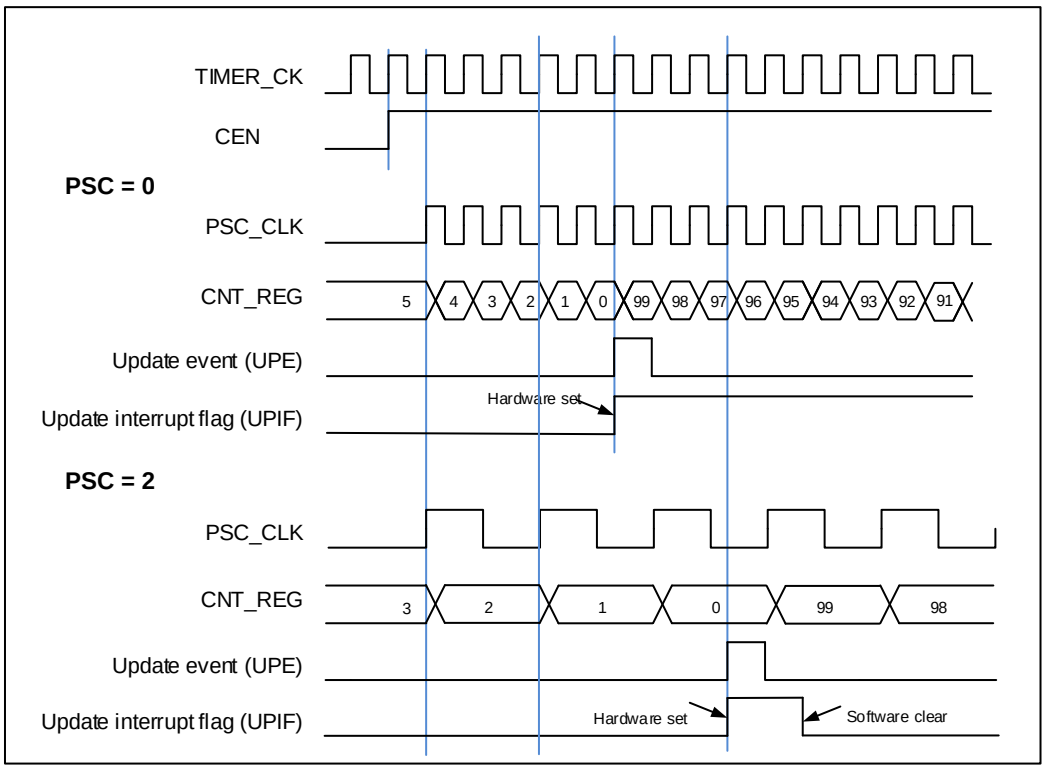
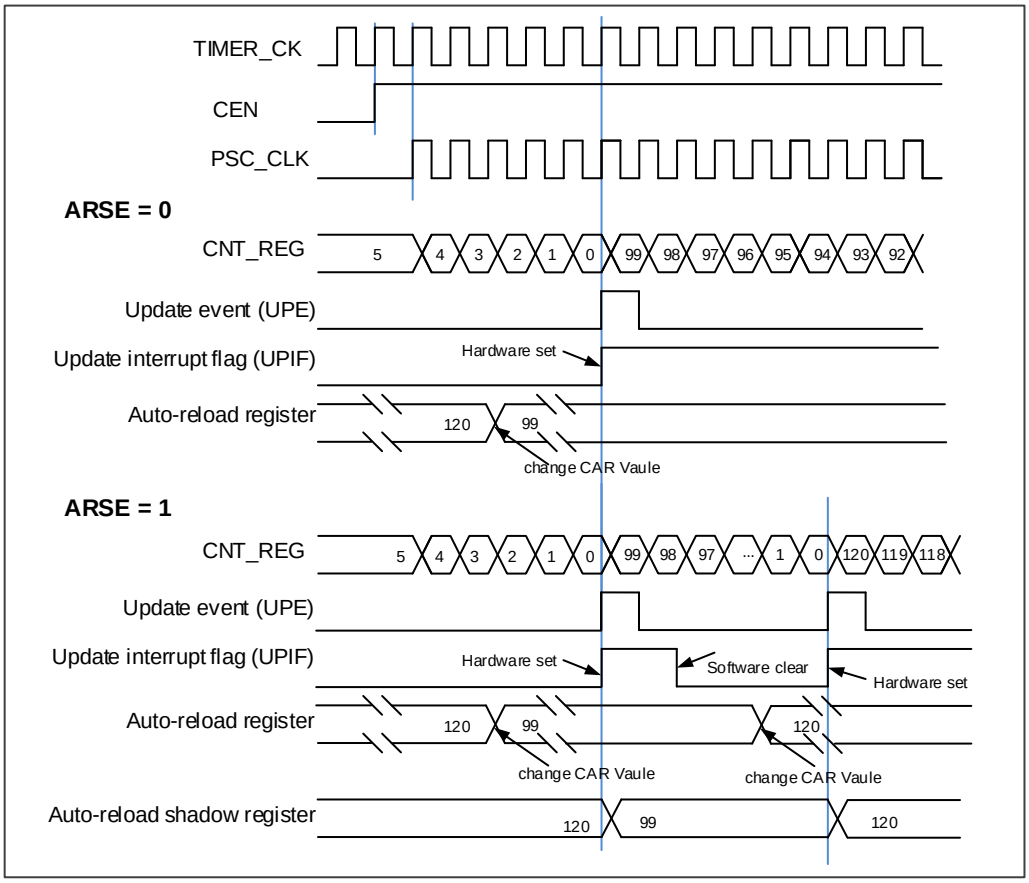


图 18-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

在中央对齐计数模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。向上计数模式中，定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央对齐计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，指示了当前的计数方向。

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

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

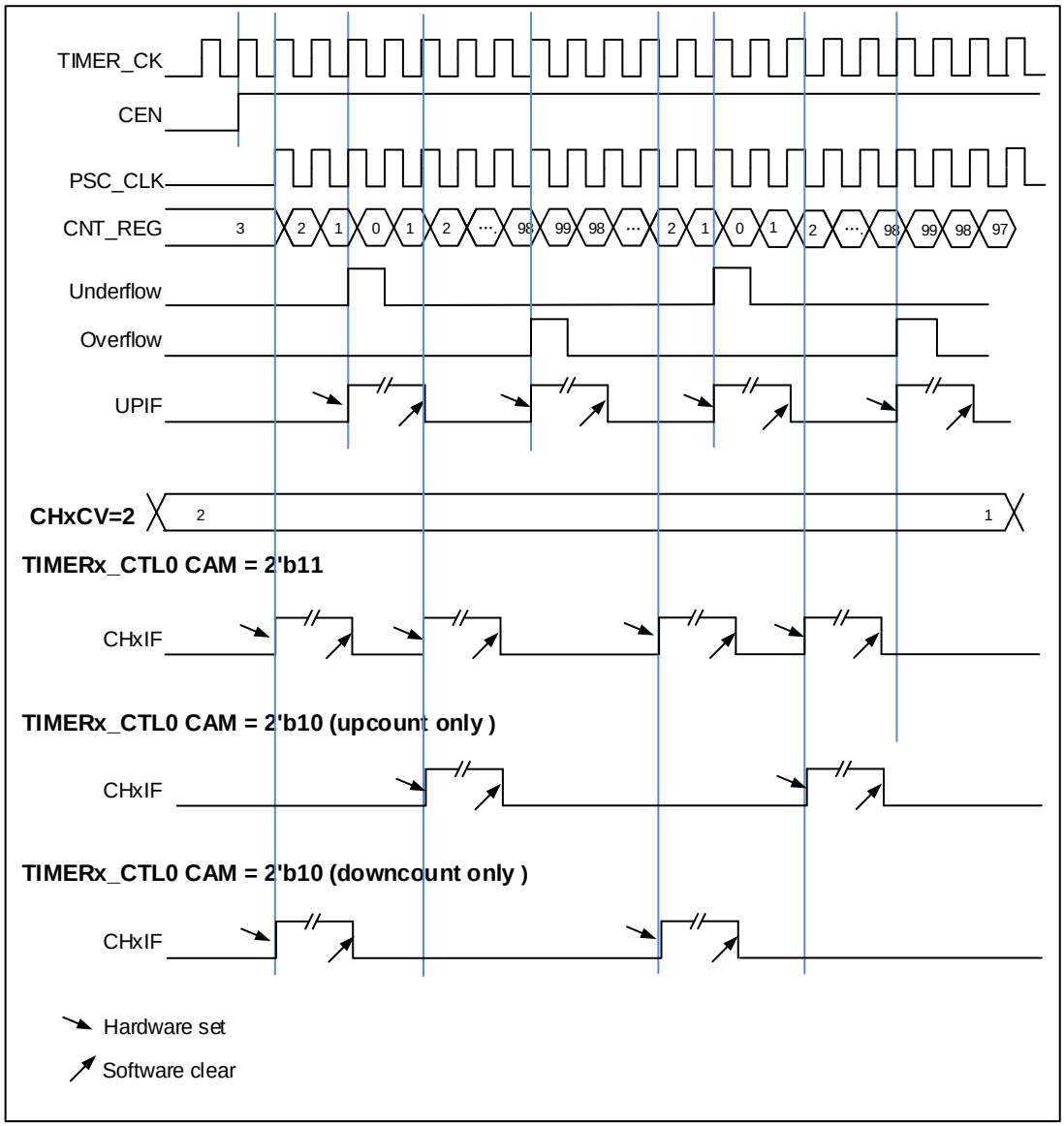
如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 18-8. 中央对齐计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，

TIMERx_PSC=0x0 时，计数器的行为。

图 18-8. 中央对齐计数模式计数器时序图



更新事件（来自上溢/下溢）频率配置

更新事件的生成频率（来自上溢和下溢事件）可以通过TIMERx_CREP寄存器进行配置。重复计数器是用来在N+1个计数周期之后产生更新事件，更新定时器的寄存器，N为TIMERx_CREP寄存器的CREP。重复计数器在每次计数器上溢和下溢时递减（向上计数模式中不存在下溢事件；向下计数模式中不存在上溢事件）。

将TIMERx_SWEVG寄存器的UPG位置1可以重载TIMERx_CREP寄存器中CREP的值并产生一个更新事件。

新写入的CREP值将在下一次更新事件到来时生效。当CREP的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的CREP值何时生效。如果在写入奇数到CREP寄存器后由软件生成更新事件，则在下溢时产生更新事件。如果在写入奇数到CREP寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 18-9. 中央对齐计数模式下计数器重复时序图

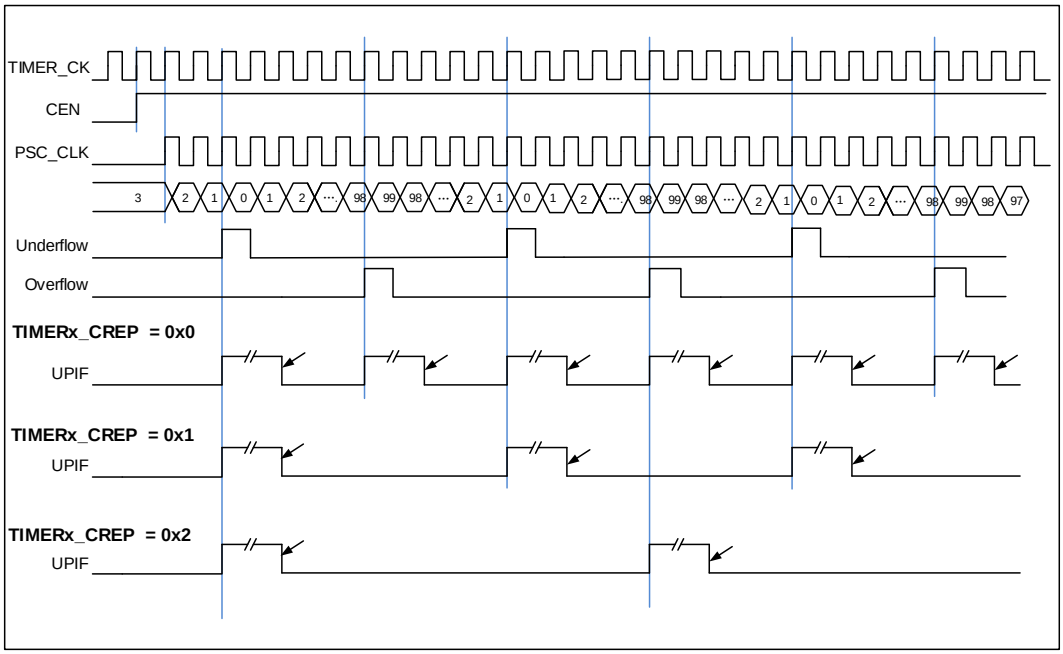


图 18-10. 在向上计数模式下计数器重复时序图

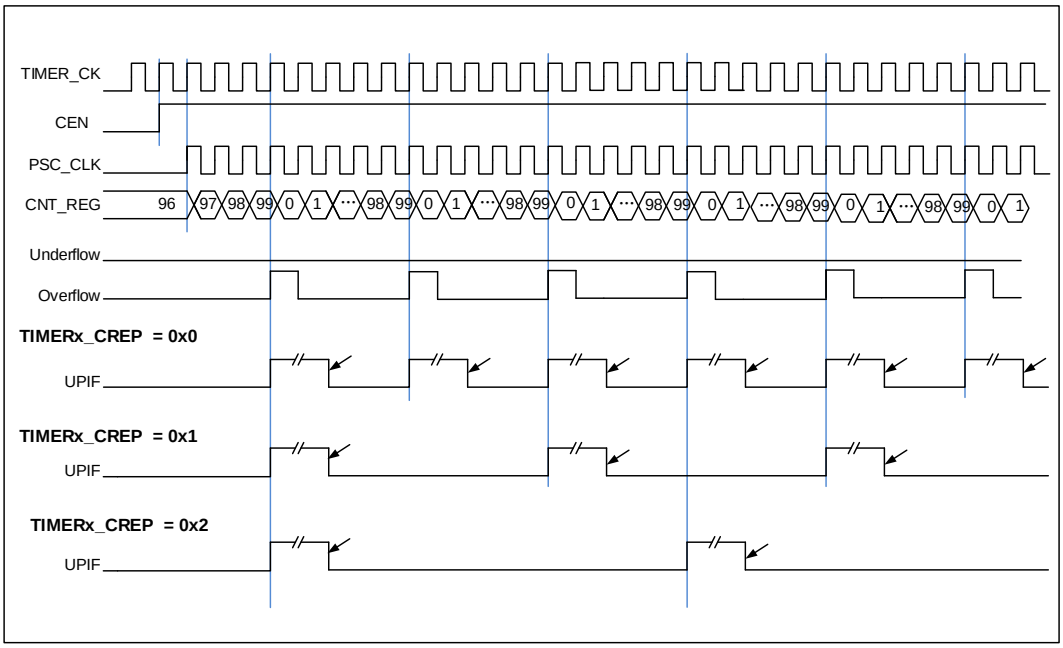
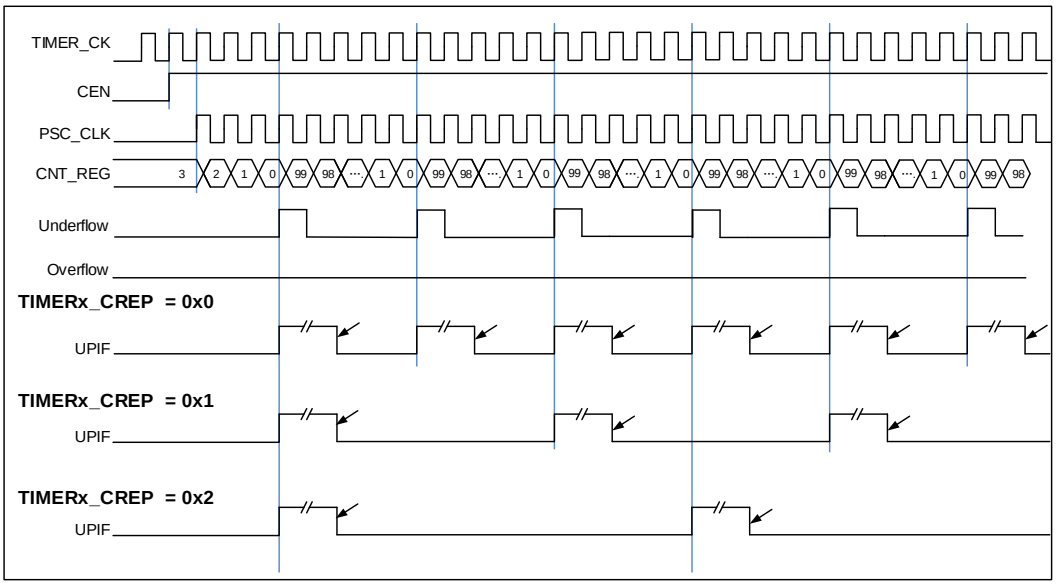


图 18-11. 在向下计数模式下计数器重复时序图



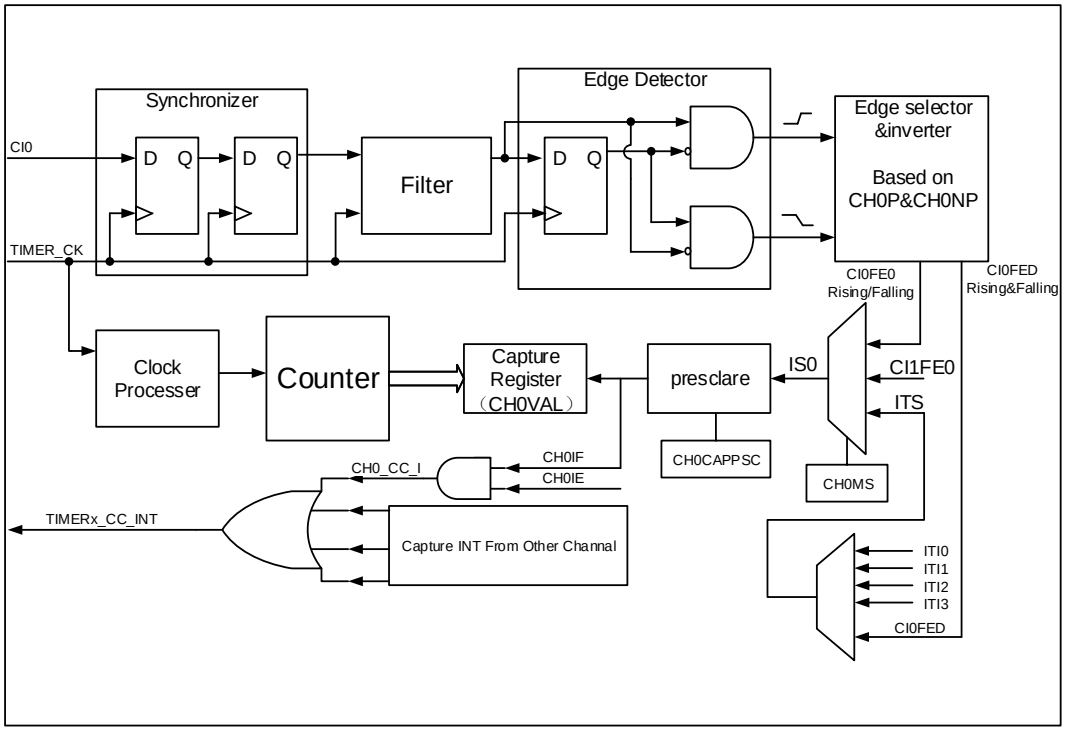
输入捕获和输出比较通道

高级定时器拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级、通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序、频率、周期、占空比等。输入级包括一个数字滤波器、一个通道极性选择、边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，如果CHxIE = 1则产生通道中断。

图 18-12. 通道输入捕获原理



通道输入信号CIx有两种选择，一种是TIMERx_CHx信号，另一种是TIMERx_CH0、TIMERx_CH1和TIMERx_CH2异或之后的信号。通道输入信号CIx先被TIMER_CK信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置CHxP选择使用上升沿或者下降沿。配置CHxMS，可以选择其他通道的输入信号，内部触发信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。在捕获事件中，CHxVAL会恢复计数器的值。

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寄存器不能再被写值。

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和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波

连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERx_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM的占空比值。

■ 通道输出比较功能

图 18-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2）

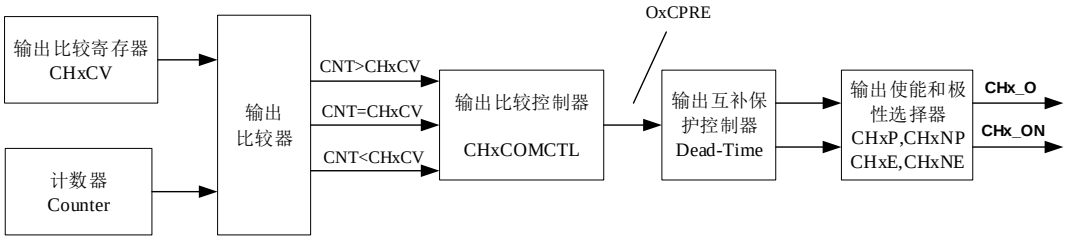


图 18-14. 通道输出比较原理

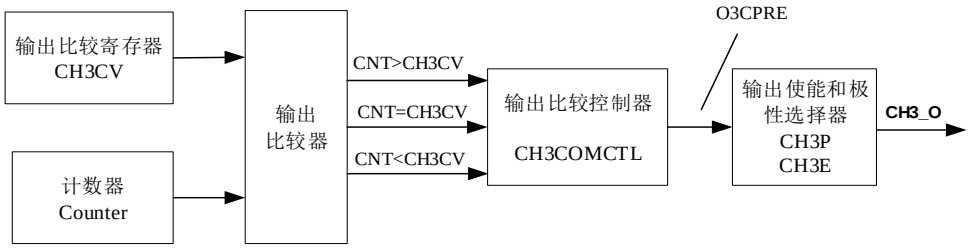


*图18-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2）*和*图18-14. 通道输出比较原理*分别给出了通道输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号（详情请见*通道输出参考信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O/CHx_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如：

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

若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

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

若OxCPRE输出有效（高）电平，则CHx_ON输出有效（低）电平；

若OxCPRE输出无效（低）电平，则CHx_ON输出无效（高）电平。

当CH0_O和CH0_ON同时输出时，CH0_O和CH0_ON的具体输出情况还与TIMERx_CCHP寄存器中的相关位（ROS、IOS、POE和DTCFG等位）有关。

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CHxVAL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、低电平或者反转。当计数器的值与CHxVAL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/反转）；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。

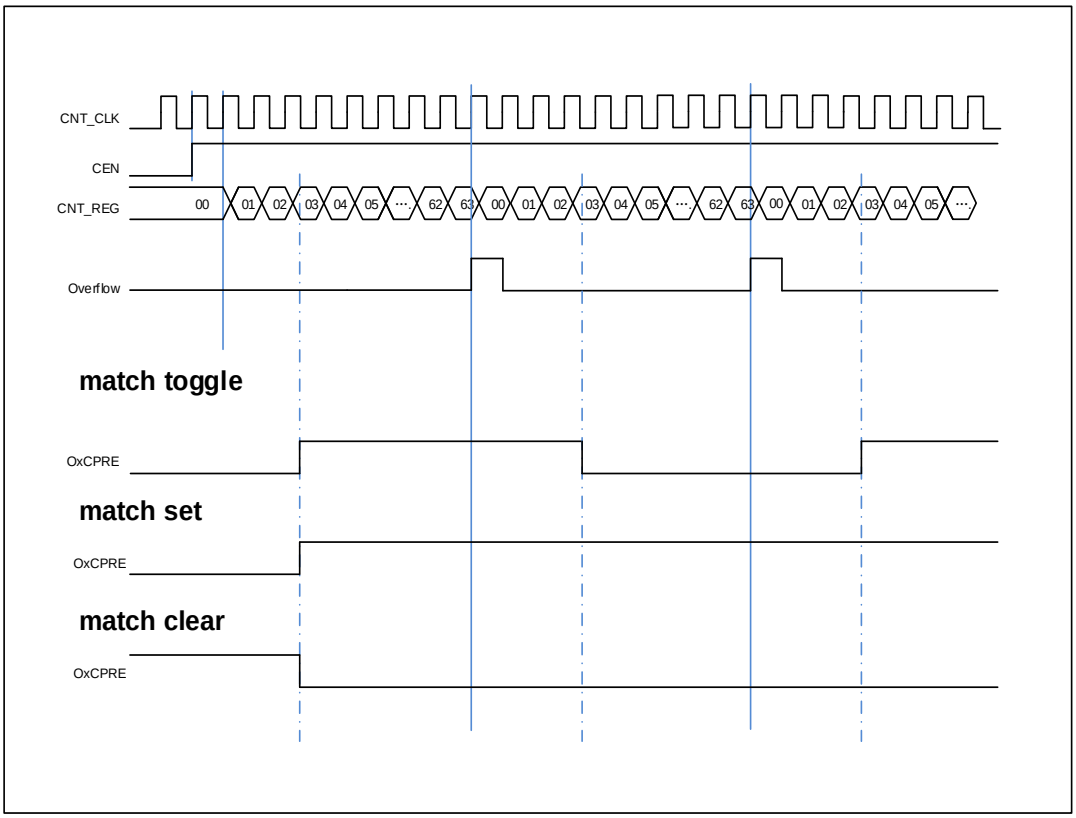
第三步： 通过CHxIE/CHxDEN位配置中断/DMA请求使能。

第四步： 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较定时：
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步： 设置CEN位使能定时器。

*图 18-15. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR = 0x63, CHxVAL = 0x3。

图 18-15. 三种输出比较模式



输出 PWM 功能

在PWM输出模式下（PWM模式0是配置CHxCOMCTL为3'b110，PWM模式1是配置CHxCOMCTL为3'b111），通道根据TIMERx_CAR寄存器和TIMERx_CHxCV寄存器的值，输出

PWM波形。

根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 18-16. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 18-17. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL = 3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR寄存器的值，则通道输出一直为有效电平；如果TIMERx_CHxCV寄存器的值等于0，则通道输出一直为无效电平。

图 18-16. EAPWM 时序图

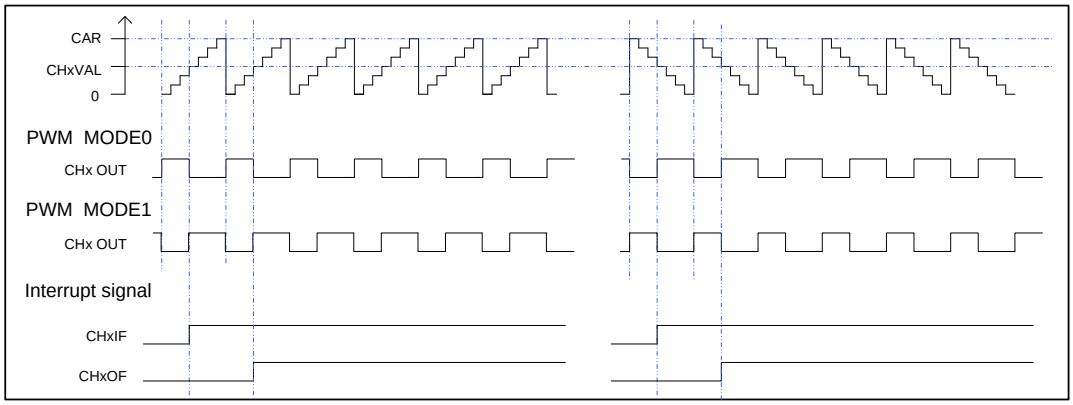
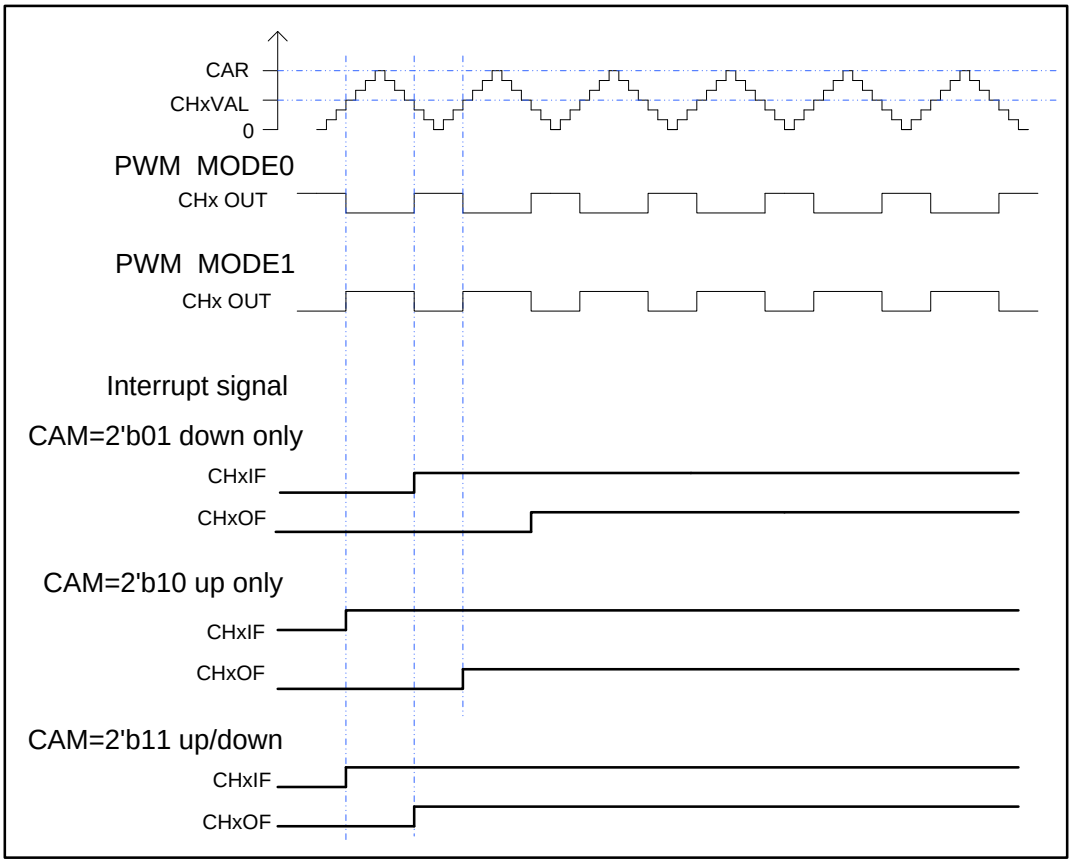


图 18-17. CAPWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL = 0x00可以保持原始电平；设置CHxCOMCTL = 0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL = 0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL = 0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL = 0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间的比较结果。

设置CHxCOMCEN = 1，当由外部ETI引脚信号产生的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。

通道输出互补 PWM

CHx_O和CHx_ON是一对互补输出通道，这两个信号不能同时有效。TIMERx有四路通道，只有前三路有互补输出通道。互补信号CHx_O和CHx_ON是由一组参数来决定：

TIMERx_CHCTL2寄存器中的CHxEN和CHxNEN位，TIMERx_CCHP寄存器和TIMERx_CTL1寄存器中的POEN、ROS、IOS、ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 18-2. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="8">0</td><td rowspan="8">0/1</td><td rowspan="4">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁用.</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O = CHxP CHx_ON = CHxNP
CHx_O/CHx_ON 输出禁用.
如果时钟使能:
CHx_O = ISOx CHx_ON = ISOxN</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O = CHxP CHx_ON = CHxNP
CHx_O/CHx_ON输出禁用.</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O = CHxP CHx_ON = CHxNP
CHx_O/CHx_ON 输出使能.
如果时钟使能:
CHx_O = ISOx CHx_ON = ISOxN</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="8">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON 输出禁用.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O 输出禁用.</td><td>CHx_ON=OxCPRE⊕CHxNP
CHx_ON 输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁用.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>CHx_ON=(!OxCPRE)⊕ CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O输出禁用.</td><td>CHx_ON = CHxNP
CHx_ON输出禁用.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O输出使能</td><td>CHx_ON=OxCPRE⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>CHx_ON = CHxNP
CHx_ON输出使能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE ⊕ CHxP
CHx_O输出使能</td><td>CHx_ON=(!OxCPRE)⊕ CHxNP
CHx_ON输出使能</td></tr>
</table>

互补 PWM 插入死区时间

设置CHxEN和CHxNEN为1'b1同时设置POEN，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对除了通道3以外的通道都有效。死区时间的细节，请参考TIMERx_CCHP寄存器。

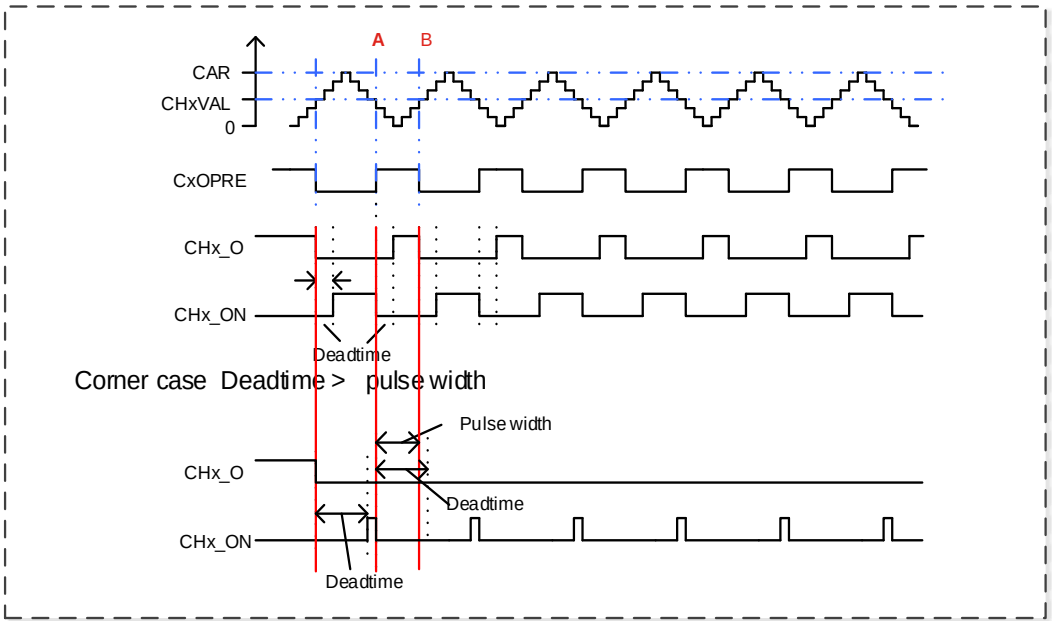
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器 = CHxVAL），OxCPRE 反转。在*图18-18. 带死区时间的通道互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 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 = CHxVAL），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。

有时会有一些极端事件发生，例如：

- 如果死区延时大于或者等于CHx_O信号的占空比，CHx_O信号一直为无效值（如*图18-18. 带死区时间的通道互补输出*）。

图 18-18. 带死区时间的通道互补输出



中止模式

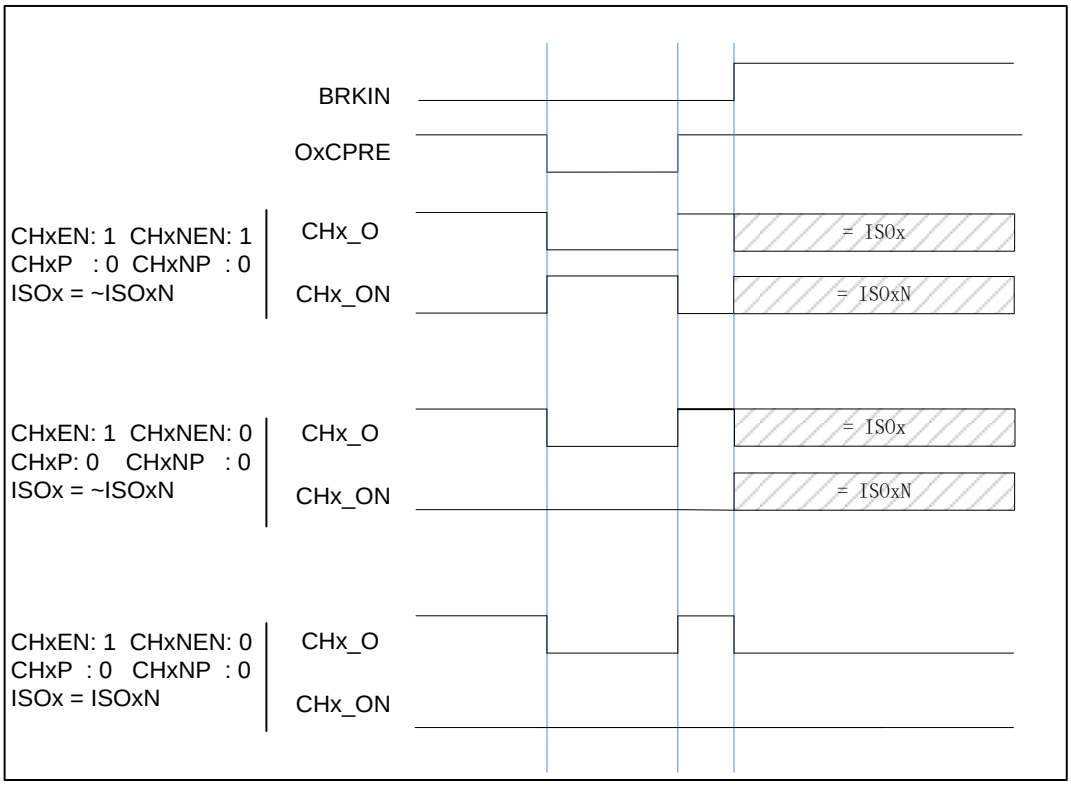
使用中止模式时，输出CHx_O和CHx_ON信号电平被以下位控制：TIMERx_CCHP寄存器的POEN、IOS和ROS位，TIMERx_CTL1寄存器的ISOx和ISOxN位。当中止事件发生时，CHx_O和CHx_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择HXTAL时钟失效事件。时钟失效事件由RCU中的时钟监视器(CKM)产生。将TIMERx_CCHP寄存器的BRKEN位置1可以使能中止功能。TIMERx_CCHP寄存器的BRKP位决定了中止输入极性。

发生中止时，POEN位被异步清除，一旦POEN位为0，CHx_O和CHx_ON被TIMERx_CTL1寄存器中的ISOx位和ISOxN驱动。如果IOS为0，定时器释放输出使能，否则输出使能仍然为高。

起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，TIMERx_INTF寄存器的BRKIF位被置1。如果BRKIE位为1，则产生中断。

图 18-19. 通道响应中止输入（高电平有效）时，输出信号的行为



正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或着可以同时有CI0FE0和 CI1FE1，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制如*表18-3. 不同编码器模式下的计数方向*所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx_CAR寄存器。

表 18-3. 不同编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
编码器模式0 SMC[2:0]=3’b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
编码器模式1 SMC [2:0]=3’b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
编码器模式2 SMC [2:0]=3’b011	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意："-" 意思是"无计数"; "X" 意思是不可能。"0" 意思是低电平, "1" 意思是高电平

图 18-20. 在编码器模式 2 且 CI0FE0 极性不反相时计数器行为

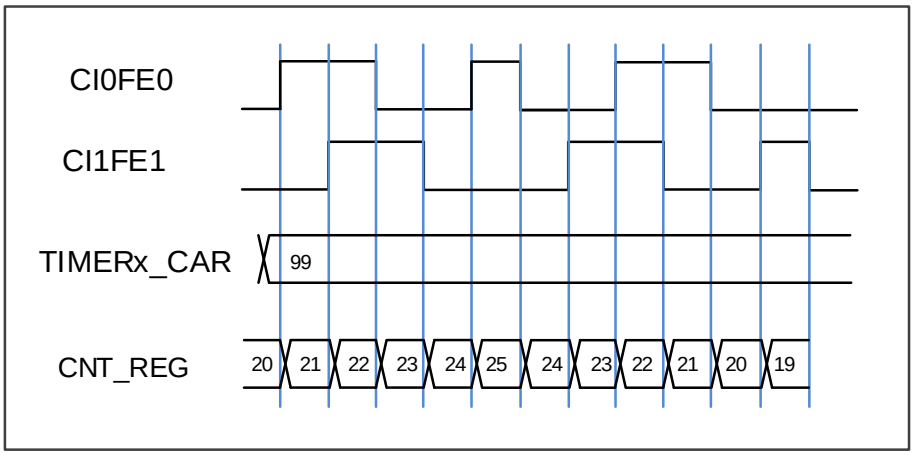
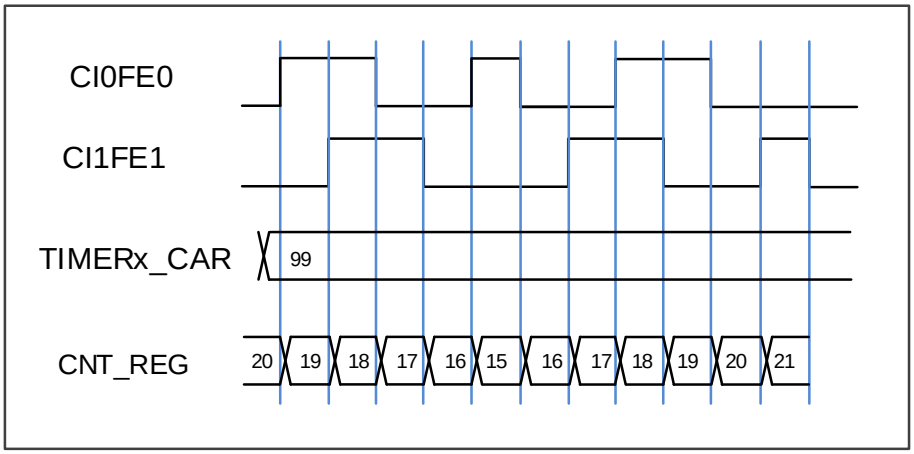


图 18-21. 在编码器模式 2 且 CI0FE0 极性反相时计数器行为



霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制BLDC电机。*图18-22. 霍尔传感器用在BLDC电机控制中*是定时器和电机的连接示意图。TIMER_in定时器（可以是高级定时器或者通用L0定时器）接收霍尔传感器的三路信号。

三个霍尔传感器信号与TIMER_in定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接，例如TRGO-ITIx，TIMER_in定时器和TIMER_out定时器可以连接在一起。TIMER_out定时器根据ITIx触发信号输出PWM波，驱动BLDC电机，控制BLDC电机的速度。这样，TIMER_in定时器和TIMER_out定时器的连接形成了一个反馈电路，可以根据需求改变配

置。

TIMER_in定时器需要具备输入异或功能，所以可以选择高级定时器和通用L0定时器。

TIMER_out定时器需要具备互补输出和死区插入功能，所以可以选择高级定时器。另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如:

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)
TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)
等等。

选择好合适的互连定时器，定时器和BLDC的线路也已经连接好，我们就可以配置定时器了。有以下关键配置：

- 设置TI0S，使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 设置CCUC和CCSE，使能ITIx直接连接到换相功能。
- 根据需求配置PWM参数。

图 18-22. 霍尔传感器用在 BLDC 电机控制中

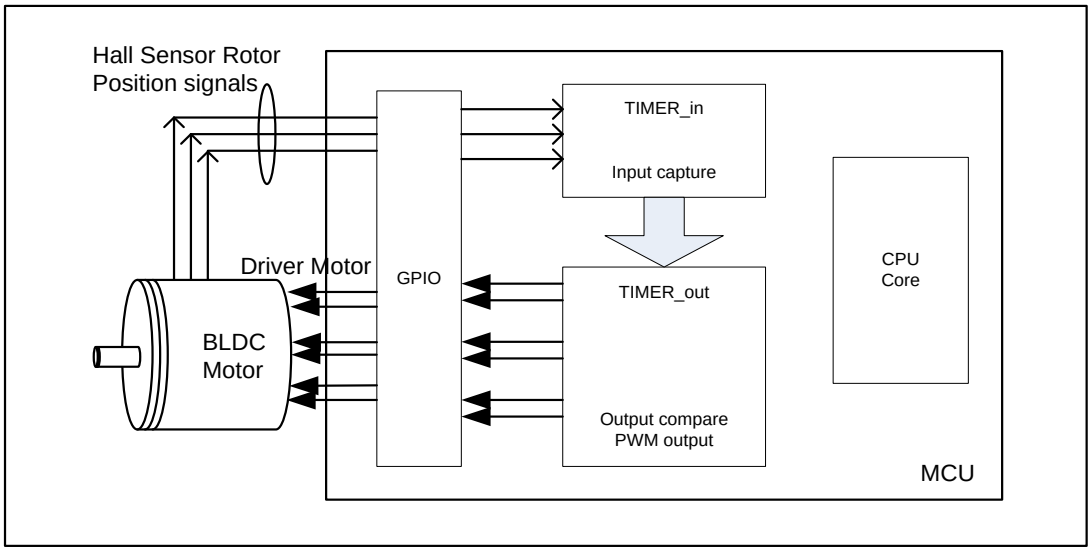
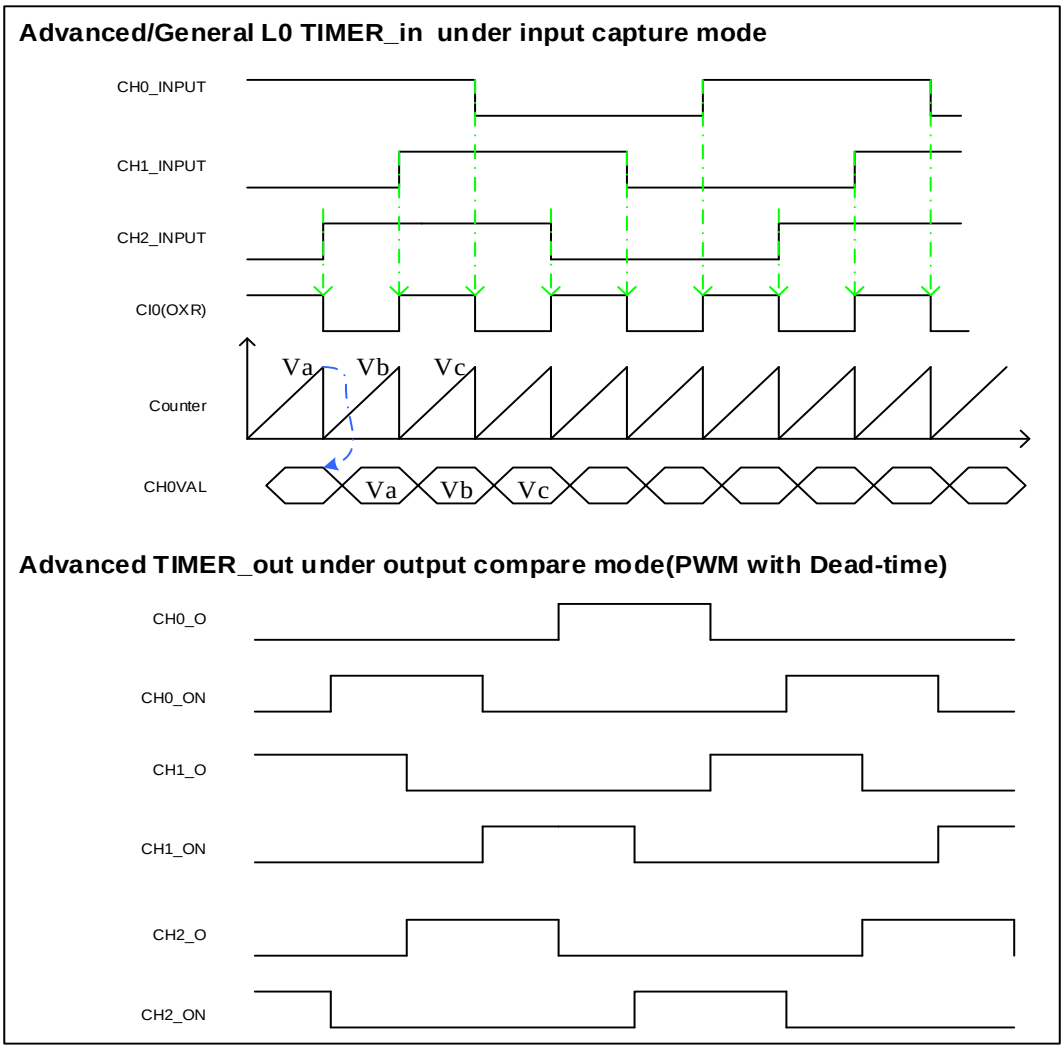


图 18-23. 两个定时器之间的霍尔传感器时序图



主-从管理

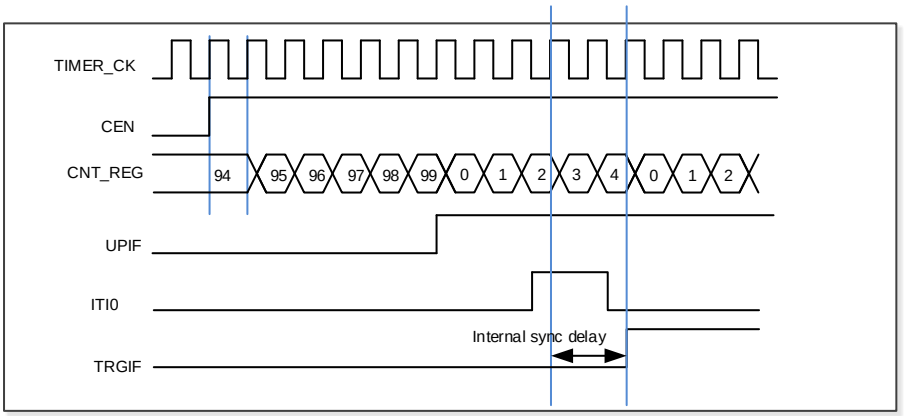
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS [2:0]来选择。

表 18-4. 从模式例子列表

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

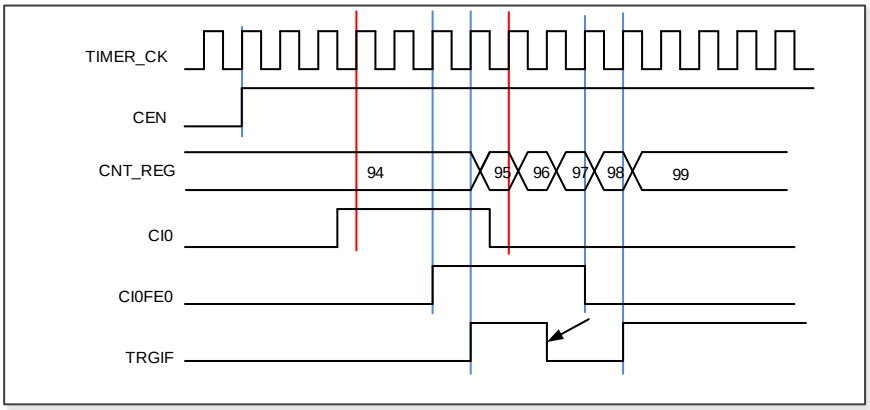
	模式选择	触发源选择	极性选择	滤波和预分频
		111: ETIFP	反相	和预分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3’b000 选择ITI0为触发源	触发源是ITI0，极性选择不可用	触发源是 ITI0，滤波和预分频不可用

图 18-24. 复位模式下的控制电路



例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3’b101 选择CI0FE0为触发源	TI0S=0.（非异或）[CH0NP==0, CH0P==0]不反相.在上升沿捕获	在这个例子中滤波被旁路

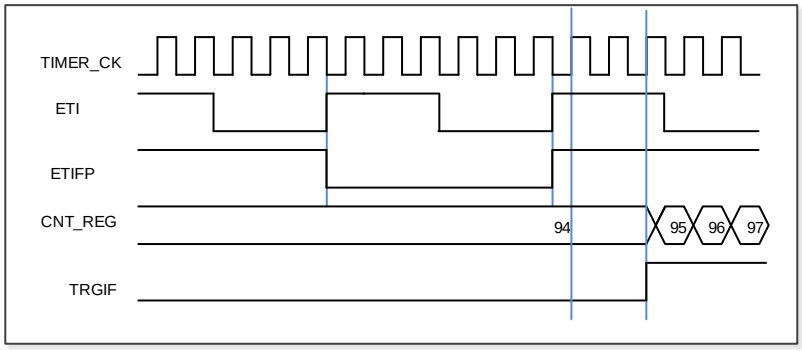
图 18-25. 暂停模式下的控制电路



例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3’b111 选择ETIF为触发源.	ETP = 0 没有极性改变	ETPSC = 1, 2分频. ETFC = 0，无滤波

	模式选择	触发源选择	极性选择	滤波和预分频

图 18-26. 事件模式下的控制电路



单脉冲模式

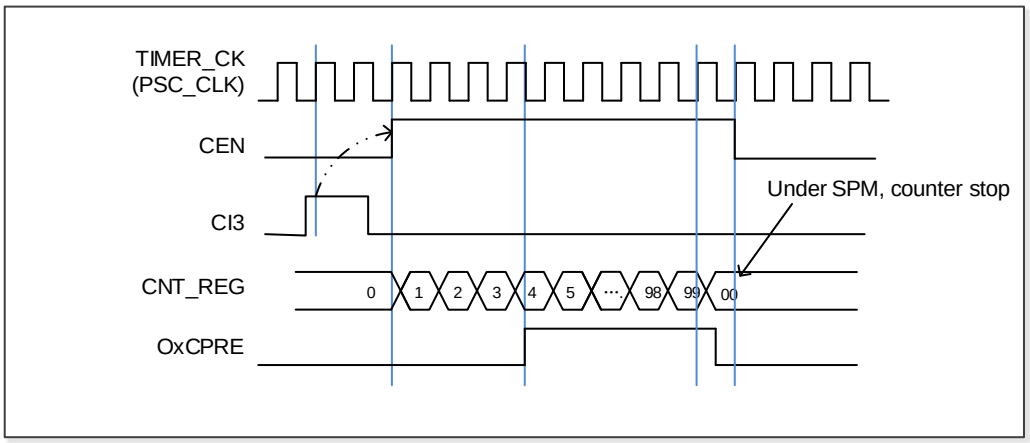
单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置CHxCOMCTL配置TIMERx为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM0或PWM1输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

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

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



定时器互连

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

*图 18-28. 定时器 0 主/从模式的例子*显示了一些主从模式触发选择的例子。

图 18-28. 定时器 0 主/从模式的例子

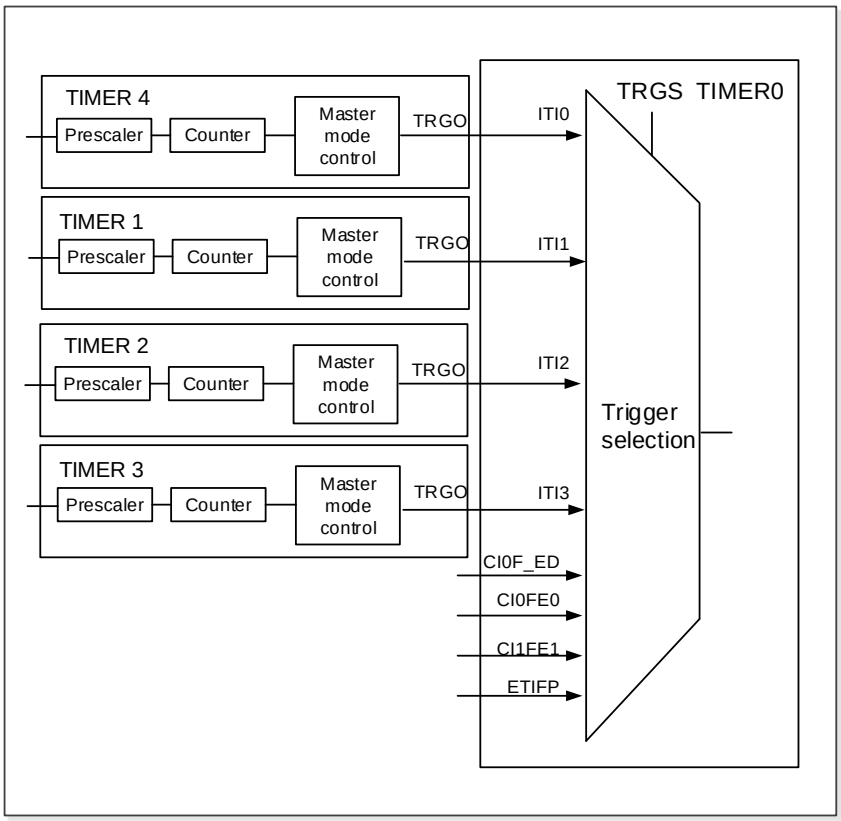


*表18-5. 定时器0和定时器7内部触发*显示了内部触发，*表18-6. 定时器0和定时器7触发输出*显示了外部输出触发。

表 18-5. 定时器 0 和定时器 7 内部触发

	ITI0	ITI1	ITI2	ITI3
TIMER0	TIMER4_TGRO	TIMER1_TGRO	TIMER2_TGRO	TIMER3_TGRO
TIMER7	TIMER0_TGRO	TIMER1_TGRO	TIMER3_TGRO	TIMER4_TGRO

表 18-6. 定时器 0 和定时器 7 触发输出

	TIMER0_TGRO	TIMER7_TGRO
TIMER1	ITI0	-
TIMER2	ITI0	-
TIMER3	ITI0	ITI3
TIMER4	-	ITI3
TIMER7	ITI0	-

注释: '-' 表示没有内部级联。

其他定时器互连的例子：

■ 定时器2作为定时器0的预分频器

参考*图 18-28. 定时器 0 主/从模式的例子*连接配置定时器 2 为定时器 0 的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出(配置TIMER2_CTL1寄存器的MMC=3’b010)。定时器2在每次计数器溢出时，输出一个周期信号；
2. 配置定时器2周期(TIMER2_CAR寄存器)；
3. 选择定时器0输入触发源为定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3’b010)；
4. 配置定时器0在外部时钟模式1 (配置TIMERx_SMCFG寄存器的SMC=3’b111)；
5. 写1到CEN位启动定时器0 (TIMER0_CTL0寄存器)；
6. 写1到CEN位启动定时器2 (TIMER2_CTL0寄存器)。

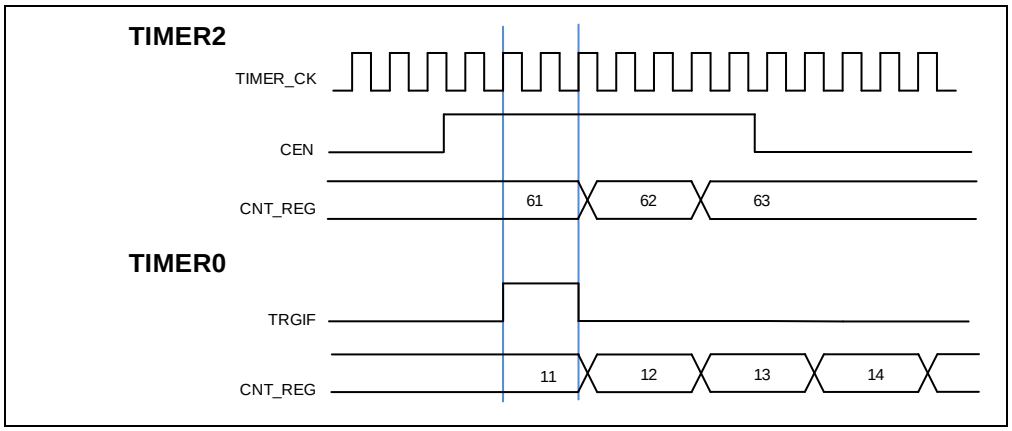
■ 用定时器2的使能信号来启动定时器0

在这个例子中，我们使用定时器 2 的使能信号来启动定时器 0，见*图 18-29. 用定时器 2 的使能信号启动定时器 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位置1，计数器计数直到禁能定时器0。在本示例中，两个定时器的计数器频率都是TIMER_CK经过预分频器3分频后的频率($f_{CNT_CLK} = f_{TIMER_CK}/3$)。定时器0的SMC配置的是事件模式，所以定时器2的禁能信号不能触发定时器0的禁能。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=3’b001)；
2. 配置定时器0选择输入触发来自定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3’b010)；
3. 配置定时器0在事件模式 (配置TIMERx_SMCFG寄存器的SMC=3’b110)；
4. 写1到CEN来开启定时器2 (TIMER2_CTL0寄存器)。

图 18-29. 用定时器 2 的使能信号启动定时器 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输入信号的上升沿来触发定时器2本身。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下：

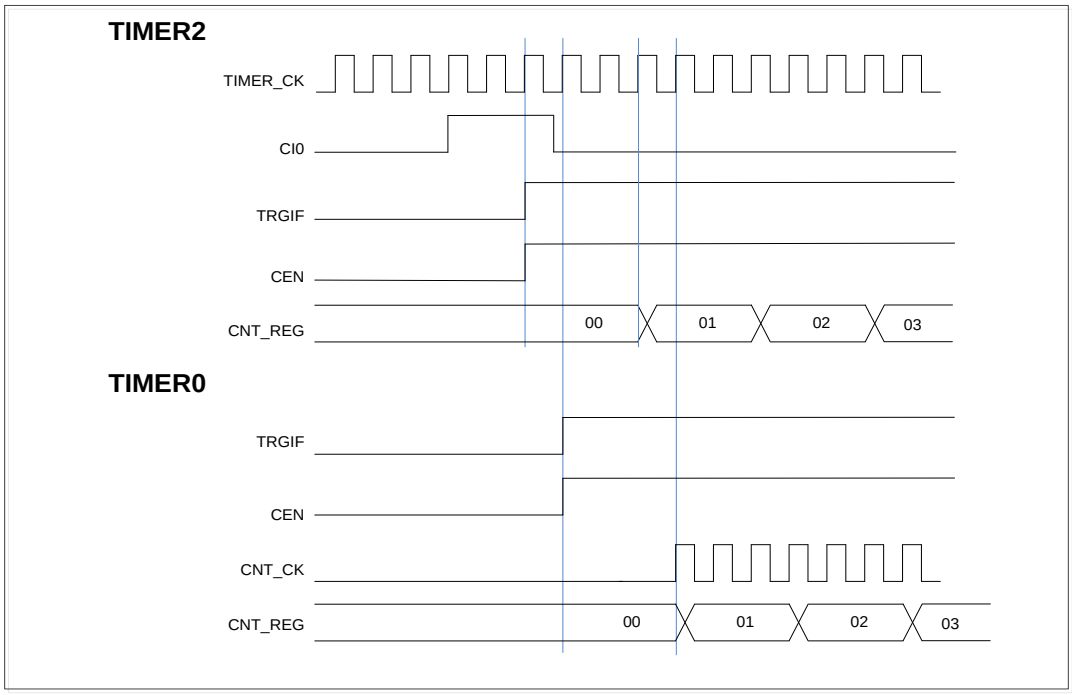
1. 配置定时器2工作在从模式来获取来自CI0的触发输入(配置TIMER2_SMCFG寄存器的

TRGS=3’b100)；

2. 配置定时器2工作在事件模式(配置TIMER2_SMCFG寄存器的SMC=3’b110)；
3. 写MSM=1(TIMER2_SMCFG寄存器)来配置定时器2工作在主/从模式；
4. 配置定时器0的触发输入来自定时器2 (配置TIMERx_SMCFG 寄存器的TRGS=3’b010)；
5. 配置定时器0工作在事件模式(配置TIMER0_SMCFG寄存器的SMC=3’b110)。

当定时器2的CI0信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 18-30. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx_DMACFG和TIMERx_DMATB。当然，必须要使能DMA请求，一些内部中断事件可以产生DMA请求。当中断事件发生后，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®-M3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。

18.1.5. TIMERx 寄存器(x=0,7)

TIMER0基地址：0x4001 2C00

TIMER7基地址：0x4001 3400

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器 1：使能TIMERx_CAR寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1 当计数器使能以后，该位不能从0x00切换到非0x00
4	DIR	方向

		0：向上计数 1：向下计数 当计数器配置为中央对齐计数模式或编码器模式时，该位只读。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	保留	CCSE
	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14	ISO3	通道3的空闲状态输出 参考ISO0位
13	ISO2N	通道2的互补通道空闲状态输出 参考ISO0N位
12	ISO2	通道2的空闲状态输出 参考ISO0位
11	ISO1N	通道1的互补通道空闲状态输出 参考ISO0N位
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISO0N	通道0的互补通道空闲状态输出 0：当POEN复位，CH0_ON设置低电平. 1：当POEN复位，CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
8	ISO0	通道0的空闲状态输出 0：当POEN复位，CH0_O设置低电平 1：当POEN复位，CH0_O设置高电平 如果CH0_ON生效，一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
7	TI0S	通道0触发输入选择 0：选择TIMERx_CH0 引脚作为通道0的触发输入 1：选择TIMERx_CH0, CH1 and CH2 引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制l 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE

		111：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O3CPRE
3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道x的DMA请求 . 1：当更新事件发生，发送通道x的DMA请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN, CHxNEN和CHxCOMCTL位）使能(CCSE=1)，这些影子寄存器更新控制如下： 0：CMTG位被置1时更新影子寄存器 1：当CMTG位被置1或检测到TRIGI上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值.
0	CCSE	换相控制影子使能 0：影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能. 1：影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

从模式配置寄存器（TIMERx_SMCFG）

地址偏移：0x08

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			保留	SMC[2:0]		
rw	rw	rw		rw				rw	rw				rw		

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定 ETI 信号的极性 0：ETI 高电平或上升沿有效 . 1：ETI 低电平或下降沿有效 .
14	SMC1	SMC 的一部分为了使能外部时钟模式 1 在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动 0：外部时钟模式 1 禁能 1：外部时钟模式 1 使能

当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3’b111。

如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF

注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0]

外部触发预分频

外部触发信号 ETIFP 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。

00：预分频禁能

01：2 分频

10：4 分频

11：8 分频

11:8 ETFC[3:0]

外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 f_{SAMP} 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

EXTFC[3:0]	次数	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{TIMER_CK}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS_CK}/2$
4’b0101	8	
4’b0110	6	$f_{DTS_CK}/4$
4’b0111	8	
4’b1000	6	$f_{DTS_CK}/8$
4’b1001	8	
4’b1010	5	$f_{DTS_CK}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS_CK}/32$
4’b1110	6	
4’b1111	8	

7 MSM

主-从模式

该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。

0：主从模式禁能

1：主从模式使能

6:4 TRGS[2:0]

触发选择

该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源

000: ITI0

001: ITI1

		010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP 从模式被使能后这些位不能改
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001：编码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 010：编码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 011：编码器模式 2. 根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/ 下计数 100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件. 101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止 110：事件模式.计数器在触发输入的上升沿启动。 111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值.
14	TRGDEN	触发DMA请求使能 0：禁止触发DMA请求 1：使能触发DMA请求
13	CMTDEN	换相DMA更新请求使能 0：禁止换相DMA更新请求 1：使能换相DMA更新请求

12	CH3DEN	通道3比较/捕获 DMA请求使能 0：禁止通道3比较/捕获DMA请求 1：使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0：禁止通道2比较/捕获DMA请求 1：使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0：禁止通道1比较/捕获DMA请求 1：使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0：禁止通道0比较/捕获DMA请求 1：使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0：禁止更新DMA请求 1：使能更新DMA请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4	CH3IE	通道3比较/捕获中断使能 0：禁止通道3中断 1：使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0：禁止通道2中断 1：使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0：禁止通道1中断 1：使能通道1中断
1	CH0IE	通道0比较/捕获中断使能 0：禁止通道0中断 1：使能通道0中断
0	UPIE	更新中断使能 0：禁止更新中断

1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0	.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值.
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生

5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置1，此位由软件清0。 0：无通道换相更新中断发生 1：通道换相更新中断发生
4	CH3IF	通道3比较/捕获中断标志 参见CH0IF描述
3	CH2IF	通道2比较/捕获中断标志 参见CH0IF描述
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0：无通道0中断发生 1：通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值.
7	BRKG	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POEN位被清0且BRKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0：不产生中止事件

		1：产生中止事件
6	TRGG	触发事件产生 此位由软件置1，由硬件自动清0。当此位被置1，根据CCSE位(TIMERx_CTL1寄存器中)的值，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0：无触发事件产生 1：产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置1，由硬件自动清0. 当此位被置1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和CHxCOMCTL) 的互补输出被更新。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4	CH3G	通道3捕获或比较事件发生 参见CH0G描述
3	CH2G	通道2捕获或比较事件发生 参见CH0G描述
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。

		011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 0 配置为输入，IS0 映射在 ITS 上 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制

参见 CH0CAPFLT 描述

11:10 CH1CAPPSC[1:0] 通道 1 输入捕获预分频器

参见 CH0CAPPSC 描述

9:8 CH1MS[1:0] 通道 1 模式选择

与输出模式相同

7:4 CH0CAPFLT[3:0] 通道 0 输入捕获滤波控制

CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。

数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。

滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

通道控制寄存器 1（TIMERx_CHCTL1）

地址偏移：0x1C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 注意：当 CH3MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用

		001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00（比较模式）时此位不能被改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00 时此位不能被改变。
2	CH2COMFEN	通道 2 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。 0：禁止通道 2 输出比较快速. 1：使能通道 2 输出比较快速。
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。 00：通道 2 配置为输出 01：通道 2 配置为输入，IS2 映射在 CI2FE2 上 10：通道 2 配置为输入，IS2 映射在 CI3FE2 上 11：通道 2 配置为输入，IS2 映射在 ITS 上. 注意：当 CH2MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH3CAPPSC[1:0]	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH3MS[1:0]	通道 3 模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道 2 输入捕获滤波控制 CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2	CH2CAPPSC[1:0]	通道 2 输入捕获预分频器 这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13	CH3P	通道3极性 参考CH0P描述
12	CH3EN	通道3使能 参考CH0EN 描述
11	CH2NP	通道2互补输出极性 参考CH0NP 描述
10	CH2NEN	通道2互补输出使能 参考CH0NEN 描述
9	CH2P	通道2极性 参考CH0P 描述
8	CH2EN	通道2使能 参考CH0EN描述
7	CH1NP	通道1互补输出极性 参考CH0NP描述
6	CH1NEN	通道1互补输出使能 参考CH0NEN描述
5	CH1P	通道1极性 参考CH0P 描述
4	CH1EN	通道1使能 参考CH0EN 描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平

		1：通道0互补输出低电平为有效电平 当TIMERx_CCHP寄存器的PROT[1:0]=11或10，并且TIMERx_CHCTL0寄存器的CH0MS[1:0]=00时此位不能被更改。
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。 0：禁止通道0互补输出 1：使能通道0互补输出
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道0配置为输入模式时，此位定义了IS0信号极性 0：上升沿。输入捕获发生在IS0的上升沿；用作外部触发器时IS0不反相。 1：下降沿。输入捕获发生在IS0的下降沿；用作外部触发器时IS0反相。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 注意：在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

重复计数寄存器（TIMERx_CREP）

地址偏移：0x30

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

CH1VAL[15:0]
rw

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

通道 2 捕获/比较寄存器（TIMERx_CH2CV）

地址偏移：0x3C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

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

通道 3 捕获/比较寄存器（TIMERx_CH3CV）

地址偏移：0x40

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:16	保留	必须保持复位值
15	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0 –有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0：禁止通道输出 1：使能通道输出 注意：仅当 CHxMS[1:0]=2’b00 时该位有效
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

13	BRKP	中止极性 此位定义了中止输入信号 BRKIN 的极性。 0：中止输入低电平有效 1：中止输入高电平有效
12	BRKEN	中止使能 此位置 1 使能中止事件和 CKM 时钟失败事件输入。 0：禁能中止输入 1：使能中止输入 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0：当 POEN 位被置 1，通道输出信号 (CHx_O/ CHx_ON)被禁止。 1：当 POEN 位被置 1，通道输出信号 (CHx_O / CHx_ON) 被使能，和 TIMER0_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0：当 POEN 位被清 0，通道输出信号(CHx_O/ CHx_ON)被禁止。 1：当 POEN 位被清 0，通道输出信号(CHx_O/ CHx_ON)被使能，和 TIMERx_CHCTL2 寄存器 CHxEN/CHxNEN 位有关。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00：禁能保护模式。无写保护。 01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。 10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位写保护。 11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。 系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。
7:0	DTCFG[7:0]	死区时间控制 DTCFG 值和死区时间的关系如下：

DTCFG[7:5]	The duration of dead-time
3’b0xx	DTCFG[7:0] * t_{DTS_CK}
3’b10x	(64+ DTCFG[5:0]) * t_{DTS_CK} *2
3’b110	(32+ DTCFG[4:0]) * t_{DTS_CK} *8
3’b111	(32+ DTCFG[4:0]) * t_{DTS_CK} *16

注意：

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

DMA 配置寄存器（TIMERx_DMACFG）

地址偏移：0x48

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

DMA 发送缓冲区寄存器（TIMERx_DMATB）

地址偏移：0x4C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述

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

18.2. 通用定时器 L0（TIMERx, x=1,2,3,4）

18.2.1. 简介

通用定时器L0（定时器1/2/3/4）是4通道定时器，支持输入捕获，输出比较，产生PWM信号控制电机和电源管理。通用定时器L0计数器是16位无符号计数器。

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。

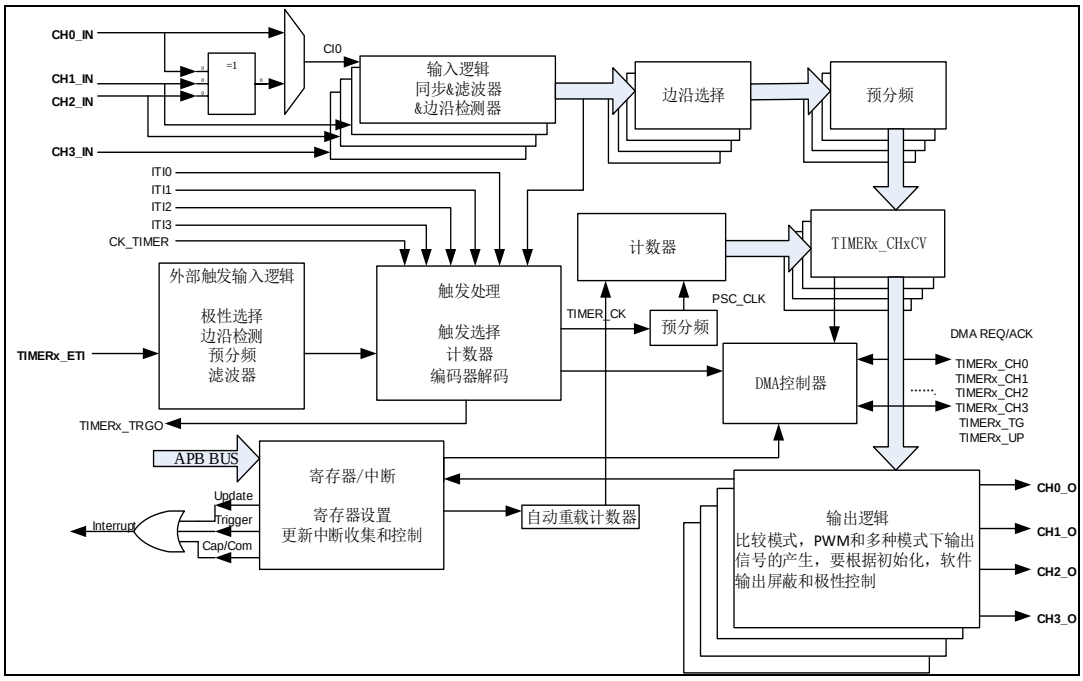
18.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央对齐计数；
- 正交编码器接口：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；
- 多个定时器的菊型链接使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

18.2.3. 结构框图

*图 18-31. 通用定时器 L0 结构框图*提供了通用定时器 L0 的内部细节

图 18-31. 通用定时器 L0 结构框图



18.2.4. 功能说明

时钟源配置

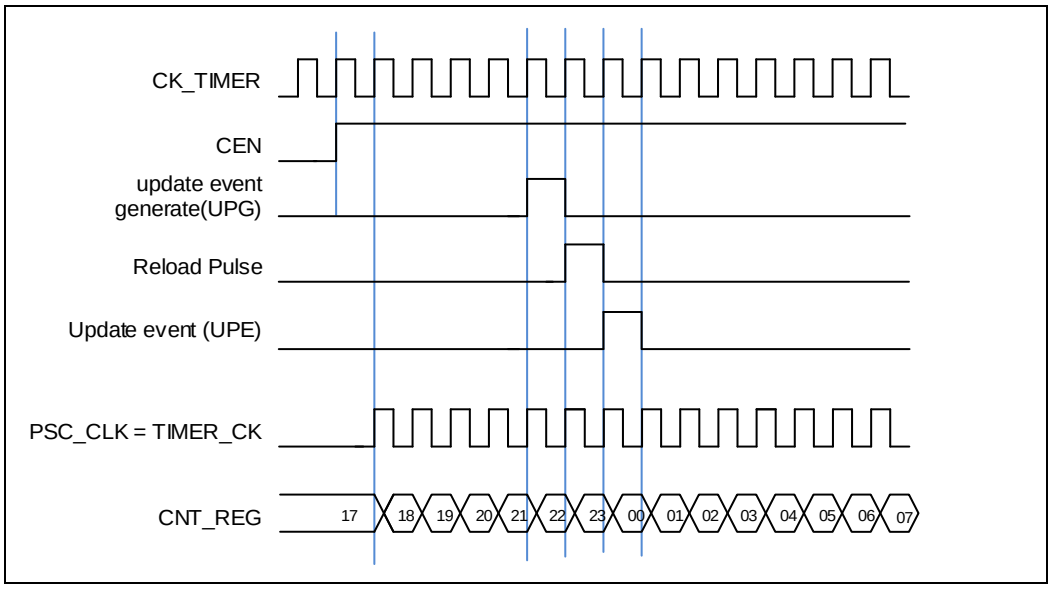
通用定时器L0可以由内部时钟源CK_TIMER或者由SMC（TIMERx_SMCFG寄存器位[2:0]）控制的复用时钟源驱动。

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

如果SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源CK_TIMER。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

如果将TIMERx_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7，预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动，在下文说明。当SMC位被设置为0x4、0x5和0x6，计数器预分频器时钟源由内部时钟CK_TIMER驱动。

图 18-32. 内部时钟分频为 1 时，计数器的时序图



■ SMC[2:0]==3’b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在TIMERx_CI0/ TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x4， 0x5或0x6来选择。CIx 是TIMERx_CIx通过数字滤波器采样后的信号。

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

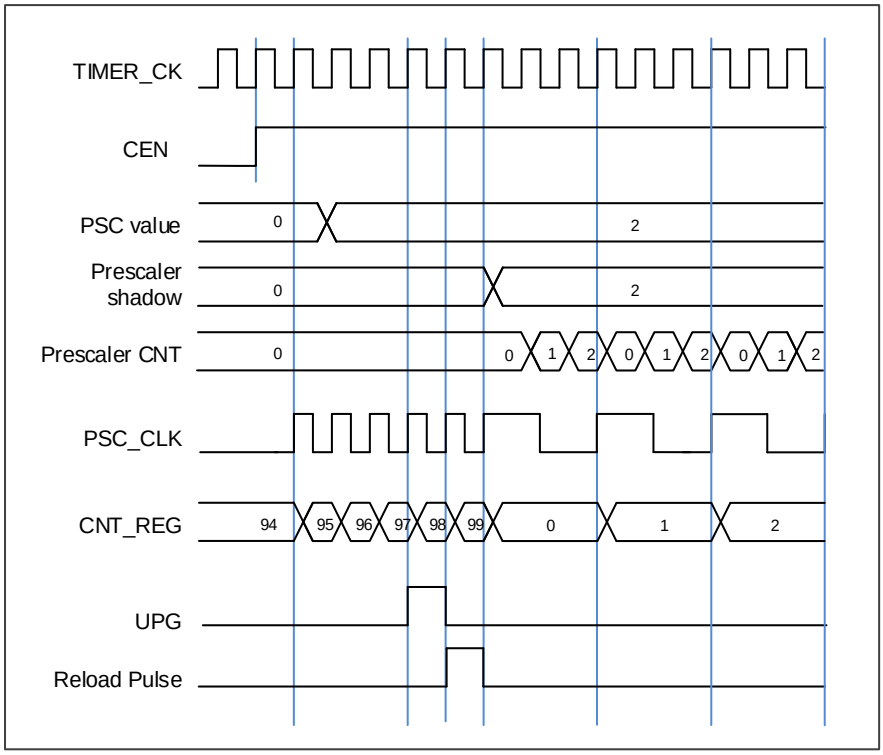
■ SMC1==1’b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 18-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-34. 向上计数时序图，PSC=0/2*和*图18-35. 向上计数时序图，在运行时改变TIMERx_CAR寄存器的值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同预分频因子下的行为。

图 18-34. 向上计数时序图，PSC=0/2

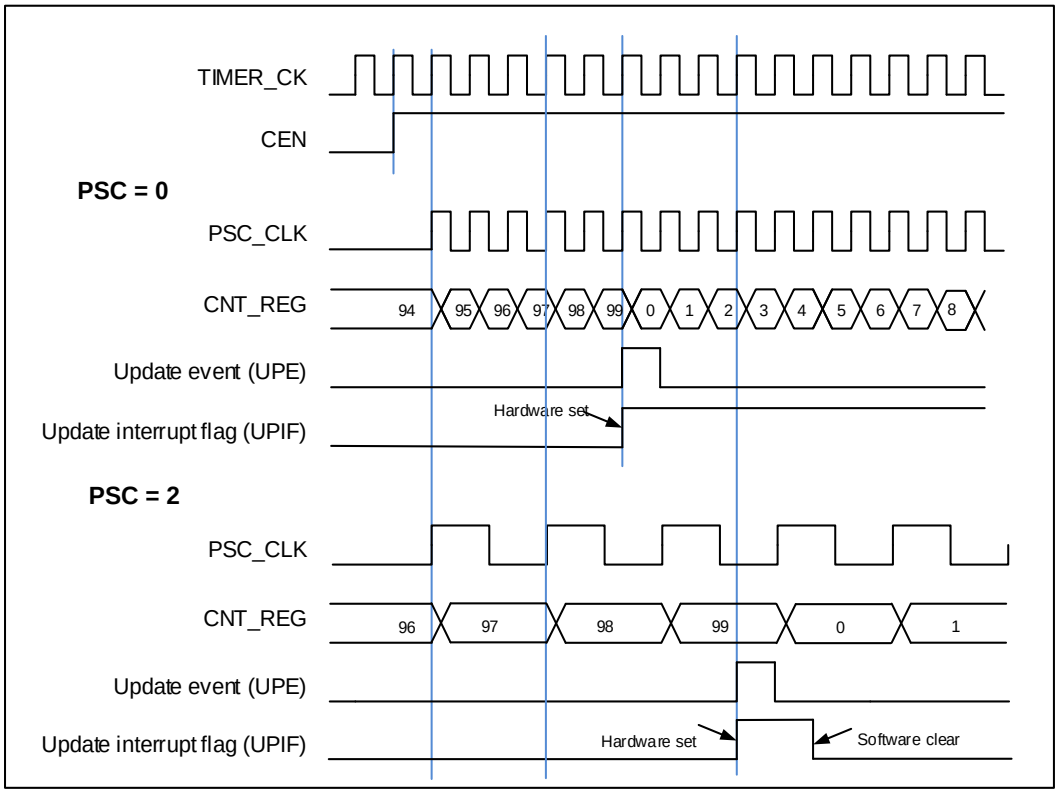
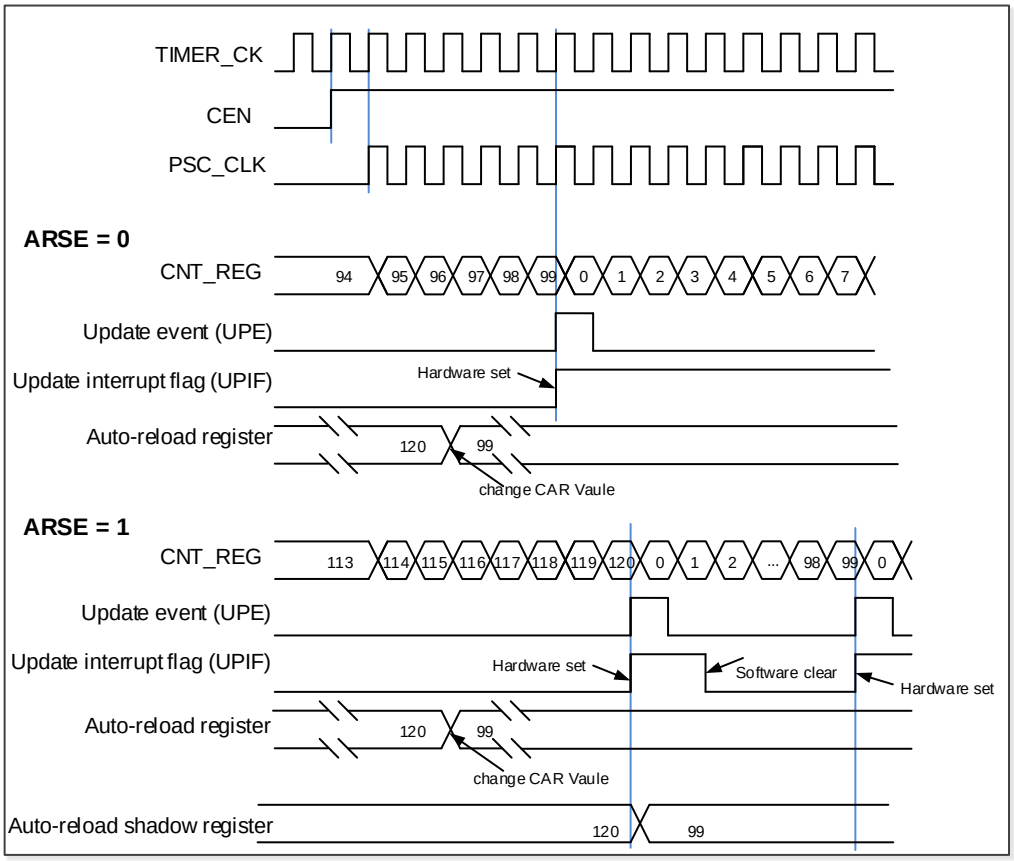


图 18-35. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

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

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-36. 向下计数时序图，PSC=0/2*和*图18-37. 向下计数时序图，在运行时改变TIMERx_CAR寄存器值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同时钟频率下的行为。

图 18-36. 向下计数时序图，PSC=0/2

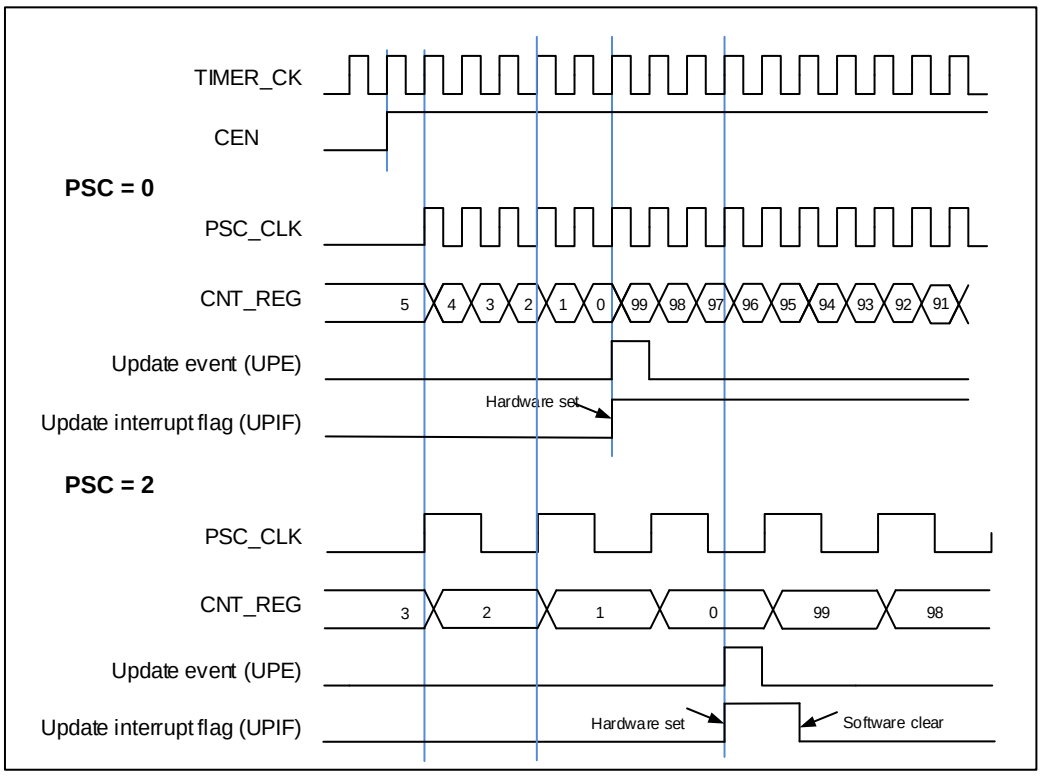
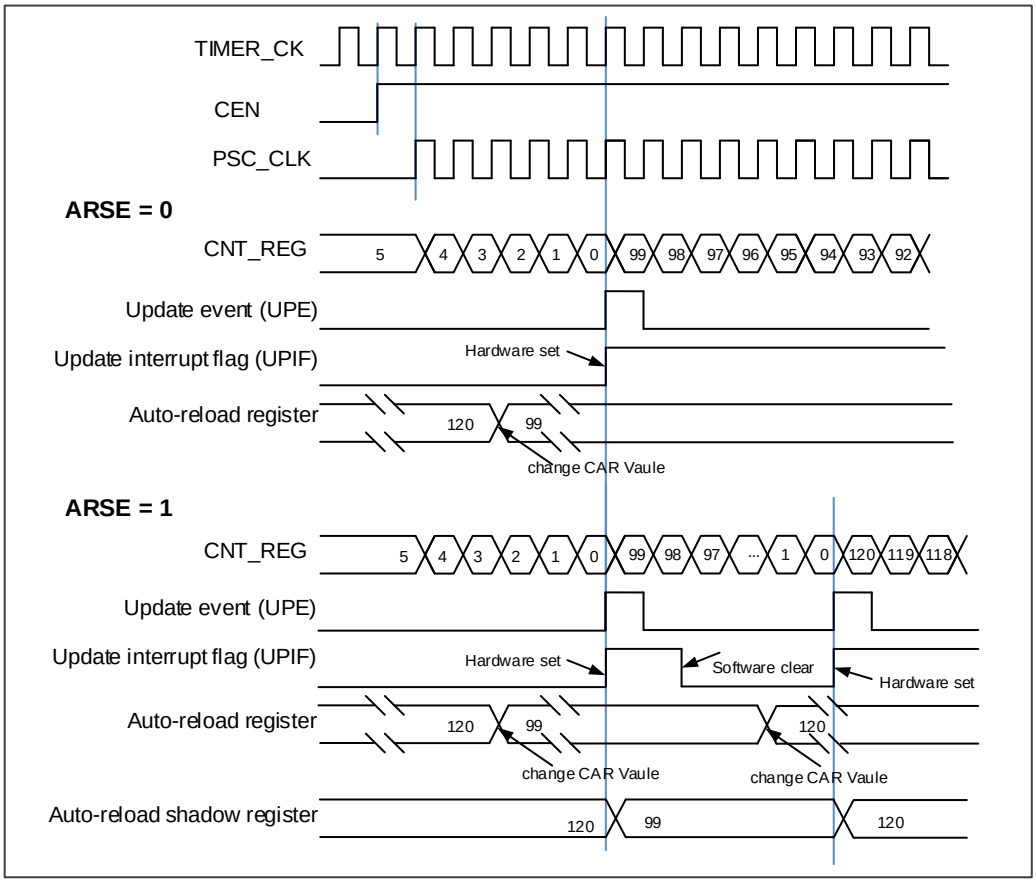


图 18-37. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

在中央对齐计数模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。向上计数模式中， 定时器模块在计数器计数到（自动加载值-1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，表明了的计数方向。

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

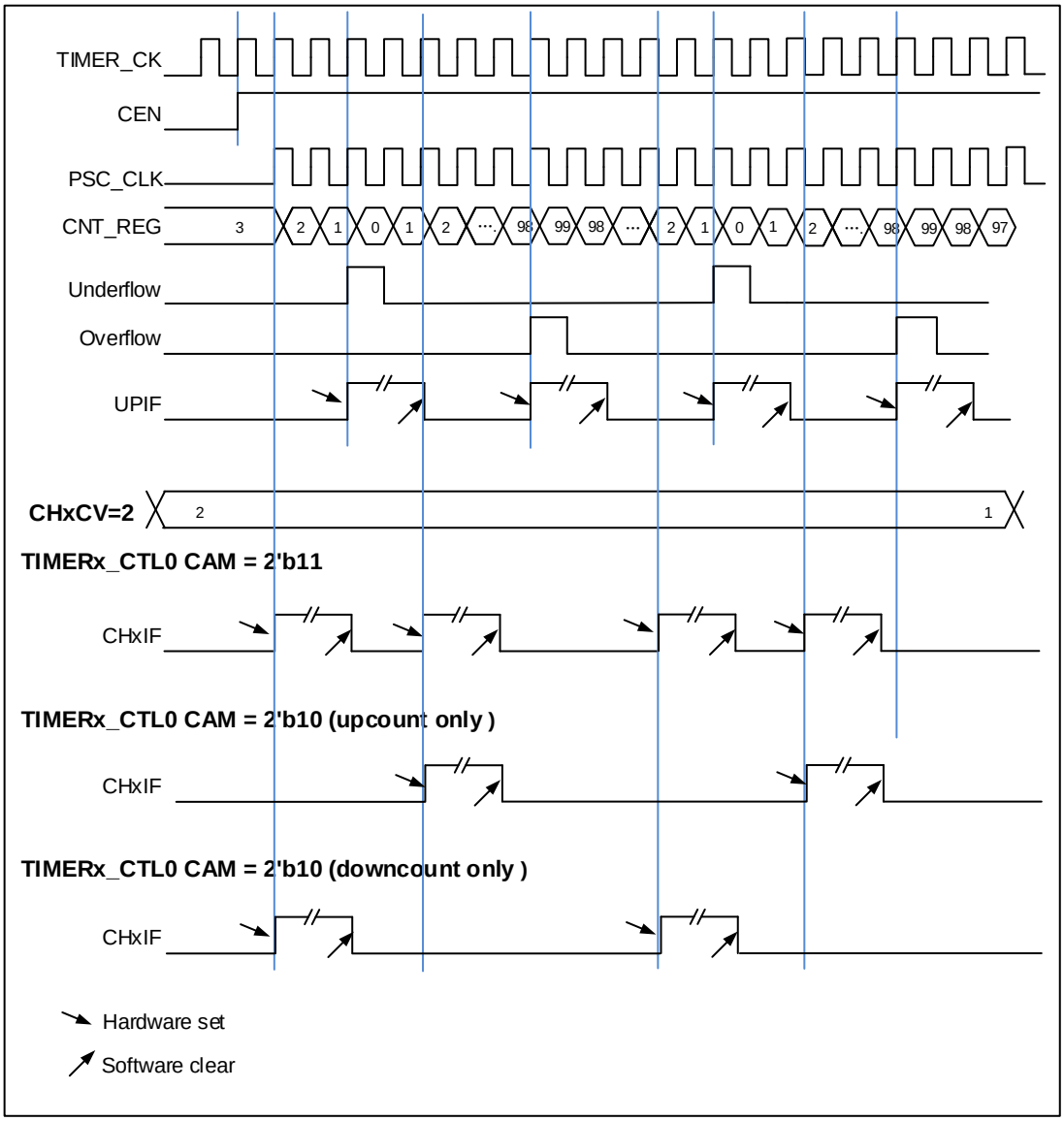
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 18-38. 中央对齐计数模式计数器时序图*。

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

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-38. 中央对齐计数模式计数器时序图*给出了一些例子，当TIMERx_CAR=0x99，TIMERx_PSC=0x0时，计数器的行为

图 18-38. 中央对齐计数模式计数器时序图



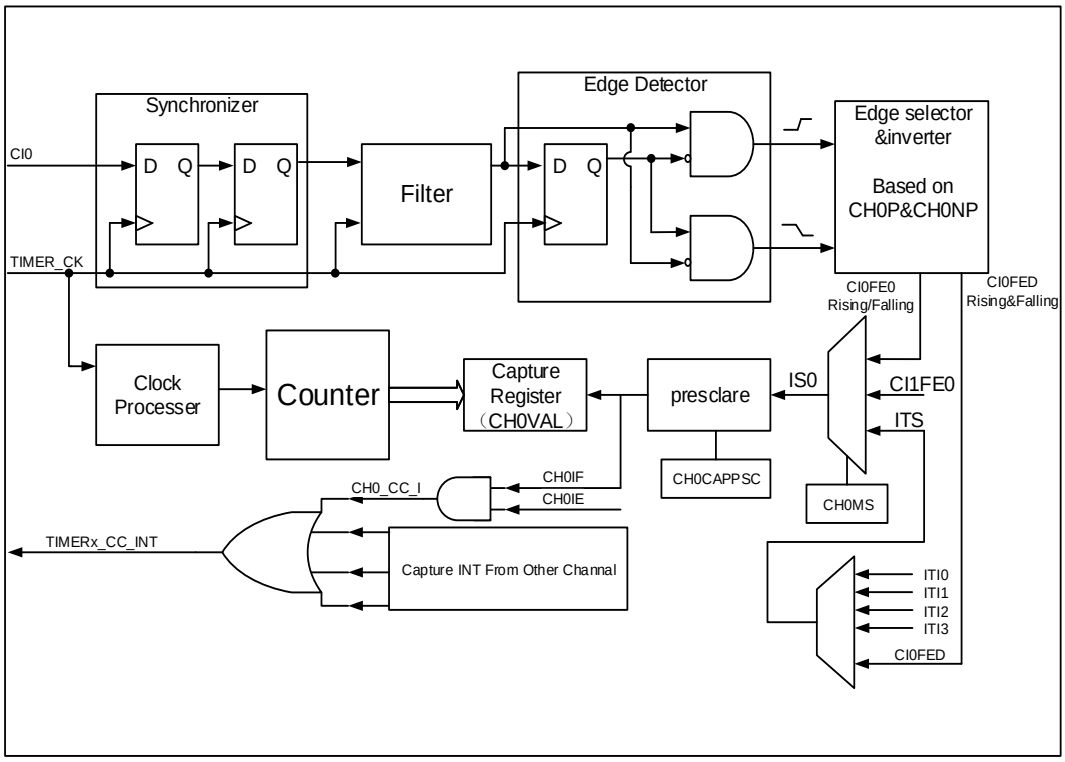
输入捕获和输出比较通道

通用定时器L0拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，如果CHxIE = 1则产生通道中断。

