Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33LF0A
低功耗 MCU 芯片

产品说明书

2024. 08

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。

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 **23**
图目录 **25**
1 产品综述 **33**
1.1 概述 33
1.2 芯片结构框图 36
1.3 产品型号列表 37
2 引脚和封装 **37**
2.1 封装和引脚排列 37
2.1.1 LQFP64 封装图（FM33LF0x6A） *37*
2.1.2 LQFP48 封装图（FM33LF0x5A） *38*
2.1.3 QFN32 封装图（FM33LF0x3A） *39*
2.1.4 引脚功能定义 *39*
2.1.5 PA 引脚组数字外设功能 *47*
2.1.6 PB 引脚组数字外设功能 *48*
2.1.7 PC 引脚组数字外设功能 *48*
2.1.8 PD 引脚组数字外设功能 *49*
2.1.9 封装尺寸图 *50*
2.2 焊接安装说明 53
2.3 MSL 等级 55
2.4 热阻特性 56
3 电参数 **57**
3.1 参数说明 57
3.2 参数测试条件 57
3.2.1 供电方案 *57*
3.3 极限参数 58
3.4 性能参数 58
3.4.1 典型工作条件 *58*
3.4.2 功耗参数 *58*
3.4.3 复位和电源监控 *59*
3.4.4 高精度基准源 *61*
3.4.5 低功耗模式唤醒时间 *61*
3.4.6 外部时钟源特性 *62*
3.4.7 内部时钟源特性 *63*
3.4.8 PLL 特性 *64*
3.4.9 ADC 特性 *65*
3.4.10 DAC 特性 *68*
3.4.11 温度传感器 *68*
3.4.12 OPA 特性 *69*
3.4.13 模拟比较器特性 *70*
3.4.14 Flash 存储器特性 *71*
3.4.15 GPIO 特性 *71*
3.4.16 LCD 特性 *72*
4 总线与存储 **73**
4.1 系统总线 73

4.2 存储空间分配 74
4.2.1 概述 *74*
4.2.2 外设模块寄存器地址分配 *76*
4.3 RAM 77
4.3.1 概述 *77*
4.3.2 奇偶校验 *77*
4.4 FLASH 78
4.4.1 概述 *78*
4.4.2 特殊信息扇区说明 *78*
4.4.3 ECC 中断 *80*
4.4.4 Flash 编程 *80*
4.4.5 Flash 的内容保护 *83*
4.5 寄存器 86
4.5.1 Flash 读取控制寄存器（FLS_RDCR） *86*
4.5.2 预取指控制寄存器（FLS_PFCR） *87*
4.5.3 用户配置字寄存器（FLS_OPTBR） *87*
4.5.4 ACLOCK 寄存器 1（FLS_ACLOCK1） *88*
4.5.5 Flash 擦写控制寄存器（FLS_EPCR） *89*
4.5.6 Flash Key 输入寄存器（FLS_KEY） *89*
4.5.7 Flash 中断使能寄存器（FLS_IER） *90*
4.5.8 Flash 标志寄存器（FLS_ISR） *90*
4.5.9 Flash ECC 控制寄存器（FLS_ECCCR） *92*
4.5.10 Flash ECC 配置寄存器（FLS_ECCR） *92*

5 电源管理单元（PMU） **93**

5.1 芯片工作电源 93
5.1.1 电源域划分 *93*
5.1.2 电源结构图 *94*
5.1.3 ADC 和基准电压 *94*
5.2 功耗模式 95
5.2.1 概述 *95*
5.2.2 功耗模式与系统频率 *96*
5.2.3 Active 模式 *96*
5.2.4 SLEEP 模式 *96*
5.2.5 DEEPSLEEP 模式 *97*
5.3 唤醒源 98
5.4 休眠唤醒后的时钟控制 98
5.5 寄存器 99
5.5.1 低功耗控制寄存器（PMU_CR） *99*
5.5.2 唤醒时间控制寄存器（PMU_WKTR） *100*
5.5.3 唤醒源标志查询寄存器（PMU_WKFR0） *100*
5.5.4 唤醒源标志查询寄存器（PMU_WKFR1） *102*
5.5.5 模拟 BUFFER 控制寄存器（PMU_BUF_CR） *102*
5.5.6 PTAT 控制寄存器（PMU_PTAT_CR） *103*

6 权限管理（MDC） **104**

6.1 寄存器 104
6.1.1 HardFault 查询寄存器 *104*

7 处理器（CPU） **106**

7.1 概述 106
7.1.1 处理器配置 *106*
7.2 寄存器 107
7.3 异常和中断 107

7.3.1 中断向量表 108
7.3.2 中断优先级 109
7.3.3 错误处理 109
7.3.4 锁定（Lockup） 109
7.4 调试特性 110
7.4.1 调试功能引脚 110
7.4.2 调试状态下的看门狗控制 110
7.4.3 DEBUG 的复位 110

8 复位管理单元（RMU） 111

8.1 概述 111
8.2 模块框图 111
8.3 上下电复位 112
8.4 软件复位 112
8.5 NRST 引脚复位 112
8.6 寄存器 113
8.6.1 PDR 配置寄存器（RMU_PDRCR） 113
8.6.2 BOR 配置寄存器（RMU_BORCR） 113
8.6.3 复位配置寄存器（RMU_LKPCR） 114
8.6.4 软件复位寄存器(RMU_SOFTRST) 115
8.6.5 复位标志寄存器(RMU_ RSTFLAG) 115
8.6.6 外设复位使能寄存器(RMU_PRSTEN) 116
8.6.7 AHB 外设复位寄存器(RMU_AHBRSTCR) 116
8.6.8 APB 外设复位寄存器 1(RMU_APBRSTCR1) 117
8.6.9 APB 外设复位寄存器 2(RMU_APBRSTCR2) 119

9 独立看门狗（IWDT） 121

9.1 概述 121
9.2 结构框图 121
9.3 IWDT 功能描述 121
9.4 IWDT 窗口功能 122
9.5 IWDT 冻结 123
9.6 寄存器 124
9.6.1 IWDT 清除寄存器（IWDT_SERV） 124
9.6.2 IWDT 配置寄存器（IWDT_CR） 124
9.6.3 IWDT 计数值寄存器（IWDT_CNT） 125
9.6.4 IWDT 窗口寄存器（IWDT_WIN） 126
9.6.5 IWDT 中断使能寄存器（IWDT_IER） 126
9.6.6 IWDT 中断标志寄存器（IWDT_ISR） 126

10 窗口看门狗（WWDT） 128

10.1 功能描述 128
10.2 结构框图 128
10.3 WWDT 工作方式 129
10.4 寄存器 130
10.4.1 WWDT 控制寄存器（WWDT_CR） 131
10.4.2 WWDT 配置寄存器（WWDT_CFGR） 131
10.4.3 WWDT 计数寄存器（WWDT_CNT） 132
10.4.4 WWDT 中断使能寄存器（WWDT_IER） 132
10.4.5 WWDT 中断标志寄存器（WWDT_ISR） 133
10.4.6 WWDT 预分频寄存器（WWDT_PSC） 133

11 时钟管理单元（CMU） 135

11.1 概述 135

11.2 时钟架构 135
11.2.1 时钟树 *135*
11.2.2 SYSCLK 切换说明 *136*
11.2.3 时钟安全性 *137*
11.2.4 主要时钟说明 *137*
11.2.5 外设模块的总线时钟和工作时钟 *138*
11.2.6 休眠模式下的外设时钟 *139*
11.2.7 LSCLK *139*
11.2.8 SYSTICK 时钟 *139*
11.3 高频 RC 振荡器(RCHF) 139
11.3.1 概述 *139*
11.3.2 软件使用指南 *140*
11.4 低功耗 RC 振荡器(RCLP) 140
11.4.1 概述 *140*
11.5 高频晶体振荡电路(XTHF) 140
11.5.1 概述 *140*
11.5.2 工作方式 *140*
11.5.3 停振检测（HFDET） *140*
11.6 低频晶体振荡电路(XTLF) 141
11.6.1 概述 *141*
11.6.2 工作方式 *141*
11.6.3 停振检测 *141*
11.7 锁相环(PLL) 142
11.7.1 概述 *142*
11.7.2 软件使用指南 *142*
11.8 低功耗模式下的时钟源 142
11.9 休眠唤醒的时钟处理 142
11.10 寄存器 143
11.10.1 系统时钟配置寄存器（CMU_SYSCLKCR） *143*
11.10.2 RCHF 时钟控制寄存器（CMU_RCHFCR） *145*
11.10.3 RCHF 调校寄存器（CMU_RCHFTR） *145*
11.10.4 PLL 控制寄存器（CMU_PLLCR） *146*
11.10.5 RCLP 控制寄存器（CMU_RCLPCR） *147*
11.10.6 RCLP 调校寄存器（CMU_RCLPTR） *147*
11.10.7 XTLF 控制寄存器（CMU_XTLFCR） *148*
11.10.8 LSCLK 选择寄存器（CMU_LSCLKSEL） *149*
11.10.9 XTHF 控制寄存器（CMU_XTHCR） *149*
11.10.10 CMU 中断使能寄存器（CMU_IER） *150*
11.10.11 CMU 中断标志寄存器（CMU_ISR） *151*
11.10.12 外设总线时钟控制寄存器 1（CMU_PCLKCR1） *151*
11.10.13 外设总线时钟控制寄存器 2（CMU_PCLKCR2） *152*
11.10.14 外设总线时钟控制寄存器 3（CMU_PCLKCR3） *153*
11.10.15 外设总线时钟控制寄存器 4（CMU_PCLKCR4） *154*
11.10.16 外设工作时钟配置寄存器 1（CMU_OPCCR1） *154*
11.10.17 外设工作时钟配置寄存器 2（CMU_OPCCR2） *156*
11.10.18 外设工作时钟使能寄存器（CMU_OPCER1） *157*
11.10.19 外设工作时钟使能寄存器（CMU_OPCER2） *158*
11.10.20 AHB Master 控制寄存器（CMU_AHBMCR） *159*

12 时钟监控（CLM） **160**

12.1 概述 160
12.2 结构框图 160
12.3 功能描述 161

12.3.1 工作模式 *161*
12.3.2 使能与关闭 *161*
12.3.3 时钟频率监控 *161*
12.3.4 超时检测 *164*
12.4 寄存器 165
12.4.1 时钟监控控制寄存器（CLM_CR） *165*
12.4.2 时钟监控配置寄存器（CLM_CFGR） *166*
12.4.3 时钟监控计数值寄存器（CLM_CNTR） *166*
12.4.4 时钟监控中断标志寄存器（CLM_ISR） *167*
12.4.5 时钟监控比较高阈值寄存器（CLM_CMPH） *167*
12.4.6 时钟监控比较低阈值寄存器（CLM_CMPL） *168*

13 电源电压监测（SVD） **169**

13.1 概述 169
13.2 结构框图 169
13.3 引脚定义 170
13.4 时钟和复位 170
13.5 功能描述 170
13.5.1 间歇使能模式 *171*
13.5.2 外部电源检测 *172*
13.6 电源检测阈值 172
13.7 寄存器 174
13.7.1 SVD 配置寄存器（SVD_CFGR） *174*
13.7.2 SVD 控制寄存器（SVD_CR） *175*
13.7.3 SVD 中断使能寄存器（SVD_IER） *176*
13.7.4 SVD 状态和标志寄存器（SVD_ISR） *176*
13.7.5 SVD 参考电压选择寄存器（SVD_VSR） *177*

14 AES 硬件运算单元（AES） **178**

14.1 功能描述 178
14.2 工作模式 178
14.3 AES 数据流处理模式 179
14.3.1 ECB 模式 *179*
14.3.2 CBC 模式 *180*
14.3.3 暂停模式 *182*
14.3.4 CTR 模式 *183*
14.3.5 CTR 模式下的暂停模式 *184*
14.3.6 GCM 模式 *184*
14.3.7 MultH 模块 *187*
14.3.8 推荐的 GCM 流程 *188*
14.4 数据类型 189
14.5 工作流程 190
14.5.1 模式 1：加密 *190*
14.5.2 模式 2：密钥扩展 *191*
14.5.3 模式 3：解密 *192*
14.5.4 模式 4：密钥扩展+解密 *192*
14.5.5 使用 MultH 模块 *193*
14.6 DMA 接口 194
14.6.1 MultH 模块与 DMA 间接口 *195*
14.7 错误标志 195
14.8 寄存器 196
14.8.1 AES 控制寄存器（AES_CR） *197*
14.8.2 AES 中断使能寄存器（AES_IER） *198*

14.8.3 AES 中断标志寄存器（AES_ISR）.....199
14.8.4 AES 数据输入寄存器（AES_DIR）.....199
14.8.5 AES 数据输出寄存器（AES_DOR）.....200
14.8.6 AES 秘钥寄存器（AES_KEYx）.....200
14.8.7 AES 初始向量寄存器（AES_IVRx）.....201
14.8.8 AES MultH 参数寄存器（AES_Hx）.....201

15 随机数发生器（TRNG）.....202

15.1 概述.....202
15.2 功能描述.....202
15.2.1 随机数产生.....202
15.2.2 工作时钟.....203
15.2.3 随机数读取.....203
15.2.4 CRC 运算.....203
15.3 寄存器.....204
15.3.1 随机数控制寄存器(RNG_CR).....204
15.3.2 随机数/CRC 结果输出寄存器(RNG_DOR).....204
15.3.3 RNG 标志寄存器(RNG_SR).....205
15.3.4 CRC 控制寄存器(RNG_CRCCR).....205
15.3.5 CRC 输入数据寄存器(RNG_CRCDIR).....206
15.3.6 CRC 标志寄存器(RNG_CRCSR).....206

16 模拟比较器（COMPARATOR）.....208

16.1 概述.....208
16.2 结构框图.....209
16.3 功能描述.....209
16.3.1 基本功能.....209
16.3.2 内部比较基准选择和产生.....209
16.3.3 时钟和复位.....210
16.3.4 引脚和内部信号连接.....210
16.3.5 比较器输出滞回.....211
16.3.6 输出屏蔽.....212
16.3.7 输出数字滤波.....213
16.3.8 功耗和速度模式.....213
16.3.9 比较器中断.....213
16.3.10 比较器输出和触发信号输出.....214
16.4 寄存器.....216
16.4.1 COMP1 控制寄存器（COMP1_CR）.....216
16.4.2 COMP2 控制寄存器（COMP2_CR）.....218
16.4.3 COMP3 控制寄存器（COMP3_CR）.....219
16.4.4 COMP4 控制寄存器（COMP4_CR）.....221
16.4.5 COMP 中断配置寄存器（COMP_ICR）.....223
16.4.6 COMP 中断标志寄存器（COMP_IF）.....224
16.4.7 COMP 缓冲器控制寄存器（COMP_BUFCR）.....224

17 除法/开方运算加速器（DIVAS）.....226

17.1 概述.....226
17.2 时钟和复位.....226
17.3 硬件除法工作流程.....226
17.4 硬件开方工作流程.....227
17.5 计算时间配置.....227
17.6 寄存器.....228
17.6.1 操作数寄存器(DIVAS_OPRD).....228

17.6.2 除数寄存器(DIVAS_DIVSOR) *229*
17.6.3 商寄存器(DIVAS_QUOT) *229*
17.6.4 余数寄存器(DIVAS_REMD) *229*
17.6.5 根寄存器(DIVAS_ROOT) *230*
17.6.6 状态标志寄存器(DIVAS_SR) *230*
17.6.7 控制寄存器(DIVAS_CR) *231*

18 双线串行总线和系统管理总线（I²C-SMBUS） **233**

18.1 概述 233
18.2 主要功能 233
18.3 模块框图 234
18.3.1 I2C-SMB 模块框图 *234*
18.4 引脚定义 235
18.5 时钟和复位 235
18.6 接口时序 236
18.6.1 通信过程 *236*
18.6.2 接口时序描述 *237*
18.6.3 总线时序参数表 *237*
18.7 I2C 功能描述 239
18.7.1 预设从机地址表 *239*
18.7.2 I2C 初始化过程 *240*
18.7.3 I2C 主机功能 *241*
18.7.4 I2C 从机功能 *258*
18.8 SMBus 功能描述 266
18.8.1 概述 *266*
18.8.2 SMBus 初始化过程 *269*
18.8.3 SMBus 主机功能 *270*
18.8.4 SMBus 超时功能 *272*
18.8.5 CRC-8 ..... *273*
18.9 寄存器 274
18.9.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1） *274*
18.9.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2） *276*
18.9.3 I2C_SMB 中断控制寄存器（I2CSMB_IER） *277*
18.9.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR） *278*
18.9.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR） *279*
18.9.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TIMINGR） *279*
18.9.7 I2C_SMB 超时寄存器（I2CSMB_TOR） *280*
18.9.8 I2C_SMB 接收数据寄存器（I2CSMB_RXDR） *281*
18.9.9 I2C_SMB 发送数据寄存器（I2CSMB_TXDR） *281*
18.9.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR） *282*
18.9.11 CRC 数据寄存器（I2CSMB_CRC_DR） *282*
18.9.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR） *283*
18.9.13 CRC 多项式寄存器（I2CSMB_CRC_POLY） *283*

19 通用异步收发传输器（UART） **285**

19.1 概述 285
19.2 结构框图 286
19.3 引脚定义 287
19.4 UART 类型区分 287
19.5 UART 字符描述 287
19.5.1 智能卡模式 *288*
19.6 功能描述 290
19.6.1 时钟结构 *290*

19.6.2 同步策略 290
19.6.3 位接收采样 291
19.6.4 数据发送 292
19.6.5 数据接收 295
19.6.6 RS232 硬件流控 296
19.6.7 低功耗休眠唤醒 297
19.6.8 使用 DMA 进行 UART 收发 298
19.6.9 DMA 模式下的发送完成中断 298
19.6.10 半双工单线通信 299
19.7 波特率发生 299
19.7.1 波特率发生 299
19.7.2 波特率自适应 300
19.8 红外调制 300
19.9 接收超时 301
19.10 发送延迟 302
19.11 LIN 总线通信功能 302
19.11.1 概述 302
19.11.2 LIN 主机操作 305
19.11.3 LIN 从机操作 305
19.11.4 LIN 接收采样与噪声检测 312
19.11.5 LIN 错误状态 313
19.11.6 LIN 自测试模式 317
19.12 寄存器 319
19.12.1 红外调制寄存器（UART_IRCR） 322
19.12.2 UARTx 模式寄存器（UARTx_MCR） 322
19.12.3 UARTx LIN 控制寄存器（UARTx_LINCR） 323
19.12.4 UARTx LIN 波特率同步寄存器（UARTx_LINBSR） 324
19.12.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR） 325
19.12.6 UARTx LIN 发送时序寄存器（UARTx_LINTTR） 325
19.12.7 UARTx LIN 超时定时器清零寄存器（UARTx_LINTOCR） 326
19.12.8 UARTx LIN Break 长度寄存器（UARTx_LINBKR） 327
19.12.9 UARTx 控制状态寄存器（UARTx_CSR） 327
19.12.10 UARTx 中断使能寄存器（UARTx_IER） 329
19.12.11 UARTx 中断标志寄存器（UARTx_ISR） 330
19.12.12 UARTx 超时和延迟寄存器（UARTx_TODR） 331
19.12.13 UARTx 接收缓冲寄存器（UARTx_RXBUF） 332
19.12.14 UARTx 发送缓冲寄存器（UARTx_TXBUF） 332
19.12.15 UATRx 波特率产生寄存器（UARTx_BGR） 333
19.12.16 UARTx 智能卡状态标志寄存器（UARTx_SCISR） 333
19.12.17 UARTx 智能卡中断使能寄存器（UARTx_SCIER） 335
19.12.18 UARTx 智能卡帧格式控制寄存器（UARTx_FFCR） 335
19.12.19 UARTx 智能卡额外保护时间寄存器（UARTx_EGTR） 337
19.12.20 UARTx 智能卡输出时钟分频寄存器（UARTx_CODR） 337

20 低功耗 UART（LPUART） 338

20.1 概述 338
20.2 结构框图 339
20.3 工作时钟 340
20.4 字符描述 340
20.5 功能描述 341
20.5.1 位接收采样和发送 341
20.5.2 接收流程 342
20.5.3 发送流程 342

20.5.4 使用 DMA 进行 LPUART 收发 *342*
20.5.5 休眠模式下的数据接收唤醒 *342*
20.5.6 DMA 模式下的发送完成中断 *343*
20.6 寄存器 344
20.6.1 LPUARTx 控制状态寄存器（LPUARTx_CSR） *344*
20.6.2 LPUARTx 中断使能寄存器（LPUARTx _IER） *346*
20.6.3 LPUARTx 中断标志寄存器（LPUARTx _ISR） *347*
20.6.4 LPUARTx 波特率调制寄存器（LPUARTx _BMR） *347*
20.6.5 LPUARTx 接收数据寄存器（LPUARTx _RXBUF） *348*
20.6.6 LPUARTx 发送数据寄存器（LPUARTx _TXBUF） *349*
20.6.7 LPUARTx 数据匹配寄存器（LPUARTx _DMR） *349*

21 串行外设接口（SPI/I2S） **350**

21.1 概述 350
21.2 SPI 功能描述 351
21.2.1 SPI 结构框图 *351*
21.2.2 引脚定义 *351*
21.2.3 SPI 接口时序 *352*
21.2.4 I/O 配置 *354*
21.2.5 全双工数据通信 *355*
21.2.6 主机发送间隔 *357*
21.2.7 可编程数据字长 *357*
21.2.8 TX-ONLY 模式 *357*
21.2.9 RX-ONLY 模式 *358*
21.2.10 主机 SSN 控制 *358*
21.2.11 数据冲突 *359*
21.2.12 使用 DMA 进行 SPI 收发 *359*
21.3 I2S 功能描述 361
21.3.1 I2S 结构框图 *361*
21.3.2 引脚定义 *361*
21.3.3 协议支持 *361*
21.3.4 时钟产生 *365*
21.3.5 I2S 主机功能 *366*
21.3.6 I2S 从机功能 *367*
21.3.7 帧格式错误 *368*
21.3.8 DMA 支持 *369*
21.4 寄存器 370
21.4.1 SPI 控制寄存器 1（SPIx_CR1） *370*
21.4.2 SPI 控制寄存器 2（SPIx_CR2） *371*
21.4.3 SPI 控制寄存器 3（SPIx_CR3） *373*
21.4.4 SPI/I2S 中断控制寄存器（SPIx_IER） *374*
21.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR） *374*
21.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF） *376*
21.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF） *376*
21.4.8 I2S 配置寄存器（SPIx_I2SCR） *377*
21.4.9 I2S 预分频寄存器（SPIx_I2SPR） *378*

22 单边沿半字节传输模块（SENT） **379**

22.1 概述 379
22.2 消息格式 379
22.2.1 Data Nibble *379*
22.2.2 同步脉冲（Sync Pulse） *380*
22.2.3 Status and communication nibble *380*
22.2.4 消息帧格式 *382*

22.2.5 Optional Pause Pulse *382*
22.2.6 CRC 校验 *382*
22.3 结构框图 383
22.4 功能描述 383
22.4.1 发送 Tick 时间 *383*
22.4.2 发送模式 *384*
22.4.3 接收 Tick 时间 *384*
22.4.4 接收模式 *386*
22.4.5 接收消息诊断 *386*
22.5 寄存器 387
22.5.1 SENT 控制寄存器（SENT_CR） *387*
22.5.2 SENT 配置寄存器（SENT_CFGR） *388*
22.5.3 SENT 预分频寄存器（SENT_PSR） *389*
22.5.4 SENT 帧长度寄存器（SENT_FTR） *389*
22.5.5 SENT 状态和标志寄存器（SENT_SR） *390*
22.5.6 SENT 数据寄存器（SENT_DR） *391*
22.5.7 SENT 同步脉冲诊断寄存器（SENT_SYNC） *392*
22.5.8 SENT 同步校准寄存器（SENT_RCAL） *392*

