Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33FR0 系列
低功耗 MCU 芯片

产品说明书

2022.12

本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。
在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。
采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。
未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。
今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(http://www.fmsh.com/)。
如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。
FUDAN MICRO 保留对资料任何信息进行更改的权利，资料的信息如有变更，恕不另行通知。FUDAN MICRO 对使用或依赖本资料中包含的任何信息等不承担任何责任。
本资料包含的所有信息均“按原样”提供，不提供任何形式的明示，暗示，法定或其他形式的保证。FUDAN MICRO 明确拒绝对适销性，非侵权性和针对特定用途的适用性方面的的所有明示或暗示保证。FUDAN MICRO 对本资料可能使用、包含或提供的任何第三方代码、软件、信息等不提供任何担保，并且用户同意仅向该等第三方寻求与此相关的任何担保索赔。 FUDAN MICRO 对于根据用户规格或为符合特定标准或公开论坛而产生的任何交付物，也不承担任何责任。FUDAN MICRO 未通过任何明示或暗示方式授予任何知识产权许可。

商标
上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。

章节列表

章节列表 **3**

表目录 **20**

图目录 **22**

1 产品综述 **28**

1.1 概述 28
1.2 芯片结构框图 31
1.3 产品型号列表 32

2 引脚和封装 **33**

2.1 封装和引脚排列 33
2.1.1 LQFP100 封装图（FM33FR0x10） 33
2.1.2 LQFP80 封装图（FM33FR0x8） 34
2.1.3 LQFP64 封装图（FM33FR0x6） 35
2.1.4 LQFP44 封装图（FM33FR0x4） 37
2.1.5 LQFP32 封装图（FM33FR0x3） 38
2.1.6 PA 引脚组数字外设功能 39
2.1.7 PB 引脚组数字外设功能 39
2.1.8 PC 引脚组数字外设功能 39
2.1.9 PD 引脚组数字外设功能 40
2.1.10 PE 引脚组数字外设功能 41
2.1.11 PF 引脚组数字外设功能 41
2.1.12 封装尺寸图 42

3 电参数 **52**

3.1 参数说明 52
3.2 推荐电源方案 52
3.2.1 供电方案 52
3.3 极限参数 54
3.4 性能参数 55
3.4.1 典型工作条件 55
3.4.2 功耗参数 55
3.4.3 复位和电源监控 57
3.4.4 内部基准电压 58
3.4.5 低功耗模式唤醒时间 59
3.4.6 外部时钟源特性 61
3.4.7 内部时钟源特性 63
3.4.8 PLL 特性 63
3.4.9 ADC 特性 65
3.4.10 温度传感器 68
3.4.11 模拟比较器特性 70
3.4.12 Flash 存储器特性 71
3.4.13 GPIO 特性 72
3.4.14 LCD 特性 73

4 总线与存储 **74**

4.1 系统总线 74
4.2 存储空间分配 76
4.2.1 概述 76

4.2.2 外设模块寄存器地址分配 78
4.3 RAM 80
4.3.1 概述 80
4.3.2 奇偶校验 80
4.4 FLASH 81
4.4.1 概述 81
4.4.2 数据 flash 81
4.4.3 特殊信息扇区说明 82
4.4.4 Flash 编程 86
4.4.5 Flash 的内容保护 89
4.5 寄存器 92
4.5.1 Flash 读取控制寄存器（FLS_RDCR） 92
4.5.2 用户配置字寄存器（FLS_OPTBR） 93
4.5.3 Flash 擦写控制寄存器（FLS_EPCR） 94
4.5.4 Flash Key 输入寄存器（FLS_KEY） 94
4.5.5 Flash 中断使能寄存器（FLS_IER） 95
4.5.6 Flash 标志寄存器（FLS_ISR） 95
4.5.7 Flash ECC 标志寄存器（FLS_ECCSR） 96
4.5.8 Flash ECC 控制寄存器（FLS_ECCCR） 97
4.5.9 ACLOCK 寄存器 1（FLS_ACLOCK1） 98
4.5.10 ACLOCK 寄存器 2（FLS_ACLOCK2） 99
4.5.11 ACLOCK 寄存器 3（FLS_ACLOCK3） 99

5 电源管理单元（PMU） 101

5.1 芯片工作电源 101
5.1.1 电源域划分 101
5.1.2 电源结构图 102
5.1.3 ADC 基准电压 102
5.1.4 片内快速建立基准源（AVREF） 102
5.1.5 内部 VREFP 产生（VREFINT_VREG） 103
5.2 功耗模式 103
5.2.1 概述 103
5.2.2 功耗模式与系统频率 104
5.2.3 Active 模式 104
5.2.4 LP Run 模式 105
5.2.5 SLEEP 模式 105
5.2.6 DEEPSLEEP 模式 106
5.3 唤醒源 107
5.4 休眠唤醒后的时钟控制 108
5.5 寄存器 109
5.5.1 低功耗控制寄存器（PMU_CR） 109
5.5.2 唤醒时间控制寄存器（PMU_WKTR） 110
5.5.3 唤醒源标志查询寄存器（PMU_WKFR） 111
5.5.4 PMU 中断使能寄存器（PMU_IER） 112
5.5.5 PMU 中断标志寄存器（PMU_ISR） 112
5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR） 113
5.5.7 VREFP 控制寄存器（PMU_VREFP_CR） 113
5.5.8 VREFP 配置寄存器（PMU_VREFP_CFGR） 114
5.5.9 VREFINT 调校寄存器（PMU_VREFINT_TR） 114
5.5.10 PTAT 控制寄存器（PMU_PTAT_CR） 115

6 VREFINT 基准电压（VREFINT_VREG） 116

6.1 概述 116
6.2 功能描述 116

6.2.1 概述 116
6.2.2 参考电压 119
6.2.3 输出电压 119
6.3 寄存器 120
7 权限管理（MDC） 121
7.1 寄存器 121
7.1.1 HardFault 查询寄存器 121
8 处理器（CPU） 123
8.1 概述 123
8.1.1 处理器配置 123
8.2 寄存器 124
8.3 异常和中断 125
8.3.1 中断向量表 125
8.3.2 中断优先级 125
8.3.3 错误处理 125
8.3.4 锁定（Lockup） 126
8.4 调试特性 127
8.4.1 调试功能引脚 127
8.4.2 调试状态下的看门狗控制 127
8.4.3 DEBUG 的复位 127
9 中断源多路选择（INTMUX） 128
9.1 概述 128
9.2 中断向量表 128
9.3 中断向量多路选择 129
9.4 寄存器 130
9.4.1 中断源多路选择控制寄存器（INTMUX_CR） 130
10 复位管理单元（RMU） 131
10.1 概述 131
10.2 模块框图 132
10.3 上下电复位 133
10.4 软件复位 134
10.5 NRST 引脚复位 134
10.6 寄存器 135
10.6.1 PDR 配置寄存器（RMU_PDRCR） 135
10.6.2 BOR 配置寄存器（RMU_BORCR） 135
10.6.3 复位配置寄存器（RMU_LKPCR） 136
10.6.4 软件复位寄存器（RMU_SOFTRST） 137
10.6.5 复位标志寄存器（RMU_RSTFR） 137
10.6.6 外设复位使能寄存器（RMU_PRSTEN） 138
10.6.7 AHB 外设复位寄存器（RMU_AHBRSTCR） 138
10.6.8 APB 外设复位寄存器 1（RMU_APBRSTCR1） 139
10.6.9 APB 外设复位寄存器 2（RMU_APBRSTCR2） 141
11 独立看门狗（IWDT） 143
11.1 概述 143
11.2 结构框图 143
11.3 IWDT 功能描述 143
11.4 IWDT 窗口功能 144
11.5 IWDT 冻结 145
11.6 寄存器 146

11.6.1 IWDT 清除寄存器（IWDT_SERV）.....146
11.6.2 IWDT 配置寄存器（IWDT_CR）.....146
11.6.3 IWDT 计数值寄存器（IWDT_CNT）.....147
11.6.4 IWDT 窗口寄存器（IWDT_WIN）.....148
11.6.5 IWDT 中断使能寄存器（IWDT_IER）.....148
11.6.6 IWDT 中断标志寄存器（IWDT_ISR）.....148

12 窗口看门狗（WWDT）.....150

12.1 功能描述.....150
12.2 结构框图.....150
12.3 WWDT 工作方式.....151
12.4 寄存器.....153
12.4.1 WWDT 控制寄存器（WWDT_CR）.....153
12.4.2 WWDT 配置寄存器（WWDT_CFGR）.....153
12.4.3 WWDT 计数寄存器（WWDT_CNT）.....154
12.4.4 WWDT 中断使能寄存器（WWDT_IER）.....154
12.4.5 WWDT 中断标志寄存器（WWDT_ISR）.....155
12.4.6 WWDT 预分频寄存器（WWDT_PSC）.....155

13 时钟管理单元（CMU）.....157

13.1 概述.....157
13.2 时钟架构.....158
13.2.1 时钟树.....158
13.2.2 SYSCLK 切换说明.....159
13.2.3 时钟安全性.....159
13.2.4 主要时钟说明.....159
13.2.5 外设模块的总线时钟和工作时钟.....160
13.2.6 休眠模式下的外设时钟.....161
13.2.7 LSCLK 切换逻辑.....161
13.2.8 RTC 时钟.....162
13.3 高频晶体振荡电路(XTHF).....163
13.3.1 概述.....163
13.3.2 工作方式.....163
13.3.3 停振检测（HFDET）.....163
13.3.4 XTHF 启动过程.....163
13.4 低频晶体振荡电路(XTLF).....164
13.4.1 概述.....164
13.4.2 工作方式.....164
13.4.3 停振检测（LFDET）.....164
13.4.4 XTLF 启动过程.....164
13.5 低功耗低频环振(RCLP).....165
13.5.1 概述.....165
13.6 锁相环(PLL).....166
13.6.1 概述.....166
13.6.2 应用注意.....166
13.7 时钟校准.....166
13.8 低功耗模式下的时钟源.....167
13.9 休眠唤醒的时钟处理.....167
13.10 寄存器.....168
13.10.1 系统时钟配置寄存器（CMU_SYSCLKCR）.....169
13.10.2 RCHF 时钟控制寄存器（CMU_RCHCR）.....170
13.10.3 RCHF 调校寄存器（CMU_RCHFTR）.....171
13.10.4 PLL 控制寄存器（CMU_PLLCR）.....171

13.10.5 RCLP 调校寄存器（CMU_RCLPTR）..... 172
13.10.6 XTLF 控制寄存器（CMU_XTLFCR）..... 173
13.10.7 LSCLK 选择寄存器（CMU_ LSCLKSEL）..... 174
13.10.8 XTHF 控制寄存器（CMU_XTHFCR）..... 175
13.10.9 CMU 中断使能寄存器（CMU_IER）..... 175
13.10.10 CMU 中断标志寄存器（CMU_ISR）..... 176
13.10.11 外设总线时钟控制寄存器 1（CMU_ PCLKCR1）..... 177
13.10.12 外设总线时钟控制寄存器 2（CMU_ PCLKCR2）..... 177
13.10.13 外设总线时钟控制寄存器 3（CMU_ PCLKCR3）..... 178
13.10.14 外设总线时钟控制寄存器 4（CMU_ PCLKCR4）..... 179
13.10.15 外设工作时钟配置寄存器 1（CMU_OPCCR1）..... 180
13.10.16 外设工作时钟配置寄存器 2（CMU_OPCCR2）..... 181
13.10.17 外设工作时钟配置寄存器 3（CMU_ OPCCR3）..... 182
13.10.18 AHB Master 控制寄存器（CMU_AHBMCR）..... 183
13.10.19 时钟校准控制寄存器（CMU_CCCR）..... 184
13.10.20 时钟校准配置寄存器（CMU_CCFR）..... 184
13.10.21 时钟校准计数值寄存器（CMU_CCNR）..... 185
13.10.22 时钟校准中断标志寄存器（CMU_CCISR）..... 185

14 电源电压监测（SVD）..... 187

14.1 概述..... 187
14.2 结构框图..... 187
14.3 引脚定义..... 188
14.4 功能描述..... 188
14.5 外部电源检测..... 189
14.6 电源检测阈值..... 190
14.7 寄存器..... 194
14.7.1 SVD 配置寄存器（SVD_CFGR）..... 194
14.7.2 SVD 控制寄存器（SVD_CR）..... 194
14.7.3 SVD 中断使能寄存器（SVD_IER）..... 195
14.7.4 SVD 状态和标志寄存器（SVD_ISR）..... 196
14.7.5 SVD 参考电压选择寄存器（SVD_VSR）..... 196

15 AES 硬件运算单元（AES）..... 198

15.1 功能描述..... 198
15.2 工作模式..... 198
15.3 AES 数据流处理模式..... 199
15.3.1 ECB 模式..... 199
15.3.2 CBC 模式..... 200
15.3.3 暂停模式..... 202
15.3.4 CTR 模式..... 203
15.3.5 CTR 模式下的暂停模式..... 204
15.3.6 GCM 模式..... 204
15.3.7 MultH 模块..... 207
15.3.8 推荐的 GCM 流程..... 208
15.4 数据类型..... 209
15.5 工作流程..... 210
15.5.1 模式 1：加密..... 210
15.5.2 模式 2：密钥扩展..... 211
15.5.3 模式 3：解密..... 212
15.5.4 模式 4：密钥扩展+解密..... 212
15.5.5 使用 MultH 模块..... 213
15.6 DMA 接口..... 214

15.6.1 MultH 模块与 DMA 间接口 215
15.7 错误标志 215
15.8 寄存器 216
15.8.1 AES 控制寄存器（AES_CR） 216
15.8.2 AES 中断使能寄存器（AES_IER） 218
15.8.3 AES 中断标志寄存器（AES_ISR） 219
15.8.4 AES 数据输入寄存器（AES_DIR） 219
15.8.5 AES 数据输出寄存器（AES_DOR） 220
15.8.6 AES 秘钥寄存器（AES_KEYx） 220
15.8.7 AES 初始向量寄存器（AES_IVRx） 221
15.8.8 AES MultH 参数寄存器（AES_Hx） 221
16 随机数发生器（TRNG） 222
16.1 概述 222
16.2 功能描述 223
16.2.1 随机数产生 223
16.2.2 工作时钟 223
16.2.3 随机数读取 223
16.2.4 CRC 运算 224
16.3 寄存器 225
16.3.1 随机数控制寄存器 225
16.3.2 随机数/CRC 结果输出寄存器 225
16.3.3 RNG 标志寄存器 226
16.3.4 CRC 控制寄存器 226
16.3.5 CRC 输入数据寄存器 227
16.3.6 CRC 标志寄存器 227
17 模拟比较器（COMPARATOR） 229
17.1 概述 229
17.2 结构框图 230
17.3 功能描述 231
17.3.1 基本功能 231
17.3.2 内部比较基准选择和产生 231
17.3.3 时钟和复位 231
17.3.4 引脚和内部信号连接 232
17.3.5 窗口功能 232
17.3.6 输出数字滤波 233
17.3.7 功耗和速度模式 234
17.3.8 比较器中断 234
17.3.9 比较器输出和触发信号输出 235
17.4 寄存器 237
17.4.1 COMP 控制寄存器 1（COMP_CR1） 237
17.4.2 COMP2 控制寄存器（COMP_CR2） 238
17.4.3 COMP 中断配置寄存器（COMP_ICR） 240
17.4.4 COMP 中断标志寄存器（COMP_ISR） 240
17.4.5 COMP 缓冲器控制寄存器（COMP_BUFCR） 241
18 硬件除法器（DIVAS） 242
18.1 概述 242
18.2 工作流程 242
18.3 寄存器 242
18.3.1 被除数寄存器 243
18.3.2 除数寄存器 243

18.3.3 商寄存器 244
18.3.4 余数寄存器 244
18.3.5 状态标志寄存器 244

19 双线串行总线和系统管理总线（I²C-SMBUS） 246

19.1 概述 246
19.2 主要功能 247
19.3 模块框图 248
19.3.1 I²C-SMB 模块框图 248
19.4 引脚定义 249
19.5 时钟和复位 250
19.6 接口时序 251
19.6.1 通信过程 251
19.6.2 接口时序描述 252
19.6.3 总线时序参数表 253
19.7 I²C 功能描述 255
19.7.1 预设从机地址表 255
19.7.2 I²C 初始化过程 256
19.7.3 I²C 主机功能 257
19.7.4 I²C 从机功能 274
19.8 SMBus 功能描述 281
19.8.1 概述 281
19.8.2 SMBus 初始化过程 284
19.8.3 SMBus 主机功能 285
19.8.4 SMBus 超时功能 288
19.8.5 CRC-8 289
19.9 寄存器 290
19.9.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1） 290
19.9.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2） 292
19.9.3 I2C_SMB 中断控制寄存器（I2CSMB_IER） 293
19.9.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR） 294
19.9.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR） 295
19.9.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TCR） 295
19.9.7 I2C_SMB 超时寄存器（I2CSMB_TOR） 296
19.9.8 I2C_SMB 接收数据寄存器（I2CSMB_RXBUF） 297
19.9.9 I2C_SMB 发送数据寄存器（I2CSMB_TXBUF） 297
19.9.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR） 297
19.9.11 CRC 数据寄存器（I2CSMB_CRC_DR） 298
19.9.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR） 299
19.9.13 CRC 多项式寄存器（I2CSMB_CRC_POLY） 299

20 双线串行总线主机（I²C MASTER） 301

20.1 概述 301
20.2 结构框图 301
20.3 时钟 302
20.4 接口时序 302
20.4.1 接口时序图 302
20.4.2 接口时序描述 303
20.5 I²C 工作模式 305
20.6 I²C 地址格式 305
20.7 I²C 初始化 305
20.7.1 主机波特率配置 305
20.8 I²C 主机功能 306

20.8.1 7bit 寻址 306
20.8.2 10bit 寻址 312
20.8.3 SCL 延展（Slave Clock Stretching） 315
20.8.4 超时机制 315
20.8.5 可编程时序 316
20.9 寄存器 317
20.9.1 I2C 主机配置寄存器（I2C_MSPCFGR） 317
20.9.2 I2C 主机控制寄存器（I2C_MSPCR） 318
20.9.3 I2C 主机中断使能寄存器（I2C_MSPIER） 319
20.9.4 I2C 主机中断标志寄存器（I2C_MSPISR） 319
20.9.5 I2C 主机状态寄存器（I2C_MSPSR） 320
20.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR） 321
20.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF） 322
20.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR） 322
20.9.9 I2C 主机超时寄存器（I2C_MSPTOR） 323
21 通用异步收发传输器（UART） 324
21.1 概述 324
21.2 结构框图 325
21.3 引脚定义 326
21.4 UART 类型区分 326
21.5 UART 字符描述 327
21.5.1 UART 模式 327
21.5.2 智能卡模式 327
21.6 功能描述 329
21.6.1 时钟结构 329
21.6.2 位接收采样 329
21.6.3 数据发送 330
21.6.4 数据接收 333
21.6.5 低功耗休眠唤醒（UART0/1） 333
21.6.6 使用 DMA 进行 UART 收发 334
21.6.7 DMA 模式下的发送完成中断 334
21.6.8 半双工单线通信 335
21.7 波特率发生 335
21.7.1 波特率发生 335
21.7.1 波特率自适应 337
21.8 红外调制 337
21.9 接收超时 338
21.10 发送延迟 338
21.11 寄存器 339
21.11.1 红外调制寄存器（UART_IRCR） 342
21.11.2 UARTx 控制状态寄存器（UARTx_CSR） 342
21.11.3 UARTx 中断使能寄存器（UARTx_IER） 344
21.11.4 UARTx 中断标志寄存器（UARTx_ISR） 345
21.11.5 UARTx 超时和延迟寄存器（UARTx_TODR） 346
21.11.6 UARTx 接收缓冲寄存器（UARTx_RXBUF） 346
21.11.7 UARTx 发送缓冲寄存器（UARTx_TXBUF） 347
21.11.8 UATRx 波特率产生寄存器（UARTx_BGR） 347
21.11.9 UARTx 模式寄存器（UARTx_MCR） 348
21.11.10 UARTx 智能卡状态标志寄存器（UARTx_SCISR） 349
21.11.11 UARTx 智能卡中断使能寄存器（UARTx_SCIER） 350
21.11.12 UARTx 智能卡帧格式控制寄存器（UARTx_FFCR） 351
21.11.13 UARTx 智能卡额外保护时间寄存器（UARTx_EGTR） 352

21.11.14 UARTx 智能卡输出时钟分频寄存器（UARTx_CODR）..... 353

22 串行外设接口（SPI/I2S）..... 354

22.1 概述..... 354
22.2 SPI 功能描述..... 355
22.2.1 SPI 结构框图..... 355
22.2.2 引脚定义..... 355
22.2.3 SPI 接口时序..... 356
22.2.4 I/O 配置..... 359
22.2.5 全双工数据通信..... 360
22.2.6 TX-ONLY 模式..... 361
22.2.7 RX-ONLY 模式..... 362
22.2.8 主机 SSN 控制..... 362
22.2.9 数据冲突..... 363
22.2.10 使用 DMA 进行 SPI 收发..... 363
22.3 I2S 功能描述..... 365
22.3.1 I2S 结构框图..... 365
22.3.2 引脚定义..... 365
22.3.3 协议支持..... 365
22.3.4 时钟产生..... 369
22.3.5 I2S 主机功能..... 371
22.3.6 I2S 从机功能..... 372
22.3.7 帧格式错误..... 373
22.3.8 DMA 支持..... 373
22.4 寄存器..... 374
22.4.1 SPI 控制寄存器 1（SPIx_CR1）..... 374
22.4.2 SPI 控制寄存器 2（SPIx_CR2）..... 375
22.4.3 SPI 控制寄存器 3（SPIx_CR3）..... 377
22.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）..... 377
22.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）..... 378
22.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）..... 379
22.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）..... 380
22.4.8 I2S 配置寄存器（SPIx_I2SCR）..... 380
22.4.9 I2S 预分频寄存器（SPIx_I2SPR）..... 381

23 直接存储访问控制器（DMA）..... 383

23.1 概述..... 383
23.2 工作原理..... 384
23.3 结构框图..... 385
23.4 访问带宽..... 385
23.5 通道控制..... 386
23.5.1 DMA 请求映射..... 386
23.5.2 通道优先级..... 387
23.5.3 传输方向定义..... 387
23.5.4 循环模式..... 387
23.5.5 TSI 通道..... 387
23.6 寄存器..... 389
23.6.1 DMA 全局控制寄存器（DMA_GCR）..... 390
23.6.2 通道 x 控制寄存器（DMA_CHxCR）..... 391
23.6.3 通道 x 存储器指针寄存器（DMA_CHxMAD）..... 392
23.6.4 通道 7 控制寄存器（DMA_CH7CR）..... 393
23.6.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）..... 394
23.6.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）..... 394

23.6.7 DMA 中断状态标志寄存器（DMA_ISR）..... 395
23.6.8 TSI 通道控制寄存器（DMA_TSICR）..... 396
23.6.9 TSI 通道 x 存储器指针寄存器（DMA_TSIxMAR）..... 396
23.6.10 TSI 接近检测通道存储器指针寄存器（DMA_TSIPROXMAR）..... 397

24 循环冗余校验（CRC）..... 398

24.1 概述..... 398
24.2 软件配置过程..... 399
24.3 GOLDEN 数据..... 400
24.4 DMA 接口..... 400
24.5 FLASH 数据完整性校验..... 401
24.6 寄存器..... 402
24.6.1 CRC 数据寄存器（CRC_DR）..... 402
24.6.2 CRC 控制状态寄存器（CRC_CR）..... 402
24.6.3 CRC LFSR 寄存器（CRC_LFSR）..... 404
24.6.4 CRC 输出异或寄存器（CRC_XOR）..... 404
24.6.5 CRC 多项式寄存器（CRC_POLY）..... 405

25 高级定时器（ATIM）..... 406

25.1 概述..... 406
25.2 主要特性..... 406
25.3 结构框图..... 407
25.4 功能描述..... 408
25.4.1 定时单元..... 408
25.4.2 定时器工作模式..... 410
25.4.3 重复计数器..... 417
25.4.4 Preload 寄存器..... 418
25.4.5 计数器工作时钟..... 419
25.4.6 内部触发信号（ITRx）..... 424
25.4.7 捕捉/比较通道..... 425
25.4.8 输入捕捉模式..... 427
25.4.9 软件 Force 输出..... 429
25.4.10 输出比较模式..... 430
25.4.11 PWM 输出..... 431
25.4.12 互补输出和死区插入..... 433
25.4.13 刹车功能..... 434
25.4.14 互补输出通道信号状态逻辑表..... 436
25.4.15 6-step PWM 输出..... 437
25.4.16 单脉冲输出..... 438
25.4.17 外部事件清除 OCxREF..... 440
25.4.18 编码器接口模式（encoder interface）..... 441
25.4.19 TIM 从机模式..... 443
25.4.20 DMA 访问..... 446
25.4.21 DMA Burst..... 447
25.4.22 输入异或功能..... 448
25.4.23 Debug 模式..... 448
25.5 寄存器..... 449
25.5.1 ATIM 控制寄存器 1（ATIM_CR1）..... 449
25.5.2 ATIM 控制寄存器 2（ATIM_CR2）..... 451
25.5.3 ATIM 从机模式控制寄存器（ATIM_SMCR）..... 452
25.5.4 ATIM DMA 和中断使能寄存器（ATIM_DIER）..... 454
25.5.5 ATIM 状态寄存器（ATIM_ISR）..... 455
25.5.6 ATIM 事件产生寄存器（ATIM_EGR）..... 456

25.5.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1）.....457
25.5.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2）.....459
25.5.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER）.....462
25.5.10 ATIM 计数器寄存器（ATIM_CNT）.....463
25.5.11 ATIM 预分频寄存器（ATIM_PSC）.....463
25.5.12 ATIM 自动重载（auto-reload）寄存器（ATIM_ARR）.....464
25.5.13 ATIM 重复计数寄存器（ATIM_RCR）.....464
25.5.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1）.....465
25.5.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2）.....465
25.5.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）.....466
25.5.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4）.....466
25.5.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR）.....467
25.5.19 ATIM DMA 控制寄存器（ATIM_DCR）.....468
25.5.20 ATIM DMA 访问寄存器（ATIM_DMAR）.....469
25.5.21 ATIM 刹车输入控制寄存器（ATIM_BKCTL）.....469

26 定时器阵列单元（TAU）.....472

26.1 概述.....472
26.2 结构框图.....473
26.3 功能描述.....474
26.3.1 时钟和复位.....474
26.3.2 工作模式.....474
26.3.3 输入输出通道.....474
26.3.4 16 位计数模式.....475
26.3.5 触发信号输出（TRGO）.....477
26.3.6 32 位级联模式.....477
26.3.7 捕捉功能.....482
26.3.8 预分频.....486
26.3.9 16 位 PWM 输出.....487
26.3.10 级联 PWM 输出.....488
26.3.11 外部引脚输入数字滤波.....492
26.4 寄存器.....493
26.4.1 TAU0 控制寄存器（TAU0_CR）.....495
26.4.2 TAU0 通道 x 配置寄存器（TAU0_TxCFGR）.....495
26.4.3 TAU0 通道 x 模式寄存器（TAU0_TxMDR）.....496
26.4.4 TAU0 重载寄存器（TAU0_TxARR）.....497
26.4.5 TAU0 捕捉比较寄存器（TAU0_TxCCR）.....498
26.4.6 TAU0 中断使能寄存器（TAU0_TxIER）.....498
26.4.7 TAU0 中断标志寄存器（TAU0_TxISR）.....499
26.4.8 TAU0 计数值寄存器（TAU0_TxCNTR）.....499
26.4.9 TAU1 控制寄存器（TAU1_CR）.....500
26.4.10 TAU1 通道 x 配置寄存器（TAU1_TxCFGR）.....500
26.4.11 TAU1 通道 x 模式寄存器（TAU1_TxMDR）.....501
26.4.12 TAU1 重载寄存器（TAU1_TxARR）.....502
26.4.13 TAU1 捕捉比较寄存器（TAU1_TxCCR）.....503
26.4.14 TAU1 中断使能寄存器（TAU1_TxIER）.....503
26.4.15 TAU1 中断标志寄存器（TAU1_TxISR）.....504
26.4.16 TAU1 计数值寄存器（TAU1_TxCNTR）.....504

27 16 位基本定时器（BSTIM16）.....506

27.1 概述.....506
27.2 主要特性.....506
27.3 结构框图.....506

27.4 功能描述 507
27.4.1 定时单元 507
27.4.2 定时器工作模式 509
27.4.3 计数器工作时钟 511
27.4.1 Debug 模式 512
27.5 寄存器 513
27.5.1 BSTIM 控制寄存器 1(BSTIM16_CR1) 513
27.5.2 BSTIM 控制寄存器 2(BSTIM16_CR2) 514
27.5.3 BSTIM 中断使能寄存器(BSTIM16_IER) 515
27.5.4 BSTIM 状态寄存器(BSTIM16_ISR) 515
27.5.5 BSTIM 事件产生寄存器(BSTIM16_EGR) 516
27.5.6 BSTIM 计数器寄存器(BSTIM16_CNT) 516
27.5.7 BSTIMx 预分频寄存器(BSTIM16_PSC) 517
27.5.8 BSTIMx 自动重载（auto-reload）寄存器(BSTIM16_ARR) 517

28 16 位低功耗定时器（LPTIM16） 518

28.1 概述 518
28.2 结构框图 519
28.3 时钟和复位 519
28.4 定时器功能 520
28.4.1 普通定时器 520
28.4.2 外部脉冲触发计数 520
28.4.3 外部异步脉冲计数 521
28.4.4 Timeout 模式 521
28.5 捕捉比较功能 522
28.5.1 16bit PWM 522
28.5.2 输入捕捉 523
28.5.1 输入数字滤波 524
28.6 正交编码器 525
28.7 寄存器 527
28.7.1 LPTIM 配置寄存器（LPTIM16_CFGR） 527
28.7.2 LPTIM 计数值寄存器（LPTIM16_CNT） 529
28.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR） 529
28.7.4 LPTIM 目标值寄存器（LPTIM16_ARR） 531
28.7.5 LPTIM 中断使能寄存器（LPTIM16_IER） 532
28.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR） 532
28.7.7 LPTIM 控制寄存器（LPTIM16_CR） 534
28.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1） 534
28.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2） 534

29 实时时钟（RTC） 536

29.1 概述 536
29.2 结构框图 536
29.3 工作原理 537
29.3.1 时基计数器（LTBC） 537
29.3.2 LTBC 数字调校 538
29.3.3 BCD 时间 539
29.3.4 RTC 使能与停止 540
29.3.5 RTC 时间设置 540
29.3.6 RTC 时间读取 540
29.3.7 闰年判断 541
29.4 寄存器 542
29.4.1 RTC 写使能寄存器（RTC_WER） 542

29.4.2 RTC 中断使能寄存器（RTC_IER）..... 543
29.4.3 RTC 中断标志寄存器（RTC_ISR）..... 544
29.4.4 BCD 时间秒寄存器（RTC_BCDSEC）..... 545
29.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）..... 546
29.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）..... 546
29.4.7 BCD 时间天寄存器（RTC_BCDDAY）..... 547
29.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）..... 547
29.4.9 BCD 时间月寄存器（RTC_BCDMONTH）..... 548
29.4.10 BCD 时间年寄存器（RTC_BCDYEAR）..... 548
29.4.11 闹钟寄存器（RTC_ALARM）..... 549
29.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）..... 549
29.4.13 LTBC 数值调整寄存器（RTC_ADJUST）..... 550
29.4.14 毫秒计数值寄存器（RTC_SBSCNT）..... 551
29.4.15 RTC 控制寄存器（RTC_CR）..... 551

30 段码液晶/LED 驱动（LCD/LED）..... 553

30.1 概述..... 553
30.2 结构框图..... 553
30.3 IO 配置..... 554
30.4 功能说明..... 554
30.4.1 工作时钟和显示帧频率..... 554
30.4.2 LED 扫描波形..... 555
30.4.3 LCD Type A 扫描波形..... 556
30.4.4 LCD Type B 扫描波形..... 557
30.4.5 片内 buffer 驱动模式..... 558
30.4.6 显示闪烁功能..... 559
30.4.7 偏置电压调整..... 559
30.4.8 使用 LED 扫描功能实现键盘扫描..... 560
30.5 低功耗模式..... 561
30.6 寄存器..... 562
30.6.1 显示控制寄存器（LCD_CR）..... 562
30.6.2 显示测试控制寄存器（LCD_TEST）..... 564
30.6.3 显示频率控制寄存器（LCD_FCR）..... 565
30.6.4 闪烁时间寄存器（LCD_FLCKT）..... 565
30.6.5 显示中断使能寄存器（LCD_IER）..... 566
30.6.6 显示中断标志寄存器（LCD_ISR）..... 566
30.6.7 显示数据寄存器（LCD_DATAx）..... 567
30.6.8 LCD COM 使能控制寄存器(LCD_COMEN)..... 587
30.6.9 LCD SEG 使能控制寄存器 0(LCD_SEGEN0)..... 587
30.6.10 LCD SEG 使能控制寄存器 1(LCD_SEGEN1)..... 588

31 模数转换器（ADC）..... 589

31.1 概述..... 589
31.2 结构框图..... 590
31.3 输入通道..... 591
31.4 单端和差分输入..... 591
31.5 工作时序..... 594
31.6 功能描述..... 596
31.6.1 使用 VDDA 作为基准..... 596
31.6.2 使用 VREFINT 作为基准..... 596
31.6.3 温度传感器..... 596
31.6.4 温度传感器的斜率和标定..... 597
31.6.5 可编程采样时间..... 598

31.6.6 输出位宽选择 599
31.6.7 转换模式 599
31.6.8 转换触发 603
31.6.9 过采样和硬件平均 604
31.6.10 ADC 工作时钟 605
31.6.11 数据冲突和自动等待 607
31.6.12 DMA 608
31.6.13 模拟窗口看门狗（AWD） 614
31.6.14 ADC 校准 615
31.6.15 模拟 buffer 616
31.7 低功耗模式 616
31.8 寄存器 617
31.8.1 ADC 中断和状态寄存器（ADC_ISR） 617
31.8.2 ADC 中断使能寄存器（ ADC_IER） 618
31.8.3 ADC 控制寄存器 1（ADC_CR1） 619
31.8.4 ADC 控制寄存器 2（ADC_CR2） 619
31.8.5 ADC 校准控制寄存器（ADC_CALR） 620
31.8.6 ADC 配置寄存器 1（ADC_CFGR1） 621
31.8.7 ADC 配置寄存器 2（ADC_CFGR2） 622
31.8.8 ADC 采样时间控制寄存器（ADC_SMTR） 624
31.8.9 ADC 通道控制寄存器（ADC_CHER） 626
31.8.10 ADC 通道差分控制寄存器（ADC_DCR） 627
31.8.11 ADC 数据寄存器（ADC_DR） 627
31.8.12 AWD 阈值寄存器（ADC_HLTR） 628
31.8.13 BUF4TST 配置寄存器(BUF4TST_CFGR) 628
32 可编程胶合逻辑（PGL） 630
32.1 概述 630
32.2 结构框图 630
32.3 引脚定义 632
32.4 功能描述 632
32.4.1 LUT 真值表 632
32.4.2 LUT 输入 633
32.4.3 LUT 输出 633
32.4.4 滤波和采样 634
32.4.5 中断和触发 634
32.4.6 低功耗模式 634
32.5 寄存器 636
32.5.1 PGL 控制寄存器（PGL_CR） 636
32.5.2 PGL 配置寄存器 0（PGL_CFGR0） 636
32.5.3 PGL 配置寄存器 1（PGL_CFGR1） 638
32.5.4 PGL 配置寄存器 2（PGL_CFGR2） 639
32.5.5 PGL 配置寄存器 3（PGL_CFGR3） 640
32.5.6 PGL 中断使能寄存器（PGL_IER） 641
32.5.7 PGL 中断标志寄存器（PGL_ISR） 641
32.5.8 LUT0 真值表寄存器（PGL_LUT0） 642
32.5.9 LUT1 真值表寄存器（PGL_LUT1） 642
32.5.10 LUT2 真值表寄存器（PGL_LUT2） 643
32.5.11 LUT3 真值表寄存器（PGL_LUT3） 643
33 电容触摸按键（TSI） 645
33.1 概述 645
33.2 工作原理 645

33.2.1 电容检测 645
33.2.2 信噪比（SNR） 646
33.2.3 按键触摸判定 646
33.2.4 接近检测 647
33.2.5 按键超时检测 647
33.2.6 按键互斥 647
33.3 通道扫描 647
33.3.1 扫描过程 647
33.3.2 扫描模式 648
33.3.3 扫描触发 648
33.4 实现方案 650
33.4.1 时钟方案 650
33.4.2 数字滤波器 651
33.4.3 Baseline 跟踪 652
33.4.4 中断和状态标志 653
33.4.5 DMA 支持 653
33.5 寄存器 656
33.5.1 按键中断标志寄存器（TSI_ISR） 656
33.5.2 按键中断使能寄存器（TSI_IER） 657
33.5.3 扫描状态寄存器（TSI_SSR） 658
33.5.4 扫描配置寄存器（TSI_CFGR） 658
33.5.5 扫描控制寄存器（TSI_CR） 660
33.5.6 按键通道配置寄存器（TSI_CH_CFGR） 661
33.5.7 Raw Count 寄存器（TSI_RAW_CNTR） 661
33.5.8 时钟和周期控制寄存器（TSI_CCR） 662
33.5.9 DMA 控制寄存器（TSI_DMACR） 663
33.5.10 按键通道扫描配置寄存器（TSI_CHx_CFGR） 663
33.5.11 接近按键扫描配置寄存器（TSI_PROX_CFGR） 664
33.5.12 IDAC 调校寄存器（TSI_IDAC_TR） 665
33.5.13 IDAC 模拟控制寄存器（TSI_ANA_CR） 665
33.5.14 TSC_DFT 寄存器（TSI_TEST） 666
33.5.15 ANA TEST 寄存器（TSI_ANA_TEST） 667

34 I/O 端口（GPIO） 669

34.1 概述 669
34.2 引脚类型 669
34.2.1 GPIO，输入输出使能，可控上下拉电阻，可控开漏输出，可控驱动能力 670
34.2.2 GPIO，输入输出使能，真开漏输出（PB0, PB1） 671
34.2.3 PD10（NRST），仅上拉，无开漏输出，无可控驱动 672
34.3 IO 端口功能定义 672
34.3.1 GPIO 输入 673
34.3.2 GPIO 输出 673
34.3.3 数字外设功能 673
34.3.4 模拟功能 674
34.3.5 使用外部晶体引脚 675
34.4 NRST 引脚与 GPIO 复用 675
34.5 SWD 引脚 676
34.6 WKUPx 引脚 676
34.7 外部引脚中断（EXTI） 677
34.7.1 功能说明 677
34.7.2 应用指南 678
34.8 快速 GPIO 输出 679
34.9 寄存器 680

34.9.1 GPIO 输入使能寄存器（GPIOx_INEN）..... 683
34.9.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）..... 684
34.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN）..... 685
34.9.4 GPIO 功能选择寄存器（GPIOx_FCR）..... 685
34.9.5 GPIO 输出数据寄存器（GPIOx_DO）..... 687
34.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET）..... 688
34.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST）..... 688
34.9.8 GPIO 输入数据寄存器（GPIOx_DIN）..... 689
34.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）..... 690
34.9.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN）..... 690
34.9.11 GPIO 驱动强度配置寄存器（GPIOx_DSR）..... 691
34.9.12 GPIOC 定时器通道选择寄存器（GPIOC_TMUXR）..... 692
34.9.13 EXTI 输入选择寄存器 0（GPIO_EXTISEL0）..... 693
34.9.14 EXTI 输入选择寄存器 1（GPIO_EXTISEL1）..... 695
34.9.15 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0）..... 696
34.9.16 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1）..... 697
34.9.17 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）..... 697
34.9.18 EXTI 中断标志（GPIO_EXTIISR）..... 698
34.9.19 EXTI 输入信号寄存器（GPIO_EXTIDI）..... 698
34.9.20 FOUT 配置寄存器（GPIO_FOUTSEL）..... 699
34.9.21 WKUP 控制寄存器（GPIO_PINWKEN）..... 700

35 安全功能（SFU）..... 702

35.1 概述 702
35.2 闪存 CRC 检查 702
35.3 RAM 奇偶校验 702
35.4 RAM 写保护 703
35.5 关键寄存器写保护 703
35.6 非法地址访问检测 704
35.7 时钟频率检测 706
35.8 ADC 测试 706
35.9 寄存器 708
35.9.1 RAM 奇偶校验控制寄存器（SFU_RAMPCR）..... 708
35.9.2 RAM 奇偶校验状态寄存器（SFU_RAMPSR）..... 708
35.9.3 存储访问保护控制寄存器（SFU_MAPCR）..... 709
35.9.4 存储访问保护状态寄存器（SFU_MAPSR）..... 710

36 专用编程接口 711

36.1 概述 711
36.2 编程器使用 711

37 调试支持 712

37.1 概述 712
37.2 DEBUG 引脚 713
37.2.1 SWD 引脚 713
37.2.2 上拉电阻 713
37.3 SWD 接口协议 713
37.3.1 协议简介 713
37.3.2 传输序列 713
37.3.3 SW-DP ID code 714
37.3.4 主机读操作 715
37.3.5 主机写操作 715
37.4 SWD-DP 寄存器 716
37.4.1 寄存器列表 716

37.5 CORE DEBUG 寄存器 716
37.6 低功耗调试支持 716
37.7 DEBUG 相关的配置项 717
37.8 寄存器 717
37.8.1 MCU DEBUG 配置寄存器 717
37.8.2 HardFault 查询寄存器 718
38 器件签名信息 720
38.1 唯一器件 ID 720
版本列表 721
上海复旦微电子集团股份有限公司销售及服务网点 722

表目录

表 1-1FM33FR0xx 型号列表 32
表 3-1FM33FR0 极限参数 54
表 3-2FM33FR0 典型工作条件 55
表 3-3ACTIVE 电流参数 55
表 3-4LP RUN 电流参数 56
表 3-5 SLEEP 电流参数 56
表 3-6 DEEPSLEEP 电流参数 56
表 3-7 复位和电源监控参数 58
表 3-8 内部基准电压参数 59
表 3-9 内部基准整流参数 59
表 3-10 唤醒时间参数 59
表 3-11 低频晶体振荡器参数 61
表 3-12 高频晶体振荡器参数 62
表 3-13 内部 RC 振荡器参数 63
表 3-14 内部 RC 振荡器参数 63
表 3-15 PLL 参数 64
表 3-16 ADC 参数 67
表 3-17 ADC 采样时间 68
表 3-18 ADC 采样时间 69
表 3-19 温度传感器参数 69
表 3-20 模拟比较器参数 70
表 3-21 FLASH 参数 71
表 3-22 普通 I/O 参数 72
表 3-23 NRST 引脚参数 72
表 3-24 引脚 AC 参数 73
表 3-25 LCD 片内电阻分压 73
表 4-1 外设模块总线地址列表 79
表 8-1FM33FR0 CPU 配置简表 123
表 8-2 CORTEX-M0 内核寄存器简表 124
表 9-1 中断向量表 129
表 9-2 中断源多路选择表 129
表 11-1 IWDT 溢出周期表 144
表 12-1 WWDT 溢出周期表 152
表 17-1 比较器 1 引脚列表 232
表 17-2 比较器 2 引脚列表 232
表 17-3 比较器 1 工作模式 234
表 17-4 比较器 2 工作模式 234
表 19-1 I²C 引脚列表 249
表 19-2 I²C 接口时序要求 253
表 19-3 I²C-SMBUS 从机保留地址定义 256
表 19-4SMBUS 超时事件 282
表 20-1 I²C 接口时序要求 304
表 20-2 I²C 保留地址定义 305
表 21-1 UART 引脚列表 326
表 21-2 UART 类型列表 326
表 21-3 UART 数据帧格式 327
表 21-4 DMA 发送中断 335
表 21-5 常用时钟频率下波特率计算 336
表 22-1SPI 引脚列表 356

表 23- 2I2S 典型采样频率计算表 370
表 23- 1PDMA 通道映射列表 386
表 23- 2PDMA 各通道源选择 386
表 25- 1ENCODER INTERFACE 计数方式 441
表 26- 1 定时器级联触发选择 482
表 27- 2 定时器捕捉源选择 486
表 26- 3 定时器捕捉源选择 486
表 28- 1 ENCODER INTERFACE 计数方式 525
表 31- 1 帧频率计算公式 554
表 31- 2 典型帧频率和 DF 的关系 555
表 30- 3 LCD 与低功耗模式 561
表 31- 1 ADC 输入通道 591
表 31- 2 ADC 采样时间 598
表 31- 3 ADC 输出位宽与速度 599
表 31- 4 DMA 配置与功能 609
表 32- 1 PGL 相关引脚 632
表 32- 2 LUT 输出连接 633
表 34- 1GPIO 功能逻辑定义表 670
表 34- 2 真开漏 IO 功能逻辑定义表 672
表 35- 3FCR 定义表 673
表 34- 4PAD 通道配置 674
表 35- 1384K FLASH 版本总线访问控制表 705
表 35- 2256K FLASH 版本总线访问控制表 705
表 35- 3 128K FLASH 版本总线访问控制表 706

图目录

图 1-1 芯片结构框图 31
图 2-1 FM33FR0x10 LQFP100 封装图 33
图 2-2 FM33FR0x8 LQFP80 封装图 34
图 2-3 FM33FR0x6 LQFP64 封装图 35
图 2-4 FM33FR0x4 LQFP44 封装图 37
图 2-5 FM33FR0x3 LQFP32 封装图 38
图 2-6 LQFP100 封装尺寸图 42
图 2-7 LQFP80 封装尺寸图 44
图 2-8 LQFP64 封装尺寸图 45
图 2-9LQFP44 封装尺寸图 48
图 2-10LQFP32 封装尺寸图 50
图 3-1FM33FR0xx 供电方案 53
图 3-2 ADC 参数说明 65
图 3-3 ADC 通道输入阻抗 68
图 4-1 系统总线示意图 75
图 4-2FM33FR0 总线地址（384+32+8） 76
图 4-3FM33FR0 总线地址（256+24+8） 77
图 4-4FM33FR0 总线地址（128+16+8） 78
图 5-1 芯片电源结构图 102
图 5-2 功耗模式与系统主频 104
图 6-1 ADC 基准源 116
图 6-2 外部电源基准方案 117
图 6-3 外部独立基准方案 118
图 6-4ADC 使用内部基准方案的系统连接 119
图 10-1 芯片复位源框图 132
图 10-2 上下电复位示意图 133
图 10-3 POR, BOR, PDR 复位波形示意图 134
图 11-1 IWDT 结构框图 143
图 11-2 IWDT 窗口示意图 145
图 12-1WWDT 结构框图 150
图 12-2WWDT 窗口示意图 152
图 13-1 芯片时钟框图 158
图 14-1 低压检测电路框图 187
图 14-2 低压检测电路工作时序 188
图 15-1 ECB 模式加密流程 199
图 15-2 ECB 模式解密流程 200
图 15-3 CBC 加密过程 201
图 15-4 CBC 解密过程 202
图 15-5 暂停模式流程 202
图 15-6 CTR 加密流程 203
图 15-7 CTR 解密流程 204
图 15-8 32 位计数器和随机数的存储方式 204
图 15-9 GCM 加密流程 206
图 15-10 GCM 解密流程 207
图 15-11 MULTH 模块框图 208
图 15-12 根据数据类型存储数据的示意图 210
图 15-13 模式 1：加密流程 211
图 15-14 模式 2 示意图 211
图 15-15 模式 3 示意图 212

图 15-16 模式 4 示意图 213
图 15-17 MULTH 模块使用流程示意图 214
图 15-18 输入时 DMA 请求和数据传输示意图 214
图 15-19 输出时 DMA 请求和数据传输示意图 215
图 16-1 真随机数模块框图 223
图 16-2 真随机数模块工作时钟 223
图 17-1 比较器电路框图 230
图 17-2 比较器内建基准缓冲器 231
图 17-3 窗口比较器框图 233
图 17-4 窗口比较器波形示意图 233
图 17-5 数字滤波（DFLEN=3）波形示意图 234
图 17-6 比较器中断产生 235
图 17-7 比较器输出逻辑 235
图 17-8 比较器输出上升沿产生触发输出，无滤波 236
图 17-9 比较器输出上升沿下降沿产生触发输出，无滤波 236
图 19-1 I²C-SMB 模块框图 248
图 19-2 I²C 总线时序 251
图 19-3 数据有效时序 251
图 19-4 起始（START）与停止(STOP)命令定义 251
图 19-5 输出应答(ACK) 252
图 19-6I²C 时序参数图例 254
图 19-7 主机向 7 位地址从机写入数据时的帧格式 258
图 19-8I²C 主机软件发送数据流图 259
图 19-9 I²C 主机对 7 位地址从机发送数据流图 260
图 19-10 主机从 7 位地址从机读取数据时的帧格式 260
图 19-11 I²C 软件接收数据流图 261
图 19-12 I²C 从 7 位地址从机读取数据流图 262
图 19-13 I²C 主机接收数据等待 262
图 19-14 双向数据通信帧格式 263
图 19-15 10BIT 寻址，主机向从机写入数据 263
图 19-16I²C 软件发送数据流图 264
图 19-17 10BIT 寻址，主机从从机读取数据 265
图 19-18 I²C 软件发送数据流图 266
图 19-19 I²C 软件发送数据流图 267
图 19-20 I²C 主机 DMA 发送流程图 268
图 19-21 I²C 主机 DMA 接收流程图 270
图 19-22 I²C 多主机时钟同步 271
图 19-23 I²C 多主机数据仲裁 272
图 19-24 主机时序控制 273
图 19-25 I²C 从机数据发送波形 275
图 19-26 I²C 从机数据接收波形 276
图 19-27 I²C 从机数据接收波形（SCLSEN=0，接收溢出） 277
图 19-28 从机地址匹配唤醒 278
图 19-29 I²C 从机 DMA 接收流程图 279
图 19-30 I2C 从机 DMA 发送流程图 281
图 19-31$T_{LOW:SEXT}$ 和 $T_{LOW:MEXT}$ 波形 283
图 19-32 从机地址匹配唤醒 287
图 19-33 从机 START 唤醒 287
图 19-34SMBUS 总线时钟超时事件 288
图 19-35 主机和从机超时事件 289
图 20-1 I²C 主机模块框图 301
图 20-2I²C 总线时序 302
图 20-3 数据有效时序 302

图 20-4 起始（START）与停止(STOP)命令定义 303
图 20-5 输出应答(ACK) 303
图 20-6 主机向 7 位地址从机写入数据时的帧格式 307
图 20-7I²C 软件发送数据流图 308
图 20-8 I²C 主机对 7 位地址从机发送数据流图 309
图 20-9 主机从 7 位地址从机读取数据时的帧格式 309
图 20-10 I²C 软件发送数据流图 310
图 20-11 I²C 从 7 位地址从机读取数据流图 311
图 20-12 双向数据通信帧格式 311
图 20-13 10BIT 寻址，主机向从机写入数据 312
图 20-14I²C 软件发送数据流图 313
图 20-15 10BIT 寻址，主机从从机读取数据 313
图 20-16 I²C 软件发送数据流图 314
图 20-17 I²C 软件发送数据流图 315
图 20-20 主机时序控制 316
图 21-1 UART 接口时序 325
图 21-2 UART 字符描述 327
图 21-3 7816-3 帧结构图 328
图 21-4 位接收 16 倍采样 329
图 21-5 位接收 8 倍采样 330
图 21-6 UART 异步发送波形 2 332
图 21-7 UART 异步发送波形 3 332
图 21-8 红外调制波形 338
图 21-9 UART 发送延迟 338
图 22-1 SPI 结构框图 355
图 22-2 SPI 数据/时钟时序图（CPHA=0） 357
图 22-3 SPI 数据/时钟时序图（CPHA=1） 357
图 22-44 线半双工写操作 358
图 22-5 4 线半双工读操作（无 DUMMY CYCLE） 358
图 22-6 4 线半双工读操作（有 DUMMY CYCLE） 359
图 22-7 SPI MASTER/SPI SLAVE 互连 360
图 22-8 SPI SSN 时序图（SSNM=1） 362
图 22-9 SPI SSN 时序图（SSNM=0） 362
图 22-10I2S 结构框图 365
图 22-11PHILIPS 标准时序 366
图 22-1232/16-BITS MSB-JUSTIFIED 时序 367
图 23-1324-BITS MSB-JUSTIFIED 时序 368
图 22-14LSB-JUSTIFIED 时序 368
图 22-15PCM 标准时序 369
图 23-1 DMA 结构框图 385
图 24-1CRC 运算流程图 399
图 24-2 使用 DMA 对 RAM 中的数据进行 CRC 运算 401
图 25-1 高级定时器结构框图 407
图 25-2 预分频从 1 变为 2 的波形 409
图 25-3 预分频从 1 变为 4 的波形 409
图 25-4 向上计数波形，内部时钟不分频 410
图 25-5 向上计数波形，内部时钟 2 分频 411
图 25-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件 411
图 25-7ARPE=1（ATIM_ARR 预装载）时的更新事件 412
图 25-8 向下计数，内部时钟不分频 413
图 25-9 向下计数，内部时钟 2 分频 413
图 25-10 向下计数，内部时钟 2 分频 414
图 25-11 向下计数，不使用重复计数时的更新事件 414

图 25-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6 415
图 25-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢) 416
图 25-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出) 416
图 25-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置 417
图 25-16 ATIM 时钟源框图 419
图 25-17 内部时钟源模式，时钟分频因子为 1 419
图 25-18TI2 外部时钟连接例子 420
图 25-19 外部时钟模式 1 下的时序 420
图 25-20 外部时钟模式 1 下的时序 421
图 25-21 外部触发输入框图 422
图 25-22 外部时钟模式 2 下的时序 1 422
图 25-23 外部时钟模式 2 下的时序 2 423
图 25-24 捕获/比较通道(通道 1 输入部分) 425
图 25-25 捕获/比较通道 1 的主电路 425
图 25-26 捕获/比较通道的输出部分(通道 1 至 3) 426
图 25-27 捕获/比较通道的输出部分(通道 4) 426
图 25-28 PWM 输入捕获模式时序 427
图 25-29 输出比较模式，翻转 OC1 430
图 25-30 边沿对齐的 PWM 波形(ARR=7) 431
图 25-31 中央对齐的 PWM 波形(APR=7) 432
图 25-32 带死区插入的互补输出 433
图 25-33 死区波形延迟大于负脉冲 433
图 25-34 死区波形延迟大于正脉冲 433
图 25-35 响应刹车的输出 435
图 25-36 产生六步 PWM，使用 COM 的例子(OSSR=1) 437
图 25-37 单脉冲模式的例子 438
图 25-38 ETR 信号清除 ATIM 的 OCxREF 440
图 25-39 编码器模式下的计数器操作实例 441
图 25-40 复位模式下的时序 443
图 25-41 门控模式下的时序 444
图 25-42 触发器模式下的时序 445
图 25-43 外部时钟模式 2＋触发模式下的时序 446
图 26-1TAU 结构框图 473
图 26-2TimerX 结构框图 473
图 26-3 16bit 连续定时功能 475
图 26-4 定时器内部时钟预分频计数 476
图 26-5 定时器外部事件上升沿递增时序 476
图 26-6 定时器外部事件上升下降沿递增，2 分频时序 476
图 26-7 定时器内部信号下降沿递增 477
图 26-8 定时器级联计数（TRIGGER TO INCREMENT） 478
图 26-9 定时器级联计数（TRIGGER TO INCREMENT），从机 PRESCALE1 = 1 478
图 26-10 定时器级联计数（TRIGGER TO START） 479
图 26-11 定时器级联计数（TRIGGER TO START），从机无法产生溢出 480
图 27-12 定时器级联计数（TRIGGER TO START），从机计数预分频 481
图 26-13 定时器捕捉外部输入信号上升沿 483
图 26-14 定时器 2 分频捕捉外部输入信号下降沿 483
图 26-15 定时器脉冲宽度高电平捕捉 484
图 26-16TI00 二分频计数 487
图 26-17TI00 二分频捕捉 487
图 26-1816 位 PWM 输出示意图 487
图 26-19 从机 ARR 大于主机的情况 490
图 26-20 非对称非交叠 PWM 输出示意图 491
图 26-21Tixx 输入数字滤波示意图 492

图 27-1 16 位基本定时器结构框图 506
图 27-2 预分频从 1 变为 2 的波形 508
图 27-3 预分频从 1 变为 4 的波形 508
图 27-4 向上计数波形，内部时钟不分频 509
图 27-5 向上计数波形，内部时钟 2 分频 510
图 28-6 ARPE=0（ARR 没有预装载）时的更新事件 510
图 27-7 ARPE=1（ARR 预装载）时的更新事件 511
图 27-8 内部时钟源模式，时钟分频因子为 1 511
图 28-1 LPTIM16 结构框图 519
图 28-2 外部 ETR 脉冲上升沿触发计数 520
图 28-3 外部 ETR 脉冲异步计数（下降沿） 521
图 28-4TIMEOUT 模式 522
图 29-5PWM 输出 523
图 28-6 输入信号边沿捕捉 524
图 28-7 通道输入数字滤波 524
图 28-8 编码器模式下的计数器操作实例 526
图 29-1RTC 结构框图 536
图 29-2LTBC 结构框图 537
图 29-3RTC 时间读取流程图 541
图 30-1 LCD 显示控制模块结构框图 554
图 30-2LED 驱动波形(8 SEG, 3 DIG) 556
图 30-3LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE A) 557
图 30-4 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE B) 558
图 31-5LCD 片内电阻 BUFFER 型驱动电路 559
图 31-1 ADC 结构框图 590
图 32-2 单端输入 592
图 32-3 差分输入 592
图 31-4 差分输入信号与码字的关系 593
图 31-5 ADC 校准时序 594
图 31-6 ADC 采样转换时序 594
图 31-7 ADC 采样序列时序 595
图 31-8 ADC 输入通道示意图 598
图 31-9 ADC 单次转换全自动触发模式 600
图 31-10 ADC 单次转换半自动触发模式 602
图 31-11 ADC 连续转换模式 603
图 31-12 ADC 硬件触发源 604
图 31-13 ADC 时钟示意图 607
图 31-14 ADC 自动等待 607
图 31-15 ADC 单次全自动触发+DMA 案例 1 610
图 31-16 ADC 单次全自动触发+DMA 案例 2 611
图 31-17 ADC 单次半自动触发+DMA 612
图 31-18 ADC 全自动触发+DMA 循环模式 613
图 31-19 ADC 连续模式+DMA 循环模式 614
图 31-20 ADC 模拟看门狗阈值示意图 615
图 32-1LUT 结构框图 631
图 32-2 PGL 结构框图 631
图 32-3 LUT 实现 2 输入 NAND 示意图 633
图 32-4 输出滤波和采样 634
图 32-5 数字滤波 634
图 33-1 按键触摸与释放 647
图 33-2 通道扫描周期 648
图 33-3 开关时钟与调制器时钟 650
图 33-4 数字滤波器整体架构 651

图 33-5 16 阶均值滤波 652
图 33-6 信号与噪声阈值 652
图 33-7 SMPL_SIZE=2，扫描 4 个通道 653
图 33-8 TSI DMA 通道示意图 654
图 33-9 TSI 数据存储示意图 655
图 34-1 普通 GPIO 结构框图 670
图 34-2 真开漏 GPIO 结构框图 671
图 34-3 WKUPx 功能结构框图 676
图 34-4 引脚输入数字滤波 678
图 34-5 EXTI 信号输入示意图 678
图 37-1 CORTEX-M0+调试系统示意图 712

1 产品综述

1.1 概述

FM33FR0是基于ARM Cortex-M0的低功耗MCU，支持最大384KB程序flash、8KB数据flash和最大32KB RAM；集成12bit SAR-ADC，最大14通道按键触摸控制器，LCD/LED段码驱动，比较器等丰富外设；具备超宽工作电压范围和优异的低功耗性能。

主要特性如下：

- 宽电压范围：1.65~5.5V
- 工作温度范围（T_A）：-40℃~+105℃
- 处理器内核
  - ARM Cortex-M0
  - 支持用户/特权模式
  - 支持中断向量表重定向（VTOR）
  - 最高主频48Mhz
  - SWD调试接口
  - 24bit Systick定时器
  - 单周期乘法器
- 低功耗技术平台
  - 典型运行功耗120uA/MHz@48Mhz
  - Sleep模式：3uA
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：1.3uA
- 存储器
  - 384/256/128KB Code Flash，擦写寿命：100,000次
  - 8KB Data Flash，擦写寿命：500,000次
  - Flash数据保存时间：10年@85℃
  - 用户代码保护
  - 32/16KB RAM空间
- 丰富的模拟外设
  - 高可靠、可配置BOR电路（支持8级可编程下电复位阈值）
  - 超低功耗PDR电路（支持4级可编程下电复位阈值）
  - 可编程电源监测模块（SVD）
  - 2x低功耗模拟比较器
  - 12bit 2Msps SAR-ADC，最多23个外部输入通道

- 内置基准电压产生电路
- 温度传感器，精度+/-5℃

- 电容触摸按键
  - 最大支持14个独立按键通道
  - Baseline调校
  - 低功耗
  - 防水功能
  - 硬件滤波
- 通用通信接口
  - UART*6
  - SPI*4，主从模式
  - I2C*4，主机模式
  - I2C_SMBUS*2，主从模式
  - 7通道外设DMA
  - 可编程CRC校验模块
- 定时资源
  - TAU定时器阵列，包含12个独立通道
  - 16bit 4通道(包含3对互补通道)高级定时器*1
  - 16bit基本定时器*1
  - 24-bit Systick*1
  - 16bit 2通道低功耗定时器*1
  - 看门狗定时器*2
  - 低功耗实时时钟日历（RTC），带有数字调校功能，调校精度+/-0.476ppm
- LCD/LED段码显示驱动电路
  - LCD最大支持4COM×44SEG / 6COM×42SEG / 8COM×40SEG
  - 1/3 bias、1/4bias
  - 片内电阻分压
  - 支持休眠显示
  - 最大8×32 LED驱动
- 安全算法
  - AES硬件运算单元，128/192/256-bit
  - AES支持ECB/CBC/CTR/GCM/GMAC模式
  - 真随机数发生器
- 时钟发生电路

- 片上可配置高速RC振荡器，可配置频率输出8/16/24/32MHz，8M出厂调校误差+/-0.5%，16/24/32MHz出厂调校误差+/-1%，8MHz全温区变化小于+/-2%
- 低功耗32768Hz晶体振荡器，带有停振检测电路
- 低功耗低速RC振荡器，32.768KHz，全温区+/-8%
- 高频晶体振荡器，4~24MHz
- PLL，最高输出48MHz

- 封装：LQFP100/LQFP80/LQFP64/LQFP44/LQFP32

1.2 芯片结构框图

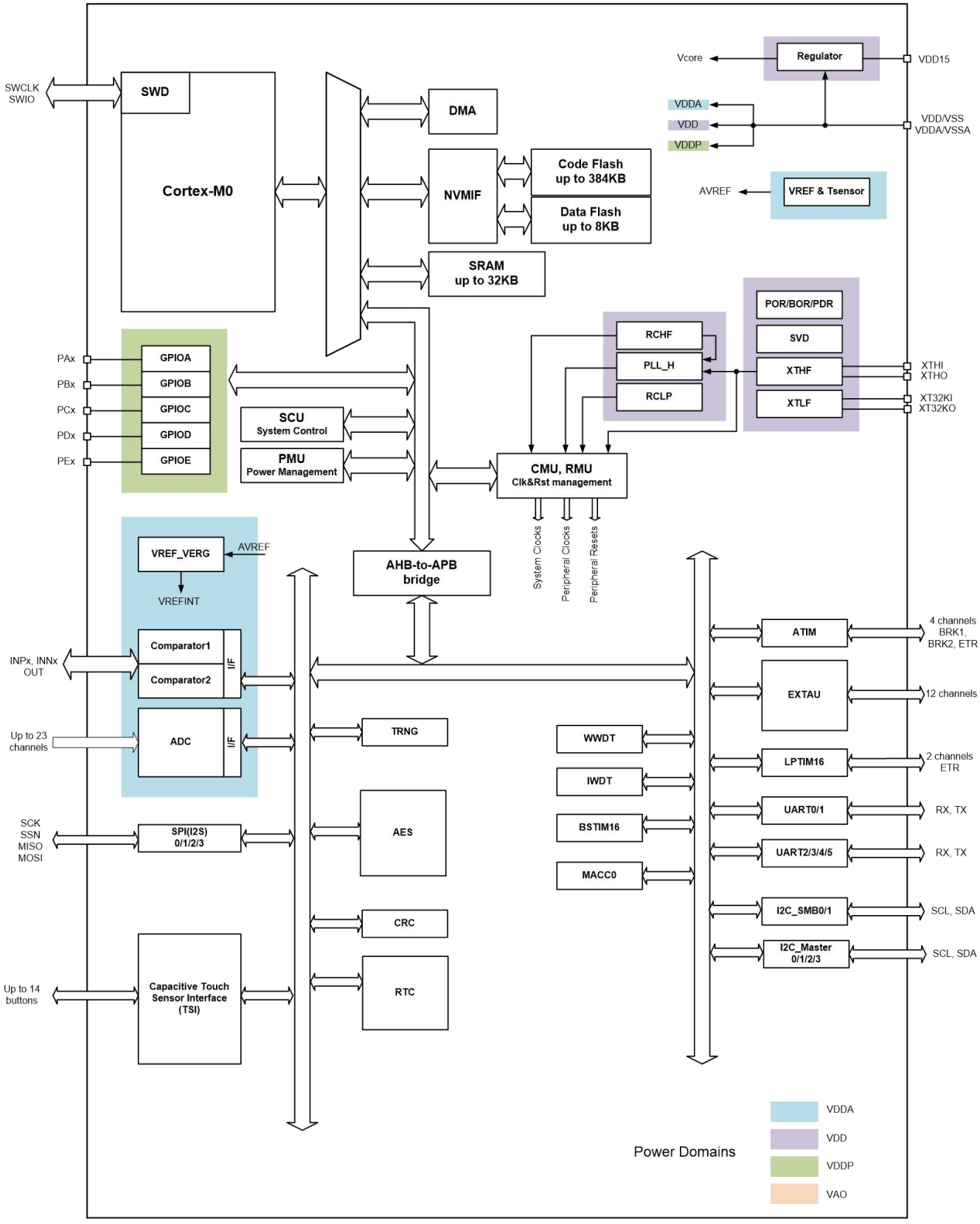


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Code Flash (Kbytes）	Data Flash (Kbytes)	RAM (Kbytes)	封装
FM33FR0510	384	8	32	LQFP100
FM33FR0410	256	8	32	LQFP100
FM33FR058	384	8	32	LQFP80
FM33FR048	256	8	32	LQFP80
FM33FR028	128	8	16	LQFP80
FM33FR056	384	8	32	LQFP64
FM33FR046	256	8	32	LQFP64
FM33FR026	128	8	16	LQFP64
FM33FR054	384	8	32	LQFP44
FM33FR044	256	8	32	LQFP44
FM33FR024	128	8	16	LQFP44
FM33FR014	64	8	16	LQFP44
FM33FR023	128	8	16	LQFP32
FM33FR013	64	8	16	LQFP32

表 1-1FM33FR0xx 型号列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP100 封装图（FM33FR0x10）

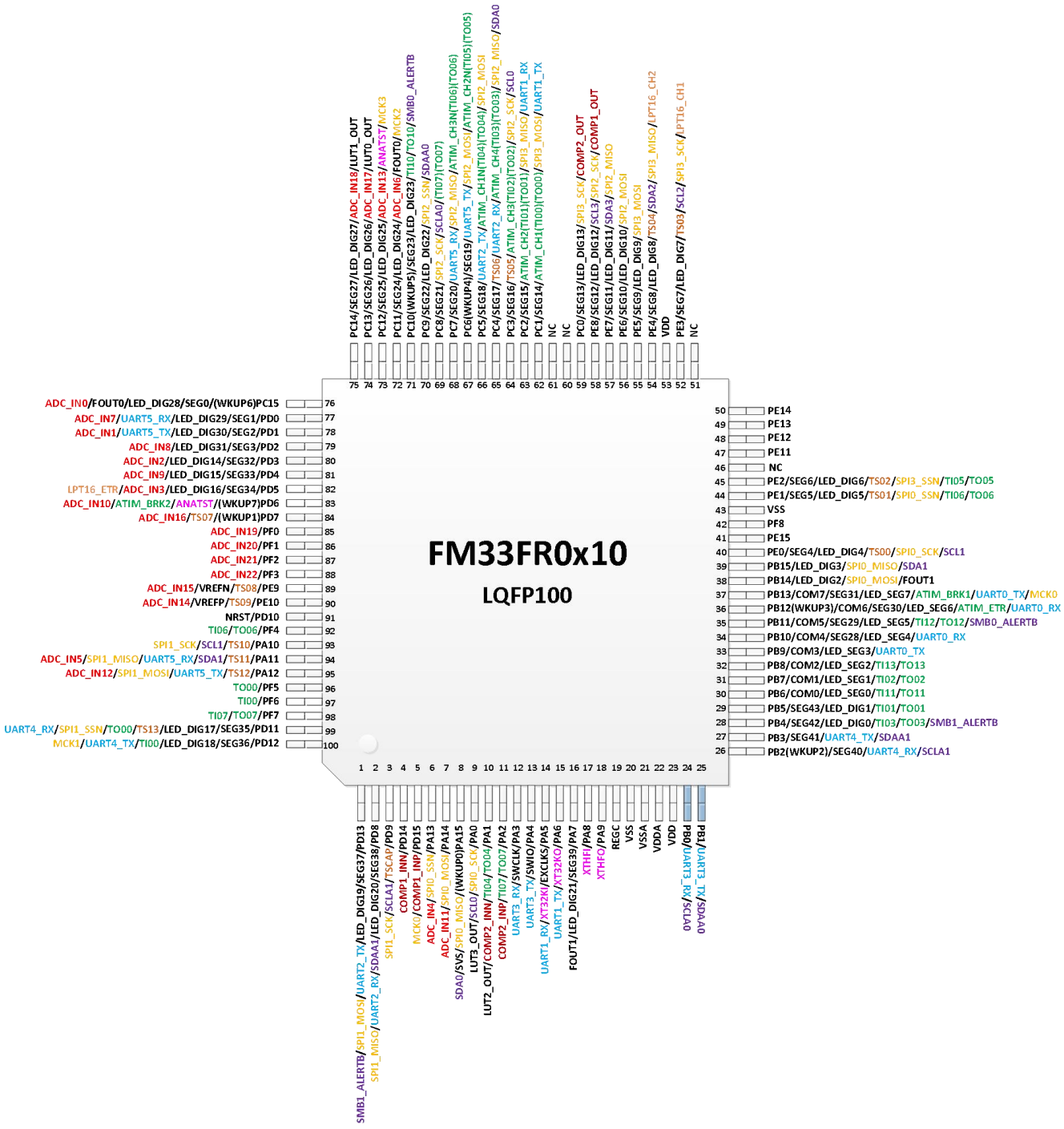


图 2-1 FM33FR0x10 LQFP100 封装图

注：蓝色底纹为真开漏引脚，最大耐压 6.5V

2.1.2 LQFP80 封装图（FM33FR0x8）

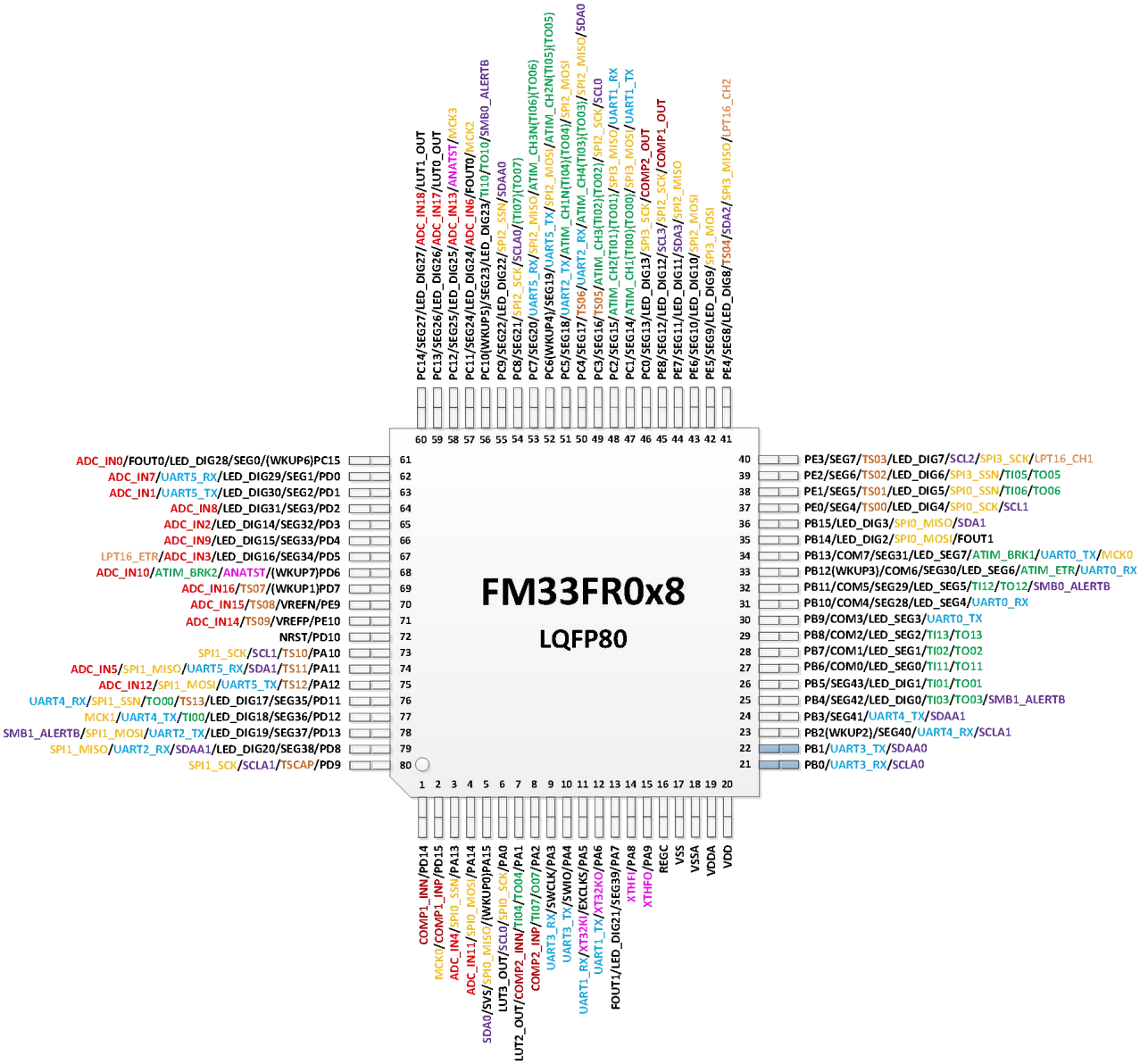


图 2-2 FM33FR0x8 LQFP80 封装图

注：蓝色底纹为真开漏引脚，最大耐压 6.5V

2.1.3 LQFP64 封装图（FM33FR0x6, 0.5mm pitch）

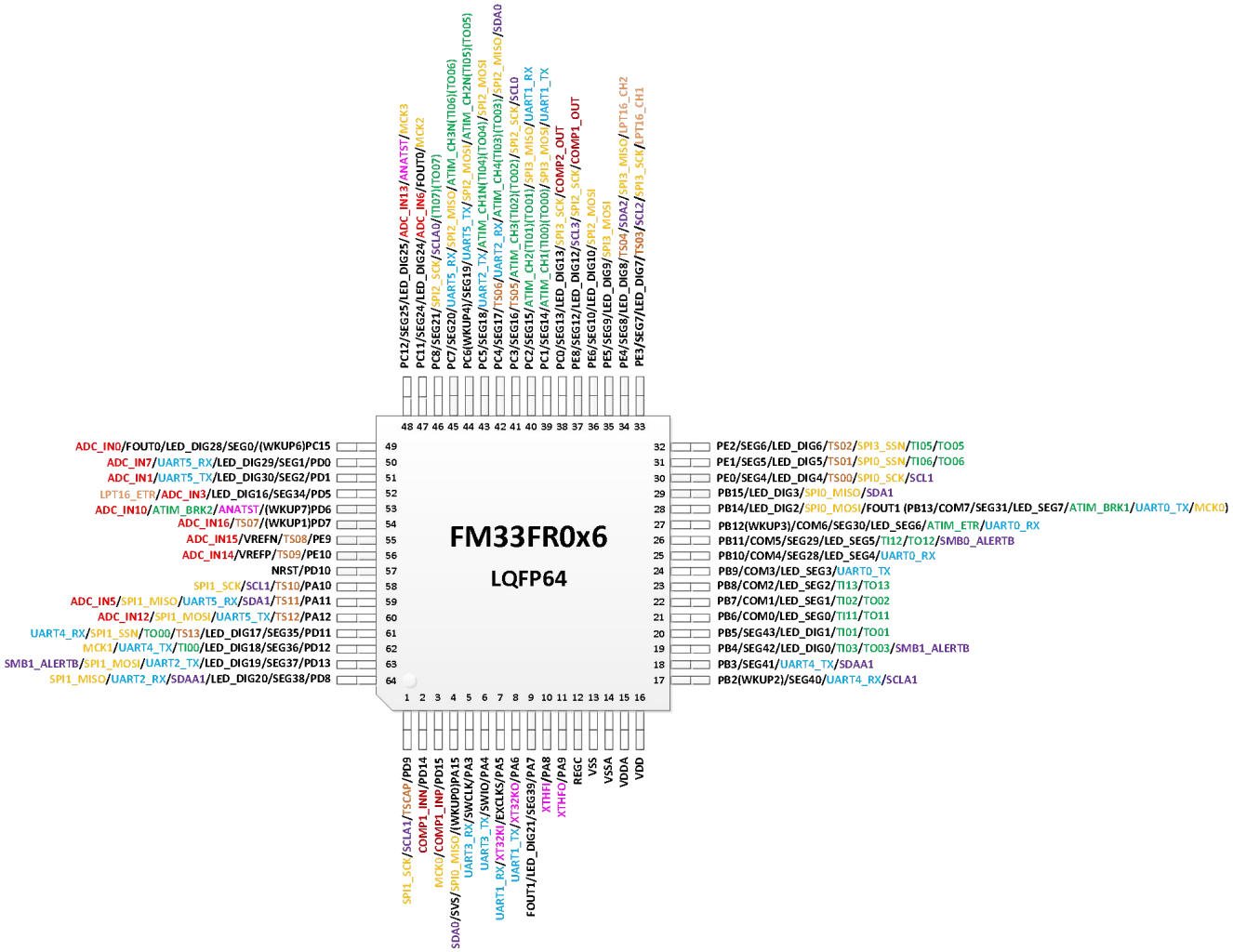


图 2-3 FM33FR0x6 LQFP64 封装图

2.1.4 LQFP64 封装图（FM33FR0x6P, 0.65mm pitch）

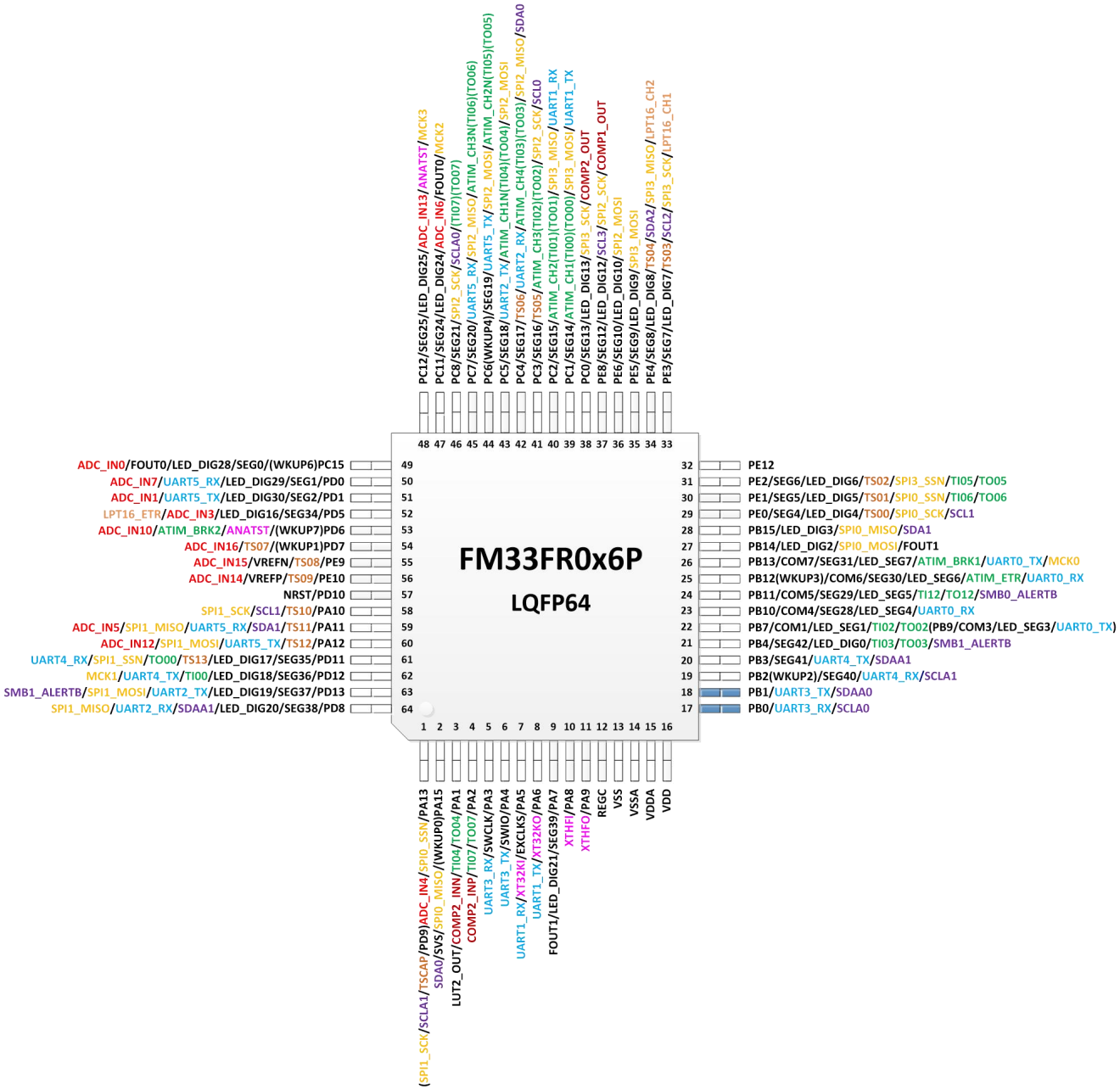


图 2-4 FM33FR0x6P LQFP64 封装图

注：蓝色底纹为真开漏引脚，最大耐压 6.5V

2.1.5 LQFP44 封装图（FM33FR0x4）

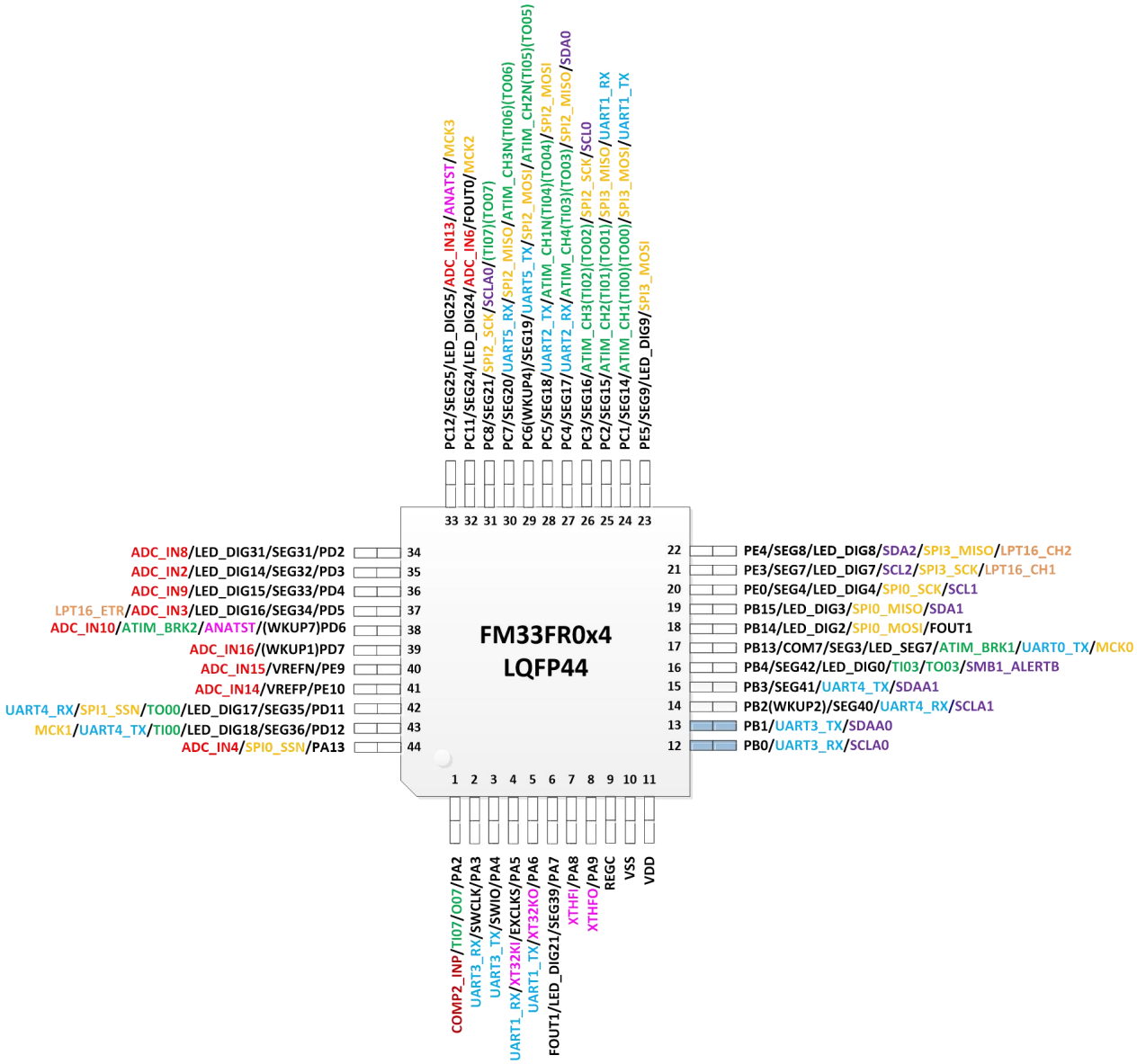


图 2-5 FM33FR0x4 LQFP44 封装图

注：蓝色底纹为真开漏引脚，最大耐压 6.5V

2.1.6 LQFP32 封装图（FM33FR0x3）

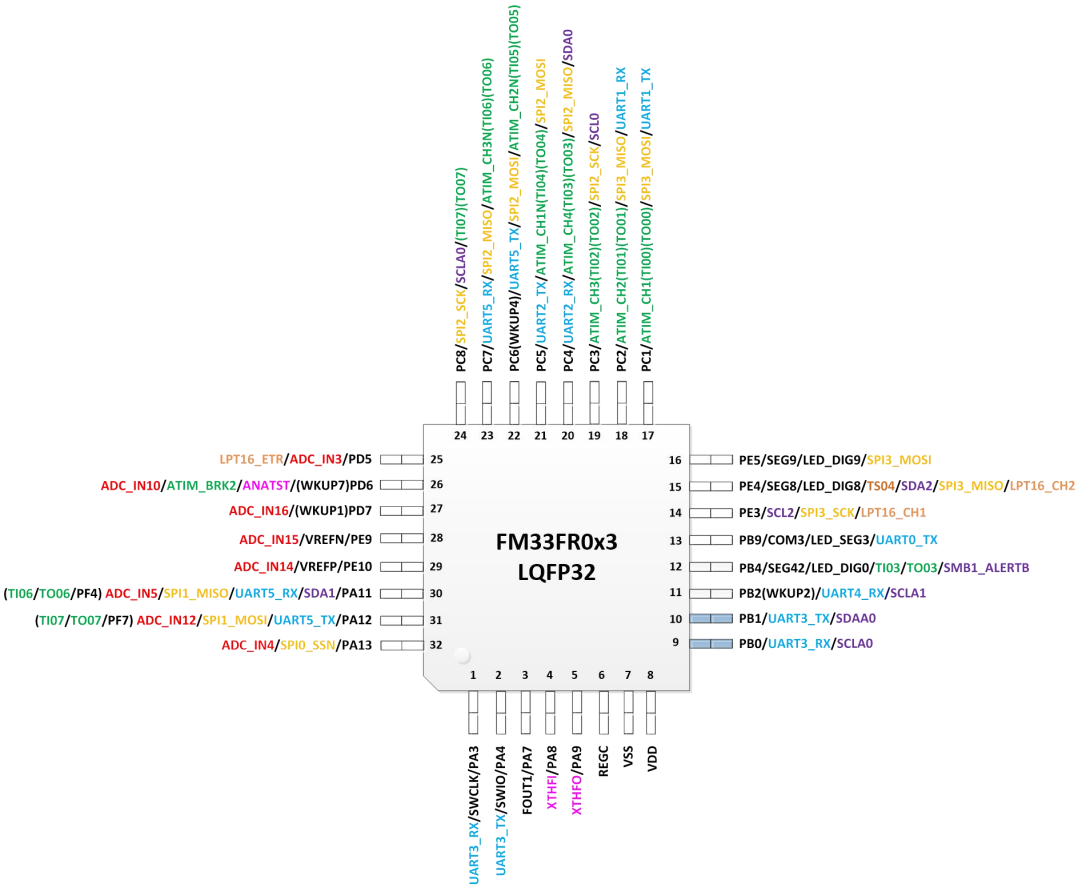


图 2-6 FM33FR0x3 LQFP32 封装图

注：蓝色底纹为真开漏引脚，最大耐压 6.5V

2.1.7 PA 引脚组数字外设功能

GPIO	PADFS			
	00	01	10	11
PA0	SPI0_SCK	SCL0	LUT3_OUT	-
PA1	TO04	TI04	LUT2_OUT	-
PA2	TO07	TI07	-	-
PA3	SWCLK	UART3_RX	-	-
PA4	SWIO	UART3_TX	-	-
PA5	UART1_RX	-	-	-
PA6	UART1_TX	-	-	-
PA7	LED_DIG21	FOUT1	-	-
PA8	-	-	-	-
PA9	-	-	-	-
PA10	SCL1	SPI1_SCK		
PA11	SDA1	SPI1_MISO	UART5_RX	
PA12	-	SPI1_MOSI	UART5_TX	
PA13	SPI0_SSN	-	-	-
PA14	SPI0_MOSI	-	-	-
PA15	SPI0_MISO	SDA0	-	-

2.1.8 PB 引脚组数字外设功能

GPIO	PBDFS			
	00	01	10	11
PB0	UART3_RX	SCLA0		
PB1	UART3_TX	SDAA0		
PB2	UART4_RX	SCLA1		
PB3	UART4_TX	SDAA1		
PB4	LED_DIG0	TI03	TO03	SMB1_ALERTB
PB5	LED_DIG1	TI01	TO01	
PB6	LED_SEG0	TI11	TO11	
PB7	LED_SEG1	TI02	TO02	
PB8	LED_SEG2	TI13	TO13	
PB9	LED_SEG3	UART0_TX		
PB10	LED_SEG4	UART0_RX		
PB11	LED_SEG5	TI12	TO12	SMB0_ALERTB
PB12	LED_SEG6	ATIM_ETR	UART0_RX	
PB13	LED_SEG7	ATIM_BRK1	UART0_TX	MCK0
PB14	LED_DIG2	SPI0_MOSI	FOUT1	
PB15	LED_DIG3	SPI0_MISO	SDA1	

2.1.9 PC 引脚组数字外设功能

GPIO	PCDFS			
	00	01	10	11
PC0	LED_DIG13	SPI3_SCK	COMP2_OUT	
PC1	ATIM_CH1	SPI3_MOSI	UART1_TX	

GPIO	PCDFS			
	00	01	10	11
	(TI00)(TO00)			
PC2	ATIM_CH2 (TI01)(TO01)	SPI3_MISO	UART1_RX	
PC3	ATIM_CH3 (TI02)(TO02)	SPI2_SCK	SCL0	
PC4	UART2_RX	ATIM_CH4 (TI03)(TO03)	SPI2_MISO	SDA0
PC5	UART2_TX	ATIM_CH1N (TI04)(TO04)	SPI2_MOSI	
PC6	UART5_TX	ATIM_CH2N (TI05)(TO05)	SPI2_MOSI	
PC7	UART5_RX	ATIM_CH3N (TI06)(TO06)	SPI2_MISO	
PC8	(TI07)(TO07)	SCLA0	SPI2_SCK	
PC9	LED_DIG22	SDAA0	SPI2_SSN	
PC10	LED_DIG23	TI10	TO10	SMB0_ALERTB
PC11	LED_DIG24	FOUT0	MCK2	
PC12	LED_DIG25	MCK3		
PC13	LED_DIG26	LUT0_OUT		
PC14	LED_DIG27	LUT1_OUT		
PC15	LED_DIG28	FOUT0		

注：与 ATIM 通道复用的 TAU 通道通过 GPIO 模块的 TMUX 寄存器控制。

2.1.10 PD 引脚组数字外设功能

GPIO	PDDFS			
	00	01	10	11
PD0	LED_DIG29	UART5_RX		
PD1	LED_DIG30	UART5_TX		
PD2	LED_DIG31			
PD3	LED_DIG14			
PD4	LED_DIG15			
PD5	LED_DIG16	LPT16_ETR		
PD6	ATIM_BRK2			
PD7				
PD8	LED_DIG20	SDAA1	UART2_RX	SPI1_MISO
PD9		SCLA1		SPI1_SCK
PD10			NRST	
PD11	LED_DIG17	TO00	SPI1_SSN	UART4_RX
PD12	LED_DIG18	TI00	MCK1	UART4_TX
PD13	LED_DIG19	SPI1_MOSI	UART2_TX	SMB1_ALERTB
PD14				
PD15		MCK0		

2.1.11 PE 引脚组数字外设功能

GPIO	PEDFS			
	00	01	10	11
PE0	LED_DIG4	SPI0_SCK	SCL1	
PE1	LED_DIG5	SPI0_SSN	TI06	TO06
PE2	LED_DIG6	SPI3_SSN	TI05	TO05
PE3	LED_DIG7	SPI3_SCK	SCL2	LPT16_CH1
PE4	LED_DIG8	SPI3_MISO	SDA2	LPT16_CH2
PE5	LED_DIG9	SPI3_MOSI		
PE6	LED_DIG10	SPI2_MOSI		
PE7	LED_DIG11	SPI2_MISO	SDA3	
PE8	LED_DIG12	SPI2_SCK	SCL3	COMP1_OUT
PE9				
PE10				
PE11				
PE12				
PE13				
PE14				
PE15				

2.1.12 PF 引脚组数字外设功能

GPIO	PFDFS			
	00	01	10	11
PF0				
PF1				
PF2				
PF3				
PF4	TO06	TI06		
PF5	TO00			
PF6	TI00			
PF7	TO07	TI07		
PF8				

2.1.13 封装尺寸图

2.1.13.1 LQFP100

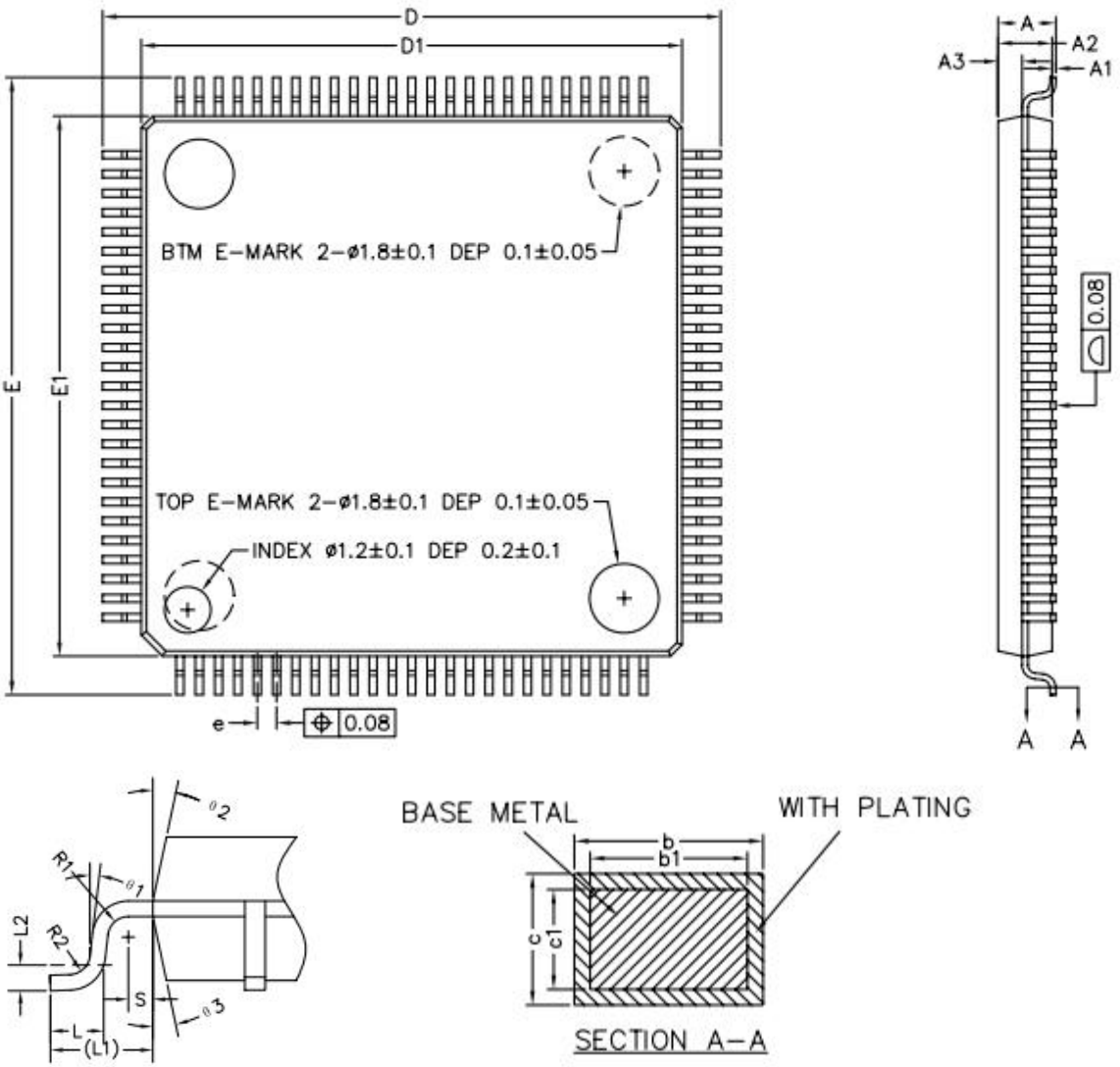


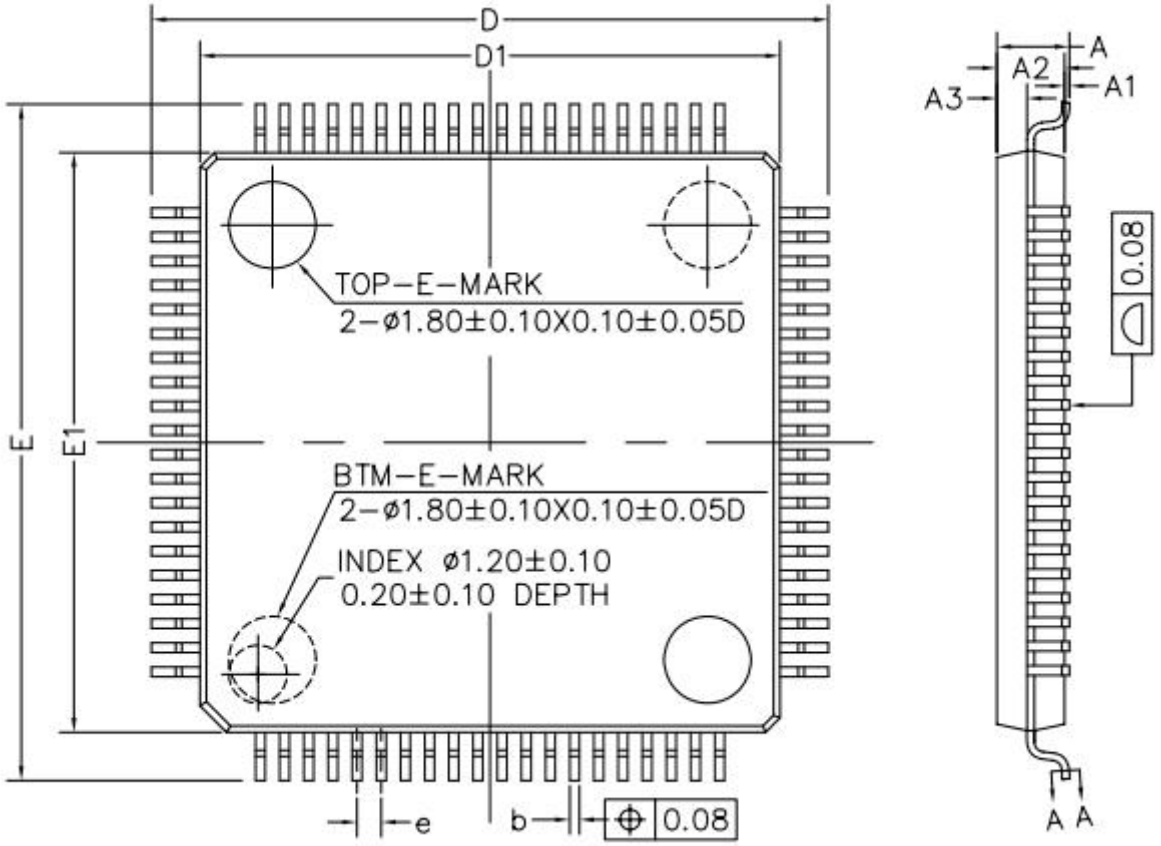
图 2-7 LQFP100 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20

Symbol	MIN	NOM	MAX
E1	13.90	14.00	14.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026BDD.

2.1.13.2 LQFP80



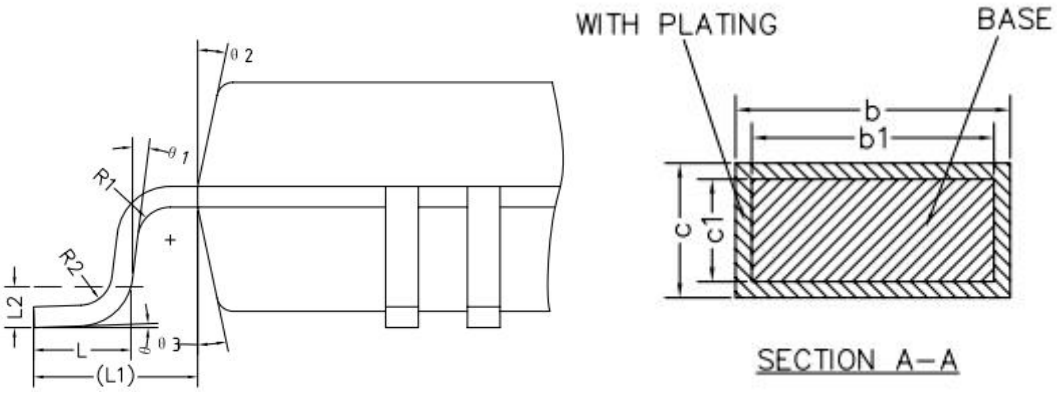


图 2-8 LQFP80 封装尺寸图

Symbol	MIN	NOM	MA
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220 WMMD-4.

2.1.13.3 LQFP64 (0.5mm pitch)

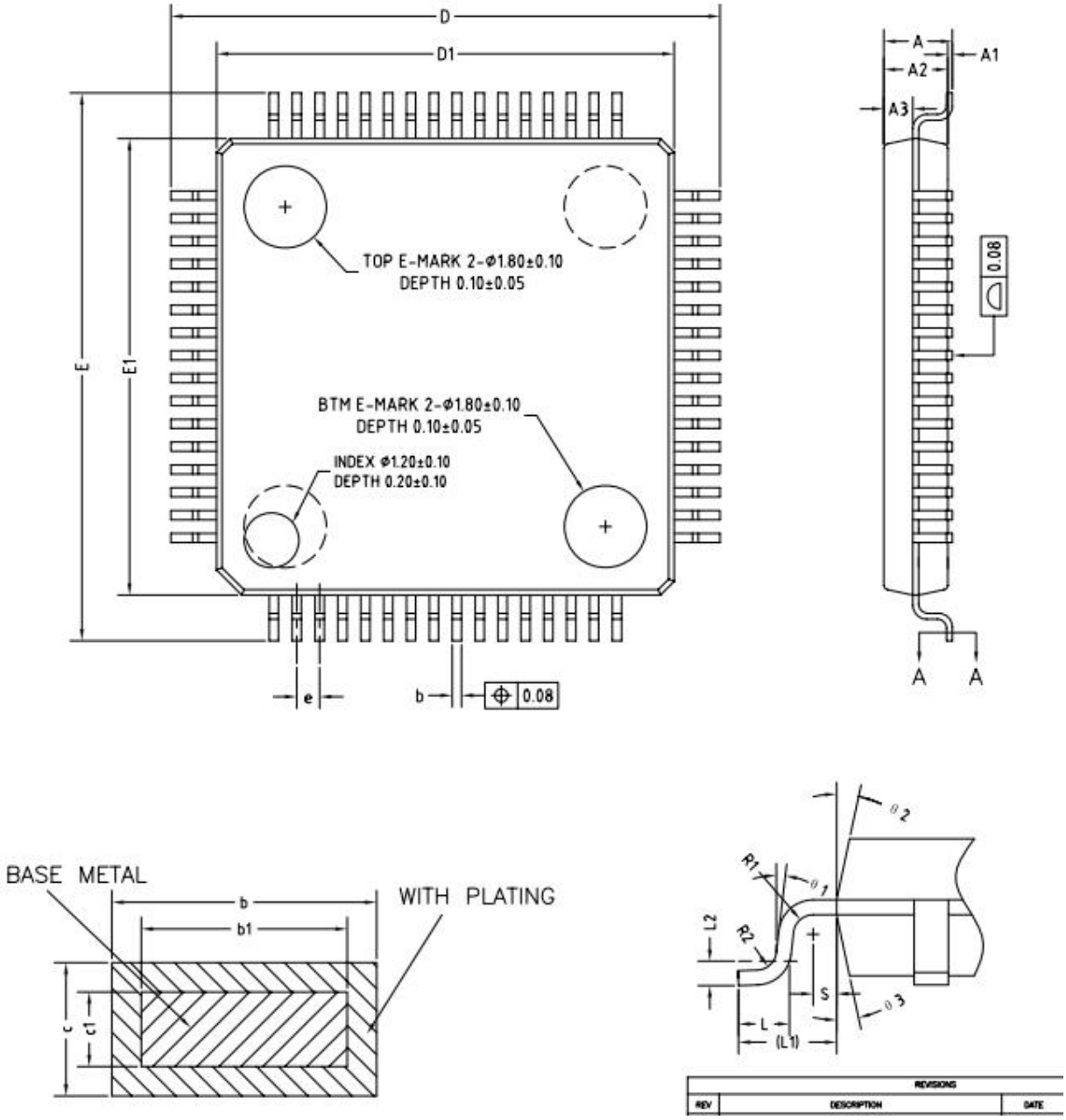


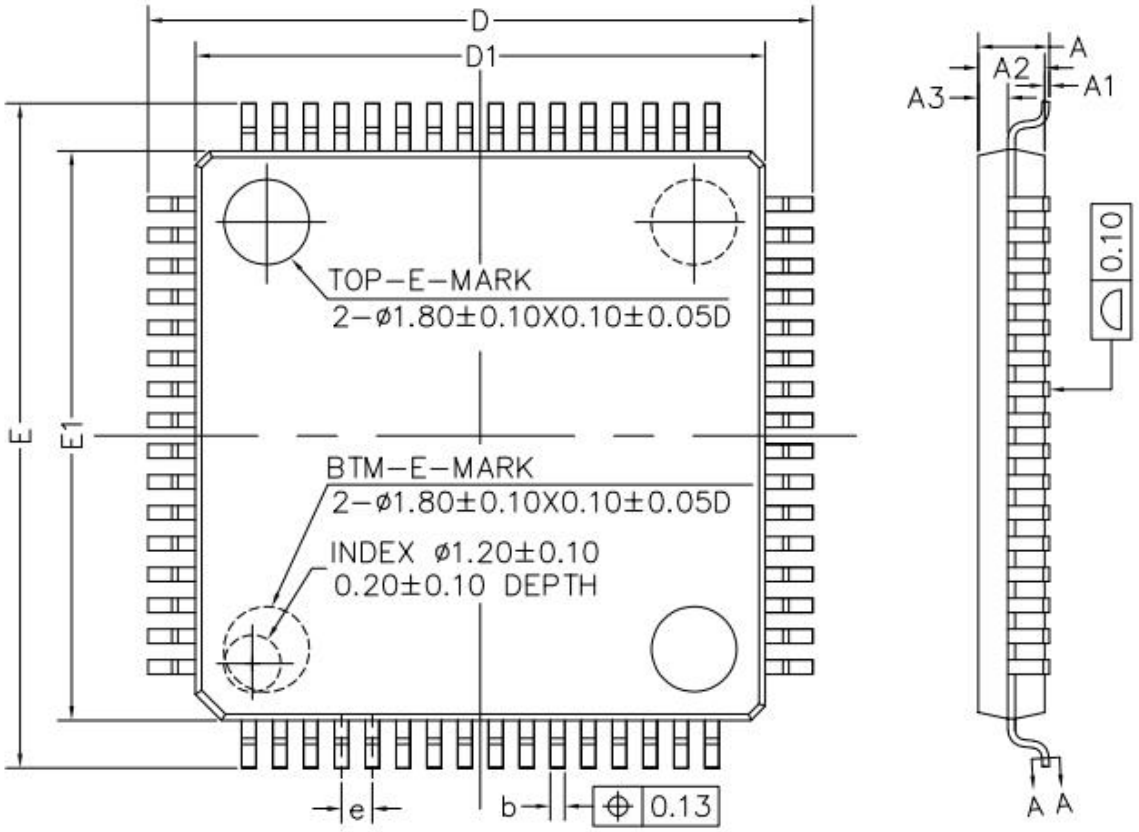
图 2-9 LQFP64 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20

Symbol	MIN	NOM	MAX
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220 WMMD-4.

2.1.13.4 LQFP64 (0.65mm pitch)



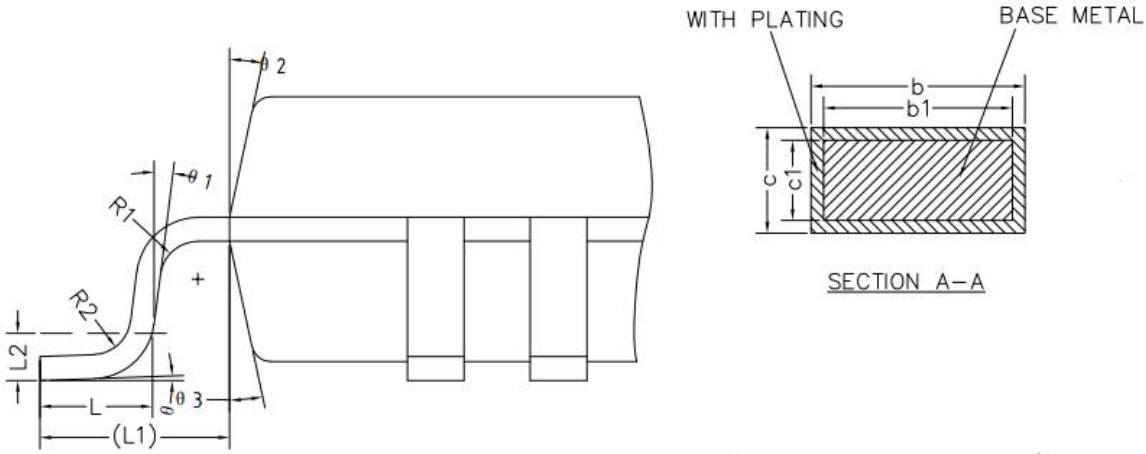


图 2-10 LQFP64 封装尺寸图(0.65mm pitch)

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.26	–	0.38
b1	0.25	0.30	0.33
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
e	0.55	0.65	0.75
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
θ	0°	3.5°	7°
θ 1	0°	–	–
θ 2	11°	12°	13°
θ 3	11°	12°	13°

2.1.13.5 **LQFP44**

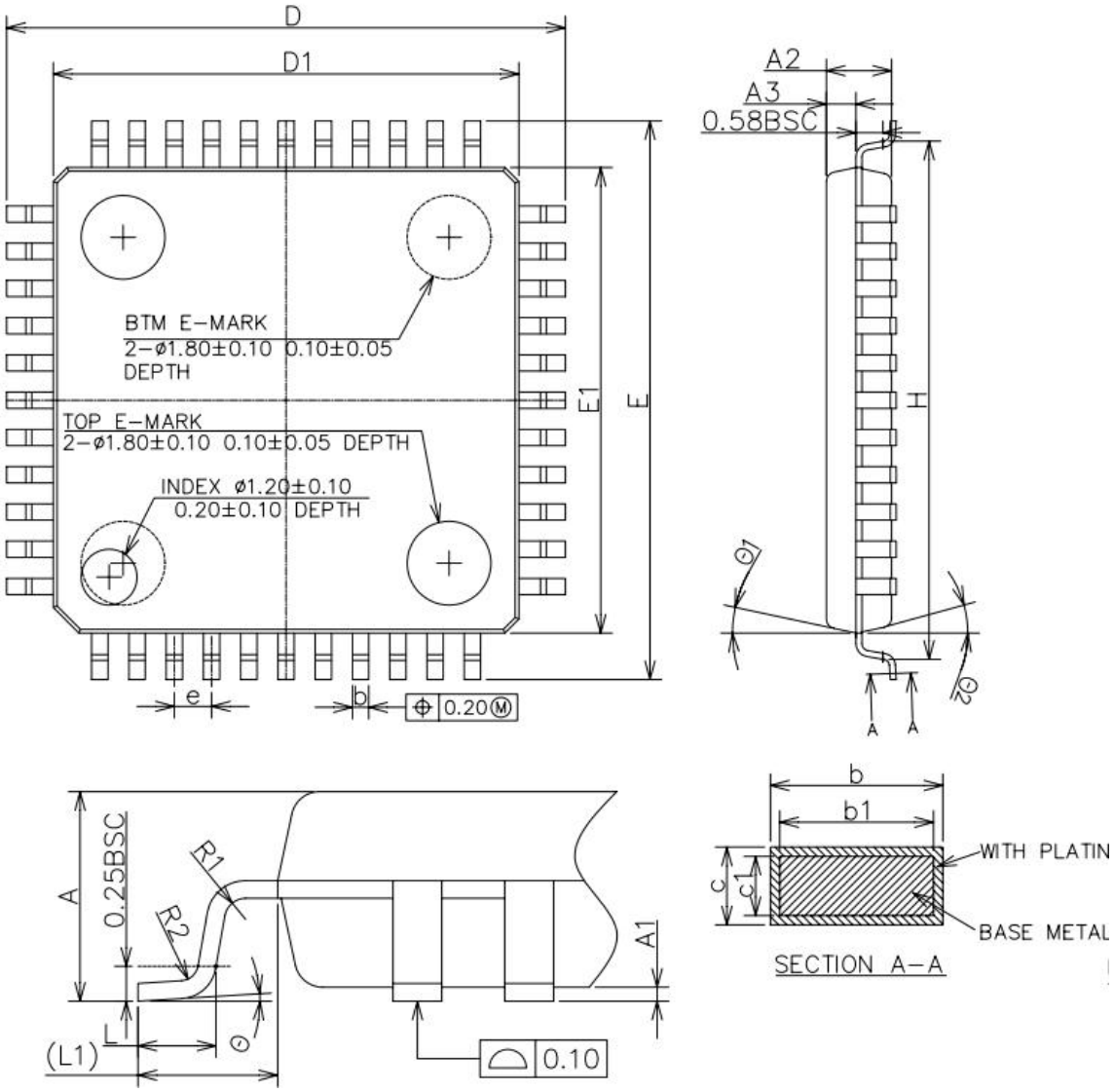


图 2-11 LQFP44 封装尺寸图

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.33	–	0.42
b1	0.32	0.35	0.38
c	0.13	–	0.18
c1	0.117	0.127	0.137
D	11.95	12.00	12.05
D1	9.90	10.00	10.10
E	11.95	12.00	12.05
E1	9.90	10.00	10.10
e	0.70	0.80	0.90
H	11.09	11.13	11.17
L	0.53	–	0.70
L1	1.00REF		
R1	0.15REF		
R2	0.13REF		
Θ	0°	3.5°	7°
Θ1	11°	12°	13°
Θ2	11°	12°	13°

NOTES:
1. ALL DIMENSIONS REFER TO JEDEC STANDARD MS026 BCB
   DO NOT INCLUDE MOLD FLASH,GATE BURR OR PROTRUSION.

2.1.13.6 **LQFP32**

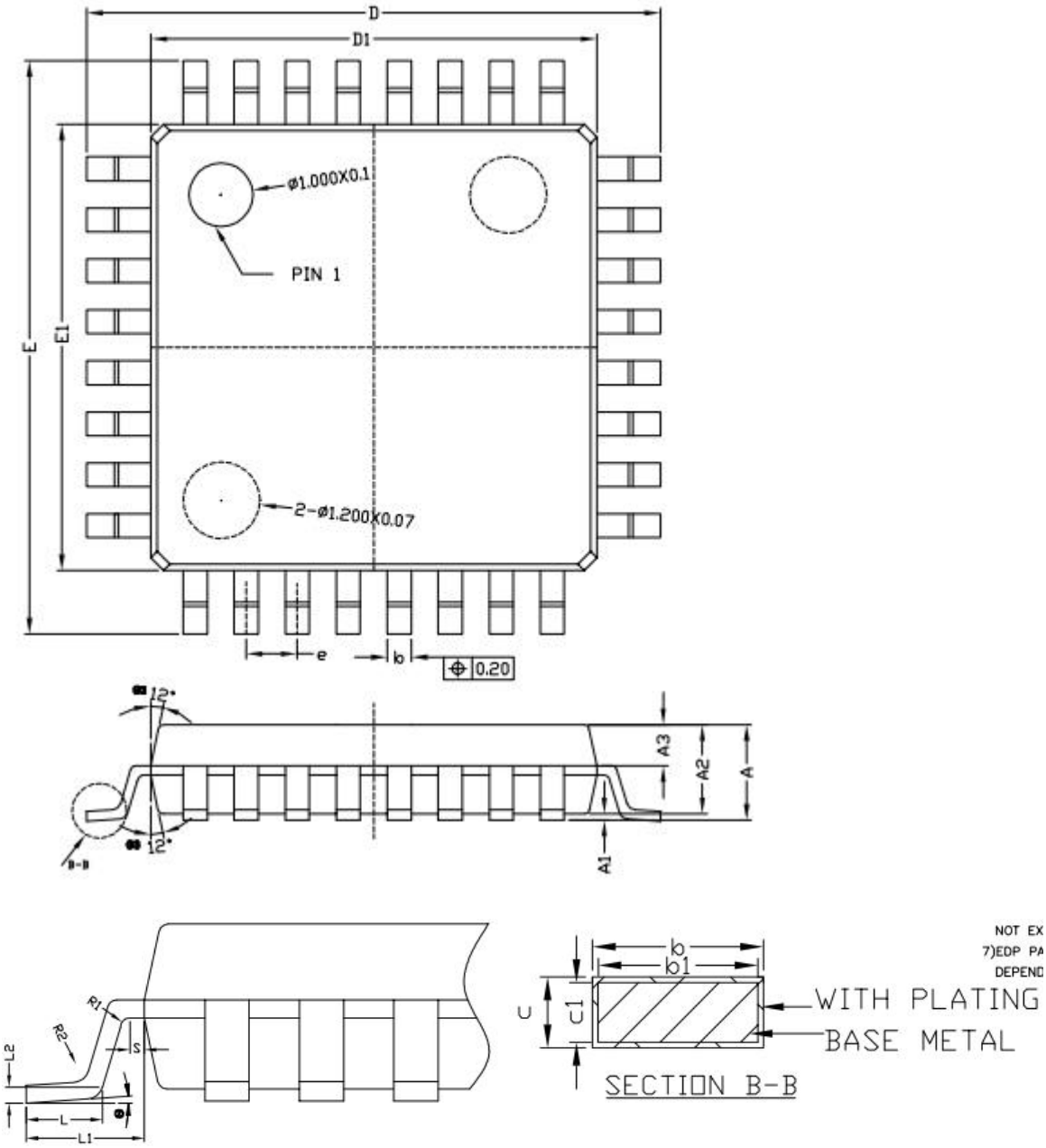


图 2-12 LQFP32 封装尺寸图

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	0.10	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32	–	0.43
b1	0.31	0.35	0.39
c	0.13	–	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.80BSC		
L	0.45	–	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

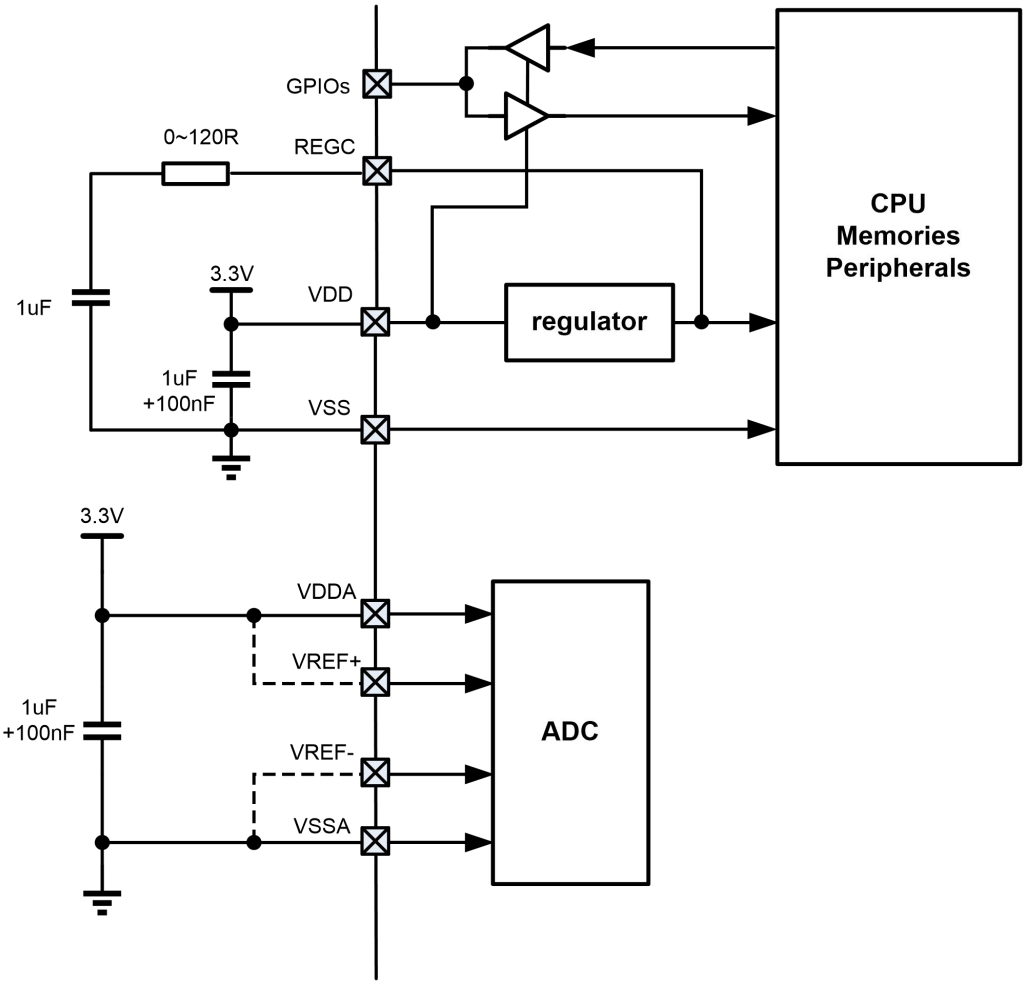
3 电参数

3.1 参数说明

电参数章节中所罗列的典型值是大量样本数据分布的中心值，常温下的最大值是由芯片量产测试所保证的。高低温下的电参数通常基于特征参数提取，由大量样本数据分布的平均值加减 3sigma 得到。

3.2 推荐电源方案

3.2.1 供电方案



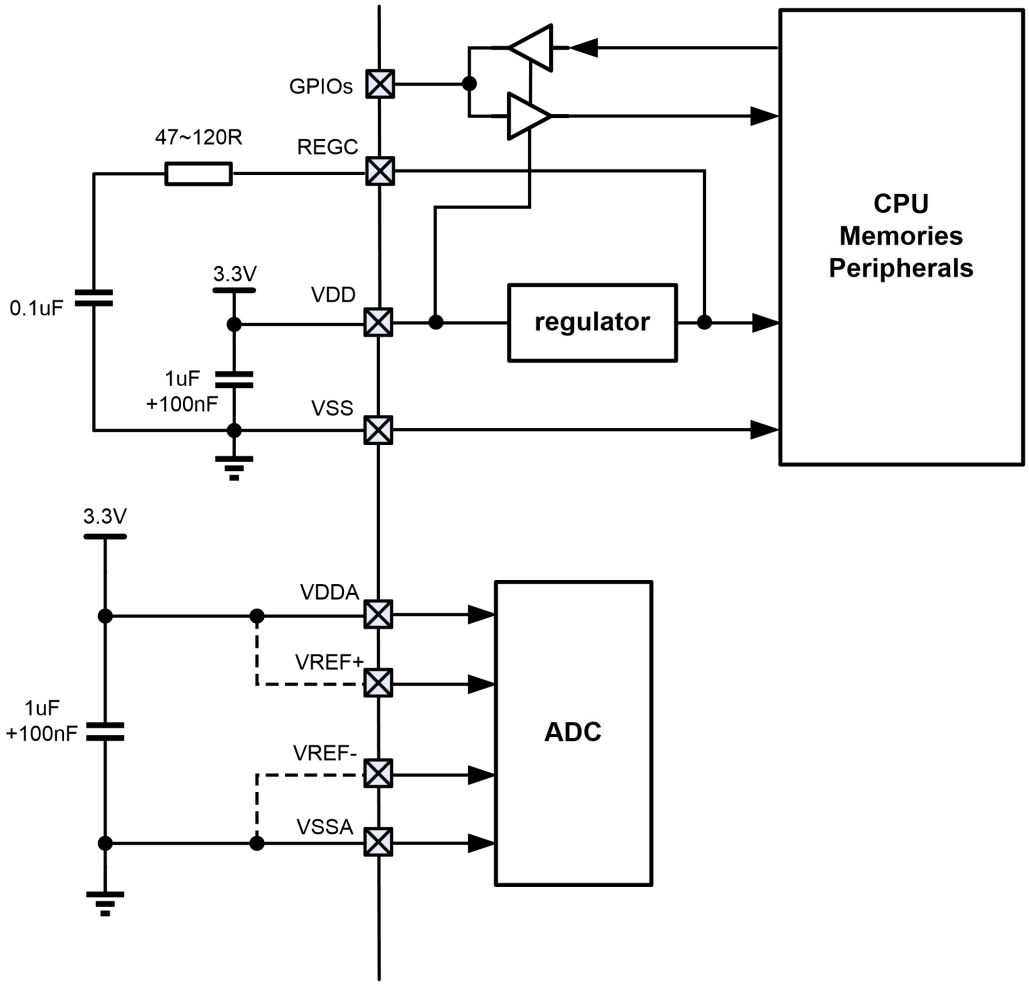


图 3-1 FM33FR0xx 供电方案

注意： 1 REGC 对地电容可以接 0.1uF 或 1uF

2 REGC 串接电阻可以增强抗干扰能力

(1) 电容为 1uF 时，REGC 串接电阻范围 0~120R，电阻的具体值可以根据干扰程度而定

(2) 电容为 0.1uF 时，没有休眠应用时 REGC 串接电阻范围 0~120R。有休眠应用时 REGC 串接电阻范围 47~120R。电阻的具体值可以根据干扰程度而定

3 电参数测试的供电方案 REGC 外接电阻为 0 欧姆对地电容为 1uF

3.3 极限参数

对芯片施加的电压、电流等超过极限参数表定义的最大范围时，可能导致芯片不可恢复的损坏；短时间超过极限参数范围则可能影响芯片的可靠性和工作寿命。

Symbol	Parameter		min	max	unit
V_{DD}-V_{SS}	电源电压 （包含 VDD、VDDA）		-0.3	6.5	V
V_{PIN}	管脚电压	All others	V_{SS}-0.3	V_{DD}+0.3	V
		PB0/1	V_{SS}-0.3	6.5	V
$\|\triangle V_{DD}\|$	VDD 和 VREFP 之间的压差		-		mV
$\|\triangle V_{SS}\|$	所有地引脚之间的压差		-	50	mV
T_A	工作环境温度		-40	105	℃
T_{STG}	存储温度		-55	150	℃
HBM	ESD HBM 模式 T_A=25℃ 测试标准符合 JEDEC JS-001			+/-4000	V
CDM	ESD CDM 模式 T_A=25℃ 测试标准符合 JEDEC JS-002			+/-1000	V
LU	IO Latchup -(0.5VDD) < VI < (1.5VDD) T_A=25℃ 测试标准符合 JESD78E			+/-200	mA
$\sum I_{VDD}$	向芯片 VDD 流入的最大电流 (source)			100	mA
$\sum I_{VSS}$	从芯片 VSS 流出的最大电流 (sink)			100	mA
$\sum I_{IO}$	所有 IO sink 的最大总和电流			100	mA
	所有 IO source 的最大总和电流			100	mA

表 3-1FM33FR0 极限参数

3.4 性能参数

3.4.1 典型工作条件

Symbol	Parameter	Conditions	min	max	unit
f_{HCLK}	AHB 时钟频率	-	0	48	MHz
f_{PCLK}	APB 时钟频率	-	0	48	
VDD	典型工作电压范围		1.65	5.5	V
VDDA	模拟电路工作电压范围		1.65	5.5	V
Vcore	内核 LDO 输出电压 Active 模式	REGC 管脚对地连接 1uF 稳压电容	1.45	1.65	V
T_J	结温		-40	125	°C

表 3-2FM33FR0 典型工作条件

3.4.2 功耗参数

芯片出厂时的功耗参数在环境温度下测试，高低温电流参数来自于特征参数提取。

测量功耗参数时，MCU 被配置为如下条件：

- 所有功能引脚被配置为 GPIO 模式，并且关闭输入和输出使能，避免引脚浮空漏电
- 除了特别声明的以外，所有外设被关闭，并停止工作时钟
- 常温下的最大功耗数据代表出厂时的测试上限标准
- 常温下的典型功耗数据代表大量样本分布的中心值
- 除非特别声明，所有功耗数据在 VDD=VREFP=5V 的条件下测试获得

3.4.2.1 Active 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=8MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	1.3	-	mA
			T_A=105℃	-	1.4	-	
		f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	2.45	-	
			T_A=105℃	-	2.6	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	3.6	-	
			T_A=105℃	-	3.78	-	
		f_{AHB}=48MHz (PLL) PLL on Flash 1 wait	T_A=25℃	-	6.6	-	
			T_A=105℃	-	6.57	-	

表 3-3ACTIVE 电流参数

注：上表参数基于特征参数提取，不包含在量产测试中

3.4.2.2 LP RUN 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{LPRUN}	LP RUN 模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=32768Hz (XTLF) PLL, RCHF off Flash 0 wait	T_A=25℃	-	25	-	uA
			T_A=85℃	-	32	-	
			T_A=105℃	-	70	-	

表 3-4LP RUN 电流参数

3.4.2.3 SLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
I_{sleep2}	Sleep 模式电流 2	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示关闭	T_A=25℃	-	3	-	uA
			T_A=85℃	-	9.6	-	
			T_A=105℃	-	40	-	

表 3-5 SLEEP 电流参数

3.4.2.4 DEEPSLEEP 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$I_{deepsleep1}$	DeepSleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持 LCD 显示关闭	T_A=25℃	-	1.3	-	uA
			T_A=85℃	-	7.2	-	
			T_A=105℃	-	35	-	

表 3-6 DEEPSLEEP 电流参数

3.4.3 复位和电源监控

芯片的复位和电源监控参数如下表。

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
t_{VDD}	电源上升速度			-		∞	us/V
	电源下降速度	PDR		200		∞	us/V
		BOR		600		∞	us/V
T_{reset_delay}	上电复位延迟时间				2		ms
T_{pdr_filter}	下电复位滤波时间				4		us
V_{POR}	上电复位电压			1.45	1.53	1.7	V
V_{BOR_fall}	下电复位电压	BORCFG==000			1.8		V
		BORCFG==001			2.0		
		BORCFG==010			2.2		
		BORCFG==011			2.4		
		BORCFG==100			2.7		
		BORCFG==101			3.0		
		BORCFG==110			3.6		
		BORCFG==111			4.0		
V_{BOR_rise}	下电复位释放电压	BORCFG==000			2.0		V
		BORCFG==001			2.2		
		BORCFG==010			2.4		
		BORCFG==011			2.6		
		BORCFG==100			2.9		
		BORCFG==101			3.2		
		BORCFG==110			3.8		
		BORCFG==111			4.2		
V_{PDR}	低功耗下电复位电压[1] T_A=-40~85C	PDRCFG==2'b10		1.36	1.45	1.52	V
		PDRCFG==2'b11		1.38	1.5	1.56	
		PDRCFG==2'b01		1.32	1.4	1.48	
		PDRCFG==2'b00		1.28	1.36	1.44	
I_{BOR}	BOR 功耗				1		uA
I_{PDR}	PDR 功耗				55		nA
V_{SVD}	电压监测阈值电平（比较基准 1.0V）	SVD[3:0]=0000	Fall		1.800		V
			Rise		1.900		
		SVD[3:0]=0001	Fall		2.014		V
			Rise		2.114		
		SVD[3:0]=0010	Fall		2.229		V
			Rise		2.329		
		SVD[3:0]=0011	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0100	Fall		2.657		V
			Rise		2.757		
		SVD[3:0]=0101	Fall		2.871		V
			Rise		2.971		
		SVD[3:0]=0110	Fall		3.086		V
			Rise		3.186		
		SVD[3:0]=0111	Fall		3.300		V
			Rise		3.400		

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
		SVD[3:0]=1000	Fall		3.514		V
			Rise		3.614		
		SVD[3:0]=1001	Fall		3.729		V
			Rise		3.829		
		SVD[3:0]=1010	Fall		3.943		V
			Rise		4.043		
		SVD[3:0]=1011	Fall		4.157		V
			Rise		4.257		
		SVD[3:0]=1100	Fall		4.371		V
			Rise		4.471		
		SVD[3:0]=1101	Fall		4.586		V
			Rise		4.686		
		SVD[3:0]=1110	Fall		4.800		V
			Rise		4.900		
		SVD[3:0]=1111	Fall		-		V
			Rise		-		

表 3-7 复位和电源监控参数

[1]基于特征参数提取

3.4.4 内部基准电压

芯片内建高精度基准电压源，为 ADC 和片外电路提供高精度、高稳定性的参考电压。

芯片出厂时，复旦微电子会在特定的电源电压和温度下，使用片内 ADC 采样基准源输出，并将转换结果保存在芯片的 Flash 中，用户应用中可以将这个转换值作为参考基准使用。

符号	参数说明	
REF_CAL	ADC 对 AVREF 输出的转换值 测试条件： T_A=30±5℃ VREFP=3V	请联系复旦微电子获取驱动库函数

内部基准源主要参数：

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
AV_{REF}	基准源输出电压	-40℃≤T_A≤105℃	0.985	1.0	1.01	V
T_{setup}	内部基准源建立时间	-	-	5		us
V_{VREF_MEAS}	出厂时测量转换 AVREF 的 VREFP 电压	-	2.997	3	3.003	V
T_{coeff}	内部基准源温度系数	-40℃≤T_A≤105℃		25	100	ppm/℃
V_{coeff}	内部基准源电压系数	1.8≤VDDA≤5V		1	3	mV/V
T_{S_VREF}	ADC 测量 AVREF 时的采样时间	AVREF buffer 使能	10			us
T_{ADC_BUF}	驱动 ADC 输入的		5			us

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
	VREF Buffer 的建立时间						
I_{REF}	基准源工作电流	T_A = 25℃ VDDA = 3V	PTAT_EN=0		2		uA
			PTAT_EN=1		3		

表 3-8 内部基准电压参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
VREFP	基准源整流输出电压	VRS=00		1.5		V
		VRS=01		2.0		
		VRS=10		2.5		
		VRS=11		3.0		
I_{VREFP}	基准源整流功耗	VDDA=3.3V	-	1	-	mA
		VDDA=5.5V	-	1.4	-	mA
VREFP accu	基准源整流输出精度	T_A=25° C，校准后	-0.1		0.1	%
T_{coeff}	VREFP 温度系数	-40℃≤T_A≤85℃				ppm /℃

表 3-9 内部基准整流参数

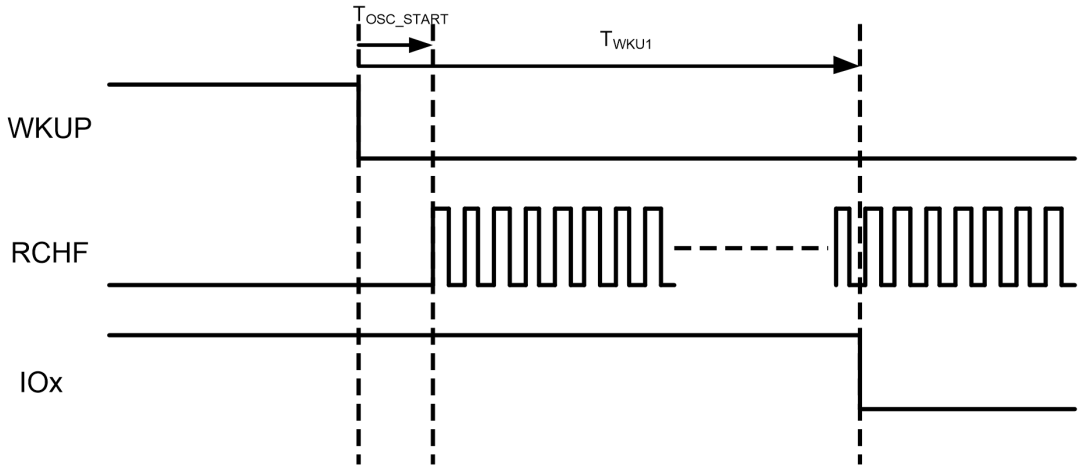
3.4.5 低功耗模式唤醒时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
T_{WKU1}	Sleep/DeepSleep 唤醒时间[1]	使用 WKUP 引脚唤醒，PRIMASK=1 禁止中断；CPU 唤醒后执行程序翻转某个 IO 输出，测量 WKUP 信号边沿到 IO 输出翻转之间的时间 F_{SYSCLK}=8Mhz	-	6.5	-	us
T_{WKU2}	LPRUN 模式唤醒时间		-	0	-	us

表 3-10 唤醒时间参数

[1] 基于特征参数提取

典型唤醒事件波形图，仅供设计参考



上图中 T_{OSC_START} 表示唤醒事件到来后 RCHF 环振起振时间，典型值小于 3us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 6.5us。

如果没有通过 PRIMASK 屏蔽中断，则唤醒事件将使 CPU 进入中断服务程序。CPU 进入中断服务程序的过程将额外引入延迟时间。

注意：以上时间评估使用 RCHF 8Mhz 为唤醒后的工作时钟，如果唤醒后选择 16Mhz 或 24Mhz 频率，则唤醒时间相应缩短。

3.4.6 外部时钟源特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{XTLF}	XTLF 振荡频率	外接 32768Hz 晶体		32768		Hz
T_{start}	XTLF 起振时间	外接 32768Hz 晶体 C_{load}=12pF XTLFIPW==3'b000		1	3	s

表 3-11 低频晶体振荡器参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率	VDD=3.3V		4	-	24	MHz
R_{fb}	反馈电阻	-		-	200	-	KΩ
gm	跨导增益	VDD=5V	HF_CFG =00000	-	1.75	-	mA/V
			HF_CFG =00001	-	3.5	-	
		VDD=3V	HF_CFG =00000	-	1.2	-	
			HF_CFG =00001	-	2.4	-	
			HF_CFG =00010	-	3.48	-	
		VDD=1.6V	HF_CFG =00000	-	0.32	-	
			HF_CFG =00100	-	1.59	-	
			HF_CFG =00111	-	2.55	-	
VDD_{rise}	XTHF 最低工作电压 HF_CFG=00000	4MHz		1.08	-	-	V
		8MHz,		1.6	-	-	
		24MHz		1.85	-	-	
	XTHF 最低工作电压 HF_CFG=11111	4MHz		0.9	-	-	
		8MHz,		1.0	-	-	
		24MHz		1.1	-	-	
IDD	XTHF 工作电流 HF_CFG=00000	4MHz		-	150	-	uA
		8MHz		-	180	-	
		16MHz			250		
		24MHz			360		
	XTHF 工作电流 HF_CFG=1000	4MHz		-	1000	-	
		8MHz		-	1050	-	
		16MHz			1200		
		24MHz			1450		
T_{start1}	XTHF 8M 起振时间	VDD=3.3V HF_CFG=00000		-	2.3	-	ms
		VDD=3.3V, HF_CFG=10000		-	0.5	-	ms
T_{start2}	XTHF 16M 起振时间	VDD=3.3V, HF_CFG=00000		-	2.1	-	ms
		VDD=3.3V,		-	0.2	-	ms

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
		HF_CFG=10000				
C_L	负载电容	-	5	-	25	pF
C_{IN}	片内集成负载电容	-	0	-	4	pF

表 3-12 高频晶体振荡器参数

注：

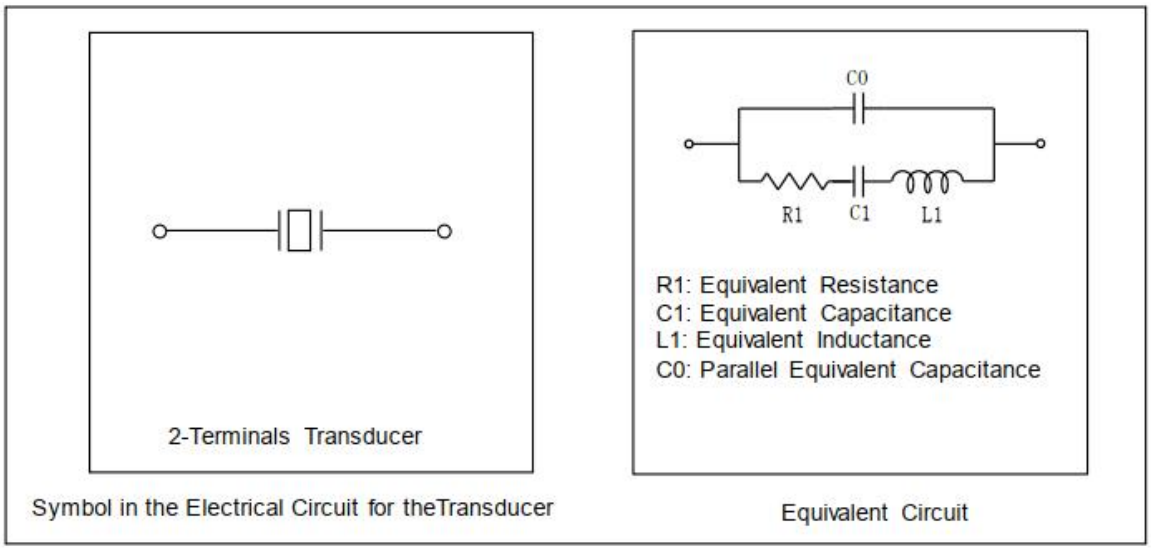
[1]以上指标基于特征参数提取

[2]推荐使用8~16MHz晶振或陶振，搭配推荐的振荡强度配置，以降低时钟振荡器的功耗和噪声辐射

XTHF 典型 gm 参数（仅供设计参考）：

HF_CFG	Gm(mA/V)		
	VDD=5V	VDD=3V	VDD=1.6V
00000	1.75	1.2	0.32
00001	3.5	2.4	0.64
00010	4.91	3.48	0.95
00011	6.67	4.68	1.28
00100	8.07	5.77	1.59
00101	9.83	6.96	1.91
00110	11.2	8.05	2.23
00111	13	9.25	2.55
01000	14.4	10.3	2.86
01001	16.2	11.5	3.18
01010	17.6	12.6	3.5

晶振或陶振的等效电路模型参见下图



振荡器起振的最小环路增益可以通过下式计算：

$$g_{mcrit} = 4 \times ESR \times (2\pi F)^2 \times (C_0 + C_L)^2$$

其中 C_L 是晶振要求的负载电容或陶振内建负载电容，ESR 可以通过下式计算：

$$ESR = R_1 \times \left(1 + \frac{C_0}{C_L}\right)^2$$

为了保证足够安全的振荡余量，通常要求 g_m 大于 5 倍 g_{mcrit}，据此可以根据晶振或陶振手册，选择合适的 XTHF 振荡强度配置。

3.4.7 内部时钟源特性

内部高频 RC 振荡器

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
f_{RCHF}[1]	RCHF 振荡频率	VDD=1.65~5.5V T_A=25℃	FSEL==2'b00	7.96	8	8.04	MHz
			FSEL==2'b01	15.84	16	16.16	
			FSEL==2'b10	23.76	24	24.24	
			FSEL==2'b11	31.68	32	32.32	
ACC_{RCHF}[2]	全温区 RCHF 变化范围	VDD=1.65~5.5V	FSEL==2'b00 T_A=-40~105℃	-2	-	2	%
			FSEL==2'b01 T_A=-40~105℃	-2.5	-	3	%
			FSEL==2'b10 T_A=-40~105℃	-3	-	4	%
			FSEL==2'b11 T_A=-40~105℃	-5	-	5	%

表 3-13 内部 RC 振荡器参数

[注1]：此项指标由量产测试保证
[注2]：此项指标基于特征参数提取

内部低频 RC 振荡器

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{RCLP}	RCLP 低功耗振荡频率	VDD=1.65~5.5V T_A=25℃		32.768		KHz
I_{DD_RCLP}	RCLP 功耗	VDD=1.65~5.5V T_A=25℃		400		nA
ACC_{RCLP}[1]	全温区 RCLP 变化范围	VDD=1.65~5.5V T_A=-40~105℃	-8		8	%

表 3-14 内部 RC 振荡器参数

[注1]：此项指标基于特征参数提取

3.4.8 PLL 特性

符号	参数说明	测试条件	参数值	单位

			最小值	典型值	最大值	
F_{PLL}	PLL 输出频率	VDD=1.8~5.5V T_A=25℃	32		48	MHz
I_{DD_PLL}	PLL 功耗	输出频率 32Mhz		450		uA
		输出频率 48Mhz		550		
t_{LOCK}	PLL 锁定时间			65		us

表 3-15 PLL 参数

3.4.9 ADC 特性

3.4.9.1 参数说明

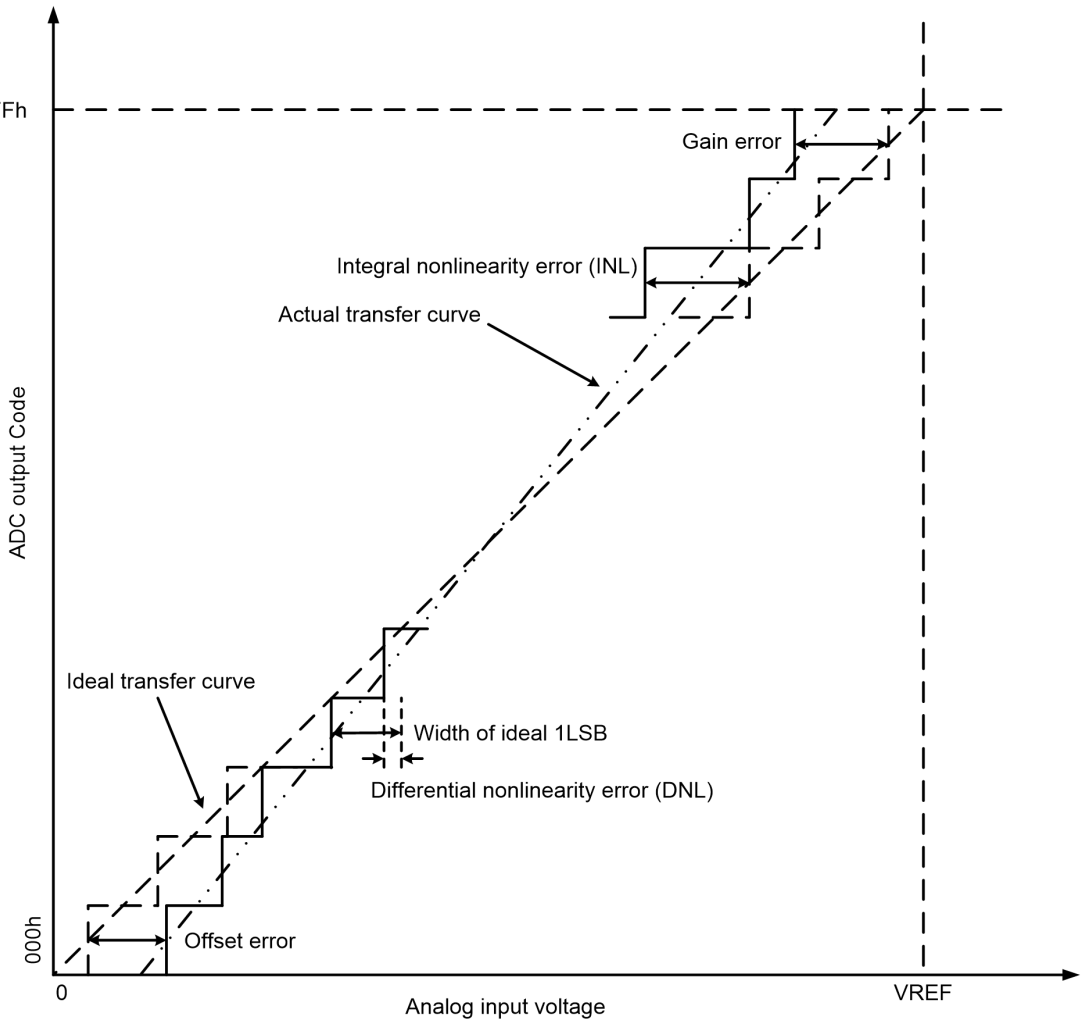


图 3-2 ADC 参数说明

差分非线性（DNL）

DNL表示理想ADC转换曲线1LSB宽度与实际ADC转换曲线1LSB宽度的差值。

积分非线性（INL）

INL表示实际ADC转换曲线与理想ADC转换曲线之间最大偏差。

失调误差（Offset error）

Offset误差表示实际ADC第一个码字跳变的位置与理想ADC第一个码字变化位置之间的差值。

增益误差（Gain error）

Gain error表示满幅输入时实际ADC最后一个码字变化的位置与理想ADC最后一个码字变化位置之间的差值。

3.4.9.2 性能指标

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
VDDA	工作电压范围	-	1.6		5.5	V
VREF+	正参考电压	-	1.5		VDDA	V
VREF-	负参考电压	-	0		0.5	V
T_J	工作结温范围	-	-40		125	°C
V_{AIN}	输入电压范围		VREF-		VREF+	V
C_s	采样保持电容	-		5		pF
R_{AIN}	外部输入阻抗	-			2	KΩ
R_{ADC}	采样开关阻抗	-		300		Ω
F_{CLK}	ADC 工作时钟频率	-			32	MHz
F_S	ADC 采样频率	VDDA=2.0~5.5V			2	Msps
		VDDA=1.6~2.0V			1.5	
T_{SAMP}	采样保持时间	-	2		512	F_{CLK}
T_{CONV}	转换时间	-		14		F_{CLK}
T_{CAL}	自校准时间	-		4096		F_{CLK}
ADC 动态性能						
ENOB	有效位数与输入信号频率的关系 VDDA=3.3V F_S=1Msps ADCCLK=XTHF F_{AIN}=29KHz -40℃≤T_A≤105℃	单端模式 VREF+=VDDA VREF-=VSSA 或 VREF+=VREFP VREF-=VREFN		10.6		bits
		差分模式 VREF+=VDDA VREF-=VSSA 或 VREF+=VREFP VREF-=VREFN		11.5		bits
	有效位数与工作电压的关系 VREF+=VDDA VREF-=VSSA F_{AIN}=29KHz -40℃≤T_A≤105℃	VDDA=5.0V F_S=1Msps		10.7		bits
		VDDA=3.3V F_S=1Msps		10.6		bits
SNDR	信噪失真比 VDDA=3.3V F_S=1Msps -40℃≤T_A≤105℃	单端模式 VREF+=VDDA VREF-=VSSA 或 VREF+=VREFP VREF-=VREFN		10.5		dB
SFDR	无杂散动态范围 VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz -40℃≤T_A≤105℃	单端模式 VREF+=VDDA VREF-=VSSA 或 VREF+=VREFP VREF-=VREFN		75.7		dB

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
		差分模式 VREF+=VDDA VREF-=VSSA 或 VREF+=VREFP VREF-=VREFN		87.1		dB
ADC 静态性能						
DNL	差分非线性 Fs=2Msps	单端模式 差分模式 VDDA=1.8~5.5V	-1		1	LSB
INL	积分非线性 Fs=2Msps	单端模式 差分模式 VDDA=1.8~5.5V	-1		1	LSB
OffsetError	失调误差	单端模式		-0.5		LSB
GainError	增益误差	单端模式		-0.1		%

表 3-16 ADC 参数

3.4.9.3 输入通道阻抗

下图表示了 ADC 输入通道的阻抗分布。

- ADC_INx 表示快速外部通道
- ADC_INy 表示慢速外部通道
- R_{IO} 表示引脚输入开关阻抗，R_{ADC1} 和 R_{ADC2} 表示 ADC 输入快速通道阻抗和慢速通道阻抗
- C_S 表示 ADC 内部采样电容，典型值 5pF
- 阻抗参数参见后续表格

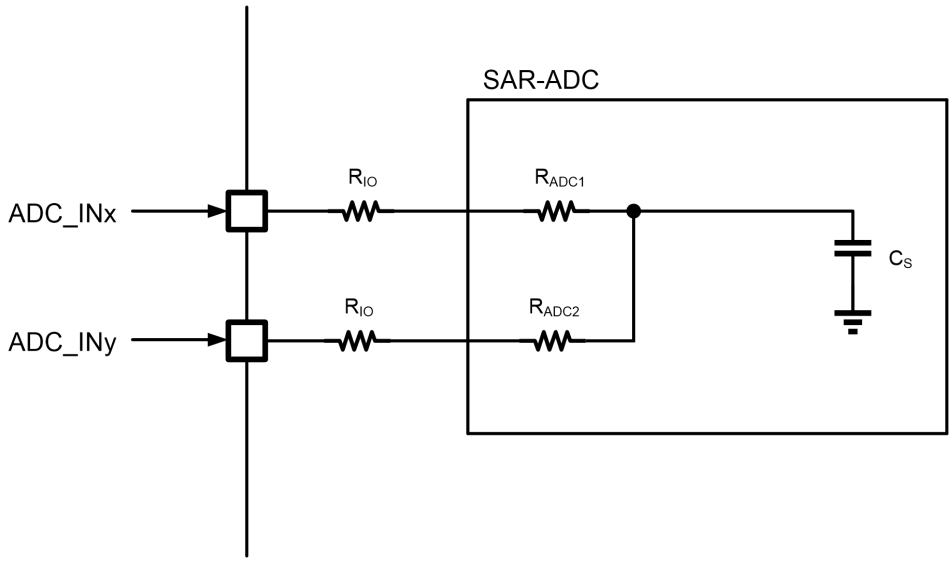


图 3-3 ADC 通道输入阻抗

3.4.9.4 采样时间

ADC 输入信号采样时间最小值由被采样的模拟信号源内阻、信号输入通道阻抗、引脚寄生电容、采样电容共同决定。

ADC 采样外部输入信号的最小采样时间要求根据下式计算：

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC} + R_{IO}\right) \times C_{ADC}$$

其中 n=12，SA=0.25LSB（指采样电容上的电压建立到被采样信号电平的 0.25LSB 误差以内），R_{AIN} 表示被采样信号源内阻，R_{IO} 表示输入 IO 阻抗，R_{ADC} 表示 ADC 输入通道阻抗，C_{ADC} 表示 ADC 采样电容。其中 R_{IO} 为 100Ω。

慢速通道的 R_{ADC} 受电源电压、温度和输入信号幅度影响，在输入信号为 VDDA/2 时开关阻抗最大。快速通道则与输入信号幅度无关。以下表格提供了在不同电源、温度条件下，不同通道的 R_{ADC} 参数，用户可以根据这些参数以及信号源特性，计算所需的最小采样时间。

符号	VDDA	温度	min	typ	max
快速通道 （单端输入阻抗 R_{ADC}）	5V	-40	-	890	950
		55	-	1015	1103
		125	-	1118	1240
	3.3V	-40	-	1020	1120
		55	-	1200	1360
		125	-	1346	1550
	1.6V	-40	-	2260	3420
		55	-	2500	3520
		125	-	2660	3630
符号	**VDDA**	**温度**	**min**	**typ**	**max**
慢速通道 输入电平 0.5*AVDD （单端输入阻抗 R_{ADC}）	5V	-40	-	1403	1530
		55	-	1600	1782
		125	-	1750	1975
	3.3V	-40	-	1690	1942
		55	-	1965	2310
		125	-	2160	5560
	1.6V	-40	-	6640	-
		55	-	5260	9100
		125	-	5000	7760

表 3-17 ADC 采样时间

3.4.10 温度传感器

芯片出厂时经过温度定标，定标条件是 VDDA=3.0V，TA=30+/-1°C。在此条件下，使用 ADC 采样

并转换温度传感器输出电压，将转换结果保存在 Flash 指定地址。

符号	参数说明	测试条件	数据保存地址
TS_CAL1	温度传感器标定值 1	VDDA=3.0V，T_A=30+/-1°C	请联系复旦微电子获取驱动库函数

表 3-18 ADC 采样时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
Reso	分辨率			0.3		°C
Slope	输出斜率[1]			2.51		mV/°C
Linerity	全温区线性度[1]		-	+/-3	+/-5	°C
I_{DDA}	温度传感器功耗（不含 ADC）[2]			1	2	uA
t_{START}	温度传感器启动时间[2]				10	us
t_{SAMPLE}	ADC 采样温度传感器输出时要求的采样时间[2]		10	-	-	us

表 3-19 温度传感器参数

[1] 基于特征参数提取

[2] 基于电路设计仿真

3.4.11 模拟比较器特性

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
VDDA	比较器工作电压范围	-		1.6		5.5	V
VI_{comp1}	比较器1输入电压范围	-		0	-	VDDA	V
VI_{comp2}	比较器2输入电压范围	-		0	-	VDDA-0.7	V
I_{comp1}	比较器1工作电流	VDDA=5V 50Khz 方波，±100mV overdrive	低功耗模式			1.3	uA
			中速模式			2	
			高速模式			30	
$T_{propagation1}$	比较器1传播延迟	VDDA=5V 200mV step 100mV overdrive	低功耗模式		2		us
			中速模式		1		
			高速模式		100		ns
T_{setup1}	比较器1建立时间	VDDA=5V	低功耗模式			20	us
			中速模式			15	
			高速模式			5	
I_{comp2}	比较器2工作电流	VDDA=5V			0.5		uA
$T_{propagation2}$	比较器2传播延迟	VDDA=5V			5		us
T_{setup2}	比较器2建立时间	VDDA=5V				15	us

表 3-20 模拟比较器参数

3.4.12 Flash 存储器特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	Code Flash size		128K	-	384K	bytes
	Data Flash size			8K		bytes
T_{PROG}	Word Program Time		-	30	-	μs
T_{ERASE}	Page/Sector Erase		-	2	-	ms
	Chip Erase		30	-	40	ms
N_{ED}	Sector Endurance	code flash	-	100,000	-	Erase/Write cycles
		data flash (ECC on)	-	500,000	-	Erase/Write cycles
T_{DR}	Data Retention	T_A=105℃ After 20K cycling	10			yrs

表 3-21 Flash 参数

3.4.13 GPIO 特性

普通 IO

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		$0.3V_{DD}$	V
V_{IH}	输入高电平		$0.7V_{DD}$		V_{DD}	V
V_{TL}	施密特输入低电平	V_{DD}=5V		2		V
		V_{DD}=3.3V		1.32		V
V_{TH}	施密特输入高电平	V_{DD}=5V		2.46		V
		V_{DD}=3.3V		1.68		V
I_{IL}	输入低漏电	V_{IL}=0V	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-1		1	μA
V_{OL}	输出低电平（IO 驱动强度 X4）	V_{DD}=5VI_{SINK}=5mA		0.1		V
		V_{DD}=5VI_{SINK}=10mA		0.2		
		V_{DD}=5VI_{SINK}=20mA		0.45		V
V_{OH}	输出高电平（IO 驱动强度 X4）	V_{DD}=5VI_{SOURCE}=5mA		4.85		V
		V_{DD}=5VI_{SOURCE}=10mA		4.7		
		V_{DD}=5VI_{SOURCE}=20mA		4.3		V
R_{PU}	弱上拉电阻			50		KΩ
R_{PD}	弱下拉电阻			50		KΩ

表 3-22 普通 I/O 参数

NRST 引脚

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{IL}	输入低电平		0		$0.3V_{DD}$	V
V_{IH}	输入高电平		$0.7V_{DD}$		V_{DD}	V
V_{TL}	施密特输入低电平	V_{DD}=5V		2		V
		V_{DD}=3.3V		1.32		V
V_{TH}	施密特输入高电平	V_{DD}=5V		2.46		V
		V_{DD}=3.3V		1.68		V
I_{IL}	输入低漏电	V_{IL}=0V	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-1		1	μA
R_{PU}	上拉电阻			5		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=5V		100		ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.8~5.5V -40℃≤T_A≤105℃		1920		us

表 3-23 NRST 引脚参数

注：

[1] 此项参数基于特征参数提取

GPIO AC 特性

IO	符号	参数说明[1]	测试条件	min	max	单位
非 FM+	Fmax	Maximum frequency	C=30pF,2.7V<Vdd<3.6V	-	45	MHz
			C=30pF,1.6V<Vdd<2.7V	-	22	

			C=10pF,2.7V<Vdd<3.6V	-	80	
			C=10pF,1.6V<Vdd<2.7V	-	40	
	Tr/Tf	Output rise and fall time	C=30pF,2.7V<Vdd<3.6V	-	8.7	ns
			C=30pF,1.6V<Vdd<2.7V	-	16.9	
			C=10pF,2.7V<Vdd<3.6V	-	3.4	
			C=10pF,1.6V<Vdd<2.7V	-	6.7	
FM+	Fmax	Maximum frequency	C=50pF,1.6V<Vdd<3.6V	-	10	MHz
	Tf	Output fall time		-	27	ns

表 3-24 引脚 AC 参数

注：

[1] 依据电路仿真，不在量产测试中测试

3.4.14 LCD 特性

片内电阻分压模式

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
I_{LCD}	片内电阻分压模式下的 LCD 工作电流（空载）[1]	VDD=5V		2		uA
		VDD=3V		1.8		
V_{LCD}	LCD 偏置电压	-	0.547× VDD		VDD	V

表 3-25 LCD 片内电阻分压

[1]基于特征参数提取

4 总线与存储

4.1 系统总线

FM33FR0xx 总线架构包含以下主要部件：

- 两个 Master
  - Cortex-M0 内核
  - DMA 控制器
- 四个 Slave
  - 内部 Flash 存储器
  - 内部 SRAM 存储器
  - GPIO 控制器模块
  - AHB-APB 总线转接桥和系统控制寄存器

FM33FR0 的系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

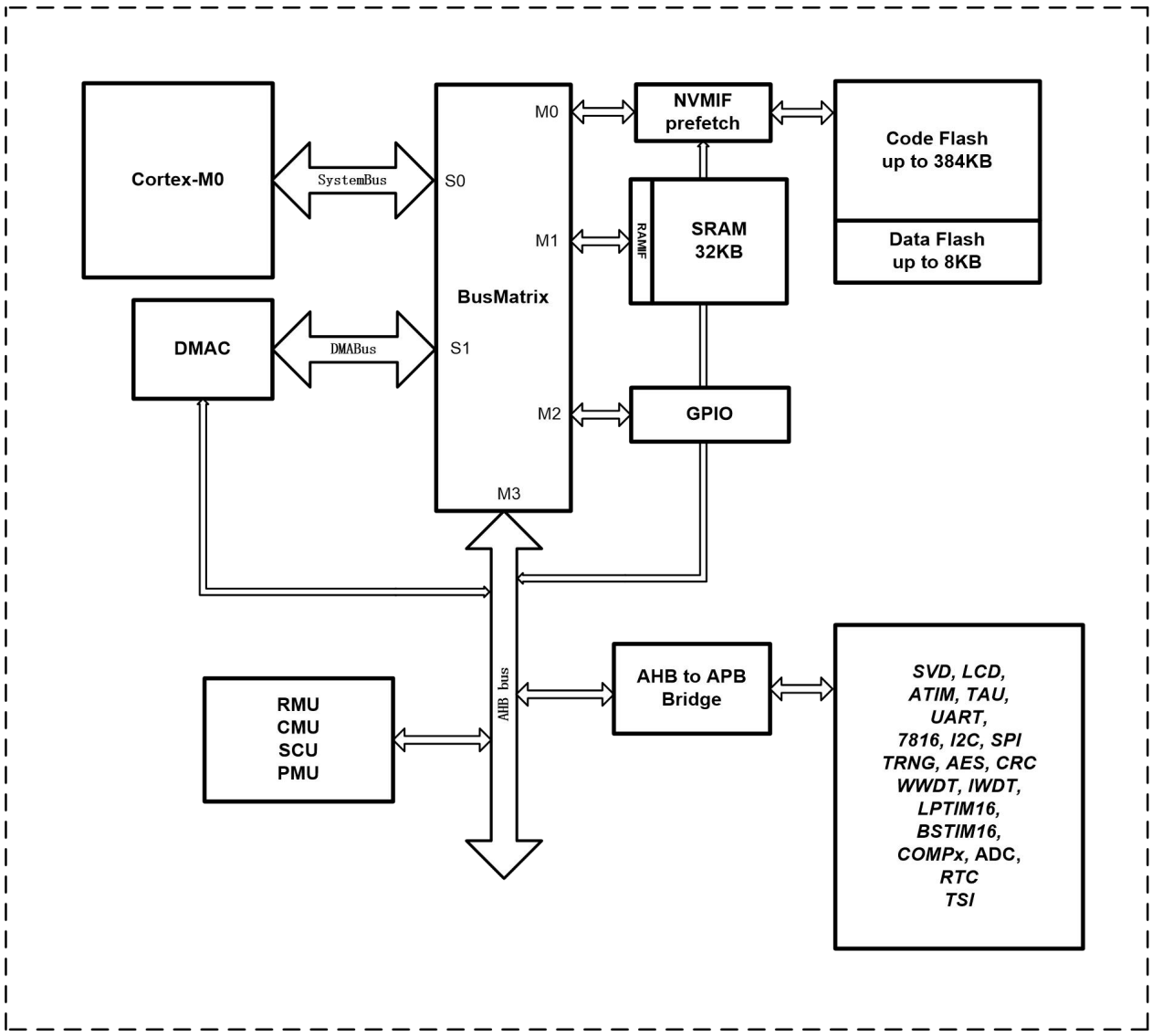

Cortex-M0
SystemBus
S0
M0
NVMIF
prefetch
Code Flash
up to 384KB
Data Flash
up to 8KB
M1
RAMF
SRAM
32KB
BusMatrix
DMAC
DMABus
S1
M2
GPIO
M3
AHB bus
AHB to APB
Bridge
RMU
CMU
SCU
PMU
SVD, LCD,
ATIM, TAU,
UART,
7816, I2C, SPI
TRNG, AES, CRC
WWDT, IWDT,
LPTIM16,
BSTIM16,
COMPx, ADC,
RTC
TSI

图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

Flash 页（Page）大小为 512 字节，每 4 个页组成一个 2K 字节的扇区（Sector）。

Flash 包含 4 个 information 页，2 个 LDT 页，1 个冗余页，1 个 DCT 页。其中 DCT 和 LDT 为芯片原厂保留页，不对用户开放。information 为用户配置页，用于保存用户配置信息。所有 option 页在地址上与 Flash 主区域互相隔离。