图 18-39. 通道输入捕获原理



通道输入信号CIx有两种选择，一种是TIMERx_CHx信号，另一种是TIMERx_CH0,TIMERx_CH1和TIMERx_CH2异或之后的信号。通道输入信号CIx先被TIMER_CK信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置CHxP选择使用上升沿或者下降沿。配置CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CxCV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_CxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE and CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

■ 通道输出比较功能

图 18-40. 通道输出比较原理（x=0,1,2,3）

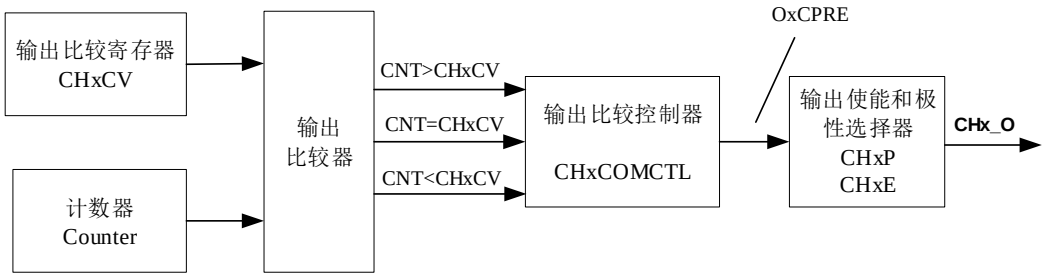


*图18-40. 通道输出比较原理（x=0,1,2,3）*给出了通道输出比较的原理电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出参考信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx_O输出使能）时：

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CxCV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CxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxCDE=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

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

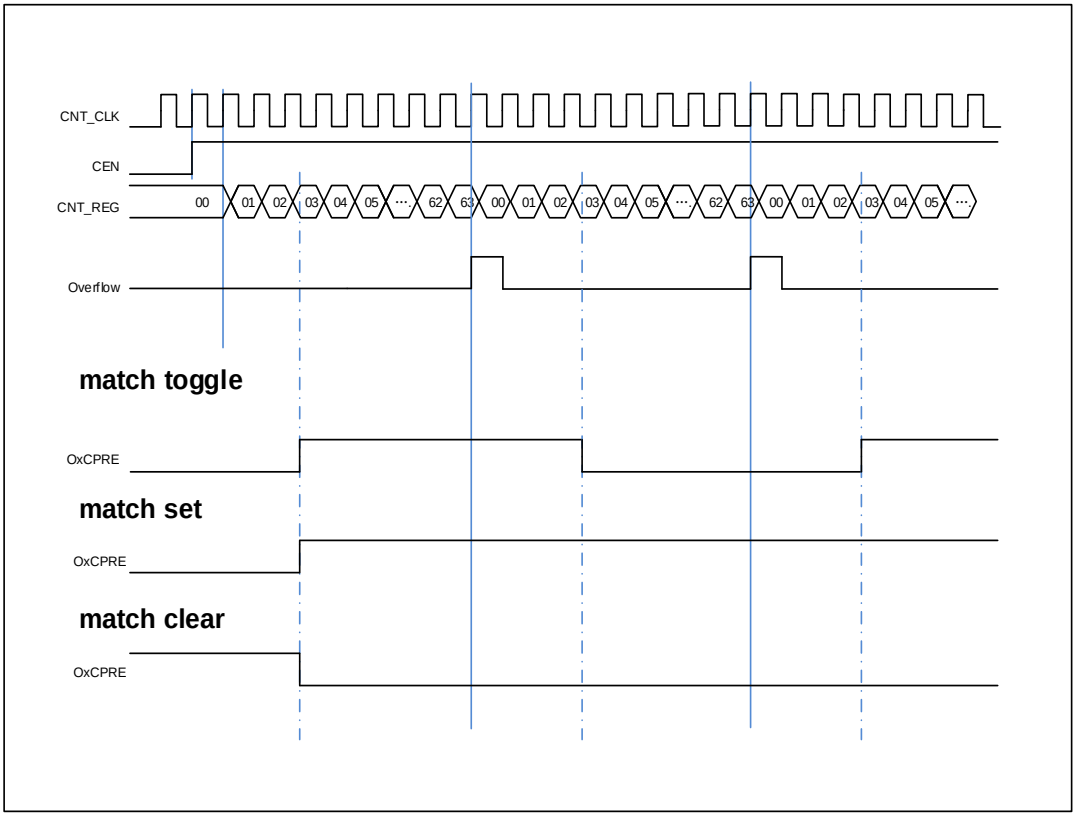
CxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 18-41. 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,

CxCV=0x3。

图 18-41. 三种输出比较模式



输出 PWM 功能

在PWM输出模式下（PWM模式0是配置CHxCOMCTL为3'b110，PWM模式1是配置CHxCOMCTL为3'b111），通道根据TIMERx_CAR寄存器和TIMERx_CHxCV寄存器的值，输出PWM波形。

根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 18-42. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 18-43. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR寄存器的值，通道输出一直为有效电平。

在PWM0模式下(CHxCOMCTL==3'b110)，如果TIMERx_CHxCV寄存器的值等于0，通道输出一直为无效电平。

图 18-42. EAPWM 时序图

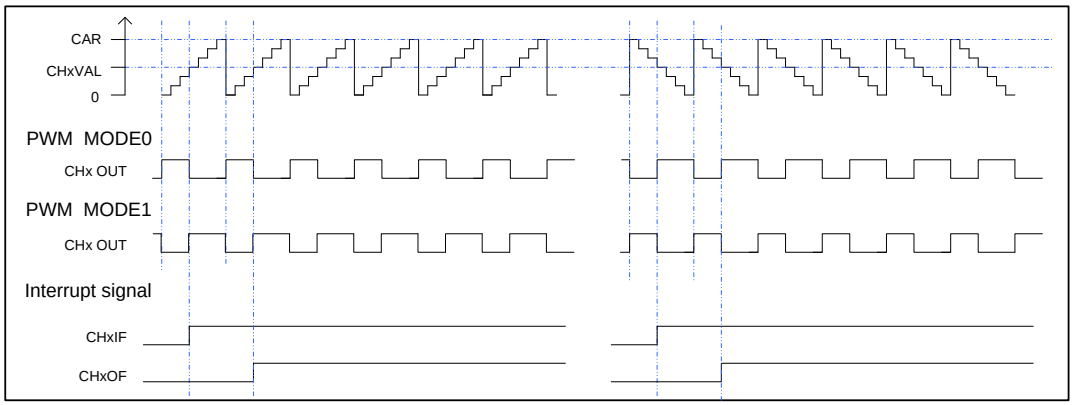
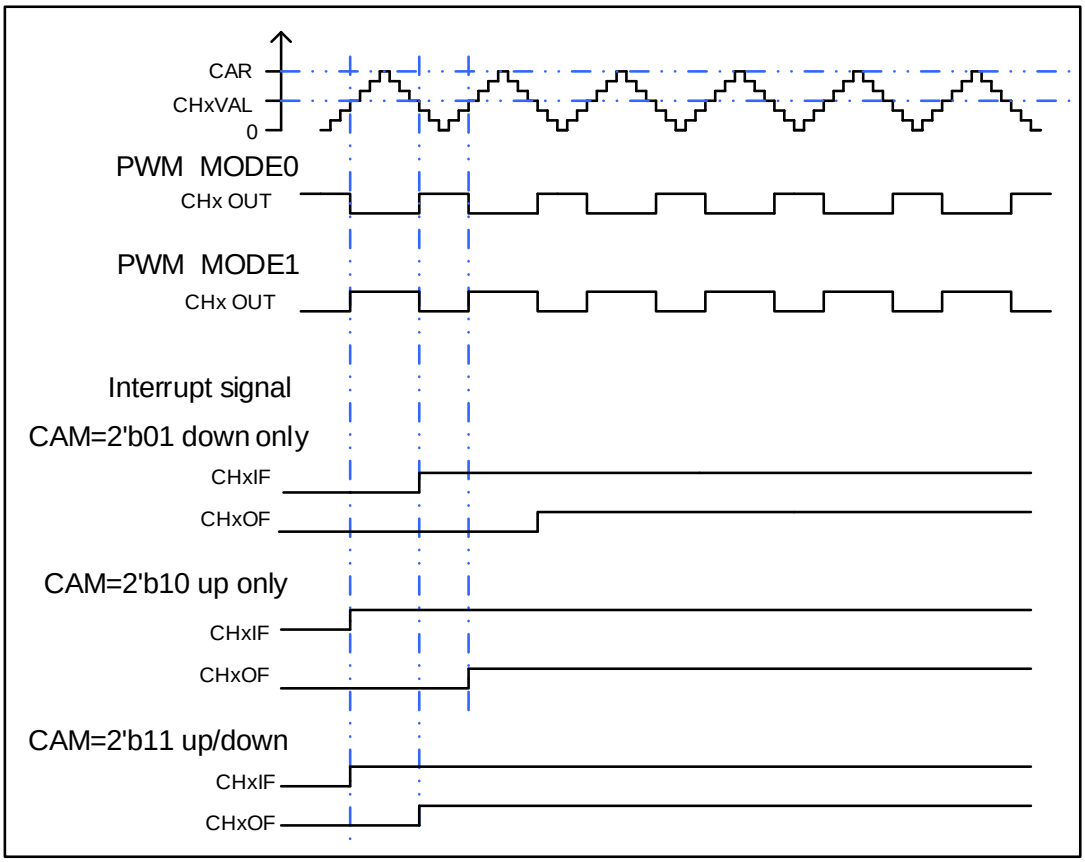


图 18-43. CAPWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存

器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

正交译码器

参考*正交译码器*。

霍尔传感器接口功能

参考*霍尔传感器接口功能*。

主-从管理

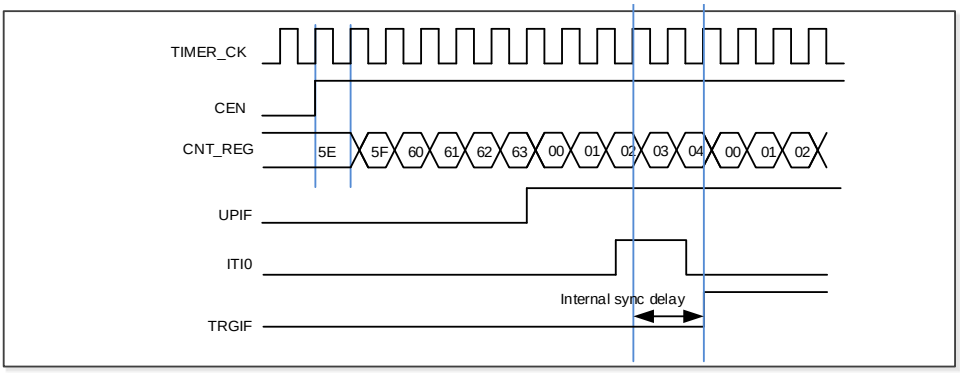
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS [2:0]来选择。

表 18-7. 从模式列表和举例

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

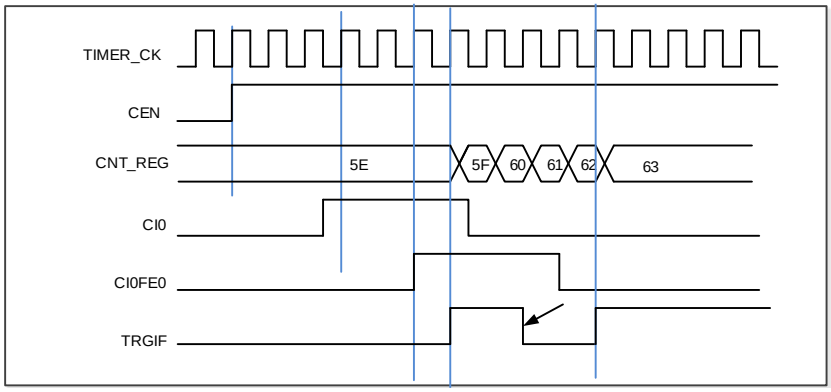
<table>
<tr><th></th><th>模式选择</th><th>触发源选择</th><th>极性选择</th><th>滤波和预分频</th></tr>
<tr><td></td><td colspan="4">

图 18-44. 复位模式下的控制电路

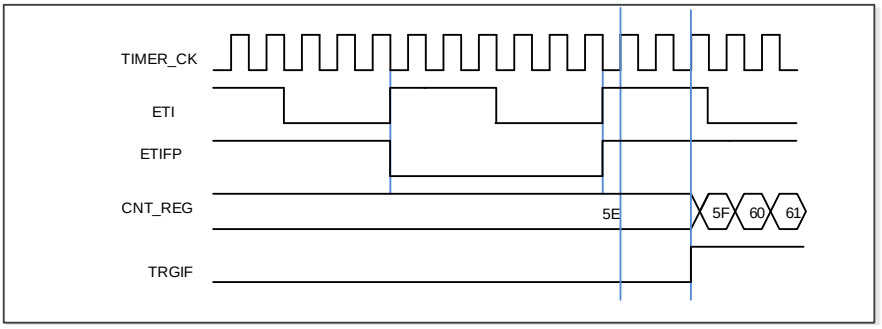


</td></tr>
<tr><td rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TRGIS[2:0]=3’b101
选择 CI0FE0 为触发源</td><td>TI0S=0.（非异或）
CH0P==0，不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">

图 18-45. 暂停模式下的控制电路



</td></tr>
<tr><td>例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TRGIS[2:0]=3’b111
选择ETIF为触发源.</td><td>ETP = 0 没有极性改变</td><td>ETPSC = 1, 2分频.
ETFC = 0，无滤波</td></tr>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频
	图 18-46. 事件模式下的控制电路 			

单脉冲模式

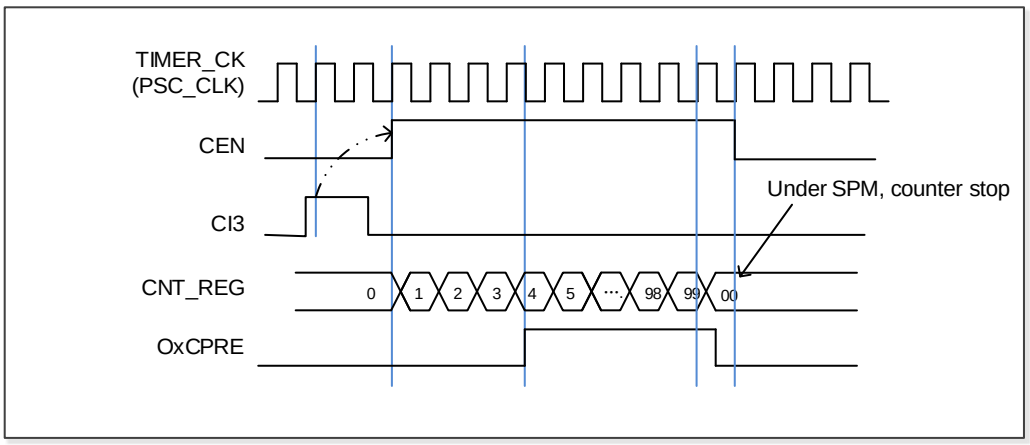
单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置CHxCOMCTL配置TIMERx为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM1或PWM2输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

*图18-47. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 18-47. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

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

*表18-8. 定时器x输入触发（x=1,2,3,4）*显示了内部触发，*表18-9. 定时器x输出触发（x=1,2,3,4）*显示了外部输出触发。

表 18-8. 定时器 x 输入触发（x=1,2,3,4）

	ITI0	ITI1	ITI2	ITI3
TIMER1	TIMER0_TGRO	TIMER1TGRO1_REMAP	TIMER2_TGRO	TIMER3_TGRO
TIMER2	TIMER0_TGRO	TIMER1_TGRO	TIMER4_TGRO	TIMER3_TGRO
TIMER3	TIMER0_TGRO	TIMER1_TGRO	TIMER2_TGRO	TIMER7_TGRO
TIMER4	TIMER1_TGRO	TIMER2_TGRO	TIMER3_TGRO	TIMER7_TGRO

TIMER1 的 ITI1 由 *AFIO 端口配置寄存器 0（AFIO_PCF0）*中的 TIMER1ITI1_REMAP 位来决定内部连接的信号源；

表 18-9. 定时器 x 输出触发（x=1,2,3,4）

	TIMER1_TGRO	TIMER2_TGRO	TIMER3_TGRO	TIMER4_TGRO
TIMER0	ITI1	ITI2	ITI3	ITI0
TIMER1	-	ITI2	ITI3	-
TIMER2	ITI1	-	ITI3	ITI2
TIMER3	ITI1	ITI2	-	-
TIMER4	ITI0	ITI1	ITI2	
TIMER7	ITI1	-	ITI2	ITI3
TIMER8	ITI0	ITI1	-	-
TIMER11	-	-	ITI0	ITI1

注释: '-' 表示没有内部级联。

其他例子参考*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 and TIMERx_DMATB。当然，必须要使能DMA请求，一些内部中断事件可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P模式，PADDR是TIMERx_DMATB寄存器地址，DMA就会访问TIMERx_DMATB寄存器。实际上，TIMERx_DMATB寄存器只是一个缓冲，定时器会将TIMERx_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器的发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

定时器调试模式

当Cortex®-M3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。

18.2.5. TIMERx 寄存器(x=1,2,3,4)

TIMER1基地址：0x4000 0000

TIMER2基地址：0x4000 0400

TIMER3基地址：0x4000 0800

TIMER4基地址：0x4000 0C00

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器 1：使能TIMERx_CAR寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1

		当计数器使能以后，该位不能从0x00切换到非0x00
4	DIR	方向 0：向上计数 1：向下计数 当计数器配置为中央对齐计数模式或编码器模式时，该位只读。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

保留	TI0S	MMC[2:0]	DMAS	保留.
	rw	rw	rw	

位/位域	名称	描述
31:8	保留	必须保持复位值
7	TI0S	通道0触发输入选择 0：选择TIMERx_CH0 引脚作为通道0的触发输入 1：选择TIMERx_CH0, CH1 and CH2 引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O3CPRE
3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道x的DMA请求 . 1：当更新事件发生，发送通道x的DMA请求
2:0	保留	必须保持复位值

从模式配置寄存器（TIMERx_SMCFG）

地址偏移：0x08

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			保留	SMC[2:0]		
rw	rw	rw		rw				rw	rw				rw		

位/位域	名称	描述

31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定 ETI 信号的极性 0：ETI 高电平或上升沿有效 . 1：ETI 低电平或下降沿有效 .
14	SMC1	SMC 的一部分为了使能外部时钟模式 1 在外部时钟模式 1，计数器由 ETIFP 信号上的任意有效边沿驱动 0：外部时钟模式 1 禁能 1：外部时钟模式 1 使能 当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3’b111。 如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF。 注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。
13:12	ETPSC[1:0]	外部触发预分频 外部触发信号 ETIFP 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。 00：预分频禁能 01：2 分频 10：4 分频 11：8 分频
11:8	ETFC[3:0]	外部触发滤波控制 外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 **f_{SAMP}** 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

EXTFC[3:0]	次数	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{TIMER_CK}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS_CK}/2$
4’b0101	8	
4’b0110	6	$f_{DTS_CK}/4$
4’b0111	8	
4’b1000	6	$f_{DTS_CK}/8$
4’b1001	8	
4’b1010	5	$f_{DTS_CK}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS_CK}/32$
4’b1110	6	
4’b1111	8	

7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP 从模式被使能后这些位不能改
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001：编码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 010：编码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 011：编码器模式 2. 根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/ 下计数 100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止 110：事件模式.计数器在触发输入的上升沿启动。 111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值.
14	TRGDEN	触发DMA请求使能 0：禁止触发DMA请求 1：使能触发DMA请求
13	保留	必须保持复位值.
12	CH3DEN	通道3比较/捕获 DMA请求使能 0：禁止通道3比较/捕获DMA请求 1：使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0：禁止通道2比较/捕获DMA请求 1：使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0：禁止通道1比较/捕获DMA请求 1：使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0：禁止通道0比较/捕获DMA请求 1：使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0：禁止更新DMA请求 1：使能更新DMA请求
7	保留	必须保持复位值.
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	保留	必须保持复位值.
4	CH3IE	通道3比较/捕获中断使能 0：禁止通道3中断 1：使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0：禁止通道2中断 1：使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0：禁止通道1中断 1：使能通道1中断
1	CH0IE	通道0比较/捕获中断使能

		0：禁止通道0中断 1：使能通道0中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值.
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道1捕获溢出标志 当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:7	保留	必须保持复位值.
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生

5	保留	必须保持复位值.
4	CH3IF	通道3比较/捕获中断标志 参见CH0IF描述
3	CH2IF	通道2比较/捕获中断标志 参见CH0IF描述
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0：无通道0中断发生 1：通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGG	保留	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值.
6	TRGG	触发事件产生 此位由软件置1，由硬件自动清0。当此位被置1，根据CCSE位(TIMERx_CTL1寄存器中)的值，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0：无触发事件产生 1：产生触发事件
5	保留	必须保持复位值.

4	CH3G	通道3捕获或比较事件发生 参见CH0G描述
3	CH2G	通道2捕获或比较事件发生 参见CH0G描述
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式

		参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不

		能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 0 配置为输入，IS0 映射在 ITS 上. 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

通道控制寄存器 1（TIMERx_CHCTL1）

地址偏移：0x1C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 注意：当 CH3MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。

		111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00（比较模式）时此位不能被改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00 时此位不能被改变。
2	CH2COMFEN	通道 2 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。 0：禁止通道 2 输出比较快速. 1：使能通道 2 输出比较快速。
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。 00：通道 2 配置为输出 01：通道 2 配置为输入，IS2 映射在 CI2FE2 上 10：通道 2 配置为输入，IS2 映射在 CI3FE2 上 11：通道 2 配置为输入，IS2 映射在 ITS 上. 注意：当 CH2MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH3CAPPSC[1:0]	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH3MS[1:0]	通道 3 模式选择 与输出模式相同

7:4 CH2CAPFLT[3:0] 通道 2 输入捕获滤波控制

CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。

数字滤波器的基本原理：根据 f_{SAMP} 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。

滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH2CAPPSC[1:0] 通道 2 输入捕获预分频器

这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH2MS[1:0] 通道 2 模式选择

与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

保留	CH3P	CH3EN	保留	CH2P	CH2EN	保留	CH1P	CH1EN	保留	CH0P	CH0EN
	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	CH3P	通道3极性 参考CH0P描述
12	CH3EN	通道3使能 参考CH0EN 描述
11:10	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
7:6	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
3:2	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IS0信号极性 0：上升沿。输入捕获发生在IS0的上升沿；用作外部触发器时IS0不反相。 1：下降沿。输入捕获发生在IS0的下降沿；用作外部触发器时IS0反相。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

CARL[15:0]
rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 注意：在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

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

通道 2 捕获/比较值寄存器（TIMERx_CH2CV）

地址偏移：0x3C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

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

通道 3 捕获/比较值寄存器（TIMERx_CH3CV）

地址偏移：0x40
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

DMA 配置寄存器（TIMERx_DMACFG）

地址偏移：0x48
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:14	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5’b0_0000 到 5’b1_0001
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当通过TIMERx_DMATB第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器（TIMERx_DMATB）

地址偏移：0x4C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

18.3. 通用定时器 L1（TIMERx, x=8,11）

18.3.1. 简介

通用定时器L1(定时器8/11)是两通道定时器，支持输入捕获和输出比较，可以产生PEM信号控制电机和电源管理。通用定时器L1含有一个16位无符号计数器。

通用定时器L1是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

定时器和定时器之间是相互独立，但是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。

通用定时器L1(定时器8/11)仅可用于超高密度(XD)产品中。

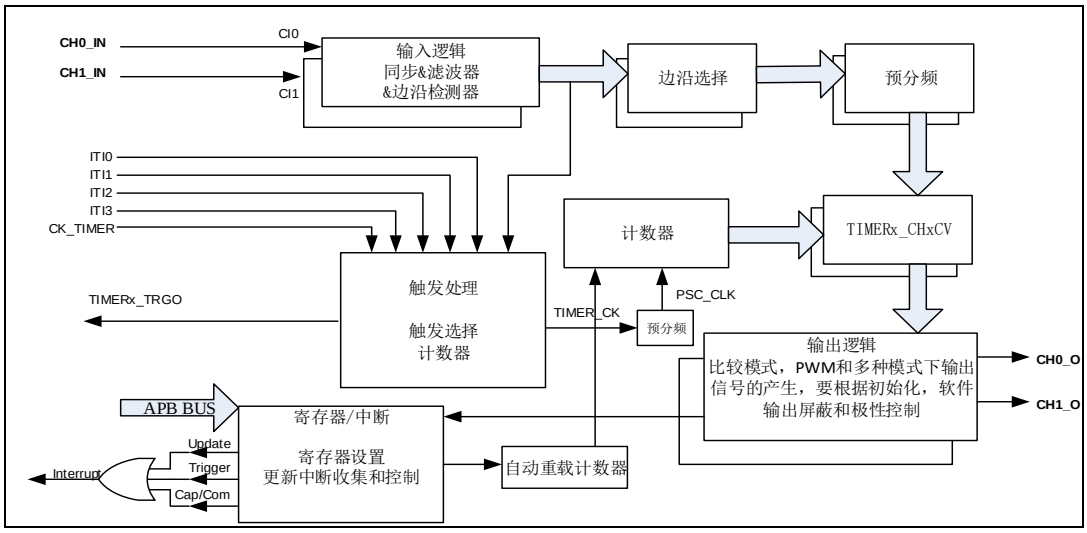
18.3.2. 主要特征

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央对齐计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊型链接使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

18.3.3. 结构框图

*图 18-48. 通用定时器 L1 结构框图*提供了通用定时器 L1 的内部配置细节。

图 18-48. 通用定时器 L1 结构框图



18.3.4. 功能说明

时钟源配置

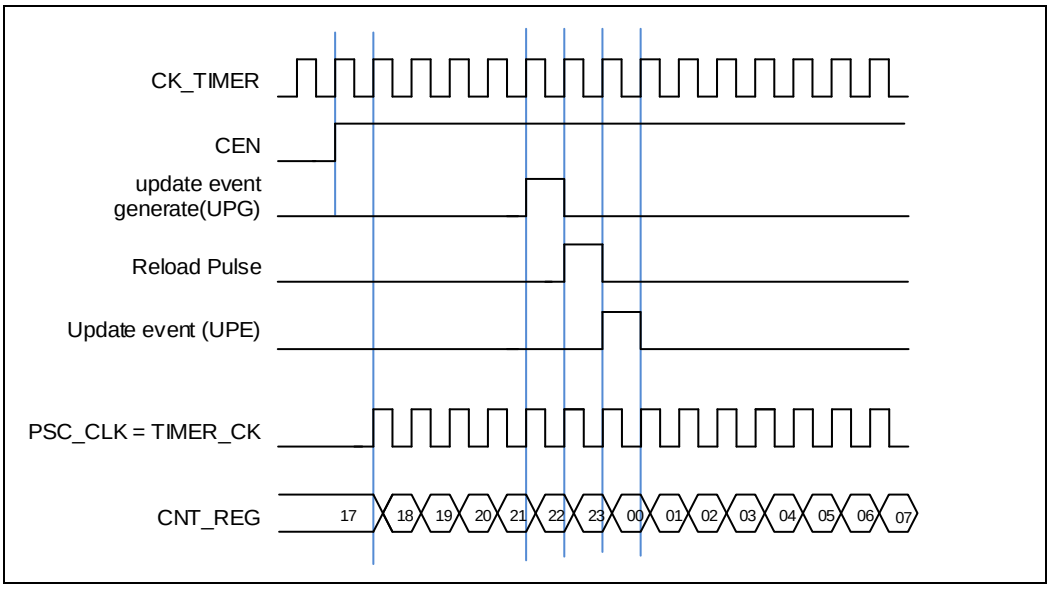
通用定时器L1可以由内部时钟源CK_TIMER或者由SMC（TIMERx_SMCFG寄存器位[2:0]）控制的复用时钟源驱动。

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

如果SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源CK_TIMER。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

如果将TIMERx_SMCFG寄存器的SMC[2:0]设置为0x1、0x2、0x3和0x7，预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动，在下文说明。当SMC位被设置为0x4、0x5和0x6，计数器预分频器时钟源由内部时钟CK_TIMER驱动。

图 18-49. 内部时钟分频为 1 时，计数器的时序图



■ SMC[2:0]==3’b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

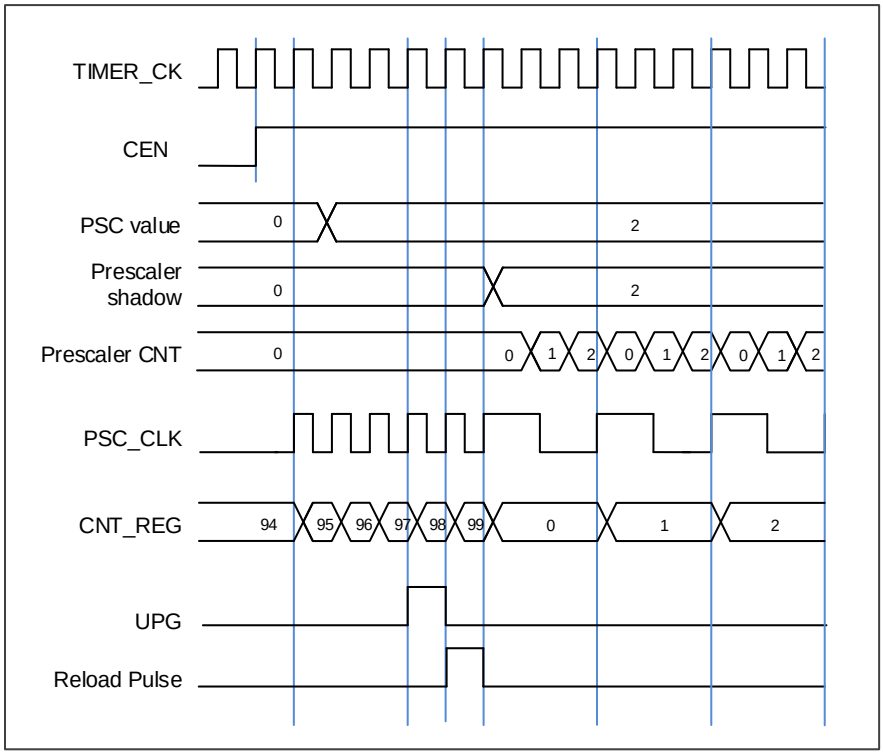
计数器预分频器可以在TIMERx_CI0/ TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x4， 0x5或0x6来选择。CIx 是TIMERx_CIx通过数字滤波器采样后的信号。

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

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 18-50. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-51. 向上计数时序图，PSC=0/2*和*图18-52. 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 18-51. 向上计数时序图，PSC=0/2

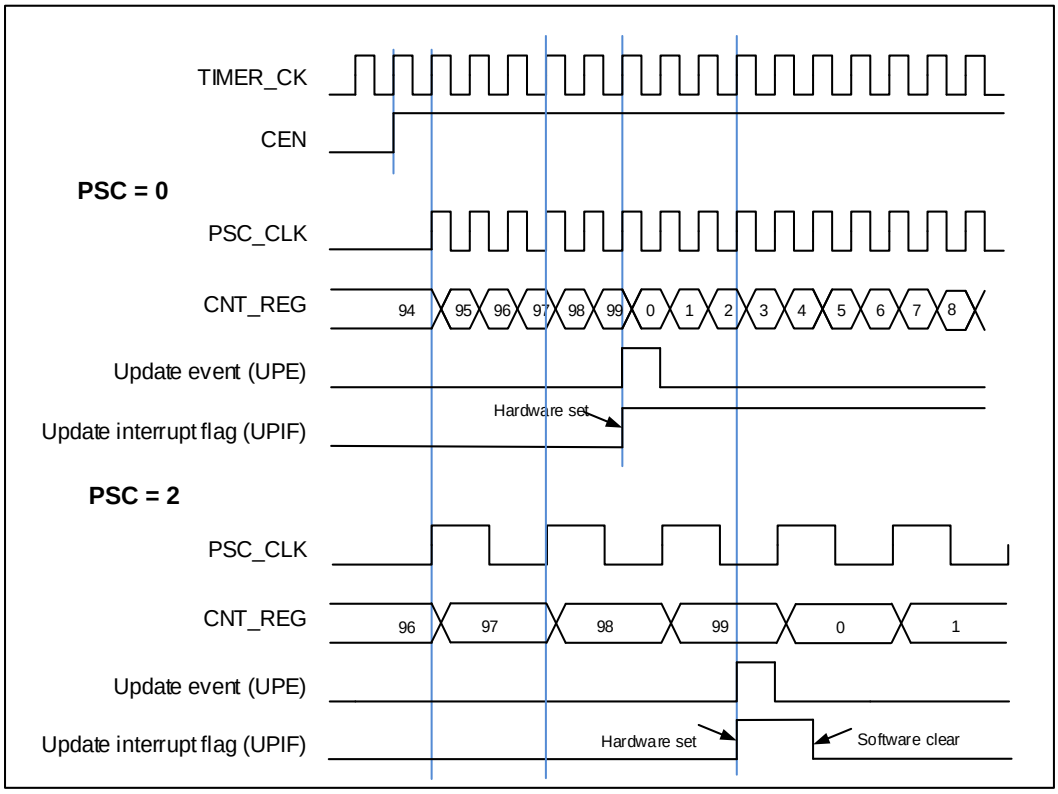
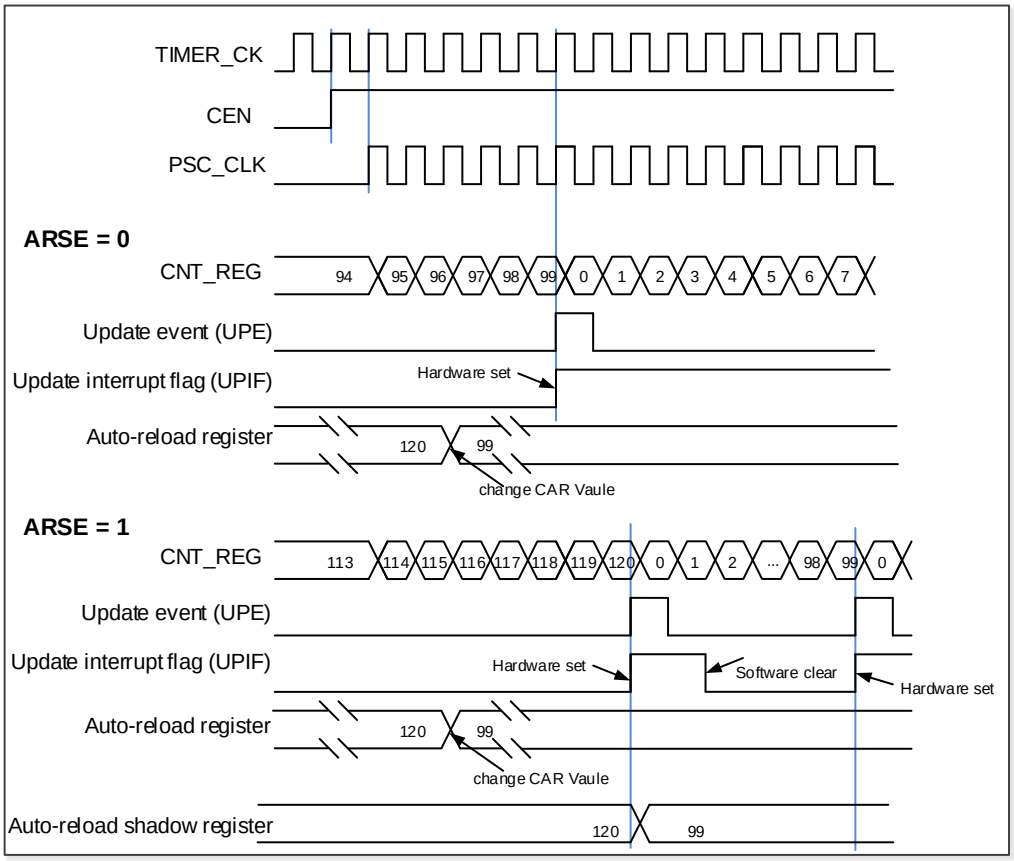


图 18-52. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

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

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-53. 向下计数时序图，PSC=0/2*和*图18-54. 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 18-53. 向下计数时序图，PSC=0/2

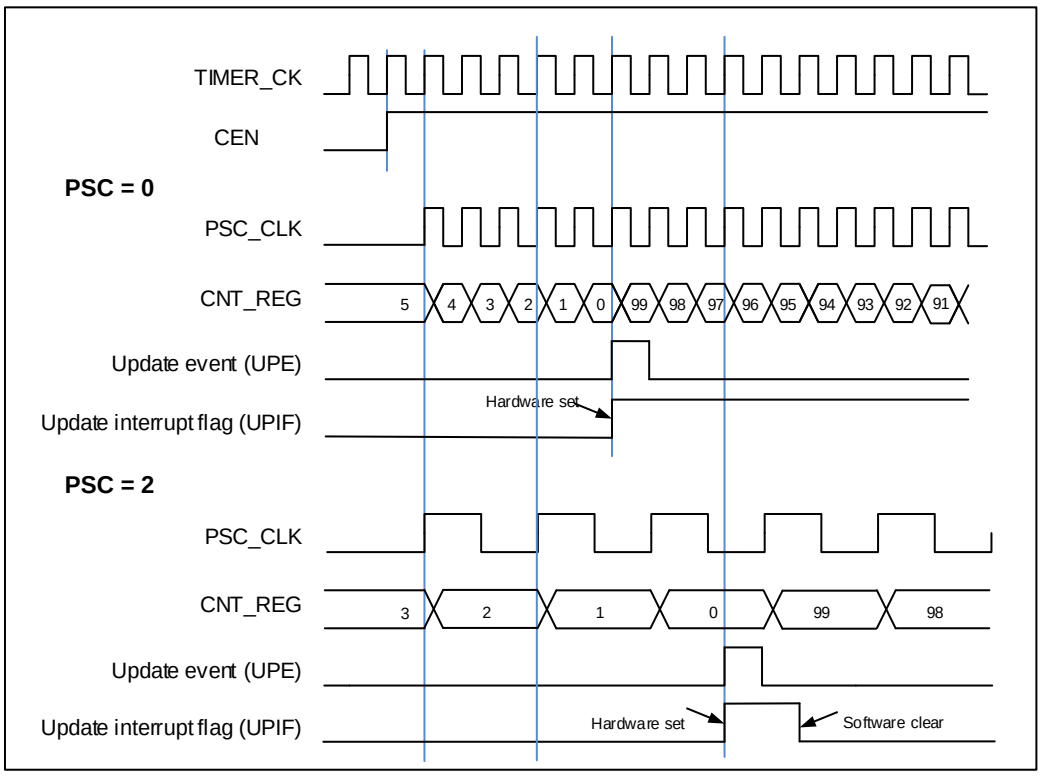
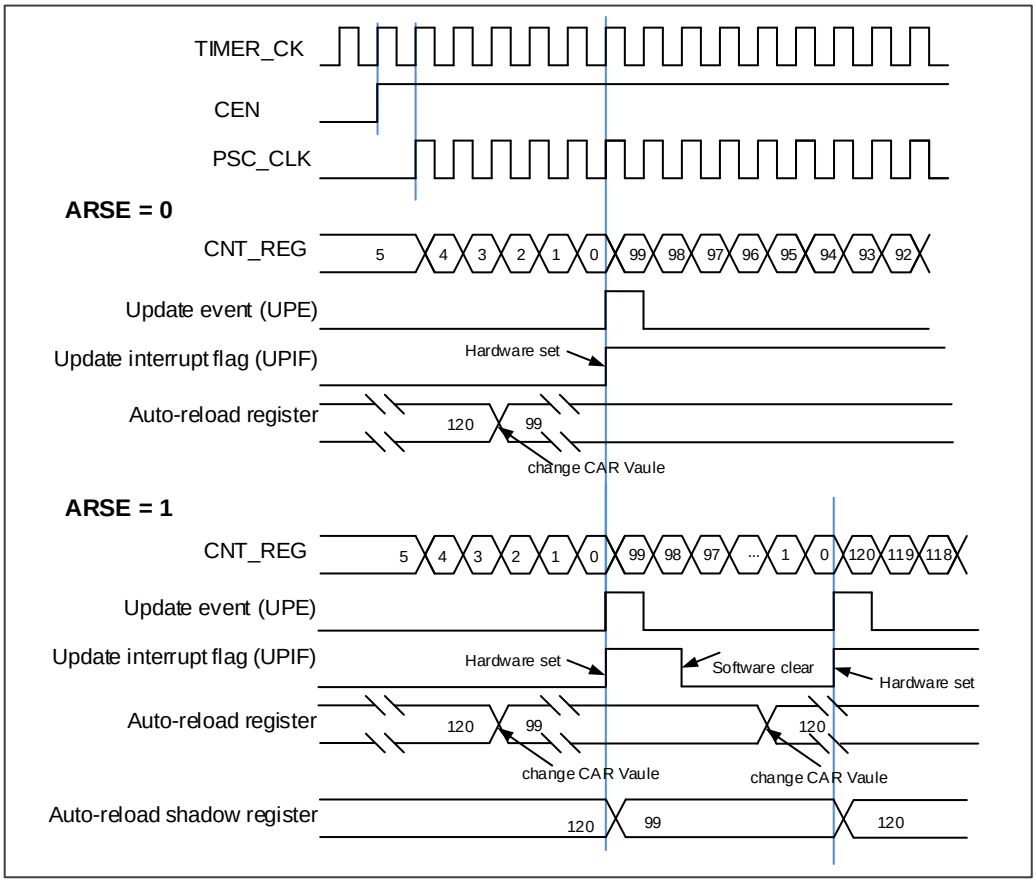


图 18-54. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

在中央对齐计数模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。向上计数模式中， 定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央对齐计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，指示了当前的计数方向。

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

上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1，然而CHxIF位置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考*图18-55. 中央对齐计数模式计数器时序图*。

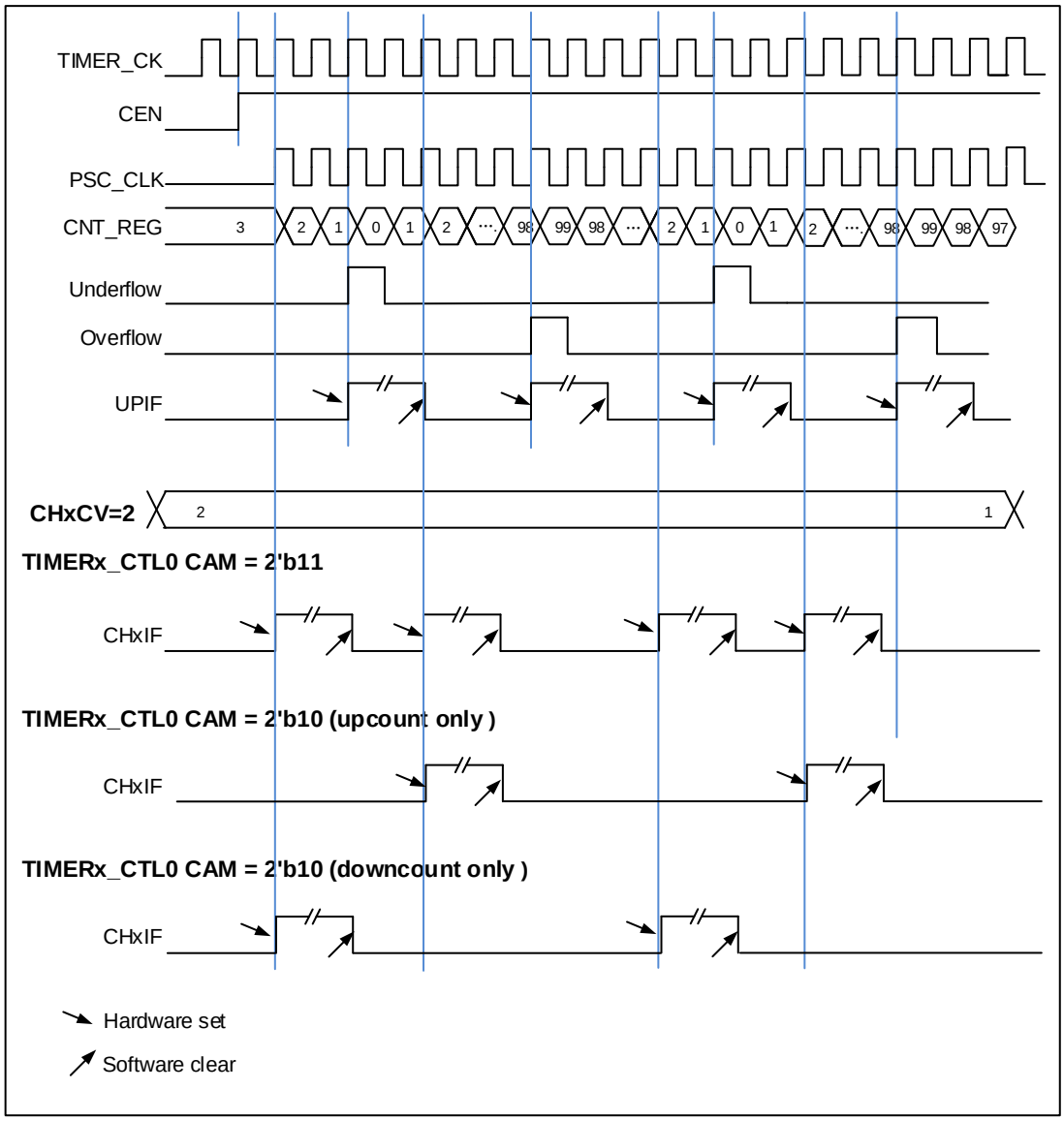
如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 18-55. 中央对齐计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，

TIMERx_PSC=0x0 时，计数器的行为。

图 18-55. 中央对齐计数模式计数器时序图



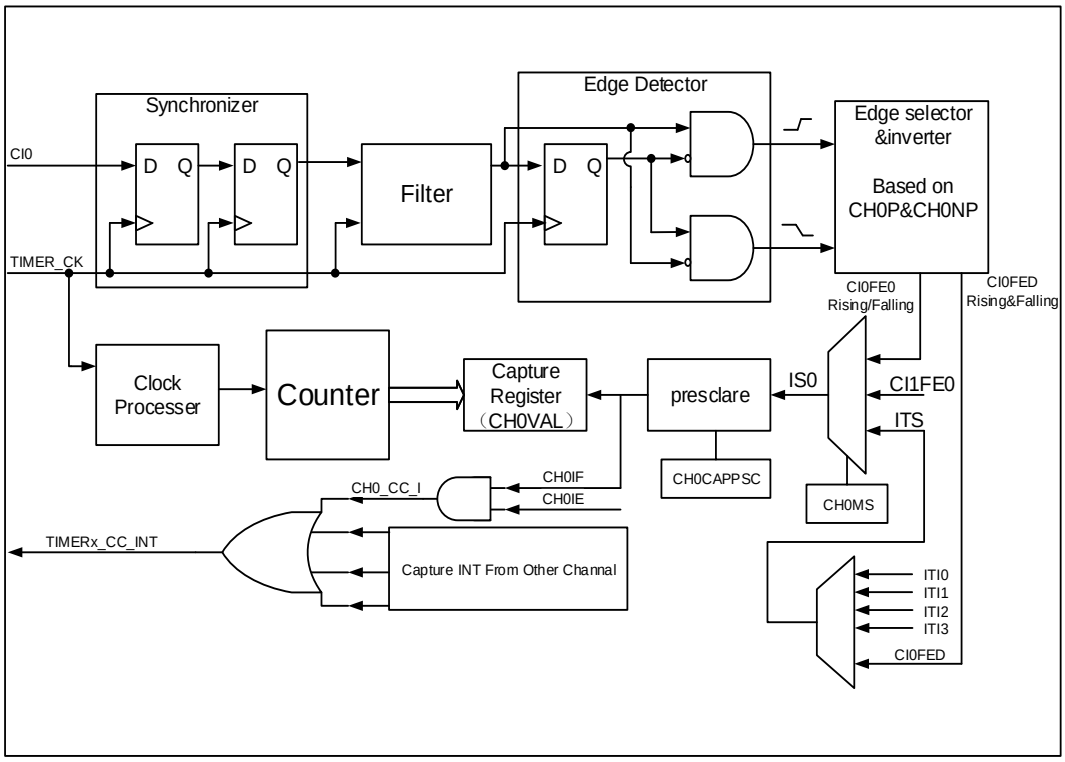
输入捕获和输出比较通道

通用定时器L1拥有两个个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，如果CHxIE = 1则产生通道中断。

图 18-56. 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，CxCV存储计数器的值。

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE and CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2’b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2’b10，选择通道1捕获信号为CI0

并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

■ 通道输出比较功能

图 18-57. 通道输出比较原理（x=0,1）

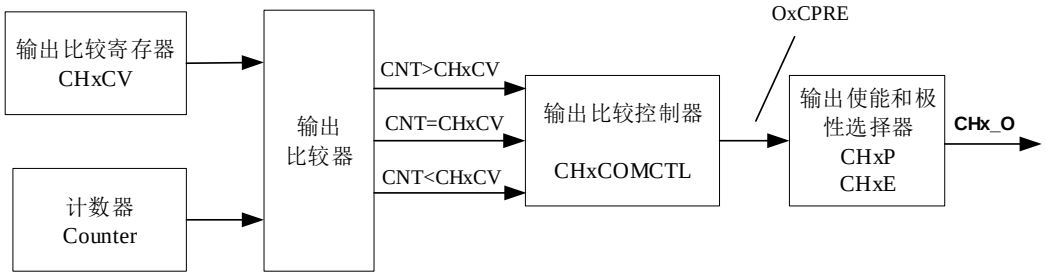


*图18-57. 通道输出比较原理（x=0,1）*给出了通道输出比较的原理电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出参考信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx_O输出使能）时：

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CxCV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CxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxCDE=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

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

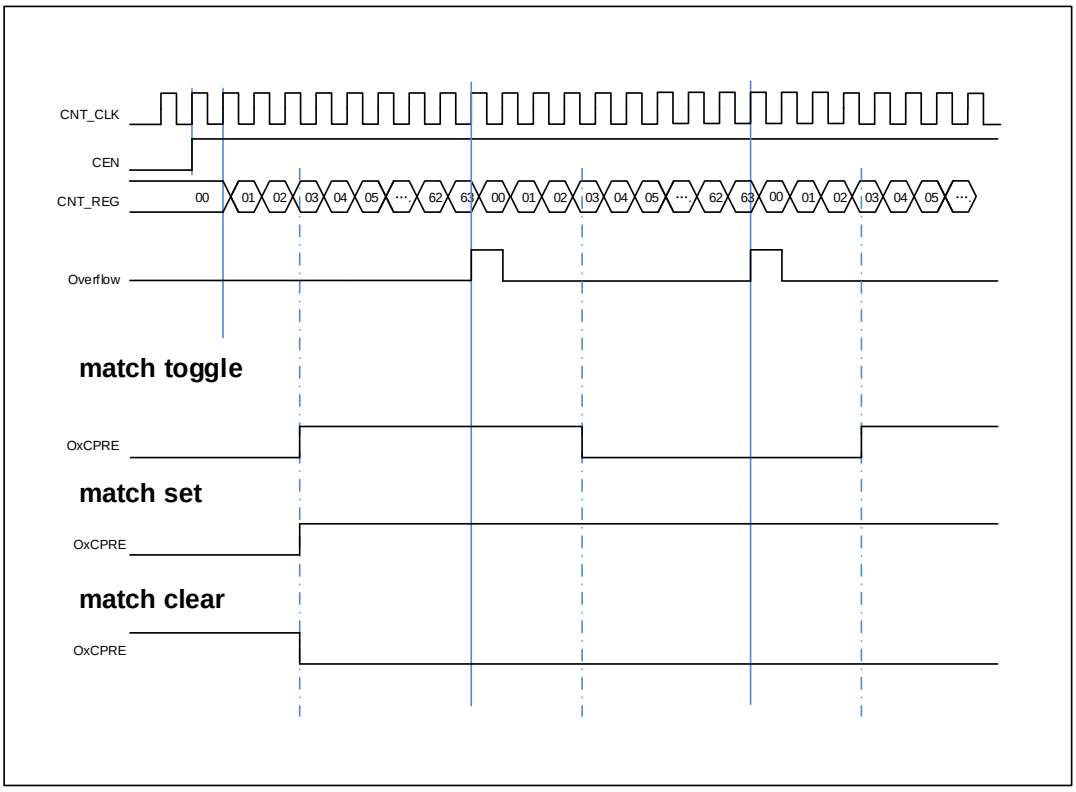
CxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 18-58. 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,

CxCV=0x3。

图 18-58. 三种输出比较模式



输出 PWM 功能

在PWM输出模式下（PWM模式0是配置CHxCOMCTL为3’b110，PWM模式1是配置CHxCOMCTL为3’b111），通道根据TIMERx_CAR寄存器和TIMERx_CHxCV寄存器的值，输出PWM波形。

根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 18-59. 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 18-60. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3’b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR寄存器的值，通道输出一直为有效电平。

在PWM0模式下(CHxCOMCTL==3’b110)，如果TIMERx_CHxCV寄存器的值等于0，通道输出一直为无效电平。

图 18-59. EAPWM 时序图

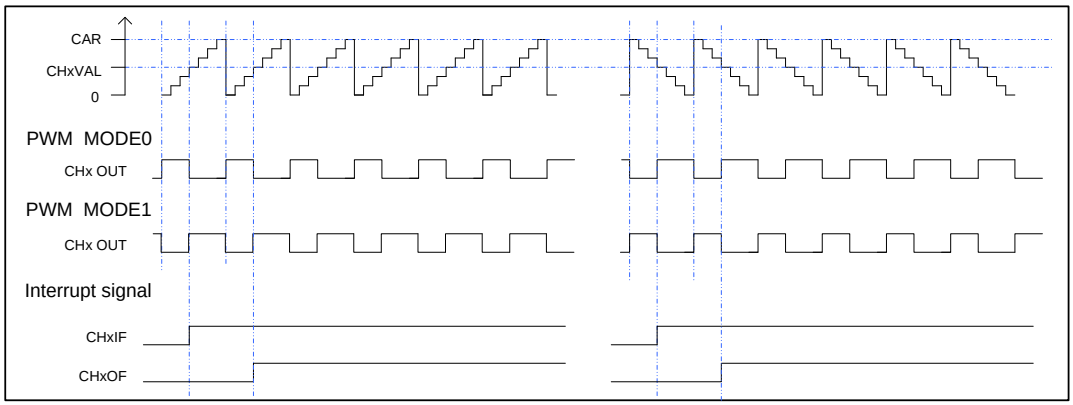
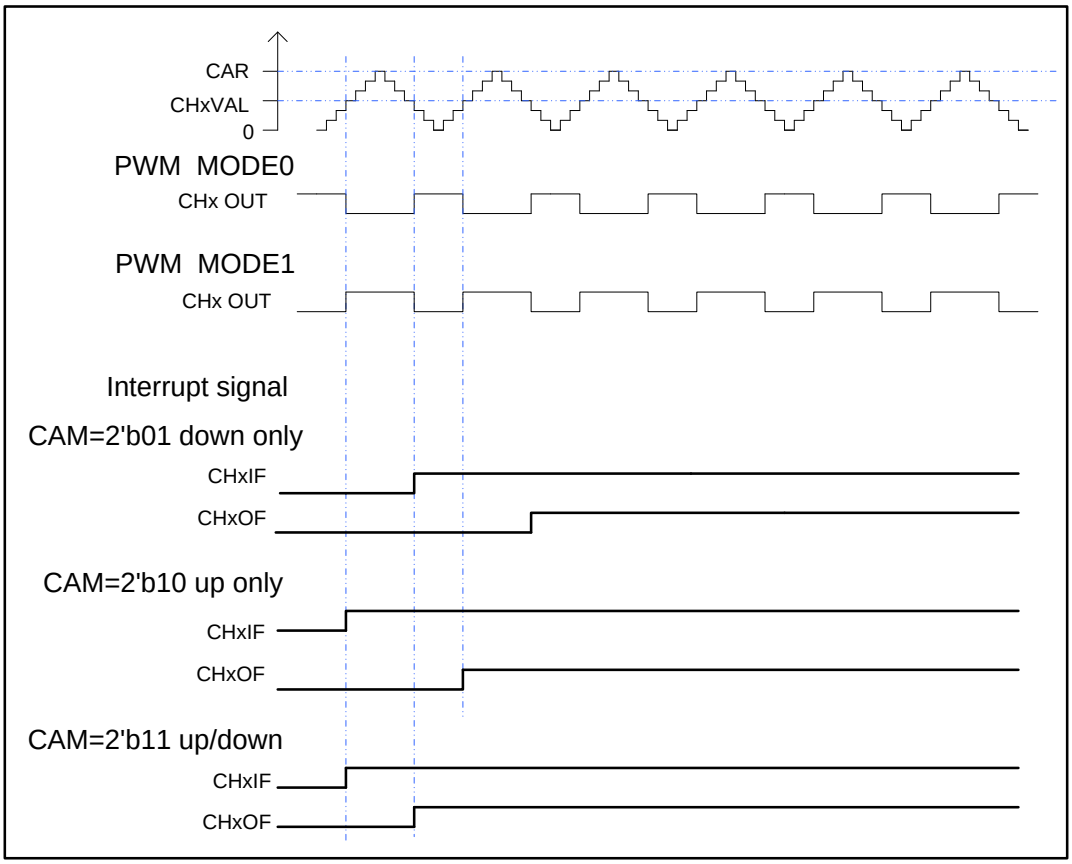


图 18-60. CAPWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存

器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

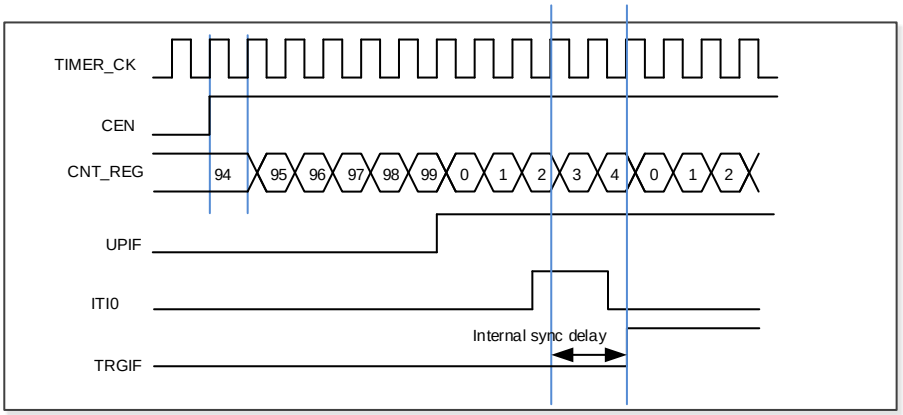
主-从管理

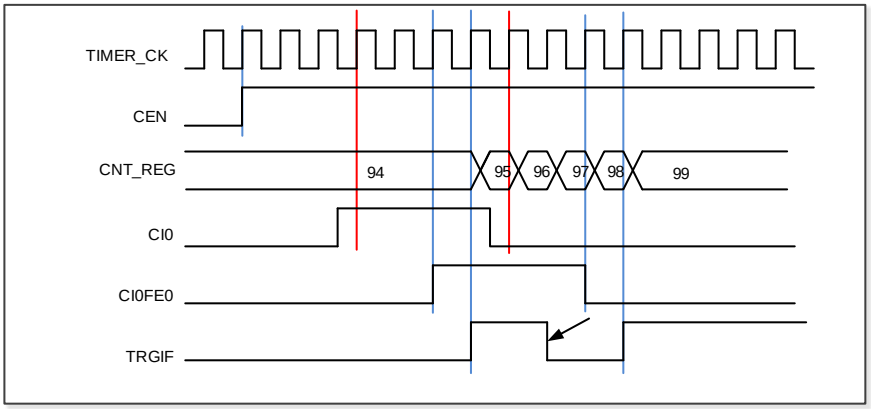
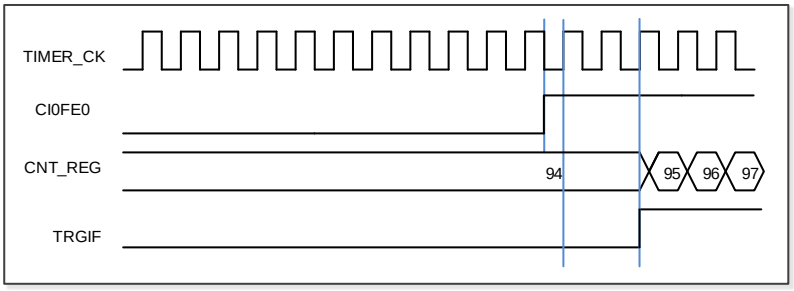
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS[2:0]来选择。

表 18-10. 从模式列表和举例

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

图 18-61. 复位模式下的控制电路



<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 rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源</td><td>CH0P==0，不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 18-62. 暂停模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源</td><td>CH0P==0，不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 18-63. 事件模式下的控制电路
 
</td></tr>
</table>

单脉冲模式

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

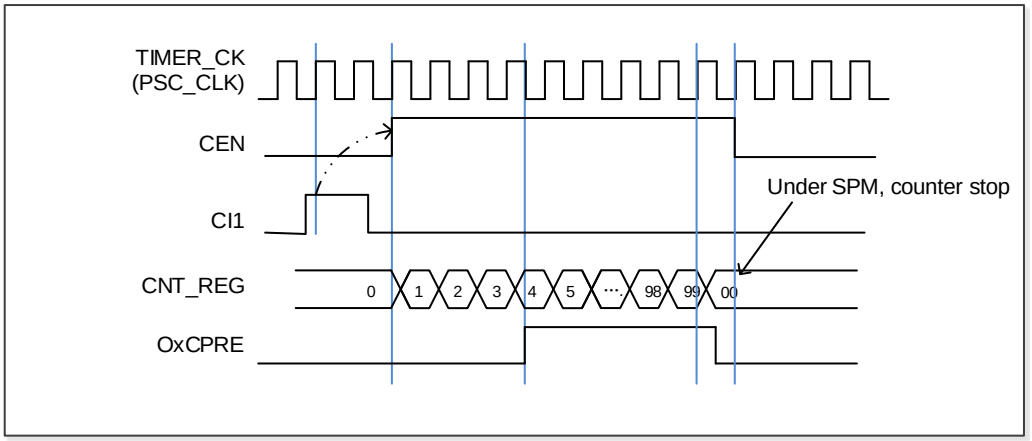
一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户

可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM1或PWM2输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

*图18-64. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 18-64. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

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

*表18-11. 定时器x输入触发（x=8,11）*显示了内部触发。

表 18-11. 定时器 x 输入触发（x=8,11）

	ITI0	ITI1	ITI2	ITI3
TIMER8	TIMER1_TGRO	TIMER2_TGRO	TIMER9_TGRO	TIMER10_TGRO
TIMER11	TIMER3_TGRO	TIMER4_TGRO	TIMER12_TGRO	TIMER13_TGRO

其他例子参考*定时器互连*。

定时器调试模式

当Cortex®-M3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

18.3.5. TIMERx 寄存器(x=8,11)

TIMER8基地址：0x4001 4C00

TIMER11基地址：0x4001 1800

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器 1：使能TIMERx_CAR寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1 当计数器使能以后，该位不能从0x00切换到非0x00
4	DIR	方向

		0：向上计数 1：向下计数 当计数器配置为中央对齐计数模式或编码器模式时，该位只读。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0： 计数器禁能 1： 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。

从模式配置寄存器（TIMERx_SMCFG）

地址偏移：0x08

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过TRIGI和TRGO，定时器被连接在一起，TRGO用做启动事件。 0：主从模式禁能 1：主从模式使能
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留 从模式被使能后这些位不能改
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 000：关闭从模式。如果CEN=1，则预分频器直接由内部时钟驱动。 001：保留。 010：保留。 011：保留。 100： 复位模式。选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 101：暂停模式。当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止。 110：事件模式。计数器在触发输入的上升沿启动。 111：外部时钟模式0。选中的触发输入的上升沿驱动计数器。

中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGIE	保留			CH1IE	CH0IE	UPIE
									rw				rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值.
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5:3	保留	必须保持复位值.
2	CH1IE	通道1比较/捕获中断使能 0：禁止通道1中断 1：使能通道1中断
1	CH0IE	通道0比较/捕获中断使能 0：禁止通道0中断 1：使能通道0中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CH1OF	CH0OF	保留		TRGIF	保留			CH1IF	CH0IF	UPIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:11	保留	必须保持复位值.
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道1捕获溢出标志 当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0. 0： 无捕获溢出中断发生

		1： 发生了捕获溢出中断
8:7	保留	必须保持复位值.
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生
5:3	保留	必须保持复位值.
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0：无通道0中断发生 1：通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	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			CH1G	CH0G	UPG
									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
5:3	保留	必须保持复位值.
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述

11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 0 配置为输入，IS0 映射在 ITS 上 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}

4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

通道控制寄存器 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										CH1P	CH1EN	保留		CH0P	CH0EN
										rw	rw			rw	rw

位/位域	名称	名称
31:6	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
3:2	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IS0信号极性 0：上升沿。输入捕获发生在IS0的上升沿；用作外部触发器时IS0不反相。 1：下降沿。输入捕获发生在IS0的下降沿；用作外部触发器时IS0反相。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

保留

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

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

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 注意：在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

CH0VAL[15:0]
Rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

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

18.4. 通用定时器 L2（TIMERx, x=9,10,12,13）

18.4.1. 简介

通用定时器L2 (TIMER9/10/12/13)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。

通用定时器L2 (TIMER9/10/12/13)仅可用于超高密度(XD)产品中。

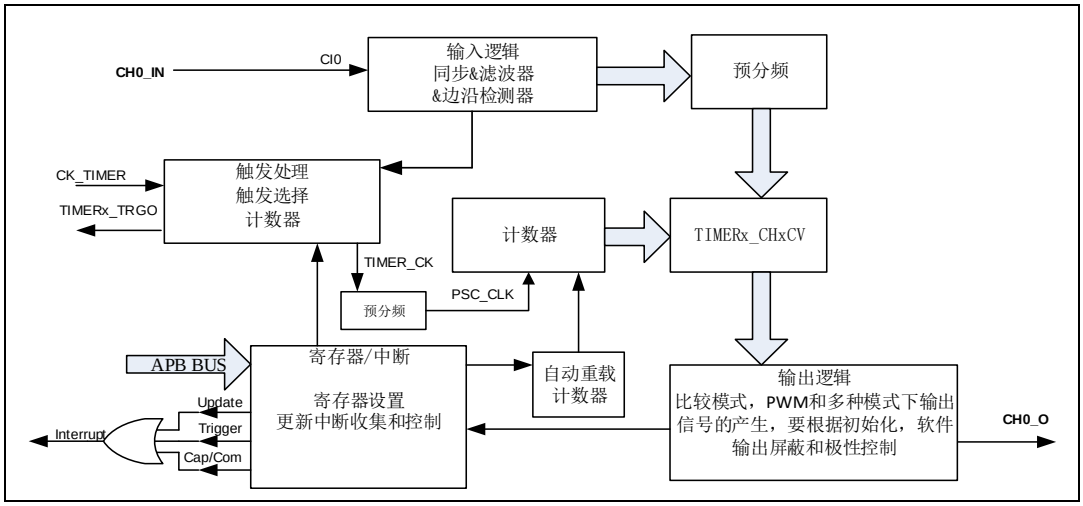
18.4.2. 主要特征

- 总通道数：1；
- 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，向下计数和中央对齐计数；
- 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；
- 自动重装载功能；
- 中断输出：更新事件，比较/捕获事件；

18.4.3. 结构框图

*图 18-65. 通用定时器 L2 结构框图*提供了通用定时器 L2 的内部配置细节

图 18-65. 通用定时器 L2 结构框图



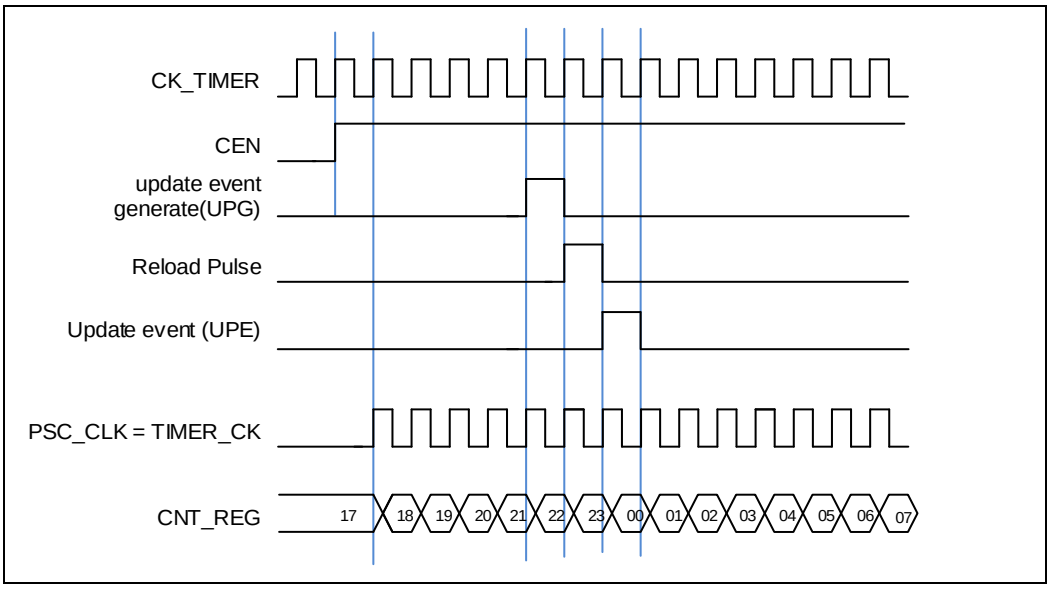
18.4.4. 功能说明

时钟源配置

通用定时器L2由内部时钟源CK_TIMER驱动

通用定时器L2仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

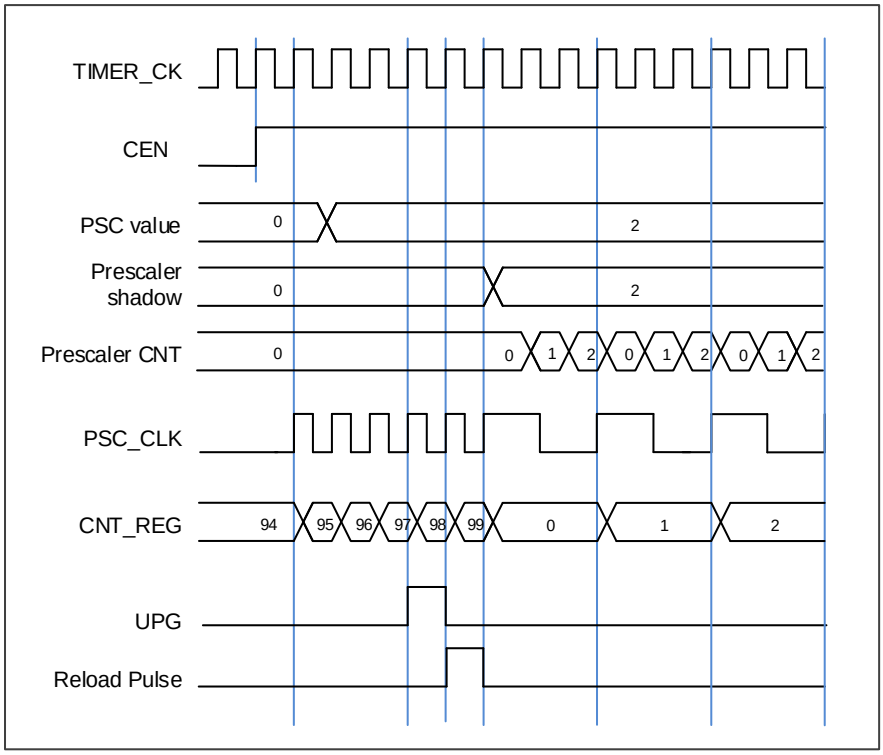
图 18-66. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 18-67. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

图 18-68. 向上计数时序图，PSC=0/2 和 *图 18-69. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频

因子下的行为。

图 18-68. 向上计数时序图，PSC=0/2

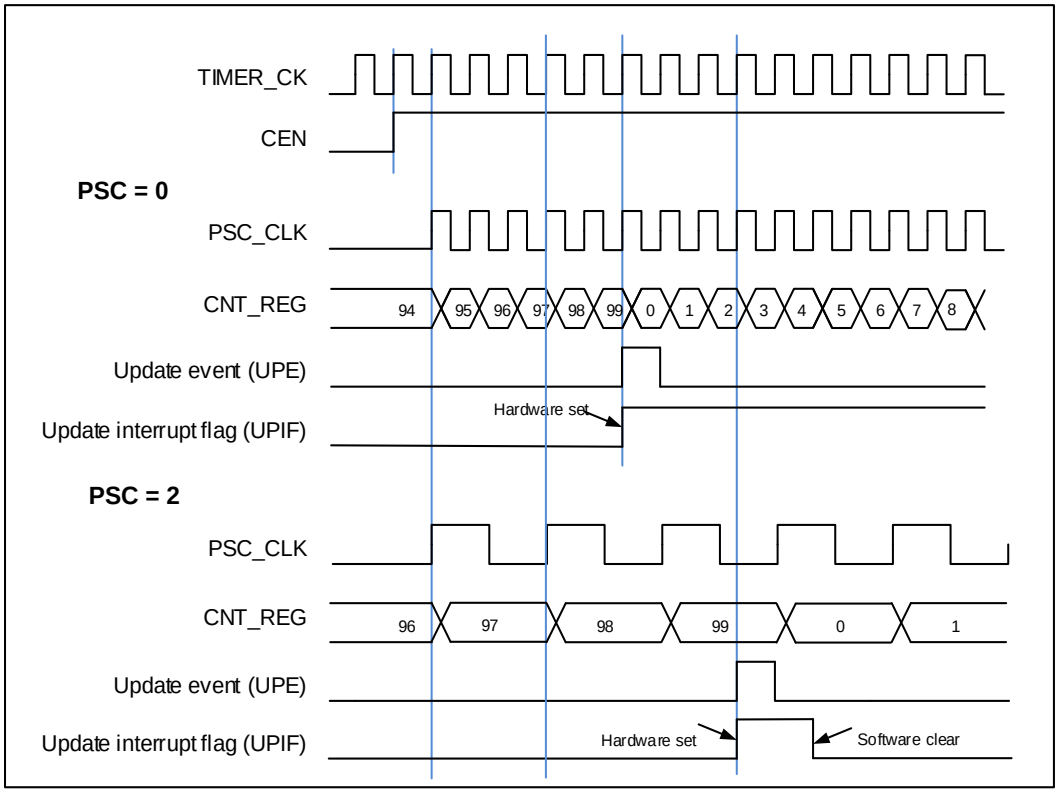
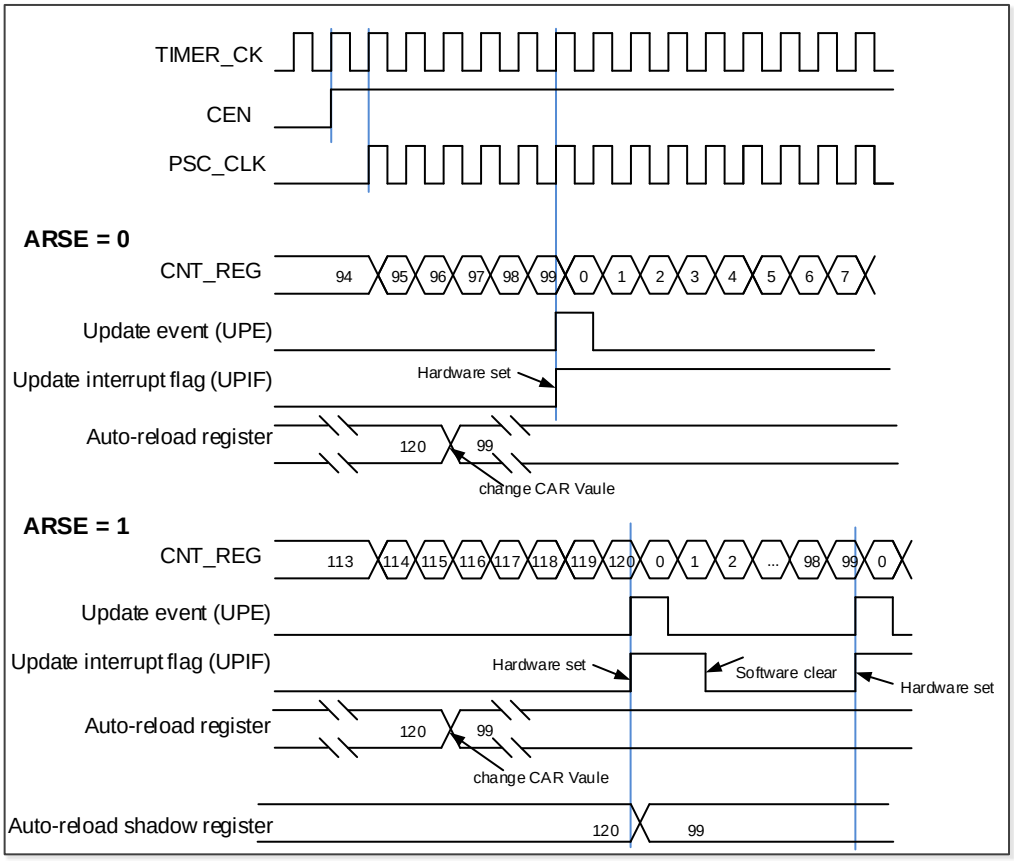


图 18-69. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

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

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-70. 向下计数时序图，PSC=0/2*和*图18-71. 向下计数时序图，在运行时改变TIMERx_CAR寄存器值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同时钟频率下的行为。

图 18-70. 向下计数时序图，PSC=0/2

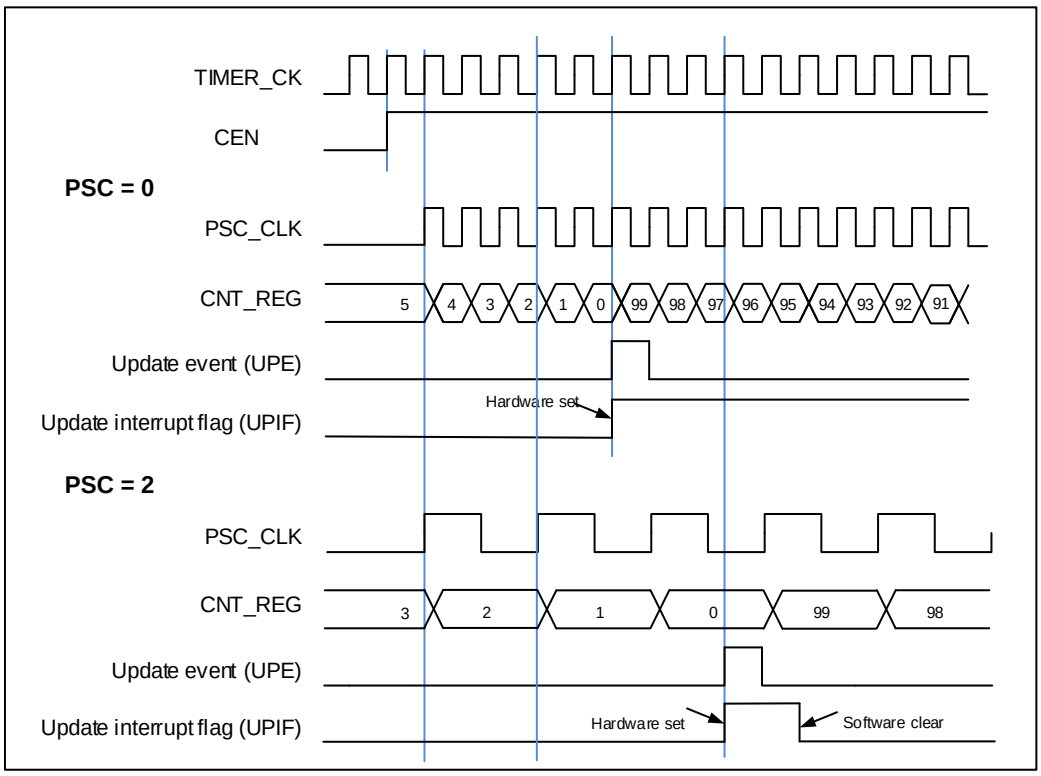
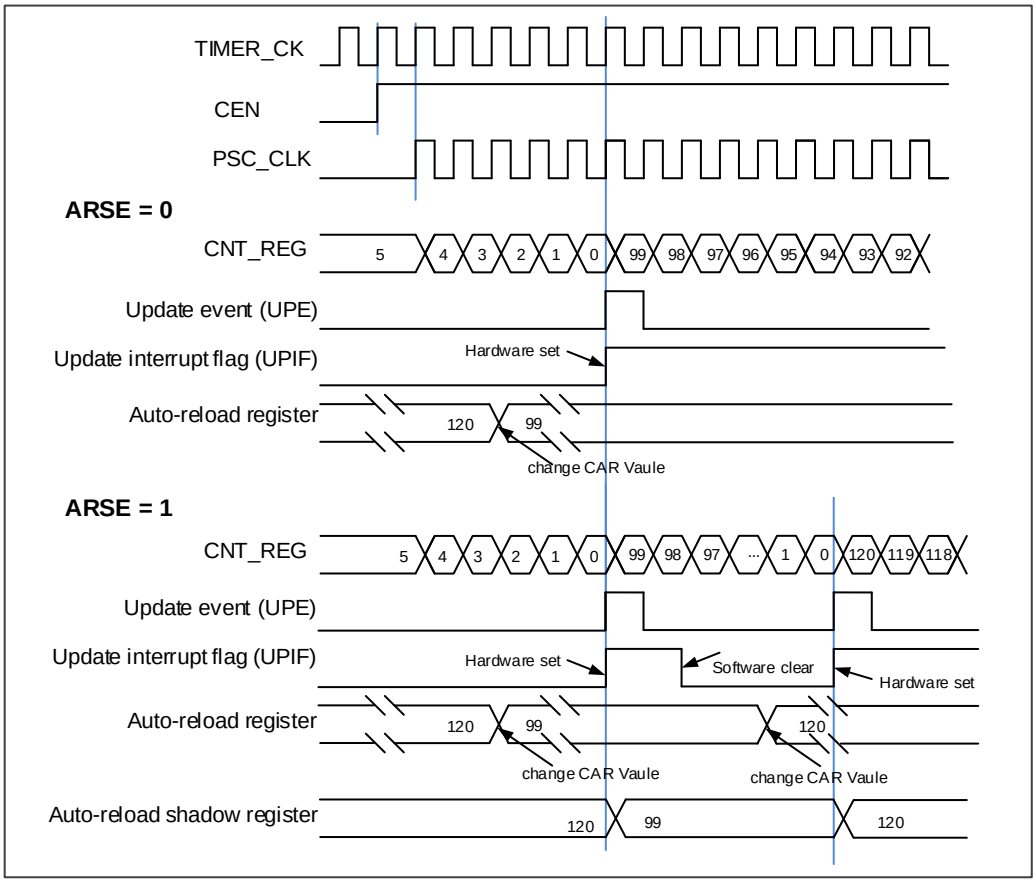


图 18-71. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

在中央对齐计数模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。向上计数模式中， 定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央对齐计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，指示了当前的计数方向。

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

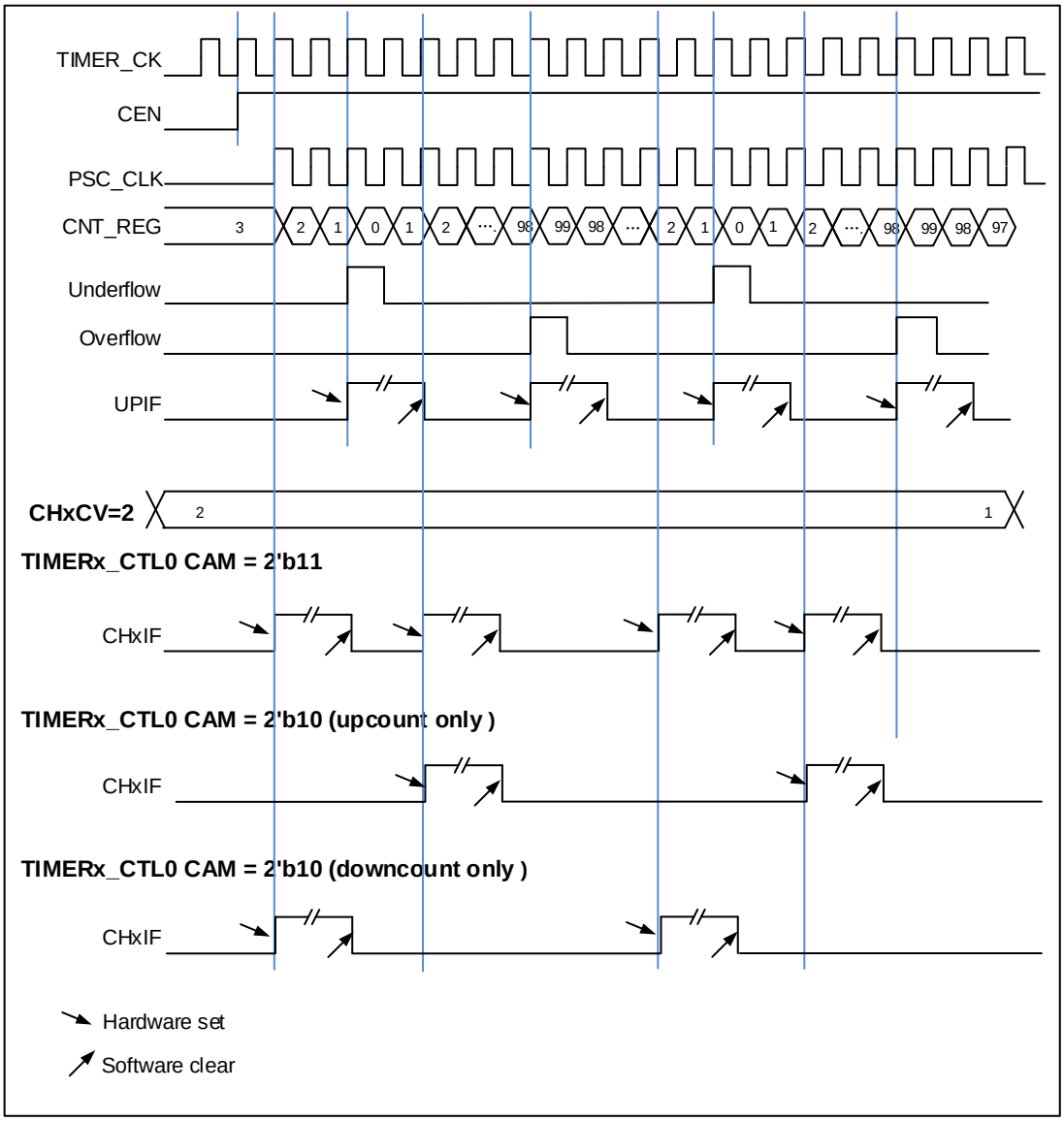
上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1，然而CHxIF位置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考*图18-72. 中央对齐计数模式计数器时序图*。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-72. 中央对齐计数模式计数器时序图*给出了一些例子，当TIMERx_CAR=0x99，TIMERx_PSC=0x0时，计数器的行为。

图 18-72. 中央对齐计数模式计数器时序图



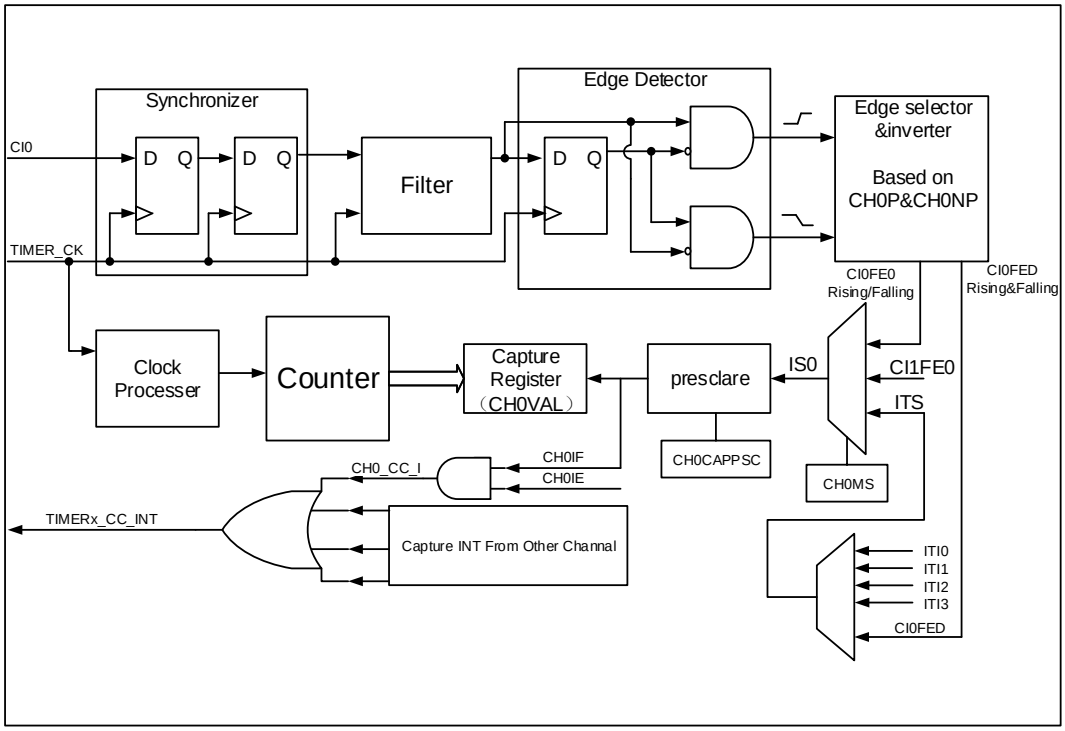
输入捕获和输出比较通道

通用定时器L2只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，如果CHxIE = 1则产生通道中断。

图 18-73. 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，CxCV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_CxCV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

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CxCV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CxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxCDE=1则会产生DMA请求。

配置步骤如下：

第一步： 时钟配置：
配置定时器时钟源，预分频器等。

第二步： 比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。

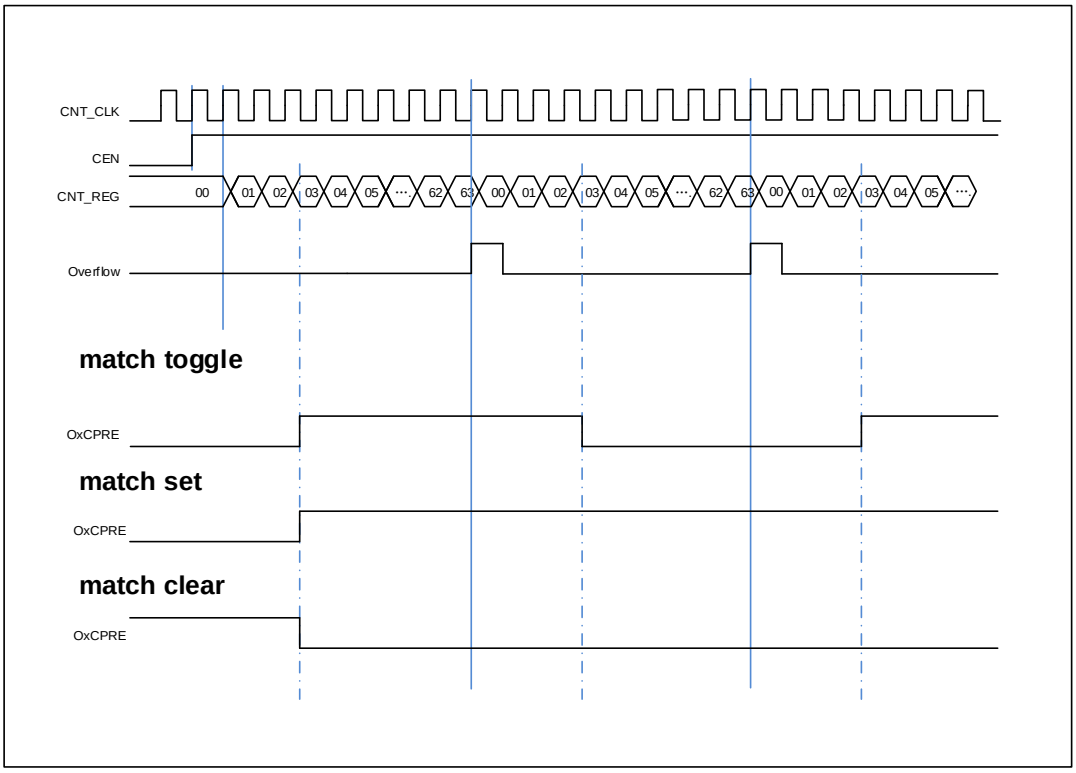
第三步： 通过CHxIE位配置中断使能。

第四步： 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
CxCV可以在运行时根据你所期望的波形而改变。

第五步： 设置CEN位使能定时器。

*图 18-74. 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, CxCV=0x3。

图 18-74. 三种输出比较模式



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

定时器调试模式

当Cortex®-M3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。

18.4.5. TIMERx 寄存器(x=9,10,12,13)

TIMER9基地址： 0x4001 5000

TIMER10基地址： 0x4001 5400

TIMER12基地址： 0x4000 1C00

TIMER13基地址： 0x4000 2000

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	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	保留	UPS	UPDIS	CEN
						rw		rw	rw		rw		rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER} /2$ 10：$f_{DTS}= f_{CK_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器 1：使能TIMERx_CAR寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1

		当计数器使能以后，该位不能从0x00切换到非0x00
4	DIR	方向 0：向上计数 1：向下计数 当计数器配置为中央对齐计数模式或编码器模式时，该位只读。
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、暂停模式和编码器模式才能工作。

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：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O3CPRE
3:0	保留	必须保持复位值

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：通道0配置为输入，IS0映射在CI0FE1上 11：通道 0 配置为输入，IS0 映射在 ITS 上 注意：当CH0MS[1:0]=11时，需要通过TRGS位（位于TIMERx_SMCFG寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值.
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2	CH0CAPPSC[1:0]	通道0输入捕获预分频器 这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每2个事件触发一次捕获 10：每4个事件触发一次捕获 11：每8个事件触发一次捕获
1:0	CH0MS[1:0]	通道0模式选择 与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0P	CH0EN
														rw	rw

位/位域	名称	描述
31:2	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IS0信号极性 0：上升沿。输入捕获发生在IS0的上升沿；用作外部触发器时IS0不反相。 1：下降沿。输入捕获发生在IS0的下降沿；用作外部触发器时IS0反相。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 注意：在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

18.5. 基本定时器（TIMERx, x=5,6）

18.5.1. 简介

基本定时器(定时器5/6)包含一个无符号16位计数器。可以被用作通用定时器和为DAC (数字到模拟转换器)提供时钟。基本定时器可以配置产生DMA请求，TRGO触发连接到DAC。

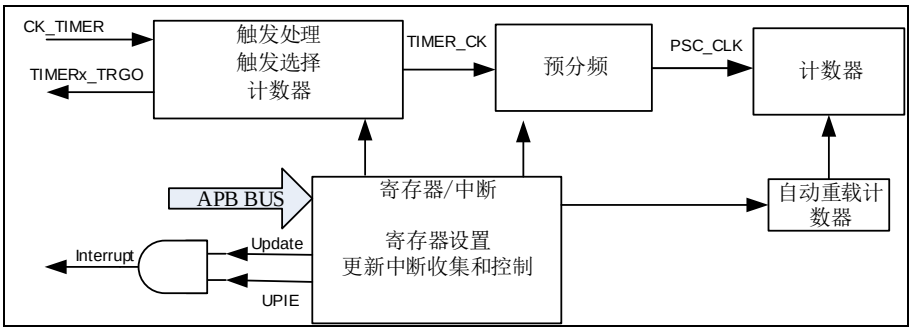
18.5.2. 主要特征

- 计数器宽度：16位；
- 时钟源只有内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 支持单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件；

18.5.3. 结构框图

*图 18-75. 基本定时器结构框图*提供了基本定时器内部配置的细节

图 18-75. 基本定时器结构框图



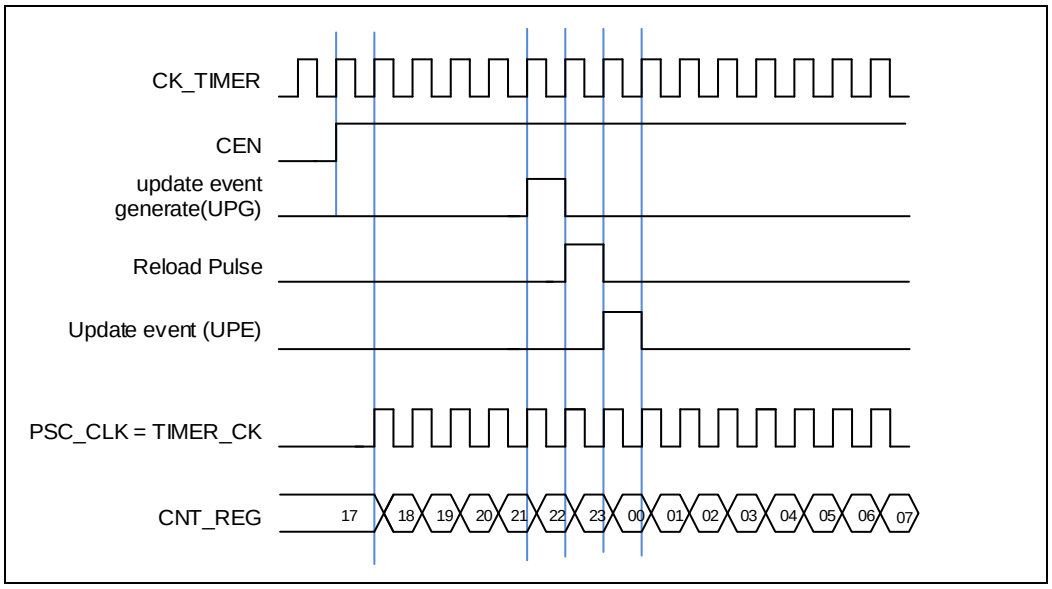
18.5.4. 功能说明

时钟源配置

基本定时器可以由内部时钟源CK_TIMER驱动。

基本定时器仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

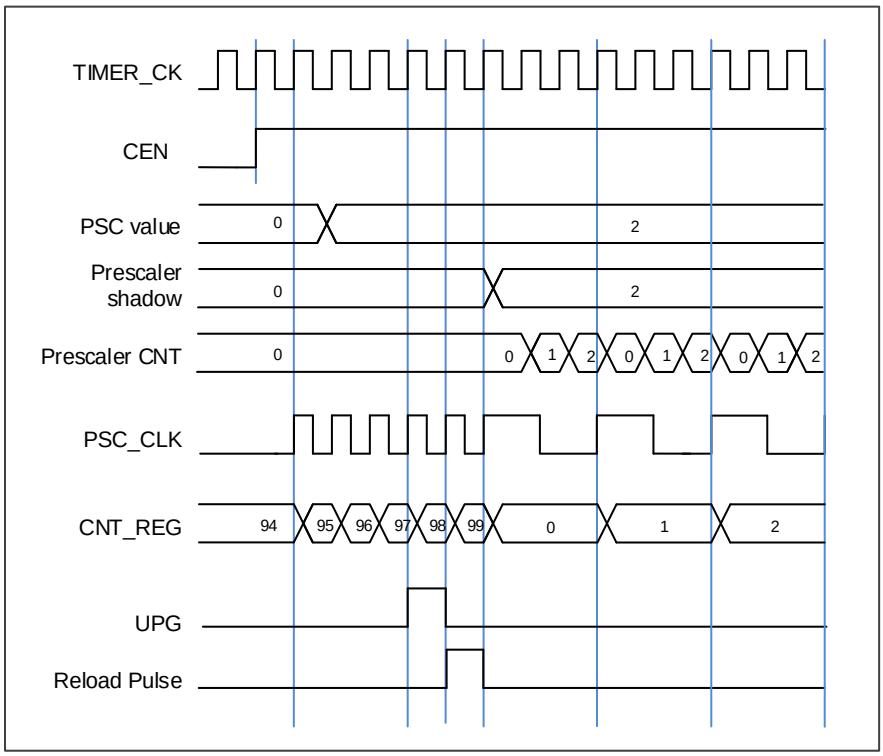
图 18-76. 内部时钟分频为 1 时，计数器的时序图



时钟预分频

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 18-77. 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。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图18-78. 向上计数时序图，PSC=0/2*和*图18-79. 向上计数时序图，在运行时改变TIMERx_CAR寄存器的值*给出了一些例子，当TIMERx_CAR=0x99时，计数器在不同预分频因子下的行为。