23 经典控制器区域网络（FSCAN） **394**

23.1 概述 394
23.2 结构框图 395
23.3 功能描述 395
23.3.1 时钟和复位 *395*
23.3.2 Bit timing *396*
23.3.3 Bit stream processor *396*
23.3.4 控制器工作模式 *397*
23.3.5 寄存器写权限与控制器模式 *398*
23.3.6 消息帧存储和消息帧结构（Message storage and structure） *398*
23.3.7 控制器初始化 *400*
23.3.8 消息发送 *400*
23.3.9 消息接收 *402*
23.3.10 消息滤波器（Acceptance Filter） *402*
23.3.11 错误管理 *405*
23.3.12 异步唤醒 *407*
23.4 寄存器 409
23.4.1 CAN 控制寄存器（CAN_CR） *410*
23.4.2 CAN 模式选择寄存器（CAN_MSR） *411*
23.4.3 CAN 波特率预分频寄存器（CAN_BRPR） *412*
23.4.4 CAN 位时序寄存器（CAN_BTR） *412*
23.4.5 CAN 错误计数寄存器（CAN_ECR） *413*
23.4.6 CAN 错误状态寄存器（CAN_ESR） *414*
23.4.7 CAN 状态标志寄存器（CAN_SR） *415*
23.4.8 CAN 接收中断标志寄存器（CAN_RXISR） *416*
23.4.9 CAN 接收中断使能寄存器（CAN_RXIER） *417*
23.4.10 CAN 发送缓冲控制寄存器（CAN_TXBCR） *418*
23.4.11 CAN 发送中断标志寄存器（CAN_TXBSR） *419*
23.4.12 CAN 发送缓冲 ID 寄存器（CAN_TXBxIDR, x=0,1,2） *420*
23.4.13 CAN 发送缓冲 DLC 寄存器（CAN_TXBxDLCR, x=0,1,2） *420*
23.4.14 CAN 发送缓冲 DataWord1 寄存器（CAN_TXBxDW1R, x=0,1,2） *421*
23.4.15 CAN 发送缓冲 DataWord2 寄存器（CAN_TXBxDW2R, x=0,1,2） *421*
23.4.16 CAN 接收滤波控制寄存器（CAN_AFCR） *422*
23.4.17 CAN 接收滤波配置寄存器（CAN_AFCFGR） *422*
23.4.18 CAN 接收滤波掩码寄存器 x（CAN_AFMRx） *423*

23.4.19 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx） *424*
23.4.20 CAN 接收 FIFO ID 寄存器（CAN_RXF_IDR） *424*
23.4.21 CAN 接收 FIFO DLC 寄存器（CAN_RXF_DLCR） *425*
23.4.22 CAN 接收 FIFO DataWord1 寄存器（CAN_RXF_DW1R） *425*
23.4.23 CAN 接收 FIFO DataWord2 寄存器（CAN_RXF_DW2R） *426*

24 直接存储访问控制器（DMA） **427**

24.1 概述 427
24.2 结构框图 428
24.3 功能描述 428
24.3.1 外设 DMA 传输 *428*
24.3.2 外设请求的处理 *429*
24.3.3 通道优先级仲裁 *429*
24.3.4 传输方向定义 *430*
24.3.5 地址指针 *430*
24.3.6 访问带宽 *431*
24.3.7 传输长度 *431*
24.3.8 循环模式 *431*
24.3.9 双缓冲区模式 *433*
24.4 通道控制 434
24.4.1 DMA 请求映射 *434*
24.4.2 Request Select *436*
24.5 寄存器 437
24.5.1 DMA 全局控制寄存器（DMA_GCR） *438*
24.5.2 DMA 软件请求寄存器寄存器（DMA_SWRR） *438*
24.5.3 DMA 中断状态标志寄存器（DMA_ISR） *439*
24.5.4 通道 x 控制寄存器（DMA_CHxCR） *440*
24.5.5 通道 x 配置寄存器（DMA_CHxCFGR） *440*
24.5.6 通道 x 外设地址寄存器（DMA_CHxPAR） *441*
24.5.7 通道 x 存储器地址寄存器 0（DMA_CHxMAR0） *442*
24.5.8 通道 x 存储器地址寄存器 1（DMA_CHxMAR1） *443*
24.5.9 通道 7 控制寄存器（DMA_CH7CR） *443*
24.5.10 通道 7 Flash 指针寄存器（DMA_CH7FAR） *444*
24.5.11 通道 7 RAM 指针寄存器（DMA_CH7RAR） *445*

25 循环冗余校验（CRC） **446**

25.1 概述 446
25.2 软件配置过程 447
25.3 GOLDEN 数据 448
25.4 DMA 接口 448
25.5 FLASH 数据完整性校验 449
25.6 寄存器 450
25.6.1 CRC 数据寄存器（CRC_DR） *450*
25.6.2 CRC 控制状态寄存器（CRC_CR） *450*
25.6.3 CRC LFSR 寄存器（CRC_LFSR） *452*
25.6.4 CRC 输出异或寄存器（CRC_XOR） *452*
25.6.5 CRC 多项式寄存器（CRC_POLY） *453*

26 高级定时器（ATIM） **454**

26.1 概述 454
26.2 主要特性 454
26.3 结构框图 455
26.4 工作时钟 455
26.5 管脚和内部信号连接 456

26.6 功能描述 460
26.6.1 定时单元 460
26.6.2 定时器工作模式 462
26.6.3 重复计数器 469
26.6.4 Preload 寄存器 470
26.6.5 计数器工作时钟 471
26.6.6 内部触发信号（ITRx） 476
26.6.7 捕捉/比较通道 477
26.6.8 输入捕捉模式 479
26.6.9 软件 Force 输出 481
26.6.10 输出比较模式 482
26.6.11 PWM 输出 483
26.6.12 非对称 PWM 模式 486
26.6.13 组合 PWM 模式 488
26.6.14 组合三相 PWM 模式 489
26.6.15 互补输出和死区插入 490
26.6.16 刹车功能 491
26.6.17 互补输出通道信号状态逻辑表 497
26.6.18 6-step PWM 输出 498
26.6.19 单脉冲输出 499
26.6.20 可重触发的单脉冲输出（retriggerable one-pulse mode） 500
26.6.21 外部事件清除 OCxREF 502
26.6.22 编码器接口模式（encoder interface mode） 502
26.6.23 定时器同步 517
26.6.24 触发信号输出 520
26.6.25 DMA 访问 522
26.6.26 DMA Burst 523
26.6.27 输入异或功能 523
26.6.28 霍尔传感器接口 524
26.6.29 Debug 模式 524
26.7 寄存器 526
26.7.1 ATIM 控制寄存器 1 526
26.7.2 ATIM 控制寄存器 2 528
26.7.3 ATIM 从机模式控制寄存器 530
26.7.4 ATIM DMA 和中断使能寄存器（ATIM_IER） 532
26.7.5 ATIM 状态寄存器（ATIM_ISR） 534
26.7.6 ATIM 事件产生寄存器（ATIM_EGR） 535
26.7.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1） 536
26.7.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2） 539
26.7.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER） 541
26.7.10 ATIM 计数器寄存器（ATIM_CNT） 542
26.7.11 ATIM 预分频寄存器（ATIM_PSC） 543
26.7.12 ATIM 自动重载（auto-reload）寄存器（ATIM_ARR） 543
26.7.13 ATIM 重复计数寄存器（ATIM_RCR） 544
26.7.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1） 544
26.7.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2） 545
26.7.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3） 545
26.7.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4） 546
26.7.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR） 546
26.7.19 ATIM DMA 控制寄存器（ATIM_DCR） 548
26.7.20 ATIM DMA 访问寄存器（ATIM_DMAR） 549
26.7.21 ATIM 捕捉/比较模式寄存器 3（ATIM_CCMR3） 550
26.7.22 ATIM 比较寄存器 5（ATIM_CCR5） 551

26.7.23 ATIM 比较寄存器 6（ATIM_CCR6）.....551
26.7.24 ATIM 编码器模式控制寄存器（ATIM_ECR）.....552
26.7.25 ATIM 输入选择寄存器（ATIM_TISEL）.....553
26.7.26 ATIM 辅助功能寄存器（ATIM_AFR）.....554
26.7.27 ATIM 刹车标志寄存器（ATIM_BRKFLAG）.....555
26.7.28 DMA Burst 控制寄存器（ATIM_DCR）.....556
26.7.29 DMA burst 访问寄存器（ATIM_DMAR）.....556

27 通用定时器（GPTIM）.....558

27.1 概述.....558
27.2 主要特性.....558
27.3 结构框图.....559
27.4 引脚和内部连接.....559
27.5 功能描述.....562
27.5.1 定时单元.....562
27.5.2 定时器工作模式.....565
27.5.3 计数器工作时钟.....572
27.5.4 内部触发信号（ITRx）的捕捉.....578
27.5.5 捕捉/比较通道.....578
27.5.6 输入捕捉模式.....579
27.5.7 软件 Force 输出.....581
27.5.8 输出比较模式.....581
27.5.9 PWM 输出.....582
27.5.10 非对称 PWM 模式.....583
27.5.11 组合 PWM 模式.....585
27.5.12 单脉冲输出.....586
27.5.13 外部事件清除 OCxREF.....588
27.5.14 编码器接口模式（encoder interface）.....588
27.5.15 GPTIM 从机模式.....603
27.5.16 DMA 访问.....606
27.5.17 DMA Burst.....606
27.5.18 输入异或功能.....607
27.5.19 Debug 模式.....607
27.6 寄存器.....607
27.6.1 GPTIMx 控制寄存器 1（GPTIMx_CR1）.....609
27.6.2 GPTIMx 控制寄存器 2（GPTIMx_CR2）.....610
27.6.3 GPTIMx 从机模式控制寄存器（GPTIMx_SMCR）.....611
27.6.4 GPTIMx DMA 和中断使能寄存器（GPTIMx_DIER）.....613
27.6.5 GPTIMx 状态寄存器（GPTIMx_ISR）.....615
27.6.6 GPTIMx 事件产生寄存器（GPTIMx_EGR）.....617
27.6.7 GPTIMx 捕捉/比较模式寄存器 1（GPTIMx_CCMR1）.....617
27.6.8 GPTIMx 捕捉/比较模式寄存器 2（GPTIMx_CCMR2）.....620
27.6.9 GPTIMx 捕捉/比较使能寄存器（GPTIMx_CCER）.....622
27.6.10 GPTIMx 计数器寄存器（GPTIMx_CNT）.....623
27.6.11 GPTIMx 预分频寄存器（GPTIMx_PSC）.....624
27.6.12 GPTIMx 自动重载寄存器（GPTIMx_ARR）.....624
27.6.13 GPTIMx 捕捉/比较寄存器 1（GPTIMx_CCR1）.....625
27.6.14 GPTIMx 捕捉/比较寄存器 2（GPTIMx_CCR2）.....625
27.6.15 GPTIMx 捕捉/比较寄存器 3（GPTIMx_CCR3）.....626
27.6.16 GPTIMx 捕捉/比较寄存器 4（GPTIMx_CCR4）.....626
27.6.17 GPTIM 编码器模式控制寄存器（GPTIM_ECR）.....627
27.6.18 GPTIM 输入选择寄存器（GPTIM_TISEL）.....628
27.6.19 GPTIM 辅助功能寄存器（GPTIM_AFR）.....629

27.6.1 GPTIMx DMA 控制寄存器（GPTIMx_DCR） *629*
27.6.2 GPTIMx DMA 访问寄存器（GPTIMx_DMAR） *631*

28 定时器阵列单元（TAU） **632**

28.1 概述 632
28.2 结构框图 632
28.3 功能描述 634
28.3.1 时钟和复位 *634*
28.3.2 工作模式 *634*
28.3.3 输入输出通道 *634*
28.3.4 16 位计数模式 *635*
28.3.5 触发信号输出（TRGO） *637*
28.3.6 32 位级联模式 *637*
28.3.7 捕捉功能 *642*
28.3.8 预分频 *646*
28.3.9 16 位 PWM 输出 *646*
28.3.10 级联 PWM 输出 *647*
28.3.11 外部引脚输入数字滤波 *651*
28.4 寄存器 652
28.4.1 TAU0 控制寄存器（TAU0_CR） *653*
28.4.2 TAU0 通道 x 配置寄存器（TAU0_TxCFGR） *654*
28.4.3 TAU0 通道 x 模式寄存器（TAU0_TxMDR） *655*
28.4.4 TAU0 重载寄存器（TAU0_TxARR） *655*
28.4.5 TAU0 捕捉比较寄存器（TAU0_TxCCR） *656*
28.4.6 TAU0 中断使能寄存器（TAU0_TxIER） *656*
28.4.7 TAU0 中断标志寄存器（TAU0_TxISR） *657*
28.4.8 TAU0 计数值寄存器（TAU0_TxCNTR） *658*

29 16 位基本定时器（BSTIM16） **659**

29.1 概述 659
29.2 主要特性 659
29.3 结构框图 659
29.4 功能描述 660
29.4.1 定时单元 *660*
29.4.2 定时器工作模式 *662*
29.4.3 计数器工作时钟 *664*
29.4.1 Debug 模式 *665*
29.5 寄存器 666
29.5.1 BSTIM16 控制寄存器 1(BSTIM16_CR1) *666*
29.5.2 BSTIM16 控制寄存器 2(BSTIM16_CR2) *667*
29.5.3 BSTIM16 中断使能寄存器(BSTIM16_DIER) *668*
29.5.4 BSTIM16 状态寄存器(BSTIM16_SR) *668*
29.5.5 BSTIM16 事件产生寄存器(BSTIM16_EGR) *669*
29.5.6 BSTIM16 计数器寄存器(BSTIM16_CNT) *669*
29.5.7 BSTIM16 预分频寄存器(BSTIM16_PSC) *670*
29.5.8 BSTIM16 自动重载寄存器(BSTIM16_ARR) *670*

30 16 位低功耗定时器（LPTIM16） **672**

30.1 概述 672
30.2 结构框图 673
30.3 时钟和复位 673
30.4 定时器功能 673
30.4.1 普通定时器 *674*

30.4.2 外部脉冲触发计数 *674*
30.4.3 外部异步脉冲计数 *674*
30.4.4 Timeout 模式 *675*
30.5 捕捉比较功能 676
30.5.1 16bit PWM *676*
30.5.2 输入捕捉 *677*
30.5.1 输入数字滤波 *678*
30.6 正交编码器 679
30.7 寄存器 681
30.7.1 LPTIM16 配置寄存器（LPTIM16_CFGR） *681*
30.7.2 LPTIM16 计数值寄存器（LPTIM16_CNT） *683*
30.7.3 LPTIM16 捕捉比较控制和状态寄存器（LPTIM16_CCSR） *683*
30.7.4 LPTIM16 目标值寄存器（LPTIM16_ARR） *685*
30.7.5 LPTIM16 中断使能寄存器（LPTIM16_IER） *685*
30.7.6 LPTIM16 中断标志寄存器（LPTIM16_ISR） *686*
30.7.7 LPTIM16 控制寄存器（LPTIM16_CR） *687*
30.7.8 LPTIM16 捕捉比较寄存器 1（LPTIM16_CCR1） *688*
30.7.9 LPTIM16 捕捉比较寄存器 2（LPTIM16_CCR2） *688*

31 实时时钟（RTC） **690**

31.1 概述 690
31.2 结构框图 690
31.3 工作原理 691
31.3.1 时基计数器（LTBC） *691*
31.3.2 LTBC 数字调校 *692*
31.3.3 BCD 时间 *693*
31.3.4 RTC 使能与停止 *694*
31.3.5 RTC 时间设置 *694*
31.3.6 RTC 时间读取 *695*
31.3.7 闰年判断 *695*
31.4 寄存器 696
31.4.1 RTC 写使能寄存器（RTC_WER） *696*
31.4.2 RTC 中断使能寄存器（RTC_IER） *697*
31.4.3 RTC 中断标志寄存器（RTC_ISR） *698*
31.4.4 BCD 时间秒寄存器（RTC_BCDSEC） *699*
31.4.5 BCD 时间分钟寄存器（RTC_BCDMIN） *700*
31.4.6 BCD 时间小时寄存器（RTC_BCDHOUR） *700*
31.4.7 BCD 时间天寄存器（RTC_BCDDAY） *701*
31.4.8 BCD 时间星期寄存器（RTC_BCDWEEK） *701*
31.4.9 BCD 时间月寄存器（RTC_BCDMONTH） *702*
31.4.10 BCD 时间年寄存器（RTC_BCDYEAR） *702*
31.4.11 闹钟寄存器（RTC_ALARM） *703*
31.4.12 RTC 时间信号输出寄存器（RTC_TMSEL） *703*
31.4.13 LTBC 数值调整寄存器（RTC_ADJUST） *704*
31.4.14 毫秒计数值寄存器（RTC_SBSCNT） *705*
31.4.15 RTC 控制寄存器（RTC_CR） *705*

32 段码液晶驱动（LCD） **706**

32.1 概述 706
32.2 结构框图 706
32.3 IO 配置 707
32.4 功能说明 707
32.4.1 工作时钟和显示帧频率 *707*

32.4.2 LCD Type A 扫描波形 *708*
32.4.3 LCD Type B 扫描波形 *709*
32.4.4 片内 buffer 驱动模式 *710*
32.4.5 显示闪烁功能 *711*
32.4.6 偏置电压调整 *711*
32.5 寄存器 713
32.5.1 显示控制寄存器（LCD_CR） *713*
32.5.2 显示测试控制寄存器（LCD_TEST） *716*
32.5.3 测试模式下引脚输出数据寄存器 *716*
32.5.4 显示频率控制寄存器（LCD_FCR） *717*
32.5.5 闪烁时间寄存器（LCD_FLKT） *717*
32.5.6 显示中断使能寄存器（LCD_IER） *718*
32.5.7 显示中断标志寄存器（LCD_ISR） *718*
32.5.8 显示数据寄存器（LCD_DATAx） *719*
32.5.9 LCD COM 使能控制寄存器（LCD_COMEN） *724*
32.5.10 LCD SEG 使能控制寄存器 0（LCD_SEGEN0） *725*

33 LED 驱动（LED） **727**

33.1 概述 727
33.2 IO 配置 727
33.3 功能说明 727
33.3.1 扫描波形 *727*
33.3.2 工作时钟 *728*
33.3.3 PWM 调光 *728*
33.3.4 扫描时序 *729*
33.3.5 点数据 *730*
33.3.6 输出电平 *730*
33.3.7 中断 *730*
33.3.8 DMA 支持 *730*
33.3.9 按键扫描 *731*
33.4 寄存器 732
33.4.1 LED 控制寄存器 *733*
33.4.2 LED 配置寄存器 *733*
33.4.3 LED 中断使能寄存器 *735*
33.4.4 LED 中断标志寄存器 *735*
33.4.5 PWM 比较值寄存器 A *736*
33.4.6 PWM 比较值寄存器 B *736*
33.4.7 PWM 比较值寄存器 C *737*
33.4.8 PWM 比较值寄存器 D *737*
33.4.9 LED 点数据寄存器 x (x=0~9) *738*
33.4.10 DMA 控制寄存器 *738*
33.4.11 PWMCRx 的 DMA 访问寄存器 *739*
33.4.12 DOTxDR 的 DMA 访问寄存器 *739*
33.4.13 LED 扫描当前 COM 线 *740*

34 模数转换器（ADC） **741**

34.1 概述 741
34.2 结构框图 742
34.3 输入通道 742
34.4 时钟与复位 743
34.4.1 时钟 *743*
34.4.2 复位 *743*
34.5 工作时序 743

34.6 功能描述 745
34.6.1 ADC 转换队列（ACQ） 745
34.6.2 注入 ACQ 转换 746
34.6.3 扫描 ACQ 转换 748
34.6.4 触发源 751
34.6.5 使用 VDDA 作为基准 752
34.6.6 温度传感器 753
34.6.7 可编程采样时间 753
34.6.8 过采样和硬件平均 755
34.6.9 数据冲突和自动等待 755
34.6.10 DMA 755
34.6.11 模拟窗口看门狗（AWD） 756
34.6.12 ADC 校准 756
34.7 低功耗模式 757
34.8 寄存器 757
34.8.1 ACQ 中断和状态寄存器（ACQ_ISR） 758
34.8.2 ACQ 中断使能寄存器（ACQ_IER） 759
34.8.3 通用中断和状态寄存器（ADC_GISR） 760
34.8.4 通用中断使能寄存器（ADC_GIER） 760
34.8.5 控制寄存器 1（ADC_CR1） 761
34.8.6 控制寄存器 2（ADC_CR2） 762
34.8.7 配置寄存器 1（ADC_CFGR1） 762
34.8.8 配置寄存器 2（ADC_CFGR2） 763
34.8.9 校准数据寄存器（ADC_OSR） 765
34.8.10 ACQ 标志寄存器（ACQ_SR） 765
34.8.11 Scan ACQ 控制寄存器（ACQ_SACR） 766
34.8.12 AWD 阈值配置寄存器（ADC_HLTR） 766
34.8.13 AWD 阈值配置寄存器（ADC_AWDCR） 767
34.8.14 ACQ0 控制寄存器（ACQ0_CR） 767
34.8.15 ACQ1 控制寄存器（ACQ1_CR） 768
34.8.16 ACQ2 控制寄存器（ACQ2_CR） 769
34.8.17 ACQ3 控制寄存器（ACQ3_CR） 770
34.8.18 SCAN_ACQ 控制寄存器（SCAN_ACQ_CR） 771
34.8.19 ACQ0 数据寄存器（ACQ0_DR） 771
34.8.20 ACQ1 数据寄存器（ACQ1_DR） 772
34.8.21 ACQ2 数据寄存器（ACQ2_DR） 772
34.8.22 ACQ3 数据寄存器（ACQ3_DR） 773
34.8.23 SCAN_ACQ 数据寄存器（SCAN_ACQ_DR） 773

35 运算放大器（OPA） 775

35.1 概述 775
35.2 结构框图 775
35.3 引脚定义 776
35.4 功能描述 776
35.4.1 时钟和复位 776
35.4.2 Standalone 模式（non-inverting amplifier） 776
35.4.3 Standalone 模式（inverting amplifier） 777
35.4.4 Buffer 模式 777
35.4.5 Offset 校准 778
35.5 寄存器 780
35.5.1 OPAx 控制寄存器（OPAx_CR） 780
35.5.2 OPAx 校准寄存器（OPAx_CALR） 781
35.5.3 OPAx 校准输出寄存器（OPAx_COR） 781

36 数模转换器（DAC） 783

36.1 概述 783
36.2 结构框图 783
36.3 引脚定义 783
36.4 功能描述 784
36.4.1 工作时钟与信号时序 784
36.4.2 DAC 输出模式 784
36.4.3 DAC 触发源选择 786
36.4.4 DAC 输出电压 787
36.4.5 DAC 输出缓冲 787
36.4.6 DMA 787
36.5 寄存器 788
36.5.1 DAC 控制寄存器 1（DAC_CR1） 788
36.5.2 DAC 控制寄存器 2（DAC_CR2） 789
36.5.3 DAC 配置寄存器（DAC_CFGR） 789
36.5.4 DAC 软件触发寄存器（DAC_SWTRGR） 790
36.5.5 DAC 数据保持寄存器（DAC_DHR） 790
36.5.6 DAC 状态标志寄存器（DAC_ISR） 791
36.5.7 DAC 中断使能寄存器（DAC_IER） 791

37 可编程胶合逻辑（PGL） 793

37.1 概述 793
37.2 结构框图 793
37.3 引脚定义 795
37.4 功能描述 795
37.4.1 LUT 真值表 795
37.4.2 LUT 输入 796
37.4.3 LUT 输出 796
37.4.4 滤波和采样 797
37.4.5 中断和触发 798
37.4.6 低功耗模式 798
37.5 寄存器 799
37.5.1 PGL 控制寄存器（PGL_CR） 799
37.5.2 PGL 配置寄存器 0（PGL_CFGR0） 799
37.5.3 PGL 配置寄存器 1（PGL_CFGR1） 801
37.5.4 PGL 配置寄存器 2（PGL_CFGR2） 802
37.5.5 PGL 配置寄存器 3（PGL_CFGR3） 803
37.5.6 PGL 中断使能寄存器（PGL_IER） 804
37.5.7 PGL 中断标志寄存器（PGL_ISR） 804
37.5.8 LUT0 真值表寄存器（PGL_LUT0） 805
37.5.9 LUT1 真值表寄存器（PGL_LUT1） 805
37.5.10 LUT2 真值表寄存器（PGL_LUT2） 806
37.5.11 LUT3 真值表寄存器（PGL_LUT3） 806

38 I/O 端口（GPIO） 808

38.1 概述 808
38.2 引脚类型 808
38.2.1 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 809
38.2.2 PC14(NRST)，仅上拉 810
38.3 IO 端口功能定义 810
38.3.1 GPIO 输入 811
38.3.2 GPIO 输出 811
38.3.3 数字外设功能 811