当芯片从 Flash 启动时，FM33FR0 的地址空间分配如下图（384KB code Flash，8KB data flash, 32KB RAM）：

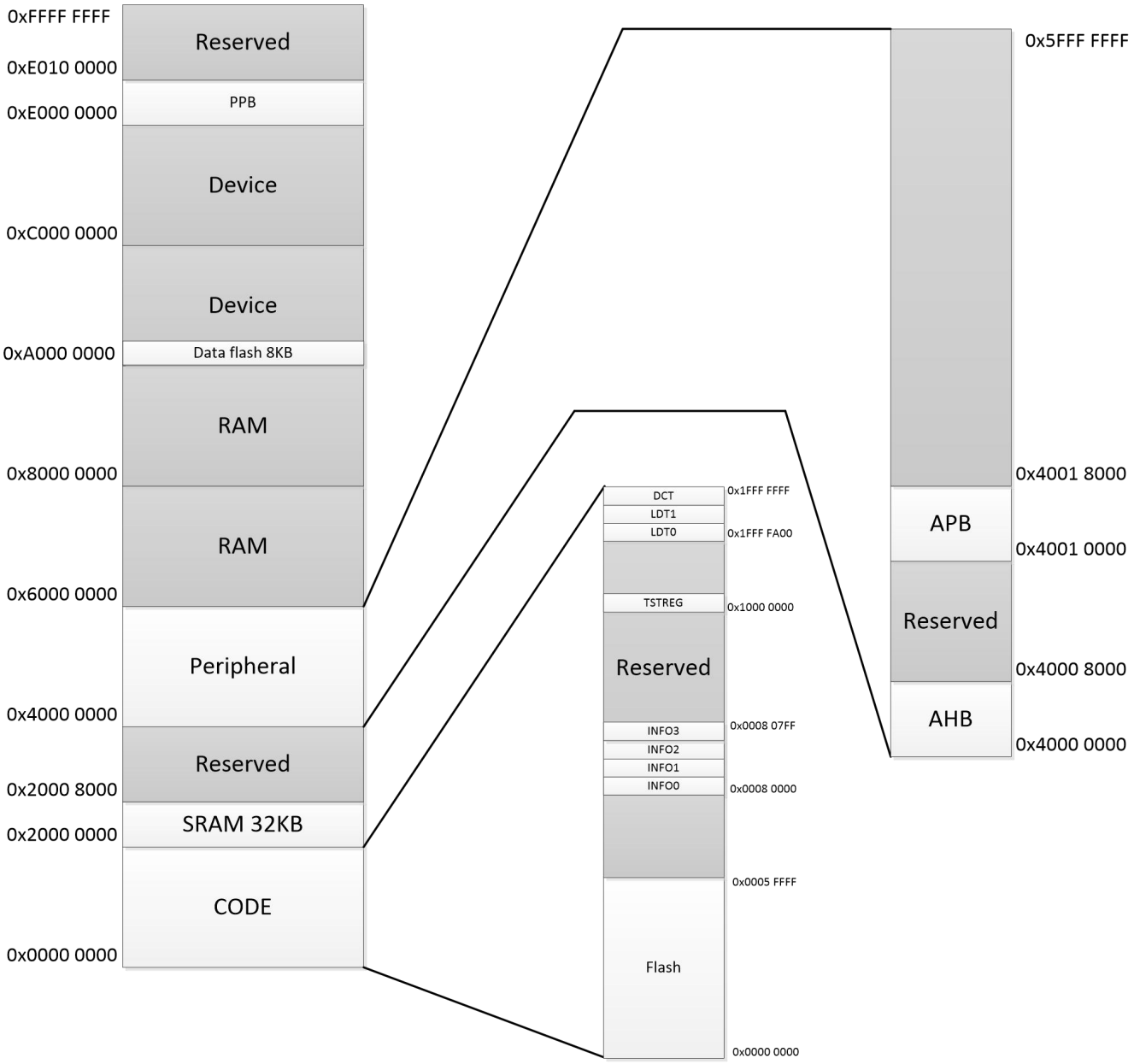


图 4-2FM33FR0 总线地址（384+32+8）

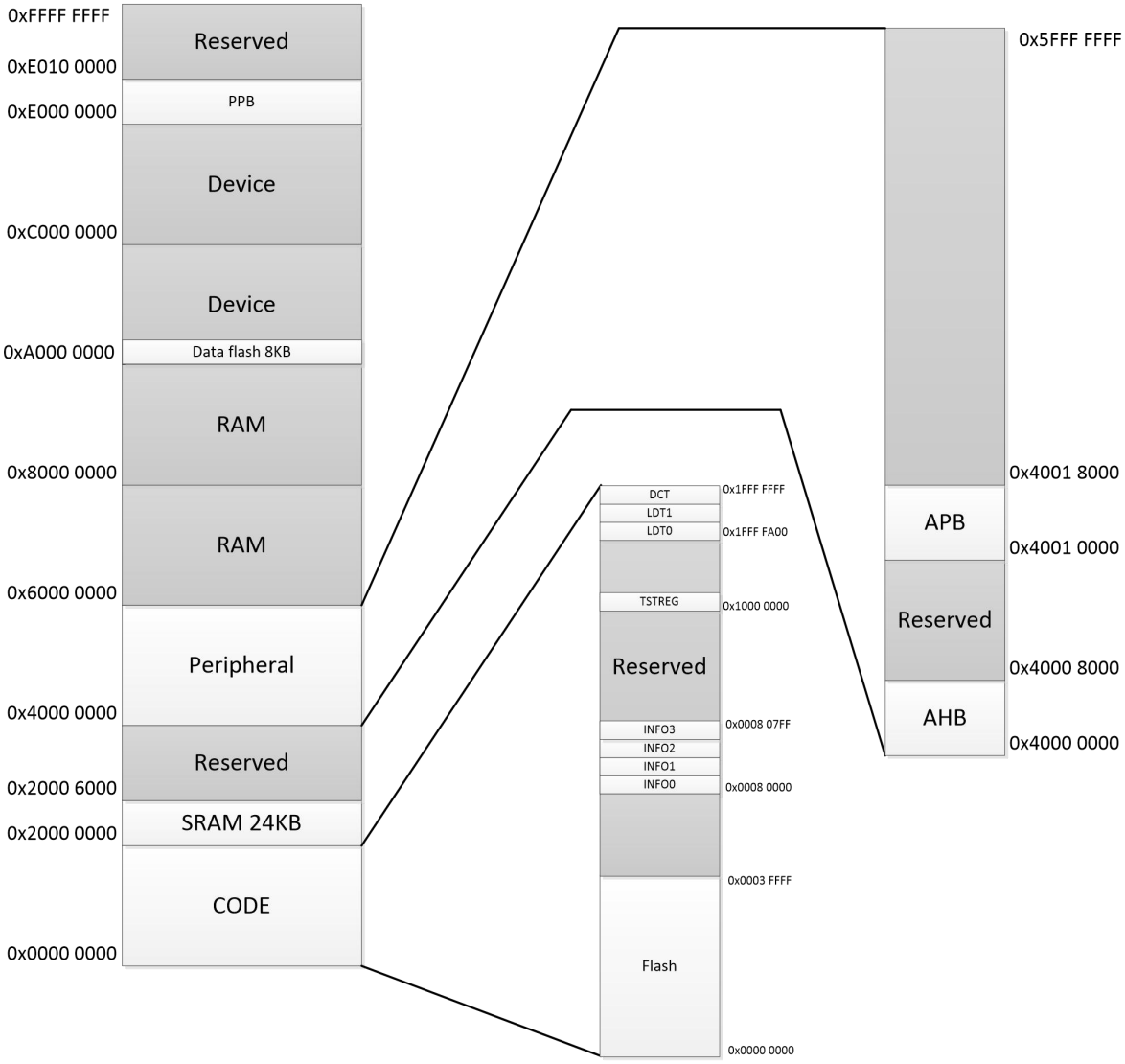


图 4- 3FM33FR0 总线地址（256+24+8）

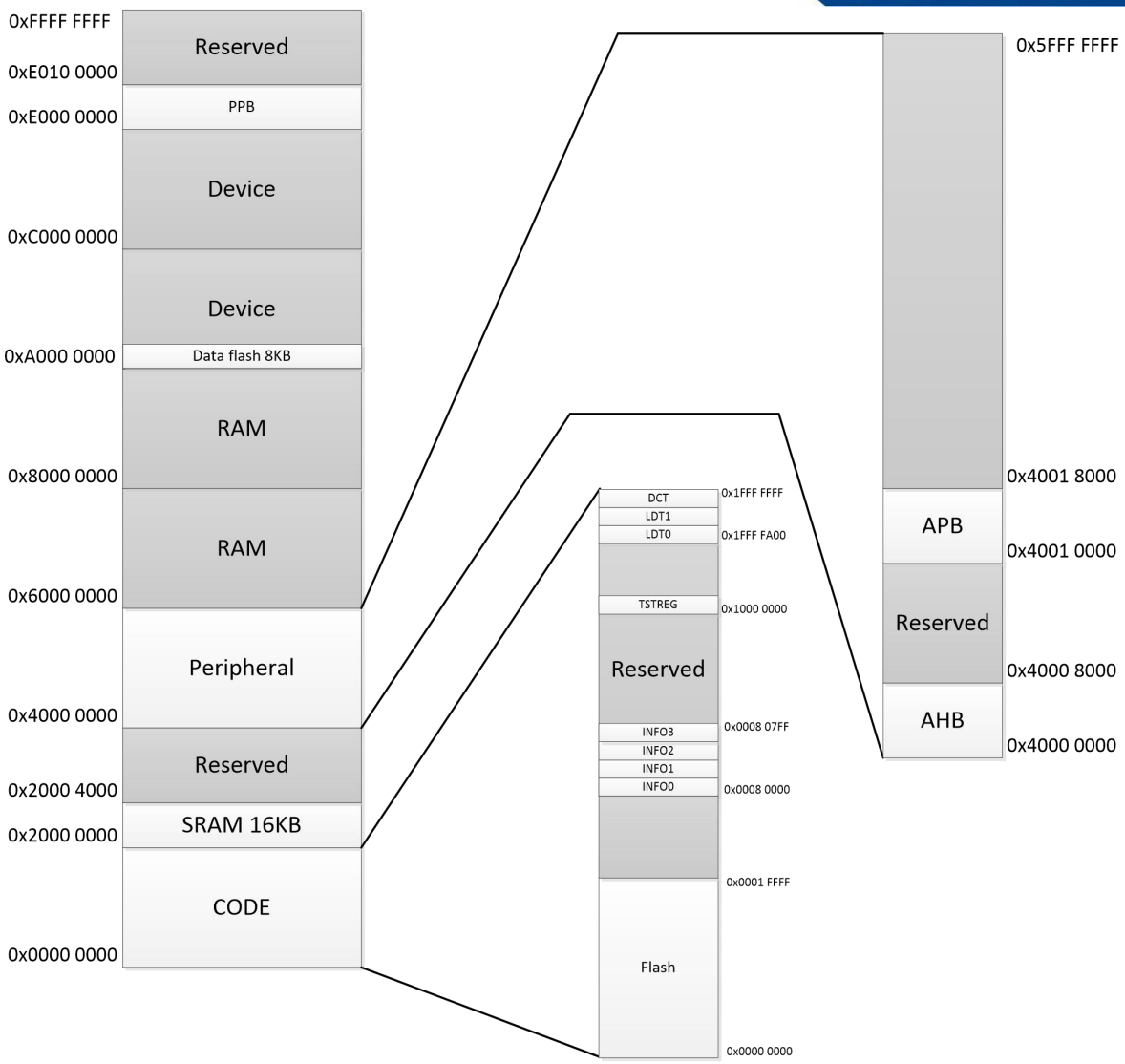


图 4- 4FM33FR0 总线地址（128+16+8）

4.2.2 外设模块寄存器地址分配

下表罗列了所有外设模块的地址空间分配范围，每个外设模块占用 1KB 地址空间。

总线	起始地址	空间	外设
AHB	0x0000_0000	384KB	Flash main array
	0x1FFF_F000	4KB	Flash Option cell array
	0x2000_0000	32KB	SRAM
	0x2000_8000	-	-
	0x4000_0000	1KB	SCU
	0x4000_0400	1KB	DMA
	0x4000_0800	1KB	RAMBIST
	0x4000_0C00	1KB	GPIO
	0x4000_1000	1KB	NVMIF
	0x4000_1400	-	-
	0x4000_1800	-	-
	0x4000_1C00	1KB	SFU
	0x4000_2000	1KB	PMU

	0x4000_2400	1KB	CMU
	0x4000_2800	1KB	RMU
	0x4001_0000	64KB	APB
APB	0x4001_0000	1KB	-
	0x4001_0400	1KB	-
	0x4001_0800	1KB	SPI2
	0x4001_0C00	1KB	LCD
	0x4001_1000	1KB	RTC
	0x4001_1400	1KB	IWDT
	0x4001_1800	1KB	WWDT
	0x4001_1C00	1KB	UART0
	0x4001_2000	1KB	UART1
	0x4001_2400	1KB	I2C_SMB0
	0x4001_2800	1KB	I2C_SMB1
	0x4001_2C00	1KB	-
	0x4001_3000	1KB	HDIV
	0x4001_3400	1KB	LPTIM16
	0x4001_3800	1KB	PGL
	0x4001_3C00	1KB	TAU
	0x4001_4000	1KB	SPI3
	0x4001_4400	1KB	-
	0x4001_4800	1KB	-
	0x4001_4C00	1KB	-
	0x4001_5000	1KB	I2C0
	0x4001_5400	1KB	I2C1
	0x4001_5800	1KB	I2C2
	0x4001_5C00	1KB	I2C3
	0x4001_6000	1KB	INTMUX
	0x4001_6400	1KB	-
	0x4001_6800	1KB	-
	0x4001_6C00	1KB	-
	0x4001_7000	1KB	-
	0x4001_7400	1KB	UART2
	0x4001_7800	1KB	UART3
	0x4001_7C00	1KB	UARTIR
	0x4001_8000	1KB	CRC
	0x4001_8400	1KB	-
	0x4001_8800	1KB	-
	0x4001_8C00	1KB	SPI1
	0x4001_9000	1KB	SPI0
	0x4001_9400	1KB	-
	0x4001_9800	1KB	ANTEST
	0x4001_9C00	1KB	COMPARATOR
	0x4001_A000	1KB	UART4
	0x4001_A400	1KB	UART5
	0x4001_A800	1KB	SVD
	0x4001_AC00	1KB	ADC
	0x4001_B000	1KB	ATIM
	0x4001_B400	1KB	BSTIM16
	0x4001_B800	1KB	AES
	0x4001_BC00	1KB	TRNG

表 4-1 外设模块总线地址列表

4.3 RAM

4.3.1 概述

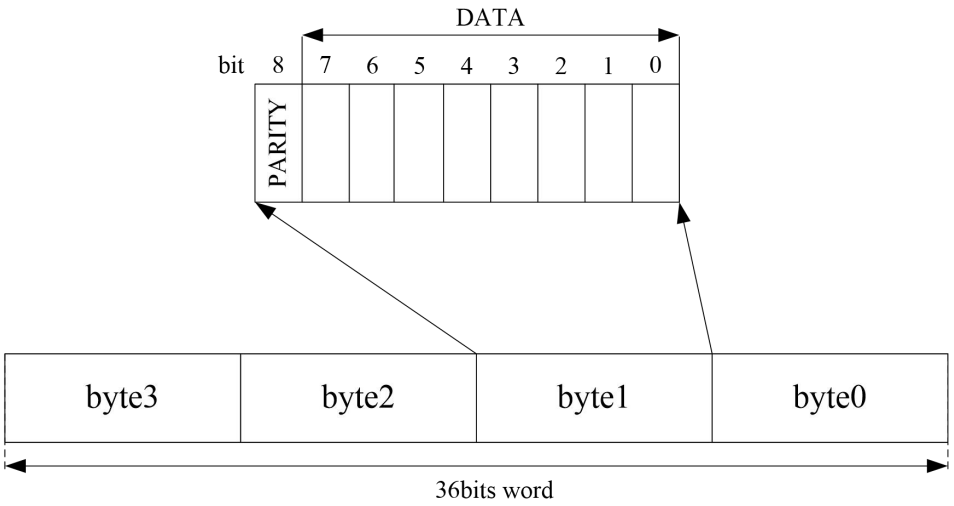
FM33FR0 含有一块 32KB RAM，SRAM 地址空间范围是 0x2000_0000~0x2000_7FFF，软件可以对 SRAM 进行字节、半字、字访问，CPU 和 DMA 都可以以最大系统频率对 SRAM 实现无等待的单周期读写。CPU 也可以从 SRAM 取指执行程序，因此在对程序效率要求高的场合，可以将部分代码导入 SRAM 中，实现最高频率下无等待的执行。

4.3.2 奇偶校验

FM33FR0 的 SRAM 支持硬件奇偶校验，每字节带有 1bit 校验位，以满足 IEC60730 标准要求的 RAM 检验功能。

在软件向 RAM 写入数据时，硬件自动以字节为单位计算校验位并与数据一同写入，软件读出时则同时读出校验位并进行实时校验，如果校验失败，则根据配置寄存器状态，产生校验错误中断或复位。

RAM 自宽增加到 36bit，每个字节内部 XOR 后形成校验位，保存在第 9bit。



注意：上电复位后 RAM 中的数据为乱码，此时读取极有可能触发校验错误中断，因此软件需在上电后首先对 RAM 内容进行初始化，再打开 RAM 校验功能。

4.4 Flash

4.4.1 概述

FM33FR0 使用的 Flash 容量为 392KB，其中包含 384KB 程序空间和 8KB 数据 flash；阵列组织格式包含 page(512B)、sector(2KB)、mat(392KB)

main array 共包含 784 个页，另有 4 个信息页，2 个 LDT（user trim）页，1 个 RED（冗余）页，1 个 DCT（flash core trim）页。Flash 支持页擦、扇区擦和全擦。

Main array 包含 ECC，数据位宽 39bit（32bit 数据及 7bit ECC），data flash 可以支持 50 万次擦写（通过 ECC 保证），但是读取周期较长（80ns），code flash 支持 10 万次擦写，读取周期为 30ns。

对 Flash 的全擦操作不会擦除 data flash，可以实现全擦程序的同时不擦除用户数据。

4.4.2 数据 flash

FM33FR0 包含 8KB 数据 flash，擦写寿命可达 50 万次，读取周期为 80ns。CPU 不能同时访问 code flash 和 data flash，只能分时访问。CPU 对 data flash 只能进行数据访问，不能执行取指操作（XN 属性），对 data flash 的取指操作将触发 hardfault。

上电后默认主频为 8MHz 的情况下可以实现 data flash 单周期访问，当主频提高后，软件必须正确配置 data flash wait cycle 来保证正确的读取 data flash。

注意 data flash 被逻辑映射到 0xA000 0000 位置。

4.4.2.1 Data flash 全擦保护

8KB data flash 在执行全擦操作时不会被擦除，data flash 只能单独执行页擦和扇区擦。

4.4.2.2 ECC

Code 和 data flash 各自有独立的 ECC 使能；上电复位后，data flash ECC 默认使能，code flash ECC 默认关闭。

注意：data flash ECC 和 code flash ECC 需同时使能以正确的产生 ECC 中断。

注意：使用 ECC 功能必须使能 flash 控制器时钟（CMU_PCLKCR2.NVM_PCE=1）

Flash ECC 中断：

- ECC detection 中断：检测到 2bit 错误，无法纠错，此时产生 NMI 中断，无中断屏蔽手段
- ECC correction 中断：纠正 1bit 错误，不影响正常运行，此时产生的中断可以屏蔽
- Data 和 code flash 可以分别产生 detection 和 correction 标志

- 错误产生时，ADDR_ECC 寄存器记录错误发生的 flash word 地址

4.4.3 特殊信息扇区说明

Flash IP 共有 6 个特殊页，说明如下

区域	说明	用途
LDT0	FMSH 数据区	保存 FMSH 的调校信息、模式字、测试数据等；软件只读
LDT1	用户选项数据区	用户选项字节（OPTBYTES）
IF	Information 区	4 个 page 共 2KB，供用户使用；

4.4.3.1 LDT0 page

LDT0为FMSH配置信息区，仅能在制造商模式下改写，用户模式下只读。

LDT0 的总线地址是 0x1FFF_FA00~0x1FFF_FBFF；以下参数软件在应用中可以从 LDT0 读取并写入对应的控制寄存器，以实现模拟参数校准。

AHB地址	物理地址 LDTEN[0]=1	[31:16]	[15:0]	说明
0x1FFF_FB08	0x0042	~AVREFCAL	AVREFCAL	3V,30C 下ADC对AVREF的转换值
0x1FFF_FB10	0x0044	~TS_CAL	TS_CAL	3V, 30C下ADC对温度传感器的电压值
0x1FFF_FB20	0x0048	~RCLP_TRIM	RCLP_TRIM	RCLP调校值
0x1FFF_FB34	0x0053	~RCHF32TRIM	RCHF32TRIM	RCHF 32MHz调校值
0x1FFF_FB38	0x0054	~RCHF24TRIM	RCHF24TRIM	RCHF 24MHz调校值
0x1FFF_FB3C	0x0055	~RCHF16TRIM	RCHF16TRIM	RCHF 16MHz调校值
0x1FFF_FB40	0x0056	~RCHF8TRIM	RCHF8TRIM	RCHF 8MHz 调校值(auto-load)
0x1FFF_FB4C	0x0059	~VREFP15	VREFP15	VREFP 1.5V档位校准值
0x1FFF_FB50	0x005A	~VREFP20	VREFP20	VREFP 2.0V档位校准值
0x1FFF_FB54	0x005B	~VREFP25	VREFP25	VREFP 2.5V档位校准值
0x1FFF_FB58	0x005C	~VREFP30	VREFP30	VREFP 3.0V档位校准值

为了保证数据可靠，LDT0中的参数都采用高低半字互为正反码校验的方式保存。软件使用这些参数时，应先进行正反码校验，结果正确的情况下可以使用，否则应保持默认参数。

详细参数格式定义如下：

助记符	[31:16]	[15:0]	功能描述
VREFCAL	{4'h0, ~VREFCAL}	{4'hF, VREFCAL}	VDDA=VREFP=3V+/-10mV, 30C+/-1C 下ADC对AVREF的转换值

助记符	[31:16]	[15:0]	功能描述
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0]表示8bit调校值
RCHF32TRIM RCHF24TRIM RCHF16TRIM RCHF8TRIM	{9'b0000_0000_0, ~RCHFtrim[6:0]}	{9'b1111_1111_1,RCHFtrim[6:0]}	RCHFtrim[6:0]表示7bit调校值
VREFP15 VREFP20 VREFP25 VREFP30	{8'b0000_0000_0, ~VREFPtrim[7:0]}	{8'b1111_1111_1,VREFPtrim[7:0]}	VREFPtrim[7:0]表示 8bit 调校值

4.4.3.2 LDT1 page

LDT1为用户配置信息区，可以在用户模式下改写（仅能使用SWD改写，即用户通过编程器改写）。

LDT1的总线地址是0x1FFF_FC00~0x1FFF_FDFF；

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FC00	~OPTBYTES[15:0]	OPTBYTES[15:0]	用户选项字节低半字
0x1FFF_FC04	~OPTBYTES[31:16]	OPTBYTES[31:16]	用户选项字节高半字
0x1FFF_FC08	LOCK1		ACLOCK 配置字，控制最低 128KB flash
0x1FFF_FC0C	LOCK2		ACLOCK 配置字，控制中间 128KB flash
0x1FFF_FC10	LOCK3		ACLOCK 配置字，控制最高 128KB flash

OPTBYTES 选项字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:24	IWDTEN	复位后 IWDT 默认状态 0x55：关闭 IWDT 其他：使能 IWDT	0xFF
23:20	IWDTSLP	配置 IWDT 在低功耗模式下是否允许停止计数 0xA：在 Sleep/DeepSleep 模式下允许应用停止 IWDT 计数 其他：任何模式下禁止应用停止 IWDT	0xA
19:16	-	-	-
15:8	ACLKEN	应用代码保护使能 0x33：禁止 ACLOCK 其他：使能 ACLOCK	0x33
7:0	DBRDPEN	调试接口访问保护使能 0xAA：关闭调试接口保护 其他：使能调试接口保护	0xAA

【注】在出厂时，SWD 接口可以任意改写 OPTBYTES；但是一旦 ACLKEN 或 DBRDP 被使能，用户必须通过 SWD 全擦 flash 后才能重新改写 OPTBYTES。

LOCK 配置字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:0	LOCK1	Block Lock 字 1，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK1[1:0]对应 Block0（Flash 最低地址 8KB 空间），LOCK1[31:30] 对应 Block15（Flash 地址空间 120~128KB），其他以此类推	0xFFFFFFFF
31:0	LOCK2	Block Lock 字 2，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK2[1:0] 对应 Block16，LOCK2[31:30] 对应 Block31（Flash 地址空间 248~256KB），其他以此类推	0xFFFFFFFF
31:0	LOCK3	Block Lock 字 3，每 2it 对应 8KB Block 11：无保护 01、10：软件读写保护，仅取指 00：软件读写保护，仅取指；SWD 读写保护 LOCK3[1:0]对应 Block32（Flash 最低地址 8KB 空间），LOCK3[31:30]对应 Block47（Flash 地址空间 376~384KB），其他以此类推	0xFFFFFFFF

LOCK bit 和 Flash 权限锁定地址的对应关系如下表：

Address	LOCK bits
0x0000_0000 ~ 0x0000_1FFF	LOCK1[1:0]
0x0000_2000 ~ 0x0000_3FFF	LOCK1[3:2]
0x0000_4000 ~ 0x0000_5FFF	LOCK1[5:4]
0x0000_6000 ~ 0x0000_7FFF	LOCK1[7:6]
0x0000_8000 ~ 0x0000_9FFF	LOCK1[9:8]
0x0000_A000 ~ 0x0000_BFFF	LOCK1[11:10]
0x0000_C000 ~ 0x0000_DFFF	LOCK1[13:12]
0x0000_E000 ~ 0x0000_FFFF	LOCK1[15:14]
0x0001_0000 ~ 0x0001_1FFF	LOCK1[17:16]
0x0001_2000 ~ 0x0001_3FFF	LOCK1[19:18]
0x0001_4000 ~ 0x0001_5FFF	LOCK1[21:20]
0x0001_6000 ~ 0x0001_7FFF	LOCK1[23:22]
0x0001_8000 ~ 0x0001_9FFF	LOCK1[25:24]
0x0001_A000 ~ 0x0001_BFFF	LOCK1[27:26]
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]
0x0002_0000 ~ 0x0002_1FFF	LOCK2[1:0]
0x0002_2000 ~ 0x0002_3FFF	LOCK2[3:2]
0x0002_4000 ~ 0x0002_5FFF	LOCK2[5:4]

Address	LOCK bits
0x0002_6000 ~ 0x0002_7FFF	LOCK2[7:6]
0x0002_8000 ~ 0x0002_9FFF	LOCK2[9:8]
0x0002_A000 ~ 0x0002_BFFF	LOCK2[11:10]
0x0002_C000 ~ 0x0002_DFFF	LOCK2[13:12]
0x0002_E000 ~ 0x0002_FFFF	LOCK2[15:14]
0x0003_0000 ~ 0x0003_1FFF	LOCK2[17:16]
0x0003_2000 ~ 0x0003_3FFF	LOCK2[19:18]
0x0003_4000 ~ 0x0003_5FFF	LOCK2[21:20]
0x0003_6000 ~ 0x0003_7FFF	LOCK2[23:22]
0x0003_8000 ~ 0x0003_9FFF	LOCK2[25:24]
0x0003_A000 ~ 0x0003_BFFF	LOCK2[27:26]
0x0003_C000 ~ 0x0003_DFFF	LOCK2[29:28]
0x0003_E000 ~ 0x0003_FFFF	LOCK2[31:30]
0x0004_0000 ~ 0x0004_1FFF	LOCK3[1:0]
0x0004_2000 ~ 0x0004_3FFF	LOCK3[3:2]
0x0004_4000 ~ 0x0004_5FFF	LOCK3[5:4]
0x0004_6000 ~ 0x0004_7FFF	LOCK3[7:6]
0x0004_8000 ~ 0x0004_9FFF	LOCK3[9:8]
0x0004_A000 ~ 0x0004_BFFF	LOCK3[11:10]
0x0004_C000 ~ 0x0004_DFFF	LOCK3[13:12]
0x0004_E000 ~ 0x0004_FFFF	LOCK3[15:14]
0x0005_0000 ~ 0x0005_1FFF	LOCK3[17:16]
0x0005_2000 ~ 0x0005_3FFF	LOCK3[19:18]
0x0005_4000 ~ 0x0005_5FFF	LOCK3[21:20]
0x0005_6000 ~ 0x0005_7FFF	LOCK3[23:22]
0x0005_8000 ~ 0x0005_9FFF	LOCK3[25:24]
0x0005_A000 ~ 0x0005_BFFF	LOCK3[27:26]
0x0005_C000 ~ 0x0005_DFFF	LOCK3[29:28]
0x0005_E000 ~ 0x0005_FFFF	LOCK3[31:30]

4.4.3.3 Information1~3 page

这3个information扇区开放给用户作为数据区使用，SWD和软件均可以读写，共1.5KB。

IF1~3的总线地址是0x0008_0200 ~ 0x0008_07FF

4.4.3.4 Information0 page（OTP）

IF0是一个OTP page，这个page在用户模式下只能编程一次，编程后不能擦除或改写。IF0的总线地址是0x0008_0000~0x0008_01FF，共512字节。

其OTP原理是：用户模式出厂后为擦除态，所有数据全FF；用户对其中某个word编程时，硬件先检查当前word内容，如果是0xFFFF_FFFF则可以直接编程，如果不是0xFFFF_FFFF则表示已经被编程过，硬件将直接撤销编程操作并置位错误中断（OTPERR）；此外，用户模式下硬件直接禁止对这个page的擦除操作。通过以上手段，实现OTP模拟。

对IF0 page的读取没有任何限制。

4.4.4 Flash 编程

4.4.4.1 概述

FM33FR0 支持以下 Flash 编程方法：

- 在系统编程（ISP）：通过 FMSH 专用编程器或者 KEIL 用户界面实施芯片编程，使用 SWD 接口
- 在应用编程（IAP）：通过 bootloader 代码实现芯片自编程，用户可定义任意串口，可用于实现程序在线升级

编程前必须对 Flash 进行擦除。Flash 支持三种擦除操作：全擦、扇区擦、页擦；注意数据 flash 不受全擦操作影响。

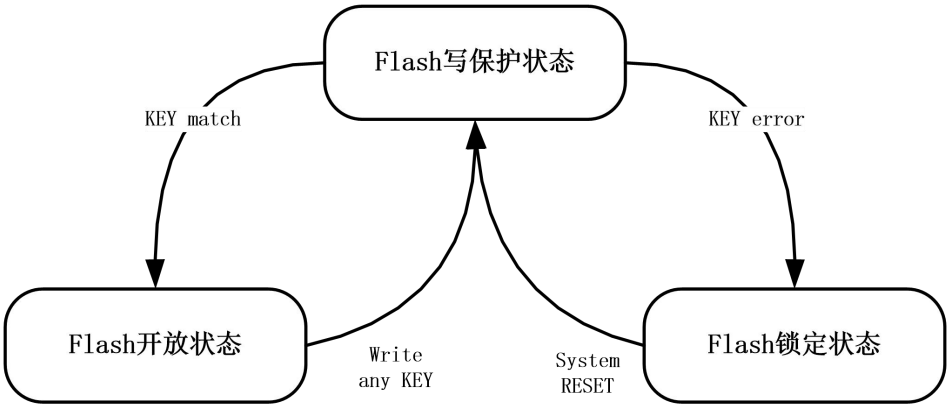
4.4.4.2 Flash 擦写时钟

执行 Flash 擦写时使用校准后的 RCHF 时钟，但是系统时钟可以是任意时钟。NVMIF 根据当前 RCHF 实际频率设置，来产生相应的计时长度。需要支持的 RCHF 频率为 8M、16M 和 24M。

4.4.4.3 Flash 擦写方法

FM33FR0 支持 Flash 擦除操作，以及单次编程和连续编程。

Flash 擦写前须进行 Key 校验，写入顺序错误或写入值错误，或者在 Flash Key 验证正确之前就进行擦除或编程 Flash 操作将会进入错误状态，并产生相应中断。Flash Key 认证错误之后将禁止擦写 Flash 直到下一次复位。而在正常擦写完成后，向 KEY 寄存器写入任意值都会使状态机返回初始的写保护状态。状态转换如下图：



软件可以通过查询 FLSISR.KEYSTA 来确认当前 Key 输入状态，详情参见寄存器说明。

4.4.4.4 全擦操作（Matrix Erase）

全擦操作只能由 SWD 接口启动，软件禁止进行全擦。全擦操作仅擦除程序 flash。SWD 启动全擦操作流程如下：

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D
- SWD 向 Flash 任意地址写擦除请求 0x1234_ABCD
- NVMIF 启动对 Flash 的全擦，并暂停任何 Master 对 Flash 的访问
- 全擦完成后置位中断标志和全擦标志（全擦标志表示程序 flash 全部擦除，任何对程序 flash 的编程将清除此标志）
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：全擦操作只能擦除Flash的main array，不会影响Fuse Array（DCT/LDT/RED/IF/DATA）

4.4.4.5 扇区擦操作（Sector Erase）

SWD 和应用代码都可以执行扇区擦。操作流程如下：

- 配置 ERTYPE 寄存器为 01
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0x3C3C_3C3C
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FLS_KEY 寄存器写任意值恢复写保护

4.4.4.6 页擦操作（Page Erase）

SWD 和应用代码都可以执行页擦。操作流程如下：

- 配置 ERTYPE 寄存器为 00 或 11
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的页内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标页是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标页的擦除，

如果被锁定则触发错误标志

- 页擦完成后置位中断标志
- 软件确认擦除结束后向 FLS_KEY 寄存器写任意值恢复写保护

4.4.4.7 单次编程

单次编程由软件发起，通过总线直接写 Flash，每次操作可执行 word 编程，操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 清除连续编程使能寄存器
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 向 Flash 目标地址写数据，如果目标地址被 ACLOCK 锁定，则触发错误标志，如果没有锁定，则执行编程
- 编程完成后置位中断标志
- 软件确认编程结束后向 FLS_KEY 寄存器写任意值恢复写保护

4.4.4.8 连续编程

连续编程指通过 DMA 的 Memory 通道一次向 Flash 写入 256 字节。连续编程时 DMA 从 RAM 指定地址读取数据，Flash 目标编程地址必须是 half-page 对齐的，也就是 Flash 地址低 8 位为 0。采用这种方式时一次编程的数据长度是固定的，主要用于快速大数据量写入。

在启动连续编程期间，DMA 完全占据 Flash 总线，暂停 CPU 对 Flash 的一切访问。连续编程的操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 置位连续编程使能寄存器（DMA 模式使能）
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- NVMIF 检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 FLS_KEY 寄存器写任意值恢复写保护

注意：如果 CPU 在 Flash 中取指时进行 Flash 擦写，则 CPU 取指将被暂停，直到擦写操作完成。如果 CPU 跳转到 RAM 中取指运行，则 Flash 擦写不会暂停 CPU 的执行。Flash 擦写过程中，若用户希望在 RAM 中执行代码时仍然能够实时响应中断，应将中断向量表重新映射到 RAM 中。

4.4.5 Flash 的内容保护

Flash 内容保护主要用于保护 Flash 中的用户代码、用户数据和用户配置信息被非授权方读取或篡改。

Flash 保护包含两种类型：Debug 接口读取保护（DBRDP-DeBug ReaD Protection）和应用代码权限保护（ACLOCK-Application Code Block Locking）。Flash 保护的控制通过 LDT1 中的 OPTBYTES 来控制。

4.4.5.1 Debug 接口保护（DBRDP）

DBRDP 的主要目的是防止非授权的第三方通过 debug 接口访问芯片 Flash 内容。

DBRDP 由 LDT1 内的 DBRDPEN 配置字使能或者禁止（0xAA 表示禁止 DBRDP，芯片出厂时默认写为 0xAA）。当 DBRDP 使能时，无法通过 SWD 接口读取或擦写 Flash，同时也无法通过 SWD 接口对 RAM 进行访问。

退出 DBRDP 的方法：通过 SWD 对 flash 进行全空间擦写，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 DBRDP，然后复位芯片；复位完成后，芯片将处于无 debug 保护状态。

4.4.5.2 应用代码保护（ACLOCK）

ACLOCK 的主要目的是防止 hacking code 读取或篡改 Flash 中的 application code。通过 ACLOCK 功能，可以设置 CPU 对 Flash 的某些区域只能进行取指操作，不能 read-as-data，也不能擦写。

ACLOCK 以 Block 为单位工作，即对 Flash 保护的颗粒度是 8KB，整个 Flash 包含 48 个 Blocks，对应每个 Block 有 2bit LOCK 信息。出厂时默认的 LOCK 字为 0xFFFF_FFFF，上电 load 的 LOCK 位全部是 11，即默认的无保护状态；当对应 LOCK 位为 01 或 10 时，此 Block 禁止 CPU 擦写和读取，只能取指；当对应的 LOCK 位为 00 时，当前 Block 禁止 CPU 擦写和读取，同时禁止 SWD 擦写和读取。芯片出厂时 LDT1 中关闭 ACLOCK 功能，用户需要通过编程器使能 ACLOCK，并且用户代码编译时要符合 ACLOCK 配置（比如不能将 literal pool 编译到被 LOCK 的 Block）。

ACLOCK 的功能：

- 无保护：所有 Block 允许 CPU 取指、读取、改写，不限制 SWD 访问
- 读写保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不限制 SWD 访问
- 软件和 SWD 保护：指定 Block 允许 CPU 取指，不允许 CPU 和 DMA 读取、擦写，不允许 SWD

读取、擦写

LOCK 位与 Block 访问权限的关系可以参照下表：

LOCK bit	软件读取	软件取指	SWD 读取和擦写
11	允许	允许	允许
01/10	禁止	允许	允许
00	禁止	允许	禁止

ACLOCK 信息在芯片复位时 load 到寄存器中，这些寄存器软件也可以置位，但是不能写 0（即只能提升保护等级）。

ACLOCK 不使能时，LOCK 寄存器内容无效。

注：ACLOCK 的权限控制针对 Flash 各个 Block，与 DBRDP 相互独立。对于 SWD 接口而言，DBRDP 的优先级高于 ACLOCK，即 DBRDP 被使能后，不论 ACLOCK 是否起效，SWD 都无法访问 Flash。

注：不建议 ACLOCK 禁止 1st block 读取，由于 CPU 复位后首先要从 0 地址读取 MSP 指针，ACLOCK 将导致 CPU 无法正常启动。

退出 ACLOCK 的方法：通过 SWD 对 flash 进行全空间擦写，全擦完成后，SWD 可以任意改写 OPTBYTES 禁止 ACLOCK，然后复位芯片；复位完成后，芯片将处于无 ACLOCK 状态。

4.4.5.3 用户模式 Flash 访问权限说明

Flash 空间访问权限分配：

Flash area	DBRDP	LOCK bits (per Block)[3]	SWD	Application
Main array	ON	00	-	R/E/W/F
		01	-	对应 Block 只能取指
		10/11	-	对应 Block 只能取指
	OFF	00	R/E/W	R/E/W/F
		01	-	对应 Block 只能取指
		10/11	对应 Block 无法访问	对应 Block 只能取指
DCT LDT0	x	x	R	R
LDT1	ON	x	R[2]	R
	OFF	x	R/E/W	R

IF1,2,3	x	x	R/E/W	R/E/W
IF0	x	x	R/W	R

注：

[1] R: Read, E: Erase, W: Write, F: Fetch

[2] 进行 flash 全擦后可以擦除 LDT1

[3] 这里假设 ACLOCKEN 有效。ACLOCKEN 无效的情况下，LOCK bits 不起作用。

4.5 寄存器

模块起始地址：0x4000_1000

offset 地址	名称	符号
0x00	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x04	--	--
0x08	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x0C	--	--
0x10	--	--
0x14	Flash 擦写控制寄存器 （Flash Erase/Program Control Register）	FLS_EPCR
0x18	Flash Key 输入寄存器 （Flash Key Register）	FLS_KEY
0x1C	Flash 中断使能寄存器 （Flash Interrupt Enable Register）	FLS_IER
0x20	Flash 标志寄存器 （Flash Interrupt Status Register）	FLS_ISR
0x24	--	--
0x28	Flash ECC 标志寄存器	FLS_ECCSR
0x38	Flash ECC 控制寄存器	FLS_ECCCR
0x60	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x64	ACLOCK 寄存器 2 （Flash Application Code Lock Register1）	FLS_ACLOCK2
0x68	ACLOCK 寄存器 3 （Flash Application Code Lock Register1）	FLS_ACLOCK3

4.5.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DATA_WAIT			CODE_WAIT	
位权限	U-0			R/W-000			R/W-00	

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0

位号	助记符	功能描述
4:2	DATA_WAIT	Data Flash 读等待周期配置 000：无等待 001：1 wait cycle … 111：7 wait cycle Data flash 实际读周期为 DATA_WAIT+1 个 AHB 时钟周期 Data Flash 最小读周期为 80ns，软件应根据当前系统时钟频率合理配置 DATA_WAIT，确保读周期不短于 80ns 比如，当系统频率 AHBCLK 为 24MHz，对应时钟周期为 41.6ns，则读取数据 flash 至少需要 2 个周期，可以配置 DATA_WAIT 为大于等于 1 的数。
1:0	CODE_WAIT	Code Flash 读等待周期配置 (Wait Cycles) 00/11：0 wait cycle 01：1 wait cycle 10：2 wait cycles CPU 主频小于等于 24MHz 时，不需要开启 wait；主频大于 24MHz 小于等于 48MHz 时使能 1 wait

4.5.2 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	IWDTEN	-					
位权限	R-0	R-1	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				ACLOCKEN		DBRDPEN	
位权限	U-0				R-01		R-01	

位号	助记符	功能描述
31	IWDTSLP	IWDT 在休眠模式下是否允许应用暂停计数 (IWDT Sleep) 1：允许应用在休眠模式下暂停 IWDT 计数 0：禁止应用在休眠模式下暂停 IWDT 计数
30	IWDTEN	IWDT 使能 1：上电后 IWDT 开始运行 0：上电后 IWDT 不运行
29:4	-	RFU：未实现，读为 0
3:2	ACLOCKEN	应用代码权限锁定使能 (AppCode Lock Enable) 00/01/11：ACLOCK 不使能 10：ACLOCK 使能

位号	助记符	功能描述
1:0	DBRDPEN	Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能

4.5.3 Flash 擦写控制寄存器（FLS_EPCR）

名称	FLS_EPCR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	ERTYPE	Flash 擦除类型配置 (Erase Type) 00/11：Page Erase 01：Sector Erase 10：Chip Erase (SWD only)
7:2	-	RFU：未实现，读为 0
1	PREQ	Program Request 软件置位，硬件完成编程后自动清零 软件置位后，在编程没有开始的情况下，软件可以清零撤销编程请求
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零 软件置位后，在擦除没有开始的情况下，软件可以清零撤销擦除请求

4.5.4 Flash Key 输入寄存器（FLS_KEY）

名称	FLS_KEY							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	KEY	Flash 擦写 Key 输入寄存器，软件或者 SWD 在启动擦写前必须正确地向此地址写入合法 KEY 序列。 (Flash Key)

4.5.5 Flash 中断使能寄存器（FLS_IER）

名称	FLS_IER							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OTPIE	AUTHIE	KEYIE	CKIE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRDIE	ERDIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	OTPIE	OTP 编程错误中断使能，1 有效 (OTP program error Interrupt Enable)
10	AUTHIE	Flash 读写权限错误中断使能，1 有效 (Flash Authentication Error Interrupt Enable)
9	KEYIE	Flash KEY 错误中断使能，1 有效 (Flash Key Error Interrupt Enable)
8	CKIE	擦写定时时钟错误中断使能，1 有效 (Erase/Program Clock Error Interrupt Enable)
7:2	-	RFU：未实现，读为 0
1	PRDIE	编程完成标志中断使能，1 有效 (Program Done Interrupt Enable)
0	ERDIE	擦写完成标志中断使能，1 有效 (Erase Done Interrupt Enable)

4.5.6 Flash 标志寄存器（FLS_ISR）

名称	FLS_ISR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				KEYSTA			-
位权限	U-0				R-000			U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				OTPERR	AUTHERR	KEYERR	CKERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRD	ERD
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:17	KEYSTA	Flash 擦写 KEY 输入状态 (Flash Key Status) 000：Flash 写保护状态，未输入 KEY 001：全擦解锁状态 010：页擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101：扇区擦解锁状态 110/111：RFU
16:12	-	RFU：未实现，读为 0
11	OTPERR	OTP page 编程权限错误，硬件置位，软件写 1 清零 (OTP program Error) 1：尝试对已编程的 OTP 字节进行编程 0：无 OTP 编程错误
10	AUTHERR	Flash 读写权限错误，读取 LOCK 块数据或对 LOCK 块擦写时置位，软件写 1 清零。(Flash Authentication Error) 1：Flash 访问权限错误 0：Flash 访问没有发生权限错误
9	KEYERR	Flash KEY 错误，硬件置位，复位清零 (Flash Key Error)
8	CKERR	擦写定时时钟错误，NVMIF 擦写 Flash 时如果 RCHF 未使能，则触发 CKERR 中断，软件写 1 清零。(Erase/Program Clock Error)
7:2	-	RFU：未实现，读为 0
1	PRD	Program Done，编程完成标志，硬件置位，软件写 1 清零
0	ERD	Erase Done，擦写完成标志，硬件置位，软件写 1 清零

4.5.7 Flash ECC 标志寄存器（FLS_ECCSR）

名称	FLS_ECCSR							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DD	CD	DC	CC	BE	-		CIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADDR

位权限	U-0							R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADDR							
位权限	R-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR							
位权限	R-0							

位号	助记符	功能描述
31	DD	Data flash ECC 检错标志 (ECC Detection in Data flash)，硬件置位，软件写 1 清零 1：数据 flash 中检测到 2bit 错误，无法纠正 0：没有检测到 2bit 错误
30	CD	Data flash ECC 纠错标志(ECC Correction in Data flash)，硬件置位，软件写 1 清零 1：数据 flash 中纠正 1bit 错误 0：没有错误
29	DC	Code flash ECC 检错标志 (ECC Detection in Code flash)，硬件置位，软件写 1 清零 1：程序 flash 中检测到 2bit 错误，无法纠正 0：没有检测到 2bit 错误
28	CC	Code flash ECC 纠错标志(ECC Correction in Code flash)，硬件置位，软件写 1 清零 1：程序 flash 中纠正 1bit 错误 0：没有错误
27	BE	ECC bit error 标志 当 ECC 检测发现数据正确但是 ECC 位错误时，此寄存器置位此时 ECC 也会产生检错标志，但是 32bit 数据本身仍是正确的可以使用
26	-	RFU：未实现，读为 0
25	-	RFU：未实现，读为 0
24	CIE	ECC 纠错中断使能 1：允许 ECCCD 和 ECCCC 产生中断 0：禁止纠错中断
23:17	-	RFU：未实现，读为 0
16:0	ADDR	ECC 错误地址，只读

4.5.8 Flash ECC 控制寄存器（FLS_ECCCR）

名称	FLS_ECCCR							
offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	CFECCCR							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DFECCCR							
位权限	R/W-0000 0001							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	CFECCCR	Code flash ECC 控制寄存器，物理上只有 1bit 软件写 0xCC 关闭 code flash ECC 功能，写 0xCC 以外任意值使能 code flash ECC 功能，默认 code flash ECC 关闭 软件读取时在 32bit 中的 bit8，bit8=1 表示 ECC 使能，bit8=0 表示 ECC 关闭
7:0	DFECCCR	Data flash ECC 控制寄存器，物理上只有 1bit 软件写 0xCC 关闭 data flash ECC 功能，写 0xCC 以外任意值使能 data flash ECC 功能，默认 data flash ECC 使能 软件读取时在 32bit 中的 bit0，bit0=1 表示 ECC 使能，bit0=0 表示 ECC 关闭

注意：当使用flash ECC功能时，CFECCCR和DFECCCR必须同时使能，并且必须置位CMU_PCLKCR2.NVM_PCE寄存器使能flash控制器时钟

4.5.9 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK1[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK1[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK1[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK1[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK1	ACLOCK 配置寄存器 1，分别用于控制 Block15~Block0 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指

位号	助记符	功能描述
		所有 bit 软件只能写 0，不能写 1。

4.5.10 ACLOCK 寄存器 2（FLS_ACLOCK2）

名称	FLS_ACLOCK2							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK2[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK2[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK2[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK2[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK2	ACLOCK 配置寄存器 2，分别用于控制 Block31~Block16 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

4.5.11 ACLOCK 寄存器 3（FLS_ACLOCK3）

名称	FLS_ACLOCK3							
offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK3[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK3[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK3[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK3[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	LOCK3	ACLOCK 配置寄存器 3，分别用于控制 Block47~Block32 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits) 11：当前 Block 允许 SWD 和软件读写 01/10：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

5 电源管理单元（PMU）

5.1 芯片工作电源

5.1.1 电源域划分

- VDD

芯片的主电源（VDD）的典型工作电压范围是 1.65~5.5V。

芯片上电时，复位释放阈值主要由 POR 电路决定，其典型复位释放电压是 1.6V。

芯片下电时，如果使能了 BOR，下电复位阈值由 BOR 电路决定，可以由软件配置 RMU_BORCR 获得 8 个阈值档位，默认值为 1.8V。如果没有使能 BOR，使能了 PDR，下电复位阈值由 PDR 电路决定，软件可以通过 RMU_PDRCR 配置 4 个档位，默认值 1.4V。

综上，芯片的 VDD 实际工作电压范围将由 BOR 和 PDR 电路配置共同决定。

注意：在任何情况下不得同时关闭 BOR 和 PDR，这样在芯片掉电时可能由于没有产生正常的复位，而导致重新上电时芯片无法正常工作。

- VDDA

VDDA 是专用的模拟电路电源，主要给 ADC、基准电压等模拟模块供电。VDDA 的工作电压范围是 1.65~5.5V，所有模拟模块在这个电压范围内都可以保证正常工作。

- REGC

内部 LDO 输出，外部靠近管脚对地连接 1uF 稳压电容。

5.1.2 电源结构图

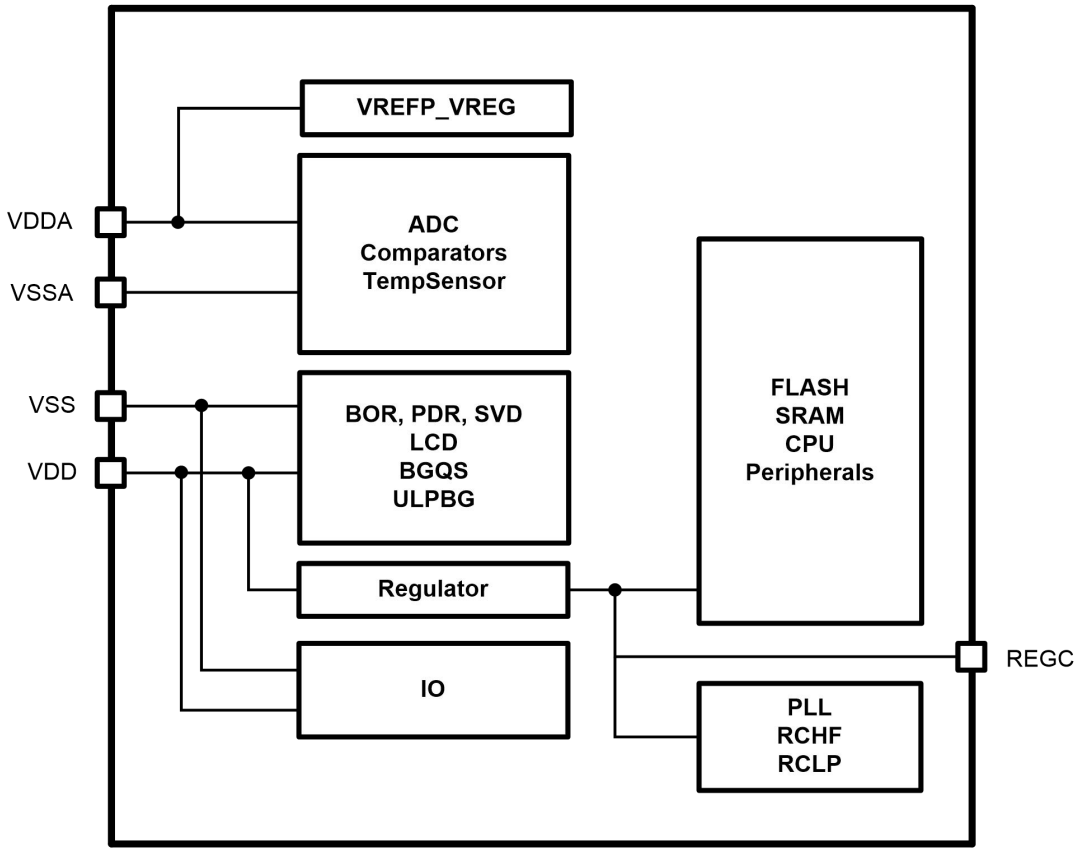


图 5-1 芯片电源结构图

5.1.3 ADC 基准电压

ADC 使用的电源地为 VDD(VDDA)和 VSS(VSSA)，其参考电压是 VREFP、VDDA 或内部产生的 VREFINT。为了提高 ADC 转换精度，降低电源噪声的影响，系统上建议对 VDDA 电源单独滤波，还可以在 PCB 上对 VDDA 和 VSSA 走线进行 shielding，以尽可能屏蔽系统噪声。

在某些低管脚数的封装中，可能没有独立的 VDDA 和 VSSA，在这种情况下，VDDA 和 VDD 在封装内部相连，VSSA 和 VSS 在封装内部相连，由于电源地噪声影响，ADC 的性能可能会有所下降。

外部输入的 VREFN、VREFP 与 PE9、PE10 这两个 GPIO 复用，当需要 ADC 使用外部基准时，必须将 GPIO 配置为模拟通道。

5.1.4 片内快速建立基准源（AVREF）

芯片内部集成一个高精度基准源 AVREF 电压 1.0V，为 LDO、Flash 和 VREFINT_VREG 提供基准电压。

5.1.5 内部 VREFP 产生（VREFINT_VREG）

参见第 6 章。

5.2 功耗模式

5.2.1 概述

上电复位后，芯片默认运行在 ACTIVE 模式，此时 CPU 正常从 flash 取指运行，所有外设模块都可以正常工作。芯片支持多种低功耗模式，软件可以在适当的场景下选择合适的低功耗模式，以平衡不同的功耗、性能、唤醒时间和唤醒条件的要求。

芯片支持的功耗模式：

- ACTIVE 模式：正常运行
- LP Run 模式：LDO 工作在超低功耗模式下，CPU 和外设只能运行在较低频率下
- SLEEP 模式：CPU 停止，Flash 停止，LDO 工作在超低功耗模式下，仅部分外设可以运行
- DEEPSLEEP 模式：CPU 停止，Flash 停止，关闭基准电压，LDO 工作在超低功耗模式下，仅部分外设可以运行

此外，ACTIVE 模式下的运行功耗也可以通过以下手段降低：

- 降低系统时钟频率
- 关闭不使用的外设的总线时钟和工作时钟

功耗模式	典型功耗	唤醒条件	芯片状态	典型唤醒时间[1]
ACTIVE	120uA/MHz		正常工作	-
LP Run	25uA@32KHz	软件主动退出	低速工作	-
SLEEP	3uA	电源检测中断 比较器中断 RTC 定时中断 IO 引脚中断 WKUPx 唤醒 32K 晶振停振 看门狗复位 NRST 引脚复位 LPTIM 中断 BSTIM 中断 等	CPU 休眠 关闭 RCHF、PLL 等 保持 BGQS 开启，关闭 LDO，RTC 走时	6.5us
DEEPSLEEP	1.3uA		CPU 休眠[2] 关闭 RCHF、PLL 等 关闭 BG_QS，关闭 LDO，RTC 走时	6.5us

注：[1] 典型唤醒时间指从唤醒信号到来，到 CPU 开始执行第一条指令的时间间隔。

[2] CPU 自身进入休眠的步骤参见 ARMv6-M 架构参考手册

[3] CPU 试图进入低功耗模式时，如果 Flash 正在擦写，则芯片自动等待 Flash 擦写结束后再进入低功耗模式。

5.2.2 功耗模式与系统频率

在不同功耗模式下，CPU 主频的限制如下图所示：

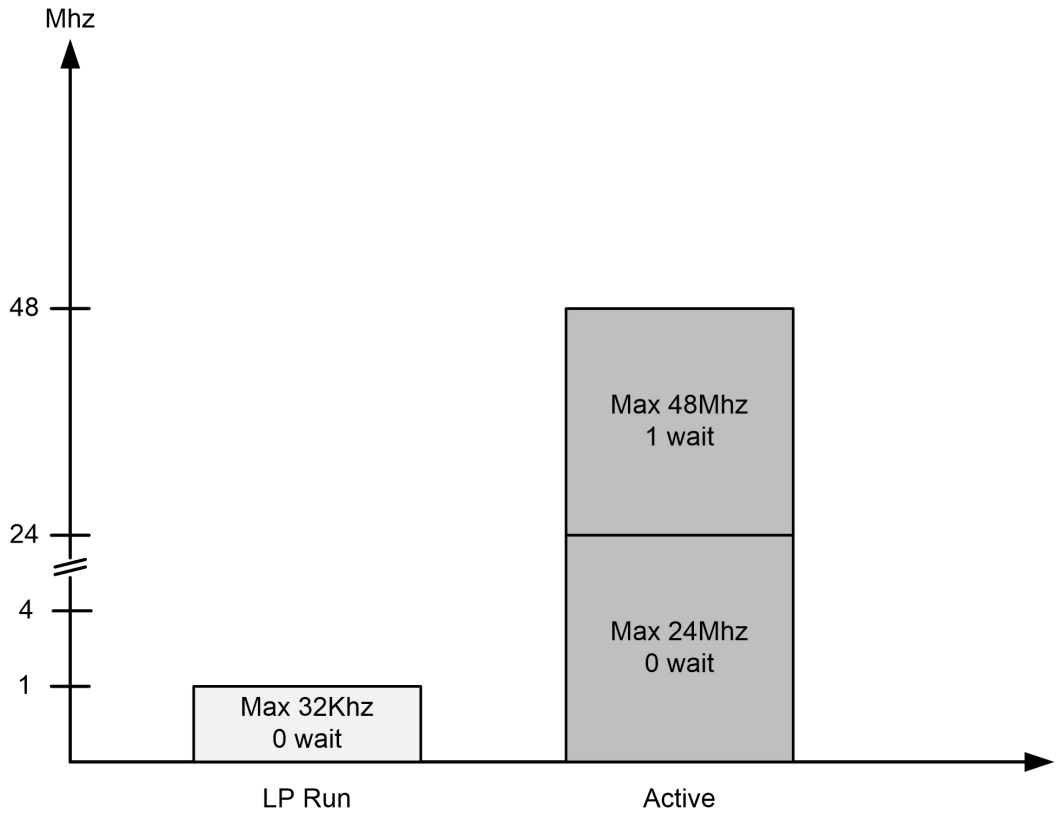


图 5-2 功耗模式与系统主频

不同功耗模式下可以接受的系统频率和可用的时钟源如下表所示。应用软件应严格遵守这个表格的规定，在低功耗模式下使用高主频可能导致系统无法正常运行。

功耗模式	CPU 频率	可用时钟源	Flash wait	外设工作时钟
ACTIVE	<=24MHz	All	0	All
	>24MHz, <=48MHz		1	
LP Run	32.768KHz	XTLF, RCLP	0	XTLF, RCLP

5.2.3 Active 模式

芯片正常工作模式。芯片上电复位完成后进入 Active 模式运行，默认的 CPU 频率是 8MHz，最高

可以运行到 48MHz。在 Active 模式下所有的数字和模拟外设都可以全速运行。

当主频高于 24MHz 时，必须配置 flash wait。

5.2.4 LP Run 模式

当芯片需要低功耗低速运行时，可进入 LP RUN 模式，此时 LDO 进入低功耗模式，内核使用 LSCLK 运行，典型频率 32.768KHz。在需要高速运行时，软件可主动退出 LP RUN 进入 ACTIVE 模式，然后再将系统时钟切换到较高频率。

进入 LPRUN 模式

进入 LPRUN 的操作步骤：

- 软件将系统时钟（SYSCLK）配置为 XTLF 或 RCLP
- 配置 PMOD 寄存器为 01
- 如果系统时钟配置不满足以上寄存器条件，则置位异常中断并且禁止进入 LPRUN

LPRUN 模式下的硬件行为

进入 LP Run 之后，硬件自动关闭 RCHF、XTHF、PLL、TRNG，随后关闭主 LDO。SVD、比较器仍可以在 LPRUN 模式下工作（VDDA 供电）。

如果软件在 LPRUN 模式下执行 WFI 指令，CPU 和 Flash 将停止活动，但是外设仍可以继续工作。

退出 LPRUN 模式

按照以下步骤退出 LPRUN 模式：

- 软件将 PMOD 寄存器配置为 00
- 软件根据需要使能 RCHF、XTHF 或 PLL
- 等待时钟建立后配置系统时钟为 RCHF、XTHF 或 PLL

LP Run 模式下 CPU 改写 PMOD 寄存器可以返回 ACTIVE，或者进入 SLEEP/DEEPSLEEP 模式。如果返回 ACTIVE，硬件自动开启主 LDO，并解除对高速时钟模块的限制。

5.2.5 SLEEP 模式

通过进入 Sleep 模式，可以大幅降低芯片功耗，并处于等待事件唤醒的状态中。

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 指令

SLEEP 模式下的硬件行为

进入 SLEEP 模式后芯片关闭 CPU 时钟，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、XTHF、TRNG。SVD、比较器仍可以在 SLEEP 模式下工作。

数字外设模块可以使用 XTLF、RCLP 等低速时钟继续工作。

退出 SLEEP 模式

按照以下步骤退出 SLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF
- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

注意：芯片退出 SLEEP 时自动打开主 LDO。

5.2.6 DEEPSLEEP 模式

DEEPSLEEP 是芯片最低功耗模式，此模式下由于关闭了内部基准源（AVREF），因此休眠功耗比 SLEEP 进一步降低大约 2uA。

进入 DEEPSLEEP 模式

软件按如下步骤进入 DEEPSLEEP 模式：

- 置位 SLPDP 寄存器
- 配置 PMOD 寄存器为 10
- 执行 WFI 指令

DEEPSLEEP 模式下的硬件行为

DEEPSLEEP 模式下，芯片自动关闭 CPU 时钟，关闭内部基准源，Flash 进入 STOP 模式，硬件自动关闭 RCHF、PLL、TRNG。比较器仍可以在 DEEPSLEEP 模式下工作。

数字外设模块可以使用 XTLF、RCLP 等低速时钟继续工作。

退出 DEEPSLEEP 模式

按照以下步骤退出 DEEPSLEEP 模式：

- 特定的中断事件发生
- 系统时钟被自动配置为 RCHF
- CPU 被唤醒，根据软件配置，唤醒后可以进入或者不进入中断服务程序

5.3 唤醒源

唤醒源	应用	可唤醒模式	
		Sleep	DeepSleep
停振检测	可屏蔽，32768Hz 晶振停振时唤醒芯片	√	√
SVD	可屏蔽，在电源电压跌落至阈值以下或升高至阈值以上时唤醒芯片	√	x
比较器	可屏蔽，用于外部事件唤醒	√	√
RTC	可屏蔽，根据需要的唤醒周期设置	√	√
IO 引脚中断	可屏蔽，用于外部事件唤醒	√	√
Debug	不可屏蔽，用于 debug 唤醒	√	√
UART0/1	可屏蔽，接收数据唤醒	√	√
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√
NRST	不可屏蔽，用于全局复位	√	√
LPTIM16	可屏蔽，用于定时唤醒	√	√
BSTIM16	可屏蔽，用于定时唤醒	√	√
I2C 从机	可屏蔽，用于从机接收唤醒	√	√
LIN 唤醒	可屏蔽，检测到 LIN 唤醒信号时唤醒芯片	√	√

通过Cortex-M0的PRIMASK功能，可以实现以上中断事件唤醒芯片，但是CPU不执行中断处理程序。此时唤醒后CPU将继续从休眠前的指令之后开始运行。

注：芯片从休眠模式唤醒后，软件可以通过查询PMU.WKFR寄存器来快速识别当前的唤醒源，唤醒源的清除需要进入各个外设模块分别完成。

5.4 休眠唤醒后的时钟控制

当芯片从Sleep/DeepSleep模式唤醒后，芯片以RCHF为时钟源。寄存器将保留休眠前RCHF的频率配置和trim值，因此唤醒后CPU运行频率将由休眠前软件配置寄存器决定（PMU_CR.WKFSEL），默认为8MHz。

5.5 寄存器

模块起始地址：0x4000_2000

offset 地址	名称	符号
0x00	低功耗控制寄存器 （Power Management Control Register）	PMU_CR
0x04	唤醒时间控制寄存器 （Wakeup Time Register）	PMU_WKTR
0x08	唤醒源标志查询寄存器 （Wakeup Source Flags Register）	PMU_WKFR
0x0C	PMU 中断使能寄存器 （PMU Interrupt Enable Register）	PMU_IER
0x10	PMU 中断标志寄存器 （PMU Interrupt and Status Register）	PMU_ISR
0x3C	模拟 BUFFER 控制寄存器 （PMU Analog Buffer Control Register）	PMU_BUFCR
0x40	VREFP 控制寄存器 （PMU VREFP Control Register）	PMU_VREFP_CR
0x44	VREFP 配置寄存器 （PMU VREFP Config Register）	PMU_VREFP_CFGR
0x48	-	-
0x4C	VREFP 调校寄存器 （PMU VREFP Trim Register）	PMU_VREFP_TR
0x50	PTAT 控制寄存器 （PMU PTAT Control Register）	PMU_PTAT_CR

5.5.1 低功耗控制寄存器（PMU_CR）

名称	PMU_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	-					AVREF_SH	
位权限	R/W-0	U-0					R/W-01	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				RFUI			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKFSEL		SLPDP	-
位权限	U-0				R/W-00		R/W-0	U-0Z
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	
位名	-				RFUI		PMOD	
位权限	U-0				R/W-00		R/W-00	

Bit	助记符	功能描述
31	**RFUI**	Dummy 寄存器
30:26	**--**	未实现，读为 0
25:24	**AVREF_SH**	AVREF 采样保持周期配置（32K 时钟周期） 00：2ms

Bit	助记符	功能描述
		01：4ms 10：0.5ms 11：1ms
23:20	--	未实现，读为 0
19:16	RFUI	Dummy 寄存器
18:12	--	未实现，读为 0
11:10	WKFSEL	Sleep/DeepSleep 唤醒后的系统频率 00：RCHF-8MHz 01：RCHF-16MHz 10：RCHF-24MHz 11：RFU
9	SLPDP	DeepSleep 控制寄存器 1：DeepSleep 模式使能，下关闭基准电压源 0：常规 Sleep 模式 在 Sleep 下，如果置位了 SLPDP 位即为 DeepSleep 模式； 该位仅在 Sleep 下有效
8:4	--	未实现，读为 0
3:2	RFUI	Dummy 寄存器
1:0	PMOD	低功耗模式配置寄存器 00：Active mode 01：LPRUN mode 10：Sleep mode / DeepSleep mode 11：RFU

5.5.2 唤醒时间控制寄存器（PMU_WKTR）

名称	PMU_WKTR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						T1A	
位权限	U-0						R/W-01	

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1:0	T1A	可编程额外唤醒延迟 在 Sleep/DeepSleep 模式下，RCHF 时钟到来后，根据此寄存器配置等待额外延迟时间后，再读取 Flash 校验字 00：0us 01：2us

Bit	助记符	功能描述
		10：4us 11：8us

5.5.3 唤醒源标志查询寄存器（PMU_WKFR）

名称	PMU_WKFR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	--	UART1 WKF	UART0 WKF	RTCWK F	SVDWK F	LFDET WKF	--	IOWKF
位权限	U-0	R-0	R-0	R-0	R-0	R-0	U-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	I2CSMB 1WKF	I2CSMB 0WKF	--	RFUI	--	--	COMP2 WKF	COMP1 WKF
位权限	R-0	R-0	U-0	R-0	U-0	U-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	--	--	--	LPT16 WKF	--	BST16 WKF	DBGW KF	--
位权限	U-0	U-0	U-0	R-0	U-0	R-0	R-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF							
位权限	R/W-00000000							

位	位名	功能描述
31	--	未实现，读为 0
30	UART1WKF	UART1 中断唤醒标志，中断撤销时硬件自动清零
29	UART0WKF	UART0 中断唤醒标志，中断撤销时硬件自动清零
28	RTCWKF	RTC 中断唤醒标志，中断撤销时硬件自动清零
27	SVDWKF	SVD 中断唤醒标志，中断撤销时硬件自动清零
26	LFDETWKF	32768Hz 晶体停振中断唤醒标志，中断撤销时硬件自动清零
25	--	未实现，读为 0
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零
23	I2CSMB1 WKF	I2C_SMB1 中断唤醒标志，中断撤销时硬件自动清零
22	I2CSMB0 WKF	I2C_SMB0 中断唤醒标志，中断撤销时硬件自动清零
21	--	未实现，读为 0
20	RFUI	Dummy 寄存器
19:18	--	未实现，读为 0
17	COMP2 WKF	比较器 2 中断唤醒标志，中断撤销时硬件自动清零
16	COMP1 WKF	比较器 1 中断唤醒标志，中断撤销时硬件自动清零
15:13	-	未实现，读为 0
12	LPT16WKF	LPTIM16 中断唤醒标志，中断撤销时硬件自动清零
11	-	未实现，读为 0
10	BST16WKF	BSTIM16 中断唤醒标志，中断撤销时硬件自动清零
9	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零
8	-	未实现，读为 0
7:0	WKPxF	NWKUPx Pin 唤醒标志，软件写 1 清零

5.5.4 PMU 中断使能寄存器（PMU_IER）

名称	PMU_IER							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SLPEIE	LPREIE
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1	SLPEIE	SLEEP 错误中断使能 1：使能 SLEEP 错误中断 0：禁止 SLEEP 错误中断
0	LPREIE	LPRUN 错误中断使能 1：使能 LPRUN 错误中断 0：禁止 LPRUN 错误中断

5.5.5 PMU 中断标志寄存器（PMU_ISR）

名称	PMU_ISR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SLPEIF	LPREIF
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1	SLPEIF	SLEEP 错误中断标志，硬件置位，软件写 1 清零 1：在 PMOD=0x02 后，CPU 执行 WFI 指令前置位了 SLPDP 寄存

Bit	助记符	功能描述
		器时置位 0：在 PMOD=0x02 后，CPU 正确进入 SLEEP
0	**LPREIF**	LPRUN 错误中断标志，硬件置位，软件写 1 清零；软件进入 LPRUN 模式时如果触发了 LPREIF，则芯片仍将停留在 ACTIVE 模式 1：LPRUN Condition Error，即进入 LPRUN 时满足如下情况： 1） 系统时钟不是 LSCLK，或 2） RCHF 使能未关闭 0：LPRUN 正常进入

5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR）

名称	PMU_BUFCR							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		AVREFBUF_OUTEN	AVREFBUF_EN	VPTATBUFFER_OUTEN	VPTATBUFFER_EN	-	
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	U-0	

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5	AVREFBUF_OUTEN	AVREF 输出缓冲输出使能，ADC 采样 AVREF 时建议使能
4	AVREFBUF_EN	AVREF 输出缓冲使能，ADC 采样 AVREF 时建议使能
3	VPTATBUFFER_OUTEN	Vptat Buffer 模块开关通道输出使能信号，高电平使能有效。(PTAT Buffer Output Enable)
2	VPTATBUFFER_EN	Vptat Buffer 模块使能信号，高电平使能有效。(PTAT Buffer Enable)
1:0	-	RFU：未实现，读为 0

5.5.7 VREFP 控制寄存器（PMU_VREFP_CR）

名称	PMU_VREFP_CR							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	VREFINT_VREG 使能寄存器 (enable) 0：关闭并 bypass VREFINT_VREG 1：使能 VREFINT_VREG

5.5.8 VREFP 配置寄存器（PMU_VREFP_CFGR）

名称	VREFP_CFGR							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						VRS	
位权限	U-0						R/W-00	

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1:0	VRS	输出电压选择 (Voltage regulation select) 00：VREFINT 输出 1.5V 01：VREFINT 输出 2.0V 10：VREFINT 输出 2.5V 11：VREFINT 输出 3.0V

5.5.9 VREFINT 调校寄存器（PMU_VREFINT_TR）

名称	PMU_VREFINT_TR							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TRIM	VREFP 输出电压调校 (trimming) 每 LSB 对应步长 0.1%，最多可以加 127steps 或者减 127steps，对应调校范围-12.7% ~ +12.7%

5.5.10 PTAT 控制寄存器（PMU_PTAT_CR）

名称	PMU_PTAT_CR							
地址	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							PTAT_EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	PTAT_EN	温度传感器使能 0：关闭温度传感器 1：使能温度传感器 注：芯片测量温度前，需要置位 PTAT_EN 和 VPTATBUF_EN，等待输出建立时间后，启动 ADC 采样并计算温度。

6 VREFINT 基准电压（VREFINT_VREG）

6.1 概述

FM33FR0可以使用VREFP、VDDA作为基准电压，或者由片内电路产生多档可配置基准电压。

内部基准电压由VREFINT_VREG电路产生，VREFINT_VREG使用1.0V AVREF为参考电压，获得3.0V、2.5V、2.0V、1.5V输出。

VREFINT_VREG产生的基准电压，可以给内部电路使用（如ADC、比较器等）。

6.2 功能描述

6.2.1 概述

VREFINT_VREG电路可以用于产生恒定的内部基准电压，此时可以使ADC的基准电源不依赖于外部电源，因此也就不受外部电源波动的影响。

ADC的正端基准可以在VREFP、VDDA、内部基准之间切换；负端基准可以在VREFN、VSSA之间切换。

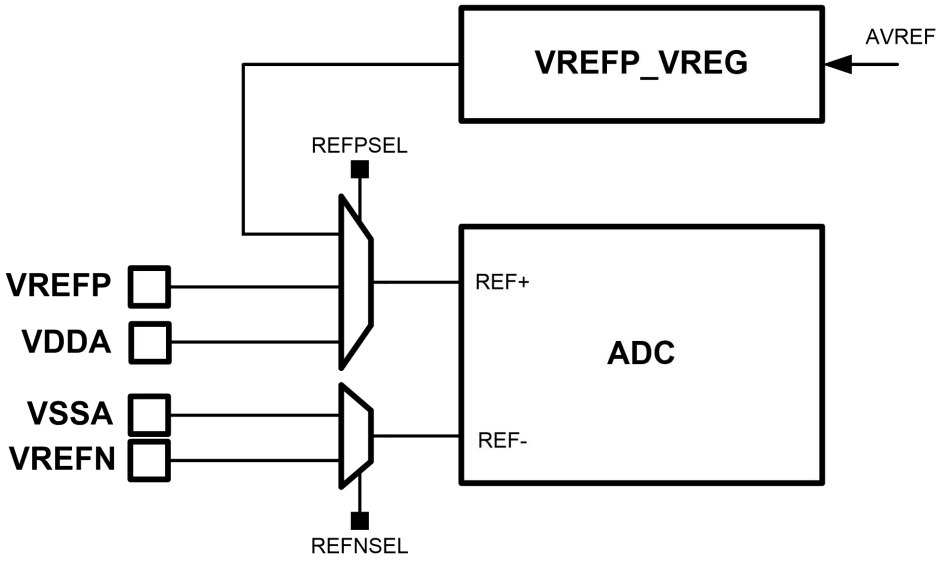


图 6-1 ADC 基准源

系统方案上可以同时支持外部基准和内部基准。

6.2.1.1 外部电源为基准

当使用外部电源作为基准时，连接关系如下图。

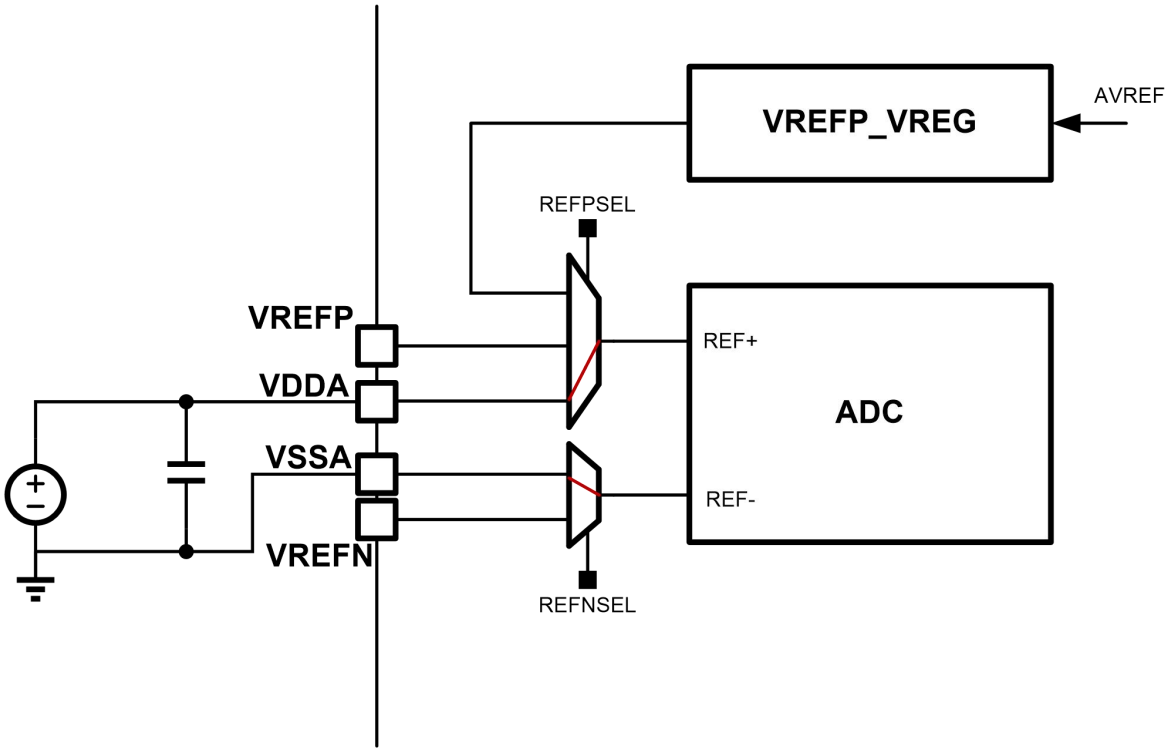


图 6-2 外部电源基准方案

6.2.1.2 外部输入独立基准

也可以支持系统上提供一个独立于电源的基准电压，此时ADC的基准源选择从VREFP引脚提供。注意芯片的VREFP和VREFN管脚与GPIO复用，使用时必须将GPIO配置为模拟通道。

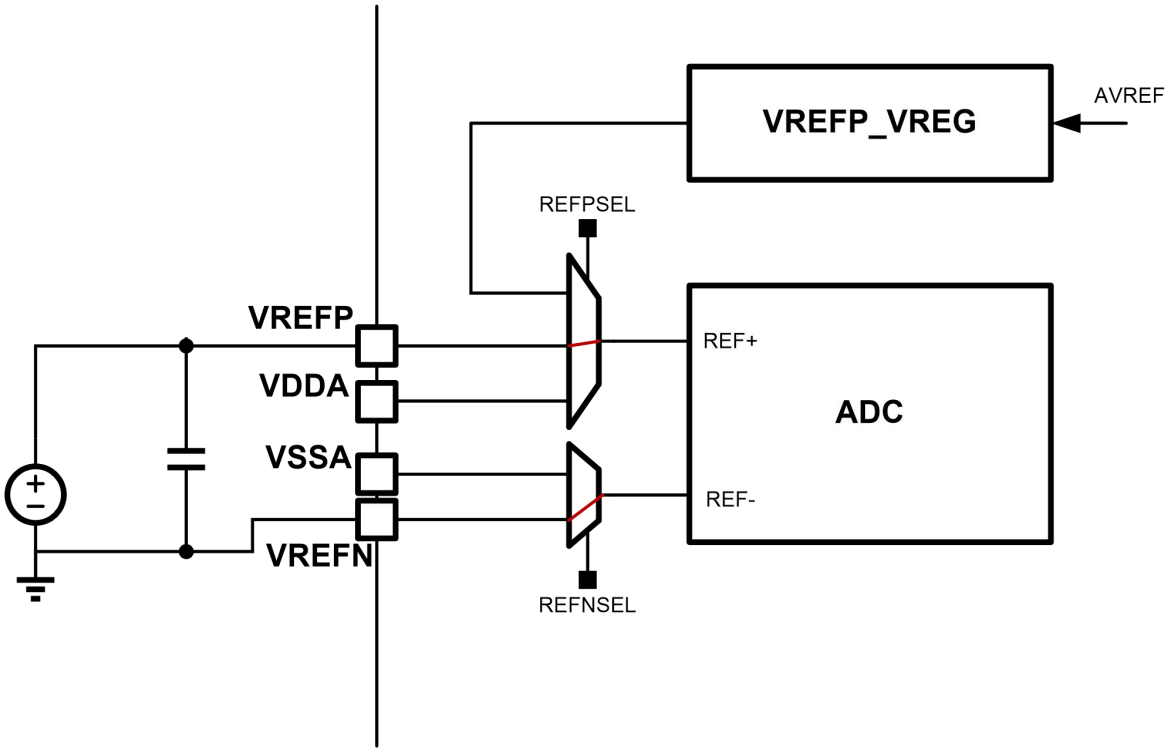


图 6-3 外部独立基准方案

6.2.1.3 内部基准

下图显示了ADC使用内部基准源的方案。如果在使用内部基准的同时，希望测量VREFN/ADC_IN15引脚上的输入信号，则必须将ADC负基准连接到VSSA。

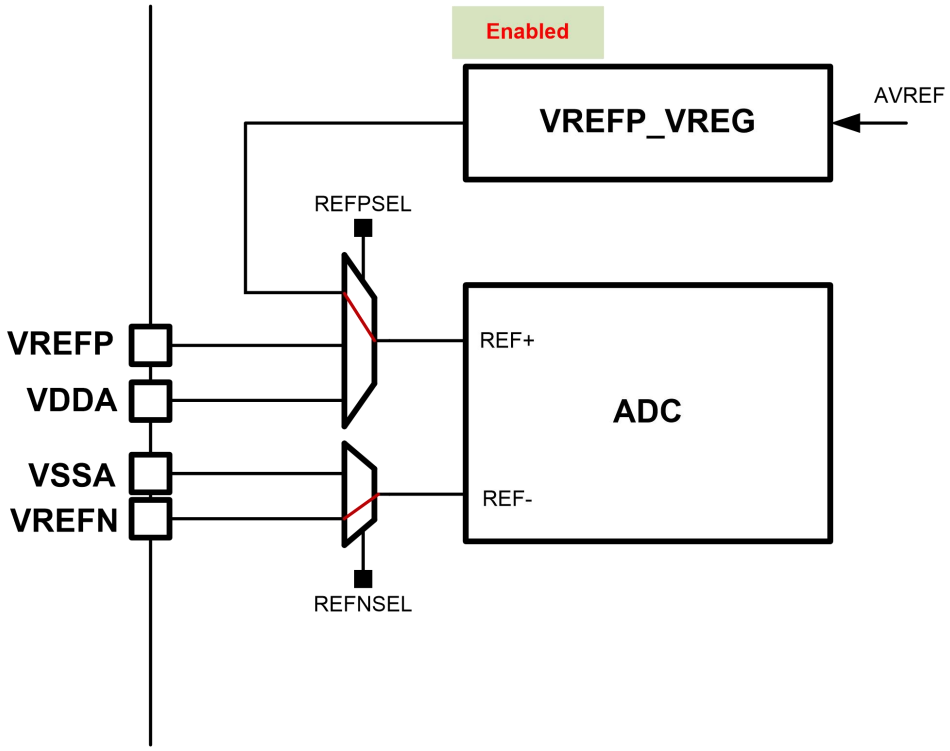


图 6-4ADC 使用内部基准方案的系统连接

6.2.2 参考电压

VREFINT_VREG的参考电压来自于AVREF，因此使用内部产生的VREFINT时不能进入休眠模式。

6.2.3 输出电压

VREFINT_VREG支持4个输出档位，并且支持至少+/-10%的输出调校范围，调校步长为0.1%，即trim后可以实现+/-0.05%的输出精度。

VREFINT_VREG的输出可以作为ADC基准电压，并可作为比较器的参考电压输入。

6.3 寄存器

参见PMU章节

7 权限管理（MDC）

7.1 寄存器

7.1.1 HardFault 查询寄存器

名称	HDFFLAG							
地址	0x40000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	SVCUNDEF_FLAG	BKPT_FLAG	TBIT_FLAG	DFLASH_FLAG	HDF_REQUEST_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	DABORT_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 1：总线传输中访问了非法地址导致 HRESP 为高产生错误 0：未访问非法地址
4	SVCUNDEF_FLAG	SVC instructions 未定义标志，写 1 清零 if the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set, the core should treat SVC instructions as though theywere UNDEFINED。
3	BKPT_FLAG	执行 BKPT 指令标志，写 1 清零 1：执行了 BKPT 指令 0：未执行 BKPT 指令
2	TBIT_FLAG	Thumb-State 标志，写 1 清零 1：切换到 ARM 状态 0：处于 Thumb-State
1	DFLS_FLAG	DFLASH 取指标志，写 1 清零 1：执行了特殊指令代码，如试图在 DFLASH 区域内取指 0：无指令代码试图在 DFLASH 被执行
0	HDF_REQU	hardfault 标志位，任何类型的 hardfault 都会导致该位置位，写 1 清零

Bit	助记符	功能描述
	EST_FLAG	1：hardfault 请求 0：无 hardfault 请求

8 处理器（CPU）

8.1 概述

FM33FR0 使用的 CPU 内核为 Cortex-M0，符合 ARMv6-M 架构和编程模型；更多信息请参考 ARM 官网 www.arm.com

其基本特性如下：

- 用户/特权模式
- VTOR（中断向量表重定向）
- NVIC 支持 32 个外部中断
- 数据监视点：1
- 硬件断点：4
- 单周期 32-bit 硬件乘法器
- SWD 调试接口

8.1.1 处理器配置

Feature	Options	FM33FR0Config
Interrupts	1~32	32
Data endianness	little/big	little
SysTick Timer	Present or absent	Present
watchpoints	0,1,2	1
breakpoints	0,1,2,3,4	4
halting debug support	Present or absent	Present
multiplier	Fast or Small	Fast
Single-Cycle IO	Present or absent	Absent
wake-up interrupt controller(WIC)	Present or absent	Present
Vector Table Offset Register	Present or absent	Present
Unprivileged/Priviledged support	Present or absent	Present
JTAGnSW	JTAG or SWD for DAP	SWD
Memory Protection Unit	Present or absent	Absent

表 8-1FM33FR0 CPU 配置简表

8.2 寄存器

主要内核寄存器列表

名字	描述
R0-R12	通用寄存器
MSP (R13)	堆栈指针；Handler 模式下使用 MSP（Main Stack Pointer），Thread 模式下通过 CONTROL 寄存器选择 MSP 或 PSP（Process Stack Pointer）使用
PSP (R13)	
LR (R14)	Link 寄存器，保存子函数/函数调用/异常处理的返回信息
PC (R15)	程序指针
PSR	包含应用程序状态（APSR）、中断程序状态（IPSR）和程序执行状态（EPSR）
PRIMASK	PRIMASK 用于屏蔽指定优先级及以下的所有中断响应
CONTROL	设置 Thread 模式下使用的堆栈指针

表 8-2 Cortex-M0 内核寄存器简表

寄存器详细定义参见 ARMv6-M 架构参考手册。

8.3 异常和中断

内核的异常和中断管理通过 NVIC 完成。NVIC 的可编程管理寄存器位于 PPB 总线的 SCS 空间内，NVIC 具有如下特性：

- 支持 32 个外部中断，5 个内部异常
- 1 个 NMI 中断
- 支持中断嵌套
- 向量化的异常入口
- 中断屏蔽

处理器内核接受一个异常请求后，首先会将内核寄存器 R0~R3、R12、R14、PC、xPSR 压入堆栈。链接寄存器 LR（R14）被更新为异常返回时使用的特殊值（EXC_RETURN），然后根据异常向量表定位异常处理程序开始执行。注意在异常处理中没有被自动压栈的寄存器，必须通过软件来保存和恢复。

8.3.1 中断向量表

参见9.2中断向量表

8.3.2 中断优先级

处理器支持 3 个固定的最高优先级及 4 个可编程优先级。当两个相同优先级的异常同时发生，则异常编号较小的异常将被首先执行。

8.3.3 错误处理

处理器只支持一种硬件错误处理方式：HardFault 异常。HardFault 优先级-1，只有 NMI 能对其抢占。

HardFault 的触发原因包含以下几种情况：

错误类型	错误条件
存储器相关	总线错误。由于在总线传输中使用了非法地址而产生的总线错误。
	试图在 XN 区域内执行程序
程序错误	执行未定义的指令
	试图切换至 ARM 状态
	试图进行非对齐的存储器访问 在更高优先级异常处理中执行 SVC 指令 执行异常返回时 EXC_RETURN 的值非法 当调试未使能时试图执行 BKPT 指令

FM33FR0 的 HardFault 触发原因可以通过寄存器查询，以帮助软件开发者定位错误原因。

8.3.4 锁定（Lockup）

当处理器在进行 HardFault 处理的过程中发生了另一个 HardFault，或者 NMI 处理期间发生了 HardFault，则处理器将进入锁定状态（停止执行），并输出 LOCKUP 信号，此时芯片将自动复位处理器内核，而不是等待看门狗溢出。

8.4 调试特性

处理器支持以下调试特性

- 程序的暂停、恢复及单步执行
- 访问内核寄存器和特殊寄存器
- 硬件断点（4 个）
- 软件断点（不限数量的 BKPT 指令）
- 数据监视点（1 个）
- 动态非侵入式存储器访问（无需停止处理器）
- SWD 接口

Cortex-M0 的调试特性是基于 ARM CoreSight 调试架构的，详情请参考《CoreSight Technology System Design Guide》和《ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2》

8.4.1 调试功能引脚

FM33FR0 使用 SWD 调试接口，用户模式下最少仅需 4 线（NRST, GND, SWIO, SWCLK）即可实现调试功能。2 线调试引脚可以复用为 GPIO，其功能由软件选择配置。

NRST 引脚用于复位芯片，通过 NRST 与 SWD 的配合，可以使芯片复位后 Halt 在第一条指令处。

调试功能引脚的复用说明参见 I/O 控制章节。

8.4.2 调试状态下的看门狗控制

看门狗在调试模式下可以保持使能或关闭。软件或 Debugger 可以通过 DBG_CR 寄存器配置看门狗打开或关闭。

8.4.3 DEBUG 的复位

内核的 DEBUG 部分不受引脚复位影响 ，这样可以在芯片上电后通过引脚复位使 CPU 内核处于复位状态，但是调试器仍可以正常与 DAP 建立通信并设置断点，在复位放开后可以使 CPU 立即进入调试模式。

建议调试器在系统复位时连接内核(在复位向量处设置断点)。

9 中断源多路选择（INTMUX）

9.1 概述

CPU 部分中断入口所连接的中断源可以由软件选择，以实现灵活的中断优先级，方便软件开发。

9.2 中断向量表

CPU 中断向量表如下，

Position	Priority	Priority type	Acronym	Description	Address
0	-	-	MSP 初值	主栈指针初始化地址	0x0000_0000
1	-3	fixed	Reset	复位向量	0x0000_0004
2	-2	fixed	NMI	WKUPx 中断 低功耗模式错误中断 存储器非法访问中断 Flash ECC 检错中断	0x0000_0008
3	-1	fixed	HardFault	HardFault 中断向量	0x0000_000C
4-10	-	-	-	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall 系统服务请求	0x0000_002C
12-13	-	-	-	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	可挂起系统服务请求	0x0000_0038
15	6	settable	Systick	内部定时器中断向量	0x0000_003C
16	7	settable	WDT	窗口看门狗或独立看门狗中断	0x0000_0040
17	8	settable	SVD	电源监测报警中断	0x0000_0044
18	9	settable	RTC	实时时钟中断	0x0000_0048
19	10	settable	MUX0	可配置多路中断	0x0000_004C
20	11	settable	TSI	按键触摸检测中断	0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable	-	Reserved	0x0000_0058
23	14	settable	MUX1	可配置多路中断	0x0000_005C
24	15	settable	MUX2		0x0000_0060
25	16	settable	MUX3		0x0000_0064
26	17	settable	MUX4		0x0000_0068
27	18	settable	MUX5		0x0000_006C
28	19	settable	MUX6		0x0000_0070
29	20	settable	MUX7		0x0000_0074
30	21	settable	MUX8		0x0000_0078
31	22	settable	I2CMx	I2C master 中断	0x0000_007C
32	23	settable	-	Reserved	0x0000_0080
33	24	settable	I2C_SMBx	I2C SMBUS 中断	0x0000_0084
34	25	settable	MUX9	可配置多路中断	0x0000_0088

Position	Priority	Priority type	Acronym	Description	Address
35	26	settable	AES	AES 中断	0x0000_008C
36	27	settable	LPTIM	LPTIM16 中断	0x0000_0090
37	28	settable	DMA	DMA 中断	0x0000_0094
38	29	settable	WKUPx	WKUP 引脚中断	0x0000_0098
39	30	settable	MUX10	可配置多路中断	0x0000_009C
40	31	settable	BSTIM	BSTIM16 中断	0x0000_00A0
41	32	settable	COMPx	COMPx 中断	0x0000_00A4
42	33	settable	MUX11	可配置多路中断	0x0000_00A8
43	34	settable	MUX12	可配置多路中断	0x0000_00AC
44	35	settable	ATIM	高级定时器中断	0x0000_00B0
45	36	settable	MUX13	可配置多路中断	0x0000_00B4
46	37	settable	EXTI	外部引脚中断	0x0000_00B8
47	38	settable	ECCC RAMP FDET	Flash ECC 纠错中断 RAM 奇偶校验错误中断 XTLF 或 XTHF 停振检测中断	0x0000_00BC

表 9-1 中断向量表

其中WKUPx中断可以接到NMI或者38#入口。通过GPIO模块的PINWKEN.WKSEL寄存器来选择中断入口地址。当配置为38#入口时，可以通过PRIMASK将WKUPx中断屏蔽，唤醒后CPU不进入中断服务程序，而是继续从休眠指令处向下执行。

9.3 中断向量多路选择

中断向量表中的 MUX 项目表示可以由寄存器选择多个中断源接入。多路映射关系如下表所示。

	SEL=00	SEL=01	SEL=10	SEL=11
MUX0	FLASH	TAU00	TAU01	TAU02
MUX1	SPI0	TAU10	TAU11	TAU06
MUX2	SPI1	TAU12	TAU01	TAU02
MUX3	UART0	TAU10	TAU11	TAU01
MUX4	UART1	TAU00	TAU01	TAU10
MUX5	UART2	TAU03	TAU04	TAU11
MUX6	UART3	TAU12	TAU13	TAU02
MUX7	UART4	TAU12	TAU13	TAU04
MUX8	UART5	TAU05	TAU06	TAU12
MUX9	CCL	TAU05	TAU06	TAU03
MUX10	LUT	TAU03	TAU04	TAU07
MUX11	SPI2	TAU11	TAU04	TAU03
MUX12	SPI3	TAU13	TAU05	TAU07
MUX13	TAU00	UART1	UART2	-

表 9-2 中断源多路选择表

9.4 寄存器

模块起始地址：0x4001_6000

offset 地址	名称	符号
0x00	中断源多路选择控制寄存器 （Interrupt Multiplexer Control Register）	INTMUX_CR

9.4.1 中断源多路选择控制寄存器（INTMUX_CR）

名称	INTMUX_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				MUX13SEL		MUX12SEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MUX11SEL		MUX10SEL		MUX9SEL		MUX8SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MUX7SEL		MUX6SEL		MUX5SEL		MUX4SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MUX3SEL		MUX2SEL		MUX1SEL		MUX0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

Bit	助记符	功能描述
31:28	--	未实现，读为 0
27:26	MUX13SEL	寄存器定义参见表 9-2 中断源多路选择表
25:24	MUX12SEL	
23:22	MUX11SEL	
21:20	MUX10SEL	
19:18	MUX9SEL	
17:16	MUX8SEL	
15:14	MUX7SEL	
13:12	MUX6SEL	
11:10	MUX5SEL	
9:8	MUX4SEL	
7:6	MUX3SEL	
5:4	MUX2SEL	
3:2	MUX1SEL	
1:0	MUX0SEL	

10 复位管理单元（RMU）

10.1 概述

复位电路特点：

- 支持多个复位源，如上下电复位、看门狗复位、软件复位、引脚复位等
- 上下电复位（BOR）监控主电源供电
- POR 上电复位典型释放电压 1.53V
- BOR 下电复位产生电压软件可配置，支持 8 个档位，可关闭。
- 低功耗下电复位电路（PDR），下电复位电压可配置为 1.4/1.45/1.5/1.55V，可关闭
- 上下电复位信号经过去抖动和延时，抗干扰能力强

进入复位状态时，所有寄存器都恢复到初始值（除 RTC 内部寄存器）；退出复位状态时，MCU 使用内部 RC 振荡器（RCHF，默认频率 8MHz）作为系统时钟。

10.2 模块框图

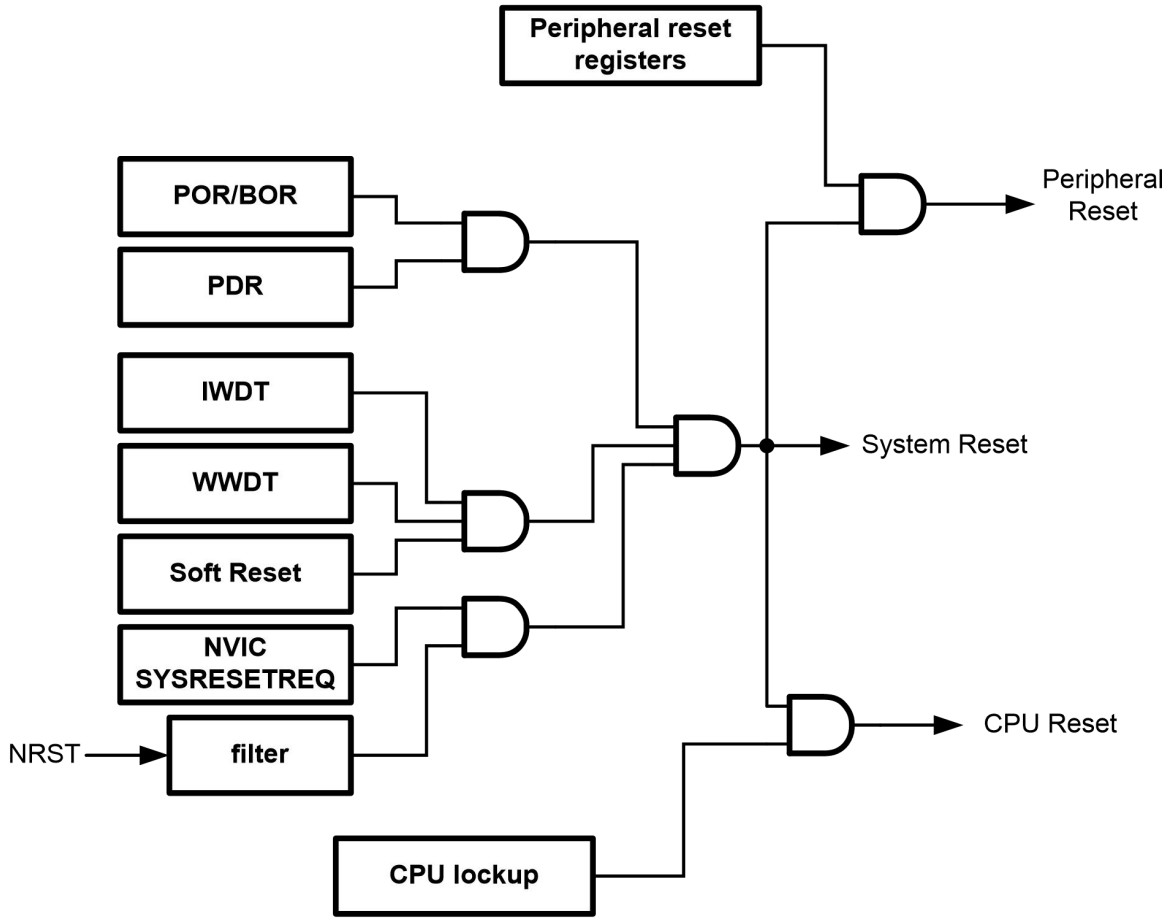


图 10-1 芯片复位源框图

10.3 上下电复位

上下电复位电路监控 VDD 电源，由 BOR、POR 和低功耗 PDR 组成。

- POR：提供上电复位
- BOR：高精度可编程下电复位，8 个档位
- PDR：超低功耗下电复位

为防止电源抖动，保证上电复位电路的抗干扰能力，对上电复位信号进行滤波和延时处理。

BOR下电复位阈值软件可配置，共8档：1.8V、2.0V、2.2V、2.4V、2.7V、3V、3.6V、4V，注意电源回升时有200mV左右的回滞窗口。

芯片上电复位放开后，BOR下电复位默认是关闭的，软件可以开启。

注意：BOR是一个下电复位电路，软件使能BOR之前，必须确保芯片电源已经上升到准备使用的BOR档位阈值之上，否则使能BOR将导致直接复位。比如，上电复位释放后，如果VDD=3V，而软件希望设置BOR阈值为3.6V，则此时软件不应使能BOR，否则使能后将立即导致复位。建议软件在使能BOR之前，先通过ADC或者SVD检测芯片电源是否达到预期电压。

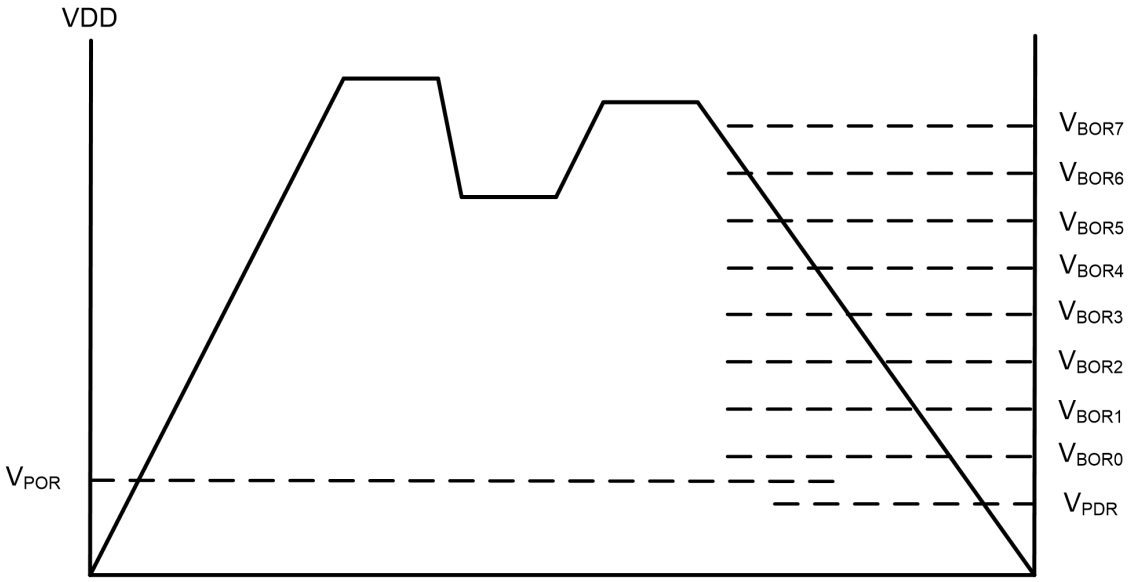


图 10-2 上下电复位示意图

假设上电后软件使能BOR并配置档位3，VDD变化导致芯片系统复位的波形示意图如下

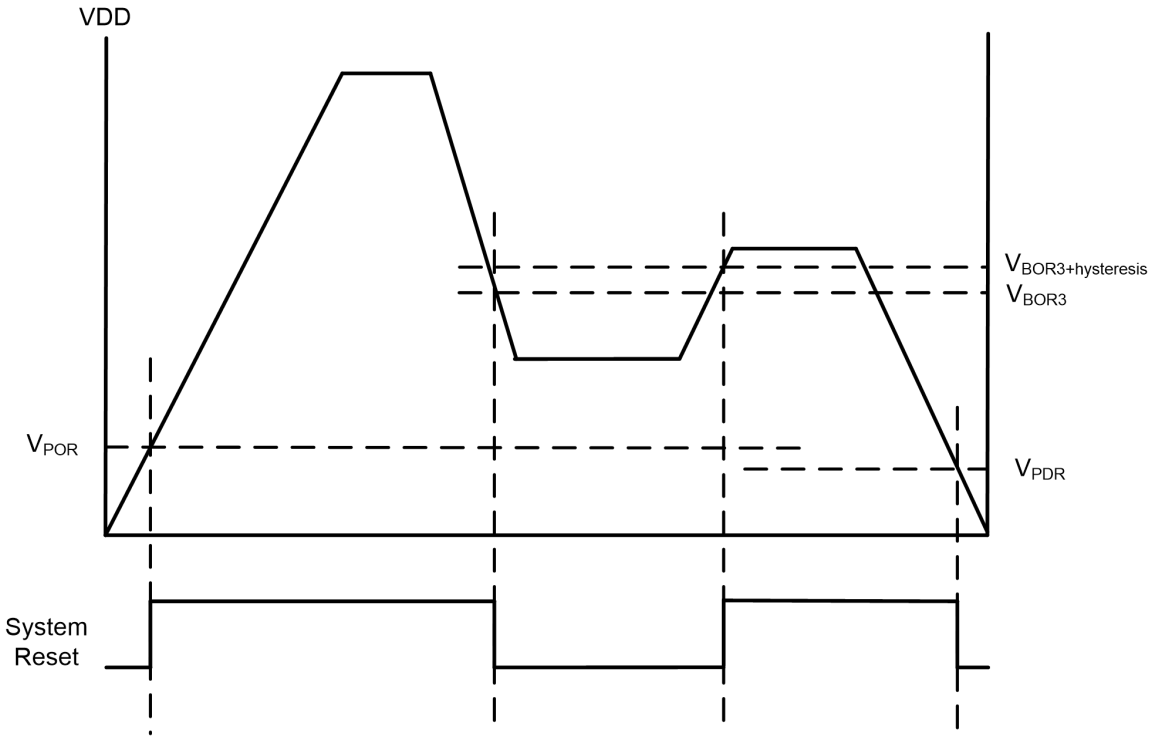


图 10-3 POR, BOR, PDR 复位波形示意图

上电复位滤波时间决定了电源上升的最短时间要求，为了保证安全的上电复位，要求上电速度不能快于2us/V，即电源从0上升到5V，用时不能短于10us。

10.4 软件复位

软复位由 CPU 写寄存器发起，操作方式为向 SOFTRST 寄存器写 0x5C5C_AABB。

另外也可以通过操作 Cortex 内核 SCB 模块的 AIRCR 寄存器，置位 SYSRESETREQ bit 的方式来实现全局复位。

10.5 NRST 引脚复位

NRST 引脚可以复用为 GPIO（PD10），系统复位后此引脚默认为 NRST 功能，软件可以通过配置 FCR 寄存器将其修改为 GPIO，此后 NRST 复位功能失效。详情参见“I/O 端口”章节。

10.6 寄存器

模块起始地址：0x40002800

offset 地址	名称	符号
0x00	PDR 配置寄存器	RMU_PDRCR
0x04	BOR 配置寄存器	RMU_BORCR
0x08	复位配置寄存器	RMU_LKPCR
0x0C	软件复位寄存器	RMU_SOFTRST
0x10	复位标志寄存器	RMU_RSTFR
0x14	外设复位控制寄存器	RMU_PRSTEN
0x18	AHB 外设复位寄存器	RMU_AHBRSTCR
0x1C	APB 外设复位寄存器 1	RMU_APBRSTCR1
0x20	APB 外设复位寄存器 2	RMU_APBRSTCR2

10.6.1 PDR 配置寄存器（RMU_PDRCR）

名称	RMU_PDRCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		EN
位权限	U-0					R/W-00		R/W-1

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2:1	**CFG**	下电复位电压配置 00—1.4V 01—1.45V 10—1.5V 11—1.55V
0	**EN**	下电复位使能 0：关闭下电复位 1：使能下电复位

10.6.2 BOR 配置寄存器（RMU_BORCR）

名称	RMU_BORCR

offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CFG			-	ENB
位权限	U-0			RW-000			U-0	R/W-1

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:2	CFG	BOR 复位电压配置 000：falling 1.8V reset, rising 2.0V release 001：falling 2.0V reset, rising 2.2V release 010：falling 2.2V reset, rising 2.4V release 011：falling 2.4V reset, rising 2.6V release 100：falling 2.7V reset, rising 2.9V release 101：falling 3.0V reset, rising 3.2V release 110：falling 3.6V reset, rising 3.8V release 111：falling 4.0V reset, rising 4.2V release
1	--	RFU：未实现，读为 0
0	ENB	BOR 下电复位使能，上电后默认关闭 0：使能 BOR 下电复位 1：关闭 BOR 下电复位

10.6.3 复位配置寄存器（RMU_LKPCR）

名称	RMU_LKPCR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						EN	-
位权限	U-0						R/W-0	U-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	EN	LOCKUP 复位使能 1：使能 LOCKUP 复位 0：屏蔽 LOCKUP 复位
0	--	RFU：未实现，读为 0

10.6.4 软件复位寄存器（RMU_SOFTRST）

名称	RMU_SOFTRST							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SOFTRST							
位权限	W							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SOFTRST							
位权限	W							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFTRST							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SOFTRST							
位权限	W							

Bit	助记符	功能描述
31:0	SOFTRST	软件写 0x5C5C_AABB 触发全局复位

10.6.5 复位标志寄存器（RMU_RSTFR）

名称	RMU_RSTFR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	RAMP_FLAG	MAP_FLAG	MDF_FLAG	NRSTN_FLAG	PRCN_FLAG	PORN_FLAG	PDRN_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TESTN_FLAG	SOFTN_FLAG	IWDTN_FLAG	-	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:15	--	RFU：未实现，读为 0
14	**RAMP_FLAG**	RAM 奇偶校验错误复位标志，高有效，软件写 1 清零
13	**MAP_FLAG**	非法地址访问复位标志，高有效，软件写 1 清零
12	**MDF_FLAG**	模式诊断超时复位标志，高有效，软件写 1 清零
11	**NRSTN_FLAG**	NRST 引脚复位标志，高有效，软件写 1 清零
10	**PRCN_FLAG**	PRC 复位标志，高有效，软件写 1 清零
9	**PORN_FLAG**	上电复位标志，高有效，软件写 1 清零
8	**PDRN_FLAG**	下电复位标志，高有效，软件写 1 清零
7	--	RFU：未实现，读为 0
6	**TESTN_FLAG**	TESTN 引脚复位标志，高有效，软件写 1 清零
5	**SOFTN_FLAG**	软件复位标志，高有效，软件写 1 清零
4	**IWDTN_FLAG**	IWDT 复位标志，高有效，软件写 1 清零
3	--	RFU：未实现，读为 0
2	**WWDTN_FLAG**	WWDT 复位标志，高有效，软件写 1 清零
1	**LKUPN_FLAG**	LOOKUP 复位标志，高有效，软件写 1 清零
0	**NVICN_FLAG**	NVIC 复位标志，高有效，软件写 1 清零

10.6.6 外设复位使能寄存器（RMU_PRSTEN）

名称	RMU_PRSTEN							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PERHRSTEN[31:24]							
位权限	W							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PERHRSTEN[23:16]							
位权限	W							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PERHRSTEN[15:8]							
位权限	W							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PERHRSTEN[7:0]							
位权限	W							

Bit	助记符	功能描述
31:0	PERHRSTEN	外设模块复位使能，32bit 虚寄存器，只写 软件对此地址写 0x1357_9BDF，使能外设复位功能，此后可以通过外设模块复位寄存器复位各个模块 软件对此地址写任意其他数据，将关闭外设复位功能

10.6.7 AHB 外设复位寄存器（RMU_AHBRSTCR）

名称	RMU_AHBRSTCR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							DMARST
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	DMARST	DMA 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

10.6.8 APB 外设复位寄存器 1（RMU_APBRSTCR1）

名称	RMU_APBRSTCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	UART3RST	UART2RST	UART1RST	UART0RST	UARTIRRST	RFU
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SPI3RST	SPI2RST	SPI1RST	SPI0RST	-			
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	INTMUXRST	LCDRST	I2CSMB1RST	I2CSMB0RST	I2C3RST	I2C2RST	I2C1RST	I2C0RST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TSIRST	-	PGLRST	ATIMRST	TAU1RST	TAU0RST	LPT16RST	BT16RST
位权限	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	UART5RST	UART5 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
30	UART4RST	UART4 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
29	UART3RST	UART3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

Bit	助记符	功能描述
28	UART2RST	UART2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
27	UART1RST	UART1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
26	UART0RST	UART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
25	UARTIRRST	UART 红外调制模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
24	RFU	Dummy 寄存器
23	SPI3RST	SPI3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22	SPI2RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
21	SPI1RST	SPI1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	SPI0RST	SPI0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
19:16	--	RFU：未实现，读为 0
15	INTMUXRST	INTMUX 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
14	LCDRST	LCD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
13	I2CSMB1RST	I2CSMB1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
12	I2CSMB0RST	I2CSMB0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
11	I2C3RST	I2C3 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
10	I2C2RST	I2C2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
9	I2C1RST	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
8	I2C0RST	I2C0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位

Bit	助记符	功能描述
		1：复位
7	TSIRST	TSI 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
6	--	RFU：未实现，读为 0
5	PGLRST	PGL 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
4	ATIMRST	ATIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
3	TAU1RST	TAU1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	TAU0RST	TAU0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
1	LPT16RST	LPTIM16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	BT16RST	BT16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

10.6.9 APB 外设复位寄存器 2（RMU_APBRSTCR2）

名称	RMU_APBRSTCR2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ADCCRST
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	-				AESRST	CRCRST	RNGRST
位权限	R/W-0	U-0				R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				HDIVRST	-	SVDRST	COMPRST
位权限	U-0				R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:25	--	RFU：未实现，读为 0
24	ADCCRST	ADC 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

Bit	助记符	功能描述
23	ADCRST	ADC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22:19	--	RFU：未实现，读为 0
18	AESRST	AES 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
17	CRCRST	CRC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
16	RNGRST	RNG 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
15:2	--	RFU：未实现，读为 0
3	HDIVRST	HDIV 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	--	RFU：未实现，读为 0
1	SVDRST	SVD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	COMPRST	COMP 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

11 独立看门狗（IWDT）

11.1 概述

独立看门狗用于监视系统运行，如果 CPU 运行异常，无法定时清狗，则看门狗在溢出后产生全局复位信号，重启系统，以避免系统锁死。独立看门狗的使能由 OPTBYTES 控制，芯片出厂默认使能看门狗，上电后硬件自动装载，软件无法使能或关闭 IWDT。

IWDT 使用独立时钟工作（RCLP），RCLP 上电后自动开启，且不能关闭。

为了便于调试，在以下情况下 IWDT 会停止运行：

- 当芯片处于调试模式时，软件可以通过配置 DBG_CR 寄存器在调试过程中暂停 IWDT
- 当 OPTBYTES 中 IWDTSLP 有效时，软件可以在休眠模式下暂停 IWDT 计数

IWDT 核心是一个 12bit 向上计数器，复位后从 0 开始递增，计数到 0xFFF 后触发 IWDT 复位。IWDT 复位是一个全局复位，效果等同于上下电复位。

IWDT 使用 RCLP 工作，确保不会停止工作。IWDT 带有除 128 预分频器，计数器长度为 12bit。

IWDT 支持可编程窗口功能，软件只能在允许的窗口内清狗，窗口外清狗将触发 IWDT 复位。

11.2 结构框图

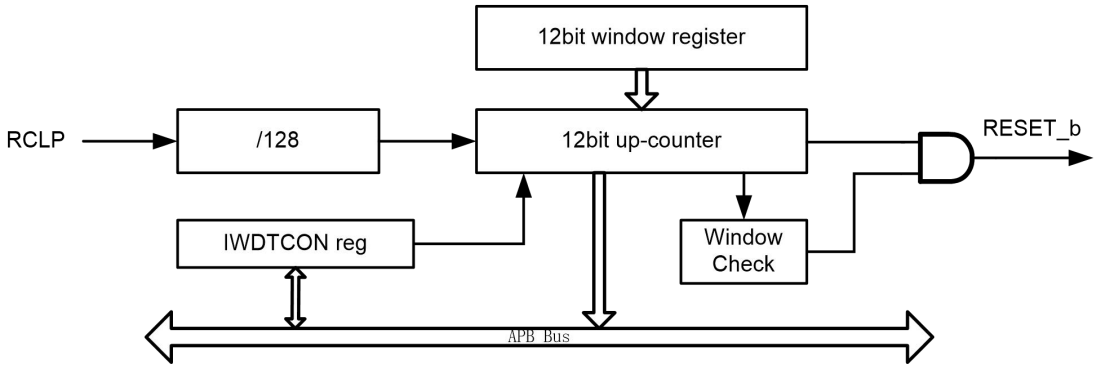


图 11-1 IWDT 结构框图

11.3 IWDT 功能描述

CPU 正常运行时，看门狗应使用较短的溢出周期，而在 SLEEP/DEEPSLEEP 等低功耗模式下，为了使芯片尽可能长时间的停留在低功耗模式下，则看门狗应使用较长的溢出周期。

为了兼容两者的不同应用需求，软件可以实时修改 IWDT 的溢出周期配置。为避免不当操作引发不可预计的后果，软件在更新溢出周期配置时应遵循以下操作步骤：

- 确保看门狗正在运行
- 首先进行一次清狗操作
- 随后改写 IWDT_CR 寄存器，选择合适的溢出周期
- 读 IWDT_CR，确保写入正确
- 溢出周期更新完毕，CPU 正常运行

IWDT 使用 RCLP 工作，内部预分频 128，分频后的计数器溢出长度可配置为 1~4096（共 8 个可用档位），溢出时间长度计算公式如下：

tWWDT = T_{RCLP} * 128 * IWDTOVP

RCLP 频率	溢出长度配置	溢出时间（ms）
32.768KHz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

表 11-1 IWDT 溢出周期表

11.4 IWDT 窗口功能

IWDT支持可编程清狗窗口功能。IWDT_WIN寄存器用于定义允许的清狗窗口，只有当计数器计数值大于等于IWDT_WIN的值时，清狗操作才是合法的，在窗口之外清狗将会直接出发IWDT复位。

芯片复位后IWDT_WIN为全0，即默认允许软件在任何位置清狗。

软件可以在IWDT运行过程中实时修改IWDT_WIN寄存器。软件清狗时必须读取并确认当前计数值是否在允许清狗的范围内。

当IWDT计数值进入清狗窗口时，IWDT会触发一个中断标志寄存器，通知软件当前计数值已经进入清狗窗口。

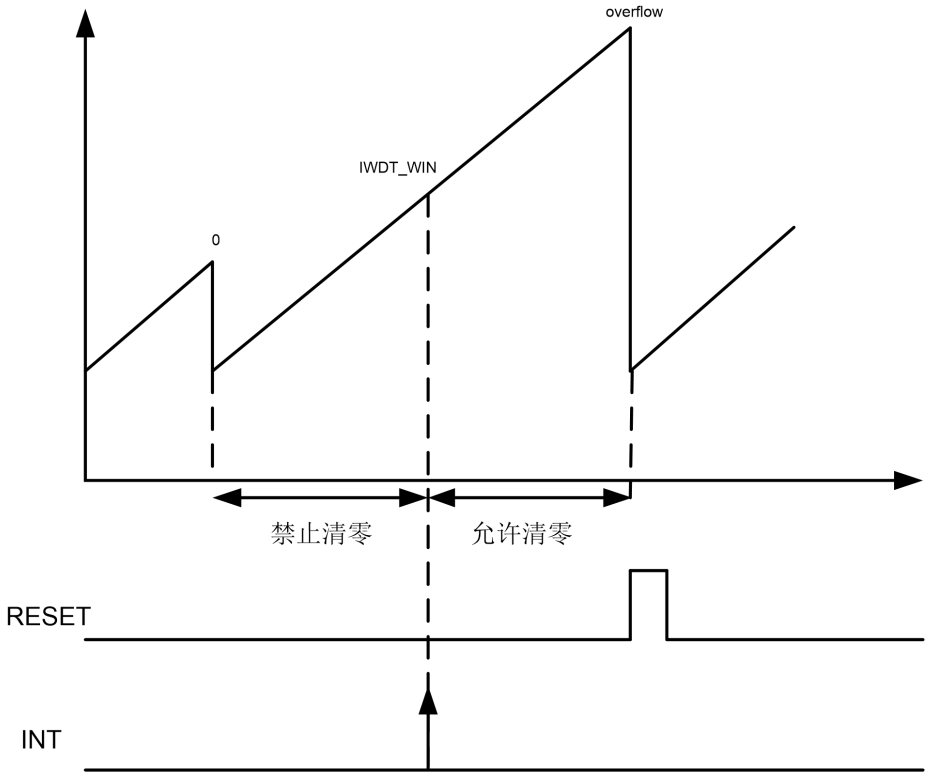


图 11-2 IWDT 窗口示意图

11.5 IWDT 冻结

用户可以通过OPTBYTES配置是否允许IWDT在休眠模式下冻结计数。

当OPTBYTES中的IWDTSLP有效，并且软件将IWDT_FREEZE寄存器置位时，当芯片进入Sleep/DeepSleep模式，IWDT计数值自动冻结（注意不是关闭IWDT，只是计数值保持当前值不再递增）。

11.6 寄存器

模块起始地址：0x4001_1400

offset 地址	名称	符号
0x00	IWDT 清除寄存器 （IWDT Service Register）	IWDT_SERV
0x04	IWDT 配置寄存器 （IWDT Config Register）	IWDT_CR
0x08	IWDT 计数值寄存器 （IWDT Counter Register）	IWDT_CNT
0x0C	IWDT 窗口寄存器 （IWDT Window Register）	IWDT_WIN
0x10	IWDT 中断使能寄存器 （IWDT Interrupt Enable Register）	IWDT_IER
0x14	IWDT 中断标志寄存器 （IWDT Interrupt Status Register）	IWDT_ISR

11.6.1 IWDT 清除寄存器（IWDT_SERV）

名称	IWDT_SERV							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SERV[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SERV[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SERV[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SERV[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	**SERV**	上电后 IWDT 状态受 LDT1 区域 IWDTEN 控制。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

11.6.2 IWDT 配置寄存器（IWDT_CR）

名称	IWDT_CR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				FREEZE	-		
位权限	U-0				R/W-0	U-0		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					OVP		
位权限	U-0					R/W-001		

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	FREEZE	IWDT 休眠冻结，仅在 OPTBYTES 中 IWDTSLP 配置有效时起作用(Freeze in Sleep Enable) 1：Sleep/DeepSleep 模式下冻结 IWDT 计数 0：Sleep/DeepSleep 模式下保持 IWDT 运行
10:3	-	RFU：未实现，读为 0
2:0	OVP	配置 IWDT 看门狗溢出时间 (Overflow Period) 000：125ms 001：250ms 010：500ms 011：1s 100：2s 101：4s 110：8s 111：16s

11.6.3 IWDT 计数值寄存器（IWDT_CNT）

名称	IWDT_CNT							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	CNT	IWDT 当前计数值，软件只读 (IWDT Counter Value,read only) 由于计数器工作时钟与 APB 总线为异步关系，软件读取计数值时应连续读取 2 次以上，为相同值时才认为是稳定结果

11.6.4 IWDT 窗口寄存器（IWDT_WIN）

名称	IWDT_WIN							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WIN[11:8]			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WIN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	WIN	IWDT 窗口寄存器 (IWDT Window)

11.6.5 IWDT 中断使能寄存器（IWDT_IER）

名称	**IWDT_IER**							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	IWDT 中断使能 (IWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

11.6.6 IWDT 中断标志寄存器（IWDT_ISR）

名称	**IWDT_ISR**
Offset	0x14

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							WINF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WINF	IWDT 进入窗口中断标志，写 1 清零 (Window Flag,write 1 to clear) 0：无中断产生 1：计数值进入清狗窗口

12 窗口看门狗（WWDT）

12.1 功能描述

带窗口的看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位全芯片，避免不可预计的后果。

WWDT 在芯片上电后默认关闭，软件启动 WWDT 后，不能再关闭，直到下一次复位。低功耗休眠模式下 WWDT 停止运行。

为了保证同步性和实时性，WWDT 使用 CPU 时钟工作，内部有一个预分频电路，以产生同步计数使能信号。

在以下情况时 WWDT 产生 CPU 复位：

- 计数器溢出
- 对 WWDT 清零寄存器写 0xAC 以外的值（可用于触发 CPU 软复位）
- 在窗口关闭期内对 WWDT 清零寄存器写 0xAC

当计数器达到溢出时间的 75%时，会触发一个预警中断。

12.2 结构框图

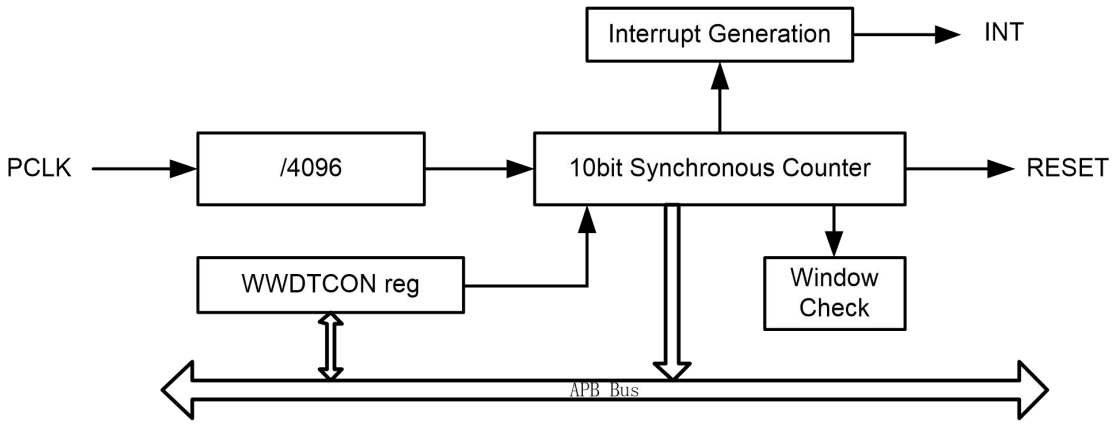


图 12-1WWDT 结构框图

12.3 WWDT 工作方式

WWDT 在芯片复位后默认关闭，软件需对 WWDT_CR 寄存器写入 0x5A 来启动 WWDT。WWDT 启动后，如果软件在 open window 内对 WWDT_CR 写 0xAC，将清零计数器。WWDT 一旦使能后不能关闭，直到下一次复位，WWDT 复位发生后将会关闭 WWDT。

WWDT 使用 PCLK 工作，内部预分频 4096，分频后的计数器溢出长度可配置为 1~1024（共 8 个可用档位），溢出时间长度计算公式如下：

$tWWDT = T_{APBCLK} * 4096 * N_{CFG}$

下表为计算示例：

APBCLK 频率	溢出长度配置	溢出时间（ms）
48MHz	1	0.085
	4	0.341
	16	1.365
	64	5.461
	128	10.922
	256	21.845
	512	43.69
	1024	87.38
32MHz	1	0.128
	4	0.512
	16	2.048
	64	8.192
	128	16.384
	256	32.768
	512	65.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	131.072
	1024	262.144
8MHz	1	0.512
	4	2.048
	16	8.192
	64	32.768
	128	65.536
	256	131.072
	512	262.144

APBCLK 频率	溢出长度配置	溢出时间（ms）
	1024	524.288

表 12-1 WWDT 溢出周期表

WWDT 只允许在 open window 内进行清除，否则将直接触发复位。使能窗口为计数器的后半周期，软件在清零看门狗之前应注意查询计数值。

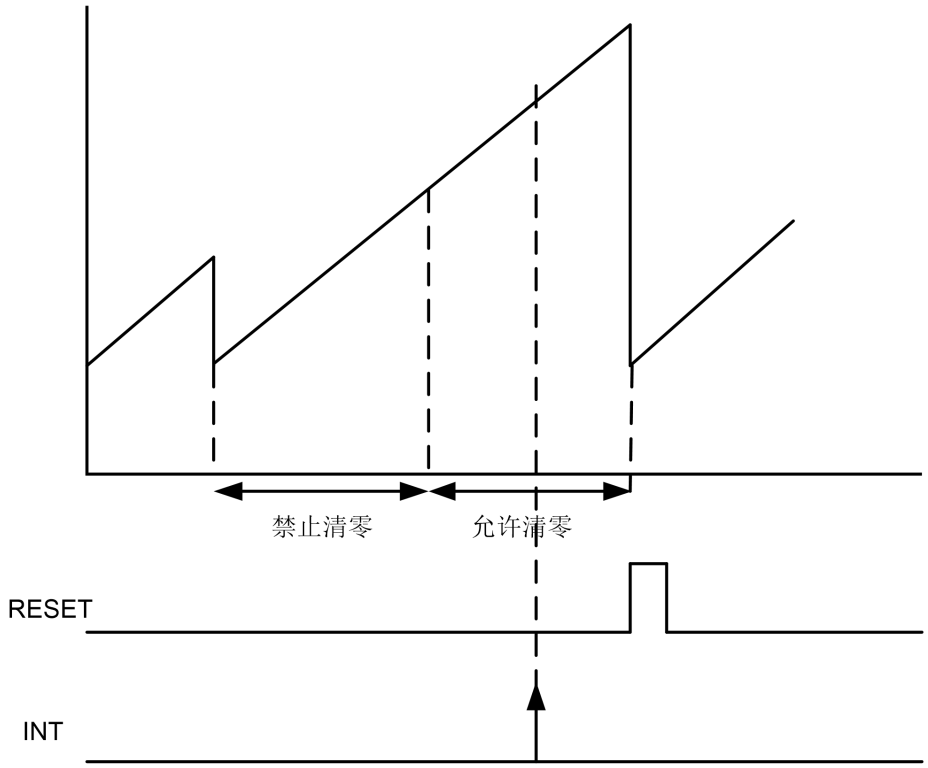


图 12-2WWDT 窗口示意图

12.4 寄存器

模块起始地址：0x4001_1800

offset 地址	名称	符号
0x00	WWDT 控制寄存器 （WWDT Control Register）	WWDT_CR
0x04	WWDT 配置寄存器 （WWDT Config Register）	WWDT_CFGR
0x08	WWDT 计数值寄存器 （WWDT Counter Register）	WWDT_CNT
0x0C	WWDT 中断使能寄存器 （WWDT Interrupt Enable Register）	WWDT_IER
0x10	WWDT 中断标志寄存器 （WWDT Interrupt Status Register）	WWDT_ISR
0x14	WWDT 预分频寄存器 （WWDT Prescaler Register）	WWDT_PSC

12.4.1 WWDT 控制寄存器（WWDT_CR）

名称	WWDT_CR							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CON							
位权限	W-0000 0000							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	CON	当 CPU 向此地址写入 0x5A 时启动 WWDT 定时器 (WWDT Control,write only) 在启动 WWDT 后，当 CPU 向此地址写入 0xAC 时清零计数器

12.4.2 WWDT 配置寄存器（WWDT_CFGR）

名称	WWDT_CFGR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CFG		
位权限	U-0					R/W-011		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	CFG	配置 WWDT 看门狗溢出时间，复位值 011，由于上电后系统时钟默认为 8Mhz，所以默认溢出周期大约 32ms (WWDT Config) 000：T_{PCLK} * 4096 * 1 001：T_{PCLK} * 4096 * 4 010：T_{PCLK} * 4096 * 16 011：T_{PCLK} * 4096 * 64 100：T_{PCLK} * 4096 * 128 101：T_{PCLK} * 4096 * 256 110：T_{PCLK} * 4096 * 512 111：T_{PCLK} * 4096 * 1024

12.4.3 WWDT 计数寄存器（WWDT_CNT）

名称	WWDT_CNT							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CNT[9:8]	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:0	CNT	WWDT 计数寄存器值，软件可通过查询此寄存器了解 WWDT 计时进度 (WWDT Counter value,read only)

12.4.4 WWDT 中断使能寄存器（WWDT_IER）

名称	WWDT_IER
Offset	0x0C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IE
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	IE	WWDT 中断使能 (WWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

12.4.5 WWDT 中断标志寄存器（WWDT_ISR）

名称	WWDT_ISR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							NOVF
位权限	U-0							R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	NOVF	WWDT 75%计时中断标志，写 1 清零 (Near Overflow Flag,write 1 to clear) 0：无中断产生 1：中断标志置位 如果 IE=1，则此寄存器置位将触发中断

12.4.6 WWDT 预分频寄存器（WWDT_PSC）

名称	WWDT_PSC
Offset	0x14

名称	WWDT_PSC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				DIV_CNT[11:8]			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIV_CNT[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11:0	DIV_CNT	WWDT 的 4096 预分频计数器当前计数值，只读 (WWDT prescaler Divider Counte,read only)

13 时钟管理单元（CMU）

13.1 概述

芯片内包含32.768KHz低频晶体振荡电路(XTLF)、4~24MHz高频晶体振荡器、最高32MHz高频RC振荡器(RCHF)、32.768KHz低功耗内部环振（RCLP）、高频晶体振荡器和一个锁相环（PLL）。芯片内部的时钟产生模块整合这些时钟源，产生各个模块工作所需要的时钟。

特点：

- 系统主时钟可选多个时钟源
- 时钟可在系统运行中实时切换
- 低频晶体振荡器配备停振检测电路
- 部分外设模块独立工作时钟（与 CPU 和总线时钟解耦）
- CPU 和总线最高频率 48MHz

13.2 时钟架构

13.2.1 时钟树

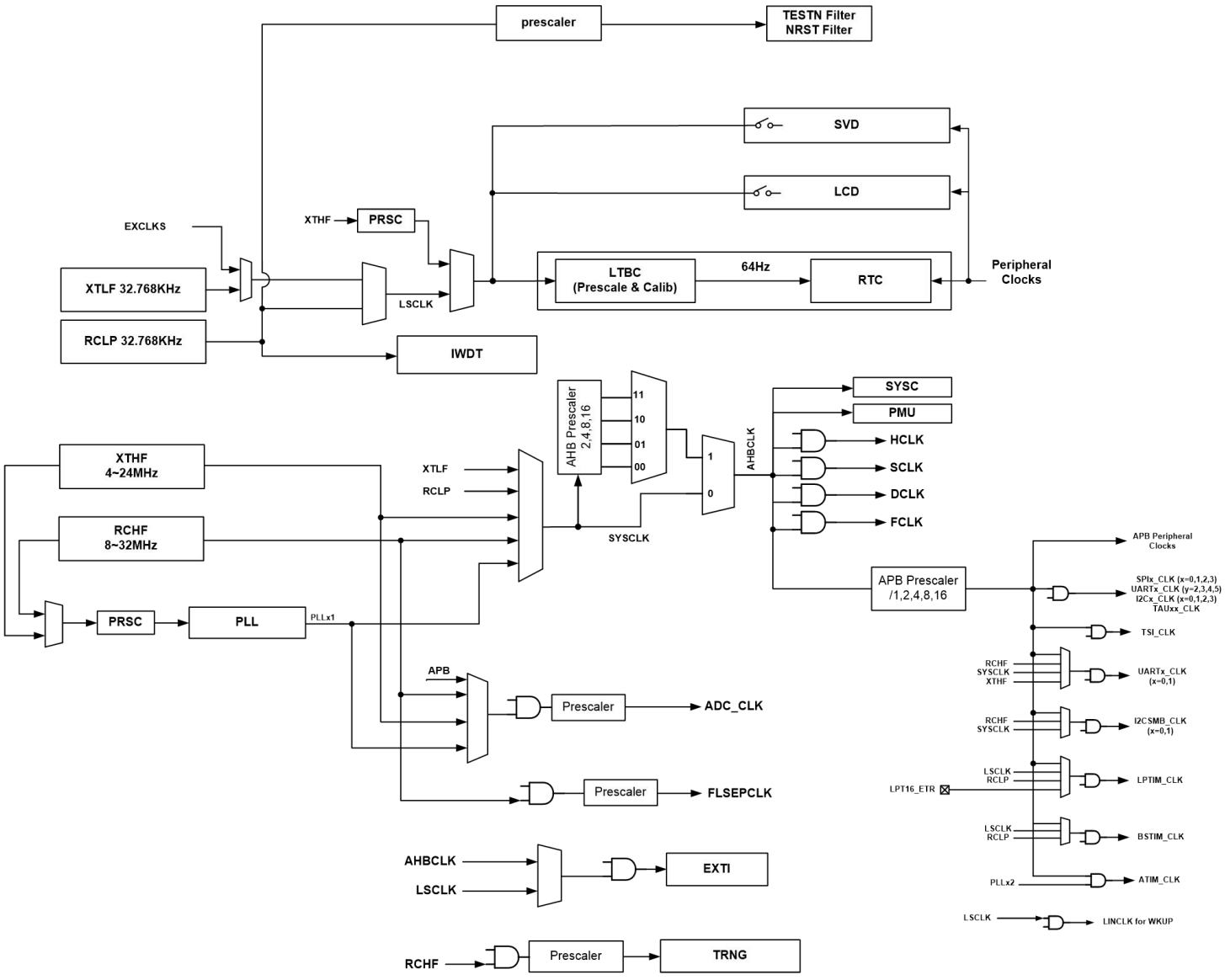


图 13-1 芯片时钟框图

系统主时钟(SYSCLK)可由XTLF、RCHF、XTHF、PLL、RCLP及它们的分频时钟产生。上电默认使用8MHzRCHF的不分频时钟作为系统主时钟，各外设模块的时钟可以分别独立控制。芯片工作时可以只打开需要工作的模块时钟，其他模块的时钟可关闭，以节省功耗。APB总线时钟APBCLK可以是AHBCLK的分频或同频时钟，用于驱动APB总线上的低速外设。

13.2.2 SYSCLK 切换说明

SYSCLK是系统主时钟，从SYSCLK可以得到AHBCLK和APBCLK等总线时钟以及CPU运行所需的时钟。

SYSCLK选择任何一个时钟源时，硬件都要检查对应时钟源是否开启，如果时钟源没有被使能（或发生停振），则软件切换操作无效，SYSCLKSEL寄存器不会被改写，时钟切换也不会发生，同时SYSCKE_IF中断标志寄存器置位，可以产生中断事件。

目标时钟	切换条件
RCHF	RCHF 使能
XTHF	XTHF 使能且未停振
PLL	PLL 使能，并且： 1， 如果 PLL 参考时钟是 XTHF，则 XTHF 必须使能且未停振 2， 如果 PLL 参考时钟是 RCHF，则 RCHF 必须使能
XTLF	XTLF 未停振
RCLP	-

13.2.3 时钟安全性

当使用XTLF、XTHF为系统时钟时，或者使用PLL为系统时钟并且PLL参考时钟为XTHF时，必须考虑时钟安全性。

在以上条件下，如果发生时钟停振，停振检测输出将强制启动RCHF，并将SYSCLK切换逻辑复位成RCHF通路。

- SYSCLK为XTLF，当XTLF停振时自动将SYSCLK切换到RCHF
- SYSCLK为XTHF，当XTHF停振时自动将SYSCLK切换到RCHF
- SYSCLK为PLL并且PLL参考时钟选择为XTHF，当XTHF停振时自动将SYSCLK切换到RCHF

以上切换发生的同时，停振检测电路会产生停振中断，通知软件处理异常。

13.2.4 主要时钟说明

时钟	源头	说明
LSCLK	XTLF, RCLP	32.768KHz 低频系统时钟，晶体停振时可以切换到 RCLP 主要用于 RTC、IWDT、SVD、LCD
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF	32K~48MHz，经过分频后得到 AHBCLK
HCLK(AHBCLK)	SYSCLK	AHB 总线时钟，用于驱动 CPU、RAM、Flash

时钟	源头	说明
		和高速外设
SCLK	SYSCLK	CPU 内核系统时钟
DCLK	SYSCLK	CPU 内核 Debug 时钟（当仿真器连接时这个时钟必须活动）
FCLK	SYSCLK	Free-Running 时钟，提供给 CPU 内核 WIC 模块，以及 APB 桥
APBCLK	AHBCLK	APB 总线时钟，用于驱动低速外设

13.2.5 外设模块的总线时钟和工作时钟

部分外设模块的总线时钟和工作时钟互相独立。

其中总线时钟用于AHB或APB总线访问，在软件访问外设的功能寄存器时，必须先通过外设总线时钟控制寄存器来使能对应的总线时钟。

而外设的工作时钟为外设实际工作使用的时钟，这个时钟可能不同于APBCLK或AHBCLK，外设模块工作前，需要通过外设工作时钟寄存器来选择所需的时钟源，并打开时钟门控。

而对于工作时钟和总线时钟统一的外设模块，则仅需使能总线时钟就可以正常工作了。

模块	总线时钟	工作时钟
独立工作时钟外设		
UARTx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
		XTHF
		RCHF
I2C_SMBUSx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
ATIM	APBCLK	APBCLK
		PLLx2_CLK
LPTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		LPT16_ETR
BSTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
ADC	APBCLK	XTHF
		RCHF

模块	总线时钟	工作时钟
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APBCLK	RCHF
IWDT	APBCLK	RCLP
LCD	APBCLK	LSCLK
RTC	APBCLK	LSCLK
非独立工作时钟外设		
PMU	AHBCLK	
DMA	AHBCLK	
TSI	APBCLK	
UARTx (y=2,3,4,5)	APBCLK	
SPIx (x=0,1,2,3)	APBCLK	
I2C master x (x=0,1,2,3)	APBCLK	
TAU	APBCLK	
HDIV	APBCLK	
AES	APBCLK	
CRC	APBCLK	
WWDT	APBCLK	
PGL	APBCLK	
COMPx (x=0,1)	APBCLK	

13.2.6 休眠模式下的外设时钟

Sleep/DeepSleep模式下，SYSCLK被关闭，因此在休眠模式下AHBCLK和APBCLK都不工作，所有基于AHBCLK或APBCLK的外设都停止工作。但是，使用独立与总线时钟工作的外设仍可以继续工作，比如RTC、LPTIM16、BSTIM16。

为了让上述外设在休眠模式下继续工作，软件需要在休眠前确保上述外设使用SYSCLK和总线时钟以外的时钟工作。

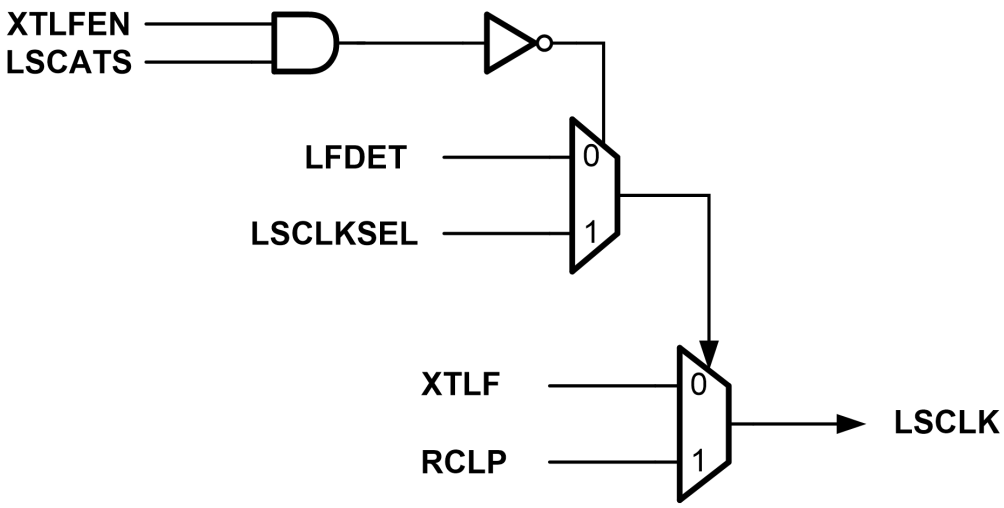
13.2.7 LSCLK 切换逻辑

LSCLK是供RTC、SVD和LCD驱动使用的低速时钟，典型频率32K左右。LSCLK的源头是XTLF或者RCLP，芯片支持两者之间的自动切换或者由软件手动切换。

LSCLK的自动切换功能，由LSACTS寄存器配置，仅在XTLF使能的情况下有效，此时假设XTLF是主时钟，RCLP是备份时钟，仅用于防止XTLF异常停振。所以LSCATS只在XTLF使能的情况下起作

用，当XTLF出现意外停振，FDET输出的停振信号将LSCLK自动切换到RCLP。

在LSCLK自动切换不使能的情况下，软件可以通过LSCLKSEL寄存器实现对LSCLK的手动切换。



芯片上电复位后，XTLFEN=0，LFDET输出不停振，LSCLKSEL默认选中RCLP，XTLF无输出；因此上电后LSCLK默认是RCLP。

此后，软件如果希望使用XTLF，则应当使能XTLF并轮询LFDET输出，直到确认XTLF起振后，将LSCATS置位，或者LSCLKSEL.SEL写0xAA。

当LSCLK用作系统时钟（SYSCLK）时，建议软件打开停振自动切换功能。

13.2.8 RTC 时钟

RTC可以使用XTLF、RCLP和XTHF时钟工作。通常情况下，使用32768Hz的XTLF晶振来做RTC高精度计时。使用RCLP时，RTC走时精度无法保证。使用XTHF时，为了得到精确的定时，应使用32768Hz的整数倍频率晶体，比如4.096MHz或者8.192MHz晶体。

XTHF提供给RTC使用前，先经过一个预分频电路，得到尽可能接近32768Hz的时钟。假设使用4.096MHz晶体，预分频系数为125的情况下，可以得到准确的32768Hz时钟。预分频电路未同步计数器设计，输出分频时钟不保证为50%占空比。

13.3 高频晶体振荡电路(XTHF)

13.3.1 概述

通过外接高频晶体，XTHF能够为MCU提供高精度的高频时钟源。静态和负载电容应尽可能靠近XTHF引脚布置，其中负载电容大小应合理选择，以适配所选用的晶体类型。

XTHF可以适配4~24MHz晶体。软件可以通过XTHFCR.EN寄存器使能或关闭XTHF时钟。

XTHF包含4pF内部负载电容，可以在一定范围内微调振荡频率。

13.3.2 工作方式

XTHF上电后默认关闭。上电复位完成后，软件可以根据需要打开XTHF。由于晶振引脚与GPIO复用，软件使能XTHF前，需要将PA8和PA9引脚配置为模拟功能。

13.3.3 停振检测（HFDET）

FM33FR0带有片上停振检测电路，与XTHF电路一起使能或关闭。停振检测使能后可以持续检测XTHF输出，当发现XTHF停振时，会产生报警中断，同时产生高级定时器刹车信号；如果XTHF正在被直接或者间接的用作系统工作时钟（直接指SYSCLK选为XTHF，间接指SYSCLK选为PLL同时PLL使用XTHF为输入参考时钟），则停振信号将自动使能RCHF并将SYSCLK切换到RCHF，以避免高频晶体意外停振导致系统异常。

停振检测电路总是与XTHF同时打开或关闭，无法单独关闭，一旦XTHF使能，停振检测电路就会自动打开；当XTHF关闭时，停振检测也会自动关闭，避免误触发停振报警。

HFDET停振检测阈值约200KHz，即XTHF时钟频率低于200KHz时触发停振报警。

13.3.4 XTHF 启动过程

软件使能XTHF后，硬件等待XTHF输出稳定后自动置位RDY标志寄存器。

XTHF关闭或者HFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTHF后应等待RDY标志置位，再将XTHF作为系统时钟使用。

13.4 低频晶体振荡电路(XTLF)

13.4.1 概述

低频晶体振荡电路通过外接32768Hz晶体提供稳定的振荡源，功耗极低，主要用来给实时时钟(RTC)模块提供输入时钟。XTLF的振荡强度可调，用户可根据需要选择振荡强度，达到振荡能力与功耗的平衡。XTLF的反馈电阻集成在芯片内部，用户需要在振荡引脚上外加负载电容。

芯片内部集成了一个停振检测电路（LFDET），用来检测XTLF是否停振。一旦检测到XTLF停振，将产生XTLF停振中断，通知CPU及时处理。

软件可以使能或关闭XTLF。为了提高抗干扰能力，采用4bit的XTLFEN控制位。

XTLF也可以选择直接从外部输入时钟工作，输入时钟从XT32KI引脚灌入，此时需要将XTLF模块关闭，将XT32KI引脚配置为GPIO输入，并置位EXCKSEN寄存器。EXCKSEN寄存器用于控制时钟MUX选择GPIO输入还是XTLF输出。

13.4.2 工作方式

XTLF上电后默认关闭，软件启动，默认使用中等强度，以缩短起振时间，相应的振荡功耗也较大。典型的起振时间小于1s。当振荡器充分起振后，软件可以通过配置寄存器降低振荡功耗。

13.4.3 停振检测（LFDET）

FM33FR0带有片上停振检测电路，使能后可以持续检测XTLF输出，当发现XTLF停振时，产生报警中断，软件可以通过LSCATS寄存器决定是否自动将LSCLK切换到RCLP。

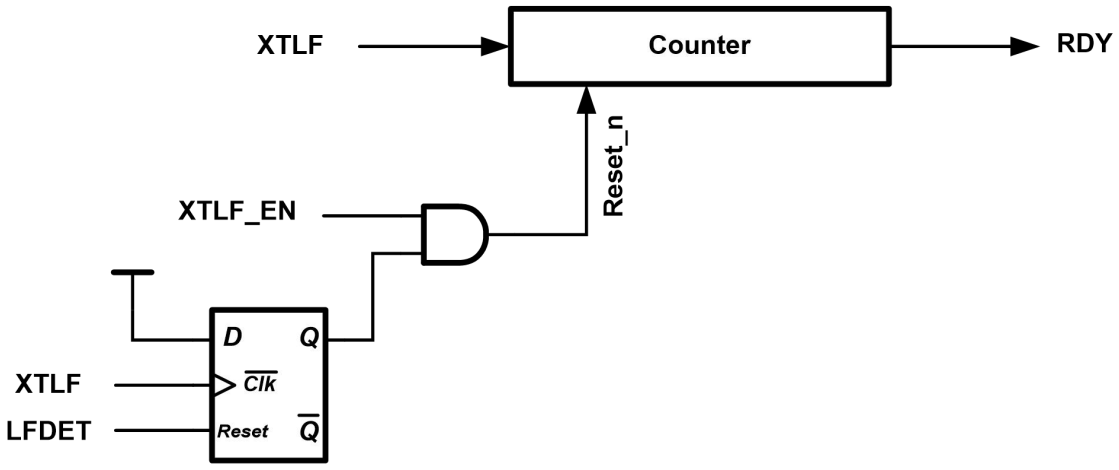
当LSCATS=1时，FDET检测到XTLF停振时，硬件会自动使能RCLP并将LSCLK切换为RCLP输出；当LSCATS=0时，停振检测只会产生报警中断，并不会自动切换时钟。

在XTLF停振状态下，软件也可以通过置位LSCATS来切换XTLF。

停振检测电路总是与XTLF同时打开或关闭，无法单独关闭，一旦XTLF使能，停振检测电路就会自动打开；当XTLF关闭时，停振检测也会自动关闭，避免误触发停振报警。

13.4.4 XTLF 启动过程

软件使能XTLF后，硬件等待XTLF输出稳定后自动置位RDY标志寄存器。



XTLF关闭或者LFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTLF后应等待RDY标志置位，再使用XTLF时钟。

13.5 低功耗低频环振(RCLP)

13.5.1 概述

RCLP是一个超低功耗内部环振，典型功耗仅400nA左右，典型输出频率32.768KHz。这个环振主要用作独立看门狗工作时钟，上电后自动启动，不可关闭。

13.6 锁相环(PLL)

13.6.1 概述

锁相环输入参考时钟可以是RCHF或XTHF分频，输入频率固定为1MHz，最高输出频率可达48MHz以及其2倍频。软件使用PLL作为系统时钟前，需配置输入参考时钟和倍频系数。

13.6.2 应用注意

出于可靠性考虑，软件需注意以下几点：

- 软件选择PLL输入时必须保证RCHF或XTHF为使能状态
- PLL输出选为SYSCLK时不能关闭PLL
- 软件应等待PLL锁定后再将SYSCLK配置为PLL输出

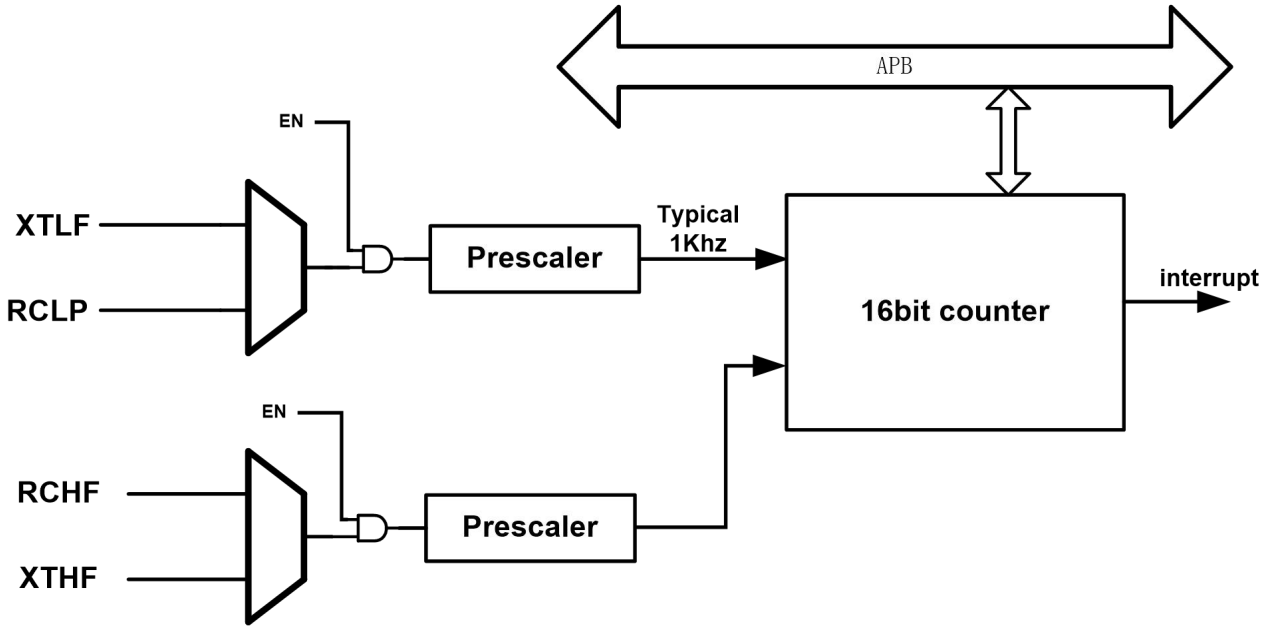
配置PLL输出48MHz，并使系统以48MHz主频运行：

- 配置PLLCR寄存器，选择输入时钟源和输出时钟频率
- 设置Flash wait cycle为1
- 将SYSCLK时钟选择为PLL输出

13.7 时钟校准

时钟校准电路是一个同步计数器，用于不同时钟之间互相校准。

时钟校准电路的核心是16位同步计数器，预分频器和输入信号源选择。如下图。



时钟校准的主要基准是XTLF，XTLF或RCLP经过预分频后得到周期性参考信号，典型频率1Khz左右。参考信号上升沿触发计数器计数，并在下一个上升沿停止计数器，计数器中的计数值反映了参考信号和被测时钟之间的频率关系。

比如，参考信号选择为32768Hz的32分频，即1024Hz；被测时钟为RCHF 8MHz，如果频率为准确的8MHz，则参考信号的一个周期内定时器计数值应该在7812附近，如果计数值偏离较多，则可以根据计数值计算频率误差并校准RCHF输出。

如果被测信号频率较高，比如RCHF 32MHz，在校准前应对被测时钟进行预分频，确保计数值不会溢出。

在RCHF完成校准的情况下，可以反过来对RCLP进行校准。

13.8 低功耗模式下的时钟源

在低功耗模式下，部分时钟源被硬件强制关闭，而另外一部分时钟源则仍可以保持工作。具体参见下表：

时钟源	LPRUN/Sleep/DeepSleep	说明
RCHF	X	硬件强制关闭
PLL	X	
XTHF	X	
RCLP	O	软件配置使能或关闭
XTLF	O	

13.9 休眠唤醒的时钟处理

当芯片从Sleep/DeepSleep模式下唤醒时，硬件自动打开RCHF并恢复到休眠前配置的频率输出；同时将SYSCLKSEL寄存器复位成000，将系统时钟选为RCHF，AHBPRES寄存器复位为000；因此芯片唤醒后默认将使用RCHF时钟工作。

13.10 寄存器

模块起始地址：0x4000_2400

offset 地址	名称	符号
0x00	系统时钟控制寄存器（System Clock Control Register）	CMU_SYSCLKCR
0x04	RCHF 控制寄存器（RCHF Control Register）	CMU_RCHFCR
0x08	RCHF 调校寄存器（RCHF Trim Register）	CMU_RCHFTR
0x0C	PLL 控制寄存器（PLL Control Register）	CMU_PLLCR
0x14	RCLP 调校寄存器（RCLP Trim Register）	CMU_RCLPTR
0x18	XTLF 控制寄存器（XTLF Control Register）	CMU_XTLFCR
0x1C	LSCLK 选择寄存器（LSCLK Select Register）	CMU_LSCLKSEL
0x20	XTHF 控制寄存器（XTHF Control Register）	CMU_XTHFCR
0x2C	CMU 中断使能寄存器（Interrupt Enable Register）	CMU_IER
0x30	CMU 中断标志寄存器（Interrupt Status Register）	CMU_ISR
0x34	外设总线时钟控制寄存器 1（Peripheral bus Clock Control Register1）	CMU_PCLKCR1
0x38	外设总线时钟控制寄存器 2（Peripheral bus Clock Control Register2）	CMU_ PCLKCR2
0x3C	外设总线时钟控制寄存器 3（Peripheral bus Clock Control Register3）	CMU_ PCLKCR 3
0x40	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_ PCLKCR 4
0x44	外设工作时钟配置寄存器 1（Peripheral Clock Config Register1）	CMU_OPCCR1
0x48	外设工作时钟配置寄存器 2（Peripheral Clock Config Register 2）	CMU_OPCCR2
0x4C	外设工作时钟配置寄存器 3（Peripheral Clock Config Register 3）	CMU_OPCCR3
0x50	AHB Master 控制寄存器（AHB Master Control Register）	CMU_AHBMCR
0x54	时钟校准控制寄存器（Clock Calibration Control Register）	CMU_CCCR
0x58	时钟校准配置寄存器（Clock Calibration Config Register）	CMU_CCFR
0x5C	时钟校准计数值寄存器	CMU_CCNR

offset 地址	名称	符号
	（Clock Calibration Counter Register）	
0x60	时钟校准中断标志寄存器（Clock Calibration Interrupt Status Register）	CMU_CCISR

13.10.1 系统时钟配置寄存器（CMU_SYSCLKCR）

名称	CMU_SYSCLKCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				LSCATS	-	SLP_EN EXTI	-
位权限	U-0				R/W-1	U-0	R/W-1	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					APBPRES		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		-			AHBPRES		
位权限	R/W-00		U-0			R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	STCLKSEL		-			SYSCLKSEL		
位权限	R/W-00		U-0			R/W-000		

Bit	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27	LSCATS	LSCLK 自动切换使能 0：当检测到 XTLF 异常停振时，不会自动将 LSCLK 切换到 RCLP，软件可以通过写 LSCLKSEL 寄存器手动切换到 RCLP 1：当检测到 XTLF 异常停振时，自动使能 RCLP 并将 LSCLK 切换到 RCLP
26	--	RFU：未实现，读为 0
25	SLP_ENEXTI	Sleep/DeepSleep 模式下 EXTI 采样设置 1：Sleep/DeepSleep 模式下使能外部引脚中断采样（采样时钟为 LSCLK） 0：Sleep/DeepSleep 模式下禁止外部引脚中断采样（将无法产生 EXTI 中断）
24:19	--	RFU：未实现，读为 0
18:16	APBPRES	APB 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
15:11	--	RFU：未实现，读为 0
10:8	AHBPRES	AHB 时钟分频选择 0xx：不分频 100：2 分频

Bit	助记符	功能描述
		101：4 分频 110：8 分频 111：16 分频
7:6	STCLKSEL	CPU 内核 systick 工作时钟选择 00：SCLK 01：LSCLK 10：RFU 11：RFU
5:3	--	RFU：未实现，读为 0
2:0	SYSCLKSEL	系统时钟源选择 000：RCHF 001：XTHF 010：PLL 011：RCHF 100：RCHF 101：XTLF 110：RCLP 111：RCHF

13.10.2 RCHF 时钟控制寄存器（CMU_RCHCR）

名称	CMU_RCHFCR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				FSEL			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-1

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:16	FSEL	RCHF 频率选择寄存器 0000：8MHz 0001：16MHz 0010：24MHz 0011：32MHz 其他：RFU
15:1	--	RFU：未实现，读为 0
0	EN	RCHF 使能寄存器 1：使能 RCHF 0：关闭 RCHF

13.10.3 RCHF 调校寄存器（CMU_RCHFTR）

名称	CMU_RCHFTR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TRIM	RCHF 频率调校寄存器，0x00 表示频率最低，0x7F 表示频率最高，调校范围为中心频率+/-30%，调校步长为中心频率 0.5% 上电后芯片自动从 LDT0 读取 8MHz 调校值并写入此寄存器 软件使用其他频率时，可以自行从 LDT0 指定地址读取调校信息并写入此寄存器，从而确保输出频率准确。

13.10.4 PLL 控制寄存器（CMU_PLLCR）

名称	CMU_PLLCR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						DB	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB							
位权限	R/W-0001 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC			OSEL	-	INSEL	EN
位权限	R/Dy-0	R/W-000			R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25:16	DB	PLL 倍频比，输出时钟频率为 1M*(PLLDB+1) 0x1F：输出 32 倍频 .....