图 18-78. 向上计数时序图，PSC=0/2

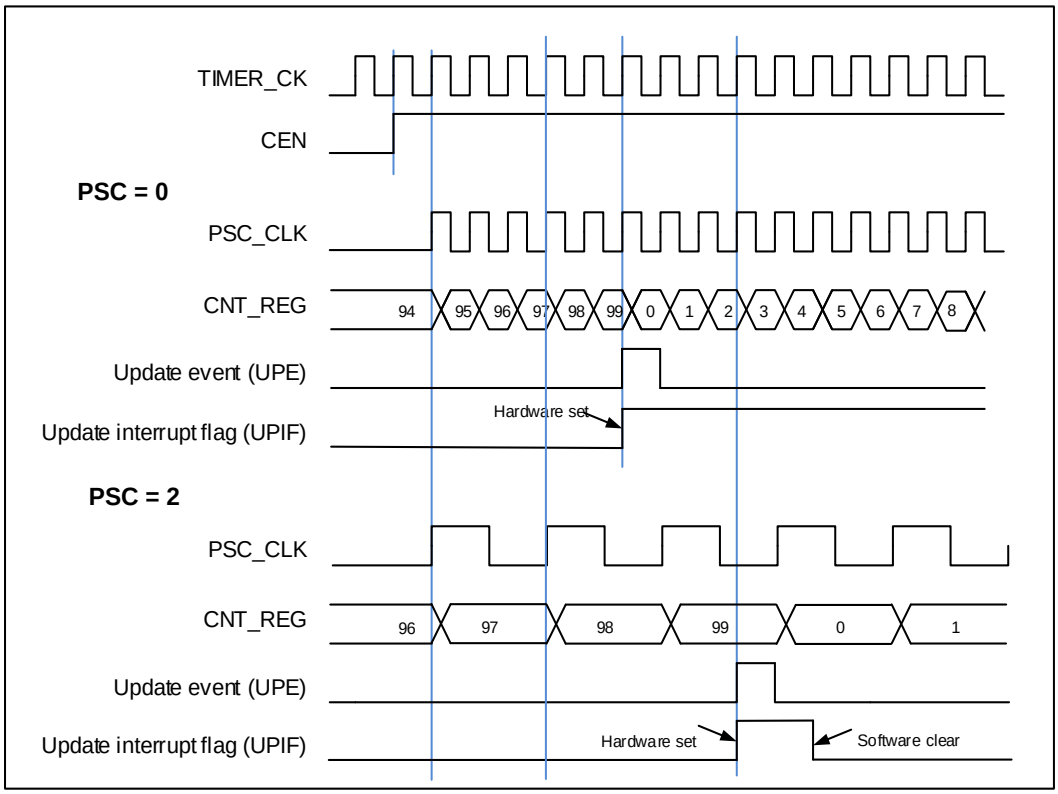
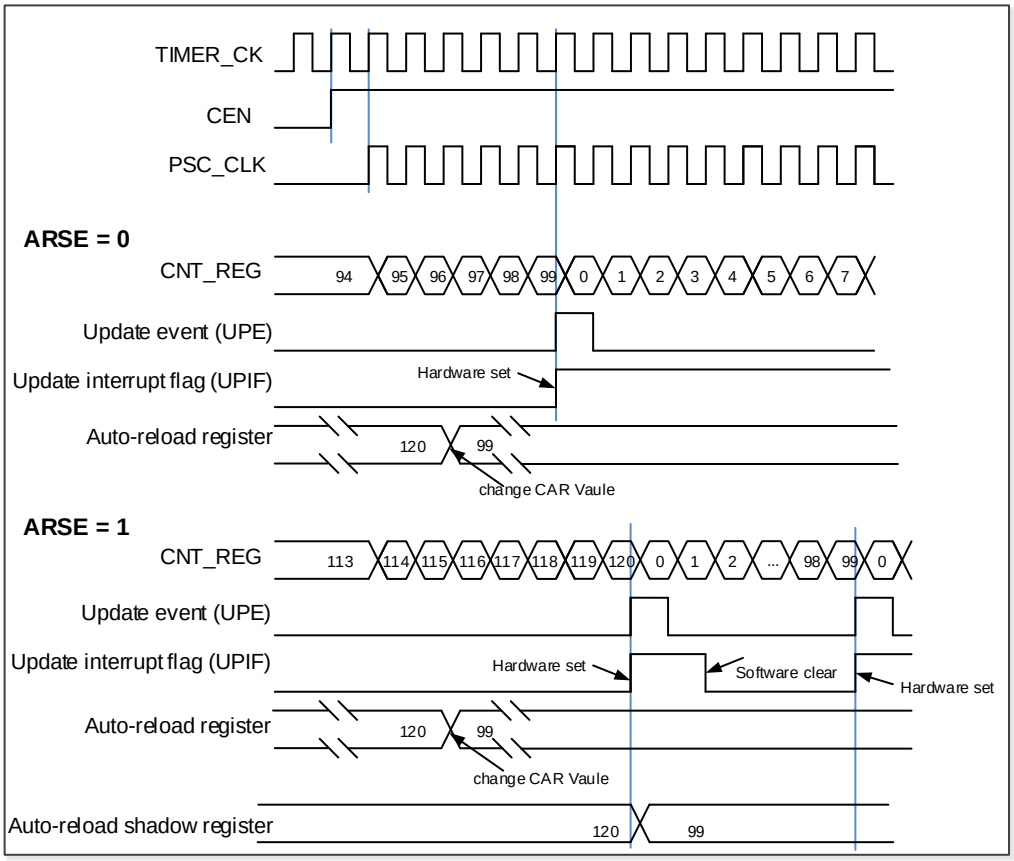


图 18-79. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



单脉冲模式

单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。

一旦设置定时器运行在单脉冲模式下，需要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

定时器调试模式

当Cortex®-M3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。

18.5.5. TIMERx 寄存器(x=5,6)

TIMER5基地址： 0x4000 1000

TIMER6基地址： 0x4000 1400

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								ARSE	保留			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器。 1：使能TIMERx_CAR寄存器的影子寄存器。
6:4	保留	必须保持复位值
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢

		复位模式产生的更新 1：更新事件禁能. 注意：当该位被置1时，UPG位被置1或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

保留

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

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

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

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

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入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。 注意：在定时器被配置为输入捕获模式时，该寄存器需要被配置成一个大于用户期望值的非 0 值(例如 0xFFFF)。

19. 通用同步异步收发器（USART）

19.1. 简介

通用同步异步收发器（USART）提供了一个灵活方便的串行数据交换接口，数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART提供了可编程的波特率发生器，能对UCLK进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和Modem流控操作（CTS/RTS）。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

所有USART都支持DMA功能，以实现高速率的数据通信。

19.2. 主要特征

- NRZ标准格式（Mark/Space）；
- 全双工异步通信；
- 可编程的波特率产生器：
  - 由外设时钟分频产生，其中USART0/5由PCLK2分频得到，USART1/2和UART3/4/6/7由PCLK1分频得到；
  - 16倍过采样；
  - 当时钟频率为120M，过采样为16，最高速度可到7.5MBits/s；
- 完全可编程的串口特性：
  - 偶校验位，奇校验位，无校验位的生成/检测；
  - 数据位（8或9位）；
  - 产生0.5、1、1.5或者2个停止位；
- 发送器和接收器可分别使能；
- 支持硬件Modem流控操作（CTS/RTS）；
- DMA访问数据缓冲区；
- LIN断开帧的产生和检测；
- 支持红外数据协议（IrDA）；
- 同步传输模式以及为同步传输输出发送时钟；
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式（T=0）；
  - 块模式（T=1）；
  - 直接和反向转换；
- 多处理器通信：
  - 如果地址不匹配，则进入静默模式；
  - 通过线路空闲检测或者地址匹配检测从静默模式唤醒；
- 多种状态标志：

- 传输检测标志：接收缓冲区不为空（RBNE），发送缓冲区为空（TBE），传输完成（TC），忙（BSY）；
- 错误检测标志：过载错误（ORERR），噪声错误（NERR），帧格式错误（FERR），奇偶校验错误（PERR）；
- 硬件流控操作标志：CTS变化（CTSF）；
- LIN模式标志：LIN断开检测（LBDF）；
- 多处理器通信模式标志：IDLE帧检测（IDLEF）；
- 智能卡模式标志：块结束（EBF）和接收超时（RTF）；
- 若相应的中断使能，这些事件发生将会触发中断；

USART0/1/2/5完全实现上述功能，但是UART3/4/6/7只实现了上面所介绍功能的部分，下面这些功能在UART3/4/6/7中没有实现：

- 智能卡模式；
- 同步模式；
- 硬件流操作（CTS/RTS）。

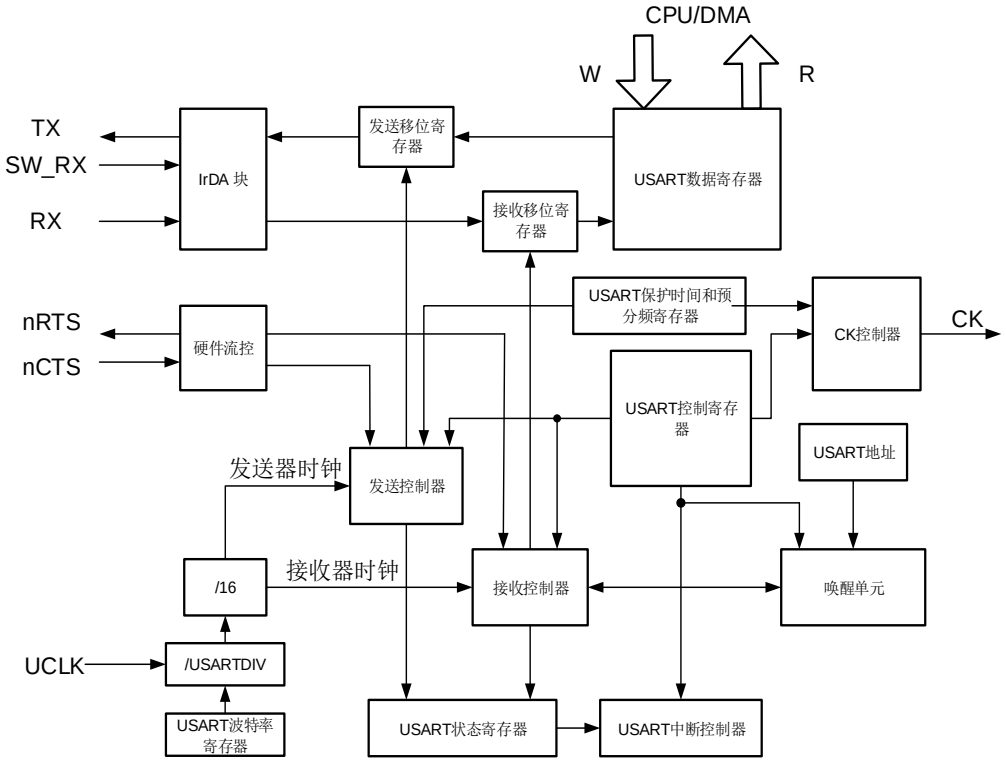
19.3. 功能说明

USART接口通过*表19-1. USART重要引脚描述*中主要引脚从外部连接到其他设备。

表 19-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O（单线模式/智能卡模式）	发送数据。当 USART 使能后，若无数据发送，默认为高电平
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

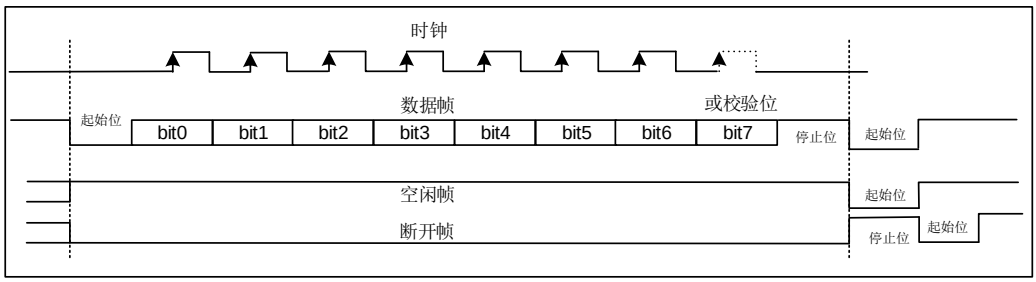
图 19-1. USART 模块内部框图



19.3.1. USART 帧格式

USART数据帧开始于起始位，结束于停止位。USART_CTL0寄存器中WL位可以设置数据长度。将USART_CTL0寄存器中PCEN置位，最后一个数据位可以用作奇偶校验位。若WL位为0，第七位为奇偶校验位。若WL位置1，第八位为奇偶校验位。USART_CTL0寄存器中PM位用于选择奇偶校验位的计算方法。

图 19-2. USART 字符帧（8 数据位和 1 停止位））



在发送和接收中，停止位可以由USART_CTL1寄存器中STB[1:0]位域配置。

表 19-2. 停止位配置

STB[1:0]	停止位长度（位）	功能描述
00	1	默认值
01	0.5	智能卡模式接收
10	2	标准 USART，单线以及调制解调模式

STB[1:0]	停止位长度（位）	功能描述
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断开帧。USART数据帧的传输速度由UCLK时钟频率，波特率发生器的配置共同决定。

19.3.2. 波特率发生

波特率分频系数是一个16位的数字，包含12位整数部分和4位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使USART能够产生所有标准波特率。

波特率分频系数（USARTDIV）与系统时钟具有如下关系：

$$USARTDIV=\frac{UCLK}{16\times Baud\ Rate} \quad (19\text{-}1)$$

例如，当过采样是16：

1. 由USART_BAUD寄存器的值得到USARTDIV：
   假设USART_BAUD=0x21D，则INTDIV=33（0x21），FRADIV=13（0xD）。
   UASRTDIV=33+13/16=33.81。

2. 由USARTDIV得到USART_BAUD寄存器的值：
   假设要求UASRTDIV=30.37，INTDIV=30（0x1E）。
   16*0.37=5.92，接近整数6，所以FRADIV=6（0x6）。
   USART_BAUD=0x1E6。

注意：若取整后FRADIV=16（溢出），则进位必须加到整数部分。

19.3.3. USART 发送器

如果USART_CTL0寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART_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。

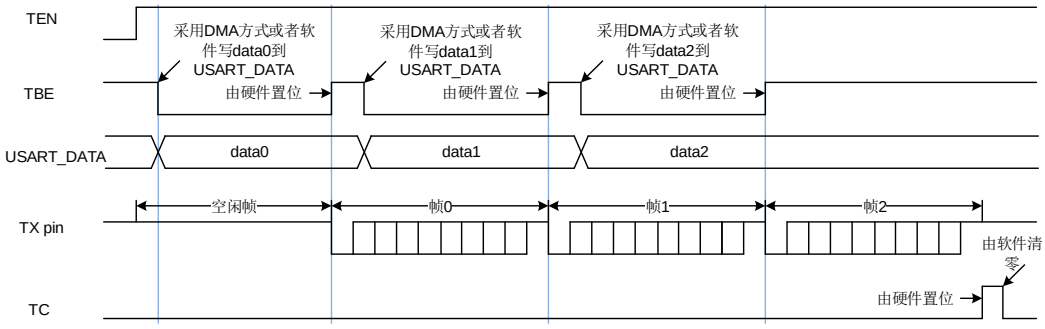
系统上电后，TBE默认为高电平。在USART_STAT0寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_DATA寄存器。当数据写入USART_DATA寄存器，TBE位将被清0。在数据由USART_DATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_DATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_DATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

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19-3. USART发送步骤*给出了USART发送步骤。软件操作按以下流程进行：

1. 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 19-3. USART 发送步骤



在禁用USART或进入低功耗状态之前，必须等待TC置位。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。

19.3.4. USART 接收器

上电后，USART接收器使能按以下步骤进行：

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。

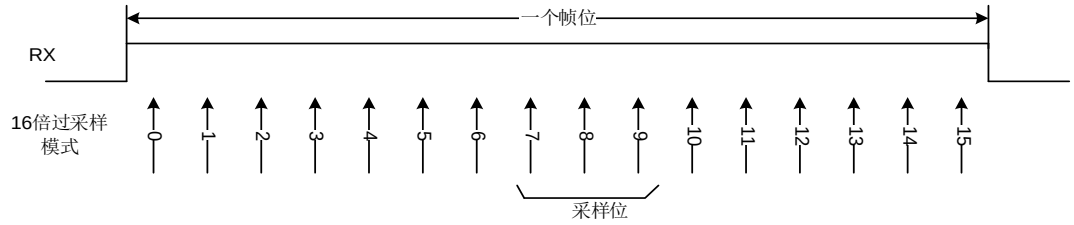
当接收到一个数据帧，USART_STAT0寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT0寄存器中可以观察接收状态标志。

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。在16倍过采样模式中，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能DMA，并置位USART_CTL2寄存器中ERRIE，将会产生中断。

图 19-4. 过采样方式接收一个数据位



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。

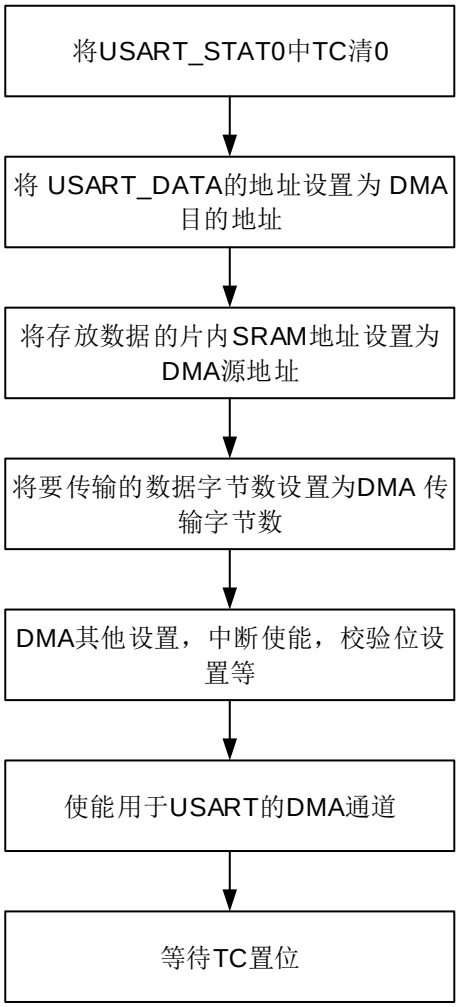
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

19.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART_CTL2寄存器中DENT位可以使能DMA发送，置位USART_CTL2寄存器中DENR位可以使能DMA接收。

当DMA用于USART发送时，DMA将数据从片内SRAM传送到USART的数据缓冲区。配置步骤如*图19-5. 采用DMA方式实现USART数据发送配置步骤*所示。

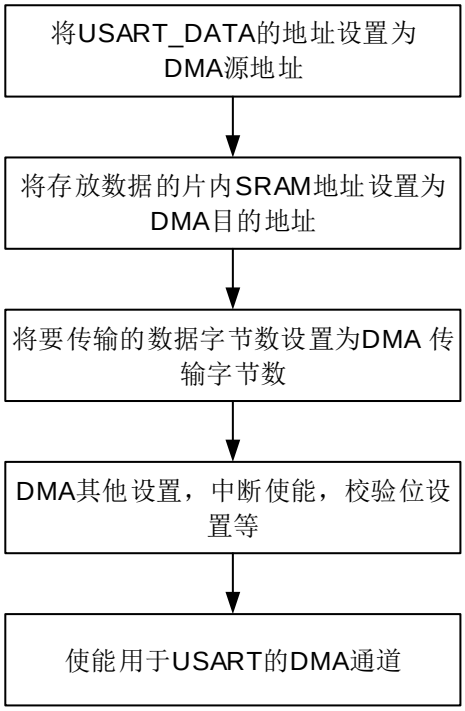
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_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19-6. 采用DMA方式实现USART数据接收配置步骤*所示。如果将USART_CTL2寄存器中ERRIE位置1，USART_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 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤

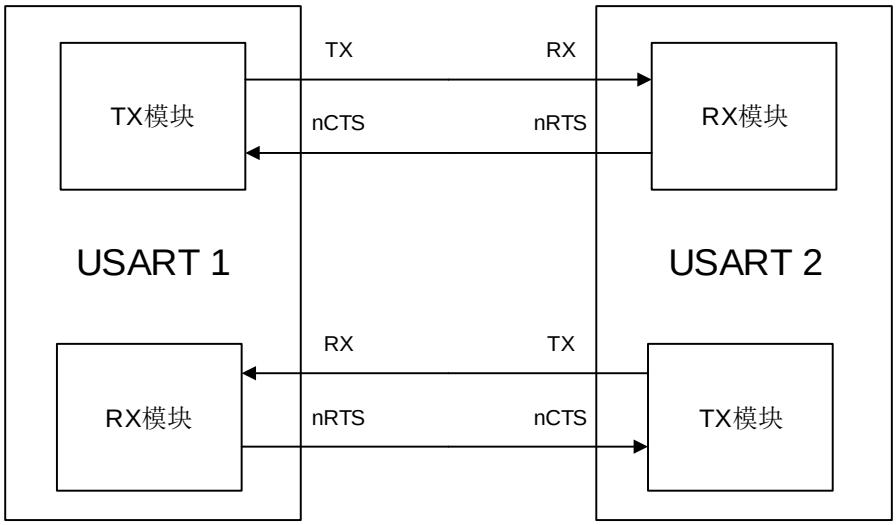


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

19.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 19-7. 两个 USART 之间的硬件流控制



RTS 流控

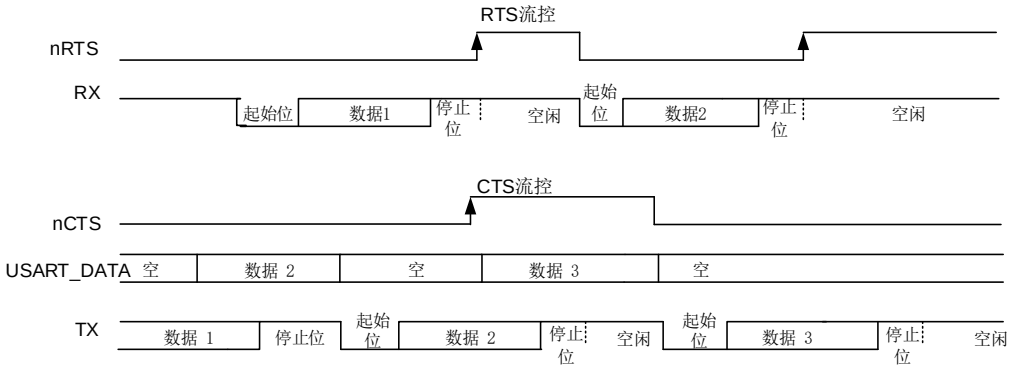
USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平，可

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 19-8. 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。

19.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART_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。

如果采用地址标记检测，默认情况下，接收器对地址字节不做奇偶校验。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。

19.3.8. LIN 模式

将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。

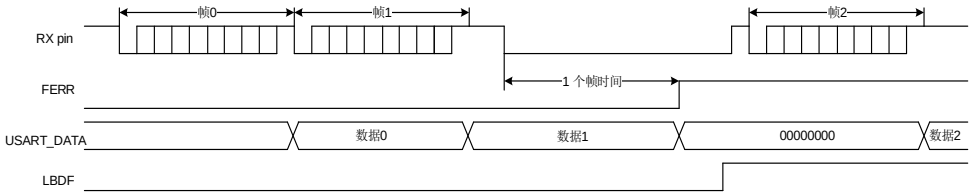
在LIN模式下，USART_CTL1寄存器中CKEN，STB[1:0]以及USART_CTL2的SCEN，HDEN，IREN位都应该被清0。

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。数据位的长度只能是8。当USART_CTL0寄存器中SBKCMD置位时，USART连续发送13个0，再发送一个停止位。

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。

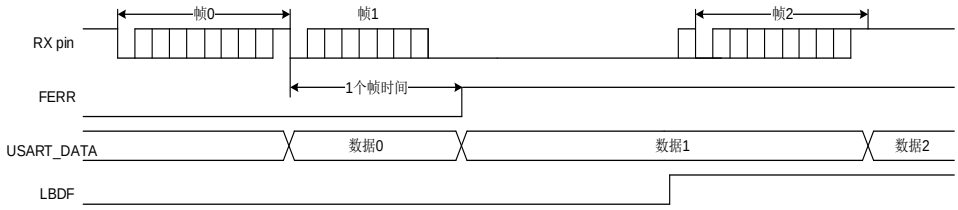
如*图19-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

图 19-9. 空闲状态下检测断开帧



如*图19-10. 数据传输过程中检测断开帧*所示，如果断开帧发生在数据传输过程中，当前传输帧发生错误，FERR置位。

图 19-10. 数据传输过程中检测断开帧



19.3.9. 同步通信模式

USART支持主机模式下的全双工同步串行通信，可以通过置位USART_CTL1的CKEN位来使能。在同步模式下，USART_CTL1的LMEN和USART_CTL2的SCEN，HDEN，IREN位应该被清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 19-11. 同步模式下的 USART 示例

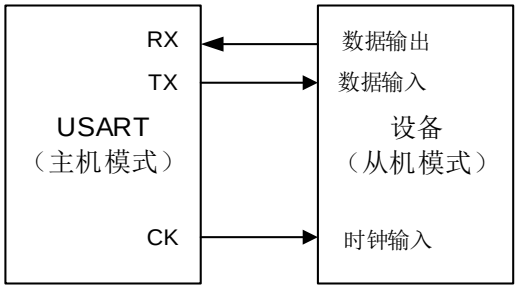
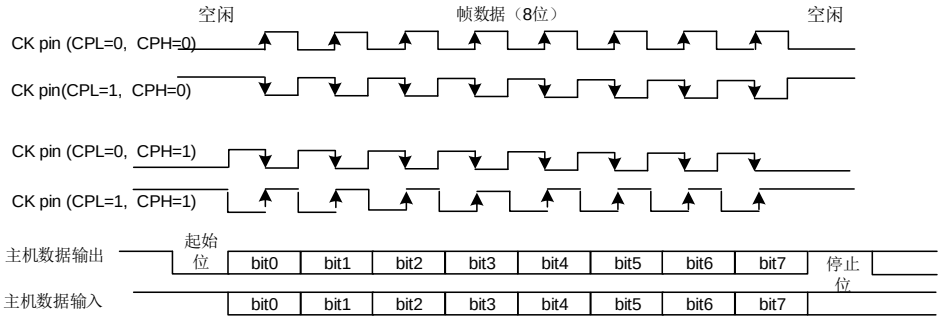


图 19-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

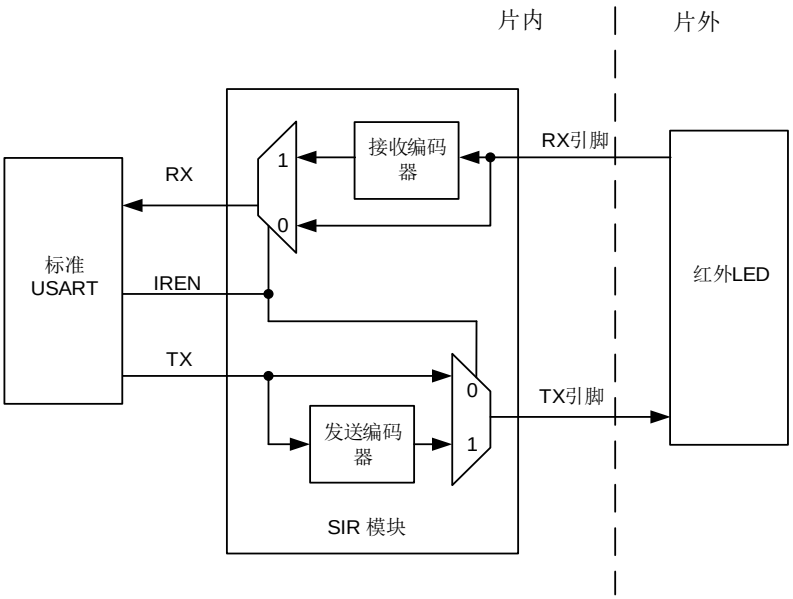


19.3.10. 串行红外（IrDA SIR）编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位将被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

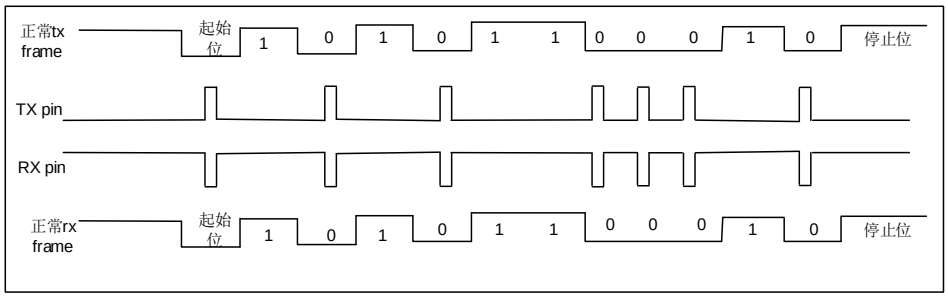
图 19-13. IrDA SIR ENDEC 模块



在IrDA模式下，TX引脚电平与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 19-14. IrDA 数据调制



将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收器解码器工作模式与正常IrDA模式相同。

19.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。

在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位清零。

半双工模式下，TX引脚和RX引脚将从内部连接到一起，RX引脚不再使用。TX引脚应该被配置为开漏输出模式。通信冲突由软件处理。

19.3.12. 智能卡（ISO7816）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 19-15. ISO7816-3 数据帧格式

字节模式（T=0）

相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2.5位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将要重新发送数据。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。

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。

19.3.13. USART 中断

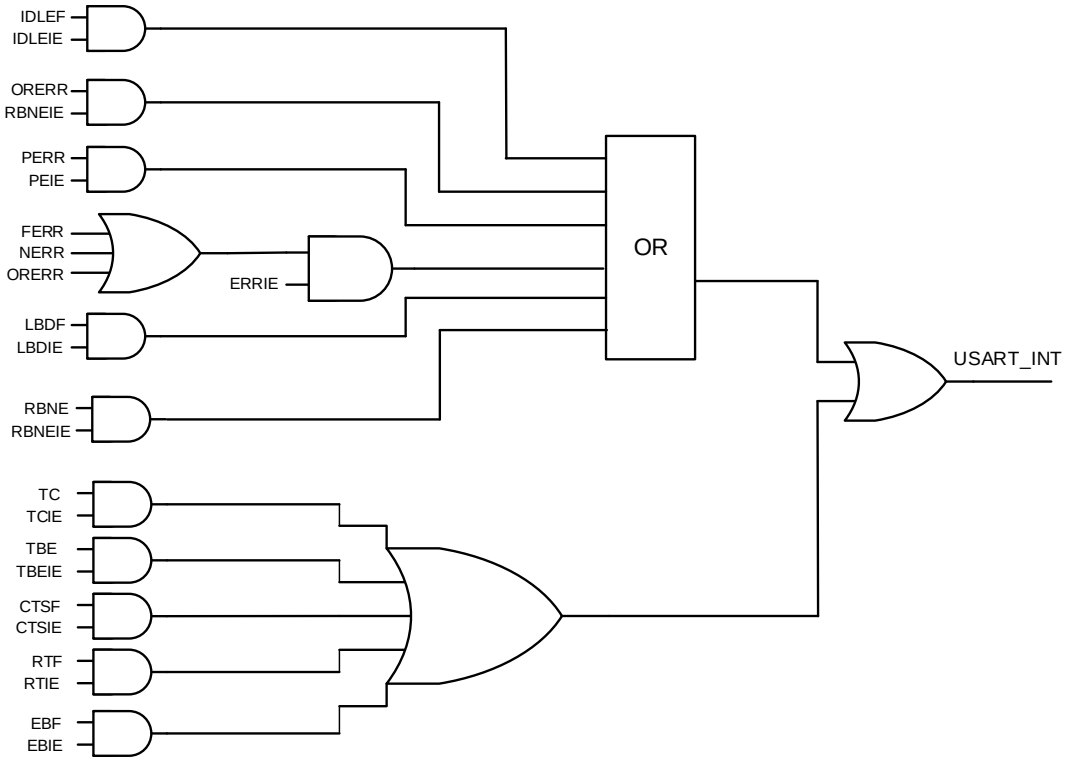
USART中断事件和标志如*表19-3. USART中断请求*所示：

表 19-3. USART 中断请求

中断事件	事件标志	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		
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
接收错误（噪声错误、溢出错误、帧错误）当DMA接收使能时	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 19-16. USART 中断映射框图



19.4. USART 寄存器

USART0 基地址：0x4001 3800

USART1 基地址：0x4000 4400

USART2 基地址：0x4000 4800

UART3 基地址：0x4000 4C00

UART4 基地址：0x4000 5000

USART5 基地址：0x4001 7000

UART6 基地址：0x4000 7800

UART7 基地址：0x4000 7C00

19.4.1. 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。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

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UART3/4/6/7无效。
8	LBDF	LIN断开检测标志 如果设置了USART_CTL1寄存器中LMEN位，当检测到LIN断开，该位由硬件置位。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 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USART_DATA可清除该位。 0：没检测到校验错误。 1：检测到校验错误。

19.4.2. 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。

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

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

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发送数据，或读这些位的值来获取接收数据。 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。

19.4.3. 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。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	INTDIV[11:0]	波特率分频器的整数部分。
3:0	FRADIV [3:0]	波特率分频器的小数部分。

19.4.4. 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
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

位/位域	名称	描述
31:14	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 当USART被使能（UEN=1）时，该位域不能改写。
11	WM	从静默模式唤醒方法 0：空闲线 1：地址掩码 当USART被使能（UEN=1）时，该位域不能改写。
10	PCEN	奇偶校验控制使能 0：奇偶校验控制禁用 1：奇偶校验控制被使能 当USART被使能（UEN=1）时，该位域不能改写。
9	PM	奇偶校验模式 0：偶校验 1：奇校验 当USART被使能（UEN=1）时，该位域不能改写。

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线检测中断禁用使能
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

19.4.5. 控制寄存器 1（USART_CTL1）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	保留	ADDR[3:0]			
	rw	rw		rw	rw	rw	rw		rw	rw		rw			

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	LMEN	LIN模式使能 0：LIN模式禁用 1：LIN模式使能 当USART被使能（UEN=1）时，该位域不能改写。
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 当USART被使能（UEN=1）时，该位域不能改写。 对于UART3/4/6/7，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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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平 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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

		当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。
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。

19.4.6. 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					CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
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UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7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 当USART被使能（UEN=1）时，该位域不能改写。
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 当USART被使能（UEN=1）时，该位域不能改写。
1	IREN	IrDA模式使能 0：IrDA禁用 1：IrDA使能 当USART被使能（UEN=1）时，该位域不能改写。
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。

19.4.7. 保护时间和预分频器寄存器（USART_GP）

地址偏移：0x18
复位值：0x0000 0000

当USART被使能（UEN=1）时，该位域不能改写。

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	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。 当USART被使能（UEN=1）时，该位域不能改写。 该位对于UART3/4/6/7无效。
7:0	PSC[7:0]	使能USART IrDA低功耗模式，这些位用来设定将外设时钟（PCLK1/PCLK2）分频产生低功耗频率的分频系数。 00000000：保留 – 不要写入该值 00000001：对源时钟1分频 00000010：对源时钟2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。

当USART被使能（UEN=1）时，该位域不能改写。

19.4.8. 控制寄存器 3（USART_CTL3）

地址偏移：0x80

复位值：0x0000 0000

UART3/4/6/7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，该位域不能改写。
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，该位域不能改写。
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，该位域不能改写。

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，该位域不能改写。
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：块中断使能 1：块中断禁用
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：接收超时中断使能 1：接收超时中断禁用
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被使能

19.4.9. 接收超时寄存器（USART_RT）

地址偏移：0x84

复位值：0x0000 0000

UART3/4/6/7未使用该寄存器。

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述

31:24	BL[7:0]	块长度 这些位用于设定智能卡T=1的接收时，块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC） – 1。 这个值可以在块接收开始去设置（用于需要从块的序言提取块的长度的情形），这个值在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。 在其他模式下，当REN=0 （禁用接收器）或者当USART_STAT1寄存器的EBF位被写0时，块的长度计数器被清0。
23:0	RT[23:0]	接收器超时阈值 该位域用于指定接收超时值，单位是波特时钟的时长。 标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，USART_STAT1寄存器中RTF标志被置位。 在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始算的。 这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。

19.4.10. 状态寄存器 1（USART_STAT1）

地址偏移：0x88

复位值：0x0000 0000

UART3/4/6/7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。

20. 内部集成电路总线接口（I2C）

20.1. 简介

I2C（内部集成电路总线）模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式和快速模式，具备 CRC 计算和校验功能、支持 SMBus（系统管理总线）和 PMBus（电源管理总线），此外还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

20.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口。
- 同一接口既可实现主机功能又可实现从机功能。
- 主从机之间的双向数据传输。
- 支持 7 位和 10 位的地址模式和广播寻址。
- 支持 I2C 多主机模式。
- 支持标速（最高 100 kHz）和快速（最高 400 kHz）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 2.0 和 PMBus。
- 两个中断：字节成功传输中断和错误事件中断。
- 可选择的 PEC（报文错误校验）生成和校验。

20.3. 功能描述

I2C 接口的内部结构如*图 20-1. I2C 模块框图*所示。

图 20-1. I2C 模块框图

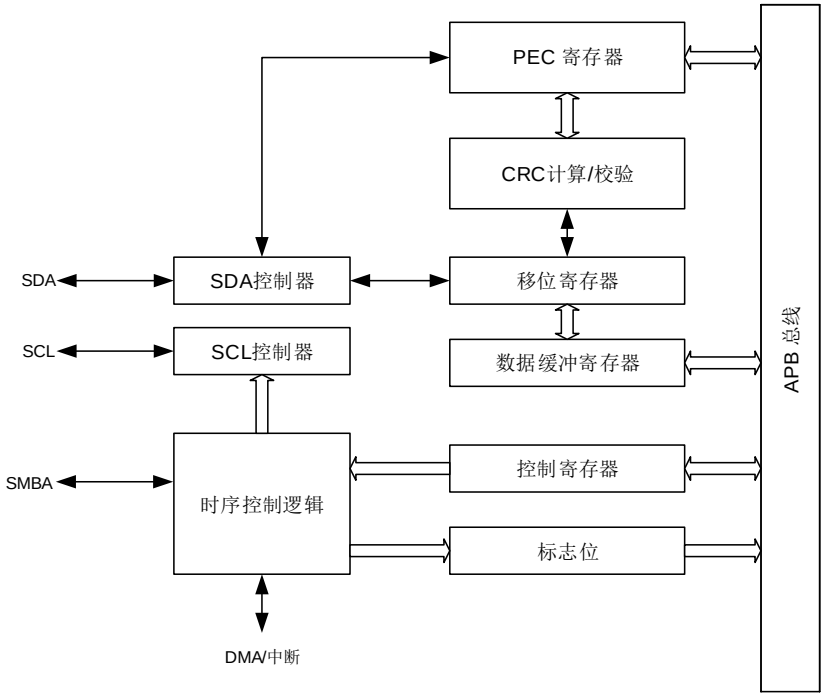


表 20-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	在不破坏信息的前提下同时控制总线的多个主机
同步	同步两个或更多设备之间的时钟信号的过程
仲裁	如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏，保证上述的过程叫仲裁

20.3.1. SDA 线和 SCL 线

I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。

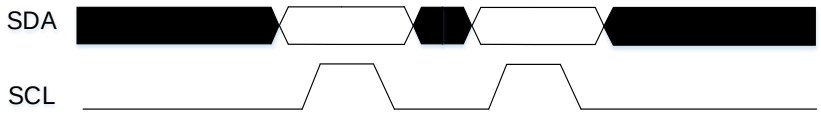
SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 Kbit/s，在快速模式下可以达到 400 Kbit/s。由于 I2C 总线上可能会连接不同工艺的设备（CMOS，NMOS，双极性器件），逻辑‘0'和逻辑‘1’的电平并不是

固定的，取决于 V_{DD} 的实际电平。

20.3.2. 数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线 SDA 的电平状态才能跳变（如*图 20-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲

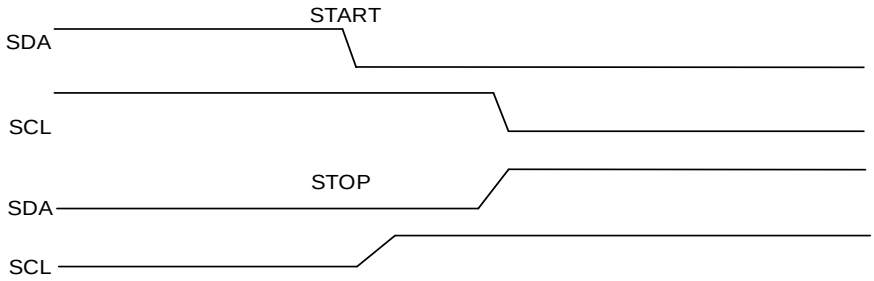
图 20-2. 数据有效性



20.3.3. 开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP（参见*图 20-3. 开始和停止信号*）。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 信号定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 20-3. 开始和停止信号

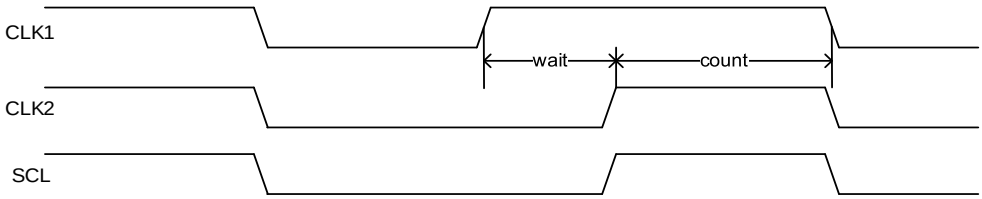


20.3.4. 时钟同步

两个主机可以同时在空闲总线上开始传送数据，因此必须通过一些机制来决定哪个主机获取总线的控制权并完成传输，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

时钟同步通过 SCL 线的线与来实现。这就是说 SCL 线的高到低切换会使器件开始计数它们的低电平周期，而且一旦主器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平(参见*图 20-4. 时钟同步*)。但是如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此 SCL 线被有最长低电平周期的器件保持低电平。此时低电平周期短的器件会进入高电平的等待状态。

图 20-4. 时钟同步



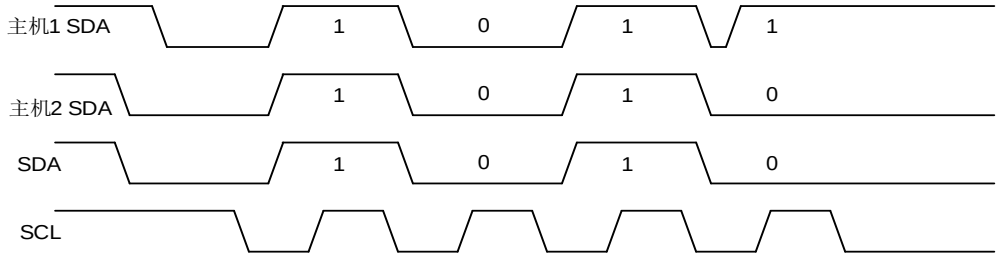
20.3.5. 仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。

只有在总线空闲的时候主机才可以启动传输。两个主机可能在START信号的最短保持时间内在总线上产生一个有效的START信号，这种情况下需要仲裁来决定由哪个主机来完成传输。

仲裁逐位进行，在每一位的仲裁期间，当SCL为高时，每个主机都检查SDA电平是否和自己发送的相同。仲裁的过程需要持续很多位。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到SDA电平为低，则认为自己仲裁失败并关闭自己的SDA输出驱动，而另一个主机则继续完成自己的传输。

图 20-5. SDA 线仲裁



20.3.6. I2C 通讯流程

每个I2C设备（不管是微控制器，LCD驱动，存储器或者键盘接口）都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C从机检测到I2C总线上的START信号之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，当两个地址相同时，I2C从机将发送一个确认应答（ACK），并响应总线的后续命令：发送或接收所需数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址（0x00）发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START信号和STOP信号来开始和结束一次传输，并且负责产生SCL时钟。

图 20-6. 7 位地址的 I2C 通讯流程

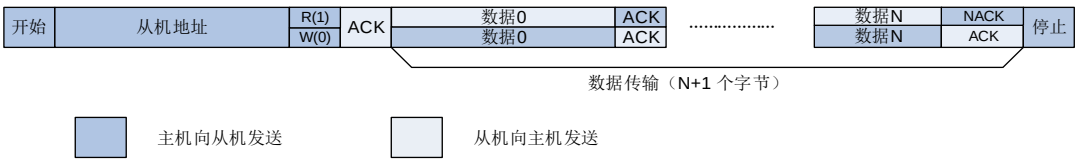


图 20-7. 10 位地址的 I2C 通讯流程（主机发送）

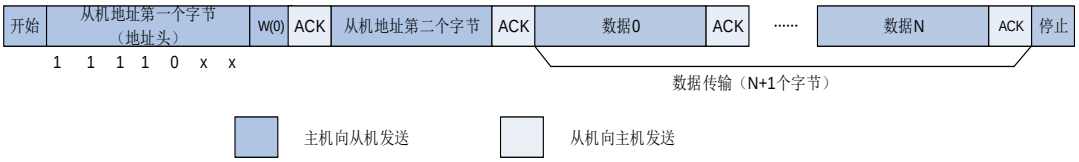
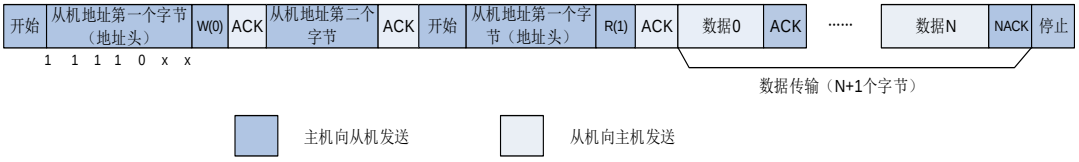


图 20-8. 10 位地址的 I2C 通讯流程（主机接收）



20.3.7. 软件编程模型

一个I2C设备例如LCD驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，I2C设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。

不管I2C设备是主机还是从机，都可以发送或接收数据，因此，I2C设备有以下4种运行模式：

- 主机发送方
- 主机接收方
- 从机发送方
- 从机接收方

I2C模块支持以上四种模式。系统复位以后，I2C默认工作在从机模式下。通过软件配置使I2C在总线上发送一个START信号之后，I2C变为主机模式，软件配置在I2C总线上发送STOP信号后，I2C又变回从机模式。

从机发送模式下的软件流程

如*首先，软件应该使能 I2C 外设时钟，以及配置 I2C_CTL1 中时钟相关寄存器来确保正确的 I2C 时序。使能和配置以后，I2C 运行在默认的从机模式状态，等待 I2C 总线上的 START 信号和地址。*

1. 当接收到一个START信号及随后的地址后，地址可以是7位格式也可以是10位格式，I2C硬件将I2C_STAT0寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START并发送一个地址头到I2C总线。从机在检测到START和紧接着的地址头之后会继续将ADDSEND位置1。软件可以通过读I2C_STAT0寄存器和接着读I2C_STAT1寄存器来第二次清除ADDSEND位。

2. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C_DATA都是空的，硬件将TBE位置1。软件此时可以写入第一个字节数据到I2C_DATA寄存器，但是TBE位并没有被清0，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。

3. 第一个字节的发送期间，软件可以写第二个字节到I2C_DATA，此时TBE位被清0，因为I2C_DATA寄存器和移位寄存器都不是空。

4. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C_DATA，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C_DATA寄存器。

5. 倒数第二个字节发送期间，软件写最后一个数据到I2C_DATA寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP信号时被清0。

6. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C从机的AERR（应答错误）会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

图20-9. 从机发送模式（10位地址模式）所示，在从机模式下要发送数据，软件应该按照以下步骤来运行操作：

7. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待I2C总线上的START信号和地址。

8. 当接收到一个START信号及随后的地址后，地址可以是7位格式也可以是10位格式，I2C硬件将I2C_STAT0寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START并发送一个地址头到I2C总线。从机在检测到START和紧接着的地址头之后会继续将ADDSEND位置1。软件可以通过读I2C_STAT0寄存器和接着读I2C_STAT1寄存器来第二次清除ADDSEND位。

9. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C_DATA都是空的，硬件将TBE位置1。软件此时可以写入第一个字节数据到I2C_DATA寄存器，但是TBE位并没有被清0，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。

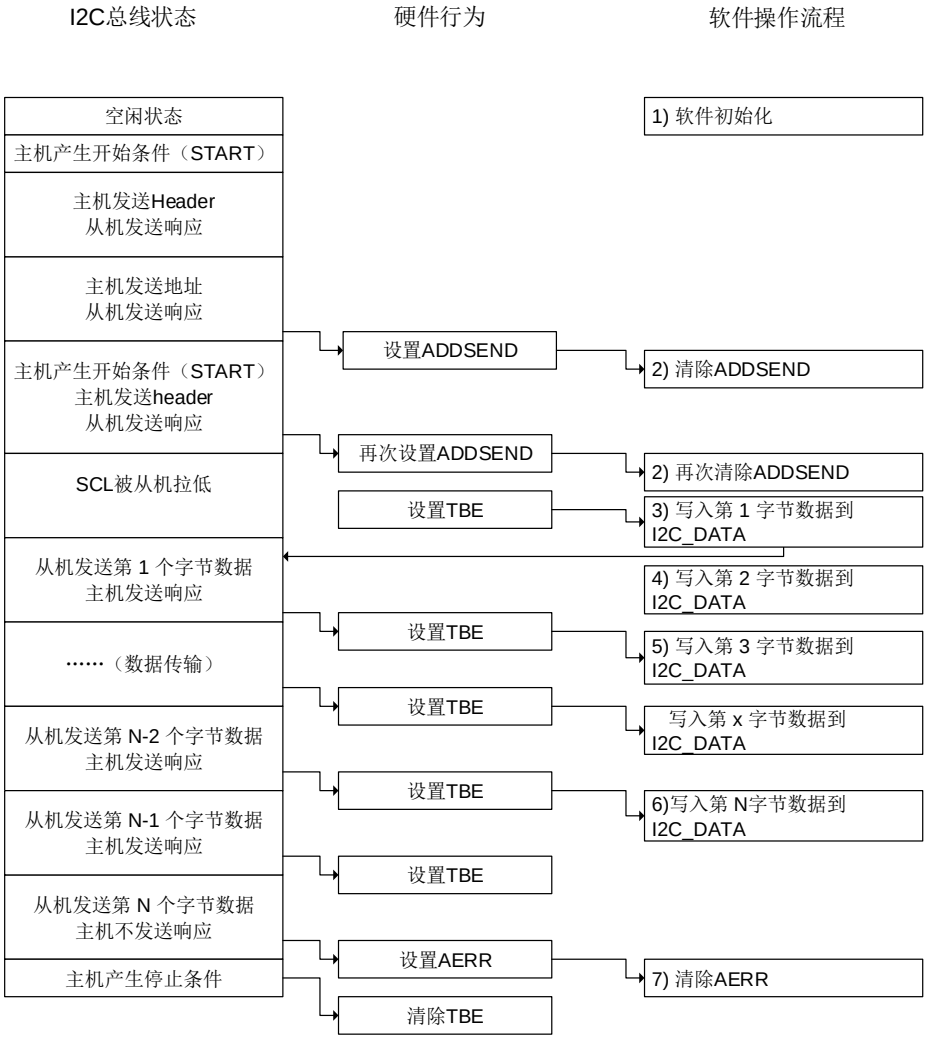
10. 第一个字节的发送期间，软件可以写第二个字节到I2C_DATA，此时TBE位被清0，因为I2C_DATA寄存器和移位寄存器都不是空。

11. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C_DATA，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C_DATA寄存器。

12. 倒数第二个字节发送期间，软件写最后一个数据到I2C_DATA寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP信号时被清0。

13. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C从机的AERR（应答错误）会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

图 20-9. 从机发送模式（10 位地址模式）



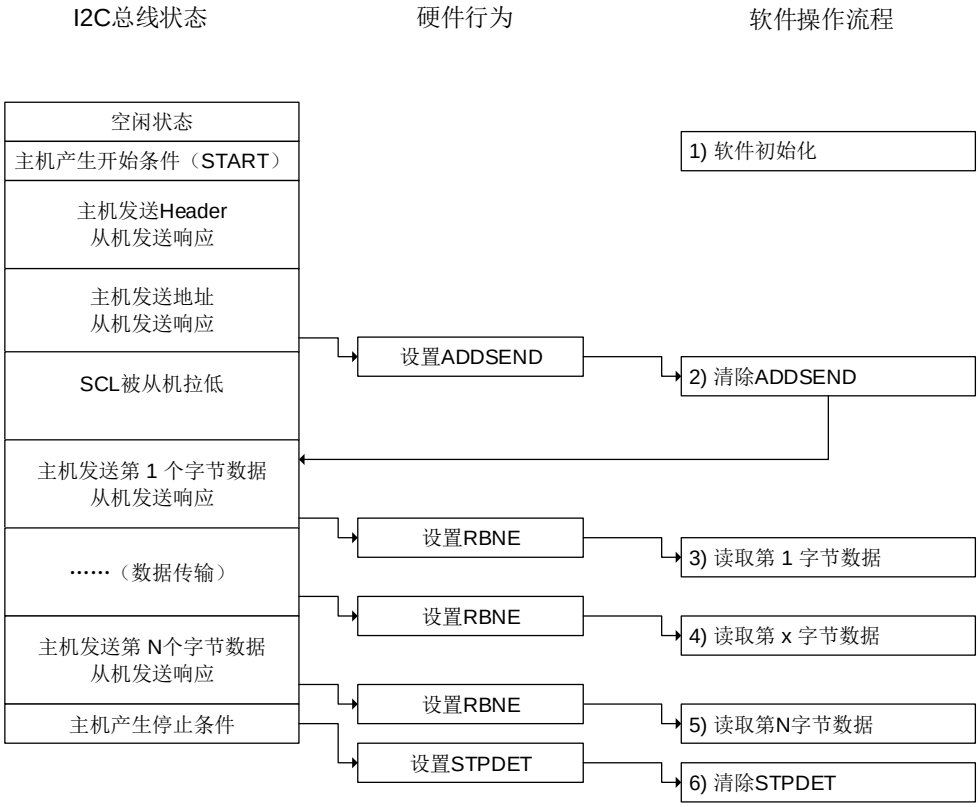
从机接收模式下的软件流程

如*图 20-10. 从机接收模式（10 位地址模式）*所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号以及地址。
2. 在接收到START起始信号和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器0的ADDSEND位置1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。当ADDSEND位被清0时，I2C就开始接收来自I2C总线的数据。
3. 当接收到第一个字节时，RBNE位被硬件置1，软件可以读取I2C_DATA寄存器的第一个字节，此时RBNE位也被清0。
4. 任何时候RBNE被置1，软件可以从I2C_DATA寄存器读取一个字节。
5. 接收到最后一个字节后，RBNE被置1，软件可以读取最后的字节。

6. 当I2C检测到I2C总线上一个STOP信号，STPDET位被置1，软件通过先读I2C_STAT0寄存器再写I2C_CTL0寄存器来清除STPDET位。

图 20-10. 从机接收模式（10 位地址模式）



主机发送模式下的软件流程

如*图20-11. 主机发送模式（10位地址模式）*所示，在主机模式下发送数据到I2C总线时，软件应该遵循这些步骤来运行I2C模块：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。

2. 软件将START位置1，在I2C总线上产生一个START信号。

3. 发送一个START信号后，I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。

5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C_DATA都是空的，所以硬件将

TBE位置1。此时软件可以写第一个字节数据到I2C_DATA寄存器，但是TBE位此时不会被清零，因为写入I2C_DATA寄存器的字节会被立即移入内部移位寄存器。当移位寄存器非空时，I2C就开始发送数据到总线。

6. 在第一个字节的发送过程中，软件可以写第二个字节到I2C_DATA，此时TBE会被清零，因为I2C_DATA寄存器和移位寄存器都不为空。

7. 任意时刻TBE被置1，软件都可以向I2C_DATA寄存器写入一个字节，只要还有数据待发送。

8. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到I2C_DATA来清除TBE标志位，此后就不用关心TBE位的状态。TBE位会在倒数第二个字节发送完成后被置起，直到发送STOP信号时被清零。

9. 最后一个字节发送结束后，I2C主机将BTC位置起，因为移位寄存器和I2C_DATA寄存器此时都为空。软件此时应该配置STOP来发送一个STOP信号，此后TBE和BTC状态位都将被清0。

图 20-11. 主机发送模式（10 位地址模式）

主机接收模式下的软件流程

在主机接收模式下，主机需要为最后一个字节接收产生NACK，然后发送STOP信号。因此，需

要特别注意以确保最后接收到数据的正确性。下面提供了两种针对主机接收模式的软件编程方案：方案A和B。方案A需要保证软件能对I2C事件进行快速响应，方案B则不需要。

方案 A

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该再次将START位置1来重新产生一个START。在START产生后，SBSEND位会被置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。
5. 当接收到第一个字节时，硬件会将RBNE位置1。此时软件可以从I2C_DATA寄存器读取第一个字节，之后RBNE位被清0。
6. 此后任何时候RBNE被置1，软件就可以从I2C_DATA寄存器读取一个字节。
7. 接收完倒数第二个字节（N-1）数据之后，软件应该立即将ACKEN位清0，并将STOP位置1，这一过程需要在最后一个字节接收完毕之前完成，以确保NACK发送给最后一个字节。
8. 最后一个字节接收完毕后，RBNE位被置1，软件可以读取最后一个字节。由于ACKEN已经在前一步骤中被清0，I2C不再为最后一个字节发送ACK，并在最后一个字节发送完毕后产生一个STOP信号。

以上步骤要求字节数目N>1，如果N=1，步骤7应该在步骤4之后就执行，且需要在字节接收完成之前完成。

图 20-12. 主机接收使用方案 A 模式（10 位地址模式）

方案 B

1. 首先，软件应该使能I2C外设时钟，配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。初始化完成之后，I2C运行在默认的从机模式状态，等待START信号和地址。

2. 软件将START位置1，从而在I2C总线上产生一个START信号。

3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头之后会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该接着将START位再次置1来产生一个开始信号，START被发送出去以后SBSEND位被再次置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。

5. 当第一个字节被接收时，RBNE位会被硬件置1。此时软件可从I2C_DATA寄存器读取出第一个字节，同时RBNE位被清0。

6. 此后任何时刻，只要RBNE位被置1，软件就可以从I2C_DATA寄存器读取一个字节的数据，直到主机接收了N-3个字节。

如*图20-13. 主机接收使用方案B模式（10位地址模式）*所示，第N-2个字节还没被软件读出，之后第N-1个字节被接收，此时BTC和RBNE都被置位，总线就会被主机锁死以阻止最后一个字节的接收。然后软件应该清除ACKEN位。

1. 软件从I2C_DATA读出倒数第三个（N-2）字节数据，同时也将BTC位清0。此后第N-1个字节从移位寄存器被移到I2C_DATA，总线得到释放然后开始接收最后一个字节，由于ACKEN已经被清除，因此主机不会给最后一个字节数据发送ACK响应。

2. 最后一个字节接收完毕后，硬件再次把BTC位和RBNE置1，并拉低SCL，软件将STOP位置1，主机发出一个STOP信号。

3. 软件读取第N-1个字节，清除BTC。此后最后一个字节从移位寄存器被移动到I2C_DATA。

4. 软件读取最后一个字节，清除RBNE。

以上步骤需要字节数字N>2，N=1和N=2的情况近似：

N=1

在第 4 步，软件应该在清除 ADDSEND 位之前将 ACKEN 位清 0，在清除 ADDSEND 位之后将 STOP 位置 1。当 N=1 时步骤 5 是最后一步。

N=2

在第2步，软件应该在START置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACKEN位清0。在第5步，软件应该一直等到BTC位被置1然后将STOP位置1且读取I2C_DATA两次。

图 20-13. 主机接收使用方案 B 模式（10 位地址模式）

20.3.8. SCL 线控制

SCL 线拉低功能是为了避免在接收时发生上溢错误以及在发送时发生下溢错误。如在软件编程模型中所示，在发送模式，当 TBE 和 BTC 被置位，发送器保持 SCL 线为低电平直到下一个发送数据写入传输缓冲区寄存器。在接收模式，当 RBNE 和 BTC 被置位，发送器保持 SCL 线为低电平直到传输缓冲区寄存器里的数据被读出。

当工作在从模式的时候，可以通过置位 I2C_CTL0 寄存器的 SS 位禁止 SCL 线拉低功能。如果该位置位，软件要能足够快的处理 TBE，RBNE 和 BTC 状态，否则上溢或下溢的情况可能会发生。

20.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 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 TBE 或 RBNE 位置 1 时，自动进行一次写或读操作，从而减轻了 CPU 的负荷，具体 DMA 的配置请参看 DMA 相关章节。

DMA 请求通过 I2C_CTL1 寄存器的 DMAON 位使能。该位应该在清除 ADDSEND 状态位之后被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSEND 事件前被置位。

参考 DMA 控制器的关于 DMA 的配置方法说明。DMA 必须在 I2C 传输开始之前配置和使能。当指定个数的字节已经传输完成，DMA 会发送一个传输结束（EOT）信号给 I2C 接口，并产生一个 DMA 传输完成中断。

当主机接收两个或两个以上字节时，需将 I2C_CTL1 寄存器的 DMALST 位置位。在接收到最后一个字节之后，I2C 主机发送 NACK。在 DMA 传输完成中断 ISR 中，通过置位 STOP 位，产生一个停止信号。

当主机仅接收一个字节时，清除ADDSEND状态前ACKEN位必须被清除。在清除ADDSEND状态后或在DMA传输完成中断ISR中，通过置位STOP位，产生一个停止信号。

20.3.10. 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，CRC多项式为x8 + x2 + x + 1，和SMBus协议兼容。将PECEN位置1就可以使能PEC功能。PEC将会计算所有通过I2C总线发送的数据（包括地址）。在非DMA模式下，软件可以通过配置PECTRANS来控制I2C在最后一个字节发送完毕后发送PEC值，或者在接收完成后检查接收到的PEC值是否正确。在DMA模式下，如果PECEN位和PECTRANS位被置1，I2C将自动发送或者检查PEC值。

20.3.11. SMBus 支持

系统管理总线（System Management Bus，简写为SMBus或SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus最常见于计算机主板，主要用于电源传输ON / OFF指令的通信。SMBus是I2C的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见Smart Battery Data）。

SMBus 协议

SMBus上每个报文交互都遵从SMBus协议中预定义的格式。SMBus是I2C规范中数据传输格式的子集。只要I2C设备可通过SMBus协议之一进行访问，便视为兼容SMBus规范。不符合这些协议的I2C设备，将无法被SMBus和ACPI规范所定义的标准方法访问。

地址解析协议

SMBus是基于I2C硬件实现的，它使用了I2C的硬件寻址方式，但在I2C的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是SMBus规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户接口。在此协议中，系统中的host与设备之间有一个重要的区别，即host具有分配地址的功能。

超时特性

SMBus有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为10kHz——为了防止长时间锁死总线。I2C在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续I2C会话。I2C总线协议中并没有限制这个延时的上限，但在SMBus系统中，这个时间被限定为25ms。按照SMBus协议的假定，如果某个会话耗时太久，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。

报文错误校验

SMBus 2.0以及1.1都采用了报文错误校验（Packet Error Checking，缩写为PEC）。在这种模式中，每次会话最后都将传输PEC字节。该字节是按照CRC-8校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为x8+x2+x+1（CRC-8-ATMHEC算法，初始化为0）。

SMBus 警报

SMBus还有一个额外的中断信号，称为SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus中还定义了较少见的“主机提醒协议”，基于I2C多主模式实现类似的提醒功能，但是可以传递更多数据。

SMBus 通讯流程

SMBus的通讯流程和标准I2C的流程相似。为了使用SMBus模式，在程序中需要配置几个SMBus特定的寄存器，响应一些SMBus特定标志位，实现那些在SMBus手册中介绍的上层协议。

1. 在通信之前，需要将I2C_CTL0中SMBEN位置1，并且根据需求，配置SMBSEL和ARPEN的值。
2. 为了支持ARP协议（ARPEN=1），在SMBus主机模式下（SMBSEL=1），软件需要响应标志位HSTSMB（在SMBus从机模式下，响应DEFSMB标志位），并实现ARP协议中的功能。
3. 为了支持SMBus警告模式，软件应该响应SMBALT标志位，并实现相应的功能。

20.3.12. 状态、错误和中断

I2C有一些状态、错误标志位，通过设置一些寄存器位，便可以从这些标志触发中断（详情参见*I2C寄存器*）。

表 20-2. 事件状态标志位

事件标志位名称	说明
SBSEND	主机发送 START 信号
ADDSEND	地址发送和接收
ADD10SEND	10 位地址模式中地址头发送
STPDET	监测到 STOP 信号
BTC	字节发送结束
TBE	发送时 I2C_DATA 为空
RBNE	接收时 I2C_DATA 非空

表 20-3. I2C 错误标志位

I2C 错误名称	说明
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	当禁用 SCL 拉低后，发生了溢出或下溢
AERR	没有接收到应答
PECERR	CRC 值不相同
SMBTO	SMBus 模式下总线超时
SMBALT	SMBus 警报

20.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

I2C2 基地址：0x4000 C000

20.4.1. 控制寄存器 0（I2C_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRESET	保留	SALT	PECTRANS	POAP	ACKEN	STOP	START	SS	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SRESET	软件复位 I2C，软件应该在 I2C 总线被释放时复位 I2C。 0：I2C 未复位 1：I2C 复位
14	保留	必须保持复位值。
13	SALT	软件置 1 和清 0，硬件清 0。 通过 SMBA 引脚发出警报。 0：不通过 SMBA 发布警告 1：通过 SMBA 引脚发送警告
12	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者监测到 START / STOP 信号，或 I2CEN=0。 0：不传输 PEC 值 1：传输 PEC 值
11	POAP	ACK / PEC 的位置含义 软件置 1 和清 0，当 I2CEN=0 时，硬件清 0。 0：ACKEN 位决定对当前正在接收的字节是否发送 ACK，PECTRANS 位表明正在接收的字节是否为 PEC 1：ACKEN 位决定是否对下一个字节发送 ACK，PECTRANS 位表明下一个即将被

		接收的字节是 PEC。
10	ACKEN	是否发送 ACK 软件置 1 和清 0，当 I2CEN=0 时硬件清 0。 0：不发送 ACK 1：发送 ACK
9	STOP	I2C 总线上产生一个 STOP 信号 软件置 1 和清 0，SMBus 超时时，硬件置 1，监测到 STOP 信号时，硬件清 0。 0：不发送 STOP 1：发送 STOP
8	START	I2C 总线上产生一个 START 信号 软件置 1 和清 0，当监测到 START 信号或 I2CEN=0 时由硬件清 0。 0：不发送 START 1：发送 START
7	SS	在从机模式下数据未就绪时是否将 SCL 拉低 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL
6	GCEN	是否响应对地址（0x00）的广播呼叫 0：从机不响应广播呼叫 1：从机将响应广播呼叫
5	PECEN	PEC 计算使能 0：PEC 计算禁能 1：PEC 计算使能
4	ARPEN	SMBus 下 ARP 协议使能 0：ARP 禁能 1：ARP 使能
3	SMBSEL	SMBus 类型选择 0：从机 1：主机
2	保留	必须保持复位值。
1	SMBEN	SMBus / I2C 模式开关 0：I2C 模式 1：SMBus 模式
0	I2CEN	I2C 外设使能 0：禁能 I2C 1：使能 I2C

20.4.2. 控制寄存器 1（I2C_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			DMALST	DMAON	BUFIE	EVIE	ERRIE	保留		I2CCLK[5:0]					
			rw	rw	rw	rw	rw			rw					

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	DMALST	DMA 最后传输配置 0：下一个 DMA EOT 不是最后传输 1：下一个 DMA EOT 是最后传输
11	DMAON	DMA 模式开关 0：DMA 模式关 1：DMA 模式开
10	BUFIE	缓冲区中断使能 0：禁用缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时不产生中断。 1：使能缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时产生中断。
9	EVIE	事件中断使能 0：禁用事件中断 1：使能事件中断，意味着当 SBSEND、ADDSEND、ADD10SEND、STPDET 或 BTC 标志位有效或当 BUFIE = 1 时 TBE = 1 或 RBNE = 1 时产生中断。
8	ERRIE	错误中断使能 0：禁用错误中断 1：使能错误中断，意味着当 BERR、LOSTARB、AERR、OUERR、PECERR、SMBTO 或 SMBALT 标志位生效时产生中断。
7:6	保留	必须保持复位值。
5:0	I2CCLK[5:0]	I2C 外设时钟频率 I2CCLK[5:0]应该是输入 APB1 时钟频率，最低 2MHz。 000000 - 000001：无时钟 000010 - 111100：2 MHz~60MHz 111101 - 111111：由于 APB1 时钟限制，无时钟 **注意：** 在标准模式下，APB1 时钟频率需大于或者等于 2MHz。在快速模式下，APB1 时钟

频率需大于或者等于 8MHz。

20.4.3. 从机地址寄存器 0（I2C_SADDR0）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按半字（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
ADDFORMAT	保留					ADDRESS[9:8]		ADDRESS[7:1]							ADDRESS0
rw						rw		rw							rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDFORMAT	I2C 从机地址模式 0：7 位地址 1：10 位地址
14:10	保留	必须保持复位值。
9:8	ADDRESS[9:8]	10 位地址的最高两位
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位
0	ADDRESS0	10 位地址的第 0 位

20.4.4. 从机地址寄存器 1（I2C_SADDR1）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								ADDRESS2[7:1]							DUADEN
											rw				rw

位/位域	名称	描述
31:8	保留	必须保持复位值。

7:1	ADDRESS2[7:1]	从机在双重地址模式下第二个 I2C 地址
0	DUADEN	双重地址模式使能 0：禁用双重地址模式 1：使能双重地址模式

20.4.5. 传输缓冲区寄存器（I2C_DATA）

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TRB[7:0]	数据发送接收缓冲区

20.4.6. 传输状态寄存器 0（I2C_STAT0）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALT	SMBTO	保留	PECERR	OUERR	AERR	LOSTARB	BERR	TBE	RBNE	保留	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SMBALT	SMBus 警报状态 硬件置 1，软件写 0 清 0。 0：SMBA 引脚未被拉低（从机模式）或未监测到警报（主机模式） 1：SMBA 引脚被拉低且接收到警报地址（从机模式）或监测到警报（主机模式）

14	SMBTO	SMBus 模式下超时信号 硬件置 1，软件写 0 清 0。 0：无超时错误 1：超时事件发生（SCL 被拉低达 25ms）
13	保留	必须保持复位值。
12	PECERR	接收数据时 PEC 错误 硬件置 1，软件写 0 清 0。 0：接收到 PEC 且校验正确 1：接收到 PEC 但检验错误，此时不管 ACKEN 位的值，I2C 将发送 NACK
11	OUERR	当禁用 SCL 拉低功能后，在从机模式下发生了上溢或下溢事件。在从机接收模式下，假如 I2C_DATA 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生上溢错误。在从机发送模式下，假如当前字节已经发送完成，而 I2C_DATA 仍然为空，就会发生下溢错误。 硬件置 1，软件写 0 清 0。 0：无上溢或下溢错误发生 1：发生上溢或下溢错误
10	AERR	应答错误 硬件置 1，软件写 0 清 0。 0：未发生应答错误 1：发生了应答错误
9	LOSTARB	主机模式下仲裁丢失 硬件置 1，软件写 0 清 0。 0：无仲裁丢失 1：发生仲裁丢失，I2C 模块返回从机模式。
8	BERR	总线错误，表示 I2C 总线上发生了预料之外的 START 信号或 STOP 信号。 硬件置 1，软件写 0 清 0。 0：无总线错误 1：发生了总线错误
7	TBE	发送期间 I2C_DATA 为空 硬件从 I2C_DATA 寄存器移动一个字节到移位寄存器之后将此位置 1，软件写一个字节到 I2C_DATA 寄存器清除该位。如果移位寄存器和 I2C_DATA 寄存器都是空的，写 I2C_DATA 寄存器将不会清除 TBE 位（详见主机/从机发送模式下的软件操作流程） 0：I2C_DATA 非空 1：I2C_DATA 空，软件可以写
6	RBNE	接收期间 I2C_DATA 非空 硬件从移位寄存器移动一个字节到 I2C_DATA 寄存器之后将此位置 1，读 I2C_DATA 可以清除此位。如果 BTC 和 RBNE 都被置 1，读 I2C_DATA 将不会清除 RBNE，因为移位寄存器的字节将被立即移到 I2C_DATA。 0：I2C_DATA 为空

		1：I2C_DATA 非空，软件可以读
5	保留	必须保持复位值。
4	STPDET	从机模式下监测到 STOP 信号 此位被硬件置 1，先读 I2C_STAT0 然后写 I2C_CTL0 可以清除此位。 0：从机模式下未监测到 STOP 信号 1：从机模式下监测到 STOP 信号
3	ADD10SEND	主机模式下 10 位地址的地址头被发送 该位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清除此位。 0：主机模式下未发送 10 位地址的地址头 1：主机模式下发送 10 位地址的地址头
2	BTC	字节发送结束 接收模式下，如果一个字节已经被移位寄存器接收但是此时 I2C_DATA 寄存器仍然是满的；或者发送模式下，一个字节已经被移位寄存器发送但是 I2C_DATA 寄存器仍然是空的，如果使能了 SCL 拉低功能，位硬件就会置起 BTC 标志。 此位由硬件置 1。 可由以下三种方式清除： 1、软件清除：读 I2C_STAT0，然后读或写 I2C_DATA 寄存器清除此位 2、硬件清除：发送一个 STOP 或 START 信号 3、寄存器 I2C_CTL0 中 I2CEN=0 0：未发生 BTC 1：发生了 BTC
1	ADDSEND	主机模式下：成功发送了地址并收到 ACK 从机模式下：接收到的地址与自身的地址匹配 此位由硬件置 1，软件读 I2C_STAT0 寄存器和读 I2C_STAT1 清 0。 0：从机模式下，未收到地址或者收到的地址不匹配；主机模式下，无地址被发送或地址已发送但未收到从机回复的 ACK 1：从机模式下，接收到的地址与自身的地址匹配；主机模式下，地址已发送并收到 ACK
0	SBSEND	主机模式下发送 START 信号 此位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清 0。 0：未发送 START 条件 1：START 条件被发送

20.4.7. 传输状态寄存器 1（I2C_STAT1）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PECV[7:0]								DUMODF	HSTSMB	DEFSMB	RXGC	保留	TR	I2CBSY	MASTER
r								r	r	r	r		r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	PECV[7:0]	当 PEC 使能后硬件计算出的 PEC 值。
7	DUMODF	从机模式下双标志位表明哪个地址和双地址模式匹配 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：地址和 I2C_SADDR0 匹配 1：地址和 I2C_SADDR1 匹配
6	HSTSMB	从机模式下监测到 SMBus 主机地址头 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未监测到 SMBus 主机地址头 1：监测到 SMBus 主机地址头
5	DEFSMB	SMBus 设备缺省地址 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：SMBus 设备没有接收到缺省地址 1：SMBus 设备接收到缺省地址
4	RXGC	是否接收到广播地址（00h） STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未接收到广播呼叫地址 1：接收到广播呼叫地址
3	保留	必须保持复位值。
2	TR	I2C 作发送端还是接收端 STOP 或 START 信号产生后或 I2CEN 或 LOSTARB=1 时此位由硬件清 0。 0：接收端 1：发送端
1	I2CBSY	忙标志 STOP 信号后硬件清 0。 0：无 I2C 通讯 1：I2C 正在通讯
0	MASTER	主机模式 表明 I2C 时钟在主机模式还是从机模式的标志位。 该位在 START 信号产生后由硬件置 1。 该位在 STOP 信号产生后或 I2CEN=0 或 LOSTARB=1 时此位由硬件清 0。 0：从机模式 1：主机模式

20.4.8. 时钟配置寄存器（I2C_CKCFG）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FAST	主机模式下 I2C 速度选择 0：标准速度 1：快速
14	DTCY	快速模式下占空比 0：$T_{low}/T_{high}=2$ 1：$T_{low}/T_{high}=16/9$
13:12	保留	必须保持复位值。
11:0	CLKC[11:0]	主机模式下 I2C 时钟控制 标准速度模式下：$T_{high}=T_{low}=CLKC*T_{PCLK1}$ 如果 DTCY=0，快速模式下： $T_{high}=CLKC*T_{PCLK1}$，$T_{low}=2*CLKC*T_{PCLK1}$ 如果 DTCY=1，快速模式下： $T_{high}=9*CLKC*T_{PCLK1}$，$T_{low}=16*CLKC*T_{PCLK1}$ **注意：**如果 DTCY=0，当 PCLK1 为 3 的整数倍时，波特率会比较准确。如果 DTCY=1，当 PCLK1 为 25 的整数倍时，波特率会比较准确。

20.4.9. 上升时间寄存器（I2C_RT）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

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

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

rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	RISETIME[5:0]	主机模式下最大上升时间 RISETIME 值应该为 SCL 最大上升时间加 1

21. 串行外设接口/片上音频接口（SPI/I2S）

21.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。SPI0还支持SPI四线主机模式。

片上音频接口（Inter-IC Sound，缩写为I2S）支持四种音频标准，分别是I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

21.2. 主要特征

21.2.1. SPI 主要特征

- 具有全双工、半双工和单工模式的主从操作。
- 16位宽度，独立的发送和接收缓冲区。
- 8位或16位数据帧格式。
- 低位在前或高位在前的数据位顺序。
- 软件和硬件NSS管理。
- 硬件CRC计算、发送和校验。
- 发送和接收支持DMA模式。
- 支持SPI四线功能的主机模式（只有SPI0）。

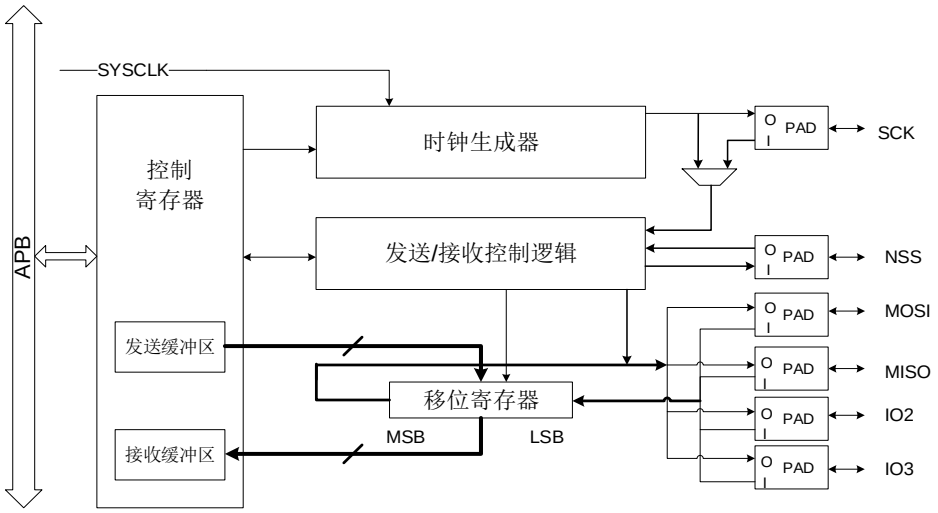
21.2.2. I2S 主要特征

- 具有发送和接收功能的主从操作。
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。
- 数据长度可以为16位，24位和32位。
- 通道长度为16位或32位。
- 16位缓冲区用于发送和接收。
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率。
- 可编程空闲状态时钟极性。
- 可以输出主时钟（MCK）。
- 发送和接收支持DMA功能。

21.3. SPI 功能说明

21.3.1. SPI 结构框图

图 21-1. SPI 结构框图



21.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 21-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 从机双向性模式：数据发送和接收线
MOSI	I/O	主机：数据发送线 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。 从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号

SPI 四线配置

SPI默认配置为单路模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于SPI0）。SPI四线模式只能工作在主机模式。

通过配置SPI_QCTL中的IO23_DRV位，在常规非四线SPI模式下，软件可以驱动IO2引脚和IO3引脚为高电平。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

表 21-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

21.3.3. SPI 时序和数据帧格式

SPI_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。

图 21-2. 常规模式下的 SPI 时序图

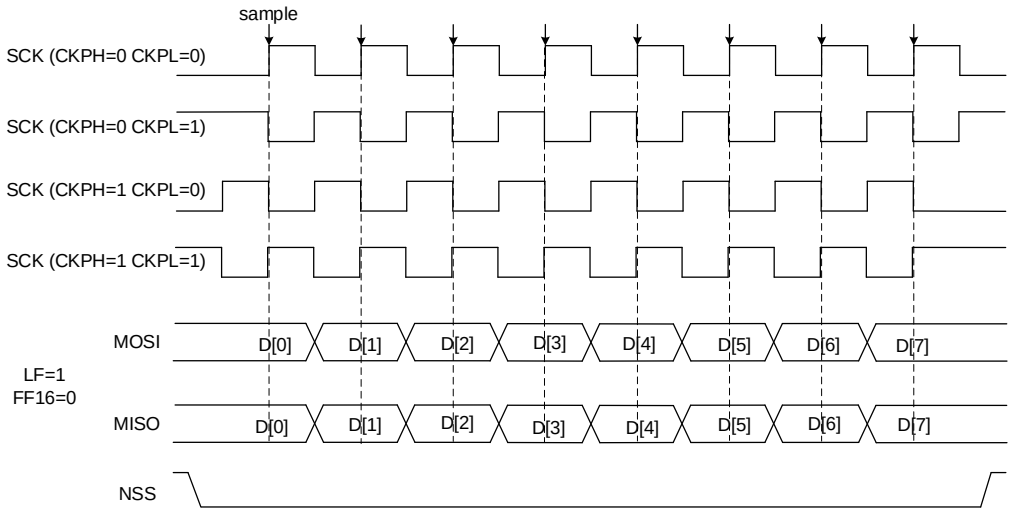
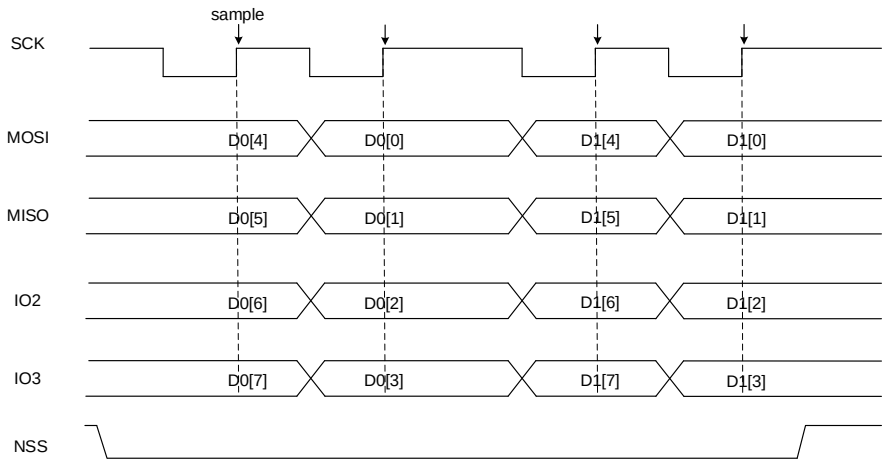


图 21-3. SPI 四线模式下的 SPI 时序图（CKPL=1, CKPH=1, LF=0）



在常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。在SPI四线模式下，数据帧长度固定为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。

21.3.4. NSS 功能

从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

表 21-3. 从机模式 NSS 功能

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0：NSS 电平为低 SWNSS = 1：NSS 电平为高

主机模式

在主机模式（MSTMOD=1）下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式（SWNSSEN=0, NSSDRV=0）或者软件模式（SWNSSEN=1）。一旦NSS引脚（在硬件NSS模式下）或SWNSS位（在软件NSS模式下）被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用NSS引脚控制SPI从设备，NSS应该配置为硬件输出模式

（SWNSSEN=0, NSSDRV=1）。使能SPI之后，NSS保持高电平，当发送或接收过程开始时，NSS变为低电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

表 21-4. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV：不要求	适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

21.3.5. SPI 运行模式

表 21-5. SPI 运行模式

模式	描述	寄存器配置	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 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用

模式	描述	寄存器配置	使用的数据引脚
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 21-4. 典型的全双工模式连接

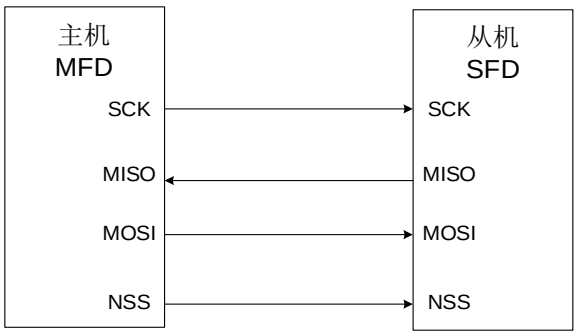


图 21-5. 典型的单工模式连接（主机：接收，从机：发送）

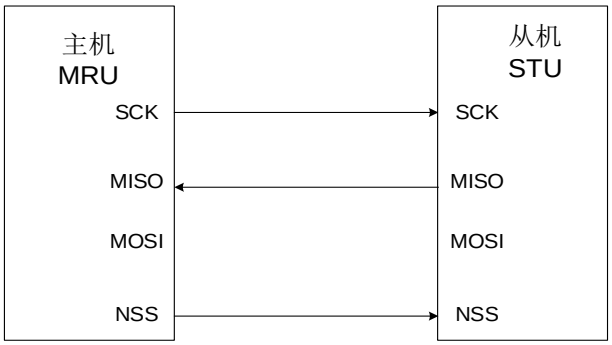


图 21-6. 典型的单工模式连接（主机：只发送，从机：接收）

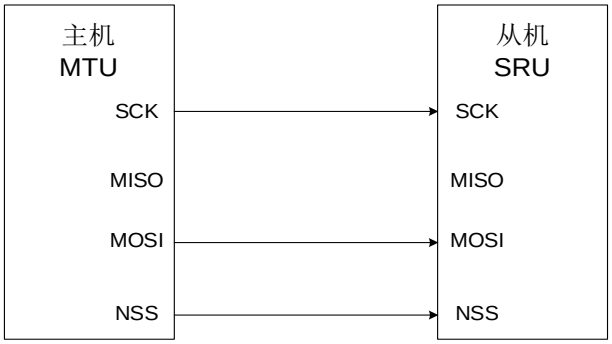
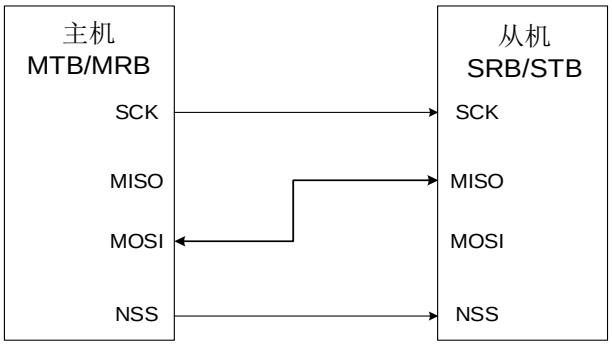


图 21-7. 典型的双向线连接



SPI 初始化流程

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号。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. 根据***SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位；
7. 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；

8. 使能SPI（将SPIEN位置1）。

注意：在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF位。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区空）位置1。TBE标志位置1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且RBNE（接收缓冲区非空）位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，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）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU，MTB，STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式（SRU或SRB）与全双工模式类似。

SPI 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。

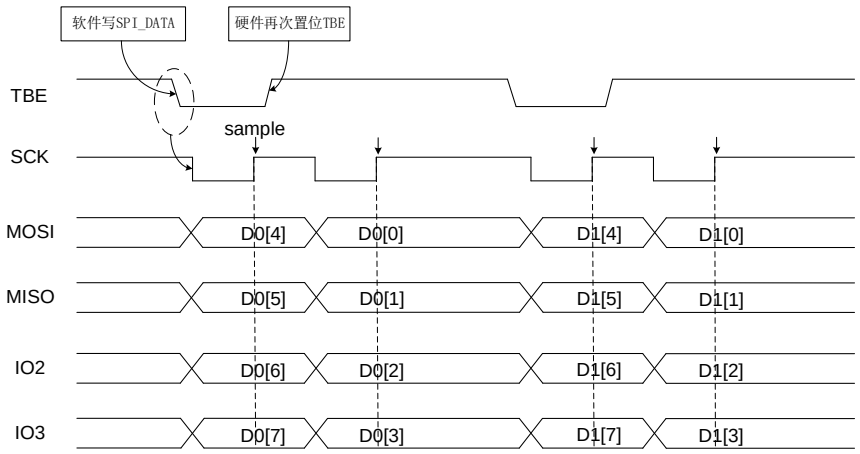
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 21-8. SPI 四线模式写操作时序图



四线读模式

当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI

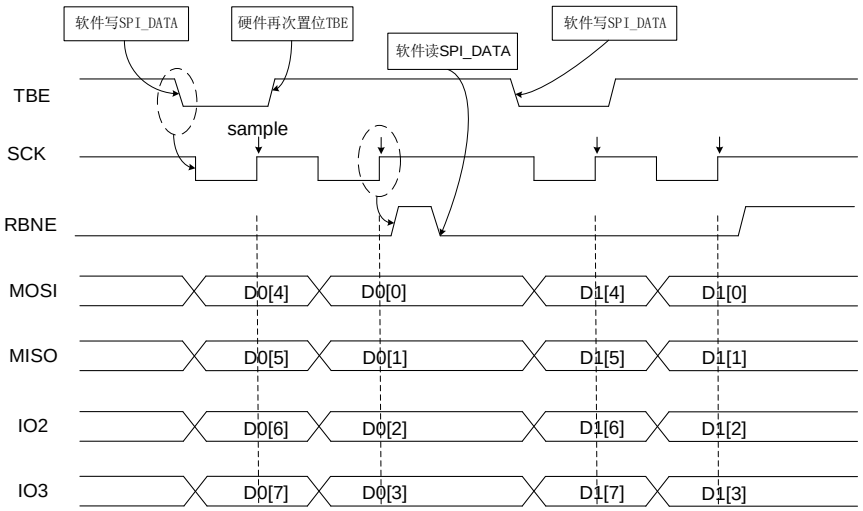
功能；

3. 写任意数据（例如0xFF）到SPI_DATA寄存器；

4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；

5. 写任意数据（例如0xFF）到SPI_DATA寄存器，以接收下一个字节数据。

图 21-9. SPI 四线模式读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能：

MFD SFD

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

MTU MTB STU STB

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

MRU MRB

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

21.3.6. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

21.3.7. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

如果是8位数据长度，CRC计算基于CRC8标准进行。如果是16位数据长度，CRC计算基于CRC16标准进行。如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

注意：当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

21.3.8. SPI 中断

状态标志位

- **发送缓冲区空标志位（TBE）**

当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区。

- **接收缓冲区非空标志位（RBNE）**

当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

■ **SPI 通信进行中标志位（TRANS）**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

■ **配置错误标志（CONFERR）**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

■ **接收过载错误（RXORERR）**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **CRC错误（CRCERR）**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

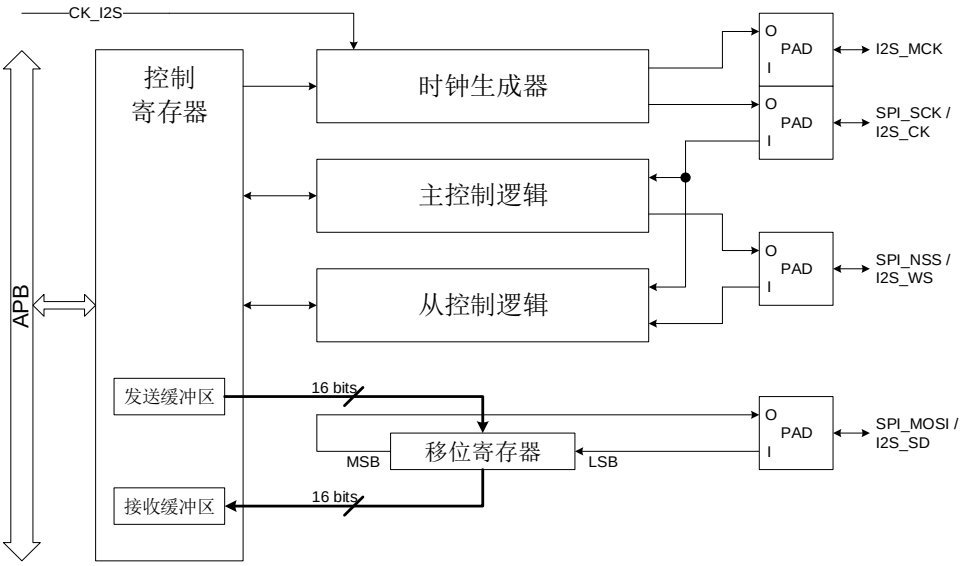
表 21-6. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE
RXORERR	接收过载错误	读SPI_DATA寄存器，然后读 SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	

21.4. I2S 功能说明

21.4.1. I2S 结构框图

图 21-10. I2S 结构框图



I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成I2S通信时钟。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

21.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它提供了一个256倍于Fs的时钟频率，其中Fs是音频采样率。

21.4.3. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道（左通道和右通道）的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数

据长度为24位或32位的数据帧传输，SPI_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI_DATA寄存器只需被访问1次。如需将16位数据打包成32位数据帧，硬件会自动插入16位0将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，I2S_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 21-11. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

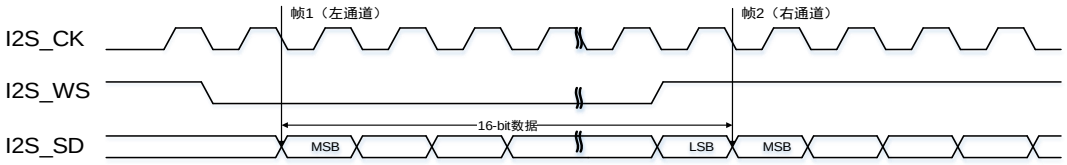
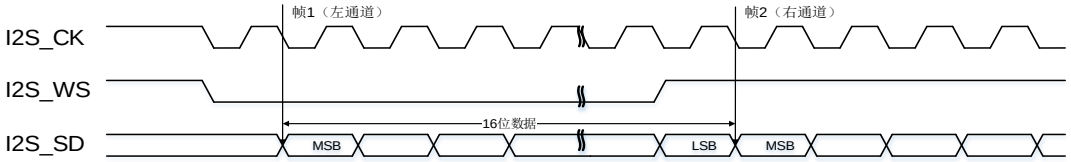


图 21-12. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。

图 21-13. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

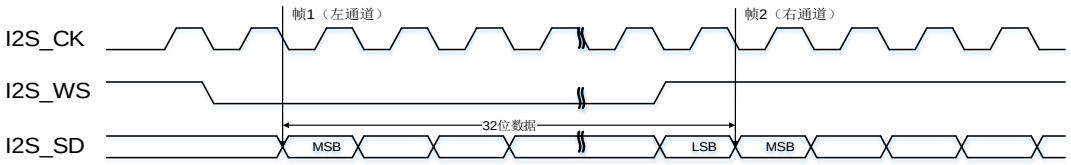
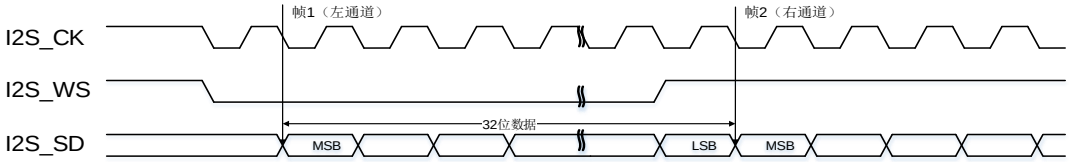


图 21-14. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

图 21-15. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

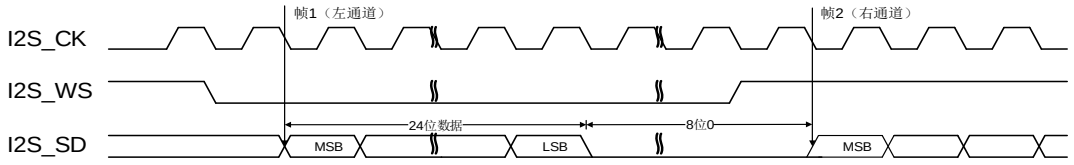
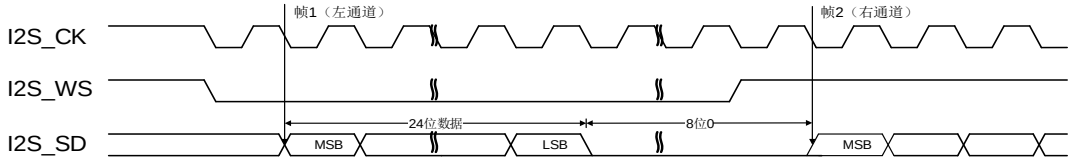