38.3.4 模拟功能 *812*
38.3.5 使用外部晶体引脚 *813*
38.4 NRST 引脚与 GPIO 复用 813
38.5 SWD 引脚 813
38.6 WKUPx 引脚 813
38.7 外部引脚中断（EXTI） 814
38.7.1 功能说明 *814*
38.7.2 应用指南 *816*
38.8 快速 GPIO 输出 817
38.9 寄存器 818
38.9.1 GPIO 输入使能寄存器（GPIOx_INEN） *820*
38.9.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN） *821*
38.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN） *821*
38.9.4 GPIO 功能选择寄存器（GPIOx_FCR） *822*
38.9.5 GPIO 输出数据寄存器（GPIOx_DO） *824*
38.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET） *825*
38.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST） *825*
38.9.8 GPIO 输入数据寄存器（GPIOx_DIN） *826*
38.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS） *826*
38.9.10 GPIO 驱动强度配置寄存器（GPIOx_DSR） *827*
38.9.11 EXTI 输入选择寄存器 0（GPIO_EXTISEL0） *827*
38.9.12 EXTI 边沿选择和使能寄存器 0（GPIO_EXTIEDS0） *829*
38.9.13 EXTI 数字滤波控制寄存器（GPIO_EXTIDF） *830*
38.9.14 EXTI 中断标志（GPIO_EXTIISR） *831*
38.9.15 EXTI 输入信号寄存器（GPIO_EXTIDI） *831*
38.9.16 FOUT 配置寄存器（GPIO_FOUTSEL） *832*
38.9.17 WKUP 控制寄存器（GPIO_PINWKEN） *833*

39 安全功能（SFU） **834**

39.1 概述 834
39.2 闪存 CRC 检查 834
39.3 RAM 自检 834
39.4 RAM 奇偶校验 835
39.5 RAM 写保护 835
39.6 关键寄存器写保护 835
39.7 非法地址访问检测 837
39.8 时钟频率检测 839
39.9 晶振停振检测 839
39.10 电源电压检测 839
39.11 IO 输出状态回读 840
39.12 LIN 自测试 840
39.13 CAN 自测试 840
39.14 ADC 测试 840
39.15 DAC 自测试 841
39.16 温度检测 841
39.17 寄存器 842
39.17.1 RAM 奇偶校验控制寄存器（SFU_RAMP_CR） *842*
39.17.2 RAM 奇偶校验状态寄存器（SFU_RAMP_SR） *842*
39.17.3 存储访问保护控制寄存器（SFU_MAP_CR） *843*
39.17.4 存储访问保护状态寄存器（SFU_MAP_SR） *844*

40 专用编程接口 **846**

40.1 概述 846

40.2 编程器使用 846

41 调试支持 847

41.1 概述 847
41.2 DEBUG 引脚 848
41.2.1 SWD 引脚 848
41.2.2 上拉电阻 848
41.3 SWD 接口协议 849
41.3.1 协议简介 849
41.3.2 传输序列 849
41.3.3 SW-DP ID code 850
41.3.4 主机读操作 850
41.3.5 主机写操作 851
41.4 SWD-DP 寄存器 852
41.4.1 寄存器列表 852
41.5 CORE DEBUG 寄存器 852
41.6 低功耗调试支持 852
41.7 DEBUG 相关的配置项 853
41.8 寄存器 853
41.8.1 MCU DEBUG 配置寄存器 853

42 器件签名信息 855

42.1 唯一器件 ID 855
42.2 存储器容量查询 855

版本列表 856

上海复旦微电子集团股份有限公司销售及服务网点 857

表目录

表 1-1 型号列表.....37
表 2-1 引脚列表.....47
表 3-1 FM33LF0A 极限参数.....58
表 3-2 FM33LF0A 典型工作条件.....58
表 3-3 ACTIVE 电流参数.....59
表 3-4ADC 参数.....66
表 3-5ADC 输入阻抗.....68
表 3-6 DAC 参数.....68
表 3-7 温度传感器参数.....69
表 3-8 OPA 参数.....70
表 4-1 外设模块总线地址列表.....77
表 7-1 FM33LF0xxA CPU 配置简表.....106
表 7-2 CORTEX-M0 内核寄存器简表.....107
表 7-3 FM33LF0xxA 中断向量表.....109
表 9-1 IWDT 溢出周期表.....122
表 10-1 WWDT 溢出周期表.....130
表 16-1 比较器 1 引脚列表.....210
表 16-2 比较器 2 引脚列表.....211
表 16-3 比较器 3 引脚列表.....211
表 16-4 比较器 4 引脚列表.....211
表 16-5 比较器工作模式.....213
表 18-1 I²C 引脚列表.....235
表 18-2 I²C 接口时序要求.....238
表 18-3 I²C-SMBUS 从机保留地址定义.....240
表 18-4SMBUS 超时事件.....267
表 19-1 UART 引脚列表.....287
表 19-2 UART 类型列表.....287
表 19-3 UART 数据帧格式.....288
表 19-4 DMA 发送中断.....299
表 19-5 常用时钟频率下波特率计算.....300
表 20-1 LPUART 数据帧格式.....340
表 20-2 LPUART 数据位调制系数.....341
表 20-3 LPUART DMA 中断说明.....343
表 21-1SPI 引脚列表.....352
表 21-2I2S 典型采样频率计算表.....366
表 24-1 PDMA 通道映射列表.....436
表 26-1 外部引脚连接.....456
表 26-2 内部信号连接.....457
表 26-3ITR 信号连接.....457
表 26-4ETR 信号连接.....458
表 26-5 TI1_IN 信号连接.....458
表 26-6 TI2_IN 信号连接.....459
表 26-7 TI3_IN 信号连接.....459
表 26-8 TI4_IN 信号连接.....459
表 26-9OCREF_CLR 信号连接.....460
表 26-10 刹车行为.....496
表 26-11 ENCODER INTERFACE 计数方式（CC1P=CC2P=0）.....503
表 26-12 DIRECTIONAL CLOCK MODE 计数方式.....506
表 27-1GPTIM 输入输出引脚.....559

表 27-2GPTIM 内部输入输出信号 560
表 27-3GPTIM 内部触发信号连接 560
表 27-4GPTIM 外部触发信号连接 561
表 27-5GPTIM 通道 1 信号连接 561
表 27-6GPTIM 通道 2 信号连接 562
表 27-7GPTIM 通道 3 信号连接 562
表 27-8GPTIM 通道 4 信号连接 562
表 27-9 ENCODER INTERFACE 计数方式（CC1P=CC2P=0） 589
表 27-10 DIRECTIONAL CLOCK MODE 计数方式 592
表 28-1 定时器级联触发选择 642
表 28-2 定时器捕捉源选择 646
表 30-2 ENCODER INTERFACE 计数方式 679
表 32-1 帧频率计算公式 707
表 32-2 典型帧频率和 DF 的关系 708
表 34-1 ADC 输入通道 743
表 34-2 ACQ 触发源 752
表 34-3 ADC 采样时间 754
表 35-1 OPA 引脚列表 776
表 36-1 DAC 相关引脚 784
表 36-2 DAC 触发源 786
表 37-1 PGL 相关引脚 795
表 37-2 LUT 输出连接 796
表 38-1GPIO 功能逻辑定义表 809
表 38-2 FCR 定义表 811
表 38-3PAD 通道配置 812
表 39-164K FLASH 版本总线访问控制表 838
表 39-2 32K FLASH 版本总线访问控制表 839

图目录

图 1-1 芯片结构框图 36
图 2-1 FM33LF0x5A LQFP48 封装图 38
图 2-2 FM33LF0x3A QFN32 封装图 39
图 2-3 LQFP48 封装尺寸图 52
图 2-4 JEDEC 标准的耐热回流温度曲线 55
图 3-1 FM33LF0A 测试方案 57
图 3-2 ADC 参数说明 65
图 3-3 ADC 通道输入阻抗 67
图 3-4 温度传感器输出曲线 69
图 4-1 系统总线示意图 74
图 4-2 FM33LF01xA 总线地址 75
图 5-1 芯片电源结构图 94
图 5-2 功耗模式与系统主频 96
图 8-1 芯片复位源框图 111
图 9-1 IWDT 结构框图 121
图 9-2 IWDT 窗口示意图 123
图 10-1 WWDT 结构框图 128
图 10-2 WWDT 窗口示意图 130
图 11-1 芯片时钟框图 136
图 12-1 CLM 电路框图 160
图 12-2 参考时钟计数和监控时钟计数 162
图 12-3 监控时钟频率下溢出 162
图 12-4 监控时钟频率上溢出 163
图 12-5 超时检测电路 164
图 12-6 TIME-OUT 复位 164
图 13-1 低压检测电路框图 169
图 13-2 低压检测电路工作时序 170
图 13-3 电源检测电路间歇工作模式 171
图 14-1 ECB 模式加密流程 179
图 14-2 ECB 模式解密流程 180
图 14-3 CBC 加密过程 181
图 14-4 CBC 解密过程 182
图 14-5 暂停模式流程 182
图 14-6 CTR 加密流程 183
图 14-7 CTR 解密流程 184
图 14-8 32 位计数器和随机数的存储方式 184
图 14-9 GCM 加密流程 186
图 14-10 GCM 解密流程 187
图 14-11 MULTH 模块框图 188
图 14-12 根据数据类型存储数据的示意图 190
图 14-13 模式 1：加密流程 191
图 14-14 模式 2 示意图 191
图 14-15 模式 3 示意图 192
图 14-16 模式 4 示意图 193
图 14-17 MULTH 模块使用流程示意图 194
图 14-18 输入时 DMA 请求和数据传输示意图 194
图 14-19 输出时 DMA 请求和数据传输示意图 195
图 15-1 真随机数模块框图 202
图 15-2 真随机数模块工作时钟 203

图 16-1 比较器电路框图 209
图 16-2 比较器内建基准缓冲器 210
图 16-3 比较器输出滞回 212
图 16-4 比较器输出屏蔽功能 212
图 16-5 数字滤波（DFLEN=3）波形示意图 213
图 16-6 比较器中断产生 214
图 16-7 比较器输出逻辑 214
图 16-8 比较器输出上升沿产生触发输出，无滤波 215
图 16-9 比较器输出上升沿下降沿产生触发输出，无滤波 215
图 18-1 I²C-SMB 模块框图 234
图 18-2 I²C 总线时序 236
图 18-3 数据有效时序 236
图 18-4 起始（START）与停止(STOP)命令定义 236
图 18-5 输出应答(ACK) 237
图 18-6 I²C 时序参数图例 238
图 18-7 主机向 7 位地址从机写入数据时的帧格式 242
图 18-8 I²C 主机软件发送数据流图 243
图 18-9 I²C 主机对 7 位地址从机发送数据流图 244
图 18-10 主机从 7 位地址从机读取数据时的帧格式 244
图 18-11 I²C 软件接收数据流图 245
图 18-12 I²C 从 7 位地址从机读取数据流图 246
图 18-13 I²C 主机接收数据等待 246
图 18-14 双向数据通信帧格式 247
图 18-15 10BIT 寻址，主机向从机写入数据 247
图 18-16 I²C 软件发送数据流图 248
图 18-17 10BIT 寻址，主机从从机读取数据 249
图 18-18 I²C 软件发送数据流图 250
图 18-19 I²C 软件发送数据流图 251
图 18-20 I²C 主机 DMA 发送流程图 252
图 18-21 I²C 主机 DMA 接收流程图 254
图 18-22 I²C 多主机时钟同步 255
图 18-23 I²C 多主机数据仲裁 256
图 18-24 主机时序控制 257
图 18-25 I²C 从机数据发送波形 259
图 18-26 I²C 从机数据接收波形 260
图 18-27 I²C 从机数据接收波形（SCLSEN=0，接收溢出） 261
图 18-28 从机地址匹配唤醒 262
图 18-29 I²C 从机 DMA 接收流程图 263
图 18-30 I2C 从机 DMA 发送流程图 264
图 18-31 SDA 输出延迟波形 265
图 18-32 TLOW:SEXT 和 TLOW:MEXT 波形 268
图 18-33 从机地址匹配唤醒 271
图 18-34 从机 START 唤醒 272
图 18-35 SMBUS 总线时钟超时事件 272
图 18-36 主机和从机超时事件 273
图 19-1 UART 接口时序 286
图 19-2 UART 字符描述 288
图 19-3 7816-3 帧结构图 289
图 19-4 起始位同步 291
图 19-5 位接收 16 倍采样 291
图 19-6 位接收 8 倍采样 291
图 19-7 UART 异步发送波形 1 293
图 19-8 UART 异步发送波形 2 294

图 19-9 UART 异步发送波形 3 294
图 19-10 7816 发送和重发 295
图 19-11 两个 UART 之间的硬件流控 296
图 19-12 nRTS 流控 296
图 19-13 nCTS 流控 297
图 19-14 红外调制波形 301
图 19-15 UART 发送延迟 302
图 19-16 LIN 总线拓扑 302
图 19-17 LIN 总线帧结构 303
图 19-18 LIN BREAK FIELD 303
图 19-19 LINSYNC FIELD 303
图 19-20 LIN 帧 ID 304
图 19-21 LIN 总线唤醒信号 306
图 19-22 NORMAL BREAK 308
图 19-23 BREAK ERROR 308
图 19-24 BREAK 超时 309
图 19-25 连续 BREAK 后正常接收 309
图 19-26 LIN 波特率自适应过程 310
图 19-27 波特率误差较大，无 PRE-SYNC 311
图 19-28 LIN 从机接收流程 312
图 19-29 字节间隔和应答间隔 312
图 19-30 LIN BIT SAMPLING 313
图 19-31 LIN BIT ERROR 冲突检测 314
图 19-32 LIN BIT ERROR 监测点 314
图 19-33 PRE-SYNC 后发现 BREAK 长度不足，BREAKERROR 316
图 19-34 SYNC 段数据错误 316
图 19-35 SYNC 段 0XFF 或 0X00，无 PRE-SYNC 317
图 20-1 LPUART 结构框图 339
图 20-2 字符描述 340
图 21-1 SPI 结构框图 351
图 21-2 SPI 数据/时钟时序图（CPHA=0） 352
图 21-3 SPI 数据/时钟时序图（CPHA=1） 353
图 21-4 4 线半双工写操作 353
图 21-5 4 线半双工读操作（无 DUMMY CYCLE） 354
图 21-6 4 线半双工读操作（有 DUMMY CYCLE） 354
图 21-7 SPI MASTER/SPI SLAVE 互连 355
图 21-8 SPI MASTER 发送，WAIT=0 357
图 21-9 SPI MASTER 发送，WAIT=1 357
图 21-10 SPI SSN 时序图（SSNM=1，CPHA=0） 358
图 21-11 SPI SSN 时序图（SSNM=0） 358
图 21-12 I2S 结构框图 361
图 21-13 PHILIPS 标准时序 362
图 21-14 32/16-BITS MSB-JUSTIFIED 时序 363
图 21-15 24-BITS MSB-JUSTIFIED 时序 363
图 21-16 LSB-JUSTIFIED 时序 364
图 21-17 PCM 标准时序 365
图 22-1 NIBBLE 波形示意图 380
图 22-2 数据 0 和数据 7 的 NIBBLE 波形 380
图 22-3 短消息格式 OPTION CHANNEL 381
图 22-4 加强消息格式 OPTION CHANNEL 382
图 22-5 消息帧格式 382
图 22-6 SENT 模块结构框图 383
图 23-1 CAN 模块结构框图 395

图 23-2 CAN BIT TIMING 396
图 23-3 节点错误状态转移 406
图 24-1 DMA 结构框图 428
图 24-2 外设请求处理 429
图 24-3 活动寄存器和影子寄存器 431
图 24-4 循环模式下的传输长度寄存器 432
图 24-5 循环模式下的源/目的地址寄存器和存储器指针 432
图 24-6 DOUBLE BUFFERING 433
图 25-1 CRC 运算流程图 447
图 25-2 使用 DMA 对 RAM 中的数据进行 CRC 运算 448
图 26-1 高级定时器结构框图 455
图 26-2 预分频从 1 变为 2 的波形 461
图 26-3 预分频从 1 变为 4 的波形 461
图 26-4 向上计数波形，内部时钟不分频 462
图 26-5 向上计数波形，内部时钟 2 分频 463
图 26-6 ARPE=0（ATIM_ARR 没有预装载）时的更新事件 463
图 26-7 ARPE=1（ATIM_ARR 预装载）时的更新事件 464
图 26-8 向下计数，内部时钟不分频 465
图 26-9 向下计数，内部时钟 2 分频 465
图 26-10 向下计数，内部时钟 2 分频 466
图 26-11 向下计数，不使用重复计数时的更新事件 466
图 26-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6 467
图 26-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢) 468
图 26-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出) 468
图 26-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置 469
图 26-16 ATIM 时钟源框图 471
图 26-17 内部时钟源模式，时钟分频因子为 1 471
图 26-18 TI2 外部时钟连接例子 472
图 26-19 外部时钟模式 1 下的时序 472
图 26-20 外部时钟模式 1 下的时序 473
图 26-21 外部触发输入框图 474
图 26-22 外部时钟模式 2 下的时序 1 474
图 26-23 外部时钟模式 2 下的时序 2 475
图 26-24 捕获/比较通道(通道 1 输入部分) 477
图 26-25 捕获/比较通道 1 的主电路 477
图 26-26 捕获/比较通道的输出部分(通道 1 至 3) 478
图 26-27 捕获/比较通道的输出部分(通道 4) 478
图 26-28 PWM 输入捕获模式时序 479
图 26-29 输出比较模式，翻转 OC1 482
图 26-30 边沿对齐的 PWM 波形(ARR=7, PWM MODE 1) 484
图 26-31 边沿对齐的 PWM 波形(PWM MODE 1) 484
图 26-32 中央对齐的 PWM 波形(APR=7, PWM MODE 1) 485
图 26-33 非对称 PWM 输出 486
图 26-34 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 1） 487
图 26-35 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 2） 487
图 26-36 通道 1/3 组合 PWM 输出 488
图 26-37 组合三相 PWM 输出 489
图 26-38 带死区插入的互补输出 490
图 26-39 死区波形延迟大于负脉冲 490
图 26-40 死区波形延迟大于正脉冲 490
图 26-41 死区插入后的互补输出波形与 CCxP 和 CCxNP 的关系 491
图 26-42 模式与输出状态术语 492
图 26-43 刹车输入结构 493

图 26-44 响应刹车的输出 495
图 26-45 产生六步 PWM，使用 COM 的例子(OSSR=1) 498
图 26-46 单脉冲模式的例子 499
图 26-47 重触发 OPM 模式 1，向上计数，CCR=0 500
图 26-48 重触发 OPM 模式 2，向上计数，CCR=0 501
图 26-49 重触发 OPM 模式 1，向下计数，CCR=0 501
图 26-50 重触发 OPM 模式 2，向下计数，CCR=0 502
图 26-51 ETR 信号清除 ATIM 的 OCxREF 502
图 26-52 编码器模式下的计数器操作实例 503
图 26-53 正交编码器不同边沿计数模式示例 504
图 26-54 时钟方向编码器示例 505
图 26-55 带方向的时钟编码器（CC1P=CC2P=0） 506
图 26-56 带方向的时钟编码器（CC1P=CC2P=1） 506
图 26-57 编码器 INDEX 定义 507
图 26-58 IPOS[1:0]=11 时的 INDEX 识别 508
图 26-59 IPOS[1:0]=11, INDEX GATED ON A, ARR=9 509
图 26-60 IPOS[1:0]=00, INDEX UNGATED, ARR=9 509
图 26-61 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9 510
图 26-62 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LEADING CHANNEL B 511
图 26-63 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING CHANNEL B 511
图 26-64 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING B, ZOOMED 512
图 26-65 IPOS[1:0]=01, UNGATED INDEX 512
图 26-66 IPOS[0]=1, CLOCK+DIRECTION MODE 513
图 26-67 IPOS[0]=1, DIRECTIONAL CLOCK MODE 514
图 26-68 正交编码状态转移 514
图 26-69 INDEX 错误撤销 515
图 26-70 INDEX 错误置位 515
图 26-71 向下计数时 INDEX 错误置位 516
图 26-72 COUNTER MODE PRELOAD 516
图 26-73 复位模式下的时序 517
图 26-74 门控模式下的时序 518
图 26-75 多 TIMER 同时使能 519
图 26-76 触发器模式下的时序 519
图 26-77 外部时钟模式 2＋触发模式下的时序 520
图 26-78 霍尔传感器接口 524
图 27-1 通用定时器架构示意图 559
图 27-2 预分频从 1 变为 2 的波形 564
图 27-3 预分频从 1 变为 4 的波形 564
图 27-4 向上计数波形，内部时钟不分频 565
图 27-5 向上计数波形，内部时钟 2 分频 566
图 27-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件 566
图 27-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件 567
图 27-8 向下计数，内部时钟不分频 568
图 27-9 向下计数，内部时钟 2 分频 568
图 27-10 向下计数，内部时钟 2 分频 569
图 27-11 向下计数，不使用重复计数时的更新事件 569
图 27-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6 570
图 27-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢) 571
图 27-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出) 571
图 27-15 GPTIM 时钟源框图 572
图 27-16 内部时钟源模式，时钟分频因子为 1 572
图 27-17 TI2 外部时钟连接例子 573
图 27-18 外部时钟模式 1 下的时序 573

图 27-19 外部时钟模式 1 下的时序 574
图 27-20 外部触发输入框图 575
图 27-21 外部时钟模式 2 下的时序 1 575
图 27-22 外部时钟模式 2 下的时序 2 576
图 27-23 捕获/比较通道(通道 1 输入部分) 578
图 27-24 捕获/比较通道 1 的主电路 579
图 27-25 捕获/比较通道的输出部分 579
图 27-26 PWM 输入捕获模式时序 580
图 27-27 输出比较模式，翻转 OC1 581
图 27-28 边沿对齐的 PWM 波形(ARR=7) 582
图 27-29 中央对齐的 PWM 波形(APR=7) 583
图 27-30 非对称 PWM 输出 584
图 27-31 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 1） 584
图 27-32 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 2） 585
图 27-33 通道 1/3 组合 PWM 输出 586
图 27-34 单脉冲模式的例子 587
图 27-35 ETR 信号清除 GPTIM 的 OCXREF 588
图 27-36 编码器模式下的计数器操作实例 589
图 27-37 正交编码器不同边沿计数模式示例 590
图 27-38 时钟方向编码器示例 591
图 27-39 带方向的时钟编码器（CC1P=CC2P=0） 592
图 27-40 带方向的时钟编码器（CC1P=CC2P=1） 592
图 27-41 编码器 INDEX 定义 593
图 27-42 IPOS[1:0]=11 时的 INDEX 识别 594
图 27-43 IPOS[1:0]=11, INDEX GATED ON A, ARR=9 595
图 27-44 IPOS[1:0]=00, INDEX UNGATED, ARR=9 595
图 27-45 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9 596
图 27-46 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LEADING CHANNEL B 597
图 27-47 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING CHANNEL B 597
图 27-48 IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING B, ZOOMED 598
图 27-49 IPOS[1:0]=01, UNGATED INDEX 598
图 27-50 IPOS[0]=1, CLOCK+DIRECTION MODE 599
图 27-51 IPOS[0]=1, DIRECTIONAL CLOCK MODE 600
图 27-52 正交编码状态转移 600
图 27-53 INDEX 错误撤销 601
图 27-54 INDEX 错误置位 601
图 27-55 向下计数时 INDEX 错误置位 602
图 27-56 COUNTER MODE PRELOAD 603
图 27-57 复位模式下的时序 603
图 27-58 门控模式下的时序 604
图 27-59 触发器模式下的时序 605
图 27-60 外部时钟模式 2＋触发模式下的时序 605
图 28-1 TAU 结构框图 633
图 28-2 TIMERX 结构框图 633
图 28-3 16BIT 连续定时功能 635
图 28-4 定时器内部时钟预分频计数 635
图 28-5 定时器外部事件上升沿递增时序 636
图 28-6 定时器外部事件上升下降沿递增，2 分频时序 636
图 28-7 定时器内部信号下降沿递增 637
图 28-8 定时器级联计数（TRIGGER TO INCREMENT） 638
图 28-9 定时器级联计数（TRIGGER TO INCREMENT），从机 PRESCALE1 = 1 638
图 28-10 定时器级联计数（TRIGGER TO START） 639
图 28-11 定时器级联计数（TRIGGER TO START），从机无法产生溢出 640