Bit	助记符	功能描述
		0x2F：输出 48 倍频 *注：请勿使用高于 48 倍频的设置。*
15:8	--	RFU：未实现，读为 0
7	LOCKED	PLL 锁定标志，软件通过查询此寄存器确认 PLL 已经处于锁定状态 1：PLL 已锁定 0：PLL 未锁定
6:4	REFPRSC	PLL 参考时钟预分频（目标是产生 1MHz 参考时钟给 PLL） 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：12 分频 101：16 分频 110：24 分频 111：32 分频
3	OSEL	PLL 输出选择寄存器 0：选择 PLL 一倍输出作为数字电路内的 PLL 时钟 1：选择 PLL 两倍输出作为数字电路内的 PLL 时钟
2	--	RFU：未实现，读为 0
1	INSEL	PLL 输入选择寄存器 0：RCHF 1：XTHF
0	EN	PLL 使能寄存器 1：使能 PLL 0：关闭 PLL

13.10.5 RCLP 调校寄存器（CMU_RCLPTR）

名称	CMU_RCLPTR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TRIM	RCLP 调校值寄存器 0000 0000：频率最低

Bit	助记符	功能描述
		…… 1111 1111：频率最高

13.10.6 XTLF 控制寄存器（CMU_XTLFCR）

名称	CMU_XTLFCR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WAIT			EXCKSEN	EN			
位权限	R/W-111			R/W-0	R/W-0			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			RDY	IPW			
位权限	U-0			R-0	RW-1000			

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
15:13	WAIT	XTLF 启动等待时间配置，单位是 XTLF 时钟周期 000：128 001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12	EXCKSEN	外部输入低频时钟使能 0：芯片内部使用 XTLF 晶体输出 1：芯片内部使用 XT32KI 引脚输入的外部时钟
11:8	EN	XTLF 使能寄存器，上电默认 XTLF 关闭 1010：关闭 XTLF 和 FDET 0101：使能 XTLF 和 FDET 物理上只有 1bit，当 XTLF 工作时，软件必须写入 1010 才能将其关闭，当 XTLF 不工作时，软件必须写入 0101 才能启动
7:5	--	RFU：未实现，读为 0
4	RDY	XTLF 稳定标志，硬件置位，只读 关闭 XTLF 或者 XTLF 发生停振时自动清零
3:0	IPW	XTLF 工作电流选择，电流越大表示振荡强度越高，上电复位后使用 1000 档位起振，正常工作时推荐使用 1100 或 1011 档位，实际应根据适配晶体的实测负阻特性选择合适的电流大小 0000 : 850nA 0001 : 800nA

Bit	助记符	功能描述
		0010 : 750nA 0011 : 700nA 0100 : 650nA 0101 : 600nA 0110 : 550nA 0111 : 500nA 1000 : 450nA (default) 1001 : 400nA 1010 : 350nA 1011 : 300nA 1100 : 250nA 1101 : 200nA 1110 : 150nA 1111 : 100nA

13.10.7 LSCLK 选择寄存器（CMU_ LSCLKSEL）

名称	CMU_LSCLKSEL							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							STA
位权限	U-0							R-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEL							
位权限	R/W-xxxx xxxx							

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	STA	LSCLK 状态标志寄存器，只读 0：当前 LSCLK 源头是 XTLF 1：当前 LSCLK 源头是 RCLP
7:0	SEL	LSCLK 时钟手动切换寄存器，物理实现上只有 1bit；复位值选中 RCLP； 当 LSCLK 为 XTLF 时，软件对此地址写 0x55，会将 LSCLK 源头切换到 RCLP 当 LSCLK 为 RCLP 时，软件对此地址写 0xAA，会将 LSCLK 源头切换到 XTLF 写任意其他值，不改变当前 LSCLK；此寄存器仅在 LSCATS 为 0 时有效

13.10.8 XTHF 控制寄存器（CMU_XTHFCR）

名称	CMU_XTHFCR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					CAP_CFG		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WAIT			HF_CFG				
位权限	R/W-111			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RDY	EN
位权限	U-0						R-0	R/W-0

Bit	助记符	功能描述
31	RFUI	保留
30:19	--	RFU：未实现，读为 0
18:16	CAP_CFG	XTHF 的片内电容配置
15:13	WAIT	XTHF 启动等待时间配置，单位是 XTHF 时钟周期 000：128 001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12:8	HF_CFG	XTHF 振荡强度配置 00000：最弱 ..... 11111：最强
7:2	--	RFU：未实现，读为 0
1	RDY	XTHF 稳定标志，硬件置位，只读 关闭 XTHF 或者 XTHF 发生停振时自动清零
0	EN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

13.10.9 CMU 中断使能寄存器（CMU_IER）

名称	CMU_IER							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					SYSCKE_IE	HFDET_IE	LFDET_IE
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	SYSCKE_IE	SYSCLK 时钟选择错误中断使能寄存器，1 有效
1	HFDET_IE	XTHF 高频检测报警中断使能，1 有效
0	LFDET_IE	XTLF 低频检测报警中断使能，1 有效

13.10.10 CMU 中断标志寄存器（CMU_ISR）

名称	CMU_ISR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						HFDETO	LFDETO
位权限	U-0						R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					SYSCSE_IF	HFDETIF	LFDETIF
位权限	U-0					R/W-0	R/W -0	R/W -0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	HFDETO	高频晶体停振检测模块输出 1：XTHF 未停振 0：XTHF 停振
8	LFDETO	低频晶体停振检测模块输出 1：XTLF 未停振 0：XTLF 停振
7:3	--	RFU：未实现，读为 0
2	SYSCSE_IF	SYSCLK 时钟选择错误中断标志。 当被选择的目标时钟没有使能，或者已经停振时，时钟切换被禁止，同时置位此标志寄存器。软件写 1 清零。
1	HFDETIF	高频停振检测中断标志寄存器，XTHF 停振时硬件异步置位，软件

Bit	助记符	功能描述
		写 1 清零；只有在 HFDETO 不为 0 的情况下才能够清除此寄存器
0	LFDETIF	低频停振检测中断标志寄存器，XTLF 停振时硬件异步置位，软件写 1 清零；只有在 LFDETO 不为 0 的情况下才能够清除此寄存器

13.10.11 外设总线时钟控制寄存器 1（CMU_ PCLKCR1）

名称	CMU_PCLKCR1							
offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						COMP_ PCE	SVD_PC E
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD_PC E	ANT_PC E	IWDT_P CE	SCU_PC E	PMU_PC E	RTC_PC E	LPT16_P CE	-
位权限	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	U-0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	COMP_PCE	Comparator 总线时钟使能，高使能
8	SVD_PCE	SVD 总线时钟使能，高使能
7	PAD_PCE	PADCFG 总线时钟使能，高使能
6	ANAC_PCE	模拟测试 buffer 总线时钟使能，高使能 此寄存器用于控制 BUF4TST 的总线时钟
5	IWDT_PCE	IWDT 总线时钟使能，高使能
4	SCU_PCE	SCU 总线时钟使能，高使能
3	PMU_PCE	PMU 总线时钟使能，高使能
2	RTC_PCE	RTC 总线时钟使能，高使能
1	LPT16_PCE	LPTIM16 总线时钟使能，高使能
0	--	RFU：未实现，读为 0

13.10.12 外设总线时钟控制寄存器 2（CMU_ PCLKCR2）

名称	CMU_PCLKCR2							
offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			INTMUX_PCE	PGL_PCE	TSI_PCE	DIVAS_PCE	ADC_PCE
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT_PCE	RAMBIST_PCE	FLASH_PCE	DMA_PCE	LCD_PCE	AES_PCE	RNG_PCE	CRC_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	INTMUX_PCE	INTMUX 总线时钟使能，高使能
11	PGL_PCE	PGL 总线时钟使能，高使能
10	TSI_PCE	TSI 总线时钟使能，高使能
9	DIVAS_PCE	DIVAS 总线时钟使能，高使能
8	ADC_PCE	ADC 总线时钟使能，高使能
7	WWDT_PCE	WWDT 总线时钟使能，高使能
6	RAMBIST_PCE	RAMBIST 总线时钟使能，高使能
5	FLASH_PCE	Flash 擦写控制器总线时钟使能，高使能
4	DMA_PCE	DMA 总线时钟使能，高使能
3	LCD_PCE	LCD 总线时钟使能，高使能
2	AES_PCE	AES 总线时钟使能，高使能
1	RNG_PCE	RNG 总线时钟使能，高使能
0	CRC_PCE	CRC 总线时钟使能，高使能

13.10.13 外设总线时钟控制寄存器 3（CMU_ PCLKCR3）

名称	CMU_PCLKCR3							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		I2CSMB1_PCE	I2CSMB0_PCE	I2C3_PCE	I2C2_PCE	I2C1_PCE	I2C0_PCE
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							RFU
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	UCIR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	UART2_PCE	UART1_PCE	UART0_PCE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SPI3_PCE	SPI2_PCE	SPI1_PCE	SPI0_PCE
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:30	-	RFU：未实现，读为 0
29	I2CSMB1_PCE	I2C_SMBUS1 总线时钟使能，高有效
28	I2CSMB0_PCE	I2C_SMBUS0 总线时钟使能，高有效

位号	助记符	功能描述
27	I2C3_PCE	I2C3 总线时钟使能，高有效
26	I2C2_PCE	I2C2 总线时钟使能，高有效
25	I2C1_PCE	I2C1 总线时钟使能，高有效
24	I2C0_PCE	I2C0 总线时钟使能，高有效
23:17	-	RFU：未实现，读为 0
16	RFU	Dummy 寄存器
15	-	RFU：未实现，读为 0
14	UCIR_PCE	UART 红外调制工作时钟使能，高有效
13	UART5_PCE	UART5 总线时钟使能，高有效
12	UART4_PCE	UART4 总线时钟使能，高有效
11	UART3_PCE	UART3 总线时钟使能，高有效
10	UART2_PCE	UART2 总线时钟使能，高有效
9	UART1_PCE	UART1 总线时钟使能，高有效
8	UART0_PCE	UART0 总线时钟使能，高有效
7:4	-	RFU：未实现，读为 0
3	SPI3_PCE	SPI3 总线时钟使能，高有效
2	SPI2_PCE	SPI2 总线时钟使能，高有效
1	SPI1_PCE	SPI1 总线时钟使能，高有效
0	SPI0_PCE	SPI0 总线时钟使能，高有效

13.10.14 外设总线时钟控制寄存器 4（CMU_ PCLKCR4）

名称	CMU_PCLKCR4							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							BT16_PCE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		TAU1_PCE	TAU0_PCE	-			ATIM_PCE
位权限	U-0		R/W-0	R/W-0	U-0			R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	BT16_PCE	BSTIM16 总线时钟使能，高有效
7:6	-	RFU：未实现，读为 0
5	TAU1_PCE	定时器阵列 1 总线时钟使能，高有效
4	TAU0_PCE	定时器阵列 0 总线时钟使能，高有效
3:1	-	RFU：未实现，读为 0
0	ATIM_PCE	ATIMER 总线时钟使能，高有效

13.10.15 外设工作时钟配置寄存器 1（CMU_OPCCR1）

名称	CMU_OPCCR1							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	EXTICKS	-				RFU	
位权限	U-0	R/W-0	U-0				R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				I2CSMB1CKS		I2CSMB0CKS	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BT16CKS		-		LPT16CKS		-	
位权限	R/W-00		U-0		R/W-00		U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ATCKS	-	-		UART1CKS		UART0CKS	
位权限	R/W-0	U-0	U-0		R/W-00		R/W-00	

位号	助记符	功能描述
31	-	RFU：未实现，读为 0
30	EXTICKS	EXTI 中断采样时钟选择 1：外部引脚中断使用 LSCLK 采样 0：外部引脚中断使用 AHBCLK 采样 *建议在关闭所有 EXTI 中断的情况下设置，设置完成后再使能 EXTI 中断
29:26	-	RFU：未实现，读为 0
25:24	RFU	Dummy 寄存器
23:20	-	RFU：未实现，读为 0
19:18	I2CSMB1CKS	I2C_SMB1 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
17:16	I2CSMB0CKS	I2C_SMB0 主机工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
15:14	BT16CKS	BSTIM16 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：RFU
13:12	-	RFU：未实现，读为 0
11:10	LPT16CKS	LPTIM16 工作时钟选择 00：APBCLK 01：LSCLK

位号	助记符	功能描述
		10：RCLP 11：RFU
9:8	-	RFU：未实现，读为 0
7	ATCKS	ATIM 工作时钟源选择寄存器 0：APBCLK 1：PLL 两倍频
6:4	-	RFU：未实现，读为 0
3:2	UART1CKS	UART1 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
1:0	UART0CKS	UART0 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF

13.10.16 外设工作时钟配置寄存器 2（CMU_OPCCR2）

名称	CMU_OPCCR2							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RTCPRSC							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTCPRSC		-					RTCCKS
位权限	R/W-00		U-0					R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			RNGPRSC			-	
位权限	U-0			R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			ADCPRSC			ADCCKS	
位权限	U-0			R/W-000			R/W-00	

Bit	助记符	功能描述
31:22	RTCPRSC	使用 XTHF 作为 RTC 时的 XTHF 预分频系数，分频比为： N= RTCPRSC+1 比如 RTCPRSC=124，分频比=125，当 XTHF 外接 4.096MHz 晶体时可以产生 32768Hz 时钟给 RTC
21:17	--	RFU：未实现，读为 0
16	RTCCKS	RTC 工作时钟选择 0：LSCLK 1：XTHF 分频
15:13	--	RFU：未实现，读为 0
12:10	RNGPRSC	随机数发生器工作时钟分频 000：不分频

Bit	助记符	功能描述
		001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
9:5	--	RFU：未实现，读为 0
4:2	ADCPRSC	ADC 工作时钟预分频 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
1:0	ADCCKS	ADC 工作时钟选择 00：RFU 01：RCHF 10：XTHF 11：PLL

13.10.17 外设工作时钟配置寄存器 3（CMU_ OPCCR3）

名称	CMU_OPCCR3							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	FLASHCKE	-	RFU	-			RNGCKE
位权限	R/W-0	R/W-0	U-0	R/W-0	U-0			R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		I2CSMB1_CKE	I2CSMB0_CKE	-			ADCCKE
位权限	U-0		R/W-0	R/W-0	U-0			R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ATCKE	-	-	-		-	UART1CKE	UART0CKE
位权限	R/W-0	U-0	U-0	U-0		U-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				BT16CKE	-	LPT16CKE	-
位权限	U-0				R/W-0	U-0	R/W-0	U-0

Bit	助记符	功能描述
31	EXTICKE	外部引脚中断采样时钟使能，高有效
30	FLASHCKE	Flash 擦写时钟使能，高有效
29	--	RFU：未实现，读为 0
28	RFU	Dummy 寄存器
27:25	--	RFU：未实现，读为 0

Bit	助记符	功能描述
24	RNGCKE	随机数发生器工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
23:22	--	RFU：未实现，读为 0
21	I2CSMB1_CKE	I2C_SMB1 工作时钟使能，高有效
20	I2CSMB0_CKE	I2C_SMB0 工作时钟使能，高有效
19:17	--	RFU：未实现，读为 0
16	ADCCKE	ADC 工作时钟使能，高有效
15	ATCKE	高级定时器工作时钟使能，高有效
14	--	RFU：未实现，读为 0
13	--	RFU：未实现，读为 0
12:11	--	RFU：未实现，读为 0
10	--	RFU：未实现，读为 0
9	UART1CKE	UART1 工作时钟使能，高有效
8	UART0CKE	UART0 工作时钟使能，高有效
7:4	--	RFU：未实现，读为 0
3	BT16CKE	BSTIM16 工作时钟使能，高有效
2	--	RFU：未实现，读为 0
1	LPT16CKE	LPTIM16 工作时钟使能，高有效
0	--	RFU：未实现，读为 0

13.10.18 AHB Master 控制寄存器（CMU_AHBMCR）

名称	CMU_AHBMCR							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	RFUI	-					
位权限	R/W-1	R/W-0	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							MPRIL
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:30	RFUI	保留位
29:1	--	RFU：未实现，读为 0
0	MPRIL	AHB Master 优先级配置寄存器 0：DMA 优先 1：CPU 优先

13.10.19 时钟校准控制寄存器（CMU_CCCR）

名称	CMU_CCCR							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						IE	EN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	IE	时钟校准中断使能 (Clock Calibration interrupt enable) 0：禁止中断 1：允许中断
0	EN	时钟校准使能 1：启动校准 0：关闭校准，校准电路复位

13.10.20 时钟校准配置寄存器（CMU_CCFR）

名称	CMU_CCFR							
offset	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CALPSC		REFPSC	
位权限	U-0				R/W-00		R/W-10	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					CALSEL		REFSEL
位权限	U-0					R/W-00		R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:10	CALPSC	校准时钟预分频 (Calibration clock prescaler) 00：不分频 01：2 分频

Bit	助记符	功能描述
		10：4 分频 11：8 分频
9:8	REFPSC	参考时钟预分频 (Reference clock prescaler) 00：8 分频 01：16 分频 10：32 分频 11：64 分频
7:3	--	RFU：未实现，读为 0
2:1	CALSEL	校准时钟选择 (Calibration clock select) 00：RFU 01：RCHF 10：RFU 11：XTHF
0	REFSEL	参考时钟选择 (Reference clock select) 0：XTLF 1：RCLP

13.10.21 时钟校准计数值寄存器（CMU_CCNR）

名称	CMU_CCNR							
offset	0x5C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	CNT	时钟校准计数器计数值 (Clock calibration counter) 校准周期结束后软件通过读取计数值来计算目标时钟频率

13.10.22 时钟校准中断标志寄存器（CMU_CCISR）

名称	CMU_CCISR							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							IF
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	IF	时钟校准中断标志 (Clock calibration interrupt flag) 校准周期完成后硬件置位，软件写 1 清零

14 电源电压监测（SVD）

14.1 概述

电源检测电路主要用来监测外部主电源的供电情况，及时检测到外部主电源欠压或恢复的情况，并给出中断信号。电源检测电路可关断以节省功耗。

特点：

- 监测主电源，电压低于或高于设定的阈值时产生中断
- 低压检测 15 级可编程阈值档位，档位间隔 0.214V
- 电压检测迟滞窗口 0.1V
- 可关断
- 支持 1 个外部通道直接输入与内部基准电压源比较
- 外部通道支持 100mV 窗口
- deepsleep 模式下不支持

14.2 结构框图

下图是电源检测电路的模块框图。

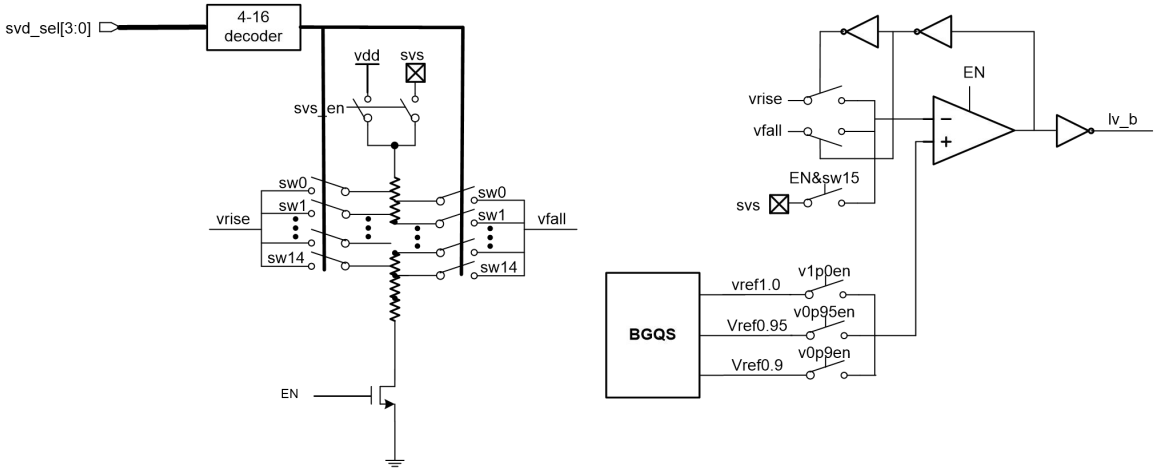


图 14-1 低压检测电路框图

SVD有内部通道和外部通道，内部通道用于芯片电源检测，外部通道用于外部输入信号与内部基准电压比较。

SVD工作时序示意图：

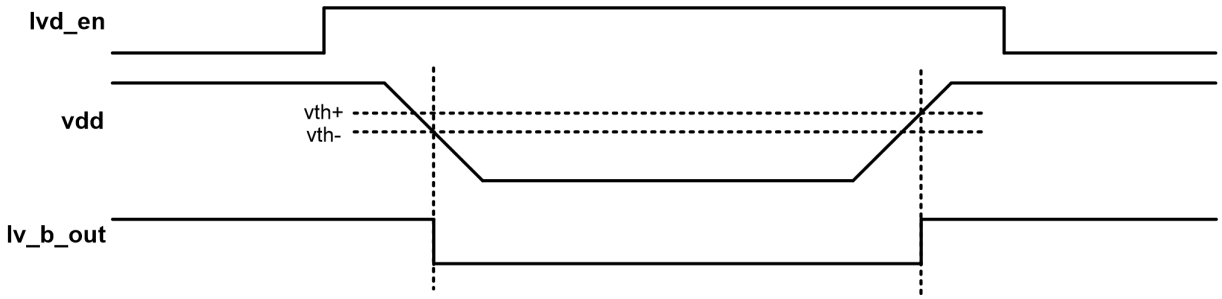


图 14-2 低压检测电路工作时序

14.3 引脚定义

SVD模块可以直接检测芯片电源（VDD），也可以通过1个SVS引脚检测外部电压信号。

检测SVS输入时，需要将GPIO的FCR寄存器配置为11（analog function）。

14.4 功能描述

电源检测电路可以用来检测主电源电压及外部电压。电源电压通过分压电阻产生15级检测电平，每级相差0.214V；另外还支持外部输入检测通道。VDD分压通过多路选择器送入比较器，与内部参考电压相比较，根据低压报警阈值设置，若待检测电平低于参考电压，引起输出电压跳变，会产生欠压中断，通知MCU及时处理该事件；而当VDD恢复至阈值以上（有大约0.1V迟滞窗口），则会产生欠压恢复中断。

电源检测电路可由软件配置使能或禁止工作。

常使能条件下SVD从欠压到过压有0.1V回滞窗口；对于外部通道，可以通过软件配合，即欠压中断后人为设置一个较高的过压阈值，来解决窗口问题。相应的，基准电压需要输出1.0V、0.95V、0.9V三个基准电压。

常使能情况下，软件开启SVD后经过一到两个LSCLK时钟同步周期后，SVD就会开始工作。SVD开启后到输出稳定建立大约需要100us时间，软件读取SVD输出时需要注意。

如果芯片进入休眠模式后关闭了所有时钟，又希望使用SVD，则需要在休眠前将SVD设置为常使能，并且关闭数字滤波功能。

工作模式说明：

- 在常使能/内部通道模式下，检测阈值有窗口，下降阈值和上升阈值窗口为0.1V，不使能到使能

时检测下降阈值。

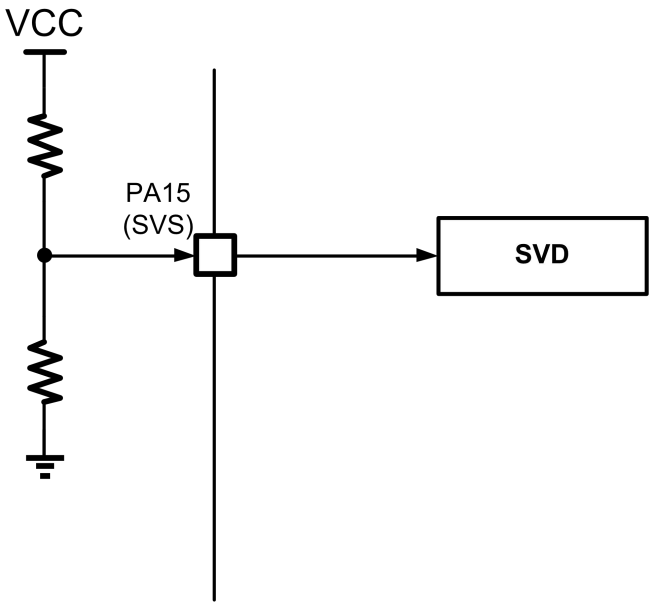
- 在常使能/外部通道模式下，输入的基准电压为三档位输入，分别为1.0V、0.95V、0.9V，检测阈值没有窗口，需要软件配合，即在检测到欠压时，软件将阈值档位调高一档；在检测到非欠压时，软件将档位恢复。

14.5 外部电源检测

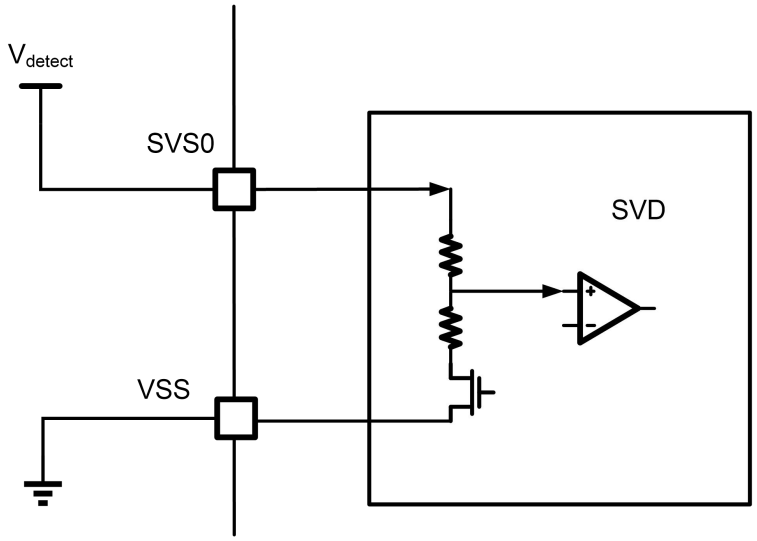
电源SVD除了可以检测芯片，也可以对外部电压信号进行掉电或上电检测。

外部电源检测通过SVS引脚实现，SVS的输入可以采用外部电阻分压或内部电阻分压后，再输入到比较器进行检测。

下图为外部电阻分压的外部电源检测：



下图为内部电阻分压的外部信号检测：



寄存器配置方法如下表：

SVSEN	SVDLVL	说明
0	X	外部电源检测通道关闭，仅检测内部电源电压
1	1111	外部电压输入不做内部分压，直接输入到比较器与内部基准电压比较
	0000~1110	外部电压输入先经过内部电阻分压，然后再输入到比较器与内部基准电压比较 分压后的档位参见后续章节描述

14.6 电源检测阈值

通过SVSEN和SVDLVL寄存器可以选择电压检测对象和检测阈值。

内部电源检测：SVSxEN = 0，{VREF1P0EN, VREF0P95EN, VREF0P9EN} = 100，比较基准1.0V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.800	1.900
0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686

1110	4.800	4.900
1111	N/A	N/A

内部电源检测：SVSxEN = 0，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 010，比较基准0.95V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	N/A	N/A

内部电源检测：SVSxEN = 0，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 001，比较基准0.9V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476
0110	2.572	2.655
0111	2.750	2.833
1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	N/A	N/A

外部电压检测：SVSxEN =1，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 100，比较基准1.0V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.800	1.900

0001	2.014	2.114
0010	2.229	2.329
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	1.2	1.2

外部电压检测：SVSxEN =1，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 010，比较基准0.95V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.650	1.742
0001	1.846	1.938
0010	2.043	2.135
0011	2.239	2.331
0100	2.436	2.527
0101	2.632	2.723
0110	2.829	2.921
0111	3.025	3.117
1000	3.221	3.313
1001	3.418	3.510
1010	3.614	3.706
1011	3.811	3.902
1100	4.007	4.098
1101	4.204	4.296
1110	4.400	4.492
1111	1.1	1.1

外部电压检测：SVSxEN =1，{VREF1P0EN, VREF0P95EN, VREF0P9EN } = 001，比较基准0.9V

SVDLVL	下降阈值（V）	上升阈值（V）
0000	1.500	1.583
0001	1.678	1.762
0010	1.858	1.941
0011	2.036	2.119
0100	2.214	2.298
0101	2.393	2.476
0110	2.572	2.655
0111	2.750	2.833

1000	2.928	3.012
1001	3.108	3.191
1010	3.286	3.369
1011	3.464	3.548
1100	3.643	3.726
1101	3.822	3.905
1110	4.000	4.083
1111	1.0	1.0

14.7 寄存器

模块偏移地址：0x4001_A800

offset 地址	名称	符号
0x00	SVD 配置寄存器 （SVD Config Register）	SVD_CFGR
0x04	SVD 控制寄存器 （SVD Control Register）	SVD_CR
0x08	SVD 中断使能寄存器 （SVD Interrupt Enable Register）	SVD_IER
0x0C	SVD 状态和标志寄存器 （SVD Interrupt Status Register）	SVD_ISR
0x10	SVD 参考电压选择寄存器 （SVD reference Voltage Select Register）	SVD_VSR

14.7.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LVL				DFEN	MOD	RFU	
位权限	R/W-0000				R/W-1	R/W-0	R/W-00	

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:4	LVL	SVD 报警阈值设置，档位定义参见 14.6 电源检测阈值 (SVD threshold level)
3	DFEN	数字滤波使能（SVDMODE=1 时必须置 1）(Digital Filter Enable) 1：启动 SVD 输出的数字滤波 0：关闭 SVD 输出的数字滤波
2	MOD	SVD 工作模式选择，配置模式后还要置位 SVDEN 才会启动 SVD (SVD Mode) 0：常使能模式 注意：禁止写 1
1:0	RFU	Dummy 寄存器

14.7.2 SVD 控制寄存器（SVD_CR）

名称	SVD_CR
offset	0x04

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TE
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SVS0EN	EN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	TE	SVD 测试使能，避免写 1 (SVD test enable)
7:2	--	RFU：未实现，读为 0
1	SVS0EN	SVS0 外部电源检测通道控制信号 (SVS0 external monitor channel enable) 0：SVS0 通道关闭 1：SVS0 通道使能 当 SVS0EN=1 时，根据 SVDLVL 寄存器可以设置 SVS 输入后是否经过内部电阻分压；如果 SVDLVL=1111，则 SVS 输入不做分压，如果 SVDLVL != 1111，则 SVS 输入经过内部电阻分压。
0	EN	SVD 使能 (SVD enable) 1：启动 SVD 0：关闭 SVD

14.7.3 SVD 中断使能寄存器（SVD_IER）

名称	SVD_IER							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PFIE	PRIE
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	PFIE	电源跌落中断使能寄存器 (Power Fall interrupt enable)

Bit	助记符	功能描述
		1：允许电源跌落中断 0：禁止中断
0	PRIE	电源恢复中断使能寄存器 (Power Rise interrupt enable) 1：允许电源恢复中断 0：禁止中断

14.7.4 SVD 状态和标志寄存器（SVD_ISR）

名称	SVD_ISR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SVDO
位权限	U-0							R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SVDR	-					PFF	PRF
位权限	R	U-0					R/W-0	R/W-0

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	SVDO	SVD 电源检测输出 1：电源电压高于 SVD 当前阈值 0：电源电压低于 SVD 当前阈值
7	SVDR	SVD 输出锁存信号，数字电路锁存的 SVD 状态
6:2	--	RFU：未实现，读为 0
1	PFF	电源跌落中断标志寄存器，电源电压跌落到 SVD 阈值之下时置位，软件写 1 清零
0	PRF	电源恢复中断标志寄存器，电源电压上升到 SVD 阈值之上时置位，软件写 1 清零

14.7.5 SVD 参考电压选择寄存器（SVD_VSR）

名称	SVD_VSR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit0	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					V1P0EN	V0P95EN	V0P9EN
位权限	U-0					R/W-1	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	V1P0EN	1.0V 基准输入使能信号 (1.0V reference enable) 1：使能 1.0V 基准输入 0：关闭 1.0V 基准输入
1	V0P95EN	0.95V 基准输入使能信号 (0.95V reference enable) 1：使能 0.95V 基准输入 0：关闭 0.95V 基准输入
0	V0P9EN	0.9V 基准输入使能信号 (0.9V reference enable) 1：使能 0.9V 基准输入 0：关闭 0.9V 基准输入

15 AES 硬件运算单元（AES）

15.1 功能描述

AES单元主要功能如下：

- 支持解密密钥扩展
- 支持128bit/192bit/256bit的密钥长度
- 支持ECB，CBC，CTR，GCM
- 支持DMA进行自动数据传输
- 支持GF（2^128）域下的乘法，支持GMAC

15.2 工作模式

AES有4种工作模式，通过配置MODE[1:0]寄存器设置。

模式1：用存储在AES_KEYx寄存器中的密钥加密。

模式2：密钥扩展，把初始存储在AES_KEYx寄存器的加密密钥覆盖成在密钥扩展完成后存储在内部寄存器的密钥计算结果。

模式3：用存储在AES_KEYx寄存器中的解密密钥（预计算的）解密。

模式4：用存储在AES_KEYx寄存器中的加密密钥进行密钥扩展和解密。（在CTR模式下不使用）

首先通过配置MODE[1:0]寄存器确定工作模式，MODE寄存器必须在AES使能前（EN=0时）才能够配置。KEY寄存器也应该在AES使能前配置。之后配置数据流处理模式寄存器CHMOD[1:0]，在CBC/CTR/GCM模式下还需要配置IV寄存器。

接着可以使能EN，在模式1/模式3/模式4下，AES模块等待软件往AES_DIR寄存器写入输入数据，写4次写完128bit后AES开始计算。在模式2时，使能EN后就马上进行密钥扩展运算了。

计算完成后标志CCF会置起，如果CCFIE=1，会产生一个中断信号。软件再从AES_DOR寄存器中读4次共128bit的结果。

AES还支持DMA模式。通过配置DMAOEN=1和DMAIEN=1，AES可以配合DMA连续的处理数据，无需CPU的介入。

错误标志RDERR和WRERR会在一次错误的读写操作时置起，如果ERRIE使能，还会产生相应的错

误中断。AES在产生错误后还会继续正常工作。

通过重置EN寄存器能够在任何时候复位AES模块。

15.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

15.3.1 ECB 模式

默认的工作模式，该模式下无需使用IV寄存器，每个block单独进行加解密计算。加解密流程如图15-1和图15-2所示。

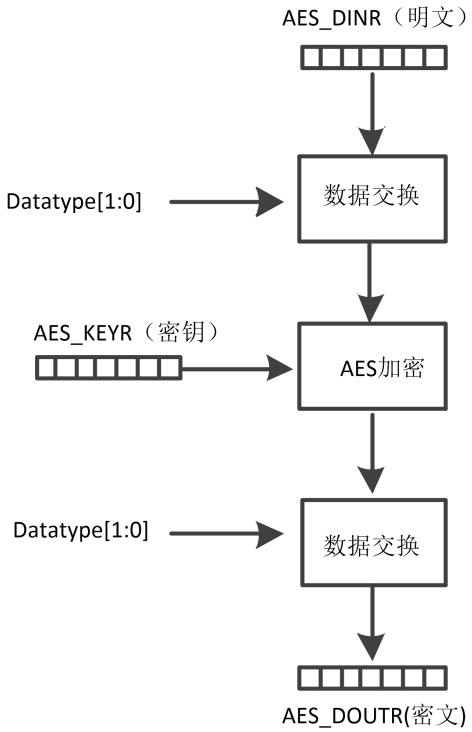


图 15-1 ECB 模式加密流程

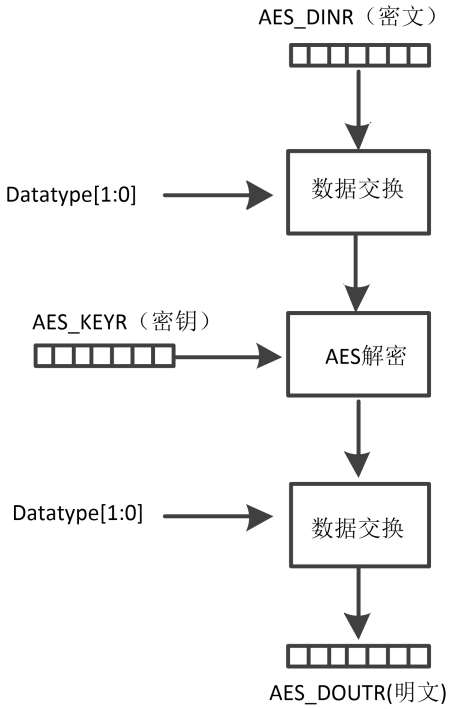


图 15-2 ECB 模式解密流程

15.3.2 CBC 模式

每个block的明文数据与前一block的加密结果异或后作为加密的数据输入。第一个block需要一个初始的IVRx寄存器值。加密时异或操作在加密前而解密时异或操作在加密后。工作流程如图15-3和图15-4所示。

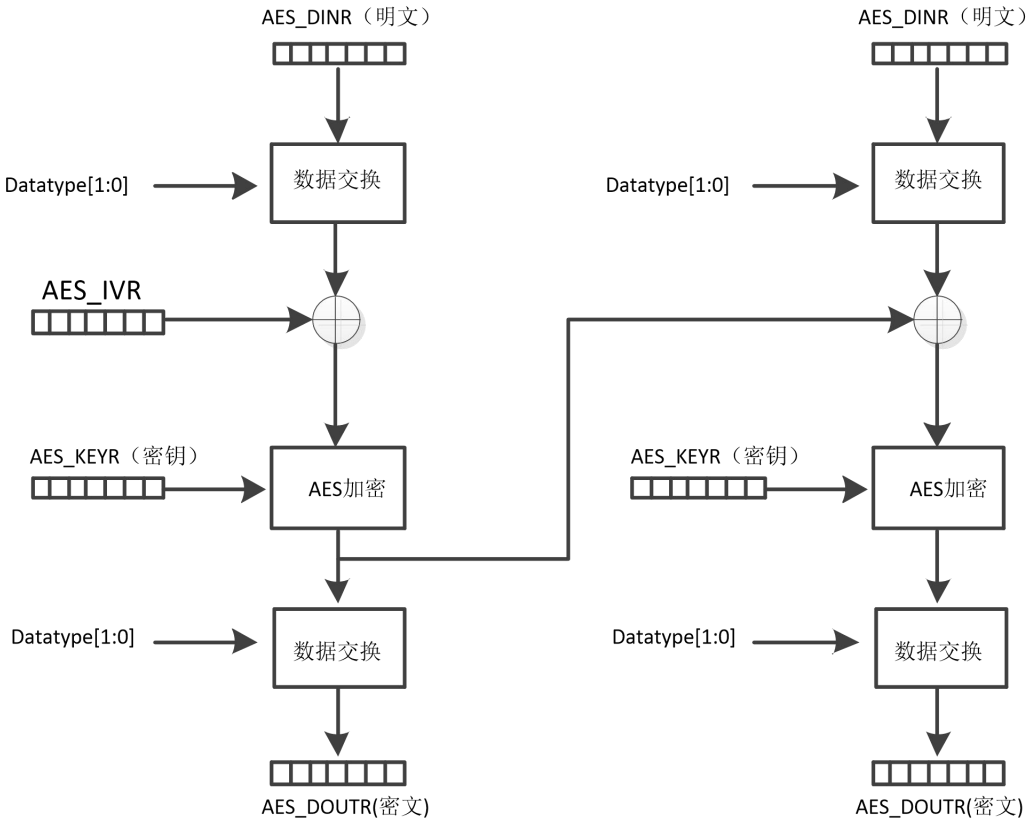


图 15-3 CBC 加密过程

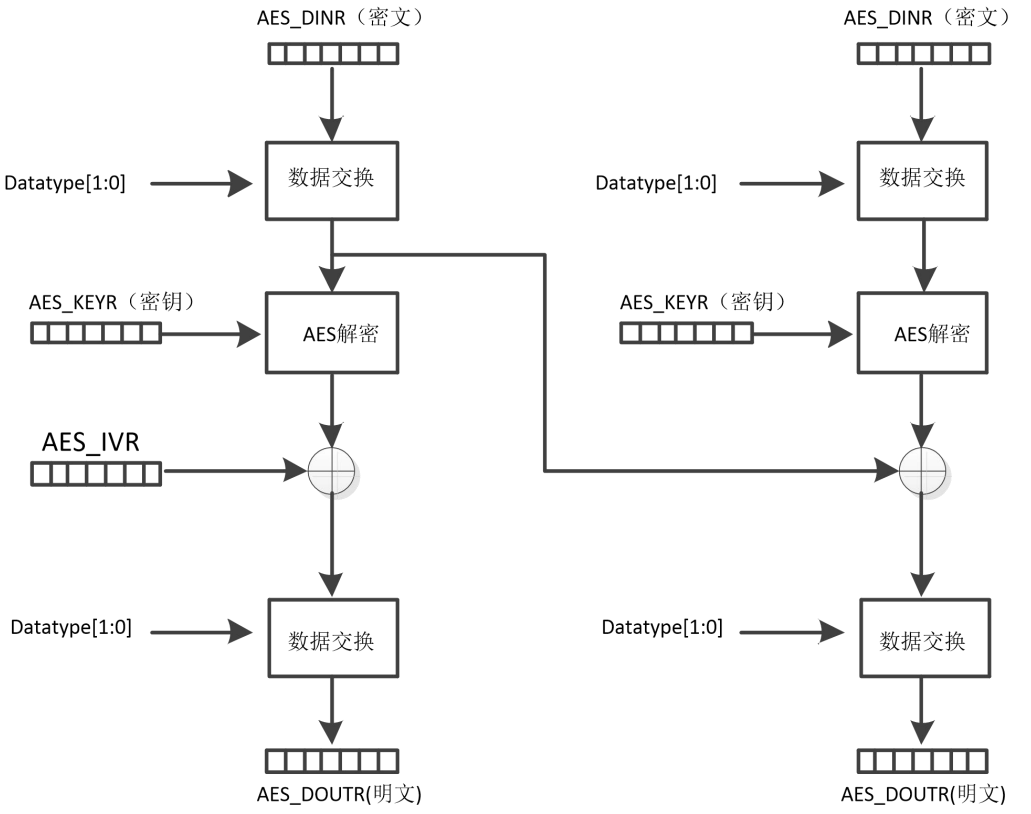


图 15-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

15.3.3 暂停模式

如果一个更高优先级的数据需要处理，当前的数据运算是可以暂停的。暂停的数据处理在加解密运算模式下都能够恢复。仅在CPU参与的模式下可用，DMA模式下不可用。

正确的工作流程为：数据在一个block的结果被读完后暂停。

通过对EN bit写0暂停AES。软件读AES_IVRx寄存器中的值并存储，在恢复运算时该值需要被写入AES_IVRx寄存器。

流程如图15-5所示



图 15-5 暂停模式流程

15.3.4 CTR 模式

该模式下，一个32bit的计数器和一个随机数被用作加解密模块的输入。结果与明文数据进行异或。流程如图15-6和图15-7所示。

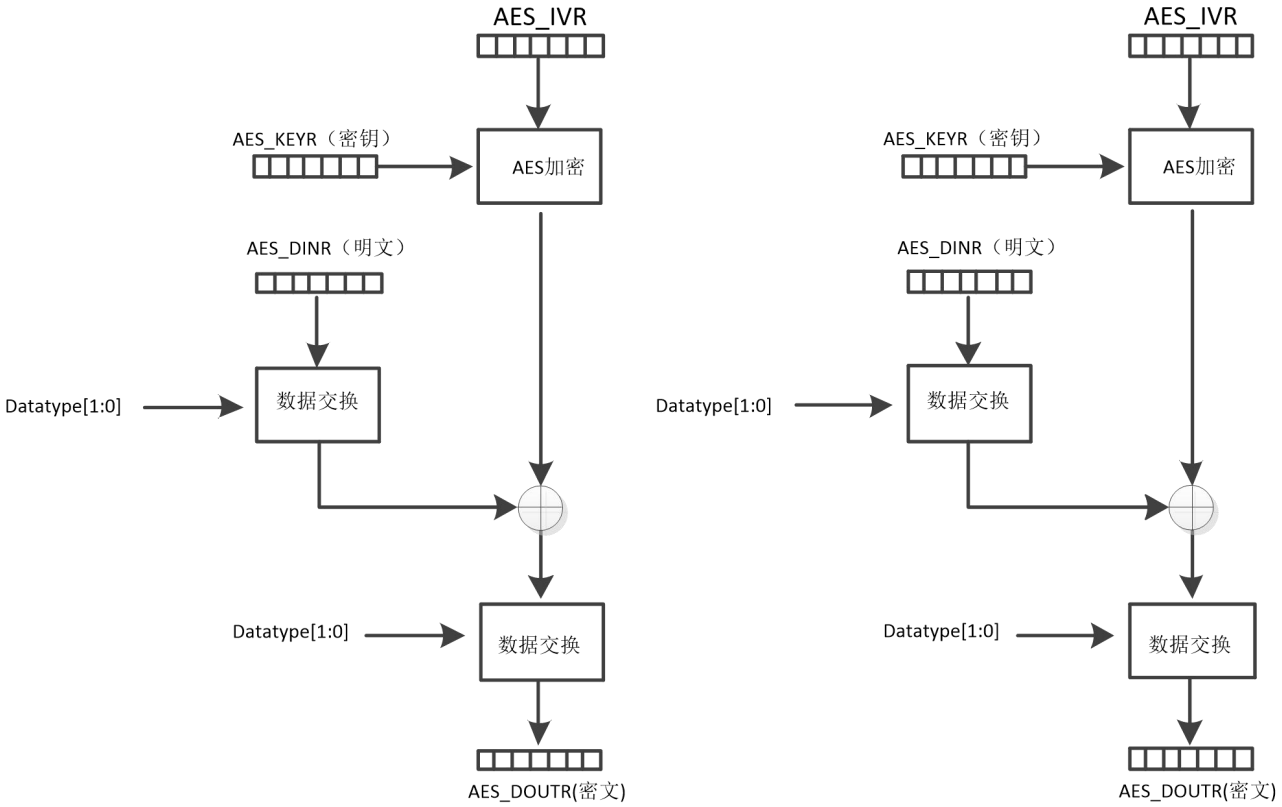


图 15-6 CTR 加密流程

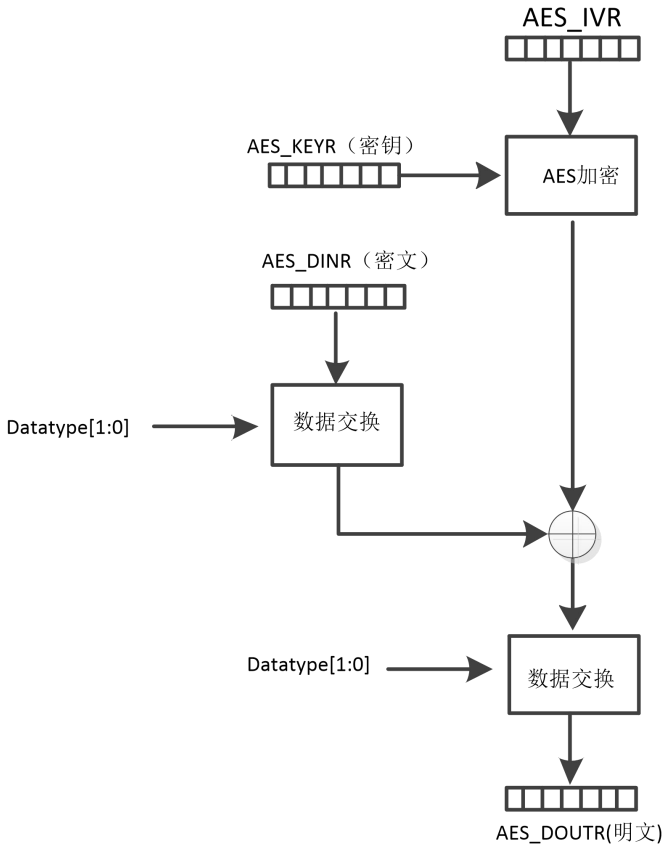


图 15-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图15-8所示



图 15-8 32 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

15.3.5 CTR 模式下的暂停模式

与CBC下暂停模式类似。参考CBC下暂停模式。

15.3.6 GCM 模式

具体可以参考文档The Galois/Counter Mode of Operation (GCM)

GCM的加密按照以下公式定义：

$$
\begin{aligned}
H &= E(K, 0^{128}) \\
Y_0 &= \begin{cases} IV \| 0^{31}1 & \text{if len}(IV) = 96 \\ \text{GHASH}(H, \{\}, IV) & \text{otherwise.} \end{cases} \\
Y_i &= \text{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \\
C_i &= P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n-1 \\
C_n^* &= P_n^* \oplus \text{MSB}_u(E(K, Y_n)) \\
T &= \text{MSB}_t(\text{GHASH}(H, A, C) \oplus E(K, Y_0))
\end{aligned}
$$

其中GHASH函数的定义为$\text{GHASH}(H, A, C) = X_{m+n+1}$，其中X的定义为

$$
X_i = \begin{cases}
0 & \text{for } i = 0 \\
(X_{i-1} \oplus A_i) \cdot H & \text{for } i = 1, \ldots, m-1 \\
(X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H & \text{for } i = m \\
(X_{i-1} \oplus C_i) \cdot H & \text{for } i = m+1, \ldots, m+n-1 \\
(X_{m+n-1} \oplus (C_m^* \| 0^{128-u})) \cdot H & \text{for } i = m+n \\
(X_{m+n} \oplus (\text{len}(A) \| \text{len}(C))) \cdot H & \text{for } i = m+n+1.
\end{cases}
$$

GCM模式的加解密流程如图15-9，图15-10所示。

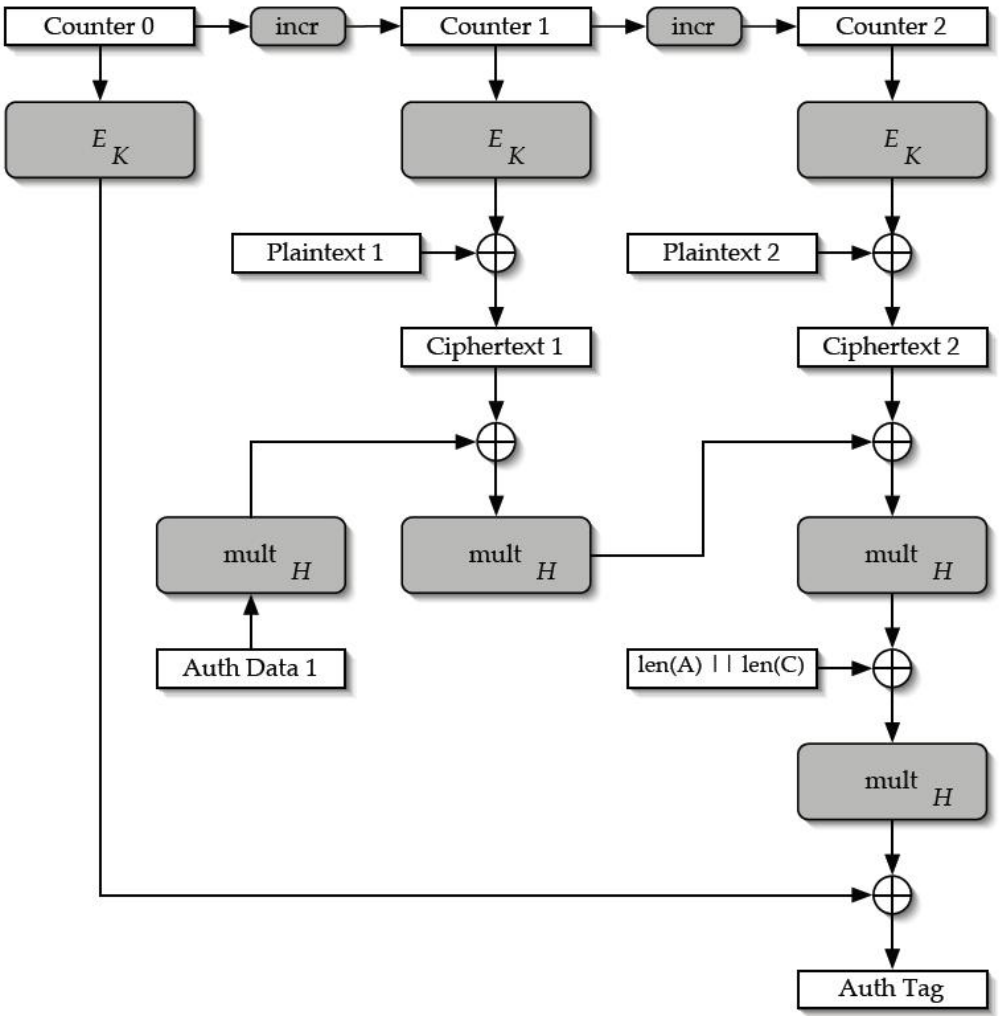


图 15-9 GCM 加密流程

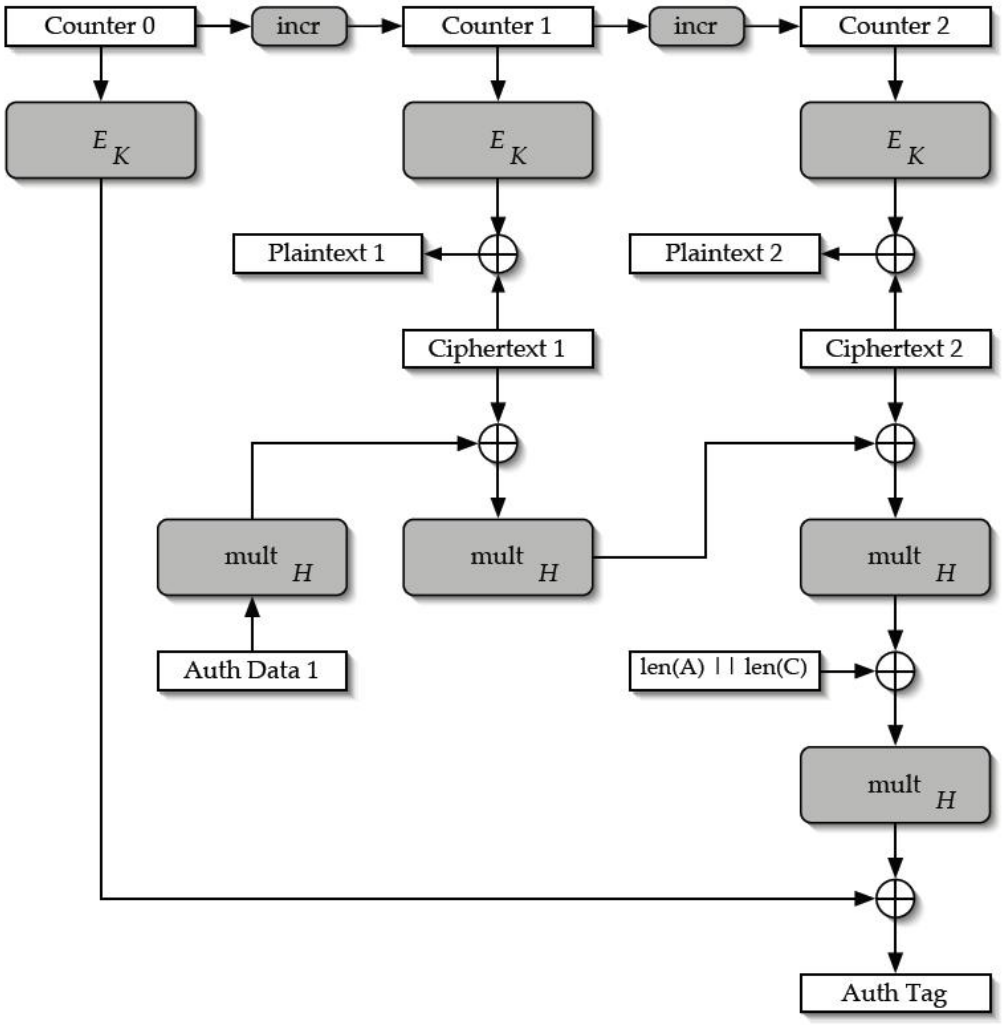


图 15-10 GCM 解密流程

图中Ek表示AES加密模块。multH模块是一个GF（2^128）域上的乘法。Incr表示计数器加一。

GCM模式由软件配合实现，硬件提供一个AES模块和multH模块供软件调度。GCM模式加解密的过程与CTR模式相同。认证过程通过软件调度multH模块实现。

15.3.7 MultH 模块

GF（2^128）上的乘法使用如下算法实现。

Algorithm 1 Multiplication in $GF(2^{128})$. Computes the value of $Z = X \cdot Y$, where X, Y and $Z \in GF(2^{128})$.

$Z \leftarrow 0, V \leftarrow X$
for $i = 0$ to 127 **do**
  if $Y_i = 1$ **then**
    $Z \leftarrow Z \oplus V$
  end if
  if $V_{127} = 0$ **then**
    $V \leftarrow \text{rightshift}(V)$
  else
    $V \leftarrow \text{rightshift}(V) \oplus R$
  end if
end for
return Z

MultH模块的输入输出寄存器复用AES的寄存器。模块框图如图15-11所示。

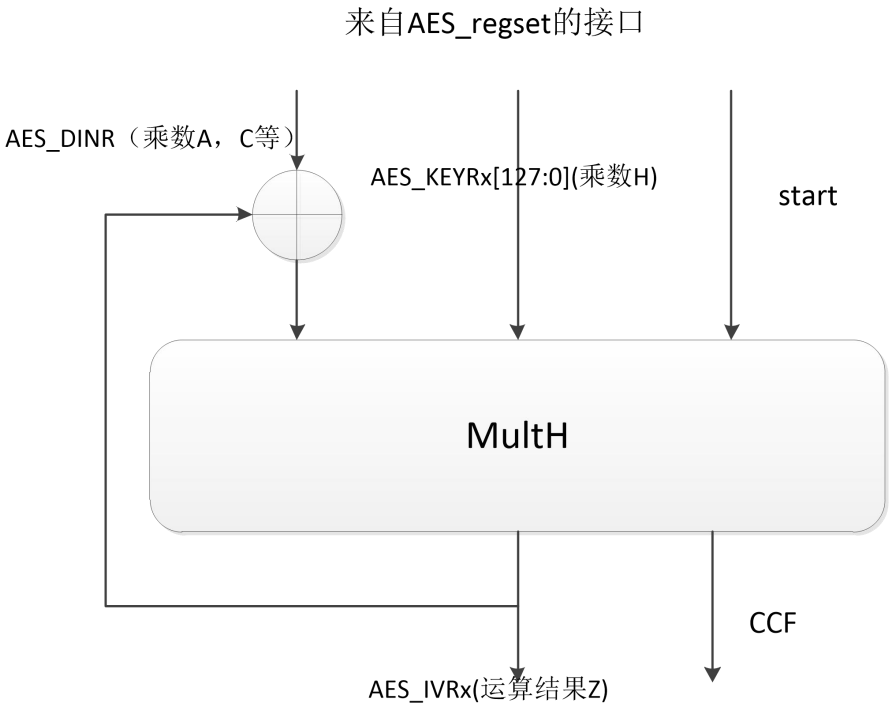


图 15-11 multH 模块框图

multH模块的输入寄存器复用AES的输入寄存器AES_DINR和AES_KEYx的低128bit。输出寄存器复用AES_IVR寄存器。使用时配置CHMOD[1:0]寄存器为MultH模式，接着配置好AES_KEYx和AES_IVR寄存器输入和输出各128bit，使能EN，向AES_DINR输入数据，等待CCF置起即计算完成。

15.3.8 推荐的 GCM 流程

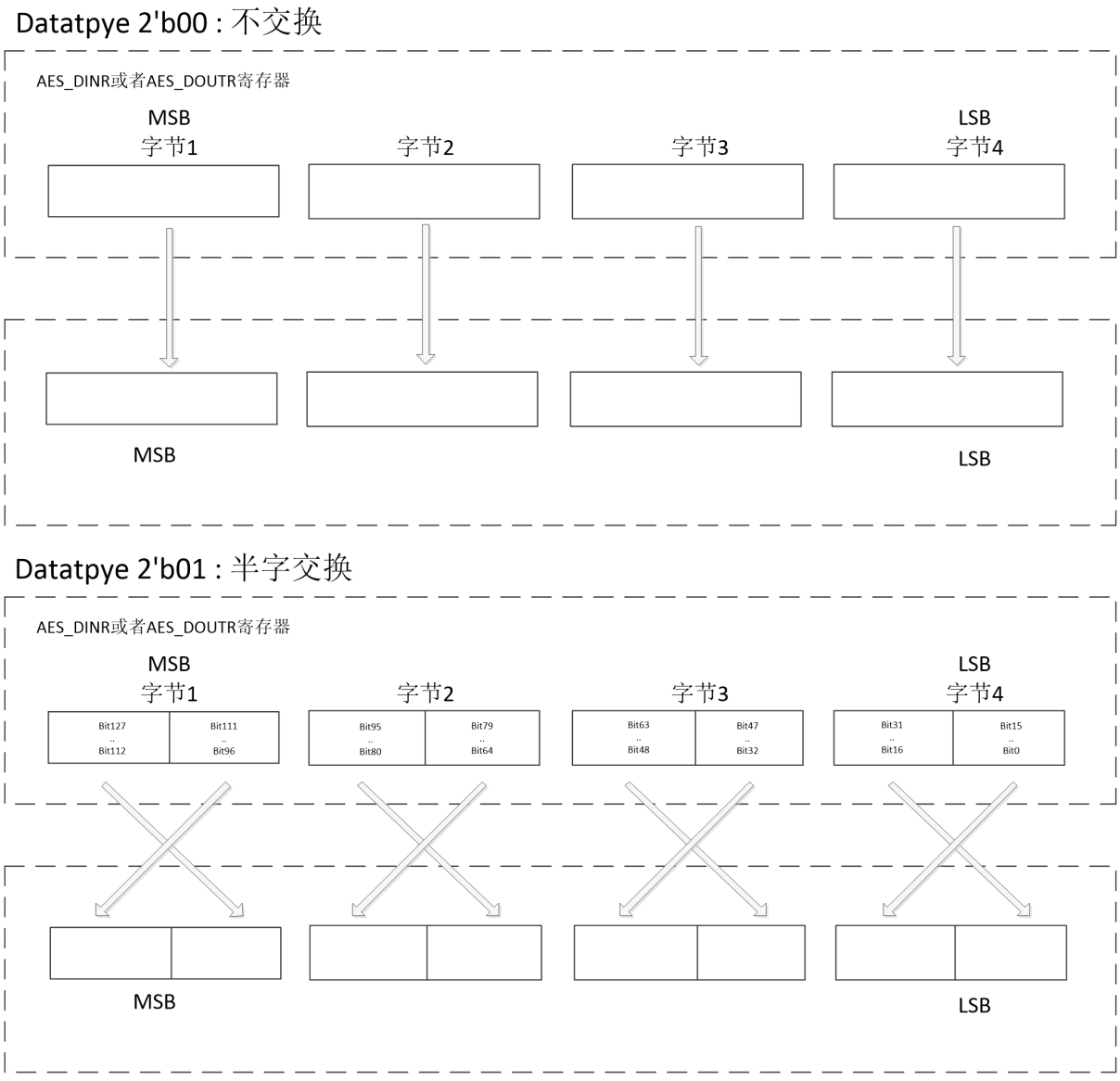
GCM模式的实现需要软硬件配合，本文档提供一种推荐的使用方法。

GCM模式的加解密过程和CTR模式相同。认证过程时仅使用MultH模块而不用AES加解密。

- 调用一次AES模块计算H。并存储。
- 调用一次AES模块计算E（K，Y0），并存储。
- 使用CTR模式开始连续数据的AES加解密操作。IV寄存器初值为Y1
- 使用multH模块连续计算GHASH结果
- 最终GHASH的结果异或上E（K，Y0）即可计算得到tag的值。

15.4 数据类型

AES一次读写32bit数据，每32bit可以根据DATATYPE[1:0]寄存器的设置按照不同的方式交换数据的顺序。如图15-12所示。



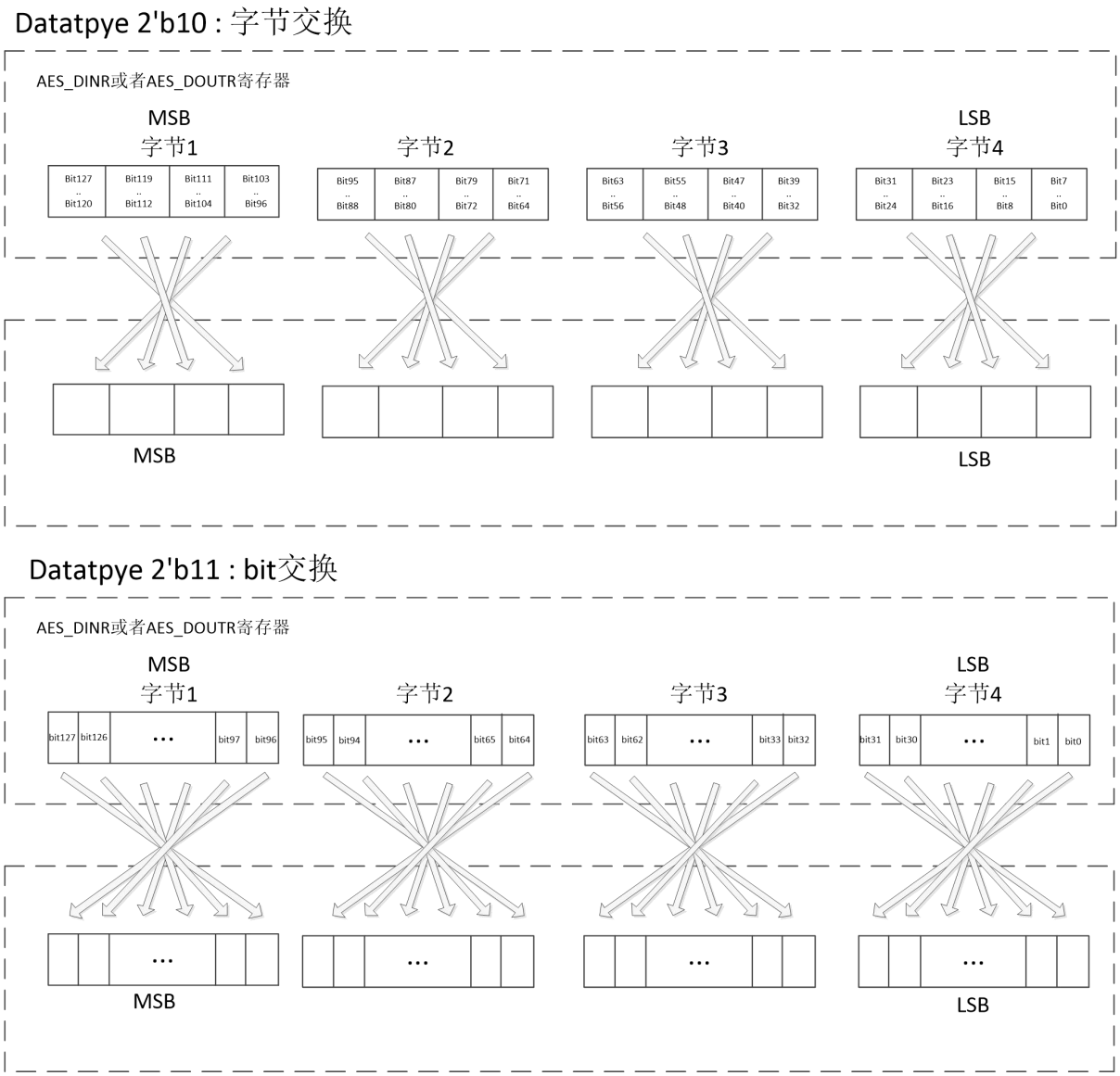


图 15-12 根据数据类型存储数据的示意图

15.5 工作流程

15.5.1 模式 1：加密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=00，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYRx寄存器，CTR和CBC模式下写AES_IVRx寄存器
- 写EN=1，使能AES
- 写AES_DIR 寄存器4次
- 等待CCF标志置起

- 从AES_DOR分4次读出加密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行加密

步骤5-7如图所示。

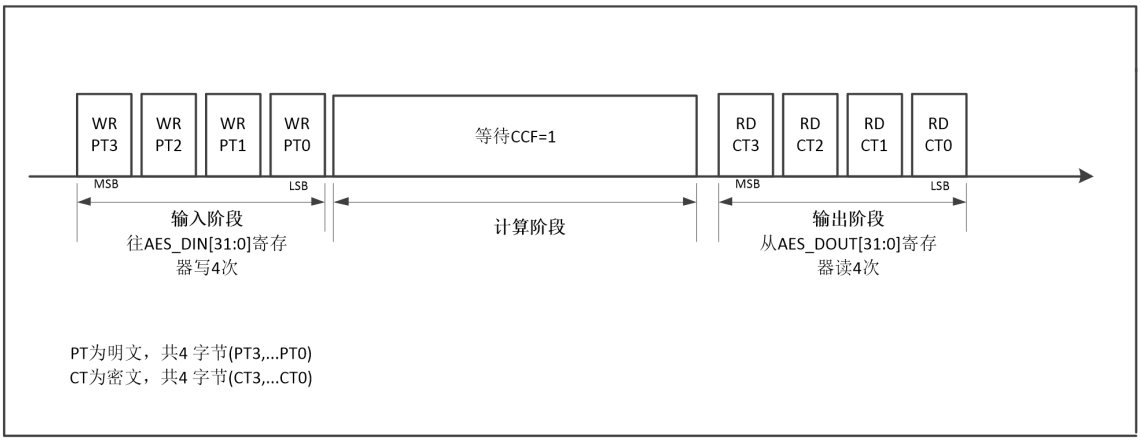


图 15-13 模式 1：加密流程

15.5.2 模式 2：密钥扩展

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=01，CHMOD[1:0]寄存器的值不关心。
- 写AES_KEYx寄存器。
- 写EN=1，使能AES
- 等待CCF标志置起
- 清除CCF标志，扩展完的key自动写回AES_KEYx寄存器。如果需要的话可以读取AES_KEYx寄存器获取结果。想要重新计算扩展密钥，重复步骤3,4,5,6。

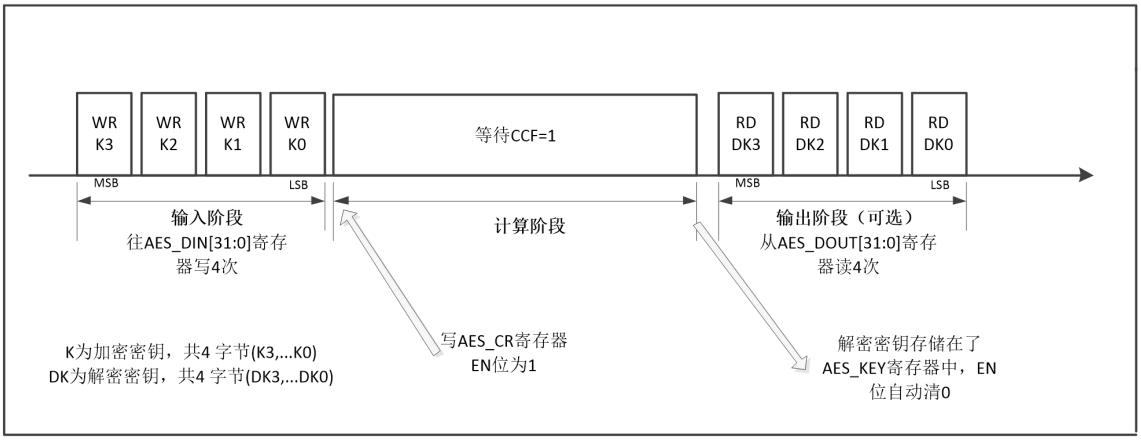


图 15-14 模式 2 示意图

15.5.3 模式 3：解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=10，设置流数据处理模式寄存器CHMOD[1:0]
- 写AES_KEYx寄存器（如果已经通过模式2计算得到了扩展密钥则可跳过这个步骤），CTR和CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DIR 寄存器4次
- 等待CCF标志置起
- 从AES_DOR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

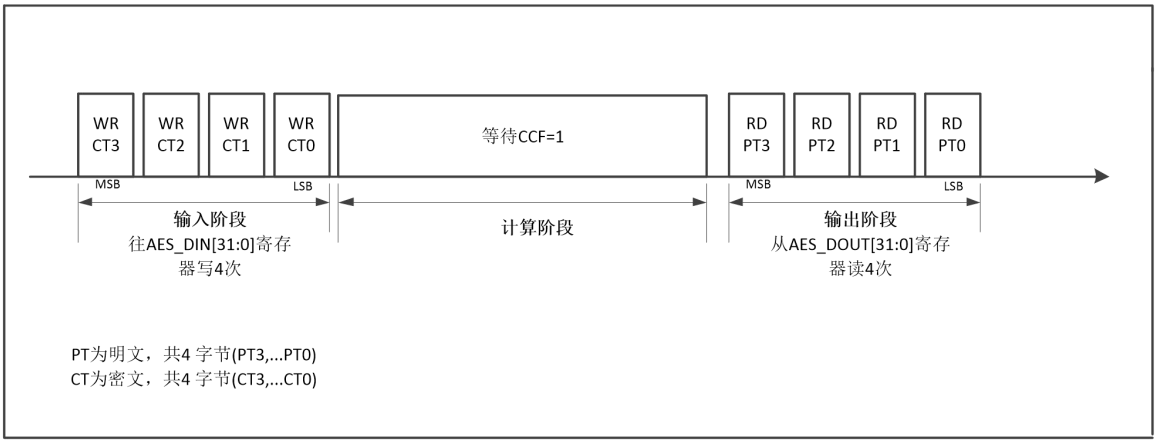


图 15-15 模式 3 示意图

15.5.4 模式 4：密钥扩展+解密

- 复位EN 重置AES模块
- 设置模式寄存器mode[1:0]=11，设置流数据处理模式寄存器CHMOD[1:0]。该模式在CTR模式下被禁止使用。如果设置mode[1:0]=11，CHMOD[1:0]=10，将强制进入CTR解密模式。
- 写AES_KEYx寄存器，CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DIR 寄存器4次
- 等待CCF标志置起
- 从AES_DOR分4次读出解密结果
- 对于同一个key，重复步骤5,6,7对接下来的128bit block进行解密

注意：该模式下AES_KEYx寄存器内存储的一直是加密密钥，扩展密钥每次都会在内部被重新计算而不会被存储到AES_KEYx寄存器中。

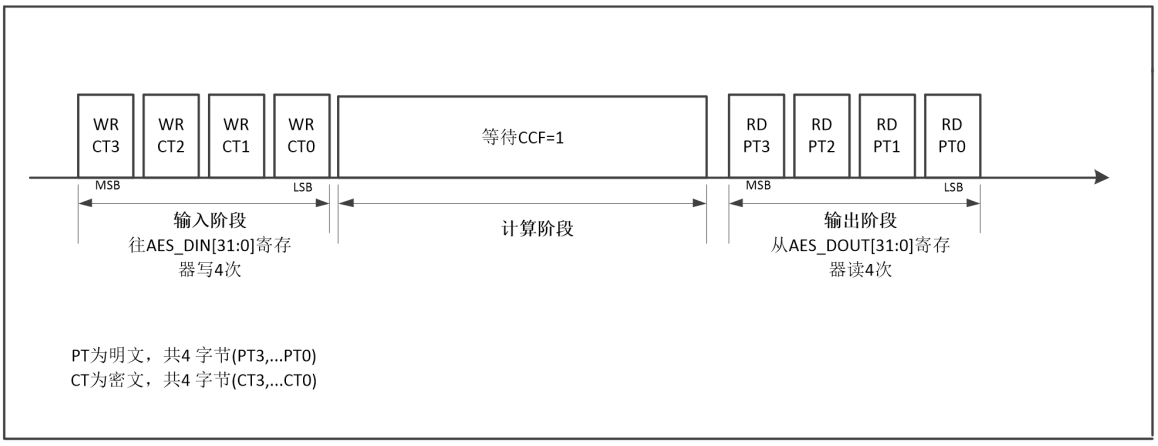


图 15-16 模式 4 示意图

15.5.5 使用 MultH 模块

- 复位EN 重置AES模块。
- 设置流数据处理模式寄存器CHMOD[1:0]=11。该模式下mode[1:0]寄存器的值不能够是01配置在模式2：密钥扩展下。同时配置mode[1:0]=01和CHMOD[1:0]=11会由于mode寄存器优先值更高而进行密钥扩展操作。
- 写AES_Hx寄存器，若为第一轮计算，则初始值为0x00000000。
- 写EN=1，使能multH模块。
- 写AES_DIR 寄存器4次。MultH模块会把上一次的计算结果异或上AES_DIR寄存器输入的值做为multH模块的一个乘数。所以把上一轮的计算结果赋为0x00000000，即实现了直接把AES_DIR寄存器输入的值做为multH模块的一个乘数的功能。
- 等待CCF标志置起
- 从AES_IVR寄存器中读出计算结果，可以通过IVRSWAP寄存器控制读取的数据是否交换顺序后再放到总线上
- 对于同一个H，重复步骤5,6进行连续计算。即可实现了一个GMAC的功能。

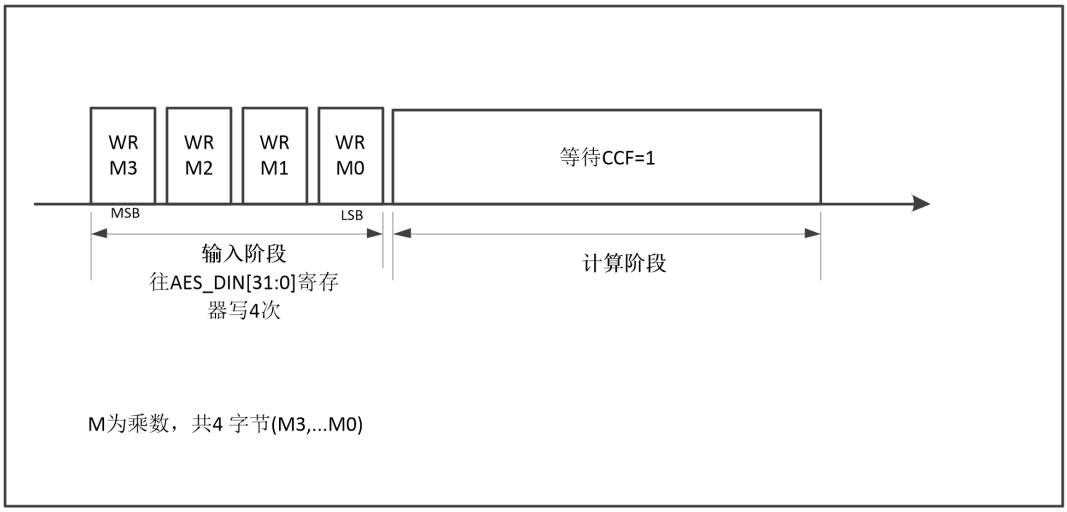


图 15-17 multH 模块使用流程示意图

15.6 DMA 接口

- 一个输入的请求通道：当DMAIEN为1时，每当AES在需要输入数据写入AES_DIR寄存器的时候发起一个DMA的请求。
- 一个输入的请求通道：当DMAOEN为1时，每当AES在需要从AES_DOR寄存器输出数据的时候发起一个DMA的请求。

每个阶段产生4次请求，在AES模块被关闭前对DMA的请求会一直产生。AES计算完128比特后就自动取新数据进行下次计算。

注意：DMA模式下DMAOEN=1时，CCF标志可能为高。

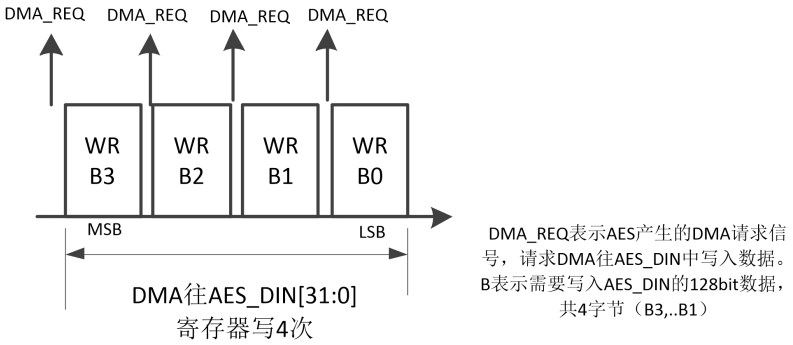


图 15-18 输入时 DMA 请求和数据传输示意图

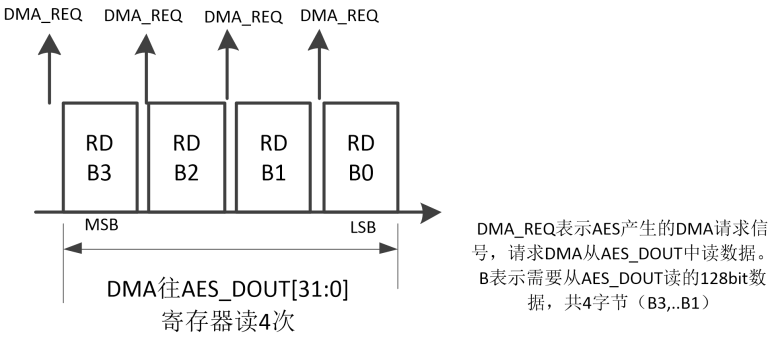


图 15-19 输出时 DMA 请求和数据传输示意图

15.6.1 MultH 模块与 DMA 间接口

MultH计算也可以通过DMA计算。当DMAIEN为1及CHMOD[1:0]=11时，每当AES在需要输入数据写入AES_DINR寄存器的时候发起一个DMA的请求。该模式下配置DMAOEN=1无效AES不会产生DMA请求。

15.7 错误标志

在计算和输入阶段发生一个读操作，置起RDERR。

在计算和输出阶段发生一个写操作，置起WRERR。

产生错误后AES模块不会被硬件自动停止，会像正常一样继续运算。

15.8 寄存器

模块基地址：0x4001_B800

offset 地址	名称	符号
0x00	AES 控制寄存器（AES Control Register）	AES_CR
0x04	AES 中断使能寄存器（AES Interrupt Enable Register）	AES_IER
0x08	AES 中断标志寄存器（AES Interrupt Status Register）	AES_ISR
0x0C	AES 数据输入寄存器（AES Data Input Register）	AES_DIR
0x10	AES 数据输出寄存器（AES Data Output Register）	AES_DOR
0x14	AES 秘钥寄存器 0（AES Key Register 0）	AES_KEY0
0x18	AES 秘钥寄存器 1（AES Key Register 1）	AES_KEY1
0x1C	AES 秘钥寄存器 2（AES Key Register 2）	AES_KEY2
0x20	AES 秘钥寄存器 3（AES Key Register 3）	AES_KEY3
0x24	AES 秘钥寄存器 4（AES Key Register 4）	AES_KEY4
0x28	AES 秘钥寄存器 5（AES Key Register 5）	AES_KEY5
0x2C	AES 秘钥寄存器 6（AES Key Register 6）	AES_KEY6
0x30	AES 秘钥寄存器 7（AES Key Register 7）	AES_KEY7
0x34	AES 初始向量寄存器 0（AES Initial Vector Register 0）	AES_IVR0
0x38	AES 初始向量寄存器 1（AES Initial Vector Register 1）	AES_IVR1
0x3C	AES 初始向量寄存器 2（AES Initial Vector Register 2）	AES_IVR2
0x40	AES 初始向量寄存器 3（AES Initial Vector Register 3）	AES_IVR3
0x44	AES MultH 参数寄存器 0（AES MultH parameter Register 0）	AES_H0
0x48	AES MultH 参数寄存器 1（AES MultH parameter Register 1）	AES_H1
0x4C	AES MultH 参数寄存器 2（AES MultH parameter Register 2）	AES_H2
0x50	AES MultH 参数寄存器 3（AES MultH parameter Register 3）	AES_H3

15.8.1 AES 控制寄存器（AES_CR）

名称	AES_CR

offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	KEYLEN		DMAOEN	DMAIEN	IVRSWAP		-
位权限	U-0	R/W-00		R/W-0	R/W-0	R/W-00		U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	CHMOD		MODE		DATATYP		EN
位权限	U-0	R/W-00		R/W-00		R/W-00		R/W-0

Bit	助记符	功能描述
31:15	--	RFU：未实现，读为 0
14:13	KEYLEN	AES 加密密钥长度，AESEN=1 时不可修改。 (Key Length) 00：128bit 01：192bit 10：256bit 11：保留
12	DMAOEN	DMA 数据自动读出使能 (DMA output enable) 0：不开启 1：开启 该位置位后在模式 1，模式 3 和模式 4 下 AES 模块会自动产生 AES->RAM 的传输请求。模式 2 下不会产生。
11	DMAIEN	开启 DMA 数据自动写入使能 (DMA input enable) 0：不开启 1：开启 该位设置为 1 后在模式 1，模式 3 和模式 4 以及 MultH 模式下 AES 模块会自动产生 RAM->AES 的传输请求。模式 2 下不会产生。
10:9	IVRSWAP	IVR 寄存器读出交换功能 (IVR register read-out swapping) 此寄存器仅针对 IVR 寄存器读取，读出的数据在放到系统总线上之前先经过顺序交换处理 00：32bit 数据不交换 01：16bit 数据半字交换 10：8bit 数据字节交换 11：1bit 数据比特交换
8:7	--	RFU：未实现，读为 0
6:5	CHMOD	AES 数据流处理模式，AESEN=1 时不可修改。 (Cipher Mode) 00：ECB 01：CBC 10：CTR 11：使用 MultH 模块
4:3	MODE	AES 工作模式，AESEN=1 时不可修改。

Bit	助记符	功能描述
		(operation MODE) 00：模式 1：加密 01：模式 2：密钥扩展 10：模式 3：解密 11：模式 4：密钥扩展+解密 CTR 模式下配置成模式 4 将自动进入 CTR 的解密模式。即在 CHMOD=2'b10 时配置 MODE=2'b11，AES 将按照 MODE=2'b10 的情形执行。
2:1	DATATYP	选择数据类型，AESEN=1 时不可修改。具体交换规则可参考 AES 数据类型章节。 (Data type) 00：32bit 数据不交换 01：16bit 数据半字交换 10：8bit 数据字节交换 11：1bit 数据比特交换
0	EN	AES 使能 (AES enable) 0：不使能 1：使能 在任何时候清除 AESEN 位都能够复位 AES 模块 在模式 2 下该位会在一次计算完成后硬件自动清 0

15.8.2 AES 中断使能寄存器（AES_IER）

名称	AES_IER							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR_IE	RDERR_IE	CCF_IE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2	WRERR_IE	写错误中断使能，1 有效。(Write Error interrupt enable)
1	RDERR_IE	读错误中断使能，1 有效。(Read Error interrupt enable)
0	CCF_IE	AES 计算完成中断使能，1 有效。(Cipher Complete Interrupt enable)

15.8.3 AES 中断标志寄存器（AES_ISR）

名称	AES_ISR							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WRERR	RDERR	CCF
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	WRERR	写错误标志：在计算或输出阶段发生写操作时置位，软件写 1 清零
1	RDERR	读错误标志：在计算或输入阶段发生读操作时置位，软件写 1 清零
0	CCF	AES 计算完成标志，软件写 1 清零 1：计算完成 0：计算没有完成

15.8.4 AES 数据输入寄存器（AES_DIR）

名称	AES_DIR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	DIN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	DIN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	DIN[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	DIN	数据输入寄存器，当 AES 需要输入加解密数据时，应该往该寄存器连续写 4 次。(AES Data Input) 模式 1（加密）：把明文从 MSB 到 LSB 分 4 次写入。 模式 2（密钥扩展）：无需使用数据输入寄存器 模式 3 和模式 4（解密）：把密文从 MSB 到 LSB 分 4 次写入。

Bit	助记符	功能描述
		MultH 模式：把乘数 A 或 C 从 MSB 到 LSB 分 4 次写入。

15.8.5 AES 数据输出寄存器（AES_DOR）

名称	AES_DOR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DOUT[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	DOUT[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	DOUT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	DOUT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:0	DOUT	数据输出寄存器，当 AES 计算完成后，可以分四次读出加解密的结果。(AES Data Output) 模式 1（加密）：把密文从 MSB 到 LSB 分 4 次读出。 模式 2（密钥扩展）：无需使用数据输输出寄存器 模式 3 和模式 4（解密）：把明文从 MSB 到 LSB 分 4 次输出。 MultH 模式：运算结果存储在 IVR 寄存器中，无需读取 AES_DOR 寄存器。

15.8.6 AES 秘钥寄存器（AES_KEYx）

名称	AES_KEYx(x=0,1,2,3,4,5,6,7)							
offset	0x14 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEYx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	KEYx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	KEYx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	KEYx[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	KEYx	AES 运算秘钥，最长 256bit，AESKEY0 存放秘钥最低 32bit，

Bit	助记符	功能描述
		AESLKEY7 存放秘钥最高 32bit。(AES Key)

15.8.7 AES 初始向量寄存器（AES_IVRx）

名称	AES_IVRx(x=0,1,2,3)							
offset	0x34 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IVRx[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	IVRx[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	IVRx[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	IVRx[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	IVRx	AES 运算 128bit 初始向量，在 MultH 模式下保存运算结果。 (AES Initial Vector Registers)

15.8.8 AES MultH 参数寄存器（AES_Hx）

名称	AES_Hx(x=0,1,2,3)							
地址	0x44 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Hx[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Hx[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Hx[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Hx[7:0]							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:0	Hx	MultH 运算 128bit 输入 H 参数 (H Parameter) H0 保存 H[31:0]，H3 保存 H[127:96]

16 随机数发生器（TRNG）

16.1 概述

FM33FR0使用2个Galois真随机噪声源作为真随机数种子，配合简单在线检测（32位全0全1检测）、LFSR后处理、伪随机LFSR共同组成芯片的随机数发生器。

TRNG的启动测试和完整的在线测试功能需要固件实现。

Galois噪声源的采样和LFSR建议使用RCHF 4MHz时钟。两次取32bit随机数之间的间隔不得小于32个时钟周期。

真随机数发生器通过了FIPS PUB140-2测试，成功率99.9%。

16.2 功能描述

16.2.1 随机数产生

下图为真随机数发生器结构框图。

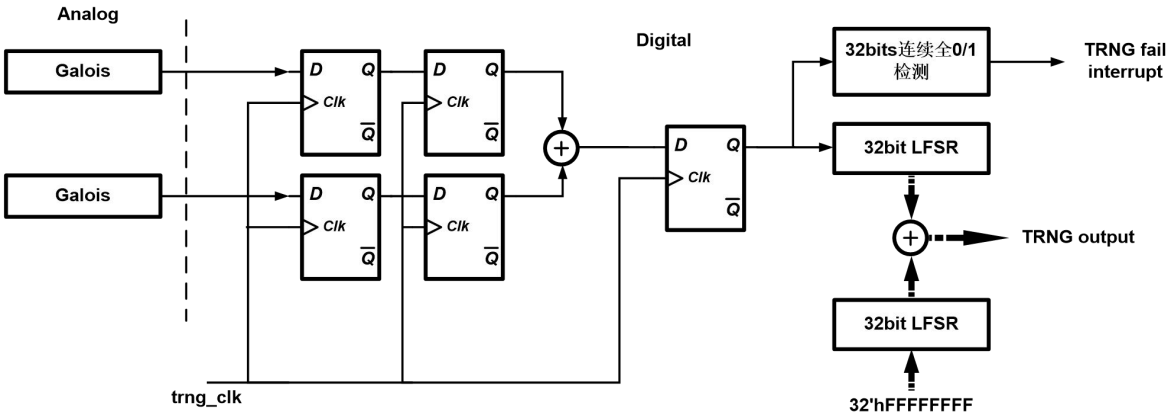


图 16-1 真随机数模块框图

真随机噪声源为2个Galois环振，Galois环振输出在数字电路内部异或并使用系统时钟采样，然后进行LFSR后处理。LFSR后处理之前经过随机数在线检测，如果发现连续32bit全0或全1的情况，则产生TRNG失效报警中断。同时为了避免小概率的真随机数性能不良情况，另外使用一组LFSR以32'hFFFFFFFF为初始值，与后处理LFSR同步运算，并以两组LFSR按位异或后的结果作为最终的32bit随机数输出。

16.2.2 工作时钟

随机数发生器的工作时钟采用RCHF的分频时钟，独立于APBCLK。为了保证随机数质量，一般建议应用使用4M时钟作为随机数工作时钟，并根据4M目标频率配置CMU模块中随机数工作时钟分频寄存器（OPCCR2.RNGPRSC）。工作时钟示意图如下：

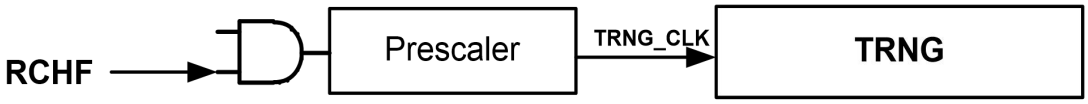


图 16-2 真随机数模块工作时钟

16.2.3 随机数读取

当随机数模块被使能后，真随机噪声源和LFSR后处理模块同时开始工作。软件通过读取

RNG_DOR.OUT寄存器，每次读出32bit随机数。由于LFSR循环移位周期是32cycle，为保证随机数质量，应用应保证两次读取RNG_DOR.OUT之间的间隔大于32个TRNG_CLK周期。

举例来说，假设TRNG_CLK为4MHz，则两次读取RNG_DOR.OUT的间隔不应小于8us。

16.2.4 CRC 运算

用作随机数后处理的LFSR也可用于进行CRC计算。

在进行CRC运算时，两组32bit LFSR分别作为输入数据寄存器和CRC运算寄存器，一次可以运算32bit数据的CRC结果。CRC运算前CPU需查询当前LFSR是否被占用，如LFSR空闲，方可以使用CRC功能。

CPU一旦启动CRC运算，LFSR自动置为复位值，随后进行32bit运算，运算结束后清除CRC启动寄存器，不产生中断；软件启动CRC后应连续查询启动寄存器状态，直到运算结束后再读取结果。

CRC多项式：

CRC32=X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+X0

软件操作流程：

- 查询RBUSY，确认LFSR不在运行中
- 将待运算数据写入CRCDIR.IN
- 置位CRCCR.EN
- 查询并等待CRCCR.EN被清零
- 从DOR.OUT读出运算结果

16.3 寄存器

模块偏移地址：0x4001_BC00

offset 地址	名称	符号
0x00	随机数控制寄存器	RNG_CR
0x04	随机数/CRC 结果输出寄存器	RNG_DOR
0x10	随机数标志寄存器	RNG_SR
0x14	CRC 控制寄存器	RNG_CRCCR
0x18	CRC 输入数据寄存器	RNG_CRCDIR
0x1C	CRC 标志寄存器	RNG_CRCSR

16.3.1 随机数控制寄存器

名称	RNG_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	RNG 使能寄存器，软件写 1 启动，完成随机数产生后自动清零 1：启动 RNG 0：关闭 RNG

16.3.2 随机数/CRC 结果输出寄存器

名称	**RNG_DOR**							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	OUT[31:24]							
位权限	R							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OUT[23:16]							
位权限	R							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OUT[15:8]							
位权限	R							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OUT[7:0]							
位权限	R							

Bit	助记符	功能描述
31:0	OUT	随机数生成结果或 CRC 运算结果寄存器

16.3.3 RNG 标志寄存器

名称	RNG_SR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RBUSY	RNF
位权限	U-0						R-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	RBUSY	RNG 忙闲标志 1：RNG 正在生成随机数，此时软件不得使用 CRC 功能 0：RNG 空闲
0	RNF	随机数生成失败标志 1：随机数未能通过质量检测 0：随机数通过质量检测

16.3.4 CRC 控制寄存器

名称	RNG_CRCCR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN

位权限	U-0	R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	CRC 使能控制寄存器，软件写 1 启动 CRC，运算完成后硬件自动清零 1：CRC 使能 0：CRC 关闭

16.3.5 CRC 输入数据寄存器

名称	RNG_CRCDIR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IN[31:24]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IN[23:16]							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IN[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	IN	CRC 运算数据输入寄存器

16.3.6 CRC 标志寄存器

名称	RNG_CRCSR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							DONE
位权限	U-0							R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	DONE	CRC 计算完成标志，软件写 0 清零 1：CRC 计算完成 0：CRC 计算未完成

17 模拟比较器（Comparator）

17.1 概述

芯片集成2个比较器，支持以下特性：

- 1个rail-to-rail比较器，支持轨到轨输入，多种功耗模式
- 1个低功耗比较器，典型功耗200nA
- 灵活的输入选择
  - IO引脚输入
  - 内部基准电压及其分压和升压
- 中断事件可唤醒MCU
- 输出信号可连接到GPIO，也可以作为触发源连接到定时器、ADC
- 内建输入基准Buffer
- 窗口比较器功能
- 可编程的输出数字滤波功能

17.2 结构框图

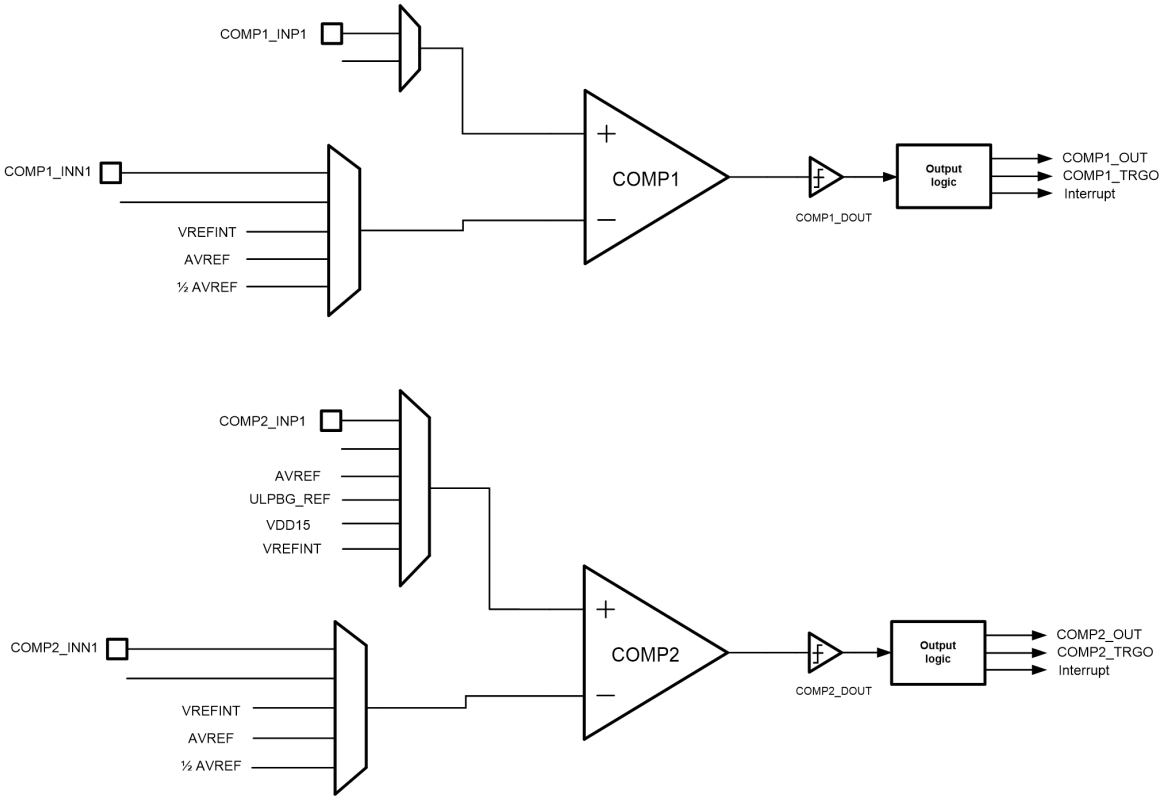


图 17-1 比较器电路框图

比较器结构如上图所示，基准电压入经过BUFFER模块后输出基准电压和基准电压分压，如果比较基准为AVREF本身，则可以关闭buffer节省功耗。

比较器1输入电压范围0~VDD，建立时间小于15us，典型传输延迟小于1us，典型功耗2uA。

比较器2输入电压范围0~VDD-0.7V，建立时间小于10us，传输延迟小于5us，典型功耗200nA。

- COMPx_INPy/COMPx_INNy：连接到GPIO的正负端输入信号
- COMPx_OUT：连接到GPIO的输出信号
- VREFINT：内部产生的基准电压，多档位输出可选，参见VREFINT_VREG章节
- AVREF：1.0V内部基准电压
- VDD15：内部LDO输出，约1.55V
- ULPBG_REF：约1.16V内部基准电压

17.3 功能描述

17.3.1 基本功能

比较器比较正端输入电压和负端输入电压，当正端电压高于负端电压时，输出逻辑高电平，反之输出逻辑低电平。

正端输入电压可以从多个引脚输入选择，负端输入电压可以选择引脚输入或者内部基准电压。

比较器输出的逻辑信号可以经过数字滤波和极性控制后输出，或产生中断信号。

比较器可以配置为快速模式或者低功耗模式，两种模式下比较器工作电流和传输延迟指标不同，应用上可以根据需要选择合适的工作模式。

17.3.2 内部比较基准选择和产生

比较器负端可以选择输入芯片内部比较基准电压，比较基准来自于AVREF及其分压。

当使用AVREF分压时，必须使能分压buffer并选择buffer输入源。通过BUFENB寄存器来使能比较器模块内建的基准源buffer。

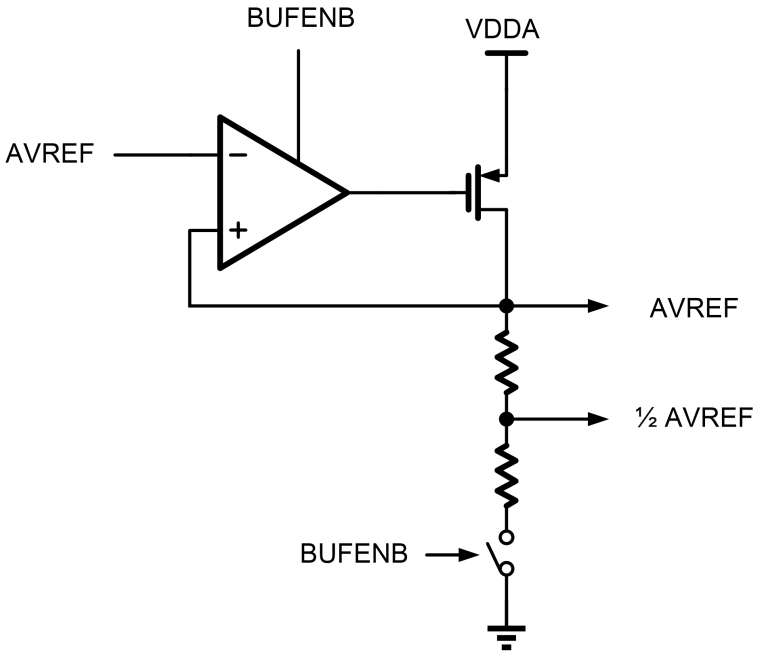


图 17-2 比较器内建基准缓冲器

17.3.3 时钟和复位

比较器模块的寄存器时钟由CMU模块提供，复位控制由RMU模块提供。

在操作比较器模块寄存器之前必须在RMU模块中清除比较器复位，并在CMU中使能比较器工作时钟。

比较器本身工作不依赖于时钟，因此可以在各种低功耗模式下工作，应用仅需在进入低功耗之前完成对比较器的寄存器配置。

17.3.4 引脚和内部信号连接

比较器输入信号可以映射到以下IO引脚上，使用外部引脚输入时需要将对应的GPIO配置为模拟通道功能。

COMP1	引脚	寄存器
COMP1_INP1	PD15	COMP1_CR.V1PSEL=0
COMP1_INN1	PD14	COMP1_CR.V1NSEL=000
COMP1_OUT	PE8	-

表 17-1 比较器 1 引脚列表

COMP2	引脚	寄存器
COMP2_INP1	PA2	COMP2_CR.V2PSEL=000
COMP2_INN1	PA1	COMP2_CR.V2NSEL=000
COMP2_OUT	PC0	-

表 17-2 比较器 2 引脚列表

17.3.5 窗口功能

窗口比较器用于监视输入模拟电压是否处于某个设定的阈值范围之内，当输入信号电平高于高阈值或者低于低阈值时，比较器输出翻转。

窗口比较器功能需要同时使用两个比较器（COMP1和COMP2），被监视的输入电压被同时连接到两个比较器的正端输入，两个比较器的负端输入分别接入高阈值和低阈值电压，两个比较器的输出被异或后输出。窗口模式下有效输出为比较器1的输出信号。

窗口比较器结构如下图所示：

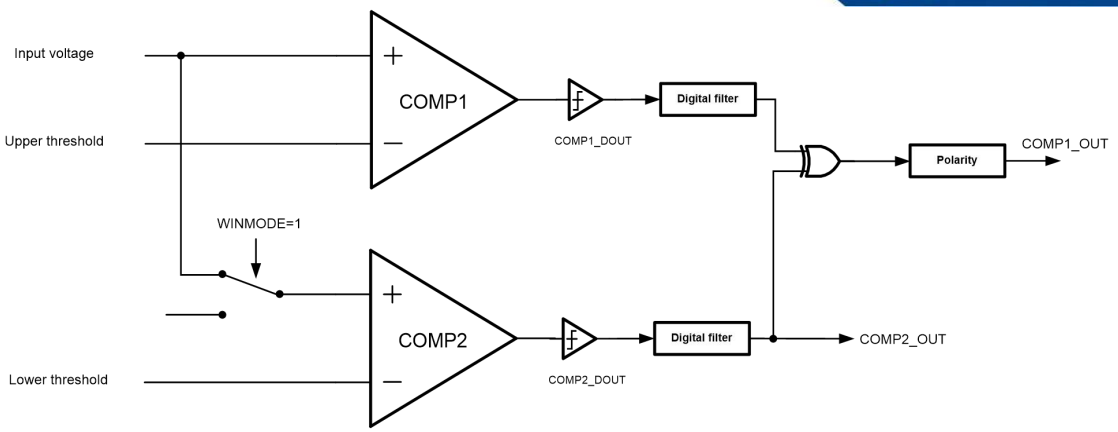


图 17-3 窗口比较器框图

窗口比较器工作时的波形示意图如下（关闭数字滤波，输出极性不取反）。当输入电压超出上下阈值设定的范围时，输出信号翻转，并且产生OOW_IF（Out-Of-Window）中断，而WIN_IF中断在输出信号每次越过阈值时产生。

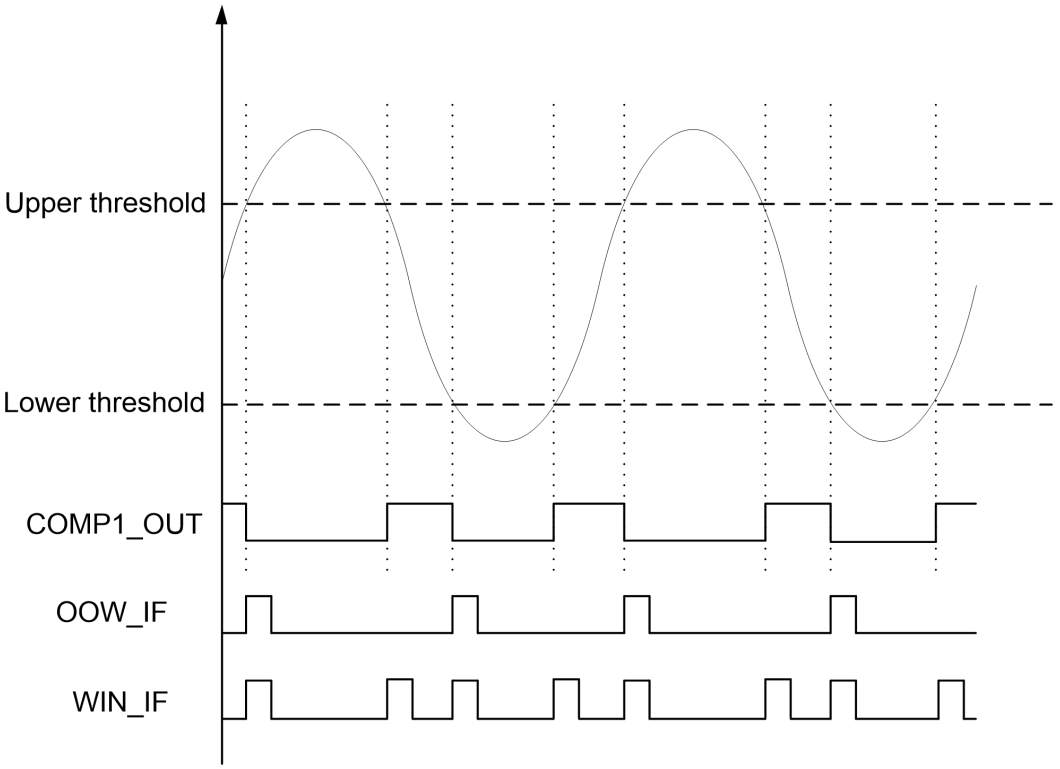


图 17-4 窗口比较器波形示意图

17.3.6 输出数字滤波

比较器输出支持数字滤波功能。滤波方式是采用APBCLK连续采样比较器输出，保持DFLEN定义的周期数为相同电平时，才认为这个电平有效。

在休眠模式下，由于APBCLK关闭，无法使用数字滤波功能，应用中如果需要在休眠模式下使用比较器，软件必须在休眠前关闭比较器数字滤波功能。

下图是DFLEN=3的示意图：

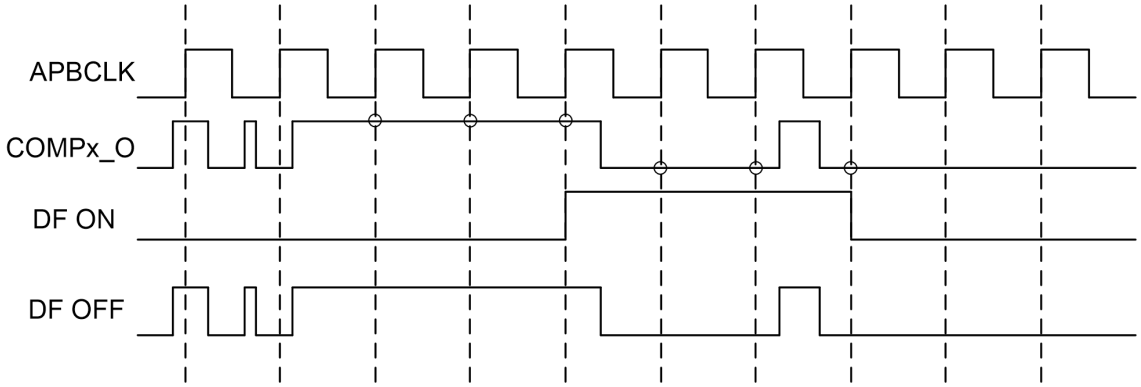


图 17-5 数字滤波（DFLEN=3）波形示意图

17.3.7 功耗和速度模式

比较器1支持轨到轨输入，输入电压范围0~VDD，支持低功耗模式、中速模式和高速模式，用于平衡不同应用下的功耗和延迟。

芯片上电复位后，两个比较器默认处于低功耗模式，应用可以根据需要配置其工作模式。

模式	典型功耗	典型传输延迟	典型建立时间	使用条件
高速模式		50ns		所有芯片功耗模式下都可以使用
中速模式	<2uA	0.6us	5us	
低功耗模式	<1uA	2us	15us	

表 17-3 比较器 1 工作模式

比较器2为低功耗比较器，输入电压范围0~VDD-0.7V，建立时间小于10us，传输延迟小于5us，典型功耗200nA。

模式	典型功耗	典型传输延迟	典型建立时间	使用条件
低功耗模式	0.2uA	5us	10us	所有芯片功耗模式下都可以使用

表 17-4 比较器 2 工作模式

17.3.8 比较器中断

比较器输出可以在上升沿、下降沿上分别产生独立的中断事件。CMPxIE寄存器可以使能或禁止中断输出。CMPxIF标志寄存器在中断事件发生时置位，软件写1清零。软件也可以通过CMPxO寄存器直接读取比较器的输出值。

比较器输出的边沿检测采用异步电路实现，无需工作时钟，因此可以在芯片处于休眠模式的情况下产生中断，唤醒芯片。

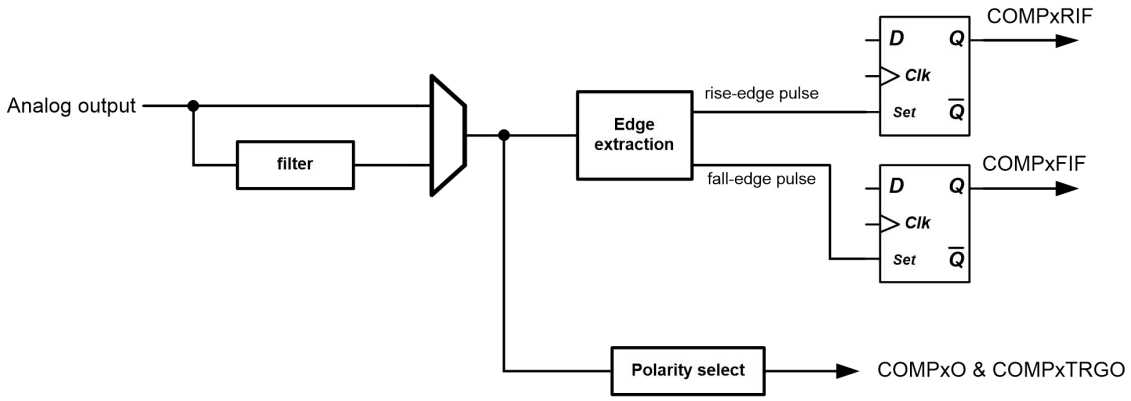


图 17-6 比较器中断产生

17.3.9 比较器输出和触发信号输出

比较器可以直接输出比较结果，还可以给其他外设电路输出触发信号。

输出逻辑示意图如下。

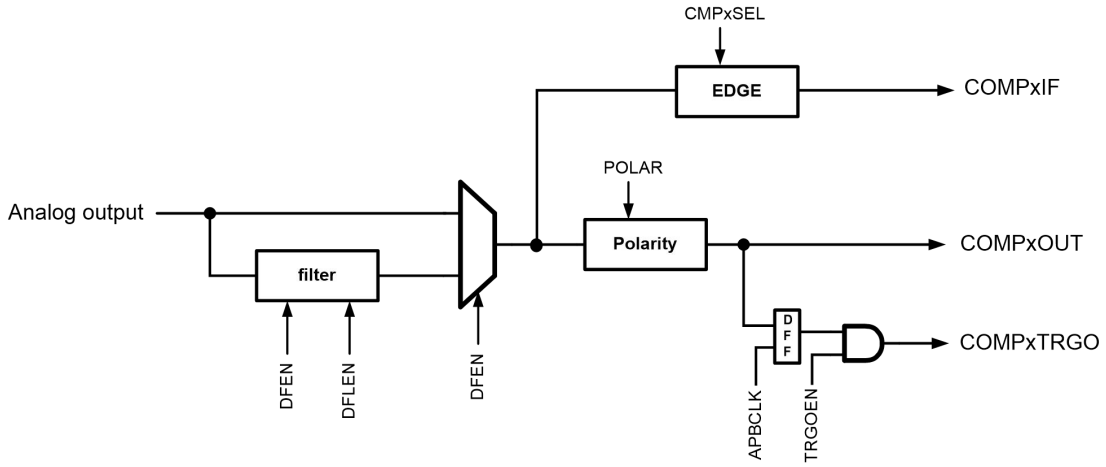


图 17-7 比较器输出逻辑

比较器输出

比较器比较结果（COMPxOUT）可以被软件实时查询，也能够输出到GPIO上。输出信号可以选择是否经过数字滤波。

比较器触发信号输出

可以在比较器输出的上升沿、下降沿分别或同时产生触发信号输出。需要输出触发信号时，必须使能COMP总线时钟，当触发事件发生时，触发信号被同步到APBCLK上升沿输出。触发信号可以被

连接到定时器的内部触发输入，或者ADC的内部触发输入。

TRGOEN用于使能或禁止触发信号输出。

下图为比较器输出上升沿产生触发信号的例子。

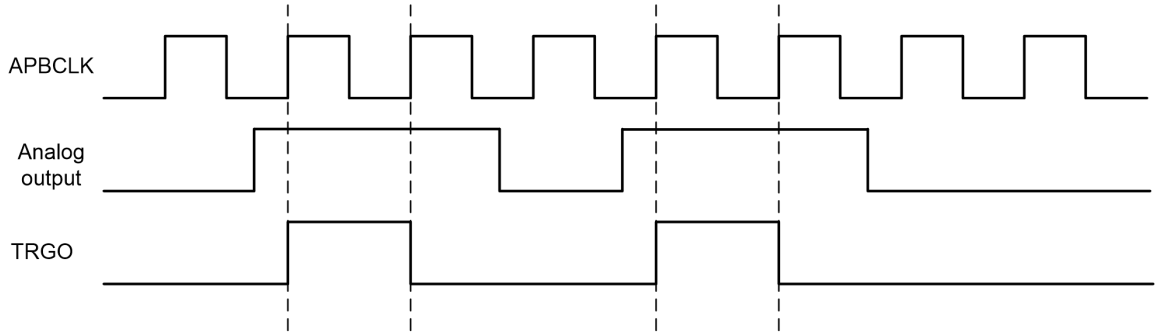


图 17-8 比较器输出上升沿产生触发输出，无滤波

下图为比较器输出上升沿和下降沿都产生触发信号的例子。

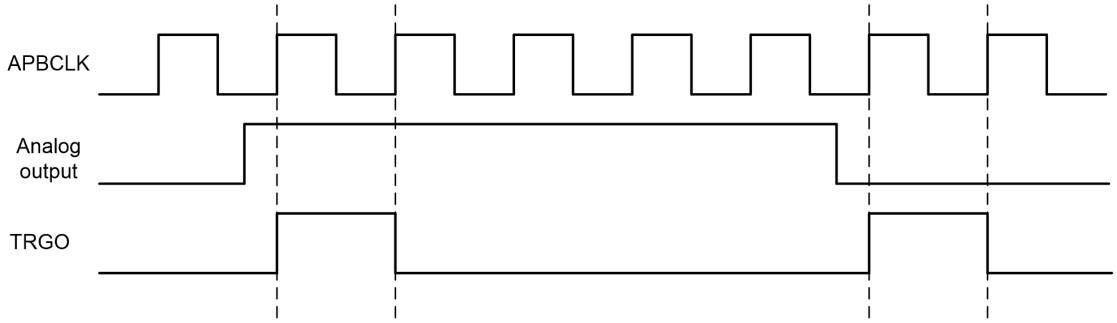


图 17-9 比较器输出上升沿下降沿产生触发输出，无滤波

比较器的触发输出可以作为启动ADC转换的触发信号，请参考31.6.8转换触发。

17.4 寄存器

模块基地址：0x4001_9C00

offset 地址	名称	符号
0x00	COMP 控制寄存器 1	COMP_CR1
0x04	COMP 控制寄存器 2	COMP_CR2
0x08	-	-
0x0C	COMP 中断配置寄存器	COMP_ICR
0x10	COMP 中断标志寄存器	COMP_ISR
0x14	COMP 缓冲器控制寄存器	COMP_BUFCR

17.4.1 COMP 控制寄存器 1（COMP_CR1）

名称	COMP_CR1							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					WINMODE	POLAR	DFEN
位权限	R/W-00000					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-					CMP1O
位权限	R/W-00		U-0					R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			V1PSEL	V1NSEL			CMP1EN
位权限	U-0			R/W-0	R/W-00			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 1 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1（最小滤波长度为 3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样

Bit	助记符	功能描述
18	WINMODE	比较器 1 窗口模式控制寄存器 0：禁止窗口模式 1：使能窗口模式
17	POLAR	比较器 1 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 1 输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	MODE	比较器 1 工作模式 00/11：低功耗模式 01：中速模式 10：高速模式
13:9	--	RFU：未实现，读为 0
8	CMP1O	比较器 1 输出，软件只读
7:5	--	RFU：未实现，读为 0
4	V1PSEL	比较器 1 正极输入选择 0：COMP1_INP1 1：RFU
3:1	V1NSEL	比较器 1 负极输入选择 000：COMP1_INN1 001：RFU 010：VREFINT 011：AVREF 100：1/2 AVREF 101：RFU 110：RFU 111：RFU
0	CMP1EN	比较器 1 使能位 0：关闭比较器 1 1：使能比较器 1

17.4.2 COMP2 控制寄存器（COMP_CR2）

名称	COMP_CR2							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					WINMODE	POLAR	DFEN
位权限	R/W-00000					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							CMP2O
位权限	U-0							R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V2PSEL			V2NSEL			CMP2EN

位权限	U-0	R/W-000	R/W-000	R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 2 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样 00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	WINMODE	比较器 2 窗口模式控制寄存器 0：禁止窗口模式 1：使能窗口模式
17	POLAR	比较器 2 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 2 输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	--	RFU：未实现，读为 0
13:9	--	RFU：未实现，读为 0
8	CMP2O	比较器 2 输出，软件只读
7	--	RFU：未实现，读为 0
6:4	V2PSEL	比较器 2 正极输入选择 000：COMP2_INP1 001：RFU 010：AVREF 011：ULPBG_VREF(1.16V) 100：VDD15 101：VREFP
3:1	V2NSEL	比较器 2 负极输入选择 000：COMP2_INN1 001：RFU 010：VREFP 011：AVREF 100：1/2 AVREF 101：RFU 110：RFU 111：RFU
0	CMP2EN	比较器 2 使能位

Bit	助记符	功能描述
		0：关闭比较器 2 1：使能比较器 2

17.4.3 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						OOW_IE	WIN_IE
位权限	U-0						R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				CMP2SEL		-	CMP2IE
位权限	U-0				R/W-00		U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CMP1SEL		-	CMP1IE
位权限	U-0				R/W-00		U-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25	OOW_IE	Out-Of-Window 中断使能
24	WIN_IE	Window 中断使能
23:12	--	RFU：未实现，读为 0
11:10	CMP2SEL	比较器 2 中断源选择 00/11：比较器 2 输出上升或下降沿产生中断 01：比较器 2 输出上升沿产生中断 10：比较器 2 输出下降沿产生中断
9	--	RFU：未实现，读为 0
8	CMP2IE	比较器 2 中断使能，1 有效
7:4	--	RFU：未实现，读为 0
3:2	CMP1SEL	比较器 1 中断源选择 00/11：比较器 1 输出上升或下降沿产生中断 01：比较器 1 输出上升沿产生中断 10：比较器 1 输出下降沿产生中断
1	--	RFU：未实现，读为 0
0	CMP1IE	比较器 1 中断使能，1 有效

*注：为了避免误触发中断，应在关闭中断使能的情况下设置中断源选择寄存器。

17.4.4 COMP 中断标志寄存器（COMP_ISR）

名称	COMP_ISR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			OOW_IF	WIN_IF	-	CMP2IF	CMP1IF
位权限	U-0			R/W-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4	OOW_IF	Out-of-Window 中断标志，硬件置位，软件写 1 清零
3	WIN_IF	Window 中断标志，硬件置位，软件写 1 清零
2	--	RFU：未实现，读为 0
1	CMP2IF	比较器 2 中断标志，硬件置位，软件写 1 清零
0	CMP1IF	比较器 1 中断标志，硬件置位，软件写 1 清零

17.4.5 COMP 缓冲器控制寄存器（COMP_BUFCR）

名称	COMP_BUFCR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					BYP	-	ENB
位权限	U-0					R/W-0	U-0	R/W-1

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	BYP	缓冲器 bypass 使能 0：不 bypass 1：bypass
1	--	RFU：未实现，读为 0
0	ENB	缓冲器使能 0：使能基准缓冲器 1：关闭基准缓冲器

18 硬件除法器（DIVAS）

18.1 概述

DIVAS用于帮助软件加速除法运算。

DIVAS包含的硬件除法器是一个有符号数整数除法器，可以输入32bit被除数和16bit除数，输出32bit商和32bit余数。

乘法功能在Cortex-M0内核中完成。

特点：

- 有符号整数运算（二进制补码格式）
- 32bit被除数、16bit除数
- 输出32bit商和32bit余数
- 除以0警告
- 一次计算需要8个周期

18.2 工作流程

软件按照如下步骤调用硬件除法器。

- 向DIVEND寄存器写入32bit被除数（二进制补码）
- 向DIVSOR寄存器写入16bit除数（二进制补码）
- 硬件除法器在软件写入DIVSOR后自动开始运算，同时置位BUSY寄存器
- 软件查询BUSY标志，直到运算完成后BUSY自动清零
- 查询DIV_BY_0标志
- 读取QUOT寄存器中的商
- 读取REMD寄存器中的余数

18.3 寄存器

模块起始地址：0x4001 3000

offset 地址	名称	符号
0x00	被除数寄存器	DIVAS_END

offset 地址	名称	符号
0x04	除数寄存器	DIVAS_SOR
0x08	商寄存器	DIVAS_QUOT
0x0C	余数寄存器	DIVAS_REMD
0x10	状态标志寄存器	DIVAS_SR

18.3.1 被除数寄存器

名称	DIVAS_END							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIVEND[31:24]							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DIVEND[23:16]							
位权限	R/W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVEND[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVEND[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:0	DIVEND	32bit 有符号被除数

18.3.2 除数寄存器

名称	DIVAS_SOR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVSOR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVSOR[7:0]							
位权限	R/W-00000001							

位号	位名	说明
31:0	--	RFU：未实现，读为 0
15:0	DIVSOR	16bit 有符号除数

18.3.3 商寄存器

名称	DIVAS_QUOT							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QUOT[31:24]							
位权限	R							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QUOT [23:16]							
位权限	R							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QUOT [15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QUOT [7:0]							
位权限	R							

位号	位名	说明
31:0	QUOT	32bit 有符号商 （仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

18.3.4 余数寄存器

名称	DIVAS_REMD							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REMD[15:8]							
位权限	R							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REMD[7:0]							
位权限	R							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	REMD	16bit 有符号余数 （仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

18.3.5 状态标志寄存器

名称	DIVAS_SR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DIV_BY_0	BUSY
位权限	U-0						R-0	R-0

位号	位名	说明
31:2	--	RFU：未实现，读为 0
1	DIV_BY_0	**除数为 0 标志** 1 = 除数为 0 0 = 除数不为 0
0	BUSY	**运算过程指示** 1 = HDIV 在计算过程中，结果未就绪 0 = 计算完毕，结果就绪 软件在写入除数后，DIVAS 开始计算，软件应查询 BUSY 为低后再读取商和余数寄存器

19 双线串行总线和系统管理总线（I²C-SMBus）

19.1 概述

I²C-SMB 模块实现 MCU 与外部 I²C 或 SMBus 器件之间的同步通信。

本模块支持 I²C 的单主机、多主机和从机模式，兼容标准模式（standard-mode 100Kbps）、快速模式（fast-mode 400Kbps）、增强快速模式（fast-mode+ 1Mbps）。

本模块也可以支持 SMBus rev3.0（System Management Bus）和 PMBus（Power Management Bus）协议规范。

本芯片最大支持 2 路独立 I2C-SMBus。

19.2 主要功能

- I2C 协议兼容
  - 主机和从机模式
  - 多主机通信
  - 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)和 Fm+(1Mbps)
  - 7 位和 10 位寻址
  - General Call
  - 支持可编程的 setup/hold 时序
  - 支持时钟延展
- SMBus rev 3.0 兼容
  - 硬件 PEC 和 ACK
  - 支持 ARP（Address Resolution Protocol）
  - 支持 Host 和 Device 功能
  - SMBus alert 信号
  - 超时检测和 IDLE 检测
- PMBus rev 1.3 兼容
- 独立时钟工作，不依赖于 APB 总线时钟
- 支持 DMA 数据搬运
- 低功耗特性
  - 低功耗从机设计，不需要片上时钟即可完成数据收发
  - 从机唤醒功能：支持从机地址匹配唤醒、数据帧接收完成唤醒或 START 检测唤醒

19.3 模块框图

19.3.1 I²C-SMB 模块框图

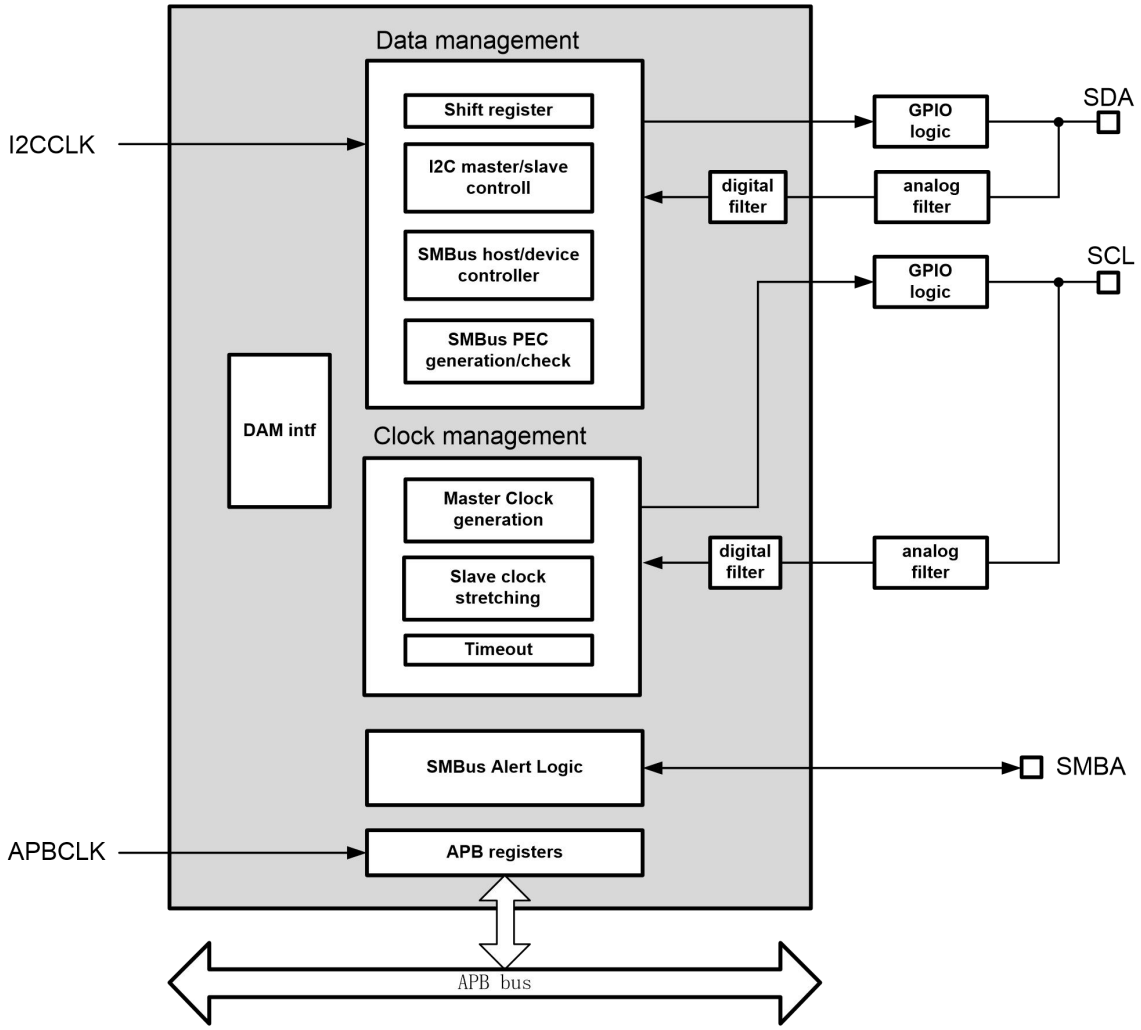


图 19-1 I²C-SMB 模块框图

19.4 引脚定义

I²C_SMB 模块最多使用 3 个引脚与外部器件通信：

引脚	I²C-SMBx	符号	功能
PB0,PC8	I2C_SMB0	SCLA0	串行时钟
PB1,PC9		SDAA0	串行数据
PB11,PC10		SMB0_ALERTB	SMBus Alertb
PD9,PB2	I2C_SMB1	SCLA1	串行时钟
PD8,PB3		SDAA1	串行数据
PD13,PB4		SMB1_ALERTB	SMBus Alertb

表 19-1 I²C 引脚列表

19.5 时钟和复位

I²C-SMBus 主机和从机都采用了双时钟结构：

- 主机和从机的总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 I²C 内部寄存器时，必须使能 PCLK。
- 主机的数据收发时钟用 I2CCLK 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK，能够独立于 APBCLK 工作。必须使能 I2CCLK 才能进行数据收发。
- 从机通过采样 SCL 输入的有效沿来进行数据收发

PCLK 和 I2CCLK 的控制都在 CMU 模块内完成，进行 I²C 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 I²C 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，I²C 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 I²C 以较高的波特率进行数据通信。

理论上 PCLK 和波特率时钟之间没有相对关系的约束，波特率时钟可以快于或者慢于 PCLK。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

注意：模块工作前需先清除 RMU 模块中的 I2CSMB 复位寄存器（I2CSMBRST）。

19.6 接口时序

19.6.1 通信过程

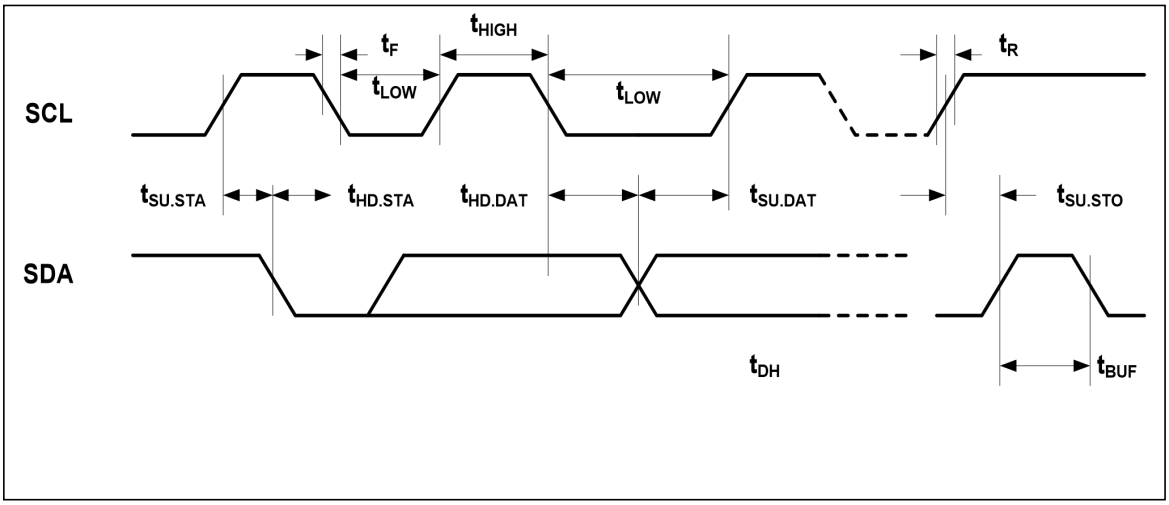


图 19-2 I²C 总线时序

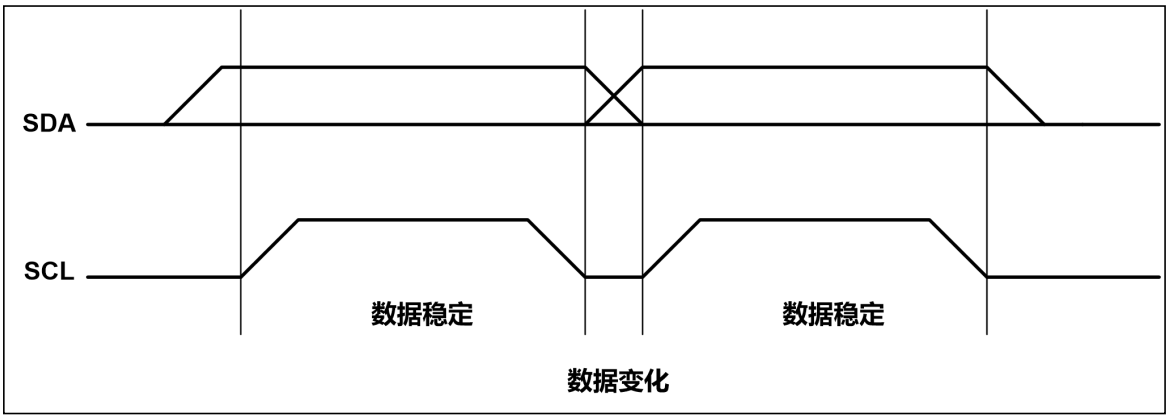


图 19-3 数据有效时序

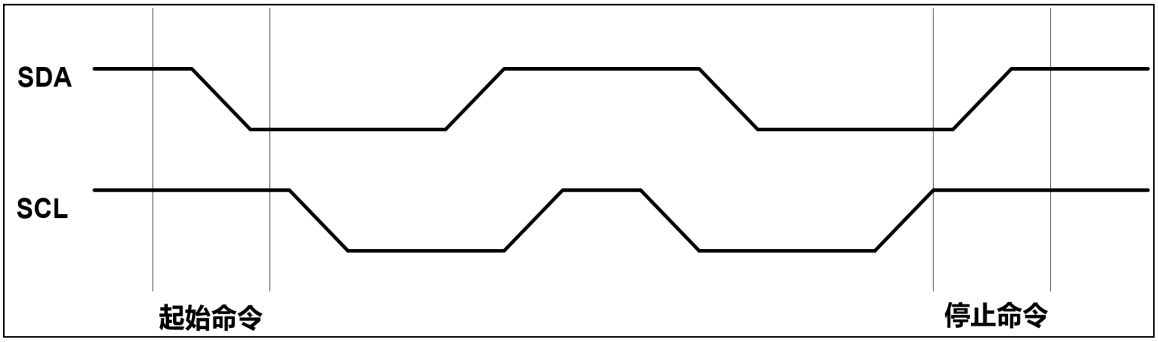


图 19-4 起始（Start）与停止(Stop)命令定义

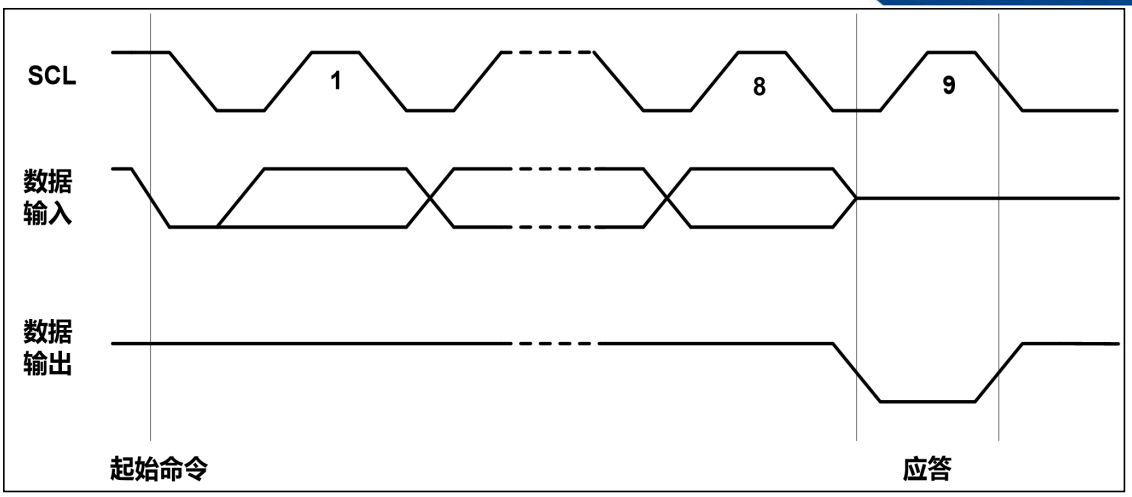


图 19-5 输出应答(ACK)

19.6.2 接口时序描述

时钟有效时序：SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化（参见图 19-3）；当数据在 SCL 为高时变化，将视为下文所述的一个起始或停止命令。

起始命令：当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始（参见图 19-4）。

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使从机进入等待态低功耗模式（参见图 19-4）。

输出应答：SDA 上的数据都是以 8 位为一组串行输入和输出的，MSB 先发，接收方在收完每个字节后应当在第 9 个周期回发一个回应 acknowledge 位（以下简称 ACK），ACK 的时钟由主机提供。发送方在 ACK 期间悬空 SDA，接收方须将 SDA 拉低，确保 ACK 时钟高电平期间 SDA 为低，形成有效的 ACK 信号（参见图 19-5）。

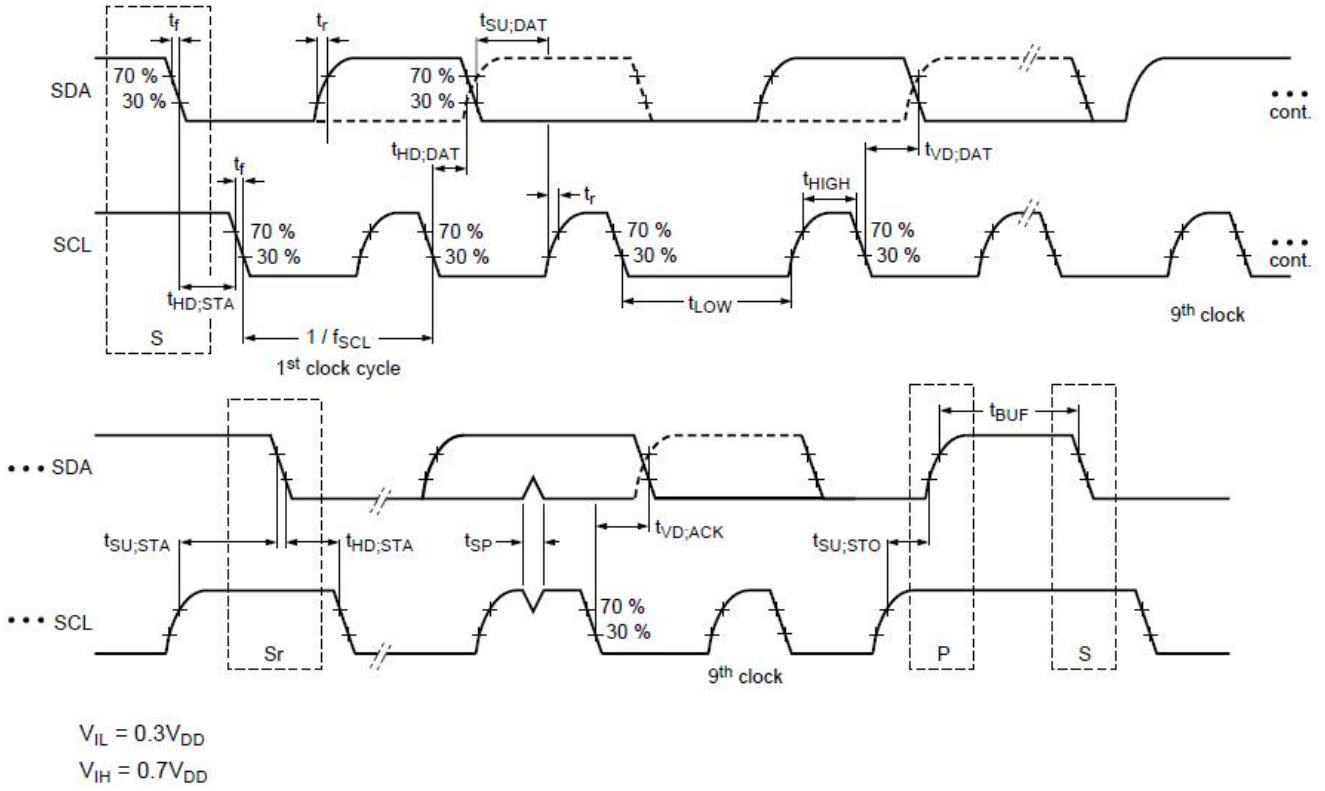
19.6.3 总线时序参数表

I²C 协议规定的总线时序参数表如下。

参数	符号	Standard mode(100K)		Fast mode (400K)		Fast mode plus (1M)		单位
		min	max	min	max	min	max	
SCL 时钟频率	F_{SCL}	0	100	0	400	0	1000	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	-	0.6	-	0.26	-	us
启动条件保持时间	$T_{HD:STA}$	4.0	-	0.6	-	0.26	-	us
时钟低电平时间	T_{LOW}	4.7	-	1.3	-	0.5	-	us
时钟高电平时间	T_{HIGH}	4.0	-	0.6	-	0.26		us
数据输入建立时间	$T_{SU:DAT}$	250	-	100	-	50	-	ns
数据输入保持时间	$T_{HD:DAT}$	0	-	0	-	0	-	us
SDA 和 SCL 上升时间	T_R	-	1000	20	300		120	ns
SDA 和 SCL 下降时间	T_F	-	300	20x(VDD /5.5)	300	20x(VDD /5.5)	120	ns
停止条件建立时间	$T_{SU:STO}$	4.0	-	0.6	-	0.26	-	us
STOP 和 START 之间的总线空闲时间	T_{BUF}	4.7	-	1.3	-	0.5	-	us
总线的容性负载	Cb	-	400	-	400	-	550	pF
数据有效时间	$t_{VD;DAT}$	-	3.45	-	0.9	-	0.45	us
ACK 有效时间	$t_{VD;ACK}$	-	3.45	-	0.9	-	0.45	us
噪声容限低值	V_{nL}	$0.1V_{DD}$	-	$0.1V_{DD}$	-	$0.1V_{DD}$	-	V
噪声容限高值	V_{nH}	$0.2V_{DD}$	-	$0.2V_{DD}$	-	$0.2V_{DD}$	-	V

表 19-2 I²C 接口时序要求

以上各项参数可以在下图中找到对应的图示。



$V_{IL} = 0.3V_{DD}$

$V_{IH} = 0.7V_{DD}$

图 19- 6I²C 时序参数图例

19.7 I²C 功能描述

I²CSMB 模块包含主机、从机、IO 接口控制和全局配置寄存器。其中主机部分可以兼容 I²C master 和 SMBu master，从机部分可以兼容 I²C slave 和 SMBus slave；同时此模块还可以支持 SMBus Host 功能。

模块支持以下工作模式。

- I²C 主机发送
- I²C 主机接收
- I²C 从机接收
- I²C 从机发送

芯片上电后 I²CSMB 模块默认关闭，主机和从机都不工作。软件通过置位 EN 寄存器使能 I²CSMB 模块，模块使能后默认处于从机模式，并准备好接收地址和数据。当软件对 START 寄存器写 1 后，模块在总线上发起 START 时序，此时模块自动进入主机模式。当总线仲裁失败，或者发送完 STOP 时序后，模块自动从主机模式回到从机模式，以便于兼容多主机总线应用。

19.7.1 预设从机地址表

I²C和SMBus协议规定了一部分预设地址，模块能够自动识别其中一部分预设地址，其他预设地址由软件进行协议层判断和处理。

对于10bit从机地址应用，即A10EN=1的情况下，要求第一个字节必须以11110开头，否则将触发ADDR_ERROR错误标志。而在A10EN=0的情况下，如果从机收到了11110开头的地址字节，也会置位ADDR_ERROR错误标志。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
0001 000	X	SMBus host
0001 001	X	Smart Battery Charger
0001 010	X	Smart Battery Selector Smart Battery System Manager
0001 011	X	Smart Battery
0001 100	X	SMBus Alert Response Address
0101 000	1	PMBus ZONE READ

0101 100	X	Reserved by SMBus for LCD controller
0101 101	X	Reserved by SMBus for CCFL Backlight driver
0110 111	0	PMBus ZONE WRITE
1000 0XX	X	Reserved by SMBus for PCMCIA Socket Controller
1000 100	X	Reserved by SMBus for VGA controller
1001 0XX	X	Prototype Address
1100 001	X	SMBus Device Default Address
1111 0XX	X	10bit slave addressing
1111 1XX	X	Reserved for future purpose

表 19- 3 I²C-SMBus 从机保留地址定义

在使能了SMBus功能后，模块能够自动识别SMBus host, SMBus Alert Response Address, SMBus Device Default Address，并给出相应的标志和中断事件。

19.7.2 I²C 初始化过程

进行I²C通信前必须正确的初始化I²CSMB模块，建议软件按照以下步骤进行初始化操作：

- 清零RMU模块的I2CSMBxRST寄存器，确保I²C模块不处于复位状态
- 置位CMU模块的I2CSMBx_PCE寄存器，使能I²CSMB模块寄存器总线接口时钟
- 配置CMU模块的I2CSMBxCKS和I2CSMBxCKE寄存器，选择并使能I²CSMBx工作时钟（如果是从机模式，不需要这个步骤）
- 根据需要使能或禁止模拟滤波功能（SCL和SDA输入模拟滤波，>50ns）
- 根据需要使能或禁止数字滤波功能

IO配置

需要将用于I²C通信的GPIO配置为数字外设功能。用于I²C通信的IO必须配置为开漏输出模式，支持Fm+模式的IO必须具有20mA的sink能力，支持HS模式的IO则必须具备电流源驱动能力。

请查询芯片数据手册以获得更多IO相关信息。

主机波特率配置

I²C主机需要在使能前配置通信波特率，而从机通信速率由主机发送的SCL决定，因此不需要配置。

BRGH和BRGL波特率配置寄存器用于产生通信波特率。BRGH和BRGL是9 bit波特率分频系数，波特率计算公式如下：

SCL周期T_{SCL}＝T_{BRGH} + T_{BRGL}

其中BRGH定义SCL高电平宽度，BRGL定义SCL低电平宽度

T_{BRGH}＝T_{I2CCLK}×(BRGH+1)

T_{BRGL}＝T_{I2CCLK}×(BRGL+1)

T_{I2CCLK}为I²C工作时钟周期

例如对于100k波特率，T_{SCL}=10us；若I²C工作时钟为8M，则T_{I2CCLK}=125ns。假设要求SCL占空比为50%，即SCL高低电平都是5us宽度，根据以上公式可以计算得到BRGH=BRGL=39

噪声滤波

SDA和SCL输入都支持噪声滤波功能。噪声滤波包含模拟滤波和数字滤波两部分。

模拟滤波的滤波长度不小于50ns，符合I²C协议要求的fast-mode和fast-mode plus模式下对至少50ns噪声脉冲的抑制能力。软件可以通过ANFEN寄存器使能或禁止模拟滤波功能，不论是主机还是从机模式，都可以使能模拟滤波。

数字滤波功能由DNF寄存器单独使能或禁止，主机和从机都可以使能数字滤波。使能了数字滤波后，SDA和SCL总线输入到模块内部信号，在连续采样规定次数一致后，才会改变，此功能可以抑制一定宽度的毛刺信号，滤波长度由DNF×T_{I2CCLK}决定。

注意，使能数字滤波的情况下，从机必须使能I2CSMBCLK，因此无法在休眠模式下使用。如果从机希望不使用I2CSMBCLK工作，则必须关闭数字滤波。

19.7.3 I²C 主机功能

I²C总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。本模块的I²C主机功能支持以下特性：

- 支持多主机总线
- 支持时钟同步
- 支持时钟仲裁
- 7位或10位从机寻址
- 支持从机时钟延展

- 可编程总线时序
- DMA支持

19.7.3.1 7bit 寻址和数据通信流程

在7bit寻址时，主机发送的第一个字节包含从机地址和传输方向位（$R/\overline{W}$），根据$R/\overline{W}$决定后续传输是主机向从机写入数据（$R/\overline{W}$=0）或主机从从机读取数据（$R/\overline{W}$=1）。

名字	Slave Address Byte							
位	7	6	5	4	3	2	1	0
位名	address							R/W

位描述：

位	位名	功能
7-1	address	Slave device address
0	R/W	0: Write 表示发送数据（master 发送） 1: Read 表示请求数据（slave 回发）

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

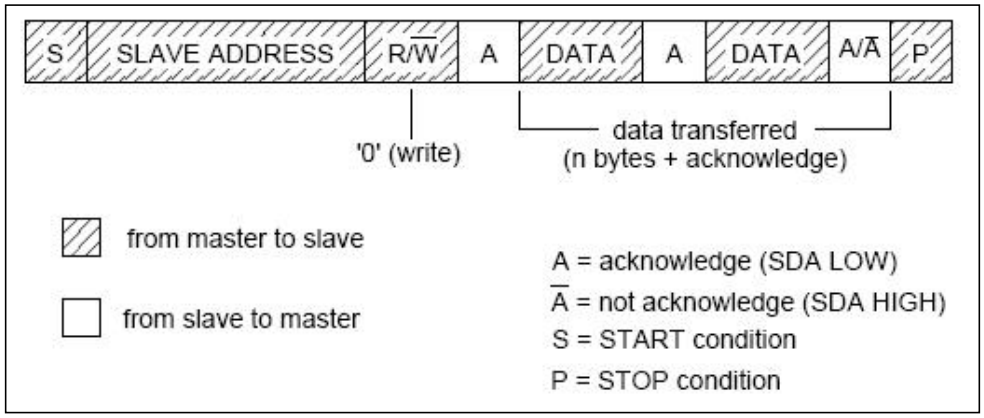


图 19-7 主机向 7 位地址从机写入数据时的帧格式

1、主机清零 RWN 寄存器，并发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到

ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

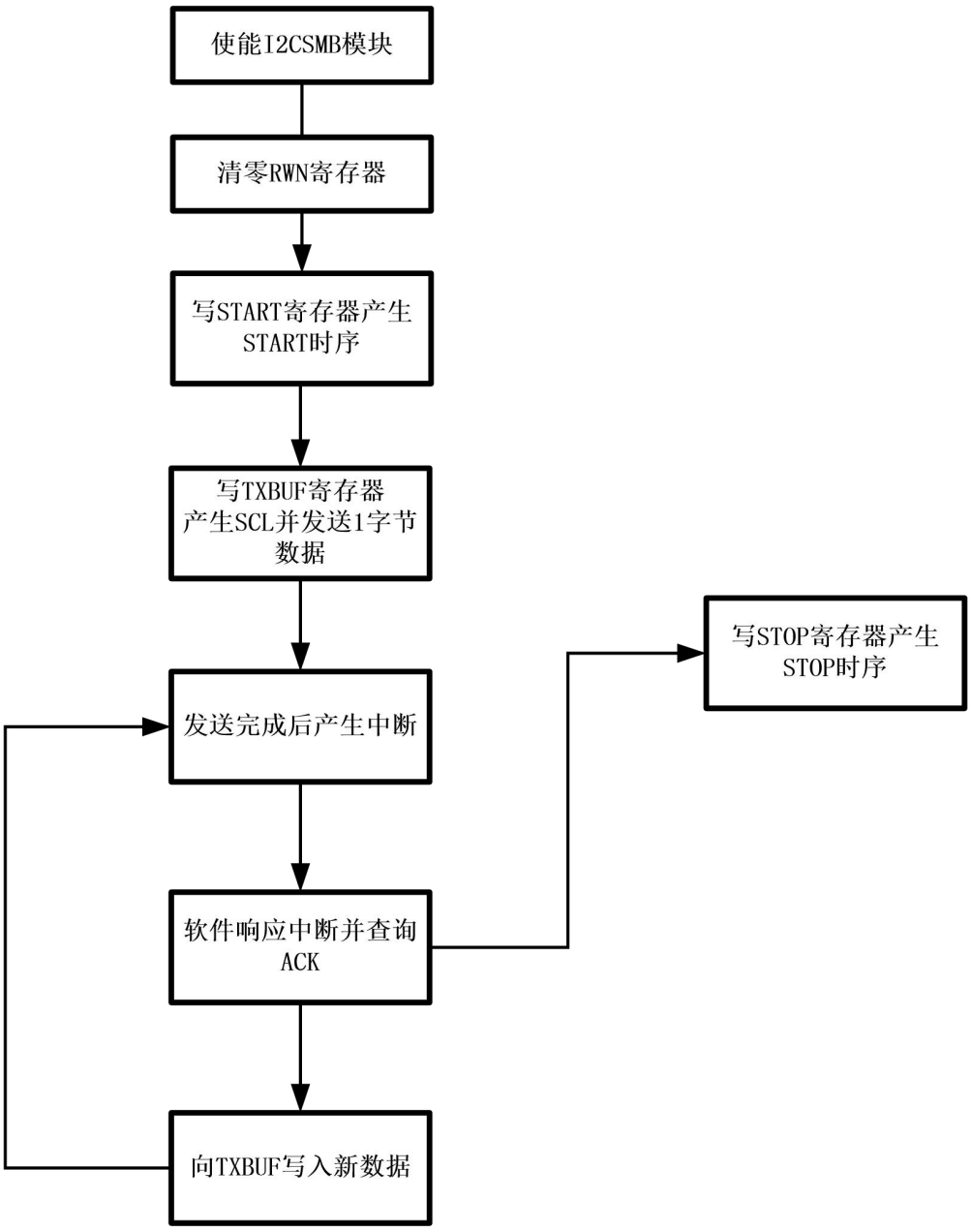


图 19-8I²C 主机软件发送数据流图

I²C主机对7位地址从机写入数据的波形示意图如下：

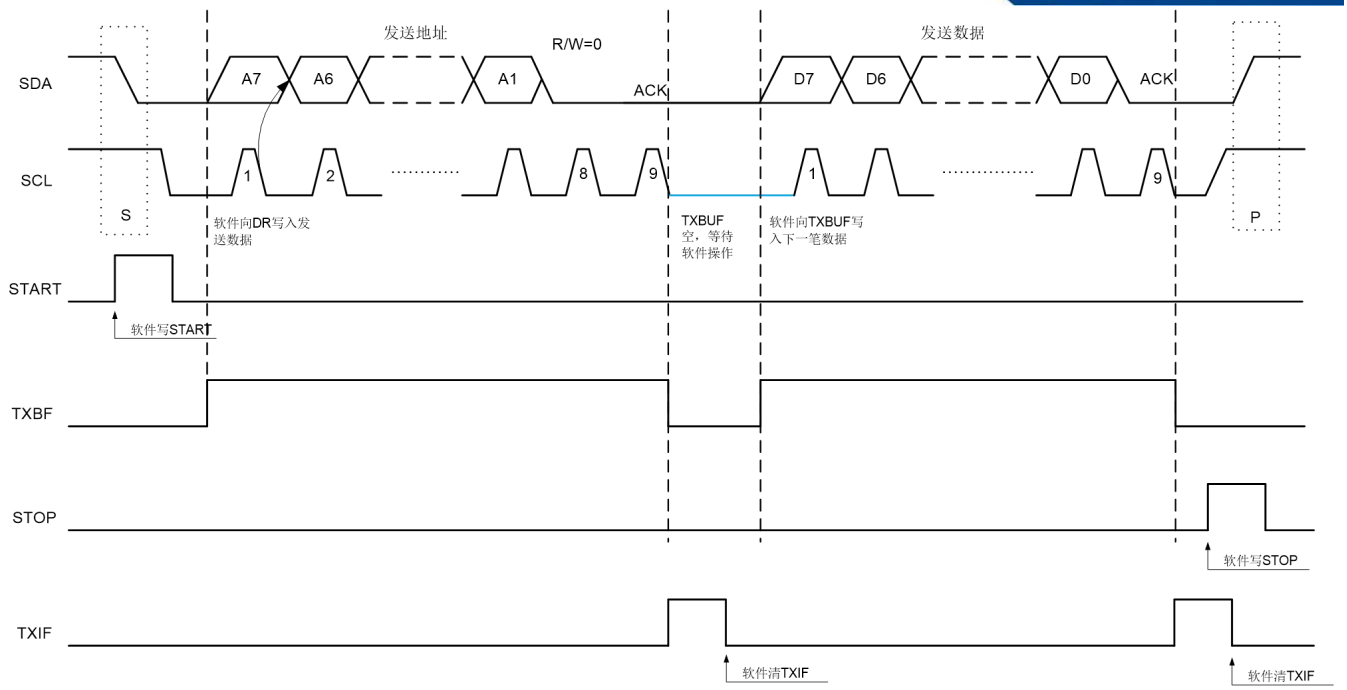


图 19-9 I²C 主机对 7 位地址从机发送数据流图

当数据寄存器TXBUF为空时，硬件会保持SCL为低电平，等待软件将下一笔数据写入TXBUF。如上图SCL蓝色部分所示。

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

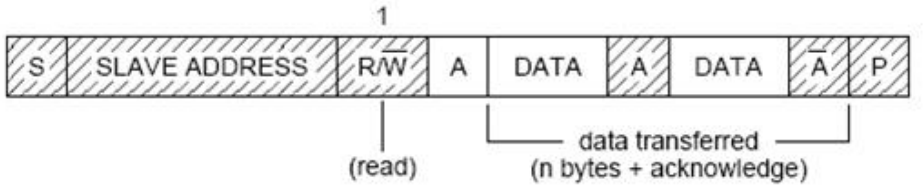


图 19-10 主机从 7 位地址从机读取数据时的帧格式

1、 主机置位 RWN 寄存器并发起 START 时序

2、 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 此时重新置位 RWN，主机转为接收状态

4、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

5、 每次接收完 1 字节数据后软件置位 RWN 开始读取下一个字节

6、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

7、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

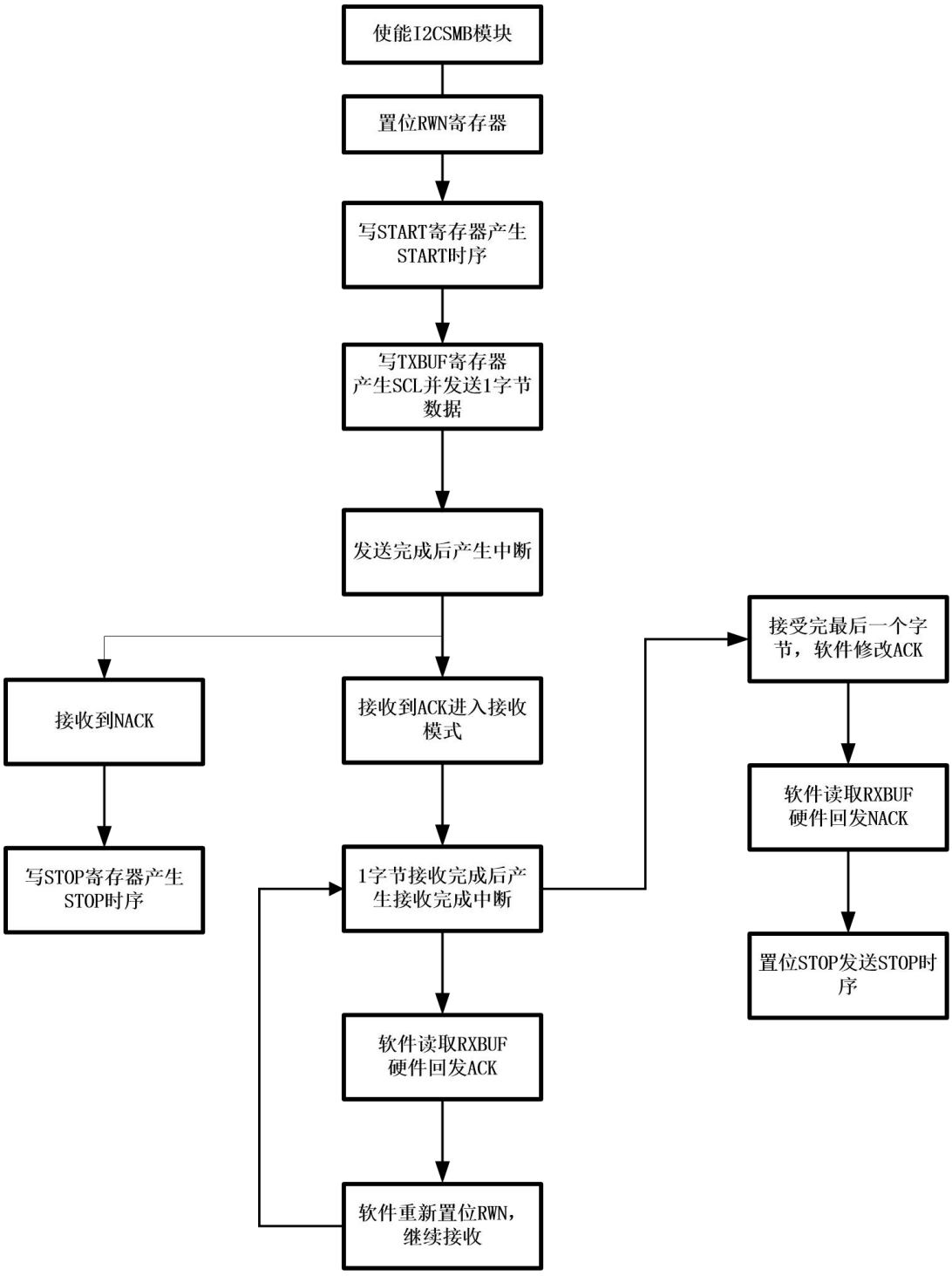


图 19-11 I²C 软件接收数据流图

主机每次接收完从机发送的数据后，根据ACKEN寄存器回发响应。ACKEN复位值为0，即默认状态

下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKEN寄存器改写为1。ACKEN为1的情况下，主机在发送完响应后会自动清零ACKEN。

I²C主机从7位地址从机读出数据的波形示意图如下：

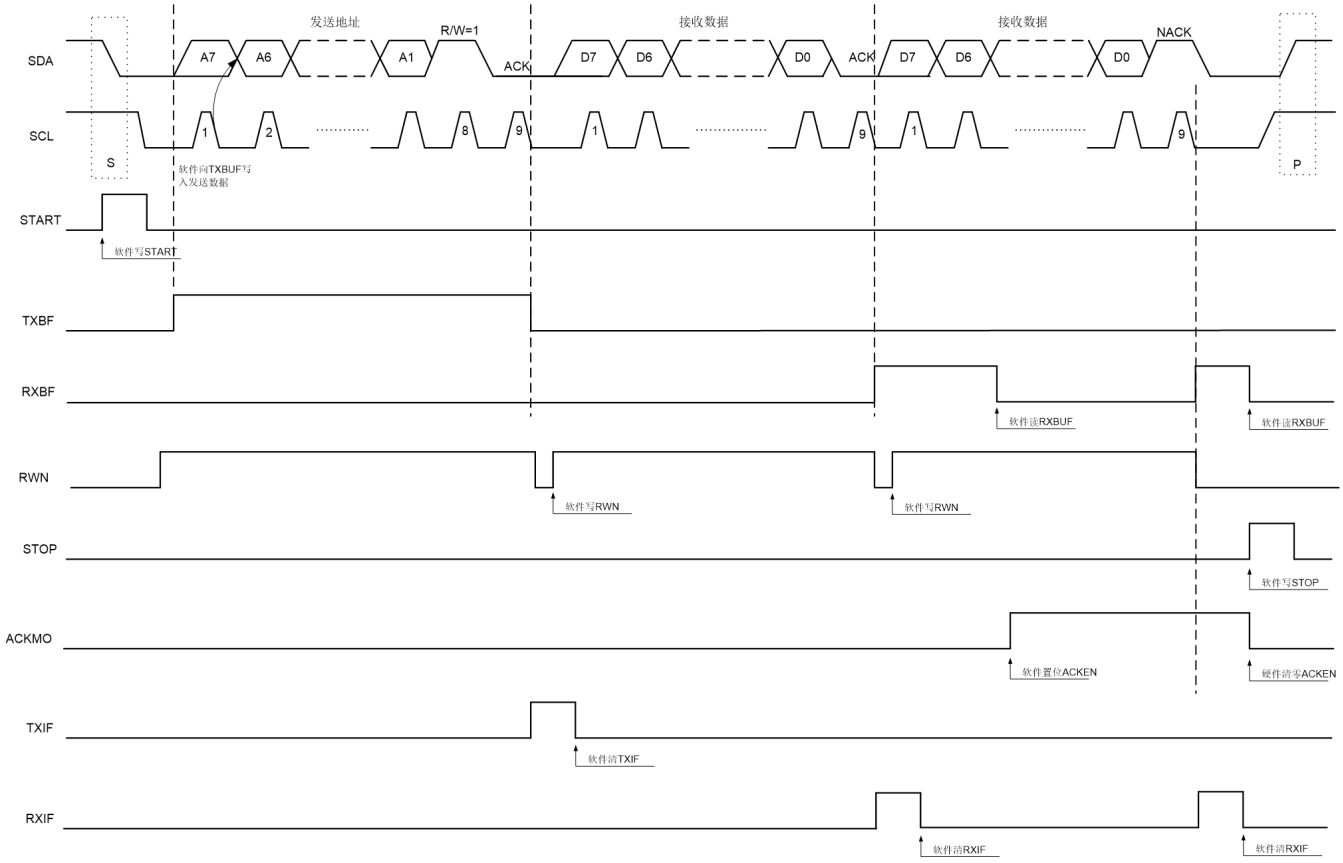


图 19-12 I²C 从 7 位地址从机读取数据流图

在主机连续接收过程中，软件通过置位RWN寄存器读取从机数据。当软件没有写RWN时，主机会保持等待，避免数据接收溢出。

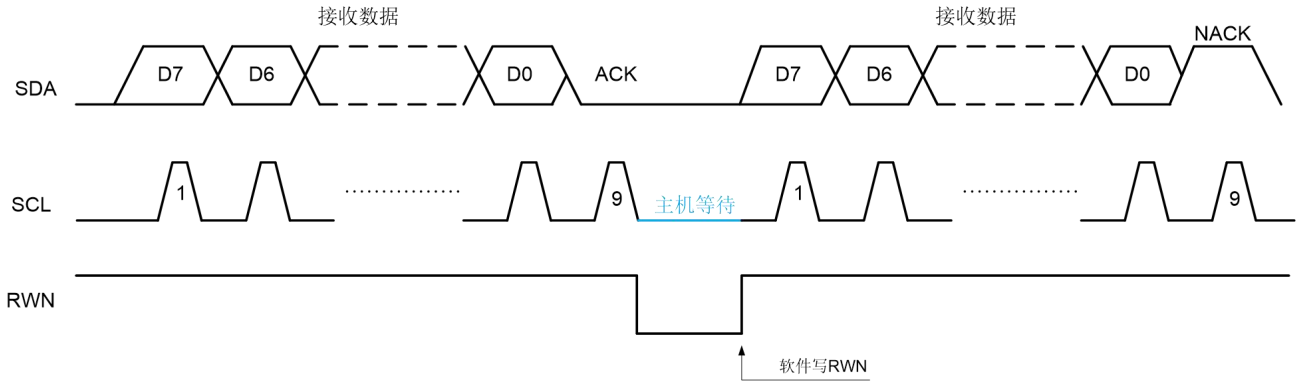


图 19-13 I²C 主机接收数据等待

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

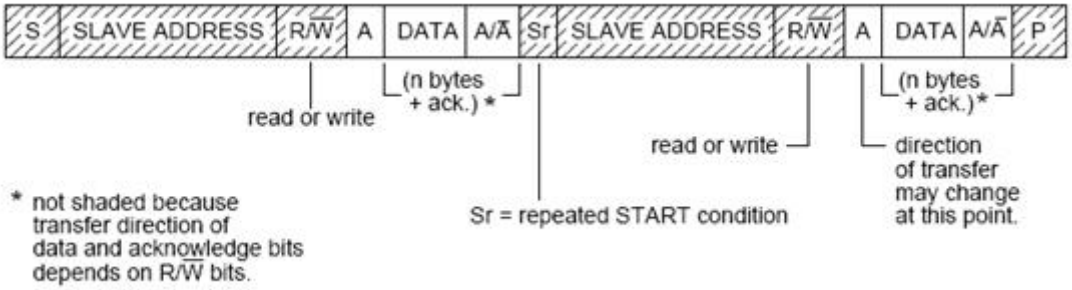


图 19-14 双向数据通信帧格式

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送RESTART时序和从机地址字节来修改传输方向。

19.7.3.2 10bit 寻址和数据通信流程

在10bit寻址时，主机发送的第一个字节包含部分从机地址（11110_A9_A8）和传输方向位（$R/\overline{W}$），第二个字节包含剩余从机地址（A7~A0）。两个字节地址发送完成后，再进行数据传输。

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

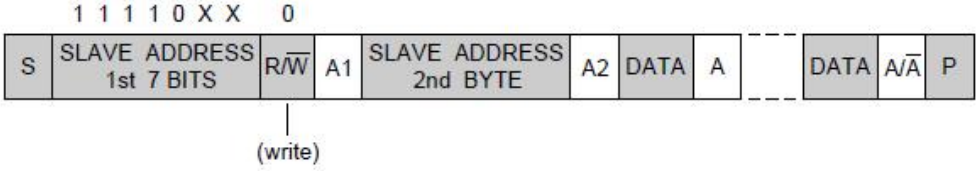


图 19-15 10bit 寻址，主机向从机写入数据

1、主机清零 RWN 寄存器并发起 START 时序

2、主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

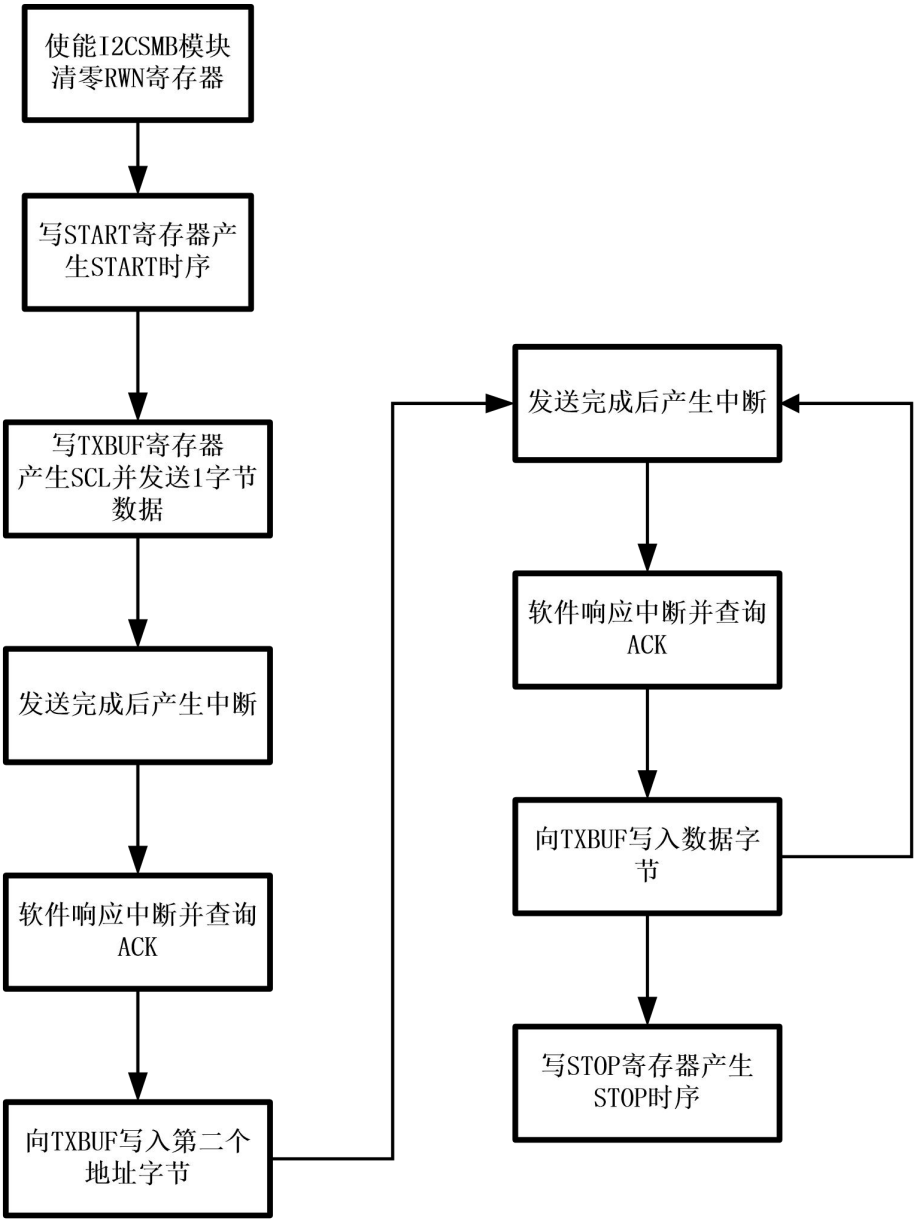


图 19-16I²C 软件发送数据流图

主机从从机读取数据

典型的10bit寻址，主机从从机读取数据的数据流图如下图所示。

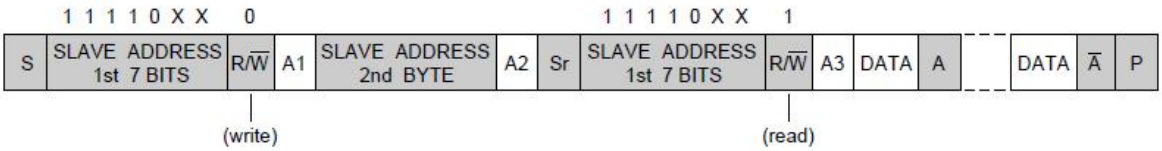


图 19-17 10bit 寻址，主机从从机读取数据

1、 主机清零 RWN 寄存器并发起 START 时序

2、 主机发送第一字节从机地址，包含 5 位前导码 11110、2 位从机地址最高位和 1 位 R/W 标志位

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 RESTART 时序

5、 置位 RWN 寄存器

6、 主机再次发送第一字节从机地址，将 R/W 为改为 1

7、 主机自动转为接收状态

8、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

9、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

10、 主机发送 STOP 时序终止读取

软件启动 I^2C 接收的操作流程如下图：

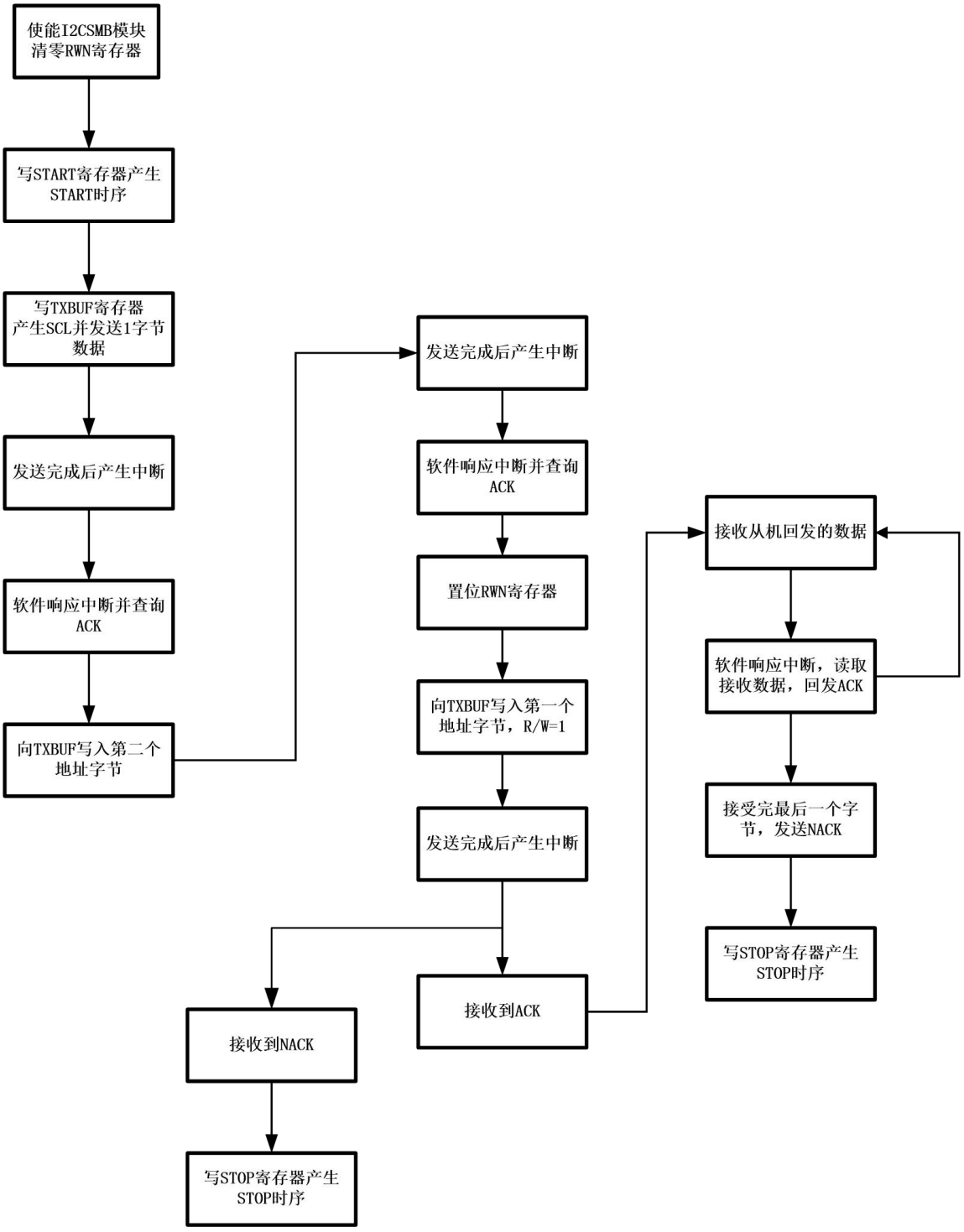


图 19-18 I²C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

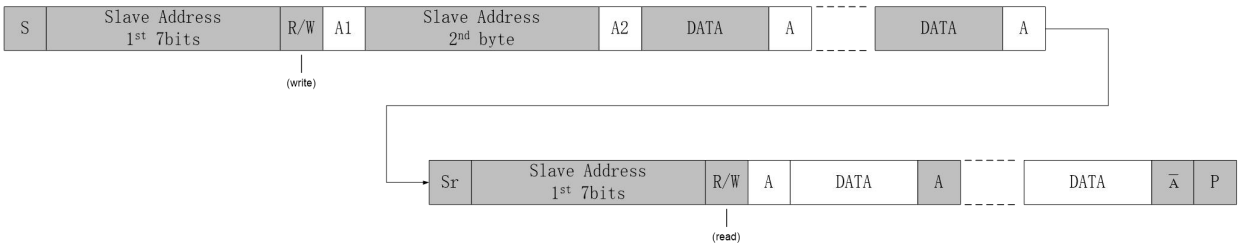


图 19-19 I²C 软件发送数据流图

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送RESTART时序和1st从机地址字节来修改传输方向。

19.7.3.3 DMA

I²C 主机支持 DMA，需要注意的是，必须在 I²CSMB 模块的总线时钟（I2CSMBx_PCE）使能的情况下，才能使用 DMA 功能。

主机使用DMA向从机写入数据

主机使用 DMA 发送数据时，包括从机地址字节和发送数据在内的所有数据都需要事先写入 RAM 中，并通过 DMA 请求发送出去。软件应事先将目标 DMA 通道配置为 I2CSMB_TX。

在 TXDMAEN=1 的情况下，START 置位，如果数据缓存寄存器 TXBUF 为空，I²CSMB 模块将产生 DMA 请求，DMA 模块响应请求后将 RAM 中的待发数据写入 TXBUF，开始数据发送（第一个字节是从机地址）。DMA 发送模式下，I²CSMB 并不检查发送数据的合法性，软件必须保证 RAM 中的数据是正确的。

每个字节发送完成后，I²CSMB 检查从机 ACK，如果 ACK 正确则产生新的 DMA 请求，如果收到 NACK 则产生 NACK 中断，并不再产生 DMA 请求。

当 DMA 完成指定长度的数据发送后，产生 DMA 传输完成中断，此时可以由软件置位 STOP 寄存器产生 STOP 时序，也可以由 I²CSMB 硬件根据 DMA 传输完成信号自动置位 STOP 寄存器产生 STOP 时序。可以通过设置 AUTOEND 寄存器来选择所需的策略。

主机使用 DMA 进行发送的流程如下图：

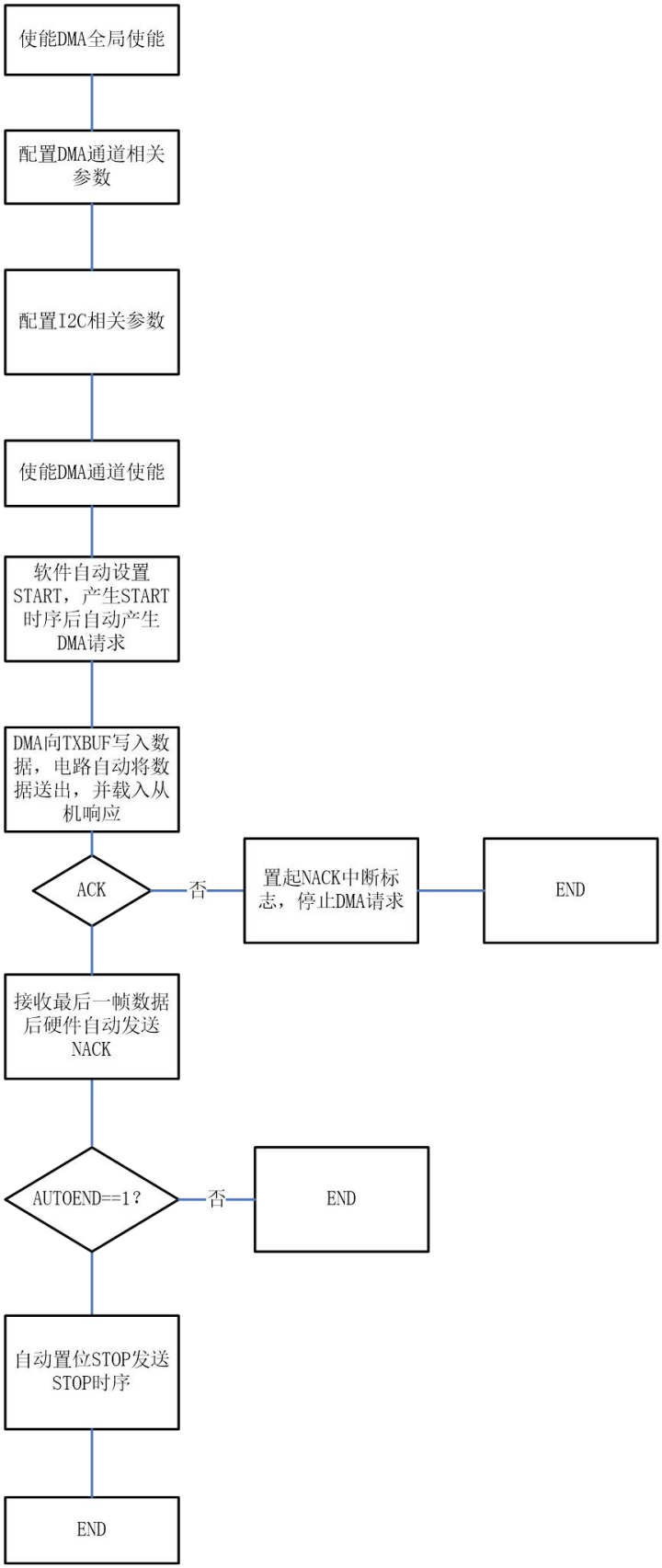


图 19-20 I²C 主机 DMA 发送流程图

主机使用DMA从从机读取数据

这种场景下，从机寻址字节必须由软件发送。软件应事先将目标 DMA 通道配置为 I2CSMB_RX。软件首先发送完从机地址后，设置 RXDMAEN=1，然后使能对应的 DMA 通道为 I2CSMB_RX，I²CSMB 自动进入接收模式，并在每个字节接收完成后产生 DMA 请求，通知 DMA 来读取 RXBUF 内容，同时向从机回发 ACK。