图 21-16. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全是0。

图 21-17. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

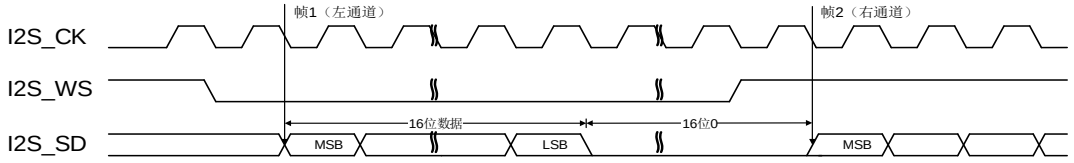
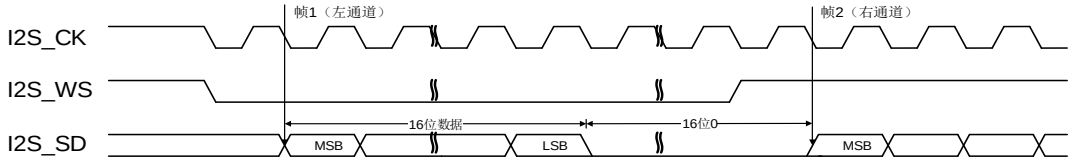


图 21-18. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 21-19. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

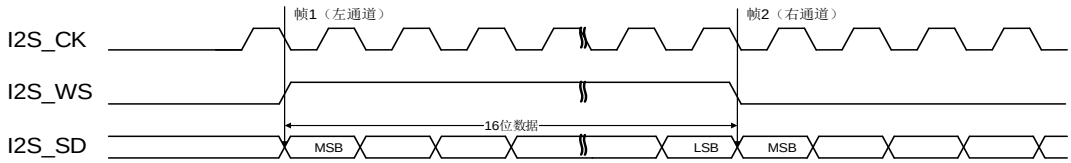


图 21-20. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）

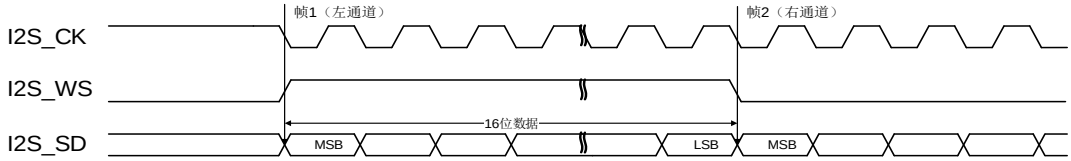


图 21-21. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

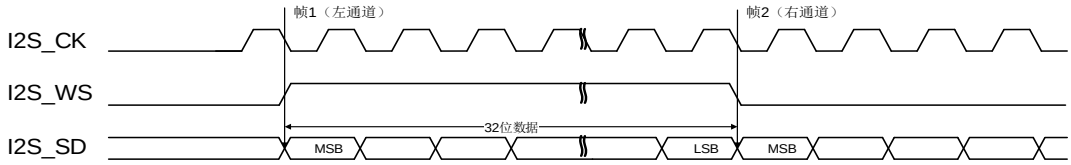


图 21-22. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）

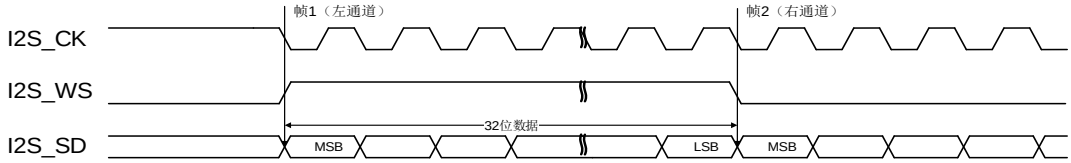


图 21-23. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

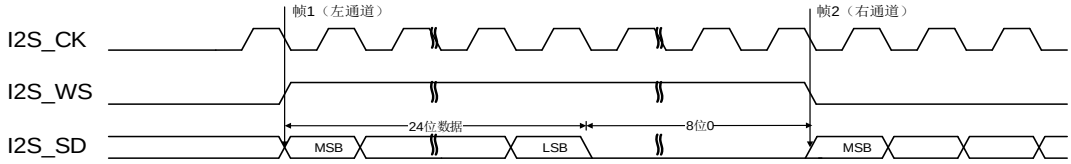


图 21-24. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）

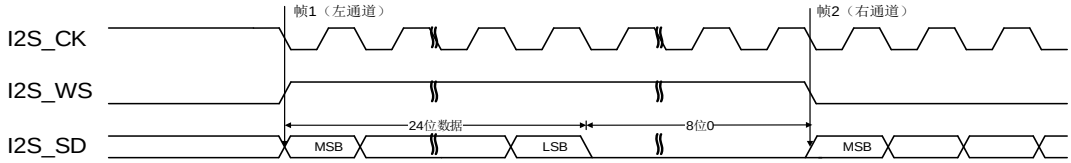


图 21-25. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

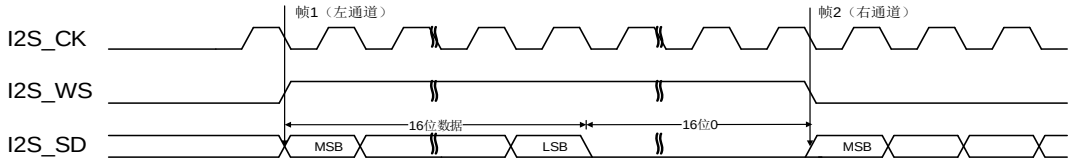
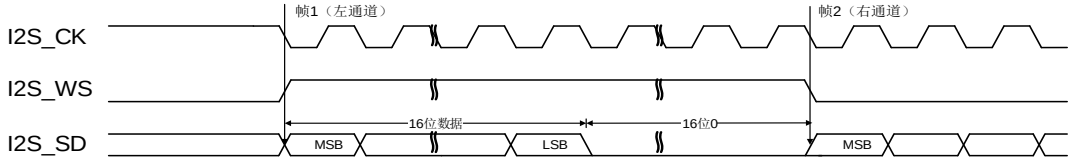


图 21-26. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 21-27. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

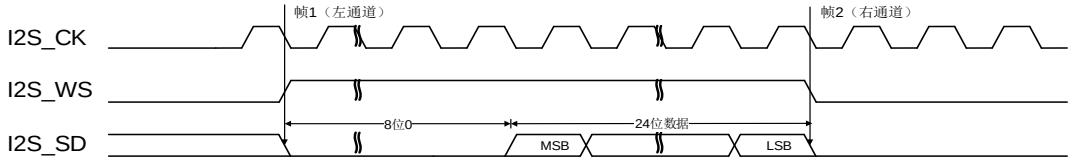
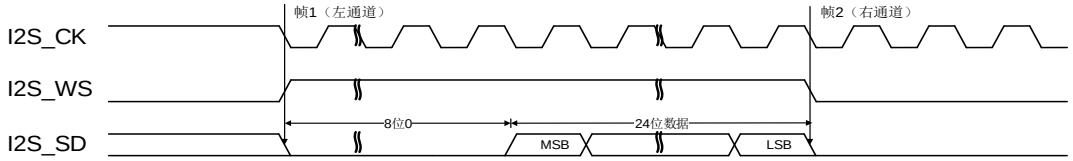


图 21-28. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

图 21-29. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

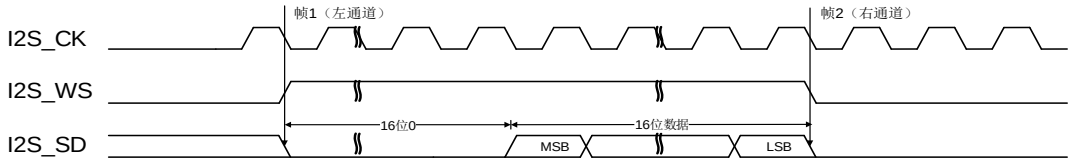
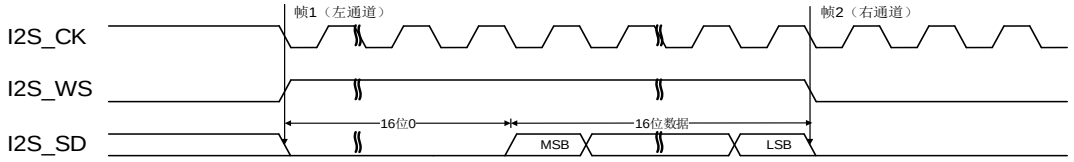


图 21-30. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 21-31. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

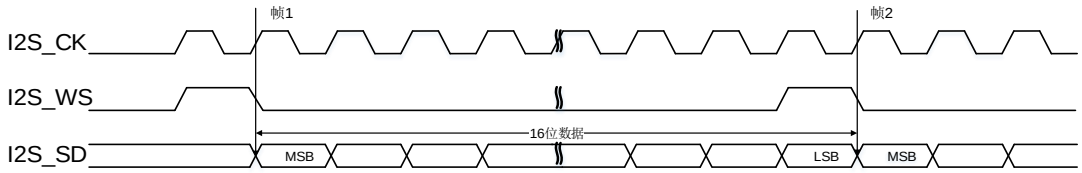


图 21-32. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

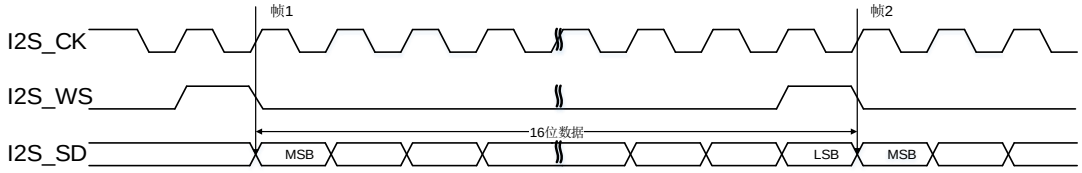


图 21-33. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

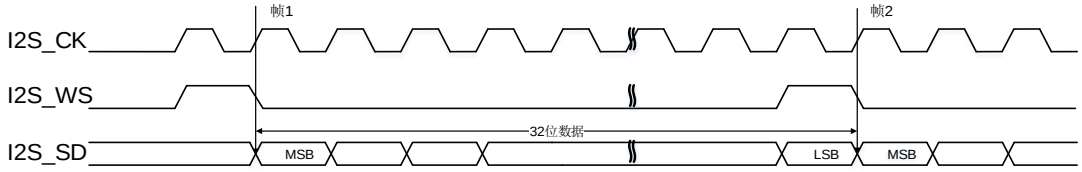


图 21-34. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

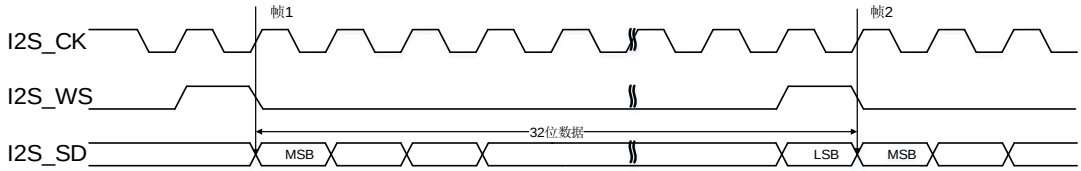


图 21-35. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

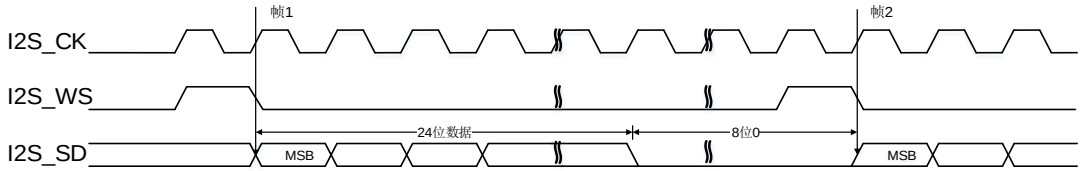


图 21-36. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

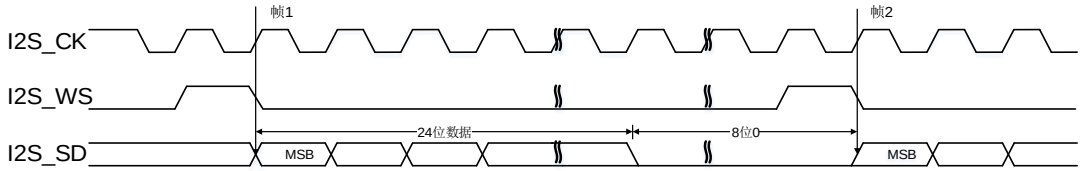


图 21-37. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

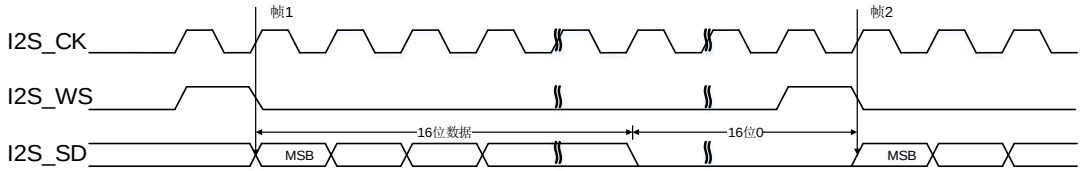
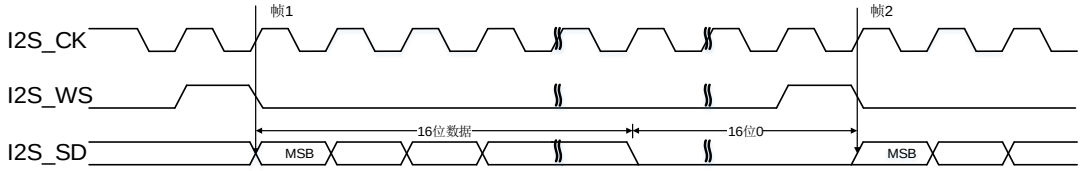


图 21-38. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



长帧同步模式的各种配置情况时序图如下所示。

图 21-39. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

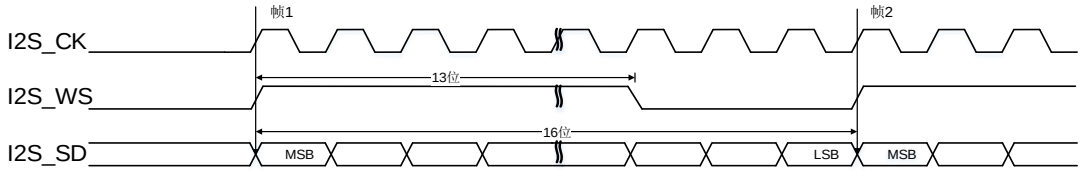


图 21-40. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

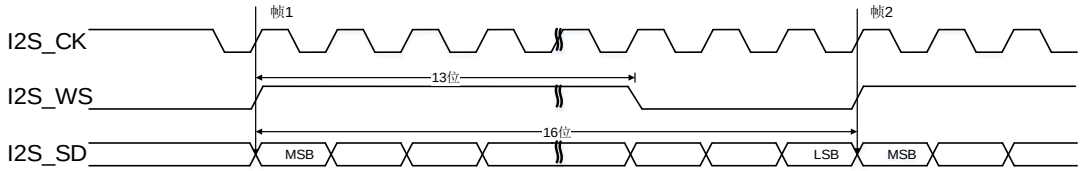


图 21-41. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

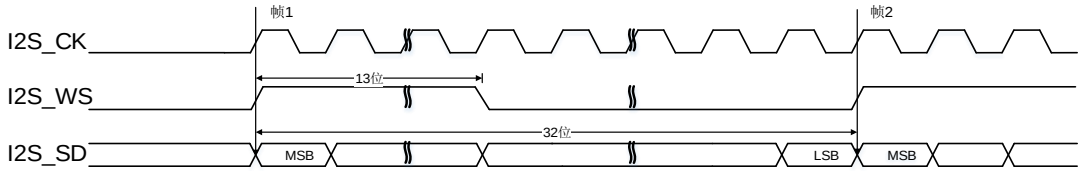


图 21-42. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

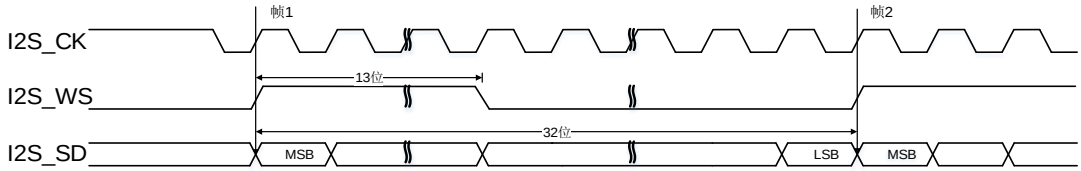


图 21-43. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

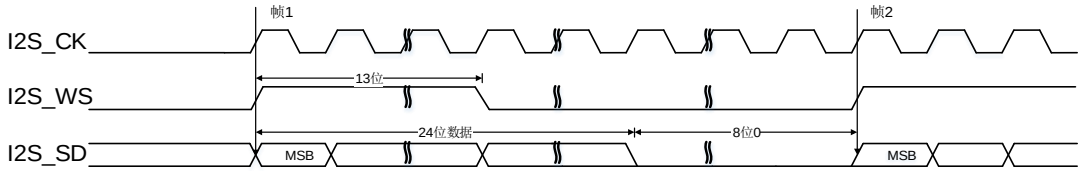


图 21-44. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

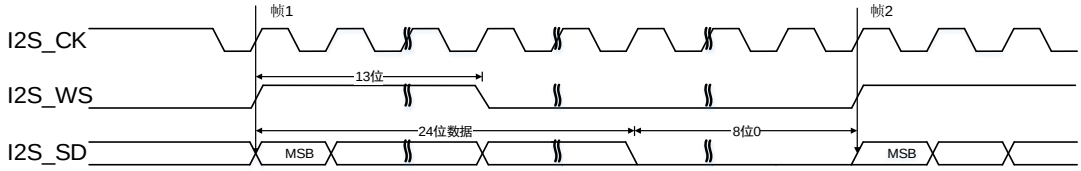


图 21-45. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

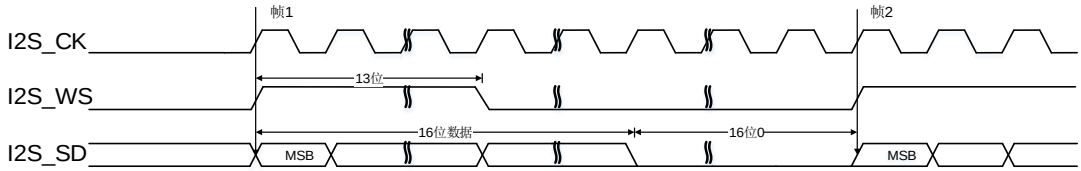
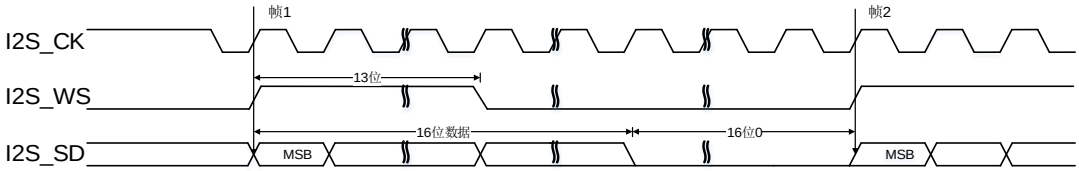
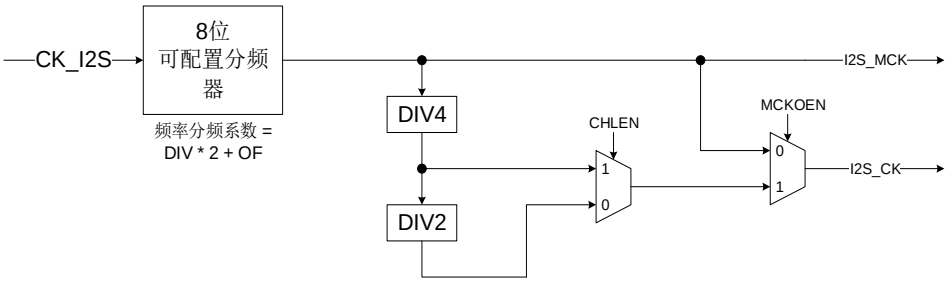


图 21-46. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



21.4.4. I2S 时钟

图 21-47. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图21-47. 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。I2S 比特率可以通过*表 21-7. I2S 比特率计算公式*所示的公式计算。

表 21-7. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

音频采样率（Fs）和I2S比特率的关系由如下公式定义：

Fs = I2S比特率 / (通道长度 * 通道数)

所以，为了得到期望的音频采样率，时钟生成器需要按*表21-8. 音频采样频率计算公式*所列的公式进行配置。

表 21-8. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

I2S时钟源可以由PLLI2S和外部I2S_CKIN引脚提供，在RCU模块进行配置。软件需要仔细计算I2S分频因子和PLLI2S以得到精确的音频采样频率。如果PLLI2S精度不能满足应用程序的要求，

可以通过I2S_CKIN引脚提供精确的外部I2S时钟。

21.4.5. 运行

运行模式

运行模式是通过SPI_I2SCTL寄存器的I2SOPMOD[1:0]位域来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下I2S接口信号的方向如*表21-9. 各种运行模式下I2S接口信号的方向*所示。

表 21-9. 各种运行模式下 I2S 接口信号的方向

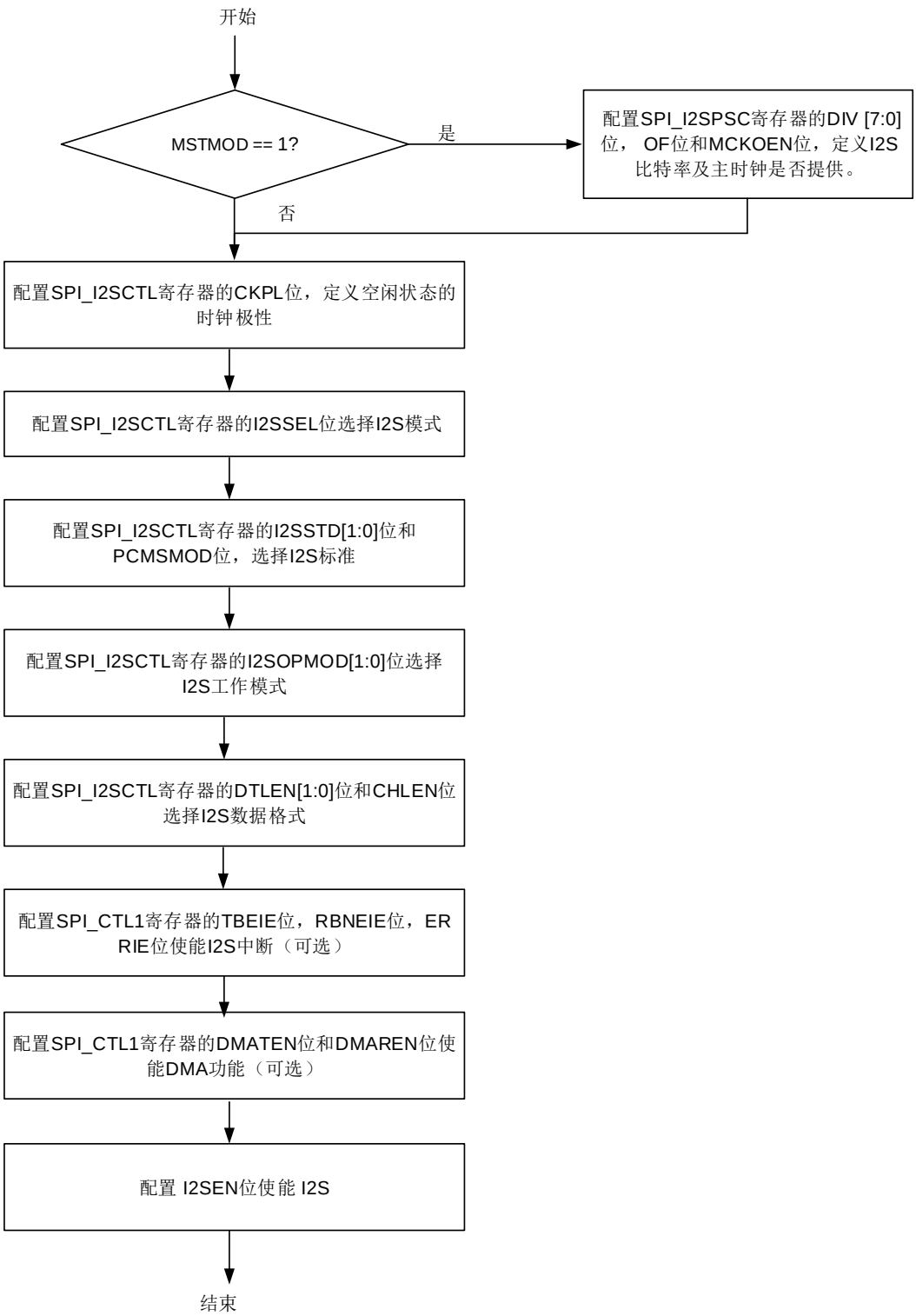
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程如*图21-48. I2S初始化流程*所示。

图 21-48. I2S 初始化流程



I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位置位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空（TBE为1），且移位

寄存器中没有发送序列。当16位数据被写入SPI_DATA寄存器时（TBE变为0），数据立即从发送缓冲区装载到移位寄存器中（TBE变为1）。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出（高位先发）。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

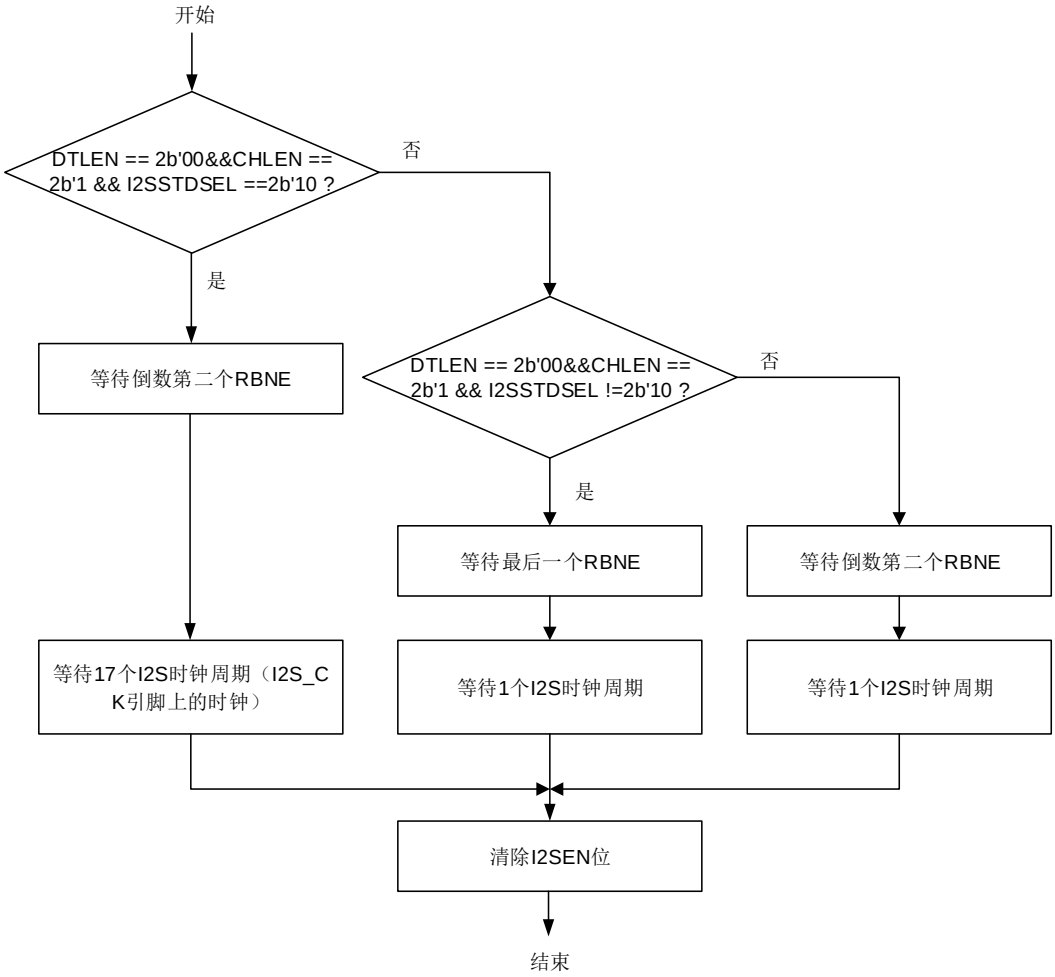
I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志置位表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空（RBNE为0）。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区（RBNE变为1）。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如*图21-49. I2S主机接收禁能流程*所示。

图 21-49. 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。

21.4.6. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

21.4.7. I2S 中断

状态标志位

SPI_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

- **发生缓冲区空标志（TBE）**

  当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个数据写入发送缓冲区。

- **接收缓冲区非空标志（RBNE）**

  接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

- **I2S通信进行中标志（TRANS）**

  TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法进行软件操作。该标志位不会产生如何中断。

- **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

有两个错误标志：

- **发送欠载错误标志（TXURERR）**

  在从发送模式下，有效的SCK信号开始发送，当发送缓冲区为空时，发送欠载错误标志TXURERR置位。

- **接收过载错误标志（RXORERR）**

  当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。

*表21-10. I2S中断*总结了I2S中断事件和相应的使能位。

表 21-10. I2S 中断

中断事件	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	

21.5. 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

21.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

该寄存器在I2S模式下没有意义。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式 0：2线单向传输模式 1：1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0：工作在只接收模式 1：工作在只发送模式
13	CRCEN	CRC计算使能 0：禁能CRC计算。 1：使能CRC计算。
12	CRCNT	下一次传输CRC 0：下一次传输值为数据 1：下一次传输值为CRC值（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
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。
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
6	SPIEN	SPI使能 0：禁能SPI 1：使能SPI
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用SPI0时，PCLK=PCLK2，当使用SPI1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采集第一个数据。

21.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	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	保留		NSSDRV	DMATEN	DMAREN
								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/接收FIFO非空中断使能 0：禁能RBNE中断 1：使能RBNE中断。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：禁能错误中断 1：使能错误中断。当CRCERR位，CONFERR位，RXORERR位或者TXURERR位置1时，产生中断。
4:3	保留	必须保持复位值。
2	NSSDRV	NSS输出使能 0：禁能主机NSS输出 1：使能主机NSS输出
1	DMATEN	发送缓冲区/发送FIFO DMA使能 0：禁能发送缓冲区/发送FIFO DMA 1：使能发送缓冲区/发送FIFO DMA。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。
0	DMAREN	接收缓冲区/接收FIFO DMA使能 0：禁能接收缓冲区/接收FIFO DMA 1：使能接收缓冲区/接收FIFO DMA。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。

21.5.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值0x0000 0002

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
								r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
31:8	保留	必须保持复位值。
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：先写SPI_STAT寄存器，然后读或写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，通过写SPI_STAT寄存器清除。 SPI模式下不使用该位。
2	I2SCH	I2S通道标志 0：下一个将要发送或接收的数据属于左通道 1：下一个要发送或接收的数据属于右通道

		该位由硬件置位和清除。 SPI模式下该位无用，I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RBNE	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

21.5.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。 当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。

21.5.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10
复位值：0x0000 0007

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

21.5.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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RCR[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RCR [15:0]。 硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

21.5.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	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TCR[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TCR[15:0]。 硬件在发送出每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。不同的数据帧格式（SPI_CTL0中的LF位决定）将会得到不同的CRC值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

21.5.8. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	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。

21.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）

地址偏移：0x20
复位值：0x0000 0002

该寄存器只能按字（32位）访问。

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

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

位/位域	名称	描述

31:10	保留	必须保持复位值。
9	MCKOEN	I2S_MCK输出使能 0：I2S_MCK输出禁止 1：I2S_MCK输出使能 当I2S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。

21.5.10. SPI0 四线 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													IO23_DRV	QRD	QMOD
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	IO23_DRV	IO2和IO3输出使能 0：单路模式下IO2和IO3输出关闭 1：单路模式下IO2和IO3输出高电平 该位仅适用于SPI0。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。
0	QMOD	四线SPI模式使能 0：SPI工作在单路模式

1：SPI工作在四线模式

该位仅能在SPI未通信时配置（TRANS位清零）。

该位仅适用于SPI0。

22. 数字摄像头接口（DCI）

22.1. 简介

数字摄像头接口是一个同步并行接口，可以从数字摄像头捕获视频和图像信息。它支持不同的颜色空间，例如YUV/RGB以及压缩数据JPEG。

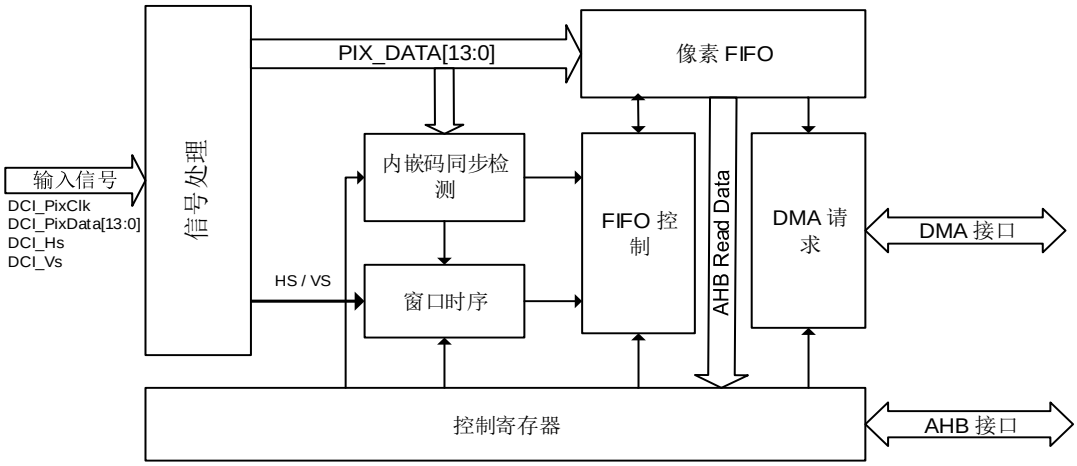
22.2. 主要特征

- 数字视频和图像的捕获；
- 支持8位、10位、12位或14位并行接口；
- DMA高效传输；
- 支持视频和图像裁剪；
- 支持不同的像素数字编码格式，如YCbCr422/RGB565；
- 支持JPEG压缩格式；
- 支持内嵌码同步和硬件同步。

22.3. 结构框图

数字摄像头接口包含以下模块：信号处理单元、像素FIFO、FIFO控制器、窗口时序发生器、内嵌码同步检测器、DMA接口和控制寄存器。

图 22-1. DCI 模块示意图



信号处理单元根据外部输入信号，产生有用的信号信息，为其他的内部模块所用。为确保信号处理单元工作正常，HCLK的频率要高于像素时钟频率的2.5倍。

内嵌码同步检测用于内嵌码同步模式。DCI使用内嵌码同步模式时，视频同步信息内嵌于像素数据，并无硬件水平或垂直同步信号(DCI_Hs或DCI_Vs)。DCI通过内嵌码同步检测器从像素数据提取同步信息，然后根据这些信息重新恢复水平和垂直同步信号。

窗口时序模块具有图片剪裁功能。该模块通过来自DCI接口或内嵌码同步检测器的同步信号计算像素点的位置，然后根据寄存器DCI_CWSPOS和DCI_CWSZ的配置决定是否接收该像素点数据。

DCI用一个4字(32位)FIFO缓存接收到的数据。如果DMA模式使能，每当收到一个完整的32位数据的时候，DMA接口就会发出一个DMA请求。控制寄存器提供DCI和软件之间的接口。

22.4. 信号描述

表 22-1. DCI 引脚

方向	名称	位宽	描述
I	DCI_PixClk	1	DCI 像素时钟
I	DCI_PixData	14	DCI 像素数据
I	DCI_Hs	1	DCI 水平同步
I	DCI_Vs	1	DCI 垂直同步

22.5. 功能说明

22.5.1. DCI 硬件同步模式

在DCI硬件同步模式(DCI_CTL寄存器的ESM为0)，DCI_Hs和DCI_Vs信号分别用来表示一行的开始和一帧的开始。DCI在DCI_PixClk的上升沿或下降沿(时钟的极性通过DCI_CTL寄存器的CKS位配置)，从DCI_PixData[13:0]，捕获像素数据。

图 22-2. 硬件同步模式

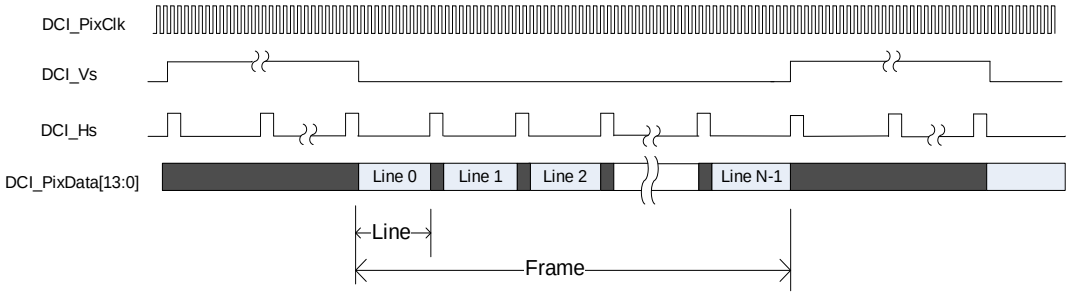
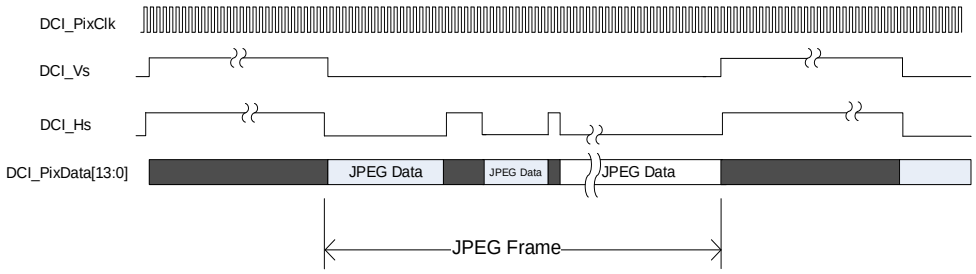


*图 22-2. 硬件同步模式*假设 DCI_Hs 和 DCI_Vs 消隐期间的极性为高电平，所以 DCI_PixData 线仅在 DCI_Hs 和 DCI_Vs 都为低电平期间是有效的。

JPEG 子模式

DCI在硬件同步模式时，支持JPEG视频/图像压缩格式。在JPEG子模式(DCI_CTL寄存器JM置1)，DCI_Vs表示一帧的开始，DCI_Hs用作数据流有效信号。

图 22-3. 硬件同步模式之 JPEG 子模式



22.5.2. 内嵌码同步模式

DCI支持内嵌码同步模式。这一模式仅用到DCI接口的DCI_PixData和DCI_PixClk信号，同步信息内嵌在像素数据中。通过置位DCI_CTL寄存器的ESM位，并且清除JM位，使能内嵌码同步模式。

在内嵌码同步模式，行和帧同步信息被编码为同步码并嵌入像素数据中。有4种同步码：行开始(LS)，行结束(LE)，帧开始(FS)和帧结束(FE)。该模式数据宽度强制为8，并且每个同步码由4字节序列组成：FF-00-00-MN，MN在DCI_SC寄存器定义。在内嵌码同步模式，0xFF和0x00不应出现在像素数据中以避免误解。

使能内嵌码同步模式之后，DCI开始检测同步码，并恢复行/帧同步信息。例如，如果DCI检测到一个帧结束码以及一个帧开始码，它开始捕获新的帧。

当检测到一个同步码，通过配置DCI_SCUMSK，可能仅需要比较FF_00_00_MN 序列MN字节的几位。DCI仅比较DCI_SCUMSK寄存器的非屏蔽位。例如：DCI_SC寄存器的LS位为A5，DCI_SCUMSK的LSM位是F0，DCI将仅比较LS同步码的高4位，因此FF-00-00-A6也将被检测为LS码。

22.5.3. 用快照或连续捕获模式捕获数据

DCI支持两种捕获模式：快照和连续捕获。捕获模式通过DCI_CTL寄存器的SNAP位配置。

正确配置之后，使能DCI并置位DCI_CTL寄存器的CAP位，DCI开始检测帧开始信号。一旦检测到帧开始信号，DCI开始捕获数据。在快照模式(SNAP=1)，当一帧被捕获之后，DCI自动停止捕获并清除CAP位，而若在连续模式，DCI将准备捕获下一帧。在连续模式，DCI捕获频率在FR[1:0]位域定义。例如，如果FR[1:0]=00，DCI捕获每一帧，如果FR[1:0]=01，DCI将每隔一帧捕获一次。

在连续模式，当DCI正在捕获数据的时候，软件可以在任意时间清除CAP位，但DCI并不立即停止捕获。它总是在捕获当前帧之后停止。软件应读回CAP位，以确认是否DCI停止生效。

22.5.4. 窗口功能

DCI支持窗口功能，该功能能够从捕获到的帧剪裁图像的一部分。该功能通过设置DCI_CTL寄存器的WDEN位，在JPEG子模式使能该功能是禁止的。

在捕获期间，DCI不断的计数和计算像素的水平和垂直位置，并且将该位置与剪裁窗口寄存器(DCI_CWSPOS和DCI_CWSZ)的值进行比较，然后丢弃剪裁窗口外的像素数据，仅将位于窗口内的数据压入数据FIFO。

如果一帧已经结束，但DCI_CWSZ定义的垂直行数还没有到达，这种情况下也将触发帧结束标志并且DCI停止捕获。

22.5.5. 像素格式，数据填充和 DMA 接口

DCI支持包含YCbCr422/RGB565等多种像素编码格式，但是DCI只接收这些像素数据，将像素数据补充成全字，并将其压入像素FIFO。DCI不执行任何像素格式转换或数据处理，不关心像素格式细节。

DCI使用32位宽的数据缓冲器在DCI接口和像素FIFO之间传递数据。在这一模块有两种填充方法：字节填充和半字填充，具体使用哪一种取决于DCI接口的数据宽度。数据宽度由DCI_CTL寄存器的DCIF[1:0]配置，在JPEG子模式和内嵌码同步模式，数据宽度固定为8。

每当收到一个完整的32位数据的时候，DMA接口发送DMA请求。

字节填充模式

如果DCI接口是8位，使用字节填充模式。在字节填充模式下，四个字节被填充到32位数据缓冲区，在Non-JPEG模式，如果数据缓冲区满或者到达行尾，DCI将压32位数据缓冲区的数据进入像素FIFO。在JPEG子模式，如果数据缓冲区满或者到达帧结束，DCI接口将压32位数据缓冲区的数据进入像素FIFO。

表 22-2. 字节填充模式下的存储视图

D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]
D7[7:0]	D6[7:0]	D5[7:0]	D4[7:0]

半字填充模式

如果DCI接口配置为10/12/14位，使用半字填充模式。在该模式下，通过高位填0，每像素数据扩展为16位。所以32位宽的数据缓冲区可以包含两个像素数据。当缓冲区满或行结束的时候，DC将压数据进入像素FIFO。

表 22-3. 半字填充模式下的存储视图

2’b00	D1[13:0]	2’b00	D0[13:0]
2’b00	D3[13:0]	2’b00	D2[13:0]
2’b00	D5[13:0]	2’b00	D4[13:0]
2’b00	D7[13:0]	2’b00	D6[13:0]

22.6. 中断

DCI有几个状态和错误标志位，中断可以根据这些标志判断。如果使能DCI_INTEN相应的使能

位，这些状态和错误标志触发DCI全局中断。这些标志可以通过写1到DCI_INTC寄存器清除。

表 22-4. 状态/错误标志

状态标志名	描述
ELF	行结束标志
EFF	帧结束标志
OVRF	FIFO 溢出标志
VSF	帧垂直同步消隐标志
ESEF	内嵌同步错误标志

22.7. DCI 寄存器

DCI 基地址：0x5005 0000

22.7.1. 控制寄存器（DCI_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DCIEN	保留		DCIF[1:0]		FR[1:0]		VPS	HPS	CKS	ESM	JM	WDEN	SNAP	CAP
	rw			rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	DCIEN	DCI使能 0：DCI禁止 1：DCI使能
13:12	保留	必须保持复位值。
11:10	DCIF[1:0]	DCI数据格式 00：每个像素时钟捕获8位数据 01：每个像素时钟捕获10位数据 10：每个像素时钟捕获12位数据 11：每个像素时钟捕获14位数据
9:8	FR[1:0]	帧频率 在连续捕获模式，FR定义帧捕获频率 00：捕获所有帧 01：每隔一帧捕获一次 10：每隔三帧捕获一次 11：保留
7	VPS	垂直同步极性选择 0：消隐期间低电平 1：消隐期间高电平
6	HPS	水平同步极性选择 0：消隐期间低电平

		1：消隐期间高电平
5	CKS	时钟极性选择 0：下降沿捕获 1：上升沿捕获
4	ESM	内嵌码同步模式 0：禁止内嵌码同步模式 1：使能内嵌码同步模式
3	JM	JPEG子模式 0：禁止JPEG子模式 1：使能JPEG子模式
2	WDEN	窗口使能 0：禁止窗口功能 1：使能窗口功能
1	SNAP	快照模式 0：连续捕获模式 1：快照模式
0	CAP	使能捕获 0：禁止帧捕获 1：使能帧捕获

22.7.2. 状态寄存器 0（DCI_STAT0）

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													FV	VS	HS
													r	r	r

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	FV	FIFO有效 0：FIFO没有有效像素数据 1：FIFO中像素数据有效

1	VS	VS引脚状态 0：不在垂直消隐期间 1：处于垂直消隐期间
0	HS	HS引脚状态 0：不在水平消隐期间 1：处于水平消隐期间

22.7.3. 状态寄存器 1（DCI_STAT1）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ELF	VSF	ESEF	OVRF	EFF
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	ELF	行结束标志 0：没有行结束标志 1：DCI捕获到一行
3	VSF	垂直同步标志 0：没有垂直同步标志 1：检测到垂直同步消隐
2	ESEF	内嵌码同步错误标志 0：没有内嵌码同步错误标志 1：检测到内嵌码同步错误
1	OVRF	FIFO溢出标志 0：没有FIFO溢出 1：发生FIFO溢出
0	EFF	帧结束标志 0：没有帧结束标志 1：帧被DCI捕获

22.7.4. 中断使能寄存器（DCI_INTEN）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ELIE	VSIE	ESEIE	OVRIE	EFIE
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	ELIE	行结束中断使能 0：行结束标志不产生中断 1：行结束标志产生中断
3	VSIE	垂直同步中断使能 0：垂直同步标志不产生中断 1：垂直同步标志产生中断
2	ESEIE	内嵌码同步错误中断使能 0：内嵌码同步错误标志不产生中断 1：内嵌码同步错误标志产生中断
1	OVRIE	FIFO溢出中断使能 0：FIFO溢出不产生中断 1：FIFO溢出产生中断
0	EFIE	帧结束中断使能 0：帧结束标志不产生中断 1：帧结束标志产生中断

22.7.5. 中断标志寄存器（DCI_INTF）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ELIF	VSIF	ESEIF	OVRIF	EFIF
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	ELIF	行结束中断标志
3	VSMF	垂直同步中断标志
2	ESEMF	内嵌码同步错误中断标志
1	OVRMF	FIFO溢出中断标志
0	EFMF	帧结束中断标志

22.7.6. 中断标志清除寄存器（DCI_INTC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											ELFC	VSFC	ESEFC	OVRFC	EFFC
											w	w	w	w	w

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	ELFC	行结束中断标志清除 写1清除行结束中断标志
3	VSFC	垂直同步标志清除 写1清除垂直同步标志
2	ESEFC	内嵌码同步错误标志清除 写1清除内嵌码同步错误标志
1	OVRFC	FIFO溢出标志清除 写1清除FIFO溢出标志
0	EFFC	帧结束中断标志清除 写1清除帧结束中断标志

22.7.7. 同步码寄存器（DCI_SC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:24	FE[7:0]	内嵌同步模式的帧结束码
23:16	LE[7:0]	内嵌同步模式的行结束码
15:8	LS[7:0]	内嵌同步模式的行开始码
7:0	FS[7:0]	内嵌同步模式的帧开始码

22.7.8. 同步码屏蔽寄存器（DCI_SCUMSK）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:24	FEM[7:0]	内嵌码同步模式下非屏蔽帧结束码
23:16	LEM[7:0]	内嵌码同步模式下非屏蔽行结束码
15:8	LSM[7:0]	内嵌码同步模式下非屏蔽行开始码
7:0	FSM[7:0]	内嵌码同步模式下非屏蔽帧开始码

22.7.9. 剪裁窗口开始位置寄存器（DCI_CWSPOS）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			WVSP[12:0]												
			rw												

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:16	WVSP[12:0]	窗口垂直开始位置 值为0表示着第一行，以此类推
15:14	保留	必须保持复位值。
13:0	WHSP[13:0]	窗口水平开始位置 值为0表示着第一个像素，以此类推

22.7.10. 剪裁窗口大小寄存器（DCI_CWSZ）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		WVSZ[13:0]													
		rw													

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	WVSZ[13:0]	窗口垂直大小 WVSZ=X表示X+1行
15:14	保留	必须保持复位值。

13:0	WHSZ[13:0]	窗口水平大小 WHSZ=X表示一行有X+1个像素点

22.7.11. 数据寄存器（DCI_DATA）

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DT3[[7:0]								DT2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT1[7:0]								DT0[7:0]							
r								r							

位/位域	名称	描述
31:24	DT3[7:0]	像素字节3
23:16	DT2[7:0]	像素字节2
15:8	DT1[7:0]	像素字节1
7:0	DT0[7:0]	像素字节0

23. TFT-LCD 接口（TLI）

23.1. 简介

TLI(TFT-LCD接口)连接同步的LCD接口，并且为无源LCD显示屏提供提供像素数据，时钟以及时序信号。它支持不同的完全可编程的时序参数显示。一个内嵌的DMA不断的从系统存储器搬移数据到TLI然后输出到外部的LCD显示。TLI支持两个独立的显示层，并支持层窗口和层混叠功能。

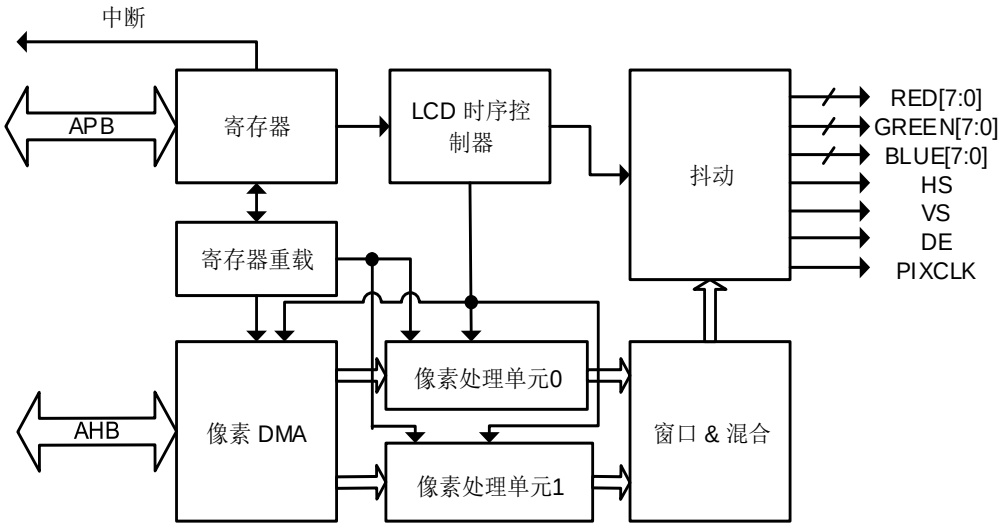
23.2. 主要特征

- 每像素最多24位并行数据输出；
- 支持高达800*600的分辨率；
- 完全可编程的时序参数；
- 内嵌DMA处理像素数据拷贝；
- 带有窗口和混合功能的两个独立的层；
- 支持多种像素格式：ARGB8888，RGB888，RGB565等；
- 支持CLUT(颜色查找表)和色键格式；
- 像素低位的抖动操作。

23.3. 结构框图

*图 23-1. TLI 模块框图*展示了 TLI 模块的结构框图。在 TLI 模块有 3 个时钟域。寄存器工作在 APB 时钟域，通过 APB 总线访问。像素 DMA 模块工作在 AHB 时钟域，从系统存储器获取像素数据需要使用 AHB 总线。剩下的模块工作在 TLI 时钟域。TLI 时钟由 PLLTSEL，PLLTPSC，PLLTMF，PLLTRPSC 和 TLIPSC 配置，具体请参考 RCU 模块的 ***PLLT 配置寄存器（RCU_PLLTCFG***。

图 23-1. TLI 模块框图



23.4. 信号线描述

TLI 提供一个 24 位 RGB 并行显示接口，如*表 23-1. TLI 提供的显示接口的引脚*所示。

表 23-1. TLI 提供的显示接口的引脚

方向	名称	位宽	描述
O	HS	1	水平同步
O	VS	1	垂直同步
O	DE	1	数据使能
O	PIXCLK	1	像素时钟
O	RED[7:0]	8	红色像素数据
O	GREEN[7:0]	8	绿色像素数据
O	BLUE[7:0]	8	蓝色像素数据

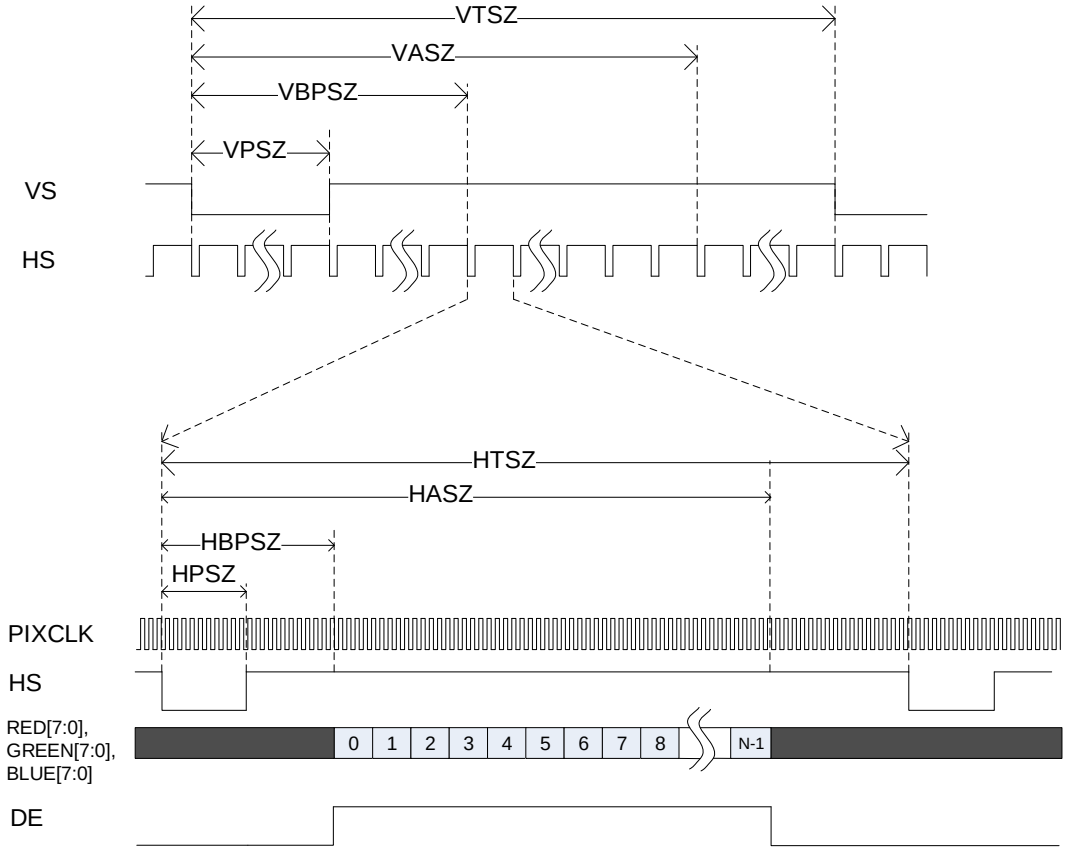
23.5. 功能说明

23.5.1. LCD 显示时序

LCD 接口是一个同步数据接口，包括像素时钟，像素数据以及水平和垂直同步信号。*图 23-2. 显示时序图*展示了一帧的 HS 和 VS 信号时序。时序参数在 TLI_SPSZ，TLI_BPSZ，TLI_ASZ

和 TLI_TSZ 寄存器中配置。这些时序值寄存器假设第一个点的位置是(0,0)。

图 23-2. 显示时序图



23.5.2. 像素 DMA 功能

根据寄存器模块的配置，像素DMA不断的从存储器读像素数据到内部PPU(像素处理单元)的像素缓冲区。

使能之后，像素DMA开始从系统取像素数据，只要像素缓冲区未满，就将这些数据压进PPU的像素缓冲区。

TLI支持两个独立的帧层，在系统中每层有独立的帧缓冲区地址。像素DMA仅有一个AHB访问端口，所以如果两层都使能的话，在取像素的时候，它将交替读取两层的数据。

TLI_LxFBADDR寄存器的FBADD位定义了每层的帧缓冲区地址。

TLI_LxFLLEN寄存器的FLL定义了一帧数据行的长度，以字节为单位。如果一行的长度是N，FLL=N+3。

在系统存储器里，两行之间可能有一些间隔存储空间，这一间隔空间信息在TLI_LxFLLEN寄存器的STDOFF位域定义。例如，如果某行第一个像素的地址是M，那么下一行第一个像素的地址将是M+STDOFF。如果两行之间无间隔存储空间，则STDOFF=FLL-3。

TLI_LxFTLN寄存器的FTLN位域定义了一帧数据的行数。

23.5.3. 像素格式

像素DMA以字为单位，将像素数据压入PPU，然后由PPU负责将各种像素格式转换成内部ARGB8888格式。如*表23-2. 八种像素格式*所示，TLI支持多达8种像素格式。TLI_LxPPF寄存器的PPF[2:0]位域定义了像素格式。

ARGB8888格式要求每通道(Alpha, Red, Green和Blue)有8位数据。但是ARGB1555 和ARGB4444格式的某些通道是少于8位的。 PPU通过拷贝高位填充到低位的方式，将其转换成ARGB8888。当处理RGB888和RGB565格式时，PPU假设Alpha=255，并且如果通道的位数少于8，也将拷贝高位填充到低位。

AL88，AL44和L8格式是LUT(颜色查找表)格式。在这些通道里，L是颜色查找表的地址。TLI有两个内部颜色查找表：每层各一个。内部颜色查找表的大小是256x24位(256个节点，每节点存储24位RGB值)。当处理LUT格式像素时，PPU从颜色查找表读出一个节点，并用这个节点值作为RGB值。由于颜色查找表的地址是8位的，如果L通道的位数少于8位，PPU也将拷贝高位填充到低位。颜色查找表的节点在复位后是不会被初始化的，因此在显示一个颜色查找表格式层之前，应用程序应该用TLI_LxLUT寄存器，写入适当的值初始化颜色查找表。TLI_LxLUT寄存器是只写寄存器，对该寄存器的写操作将会对查色表写入一个节点。

每层都支持色键模式。TLI_LxCKEY 寄存器定义了一个RGB值。当某层的色键模式使能，PPU将会把该层每一个像素的RGB值与TLI_LxCKEY中的值相比较，如果值匹配，将会置该像素的ARGB值为0。

表 23-2. 八种像素格式

PPF[2:0]	像素格式
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444
111	AL88
101	L8
110	AL44

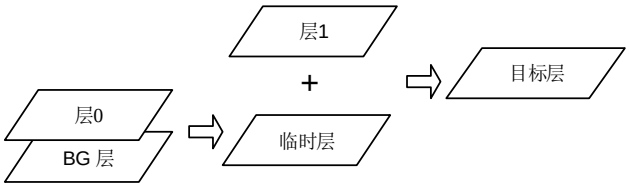
23.5.4. 层窗口和混合功能

TLI每层都支持窗口功能以及两层的混合功能。TLI首先执行每层的窗口操作，然后将两层混合成一帧。

窗口功能定义了一个显示窗口，每层有独立的窗口参数，由寄存器TLI_LxHPOS和TLI_LxVPOS定义。窗口参数定义了一层内的显示窗口。窗口内的像素将保持它的原值，但是窗口外的像素值将被TLI_LxDC寄存器定义的像素值替代。

混合单元首先混合层0和BG层得到一个临时层，然后混合层1和临时层，得到目的层。BG层的ARGB值在TLI_BGC寄存器定义。如果某层被禁止，混合功能使用该层的默认颜色。

图 23-3. 混合过程框图



混合公式

通用混合公式:

$$BC=BF_1 \times C+BF_2 \times C_S \quad (式\ 23\text{-}1)$$

- *BC* 是混合的颜色
- BF_1 是混合因子 1
- *C* 是当前层颜色
- BF_2 是混合因子 2
- C_S 是底层混合的颜色

当前像素的混合因子有两种取值，由寄存器配置。一种是归一化的像素Alpha乘以归一化的恒定Alpha，另一种是归一化的恒定Alpha。

23.5.5. Layer 配置重载

如上面所描述的，每层有自己的帧缓冲区，像素格式，窗口，默认颜色配置寄存器并且每个寄存器都有影子寄存器。影子寄存器与真正的寄存器共享相同的地址。每次当应用程序对层相关的寄存器地址执行写操作，相应的影子寄存器立即更新，但是直到一个重载操作前真正寄存器的值是不会改变的，而只有真正寄存器的值才是影响TLI功能的。

应用程序有两种方法触发一个重载操作：请求重载和帧消隐重载。对于请求重载模式，在应用程序设置TLI_RL寄存器的RQR位之后，TLI立即加载影子寄存器的值到真正寄存器。对于帧消隐重载模式，设置TLI_RL寄存器的FBR位之后，TLI等待帧垂直消隐，然后加载影子寄存器的值到真正寄存器。在两种模式下，重载成功完成之后，硬件自动清除RQR或FBR位。

23.5.6. 抖动

抖动模块为每一个像素通道加一个2位的伪随机值。当18位接口用来显示24位数据的时候，该功能能够使图像更平滑。应用程序可以用TLI_CTL寄存器的DFEN位开启这一功能。

23.5.7. 中断

TLI中有以下错误和状态标志位。中断可以从这些状态判定，状态标志可以触发一个全局中断，错误标志将触发错误中断。

表 23-3. 状态标志

状态标志位	描述
LMF	行标记标志

LCRF	层配置重载标志

表 23-4. 错误标志

错误标志位	描述
TEF	传输错误标志
FEF	FIFO 错误标志

23.6. TLI 寄存器

TLI 基地址：0x4001 6800

23.6.1. 同步脉冲宽度寄存器（TLI_SPSZ）

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				HPSZ[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HPSZ[11:0]	水平同步脉冲宽度 HPSZ值应该配置成水平同步脉冲像素的个数减1。
15:12	保留	必须保持复位值。
11:0	VPSZ[11:0]	垂直同步脉冲宽度 VPSZ值应该配置成垂直同步脉冲像素的个数减1。

23.6.2. 后沿宽度寄存器（TLI_BPSZ）

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				HBPSZ[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HBPSZ[11:0]	水平后沿加同步脉冲的宽度 HBPSZ值应该配置成水平后沿像素个数加同步脉冲像素个数减1
15:12	保留	必须保持复位值。
11:0	VBPSZ[11:0]	垂直后沿加同步脉冲的宽度 VBPSZ值应该配置成垂直后沿像素个数加同步脉冲像素个数减1

23.6.3. 有效宽度寄存器（TLI_ASZ）

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				HASZ[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HASZ[11:0]	水平有效宽度加后沿像素和水平同步像素宽度 HASZ值应该配置成水平有效宽度加后沿像素和水平同步像素个数减1。
15:12	保留	必须保持复位值。
11:0	VASZ[11:0]	垂直有效宽度加后沿像素和垂直同步像素宽度 VASZ值应该配置成垂直有效宽度加后沿像素和垂直同步像素个数减1。

23.6.4. 总宽度寄存器(TLI_TSZ）

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				HTSZ[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HTSZ[11:0]	显示器的水平总宽度，包括有效宽度，后沿，同步脉冲和前沿 HTSZ值应该配置成水平有效宽度像素的个数加后沿像素，前沿像素和同步脉冲像素减1。
15:12	保留	必须保持复位值。
11:0	VTSZ[11:0]	显示器的垂直总宽度，包括有效宽度，后沿，同步脉冲和前沿 VTSZ值应该配置成垂直有效宽度像素的个数加后沿像素，前沿像素和同步脉冲像素减1。

23.6.5. 控制寄存器（TLI_CTL）

地址偏移：0x18
复位值：0x0000 2220

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPPS	VPPS	DEPS	CLKPS	保留											DFEN
rw	rw	rw	rw												rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RDB[2:0]			保留	GDB[2:0]			保留	BDB[2:0]			保留			TLIEN
	r				r				r						rw

位/位域	名称	描述
31	HPPS	水平脉冲极性选择 0：水平同步脉冲低电平有效 1：水平同步脉冲高电平有效
30	VPPS	垂直脉冲极性选择 0：垂直同步脉冲低电平有效 1：垂直同步脉冲高电平有效
29	DEPS	数据使能极性选择 0：数据使能低电平有效 1：数据使能高电平有效

28	CLKPS	像素时钟极性选择 0：像素时钟是TLI时钟 1：像素时钟是TLI时钟翻转
27:17	保留	必须保持复位值。
16	DFEN	抖动功能使能 0：禁止抖动功能 1：使能抖动功能
15	保留	必须保持复位值。
14:12	RDB[2:0]	红色通道抖动位数 固定为2，只读
11	保留	必须保持复位值。
10:8	GDB[2:0]	绿色通道抖动位数 固定为2，只读
7	保留	必须保持复位值。
6:4	BDB[2:0]	蓝色通道抖动位数 固定为2，只读
3:1	保留	必须保持复位值。
0	TLIEN	TLI使能位 0：禁止TLI 1：使能TLI

23.6.6. 重载层配置寄存器（TLI_RL）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	FBR	帧消隐重载请求 此位通过软件置位，在重载之后由硬件清除。

		0：禁止重载 1：层配置将在帧消隐时被重载进入真正寄存器
0	RQR	立即重载请求 此位通过软件置位，在重载之后由硬件清除。 0：禁止重载 1：层配置将在该位置位之后被重载进入真正寄存器

23.6.7. 背景色配置寄存器（TLI_BGC）

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								BVR[7:0]							
												rw			

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	BVR[7:0]	背景红色值
15:8	BVG[7:0]	背景绿色值
7:0	BVB[7:0]	背景蓝色值

23.6.8. 中断使能寄存器（TLI_INTEN）

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
保留												LCRIE	TEIE	FEIE	LMIE
												rw	rw	rw	rw

位/位域	名称	描述

31:4	保留	必须保持复位值。
3	LCRIE	层配置重载中断使能 0：层配置重载标志将不产生中断 1：层配置重载标志将产生中断
2	TEIE	传输错误中断使能 0：传输错误标志将不产生中断 1：传输错误标志将产生中断
1	FEIE	FIFO错误中断使能 0：FIFO错误标志将不产生中断 1：FIFO错误标志将产生中断
0	LMIE	行标记中断使能 0：行标记标志将不产生中断 1：行标记标志将产生中断

23.6.9. 中断标志寄存器（TLI_INTF）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												LCRF	TEF	FEF	LMF
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	LCRF	层配置重载标志 0：无层配置重载标志出现 1：由TLI_RL寄存器的FBR位置位触发了层配置重载
2	TEF	传输错误标志 0：无传输错误 1：一个传输错误在AHB总线上出现
1	FEF	FIFO错误标志 0：无FIFO错误标志 1：出现FIFO下溢错误
0	LMF	行标记标志

0：没有行标记标志

1：行数达到TLI_LM寄存器中设置的特定值

23.6.10. 中断标志清除寄存器（TLI_INTC）

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												LCRC	TEC	FEC	LMC
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	LCRC	层配置重载标志清除 写1清除层配置重载标志
2	TEC	传输错误标志清除 写1清除传输错误标志
1	FEC	FIFO错误标志清除 写1清除FIFO错误标志
0	LMC	行标记标志清除 写1清除行标记标志

23.6.11. 行标记寄存器（TLI_LM）

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:11	保留	必须保持复位值
10:0	LM[10:0]	行标记值 当行数到达该值，TLI_INTF寄存器的LMF位将置位。

23.6.12. 当前像素位置寄存器（TLI_CPPOS）

地址偏移：0x44
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:16	HPOS[15:0]	水平位置 当前显示的像素的水平位置
15:0	VPOS[15:0]	垂直位置 当前显示的像素的垂直位置

23.6.13. 状态寄存器（TLI_STAT）

地址偏移：0x48
复位值：0x0000 000F

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												HS	VS	HDE	VDE
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	HS	TLI当前的HS状态

2	VS	TLI当前的VS状态
1	HDE	当前的HDE状态 0：TLI_CPPOS寄存器HPOS并未位于TLI_BPSZ寄存器HBPSZ与TLI_ASZ寄存器HASZ之间 1：TLI_CPPOS寄存器HPOS位于TLI_BPSZ寄存器HBPSZ与TLI_ASZ寄存器HASZ之间
0	VDE	当前的VDE状态 0：TLI_CPPOS寄存器VPOS并未位于TLI_BPSZ寄存器VBPSZ与TLI_ASZ寄存器HASZ之间 1：TLI_CPPOS寄存器VPOS位于TLI_BPSZ寄存器VBPSZ与TLI_ASZ寄存器VASZ之间

23.6.14. 第 x 层控制寄存器（TLI_LxCTL）

地址偏移：0x84 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	LUTEN	LUT使能 0：禁止LUT 1：使能LUT
3:2	保留	必须保持复位值。
1	CKEYEN	色键使能 0：禁止色键功能 1：使能色键功能
0	LEN	层使能 0：禁止层 1：使能层

23.6.15. 第 x 层水平位置参数寄存器（TLI_LxHPOS）

地址偏移：0x88 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				WRP[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	WRP[11:0]	窗口右侧位置
15:12	保留	必须保持复位值。
11:0	WLP[11:0]	窗口左侧位置

23.6.16. 第 x 层垂直位置参数寄存器（TLI_LxVPOS）

地址偏移：0x8C + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				WBP[11:0]											
				rw											

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	WBP[11:0]	窗口底部位置
15:12	保留	必须保持复位值。
11:0	WTP[11:0]	窗口顶部位置

23.6.17. 第 x 层色键值寄存器（TLI_LxCKEY）

地址偏移：0x90 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:16	CKEYR [7:0]	色键红色值
15:8	CKEYG [7:0]	色键绿色值
7:0	CKEYB [7:0]	色键蓝色值

如果某层的像素RGB值等于TLI_LxCKEY寄存器定义的值，该像素RGB值复位为0。这意味着这些像素对其它层来说是透明的。

23.6.18. 第 x 层像素格式寄存器（TLI_LxPPF）

地址偏移：0x94 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	PPF[2:0]	像素格式 这些位配置像素格式 000：ARGB8888 001：RGB888 010：RGB565

011：ARGB1555

100：ARGB4444

101：L8

110：AL44

111：AL88

23.6.19. 第 x 层恒定 Alpha 寄存器（TLI_LxSA）

地址偏移：0x98 + 0x80 * x (x=0 or 1)

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
保留															

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	SA[7:0]	恒定Alpha 可用于计算混合因子。

23.6.20. 第 x 层默认颜色寄存器（TLI_LxDC）

地址偏移：0x9C + 0x80 * x (x=0 or 1)

复位：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:24	DCA[7:0]	默认颜色ALPHA
23:16	DCR[7:0]	默认颜色红色
15:8	DCG[7:0]	默认颜色绿色

7:0	DCB[7:0]	默认颜色蓝色

注意：当该层被禁止或TLI_LxHPOS和TLI_LxVPOS定义的窗口之外，默认颜色值生效。

23.6.21. 第 x 层混合寄存器（TLI_LxBLEND）

地址偏移：0xA0 + 0x80 * x (x=0 or 1)
复位值：0x0000 0607

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					ACF1[2:0]			保留					ACF2[2:0]		
					rw									rw	

位/位域	名称	描述
31:11	保留	必须保持复位值。
10:8	ACF1[2:0]	Alpha混合因子1计算方法 000：保留 001：保留 010：保留 011：保留 100：归一化的恒定Alpha 101：保留 110：归一化的像素Alpha乘以归一化的恒定Alpha 111：保留
7:3	保留	必须保持复位值。
2:0	ACF2[2:0]	Alpha混合因子2计算方法 000：保留 001：保留 010：保留 011：保留 100：保留 101：1-归一化的恒定Alpha 110：保留 111：1-归一化的像素Alpha乘以归一化的恒定Alpha

23.6.22. 第 x 层帧基地址寄存器（TLI_LxFBADDR）

地址偏移：0xAC + 0x80 * x (x=0 or 1)

复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:0	FBADD[31:0]	帧缓冲区基地址 帧缓冲区基地址

23.6.23. 第 x 层行长度寄存器（TLI_LxFLLEN）

地址偏移：0xB0 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		STDOFF[13:0]													
		rw													

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:16	STDOFF[13:0]	步幅偏移 这个值定义了从某行起始处到下一行起始处之间的字节数
15:14	保留	必须保持复位值。
13:0	FLL[13:0]	行长度 这个值为一行的字节数+3

23.6.24. 第 x 层总行数寄存器（TLI_LxFTLN）

地址偏移：0xB4 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

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

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

位/位域	名称	描述
31:11	保留	必须保持复位值。
10:0	FTLN[10:0]	总行数 这个值定义了一帧行数

23.6.25. 第 x 层颜色查找表寄存器（TLI_LxLUT）

地址偏移：0xC4 + 0x80 * x (x=0 or 1)
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TADD [7:0]								TR[7:0]							
w								w							

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

位/位域	名称	描述
31:24	TADD[7:0]	颜色查找表写地址 颜色查找表位于该地址的节点的值，将由写入的TR，TG和TB值更新。
23:16	TR[7:0]	LUT节点的红色值
15:8	TG[7:0]	LUT节点的绿色值
7:0	TB[7:0]	LUT节点的蓝色值

24. SDIO 接口（SDIO）

24.1. 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 AHB 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。

24.2. 主要特征

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。

24.3. 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。

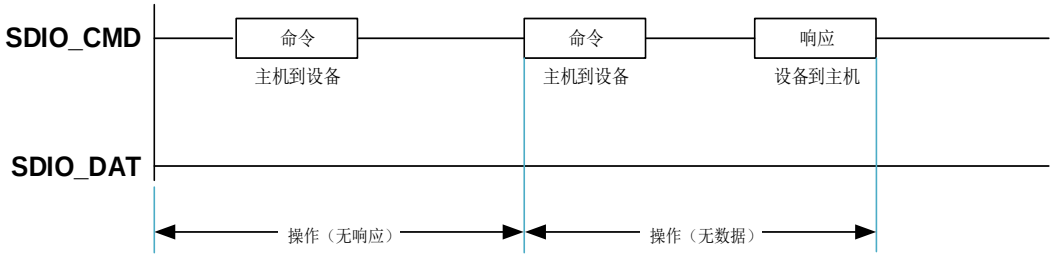
有几种不同类型的操作。一般操作总是包含一个命令和响应。此外，一些操作还有一个数据令牌。还有一些其他操作直接将他们的信息包含在命令或响应结构中。在这种情况下，操作没有数据令牌。在 DAT0-DAT7 和 CMD 信号线上的比特位根据主机时钟同步传输。

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 24-1. 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 24-1. 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 24-2. SDIO 多块读操作*所示，多个块的写操作如*图 24-3. 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 24-2. SDIO 多块读操作

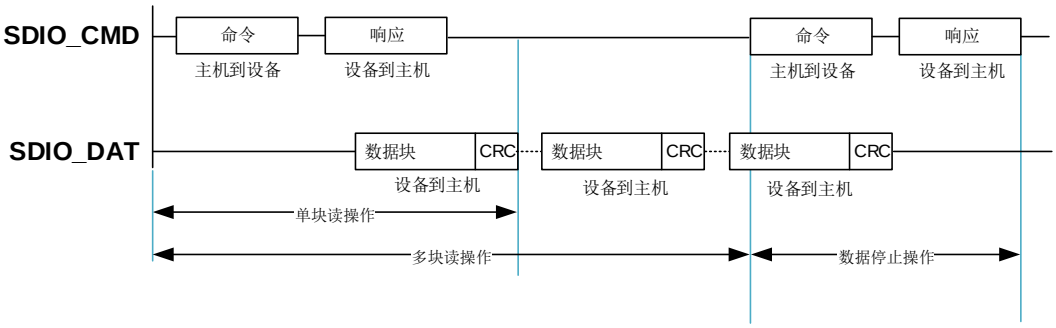
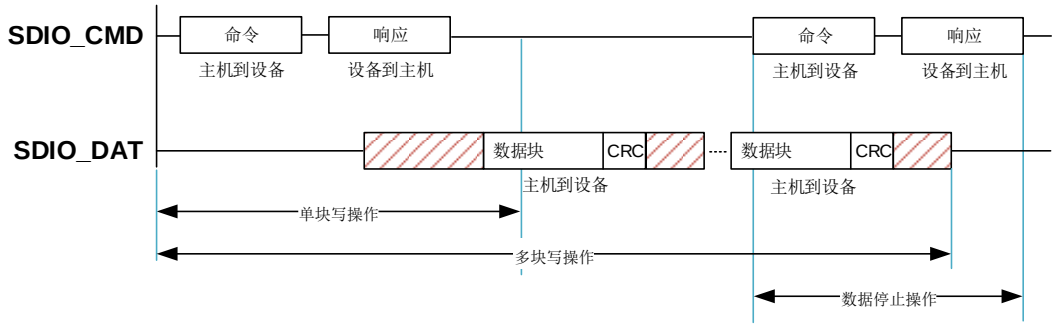


图 24-3. SDIO 多块写操作



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 24-4. SDIO 数据流读操作*和*图 24-5. SDIO 数据流写操作*分别是数据流的读和写操作。

图 24-4. SDIO 数据流读操作

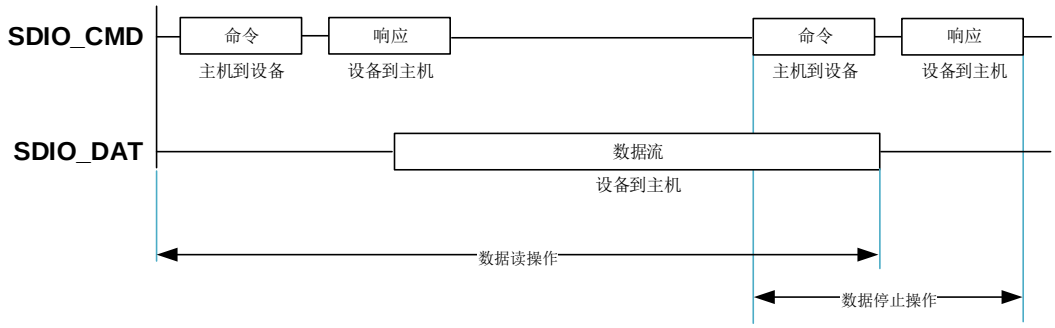
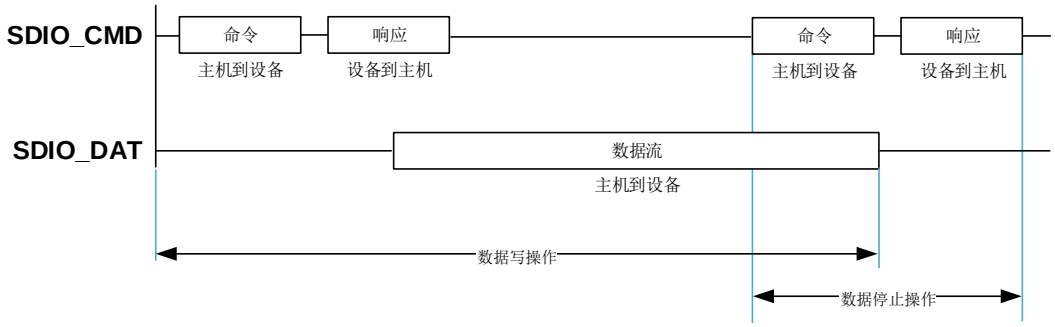


图 24-5. SDIO 数据流写操作

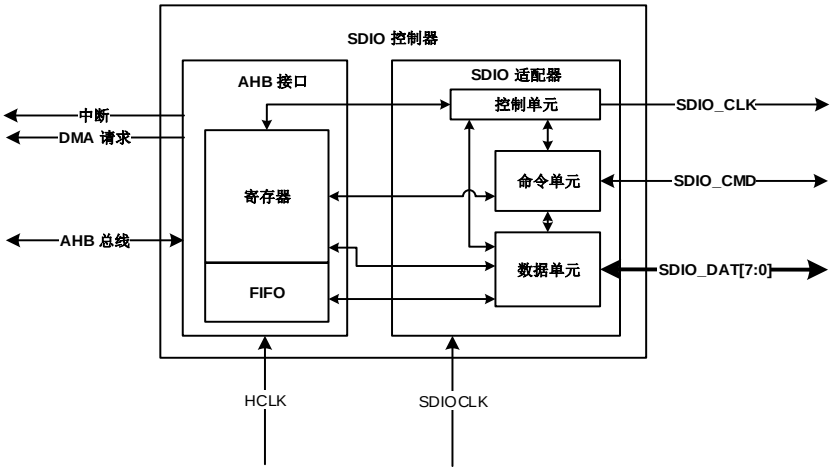


24.4. SDIO 功能描述

*图 24-6. 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；
- AHB 接口：包括通过 AHB 总线访问的寄存器、用于数据传输的 FIFO 单元以及产生中断和 DMA 请求信号。

图 24-6. SDIO 框图



24.4.1. 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 0MHz 到 20MHz 之间，对于 MMC 卡 V4.2 版本可以在 0 MHz 到 48MHz 之间，对于 SD 或 SD I/O 卡可以在 0MHz 到 25MHz。

SDIO 使用两个时钟信号：SDIO 适配器时钟(SDIOCLK = HCLK)和 AHB 总线时钟(HCLK)。

SDIO_CMD：该信号是双向命令通道，用于卡的初始化和命令的传输。命令从 SDIO 控制器发送到卡，响应从卡发送到主机。CMD 信号有两种操作模式：用于初始化的开漏模式（仅用于 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 24-1. SDIO I/O 定义

引脚功能	方向	描述
SDIO_CLK	O	SD/SD I/O /MMC 时钟
SDIO_CMD	I/O	命令的输入/输出
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 AHB 接口仍在工作。所以，FIFO 可以通过 AHB 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：数据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：数据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

24.4.2. AHB 接口

AHB 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）。
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）。

24.5. 卡功能描述

24.5.1. 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。

24.5.2. 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 24-7. 命令标记格式*所示，需要 1.92us（25MHz）、0.96us（50MHz）和 0.92us（52MHz）的发送时间。

图 24-7. 命令标记格式

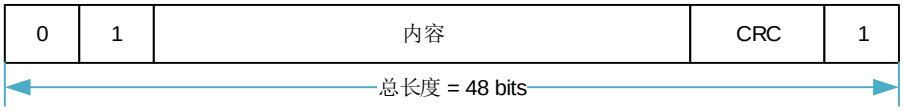


表 24-2. 命令格式

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 24-3. 卡命令类 (CCCs)*）。每类支持一组卡的功能。*表 24-3. 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，设置在每个卡的卡的特定数据（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 24-3. 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					+							

	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
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 24-4. 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。

命令索引	类型	参数	响应格式	简称	描述
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。
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 24-5. 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

命令索引	类型	参数	响应格式	简称	描述
					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 24-6. 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_STOP	从卡中读取数据流，起始于给定的地址，直至收到 STOP_TRANSMISSION 命令。
CMD20	adtc	[31:0]数据地址	R1	WRITE_DAT_UNTIL_STOP	从主机写数据流，起始于给定的地址，直至收到 STOP_TRANSMISSION 命令。
注意：传输的数据不能跨越物理块边界，除非 READ_BLK_MISALIGN 在 CSD 寄存器中被设置。					

表 24-7. 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 24-5. 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 24-8. 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)
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 24-9. 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 24-10. 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 24-5. 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 24-11. 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 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 24-12. 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。

命令索引	类型	参数	响应格式	简称	描述
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 [30:28] 功能数目 [27] 块模式 [26] OP 码 [25:9] 寄存器地址 [8:0] 字节/块数		IO_RW_EXTENDED	该命令允许用一个简单命令读取或写入大量的 I/ O 寄存器。
注意： 1.CMD39, CMD40 仅用于 MMC 卡 2. CMD52, CMD53 仅用于 SD I/O 卡					

表 24-13. 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）。

24.5.3. 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：

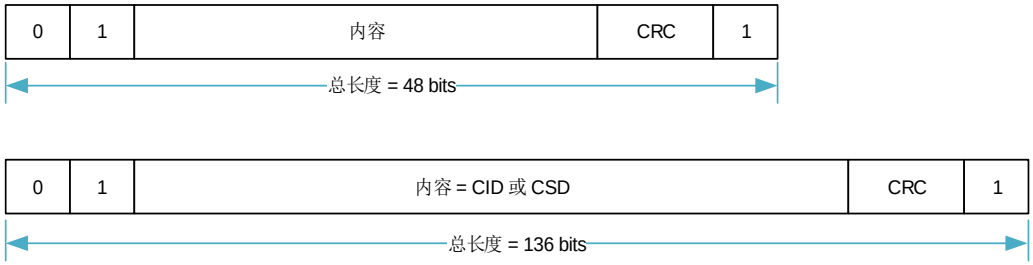
- **R1 / R1b**：普通命令响应
- **R2**：CID, CSD 寄存器
- **R3**：OCR 寄存器
- **R4**：Fast I/O
- **R5**：中断请求
- **R6**：发布的 RCA 响应
- **R7**：卡接口条件

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 24-8. 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 24-8. 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 24-14. 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 24-15. 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 24-16. 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 24-17. 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 24-18. 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 24-19. 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 24-20. 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 24-21. 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 24-22. 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

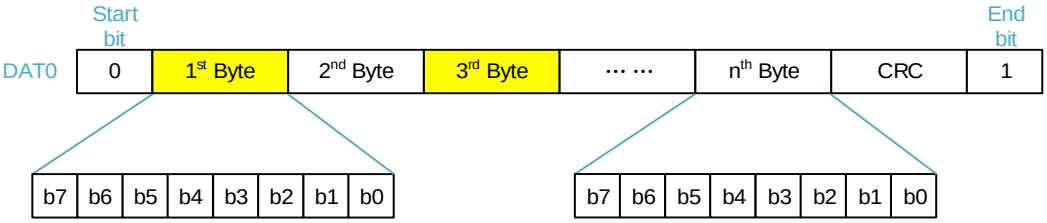
24.5.4. 数据包格式

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

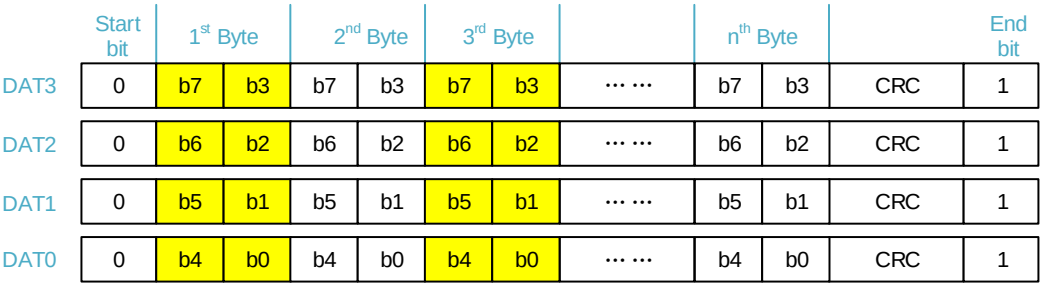
卡复位和初始化之后，只有 DAT0 被用于传输数据。其他引脚可以用于其他用处。*图 24-9. 1 位数据总线宽度*，*图 24-10. 4 位数据总线宽度*和*图 24-11. 8 位数据总线宽度*显示了数据宽度是 1 位，4 位和 8 位时的数据包格式。

图 24-9. 1 位数据总线宽度



4 位数据包格式

图 24-10. 4 位数据总线宽度



8 位数据包格式

图 24-11. 8 位数据总线宽度

	Start bit	1st Byte	2nd Byte	3rd Byte						nth Byte		End bit
DAT7	0	b7	b7	b7				… …		b7	CRC	1
DAT6	0	b6	b6	b6				… …		b6	CRC	1
DAT5	0	b5	b5	b5				… …		b5	CRC	1
DAT4	0	b4	b4	b4				… …		b4	CRC	1
DAT3	0	b7	b3	b7				… …		b3	CRC	1
DAT2	0	b6	b2	b6				… …		b2	CRC	1
DAT1	0	b5	b1	b5				… …		b1	CRC	1
DAT0	0	b4	b0	b4				… …		b0	CRC	1

24.5.5. 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（有命令延迟）。
•C：读可清除。

表 24-23. 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；	C

位	标识符	类型	数值	说明	清除条件
				被暂时或者永久写保护的卡被擦除。	
14	CARD_ECC_DISABLED	SX	’0’= 使能 ’1’= 失能	执行命令时未使用内部 ECC。	A
13	ERASE_RESET	SR	’0’= 清除 ’1’= 设置	因为收到一个擦除顺序之外的命令，擦除序列在执行前被清除。	C
[12:9]	CURRENT_STATE	SX	0 = 空闲 1 = 就绪 2 = 识别 3 = 待机 4 = 传输 5 = 发送数据 6 = 接收数据 7 = 编程 8 = 断开 9-14 = 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 状态寄存器大小是一个 512 比特数据块。该寄存器的内容连同一个 16 位 CRC 通过 DAT 总线被发送到主机上。SD 状态通过 DAT 总线被发送到主机上，作为 ACMD13 的响应（CMD55 接着用 CMD13）。ACMD13 只能在“传送状态”被发送到存储卡（卡被选中）。SD 状态结构将在下面描述。

“类型”和“清除条件”的缩写与上述卡状态描述相同。

表 24-24. SD 状态

位	标识符	类型	数值	描述	清除条件
[511:510]	DAT_BUS_WIDTH	SR	’00’= 1 (默认) ‘01’= 保留 ‘10’= 4 位宽 ‘11’= 保留	由 SET_BUS_WIDTH 命令显示当前定义的数据总线宽度	A
509	SECURED_MODE	SR	’0’= 未处于安全模式 ’1’= 处于安全模式	卡处于操作的安全模式（参考“SD 安全规范”）。	A
[508:496]	保留				
[495:480]	SD_CARD_TYPE	SR	下列卡目前被定义为： ‘0000’= 通用 SD 读/写卡 ‘0001’= SD ROM 卡 '0002'= OTP	低 8 位在未来被用来定义 SD 存储卡的不同变种（每个位将定义不同的 SD 卡类型）。高 8 位将被用来定义不符合当前 SD 物理层规范的 SD 卡。	A
[479:448]	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 24-25. 移动性能字段*中定义。

表 24-25. 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 24-26. 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

AU_SIZE	数值定义
9h	4 MB
Ah	8 MB
Bh	12 MB
Ch	16 MB
Dh	24 MB
Eh	32 MB
Fh	64 MB

最大 AU 大小，取决于卡的容量，由*表 24-26. AU_SIZE 字段*中定义。卡可以任意的设置 AU 大小（由*表 24-27. 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 24-27. 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 24-28. 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：

$$\text{Erase timeout of X AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad \text{(式 24-1)}$$

表 24-29. 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

ERASE_TIMEOUT	数值定义
02	2 秒
03	3 秒
.....	.....
63	63 秒

如果 ERASE_SIZE 字段被设置为 0，则该字段应该设置为 0。

ERASE_OFFSET

这 2 位字段表示 T_{OFFSET}，可以选择如*表 24-30. 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 24-30. 擦除偏移字段

ERASE_OFFSET	数值定义
0h	0 秒
1h	1 秒
2h	2 秒
3h	3 秒

24.6. 编程序列

24.6.1. 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：

- 检测卡是否连接；
- 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。
- 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。
- 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。

24.6.2. 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。

24.6.3. 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。

24.6.4. 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，中断传输和在数据状态等待停止命令。

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。

24.6.5. 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：

$$\max\ \text{write frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}} - 100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad \text{(式 24-2)}$$

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，所花费的时间来传送数据（由总线时钟速率的定义）必须

比它需要读出主存储器字段的时间下（由定义卡在 CSD 寄存器）。因此，最大的时钟频率为流读取操作由下式给出下面的公式：

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：

$$\max\ \text{read}\ \text{frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}} - 100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad \text{(式 24-3)}$$

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。

24.6.6. 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。

24.6.7. 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。

24.6.8. 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。

24.6.9. 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 的值不为 0），卡在每次上电后会自动上锁。

与存在的 CSD 寄存器写命令相同，上锁/解锁命令也只在卡的传输态有效。这意味着，上锁/解锁命令不包含地址参数，且必须在使用该命令前卡必须被选中。

卡上锁/解锁命令与卡单块写命令有着相同的结构和总线事务类型。传输的数据块包含命令所有需要的信息（密码设置模式，密码本身，卡上锁/解锁等）。*表 24-31. 上锁/解锁数据结构*为上锁/解锁命令的结构。

表 24-31. 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

1. 如果卡之前未被选中，使用 CMD7 选中卡；
2. 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；
3. 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；
4. 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

1. 如果卡之前未被选中，使用 CMD7 选中卡；

2. 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；
3. 在数据线上，以合适的数据块大小发送卡上锁/解锁命令，包含 16 位的 CRC。数据块指示模式（CLR_PWD），密码长度（PWDS_LEN）和密码本身。如果 PWD 和 PWD_LEN 的内容与发送的密码和其大小匹配，PWD 寄存器的内容会被清除，同时 PWD_LEN 被设为 0。如果密码不正确，状态寄存器中的 LOCK_UNLOCK_FAILED 会被置位。

卡上锁

1. 如果卡之前未被选中，使用 CMD7 选中卡；
2. 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；
3. 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

1. 如果卡之前未被选中，使用 CMD7 选中卡；
2. 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；
3. 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。

24.7. 特定操作

24.7.1. 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 24-12. 通过停止 SDIO_CLK 的读等待操作*和*图 24-13. 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 24-12. 通过停止 SDIO_CLK 的读等待操作

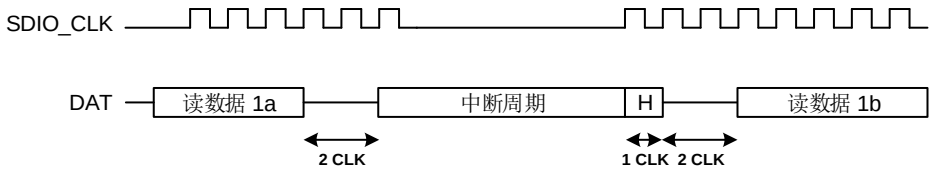
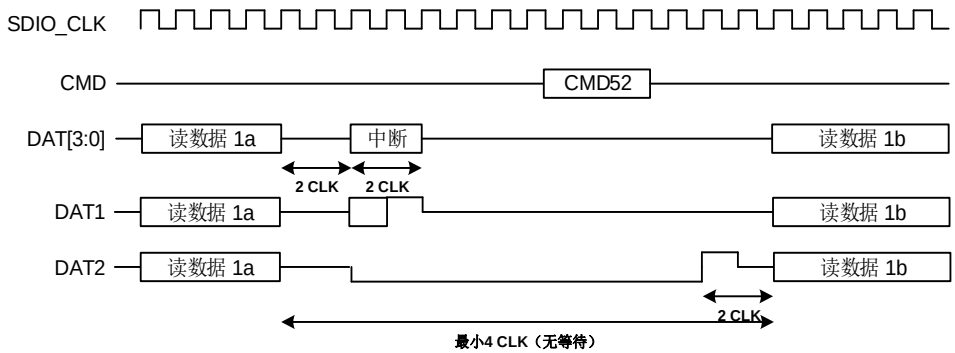


图 24-13. 使用 SDIO_DAT[2]信号线的读等待操作



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。

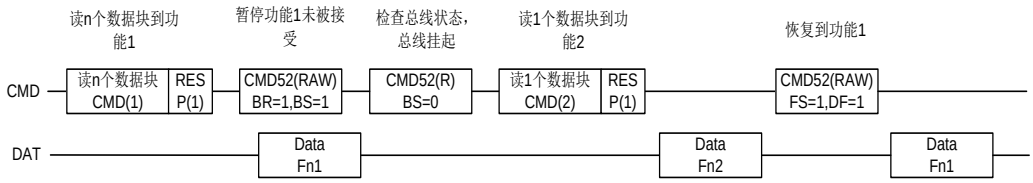
SD I/O 暂停/恢复操作

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 24-14. 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 24-14. 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 24-15. 读中断周期时序*显示单个数据读周期的中断周期时序。

图 24-15. 读中断周期时序

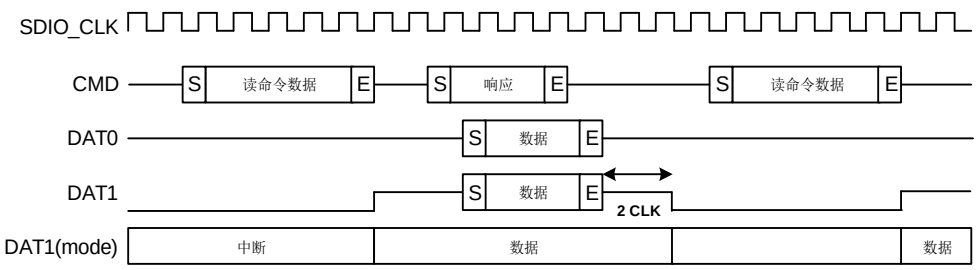
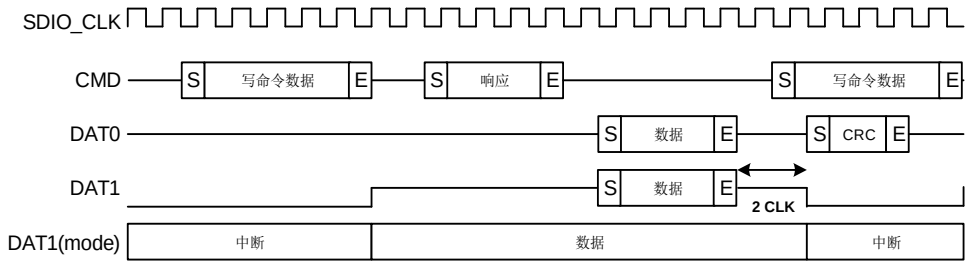