图 28-12 定时器级联计数（TRIGGER TO START），从机计数预分频 641
图 28-13 定时器捕捉外部输入信号上升沿 643
图 28-14 定时器 2 分频捕捉外部输入信号下降沿 643
图 28-15 定时器脉冲宽度高电平捕捉 644
图 28-16 TI00 二分频计数 646
图 28-17 TI00 二分频捕捉 646
图 28-18 16 位 PWM 输出示意图 647
图 28-19 非对称级联 PWM 输出示意图 648
图 28-20 从机 ARR 大于主机的情况 649
图 28-21 非对称非交叠 PWM 输出示意图 651
图 28-22 TIxx 输入数字滤波示意图 651
图 29-1 16 位基本定时器结构框图 659
图 29-2 预分频从 1 变为 2 的波形 661
图 29-3 预分频从 1 变为 4 的波形 661
图 29-4 向上计数波形，内部时钟不分频 662
图 29-5 向上计数波形，内部时钟 2 分频 663
图 29-6 ARPE=0（ARR 没有预装载）时的更新事件 663
图 29-7 ARPE=1（ARR 预装载）时的更新事件 664
图 29-8 内部时钟源模式，时钟分频因子为 1 664
图 30-1 LPTIM16 结构框图 673
图 30-2 外部 ETR 脉冲上升沿触发计数 674
图 30-3 外部 ETR 脉冲异步计数（下降沿） 675
图 30-4 TIMEOUT 模式 676
图 30-5 PWM 输出 677
图 30-6 输入信号边沿捕捉 678
图 30-7 通道输入数字滤波 678
图 30-8 编码器模式下的计数器操作实例 679
图 31-1 RTC 结构框图 690
图 31-2 LTBC 结构框图 691
图 31-3 RTC 时间读取流程图 695
图 32-1 LCD 显示控制模块结构框图 707
图 32-2 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE A) 709
图 32-3 LCD 驱动波形(1/4 DUTY, 1/3 BIAS, TYPE B) 710
图 32-4 LCD 片内电阻 BUFFER 型驱动电路 711
图 34-1 ADC 结构框图 742
图 34-2 ADC 采样转换时序 744
图 34-3 ADC 采样序列时序 744
图 34-4 ACQ 队列控制器 745
图 34-5 注入时序 748
图 34-6 SCAN_ACQ 单次转换全自动触发模式 749
图 34-7 SCAN_ACQ 单次转换半自动触发模式 750
图 34-8 SCAN_ACQ 连续转换模式 751
图 34-9 ADC 输入通道示意图 754
图 34-10 ADC 模拟看门狗阈值示意图 756
图 35-1 OPA 电路框图 775
图 35-2 OPA 非反相放大 776
图 35-3 OPA 反相放大 777
图 35-4 OPA 缓冲器模式 778
图 35-5 OPA 为 DAC 提供输出缓冲 778
图 36-1 DAC 结构框图 783
图 36-2 DAC 工作时序 784
图 36-3 DAC 连续输出模式 785
图 36-4 DAC 触发输出模式 786

图 37-1 LUT 结构框图 794
图 37-2 PGL 结构框图 794
图 37-3 LUT 实现 2 输入 NAND 示意图 796
图 37-4 输出滤波和采样 797
图 37-5 数字滤波 797
图 37-6 使用 LUT0 为 UART 输入做数字滤波 798
图 38-1 普通 GPIO 结构框图 809
图 38-2 WKUPx 功能结构框图 814
图 38-3 引脚输入数字滤波 815
图 38-4 EXTI 信号输入示意图 815
图 41-1 CORTEX-M0 调试系统示意图 847

1 产品综述

1.1 概述

FM33LF0A的主要特性如下：

- 宽电压范围：1.8~5.5V
- 工作温度范围（T_A）：-40℃~+125℃
- 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72MHz
  - 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100uA/MHz@72Mhz
  - Sleep模式：3.5uA@5V
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：1.3uA@5V
- 存储器
  - 64KB Flash w/ ECC
  - Flash擦写寿命：100,000次
  - Flash数据保存时间：15年@average 85℃
  - 用户代码保护
  - 8KB RAM w/ parity
- 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）
  - 4x快速模拟比较器
  - 2xOPA
  - 12bit 2Msps SAR-ADC
  - 10bit DAC
  - 温度传感器，精度+/-5℃

- 通用通信接口
  - UART*5（具备SmartCard和LIN功能）
  - LPUART*2
  - SPI*2
  - I2CSMBus*2
  - FSCAN*1，支持CAN2.0B
  - SENT*2
  - 4通道外设DMA
  - 可编程CRC校验模块
- 定时资源
  - 16bit高级定时器*1
  - 16bit通用定时器*2
  - 16bit定时器阵列（TAU）*8ch
  - 16bit基本定时器*1
  - 24-bit Systick*1
  - 16bit低功耗定时器*1
  - 看门狗定时器*2
  - 低功耗实时时钟日历（RTCC），带有数字调校功能，调校精度+/-0.476ppm
- LCD显示控制电路
  - 支持(1/2/3/4)COM*32SEG，6COM*30SEG，8COM*28SEG
  - 1/3 bias、1/4bias
  - 片内电阻分压
  - 支持休眠显示
- LED驱动电路
  - 10COM*16DIG
  - PWMdimming
  - Guardband
- 支持除法器和开方运算
- 加密算法
  - 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，出厂调校误差+/-0.5%，8MHz全温区变化小于+/-1%

- 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，32KHz，全温区+/-3%
- 高频晶体振荡器，4~24MHz
- PLL，最高输出72MHz

- 封装：LQFP48、QFN32

1.2 芯片结构框图

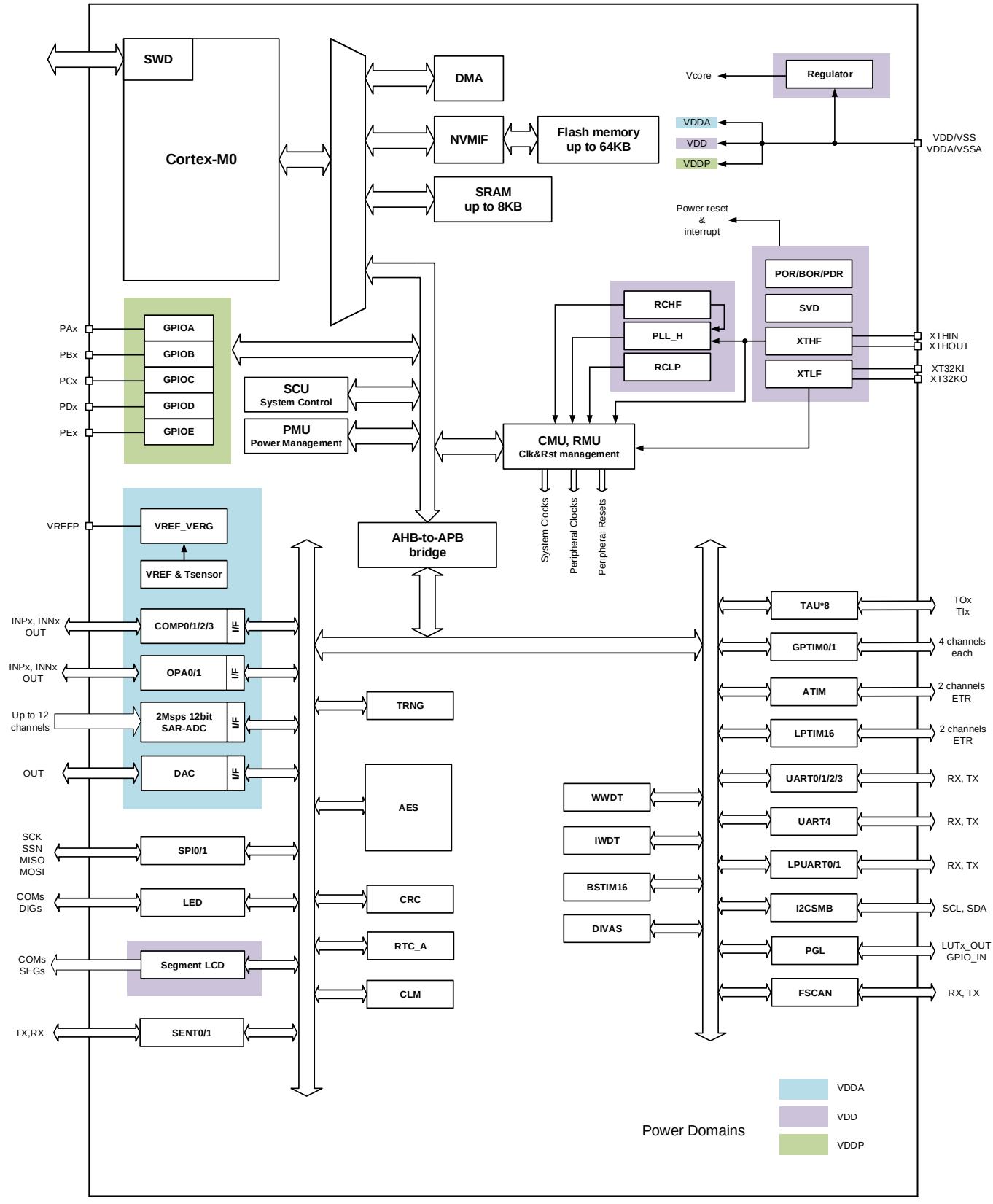


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Flash 容量（KBytes）	RAM 容量（KBytes）	封装
FM33LF015A	64	8	LQFP48
FM33LF013A	64	8	QFN32

表 1-1 型号列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP64 封装图（FM33LF0x6A）

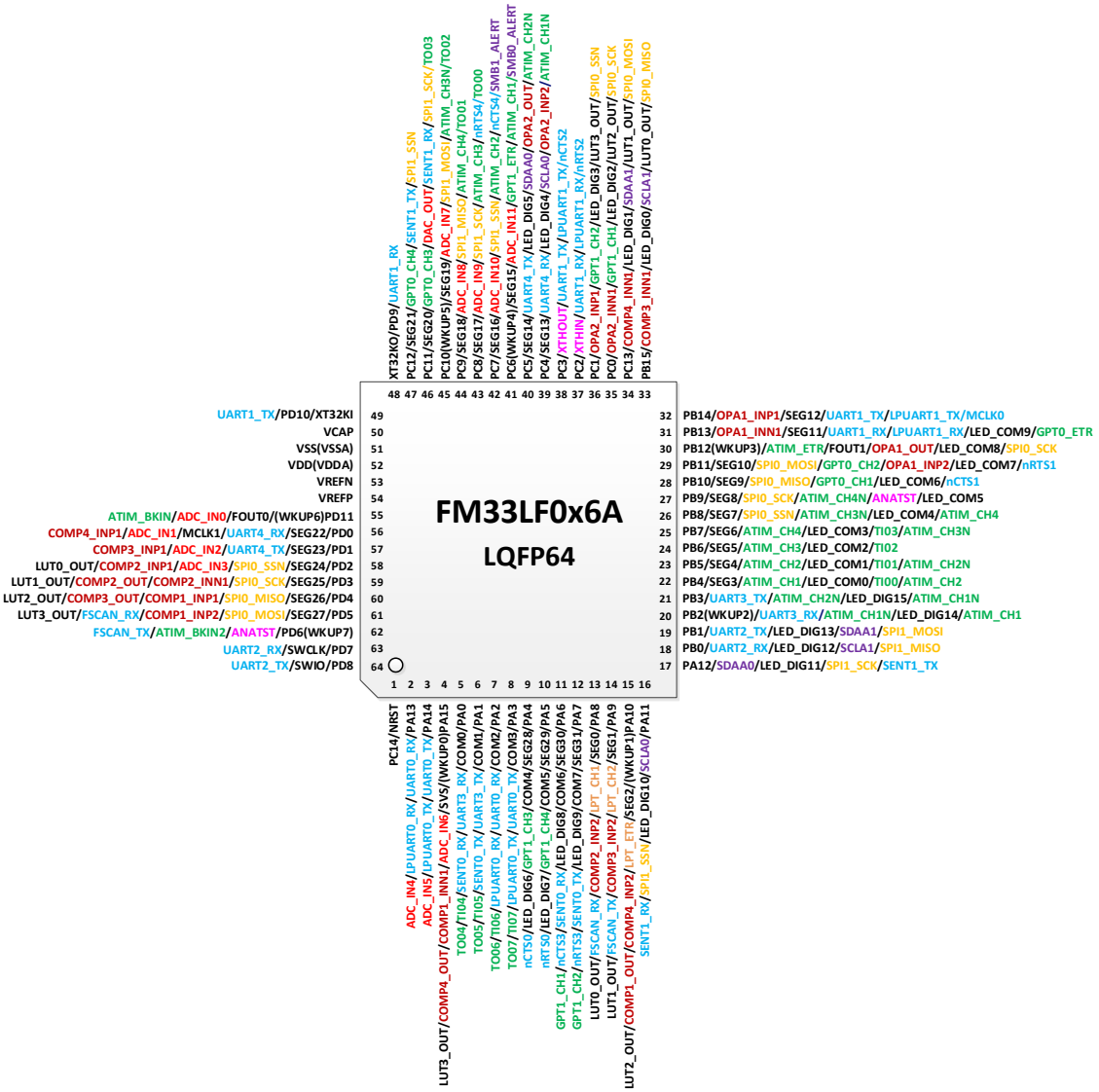


图 2-1 FM33LF0x6A LQFP64 封装图

2.1.2 LQFP48 封装图（FM33LF0x5A）

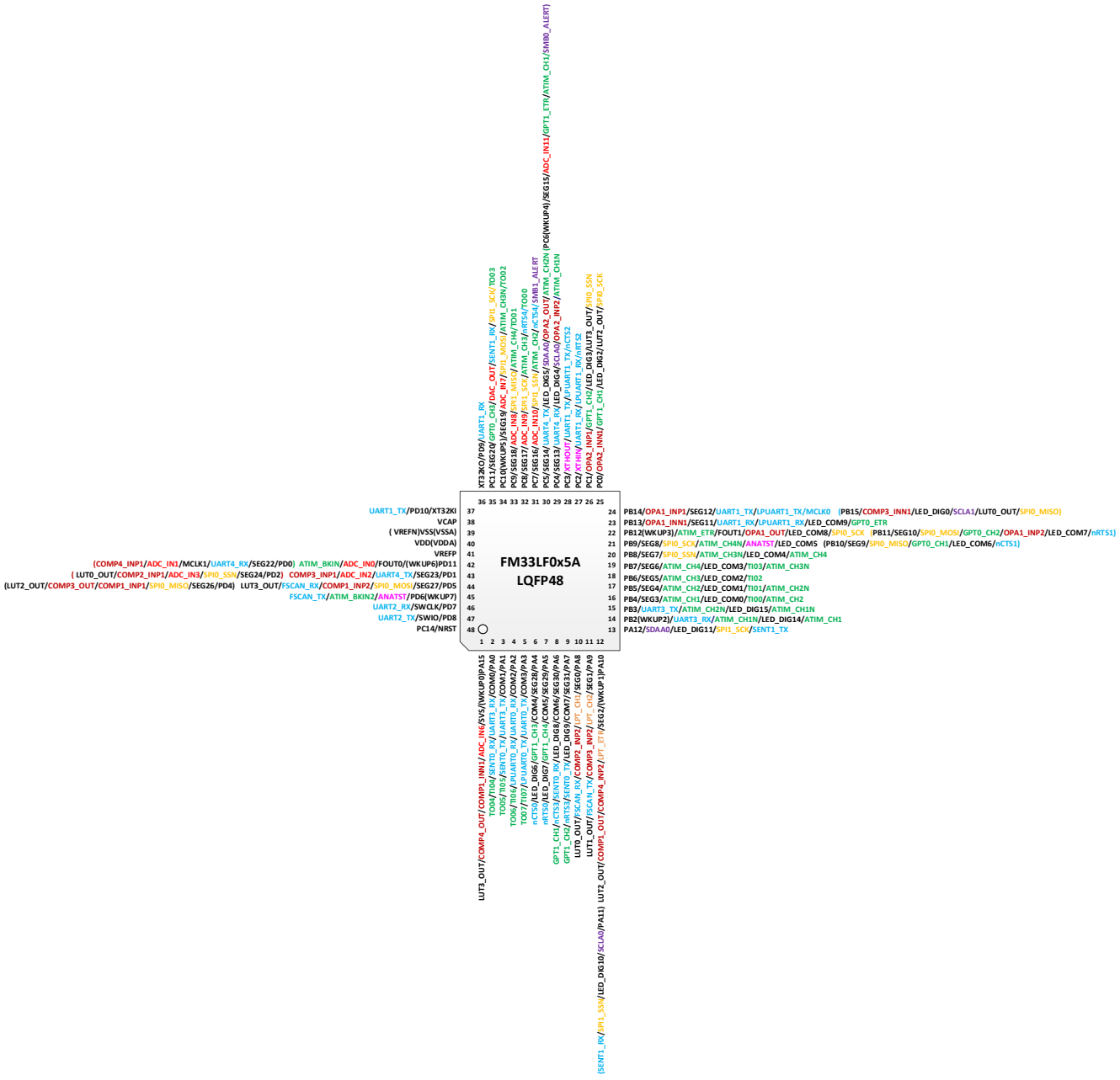


图 2-2 FM33LF0x5A LQFP48 封装图

2.1.3 QFN32 封装图（FM33LF0x3A）

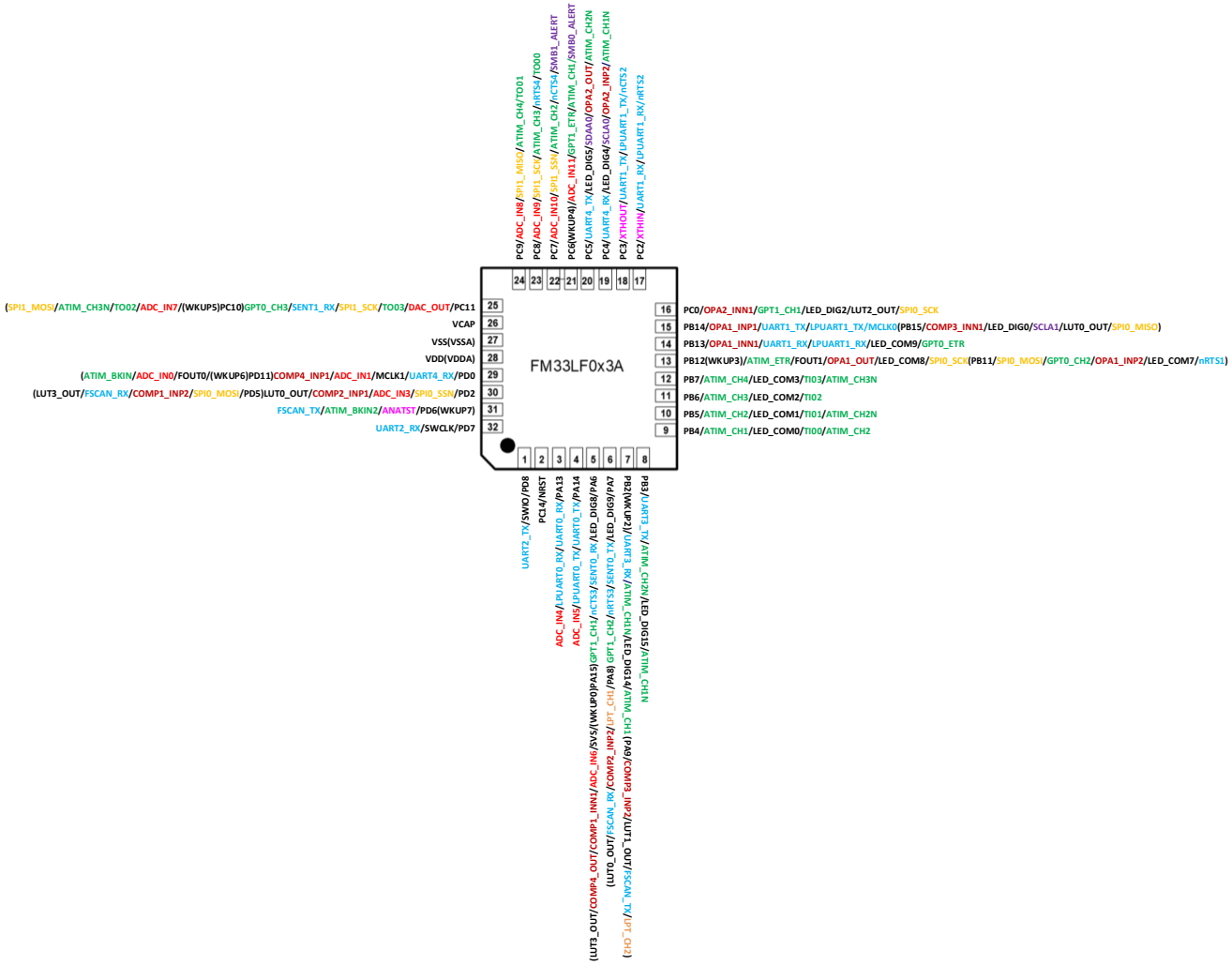


图 2-3 FM33LF0x3A QFN32 封装图

2.1.4 引脚功能定义

LQFP64	LQFP48	QFN32	Pin Function	Descriptions
1	48	2	NRST	全局复位引脚
			PC14	GPIO
2		3	PA13	GPIO
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
			ADC_IN4	ADC 输入通道
3		4	PA14	GPIO
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
			ADC_IN5	ADC 输入通道
4	1	5	PA15	GPIO

			WKUP0	外部唤醒引脚
			SVS	外部电源检测
			ADC_IN6	ADC 输入通道
			COMP1_INN1	比较器输入
			COMP4_OUT	比较器输出
			LUT3_OUT	LUT3 输出
5	2		PA0	GPIO
			COM0	LCD 驱动 COM 端
			UART3_RX	UART 接收
			SENT0_RX	SENT 输入
			TI04	定时器输入
			TO04	定时器输出
6	3		PA1	GPIO
			COM1	LCD 驱动 COM 端
			UART3_TX	UART 发送
			SENT0_TX	SENT 输出
			TI05	定时器输入
			TO05	定时器输出
7	4		PA2	GPIO
			COM2	LCD 驱动 COM 端
			UART0_RX	UART 接收
			LPUART0_RX	低功耗 UART 接收
			TI06	定时器输入
			TO06	定时器输出
8	5		PA3	GPIO
			COM3	LCD 驱动 COM 端
			UART0_TX	UART 发送
			LPUART0_TX	低功耗 UART 发送
			TI07	定时器输入
			TO07	定时器输出
9	6		PA4	GPIO
			COM4/SEG28	LCD 驱动 COM/SEG 端
			GPT1_CH3	通用定时器外部通道
			LED_DIG6	LED 输出
			nCTS0	UART 硬件流控
10	7		PA5	GPIO
			COM5/SEG29	LCD 驱动 COM/SEG 端
			GPT1_CH4	通用定时器外部通道
			LED_DIG7	LED 输出
			nRTS0	UART 硬件流控
11	8	5	PA6	GPIO
			COM6/SEG30	LCD 驱动 COM/SEG 端
			SENT0_RX	SENT 输入

			LED_DIG8	LED 输出
			nCTS3	UART 硬件流控
			GPT1_CH1	GPTIM1 通道 1
12	9	6	PA7	GPIO
			COM7/SEG31	LCD 驱动 COM/SEG 端
			SENT0_TX	SENT 输出
			LED_DIG9	LED 输出
			nRTS3	UART 硬件流控
			GPT1_CH2	GPTIM1 通道 2
13	10	6	PA8	GPIO
			SEG0	LCD 驱动 SEG 端
			LPT_CH1	低功耗定时器外部通道
			FSCAN_RX	FSCAN 接收
			COMP2_INP2	比较器输入
			LUT0_OUT	LUT 输出
14	11	7	PA9	GPIO
			SEG1	LCD 驱动 SEG 端
			LPT_CH2	低功耗定时器外部通道
			FSCAN_TX	FSCAN 发送
			COMP3_INP2	比较器输入
			LUT1_OUT	LUT 输出
15	12		PA10	GPIO
			WKUP1	外部唤醒引脚
			SEG2	LCD 驱动 SEG 端
			LPT_ETR	低功耗定时器外部触发输入
			COMP1_OUT	比较器输出
			COMP4_INP2	比较器输入
			LUT2_OUT	LUT 输出
16	12		PA11	GPIO
			LED_DIG10	LED 输出
			SPI1_SSN	SPI 片选
			SENT1_RX	SENT 接收
			SCLA0	I2C 时钟
17	13		PA12	GPIO
			SENT1_TX	SENT 发送
			LED_DIG11	LED 输出
			SPI1_SCK	SPI 时钟
			SDAA0	I2C 数据
18			PB0	GPIO
			UART2_RX	UART 接收
			SCLA1	I2C 时钟
			LED_DIG12	LED 输出