当 DMA 传输达到指定长度后，DMA 的传输完成标志将通知 I2CSMB 回发 NACK。随后根据 AUTOEND 寄存器配置，可以由软件或硬件置位 STOP 寄存器产生 STOP 时序。

注意：当I2CSMB主机通过DMA进行数据接收时，在不同AUTOEND配置和相同DMA传输长度（CHxTSIZE）配置下，DMA接收字节数会有差别。当AUTOEND=0时，接收字节数为CHxTSIZE+1；当AUTOEND=1时，接收字节数为CHxTSIZE。

主机使用DMA进行接收的流程如下图：

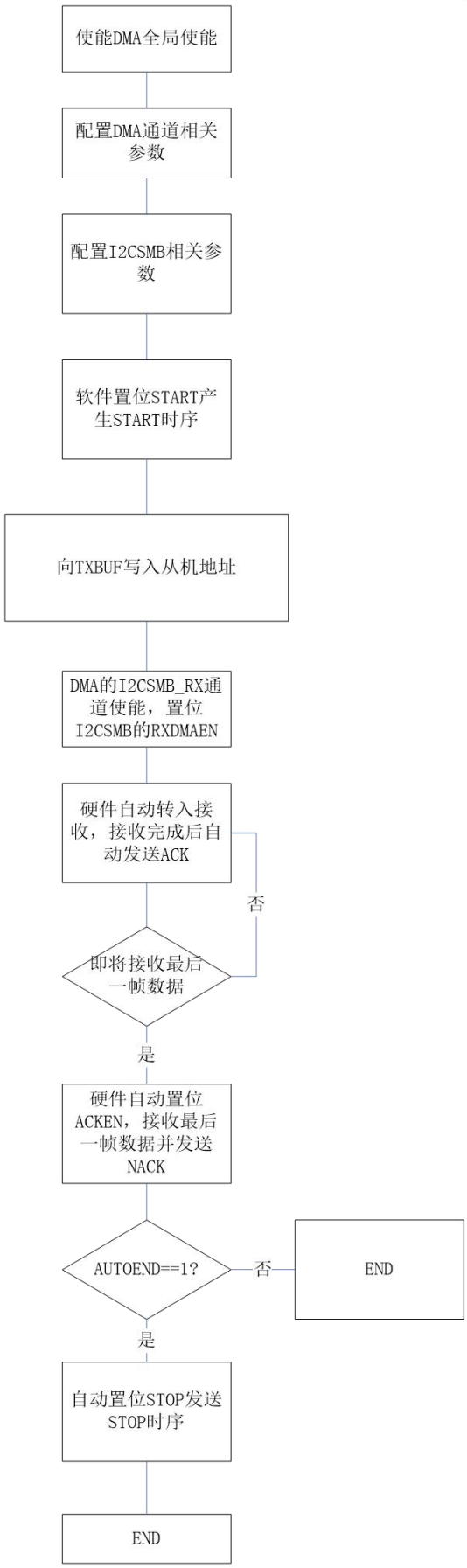


图 19-21 I²C 主机 DMA 接收流程图

19.7.3.4 时钟延展（Slave Clock Stretching）

I²C 总线运行低速从机通过拉低 SCL 的方式通知主机暂停数据通信。I²C 主机必须支持这一特性，因此在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，需要自动检查总线上 SCL 的实际电平，如果不是高电平，意味着从机正在进行 SCL 延展，主机会持续监控 SCL 电平，直到 SCL 为高，才开始后续操作。

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。如果使能了模拟滤波或数字滤波，则主机检查的 SCL 为总线输入经过滤波后的信号。

19.7.3.5 多主机时钟同步（Clock Synchronization）

时钟同步是I²C协议针对多主机通信的一种同步机制。当多个主机在空闲总线上同时开始传输数据时，时钟同步机制保证了总线时钟不会出现冲突状态。

时钟同步采用线与机制，当多个主机拉低SCL时，SCL低电平时间较短的主机在试图拉高SCL时会发现其他主机仍在驱动SCL低电平，此时主机应保持无驱动的等待状态直到SCL总线被拉高，然后再开始计时SCL高电平宽度。采用线与的方式，总线上的同步时钟的低电平总是由SCL低电平长度最长的主机决定，而SCL高电平则由高电平长度最短的主机决定。

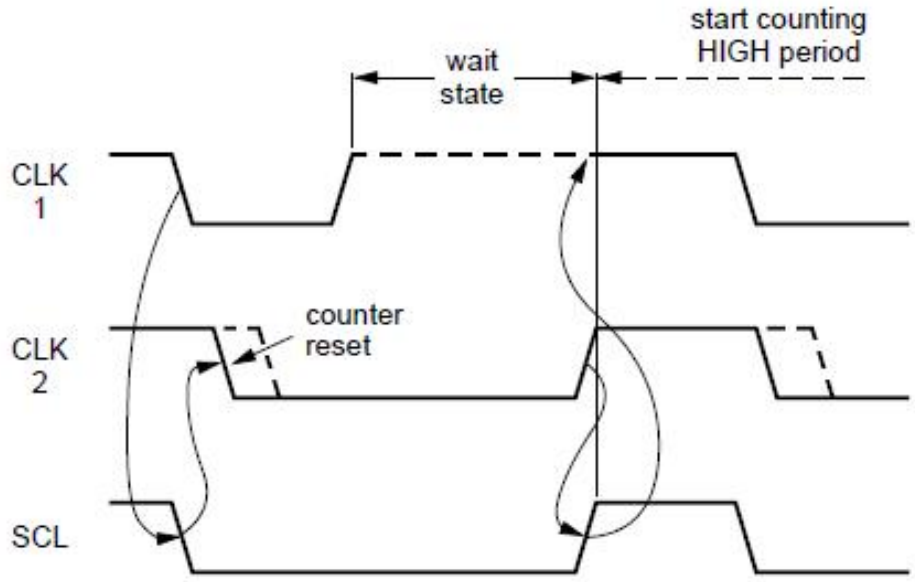


图 19-22 I²C 多主机时钟同步

19.7.3.6 多主机仲裁（Arbitration）

总线仲裁同样仅适用于多主机总线。当多个主机在START条件最小保持时间（$t_{HD,STA}$）内同时发送START条件时，必须通过仲裁机制避免总线冲突并保证数据通信的完整性。

仲裁是基于每个数据bit进行的，当SCL为高电平时，每个主机应实时检测总线上的SDA信号是否与自己发送的数据电平相符。注意到总线的线与特点，因此数据冲突只会在本机发送1，而SDA总线为0的情况下被检测到。此时本主机发现自己输掉仲裁，应主动关闭SDA输出驱动逻辑，便于其他主机继续正常通信，如果模块同时使能了从机模式，模块会立即进入从机接收状态。

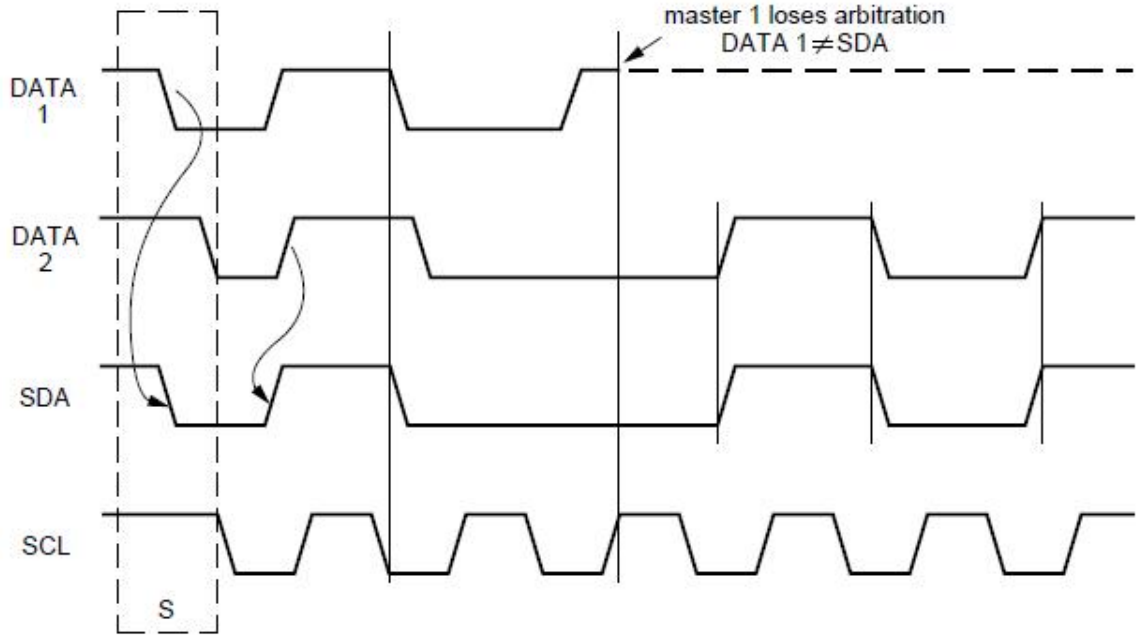


图 19-23 I²C 多主机数据仲裁

19.7.3.7 超时机制

I²C主机还实现了超时机制，即发现从机长时间拉低SCL导致总线无法通信的情况下，产生超时报警中断并返回IDLE状态。

当主机检测到SCL延展，其内部定时器开始计时，主机设定的SCL延展超时的时长最长是4096个SCL周期，假设波特率为100K，则超时周期大约是40ms，如果波特率是400K，则超时周期大约是10ms。通过12bit的TIMEOUT寄存器，软件可以设置超时周期。软件必须在EN为0的情况下设置TIMEOUT寄存器，此寄存复位值为0xFFF，即表示最长4096*T_{SCL}的超时周期，当检测到SCL延展后，TIMEOUT寄存器开始向下递减，当计数到0时，计数停止，TIMEOUT寄存器被复位到0xFFF，同时触发超时中断。因此通过修改TIMEOUT初始值，可以设定超时周期。

$T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

当发生TIMEOUT中断时，建议软件复位I²CSMB模块。

此功能可以被关闭，如果关闭硬件超时，软件也可以通过定时器结合SCL引脚状态判断来自行实现任意长度的超时判决。

19.7.3.8 可编程时序

I²C 模块的主机模式提供了灵活的时序编程特性，允许用户定义 SCL 时钟的低电平宽度、高电平宽度，SDA 数据的建立和保持时间。

通过 BRGH/BRGL 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立时间长度。

SDAHD 时序对于 SDA 数据位和 ACK/NACK 都同样有效。

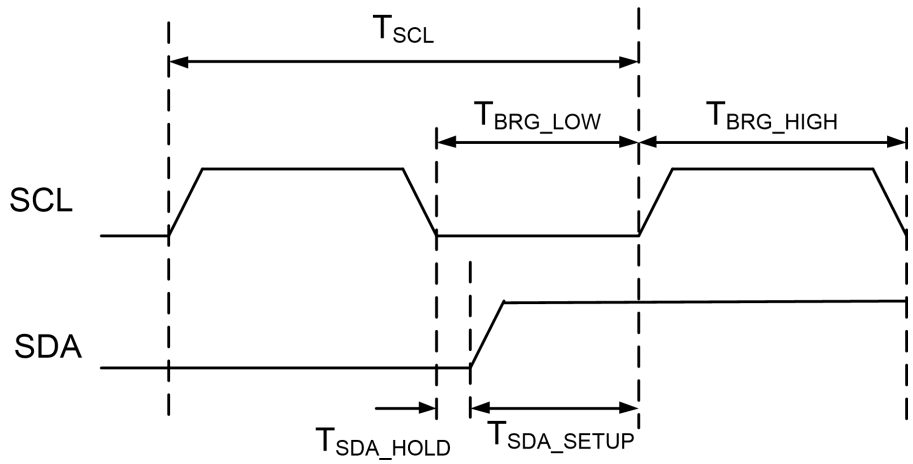


图 19-24 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG\_LOW} + T_{BRG\_HIGH}$

$T_{SDA\_SETUP} = T_{BRG\_LOW} - T_{SDA\_HOLD}$

注意，应用中对BGRH、BRGL和SDAHD寄存器的配置必须满足以下要求，如果违反这些要求将导致异常的总线时序：

BRGH >= 2

BRGL >= 2

BRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

19.7.4 I²C 从机功能

从机接收完 1 字节数据后，产生中断通知 CPU 处理数据，在 CPU 取走数据前硬件可以将 SCL 拉低（软件可以控制是否使能 SCL 延展功能），通知发送方正忙，发送方应暂停发送直到 SCL 放开。若接收方无法响应 ACK，发送方检测 ACK 失败后应发送 P 终止通信或者发送 Sr 开始新的通信。

从机发送完 1 字节数据后，产生中断通知 CPU，硬件拉低 SCL 令主机等待，CPU 响应中断并准备好下一字节数据后再放开 SCL，主机继续发送 SCL 使从机继续数据发送。

19.7.4.1 从机寻址

根据 A10EN 寄存器状态，从机可以支持 7bit 或者 10bit 寻址过程。从机地址由 ADDR 寄存器定义。

对于 10bit 从机地址应用，即 A10EN=1 的情况下，要求第一个字节必须以 11110 开头，否则将触发 ADDR_ERROR 错误标志。而在 A10EN=0 的情况下，如果从机收到了 11110 开头的地址字节，也会置位 ADDR_ERROR 错误标志。

19.7.4.2 从机输入滤波

从机 SCL 和 SDA 输入支持模拟滤波和数字滤波。在使用数字滤波时，必须使能 I2CSMBCLK。如果仅使用模拟滤波，则可以不需要 I2CSMBCLK 工作。

在休眠模式下如果希望 I²C 从机继续工作，则不能使能数字滤波功能，否则由于数字滤波电路无法工作将导致 I²C 模块无法检测总线电平变化。

19.7.4.3 从机发送数据

推荐操作流程：

- 从机接收到地址字节（R/W=1），回发ACK，产生地址匹配中断
- SCLSEN=1，硬件自动进行SCL延展，从机进入发送状态
- 软件响应中断，查询R/W标志，确认是从机发送
- 软件将待发送数据写入TXBUF
- 硬件自动释放SCL
- 新的SCL到来，SDA开始输出
- 接收ACK并产生发送完成中断

- 重复数据发送过程直到接收到STOP时序，或接收到主机NACK

下图是一个典型的从机数据发送波形示意图：

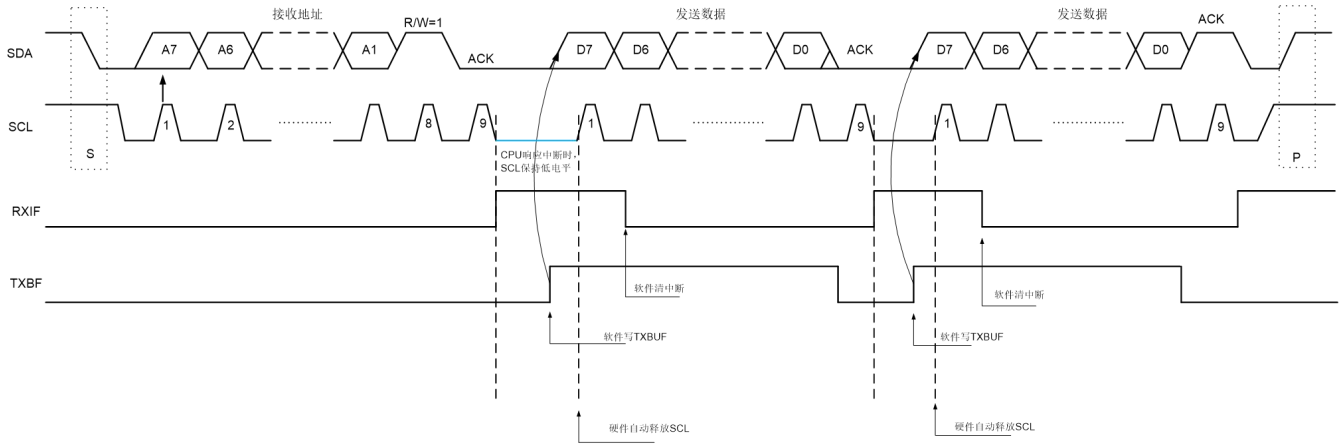


图 19-25 I²C 从机数据发送波形

在从机发送流程中，当从机收到正确地址时，ADIF标志置位，地址字节会被写入RXBUF。硬件自动拉低SCL信号等待软件写入TXBUF，当软件写TXBUF后TXIF置位，同时硬件释放SCL。

19.7.4.4 从机接收数据

推荐操作流程：

- 从机接收到地址字节（R/W=0），回发ACK，产生地址匹配中断
- SCLSEN=1，硬件自动进行SCL延展，从机保持接收状态
- 软件响应中断，查询R/W标志，确认是从机接收
- 软件读RXBUF硬件自动释放SCL，开始接收数据
- 主机数据字节到来，字节接收完成后硬件置位RXIF标志
- 从机回发ACK，并产生接收完成中断
- 硬件自动进行SCL延展（SCLSEN=1）
- 软件响应中断，读取RXBUF，硬件自动清零RXIF标志
- 硬件自动释放SCL
- 重复数据接收过程直到接收到STOP时序，或者软件将ACKEN置0

下图是一个典型的从机数据接收波形示意图（SCLSEN=1）：

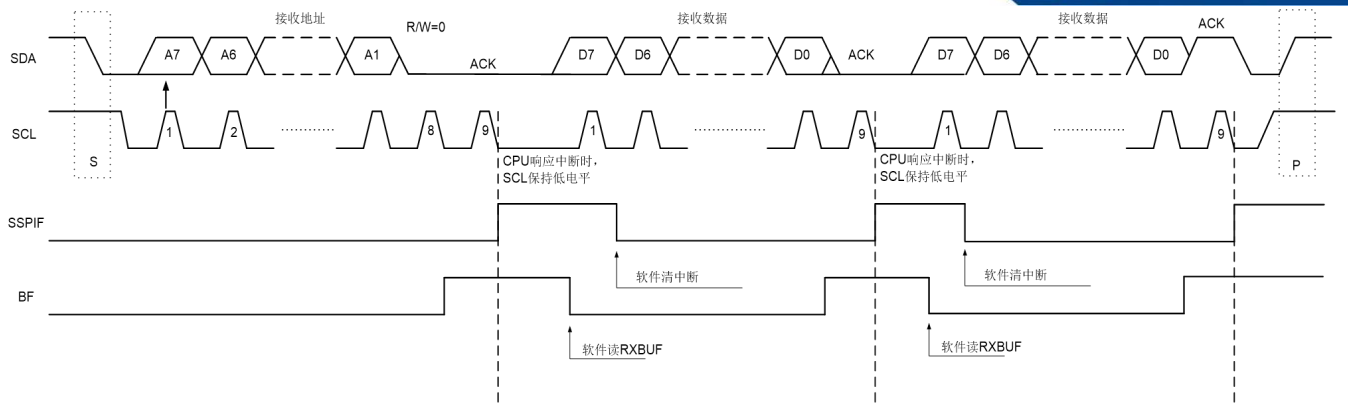


图19-26 I²C从机数据接收波形

从机接收过程中，从机首先收到地址字节，地址匹配的情况下，ADIF标志置位，地址字节将被写入RXBUF并置位RXIF标志，然后硬件拉低SCL。当软件读取RXBUF后，RXIF标志自动清零，硬件释放SCL，可以进行后续数据接收。

注意：从机接收流程中地址字节会被写入RXBUF并导致RXIF置位，软件需要读取RXBUF来清零RXIF并释放SCL。

从机接收数据可以被动结束通信或主动结束通信。

如果主机主动下发STOP，则从机被动结束本次通信。或者，软件在中断处理程序中将ACKEN寄存器置1，则从机在接收完下一个字节后，将回发NACK，主机接收NACK后将下发STOP结束本次通信。

从机SCL延展

I²C从机默认使能SCL延展（slave clock stretching），但是软件可以关闭这个功能（SCLSEN寄存器）以适应不支持从机SCL延展的主机。

当SCL延展使能的情况下，数据接收完成后，软件只有在SCL延展期间读取接收缓冲区时，才能清零RXIF标志。如果接收中出现了数据溢出，OVF标志置位，此时硬件回发NACK，并且SCL不再被延展，以便主机下发STOP；OVF置位的情况下，建议软件等待STOP标志置位，再读取接收缓冲区清零RXIF标志。

接收数据溢出

当从机接收缓冲区满（RXIF=1）时，如果又收到新的数据，则发生接收溢出，OVF标志置位。接收缓冲区中的老数据将被新的数据覆盖。只有在从机关闭了SCL延展功能的情况下，才有可能发生接收数据溢出。

下图是SCLSEN=0情况下发生数据接收溢出的示意图：

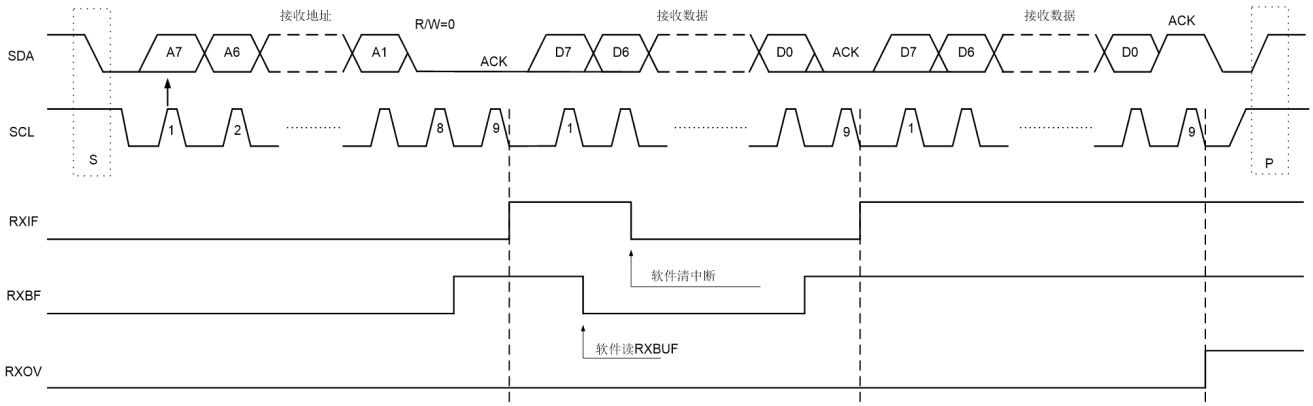


图19-27 I²C从机数据接收波形（SCLSEN=0，接收溢出）

19.7.4.5 从机低功耗接收唤醒

当芯片处于休眠模式时，I²C从机支持START时序唤醒和地址匹配唤醒。

- START时序唤醒RCHF时钟用于从机地址接收
- 从机地址匹配的情况下唤醒MCU进行后续通信

软件也可以选择在检测到START时序时直接唤醒MCU。如果配置为START仅唤醒RCHF时钟，则从机检测到START事件后自动打开RCHF，以便模块能够完成地址字节接收和超时检测。

如果时钟唤醒后发现地址不匹配，则I²C从机不会唤醒MCU，而是自动关闭RCHF。

注意：使用I²C从机休眠唤醒功能时，必须将I2CSMBCLK配置为RCHF，并关闭数字滤波功能。

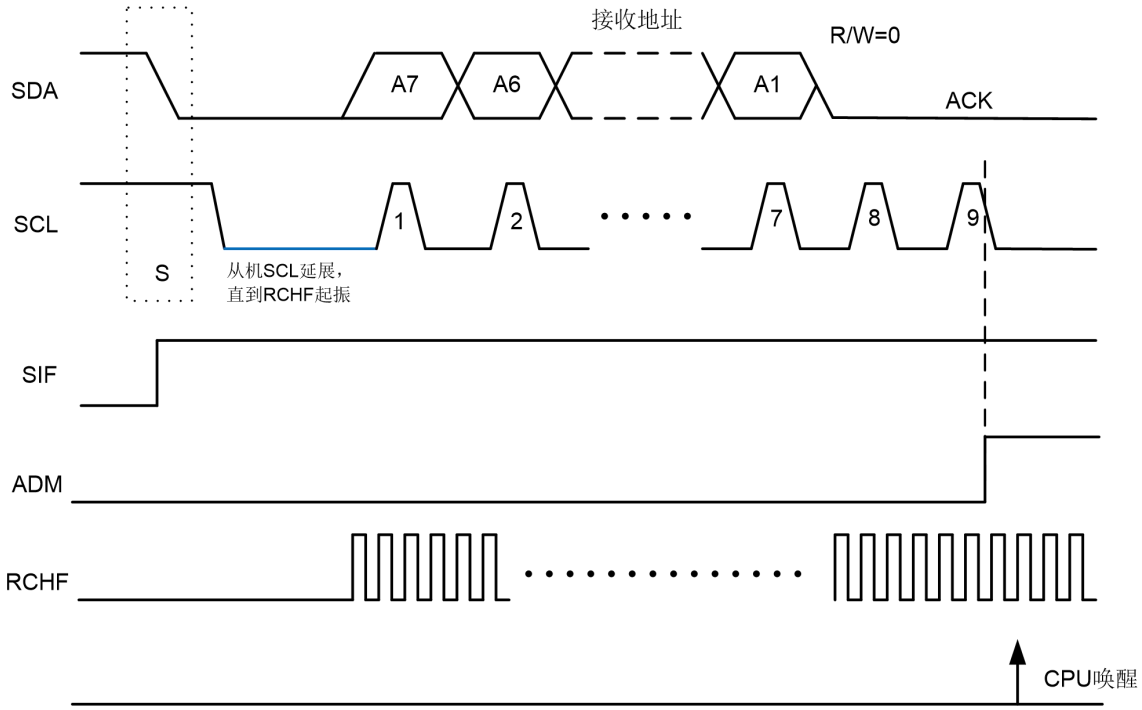


图 19-28 从机地址匹配唤醒

19.7.4.6 DMA

I²C 从机支持 DMA 操作，需要注意的是，必须在 I²C 的总线时钟（I2CSMB_PCE）使能的情况下才能进行 DMA 操作。总线时钟被用于产生 DMA 请求并接收 DMA 应答。

从机使用DMA接收数据

当 I²C 从机接收到正确的地址后，产生 ADIF 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 0 表示主机准备向从机写入数据。此时软件可以配置特定 DMA 通道为 I2CSMB_RX，并使能 I²C 从机的 RXDMAEN；随后每次从机完成一个字节的接收，将产生 DMA 请求，通知 DMA 来读取 RXBUF。

结束 DMA 从机接收有两种可能：

1）数据传输长度还未达到 DMA 长度配置，主机就下发了 STOP 时序，软件应响应 STOP 中断并主动处理这种情况；

2）数据传输长度达到 DMA 长度配置，但是由于 DMA 请求是在从机回发 ACK 后产生，所以软件应响应 DMA 传输完成中断，并将 ACKEN 置 1，这样从机会在接收完下一个字节后，回发 NACK，结束本次通信。

从机使用 DMA 进行接收的流程如下图：

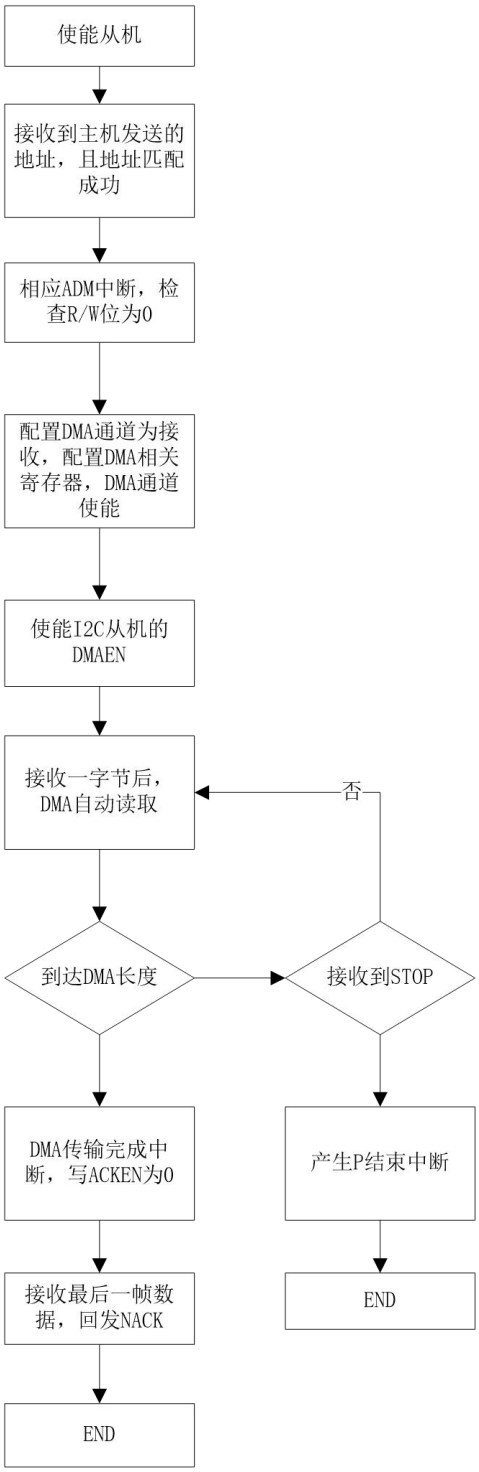


图19-29 I²C从机DMA接收流程图

从机使用DMA发送数据

当 I²C 从机接收到正确的地址后，产生 ADIF 中断标志，软件响应中断后，查询接收到的 R/W 位，如果为 1 表示主机准备从从机读出数据。此时软件需要配置特定 DMA 通道为 I2CSMB_TX，并使能

I2C 从机的 TXDMAEN；随后当从机数据缓存 TXBUF 为空时，将产生 DMA 请求，通知 DMA 写入 TXBUF。

只有主机回发 NACK 才能结束读取操作。当读取数据长度大于 DMA 设置的传输长度时，由于 DMA 不再响应 I²C 请求，从机将一直拉低 SCL，直到软件关闭 I²C 从机模块。

从机使用 DMA 进行发送的流程如下图：

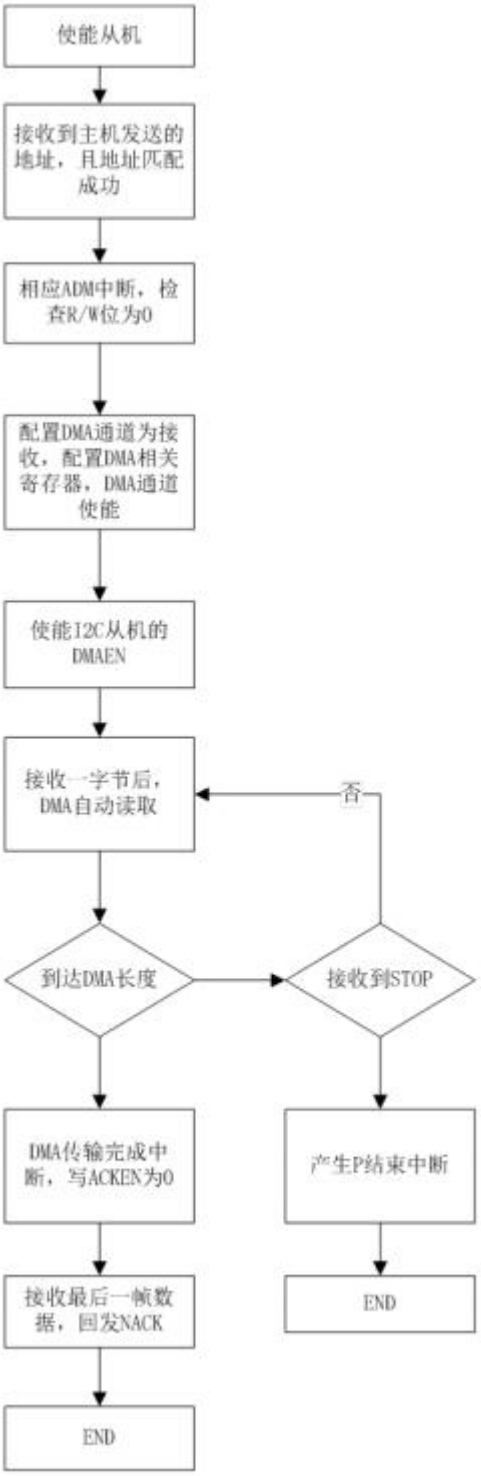


图19-30 I2C从机DMA发送流程图

19.8 SMBus 功能描述

19.8.1 概述

SMBus是基于I²C物理层的双线同步通信协议，可用于系统控制和电源管理任务，支持动态地址分配以实现热插拔应用。本模块支持SMBus rev3.0，更多关于SMBus协议的详情请参见（http://smbus.org）

SMBus规定了三种类型的设备：

- 从机：接收主机发送的命令并给出响应
- 主机：发送命令和时钟，能够启动或终止通信
- Host：一种特殊的主机，能够提供面向系统CPU的接口。Host必须能够作为主机和从机，并支持host notify protocol。整个系统中只允许存在一个host。

本模块可以支持从机、主机、host模式。

SMBus协议层是一个基于命令的操作方式，参见SMBus Specification了解协议定义的命令集。

Address resolution protocol (ARP)

SMBus通过ARP协议可以实现对从机的动态地址分配，因此可以实现系统“热插拔”。本外设模块支持SMBus默认从机地址（device default address: 1100 001）识别，通过置位SMBDEN寄存器来使能此功能。

为了实现ARP，从机软件需要维护一个128位的SMBus系统唯一ID（UDID）；当从机接收到device default address时，中断标志置位通知软件系统主机正在进行ARP枚举协议，软件应完成ARP协议流程以实现动态从机地址分配。

Host Notify protocol

本模块支持host notify协议，当SMBHEN寄存器置位时，外设能够响应SMBus host address（0001 000x），并且能够同时作为系统主机和host从机进行通信。

SMBus Alert

SMBus Alert是一个可选的信号，一个SMBus从机可以通过拉低SMBus Alert来向host申请发起通信。Host接收到SMBALERT#信号后，通过Alert Response Address向所有具备SMBALERT#能力的从机发起访问。只有拉低SMBALERT#的从机会响应Alert Response Address寻址。

在SMBus从机模式下，当ALERTEN=1的情况下，软件可以通过置位ALERT寄存器拉低SMBA引脚。当从机接收到ARA并识别到自己的地址后，从机应清零ALERT寄存器以释放SMBALERT#信号。

在SMBus Host模式下，如果ALERTEN寄存器置位，当检测到SMBA引脚上的下降沿，ALERT标志寄存器置位并可以产生相应的中断事件。

Packet Error Checking (PEC)

SMBus的数据包检错功能通过CRC-8检错码实现，PEC是SMBus器件的可选功能，但是需要支持ARP的器件必须支持PEC。

检错码被附加在数据通信的最后，CRC-8计算被应用于所有的消息字节，包含地址和R/W位，不包含ACK/NACK、START、STOP、RESTART。

SMBus从机在接收过程中发现PEC错误时，回发NACK指示PEC错误。不过由于ACK/NACK位本身也有可能出现总线传输错误，主机不应将从机回发ACK视为PEC正确。从机模式下本模块可以支持回发或不回发PEC。

PEC使用的CRC-8多项式：$C(x)=x^8 + x^2 + x^1 + 1$

Timeouts

本模块包含一组硬件定时器用于实现SMBus协议规定的三种超时类型。当模块检测到超时事件时，需要复位状态机，停止数据收发，恢复IO默认无驱动上拉状态，并准备好接收新的START事件。

Timeout只检测时钟信号，当总线时钟被拉低后，超时计数器开始工作。

符号	参数	limits		unit
		min	max	
$t_{TIMEOUT}$	SCL 低电平超时	25	35	ms
$t_{LOW:SEXT}$	累积 SCL 低电平扩展时间（slave）	-	25	ms
$t_{LOW:MEXT}$	累积 SCL 低电平扩展时间（master）	-	10	ms

表 19-4SMBus 超时事件

1. $t_{LOW:SEXT}$是一个从机在一次通信中（START到STOP）允许的时钟拉低累积时间
2. $t_{LOW:MEXT}$是一个主机在每个字节内（START-to-ACK, ACK-to-ACK, ACK-to-STOP）允许的时钟拉低累积时间

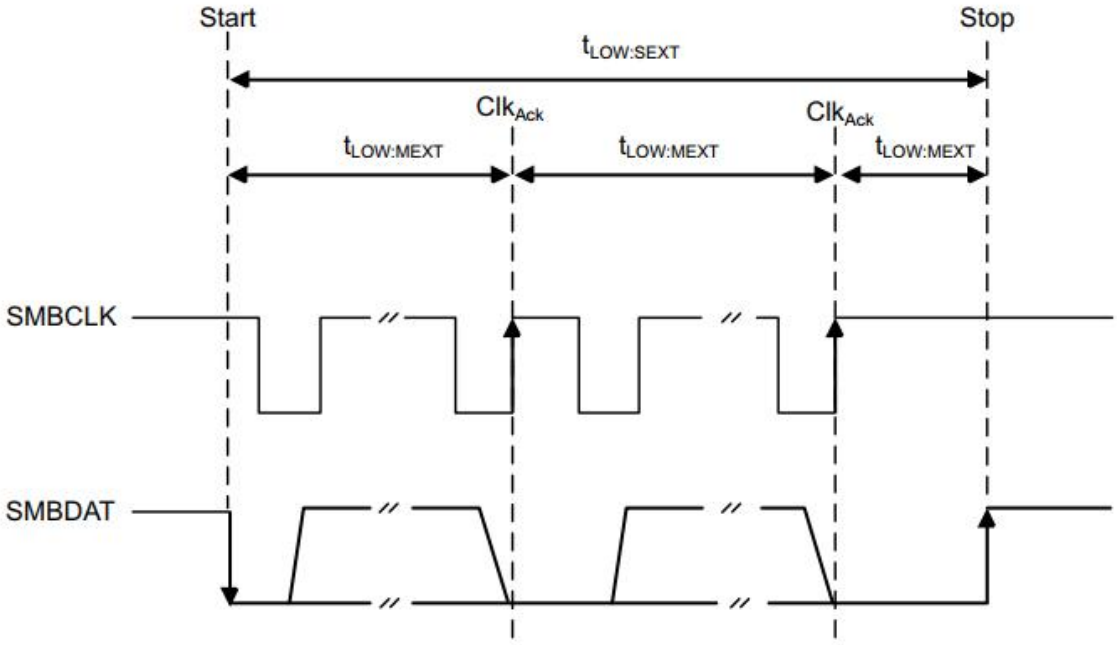


图 19-31$t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 波形

总线IDLE检测

SMBus主机在检测到总线时钟和数据保持高电平超过$t_{HIGH,MAX}$时认为总线已经处于空闲状态，可以发起通信。

本模块支持硬件总线IDLE检测，t_{IDLE}判决时间由TIMEOUT寄存器定义。

19.8.2 SMBus 初始化过程

除了I²C相关的初始化，SMBus特殊的初始化过程包含以下内容。

特殊地址（从机模式）

- SMBus Device Default address (1100 001)通过SMBDEN寄存器使能
- SMBus Host address (0001 000)通过SMBHEN寄存器使能
- Alert Response address (0001 100)通过ALERTEN寄存器使能

Packet Error Checking

PEC校验字节的计算通过模块自带的CRC-8功能实现。

SMBus在收发数据的同时，如果置位了CRCEN，则硬件将接收到的数据bit同步送入CRC-8进行串行移位计算。

在SMBus从机接收模式下，软件应根据PEC比对结果控制PEC字节对应的ACK或者NACK回发。

Timeout检测

超时检测使用TIMEOUTx硬件定时器实现。其中TIMEOUTA是一个12bit定时器，在TIDLE寄存器清零的情况下，TIMEOUTA用于总线SCL低电平超时检测。当SCL拉低时间超过(TIMEOUTA+1)×2048×t_{I2CCLK}时，总线超时标志寄存器被硬件置位，并可以产生中断事件。

另一个TIMEOUTB 12bit定时器用于$t_{LOW:SEXT}$和$t_{LOW:MEXT}$超时检测。当累积SCL延展时间超过(TIMEOUTB+1)×2048×t_{I2CCLK}时，时钟延展超时标志寄存器被硬件置位，并可以产生中断事件。

Bus IDLE检测

当TIDLE寄存器置位时，TIMEOUTA用于总线IDLE检测。如果总线上的SCL和SDA都保持高电平超过(TIMEOUTA+1)×4×t_{I2CCLK}，总线空闲标志寄存器被硬件置位，并可以产生中断事件。

19.8.3 SMBus 主机功能

19.8.3.1 SMBus 主机发送

SMBus主机发送数据可以通过软件或者DMA方式进行。

为了支持PEC功能，I2CSMBUS模块自带CRC-8校验单元，在使能CRC的情况下（CRCEN=1），CRC-8单元自动完成对发送数据的CRC计算。软件需要在数据发送完成后，从CRCDR读取发送数据的CRC结果并发送出去。

DMA发送时，有两种操作方法：

- 关闭AUTOEND功能，在DMA发送完全部数据字节后，再由软件从CRCDR读取PEC字节并发送出去
- 预先对要发送的数据块预计算CRC结果，紧跟着保存在数据块后面；DMA发送时可以使能AUTOEND，发送长度为数据块长度+1字节CRC

19.8.3.2 SMBus 主机接收

SMBus主机接收数据可以通过软件或者DMA方式进行。

使用软件接收数据时，工作流程与I^2C主机接收类似，主要区别在于PEC功能。接收过程中的CRC计算有两种方案：

- CRCEN=0：软件处理CRC计算，每收到1个字节数据，软件负责将数据写入外设CRC模块进行计算。完成所有数据接收后，软件读取CRC计算结果，与最后收到的PEC字节比较。
- CRCEN=1：SMBus模块内部自动进行CRC计算，调用模块内部的CRC-8单元，每收到1字节数据，同步更新CRC-8结果

对于CRCEN=1的接收模式，由于硬件无法自动区分数据字节和CRC字节，软件应在收完最后一个字节数据，并且在开始接收CRC字节之前，及时关闭CRCEN并读取CRC结果。如果软件不能及时处理，则CRC字节接收将改变数据校验结果。

使用DMA接收数据时，如果使能了CRC，则I2CSMBUS模块自动完成接收数据的CRC计算，并将结果保存在CRC寄存器中。这种情况下，为了避免CRC字节接收覆盖数据CRC结果，应将DMA接收长度配置为数据字节长度，不含CRC字节。

19.8.3.3 SMBus 从机发送

SMBus从机发送数据流程与I²C从机类似，置位CRCEN后，PEC通过CRC-8单元计算，并由软件完成PEC字节的发送。

DMA发送时，有两种操作方法：

- 关闭AUTOEND功能，在DMA发送完全部数据字节后，再由软件从CRCDR读取PEC字节并发送出去
- 预先对要发送的数据块预计算CRC结果，紧跟着保存在数据块后面；DMA发送时可以使能AUTOEND，发送长度为数据块长度+1字节CRC

19.8.3.4 SMBus 从机接收

SMBus从机接收数据流程与I²C从机类似，PEC计算同样有两种方式，参考SMBus主机接收章节。

19.8.3.5 SMBus 从机唤醒

在低功耗模式下，SMBus可以在接收到START事件时唤醒MCU，也可以只唤醒RCHF时钟用于从机地址接收。如果配置为唤醒MCU，则从机检测到START事件后会将MCU从休眠模式唤醒。如果配置为仅唤醒RCHF时钟，则从机检测到START事件后自动打开RCHF，以便模块能够完成地址字节接收和超时检测。

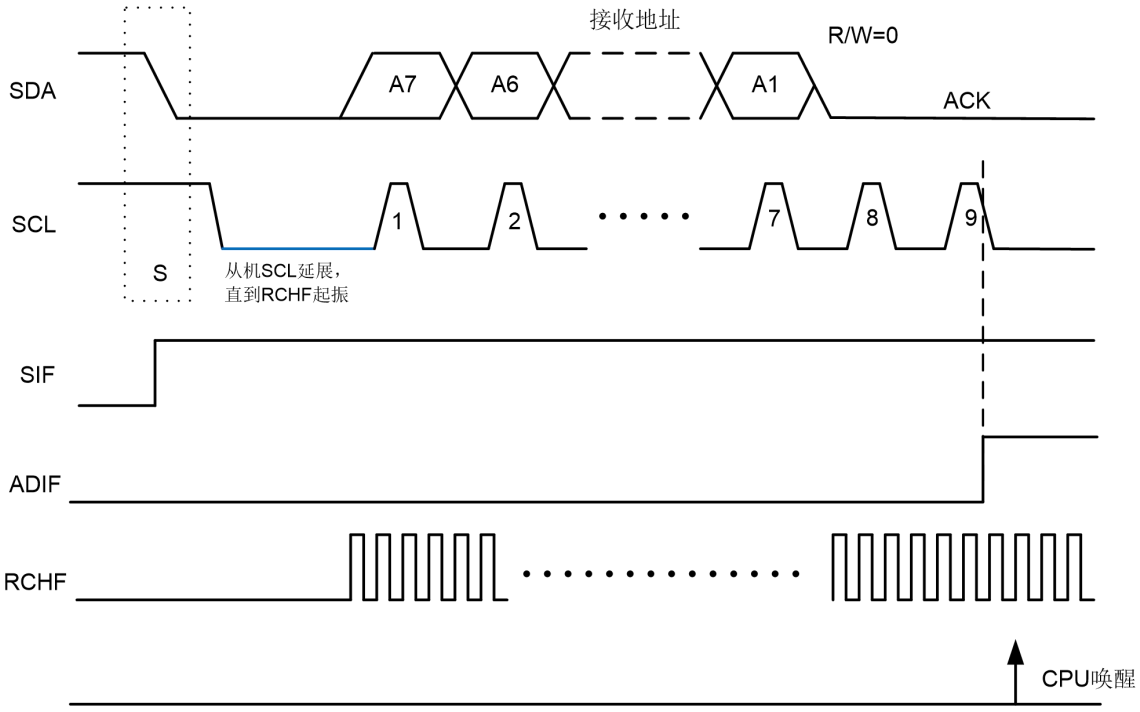


图 19-32 从机地址匹配唤醒

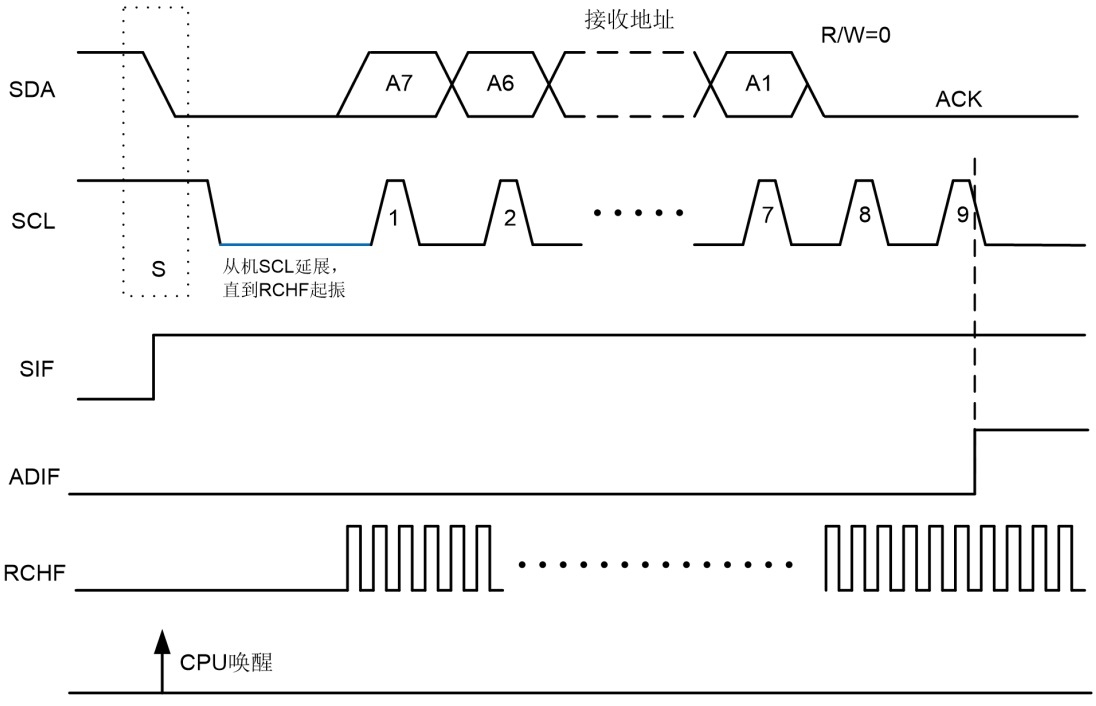


图 19-33 从机 START 唤醒

注意：使用SMBus从机唤醒功能时，必须将I2CSMBCLK配置为RCHF

19.8.4 SMBus 超时功能

SMBus的主机和从机模式都支持$t_{TIMEOUT}$检测，并在检测到超时事件时产生超时中断。

下图显示了总线时钟超时的定义：

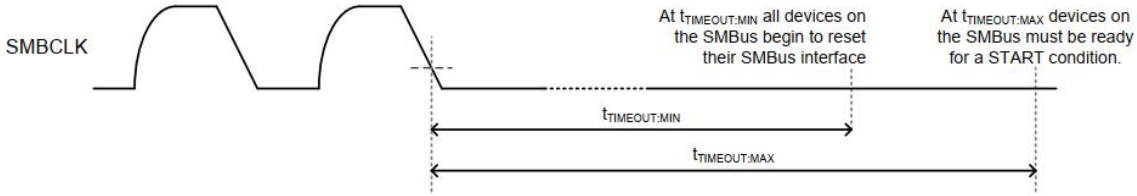


图 19- 34SMBus 总线时钟超时事件

TIMEOUTA寄存器用于定义超时长度，超时时间由下式定义：

$(TIMEOUTA+1)\times2048\times t_{I2CCLK}$

SMBus spec规定的总线超时范围是25~35ms，软件可以根据应用计算合理的TIMEOUTA参数。

当TOEN寄存器置位后，超时计数器在第一次检测到SCL低电平时开始工作，并在每次检测到时钟高电平时自动清零计数器。如果超时计数器溢出，则超时中断寄存器被置位，超时计数器自动关闭，TOEN清零，直到软件下一次启动。

SMBus主机支持$t_{LOW:SEXT}$检测，当主机检测到从机拉低时钟的总时间超过协议规定的上限时，SMBus主机产生超时中断，软件可以选择放弃当前传输，并发送STOP。

SMBus从机支持$t_{LOW:MEXT}$检测，当从机检测到主机拉低时钟的总时间超过协议规定的上限时，SMBus从机产生超时中断，后续行动策略应由软件定义。

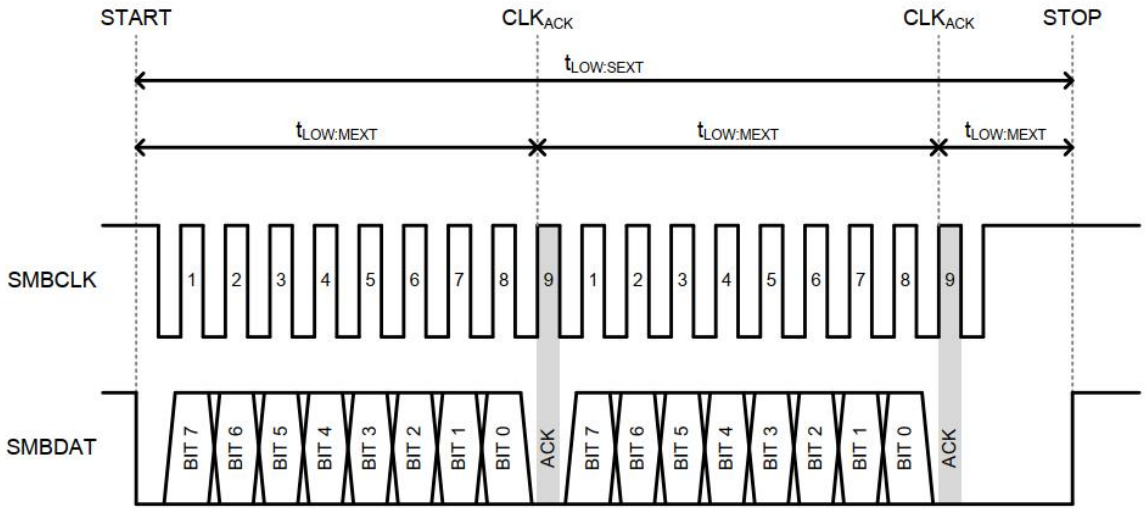


图 19-35 主机和从机超时事件

19.8.5 CRC-8

I2CSMBUS 模块自带 CRC-8 校验单元，采用串行移位的方式实现，初始值和多项式可配置。以下 Golden 数据供应用中测试及校验使用。

多项式	输入序列	初始值(16 进制)		
		全 0	全 F	6363
		CRC 计算结果（16 进制）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96

CRC 功能通道置位 CRCEN 寄存器使能，通信结束后再 CRCDR 寄存器中保存计算结果。

19.9 寄存器

I2CSMB0模块基地址：0x40012400

I2CSMB1模块基地址：0x40012800

offset 地址	名称	符号
I2C_SMB （模块基地址：）		
0x00000000	I2C_SMB 控制寄存器 1（I2C SMBus Control Register）	I2CSMB_CR1
0x00000004	I2C_SMB 控制寄存器 2（I2C SMBus Control Register）	I2CSMB_CR2
0x00000008	I2C_SMB 中断控制寄存器（ I2C SMBus Intterupt Enable Register）	I2CSMB_IER
0x0000000C	I2C_SMB 状态和标志寄存器（ I2C Master Interrupt Status Register）	I2CSMB_ISR
0x00000010	I2C_SMB 波特率设置寄存器	I2CSMB_BGR
0x00000014	I2C_SMB 主机时序控制寄存器	I2CSMB_TCR
0x00000018	I2C_SMB 超时寄存器	I2CSMB_TOR
0x0000001C	I2C_SMB 接收数据寄存器	I2CSMB_RXBUF
0x00000020	I2C_SMB 发送数据寄存器	I2CSMB_TXBUF
0x00000024	I2C_SMB 从机地址寄存器	I2CSMB_SADR
0x00000028	CRC 数据寄存器	I2CSMB_CRC_DR
0x0000002C	LFSR 寄存器	I2CSMB_CRC_LFSR
0x00000030	CRC 多项式寄存器	I2CSMB_CRC_POLY

19.9.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1）

名称	I2CSMB_CR1							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AUTOEND	ALERTEN	SMBDEN	SMBHEN	GCEN	WKUPEN		SCLSEN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-1
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXDMAEN	TXDMAEN	-	ANFEN	DNF			
位权限	R/W-0	R/W-0	U-0	R/W-0	R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CRCEN	A10EN	-	EN

位权限	U-0	R/W-0	R/W-0	U-0	R/W-0

位号	位名	说明
31:24	--	RFU：未实现，读为 0
23	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
22	ALERTEN	SMBus Alert 引脚功能使能 (SMBus Alert pin enable) 1：使能 Alert 功能 device 模式下可以通过写 ALERT 寄存器拉低 Alert 引脚，并且能够响应 Alert Response Address 0001100x (ACK) host 模式下可以响应 Alert 信号 0：禁止 Alert 功能
21	SMBDEN	SMBus device default address 使能 1：响应 SMBus device default address 1100001x (ACK) 0：不响应 SMBus device default address 1100001x (NACK)
20	SMBHEN	SMBus host address 使能 1：响应 host address 0001 000x (ACK) 0：不响应 host address 0001 000x (NACK)
19	GCEN	General Call 响应使能 1：响应 general call 0000 0000 (ACK) 0：不响应 general call 0000 0000 (NACK)
18:17	WKUPEN	休眠唤醒功能使能(Wakeup Enable) 00：禁止从机休眠唤醒功能 01：START 时序唤醒时钟，不唤醒 MCU 10：START 时序同时唤醒时钟和 MCU 11：START 时序唤醒时钟，地址匹配唤醒 MCU
16	SCLSEN	从机时钟延展功能使能，仅在从机模式下有效(slave SCL stretching enable) 1：使能时钟延展 0：禁止时钟延展
15	RXDMAEN	接收 DMA 使能(Receiver DMA Enable) 1：使能数据接收 DMA 功能 0：禁止数据接收 DMA 功能
14	TXDMAEN	发送 DMA 使能(Transmitter DMA Enable) 1：使能数据发送 DMA 功能 0：禁止数据发送 DMA 功能
13	--	RFU：未实现，读为 0
12	ANFEN	模拟滤波使能 (Analog Noise Filter Enable) 1：使能总线输入模拟滤波 0：禁止模拟滤波
11:8	DNF	数字滤波控制(Digital Noise Filter) 0000：禁止数字滤波 0001：数字滤波长度 1 个 I2CSMBCLK 0010：数字滤波长度 2 个 I2CSMBCLK …… 1111：数字滤波长度 15 个 I2CSMBCLK
7:4	--	RFU：未实现，读为 0
3	CRCEN	CRC 计算使能 1：使能 CRC 计算功能

位号	位名	说明
		0：关闭 CRC 计算
2	A10EN	10bit 地址使能，仅从机模式有效 1：从机使用 10bit 地址 0：从机使用 7bit 地址
1	--	RFU：未实现，读为 0
0	EN	I2C_SMB 模块使能控制位 (module enable) 1：模块使能 0：模块禁止

19.9.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2）

名称	I2CSMB_CR2							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		ALERT	ACKEN	RWN	STOP	RESTART	START
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:6	--	RFU：未实现，读为 0
5	ALERT	SMBus Alert pin 控制寄存器，仅在 ALERTEN=1 时有效 1：SMBus 模块拉低 Alert 引脚 0：SMBus 模块撤销 Alert 引脚
4	ACKEN	主机或从机接收模式下，是否回应 ACK 信号 (ACK output enable) 1：回发 NACK 0：回发 ACK
3	RWN	主机读写方向控制寄存器，此位定义了主机传输方向，应与 Address 字节的 bit0 保持一致，仅主机模式有效。 1：主机读 0：主机写 注意：在主机接收时，每收完 1 个字节数据，硬件会自动清零 RWN，因此连续接收时软件需要在每个字节接收完成后重新置位 RWN
2	STOP	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RESTART	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start Enable)
0	START	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

19.9.3 I2C_SMB 中断控制寄存器（I2CSMB_IER）

名称	I2CSMB_IER							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDLIE	ERRIE	SIE	PIE	NACKIE	ADIE	TXIE	RXIE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7	IDLIE	总线空闲中断使能寄存器 (Bus Idle interrupt enable) 1：允许总线空闲中断 0：禁止总线空闲中断
6	ERRIE	错误中断使能寄存器 (Error event interrupt enable) 1：允许错误中断 0：禁止错误中断 *当 ERRIE 置位后，以下事件可以触发 CPU 中断* *仲裁失败* *总线错误* *数据溢出* *超时事件* *Alert 检测*
5	SIE	START 中断使能 (START event interrupt enable) 1：允许 START 事件产生中断 0：禁止 START 事件产生中断
4	PIE	STOP 中断使能 (STOP event interrupt enable) 1：允许 STOP 事件产生中断 0：禁止 STOP 事件产生中断
3	NACKIE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
2	ADIE	从机地址匹配中断使能(address match interrupt enable) 1：允许从机地址匹配产生中断 0：禁止从机地址匹配产生中断
1	TXIE	发送完成中断使能(transmit finished interrupt enable) 1：允许发送完成产生中断 0：禁止发送完成产生中断
0	RXIE	接收完成中断使能(receive finished interrupt enable) 1：允许接收完成产生中断 0：禁止接收完成产生中断

19.9.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR）

名称	I2CSMB_ISR							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							SDIR
位权限	U-0							R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	-	ALIF	TOB	TOA	OVF	BERR	ARLO
位权限	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUSY	IDLIF	SIF	PIF	NACKIF	ADIF	TXIF	RXIF
位权限	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:17	--	RFU：未实现，读为 0
16	SDIR	从机传输方向，仅从机模式有效，只读；硬件在从机地址匹配后更新 0：主机写操作，从机接收 1：主机读操作，从机发送
15:14	--	RFU：未实现，读为 0
13	ALIF	Alert 标志，SMBus host 模式下有效，检测到 Alert 引脚拉低时硬件置位，软件写 1 清零，写 0 无效
12	TOB	Timeout B 标志，发生 $t_{LOW:MEXT}$ 或 $t_{LOW:SEXT}$ 超时时硬件置位，软件写 1 清零，写 0 无效
11	TOA	Timeout A 标志，发生 SCL 超时硬件置位，软件写 1 清零，写 0 无效
10	OVF	数据溢出标志，在 RXBUF 或 TXBUF 非空时，有新的数据到来，触发硬件置位，软件写 1 清零，写 0 无效 (Overflow Flag)
9	BERR	总线错误标志，表示传输过程中检测到错误的 START 或 STOP 时序，硬件置位，软件写 1 清零，写 0 无效 (Bus Error)
8	ARLO	仲裁失败标志，硬件置位，软件写 1 清零，写 0 无效 (Arbitration Lost)
7	BUSY	总线忙标志，只读 1：总线正在通信中 0：总线空闲
6	IDLIF	SMBus 总线 IDLE 事件标志，硬件置位，软件写 1 清零，写 0 无效(SMBus IDLE)
5	SIF	START 中断标志，从机模式有效。硬件置位，软件写 1 清零，写 0 无效(START event interrupt flag)
4	PIF	STOP 中断标志，从机模式有效。硬件置位，软件写 1 清零 (STOP event interrupt flag)
3	NACKIF	主机/从机发送模式下 NACK 中断标志寄存器，硬件置位，软件写 1 清零，写 0 无效(Non-ACK interrupt flag)
2	ADIF	从机地址匹配中断标志，硬件置位，软件写 1 清零，写 0 无效(address

位号	位名	说明
		match interrupt flag)
1	TXIF	发送完成中断标志，硬件置位，软件写 1 清零，写 0 无效(transmit finished interrupt flag)
0	RXIF	接收完成中断标志，硬件置位，软件写 1 清零，写 0 无效(receive finished interrupt flag)

19.9.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR）

名称	I2CSMB_BGR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BRGH[7:0]							
位权限	R/W-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							BRGL[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRGL[7:0]							
位权限	R/W-00010011							

位号	位名	说明
31:25	--	RFU：未实现，读为 0
24:16	BRGH	主机发送的 SCL 时钟高电平宽度，以 I2CSMB 工作时钟计数 (Master SCL High level length)
15:9	--	RFU：未实现，读为 0
8:0	BRGL	主机发送的 SCL 时钟低电平宽度，以 I2CSMB 工作时钟计数 (Master SCL Low level length)

19.9.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TCR）

名称	I2CSMB_TCR							
Offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	SDAHD[7:0]
位权限	R/W-00001010

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2CSMB 工作时钟计数 (SDA hold delay) 注意：最小有效值为 1，最大有效值为 BRGL

19.9.7 I2C_SMB 超时寄存器（I2CSMB_TOR）

名称	I2CSMB_TOR							
Offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TEXTEN	-			TIMEOUTB			
位权限	R/W-0	U-0			R/W-1111			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TIMEOUTB							
位权限	R/W-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TOEN	-		TIDLE	TIMEOUTA			
位权限	R/W-0	U-0		R/W-0	R/W-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUTA							
位权限	R/W-1111 1111							

位号	位名	说明
31	TEXTEN	累积扩展时钟超时使能 1：主机模式下检测 $t_{LOW:MEXT}$ 超时，从机模式下检测 $t_{LOW:SEXT}$ 超时 0：关闭扩展时钟超时检测
30:28	--	RFU：未实现，读为 0
27:16	TIMEOUTB	用于定义累积扩展时钟超时阈值 主机模式下定义 $t_{LOW:MEXT}$，从机模式下定义 $t_{LOW:SEXT}$ $t_{LOW:EXT}=(TIMEOUTB+1)\times2048\times t_{I2CSMBCLK}$ 注意：禁止在 TEXTEN=1 的情况下改写此寄存器
15	TOEN	时钟超时检测使能 1：SCL 低电平超时检测使能：当 TIDLE=0 时检测 $t_{TIMEOUT}$，TIDLE=1 时检测 t_{IDLE}，阈值由 TIMEOUTA 定义 0：SCL 低电平超时检测禁止
14:13	--	RFU：未实现，读为 0
12	TIDLE	总线 idle 检测使能 1：TIMEOUTA 用于检测总线空闲 0：TIMEOUTA 用于检测 SCL 低电平超时
11:0	TIMEOUTA	TIDLE=0 时用于检测 SCL 超时， $t_{TIMEOUT}=(TIMEOUTA+1)\times2048\times t_{I2CSMBCLK}$ TIDLE=1 时用于检测总线空闲， $t_{IDLE}=(TIMEOUTA+1)\times4\times t_{I2CSMBCLK}$ 注意：禁止在 TOEN=1 的情况下改写此寄存器

19.9.8 I2C_SMB 接收数据寄存器（I2CSMB_RXBUF）

名称	I2CSMB_RXBUF							
Offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	RXBUF	8bit 接收数据，只读

19.9.9 I2C_SMB 发送数据寄存器（I2CSMB_TXBUF）

名称	I2CSMB_TXBUF							
Offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	TXBUF	8bit 发送数据

19.9.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR）

名称	I2CSMB_SADR

Offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SAEN	MASK			-		ADDR	
位权限	R/W-0	R/W-000			U-0		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15	SAEN	从机地址使能 (Slave address enable) 1：使能从机地址匹配，ACK 正确地址 0：禁止从机地址匹配，NACK 任何地址
14:12	MASK	从机地址掩码 000：无掩码 001：mask ADDR[0]，即 ADDR[0]不参与地址比较 010：mask ADDR[1:0]，即 ADDR[1:0]不参与地址比较 011：mask ADDR[2:0]，即 ADDR[2:0]不参与地址比较 100：mask ADDR[3:0]，即 ADDR[3:0]不参与地址比较 101：mask ADDR[4:0]，即 ADDR[4:0]不参与地址比较 110：mask ADDR[5:0]，即 ADDR[5:0]不参与地址比较 111：mask ADDR[6:0]，即 ADDR[6:0]不参与地址比较
11:10	--	RFU：未实现，读为 0
9:0	ADDR	10bit 从机地址，当 A10EN=1 时，10bit 地址全部有效，当 A10EN=0 时，只有低 7bit 地址有效

19.9.11 CRC 数据寄存器（I2CSMB_CRC_DR）

名称	I2CSMB_CRC_DR							
Offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R -1111 1111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**DR**	CRCDR 用于在运算结束后保存 CRC 计算结果。(CRC Data Register)

19.9.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR）

名称	I2CSMB_CRC_LFSR							
Offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**LFSR**	CRC 线性反馈移位寄存器 (Linear Feedback Shift Register) 运算开始前可以由软件写入 CRC 初始值

19.9.13 CRC 多项式寄存器（I2CSMB_CRC_POLY）

名称	I2CSMB_CRC_POLY							
Offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0000 0111							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0

位号	助记符	功能描述
7:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

20 双线串行总线主机（I²C master）

20.1 概述

I²C 主机模块实现 MCU 与外部 I2C 接口器件之间的同步通信。支持 I²C 的主机模式，不支持多主机模式。

特点：

- 4 路独立 I²C 主机接口
- 支持主机模式，不支持多主机模式，不支持从机模式
- 支持 7 位或 10 位从机地址
- 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)和 Fm+(1Mbps)

20.2 结构框图

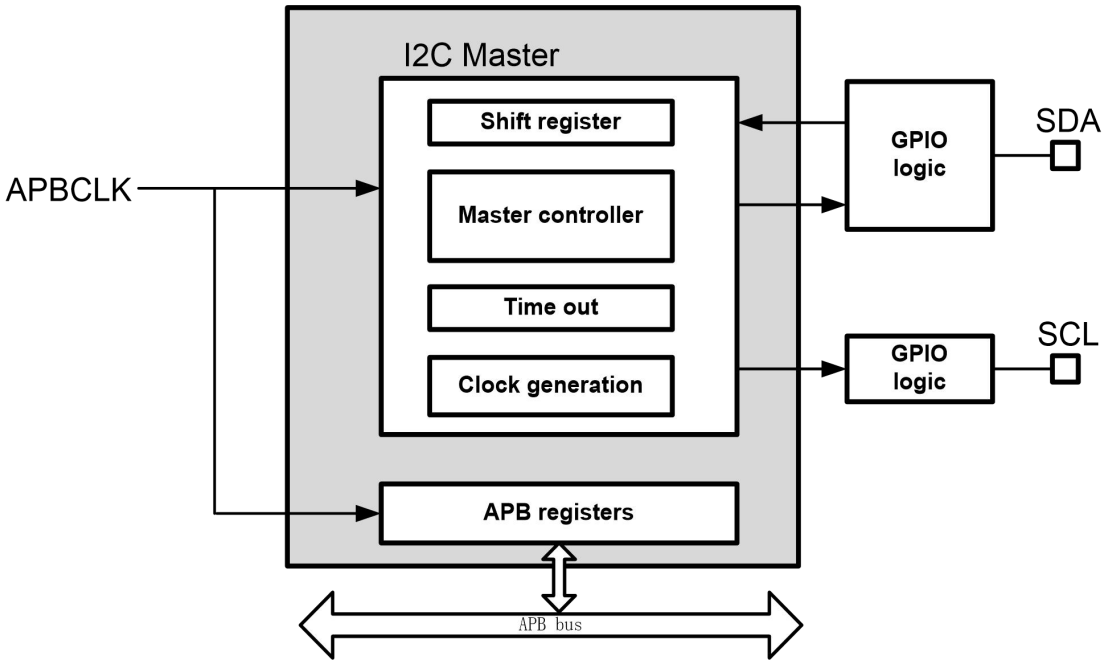


图 20-1 I²C 主机模块框图

20.3 时钟

I²C 主机使用 APBCLK 同步工作。时钟使能控制在 CMU 模块内完成，进行 I²C 通信前必须正确配置相应的 CMU 控制寄存器。

20.4 接口时序

20.4.1 接口时序图

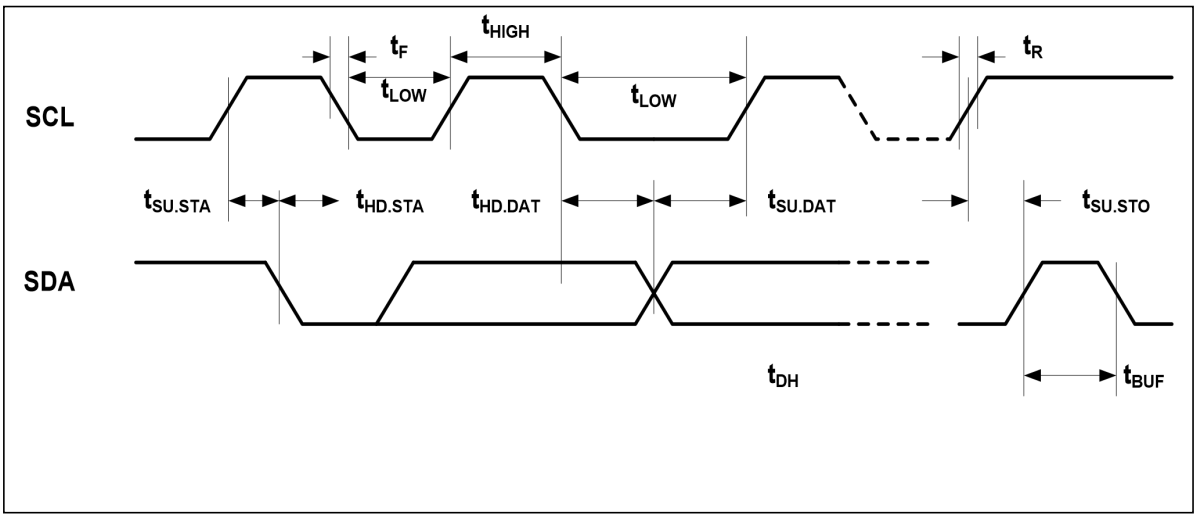


图 20-2I²C 总线时序

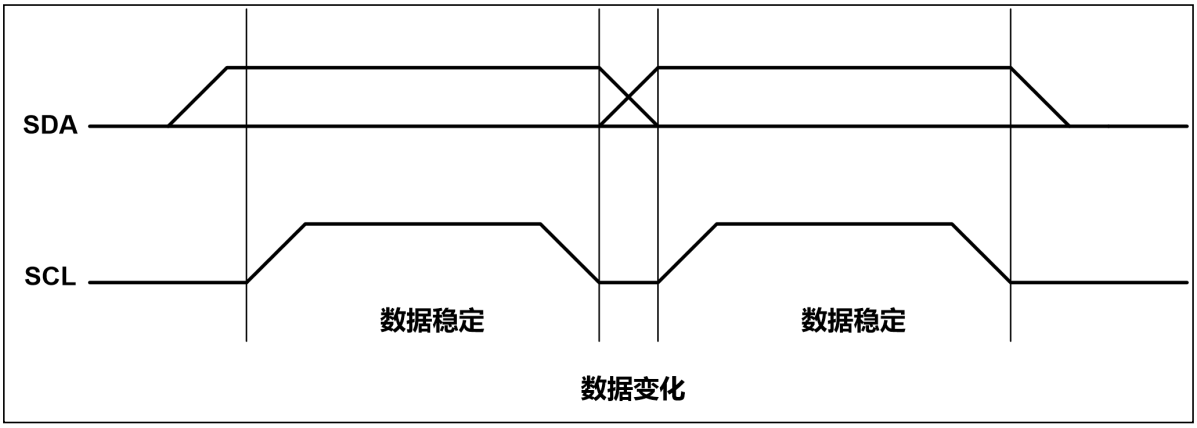


图 20-3 数据有效时序

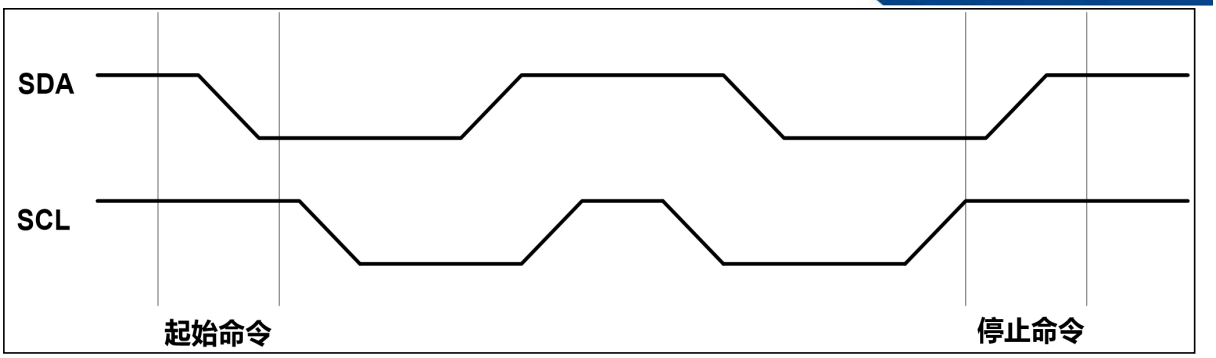


图 20-4 起始（Start）与停止(Stop)命令定义

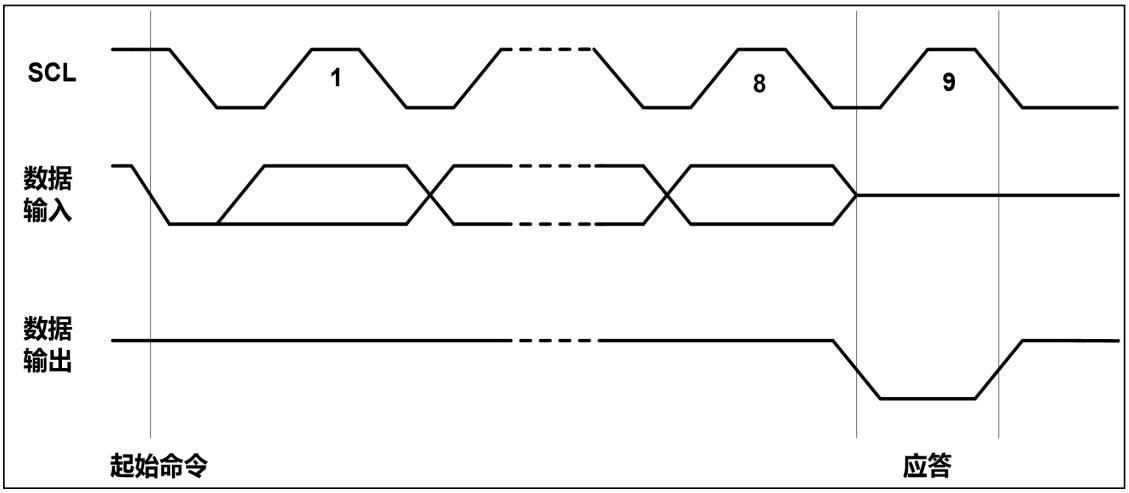


图 20-5 输出应答(ACK)

20.4.2 接口时序描述

时钟有效时序：SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化；当数据在 SCL 为高时变化，将视为下文所述的一个起始或停止命令。

起始命令：当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始。

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使 EEPROM 进入等待态低功耗模式。

输出应答：SDA 上的数据都是以 8 位为一组串行输入和输出的，MSB 先发，接收方在收完每个字节后应当在第 9 个周期回发一个回应 acknowledge 位（以下简称 ack），ack 的时钟由主机提供。发送方在 ack 期间悬空 SDA，接收方须将 SDA 拉低，确保 ack 时钟高电平期间 SDA 为低，形成有效的 ack 信号。

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	F_{SCL}	0	100	0	400	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	—	0.6	—	us
启动条件保持时间	$T_{HD:STA}$	4.0	—	0.6	—	us
时钟低电平时间	T_{LOW}	4.7	—	1.3	—	us
时钟高电平时间	T_{HIGH}	4.0	—	0.6	—	us
数据输入建立时间	$T_{SU:DAT}$	250	—	100(4)	—	ns
数据输入保持时间	$T_{HD:DAT}$	5.0 0(2)	— 3.45(3)	— 0(2)	— 0.9(3)	us us
SDA 和 SCL 上升时间	T_R	—	1000	20+0.1Cb(5)	300	ns
SDA 和 SCL 下降时间	T_F	—	300	20+0.1Cb(5)	300	ns
停止条件建立时间	$T_{SU:STO}$	4.0	—	0.6	—	us
总线空闲时间	T_{BUF}	4.7	—	1.3	—	us
总线的容性负载	Cb	—	400	—	400	Pf
噪声容限低值	V_{nL}	$0.1V_{DD}$	—	$0.1V_{DD}$	—	V
噪声容限高值	V_{nH}	$0.2V_{DD}$	—	$0.2V_{DD}$	—	V

表 20-1 I^2C 接口时序要求

20.5 I²C 工作模式

I²C模块支持以下工作模式：

- 主机接收
- 主机发送

芯片上电后I²C模块默认关闭。软件需要根据应用选择模块工作模式，通过设置MSPEN来使能主机通信。

20.6 I²C 地址格式

I²C总线协议定义了以下保留地址，对其中多数保留地址，I²C硬件不做合法性判断，由软件保证地址的合法性。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

表 20-2 I²C 保留地址定义

20.7 I²C 初始化

进行I²C通信前必须正确的初始化I²C模块，建议软件按照以下步骤进行初始化操作：

- 清零RMU模块的I2CRST寄存器，确保I²C模块不处于复位状态
- 置位CMU模块的I2C_PCE寄存器，使能I²C模块寄存器总线接口时钟

20.7.1 主机波特率配置

I²C 主机需要在使能前配置通信波特率。

MSPBRG[8:0]波特率配置寄存器用于产生通信波特率。MSPBRG 是 9 bit 波特率分频系数，波特率计算公式如下：

$T_{SCL}=2T_{BRG}$

T_{BRG}＝2×T_{I2CCLK}×(MSPBRG[8:0]+1)；T_{I2CCLK}为 I²C 工作时钟周期，即：

MSPBRG = F_{I2CCLK}/(4*F_{SCL}) – 1

例如对于 100k 波特率，若 I²C 工作时钟为 8M，则 MSPBRG=19。

20.8 I²C 主机功能

FM33FR0的I²C主机模式不支持多主机总线，因此挂在总线上的其他设备都是从机。总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。

I²C总线通信总是由主机发起，主机模式支持7bit或10bit寻址。

20.8.1 7bit 寻址

在7bit寻址时，主机发送的第一个字节包含从机地址和传输方向位（$R/\overline{W}$），根据$R/\overline{W}$决定后续传输是主机向从机写入数据（$R/\overline{W}$=0）或主机从从机读取数据（$R/\overline{W}$=1）。

名字	Slave Address Byte							
位	7	6	5	4	3	2	1	0
位名	address							R/W

位描述：

位	位名	功能
7-1	address	Slave device address
0	R/W	0: Write 表示发送数据（master 发送） 1: Read 表示请求数据（slave 回发）

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

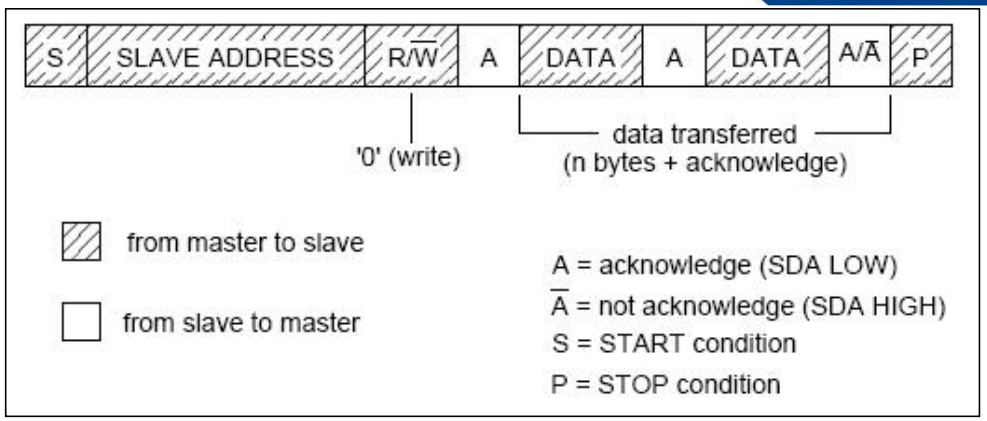


图 20-6 主机向 7 位地址从机写入数据时的帧格式

1、主机发起 START 时序

2、主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

6、若从机无法响应 ACK，主机检测到 NACK 后应发送 STOP 时序终止发送

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

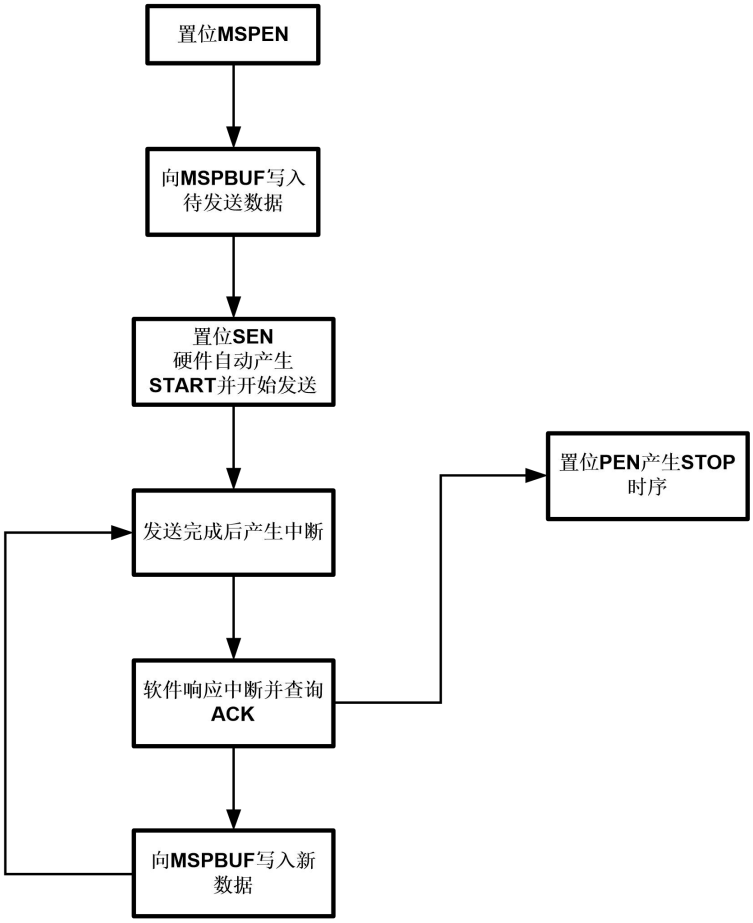


图 20-7I²C 软件发送数据流图

I2C主机对7位地址从机写入数据的波形示意图如下：

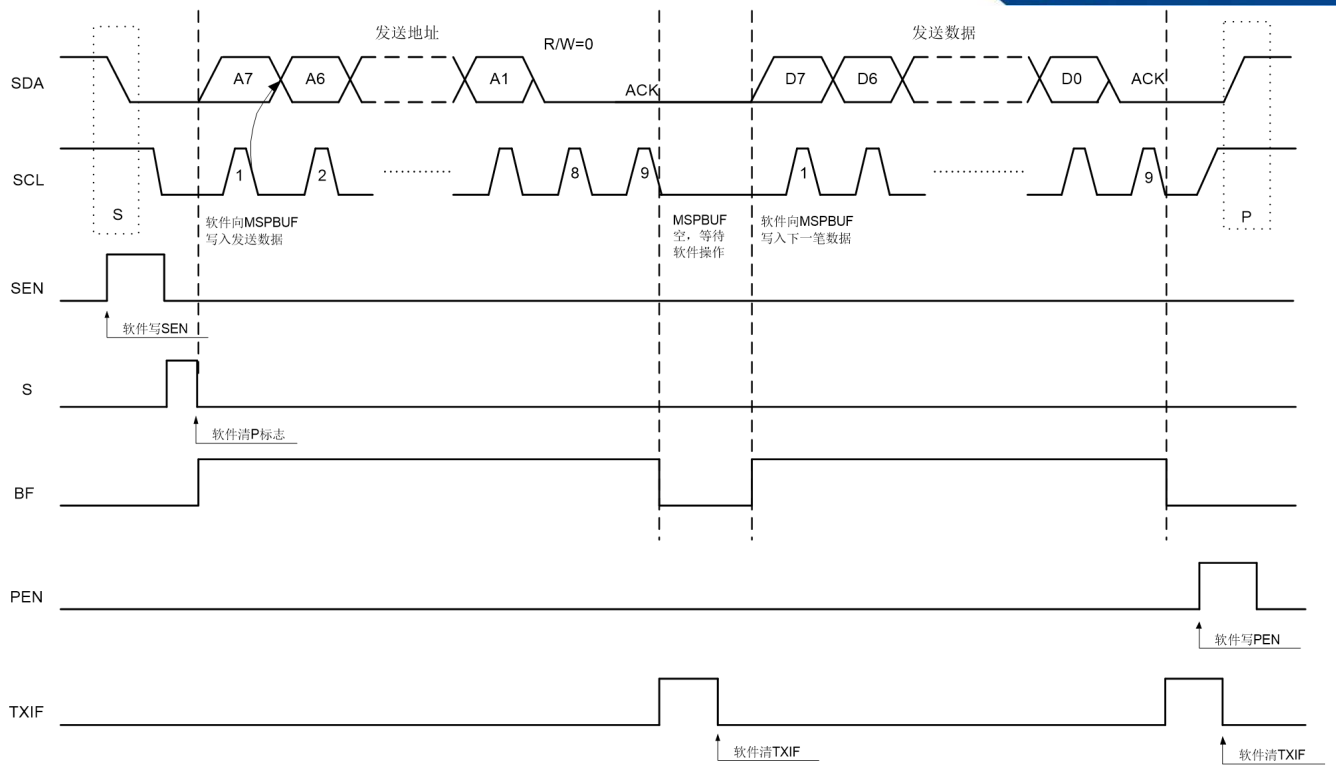


图 20-8 I²C 主机对 7 位地址从机发送数据流图

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

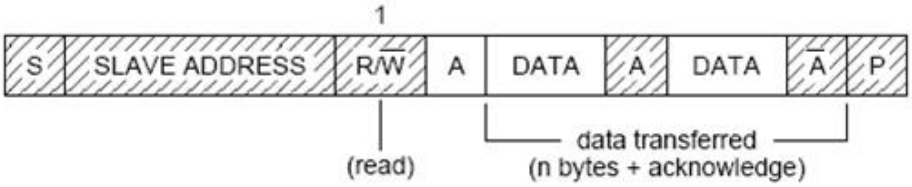


图 20-9 主机从 7 位地址从机读取数据时的帧格式

1、 主机发起 START 时序

2、 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 此时设置 MSPCR.RCEN 为 1，主机自动转为接收状态

4、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

5、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

6、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

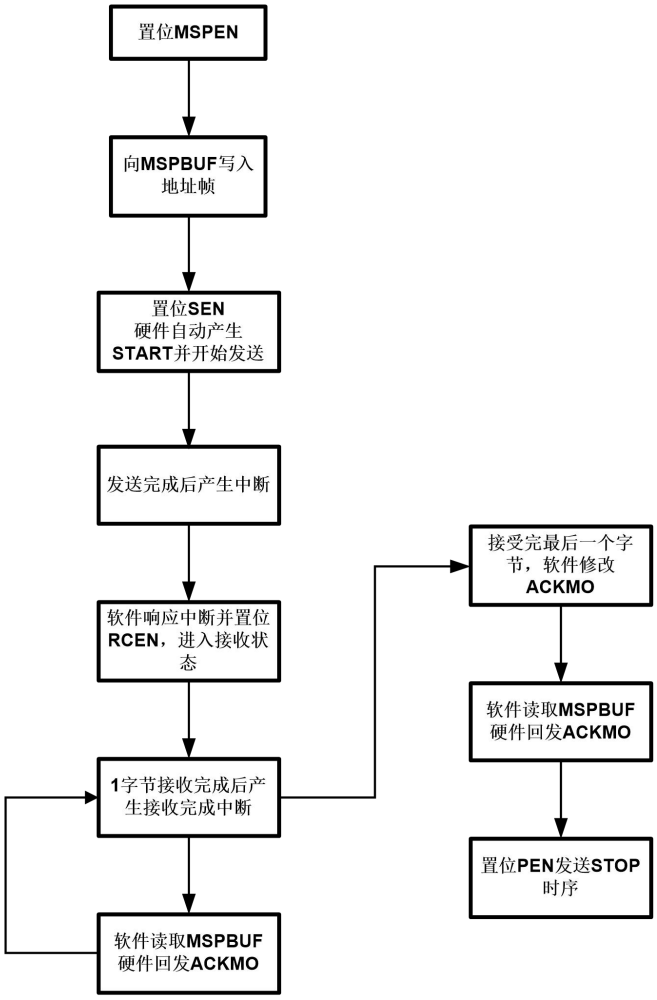


图 20-10 I²C 软件发送数据流图

主机每次接收完从机发送的数据后，根据ACKMO寄存器回发响应。ACKMO复位值为0，即默认状态下主机回发ACK。如果软件希望主机在接收完成后回发NACK，则需要在前一个字节接收完成中断中将ACKMO寄存器改写为1。ACKMO为1的情况下，主机在发送完响应后会自动清零ACKMO。

I²C主机从7位地址从机读出数据的波形示意图如下：

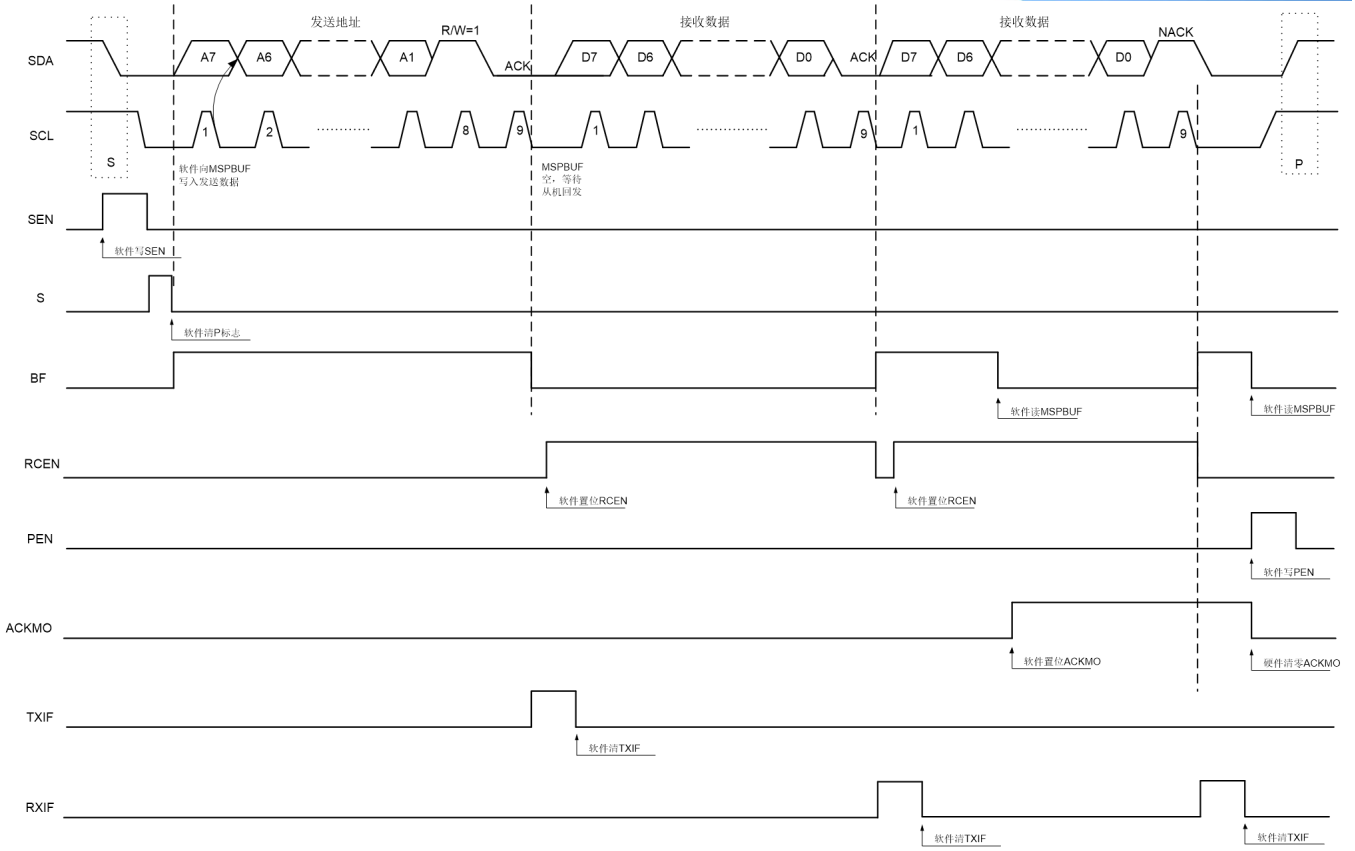


图 20-11 I²C 从 7 位地址从机读取数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以有数据读取。

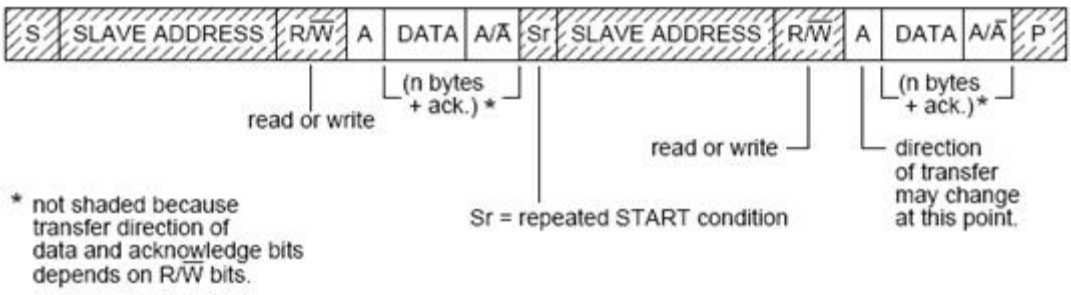


图 20-12 双向数据通信帧格式

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送RESTART时序和从机地址字节来修改传输方向。

20.8.2 10bit 寻址

在10bit寻址时，主机发送的第一个字节包含部分从机地址（11110_A9_A8）和传输方向位（$R/\overline{W}$），第二个字节包含剩余从机地址（A7~A0）。两个字节地址发送完成后，再进行数据传输。

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

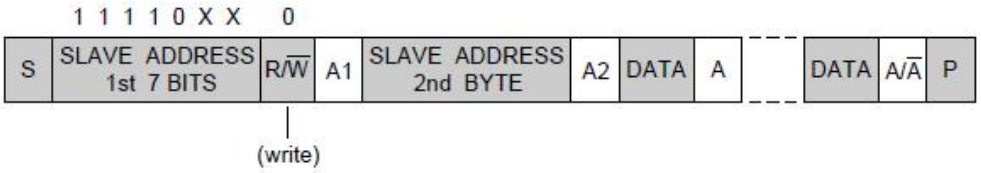


图 20-13 10bit 寻址，主机向从机写入数据

1、主机发起 START 时序

2、主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

软件启动 I²C 主机发送的操作流程如下图：

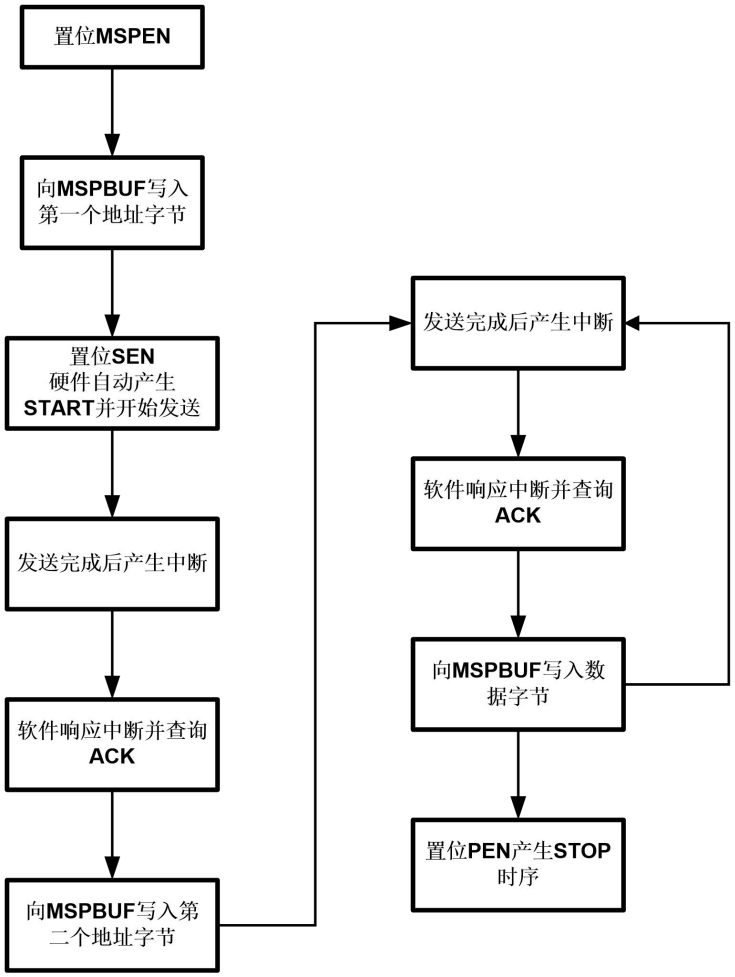


图 20-14I²C 软件发送数据流图

主机从从机读取数据

典型的10bit寻址，主机从从机读取数据的数据流图如下图所示。

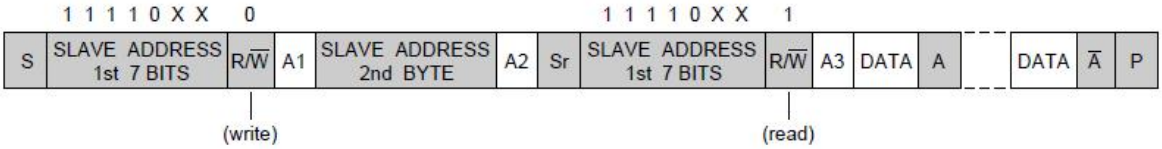


图 20-15 10bit 寻址，主机从从机读取数据

1、 主机发起 START 时序

2、 主机发送第一字节从机地址，包含 5 位前导码 11110、2 位从机地址最高位和 1 位 R/W 标志位，数据读取时 R/W 位为 1

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 RESTART 时序

5、 主机再次发送第一字节从机地址，将 R/W 为改为 0

6、 此时设置 MSPCR.RCEN 为 1，主机转为接收状态

7、 主机开始接收第一字节 8 位数据，并在第 9 个 SCL 向从机发送有效 ACK,从而继续读取下一字节 8 位数据

8、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

9、 主机发送 STOP 时序终止读取

软件启动 I²C 接收的操作流程如下图：

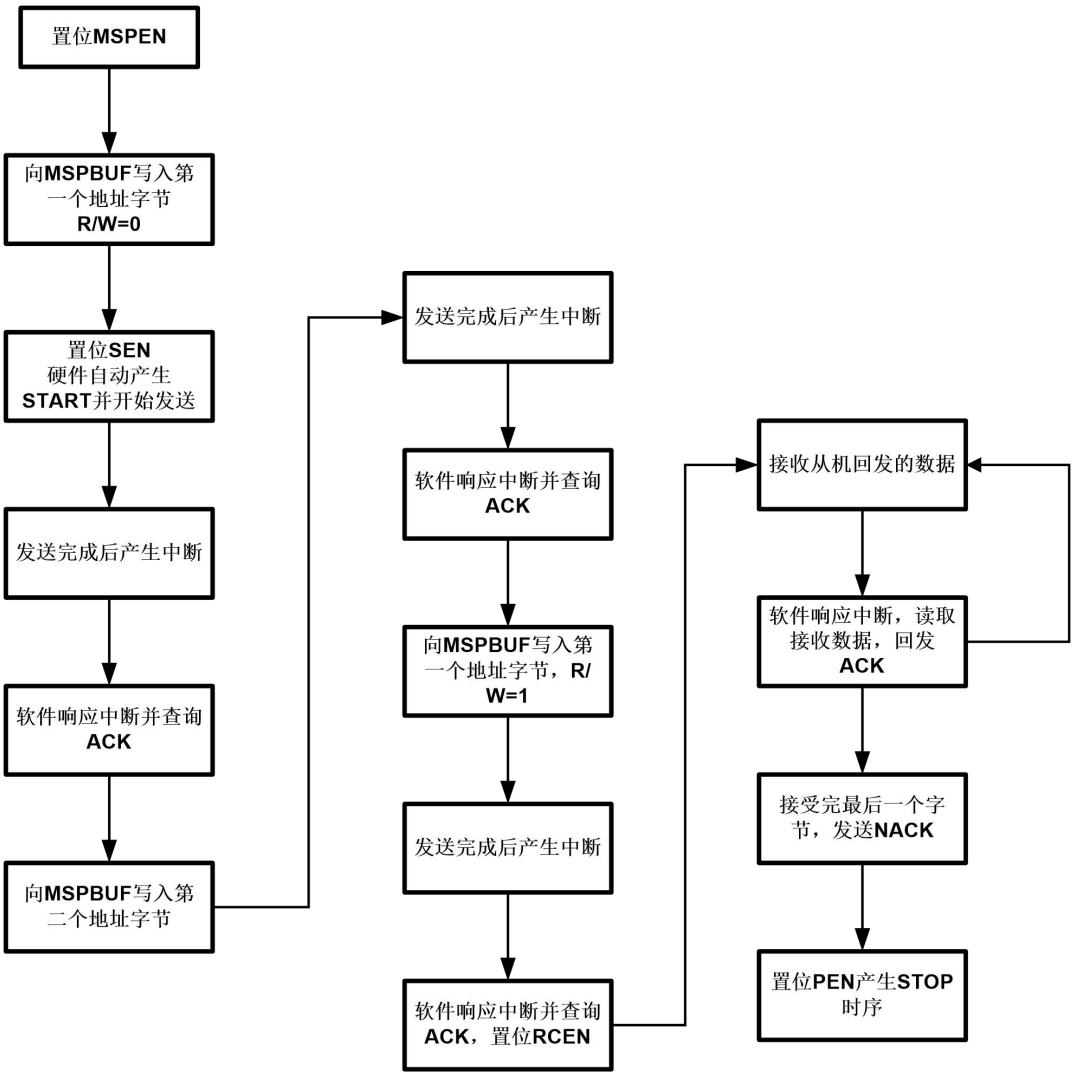


图 20-16 I²C 软件发送数据流图

双向数据传输（组合模式）

典型的双向数据读写流图如下图所示。在主机发送或读取数据过程中，主机可以通过发送 Repeated Start 时序来重新启动一次新的发送或读取通信，所以主机在一次通信中，即可以有数据发送也可以

有数据读取。

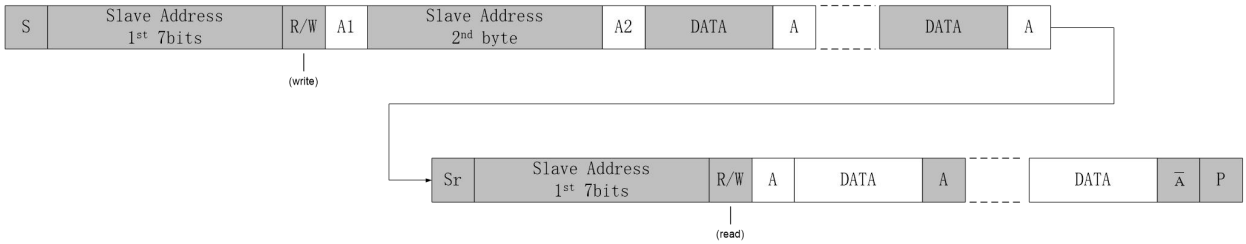


图 20-17 I²C 软件发送数据流图

组合传输的软件操作流程与单向传输类似，只是在某个字节收发完成后，通过发送RESTART时序和1st从机地址字节来修改传输方向。

20.8.3 SCL 延展（Slave Clock Stretching）

I²C 总线运行低速从机通过拉低 SCL 的方式通知主机暂停数据通信。I²C 主机必须支持这一特性，因此在每个字节收发起始位置处，主机在尝试发送 SCL 高电平后，需要自动检查总线上 SCL 的实际电平，如果不是高电平，意味着从机正在进行 SCL 延展，主机会持续监控 SCL 电平，直到 SCL 为高，才开始后续操作。

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。

20.8.4 超时机制

I²C主机还实现了超时机制，当主机检测到SCL延展，其内部定即发现从机长时间拉低SCL导致总线无法通信的情况下，产生超时报警中断并返回IDLE状态。

时器开始计时，主机设定的SCL延展超时的时长最长是4096个SCL周期，假设波特率为100K，则超时周期大约是40ms，如果波特率是400K，则超时周期大约是10ms。

通过12bit的TIMEOUT寄存器，软件可以设置超时周期。软件必须在MSPEN为0的情况下设置TIMEOUT寄存器，此寄存复位值为0xFFF，即表示最长4096*T_{SCL}的超时周期，当检测到SCL延展后，TIMEOUT寄存器开始向下递减，当计数到0时，计数停止，TIMEOUT寄存器被复位到0xFFF，同时触发超时中断。因此通过修改TIMEOUT初始值，可以设定超时周期。

$T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

当发生TIMEOUT中断时，建议软件复位I²C模块。

此功能可以被关闭，如果关闭硬件超时，软件也可以通过定时器结合SCL引脚状态判断来自行实现任意长度的超时判决。

20.8.5 可编程时序

I²C 模块的主机模式提供了灵活的时序编程特性，允许用户定义 SCL 时钟的低电平宽度、高电平宽度，SDA 数据的建立和保持时间。

通过 MSPBRG 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立保持时间长度。此功能对于发送数据和发送 ACK 都有效。

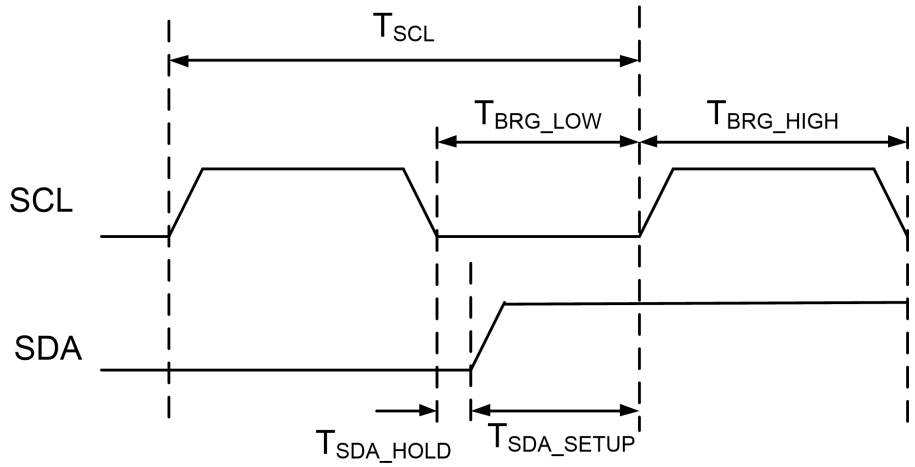


图 20-18 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG\_LOW} + T_{BRG\_HIGH}$

$T_{SDA\_SETUP} = T_{BRG\_LOW} - T_{SDA\_HOLD}$

注意，应用中对MSPBGRH、MSPBRGL和SDAHD寄存器的配置必须满足以下要求，如果违反这些要求将导致异常的总线时序：

MSPBRGH >= 2

MSPBRGL >= 2

MSPBRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

20.9 寄存器

I2C0模块基地址：0x40015000

I2C1模块基地址：0x40015400

I2C2模块基地址：0x40015800

I2C3模块基地址：0x40015C00

offset 地址	名称	符号
0x00	I2C 主机配置寄存器（I2C Master Config Register）	I2C_MSPCFGR
0x04	I2C 主机控制寄存器（I2C Master Control Register）	I2C_MSPCR
0x08	I2C 主机中断使能寄存器（ I2C Master Intterupt Enable Register）	I2C_MSPIER
0x0C	I2C 主机中断标志寄存器（ I2C Master Interrupt Status Register）	I2C_MSPISR
0x10	I2C 主机状态寄存器（I2C Master Status Register）	I2C_MSPSR
0x14	I2C 主机波特率寄存器（I2C Master Baud rate Generator Register）	I2C_MSPBGR
0x18	I2C 主机收发缓存寄存器（I2C Master transfer Buffer）	I2C_MSPBUF
0x1C	I2C 主机时序控制寄存器（ I2C Master Timing Control Register）	I2C_MSPTCR
0x20	I2C 主机超时寄存器（I2C Master Time-Out Register）	I2C_MSPTOR

20.9.1 I2C 主机配置寄存器（I2C_MSPCFGR）

名称	I2C_MSPCFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						AUTOEND	MSP_DMAEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TOEN	MSPEN
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:15	--	RFU：未实现，读为 0
17	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
16	MSP_DMAEN	主机 DMA 使能 (Master DMA enable) 0：关闭 DMA 功能 1：使能 DMA 功能
15:2	-	RFU：未实现，读为 0
1	TOEN	SCL 拉低超时使能 (Time Out enable) 1：使能超时功能，超时周期由 MSPTOR 寄存器定义 0：关闭超时功能
0	MSPEN	I2C 主机模块使能控制位 (Master enable) 1：I2C 主机使能 0：I2C 主机禁止

20.9.2 I2C 主机控制寄存器（I2C_MSPCR）

名称	I2C_MSPCR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RCEN	PEN	RSEN	SEN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3	RCEN	主机接收模式下，接收使能位 (Receive enable) 1：主机接收使能 0：接收禁止 主机通信中，软件在发送完地址字节后，通过置位 RCEN 将传输方向切换为主机接收，然后可以接收来自于从机的数据。RCNE 在接收过程中保持为 1，直到软件置位 PEN 发送 STOP 时序。
2	PEN	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RSEN	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start Enable)
0	SEN	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

20.9.3 I2C 主机中断使能寄存器（I2C_MSPIER）

名称	I2C_MSPIER							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOLE	OVTE	SE	PE	NACKE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOLE	WCOL 中断使能寄存器 (Write collision interrupt enable) 1：允许写冲突中断 0：禁止写冲突中断
5	OVTE	SCL 超时中断使能寄存器 (SCL overtime enable) 1：允许超时中断 0：禁止超时中断
4	SE	START 时序中断使能寄存器 (Start interrupt enable) 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	STOP 时序中断使能寄存器 (Stop interrupt enable) 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
1	TXIE	I2C 主机发送完成中断使能(Trasnmit done interrupt enable) 1：允许发送完成中断 0：禁止发送完成中断
0	RXIE	I2C 主机接收完成中断使能(Receive done interrupt enable) 1：允许接收完成中断 0：禁止接收完成中断

20.9.4 I2C 主机中断标志寄存器（I2C_MSPISR）

名称	I2C_MSPISR							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	WCOL	OVT	S	P	ACKSTA	TXIF	RXIF
位权限	U-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOL	写冲突检测位，MCU 只能在完成 START 时序或发送完成一帧读写之后才能写 MSPBUF，否则发生写冲突；硬件置位，软件写 1 清零(Write Collision Interrupt Flag,write 1 to clear) 1：发送写冲突 0：未发生冲突
5	OVT	SCL 超时中断标志，仅在 TOEN 为 1 时工作(SCL OverTime Interrupt Flag) 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	START 时序发送完成中断标志，硬件置位，软件读取后清零(Start Interrupt flag)
3	P	STOP 时序发送完成中断标志，硬件置位，软件读取后清零(Stop interrupt flag)
2	ACKSTA	主控发送模式下，来自从机的回应信号；当主机发送后收到 NACK，此标志可以产生中断；硬件置位，软件写 1 清零。(Acknowledge Status Flag ,write 1 to clear) 1：从机回应 NACK 0：从机回应 ACK
1	TXIF	I2C 主机发送完成中断标志，硬件置位，软件写 1 清零(Trasnmit done interrupt flag,write 1 to clear) 此标志寄存器在主机接收完从机回发的 ACK 或 NACK 后置位。
0	RXIF	I2C 主机接收完成中断标志，硬件置位，软件写 1 清零(Receive done interrupt flag,write 1 to clear) 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

20.9.5 I2C 主机状态寄存器（I2C_MSPSR）

名称	I2C_MSPSR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	BUSY	RW	-	BF	-	ACKMO
位权限	U-0	R-0	R-0	U-0	R-0	U-0	R/W-0

位号	位名	说明
31:6	--	RFU：未实现，读为 0
5	BUSY	I2C 通信状态位 (Busy) 1：接口处于读写状态，正在进行数据传输， 0：已完成数据传输
4	RW	I2C 传输方向状态位 (Read/Write) 1：主机从从机读取数据 0：主机向从机写入数据
3	-	RFU：未实现，读为 0
2	BF	缓冲器满状态位 (Buffer Full) 接收： 1：接收完成，MSPBUF 满 0：接收未完成，MSPBUF 空 发送： 1：正在发送，MSPBUF 满 0：发送完成，MSPBUF 空
1	-	RFU：未实现，读为 0
0	ACKMO	主控接收模式下，主机回应信号的状态 (Ack Master output) 1：主机回发 NACK 0：主机回发 ACK ***注意：必须在 P 标志寄存器被清零的情况下，软件才能置位 ACKMO***

20.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR）

名称	I2C_MSPBGR							
Offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							MSPBRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSPBRGH[7:0]							
位权限	R/W-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MSPBRGL[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBRGL[7:0]							
位权限	R/W-00010011							

位号	位名	说明
31:25	--	RFU：未实现，读为 0
24:16	MSPBRGH	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 (Master SCL High level length)

15:9	-	RFU：未实现，读为 0
8:0	MSPBRGL	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 (Master SCL Low level length)

20.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF）

名称	I2C_MSPBUF							
Offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SSPBUF							
位权限	R/W-00000000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	MSPBUF	MSPBUF[7:0]：数据的读写通过对 MSPBUF 的操作完成。发送时，对 MSPBUF 执行写操作，同时也载入数据收发移位寄存器；接收时，MSPBUF 与数据收发移位寄存器组成双缓冲结构，读出数据为 MSPBUF 的数据。接收完一个字节的数据，数据收发移位寄存器将数据载入 MSPBUF，同时置位 RXIF。数据收发移位寄存器不是直接寄存器，没有物理地址。 (Master data Buffer)

20.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR）

名称	I2C_MSPTCR							
Offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SDAHD[7:0]							
位权限	R/W-00001010							

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2C 工作时钟计数 (SDA hold delay) 注意：最小有效值为 1，最大有效值为 MSPBRGL

20.9.9 I2C 主机超时寄存器（I2C_MSPTOR）

名称	I2C_MSPTOR							
Offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					TIMEOUT[11:8]			
位权限					R/W-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUT[7:0]							
位权限	R/W-1111 1111							

位号	位名	说明
31:12	--	RFU：未实现，读为 0
11:0	TIMEOUT	定义从机 SCL 低电平延展超时周期，软件可以在 MSPEN=0 的情况下改写 (SCL stretching Time Out) $T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

21 通用异步收发传输器（UART）

21.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 6路独立通道（UART0, UART1,UART2, UART3, UART4, UART5）
- 全双工通信口
- 支持半双工单线通信
- 支持RX-TX端口交换
- UART具有数据接收完成/接收错误中断，并提示错误类型
- 可配置数据长度，支持6、7、8、9bits
- 可配置的停止位-支持1个停止位或2个停止位
- 可配置为红外调制输出功能，且载波频率可设置，及载波占空比可设置
- 支持DMA
- 支持接收超时机制(UART0~UART5)
- 支持发送延迟功能(UART0~UART5)
- 支持低功耗休眠唤醒(UART0、UART1)

UART2~5支持smart-card模式（ISO7816-3），包括以下特性：

- 具备卡时钟输出端口，输出频率在1MHz~5MHz之间可设
- 位传输方向可配置，支持MSB First或LSB First
- 错误信号宽度可配置为1/1.5/2个ETU
- 发送数据支持传输错误重发机制，重发次数可配置为0~3次
- 支持EGT可设0~256，并支持多种超时中断
- 具有数据接收完成/接收错误中断，并提示错误类型
- 发送中断产生条件可配置为缓冲区空或移位寄存器空
- 支持DMA接口

21.2 结构框图

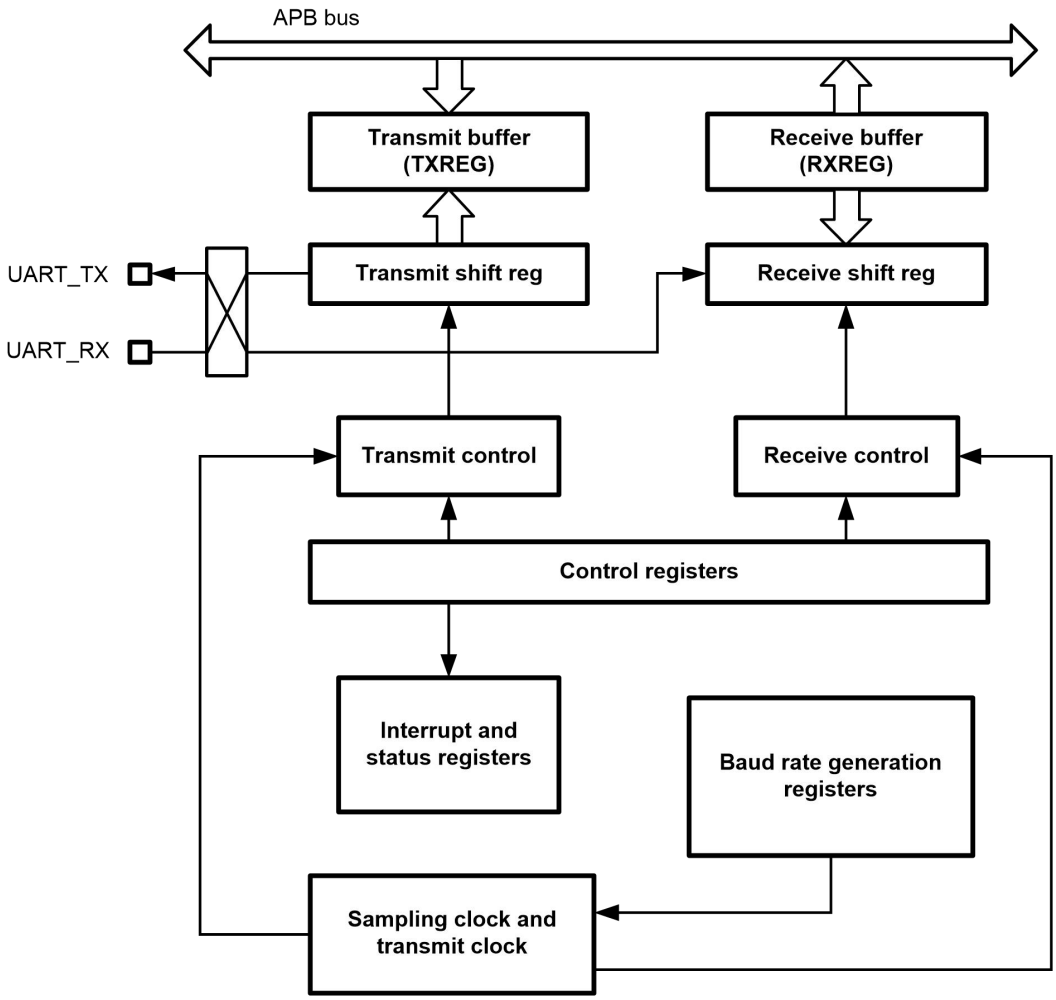


图 21-1 UART 接口时序

21.3 引脚定义

UART 模块使用 2 个引脚与外部器件通信，每个 UART 的收发信号可能被映射到不同的 GPIO 上。

下表为 FM33FR0 的引脚映射关系

引脚			UARTx	符号
PB10	PB12	-	UART0	UART0_RX
PB9	PB13	-		UART0_TX
PC2	PA5	-	UART1	UART1_RX
PC1	PA6	-		UART1_TX
PC4	PD8	-	UART2	UART2_RX
PC5	PD13	-		UART2_TX
PA3	PB0	-	UART3	UART3_RX
PA4	PB1	-		UART3_TX
PB2	PD11	-	UART4	UART4_RX
PB3	PD12	-		UART4_TX
PC7	PD0	PA11	UART5	UART5_RX
PC6	PD1	PA12		UART5_TX

表 21-1 UART 引脚列表

智能卡模式下，UART_RX 复用为 7816 时钟输出，UART_TX 复用为 7816 数据信号线。

21.4 UART 类型区分

芯片集成了多种不同类型的UART，其差异如下表所示：

UART 特性	UART0/1	UART2/3/4/5
DMA 支持	Y	Y
半双工/全双工	Y	Y
红外发射	Y	Y
双时钟域（工作时钟独立于总线）	Y	-
半双工单线通信	Y	Y
休眠唤醒	Y	-
接收超时	Y	Y
发送延迟	Y	Y
数据长度	6、7、8、9bits	
智能卡模式	N	Y

表 21-2 UART 类型列表

21.5 UART 字符描述

21.5.1 UART 模式

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

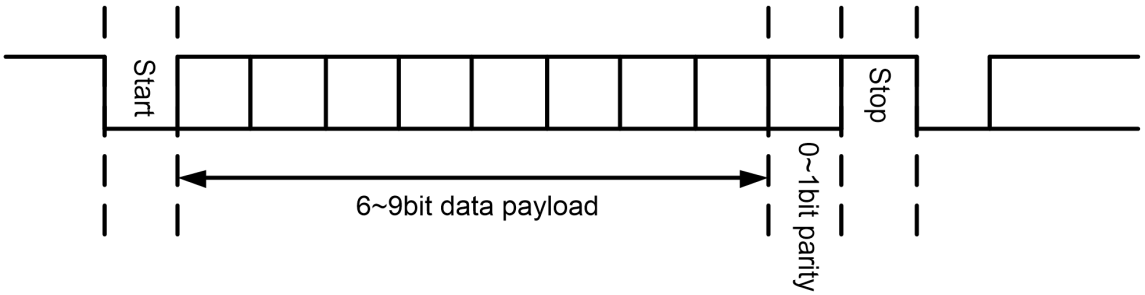


图 21-2 UART 字符描述

UART 支持多种帧格式，由 UARTxCSR.PDSEL 寄存器和 UARTxCSR.PARITY 寄存器控制。见下表：

PDSEL	PARITY	帧格式[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

表 21-3 UART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

21.5.2 智能卡模式

ISO7816-3协议规定的字符格式如下图。

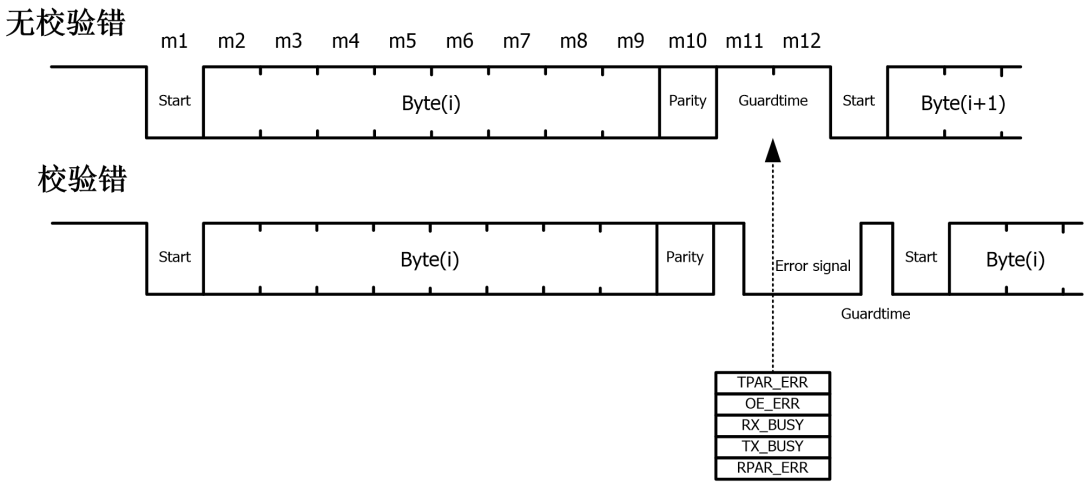


图 21-3 7816-3 帧结构图

参照 7816 协议标准，7816 基本接口时序如下：

- 一个起始位后跟8个数据位及1个校验位，以1-ETU或2-ETU的GUARDTIME结束。
- 单字节数据长度最小为11-ETU或12-ETU。
- 第10.5个ETU接收电路校验接收数据，若校验正确，则插入2个ETU的GUARDTIME，确保数据长度为12-ETU，并在第11个ETU时令RXBUSY无效并产生可能的OVERR标志，完成数据接收；若接收校验出错，则在第10.5-ETU拉低IO，产生ERROR SIGNAL。ERROR SIGNAL最短1个ETU，最长2个ETU。并在第11个ETU根据需要产生RPARERR标志。
- 第11个ETU时发送电路未采样到ERROR SIGNAL，则说明发送数据正确，数据发送完成，令TX_BUSY无效。
- 若第11个ETU发送电路采样到ERROR SIGNAL，则说明发送数据错误，根据设定产生需要的TPAR_ERR或等待2个ETU后重发数据。

UART模块用于智能卡通信时，必须设置数据长度为8bit，并且带校验位。

21.6 功能描述

21.6.1 时钟结构

UART0 和 UART1 采用了多时钟结构：

- 总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 UART 内部寄存器时，必须使能 UARTx_PCE。
- 数据收发时钟用 UCLK 表示，除了可以来源于 APBCLK，还可以来源于 RCHF、SYSCLK、XTHF，能够独立于 APBCLK 工作。必须使能 UCLK 才能进行数据收发。

PCLK 和 UCLK 的控制都在 CMU 模块内完成，进行 UART 通信前必须正确配置相应的 CMU 控制寄存器。

采用双时钟结构，可以使 UART0 和 UART1 的工作不受限于 APBCLK 的配置，当某些外设需要工作在很高的 APBCLK 频率上时，UART 仍可以工作在降低的频率上；或者反过来，CPU 工作在较低的频率上，也不影响 UART 以较高的波特率进行数据通信。

理论上 PCLK 和 UCLK 之间没有相对关系的约束，UCLK 可以快于或者慢于 PCLK。但是应用需要注意当两者频率相差较大时，CPU 或者 DMA 是否来得及进行数据搬运。

与 UART0 和 UART1 不同的是，UART2~5 采用单时钟结构，此时 UCLK=PCLK，UART 的数据收发时钟也是来源于 APBCLK 的。

21.6.2 位接收采样

UART 对接收数据进行波特率的 8 倍或 16 倍过采样，并在每个 bit 的中间位置进行三中取二的多数判决，以提高对信号噪声的抑制能力。

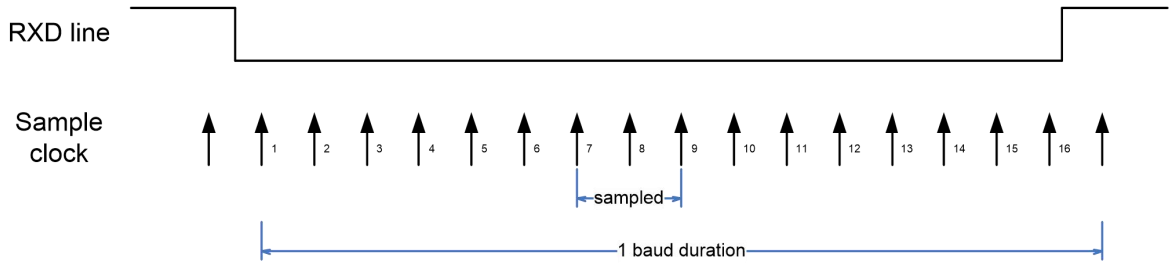


图 21-4 位接收 16 倍采样

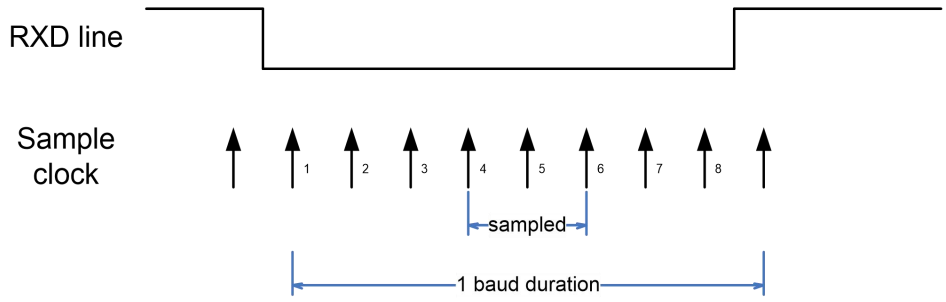


图 21-5 位接收 8 倍采样

接收移位寄存器收到的 bit 位是多数判决的结果。例如三次采样结果是 001，则判决为 0；如果是 011，则判决为 1。

通过 OVSM 寄存器可以配置 UART 接收数据时的过采样倍数。如果 UART 对输入信号进行 16 倍过采样，要求 SPBRG 配置不能小于 16，即 UART 工作时钟必须至少是波特率的 16 倍。如果 UART 对输入信号进行 8 倍过采样，要求 SPBRG 配置不能小于 8，即 UART 工作时钟必须至少是波特率的 8 倍。

选择较小的过采样倍数时，可以获得较高的通信波特率。

21.6.3 数据发送

在发送模式下，UART 的串行数据发送电路主要包括一个发送移位寄存器(TSR)，TSR 功能是将数据逐个移位送出。待发数据必须先写到发送缓冲区中。当软件置位 TXEN 寄存器后，如果发送缓冲区非空，UART 将缓冲区数据载入 TSR 并开始移位输出。

注：由于寄存器操作时钟和波特率时钟是异步关系，当发送开始时，需要等待波特率时钟到来，因此从 TXEN 置位到 UART 开始发送 Start 位之间，有最大 1 个 baud 的延迟。

TXBE 和 TXSE 是发送中断标志位，分别表示发送缓冲区空和 TSR 空，软件可以选择在合适的时间点产生发送完成中断。

一般情况下，一开始 TSR 寄存器是空的，数据的发送需先设定波特率 SPBRG，使能发送模块(设定 TXEN 为 1)，然后写入 TXBUF 寄存器开始发送。也可以在设定好波特率 SPBRG 后，先写入 TXBUF 寄存器，然后再设定 TXEN 使能发送模块来开始数据发送。如果在数据发送过程中将发送模块使能位 TXEN 清 0，那么数据发送工作就会被中断，发送模块也会被复位。

下图为UART异步发送的例子。这个示例中软件首先向TXBUF写入数据，然后通过置位TXEN启动发送。

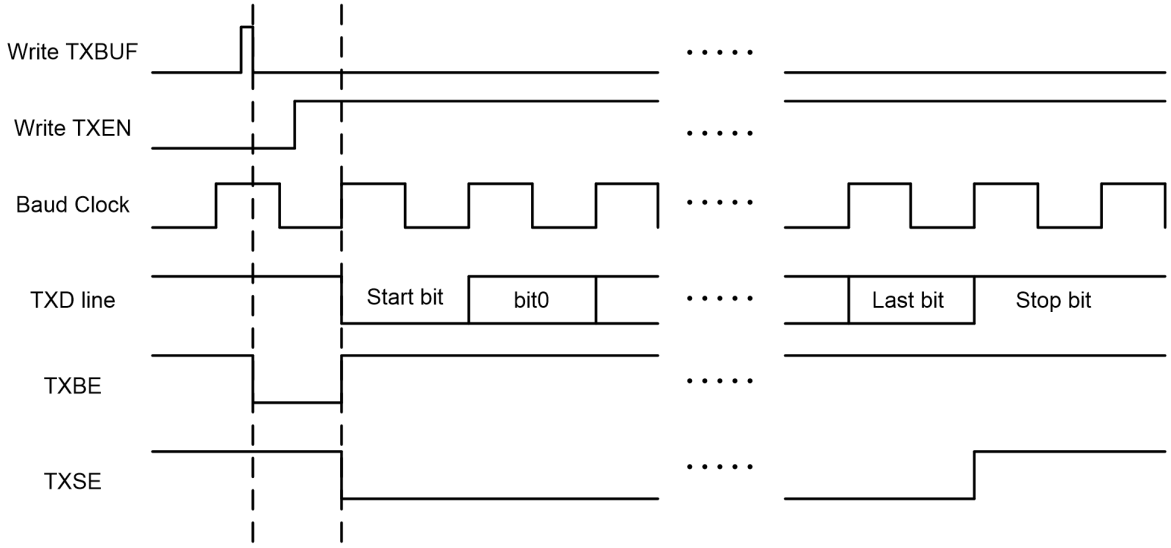


图 21-错误!未定义书签。 UART 异步发送波形 1

上图中推荐的操作步骤如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 TXSE_IE 或者 TXBE_IE
- 决定数据发送的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 如果希望发送的串行数据红外调制，向 IRCR 寄存器写入合适的值来获得相应的调制频率和占空比，并置位 TXIREN
- 将待发送的数据写入 TXBUF 寄存器（自动启动发送）
- 使能发送模块：置位 TXEN

软件也可以先置位TXEN再写入TXBUF，此时UART会在数据写入TXBUF后立刻开始发送流程。

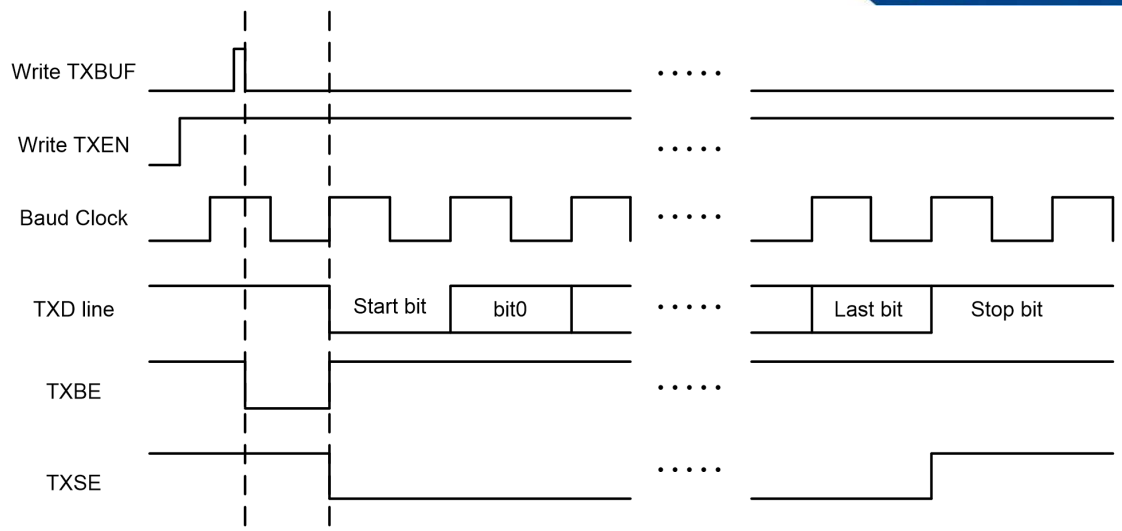


图 21-6 UART 异步发送波形 2

当TXBUF为空时，软件可以立即写入下一个待发送数据，以实现连续无间隔的数据发送。

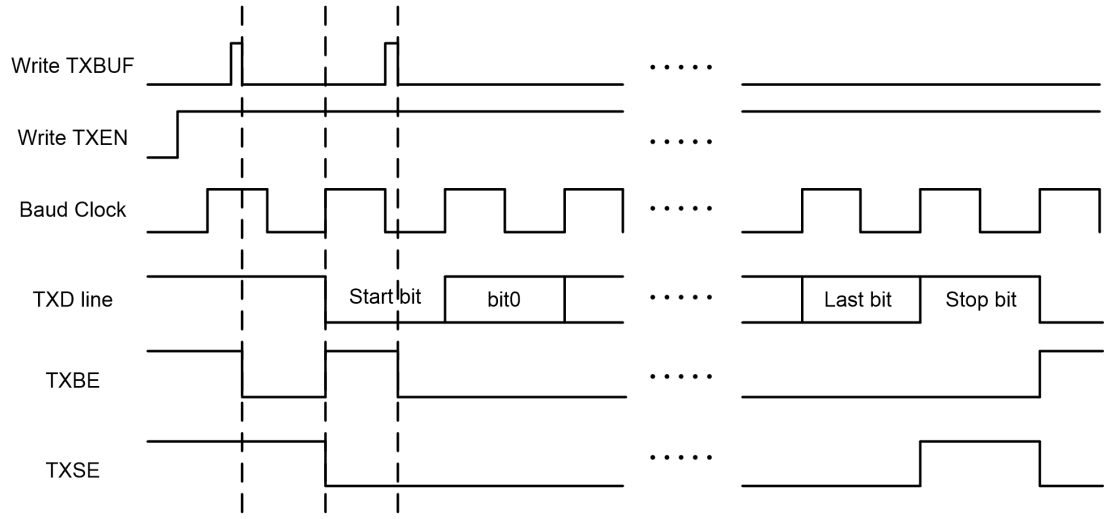


图 21-7 UART 异步发送波形 3

智能卡模式

在智能卡模式下，UART完成数据和校验位发送后，会检测从机是否回发错误信号。如果没有检测到错误信号，则发送停止位结束通信；如果检测到错误信号，则启动自动重发流程。当自动重发超过寄存器设定的次数，但是仍接收到错误信号，则UART放弃重发并置位发送错误中断；如果自动重发后没有检测到错误信号，则发送停止位结束通信。

在发送正常结束后，UART根据寄存器设置等待额外保护时间，并在保护时间结束后才置位发送完成中断。

21.6.4 数据接收

UART 的串行数据接收电路主要包括一个接收移位寄存器(RSR)。当接收到停止位后，RSR 就把接收到的数据送入接收缓冲区(RXBUF)，传送完成后，在每次接收数据送入接收缓冲区后将中断标志 RXBF 置 1。当接收缓冲区已满时，RSR 接收到一帧数据后仍会将其写入接收缓冲区，即覆盖缓冲区中原有数据，并且再次置位 RXBF，同时发生接收溢出错误，OERR 被置 1；软件写 1 或者读取 RXBUF 都可以清除 OERR 标志。

接收过程中，如果没有检测到正确的停止位，则发生帧格式错，FERR 被置 1；如果发生奇偶校验错，标志位 PERR 被置 1。

推荐的异步接收操作如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 RXBF_IE
- 设置数据接收的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 使能接收模块：置位 RXEN
- 在一帧接收完毕时，RXBF 位会置 1，如果 RXBF_IE 位为 1，将会产生中断
- 读取 PERR、FERR、OERR 寄存器，判断是否有数据错误或者溢出
- 读取 RXBUF 寄存器中的接收数据

智能卡模式

智能卡模式下，UART完成数据和校验位接收后，自动判断校验位是否正确。如果校验位正确，则判决停止位，并在正确的停止位后置位接收完成中断，或在错误的停止位置位帧格式错误中断。如果校验位错误，则自动回发错误信号并等待对方重发；如果重发次数达到了寄存器设定的上限，仍没有接收到正确的帧，则停止回发错误信号，并置位接收错误中断。

21.6.5 低功耗休眠唤醒（UART0/1）

UART0和UART1支持RXD下降沿触发的芯片休眠唤醒。当置位了NEWUP寄存器之后，RXD输入上的下降沿事件（低电平持续时间>100ns）将会使芯片从休眠模式下唤醒，借助这个功能，可以实现UART0/1在休眠模式下接收数据。

软件配置方法如下：

- 配置UART寄存器，使能NEWUP，使能NEWUP_IE
- 根据需要配置波特率分频寄存器

- 将对应GPIO配置为UART数据接收功能
- 置位RXEN，使能接收
- 软件设置芯片进入休眠，等待UART接收事件

21.6.6 使用 DMA 进行 UART 收发

当 UART 模块被使能后，UART 模块在发送缓冲寄存器空和接收缓冲寄存器满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 UART 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 UART 请求，并完成 RAM 和 UART 之间的数据搬运。

应用举例：使用 DMA 进行 UART0 接收

- 将 DMA 通道 1 或 3 配置为 RXD0
- 设置对应通道参数：RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 UART 模块参数
- 使能 UART 模块接收使能，等待数据接收
- 收到数据后 UART 自动产生 DMA 请求
- DMA 响应请求，读取 UART 接收缓存寄存器，写入指定 RAM 地址

21.6.7 DMA 模式下的发送完成中断

当 UART 通过 DMA 进行数据发送时，DMA 会在指定长度的数据传输完成后产生 DMA 通道中断。但是当通道中断产生时，最后一帧数据刚刚被写入 UART 发送缓冲区，还未被发送出去。

通过配置 DMATXIFCFG 寄存器，可以实现 DMA 传输完成、并且最后一帧数据发送完成的情况下，产生一个发送完成中断（缓冲区空或者移位寄存器空），以便实现所有数据全部发送出去后，再中断 CPU 的应用场景。

软件工作流程说明如下：

- 配置DMA通道为UART发送
- 关闭DMA通道中断使能
- 置位DMATXIFCFG寄存器，仅允许最后一帧数据产生中断输出
- 准备待发送数据，使能DMA
- 置位UART TXBE_IE或TXSE_IE寄存器，允许中断产生
- UART连续发送，直到最后一帧，发送期间不会产生TXBE或TXSE中断

- 最后一帧发送完成后，UART产生TXBE或TXSE中断

下表假设 UART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
0	x	1~N	每帧发送完成后置位	不产生
1	0	1~N	每帧发送完成后置位	产生
	1	1~N-1	每帧发送完成后不置位	不产生
		N	每帧发送完成后置位	产生

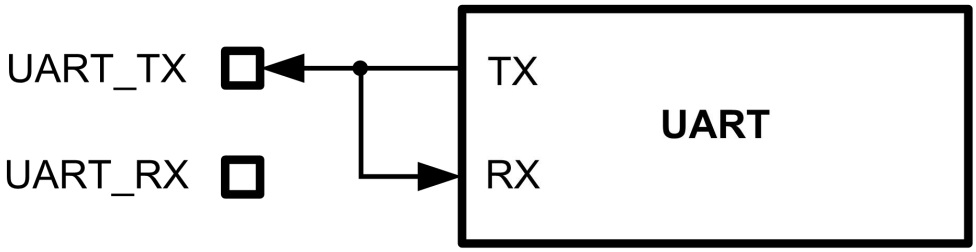
表 21-4 DMA 发送中断

21.6.8 半双工单线通信

当UARTMD寄存器置位时，UART以半双工模式工作，此时仅需一根信号线即可完成通讯。半双工模式仅在UART功能时可以使用，禁止在智能卡模式下使用，即置位UARTMD时必须保证MODE=0。

当UARTMD置位后：

- TX和RX信号在内部连接
- RX引脚不再需要使用，可以用于GPIO或者其他功能
- 当UART不发送数据时，TX引脚输出使能自动关闭，输入使能自动打开，引脚保持浮空状态，需要通过上拉电阻赋予确定电平
- TX引脚可以配置为开漏输出，数据冲突必须由软件处理
- 数据发送时UART自动屏蔽数据接收



21.7 波特率发生

21.7.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

UART 波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 UART4 和 UART5，F_{CLK} 就是 APBCLK；对于 UART0 和 UART1，F_{CLK} 是独立与 APBCLK 的工作时钟。

为了支持全双工通信，接收和发送波特率单独产生；

下表是常用系统时钟频率下的波特率：

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300	0	80000	299.9991	-0.00031	106667
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

表 21-5 常用时钟频率下波特率计算

智能卡模式下，通信波特率由 UARTx_CODR 和 UARTx_BRG 寄存器共同决定。

UARTx_CODR 用于对 APBCLK 预分频得到 7816 接口时钟，并且这个时钟可以输出给外部卡片。

$F_{7816}=F_{APBCLK}/(CLKDIV+1)$

7816 通信波特率使用 7816 接口时钟产生：

$Baud = F_{7816}/(SPBRG + 1)$

智能卡模式下SPBRG最小可用值是0x1，禁止配置0x0

21.7.1 波特率自适应

利用 Timer 的 Capture 功能，可以实现波特率自适应功能。可实现的一种方法为，外部 UART 设备按约定的数据内容(比如 0xF8)发送一帧，由 Timer 对该帧数据的高电平脉宽进行计数，MCU 读取 Timer 捕捉结果计算得到波特率因子，并写入波特率发生寄存器中，作为波特率发生的时钟分频计数值 X 使用。这时接收状态复位，重新等待起始位，以写入的波特率因子所产生的波特率接收数据。参考 Timer 章节。

21.8 红外调制

TZBRG 寄存器保存一个 11 位的分频系数 X，其值为 0~2047 之间的任一整数。所有 UART 共用一个红外调制频率发生器。

红外调制频率计算公式：

$FIR = F_{APBCLK} / (TZBRG+ 1)$

红外调制的方式为：发送数据 0 时调制红外频率，发送数据 1 时不调制。为满足 PNP 和 NPN 两种红外驱动管的需求，寄存器 IRFLAG 位控制红外调制输出的极性。IRFLAG=0 时为正极性输出，适合 PNP 管驱动；IRFLAG=1 时为负极性输出，适合 NPN 管驱动。

TH 寄存器用于配置红外调制占空比

占空比：Y = (TZBRG[10:4] * TH) /(TZBRG+1)

当 TH=4’b0000 时，占空比为 Y = (TZBRG[10:1]+1)/(X+1)；

当 TZBRG[10:4]=7’h00 时，占空比为 Y = TH/(TZBRG[3:0]+ 1)；若此时 TH>TZBRG [3:0]，则红外调制时钟 IRCLK 为固定高电平。

当红外调制极性反向时（IRFLAG=1），占空比也为 1-Y

红外调制波形见下图：

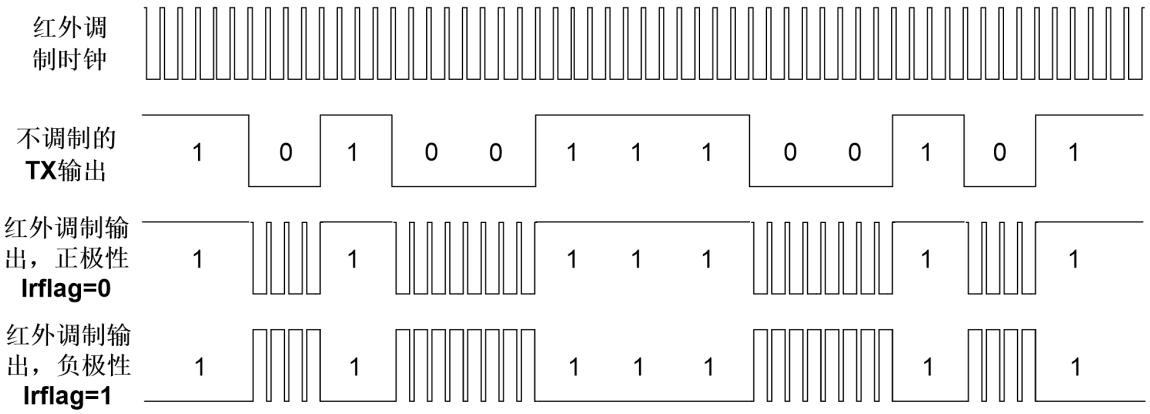


图 21-8 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

21.9 接收超时

针对 MODBUS 等时间敏感型应用，设计了接超时机制。当使能 RXTOEN 寄存器后，超时计数器以波特率时钟计数，当每次收到一个完整的数据帧，将清零超时计数器并重新开始计数。超时溢出的上限值可以由软件配置，最大 255 波特。

21.10 发送延迟

通过 TXDLY_LEN 寄存器，可以控制两个数据帧发送之间的间隔时间，单位是波特。发送延迟是从上一帧最后一个 STOP 位结束，到下一帧起始位之间的间隔。

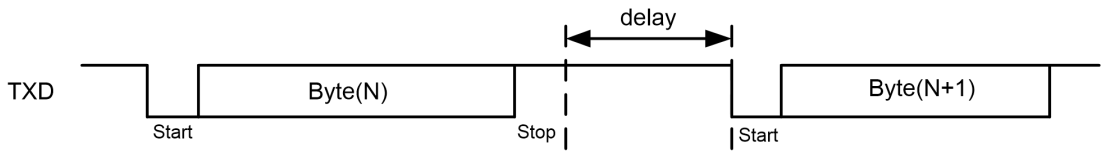


图 21-9 UART 发送延迟

21.11 寄存器

UART0模块基地址：0x4001 1C00

UART1模块基地址：0x4001 2000

UART2模块基地址：0x4001 7400

UART3模块基地址：0x40017800

UART4模块基地址：0x4001 A000

UART5模块基地址：0x4001 A400

UARTIR模块基地址：0x40017C00

offset 地址	名称	符号
UARTIR 寄存器		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
UART0 寄存器		
0x00	UART0 控制状态寄存器 （UART0 Control Status Register）	UART0_CSR
0x04	UART0 中断使能寄存器 （UART0 Interrupt Enable Register）	UART0_IER
0x08	UART0 中断标志寄存器 （UART0 Interrupt Status Register）	UART0_ISR
0x0C	UART0 超时和延迟寄存器 （UART0 Time-Out and Delay Register）	UART0_TODR
0x10	UART0 接收缓冲寄存器 （UART0 Receive Buffer）	UART0_RXBUF
0x14	UART0 发送缓冲寄存器 （UART0 Transmit Buffer）	UART0_TXBUF
0x18	UATR0 波特率产生寄存器 （UART0 Baud rate Generator Register）	UART0_BGR
0x20	UART0 模式寄存器 （UART0 Mode Control Register）	UART0_MCR
0x24	-	
0x28	-	
UART1 寄存器		
0x00	UART1 控制状态寄存器 （UART1 Control Status Register）	UART1_CSR
0x04	UART1 中断使能寄存器 （UART1 Interrupt Enable Register）	UART1_IER
0x08	UART1 中断标志寄存器 （UART1 Interrupt Status Register）	UART1_ISR
0x0C	UART1 超时和延迟寄存器 （UART1 Time-Out and Delay Register）	UART1_TODR
0x10	UART1 接收缓冲寄存器 （UART1 Receive Buffer）	UART1_RXBUF
0x14	UART1 发送缓冲寄存器	UART1_TXBUF

offset 地址	名称	符号
	（UART1 Transmit Buffer）	
0x18	UATR1 波特率产生寄存器 （UART1 Baud rate Generator Register）	UART1_BGR
0x20	UART1 模式寄存器 （UART1 Mode Control Register）	UART1_MCR
0x24	-	
0x28	-	
UART2 寄存器		
0x00	UART2 控制状态寄存器 （UART2 Control Status Register）	UART2_CSR
0x04	UART2 中断使能寄存器 （UART2 Interrupt Enable Register）	UART2_IER
0x08	UART2 中断标志寄存器 （UART2 Interrupt Status Register）	UART2_ISR
0x0C	UART2 超时和延迟寄存器 （UART2 Time-Out and Delay Register）	UART2_TODR
0x10	UART2 接收缓冲寄存器 （UART2 Receive Buffer）	UART2_RXBUF
0x14	UART2 发送缓冲寄存器 （UART2Transmit Buffer）	UART2_TXBUF
0x18	UATR2 波特率产生寄存器 （UART2 Baud rate Generator Register）	UART2_BGR
0x20	UART2 模式寄存器 （UART2 Mode Control Register）	UART2_MCR
0x40	UART2 智能卡状态标志寄存器 （UART2 Smart Card Interrupt Status Register）	UART2_SCISR
0x44	UART2 智能卡中断使能寄存器 （UART2 Smart Card Interrupt Enable Register）	UART2_SCIER
0x48	UART2 智能卡帧格式控制寄存器 （UART2 Smart Card Frame Format Control Register）	UART2_FFCR
0x4C	UART2 智能卡额外保护时间寄存器 （UART2 Smart Card Extra Guard Time Register）	UART2_EGTR
0x50	UART2 智能卡输出时钟分频寄存器 （UART2 Smart Card Clock Output Register）	UART2_CODR
UART3 寄存器		
0x00	UART3 控制状态寄存器 （UART3 Control Status Register）	UART3_CSR
0x04	UART3 中断使能寄存器 （UART3 Interrupt Enable Register）	UART3_IER
0x08	UART3 中断标志寄存器 （UART3 Interrupt Status Register）	UART3_ISR
0x10	UART3 接收缓冲寄存器 （UART3 Receive Buffer）	UART3_RXBUF
0x14	UART3 发送缓冲寄存器 （UART3Transmit Buffer）	UART3_TXBUF
0x18	UATR3 波特率产生寄存器 （UART3 Baud rate Generator Register）	UART3_BGR
0x20	UART3 模式寄存器 （UART3 Mode Control Register）	UART3_MCR
0x40	UART3 智能卡状态标志寄存器 （UART3 Smart Card Interrupt Status Register）	UART3_SCISR

offset 地址	名称	符号
0x44	UART3 智能卡中断使能寄存器 （UART3 Smart Card Interrupt Enable Register）	UART3_SCIER
0x48	UART3 智能卡帧格式控制寄存器 （UART3 Smart Card Frame Format Control Register）	UART3_FFCR
0x4C	UART3 智能卡额外保护时间寄存器 （UART3 Smart Card Extra Guard Time Register）	UART3_EGTR
0x50	UART3 智能卡输出时钟分频寄存器 （UART3 Smart Card Clock Output Register）	UART3_CODR
UART4 寄存器		
0x00	UART4 控制状态寄存器 （UART4 Control Status Register）	UART4_CSR
0x04	UART4 中断使能寄存器 （UART4 Interrupt Enable Register）	UART4_IER
0x08	UART4 中断标志寄存器 （UART4 Interrupt Status Register）	UART4_ISR
0x10	UART4 接收缓冲寄存器 （UART4 Receive Buffer）	UART4_RXBUF
0x14	UART4 发送缓冲寄存器 （UART4 Transmit Buffer）	UART4_TXBUF
0x18	UATR4 波特率产生寄存器 （UART4 Baud rate Generator Register）	UART4_BGR
0x20	UART4 模式寄存器 （UART4 Mode Control Register）	UART4_MCR
0x40	UART4 智能卡状态标志寄存器 （UART4 Smart Card Interrupt Status Register）	UART4_SCISR
0x44	UART4 智能卡中断使能寄存器 （UART4 Smart Card Interrupt Enable Register）	UART4_SCIER
0x48	UART4 智能卡帧格式控制寄存器 （UART4 Smart Card Frame Format Control Register）	UART4_FFCR
0x4C	UART4 智能卡额外保护时间寄存器 （UART4 Smart Card Extra Guard Time Register）	UART4_EGTR
0x50	UART4 智能卡输出时钟分频寄存器 （UART4 Smart Card Clock Output Register）	UART4_CODR
UART5 寄存器		
0x00	UART5 控制状态寄存器 （UART5 Control Status Register）	UART5_CSR
0x04	UART5 中断使能寄存器 （UART5 Interrupt Enable Register）	UART5_IER
0x08	UART5 中断标志寄存器 （UART5 Interrupt Status Register）	UART5_ISR
0x10	UART5 接收缓冲寄存器 （UART5 Receive Buffer）	UART5_RXBUF
0x14	UART5 发送缓冲寄存器 （UART5 Transmit Buffer）	UART5_TXBUF
0x18	UATR5 波特率产生寄存器 （UART5 Baud rate Generator Register）	UART5_BGR
0x20	UART5 模式寄存器 （UART5 Mode Control Register）	UART5_MCR
0x40	UART5 智能卡状态标志寄存器 （UART5 Smart Card Interrupt Status Register）	UART5_SCISR
0x44	UART5 智能卡中断使能寄存器	UART5_SCIER

offset 地址	名称	符号
	（UART5 Smart Card Interrupt Enable Register）	
0x48	UART5 智能卡帧格式控制寄存器 （UART5 Smart Card Frame Format Control Register）	UART5_FFCR
0x4C	UART5 智能卡额外保护时间寄存器 （UART5 Smart Card Extra Guard Time Register）	UART5_EGTR
0x50	UART5 智能卡输出时钟分频寄存器 （UART5 Smart Card Clock Output Register）	UART5_CODR

21.11.1 红外调制寄存器（UART_IRCR）

名称	UART_IRCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG[10:8]		
位权限	R/W-0	R/W-0000				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG[7:0]							
位权限	R/W-11010010							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性 (Infra Red) 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数 (Tranmission High Duty)
10:0	**TZBRG**	红外调制频率 (Transmission Baud Rate)

21.11.2 UARTx 控制状态寄存器（UARTx_CSR）

名称	**UARTx_CSR(x=0~5)**							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						TXIREN	RXTOEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	CKOEN	OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG

位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-01		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:25	-	未实现：读为0
24	**BUSY**	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23:18	-	未实现：读为0
17	**TXIREN**	发送红外调制使能 (Transmit Infra-red modulation Enable) 1：使能红外调制发送 0：关闭红外调制发送
16	**RXTOEN**	接收超时使能 (Receive Time-Out Enable) 1：使能接收超时功能 0：关闭接收超时功能
15	-	未实现：读为0
14	**CKOEN**	U7816 时钟 CLK 输出使能控制位 (Clock output Enable) 1：7816 时钟输出使能 0：7816 时钟输出禁止 *注：仅 UART2/3/4/5 在智能卡模式下有效*
13	**OVSM**	接收过采样模式选择 (Oversampling mode) 0：16 倍过采样 1：8 倍过采样
12	**IOSWAP**	RX 和 TX 引脚交换 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
11	**NEWUP**	UART RX下降沿唤醒功能使能寄存器（仅UART0和UART1有效）(Negedge Wakeupenable) 1：使能RX下降沿唤醒 0：禁止RX下降沿唤醒
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在UART通过DMA进行发送时有效 (DMA transmit interrupt enable) 1：IE=1的情况下，DMA模式下发送完最后一帧后，允许中断信号输出；最后一帧之前的数据帧发送完成后不允许中断信号输出 0：是否允许中断信号输出仅由IE决定
9	**BITORD**	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	**STOPCFG**	停止位宽度配置，仅对发送帧格式有效，接收时不判断停止位个数 (Stop bit config) 0：1位停止位 1：2 位停止位
7:6	**PDSEL**	每帧的数据长度选择；此寄存器对数据发送和接收同时有效 (Payload data length Select) 00：7 位数据 01：8 位数据 10：9 位数据 11：6 位数据

位号	助记符	功能描述
5:4	**PARITY**	校验位配置；此寄存器对数据发送和接收同时有效 00：无校验位 01：偶校验 10：奇校验 11：RFU
3	**RXPOL**	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	**TXPOL**	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	**RXEN**	接收使能，1 有效 (Receive Enable)
0	**TXEN**	发送使能，1 有效 (Transmit Enable)

21.11.3 UARTx 中断使能寄存器（UARTx_IER）

名称	UARTx_IER(x=0~5)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					-	-	-
位权限	U-0					U-0	U-0	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	U-0				R/W-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWUP_IE	-					TXBE_IE	TXSE_IE
位权限	R/W-0	U-0					R/W-0	R/W-0

位号	助记符	功能描述
31:27	-	**未实现：读为0**
26	-	**未实现：读为0**
25	-	**未实现：读为0**
24	-	**未实现：读为0**
23:17	-	**未实现：读为0**
16	-	**未实现：读为0**
15:12	-	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能，1 有效 (Receive Time-Out Interrupt Enable) （仅 UART0 和 UART1 有效）
10	**RXERR_IE**	接收错误中断使能，1 有效(Receive Error Interrupt Enable)
9	-	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)

位号	助记符	功能描述
7	**NEWUP_IE**	RX 下降沿异步检测中断使能，1 有效 (Negedge Wakeup Interrupt Enable)
6:2	**-**	**未实现：读为 0**
1	**TXBE_IE**	发送缓存空中断使能，1 有效 (Transmit Buffer Empty Interrupt Enable)
0	**TXSE_IE**	发送缓存空且发送移位寄存器空中断使能，1 有效 (Transmit Shift register Empty Interrupt Enable)

21.11.4 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0~5)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					-	-	-
位权限	U-0					U-0	U-0	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	-				PERR	FERR	OERR
位权限	U-0	U-0				R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RXTO	-		RXBF
位权限	U-0				R/W-0	U-0		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWKF	-				TX_OERR	TXBE	TXSE
位权限	R/W-0	U-0				R/W-0	R-1	R/W-0

位号	助记符	功能描述
31:27	**-**	未实现：读为0
26	**-**	未实现：读为0
25	**-**	未实现：读为0
24	**-**	未实现：读为0
23	**-**	未实现：读为0
22:19	**-**	未实现：读为0
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error,write 1 to clear)
17	**FERR**	帧格式错误中断标志，硬件置位，软件写 1 清零 (Frame Error flag,write 1 to clear)
16	**OERR**	接收缓存溢出错误中断标志，当接收缓存满的情况下，收到新的数据时置位；硬件置位，软件写 1 或者读取 RXBUF 时清零 接收溢出时，接收缓冲器中原有的数据被新数据覆盖。 (RX buffer Overflow Error flag,write 1 to clear)
15:12	**-**	未实现：读为0
11	**RXTO**	接收超时中断标志，硬件置位，软件写 1 清零 (Receive Time-Out flag,write 1 to clear)
10:9	**-**	未实现：读为0
8	**RXBF**	接收缓存满中断标志，硬件置位，软件写 1 或者读取 RXBUF 时清零 (Receive Buffer Full flag write 1 to clear)

位号	助记符	功能描述
7	**NEWKF**	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零 (Negedge Wakeup Flag write 1 to clear) （仅 UART0 和 UART1 有效）
6:3	**-**	未实现：读为 0
2	**TX_OERR**	发送缓存不为空时,写入新的 TXBUF 硬件置位,软件写 1 清零 (Transmit Buffer Overflow Error flag)
1	**TXBE**	发送缓存空中断标志，硬件置位，软件写入 TXBUF 时清零 (Transmit Buffer Empty flag)
0	**TXSE**	发送缓存空且移位寄存器发送完成中断标志，硬件置位，软件写 1 或者软件写发送缓存时清零 (Transmit Shift register Empty flag,write 1 to clear)

21.11.5 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(x=0~5)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15:8	**TXDLY_LEN**	发送延迟，最大 255baud (Transmit Delay Length)
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud (Receive Time-Out Length)

21.11.6 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF(x=0~5)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]

位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**RXBUF**	接收数据缓冲寄存器数据(Receive buffer)

7位收发时，接收的7bits数据存入RXBUF[6:0]

21.11.7 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF(x=0~5)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**TXBUF**	发送数据缓冲寄存器数据(Transmit Buffer)

7位收发时，发送的7bits数据写入TXBUF[6:0]

21.11.8 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR(x=0~5)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG[15:8]							
位权限	R/W-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG[7:0]							

位权限	R/W-01000001

位号	助记符	功能描述
31:16	-	未实现：读为0
15:0	**SPBRG**	波特率产生器寄存器值(Serial Port Baud Rate Generation) 软件写 SPBRG 将导致波特率分频计数器清零并重新开始计数， 软件应避免在正常通信过程中改写 SPBRG

波特率计算详见波特率发生章节

注：(1)OVSM=0

当SPBRG <= 0x000F时，UARTDIV=16'H000F;

当SPBRG >0x000F时，UARTDIV=SPBRG;

(3) OVSM=1

当SPBRG <= 0x0007时，UARTDIV=16'H0007;

当SPBRG >0x0007时，UARTDIV=SPBRG;

智能卡模式下，Baud ＝ F_{7816}/(SPBRG ＋ 1)，F_{7816}为智能卡输出工作时钟频率，由CODR寄存器配置

智能卡模式下SPBRG最小可用值是0x1，禁止配置0x0

21.11.9 UARTx 模式寄存器（UARTx_MCR）

名称	UARTx_MCR(x=0~5)							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					UARTMD	-	MODE
位权限	U-0					R/W-0	U-0	R/W-0

*UART0和UART1

位号	助记符	功能描述
31:3	-	未实现：读为0
2	**UARTMD**	UART 半双工单线模式选择 0：双线 UART 1：单线半双工 UART 注意，此寄存器仅在 MODE=0 时有效
1	-	未实现：读为0
0	**MODE**	模块控制寄存器 RFU，保持 0

*UART2、UART3、UART4、UART5

位号	助记符	功能描述
31:3	-	未实现：读为0
2	**UARTMD**	UART 半双工单线模式选择 0：双线 UART 1：单线半双工 UART 注意，此寄存器仅在 MODE=0 时有效
1	-	未实现：读为0
0	**MODE**	模块控制寄存器 0：标准 UART 模式 1：智能卡模式（ISO7816-3）

21.11.10 UARTx 智能卡状态标志寄存器（UARTx_SCISR）

名称	UARTx_SCISR (x=2,3,4,5)							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					WAIT_RPT	TXBUSY	RXBUSY
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TPARERR	RPARERR	FRERR	OVERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RXIF	TXIF	ERRIF
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:19	--	RFU：未实现，读为 0
18	WAIT_RPT	U7816 接口发送了错误信号，正在等待对方重发数据；(Waiting for Repeat flag) 状态机进入发送错误信号状态时置位，收到数据起始位或者进入发送状态时硬件清零； 软件只读。
17	TXBUSY	发送数据忙标志。（发送完成后自动清零）(Transmission busy flag) 1：处于数据发送状态，发送移位寄存器正在发送数据。（开始发送起始位置 1，停止位中间清零） 0：数据发送空闲
16	RXBUSY	接收数据忙标志。（接收完成后自动清零）(Receiving busy flag) 1：处于数据接收状态，接收移位寄存器正在接收数据。（收到起始位置 1，收到停止位清零，若接收数据出错需重发，则回发 error signal 时清零） 0：数据接收空闲
15:12	--	RFU：未实现，读为 0

Bit	助记符	功能描述
11	TPARERR	发送数据奇偶校验错误标志位。硬件置位，写 1 清零 (Transmit Parity Error,write 1 to clear)
10	RPARERR	接收数据奇偶校验错误标志位。硬件置位，写 1 清零 (Receive Parity Error flag,write 1 to clear)
9	FRERR	接收帧格式错误标志位。硬件置位，写 1 清零 (Frame Error flag,write 1 to clear) 1：帧格式有错误，接收到的 frame 字节长度有误或接收到的 frame 或者 stop 位有误 0：接收数据时无帧格式错误
8	OVERR	接收溢出错误标志位。硬件置位，写 1 清零 (Receive Overflow Error,write 1 to clear) 1：接收缓冲寄存器未被读出，又接收到新的数据，溢出错误标志有效。原接收缓冲寄存器内数据被新覆盖 0：无溢出错误
7:3	--	RFU：未实现，读为 0
2	RXIF	接收完成标志(Receive interrupt flag)，U7816 接口控制器每收到 1byte 数据，根据接收的通道相应发出一次中断。硬件置位，读数据接收缓冲寄存器清零 1：接收到 1byte 数据，数据接收缓冲器满 0：未接收到数据，数据接收缓冲器空
1	TXIF	发送缓冲区空标志(Transmit interrupt flag)，上电复位后此标志就自动置位，表示缓冲区空，可以写入数据。软件写入数据后标志自动清除，数据从发送缓存移入移位寄存器后置 1 1：数据发送缓冲器空 0：数据发送缓冲器内有数据待发送
0	ERRIF	错误标志(Error interrupt flag)，寄存器配置出错或传输过程中出错。此 bit 是 TPARERR、RPARERR、FRERR、OVERR 的或。软件通过清除以上错误标志寄存器来清除此 bit。

*仅智能卡模式下有效

21.11.11 UARTx 智能卡中断使能寄存器（UARTx_SCIER）

名称	UARTx_SCIER (x=2,3,4,5)							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RXIE	TXIE	LSIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	RXIE	数据接收中断使能位。对应 RXIF 中断标志位 1 = 数据接收中断使能位 0 = 数据接收中断使能位
1	TXIE	数据发送中断使能位。对应 TXIF 中断标志位 1 = 数据发送中断使能 0 = 数据发送中断禁止
0	LSIE	线路状态中断使能位。对应 ERRIF 中断标志位 1 = 线路状态中断使能 0 = 线路状态中断禁止

21.11.12 UARTx 智能卡帧格式控制寄存器（UARTx_FFCR）

名称	UARTx_FFCR (x=2,3,4,5)							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					ERSW		ERSGD
位权限	U-0					R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BGTEN	REP_T	PAR		FREN	TREPEN	RREPEN	DICONV
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:11	DUMMY	可读写，无功能
10:9	ERSW	ERROR SIGNAL 宽度选择 (Error Signal Width) 11：ERROR SIGNAL 宽度为 1ETU; 10：ERROR SIGNAL 宽度为 1.5ETU; 01：ERROR SIGNAL 宽度为 2ETU; 00：ERROR SIGNAL 宽度为 2ETU;
8	ERSGD	ERROR SIGNAL 后 GUARDTIME 宽度选择（仅在发送时有效）(Error Signal Guard Time) 1：ERROR SIGNAL 后 GUARDTIME 为 1~1.5ETU。 0：ERROR SIGNAL 后 GUARDTIME 为 2~2.5ETU。 ERROR SIGNAL 宽度为整数 ETU 时 GUARDTIME 为 1.5 或 2.5ETU；ERROR SIGNAL 宽度为 1.5ETU 时 GUARDTIME 为 1 或 2ETU
7	BGTEN	BGT 控制位。控制接收->发送之间是否插入 BGT。BGT 是接收->发送之间需要的最小时间（block guard time enable） 1：BGT 使能，插入 Block guard time(12 etu);

Bit	助记符	功能描述
		0：BGT 禁止，不插入 Block guard time(12 etu);
6	REP_T	控制接收数据奇偶校验出错时自动重发次数 (Repeated Times) 1：3 次 0：1 次
5:4	PAR	奇偶校验类型选择 (Parity) 00：Even 01：Odd 10：Always 1 11：不校验，处理
3	FREN	Guard Time 接收长度控制位(Receive short Frame) 1：Guard time 为 1 etu 0：Guard time 为 2 etu
2	TREPEN	发送数据奇偶校验错的处理方式选择 (Transmit Repeat Enable) 1：收到奇偶校验出错标志（error signal），根据 T＝0 协议自动进行回发。在单一 byte 重复发送次数超过 REP_T 后，置 tx_parity_err 标志，进行中断 0：收到 Error signal 时不进行自动回发，置 tx_parity_err 标志，直接中断
1	RREPEN	接收数据奇偶校验错的处理方式选择 (Receive Repeat Enable) 1：奇偶校验错，根据 T=0 协议自动回发 ERROR SIGNAL。单一 BYTE 连续接收次数超过 REP_T 后，置 RX_PARITY_ERR 标志，进行中断 0：奇偶校验错，不自动发送 ERROR SIGNAL，置 RX_PARITY_ERR 标志，进行中断
0	DICONV	传输次序，编码方式选择 1 = 反向编码，先收发 MSB；(收发数据+校验位)反逻辑电平 0 = 正向编码，先收发 LSB ； (收发数据+校验位)正逻辑电平

21.11.13 UARTx 智能卡额外保护时间寄存器（UARTx_EGTR）

名称	UARTx_EGTR (x=2,3,4,5)							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXEGT							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	TXEGT	发送时插入的 EGT 时间（以 ETU 为单位） (Transmit Extra Guard Time)