图 24-16. 写中断周期时序



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 24-17. 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 24-18. 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 24-17. 4 位模式下多块读中断周期时序

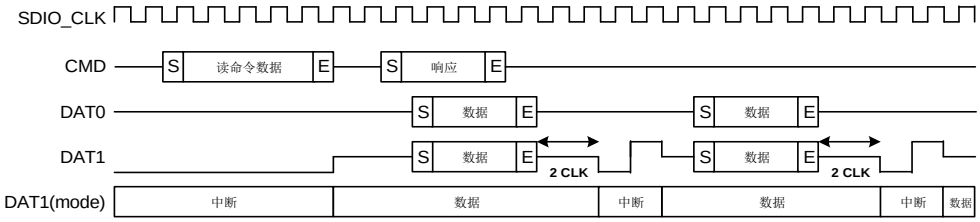
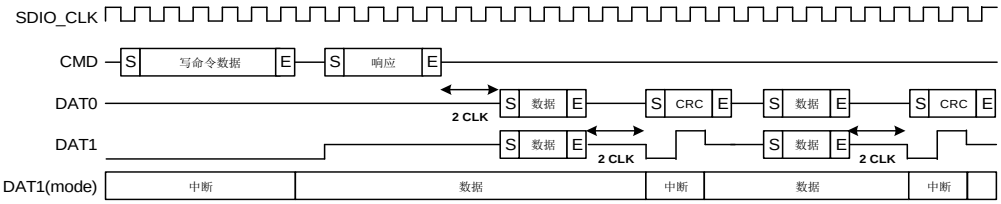


图 24-18. 4 位模式下多块写中断周期时序



24.7.2. 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）。

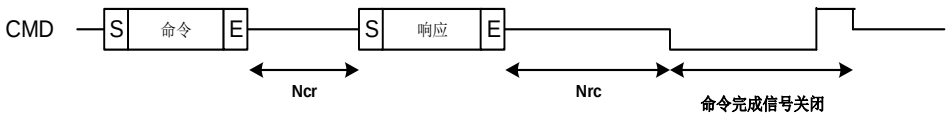
在主机从设备检测到一个命令完成信号之后，应该发送 FAST_IO（CMD39）命令来读取 ATA 状态寄存器以确定 ATA 命令的结束状态。

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 24-19. 命令完成信号关闭操作



24.8. SDIO 寄存器

SDIO 基地址：0x4001 8000

24.8.1. 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。

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

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

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。

24.8.2. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV[8]	保留														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	HWCLKEN	CLKEDGE	BUSMODE[1:0]		CLKBYP	CLKPWRSAV	CLKEN	DIV[7:0]							
	rw	rw	rw		rw	rw	rw	rw							

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。

24.8.3. 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。

24.8.4. 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。

24.8.5. 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。

24.8.6. 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 24-32. 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 24-32. 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

24.8.7. 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，数据超时寄存器和数据长度寄存器应在写数据控制寄存器前更新。

24.8.8. 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，数据超时寄存器和数据长度寄存器应在写数据控制寄存器前更新。

24.8.9. 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	TRANS MOD	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。

24.8.10. 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。

24.8.11. 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）

24.8.12. 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	RXOREC	TXUREC	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。

24.8.13. 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 写 1 使能中断。
22	SDIOINTIE	SD I/O 中断已接收中断使能 写 1 使能中断。
21	RXDTVALIE	接收 FIFO 中的数据有效中断使能 写 1 使能中断。
20	TXDTVALIE	发送 FIFO 中的数据有效中断使能

		写 1 使能中断。
19	RFEIE	接收 FIFO 空中断使能 写 1 使能中断。
18	TFEIE	发送 FIFO 空中断使能 写 1 使能中断。
17	RFFIE	接收 FIFO 满中断使能 写 1 使能中断。
16	TFFIE	发送 FIFO 满中断使能 写 1 使能中断。
15	RFHIE	接收 FIFO 半满中断使能 写 1 使能中断。
14	TFHIE	发送 FIFO 半满中断使能 写 1 使能中断。
13	RXRUNIE	正在接收数据中断使能 写 1 使能中断。
12	TXRUNIE	正在传输数据中断使能 写 1 使能中断。
11	CMDRUNIE	正在传输命令中断使能 写 1 使能中断。
10	DTBLKENDIE	数据块已发送/已接收中断使能 写 1 使能中断。
9	STBITEIE	起始位错误中断使能 写 1 使能中断。
8	DTENDIE	数据结束中断使能 写 1 使能中断。
7	CMDSENDIE	命令已发送中断使能 写 1 使能中断。
6	CMDRECVIE	命令响应已接收中断使能 写 1 使能中断。
5	RXOREIE	接收 FIFO 上溢错误中断使能 写 1 使能中断。
4	TXUREIE	发送 FIFO 下溢错误中断使能 写 1 使能中断。
3	DTTMOUTIE	数据超时中断使能 写 1 使能中断。

2	CMDTMOUTIE	命令响应超时中断使能 写 1 使能中断。
1	DTCRCERRIE	数据 CRC 错误中断使能 写 1 使能中断。
0	CCRCERRIE	命令响应 CRC 错误中断使能 写 1 使能中断。

24.8.14. 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。

24.8.15. 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。

25. 外部存储器控制器（EXMC）

25.1. 简介

外部存储器控制器EXMC，用来访问各种片外存储器，通过配置寄存器，EXMC可以把AMBA协议转换为专用的片外存储器通信协议，包括SRAM，ROM，NOR Flash，PSRAM，NAND Flash，PC Card和SDRAM。用户还可以调整相关的时间参数来提高通信效率。EXMC模块划分为许多个子Bank，每个Bank支持特定的存储器类型，用户可以通过配置Bank的寄存器来控制外部存储器。

25.2. 主要特征

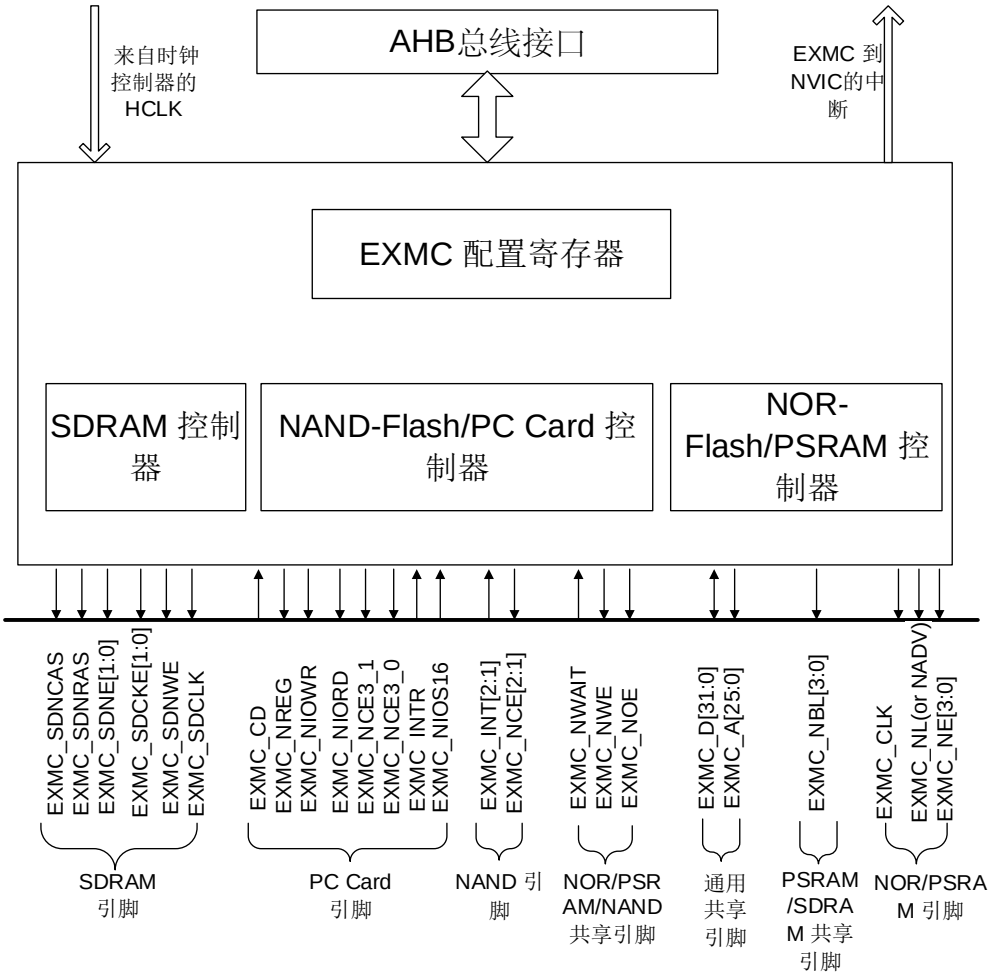
- 支持片外存储器类型：
  - SRAM
  - PSRAM/SQPI-PSRAM
  - ROM
  - NOR Flash
  - 8 位或 16 位 NAND Flash
  - 16 位 PC Card
  - SDRAM
- AMBA 协议与各种片外存储器协议转换。
- 可编程的时序参数，可以满足用户特定需求。
- 每个 Bank 有独立的片选信号。
- 对于部分存储器类型支持独立的读写时序。
- 对于 NAND Flash 内置硬件 ECC。
- 支持 8 位，16 位，32 位总线带宽。
- NOR Flash 和 PSRAM 支持地址总线和数据总线的复用。
- 提供写使能和字节选择信号。
- 当 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。

25.3. 功能说明

25.3.1. 结构框图

EXMC由6个模块组成：AHB总线接口，EXMC配置寄存器，NOR/PSRAM控制器，NAND/PC Card控制器，SDRAM控制器和外部设备接口。AHB时钟（HCLK）是参考时钟。

图 25-1. 系统架构



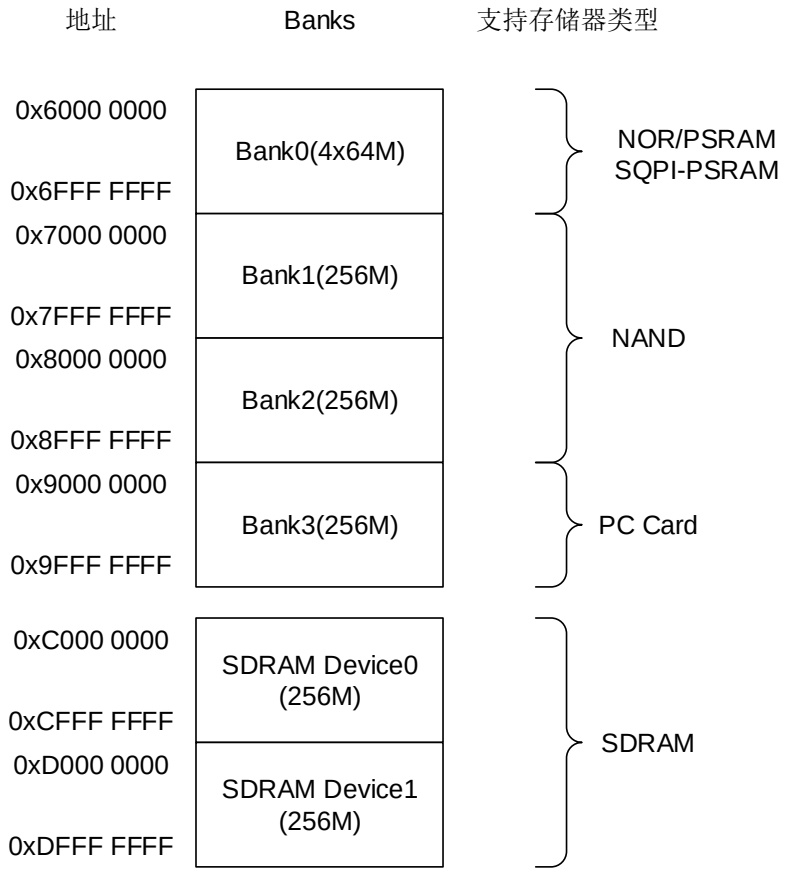
25.3.2. EXMC 访问基本规范

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、SDRAM，则可通过它的字节通道 NBL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

25.3.3. 外部设备地址映射

图 25-2. EXMC Bank 划分



EXMC将外部存储器分成多个Bank，每个Bank占256M字节，其中Bank0又分为4个Region，每个Region占64M字节。Bank1和Bank2又都被分成2个Section，分别是属性存储空间和通用存储空间。Bank3分成3个Section，分别是属性存储空间，通用存储空间和I/O存储空间。

每个Bank和Region都有独立的片选控制信号，也都能进行独立的配置。

Bank0用于访问NOR、PSRAM设备。

Bank1和Bank2用于连接NAND Flash，且每个Bank连接一个NAND。

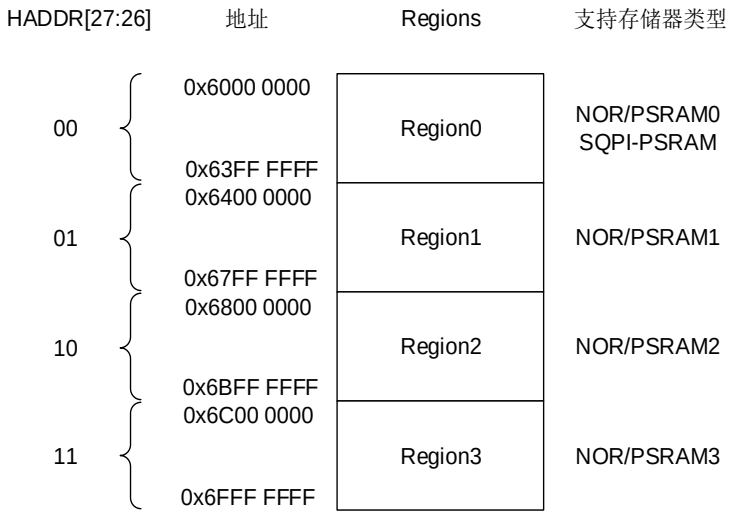
Bank3用于连接PC卡。

SDRAM Device0和SDRAM Device1用于连接SDRAM。

NOR 和 PSRAM 的地址映射

*图 25-3. 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 25-3. Bank0 地址映射



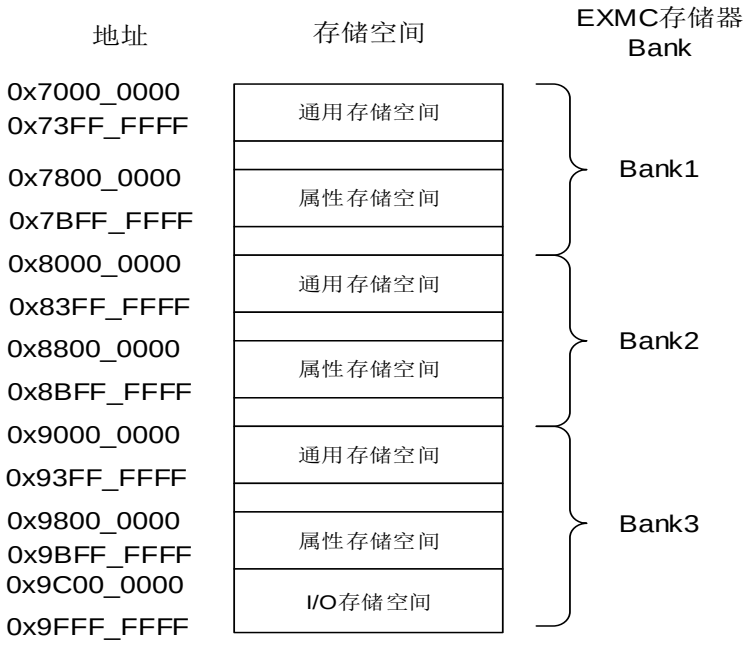
由于HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但EXMC能实现对HADDR的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是 8 位按字节对齐，EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；
- 如果外部存储器的数据宽度是 16 位按半字对齐，就需要将 HADDR 的字节地址转化为半字地址之后再连接外存储器。EXMC 内部将 HADDR[25:1]与 EXMC_A[24:0]相连，然后 EXMC_A[24:0]与外部存储器的地址线相连；
- 如果外部存储器的数据宽度是 32 位按字对齐，就需要将 HADDR 的字节地址转化为字地址之后再连接外存储器。EXMC 内部将 HADDR[25:2]与 EXMC_A[23:0]相连，然后 EXMC_A[23:0]与外部存储器的地址线相连。

NAND/PC Card 地址映射

Bank1 和 Bank2 用来访问 NAND Flash，Bank3 用来访问 PC Card。每个 Bank 如*图 25-4. NAND/PC Card 地址映射*被分为多个存储空间。

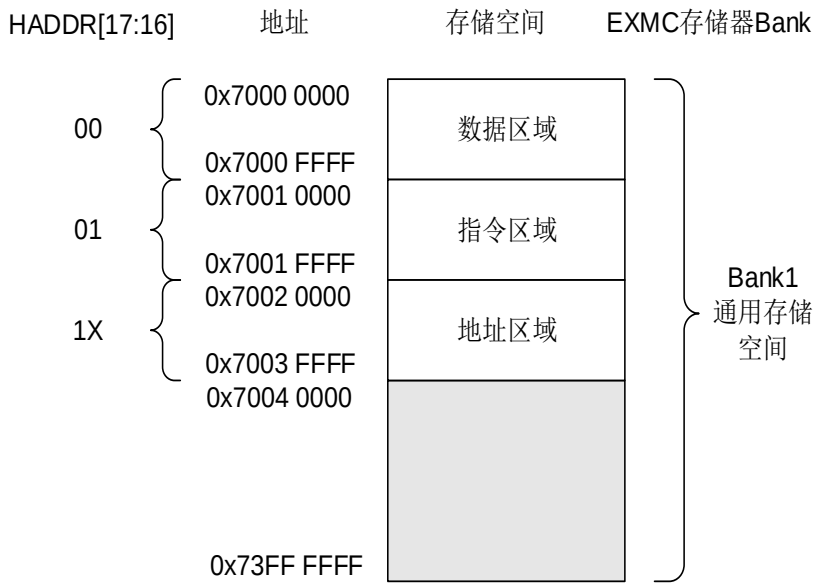
图 25-4. NAND/PC Card 地址映射



NAND 地址映射

对于 NAND FLASH，通用和属性空间又可以细划分为 3 个区域。*图 25-5. 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 25-5. 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：

地址区：指定操作NAND FLASH的地址，软件在地址区写入地址。在地址传输过程中，EXMC会使能地址锁存信号（ALE），ALE映射到EXMC_A[17]。

指令区：指定NAND FLASH将要执行的指令，软件在命令区写入指令。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]。

数据区：NAND FLASH读写数据，软件在数据区读出或写入数据。当EXMC在数据发送模式，软件需要在数据区写入数据，当EXMC在数据接收模式，软件需要在数据区读取数据。由于NAND FLASH会自动累加其内部操作地址，故在读写时不需要软件修改操作地址。

SDRAM 地址映射

HADDR[28]位用来选两个 SDRAM Bank，如*图 25-6. SDRAM 地址映射*所示。

图 25-6. SDRAM 地址映射

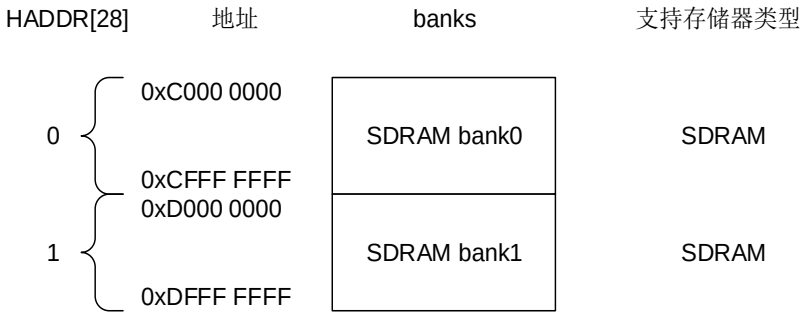


*表 25-1. SDRAM 地址映射*展示了 SDRAM 的 13 位行地址和 11 位列地址的配置映射：

表 25-1. SDRAM 地址映射

存储器数据宽度	内部bank	行地址	列地址	最大存储容量
8-bit	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 Mbytes: 4 x 8K x 2K
16-bit	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 Mbytes: 4 x 8K x 2K x 2
32-bit	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256 Mbytes: 4 x 8K x 2K x 4

25.3.4. NOR/PSRAM 控制器

EXMC模块的NOR/PSRAM控制器控制Bank0，它可以支持NOR Flash、PSRAM、SRAM、ROM和CRAM外部存储器。EXMC对Bank0每个Region输出一个唯一的片选信号，NE[x]（x=0..3），用于在4个Region中进行片选，所有其他的信号都是共享的。每个Region都有专门的寄存器控制。但是只有Region0支持访问SQPI-PSRAM存储器。

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 25-2. NOR Flash 接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
非复用 EXMC_A[25:0]	输出	异步/同步	地址总线
复用 EXMC_A[25:16]			
EXMC_D[15:0]	输入/输出	异步/同步（复用）	地址/数据总线
	输入/输出	异步/同步（非复用）	数据总线
EXMC_NE[x]	输出	异步/同步	片选，x=0/1/2/3
EXMC_NOE	输出	异步/同步	读使能
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL（NADV）	输出	异步/同步	地址有效

表 25-3. PSRAM 非复用接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
EXMC_A[25:0]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步	数据总线
EXMC_NE[x]	输出	异步/同步	片选，x=0/1/2/3
EXMC_NOE	输出	异步/同步	读使能
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL（NADV）	输出	异步/同步	地址锁存信号
EXMC_NBL[1]	输出	异步/同步	高字节使能
EXMC_NBL[0]	输出	异步/同步	低字节使能

表 25-4. SQPI-PSRAM 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	时钟信号
EXMC_NE[0]	输出	同步	片选，低有效
EXMC_D[0]	输入/输出	同步	数据，命令信号
EXMC_D[1]	输入/输出	同步	数据信号（SPI/SQPI/QPI 模式）
EXMC_D[3:2]	输入/输出	同步	数据信号（SQPI/QPI 模式）

支持的存储器访问模式

*表 25-5. EXMC 对 NOR，PSRAM 和 SRAM 支持的访问模式*列出了 EXMC 对 NOR，PSRAM 和 SRAM 支持的访问模式。

表 25-5. EXMC 对 NOR，PSRAM 和 SRAM 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
	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	
SRAM and ROM	异步	R	8	8	
	异步	R	8	16	
	异步	R	16	8	分成 2 次 EXMC 访问
	异步	R	16	16	
	异步	R	32	8	分成 4 次 EXMC 访问
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	8	8	
	异步	W	8	16	使用字节信号 NBL[1:0]
	异步	W	16	8	
	异步	W	16	16	
	异步	W	32	8	
	异步	W	32	16	

NOR Flash/PSRAM 控制时序

EXMC为SRAM、ROM、PSRAM、NOR Flash等外部静态存储器提供可编程的时序参数以及多种时序模型以满足不同的需求。

表 25-6. NOR/PSRAM 控制时序参数

参数	功能	访问模式	单位	最小值	最大值
CKDIV	同步时钟分频比	同步	HCLK	2	16
DLAT	数据延迟	同步	EXMC_CLK	2	17
BUSLAT	总线延迟	异步/同步读	HCLK	1	16
DSET	数据建立时间	异步	HCLK	2	256
AHLD	地址保持时间	异步（复用）	HCLK	2	16
ASET	地址建立时间	异步	HCLK	1	16

表 25-7. EXMC 时序模型

时序模型		扩展模式	模式描述	写时序参数	读时序参数
异步	模式 1	0	SRAM/PSRAM/CRAM	DSET ASET	DSET ASET
	模式 2	0	NOR Flash	DSET ASET	DSET ASET
	模式 A	1	SRAM/PSRAM/CRAM 在数据阶段 EXMC_OE 翻转	WDSET WASET	DSET ASET
	模式 B	1	NOR Flash	WDSET WASET	DSET ASET
	模式 C	1	NOR Flash 在数据阶段 EXMC_OE 翻转	WDSET WASET	DSET ASET
	模式 D	1	有地址保持功能	WDSET WAHLD WASET	DSET AHLD ASET
	模式 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 E	0	NOR/PSRAM/CRAM 同步读 PSRAM/CRAM 同步写	DLAT CKDIV	DLAT CKDIV
	模式 SM	0	NOR Flash 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如*表 25-7. EXMC 时序模型*所示，EXMC 模块 NOR Flash/PSRAM 控制器可以提供多种时序模型。用户可以通过修改*表 25-6. NOR/PSRAM 控制时序参数*中列出的参数来使之适合不同类型外部存储器的时序以及满足用户的要求。当将寄存器 EXMC_SNCTLx 位 EXMODEN 置 1 使能扩展模式后，可以通过寄存器 EXMC_SNTCFGx 和 EXMC_SNWTCFGx 将读写配置成独立的时序。

异步访问时序

模式1 – SRAM/CRAM

图 25-7. 模式 1 读时序

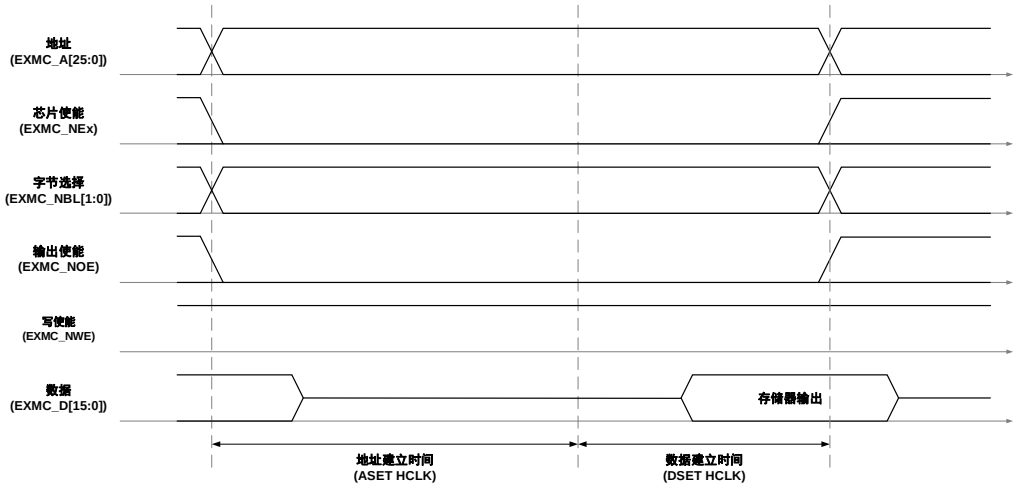


图 25-8. 模式 1 写时序

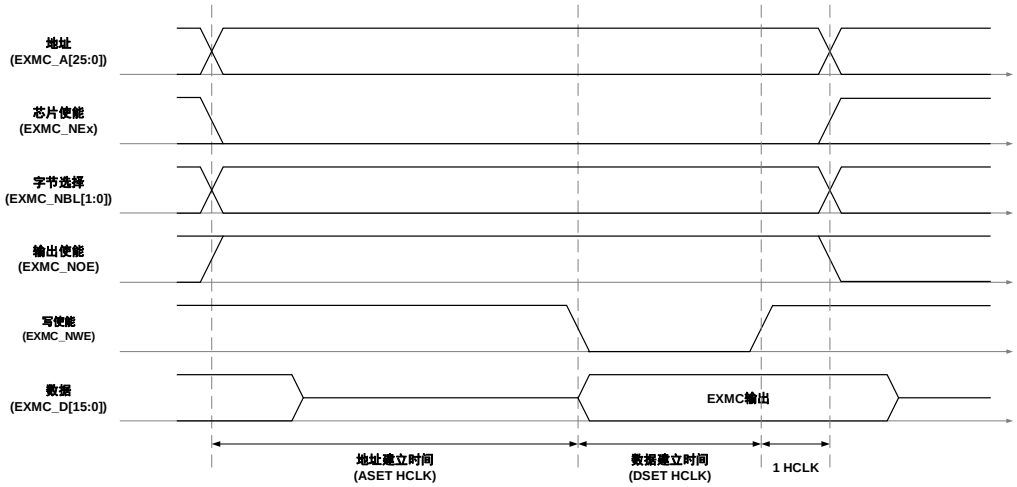


表 25-8. 模式 1 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWAIT	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响

位域/位	名称	参考设定值
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，除了 2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
29-28	ASYNCMOD	无影响
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+1 HCLK 时钟周期，读操作为 DSET HCLK 时钟周期）
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户

模式A – SRAM/PSRAM（CRAM）OE翻转

图 25-9. 模式 A 读时序

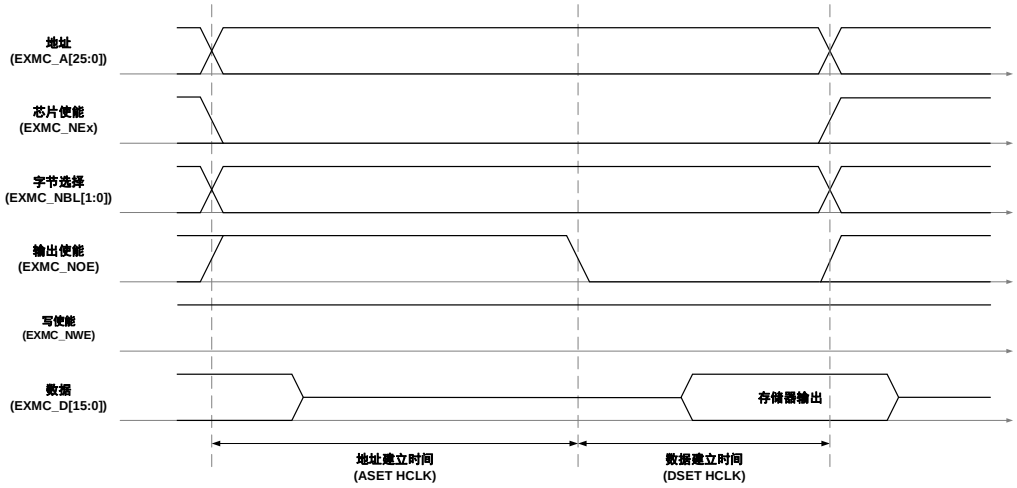
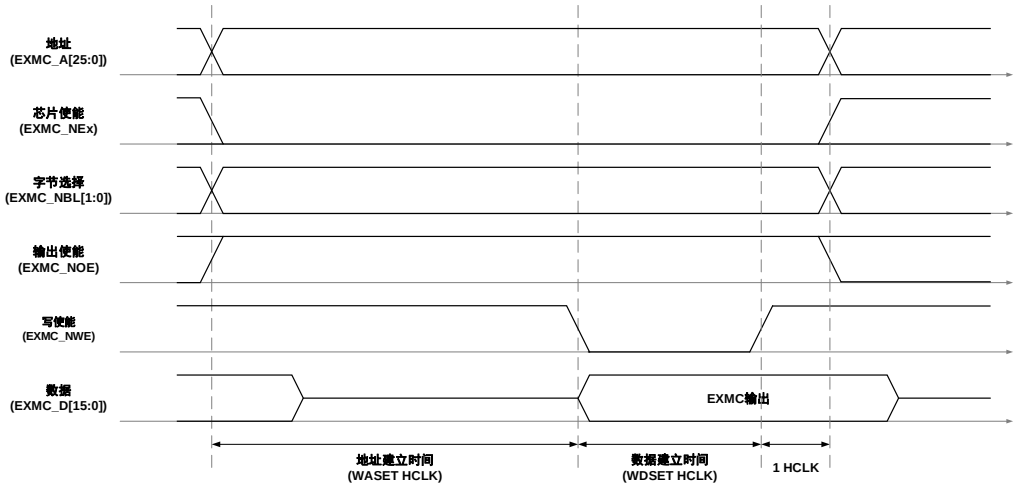


图 25-10. 模式 A 写时序



模式A和模式1的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式A的写时序独立于读时序。

表 25-9. 模式 A 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，除了 2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（Read）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+1 HCLK 时钟周期，读操作为 DSET HCLK 时钟周期）
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（Write）		
31-30	保留	0x0
29-28	WASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	保留	0x0
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式2/B – NOR Flash

图 25-11. 模式 2/B 读时序

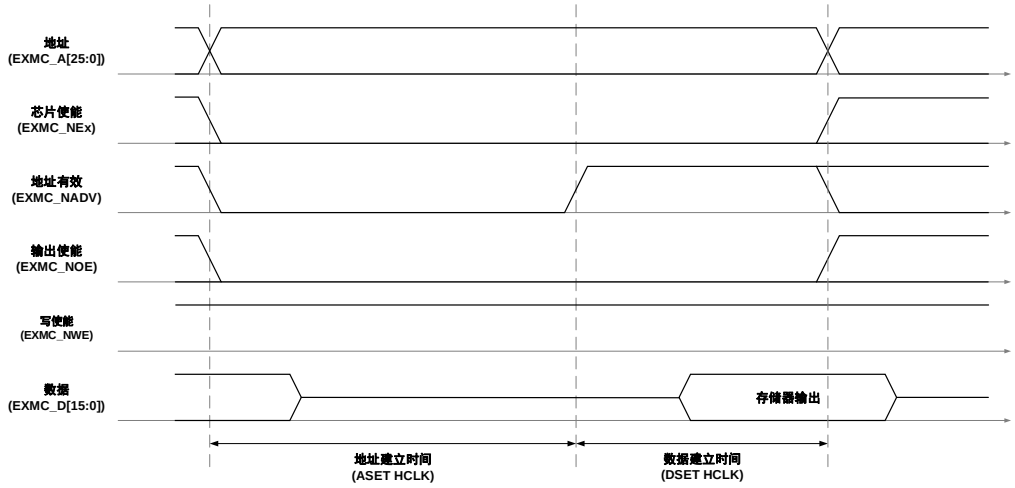


图 25-12. 模式 2 写时序

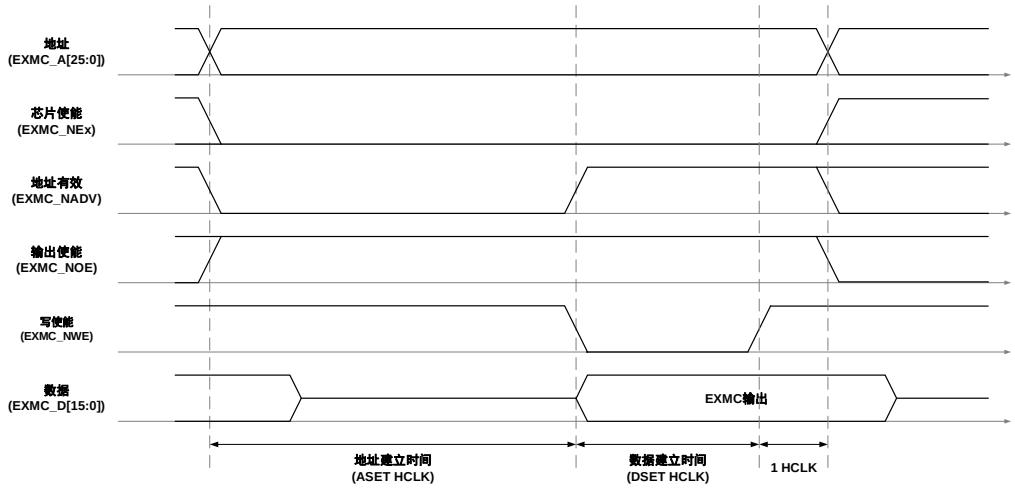


图 25-13. 模式 B 写时序

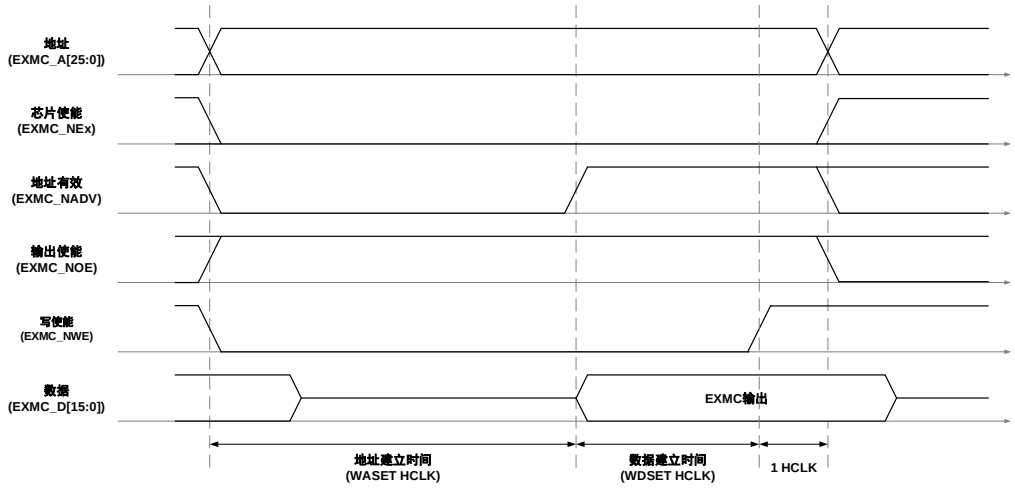


表 25-10. 模式 2/B 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx（模式 2，模式 B）		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	模式 2：0x0，模式 B：0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx（模式 2 读/写操作，模式 B 读操作）		
31-30	保留	0x0000
29-28	ASYNCMOD	模式 B：0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx（模式 B 写操作）		
31-30	保留	0x0000
29-28	WASYNCMOD	模式 B：0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	保留	0x000
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式C – NOR Flash OE翻转

图 25-14. 模式 C 读时序

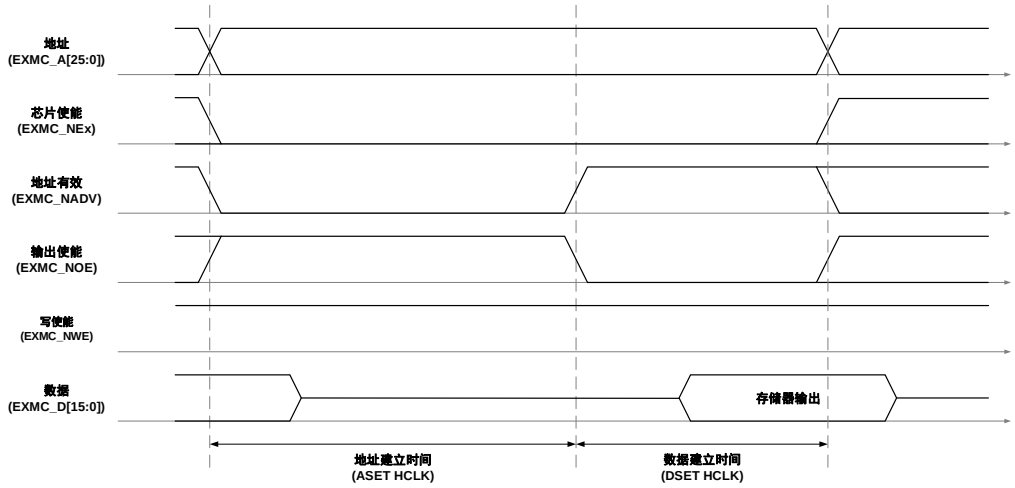
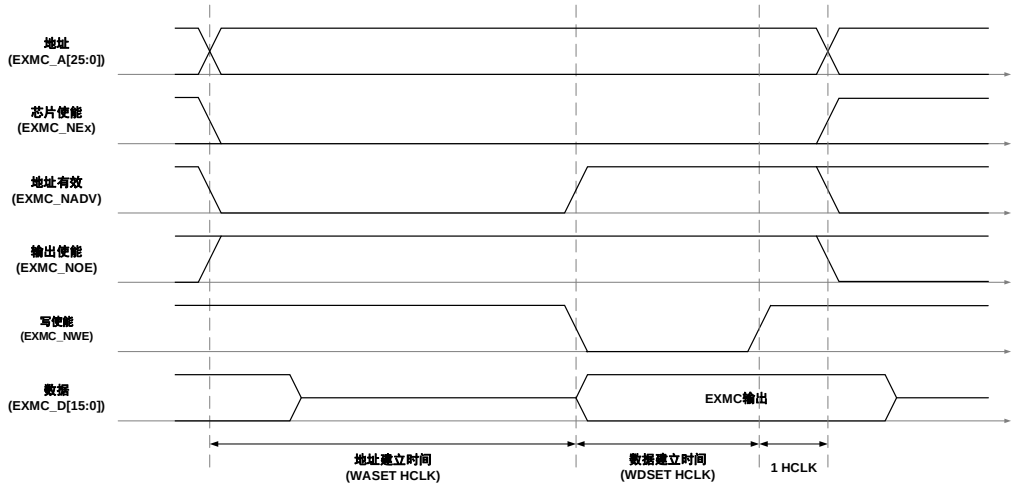


图 25-15. 模式 C 写时序



模式C和模式1的区别在于写时序，当两个模式的寄存器有相同的时序配置时，模式C的写时序独立于读时序。

表 25-11. 模式 C 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0

位域/位	名称	参考设定值
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	Nor Flash：2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0000
29-28	ASYNCMOD	模式 C：0x2
27-24	DLAT	0x0
23-20	CKDIV	0x0
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 C：0x2
27-24	DLAT	0x0
23-20	CKDIV	0x0
19-16	保留	0x0
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式D – 带地址扩展的异步操作

图 25-16. 模式 D 读时序

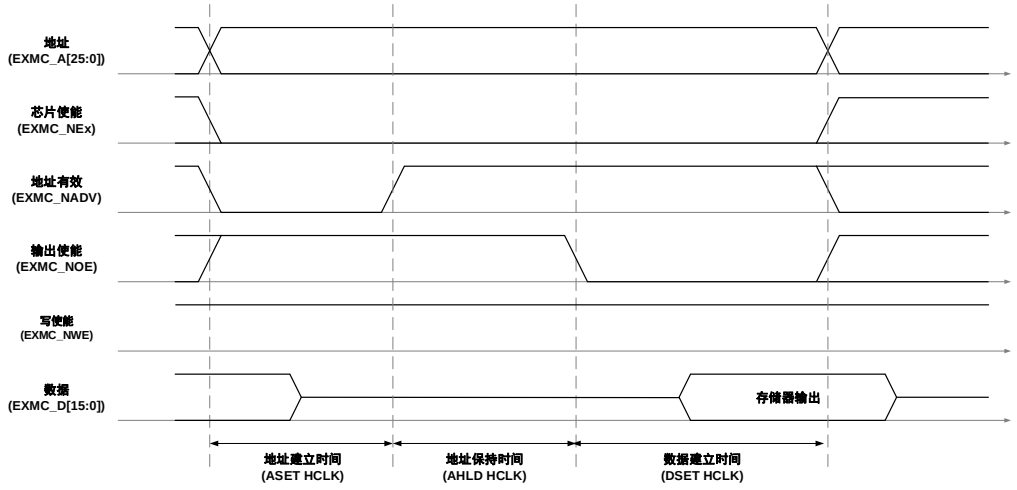


图 25-17. 模式 D 写时序

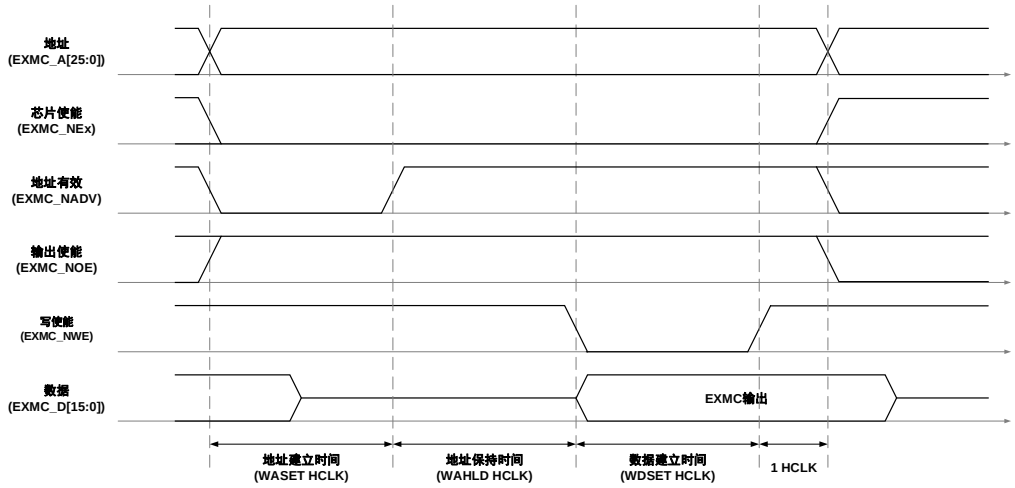


表 25-12. 模式 D 相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	模式 D：0x3
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

位域/位	名称	参考设定值
EXMC_SNWTCFGx		
31-30	保留	0x0
29-28	WASYNCMOD	模式 D：0x3
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	保留	0x0
15-8	WDSET	取决于存储器与用户
7-4	WAHLD	取决于存储器与用户
3-0	WASET	取决于存储器与用户

模式M – NOR Flash地址/数据总线复用

图 25-18. 复用模式读时序

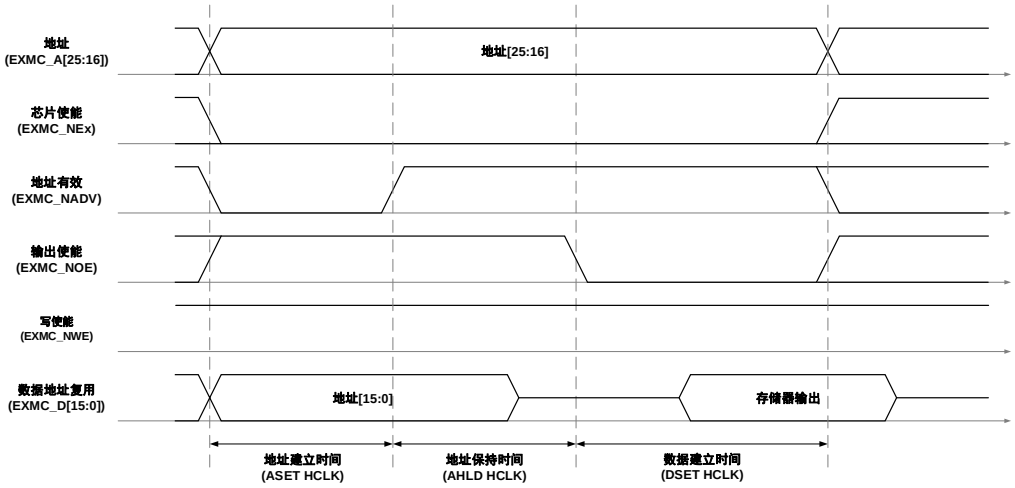


图 25-19. 复用模式写时序

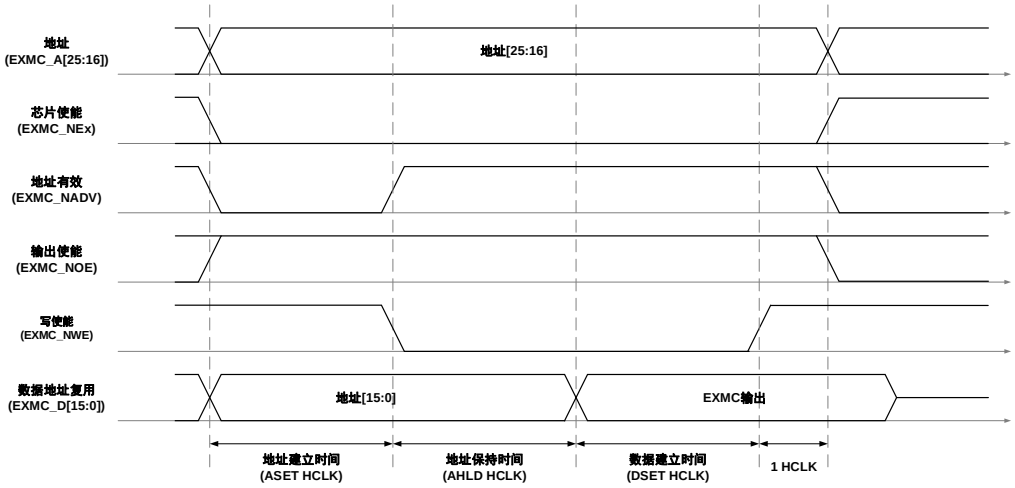


表 25-13. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000

位域/位	名称	参考设定值
19	SYNCWR	0x0
18-16	保留	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WREN	取决于存储器
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	0x2：NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFGx		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

异步通信的等待时间：

等待功能由寄存器EXMC_SNCTLx位ASYNCWAIT控制。在访问外部存储器期间，若使能异步等待功能（ASYNCWAIT=1），数据建立时间将会自动延长。延长时间的计算如下：

1. 若存储器等待信号与 NOE/NWE 信号对齐：

$$T_{DATA_SETUP} \geq maxT_{WAIT_ASSERTION} + 4HCLK \quad (25\text{-}1)$$

2. 若存储器等待信号与 NE 对齐：

如果

$$maxT_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (25\text{-}2)$$

$$T_{DATA_SETUP} \geq (maxT_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (25\text{-}3)$$

否则

$$T_{DATA_SETUP} \geq 4HCLK \quad (25\text{-}4)$$

图 25-20. 异步等待有效时的读时序

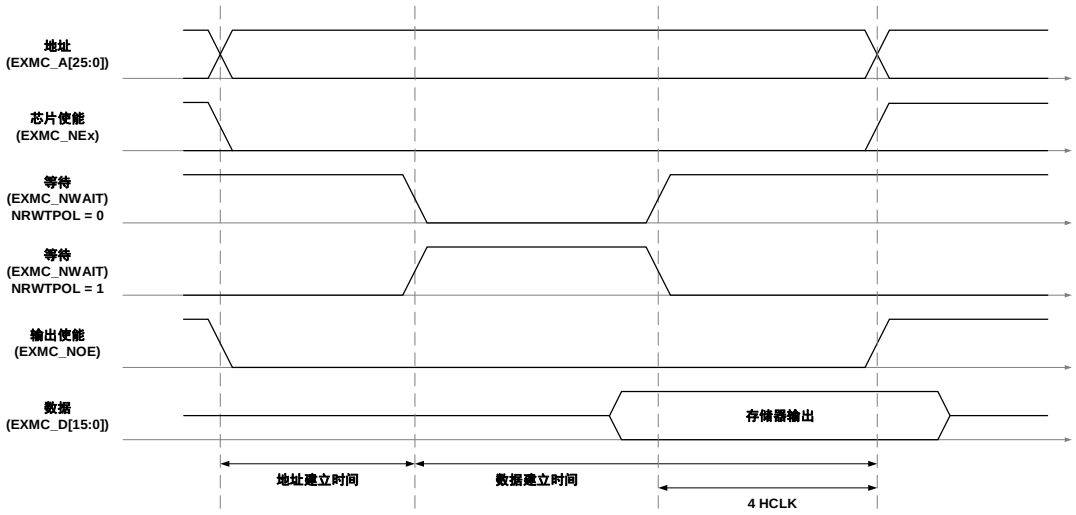
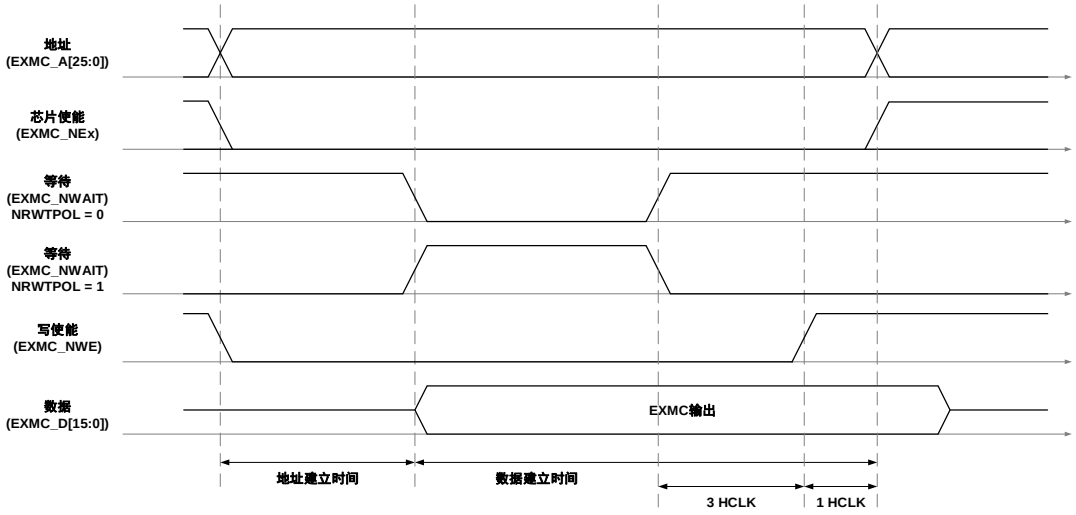


图 25-21. 异步等待有效时的写时序



同步访问时序

同步访问模式中，存储器时钟（EXMC_CLK）与系统时钟（HCLK）关系如下：

$$EXMC_CLK=\frac{HCLK}{CKDIV+1} \quad (25\text{-}5)$$

其中CKDIV是同步时钟分频比，通过配置寄存器EXMC_SNTCFGx中的CKDIV位来设置不同的值。

■ 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：

$$NOR\ 闪存延迟=DLAT+2 \quad (25\text{-}6)$$

NOR 闪存延迟包含 NADV，二者之间的关系为：

NOR 闪存延迟=DLAT+3 （25-7）

■ 数据等待

用户需要保证 EXMC_NWAIT 信号与外部设备一致。该信号通过寄存器 EXMC_SNCTLx 来设置， NRWTEN 位使能， NRWTCFG 位决定 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， NRWTPOL 位设置 EXMC_NWAIT 信号极性。

在 NOR Flash 的同步突发模式中，当寄存器 EXMC_SNCTLx 位 NRWTEN 置 1，在数据延迟之后后检测到 EXMC_NWAIT 信号。如果检测到 EXMC_NWAIT 有效，就会插入等待时钟，直到 EXMC_NWAIT 变为无效。

■ 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。

■ 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。

同步复用突发读时序 – NOR，PSRAM（CRAM）

图 25-22. 同步复用突发传输读时序

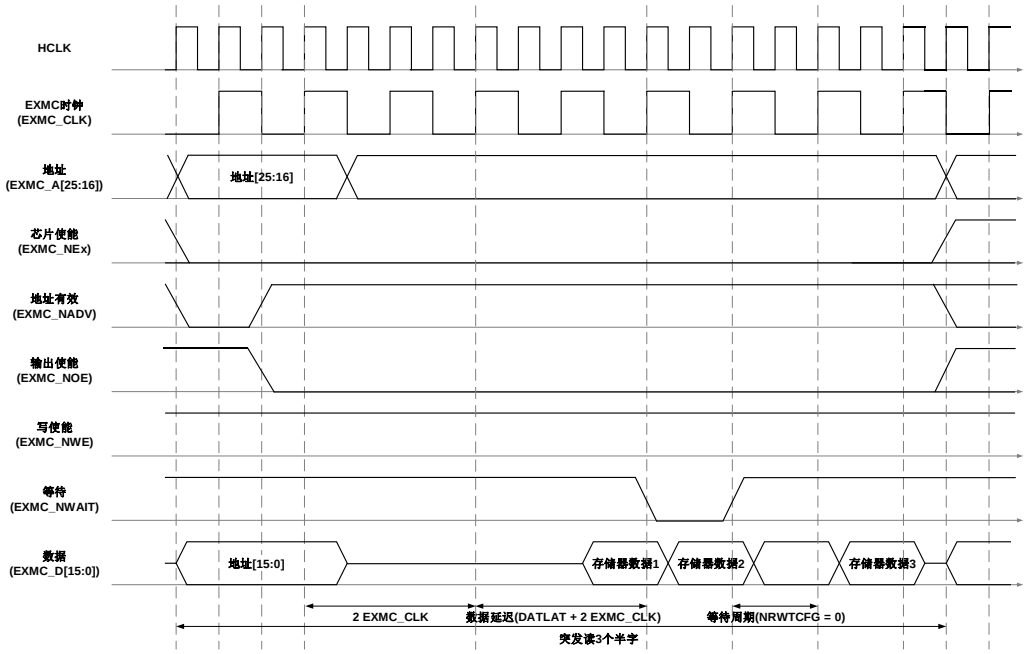


表 25-14. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
位域/位	名称	参考设定值
31-20	保留	0x000
19	SYNCWR	无影响
18-16	保留	0x0
15	ASYNCWTEN	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	WREN	无影响
11	NRWTCFG	取决于存储器
10	WRAPEN	0x0
9	NRWTPOL	取决于存储器
8	SBRSTEN	0x1，突发读使能
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	取决于存储器，0x1/0x2
1	NRMUX	0x1，取决于存储器与用户
0	NRBKEN	0x1
EXMC_SNTCFGx（Read）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟

位域/位	名称	参考设定值
23-20	CKDIV	上图设置：0x1，EXMC_CLK=2HCLK
19-16	BUSLAT	EXMC_NE[x]上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

同步复用突发写时序 – NOR，PSRAM（CRAM）

图 25-23. 同步复用突发传输写时序

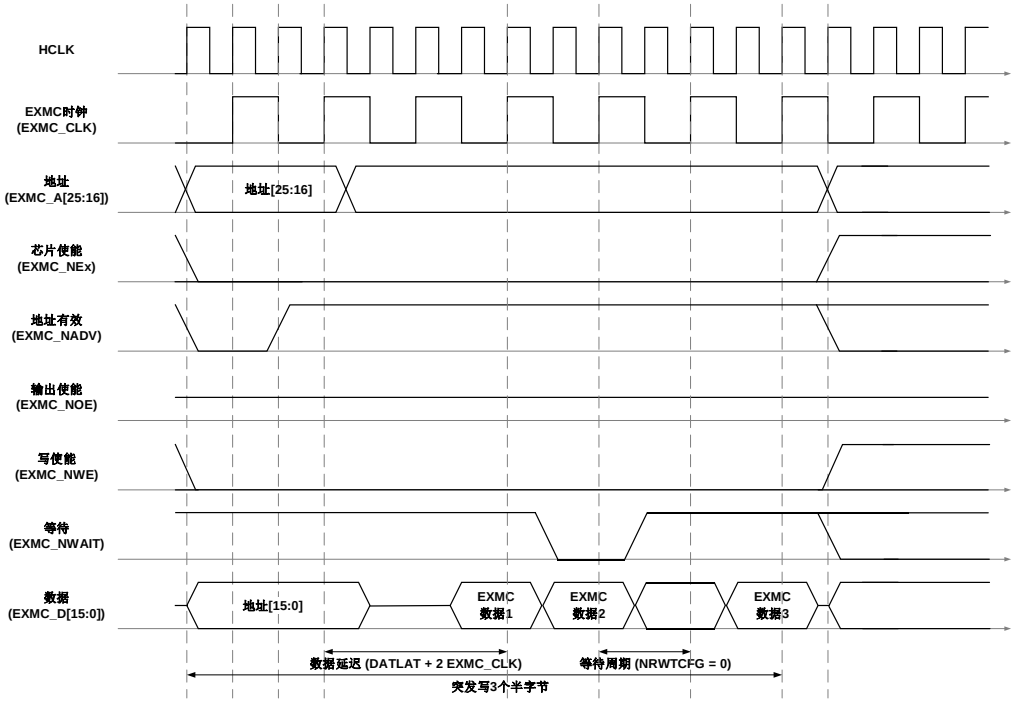


表 25-15. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_SNCTLx		
31-20	保留	0x000
19	SYNCWR	0x1，同步写使能
18-16	保留	0x0
15	AYSNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	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

位域/位	名称	参考设定值
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1，取决于用户
0	NRBKEN	0x1
EXMC_SNTCFGx（Write）		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	上图设置：0x1，EXMC_CLK=2HCLK
19-16	BUSLAT	无影响
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

SPI/QPI-PSRAM 访问时序

SPI/QPI-PSRAM 由 EXMC 的 Bank0，Region0 操作，它是具有 SPI 和 QPI 接口的 PSRAM。具有 6 个 IO 口，包含片选，时钟和 4 个数据 IO 口，详细见*表 25-16. SPI/QPI 接口*。

表 25-16. SPI/QPI 接口

信号	传输方向	SPI 模式	QPI 模式
EXMC_CLK	输出	串行时钟	
EXMC_NE[0]	输出	芯片使能（低有效）	
EXMC_D[0]	输入/输出	串行输出	数据 IO[0]
EXMC_D[1]	输入/输出	串行输入	数据 IO[1]
EXMC_D[2]	输入/输出	X	数据 IO[2]
EXMC_D[3]	输入/输出	X	数据 IO[3]

■ 控制器初始化

第一步，用户需要设置 SPI 初始化寄存器 EXMC_SINIT。由 POL 位设置数据采样时钟的有效边沿，IDL 位设置设备 ID 长度，ADRBIT 位设置地址长度，CMDBIT 位设置命令长度。

■ 读写操作

EXMC 支持 SPI，QPI 和 SQPI 三种访问模式，访问模式必须在读写操作之前设置。读写的访问模式由 RMODE 和 WMODE 设置，等待周期由 RWAITCYCLE 和 WWAITCYCLE 位设置，操作指令由 RCMD 和 WCMD 设置。这些位都在 EXMC_SRCMD 和 EXMC_SWCMD 寄存器。

在模式选择完毕以后，读写操作就和普通的 NOR Flash 一样，通过 EXMC 的 Bank0 的 Region 进行。

■ 读设备 ID

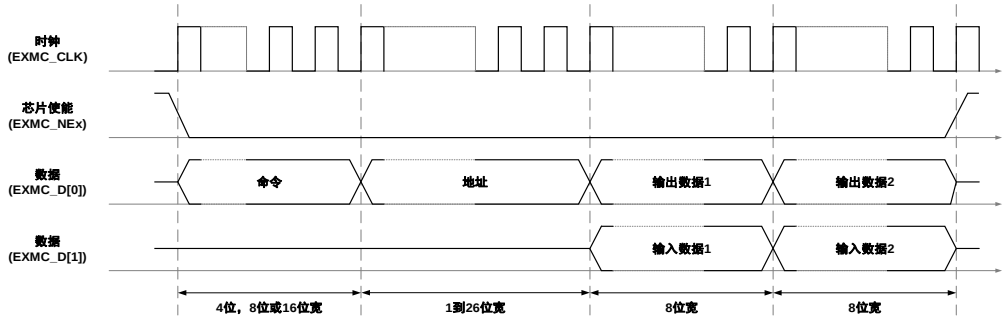
读设备 ID 是一个特殊的指令，需要查询 SC 位直到它为 0，然后把它置 1。然后低 32 位的 ID 会存储在 EXMC_SIDL 寄存器中，高 32 位 ID 会存储在寄存器 EXMC_SIDH 中。

■ SPI-PSRAM 访问时序

在 SPI 模式中，EXMC 可以使用 SPI 协议，通过 4 个 IO 口，时钟，片选和输入输出数据线与外部存储器通信。如*图 25-24. SPI-PSRAM 访问时序*所示，设置外部存储器模式的指令先通过输出数据线串行输出，然后是与 EXMC 设置有关的任意大小的地址，最后是读写的数据。读写操作时，写是通过数据输出线，读是通过数据输入线。

当 SADRBIT[4:0] = 24，CMDBIT[1:0] = 1 时，SPI-PSRAM 通信如下：

图 25-24. SPI-PSRAM 访问时序

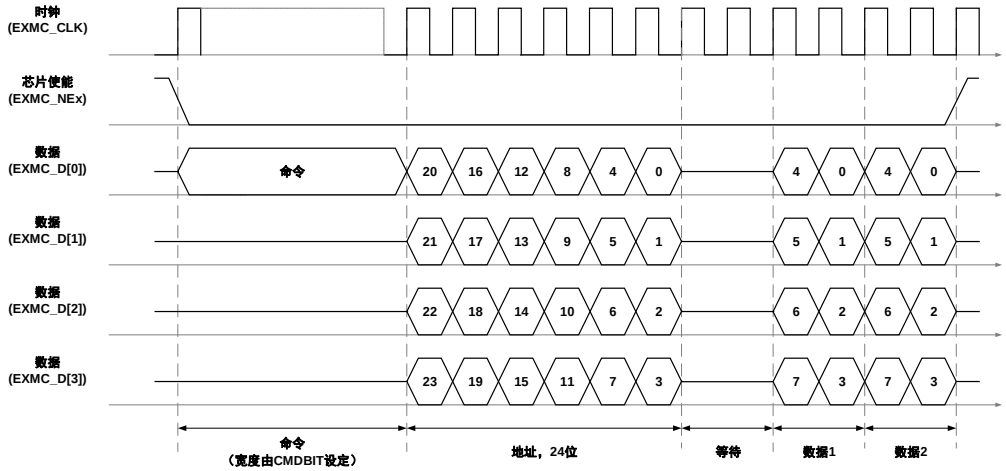


■ SQPI - PSRAM 访问时序

在 SQPI 模式中，EXMC 在指令阶段使用 SPI 协议，地址和数据阶段使用 QSPI 协议，通过 6 个 IO 口，时钟，片选和 4 个数据线与外部存储器通信。如*图 25-25. SQPI-PSRAM 访问时序*所示，设置外部存储器模式的指令先通过数据线 data[0]串行输出，之后是通过 4 个数据线并行输入/输出的地址和数据。

当 ADRBIT[4:0] = 24，CMDBIT[1:0] = 1（可以使用其他值），RWAITCYCLE[3:0] = WWAITCYCLE[3:0] = 2（可以使用其他值），SQPI-PSRAM 通信如下：

图 25-25. SQPI-PSRAM 访问时序

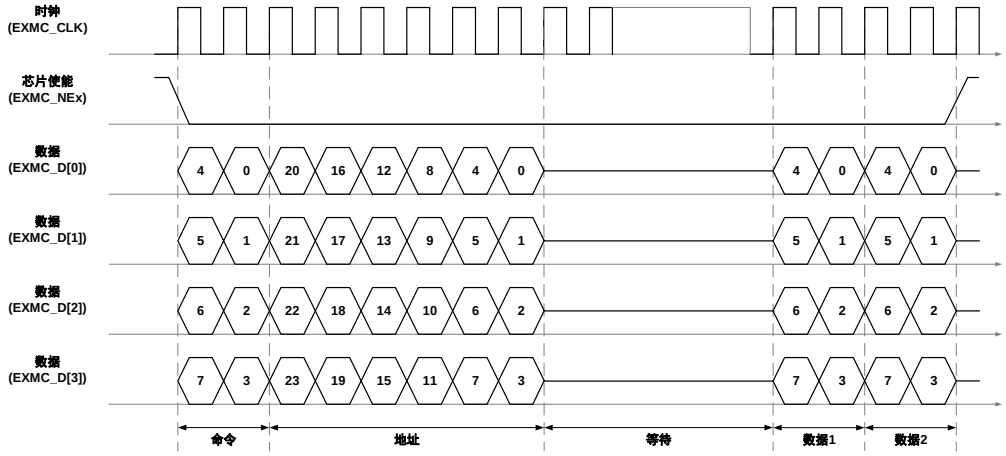


■ QPI – PSRAM 访问时序

QPI 与 SQPI 唯一的区别就是指令也是通过并行 4 个数据线输出的。

当 ADRBIT[4:0] = 24，CMDBIT[1:0] = 1，QPI-PSRAM 通信如下：

图 25-26. QPI-PSRAM 访问时序



25.3.5. NAND Flash 或 PC Card 控制器

EXMC模块Bank1、Bank2支持NAND FLASH，Bank3支持PC Card设备。对于每个Bank，EXMC提供独立的寄存器来配置访问时序，支持8位、16位的NAND FLASH以及16位PC卡。对于NAND FLASH，EXMC还提供ECC计算模块，保证数据传输和保存的鲁棒性。

NAND Flash/PC Card 接口功能

表 25-17. 8 位/16 位 NAND 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[17]	输出	NAND Flash 地址锁存（ALE）
EXMC_A[16]	输出	NAND Flash 命令锁存（CLE）
EXMC_D[7:0]/ EXMC_D[15:0]	输入/输出	8 位复用，双向地址/数据总线
		16 位复用，双向地址/数据总线
EXMC_NCE[x]	输出	片选，x = 1，2
EXMC_NOE（NRE）	输出	输出使能
EXMC_NWE	输出	写使能
EXMC_NWAIT/ EXMC_INT[x]	输入	NAND Flash 就绪/忙输入信号 EXMC，x=1，2

表 25-18. 16 位 PC Card 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[10:0]	输出	地址总线
EXMC_NIOS16	输入	仅适合 16 位传输的 I/O 空间的数据传输宽度（必须接地）
EXMC_NIORD	输出	I/O 空间输出使能
EXMC_NIOWR	输出	I/O 空间写使能
EXMC_NREG	输出	决定访问通用空间还是属性空间
EXMC_D[15:0]	输入/输出	双向数据总线
EXMC_NCE3_x	输出	片选（x=0，1）
EXMC_NOE	输出	输出使能

EXMC 引脚	传输方向	功能描述
EXMC_NWE	输出	写使能
EXMC_NWAIT	输入	PC Card 等待信号
EXMC_INTR	输入	PC Card 中断输入信号
EXMC_CD	输入	PC Card 卡存在检测信号，高有效

支持的存储器访问模式

表 25-19. Bank1/2/3 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/PC Card	异步	R	8	
	异步	W	8	不支持此操作
	异步	R	16	
	异步	W	16	
	异步	R	32	分成 2 次 EXMC 访问
	异步	W	32	

NAND Flash/PC Card 的控制时序

EXMC能够为NAND Flash、PC卡等设备产生合适的时序信号。每个Bank都有相应的寄存器来对外部存储器进行管理和控制，例如EXMC_NPCTx、EXMC_NPINTENx、EXMC_NPCTCFGx、EXMC_NPATCFGx、EXMC_PIOTCFG3、EXMC_NECCx，其中寄存器EXMC_NPCTCFGx、EXMC_NPATCFGx、EXMC_NPATCFG3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。

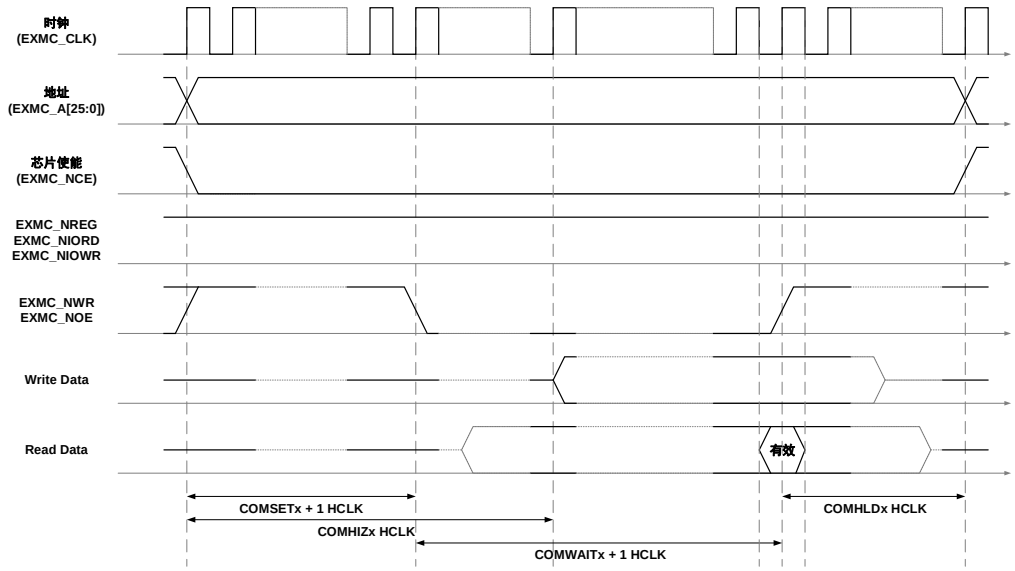
表 25-20. NADN/PC Card 可编程参数

参数	读/写	单位	功能描述	NAND Flash/ PC Card	
				最小值	最大值
存储器数据总线高阻时间（HIZ）	W/R	HCLK	启动写操作之后保持数据总线为高阻态的时间	0	255
存储器保持时间（HLD）	W/R	HCLK	在发送命令结束后保持地址的（HCLK）时钟周期数目，写操作时也是数据的保持时间	1	255
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	1	256

参数	读/写	单位	功能描述	NAND Flash/ PC Card	
				最小值	最大值
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	256

*图 25-27. NAND/PC Card 通用空间操作时序*给出了在通用存储空间中操作的可编程参数定义，属性存储空间和 I/O 空间（只适用于 PC Card）中操作与此相似。

图 25-27. NAND/PC Card 通用空间操作时序



NAND Flash 操作

EXMC在对NAND Flash发送命令或地址时，需要利用其命令锁存信号（A[16]）或地址锁存信号（A[17]）这两条地址线，即CPU需要在特定的地址进行写操作。

示例：NAND Flash读操作步骤：

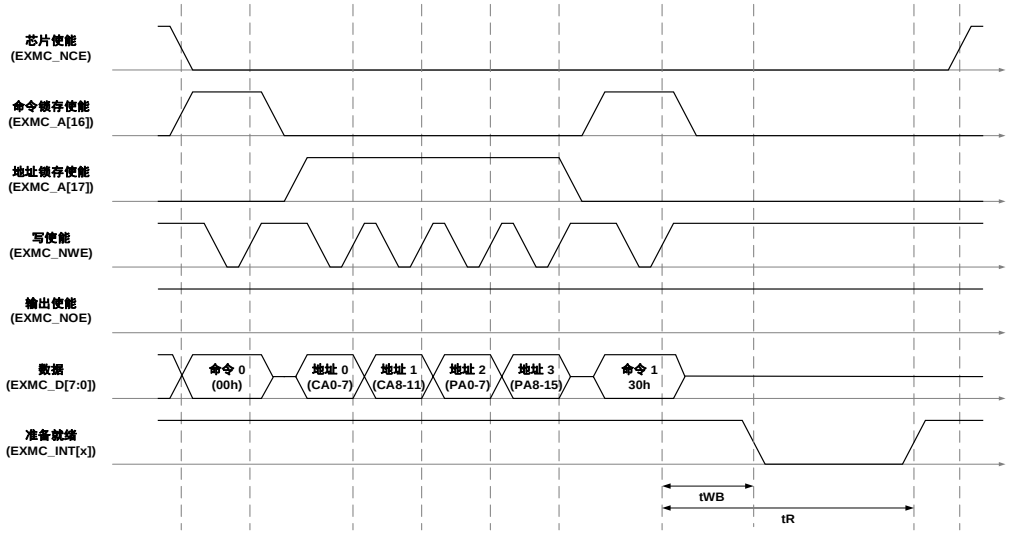
1. 配置 EXMC_NPCTLx、EXMC_NPCTCFGx，若需要预等待功能，还需配置 EXMC_NPATCFGx；
2. 往通用空间写入 NAND Flash 读数据命令，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_CLE（A[16]）变为有效电平（高），则被 NAND 认为写入命令；
3. 往通用空间写入读操作的起始地址，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_ALE（A[17]）变为有效电平（高），则被 NAND 认为写入地址；
4. 等待 NAND 就绪信号，NAND 控制器会在这期间将和 EXMC_NCE 一直保持有效；
5. 从通用空间的数据区逐字节的读出数据；
6. 在不写入新的命令和地址，可以自动读出 NAND 下一页数据；或转到步骤 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 25-28. NCE 敏感 NAND Flash 访问时序



1. 往 NAND 的通用空间命令区写入命令 CMD0
2. 往 NAND 的通用空间地址区写入操作地址 ADD0
3. 往 NAND 的通用空间地址区写入操作地址 ADD1
4. 往 NAND 的通用空间地址区写入操作地址 ADD2
5. 往 NAND 的通用空间地址区写入操作地址 ADD3
6. 往 NAND 的属性空间命令区写入命令 CMD1

在 6）中写命令操作，EXMC 使用的是寄存器 EXMC_NPATCFGx 定义的时序。经过 ATTHLD 时间后，NAND Flash 等待 EXMC_INTx 信号，ATTHLD 要大于 t_{WB}（EXMC_NWE 高到 EXMC_INTx 低）。对于那些对片选信号敏感的 NAND Flash，在最后一个地址字节输入后，一直到 B/NB 就绪状态到来的这段时间中，要求片选信号 NCE 一直保持低电平。这里可以通过配置属性存储空间的 ATTHT 的值来满足 t_{WB} 的时序，这样 MCU 只有在写入最后一个地址字节时才使用属性存储空间的时序，而在其他时候都使用通用存储空间的时序。

NAND Flash 的 ECC 计数模块

EXMC模块中的Bank1和Bank2各有一个ECC计算的硬件模块，用户可以根据EXMC_NPCTLx中的ECCSZ来选择ECC计算的页面大小，通过ECC计算可以矫正1个bit的错误并且能检测2个bit的错误。

当NAND存储器块使能，ECC模块就会检测D[15:0]以及EXMC_NCE、EXMC_NEW信号。当已经完成ECCSZ大小字节的读写操作时，软件必须读出EXMC_NECCx中的结果值。如果需要再次开始ECC计算，软件需要先将EXMC_NPCTLx中ECCEN清0来清除EXMC_NECCx中的值，再将ECCEN置1来重新启动ECC计算。

PC/CF Card 访问

EXMC 的 Bank3 专用于访问 PC/CF Card，同时支持存储器和 IO 模式。Bank3 分为 3 个子空间，分别为存储空间，属性空间和 IO 空间。

EXMC_NCE3_0 和 EXMC_NCE3_1 是字节选择信号，当仅有 EXMC_NCE3_0 有效时，低字节或高字节的选择取决于 EXMC_A[0]，当仅有 EXMC_NCE3_1 有效时，硬件不支持，当 EXMC_NCE3_0 和 EXMC_NCE3_1 都有效时，16 位操作。复位 NDTP 来选择 PC/CF Card 作为外部存储器，寄存器 EXMC_NPCTLx 位 NDW 必须设置为 01 来保证 EXMC 的正确操作。

下面是对不同空间的访问：

- 通用空间：EXMC_NCE3_x（x= 0，1）是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 位高电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
- 属性空间：EXMC_NCE3_x（x= 0，1）是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 位低电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
- IO 空间：EXMC_NCE3_x（x= 0，1）是片选信号，表示同时支持 8 位和 16 位的访问操作。在 EXMC_NREG 位低电平时，EXMC_NIOWR 为低电平时写操作，EXMC_NIORD 为低电平时读操作。

AHB 访问 16 位的 PC/CF Card：

- 通用空间：数据存储的位置，支持字节和半字访问，奇地址禁止字节访问。当 AHB 进行字访问，EXMC 会自动分成两次连续的半字操作。在 EXMC_NREG 位高电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
- 属性空间：配置信息存储的位置，仅偶地址支持字节访问，半字访问会被转换为单次字节操作，字访问会被转换为两次字节访问。半字与字访问时，只有 EXMC_NCE3_0 有效。在 EXMC_NREG 位低电平时，EXMC_NWE 为低电平时写操作，EXMC_NOE 为低电平时读操作。
- IO 空间：同时支持字节和半字访问，EXMC_NREG 位低电平时，EXMC_NIOWR 为低电平时写操作，EXMC_NIORD 为低电平时读操作。

25.3.6. SDRAM 控制器

主要特性

- 两个可独立配置的 SDRAM 设备；
- 8 位，16 位，32 位数据带宽；
- 多达 13 位行地址、11 位列地址、2 位内部 bank 地址；
- 支持存储器大小：4x16Mx32bit（256MB），4x16Mx16bit（128MB），4x16Mx8bit（64MB）；
- AHB 字、半字、字节访问；
- 为每个存储器 bank 提供独立的片选控制；

- 每个存储器 bank 可独立配置；
- 写使能和字节选择输出；
- 自动进行行和 bank 边界管理；
- 多个 bank 的乒乓访问；
- SDRAM 时钟可以为 $f_{HCLK}/2$ 或者 $f_{HCLK}/3$；
- 可编程的时序参数；
- 可编程刷新速率的自动刷新操作；
- 通过软件进行上电初始化；
- CAS 延迟可设置为 1、2、3 个时钟周期；
- 具有 16x35 位深度的写数据 FIFO；
- 具有 16x31 位深度的写地址 FIFO；
- 6x32 位深度的可缓存的读数据 FIFO；
- 6x14 位深度的可缓存读地址 FIFO；
- 可调整的读数据采样时钟；
- 自刷新模式；
- 掉电模式。

SDRAM 简介

同步动态随机存储器（SDRAM）是通过外部同步时钟刷新的动态随机存储器（DRAM），它的同步时钟由EXMC的EXMC_SDCLK引脚提供，通过配置寄存器EXMC_SDCTLx位SDCLK时钟频率可设置为$f_{HCLK}/2$或者$f_{HCLK}/3$。指令和数据在时钟的上升沿锁存，在下降沿改变。

SDRAM内部分为多个叫做Bank的区域，允许设备以交错的方式进行访问，以获取更大的并发性和数据传输量。每个Bank可以认为是一个矩阵，其中每个地址对应存储器存储宽度的空间，矩阵由行和列构成，因此存储器的Bank大小可以认为是存储器数据宽度*行数*列数。用户可以通过设置寄存器EXMC_SDCTLx位NBK，SDW，RAW，CAW使EXMC可以与不同的SDRAM进行通信。

由于易失的本征特性，SDRAM需要周期性的刷新。EXMC支持两种刷新模式，自刷新和自动刷新。自刷新是在EXMC挂起的低功耗模式中使用，由SDRAM内部计数提供时钟，内部进行刷新。自动刷新是由EXMC周期性的提供刷新命令，因为此时SDRAM需要进行数据传输，刷新间隔由寄存器EXMC_SDARI位ARINTV决定，连续刷新次数由寄存器EXMC_SDCMD位NARF决定。刷新命令优先级高于其他的包括读写命令，来保证数据的正常存储，当SDRAM同时收到刷新命令与读写命令时，读写命令需要等待刷新命令完成才能进行。如果在前一个刷新命令未完成时，再次接收到刷新命令，寄存器EXMC_SDSTAT刷新错误标志位（REIF）会被置位，同时如果刷新错误中断使能（REIE），将会发生刷新错误中断。

CAS延迟是读命令和数据线出现第一个可读数据之间的延迟，可以通过寄存器EXMC_SDCTLx位CL设置。

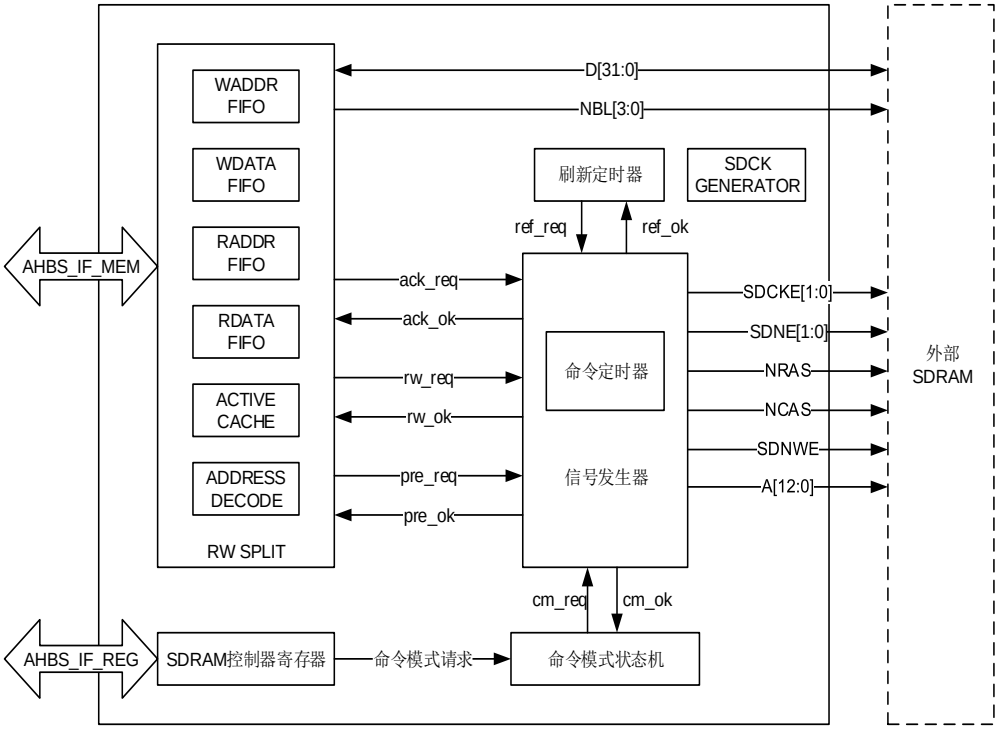
对不同的SDRAM需要参考其手册，使用模式寄存器进行设置，包含突发长度，突发类型，CAS延迟，写模式。在寄存器EXMC_SDCMD位MRC中设置，会通过CMD命令发送给SDRAM。在读写操作之前，需要发送读取模式寄存器命令，否则SDRAM无法工作。

SDRAM 控制器简介

同步动态随机存储器控制器（SDRAMC）是MCU和SDRAM之间的接口。它把AHB的操作根据SDRAM协议转换为对SDRAM的操作，同时配置寄存器EXMC_SDTCFG满足时序要求。

SDRAMC 包含 4 个模块，读写预处理模块，控制寄存器，有限状态机和信号发生器。使用两组 FIFO 来提高存储器访问效率，一组用来写地址和数据，另外一组用来读地址数据。SDRAMC 模块由*图 25-29. SDRAM 系统架构*所示。

图 25-29. SDRAM 系统架构



信号发生器处理状态机，刷新定时器，读写模块产生的请求。

命令定时器由遵守SDRAM时序协议的计数器组成。

SDRAM 命令由 SDRAM 控制器接口发出，可见*表 25-21. SDRAM 命令真值表*:

表 25-21. SDRAM 命令真值表

SDNE	NRAS	NCAS	SDNWE	A[n]	A[10]	A[m]	命令
H	X	X	X	X	X	X	命令禁止（无操作）
L	H	H	H	X	X	X	无操作
L	H	H	L	X	X	X	中止突发传输
L	H	L	H	Bank	L	Col	突发读选择行
L	H	L	H	Bank	H	Col	预充电完成后，突发读选择行
L	H	L	L	Bank	L	Col	突发写选择行
L	H	L	L	Bank	H	Col	预充电完成后，突发写选择行

SDNE	NRAS	NCAS	SDNWE	A[n]	A[10]	A[m]	命令
L	L	H	H	Bank	Row	Row	行使能命令，之后可进行读写
L	L	H	L	Bank	L	X	预充电命令，关闭当前 Bank 的选择行
L	L	H	L	X	H	X	全局预充电命令，关闭所有 Bank 的选择行
L	L	L	H	X	X	X	SDCKE = 1 时自动刷新模式 SDCKE = 0 时自刷新模式
L	L	L	L	L	Mode	Mode	加载模式寄存器

SDRAM 控制器操作

IO 配置

SDRAMC 的 IO 口必须在与 SDRAM 通信之前配置，SDRAM 的 IO 配置如*表 25-22. SDRAM IO 定义*所示：

表 25-22. SDRAM IO 定义

信号	传输方向	描述
EXMC_SDCLK	O	SDRAM 存储器时钟
EXMC_SDCKE[0]	O	SDRAM device 0 的时钟使能信号
EXMC_SDCKE[1]	O	SDRAM device 1 的时钟使能信号
EXMC_SDNE[0]	O	SDRAM device 0 的片选信号，低电平有效
EXMC_SDNE [1]	O	SDRAM device 1 的片选信号，低电平有效
EXMC_NRAS	O	行地址选通，低电平有效
EXMC_NCAS	O	列地址选通，低电平有效
EXMC_SDNWE	O	写使能，低电平有效
EXMC_A[12:0]	O	地址
EXMC_A[15:14]	O	Bank 地址
EXMC_D[31:0]	I/O	读/写数据
EXMC_NBL[3:0]	O	写数据标记（掩码）

控制器初始化

用户需要按照以下步骤来初始化 SDRAM 控制器，支持同时初始化两个 SDRAM，由寄存器 EXMC_SDCMD 位 DS0 和 DS1 决定。为了保证读写的可靠性，必须先进行初始化。

1. 控制参数：控制配置寄存器 EXMC_SDCTLx 指定 SDRAM 的存储器行列数，时钟配置和读写方法。
2. 时序参数：时序配置寄存器 EXMC_SDTCFGx 需要根据 SDRAM 数据手册来配置。RPD 和 ARFD 在 EXMC_SDTCFG0 来配置，EXMC_SDTCFG1 这两位保留。
3. 使能 SDCLK：使用到的 SDRAM 需要给寄存器 EXMC_SDCMD 位域 CMD 写'0b001'使能相应设备的 SDCLK 信号，DS0 和 DS1 决定接收命令的设备。
4. 上电延迟：100us
5. 预充电：命令会对 SDRAM 的所有 Bank 进行复位，并使 SDRAM 回到空闲状态。给寄存

器 EXMC_SDCMD 位域 CMD 写'0b010'使能相应设备的 SDCLK 信号，DS0 和 DS1 决定接收命令的设备。

6. 设置自刷新模式：给寄存器 EXMC_SDCMD 位域 CMD 写'0b011'发送自刷新命令。用户也可以通过设置位 NARF 来设置连续刷新次数，DS0 和 DS1 决定接收命令的设备。
7. 模式寄存器配置：模式寄存器通过写寄存器 EXMC_SDCMD 位域 MRC 来设置，其中定义了突发长度，突发类型，CAS 延迟和读写模式，这些需要参考 SDRAM 用户手册。CAS 延迟必须与寄存器 EXMC_SDCTLx 位域 CL 对应，突发长度设为 1 来保证数据正常传输。如果两个 SDRAM 的模式寄存器内容不同，需要通过 DS0 和 DS1 单独选择设备来配置。
8. 设置自刷新频率：自刷新频率与刷新周期相对应，用户必须确保刷新周期满足 SDRAM 的要求。

这里控制器已经完成初始化，可以与 SDRAM 通信。如果发生了复位，初始化需要按照上述步骤重复一遍。在读写操作之前，要保证控制器至少初始化一遍。

预充电

若SDRAM控制器在存取时需要进行行切换，那么首先需要将该行地址对应块的读写放大器去使能，使其进入空闲状态，为下一行的读写操作进行准备。这个过程叫做预充电，或者行去使能。预充电可以由控制器的全局预充电命令（Precharge-All）独立激发，或者是在读写完成后自动激发。行预充电延时（RPD）代表SDRAM行切换的最小时间，他是预充电完成到下一次行使能命令的最小时间间隔

行使能

行使能命令将行地址所在的块使能，完整的行地址由2比特的块地址EXMC_A[15:14]和13比特的行地址EXMC_A[12:0]组成。行使能会将所选行的16384比特信息读入读写放大器，这个过程也叫做行开启，该命令的一个副作用就是对所选行的存储单元进行了刷新。

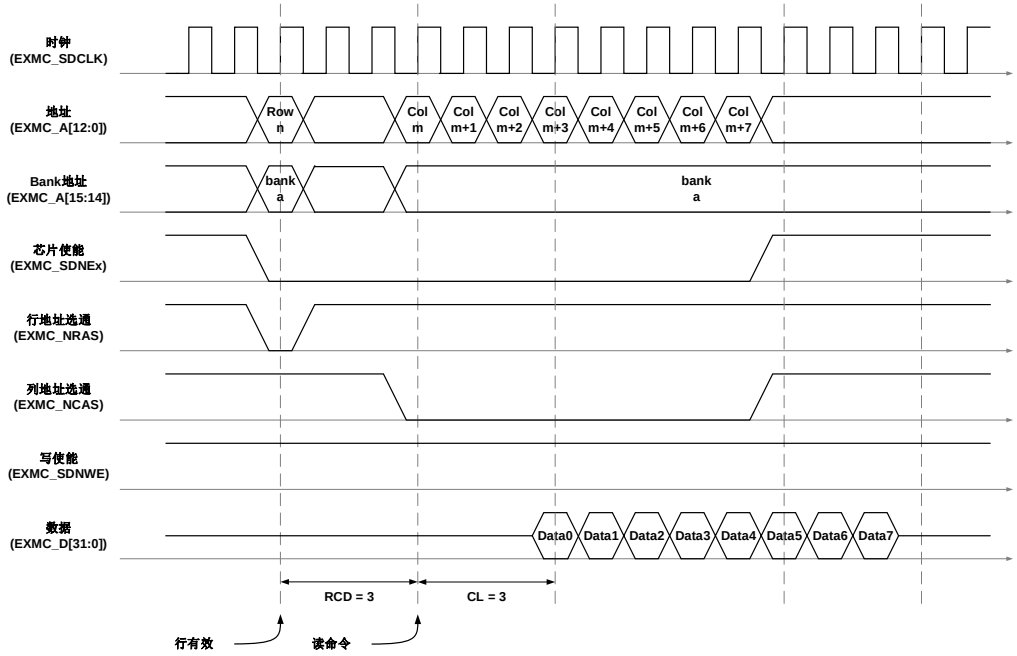
一旦行使能，读写操作就可以顺利的进行，但是行使能需要一定的时间，这个时间间隔叫做行列延时，它是行寻址到列寻址的最小时间间隔。对SDRAM控制器进行配置时的行列延时（RCD），是包含SDRAM行列延时的最小时钟周期数，它代表了行使能到SDRAM读写间的最小等待时间。在这段时间中，用户可向其它的块地址发出控制命令，因为SDRAM控制器对块的操作是独立进行的。

读写访问

控制器可以把AHB的单次或突发读操作转换成单次的存储器读操作。为了连续访问，控制器通常会保存之前操作的行号，若下一次读操作是在相同的行号或是已经使能的其他行号，操作会未中断的执行，否则需要先执行取消使能当前行和使能需要操作的行，最后才会执行操作。FIFO设计用来在CAS延迟和管道延迟的时候缓存数据。

*图 25-30. 突发读操作*是对一个未被使能的行突发读操作，在读之前发送了行使能指令。若对一个已经使能的行读操作，只需要发送列地址，行地址无需发送。

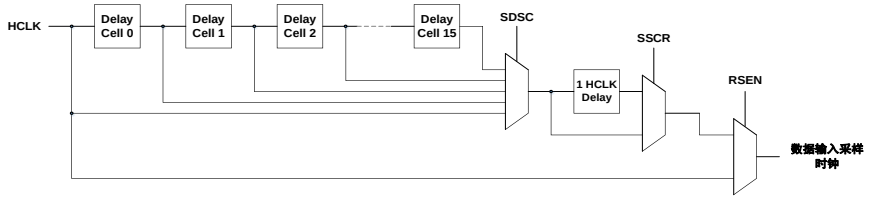
图 25-30. 突发读操作



当HCLK采样数据不准的时候，可以使用根据HCLK微调的内部可调时钟。当使能这个时钟的时候，读取的数据在进入AHB总线之前会先存储在异步FIFO中。2-3个HCLK延迟会被添加到读命令过程中。

在HCLK输入到信号发生器，增加了时钟延迟模块，这个延迟的时钟用来采样输入数据。延迟模块可以通过寄存器EXMC_SDRSCTL来控制，其中RSEN位选择是否使用HCLK延迟，SSCR位选择是否额外增加一个HCLK延迟，SDSC选择增加多少个HCLK延迟。可增加的延迟范围为0到15。

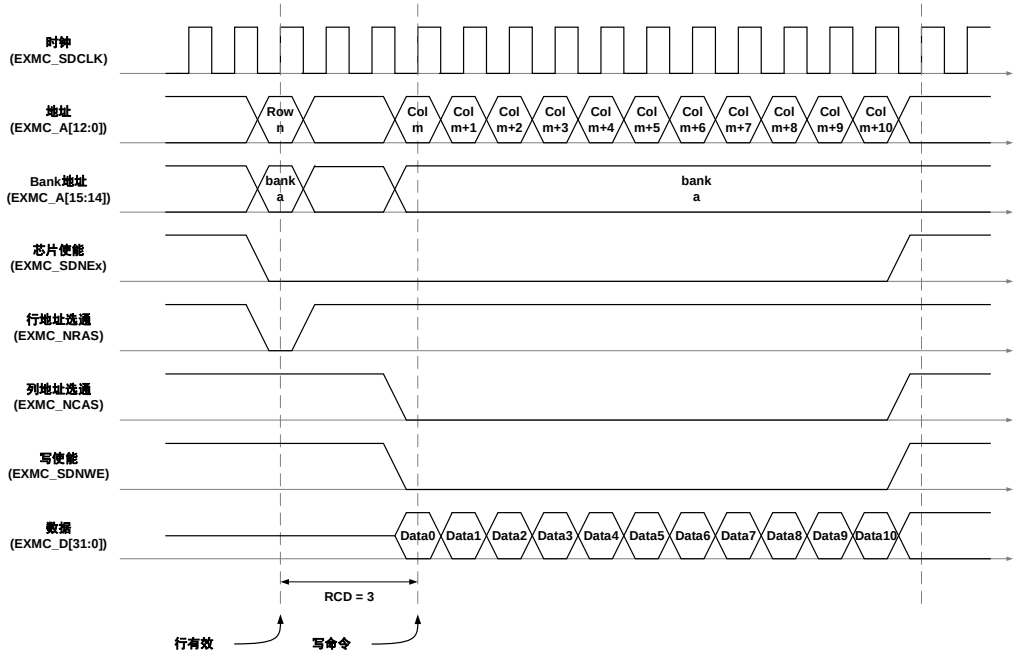
图 25-31. 数据采样时钟延迟模块



控制器可以把AHB的单次或突发写操作转换成单次的存储器写操作。写操作之前必须失能写保护位（寄存器EXMC_SDCTL位WPEN）。为了连续访问，控制器通常会保存之前操作的行号，若下一次操作是在相同的行号或是已经使能的其他行号，操作会未中断的执行，否则需要先执行取消使能当前行和使能需要操作的行，最后才会执行操作。

*图 25-32. 突发写操作*是对一个未被使能的行突发写操作，在写之前发送了行使能指令。若对一个已经使能的行写操作，只需要发送列地址，行地址无需发送。

图 25-32. 突发写操作



读写命令预处理模块接收AHB命令，然后根据AHB总线和SDRAM接口的数据总线宽度将AHB命令转换成单个的SDRAM访问。

在读写命令预处理模块中，有两个写FIFO，用于缓冲AHB写命令的地址和数据。当两个写FIFO都不为空时，产生写访问。

当寄存器EXMC_SDCTL0位BRSTRD置1时，读写命令预处理模块能够预处理下一个读访问。读FIFO被用来存储在CAS延迟（由EXMC_SDCTLx中的CL位配置）和PIPED延迟（由EXMC_SDCTL0中的PIPED位配置）期间提前读出的数据。