			SPI1_MISO	SPI 数据
19			PB1	GPIO
			UART2_TX	UART 发送
			SDAA1	I2C 数据
			LED_DIG13	LED 输出
			SPI1_MOSI	SPI 数据
20	14	7	PB2	GPIO
			WKUP2	外部唤醒引脚
			ATIM_CH1N	高级定时器通道
			UART3_RX	UART 接收
			LED_DIG14	LED 输出
			ATIM_CH1	高级定时器通道
21	15	8	PB3	外部唤醒引脚
			ATIM_CH2N	高级定时器通道
			UART3_TX	UART 发送
			LED_DIG15	LED 输出
			ATIM_CH1N	高级定时器通道
22	16	9	PB4	GPIO
			SEG3	LCD 驱动 SEG 端
			ATIM_CH1	高级定时器通道
			LED_COM0	LED 输出
			TI00	定时器输入
			ATIM_CH2	高级定时器通道
23	17	10	PB5	GPIO
			SEG4	LCD 驱动 SEG 端
			ATIM_CH2	高级定时器通道
			LED_COM1	LED 输出
			TI01	定时器输入
			ATIM_CH2N	高级定时器通道
24	18	11	PB6	GPIO
			SEG5	LCD 驱动 SEG 端
			ATIM_CH3	高级定时器通道
			LED_COM2	LED 输出
			TI02	定时器输入
25	19	12	PB7	GPIO
			SEG6	LCD 驱动 SEG 端
			ATIM_CH4	高级定时器通道
			LED_COM3	LED 输出
			TI03	定时器输入
			ATIM_CH3N	高级定时器通道
26	20		PB8	GPIO
			SEG7	LCD 驱动 SEG 端
			ATIM_CH3N	高级定时器通道

			LED_COM4	LED 输出
			SPI0_SSN	SPI 片选
			ATIM_CH4	高级定时器通道
27	21		PB9	GPIO
			SEG8	LCD 驱动 SEG 端
			ANTEST	模拟测试通道
			SPI0_SCK	SPI 时钟
			LED_COM5	LED 输出
			ATIM_CH4N	高级定时器通道
28			PB10	GPIO
			SEG9	LCD 驱动 SEG 端
			LED_COM6	LED 输出
			SPI0_MISO	SPI 数据线
			GPT0_CH1	通用定时器外部通道
			nCTS1	UART 硬件流控
29	22	13	PB11	GPIO
			SEG10	LCD 驱动 SEG 端
			OPA1_INP2	OPA 输入
			LED_COM7	LED 输出
			SPI0_MOSI	SPI 数据线
			GPT0_CH2	通用定时器外部通道
			nRTS1	UART 硬件流控
30	22	13	PB12	GPIO
			WKUP3	外部唤醒引脚
			FOUT1	时钟频率输出
			ATIM_ETR	高级定时器外部触发
			OPA1_OUT	OPA 输出
			LED_COM8	LED 输出
			SPI0_SCK	SPI 时钟
31	23	14	PB13	GPIO
			SEG11	LCD 驱动 SEG 端
			OPA1_INN1	OPA 输入
			LED_COM9	LED 输出
			GPT0_ETR	通用定时器外部触发输入
			UART1_RX	UART 接收
			LPUART1_RX	LPUART 接收
32	24	15	PB14	GPIO
			SEG12	LCD 驱动 SEG 端
			OPA1_INP1	OPA 输入
			MCLK0	I2S 时钟输出
			UART1_TX	UART 发送
			LPUART1_TX	LPUART 发送

33	24	15	PB15	GPIO
			SPI0_MISO	SPI 数据线
			SCLA1	I2C 时钟
			LED_DIG0	LED 输出
			LUT0_OUT	LUT 输出
			COMP3_INN1	比较器输入
34			PC13	GPIO
			SDAA1	I2C 数据
			LED_DIG1	LED 输出
			LUT1_OUT	LUT 输出
			COMP4_INN1	比较器输入
			SPI0_MOSI	SPI 数据线
35	25	16	PC0	GPIO
			LUT2_OUT	LUT 输出
			OPA2_INN1	OPA 输入
			LED_DIG2	LED 输出
			GPT1_CH1	通用定时器外部通道
			SPI0_SCK	SPI 时钟线
36	26		PC1	GPIO
			LUT3_OUT	LUT 输出
			OPA2_INP1	OPA 输入
			LED_DIG3	LED 输出
			GPT1_CH2	通用定时器外部通道
			SPI_SSN	SPI 片选信号
37	27	17	PC2	GPIO
			XTHIN	高频晶振输入
			UART1_RX	UART 接收
			LPUART1_RX	低功耗 UART 接收
			nRTS2	UART 硬件流控
38	28	18	PC3	GPIO
			XTHOUT	高频晶振输出
			UART1_TX	UART 发送
			LPUART1_TX	低功耗 UART 发送
			nCTS2	UART 硬件流控
39	29	19	PC4	GPIO
			SEG13	LCD 驱动 SEG 端
			OPA2_INP2	OPA 输入
			ATIM_CH1N	高级定时器通道
			SCLA0	I2C 时钟
			UART4_RX	UART 接收
			LED_DIG4	LED 输出
40	30	20	PC5	GPIO
			SEG14	LCD 驱动 SEG 端

			OPA2_OUT	OPA 输出
			ATIM_CH2N	高级定时器通道
			SDAA0	I2C 数据
			UART4_TX	UART 接收
			LED_DIG5	LED 输出
41	30	21	PC6	GPIO
			WKUP4	外部唤醒引脚
			ADC_IN11	ADC 输入
			SEG15	LCD 驱动 SEG 端
			GPT1_ETR	通用定时器外部触发输入
			ATIM_CH1	高级定时器通道
			SMB0_ALERT	SMBus 信号
42	31	22	PC7	GPIO
			SEG16	LCD 驱动 SEG 端
			SPI1_SSN	SPI 片选
			ATIM_CH2	高级定时器通道
			ADC_IN10	ADC 输入通道
			nCTS4	UART 硬件流控
			SMB1_ALERT	SMBus 信号
43	32	23	PC8	GPIO
			SEG17	LCD 驱动 SEG 端
			SPI1_SCK	SPI 时钟
			ATIM_CH3	高级定时器通道
			ADC_IN9	ADC 输入通道
			nRTS4	UART 硬件流控
			TO00	定时器输出
44	33	24	PC9	GPIO
			SEG18	LCD 驱动 SEG 端
			SPI1_MISO	SPI 数据
			ATIM_CH4	高级定时器通道
			ADC_IN8	ADC 输入通道
			TO01	定时器输出
45	34	25	PC10	GPIO
			WKUP5	外部唤醒引脚
			SEG19	LCD 驱动 SEG 端
			SPI1_MOSI	SPI 数据
			ATIM_CH3N	高级定时器通道
			ADC_IN7	ADC 输入通道
			TO02	定时器输出
46	35	25	PC11	GPIO
			SEG20	LCD 驱动 SEG 端
			SENT1_RX	SENT 接收

			GPT0_CH3	通用定时器外部通道
			DAC_OUT	DAC 输出
			SPI1_SCK	SPI 时钟
			TO03	定时器输出
47			PC12	GPIO
			SEG21	LCD 驱动 SEG 端
			SENT1_TX	SENT 发送
			GPT0_CH4	通用定时器外部通道
			SPI1_SSN	SPI 片选
48	36		PD9	GPIO
			UART1_RX	UART 接收
			XT32KO	32768Hz 晶振输出脚
49	37		PD10	GPIO
			UART1_TX	UART 发送
			XT32KI	32768Hz 晶振输入脚
50	38	26	VCAP	内核 LDO 输出
51	39	27	VSS	地
52	40	28	VDD	电源
53			VREFN	ADC 基准负端
54	41		VREFP	ADC 基准正端
55	42	29	PD11	GPIO
			WKUP6	外部唤醒引脚
			FOUT0	时钟频率输出
			ATIM_BKIN	高级定时器刹车输入
			ADC_IN0	ADC 输入通道
56	42	29	PD0	GPIO
			SEG22	LCD 驱动 SEG 端
			UART4_RX	UART 接收
			COMP4_INP1	比较器输入
			ADC_IN1	ADC 输入通道
			MCLK1	I2S 时钟输出
57	43		PD1	GPIO
			SEG23	LCD 驱动 SEG 端
			UART4_TX	UART 发送
			COMP3_INP1	比较器输入
			ADC_IN2	ADC 输入通道
58	43	30	PD2	GPIO
			SEG24	LCD 驱动 SEG 端
			SPI0_SSN	SPI 片选
			COMP2_INP1	比较器输入
			ADC_IN3	ADC 输入通道
			LUT0_OUT	LUT 输出
59			PD3	GPIO

			SEG25	LCD 驱动 SEG 端
			SPI0_SCK	SPI 时钟
			COMP2_INN1	比较器输入
			COMP2_OUT	比较器输出
			LUT1_OUT	LUT 输出
60	44		PD4	GPIO
			SEG26	LCD 驱动 SEG 端
			SPI0_MISO	SPI 数据
			COMP1_INP1	比较器输入
			COMP3_OUT	比较器输出
			LUT2_OUT	LUT 输出
61	44	30	PD5	GPIO
			SEG27	LCD 驱动 SEG 端
			SPI0_MOSI	SPI 数据
			COMP1_INP2	比较器输入
			FSCAN_RX	FSCAN 接收
			LUT3_OUT	LUT 输出
62	45	31	PD6	GPIO
			WKUP7	外部唤醒引脚
			ANATST	模拟测试通道
			FSCAN_TX	FSCAN 发送
			ATIM_BKIN2	高级定时器刹车输入
63	46	32	PD7	GPIO
			SWCLK	SWD 接口时钟
			UART2_RX	UART 接收
64	47	1	PD8	GPIO
			SWIO	SWD 接口数据
			UART2_TX	UART 发送

表 2-1 引脚列表

2.1.5 PA 引脚组数字外设功能

GPIO	PADFS			
	00	01	10	11
PA0	UART3_RX	SENT0_RX	TI04	TO04
PA1	UART3_TX	SENT0_TX	TI05	TO05
PA2	UART0_RX	LPUART0_RX	TI06	TO06
PA3	UART0_TX	LPUART0_TX	TI07	TO07
PA4	GPT1_CH3	LED_DIG6	nCTS0	-
PA5	GPT1_CH4	LED_DIG7	nRTS0	-
PA6	LED_DIG8	SENT0_RX	nCTS3	GPT1_CH1
PA7	LED_DIG9	SENT0_TX	nRTS3	GPT1_CH2
PA8	LPT_CH1	FSCAN_RX	LUT0_OUT	-
PA9	LPT_CH2	FSCAN_TX	LUT1_OUT	-

GPIO	PADFS			
	00	01	10	11
PA10	LPT_ETR	COMP1_OUT	LUT2_OUT	-
PA11	SCLA0	LED_DIG10	SPI1_SSN	SENT1_RX
PA12	SDAA0	LED_DIG11	SPI1_SCK	SENT1_TX
PA13	UART0_RX	LPUART0_RX	-	-
PA14	UART0_TX	LPUART0_TX	-	-
PA15	COMP4_OUT	LUT3_OUT	-	-

2.1.6 PB 引脚组数字外设功能

GPIO	PBDFS			
	00	01	10	11
PB0	UART2_RX	LED_DIG12	SCLA1	SPI1_MISO
PB1	UART2_TX	LED_DIG13	SDAA1	SPI1_MOSI
PB2	UART3_RX	ATIM_CH1N	LED_DIG14	ATIM_CH1
PB3	UART3_TX	ATIM_CH2N	LED_DIG15	ATIM_CH1N
PB4	ATIM_CH1	LED_COM0	TI00	ATIM_CH2
PB5	ATIM_CH2	LED_COM1	TI01	ATIM_CH2N
PB6	ATIM_CH3	LED_COM2	TI02	-
PB7	ATIM_CH4	LED_COM3	TI03	ATIM_CH3N
PB8	SPI0_SSN	ATIM_CH3N	LED_COM4	ATIM_CH4
PB9	SPI0_SCK	ATIM_CH4N	LED_COM5	-
PB10	SPI0_MISO	GPT0_CH1	LED_COM6	nCTS1
PB11	SPI0_MOSI	GPT0_CH2	LED_COM7	nRTS1
PB12	ATIM_ETR	FOUT1	LED_COM8	SPI0_SCK
PB13	UART1_RX	LPUART1_RX	LED_COM9	GPT0_ETR
PB14	UART1_TX	LPUART1_TX	-	MCLK0
PB15	LED_DIG0	SCLA1	LUT0_OUT	SPI0_MISO

2.1.7 PC 引脚组数字外设功能

GPIO	PCDFS			
	00	01	10	11
PC0	GPT1_CH1	LED_DIG2	LUT2_OUT	SPI0_SCK
PC1	GPT1_CH2	LED_DIG3	LUT3_OUT	SPI0_SSN
PC2	UART1_RX	LPUART1_RX	-	nRTS2
PC3	UART1_TX	LPUART1_TX	-	nCTS2
PC4	UART4_RX	LED_DIG4	SCLA0	ATIM_CH1N
PC5	UART4_TX	LED_DIG5	SDAA0	ATIM_CH2N
PC6	GPT1_ETR	ATIM_CH1	-	SMB0_ALERT
PC7	SPI1_SSN	ATIM_CH2	nCTS4	SMB1_ALERT
PC8	SPI1_SCK	ATIM_CH3	nRTS4	TO00
PC9	SPI1_MISO	ATIM_CH4	-	TO01
PC10	SPI1_MOSI	ATIM_CH3N	-	TO02
PC11	GPT0_CH3	SENT1_RX	SPI1_SCK	TO03
PC12	GPT0_CH4	SENT1_TX	SPI1_SSN	-

GPIO	PCDFS			
	00	01	10	11
PC13	LED_DIG1	SDAA1	LUT1_OUT	SPI0_MOSI
PC14	-	-	-	-

2.1.8 PD 引脚组数字外设功能

GPIO	PDDFS			
	00	01	10	11
PD0	UART4_RX	-	-	MCLK1
PD1	UART4_TX	-	-	-
PD2	SPI0_SSN	-	LUT0_OUT	-
PD3	SPI0_SCK	COMP2_OUT	LUT1_OUT	-
PD4	SPI0_MISO	COMP3_OUT	LUT2_OUT	-
PD5	SPI0_MOSI	FSCAN_RX	LUT3_OUT	-
PD6	ATIM_BKIN2	FSCAN_TX	-	-
PD7	SWCLK	UART2_RX	-	-
PD8	SWIO	UART2_TX	-	-
PD9	UART1_RX	-	-	-
PD10	UART1_TX	-	-	-
PD11	FOUT0	ATIM_BKIN	-	-

2.1.9 封装尺寸图

2.1.9.1 LQFP64

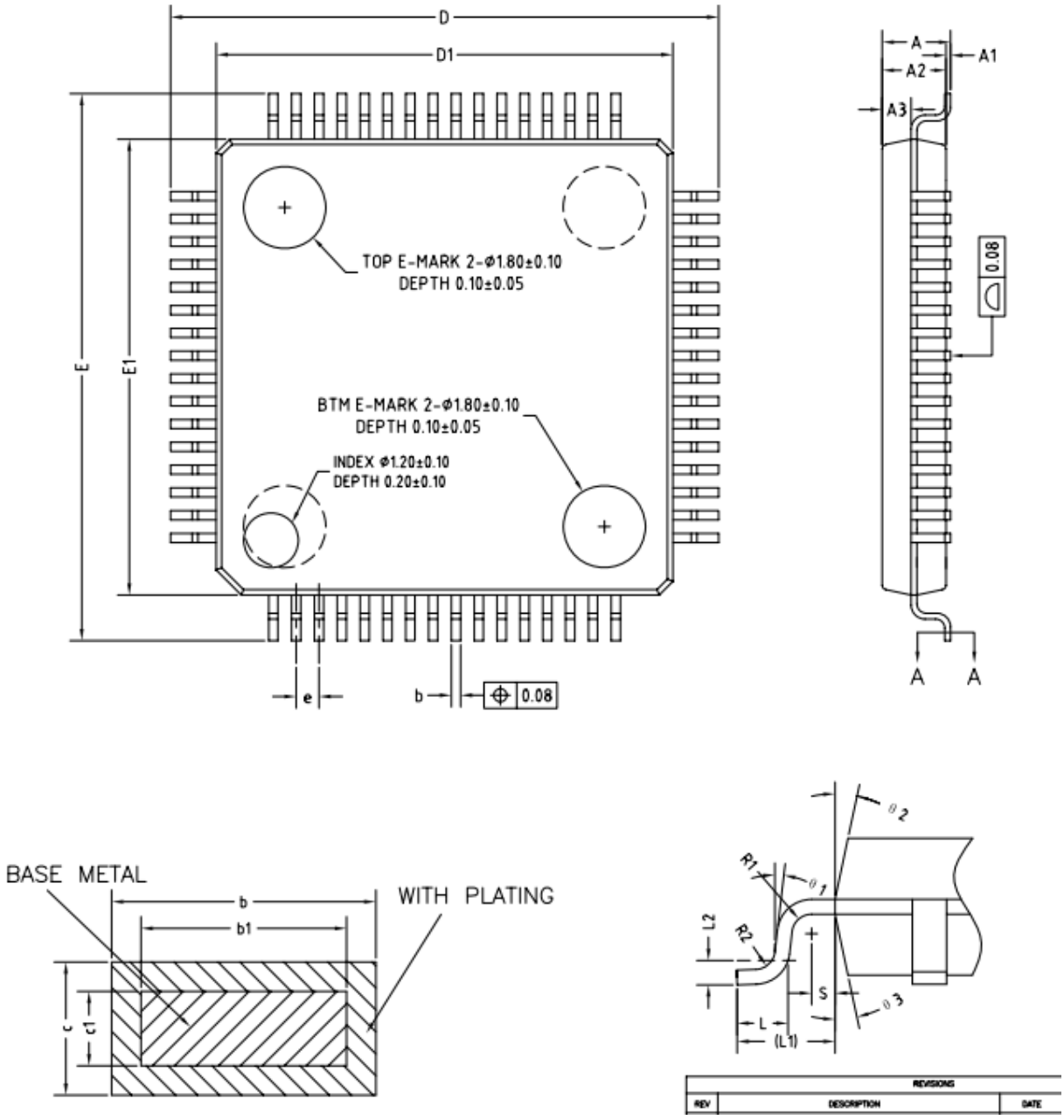


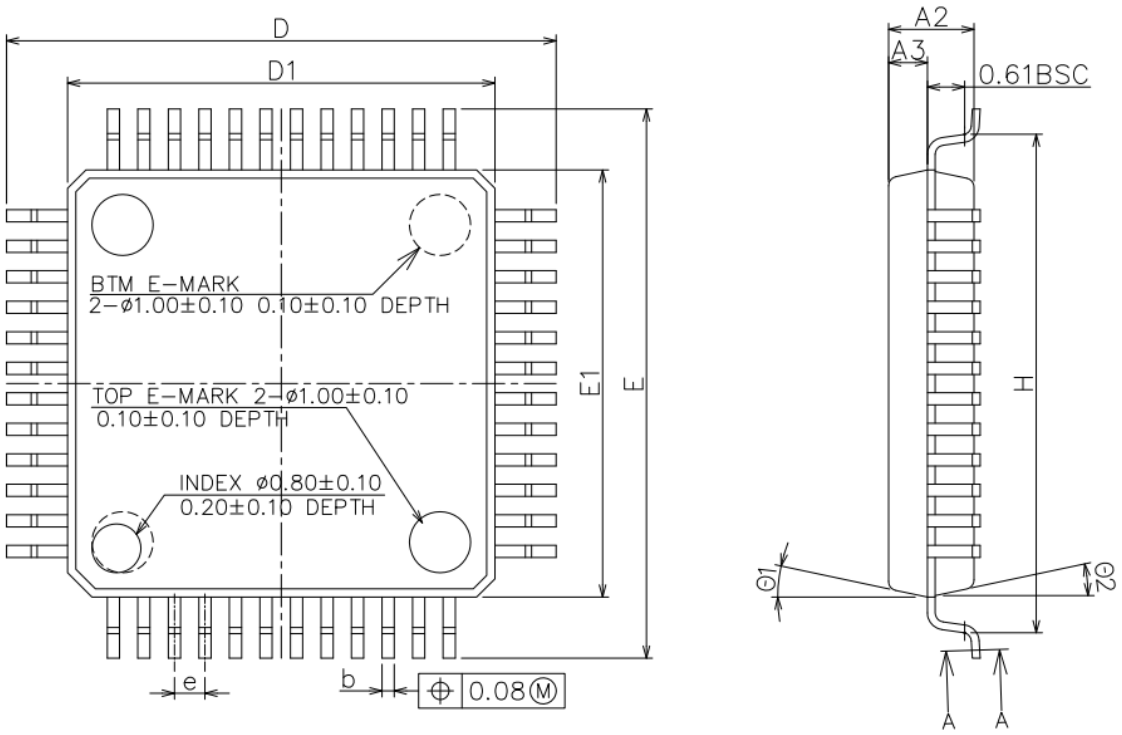
图 2-4 LQFP64 封装尺寸图

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

Symbol	MIN	NOM	MAX
D	11.80	12.00	12.20
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.

2.1.9.2 LQFP48



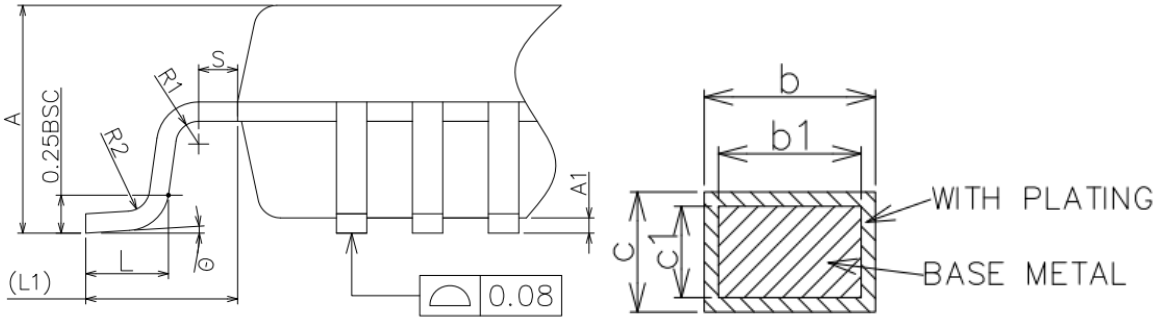


图 2-5 LQFP48 封装尺寸图

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	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	0.50BSC		
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 MS-026 BDD.