21.11.14 UARTx 智能卡输出时钟分频寄存器（UARTx_CODR）

名称	UARTx_CODR (x=2,3,4,5)							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			CLKDIV				
位权限	U-0			R/W-00000				

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	CLKDIV	U7816 时钟输出分频控制寄存器(Clock Divider)，控制 7816 工作时钟分频数。 U7816 工作时钟与 APBCLK 的分频关系： $F_{7816}=F_{APBCLK}/(CLKDIV+1)$ 特殊情况：CLK_DIV 设置成 0 或 1 时，$F_{7816}=F_{APBCLK}/2$ 注：7816 协议规定的工作时钟范围是 1~5MHZ。

22 串行外设接口（SPI/I2S）

22.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过 4 线交换数据的串行同步通讯手段。SPI 接口模块可配置为主设备或从设备，实现与外部的 SPI 通信。

I2S 同样是一个同步串行通信外设，能够以主机或从机模式进行半双工通信，可以支持飞利浦 I2S 标准，MSB-LSB- justified，或 PCM 标准。

本模块可以配置为 SPI 或 I2S 功能。

SPI 特性：

- 全双工4线串行同步收发（SCLK, MOSI, MISO, SSN）
- MISO和MOSI可交换引脚顺序
- 半双工4线串行同步收发（SCLK, SDATA, SSN, DCN），用于TFT屏驱动
- 4路独立通道
- 主从模式
- 可编程时钟极性和相位
- 可编程比特速率
- 可编程数据字长（8/16/24/32bits）
- 最大波特率为$F_{APBCLK}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持DMA

I2S特性：

- 支持主机和从机模式，半双工通信
- 8bit 时钟预分频电路
- 支持 16/24/32bit 数据格式
- 音频通道数据帧长度为 16bit 或 32bit
- 可编程时钟极性
- 支持 I2S 协议：
  - ✓ Philips I2S 标准

- ✓ MSB-justified 标准
- ✓ LSB-justified 标准
- ✓ PCM 标准

- 支持 DMA
- 支持主机时钟输出（256×Fs，Fs 为音频采样频率）

22.2 SPI 功能描述

22.2.1 SPI 结构框图

下图为 SPI 模块的结构示意图。

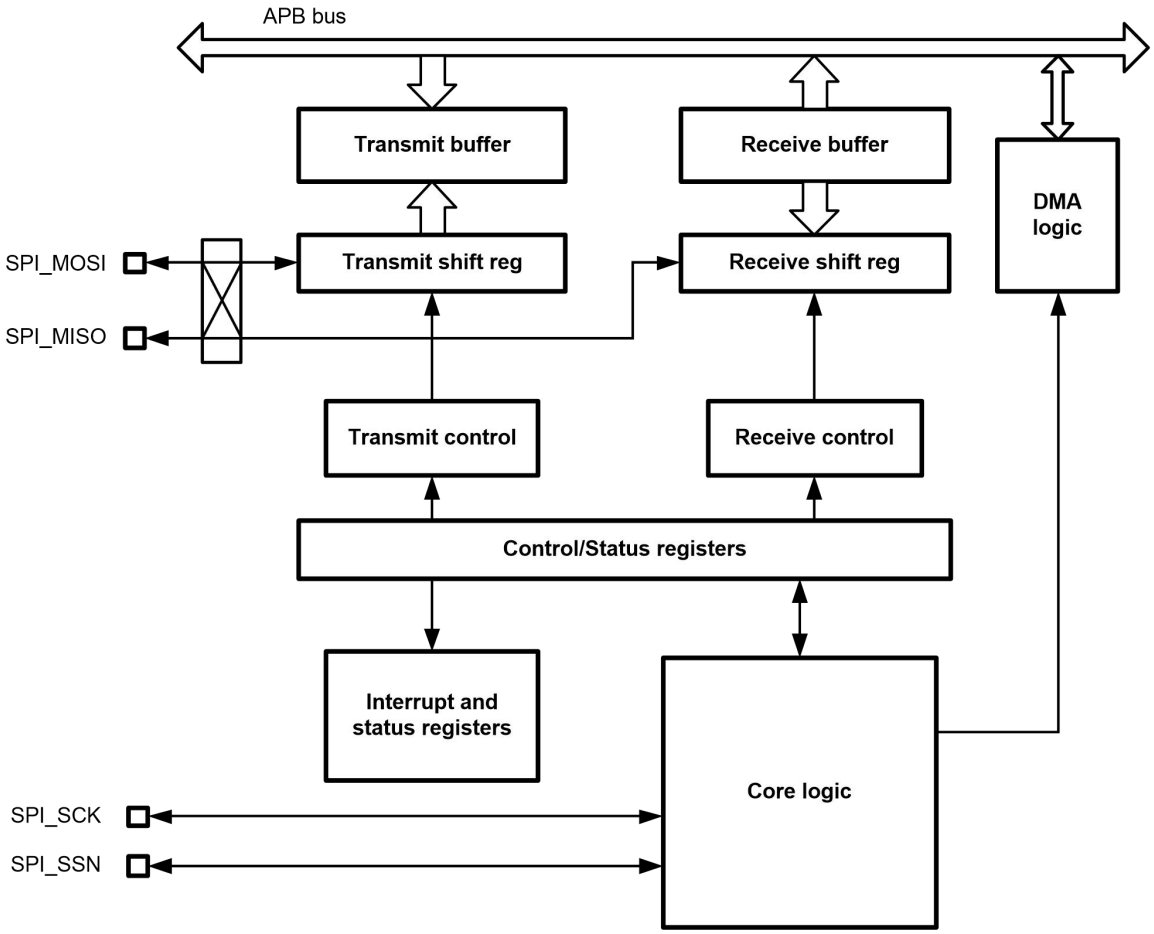


图 22-1 SPI 结构框图

22.2.2 引脚定义

SPI 模块使用 4 个引脚与外部器件通信，在全双工和半双工模式下，这四个引脚的功能定义有所不同，如下表所示：

引脚	SPIx	全双工	功能	半双工	功能
PE1、PA13	SPI0	SSN	片选信号	SSN	片选信号
PE0、PA0		SCLK	时钟	SCLK	时钟
PB14、PA15		MISO	主机输入从机输出	DCN	命令/数据标识
PB15、PA14		MOSI	主机输出从机输入	SDATA	双向数据
PD11	SPI1	SSN	片选信号	SSN	片选信号
PA10、PD9		SCLK	时钟	SCLK	时钟
PA11、PD8		MISO	主机输入从机输出	DCN	命令/数据标识
PA12、PD13		MOSI	主机输出从机输入	SDATA	双向数据
PC9	SPI2	SSN	片选信号	SSN	片选信号
PE8、PC3、PC8		SCLK	时钟	SCLK	时钟
PE7、PC4、PC7		MISO	主机输入从机输出	DCN	命令/数据标识
PE6、PC5、PC6		MOSI	主机输出从机输入	SDATA	双向数据
PE2	SPI3	SSN	片选信号	SSN	片选信号
PE3、PC0		SCLK	时钟	SCLK	时钟
PE4、PC2		MISO	主机输入从机输出	DCN	命令/数据标识
PE5、PC1		MOSI	主机输出从机输入	SDATA	双向数据

表 22-1SPI 引脚列表

22.2.3 SPI 接口时序

为了兼容不同的 SPI 外设，SPI 串行时钟的时序可以通过时钟相位选择位（CPHA）和时钟极性选择位(CPOL)设置产生 4 种不同组合。为保证数据正确传输，主从器件的时序配置必需一致。

当处于从器件模式或 SPI 系统使能位（SPIEN）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

22.2.3.1 CPHA=0

CPHA=0 时，SPI 模块在串行时钟的第一个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，SPI 在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，SPI 在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据。

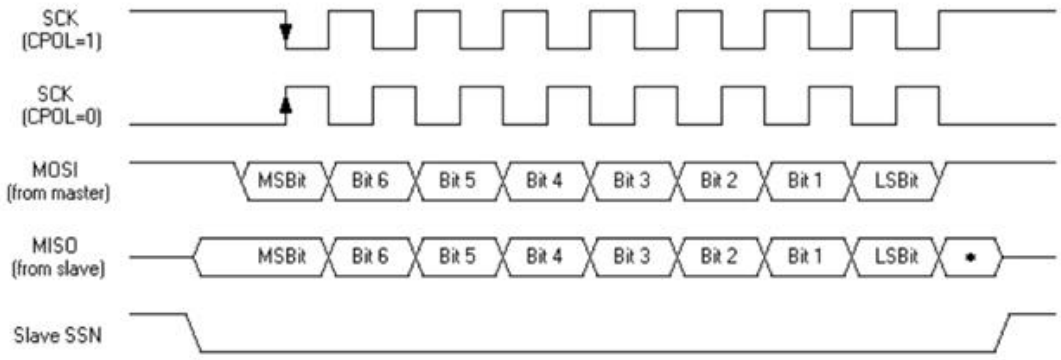


图 22-2 SPI 数据/时钟时序图（CPHA=0）

22.2.3.2 CPHA=1

CPHA=1 时，SPI 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据。

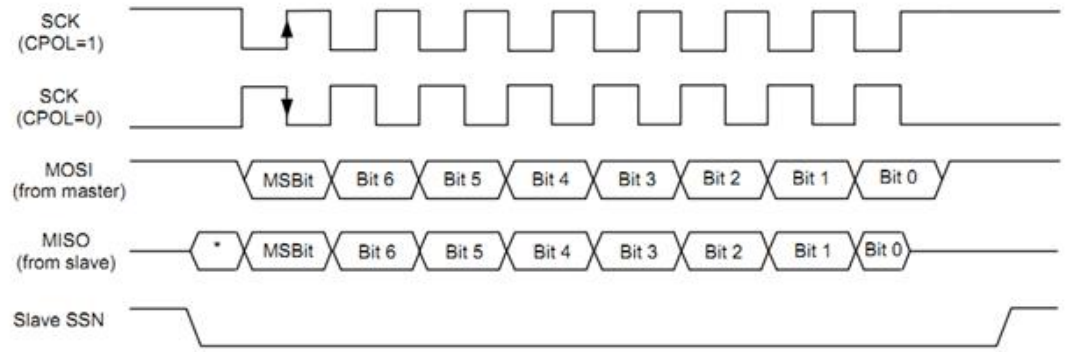


图 22-3 SPI 数据/时钟时序图（CPHA=1）

22.2.3.3 4 线半双工模式（主机）

4线半双工模式可以支持与点阵液晶或TFT屏的交互通信。在这种模式下，通过DCN信号的高低来区分当前发送的是命令帧还是数据帧。双向数据都通过SDATA（MOSI）引脚收发，由硬件自动完成数据方向切换。FM33FR0的SPI仅支持4线半双工主机模式，不支持从机模式。

所有通信都由主机发起，主机首先发送命令帧，然后再进行数据帧传输。命令帧和数据帧通过DCN信号线区分。主机可以通过4线半双工接口向从机写入数据，或从从机读取数据。

4线半双工写操作

软件通过清零HD_RW寄存器，表示当前主机要发起写操作。

主机发起写操作前，首先发送写命令帧。当写命令帧发送完毕后，如果发送缓冲区为空，则硬件将拉高SSN并停止SCLK发送；如果发送缓冲区已经写入了新的数据，则硬件会连续发送后续的数据帧。

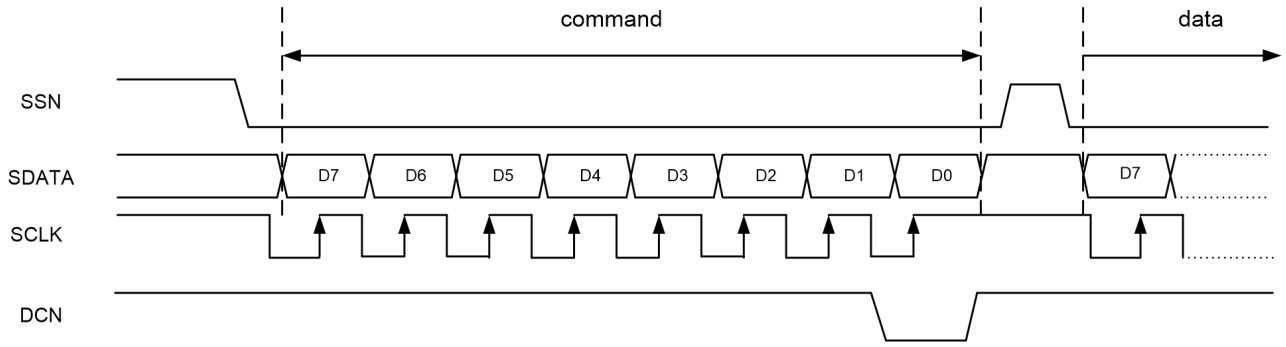


图 22-44 线半双工写操作

DCN在第8个时钟上升沿采样判决，如果为0，表示当前帧是命令帧。发送命令帧前，软件需要将DCN寄存器写0，命令帧发送完成后硬件自动将DCN寄存器置位。

4线半双工读操作

软件通过置位HD_RW寄存器，表示当前主机要发起读操作。

4线半双工读操作支持8位、24位和32位读取。主机发起读操作时，首先发送读命令帧。当读命令帧发送完毕后，可以根据寄存器配置发送1个dummy cycle，在dummy cycle期间，SCLK时钟正常发送，但是主机不驱动SDATA，也不接受SDATA输入。

完成命令帧和dummy cycle（可选）后，4线半双工SPI自动进入接收状态，SDATA信号改由从机驱动，主机收到的数据帧将被写入接收缓冲区。每个数据帧接收完成后，将置位RXBF中断标志寄存器。软件应及时读取接收缓冲区中的数据，如果接收缓冲区和接收移位寄存器都处于满状态，硬件会停止SCLK发送，暂停从从机读取数据，直到软件或DMA读取了接收缓冲区。

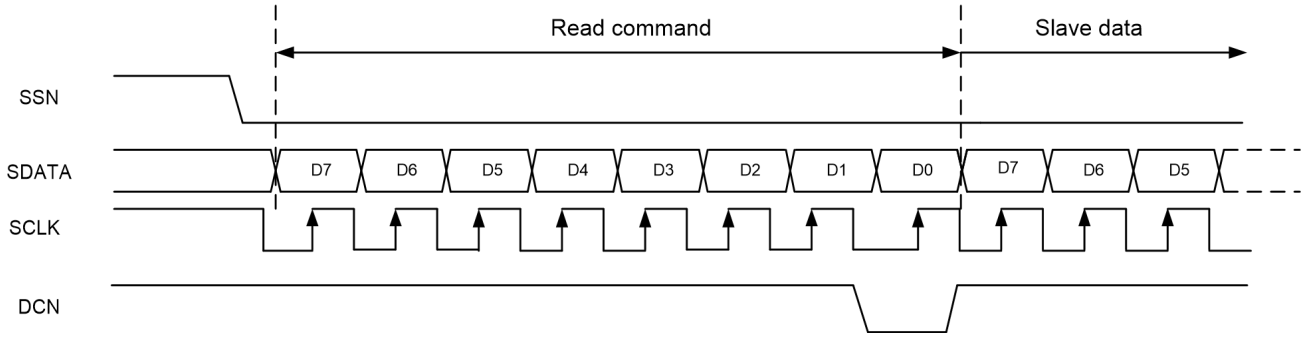


图 22-5 4 线半双工读操作（无 dummy cycle）

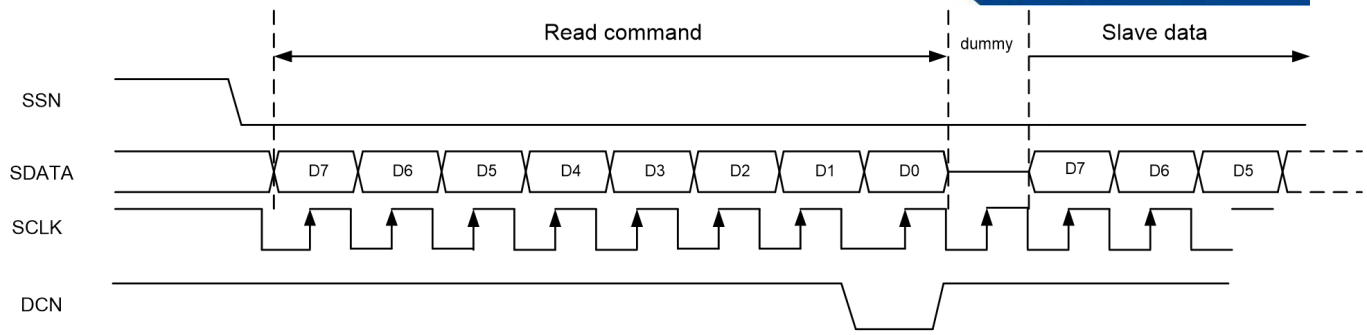


图 22-6 4 线半双工读操作（有 dummy cycle）

22.2.4 I/O 配置

主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。数据传输时 MSB 在前。

主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。数据传输时 MSB 在前。

串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，通常主机 SSN 输出与从机 SSN 输入直接相连。如果总线上只有一个从机，也可以通过将从机 SSN 接地使从机常使能，此时主机 SSN 输出可以用作 GPIO。

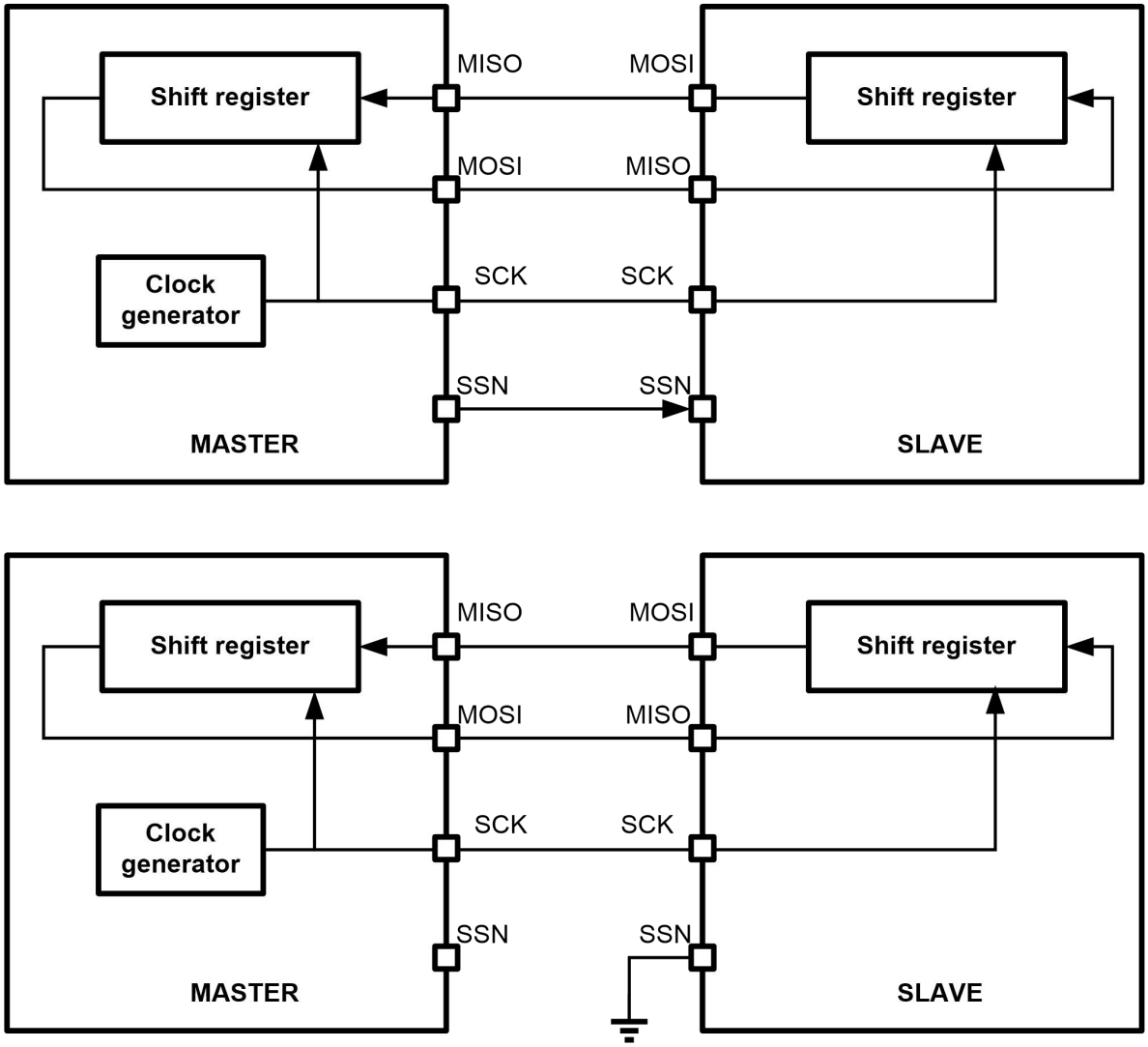


图 22-7 SPI Master/SPI Slave 互连

22.2.5 全双工数据通信

SPI模块默认为全双工通信，如果需要实现连续不间断的数据通信，软件需要确保TX BUFFER非空。即使软件只用SPI进行数据接收，由于SPI的全双工属性，软件仍需要对TX BUFFER进行写操作，此时写入的是无效数据，可根据MOSI无效状态配置写入全0或全F。

发送缓冲区

软件或DMA将待发送数据写入发送缓冲区（SPIxTXBUF寄存器），当发送开始时，硬件将数据从发送缓冲区拷贝到移位寄存器并开始发送。数据从发送缓冲区转移至移位寄存器后，发送缓存空标志（TXBE）被置位，表示可以向TXBUF写入新数据；如果TXIE寄存器置位，则产生中断。通过向TXBUF写入数据，可以清零TXBE寄存器。

如果在移位寄存器移位完成前，新的数据被写入发送缓冲区，则可以保证连续不断的数据发送。在TXBE为0的情况下写TXBUF，则会产生数据冲突，参见22.2.9数据冲突。

发送移位寄存器中的数据被完全发送出去后，发送移位寄存器空标志TXSE置位，TXSE软件写1清0。软件可以通过查询TXSE了解数据是否发送完毕。

接收缓冲区

当SPI完成一帧数据接收后，收到的数据将从移位寄存器拷贝到接收缓冲区（SPIxRXBUF寄存器），同时RXBF标志被置位，表示RXBUF中已有数据待处理。如果RXIE寄存器置位，则产生中断。通过读取RXBUF可以清零RXBF标志。

在RXBF没有置位的情况下读RXBUF，将返回上一次接收到的数据；如果应用没有及时处理RXBF，新的数据在RXBF置位的情况下完成接收，则产生数据冲突，参见22.2.9数据冲突。

BUSY标志

当SPI正在进行数据收发时，BUSY寄存器置位。此寄存器在某些场景下可以用来判断最后一帧数据是否传输完毕。比如TXBE只是表示数据已经进入移位发送，但是真正发送完成，需要等待BUSY标志清零。也可以通过TXSE判断数据发送完成。

如何启动SPI通信

主机模式下，建议遵循以下步骤启动SPI通信：

- 应用配置SPI模块
- 置位SPIEN
- 向TXBUF写入数据，SPI模块自动开始发送SCK并进行数据收发

从机模式下，建议应用在主机开始发送SCK之前完成配置和使能，并将第一帧待发送数据写入TXBUF，等待主机发送SCK开始通信。

如何结束SPI通信

主机模式下，建议遵循以下步骤结束SPI通信：

- 等待RXBF和TXBE标志置位，此时移位寄存器中还有最后一帧数据正在发送
- 查询BUSY标志，直到BUSY为0，最后一帧数据收发完成
- 关闭SPI模块，如果需要，读取最后一帧接收数据

从机模式下，应用可以在读取任意一帧数据后关闭SPI模块，关闭前已经被移入移位寄存器的数据将被忽略。

22.2.6 TX-ONLY 模式

某些时候SPI通信是半双工的，在主机仅需进行发送的情况下，通过置位TXO寄存器进入TX-ONLY

模式，此时MISO收到的数据不会被写入RX Buffer中，相应的也不会置位RXBF中断标志。

通过置位TXO_AC，可以实现TXO自动清零功能。在TX-ONLY模式下，如果TX buffer空（TXBE置位）并且发送移位寄存器空，则TXO寄存器自动清零，退出TX-ONLY状态。

22.2.7 RX-ONLY 模式

SPI主机仅需进行接收的情况下，通过置位RXO寄存器进入RX-ONLY模式，此时SPI模块无需软件对TX Buffer进行写操作，即可进行连续不断的数据接收，此时MOSI将保持IDLE电平，并且不会置位TXBE中断标志寄存器。

22.2.8 主机 SSN 控制

SPI模块主机支持硬件或软件控制SSN信号。

当SSNSEN寄存器清零时，SSN由硬件电路控制；如果SSNM寄存器置位，则SPI每发完一帧数据后，将拉高SSN，SSN高电平时间由WAIT寄存器配置（若干个SCK时钟周期）；

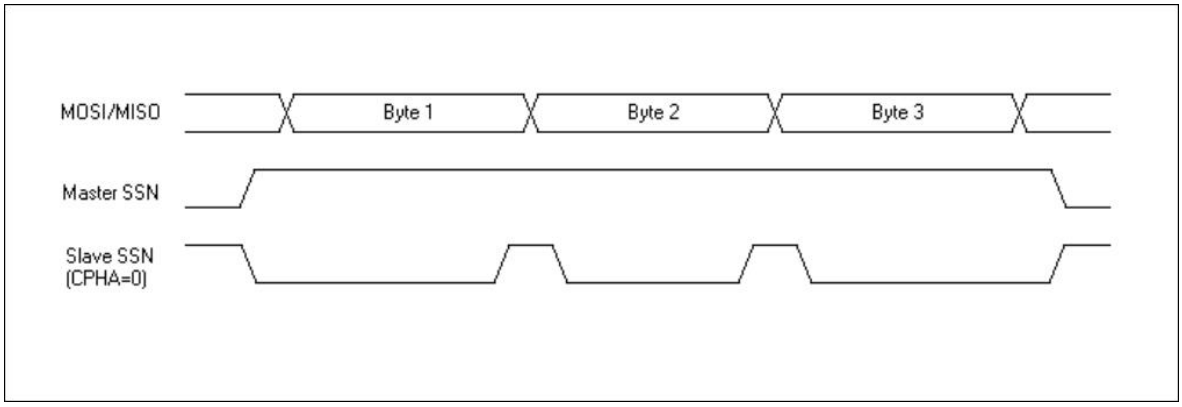


图 22-8 SPI SSN 时序图（SSNM=1）

如果SSNM寄存器复位，则SPI每发完一帧数据后不会拉高SSN，而是直接进入下一帧数据发送。

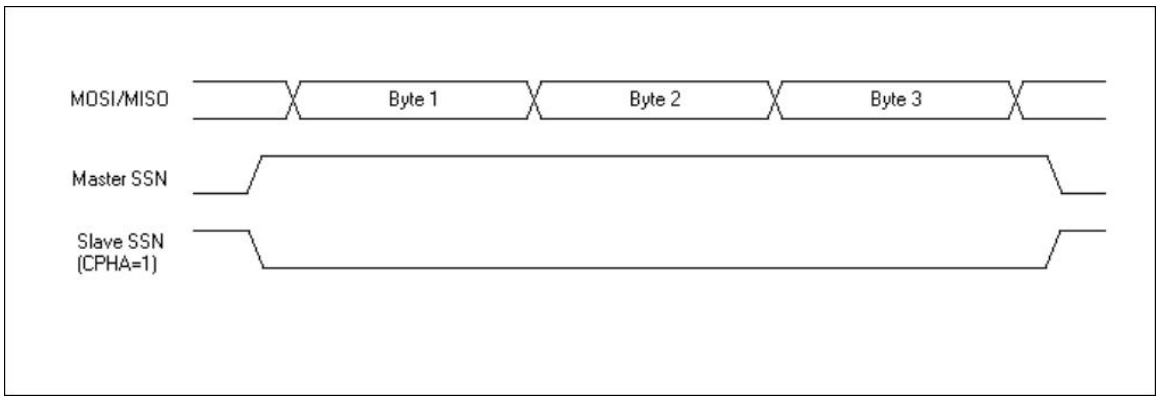


图 22-9 SPI SSN 时序图（SSNM=0）

当SSNSEN寄存器置位时，SSN直接由软件控制。软件通过写SPIxCR2.SSN寄存器位，可以直接操作SPI主机发送的SSN电平。

22.2.9 数据冲突

当 SPI 的 TX Buffer 数据尚未被读进移位寄存器，或者 SPI 的 RX Buffer 中的数据未被软件或 DMA 读取时，对 TX Buffer 或 RX Buffer 的写操作会产生对应的冲突错误，TXCOL/RXCOL 位会置起，产生中断。导致冲突的写入数据将被忽略。数据冲突错误在主从模式下都会产生。

对 TX Buffer 的写操作，由芯片内部的 Master 模块发起，包括 CPU、DMA 等等。对 RX Buffer 的写操作，则由外部 SPI 器件发起。

当数据冲突发生时，TX Buffer 和 RX Buffer 内原有数据不会被刷新，新写入的数据丢失。

22.2.10 使用 DMA 进行 SPI 收发

当 SPI 模块被使能后，SPI 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 SPI 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 SPI 请求，并完成 RAM 和 SPI 之间的数据搬运。

注意：如果使用 DMA 进行收发全双工通信，软件应先使能 DMA 发送通道，再使能 DMA 的接收通道；反之可能会导致 SPI 额外发送一个字节的 dummy 数据。

使用 DMA 进行 SPI 接收

- 将 DMA 通道 3 或 5 配置为 SPI_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 SPI 模块参数
- 使能 SPI 模块，等待数据接收
- 收到数据后 SPI 自动产生 DMA 请求
- DMA 响应请求，读取 SPI 接收缓存寄存器，写入指定 RAM 地址
- 当指定长度的 DMA 传输结束后，DMA 将忽略后续请求并产生传输完成中断，软件应处理中断并关闭 SPI
- 如果关闭 SPI 前又有数据被接收，软件可以通过写 RXBFC 清除 RXBUF

使用 DMA 进行 SPI 发送

DMA发送过程与上述接收过程类似，主要差别是，当指定长度的DMA传输结束后，软件不能立即关闭SPI，因为此时最后一帧数据还在移位发送中，因此软件需要查询BUSY标志直到移位发送结束，再关闭SPI模块。

数据帧长度与 RAM 数据组织方式

SPI 传输帧长度可以配置为 8、16、24、32bit。

当数据帧长度为 8bit 时，DMA 每次搬运 1byte，4 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data3，data2，data1，data0 }

当数据帧长度为 16bit 时，DMA 每次搬运 2bytes，2 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data1，data0 }

当数据帧长度为 24bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址，但是有效数据仅占用 RAM 字内低 24bit：

RAM word: { 8'h0，data0 }

当数据帧长度为 32bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址：

RAM word: { data0 }

22.3 I2S 功能描述

22.3.1 I2S 结构框图

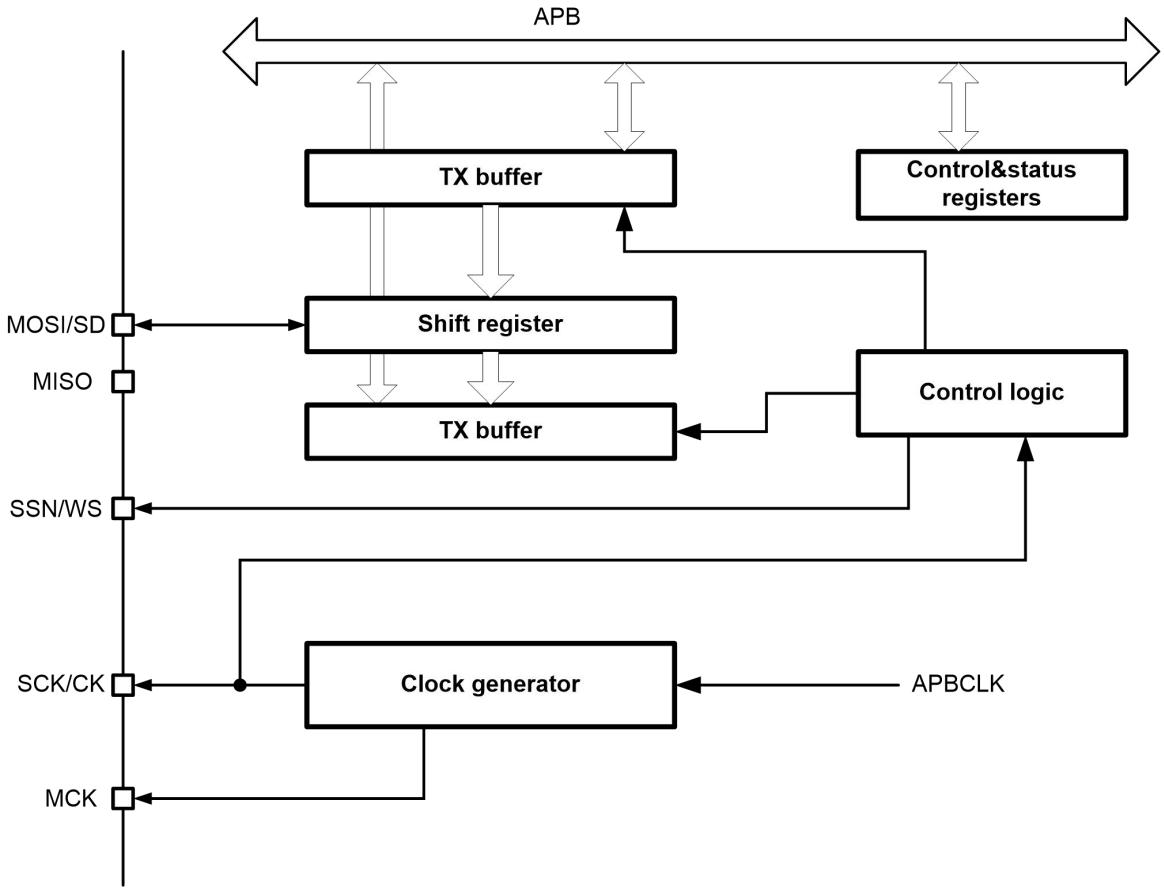


图 22-10I2S 结构框图

22.3.2 引脚定义

I2S与SPI模块共享3个引脚：

- SD：双向串行数据，与MOSI复用
- WS：声道选择，与SSN复用
- CK：串行时钟，与SCK复用

除了以上引脚，I2S还需要一个独立的主机时钟输出引脚：

- MCK：Master时钟输出，仅在主机模式下有效，输出频率为256*Fs，其中Fs为音频采样频率

22.3.3 协议支持

I2S 协议通过时分复用进行左右两个通道的数据收发。然而，I2S 模块只有 1 个 16bit 数据寄存器，

软件需要通过 CHSIDE 寄存器来区分数据所属的声道。在通信过程中，左声道数据总是先于右声道传输。在 PCM 模式下，CHSIDE 寄存器没有意义。

I2S 模块支持如下数据帧格式：

- 16bit 帧中传输 16bit 数据
- 32bit 帧中传输 16、24、32bit 数据

当使用 32bit 帧传输 16 位数据时，仅高 16 位有效，低 16 位被硬件固定为全 0。

24bit 或 32bit 数据帧传输需要 CPU 或 DMA 的 2 次读写操作（每次读写 16bit）。当使用 32bit 帧传输 24 位数据时，低 8 位被固定成全 0。

针对所有通信协议标准，都是最高位先发（MSB first）。

I2S 模块支持 4 种协议标准，通过 I2SSTD 寄存器配置当前格式。

22.3.3.1 Philips standard

Philips 标准使用 WS 信号表示当前发送的是哪个声道。WS 在数据帧 1st bit 之前一个时钟周期被拉低，表示左声道开始传输；左声道传输完成后，开始右声道传输。

典型数据传输波形：

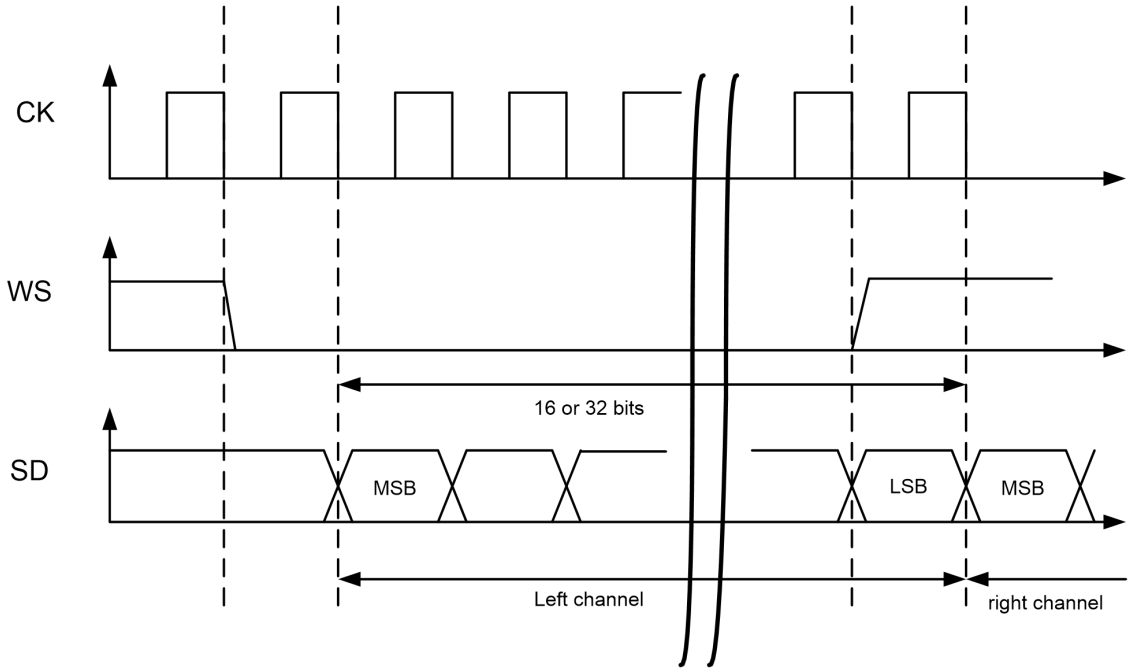


图 22- 11Philips 标准时序

22.3.3.2 MSB-justified standard

此标准与 Philips 的差异是，WS 信号与 1st bit 同步产生，并且 WS 高电平代表左声道。

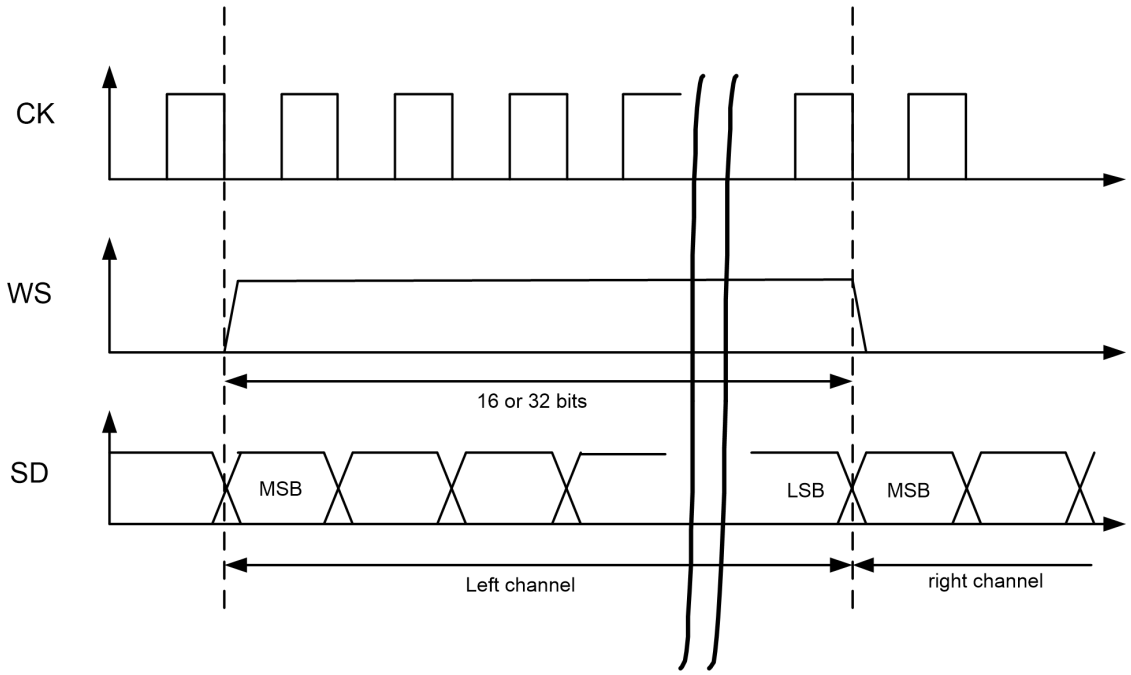


图 22-1232/16-bits MSB-justified 时序

在 32 位数据帧中传输 24 位数据时，MSB justified 格式将有效数据先发，最后补 0：

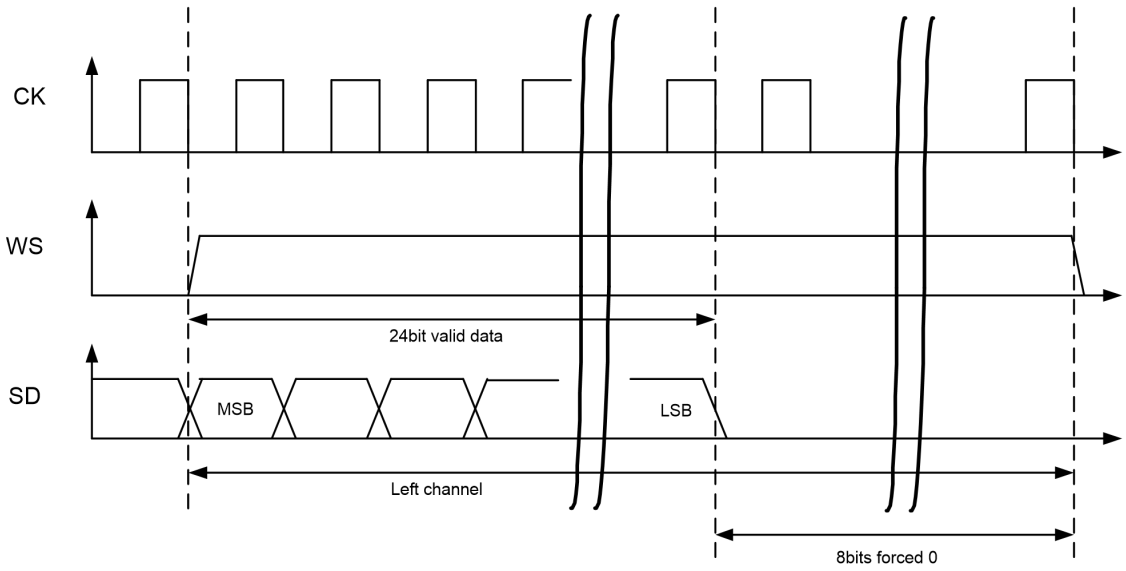


图 22-1324-bits MSB-justified 时序

在 32 位数据帧中传输 16 位数据时，MSB justified 格式同样是将有效数据先发，最后补 0。

22.3.3.3 LSB-justified standard

LSB justified 标准与 MSB justified 类似，只是在有效数据长度不到数据帧长的情况下，将有效数据靠右对齐，先发填充的 0。

在 32 位数据帧中传输 24 位数据时的波形：

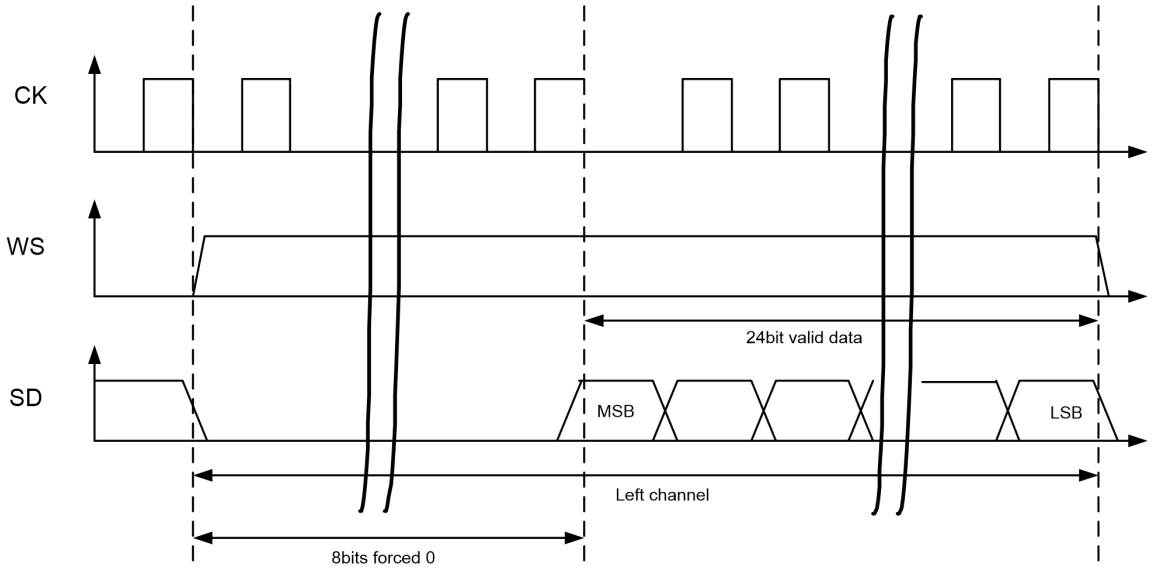


图 22-14LSB-justified 时序

22.3.3.4 PCM standard

针对 PCM 标准，无需使用声道信息。PCM 标准下支持长帧和短帧模式，通过 PCMSYNC 寄存器配置。WS 信号在 PCM 模式下用于数据帧起始同步，不代表左右声道识别。

长帧格式下，WS 信号固定置位 13bit 长度；短帧格式下，WS 信号只置位 1bit 长度。

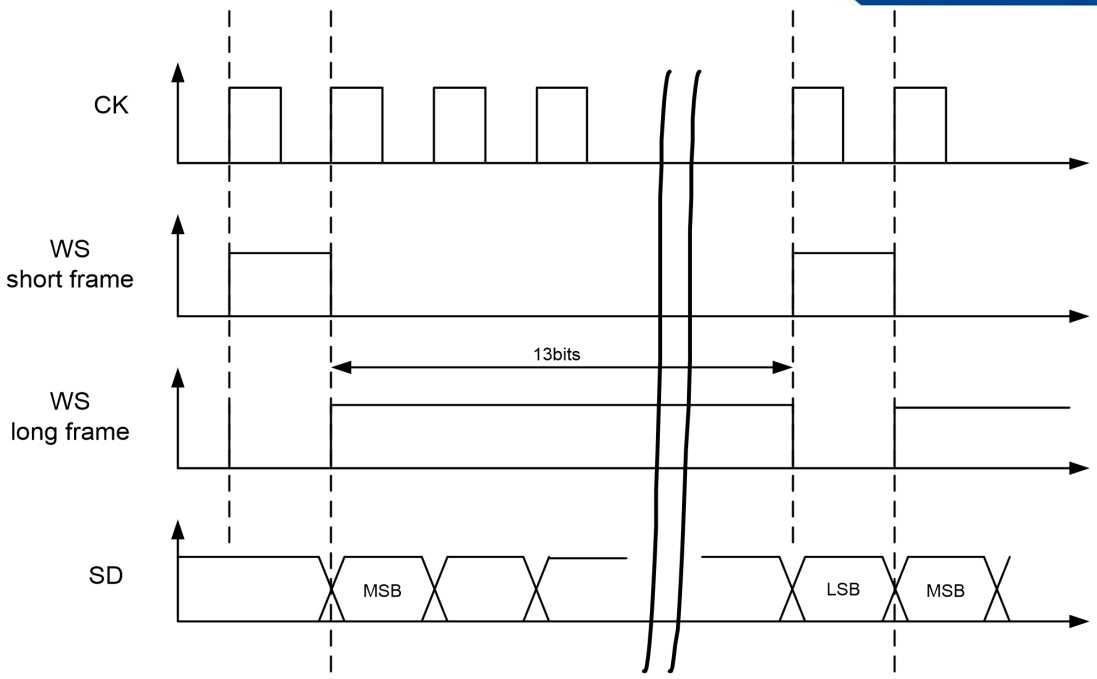


图 22-15PCM 标准时序

22.3.4 时钟产生

I2S 数据码率由下式决定：

I2S 码率 = 每声道数据帧位数×声道数×音频采样频率

假设采用 16 位音频帧格式，左右声道，音频采样率 F_S，则码率计算为：

I2S 码率 = 16×2×F_S

假设采用 32 位音频帧格式，左右声道，，音频采样率 F_S，则码率计算为：

I2S 码率 = 32×2×F_S

音频采样的频率包括：192Khz、96Khz、48Khz、44.1Khz、32Khz、22.05Khz、16Khz、11.025Khz、8Khz 等。为了获得以上目标频率，按照以下公式计算分频寄存器的值（CLKDIV 和 ODD 都是分频寄存器）：

如果输出 MCK：F_S = APBCLK / [(256)×((2×CLKDIV) + ODD)]，

如果不输出 MCK：F_S = APBCLK / [32×(CHLEN+1)×((2×CLKDIV) + ODD)]，

CHLEN=0：声道帧长 16bit

CHLEN=1：声道帧长 32bit

CLKDIV 寄存器定义了 50%占空比下时钟 CK 的高低电平宽度，ODD 寄存器用于调整高电平宽度，从而实现理论上的小数分频效果。

I2S 主机模式下可以通过 MCKOE 寄存器选择输出或者不输出 MCK，MCK 频率是 256×Fs

假设 APBCLK 为 48MHz，下表罗列了针对不同 Fs 的参考配置和实际采样频率误差。

APBCLK (MHz)	CHLEN	CLKDIV	ODD	MCK	Target Fs (Hz)	Actual Fs	Error
48	16	8	0	No	96000	93750	2.3438%
48	32	4	0	No	96000	93750	2.3438%
48	16	15	1	No	48000	48387.0968	0.8065%
48	32	8	0	No	48000	46875	2.3438%
48	16	17	0	No	44100	44117.6471	0.0400%
48	32	8	1	No	44100	44117.6471	0.0400%
48	16	23	1	No	32000	31914.8936	0.2660%
48	32	11	1	No	32000	32608.6957	1.9022%
48	16	34	0	No	22050	22058.8235	0.0400%
48	32	17	0	No	22050	22058.8235	0.0400%
48	16	47	0	No	16000	15957.4468	0.2660%
48	32	23	1	No	16000	15957.4468	0.2660%
48	16	68	0	No	11025	11029.4118	0.0400%
48	32	34	0	No	11025	11029.4118	0.0400%
48	16	94	0	No	8000	7978.7234	0.2660%
48	32	47	0	No	8000	7978.7234	0.2660%
48	16	2	0	Yes	48000	46875	2.3438%
48	32	2	0	Yes	48000	46875	2.3438%
48	16	2	0	Yes	44100	46875	6.2925%
48	32	2	0	Yes	44100	46875	6.2925%
48	16	3	0	Yes	32000	31250	2.3438%
48	32	3	0	Yes	32000	31250	2.3438%
48	16	4	1	Yes	22050	20833.3333	5.5178%
48	32	4	1	Yes	22050	20833.3333	5.5178%
48	16	6	0	Yes	16000	15625	2.3438%
48	32	6	0	Yes	16000	15625	2.3438%
48	16	8	1	Yes	11025	11029.4118	0.0400%
48	32	8	1	Yes	11025	11029.4118	0.0400%
48	16	11	1	Yes	8000	8152.17391	1.9022%
48	32	11	1	Yes	8000	8152.17391	1.9022%

表 22-2I2S 典型采样频率计算表

22.3.5 I2S 主机功能

主机模式下，模块负责产生 CK 和 WS 信号，MCK 可以配置为输出或不输出。

22.3.5.1 操作流程

I2S 模块通用操作流程如下：

- 配置 CLKDIV 和 ODD 寄存器，获得通信所需的串行时钟频率
- 通过 CKPOL 寄存器配置总线时钟在静态下的电平
- 配置 I2SCR 寄存器，设置模块的工作模式、协议格式、帧长度等
- 使能需要的中断或 DMA
- 置位 I2SEN 使能模块通信

22.3.5.2 发送流程

在 I2S 使能的情况下，软件或 DMA 写 TXBUF 将启动总线发送。数据发送时，硬件会自动根据当前发送的是左声道还是右声道数据，控制 WS 输出电平，并且置位或复位 CHSIDE 寄存器表征当前正在发送的声道。

当 I2S 总线空闲时，写入 TXBUF 的数据会被立即载入移位寄存器进行发送，同时 TXBE 中断标志寄存器置位，表示发送缓冲区空，软件可以在 TXBE 置位后写入下一笔数据，或者在使能 DMA 的情况下由 DMA 写入。

为了保证连续不间断的数据发送，软件必须保证在当前移位寄存器发送完成前，将后续 16 位数据填入 TX TXBUF。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXBE 置位并且 BUSY 清零。

22.3.5.3 接收流程

无论配置的数据帧长是多少，实际数据接收缓冲区宽度是固定的 16bit。一旦移位寄存器完成 16 位接收，数据将自动载入 RXBUF，同时置位 RXBF 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXBF 置位后读取接收缓冲寄存器，或者在使能 DMA 的情况下由 DMA 读取。

软件应确保在新的数据帧接收完成前，读取 RXBUF 中的数据。如果新的数据到来时 RXBUF 非空，则 RXCOL 错误标志寄存器将被置位。

22.3.6 I2S 从机功能

22.3.6.1 操作流程

在从机模式下，I2S 的 CK 和 WS 信号来自于外部主机，通信码率由主机决定，因此用户无需配置时钟分频寄存器。软件需要将 I2S 从机配置为发送或接收模式。

- 配置 I2S 为从机模式，配置协议格式、帧长度，选择 I2S 为发送或接收模式。
- 根据需要使能中断或 DMA
- 置位 I2SEN 寄存器

22.3.6.2 发送流程

当 I2S 从机配置为发送模式，主机在总线上发送时钟和 WS 时，从机自动开始发送数据。软件必须在通信开始前将待发数据写入 TXBUF。

CHSIDE 寄存器用于表征当前通信的声道，此寄存器在从机模式下根据主机发送的 WS 信号自动更新。

当第一个 I2S 时钟到来时，TXBUF 中的数据被载入移位寄存器开始发送，此时 TXBE 中断标志寄存器置位，软件或 DMA 可以将下一笔发送数据写入 TXBUF。

为了保证连续不间断的数据发送，软件必须保证在当前移位寄存器发送完成前，将后续 16 位数据填入 TXBUF。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXBE 置位并且 BUSY 清零。

22.3.6.3 接收流程

当 I2S 从机配置为发送模式，主机在总线上发送时钟和 WS 时，从机自动开始接收数据。一旦移位寄存器完成接收，数据将自动载入 RXBUF，同时置位 RXBF 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXBF 置位后读取接收缓冲寄存器，或者在使能 DMA 的情况下由 DMA 读取。

CHSIDE 寄存器用于表征当前通信的声道，此寄存器在从机模式下根据主机发送的 WS 信号自动更新。

软件应确保在新的数据帧接收完成前，读取 RXBUF 中的数据。如果新的数据到来时 RXBUF 非空，则 RXCOL 错误标志寄存器将被置位。

22.3.7 帧格式错误

I2S 从机模式下，如果主机发送的 WS 信号变化不符合协议规范，则 I2S 模块将置位 SERR 错误中断标志信号，表示帧格式错误。

22.3.8 DMA 支持

I2S 模块支持通过 DMA 进行数据收发。当 I2S 模块被使能后，I2S 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 I2S 外设(SPIx_TX)，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 I2S 请求，并完成 RAM 和 I2S 之间的数据搬运。

22.4 寄存器

SPI0模块基地址：0x40019000

SPI1模块基地址：0x40018C00

SPI2模块基地址：0x40010800

SPI3模块基地址：0x40014000

offset 地址	名称	符号
SPIx 寄存器 (x=0,1,2,3)		
0x00	SPIx 控制寄存器 1 （SPIx Control Register1）	SPIx_CR1
0x04	SPIx 控制寄存器 2 （SPIx Control Register2）	SPIx_CR2
0x08	SPIx 控制寄存器 3 （SPIx Control Register3）	SPIx_CR3
0x0C	SPIx 中断使能寄存器 （SPIx Interrupt Enable Register）	SPIx_IER
0x10	SPIx 中断状态寄存器 （SPIx Status Register）	SPIx_ISR
0x14	SPIx 发送数据缓冲寄存器 （SPIx Transmit Buffer）	SPIx_TXBUF
0x18	SPIx 接收数据缓冲寄存器 （SPIx Receive Buffer）	SPIx_RXBUF
0x20	I2S 配置寄存器 （SPIx I2S Control Register）	SPIx_I2SCR
0x24	I2S 预分频寄存器 （SPIx I2S Prescaler Register）	SPIx_I2SPR

22.4.1 SPI 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1 (x=0,1,2,3)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	MSPA	SSPA	MM
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	CPHOL	CPHA
位权限	R/W-00		R/W-001			R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
11	**IOSWAP**	MOSI 和 MISO 引脚交换 (IO swapping)

Bit	助记符	功能描述
		0：默认引脚顺序 1：交换引脚顺序
10	**MSPA**	Master Sampling Position Adjustment，Master 对 MISO 信号的采样位置调整，用于高速通信时补偿 PCB 走线延迟 1：采样点延迟半个 SCK 周期 0：不调整
9	**SSPA**	Slave Sending Position Adjustment，Slave MISO 发送位置调整 1：提前半个 SCK 周期发送 0：不调整
8	**MM**	Master/Slave 模式选择。 1：Master 模式 0：Slave 模式
7:6	**WAIT**	Master 模式下，每发送完一帧后加入至少(1+WAIT)个 SCK cycle 等待时间，再传输下一帧的数据。如果 SSN 由硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。
5:3	**BAUD**	Master 模式波特率配置位: 000： $f_{APBCLK}/2$ 001： $f_{APBCLK}/4$ 010： $f_{APBCLK}/8$ 011： $f_{APBCLK}/16$ 100： $f_{APBCLK}/32$ 101： $f_{APBCLK}/64$ 110： $f_{APBCLK}/128$ 111： $f_{APBCLK}/256$ 当通信正在进行的时候，不能修改这些位。
2	**LSBF**	帧格式 (LSB First) 0：先发送 MSB 1：先发送 LSB 注：当通信在进行时不能改变该位的值。
1	**CPHOL**	时钟极性选择 (Clock Polarity) 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
0	**CPHA**	时钟相位选择 (Clock Phase) 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

22.4.2 SPI 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2 (x=0,1,2,3)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN	-			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	U-0			R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**DUMMY_EN**	4 线半双工协议下是否在读操作中插入 dummy cycle (Dummy cycle Enable) 0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	--	RFU：未实现，读为 0
11	**RXO**	RXONLY 控制位，此寄存器置位时，SPI 可以连续接收，无需软件写 TXBUF (Receive Only mode) 1：启动 Master 的单接收模式 0：关闭单接收模式（收发全双工）
10:9	**DLEN**	通信数据字长配置 (Data Length) 00：8bit 01：16bit 10：24bit 11：32bit
8	**HALFDUPLEX**	通信模式选择 (Half-Duplex mode) 0：标准 SPI 模式，4 线全双工 1：DCN 模式，4 线半双工
7	**HD_RW**	半双工模式下主机读写操作配置 (Read/Write config for Half-Duplex mode) 0：4 线半双工协议下主机写入从机 1：4 线半双工协议下主机读取从机
6	**CMD8b**	半双工模式下定义 command 帧长度 (Command 8 bits) 1：command 帧固定为 8bit 0：command 帧长度由 DLEN 定义
5	**SSNM**	Master 模式下 SSN 控制模式选择 (SSN mode) 1：每发送完一帧后 Master 拉高 SSN，维持高电平时间由 WAIT 寄存器控制 0：每发送完一帧后 Master 保持 SSN 为低
4	**TXO_AC**	TXONLY 硬件自动清空的使能 (TXONLY auto-clear enable) 1：TXONLY 硬件自动清零有效，软件使能 TXO 后，等待发送完毕后，硬件清零 0：关闭 TXONLY 硬件自动清零
3	**TXO**	TXONLY 控制位 (Transmit Only mode enable) 1：启动 Master 的单发送模式 0：关闭单发送模式（收发全双工）
2	**SSN**	Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平 1：SSN 输出高电平 0：SSN 输出低电平
1	**SSNSEN**	Master 模式下，软件控制 SSN 使能 (SSN Software Enable)

Bit	助记符	功能描述
		1：Master 模式下 SSN 输出由软件控制 0：Master 模式下 SSN 输出由硬件自动控制
0	**SPIEN**	SPI 使能(SPI enable) 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

22.4.3 SPI 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3 (x=0,1,2,3)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TXBFC	RXBFC	MERRC	SERRC
位权限	U-0				W-0	W-0	W-0	W-0

Bit	助记符	功能描述
31:4	**--**	RFU：未实现，读为 0
3	**TXBFC**	Transmit Buffer Clear，软件写 1 清除发送缓存，写 0 无效
2	**RXBFC**	Receive Buffer Clear，软件写 1 清除发送缓存，写 0 无效
1	**MERRC**	Master Error Clear，软件写 1 清除 SPIx_ISR.MERR 寄存器
0	**SERRC**	Slave Error Clear，软件写 1 清除 SPIx_ISR.SERR 寄存器

22.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）

名称	SPIx_IER (x=0,1,2,3)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SSNRIE	SSNFIE	TXSEIE	ERRIE	TXBEIE	RXIE
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	SSNRIE	从机 SSN 上升沿中断使能
4	SSNFIE	从机 SSN 下降沿中断使能
3	TXSEIE	发送移位寄存器空中断使能(Transmit shift register empty interrupt enable)
2	ERRIE	SPI 错误中断使能 (Error Interrupt Enable)
1	TXBEIE	发送缓冲空中断使能 (Transmit buffer empty Interrupt Enable)
0	RXIE	接收完成中断使能 (Receive Interrupt Enable)

22.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR (x=0,1,2,3)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DCN_TX	-	RXCOL	TXCOL	BUSY
位权限	U-0			R/W-1	U-0	R/W/Dy-0	R/W/Dy-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHSIDE	MERR	SERR	SSNR	SSNF	TXSE	TXBE	RXBF
位权限	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R-1	R-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	DCN_TX	半双工模式下 (HALFDUPLEX=1)，配置在每个数据帧的最后 bit 发送的 DCN 信号电平 (Data/Command transmit config) 0：DCN=0，表示命令帧 1：DCN=1，表示数据帧 软件应在发送前设置 DCN_TX 寄存器，如果 DCN_TX=0，硬件在完成一帧发送后，自动将 DCN_TX 置 1，即默认只会发送一个命令帧，后续都是数据帧。
11	--	RFU：未实现，读为 0
10	RXCOL	接收缓存溢出，软件写 1 清零 (Receive Collision flag,write 1 to flag)
9	TXCOL	发送缓存溢出，软件写 1 清零 (Transmit Collision flag,write 1 to clear)
8	BUSY	SPI/I2S 空闲标志，只读(busy flag) 1：SPI/I2S 传输进行中 0：SPI/I2S 传输空闲 *注意：此标志在 SPI/I2S 波特率时钟下置位，当软件使能 SPI/I2S 发送后，需等待波特率时钟到来才会置位*
7	CHSIDE	声道标志寄存器，仅 I2S 模式有效，用于表示当前正在传输的

Bit	助记符	功能描述
		数据属于哪个声道 0：左声道 1：右声道
6	**MERR**	Master Error 标志(Master Error flag)，硬件置位 当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	**SERR**	Slave Error 标志(Slave Error flag)，硬件置位 SPI 模式：当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位 I2S 模式：当主机在错误的位置改变 WS 信号电平时置位
4	**SSNR**	从机接收的 SSN 信号上升沿标志位 (slave SSN rising flag)，硬件置位，软件写 1 清零 1：从机接收到 SSN 上升沿 0：从机没有接收到 SSN 上升沿 此标志仅 SPI 从机模式下有效
3	**SSNF**	从机接收的 SSN 信号下降沿标志位 (slave SSN falling flag)，硬件置位，软件写 1 清零 1：从机接收到 SSN 下降沿 0：从机没有接收到 SSN 下降沿 此标志仅 SPI 从机模式下有效
2	**TXSE**	TX Shift Register Empty 标志位 (TX Shift Register Empty flag) 1：发送移位寄存器空 0：发送移位寄存器非空
1	**TXBE**	TX Buffer Empty 标志位(TX Buffer Empty flag) 1：发送缓存空，软件写 TXBUF 清零 0：发送缓存满
0	**RXBF**	RX Buffer Full 标志位(RX Buffer Full flag) 1：接收缓存满，软件读 RXBUF 清零 0：接收缓存空

22.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF (x=0,1,2,3)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF							
位权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TXBUF							
位权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXBUF							
位权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	W-00000000							

Bit	助记符	功能描述

Bit	助记符	功能描述
31:0	**TXBUF**	SPI/I2S 发送缓存(Transmit Buffer)

22.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF (x=0,1,2,3)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RXBUF							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXBUF							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-00000000							

Bit	助记符	功能描述
31:0	**RXBUF**	SPI/I2S 接收缓存(Receive Buffer)

22.4.8 I2S 配置寄存器（SPIx_I2SCR）

名称	SPIx_I2SCR (x=0,1,2,3)							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				I2SSTD		CHLEN	CKPOL
位权限	U-0				R/W-00		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DLEN		PCMSYNC	I2SMM		DLYEN	I2SMOD	I2SEN
位权限	R/W-00		R/W-0	R/W-0		R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:10	**I2SSTD**	I2S 标准选择 00：Philips 标准 01：MSB-justified 10：LSB-justified

Bit	助记符	功能描述
		11：PCM 标准
9	**CHLEN**	声道长度（每个声道的 bit 数） 0：16bit 1：32bit 仅 DLEN=00 的情况下有效，否则 CHLEN 固定为 32bit
8	**CKPOL**	I2S 空闲时钟极性 0：CK 空闲时保持低电平 1：CK 空闲时保持高电平
7:6	**DLEN**	I2S 通信数据长度 00：16bit 01：24bit 10：32bit 11：RFU
5	**PCMSYNC**	PCM 帧同步 0：短帧同步 1：长帧同步 仅 I2SSTD=11 时有效
4:3	**I2SMM**	I2S Master Mode 00：从机接收 01：从机发送 10：主机接收 11：主机发送
2	**DLYEN**	WS 和 SD 数据延迟使能，仅 I2S 主机模式有效 0：关闭输出延迟 1：WS 和 SD 输出经过 100ns 模拟延迟
1	**I2SMOD**	外设模式选择 0：SPI 模式 1：I2S 模式
0	**I2SEN**	I2S 使能 1：使能 I2S 模块 0：关闭 I2S 模块 此寄存器仅在 I2SMOD=1 的情况下有效

22.4.9 I2S 预分频寄存器（SPIx_I2SPR）

名称	SPIx_I2SPR (x=0,1,2,3)							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						MCKOE	ODD
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CLKDIV							

位权限	R/W-0000 0000

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	**MCKOE**	Master 时钟输出使能（仅主机模式有效） 0：禁止 MCK 输出 1：使能 MCK 输出
8	**ODD**	波特率预分频小数部分 0：实际分频系数是 CLKDIV*2 1：实际分频系数是 CLKDIV*2 + 1
7:0	**CLKDIV**	I2S 波特率分频寄存器，禁止使用 0 和 1 此寄存器定义 CK 高低电平宽度（单位是 APBCLK 周期），CK 频率是 F_{APBCLK}/(CLKDIV*2)

23 直接存储访问控制器（DMA）

23.1 概述

- 7通道外设PDMA，支持Peripherals<>RAM传输
- 1通道TSI电容触摸PDMA，支持14个通道指针
- 1通道存储器MDMA，支持Flash<>RAM传输
- 外设DMA传输由外设请求触发，DMA工作期间不影响CPU运行
- 外设通道最大传输长度8192，支持byte/half-word/word传输
- Flash->RAM通道最大传输长度4096，只支持word传输
- 支持Flash连续编程（RAM->Flash），需要预先进行擦除，一次编程固定为256字节
- RAM指针递增、递减
- 可产生半程中断和全程中断
- 通道优先级可配置（4级优先级）

23.2 工作原理

外设 DMA 为 Peripheral<>RAM 通道，采用外设请求触发方式进行数据传输，每个外设通道都可以支持外设->RAM 或者 RAM->外设的数据传输，并且根据目标外设类型的不同，自适应选择 byte/half-word/word 传输方式。DMA 作为 Master，在收到 request 后将发起 AHB transactions 进行数据操作，外设目标地址根据通道接入选择自动定位，RAM 目标地址则根据寄存器配置定位。

TSI DMA 为电容按键触摸控制器与 RAM 之间的专用通道，可以支持最大 14 个扫描通道，并将扫描数据分散到 RAM 不同地址。

每个 channel 可以从多个外设中选择一个作为 source 或 destination，同时软件可以设置通道优先级，当两个通道同时要访问 RAM 时，由优先级决定谁先访问，另一个通道将被挂起，直到优先通道访问完毕。

外设请求可以是准备发送（RAM/Flash->Peripheral）或接收完成（Peripheral->RAM），数据传输通过 AHB 总线完成，当 DMA 访问外设时，CPU 对同一个外设的访问将引起冲突，哪个 Master 访问被挂起取决于 BusMatrix 设置的仲裁优先级。这里需要注意的是，由于大部分外设都被挂在 APB 总线上，APB 映射到 AHB 仅为一个 slave，因此当 DMA 访问 APB 中任意外设时，CPU 即使访问 APB 下的其他外设，也同样会引起总线仲裁。通过 DIR 寄存器可以配置每个通道的传输方向，软件必须保证传输方向配置与实际挂载到这个通道上的外设请求相一致。比如通道 1 当前挂载的外设请求是 UART0 接收，则必须将 DIR 寄存器配置为 0（数据从外设读出，写入 RAM），每次 UART0 接收完一帧数据，将发送 RXD0 请求给 DMA，DMA 响应请求后，从 UART0 接收缓存寄存器读取数据，如果 DIR 被错误的配置为 1，则 DMA 对 UART0 接收缓存寄存器的写操作将被 UART0 忽略。

软件可设置 DMA 的存储器指针，用于配置 DMA 传输的起始地址，可以选择指针递增或递减方式。另有 TSIZE 寄存器配置传输次数，根据起始地址和传输次数，计算得到终止地址，当存储器指针指向终止地址时，本次传输结束，关闭通道。

当 channel 被使能后，DMA 就准备好接受通道所选中的外设请求。当配置传输长度一半的字节被传输后，一个 半程中断置位；当配置传输长度全部完成后，全程中断置位。上述中断都可以被相应的中断使能寄存器屏蔽。

在 DMA 一个完整 transfer block 完成之前，软件随时可以关闭 channel 使能，此时 DMA 将被挂起，如果软件此后重新使能通道，则 DMA 继续执行之前挂起的操作。

23.3 结构框图

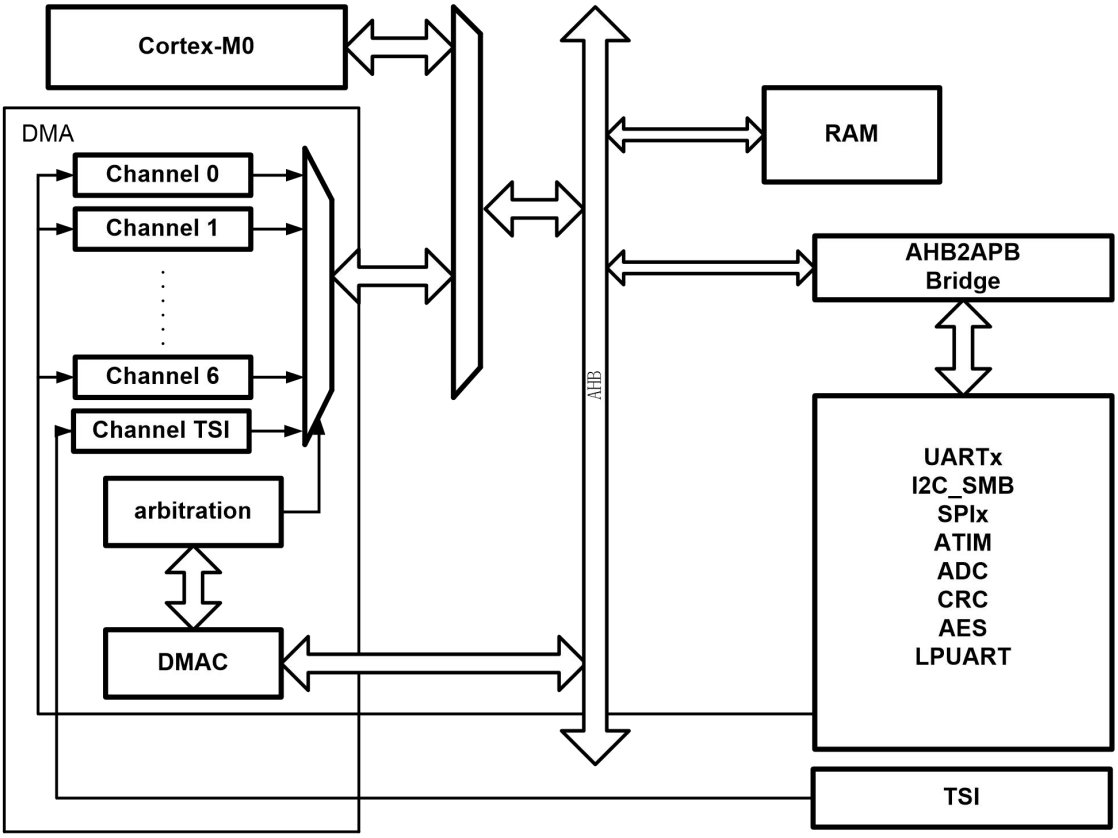


图 23-1 DMA 结构框图

23.4 访问带宽

DMA 外设通道支持字节/半字/字访问，每个通道都可以通过通道控制寄存器中的 BDW 位来配置传输带宽。

23.5 通道控制

23.5.1 DMA 请求映射

DMA 共有 7 个优先级可配的外设通道，每个通道可接受 8 个请求响应，根据每个通道的配置寄存器选择其中一个请求送入通道控制器，通道控制器根据各个通道的 busy 状态和优先级选择其中一个通道请求进行响应处理，外设请求映射如下。

编号	外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
0	***ADC***	ADC				ADC		ADC
1	***SPI0***	SPI0_RX	SPI0_RX	SPI0_TX				
2	***SPI1***				SPI1_RX	SPI1_TX	SPI1_RX	SPI1_TX
3	***SPI2***			SPI2_RX		SPI2_TX	SPI2_RX	SPI2_TX
4	***SPI3***	SPI3_RX	SPI3_TX					
5	***UART0***		RXD0	TXD0	RXD0	TXD0		
6	***UART1***				RXD1	TXD1	RXD1	TXD1
7	***UART2***	RXD2	TXD2		TXD2		RXD2	
8	***UART3***			RXD3			RXD3	TXD3
9	***UART4***			RXD4	TXD4			
10	***UART5***			RXD5	TXD5	RXD5		TXD5
11	***RFU***	RFU	RFU		RFU		RFU	
12	***I2C_SMB0***		I2CSMB0_RX	I2CSMB0_TX			I2CSMB0_RX	I2CSMB0_TX
13	***I2C_SMB1***					I2CSMB1_RX		I2CSMB1_TX
14	***AES***	AES_IN	AES_OUT					
15	***CRC***	CRC					CRC	
16	***ATIM***	ATIM_CH1	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV		
		8	8	8	8	8	8	8

表 23-1PDMA 通道映射列表

每个通道通过DMA_CHxCR.SSEL寄存器选择数据源

SSEL	通道0	通道1	通道2	通道3	通道4	通道5	通道6
000	ADC	SPI0_RX	SPI0_TX	SPI1_RX	ADC	SPI1_RX	ADC
001	SPI0_RX	SPI3_TX	SPI2_RX	RXD0	SPI1_TX	SPI2_RX	SPI1_TX
010	SPI3_RX	RXD0	TXD0	RXD1	SPI2_TX	RXD1	SPI2_TX
011	RXD2	TXD2	RXD3	TXD2	TXD0	RXD2	TXD1
100	RFU	RFU	RXD4	TXD4	TXD1	RXD3	TXD3
101	AES_IN	I2CSMB0_RX	RXD5	TXD5	RXD5	RFU	TXD5
110	CRC	AES_OUT	I2CSMB0_TX	RFU	I2CSMB1_RX	I2CSMB0_RX	I2CSMB0_TX
111	ATIM_CH1	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV	CRC	I2CSMB1_TX

表 23-2PDMA 各通道源选择

外设请求映射通过CHxSSEL寄存器配置，上表中从上到下分别表示CHxSSEL=0~7情况下有效的外设请求信号。比如针对通道0，当CH0SSEL=2时，被选中的外设请求是SPI3_RX。

23.5.2 通道优先级

DMA 总共有 7 个外设通道，每个通道的优先级别可以通过寄存器配置为：very high,high,low,very low。当多个通道配置为相同优先级别时，通道序号越大，优先级别越低。

DMA 每搬运完一次数据都会重新进行通道请求选择，假设通道 0 传输长度为 3，通道 1 传输长度为 2。当通道 0 完成第二次传输准备进行第三次数据搬运时，通道 1 请求响应置起，这时通道控制器根据通道优先级切换至通道 1 数据搬运，直至通道 1 数据全部搬运完成，通道寄存器再切换回通道 0 完成剩下的数据搬运。

23.5.3 传输方向定义

在 DMA 通道定义规则中，RX 表示 DMA 从外设读取数据，写入 RAM，TX 表示 DMA 从 RAM/Flash 读取数据，写入外设。

软件在配置每个通道的外设分配之后，还需要配置 CHx_DIR 寄存器设定通道传输方向，错误的方向设置会导致 DMA 无法正常工作。

23.5.4 循环模式

外设 DMA 通道支持循环模式（Circular mode）。循环模式下，当 CHxTSIZE 寄存器定义的传输长度完成后，DMA 不会自动停止，而是返回 RAM 指针寄存器定义的起始地址，继续传输。DMA 的半程中断和全程中断还是会正常置起，DMA 不会终止传输，直到软件关闭通道。

通过置位 CHxCR.CIRC 寄存器使能循环模式。

存储 DMA 通道不支持循环模式。

23.5.5 TSI 通道

TSI 是一个专用 DMA 通道，用于将 TSI 扫描数据发送给 RAM 存储。TSI 通道只有一个源头，因此不需要数据源选择，但是其目标地址支持 14 个指针，用于将不同扫描通道的数据发送到 RAM 不同地址段。TSIZE 指定了扫描数据传输的长度，即每个扫描通道在 RAM 中的存储区深度（最大 256 次传输），配合循环模式可以实现在固定区域的循环存储。

TSI 传输带宽为固定的半字（16bit），存储器指针范围仅包含 RAM 物理地址，并且起始地址必须 word 对齐，不支持将指针起始地址指向高半字地址。

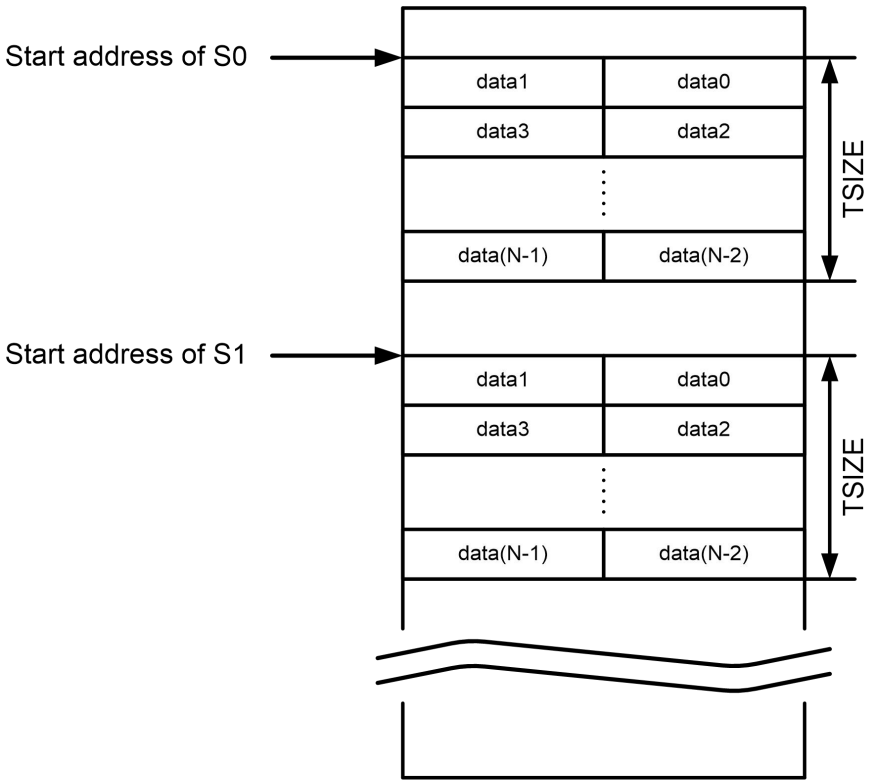


图 23-2 TSI 数据存储示意图

23.6 寄存器

模块起始地址：0x4000_0400

offset 地址	名称	符号
0x00	DMA 全局控制寄存器（DMA Global Control Register）	DMA_GCR
0x04	通道 0 控制寄存器（Channel 0 Control Register）	DMA_CH0CR
0x08	通道 0 存储器指针寄存器（Channel 0 Memory Address Register）	DMA_CH0MAR
0x0C	通道 1 控制寄存器（Channel 1 Control Register）	DMA_CH1CR
0x10	通道 1 存储器指针寄存器（Channel 1 Memory Address Register）	DMA_CH1MAR
0x14	通道 2 控制寄存器（Channel 2 Control Register）	DMA_CH2CR
0x18	通道 2 存储器指针寄存器（Channel 2 Memory Address Register）	DMA_CH2MAR
0x1C	通道 3 控制寄存器（Channel 3 Control Register）	DMA_CH3CR
0x20	通道 3 存储器指针寄存器（Channel 3 Memory Address Register）	DMA_CH3MAR
0x24	通道 4 控制寄存器（Channel 4 Control Register）	DMA_CH4CR
0x28	通道 4 存储器指针寄存器（Channel 4 Memory Address Register）	DMA_CH4MAR
0x2C	通道 5 控制寄存器（Channel 5 Control Register）	DMA_CH5CR
0x30	通道 5 存储器指针寄存器（Channel 5 Memory Address Register）	DMA_CH5MAR
0x34	通道 6 控制寄存器（Channel 6 Control Register）	DMA_CH6CR
0x38	通道 6 存储器指针寄存器（Channel 6 Memory Address Register）	DMA_CH6MAR
0x3C	通道 7 控制寄存器（Channel 11 Control Register）	DMA_CH7CR
0x40	通道 7 Flash 指针寄存器（Channel 11 Flash Address Register）	DMA_CH7FLSAD
0x44	通道 7 RAM 指针寄存器（Channel 11 RAM Address Register）	DMA_CH7RAMAD
0x48	DMA 中断状态标志寄存器（DMA Interrupt Status Register）	DMA_ISR
0x68	TSI 通道控制寄存器（DMA TSI Control Register）	DMA_TSICR
0x6C	TSI 扫描通道 0 存储器指针寄存器（DMA TSI Channle0 Memory Address Register）	DMA_TSI0MAR
0x70	TSI 扫描通道 1 存储器指针寄存器（DMA TSI Channle1 Memory Address	DMA_TSI1MAR

offset 地址	名称	符号
	Register）	
0x74	TSI 扫描通道 2 存储器指针寄存器（DMA TSI Channle2 Memory Address Register）	DMA_TSI2MAR
0x78	TSI 扫描通道 3 存储器指针寄存器（DMA TSI Channle3 Memory Address Register）	DMA_TSI3MAR
0x7C	TSI 扫描通道 4 存储器指针寄存器（DMA TSI Channle4 Memory Address Register）	DMA_TSI4MAR
0x80	TSI 扫描通道 5 存储器指针寄存器（DMA TSI Channle5 Memory Address Register）	DMA_TSI5MAR
0x84	TSI 扫描通道 6 存储器指针寄存器（DMA TSI Channle6 Memory Address Register）	DMA_TSI6MAR
0x88	TSI 扫描通道 7 存储器指针寄存器（DMA TSI Channle7 Memory Address Register）	DMA_TSI7MAR
0x8C	TSI 扫描通道 8 存储器指针寄存器（DMA TSI Channle8 Memory Address Register）	DMA_TSI8MAR
0x90	TSI 扫描通道 9 存储器指针寄存器（DMA TSI Channle9 Memory Address Register）	DMA_TSI9MAR
0x94	TSI 扫描通道 10 存储器指针寄存器（DMA TSI Channle10 Memory Address Register）	DMA_TSI10MAR
0x98	TSI 扫描通道 11 存储器指针寄存器（DMA TSI Channle11 Memory Address Register）	DMA_TSI11MAR
0x9C	TSI 扫描通道 12 存储器指针寄存器（DMA TSI Channle12 Memory Address Register）	DMA_TSI12MAR
0xA0	TSI 扫描通道 13 存储器指针寄存器（DMA TSI Channle13 Memory Address Register）	DMA_TSI13MAR
0xA4	TSI 接近扫描通道存储器指针寄存器（DMA TSI Channle Prox Memory Address Register）	DMA_TSIPROXMAR

23.6.1 DMA 全局控制寄存器（DMA_GCR）

名称	DMA_GCR							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	DMA_GCR							
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						ADDRERR_EN	EN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	**ADDRERR_EN**	DMA 错误地址中断使能 (DMA address error interrupt enable) 1：允许错误地址中断 0：禁止错误地址中断
0	**EN**	DMA 全局使能 (DMA enable) 1：DMA 使能 0：DMA 关闭

23.6.2 通道 x 控制寄存器（DMA_CHxCR）

名称	DMA_CHxCR(x=0~6)							
Offset	0x04 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			TSIZE[12:8]				
位权限	U-0			R/W-0 0000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		INC	SSEL		
位权限	U-0		R/W-00		R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SUSP	DIR	BDW		CIRC	FTIE	HTIE	EN
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:29	-	RFU：未实现，读为 0
28:16	**TSIZE**	Channelx 传输长度，最大 8192 次传输，实际传输长度是 TSIZE+1 *注意：禁止将 TSIZE 配置为 0*
15:14	-	RFU：未实现，读为 0
13:12	**PRI**	Channelx 优先级(Channels Priority) 00：Low 01：Medium 10：High 11：Very High
11	**INC**	RAM 地址增减设置 (Channelx Ram address Incremental)

Bit	助记符	功能描述
		1：RAM 地址递增 0：RAM 地址递减
10:8	**SSEL**	Channelx 外设请求映射 (Channelx request Source Select) 每个通道可以接受 8 个外设请求，外设请求的映射参见表 23-2PDMA 各通道源选择
7	**SUSP**	通道暂停控制(Suspend) 1：暂停本通道传输，内部传输状态不复位 0：恢复通道传输 当软件置位 SUSP 寄存器时，数字电路在完成当前传输后暂停 DMA 通道工作，仅在 EN=1 的情况下有效 SUSP 清零后 DMA 通道恢复传输
6	**DIR**	通道传输方向 (Direction) 0：从外设读取数据写入 RAM 1：从 RAM 读取数据写入外设
5:4	**BDW**	传输带宽设置(Bandwidth) 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	**CIRC**	循环缓冲模式 (Circular mode enable) 0：关闭循环模式 1：使能循环模式
2	**FTIE**	Channelx 传输完成中断使能 (Channelx Finished-Transfer Interrupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**HTIE**	Channelx 半程传输完成中断使能 (Channelx Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channelx 使能 (Channelx Enable) 1：启动通道 0 0：关闭通道 0

23.6.3 通道 x 存储器指针寄存器（DMA_CHxMAD）

名称	DMA_CHxMAD(x=0~6)							
Offset	0x08 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MEMAD[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MEMAD[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MEMAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD[7:0]							

名称	DMA_CHxMAD(x=0~6)
位权限	R/W-0000 0000

位号	助记符	功能描述
31:0	**MEMAD**	Channelx 存储器指针地址，DMA 传输启动前软件向此寄存器写入存储器目标地址。(Channel x Memory Address pointer) 当指针指向空地址时，DMA 访问将触发 hardfault 当指针指向 Flash 时，禁止向 Flash 写入数据。 软件可以查询当前 DMA 传输的目标存储器地址。

23.6.4 通道 7 控制寄存器（DMA_CH7CR）

名称	DMA_CH7CR							
Offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				TSIZE[11:8]			
位权限	U-0				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		-	DIR	RI	FI
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					FTIE	HTIE	EN
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:28	**-**	RFU：未实现，读为 0
27:16	**TSIZE**	Channel7 传输长度 (Channel 7 Transfer Size)，1-4096 次传输，仅在 Flash->RAM 传输时有效，RAM->Flash 传输为固定长度 64 次传输
15:14	**-**	RFU：未实现，读为 0
13:12	**PRI**	Channel7 优先级 (Channel 7 Priority) 00：Low 01：Medium 10：High 11：Very High
11	**-**	RFU：未实现，读为 0
10	**DIR**	Channel7 传输方向 (Channel 7 Direction) 1：Flash->RAM 传输 0：RAM->Flash 传输
9	**RI**	Channel7 RAM 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Ram Incremental) 1：RAM 地址递增 0：RAM 地址递减
8	**FI**	Channel7 Flash 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Flash Incremental)

位号	助记符	功能描述
		1：Flash 地址递增 0：Flash 地址递减
7:3	-	RFU：未实现，读为 0
2	**FTIE**	Channel7 传输完成中断使能 (Channel 7 Finished-Transfer Interupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**HTIE**	Channel7 半程传输完成中断使能 (Channel 7 Half-Transfer Interrupt Enable) 1：使能半程中断 0：关闭半程中断
0	**EN**	Channel7 使能 (Channle 7 Enable) 1：启动通道 7 0：关闭通道 7

23.6.5 通道 7 Flash 指针寄存器（DMA_CH7FLSAD）

名称	DMA_CH7FLSAD							
Offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DFLS	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							FLSAD[16]
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FLSAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FLSAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**DFLS**	Flash 指针对象配置 0：Flash 地址指针指向 code flash 1：Flash 地址指针指向 data flash
30:17	-	RFU：未实现，读为 0
16:0	**FLSAD**	Channel7 Flash 指针地址（word 地址），DMA 传输启动前软件向此寄存器写入 Flash 目标地址，DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 Flash Address pointer) 软件可以查询当前 DMA 传输的目标 Flash 地址 此寄存器低位（bit5-0）仅在 Flash->RAM 传输中有效，RAM->Flash 传输中默认对齐 Flash 的 half-sector 起始地址

23.6.6 通道 7 RAM 指针寄存器（DMA_CH7RAMAD）

名称	DMA_CH7RAMAD

名称	DMA_CH7RAMAD							
Offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		RAMAD[13:8]					
位权限	U-0		R/W-00 0000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:14	-	RFU：未实现，读为 0
13:0	**RAMAD**	Channel7 RAM 字指针地址，DMA 传输启动前软件向此寄存器写入 RAM 目标地址（word 地址），DMA 启动后此寄存器随 DMA 传输自增或自减 (Channel 7 RAM Address pointer) 软件可以查询当前 DMA 传输的目标 RAM 地址

23.6.7 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
Offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							ADDRERR
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHFT[7:0]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHHT[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:17	-	RFU：未实现，读为 0
16	**ADDRERR**	DMA 传输地址错误标志，硬件置位，软件写 1 清零 (DMA address error flag,write 1 to clear)
15:8	**CHFT**	DMA 通道 x 传输完成标志(DMA channel Finished-Transfer Flag,write 1 to clear)，硬件置位，软件写 1 清零 1：对应通道传输完成 0：对应通道传输未完成

位号	助记符	功能描述
7:0	**CHHT**	DMA 通道 x 传输半程标志 (DMA channel Half-Transfer Flag,write 1 to clear)，硬件置位，软件写 1 清零

23.6.8 TSI 通道控制寄存器（DMA_TSICR）

名称	DMA_TSICR							
Offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		PRI		INC	-		
位权限	U-0		R/W-00		R/W-0	U-0		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				CIRC	FTIE	-	EN
位权限	U-0				R/W-0	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31:14	**-**	RFU：未实现，读为 0
13:12	**PRI**	TSI Channel 优先级 (TSI Channel Priority) 00：Low 01：Medium 10：High 11：Very High
11	**INC**	RAM 地址增减设置 (TSI Channel Ram address Incremental) 1：RAM 地址递增 0：RAM 地址递减
10:4	**-**	RFU：未实现，读为 0
3	**CIRC**	循环缓冲模式 (Circular mode enable) 0：关闭循环模式 1：使能循环模式
2	**FTIE**	TSI Channel 传输完成中断使能 (TSI Channel Finished-Transfer Interrupt Enable) 1：使能传输完成中断 0：关闭传输完成中断
1	**-**	RFU：未实现，读为 0
0	**EN**	TSI Channel 使能 (TSI Channel Enable) 1：启动通道 0：关闭通道

23.6.9 TSI 通道 x 存储器指针寄存器（DMA_TSIxMAR）

名称	DMA_TSIxMAR (x=0~13)
Offset	0x6C + x*0x4

名称	DMA_TSIxMAR (x=0~13)							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			MEMAD[12:8]				
位权限	U-0			R/W-0000 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:0	**MEMAD**	TSI 扫描通道 x 存储器指针，寻址 8K word RAM 空间，总是对齐 word 地址

23.6.10 TSI 接近检测通道存储器指针寄存器（DMA_TSIPROXMAR）

名称	DMA_TSIPROXMAR							
Offset	0x9C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			MEMAD[12:8]				
位权限	U-0			R/W-0000 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:0	**MEMAD**	TSI 接近检测通道存储器指针，寻址 8K word RAM 空间，总是对齐 word 地址

24 循环冗余校验（CRC）

24.1 概述

循环冗余校验(Cyclic Redundancy Check，CRC)是最为常用的计算机和仪表数据通信的校验方法，FM33FR0中CRC计算单元为完全独立模块，通过软件控制可进行7816、I2C、UART和SPI模块有串行数据流接口的收发CRC计算和校验。

此外，CRC也可进行Flash内容的完整性校验。通过结合DMA，可以实时计算Flash中程序内容的CRC结果，并生成一个完整性签名，与程序一同保存在Flash中。通过校验这个CRC签名，可以验证Flash内容是否正确、完整。

- 支持7/8/16/32位CRC，支持任意多项式
- 初值可设置
- 支持输入输出数据顺序自动调整（以字节、半字、或全字为单位）
- 支持对输出结果异或

24.2 软件配置过程

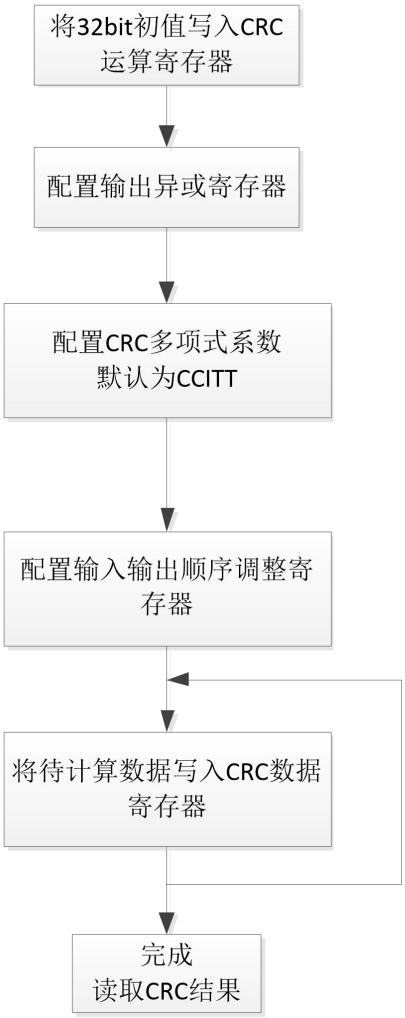


图 24-1CRC 运算流程图

CRC 配置及计算流程如下：

- CRC开始计算的时候，配置运算移位寄存器中的初始值，范围是0x0000_0000~0xFFFF_FFFF。
- 配置输出异或寄存器CRC_XOR
- 软件需配置好输入RFLTIN处理使能；输出RFLTO和XOR处理使能
- 软件将需要计算CRC码的数据放入数据寄存器(CRC_DR)，然后自动开始计算逐次移位
- 计算完毕后，结果数据回写到数据寄存器，软件根据当前计算状态BUSY位来判断是否能取结果
- 若多项式为7bit多项式则结果为CRC_DR[6:0]，若多项式为8bit多项式则结果为CRC[7:0]，若多项式为16bit多项式则结果为CRC_DR[15:0]，若多项式为32bit多项式则结果为CRC_DR[31:0]；
- 计算完前一次CRC后，数据寄存器中会保留前一次结果，作为后续数据的移位寄存器初始值。在多次连续触发CRC计算后，软件最终读取的是累积计算的完整数据的CRC值。

24.3 Golden 数据

提供 Golden 数据表格供应用中测试及校验使用。

多项式	输入序列	初始值(16 进制)		
		全 0	全 F	6363
		CRC 计算结果（16 进制）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96
CRC-16	5A5A	5DD9	DDD4	9696
	11223344	7D35	7D11	4698
CRC-CCITT	5A5A	1ACB	07C4	1877
	11223344	DD33	59F3	DD06

24.4 DMA 接口

CRC与DMA之间通道为单向的（RAM->CRC）。CRC模块可以通过DMA模块读取并校验RAM数据，其工作流程如图所示。CRC向DMA发起请求，DMA接收请求后，读取RAM并将数据写入CRC模块的CRCDR寄存器中。CRC模块接收到数据后，撤销DMA请求并开始计算校验值，校验完成后，CRC模块重新置起DMA请求。

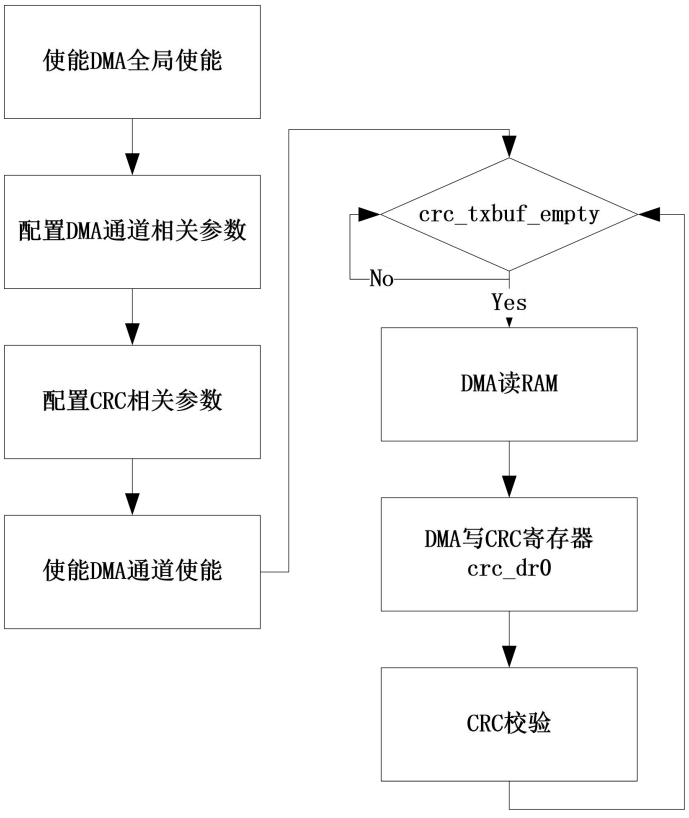


图 24-2 使用 DMA 对 RAM 中的数据进行 CRC 运算

24.5 Flash 数据完整性校验

如果需要CRC进行Flash内容的完整性校验，解决方案是使用DMA，将存储器指针直接指向flash，DMA可以直接将DMA数据搬运到CRC模块进行校验。

24.6 寄存器

模块起始地址：0x4001_8000

offset 地址	名称	符号
0x00000000	CRC 数据寄存器（CRC Data Register）	CRC_DR
0x00000004	CRC 控制状态寄存器（CRC Control Register）	CRC_CR
0x00000008	CRC LFSR 寄存器（CRC Linear Feedback Shift Register）	CRC_LFSR
0x0000000C	CRC 输出异或寄存器（CRC output XOR Register）	CRC_XOR
0x0000001C	CRC 多项式寄存器（CRC Polynominal Register）	CRC_POLY

24.6.1 CRC 数据寄存器（CRC_DR）

名称	CRC_DR							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**DR**	CRCDR 用于作为数据输入寄存器，并且在运算结束后保存 CRC 计算结果。(CRC Data Register) 作为输入时：若 word 操作使能，则对 CRCDR[31:0]进行计算，共 4 次 byte 运算（由低到高）；否则对 CRCDR[7:0]进行计算，共 1 次 byte 运算。 保存结果时：若为 7 位多项式结果保存在 CRCDR[6:0]，若为 8 位多项式结果保存在 CRCDR[7:0]，若为 16 位多项式结果保存在 CRCDR[15:0]，若为 32 位多项式结果保存在 CRCDR[31:0]。

24.6.2 CRC 控制状态寄存器（CRC_CR）

名称	CRC_CR							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OPWD	-
位权限	U-0						R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-01	

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	**OPWD**	WORD 操作使能 (Operation by Word) 0：字节操作，CRC 计算仅针对 CRCDR 最低字节进行 1：字操作，CRC 计算针对 CRCDR 全部 4 字节进行
8	-	RFU：未实现，读为 0
7:6	**RFLTIN**	CRC 输入反转控制 (Reflected Input) 00：输入不反转 01：输入按字节反转 10：输入按半字反转 11：输入按字反转 例如：计算数据为 0x11223344， 如果 RFLTIN==01，则将数据变为 0x8844CC22，再进行计算 如果 RFLTIN==10，则将数据变为 0x448822CC，再进行计算 如果 RFLTIN==11，则将数据变为 0x22CC4488，再进行计算
5	**RFLTO**	CRC 输出反转控制 (Reflected Output) 0：输入不反转 1：输入按字节反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 0x1234，则输出的结果为 0x2C48 如果 RFLTO==0，则直接输出 0x1234 注意：此结果不一定为最终输出结果，还需要看 XOR 是否为 1，详见本寄存器 bit2 说明
4	**RES**	CRC 结果标志位，只读 (Result Flag) 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 (Busy) 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 (Output XORed with CRC_XOR register enable) 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**SEL**	CRC 多项式位宽选择 (Polynomial width Selection) 00：32 位 01：16 位 10：8 位 11：7 位

24.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR[31:24]							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR[23:16]							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR[15:8]							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:0	**LFSR**	CRC 线性反馈移位寄存器 (Linear Feedback Shift Register) 运算开始前可以由软件写入 CRC 初始值

24.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	XOR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	XOR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XOR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XOR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**XOR**	CRC 运算结果异或寄存器 (eXclusive OR) 当 CRC_CR.XOR 为 1 时，CRC 结果输出前将异或此寄存器的数据。

24.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
Offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	POLY[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	POLY[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	POLY[15:8]							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY[7:0]							
位权限	R/W-0010 0001							

位号	助记符	功能描述
31:0	**POLY**	CRC 运算多项式系数（CRC Polynominals）

25 高级定时器（ATIM）

25.1 概述

FM33FR0包含一个高级定时器。

高级定时器包含一个16bit自动重载计数器及一个可编程预分频器。

高级定时器可以支持多种应用，包括如捕捉、输出比较、PWM、带死区插入的互补PWM。

25.2 主要特性

- 16bit向上、向下、双向自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- 4个独立通道可用于输入捕捉、输出比较、PWM、单脉冲输出
- 可编程死区插入的互补输出
- 重复计数器，支持定时器多个循环后更新状态
- 两路刹车引脚输入、比较器刹车、SVD刹车，刹车信号滤波和极性选择，刹车信号组合配置
- 支持在以下事件发生时产生中断或DMA事件
  - 计数器上/下溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕捉
  - 输出比较
  - 刹车输入
- 支持增量正交编码器和霍尔传感器

术语：

- OCxREF信号为高称为有效电平（active），为低称为无效电平（inactive）
- 空闲模式（IDLE mode）：相对运行模式而言，指发生刹车事件时MOE=0
- 输出禁止（output disables）：GPIO输出使能关闭，不受TIMER驱动
- Off-state：GPIO输出使能打开，但是TIMER输出无效状态
- 无效状态（inactive state）：互补通道中的一路或两路输出无效电平时的状态

25.3 结构框图

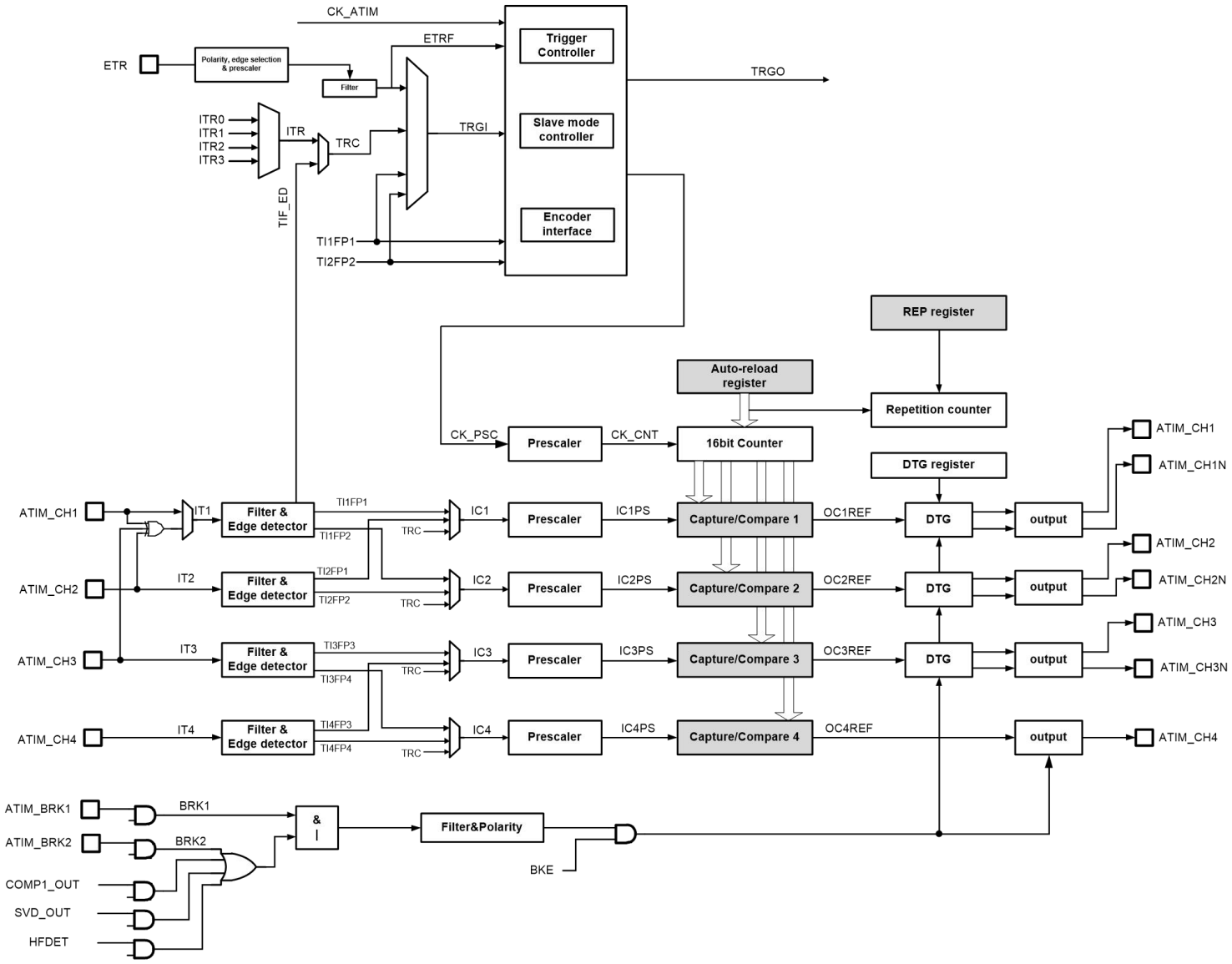


图 25-1 高级定时器结构框图

25.4 功能描述

25.4.1 定时单元

高级定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（ATIM_CNT）
- 预分频寄存器（ATIM_PSC）
- 自动重载寄存器（ATIM_ARR）
- 重复计数寄存器（ATIM_RCR）

ARR包含预装载功能，该功能通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=0时，对ARR寄存器执行写入，写入数据将直接传入到影子寄存器；当ARPE=1时，对ARR寄存器执行写入的数据在update event（ATIM_CNT上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发ARR更新（UEV）。

ATIM_CNT工作时钟由ATIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

ATIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际不改写影子寄存器，只有当新的update event到来时，才会从PSC更新至影子寄存器。因此在CNT计数过程中，软件可以实时改写PSC，而新的预分频比将在下一更新事件发生时被采用。

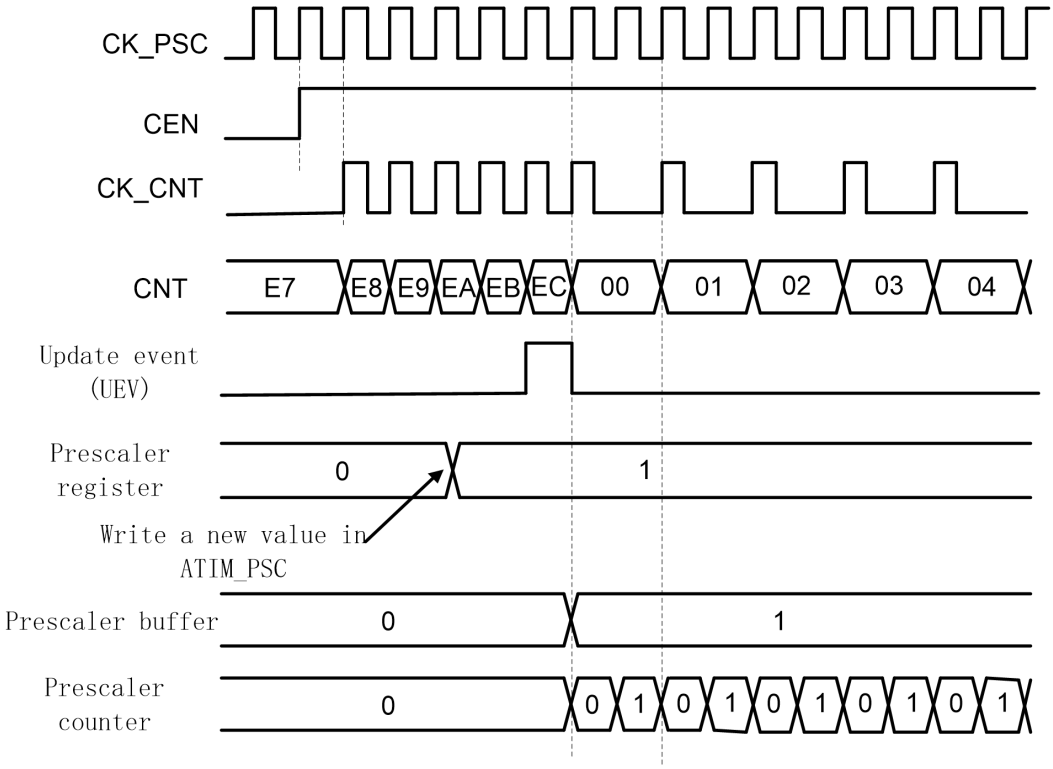


图 25-2 预分频从 1 变为 2 的波形

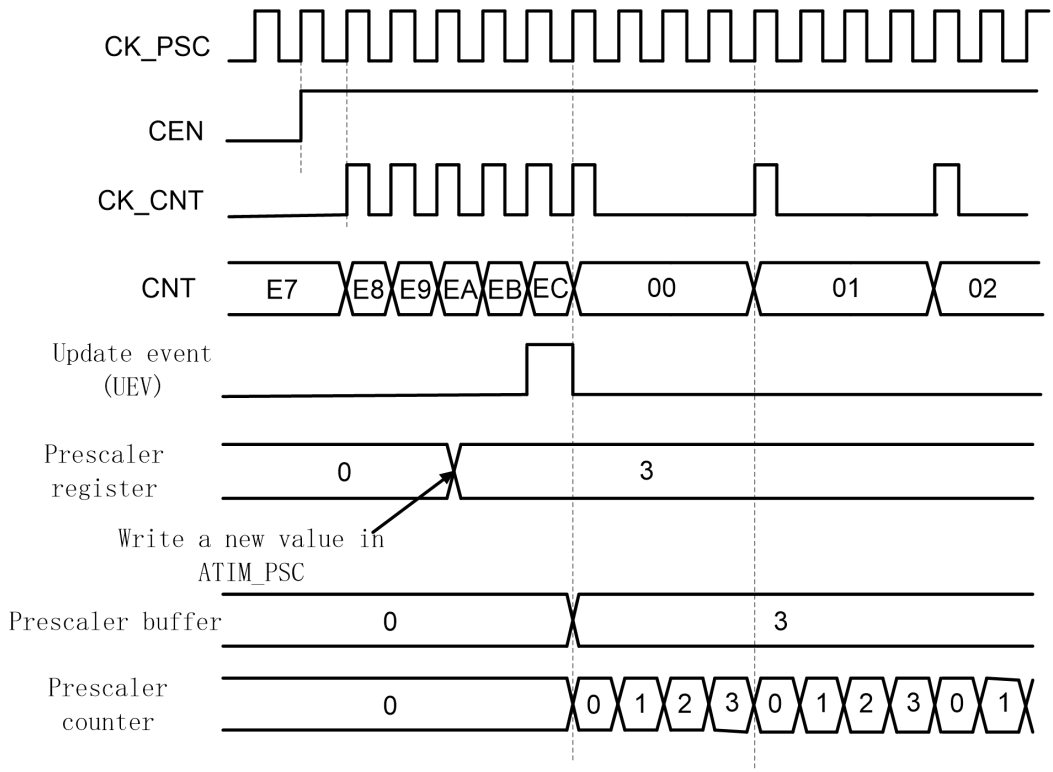


图 25-3 预分频从 1 变为 4 的波形

25.4.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

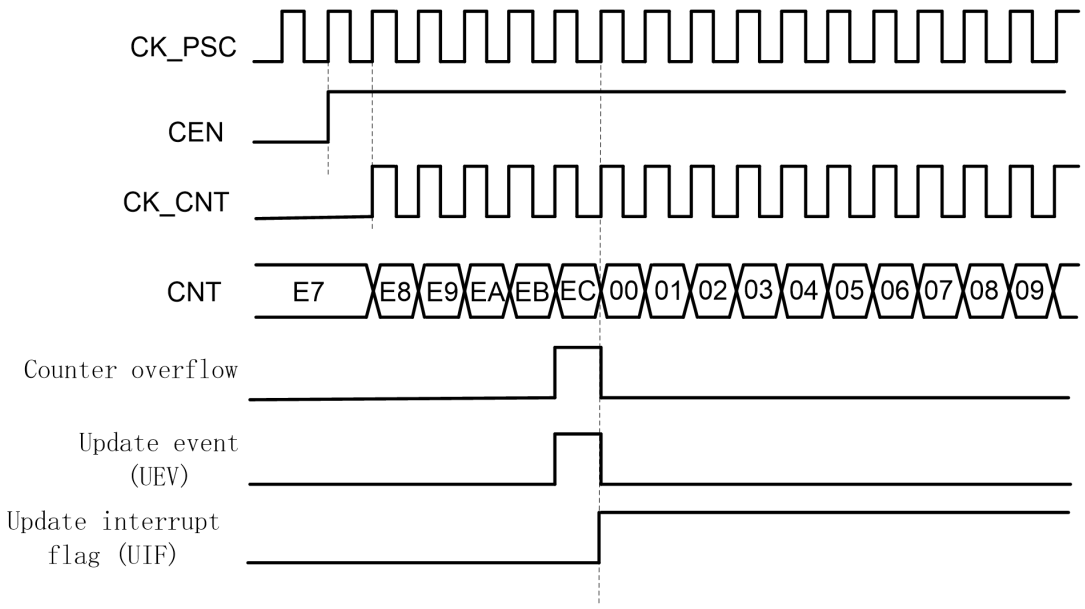


图 25-4 向上计数波形，内部时钟不分频

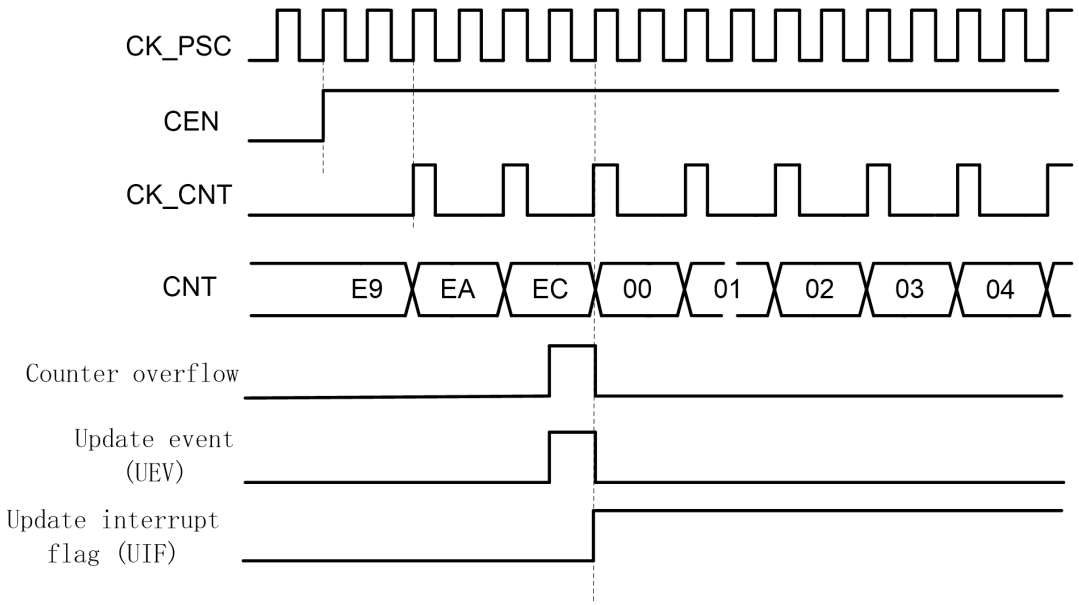


图 25-5 向上计数波形，内部时钟 2 分频

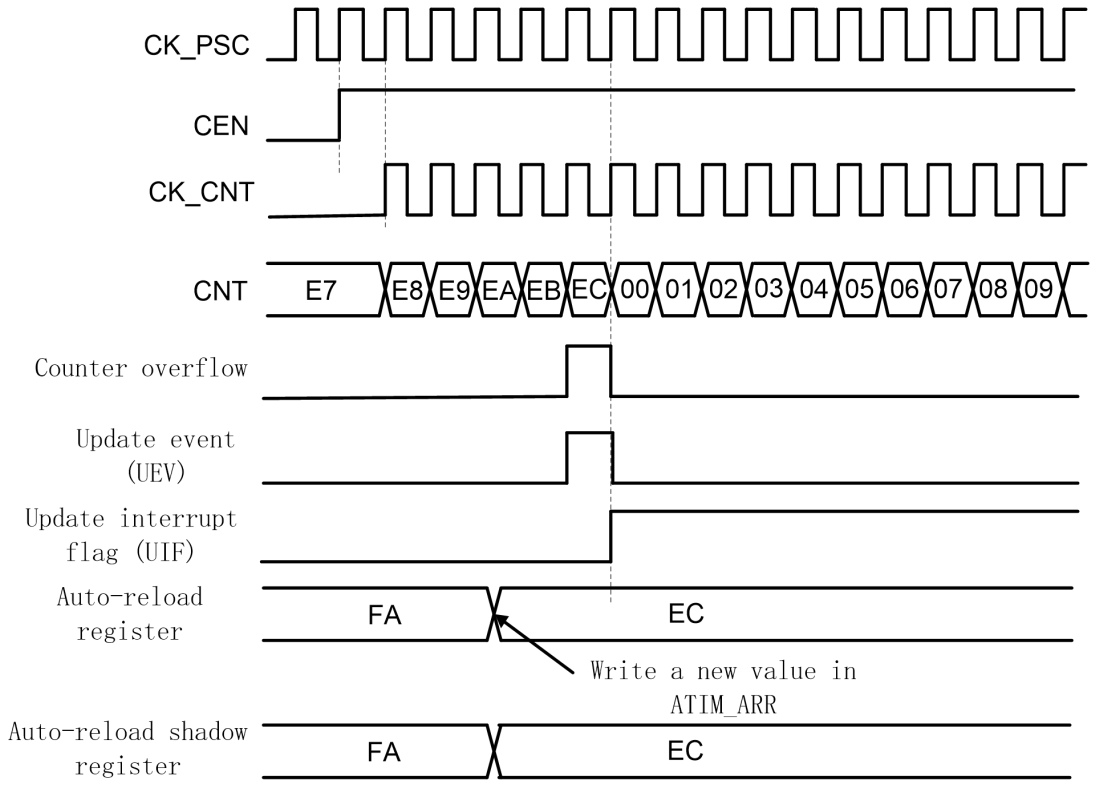


图 25-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

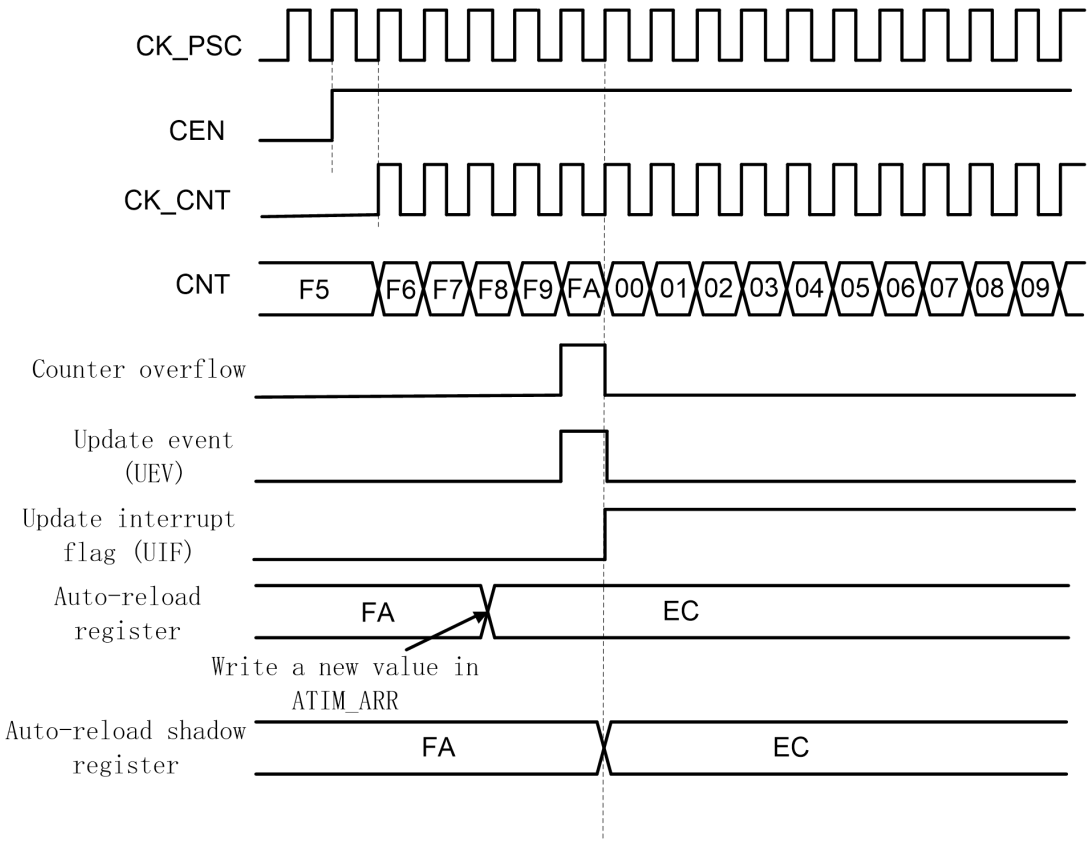


图 25-7ARPE=1（ATIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从ARR值开始递减，到0后产生下溢出事件，并且重新从ARR开始计数。

如果使能了重复计数功能，则计数器按照RCR的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置UG寄存器直接触发update event，此时CNT和预分频计数器自动清零。设置UG寄存器是否触发UIF（Update Interrupt Flag）中断标志置位由URS寄存器的设置决定。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- RCR影子寄存器被更新为ATIM_RCR内容
- ARR影子寄存器被更新为ATIM_ARR内容
- PSC影子寄存器被更新为ATIM_PSC内容

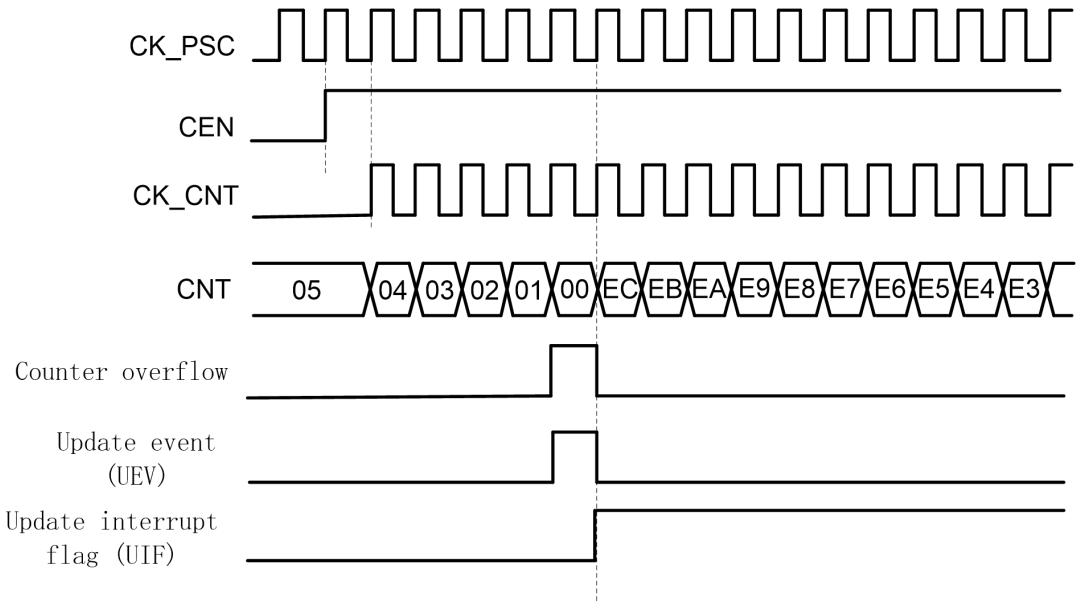


图 25-8 向下计数，内部时钟不分频

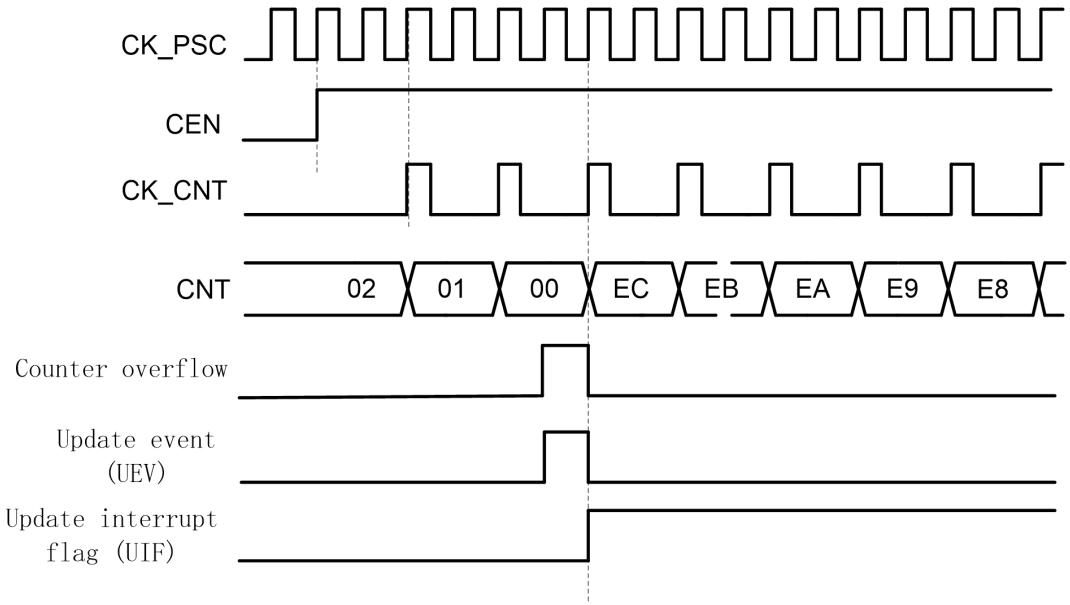


图 25-9 向下计数，内部时钟 2 分频

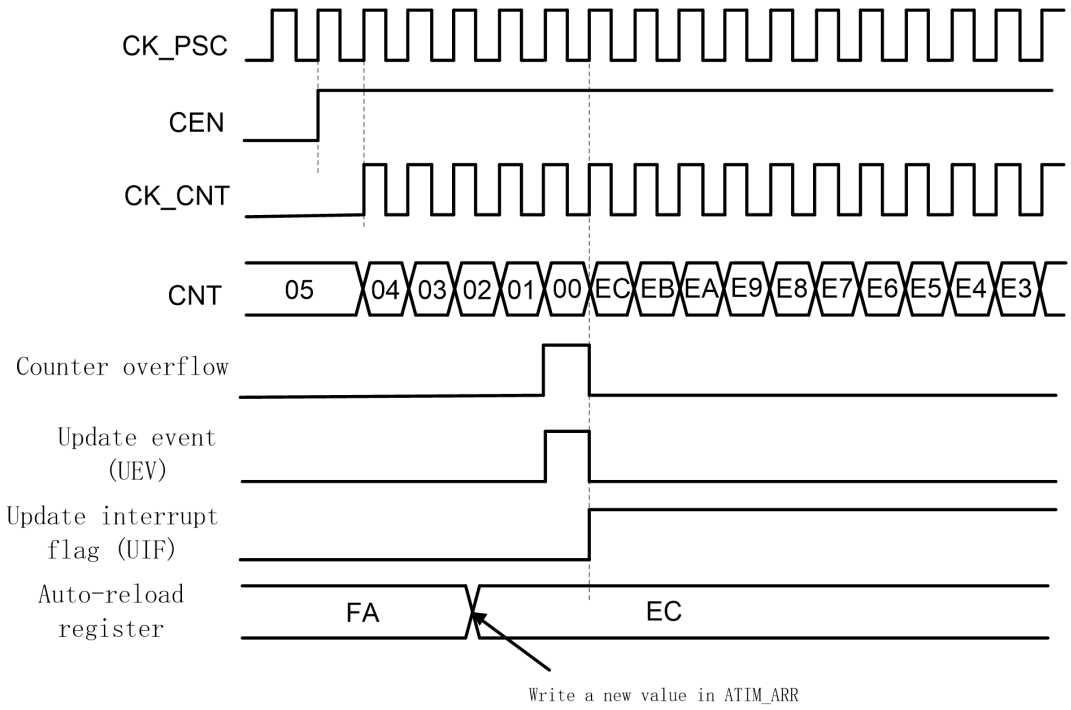


图 25-10 向下计数，内部时钟 2 分频

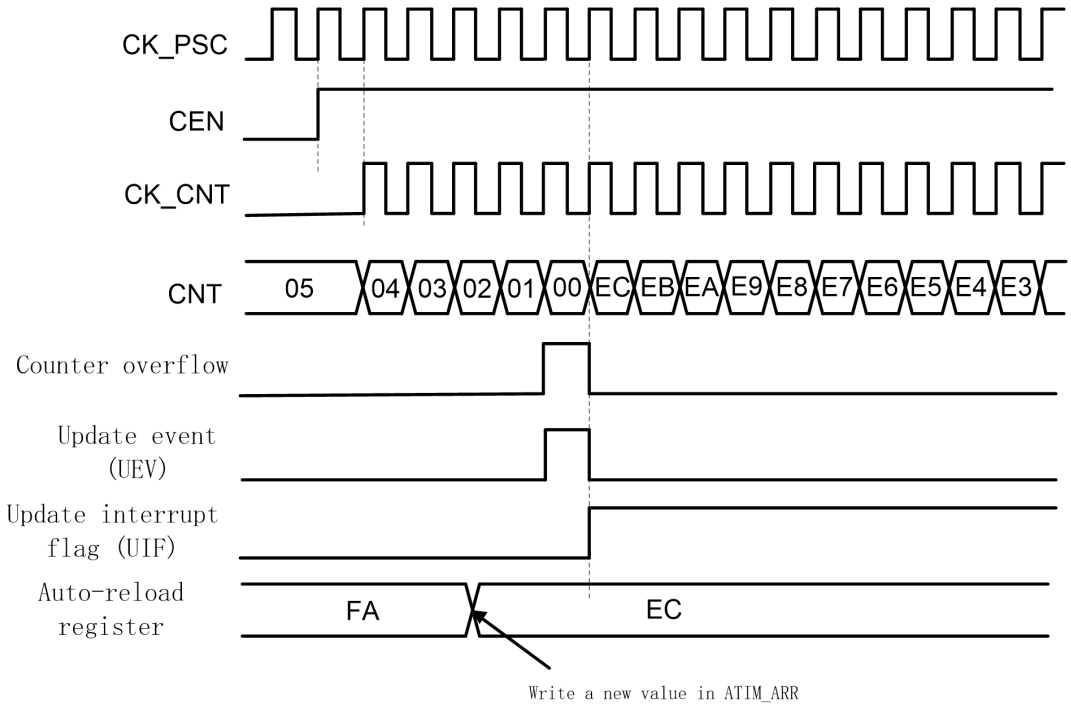


图 25-11 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从0开始向上计数，到ARR-1产生上溢出事件，然后从ARR开始向下计数到1，产生下溢出事件，再从0重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当CMS!=00时为中心对齐计数，当CMS=01时，输出比较功能仅在向下计数时有效，当CMS=10时，输出比较功能仅在向上计数时有效，当CMS=11时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在overflow和underflow的事件上都会更新 ARR、PSC和RCR的影子寄存器。

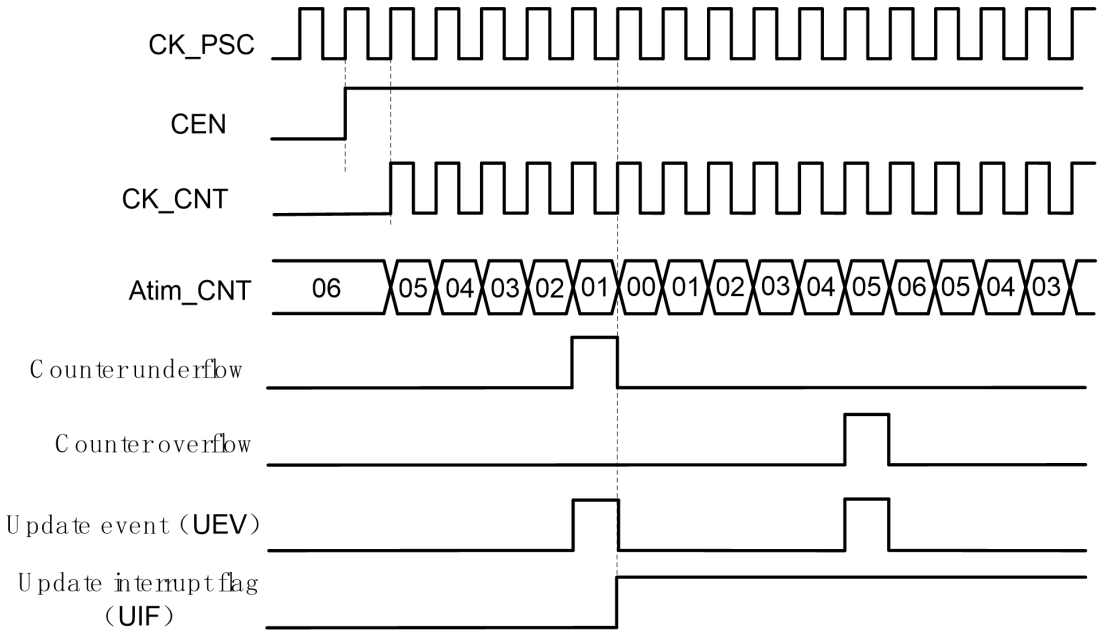


图 25-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

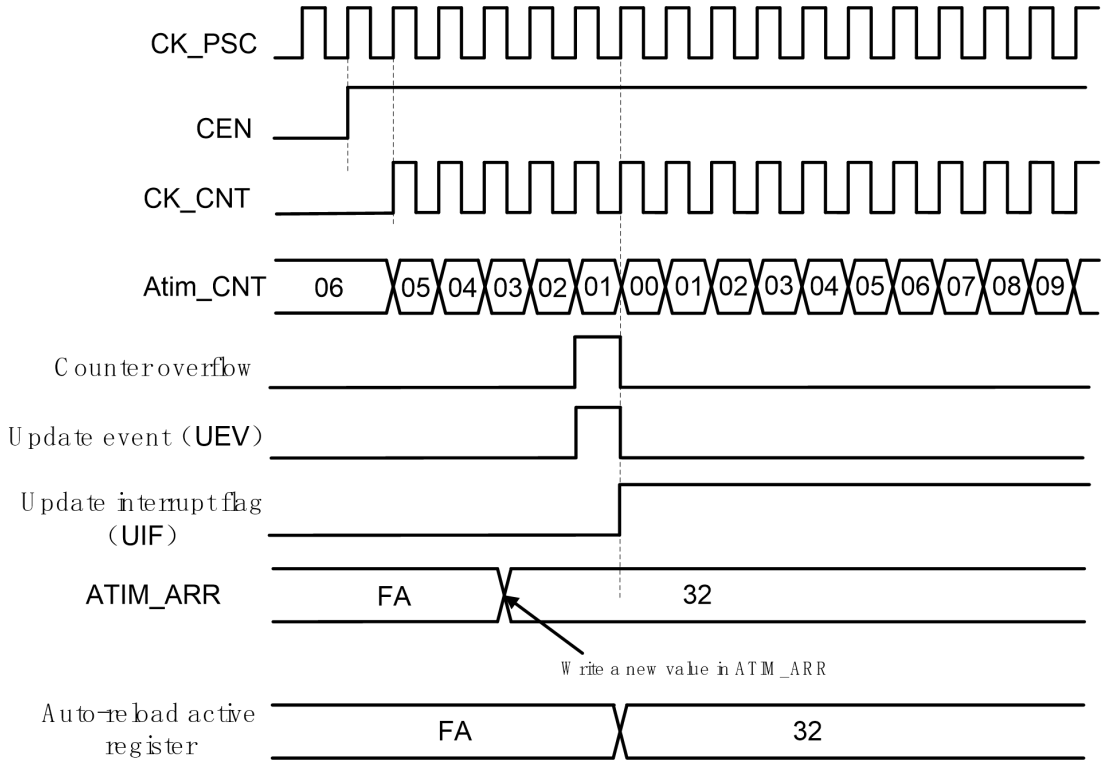


图 25-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

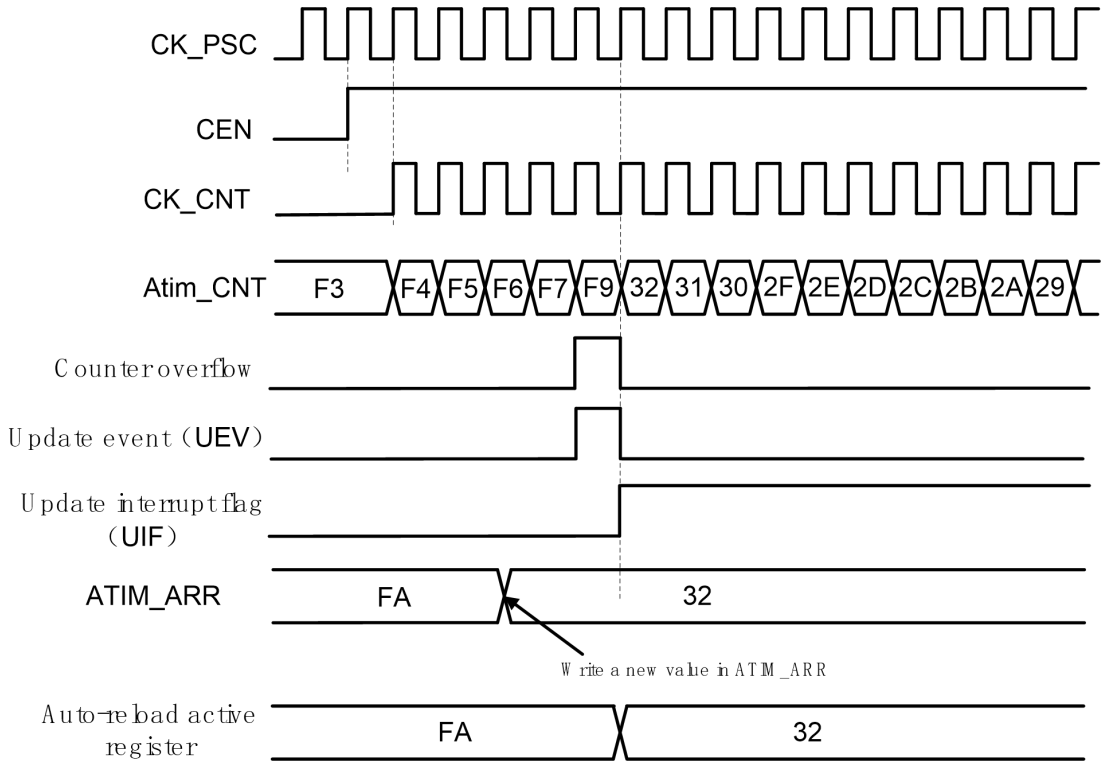


图 25-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

25.4.3 重复计数器

Update event在计数器overflow或underflow，并且重复计数器为0 的情况下产生。这意味着ARR、PSC、CCR（比较/捕捉寄存器，输出比较模式下）的preload寄存器会在N+1次overflow或underflow之后，才将数据传输给影子寄存器，其中N是RCR寄存器值。

重复计数器在以下情况下递减：

- 向上计数模式下发生上溢出
- 向下计数模式下发生下溢出
- 中心计数模式下每次上溢出或者下溢出

注意，当update event由软件或slave mode controller触发时，更新事件会立即发生，而不管当前RCR是什么值，同时重复计数器也会被立即更新为RCR的值。

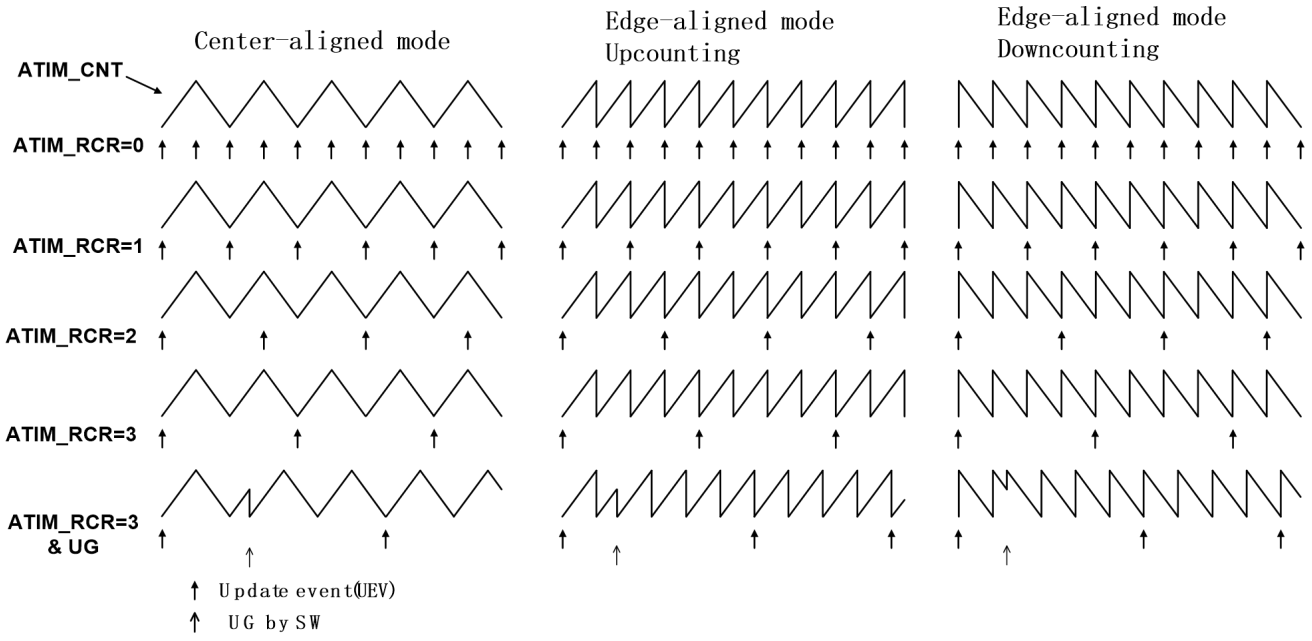


图 25-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

25.4.4 Preload 寄存器

以下功能寄存器支持preload功能：

- 自动重载寄存器ARR
- 重复计数寄存器RCR
- 预分频寄存器PSC（不可关闭preload功能）
- 通道控制寄存器CCR
- CcxE和CcxNE控制寄存器
- OcxM控制寄存器

以上寄存器，除了PSC之外，都可以由软件选择使能或者禁止preload功能。

具备preload功能的寄存器，包含两组物理实体：

- Shadow register（影子寄存器）：实际定时器正在使用的寄存器
- Preload register（预装载寄存器）：软件可以访问的寄存器

当禁止preload时，具备preload功能的寄存器特性如下：

- Preload寄存器可以实时由软件访问、改写
- Shadow寄存器与Preload寄存器同步更新

如果使能了preload，则：

- 所有软件操作访问的是preload寄存器
- 当update event发生时，所有preload寄存器内容将同步被转移到对应的shadow寄存器

25.4.5 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（Tix）——外部时钟模式1
- 外部引脚触发输入（ETR）——外部时钟模式2
- 内部触发（ITRx）——使用一个timer的触发输出（TGO）作为计数时钟

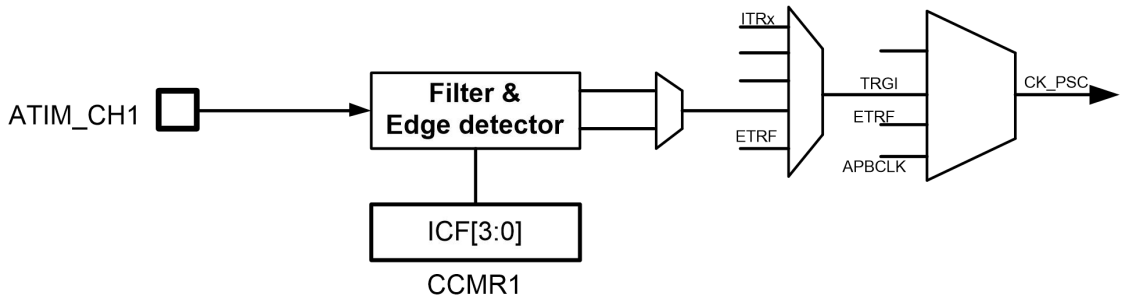


图 25-16 ATIM 时钟源框图

25.4.5.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

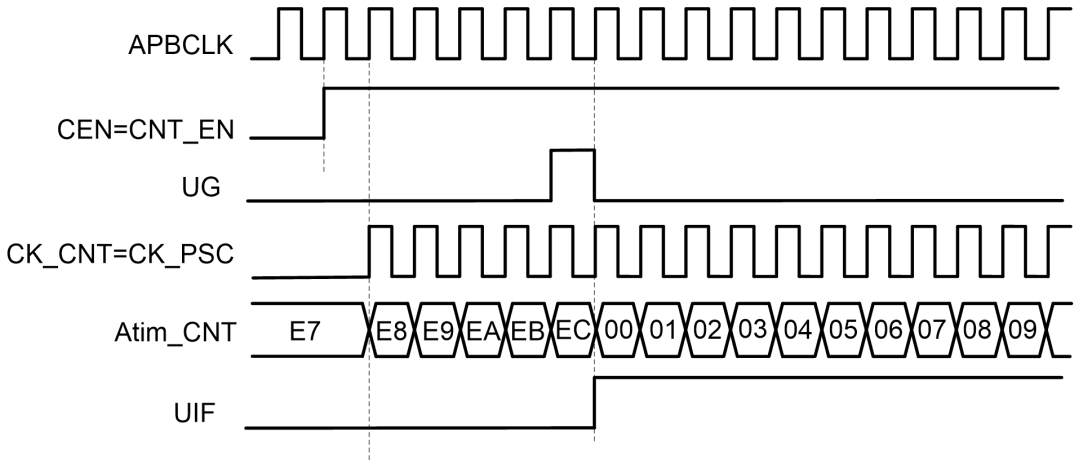


图 25-17 内部时钟源模式，时钟分频因子为 1

25.4.5.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置SMS=111，计数边沿可以配置为上升或下降沿。

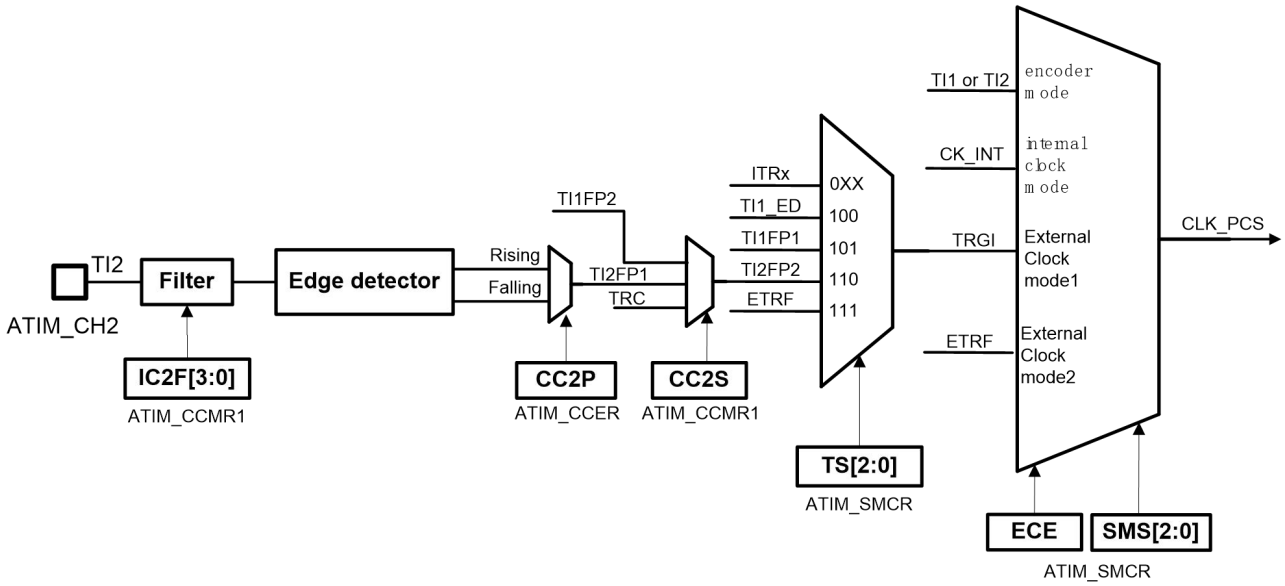


图 25-18TI2 外部时钟连接例子

外部输入信号在触发计数器计数前，会先经过内部时钟的同步过程，同时输入信号的有效沿会触发TIF标志

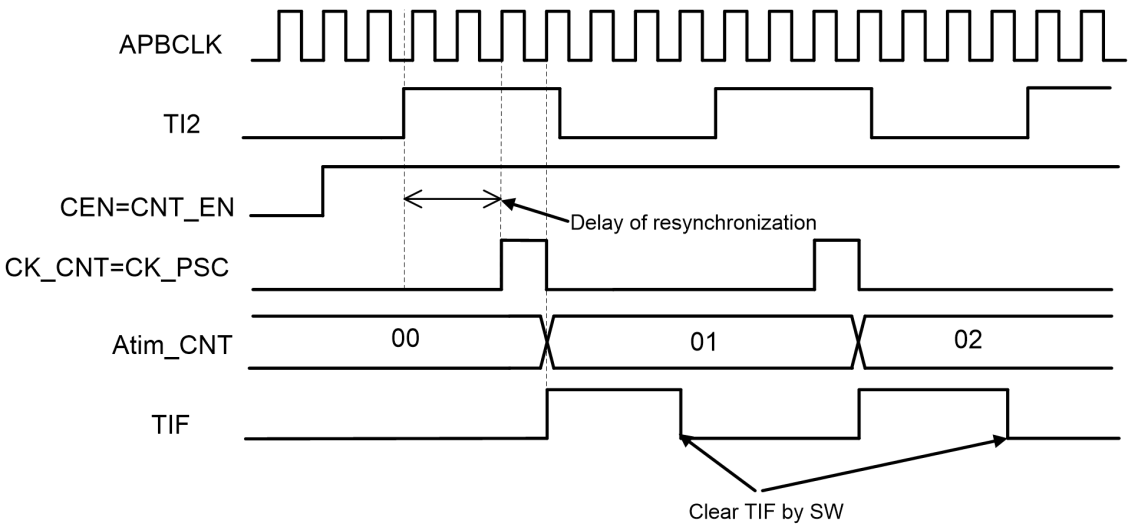


图 25-19 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能ATIM的内部时钟（APBCLK），因为ATIM要使用APB_CLK来对外部输入时钟进行同步和滤波。在外部时钟模式1下，外部输入时钟首先经过滤波和边沿选择，得到

有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的2级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于2个APB_CLK周期。

此模式下只有通道1和2的输入可以用做时钟输入，所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置ATIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置ATIM_SMCR.SMCR=111
- 选择触发输入源，配置ATIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置ATIM_CCER.CC2E=1
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

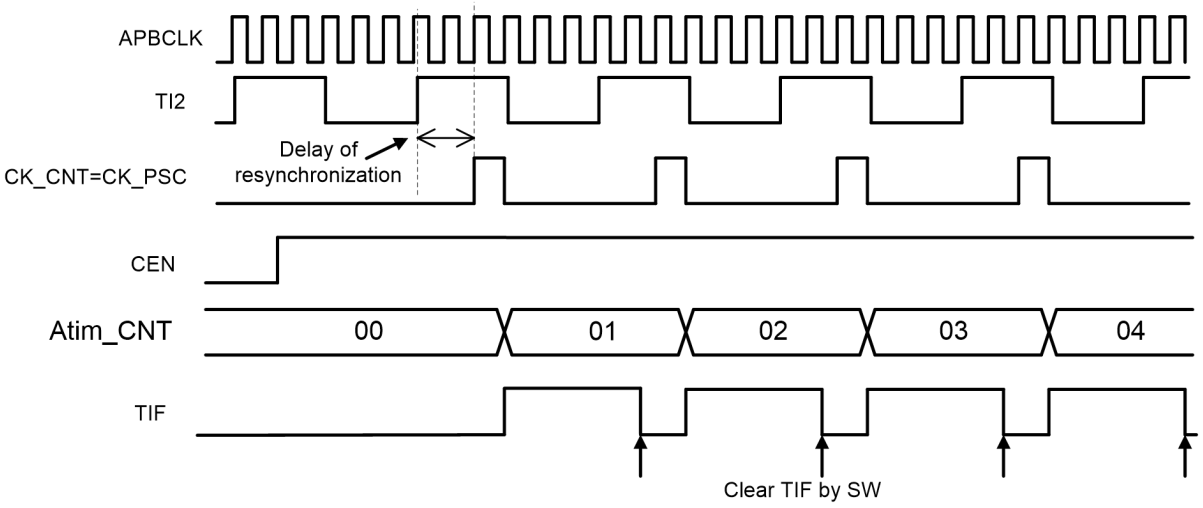


图 25- 20 外部时钟模式 1 下的时序

25.4.5.3 外部时钟模式 2

此模式下使用ATIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

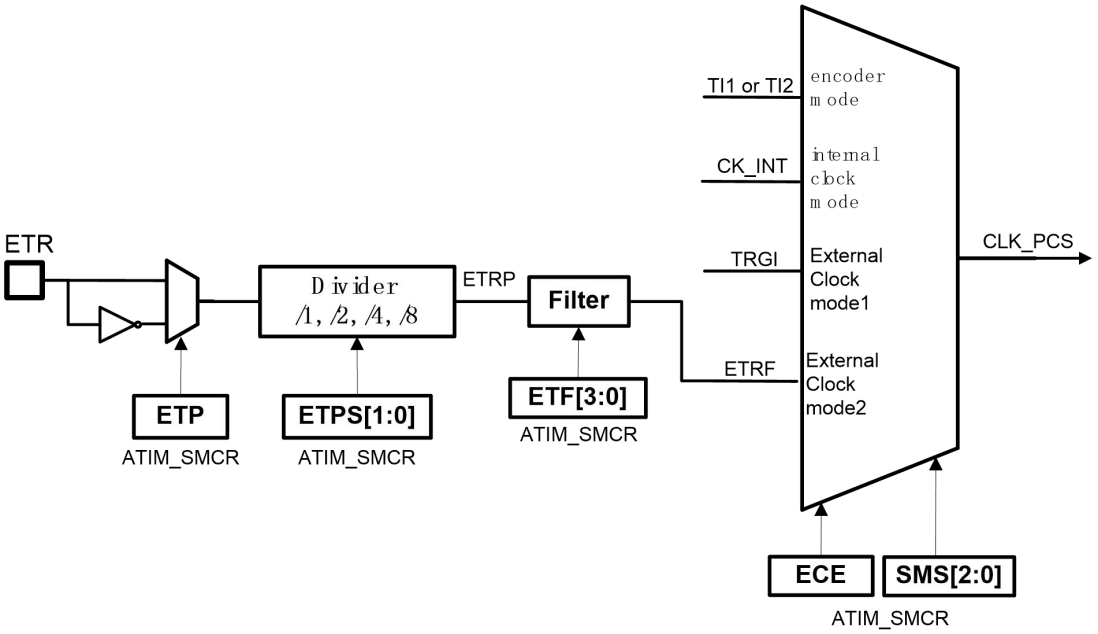


图 25-21 外部触发输入框图

下图是使用ETR二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于ETR输入上升沿。

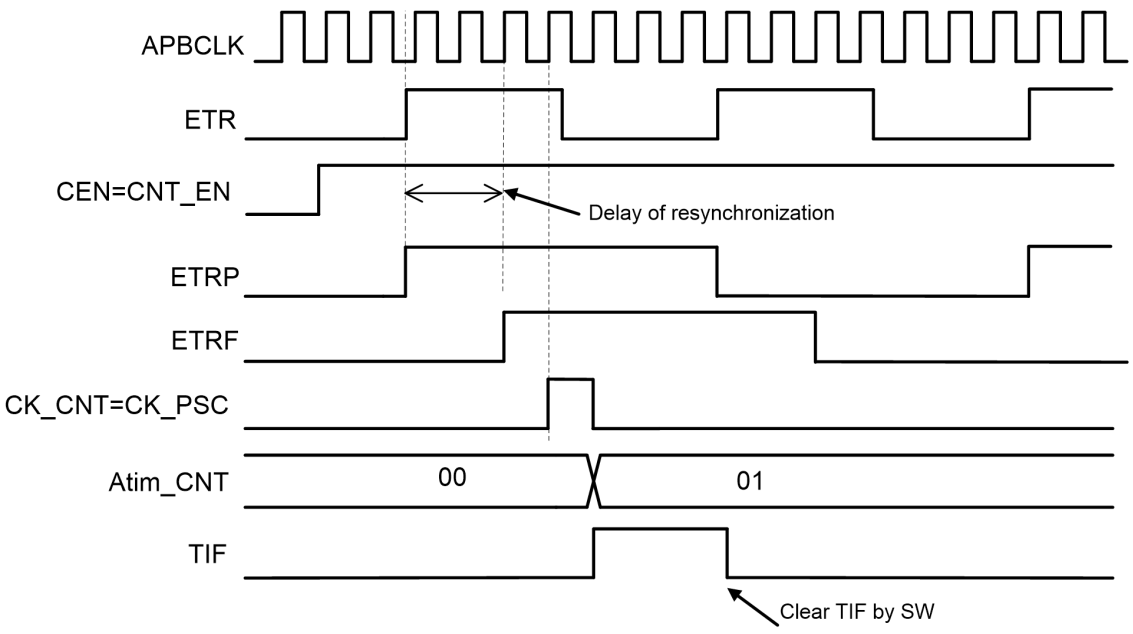


图 25-22 外部时钟模式 2 下的时序 1

与外部时钟模式1的主要差别是，ETR输入直接被分频后再进行滤波，产生CK_PSC时钟，这意味着可以支持ETR输入频率高于APB_CLK的应用场景，这种情况下，需要首先对ETR输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1，ATIM_SMCR.SMS=000
- 使能计数器，配置ATIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

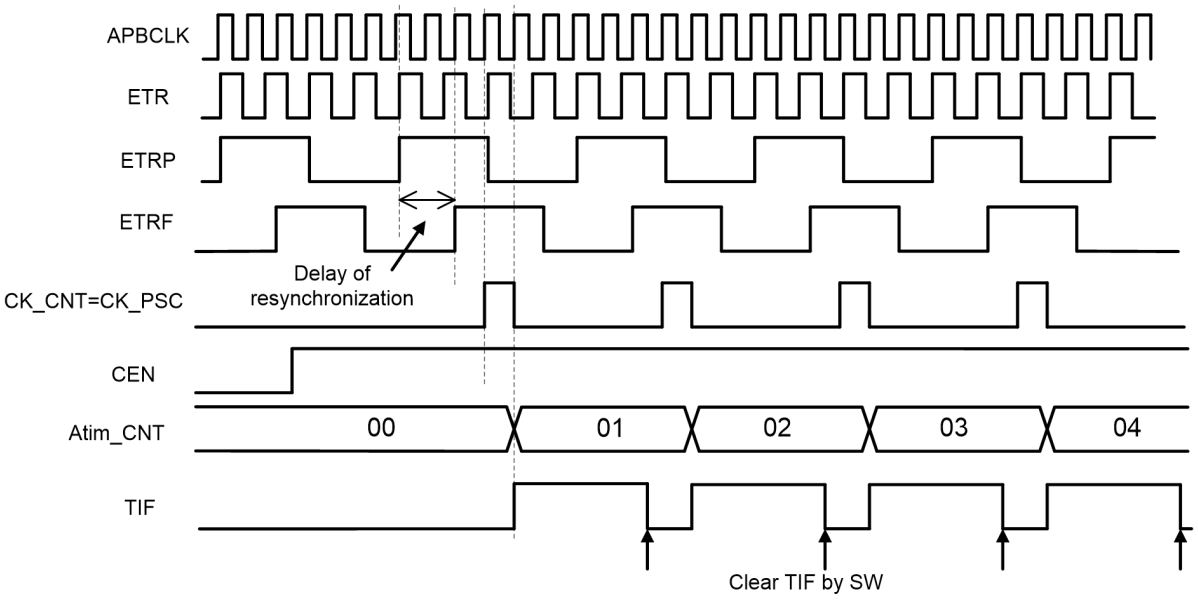


图 25-23 外部时钟模式 2 下的时序 2

在使用外部时钟模式2时，仍可以将ATIM配置为slave模式：比如使用ETR输入计数，同时使用另一个Timer的TRGO作为触发信号，当触发事件到来时，复位计数器重新开始计数。

25.4.6 内部触发信号（ITRx）

ATIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIM	TAU10_TRGO	TAU11_TRGO	COMP1	COMP2

25.4.7 捕捉/比较通道

ATIM包含4个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样Tix输入并产生滤波后的信号TixF，然后边沿检测和极性选择产生对应的TixFPx信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

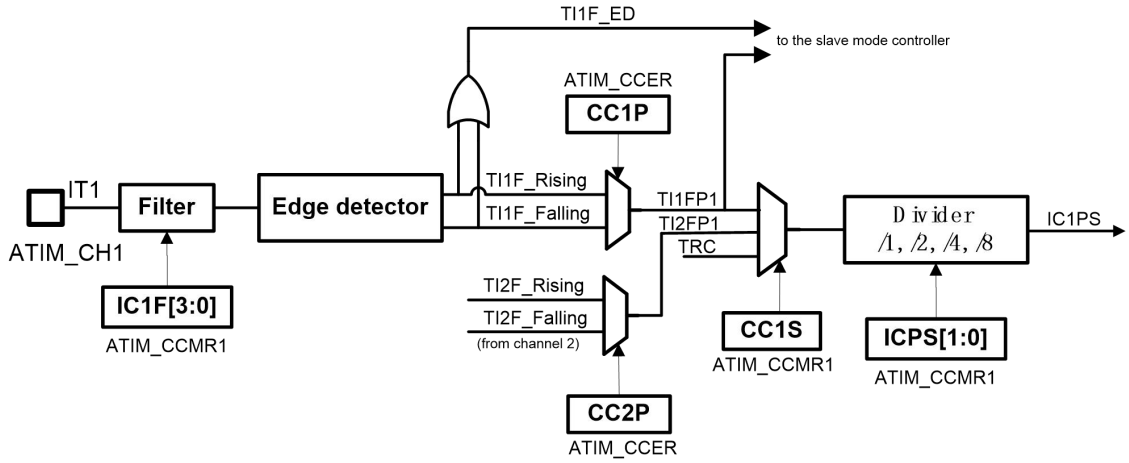


图 25-24 捕获/比较通道(通道 1 输入部分)

输出级电路会产生一个输出基准信号OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道1~3支持互补输出和死区插入，通道4则比较简单，不支持互补输出。

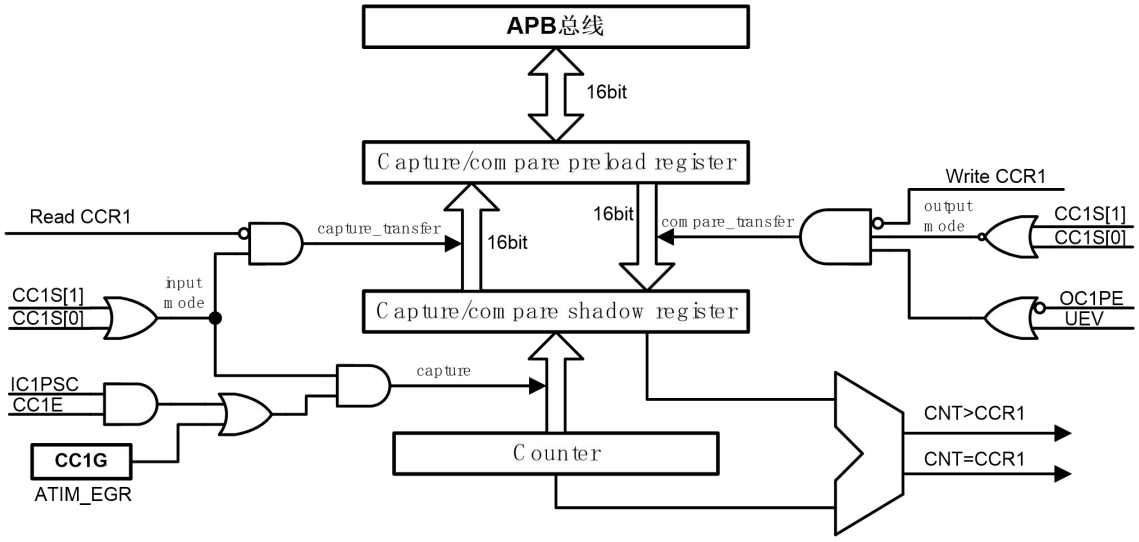


图 25-25 捕获/比较通道 1 的主电路

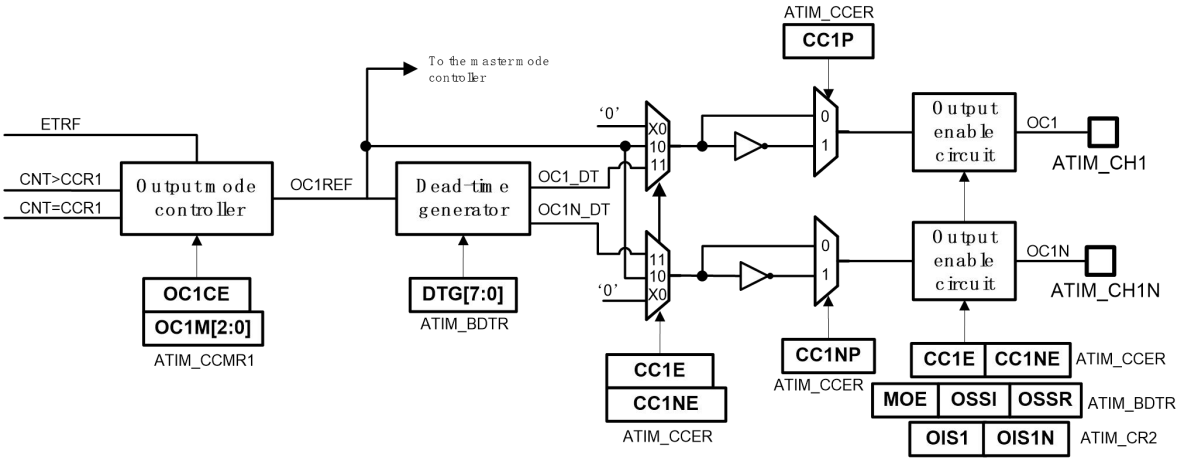


图 25-26 捕获/比较通道的输出部分(通道 1 至 3)

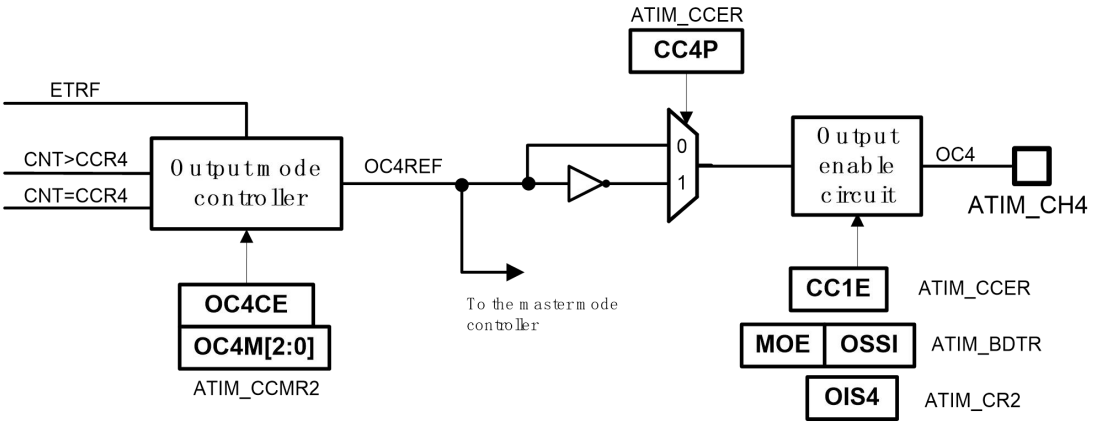


图 25-27 捕获/比较通道的输出部分(通道 4)

捕捉/比较寄存器（CCR）包含preload寄存器和shadow寄存器，软件读写总是访问preload寄存器。在捕捉模式下，捕捉值保存在shadow寄存器中并复制到preload寄存器。在比较模式下，preload寄存器的值被拷贝到shadow寄存器用来与计数器比较。

25.4.8 输入捕捉模式

当Icx信号上出现预期的电平变换，将触发一次capture，当前计数器值被锁存进CCR，与此同时，CcxIF中断标志置位，并且可以触发对应的中断或者DMA请求。如果一个捕捉事件在CcxIF为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR中上次捕捉值被覆盖）。CcxIF可以由软件清零，或者通过读取CCR寄存器自动清零。CcxOF标志通过软件写1清零。

通过两个或更多通道配合，可以实现PWM信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从TI1引脚输入，芯片内部将滤波后的信号取上升沿得到TI1FP1，将滤波后的信号取下降沿得到TI1FP2，将TI1FP1输入给捕捉通道1，将TI1FP2输入给捕捉通道2，即可实现通道1对输入信号上升沿捕捉，同时通道2对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过CCR1和CCR2寄存器的值，即可计算输入信号的周期和占空比。

实现在TI1输入的上升沿捕获计数器的值到ATIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置ATIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置ATIM_CCER.CC1E=1

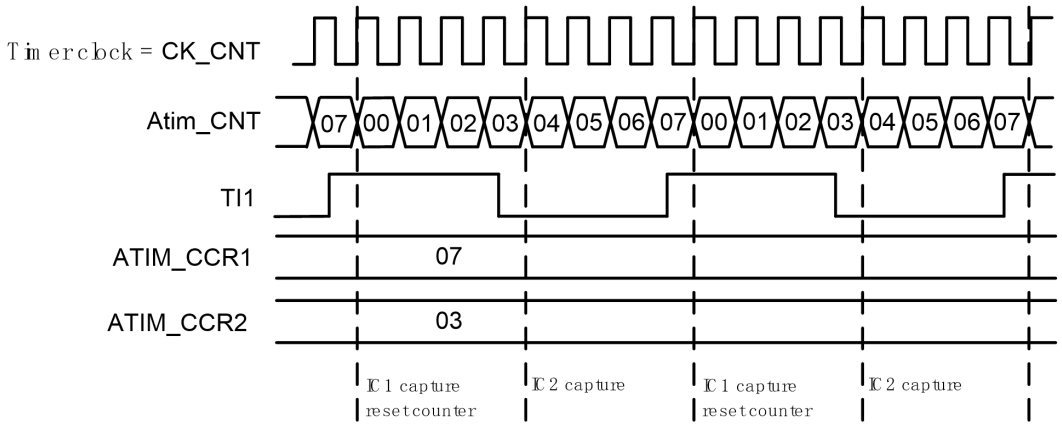


图 25-28 PWM 输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0确保之后通道配置成功
- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置ATIM_CCMR1.CC1S=01,

ATIM_CCMR1.CC2S=10

- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=1
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]，ATIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0] ，ATIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

25.4.9 软件 Force 输出

在比较输出模式下，软件可以直接将OCxREF force成特定电平，而独立于CCR和计数器的比较结果。

软件通过写OcxM=101寄存器，可以直接将OCxREF强制为有效（OCxREF固定为高有效），通过写OcxM=100可以直接将OCxREF强制为无效（低电平）。但是软件force操作不会取消比较过程，CCR和计数器的比较还会一直进行。

25.4.10 输出比较模式

输出比较模式下，当CCR与计数器值相等，OCxREF可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA请求可以发送。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

- 选择计数时钟（内部、外部、预分频等）
- 向ARR和CCR寄存器写入期望数据
- 根据需要设置中断使能和DMA使能
- 选择输出模式
- 使能计数器

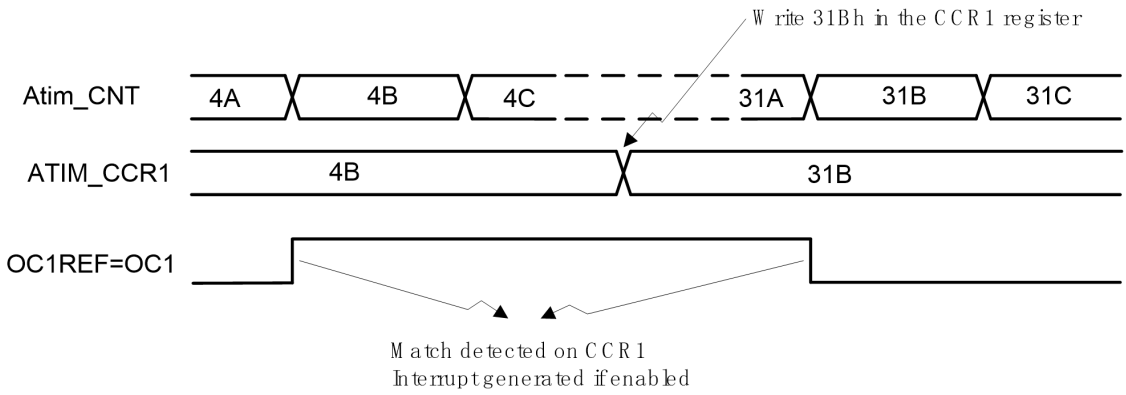


图 25-29 输出比较模式，翻转 OC1

在不使能preload的情况下，软件可以随时改写CCR寄存器实现对输出波形的实时控制。如果使能了preload，则CCR shadow寄存器仅在下一次update event发生时更新为preload寄存器的内容。

25.4.11 PWM 输出

PWM模式可以输出脉宽调制信号，其周期由ARR寄存器决定，占空比由CCR寄存器决定。

输出信号的极性可以由CCxP寄存器配置。PWM模式工作中，CNT和CCR实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM输出也支持边缘对齐和中央对齐模式。

PWM边缘对齐模式

在向上计数的情况下，配置为PWM模式1时，OCxREF信号在CNT<CCR时为高电平，否则为低电平。如果CCR值大于ARR值，则OCxREF被固定为1；如果CCR为0则OCxREF被固定为0.