读数据FIFO能够最多缓存6个32位的读数据字，同时地址FIFO携带6个14位的读地址标签，这些标签用来标识6个32位的读数据字中每一个。每个地址标签由11位列地址，2位Bank地址和1位SDRAM设备选择位。

当在AHB总线上出现一个读命令时，读写命令预处理模块将首先检查这个地址是否和某个地址标签匹配，如果匹配，则直接从FIFO中读取数据。否则，向存储器发一个新的读命令，FIFO会被新的数据更新。如果FIFO满了，老的数据会被丢失。

读FIFO操作，如*图25-33. FIFO未命中时的读访问(BRSTRD=1，CL=2，SDCLK=2，PIPED=2)*所示。

图 25-33. FIFO 未命中时的读访问（BRSTRD=1，CL=2，SDCLK=2，PIPED=2）

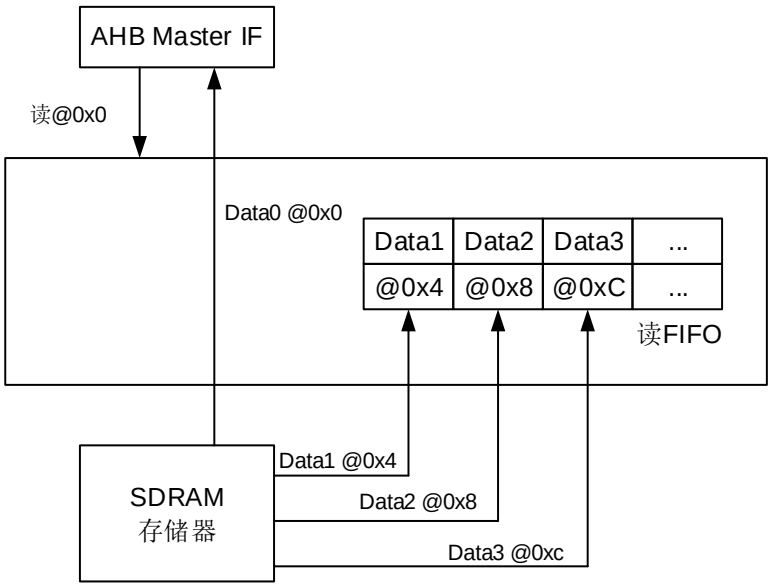
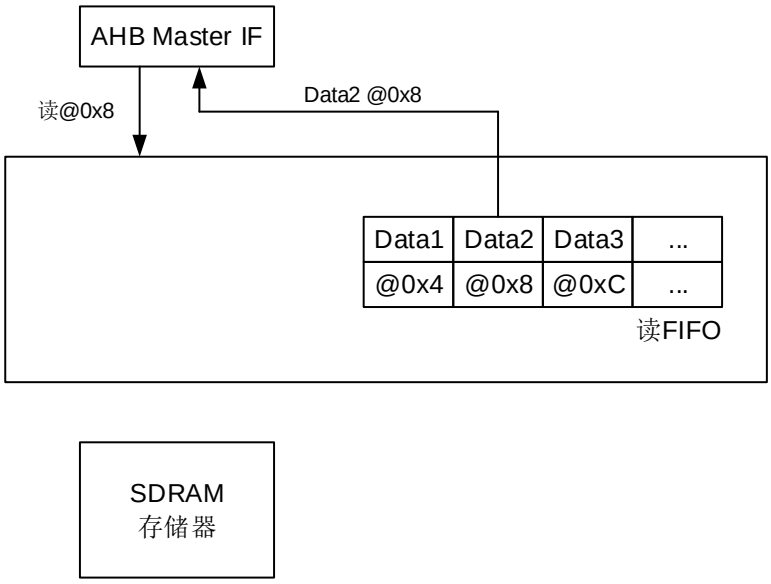


图 25-34. FIFO 命中时的读访问（BRSTRD=1）



当一个写访问或者预充电命令出现时，读FIFO缓冲区中的数据就会被清除掉，用以填充新的数据。

地址译码器子模块会根据外部存储器设备的配置将AHB总线地址转化成片选、内部bank地址、行地址和列地址。

使能缓存子模块记录着内部banks是否处于使能状态。当一个内部bank处于使能状态，则相应的行地址也会被记录。当AHB访问或者自动刷新命令出现时，读写命令预处理模块将会查询这个记录，并且决定是否生成使能或预充电命令。

在读/写操作之前，目标行必须被使能，EXMC_A[15:14]选择Bank，EXMC_A[12:0]选择行。被选择的行在预充电命令出现前会一直有效。预充电命令用来取消选择特定Bank或者所有Bank使能的行，必须在使能同Bank其他行之前发出。使能和预充电由EXMC自动执行，它的正确性

取决于存储器的相关配置和关于使能和预充电的读写时序配置。

图 25-35. 跨边界读操作

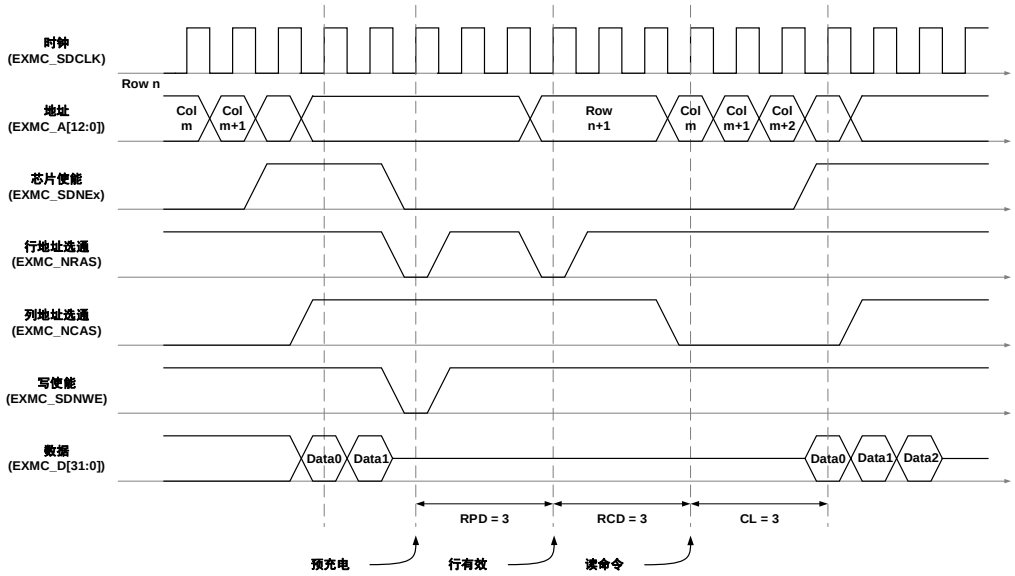
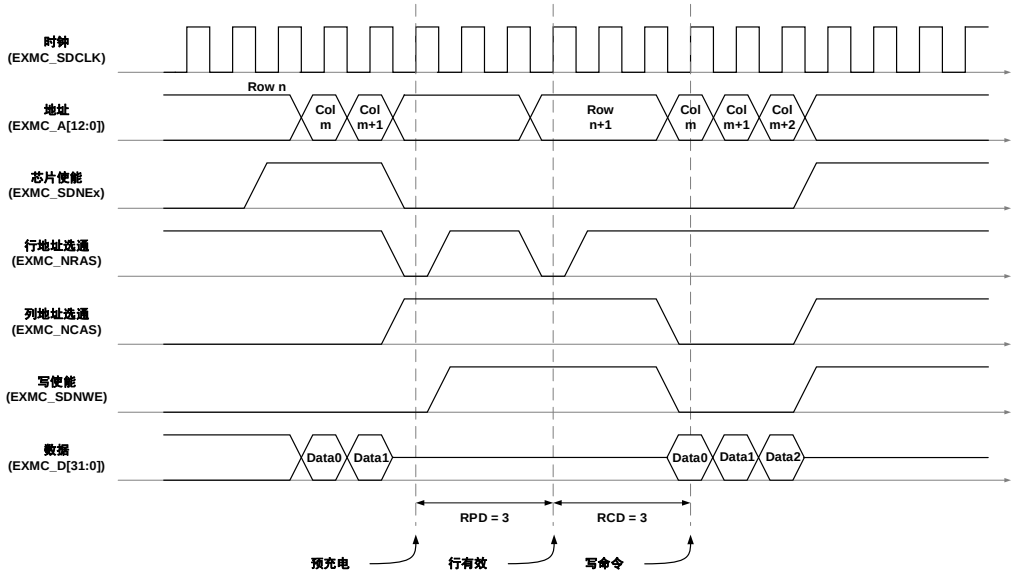


图 25-36. 跨边界写操作



上图描述了再跨行边界时的读写操作时序，会按照以下步骤自动执行：

1. 预充电当前行；
2. 使能下一行；
3. 读写操作。

预充电延迟（PRD）和行到列延迟（RCD）会根据寄存器 EXMC_SDTCFGx 设置。其他时序参数必须参照 SDRAM 标准要求。

当读写操作发生在 Bank 边界时，会有以下两种情况：

1. 当前 Bank 不是最后一个 Bank，使能下一个 Bank 的第一行，支持任意的行，列，总线宽度设置。

2. 当前 Bank 是最后一个 Bank，行，列，总线宽度设置为 13 位，11 位，32 位。假设当前操作的 SDRAM 位 device0，控制器会在 device1 上继续操作。

低功耗模式

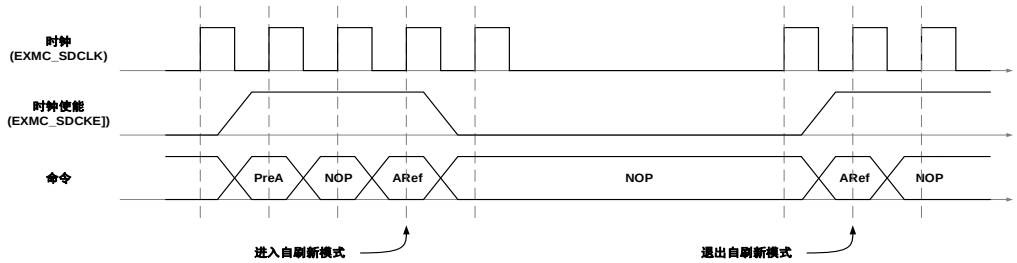
EXMC 支持两种低功耗模式：

1. 自刷新模式：在自刷新模式中，在没有外部时钟（EXMC_CLK）的情况下，刷新由 SDRAM 本身提供来保持数据。通过 DS0 和 DS1 选择设置，寄存器 EXMC_SDCMD 位域 CMD 写入'0b101'来发送自刷新指令。EXMC_SDCLK 在发送自刷新指令给所有设备或一个未初始化的设备之后经过 RASD 的延迟停止。
2. 掉电模式：在掉电模式中，刷新指令由 SDRAM 控制器提供，通过 DS0 和 DS1 选择设置，寄存器 EXMC_SDCMD 位域 CMD 写入'0b110'来进入掉电模式。如果写数据 FIFO 非空，在掉电模式使能之前，所有数据都会发送给存储器。

命令模式状态机也控制正常模式和低功耗模式（自刷新/掉电）之间的转换过程。

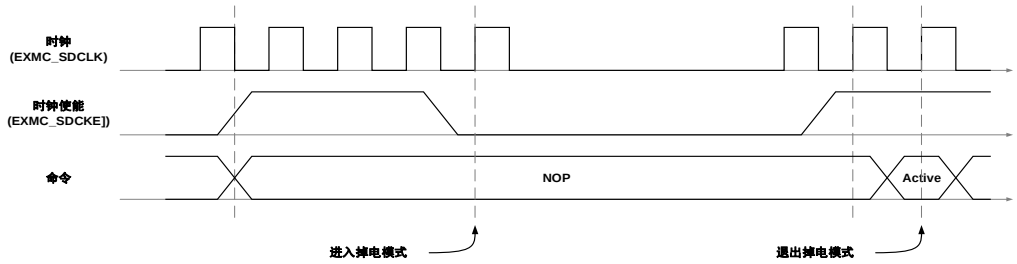
当读写访问出现时，SDRAM 控制器会从自刷新模式退出到正常模式。如果在 SDRAM 控制器进入自刷新模式时出现读写访问，则自刷新的进入过程会被中断，并且在读写访问完成后 SDRAM 控制器会停留在正常模式。

图 25-37. 自刷新模式进入和退出的处理



如果在SDRAM控制器处于掉电模式时出现自动刷新请求，那么SDRAM控制器会退出到正常模式，发"预充电所有存储区域"命令和自动刷新命令序列，然后再一次自动进入掉电模式。

图 25-38. 掉电模式进入和退出的处理



状态和中断

寄存器EXMC_SDSTAT的准备未完成状态位NRDY指示SDRAM是否准备完成接受新的命令。在控制器发送新的命令之后，该位会被清除。

寄存器EXMC_SDSTAT的STA0和STA1定义SDRAM的device0和device1的状态，0x00为普通

模式，0b01为自刷新模式，0b10为掉电模式。

若前一个刷新指令未完成时，接收到了新的刷新指令，寄存器EXMC_SDSTAT刷新错误标志位（REIF）会被置位，若之前将REIE位置位，还会产生中断。该位通过寄存器EXMC_SDARI位REC置位来清除。

25.4. EXMC 寄存器

EXMC基地址：0xA000 0000

25.4.1. NOR/PSRAM 控制器寄存器

SRAM/NOR Flash 控制寄存器（EXMC_SNCTLx）（x=0，1，2，3）

地址偏移：0x00 + 8 * x，（x = 0，1，2，3）
复位值：0x0000 30DA

该寄存器只能按字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXMODEN	NRWTEN	WREN	NRWTCFG	WRAPEN	NRWTPOL	SBRSTEN	保留	NREN	NRW[1:0]		NRTP[1:0]		NRMUX	NRBKEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw		rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	SYNCWR	选择写操作模式 0：异步写操作 1：同步写操作
18:16	保留	必须保持复位值。
15	ASYNCWAIT	异步等待功能使能位 0：禁用异步等待功能 1：使能异步等待功能
14	EXMODEN	扩展模式使能 0：禁用扩展模式，即不使用 EXMC_SNWTCFGx。 1：使能扩展模式
13	NRWTEN	NWAIT 信号使能 对于存储器的突发模式访问，该位使能/禁用等待状态插入 NWAIT 信号功能。 0：成组传输模式时，禁用 NWAIT 信号 1：成组传输模式时，使能 NWAIT 信号
12	WREN	写操作使能 0：禁止 EXMC 对外部存储器的写操作，否则产生一个 AHB 错误。 1：允许 EXMC 对外部存储器的写操作（复位缺省值）

11	NRWTCFG	NWAIT 信号配置，只在同步模式有效 0：NWAIT 信号在等待状态前的一个数据周期有效 1：NWAIT 信号在等待状态期间有效
10	WRAPEN	非对齐成组模式使能 0：禁止非对齐成组操作 1：允许非对齐成组操作
9	NRWTPOL	NWAIT 信号极性 0：NWAIT 低电平有效 1：NWAIT 高电平有效
8	SBRSTEN	同步突发模式使能 0：禁止同步突发模式 1：使能同步突发模式
7	保留	必须保持复位值
6	NREN	NOR 闪存访问使能 0：禁止 NOR Flash 访问 1：允许 NOR Flash 访问
5:4	NRW[1:0]	存储器数据宽度 00：8 位 01：16 位（复位缺省值） 10/11：保留
3:2	NRTP[1:0]	存储器类型 00：SRAM、ROM 01：PSRAM（CRAM） 10：NOR Flash 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：使能地址/数据复用功能
0	NRBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块

SRAM/NOR Flash 时序寄存器（EXMC_SNTCFGx）（x=0，1，2，3）

地址偏移：0x04 + 8 * x，（x = 0，1，2，3）
复位值：0x0FFF FFFF

该外设寄存器可以32位访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		ASYNCMOD[1:0]		DLAT[3:0]				CKDIV[3:0]				BUSLAT[3:0]			

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	ASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式中使用 00：模式 A 01：模式 B 10：模式 C 11：模式 D
27:24	DLAT[3:0]	NOR Flash 数据延时，仅在同步模式有效 0x0：第一数据的保持时间为 2 个 EXMC_CLK 时钟周期 0x1：第一数据的保持时间为 3 个 EXMC_CLK 时钟周期 …… 0xF：第一数据的保持时间为 17 个 EXMC_CLK 时钟周期
23:20	CKDIV[3:0]	同步模式时钟分频比，仅在同步模式有效 0x0：保留 0x1：EXMC_CLK 周期=2 个 HCLK 周期 …… 0xF：EXMC_CLK 周期=16 个 HCLK 周期
19:16	BUSLAT[3:0]	总线延迟时间 在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最小时间 0x0：总线延迟=1 个 HCLK 周期 0x1：总线延迟=2 个 HCLK 周期 …… 0xF：总线延迟=16 个 HCLK 周期
15:8	DSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=2 个 HCLK 周期 …… 0xFF：数据建立时间=256 个 HCLK 周期
7:4	AHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在模式 D 与复用模式有效 0x0：保留 0x1：地址建立时间=2 个 HCLK ……

		0xF：地址建立时间=16 个 HCLK
3:0	ASET[3:0]	异步地址建立时间 该位域设置地址建立时间 注意：该位域仅在 SRAM，ROM，NOR Flash 的异步模式有效 0x0：地址建立时间= 1 个 HCLK …… 0xF：地址建立时间= 16 个 HCLK

SRAM/NOR Flash 写时序寄存器（EXMC_SNWTCFGx）（x=0，1，2，3）

地址偏移：0x104 + 8 * x，（X = 0，1，2，3）
复位值：0x0FFF FFFF

该寄存器仅在扩展模式使能（寄存器EXMC_SNCTL位EXMODEN置1）后有效。该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:28	WASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式中使用 00：模式 A 01：模式 B 10：模式 C 11：模式 D
27:24	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	保留	必须保持复位值。

15:8	WDSET[7:0]	异步数据建立时间 该位域仅在异步模式有效 0x00：保留 0x01：数据建立时间=2 个 HCLK 周期 …… 0xFF：数据建立时间=256 个 HCLK 周期
7:4	WAHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在模式 D 与复用模式有效 0x0：保留 0x1：地址建立时间=2 个 HCLK …… 0xF：地址建立时间=16 个 HCLK
3:0	WASET[3:0]	异步地址建立时间 该位域设置地址建立时间 注意：该位域仅在 SRAM，ROM，NOR Flash 的异步模式有效 0x0：地址建立时间= 1 个 HCLK …… 0xF：地址建立时间= 16 个 HCLK

25.4.2. NAND Flash/PC Card 控制器寄存器

NAND Flash/PC Card 控制寄存器（EXMC_NPCTLx）（x=1，2，3）

地址偏移：0x40 + 0x20 * x，（x = 1，2，3）
复位值：0x0000 0008

该外设寄存器可以32位访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												ECCSZ[2:0]			ATR[3]
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATR[2:0]			CTR[3:0]				保留		ECCEN	NDW[1:0]		NDTP	NDBKEN	NDWTEN	保留
	rw				rw				rw	rw		rw	rw	rw	

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:17	ECCSZ[2:0]	ECC 块大小 000：256 字节 001：512 字节 010：1024 字节 011：2048 字节 100：4096 字节

		101：8192 字节
16:13	ATR[3:0]	ALE 至 RE 的延迟 0x0：1 个 HCLK …… 0xF：16 个 HCLK
12:9	CTR[3:0]	CLE 至 RE 的延迟 0x0：1 个 HCLK 0x1：2 个 HCLK …… 0xF：16 个 HCLK
8:7	保留	必须保持复位值。
6	ECCEN	ECC 使能 0：关闭 ECC，并复位 EXMC_NECCx 1：打开 ECC
5:4	NDW[1:0]	外部存储器数据宽度 00：8 位 01：16 位 其他：保留 注意：对于 PC/CF Card，数据宽度必须选择 16 位
3	NDTP	外部存储器的类型 0：PC Card，CF Card，PCMCIA 1：NAND Flash
2	NDBKEN	存储块使能 0：禁用对应的存储器块 1：使能对应的存储器块
1	NDWTEN	NWAIT 信号使能位 0：关闭等待功能 1：使能等待功能
0	保留	必须保持复位值。

NAND Flash/PC Card 中断使能寄存器（EXMC_NPINTENx）（x=1，2，3）

地址偏移：0x44 + 0x20 * x，（x = 1，2，3）
复位值：0x0000 0042（对于bank1和bank2）0x0000 0042（对于bank3）

该外设寄存器可以32位访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									FFEPT	INTFEN	INTHEN	INTREN	INTFS	INTHS	INTRS
									r	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	FFEPT	FIFO 空标志位 0：FIFO 非空 1：FIFO 空
5	INTFEN	中断下降沿检测使能 0：禁用中断下降沿检测 1：使能中断下降沿检测
4	INTHEN	中断高电平检测使能 0：禁用中断高电平检测 1：使能中断高电平检测
3	INTREN	中断上升沿中断检测使能 0：禁用中断上升沿检测 1：使能中断上升沿检测
2	INTFS	中断下降沿状态 0：没有检测到中断下降沿 1：检测到中断下降沿
1	INTHS	中断高电平状态 0：没有检测到中断高电平 1：检测到中断高电平
0	INTRS	中断上升沿状态 0：没有检测到中断上升沿 1：检测到中断上升沿

NAND Flash/PC Card 通用空间时序寄存器（EXMC_NPCTCFGx）（x=1，2，3）

地址偏移：0x48 + 0x20 * x，（x = 1，2，3）
复位值：0xFFFF FFFF

这些操作适用于以下类型的外部存储器的通用存储空间16位的PC Card，CF Card和NAND Flash。该外设寄存器可以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：256 个 HCLK
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：255 个 HCLK
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：256 个 HCLK （加上 NWAIT 时钟周期）
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：256 个 HCLK

NAND Flash/PC Card 属性空间时序寄存器（EXMC_NPATCFGx）（x=1，2，3）

地址偏移：0x4C + 0x20 * x，（x = 1，2，3）
复位值：0xFFFF FFFF

这些操作适用于以下类型的外部存储器的属性存储空间8位的PC Card和NAND Flash。该外设寄存器可以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：256 个 HCLK
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：255 个 HCLK
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：256 个 HCLK （加上 NWAIT 时钟周期）
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：256 个 HCLK

PC Card I/O 空间时序寄存器（EXMC_PIOTCFG3）

地址偏移：0xB0
复位值：0xFFFF FFFF

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:24	IOHIZ[7:0]	I/O 空间数据总线的高阻时间 定义在 IO 空间进行写操作后数据总线保持高阻态时间 0x00：0 个 HCLK …… 0xFF：255 个 HCLK
23:16	IOHLD[7:0]	I/O 空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00：保留 0x01：1 个 HCLK …… 0xFF：255 个 HCLK
15:8	IOWAIT[7:0]	I/O 空间的等待时间 定义了保持命令的最小时间 0x00：保留 0x01：2 个 HCLK （加上 NWAIT 时钟周期） …… 0xFF：256 个 HCLK （加上 NWAIT 时钟周期）
7:0	IOSET[7:0]	I/O 空间的建立时间 定义地址信号的建立时间 0x00：1 个 HCLK …… 0xFF：256 个 HCLK

NAND Flash ECC 结果寄存器（EXMC_NECCx）（x=1，2）

地址偏移：0x54+0x20 * x，(x=1, 2)
复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:0	ECC[31:0]	ECC计算结果

ECCSZ[2:0]	NAND Flash 页大小	ECC 位
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

25.4.3. SDRAM 控制器寄存器

SDRAM 控制寄存器 （EXMC_SDCTLx）（x=0，1）

地址偏移：0x140+4*x，（x = 0，1）

复位值：0x0000 02D0

该外设寄存器可以32位访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PIPED[1:0]		BRSTRD	SDCLK[1:0]		WPEN	CL[1:0]		NBK	SDW[1:0]		RAW[1:0]		CAW[1:0]	
	rw		rw	rw		rw	rw		rw	rw		rw		rw	

位/位域	名称	描述
31:15	保留	必须保持复位值。
14:13	PIPED[1:0]	流水线读数据 这些位用于指定在 CAS 延迟之后再延迟多少个 HCLK 时钟周期才去读数据 00：延迟 0 个 HCLK 周期 01：延迟 1 个 HCLK 周期 10：延迟 2 个 HCLK 周期 11：保留 注意：寄存器 EXMC_SDCTL1 相应位保留
12	BRSTRD	突发读开关

		当该位被置位时，会在 CAS 延迟期间预期处理下一个读命令，并将数据存储到读 FIFO 中。 0：禁用突发读 1：使能突发读 注意：寄存器 EXMC_SDCTL1 相应位保留
11:10	SDCLK[1:0]	SDRAM 时钟配置 这些位指定了两个 SDRAM device 的时钟周期。如果需要修改存储器时钟配置，首先需要将存储器时钟禁用，并且在修改配置后将存储器重新初始化。 00：SDCLK 存储器时钟禁用 01：保留 10：SDCLK 存储器周期为 2 个 HCLK 11：SDCLK 存储器周期为 3 个 HCLK 注意：寄存器 EXMC_SDCTL1 相应位保留
9	WPEN	写保护 该位禁用写保护功能 0：禁用写保护，允许写访问 1：使能写保护，忽略写访问
8:7	CL[1:0]	CAS 延迟 这些位用于设定 SDRAM CAS 延迟多少个 SDRAM 存储器时钟周期单元 00：保留不使用 01：1 个周期 10：2 个周期 11：3 个周期
6	NBK	内部 Bank 的个数 该位指定内部 Bank 的个数 0：2 个内部 Banks 1：4 个内部 Banks
5:4	SDW[1:0]	SDRAM 数据总线宽度 该位指定 SDRAM 存储器数据总线宽度 00：8 位 01：16 位 10：32 位 11：保留
3:2	RAW[1:0]	行地址位宽 这些位用于指定行地址的比特宽度 00：11 位 01：12 位 10：13 位 11：保留
1:0	CAW[1:0]	列地址位宽

这些位用于指定列地址的比特宽度

00：8 位

01：9 位

10：10 位

11：11 位

SDRAM 时序寄存器 （EXMC_SDTCFGx）（x=0，1）

地址偏移：0x148+4*x，（x = 0，1）

复位值：0x0FFF FFFF

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	RCD[3:0]	行到列的延迟 这些位指定了使能命令与读/写命令之间延迟多少 SDRAM 时钟周期单元 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期
23:20	RPD[3:0]	行预充电延迟 这些位指定了预充电命令与下一个命令之间延迟多少 SDRAM 存储器时钟周期单元 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，RPD 必须用较慢设备的时序来配置
19:16	WRD[3:0]	写恢复延迟 这些位指定写命令和预充电命令之间延迟多少 SDRAM 存储器时钟周期单元 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，

		WRD 必须用较慢设备的时序来配置
15:12	ARFD[3:0]	自动刷新延迟 这些位指定两个连续的刷新命令之间的延迟，在同一个内部 bank 上两个使能命令之间的延迟，以及刷新命令和使能命令之间的延迟，延迟时间 SDRAM 存储器时钟周期为单位 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期 注意：寄存器 EXMC_SDTCFG1 相应位保留，如果两个 SDRAM 存储器都被使用，AFRD 必须用较慢设备的时序来配置
11:8	RASD[3:0]	行地址选择延迟 这些位指定了使能命令与预充电命令之间延迟多少 SDRAM 时钟周期单元，也指定了两个连续的自刷新命令之间的最小延迟 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期
7:4	XSRD[3:0]	退出自刷新延迟 这些位指定了从自刷新命令到使能命令之间延迟多少个 SDRAM 存储器时钟周期单元 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期
3:0	LMRD[3:0]	加载模式寄存器延迟 这些位指定加载模式寄存器命令与刷新或使能命令之间延迟多少 SDRAM 存储器时钟周期单元 0x0：1 个周期 0x1：2 个周期 .... 0xF：16 个周期

SDRAM 命令寄存器（EXMC_SDCMD）

地址偏移：0x150
复位值：0x0000 0000

该外设寄存器可以32位访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										MRC[12:7]					
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRC[6:0]							NARF[3:0]				DS0	DS1	CMD[2:0]		
rw							rw				rw	rw	rw		

位/位域	名称	描述
31:22	保留	必须保持复位值。
21:9	MRC[12:0]	模式寄存器内容 这些位指定 SDRAM 模式寄存器的内容，这些内容在 CMD=“100”时进行编程
8:5	NARF[3:0]	连续的自动刷新个数 这些位指定在 CMD=“011”时，发出多少个连续自动刷新周期 0x0：1 个自动刷新周期 0x1：2 个自动刷新周期 .... 0xE：15 个自动刷新周期 0xF：保留
4	DS0	选择 SDRAM device 0 该位指示 SDRAM device 0 是否被选择 0：SDRAM device 0 没有被选择 1：SDRAM device 0 被选择
3	DS1	选择 SDRAM device 1 该位指示 SDRAM device 1 是否被选择 0：SDRAM device 1 没有被选择 1：SDRAM device 1 被选择
2:0	CMD[2:0]	命令 这些位指定发送到 SDRAM 设备上的命令 000：正常操作模式 001：时钟使能命令 010：所有存储区预充电命令 011：自动刷新命令 100：加载模式寄存器命令 101：自刷新命令 110：掉电模式进入命令 111：保留 Note：发送命令时，至少需要选择一个设备（DS1 orDS0）。如果两个设备同时使用，必须同时选择两个设备发送命令。

SDRAM 自动刷新间隔寄存器（EXMC_SDARI）

地址偏移：0x154

复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	REIE	硬件强制清 0 0：中断禁止 1：状态寄存器 REIF 位置 1 发生中断
13:1	ARINTV[12:0]	自动刷新间隔 这些位指定两个连续的自动刷新命令之间间隔多少存储器时钟周期单元 ARFITV=（SDRAM 刷新周期 / 行数）-20
0	REC	清除刷新错误标志 该位置 1 会清除状态寄存器 REIF 位 0：没有效果 1：清除刷新错误标志

SDRAM 状态寄存器（EXMC_SDSTAT）

地址偏移：0x158

复位值：0x0000 0000

该外设寄存器可以32位访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										NRDY	STA1[1:0]		STA0[1:0]		REIF
										r	r		r		r

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	NRDY	非就绪状态 该位指定 SDRAM 控制器是否已经准备接收一个新的命令

		0：SDRAM 控制器准备好接收新命令 1：SDRAM 控制器没有准备好接收新命令
4:3	STA1[1:0]	Device1 状态 该位定义 SDRAM device1 的状态 00：正常状态 01：自刷新状态 10：掉电状态
2:1	STA0[1:0]	Device 0 状态 该位定义 SDRAM device 0 的状态 00：正常状态 01：自刷新状态 10：掉电状态
0	REIF	刷新错误标志 0：无刷新错误 1：出现刷新错误。若中断使能位置 1（REIE），则产生中断。

SDRAM 读采样控制寄存器（EXMC_SDRSCTL）

地址偏移：0x180

复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	SDSC[3:0]	选择读数据的采样时钟的延迟单元 0x0：0 个延迟单元 0x1：1 个延迟单元 …… 0xF：15 个延迟单元
3:2	保留	必须保持复位值。
1	SSCR	选择读数据的采样周期 0：除延迟之外，为读数据采样时钟增加 0 个额外的 HCLK 周期 1：除延迟之外，为读数据采样时钟增加 1 个额外的 HCLK 周期
0	RSEN	读采样使能

0：禁止读采样

1：使能读采样

25.4.4. SQPI-PSRAM 控制器寄存器

SPI 初始化寄存器（EXMC_SINIT）

地址偏移：0x310

复位值：0x1801 0000

该外设寄存器可以32位访问。

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

位/位域	名称	描述
31	POL	读数据时的采样极性 0：上升沿时进行采样（缺省值） 1：下降沿时进行采样
30:29	IDL[1:0]	SPI PSRAM ID 长度 00：64 位 01：32 位 10：16 位 11：8 位
28:24	ADRBIT[4:0]	SPI PSRAM 地址位数 范围由 1 到 26（缺省值为 24） 0x00：保留 0x01：1 位地址 …… 0x1A：26 位地址 0x1B：保留 …… 0x1F：保留
23:18	保留	必须保持复位值。
17:16	CMDBIT[1:0]	SPI PSRAM 命令位数 00：4 位 01：8 位（缺省值） 10：16 位

		11：保留
15:0	保留	必须保持复位值

SPI 读命令寄存器 （EXMC_SRCMD）

地址偏移：0x320
复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31	RDID	发送读 SPI PSRAM ID 的命令，命令码和模式分别通过 RCMD 和 RMODE 设置
30:22	保留	必须保持复位值。
21:20	RMODE[1:0]	SPI PSRAM 读命令模式 00：非 SPI 模式 01：SPI 模式 10：SQPI 模式 11：QPI 模式
19:16	RWAITCYCLE[3:0]	读数据时地址阶段结束后等待的周期数
15:0	RCMD[15:0]	SPI 读命令的命令码 CMDBIT 不同时，RCMD 有效位不同： CMDBIT=00，RCMD[3:0]有效 CMDBIT=01，RCMD[7:0]有效 CMDBIT=10，RCMD[15:0]有效

注意：在向 RDID 位写 1 之前，你必须确保该位已被清除，RDID 置 1 之后，必须等待 RDID 被清除

SPI 写命令寄存器（EXMC_SWCMD）

地址偏移：0x330
复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31	SC	发送 SPI PSRAM 没有地址和数据阶段的特殊命令，命令码和模式分别由 WCMD 和 WMODE 设置
30:22	保留	必须保持复位值。
21:20	WMODE[1:0]	SPI PSRAM 写命令模式 00：非 SPI 模式 01：SPI 模式 10：SQPI 模式 11：QPI 模式
19:16	WWAITCYCLE[3:0]	写数据时地址阶段结束后等待的周期数
15:0	WCMD[15:0]	SPI 写命令的命令码

注意：在向 SC 位写 1 之前，你必须确保该位已被清除，SC 置 1 之后，必须等待 SC 被清除

SPI ID 低位寄存器（EXMC_SIDL）

地址偏移：0x340

复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:0	SIDL[31:0]	ID 低位数据 当 IDL=00 或 01 时，SIDL[31:0]有效 当 IDL=10 时，SIDL[15:0]有效 当 IDL=11 时，SIDL[7:0]有效

SPI ID 高位寄存器（EXMC_SIDH）

地址偏移：0x350

复位值：0x0000 0000

该外设寄存器可以32位访问。

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

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

位/位域	名称	描述
31:0	SIDH[31:0]	ID 高位数据 仅在 IDL=00 时有效

26. 控制器局域网络（CAN）

26.1. 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 总线控制器作为 CAN 网络接口，遵循 CAN 总线协议 2.0A 和 2.0B。CAN 总线控制器可以处理总线上的数据收发，在 GD32F205 和 GD32F207 系列产品中，CAN 具有 28 个过滤器。过滤器用于筛选并接收用户需要的消息。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。

26.2. 主要特征

- 支持 CAN 总线协议 2.0A 和 2.0B；
- 通信波特率最大为 1Mbit/s；
- 支持时间触发通信；
- 中断使能和清除。

发送功能

- 3 个发送邮箱；
- 支持优先级发送；
- 支持发送时间戳。

接收功能

- 2 个深度为 3 的接收 FIFO；
- 在 GD32F205 和 GD32F207 系列产品中，具有 28 个过滤器；
- FIFO 锁定功能。

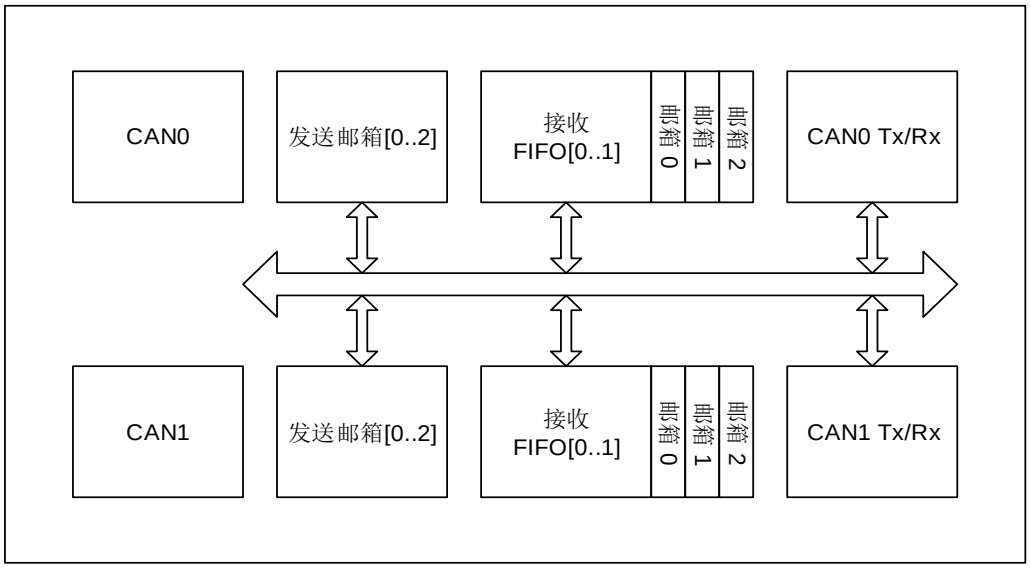
时间触发通信

- 在时间触发通信模式下取消自动重传；
- 16 位定时器；
- 接收时间戳；
- 发送时间戳。

26.3. 功能说明

CAN 模块结构框图如*图 26-1. CAN 模块结构框图*所示。

图 26-1. CAN 模块结构框图



26.3.1. 工作模式

CAN 总线控制器有 3 种工作模式：

- 睡眠工作模式；
- 初始化工作模式；
- 正常工作模式。

睡眠工作模式

芯片复位后，CAN总线控制器处于睡眠工作模式。该模式下CAN总线控制器的时钟停止工作并处于一种低功耗状态。

将CAN_CTL寄存器的SLPWMOD位置1，可以使CAN总线控制器进入睡眠工作模式。当进入睡眠工作模式后，CAN_STAT寄存器的SLPWS位将被硬件置1。

将CAN_CTL寄存器的AWU位置1，并当CAN检测到总线活动时，CAN总线控制器将自动退出睡眠工作模式。将CAN_CTL寄存器的SLPWMOD位清0，也可以退出睡眠工作模式。

由睡眠模式进入初始化工作模式：将CAN_CTL寄存器的IWMOD位置1，SLPWMOD位清0。

由睡眠模式进入正常工作模式：将CAN_CTL寄存器的IWMOD位和SLPWMOD位清0。

初始化工作模式

如果需要配置 CAN 总线通信参数，CAN 总线控制器必须进入初始化工作模式。将 CAN_CTL 寄存器的 IWMOD 位置 1，使 CAN 总线控制器进入初始化工作模式，将其清 0 则离开初始化工作模式。在进入初始化工作模式后，CAN_STAT 寄存器的 IWS 位将被硬件置 1。

由初始化模式进入睡眠模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，IWMOD 位清 0。

由初始化模式进入正常工作模式：CAN_CTL 寄存器的 SLPWMOD 位和 IWMOD 位清 0。

正常工作模式

在初始化工作模式中配置完 CAN 总线通信参数后，将 CAN_CTL 寄存器的 IWMOD 位清 0 可以进入正常工作模式并与 CAN 总线网络中的节点进行正常通信。

由正常工作模式进入睡眠工作模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，并等待当前数据收发过程结束。

由正常工作模式初始化工作模式：CAN_CTL 寄存器的 IWMOD 位置 1，并等待当前数据收发过程结束。

26.3.2. 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

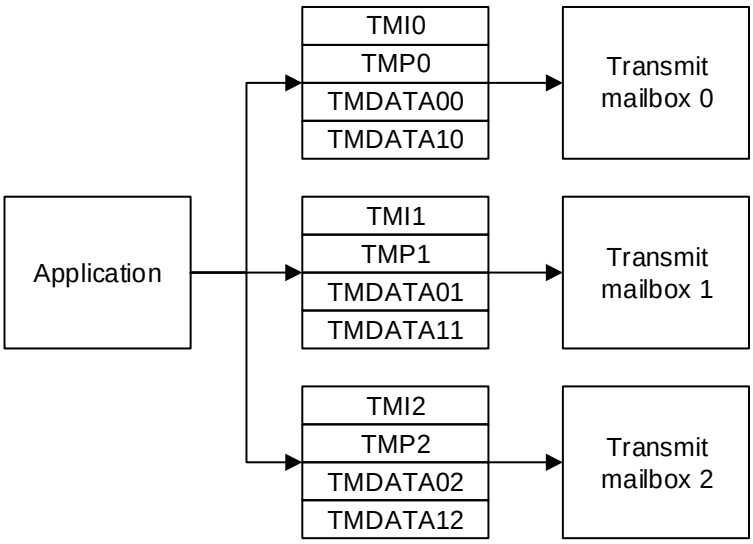
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN_BT 寄存器中的 LCMOD 和 SCMOD 清 0。

26.3.3. 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 26-2. 发送寄存器*所示。

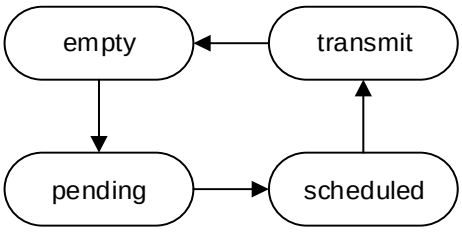
图 26-2. 发送寄存器



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 TEN 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 总线发送数据，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 26-3. 发送邮箱状态转换*所示。

图 26-3. 发送邮箱状态转换



发送状态和错误信息

CAN_TSTAT寄存器中的MTF，MTFNERR，MAL和MTE用来说明发送状态和错误信息。

- MTF：发送完成标志位。当数据发送完成时，MTF 置 1。
- MTFNERR：无错误发送完成标志位。当数据发送完成且没有错误时，MTFNERR 置 1。
- MAL：仲裁失败标志位。当发送数据过程中出现仲裁失败时，MAL置1。
- MTE：发送错误标志位。当发送过程中检测到总线错误时，MTE置1。

数据发送步骤

数据发送步骤如下：

第1步：选择一个空闲发送邮箱；
第2步：根据应用程序要求，配置4个发送寄存器；
第3步：将CAN_TMIx寄存器的TEN置1；
第4步：检测发送状态和错误信息。典型情况是检测到MTF和MTFNERR置1，说明数据被成功发送。

发送选项

中止数据发送
将CAN_TSTAT寄存器的MST置1，可以中止数据发送。
当发送邮箱处于**pending**和**scheduled**状态，CAN_TSTAT寄存器的MST置1可以立即中止数据发送。
当发送邮箱处于**transmit**状态，则面临两种情况。一种情况是数据发送被成功地完成，MTF和MTFNERR为1，这时发送邮箱将转换为**empty**状态。相对的，如果数据发送过程中出现了问题，这时发送邮箱将转换为**scheduled**状态，这时数据发送被中止。

发送优先级
当有2个及其以上发送邮箱等待发送时，寄存器CAN_CTL的TFO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。

26.3.4. 数据接收

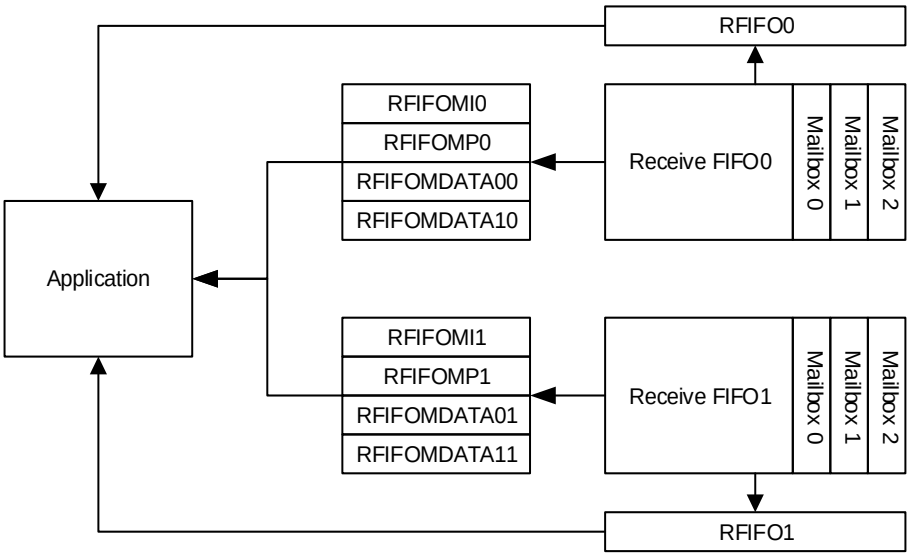
接收寄存器

应用程序通过2个深度为3的FIFO接收来自CAN网络的数据。

寄存器CAN_RFIFOx可以操作FIFO，也包含FIFO状态。寄存器CAN_RFIFOMIx，CAN_RFIFOMPx，CAN_RFIFOMDATA0x和CAN_RFIFOMDATA1x用于接收数据帧。

如*图26-4. 接收寄存器*所示。

图 26-4. 接收寄存器



接收 FIFO

每个接收FIFO包含3个接收邮箱，用来存储接收数据帧。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置1释放邮箱。

接收 FIFO 状态信息

接收FIFO状态信息包含在寄存器CAN_RFIFOx中。

RFL：FIFO中包含的帧数量。FIFO为空时，RFL为0；FIFO为满时，RFL为3。

RFF：FIFO满状态标志位。这时RFL为3。

RFO：FIFO溢出标志位。当FIFO已经包含了3个数据帧时，新的数据帧到来使FIFO发生溢出。如果CAN_CTL寄存器的RFOD位被置1，新的数据帧将丢弃。如果该位被清0，新的数据帧将覆盖接收FIFO中最后一帧数据。

数据接收步骤

第1步：查看FIFO中帧的数量；

第 2 步 ： 通 过 CAN_RFIFOMIx ， CAN_RFIFOMPx ， CAN_RFIFOMDATA0x 和 CAN_RFIFOMDATA1x读取数据；

第3步：将寄存器CAN_RFIFOx的RFDx置1释放邮箱，并且等待其由硬件自动清0。

26.3.5. 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

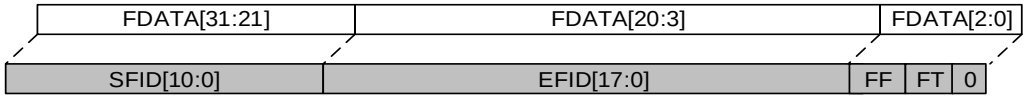
FIFO，并丢弃没有通过过滤的帧。

过滤器位宽

在GD32F205和GD32F207系列产品中，过滤器包含28个单元，它们是bank0到bank27。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。

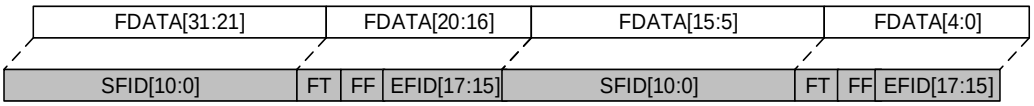
32-bit位宽CAN_FDATA包含字段：SFID[10:0]，EFID[17:0]，FF和FT。如*图26-5. 32-bit位宽过滤器*所示。

图 26-5. 32-bit 位宽过滤器



16-bit位宽CAN_FDATA包含字段：SFID[10:0]，FT，FF和EFID[17:15]。如*图26-6. 16-bit位宽过滤器*所示。

图 26-6. 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26-7. 32-bit位宽掩码模式过滤器*所示。

图 26-7. 32-bit 位宽掩码模式过滤器

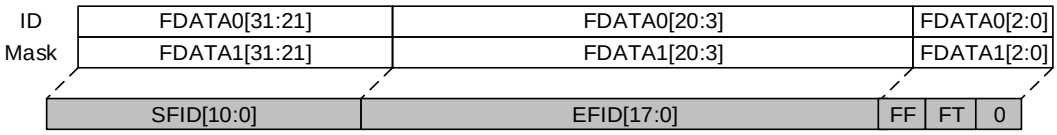
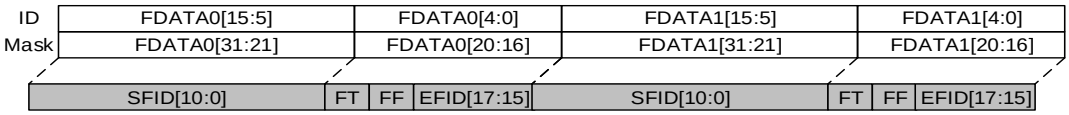


图 26-8. 16-bit 位宽掩码模式过滤器



列表模式

对于一个接收数据帧的标识符（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26-9. 32-bit位宽列表模式过滤器*所示。

图 26-9. 32-bit 位宽列表模式过滤器

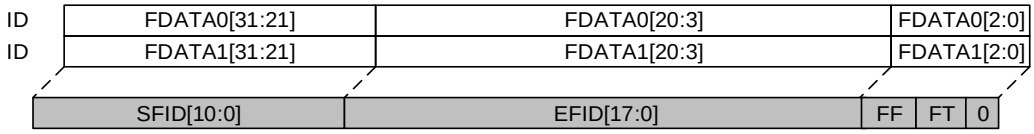
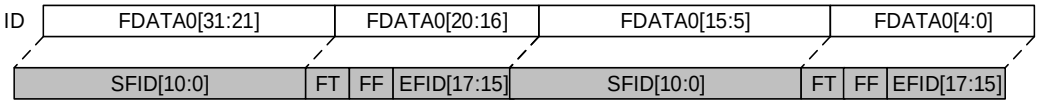


图 26-10. 16-bit 位宽列表模式过滤器



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26-1. 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 26-1. 32-bit 过滤序号

过滤单元	过滤数据寄存器	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。*表26-2. 过滤索引*是一个过滤索引的例子。

在*表26-2. 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 26-2. 过滤索引

过滤单元	FIFO0	是否激活	过滤序号	过滤单元	FIFO1	是否激活	过滤序号
0	F0DATA0-32bit-ID	Yes	0	2	F2DATA0[15:0]-16bit-ID	Yes	0
	F0DATA1-32bit-Mask				F2DATA0[31:16]-16bit-Mask		
1	F1DATA0-32bit-ID	Yes	1		F2DATA1[15:0]-16bit-ID		1
	F1DATA1-32bit-ID		2		F2DATA1[31:16]-16bit-Mask		
3	F3DATA0[15:0]-16bit-ID	No	3	4	F4DATA0-32bit-ID	No	2
	F3DATA0[31:16]-16bit-Mask				F4DATA1-32bit-Mask		
	F3DATA1[15:0]-16bit-ID		4	5	F5DATA0-32bit-ID	No	3
	F3DATA1[31:16]-16bit-Mask				F5DATA1-32bit-ID		4
7	F7DATA0[15:0]-16bit-ID	No	5	6	F6DATA0[15:0]-16bit-ID	Yes	5
	F7DATA0[31:16]-16bit-ID		6		F6DATA0[31:16]-16bit-ID		6
	F7DATA1[15:0]-16bit-ID		7		F6DATA1[15:0]-16bit-ID		7
	F7DATA1[31:16]-16bit-ID		8		F6DATA1[31:16]-16bit-ID		8
8	F8DATA0[15:0]-16bit-ID	Yes	9	10	F10DATA0[15:0]-16bit-ID	No	9
	F8DATA0[31:16]-16bit-ID		10		F10DATA0[31:16]-16bit-Mask		
	F8DATA1[15:0]-16bit-ID		11		F10DATA1[15:0]-16bit-ID		10
	F8DATA1[31:16]-16bit-ID		12		F10DATA1[31:16]-16bit-Mask		
9	F9DATA0[15:0]-16bit-ID	Yes	13	11	F11DATA0[15:0]-16bit-ID	No	11
	F9DATA0[31:16]-16bit-Mask				F11DATA0[31:16]-16bit-ID		12
	F9DATA1[15:0]-16bit-ID		14		F11DATA1[15:0]-16bit-ID		13
	F9DATA1[31:16]-16bit-Mask				F11DATA1[31:16]-16bit-ID		14
12	F12DATA0-32bit-ID	Yes	15	13	F13DATA0-32bit-ID	Yes	15
	F12DATA1-32bit-Mask				F13DATA1-32bit-ID		16

优先级

过滤器优先级顺序如下：

1. 32-bit位宽模式高于16-bit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。

26.3.6. 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。

26.3.7. 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中的MTF置1，而发送状态信息可以通过MTFNERR，MAL和MTE获得。

位时序（Bit time）

CAN协议采用位同步传输方式。这种方式不仅增大了传输容量，而且意味着需要一种复杂的位同步方法。面向字节传输的位同步方式适用于接收在每个字节前都有起始位的情况，而同步传输协议只要求数据帧的最开始有一个起始位。为保证接收器能正确读取信息，需要不断地进行重新同步。因此，在相位缓冲段采样点前面和后面都应该插入一个帧间隔。

可以通过位操作仲裁方式访问CAN总线。信号从发送器到接收器，再回到发送器必须在一个位时间内完成。为了达到同步的目的，除了相位缓冲段外，还需要一个传输延时段。在信号传输过程中，传输延时段被视为发送或接收延时。

CAN总线控制器将位时间分为3个部分。

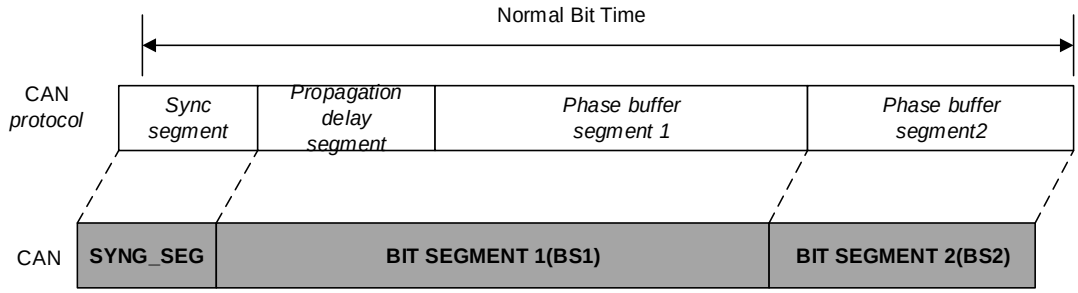
同步段（Synchronization segment），记为SYNC_SEG。该段占用1个时间单元（$1 \times t_q$）。

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26-11. 位时序*所示。

图 26-11. 位时序



再同步补偿宽度SJW（resynchronization Jump Width）对CAN网络节点同步误差进行补偿，

占用1到4个时间单元。

有效跳变定义为，在CAN控制器，没有发送显性位时，一个位时间内显性位到隐性位的第一次转变。

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。

波特率

CAN时钟从APB1总线上获得，波特率计算公式如下：

$$\text{BaudRate}=\frac{1}{\text{Normal Bit Time}} \quad (式26\text{-}1)$$

$$\text{Normal Bit Time}=t_{SYNC_SEG}+t_{BS1}+t_{BS2} \quad (式26\text{-}2)$$

其中：

$$t_{SYNC_SEG}=1\times t_q \quad (式26\text{-}3)$$

$$t_{SYNC_SEG}=1\times t_q \quad (式26\text{-}4)$$

$$t_{BS2}=(1+BT.BS2)\times t_q \quad (式26\text{-}5)$$

$$t_q=(1+BT.BAUDPSC)\times t_{PCLK1} \quad (式26\text{-}6)$$

26.3.8. 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，记为TECNT）和接收错误计数值（Receive Error Counter，记为RECNT）反映，其值会根据错误的情况由硬件增加或减少，软件可以通过这些值判断CAN网络的稳定性。关于错误计数值的详细信息请参考CAN协议相关章节。

通过使能CAN_INTEN寄存器中的相应位(ERRIE等)，软件可以在检测到错误时产生相应中断。

离线恢复

当TECNT大于255时，CAN总线控制器进入离线状态，这时寄存器CAN_ERR中的BOERR置1，并且发送和接收失效。

根据寄存器CAN_CTL中的ABOR配置，离线恢复（变为主动错误状态）有2种方式。这两种方式都要求处于离线状态的CAN总线控制器检测到CAN协议所定义的离线恢复序列（在CAN_RX检测到128次连续11个位的隐性位）时，才会自动恢复。

如果ABOR为1，将在检测到离线恢复序列后自动恢复。

如果ABOR为0，则必须先将CAN_CTL中的IWMOD置1进入初始化工作模式，然后进入正常工作模式并在检测到离线恢复序列后恢复。

26.3.9. 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

在CAN_INTEN寄存器中TMEIE置1时，发送中断可由以下条件之一产生：

- 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被置位。

26.4. CAN 寄存器

CAN0基地址：0x4000 6400

CAN1基地址：0x4000 6800

26.4.1. 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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	保留							TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMOD	IWMOD
rs								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	DFZ	调试冻结 如果 DBG_CTL0 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_CTL0 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

26.4.2. 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

26.4.3. 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 当发生仲裁失败时，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。 当发生仲裁失败时该位被置 1。
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_TIM2 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 当发生仲裁失败时，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_TIM1 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 当发生仲裁失败时，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_TIM0 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

26.4.4. 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										RFD0	RFO0	RFF0	保留	RFL0[1:0]	
										rs	rc_w1	rc_w1		r	

位/位域	名称	描述

31:6	保留	必须保持复位值。
5	RFD0	释放一次 FIFO0 中的数据 该位被置 1， 将释放 FIFO0 中的一帧数据。 FIFO 释放相应的数据空间后，该位被清 0。
4	RFO0	接收 FIFO0 溢出 当接收 FIFO0 溢出时被置位，由软件写 1 清 0。 0：接收 FIFO0 没有溢出 1：接收 FIFO0 溢出
3	RFF0	接收 FIFO0 满 当接收 FIFO0 满时被置位，由软件写 1 清 0。 0：接收 FIFO0 不满 1：接收 FIFO0 满
2	保留	必须保持复位值。
1:0	RFL0[1:0]	接收 FIFO0 中帧的数量

26.4.5. 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										RFD1	RFO1	RFF1	保留	RFL1[1:0]	
										rs	rc_w0	rc_w1		r	

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	RFD1	释放一次 FIFO1 中的数据 该位被置 1，将释放 FIFO1 中的一帧数据。 FIFO 释放相应的数据空间后，该位被清 0。
4	RFO1	接收 FIFO1 溢出 当接收 FIFO1 溢出时被置位，由软件写 0 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

26.4.6. 中断使能寄存器（CAN_INTEN）

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	保留			ERRNIE	BOIE	PERRIE	WERRIE	保留	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	SLPWIE	睡眠中断使能 0：禁用睡眠中断 1：使能睡眠中断
16	WIE	唤醒中断使能 0：禁用唤醒中断 1：使能唤醒中断
15	ERRIE	错误中断使能 0：禁用错误中断 1：使能错误中断
14:12	保留	必须保持复位值。
11	ERRNIE	错误种类中断使能 0：禁用错误种类中断 1：使能错误种类中断
10	BOIE	离线中断使能 0：禁用离线中断 1：使能离线中断
9	PERRIE	被动错误中断使能 0：禁用被动错误 1：使能被动错误
8	WERRIE	警告错误中断使能

		0：禁用警告错误中断 1：使能警告错误中断
7	保留	必须保持复位值。
6	RFOIE1	接收 FIFO1 溢出中断使能 0：禁用接收 FIFO1 溢出中断 1：使能接收 FIFO1 溢出中断
5	RFFIE1	接收 FIFO1 满中断使能 0：禁用接收 FIFO1 满中断 1：使能接收 FIFO1 满中断
4	RFNEIE1	接收 FIFO1 非空中断使能 0：禁用接收 FIFO1 非空中断 1：使能接收 FIFO1 非空中断
3	RFOIE0	接收 FIFO0 溢出中断使能 0：禁用接收 FIFO0 溢出中断 1：使能接收 FIFO0 溢出中断
2	RFFIE0	接收 FIFO0 满中断使能 0：禁用接收 FIFO0 满中断 1：使能接收 FIFO0 满中断
1	RFNEIE0	接收 FIFO0 非空中断使能 0：禁用接收 FIFO0 非空中断 1：使能接收 FIFO0 非空中断
0	TMEIE	发送邮箱空中断使能 0：禁用发送邮箱空中断 1：使能发送邮箱空中断

26.4.7. 错误寄存器（CAN_ERR）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RECNT[7:0]								TECNT[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									ERRN[2:0]			保留	BOERR	PERR	WERR
									rw				r	r	r

位/位域	名称	描述

31:24	RECNT[7:0]	接收错误计数值
23:16	TECNT[7:0]	发送错误计数值
15:7	保留	必须保持复位值。
6:4	ERRN[2:0]	错误种类 ERRN 由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ERRN 为 0。软件可以设置 ERRN 为 0b111。 000：无错误 001：填充错误 010：格式错误 011：ACK 错误 100：位隐性错 101：位显性错误 110：CRC 错误 111：软件设置值
3	保留	必须保持复位值。
2	BOERR	离线错误 当 TEC 上溢（超过 255）时，CAN 总线控制器进入离线状态，该位被置 1。
1	PERR	被动错误 当 TECNT 或者 RECNT 大于 127 时，该位由硬件置 1。
0	WERR	警告错误 当 TECNT 或 RECNT 大于等于 96 时，该位由硬件置 1。

26.4.8. 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[1:0]		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						BAUDPSC[9:0]									
										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
29:26	保留	必须保持复位值。
25:24	SJW[1:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[1:0]+1
23	保留	必须保持复位值。
22:20	BS2[2:0]	位段 2 位段 2 占用的时间单元数量=BS2[2:0]+1
19:16	BS1[3:0]	位段 1 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	BAUDPSC[9:0]	波特率分频系数

26.4.9. 发送邮箱标识符寄存器（CAN_TMIx）（x=0..2）

地址偏移：0x180，0x190，0x1A0
复位值：0xXXXX XXXX（bit0=0）

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFID[10:0]/EFID[28:18]											EFID[17:13]				
rw											rw				

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

位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]：标准格式帧标识符 EFID[28:18]：扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]：扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]：扩展格式帧标识符
2	FF	帧格式 0：标准格式帧 1：扩展格式帧
1	FT	帧种类

		0：数据帧 1：遥控帧
0	TEN	发送使能 当应用程序想要发送数据时，该位被置 1 将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清 0。 0：禁用发送 1：使能发送

26.4.10. 发送邮箱属性寄存器（CAN_TMPx）（x=0..2）

地址偏移：0x184，0x194，0x1A4
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	保留				DLENC[3:0]			
							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。
7:4	保留	必须保持复位值。
3:0	DLENC[3:0]	数据长度，DLENC[3:0]表示帧内数据长度。

26.4.11. 发送邮箱 data0 寄存器（CAN_TMDATA0x）（x=0..2）

地址偏移：0x188，0x198，0x1A8
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

26.4.12. 发送邮箱 data1 寄存器（CAN_TMDATA1x）（x=0..2）

地址偏移：0x18C，0x19C，0x1AC
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]								
			rw								rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]							DB4[7:0]								
			rw								rw				

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

26.4.13. 接收 FIFO 邮箱标识符寄存器（CAN_RFIFOMIx）（x=0,1）

地址偏移：0x1B0，0x1C0
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
SFID[10:0]/EFID[28:18]											EFID[17:13]				
					r								r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFID[12:0]													FF	FT	保留

r r r

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。

26.4.14. 接收 FIFO 邮箱属性寄存器（CAN_RFIFOMPx）（x=0,1）

地址偏移：0x1B4，0x1C4
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:16	TS[15:0]	时间戳 接收时间戳
15:8	FI[7:0]	过滤索引 帧通过过滤器时的过滤序号
7:4	保留	必须保持复位值。
3:0	DLENC[3:0]	数据长度 DLENC[3:0]表示帧内数据长度。

26.4.15. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x=0,1）

地址偏移：0x1B8，0x1C8
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

26.4.16. 接收 FIFO 邮箱 data1 寄存器（CAN_RFIFOMDATA1x）（x=0,1）

地址偏移：0x1BC，0x1CC
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

26.4.17. 过滤器控制寄存器（CAN_FCTL）（仅 CAN0 可用）

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。
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

26.4.18. 过滤器模式配置寄存器（CAN_FMCFG）（仅 CAN0 可用）

地址偏移：0x204

复位值：0x0000 0000

该寄存器只能按字（32位）访问

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

26.4.19. 过滤器位宽配置寄存器（CAN_FSCFG）（仅 CAN0 可用）

地址偏移：0x20C

复位值：0x0000 0000

该寄存器只能按字（32位）访问

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

26.4.20. 过滤器关联 FIFO 寄存器（CAN_FAFIFO）（仅 CAN0 可用）

地址偏移：0x214

复位值：0x0000 0000

该寄存器只能按字（32位）访问

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

26.4.21. 过滤器激活寄存器（CAN_FW）（仅 CAN0 可用）

地址偏移：0x21C

复位值：0x0000 0000

该寄存器只能按字（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

26.4.22. 过滤器（x）数据（y）寄存器（CAN_FxDATAy）（x=0..27, y=0,1）（仅 CAN0 可用）

地址偏移：0x240+8*x+4*y，（x=0..27，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

27. 以太网（ENET）

27.1. 简介

本章描述的以太网外设模块只适用于 GD32F20x 产品。

该以太网模块包含 10/100Mbps 以太网 MAC（媒体访问控制器），采用 DMA 优化数据帧的发送与接收性能，支持 MII（媒体独立接口）与 RMII（简化的媒体独立接口）两种与物理层(PHY)通讯的标准接口，实现以太网数据帧的发送与接收。以太网模块遵守 IEEE 802.3-2002 标准和 IEEE 1588-2002 标准。

27.2. 主要特性

MAC 特性

- 支持 10Mbps 或 100Mbps 数据传输速率；
- 支持 MII 和 RMII 接口；
- 支持调试用回环模式；
- 支持符合 CSMA/CD 协议的半双工背压通讯；
- 支持符合 IEEE 802.3x 的流控通讯。在当前帧发送完毕后，根据接收的暂停帧中暂停时间延迟发送。在全双工/半双工模式下，MAC 根据 RxFIFO 的填充程度自动发送暂停帧/背压信号；
- 支持符合 IEEE 802.3x 的全双工流控通讯，当输入流控信号失效时，自动发送零时间片暂停帧。支持符合 IEEE 802.3x 的半双工流控通讯，支持根据 RxFIFO 的填充程度（直通模式）自动发送背压信号；
- 可选择在发送操作时自动生成校验/填充位；
- 可选择在接收操作时自动去除校验/填充位；
- 帧长度可配置，支持最长为 16K 字节的标准帧；
- 帧间隙可配置（40~96 位时间，以 8 位为单位改变）；
- 支持多种模式的接收过滤；
- 支持检测接收帧的 IEEE 802.1Q VLAN 标签；
- 支持强制网络统计标准(RFC2819/RFC2665)；
- 支持两种唤醒帧检测：LAN 远程唤醒帧和 AMD 的 Magic PacketTM 帧；
- 支持对 IPv4 报头校验和以及 IPv4 或 IPv6 数据格式封装的 TCP、UDP 或 ICMP 的校验和进行检查的功能；
- 支持 IEEE 1588-2008 标准定义的以太网帧时间戳，并将其按 64 位记录于帧状态中；
- 相互独立的两个 2K 字节的 FIFO 分别用于发送与接收；
- 在延迟冲突、过度冲突、过度顺延和下溢情况下丢弃帧；
- 在存储转发模式下，可设置对发送帧，计算并插入 IPv4 的报头校验和及 TCP、UDP 或 ICMP 的校验和。

DMA 特性

- 支持环结构或链结构两种形式的描述符列表；

- 每个描述符可以传输最高为 8K 字节的数据；
- 中断可配置，适用于多种工作状态；
- 支持轮询或固定优先级两种方式仲裁 DMA 发送和接收控制器的请求。

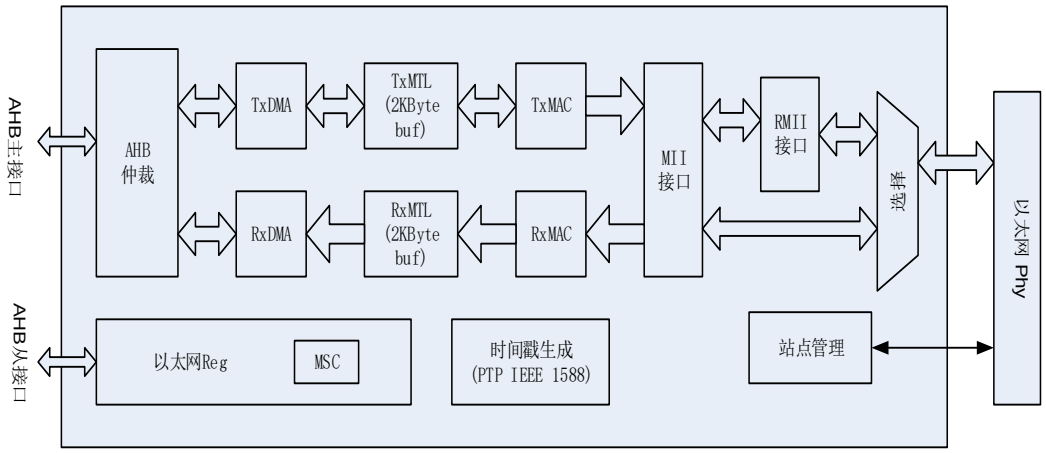
PTP特性

- 支持符合 IEEE1588 的时间同步功能；
- 支持粗/精调两种校正方法；
- 输出秒脉冲；
- 达到预设目标时间时触发中断。

27.2.1. 模块框图

以太网模块由 MAC（介质访问控制器）模块、MII/RMII 模块和一个以描述符形式控制的 DMA 模块组成。

图 27-1. 以太网模块框图



MAC 模块通过 MII 或 RMII 与片外 PHY 连接。通过对 AFIO_PCF0 寄存器的相关位进行设置，可以选择使用哪种接口。SMI（站点管理接口）用于配置和管理外部 PHY。

发送数据模块包括：

- TxDMA 控制器，用于从存储器中读取描述符和数据，以及将状态写入存储器；
- TxMTL，用于对发送数据的控制，管理和存储。TxMTL 内含 TxFIFO，用于缓存待 MAC 发送的数据；
- MAC 发送控制寄存器组，用于管理和控制数据帧的发送。

接收数据模块包括：

- RxDMA 控制器，用于从存储器中读取描述符，以及将数据与状态写入存储器；
- RxMTL，用于对接收数据的控制，管理和存储。RxMTL 实现了 RxFIFO，用于存储待转发到系统存储的帧数据；
- MAC 接收控制寄存器组，用于管理数据帧的接收和标示接收状态。MAC 内含接收过滤器，采用多种过滤机制，滤除特定的以太网帧。

注意：在使用以太网模块时，AHB 的频率应至少为 25MHz。

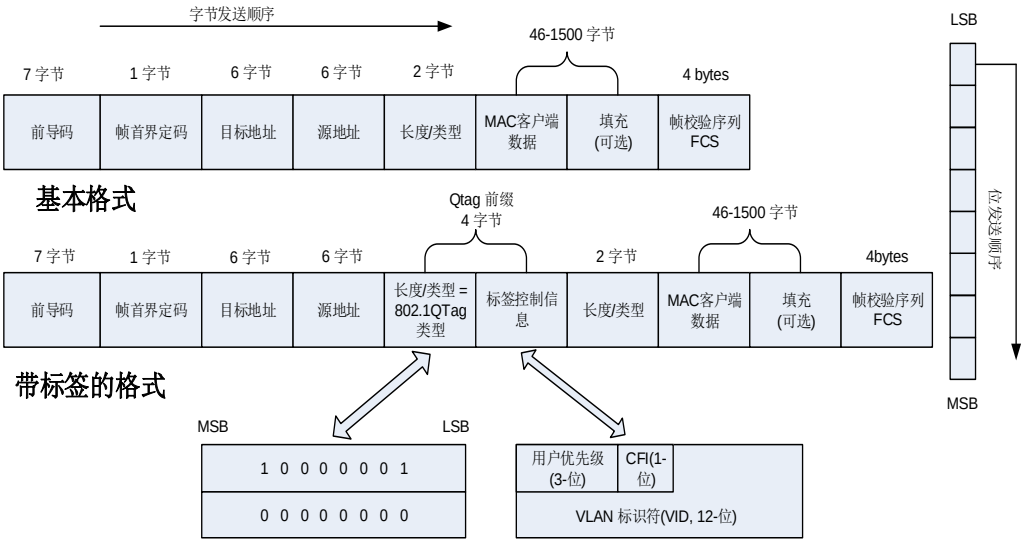
27.2.2. MAC 802.3 以太网数据包描述

MAC 的数据通信可使用两种帧格式：

- 基本的 MAC 帧格式；
- 带标签的 MAC 帧格式（对基本的 MAC 帧格式的扩展）。

*图 27-2. MAC/带标签的 MAC 帧格式*描述了帧结构（基本的和带标签的）：

图 27-2. MAC/带标签的 MAC 帧格式



注意：除了帧校验序列，以太网控制器发送每个字节时都按照低位先出的次序进行传输。

CRC 计算包括帧数据的所有字节除去前导码和帧首界定码域。以太网帧的 32 位 CRC 生成多项式为 0x04C11DB7，且此多项式用于以太网模块中所有的 32 位 CRC 计算，如下式所示：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

27.2.3. 以太网信号描述

*表 27-1. 以太网模块引脚配置*列出了 MAC 模块所用引脚在 MII/RMII 模式下默认及重映射的功能和具体配置。

表 27-1. 以太网模块引脚配置

MAC 信号	引脚	引脚配置	MII 默认	MII 重映射	RMII 默认	RMII 重映射
ETH_MDC	PC1	推挽复用输出，高速(50MHz)	MDC		MDC	
ETH_MII_TXD2	PC2	推挽复用输出，高速(50MHz)	TXD2			
ETH_MII_TX_CLK	PC3	浮空输入(复位状态)	TX_CLK			
ETH_MII_CRS	PA0	浮空输入(复位状态)	CRS			

MAC 信号	引脚	引脚配置	MII 默认	MII 重映射	RMII 默认	RMII 重映射
ETH_RX_CLK ETH_RMII_REF_CLK	PA1	浮空输入(复位状态)	RX_CLK		REF_CLK	
ETH_MDIO	PA2	推挽复用输出，高速(50MHz)	MDIO		MDIO	
ETH_MII_COL	PA3	浮空输入(复位状态)	COL			
ETH_MII_RX_DV ETH_RMII_CRS_DV	PA7	浮空输入(复位状态)	RX_DV		CRS_DV	
ETH_MII_RXD0 ETH_RMII_RXD0	PC4	浮空输入(复位状态)	RXD0		RXD0	
ETH_MII_RXD1 ETH_RMII_RXD1	PC5	浮空输入(复位状态)	RXD1		RXD1	
ETH_MII_RXD2	PB0	浮空输入(复位状态)	RXD2			
ETH_MII_RXD3	PB1	浮空输入(复位状态)	RXD3			
ETH_PPS_OUT	PB5	推挽复用输出，高速(50MHz)	PPS_OUT		PPS_OUT	
ETH_MII_TXD3	PB8	推挽复用输出，高速(50MHz)	TXD3			
ETH_MII_RX_ER	PB10	浮空输入(复位状态)	RX_ER			
ETH_MII_TX_EN ETH_RMII_TX_EN	PB11	推挽复用输出，高速(50MHz)	TX_EN		TX_EN	
ETH_MII_TXD0 ETH_RMII_TXD0	PB12	推挽复用输出，高速(50MHz)	TXD0		TXD0	
ETH_MII_TXD1 ETH_RMII_TXD1	PB13	推挽复用输出，高速(50MHz)	TXD1		TXD1	
ETH_RMII_CRS_DV	PD8	浮空输入(复位状态)		RX_DV		CRS_DV
ETH_MII_RXD0 ETH_RMII_RXD0	PD9	浮空输入(复位状态)		RXD0		RXD0
ETH_MII_RXD1 ETH_RMII_RXD1	PD10	浮空输入(复位状态)		RXD1		RXD1
ETH_MII_RXD2	PD11	浮空输入(复位状态)		RXD2		
ETH_MII_RXD3	PD12	浮空输入(复位状态)		RXD3		

27.3. 功能说明

27.3.1. 接口配置

以太网模块通过 MII/RMII 接口与片外 PHY 连接，传送与接收以太网包。MII 或 RMII 模式由软件选择并通过 SMI 接口对 PHY 进行管理。

SMI：站点管理接口

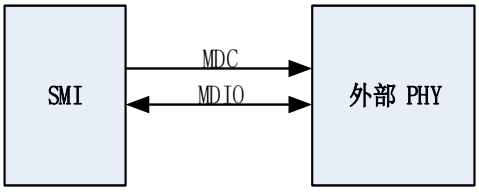
SMI 用于访问和设置 PHY 的配置。

站点管理接口(SMI)通过 MDC 时钟线与 MDIO 数据线与外部 PHY 通讯，可以通过其访问任意 PHY 的任意寄存器。SMI 接口可以支持最多 32 个 PHY，但在任意时刻只能访问一个 PHY 的一个寄存器。

MDC 时钟线和 MDIO 数据线具体作用如下：

- MDC：最高频率为 2.5MHz 的时钟信号，在空闲状态下该引脚保持为低电平状态。在传输数据时该信号的高电平和低电平的最短保持时间为 160ns，信号的最小周期为 400ns；
- MDIO：用于与 PHY 之间的数据传输，与 MDC 时钟线配合，接收/发送数据。

图 27-3. 站点管理接口信号



SMI 写操作

应用程序需将要传输的数据写入 ENET_MAC_PHY_DATA 寄存器中，并对 ENET_MAC_PHY_CTL 寄存器相关位进行操作：

1) 设置 PHY 设备地址和将要操作的 PHY 寄存器地址，并将 PW 位置为 1，使能写模式；
2) 将 PB 位置 1 开始传输。在传输过程中 PB 位一直为高，直到传输完成，硬件将会自动清除 PB 位。

应用程序可以通过 PB 位判断传输是否完成。在 PB 位置 1 期间，由于操作正在运行，因此不能修改 PHY 控制寄存器和 PHY 数据寄存器的内容。在将 PB 位置位之前，应用程序必须确保该位读出为'0'。

SMI 读操作

应用程序对 ENET_MAC_PHY_CTL 寄存器相关位进行操作：

1) 设置 PHY 设备地址和将要操作的 PHY 寄存器地址，并将 PW 位置为 0，使能读模式；
2) 将 PB 位置 1 开始数据接收。在接收过程中 PB 位一直为高，直到接收完成，硬件将会自动清除 PB 位。

应用程序可以通过 PB 位判断传输是否完成。在 PB 位置 1 期间，由于操作正在运行，因此不能修改 PHY 控制寄存器和 PHY 数据寄存器的内容。在将 PB 位置位之前，应用程序必须确保该位读出为'0'。

注意：由于 PHY 寄存器地址 16-31 的寄存器内容由各厂商自由定义，所以在访问不同 PHY 设备的这部分寄存器时，需要根据厂商手册对应用程序进行不同的设置。GD32E50x 固件库当前支持的 PHY 设备详情请参考固件库相关手册说明。

SMI 时钟选择

SMI 接口的时钟源由 AHB 时钟分频得到。为了保证 MDC 时钟频率不超过 2.5MHz，需根据 AHB 时钟频率对 PHY 控制寄存器中相关位进行设置，选择合适的分频系数。*表 27-2. 时钟范围*列出了对应 AHB 时钟范围的分频系数的选择。

表 27-2. 时钟范围

AHB 时钟	MDC 时钟	选择位
150~180MHz	AHB clock/102	0x4
35~60MHz	AHB clock/26	0x3
20~35MHz	AHB clock/16	0x2
100~180 MHz	AHB clock/62	0x1
60~100MHz	AHB clock/42	0x0

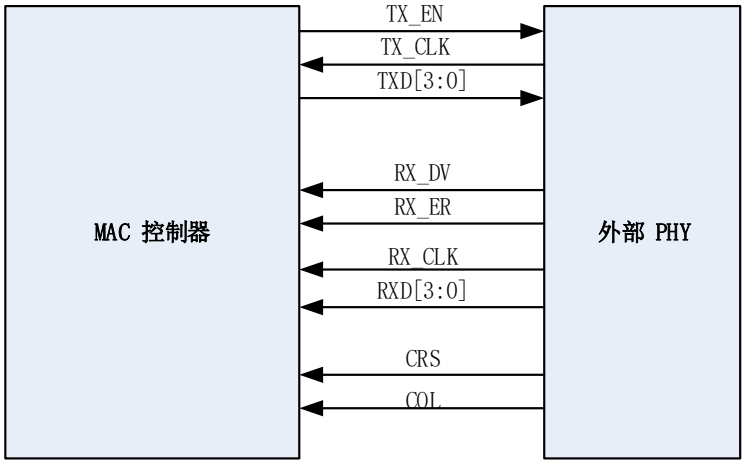
MII/RMII 的选择

当以太网控制器处于复位状态或者在使能时钟前，通过对 AFIO_PCF0 寄存器的 ENET_PHY_SEL 位进行配置，选择使用 MII 或者 RMII 模式，默认为 MII 模式。

MII：媒体独立接口

媒体独立接口(MII)用于 MAC 与外部 PHY 互联，支持 10Mbit/s 和 100Mbit/s 的数据传输模式。

图 27-4. 媒体独立接口(MII)信号线



- **MII_TX_CLK**：发送数据使用的时钟信号，对于 10Mbit/s 的数据传输，此时钟为 2.5MHz，对于 100M 位/s 的数据传输，此时钟为 25MHz；

- **MII_RX_CLK**：接收数据使用的时钟信号，对于 10Mbit/s 的数据传输，此时钟为 2.5MHz，对于 100Mbit/s 的数据传输，此时钟为 25MHz；

- **MII_TX_EN**：发送使能信号，此信号必须与数据前导符的起始位同步出现，并在传输完毕前一直保持；

- **MII_TXD[3:0]**：发送数据线，每次传输 4 位数据，数据在 MII_TX_EN 信号有效时有效。

MII_TXD[0]是数据的最低有效位，MII_TXD[3]是最高有效位。当 MII_TX_EN 信号无效时，PHY 忽略传输的数据；

- **MII_CRS**：载波侦听信号，仅工作在半双工模式下，由 PHY 控制。当发送或接收介质非空闲时，此信号有效。PHY 必需保证 MII_CRS 信号在发生冲突的整个时间段内都保持有效。此信号不需要与发送/接收的时钟同步；

- **MII_COL**：冲突检测信号，仅工作在半双工模式下，由 PHY 控制。当检测到介质发生冲突时，此信号有效，并且在整个冲突的持续时间内，保持此信号有效。此信号不需要与发送/接收的时钟同步；

- **MII_RXD[3:0]**：接收数据线，每次接收 4 位数据，数据在 MII_RX_DV 信号有效时有效。MII_RXD[0]是数据的最低位，MII_RXD[3]是最高位。当 MII_RX_DV 无效，而 MII_RX_ER 有效时，MII_RXD[3:0]数据值代表特定的信息（请参考*表 27-3. 接收接口信号编码*）；

- **MII_RX_DV**：接收数据使能信号，由 PHY 控制，当 PHY 准备好数据供 MAC 接收时，该信号有效。此信号必须和帧数据的第一个 4 位同步出现，并保持有效直到数据传输完成。在传送最后 4 位数据后的第一个时钟之前，此信号必须变为无效状态。为了正确地接收帧，有效电平不能滞后于数据线上的帧首界定码出现；

- **MII_RX_ER**：接收出错信号，保持一个或多个时钟周期(MII_RX_CLK)的有效状态，表明 MAC 在接收过程中检测到错误。具体错误原因需结合 MII_RX_DV 的状态及 MII_RXD[3:0]的数据值，详见*表 27-3. 接收接口信号编码*。

表 27-3. 接收接口信号编码

MII_RX_ER	MII_RX_DV	MII_RXD[3:0]	说明
0	0	0000 to 1111	正常的帧间隔
0	1	0000 to 1111	正常的数据接收
1	0	0000	正常的帧间隔
1	0	0001 to 1101	保留
1	0	1110	载波错误指示
1	0	1111	保留
1	1	0000 to 1111	数据接收出错

MII 时钟源

为了产生 TX_CLK 和 RX_CLK 时钟信号，外部 PHY 模块必需有来自外部的 25MHz 时钟驱动。该时钟不需要与 MAC 时钟相同。可以使用外部的 25MHz 晶振或者微控制器的时钟输出引脚 CK_OUT0 提供这一时钟。当时钟来源为 CK_OUT0 引脚时需配置合适的 PLL，保证 CK_OUT0 引脚输出的时钟为 25MHz。

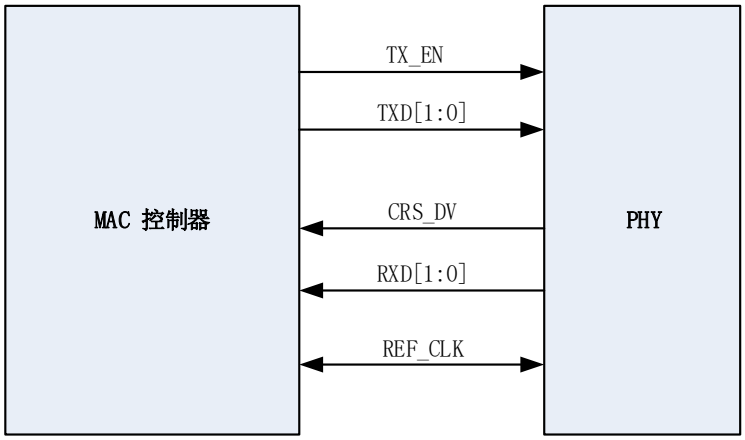
RMII：精简媒体独立接口

精简媒体独立接口(RMII)规范减少了以太网通信所需要的引脚数。根据 IEEE 802.3 标准，MII 接口需要 16 个引脚用于数据和控制信号，而 RMII 标准则将引脚数减少到了 7 个。

RMII 具有以下特性：

- 只有一个时钟信号，且该时钟信号需要提高到 50MHz
- MAC 和外部的以太网 PHY 需要使用同样的时钟源
- 使用 2 位宽度的数据收发

图 27-5. 精简媒体独立接口(RMII)信号线



MII/RMII 位传输顺序

不论选择的是 MII 还是 RMII 接口，发送接收的次序都是低位先出。

MII 和 RMII 之间的区别主要是数据位数和发送次数的不同。MII 上是先发送/接收低 4 位数据，再发送/接收高 4 位。在 RMII 上则是先发送/接收最低 2 位数据，再次低 2 位数据，次高 2 位数据，和最高 2 位数据。

例如：一个字节数据为 10011101b（从左到右顺序：高位到低位）

使用 MII 发送需 2 个时钟周期：1101 -> 1001（从左到右顺序：高位到低位）

使用 RMII 发送需 4 个时钟周期：01 -> 11 -> 01 -> 10（从左到右顺序：高位到低位）

RMII 时钟源

通过将相同的时钟源接到 MAC 和以太网 PHY 的 REF_CLK 引脚保证两者时钟源的同步。可以通过外部的 50MHz 信号或者微控制器的 CK_OUT0 引脚提供这一时钟。当时钟来源 CK_OUT0 引脚时需配置合适的 PLL，保证 CK_OUT0 引脚输出的时钟为 50MHz。

27.3.2. MAC 功能简介

MAC 模块能够实现以下功能：

数据封装（传送和接收）

- 帧检测/解码、帧边界定界；
- 寻址（管理源地址和目标地址）；
- 错误检测。

介质访问管理（半双工模式下）

- 介质分配（防止冲突）；
- 冲突解决（处理冲突）。

MAC 模块可以在两种模式下工作：

- 半双工模式：通过 CSMA/CD 算法来抢占对物理介质的访问，在同一时间只有一个传输方向的两个站点有效。
- 全双工模式：满足以下条件时，可同时进行收发而不发生冲突：1）物理介质支持同时进行收发操作。2）只有两个站点接入 LAN，且两个站点都配置为全双工模式。

MAC 的发送流程

所有的发送均由以太网模块中专用 DMA 控制器和 MAC 控制。在收到应用程序发送指令后，DMA 将发送帧从系统存储区读出并存入深度为 2K 的 TxFIFO 中，之后根据选择的模式（直通或者存储转发模式，具体定义请查看下段）将数据取出到 MAC 控制器，通过 MII/RMII 接口发送到以太网 PHY，并可以选择配置使 MAC 控制器自动将硬件计算的 CRC 值添加到数据帧的帧校验序列中。当 MAC 控制器收到来自 TxFIFO 的帧结束信号后，完成整个传输过程。传输完毕后，传输状态信息将会由 MAC 控制器生成并写回到 DMA 控制器中，应用程序可以通过 DMA 当前发送描述符查询发送状态。

TxFIFO 取出数据到 MAC 控制器的操作有两种模式：

- 在直通模式（Cut-Through）下，当 FIFO 中的数据等于或超过了所设置的阈值时（或者在达到阈值之前写入了 EOF），数据会从 FIFO 中取出并送入到 MAC 控制器中。这个阈值可以通过 ENET_DMA_CTL 寄存器的 TTHC 位来设置；
- 在存储转发模式（Store-and-Forward）下，只有当一个完整的帧写入 FIFO 之后，FIFO 中的数据才会被送入 MAC 控制器。但还有一种情况下，帧没有被完整写入 FIFO，FIFO 也会取出数据。这种情况为 TxFIFO 的大小小于要发送的以太网帧长度，那么在 TxFIFO 即将全满时，数据会被送入到 MAC 控制器。

特殊情况处理

在传输过程中，如果空闲的 DMA 发送描述符不足，或者误操作了 ENET_DMA_CTL 的 FTF 位清空了 FIFO（此位置 1 时将清空 TxFIFO 中的数据并将 FIFO 的指针复位，清空操作完成后由硬件将此位清零），则将导致不能及时连续的发送数据，此时 MAC 控制器会标识数据下溢状态。对于只收到帧起始信号却没有收到帧结束信号的情况，MAC 会忽略第二帧数据的帧起始，而将第二帧作为前一帧的延续。

若被发送的一帧占用两个 DMA 发送描述符，则第一个描述符的首段位(FSG)和末段位(LSG)应为 10b，第二个描述符的应为 01b。若第一个描述符与第二个描述符的 FSG 位都置位了，且第一个描述符的 LSG 位复位了，则将忽略第二个描述符的 FSG 位，并认为这两个描述符为只发送一个帧。

若发送 MAC 帧的数据域长度小于 46 或者带标签的 MAC 帧的数据域长度小于 42，可以选择配置 MAC 控制器自动填充内容为 0 的数据，使帧数据域的长度符合 IEEE 802.3 规范的相关定义。若执行了自动填充 0 功能，则 MAC 将忽略 DMA 描述符 DCRC 位的配置，自动计算并添加 CRC 值到帧的帧校验序列中。

MAC 的发送管理

Jabber 定时器

为了防止出现一个站点长时间占用 PHY 的情况，以太网内置的 Jabber 定时器会在以太网帧发送超过 2048 字节后终止发送。默认情况下，Jabber 定时器是使能的，因此当以太网帧发送超过 2048 字节，则 MAC 将只发送 2048 字节，并丢弃剩余的帧数据。

冲突处理机制：重发

在半双工模式下，MAC 发送数据帧时可能会发生冲突。当发生冲突事件的时候如果 FIFO 中只有不超过 96 个字节的帧数据被取出到了 MAC 中，那么帧重发功能将被激活。重发功能激活后，MAC 会中止当前的传输，然后重新从 FIFO 中读取数据并发送。当发生冲突事件的时候如果已有超过 96 个字节的数据从 FIFO 中取出到 MAC 中，那么 MAC 会中止当前的传输但不会激活重发功能，然后在描述符中置位 LCO 以通知应用程序。

发送状态信息字

发送状态信息字包括了许多用于应用程序的发送状态标志，在 MAC 控制器完成帧的发送后，将会更新发送状态信息。如果使能了 IEEE 1588 时间戳功能，时间戳的值将会随发送状态信息一起写回到发送描述符中。

清空 TxFIFO 操作

将 ENET_DMA_CTL 寄存器的 FTF 位（位 20）置 1 将清空 TxFIFO，并将 FIFO 数据指针复位。无论 TxFIFO 是否正在取出数据到 MAC 中，清空操作都会立刻执行。因此这也将导致 MAC 控制器产生数据下溢事件，并终止发送当前帧，同时返回该帧的状态信息和发送状态信息字到应用程序。并标记数据下溢位和清空位（TDES0 的位 1 和位 13）。在应用程序(DMA)接收到所有被清空帧的状态信息字以后，清空操作完成。清空操作完成后，ENET_DMA_CTL 寄存器的 FTF 位将自动清'0'。当收到清空操作指令，所有从 FIFO 取出到 MAC 的数据都将被丢弃，直到收到 FSG 位为 1 的描述符。

发送流控

MAC 控制器主要通过背压（半双工模式）和暂停控制帧（全双工模式）来管理帧的发送流控。

- 半双工模式流控：背压

当 MAC 采用半双工模式进行通讯时，如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，有两种情况可以触发背压流控。背压流控是通过发送一个 32 位的堵塞信号 0x5555 5555，通知所有其他站点发生了冲突。两种触发情况中，第一种是通过置位 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位来使能发送流控。第二种情况在接收帧时发生，MAC 在接收帧的过程中，RxFIFO 中字节数不断增大，当接收数目超过流控激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将置位背压挂起标志。若背压挂起标志置位了，且又有新的帧到来，MAC 将发送堵塞信号以延迟一段背压时间再接收帧。在背压时间结束后，PHY 会重新发送这个新的帧。若在背压期间，RxFIFO 中字节数大于等于流控失活阈

值（ENET_MAC_FCTH 寄存器中的 RFD 位），则 MAC 会再次发送背压信号；反之，则 MAC 将复位背压挂起标志，并可以接收新的帧，不再发送堵塞信号。

■ 全双工模式流控：暂停帧

对于全双工模式，MAC 控制器使用“暂停帧”进行流控制。这种方式可以使接收端能够命令发送端暂停一段时间再发送，如当接收缓冲区快要溢出的情况。如果设置了发送流控使能位(ENET_MAC_FCTL 寄存器的 TFCEN 位)，在全双工模式下，MAC 会在以下两种情况下产生并发送暂停帧。两种情况分别为：

1) 应用程序把 ENET_MAC_FCTL 寄存器的 FLCB/BKPA 位置位，将立即发送一个暂停帧。这个暂停帧指定的暂停时间为ENET_MAC_FCTL 寄存器中PTM位配置好的暂停时间值。如果应用程序前面要求了一段时间的暂停，但在这段时间内，应用程序准备好了，可以不需要剩余的暂停时间了，这时应用程序需要发一个零时间片暂停帧来通知发送方可以继续发送了。零时间片暂停帧是通过设置 ENET_MAC_FCTL 寄存器中的 PTM 位为 0，并将 FLCB/BKPA 位置位来发送的；
2) 在 RxFIFO 满足一定的条件下，MAC 会自动发送暂停帧。在接收过程中，RxFIFO 不停地有数据进来，同时 RxFIFO 也取出数据给 RxDMA，如果 RxFIFO 取出数据的频率小于其接收数据的频率，RxFIFO 中的数据就会越来越多。一旦 RxFIFO 中的数据量超过了流控的激活阈值（ENET_MAC_FCTH 寄存器中的 RFA 位），MAC 将发送一个暂停时间为 PTM 位定义的值的暂停帧。发送暂停帧之后，MAC 将启动一个计数器，计数器的时间由 ENET_MAC_FCTL 寄存器的 PLTS 位定义，当到了计数器规定的时间，MAC 将重新检查 RxFIFO。此时若 RxFIFO 中的数据量仍然大于流控激活阈值，MAC 将再次发送一个暂停帧。若 RxFIFO 中的数据量小于流控失活阈值，并且 ENET_MAC_FCTL 寄存器中的 DZQP 位被复位，则 MAC 将发送一个零时间片暂停帧。这个零时间片暂停帧用于指示远程站点结束暂停，本地缓存区已经准备好接收新的数据帧。

帧间隔管理

MAC 管理两个帧之间的时间间隔。两个帧之间的时间间隔称为帧间隙时间。在全双工模式下，在完成帧发送后，或者 MAC 进入空闲状态时，帧间隙计数器开始计数。如果在帧间隙时间未到达 ENET_MAC_CFG 寄存器中 IGBS 位所配置的值时，来了新的发送帧，则这个发送帧将被延迟发送直到达到帧间隙时间值。若这个新的发送帧在帧间隙时间之后到达，则会立即发送该帧。在半双工模式下，MAC 遵循截断二进制指数退让算法，简要来说，就是在前一个发送帧发送完成之后，或者 MAC 进入空闲状态时，帧间隙计数器开始计数。在帧间隙时间内，可能会有 3 种情况会发生：

1) 如果在帧间隙时间的前 2/3 时间检测到载波信号，帧间隙计数器将复位并重新计数；
2) 如果在帧间隙时间的后 1/3 时间里检测到载波信号，帧间隙计数器不会复位，将继续计数，当帧间隙时间到达后，MAC 发送新的帧；
3) 如果在整个帧间隙时间内都没有检测到载波信号，则在到达帧间隙时间后停止帧时隙计数器，并在之前有帧被延迟的情况下立即发送新的帧。

发送校验和模块

以太网控制器具有发送校验和的功能，支持计算校验和，并在发送时插入计算结果，以及在接

收时侦测校验和错误。本节描述了发送帧的校验和模块的操作功能。

注意：只有将 ENET_DMA_CTL 寄存器的 TSFD 位置为'1'（TxFIFO 设置成存储转发模式），同时必须保证 TxFIFO 的深度足够容纳将要发送的完整帧时，才能使能此功能。若 FIFO 深度小于帧长度，则仅仅计算和插入 IPv4 报头的校验和域。

欲了解 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 报头的规范，请分别查阅 IETF 规范 RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

■ IP 头校验和

若以太网帧的类型域值为 0x0800 同时 IP 数据包的版本域值为 0x4，则校验和模块标记其为 IPv4 数据包并会用计算结果取代帧的校验和域的内容。IPv6 的报头不包含校验和域，因此校验和模块不会改变 IPv6 报头的值。IP 头校验和计算完毕之后，其结果会写到 IPHE 位（TDES0 的位 16）。当发生下述情况时，IPHE 错误状态位会被硬件置'1'：

1) 对于IPv4数据帧：

   a) 接收到的以太网类型域值为0x0800，但IP报头版本域的值不等于0x4；
   b) IPv4报头长度域的值大于帧的总长度；
   c) IPv4报头长度域值小于IP报头总长0x5（20字节）。