2.1.9.3 QFN32

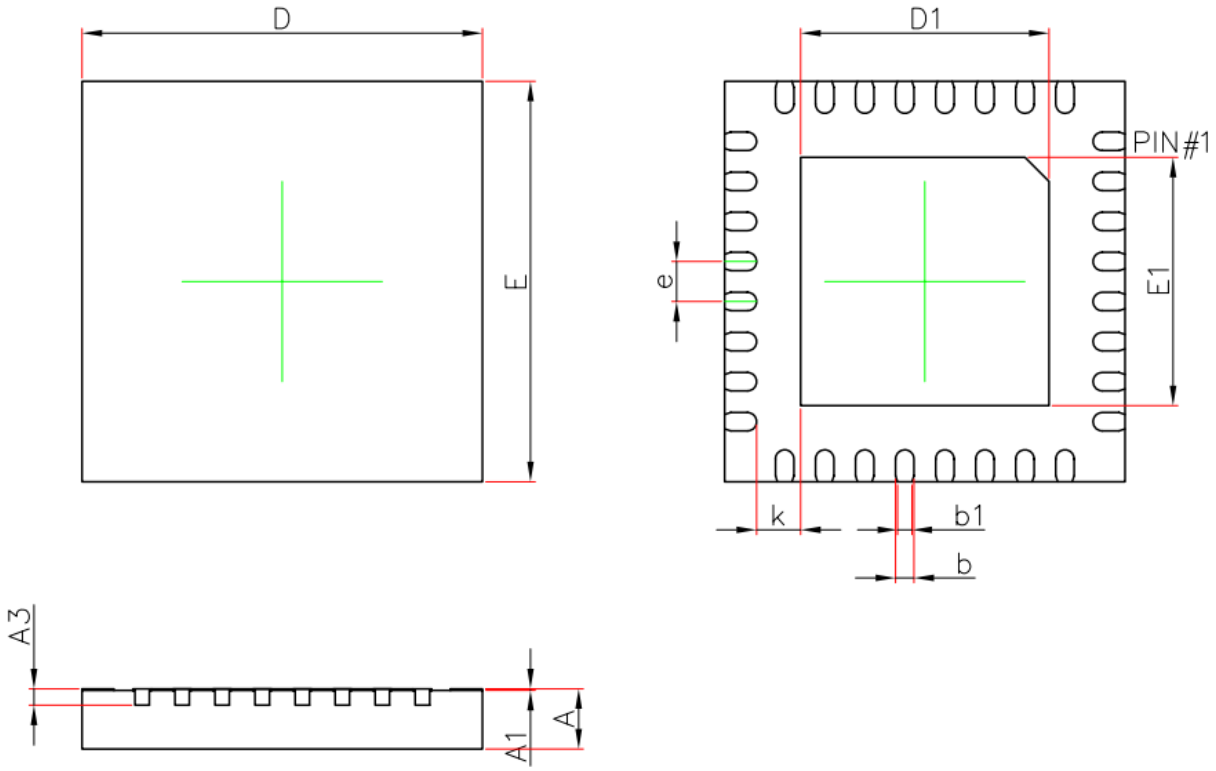


Figure 2-1 QFN32 outline

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700	0.800	0.028	0.031
A1	0.000	0.050	0.000	0.002
A3	0.203 REF.		0.008 REF.	
b	0.180	0.300	0.007	0.012
b1	0.130	0.230	0.005	0.009
D	4.900	5.100	0.193	0.201
D1	3.000	3.200	0.118	0.126
E	4.900	5.100	0.193	0.201
E1	3.000	3.200	0.118	0.126
e	0.500 BSC.		0.020 BSC.	
k	0.550 REF.		0.022 REF.	
L	0.324	0.476	0.013	0.019

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

2.2 焊接安装说明

复旦微电子芯片采用无铅工艺封装。回流焊工艺参数建议遵循JEDEC标准进行设定。

根据JEDEC标准J-STD-020，无铅工艺回流焊时的峰值温度设定建议如下表。用户可根据芯片不同厚度和体积的规格，在下表中选择合适的回流焊峰值温度。

封装厚度	塑封体体积 mm^3 <350	塑封体体积 mm^3 350 - 2000	塑封体体积 mm^3 >2000
<1.6mm	260℃	260℃	260℃
1.6~2.5 mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

下表给出了各种封装形式的回流焊峰值温度：

封装类型	塑封体厚度 mm	塑封体体积 mm^3	回流焊峰值温度
LQFP48	1.4	68.6	260℃

焊接曲线设定请参考JEDEC标准J-STD-020，无铅工艺回流焊温度曲线设定的说明进行设置。

Profile Feature	Pb-Free Assembly
Preheat/Soak Temperature Min (T_{smin}) Temperature Max (T_{smax}) Time (t_s) from (T_{smin} to T_{smax})	 150 °C 200 °C 60-120 seconds
Ramp-up rate (T_L to T_p)	3 °C/second max.
Liquidous temperature (T_L) Time (t_L) maintained above T_L	217 °C 60-150 seconds
Peak package body temperature (T_p)	For users T_p must not exceed the Classification temp in Table 4-2. For suppliers T_p must equal or exceed the Classification temp in Table 4-2.
Time (t_p)* within 5 °C of the specified classification temperature (T_c), see Figure 5-1.	30* seconds
Ramp-down rate (T_p to T_L)	6 °C/second max.
Time 25 °C to peak temperature	8 minutes max.

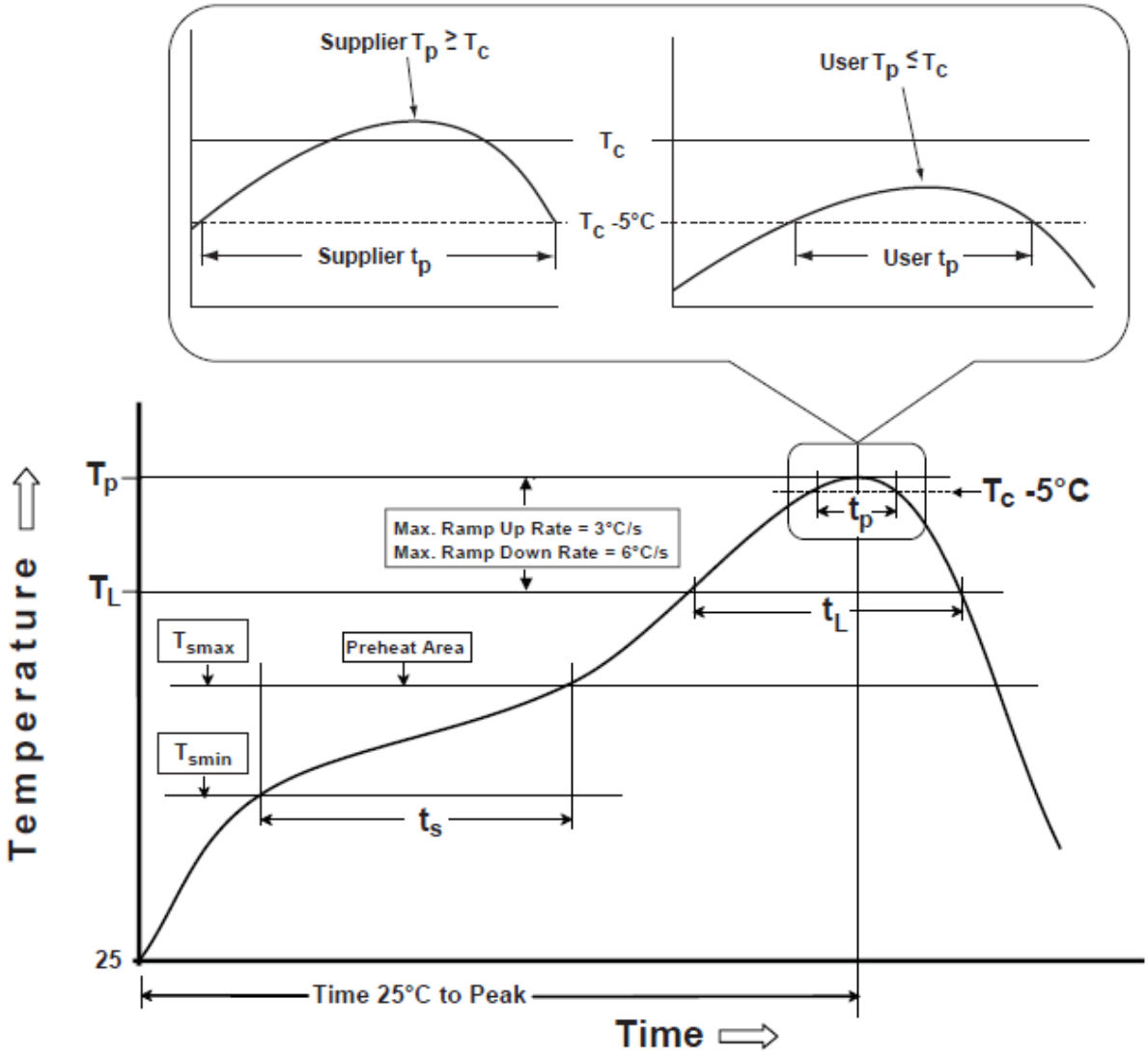


图 2-6JEDEC 标准的耐热回流温度曲线

特别声明：

- 芯片在上板焊接之前，请观察湿度卡是否变色以确认湿敏包装是否完好。
- 如无特殊指定，回流焊次数请勿超过3次

2.3 MSL 等级

本芯片湿敏等级为MSL3，依据JEDEC标准：J-STD-020。请在包装打开置于非干燥环境一周内，进行焊接操作。

2.4 热阻特性

芯片结温（T_J）可以通过以下公式计算。

$$T_J = T_A + P_D \times \Theta_{JA}$$

其中：

- T_A 为工作环境温度，单位°C
- Θ_{JA} 是封装热阻系数，单位°C/W
- P_D 是芯片功耗，包含内核功耗和 IO 功耗，单位 W；其中 IO 功耗由下式计算：

$$P_{IO} = \sum(V_{OL} \times I_{OL}) + \sum((V_{DDIO} - V_{OH}) \times I_{OH})$$

不同封装形式的热阻系数可以参考下表：

封装形式	封装尺寸	参考热阻 Θ_{JA}（℃/W）
LQFP48	7x7x1.4mm	50.3
QFN32	5x5x0.8mm	30

举例：

- 假设平均工作环境温度 T_A=55°C
- 芯片内核电流 IDD=5mA，VDD=3.6V
- 10 个 IO 同时输出低电平，每个 IO sink 5mA，VOL=0.3V
- 10 个 IO 同时输出高电平，每个 IO source 5mA，VOH=2.9V

计算芯片功耗：

$$P_D = P_{INT} + P_{IO} = 5mA \times 3.6V + 10 \times 5mA \times 0.3V + 10 \times (3.6V - 2.9V) \times 5mA = 68mW$$

对于 LQFP48 封装的产品，Θ_{JA}=50.3°C/W，因此可以计算芯片结温：

$$T_J = T_A + P_D \times \Theta_{JA} = 55 + 0.068 \times 50.3 = 58.4204°C$$

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 参数测试条件

3.2.1 供电方案

如无特别说明，芯片测试时采用下图所示的电源供电方案。

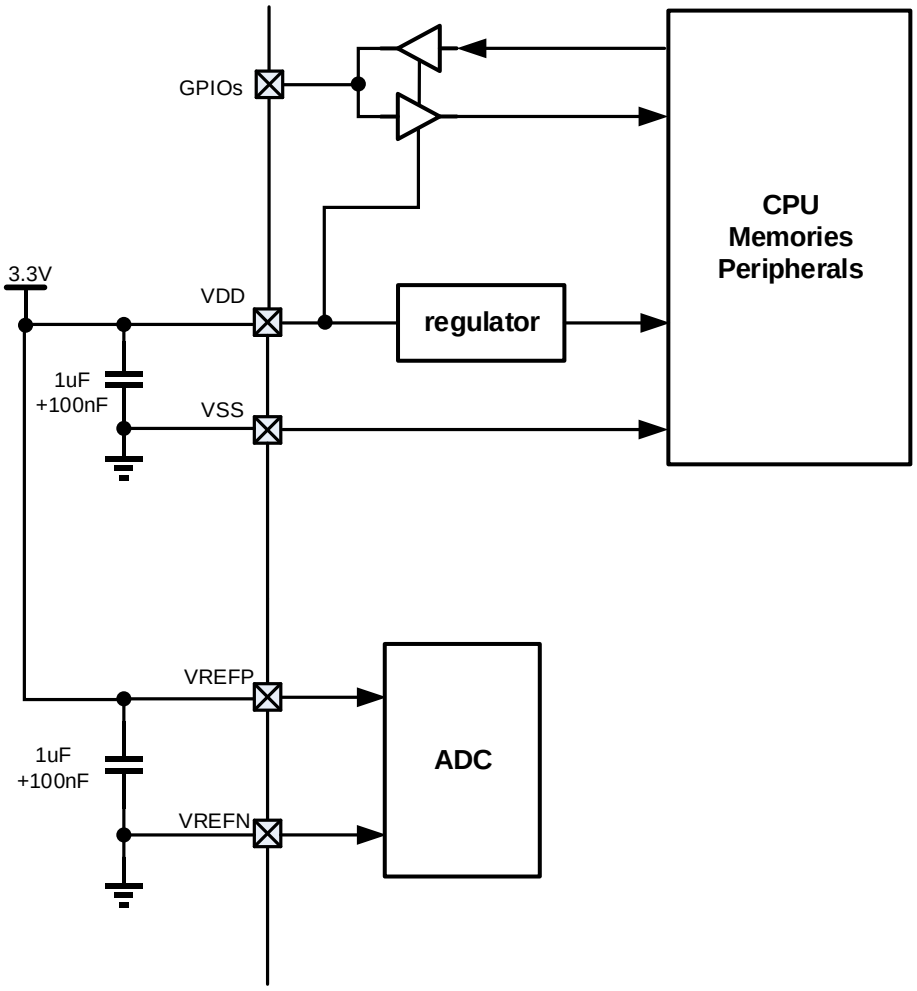


图 3-1 FM33LF0A 测试方案

注意，VDD 和 VREFP 需要使用同一电源。

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	V_{SS}-0.3	6.5	V
VREFP-VREFN	ADC 基准电压	-0.3	6.5	V
$\|\triangle V_{SS}\|$	所有地引脚之间的压差	-	50	mV
T_A	工作温度	-40	125	℃
T_J	芯片结温	-40	135	℃
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=125℃ 测试标准符合 JESD78E	-	+/-100	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 3-1 FM33LF0A 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	72	MHz
f_{PCLK}	APB 时钟频率	-	0	72	
VDD	典型工作电压范围		1.8	5.5	V

表 3-2 FM33LF0A 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=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}=16MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	2.3	-	mA
			T_A=125℃	-	2.4	-	
		f_{AHB}=72MHz PLL on Flash 2 wait	T_A=25℃	-	7.2	-	mA
			T_A=125℃		7.4		

表 3-3 ACTIVE 电流参数

3.4.2.2 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_{sleep1}	Sleep 模式电流	BOR、SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持	T_A=25℃	-	3.5		uA
			T_A=85℃		7		
			T_A=125℃	-	TBD		

表 3-4 SLEEP 电流参数

3.4.2.3 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	
$I_{dpsleep1}$	DeepSleep 模式电流	BOR、SVD 关闭 XTLF 关闭 CPU、RAM、外设数据保持	T_A=25℃	-	1.3	-	uA
			T_A=85℃		5		
			T_A=125℃	-	TBD	-	

表 3-5 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		2		∞	us/V
	电源下降速度	PDR	100		∞	us/V

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
		BOR		30		∞	us/V
T_{reset_delay}	上电复位延迟时间				0.5		ms
T_{pdr_filter}	下电复位滤波时间				4		us
V_{POR}	上电复位电压[1]	-40℃≤T_A≤125℃		1.45	1.55	1.65	V
V_{BOR}	下电复位电压 -40℃≤T_A≤125℃[1]	BORCFG==2'b000		1.65	1.75	1.85	V
		BORCFG==2'b001		1.85	1.95	2.05	
		BORCFG==2'b010		2.05	2.15	2.25	
		BORCFG==2'b011		2.23	2.35	2.47	
		BORCFG==2'b100		2.5	2.65	2.8	
		BORCFG==2'b101		2.8	2.95	3.1	
		BORCFG==2'b110		3.35	3.55	3.75	
		BORCFG==2'b111		3.65	3.85	4.05	
V_{PDR}	低功耗下电复位电压 -40℃≤T_A≤125℃[1]	PDRCFG==2'b00		1.23	1.33	1.4	V
		PDRCFG==2'b01		1.28	1.38	1.45	
		PDRCFG==2'b10		1.32	1.42	1.48	
		PDRCFG==2'b11		1.4	1.48	1.55	
V_{SVD}	电压监测阈值电平	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		
		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 3-6 复位和电源监控参数

注：

[1] 基于特征参数提取

3.4.4 高精度基准源

芯片内建高精度基准电压源，为 ADC 和 OPA 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。详细使用方法请参考复旦微电子驱动库函数。

符号	参数说明
REF_CAL	ADC 对 VREF 输出的转换值 测试条件： T_A=30±1℃ VDDA=3V±10mV
REF_RAW	VREF 输出的电压值 测试条件： T_A=30±1℃ VDDA=3V±10mV

高精度基准源主要参数：

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
V_{REF}	基准源输出电压[1]	-40℃≤T_A≤125℃		0.97	1.0	1.03	V
T_{coeff}	内部基准源温度系数	-40℃≤T_A≤125℃		-	25	100	ppm/℃
T_{S_VREF}	ADC 测量 VREF 时的采样时间	预先使能 VREF Buffer		10	-	-	us
T_{ADC_BUF}	驱动 ADC 输入的 VREF Buffer 的建立时间[1]	VDD=3V ADC 采样值稳定到 1LSB		-	-	100	us
I_{REF}	基准源工作电流[2]	T_A = 25℃ VDD=3V	PTAT_EN=0	-	1.8	-	uA
			PTAT_EN=1	-	2.6	-	

表 3-7 基准源参数

注：

[1] 基于特征参数提取，不包含在量产测试中

[2] 基于电路仿真

3.4.5 低功耗模式唤醒时间

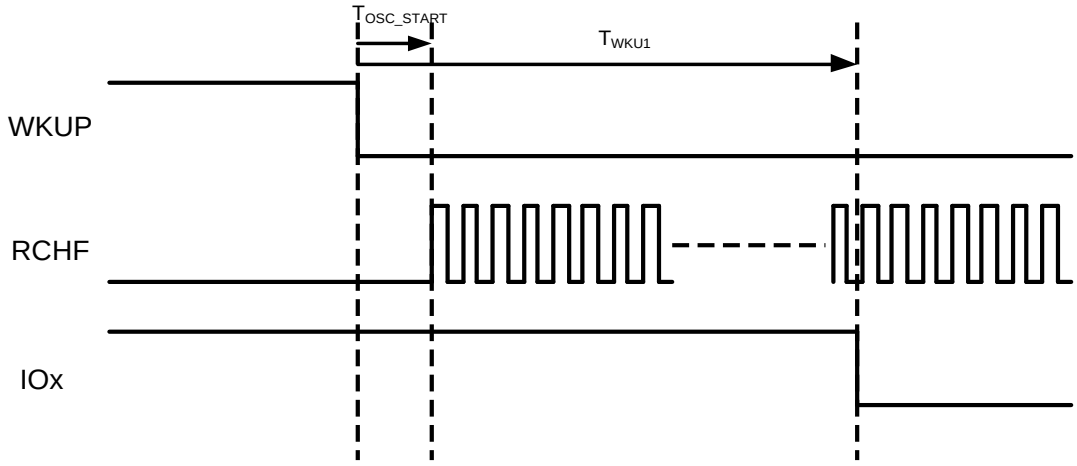
符号	参数说明	测试条件	参数值	单位

			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	-	6.8	-	us

表 3-8 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 3.5us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 6.8us。

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 3-9 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[1]	VDD=5V		4	-	32	MHz
R_{fb}	反馈电阻[2]	-		-	200	-	KΩ
gm	跨导增益[2]	VDD=5V	Gmcfg=1000	15.8	17.04	-	mA/V
		VDD=3.3V		14.32	15.06	-	
VDD_{min}	晶振最低工作电压[1]	8MHz, CFG=1000		1.14	-	-	V
		16MHz, CFG=1000		1.17	-	-	
		24MHz CFG=1000		1.24	-	-	
	陶振最低工作电压[1]	8MHz, CFG=1000		1.13	-	-	
		16MHz, CFG=1000		1.28	-	-	
IDD	晶振工作电流[1] VDD=5V	8MHz, CFG=1000		-	400	-	uA
		16MHz, CFG=1000		-	510	-	
		24MHz, CFG=1000		-	620	-	
	陶振工作电流[1] VDD=5V	8MHz, CFG=1000		-	398	-	
		16MHz, CFG=1000		-	606	-	
T_{start1}	晶振 8M 起振时间[1]	VDD=5V, CFG=1000		-	0.3	-	ms
		VDD=3.3V, CFG=1000		-	0.49	-	ms
T_{start2}	晶振 16M 起振时间[1]	VDD=5V, CFG=1000		-	0.19	-	ms
		VDD=3.3V, CFG=1000		-	0.26	-	ms
T_{start3}	晶振 24M 起振时间[1]	VDD=5V, CFG=1000		-	0.32	-	ms
		VDD=3.3V, CFG=1000		-	0.43	-	ms
T_{start4}	陶振 8M 起振时间[1]	VDD=5V, CFG=1000		-	0.028	-	ms
	陶振 16M 起振时间[1]	VDD=5V, CFG=1000		-	0.081	-	ms
C_L	负载电容[2]	-		5	-	25	pF

表 3-10 高频晶体振荡器参数

注：

[1]指标基于特征参数提取

[2]指标基于仿真

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=1.8~5.5V	FSEL==2'b00	7.96	8	8.04	MHz
			FSEL==2'b01	15.92	16	16.08	
			FSEL==2'b10	23.88	24	24.12	
ACC_{RCHF}[2]	全温区 RCHF 变化范围	VDD=1.8~5.5V	FSEL==2'b00 T_A=-40~+125℃	-2.5	-	2.5	%
			FSEL==2'b01 T_A =-40~+125℃	-3	-	3	%

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
		FSEL==2'b10 T_A =-40~+125℃	-4	-	4	%

表 3-11 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=1.8~5.5V T_A =25℃	32.4	32.8	33.2	KHz
I_{DD_RCLP}	RCLP 功耗	VDD=1.8~5.5V T_A =25℃		350		nA
t_{START}	RCLP 启动时间	VDD=1.8~5.5V T_A =25℃		380		us

表 3-12 内部 RC 振荡器参数

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 =-40~125℃	32	-	72	MHz
I_{DD_PLL}	PLL 功耗	输入频率 1Mhz，输出频率 32Mhz	-	400	-	uA
		输入频率 1Mhz，输出频率 72Mhz	-	450	-	
t_{LOCK}	PLL 锁定时间		-	65	95	us

表 3-13 PLL 参数

3.4.9 ADC 特性

3.4.9.1 参数说明

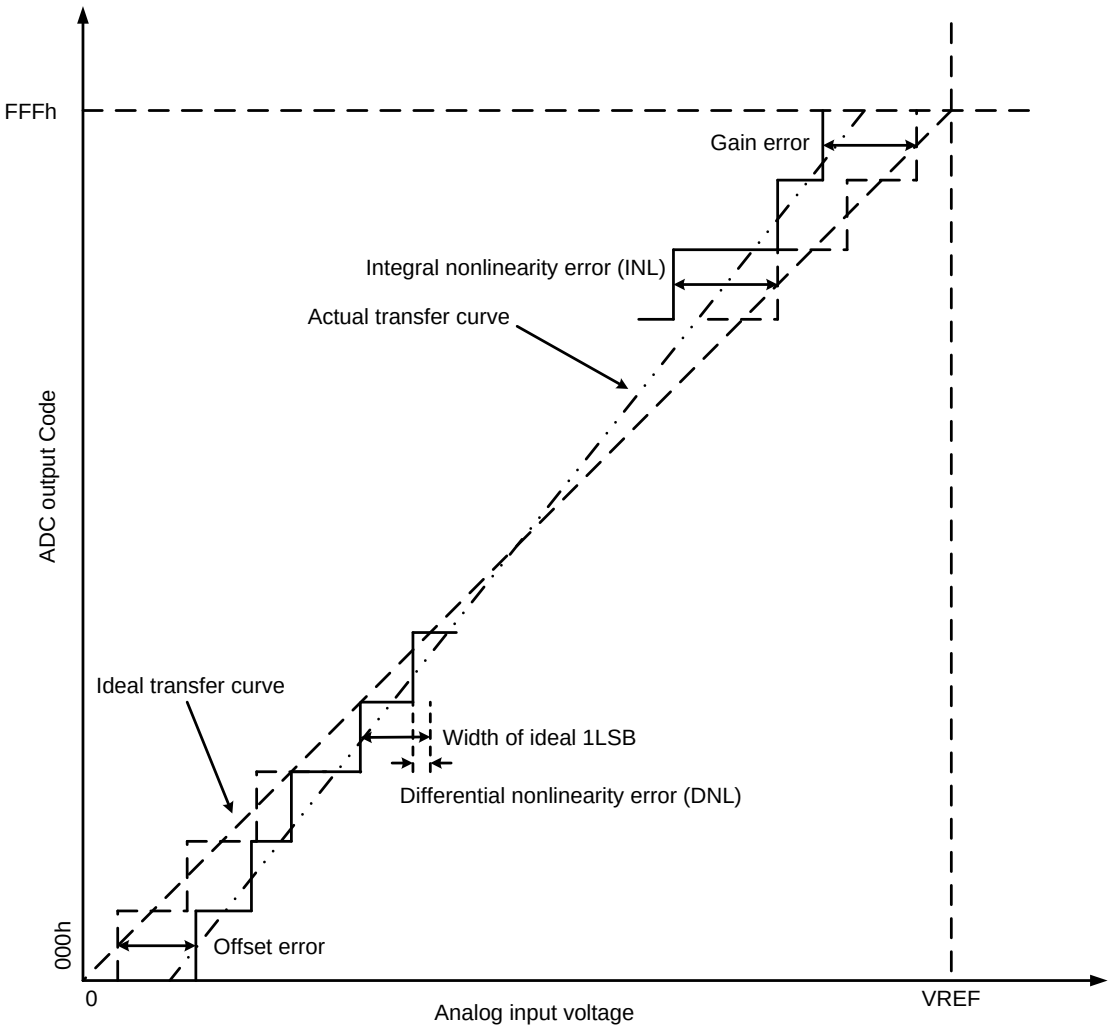


图 3-2 ADC 参数说明

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	-		1.8		5.5	V
VREF+	正参考电压	-		1.8		VDDA	V
V_{AIN}	输入电压范围	-		VREF-		VREF+	V
C_s	采样保持电容	-			3		pF
F_{CLK}	ADC 工作时钟频率	-				32	MHz
F_S	ADC 采样频率	VDDA=2.5~5.5V				2	Msps
		VDDA=1.8~2.5V				1	
T_{SAMP}	采样保持时间	-		2		512	F_{CLK}
T_{CONV}	转换时间	-			14		F_{CLK}
IDD	ADC 工作电流	VDDA=3.3V	Fs=2Msps	-	260	-	uA
		VDDA=5V	Fs=2Msps	-	450	-	uA
ADC 动态性能[1]							
ENOB	VDDA=5V VREF+=VREFP F_S=2Msps T_A=25℃ ADCCLK=XTHF	单端模式 F_{AIN}=29KHz		-	10.3	-	bits
	VDDA=3.3V VREF+=VREFP F_S=2Msps T_A=25℃ ADCCLK=XTHF	单端模式 F_{AIN}=29KHz		-	10.2	-	bits
ADC 静态性能							
DNL	差分非线性 VDDA=5V Fs=2Msps			-1	-	1	LSB
INL	积分非线性 VDDA=5V Fs=2Msps			-1.5	-	1.5	LSB
OffsetError	失调误差 校准后	单端模式			-0.5		LSB
GainError	增益误差 校准后	单端模式			-0.1		%