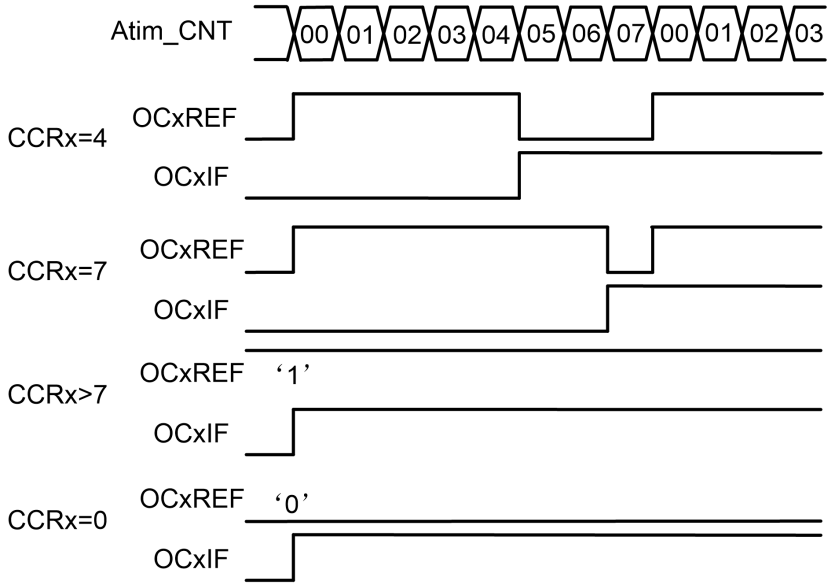


图 25-30 边沿对齐的 PWM 波形(ARR=7)

在向下计数时，OCxREF电平高低定义与向上计数时相同。

PWM中央对齐模式

OCxREF电平定义与边缘对齐模式相同。下图是一个示例：

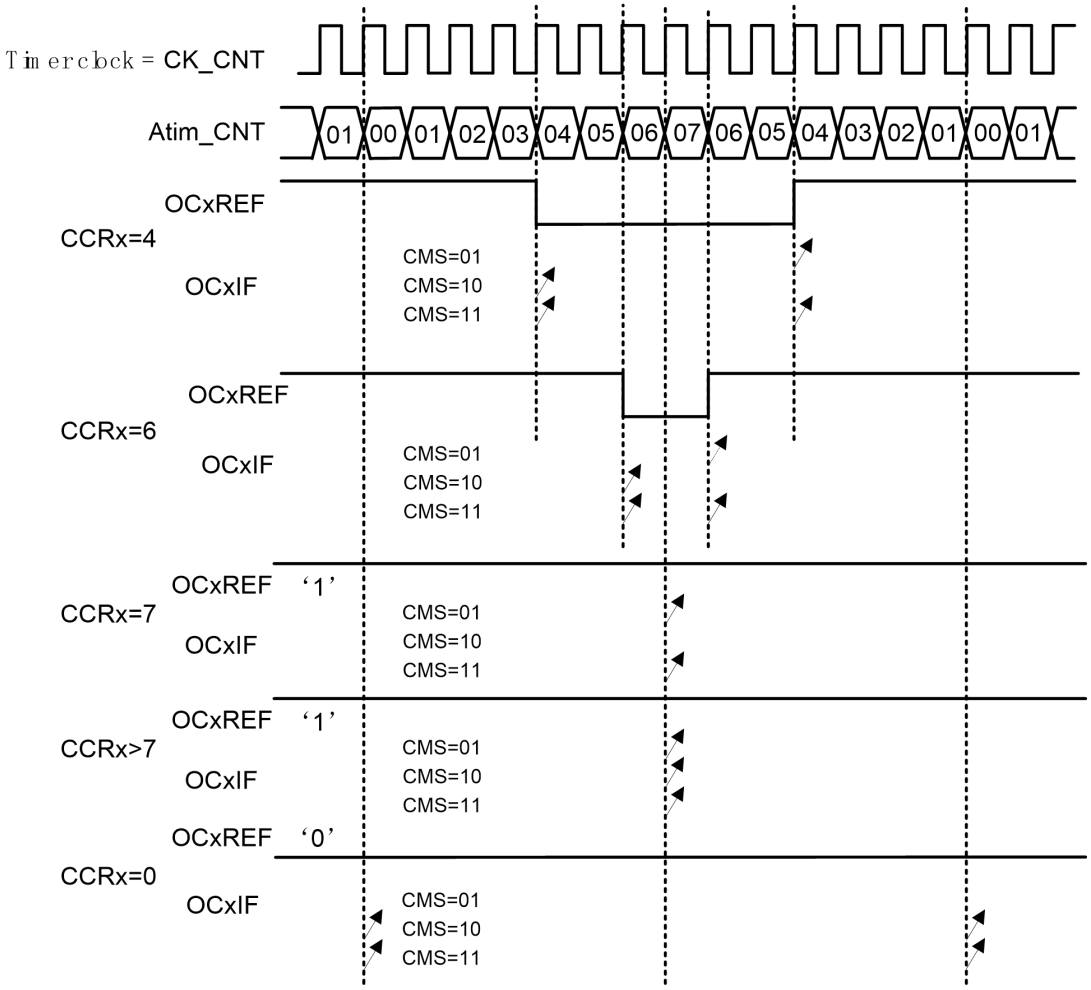


图 25-31 中央对齐的 PWM 波形(APR=7)

当启动中央对齐计数时，一开始的计数方向是由DIR寄存器决定的；随后在计数过程中，DIR寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过UG寄存器做一次update，并且在计数过程中不要改写计数器。

25.4.12 互补输出和死区插入

ATIM的通道1~3支持互补输出和死区插入。DTG[7:0]寄存器用于设置死区时间（对所有通道同时有效）。输出信号Ocx与参考信号OCxREF同相，OcxN与参考信号反相；Ocx的上升沿是OCxREF上升沿的delay，OcxN的上升沿是OCxREF下降沿的delay。

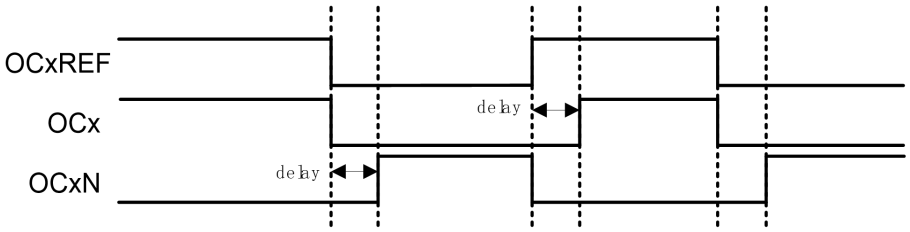


图 25-32 带死区插入的互补输出

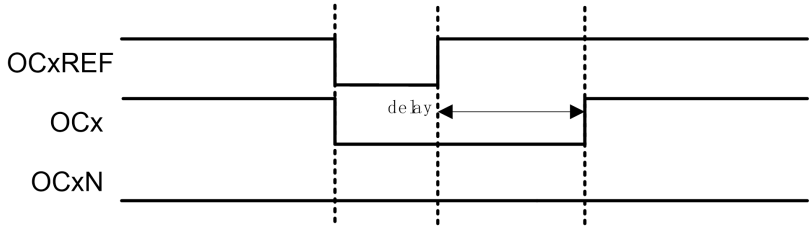


图 25-33 死区波形延迟大于负脉冲

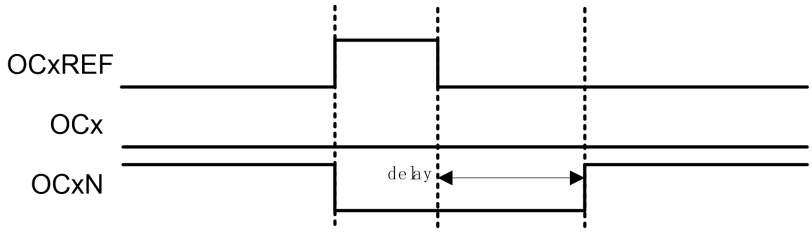
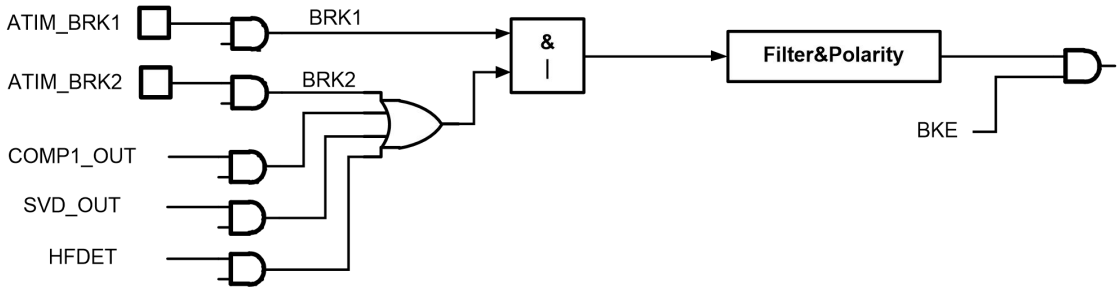


图 25-34 死区波形延迟大于正脉冲

25.4.13 刹车功能

刹车功能可以使用外部BRK引脚输入的2路刹车信号，或者比较器、SVD、XTHF停振检测产生的有效输出；上电复位后刹车电路被禁止，用户通过置位BKE寄存器使能刹车功能；2路刹车输入可以配置为相与或者相或操作。组合后的刹车信号可以配置有效极性，以及数字滤波。

刹车输入控制逻辑如下图所示：



ATIM_BRKx复用GPIO功能，当GPIO设置为数字外设功能时，其输入信号直接连接到ATIM的刹车输入上；当GPIO设置为其他功能时，ATIM的刹车输入端口被固定成1。通过BRKxGATE寄存器，可以控制门控后的BRKx信号的实际电平，软件能够灵活的将不使用的BRKx设置为0或者1电平，以适应后级逻辑电路的需要。

当一个刹车事件发生时：

- 输出使能寄存器被异步清零，可以通过OSSI寄存器选择输出被强制为inactive/idle/reset状态
- 每个输出通道被驱动为OISx寄存器定义的电平
- 当互补输出使能时，输出被异步置位成inactive和reset状态，死区插入电路开始工作，在死区时间后驱动输出为OISx和OISxN定义的电平
- 刹车标志寄存器置位，根据配置可以触发中断或DMA
- 如果使能了自动输出（AOE=1），输出使能位（MOE）将在下一个update event发生时被自动置位；否则MOE将保持为0直到被软件重新置位。

注意BRK信号是电平有效的，因此在BRK保持有效的情况下，无法使能MOE，同时刹车标志BIF也无法清除。

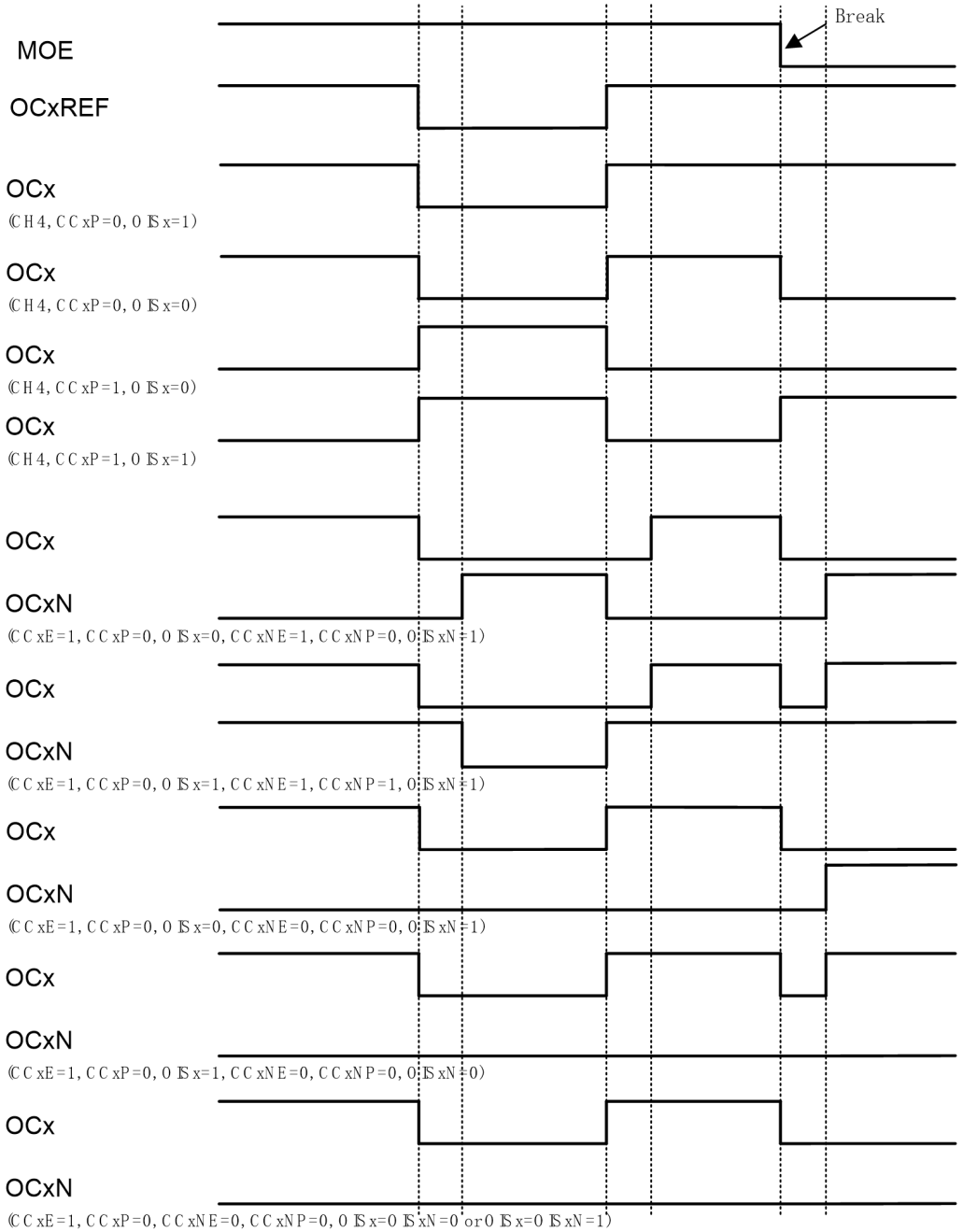


图 25-35 响应刹车的输出

25.4.14 互补输出通道信号状态逻辑表

以下是控制寄存器和互补输出通道的状态对应表，其中 MOE 为定时器总输出使能位，OSSI 定义 IDLE 状态（MOE=0）下是关闭 IO 输出还是进入 off state，OSSR 定义 RUN 状态（MOE=1）下的是关闭 IO 输出还是进入 off state。

控制寄存器					输出状态	
MOE	**OSSI**	**OSSR**	**CcxE**	**CcxNE**	**Ocx 输出状态**	**OcxN 输出状态**
1	X	0	0	0	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	输出关闭（不由ATIM驱动）,OcxN=0,OcxN_EN=0
		0	0	1	输出关闭（不由ATIM驱动）,Ocx=0,Ocx_EN=0	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		0	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Output Disabled (not driven by the timer) OcxN=0, OcxN_EN=0
		0	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) Ocx=CCxP, Ocx_EN=0	Output Disabled (not driven by the timer) OcxN=CCxNP, OcxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) Ocx=CCxP, Ocx_EN=1	OCxREF + Polarity OcxN=OCxREF xor CCxNP, OcxN_EN=1
		1	1	0	OCxREF + Polarity Ocx=OCxREF xor CCxP, Ocx_EN=1	Off-State (output enabled with inactive state) OcxN=CCxNP, OcxN_EN=1
		1	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
0	0	X	0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	0		0	1	输出关闭（不由ATIM驱动） 如果无时钟：Ocx=CCxP, Ocx_EN=0, OcxN=CCxNP, OcxN_EN=0 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	0		1	0		
	0		1	1		
	1		0	0	输出关闭（不由ATIM驱动） Ocx=CCxP, Ocx_EN=0	输出关闭（不由ATIM驱动） OcxN=CCxNP, OcxN_EN=0
	1		0	1	Off-state（输出使能，inactive输出） 如果无时钟：Ocx=CCxP, Ocx_EN=1, OcxN=CCxNP, OcxN_EN=1 如果有时钟：经过死区时间后Ocx=OISx, OcxN=OISxN	
	1		1	0		
	1		1	1		

25.4.15 6-step PWM 输出

当某个通道使用互补输出时，OcxM, CcxE, CcxNE寄存器支持preload功能，preload寄存器的值在换相（COM）事件发生时被装载到shadow寄存器中。用户因此可以预先设置下一步配置，并在COM事件发生时同步更新所有通道。COM事件可以由软件写ATIM_EGR中的COM位触发，或者由TRGI上升沿硬件触发。

当COM事件发生时，换相标志寄存器置位，并且可以产生中断或DMA请求。

下图是一个6步换相控制的例子，当COM事件发生时，三个例子显示不同配置下的输出变化。

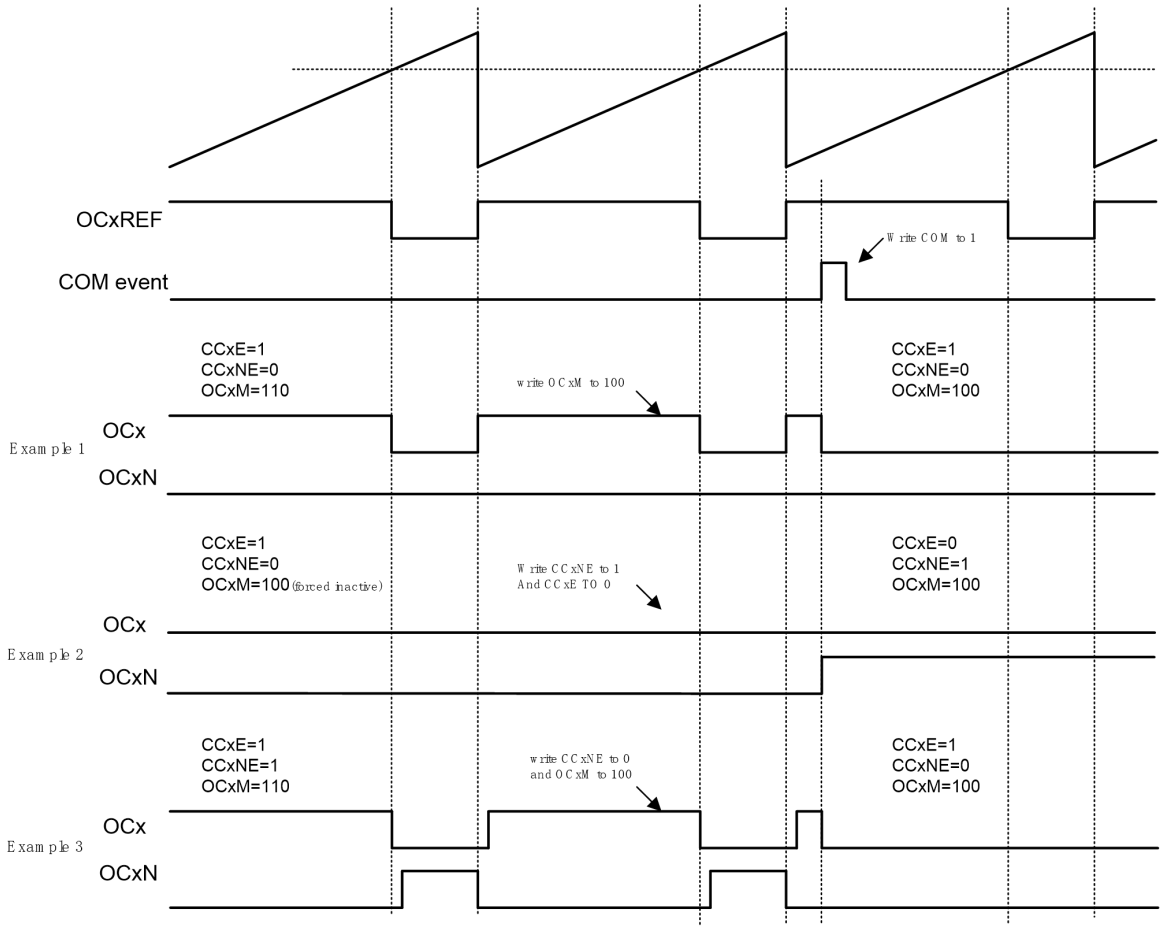


图 25-36 产生六步 PWM，使用 COM 的例子(OSSR=1)

25.4.16 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次update event到来时，计数器会自动停止。只有当CCR和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求CNT<CCR<=ARR，在向下计数时，要求CNT>CCR

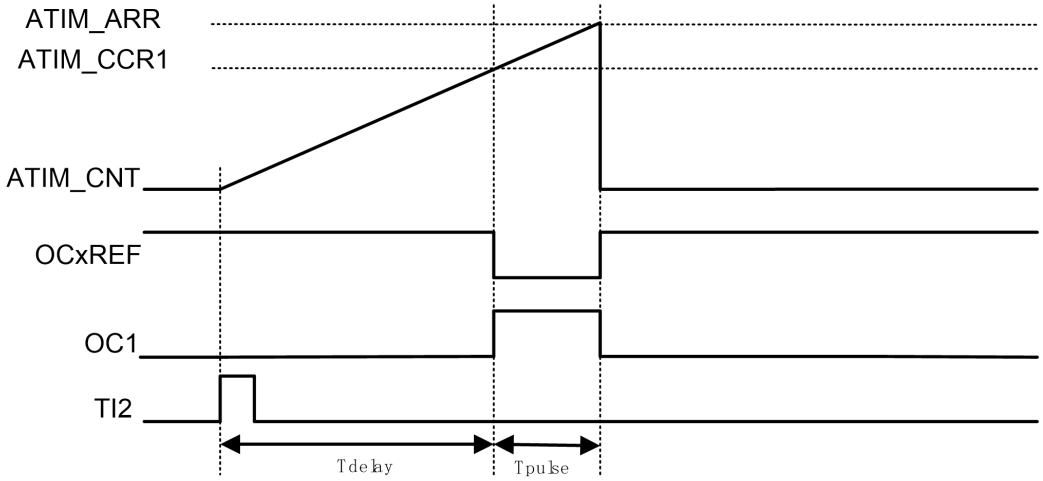


图 25-37 单脉冲模式的例子

上图是以TI2输入为计数器触发信号，计数值等于CCR后OCxREF输出低电平，计数到ARR后OCxREF回到高电平，并且计数器回滚到0，停止计数。

实现上述功能TI2作为输入触发的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH2功能
- 关闭通道使能，配置ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC2P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置ATIM_CCER.CC2E=1

实现上述功能OC1作为输出的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置ATIM_CCMR1.CC1S=00

- 选择计数有效沿，配置ATIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置ATIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- ATIM_CCR1的值决定了Tdelay
- ATIM_ARR和ATIM_CCR1的差值决定了Tpulse（ATIM_ARR-ATIM_CCR1）
- 设置为单脉冲模式，配置ATIM_CR1.OPM=1

25.4.17 外部事件清除 OCxREF

OCxREF的有效状态未高电平，通过对外部ETR引脚施加高电平，可以直接拉低OCxREF，直到下一次update event。此功能仅在输出比较和PWM模式下有效。使能此功能需要将OcxCE置1。

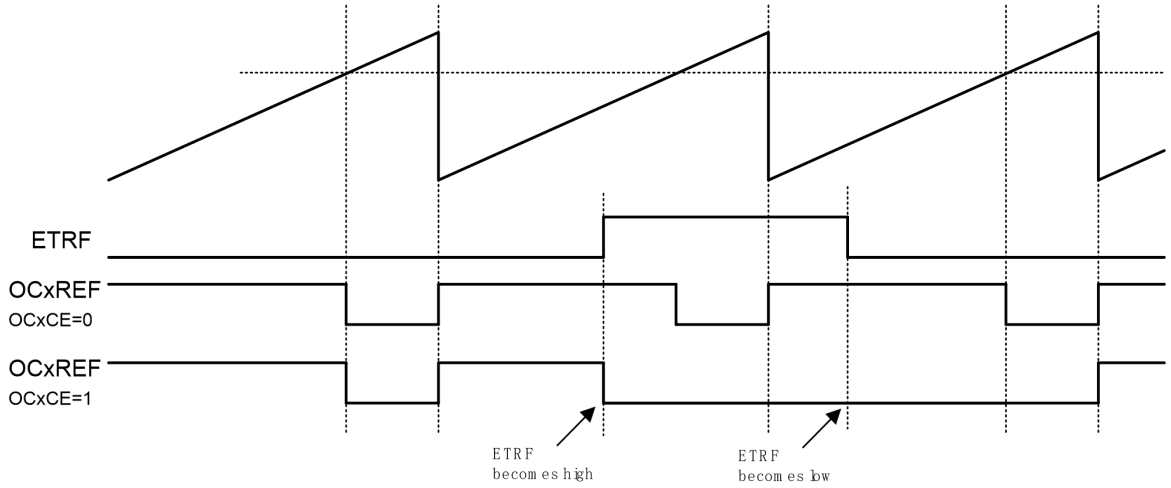


图 25-38 ETR 信号清除 ATIM 的 OCxREF

25.4.18 编码器接口模式（encoder interface）

编码器接口模式涉及到两个外部输入信号，ATIM根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（TI1 对应TI2，TI2对应TI1）	TI1信号		TI2信号	
		上升	下降	上升	下降
仅在TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在TI1 和TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表25-1encoder interface计数方式

比如在计数器以TI1信号为时钟计数时，如果TI1上升沿采样到TI2为高电平，则计数器递减；如果TI1下降沿采样到TI2为高电平，则计数器递增。

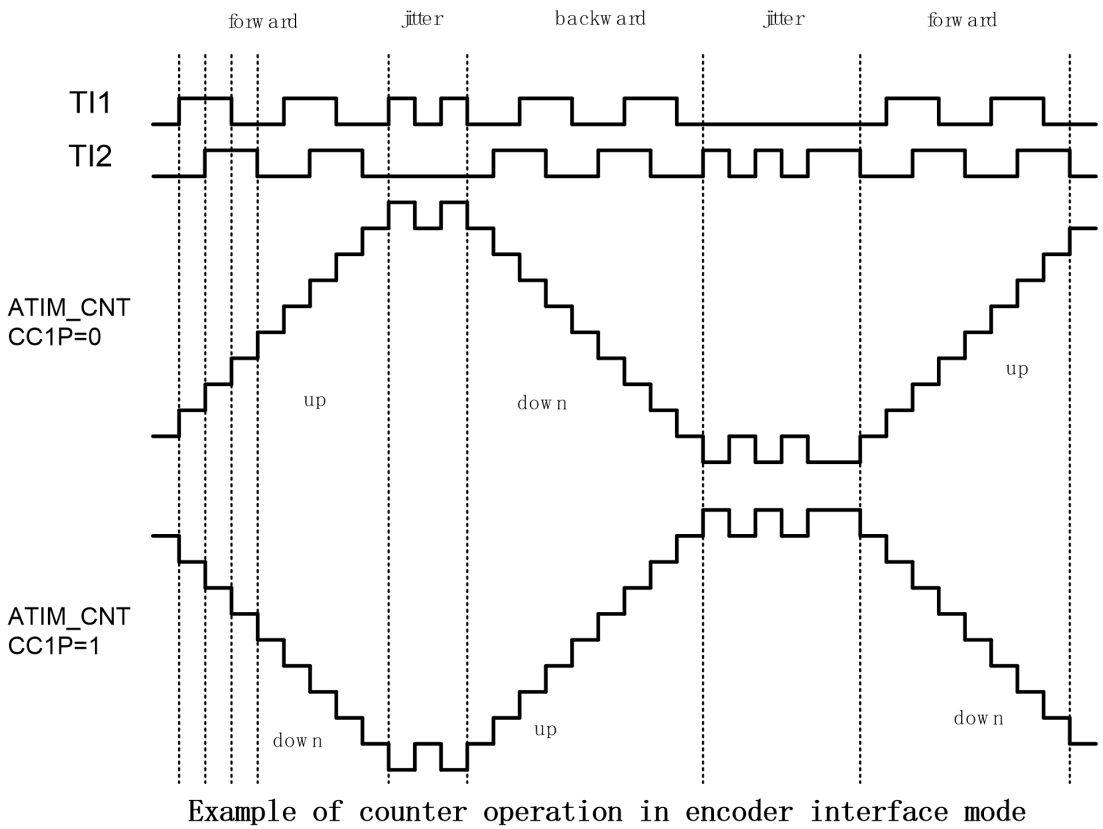


图 25-39 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_CH2功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01，ATIM_CCMR1.CC2S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置ATIM_SMCR.SMS[2:0]=011
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

25.4.19 TIM 从机模式

ATIM作为slave时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致TIM内部所有preload寄存器重新初始化，CNT回到0开始计数。以下图为例，计数器正常计数，外部TI1输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

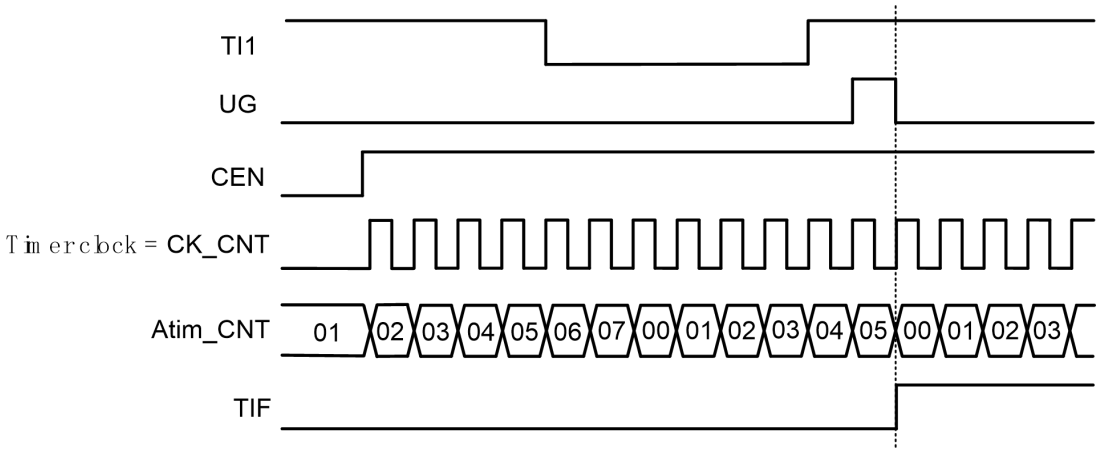


图 25-40 复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能

- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置ATIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置ATIM_CCER.CC1E=1
- 使能计数器，配置ATIM_CR1.CEN=1

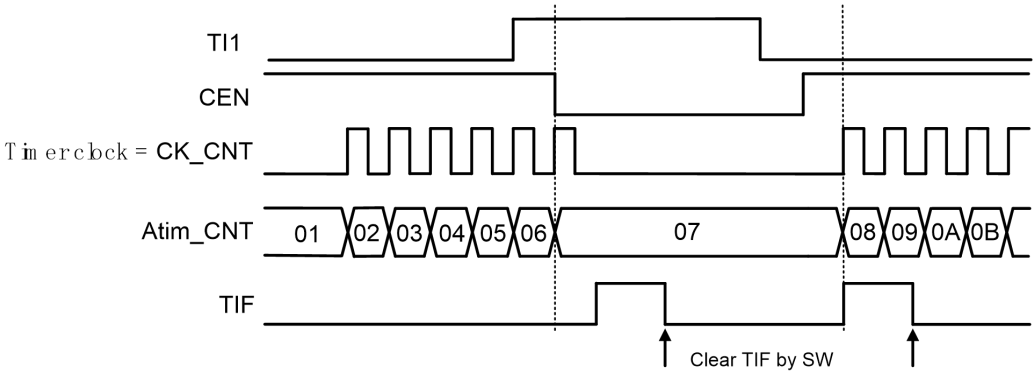


图 25-41 门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

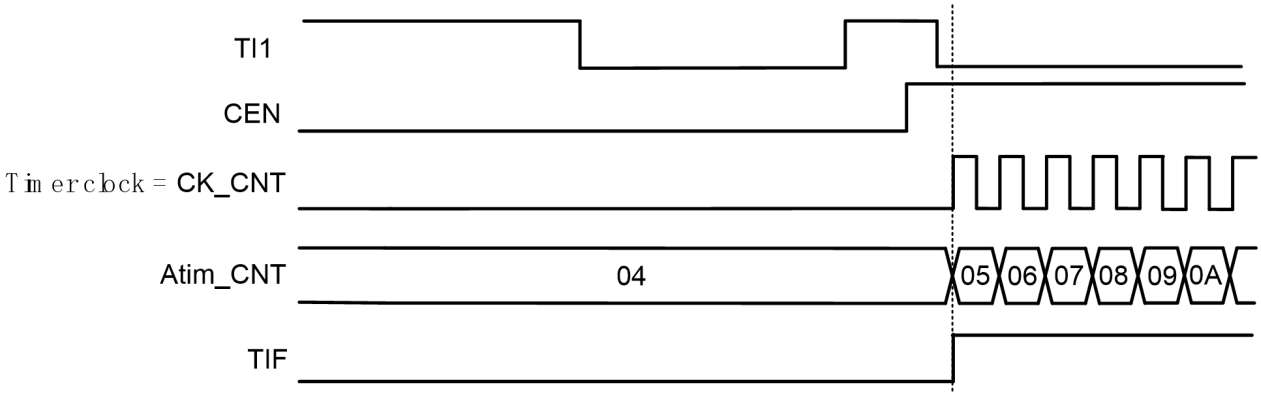


图 25-42 触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将ETR设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到TI1的上升沿之后，计数器开始以ETR输入的上升沿计数。

下图例中的配置如下：

- 在GPIO模块中，配置相应管脚为ATIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，ATIM_SMCR.ETP=0
- 设置ETR分频比，配置ATIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，ATIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,ATIM_SMCR.ECE=1
- 关闭通道使能，配置ATIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置ATIM_CCMR1.CC1S=01
- 选择计数有效沿，配置ATIM_CCER.CC1P=0
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置ATIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置ATIM_CCER.CC1E=1

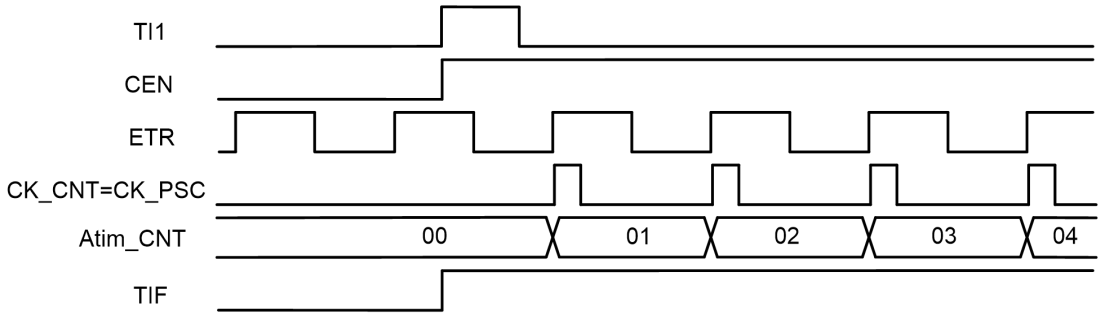


图 25-43 外部时钟模式 2＋触发模式下的时序

25.4.20 DMA 访问

ATIM支持7种DMA请求，分别为4个CC通道请求、外部触发请求、用户软件触发请求和COM触发请求。

其中每个CC通道各自产生一个DMA请求，在捕捉模式下用于将CCRx中的内容传输给RAM，在比较模式下则用于将RAM中的数据写入CCRx；CC通道的DMA请求可以配置为单次传输或Burst传输（CCxBURSTEN），单次传输仅访问CCRx寄存器，Burst传输则根据DCR寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件、软件触发事件和COM事件也可以产生DMA请求，当这些请求发生时，会启动DMA Burst传输，向ATIM内部1个或多个寄存器写入数据，或者从ATIM读取1个或多个寄存器值。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

25.4.21 DMA Burst

ATIM支持DMA和DMA-Burst访问，可以配置ATIM在特定事件发生时触发DMA请求，可以将CCR中的捕捉结果写入RAM，或者从RAM中将一个或多个寄存器内容写入ATIM的preload寄存器中。

DMA-Burst支持一个事件触发连续多次DMA请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA控制器需将外设目标地址指向一个虚拟寄存器ATIM_DMAR。在特定的定时器事件发生时，ATIM会连续发射多个DMA请求。每个DMA对ATIM_DMAR的写操作都会被ATIM重新定向到实际的功能寄存器上。

DBL寄存器用于设置DMA burst长度，DBA寄存器用于设置DMA访问ATIM内部的基地址（相对于ATIM_CR的offset）。

25.4.22 输入异或功能

通道1~3的输入信号可以被异或起来之后，接入到通道1的滤波和边沿电路输入，用于通道1的输入捕捉或者触发。

ATIM_CR2寄存器的TI1S位用于选择通道1的输入是否来自于三个通道输入的异或。

25.4.23 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DBG模块的DBG_TIMx_STOP寄存器定义。

Debug时当定时器被停止后，其输出会被禁止（MOE清零），根据寄存器配置，此时的输出信号可以被force成inactive或由GPIO模块控制。DMA-Burst模式下，DMA所有访问都要指向DMAR虚拟寄存器，由ATIM自动根据访问来累加内部offset地址。DBA寄存器用于指定ATIM内部首次DMA传输的目标地址，而DBL用于指定Burst长度

25.5 寄存器

ATIM模块基地址：0x4001B000

offset 地址	名称	符号
0x00	ATIM 控制寄存器 1 （ATIM Control Register1）	ATIM_CR1
0x04	ATIM 控制寄存器 2 （ATIM Control Register1）	ATIM_CR2
0x08	ATIM 从机模式控制寄存器 （ATIM Slave Mode Control Register）	ATIM_SMCR
0x0C	ATIM DMA 和中断使能寄存器 （ATIM DMA and Interrupt Enable Register）	ATIM_DIER
0x10	ATIM 状态寄存器 （ATIM Interrupt Status Register）	ATIM_SR
0x14	ATIM 事件产生寄存器 （ATIM Event Generation Register）	ATIM_EGR
0x18	ATIM 捕捉/比较模式寄存器 1 （ATIM Capture/Compare Mode Register1）	ATIM_CCMR1
0x1C	ATIM 捕捉/比较模式寄存器 2 （ATIM Capture/Compare Mode Register2）	ATIM_CCMR2
0x20	ATIM 捕捉/比较使能寄存器 （ATIM Capture/Compare Enable Register）	ATIM_CCER
0x24	ATIM 计数器寄存器 （ATIM Counter Register）	ATIM_CNT
0x28	ATIM 预分频寄存器 （ATIM Prescaler Register）	ATIM_PSC
0x2C	ATIM 自动重载寄存器 （ATIM Auto-Reload Register）	ATIM_ARR
0x30	ATIM 重复计数寄存器 （ATIM Repetition Counter Register）	ATIM_RCR
0x34	ATIM 捕捉/比较寄存器 1 （ATIM Capture/Compare Register1）	ATIM_CCR1
0x38	ATIM 捕捉/比较寄存器 2 （ATIM Capture/Compare Register2）	ATIM_CCR2
0x3C	ATIM 捕捉/比较寄存器 3 （ATIM Capture/Compare Register3）	ATIM_CCR3
0x40	ATIM 捕捉/比较寄存器 4 （ATIM Capture/Compare Register4）	ATIM_CCR4
0x44	ATIM 刹车和死区控制寄存器 （ATIM Break and Deadtime Register）	ATIM_BDTR
0x48	ATIM DMA 控制寄存器 （ATIM DMA Control Register）	ATIM_DCR
0x4C	ATIM DMA 访问寄存器 （ATIM DMA Access Register）	ATIM_DMAR
0x60	ATIM 刹车输入控制寄存器 （ATIM Break Control Register）	ATIM_BKCTL

25.5.1 ATIM 控制寄存器 1（ATIM_CR1）

名称	ATIM_CR1

地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						CKD	
位权限	U-0						RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9:8	**CKD**	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比） 00：$t_{DTS}=t_{CK\_INT}$ 01：$t_{DTS}=2*t_{CK\_INT}$ 10：$t_{DTS}=4*t_{CK\_INT}$ 11：RFU，禁止使用
7	**ARPE**	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	**CMS**	计数器对齐模式选择 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	**DIR**	计数方向寄存器 0：向上计数 1：向下计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读
3	**OPM**	单脉冲输出模式 0：Update Event 发生时计数器不停止 1：Update Event 发生时计数器停止（自动清零 CEN）
2	**URS**	更新请求选择 0：以下事件能够产生 update 中断或 DMA 请求 - 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求
1	**UDIS**	禁止 update 0：使能 update 事件；以下事件发生时产生 update 事件

Bit	助记符	功能描述
		- 计数器上溢出或下溢出 - 软件置位 UG 寄存器 - 从机控制器产生 update 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。
0	**CEN**	计数器使能 0：计数器关闭 1：计数器使能 注意：外部触发模式可以自动置位 CEN

25.5.2 ATIM 控制寄存器 2（ATIM_CR2）

名称	ATIM_CR2							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	CCUS	-	CCPC
位权限	RW-0	RW-000			RW-0	RW-0	U-0	RW-0

Bit	助记符	功能描述
31:15	**--**	RFU：未实现，读为 0
14	**OIS4**	参考 OIS1
13	**OIS3N**	参考 OIS1N
12	**OIS3**	参考 OIS1
11	**OIS2N**	参考 OIS1N
10	**OIS2**	参考 OIS1
9	**OIS1N**	定义 OC1N 的输出 IDLE 状态 0：当 MOE=0 时，经过 dead time 后，OC1N=0 1：当 MOE=0 时，经过 dead time 后，OC1N=1
8	**OIS1**	定义 OC1 的输出 IDLE 状态 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=0 1：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=1
7	**TI1S**	ATIM 输入 TI1 选择 0：ATIM_CH1 引脚连接到 TI1 输入 1：ATIM_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：ATIM_EGR 的 UG 寄存器被用作 TRGO

Bit	助记符	功能描述
		001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	**CCDS**	捕捉/比较 DMA 选择 0：捕捉/比较事件发生时发送 DMA 请求 1：Update Event 发生时发送 DMA 请求
2	**CCUS**	捕捉/比较控制寄存器更新选择 0：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们仅在置位 COMG 寄存器时更新 1：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们在置位 COMG 寄存器或者 TRGI 上升沿时更新
1	--	RFU：未实现，读为 0
0	**CCPC**	捕捉/比较预装载控制 0：CcxE, CcxNE, OcxM 寄存器不使能 preload 1：CcxE, CcxNE, OcxM 寄存器使能 preload 注意：此寄存器仅在拥有互补输出功能的通道上有效

25.5.3 ATIM 从机模式控制寄存器（ATIM_SMCR）

名称	ATIM_SMCR							
地址	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
位权限	RW-0	RW-0	RW-00		RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			-	SMS		
位权限	RW-0	RW-000			U-0	RW-000		

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**ETP**	外部触发信号极性配置 0：高电平或上升沿有效 1：低电平或下降沿有效

Bit	助记符	功能描述
14	**ECE**	外部时钟使能 0：关闭外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿
13:12	**ETPS**	外部触发信号预分频寄存器 外部触发信号 ETRP 的频率最多只能是 ATIM 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8 分频
11:8	**ETF**	外部触发信号滤波时钟和长度选择 0000：无滤波 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	主机从机模式选 0：无动作 1：TRGI 触发的动作被延迟，以便于通过 TRGO 将当前定时器和从机定时器完美同步起来
6:4	**TS**	触发选择，用于选择同步计数器的触发源 000：内部触发信号（ITR0） 001：内部触发信号（ITR1） 010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2） 111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	**--**	RFU：未实现，读为 0
2:0	**SMS**	从机模式选择 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1

Bit	助记符	功能描述
		电平高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

25.5.4 ATIM DMA 和中断使能寄存器（ATIM_DIER）

名称	ATIM_DIER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
位权限	U-0				RW-0	RW-0	RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
18	**CC3BURSTEN**	捕捉比较通道 3 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
17	**CC2BURSTEN**	捕捉比较通道 2 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
16	**CC1BURSTEN**	捕捉比较通道 1 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	--	RFU：未实现，读为 0
14	**TDE**	外部触发 DMA 请求使能 0：从机模式下，禁止外部触发事件产生 DMA 请求 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器）

Bit	助记符	功能描述
13	**COMDE**	COM 事件 DMA 请求使能 0：COM 事件发生时，禁止产生 DMA 请求 1：COM 事件发生时，允许产生 DMA 请求
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 0：禁止 CC4 DMA 请求 1：允许 CC4 DMA 请求
11	**CC3DE**	捕捉比较通道 3 的 DMA 请求使能 0：禁止 CC3 DMA 请求 1：允许 CC3 DMA 请求
10	**CC2DE**	捕捉比较通道 2 的 DMA 请求使能 0：禁止 CC2 DMA 请求 1：允许 CC2 DMA 请求
9	**CC1DE**	捕捉比较通道 1 的 DMA 请求使能 0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 0：Update Event 发生时，禁止产生 DMA 请求 1：Update Event 发生时，允许产生 DMA 请求
7	**BIE**	刹车事件中断使能 0：禁止刹车事件中断 1：允许刹车事件中断
6	**TIE**	触发事件中断使能 0：禁止触发事件中断 1：允许触发事件中断
5	**COMIE**	COM 事件中断使能 0：禁止 COM 事件中断 1：允许 COM 事件中断
4	**CC4IE**	捕捉/比较通道 4 中断使能 0：禁止捕捉/比较 4 中断 1：允许捕捉/比较 4 中断
3	**CC3IE**	捕捉/比较通道 3 中断使能 0：禁止捕捉/比较 3 中断 1：允许捕捉/比较 3 中断
2	**CC2IE**	捕捉/比较通道 2 中断使能 0：禁止捕捉/比较 2 中断 1：允许捕捉/比较 2 中断
1	**CC1IE**	捕捉/比较通道 1 中断使能 0：禁止捕捉/比较 1 中断 1：允许捕捉/比较 1 中断
0	**UIE**	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

25.5.5 ATIM 状态寄存器（ATIM_ISR）

名称	**ATIM_ISR**							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0			RW-0	RW-0	RW-0	RW-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 0：无 overcapture 事件 1：在 CC1IF 标志为 1 的情况下发生新的捕捉
8	--	RFU：未实现，读为 0
7	**BIF**	刹车事件中断标志，硬件置位，软件写 1 清零
6	**TIF**	触发事件中断标志，硬件置位，软件写 1 清零
5	**COMIF**	COM 事件中断标志，硬件置位，软件写 1 清零
4	**CC4IF**	捕捉/比较通道 4 中断标志 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果 CC1 通道配置为输入：发生捕捉事件时置位，软件写 1 清零，或者软件读 ATIM_CCR1 自动清零。
0	**UIF**	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器 -URS=0 且 UDIS=0 的情况下，触发事件初始化计数器

25.5.6 ATIM 事件产生寄存器（ATIM_EGR）

名称	ATIM_EGR

地址	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
位权限	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7	**BG**	软件刹车，软件置位此寄存器产生刹车事件，硬件自动清零
6	**TG**	软件触发，软件置位此寄存器产生触发事件，硬件自动清零
5	**COMG**	软件 COM 事件，硬件置位，软件写 1 清零
4	**CC4G**	捕捉/比较通道 4 软件触发，参考 CC1G
3	**CC3G**	捕捉/比较通道 3 软件触发，参考 CC1G
2	**CC2G**	捕捉/比较通道 2 软件触发，参考 CC1G
1	**CC1G**	捕捉/比较通道 1 软件触发 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求
0	**UG**	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。

25.5.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR1							
地址	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE
14:12	**OC2M**	输出比较 2 模式配置，参考 OC1M
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE
9:8	**CC2S**	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7	**OC1CE**	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M**	输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低；在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC1REF 在 CNT<CCR1 时置低，否则置高；在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	**OC1PE**	输出比较 1 预装载使能 0：CCR1 preload 寄存器无效，CCR1 可以直接写入 1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC1FE**	输出比较 1 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC

Bit	助记符	功能描述
		注意：CC1S 仅在通道关闭时（CC1E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:12	IC2F	输入捕捉 2 滤波
11:10	IC2PSC	输入捕捉 2 预分频
9:8	CC2S	捕捉/比较 2 通道选择 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC3 映射到 TI2 10：CC2 通道配置为输入，IC3 映射到 TI1 11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	IC1F	输入捕捉 1 滤波 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	IC1PSC	输入捕捉 1 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	CC1S	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

25.5.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能

名称	ATIM_CCMR2
地址	0x1C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC4F				IC4PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC			
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

输出比较模式

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**OC4CE**	输出比较 4 清零使能，参考 OC1CE
14:12	**OC4M**	输出比较 4 模式配置，参考 OC1M
11	**OC4PE**	输出比较 4 预装载使能，参考 OC1PE
10	**OC4FE**	输出比较 4 快速使能，参考 OC1FE
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7	**OC3CE**	输出比较 3 清零使能 0：OC3REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC3REF
6:4	**OC3M**	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 000：输出比较寄存器 CCR3 和计数器 CNT 的比较结果不会影响输出 001：CCR3=CNT 时，将 OC1REF 置高 010：CCR3=CNT 时，将 OC1REF 置低 011：CCR3=CNT 时，翻转 OC1REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 –在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低；在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM 模式 2 –在向上计数时，OC3REF 在 CNT<CCR3 时置低，否则置高；在向下计数时，OC3REF 在 CNT>CCR3 时置高，否则置低
3	**OC3PE**	输出比较 3 预装载使能 0：CCR3 preload 寄存器无效，CCR3 可以直接写入 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是

Bit	助记符	功能描述
		访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	**OC3FE**	输出比较 3 快速使能 0：关闭快速使能，trigger 输入不会影响比较输出 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

输入捕捉模式

Bit	助记符	功能描述
31:16	**--**	RFU：未实现，读为 0
15:12	**IC4F**	输入捕捉 4 滤波
11:10	**IC4PSC**	输入捕捉 4 预分频
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 1 滤波 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位

Bit	助记符	功能描述
1:0	**CC3S**	捕捉/比较 3 通道选择 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC1 映射到 TI3 10：CC3 通道配置为输入，IC1 映射到 TI4 11：CC3 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

25.5.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER）

名称	ATIM_CCER							
地址	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
位权限	U-0		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:14	**--**	RFU：未实现，读为 0
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E
11	**CC3NP**	捕捉/比较 3 互补输出极性，参考 CC1NP
10	**CC3NE**	捕捉/比较 3 互补输出使能，参考 CC1NE
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E
7	**CC2NP**	捕捉/比较 2 互补输出极性，参考 CC1NP
6	**CC2NE**	捕捉/比较 2 互补输出使能，参考 CC1NE
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E
3	**CC1NP**	捕捉/比较 1 互补输出极性 0：OC1N 高电平为 active 1：OC1N 低电平为 active
2	**CC1NE**	捕捉/比较 1 互补输出使能 0：OC1N 无效，OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定 1：OC1N 有效
1	**CC1P**	捕捉/比较 1 输出极性 CC1 通道配置为输出时 0：OC1 高电平 active

Bit	助记符	功能描述
		1：OC1 低电平 active CC1 通道配置为输入时 0：非取反模式–捕捉在 IC1 的上升沿进行 1：取反模式–捕捉在 IC1 的下降沿进行
0	**CC1E**	捕捉/比较 1 输出使能 CC1 通道配置为输出时 0：OC1 使能 1：OC1 关闭 CC1 通道配置为输入时 0：关闭捕捉功能 1：使能捕捉功能

25.5.10 ATIM 计数器寄存器（ATIM_CNT）

名称	ATIM_CNT							
地址	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CNT**	计数器值

25.5.11 ATIM 预分频寄存器（ATIM_PSC）

名称	ATIM_PSC							
地址	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	RW-00000000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

25.5.12 ATIM 自动重载（auto-reload）寄存器（ATIM_ARR）

名称	ATIM_ARR							
地址	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	RW-11111111							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

25.5.13 ATIM 重复计数寄存器（ATIM_RCR）

名称	ATIM_RCR							
地址	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	REP[7:0]
位权限	RW-00000000

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**REP**	重复计数值。REP 不为 0 时，每次 update 条件发生时 REP 递减，当 REP=0 时触发 update 事件

25.5.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1）

名称	ATIM_CCR1							
地址	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR1**	捕捉/比较通道 1 寄存器 **如果通道 1 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC1 输出 **如果通道 1 配置为输入：** CCR1 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR1 为只读

25.5.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2）

名称	ATIM_CCR2							
地址	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	RW-00000000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR2**	捕捉/比较通道 2 寄存器 **如果通道 2 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 **如果通道 2 配置为输入：** CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

25.5.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）

名称	ATIM_CCR3							
地址	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR3**	捕捉/比较通道 3 寄存器 **如果通道 3 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

25.5.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4）

名称	ATIM_CCR4							
地址	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR4**	捕捉/比较通道 4 寄存器 **如果通道 4 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC4 输出 **如果通道 4 配置为输入：** CCR4 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR4 为只读

25.5.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR）

名称	ATIM_BDTR							
地址	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DTG							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**MOE**	输出使能总控制位 此寄存器控制所有通道的输出使能，每个通道独立的输出使能还需要 CcxE 和 CcxNE 来控制。MOE 由软件置位，或者在 AOE=1 的情况下硬件触发自动置位。当刹车输入有效时，MOE 被硬件异步清零。 0：关闭 OC 和 OCN 输出，具体 IO 输出状态由 OSSI 决定 1：使能 OC 和 OCN 输出（仍需各个通道的 CcxE 和 CcxNE 状态来决定是否输出）
14	**AOE**	自动输出使能

Bit	助记符	功能描述
		0：MOE 仅能由软件置位 1：MOE 可以软件置位，或者由 update 事件自动置位
13	**BKP**	刹车极性 0：刹车输入为低电平有效 1：刹车输入为高电平有效
12	**BKE**	刹车使能 0：禁止刹车输入 1：允许刹车输入
11	**OSSR**	运行状态下的输出关闭状态选择 仅在 MOE=1 的情况下，针对使能了互补输出的通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 驱动 GPIO 为无效状态
10	**OSSI**	IDLE 状态下的输出关闭状态选择 仅在 MOE=0 的情况下，针对输出通道有效。 0：输出通道不使能时，OC 和 OCN 不驱动 GPIO 1：输出通道不使能时，OC 和 OCN 先驱动无效电平，待死区时间结束后，OC=OISx 并且 OCN=OISxN
9:8	**LOCK**	寄存器写保护配置 00：无写保护 01：保护等级 1 – DTG, OISx, OISxN, BKE, BKP, AOE 不能改写 10：保护等级 2 –在等级 1 基础上，CCxP, CCxNP, OSSR, OSSI 不能改写 11：保护等级 3 –在等级 2 基础上，OcxM, OcxPE 在相应通道配置为输出时不能改写 注意：LOCK 寄存器在复位后仅能改写 1 次，写保护后的寄存器只有在下一次芯片复位后才能重新写入。
7:0	**DTG**	死区时间插入，用于配置互补输出插入的死区时间长度 DTG[7:5]=0xx：DT=DTG[7:0] * t_{DTS} DTG[7:5]=10x：DT=(64+DTG[5:0]) * 2 * t_{DTS} DTG[7:5]=110：DT=(32+DTG[4:0]) * 8 * t_{DTS} DTG[7:5]=111：DT=(32+DTG[4:0]) * 16 * t_{DTS}

25.5.19 ATIM DMA 控制寄存器（ATIM_DCR）

名称	ATIM_DCR							
地址	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DBL				
位权限	U-0			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				

位权限	U-0	RW-00000

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 对 ATIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~18 00000：长度=1 00001：长度=2 …… 10001：长度=18 其他：无效值，禁止写入
7:5	--	RFU：未实现，读为 0
4:0	**DBA**	DMA 基地址，定义指向寄存器的偏移地址 00000：ATIM_CR1 00001：ATIM_CR2 00010：ATIM_SMCR …… 注意：当 DBA+DBL 超出了 ATIM 寄存器地址范围，则实际 burst 传输到 ATIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

25.5.20 ATIM DMA 访问寄存器（ATIM_DMAR）

名称	ATIM_DMAR							
地址	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR[31:24]							
位权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR[23:16]							
位权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:0	**DMAR**	DMA burst 访问寄存器 在使用 DMA burst 传输时，将 DMA 通道外设地址设置为 ATIM_DMAR，ATIM 会根据 DBL 的值产生多次 DMA 请求

25.5.21 ATIM 刹车输入控制寄存器（ATIM_BKCTL）

名称	ATIM_BKCTL
地址	0x60

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						BRK2GATE	BRK1GATE
位权限	U-0						RW-1	RW-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BRKF				BRKCOMB	HFDET_BRKEN	SVD_BRKEN	COMP_BRKEN
位权限	RW-0000				RW-0	RW-0	RW-0	RW-0

Bit	助记符	功能描述
31:10	-	RFU，未实现，读为 0
9	**BRK2GATE**	ATIM_BRK2 引脚输入门控信号 0：将 ATIM_BRK2 的输入门控成 0 1：不门控
8	**BRK1GATE**	ATIM_BRK1 引脚输入门控信号 0：将 ATIM_BRK2 的输入门控成 0 1：不门控
7:4	**BRKF**	刹车信号的滤波时钟和长度选择 0000：无滤波 0001：$f_{SAMPLING}=f_{CK\_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK\_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK\_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3	**BRKCOMB**	刹车组合控制 0：两路刹车信号相或 1：两路刹车信号相与
2	**HFDET_BRKEN**	XTHF 停振检测刹车信号使能 0：禁止 HFDET 刹车信号 1：使能 HFDET 刹车信号
1	**SVD_BRKEN**	SVD 刹车信号使能 0：禁止 SVD 刹车信号 1：使能 SVD 刹车信号
0	**COMP_BRKE**	比较器输出刹车信号使能

Bit	助记符	功能描述
	N	0：禁止比较器刹车信号 1：使能比较器刹车信号

26 定时器阵列单元（TAU）

26.1 概述

TAU是16bit定时器组成的阵列，支持以下功能

- 2组共12个独立的16bit向上计数器（TAU00~TAU07，TAU10~TAU13）
- Auto-reload定时
- 外部事件计数
- 输入边沿捕捉（事件触发、自由计数）
- 输入预分频
- 脉冲宽度或周期捕捉（PWC模式）
- 脉冲宽度调制（PWM模式）
- 可以级联成32bit定时器

8个独立定时器结构相同，仅是输入输出通道不同；每个独立定时器包含计数器、自动重载寄存器、捕捉/比较寄存器。定时器启动后从0开始计数，计数至自动重载寄存器的值溢出信号，同时计数器回零重新开始计数。定时器允许在计数过程中改变计数初值，完成动态的定时时间更新；

支持2种边沿捕捉模式：

✓ 事件触发捕捉：使能后计数器保持0，捕捉到第一个有效沿之后定时器才开始计数；

✓ 自由计数捕捉：定时器使能后立即由0开始计数，在有效捕捉信号沿到来时锁存当前计数值，同时产生捕捉中断，在下一有效捕捉沿到来时再次锁存当前计数值并产生捕捉中断，在计数溢出后产生溢出中断。

脉冲宽度测量模式（PWC）在第一个有效沿到来后开始计数，第二个有效沿到来后停止计数，两个有效沿可以独立配置上升或下降，即可以实现正脉冲宽度捕捉、负脉冲宽度捕捉、正沿周期捕捉、负沿周期捕捉。

每个定时器支持外部引脚输入和外部引脚输出。其输入信号也可以重定向到内部信号，用于内部信号的测量和捕捉功能。

26.2 结构框图

TAU整体结构示意图如下，包含TAU0和TAU1两个阵列。

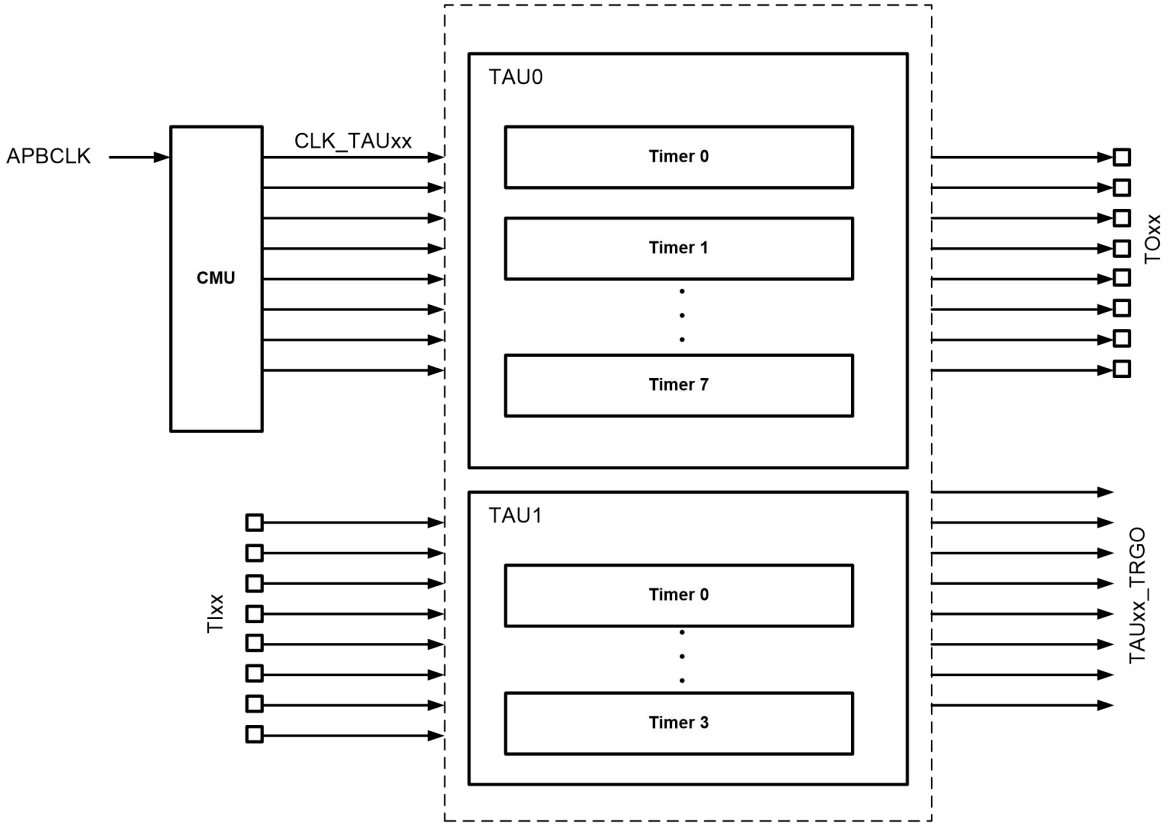


图 26-1TAU 结构框图

独立定时器结构框图如下：

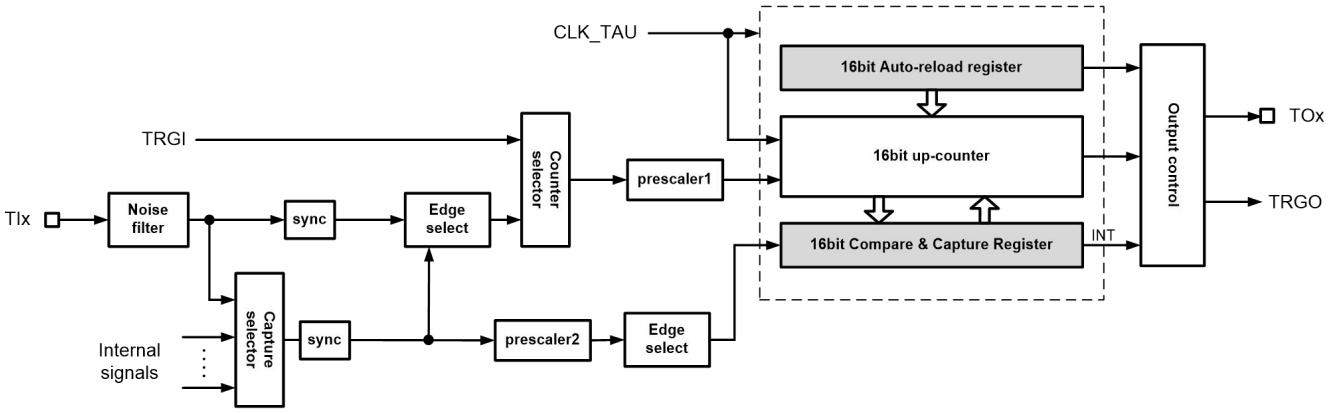


图 26-2TimerX 结构框图

定时器支持级联模式，此时一个16位定时器作为主机，一个或者多个16位定时器作为从机，从而实现32位级联定时器的功能。级联模式下，以master输出的溢出中断作为slave的计数触发信号，同时

将输入到master的捕捉信号连接到slave的捕捉信号输入。

26.3 功能描述

26.3.1 时钟和复位

TAU使用APBCLK工作，在使能TAU开始工作前，必须使能其总线时钟，即置位TAUx_PCE寄存器，参见CMU章节。比如想要使能TAU0中的Timer0工作，应置位TAU0_PEC寄存器。

通过TAUx_RST寄存器可以实现定时器的复位，参见RMU章节。如果想要使能TAU0中的Timer3工作，应清零TAU0_RST寄存器。

26.3.2 工作模式

定时器支持以下工作模式：

- 计数模式
  - 单次计数和连续计数
  - 内部时钟计数
  - 外部信号计数
  - 内部信号计数
  - 定时器级联计数
- 捕捉模式
  - 周期捕捉
  - 电平宽度捕捉
  - 单次捕捉和连续捕捉
- 比较模式
  - PWM输出
  - 单次脉冲输出
  - 外部信号分频

26.3.3 输入输出通道

每个定时器包含多个输入通道：

- 外部引脚输入ETIx：用于计数或者捕捉
- 内部信号输入：用于计数或者捕捉
- 其他定时器输入：用于级联计数或级联捕捉

每个定时器包含2个输出通道：

- 外部引脚输出ETOx：用于波形输出
- 触发信号输出：用于级联或其他外设定时触发

26.3.4 16 位计数模式

定时器计数模式启动后从0开始计数，计数值达到ARR时产生计数溢出中断。

定时器计数模式支持单次计数和连续计数功能。单次计数（one-shot）在定时器被触发后

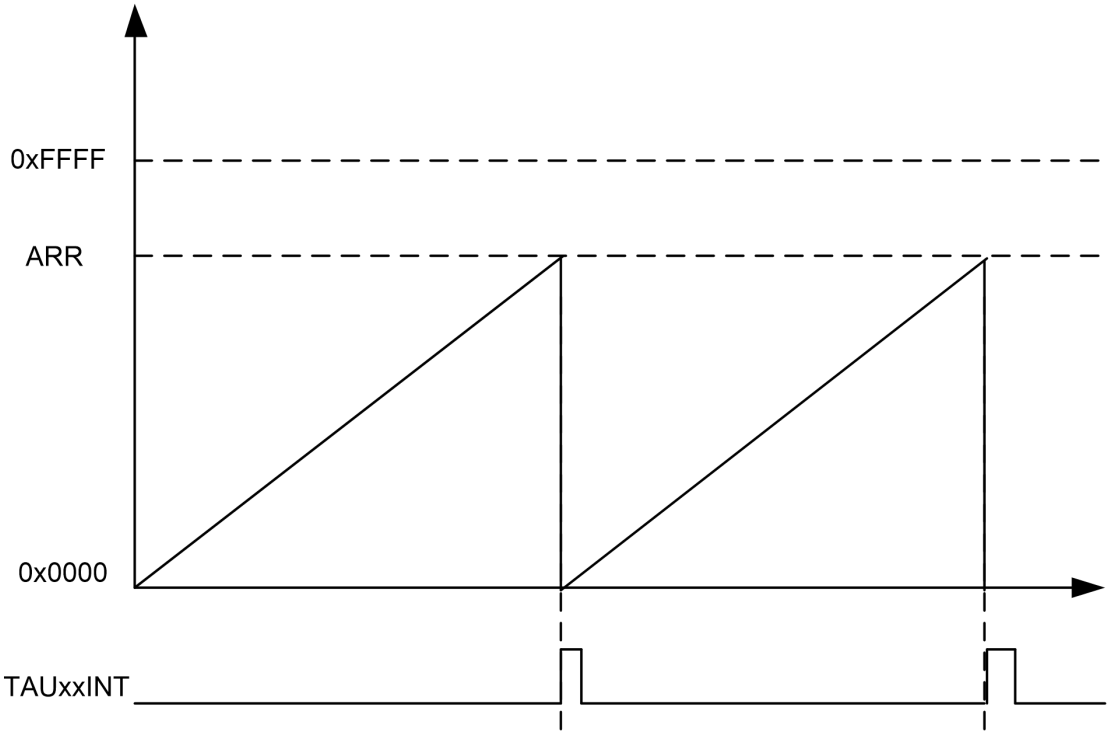


图 26-3 16bit 连续定时功能

定时器计数源可以基于工作时钟（及其分频）、外部输入信号、内部输入信号、Master定时器触发信号。

内部时钟计数

计数源通过CNTSEL寄存器选择为APBCLK时，定时器在每个APBCLK上升沿递增。此时如果配置了Prescale1（预分频寄存器1），则定时器以分频后的频率递增，如下图所示。

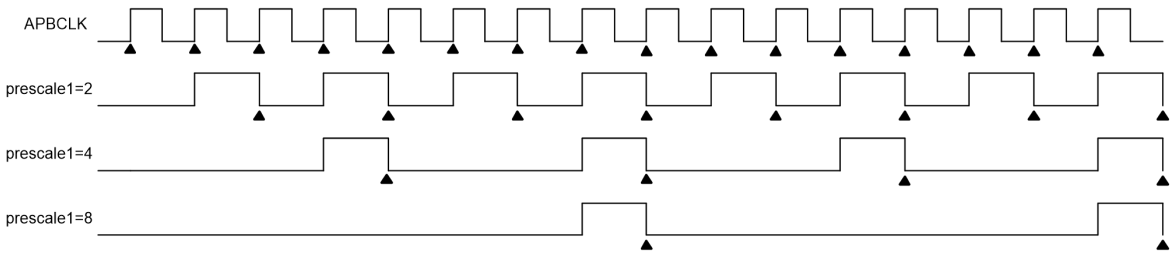


图 26-4 定时器内部时钟预分频计数

上图显示了基本定时功能，分别使用APBCLK不分频、2分频、4分频、8分频作为计数源，黑色三角表示实际计数器递增的位置，由于定时器是同步工作的，计数器递增都发生在APBCLK的上升沿。

外部输入信号计数

定时器可以基于外部输入信号计数，可以选择使用外部信号的上升沿、下降沿、或者上升下降沿计数，并且同样可以实现分频计数。

下图为ETIx上升沿事件计数的例子，黑色三角表示计数器递增位置。为了实现同步计数，输入信号在进行边沿检测前首先被APBCLK采样同步。

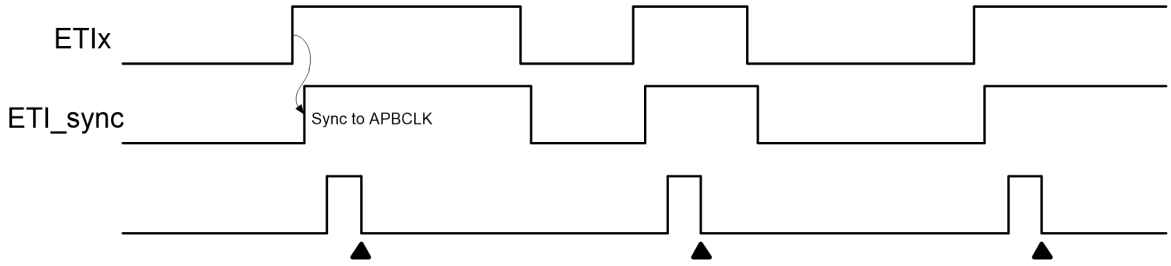


图 26-5 定时器外部事件上升沿递增时序

下图为ETIx上升下降沿事件计数，并且分频系数为2的例子。

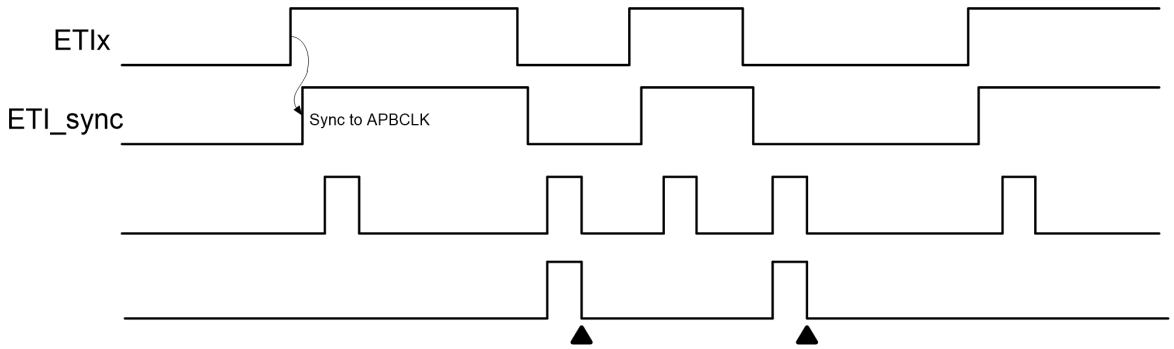


图 26-6 定时器外部事件上升下降沿递增，2 分频时序

内部输入信号计数

通过配置CNTSEL寄存器也可以实现对内部信号特定边沿的计数，比如内部时钟、比较器输出、UART

接收数据等。

下图为定时器在比较器输出下降沿计数，可实现累计模拟输入信号向下翻越比较器比较基准的次数。

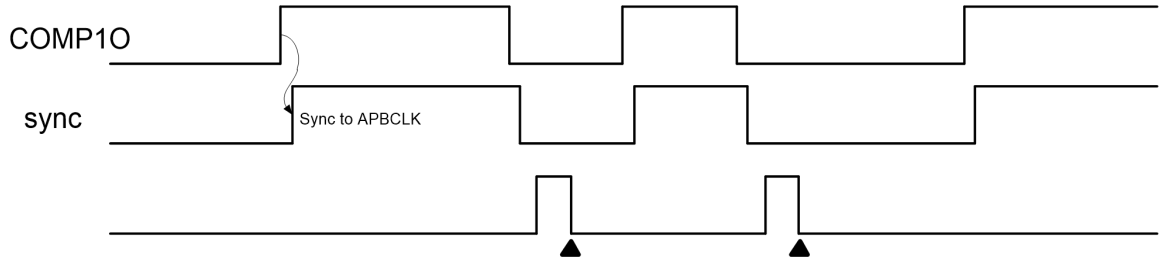


图 26-7 定时器内部信号下降沿递增

26.3.5 触发信号输出（TRGO）

TAU的每个16bit独立定时器都可以输出触发信号，当定时器计数值匹配CCR时，产生一个TRGO信号，宽度为1个APBCLK周期。此信号可以连接到TAU的其他定时器用于形成32bit级联定时器，或者输出到其他外设模块，用于定时触发其他外设工作。

TAU的12个独立定时器共可以输出12个TRGO：TAUxx_TRGO

TAU输出的TO信号除了可以连接搭配GPIO之外，也可以连接到其他外设作为异步触发信号使用。

26.3.6 32 位级联模式

级联计数模式下，作为master的定时器在计数器匹配CCR时发送TRGO触发信号，其他定时器如果使能slave模式，会在接收到TRGO信号后进行计数器递增。

从机模式有两种工作方式：触发递增（trigger to increment）和触发启动（trigger to start），通过SLVMD寄存器配置。

- 触发递增：每个触发信号导致从机计数器+1
- 触发启动：触发信号导致从机计数器从0开始计数，直到计数值等于ARR停止

下图是触发递增模式示例，以TAU0x为master，TAU0y为slave，当slave计数值到达ARR后产生从机OVIF，并回到0x0000重新计数。

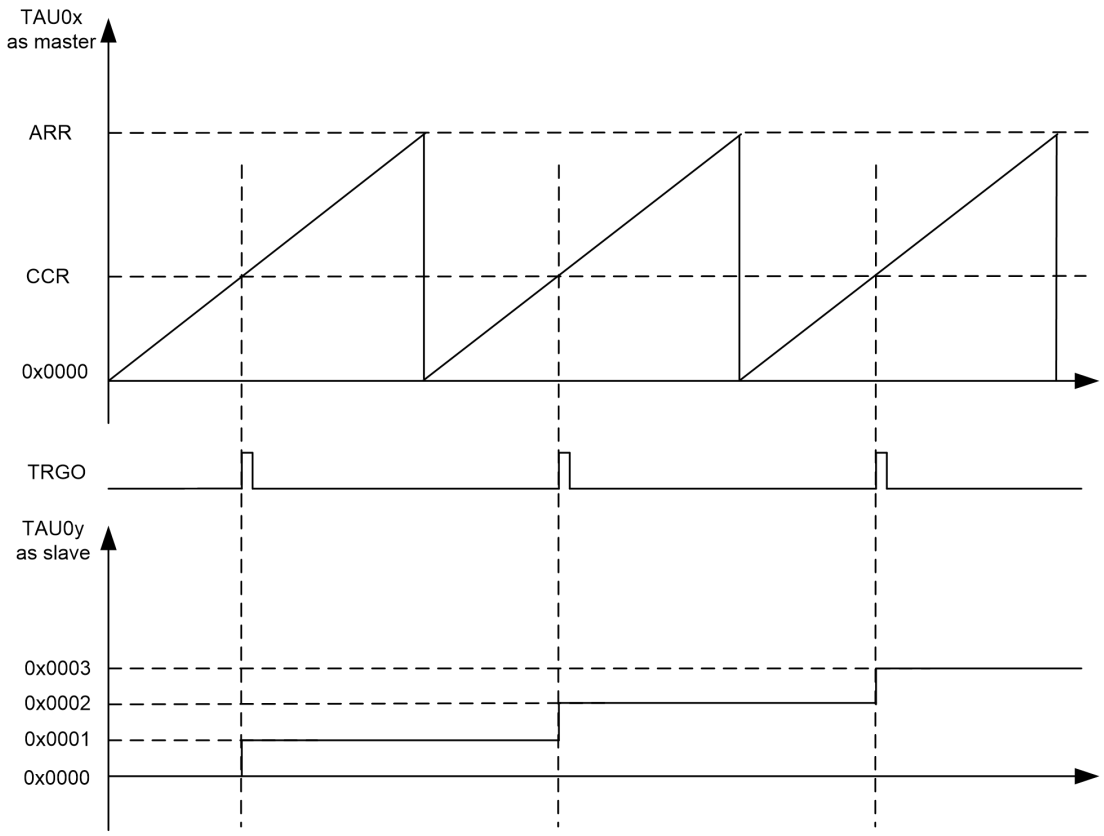


图 26-8 定时器级联计数（trigger to increment）

如果从机配置了计数源预分频，则主机多次触发才会导致从机递增一次。

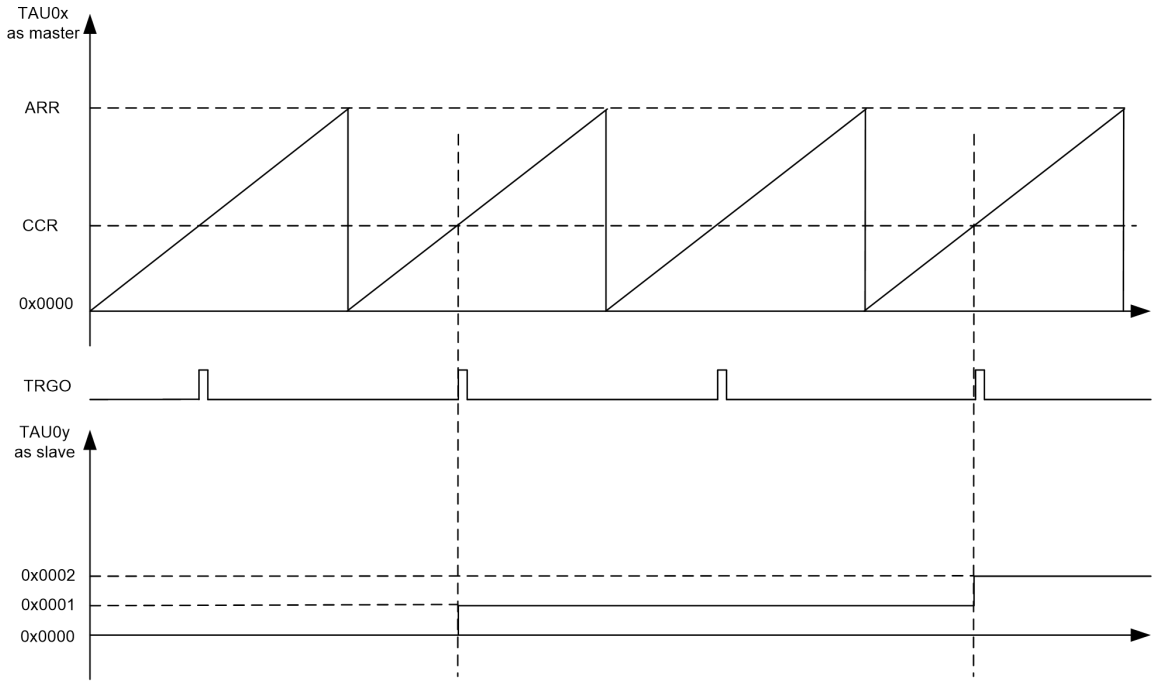


图 26-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1

下图是触发启动模式示例，以TAU0x为master，TAU0y为slave，当slave计数值到达ARR后产生从机OVIF，并回到0x0000等待下一个触发信号。

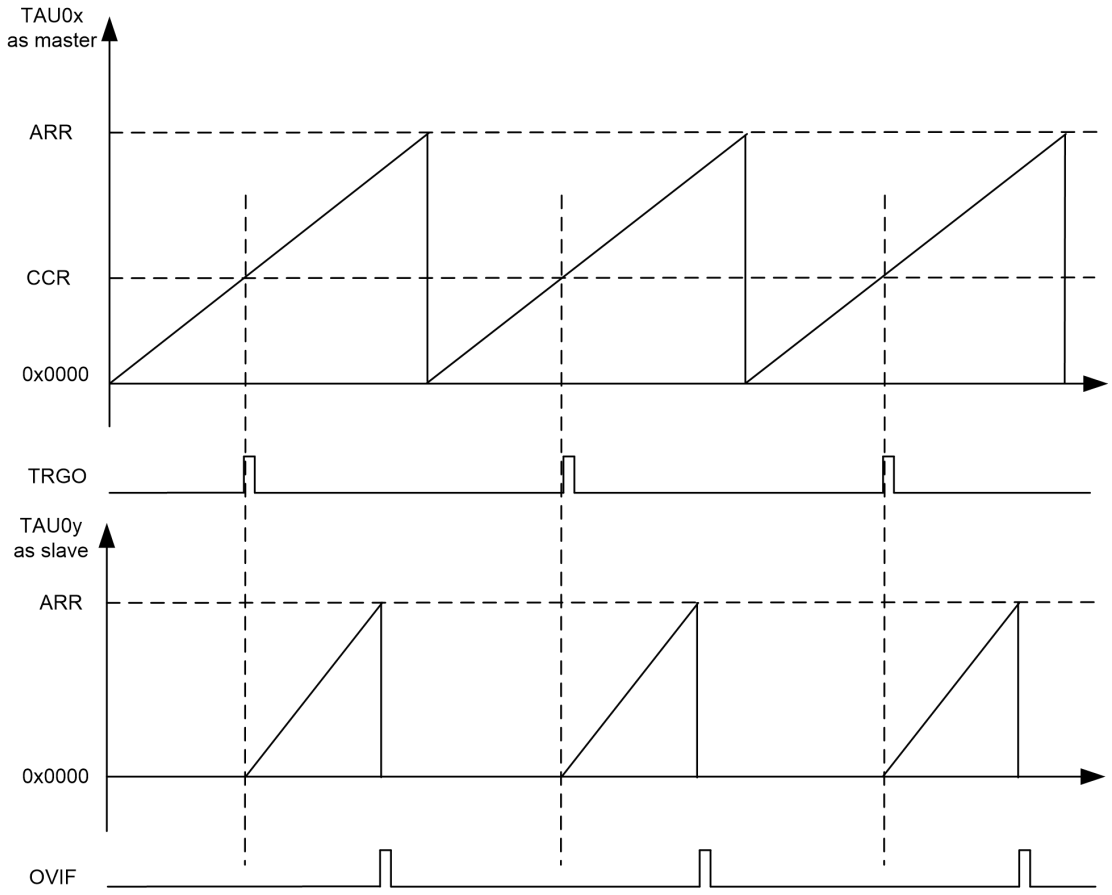


图 26-10 定时器级联计数（trigger to start）

注意，在触发启动模式下，软件应该合理配置主机从机的ARR参数，否则可能导致从机永远无法达到ARR，则从机OVIF永远不会置起，如下图所示。

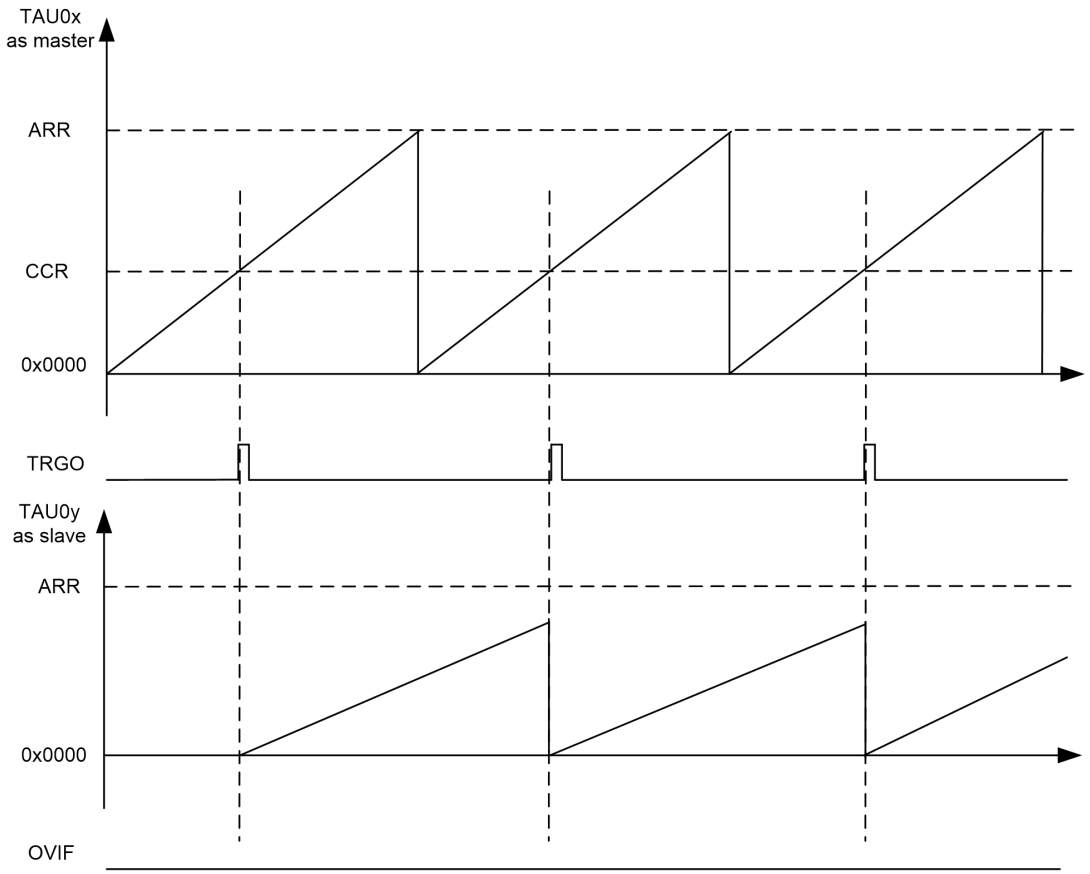


图 26-11 定时器级联计数（trigger to start），从机无法产生溢出

触发启动（trigger to start）模式下，主机和从机也可以配置不同的计数源预分频值，此时从机被触发后，计数器递增频率与主机不同。

下图为主机计数频率较高，而从机计数频率较低的例子。

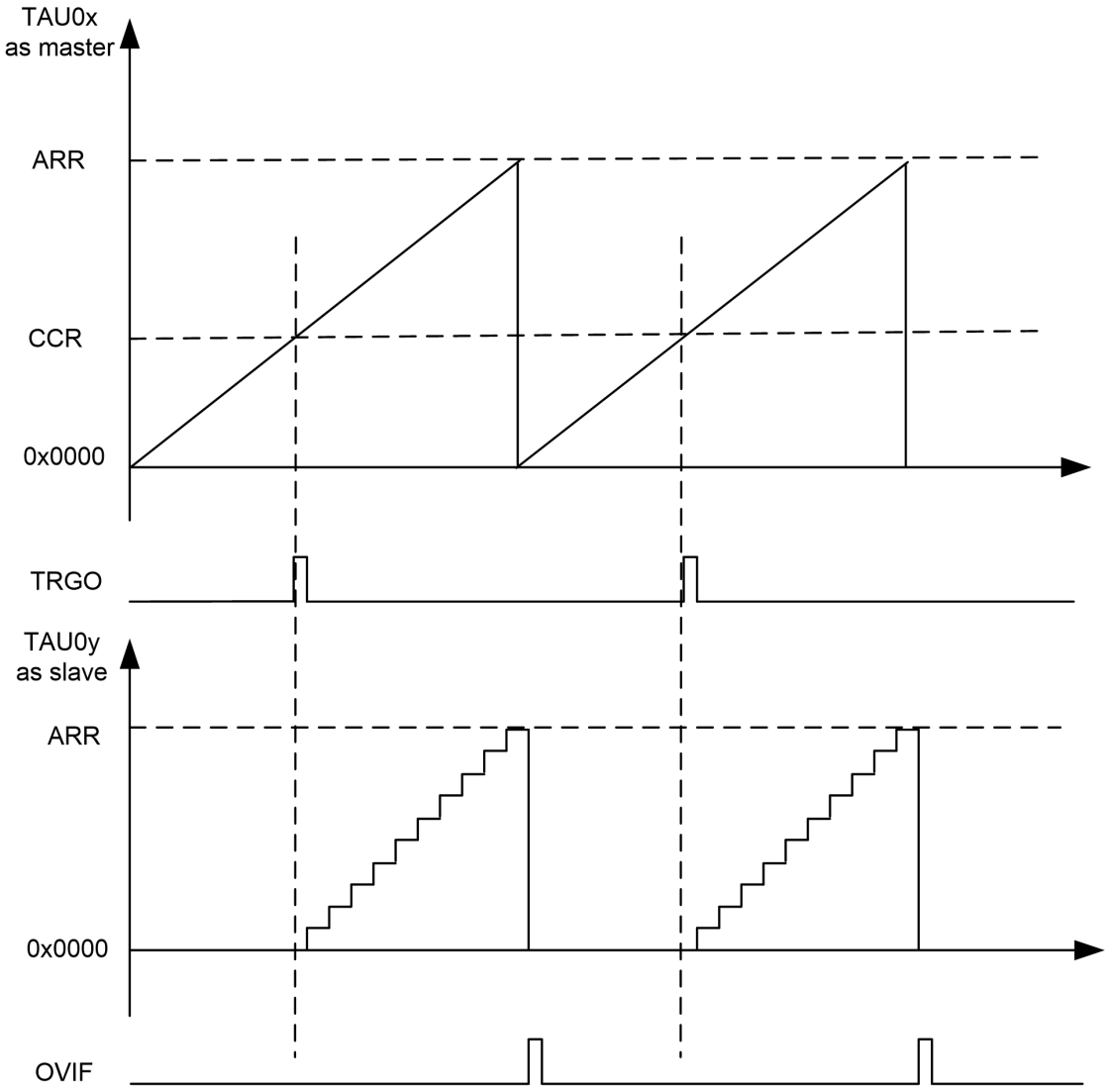


图 26-12 定时器级联计数（trigger to start），从机计数预分频

注意：只有TAU0支持级联计数模式，TAU1不支持级联计数。

TS寄存器（Trigger Select）用于选择从机模式下的主机触发源，每个定时器有2bit TS，可以选择4个主机触发输入。定时器0不支持从机模式，定时器1~7的主机触发输入选择参照以下表格。

TAU0 定时器#	TS			
	00	01	10	11
0	-	-	-	-
1	Timer0	-	-	-
2	Timer0	Timer1	-	-
3	Timer0	Timer1	Timer2	-
4	Timer0	Timer1	Timer2	Timer3
5	Timer0	Timer2	Timer3	Timer4

TAU0 定时器#	TS			
	00	01	10	11
6	Timer0	Timer3	Timer4	Timer5
7	Timer0	Timer4	Timer5	Timer6

表 26-1 定时器级联触发选择

注意：如果需要在级联模式下进行捕捉，则必须将输入到master的被捕捉信号同时连接到slave的捕捉信号输入。

26.3.7 捕捉功能

定时器支持两种捕捉模式：

- ✓ 清零捕捉：使能后计数器保持0，捕捉到第一个有效沿之后定时器才开始计数；并且每次捕捉事件将自动清零计数器。
- ✓ 自由计数捕捉：定时器使能后立即由0开始计数，在有效捕捉信号沿到来时锁存当前计数值，同时产生捕捉中断，在下一有效捕捉沿到来时再次锁存当前计数值并产生捕捉中断，在计数溢出后产生溢出中断。

捕捉事件发生时，当前计数值被保存在TAUx_TyCCR寄存器中供软件读取。

定时器使用APBCLK采样其输入信号进行计数，所以应用中应避免被采样的输入信号频率高于APBCLK时钟频率。

软件可以配置信号捕捉采用上升沿、下降沿、还是同时捕捉上升下降沿。对被捕捉的信号边沿还可以进行预分频。

周期捕捉模式（MD=10）用于捕捉输入信号的周期，即相同边沿之间的间隔。

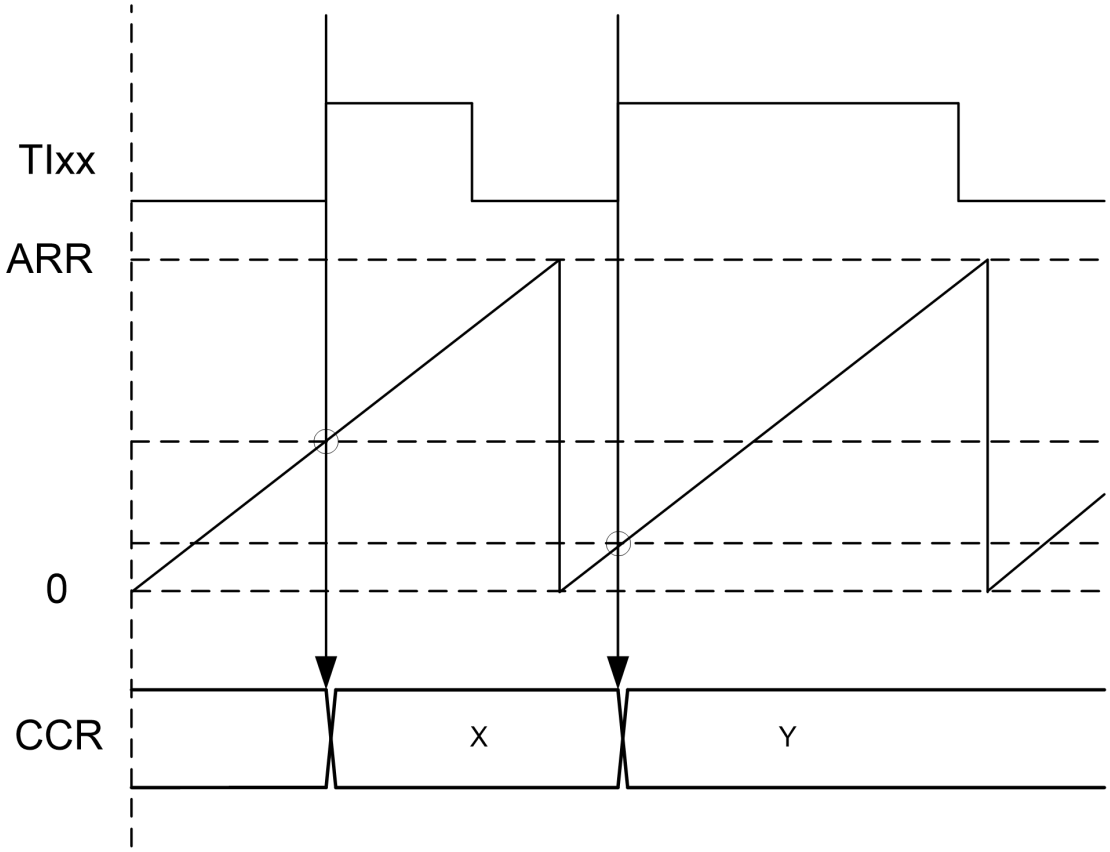


图 26-13 定时器捕捉外部输入信号上升沿

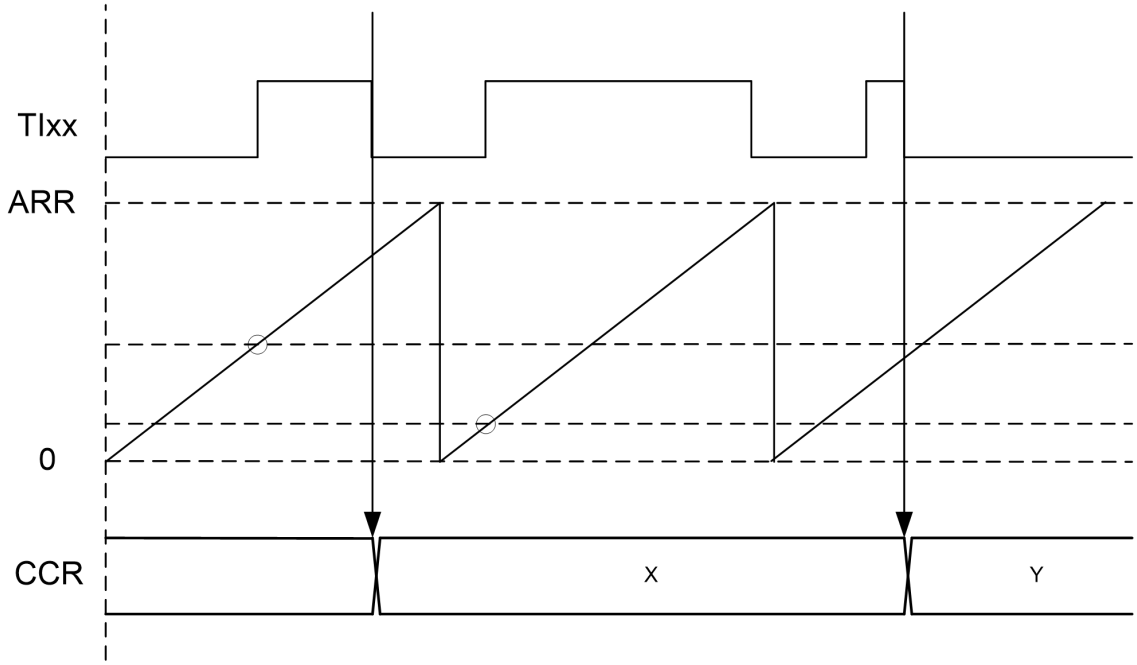


图 26-14 定时器 2 分频捕捉外部输入信号下降沿

脉冲宽度捕捉（PWC）功能可以用于测量输入信号的高电平宽度或者低电平宽度，需要配置MD寄存器为11。PWC模式下，如果捕捉沿配置为上升沿，则定时器在捕捉到第一个上升沿时开始计数，紧接着在捕捉到第一个下降沿时停止计数，此时定时器计数值就表示输入信号高电平的宽度；如果捕捉沿配置为下降沿，则定时器在捕捉到第一个下降沿时开始计数，紧接着在捕捉到第一个上升沿时停止计数，此时定时器计数值就表示输入信号低电平的宽度；如果捕捉沿配置为上升下降沿，则定时器在捕捉到第一个任意边沿时开始计数，紧接着在捕捉到下一个边沿时停止计数，此时定时器计数值就表示输入信号电平的宽度。

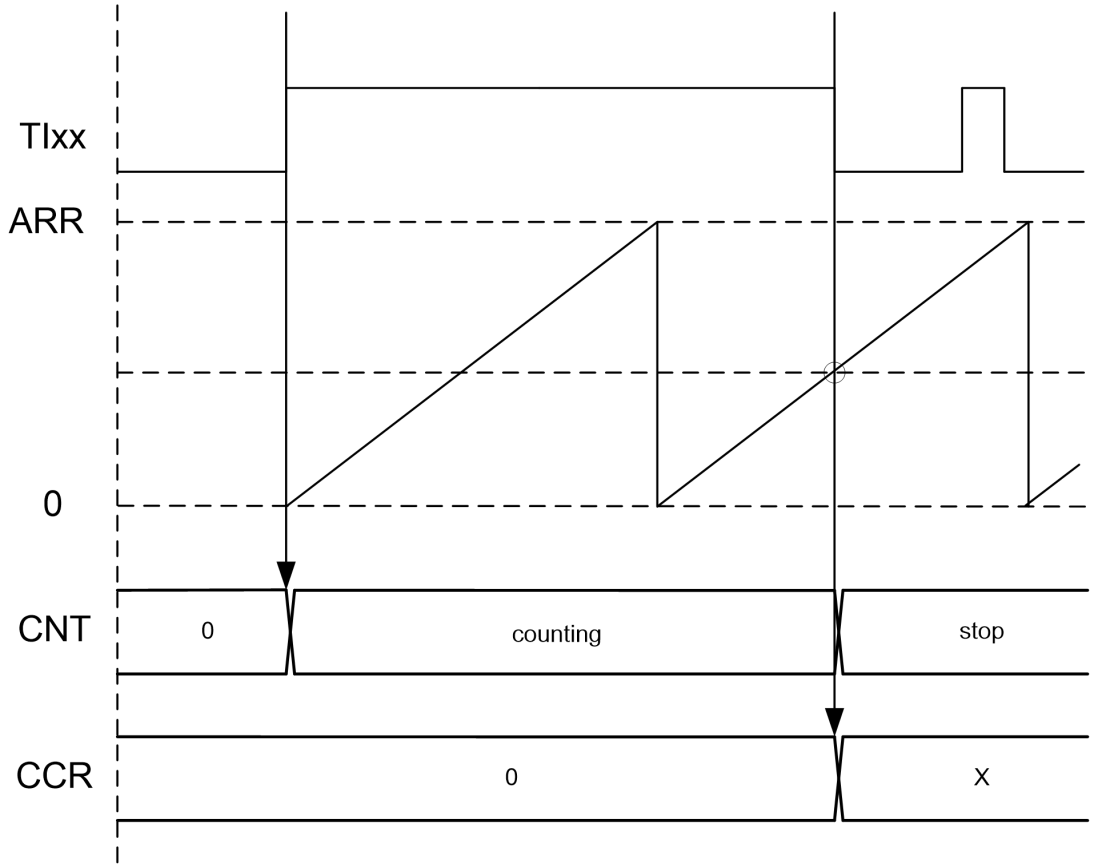


图 26-15 定时器脉冲宽度高电平捕捉

捕捉功能支持对内部信号的捕捉，通过TAU配置寄存器中的CAPSEL寄存器，可以选择每个定时器的捕捉源。

TAU0定时器	CAPSEL	被捕捉信号
Timer0	000	TI00
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP

Timer1	000	TI01
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer2	000	TI02
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer3	000	TI03
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer4	000	TI04
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	-
	110	-
	111	-
Timer5	000	TI05
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	-
	110	-
	111	-
Timer6	000	TI06
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-
	111	-
Timer7	000	TI07
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-

	111	-

表 26-2 定时器捕捉源选择

TAU1定时器	CAPSEL	被捕捉信号
Timer0	000	TI10
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer1	000	TI11
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer2	000	TI12
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer3	000	TI13
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP

表 26-3 定时器捕捉源选择

单次捕捉和连续捕捉

捕捉模式下CAPONCE寄存器可以选择单次捕捉模式或连续捕捉模式。当CAPONCE=1时，定时器完成一个捕捉周期后（周期捕捉完成一个周期捕捉，脉宽捕捉完成电平宽度捕捉），定时器自动关闭。当CAPONCE=0，定时器连续工作，直到软件关闭。

26.3.8 预分频

定时器可以对计数源和捕捉源进行预分频。

当对计数源进行预分频时，每隔特定数量的计数事件，才会进行一次计数值递增。下图以TI00上升沿计数、2分频为例：

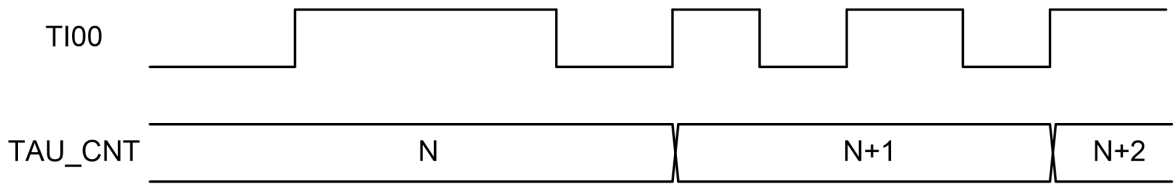


图 26-16TI00 二分频计数

当对捕捉源进行预分频时，每隔特定数量的捕捉事件，才会进行一次计数值捕捉。下图是对TI00上升沿进行捕捉、2分频的例子：

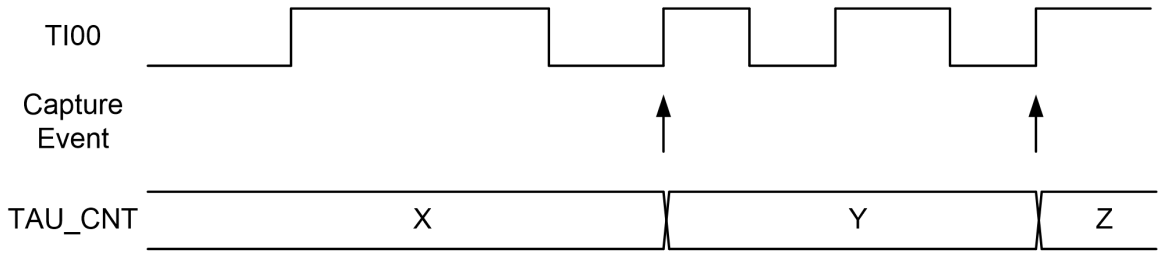


图 26-17TI00 二分频捕捉

26.3.9 16 位 PWM 输出

单个TAU支持独立16bitPWM输出，启动PWM之后定时器从初值开始计数，当计数值等于比较寄存器时，PWM输出置1，当计数值溢出时PWM输出置0，16bit PWM如下图所示：

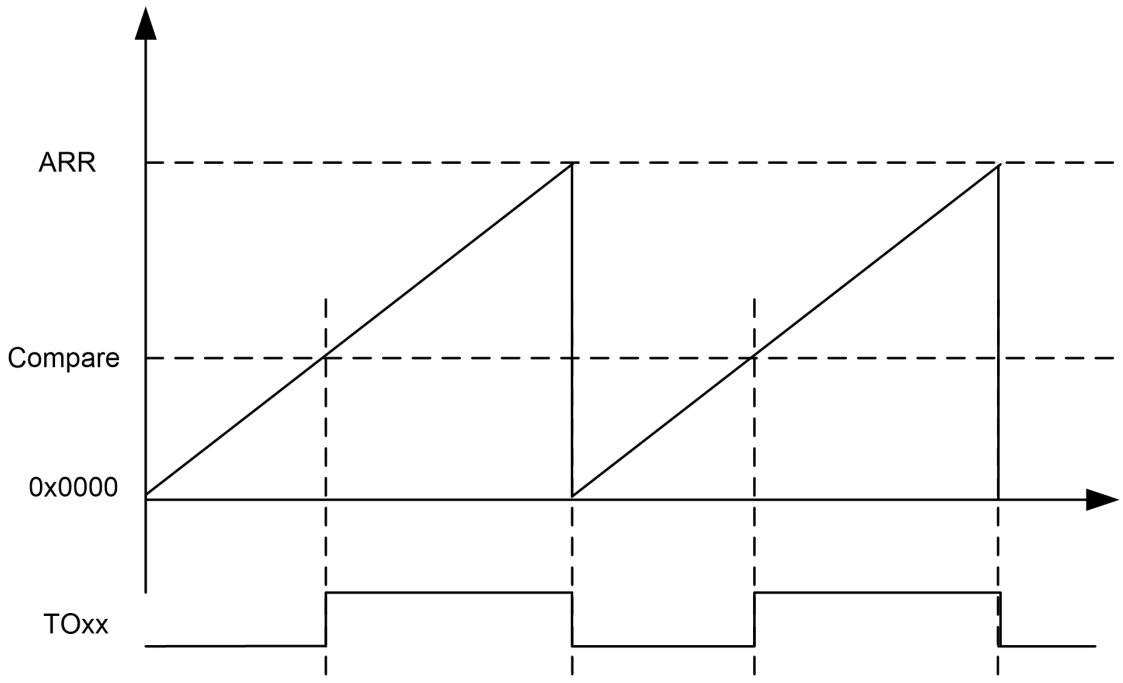


图 26-1816 位 PWM 输出示意图

PWM周期由ARR寄存器决定：

$$Period_{PWM} = T_{CNT} \times (\mathrm{ARR}+1)$$

PWM占空比由初值寄存器和比较寄存器共同决定：

$$Duty_{PWM} = T_{CNT} \times \frac{(ARR+1)-CCR}{ARR+1}$$

其中T_{CNT}为计数时钟周期。输出PWM信号极性可以取反，由OPOL寄存器控制。以上公式为输出极性不取反的情况，如果输出极性取反，则占空比变为1-$Duty_{PWM}$。

26.3.10 级联 PWM 输出

通过级联模式，两个独立定时器可以组合输出更为复杂的PWM波形。

比如将TAU0x和TAU0y级联，以TAU0x为主机，TAU0y采用触发启动模式，并且配置为PWM输出。从机接收到触发信号后开始计数，计数到ARR后自动清零并停止计数，直到下一次触发信号到来。下图中TAU0y的TO能够输出以ARR0x为周期，高电平宽度为ARR0y-CCR0y，前半部分低电平宽度为CCR0x+CCR0y，后半部分低电平宽度为ARR0x-ARR0y-CCR0x的特殊非对称PWM波形。

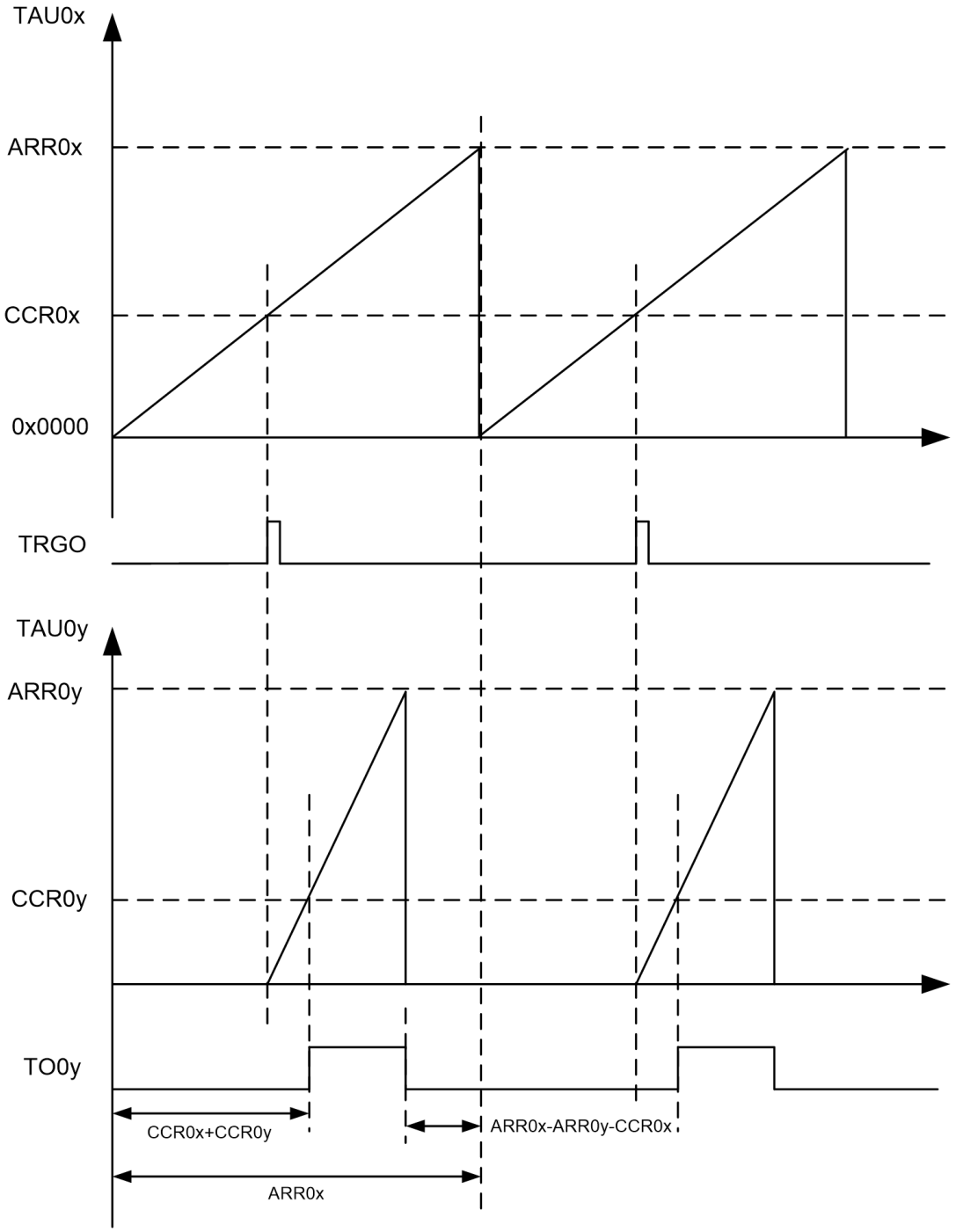


图 26-19 非对称级联 PWM 输出示意图

使用级联PWM输出时，软件必须合理配置主机和从机的ARR值，以获得符合期望的波形输出。下图为从机ARR大于主机ARR的例子。

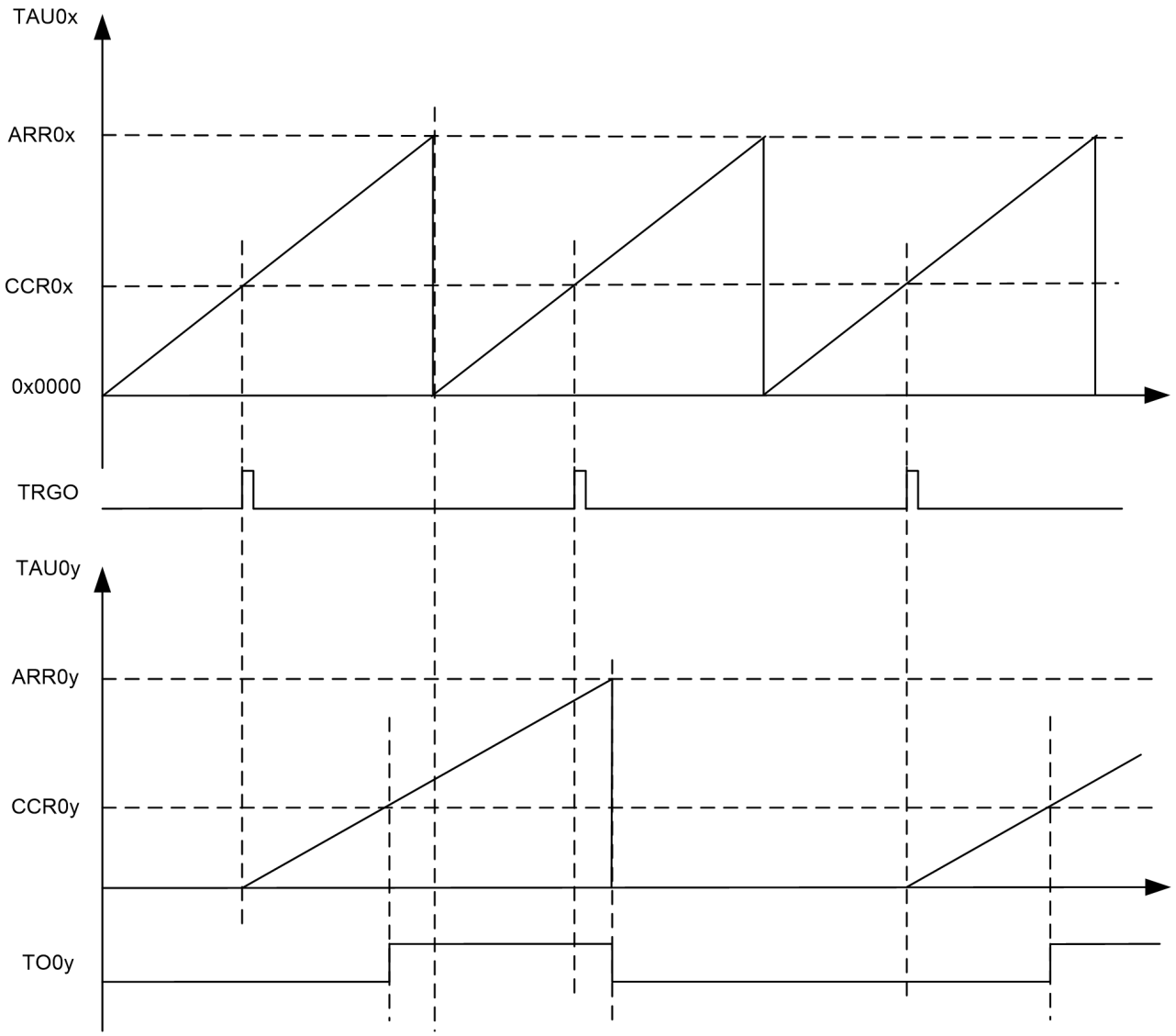


图 26-19 从机 ARR 大于主机的情况

级联定时器再搭配第三个独立定时器，可以输出两路非交叠PWM，如下图所示。

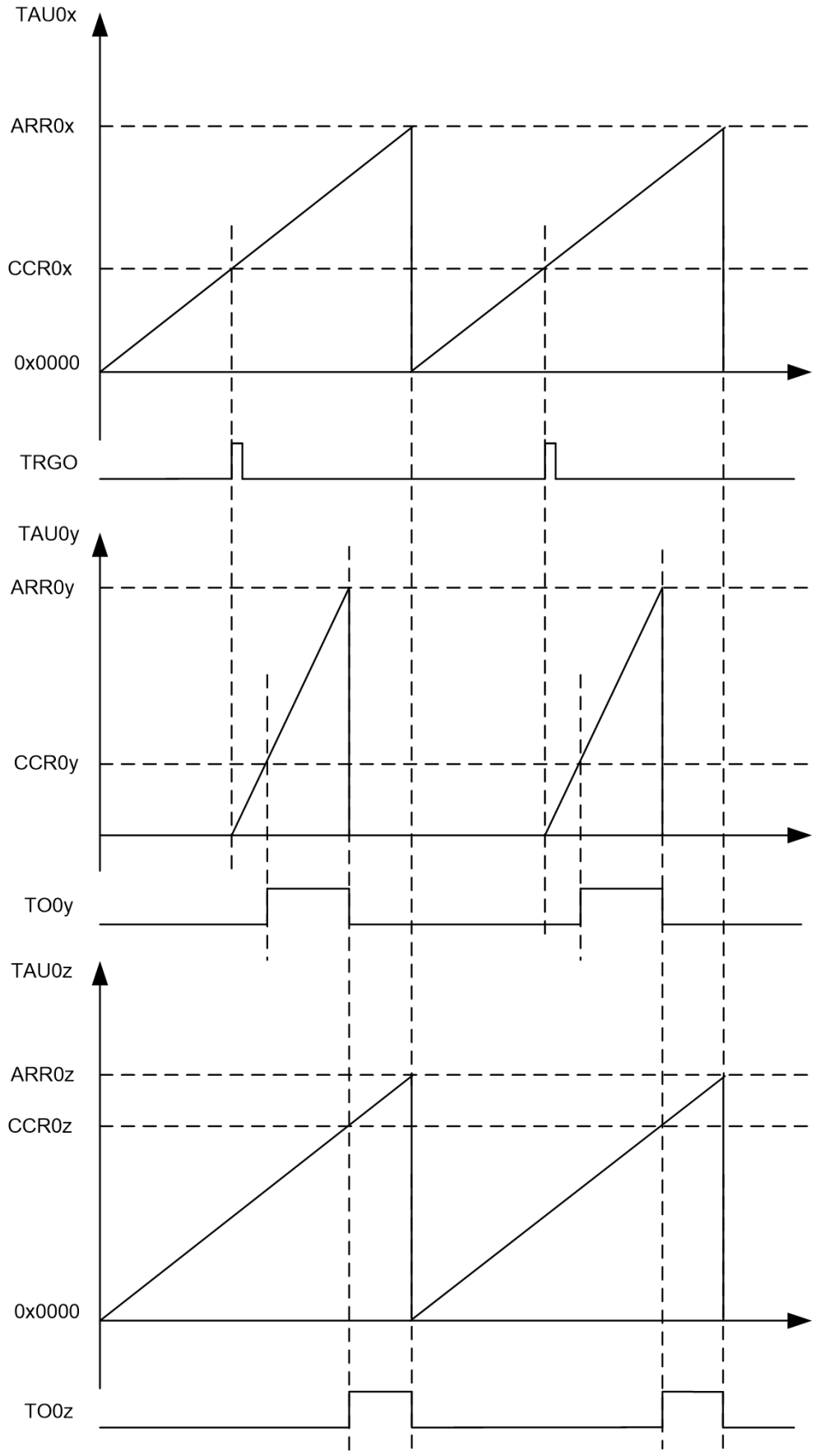


图 26-20 非对称非交叠 PWM 输出示意图

26.3.11 外部引脚输入数字滤波

扩展定时器的外部引脚输入TIxx可以选择是否使能数字滤波，以抑制信号噪声。使能数字滤波的情况下，使用APBCLK连续3次采样相同才认为是输入变化有效；不使能数字滤波时，扩展定时器直接对管脚输入进行采样。数字滤波示意图如下：

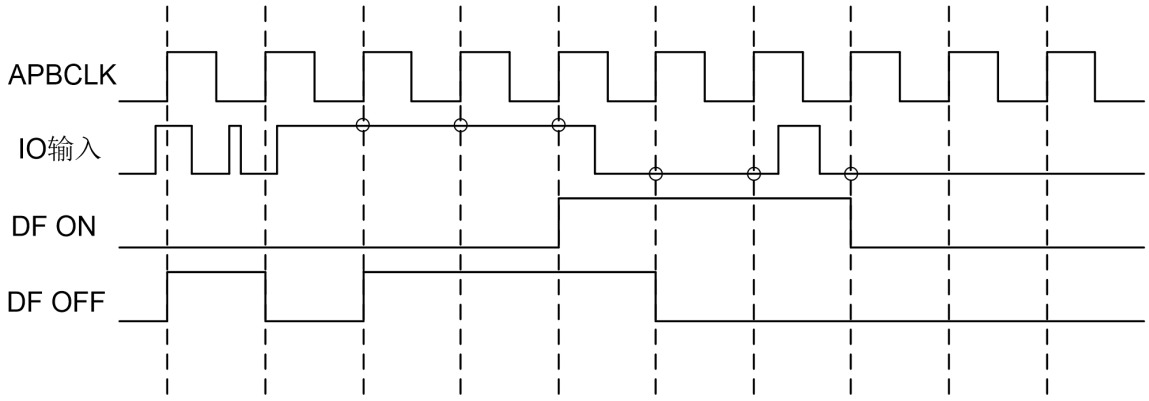


图 26-21 TIxx 输入数字滤波示意图

26.4 寄存器

TAU模块起始地址：0x40013C00

offset 地址	名称	符号
0x00	TAU0 控制寄存器	TAU0_CR
0x04	TAU1 控制寄存器	TAU1_CR
0x10	TAU0 通道 x 配置寄存器	TAU0_TxCFGR
0x14		
0x18		
0x1C		
0x20		
0x24		
0x28		
0x2C		
0x30	TAU0 通道 x 模式寄存器	TAU0_TxMDR
0x34		
0x38		
0x3C		
0x40		
0x44		
0x48		
0x4C		
0x50	TAU0 重载寄存器	TAU0_TxARR
0x54		
0x58		
0x5C		
0x60		
0x64		
0x68		
0x6C		
0x70	TAU0 捕捉比较寄存器	TAU0_TxCCR
0x74		
0x78		
0x7C		
0x80		
0x84		
0x88		
0x8C		
0x90	TAU0 中断使能寄存器	TAU0_TxIER
0x94		
0x98		
0x9C		
0xA0		
0xA4		

0xA8		
0xAC		
0xB0	TAU0 中断标志寄存器	TAU0_TxISR
0xB4		
0xB8		
0xBC		
0xC0		
0xC4		
0xC8		
0xCC		
0xD0	TAU0 中断标志寄存器	TAU0_TxCNT
0xD4		
0xD8		
0xDC		
0xE0		
0xE4		
0xE8		
0xEC		
0xF0	TAU1 通道 x 配置寄存器	TAU1_TxCFGR
0xF4		
0xF8		
0xFC		
0x100	TAU1 通道 x 模式寄存器	TAU1_TxMDR
0x104		
0x108		
0x10C		
0x110	TAU1 重载寄存器	TAU1_TxARR
0x114		
0x118		
0x11C		
0x120	TAU1 捕捉比较寄存器	TAU1_TxCCR
0x124		
0x128		
0x12C		
0x130	TAU1 中断使能寄存器	TAU1_TxIER
0x134		
0x138		
0x13C		
0x140	TAU1 中断标志寄存器	TAU1_TxISR
0x144		
0x148		
0x14C		
0x150	TAU1 中断标志寄存器	TAU1_TxCNT
0x154		
0x158		
0x15C		

26.4.1 TAU0 控制寄存器（TAU0_CR）

名称	TAU0_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EN[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	未实现：读为0
7:0	EN	定时器使能，每个 bit 单独控制一路定时器，1 使能 Bit0：Timer0 Bit1：Timer1 …… Bit7：Timer7 *注意，应在完成所有配置寄存器写入后，最后置位 EN 寄存器。EN 置位后，定时器立即开始工作。*

26.4.2 TAU0 通道 x 配置寄存器（TAU0_TxCFGR）

名称	TAU0_TxCFGR (x=0~7)							
offset	0x10~0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	CAPCLR	CAPONCE	NF	CAPEDGE		CNTEDGE	
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	TS		CAPSEL[2:0]			CNTSEL[1:0]	
位权限	R/W-0	R/W-00		R/W-000			R/W-00	

位号	位名	说明
31	--	未实现，读为0
30	CAPCLR	清零捕捉模式 0：自由计数捕捉模式 1：清零捕捉模式
29	CAPONCE	单次捕捉控制 1：单次捕捉有效，在捕捉到一次脉冲周期后计数器停止，若需要再次捕捉需重新启动 0：连续捕捉
28	NF	噪声滤波 0：关闭数字滤波 1：打开数字滤波
27:26	CAPEDGE	捕捉源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
25:24	CNTEDGE	计数源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
23:16	PRESCALE1	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	输出极性选择 0：输出不取反 1：输出取反
6:5	TS	从机触发源选择（Trigger Select） 仅在从机模式下（slave）有效，每个定时器详细的触发选择参见26.3.632位级联模式
4:2	CAPSEL	捕捉源选择寄存器 000：外部引脚输入（Tixx） 其他：参见26.3.7捕捉功能
1:0	CNTSEL	计数源选择寄存器 00/11：APBCLK 01：外部引脚输入（Tixx） 10：捕捉信号输入 （CAPSEL选择后的输出）

26.4.3 TAU0 通道 x 模式寄存器（TAU0_TxMDR）

名称	TAU0_TxMDR (x=0~7)							
offset	0x30~0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			SLVMD	TOEN	SLV	MD	
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-00	

位号	位名	说明
31:5	--	未实现，读为0
4	SLVMD	Slave模式选择，仅SLV=1的情况下有效 0：触发递增模式（trigger to increment），每个触发信号导致从机计数器加1 1：触发启动模式（trigger to start），每个触发信号导致从机从0开始计数，直到ARR停止
3	TOEN	TO输出使能 0：关闭TO输出 1：使能TO输出
2	SLV	Slave模式使能 0：主机模式 1：从机模式，可以接收其他Timer发送的TRGO *注意：Timer00无从机模式，此bit无效。*
1:0	MD	定时器工作模式 00：计数模式 01：比较输出模式（PWM） 10：输入捕捉模式（周期捕捉） 11：输入捕捉模式（脉宽捕捉）

26.4.4 TAU0 重载寄存器（TAU0_TxARR）

名称	TAU0_TxARR (x=0~7)							
地址	0x50~0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	ARR	自动重载寄存器 计数器计数值等于 ARR 时产生溢出信号并回到 0 开始重新计数。

26.4.5 TAU0 捕捉比较寄存器（TAU0_TxCCR）

名称	TAU0_TxCCR(x=0~7)							
地址	0x70~0x8C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	CCR	捕捉比较寄存器

26.4.6 TAU0 中断使能寄存器（TAU0_TxIER）

名称	TAU0_TxIER(x=0~7)							
地址	0x90~0xAC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---					CMPIE	CAPIE	OVIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:3	---	未实现：读为 0
2	CMPIE	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	定时器捕捉中断使能 1 = 使能 0 = 禁止

位号	位名	说明
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

26.4.7 TAU0 中断标志寄存器（TAU0_TxISR）

名称	TAU0_TxISR(x=0~7)							
地址	0xB0~0xCC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				EDGESTA	CMPIF	CAPIF	OVIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	---	未实现：读为 0
3	EDGESTA	捕捉沿状态标志，软件写 1 清 0 1 = 脉冲宽度捕捉模式时表示捕捉到下沿 0 = 脉冲宽度捕捉模式时表示捕捉到上沿
2	CMPIF	比较状态，软件写 1 清 0 1 =当前计数器的值大于等于比较寄存器的值 0 =当前计数器的值小于比较寄存器的值
1	CAPIF	定时器捕捉产生信号，软件写1清0 1 = 捕捉到指定的沿 0 = 未捕捉到指定的沿
0	OVIF	定时器溢出信号，当计数器的值达到ARR时将置位，软件写1清0 1 = 产生计数溢出 0 = 未产生溢出

26.4.8 TAU0 计数值寄存器（TAU0_TxCNTR）

名称	TAU0_TxCNTR (x=0~7)							
地址	0xD0~0xEC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	---	未实现：读为 0
15:0	CNT	计数器当前计数值，只读

26.4.9 TAU1 控制寄存器（TAU1_CR）

名称	TAU1_CR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				EN[3:0]			
位权限	U-0				R/W-0000			

位号	位名	说明
31:8	--	未实现：读为0
7:0	EN	定时器使能，每个 bit 单独控制一路定时器，1 使能 Bit0：Timer0 Bit1：Timer1 …… Bit3：Timer3 *注意，应在完成所有配置寄存器写入后，最后置位 EN 寄存器。EN 置位后，定时器立即开始工作。*

26.4.10 TAU1 通道 x 配置寄存器（TAU1_TxCFGR）

名称	TAU1_TxCFGR (x=0~3)							
offset	0xF0~0xFC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		CAPONCE	NF	CAPEDGE		CNTEDGE	

位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	TS		CAPSEL[2:0]			CNTSEL[1:0]	
位权限	R/W-0	R/W-00		R/W-000			R/W-00	

位号	位名	说明
31:30	--	未实现，读为0
29	CAPONCE	单次捕捉控制 1：单次捕捉有效，在捕捉到一次脉冲周期后计数器停止，若需要再次捕捉需重新启动 0：连续捕捉
28	NF	噪声滤波 0：关闭数字滤波 1：打开数字滤波
27:26	CAPEDGE	捕捉源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
25:24	CNTEDGE	计数源边沿选择 00/11：上升沿和下降沿 01：上升沿 10：下降沿
23:16	PRESCALE1	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	输出极性选择 0：输出不取反 1：输出取反
6:5	TS	从机触发源选择（Trigger Select） 仅在从机模式下（slave）有效，每个定时器详细的触发选择参见26.3.632位级联模式
4:2	CAPSEL	捕捉源选择寄存器 000：外部引脚输入（Tixx） 其他：参见26.3.7捕捉功能
1:0	CNTSEL	计数源选择寄存器 00/11：APBCLK 01：外部引脚输入（Tixx） 10：捕捉信号输入 （CAPSEL选择后的输出）

26.4.11 TAU1 通道 x 模式寄存器（TAU1_TxMDR）

名称	TAU1_TxMDR (x=0~3)
offset	0x100~0x10C

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TOEN	SLV	MD	
位权限	U-0				R/W-0	R/W-0	R/W-00	

位号	位名	说明
31:4	--	未实现，读为0
3	TOEN	TO输出使能 0：关闭TO输出 1：使能TO输出
2	SLV	Slave模式使能 0：主机模式 1：从机模式，可以接收其他Timer发送的TRGO *注意：TAU1无从机模式，此bit无效。*
1:0	MD	定时器工作模式 00：计数模式 01：比较输出模式（PWM） 10：输入捕捉模式 11：RFU

26.4.12 TAU1 重载寄存器（TAU1_TxARR）

名称	TAU1_TxARR (x=0~3)							
地址	0x110~0x11C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0

位号	位名	说明
15:0	ARR	自动重载寄存器 计数器计数值等于 ARR 时产生溢出信号并回到 0 开始重新计数。

26.4.13 TAU1 捕捉比较寄存器（TAU1_TxCCR）

名称	TAU1_TxCCR (x=0~3)							
地址	0x120~0x12C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR[15:8]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	未实现，读为0
15:0	CCR	捕捉比较寄存器

26.4.14 TAU1 中断使能寄存器（TAU1_TxIER）

名称	TAU1_TxIER (x=0~3)							
地址	0x130~0x13C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---					CMPIE	CAPIE	OVIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:3	---	未实现：读为 0
2	CMPIE	定时器比较中断使能 1 = 使能 0 = 禁止

位号	位名	说明
1	CAPIE	定时器捕捉中断使能 1 = 使能 0 = 禁止
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

26.4.15 TAU1 中断标志寄存器（TAU1_TxISR）

名称	TAU1_TxISR (x=0~3)							
地址	0x140~0x14C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---				EDGESTA	CMPIF	CAPIF	OVIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	---	未实现：读为 0
3	CMPIF	比较状态，软件写 1 清 0 1 =当前计数器的值大于等于比较寄存器的值 0 =当前计数器的值小于比较寄存器的值
2	EDGESTA	捕捉沿状态标志，软件写 1 清 0 1 = 脉冲宽度捕捉模式时表示捕捉到下沿 0 = 脉冲宽度捕捉模式时表示捕捉到上沿
1	CAPIF	定时器捕捉产生信号，软件写1清0 1 = 捕捉到指定的沿 0 = 未捕捉到指定的沿
0	OVIF	定时器溢出信号，当计数器的值达到ARR时将置位，软件写1清0 1 = 产生计数溢出 0 = 未产生溢出

26.4.16 TAU1 计数值寄存器（TAU1_TxCNTR）

名称	TAU1_TxCNTR (x=0~3)							
地址	0x150~0x15C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	---	未实现：读为 0
15:0	CNT	计数器当前计数值，只读

27 16 位基本定时器（BSTIM16）

27.1 概述

FM33FR0包含1个16位基本定时器。

基本定时器包含一个16bit自动重载计数器及一个可编程预分频器。

基本定时器主要用来产生系统时基，也可以产生触发事件来驱动ADC采样。

27.2 主要特性

- 16bit向上计数自动重载计数器
- 16bit可编程预分频器，支持实时调整计数时钟分频
- ADC定时触发功能
- 计数器溢出时产生中断

27.3 结构框图

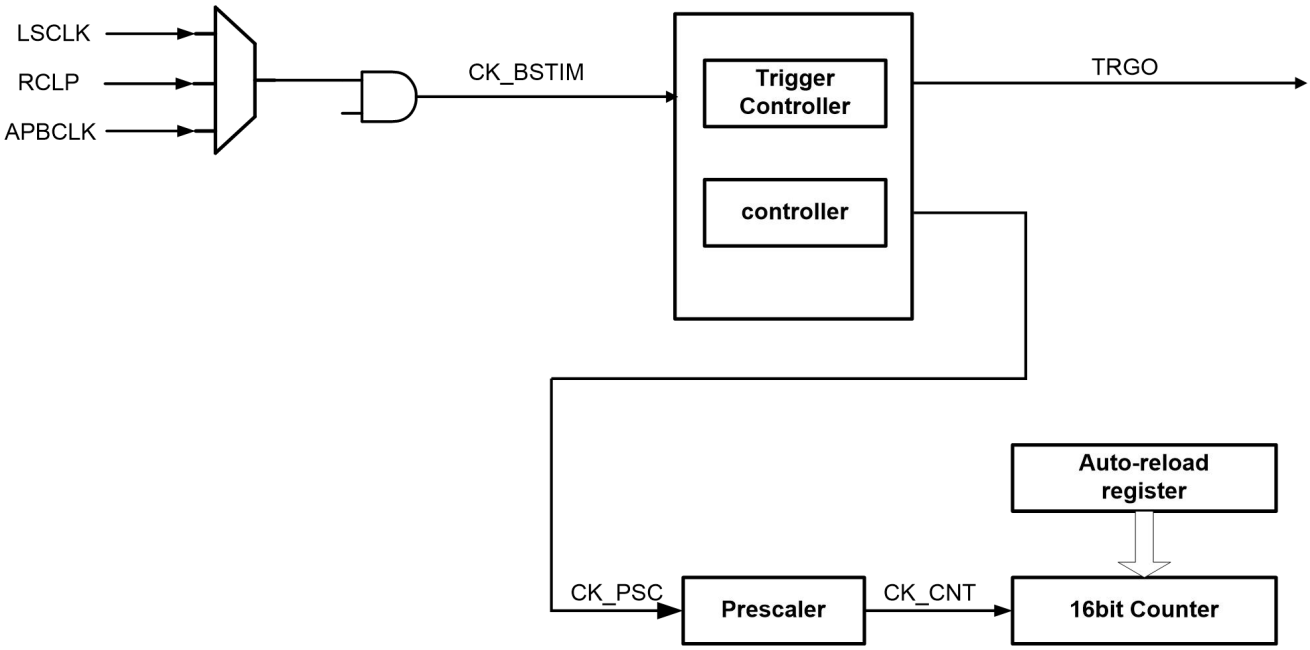


图 27-1 16 位基本定时器结构框图

27.4 功能描述

27.4.1 定时单元

基本定时器的定时单元由一个16位计数器和自动重载寄存器组成。计数器向上计数。计数时钟可以通过16位预分频器对APBCLK进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（BSTIM_CNT）
- 预分频寄存器（BSTIM_PSC）
- 自动重载寄存器（BSTIM_ARR）

ARR包含preload功能，软件读写ARR可以直接起效，或者只是访问其缓存，通过ARPE（Auto Reload Preload Enable）寄存器控制。当ARPE=1时，软件读写ARR都是访问其缓存寄存器，当update event（ATIM_CNT上溢出或者下溢出）发生时，会将缓存寄存器内的数据更新到ARR中。软件也可以通过寄存器操作主动触发ARR更新。

BSTIM_CNT工作时钟由BSTIM_PSC产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT才开始计数。当CNT=ARR时，本轮计数结束，发送update event。

BSTIM_PSC是一个同步预分频器，能够对APBCLK进行1~65536分频。PSC寄存器同样被缓存，改写PSC实际是改写缓存寄存器，只有当新的update event到来时，才会从缓存寄存器更新PSC。因此在CNT计数过程中，软件可以实时改写PSC。

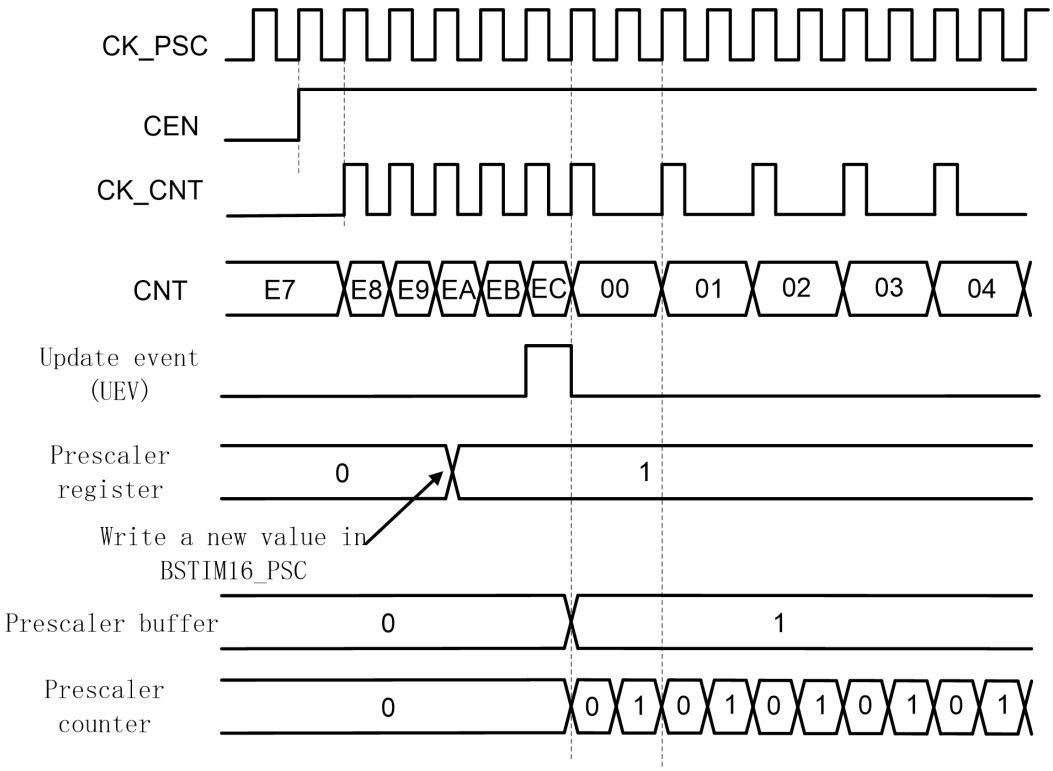


图 27-2 预分频从 1 变为 2 的波形

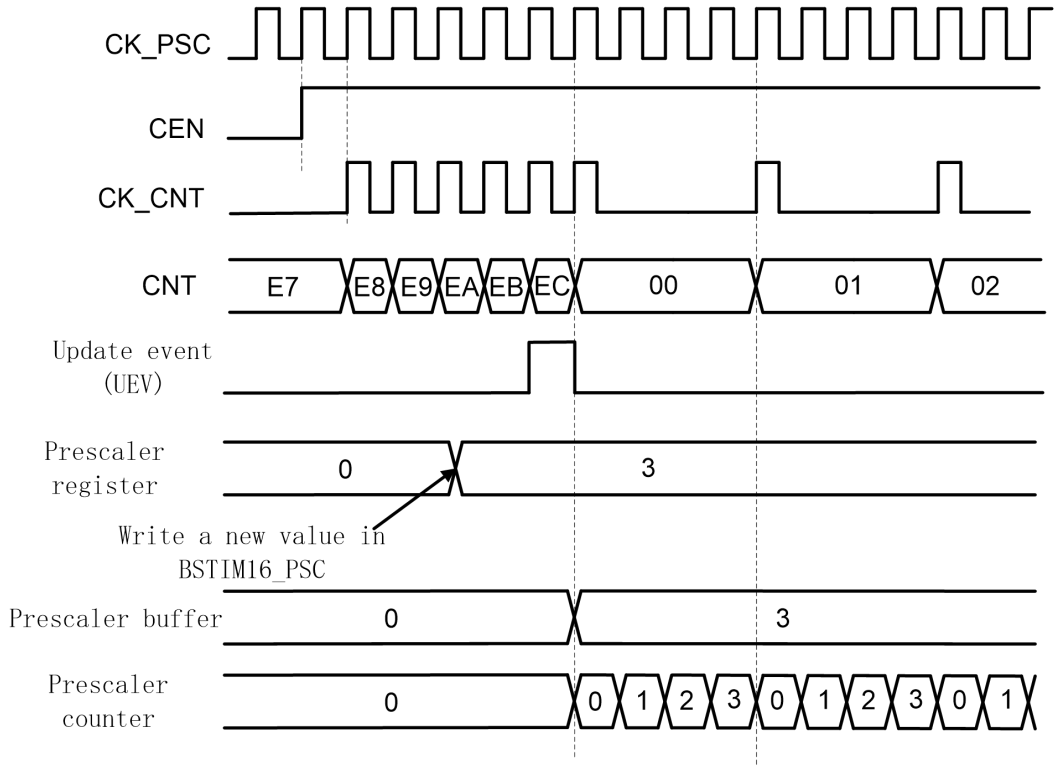


图 27-3 预分频从 1 变为 4 的波形

27.4.2 定时器工作模式

通用定时器只支持向上计数模式。

向上计数

此模式中，计数器使能后从0开始计数，直到CNT=ARR，产生溢出事件，然后重新从0开始计数。

软件可以通过设置UG寄存器直接触发update event，此时CNT自动清零。

通过设置UDIS寄存器可以禁止update event，这样可以避免将preload寄存器中的值更新到工作寄存器中。

当update event发生时，以下寄存器被更新，并且UIF置位：

- BSTIM_ARR更新为缓存中的值
- BSTIM_PSC更新为缓存中的值

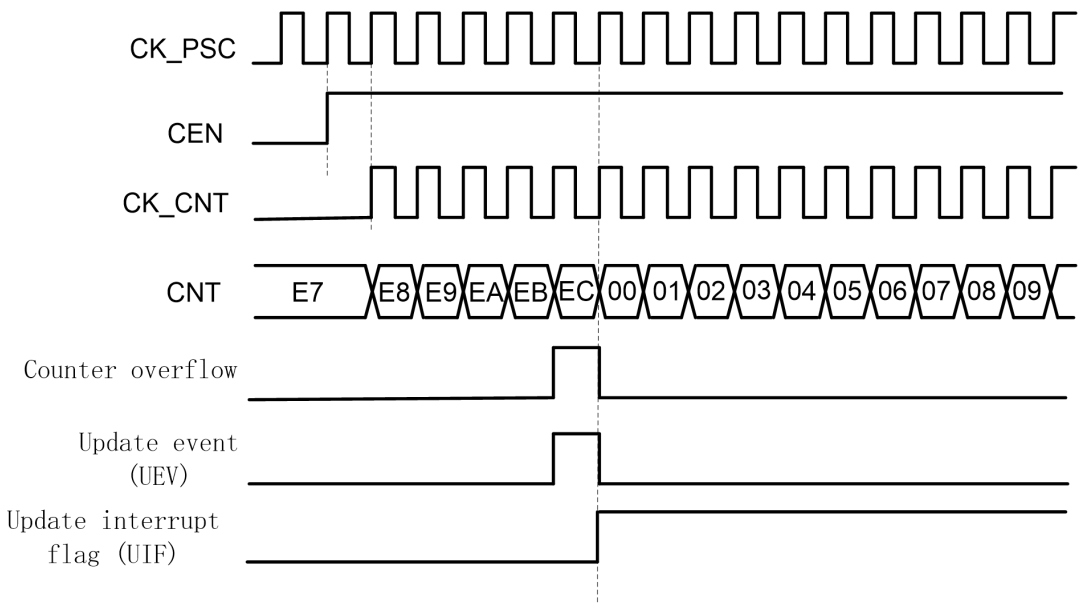


图 27-4 向上计数波形，内部时钟不分频

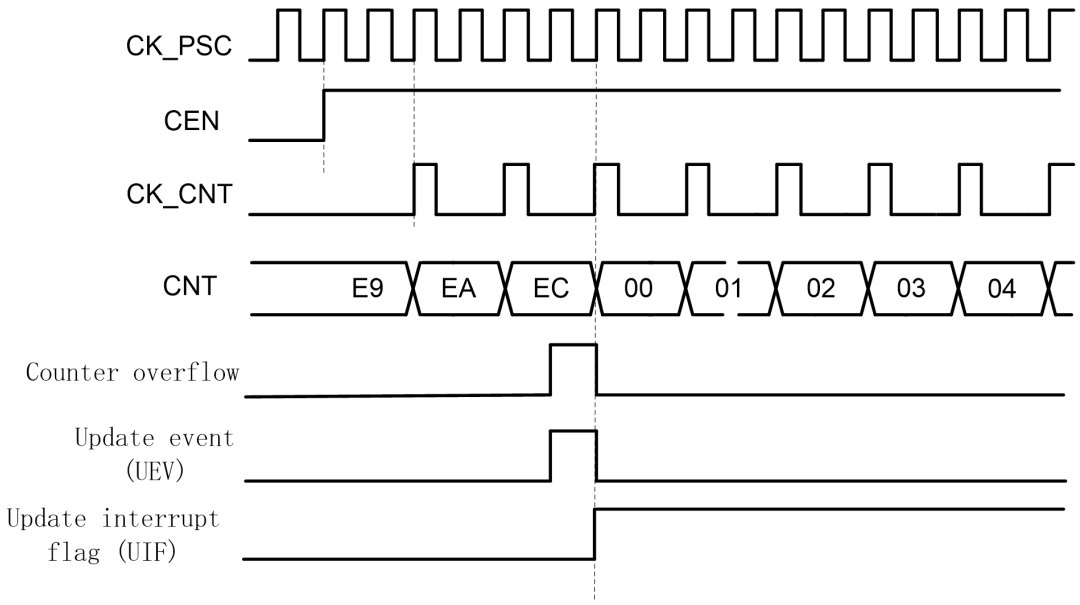


图 27-5 向上计数波形，内部时钟 2 分频

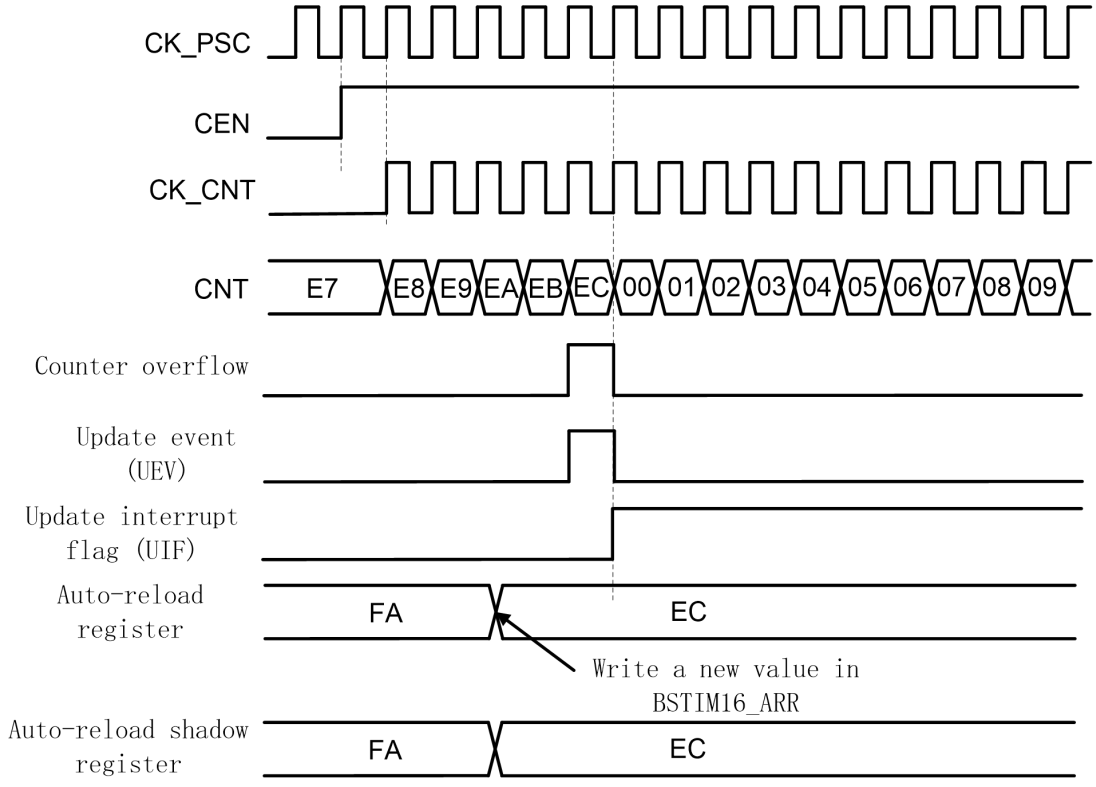


图 27-6 ARPE=0（ARR 没有预装载）时的更新事件

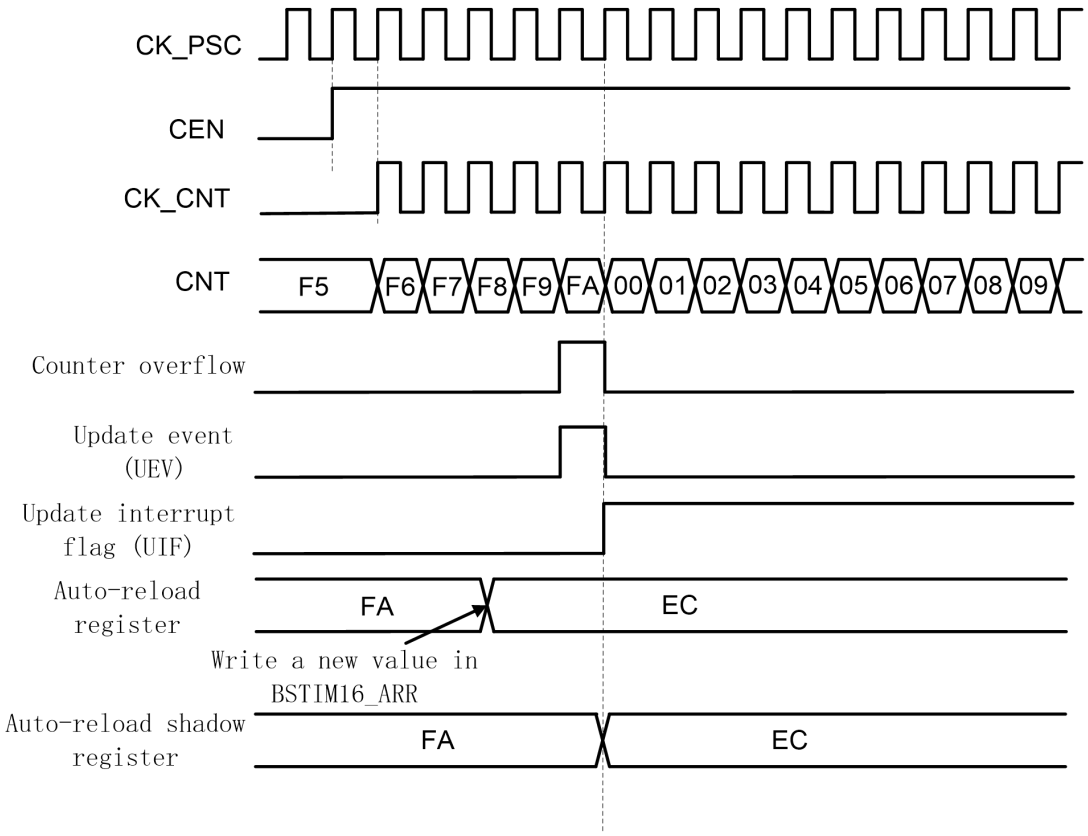


图 27-7 ARPE=1（ARR 预装载）时的更新事件

27.4.3 计数器工作时钟

BSTIM使用内部时钟工作，CEN、UG等寄存器位都是软件控制

软件操作UG寄存器后，update信号经过CLK_PSC同步后，计数器值将被重新初始化。

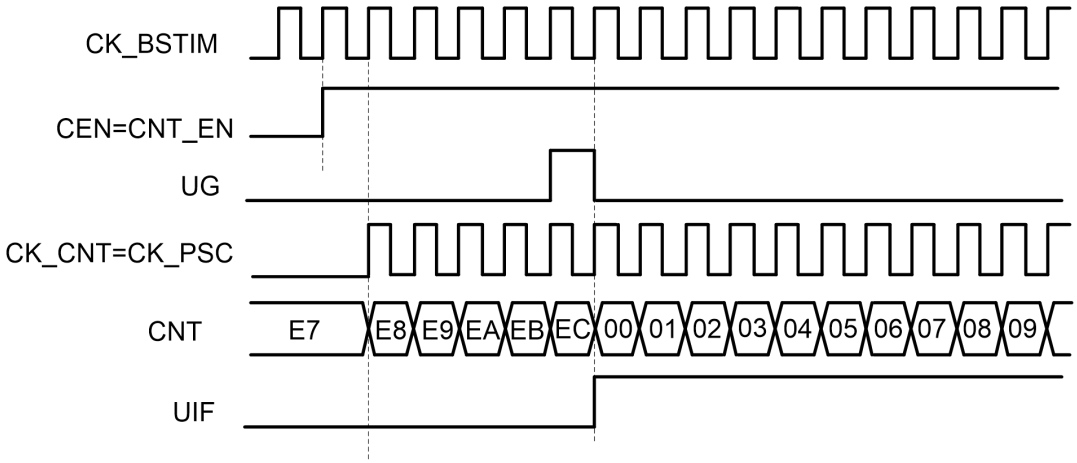


图 27-8 内部时钟源模式，时钟分频因子为 1

27.4.1 Debug 模式

当Cortex-M0进入debug模式后，定时器可以停止或继续工作，其行为由DBG模块的DBG_BT16_STOP寄存器定义。

27.5 寄存器

BSTIM16模块起始地址：0x4001_B400

offset 地址	名称	符号
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM16_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM16_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM16_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM16_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM16_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM16_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM16_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM16_ARR

27.5.1 BSTIM 控制寄存器 1(BSTIM16_CR1)

名称	BSTIM16_CR1							
地址	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	-			OPM	URS	UDIS	CEN
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:8	-	RFU，未实现，读为 0
7	ARPE	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:4	-	RFU，未实现，读为 0
3	OPM	单脉冲输出模式 0：Update Event 发生时计数器不停止

Bit	助记符	功能描述
		1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	更新请求选择 0：以下事件能够产生 update 中断 - 计数器溢出 - 软件置位 UG 寄存器 1：仅计数器溢出出会产生 update 中断或 DMA 请求
1	UDIS	禁止 update 0：使能 update 事件；以下事件发生时产生 update 事件 - 计数器溢出 - 软件置位 UG 寄存器 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位时重新初始化计数器和预分频器。
0	CEN	计数器使能 0：计数器关闭 1：计数器使能

27.5.2 BSTIM 控制寄存器 2(BSTIM16_CR2)

名称	BSTIM16_CR2							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

Bit	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 CEN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO Others：RFU 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3:0	-	RFU，未实现，读为 0

27.5.3 BSTIM 中断使能寄存器(BSTIM16_IER)

名称	BSTIM16_IER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIE
位权限	U-0							RW-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	UIE	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

27.5.4 BSTIM 状态寄存器(BSTIM16_ISR)

名称	BSTIM16_ISR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UIF
位权限	U-0							RW-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	UIF	Update 事件中断标志，硬件置位，软件写 1 清零。 当以下事件发生时，UIF 置位，并更新 shadow 寄存器 -重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出 -URS=0 且 UDIS=0 的情况下，置位 UG 寄存器

27.5.5 BSTIM 事件产生寄存器(BSTIM16_EGR)

名称	BSTIM16_EGR							
地址	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							UG
位权限	U-0							W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	UG	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器。 *注意：由于 BSTIM 计数器工作时钟与系统总线时钟 APBCLK 是互相独立的，当软件置位 UG 寄存器后，UIF 并不会立即置位，而是需要经过 CK_BSTIM 同步后才置位，两者之间的延迟与 APBCLK 和 CK_BSTIM 之间的频率及相位关系有关，并非确定值。*

27.5.6 BSTIM 计数器寄存器(BSTIM16_CNT)

名称	BSTIM16_CNT							
地址	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	CNT	计数器值

27.5.7 BSTIMx 预分频寄存器(BSTIM16_PSC)

名称	BSTIM16_PSC							
地址	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**PSC**	计数器时钟（CK_CNT）预分频值 $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

27.5.8 BSTIMx 自动重载（auto-reload）寄存器(BSTIM16_ARR)

名称	BSTIM16_ARR							
地址	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	RW-11111111							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**ARR**	计数溢出时的自动重载值 这是一个 preload 寄存器，在 update 事件发生时其内容被载入 shadow 寄存器

28 16 位低功耗定时器（LPTIM16）

28.1 概述

LPTIM16是16bits低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM16在在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM16甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。LPTIM16的主要特性有：

- 1 个独立的 16bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK、RCLP、APBCLK
  - 外部时钟源：LPT16_ETR（带有模拟滤波）
- 2 通道 16bit 捕捉/比较寄存器
- 16bit 自动重载寄存器
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 16bit PWM 输出
- 16bit 输入信号捕捉
- 触发信号输出
- 双通道正交编码器

28.2 结构框图

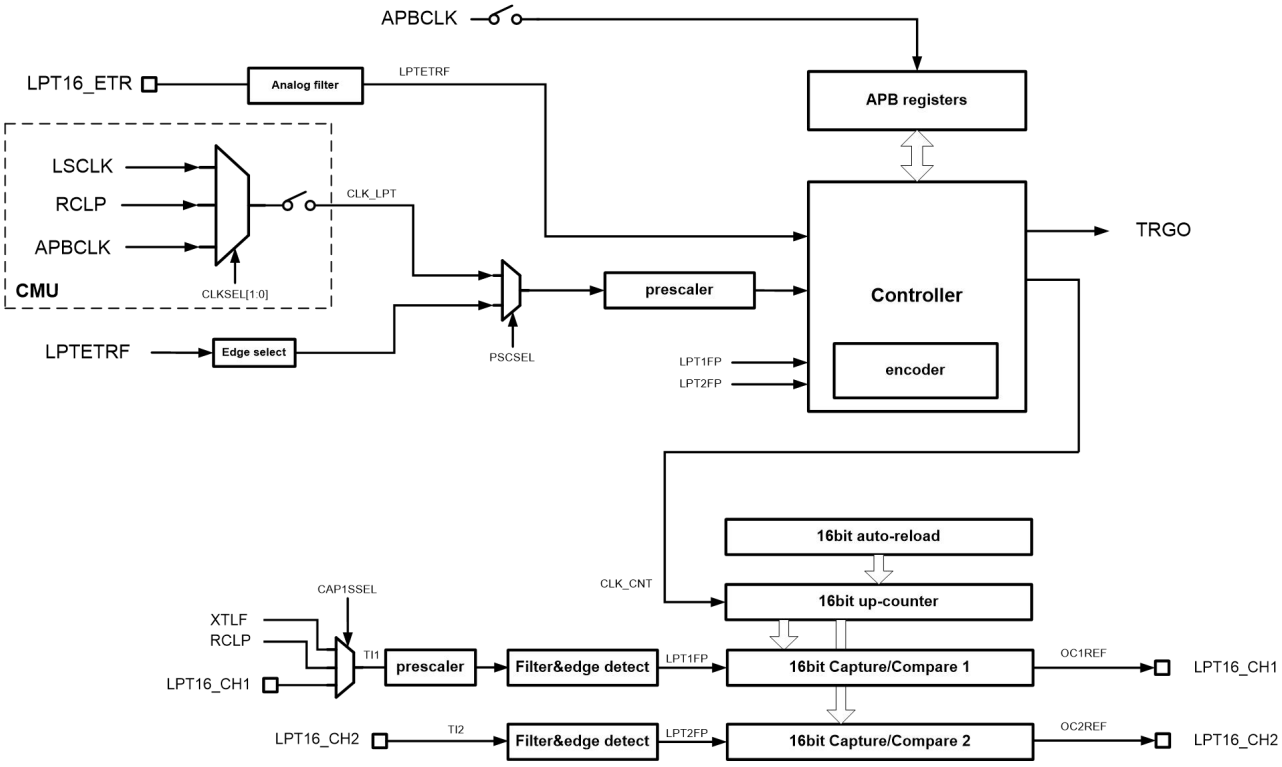


图 28-1 LPTIM16 结构框图

28.3 时钟和复位

LPTIM16的控制和状态寄存器都位于APB总线，因此软件访问寄存器前必须先使能外设总线时钟，详情参见13时钟管理单元（CMU）。

LPTIM16定时器的工作时钟独立于系统总线时钟，可以从多个独立时钟源中选择。通过配置CMU寄存器可以选择LPTIM16的计数时钟源。为了保证定时器工作稳定可靠，禁止在EN为1的情况下修改计数时钟。

在选择了合适的计数时钟之后，还可以通过DIVSEL寄存器对其进行预分频，以获得更低的工作时钟频率。同样，必须在EN为0的情况下修改DIVSEL。

LPTIM16模块可以通过操作LPT16RST寄存器来复位和撤销复位，详情参见RMU章节。

28.4 定时器功能

LPTIM16支持4种定时器工作模式：普通定时器、外部脉冲触发计数、外部异步脉冲计数、Timeout模式。

28.4.1 普通定时器

当LPTCFG.TMODE=00时，LPTIM16为普通定时器工作模式

- 使用多路选择后的CLK_LPT时钟计数
- 需要配置CMU模块中的OPCR1.LPT16CKS寄存器，选择合适的计数时钟
- LPTIM16_CR.EN使能置位后有两个计数时钟的同步过程
- 使能后定时器即开始向上计数，直到计数值等于ARR

单次计数和连续计数

LPTIM16有两种计数模式——单次计数和连续计数。

连续计数模式：计数器启动后保持运行，直到被关闭为止。计数器达到目标值ARR后回到0重新开始计数，并产生溢出中断。

单次计数模式：计数器被触发后计数到目标值ARR后回到0，并自动停止，产生溢出中断，同时硬件自动清除EN。

28.4.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTIM16_CFGR.TMODE=01）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用。LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTIM16_CFGR.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图举例说明了LPT16_ETR触发计数，上升沿有效的情况。

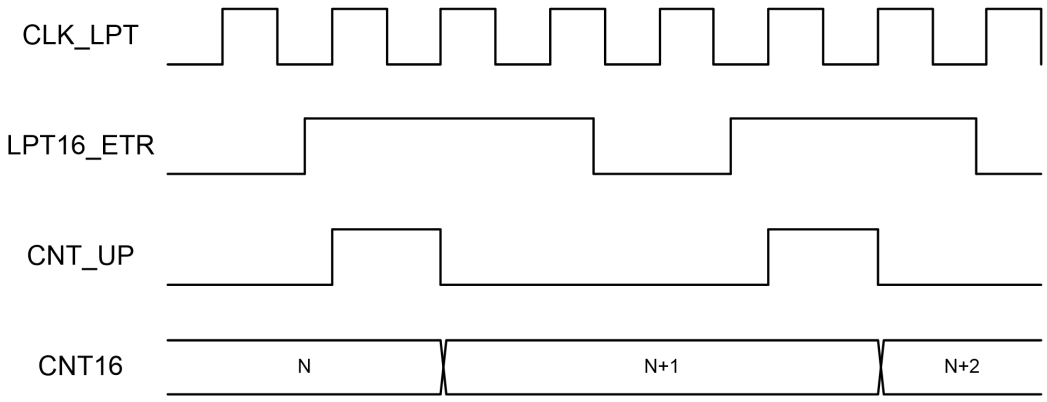


图 28-2 外部 ETR 脉冲上升沿触发计数

28.4.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTIM16_CFGR.TMODE=10）下，LPTIM16将LPT16_ETR引脚输入的信号直接作为计数时钟使用。这种情况下，LPTIM16全异步工作，不需要使能任何内部时钟。软件可以通过LPTIM16_CFGR.EDGESEL来选择定时器使用ETR上升沿还是下降沿计数。由于这种模式下LPT16_ETR引脚上的任何干扰信号都有可能引起定时器误动作，因此推荐使能ETR输入模拟滤波功能，能够滤除大约100ns以内的glitch信号。