2) 对于IPv6数据帧：

   a) 接收到的以太网类型域值为0x86dd，但IP报头版本域的值不等于0x6；
   b) 帧在完整接收IPv6报头（40字节）或者扩展报头（扩展报头包含报头长度域）之前结束。

■ TCP/UDP/ICMP 校验和

若以太网帧的类型域值为 0x0800 同时 IP 数据包的版本域值为 0x4，则校验和模块标记其为 IPv4 数据包并会用计算结果取代帧的校验和域的内容。IPv6 的报头不包含校验和域，因此校验和模块不会改变 IPv6 报头的值。IP 头校验和计算完毕之后，其结果会写到 IPHE 位（TDES0 的位 16）。当发生下述情况时，IPHE 错误状态位会被硬件置'1'：

1) 对于IPv4数据帧：
   a) 接收到的以太网类型域值为0x0800，但IP报头版本域的值不等于0x4；
   b) IPv4报头长度域的值大于帧的总长度；
   c) IPv4报头长度域值小于IP报头总长0x5（20字节）。

2) 对于IPv6数据帧：
   a) 接收到的以太网类型域值为0x86dd，但IP报头版本域的值不等于0x6；
   b) 帧在完整接收IPv6报头（40字节）或者扩展报头（扩展报头包含报头长度域）之前结束。

■ TCP/UDP/ICMP校验和

校验和模块通过分析IPv4或IPv6报头(包括扩展报头)来判断帧的类型（TCP、UDP或ICMP）。

当帧发生以下情况时，将绕过校验和功能，校验和模块不对这些帧进行处理：

1) 不完整的IPv4或IPv6帧；

2) 包含安全功能的IP帧(如验证报头或者封装有安全数据)；
3) 非TCP/UDP/ICMPv4/ICMPv6数据的IP帧；
4) 带路由报头的IPv6帧。

校验和模块会对TCP、UDP或者ICMP的数据进行计算，并插入报头的相应域。它有以下2种工作模式：

1) 校验和计算不包括TCP、UDP或者ICMPv6的伪首部。并假定输入帧的校验和字段已有值。校验和字段包含在校验和计算中，在计算完成后插入并替换原校验和域的值；
2) 校验和计算包括TCP、UDP或者ICMPv6的伪首部。将传输帧的校验和字段清零。进行校验和的计算，计算完成后插入传输帧的原校验和域。

校验和计算完毕之后，其结果会写到IPPE位（TDES0的位12）。当发生下述情况时，IPPE错误状态位会被硬件置'1'：

1) 在存储转发模式下，帧未被完整写入FIFO之前就被转发给MAC控制器；
2) 帧已发送完毕，但MAC从FIFO中取出的数据包字节数小于IP报头中数据长度域标明的字节数。

如果数据包长度大于标明的长度，不会报告错误，之后的数据会被当成填充字节而丢弃。如果检测到第一类错误情况，校验和的值不会插入TCP、UDP或者ICMP报头。如果检测到第二类错误情况，仍然会把校验和计算结果插入报头的相应域。

注意：无论采用哪种模式，对于IPv4上的ICMP数据包，由于这类数据包没有定义伪报头，为正确计算其校验和，校验和域内容必须为0x0000。

MAC 接收过滤器

MAC 过滤分为错误过滤（诸如过短帧、CRC 错误以及坏帧的过滤）和地址过滤。此部分主要讨论地址过滤。

地址过滤

地址过滤利用静态物理地址（MAC 地址）过滤和多播 HASH 列表过滤实现。若 ENET_MAC_FRMF 过滤器寄存器的 FAR 位为'0'（默认值），则只有通过地址过滤的帧才会被接收。该功能会根据应用程序设定的参数(帧过滤器寄存器)对单播帧或多播帧的目的与/或源地址进行过滤（通过目标地址的 I/G 位可判断是单播帧还是多播帧）并报告相应的地址过滤结果，所有不能通过过滤器的帧将被丢弃。

注意：若 ENET_MAC_FRMF 过滤器寄存器的 FAR 位为'1'，则所有帧都会被接收。在这种情况下，帧过滤结果仍会更新到接收描述符中，但帧过滤结果不会影响到帧是否会被过滤。

单播目标地址过滤器

通过对 ENET_MAC_FRMF 寄存器 HUF 位的设置，可以选择使用静态物理地址（HUF 位为'0'）或者 HASH 列表（HUF 位为'1'）的方式实现单播过滤。

■ 静态物理地址过滤

MAC 控制器支持多达 4 个 MAC 地址对单播地址进行完美过滤。在这种方式下，MAC 会把接收到帧的 6 个字节单播地址与设好的 MAC 地址寄存器逐位比较，检查是否相符。对于 MAC 地址 0 寄存器始终使能,对于 MAC 地址 1-MAC 地址 3 寄存器分别有对应的使能位。MAC 地址 1-MAC 地址 3 寄存器的每一个字节都可以通过相应 MAC 地址的高寄存器的屏蔽字节控制位（MB 位）来设置是否与接收帧的目标地址相应字节比较。

- HASH 列表过滤

这种过滤使用一种 HASH 机制。MAC 利用 64 位的 HASH 列表对单播地址进行不完美过滤。这种过滤模式遵循以下两个过滤步骤：

1) MAC计算接收帧的目标地址的CRC值；
2) 取CRC计算结果高6位作为索引检索HASH列表。如果CRC值对应的HASH列表上的相应位为'1'，则该帧能通过HASH过滤器，反之则该帧不能通过HASH过滤器。

这种类型过滤器的优点是可以仅用一个小表就覆盖任何可能的地址。缺点是过滤不完全，即有时应该丢弃的帧也会被接收。

多播目标地址过滤器

将帧过滤寄存器 ENET_MAC_FRMF 的 MFD 位清零，可以开启 MAC 多播地址过滤功能。此时根据帧过滤寄存器 ENET_MAC_FRMF 的 HMF 位的取值可以选择类似于单播目标地址过滤的两种方式进行地址过滤。

HASH 或者完美地址过滤器

设置帧过滤器寄存器 ENET_MAC_FRMF 的 HPFLT 位为'1'，并设置相应的 HUF 位（对单播帧）或者 HMF 位（对多播帧）为'1'，则可以将过滤器配置成只要接收帧的目标地址匹配 HASH 过滤器或者物理地址过滤器之一，就令帧通过。

广播地址过滤器

默认情况下，MAC 无条件的接收任何广播帧。但当设置帧过滤寄存器 ENET_MAC_FRMF 的 BFRMD 位为'1'时，MAC 将丢弃接收到的所有广播帧。

单播源地址过滤器

使能 MAC 地址 1-MAC 地址 3 寄存器，并设置其对应 MAC 地址高寄存器的 SAF 位为'1'，MAC 可以将 MAC 地址 1-MAC 地址 3 寄存器中设置的物理(MAC)地址与接收帧的源地址进行比较并过滤。MAC 也支持对源地址的成组过滤。若设置帧过滤寄存器 ENET_MAC_FRMF 的 SAFLT 位为'1'，MAC 会丢弃没能通过源地址过滤的帧，同时过滤结果会通过 DMA 接收描述符的 RDES0 的 SAFF 位反映出来。当 SAFLT 位为'1'的同时，目标地址过滤器也在工作，此时 MAC 控制器以两个滤波器结果的逻辑"与"形式判定帧是否通过。这意味着，只要帧没能通过其中一个过滤器，就会被丢弃。MAC 只会把通过全部过滤器的帧转发给应用程序。

逆转过滤操作

无论是目标地址过滤还是源地址过滤，都能在过滤器输出端逆转过滤结果。即地址与过滤器匹

配时，帧不通过；不匹配时帧通过。通过设置帧过滤寄存器 ENET_MAC_FRMF 的 DAIFLT 位和 SAIFLT 位为'1'可以实现这一功能。DAIFLT 位作用于单播和多播帧的目标地址的过滤结果，SAIFLT 位作用于单播和多播帧的源地址的过滤结果。

下面*表 27-4. 目标地址过滤器结果列表*和*表 27-5. 源地址过滤器结果列表*总结了目标地址和源地址过滤器在不同设置下的工作状态。

表 27-4. 目标地址过滤器结果列表

帧类型	PM	HPFLT	HUF	DAIFLT	HMF	MFD	BFRMD	目标地址过滤器操作
广播帧	1	-	-	-	-	-	-	通过
	0	-	-	-	-	-	0	通过
	0	-	-	-	-	-	1	不通过
单播帧	1	-	-	-	-	-	-	所有帧通过
	0	-	0	0	-	-	-	匹配完美/组过滤器时通过
	0	-	0	1	-	-	-	匹配完美/组过滤器时不通过
	0	0	1	0	-	-	-	匹配HASH过滤器时通过
	0	0	1	1	-	-	-	匹配HASH过滤器时不通过
	0	1	1	0	-	-	-	匹配HASH或者完美/组过滤器时通过
	0	1	1	1	-	-	-	匹配HASH或者完美/组过滤器时不通过
多播帧	1	-	-	-	-	-	-	所有帧通过
	-	-	-	-	-	1	-	所有帧通过
	0	-	-	0	0	0	-	匹配完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	0	1	0	-	匹配HASH过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	1	-	0	1	0	-	匹配HASH或者完美/组过滤器时通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	-	-	1	0	0	-	匹配完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	0	-	1	1	0	-	匹配HASH过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧
	0	1	-	1	1	0	-	匹配HASH或者完美/组过滤器时不通过，如果PCFRM = 0x，丢弃暂停控制帧

表 27-5. 源地址过滤器结果列表

帧类型	PM	SAIFLT	SAFLT	源地址过滤器操作
单播帧	1	-	-	所有帧通过
	0	0	0	匹配完美/组过滤器时返回通过状态，不匹配时状态为不通过，但不丢弃不通过的帧
	0	1	0	匹配完美/组过滤器时返回不通过状态，但不丢弃帧
	0	0	1	匹配完美/组过滤器时通过，丢弃不通过的帧
	0	1	1	匹配完美/组过滤器时不通过，丢弃不通过的帧

混杂模式

若设置 ENET_MAC_FRMF 寄存器的 PM 位为'1'将使能混杂模式，此时地址过滤器无效，所有帧均可通过过滤器。同时接收状态信息的目标地址/源地址错误位总是为'0'。

暂停控制帧过滤

MAC 会检测接收到的控制帧内的 6 字节目标地址域，若 ENET_MAC_FCTL 寄存器的 UPFDT 位设为 0，则判断目标地址域的值是否符合 IEEE 802.3 规范控制帧的唯一值(0x0180 C200 0001)。若 ENET_MAC_FCTL 寄存器的 UPFDT 位设为 1，则在与 IEEE 802.3 规范定义的唯一值比较外，同时与控制器所设置的 MAC 地址逐位比较。如果目标地址域比较通过且接收流控制被使能（ENET_MAC_FCTL 的 RFCEN 位被置 1），则相应暂停控制帧功能将被触发。这个通过过滤的暂停帧是否会被转发给应用取决于 ENET_MAC_FRMF 寄存器的 PCFRM[1:0]位设置。

MAC 的接收流程

MAC 接收到的帧都会被送入 RxFIFO 中。MAC 接收到帧后会剥离其前导码和帧首界定码，并从帧首界定码后的第一个字节（目标地址）开始向 FIFO 发送帧数据。如果使能了 IEEE 1588 时间戳，MAC 会在检测到帧的帧首界定码的时候记录下系统的当前时间。如果这个帧通过地址过滤器的检查，MAC 会把这个时间戳通过接收描述符一并发给应用程序。

若 ENET_MAC_CFG 寄存器的 APCD 位置位，且接收到的帧长度/类型域的值小于 0x600 时，MAC 将自动剥离填充域和帧校验序列。MAC 会在向 RxFIFO 发送完帧长度/类型域规定字节数后，丢弃包括帧校验序列在内的余下字节。如果长度/类型域的值大于或等于 0x600，不管自动 CRC 剥离选项是否使能，MAC 都会把所有接收到的以太网帧数据发送到接收 FIFO。

若看门狗定时器被使能（ENET_MAC_CFG 寄存器中的 WDD 位被复位），当帧长度超过 2048 字节（目标地址 + 源地址 + 长度/类型 + 数据 + 帧校验序列）时将被切断。即使看门狗定时器被禁能，MAC 仍然会切断长度大于 16384 字节（16KB）的帧。

当 RxFIFO 工作于直通模式时，如果 FIFO 中的数据量大于门限值（可通过 ENET_DMA_CTL 寄存器的 RTHC 位设置），就开始从 FIFO 中取出数据，并通知 DMA 接收。当 FIFO 完成取出整个帧后，MAC 控制器将接收状态信息字发送给 DMA 控制器以回写到接收描述符中。在这种模式下，假如一个帧开始由 FIFO 取出由 DMA 发送到应用程序，则即使检测到错误，帧也会一直接收直到整个帧接收完毕。由于错误信息也要等到此时才会发送给 DMA 控制器，此时帧的前部分已经被 DMA 接收，所以在这种模式下将 MAC 设置成将所有错误帧丢弃将无效。

当 RxFIFO 工作于存储转发模式（通过 ENET_DMA_CTL 寄存器的 RSFD 位设置）时，DMA 只在 RxFIFO 完整地收到一帧后，才将其读出。此模式下，如果 MAC 设置成将所有错误帧丢弃，那么 DMA 只会读出合法的帧，并转发给应用程序。一旦 MAC 在接口上检测到帧首界定码就会启动接收过程。MAC 控制器在处理帧之前会剥离前导码和帧首界定码。会通过过滤器检查帧的报头，并用帧校验序列核对帧的 CRC 值。如果帧没能通过地址滤波器，MAC 控制器就会丢弃该帧。

MAC 的接收管理

多个帧的接收处理

与 TxFIFO 不同，由于帧的状态信息紧随在帧数据之后，MAC 可以判断接受帧的状态，因此第二个接收帧的传送是紧接着第一个接收帧的数据与状态信息的，只要 RxFIFO 未满，就可以存放任意数量的帧。

接收流控

在全双工模式下，MAC 能够检测暂停帧，并按照暂停帧中的暂停时间域参数，在暂停一定时间后再发送数据。可以通过设置 ENET_MAC_FCTL 寄存器的 RFCEN 位，使能或者取消暂停帧检测功能。如果没有使能该功能，则 MAC 会忽略接收到的暂停帧。若使能了该功能，MAC 将能够对接收到的暂停帧进行解码。类型域、操作数域和暂停时间域都将能够被 MAC 识别。在暂停期间，如果收到一个新的暂停帧，则新的暂停时间将立即被加载到暂停时间计数器中。如果接到收的暂停时间域值为 0，则 MAC 会停止暂停时间计数器，恢复数据的发送。通过配置 ENET_MAC_FRMF 寄存器的 PCFRM 位值，来处理这些接收到的控制帧。

接收校验和模块

置位 ENET_MAC_CFG 寄存器的 IPFCO 位，可以使能接收校验和模块。接收校验和模块可以计算 IPv4 报头的校验和，并检查它是否与 IPv4 报头的校验和域的内容相匹配以外。MAC 可根据检查接收到的以太网帧类型域是 0x0800 还是 0x86dd，来判别是 IPv4 帧还是 IPv6 帧，这个方法也用于带 VLAN 标签的帧识别。DMA 接收描述符的报头校验和错误位（RDES0 的第 7 位）反映了对报头的校验和结果，该位在接收到的 IP 报头出现下述错误时被置 1：

- 计算的 IPv4 报头的校验和值与其校验和域的内容不匹配；
- 以太网类型域值指示的数据类型与 IP 报头版本域不匹配；
- 接收到的帧长少于 IPv4 报头长度域指示的长度，或者 IPv4/IPv6 报头少于 20 字节。

接收校验和模块还能识别 IP 数据包的数据类型是 TCP、UDP 还是 ICMP，并按照 TCP、UDP 或 ICMP 的规范计算它们的校验和。计算过程包括 TCP/UDP/ICMPv6 伪报头的数据。DMA 接收描述符（RDES0 的第 0 位）的数据校验和错误位反映了对数据的校验和结果，该位在接收到的 IP 数据包数据出现下述错误时被置 1：

- 计算的 TCP、UDP 或 ICMP 校验和与其帧的 TCP、UDP 或 ICMP 校验和域值不匹配；
- 收到的 TCP、UDP 或者 ICMP 数据长度与 IP 报头给出的长度不符。

接收校验和模块不计算下列情况：不完整的 IP 数据包、带安全功能的 IP 数据包、IPv6 路由报头以及数据类型不是 TCP、UDP 或者 ICMP 的数据包。

错误处理

- 在从 MAC 接收到 EOF 之前，RxFIFO 已满。则 MAC 控制器会将整个帧丢弃并返回一个溢出状态。同时将溢出计数器加 1。
- 若 RxFIFO 设置成存储转发模式，MAC 可以过滤并丢弃所有的错误帧。但根据 ENET_DMA_CTL 寄存器的 FERF 和 FUF 位的设置，RxFIFO 仍可以接收错误帧和长度

低于最小帧长的帧。

- 若 RxFIFO 设置成直通模式，并不能将所有的错误帧都丢弃，仅当 DMA 从 RxFIFO 读出帧的 SOF 时，RxFIFO 也已获得了该帧的错误状态时可以丢弃错误帧。

接收状态信息字

在接收以太网帧结束时，MAC 将会分析并记录关于帧以及接收过程的一些状态信息，并把详细的接收状态信息回写到 DMA 接收描述符和 DMA 状态标志。应用程序可以通过检查这些状态位来实现上层协议。

注意：帧长度值为 0 意味着由于某种原因（例如 FIFO 溢出或在接收过程中动态地修改了过滤器的值，导致未通过过滤器的情况等）造成写入 FIFO 的帧不完整。

MAC 回环模式

通常地，回环模式用于应用程序对系统硬件和软件的测试与调试。通过将 ENET_MAC_CFG 寄存器的 LBM 位置'1'，可以使能 MAC 回环模式。在该模式下，MAC 发射端把帧发送到自身的接收端上。该模式默认为关闭。

27.3.3. MAC 统计计数器：MSC

为了了解发送和接收帧的统计情况，利用一组计数器来收集相关的统计数据。这些MAC计数器被称为MAC统计计数器(MSC)。在章节"***ENET寄存器***"中有这些寄存器的详细功能说明。

当发送帧没有出现下述错误时，该帧被认为是"好帧"，MSC发送计数器会自动更新：

- 帧溢出；
- 没有载波；
- 载波丢失；
- 顺延(Deferral)过多；
- 延迟冲突；
- 过度冲突；
- Jabber超时。

当接收帧没有出现下述错误时，该帧被认为是"好帧"，MSC接收计数器会自动更新：

- 对齐错误；
- CRC错误（CRC计算结果与帧校验序列值不一致）；
- 过短帧（帧长少于64字节）；
- 长度错误（长度域值与实际接收到的字节数不符）；
- 超出范围（长度域值超过IEEE 802.3所规定的最大值，即对于非标签帧最大值为1518字节，对于VLAN标签帧最大值为1522字节）；
- MII_RX_ER输入错误。

注意：当被丢弃的帧是长度小于6字节的过短帧(没有完整接收到目标地址)时，MSC接收计数器也会更新。

27.3.4. 唤醒管理：WUM

以太网模块支持两种将系统从深度睡眠模式唤醒的方法。分别为远程唤醒帧和 Magic Packet 唤醒帧。为了减小功耗，可以使主机系统和以太网模块进入低功耗状态，从而可以停止由 HCLK 驱动的电路以及发送时钟。但由接收时钟驱动的电路将继续工作，以监听唤醒帧。如果将 ENET_MAC_WUM 寄存器的 PWD 位置 1，则以太网模块进入低功耗状态。在低功耗状态下，MAC 会丢弃所有的帧，直到退出低功耗状态。此时可以采用上述的两种方式能够退出低功耗状态。将 ENET_MAC_WUM 寄存器的 WFEN 置 1，以设置当收到远程唤醒帧时唤醒以太网模块，或将 ENET_MAC_WUM 寄存器的 MPEN 置 1，以设置当收到 Magic Packet 唤醒帧时唤醒以太网模块。当任一个唤醒功能被使能，一旦 MAC 接收到相应的唤醒帧，以太网模块将产生一个唤醒中断，并退出低功耗状态。

远程唤醒帧检测

将 ENET_MAC_WUM 寄存器的 WFEN 置 1 可以使能远程唤醒检测。当 MAC 处于低功耗状态，且远程唤醒使能位为'1'时，MAC 会进行唤醒帧过滤。如果输入帧通过了过滤器命令的地址过滤，而且过滤器 CRC-16 与被检查的输入帧匹配，则认为接收到唤醒帧，随后 MAC 即恢复正常工作。即便唤醒帧的长度超过了 512 字节，只要该帧有正确的 CRC 值，它仍然被认为是有效的。在接收到远程唤醒帧时还会将 ENET_MAC_WUM 寄存器的 WUFR 位置 1。如果远程唤醒中断没有被屏蔽，那么此时还将产生一个 WUM 中断。

远程唤醒帧过滤器寄存器

唤醒帧过滤器寄存器一共有 8 个，但这些寄存器共用一个相同的偏移地址。在完成对某个过滤器寄存器的读或写的时候，内部的指针会自动指到下一个过滤器寄存器。不论是读还是写操作，强烈建议连续 8 次的操作。也就是说，对其设置时需要将设置的值分为 8 次逐一写入唤醒帧过滤器寄存器地址，读取的时候也是需要连续读 8 次唤醒帧过滤器寄存器，才能将所有值读出。

图 27-6. 唤醒帧过滤器寄存器

<table>
<tr><td>唤醒帧过滤器寄存器0</td><td colspan="8">过滤器0字节屏蔽</td></tr>
<tr><td>唤醒帧过滤器寄存器1</td><td colspan="8">过滤器1字节屏蔽</td></tr>
<tr><td>唤醒帧过滤器寄存器2</td><td colspan="8">过滤器2字节屏蔽</td></tr>
<tr><td>唤醒帧过滤器寄存器3</td><td colspan="8">过滤器3字节屏蔽</td></tr>
<tr><td>唤醒帧过滤器寄存器4</td><td>保留</td><td>过滤器3命令</td><td>保留</td><td>过滤器2命令</td><td>保留</td><td>过滤器1命令</td><td>保留</td><td>过滤器0命令</td></tr>
<tr><td>唤醒帧过滤器寄存器5</td><td colspan="2">过滤器3偏移</td><td colspan="2">过滤器2偏移</td><td colspan="2">过滤器1偏移</td><td colspan="2">过滤器0偏移</td></tr>
<tr><td>唤醒帧过滤器寄存器6</td><td colspan="4">过滤器1 CRC-16</td><td colspan="4">过滤器0 CRC-16</td></tr>
<tr><td>唤醒帧过滤器寄存器7</td><td colspan="4">过滤器3 CRC-16</td><td colspan="4">过滤器2 CRC-16</td></tr>
</table>

■ 过滤器 n 字节屏蔽

该寄存器定义了过滤器 n（n=0，1，2，3）使用帧的哪些字节来检查判断是否为唤醒帧。其第 31 位必须为'0',位[30:0]是字节屏蔽位。如果过滤器 n（n=0，1，2，3）的第 m 位（m=0~30）为'1'，则唤醒帧检测的 CRC 模块会处理输入帧的第[过滤器 n 偏移+m]字节，否则忽略之。

■ 过滤器 n 命令

共 4 位控制过滤器 n 的工作模式。最高位位 3 为地址类型选择，如果该位为'1'，则只检测多播帧；如果该位为'0'，则只检测单播帧。位 2 和位 1 必须保持为 0。位 0 是过滤器 n 的使能位，置位时使能过滤器 n，反之禁能过滤器 n。

■ 过滤器 n 偏移

与过滤器 n 字节屏蔽配合使用。该寄存器定义了过滤器 n 要检查的首字节在帧内的偏移量。最小允许取值是 12，代表了帧的第 13 个字节（偏移值为 0 表示帧的第 1 个字节）。

■ 过滤器 n CRC-16

该寄存器包含了预先写入的 CRC-16 码，用于与帧数据（考虑了过滤器 n 偏移以后且对应的字节屏蔽为 1 的时候的帧字节）计算的 CRC-16 值进行比较。CRC-16 的生成多项式为 0x8005。

Magic Packet 检测

另一种唤醒方法是检测 Magic Packet 唤醒帧（见 AMD 公司的"Magic Packet 技术"）。一个 Magic Packet 帧是一种特殊构成的数据包，专门用于唤醒。这种包可以被以太网模块接收、分析和识别，并用于触发一个唤醒事件。设置 ENET_MAC_WUM 寄存器的 MPEN 位为'1'可以使能此功能。这种类型的帧格式如下：目的和源地址域之后的任何位置连续 6 字节全 1(0xFFFF FFFF FFFF)，接着是在没有任何中断和暂停的情况下有 16 个重复的 MAC 地址；如果这 16 次重复间有任何的间断，则需要重新在输入帧里检测 0xFFFF FFFF FFFF。WUM 模块会持续监视每一个发向本节点的帧，那些通过地址过滤的 Magic Packet 帧，MAC 会进一步检测其是否符合 Magic Packet 的格式，一旦通过检测将会使 MAC 从低功耗状态下唤醒。设备也接受多播帧作为 Magic Packet 帧。

下面是一个站地址为 0xAABB CCDD EEFF 的 Magic Packet 帧实例（MISC 表示包内各种附加的数据字节）：

<DESTINATION><SOURCE><MISC>

……FF FF FF FF FF FF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF AABB CCDD EEFF

<MISC><FCS>

一旦检测到 Magic Packet 帧，ENET_MAC_WUM 寄存器的位 MPKR 会被置 1。如果使能了 Magic Packet 中断，此时还将产生对应中断。

系统在低功耗期间注意事项

在 MCU 处于深度睡眠模式时，若使能外部中断线 19，则以太网的 WUM 模块仍能够检测帧。

由于 MAC 在低功耗状态也需要进行 Magic Packet/远程唤醒帧检测，因此 ENET_MAC_CFG 寄存器的 REN 位必须保持为'1'。在低功耗状态时需要把 ENET_MAC_CFG 寄存器的 TEN 位清'0'来关闭发送功能。此外，由于不需要把 Magic Packet/远程唤醒帧转发给应用，因此在低功耗状态时也要关闭以太网 DMA 模块，可以通过设置 ENET_DMA_CTL 寄存器的 STE 位和 SRE 位（分别对应 TxDMA 和 RxDMA）为'0'来关闭以太网 DMA。

推荐的进入低功耗状态和唤醒步骤如下：

在 MCU 处于深度睡眠模式时，若使能外部中断线 19，则以太网的 WUM 模块仍能够检测帧。由于 MAC 在低功耗状态也需要进行 Magic Packet/远程唤醒帧检测，因此 ENET_MAC_CFG 寄存器的 REN 位必须保持为'1'。在低功耗状态时需要把 ENET_MAC_CFG 寄存器的 TEN 位清'0'来关闭发送功能。此外，由于不需要把 Magic Packet/远程唤醒帧转发给应用，因此在低功耗状态时也要关闭以太网 DMA 模块，可以通过设置 ENET_DMA_CTL 寄存器的 STE 位和 SRE 位（分别对应 TxDMA 和 RxDMA）为'0'来关闭以太网 DMA。

推荐的进入低功耗状态和唤醒步骤如下：

1) 等待当前帧发送完毕，然后将 ENET_DMA_CTL 寄存器的 STE 位复位来关闭 TxDMA；
2) 把 ENET_MAC_CFG 寄存器的 TEN 位和 REN 位清'0'，来关闭 MAC 发射器和 MAC 接收器；
3) 观察 ENET_DMA_STAT 寄存器位 RS，等待 RxDMA 把 RxFIFO 里的所有帧读出，再关闭 RxDMA；
4) 配置并使能外部中断线 19，使其能产生事件或者中断。如果配置了外部中断线 19 产生中断，则还需要编写中断处理程序 ENET_WKUP_IRQ，在其中清除外部中断线 19 的中断标志位；
5) 设置 ENET_MAC_WUM 寄存器的 MPEN 或 WFEN 位（或两位）为'1'，使能 Magic Packet/远程唤醒帧检测（或两种功能）；
6) 设置 ENET_MAC_WUM 寄存器的 PWD 位为'1'，使能低功耗模式；
7) 设置 ENET_MAC_CFG 寄存器的 REN 位为'1'，打开 MAC 接收器；
8) 设置使 MCU 进入深度睡眠模式；
9) 在接收到有效的唤醒帧后，以太网模块退出低功耗状态；
10) 读取 ENET_MAC_WUM 寄存器来清除电源管理事件标志位，打开 MAC 发送器，以及 TxDMA 和 RxDMA；
11) 设置系统时钟：使能 HXTAL 并配置 RCU 时钟参数。

27.3.5. 精确时间协议：PTP

协议的大部分是通过 UDP 层之上的应用程序软件实现的。MAC 的 PTP 模块主要是支持记录 PTP 包从以太网端口发出和收到的准确时间，并将其返回给应用程序。

关于精确时间协议(PTP)的具体内容可参见 IEEE 1588™相关文档。

基准时钟源

IEEE 1588 协议规定，通过一个 64 位寄存器来获得系统基准时间，其中高 32 位提供秒级的时间信息，低 32 位提供纳秒级的时间信息。

PTP 基准时钟输入用来生成系统基准时间（也称为系统时间），以及获取 PTP 帧的时间戳值。其频率必须大于或等于时间戳计数器的分辨率。主节点和从节点之间的时间同步精度在 0.1us 左右。

同步精度

时间同步的精度取决于以下几个因素：

1) PTP 基准时钟输入的频率；
2) 所用晶体振荡器的特性（频漂）；
3) 同步流程的执行频度。

系统时间校准方法

64 位 PTP 系统时间由 PTP 输入基准时钟来更新。这个 PTP 系统时间用来作为记录发送/接收时间戳的依据。该系统时间的初始化和校准支持两种模式：粗调和精调。校准的目的是纠正频率偏移。

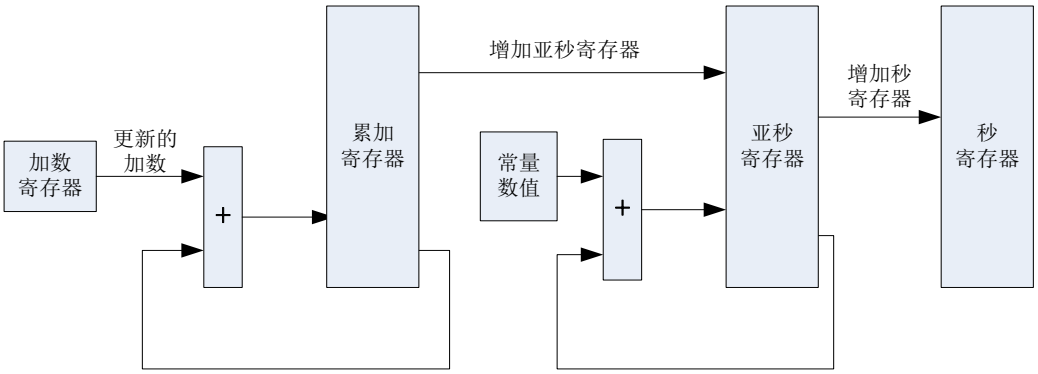
若选择了粗调的方式，则可通过配置 PTP 时间戳更新寄存器(ENET_PTP_TSUH 和 ENET_PTP_TSUL)，来进行系统时间初始化和校准。如果 TMSSTI 位被置位，则 PTP 时间戳更新寄存器被用于初始化。如果 TMSSTU 位被置位，则 PTP 时间戳更新寄存器被用于系统时间的调整，加上或者减去这个寄存器值进行校准。

若选择了精调的方式，则需要一段时间才能完成。由应用程序确定精调的频率，以确保从时钟能线性地同步于主时钟，避免不可预知的大的抖动。

这种方法是指，在每个 HCLK 周期把加数寄存器 ENET_PTP_TSADDEND 中的值加入累加器。当累加器溢出时会产生脉冲令时间戳低寄存器 ENET_PTP_TSL 的值增加。增加的值由亚秒递增寄存器 ENET_PTP_SSINC 中的值决定。

*图 27-7. 系统时钟精细校准方法*演示了精调算法的流程：

图 27-7. 系统时钟精细校准方法



下面是一个具体例子用于说明精调方式如何更新系统时间：

假设系统时钟更新电路的精度需要达到 20ns，即更新的频率为 50MHz。假设基准时钟 HCLK 是 75MHz，计算频率比得 75/50 = 1.5。因此，写入的加数（ENET_PTP_TSADDEND 寄存器

的 TMSA 位）是 $2^{32}/1.5$，等于 0xAAAA AAAA。如果基准时钟频率漂低，假设降至 65MHz，此时频率比变成 65/50 = 1.3，写入加数寄存器的值应当是 $2^{32}/1.30$ = 0xC4EC 4EC4。如果基准时钟漂高，假设升到 85MHz，加数寄存器的值必须为 0xA000 0000。初始时，将加数寄存器设为从时钟频率 Clock Addend Value(0)，该值按上述进行计算。除了配置加数计数器之外，还需对亚秒递增寄存器进行设置才能保证达到 20ns 的精度。每次累加寄存器溢出后，该寄存器的值会对时间戳低寄存器进行更新。由于时间戳低寄存器（位 0 到位 30）表示系统时间的亚秒值，因此精度为 10^9ns/2^{31} = 0.46ns。所以为了使系统时间精度达到 20ns，亚秒递增寄存器的值应该设为 20/0.46 = 43。

注意：下文描述的算法是以主从设备之间传输的时延 Master-to-Slave-Delay 恒定为基础的，通过该算法在若干个 Sync 周期内确定同步频率比。

算法如下：

- 定义主设备发送一个 SYNC 消息到从设备时的时间：MSYNCT(n)
  定义从设备的本地时间 SLOCALT(n)
  定义主设备的本地时间 MLOCALT(n)
  计算：MLOCALT(n) = MSYNCT(n) + Master-to-Slave-Delay(n)
- 定义发送两次 SYNC 消息之间的主设备时钟计数：MCLOCKC(n)
  计算：MCLOCKC(n) = MLOCALT(n) – MLOCALT(n-1)
  定义接收两次 SYNC 消息之间的从设备时钟计数：SCLOCKC(n)
  计算：SCLOCKC (n) = SLOCALT (n) - SLOCALT (n-1)
- 定义两个计算之间的差值：DIFFCC(n)
  计算：DIFFCC(n) = MCLOCKC(n) - SCLOCKC(n)
- 定义从时钟频率调整系数：SCFAF(n)
  计算：SCFAF(n) = (MCLOCKC(n) + DIFFCC(n)) / SCLOCKC(n)
- 定义加数寄存器的时钟加数值：Clock Addend Value(n)
  Clock Addend Value (n) = SCFAF (n) * Clock Addend Value (n-1)

注意：实际操作中，可能需要多个 SYNC 消息来完成主从设备的同步。

系统时间初始化流程

设置 ENET_PTP_TSCTL 寄存器的位 TMSEN 为'1'，可以使能时间戳功能。不过在把该位置'1'以后，必须首先初始化时间戳计数器来开始时间戳操作。初始化步骤如下：

1) 置位 ENET_MAC_INTMSK 寄存器的位 9，以屏蔽时间戳触发中断；
2) 置位 ENET_PTP_TSCTL 寄存器位 0，以使能时间戳；
3) 根据期望时钟精度配置亚秒递增寄存器；
4) 若希望采用精调校准方式，则配置时间戳加数寄存器，并置位 ENET_PTP_TSCTL 寄存器的位 5。若希望采用粗调校准方式，则忽略第 4-6 步，直接跳至第 7 步；
5) 轮询 ENET_PTP_TSCTL 寄存器的位 TMSARU，直到其变为'0'；
6) 将 ENET_PTP_TSCTL 寄存器的位 1 置位，来选择使用精调校准方式；
7) 把希望设置的系统时间值写入时间戳更新高寄存器和时间戳更新低寄存器；
8) 置位 ENET_PTP_TSCTL 寄存器的位 2，以初始化时间戳；
9) 一旦初始化成功后，时间戳计数器就开始工作。

用粗调方式更新系统时间的步骤

1) 在时间戳更新高寄存器和时间戳更新低寄存器中写入偏移值（可以是负值）；
2) 置位 ENET_PTP_TSCTL 寄存器的位 3(TMSSTU)，以更新时间戳寄存器；
3) 轮询 TMSSTU 位，直到其被清'0'后完成。

用精调方式更新系统时间的步骤

1) 利用前述"*PTP基准时钟输入的频率；*
4) 所用晶体振荡器的特性（频漂）；
5) 同步流程的执行频度。

2) **系统时间校准方法**"介绍的算法，计算出期望的系统时钟频率所对应的加数寄存器的值；
3) 将值写入加数计数器，并设置ENET_PTP_TSCTL寄存器位5为'1'将该值更新到PTP模块；
4) 把要求的期望时间写入期望时间高和期望时间低寄存器，并设置ENET_MAC_INTMSK寄存器的位9为'0'来允许时间戳中断；
5) 设置ENET_PTP_TSCTL寄存器的位4(TMSITEN)为'1'使能时间戳中断；
6) 在这个事件产生中断时，读出ENET_MAC_INTF寄存器的值以清除相应的中断标志位；
7) 重新用旧值编写时间戳加数寄存器，并设置ENET_PTP_TSCTL寄存器位5为'1'将值更新到PTP模块。

带 PTP 功能的帧的发送与接收

在使能了 IEEE 1588(PTP)时间戳功能后，在发送帧的帧首界定码从 MAC 输出或者 MAC 接收到帧的帧首界定码的时候，时间戳值被记录。每一个等待发送的帧在 DMA 发送描述符中都有一个标志，指示是否需要记录这个帧的时间戳。

记录下来的时间戳会和帧的发送/接收状态信息一起，存放在相应的发送/接收描述符里。64 位的发送帧时间戳写入 DMA 发送描述符，64 位的接受帧时间戳写入 DMA 接收描述符。具体描述见后续的"*TxDMA 描述符*"和"*带 IEEE1588 时间戳的 RxDMA 描述符格式*

如果使能了 IEEE 1588 功能，则 MAC 控制器会在带时间戳的帧接收完成之后，DMA 关闭描述符之前（DAV 位清'0'），将时间戳写入 RDES2 和 RDES3。

RxDMA 描述符"。

PTP 触发与 TIMER1 的内部连接

MAC 可以在系统时间大于等于期望时间的时候提供触发中断。使用中断会引入一段已知的中断时延再加上不确定的命令执行时间。为了计算这部分已知的中断时延时间，在系统时间大于期望值的时候，PTP 会将一个输出信号置高。将 AFIO_PCF0 寄存器的位 29 设为 0，可将此输出信号内部连接到 TIMER1 的 ITI0 输入上。利用这个信号，由于 TIMER1 的时钟与 PTP 基准时钟(HCLK)是同步的，因此不再有任何不确定的误差。

PTP 秒脉冲输出信号

将 ETH_PPS_OUT 设为 AF 复用推挽功能，可以使能 PPS 输出功能。该功能可以输出脉冲宽度为默认 125ms 的脉冲，用于检查网络全部节点之间的同步。为了测试本地从时钟和主时钟

之间的差别，可以把主从设备的 PPS 输出都连接到示波器，以测量 2 个时钟之间的差别。

27.3.6. DMA 控制器描述

为了减少 CPU 的干预，设计了以太网专用 DMA 控制器，用于实现 FIFO 和系统存储之间的帧数据传输。DMA 和 CPU 之间的通讯通过 2 种数据结构实现：

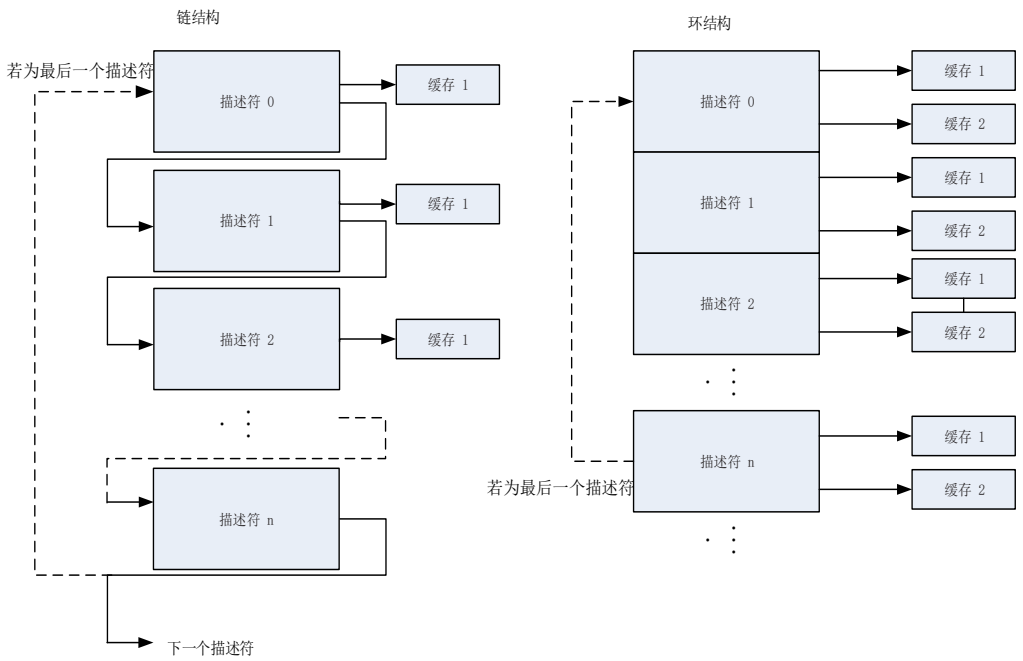
- 描述符列表（链结构或环结构）和数据缓存；
- 控制和状态寄存器。

应用程序需要开辟存储描述符列表及数据缓存用到的物理内存。在存储器里，描述符是以指向缓存的指针的形式存放。有 2 个描述符队列，一个用作发送，另一个用作接收。两个队列的基地址分别存放在 ENET_DMA_TDTADDR 寄存器和 ENET_DMA_RDTADDR 寄存器中。发送描述符由四个描述符字 TDES0-TDES3 组成。同样的，接收描述符由四个描述符字 RDES0-RDES3 组成。每个描述符可以指向最多 2 个缓存用来存储帧的数据。根据描述符列表类型是环结构还是链结构，来决定第二个缓存是被配置为第二个数据存储地址，还是下一个描述符地址。数据缓存存放在 MCU 的物理内存里，可以存放一个帧的全部或者部分，但是不允许存放不属于同一个帧的数据。描述符队列可以是显性(链结构)或者隐性(环结构)的方式前向连接的。通过设置接收描述符的 RDES1 位 14 和发送描述符的 TDES0 位 20 为'1'，可以实现描述符的显性连接，此时 RDES2 及 TDES2 中将存放缓存地址，RDES3 及 TDES3 中将存放下一个描述符的地址，这种链接的描述符也可以称为描述符的链结构。通过设置接收描述符的 RDES1 位 14 和发送描述符的 TDES0 位 20 为'0'，可以实现描述符的隐性连接，此时 RDES2 及 TDES2，RDES3 及 TDES3 中都将存放缓存地址，这种链接的描述符也可以称为描述符的环结构。在使用当前的描述符所指向的缓存地址时，描述符指针就指向下一个描述符。当使用链结构时，描述符指针指向的是第二个缓存。当使用环结构，根据下式计算描述符指针下一个所指向的地址：

$$\text{下个描述符地址} = \text{当前描述符地址} + 16 + DPSL*4$$

若当前描述符是描述符列表的最后一个描述符，环结构下必须设置 TDES0 的位 21 或 RDES1 的位 15 以标识当前描述符为列表的最后一个。此时下一个描述符又指向描述符列表的第一个。链结构下还可以通过设置 TDES3 或 RDES3 的值指向描述符列表中第一个的地址。DMA 一旦检测到帧结束就会跳到下一个帧的缓存。

图 27-8. 描述符的环结构和链结构



数据缓存地址对齐

以太网 DMA 控制器支持所有对齐类型：字节对齐，半字对齐，字对齐。这意味着应用程序可将发送和接收数据缓存地址配置到任意地址。但是，在 DMA 发起传输的时候，总是以字对齐的方式访问地址。对于读和写缓存的访问也不一样。示例如下：

- 读缓存示例：如果发送缓存的地址为 0x2000 0AB2，并需要传输 15 字节。在开始读操作后，DMA 实际会从地址 0x2000 0AB0，0x2000 0AB4，0x2000 0AB8，0x2000 0ABC 和 0x2000 0AC0 先读 5 个字，但是在往 FIFO 发送数据的时候，会丢弃头 2 个字节(0x2000 0AB0 和 0x2000 0AB1)和最后 3 个字节（0x2000 0AC1，0x2000 0AC2 和 0x2000 0AC3）。
- 写缓存示例：如果接收缓存的地址为 0x2000 0CD2，并需要传输 16 字节。在开始写操作后，DMA 实际会从地址 0x2000 0CD0 到 0x2000 0CE0 先写 5 个 32 位数据。但是头 2 个字节（0x2000 0CD0 和 0x2000 0CD1）和末尾的 2 个字节（0x2000 0CE2 和 0x2000 0CE3）会用虚拟字节替代。

注意：DMA 控制器不会写任何数据到定义的缓存区之外的地址。

缓冲区有效长度

发送帧的过程中，TxDMA 会传输与 TDES1 中标明的缓存有效长度的字节给 MAC 控制器。如前所述，一个发送帧可以用多个描述符来描述一个帧，即一个帧的数据可以处于多个不同的缓存中。如果 DMA 控制器读取的描述符 TDES0 的 FSG 位为'1'，那么 DMA 就明确了当前缓存存储的是一个新的帧，并标记发送的第一个字节是帧首。如果 DMA 控制器读取的描述符 TDES0 的 LSG 位为'1'，则 DMA 就明确了当前缓存存储的是当前帧的最后一部分数据。通常来说一个帧只存在一个缓存里（因为缓存的大小对于一个正常的帧来说足够大了），因此 FSG 和 LSG 位会在一个相同的描述符中同时置位。

接收帧的过程中，接收帧的缓存长度域值必须是字对齐的。对于字对齐或非字对齐的缓存地址，接收操作与发送操作不大相同。如果接收缓存地址是字对齐的，则与发送流程是类似的，缓存的有效长度为由 RDES1 中配置的值。如果接收缓存地址是非字对齐的，则缓存的有效长度将小于 RDES1 中配置的值。缓存有效长度值应为 RDES1 中配置的值减去缓存地址的低 2 位值。例如，假设缓存的总大小为 2048 字节，缓存地址为 0x2000 0001，地址的低 2 位值为 0b01，那么缓存有效长度为 2047 个字节，范围从 0x2000 0001（帧首）到 0x2000 07FF。

当收到了一个帧起始 SOF，则 DMA 控制器将 FSG 位置位，当收到一个帧结束 EOF 时，则 LSG 位被置位。如果接收缓存长度域值配置的足够大，能放下整个帧，则 FSG 和 LSG 位将在同个描述符中被置位。实际接收的帧长度可从 RDES0 的 FRML 位域获取。从而应用程序可计算未被使用的缓存空间。RxDMA 总是用新的描述符来接收下一帧。

TxDMA 和 RxDMA 的仲裁器

DMA 的仲裁器设计了两种仲裁方式用于提高 DMA 发送与接收控制器的效率：固定和轮询优先级。设置 ENET_DMA_BCTL 寄存器 DAB 位为'0'，选择轮询优先级，在 TxDMA 和 RxDMA 同时要求访问数据总线的时候，按照 ENET_DMA_BCTL 寄存器 RTPR 位设定的比例对其访问进行分配。设置 DAB 位为'1'选择固定优先级，此时 RxDMA 和 TxDMA 同时要求访问时，RxDMA 总是对总线拥有更高的访问优先级。

对 DMA 的错误响应

若 DMA 在传输过程中出现了错误的总线响应，那么 DMA 控制器认为发生了一个致命错误，会立刻停止所有操作，并更新状态寄存器 ENET_DMA_STAT。在发生类似的致命错误（响应错误）之后，应用程序必须复位以太网外设并重新初始化 DMA，DMA 才能恢复操作。

TxDMA 与 RxDMA 控制器的初始化

在使用 DMA 控制器之前，必须按如下步骤对 DMA 进行初始化：

1) 对ENET_DMA_BCTL寄存器进行总线访问参数的相关设置；
2) 对ENET_DMA_INTEN寄存器进行设置，屏蔽不需要的中断源；
3) 将发送描述符列表和接收描述符列表的基地址分别写入ENET_DMA_TDTADDR寄存器与ENET_DMA_RDTADDR寄存器中；
4) 对相关的寄存器进行期望的过滤器配置；
5) 根据从PHY读出的自协商的结果，设置SPD位和DPM位的值，来选择通讯模式（半/全双工）及通讯速度（10Mbit/s或100Mbit/s）。将ENET_MAC_CFG寄存器的TEN和REN位置'1'，使能MAC的发送和接收操作；
6) 设置ENET_DMA_CTL寄存器的位STE和位SRE为'1'，使能DMA发送和接收器。

注意：如果 HCLK 频率过低，应用程序可以先使能 DMA 接收器，再将 ENET_MAC_CFG 寄存器的 REN 位置'1'，以避免 RxFIFO 在启动的时候溢出。

TxDMA 设置

发送缓存区中第二帧操作

如果 ENET_DMA_CTL 寄存器中 OSF 位为'0'，则发送顺序为：首先读取发送描述符，然后从系统存储读取数据写到 FIFO，再将帧数据通过 MAC 放到接口上，最后等待数据发送完毕后将发送状态写回描述符。

上述是 TxDMA 的标准发送流程，但当 HCLK 远远大于 TX_CLK 时，在发送两个帧时发送效率将显著降低。

为避免上述提及的情况，应用程序可将 OSF 位置位。在此情况下，第二帧的数据可以不等待第一帧的描述符状态信息被写回，就先读取内存里的第二帧数据，并把它们送进 FIFO。OSF 功能仅在两相邻帧之间起作用。

TxDMA 操作模式(A)（默认）：非 OSF

在默认模式下，TxDMA 控制器的工作流程如下：

1) 初始化帧数据到发送缓存，并对发送描述符(TDES0-TDES3)进行设置，置TDES0的DAV位为'1'；
2) 将ENET_DMA_CTL寄存器的STE位置为'1'，使能TxDMA控制器；
3) TxDMA控制器开始轮询发送描述符列表来获取待发送的帧。如果TxDMA检测到TDES0[31]的位DAV为0，或者发生了错误，则控制器就会终止传输进入挂起状态，并设置ENET_DMA_STAT寄存器的发送缓存不可用位(位2)和正常中断汇总位(位16)为'1'。如果处于挂起状态，则发送控制器操作跳至步骤8；
4) 如果取到的描述符标志位显示该描述符由DMA占有(DAV位被置'1')，那么DMA从描述符中解析出所配置的发送帧以及发送数据缓存的地址；
5) DMA从内存中取出数据并将数据存入TxFIFO；
6) TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去（LSG位置位）。如果当前描述符的LSG位为'0'，则在所有缓存数据送入TxFIFO之后，将DAV位清零以关闭这个描述符。然后TxDMA控制器等待写回描述符状态，以及IEEE 1588时间戳值（如果使能了时间戳功能）；
7) 在整个帧发送完成以后，仅当TDES0[30]位INTC为'1'时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。然后DMA控制器返回步骤3，继续处理下一帧；
8) 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值，并清除发送溢出标志位，TxDMA将重新回到运行状态，尝试重新获取描述符。发送控制器操作回到步骤3。

TxDMA 操作模式(B)：OSF

在操作第二帧(OSF)模式下，TxDMA 可以不必等到前一帧的状态信息写回，就发送下一帧。如果系统时钟频率远远大于 MAC 频率（10Mbit/s 或 100Mbit/s），这种情况 OSF 模式可以提高发送效率。设置 ENET_DMA_CTL 寄存器的位 OSF 为'1'，进入此模式。DMA 在发送完前一帧数据后，不必等到前一阵的状态写回，而是立即查询第二帧的发送描述符，如果第二帧发送

描述符的 DAV 位与 FSG 位都置 1，那么 TxDMA 立即读取第二帧的帧数据并将其存入 MAC FIFO。

在 OSF 模式下，TxDMA 的操作流程如下：

1) 按照TxDMA默认模式的步骤1-6操作；
2) DMA不等到关闭前一帧的最后一个描述符（LSG位为'1'），就直接取下一个描述符；
3) 如果取到的描述符标明被DMA占有(DAV位为'1')，那么就从解析的发送缓存地址中读取下一帧的数据。如果DAV位为'0'即DMA不占有这个描述符，则TxDMA进入挂起状态并跳到步骤7；
4) TxDMA控制器会一直轮询描述符列表直到帧结尾被传送出去。如果一个帧由多个描述符描述，则中间描述符会在获取之后就被关闭；
5) TxDMA等待前一帧的发送状态信息和时间戳（如果使能了时间戳功能），在接收到状态信息后，DMA会把DAV位为'0'的状态信息写入TDES0，将该描述符的占有权交还给CPU进行操作；
6) 在整个帧发送完成以后，仅当TDES0[30]位INTC为'1'时，发送状态位（ENET_DMA_STAT寄存器中的TS位）会被置位。此时若使能了DMA中断，将进入相应中断。如果前一个帧返回的状态信息正常则跳到步骤3。若显示有数据下溢错误，TxDMA进入挂起状态，并跳到步骤7；
7) 在挂起状态下，如果TxDMA收到一个发送帧的待处理的状态信息和时间戳（若使能了时间戳），则TxDMA将这些信息写入发送描述符，并将相应描述符的的DAV位清零。随后设置相关的中断标志位并回到暂停状态；
8) 在挂起状态下，如果向发送查询使能寄存器ENET_DMA_TPEN写入任意值并将溢出中断标志位清零，TxDMA将回到运行状态，尝试重新获取描述符。发送控制器操作根据是否有待处理的状态信息跳到步骤1或者步骤2。

发送帧格式

根据前述的 IEEE 802.3 规范，一个正常的发送帧应该由以下及部分构成：前导码，帧首界定码 SFD，目标地址 DA，源地址 SA，QTAG 前缀（可选），长度/类型域 LT，数据，PAD 填充域（可选），和帧校验序列 FCS。

前导码和帧首界定码都是由 MAC 自动生成的，因此应用程序只需要存储目标地址，源地址，QTAG（若需要），长度/类型，数据，填充域（若需要），帧校验序列（若需要）。如果帧需要填充位，即缓存中没有存储填充位和帧校验序列部分，则应用程序可配置自动生成帧校验序列和填充位功能。如果帧仅需帧校验序列，即缓存中没有存储帧校验序列部分，则应用程序可配置自动生成帧校验序列。DPAD 位和 DCRC 位用于配置填充位和帧校验序列的自动生成。

发送帧处理

如前所述，一个帧可以分散在不同缓存内，这意味着需要多个描述符。当 FSG 位置位，表示当前描述符指向的缓存为帧头，当 LSG 位置位，表示当前描述符指向的缓存为帧尾。对于当前帧其他描述符（LSG 位为'0'的描述符），TxDMA 控制器仅修改清零其 DAV 位。在这最后一个缓存的数据发送完毕以后，DMA 会将整个帧的发送状态信息，写入最后一个的发送描述符 TDES0 并返回。将数据从系统存储传输到 FIFO，开始发送数据，但实际上真正的数据发送是由 TxDMA 模式决定的：直通模式和存储转发模式。直通模式在 FIFO 中的字节数大于所配置

的阈值时，数据将取出到 MAC 发送。存储转发模式在整个帧数据都传入 FIFO 后或 FIFO 快要填满时再取出数据给 MAC 进行发送。

发送查询挂起后的处理

当传输开始后 DMA 会不断对发送描述符进行查询，当发生如下情况时，会导致 DMA 进入挂起状态，并暂停发送。此时当前描述符固定为暂停前的最后一个描述符。

- DMA 检测到描述符 TDES0 的 DAV 位为 0，此时 CPU 占有描述符，则会进入挂起状态，并暂停查询。同时设置 ENET_DMA_STAT 寄存器的正常中断总结位 NI 和发送缓存不可用位 TBU 为'1'。
- 当接口在发送帧的过程中 MAC FIFO 为空，意味着检测到了数据下溢错误。在此情况下，设置 ENET_DMA_STAT 寄存器的异常中断总结位 AI 和发送数据下溢位 TU 为'1'，同时把该信息写入 TDES0。

带 IEEE 1588 时间戳的 TxDMA 描述符格式

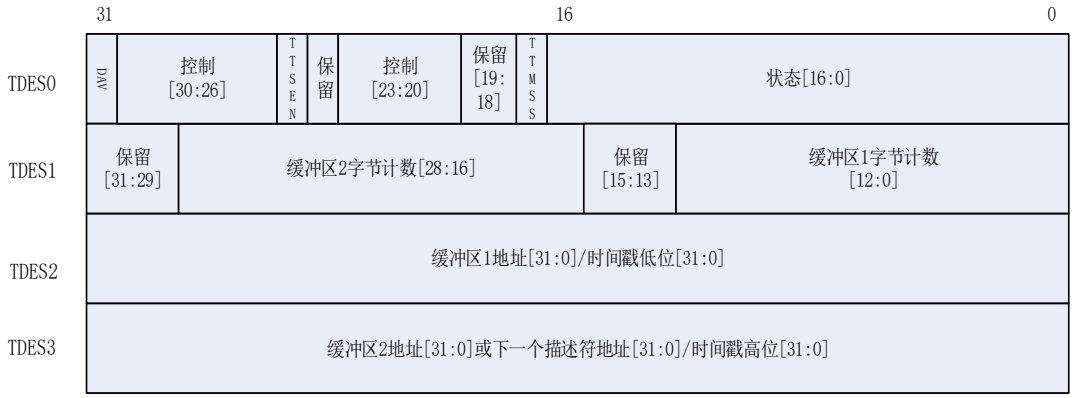
如果设置 TTSEN 位为'1'，则使能了 IEEE1588 功能。TxDMA 控制器会在帧发送完成后，将时间戳写入描述符 TDES2 和 TDES3 的同时设置 TTMSS 位为'1'。

TxDMA 描述符

TxDMA 描述符结构体包含 4 个 32 位字，TDES0~TDES3。TDES0，TDES1，TDES2 和 TDES3 的位定义如下：

注意：若一个帧由多个描述符表示，则对于描述符的控制位（除了 INTC 位）只有第一个描述符的才有效。状态信息和时间戳（若使能了时间戳功能）只写回到最后一个描述符。

图 27-9. 发送描述符



- TDES0：发送描述符字 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	INTC	LSG	FSG	DCRC	DPAD	TTSEN	保留	CM[1:0]		TERM	TCHM	保留		TTMSS	IPHE
rw	rw	rw	rw	rw	rw	rw		rw		rw	rw			rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FRMF	IPPE	LCA	NCA	LCO	ECO	VFRM	COCNT[3:0]				EXD	UFE	DB
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位/位域	名称	描述
31	DAV	DAV 位 DMA 会在将帧完整传输或者描述符指向的缓存里的数据全部被读出以后把该位清'0'。当一个帧位于多个缓存中时，第一个缓存描述符的 DAV 位，必须在后面缓存描述符的 DAV 位全部置'1'以后，才能置'1' 0：表示 CPU 占有描述符 1：表示 DMA 占有描述符
30	INTC	完成时中断位 在最后分块位置位后，此位才有效 0：帧发送完成时，ENET_DMA_STAT 寄存器的 TS 位不被置位 1：帧发送完成时，ENET_DMA_STAT 寄存器的 TS 位被置位
29	LSG	最后分块位 此位指示缓存包含帧的最后一个分块 0：该描述符缓存中没有存放帧的最后一个分块 1：该描述符缓存中存放有帧的最后一个分块
28	FSG	第一分块位 此位指示缓存包含帧的第一个分块 0：该描述符缓存中没有存放帧的第一个分块 1：该描述符缓存中存放有帧的第一个分块
27	DCRC	不计算 CRC 位 该位只有在 FSG 位(TDES0[28])为'1'时才有效 0：MAC 自动插入循环冗余检测(CRC)域 1：MAC 不插入循环冗余检测(CRC)域
26	DPAD	不填充位 该位只在 FSG 位(TDES0[28])为'1'时有效 0：MAC 对帧长不足 64 字节的帧自动添加填充字节，并且插入 CRC 数值。忽略 DCRC 位(TDES0[27])的值 1：MAC 不对帧长不足 64 字节的帧自动填充字节
25	TTSEN	使能发送时间戳位 该位只在 FSG 位(TDES0[28])为'1'时有效 0：发送时间戳功能失能 1：当 ENET_PTP_TSCTL 寄存器的 TMSEN 位(位 0)为'1'时，对应的帧会打开 IEEE1588 硬件时间戳功能
24	保留	必须保持复位值。
23:22	CM[1:0]	校验和插入模式位 0x0：不插入校验和 0x1：只使能硬件 IP 报头的校验和计算和插入 0x2：使能硬件 IP 报头和数据域的校验和计算和插入，但是不计算伪报头的校验和

		0x3：使能硬件 IP 报头和数据域的校验和计算和插入，也计算伪报头的校验和
21	TERM	环形发送结束模式位 该位仅在环模式下使用，且比 TCHM 位具有更高优先级 0：当前描述符还不是描述符队列的最后一个 1：当前描述符到达描述符队列的最后一个，DMA 返回列表的基地址
20	TCHM	第二地址链表模式位 该位在链模式下使用。该位为'1'时，则 TB2S 位域(TDES1[28:16])的值不起作用 0：描述符里的第二个地址是第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，而不是第二个缓存的地址
19:18	保留	必须保持复位值。
17	TTMSS	发送时间戳状态位 该位只在 LSG 位(TDES0[29])为'1'时才有效 0：还未记录帧的时间戳信息 1：记录下了描述符对应的帧时间戳，记录的时间戳放在 TDES2 和 TDES3 处。
16	IPHE	IP 报头错误位 发生下列任意一种情况，则产生 IP 报头错误： IPv4 帧： 1)报头长度域值小于 0x5 2)报头长度域值与报头的长度不符 3)报头版本域值与帧长度/类型域值不匹配 IPv6 帧： 1)主报头长度不足 40 字节 2)报头版本域值与帧长度/类型域值不匹配 0：未发现 IP 数据包报头的错误 1：MAC 发送端发现了 IP 数据包报头的错误
15	ES	错误汇总 该位为下列位的逻辑"或"： TDES0[16]：IP 报头错误 TDES0[14]：Jabber 超时 TDES0[13]：帧清空 TDES0[12]：IP 数据错误 TDES0[11]：载波丢失 TDES0[10]：无载波 TDES0[9]：延迟冲突 TDES0[8]：过度冲突 TDES0[2]：过度顺延 TDES0[1]：数据下溢错误
14	JT	Jabber 超时位 该位仅当 JBD 位复位时才会被置'1' 0：未发生 Jabber 超时

		1：MAC 发送端发生了 Jabber 超时
13	FRMF	帧清空位 置 1 时，清空 TxFIFO 中的数据
12	IPPE	IP 数据错误位 发送端会核对 IPv4 或者 IPv6 报头的数据长度域值与实际收到的 TCP、UDP 和 ICMP 数据数目，不符合就置'1'报错 0：未发生 IP 数据错误 1：MAC 发送端发现了 IP 数据包的 TCP、UDP 或者 ICMP 的 IP 数据错误
11	LCA	载波丢失位 在发送时，如果 CRS 信号在一个或一个以上发送时钟周期中为无效状态，并且没有发生冲突，则载波丢失将概率性发生 该位只有在半双工模式下有效 0：未发生载波丢失 1：帧发送的时候发生了载波丢失
10	NCA	无载波位 0：PHY 的载波侦听信号有效 1：帧发送的时候 PHY 的载波侦听信号无效
9	LCO	延迟冲突位 如果冲突在 64 字节（包括前导符）发送之后发生，则这种情况称作延迟冲突 0：未发生延迟冲突 1：发生了延迟冲突 **注意：**如果溢出错误位 UFE 置'1'，该位无效
8	ECO	过度冲突位 如果 MAC 设置寄存器的 RTD(不进行重试)位为'1'，那么在发生一次冲突后，该位就置'1' 如果 MAC 设置寄存器的 RTD(不进行重试)位为'0'，那么在连续发生 16 次冲突后，该位置'1' 若该位置位，则中止当前帧的发送 0：未发生过度冲突 1：发生了过度冲突
7	VFRM	VLAN 帧位 0：发送帧为普通帧 1：发送的帧是 VLAN 帧
6:3	COCNT[3:0]	冲突计数位 该 4 位值记录了帧发送出去前出现的冲突次数。在 ECO 位(TDES0[8])为'1'时，该位无效
2	EXD	过度顺延位 当 MAC 设置寄存器的顺延位 DFC 为'1'时有效

		0：未发生过度顺延 1：由于顺延超过 3036 字节的时间而结束发送
1	UFE	数据下溢错误位 数据下溢错误表示由于从系统存储传输数据到 FIFO 的速度过慢，导致 DMA 在发送帧的时候遇到了空的缓存。发送过程进入挂起状态，并将 ENET_DMA_STAT 寄存器的发送数据下溢位 TU(位 5)和发送状态位 TS(位 0)都置'1' 0：未发生数据下溢错误 1：发生了数据下溢错误，MAC 中止帧的发送
0	DB	顺延位 该位指示了是否由于载波侦听信号 CRS 在 MAC 发送帧之前被占用，而导致发生帧的顺延 该位只在半双工模式下有效 0：未发生发送顺延 1：MAC 发生了顺延，推迟发送

■ TDES1：发送描述符字 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			TB2S[12:0]												
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TB1S[12:0]												
									rw						

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:16	TB2S[12:0]	发送缓存 2 大小 这些位给出了第二个数据缓存的大小(以字节记)，如果 TDES0[20]位 TCHM 为'1'时，这些位无效
15:13	保留	必须保持复位值。
12:0	TB1S[12:0]	发送缓存 1 大小 这些位给出了第一个数据缓存的大小(以字节记)，如果它的值是 0，则 DMA 跳过这个缓存，并根据 TDES0[20]位使用缓存 2(TCHM=0)或者下一个描述符的缓存(TCHM=1)

■ TDES2：发送描述符字 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB1AP/TTSL[31:16]															
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB1AP/TTSL[15:0]															

rw

位/位域	名称	描述
31:0	TB1AP/TTSL[31:0]	发送缓存 1 地址指针/发送帧时间戳低 32 位 在发送帧之前，应用程序必须对这些位进行配置发生缓存 1 地址(TB1AP)，等到数据发送完后，DMA 可以用它们存放帧的时间戳低 32 位。 当这些位的值表示缓存 1 的物理地址(TTSL)时，对缓存的地址对齐不做限制。 当这些位的值表示时间戳低 32 位(TB1AP)时，当前描述符的 TTSEN 位和 LSG 位必须置位。

■ TDES3：发送描述符字 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TB2AP/TTSH[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB2AP/TTSH[15:0]															
rw															

位/位域	名称	描述
31:0	TB2AP/TTSH[31:0]	发送缓存 2 地址指针(下个描述符地址)/发送帧时间戳高 32 位 在发送帧之前，应用程序必须对这些位进行配置发生缓存 2 地址(TB2AP)，或者配置下一个描述符地址（由描述符类型是链型还是环型决定）。等到数据发送完后，DMA 可以用它们存放帧的时间戳高 32 位 TTSH。 当这些位的值表示缓存 2 的物理地址时(TCHM=0)，对缓存的地址对齐不做限制。 当这些位的值表示下个描述符地址时(TCHM=1)，这些位必须是字对齐的。当这些位的值表示时间戳高 32 位时，当前描述符的 TTSEN 位和 LSG 位必须置位。

RxDMA 设置

RxDMA 控制器的工作流程如下：

1. DMA 接收描述符初始化，置 RDES0[31]的 DAV 位为'1'；
2. 将 ENET_DMA_CTL 寄存器的 SRE 位置为'1'，使能 RxDMA 控制器。DMA 进入运行状态后，会从 ENET_DMA_RDTADDR 寄存器配置的描述符列表基地址获取接收描述符。如果获取的描述符 DAV 位为 1，则当前描述符开始接收帧。但如果检测到取到的描述符正在被 CPU 操作而不可用(DAV=0)，则 DMA 进入挂起状态，跳到步骤 9；
3. 如果获取的描述符显示描述符由 DMA 占有(DAV=1)，那么该描述符的控制位和缓存地址就会被 DMA 所记录；
4. 处理接收到的帧，并从 RxFIFO 将数据传输到接收缓存；
5. 如果缓存被填满或者帧传输结束，接收控制器会从描述符队列中获取下一个接收描述符；
6. 如果当前帧传输结束，DMA 操作跳到步骤 7。如果当前帧传输没有结束(未接收到帧尾 EOF)，则可能发生两种情况：
   - 下一个描述符的 DAV 位为'0'。如果接收帧清空功能使能，则 RxDMA 控制器将 RDES0 的描述符错误位 DERR 位置位。然后 RxDMA 控制器将当前描述符的

DAV 位清零以关闭描述符，并根据帧清空功能是否使能来确定是否置位 LSG 位（若使能则置位 LSG，反之则不置位 LSG）。之后 DMA 操作跳到步骤 8；

- 下一个描述符的 DAV 位为'1'。那么 RxDMA 将 DAV 位清零以关闭当前描述符，之后操作退回步骤 4。

7. 如果使能了 IEEE 1588 时间戳功能，在接收帧完成后 DMA 控制器会把获取的时间戳的低位和高位（如果接收帧符合需要记录时间戳的帧的条件），分别写入当前描述符的 RDES2 和 RDES3。同时 DMA 把从 MAC 处返回的接收状态信息写入 RDES0，并把 DAV 位清'0'，把 LSG 位置'1'；
8. 如果新获取的描述符 DAV 位为'1'，则 RxDMA 控制器操作跳动步骤 4。如果 DAV 位为'0'，则 RxDMA 控制器进入挂起状态，并设 ENET_DMA_STAT 寄存器的 RBU 位为 1。如果使能了接收帧清空功能，则在 DMA 进入挂起状态之前，控制器会清空接收帧；
9. 在挂起状态下，有两种方法退出该状态。一种方法是向接送查询使能寄存器 ENET_DMA_RPEN 中写入任意值。另一种方法是 RxFIFO 收到下一帧数据，这意味着在直通模式下时，帧数据字节数需要大于设置的阈值，或者在存储转发模式下，需要收到整个帧。当 DMA 退出暂停状态后，DMA 会获取下一个描述符，并跳到步骤 2。

获取接收描述符

只要满足下列条件任意一个或多个，DMA 就会尝试获取接收描述符：

- 在寄存器 ENET_DMA_CTL 的接收开始/停止位 SRE 从'0'变为'1'，使 DMA 控制器进入运行状态的时候；
- 当前描述符的整个缓存大小（对于链结构为缓存 1，对于环结构为缓存 1 和 2）不足以接收整个帧，也就是说在接收到帧的结尾之前，当前描述符的缓存已满；
- 在一个完整的帧传送到接收缓存之后，并在当前描述符关闭之前；
- 在挂起状态时，MAC 接收到新的帧；
- 向接送查询使能寄存器 ENET_DMA_RPEN 写入任意值。

接收帧处理

当接口上出现一个帧的时候，MAC 开始接收帧。同时，地址过滤模块开始工作，如果这个帧没有通过地址过滤，则 MAC RxFIFO 将忽略该帧，不会将其通过 RxDMA 转发给接收缓存。如果这个帧通过了地址过滤，则其在不同的转发条件满足时会被转发给接收缓存。在直通模式下，这个转发条件是指接收的帧长大于等于设好的接收阈值。在存储转发模式下，这个转发条件是指 FIFO 里存入了完整的帧时。在接收帧的过程中，当发生以下任意一种情况时，将丢弃 RxFIFO 中的数据，并且不转发数据：1）RxFIFO 中数据少于 64 字节。2）在接收过程中发生了冲突。3）提前终止接收帧。

当满足转发条件时，RxDMA 控制器开始将数据从 RxFIFO 中传输到接收缓存中。若当前缓存中包含了帧起始，则在 RxDMA 控制器写回帧接收状态的时候会将 RDES0 中的 FDES 位置位，以表明这个描述符中存储的是帧的第一部分。若当前缓存中包含了帧结尾，则在 RxDMA 控制器写回帧接收状态的时候会将 RDES0 中的 LDES 位置位，以表明这个描述符中存储的是帧的最后一部分。通常当接收缓存大小大于接收帧的长度时，FDES 位和 LDES 位会在同一个描述符中置位。当缓存接收到了帧结尾，或者当前描述符的缓存不足以存储整个帧时，RxDMA 将获取下一个接收描述符，并将上一个描述符的 RDES0 的 DAV 位清零以关闭上个描述符。当 DINTC=0，当 LDES 位置位时，描述符其他状态也会更新，并且 ENET_DMA_STAT 寄存

器中的 RS 位将置位；当 DINTC=1 时，RS 位不会置位。当接收到一个新的帧时，如果描述符的 DAV 位为'1'，则重复上述的 RxDMA 控制器操作。如果描述符的 DAV 位为'0'，则 DMA 控制器进入挂起状态，并设置 ENET_DMA_STAT 寄存器的 RBU 位为'1'。记录描述符列表地址指针当前值，并在退出挂起状态后作为描述符开始的地址。

挂起状态时接收到新的帧时的处理

在挂起状态时，当接收到一个新的帧，并且满足转发条件时（转发条件如上所述），RxDMA 将获取帧的描述符。如果 RDES0 的 DAV 位为'1'，则 RxDMA 控制器退出挂起状态，返回运行状态开始接收帧。但当 RDES0 的 DAV 位为'0'，则应用程序可以通过配置 ENET_DMA_CTL 寄存器中 DAFRF 位来选择是否清空 RxFIFO 中的帧。如果 DAFRF=0，则 RxDMA 控制器将丢弃 FIFO 所接收的帧数据，并将丢失帧计数器 MSFC 加 1。若 DAFRF=1，则可以阻止丢弃 RxFIFO 顶部的帧，除非 RxFIFO 满，丢失帧计数器 MSFC 的值不会增加。在 DAV 位为'0'时，ENET_DMA_STAT 寄存器中的 RBU 位将被置位，RxDMA 控制器仍处于挂起状态。

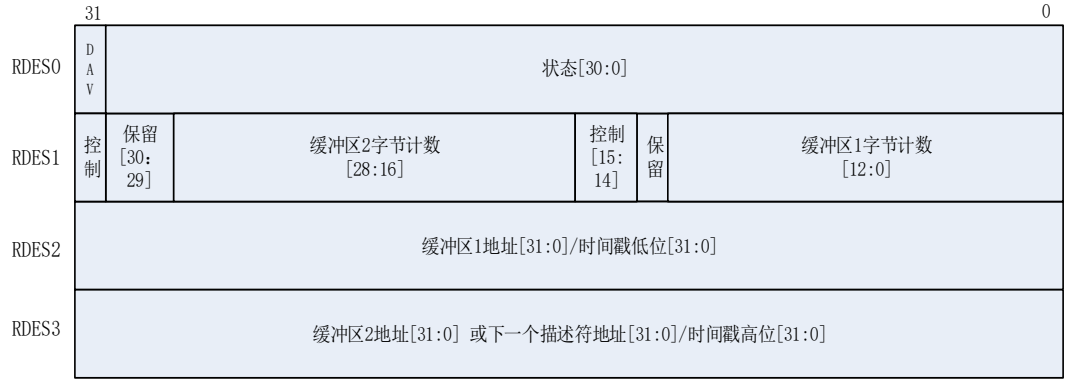
带 IEEE1588 时间戳的 RxDMA 描述符格式

如果使能了 IEEE 1588 功能，则 MAC 控制器会在带时间戳的帧接收完成之后，DMA 关闭描述符之前（DAV 位清'0'），将时间戳写入 RDES2 和 RDES3。

RxDMA 描述符

RxDMA 描述符结构体包含 4 个 32 位字，RDES0~RDES3。RDES0，RDES1，RDES2 和 RDES3 的位定义如下：

图 27-10. 接收描述符



■ RDES0：接收描述符字 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAV	DAFF	FRML[13:0]													
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRS	DERR	SAFF	LERR	OERR	VTAG	FDES	LDES	IPHERR/TSV	LCO	FRMT	RWDT	RERR	DBERR	CERR	PCERR/EXSV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	DAV	描述符转发条件位 该位指示了 DMA 控制器是否可以占有该描述符。DMA 会在帧完整接收或者描述符指向的缓存被填满以后把该位清'0' 0：表示 CPU 占有描述符 1：表示 DMA 占有描述符
30	DAFF	未通过目标地址过滤器位 0：接收帧通过目标地址过滤器 1：接收帧没有通过目标地址过滤器
29:16	FRML	帧长位 这些位表示了传送到输入缓存的接受帧字节长度。该位只在 LDES 位(RDES0[8])为'1'且描述符错误位 DERR(RDES0[14])为'0'时有效。若 LDES 位和错误汇总位 ERRS 位(RDES0[15])都为'0'，则这些位表示当前接收帧已经传送到内存里的累计字节数
15	ERRS	错误汇总位 该位只在 LDES 位(RDES0[8])为'1'时有效 该位为下列位的逻辑"或"： RDES0[14]：描述符错误 RDES0[11]：溢出错误 RDES0[6]：延迟冲突 RDES0[4]：看门狗超时 RDES0[3]：接收错误 RDES0[1]：CRC 错误 RDES0[7] = 0，RDES0[5] = 1 且 RDES0[0] = 1：数据校验和错误 RDES0[7] = 1，RDES0[5] = 1 且 RDES0[0] = 0：报头校验和错误 RDES0[7] = 1，RDES0[5] = 1 且 RDES0[0] = 1：报头及数据校验和错误
14	DERR	描述符错误位 该位在 LDES 位(RDES0[8])为'1'时才有效 当当前描述符的缓存大小小于接收帧帧长同时 DMA 又无法占有下一个描述符，将发生描述符错误 0：未发生描述符错误 1：发生了描述符错误
13	SAFF	未通过源地址过滤器位 0：未发生未通过源地址过滤器事件 1：帧没有通过源地址过滤器
12	LERR	长度错误位 该位仅在 RDES0[5]位 FRMT 为'0'时才有效 该位指示了接收到的以太网帧头长度/类型域的值与接收帧的实际长度不匹配 0：未发生长度错误 1：发生了长度错误

11	OERR	溢出错误位 当 RxFIFO 发生了溢出，而接收帧已有部分被传送到输入缓存时，溢出错误位置位 0：未发生溢出错误 1：发生了 RxFIFO 溢出，帧数据无效
10	VTAG	VLAN 标签位 0：接收帧非 VLAN 帧 1：当前的接收帧为 VLAN 帧
9	FDES	第一个描述符 该位表示当前描述符存放了接受帧的第一部分 0：当前描述符未存放帧的第一部分 1：当前描述符存放了帧的第一部分
8	LDES	最后一个描述符 该位表示当前描述符存放了接受帧最后一部分 0：当前描述符未存放帧的最后一部分 1：当前描述符存放了帧的最后一部分
7	IPHERR	IP 帧报头校验和错误 0：未发生 IPv 报头校验和错误 1：发生了 IPv4 或者 IPv6 的报头错误。错误可能是由于以太网类型域和 IP 版本域值不匹配，IPv4 报头校验和不对或者以太网帧的 IP 报头字节数不足
6	LCO	延迟冲突位 延迟冲突表示在接收到 64 字节数据后发生了冲突 该位仅在半双工模式下有效 0：未发生延迟冲突 1：接收帧的过程发生了延迟冲突
5	FRMT	帧类型 0：接收到的帧是 IEEE802.3 帧 1：接收到的帧是以太网类型的帧（以太网帧头长度/类型域的值大于等于 0x0600，当接收帧为过短帧时（帧长小于 14 字节）该位无效）
4	RWDT	接收看门狗超时 当 WDD=0，该位表示已接收到超过 2048 字节的帧数据 当 WDD=1，该位表示已接收到超过 16384 字节的帧数据 0：未发生接收看门狗超时 1：在接收帧的过程中发生了看门狗超时，当前接收帧将被截断
3	RERR	接收错误 该位表示帧接收过程中，在 RX_DV 信号有效时收到有效接口信号 RX_ER 0：未发生接收错误 1：发生了接收错误
2	DBERR	Dribble 位错误 该位只在 MII 模式下有效，表示接收到的帧长不是字节的整数倍

		0：未发生 Dribble 位错误 1：发生了 Dribble 位错误
1	CERR	CRC 错误 该位仅在 LDES 位(RDES0[8])为'1'时有效，表示接受帧的帧校验序列与硬件计算结果不匹配 0：未发生 CRC 错误 1：检测到接收帧发生了 CRC 错误
0	PCERR	数据校验和错误 0：未发生数据校验和错误 1：硬件计算的 TCP、UDP 或 ICMP 校验与接收到帧的 TCP、UDP 或 ICMP 的校验和域值不相符。 在接收到以太网帧的数据长度和 IPv4 或 IPv6 数据包长度域的值不符时，该位也会置'1'。

*表 27-6. 接收描述符 0 错误状态描述*显示了 RDES0 中位 7，5，0 取值的含义：

表 27-6. 接收描述符 0 错误状态描述

位7: IPHERR	位5: FRMT	位0: PCERR	帧状态
0	0	0	IEEE802.3类型帧（长度域值小于0x0600或非标签帧）
0	0	1	IPv4/IPv6类型帧，未检测到IP报头校验和错误；由于不支持的数据格式，未执行数据校验和检测
0	1	0	IPv4/IPv6类型帧，未检测到校验和错误
0	1	1	IPv4/IPv6类型帧，检测到数据校验和错误。 该错误可能原因： 1) 计算的校验和值与接收帧中的校验和域值不匹配 2) 接收帧长度域值与实际接收帧字节数不符
1	0	0	保留
1	0	1	类型帧（长度/类型域值大于等于0x600）或标签帧，但非IPv4/IPv6帧，不检测数据校验和
1	1	0	IPv4/IPv6类型帧，检测到报头校验和错误。 该错误可能原因： 1) 类型域值与数据包中的版本值不一致 2) 计算的报头校验和值与报头校验和域值不匹配 3) IP报头字节数不足
1	1	1	IPv4/IPv6类型帧，检测到IP报头校验和错误和数据校验和错误

■ RDES1：接收描述符字 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DINTC	保留		RB2S[12:0]												
rw									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RERM	RCHM	保留	RB1S[12:0]												

Rw rw rw

位/位域	名称	描述
31	DINTC	关闭接收完成中断 0：接收完成后 ENET_DMA_STAT 寄存器的 RS 位(位 6)会置 1，此时若使能了相应中断则会触发中断 1：在接收帧成功接收完成后，ENET_DMA_STAT 寄存器的 RS 位(位 6)不会置 1，这样相应中断也不会被触发
30:29	保留	必须保持复位值。
28:16	RB2S[12:0]	接收缓存 2 大小 表示接收缓存 2 的大小(以字节为单位)。缓存大小必须被设为 4 的倍数。这些位在 RDES1[14]位为'1'时被忽略
15	RERM	接收描述符环形结构结尾 该位表示到达的是描述符列表中的最后一个描述符，下个描述符自动返回列表的基地址 0：当前描述符不是最后一个描述符 1：到达描述符列表的最后一个描述符
14	RCHM	第二地址链表 0：描述符里的第二个地址指向第二缓存的地址 1：描述符里的第二个地址是下一个描述符的地址，该位仅在在链模式下使用。忽略 RBS2(RDES1[28:16])的值 **注意：**当 RERM=1，则即使该位置位，下个描述符也将返回列表基地址
13	保留	必须保持复位值。
12:0	RB1S[12:0]	接收缓存 1 大小 表示接收缓存 1 的大小(以字节为单位)。缓存大小必须被设为 4 的倍数。如果这些位为 0，DMA 忽略该缓存，直接取接收缓存 2 中的数据(RCHM=0)或者下一个描述符(RCHM=1)

■ RDES2：接收描述符字 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB1AP/RTSL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB1AP/RTSL[15:0]															
rw															

位/位域	名称	描述
31:0	RBAP1/RTSL[31:0]	接收缓存 1 地址指针/发送帧时间戳低 32 位 这些位有 2 个功能：缓冲区 1 的地址指针(RB1AP)和时间戳低 32 位(RTSL)。 RB1AP：在 DMA 控制器获取该描述符之前，配置该位为缓存区 1 的地址。如果位 RB1S 不为'0'，则用该位的地址来存储接收的数据帧。对缓存的地址对齐不做限制。 RTSL：当使能了时间戳功能同时 LDES 位为'1'时，如果接收帧通过了地址过滤，

并且置位了对应的帧类型使能位，则 DMA 会将时间戳低 32 位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持 RB1AP 的值。

■ RDES3：接收描述符字 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB2AP/RTSH[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB2AP/RTSH[15:0]															
rw															

位/位域	名称	描述
31:0	RB2AP/RTSH[31:0]	接收缓存 2 地址指针(或者下一个描述符地址指针)/接收帧时间戳高 32 位 这些位有 2 个功能：数据接收时这些位存放缓冲区 2 的地址或下个描述符的地址(RB2AP)，或帧时间戳高 32 位(RTSH) RB2AP：在 DMA 控制器获取该描述符之前，配置该位为缓存区 2 的地址(RCHM=0)或下个描述符地址(RCHM=1)。在 RCHM=0 时，如果位 RB1S 不为'0'，则用该位的地址来存储接收的数据帧。如果 RCHM=1，且 RERM=1，则忽略这些位。 如果该地址表示下个描述符地址，则地址需为字对齐的，没有其他的限制条件。 RTSH：当使能了时间戳功能同时帧的最后部分 LDES 位为'1'时，如果接收帧通过了地址过滤，并且置位了对应的帧类型使能位，则 DMA 会将时间戳高 32 位写入这些位。如果接收帧没有置位对应的帧类型使能位，则这些位会保持 RB2AP 的值。

27.3.7. 典型的以太网配置流程示例

在上电复位或系统复位之后，应用程序可按以下的典型操作流程来配置并启动以太网模块：

■ 使能以太网时钟：

配置RCU模块来使能HCLK时钟和以太网发送/接收时钟。

■ 配置通讯接口：

配置AFIO_PCF0，选择接口模式(MII或RMII)；
配置GPIO模块，将相应的功能脚映射到复用功能上。

■ 等待复位完成：

轮询 ENET_DMA_BCTL 寄存器直到 SWR 位复位（SWR 位在上电复位后或系统复位后默认置位）。

■ 获取并配置PHY寄存器参数：

根据 HCLK 频率，配置 SMI 时钟频率，并访问 PHY 寄存器获取 PHY 的信息（例如是否支持半/全双工，是否支持 10M/100Mbit 速度等等）。根据外部 PHY 支持的模式，配置 ENET_MAC_CFG 寄存器使与 PHY 寄存器信息一致。

- 初始化以太网DMA模块用于数据传输：

  配置 ENET_DMA_BCTL，ENET_DMA_RDTADDR，ENET_DMA_TDTADDR 和 ENET_DMA_CTL 寄存器，完成 DMA 模块初始化（详细信息请参考 ***DMA 控制器描述***章节）。

- 初始化用于存放描述符列表以及数据缓存的物理内存空间：

  根据 ENET_DMA_RDTADDR 和 ENET_DMA_TDTADDR 寄存器中的地址，初始化发送和接收描述符(DAV=1)，以及数据缓存。

- 使能MAC和DMA模块，开始发送和接收：

  置位 ENET_MAC_CFG 寄存器中的 TEN 和 REN 位，开启 MAC 发送器和接收器。置位 ENET_DMA_CTL 寄存器中的 STE 位和 SRE 位，使能 DMA 的发送和接收。

- 如果有帧要发送：
  1) 选择一个或多个描述符发送描述符，将发送帧数据写到 TDES 中指定的缓存地址中；
  2) 将这些发送帧描述符中的 DAV 位置位；
  3) 写入任意值到 ENET_DMA_TPEN 寄存器中，使 TxDMA 退出挂起模式，开始发送数据；
  4) 有两种方法来确定当前帧是否发送完毕。第一种方法为轮询当前描述符的 DAV 位直到其复位；第二种方法仅适用于当 INTC 位为 1 的情况，应用程序可以轮询 ENET_DMA_STAT 寄存器的 TS 位直到其置位。
- 如果有帧要接收：
  1) 查看描述符列表中的第一个接收描述符（其地址在 ENET_DMA_RDTADDR 寄存器中配置）；
  2) 如果 RDES0 的 DAV 位复位，则说明描述符已被使用过，且接收缓存空间已存储了接收帧；
  3) 处理接收帧数据；
  4) 置位当前描述符的 DAV 位，以复用当前描述符接收新的帧；
  5) 查看列表中的下一个描述符，跳到步骤 2。

27.3.8. 以太网中断

以太网部分一共有 2 个中断向量，一个用于以太网正常操作，另一个用于映射到 EXTI 线 19 的以太网唤醒事件(检测唤醒帧或者 Magic Packet)。

第一个中断向量用于由 MAC 和 DMA 产生的中断。关于 ***MAC 中断***和 ***DMA 中断***的详细介绍请看下文。

第二个中断向量用于 WUM 模块在唤醒事件时产生的中断，唤醒事件为远程唤醒帧接收事件或 Magic Packet 唤醒帧接收事件。唤醒事件映射到 EXTI 线 19 上，若使能了 EXTI 线 19 的上升沿中断，则唤醒事件可以使微控制器退出深度睡眠模式。此外，若使能了 WUM 中断，则 EXTI 线 19 中断和以太网中断都会被触发。

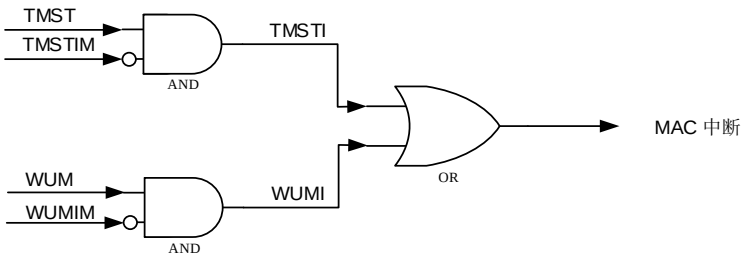
注意：由于 WUM 寄存器位于 RX_CLK 域，在应用程序读 WUM 寄存器后，到这些标志位被

清除可能会有可观的延迟（延迟由 HCLK 和 RX_CLK 时钟频率之间差异决定）。为避免两次进入同一个中断，强烈建议应用程序在中断里等待唤醒帧接收标志位 WUFR 和 Magic Packet 接收标志位 MPKR 变为 0 后，再退出中断服务程序。

MAC 中断

MAC 控制器有多个中断触发源。ENET_MAC_INTF 寄存器描述了所有可产生的 MAC 中断类型，每个位都有对应的中断屏蔽位来防止某一事件引发中断。MAC 中断信号为 MAC 所有中断的逻辑或。

图 27-11. MAC 控制器中断示意图



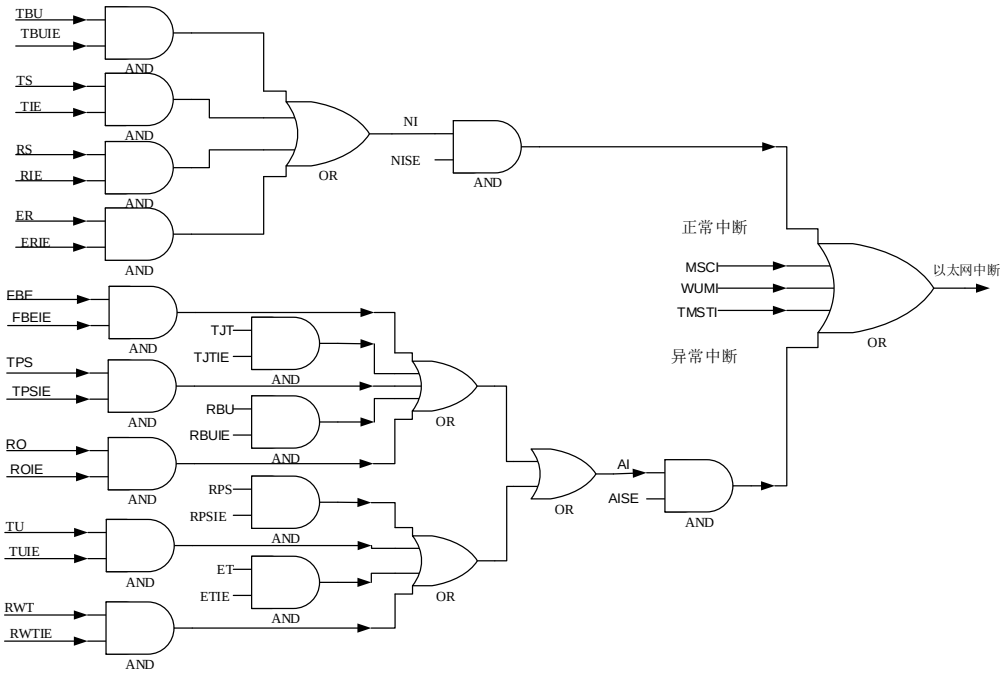
DMA 中断

DMA 控制器有两种类型中断事件：正常类和异常类。

无论什么类型的中断事件，都具有相应的中断使能位（屏蔽位）来控制是否产生中断。当所有中断事件都被清除，或中断使能位被清除，则相应的中断汇总位也被清除。如果正常类和异常类中断都被清除，则 DMA 中断将被清除。

*图27-12. 以太网中断示意图*示意了以太网模块的中断连接：

图 27-12. 以太网中断示意图



27.4. ENET 寄存器

ENET基地址：0x4002 8000

27.4.1. MAC 配置寄存器（ENET_MAC_CFG）

地址偏移：0x0000
复位值：0x0000 8000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
MAC配置寄存器是MAC的工作模式寄存器。它定义了接收和发送的工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								WDD	JBD	保留		IGBS[2:0]			CSD
								rw	rw				rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPD	ROD	LBM	DPM	IPFCO	RTD	保留	APCD	BOL[1:0]		DFC	TEN	REN	保留	
	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw		

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	WDD	关闭看门狗 该位表示已接收到了最大字节数的数据，超过的部分将被切断 0：MAC允许接收不超过2048字节的帧 1：MAC关闭接收看门狗定时器，此时最多可接收16384字节的帧
22	JBD	不检测Jabber 该位表示发送帧最大允许的发送字节数，超过的部分将被截断 0：MAC允许的最大发送字节数为2048字节 1：MAC关闭发送Jabber定时器，此时最多可发送16384字节的帧
21:20	保留	必须保持复位值。
19:17	IGBS[2:0]	帧间间隙选择位 这些位用于选择2个相邻发送帧之间的最短发送间隙 0x0：96位时间 0x1：88位时间 0x2：80位时间 0x3：72位时间 0x4：64位时间 0x5：56位时间（半双工模式下不可用） 0x6：48位时间（半双工模式下不可用） 0x7：40位时间（半双工模式下不可用）
16	CSD	关闭载波侦听功能

		0：MAC载波信号错误时会报错，并终止发送 1：在半双工模式下，MAC在发送帧过程中忽略MII的CRS信号，发送过程中载波丢失或者没有载波都不会报错
15	保留	必须保持复位值。
14	SPD	快速以太网速度 该位表示快速以太网模式下的速度： 0：10 Mbit/s 1：100 Mbit/s
13	ROD	关闭自接收功能 该位在全双工模式下无意义 0：MAC在发送时接收所有来自PHY的数据包 1：MAC在半双工模式下不接受帧
12	LBM	回环模式 0：MAC在普通模式下工作 1：MAC在MII的回环模式下工作
11	DPM	双工模式 0：半双工模式使能 1：全双工模式使能
10	IPFCO	IP帧数据校验和 0：禁止接收端TCP/UDP/ICMP报头的校验和检验功能 1：使能接收端的帧数据校验和检测功能
9	RTD	不尝试重试 该位只在半双工模式下有效 0：MAC会在发生冲突后按照BOL位的设定重发高达16次 1：帧仅发送一次
8	保留	必须保持复位值
7	APCD	自动填充/CRC剥离 该位仅在非标签帧，且其长度域值小于等于1536时有效 0：MAC会转发所有接收到的帧，而不改变帧的内容 1：MAC会去除帧的填充字节和CRC域
6:5	BOL[1:0]	退后限制 在发生冲突后，MAC在重发当前帧之前需要延迟一段时间。这个延迟时间(dt)的时基单元称为时间间隙，一个时间间隙为512位时间。这个延迟时间(dt)是由下式计算得的随机整数值：$0 \leq dt < 2^k$ 0x0：k = min(n, 10) 0x1：k = min(n, 8) 0x2：k = min(n, 4) 0x3：k = min(n, 1) 其中n = 重发次数

		注意：这些位只在半双工模式下有效。
4	DFC	顺延检验 0：禁止MAC顺延检验功能。MAC会延迟发送直到CRS信号失效。 1：MAC顺延检验功能使能。如果延迟超过24288位时间，则会发生过度顺延错误，并且MAC将中止发送。但如果在顺延时间内检测到有效的CRS(载波侦听)信号，则会将顺延计数器重置为0，重新启动顺延计时 **注意：**该位只在半双工模式下有效
3	TEN	使能发送器 0：MAC关闭发送状态机，若当前帧正在发送则在完成发送后关闭 1：MAC使能发送状态机
2	REN	使能接收器 0：MAC关闭接收状态机，若当前帧正在接收则在接收完成后关闭 1：MAC使能接收状态机
1:0	保留	必须保持复位值。

27.4.2. MAC 帧过滤器寄存器（ENET_MAC_FRMF）

地址偏移：0x0004
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。
MAC帧过滤器寄存器包含了接收帧的过滤模式位。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					HPFLT	SAFLT	SAIFLT	PCFRM[1:0]		BFRMD	MFD	DAIFLT	HMF	HUF	PM
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	FAR	接收所有帧 该位控制帧过滤器功能 0：只有通过了地址过滤器的接收帧才会被转发给应用程序 1：所有接收到的帧都会被转发给应用程序，但过滤的结果会反映在更新接收描述符状态信息的相应标志位
30:11	保留	必须保持复位值。
10	HPFLT	HASH或者完美过滤 0：如果HMF位或者HUF位置'1'，符合HASH过滤器的帧才能通过接收地址过滤 1：如果HMF位或者HUF位置'1'，接收帧通过HASH过滤器或者完美过滤器中任一种，就认为通过接收地址过滤