表 3-4ADC 参数

注：

[1]基于特征参数提取

[2]为了减少 IO 噪声对 ADC 性能的影响，推荐使用 VREFP 和 VREFN 引脚作为 ADC 的参考电压

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 3pF
- 阻抗参数参见后续表格

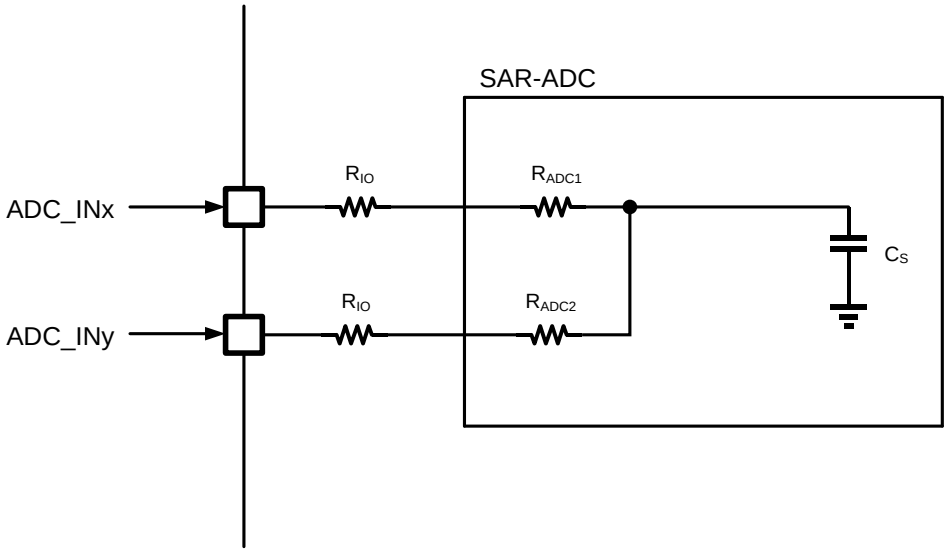


图 3-3 ADC 通道输入阻抗

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	参数值	单位

			最小值	典型值	最大值	
外部通道，单端输入						
R_{ADC}	5V	25C	-	1150	1200	Ω
		125C	-	1250	1300	
		-40C	-	1125	1150	
	3.3V	25C	-	1250	1300	
		125C	-	1350	1450	
		-40C	-	1200	1240	
	2.5V	25C	-	1350	1450	
		125C	-	1500	1650	
		-40C	-	1250	1350	

表 3-5ADC 输入阻抗

3.4.10 DAC 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	工作电压范围	-	2.2	-	5.5	V
VREF	参考电压范围	-	2.2	-	VDDA	V
IDD_{DAC}	VDDA=VREF=3.3V 无负载	Middle code (0x1FF)		23		uA
		Max code (0x3FF)		8		
	VDDA=VREF=5V 无负载	Middle code (0x1FF)		100		uA
		Max code (0x3FF)		36		
DNL	差分非线性[1]		-0.5	-	1	LSB
INL	积分非线性[1]		-0.5	-	0.5	LSB
OffsetError	失调误差[1]			-0.5		LSB
GainError	增益误差[1]			-0.3		%
t_{setup}	满幅输出建立时间	Code=0x3FF		3		us

表 3-6 DAC 参数

[1]基于特征参数提取

3.4.11 温度传感器

芯片出厂时经过温度定标，定标条件是 VDD=3.0V，T_A=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。详细使用方法请参考复旦微电子提供的库函数。

符号	参数说明	测试条件
TS_CAL1	温度传感器标定值 1	VDD=3.0V，T_A=30+/-1°C

注：根据 TS_CAL1 的数值，可以计算温度定标时温度传感器在 30° C 下的输出电压绝对值。

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	

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[1]	VDDA=VREF+=5V		2.18		LSB/°C
		VDDA=VREF+=3V		3.66		
Slope	输出斜率[1]	T_A=-40~+125° C VDDA=1.8~5.5V		2.68		mV/°C
Linerity	全温区线性度[1]		-	+/-2	+/-5	°C
I_{DDA}	温度传感器功耗（不含ADC）[2]	VDDA=3.3V		0.8		uA
t_{START}	温度传感器启动时间，包含输出 buffer 建立时间[2]	Bandgap 已经使能，置位 PTAT_EN 寄存器、VPTATBUFFER_OUTEN、VPTATBUFFER_EN			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 3-7 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

温度传感器输出曲线示意图如下。

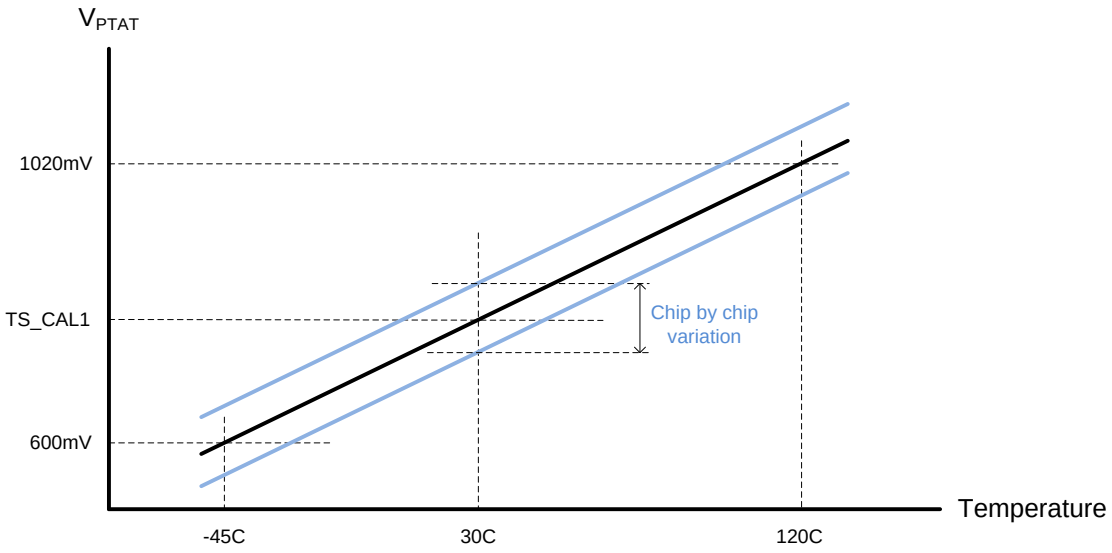


图 3-4 温度传感器输出曲线

温度传感器输出电压只和芯片基底温度有关，而与芯片当前工作电源电压无关。

3.4.12 OPA 特性

除非特别声明，以下参数基于特征参数提取。

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	

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			2.5	-	5.5	V
Vos	输入 offset 电压	校准前 VDDA=2.5~5.5V VCM=VDDA/2 T_A=-40~125C		-	+/-3	+/-8	mV
Ibias	输入偏置电流	VDDA=5V VCM=VDDA/2 T_A=25C		-	200	1000	pA
Ios	输入失调电流			-	50	100	pA
TRIMSTEP_P	低共模输入电压 offset trim 步长（0.1xVDDA）	VDDA=5V VCM=VDDA/2 T_A=25C			200		uV
TRIMSTEP_N	高共模输入电压 offset trim 步长（0.9xVDDA）				200		
I_{LOAD}	输出饱和驱动电流	R_L=10KΩ	VDDA=2.5V	-	250	-	uA
			VDDA=3.3V	-	330	-	
			VDDA=5V	-	500	-	
C_{LOAD}	容性负载					20	pF
GBW	-3dB 带宽	VDDA=5V			12		MHz
SR	压摆率	VDDA=2.5V			9.5		V/us
		VDDA=3.3V			10.5		
		VDDA=5V			15		
t_{START}	启动时间	VDDA=5V			1.2	5	us
I_{DDA}	功耗	VDDA=5V VCM=VDDA/2			1.2		mA

表 3-8 OPA 参数

[1] 基于电路设计仿真

3.4.13 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	-		1.8	-	5.5	V
VI_{comp1}	比较器输入电压范围	-		0	-	VDDA	V
Ibias	输入偏置电流	-			200	1000	pA
I_{comp12}	比较器工作电流	VDDA=5V 50Khz 方波， ±100mV overdrive	低功耗模式	-	3	-	uA
			高速模式	-	15	-	
$T_{propagation12}$	比较器传播延迟	VDDA=5V 200mV step 100mV overdrive	低功耗模式	-	0.8	-	us
			高速模式	-	200	-	ns

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$T_{setup12}$	比较器建立时间	VDDA=5V	低功耗模式	-	4	-	us
			高速模式	-	1	-	

表 3-19 模拟比较器参数

3.4.14 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	
	Flash size			-	64K	bytes
T_{PROG}	Word Program Time			-	25	μs
T_{ERASE}	Sector/Block Erase		-	2	-	ms
	Chip Erase		-	8	-	ms
N_{ED}	Sector Endurance		100,000			Erase/Write cycles
T_{DR}	Data Retention	T_A =85℃ After 100K cycling	15			yrs

表 3-20 Flash 参数

3.4.15 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
I_{IL}	输入低漏电	V_{IL}=0V	-100		100	nA
I_{IH}	输入高漏电	V_{IH}=5V	-100		100	nA
V_{OL}	输出低电平	VDD=3V Isink=5mA		0.2		V
		V_{DD}=5V I_{SINK}=10mA		0.3		
V_{OH}	输出高电平	VDD=3V Isource=5mA		2.7		V
		V_{DD}=5V I_{SOURCE} =10mA		4.5		
R_{PU}	弱上拉电阻			100		KΩ

表 3-21 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

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_{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			10		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=5V		100		ns
		VDD=3V				ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.8~5.5V -40℃≤T_A≤125℃	50		100	us

表 3-22 NRST 引脚参数

注：

[1] 此项参数基于特征参数提取

3.4.16 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				
V_{LCD}	LCD 偏置电压	-	0.547×VDD		VDD	V

表 3-23 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

FM33LF0A 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

系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

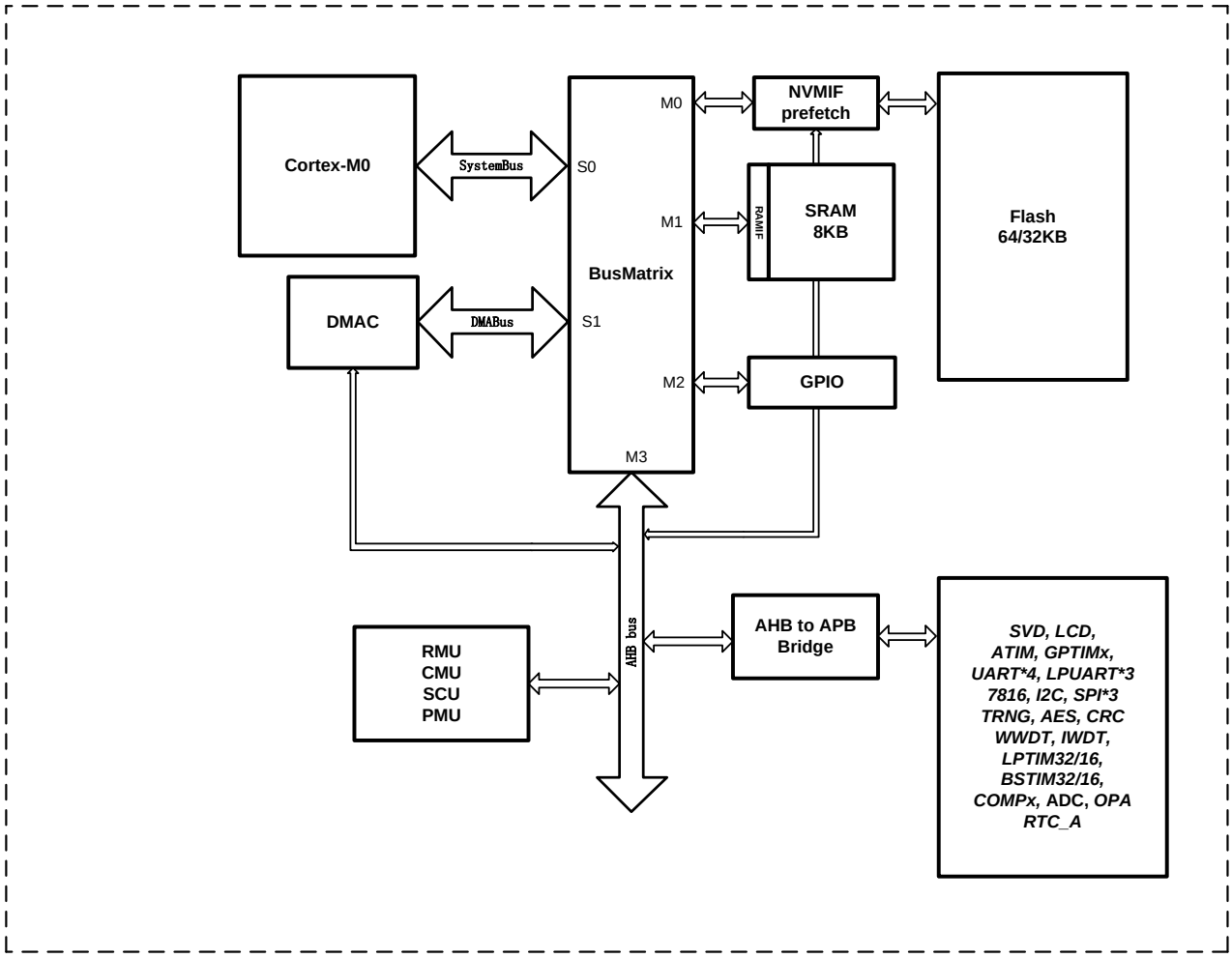


图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

Flash 页（Page）大小为 512 字节，扇区（Sector）大小为 2K 字节，每 16 个页组成一个 8K 字节的块（Block）。

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

当芯片从 Flash 启动时，FM33LF0A 地址空间分配如下图（64KB Flash，8KB RAM）：

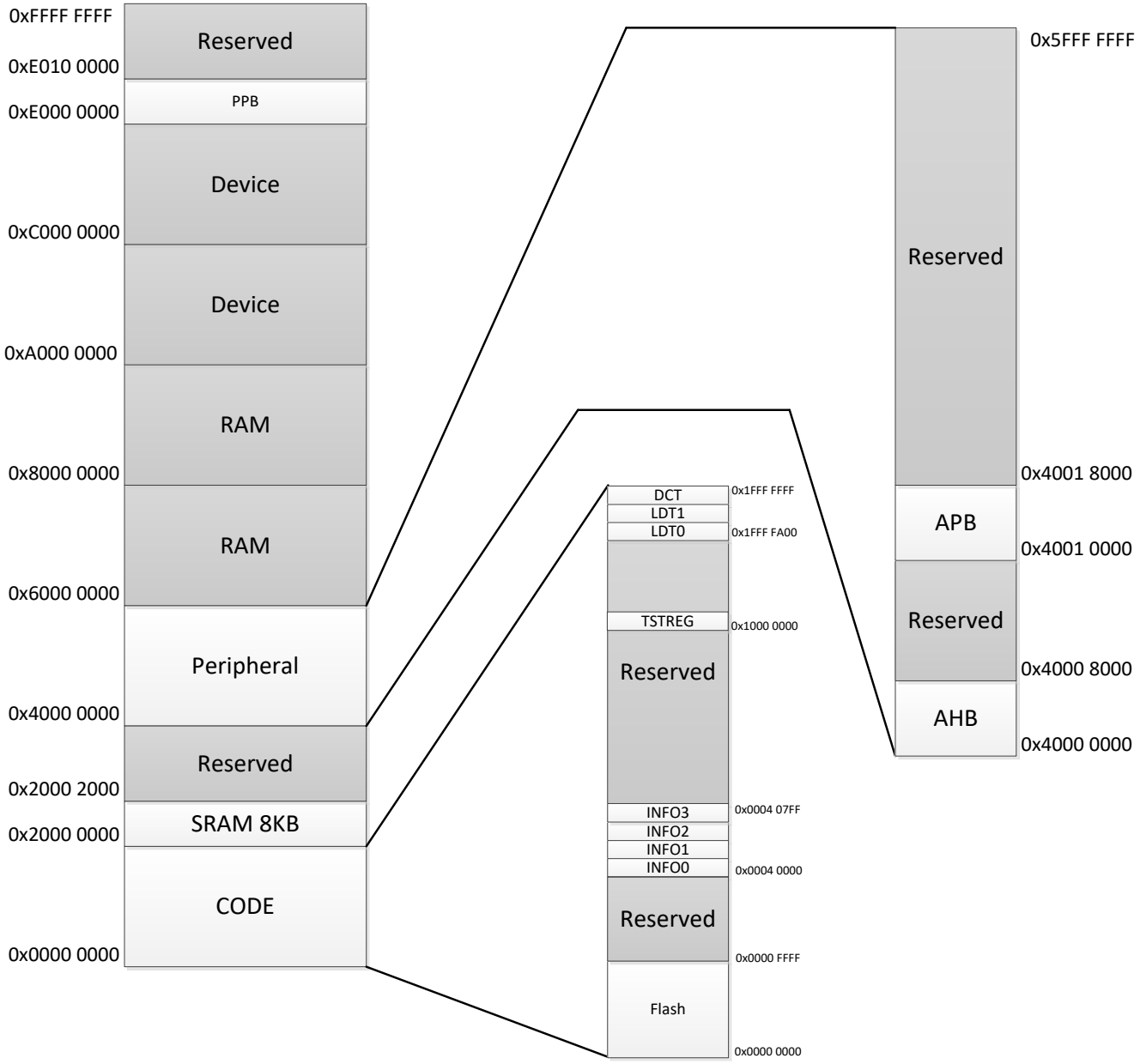


图 4-2 FM33LF01xA 总线地址

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	64KB	Flash main array
	0x1FFF_F000	4KB	Flash Option cell array
	0x2000_0000	8KB	SRAM
	0x2000_8000	-	-
	0x4000_0000	1KB	SCU
	0x4000_0400	1KB	DMA
	0x4000_0800	-	-
	0x4000_0C00	1KB	GPIO
	0x4000_1000	1KB	NVMIF
	0x4000_1400	-	-
	0x4000_1800	-	-
	0x4000_1C00	-	-
	0x4000_2000	1KB	PMU
	0x4000_2400	1KB	CMU
	0x4000_2800	1KB	RMU
	0x4000_2C00	1KB	SFU
	0x4000_FC00	1KB	RAMBIST
	0x4001_0000	64KB	APB
APB	0x4001_0000	1KB	CRC
	0x4001_0400	1KB	SPI0
	0x4001_0800	1KB	SPI1
	0x4001_0C00	1KB	LCD
	0x4001_1000	1KB	RTC
	0x4001_1400	1KB	IWDT
	0x4001_1800	1KB	WWDT
	0x4001_1C00	1KB	I2CSMB0
	0x4001_2000	1KB	UART0
	0x4001_2400	1KB	I2CSMB1
	0x4001_2800	1KB	SVD
	0x4001_2C00	1KB	-
	0x4001_3000	1KB	TAU
	0x4001_3400	1KB	-
	0x4001_3800	1KB	AES
	0x4001_3C00	1KB	TRNG
	0x4001_4000	1KB	LPUART0
	0x4001_4400	1KB	LPUART1
	0x4001_4800	1KB	-
	0x4001_4C00	1KB	GPTIM0
	0x4001_5000	1KB	-
	0x4001_5400	1KB	COMPx
	0x4001_5800	1KB	OPA
	0x4001_5C00	1KB	ADC
	0x4001_6000	1KB	DAC
	0x4001_6400	1KB	GPTIM1
	0x4001_6800	1KB	UART1
	0x4001_6C00	1KB	PGL
	0x4001_7000	1KB	UART2

	0x4001_7400	1KB	UART4
	0x4001_7800	1KB	-
	0x4001_7C00	1KB	UARTIR
	0x4001_8000	1KB	ATIM
	0x4001_8400	1KB	-
	0x4001_8800	1KB	LPTIM16
	0x4001_8C00	1KB	BSTIM16
	0x4001_9000	1KB	ANTEST(BUF4TST)
	0x4001_9400	1KB	-
	0x4001_9800	1KB	-
	0x4001_9C00	1KB	HDIV
	0x4001_A000	1KB	SENT0
	0x4001_A400	1KB	FSCAN
	0x4001_A800	1KB	SENT1
	0x4001_AC00	1KB	LED
	0x4001_B000	1KB	CLM0
	0x4001_B400	1KB	CLM1
	0x4001_B800	1KB	-
	0x4001_BC00	1KB	-

表 4-1 外设模块总线地址列表

4.3 RAM

4.3.1 概述

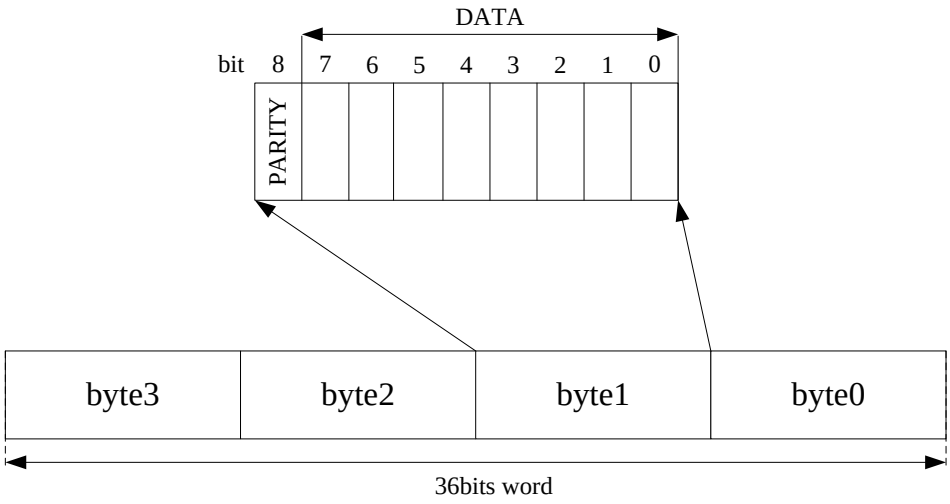
SRAM 地址空间范围是 0x2000_0000~0x2000_1FFF，软件可以对 SRAM 进行字节、半字、字访问，CPU 和 DMA 都可以以最大系统频率对 SRAM 实现无等待的单周期读写。CPU 也可以从 SRAM 取指执行程序，因此在对程序效率要求高的场合，可以将部分代码导入 SRAM 中，实现最高频率下无等待的执行。