下图举例说明了外部异步脉冲计数，下降沿有效的情况。

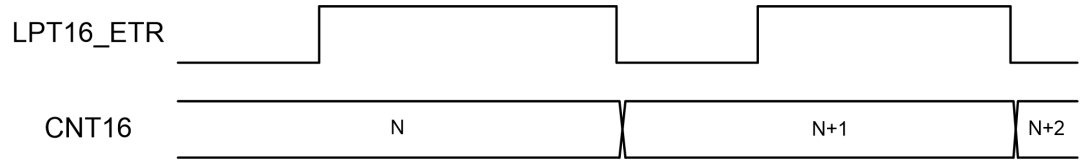


图 28-3 外部 ETR 脉冲异步计数（下降沿）

28.4.4 Timeout 模式

Timeout模式（LPTIM16_CFGR.TMODE=11）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用，定时器使用内部时钟CLK_LPT工作。Timeout模式下定时器启动后，不会立即开始计数，而是等待第一个LPT16_ETR信号的有效沿到来。当第一个有效沿到来后，触发定时器开始自由计数，此后每个新的ETR有效沿都会清零计数器，并重新开始计数。根据外部输入ETR信号的实际频率，合理配置计数器工作时钟和溢出上限ARR，可以保持定时器不会溢出。如果定时器出现溢出，则表示在规定时间间隔内没有预期的ETR事件到来，则定时器产生溢出中断，计数值回到0，并自动清除EN结束计数过程。

LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发计数器清零重新计数。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTIM16_CFGR.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图是timeout模式下使用LPT16_ETR上升沿清零，并最终溢出的例子。

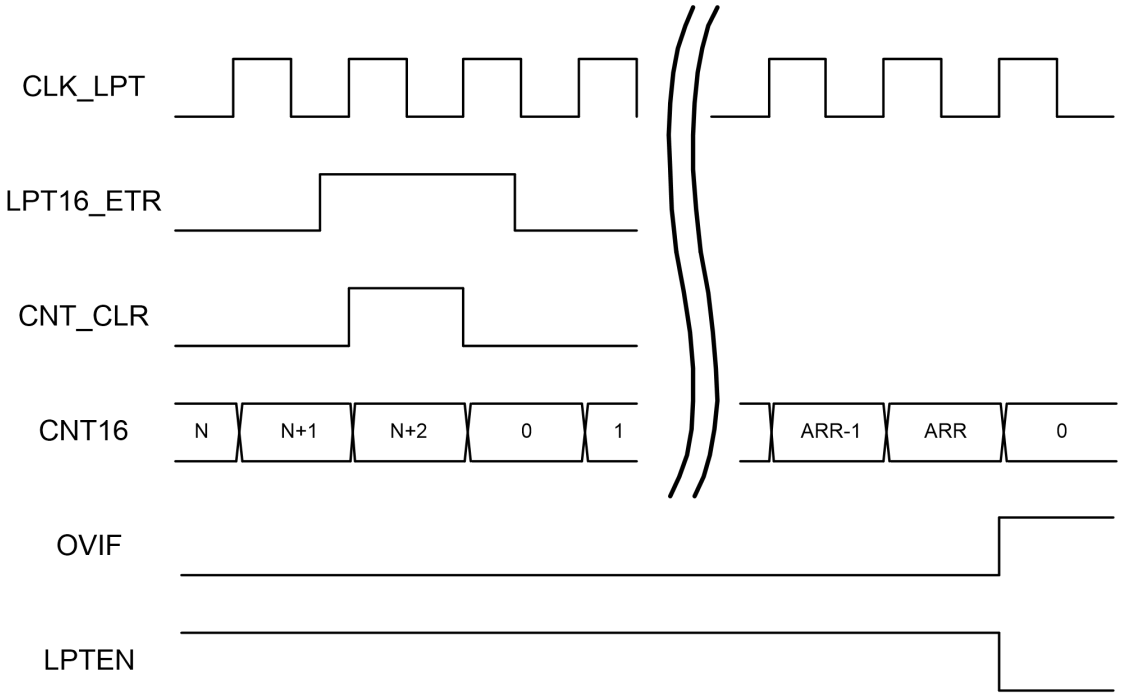


图 28-4TimeOut 模式

使用TimeOut模式，并使能LPTIM中断，在芯片休眠时可以实现外部信号触发的超时唤醒功能。此时只要LPT16_ETR管脚上有周期性信号输入，就能使芯片保持休眠，而一旦超过规定时间内没有新的触发信号到来，LPTIM超时溢出中断将唤醒芯片。

28.5 捕捉比较功能

LPTIM16带有两个独立的16bit捕捉比较通道，以16bit定时器为时基，结合CCRx寄存器，可以实现两路16bit PWM输出，或16bit输入捕捉功能。

28.5.1 16bit PWM

LPTIM16的两个独立捕捉/比较通道都可以输出16bit PWM波形。PWM功能需要将捕捉/比较通道配置为比较输出。

使能PWM功能后LPTIM16从0开始计数，计数值等于比较值CCRx正极性时输出高电平，计数值等于目标值寄存器ARR时正极性时输出低电平；PWM周期由ARR寄存器决定，占空比由CCRx寄存器决定。LPTIM16_CCSR.CCxP寄存器可以配置输出波形的极性。

实现PWM输出功能，需要将LPTIM16_CCSR.CCxS配置为10，此时LPT_CHx成为输出通道，相应的GPIO自动使能输出功能（软件需将GPIO配置为数字外设功能）。

下图是PWM输出，CCxP=1的例子。

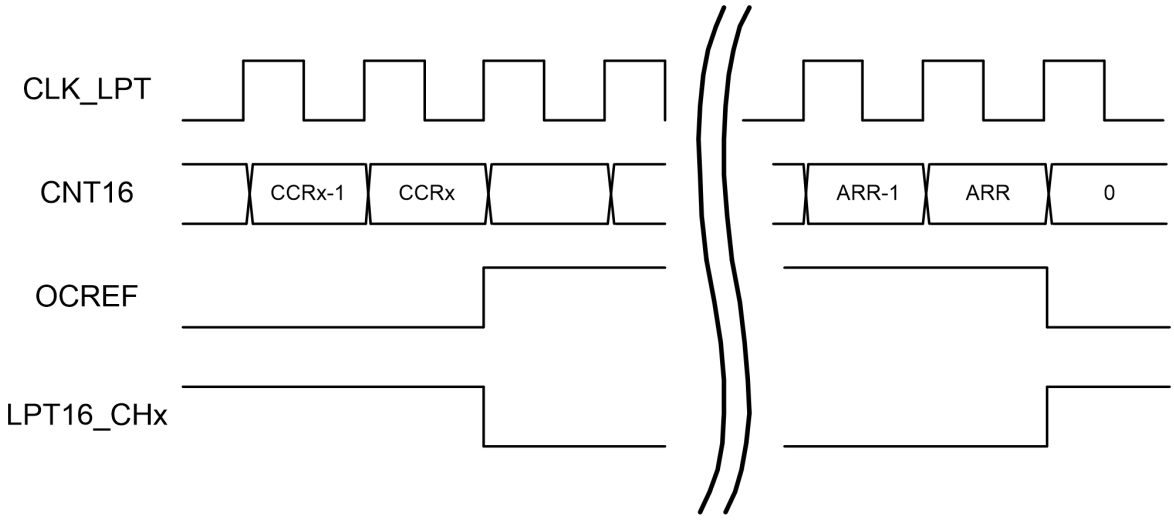


图 28-5PWM 输出

28.5.2 输入捕捉

LPTIM16的两个捕捉/比较通道可以实现两路独立的输入信号周期或电平宽度捕捉功能。

输入捕捉可以配置为针对输入信号的上升沿、下降沿或上升下降沿进行捕捉。每次捕捉发生时，CAPxEDGE寄存器会指示当前捕捉到的是上升沿还是下降沿。

LPTIM16的通道1可以对外部引脚输入或者芯片内部时钟信号（XTLF、RCLP）进行捕捉，对内部时钟信号的周期捕捉可以用于软件配合的时钟频率校准；而通道2只能对外部引脚输入信号进行捕捉。

通道1内置预分频器，经过对输入信号分频，可以对高频时钟进行分频后的周期捕捉和校准。

使能输入模式后，16bit计数器作为时基自由计数，当被捕捉信号的有效边沿到来后，当前计数值被锁存入CCRx寄存器，并产生捕捉中断；软件读取CCRx寄存器时，硬件都会自动清除捕捉中断，此外捕捉中断标志也可以由软件写1清零。当捕捉中断未被清除时，又有新的捕捉事件到来，会置位捕捉冲突中断标志（CAPxOVR）。

下图是对输入信号上升下降沿进行捕捉的例子。

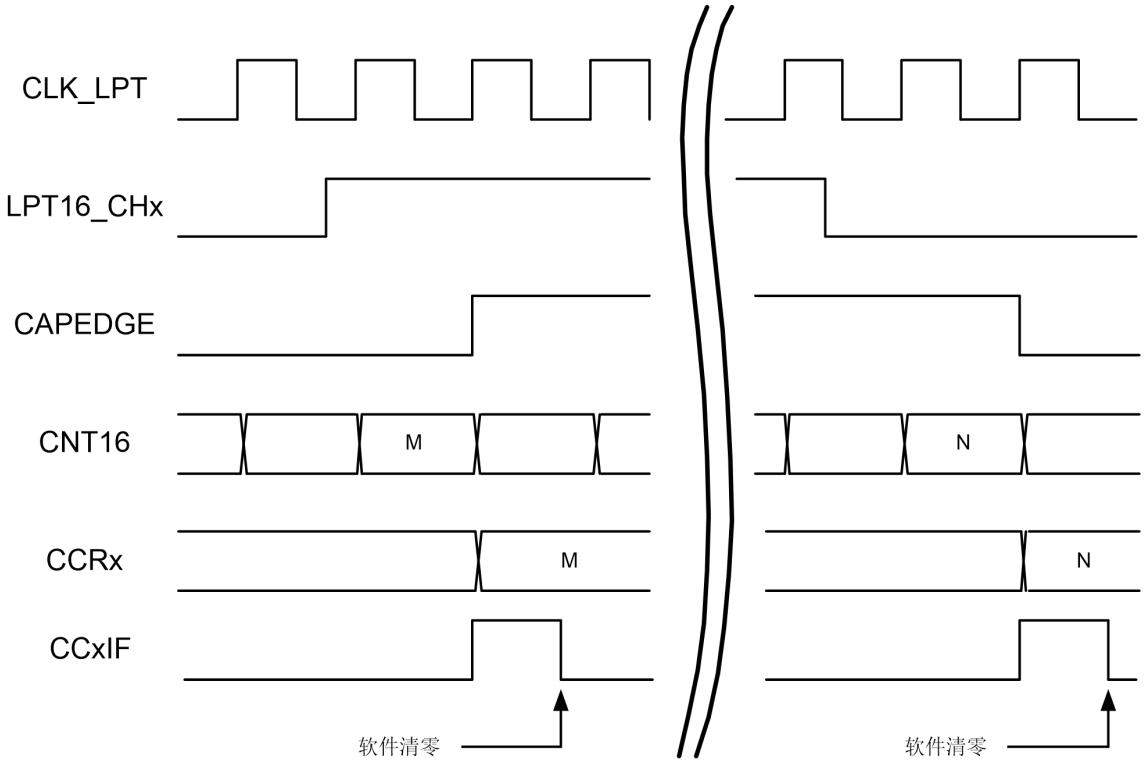


图 28-6 输入信号边沿捕捉

28.5.1 输入数字滤波

两个输入通道可以独立使能或关闭输入数字滤波功能。使能数字滤波的情况下，使用CLK_LPT计数时钟连续采样输入信号三次，相同电平认为有效。

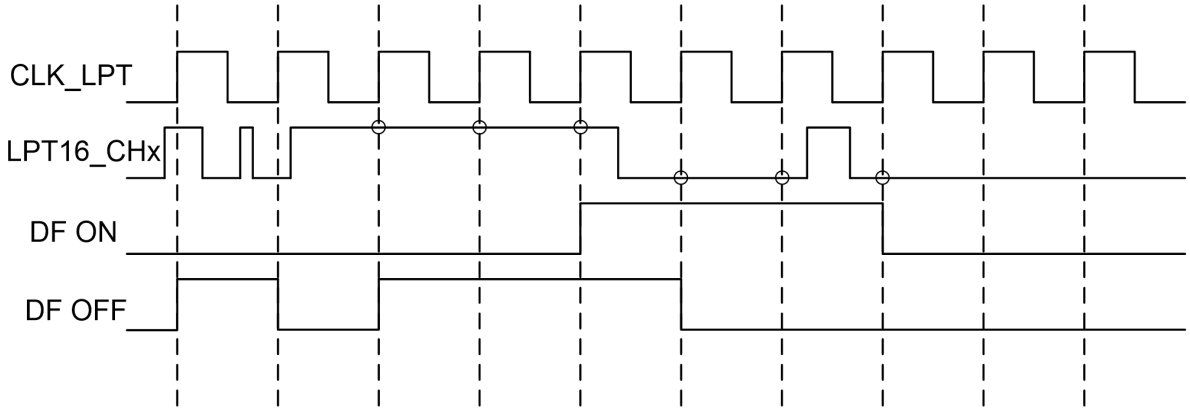


图 28-7 通道输入数字滤波

28.6 正交编码器

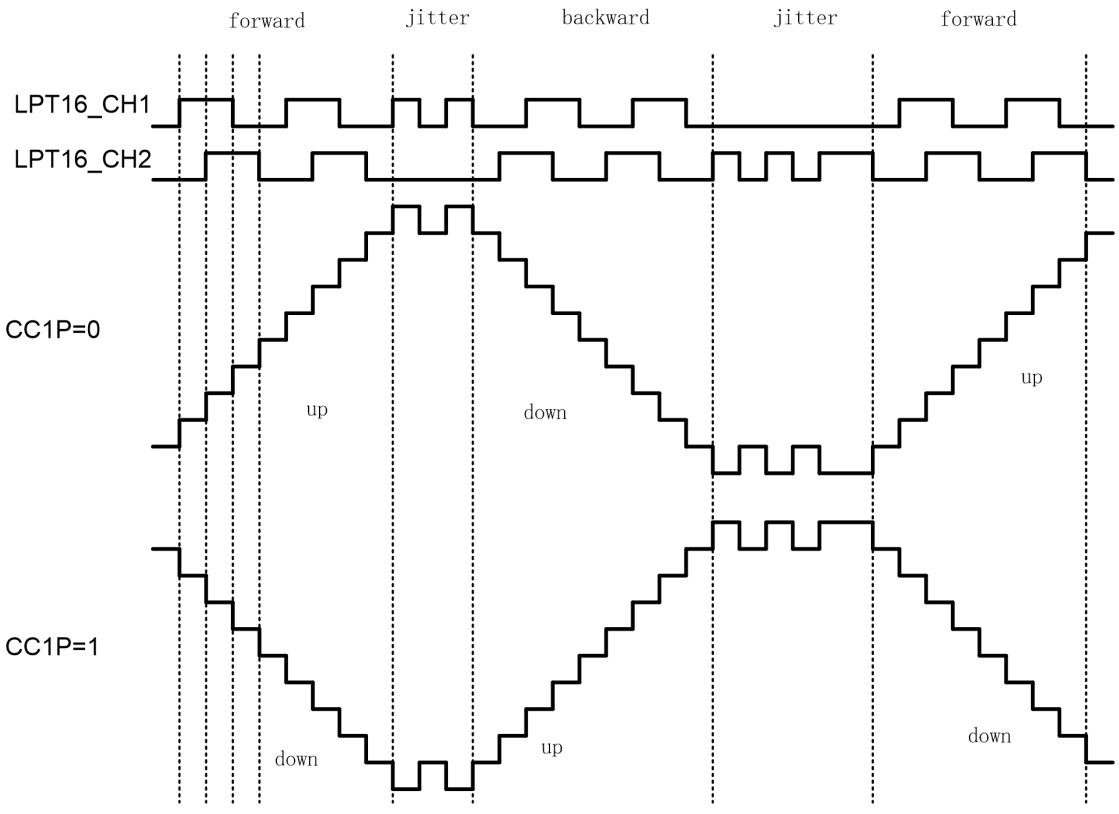
编码器接口模式涉及到两个外部输入信号，LPTIM16根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

有效沿	对应信号的电平（LPT1FP对应LPT2FP，LPT2FP对应LPT1FP）	LPT1FP信号		LPT2FP信号	
		上升	下降	上升	下降
仅在LPT1FP处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在LPT2FP处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在LPT1FP和LPT2FP处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

表 28-1 encoder interface 计数方式

比如在计数器以LPT1FP信号为时钟计数时，如果LPT1FP上升沿采样到LPT2FP为高电平，则计数器递减；如果LPT1FP下降沿采样到LPT2FP为高电平，则计数器递增。（同步计数，CLK_LPT采样取沿后进行计数，参考ATIM设计）

下图为编码器在LPT1FP和LPT2FP边沿同时计数的例子。由于CC1P能够控制输入信号LPT16_CH1的极性，所以当CC1P为1时输入取反，相应的计数器增减方向也会反向。



Example of counter operation in encoder interface mode

图 28-8 编码器模式下的计数器操作实例

正交编码信号使用LPT16_CHx输入，可以选择对输入信号是否进行数字滤波。注意LPT16_CH1通道上还有预分频电路，在进行正交编码计数时应关闭预分频。

QDIR寄存器用于表示当前输入信号相位关系，当CH1相位领先于CH2时，QDIR=0，反之QDIR=1，此寄存器只读。

当CH1和CH2发生相位翻转时，QDCIF中断标志置位，表示出现换向，如果QDCIE寄存器置位，则产生中断事件。

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为LPT16_CH1，LPT16_CH2功能
- 选择输入通道，配置LPTIM16_CCSR.CC1S=01，LPTIM16_CCSR.CC2S=01
- 选择计数有效沿（输入信号极性），配置LPTIM16_CCSR.CC1P和CC2P
- 设定编码器模式，配置LPTIM16_CFGR.QEMD寄存器
- 使能LPTIM16，置位LPTIM_CR.EN

28.7 寄存器

模块基地址：0x4001_3400

offset 地址	名称	符号
0x00	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM16_CFGR
0x04	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM16_CNT
0x08	LPTIM 捕捉比较控制和状态寄存器 （ LPTIM Capture/Compare Control and Status Register）	LPTIM16_CCSR
0x0C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM16_ARR
0x10	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM16_IER
0x14	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM16_ISR
0x18	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM16_CR
0x20	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM16_CCR1
0x24	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM16_CCR2

28.7.1 LPTIM 配置寄存器（LPTIM16_CFGR）

名称	LPTIM16_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					MMS		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	PSCSEL	-	DIVSEL			-	
位权限	U-0	R/W-0	U-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		QEMD		ONST	TMODE	
位权限	R/W-0	R/W-00		R/W-00		R/W-0	R/W-00	

位号	位名	说明
31:25	--	**未实现：读为0**
24	**ETR_AFEN**	LPT16_ETR 输入模拟滤波使能(External Trigger input Analog Filter Enable) 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns

位号	位名	说明
23:19	--	未实现：读为0
18:16	**MMS**	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：RFU 001：计数器使能信号 EN 被用作 TRGO 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：RFU 111：RFU 注意：从机必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
15	--	未实现：读为0
14	**PSCSEL**	时钟预分频输入选择(Prescaler input Select) 0：CLKSEL 选择的时钟 1：LPTETRF
13	--	未实现：读为0
12:10	**DIVSEL**	计数时钟分频选择(Counter Clock Divider Select) 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
9:8	--	未实现：读为0
7	**EDGESEL**	ETR 输入边沿选择(ETR Clock Edge Select) 0：LPT_ETR 的上升沿计数 1：LPT_ETR 的下降沿计数
6:5	**TRIGCFG**	外部触发边沿选择（需使用内部时钟同步采样 LPT_ETR）(ETR trigger Configuration) 00: LPT_ETR 输入信号上升沿触发 01: LPT_ETR 输入信号下降沿触发 10/11: LPT_ETR 输入信号上升下降沿触发
4:3	**QEMD**	正交编码器模式 (Quad encoder mode)，仅TMODE=00时有效 00：关闭正交编码器 01：Encoder 模式 1；计数器使用 LPT2FP 边沿，根据 LPT1FP 电平高低来计数 10：Encoder 模式 2；计数器使用 LPT1FP 边沿，根据 LPT2FP 电平高低来计数 11：Encoder 模式 3；计数器同时使用 LPT1FP 和 LPT2FP 边沿，根据其他输入信号电平来计数

位号	位名	说明
2	**ONST**	单次计数模式使能(One State Timer) 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。
1:0	**TMODE**	工作模式选择(Timer operation Mode) 00：普通定时器模式 01：Trigger 脉冲触发计数模式 10：外部异步脉冲计数模式 11：Timeout 模式

28.7.2 LPTIM 计数值寄存器（LPTIM16_CNT）

名称	LPTIM16_CNT							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	**--**	未实现：读为0
15:0	**CNT**	16bit 计数器当前计数值(Counter 16bits-wide)

28.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR）

名称	LPTIM16_CCSR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CAP1PSC						CAP1SSEL	
位权限	R/W-00 0000						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		CAP2EDGE	CAP1EDGE	-	-	CC2P	CC1P
位权限	U-0		R-0	R-0	U-0	U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC2DF	CC1DF	CAPCFG2		CAPCFG1	
位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	CC2S	CC1S
位权限	U-0	R/W-00	R/W-00

位号	位名	说明
31:26	**CAP1PSC**	通道1输入预分频 (Capture channel 1 prescaler) 0x00：不分频 ..... 0x3F：64分频
25:24	**CAP1SSEL**	通道 1 捕捉源选择(Capture channel 1 source select)，仅在 CH1 通道配置为输入捕捉时有效 00：LPT16_CH1 输入 01：XTLF 10：RCLP 11：RFU
23	--	未实现：读为0
22	--	未实现：读为0
21	**CAP2EDGE**	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新(Channel2 Captured Edge) 0：上升沿 1：下降沿
20	**CAP1EDGE**	通道 1 当前被捕捉的边沿，在 CC1IF 置位时更新(Channel 1 Captured Edge) 0：上升沿 1：下降沿
19	--	未实现：读为0
18	--	未实现：读为0
17	**CC2P**	通道2配置为比较功能时：通道2比较输出波形极性选择 (Channel 2 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
		通道2配置为输入捕捉功能时：通道2输入波形极性选择 (Channel 2 input polarity) 0：输入不取反 1：输入取反
16	**CC1P**	通道1比较输出波形极性选择 (Channel 1 compare output Polarity) 0：正极性波形，起始为低，计数值==比较值时置高，计数值==ARR时恢复为低 1：负极性波形，正极性波形取反
		通道1配置为输入捕捉功能时：通道1输入波形极性选择 (Channel 1 input polarity) 0：输入不取反 1：输入取反
15:14	--	未实现：读为0
13	**CC2DF**	通道2输入数字滤波 (Channel 2 input digital filter) 1：使能数字滤波 0：关闭数字滤波

位号	位名	说明
12	**CC1DF**	通道1输入数字滤波 (Channel 1 input digital filter) 1：使能数字滤波 0：关闭数字滤波
11:10	**CAPCFG2**	通道 2 捕捉边沿选择(Channel 2 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
9:8	**CAPCFG1**	通道 1 捕捉边沿选择(Channel 1 Capture edge Config) 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
7:6	--	未实现：读为0
5:4	--	未实现：读为0
3:2	**CC2S**	通道 2 捕捉/比较功能使能(Channel 2 Capture/Compare Select) 00,11：禁止通道 2 捕捉/比较功能 01：使能通道 2 捕捉功能（LPT16_CH2 为输入） 10：使能通道2比较功能（LPT16_CH2为输出）
1:0	**CC1S**	通道 1 捕捉/比较功能使能(Channel 1 Capture/Compare Select) 00,11：禁止通道 1 捕捉/比较功能 01：使能通道 1 捕捉功能（LPT16_CH1 为输入） 10：使能通道 1 比较功能（LPT16_CH1 为输出）

28.7.4 LPTIM 目标值寄存器（LPTIM16_ARR）

名称	LPTIM16_ARR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	未实现：读为0
15:0	**ARR**	自动重载目标寄存器(Auto-Reload Register) 当计数器计数值等于 ARR 时，计数器回到初值重新开始向上计数

28.7.5 LPTIM 中断使能寄存器（LPTIM16_IER）

名称	LPTIM16_IER							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OVR2IE	OVR1IE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	QDCIE	-			CC2IE	CC1IE
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:10	--	**未实现：读为0**
9	**OVR2IE**	通道 2 捕捉溢出中断使能(Channel 2 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
8	**OVR1IE**	通道 1 捕捉溢出中断使能(Channel 1 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
7	**TRIGIE**	外部触发到来中断使能位(External Trigger Interrupt Enable) 1：外部触发到来中断使能 0：外部触发到来中断禁止
6	**OVIE**	计数器溢出中断使能位(Counter Over-Flow Interrupt Enable) 1：计数器溢出中断使能 0：计数器溢出中断禁止
5	**QDCIE**	Quadrature Decoder 方向改变中断使能 (Quad-decoder Interrupt Enable) 1：使能 QDCIF 中断 0：禁止 QDCIF 中断
4:2	--	**未实现：读为0**
1	**CC2IE**	捕捉/比较通道 2 中断使能位(Capture/Compare channel 2 Interrupt Enable) 1：捕捉/比较通道 2 中断使能 0：捕捉/比较通道 2 中断禁止
0	**CC1IE**	捕捉/比较通道 1 中断使能位(Capture/Compare channel 1 Interrupt Enable) 1：捕捉/比较通道 1 中断使能 0：捕捉/比较通道 1 中断禁止

28.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR）

名称	LPTIM16_ISR

offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名			QDIR	-			CAP2OVR	CAP1OVR
位权限			R-0	U-0			R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	QDCIF	-			CC2IF	CC1IF
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:14	--	**未实现：读为0**
13	**QDIR**	Quadrature Decoder 方向寄存器 (Quad-decoder Direction) 0：CH1 领先于 CH2 1：CH1 滞后于 CH2
12:10	--	**未实现：读为0**
9	**CAP2OVR**	通道 2 捕捉溢出，硬件置位，软件写 1 清零(Channel 2 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC2IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
8	**CAP1OVR**	通道 1 捕捉溢出，硬件置位，软件写 1 清零(Channel 1 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC1IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
7	**TRIGIF**	外部触发到来中断标志位，写 1 清零(External Trigger Interrupt Flag) 1：外部触发到来中断产生 0：无中断产生
6	**OVIF**	计数器溢出中断使能位，写 1 清零(Counter Over-Flow Interrupt Flag) 1：计数器溢出中断产生 0：无中断产生
5	**QDCIF**	Quadrature Decoder 方向改变中断，硬件置位，软件写 1 清零(Quad-decoder Direction Change Interrupt Flag) 0：没有改变方向 1：方向改变
4:2	--	**未实现：读为0**
1	**CC2IF**	捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 2 Interrupt Flag) 1：计数器值和比较值 2 匹配，或者发生捕捉事件 0：无中断产生
0	**CC1IF**	捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 1 Interrupt Flag) 1：计数器值和比较值 1 匹配，或者发生捕捉事件 0：无中断产生

28.7.7 LPTIM 控制寄存器（LPTIM16_CR）

名称	LPTIM16_CR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	位名	说明
31:1	--	**未实现：读为0**
0	**EN**	LPTIM 使能位(LPTIM Enable) 1：使能计数器计数 0：禁止计数器计数

28.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1）

名称	LPTIM16_CCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	**未实现：读为0**
15:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

28.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2）

名称	LPTIM16_CCR2

offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	**--**	**未实现：读为0**
15:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

29 实时时钟（RTC）

29.1 概述

实时时钟(RTC)模块可长时间维持精确计时，功耗极低，在所有功耗模式下都可以工作。

主要特性如下：

- BCD 时间，完整万年历（00~99 年）
- 周期唤醒中断
- 闹钟功能
- 可配置周期定时信号输出
- 数字调校，精度+/-0.477ppm
- 反馈电阻集成
- RTC 计时器部分不复位

29.2 结构框图

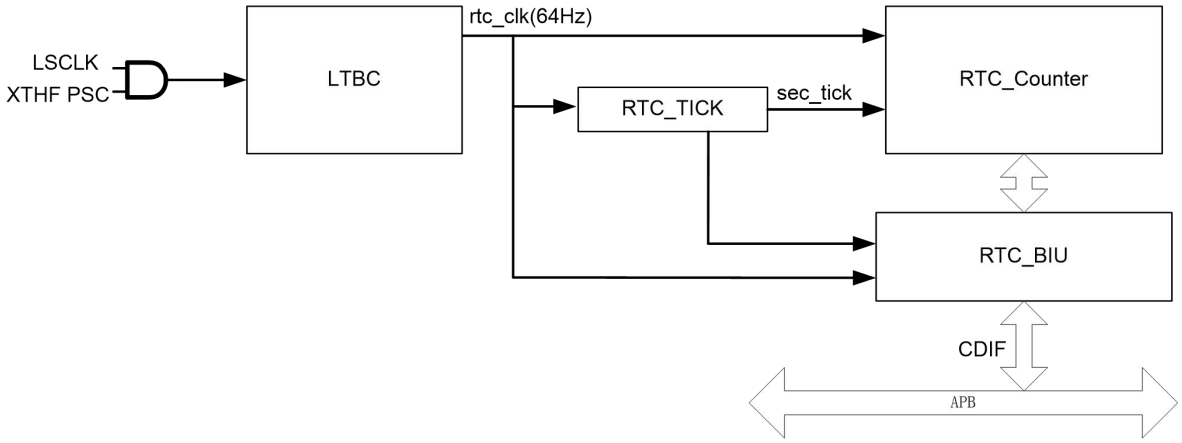


图 29-1RTC 结构框图

LTBC 模块为低功耗时基计数器模块，用于产生系统所需的低速工作时钟，具体描述见 29.3.1 节 LTBC 功能介绍。

RTC_TICK 模块主要用来产生每秒跳变的秒冲信号 sec_tick 以及产生中断需要的 2Hz，4Hz，8Hz，16Hz 等信号。Sec_tick 信号输出到 RTC_Counter 模块用来实现万年历的计数器同步。

RTC_Counter 模块是 RTC 的万年历实现模块，包括秒计数器，分钟计数器，小时计数器，天计数器，周天计数器，月计数器以及年计数器。模块可以实现闰年的自动识别。

29.3 工作原理

RTC 上电后不复位，因此正常工作前需要软件置入当前时间。走时时钟最好使用 32.768KHz 晶体振荡器。由于晶体振荡器有可能停振，为了保证可靠性，停振检测电路使能后不断检测 32.768KHz 振荡器输出，一旦发现停振，则产生报警中断。同时，软件可以配置 XTLF 停振时是否自动将 RTC 时钟切换到 RCLP，如果使能这一功能，则 RTC 走时有一定误差，但是并不会停止；如果未使能自动切换，也可以由软件响应停振中断后进行相应处理。

29.3.1 时基计数器（LTBC）

低功耗时基计数器(LTBC)模块用于产生系统所需的低速工作时钟，功能包括：

- 通过对 LSCLK 的预分频得到 64Hz 的 RTC 工作时钟
- 可通过调整计数周期实现 RTC 时钟的数字调校，每 32s 调校一次可实现最小步长为 0.952ppm，调校后理论精度+/-0.476ppm
- 可产生 1KHz、256Hz、64Hz、16Hz、4Hz、1Hz 周期中断，其中 1K 和 256Hz 是未经调校的，其他是经过数字调校的（如果使能了数字调校）
- 64Hz 预分频电路不受芯片复位影响
- 1/256s 精度授时

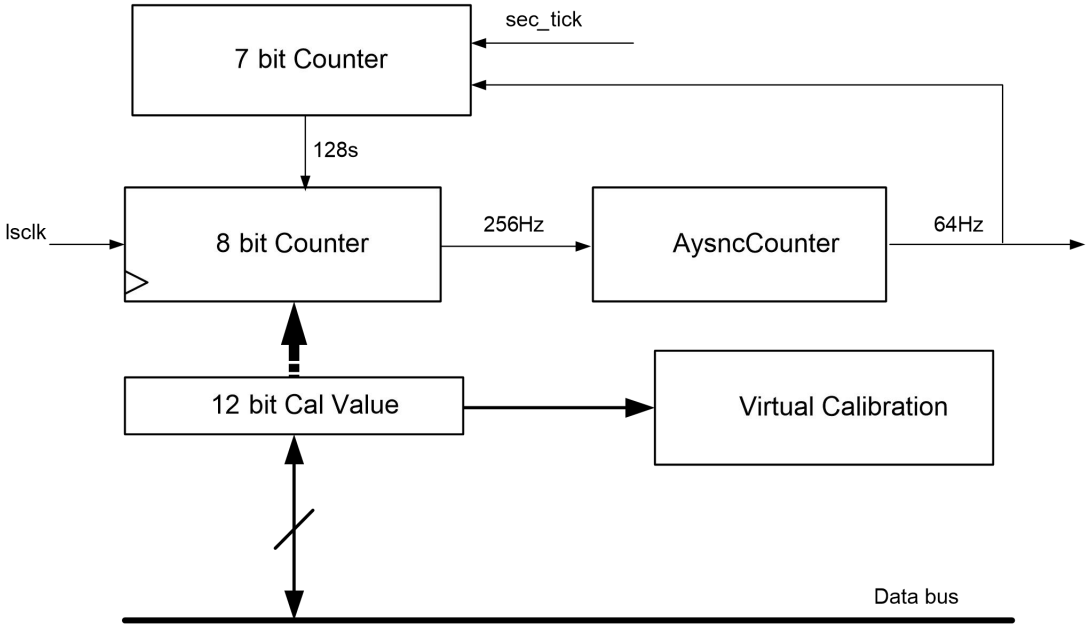


图 29-2LTBC 结构框图

29.3.2 LTBC 数字调校

LTBC主要由同步预分频计数器、异步分频计数器、时钟调校值寄存器、虚拟调校电路和控制寄存器组成。

数字调校的目的是使RTC能够在较长周期内获得平均准确的计时。由于RTC的时钟源是32768Hz，因此数字调校的最小步长是30.5us，如果在1秒内调整一次，则最高精度只能达到30.517ppm。为了得到更高精度，必须在更长时间周期内进行调整。FM33FR0以32s为一个调校周期，每个周期内可以调整0~+/-511个32768Hz时钟周期，因此最高精度为30.5us/32s=0.952ppm，最大调校范围为+/-(511*30.517us/32s)=+/-487ppm，调校后平均最小时钟误差为+/-0.476ppm。

调校值由10bit寄存器组成，其中最高位为符号位，表示计数值增减，其余9bit表示增减的绝对值。为了提高每秒的平均精度，避免较大的秒间跃变，采取将32s调校值平均分配到每秒内的做法，其实现方法如下：

除了最高符号位，其余9bit可分为高4bit的公共值和5bit私有值，其中公共值表示32s内每秒都要调整的值，私有值表示32s内部分秒需要加减1。

Bit9	Bit[8:5]	Bit[4:0]
Sign	Common Value （C）	Differential Value (D)

调校值公式可表示为：Correction(ppm) = (C*32 + D)*30.517/32000000

假设只使时钟增加0.953ppm，即32s周期只增加一个30.5us，调校值写为0_0000_00001，所以公共值为0，私有值为1，只需要对32s内的一个秒周期加1即可；假设增加487ppm，即32s周期内增加511个30.5us，调校值写为0_1111_11111，公共值为15，私有值为31，表示32s中每秒都要加15，同时其中还有31s需要额外加1。

调校值举例：

ppm	ADJUST[1]	Common	Differential	Expression
0.953	0_0000_00001	0	1	1*30.517/32000000
-125.88	1_0100_00100	4	4	(4*32+4)*30.517/32000000
32.42	0_0001_00010	1	2	(1*32+2)*30.517/32000000
487.32	0_1111_11111	15	31	(15*32+31)*30.517/32000000

注：

[1] ADJUST: Clock Error Adjustment Register

通过以上方式，可以得到平滑调整的秒周期，秒和秒之间最大只差30.5ppm，可以避免集中式调整引入的秒周期抖动。

此方案的另一个好处是，由于每个秒周期内最多只会加减15+1=16个32768Hz时钟，这表示同步计数预分频器的长度可以做的很短，理论上只要8bit（正常计数范围0~127，调校时扩展计数范围0~143，终值下限111，上限143），得到典型值256Hz同步分频输出，随后使用异步分频得到64Hz RTC走时时钟即可，较少的同步时钟负载能够降低功耗、增强EMC抗干扰性能。

为避免时序冲突，软件应在秒中断后更新ADJUST并启动时钟调校。

以ADJUST=0_0001_0000000为例，在每一秒的最后添加1个32768周期。

29.3.3 BCD 时间

秒计时

秒计时仅需7bit，从0计数到59，其中bit[3:0]为1秒单位，计数范围0-9；bit[6:4]为10秒单位，计数范围0-5。当计数满60s后触发秒进位信号使分钟计数器加1。

Bit6-4	Bit3-0
0-5	0-9

分钟计时

分计时也仅需7bit，计数范围与秒相同，因此实现方法也相同。

Bit6-4	Bit3-0
0-5	0-9

小时计时

小时计数范围为 0-24，仅需 6bit：

Bit5-4	Bit3-0
0-2	0-9

天计时

天计数范围为 1-31，仅需 6bit，从 1 开始计数，根据月份以及闰年计数到 28/29/30/31，计满后触发天进位信号使月计数器加 1。

Bit5-4	Bit3-0
0-3	0-9

星期计时

星期计数范围为 0-6，仅需 3bit，从 0 到 6 循环计数。

Bit2-0
0-6

月计时

月计数范围为 1-12，仅需 5bit，从 1 开始计数到 12，计满后触发月进位信号使年计数器加 1。

Bit4	Bit3-0
0-1	0-9

年计时

年计数范围为 0-99，需 8bit，从 0 到 99 循环计数。

Bit7-4	Bit3-0
0-9	0-9

29.3.4 RTC 使能与停止

RTC 可以由软件使能或关闭。上电后 RTC 默认关闭，软件在 XTLF 起振后通过操作 EN 寄存器使能 RTC 走时。

当 EN 寄存器清零时，RTC 内部走时时钟被门控，以节省功耗。

29.3.5 RTC 时间设置

软件可以在任意时刻直接设置 RTC 时间寄存器，通常建议在 XTLF 完成起振后再设置时间；由于设置时间寄存器的操作与 RTC 走时为异步操作关系，建议软件在秒中断事件之后进行时间设置，并且在置时后读出时间值校验。

注意，硬件并不检查时间合法性，软件须保证写入的 BCD 时间正确。

同时 FM33FR0 支持 ms 级授时，即可以设置时间到 3.9ms 级别精度（1/256s）。此外，当软件写入秒时间时，硬件自动清零 64Hz->1Hz 的秒内计数器，以便实现秒对齐。

为了提高抗干扰能力，FM33FR0 提供时间写保护功能，必须先对写保护寄存器写入 0xACACACAC，才能改写时间寄存器，置时完成后软件可以通过写入任意其他值来禁止时间寄存器的写入，恢复写保护。

29.3.6 RTC 时间读取

时间读取方式 1：

- 读当前时间寄存器值
- 再次读当前时间寄存器值

- 如果 2 次读取内容一致，则为正确的当前时间；如果两次读取内容不一致，则重复前两个步骤。

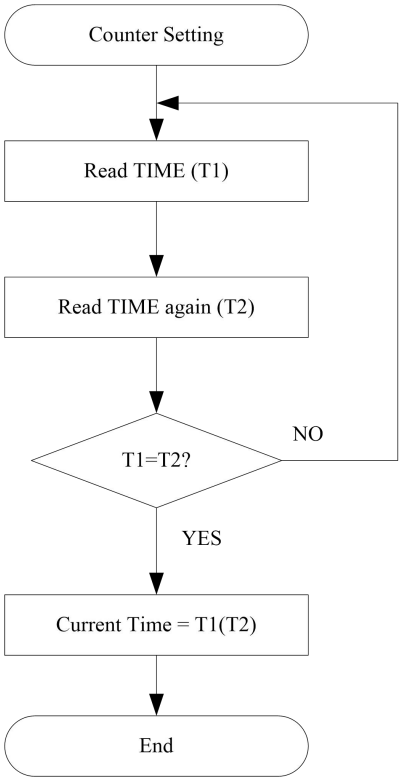


图 29-3RTC 时间读取流程图

时间读取方式 2：

软件在 1s 中断发生后立即读取时间寄存器，能保证读到正确的当前时间值。

29.3.7 闰年判断

FM33FR0 的 RTC 模块会自动判断闰年。

闰年的条件：(mod 400 == 0) or (mod 4 ==0 and mod 100 <> 0)

这里只考虑 2000-2099 范围，判断方法是，十进制的年份数低 2 位为 00 就是闰年，由于寄存器中写的是 BCD 码，判断前需要把 BCD 码转换成十进制，方法是 year 的高 4 位乘 10 加上低 4 位，其中乘 10 可以用*8+*2 的方式实现。

29.4 寄存器

模块基地址：0x4001_1000

offset 地址	名称	符号
0x00000000	RTC 写使能寄存器（RTC Write Enable Register）	RTC_WER
0x00000004	RTC 中断使能寄存器（RTC Interrupt Enable Register）	RTC_IER
0x00000008	RTC 中断标志寄存器（RTC Interrupt Status Register）	RTC_ISR
0x0000000C	BCD 时间秒寄存器（BCD format time second registers）	RTC_BCDSEC
0x00000010	BCD 时间分钟寄存器（BCD format time minute registers）	RTC_BCDMIN
0x00000014	BCD 时间小时寄存器（BCD format time hour registers）	RTC_BCDHOUR
0x00000018	BCD 时间天寄存器（BCD format time day registers）	RTC_BCDDAY
0x0000001C	BCD 时间星期寄存器（BCD format time week registers）	RTC_BCDWEEK
0x00000020	BCD 时间月寄存器（BCD format time month registers）	RTC_BCDMONTH
0x00000024	BCD 时间年寄存器（BCD format time year registers）	RTC_BCDYEAR
0x00000028	闹钟寄存器（RTC Alarm Register）	RTC_ALARM
0x0000002C	RTC 时间信号输出寄存器（RTC Time Mark Select）	RTC_TMSEL
0x00000030	LTBC 数值调整寄存器（RTC time Adjust Register）	RTC_ADJUST
0x0000003C	毫秒计数值寄存器（RTC Sub-Second Counter）	RTC_SBSCNT
0x00000040	RTC 控制寄存器（RTC Control Register）	RTC_CR

29.4.1 RTC 写使能寄存器（RTC_WER）

名称	RTC_WER							
Offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	WE
位权限	U-0	R/W-0

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	WE	RTC 写使能寄存器 (RTC Write Enable) 当 CPU 向 RTCWE 写入 0xACACACAC 时，允许 CPU 向 RTC 的 BCD 时间寄存器写入初值，这时 WE 置 1；当 CPU 向 RTCWE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 WE 清 0。

29.4.2 RTC 中断使能寄存器（RTC_IER）

名称	RTC_IER							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			ADJ_IE	ALARM_IE	1KHZ_IE	256HZ_IE	64HZ_IE
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IE	8HZ_IE	4HZ_IE	2HZ_IE	SEC_IE	MIN_IE	HOUR_IE	DAY_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IE	调校周期中断使能 (time Adjust Interrupt Enable) 1：中断使能打开 0：中断使能禁止
11	ALARM_IE	闹钟中断使能 (Alarm Interrupt Enable) 1：中断使能打开 0：中断使能禁止
10	1KHZ_IE	1khz 中断使能 (1Khz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
9	256HZ_IE	256hz 中断使能 (256hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
8	64HZ_IE	64hz 中断使能 (64hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
7	16HZ_IE	16hz 中断使能 (16hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止

位号	助记符	功能描述
6	8HZ_IE	8hz 中断使能 (8hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
5	4HZ_IE	4hz 中断使能 (4hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
4	2HZ_IE	2hz 中断使能 (2hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
3	SEC_IE	秒中断使能 (1hz periodic Interrupt Enable) 1：中断使能打开 0：中断使能禁止
2	MIN_IE	分中断使能 (Minute Interrupt Enable) 1：中断使能打开 0：中断使能禁止
1	HOUR_IE	小时中断使能 (Hour Interrupt Enable) 1：中断使能打开 0：中断使能禁止
0	DAY_IE	天中断使能 (Day Interrupt Enable) 1：中断使能打开 0：中断使能禁止

29.4.3 RTC 中断标志寄存器（RTC_ISR）

名称	RTC_ISR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			ADJ_IF	ALARM_IF	1KHZ_IF	256HZ_IF	64HZ_IF
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IF	8HZ_IF	4HZ_IF	2HZ_IF	SEC_IF	MIN_IF	HOUR_IF	DAY_IF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12	ADJ_IF	调校周期中断标志。写 1 清零 (time Adjust Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
11	ALARM_IF	闹钟中断标志。写 1 清零 (Alarm Interrupt Flag, write 1 to clear) 1：中断置位

位号	助记符	功能描述
		0：无中断产生
10	1KHZ_IF	1khz 中断标志。写 1 清零 (1Khz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
9	256HZ_IF	256hz 中断标志。写 1 清零 (256hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
8	64HZ_IF	64hz 中断标志。写 1 清零 (64hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
7	16HZ_IF	16hz 中断标志。写 1 清零 (16hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
6	8HZ_IF	8hz 中断标志。写 1 清零 (8hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
5	4HZ_IF	4hz 中断标志。写 1 清零 (4hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
4	2HZ_IF	2hz 中断标志。写 1 清零 (2hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
3	SEC_IF	秒中断标志。写 1 清零 (1hz periodic Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
2	MIN_IF	分中断标志。写 1 清零 (Minute Interrupt Flag, write 1 to clear) 1：中断置位 0：无中断产生
1	HOUR_IF	小时中断标志。写 1 清零 (Hour Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生
0	DAY_IF	天中断标志。写 1 清零 (Day Interrupt Flag,write 1 to clear) 1：中断置位 0：无中断产生

29.4.4 BCD 时间秒寄存器（RTC_BCDSEC）

名称	RTC_BCDSEC							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	SEC	秒时间数值，BCD 格式。(Binary-Coded Decimal format Seconds Register)

29.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）

名称	RTC_BCDMIN							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MIN						
位权限	U-0	R/W-xxx xxxx						

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
6:0	MIN	分钟时间数值，BCD 格式。(Binary-Coded Decimal format Minutes Register)

29.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）

名称	RTC_BCDHOUR							
Offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		HOUR					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	HOUR	小时数值，BCD 格式。(Binary-Coded Decimal format Hours Register)

29.4.7 BCD 时间天寄存器（RTC_BCDDAY）

名称	RTC_BCDDAY							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAY					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	DAY	天数值，BCD 格式。(Binary-Coded Decimal format Date Register)

29.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）

名称	RTC_BCDWEEK							
Offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					WEEK		
位权限	U-0					R/W-xxx		

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	WEEK	周数值，BCD 格式。(Binary-Coded Decimal format Week Register)

29.4.9 BCD 时间月寄存器（RTC_BCDMONTH）

名称	RTC_BCDMONTH							
Offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			MONTH				
位权限	U-0			R/W-x xxxx				

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4:0	MONTH	月数值，BCD 格式。(Binary-Coded Decimal format Month Register)

29.4.10 BCD 时间年寄存器（RTC_BCDYEAR）

名称	RTC_BCDYEAR							
Offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	YEAR							

位权限	R/W-xxxx xxxx

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	YEAR	年数值，BCD 格式。(Binary-Coded Decimal format Year Register)

29.4.11 闹钟寄存器（RTC_ALARM）

名称	RTC_ALARM							
Offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HOUR					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
位名	-	MIN						
位权限	U-0	R/W-000 0000						
位	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
位名	-	SEC						
位权限	U-0	R/W-000 0000						

位号	助记符	功能描述
31:22	-	RFU：未实现，读为 0
21:16	HOUR	闹钟的小时数值。(Alarm Hour Register)
15	-	RFU：未实现，读为 0
14:8	MIN	闹钟的分数值。(Alarm Minute Register)
7	-	RFU：未实现，读为 0
6:0	SEC	闹钟的秒数值。(Alarm Second Register)

29.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）

名称	RTC_TMSEL							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TMSEL			

位权限	U-0	R/W-0000

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3:0	TMSEL	频率输出选择信号：(Time Mark Select) 0000：RFU 0001：RFU 0010：输出秒计数器进位信号，高电平宽度 1s 0011：输出分计数器进位信号，高电平宽度 1s 0100：输出小时计数器进位信号，高电平宽度 1s 0101：输出天计数器进位信号，高电平宽度 1s 0110：输出闹钟匹配信号 0111：输出 32 秒方波信号 1000：RFU 1001：反向输出秒计数器进位信号 1010：反向输出分计数器进位信号 1011：反向输出小时计数器进位信号 1100：反向输出天计数器进位信号 1101：反向输出闹钟匹配信号 1110：RFU 1111：输出 RTC 内部秒时标方波

29.4.13 LTBC 数值调整寄存器（RTC_ADJUST）

名称	RTC_ADJUST							
Offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ADSIGN	ADJUST[8]
位权限	U-0						R/W-x	R/W-x
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJUST[7:0]							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	ADSIGN	LTBC 补偿方向 (Adjust Sign) 0：表示增加计数初值

位号	助记符	功能描述
		1：表示减少计数初值
8:0	ADJUST	LTBC 补偿调整数值 (Time Adjust)

29.4.14 毫秒计数值寄存器（RTC_SBSCNT）

名称	RTC_SBSCNT							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSCNT							
位权限	R/W-xxxx xxxx							

位号	助记符	功能描述
31:8	-	RFU：未实现，读为 0
7:0	MSCNT	毫秒计数器值，有效位 8bit，精度 3.9ms。(Milli-Second Counter)

29.4.15 RTC 控制寄存器（RTC_CR）

名称	RTC_CR							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	RTC 使能寄存器 (RTC enable) 0：关闭 RTC（内部走时时钟关闭） 1：使能 RTC

30 段码液晶/LED 驱动（LCD/LED）

30.1 概述

LCD 显示驱动模块用于驱动段码式液晶屏，能够支持 4、6、8COM，最大显示段数分别为 176 段（4COM）、252 段（6COM）和 320 段（8COM）。

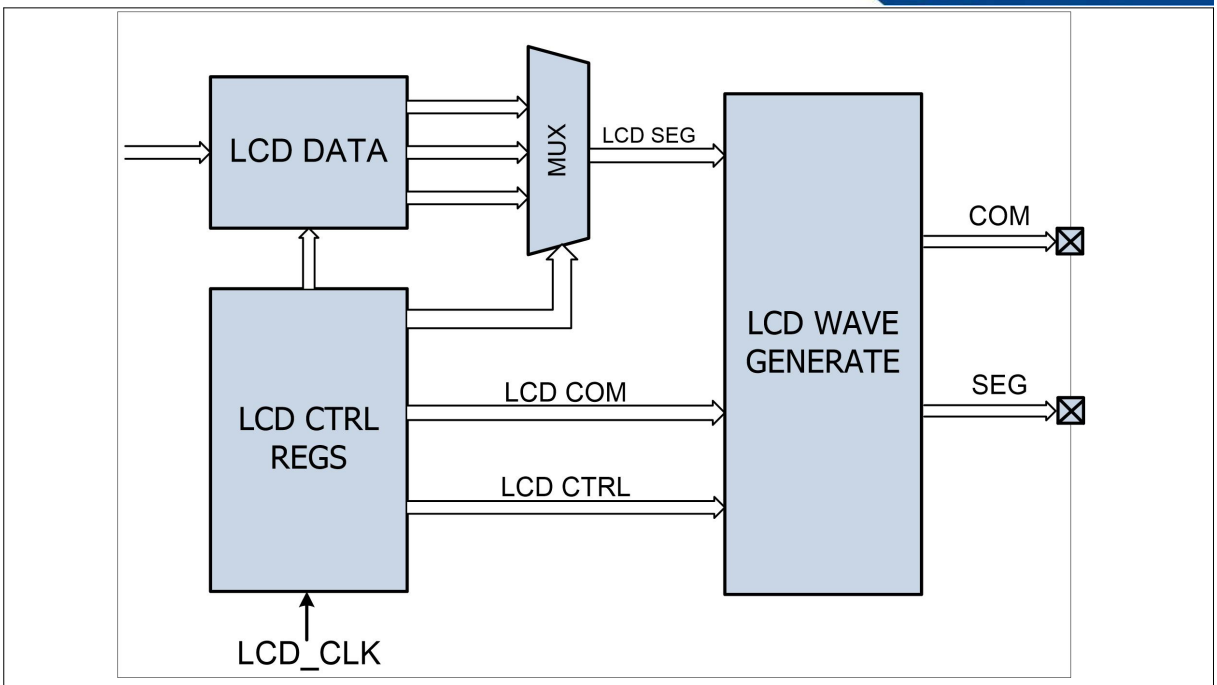
LED 显示驱动支持最大 8×32 的 LED 数码管扫描功能。

主要特点：

- 最大支持 8×40、6×42、4×44 的 LCD 显示段数
- 最大支持 8×32 的 LED 驱动段数
- LCD 支持 1/3bias、1/4bias
- LCD 支持 16 级灰度可调
- LCD 驱动支持片内电阻型模式
- 支持闪烁功能，且闪烁频率可调
- 支持间歇式点亮功能，点亮、熄灭时间可配置
- 支持全亮、全灭功能
- 低功耗，LCD 驱动可以在 Active 模式、Sleep 模式和 DeepSleep 模式下工作
- 支持 Type A 和 Type B 两种 LCD 驱动波形（可配置）
- 典型帧刷新频率 64Hz

30.2 结构框图

LCD 显示驱动模块结构框图如图 30-1 所示：



片内 BUFFER 驱动模式

图 30-1 LCD 显示控制模块结构框图

30.3 IO 配置

LCD驱动电路工作时最多会占用48个GPIO，在使用LCD前，需要将用到的引脚设置为模拟功能（GPIOx_FCR=11），并将相应的COMEN或SEGEN打开。如果在同一个引脚上复用了LCD之外的其他模拟功能，则必须保证其他模式外设不会使用这个引脚通道。

LED驱动电路使用GPIO的数字输出功能，使用LED扫描驱动之前，需要将用到的引脚配置为数字外设功能并选择LED。

30.4 功能说明

30.4.1 工作时钟和显示帧频率

LCD/LED驱动电路使用LSCLK工作，其典型频率在32.768Hz左右。通过配置DF寄存器，可以设置LCD显示的帧频率。帧频率的计算公式如下（注意DF不能为0）：

COM 数量	帧频率 Hz	
	A 类波形	B 类波形
4	显示电路工作频率 / (4 ×DF[7:0] × 2)	显示电路工作频率 / (4 ×DF[7:0] × 4)
6	显示电路工作频率 / (6 ×DF[7:0] × 2)	显示电路工作频率 / (6 ×DF[7:0] × 4)
8	显示电路工作频率 / (8 ×DF[7:0] × 2)	显示电路工作频率 / (8 ×DF[7:0] × 4)

表30-1帧频率计算公式

下表举例说明了DF寄存器的取值与帧频率之间的关系。通常情况下将帧频率设置为60Hz左右。

帧频率（Hz）	工作时钟（Hz）	4 公共端		6 公共端		8 公共端	
		A 类	B 类	A 类	B 类	A 类	B 类
50	32768	82	41	54	27	41	20
58	32768	70	35	47	24	35	17
64	32768	64	32	42	21	32	16
70	32768	58	29	39	20	29	14
75	32768	54	27	36	18	27	13

表30-2典型帧频率和DF的关系

30.4.2 LED 扫描波形

LED扫描的SEG和DIG波形示意图如下。

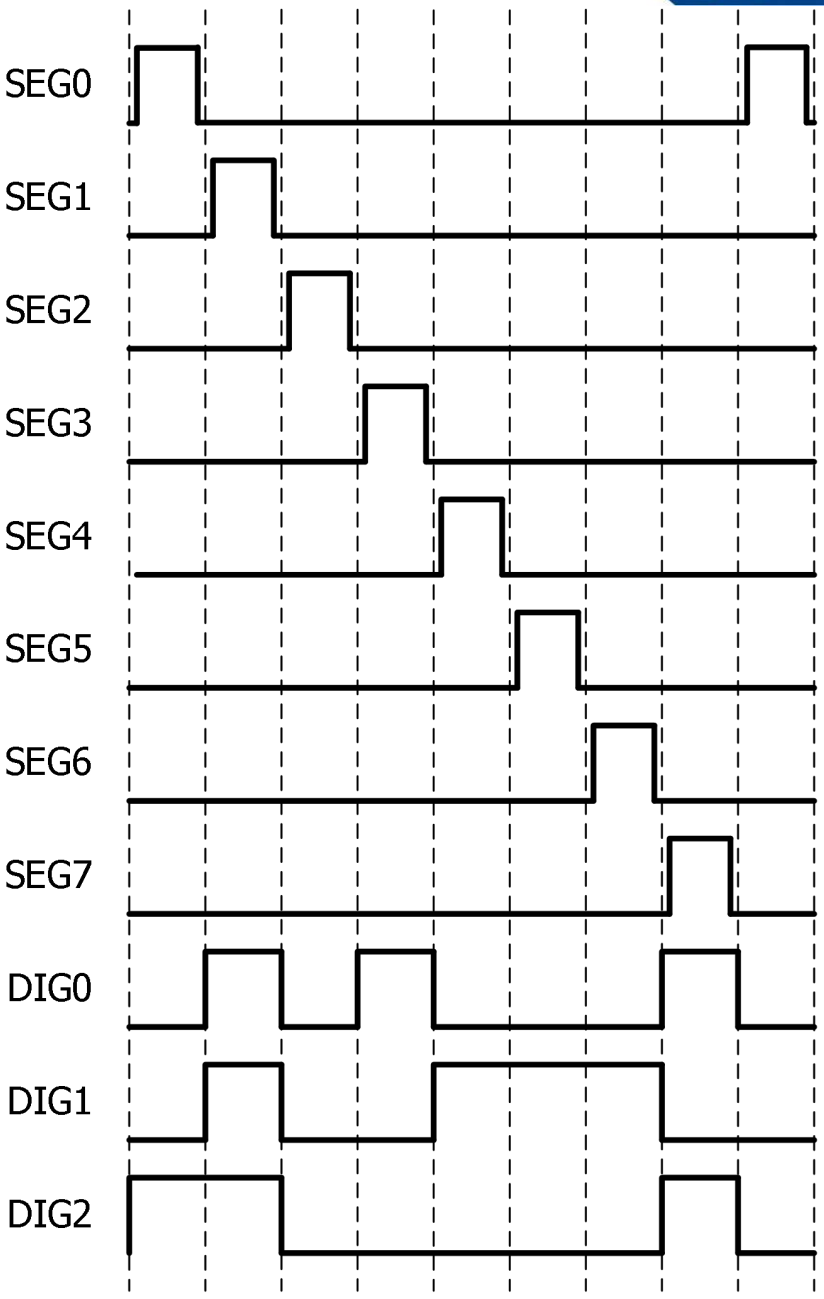


图 30-2LED 驱动波形(8 SEG, 3 DIG)

30.4.3 LCD Type A 扫描波形

下图是1/4 duty，1/3bias，TypeA类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

4个公共端会依次有效，在某个公共端有效的时间内，SEG输出合适的电平，与COM电平共同施加在LCD面板上，压差大的段码将被显示，压差小的段码不显示。

1/3 bias表示LCD驱动电路能够输出4种驱动电平，对偏置电压平均分配得到。通过LCDBIAS寄存器可以配置VLCD偏置电压大小，最大不超过电源电压。

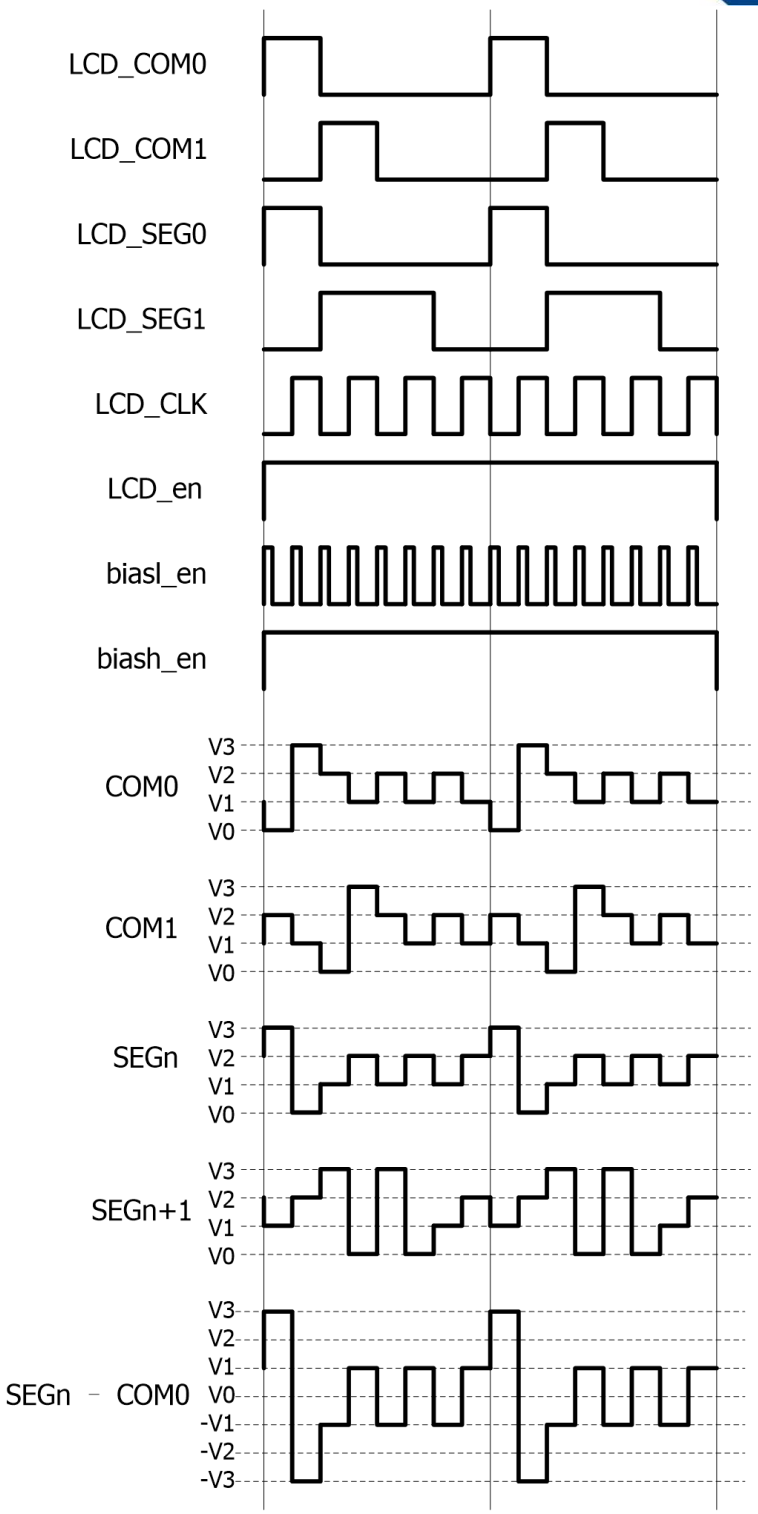


图 30-3LCD 驱动波形(1/4 duty, 1/3 bias, type A)

30.4.4 LCD Type B 扫描波形

下图是1/4 duty，1/3bias，TypeB类波形的LCD扫描波形示意图。这里只画出了两个公共端的示例。

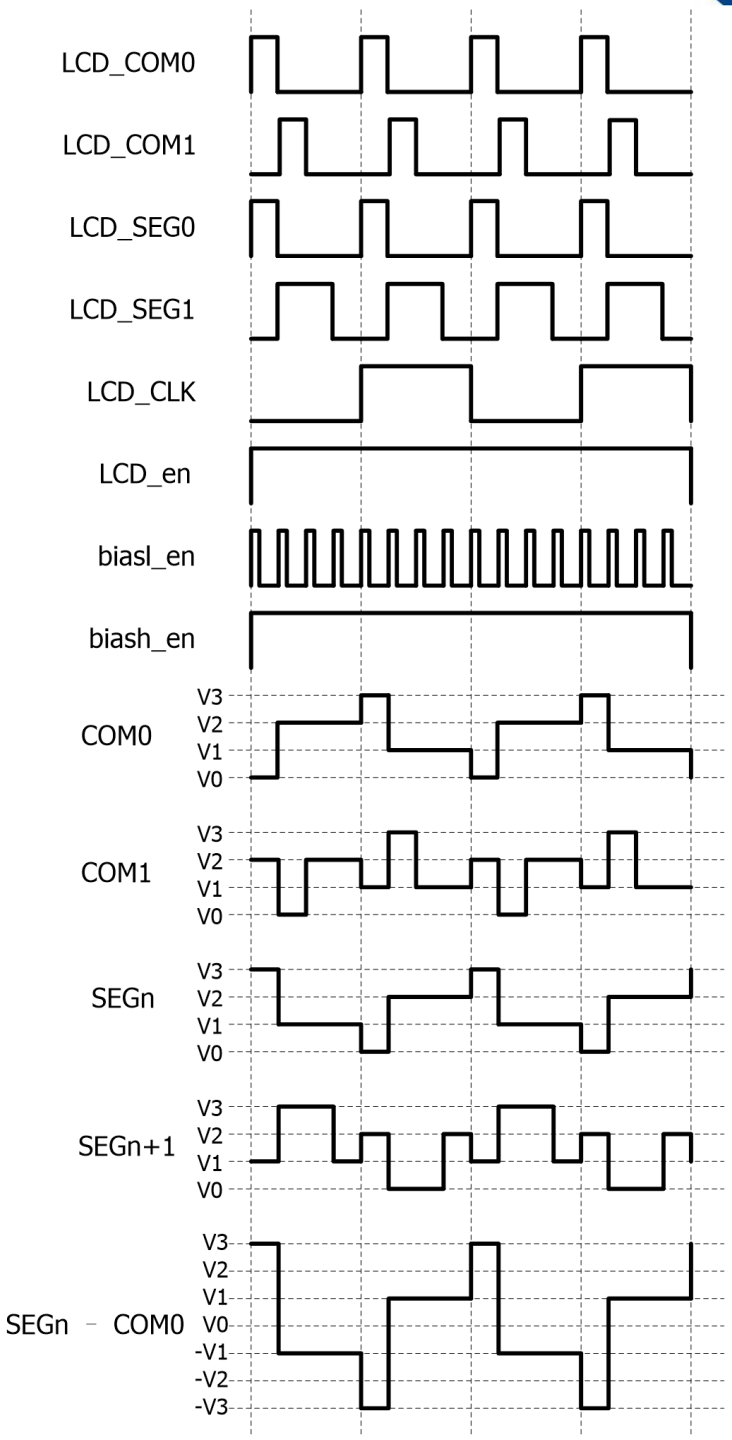


图 30-4 LCD 驱动波形(1/4 duty, 1/3 bias, type B)

30.4.5 片内 buffer 驱动模式

片内buffer驱动模式由电源电压通过分压电阻产生等分电压，分压输入到低功耗buffer以增强驱动能力，buffer输出连接至波形产生模块后产生COM和SEG信号，此模式无需片外器件，功耗较低。

其结构示意图如下：

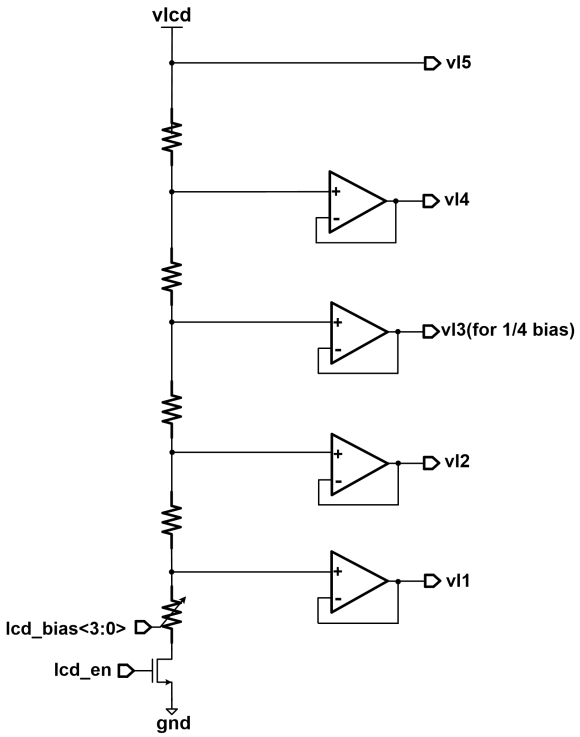


图 30- 5LCD 片内电阻 buffer 型驱动电路

驱动Buffer的输出阻抗大约为5Kohm。

通过配置LCDBIAS寄存器，可以实现对LCD驱动电压的调整，具体参见30.4.7“偏置电压调整”章节。

30.4.6 显示闪烁功能

软件可以设置显示控制寄存器 DISP_CR 中的 FLICK 位为 1，来使能显示闪烁。FLICK 使能后，根据 TON 和 TOFF 寄存器值确定闪烁频率。在使能 FLICK 功能之前应先设置 TON/TOFF 并设置 LCDEN 打开显示，若不设置 TON/TOFF，则其复位值为 0，显示会以 64Hz 闪烁。若不先打开显示，FLICK 设置无效，不会有显示。

TON/TOFF 最小步长为 T_{step} = COM*DF[7:0]*2*16/32768Hz，实际 ON/OFF 时间为 TON/TOFF * T_{step}。显示和熄灭与帧扫描同步，即在一帧扫描完后熄灭，或在一帧开始时点亮，熄灭或点亮后给出相应中断。由于帧结束信号是 64HZ 的，因此 TON/TOFF 的计数值应为寄存器设置值 x16。

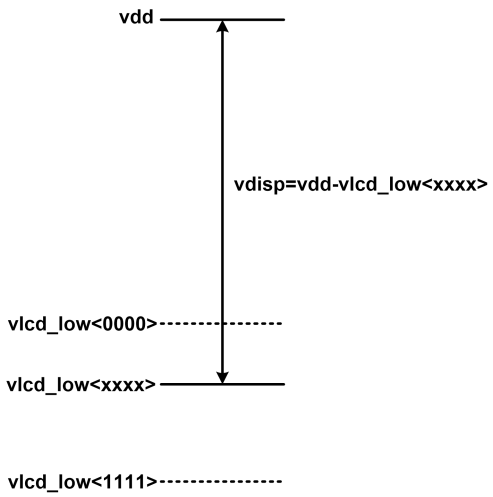
30.4.7 偏置电压调整

LCD 输出的显示电压范围可以调节以适应不同规格的液晶面板，输出电压范围可以表示为：

VDISP=VDD-VLCD_LOW

其中 VLCD_LOW 可以由 LCDBIAS[3:0]调节，LCDBIAS=0000 对应的 VLCD_LOW 电压最高，输出电压范围 VDISP=VDD-VLCD_LOW 最小；LCDBIAS=1111 对应的 VLCD_LOW 电压最低，输出

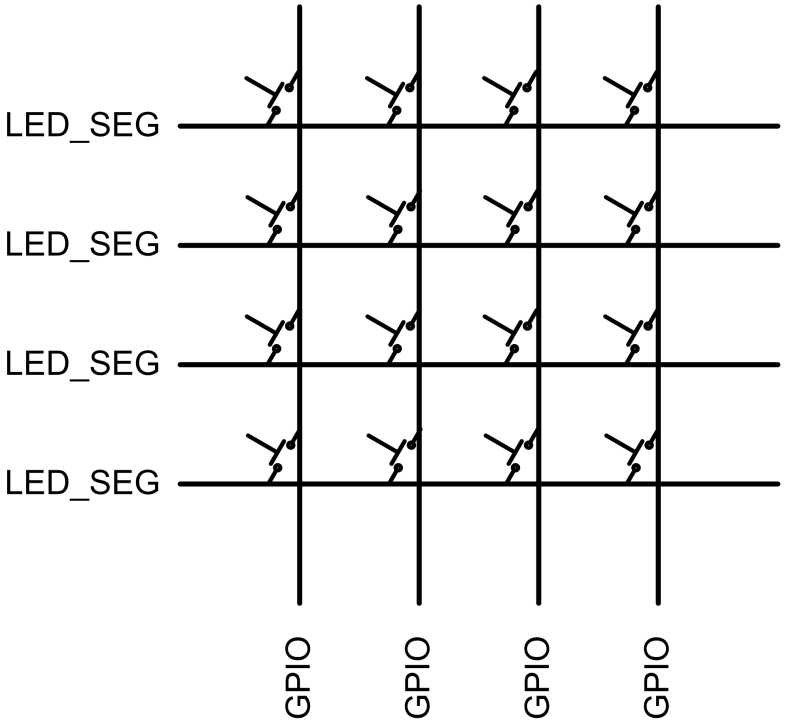
电压范围 VDISP=VDD-VLCD_LOW 最大，如下图所示：



应用中应根据实际 LCD 面板特性，选择合适的 VDISP 电压，可参考 30.6.1 中的表格进行配置。

30.4.8 使用 LED 扫描功能实现键盘扫描

以 4x4 键盘扫描为例，连接关系参见下图。



LED SEG 顺序扫描各行按键阵列，GPIO 则配置为输入，打开 EXTI 中断。

当某个按键被按下时，所在列的 GPIO 产生中断事件，通知 MCU 本列有按键按下。此时 MCU 可以去查询 SEG 脚的状态，为高电平的 SEG 脚所对应按键即为被按下的按键。

软件可以通过多轮扫描中断反复确认来实现按键去抖。

30.5 低功耗模式

当芯片进入低功耗模式时，LCD 仍然可以正常工作。当主电源掉电时，LCD 停止工作。

模式	说明
LPRUN	正常显示
Sleep/DeepSleep	正常显示

表 30-3 LCD 与低功耗模式

30.6 寄存器

LCD模块基地址：0x40010C00

offset 地址	名称	符号
0x00	显示控制寄存器（LCD Control Register）	LCD_CR
0x04	显示测试控制寄存器（LCD test Register）	LCD_TEST
0x08	显示频率控制寄存器（LCD Frequency Control Register）	LCD_FCR
0x0C	闪烁时间寄存器（LCD Flick Time Register）	LCD_FLKT
0x14	显示中断使能寄存器（LCD Interrupt Enable Register）	LCD_IER
0x18	显示中断标志寄存器（LCD Interrupt Status Register）	LCD_ISR
0x24	显示数据缓存寄存器（LCD data buffer registers ）	LCD_DATA0
0x28		LCD_DATA1
0x2C		LCD_DATA2
0x30		LCD_DATA3
0x34		LCD_DATA4
0x38		LCD_DATA5
0x3C		LCD_DATA6
0x40		LCD_DATA7
0x44		LCD_DATA8
0x48		LCD_DATA9
0x4C		LCD_DATA10
0x50	COM 使能控制寄存器（LCD COM Enable Register）	LCD_COMEN
0x54	SEG 使能控制寄存器 0（LCD SEG Enable Register0）	LCD_SEGEN0
0x58	SEG 使能控制寄存器 1（LCD SEG Enable Register1）	LCD_SEGEN1

30.6.1 显示控制寄存器（LCD_CR）

名称	LCD_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						IC_CTRL	
位权限	U-0						R/W-01	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	ENMODE	FLICK	LED_SEGPN	LED_DIGPN	LCDBIAS			
位权限	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1110			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA_MODE	LED_EN	BIASMD	ANTIPOLAR	WFT	LMUX		LCDEN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

Bit	助记符	功能描述
31:18	--	RFU：未实现，读为0
17:16	**IC_CTRL**	**偏置电路输入电流源大小控制** 00 = 电流最大 01 = 电流次大 10 = 电流次小 11 = 电流最小
15	**ENMODE**	**驱动模式选择** 0= RFU 1 = 片内电阻型驱动 注意：只有片内电阻模式需要写 1
14	**FLICK**	显示闪烁使能位 1：显示闪烁，闪烁频率由 TON 和 TOFF 寄存器设置 0：关闭闪烁
13	**LED_SEGPN**	LED_SEG 输出取反 0：不取反 1：取反
12	**LED_DIGPN**	LED_DIG 输出取反 0：不取反 1：取反
11:8	**LCDBIAS**	LCD 偏置电平选择位，用于显示灰度控制
7	**DATA_MODE**	显示数据模式，详见 30.6.7 章节 0：模式 0 1：模式 1
6	**LED_EN**	LED 显示使能位（请确保和 LCDEN 独立使能） 1：启动 LED 显示 0：关闭 LED 显示
5	**BIASMD**	偏置类型选择 1：1/3 Bias 0：1/4 Bias
4	**ANTIPOLAR**	防极化使能 1：COM 和 SEG 在 LCD 关闭情况下接地 0：COM 和 SEG 在 LCD 关闭情况下浮空
3	**WFT**	驱动波形选择 1：B类波形 0：A类波形
2:1	**LMUX**	COM 数量选择 00：4COM 01：6COM 10/11：8COM
0	**LCDEN**	LCD 显示使能位 1：启动 LCD 显示

Bit	助记符	功能描述
		0：关闭 LCD 显示

LCDBIAS[3:0]	不同 VDD 下内部偏置电压(V)			
	5	4.5	3.6	3.0
0000	2.74	2.47	1.97	1.64
0001	2.83	2.54	2.03	1.69
0010	2.92	2.62	2.10	1.75
0011	3.01	2.71	2.17	1.81
0100	3.12	2.80	2.24	1.87
0101	3.23	2.90	2.32	1.94
0110	3.35	3.01	2.41	2.01
0111	3.47	3.13	2.50	2.08
1000	3.61	3.25	2.60	2.17
1001	3.76	3.39	2.71	2.26
1010	3.93	3.53	2.83	2.35
1011	4.10	3.69	2.95	2.46
1100	4.30	3.87	3.09	2.58
1101	4.51	4.06	3.25	2.71
1110	4.75	4.27	3.42	2.85
1111	5.00	4.50	3.60	3.00

30.6.2 显示测试控制寄存器（LCD_TEST）

名称	LCD_TEST							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LCCTRL	-						TESTEN
位权限	R/W-0	U-0						R/W-0

位号	位名	说明
31:8	---	未实现，读为0
7	LCCTRL	LCD测试控制位，仅在测试模式下有效 COM、SEG 输出电平由测试模式下的引脚输出数据寄存器决定。不同设置下 SEG 或 COM 输出的结果参见后文表格。
6:0	---	未实现，读为0

位号	位名	说明
0	**TESTEN**	测试模式使能位 1 = LCD 测试模式使能。在 LCD 测试模式下，LCD 引脚静态输出模拟直流电平，所有与动态扫描时间以及扫描波形相关寄存器设置无效 0= 正常工作模式，测试模式无效，相关测试寄存器控制无效

测试模式下引脚输出电平：

LCCTRL	DISPDATA	COM 引脚输出电平	SEG 引脚输出电平
		1/3bias	1/3bias
0	0	V3	V2
0	1	V1	V4
1	0	V2	V3
1	1	V4	V1

30.6.3 显示频率控制寄存器（LCD_FCR）

名称	LCD_FCR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DF[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:8	**---**	未实现，读为0
7:0	**DF**	显示预分频寄存器

30.6.4 闪烁时间寄存器（LCD_FLCKT）

名称	LCD_FLCKT							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	TOFF[7:0]							
位权限	R/W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TON[7:0]							
位权限	R/W-00000000							

位号	位名	说明
31:16	---	**未实现，读为0**
15:8	**TOFF**	**闪烁显示熄灭时间寄存器** 闪烁显示时的熄灭时间=TOFF * T_{step}秒，最小步长T_{step} = COM*DF[7:0]*2*16/32768Hz
7:0	**TON**	**闪烁显示点亮时间寄存器** 闪烁显示时的点亮时间=TON * T_{step}秒，最小步长T_{step} = COM*DF[7:0]*2*16/32768Hz

30.6.5 显示中断使能寄存器（LCD_IER）

名称	LCD_IER							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---	---	---	---	---	---	DONIE	DOFFIE
位权限	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

位号	位名	说明
31:2	---	**未实现，读为0**
Bit1	**DONIE**	**显示点亮中断使能位** 1 = 显示点亮中断使能 0 = 显示点亮中断禁止
Bit0	**DOFFIE**	**显示熄灭中断使能位** 1 = 显示熄灭中断使能 0 = 显示熄灭中断禁止

30.6.6 显示中断标志寄存器（LCD_ISR）

名称	LCD_ISR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---	---	---	---	---	---	DONIF	DOFFIF
位权限	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

位号	位名	说明
31:2	**---**	**未实现，读为0**
1	**DONIF**	**显示点亮中断标志** 显示由灭变亮时硬件产生中断标志，硬件置位，软件清零
0	**DOFFIF**	**显示熄灭中断标志** 显示由亮变灭时硬件产生中断标志，硬件置位，软件清零

30.6.7 显示数据寄存器（LCD_DATAx）

LCD 显示模块内有 11 个 32 bit 的显示数据寄存器。均为可读可写，复位值为 0。

名称	LCD_DATAx(x=0~11)							
地址	0x24~0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DSDA31	DSDA30	DSDA29	DSDA28	DSDA27	DSDA26	DSDA25	DSDA24
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DSDA23	DSDA22	DSDA21	DSDA20	DSDA19	DSDA18	DSDA17	DSDA16
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DSDA15	DSDA14	DSDA13	DSDA12	DSDA11	DSDA10	DSDA9	DSDA8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DSDA7	DSDA6	DSDA5	DSDA4	DSDA3	DSDA2	DSDA1	DSDA0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:0	DSDAx	LCD 显示数据

在 LCD 的 4/6/8COM 模式，显示数据寄存器的内容定义有所不同。详见以下各节描述。

LED 的 4/6/8COM 模式，DISPDATA0~DISPDATA7 存放 LED_SEG0~LED_SEG7 的 LED_DIG。

若 DATA_MODE 为 0，则 DISPDATA0~DISPDATA7 存放方式为横向。

若 DATA_MODE 为 1，则 DISPDATA0~DISPDATA7 存放方式为纵向。

30.6.7.1 4COM 显示数据寄存器（DATA_MODE=0）

名称	4com 显示数据寄存器							
offset	0x24 ~ 0x38							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	4com 显示数据寄存器							
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

名称	4com 显示数据寄存器							
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

30.6.7.2 6COM 显示数据寄存器（DATA_MODE=0）

名称	6com 显示数据寄存器							
offset	0x24 ~ 0x40							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG43 COM0	SEG42 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG43 COM1	SEG42 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2

名称	6com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG43 COM2	SEG42 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG43 COM3	SEG42 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM4	SEG30 COM4	SEG43 COM4	SEG42 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5

名称	6com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM5	SEG30 COM5	SEG43 COM5	SEG42 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
DISPDATA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG33 COM3	SEG32 COM3	SEG41 COM2	SEG40 COM2	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2
DISPDATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5	SEG41 COM4	SEG40 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG41 COM5	SEG40 COM5	SEG39 COM5	SEG38 COM5

30.6.7.3 8COM 显示数据寄存器（DATA_MODE=0）

名称	8com 显示数据寄存器							
offset	0x24 ~ 0x48							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	8com 显示数据寄存器							
	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM4	SEG42 COM4	SEG41 COM4	SEG40 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	8com 显示数据寄存器							
	SEG43 COM5	SEG42 COM5	SEG41 COM5	SEG40 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
DISPDATA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM6	SEG6 COM6	SEG5 COM6	SEG4 COM6	SEG3 COM6	SEG2 COM6	SEG1 COM6	SEG0 COM6
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM6	SEG14 COM6	SEG13 COM6	SEG12 COM6	SEG11 COM6	SEG10 COM6	SEG9 COM6	SEG8 COM6
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM6	SEG22 COM6	SEG21 COM6	SEG20 COM6	SEG19 COM6	SEG18 COM6	SEG17 COM6	SEG16 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM6	SEG42 COM6	SEG41 COM6	SEG40 COM6	SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6
DISPDATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM7	SEG6 COM7	SEG5 COM7	SEG4 COM7	SEG3 COM7	SEG2 COM7	SEG1 COM7	SEG0 COM7
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM7	SEG14 COM7	SEG13 COM7	SEG12 COM7	SEG11 COM7	SEG10 COM7	SEG9 COM7	SEG8 COM7
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM7	SEG22 COM7	SEG21 COM7	SEG20 COM7	SEG19 COM7	SEG18 COM7	SEG17 COM7	SEG16 COM7
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM7	SEG42 COM7	SEG41 COM7	SEG40 COM7	SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7
DISPDATA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	8com 显示数据寄存器							
	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3
DISPDATA9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM5	SEG38 COM5	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM6	SEG38 COM6	SEG37 COM6	SEG36 COM6	SEG35 COM6	SEG34 COM6	SEG33 COM6	SEG32 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM7	SEG38 COM7	SEG37 COM7	SEG36 COM7	SEG35 COM7	SEG34 COM7	SEG33 COM7	SEG32 COM7

30.6.7.4 4COM 显示数据寄存器（DATA_MODE=1）

名称	4com 显示数据寄存器							
offset	0x24 ~ 0x4C							
DISPDATA0					Bit3	Bit2	Bit1	Bit0
					SEG0 COM3	SEG0 COM2	SEG0 COM1	SEG0 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG1 COM3	SEG1 COM2	SEG1 COM1	SEG1 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG2 COM3	SEG2 COM2	SEG2 COM1	SEG2 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG3 COM3	SEG3 COM2	SEG3 COM1	SEG3 COM0
DISPDATA1					Bit3	Bit2	Bit1	Bit0
					SEG4 COM3	SEG4 COM2	SEG4 COM1	SEG4 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG5 COM3	SEG5 COM2	SEG5 COM1	SEG5 COM0

名称	4com 显示数据寄存器							
					Bit19	Bit18	Bit17	Bit16
					SEG6 COM3	SEG6 COM2	SEG6 COM1	SEG6 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG7 COM3	SEG7 COM2	SEG7 COM1	SEG7 COM0
DISPDATA2					Bit3	Bit2	Bit1	Bit0
					SEG8 COM3	SEG8 COM2	SEG8 COM1	SEG8 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG9 COM3	SEG9 COM2	SEG9 COM1	SEG9 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG10 COM3	SEG10 COM2	SEG10 COM1	SEG10 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG11 COM3	SEG11 COM2	SEG11 COM1	SEG11 COM0
DISPDATA3					Bit3	Bit2	Bit1	Bit0
					SEG12 COM3	SEG12 COM2	SEG12 COM1	SEG12 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG13 COM3	SEG13 COM2	SEG13 COM1	SEG13 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG14 COM3	SEG14 COM2	SEG14 COM1	SEG14 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG15 COM3	SEG15 COM2	SEG15 COM1	SEG15 COM0
DISPDATA4					Bit3	Bit2	Bit1	Bit0
					SEG16 COM3	SEG16 COM2	SEG16 COM1	SEG16 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG17 COM3	SEG17 COM2	SEG17 COM1	SEG17 COM0

名称	4com 显示数据寄存器							
					Bit19	Bit18	Bit17	Bit16
					SEG18 COM3	SEG18 COM2	SEG18 COM1	SEG18 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG19 COM3	SEG19 COM2	SEG19 COM1	SEG19 COM0
DISPDATA5					Bit3	Bit2	Bit1	Bit0
					SEG20 COM3	SEG20 COM2	SEG20 COM1	SEG20 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG21 COM3	SEG21 COM2	SEG21 COM1	SEG21 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG22 COM3	SEG22 COM2	SEG22 COM1	SEG22 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG23 COM3	SEG23 COM2	SEG23 COM1	SEG23 COM0
DISPDATA6					Bit3	Bit2	Bit1	Bit0
					SEG24 COM3	SEG24 COM2	SEG24 COM1	SEG24 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG25 COM3	SEG25 COM2	SEG25 COM1	SEG25 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG26 COM3	SEG26 COM2	SEG26 COM1	SEG26 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG27 COM3	SEG27 COM2	SEG27 COM1	SEG27 COM0
DISPDATA7					Bit3	Bit2	Bit1	Bit0
					SEG28 COM3	SEG28 COM2	SEG28 COM1	SEG28 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG29 COM3	SEG29 COM2	SEG29 COM1	SEG29 COM0

名称	4com 显示数据寄存器							
					Bit19	Bit18	Bit17	Bit16
					SEG30 COM3	SEG30 COM2	SEG30 COM1	SEG30 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG31 COM3	SEG31 COM2	SEG31 COM1	SEG31 COM0
DISPDATA8					Bit3	Bit2	Bit1	Bit0
					SEG32 COM3	SEG32 COM2	SEG32 COM1	SEG32 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG33 COM3	SEG33 COM2	SEG33 COM1	SEG33 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG34 COM3	SEG34 COM2	SEG34 COM1	SEG34 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG35 COM3	SEG35 COM2	SEG35 COM1	SEG35 COM0
DISPDATA9					Bit3	Bit2	Bit1	Bit0
					SEG36 COM3	SEG36 COM2	SEG36 COM1	SEG36 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG37 COM3	SEG37 COM2	SEG37 COM1	SEG37 COM0
					Bit19	Bit18	Bit17	Bit16
					SEG38 COM3	SEG38 COM2	SEG38 COM1	SEG38 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG39 COM3	SEG39 COM2	SEG39 COM1	SEG39 COM0
DISPDATA10					Bit3	Bit2	Bit1	Bit0
					SEG40 COM3	SEG40 COM2	SEG40 COM1	SEG40 COM0
					Bit11	Bit10	Bit9	Bit8
					SEG41 COM3	SEG41 COM2	SEG41 COM1	SEG41 COM0

名称	4com 显示数据寄存器							
					Bit19	Bit18	Bit17	Bit16
					SEG42 COM3	SEG42 COM2	SEG42 COM1	SEG42 COM0
					Bit27	Bit26	Bit25	Bit24
					SEG43 COM3	SEG43 COM2	SEG43 COM1	SEG43 COM0

30.6.7.5 6COM 显示数据寄存器（DATA_MODE=1）

名称	6com 显示数据寄存器							
offset	0x24 ~ 0x4C							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG0 COM5	SEG0 COM4	SEG0 COM3	SEG0 COM2	SEG0 COM1	SEG0 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG1 COM5	SEG1 COM4	SEG1 COM3	SEG1 COM2	SEG1 COM1	SEG1 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG2 COM5	SEG2 COM4	SEG2 COM3	SEG2 COM2	SEG2 COM1	SEG2 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG3 COM5	SEG3 COM4	SEG3 COM3	SEG3 COM2	SEG3 COM1	SEG3 COM0
DISPDATA1			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG4 COM5	SEG4 COM4	SEG4 COM3	SEG4 COM2	SEG4 COM1	SEG4 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG5 COM5	SEG5 COM4	SEG5 COM3	SEG5 COM2	SEG5 COM1	SEG5 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG6 COM5	SEG6 COM4	SEG6 COM3	SEG6 COM2	SEG6 COM1	SEG6 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG7 COM5	SEG7 COM4	SEG7 COM3	SEG7 COM2	SEG7 COM1	SEG7 COM0
DISPDATA2			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	6com 显示数据寄存器							
			SEG8 COM5	SEG8 COM4	SEG8 COM3	SEG8 COM2	SEG8 COM1	SEG8 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG9 COM5	SEG9 COM4	SEG9 COM3	SEG9 COM2	SEG9 COM1	SEG9 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG10 COM5	SEG10 COM4	SEG10 COM3	SEG10 COM2	SEG10 COM1	SEG10 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG11 COM5	SEG11 COM4	SEG11 COM3	SEG11 COM2	SEG11 COM1	SEG11 COM0
DISPDATA3			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG12 COM5	SEG12 COM4	SEG12 COM3	SEG12 COM2	SEG12 COM1	SEG12 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG13 COM5	SEG13 COM4	SEG13 COM3	SEG13 COM2	SEG13 COM1	SEG13 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG14 COM5	SEG14 COM4	SEG14 COM3	SEG14 COM2	SEG14 COM1	SEG14 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG15 COM5	SEG15 COM4	SEG15 COM3	SEG15 COM2	SEG15 COM1	SEG15 COM0
DISPDATA4			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG16 COM5	SEG16 COM4	SEG16 COM3	SEG16 COM2	SEG16 COM1	SEG16 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG17 COM5	SEG17 COM4	SEG17 COM3	SEG17 COM2	SEG17 COM1	SEG17 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG18 COM5	SEG18 COM4	SEG18 COM3	SEG18 COM2	SEG18 COM1	SEG18 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG19 COM5	SEG19 COM4	SEG19 COM3	SEG19 COM2	SEG19 COM1	SEG19 COM0
DISPDATA5			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	6com 显示数据寄存器							
			SEG20 COM5	SEG20 COM4	SEG20 COM3	SEG20 COM2	SEG20 COM1	SEG20 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG21 COM5	SEG21 COM4	SEG21 COM3	SEG21 COM2	SEG21 COM1	SEG21 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG22 COM5	SEG22 COM4	SEG22 COM3	SEG22 COM2	SEG22 COM1	SEG22 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG23 COM5	SEG23 COM4	SEG23 COM3	SEG23 COM2	SEG23 COM1	SEG23 COM0
DISPDATA6			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG24 COM5	SEG24 COM4	SEG24 COM3	SEG24 COM2	SEG24 COM1	SEG24 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG25 COM5	SEG25 COM4	SEG25 COM3	SEG25 COM2	SEG25 COM1	SEG25 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG26 COM5	SEG26 COM4	SEG26 COM3	SEG26 COM2	SEG26 COM1	SEG26 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG27 COM5	SEG27 COM4	SEG27 COM3	SEG27 COM2	SEG27 COM1	SEG27 COM0
DISPDATA7			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG28 COM5	SEG28 COM4	SEG28 COM3	SEG28 COM2	SEG28 COM1	SEG28 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG29 COM5	SEG29 COM4	SEG29 COM3	SEG29 COM2	SEG29 COM1	SEG29 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG30 COM5	SEG30 COM4	SEG30 COM3	SEG30 COM2	SEG30 COM1	SEG30 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG31 COM5	SEG31 COM4	SEG31 COM3	SEG31 COM2	SEG31 COM1	SEG31 COM0
DISPDATA8			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	6com 显示数据寄存器							
			SEG32 COM5	SEG32 COM4	SEG32 COM3	SEG32 COM2	SEG32 COM1	SEG32 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG33 COM5	SEG33 COM4	SEG33 COM3	SEG33 COM2	SEG33 COM1	SEG33 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG34 COM5	SEG34 COM4	SEG34 COM3	SEG34 COM2	SEG34 COM1	SEG34 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG35 COM5	SEG35 COM4	SEG35 COM3	SEG35 COM2	SEG35 COM1	SEG35 COM0
DISPDATA9			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG36 COM5	SEG36 COM4	SEG36 COM3	SEG36 COM2	SEG36 COM1	SEG36 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG37 COM5	SEG37 COM4	SEG37 COM3	SEG37 COM2	SEG37 COM1	SEG37 COM0
			Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
			SEG38 COM5	SEG38 COM4	SEG38 COM3	SEG38 COM2	SEG38 COM1	SEG38 COM0
			Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
			SEG39 COM5	SEG39 COM4	SEG39 COM3	SEG39 COM2	SEG39 COM1	SEG39 COM0
DISPDATA10			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
			SEG40 COM5	SEG40 COM4	SEG40 COM3	SEG40 COM2	SEG40 COM1	SEG40 COM0
			Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
			SEG41 COM5	SEG41 COM4	SEG41 COM3	SEG41 COM2	SEG41 COM1	SEG41 COM0

30.6.7.6 8COM 显示数据寄存器（DATA_MODE=1）

名称	8com 显示数据寄存器							
offset	0x24 ~ 0x4C							
DISPDATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG0 COM7	SEG0 COM6	SEG0 COM5	SEG0 COM4	SEG0 COM3	SEG0 COM2	SEG0 COM1	SEG0 COM0

名称	8com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG1 COM7	SEG1 COM6	SEG1 COM5	SEG1 COM4	SEG1 COM3	SEG1 COM2	SEG1 COM1	SEG1 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG2 COM7	SEG2 COM6	SEG2 COM5	SEG2 COM4	SEG2 COM3	SEG2 COM2	SEG2 COM1	SEG2 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG3 COM7	SEG3 COM6	SEG3 COM5	SEG3 COM4	SEG3 COM3	SEG3 COM2	SEG3 COM1	SEG3 COM0
DISPDATA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG4 COM7	SEG4 COM6	SEG4 COM5	SEG4 COM4	SEG4 COM3	SEG4 COM2	SEG4 COM1	SEG4 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG5 COM7	SEG5 COM6	SEG5 COM5	SEG5 COM4	SEG5 COM3	SEG5 COM2	SEG5 COM1	SEG5 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG6 COM7	SEG6 COM6	SEG6 COM5	SEG6 COM4	SEG6 COM3	SEG6 COM2	SEG6 COM1	SEG6 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG7 COM7	SEG7 COM6	SEG7 COM5	SEG7 COM4	SEG7 COM3	SEG7 COM2	SEG7 COM1	SEG7 COM0
DISPDATA2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG8 COM7	SEG8 COM6	SEG8 COM5	SEG8 COM4	SEG8 COM3	SEG8 COM2	SEG8 COM1	SEG8 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG9 COM7	SEG9 COM6	SEG9 COM5	SEG9 COM4	SEG9 COM3	SEG9 COM2	SEG9 COM1	SEG9 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG10 COM7	SEG10 COM6	SEG10 COM5	SEG10 COM4	SEG10 COM3	SEG10 COM2	SEG10 COM1	SEG10 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG11 COM7	SEG11 COM6	SEG11 COM5	SEG11 COM4	SEG11 COM3	SEG11 COM2	SEG11 COM1	SEG11 COM0
DISPDATA3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG12 COM7	SEG12 COM6	SEG12 COM5	SEG12 COM4	SEG12 COM3	SEG12 COM2	SEG12 COM1	SEG12 COM0

名称	8com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG13 COM7	SEG13 COM6	SEG13 COM5	SEG13 COM4	SEG13 COM3	SEG13 COM2	SEG13 COM1	SEG13 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG14 COM7	SEG14 COM6	SEG14 COM5	SEG14 COM4	SEG14 COM3	SEG14 COM2	SEG14 COM1	SEG14 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG15 COM7	SEG15 COM6	SEG15 COM5	SEG15 COM4	SEG15 COM3	SEG15 COM2	SEG15 COM1	SEG15 COM0
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG16 COM7	SEG16 COM6	SEG16 COM5	SEG16 COM4	SEG16 COM3	SEG16 COM2	SEG16 COM1	SEG16 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG17 COM7	SEG17 COM6	SEG17 COM5	SEG17 COM4	SEG17 COM3	SEG17 COM2	SEG17 COM1	SEG17 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG18 COM7	SEG18 COM6	SEG18 COM5	SEG18 COM4	SEG18 COM3	SEG18 COM2	SEG18 COM1	SEG18 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG19 COM7	SEG19 COM6	SEG19 COM5	SEG19 COM4	SEG19 COM3	SEG19 COM2	SEG19 COM1	SEG19 COM0
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG20 COM7	SEG20 COM6	SEG20 COM5	SEG20 COM4	SEG20 COM3	SEG20 COM2	SEG20 COM1	SEG20 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG21 COM7	SEG21 COM6	SEG21 COM5	SEG21 COM4	SEG21 COM3	SEG21 COM2	SEG21 COM1	SEG21 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG22 COM7	SEG22 COM6	SEG22 COM5	SEG22 COM4	SEG22 COM3	SEG22 COM2	SEG22 COM1	SEG22 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG23 COM7	SEG23 COM6	SEG23 COM5	SEG23 COM4	SEG23 COM3	SEG23 COM2	SEG23 COM1	SEG23 COM0
DISPDATA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG24 COM7	SEG24 COM6	SEG24 COM5	SEG24 COM4	SEG24 COM3	SEG24 COM2	SEG24 COM1	SEG24 COM0

名称	8com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG25 COM7	SEG25 COM6	SEG25 COM5	SEG25 COM4	SEG25 COM3	SEG25 COM2	SEG25 COM1	SEG25 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG26 COM7	SEG26 COM6	SEG26 COM5	SEG26 COM4	SEG26 COM3	SEG26 COM2	SEG26 COM1	SEG26 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG27 COM7	SEG27 COM6	SEG27 COM5	SEG27 COM4	SEG27 COM3	SEG27 COM2	SEG27 COM1	SEG27 COM0
DISPDATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG28 COM7	SEG28 COM6	SEG28 COM5	SEG28 COM4	SEG28 COM3	SEG28 COM2	SEG28 COM1	SEG28 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG29 COM7	SEG29 COM6	SEG29 COM5	SEG29 COM4	SEG29 COM3	SEG29 COM2	SEG29 COM1	SEG29 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG30 COM7	SEG30 COM6	SEG30 COM5	SEG30 COM4	SEG30 COM3	SEG30 COM2	SEG30 COM1	SEG30 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM7	SEG31 COM6	SEG31 COM5	SEG31 COM4	SEG31 COM3	SEG31 COM2	SEG31 COM1	SEG31 COM0
DISPDATA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG32 COM7	SEG32 COM6	SEG32 COM5	SEG32 COM4	SEG32 COM3	SEG32 COM2	SEG32 COM1	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG33 COM7	SEG33 COM6	SEG33 COM5	SEG33 COM4	SEG33 COM3	SEG33 COM2	SEG33 COM1	SEG33 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG34 COM7	SEG34 COM6	SEG34 COM5	SEG34 COM4	SEG34 COM3	SEG34 COM2	SEG34 COM1	SEG34 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG35 COM7	SEG35 COM6	SEG35 COM5	SEG35 COM4	SEG35 COM3	SEG35 COM2	SEG35 COM1	SEG35 COM0
DISPDATA9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG36 COM7	SEG36 COM6	SEG36 COM5	SEG36 COM4	SEG36 COM3	SEG36 COM2	SEG36 COM1	SEG36 COM0

名称	8com 显示数据寄存器							
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG37 COM7	SEG37 COM6	SEG37 COM5	SEG37 COM4	SEG37 COM3	SEG37 COM2	SEG37 COM1	SEG37 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG38 COM7	SEG38 COM6	SEG38 COM5	SEG38 COM4	SEG38 COM3	SEG38 COM2	SEG38 COM1	SEG38 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM7	SEG39 COM6	SEG39 COM5	SEG39 COM4	SEG39 COM3	SEG39 COM2	SEG39 COM1	SEG39 COM0

30.6.8 LCD COM 使能控制寄存器(LCD_COMEN)

名称	LCD_COMEN							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	-	-	COMEN 3	COMEN 2	COMEN 1	COMEN 0
位权限	R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	COMEN	LCD COM(LED SEG)输出使能控制 1: COM 输出使能 0: COM 输出禁止

30.6.9 LCD SEG 使能控制寄存器 0(LCD_SEGEN0)

名称	LCD_SEGEN0							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SEGEN31	SEGEN30	SEGEN29	SEGEN28	SEGEN27	SEGEN26	SEGEN25	SEGEN24
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:0	SEGENx	LCD SEG(LED DIG)输出使能控制 1: SEG 输出使能 0: SEG 输出禁止

30.6.10 LCD SEG 使能控制寄存器 1(LCD_SEGEN1)

名称	LCD_SEGEN1							
offset	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	-	-	-	SEGEN43	SEGEN42	SEGEN41	SEGEN40
位权限	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGEN39	SEGEN38	SEGEN37	SEGEN36	SEGEN35	SEGEN34	SEGEN33	SEGEN32
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:0	SEGENx	LCD SEG(LED DIG)输出使能控制 1: SEG 输出使能 0: SEG 输出禁止

31 模数转换器（ADC）

31.1 概述

FM33FR0 带有 2Msps 12bit SAR-ADC，可实现温度、电池电压或其他直流信号的测量功能。主要特点为：

- 工作电压 1.6～5.5V
- 输入信号幅度 0~VREF+
- 可灵活选择基准源
- 最高采样率 2Msps（F_{ADC}=32MHz）
- 最大 23 个外部输入通道：
  - 14 个快速通道，可以组成 7 个差分输入对
  - 9 个慢速通道，仅支持单端输入
  - 部分慢速通道内建缓冲器，可用于弱驱动信号测量
- 3 个内部采样通道
  - 温度传感器
  - 内部基准测量
  - 升压后的内部基准测量
- 可配置的采样保持时间
- 支持单次转换和连续转换
- 支持 DMA
- 支持过采样硬件平均，最高 16bit 输出（256 次平均）
- 超低功耗架构

31.2 结构框图

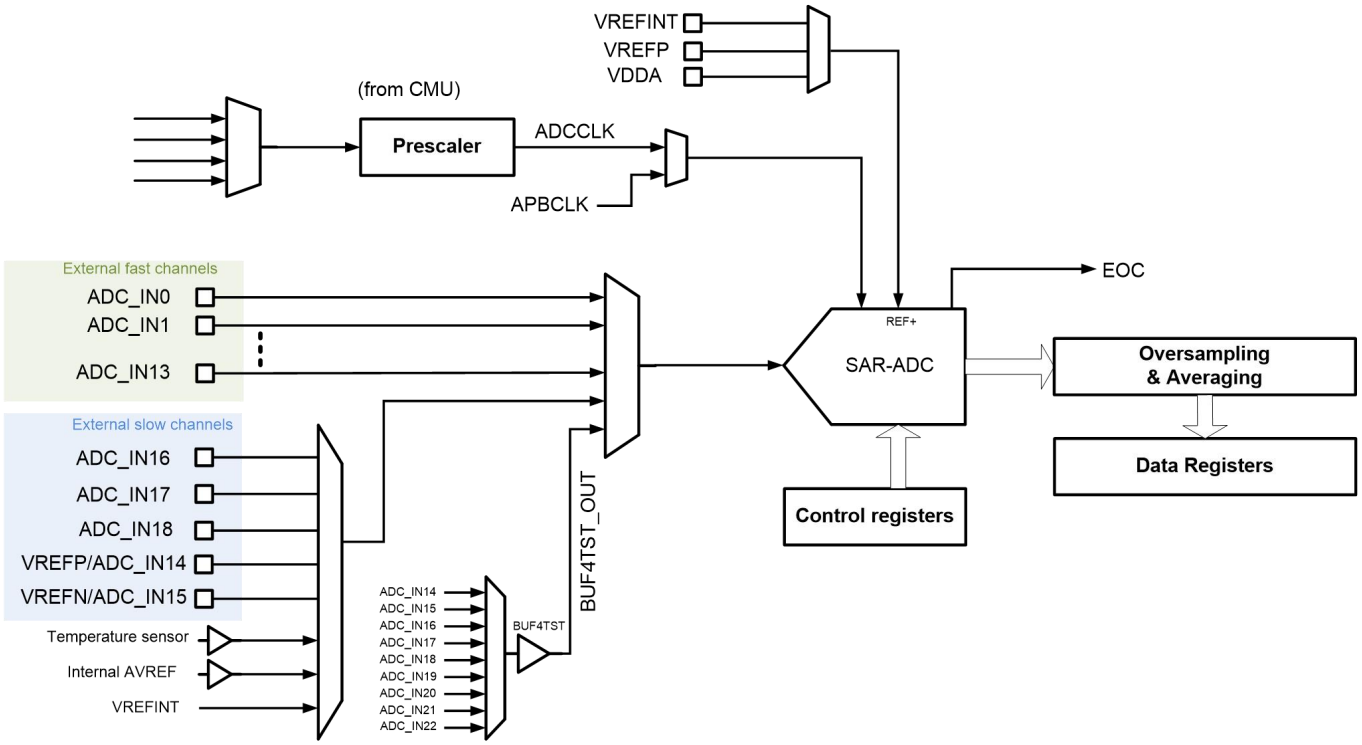


图 31-1 ADC 结构框图

31.3 输入通道

ADC 最大支持 4 个内部通道和 19 个外部通道。

通道	IO	说明	
ADC_IN0	PC15	外部快速通道，单端或差分输入 其中差分对组合关系如下： ADC_IN0 – ADC_IN7 ADC_IN1 – ADC_IN8 ADC_IN2 – ADC_IN9 ADC_IN3 – ADC_IN10 ADC_IN4 – ADC_IN11 ADC_IN5 – ADC_IN12 ADC_IN6 – ADC_IN13	
ADC_IN1	PD1		
ADC_IN2	PD3		
ADC_IN3	PD5		
ADC_IN4	PA13		
ADC_IN5	PA11		
ADC_IN6	PC11		
ADC_IN7	PD0		
ADC_IN8	PD2		
ADC_IN9	PD4		
ADC_IN10	PD6		
ADC_IN11	PA14		
ADC_IN12	PA12		
ADC_IN13	PC12		
ADC_IN14	PE10/VREFP	外部慢速通道，单端输入，可以通过模拟 buffer 缓冲，缓冲时使用 BUF4TST_OUT 通道 ADC_IN19~22 只能使用 BUF4TST_OUT 通道输出 ***注意，当使用 ADC_IN14 和 ADC_IN15 通道时，必须将 ADC 基准源选择为 VDDA 或者 VREFINT***	
ADC_IN15	PE9/VREFN		
ADC_IN16	PD7		
ADC_IN17	PC13		
ADC_IN18	PC14		
ADC_IN19	PF0		
ADC_IN20	PF1		
ADC_IN21	PF2		
ADC_IN22	PF3		
BUF4TST_OUT	N/A	内部测试 buffer 输出	内部慢速通道
TS		温度传感器采样通道	
AVREF		内部 1.0V 基准源采样通道	
VREFINT		内部产生的基准电压	

表 31-1 ADC 输入通道

31.4 单端和差分输入

ADC 支持单端输入和差分输入模式。

单端模式下 ADC 转换的是单个输入引脚对地的电压值，输入幅度范围是 0~VREF+(VDDA)。为了避免可能的波形削顶风险导致输入失真，一般建议输入信号最大幅值不超过 0.95*VREF+。

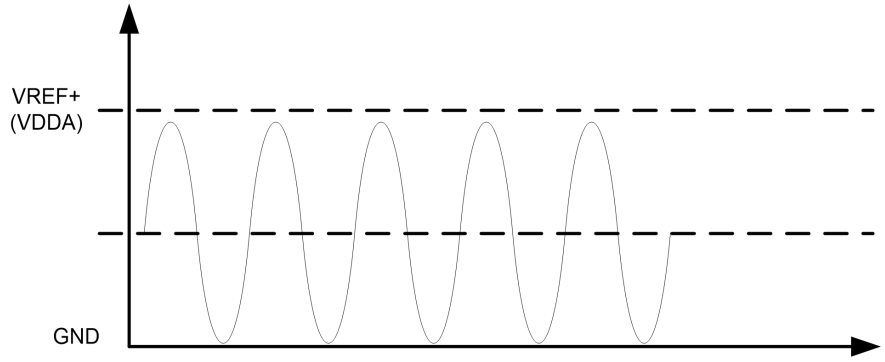


图 31-2 单端输入

差分输入模式下，ADC 转换的是差分输入引脚对 VIN+和 VIN-之间的差值，输入信号为(VIN+) – (VIN-)，输入范围是-VREF~+VREF。使用差分输入方式，可以获得更好的共模噪声抑制效果，因此当被采样信号源远离 ADC 输入引脚时，推荐使用差分对方式提高信噪比。为达到最佳性能，应尽量设置差分输入信号的共模电平为 VREF/2。

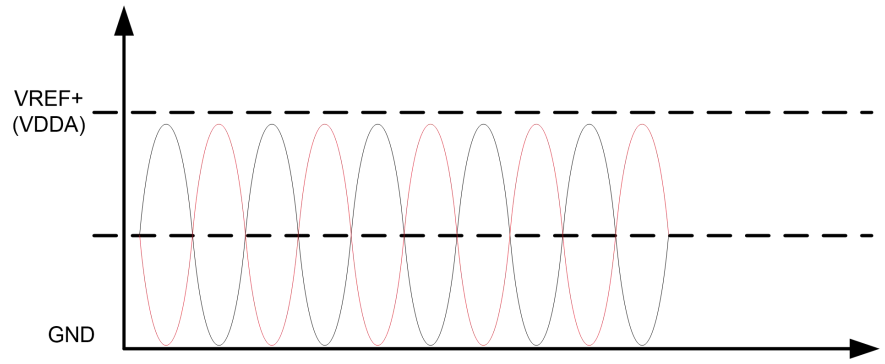


图 31-3 差分输入

差分模式码字与输入信号电平的对应关系：

$$\left(ADC\text{转换值-}2048\right)\times 2LSB=\frac{VP-VN}{VREF}\times 4096LSB$$

码值与差分输入信号的关系可以用下图表示：

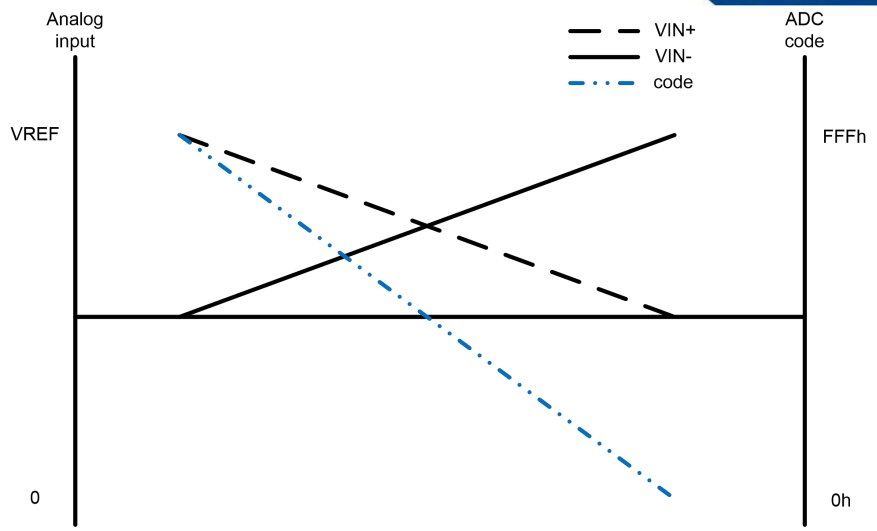


图 31-4 差分输入信号与码字的关系

自动转换多个通道时，可以实现差分通道和单端通道的交织。举个简单的例子，如果希望连续采样 ADC_IN0 和 ADC_IN7 组成的差分通道，以及 ADC_IN2 的单端通道，则软件应采取如下配置方法：

- 配置 ADC_CHER 寄存器，ECH0, ECH7, ECH2 置 1
- 配置 ADC_DCR 寄存器，AINS[0]置 1
- 选择单次转换或连续转换
- 启动 ADC 开始转换

ADC 启动后，ADC_IN0 和 ADC_IN7 通道同时使能，ADC 对差分输入信号进行采样转换；完成后关闭 ADC_IN0 和 ADC_IN7 通道，使能 ADC_IN2 通道，ADC 对单端输入信号进行采样转换。

注意：差分对工作时仅需使能低位编号的通道即可，比如上面的例子中差分对 IN0 和 IN7 仅需使能 ADC_IN0 通道，并置位 AINS[0]

31.5 工作时序

ADC 有 offset 校准和正常工作两种时序。

上电后建议先进行一次校准，以获得更好的性能。校准后无需重新校准，除非芯片发生全局复位，或者电源电压、温度发生较大变化。Offset 校准通过软件置位 CALEN 启动，校准过程包含多个采样转换周期，周期数由 OSCAL_CYCLE 寄存器配置。校准完成后置位 EOCAL 标志寄存器。

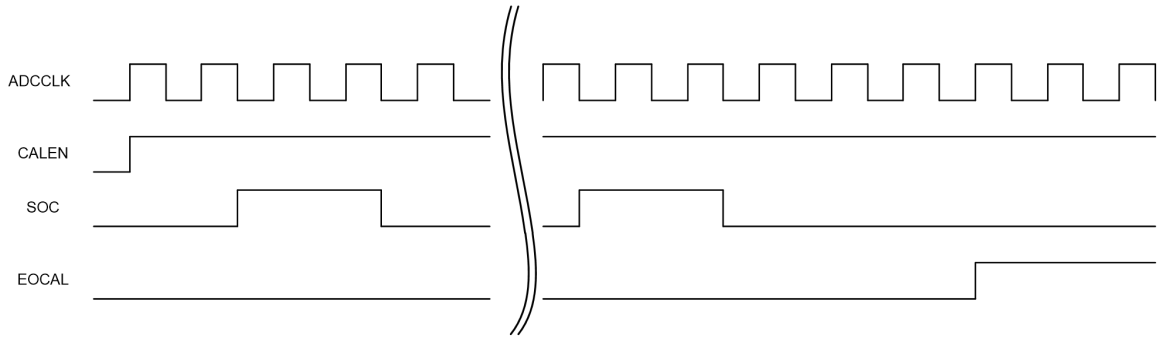


图 31-5 ADC 校准时序

采样转换时，通过 SOC 信号启动 ADC 采样，SOC 高电平宽度控制了 ADC 采样时间，SOC 变低后启动转换，转换周期为 14 个 ADC_CLK 周期，采样时间可配置，最短 2 个 ADC_CLK，最长 512 个 ADC_CLK。

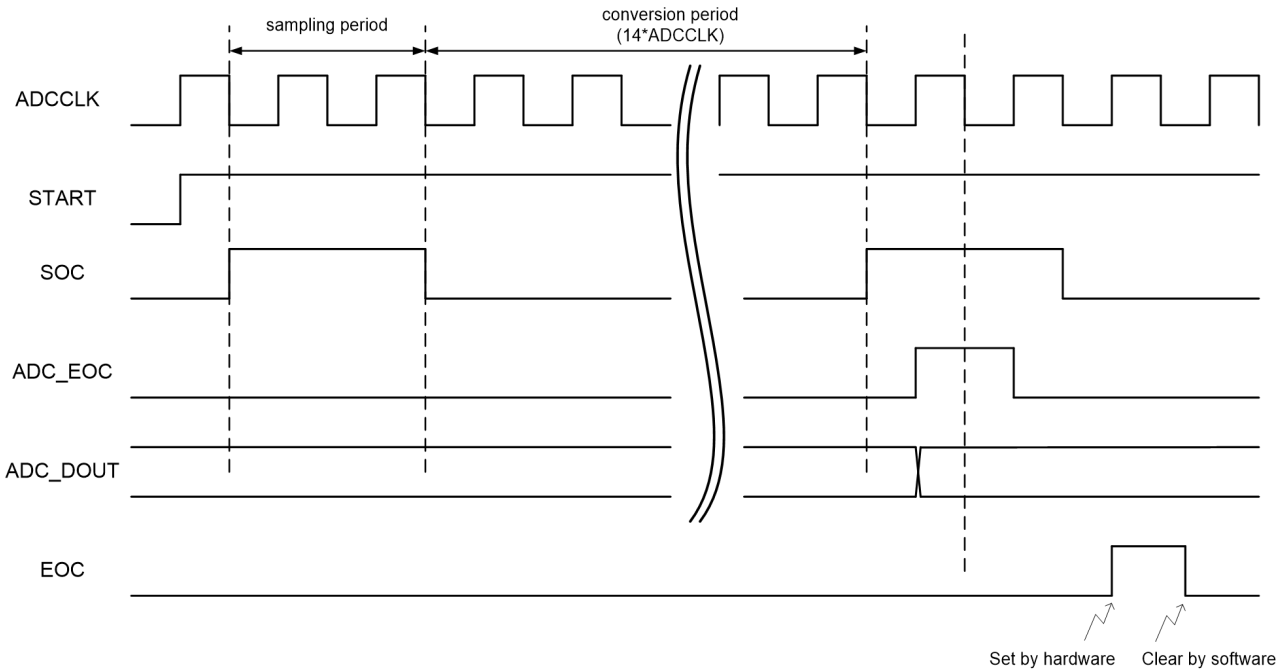


图 31-6 ADC 采样转换时序

多通道转换序列的时序示意图如下：

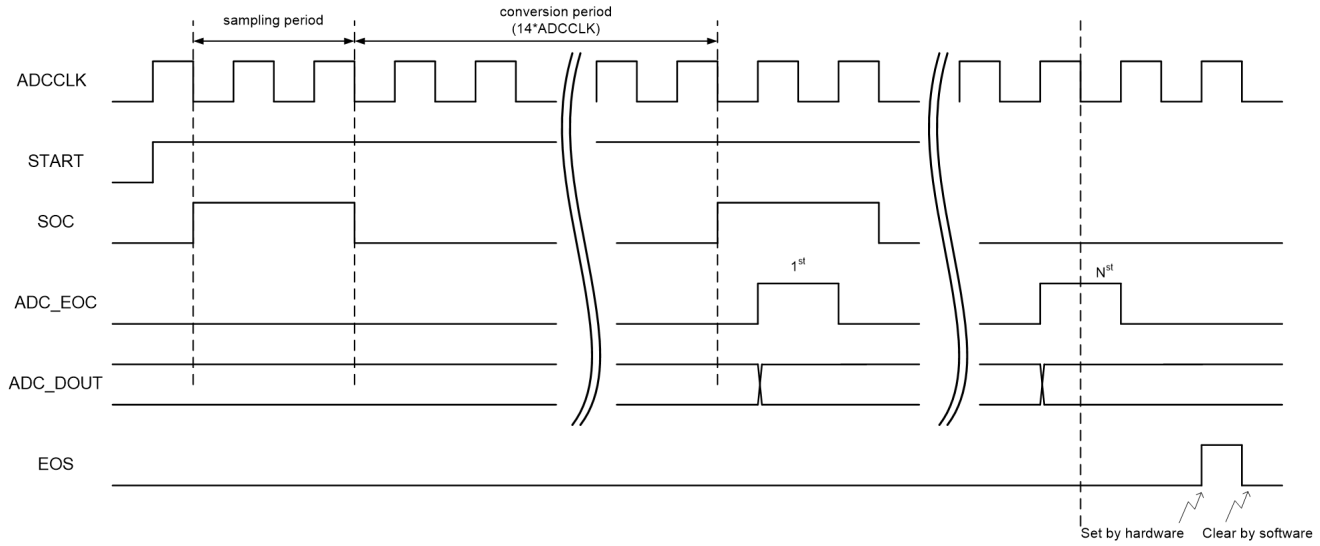


图 31-7 ADC 采样序列时序

SOC 信号对齐 ADC_CLK 的下降沿，ADC 产生的 EOC 信号对齐 ADC_CLK 的上升沿。

当 ADC_CLK 为 16MHz，采样时间配置为 2*ADCCLK 时，转换速率为 1Msps。

当 ADC_CLK 为 32MHz，采样时间配置为 2*ADCCLK 时，转换速率为 2Msps。

31.6 功能描述

31.6.1 使用 VDDA 作为基准

ADC 一般使用电源电压作为基准电压，在电源电压发生变化时，特定输入信号电平对应的转换值也会发生变化。为了能够得到准确的绝对电压，可以使用一下解决方案。

使用快速内部基准作为参考信号

芯片内部有一个快速建立的内部基准源，其输出电压是 1.0V，并且每颗芯片出厂时都经过校准，常温下精度为 1.0V+/-0.5%。

- ADC 采样并转换快速基准输出，并通过以下公式换算 VDDA 电压（假设 ADC 采样快速基准源的 12 位转换结果是 AVREF_DATA）：

$$VDDA = \frac{4095}{AVREF\_DATA} \times 1.0V$$

- 假设 ADC 对某个输入通道的采样值为 ADC_DATA，通过以下公式可以得到当前某个输入通道的实际电压（12bit 输出）

$$V_{CHANNEL} = \frac{ADC\_DATA}{AVREF\_DATA} \times 1.0V$$

31.6.2 使用 VREFINT 作为基准

如果使用 VREFINT 作为基准源，则 ADC 的工作参考电压是确定的，无需再通过采样内部基准信号来反推 ADC 参考电压。

此时采样外部通道时，输入电压的实际幅度可以通过下式确定：

$$V_{CHANNEL} = \frac{ADC\_DATA}{4095} \times VREFINT$$

31.6.3 温度传感器

根据温度传感器采样值计算当前温度

ADC 使用内部通道测量 PTAT 输出电压，得到转换数据 TS_DATA，当 ADC 工作参考电压是 VDDA 时，根据以下步骤可以计算当前实际温度：

1) 由以下公式计算当前温度传感器输出的绝对电压值

$$\text{VPTAT} = \frac{VREFIN\_CAL \times TS\_DATA}{VREFINT\_DATA \times 4095} \times 3V$$

2) 由以下公式计算温度标定时（30°C）温度传感器输出的绝对电压值

$$\text{VPTAT\_30C} = \frac{TS\_CAL1}{4095} \times 3V$$

3) 根据温度传感器输出斜率计算当前绝对温度

$$\text{Temperature} = \frac{VPTAT - VPTAT\_30C}{2.51mV/C} + 30C$$

其中，TS_DATA 是 ADC 采样当前温度传感器输出的转换值；由于不知道当前 VDDA 的准确电平，因此这个转换值需要根据 VREFINT 的转换结果进行比例缩放；TS_CAL1 是芯片生产时在 30C+/-1C、VDDA=3.0V 的条件下进行温度定标的转换结果，这个数据保存在 flash 中。

如果 ADC 当前工作参考电压是 VREFP，则当前温度可以通过下式计算：

$$Temperature = \frac{TS\_DATA \times \frac{VREFP}{3.0V} - TS\_CAL30}{slope} + 30C$$

根据温度传感器采样值进行 RTC 温度补偿

如果温度采样值只是用来做 RTC 温度补偿，则并不需要计算实际的温度值（-40~85 的十进制数），仅需要根据 $TS\_DATA \times \frac{VREFINT\_CAL}{VREFINT\_DATA}$ 的 12bit 结果（从物理原理来看这个结果就是 12bit），以 TS_CAL1 为中心点进行地址查表即可。因为上式计算结果代表的是折算到 VDDA=3V 情况下温度传感器的 12 位输出结果，它与 TS_CAL1 的差值，即为偏离 30°C 多少个 LSB，用这个信息作为地址对温度补偿校正表格查表，即可得到相应温度下的校正值。

使用温度传感器功能时，需要使能内部基准源的温度传感器输出，即同时置位 VREF_EN 和 PTAT_EN 寄存器，并且置位 VPTAT_BUFFER_EN 寄存器使能 PTAT 缓冲器，等待 5us 建立时间后，使能 ADC 对温度传感器通道进行采样。

31.6.4 温度传感器的斜率和标定

温度传感器工作电压范围为 1.8~5.5V，温度测量范围不小于-40~+85C，跨度 125C。PTAT 输出斜率为 2.51mV/°C。

31.6.5 可编程采样时间

通过调整采样时间，可以适应不同输入信号源的内阻。通过 SMTS1 和 SMTS2 寄存器可以选择采样时间：

SMTSx	Sampling cycles (T_{ADCCLK})
0000	2
0001	4
0010	8
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160
1011	192
1100	256
1101	320
1110	384
1111	512

表 31-2 ADC 采样时间

实际 ADC 的采样转换时间：

T_{CONV} = (Sampling Cycles + 14) * T_{ADCCLK}

ADC 采样时间主要由采样电容、被采样信号的输出阻抗、芯片内部输入通道阻抗和所需达到的采样精度共同决定。

下图是单端输入通道的电路结构示意：

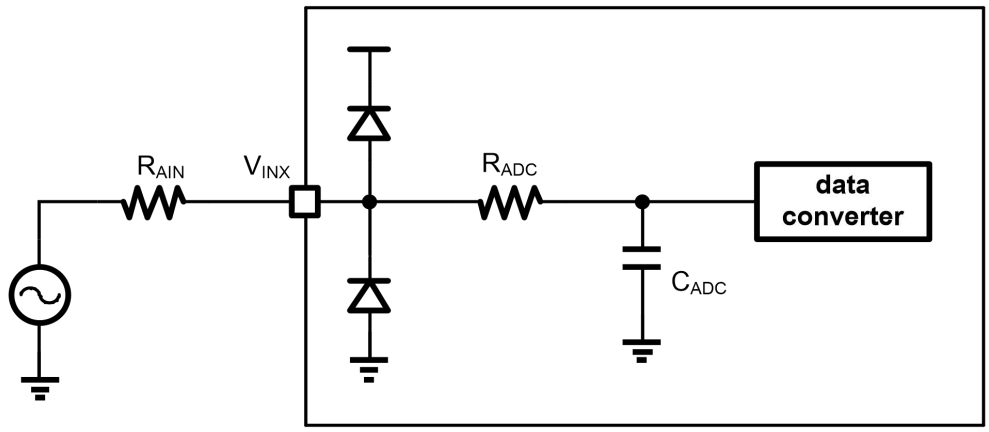


图 31-8 ADC 输入通道示意图

要求的采样时间可以根据下式估算：

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC}\right) \times C_{ADC}$$

其中，n=12，SA 表示容许的采样误差，比如 0.25 代表 1/4 LSB

应用中应根据芯片手册中的相关参数、以及系统参数，计算并确定可以接受的采样时间，并根据这个结果来配置 ADC 的工作时钟、采样周期等。

31.6.6 输出位宽选择

SAR-ADC 支持最高 12 位，最低 6 位的输出位宽设置。通过 BITSEL 寄存器可以配置输出数据位宽，以获得精度和速度的折中。

选择不同的输出位宽时，ADC 逐次逼近的转换周期数不同，而与 ADC 采样时间无关。

假设 ADC 采样时间设置为 2*T_{ADCCLK}，在 12bit 精度下，转换时间为 14*T_{ADCCLK}，总的采样转换时间为 16*T_{ADCCLK}；在 10bit 精度下，转换时间相应变为 12*T_{ADCCLK}。

BITSEL	resolution	Conversion time (T_{ADCCLK})	Max speed @ADCCLK=16Mhz
00	12	14	1Msps
01	10	12	1.14Msps
10	8	10	1.33Msps
11	6	8	1.6Msps
BITSEL	**resolution**	**Conversion time (T_{ADCCLK})**	**Max speed @ADCCLK=32Mhz**
00	12	14	2Msps
01	10	12	2.28Msps
10	8	10	2.66Msps
11	6	8	3.2Msps

表 31-3 ADC 输出位宽与速度

31.6.7 转换模式

ADC 支持以下转换模式：

- 单次转换
  - 半自动触发（SEMI-AUTOMATIC）
  - 全自动触发（AUTOMATIC）
- 连续转换

转换启动可以由软件或事件触发，通过寄存器选择多个事件触发源。

31.6.7.1 单次转换模式

支持半自动触发和全自动触发两种模式。

全自动触发模式：软件或硬件触发事件启动 ADC 转换后，ADC 会顺序采样所有被使能的通道，单个通道采样完成后，EOC（End of Conversion）标志置位，所有通道采样完成后，EOS（End of Sequence）标志置位，本次转换结束。假设通道 0、3、5 被使能

- 1st 触发事件：通道 0、3、5 被顺序采样，过程中产生三次 EOC，最终产生 EOS
- 2nd 触发事件：重复上述过程

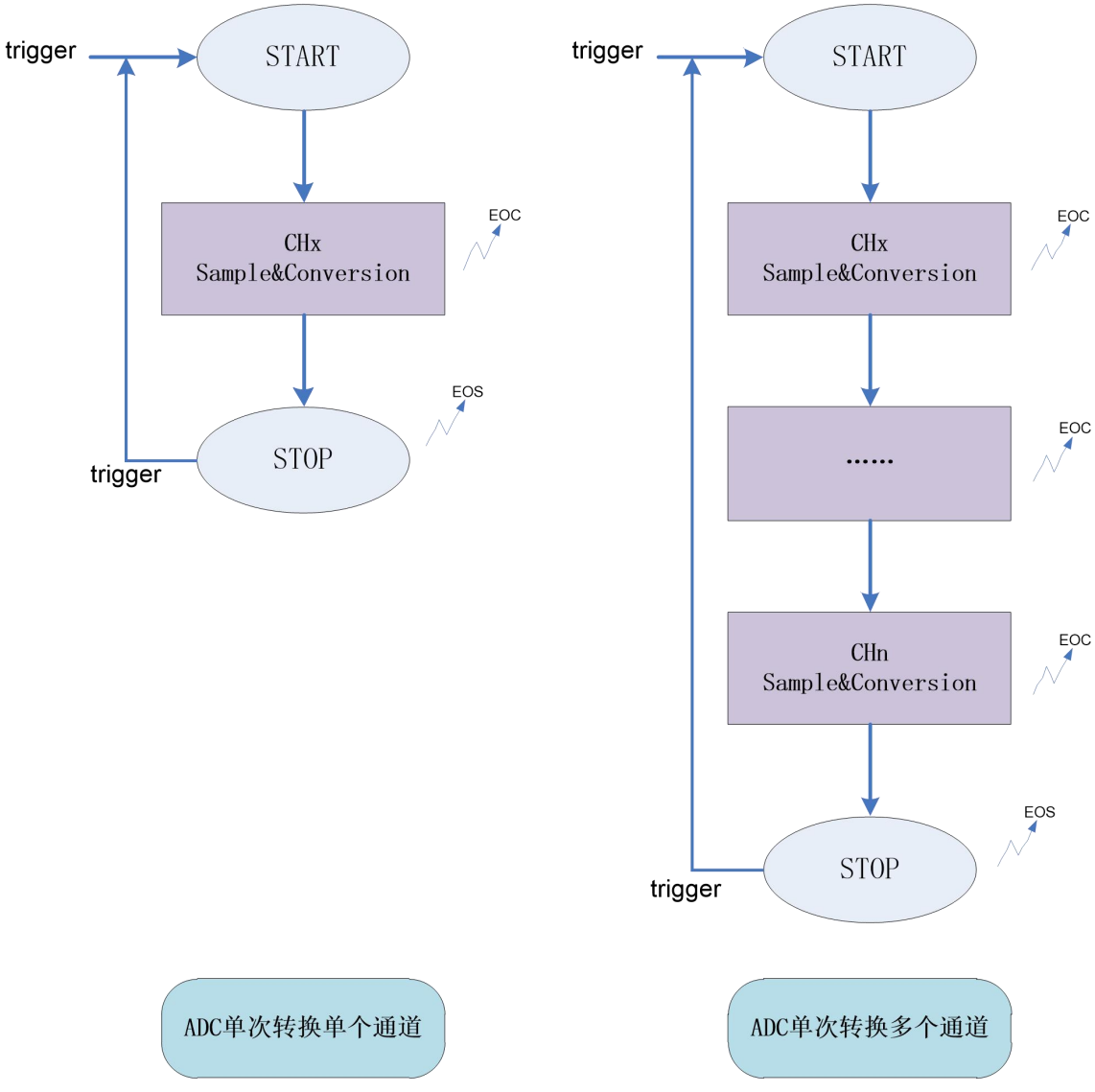


图 31-9 ADC 单次转换全自动触发模式

半自动触发模式：软件或硬件触发事件只会让 ADC 启动一次，转换一个使能通道。比如通道 0、3、

5 被使能

- 1st 触发事件：通道 0 被采样，产生 EOC
- 2nd 触发事件：通道 3 被采样，产生 EOC
- 3rd 触发事件：通道 5 被采样，产生 EOC 和 EOS
- 4th 触发事件：通道 0 被采样，产生 EOC
- 5th 触发事件：通道 3 被采样，产生 EOC

……

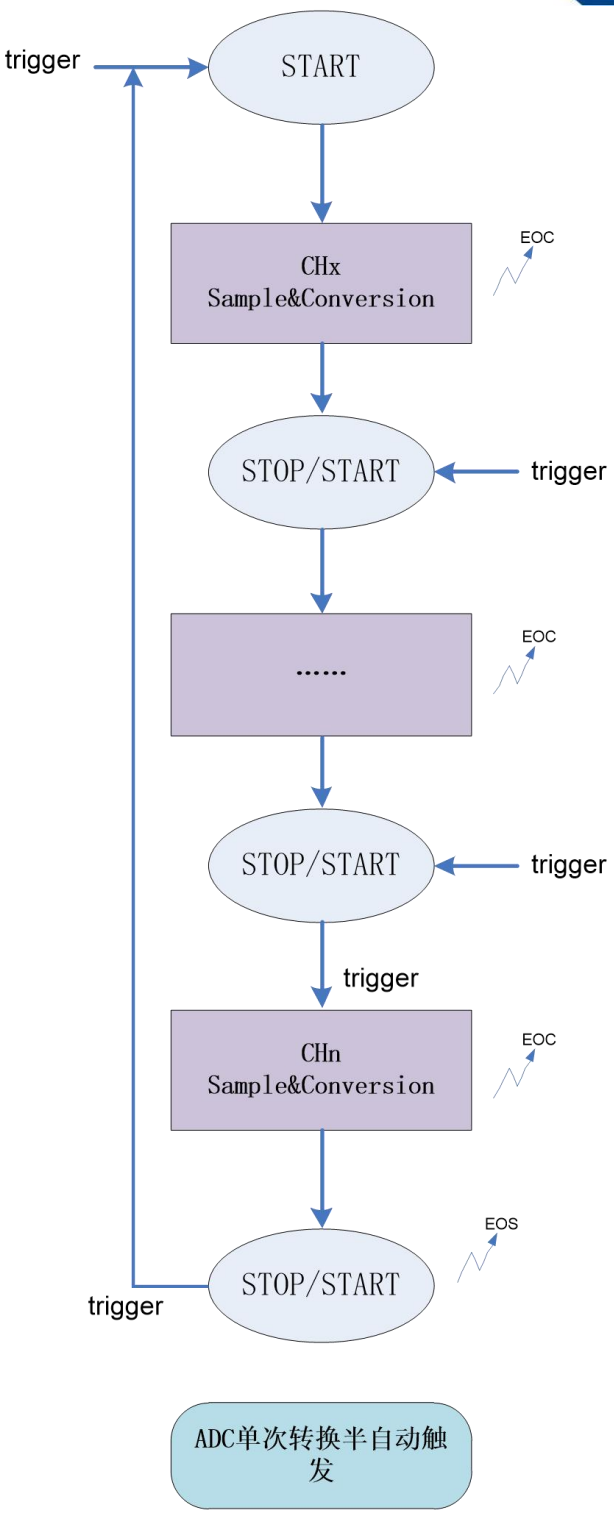


图 31-10 ADC 单次转换半自动触发模式

31.6.7.2 连续转换模式

触发事件到来后，所有使能通道被采样，并且 ADC 不会自动停止，而是循环采样，直到软件停止 ADC。

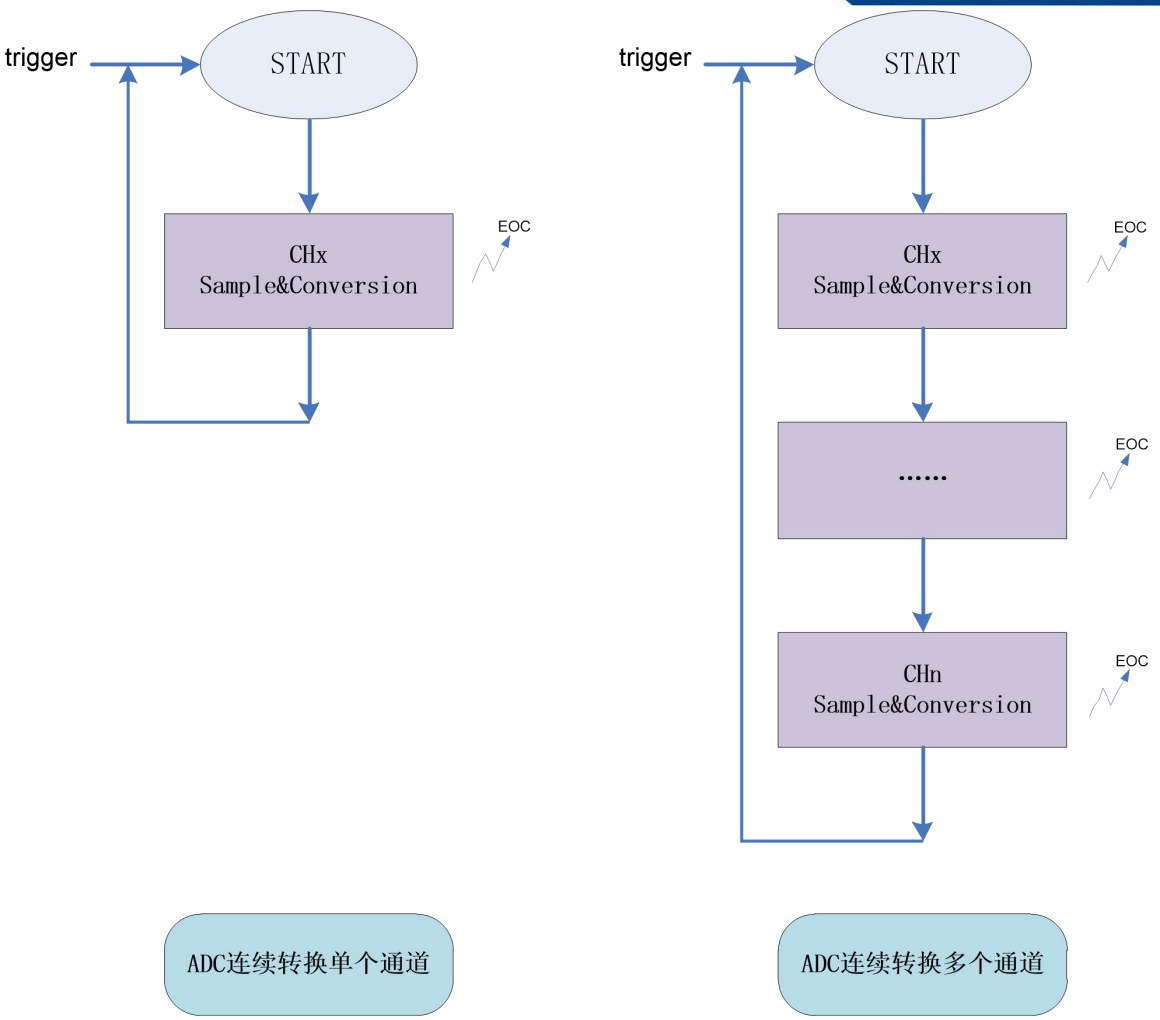


图 31-11 ADC 连续转换模式

每个通道被采样后，数据保存在 ADC_DATA 寄存器中，软件要在下次转换前及时读走数据，或者通过 DMA 进行数据搬移。如果不能及时取走数据，将引起 Overrun，置位 overrun 标志，并可以发出中断。

31.6.8 转换触发

ADC 使能后，转换触发支持软件或硬件事件触发。

软件触发

软件通过置位 SWTRIG 寄存器启动转换。

硬件触发

ADC 共有如下硬件触发源：LUTx_TRGO、RTC_TRGO、TAUxx_TRGO、BSTIM16_TRGO、

LPTIM16_TRGO、比较器输出；如果 ADC 正处于转换过程中，此时到来的触发信号会被忽略。

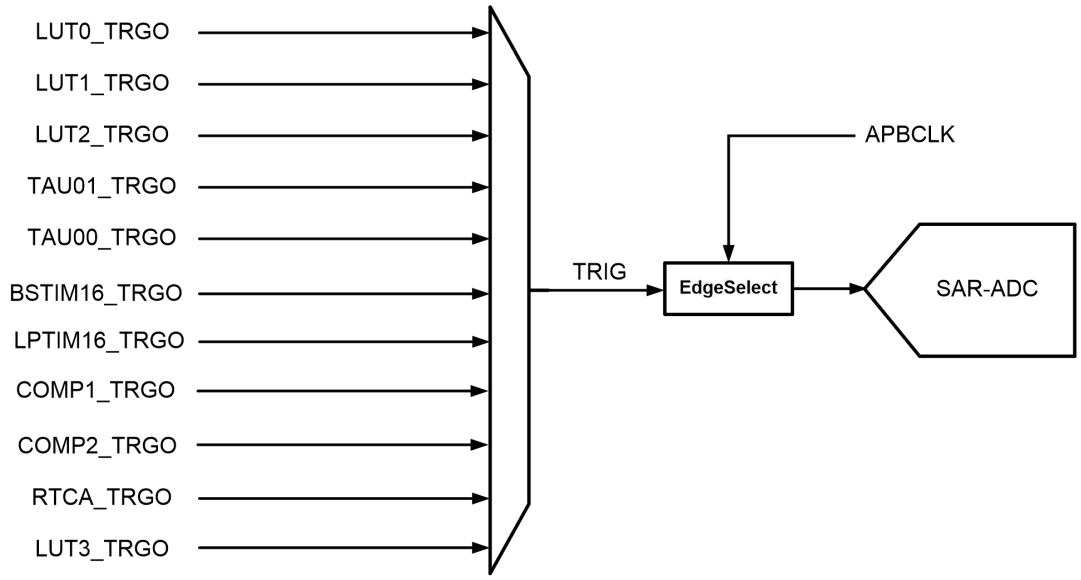


图 31-12 ADC 硬件触发源

注意：硬件触发源输入后经过 APBCLK 同步采样，因此使用硬件触发功能时必须使能 ADC 总线时钟（置位 CMU 模块中的 ADC_PCE 寄存器）。

触发源时序特征分类。

触发源	时钟域	当 ADC 工作时钟选择为 APBCLK 时的触发延迟 (T_{APBCLK})	说明
软件触发	APBCLK	3.5	可实现同步触发，确定延迟
LUTx_TRGO		4.5	
TAUx_TRGO		4.5	
COMPx_TRGO		4.5	
ATIM_TRGO	其他	2~3	异步时钟触发
BSTIM16_TRGO			
LPTIM16_TRGO			
RTC_TRGO			

31.6.9 过采样和硬件平均

ADC 支持硬件过采样平均，可以在一定程度上提高分辨率。原理是对于低速输入信号，可以通过连续多次采样后求平均的方法提高 ENOB，过采样公式如下：

$$result = \frac{\sum_{n=1}^{N} CONVERSION_n}{M}$$

其中 N 是过采样倍数，可配置为 2/4/8/16/32/64/128/256，M 为结果右移位数，最大右移 8bit；由于每次转换结果为 12bit，最大 256 次累加得到的结果为 20bit，经过移位后可以得到 12~16bit 最终

结果。ADC 输出结果最多只有 16bit，如果右移后结果超过 16bit，高位也会被丢弃。

在使能过采样的情况下，EOC 信号在 N 次连续采样后才置位，对于应用程序和 DMA 来说，感觉就好像只经过一次采样转换。

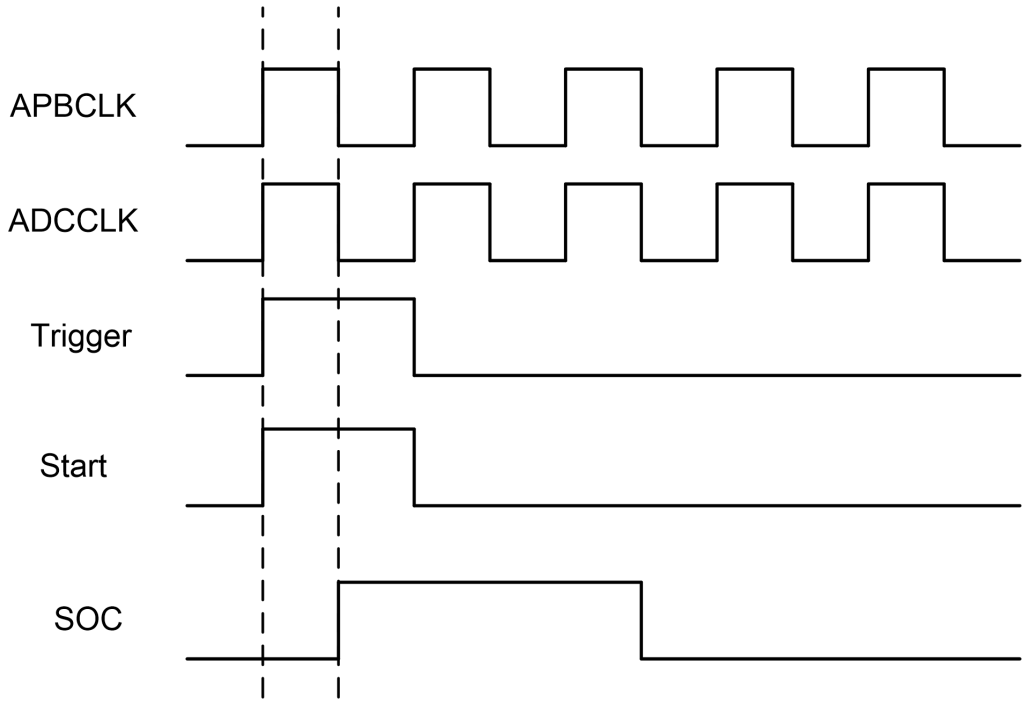
31.6.10 ADC 工作时钟

ADC 模块采用双时钟工作，其中 APBCLK 用于总线寄存器访问，并且也可以作为 ADC 工作时钟使用；ADCCLK 为 ADC 工作时钟，可以是 APBCLK 或者来自 CMU 的独立时钟。

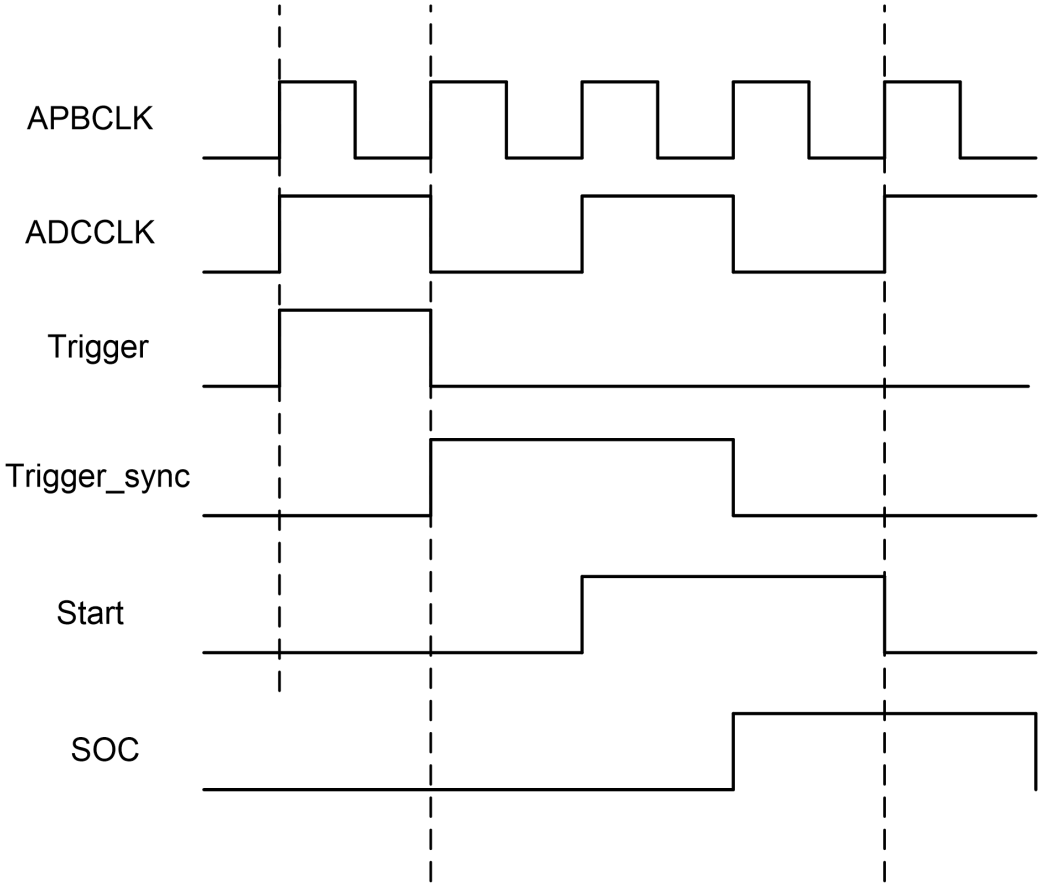
当 ADC 的采样转换工作在 APBCLK 下时，所有 APBCLK 下外设所产生的触发事件都是同步于 ADCCLK 的，则采样启动时间相对于触发事件的延迟是确定的，可用于时延敏感型应用场合。

注意，并不是所有触发源都支持确定延迟触发，只有触发信号输出为 APBCLK 时钟域的触发源才支持这一特性。参见“转换触发”章节。当 ADCCLK 选择为 APBCLK 分频时，触发事件到来与 ADC 采样启动之间的时间间隔在一定范围内有不确定性。

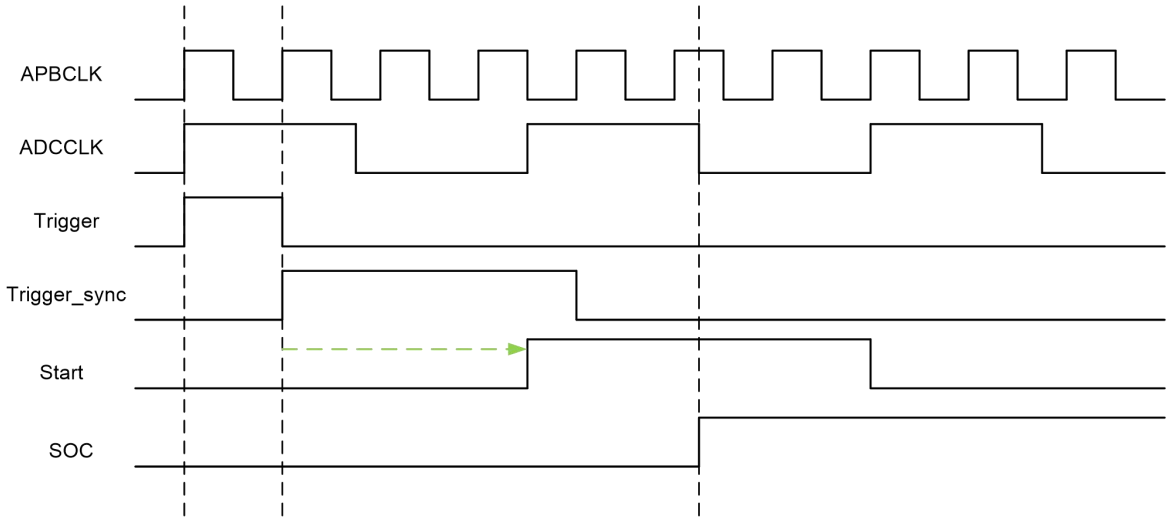
举例来说，下图显示了 ADCCLK=APBCLK 的情况（同步触发），触发延迟是固定的 2 个 APBCLK。



下图显示了 ADCCLK=APBCLK/2 的情况（作为异步时钟处理），触发延迟可能是 3 个或者 4 个 APBCLK。



下图显示了当 ADCCLK 完全异步于 APBCLK 时的触发时序。



当 ADC 的采样转换工作在独立时钟下时，ADC 工作频率可以与系统时钟频率解耦，比如 CPU 和总线可以工作在较低频率下，同时保持 ADC 工作在较高采样率，实现更高的应用灵活性。但是这种情况下，触发事件与 ADC 采样启动之间存在一个跨时钟域的重新同步延迟，这个延迟的大小与当前 APBCLK 和 ADCCLK 之间的频率及相位关系相关，有一定的变化范围。

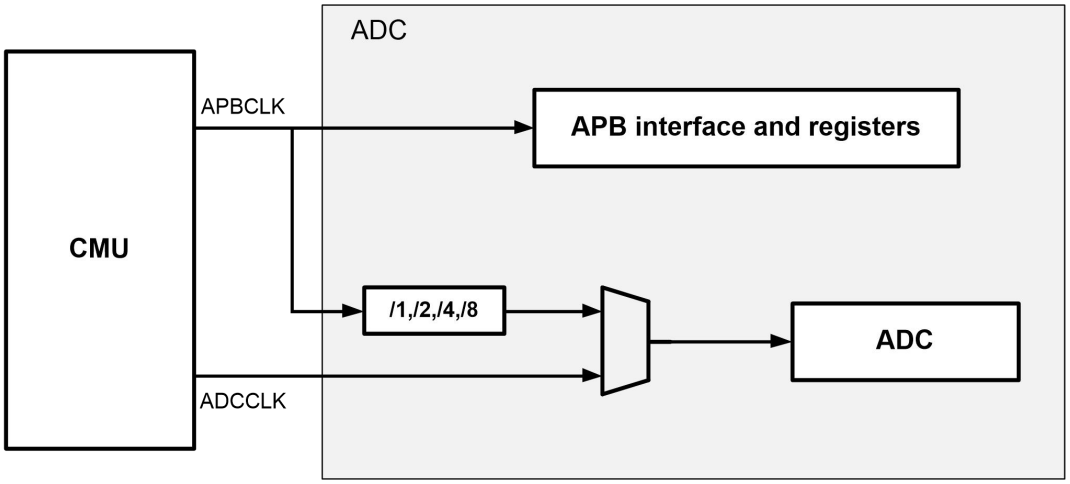


图 31-13 ADC 时钟示意图

注意，推荐使用 ADCCLK=APBCLK，即 ADC_CFGR1.CLKSEL=1。

31.6.11 数据冲突和自动等待

每次转换完成后 EOC 标志会置位，软件或 DMA 读取 ADC_DATA 寄存器后会自动清除 EOC，也可以由软件写 1 清除。当 EOC 标志没有被清除的情况下，新的转换数据到来，就会导致 data overrun；有两种 overrun 模式：

OVRMOD=0：保持旧的数据，新数据丢弃

OVRMOD=1：写数据写入覆盖旧数据

当使用 DMA 时出现 overrun，则不会发起新的 DMA request，直到 OVR 标志被软件清零

ADC 控制器还支持自动等待，如果 WAIT 寄存器被软件置位，那么在 ADC_DATA 寄存器被读取之前，ADC 控制器不会发起新的转换；在等待状态中到来的硬件触发事件也会被忽略。

下图是软件触发连续模式的情况下，使能了自动等待的示意图：

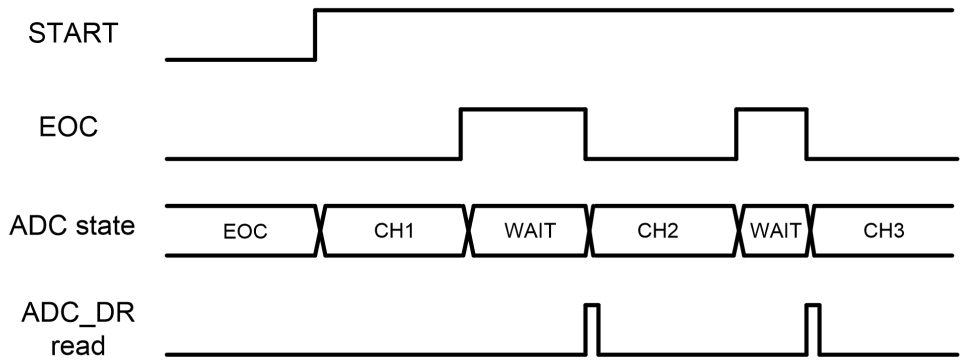


图 31-14 ADC 自动等待

注意：DMA 循环模式下请保持 WAIT=0

31.6.12 DMA

在多通道转换或连续转换时，使用 DMA 进行转换结果搬移是高效的解决方案。在使能了 DMAEN 的情况下，当每次转换完成后（EOC），ADC controller 模块会产生一个 DMA 请求，通知 DMA 将数据寄存器中的结果搬运到指定的 SRAM 地址。ADC 转换由触发事件启动，触发可以来自于软件触发或硬件触发。DMA 模式下支持一次触发后连续执行多次转换（自动模式），或者每次触发执行一次转换的方式（半自动模式）。

在自动模式下（ADC_CFGR2.SEMI=0）和半自动模式下（ADC_CFGR2.SEMI=1），ADC 的 DMA 接口都可以支持单次模式和循环模式：

单次模式

转换完成后发起数据搬运，此过程会一直重复，直到软件配置的 DMA 传输长度完成，然后 ADC 控制器会自动停止转换（通过接收 DMA 的传输完成中断标志信号），关闭 ADC，不再向 DMA 发起请求。此模式主要用于对特定模拟信号进行一定长度的采样。

循环模式

与 DMA 的循环模式相配合，ADC 不断循环转换并发起 DMA 请求，直到软件停止转换。此模式可以用于处理连续不断模拟信号采样。

通过 ADC_CFGR2.SEMI 寄存器可以配置 DMA 模式下的自动或半自动转换，通过 DMA 模块的 CHxCIRC 配置循环模式或单次模式，通过 DMA 模块的 CHxTSIZE 配置传输长度。寄存器配置组合实现的效果如下表所示。

ADC_CFGR2.CONT	ADC_CFGR2.SEMI	DMA CHxCIRC	DMA CHxTSIZE	说明
0	0	0	N	单次全自动模式，触发事件到来后，依次对使能输入通道转换 N 次并将数据搬运到 SRAM；所有使能输入通道都被采样过，ADC 自动停止并等待下一次触发；如果数据搬运次数达到 N，DMA 自动关闭。
0	0	1	N	单次全自动模式（循环存储），触发事件到来后，依次对使能输入通道不断转换并搬运数据，直到所有使能输入通道都采样过，ADC 自动停止并等待下一次触发；RAM 数据空间长度为 N，当搬运

				数据长度超过 N 之后回到 DMA 通道指针指向的起始地址，并覆盖原数据。
0	1	0	N	单次半自动模式，每个触发事件启动一次转换，依次采样所有被使能的输入通道；数据搬运次数达到 N，DMA 自动关闭。
0	1	1	N	单次半自动模式（循环存储），每个触发事件启动一次转换，依次采样所有被使能的输入通道；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到 DMA 通道指针指向的起始地址，并覆盖原数据。
1	x	0	N	连续模式，触发后 ADC 不断采样使能通道，DMA 不断搬运数据，直到完成 N 次数据搬运，DMA 自动关闭。
1	x	1	N	连续模式（循环存储），触发后 ADC 不断采样使能通道，DMA 不断搬运数据；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到起始地址覆盖原数据；直到软件关闭 DMA。

表 31-4 DMA 配置与功能

在 DMA 使能情况下，如果发生 overrun，则 ADC 控制器不再发送 DMA 请求，直到 OVR 标志被清除。

注意，在单次和连续转换模式下，都可以支持 DMA 传输；DMA 传输长度以 EOC 的次数定义，而不是 EOS，即 DMA 只关心搬运多少次 ADC_DATA。

DMA 循环模式下应保持 WAIT=0

31.6.12.1 DMA 使用案例

使用案例 1：

ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据，然后 ADC 和 DMA 暂停工作，等待第二次触发。第二次触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据。6 次 DMA 传输完成，DMA 通道关闭。

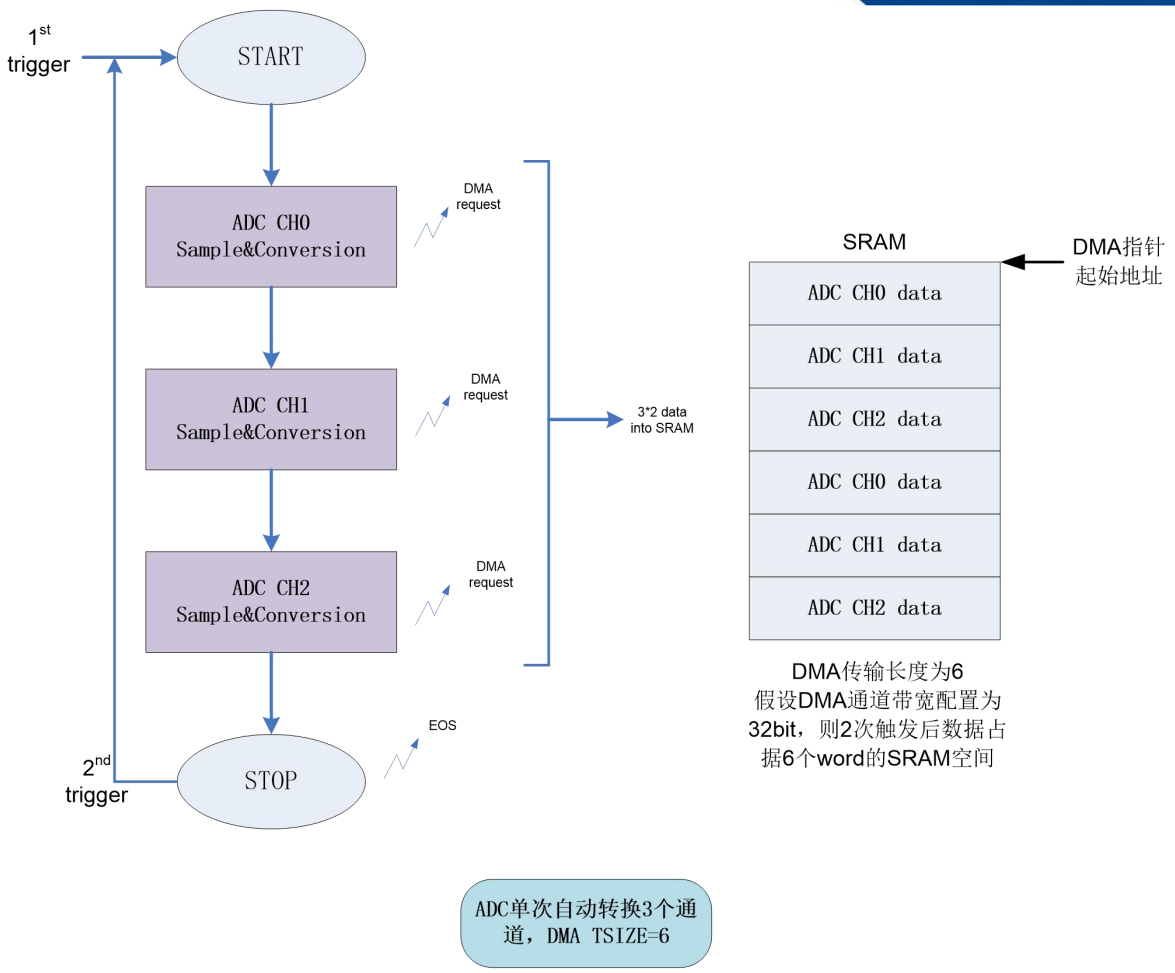


图 31-15 ADC 单次全自动触发+DMA 案例 1

使用案例 2：

ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 2。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 2 次数据后 DMA 和 ADC 自动关闭，不会对第三个输入通道进行采样。

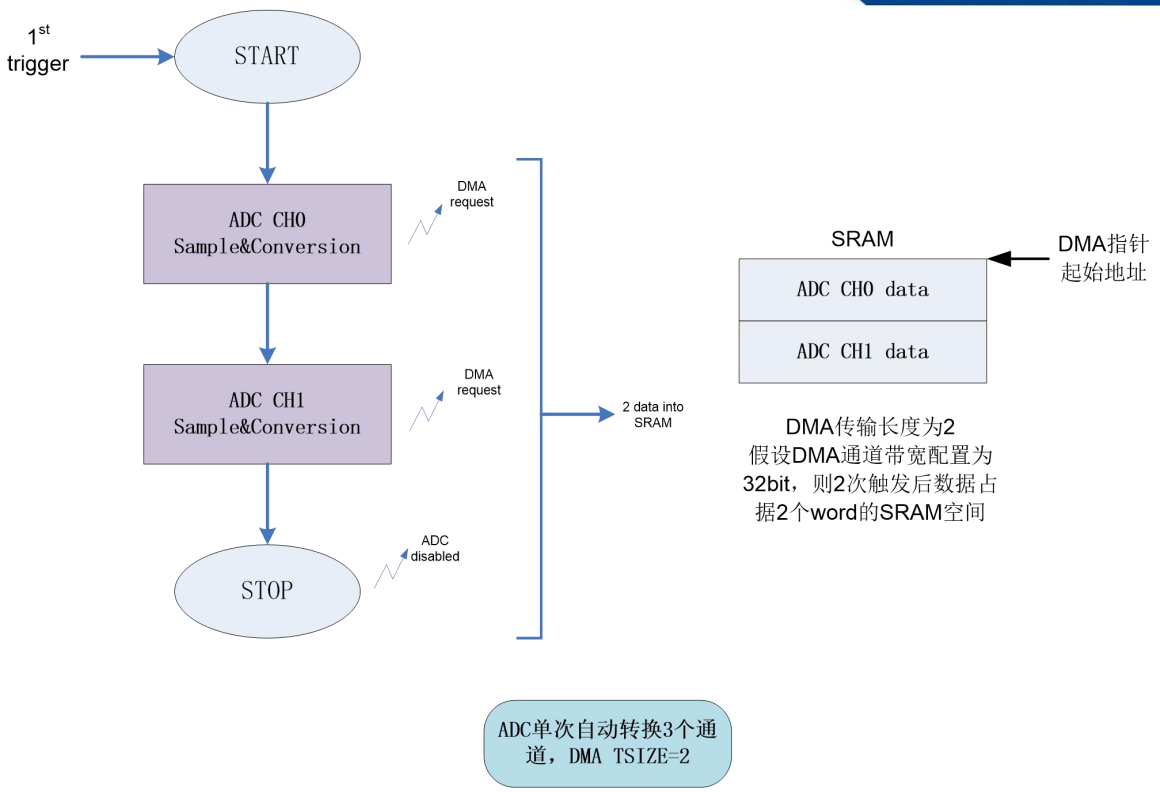


图 31-16 ADC 单次全自动触发+DMA 案例 2

使用案例 3：

ADC 配置为单次半自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。每次触发后，DMA 搬运一个转换数据，总共 6 个数据搬运完毕后，自动关闭 ADC 和 DMA。

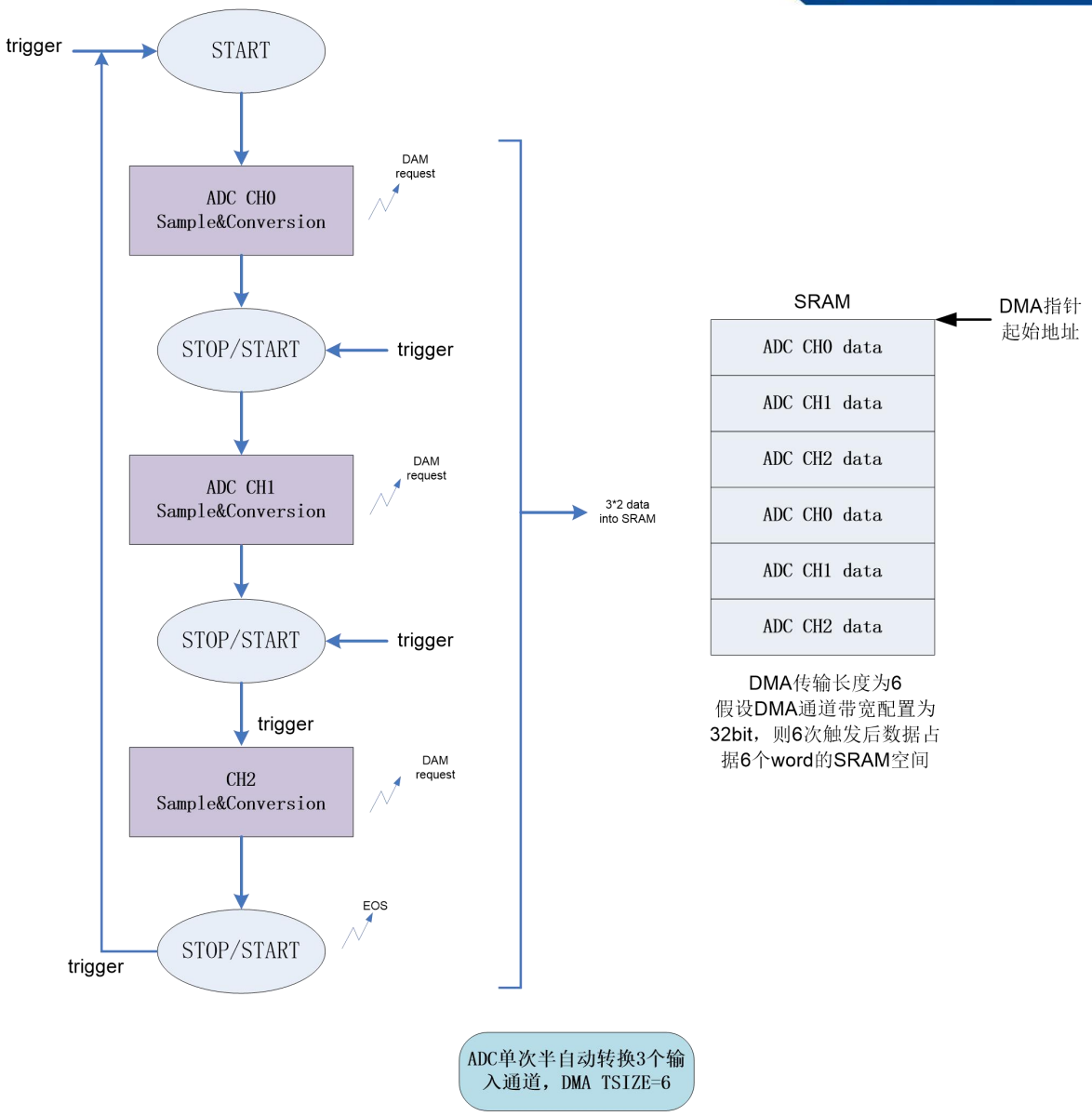


图 31-17 ADC 单次半自动触发+DMA

使用案例 4：

ADC 配置为全自动模式（SEMI=0），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 执行 3 次采样转换，DMA 搬运 3 次数据到 RAM 的 x/x+1/x+2 地址；第二次触发后，ADC 执行 3 次采样，DMA 搬运 3 次数据到 RAM 的 x+3/x+4/x+5 地址；第三次触发后，ADC 执行 3 次采样，DMA 搬运 3 个数据到 RAM 的 x/x+1/x+2 地址，循环往复直到软件关闭 DMA。