9	SAFLT	源地址过滤器 除了目标地址过滤之外，使能源地址过滤器 过滤器将接收帧的源地址域值与使能的源地址寄存器中配置的值进行比较。如果源地址值相匹配，则接收描述符中的源地址匹配状态位将置位。 0：源地址过滤器关闭 1：源地址过滤器使能
8	SAIFLT	源地址过滤结果逆转 该位将源地址比较结果逆转 0：仅在源地址过滤器结果逆转 1：使能源地址过滤器结果逆转，所有源地址符合源地址寄存器的帧会被标记为未通过源地址过滤
7:6	PCFRM[1:0]	控制帧转发位 这些位用于设置所有控制帧的转发条件（包括单播和多播暂停帧） 对于是否处理暂停控制帧，只取决于RFCEN位(ENET_MAC_FCTL[2])的值 0x0：MAC不转发任何控制帧给应用程序 0x1：MAC转发除了暂停帧以外的其他控制帧给应用程序 0x2：MAC转发所有的控制帧给应用程序，即使是没通过地址过滤器的控制帧 0x3：MAC转发通过地址过滤器的控制帧给应用程序
5	BFRMD	不接收广播帧 0：过滤器接收所有广播帧 1：过滤器不接收所有广播帧
4	MFD	关闭多播过滤器 0：是否对多播帧进行过滤，取决于HMF位的取值 1：所有的带多播目标地址的帧(帧的目标地址域中第一位为’1’，但不是所有位都为’1’)都能通过过滤器
3	DAIFLT	目标地址过滤结果逆转 该位将目标地址过滤结果逆转 0：禁用目标地址过滤结果逆转 1：使能目标地址过滤结果逆转
2	HMF	多播HASH过滤器 0：MAC会将接收到的多播帧的目标地址域和目标地址寄存器的设定值比较 1：MAC根据HASH列表对接收到的多播帧进行目标地址过滤
1	HUF	单播HASH过滤器 0：MAC会将接收到的单播帧目标地址域和目标地址寄存器的设定值比较 1：MAC根据HASH列表对接收到的单播帧进行目标地址过滤
0	PM	混杂模式 该位使地址过滤器无效，这意味着所有帧均可通过过滤器，同时接收描述符中状态信息的目标地址/源地址错误位总是为’0’ 0：禁用混杂模式 1：使能混杂模式

27.4.3. MAC hash 列表高寄存器（ENET_MAC_HLH）

地址偏移：0x0008
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	HLH[31:0]	HASH列表高位 这些位是HASH列表的高32位

27.4.4. MAC hash 列表低寄存器（ENET_MAC_HLL）

地址偏移：0x000C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	HLL[31:0]	HASH列表低位 这些位是HASH列表的低32位

27.4.5. MAC PHY 控制寄存器（ENET_MAC_PHY_CTL）

地址偏移：0x0010
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA[4:0]					PR[4:0]					保留	CLR[2:0]			PW	PB
		rw					rw					rw		rw	rc_w1

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:11	PA[4:0]	PHY地址 这些位选择想要访问的PHY地址
10:6	PR[4:0]	PHY寄存器 这些位选择想要访问的PHY寄存器
5	保留	必须保持复位值。
4:2	CLR[2:0]	时钟范围 根据HCLK的频率来决定MDC的时钟分频系数 0x0：HCLK/42 （HCLK范围：60-100 MHz） 0x1：HCLK/62 （HCLK范围：100-120 MHz） 0x2：HCLK/16 （HCLK范围：20-35 MHz） 0x3：HCLK/26 （HCLK范围：35-60 MHz） 其他：保留 -
1	PW	PHY写 该位指示了PHY的操作模式 0：对PHY进行读操作 1：对PHY进行写操作
0	PB	PHY忙 该位指示了对PHY操作的状态。由应用程序置'1'后开始对PHY的进行读或者写操作，并需等到该位在操作完成后由硬件清'0'。在写ENET_MAC_PHY_CTL寄存器和读ENET_MAC_PHY_DATA寄存器之前，该位应当为'0'

27.4.6. MAC PHY 数据寄存器（ENET_MAC_PHY_DATA）

地址偏移：0x0014

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述

31:16	保留	必须保持复位值。
15:0	PD[15:0]	PHY数据位 对于读操作，这些位为从PHY中读取的数据。对于写操作，这些位为将要写到PHY中的数据

27.4.7. MAC 流控寄存器（ENET_MAC_FCTL）

地址偏移：0x0018
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器用于配置控制帧的生成和接收。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DZQP	保留	PLTS[1:0]		UPFDT	RFCEN	TFCEN	FLCB/BKPA
								rw		rw		rw	rw	rw	rw

位/位域	名称	描述
31:16	PTM[15:0]	暂停时间 这些位用来设置暂停控制帧时间域的值
15:8	保留	必须保持复位值。
7	DZQP	关闭零时间片暂停功能 0：打开零时间片暂停控制帧自动生成功能 1：关闭零时间片暂停控制帧的自动生成
6	保留	必须保持复位值。
5:4	PLTS[1:0]	暂停低阈值 这些位设置了自动重发暂停帧的定时器阈值。这个阈值应当大于0，小于位[31:16]定义的暂停时间。低阈值的计算公式为PTM-PLTS。例如，PTM = 0x80（128个时间间隙），PLTS = 0x1（28个时间间隙），那么在第一个暂停帧发出100(128-28)个时间间隙后，将自动重发第二个暂停帧。 0x0：暂停时间 – 4个时间间隙 0x1：暂停时间 –28个时间间隙 0x2：暂停时间 – 144个时间间隙 0x3：暂停时间 – 256个时间间隙 **注意：**一个时间间隙是指MII接口发送512位(64字节)数据所需要的时间
3	UPFDT	单播暂停帧检测 0：MAC只接收符合IEEE802.3规范定义的唯一多播地址的暂停帧 1：除了唯一多播地址的暂停帧，MAC同时还会使用MAC0地址

（ENET_MAC_ADDR0H寄存器和ENET_MAC_ADDR0L寄存器）来检测暂停帧

2	RFCEN	接收流控使能位 0：MAC不解析暂停帧 1：MAC解析并处理接收到的暂停帧。MAC关闭发送器一段指定的时间（接收帧中的暂停时间域值）
1	TFCEN	发送流控使能位 0：MAC关闭发送流控功能。在全双工模式下，MAC不发送暂停帧；在半双工模式下，MAC关闭背压功能 1：MAC开启发送流控功能。在全双工模式下，MAC使能暂停帧发送；在半双工模式下，MAC使能背压功能
0	FLCB/BKPA	流控忙/背压激活 该位仅在TFCEN位置位时有效 在全双工模式下，该位可发送暂停帧；在半双工模式下，该位可激活背压功能。 在全双工模式下，应用程序要确保在写ENET_MAC_FCTL寄存器之前该位为'0'。置位该位后，MAC将发送一个暂停帧到接口，在发送控制帧的过程中，该位始终为'1'，直到暂停控制帧发送完成以后，MAC将该位重置为'0'。 在半双工模式下，设置该位为'1'可以激活背压功能。在背压功能有效时，如果MAC接收到新的帧，就会在发送端发送阻塞信号，通知有冲突发生。

27.4.8. MAC 流控阈值寄存器（ENET_MAC_FCTH）

地址偏移：0x1080
复位值：0x0000 0015

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	RFD[2:0]	流控失效阈值 这些位设置了流控失效的阈值。这个值应当小于位[2:0]定义的流控激活阈值。当RxFIFO中未处理的数据低于这些位所设置的值，流控功能将自动失效。 0x0：256字节 0x1：512字节 0x2：768字节 0x3：1024字节 0x4：1280字节 0x5：1536字节

		0x6,0x7：1792字节
3	保留	必须保持复位值。
2:0	RFA[2:0]	流控激活阈值 这些位设置了流控激活的阈值。若使能了流控功能，当RxFIFO中未处理的数据超过了这些位所设置的值，流控功能将被激活。 0x0：256字节 0x1：512字节 0x2：768字节 0x3：1024字节 0x4：1280字节 0x5：1536字节 0x6,0x7：1792字节

27.4.9. MAC VLAN 标签寄存器（ENET_MAC_VLT）

地址偏移：0x001C
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。
该寄存器包含了用来识别VLAN帧的IEEE802.1Q VLAN标签。MAC把接收到帧的第13，14字节(长度/类型域)与0x8100比较，再把之后的2个字节（第15，16字节）和VLAN标签比较。

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

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

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	VLTC	12位VLAN标签比较位 该位选择用12位或16位VLAN标签来进行比较 0：接收到的VLAN帧的全部16位数据（第15和16字节）都用来与VLTI位比对 1：仅用VLTI位[11:0]12位数据和接收到VLAN帧的相应域比对
15:0	VLTI[15:0]	VLAN标签标识符位 这些位用来识别VLAN帧的802.1Q VLAN标签格式。格式如下： VLTI[15:13]：UP（用户优先级） VLTI[12]：CFI（标准格式指示符） VLTI[11:0]：VID（VLAN标识符） 如果比较的位（当VLTC=1，则为VLTI[11:0]；当VLTC=0，则为VLTI[15:0]）值是全'0'，则MAC不再比对检验VLAN帧的第15、16字节，并将接收帧的类型域值是0x8100的帧都直接视为VLAN帧。

如果用于比较的位不是全为'0'，则使用VLTI[11:0]（VLTC=1）或VLTI[15:0]（VLTC=0）进行比较。

27.4.10. MAC 远程唤醒帧过滤器寄存器（ENET_MAC_RWFF）

地址偏移：0x0028
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器实质上是指向8个不透明的唤醒帧过滤器寄存器的指针（使用同一个偏移地址）。对该寄存器地址(偏移为0x0028)的8次连续写操作，可以写入全部8个唤醒帧过滤器寄存器；对该寄存器地址（偏移为0x0028）的8次连续读操作，可以读出全部8个唤醒帧过滤器寄存器。

图 27-13. 远程唤醒帧过滤器寄存器

	31							0
Wakeup frame filter reg0	Byte Mask of Filter-0							
Wakeup frame filter reg1	Byte Mask of Filter-1							
Wakeup frame filter reg2	Byte Mask of Filter-2							
Wakeup frame filter reg3	Byte Mask of Filter-3							
Wakeup frame filter reg4	Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
Wakeup frame filter reg5	Offset of Filter 3		Offset of Filter 2		Offset of Filter 1		Offset Filter 0	
Wakeup frame filter reg6	Filter 1 CRC - 16				Filter 0 CRC - 16			
Wakeup frame filter reg7	Filter 3 CRC - 16				Filter 2 CRC - 16			

27.4.11. MAC 唤醒管理寄存器（ENET_MAC_WUM）

地址偏移：0x002C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器设置并监控唤醒事件。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						GU	保留		WUFR	MPKR	保留		WFEN	MPEN	PWD
						rw			rc_r	rc_r			rw	rw	rs

位/位域	名称	描述
31	WUFFRPR	唤醒帧过滤器寄存器指针复位

		向该位写'1'，将会把远程唤醒帧过滤器寄存器指针ENET_MAC_RWFF复位，该位在指针复位完成后自动清'0' 0：无作用 1：复位ENET_MAC_RWFF寄存器指针
30:10	保留	必须保持复位值。
9	GU	全局单播 向该位写1，所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧 0：不是所有接收的单播帧都被认为是唤醒帧 1：所有能通过MAC地址过滤器的单播帧，都被认为是唤醒帧
8:7	保留	必须保持复位值。
6	WUFR	接收到唤醒帧 读本寄存器可以清'0'该位 0：没有接收到唤醒帧 1：接收到唤醒帧，并发生唤醒事件
5	MPKR	接收到Magic Packet 读本寄存器可以清'0'该位 0：没有接收到Magic Packet 1：接收到Magic Packet帧，并发生唤醒事件
4:3	保留	必须保持复位值。
2	WFEN	唤醒帧使能位 0：禁能在接收到唤醒帧时产生唤醒事件 1：使能在接收到唤醒帧时产生唤醒事件
1	MPEN	Magic Packet使能位 0：禁能在接收到Magic Packet唤醒帧时产生唤醒事件 1：使能在接收到Magic Packet唤醒帧时产生唤醒事件
0	PWD	低功耗位 该位由软件置位，由硬件复位。当该位置位，MAC丢弃所有接收到的帧。当发生了唤醒事件，使得退出低功耗模式，硬件会自动将该位清'0'

27.4.12. MAC 中断状态寄存器（ENET_MAC_INTF）

地址偏移：0x0038

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TMST	保留		MSCT	MSCR	MSC	WUM	保留		

rc_r r r r r

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TMST	时间戳触发状态 读 ENET_PTP_TSF 寄存器可以清'0'该位 0：系统时间值小于期望时间值 1：系统时间值等于或者超过期望时间值
8:7	保留	必须保持复位值。
6	MSCT	MSC 发送状态 0：没有产生任一 ENET_MSC_TINTF 寄存器中的中断 1：产生任一 ENET_MSC_TINTF 寄存器中的中断
5	MSCR	MSC 接收状态 0：没有产生任一 ENET_MSC_RINTF 寄存器中的中断 1：产生任一 ENET_MSC_RINTF 寄存器中的中断
4	MSC	MSC 状态 该位为 MSCT 位与 MSCR 位的逻辑或 0：MSCT 位和 MSCR 位均为'0' 1：MSCT 位和 MSCR 位中有位为'1'
3	WUM	WUM 状态 该位为 ENET_MAC_WUM 寄存器中的 WUFR 和 MPKR 位的逻辑或 0：未接收到唤醒帧或者 Magic Packet 帧 1：在低功耗模式下，接收到唤醒帧或者 Magic Packet
2:0	保留	必须保持复位值。

27.4.13. MAC 中断屏蔽寄存器（ENET_MAC_INTMSK）

地址偏移：0x003C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述

31:10	保留	必须保持复位值。
9	TMSTIM	时间戳触发中断屏蔽位 0：允许产生时间戳中断 1：禁止产生时间戳中断
8:4	保留	必须保持复位值。
3	WUMIM	WUM中断屏蔽位 0：允许由于ENET_MAC_INTF寄存器的WUM状态位置位而引发的中断 1：禁止由于 ENET_MAC_INTF 寄存器的 WUM 状态位置 1 而引发的中断
2:0	保留	必须保持复位值。

27.4.14. MAC 地址 0 高寄存器（ENET_MAC_ADDR0H）

地址偏移：0x0040
复位值：0x8000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0H[15:0]															
rw															

位/位域	名称	描述
31	MO	该位总是为'1'
30:16	保留	必须保持复位值。
15:0	ADDR0H[15:0]	MAC地址0高16位 这些位包含了 6 字节 MAC 地址 0 的高 16 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址

27.4.15. MAC 地址 0 低寄存器（ENET_MAC_ADDR0L）

地址偏移：0x0044
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0L[31:16]															
rw															

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

ADDR0L[15:0]
rw

位/位域	名称	描述
31:0	ADDR0L[31:0]	MAC地址0低32位 这些位包含了 6 字节 MAC 地址 0 的低 32 位，这些位用于作为接收帧的地址过滤，还用于发送流控中发送暂停帧时插入作为帧的源地址

27.4.16. MAC 地址 1 高寄存器（ENET_MAC_ADDR1H）

地址偏移：0x0048
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 1 进行地址过滤 1：地址过滤器使用 MAC 地址 1 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 1[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 1[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 1 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR1H [15:8] MB[4]：ENET_MAC_ADDR1H [7:0] MB[3]：ENET_MAC_ADDR1L [31:24] MB[2]：ENET_MAC_ADDR1L[23:16] MB[1]：ENET_MAC_ADDR1L[15:8] MB[0]：ENET_MAC_ADDR1L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR1H[15:0]	MAC 地址 1 高[47:32]位 这些位包含了 6 字节的 MAC 地址 1 的高 16 位

27.4.17. MAC 地址 1 低寄存器（ENET_MAC_ADDR1L）

地址偏移：0x004C
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR1L[31:0]	MAC地址1低32位 这些位包含了6字节MAC地址1的低32位

27.4.18. MAC 地址 2 高寄存器（ENET_MAC_ADDR2H）

地址偏移：0x0050
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 2 进行地址过滤 1：地址过滤器使用 MAC 地址 2 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 2[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 2[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 2 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR2H [15:8] MB[4]：ENET_MAC_ADDR2H [7:0]

		MB[3]：ENET_MAC_ADDR2L [31:24] MB[2]：ENET_MAC_ADDR2L[23:16] MB[1]：ENET_MAC_ADDR2L[15:8] MB[0]：ENET_MAC_ADDR2L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR2H[15:0]	MAC 地址 2 高[47:32]位 这些位包含了 6 字节的 MAC 地址 2 的高 16 位

27.4.19. MAC 地址 2 低寄存器（ENET_MAC_ADDR2L）

地址偏移：0x0054
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR2L[31:0]	MAC地址2低32位 这些位包含了6字节MAC地址2的低32位

27.4.20. MAC 地址 3 高寄存器（ENET_MAC_ADDR3H）

地址偏移：0x0058
复位值：0x0000 FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFE	SAF	MB[5:0]						保留							
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3H[15:0]															
rw															

位/位域	名称	描述
31	AFE	地址过滤使能 0：不使用 MAC 地址 3 进行地址过滤

		1：地址过滤器使用 MAC 地址 3 来进行完美过滤
30	SAF	源地址过滤器 0：MAC 地址 3[47:0]用来和接收帧的目标地址进行比对 1：MAC 地址 3[47:0]用来和接收帧的源地址进行比对
29:24	MB[5:0]	屏蔽字节位 当某个位置'1'时，MAC 不再把接收帧目标地址/源地址的对应字节与 MAC 地址 3 的相应字节进行比较。每个控制位对应的 MAC 地址字节如下： MB[5]：ENET_MAC_ADDR3H [15:8] MB[4]：ENET_MAC_ADDR3H [7:0] MB[3]：ENET_MAC_ADDR3L [31:24] MB[2]：ENET_MAC_ADDR3L[23:16] MB[1]：ENET_MAC_ADDR3L[15:8] MB[0]：ENET_MAC_ADDR3L [7:0]
23:16	保留	必须保持复位值。
15:0	ADDR3H[15:0]	MAC 地址 3 高[47:32]位 这些位包含了 6 字节的 MAC 地址 3 的高 16 位

27.4.21. MAC 地址 3 低寄存器（ENET_MAC_ADDR3L）

地址偏移：0x005C
复位值：0xFFFF FFFF

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3L[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR3L[31:0]	MAC地址3低32位 这些位包含了6字节MAC地址3的低32位

27.4.22. MSC 控制寄存器（ENET_MSC_CTL）

地址偏移：0x0100
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												MCFZ	RTOR	CTSR	CTR
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	MCFZ	MSC 计数器冻结位 0：MSC 计数器正常工作 1：冻结MSC计数器，保持它们的当前值。RTOR位可在计数器冻结状态时工作
2	RTOR	读时复位 0：读 MSC 计数器后，计数器不复位 1：读MSC计数器后，计数器复位
1	CTSR	计数器停止回转 0：计数器在计数到最大值后，会重新从 0 开始计数 1：计数器在计数到最大值后，不会重新从0开始计数
0	CTR	计数器复位 该位置位后，会在 1 个时钟周期后由硬件自动清零 0：无作用 1：复位所有计数器

27.4.23. MSC 接收中断状态寄存器（ENET_MSC_RINTF）

地址偏移：0x0104
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	RGUF	接收到”好”的单播帧 0：接收“好”单播帧计数器值小于最大值的一半 1：接收“好”单播帧计数器值达到最大值的一半

16:7	保留	必须保持复位值。
6	RFAE	接收到帧对齐错误 0：对齐错误接收帧计数器值小于最大值的一半 1：对齐错误接收帧计数器值达到最大值的一半
5	RFCE	接收到帧 CRC 错误 0：CRC 错误接收帧计数器值小于最大值的一半 1：CRC错误接收帧计数器值达到最大值的一半
4:0	保留	必须保持复位值。

27.4.24. MSC 发送中断状态寄存器（ENET_MSC_TINTF）

地址偏移：0x0108
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	TGF	发送”好”的帧 0：发送“好”单播帧计数器值小于最大值的一半 1：发送“好”单播帧计数器值达到最大值的一半
20:16	保留	必须保持复位值。
15	TGFMSC	发送”好”的帧时遇到 1 个以上冲突 0：1 次以上冲突后发送”好”帧计数器值小于最大值的一半 1：1次以上冲突后发送”好”帧计数器值达到最大值的一半
14	TGFSC	发送”好”的帧时仅遇到 1 个冲突 0：1 次冲突后发送”好“帧计数器值小于最大值的一半 1：1次冲突后发送”好”帧计数器值达到最大值的一半
13:0	保留	必须保持复位值。

27.4.25. MSC 接收中断屏蔽寄存器（ENET_MSC_RINTMSK）

地址偏移：0x010C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器包含当接收统计计数器达到其最大值的一半时所产生的中断的屏蔽位。

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

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

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	RGUFIM	接收到”好”的单播帧的中断屏蔽位 0：不屏蔽当 RGUF 位为’1’时发生的中断 1：屏蔽当RGUF位为’1’时发生的中断
16:7	保留	必须保持复位值。
6	RFAEIM	接收帧对齐错误中断屏蔽位 0：不屏蔽当 RFAE 位为’1’时发生的中断 1：屏蔽当RFAE位为’1’时发生的中断
5	RFCEIM	接收帧 CRC 错误中断屏蔽位 0：不屏蔽当 RFCE 位为’1’时发生的中断 1：屏蔽当RFCE位为’1’时发生的中断
4:0	保留	必须保持复位值。

27.4.26. MSC 发送中断屏蔽寄存器（ENET_MSC_TINTMSK）

地址偏移：0x0110
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器可以设置相应中断的屏蔽位。

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

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

位/位域	名称	描述

31:22	保留	必须保持复位值。
21	TGFIM	发送"好"的帧的中断屏蔽位 0：不屏蔽当 TGF 位为'1'时发生的中断 1：屏蔽当 TGF 位为'1'时发生的中断
20:16	保留	必须保持复位值。
15	TGFMSCIM	遇到 1 个以上冲突后发送"好"帧中断屏蔽位 0：不屏蔽当 TGFMSC 位为'1'时发生的中断 1：屏蔽当 TGFMSC 位为'1'时发生的中断
14	TGFSCIM	仅遇到 1 个冲突后发送"好"帧中断屏蔽位 0：不屏蔽当 TFGSC 位为'1'时发生的中断 1：屏蔽当 TFGSC 位为'1'时发生的中断
13:0	保留	必须保持复位值。

27.4.27. MSC 1 次冲突后发送"好"帧的计数器寄存器（ENET_MSC_SCCNT）

地址偏移：0x014C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计在半双工模式下，在只遇到一次冲突后发送帧成功时的帧的数目。

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

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

位/位域	名称	描述
31:0	SCC[31:0]	1 次冲突后发送好帧计数器 这些位是 1 次冲突后发送的"好"帧的计数器

27.4.28. MSC 1 次以上冲突后发送"好"帧的计数器寄存器（ENET_MSC_MSCCNT）

地址偏移：0x0150
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计在半双工模式下，遇到一次以上冲突后发送帧成功时的帧的数目。

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

r

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

r

位/位域	名称	描述
31:0	MSCC[31:0]	1 次以上冲突后发送”好”帧计数器 这些位是 1 次以上冲突后发送”好”帧的计数器

27.4.29. MSC 发送”好”帧计数器寄存器（ENET_MSC_TGFCNT）

地址偏移：0x0168
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计发送”好”帧的数目。

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

r

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

r

位/位域	名称	描述
31:0	TGF[31:0]	发送”好”帧计数器 这些位是发送“好”帧的计数器

27.4.30. MSC CRC 错误接收帧计数器寄存器（ENET_MSC_RFCECNT）

地址偏移：0x0194
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计接收帧中有CRC错误的帧的数目。

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

r

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

r

位/位域	名称	描述

31:0	RFCER[31:0]	CRC 错误接收帧计数器 这些位是接收帧中有 CRC 错误的帧的计数器

27.4.31. MSC 对齐错误接收帧计数器寄存器（ENET_MSC_RFAECNT）

地址偏移：0x0198
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计接收帧中有对齐错误帧的数目。

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

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

位/位域	名称	描述
31:0	RFAER[31:0]	对齐错误接收帧计数器 这些位是接收帧中有对齐错误的帧的计数器

27.4.32. MSC“好”单播帧接收帧计数器寄存器（ENET_MSC_RGUFCNT）

地址偏移：0x01C4
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器统计接收到”好”单播帧的数目。

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

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

位/位域	名称	描述
31:0	RGUF[31:0]	“好”单播帧接收帧计数器 这些位是接收到”好“的单播帧的计数器

27.4.33. PTP 时间戳控制寄存器（ENET_PTP_TSCTL）

地址偏移：0x0700

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。

该寄存器用于配置时间戳的产生和更新。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										TMSARU	TMSITEN	TMSSTU	TMSSTI	TMSFCU	TMSEN
										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	TMSARU	时间戳加数寄存器更新位 该位在更新完成后清'0'。该位在置位前必须确保读出为'0' 0：不将时间戳加数寄存器的值更新到 PTP 模块进行精调 1：将时间戳加数寄存器的值更新到 PTP 模块进行精调
4	TMSITEN	时间戳中断触发使能 0：禁止时间戳中断 1：使能时间戳中断，当系统时间超过期望时间寄存器的值时将会产生中断 **注意：**产生时间戳中断后，该位将会清'0'
3	TMSSTU	时间戳系统时间更新位 置位该位之前，必须确保 TMSSTU 位和 TMSSTI 位读出为'0' 0：系统时间保持不变 1：更新系统时间，在原有系统时间上加上或者减去时间戳高和低更新寄存器的值。完成更新后，硬件将会清除该位
2	TMSSTI	时间戳系统时间初始化位 置位该位之前，必须确保该位读出为'0' 0：系统时间保持不变 1：初始化系统时间，将原有系统时间替换为时间戳高和低更新寄存器的值。在初始化完成后，硬件将会清除该位
1	TMSFCU	时间戳粗调或者精调更新位 0：用粗调的方式更新系统时间戳 1：用精调的方式更新系统时间戳
0	TMSEN	时间戳使能位 0：禁止时间戳功能 1：使能接收和发送帧的时间戳功能 **注意：**每次设置该位为'1'后，都需要重新初始化系统时间

27.4.34. PTP 亚秒递增寄存器（ENET_PTP_SSINC）

地址偏移：0x0704
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器用于配置亚秒递增寄存器的8位递增值。在粗调模式下，每个HCLK时钟周期，系统时间就加一次该寄存器的值。在精调模式下，在累加器溢出时，系统时间才加一次该寄存器的值。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	STMSSI[7:0]	系统时间亚秒递增 在每次系统时间递增时，把这些位的值加到系统时间的亚秒值上

27.4.35. PTP 时间戳高寄存器（ENET_PTP_TSH）

地址偏移：0x0708
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	STMS[31:0]	系统时间秒位 这些位表示了当前系统时间的秒值

27.4.36. PTP 时间戳低寄存器（ENET_PTP_TSL）

地址偏移：0x070C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31	STS	系统时间符号位 0：时间值是正的 1：时间值是负的
30:0	STMSS[30:0]	系统时间亚秒位 这些位表示了当前系统时间的亚秒值，若需使系统时间精度达到 20ns，则亚秒精度为 0.46ns

27.4.37. PTP 时间戳高更新寄存器（ENET_PTP_TSUH）

地址偏移：0x0710
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。
使用该寄存器的值对当前系统时间替换、加或减。时间戳高和低更新寄存器可以用来初始化或更新MAC的当前系统时间。应当先写这2个寄存器，再置位时间戳控制寄存器的TMSSTI位或TMSSTU位。

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

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

位/位域	名称	描述
31:0	TMSUS[31:0]	时间戳秒更新位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减去的秒值

27.4.38. PTP 时间戳低更新寄存器（ENET_PTP_TSUL）

地址偏移：0x0714
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31	TMSUPNS	时间戳更新正或者负符号位 TMSSTI 位置'1'时，该位应当为'0' 0：在系统时间上加上时间戳更新值 1：从系统时间中减去时间戳更新值
30:0	TMSUSS[30:0]	时间戳更新亚秒位 这些位表示的值在初始化时用于替换系统时间，在更新时表示在系统时间上加上或减去的亚秒值

27.4.39. PTP 时间戳加数寄存器（ENET_PTP_TSADDEND）

地址偏移：0x0718
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器只用于系统时间更新方式为精调模式。该寄存器的值在每个时钟周期都会累加到32位累加器上，一旦该累加器溢出就更新系统时间。

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

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

位/位域	名称	描述
31:0	TMSA[31:0]	时间戳加数 这些位用于时钟同步时加到累加器上的值

27.4.40. PTP 期望时间高寄存器（ENET_PTP_ETH）

地址偏移：0x071C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	ETSH[31:0]	期望时间戳高位 这些位表示了期望时间的秒值

27.4.41. PTP 期望时间低寄存器（ENET_PTP_ETL）

地址偏移：0x0720

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

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

位/位域	名称	描述
31:0	ETSL[31:0]	期望时间戳低位 这些位表示了期望时间的纳秒值

27.4.42. DMA 总线控制寄存器（ENET_DMA_BCTL）

地址偏移：0x1000

复位值：0x0000 2101

该寄存器可以按字节（8位）、半字（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						AA	FPBL	UIP	RXDP[5:0]						FB
						rw	rw	rw	rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTPR[1:0]		PGBL[5:0]						保留	DPSL[4:0]					DAB	SWR
rw		rw							rw					rw	rs

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	AA	地址对齐 0：关闭传输地址对齐功能 1：使能传输地址对齐，如果 FB 位为'1'，AHB 接口对齐所有连续传输至起始地址的 LS 位（位 1 到位 0）。如果 FB 位为'0'，除第一次 AHB 访问的地址（访问数据缓存的起始地址）不对齐，后续的传输与地址均对齐
24	FPBL	4×PGBL 模式 0：PGBL 值（位[22:17]和位[13:8]）作为 DMA 传输长度值 1：PGBL 值（位[22:17]和位[13:8]）乘以 4 作为 DMA 传输长度值
23	UIP	使用分散 PGBL 0：PGBL 值（位[13:8]）对 DMA 接收和发送控制器都有效 1：RXDP[5:0]位用于 RxDMA 的传输长度值，PGBL[5:0]位用于 TxDMA 的传输长度值
22:17	RXDP[5:0]	RxDMA PGBL 位 如果 UIP=0，则这些位无效。仅当 UIP=1 时，这些位定义了一次 DMA 转发的最大数据传输次数。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留
16	FB	固定传输位 0：AHB 在连续传输时，只用 SINGLE 和 INCR 数据传输操作 1：AHB 在连续传输时，用 SINGLE，INCR4，INCR8 和 INCR16 数据传输操作 **注意：**MB 和 FB 位应当且必须只有其中一位为'1'
15:14	RTPR[1:0]	接收发送优先级比率 这些位表示 RxDMA 和 TxDMA 之间的访问优先级比率。 0x0：RxDMA：TxDMA = 1：1 0x1：RxDMA：TxDMA = 2：1 0x2：RxDMA：TxDMA = 3：1 0x3：RxDMA：TxDMA = 4：1 **注意：**该位只在 DMA 仲裁模式为循环模式（DAB=0）时有效

13:8	PGBL[5:0]	可编程的数据传输长度位 这些位定义了一次 DMA 转发的最大数据传输次数。如果 UIP=1，则这些位仅用于 TxDMA 传输。如果 UIP=0 时，则这些位同时用于 TxDMA 和 RxDMA 传输。 0x01：最大数据传输次数为 1 0x02：最大数据传输次数为 2 0x04：最大数据传输次数为 4 0x08：最大数据传输次数为 8 0x10：最大数据传输次数为 16 0x20：最大数据传输次数为 32 其他：保留
7	保留	必须保持复位值。
6:2	DPSL[4:0]	描述符跳跃长度 这些位仅对于环模式的两个描述符有效，定义了两个无链接的描述符之间从当前描述符的结尾到下一个描述符开头的地址差值，单位为字(32 位)。若 DPSL 域为 0 则 DMA 认为描述符是相邻地连续排列的
1	DAB	DMA 仲裁位 该位指示了 TxDMA 和 RxDMA 之间的仲裁模式 0：根据 RTPR 位的值以循环方式仲裁 1：固定模式，接收的优先级高于发送
0	SWR	软件复位 在所有时钟域的复位操作完成之后，该位将由硬件清零 **注意：**在写任何 MAC 的寄存器前，应当确保该位为'0' 0：MAC 内部寄存器正常工作 1：复位 MAC 所有内核寄存器

27.4.43. DMA 发送查询使能寄存器（ENET_DMA_TPEN）

地址偏移：0x1004
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器用于TxDMA对发送描述符列表的查询。TxDMA通常因为发送帧的数据下溢错误或者描述符被CPU占有(DAV=0)而进入暂停状态。可以对该寄存器写任意值使能发送查询。

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

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

位/位域	名称	描述
31:0	TPE[31:0]	发送查询使能位 对这些位写任意值，DMA使能发送查询，将查询当前描述符（描述符地址在ENET_DMA_CTDADDR寄存器中）是否被CPU占有。如果不是(DAV=1)，则描述符可用，DMA退出暂停状态并恢复工作。如果是(DAV=0)，则TxDMA回到暂停状态，并把ENET_DMA_STAT的位TBU置'1'。

27.4.44. DMA 接收查询使能寄存器（ENET_DMA_RPEN）

地址偏移：0x1008
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器用于RxDMA对接收描述符列表的查询。对该寄存器写任意值可以使能接收查询。

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

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

位/位域	名称	描述
31:0	RPE[31:0]	接收查询使能位 对这些位写任意值，DMA使能接收查询，将查询当前描述符（描述符地址在ENET_DMA_CRDADDR寄存器中）是否被CPU占有。如果不是(DAV=1)，则描述符可用，DMA退出暂停状态并恢复工作。如果是(DAV=0)，则TxDMA回到暂停状态，并把ENET_DMA_STAT的位RBU置'1'。

27.4.45. DMA 接收描述符列表地址寄存器（ENET_DMA_RDTADDR）

地址偏移：0x100C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
接收描述符列表寄存器指向接收描述符队列的开始。描述符队列位于物理内存，并且其地址必须字对齐。只有在接收停止的时候，才允许写该寄存器。在开启RxDMA接收流程之前，必须正确配置该寄存器。

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

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

位/位域	名称	描述
31:0	SRT[31:0]	接收队列基址 这些位包含了接收描述符队列第一个描述符的地址。SRT[1:0]的取值默认为'0'， 因此SRT[1:0]这两个最低位是只读的

27.4.46. DMA 发送描述符列表地址寄存器（ENET_DMA_TDTADDR）

地址偏移：0x1010
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器指向发送描述符队列的起始。描述符队列位于物理内存，并且其地址必须字对齐。只有在发送停止的时候，才允许写该寄存器。在开启TxDMA发送流程之前，必须正确配置该寄存器。

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

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

位/位域	名称	描述
31:0	STT[31:0]	发送队列基址 这些位包含了发送描述符列表第一个描述符的地址。STT[1:0]的取值默认为'0'， 因此STT[1:0]这两个最低位是只读的

27.4.47. DMA 状态寄存器（ENET_DMA_STAT）

地址偏移：0x1014
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。
该寄存器表示DMA的状态位。读ENET_DMA_STAT 寄存器并不能清除其中的标志位。对寄存器位[16:0]（除保留位）需要写'1'才能清除，而写'0'是无效的。通过设置ENET_DMA_INTEN寄存器里的相应位，可以屏蔽位[16:0]触发的中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		TST	WUM	MSC	保留	EB[2:0]			TP[2:0]			RP[2:0]			NI
		r	r	r		r			r			r			rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AI	ER	FBE	保留		ET	RWT	RPS	RBU	RS	TU	RO	TJT	TBU	TPS	TS
rc_w1	rc_w1	rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	TST	时间戳触发状态 该位指示发生了一个时间戳中断事件。通过清除TMST标志，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：未发生时间戳中断事件 1：发生了时间戳中断事件
28	WUM	WUM状态 该位指示发生了一个WUM事件。当两个事件触发源状态都被清除时，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：WUM模块未发生中断事件 1：WUM模块发生了中断事件
27	MSC	MSC状态 该位指示发生了一个MSC事件。当所有事件触发源状态都被清除时，可以清零该位。当该位置'1'时，如果相应中断屏蔽位复位，则产生中断。 0：MSC模块未发生中断事件 1：MSC模块发生了中断事件
26	保留	必须保持复位值。
25:23	EB[2:0]	错误位状态 当FBE=1时，这些位将对AHB总线上的总线响应错误进行错误类型解析 EB[0] 1：TxDMA传输数据时出错 0：RxDMA传输数据时出错 EB[1] 1：读数据转发时出错 0：写数据转发时出错 EB[2] 1：访问描述符时出错 0：访问数据缓存时出错
22:20	TP[2:0]	发送流程状态 这些位表示TxDMA的状态 0x0：停止，接到复位或者停止发送命令 0x1：运行，正在取发送描述符 0x2：运行，正在等待状态信息 0x3：运行，正在读取内存中数据并存入TxFIFO中 0x4，0x5：保留 0x6：暂停，发送描述符不可用或发送缓存数据下溢 0x7：运行，正在关闭发送描述符
19:17	RP[2:0]	接收流程状态 这些位表示RxDMA的状态 0x0：停止，接到复位或者停止接收命令 0x1：运行，正在取接收描述符 0x2：保留

		0x3：运行，正在等待接收数据包 0x4：暂停，接收描述符不可用 0x5：运行，正在关闭接收描述符 0x6：保留 0x7：运行，正在把接收到数据包从RxFIFO转发到内存中
16	NI	正常中断汇总 该位是下列位在相应中断使能位（ENET_DMA_INTEN寄存器）使能了的情况下，其各取值的逻辑或： TS (ENET_DMA_STAT [0])：发送中断 TBU (ENET_DMA_STAT [2])：发送缓存不可用 RS (ENET_DMA_STAT [6])：接收中断 ER (ENET_DMA_STAT [14])：提前接收中断 **注意：**该位置'1'后，只有把造成该位置'1'的位清'0'(写'1')，才能把该位清'0'
15	AI	异常中断汇总 该位是下列位在相应中断使能位（ENET_DMA_INTEN寄存器）使能了的情况下，其各取值的逻辑或： TPS (ENET_DMA_STAT [1])：发送流程停止 TJT (ENET_DMA_STAT [3])：发送Jabber超时 RO (ENET_DMA_STAT [4])：RxFIFO上溢 TU (ENET_DMA_STAT [5])：发送数据下溢 RBU (ENET_DMA_STAT [7])：接收缓存不可用 RPS (ENET_DMA_STAT [8])：接收流程停止 RWT (ENET_DMA_STAT [9])：接收看门狗超时 ET (ENET_DMA_STAT [10])：提前发送中断 FBE (ENET_DMA_STAT [13])：总线致命错误 **注意：**该位置'1'后，只有把造成该位置'1'的位清'0'(写'1')，才能把该位清'0'
14	ER	提前接收状态 在接收中断位(ENET_DMA_STAT[6])置'1'时，该位自动清'0' 0：未接收到帧数据 1：接收到的数据帧已由DMA填满了第一个缓存
13	FBE	总线致命错误状态 该位指示发生了一个AHB接口响应错误，其错误类型可以由EB[2:0]位进行解释 0：未发生总线错误 1：发生了总线错误，相应的DMA控制器停止所有操作
12:11	保留	必须保持复位值。
10	ET	提前发送状态 0：发送的帧还未完全传输到TxFIFO中 1：发送的帧已经完全传输到TxFIFO中
9	RWT	接收看门狗超时状态 0：接收到的帧长度小于2048字节

		1：接收到的帧长度超过2048字节
8	RPS	接收流程停止状态 0：接收流程未停止 1：接收流程进入停止状态
7	RBU	接收缓存不可用状态 0：下一个接收描述符的DAV位为'1' 1：下一个接收描述符的DAV位为'0'，RxDMA进入暂停状态
6	RS	接收状态 0：帧接收未完成 1：帧接收完成
5	TU	发送数据下溢状态 0：未发生发送数据下溢错误 1：发送帧的过程中发生数据下溢，同时发送进入暂停状态
4	RO	接收上溢状态 0：未发生接收数据上溢错误 1：接收帧的过程中发生上溢错误。如果已有一部分帧数据转发到内存，则设置上溢错误位(RDES0[11])为'1'
3	TJT	发送Jabber超时状态 0：未发生发送Jabber定时器超时事件 1：发送Jabber定时器超时。此时中止发送进程并进入停止状态，同时设置Jabber超时位JT(TDES0[14])为'1'
2	TBU	发送缓存不可用状态 0：下一个发送描述符的DAV位为'1' 1：下一个发送描述符的DAV位为'0'，TxDMA进入暂停状态
1	TPS	发送流程停止状态 0：发送未停止 1：发送停止
0	TS	发送状态 该位仅在TDES0中LSG和INTC位都置位时，才可被置位 0：当前帧发送未完成 1：当前帧发送完成

27.4.48. DMA 控制寄存器（ENET_DMA_CTL）

地址偏移：0x1018
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。
该寄存器设定了接收和发送的工作模式和命令。在整个DMA的初始化流程中，应当最后写该

寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					DTCERFD	RSFD	DAFRF	保留		TSFD	FTF	保留			TTHC[2]
					rw	rw	rw			rw	rs				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTHC[1:0]		STE	保留					FERF	FUF	保留	RTHC[1:0]		OSF	SRE	保留
rw		rw						rw	rw		rw		rw	rw	

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	DTCERFD	不丢弃TCP/IP校验和错误帧 0：如果FERF位为'0'，MAC丢弃所有有错的帧 1：如果接收到帧仅有校验和错误时，MAC不会丢弃该帧
25	RSFD	接收存储转发 0：RxFIFO工作在直通模式，转发阈值由RTHC位决定 1：节后FIFO工作在存储转发模式，只有在帧完整写入RxFIFO后，RxDMA才会把它转发给应用程序，此时RTHC位的取值会被忽略
24	DAFRF	不清空接收帧 0：当接收描述符不可用时，RxDMA就清空RxFIFO里的接收帧 1：RxDMA不清空接收帧，即使接收描述符不可用
23:22	保留	必须保持复位值。
21	TSFD	发送存储转发 0：TxFIFO工作在直通模式，发送阈值由ENET_DMA_CTL寄存器中的TTHC位决定 1：TxFIFO工作在存储转发模式，只有在帧完整写入TxFIFO后，MAC才会把其发送出去，TTHC位的取值会被忽略 **注意：**在发送处于停止状态时，可以修改该位
20	FTF	清空TxFIFO 当此位为1时，TxFIFO控制逻辑电路被复位到初始状态，TxFIFO里所有的数据被清空/丢失。在清空操作完成后该位被自动清'0' **注意：**在该位为'0'之前，不允许写ENET_DMA_CTL寄存器
19:17	保留	必须保持复位值。
16:14	TTHC[2:0]	发送阈值控制 这三位控制直通模式下TxFIFO的阈值。 当TSFD=1时，忽略这些位。 0x0：64 0x1：128 0x2：192 0x3：256 0x4：40

		0x5：32 0x6：24 0x7：16
13	STE	开始/停止发送 0：在发送完当前帧或TxDMA进入暂停状态后，发送进程进入停止模式。保存发送描述符队列里下一发送描述符的位置，在传输重新开始时，这个描述符就变成当前描述符 1：TxDMA进入运行状态。DMA获取当前发送描述符，发送帧描述符可从ENET_DMA_TDTADDR基址获取，若上一次发送为停止状态，则也可从发送描述符队列的指针位置获取。如果当前描述符的DAV位为'0'，则TxDMA进入暂停状态，并设置TBU位为'1'。如果在未设置完其他DMA寄存器的情况下就置位该位，则会引起不可预料的后果
12:8	保留	必须保持复位值。
7	FERF	转发错误帧 0：当RxFIFO工作于直通模式(RSFD=0)时，如果在将RxFIFO数据转发到内存之前检测到了帧错误（CRC错误、冲突错误、校验和错误、看门狗超时、溢出），则RxFIFO会丢弃这个错误的帧。但如果在将RxFIFO数据转发到内存之后才检测到了帧错误，则就不会丢弃该帧。当RxFIFO工作于存储转发模式时，在接收过程中一旦检测到帧错误，就会丢弃该帧 1：除了过短帧外的所有帧都会转发给DMA
6	FUF	转发长度不够的"好"帧 0：RxFIFO丢弃所有长度小于64字节的帧，但如果在检测到过短帧之前，帧已开始转发该帧给应用程序（例如在直通模式下，帧长小于接收阈值），则将转发整个帧 1：RxFIFO把长度不够的"好"帧（帧长小于64字节但没有错误）转发给应用程序
5	保留	必须保持复位值。
4:3	RTHC[1:0]	接收阈值控制 这两位设置了在直通模式下RxFIFO的阈值。 **注意：**只有在RSFD位(位21)为'0'时，这些位才有效。在RSFD位为'1'时忽略这些位。 0x0：64 0x1：32 0x2：96 0x3：128
2	OSF	操作第二帧 0：TxDMA仅在接收到前一个帧的发送状态信息后，才开始发送下一个帧的数据 1：TxDMA在前一帧数据全部存入到TxFIFO之后，在接收到前一个帧的发送状态信息前，就开始发送下一个帧的数据
1	SRE	开始/停止接收 0：在转发完当前接收帧后，RxDMA进入停止模式。保存接收描述符队列里下一接收描述符的位置，在传输重新开始时，这个描述符就变成当前描述符。只有在接收运行时或接收暂停时，可"停止接收"

		1：把接收进程置为运行状态，DMA检查接收描述符队列的当前位置，用来处理下一个收到的帧。接收帧描述符可从ENET_DMA_RDTADDR基址获取，若上一次接收为停止状态，则也可从接收描述符队列的指针位置获取。如果获取的描述符DAV=0，那么接收进程进入暂停状态，并设置RBU位为'1'。只有在接收停止时或接收暂停时，"开始接收"命令才有效。在未设置完所有其他DMA寄存器之前发出"开始接收"命令，会引起不可预料的后果
0	保留	必须保持复位值。

27.4.49. DMA 中断使能寄存器（ENET_DMA_INTEN）

地址偏移：0x101C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器可以使能ENET_DMA_STAT寄存器反映的中断。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	ERIE	FBEIE	保留		ETIE	RWTIE	RPSIE	RBUIE	RIEN	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE
rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	NIE	正常中断汇总使能 0：屏蔽正常中断 1：使能正常中断 该位使能下列位： TS (ENET_DMA_STAT [0])：发送中断 TBU (ENET_DMA_STAT [2])：发送缓存不可用 RS (ENET_DMA_STAT [6])：接收中断 ER (ENET_DMA_STAT [14])：提前接收中断
15	AIE	异常中断汇总使能 0：屏蔽异常中断 1：使能异常中断 该位使能下列位： TPS (ENET_DMA_STAT [1])：发送流程停止 TJT (ENET_DMA_STAT [3])：发送Jabber超时 RO (ENET_DMA_STAT [4])：RxFIFO上溢 TU (ENET_DMA_STAT [5])：发送下溢 RBU (ENET_DMA_STAT [7])：接收缓存不可用 RPS (ENET_DMA_STAT [8])：接收流程停止

		RWT (ENET_DMA_STAT [9])：接收看门狗超时 ET (ENET_DMA_STAT [10])：提前发送中断 FBE (ENET_DMA_STAT [13])：总线致命错误
14	ERIE	提前接收中断使能 0：屏蔽提前接收中断 1：使能早接收中断
13	FBEIE	总线致命错误中断使能 0：屏蔽总线致命错误中断 1：使能总线致命错误中断
12:11	保留	必须保持复位值。
10	ETIE	提前发送中断使能 0：屏蔽提前发送中断 1：使能提前发送中断
9	RWTIE	接收看门狗超时中断使能 0：屏蔽接收看门狗超时中断 1：使能接收看门狗超时中断
8	RPSIE	接收流程停止中断使能 0：屏蔽接收流程停止中断 1：使能接收流程停止中断
7	RBUIE	接收缓存不可用中断使能 0：屏蔽接收缓存不可用中断 1：使能接收缓存不可用中断
6	RIE	接收中断使能 0：屏蔽接收中断 1：使能接收中断
5	TUIE	发送下溢中断使能 0：屏蔽发送数据下溢中断 1：使能发送下溢中断
4	ROIE	接收上溢中断使能 0：屏蔽接收上溢中断 1：使能接收上溢中断
3	TJTIE	发送Jabber超时中断使能 0：屏蔽发送Jabber超时中断 1：使能发送Jabber超时中断
2	TBUIE	发送缓存不可用中断使能 0：屏蔽发送缓存不可用中断 1：使能发送缓存不可用中断

1	TPSIE	发送流程停止中断使能 0：屏蔽发送流程停止中断 1：使能发送流程停止中断
0	TIE	发送中断使能 0：屏蔽发送中断 1：使能发送中断

27.4.50. DMA 丢失帧和缓存溢出计数器寄存器（ENET_DMA_MFBOCNT）

地址偏移：0x1020
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
DMA有2个计数器，用来统计接收过程中丢失帧的数目。可通过读本寄存器来获取计数器的当前值。这个计数器通常用作故障诊断。

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

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	OBFOC	OBFOC：FIFO溢出计数器溢出位(Overflow bit for FIFO overflow counter)
27:17	MSFA[10:0]	应用程序丢失的帧 这些位指示了RxFIFO丢失的帧数目
16	保留	必须保持复位值。
15:0	MSFC[15:0]	控制器丢失的帧 这些位表示了由于MCU的接收缓存不可用而导致RxDMA丢失的帧的数目。每当DMA清空一个输入帧时，这个计数器加1

27.4.51. DMA 当前发送描述符地址寄存器（ENET_DMA_CTDADDR）

地址偏移：0x1048
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
当前发送描述符寄存器指向TxDMA正在读取的发送描述符起始地址。

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

TDAP[31:16]
r

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

位/位域	名称	描述
31:0	TDAP[31:0]	发送描述符地址指针 这些位在复位时清'0'，由TxDMA在操作过程中自动更新

27.4.52. DMA 当前接收描述符地址寄存器（ENET_DMA_CRDADDR）

地址偏移：0x104C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
当前接收描述符寄存器指向RxDMA正在读取的接收描述符起始地址。

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

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

位/位域	名称	描述
31:0	RDAP[31:0]	接收描述符地址指针 这些位在复位时清'0'，由RxDMA在操作过程中自动更新

27.4.53. DMA 当前发送缓存地址寄存器（ENET_DMA_CTBADDR）

地址偏移：0x1050
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器指向TxDMA正在读取的发送缓存的地址。

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

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

位/位域	名称	描述
31:0	TBAP[31:0]	发送缓存地址指针 这些位在复位时清'0'，由TxDMA在工作过程中更新

27.4.54. DMA 当前接收缓存地址寄存器（ENET_DMA_CRBADDR）

地址偏移：0x1054
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
该寄存器地址指向RxDMA正在读取的接收缓存地址。

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

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

位/位域	名称	描述
31:0	RBAP[31:0]	接收缓存地址指针 这些位在复位时清'0'，由RxDMA在工作过程中更新

28. 通用串行总线全速接口（USBFS）

USBFS适用于GD32F205xx和GD32F207xx系列芯片。

28.1. 简介

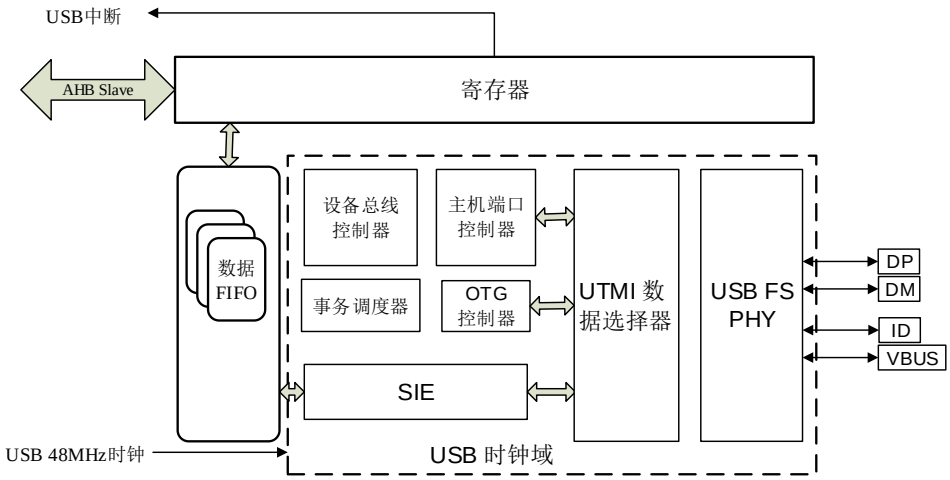
USB全速（USBFS）控制器为便携式设备提供了一套USB互联解决方案。USBFS不仅支持主机模式和设备模式，也支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG模式。USBFS包含了一个内部的全速USB PHY，并且不再需要外部PHY芯片。USBFS可以支持USB 2.0协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。

28.2. 主要特征

- 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；
- 支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG协议；
- 支持所有的4种传输方式：控制传输、批量传输、中断传输和同步传输；
- 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8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4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4个OUT端点和4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；
- 可检测ID引脚电平和VBUS电压；
- 在主机模式或者OTG A设备模式下，需要外部部件为连接的USB设备提供电源。

28.3. 结构框图

图 28-1. USBFS 结构框图



28.4. 信号线描述

表 28-1. USBFS 信号线描述

I/O 端口	类型	描述
VBUS	输入	总线电源端口
DM	输入/输出	差分信号 D-端口
DP	输入/输出	差分信号 D+端口
ID	输入	USB 识别：微连接器识别接口

28.5. 功能说明

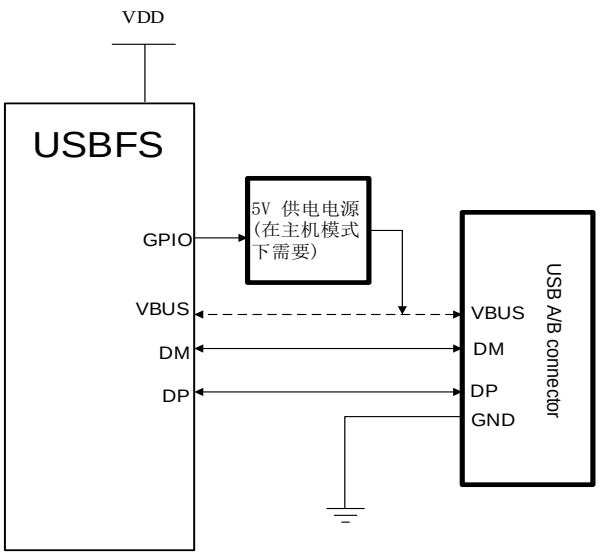
28.5.1. USBFS 时钟及工作模式

USBFS可以作为一个主机、一个设备或者一个DRD（双角色设备），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、全速的设备模式以及具备HNP和SRP的OTG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、设备或OTG模式）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28-2. 在主机或设备模式下连接示意图*所示。

图 28-2. 在主机或设备模式下连接示意图

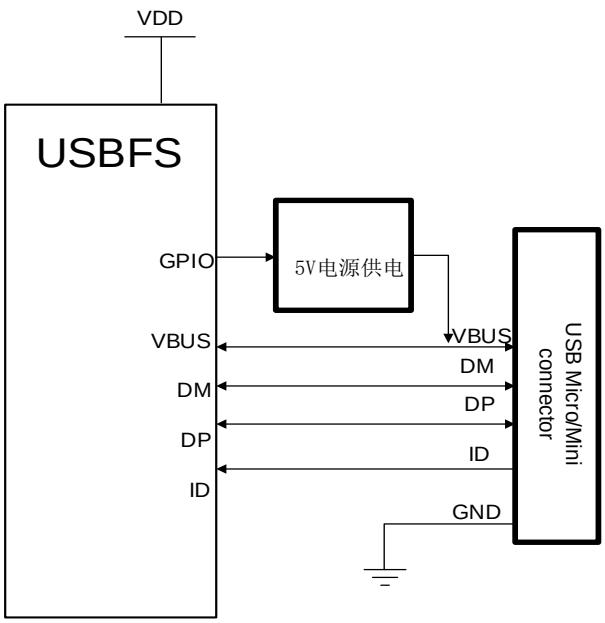


当USBFS工作在主机模式下时（FHM控制位置位、FDM控制位清除），VBUS为USB协议所定义的5V电源检测引脚。内部PHY不能提供5V VBUS电源，仅在VBUS信号线上具有电压比较器和充电放电电路。所以，如果应用需要提供VBUS电源，那么则需要一个外部的供电电源IC。在主机模式下，USBFS和USB连接头之间的VUBS连接可以被忽略，这是由于USBFS并不检测VBUS引脚的电平状态，并假定5V供电电源一直存在。

当USBFS工作在设备模式下时（FHM控制位清除、FDM控制位置位），VBUS检测电路连接到GPIO引脚。通过该GPIO引脚，USBFS持续监控VBUS电压。一旦VBUS电压升至所需有效值以上，需要立即开启DP信号线上的上拉电阻，这样则会产生一个连接状态。反之，当VBUS电压降至所需有效值以下，需要立即关闭DP信号线上的上拉电阻，这样则会产生一个断开状态。

OTG 模式连接示意图如*图 28-3. OTG 模式下连接示意图*所示。当 USBFS 工作在 OTG 模式下时，USBFS_GUSBCS 寄存器内的 FHM、FDM 控制位都应该被清除。在这种模式下，USBFS 需要以下四个引脚：DM、DP、VBUS 和 ID，并且需要使用若干个电压比较器检测这些引脚的电压。 USBFS 也包含 VBUS 充电和放电电路，用以完成 OTG 协议中所描述的 SRP 请求。OTG A 设备或 B 设备由 ID 引脚的电平状态所决定。在实现 HNP 协议的过程中，USBFS 控制上拉和下拉电阻。

图 28-3. OTG 模式下连接示意图

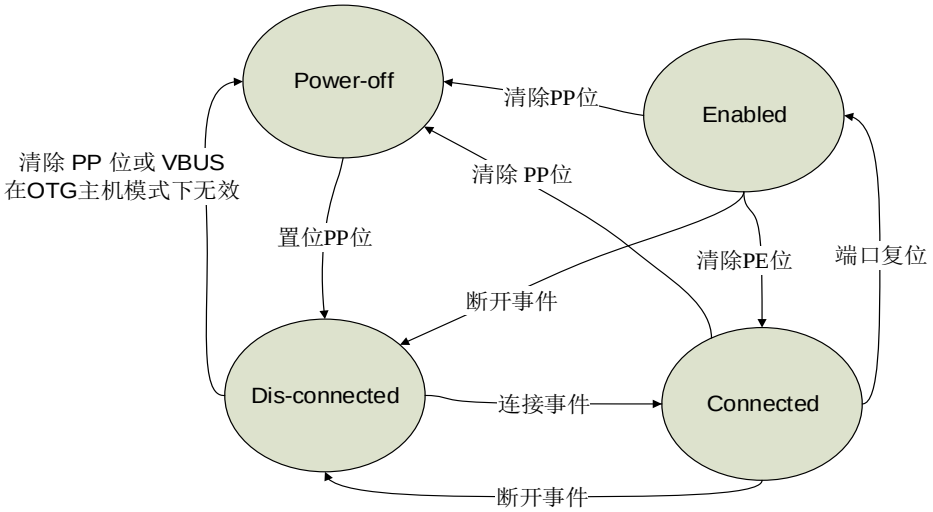


28.5.2. USB 主机功能

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 28-4. 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8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。

28.5.3. 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。

注意：VBUS引脚必须接到PA9来检测电平。

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3ms无变化，USB设备将会进入挂起状态。当USB设备在挂起状态时，软件能够关闭大部分的时钟以节省电能。USB主机可以通过在USB总线上产生恢复信号，来唤醒挂起的设备。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。

28.5.4. OTG 功能概述

USBFS支持OTG协议1.3中所描述的OTG功能，OTG功能包括SRP和HNP。

A设备和B设备

当标准A或微型A插头插入相应的插座时，具有OTG能力的USB设备为A设备。A设备向VBUS供电，并且在会话开始时默认为主机。当标准B、微型B、迷你B插头插入相应的插座或采用一端为标准A插头的不可分离电缆时，具有OTG能力的USB设备为B设备。B设备在会话开始时默认为外设。USBFS使用ID引脚电平状态决定A设备或B设备。ID引脚状态反馈在USBFS_GOTGCS寄存器的IDPS状态位。为了了解A设备和B设备之间传输的详细状态，请参考OTG1.3协议。

HNP

主机协商协议（HNP）允许主机功能在两个直接连接的OTG设备之间转换，并且用户不需要为了设备之间通信控制的改变而切换电缆线的连接。典型地，HNP协议是由B设备上的用户或应用启动，HNP只能通过设备上的微型AB插座执行。

一旦OTG设备具有一个微型AB插座，该OTG设备可通过插入的插头类型决定默认为主机或设备（微型A插头插入为主机，微型B插头插入为设备）。通过使用主机协商协议（HNP），一个默认为外设的OTG设备可以请求成为主机。主机角色切换的过程在下段中描述。此协议使用户不需要为了更改连接设备的角色而切换电缆线的连接。

当USBFS工作在OTG A主机模式时，并且其想放弃主机角色，可以首先置位USBFS_HPCS寄存器的PSP控制位来使USB总线进入挂起状态，然后B设备在3ms后进入挂起状态。如果B设备想要变为主机，软件需要置位USBFS_GOTGCS寄存器的HNPREQ控制位，然后USBFS会开始在总线上执行HNP协议，最后，HNP的结果会反馈在USBFS_GOTGCS寄存器的HNPS状态位。另外，软件总能从USBFS_GINTF寄存器的COPM状态位获取当前设备角色（主机或外设）。

SRP

会话请求协议（SRP）允许B设备请求A设备打开VBUS并启动一个会话。该协议允许A设备（或许是电池供电）当总线无活动时通过关闭VBUS以节省电能，并为B设备启动总线活动提供了一种方法。如OTG协议中所描述，OTG设备必须和几个阈值比较VBUS电压，并且将比较结果反馈在USBFS_GOTGCS寄存器的ASV和BSV状态位中。

当USBFS工作在B设备OTG模式时，软件可以通过置位USBFS_GOTGCS寄存器的SRPREQ控制位来启动一个SRP请求，并且如果SRP请求成功，USBFS会在USBFS_GOTGCS寄存器中产生一个成功标志位SRPS。

当USBFS工作在OTG A设备模式且从B设备检测到一个SRP请求时，USBFS将会置位USBFS_GINTF寄存器中的SESIF标志位。软件获取该标志位后，需要准备为VBUS引脚打开5V供电电源。

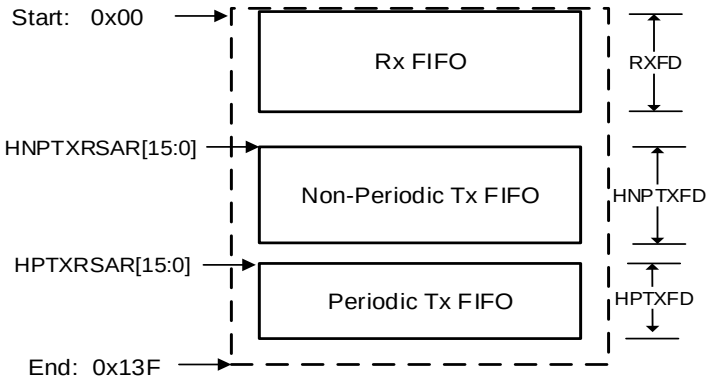
28.5.5. 数据 FIFO

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

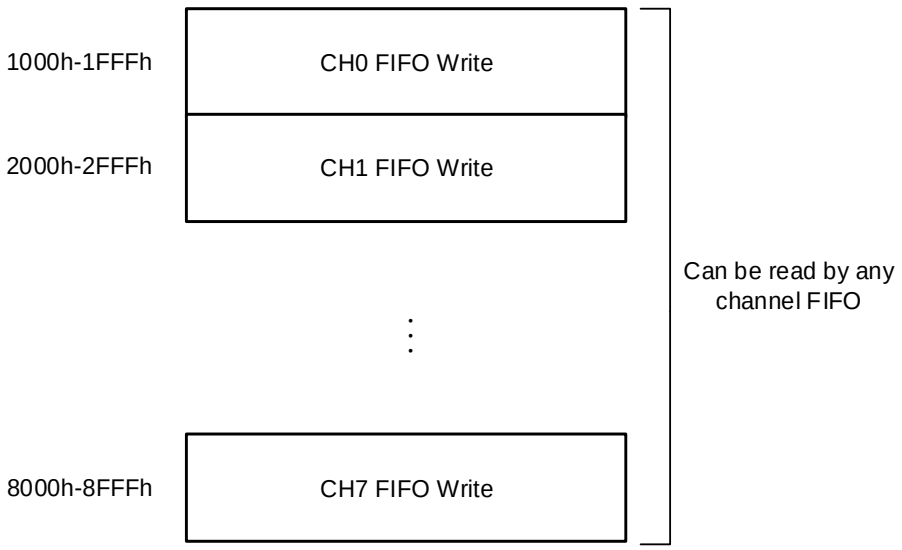
主机模式下，数据 FIFO 空间分为三个部分，分别是：用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据，所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBFS_GRFLEN、USBFS_HNPTFLEN 和 USBFS_HPTFLEN，软件可以配置以上数据 FIFO 的大小和起始偏移地址。*图 28-5. 主机模式 FIFO 空间*所描述的是 SRAM 中各 FIFO 的结构，图中的数值是按照 32 位为单位写的。

图 28-5. 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 28-6. 主机模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 可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。

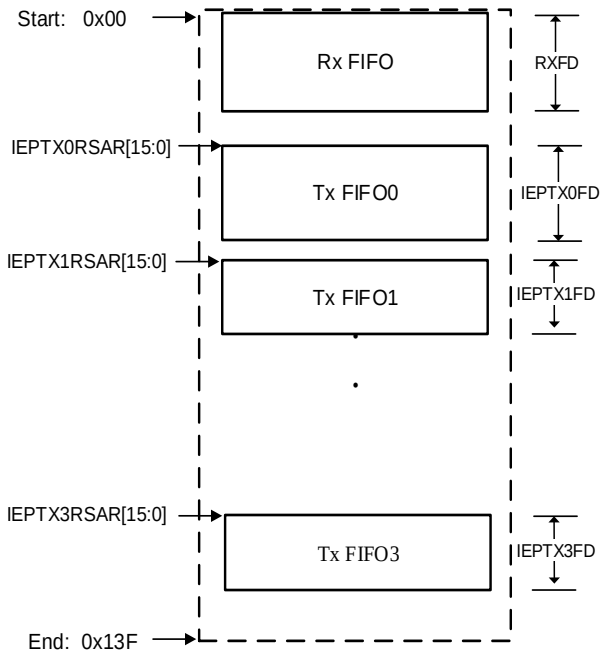
图 28-6. 主机模式 FIFO 访问寄存器映射表



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 4 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…3），程序可配置数据 FIFO 的大小和起始偏移地址。*图 28-7. 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。

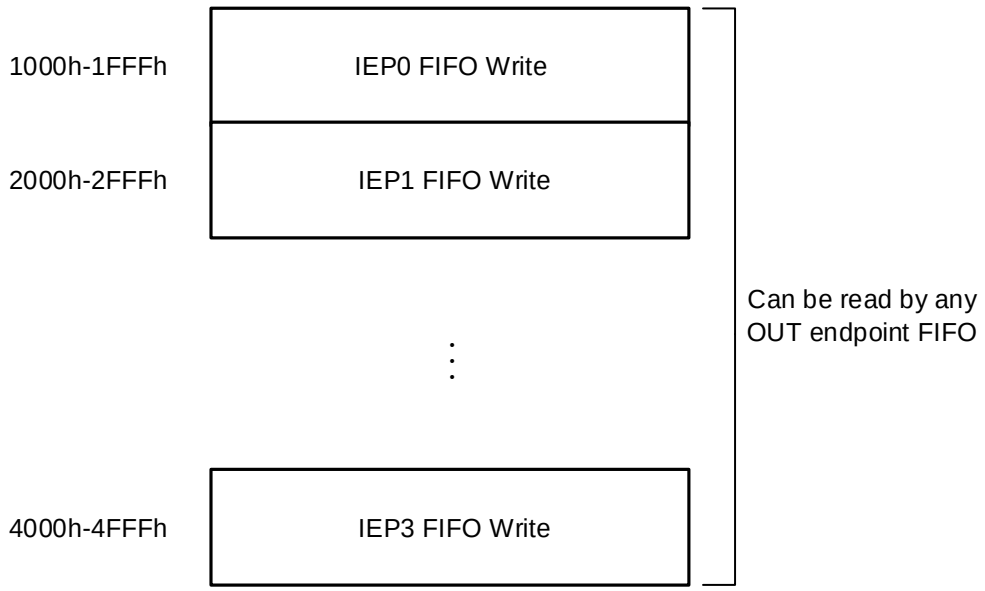
图 28-7. 设备模式 FIFO 空间



USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。*图 28-8. 设备模式 FIFO 访问寄存器映射表*所描述的是数据 FIFO 可写的寄存器存储空间，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 28-8. 设备模式 FIFO 访问寄存器映射表



28.5.6. 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、设备或OTG）、某些OTG参数和USB协议，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；
4、 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为1；
5、 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；
6、 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：

1、 初始化 USBFS 全局寄存器；
2、 初始化及使能相应通道；
3、 将数据包写入通道的 Tx FIFO（周期性 Tx FIFO 或非周期性 Tx FIFO）。在所有的数据包都被写入 FIFO 后，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；

4、 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；
5、 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；
6、 如果步骤 5 所描述的 OUT 事务完成后且步骤 2 的 PCNT 的数值比 1 大，程序将会返回步骤 3，继续发送剩下的数据包。如果步骤 5 中描述的 OUT 事务没有成功完成，程序将会返回步骤 3 来再次发送该包；
7、 在所有的传输中的所有事务都被成功送达后，USBFS 生成 TF 标志来指示传输成功结束；
8、 除能通道，当通道处于空闲状态，即可为其他传输做准备。

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、设备或 OTG）、某些 OTG 参数、USB 协议，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_DIEPnCTL 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；
4、 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 为 1；
5、 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；
6、 置位 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL 寄存器 EPEN 位使能端点。

端点除能顺序

当USBFS_DIEPnCTL或USBFS_DOEPnCTL寄存器的EPEN位被清除时，程序可以在任何时候除能端点

IN传输操作顺序：

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。

28.6. 中断

OTG 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 28-2. USBFS 全局中断*中。

表 28-2. USBFS 全局中断

中断标志	描述	运行模式
SESIF	会话中断	主机或设备模式
DISCIF	断开连接中断标志	主机模式
IDPSC	ID 引脚状态变化	主机或设备模式
PTXFEIF	周期性 Tx FIFO 空中断标志	主机模式
HCIF	主机通道中断标志	主机模式
HPIF	主机端口中断	主机模式
ISOONCIF/PXNCIF	周期性传输未完成中断标志 / 同步OUT传输未完成中断标志	主机或设备模式
ISOINCIF	同步 IN 传输未完成中断标志	设备模式

中断标志	描述	运行模式
OEPIF	OUT 端点中断标志	设备模式
IEPIF	IN 端点中断标志	设备模式
EOPFIF	周期性帧尾中断标志	设备模式
ISOOPDIF	同步 OUT 丢包中断标志	设备模式
ENUMF	枚举完成	设备模式
RST	USB 复位	设备模式
SP	USB挂起	设备模式
ESP	早挂起	设备模式
GONAK	全局OUT NAK有效	设备模式
GNPINAK	全局非周期IN NAK有效	设备模式
NPTXFEIF	非周期Tx FIFO空中断标志	主机模式
RXFNEIF	Rx FIFO非空中断标志	主机或设备模式
SOF	帧首	主机或设备模式
OTGIF	OTG 中断标志	主机或设备模式
MFIF	模式错误中断标志	主机或设备模式

唤醒中断可以在 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。

28.7. USBFS 寄存器

USBFS基地址： 0x5000 0000

28.7.1. 全局控制与状态寄存器组

全局 OTG 控制和状态寄存器 （USBFS_GOTGCS）

地址偏移：0x0000
复位值：0x0000 0800

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												BSV	ASV	DI	CIDPS
												r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DHNPEN	HHNPEN	HNPREQ	HNPS	保留						SRPREQ	SRPS
				rw	rw	rw	r							rw	r

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	BSV	B会话有效（在OTG协议中描述） 0：OTG B设备VBUS电压水平低于VBSESSVLD 1：OTG B设备VBUS电压水平不低于VBSESSVLD **注意：**仅在OTB B设备模式下可访问
18	ASV	A会话有效 A主机模式收发器状态 0：OTG A设备VBUS电压水平低于VASESSVLD 1：OTG A设备VBUS电压水平不低于VASESSVLD 在会话的开始，A设备默认是主机。 **注意：**仅在OTG A设备模式下可访问
17	DI	去抖动间隔 检测到连接的去抖动间隔。 0：当USB总线上发生插入和连接时，表示长去抖动间隔 1：当HNP协议中使用一个软连接时，指示短去抖动间隔 **注意：**仅在主机模式下可访问

16	CIDPS	ID引脚状态 连接器ID引脚的电压水平 0：USBFS工作在A设备模式 1：USBFS工作在B设备模式 **注意：**在设备和主机模式下均可访问
15:12	保留	必须保持复位值。
11	DHNPEN	设备HNP使能 使能B设备HNP功能。如果该控制位清除，当应用置位USBFS_GOTGCS寄存器中的HNPREQ控制位c时，USBFS并不启动HNP协议。 0：HNP功能不使能 1：HNP功能使能 **注意：**仅在设备模式下访问
10	HHNPEN	主机HNP使能 使能A设备HNP功能。如果该控制位清除，USBFS不能够响应B设备的HNP请求。 0：HNP功能不使能 1：HNP功能使能 **注意：**仅在主机模式下访问
9	HNPREQ	HNP请求 软件通过置位该控制位在USB总线上启动一个HNP。当USBFS_GOTGINTF寄存器中HNPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的HNPEND控制位来清除该控制位。 0：不发送HNP请求 1：发送HNP请求 **注意：**仅在设备模式下访问
8	HNPS	HNP成功标志位 当HNP成功时，该标志位由内核置位。当HNPREQ置位时，该控制位被清除。 0：HNP失败 1：HNP成功 **注意：**仅在设备模式下访问
7:2	保留	必须保持复位值。
1	SRPREQ	SRP请求 软件通过置位该控制位在USB总线上启动一个SRP会话请求。当USBFS_GOTGINTF寄存器中的SRPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的SRPEND控制位来清除该控制位。 0：没有会话请求 1：会话请求 **注意：**仅在设备模式下访问
0	SRPS	SRP会话请求成功 当SRP会话请求成功时，该标志位由内核置位。当SRPREQ控制位被置位时，该标志位被清除。

0：SRP会话请求失败

1：SRP会话请求成功

注意：仅在设备模式下访问

全局 OTG 中断状态寄存器 （USBFS_GOTGINTF）

地址偏移：0x0004

复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												DF	ADTO	HNPDET	保留
												rc_w1	rc_w1	rc_w1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						HNPEND	SRPEND	保留					SESEND	保留	
						rc_w1	rc_w1						rc_w1		

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	DF	去抖动完成 当设备连接去抖动完成时，USBFS置位该控制位 **注意：**仅在主机模式下可访问
18	ADTO	A设备超时 当A设备等待B设备连接发生超时，USBFS置位该控制位 **注意：**在设备和主机模式下，均可访问
17	HNPDET	检测到主机协商请求 当A设备检测到一个HNP请求时，USBFS置位该标志位 **注意：**在设备和主机模式下，均可访问
16:10	保留	必须保持复位值。
9	HNPEND	HNP结束 当一个HNP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中HNPS标志位，以获取HNP结果。 **注意：**在设备和主机模式下，均可访问。
8	SRPEND	SRP结束 当一个SRP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。

		注意：在设备和主机模式下，均可访问。
7:3	保留	必须保持复位值。
2	SESEND	会话结束 当VBUS电压低于Vb_ses_vld时，内核置位该标志位。
1:0	保留	必须保留复位值。

全局 AHB 控制和状态寄存器 （USBFS_GAHBCS）

地址偏移：0x0008
复位值：0x0000 0000

该寄存器只能按字（32位）访问

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

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

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 0A80

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		UTT[3:0]				HNPCEN	SRPCEN	FSSTS	保留					TOC[2:0]	
				rw		r/rw	r/rw	r						rw	

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，让修改产生作用。 **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，让修改产生作用。 **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	HNPCEN	HNP能力使能 控制HNP能力是否使能 0：HNP能力禁用 1：HNP能力使能 **注意：**在设备和主机模式下，均可访问
8	SRPCEN	SRP能力使能 控制SRP能力是否使能 0：SRP能力禁用 1：SRP能力使能 **注意：**在设备和主机模式下，均可访问
7	FSSTS	全速USB标示位，为只读位
6:3	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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					TXFNUM[4:0]					TXFF	RXFF	保留	HFCRST	HCSRST	CSRST
							rw			rs	rs		rs	rs	rs

位/位域	名称	描述
31	AHBMIDL	AHB主机空闲状态，总为1
30:11	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被刷新 00001：仅周期性Tx FIFO被刷新 1xxxx：周期性和非周期性Tx FIFO均被刷新 其他：没有数据被刷新 **设备模式：** 00000：仅Tx FIFO0被刷新 00001：仅Tx FIFO1被刷新 … 00011：仅Tx FIFO3被刷新 1XXXX：所有的Tx FIFO均被刷新 其他：没有数据被刷新
5	TXFF	Tx FIFO刷新控制位 应用通过置位该控制位来刷新Tx FIFO数据，并且TXFNUM[4:0]决定刷新的FIFO数目。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序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刷新控制位 应用通过置位该控制位来刷新Rx FIFO数据。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序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，应用程序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AHB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，应用程序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：0x0400 0021

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	SESIF	DISCIF	IDPSC	保留	PTXFEIF	HCIF	HPIF	保留		PXNCIF/ ISOONCIF	ISOINCIF	OEPIF	IEPIF	保留	
rc_w1	rc_w1	rc_w1	rc_w1		r	r	r			rc_w1	rc_w1	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIF	ISOOPDIF	ENUMF	RST	SP	ESP	保留		GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

位/位域	名称	描述
31	WKUPIF	唤醒中断标志位 当在USB总线上检测到一个恢复信号（在设备模式下）或者一个远程唤醒信号（在主机模式下），硬件将置位该中断标志位。 **注意：**在设备和主机模式下，均可访问
30	SESIF	会话中断标志位 当在A设备模式下检测到一个SRP会话请求或在B设备模式下B设备的VBUS变为可用时，硬件将置位该中断标志位 **注意：**在设备和主机模式下，均可访问
29	DISCIF	断开中断标志位 当设备断开后，将触发该标志位。 **注意：**仅在主机模式下访问
28	IDPSC	ID引脚状态改变中断标志位 当ID引脚状态改变时，内核将置位该标志位 **注意：**在设备和主机模式下，均可访问
27	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_HPCSR寄存器以获取该中断源。当产生端口中断的标志被清除后，该中断标志位将自动清除。 **注意：**仅在主机模式下访问
23:22	保留	必须保持复位值。
21	PXNCIF	周期性传输未完成中断标志位 在当前帧内，当帧结束时，周期性传输未完成，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	OTGIF	OTG中断标志位 当USBFS_GOTGINTF寄存器中标志位产生一个中断时，USBFS置位该中断标志位。软件应该读取USBFS_GOTGINTF寄存器以获取产生该中断的信号源，当USBFS_GOTGINTF寄存器中产生该中断的标志位被清除后，该中断标志位也被自动清除。 **注意：**在设备和主机模式下，均可访问

1	MFIF	模式错误中断标志位 如果软件在设备模式下操作仅主机可访问的寄存器或者在主机模式下操作仅设备可访问的寄存器，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	SESIE	DISCIE	IDPSCIE	保留	PTXFEIE	HCIE	HPIE	保留		PXNCIE/ISOONCIE	ISOINCIE	OEPIE	IEPIE	保留	
rw	rw	rw	rw		rw	rw	r			rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIE	ISOOPDIE	ENUMFIE	RSTIE	SPIE	ESPIE	保留		GONAKIE	GNPINAKIE	NPTXFEIE	RXFNEIE	SOFIE	OTGIE	MFIE	保留
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31	WKUPIE	唤醒中断使能 0：禁用唤醒中断 1：使能唤醒中断 **注意：**在主机和设备模式下，均可访问
30	SESIE	会话中断使能 0：禁用会话中断 1：使能会话中断 **注意：**在主机和设备模式下，均可访问
29	DISCIE	断开中断使能 0：禁用断开中断 1：使能断开中断

		注意：仅在设备模式下使用
28	IDPSCIE	ID引脚状态改变中断使能 0：禁用连接器ID引脚状态中断 1：使能连接器ID引脚状态中断 **注意：**在主机和设备模式下，均可访问
27	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	OTGIE	OTG中断使能 0：禁止OTG中断 1：使能OTG中断 **注意：**在设备模式下与主机模式下，均可访问
1	MFIE	模式错误中断使能 0：禁止模式错误中断 1：使能模式错误中断 **注意：**在设备模式下与主机模式下，均可访问
0	保留	必须保持复位值。

全局接收状态读取/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：

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

DPID	BCOUNT[10:0]	CNUM[3:0]
r	r	r

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：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											RPCKST[3:0]				DPID
												r			r

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

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 0200

该寄存器只能按字（32位）访问

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

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

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：0x0200 0200

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							IEP0TXFD[15:0]	HNPTXFD/							
							r/rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							[15:0]	IEP0TXRSAR	HNPTXRSAR/						
							r/rw								

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的起始地址以32位字计数

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的起始地址以32位字计数

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 0200

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	NPTXRQTOP[6:0]							NPTXRQS[7:0]							
	r							r							

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

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：请求队列满 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为满 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：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										VBUSIG	SOFOEN	VBUSBCEN	VBUSACEN	保留	PWRON
										rw	rw	rw	rw		rw

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

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	VBUSIG	V_{BUS}忽略 当该控制位被置位，USBFS并不监测V_{BUS}引脚电压，并且认为在主机和设备模式下，V_{BUS}电压一直有效，然后可释放V_{BUS}引脚作为其他用途。 0：V_{BUS}不被忽略 1：V_{BUS}被忽略，并认为V_{BUS}电压一直有效
20	SOFOEN	SOF输出使能 0：SOF脉冲输出禁止 1：SOF脉冲输出使能
19	VBUSBCEN	V_{BUS} B设备比较器使能 0：V_{BUS} B设备比较器禁止 1：V_{BUS} B设备比较器使能
18	VBUSACEN	V_{BUS} A设备比较器使能 0：V_{BUS} A设备比较器禁止 1：V_{BUS} A设备比较器使能
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

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

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

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：0x0200 0600

该寄存器只能按字（32位）访问

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

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

r/rw

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 FIFO RAM起始地址 主机周期性发送FIFO RAM起始地址以32位字计数

设备 IN 端点发送 FIFO 长度寄存器 （USBFS_DIEPxTFLEN） （x = 1..3，其中 x 为 FIFO 编号）

地址偏移：0x0104 + (FIFO_number – 1) × 0x04
复位值：0x0200 0400

该寄存器只能按字（32位）访问

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

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

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Tx FIFO RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

28.7.2. 主机控制和状态寄存器

主机控制寄存器 （USBFS_HCTL）

地址偏移：0x0400
复位值：0x0000 0000

在主机模式下，上电后，该寄存器配置USB内核。主机初始化后，无需修改。

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														CLKSEL[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CLKSEL[1:0]	USB时钟选择 01：48MHz时钟 其他：保留

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。

该寄存器只能按字（32位）访问

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

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

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

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

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

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 0200

该寄存器反映了主机周期性Tx FIFO和请求队列的当前状态。请求队列包括在主机模式下的IN、OUT或其他请求条目。

该寄存器只能按字（32位）访问

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

PTXREQT[7:0]	PTXREQS[7:0]
r	r

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

位/位域	名称	描述
31:24	PTXREQT[7:0]	周期性发送 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

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	HACHINT[7:0]	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7表示通道7

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

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值

7:0	CINTEN[7:0]	通道中断使能 0：禁用通道n中断 1：使能通道n中断 每一位表示一个通道：位0代表通道0，位7代表通道7

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			PP	PLST[1:0]		保留	PRST	PSP	PREM	保留		PEDC	PE	PCD	PCST
			rw	r			rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

位/位域	名称	描述
31:19	保留	必须保持复位值。
18:17	PS[1:0]	端口速度 反映连接到该端口的设备的枚举速度。 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V_{BUS}引脚上具有电源供应。 0：端口掉电 1：端口供电
11:10	PLST[1:0]	端口线状态 反映USB数据线当前状态

		位10：DP线状态 位11：DM线状态
9	保留	必须保持复位值。
8	PRST	端口复位 应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0：端口不在复位状态 1：端口处于复位状态
7	PSP	端口挂起 应用设置该控制位来将端口进入挂起状态。当该控制位被置位后，端口停止发送SOF令牌包。该控制位只能够通过以下操作清除。 应用置位该寄存器中的PRST控制位 置位该寄存器中的PREM控制位 检测到一个远程唤醒信号 检测到一个设备断开 0：端口不在挂起状态 1：端口处于挂起状态
6	PREM	端口恢复 应用通过置位该控制位以在USB端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。 0：无恢复驱动 1：恢复驱动
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…7，其中 x 为通道号）

地址偏移：0x0500 +（通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CEN	CDIS	ODDFRM	DAR[6:0]							保留		EPTYPE[1:0]		LSD	保留
rs	rs	rw	rw									rw		rw	

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

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。 0：偶数帧 1：奇数帧
28:22	DAR[6:0]	设备地址 与该通道通信的USB设备地址。
21:20	保留	必须保持复位值。
19:18	EPTYPE[1:0]	端点类型 与该通道通信的端点的传输类型 00：控制 01：同步 10：批量

		11：中断
17	LSD	低速设备 与该通道通信的设备是一个低速设备。
16	保留	必须保持复位值。
15	EPDIR	端点方向 与该通道通信的端点的传输方向 0：OUT 1：IN
14:11	EPNUM[3:0]	端点号 与该通道通信的端点号
10:0	MPL[10:0]	最大包长 目标端点的最大包长

主机通道 x 中断标志寄存器 （USBFS_HCHxINTF）（x = 0…7，其中 x = 通道号）

地址偏移：0x0508 +（通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含一个通道的状态和事件，当软件获取一个通道中断时，软件需要为相应通道读取该寄存器以获取产生中断的中断源。该寄存器中的标志位均由硬件置位，并且写1清除。

该寄存器只能按字（32位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					DTER	REQOVR	BBER	USBER	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Rx FIFO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…7，其中 x = 通道号）

地址偏移：0x050C +（通道号 × 0x20）
复位值：0x0000 0000

该寄存器包含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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					DTERIE	REQOVRIE	BBERIE	USBERIE	保留	ACKIE	NAKIE	STALLIE	保留	CHIE	TFIE
					rw	rw	rw	rw		rw	rw	rw		rw	rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DTERIE	数据切换错误中断使能 0：禁用数据切换错误中断 1：使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0：禁用请求队列上溢中断 1：使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0：禁用串扰错误中断 1：使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0：禁用USB总线错误中断 1：使能USB总线错误中断
6	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…7，其中 x = 通道号）

地址偏移：0x0510 +（通道号 × 0x20）
复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DPID[1:0]		PCNT[9:0]										TLEN[18:16]		
	rw		rw										rw		

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

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。当软件正确向通道的数据Tx FIFO中写入一个包时，该位域以包中字节大小进行递减。

对于IN传输，每次软件从RxFIFO中读取一个包后，该位域也以包中字节大小进行递减。

28.7.3. 设备控制和状态寄存器

设备配置寄存器 （USBFS_DCFG）

地址偏移：0x0800

复位值：0x0000 0000

在上电、枚举或执行某些控制命令后，该寄存器配置内核为设备模式。在设备初始化后，不可以改变该寄存器值。

该寄存器采用字（32位）访问

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

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

位/位域	名称	描述
31:13	保留	必须保持复位值。
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	必须保持复位值。
2	NZLSOH	非零长度OUT状态阶段握手 在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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				POIF	CGONAK	SGONAK	CGINAK	SGINAK	保留			GONS	GINS	SD	RWKUP
				rw	w	w	w	w				r	r	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	POIF	上电初始化完成 软件通过设置该位，通知USBFS寄存器从掉电模式下唤醒后已完成初始化。
10	CGONAK	清零全局OUT NAK 软件设置该位从而清零该寄存器的GONS位
9	SGONAK	设置全局OUT NAK 软件设置该位从而实现该寄存器的位GONS置位。 当GONS位为零，设置该位一段时间会引起USBFS_GINTF寄存器的GONAK标志触发，软件应该在再次写该位前清除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，软件应该在再次写该位前清除GINAK标志。
6:4	保留	必须保持复位值。
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NAK、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NAK、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

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

FNRSOF[7:0]	保留	ES[1:0]	SPST
r	r	r	r

位/位域	名称	描述
31:22	保留	必须保持复位值。
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	必须保持复位值。
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：0x0810
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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									IEPNEEN	保留	EPTXFUDEN	CITOEN	保留	EPDISEN	TFEN
									rw		rw	rw		rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
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

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

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

保留	BTBSTPEN	保留	EPRXFOVREN	STPFEN	保留	EPDISEN	TFEN
	rw		rw	rw		rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
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	必须保持复位值。
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 1：使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
2	保留	必须保持复位值。
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获取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[3:0]			
													r		

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

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	OEPITB[3:0]	设备OUT端点中断位 每个位代表一个OUT端点：位16代表OUT端点0，位19代表OUT端点3
15:4	保留	必须保持复位值。
3:0	IEPITB[3:0]	设备IN端点中断位 每个位代表一个IN端点：位0代表IN端点0，位3代表IN端点3

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

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

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

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	OEPIE[3:0]	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：位16对应OUT端点0，位19对应OUT端点3
15:4	保留	必须保持复位值。
3:0	IEPIE[3:0]	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：位0对应IN端点0，位3对应IN端点3

设备 VBUS 放电时间寄存器 （USBFS_DVBUSDT）

地址偏移：0x0828
复位值：0x0000 17D7

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															

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DVBUSDT[15:0]	设备V_{BUS}放电时间 在SRP协议中，在V_{BUS}脉冲产生后，有一个放电过程，该域定义了V_{BUS}的放电时间，真正的放电时间是1024*DVBUSDT[15:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间。

设备 VBUS 脉冲时间寄存器 （USBFS_DVBUSPT）

地址偏移：0x082C
复位值：0x0000 05B8

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															

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

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DVBUSPT[11:0]	设备V_{BUS}脉冲时间 该域定义V_{BUS}的脉冲时间，真正的充电时间是1024*DVBUSPT[15:0] *$T_{USBCLOCK}$，$T_{USBCLOCK}$是USB时钟周期时间

设备 IN 端点 FIFO 空中断使能寄存器 （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

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

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

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	IEPTXFEIE[3:0]	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
位0对应IN端点0，位5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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	保留		SNAK	CNAK	TXFNUM[3:0]				STALL	保留	EPTYPE[1:0]		NAKS	保留
rs	rs			w	w		rw			rs		r	r	r	

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

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	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
15	EPACT	端点激活 对于端点0来说，该域固定为'1'
14:2	保留	必须保持复位值。
1:0	MPL[1:0]	最大包长 此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..3, x 是端点编号）

地址偏移：0x0900 + (x * 0x20)
复位值：0x0000 0000

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1PID	SD0PID/SEVENFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	保留	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w	rw				rw/rs		rw		r	r

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

rw rw

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	必须保持复位值。
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	必须保持复位值。
10:0	MPL[10:0]	该位定义最大包长

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	保留		SNAK	CNAK	保留				STALL	SNOOP	EPTYPE[1:0]		NAKS	保留
rs	r			w	w					rs	rw	r		r	

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

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	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
15	EPACT	端点激活 对于端点0，该域固定为1
14:2	保留	必须保持复位值。

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..3, x 是端点编号）

地址偏移：0x0B00 + (x * 0x20)
复位值：0x0000 0000

软件用该寄存器控制 OUT 端点 0 以外的每个逻辑 OUT 端点

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1PID	SEVENFRM / SD0PID	SNAK	CNAK	保留				STALL	SNOOP	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w					rw/rs	rw	rw		r	r

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

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	必须保持复位值。
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 此位控制端点是否激活，当端点没有激活，忽略任何令牌，不作任何回复
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…3），（x 是端点编号）

地址偏移：0x0908 + (x * 0x20)
复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当获得一个 IN 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常由硬件置位，除了 TXFE 位，各位写 1 清零。

该寄存器采用字（32 位）访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TXFE	IEPNE	保留	EPTXFUD	CITO	保留	EPDIS	TF
								r	rc_w1		rc_w1	rc_w1		rc_w1	rc_w1

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	TXFE	发送FIFO空 IN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或设置CNAK位来清零
5	保留	必须保持复位值。
4	EPTXFUD	端点Tx FIFO下溢

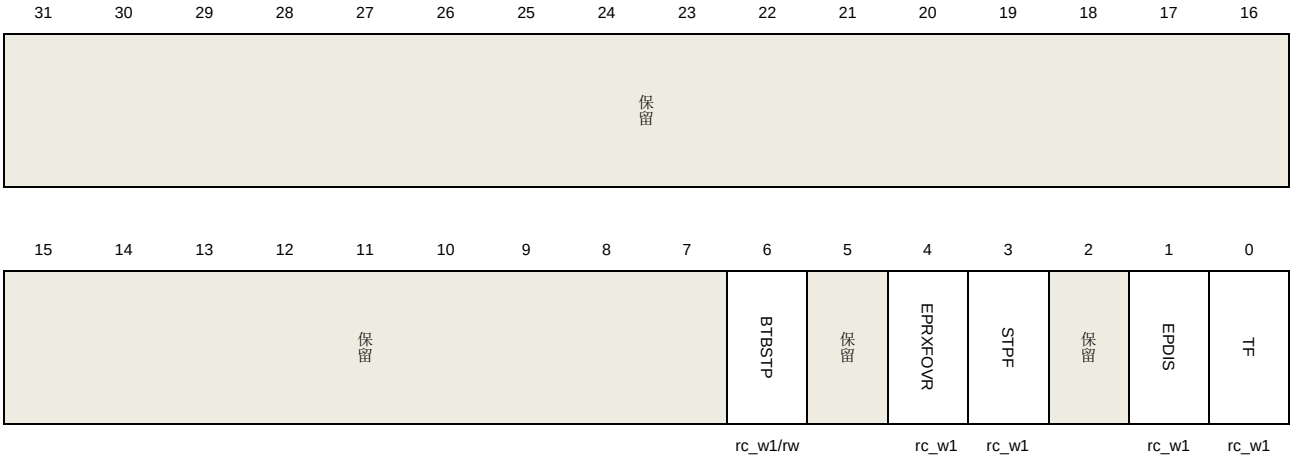
		如果当IN令牌被接收后，Tx FIFO没有包数据，该标志被触发。
3	CITO	控制IN事务超时中断 在控制IN事务中，如果设备等待的握手包超时，该标志位被触发
2	保留	必须保持复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

设备 OUT 端点 x 中断标志寄存器（USBFS_DOEPxINTF）（= 0..3, x 是端点编号）

地址偏移：0x0B08 + (x * 0x20)
复位值：0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当一个 OUT 端点的中断发生时，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	必须保持复位值。
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	必须保持复位值。
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBFS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是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	必须保持复位值。
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

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

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

位/位域	名称	描述
31:21	保留	必须保持复位值。
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递减。
18:7	保留	必须保持复位值。
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，在软件成功地将数据包写入端点的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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	STPCNT[1:0]		保留									PCNT	保留		
	rw											rw			

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

位/位域	名称	描述
31	保留	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
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，在软件成功地将数据包读取端点的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）（= 1…3, x 是端点编号）

地址偏移：0x910 + (x * 0x20)

复位值：0x0000 0000

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			PCNT[9:0]										TLEN[18:16]		
	rw							rw						rw	

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
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递减。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的数据包的总字节数，在端点使能之前，软件设置该位，在软件成功地将数据包写入端点的Tx FIFO中，该域按照数据包大小递减。

设备 OUT 端点 x 传输长度寄存器 （USBFS_DOEPxLEN）（x = 1..3, x 是端点编号）

地址偏移：0x0B10 + (x * 0x20)

复位值：0x0000 0000

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TL[1:0]	RXDPID/STPCN	PCNT[9:0]										TLEN[18:16]		
	r/rw							rw						rw	

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

位/位域	名称	描述
31	保留	必须保持复位值。
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]	包数目 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，该域数值递减。
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…3, x 是端点编号）

地址偏移：0x0918 + (x * 0x20)
复位值：0x0000 0200

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字（32 位）访问

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

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

IEPTFS[15:0]
r

位/位域	名称	描述
31:16	保留	必须保持复位值。
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：Tx FIFO满 1：1字可用 … n：n字可用

28.7.4. 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

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

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
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

29. 版本历史

表 29-1. 版本历史

版本号.	说明	日期
1.0	初稿发布	2015 年 7 月 1 日
2.0	适用于新的命名规范	2017 年 6 月 5 日
2.1	适用于新的文档规范	2018 年 10 月 25 日
2.2	1. 修改 CAN_STAT 和 CAN_TSTAT 寄存器； 2. 修改 ENET 接收描述符字 0（RDES0）Bit15 位。	2019 年 10 月 8 日
2.3	1. 修改 HXTAL 时钟范围为 4~32M； 2. 删除表 20-2 里关于 SAM_V 的描述； 3. 将 20.3.10 章节中 SMBTYPE 修改为 SMBSEL； 4. 在 16.1.3 章节添加关于喂完狗后要立刻进 deepsleep/standby 模式的注意事项。	2020 年 6 月 17 日
2.4	1. 修改 CAN_RFIFO 寄存器 RFD 位描述； 2. 更新页眉，删除首页页眉，更新 logo。	2020 年 9 月 4 日
2.5	1. 修改 ***13.4.2.控制寄存器(DBG_CTL)*** TIMER7/6/5/4 HOLD 位顺序； 2. 修改 26.4.17-26.4.22 寄存器，添加“(仅 CAN0 可用)”； 3. 修改 ***20.4.1. 控制寄存器 0（I2C_CTL0）***寄存器 SMBALT 位域描述。	2021 年 5 月 19 日
2.6	1. 修改 ***20.4.1. 控制寄存器 0（I2C_CTL0）*** 位 7 DISSTRC 为 SS，***20.4.7.传输状态寄存器 1（I2C_STAT1）***位 2TRS 为 TR，位（8-15）ECV 为 PECV； 2. 删除 ***设备 IN 端点 x 传输长度寄存器（USBFS_DIEPxLEN）（=1…3, x 是端点编号）***位 29-位 30 MCPF； 3. 修改 ***MAC 配置寄存器（ENET_MAC_CFG）***位 CSR 为 CRS。	2021 年 12 月 7 日
2.7	4. 删除***表 25 5. EXMC 对 NOR，PSRAM 和 SRAM 支持的访问模式***对 PSRAM 的同步 32 位写时 “split into 2 EXMC accesses”的注释；	2022 年 6 月 30 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 under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.

© 2022 GigaDevice – All rights reserved