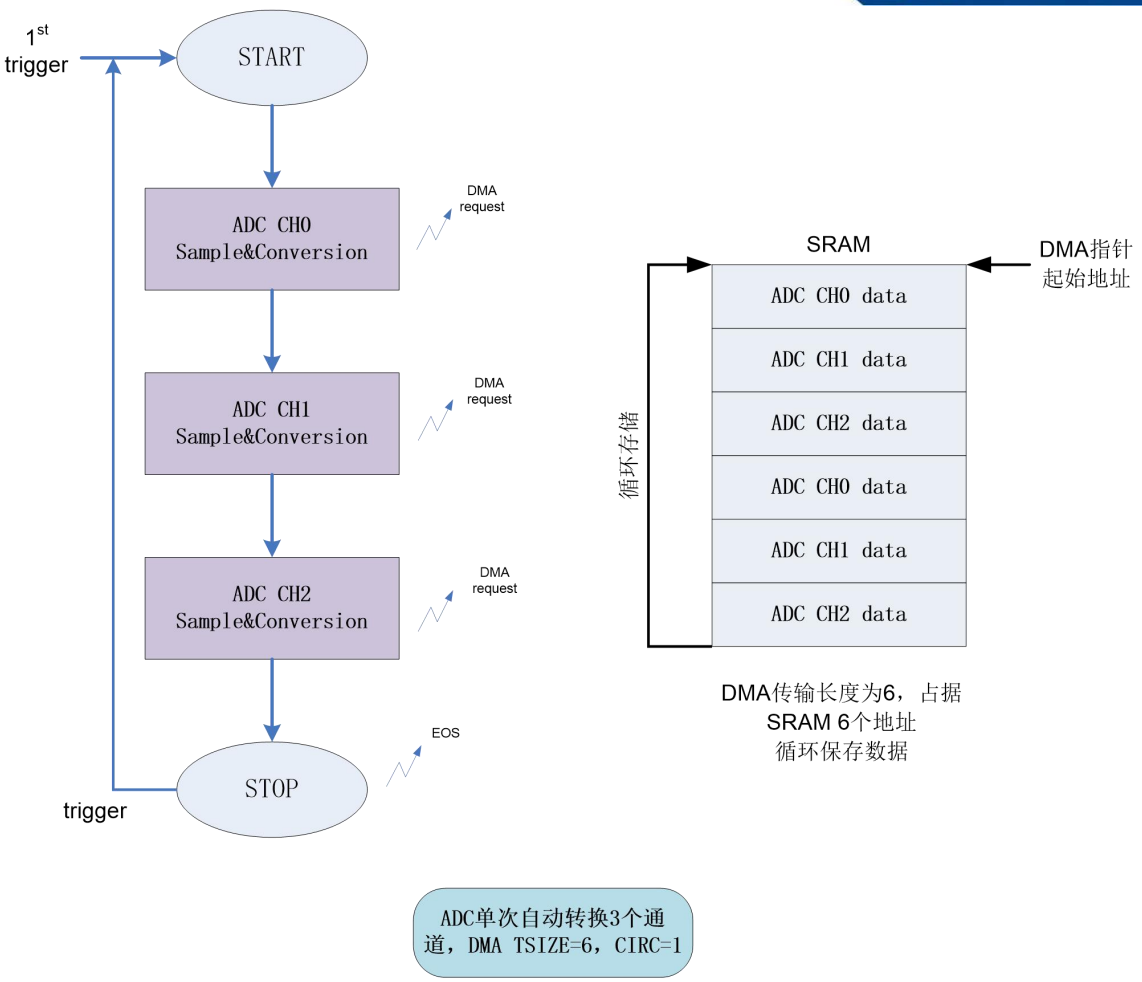


图 31-18 ADC 全自动触发+DMA 循环模式

使用案例 5：

ADC 配置为连续模式（CONT=1），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 对使能通道连续不间断采样转换，DMA 连续搬运数据到 RAM 的 x/x+1/x+2/x+3/x+4/x+5 地址；循环往复直到软件关闭 DMA。

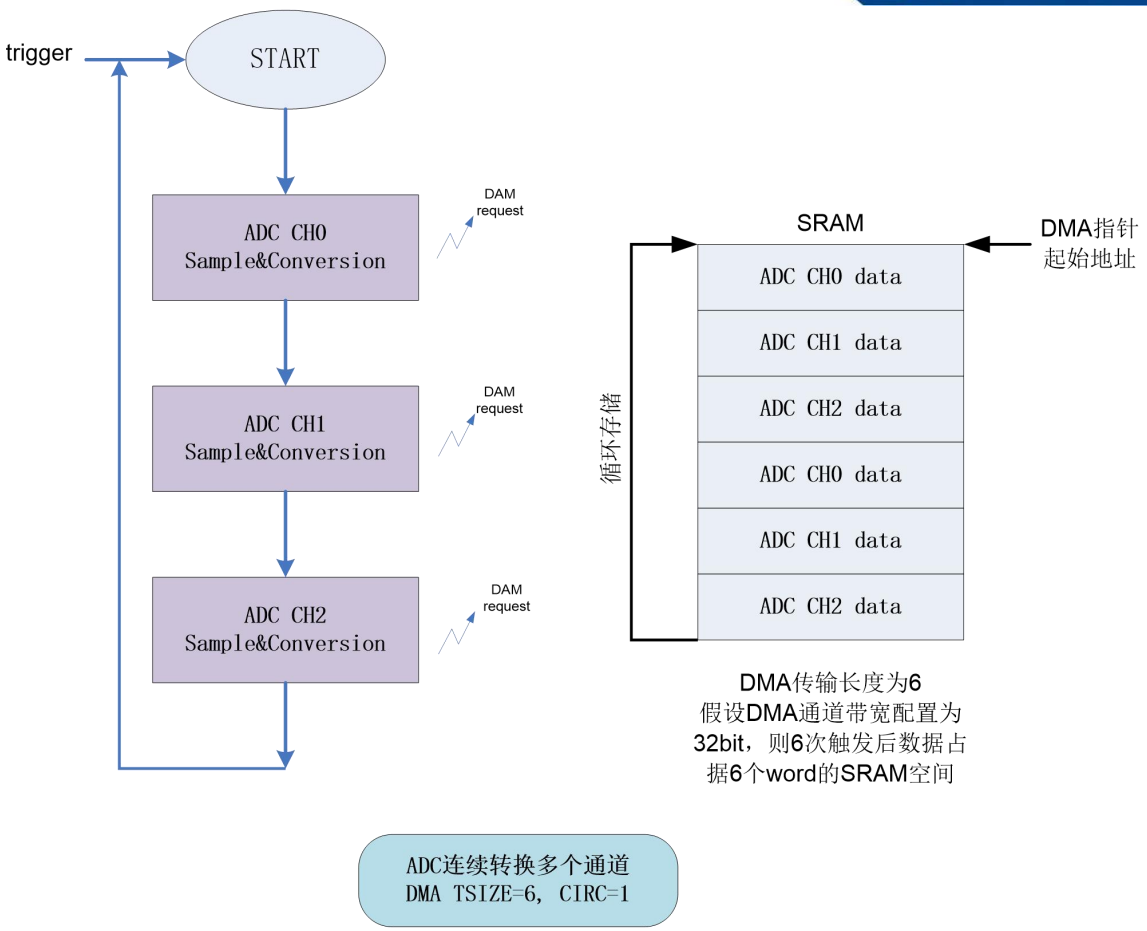


图 31-19 ADC 连续模式+DMA 循环模式

31.6.13 模拟窗口看门狗（AWD）

AWD 功能用于监视某个模拟输入通道或所有输入通道的输入信号电平是否处于寄存器设置的幅值范围之内。当 ADC 转换值高于 AWD_HT 或者低于 AWD_LT 时，都会置位中断标志寄存器。标志寄存器由软件写 1 清零。

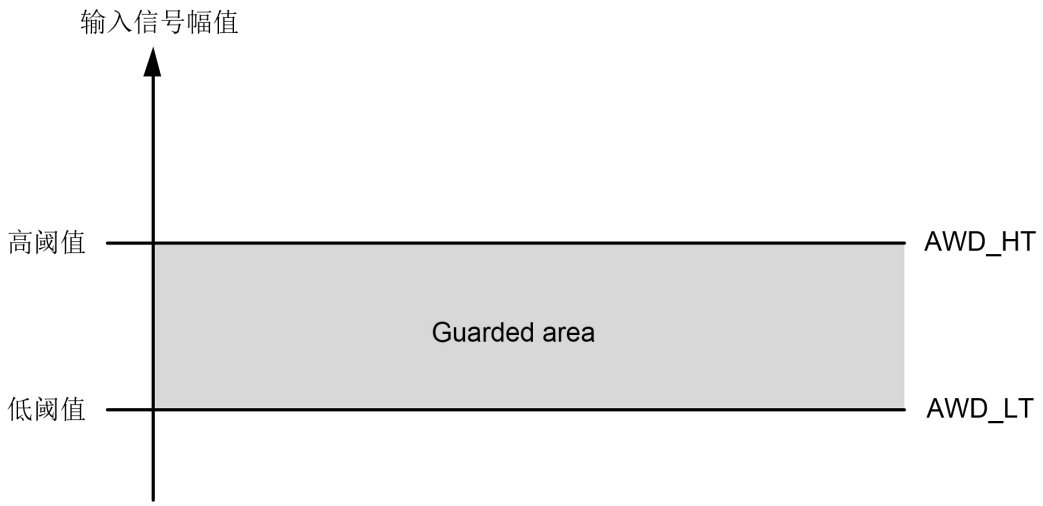


图 31-20 ADC 模拟看门狗阈值示意图

通过 AWDEN 寄存器使能模拟窗口看门狗功能，通过 AWDSC 寄存器配置单通道监视或全部通道监视。

AWD 阈值设置寄存器的位宽是 16bit。

31.6.14 ADC 校准

ADC 支持 offset 自校准，建议在芯片上电后，首先进行一次校准操作，以获得更好的精度。

Offset 校准包含两部分：比较器校准和通道（单端）校准，分别用于消除比较器 offset 和采样通道的 offset。两种校准由 CALEN 寄存器启动。

当 CALSEL=0 时，软件置位 CALEN 启动校准，硬件完成 ADC 自校准操作后会自动清零 CALEN 寄存器，校准操作结束后 EOCAL 中断标志置位，如果中断使能 EOCALIE 为 1，则产生中断通知 CPU。

注：ADC 复位后，在第一次转换之前，建议启动一次校准操作。校准完成后，重新使能 ADC 不需要再次进行校准即可进行转换。因此建议用户在芯片上电复位完成后先进行一次校准操作。当 ADC 工作环境（温度、电压）发生显著变化时，也推荐进行校准以获得更高精度。

进行校准操作后，校准参数被保存在 ADC 内部寄存器，ADC 模块复位或 ADC 控制器复位都不会清除校准寄存器内容，只有上下电复位会清除校准寄存器。

31.6.15 模拟 buffer

慢速通道 ADC_IN14~ADC_IN22 的输入可以经过模拟 buffer 缓冲，以测量高输出阻抗微弱信号。使用内部 buffer 时，对上述输入通道的采样都通过 BUF4TST_OUT 通道，软件需要事先配置 BUF4TST 输入 MUX 选择其中一路信号进行缓冲。

31.7 低功耗模式

在低功耗模式下，芯片自动关闭了所有高速时钟源，所以 ADC 无法工作。

31.8 寄存器

模块起始地址：0x4001_AC00

offset 地址	名称	符号
0x00	ADC 中断和状态寄存器（ADC Interrupt and Status Register）	ADC_ISR
0x04	ADC 中断使能寄存器（ADC Interrupt Enable Register）	ADC_IER
0x08	ADC 控制寄存器 1（ADC Control Register1）	ADC_CR1
0x0C	ADC 控制寄存器 2（ADC Control Register2）	ADC_CR2
0x10	ADC 校准控制寄存器（ADC Calibration Register）	ADC_CALR
0x14	ADC 配置寄存器 1（ADC Config Register1）	ADC_CFGR1
0x18	ADC 配置寄存器 2（ADC Config Register2）	ADC_CFGR2
0x1C	ADC 采样时间控制寄存器（ADC Sampling Time Register）	ADC_SMTR
0x20	ADC 通道控制寄存器（ADC Channel Enable Register）	ADC_CHER
0x24		ADC_DCR
0x28	ADC 数据寄存器（ADC Data Register）	ADC_DR
0x2C	ADC 软件采样控制寄存器（ADC Sampling Register）	ADC_SAMPT
0x30	ADC 模拟看门狗阈值寄存器（ADC analog watchdog Threshold Register）	ADC_HLTR

模块起始地址：0x4001_9800

offset 地址	名称	符号
0x00	BUF4TST 配置寄存器（BUF4TST Config Register）	BUF4TST_CFGR

31.8.1 ADC 中断和状态寄存器（ADC_ISR）

名称	ADC_ISR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AH	AWD_UL	EOCAL	BUSY	OVR	EOS	EOC
位权限	U-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	AWD_AH	模拟看门狗超出上限中断标志 (Analog Watchdog Above High)，当采样值高于 AWD_HT 时，硬件置位，软件写 1 清零
5	AWD_UL	模拟看门狗低于下限中断标志 (Analog Watchdog Under Low)，当采样值低于 AWD_LT 时，硬件置位，软件写 1 清零
4	EOCAL	校准结束（End Of Calibration），硬件置位，软件写 1 清零 1：校准过程结束 0：无校准过程
3	BUSY	ADC 忙标志 (ADC busy) 1：ADC 正在校准、采样或转换过程中 0：ADC 空闲
2	OVR	数据冲突标志，硬件置位，软件写 1 清零。(Over-Run) 当 ADC_DATA 寄存器中的上一次转换结果还未被读取，新的转换结果又到来时，硬件置位 OVR 标志。 0：没有数据冲突 1：出现数据冲突
1	EOS	转换序列结束（End Of Sequence） 所有使能通道都转换完成后，置位 EOS，软件写 1 清零。
0	EOC	单次转换结束（End Of Conversion） 每个通道转换完成后，置位 EOC，软件写 1 清零。

31.8.2 ADC 中断使能寄存器（ ADC_IER）

名称	ADC_IER							
Offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	AWD_AHIE	AWD_ULIE	EOCALIE	-	OVRIE	EOSIE	EOCIE
位权限	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	AWD_AHIE	模拟看门狗采样值高于上限中断使能，1 有效

Bit	助记符	功能描述
5	AWD_ULIE	模拟看门狗采样值低于上限中断使能，1 有效
4	EOCALIE	校准结束中断使能寄存器 0：禁止 EOCAL 中断 1：允许 EOCAL 中断
3	--	RFU：未实现，读为 0
2	OVRIE	数据冲突中断使能寄存器 0：禁止数据冲突中断 1：允许数据冲突中断
1	EOSIE	转换序列结束（End Of Sequence）中断使能寄存器 0：禁止 EOS 中断 1：允许 EOS 中断
0	EOCIE	单次转换结束（End Of Conversion）中断使能寄存器 0：禁止 EOC 中断 1：允许 EOC 中断

31.8.3 ADC 控制寄存器 1（ADC_CR1）

名称	ADC_CR1							
Offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						SWTRIG	ADEN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	SWTRIG	ADC 启动转换寄存器（软件触发），软件写 1 启动，硬件自动清零。
0	ADEN	ADC 使能寄存器。在启动转换前先要置位 ADEN 0：关闭 ADC 1：使能 ADC

31.8.4 ADC 控制寄存器 2（ADC_CR2）

名称	ADC_CR2							
Offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TRGCFG	
位权限	U-0						R/W-00	

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
1:0	TRGCFG	触发信号使能和极性选择。(Trigger Config) 当配置为非软件触发时，禁止软件写 SWTRIG 寄存器。 00：软件触发 01：硬件上升沿触发 10：硬件下降沿触发 11：硬件上升、下降沿都触发

31.8.5 ADC 校准控制寄存器（ADC_CALR）

名称	ADC_CALR							
Offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OSCAL_CYCLE							
位权限	R/W-0000 0111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			OFFL_EN				
位权限	U-0			R/W-0				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	VCM_CTRL	VCM_MODE	CMPRDY_DELAY		CMP_MODE	CKDIG_DELAY		CALEN
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-00		R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:16	OSCAL_CYCLE	Offset 校准周期数配置
15:13	-	RFU：未实现，读为 0
12	OFFL_EN	离线校准使能 (offline calibration enable) 0：自动校准 1：离线校准
11:8	-	RFU：未实现，读为 0
7	VCM_CTRL	共模电流配置
6	VCM_MODE	VCM 控制模式

位号	助记符	功能描述
		0：正常 1：长通
5:4	CMPRDY_DELAY	比较器延迟控制 00：延迟最小 11：验延迟最大
3	CMP_MODE	比较器模式选择 0：正常模式 1：快速模式
2:1	CKDIG_DELAY	数字时钟延迟选择 00：延迟最小 11：延迟最大
0	CALEN	Offset Calibration 使能 (Offset Calibration Enable) 软件写 1 启动校准周期，校准结束后自动清零并置位 EOCAL 寄存器。 Offset 校准分为比较器校准和通道校准，由 CALSEL 寄存器选择校准项目。两种校准都是由 CALEN 寄存器启动的。

31.8.6 ADC 配置寄存器 1（ADC_CFGR1）

名称	ADC_CFGR1							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							REFNSEL
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	APBCLK_PSC		-	BITSEL		CLKSEL	REFPSEL	
位权限	R/W-00		U-0	R/W-00		R/W-0	R/W-00	

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
8	REFNSEL	ADC 负基准选择(Negative Reference Select) 0：VSSA 1：VREFN
7:6	APBCLK_PSC	APBCLK 用作 ADC 工作时钟前的预分频 (APBCLK prescaler) 00：不分频 01：2 分频 10：4 分频 11：8 分频
5	--	RFU：未实现，读为 0
4:3	BITSEL	ADC 输出位宽选择 (Bit-width)

Bit	助记符	功能描述
		00：12 位 01：10 位 10：8 位 11：6 位
2	CLKSEL	ADC 工作时钟选择 (Clock Select) 0：使用 ADCCLK 工作 1：使用 APBCLK 工作 ***注意：推荐使用 CLKSEL=1***
1:0	REFPSEL	ADC 正基准选择 (Positive Reference Select) 00,11：使用 VDDA 作为基准 01：使用 VREFP 作为基准 10：使用 VREFINT 作为基准 *注意：当使用 VREFP 和 VREFN 引脚作为输入通道时，必须将 ADC 基准源选择为 VDDA*

31.8.7 ADC 配置寄存器 2（ADC_CFGR2）

名称	ADC_CFGR2							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	AWDCH					AWDSC	AWDEN
位权限	U-0	R/W-00000					R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OVSS				OVSR			OVSEN
位权限	R/W-0000				R/W-000			R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	IOTRFEN	-		SEMI	WAIT	CONT	OVRM
位权限	U-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTS				-	SCANDIR	-	DMAEN
位权限	R/W-0000				U-0	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30:26	AWDCH	模拟窗口看门狗监视通道选择，仅在 AWDSC=1 时有效 00000：AWD 监视 ADC_IN0 00001：AWD 监视 ADC_IN1 00010：AWD 监视 ADC_IN2 00011：AWD 监视 ADC_IN3 00100：AWD 监视 ADC_IN4 00101：AWD 监视 ADC_IN5 00111：AWD 监视 ADC_IN6 01000：AWD 监视 ADC_IN7 01001：AWD 监视 ADC_IN8 01010：AWD 监视 ADC_IN9

Bit	助记符	功能描述
		01011：AWD 监视 ADC_IN10 01100：AWD 监视 ADC_IN11 01101：AWD 监视 ADC_IN12 01110：AWD 监视 ADC_IN13 01111：AWD 监视 ADC_IN14 10000：AWD 监视 ADC_IN15 10001：AWD 监视 ADC_IN16 10010：AWD 监视 ADC_IN17 10011：AWD 监视 ADC_IN18 10100：RFU 10101：RFU 10110：RFU 其他：保留
25	AWDSC	模拟窗口看门狗单通道或全通道选择 0：AWD 监视所有被使能的外部输入通道 1：AWD 监视 AWDCH 指定的单个通道
24	AWDEN	模拟窗口看门狗使能寄存器 0：关闭 AWD 1：使能 AWD 仅能在 ADEN=0 的情况下配置此寄存器
23:20	OVSS	过采样移位控制寄存器 0000：不移位 0001：右移 1bit 0010：右移 2bit 0011：右移 3bit 0100：右移 4bit 0101：右移 5bit 0110：右移 6bit 0111：右移 7bit 1000：右移 8bit Others：RFU
19:17	OVSR	过采样率控制 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x
16	OVSEN	过采样使能 0：禁止过采样 1：使能过采样
15	--	RFU：未实现，读为 0
14	IOTRFEN	引脚触发信号数字滤波使能 0：禁止数字滤波 1：使能数字滤波
13:12	--	RFU：未实现，读为 0
11	SEMI	单次转换半自动模式，仅在单次转换（CONT=0）时有效，参见

Bit	助记符	功能描述
		“转换模式”章节 0：自动模式 1：半自动模式
10	WAIT	等待模式控制 0:无等待,如果上次转换数据没有及时读取,则可能出现Overrun 1：等待模式，在上次转换数据被读取前，不会启动下一次转换
9	CONT	连续转换模式使能 0：单次转换 1：连续转换
8	OVRM	Overrun 模式控制 0：当 overrun 发生时，保持上次数据，丢弃本次转换值 1：当 overrun 发生时，覆盖上次数据
7:4	EXTS	硬件触发源选择 0000：LUT0_TRGO 0001：LUT1_TRGO 0010：LUT2_TRGO 0011：RFU 0100：TAU01_TRGO 0101：RFU 0110：BSTIM16_TRGO 0111：LPTIM16_TRGO 1000：COMP1_TRGO 1001：COMP2_TRGO 1010：RTC_TRGO 1011：LUT3_TRGO 1100：TAU00_TRGO 1101：RFU 1110：RFU 1111：RFU *注意：必须在 TRGCFG=00 的情况下修改 EXTS 寄存器*
3	--	RFU：未实现，读为 0
2	SCANDIR	外部通道扫描顺序控制（共 26 个通道，实际只会采样被使能的通道） 0：前向扫描，ADC_IN0->ADC_IN18->内部通道 1：反向扫描，内部通道->ADC_IN18->ADC_IN0
1	--	RFU：未实现，读为 0
0	DMAEN	DMA 使能 0：禁止 DMA 1：使能 DMA

31.8.8 ADC 采样时间控制寄存器（ADC_SMTR）

名称	ADC_SMTR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SMTS2				SMTS1			
位权限	R/W-0000				R/W-0000			

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:4	SMTS2	快速采样时间控制 2（*ADC 工作时钟周期），适用于所有快速通道 **SMTSx** / **Sampling cycles** 0000 / 2 0001 / 4 0010 / 8 0011 / 12 0100 / 16 0101 / 32 0110 / 64 0111 / 80 1000 / 96 1001 / 128 1010 / 160 1011 / 192 1100 / 256 1101 / 320 1110 / 384 1111 / 512
3:0	SMTS1	慢速采样时间控制 1（*ADC 工作时钟周期），适用于所有慢速通道 **SMTSx** / **Sampling cycles** 0000 / 2 0001 / 4 0010 / 8 0011 / 12 0100 / 16 0101 / 32 0110 / 64 0111 / 80 1000 / 96 1001 / 128 1010 / 160 1011 / 192 1100 / 256 1101 / 320

Bit	助记符	功能描述	
		1110	384
		1111	512

31.8.9 ADC 通道控制寄存器（ADC_CHER）

名称	ADC_CHER							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	-	-	-	BUF4TST	AVREF	TS	VREFINT
位权限	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					ECH18	ECH17	ECH16
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ECH15	ECH14	ECH13	ECH12	ECH11	ECH10	ECH9	ECH8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30	--	RFU：未实现，读为 0
29	--	RFU：未实现，读为 0
28	--	RFU：未实现，读为 0
27	BUF4TST	内部测试 buffer 输出，写 1 使能
26	AVREF	快速基准源输出测量，写 1 使能
25	TS	温度传感器测量通道，写 1 使能
24	VREFINT	内部基准测量，写 1 使能
23:22	--	RFU：未实现，读为 0
21	--	RFU：未实现，读为 0
20	--	RFU：未实现，读为 0
19	--	RFU：未实现，读为 0
18	ECH18	ADC_IN0~18 测量通道，写 1 使能 其中，ADC_IN0~13 可以组成 7 对差分输入对，也可以作为 14 个单端输入使用 ADC_IN14~18 只能用作单端输入，其中 ADC_IN14 和 ADC_IN15 与 VREFP 和 VREFN 引脚复用
17	ECH17	
16	ECH16	
15	ECH15	
14	ECH14	
13	ECH13	
12	ECH12	
11	ECH11	

Bit	助记符	功能描述
10	ECH10	
9	ECH9	
8	ECH8	
7	ECH7	
6	ECH6	
5	ECH5	
4	ECH4	
3	ECH3	
2	ECH2	
1	ECH1	
0	ECH0	

31.8.10 ADC 通道差分控制寄存器（ADC_DCR）

名称	ADC_DCR							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		AINS					
位权限	U-0		R/W-00 0000					

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:0	AINS	差分输入对使能寄存器，用于配置 ADC_IN0~ADC_IN13 是否组成 7 组差分对。0 表示单端输入，1 表示差分输入。(Analog Input Select) AINS[0]：1 表示配置 ADC_IN0 和 ADC_IN7 为差分输入 AINS[1]：1 表示配置 ADC_IN1 和 ADC_IN8 为差分输入 AINS[2]：1 表示配置 ADC_IN2 和 ADC_IN9 为差分输入 AINS[3]：1 表示配置 ADC_IN3 和 ADC_IN10 为差分输入 AINS[4]：1 表示配置 ADC_IN4 和 ADC_IN11 为差分输入 AINS[5]：1 表示配置 ADC_IN5 和 ADC_IN12 为差分输入 AINS[6]：1 表示配置 ADC_IN6 和 ADC_IN13 为差分输入

31.8.11 ADC 数据寄存器（ADC_DR）

名称	ADC_DR

offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA[15:8]							
位权限	R-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA[7:0]							
位权限	R-0							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	DATA	ADC 转换结果 在没有使能过采样平均的情况下，结果为低 12bit；在使能过采样平均的情况下，结果为 12~16bit

31.8.12 AWD 阈值寄存器（ADC_HLTR）

名称	ADC_HLTR							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AWD_HT[15:8]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AWD_HT[7:0]							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AWD_LT[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AWD_LT[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:16	AWD_HT	AWD 监视高阈值，最长 16bit，软件根据实际所需位数设置，高位空余位数保持为 0
15:0	AWD_LT	AWD 监视低阈值，最长 16bit，软件根据实际所需位数设置，高位空余位数保持为 0

31.8.13 BUF4TST 配置寄存器(BUF4TST_CFGR)

名称	BUF4TST_CFGR
address	0x0x4001_9800

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							-
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	EN	BYPASS	SEL[4:0]				
位权限	R/W-0	R/W-0	R/W-0	R/W-0				

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	--	默认值为 0，不能改写为 1
7	--	默认值为 0，不能改写为 1
6	EN	BUFFER 使能 1 使能 0 禁止
5	BYPASS	1：模拟信号直接输出 0：模拟信号通过 buffer 输出
4:0	SEL	BUFFER 输入通道的选择 10100: ADCIN14 10101: ADCIN15 10110: ADCIN16 10111: ADCIN17 11000: ADCIN18 11001: ADCIN19 11010: ADCIN20 11011: ADCIN21 11100: ADCIN22

32 可编程胶合逻辑（PGL）

32.1 概述

可编程胶合逻辑（Programmable Glue Logic）是基于查找表（LUT）的简单可编程逻辑，其输入输出可以连接到芯片引脚、内部信号，实现一些简单的胶合逻辑，在一些应用中可以帮助系统设计减少PCB上的逻辑器件。

每个LUT包含4个输入，1个输出，1个真值表，可选的同步/滤波电路。用户通过对真值表编程，可以获得期望的组合逻辑输出表达式。每个输入信号都可以被单独屏蔽。

PGL的基本特性如下：

- 实现简单胶合逻辑，简化PCB设计
- 4个4输入查找表
- 可以通过真值表编程实现 AND,NAND,OR,NOR,XOR,NOT等逻辑表达式
- 时序同步或滤波
- 灵活的LUT输入选择：IO，内部信号，其他LUT输出
- 输出可以被连接到IO或其他外设触发

32.2 结构框图

LUT的结构如下图所示。

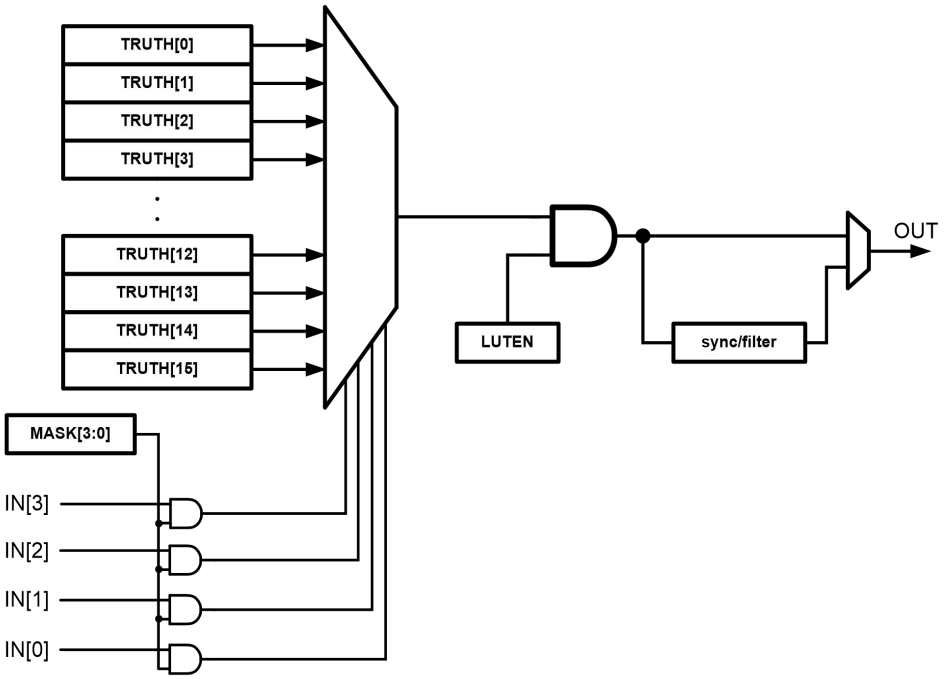


图 32-1LUT 结构框图

PGL模块共包含了4个LUT。

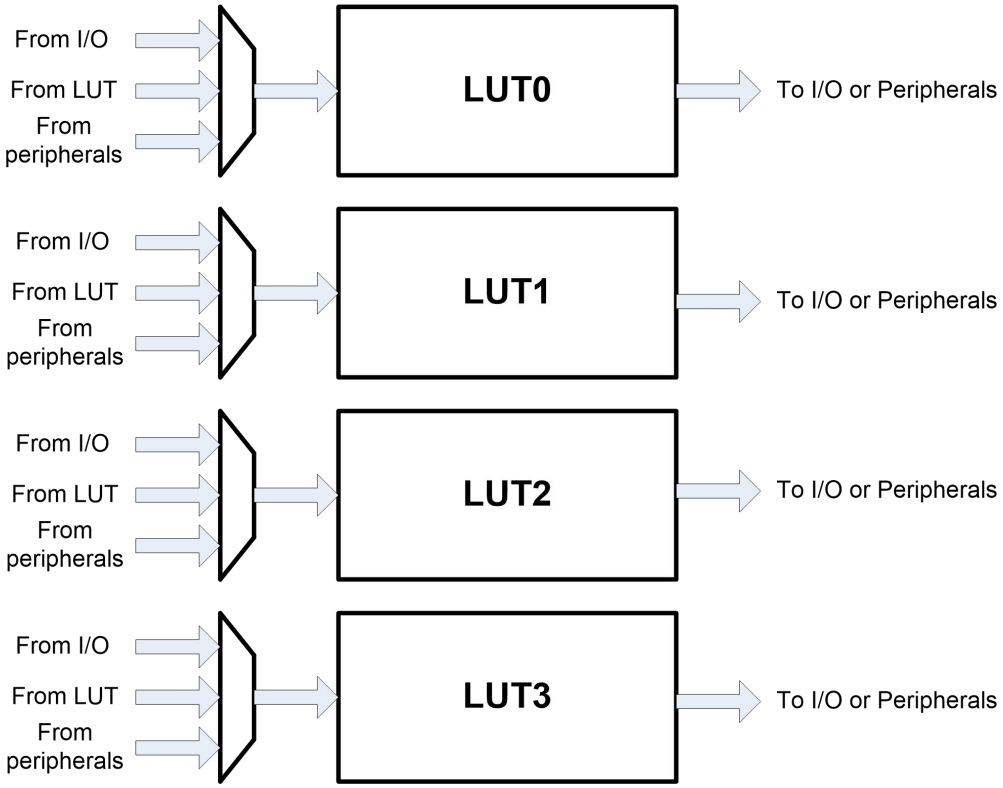


图 32-2 PGL 结构框图

32.3 引脚定义

PGL的输入输出引脚与普通GPIO复用，使用PGL功能需要将对应IO配置为GPIO输入或者输入。

引脚	类型	说明
PC2	GPIO 输入	LUT0 输入 0
PC3		LUT0 输入 1
PC4		LUT0 输入 2
PC5		LUT0 输入 3
PC6		LUT1 输入 0
PC7		LUT1 输入 1
PC8		LUT1 输入 2
PC9		LUT1 输入 3
PB7		LUT2 输入 0
PB8		LUT2 输入 1
PB9		LUT2 输入 2
PB10		LUT2 输入 3
PB11		LUT3 输入 0
PB12		LUT3 输入 1
PB13		LUT3 输入 2
PB14		LUT3 输入 3
PC13	额外数字功能	LUT0 输出
PC14		LUT1 输出
PA1		LUT2 输出
PA0		LUT3 输出

表 32-1 PGL 相关引脚

当使用LUT引脚输入时，需将对应引脚配置为GPIO输入，清除输入的MASK寄存器，并使能LUT。

当使用LUT引脚输出时，需将对应引脚配置为数字功能，并通过数字功能选择寄存器将引脚配置为LUT输出，然后使能LUT。

32.4 功能描述

32.4.1 LUT 真值表

通过输入状态组合查询LUT真值表中的数据，可以获得想要的组合逻辑功能。

举例来说，希望通过LUT0实现一个2输入NAND逻辑功能，可以采用以下配置：

- 将LUT0的输入2和输入3屏蔽为0
- LUT0的输入0和输入1连接到GPIO
- LUT0真值表寄存器低4bit写入0111

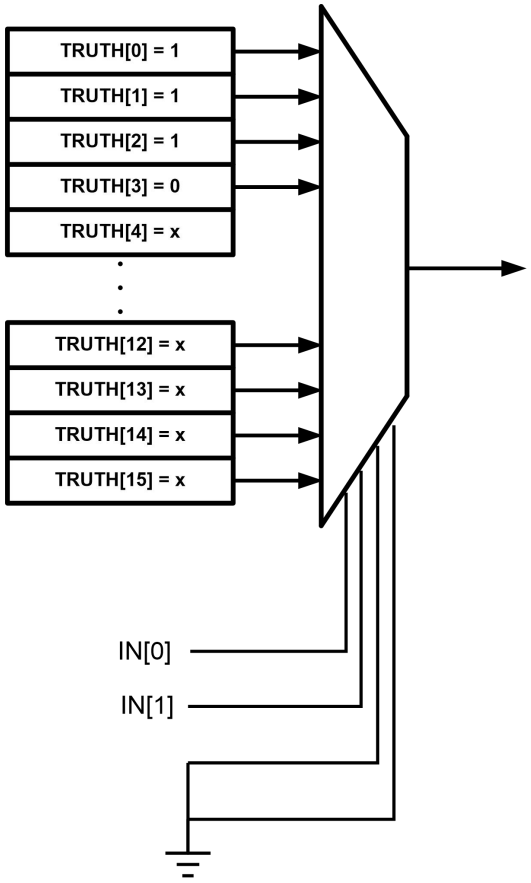


图 32-3 LUT 实现 2 输入 NAND 示意图

32.4.2 LUT 输入

每个LUT有4个输入，每个输入都可以来自于引脚、芯片外设、或其他LUT输出。

通过INSEL寄存器，可以选择LUT输入信号源，详情请参见寄存器章节。

32.4.3 LUT 输出

LUT的输出可以连接到引脚、或其他LUT输入。如下表所示。

LUT 输出	引脚	LUT 输入
LUT0 输出	PE13	LUT1,LUT2,LUT3
LUT1 输出	PC14	LUT2,LUT3
LUT2 输出	PA1	LUT3
LUT3 输出	PA0	-

表 32-2 LUT 输出连接

32.4.4 滤波和采样

LUT支持对输出的滤波和同步采样，其结构示意图如下。

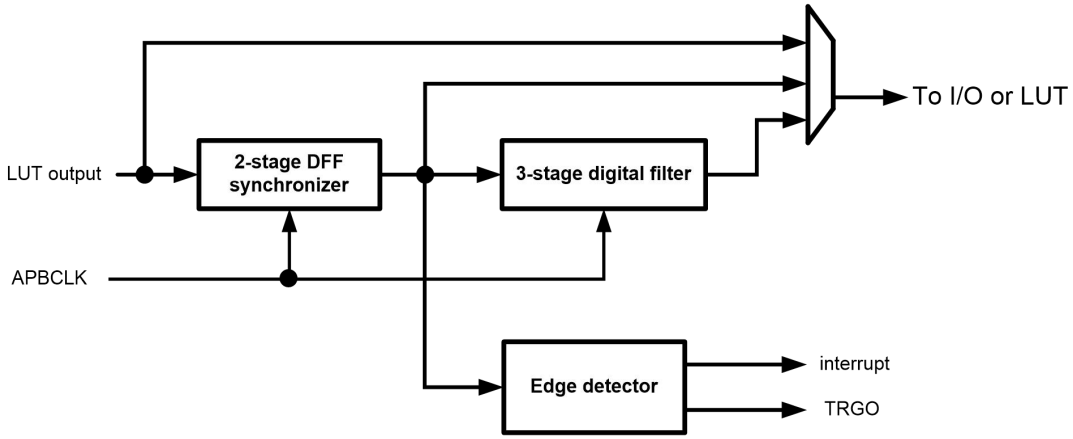


图 32-4 输出滤波和采样

数字滤波采用以下方法，当APBCLK连续采样到3个相同电平，则认为合法电平，否则滤波输出不会改变。

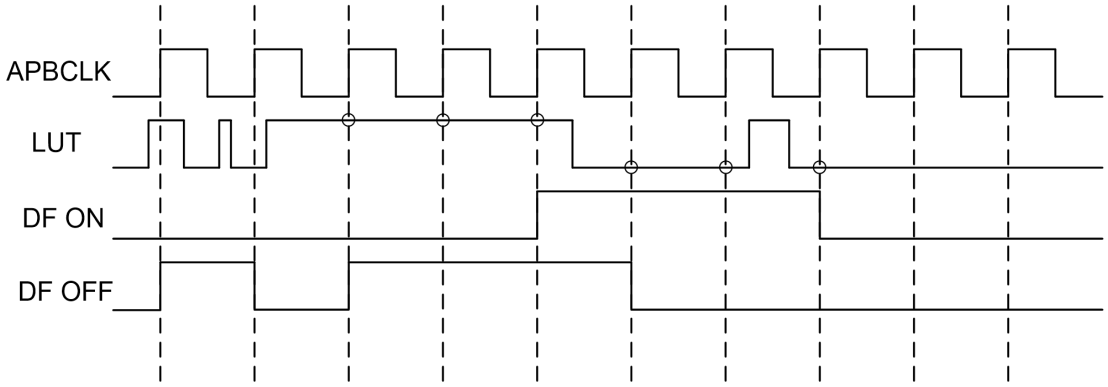


图 32-5 数字滤波

根据寄存器配置，应用可以选择LUT的输出为组合逻辑输出、APBCLK同步采样后的输出、或者为采样后经过数字滤波的输出。

经过同步采样后的信号，经过边沿产生电路，可以产生1个APBCLK周期宽度的上升沿或下降沿脉冲信号。这个同步脉冲信号可用于其它外设的trigger信号，或者产生中断。

32.4.5 中断和触发

LUT输出经过采样和边沿选择后，可以产生中断标志或者输出触发信号。

必须使能LUT外设时钟，才能产生中断或触发。

32.4.6 低功耗模式

PGL可以在低功耗模式下使用。但是要注意滤波和采样功能需要APB时钟，因此无法在休眠模式下使用。

如果只是使用LUT的组合逻辑输出功能，则应用可以在配置完寄存器之后关闭PGL外设模块时钟，以节省功耗。

32.5 寄存器

模块起始地址：0x4001_3800

offset 地址	名称	符号
0x00	PGL 控制寄存器	PGL_CR
0x04	PGL 配置寄存器 0	PGL_CFGR0
0x08	PGL 配置寄存器 1	PGL_CFGR1
0x0C	PGL 配置寄存器 2	PGL_CFGR2
0x10	PGL 配置寄存器 3	PGL_CFGR3
0x14	PGL 中断使能寄存器	PGL_IER
0x18	PGL 中断标志寄存器	PGL_ISR
0x1C	LUT0 真值表寄存器	PGL_LUT0
0x20	LUT1 真值表寄存器	PGL_LUT1
0x24	LUT2 真值表寄存器	PGL_LUT2
0x28	LUT3 真值表寄存器	PGL_LUT3

32.5.1 PGL 控制寄存器（PGL_CR）

名称	PGL_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTEN[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTEN	LUT 使能寄存器，分别控制 LUT0~3 1：使能 LUT[x] 0：关闭 LUT[x]

32.5.2 PGL 配置寄存器 0（PGL_CFGR0）

名称	PGL_CFGR0
offset	0x04

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT0 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT0 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT0 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT0 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT0 输入 1 选择寄存器 00：GPIO 01：RFU 10：COMP2_OUT 11：TAU00_TRGO
9:8	IN0SEL	LUT0 输入 0 选择寄存器 00：GPIO 01：RFU 10：COMP1_OUT 11：TAU01_TRGO
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT0 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

32.5.3 PGL 配置寄存器 1（PGL_CFGR1）

名称	PGL_CFGR1							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT1 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT1 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT1 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT1 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT1 输入 1 选择寄存器 00：GPIO 01：RFU 10：COMP2_OUT 11：TAU02_TRGO
9:8	IN0SEL	LUT1 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：TAU03_TRGO
7:4	--	RFU：未实现，读为 0

Bit	助记符	功能描述
3:0	MASK	LUT1 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

32.5.4 PGL 配置寄存器 2（PGL_CFGR2）

名称	PGL_CFGR2							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT2 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT2 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT2 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT2 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT2 输入 1 选择寄存器 00：GPIO 01：LUT1 output 10：COMP2_OUT

Bit	助记符	功能描述
		11：TAU04_TRGO
9:8	IN0SEL	LUT2 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：TAU05_TRGO
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT2 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

32.5.5 PGL 配置寄存器 3（PGL_CFGR3）

名称	PGL_CFGR3							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				EDGESEL		OUTSEL	
位权限	U-0				R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				MASK[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:18	EDGESEL	LUT3 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	LUT3 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	LUT3 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT3 输入 2 选择寄存器 00：GPIO

Bit	助记符	功能描述
		01：LUT2 output 10：RFU 11：RFU
11:10	IN1SEL	LUT3 输入 1 选择寄存器 00：GPIO 01：LUT1 output 10：COMP2_OUT 11：TAU06_TRGO
9:8	IN0SEL	LUT3 输入 0 选择寄存器 00：GPIO 01：LUT0 output 10：COMP1_OUT 11：TAU07_TRGO
7:4	--	RFU：未实现，读为 0
3:0	MASK	LUT3 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

32.5.6 PGL 中断使能寄存器（PGL_IER）

名称	PGL_IER							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTIE[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTIE	LUT 中断使能寄存器，分别控制 LUT0~3 1：使能 LUT[x]中断 0：禁止 LUT[x]中断

32.5.7 PGL 中断标志寄存器（PGL_ISR）

名称	PGL_ISR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LUTIF[3:0]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	LUTIF	LUT 中断标志寄存器，硬件置位，软件写 1 清零

32.5.8 LUT0 真值表寄存器（PGL_LUT0）

名称	PGL_LUT0							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT0 真值表，软件可以写入任意值以获得需要的逻辑组合

32.5.9 LUT1 真值表寄存器（PGL_LUT1）

名称	PGL_LUT1							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT1 真值表，软件可以写入任意值以获得需要的逻辑组合

32.5.10 LUT2 真值表寄存器（PGL_LUT2）

名称	PGL_LUT2							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT2 真值表，软件可以写入任意值以获得需要的逻辑组合

32.5.11 LUT3 真值表寄存器（PGL_LUT3）

名称	PGL_LUT3							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	TRUTH	LUT3 真值表，软件可以写入任意值以获得需要的逻辑组合

33 电容触摸按键（TSI）

33.1 概述

TSI模块用于电容式触摸检测和接近检测功能，其主要特性如下。

- 14个独立的按键传感器通道
- 高电容检测灵敏度
- 支持的按键电容范围：5~50pF
- 噪声抑制
  - Sigma-Delta电容检测
  - 多级可配置数字滤波器滤除高频噪声
  - 低辐射发散
- 基准线自动调校，补偿外界环境引起的按键电容缓慢变化
- 低功耗检测
  - 接近检测
  - 多扫描模式自动切换
  - 扫描单个按键的平均功耗：5uA @120ms扫描周期[1]
- 按键互斥功能
- 防水功能
  - 支持湿手触摸
  - 屏蔽电极驱动
  - 保护传感器

[1] 按键电容Cp=15pF，最高灵敏度，VDD=3.3V

33.2 工作原理

33.2.1 电容检测

按键触摸功能基于电容检测的原理实现。当手指接近触摸按键传感器时，引起寄生电容变化，芯片通过检测电容变化实现按键事件判决。

33.2.2 信噪比（SNR）

在触摸检测应用中，信号定义为按键触发（ON）与非触发（OFF）状态下采样值的变化量。

非触发状态下传感器寄生电容的采样值可以表示为：

$$X(t) = X0 + N0(t)$$

- X0是X(t)的均值
- N0(t)是非触发状态下的噪声成分

触发状态下的传感器采样值表示为：

$$X(t) = X1 + N1(t)$$

- X1是X(t)的均值
- N1(t)是触发状态下的噪声成分

我们定义X0为传感器采样值的基线（baseline），表示传感器非触发条件下的平均输出，而X1和X0的差值定义为信号：

$$S = X1 - X0$$

噪声成分N0和N1是相似的但是并不相同，比如当手指触摸传感器时，由于人体对于50Hz工频信号表现为天线，因此传感器会通过手指耦合到更多高频噪声，而这些高频噪声是没有手指接近时不存在的。我们定义系统噪声为按键OFF状态下的峰值噪声：

$$N = \max(N0(t)) = \max(X0(t)) - \min(X0(t))$$

因此，系统信噪比定义为SDR=S/N

为了实现可靠的按键检测，通常要求系统SNR达到5:1以上。

33.2.3 按键触摸判定

当处于OFF状态的按键信号强度超过触摸阈值（touch_thrsh+hys）后，判定为按键触摸事件；当处于ON状态的按键信号强度低于释放阈值（touch_thrsh-hys）后，判定为按键释放事件。防弹跳（debounce）寄存器定义了连续多次超过触摸阈值才能被判定为按键触摸事件，以及连续多次低于释放阈值才能被判定为按键释放事件。

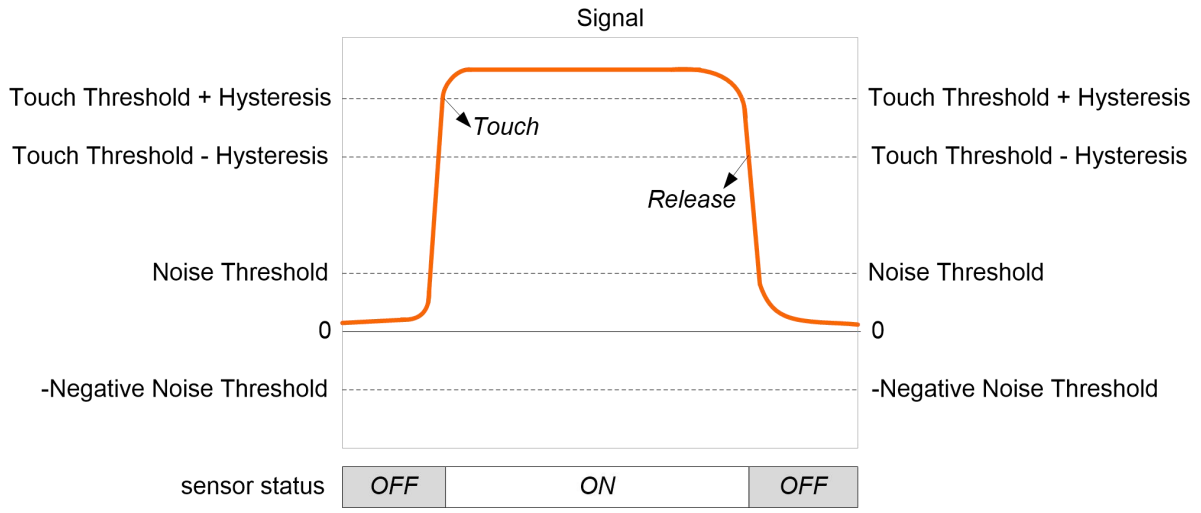


图 33-1 按键触摸与释放

注意到按键判定是基于信噪比，而信噪比是跟baseline相关的，在实际使用中受环境和噪声因素影响，必须要考虑baseline随时间变化的情况，因此应用中需要实现baseline跟踪算法。

33.2.4 接近检测

14个按键通道中的任意一个可以配置为接近检测通道，接近检测用于检测远距离接近事件，通常用于接近唤醒功能。接近检测传感器与按键触摸检测相独立。

33.2.5 按键超时检测

按键超时检测功能用于防止因导体长时间放置或靠近传感器而导致传感器一直处于ON状态。当传感器处于ON状态，超时计数器工作；当处于OFF状态，超时计数器停止工作，并清零。一旦计数值达到阈值，认为检测到超时事件，传感器状态强制切换为OFF状态，同时产生超时中断通知应用程序。

33.2.6 按键互斥

当开启按键互斥功能时，TSI最多同时只允许一个按键被判定为有效的触摸。

33.3 通道扫描

33.3.1 扫描过程

TSI最大支持14个按键通道和1个接近检测通道，TSI按照编号从小到大的规则逐个扫描使能的按键

通道（不使能的通道不被扫描），最后扫描接近检测通道。

ADC分辨率决定了每个通道单次采样时间，滤波器的配置参数和SMPL_SIZE寄存器决定了每个通道在一轮扫描中被采样的次数。比如某个通道使能了16阶均值滤波的情况下，对此通道连续采样16次后输出一个结果，实际采样次数由 SMPL_SIZE * AVG_ORGER决定。在没有使能滤波器的情况下，实际采样次数就是SMPL_SIZE。

每个通道都有独立的使能控制和滤波器参数配置寄存器。下图是一个扫描周期的例子，每个通道进行4次采样。总的扫描周期是所有通道扫描时间加上IDLE时间，总时间由SCAN_PERIOD寄存器定义，默认周期120ms，最短周期20ms，最长周期1s。较长的扫描周期可以降低系统平均功耗，但是也会延长按键反应时间。实际应用中需要合理平衡系统响应速度和整体功耗。

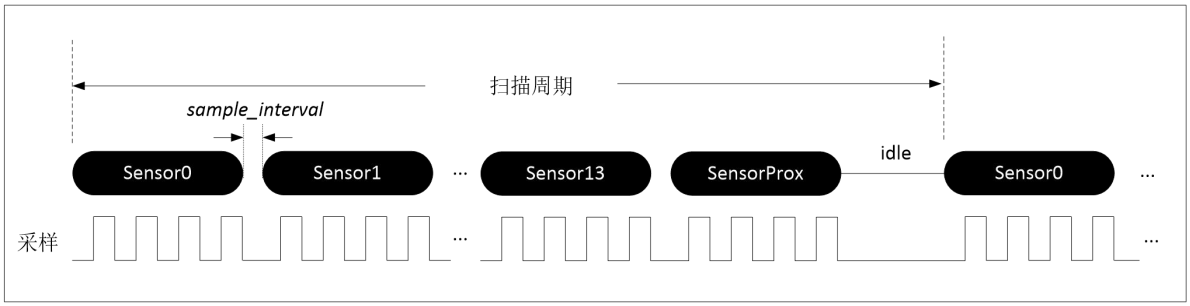


图 33-2 通道扫描周期

33.3.2 扫描模式

TSI支持以下扫描模式：

- 软件触发的单次扫描
- 软件触发的连续扫描
- 硬件触发的单次扫描
- 硬件触发的连续扫描

对于单次扫描（CONT=0），TSI触发后对所有使能通道进行一轮扫描，扫描结束后自动停止，等待下一次触发事件的到来。

对于连续扫描（CONT=1），TSI触发后循环扫描所有使能通道，直到软件关闭SEN。

33.3.3 扫描触发

按键扫描可以通过软件触发或者外设trigger信号触发。

如果使用软件触发，START置位后TSI开始扫描，直到扫描结束后START自动清零。

如果使用硬件外设触发，START寄存器无效。SEN置位后TSI模块等待选中的外设trigger信号到来。

由于外设trigger信号是系统时钟域的，使用硬件触发时必须保持TSI的总线时钟使能。

TSI模块支持以下触发信号输入：

- RTC TRGO
- ATIM TRGO
- LPTIM16 TRGO
- BSTIM16 TRGO
- TAU TRGOx

通过定时器触发可以实现灵活定时的周期性唤醒扫描，并在扫描结束后再唤醒CPU。

33.4 实现方案

33.4.1 时钟方案

模拟电路的时钟使用以下电路产生，其中switch clock可以经由PRS（伪随机序列发生器）产生，以实现频谱展宽，降低EMI效应。

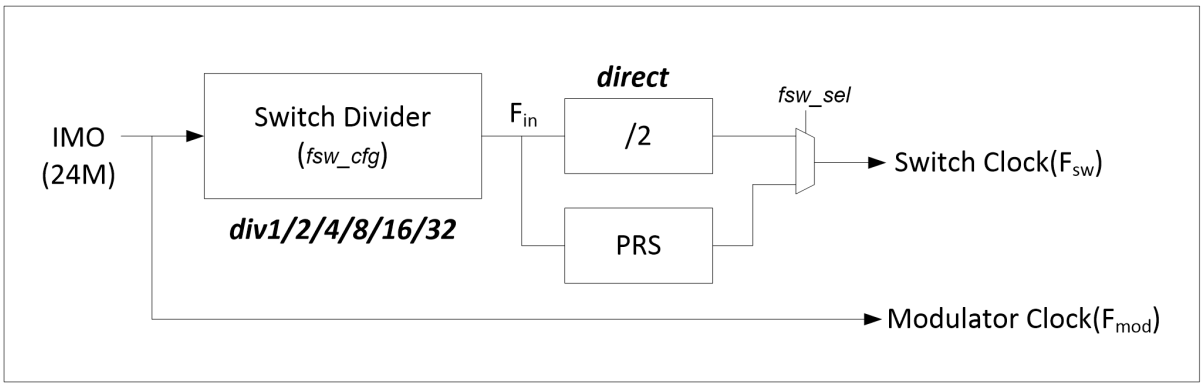


图 33-3 开关时钟与调制器时钟

注意，IMO推荐使用24MHz，在FM33FR0中工作时钟是APBCLK，可以有多种来源和不同频率，系统软件需要根据TSI实际的工作时钟频率选择合适的Fsw分频比。

当使用PRS时，平均的开关频率可以表示为：

$$F_{SW\_AVERAGE} = \frac{F_{IN}}{4}$$

在配置TSI工作时钟时，需注意满足以下条件：

- 开关频率F_{SW}必须保证按键电容C_S的“满充满放”，即

  $$T_{SW} \geq 10 \cdot R_{SerialsTotal} \cdot C_S$$

- 如果是能PRS时钟，则开关频率是变化的，其最高频率需满足上述条件
- 一次完整的按键扫描周期必须大于等于PRS周期，即

  $$\frac{2^{N\_PRS}-1}{F_{in}} \leq \frac{2^{N}-1}{F_{mod}}$$

其中N_PRS=8，即PRS循环周期为255个Fin时钟。N是sigma-delta调制器分辨率，即采样值数据位宽。Fmod是调制器工作时钟。

33.4.2 数字滤波器

TSI数字滤波器整体架构如下图所示。

包括按键触摸滤波器组和接近检测滤波器组。每组滤波器包含primary FIR滤波器（硬件均值滤波）和后续的secondary滤波器（软件IIR、中值滤波、抖动滤波）。所有14个按键通道复用同一组FIR滤波器，secondary滤波器为软件实现的辅助功能。FIR主滤波器为均值滤波，软件也可以配置是否使能。

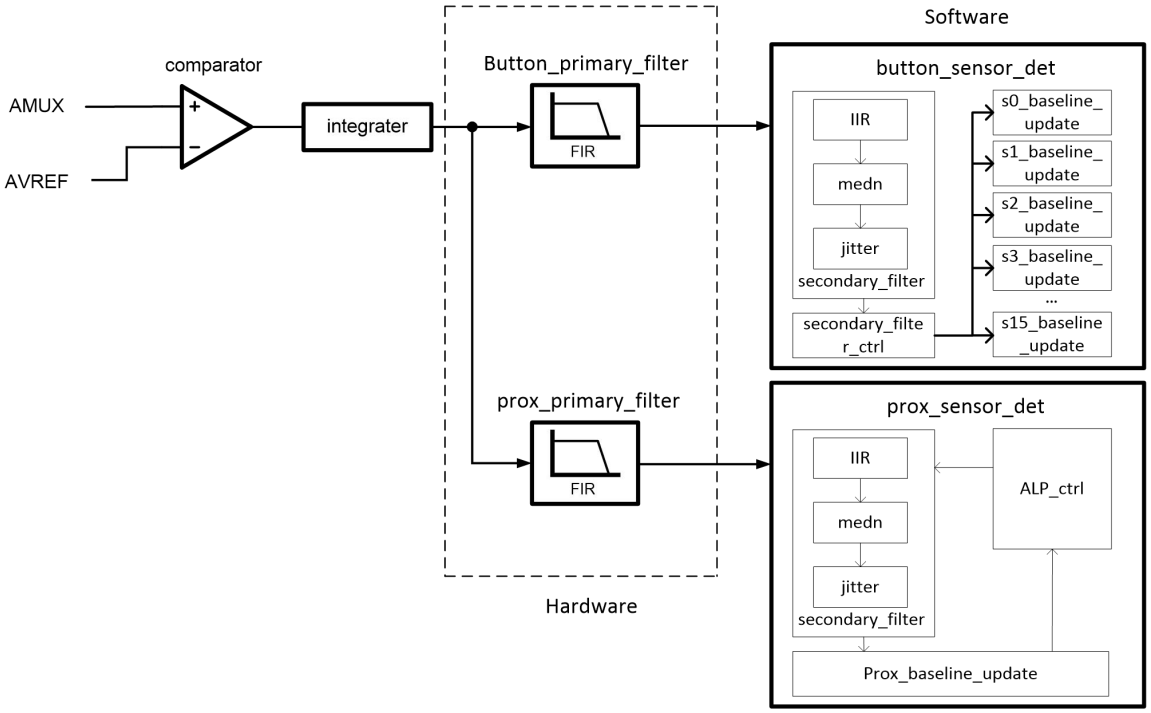


图 33-4 数字滤波器整体架构

复旦微电子提供完整的软件滤波器库函数供用户使用。

33.4.2.1 FIR 滤波器

Primary滤波器是一个FIR均值滤波器，实现方案只是对输入raw data进行均值滤波，平均阶数可以选择为2、4、8、16，传输函数为：

$$y(i) = [x(i)+x(i-1)+\ldots+x(i-n+1)]/n$$

其中n=2/4/8/16

下图为n=16的均值滤波输入输出波形对比。

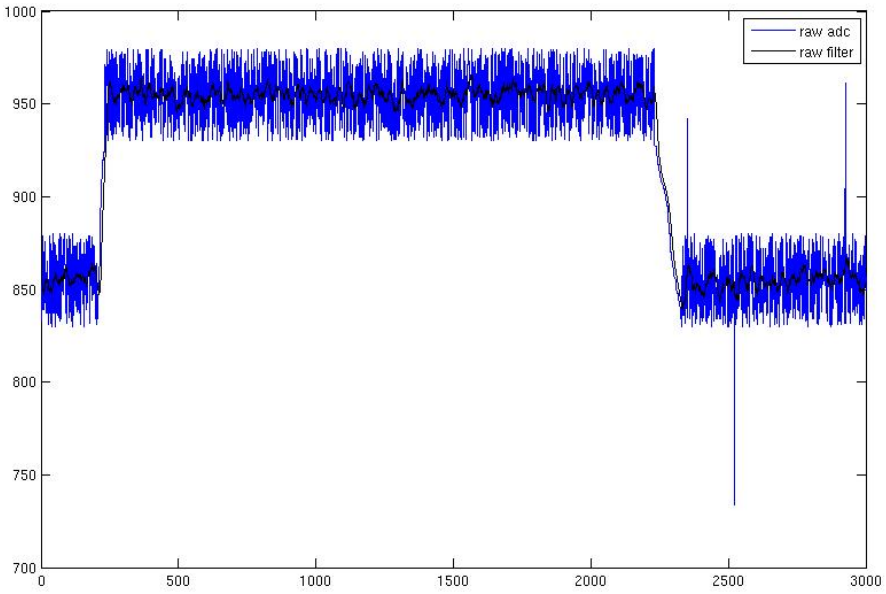


图 33-516 阶均值滤波

33.4.3 Baseline 跟踪

所谓baseline是指没有按键触发的情况下TSI读取到的传感器采样值。随着环境变化或受系统噪声影响，baseline可能会出现缓慢漂移，如果系统不能自适应的跟踪这种漂移，则在一段时间后可能导致误触发问题，或者降低系统灵敏度。为此，需要实现一种baseline自跟踪的方法，以提高系统鲁棒性。

在每个扫描周期，传感器通道得到一个原始数据采样值，并根据当前采样值与上一个扫描周期的基准线Baseline之间的差值Delta（即信号强度）所处的不同区间，来选择不同的更新策略得到当前扫描周期的Baseline。通过设置噪声阈值（Noise Threshold）、触摸阈值（Touch Threshold）和迟滞（Hysteresis）、负噪声阈值（Negative Noise Threshold）来划分信号强度的几个区间，如下图所示。

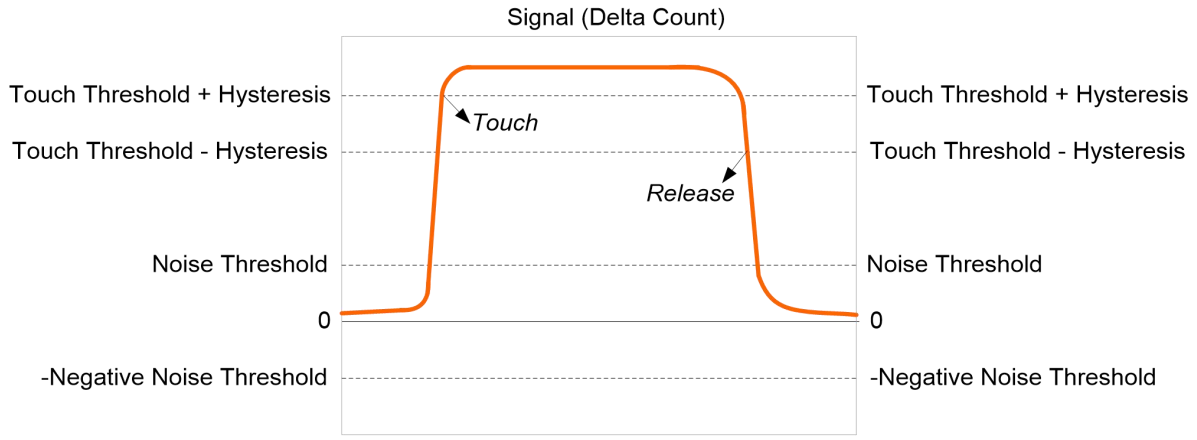


图 33-6 信号与噪声阈值

相关基线跟踪算法库函数请咨询复旦微电子获取。

33.4.4 中断和状态标志

TSI模块在扫描过程中会产生多种中断事件：

- 采样值更新事件
- 通道扫描完成事件
- 序列扫描完成事件

在每次采样完成后，硬件更新采样值数值到TSI_RAW_CNTR寄存器中，同时置位RCUF标志寄存器，在RCU_IE=1的情况下同时产生中断，通知软件读取新的采样值。如果使能了滤波器功能，则TSI_RAW_CNTR寄存器中保存的是滤波后的数据，更新频率降低为（采样频率/滤波器阶数），RUIF标志寄存器在滤波器更新输出时置位。RCUF标志由软件写1清零，或者软件/DMA读取TSI_RAW_CNTR时自动清零。

每个通道的采样次数由SMPL_SIZE寄存器配置。当完成SMPL_SIZE次采样后，当前通道扫描结束，CHF标志寄存器置位，在CH_IE=1的情况下同时产生中断。如果使能了滤波器，则SMPL_SIZE定义的是滤波器输出个数。CHF标志由软件写1清零。

当所有被使能通道都扫描完成后，SEQF标志寄存器置位，表示一轮扫描结束，在SEQ_IE=1的情况下同时产生中断。SEQF标志由软件写1清零。

下图是使能4个按键通道，每个通道采样2次的中断标志示意图：

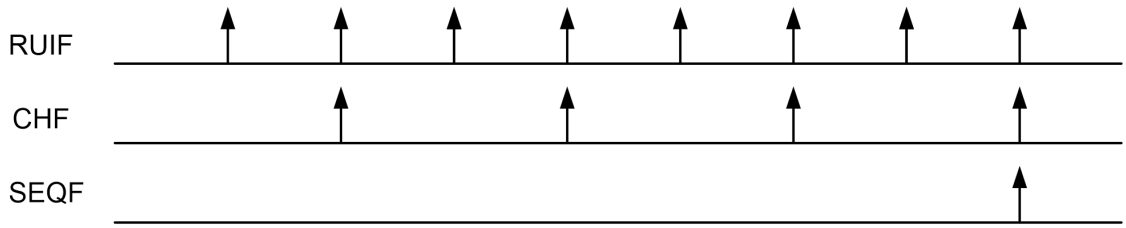


图 33-7SMPL_SIZE=2，扫描 4 个通道

33.4.5 DMA 支持

应用中可以使用DMA将TSI输出的采样值或者FIR滤波结果搬运到RAM指定地址。由于采样值和滤波输出最大都是16bit结果，因此DMA搬运采用half-word读写，每个结果占用半字RAM空间。

由于按键扫描通道比较多，DMA设计一个TSI专用通道，用于时分复用传输所有TSI通道的数据。TSI专用通道共可以提供15个RAM指针，分别用于S0~S13和Prox通道的数据传输，在不使能FIR功能时，

DMA搬运的是采样值，在使能FIR功能时，DMA搬运的是滤波后的结果。

TSI的DMA通道示意图如下：

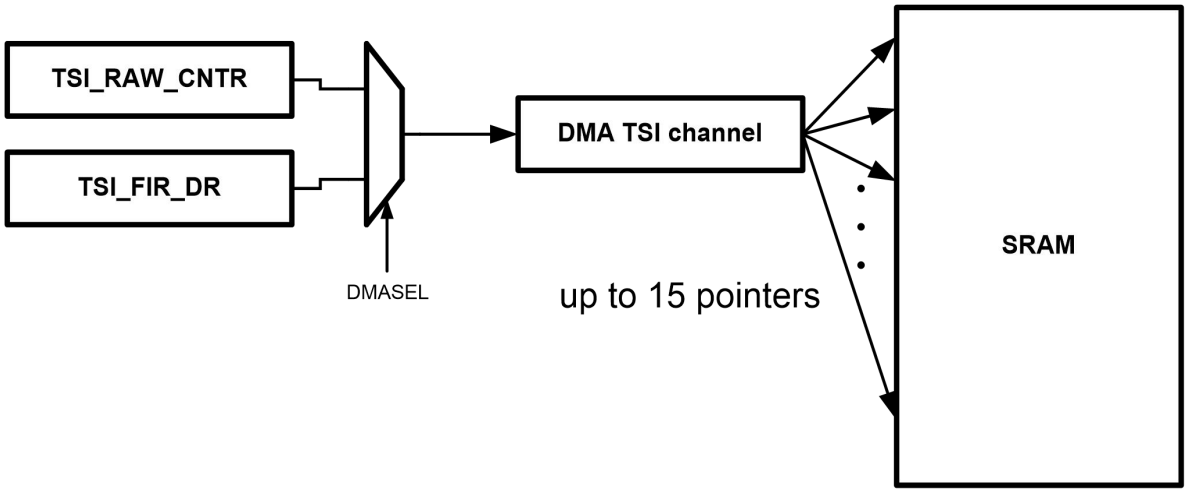


图 33-8TSI DMA 通道示意图

通过15个独立的RAM指针，应用可以将不同通道的采样值或滤波器输出搬运到RAM不同地址段，便于后续软件进一步处理。

每个通道保存的数据长度由SMPL_SIZE寄存器决定，配合循环模式使能，TSI的扫描数据会在指定存储空间内循环存储。

下图是一个示例。扫描通道指针可以指向RAM中任意地址，SMPL_SIZE=N。如果滤波器关闭，则在一轮扫描中，每个传感器被采样N次，搬运N个采样值到RAM中，假设软件使能了14个通道，则总共搬运N*14次。本轮扫描结束后，DMA通道自动关闭，并产生传输完成中断。如果循环模式使能，则RAM指针被恢复成寄存器配置值，当新的一轮扫描开始后，RAM中上一轮扫描的数据将被覆盖；如果循环模式没有使能，则RAM指针保持不变，新一轮扫描的数据将存放在上一轮数据的后面。

注意，如果滤波器使能，即被搬运的是滤波后的数据，考虑滤波器阶数M，DMA每搬运一次数据，TSI要采样M次，那么实际TSI对每个传感器通道的采样次数将是M*N。

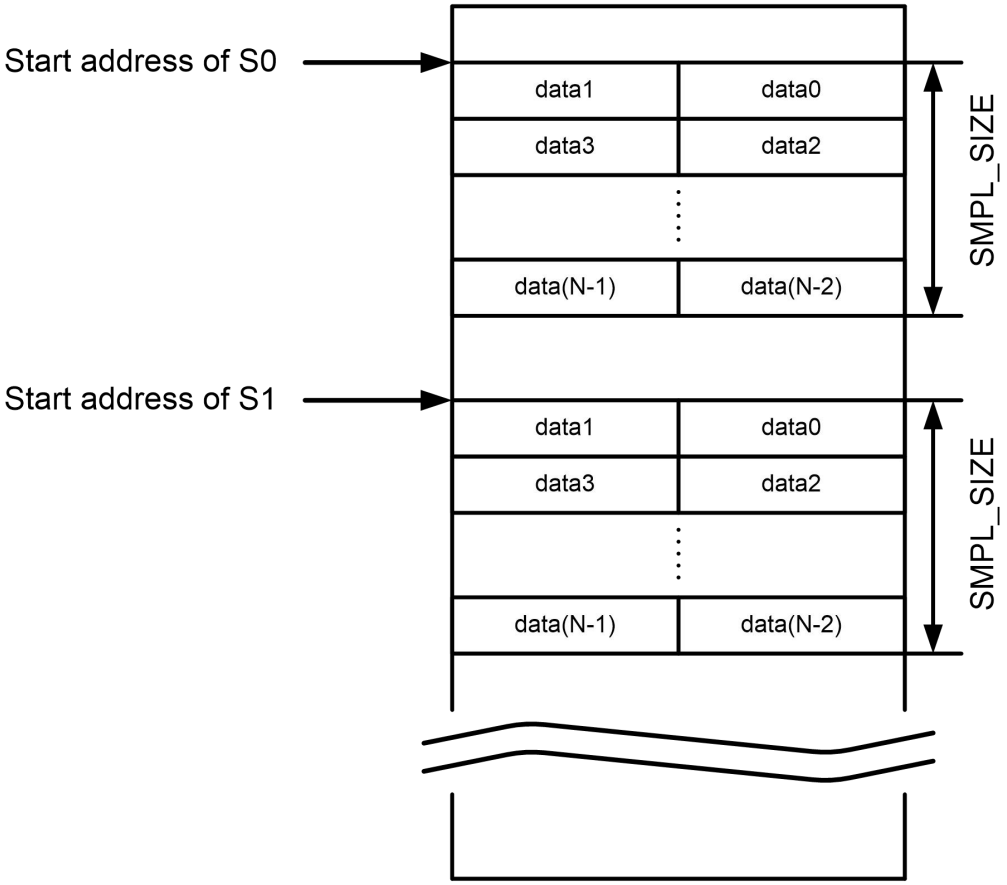


图 33-9TSI 数据存储示意图

详情参见DMA章节。

33.5 寄存器

模块起始地址：0x40010000

offset 地址	名称	符号
0x00	按键中断标志寄存器	TSI_ISR
0x04	按键中断使能寄存器	TSI_IER
0x08	扫描状态寄存器	TSI_SSR
0x0C	扫描配置寄存器	TSI_CFGR
0x10	扫描控制寄存器	TSI_CR
0x14	按键通道配置寄存器	TSI_CH_CFGR
0x18	Raw Count 寄存器	TSI_RAW_CNTR
0x1C	时钟和周期控制寄存器	TSI_CCR
0x20	DMA 控制寄存器	TSI_DMACR
0x24	按键 0 通道扫描配置寄存器	TSI_CH0_CFGR
0x28	按键 1 通道扫描配置寄存器	TSI_CH1_CFGR
0x2C	按键 2 通道扫描配置寄存器	TSI_CH2_CFGR
0x30	按键 3 通道扫描配置寄存器	TSI_CH3_CFGR
0x34	按键 4 通道扫描配置寄存器	TSI_CH4_CFGR
0x38	按键 5 通道扫描配置寄存器	TSI_CH5_CFGR
0x3C	按键 6 通道扫描配置寄存器	TSI_CH6_CFGR
0x40	按键 7 通道扫描配置寄存器	TSI_CH7_CFGR
0x44	按键 8 通道扫描配置寄存器	TSI_CH8_CFGR
0x48	按键 9 通道扫描配置寄存器	TSI_CH9_CFGR
0x4C	按键 10 通道扫描配置寄存器	TSI_CH10_CFGR
0x50	按键 11 通道扫描配置寄存器	TSI_CH11_CFGR
0x54	按键 12 通道扫描配置寄存器	TSI_CH12_CFGR
0x58	按键 13 通道扫描配置寄存器	TSI_CH13_CFGR
0x5C	接近通道扫描配置寄存器	TSI_PROX_CFGR
0x60	IDAC 调校寄存器	TSI_IDAC_TR
0x64	IDAC 模拟控制寄存器	TSI_ANA_CR
0x68	TSC_DFT 寄存器	TSI_TEST
0x6C	ANA TEST 寄存器	TSI_ANA_TEST

33.5.1 按键中断标志寄存器（TSI_ISR）

名称	TSI_ISR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					SEQF	CHF	RCUF
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	SEQF	扫描序列完成 (Sequence Finished)，所有通道完成一轮扫描后置位，软件写 1 清零
1	CHF	当前通道扫描完成 (Channel Finished)，硬件置位，软件写 1 清零
0	RCUF	Raw count 更新标志寄存器 每完成一次采样，产生新的 raw count 时置位，软件写 1 清零，或者软件读取 raw count 寄存器时自动清零

33.5.2 按键中断使能寄存器（TSI_IER）

名称	TSI_IER							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					SEQ_IE	CH_IE	RCU_IE
位权限	U-0					R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	SEQ_IE	扫描序列完成中断使能 0：禁止中断 1：使能中断
1	CH_IE	当前通道扫描完成中断使能 0：禁止中断 1：使能中断
0	RCU_IE	Raw count 更新中断使能 0：禁止中断 1：使能中断

33.5.3 扫描状态寄存器（TSI_SSR）

名称	TSI_SSR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SSR			
位权限	U-0				R-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	SSR	扫描状态寄存器（Scan Status Register），表示当前正在扫描的通道 0000：没有通道正在扫描 0001：正在扫描通道 0 0010：正在扫描通道 1 …… 1110：正在扫描通道 13 1111：正在扫描接近检测

33.5.4 扫描配置寄存器（TSI_CFGR）

名称	TSI_CFGR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SMPL_SIZE				-			CONT
位权限	R/W-0000				U-0			R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PROX_AVG_EN	-			PROX_AVG_ORDER			
位权限	R/W-0	U-0			R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	HWTRG_SEL				BUTTON_AVG_ORDER			
位权限	R/W-0000				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUTTON_AVG_EN	STOP_MOD	TMOD	SHIELD_EN	GUARD_EN	PROX_MODE	PROX_EN	SEN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:28	SMPL_SIZE	每个通道的采样次数配置。在使能滤波器的情况下，SMPL_SIZE 指的是滤波输出的个数；在没有使能滤波器的情况下，SMPL_SIZE 是 raw count 输出个数。 0000：1 次 0001：2 次 …… 1111：16 次
27:25	--	RFU：未实现，读为 0
24	CONT	连续扫描模式 0：单次扫描 1：连续扫描
23	PROX_AVG_EN	接近检测硬件均值滤波器使能 0：关闭硬件均值滤波器 1：使能硬件均值滤波器
22:20	--	RFU：未实现，读为 0
19:16	PROX_AVG_ORDER	接近检测硬件均值滤波器阶数控制，仅 PROX_AVG_EN=1 时有效 0000：2 阶 0001：4 阶 0010：8 阶 0011：16 阶 Others：RFU
15:12	HWTRG_SEL	硬件触发信号选择 0000：RTC TRGO 0001：ATIM TRGO 0010：LPTIM16 TRGO 0011：BSTIM16 TRGO 0100：TAU00 TRGO 0101：TAU01 TRGO 0110：TAU02 TRGO 0111：TAU03 TRGO 1000：TAU04 TRGO 1001：TAU05 TRGO 1010：TAU06 TRGO 1011：TAU07 TRGO 1100：TAU10 TRGO 1101：TAU11 TRGO 1110：TAU12 TRGO 1111：TAU13 TRGO
11:8	BUTTON_AVG_ORDER	按键检测硬件均值滤波器阶数控制，仅 BUTTON_AVG_EN=1 时有效 0000：2 阶 0001：4 阶 0010：8 阶 0011：16 阶 Others：RFU
7	BUTTON_AVG_EN	按键检测硬件均值滤波器使能 0：关闭硬件均值滤波器 1：使能硬件均值滤波器

Bit	助记符	功能描述
		注意：硬件均值滤波器使能后对所有按键通道有效
6	STOP_MOD	扫描停止模式 0：即刻停止，软件清零 SEN 后立即停止扫描 1：空闲停止，软件清零 SEN 后等待 TSI 完成当前通道扫描后再停止
5	TMOD	触发模式选择 (Trigger mode) 0：软件触发，软件写 START 寄存器启动扫描 1：硬件触发，外设触发事件启动扫描
4	SHIELD_EN	屏蔽电极功能使能，用于防水（小水滴）功能 0：不使能 1：使能
3	GUARD_EN	保护传感器的使能。注意，保护传感器与 S2 按键传感器复用。 0：不使能 1：使能
2	PROX_MODE	接近传感器工作模式 0：使能接近传感器时，不会停止按键通道扫描 1：使能接近传感器时，停止按键通道扫描，即被使能的按键通道也不会被扫描
1	PROX_EN	接近传感器的使能 0：不使能 1：使能 接近传感器的通道由 TSI_CH_CFGR 寄存器配置
0	SEN	扫描模块使能 0：关闭 TSI 1：使能 TSI

33.5.5 扫描控制寄存器（TSI_CR）

名称	TSI_CR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							START
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	START	软件写 1 启动扫描，所有使能通道扫描结束后硬件自动清零

Bit	助记符	功能描述
		仅 SEN=1 时可写 如果 TMOD=1，则不可写

33.5.6 按键通道配置寄存器（TSI_CH_CFGR）

名称	TSI_CH_CFGR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		PROX_CHEN[13:8]					
位权限	U-0		R/W-00 0000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PROX_CHEN[7:0]							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		BUTTON_CHEN[13:8]					
位权限	U-0		R/W-00 0000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BUTTON_CHEN[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:30	--	RFU：未实现，读为 0
29:16	PROX_CHEN	接近检测通道使能，置位的寄存器对应的通道被复用为接近检测功能 比如，PROX_CHEN=0b10_0000_1000_0000，则通道 13 和通道 7 被复用为接近检测
15:14	--	RFU：未实现，读为 0
13:0	BUTTON_CHEN	按键通道使能，置位的寄存器对应的按键通道被使能

33.5.7 Raw Count 寄存器（TSI_RAW_CNTR）

名称	TSI_RAW_CNTR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				SID			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAW_COUNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAW_COUNT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:16	SID	通道 ID 识别寄存器，用于指示当前 Raw count 是来自于哪个传感器通道的，随着 raw count 同步更新 0000：S0 通道 0001：S1 通道 0010：S2 通道 …… 1101：S13 通道 1110：接近检测通道 1111：RFU
15:0	RAW_COUNT	TSI 模块输出 raw count，有效位宽由 Resolution 寄存器决定，最大 16bit，最小 8bit

33.5.8 时钟和周期控制寄存器（TSI_CCR）

名称	TSI_CCR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SCAN_PERIOD							
位权限	R/W-0000_0101							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SCAN_INTV							
位权限	R/W-0000_0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SCAN_IDLE					PRS
位权限	U-0		R/W-00000					R/W-0

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:16	SCAN_IDLE	每个扫描周期启动后，首次扫描前的准备时间配置，计数时钟为 APBCLK，共 32*(SCAN_IDLE +1)个周期
15:8	SCAN_INTV	连续两个 sensor 扫描之间的采样间隔配置，计数时钟为 APBCLK，共 32*(SCAN_INTV+1)个周期
7:6	--	RFU：未实现，读为 0
5:1	SCAN_PERIOD	扫描周期配置寄存器，定义一轮完整扫描（所有通道）的总周期，包含通道扫描时间和停止时间 步长 20ms，缺省 120ms 0_0000：20ms 0_0001：40ms 0_0010：60ms 0_0011：80ms 0_0100：100ms 0_0101：120ms

Bit	助记符	功能描述
		…. 1_1000：500ms Others：1s
0	PRS	伪随机时钟序列使能 0：F_{SW} 采用固定频率 1：F_{SW} 采用伪随机频率，本原多项式 $f(x)=x^8+x^4+x^3+x^2+1$

33.5.9 DMA 控制寄存器（TSI_DMACR）

名称	TSI_DMACR							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	EN	DMA 使能 1：允许产生 DMA 请求 0：关闭 DMA 请求

33.5.10 按键通道扫描配置寄存器（TSI_CHx_CFGR）

名称	TSI_CHx_CFGR(x=0~13)							
offset	0x24~0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	IDAC2_CFG						
位权限	U-0	R/W-0000000						
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	IDAC1_CFG						
位权限	U-0	R/W-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	IDAC_MODE	FSW_CFG	RES
位权限	R/W-0	R/W-000	R/W-0000

Bit	助记符	功能描述
31:23	--	RFU：未实现，读为 0
22:16	IDAC2_CFG	通道 x 的 IDAC2 电流配置，1.2uA/LSB
15	--	RFU：未实现，读为 0
14:8	IDAC1_CFG	通道 x 的 IDAC1 电流配置，1.2uA/LSB
7	IDAC_MODE	通道 x 电流源模式配置 0：single IDAC 1：dual IDAC
6:4	FSW_CFG	通道 x 电容充放电开关频率配置 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 其他：不分频
3:0	RES	通道 x 扫描精度配置（N 值） 0000：8 位 0001：9 位 … 1000：16 位 其他：16 位

33.5.11 接近按键扫描配置寄存器（TSI_PROX_CFGR）

名称	TSI_PROX_CFGR							
offset	0x5C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	IDAC2_CFG						
位权限	U-0	R/W-0000000						
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	IDAC1_CFG						
位权限	U-0	R/W-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDAC_MODE	FSW_CFG			RES			
位权限	R/W-0	R/W-000			R/W-0000			

Bit	助记符	功能描述
31:23	--	RFU：未实现，读为 0
22:16	IDAC2_CFG	接近通道的 IDAC2 电流配置，1.2uA/LSB
15	--	RFU：未实现，读为 0
14:8	IDAC1_CFG	接近通道的 IDAC1 电流配置，1.2uA/LSB

Bit	助记符	功能描述
7	IDAC_MODE	接近通道电流源模式配置 0：single IDAC 1：dual IDAC
6:4	FSW_CFG	接近通道电容充放电开关频率配置 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 其他：不分频
3:0	RES	接近通道扫描精度配置（N 值） 0000：8 位 0001：9 位 … 1000：16 位 其他：16 位

33.5.12 IDAC 调校寄存器（TSI_IDAC_TR）

名称	TSI_IDAC_TR							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			IDAC2_TRIM				
位权限	U-0			R/W-10000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			IDAC1_TRIM				
位权限	U-0			R/W-10000				

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	IDAC2_TRIM	IDAC2 的校准值
7:5	--	RFU：未实现，读为 0
4:0	IDAC1_TRIM	IDAC1 的校准值

33.5.13 IDAC 模拟控制寄存器（TSI_ANA_CR）

名称	TSI_ANA_CR							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		RC_BYP	CMP_RC		CMPHS	VREFBUF_CFG	
位权限	U-0		R/W-0	R/W-01		R/W-0	R/W-10	

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	RC_BYP	比较器输入端 RC 滤波 bypass 使能，为 1 时 bypass 滤波网络
4:3	CMP_RC	比较器输入端 RC 滤波带宽配置 00：1.48M（-3dB 带宽） 01：0.629M 10：0.4M 11：0.293M
2	CMPHS	比较器高速模式使能 0：不使能； 1：使能，偏置电流翻倍
1:0	VREFBUF_CFG	Reference buffer 的充电电流配置 00：不使能 01：弱驱动能力 10：中等驱动能力 11：强驱动能力

33.5.14 TSC_DFT 寄存器（TSI_TEST）

名称	TSI_TEST							
offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					Ibp_trim		
位权限	U-0					R/W-000		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Sense_cfg				-	Ch_idle_cfg	R2d_cfg	
位权限	R/W-0000				U-0	R/W-1	R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	Sensi_valid	Test_ring_en					
位权限	U-0	R/W-0	R/W-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Test_ring_en							
位权限	R/W-00000000							

Bit	助记符	功能描述

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:24	Ibp_trim	ibp 电流 trim 档位
23:20	Sense_cfg	灵敏度测试档位配置，仅灵敏度测试时使用。正常扫描时保持全 0.setp:0.1pF 0000:无效 0001:0.1pF 0010:0.2pF … 1111:1.5pF
19	--	RFU：未实现，读为 0
18	Ch_idle_cfg	所有传感器的不活跃连接配置 0：浮空 1：强制接地
17:16	R2d_cfg	接地大电阻的配置 00:100kΩ 01:150kΩ 10:200kΩ 11:250kΩ
15	--	RFU：未实现，读为 0
14	Sensi_valid	Test_ring 的连接配置 0：test_ring 接大电阻 1：test_ring 接电容阵列
13:0	Test_ring_en	各个传感器的连接 test_ring 使能 0：对应传感器不连接 test_ring 1：对应传感器连接 test_ring

33.5.15 ANA TEST 寄存器（TSI_ANA_TEST）

名称	TSI_ANA_TEST							
offset	0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					Ibias_outen	chg_sel	Idac_comp_mode
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Idac2_ranget	Idac1_ranget	Vref_buf_ent	Comp_ent	Idac2_ent	Idac1_ent	Sigma_delta_ent	Debug_mode
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0

Bit	助记符	功能描述
10	Ibias_outen	IDAC1 的电流测试输出 0：不输出 1：输出
9	chg_sel	IDAC1 的反馈控制，仅 debug 模式下起作用 0：使用采样寄存器之后的信号作反馈 1：直接使用比较器输出作反馈
8	Idac_comp_mode	IDAC2 的补偿模式控制，非 debug 模式下起作用。在 debug 模式下，idac1_chg_sw 信号为比较器输出信号和 Idac1_ent 信号的与，idac2_chg_sw 信号为比较器输出信号和 Idac2_ent 信号的与 0：常开模式（非 debug 模式下，idac1_chg_sw 信号为比较器输出信号，idac2_chg_sw 信号一直为 1） 1：互补模式（非 debug 模式下，idac1_chg_sw 信号为比较器输出信号，idac2_chg_sw 信号为比较器输出信号的反）
7	Idac2_ranget	IDAC2 电流翻倍模式 0：不翻倍； 1：翻倍
6	Idac1_ranget	IDAC1 电流翻倍模式 0：不翻倍； 1：翻倍
5	Vref_buf_ent	0：debug 模式下 vref_buf_en、vref_buf_cona、vref_buf_conb 受内部状态机控制 1：debug 模式下 vref_buf_en 为 1，vref_buf_cona 为 1，vref_buf_conb 为 0，cmod_cona 信号为 1，cmod_conb 信号为 0.
4	Comp_ent	0：debug 模式下 comp_en 受模块内部状态机控制 1：debug 模式下 comp_en 为 1
3	Idac2_ent	0：debug 模式下 idac2_en 为 0 1：debug 模式下 idac2_en 为 1
2	Idac1_ent	0：debug 模式下 idac1_en 为 0 1：debug 模式下 idac1_en 为 1
1	Sigma_delta_ent	0：debug 模式下 delta_bias_en 受模块内部状态机控制 1：debug 模式下 delta_bias_en 为 1
0	Debug_mode	Debug 模式控制器，会影响 idac1_en、idac2_en、idac1_chg_sw、idac2_chg_sw、vref_buf_en、vref_buf_cona、vref_buf_conb、cmod_cona、cmod_conb、comp_en、delta_bias_en、s0_swen 信号。 0：所有信号受模块内部状态机控制 1：所有信号受特殊控制，详见上述各个 bit 的描述。

34 I/O 端口（GPIO）

34.1 概述

I/O 端口的主要功能特性：

- GPIO 数字输入具有施密特特性
- 部分 GPIO 输入支持模拟滤波
- 部分 GPIO 输入支持数字滤波
- GPIO 可配置为上拉、下拉、开漏输出
- Sleep/DeepSleep 模式下保持状态
- 上下电复位期间，所有 IO 控制信号被钳位到 disable 状态
- 可配置输出驱动能力（x1, x2, x3, x4）

34.2 引脚类型

FM33FR0 主要有三种类型的 GPIO 引脚，其中大部分引脚支持输入输出、数字外设功能、模拟外设通道、可控上拉电阻、可控开漏输出功能。

34.2.1 GPIO，输入输出使能，可控上下拉电阻，可控开漏输出，可控驱动能力

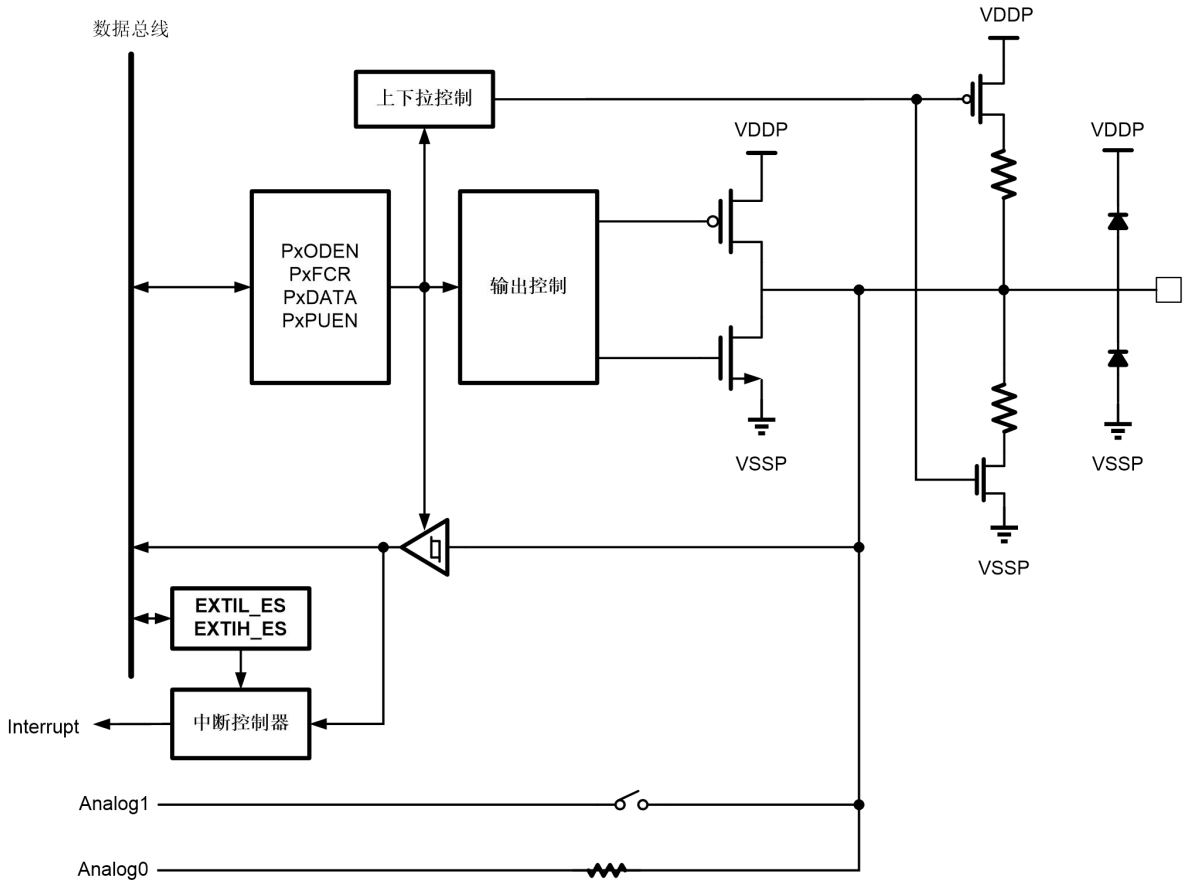


图 34-1 普通 GPIO 结构框图

控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	0	0	0/1	x	0	1	0/1
	1	1		0	0	1	
				1	1	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表 34-1GPIO 功能逻辑定义表

上下拉电阻各50K欧姆

4级驱动能力：

X1 – 0mA

X2 – 5mA

X3 – 10mA

X4 – 15mA

34.2.2 GPIO，输入输出使能，真开漏输出（PB0, PB1）

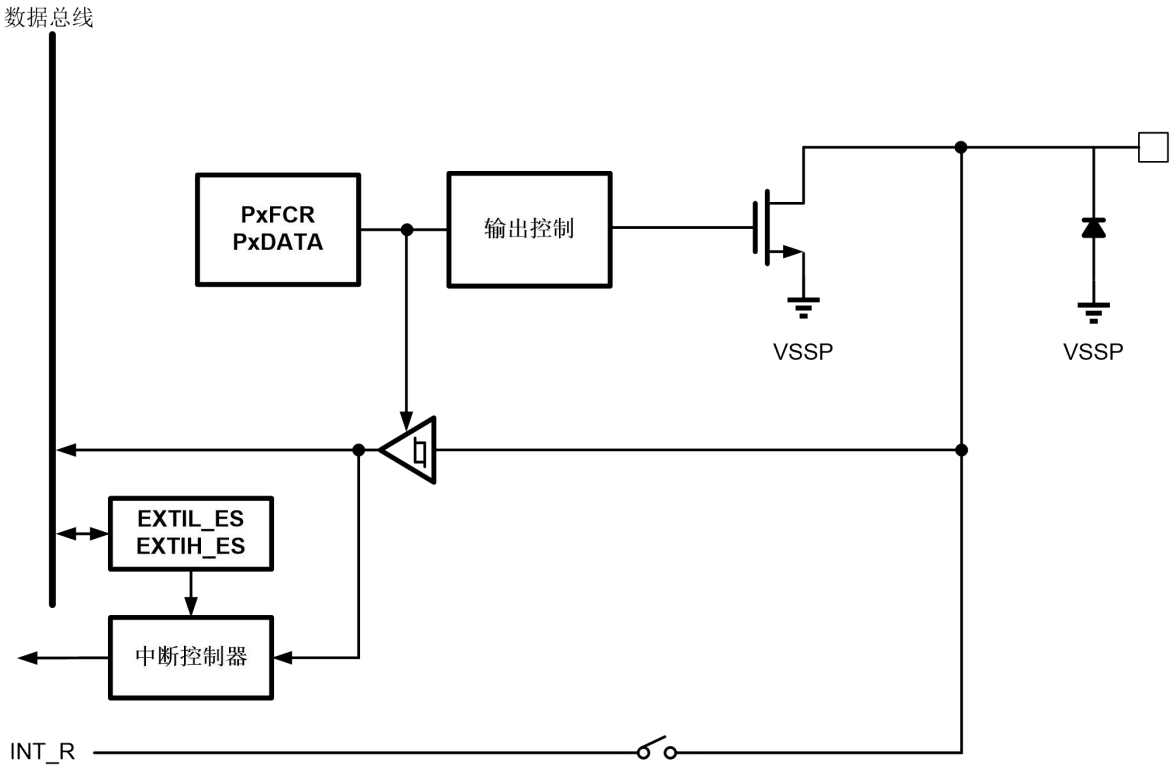


图 34-2 真开漏 GPIO 结构框图

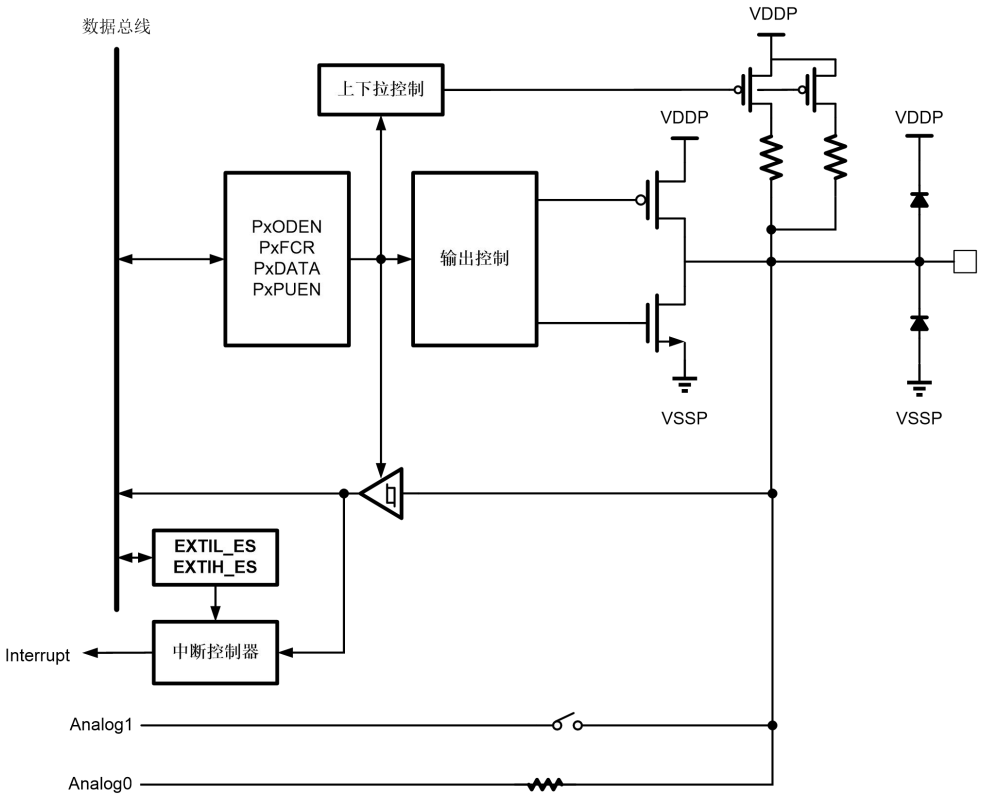
上述IO的控制逻辑定义如下：

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	0	x	0/1	0	0	1	0/1
	1			1	1	0	

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
10	x	x	0/1	外设输入功能	1	0	0/1
	x	x		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表34-2真开漏IO功能逻辑定义表

34.2.3 PD10（NRST），仅上拉，无开漏输出，无可控驱动



PD10作为GPIO时有50K可控上拉电阻，没有下拉电阻

作为NRST时，固定10K上拉电阻，不可控

作为GPIO时输出驱动能力固定为10

34.3 IO 端口功能定义

芯片大部分引脚为数模混合IO，每个通用GPIO都有4bit控制寄存器：FCR[1:0]、PUEN、ODEN，其中FCR用于选择IO引脚功能，定义如下：

FCR: Function Control Register	PAD function
00	GPIO input
01	GPIO output
10	Digital Function（数字外设功能）
11	Analog

表 34-3FCR 定义表

34.3.1 GPIO 输入

当某个 GPIO 被配置成输入功能，并且对应的输入使能寄存器被置位时：

- 输出驱动缓冲器被关闭
- 施密特触发器使能
- 上下拉电阻由 PxPUP 寄存器控制上拉或下拉
- 上下拉电阻由 PxPUDEN 寄存器控制使能或关闭
- PxDIN 寄存器直接反应 IO 上的电平状态

34.3.2 GPIO 输出

当某个 GPIO 被配置成输出功能，并且对应的输出使能寄存器被置位时：

- 输出驱动缓冲器使能
  - 开漏输出模式（PxODEN=1）：输出 0 时 IO 驱动低电平，输出 1 时 IO 关闭驱动缓冲器
  - 推挽输出模式（PxODEN=0）：输出 0 时 IO 驱动低电平，输出 1 时 IO 驱动高电平
- 上下拉电阻由 PxPUP 寄存器控制上拉或下拉
- 上下拉电阻由 PxPUDEN 寄存器控制使能或关闭
- INEN 使能，软件读取 PxDIN 寄存器能够获得 IO 上的电平状态
- 软件读取 PxDO 寄存器获得上次写入的值
- 支持 4 级驱动能力可配置，解决驱动能力和 EMI 之间的平衡

34.3.3 数字外设功能

当某个 GPIO 被配置成数字外设功能：

- IO 的输入或输出方向由所连接的外设功能决定
- 由 PxODEN 控制输出时是开漏输出还是推挽输出

- 上下拉电阻由 PxPUP 寄存器控制上拉或下拉
- 上下拉电阻由 PxPUDEN 寄存器控制使能或关闭
- 每个 IO 支持 2bit 外设功能选择，可以自由选择 4 种外设功能接入

额外的外设功能选择通过 GPIOx_DFS 寄存器实现。具体引脚功能列表参见 2 引脚和封装。

当存在多个 GPIO 连接到某个数字外设的输入时，为了避免数据冲突，按照 A->E、0->15 的优先级判决进行数据选择。比如，PA0 和 PB0 都连接某个外设输入，则 PA0 数据被送到外设端，PB0 输入信号无效。

34.3.4 模拟功能

当某个 GPIO 被配置成模拟功能：

- 输出缓冲器关闭
- 数字输入功能关闭
- 上下拉电阻关闭
- 软件读取 PxDIN 返回 0
- IO 模拟通道被连接到特定的模拟外设上
- 如果一个 IO 同时连接到多个模拟外设，则多个模拟外设在同一时刻只能使能其中一个

GPIO 有两个模拟通道，分别为电阻通道和开关通道。电阻通道无法关闭，引脚上的任何信号都会传输到芯片内部，开关通道可以通过 ANEN 信号使能或关闭。

芯片绝大部分模拟信号，比如 COM/SEG，都是连接到 GPIO 的电阻通道上，其输出电平由模块内部关闭或使能。其他模拟信号部分走电阻通道，部分走开关通道，参见下表：

模拟功能	PAD 通道	说明
COM	电阻	模块内部有开关
SEG	电阻	
XTHF	电阻	模块内部有开关
XTLF	电阻	
ADC_INx	电阻	模块内部有开关
TSI	电阻	模块内部有开关

表 34-4PAD 通道配置

34.3.5 使用外部晶体引脚

FM33FR0 支持外接 32768Hz 晶体和 4~24MHz 高频晶体。

其中 PA8 和 PA9 默认为 GPIO，配置为模拟功能之后作为 XTHFIN 和 XTHFOUT 外接高频晶体。

其中 PA5 和 PA6 默认为 GPIO，配置为模拟功能之后作为 XT32KI 和 XT32KO 外接 32768Hz 晶体。

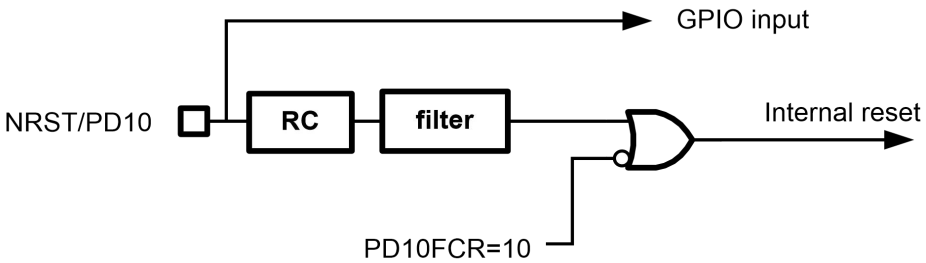
如果要使用外部 32K 时钟输入，可从 XT32KI 灌入 32K 时钟。

34.4 NRST 引脚与 GPIO 复用

NRST 为全局复位引脚，带有数字滤波，滤波采用 RCLP 时钟工作，RCLP 保持常开，无法关闭。

为了提高管脚利用率，用户模式下的 NRST 引脚也可以复用为 GPIO(PD10)。

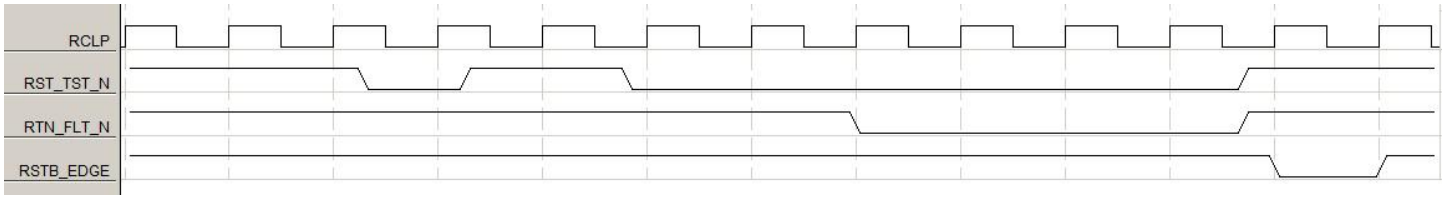
PD10FCR 寄存器复位值为 10，即复位默认该引脚为 NRST 功能，并且 PDPUEN10 复位值为 1，即默认上拉使能。软件可以通过将 PD10FCR 修改为 00 或 01 来使能其 GPIO 功能，此时 NRST 复位功能无效。PD10 作为 GPIO 时只有 50K 上拉电阻，无下拉电阻，作为 NRST 时固定为 10K 上拉电阻。



NRST 数字滤波

数字滤波采用 RCLP 连续采样 64 拍，以及输入信号高电平异步复位的方式，即确保 64 个 RCLP 周期内 NRST 信号保持稳定的低电平。由于输入信号与 RCLP 为异步关系，实际有效滤波长度可能是 63~64 个 RCLP 周期之间，即 1.95ms 的典型范围，随着温度变化可能会有 10%左右的变化。

下图为数字滤波示意图，RST_FLT_N 为数字滤波输出，RSTB_EDGE 为上升沿产生的复位信号。



34.5 SWD 引脚

ARM SWD 引脚复用 PA3(SWCLK)和 PA4(SWIO)，这两个 GPIO 上电复位后默认为 SWD 功能，并且默认使能内部上拉电阻(50K)，以省去外部上拉。

34.6 WKUPx 引脚

FM33FR0 有 8 个 WKUP 引脚，能够将芯片从 Sleep/DeepSleep 模式下唤醒，即使片上振荡器都停止工作，WKUP 仍能唤醒芯片。

WKUPx 引脚输入上升沿、下降沿或上升下降沿（软件配置）能够将芯片从休眠模式下唤醒。为了使能此功能，需将对应引脚配置为 GPIO 输入功能，并且相应的 PINWKEN 置位，注意 PAD 内部带有上拉电阻，如果配置为上升沿唤醒，则必须关闭上拉电阻。

每个支持 WKUP 功能的 IO 都带有大约 100ns 的片内模拟滤波，能够滤除输入信号上的毛刺，避免误触发。

Sleep/DeepSleep 模式下，使能了的 WKUPx 引脚上任何大于 100ns 的脉冲都会触发芯片唤醒。

8 路 WKUPx 电路结构完全独立，下图显示了其中一路 WKUP 功能的结构框图

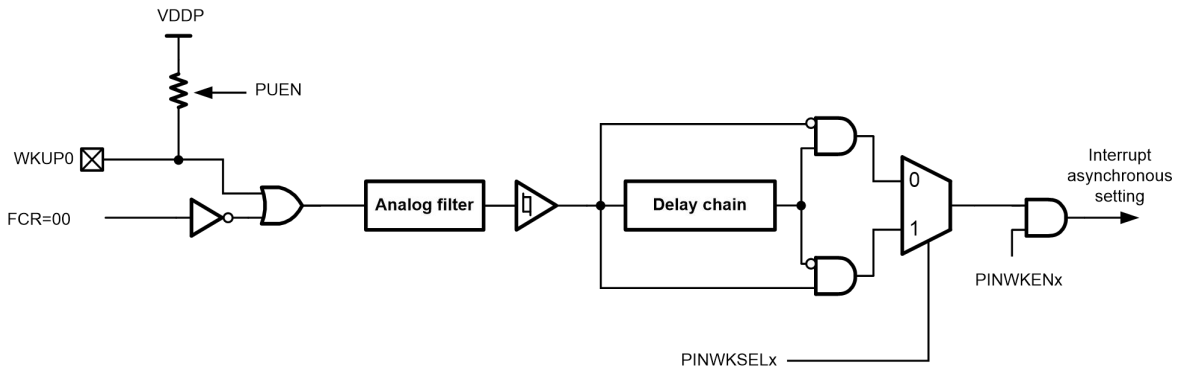


图 34- 3WKUPx 功能结构框图

WKUPx 功能使用时需要注意外部引脚输入的初始状态。在使能 WKUP 时，可能由于初态的关系导致虚假的唤醒事件，软件应注意识别并处理。

使用 WKUP 功能时，必须将对应引脚的 FCR 寄存器配置为 00（GPIO 输入），按需要设置唤醒边沿（PINWKSELx）并使能 PINWKENx 寄存器。当某个 WKUPx 引脚上产生唤醒事件后，PMU 模块内部的唤醒源标志查询寄存器内对应的 bit 位将会自动置位。

34.7 外部引脚中断（EXTI）

34.7.1 功能说明

FM33FR0 的 6 组 GPIO（A~F）最多可以产生 22 个 EXTI 中断，每组 GPIO 分别可以产生 4 个 EXTI 中断标志，最终所有的 EXTI 中断汇总到 NVIC 的#46 入口。

中断标志和引脚对应关系如下表：

GPIO	EXTI输入选择	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]
PA8~PA11	EXTI_ASEL[5:4]	EXTI[2]
PA12~PA15	EXTI_ASEL[7:6]	EXTI[3]
PB0~PB3	EXTI_BSEL[1:0]	EXTI[4]
PB4~PB7	EXTI_BSEL[3:2]	EXTI[5]
PB8~PB11	EXTI_BSEL[5:4]	EXTI[6]
PB12~PB15	EXTI_BSEL[7:6]	EXTI[7]
PC0~PC3	EXTI_CSEL[1:0]	EXTI[8]
PC4~PC7	EXTI_CSEL[3:2]	EXTI[9]
PC8~PC11	EXTI_CSEL[5:4]	EXTI[10]
PC12~PC13	EXTI_CSEL[7:6]	EXTI[11]
PD0~PD3	EXTI_DSEL[1:0]	EXTI[12]
PD4~PD7	EXTI_DSEL[3:2]	EXTI[13]
PD8~PD11	EXTI_DSEL[5:4]	EXTI[14]
PD12~PD15	EXTI_DSEL[7:6]	EXTI[15]
PE0~PE3	EXTI_ESEL[1:0]	EXTI[16]
PE4~PE7	EXTI_ESEL[3:2]	EXTI[17]
PE8~PE11	EXTI_ESEL[5:4]	EXTI[18]
PE12~PE15	EXTI_ESEL[7:6]	EXTI[19]
PF0~PF3	EXTI_FSEL[1:0]	EXTI[20]
PF4~PF7	EXTI_FSEL[3:2]	EXTI[21]

EXTISEL 寄存器用于选择某个 IO 接入 EXTI 通道，EXTI 模块可以配置是否对输入信号进行数字滤波。

数字滤波的实现方法是由 IO 采样时钟连续采样到 3 次相同电平才认为是合法电平输入，如下图所示。

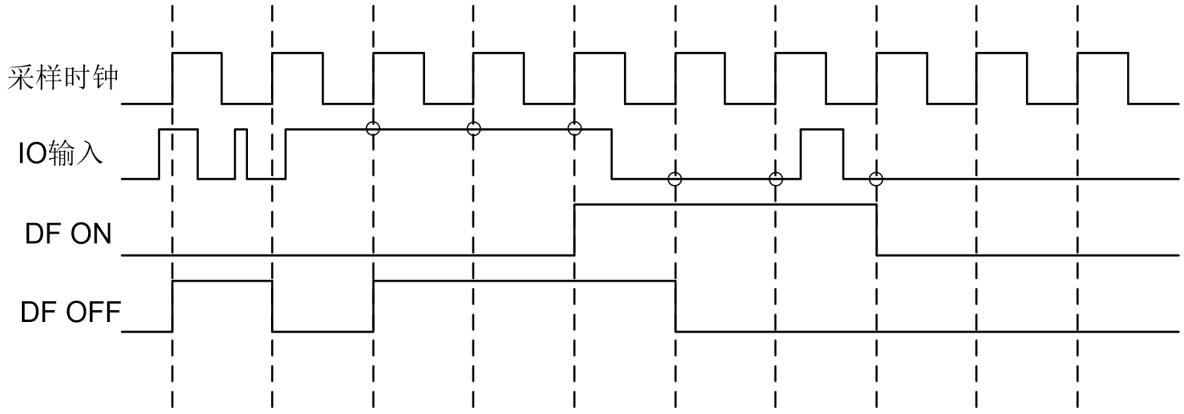


图 34-4 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

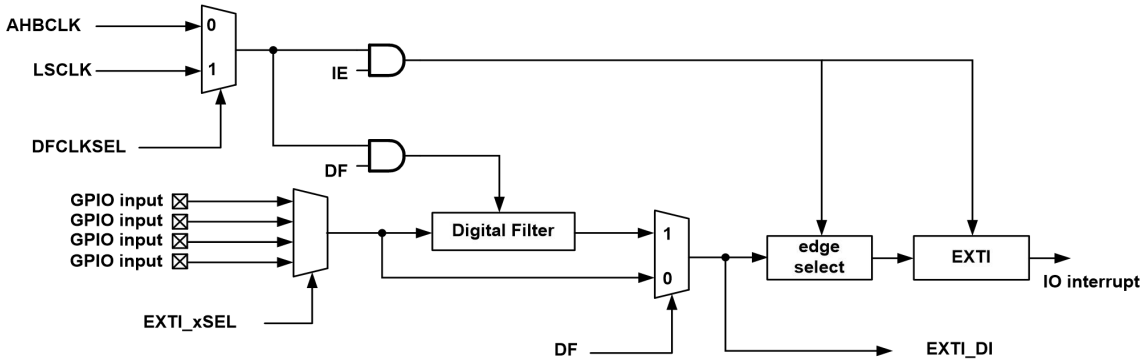


图 34-5 EXTI 信号输入示意图

用户应根据引脚功能需要使能或禁止数字滤波功能，使能数字滤波后，将根据 AHBCLK 频率不同，对 IO 输入信号引入不同的采样延迟。经过数字滤波后的输出信号，软件也可以在 EXTI_DI 寄存器读到。

EXTI还可以配置输入信号的有效边沿，支持上升沿、下降沿、上升下降沿触发中断，或者禁止EXTI中断触发，由EXTIEDS寄存器配置。

34.7.2 应用指南

如需在 Sleep/DeepSleep 模式下启动 EXTI 中断唤醒功能，推荐按照如下步骤进行操作：

- 关闭所有 EXTI 使能
- 配置 SYSCLKCR 寄存器（0x40002400）的 SLP_ENEXTI 位为 1，选择 LSCLK 进行 EXTI 采样
- 根据需要打开或关闭 EXTI 数字滤波使能
- 配置相应 GPIO 为输入
- 配置 EXTISEL 寄存器选择对应的 IO
- 置位 EXTICKE，打开 EXTI 工作时钟使能

- 等待至少 4 个 LSCLK 周期
- 配置 EXTIEDS 触发边沿选择，使能所需的 EXTI 中断
- 正常进入 Sleep 模式

芯片上电后默认关闭所有 EXTI，同时默认的引脚中断采样时钟是系统时钟 APBCLK。如果用户使用系统时钟产生 EXTI，推荐流程如下：

- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 APBCLK 周期
- 配置 EXTIEDS 触发边沿选择，使能所需的 EXTI 中断

如果希望使用低速的 LSCLK 来产生 EXTI，推荐流程如下：

- 将 EXTI 采样时钟配置为 LSCLK
- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 置位 EXTICKE，打开 EXTI 采样时钟使能
- 等待至少 4 个 LSCLK 时钟周期
- 配置 EXTIEDS 触发边沿，使能所需的 EXTI 中断

34.8 快速 GPIO 输出

FM33FR0 可以通过 set-reset 功能快速改变每个 GPIO 的输出数据，从而提高 IO 输出效率，特别是可以提高 read-modify-write 操作的效率和可靠性（atomic）。方法是每个 GPIO 组的输出数据寄存器都有 2 组 set-reset 映射虚拟地址，对 DSET 寄存器特定 bit 写 1 可以置位对应的数据寄存器的 bit 位，对 DRST 寄存器特定地址写 1 可以清除对应的数据寄存器的 bit 位。

34.9 寄存器

GPIO模块基地址：0x4000_0C00

offset 地址	名称	符号
0x00	GPIOA 输入使能寄存器（GPIOA Input Enable Register）	GPIOA_INEN
0x04	GPIOA 上下拉使能寄存器（GPIOA Pull-Up Pull-Down Enable Register）	GPIOA_PUDEN
0x08	GPIOA 开漏使能寄存器（GPIOA Open-Drain Enable Register）	GPIOA_ODEN
0x0C	GPIOA 功能选择寄存器（GPIOA Function Control Register）	GPIOA_FCR
0x10	GPIOA 输出数据寄存器（GPIOA Data Output Register）	GPIOA_DO
0x14	GPIOA 输出数据置位寄存器（GPIOA Data Set Register）	GPIOA_DSET
0x18	GPIOA 输出数据复位寄存器（GPIOA Data Reset Register）	GPIOA_DRST
0x1C	GPIOA 输入数据寄存器（GPIOA Data Input Register）	GPIOA_DIN
0x20	GPIOA 额外数字功能寄存器（GPIOA Digital Function Select）	GPIOA_DFS
0x24		
0x28	GPIOA 模拟开关使能寄存器（GPIOA Analog channel Enable Register）	GPIOA_ANEN
0x2C	GPIOA 驱动能力配置寄存器（GPIOA Driver Strength Register）	GPIOA_DSR
0x40	GPIOB 输入使能寄存器（GPIOB Input Enable Register）	GPIOB_INEN
0x44	GPIOB 上拉使能寄存器（GPIOB Pull-Up Enable Register）	GPIOB_PUEN
0x48	GPIOB 开漏使能寄存器（GPIOB Opeb-Drain Enable Register）	GPIOB_ODEN
0x4C	GPIOB 功能选择寄存器（GPIOB Function Control Register）	GPIOB_FCR
0x50	GPIOB 输出数据寄存器（GPIOB Data Output Register）	GPIOB_DO
0x54	GPIOB 输出数据置位寄存器（GPIOB Data Set Register）	GPIOB_DSET
0x58	GPIOB 输出数据复位寄存器（GPIOB Data Reset Register）	GPIOB_DRST
0x5C	GPIOB 输入数据寄存器（GPIOBData Input Register）	GPIOB_DIN
0x60	GPIOB 额外数字功能寄存器（GPIOB Digital Function Select）	GPIOB_DFS
0x64	-	-
0x68	GPIOB 模拟开关使能寄存器	GPIOB_ANEN

offset 地址	名称	符号
	（GPIOB Analog channel Enable Register）	
0x6C	GPIOB 驱动能力配置寄存器 （GPIOB Driver Strength Register）	GPIOB_DSR
0x80	GPIOC 输入使能寄存器 （GPIOC Input Enable Register）	GPIOC_INEN
0x84	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUEN
0x88	GPIOC 开漏使能寄存器 （GPIOC Opeb-Drain Enable Register）	GPIOC_ODEN
0x8C	GPIOC 功能选择寄存器 （GPIOC Function Control Register）	GPIOC_FCR
0x90	GPIOC 输出数据寄存器 （GPIOC Data Output Register）	GPIOC_DO
0x94	GPIOC 输出数据置位寄存器 （GPIOC Data Set Register）	GPIOC_DSET
0x98	GPIOC 输出数据复位寄存器 （GPIOC Data Reset Register）	GPIOC_DRST
0x9C	GPIOC 输入数据寄存器 （GPIOC Data Input Register）	GPIOC_DIN
0xA0	GPIOC 额外数字功能寄存器 （GPIOC Digital Function Select）	GPIOC_DFS
0xA4	-	-
0xA8	GPIOC 模拟开关使能寄存器 （GPIOC Analog channel Enable Register）	GPIOC_ANEN
0xAC	GPIOC 驱动能力配置寄存器 （GPIOC Driver Strength Register）	GPIOC_DSR
0xB0	GPIOC 定时器通道选择寄存器 （GPIOC Timer Mux Register）	GPIOC_TMUXR
0xC0	GPIOD 输入使能寄存器 （GPIOD Input Enable Register）	GPIOD_INEN
0xC4	GPIOD 上拉使能寄存器 （GPIOD Pull-Up Enable Register）	GPIOD_PUEN
0xC8	GPIOD 开漏使能寄存器 （GPIOD Opeb-Drain Enable Register）	GPIOD_ODEN
0xCC	GPIOD 功能选择寄存器 （GPIOD Function Control Register）	GPIOD_FCR
0xD0	GPIOD 输出数据寄存器 （GPIOD Data Output Register）	GPIOD_DO
0xD4	GPIOD 输出数据置位寄存器 （GPIOD Data Set Register）	GPIOD_DSET
0xD8	GPIOD 输出数据复位寄存器 （GPIOD Data Reset Register）	GPIOD_DRST
0xDC	GPIOD 输入数据寄存器 （GPIOD Data Input Register）	GPIOD_DIN
0xE0	GPIOD 额外数字功能寄存器 （GPIOD Digital Function Select）	GPIOD_DFS
0xE4	-	-

offset 地址	名称	符号
0xE8	GPIOD 模拟开关使能寄存器（GPIOD Analog channel Enable Register）	GPIOD_ANEN
0xEC	GPIOD 驱动能力配置寄存器（GPIOD Driver Strength Register）	GPIOD_DSR
0x100	GPIOE 输入使能寄存器（GPIOE Input Enable Register）	GPIOE_INEN
0x104	GPIOE 上下拉使能寄存器（ GPIOE Pull-Up Pull-Down Enable Register）	GPIOE_PUDEN
0x108	GPIOE 开漏使能寄存器（GPIOE Open-Drain Enable Register）	GPIOE_ODEN
0x10C	GPIOE 功能选择寄存器（GPIOE Function Control Register）	GPIOE_FCR
0x110	GPIOE 输出数据寄存器（GPIOE Data Output Register）	GPIOE_DO
0x114	GPIOE 输出数据置位寄存器（GPIOE Data Set Register）	GPIOE_DSET
0x118	GPIOE 输出数据复位寄存器（GPIOE Data Reset Register）	GPIOE_DRST
0x11C	GPIOE 输入数据寄存器（GPIOE Data Input Register）	GPIOE_DIN
0x120	GPIOE 额外数字功能寄存器（GPIOE Digital Function Select）	GPIOE_DFS
0x124		
0x128	GPIOE 模拟开关使能寄存器（GPIOE Analog channel Enable Register）	GPIOE_ANEN
0x12C	GPIOE 驱动能力配置寄存器（GPIOE Driver Strength Register）	GPIOE_DSR
0x140	GPIOF 输入使能寄存器（GPIOFInput Enable Register）	GPIOF_INEN
0x144	GPIOF 上下拉使能寄存器（ GPIOF Pull-Up Pull-Down Enable Register）	GPIOF_PUDEN
0x148	GPIOF 开漏使能寄存器（GPIOF Open-Drain Enable Register）	GPIOF_ODEN
0x14C	GPIOF 功能选择寄存器（GPIOF Function Control Register）	GPIOF_FCR
0x150	GPIOF 输出数据寄存器（GPIOF Data Output Register）	GPIOF_DO
0x154	GPIOF 输出数据置位寄存器（GPIOF Data Set Register）	GPIOF_DSET
0x158	GPIOF 输出数据复位寄存器（GPIOF Data Reset Register）	GPIOF_DRST
0x15C	GPIOF 输入数据寄存器（GPIOF Data Input Register）	GPIOF_DIN
0x160	GPIOF 额外数字功能寄存器（GPIOF Digital Function Select）	GPIOF_DFS

offset 地址	名称	符号
0x164		
0x168	GPIOF 模拟开关使能寄存器（GPIOF Analog channel Enable Register）	GPIOF_ANEN
0x16C	GPIOF 驱动能力配置寄存器（GPIOF Driver Strength Register）	GPIOF_DSR
0x1C0	EXTI 输入选择寄存器 0（External Interrupt Input Select Register0）	GPIO_EXTISEL0
0x1C4	EXTI 输入选择寄存器 1（External Interrupt Input Select Register1）	GPIO_EXTISEL1
0x1C8	EXTI 边沿选择和使能寄存器 0（External Interrupt Edge Select and Enable Register0）	GPIO_EXTIEDS0
0x1CC	EXTI 边沿选择和使能寄存器 1（External Interrupt Edge Select and Enable Register1）	GPIO_EXTIEDS1
0x1D0	EXTI 数字滤波控制寄存器（External Interrupt Digital Filter Register）	GPIO_EXTIDF
0x1D4	EXTI 中断标志寄存器（External Interrupt and Status Register）	GPIO_EXTIISR
0x1D8	EXTI 输入信号寄存器（External Interrupt Data Input Register）	GPIO_EXTIDI
0x200	FOUT 配置寄存器（Frequency Output Select Register）	GPIO_FOUTSEL
0x300	WKUP 使能寄存器（Wakeup Enable Register）	GPIO_PINWKEN

34.9.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x00 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxINEN15	PxINEN14	PxINEN13	PxINEN12	PxINEN11	PxINEN10	PxINEN9	PxINEN8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxINEN7	PxINEN 6	PxINEN 5	PxINEN 4	PxINEN 3	PxINEN 2	PxINEN 1	PxINEN 0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxINEN	GPIO 输入使能控制 0：关闭输入使能 1：打开输入使能

34.9.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）

名称	GPIOx_PUDEN (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x04 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxPUD [15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxPUD [7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxPUDEN [15:8]							
位权限	R/W-0000 0100							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxPUDEN [7:0]							
位权限	R/W-00011000							

Bit	助记符	功能描述
31:16	PxPUD	GPIO 上下拉选择控制 0：选择上拉 1：选择下拉 *注：PDPUEN[10]复位值为0，默认上拉（NRST）* *注：PAPUEN[3]和 PAPUEN[4]复位值为 0，默认弱上拉（SWD）*
15:0	PxPUDEN	GPIO 上下拉使能开关 0：关闭上下拉使能 1：打开上下拉使能 弱上拉和弱下拉为 100K *注：PDPUEN[10]复位值为1，默认上拉（NRST）* *注：PAPUEN[3]和 PAPUEN[4]复位值为 1，默认弱上拉（SWD）*

34.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x08 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PAODEN15	PAODEN14	PAODEN13	PAODEN12	PAODEN11	PAODEN10	PAODEN9	PAODEN8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAODEN7	PAODEN6	PAODEN5	PAODEN4	PAODEN3	PAODEN2	PAODEN1	PAODEN0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxODEN	GPIO 开漏输出使能 0：关闭开漏输出 1：使能开漏输出

34.9.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x0C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15FCR		Px14FCR		Px13FCR		Px12FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Px11FCR		Px10FCR		Px9FCR		Px8FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Px7FCR		Px6FCR		Px5FCR		Px4FCR	
位权限	R/W-0		R/W-0		R/W-0		R/W-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	Px3FCR	Px2FCR	Px1FCR	Px0FCR
位权限	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:30	Px15FCR	Px[15]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
29:28	Px14FCR	Px[14]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
27:26	Px13FCR	Px[13]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
25:24	Px12FCR	Px[12]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
23:22	Px11FCR	Px[11]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
21:20	Px10FCR	Px[10]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function *注：PD10FCR 复位值为 10（NRST 功能）*
19:18	Px9FCR	Px[9]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
17:16	Px8FCR	Px[8]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
15:14	Px7FCR	Px[7]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function

Bit	助记符	功能描述
13:12	Px6FCR	Px[6]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
11:10	Px5FCR	Px[5]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
9:8	Px4FCR	Px[4]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function *注：PA14FCR 复位值为 10（SWD 功能）*
7:6	Px3FCR	Px[3]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function *注：PA3FCR 复位值为 10（SWD 功能）*
5:4	Px2FCR	Px[2]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
3:2	Px1FCR	Px[1]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
1:0	Px0FCR	Px[0]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function

34.9.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x10 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							

位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDO15	PxDO14	PxDO13	PxDO12	PxDO11	PxDO10	PxDO9	PxDO8
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDO7	PxDO6	PxDO5	PxDO4	PxDO3	PxDO2	PxDO1	PxDO0
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxDO	GPIO output data register

34.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x14 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxSET15	PxSET14	PxSET13	PxSET12	PxSET11	PxSET10	PxSET9	PxSET8
位权限	W	W	W	W	W	W	W	W
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxSET7	PxSET6	PxSET5	PxSET4	PxSET3	PxSET2	PxSET1	PxSET0
位权限	W	W	W	W	W	W	W	W

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxSET	GPIO output data set register 举例：向 PADSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。

34.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST (x=A~F)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x18 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxRESET15	PxRESET14	PxRESET13	PxRESET12	PxRESET11	PxRESET10	PxRESET9	PxRESET8
位权限	W	W	W	W	W	W	W	W
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxRESET7	PxRESET6	PxRESET5	PxRESET4	PxRESET3	PxRESET2	PxRESET1	PxRESET0
位权限	W	W	W	W	W	W	W	W

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxRESET	GPIO output data reset register 举例：向 PADRST 写 0x0000_8000，则 PADO[15]清零，其余位保持不变

34.9.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x1C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDIN15	PxDIN14	PxDIN13	PxDIN12	PxDIN11	PxDIN10	PxDIN9	PxDIN8
位权限	R	R	R	R	R	R	R	R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDIN7	PxDIN6	PxDIN5	PxDIN4	PxDIN3	PxDIN2	PxDIN1	PxDIN0
位权限	R	R	R	R	R	R	R	R

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxDIN	Portx input data register 此寄存器仅占用地址空间，无物理实现。软件读此寄存器直接返回引脚输入信号，芯片并不对引脚输入进行锁存

34.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS (x=A~E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x20 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxDFS[31:24]							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxDFS[23:16]							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDFS[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDFS[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	PxDFS	Portx Digital Function Select 对于具有多个数字外设功能的引脚，通过 PxDFS 寄存器可以选择使用哪个外设功能。 注意，对于不同的 IO 分组，有效的寄存器位置是不一样的，详细定义请参考 2.1.72.1.82.1.92.1.10 2.1.10 2.1.11

34.9.10 GPIO 模拟开关使能寄存器（GPIOx_ANEN）

名称	GPIOx_ANEN (x=A~E)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x28 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxANEN[15:8]							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxANEN[7:0]							
位权限	R/W-0							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxANEN	PortX 模拟开关使能 1：使能 IO 模拟开关 0：关闭 IO 模拟开关

34.9.11 GPIO 驱动强度配置寄存器（GPIOx_DSR）

名称	GPIOx_DSR (x=A~F)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x2C + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxDS[31:24]							
位权限	R/W-1010 1010							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxDS[15:8]							
位权限	R/W-1010 1010							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDS[15:8]							
位权限	R/W-1010 1010							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDS[7:0]							
位权限	R/W-1010 1010							

Bit	助记符	功能描述
31:0	PxDS	PortX 驱动强度配置，2bit 为一组控制 1 个 GPIO 比如 PxDS[31:30]用于控制 Px15 引脚，PxDS[7:6]用于控制 Px3 引脚 00：驱动关闭 X1 01：驱动最弱 X2 10：驱动次强 X3 11：驱动最强 X4 上电后默认所有引脚驱动为（X3）

34.9.12 GPIOC 定时器通道选择寄存器（GPIOC_TMUXR）

名称	GPIOC_TMUXR							
offset	0xB0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		-		-		-	
位权限	U-0		U-0		U-0		U-0	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		-		-		PC8TMUX	
位权限	U-0		U-0		U-0		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PC7TMUX		PC6TMUX		PC5TMUX		PC4TMUX	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PC3TMUX		PC2TMUX		PC1TMUX		-	
位权限	R/W-00		R/W-00		R/W-00		U-0	

Bit	助记符	功能描述
31:18	--	RFU：未实现，读为 0
17:16	PC8TMUX	PC8 引脚定时器通道选择 00,11：RFU 01：TI07 10：TO07
15:14	PC7TMUX	PC7 引脚定时器通道选择 00,11：ATIM_CH3N 01：TI06 10：TO06
13:12	PC6TMUX	PC6 引脚定时器通道选择 00,11：ATIM_CH2N 01：TI05 10：TO05
11:10	PC5TMUX	PC5 引脚定时器通道选择 00,11：ATIM_CH1N 01：TI04 10：TO04
9:8	PC4TMUX	PC4 引脚定时器通道选择 00,11：ATIM_CH4 01：TI03 10：TO03
7:6	PC3TMUX	PC3 引脚定时器通道选择 00,11：ATIM_CH3 01：TI02 10：TO02
5:4	PC2TMUX	PC2 引脚定时器通道选择 00,11：ATIM_CH2 01：TI01 10：TO01
3:2	PC1TMUX	PC1 引脚定时器通道选择 00,11：ATIM_CH1 01：TI00

Bit	助记符	功能描述
		10：TO00
1:0	--	RFU：未实现，读为 0

34.9.13 EXTI 输入选择寄存器 0（GPIO_EXTISEL0）

名称	GPIO_EXTISEL0							
offset	0x1C0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DSEL							
位权限	R/W-00 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_CSEL							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_BSEL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_ASEL							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:24	EXTI_DSEL[5:0]	PortD EXTI 中断输入选择 EXTI_DSEL[7:6] – 00：PD12 01：RFU 10：RFU 11：RFU EXTI_DSEL[5:4] – 00：PD8 01：PD9 10：PD10 11：PD11 EXTI_DSEL[3:2] – 00：PD4 01：PD5 10：PD6 11：PD7 EXTI_DSEL[1:0] – 00：PD0 01：PD1 10：PD2 11：PD3
23:16	EXTI_CSEL[5:0]	PortC EXTI 中断输入选择 EXTI_CSEL[5:4] – 00：PC8

Bit	助记符	功能描述
		01：PC9 10：PC10 11：PC11 EXTI_CSEL[3:2] – 00：PC4 01：PC5 10：PC6 11：PC7 EXTI_CSEL[1:0] – 00：PC0 01：PC1 10：PC2 11：PC3
15:8	EXTI_BSEL[7:0]	PortB EXTI 中断输入选择 EXTI_BSEL[7:6] – 00：PB12 01：PB13 10：PB14 11：PB15 EXTI_BSEL[5:4] – 00：PB8 01：PB9 10：PB10 11：PB11 EXTI_BSEL[3:2] – 00：PB4 01：PB5 10：PB6 11：PB7 EXTI_BSEL[1:0] – 00：PB0 01：PB1 10：PB2 11：PB3
7:0	EXTI_ASEL	PortA EXTI 中断输入选择 EXTI_ASEL[7:6] – 00：PA12 01：PA13 10：PA14 11：PA15 EXTI_ASEL[5:4] – 00：PA8 01：PA9 10：PA10 11：PA11 EXTI_ASEL[3:2] – 00：PA4 01：PA5 10：PA6 11：PA7 EXTI_ASEL[1:0] –

Bit	助记符	功能描述
		00：PA0 01：PA1 10：PA2 11：PA3

34.9.14 EXTI 输入选择寄存器 1（GPIO_EXTISEL1）

名称	GPIO_EXTISEL1							
offset	0x1C4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_FSEL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_ESEL							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	EXTI_FSEL[7:0]	PortF EXTI 中断输入选择 EXTI_FSEL[7:6] – 00：- 01：- 10：- 11：- EXTI_FSEL[5:4] – 00：- 01：- 10：- 11：- EXTI_FSEL[3:2] – 00：PF4 01：PF5 10：PF6 11：- EXTI_FSEL[1:0] – 00：PF0 01：PF1 10：PF2 11：PF3
7:0	EXTI_ESEL	PortE EXTI 中断输入选择 EXTI_ESEL[7:6] – 00：PE12 01：PE13

Bit	助记符	功能描述
		10：PE14 11：PE15 EXTI_ESEL[5:4] – 00：PE8 01：PE9 10：PE10 11：PE11 EXTI_ESEL[3:2] – 00：PE4 01：PE5 10：PE6 11：PE7 EXTI_ESEL[1:0] – 00：PE0 01：PE1 10：PE2 11：PE3

34.9.15 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0）

名称	GPIO_EXTIEDS0							
offset	0x1C8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI15_EDS		EXTI14_EDS		EXTI13_EDS		EXTI12_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI11_EDS		EXTI10_EDS		EXTI9_EDS		EXTI8_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI7_EDS		EXTI6_EDS		EXTI5_EDS		EXTI4_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI3_EDS		EXTI2_EDS		EXTI1_EDS		EXTI0_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	

Bit	助记符	功能描述
31:30	EXTI15_EDS	EXTI[15]边缘触发选择 00: rising 01: falling 10: both 11: disable
…	…	
1:0	EXTI0_EDS	EXTI1[0] 边缘触发选择 00: rising 01: falling 10: both 11: disable

34.9.16 EXTI 边沿选择和使能寄存器 1（GPIO_EXTIEDS1）

名称	GPIO_EXTIEDS1							
offset	0x1CC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名								
位权限								
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名								
位权限								
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					EXTI21_EDS		EXTI20_EDS	
位权限					R/W-11		R/W-11	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI19_EDS		EXTI18_EDS		EXTI17_EDS		EXTI16_EDS	
位权限	R/W-11		R/W-11		R/W-11		R/W-11	

Bit	助记符	功能描述
11:10	EXTI21_EDS	EXTI[21]边缘触发选择 00: rising 01: falling 10: both 11: disable
…	…	
1:0	EXTI16_EDS	EXTI1[16] 边缘触发选择 00: rising 01: falling 10: both 11: disable

34.9.17 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）

名称	GPIO_EXTIDF							
offset	0x1D0							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		EXTI_DF[21:16]					
位权限	U-0		R/W -0000 0000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DF[15:8]							
位权限	R/W -0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DF[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:0	EXTI_DF	EXTI 输入数字滤波功能使能 0：关闭 EXTI 数字滤波 1：使能 EXTI 数字滤波

34.9.18 EXTI 中断标志（GPIO_EXTIISR）

名称	GPIO_EXTIISR							
offset	0x1D4							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		EXTI[21:16]					
位权限	U-0		R/W-00 0000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:0	EXTI	外部引脚中断标志寄存器，共可以产生 22 个引脚中断 硬件置位，软件写 1 清零

34.9.19 EXTI 输入信号寄存器（GPIO_EXTIDI）

名称	GPIO_EXTIDI							
offset	0x1D8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DI[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DI[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	EXTI_DI	EXTI 输入信号只读寄存器，软件读取此寄存器可以观察 EXTI

Bit	助记符	功能描述
		的 22 个输入信号的当前状态 *注：当使能了数字滤波后，软件可以从这个寄存器读取到某个 IO 输入信号滤波后的状态。*

34.9.20 FOUT 配置寄存器（GPIO_FOUTSEL）

名称	GPIO_FOUTSEL							
offset	0x200							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FOUT1SEL				FOUT0SEL			
位权限	R/W-0000				R/W-0000			

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:4	FOUT1SEL	PB14 或 PA7 输出选择 0000: XTLF 0001: RCLP 0010: ADCCLK 0011: LSCLK 0100: EOC 0101: RTCTM 0110: PLLO/64 0111: EOCAL 1000: APBCLK/64 1001: PLLO 1010: - 1011: RCHF 1100: XTHF/64 1101: ADCCLK/64 1110: CLK8K 1111: XTHF
3:0	FOUT0SEL	PC11 或 PC15 输出选择 0000: XTLF 0001: RCLP 0010: RCHF/64 0011: LSCLK 0100: AHBCLK/64 0101: RTCTM 0110: PLLO/64 0111: RTCCLK64Hz 1000: APBCLK/64

Bit	助记符	功能描述
		1001: PLLO 1010: - 1011: RCHF 1100: XTHF/64 1101: - 1110: CLK8K 1111: ADC_CLK

34.9.21 WKUP 控制寄存器（GPIO_PINWKEN）

名称	GPIO_PINWKEN							
offset	0x300							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKSEL	-			PINWKSEL			
位权限	R/W-0	U-0			R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PINWKSEL							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PINWKSEL						-	
位权限	R/W-00000000						U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN[7:0]							
位权限	R/W-00000000							

Bit	助记符	功能描述
31	WKSEL	WKUPx 唤醒中断入口选择 0：WKUPx 中断为 NMI 中断 1：WKUPx 中断为 39 号中断 【注】如果需要实现唤醒后不进入中断服务程序，直接运行主程序，可以将 WKSEL 设为 1，并置位 PRIMASK 寄存器后进入休眠；当 WKUPx 事件到来后，芯片会退出休眠模式，但是由于 PRIMASK=1，不会进入中断服务程序。
30:28	--	RFU：未实现，读为 0
27:10	PINWKSEL	WKUP 边沿选择 00：对应的 WKUP 引脚为下降沿唤醒 01：对应的 WKUP 引脚为上升沿唤醒 10/11：对应的 WKUP 引脚为上升、下降沿都能唤醒 寄存器对应顺序： Bit[11:10]对应 WKUP0 Bit[13:12]对应 WKUP1 Bit[15:14]对应 WKUP2 Bit[17:16]对应 WKUP3 Bit[19:18]对应 WKUP4 Bit[21:20]对应 WKUP5 Bit[23:22]对应 WKUP6 Bit[25:24]对应 WKUP7 Bit[27:26]对应 WKUP8

Bit	助记符	功能描述
9	--	RFU：未实现，读为 0
7:0	PINWKEN	WKUP 引脚使能信号 1：对应的 WKUP 引脚功能有效 0：对应的 WKUP 引脚功能无效 PINWKEN[x]控制 WKUPx 引脚的使能

35 安全功能（SFU）

35.1 概述

为了对应 IEC60730 和 IEC61508 安全标准，FM33FR0 内置了以下安全功能，目的是通过芯片自诊断设计，在故障和异常发生时，及时的停止工作。

- 闪存内容 CRC 检查（通过 DMA 实现，检查时 CPU 可以在 RAM 中运行程序代码）
- RAM 数据奇偶校验功能，当校验失败时可以选择产生中断或复位
- RAM 数据写保护功能
- 非法地址访问检测功能
- 频率检测功能
- ADC 测试功能

35.2 闪存 CRC 检查

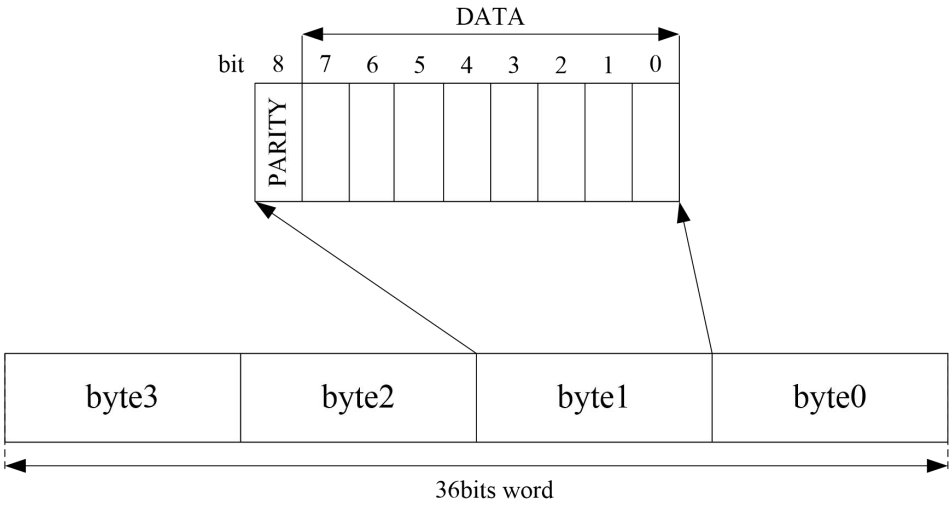
使用 DMA 实现对闪存内容的自检。为了不影响程序执行，启动闪存自检前应在 RAM 中执行代码，然后将 DMA 通道触发配置为 CRC，将源地址指向代码闪存，DMA 启动后，硬件自动将闪存内的数据搬运至 CRC 模块进行校验，完成全部闪存检验后，程序读取 CRC 结果并与预期值比较，以判断闪存内容是否存在问题。闪存检查的长度由 DMA 的 CHxTSIZE 寄存器控制，最大支持 384KB 闪存的全空间校验，用于检查程序完整性。

35.3 RAM 奇偶校验

FM33FR0 的 SRAM 支持硬件奇偶校验，每字节带有 1bit 校验位，以满足 IEC60730 标准要求的 RAM 检验功能。

在软件向 RAM 写入数据时，硬件自动以字节为单位计算校验位并与数据一同写入，软件读出时则同时读出校验位并进行实时校验，如果校验失败，则根据配置寄存器状态，产生校验错误中断或复位。

RAM 字宽增加到 36bit，每个字节内部 XOR 后形成校验位，保存在第 9bit。



注意：上电复位后 RAM 中的数据为乱码，此时读取极有可能触发校验错误中断，因此软件需在上电后首先对 RAM 内容进行初始化，再打开 RAM 校验功能。

上电复位后，SRAM 奇偶校验功能默认关闭，软件应完成 RAM 初始化之后再开启校验功能，并且可以配置校验错误中断或复位是否使能。

35.4 RAM 写保护

为了满足 IEC61508 的技术要求，RAM 写保护功能用于防止在 CPU 失控时改写 RAM 中的关键数据。RAM 写保护最多覆盖 1KB RAM 空间，以 256B 为颗粒度，被保护的空间不允许写入，只允许读出。

注意：RAM 写保护使能后，CPU、DMA、RAMBIST 都不能改写被保护地址。

35.5 关键寄存器写保护

为了保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护关键的控制寄存器，使其免遭改写。寄存器写保护用于保护 GPIO 控制寄存器、时钟控制寄存器、下电复位和电源检测控制寄存器、RAM 校验控制寄存器。

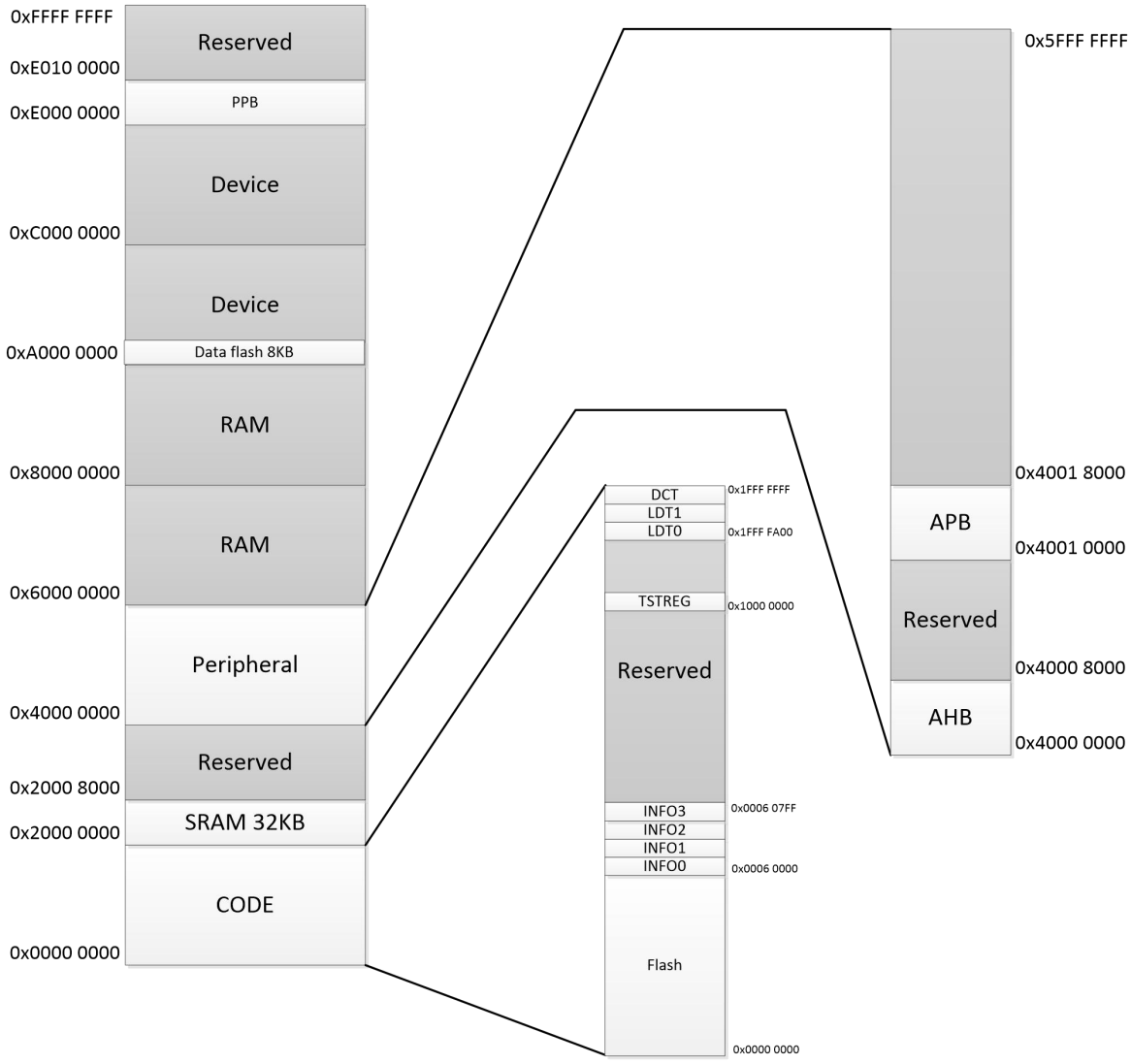
当写保护使能后，被保护的寄存器不能改写，但是仍可以正常读取。

参见存储访问保护控制寄存器（SFU_MAPCR）

35.6 非法地址访问检测

IEC60730 标准强制要求检查 CPU 和中断功能的运行是否正常，如果发现程序访问了系统中的空白地址，则认为程序执行异常，可以根据 MAP_SEL 寄存器配置产生异常访问中断（NMI 中断）或者复位。非法地址访问功能通过 MAP_EN 寄存器来使能或关闭。当软件对 MAP_EN 置位后，无法再清零，直到下一次芯片复位。

下图中，灰色地址区域禁止任何形式的访问（禁止读、写、取指），白色地址区域则具有不同的访问权限控制，详情参见后续表格。注意，此非法地址访问检测不区分 CPU 或 DMA 发起的访问。



地址访问权限表（FM33FR05x）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0005FFFF	O	O	O

属性	地址范围		Read	Write	Fetch
	min	Max			
Data Flash	0xA0000000	0xA0001FFF	O	O	X
Info Flash	0x00060000	0x000607FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20007FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB	0x40010000	0x4001FFFF	O	O	X
Others	-	-	X	X	X

表 35-1384K flash 版本总线访问控制表

地址访问权限表（FM33FR04x）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0003FFFF	O	O	O
Data Flash	0x00040000	0x000407FF	O	O	X
Info Flash	0x00060000	0x000607FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20001FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB	0x40010000	0x4001FFFF	O	O	X
Others	-	-	X	X	X

表 35-2256K flash 版本总线访问控制表

地址访问权限表（FM33FR02x）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0001FFFF	O	O	O
Data Flash	0x00040000	0x000407FF	O	O	X
Info Flash	0x00060000	0x000607FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X

属性	地址范围		Read	Write	Fetch
	min	Max			
SRAM	0x20000000	0x20000FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB	0x40010000	0x4001FFFF	O	O	X
Others	-	-	X	X	X

表 35-3 128K flash 版本总线访问控制表

35.7 时钟频率检测

IEC60730 标准强制要求检查系统时钟振荡频率是否正常。

通过多个定时器或者时钟校准模块都可以实现此功能。比如使用 LPTIM16，将其工作时钟配置为 APBCLK，将通道 1 输入捕获目标设置为 RCLP（32KHz），通过捕捉 RCLP 的周期，可以测量系统时钟频率是否正常。

TAU、ATIM、BSTIM16 时钟校准电路，也都可以实现类似功能，详情参见相关章节。

35.8 ADC 测试

IEC60730 标准强制要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正（+）基准电压、负（–）基准电压、模拟输入通道（ADC_Inx）、温度传感器输出电压以及内部基准电压进行 A/D 转换来确认 A/D 转换器的正常运行。

通过以下步骤确认模拟多路转换器：

1.选择 ADC_INx 引脚作为 A/D 转换对象

2.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-1）

3.选择 A/D 转换器的负（– ）基准电压作为 A/D 转换对象

4.对 A/D 转换器的负（–）基准电压进行 A/D 转换（转换结果 2-1）

5.选择 ADC_INx 引脚作为 A/D 转换对象

6.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-2）

7.选择 A/D 转换器的正（+）基准电压作为 A/D 转换对象

8.对 A/D 转换器的正（+）基准电压进行 A/D 转换（转换结果 2-2）

9.选择 ADC_INx 引脚作为 A/D 转换对象

10.对 ADC_INx 引脚进行 A/D 转换（转换结果 1-3）

11.确认“转换结果 1-1”=“转换结果 1-2”=“转换结果 1-3”

12.确认“转换结果 2-1”的 A/D 转换结果全部为“0”，“转换结果 2-2”的 A/D 转换结果全部为“1”。

通过以上步骤，能确认已选择模拟多路转换器以及布线没有断线。

35.9 寄存器

安全功能模块基地址：0x4000_1C00

offset 地址	名称	符号
0x00	RAM 奇偶校验控制寄存器 (RAM Parity Control Register)	SFU_RAMPCR
0x04	RAM 奇偶校验状态寄存器 (RAM Parity Status Register)	SFU_RAMPSR
0x08	存储器访问保护控制寄存器 (Memory Access Protection Control Register)	SFU_MAPCR
0x0C	存储器访问保护状态寄存器 (Memory Access Protection Control Register)	SFU_MAPSR

35.9.1 RAM 奇偶校验控制寄存器（SFU_RAMPCR）

名称	SFU_RAMPCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					RAMP_RE	RAMP_IE	-
位权限	U-0					R/W-0	R/W-0	U-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	RAMP_RE	RAM 奇偶校验复位使能 (RAM parity reset enable) 1：奇偶校验错误发生时复位芯片 0：奇偶校验错误发生时不复位芯片
1	RAMP_IE	RAM 奇偶校验中断使能 (RAM parity interrupt enable) 1：奇偶校验错误发生时产生中断 0：奇偶校验错误发生时不产生中断
0	--	RFU：未实现，读为 0

35.9.2 RAM 奇偶校验状态寄存器（SFU_RAMPSR）

名称	SFU_RAMPSR
offset	0x04

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						RAMP_IF	-
位权限	U-0						R/W-0	U-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	RAMP_IF	RAM 奇偶校验错误标志 (RAM parity interrupt flag)，硬件置位，软件写 1 清零 1：发生了奇偶校验错误 0：未发生奇偶校验错误
0	--	RFU：未实现，读为 0

35.9.3 存储访问保护控制寄存器（SFU_MAPCR）

名称	SFU_MAPCR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MAP_SEL	RAMPP	SVDP	CCP	GPIOP	RAMWP		MAP_EN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7	MAP_SEL	非法地址访问发生时的系统行为配置 (Memory Access Protection Select) 1：非法地址访问发生时，复位芯片 0：非法地址访问发生时，产生 NMI
6	RAMPP	RAM 校验保护(RAM Parity Protection) 1：RAM 奇偶校验功能使能后不可关闭，直到芯片复位

Bit	助记符	功能描述
		0：无保护
5	SVDP	SVD/BOR 控制寄存器保护(SVD and BOR Protection) 1：SVD 和 BOR 控制寄存器不可改写 0：无保护
4	CCP	CMU 控制寄存器保护(Clock Control Protection) 1：CMU 模块的停振检测控制、系统时钟配置、时钟调校寄存器不可改写 0：无保护
3	GPIOP	GPIO 控制寄存器保护 (GPIO Protection) 1：GPIO 模块的所有控制寄存器不能改写 0：无保护
2:1	RAMWP	RAM 写保护(RAM Write Protection) 00：无写保护 01：RAM 起始地址开始的 256 字节禁止写入 10：RAM 起始地址开始的 512 字节禁止写入 11：RAM 起始地址开始的 1024 字节禁止写入
0	MAP_EN	存储器访问保护使能(Memory Access Protection Enable) 1：使能非法地址访问检测功能 0：不使能非法地址访问检测 *注意：此寄存器一旦置位，软件不能对其清零。*

35.9.4 存储访问保护状态寄存器（SFU_MAPSR）

名称	SFU_MAPSR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							MAP_IF
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	MAP_IF	存储器访问保护中断标志 (Memory Access Protection Interrupt flag)，硬件置位，软件写 1 清零。此标志置位后会触发 NMI 中断。 1：发生了非法地址访问 0：未发生非法地址访问

36 专用编程接口

36.1 概述

FM33FR0芯片可使用复旦微电子所提供的专用编程器，或者通过Bootloader下载用户程序。编程器通过专用编程接口(SWD)与芯片通信，完成程序下载，并可对Flash全空间内容进行Checksum校验。

36.2 编程器使用

编程器的使用方法请参考应用手册，或联系复旦微电子公司。

37 调试支持

37.1 概述

FM33FR0芯片基于ARM Cortex-M0处理器构建，并支持相应的debug特性。通过硬件断点（breakpoint）和数据观察点（watchpoint），调试器可以在特定指令取指和数据访问时停止CPU内核运行，检视内核寄存器和系统外设状态，并根据需要恢复内核运行。

仿真调试主机通过SWD接口与FM33FR0芯片互联，并实现仿真调试。

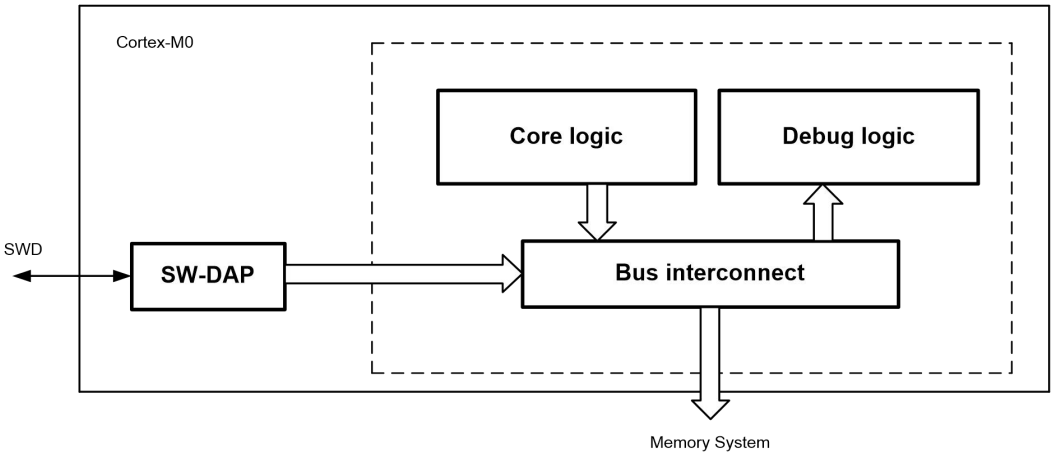


图 37-1 Cortex-M0+调试系统示意图

关于Cortex-M0+内核的debug特性，请参考ARM公司的Cortex-M0技术参考手册。

37.2 Debug 引脚

37.2.1 SWD 引脚

FM33FR0系列MCU的SWD引脚位置如下表：

SWD pins	Debug功能	引脚定义
SWDIO	SWD数据输入/输出	PA4
SWCLK	SWD时钟输入	PA3

注意：芯片复位后PA3和PA4都默认为数字输入状态，与大部分GPIO不同。

37.2.2 上拉电阻

芯片复位后，SWDIO和SWCLK引脚默认使能内部上拉（约50K欧姆），用户无需在PCB上外接上拉电阻。

37.3 SWD 接口协议

37.3.1 协议简介

SWD协议采用LSB-first进行数据收发。通过SWD接口，调试主机可以读写DPACC和APACC寄存器组。

SWIO每次切换数据方向时，总线上需插入turn-around时间，这段时间内主机和从机都不会驱动SWIO。在两次传输之间，主机必须将线驱动为低电平进入idle状态，或继续发送一次新传输的起始位继续传输，在一次数据包传输之后，主机也可以空闲，使线保持为高电平或由上拉电阻上拉。SWD协议没有明确的复位信号，在没有看到预期的信号时，主机或目标机将对复位进行检测。通过保持线为高电平持续50个时钟周期之后跟随一个读ID的请求，可以确保在检测到错误或复位之后重新同步成功。

37.3.2 传输序列

SWD每个通信传输序列包含三个部分：

1、包请求（8bits），由主机发送

2、ACK响应（3bits），由从机回发

3、数据传输（33bits），由主机或从机发送

其中包请求字节定义如下：

Bit	Name	描述
0	Start	起始位，必须是1
1	ApnDP	AP/DP选择 0：DP访问 1：AP访问
2	RnW	读写选择 0：写请求 1：读请求
4:3	A[3:2]	DP/AP寄存器的地址域
5	Parity	Bit0~Bit4数据的校验位
6	Stop	0
7	Park	主机不驱动，通过总线上拉，从机读为1

包请求发送后，总线上总是有1bit的turn-around时间。

ACK响应定义如下：

Bit	Name	描述
0:2	ACK	001：FAULT 010：WAIT 100：OK

如果主机发起读操作，或者ACK为WAIT或FAULT，则ACK之后必须插入turn-around时间。

数据传输格式如下：

Bit	Name	描述
0:31	Data	读出或写入的数据
32	Parity	32bit 数据的校验位

37.3.3 SW-DP ID code

Cortex-M0的SW-DP有一个固定的ID code：0x0BB11477

SW-DP处于非活跃状态，直到主机读取ID code。

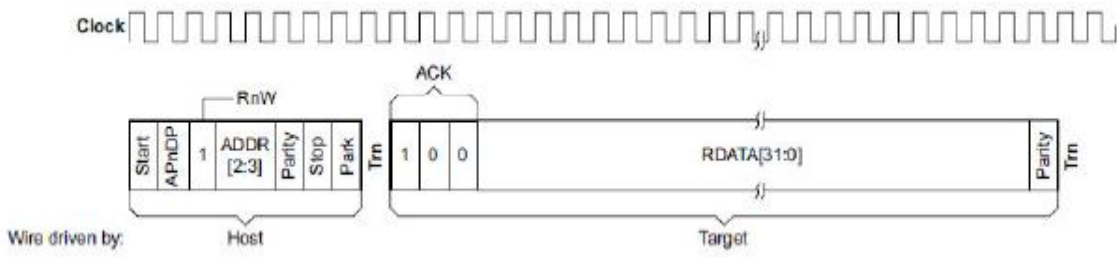
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

37.3.4 主机读操作

一次成功的读操作由以下三个阶段组成

- 一个8位的读数据包请求（request），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，WAIT响应为010，FAULT响应为001.
- 一个33位的数据读阶段（payload），从主机到目标。

默认情况下，在第一和第二阶段之间以及第三阶段之后有一个时钟的掉转周期，一次成功的读操作如下图。

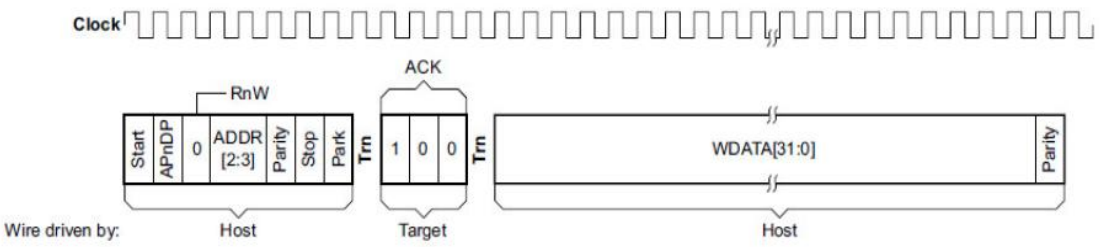


37.3.5 主机写操作

一次写操作由以下三个阶段组成

- 一个8位的写数据包请求（header），从主机到目标。
- 一个3位的应答（ack），从目标到主机。成功的OK响应为100，FAULT响应为001.
- 一个33位的数据写阶段（payload），从主机到目标。

默认情况下，每两个阶段之间都有一个时钟的掉转周期，一次成功的写操作如下图。



37.4 SWD-DP 寄存器

37.4.1 寄存器列表

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
		ABORT	WO
01	0x0	CTRL/STAT	RW
	0x1	DLCR	RW
10	x	RESEND	RO
		SELECT	WO
11	x	RDBUFF	RO

关于寄存器的详细说明，请参考Cortex-M0 Technical Reference Manual.

37.5 Core debug 寄存器

通过操作core debug寄存器可以实现内核调试。主机通过SW-DP访问以下内核调试寄存器。

Address	Name	Type	Function
0xE000EDF0	DHCSR	RW	Debug Halting Control and Status Register
0xE000EDF4	DCRSR	WO	Debug Core Register Selector Register
0xE000EDF8	DCRDR	RW	Debug Core Register Data Register
0xE000EDFC	DEMCR	RW	Debug Exception and Monitor Control Register
0xE000EE00 to 0xE000EEFF	-	-	Reserved for Debug Extension

上述debug寄存器不被系统复位影响，仅受上电复位影响。通过以下方式可以实现CPU复位后立即halt：

- 置位DEMCR寄存器的bit0 (VC_CORRESET)
- 置位DHCSR寄存器的bit0 (C_DEBUGEN)
- 执行系统复位

37.6 低功耗调试支持

通常情况下，当芯片进入Sleep/DeepSleep模式时，CPU的FCLK和HCLK都会被关闭，这样会导致host无法保持与芯片的debugger连接。为了支持低功耗模式下的debug调试，当CPU连接debugger时，FCLK和HCLK必须保持运行，即此时芯片不会真正进入休眠，调试中应注意这一点。

37.7 Debug 相关的配置项

通过配置DBG_CR寄存器，可以设置在调试状态下，芯片内部的定时器、看门狗电路是否继续工作。详情请参见DBG_CR配置寄存器。

37.8 寄存器

地址	名称	符号
0x40000004	MCU Debug 配置寄存器	DBG_CR
0x40000008	HARDFAULT 查询寄存器	HDFR

37.8.1 MCU DEBUG 配置寄存器

FM33FR0 扩展了 DBG_CR 寄存器，用于配置 Debug 状态下的看门狗和定时器。DBG_CR 寄存器可以由 SWD 接口或软件改写。

名称	DBG_CR							
地址	0x40000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						DBG_RCEN	DBG_SLEEP
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT16_STOP	DBG_BT16_STOP	-	-	-	-	-	-
位权限	R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	-	DBG_ATIM_STOP	DBG_TAU1_STOP	DBG_TAU0_STOP	DBG_WWDT_STOP	DBG_IWDT_STOP
位权限	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

Bit	助记符	功能描述
31:18	--	RFU：未实现，读为 0
17	DBG_RCEN	Debug 模式下是否支持标志寄存器读清零操作 0：debug 禁止标志读清零操作 1：debug 允许标志读清零操作
16	DBG_SLEEP	休眠模式下的 debug 配置 0：正常进入休眠，关闭 HCLK 和 FCLK，debugger 无法保持连接 1：不关闭 HCLK 和 FCLK，debugger 能够在休眠模式下保持连接
15	DBG_LPT1	Debug 状态下 LPTIM16 使能控制位

Bit	助记符	功能描述
	6_STOP	1：Debug 时关闭 LPTIM16 0：Debug 时保持 LPTIM16 原来状态
14	DBG_BT16_STOP	Debug 状态下 BSTIM16 使能控制位 1：Debug 时关闭 BSTIM16 0：Debug 时保持 BSTIM16 原来状态
13:5	--	RFU：未实现，读为 0
4	DBG_ATIM_STOP	Debug 状态下 ATIM 使能控制位 1：Debug 时关闭 ATIM 0：Debug 时保持 ATIM 原来状态
3	DBG_TAU1_STOP	Debug 状态下 TAU1 使能控制位 1：Debug 时关闭 TAU1 0：Debug 时保持 TAU1 原来状态
2	DBG_TAU0_STOP	Debug 状态下 TAU0 使能控制位 1：Debug 时关闭 TAU0 0：Debug 时保持 TAU0 原来状态
1	DBG_WWDT_STOP	Debug 状态下 WWDT 使能控制位 1：Debug 时关闭 WWDT 0：Debug 时保持 WWDT 原来状态
0	DBG_IWDT_STOP	Debug 状态下 IWDT 使能控制位 1：Debug 时关闭 IWDT 0：Debug 时保持 IWDT 开启

37.8.2 HardFault 查询寄存器

名称	HDFR							
地址	0x40000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	SVCUNDEF_FLAG	BKPT_FLAG	TBIT_FLAG	DFLS_FLAG	HDF_REQUEST_FLAG
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	DABORT_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 1：总线传输中访问了非法地址导致 HRESP 为高产生错误

Bit	助记符	功能描述
		0：未访问非法地址
4	SVCUNDEF_FLAG	SVC instructions 未定义标志，写 1 清零 if the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set, the core should treat SVC instructions as though theywere UNDEFINED。
3	BKPT_FLAG	执行 BKPT 指令标志，写 1 清零 1：执行了 BKPT 指令 0：未执行 BKPT 指令
2	TBIT_FLAG	Thumb-State 标志，写 1 清零 1：切换到 ARM 状态 0：处于 Thumb-State
1	DFLS_FLAG	DFLASH 取指标志，写 1 清零 1：执行了特殊指令代码，如试图在 DFLASH 区域内取指 0：无指令代码试图在 DFLASH 被执行
0	HDF_REQUEST_FLAG	hardfault 标志位，任何类型的 hardfault 都会导致该位置位，写 1 清零 1：hardfault 请求 0：无 hardfault 请求

38 器件签名信息

每一颗FM33FR0系列MCU都有自己的器件签名，包括存储器容量信息和唯一器件ID号。

38.1 唯一器件 ID

唯一器件ID保存在Flash中，可以通过查询0x1FFF_FA20~0x1FFF_FA2B中的数据来识别。

这12个字节中分别保存了芯片的Lot ID、Wafer ID、芯片在wafer上的坐标位置。通过这个数据，保证每颗芯片都有独立不重复的识别码。

版本列表

版本号	发布日期	页数	章节或图表	更改说明
1.0	2021.10			initial release
1.1	2022.02	780		DMA外设传输长度修改为8192次 增加使用flash ECC功能的说明
1.2	2022.04	780		更新LQFP64封装引脚图
1.3	2022.9			修订35.9.2 GPIO上下拉使能寄存器
1.4	2022.10		2.1 3	增加LQFP32封装 高温电参数更新到105C
1.5	2022.11	719		修正若干描述
1.6	2022.12		2.1.8 3.2	修正PC2的数字外设功能 增加推荐电源方案
1.7	2023.3			增加0.65mm pitch的LQFP64封装（14x14）

上海复旦微电子集团股份有限公司销售及服务网点

上海复旦微电子集团股份有限公司
地址：上海市国泰路 127 号 4 号楼
邮编：200433
电话：(86-021) 6565 5050
传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司
地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室
电话：(852) 2116 3288　2116 3338
传真：(852) 2116 0882

北京办事处
地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室
邮编：100007
电话：（86-10）8418 6608
传真：（86-10）8418 6211

深圳办事处
地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室
邮编：518028
电话：(86-0755) 8335 0911　8335 1011　8335 2011 8335 0611
传真：(86-0755) 8335 9011

台湾办事处
地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室
电话：(886-2) 7721 1889
传真：(886-2) 7722 3888

新加坡办事处
地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929
电话：(65) 6472 3688
传真：(65) 6472 3669

北美办事处
地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA
电话：(480) 857-6500 ext 18

公司网址：http://www.fmsh.com/