4.3.2 奇偶校验

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

FM33LF0A 使用的 Flash 容量为 8K*64，即 64KB，支持 ECC 1bit 纠错 2bit 检错功能(SEC-DEC)，阵列组织格式包含 page(512B)、sector(2KB)、mat(64KB)

main array 共包含 128 个扇区，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。

4.4.2 特殊信息扇区说明

Flash 的 main array 以外还有几个特殊扇区可供用户使用，说明如下：

区域	说明	用途
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；软件可以读写

4.4.2.1 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

执行 flash 全擦后，才能擦除 LDT1

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

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：上电默认关闭 IWDT 其他：上电默认使能 IWDT，软件可以开启	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，用户软件或者 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 *注意：flash 最大容量 64KB，所有只有低 16bit 有效*	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]

注意制造商模式下，ACLOCKEN 和 DBRDPEN 不起作用，不会保护 Flash 内容。

4.4.2.2 IF0~3 page

这4个IF扇区开放给用户作为数据区使用，SWD和软件均可以读写。

总线地址是0x0004_0000~0x0004_07FF，低地址为IF0，高地址为IF3，总共2KB。

注意此区域仅支持page编程，不支持sector编程。

4.4.3 ECC 中断

Flash ECC中断：

- ECC detection 中断：检测到 2bit 错误，无法纠错，产生 ECCDD 中断，汇入 FLASH 中断，标志寄存器在 FLS_ECCR 中。
- ECC correction 中断：纠正 1bit 错误，不影响正常运行，此时根据产生 ECC 错误产生的位置会分别产生 ECCCD、ECCBE 中断标志，汇入 FLASH 中断，标志寄存器在 FLS_ECCR 中。
- 错误产生时，FLS_ECCR.ADDR_ECC 寄存器记录纠错或检错发生的地址

推荐只开启ECCDD中断，对于纠错发生之后不影响Flash正常的读取。

4.4.4 Flash 编程

4.4.4.1 概述

FM33LF0A 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

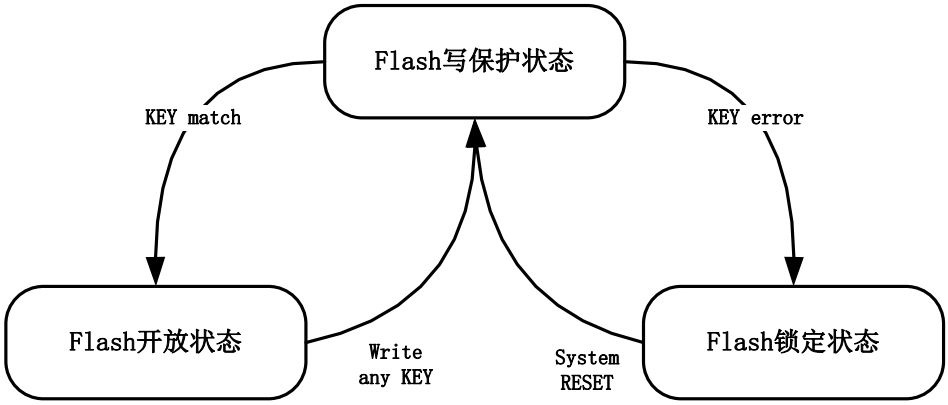
4.4.4.2 Flash 擦写时钟

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

FM33LF0A 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 FLSIF.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。SWD 可以在制造商或用户模式下启动全擦，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 main array 全部擦除，任何对 main array 的编程将清除此标志）
- 在全擦标志有效的情况下，SWD 可以任意擦写 NVR0，否则擦写 NVR0 被禁止并触发错误中断
- 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）

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 00

- 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 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 FlashKEY 寄存器写任意值恢复写保护

注意：1、当main array中某个sector被冗余修复时，对此sector的擦写实际被映射到冗余扇区
2、扇区擦操作只对main array有效

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
- 向需要擦除的扇区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 FlashKEY 寄存器写任意值恢复写保护

注意：当main array中某个page处于被修复的扇区中时，对此page的擦写将被映射到冗余扇区中的对应页

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，每次操作只能执行 word 编程，操作流程如下：

- 清除 EREQ 寄存器，置位 PREQ 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
- Byte 写入直接完成，half-word 写入则由 NVMIF 自动执行 2 次 byte program，word 编程由 NVMIF

执行 4 次 byte program

- 编程完成后置位中断标志
- 软件确认编程结束后向 FlashKEY 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 half-sector（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-sector 对齐的，也就是 Flash 地址低 6 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
- 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 每收到 1 个 word 自动完成 4 次 byte program
- 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 FlashKEY 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 扇区内的 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 main array，同时也无法通过 SWD 接口对 RAM 进行访问（防止 debugger 通过 DMA 等手段将 Flash 中数据导入 RAM 后读取）。

退出 DBRDP 的方法：通过 SWD 对 flash 进行全空间擦写（mat erase 或者伪全擦），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 8 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 LDT0 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（mat erase 或伪全擦），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	11	-	R/E/W/F
		01/10	-	对应 Block 只能取指
		00	-	对应 Block 只能取指
	OFF	11	R/E/W	R/E/W/F
		01/10	-	对应 Block 只能取指
		00	对应 Block 无法访问	对应 Block 只能取指
LDT0	x	x	R	R
LDT1	ON	x	R[2]	R
	OFF	x	R/E/W	R
IF0,1,2,3	x	x	R/E/W	R/E/W

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

用户配置寄存器起始地址：0x40000000

offset 地址	名称	符号
0x04	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x10	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1

FLASH 控制模块起始地址：0x40001000

offset 地址	名称	符号
0x00	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x04	预取指控制寄存器 （Flash Prefetch Control Register）	FLS_PFCR
0x08	--	
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
0x20	-	
0x28	Flash ECC 控制寄存器	FLS_ECCCR
0x2C	Flash ECC 配置寄存器	FLS_ECCR

4.5.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
地址	0x4000_1000							
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	-					WAIT		
位权限	U-0					R/W-000		

位号	助记符	功能描述

位号	助记符	功能描述
31:3	-	RFU：未实现，读为 0
2:0	WAIT	Flash 读等待周期配置 (Wait Cycles) 000：0 wait cycle 001：1 wait cycle 010：2 wait cycles 011：3 wait cycles 100：4 wait cycles 101：5 wait cycles 110：6 wait cycles 111：7 wait cycles CPU 主频小于等于 24MHz 时，不需要开启 wait；主频大于 24M 小于 48Mhz 时使能 1 wait，主频大于 48Mhz 时使能 2 wait，主频大于 72Mhz 时使能 3 wait

4.5.2 预取指控制寄存器（FLS_PFCR）

名称	FLS_PFCR							
地址	0x4000_1004							
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	-							PRFTEN
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	PRFTEN	指令预取指使能，在 WAIT==00 的情况下写 1 无效 (Prefetch Enable) 1：使能 Prefetch 0：禁止 Prefetch

4.5.3 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x4000_0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	-						
位权限	R-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	-				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
30:4	-	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
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

4.5.4 ACLOCK 寄存器 1（FLS_ACLOCK1）

名称	FLS_ACLOCK1							
offset	0x4000_0010							
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	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
31:16	-	RFU：未实现，读为 0
15:0	LOCK1	ACLOCK 配置寄存器低 16bit，分别用于控制 Block7~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 所有 bit 软件只能写 0，不能写 1。

4.5.5 Flash 擦写控制寄存器（FLS_EPCR）

名称	FLS_EPCR							
offset	0x4000_1014							
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
0	EREQ	Erase Request 软件置位，硬件完成擦除后自动清零

4.5.6 Flash Key 输入寄存器（FLS_KEY）

名称	FLS_KEY							
offset	0x4000_1018							
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

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)

4.5.7 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)

4.5.8 Flash 标志寄存器（FLS_ISR）

名称	FLS_ISR							
offset	0x4000_1020							
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			BTSF
位权限	U-0				R-000			R-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	BTSF	BootSwap 标志寄存器 (BootSwap Flag) 0：启动程序区为 Flash 物理地址 0000H~1FFFH 1：启动程序区为 Flash 物理地址 2000H~3FFFH
15:12	-	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，软件写 1 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

4.5.9 Flash ECC 控制寄存器（FLS_ECCCR）

名称	FLS_ECCCR							
地址	0x4000_1028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ECCR							
位权限	R-00000001							

位	位名	功能描述
31:8	RFU	未实现：读为 0
7:0	ECCR	ECC 控制寄存器 软件写 0xCC 关闭 ECC 功能，默认 ECC 有效

4.5.10 Flash ECC 配置寄存器（FLS_ECCR）

名称	FLS_ECCR							
地址	0x4000_1028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ECCDD	ECCCD	RFU		ECCBE	RFU	ECCEIE	ECCCIE
位权限	W1C-0	W1C-0	U-00		R-0	U-0	RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				RFU			
位权限	U-0000				U-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ADDR_ECC							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADDR_ECC							
位权限	R-00000000							

位	位名	功能描述
31	ECCDD	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	ECCCD	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

位	位名	功能描述
29:28	RFU	未实现：读为 0
27	ECCBE	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 也会产生纠错标志，但 32bit 数据本身仍然是正确的可以使用
26	RFU	未实现：读为 0
25	ECCEIE	ECC 错误中断使能 1：允许 ECCDD 产生中断产生中断 0：禁止错误中断产生
24	ECCCIE	ECC 纠错中断使能 1：允许 ECCCD 和 ECCBE 产生中断 0：禁止纠错中断
23:16	RFU	未实现：读为 0
15:0	ADDR_ECC	ECC 错误地址(byte 地址)，只读

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 BOR_CFG 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 PDR_CFG 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 POR、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 2.5~5.5V，所有模拟模块在这个电压范围内都可以保证正常工作。

- VCAP

VCAP 是内部 LDO 输出，系统上应尽量靠近 VCAP 引脚对地布置 0.1~1uF 退耦电容。

- VREFP

部分封装形式中有独立的 VREFP 引脚，VREFP 是 ADC 的基准电压输入，在 ADC 工作时，会从 VREFP 引脚抽取几十到上百 uA 的电流。部分封装中，VREFP 都是和 VDDA 封装在一起的。

5.1.2 电源结构图

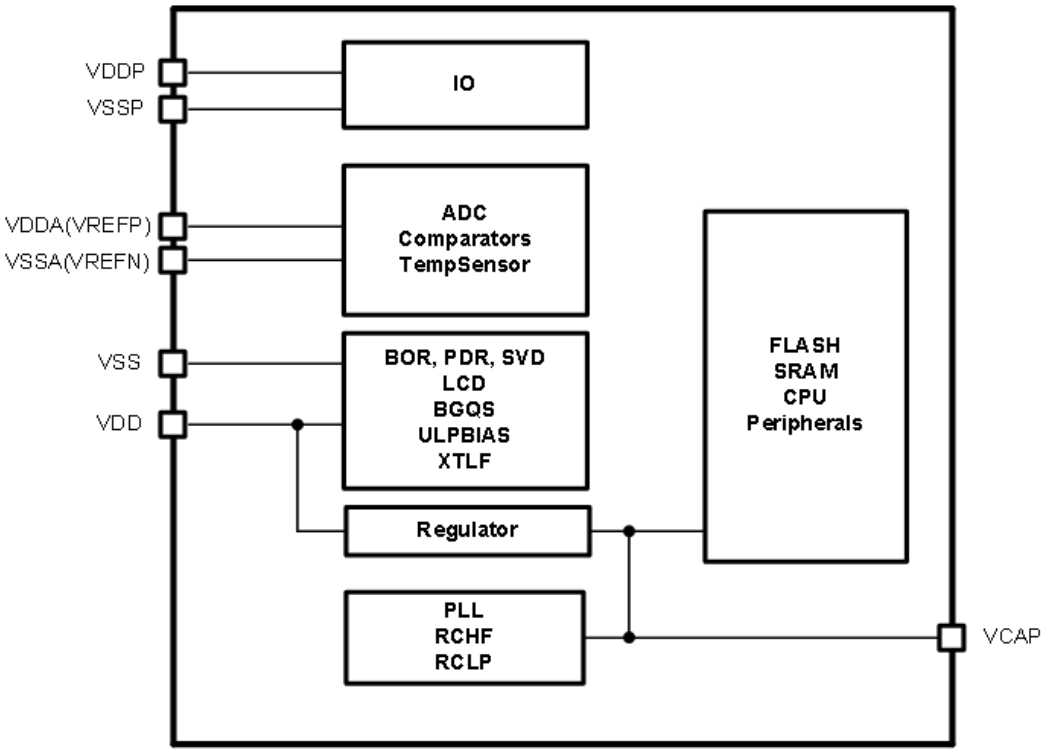


图 5-1 芯片电源结构图

5.1.3 ADC 和基准电压

ADC 使用的电源地为VDD(VDDA)和VSS(VSSA)，其参考电压是VREFP。在部分封装形式中，VDDA 和 VREFP 会被单独引出到芯片管脚，其中 VREFP 可以由系统提供，可以与 VDD 不同，也可以直接在 PCB 上与 VDD 短接在一起。为了提高 ADC 转换精度，降低电源噪声的影响，系统上建议对 VDD 电源单独滤波，还可以在 PCB 上对 VDD 和 VSS 走线进行 shielding，以尽可能屏蔽系统噪声。

VREFP 与 VDD 独立时，其输入基准可以与 VDD 不同，允许的输入范围是：

1.6V ≤ VREFP ≤ VDD(VDDA)

FM33LF0xxA 的部分封装形式中，ADC 基准源 VREFP 和 VREFN 有独立的封装引脚。

在某些低管脚数的封装中，可能没有独立的 VREFP 和 VREFN，在这种情况下，VREFP 和 VDD 在封装内部相连，VREFN 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。

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
- 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	150uA/MHz		正常工作	-
SLEEP	3.3uA	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	CPU 休眠 关闭 RCHF、PLL、ADC 等 保持 BGQS 开启，关闭 LDO15，RTC 走时 VREF1p2 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	7us
DEEPSLEEP	1.3uA		CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 关闭 BGQS，关闭 LDO15，RTC 走时 VREF1p22 由软件配置决定是否开启，开启的话增加 1.5uA 功耗	7us

注：[1] 典型唤醒时间指从唤醒信号到来，到 CPU 开始执行唤醒中断服务程序的时间间隔。

[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。

不同功耗模式之间的转换如下图所示：

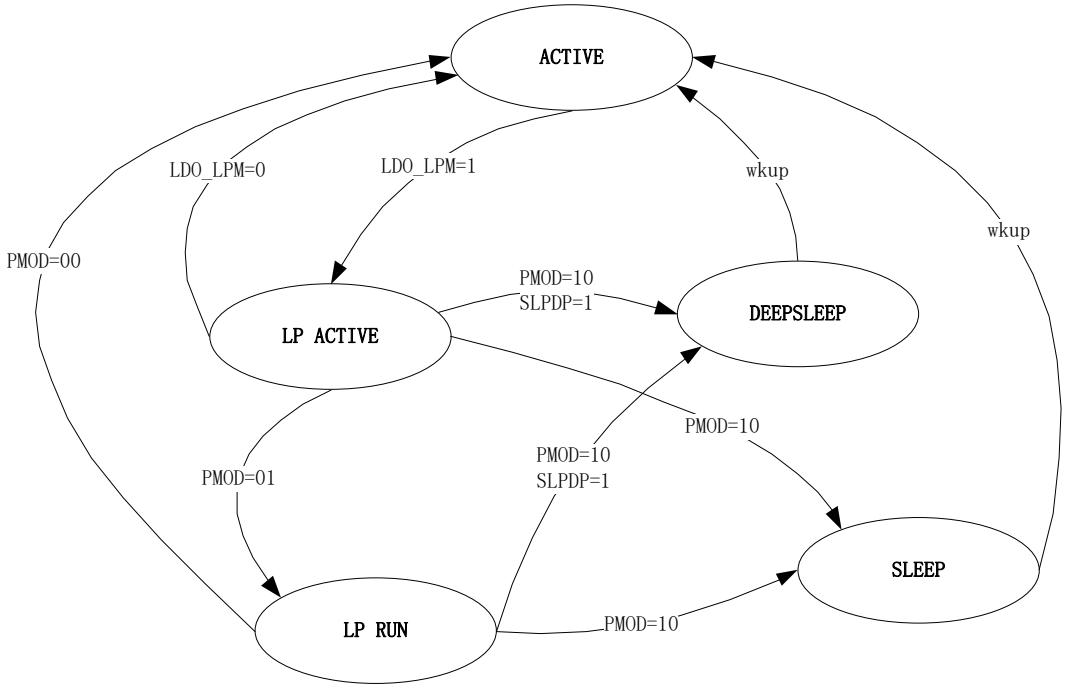


图 5-2 功耗模式与系统主频

5.2.2 功耗模式与系统频率

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	
	>48MHz, <=72MHz		2	

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 72MHz。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，建议软件开启 Flash Prefetch 来提升指令执行效率。

5.2.4 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 或 WFE 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（固定为 8MHz）
- 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

5.2.5 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：

- 配置 PMOD 寄存器为 10
- 执行 WFI 或 WFE 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；SVD、比较器仍可以在 DEEPSLEEP 模式下工作。

注意：如果 Deepsleep 下使能 SVD

- SVD 配置为低压报警常使能，则 AVREF 保持使能
- SVD 配置为低压报警间歇使能，则 AVREF 间歇启动

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，32786Hz 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	√	√
比较器	可屏蔽，用于外部事件唤醒	√	√
RTC_A	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	√	√
LPUART	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	√	√
LPTIM32	可屏蔽，用于定时唤醒	√	√
BSTIM32	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	√	√

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.WKPFLAG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.WKFSEL）。最快情况下芯片唤醒后将以24MHz时钟启动。

休眠时AHBPRES寄存器不会复位，但是SYSCLKSEL寄存器将复位成00（选择RCHF）。因此，如果休眠前系统时钟不是RCHF，则唤醒后将默认使用RCHF 不分频工作，是否经过分频由休眠前的AHBPRES寄存器决定。

5.5 寄存器

模块起始地址：0x40002000

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 0 （Wakeup Source Flags Register0）	PMU_WKFR0
0x0C	唤醒源标志查询寄存器 1 （Wakeup Source Flags Register1）	PMU_WKFR1
0x3C	模拟 BUFFER 控制寄存器 （PMU Analog Buffer Control Register）	PMU_BUFCR
0x50	PTAT 控制寄存器 （PMU PTAT Control Register）	PMU_PTATCR

5.5.1 低功耗控制寄存器（PMU_CR）

名称	PMU_CR							
offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	REV1	-						
位权限	R/W-0	U-0						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				-			
位权限	U-0				U-0			
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-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				REV		PMOD	
位权限	U-0				R/W-00		R/W-00	

Bit	助记符	功能描述
31	REV1	预留寄存器，可读写，无功能
30:12	--	未实现，读为 0
11:10	WKFSEL	SLEEP/DEEPSLEEP 唤醒频率 00/11：RCHF-8MHz 01：RCHF-16MHz 10: RCHF-24MHz
9	SLPDP	DEEPSLEEP 控制寄存器 1：芯片进入休眠模式时，进入深度休眠模式（DEEPSLEEP） 0：芯片进入休眠模式时，进入普通休眠模式（SLEEP） 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	--	RFU:未实现，读为 0
3:2	REV	预留寄存器，可读写，无功能

Bit	助记符	功能描述
1:0	PMOD	低功耗模式配置寄存器 00: ACTIVE mode 10: 休眠模式 注 1：本寄存器无法写入 01/11 注 2：从 SLEEP/DEEPSLEEP 返回 ACTIVE 时，由硬件自动改写为 00

5.5.2 唤醒时间控制寄存器（PMU_WKTR）

名称	PMU_WKTR							
offset	0x00000004							
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					U-0	R/W-01	

Bit	助记符	功能描述
31:3	--	未实现，读为 0
2	--	禁止写 1
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 00：0us 01：2us 10：4us 11：8us 推荐使用 10 或者 11

5.5.3 唤醒源标志查询寄存器（PMU_WKFR0）

名称	PMU_WKFR0							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			RTC WKF	SVD WKF	LFDET WKF	COMP3 WKF	IOWKF
位权限	U-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	-	FSCAN WKF	COMP2 WKF	COMP1 WKF	COMP0 WKF

名称	PMU_WKFR0							
位权限	U-0	R-0	R-0	U-0	R-0	R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	UART3_LINWKF	UART2_LINWKF	UART1_LINWKF	UART0_LINWKF	LPTWKF	BTWKF	RFU	DBGWKF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	UART3WKF	UART2WKF	UART1WKF	UART0WKF	-		LPU1WKF	LPU0WKF
位权限	R-0	R-0	R-0	R-0	U-00		R-0	R-0

位	位名	功能描述
31:29	--	未实现，读为 0
28	RTCWKF	RTC 中断唤醒标志，中断撤销时硬件自动清零 注：RTCA/RTCB 均可唤醒，标志均为该位
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	COMP3WKF	COMP3 中断唤醒标志，中断撤销时硬件自动清零
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零
23	RFU	RFU:未实现，读为 0
22	I2CSMB1WKF	I2CSMB1 中断唤醒标志，中断撤销时硬件自动清零
21	I2CSMB0WKF	I2CSMB0 中断唤醒标志，中断撤销时硬件自动清零
20	RFU	RFU:未实现，读为 0
19	FSCANWKF	FSCAN 中断唤醒标志，中断撤销时硬件自动清零
18	COMP2WKF	COMP2 中断唤醒标志，中断撤销时硬件自动清零
17	COMP1WKF	COMP1 中断唤醒标志，中断撤销时硬件自动清零
15	UART3_LINWKF	UART3_LIN 中断唤醒标志，中断撤销时硬件自动清零
14	UART2_LINWKF	UART2_LIN 中断唤醒标志，中断撤销时硬件自动清零
13	UART1_LINWKF	UART1_LIN 中断唤醒标志，中断撤销时硬件自动清零
12	UART0_LINWKF	UART0_LIN 中断唤醒标志，中断撤销时硬件自动清零
11	LPTWKF	LPTIMER 中断唤醒标志，中断撤销时硬件自动清零
10	BTWKF	BSTIMER 中断唤醒标志，中断撤销时硬件自动清零
9	RFU	RFU:未实现，读为 0
8	DBGWKF	DEBUG 中断唤醒标志，软件写一清零
7	UART3WKF	UART3 中断唤醒标志，中断撤销时硬件自动清零
6	UART2WKF	UART2 中断唤醒标志，中断撤销时硬件自动清零
5	UART1WKF	UART1 中断唤醒标志，中断撤销时硬件自动清零
4	UART0WKF	UART0 中断唤醒标志，中断撤销时硬件自动清零
3:2	RFU	RFU:未实现，读为 0
1	LPU1WKF	LPUART1 中断唤醒标志，中断撤销时硬件自动清零
0	LPU0WKF	LPUART0 中断唤醒标志，中断撤销时硬件自动清零

5.5.4 唤醒源标志查询寄存器（PMU_WKFR1）

名称	PMU_WKFR1							
offset	0x0000000C							
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	WKPxF							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:8	--	未实现，读为 0
7:0	WKPxF	NWKUPx Pin 唤醒标志，软件写 1 清零

5.5.5 模拟 BUFFER 控制寄存器（PMU_BUF_CR）

名称	PMU_BUF_CR							
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
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

5.5.6 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、VPTATNUF_OEN，等待输出建立时间后，启动 ADC 采样并计算温度。

6 权限管理（MDC）

6.1 寄存器

6.1.1 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	SPECIAL_OP_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	SPECIAL_OP_FLAG	特殊指令标志，写 1 清零 1：执行了特殊指令代码，如试图在 XN 区域内取指 0：无特殊指令代码被执行
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

7 处理器（CPU）

7.1 概述

FM33LF0A 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

7.1.1 处理器配置

Feature	Options	FM33LF0xxA Config
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 7-1 FM33LF0xxA CPU 配置简表

7.2 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 7-2 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。

7.3 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。

7.3.1 中断向量表

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	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	FLASH	NVMIF 中断	0x0000_004C
20	11	settable	FDET	XTLF 或 XTHF 停振检测中断	0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable	RAMP	RAM 奇偶校验错误中断	0x0000_0058
23	14	settable	SPI0	SPI 中断	0x0000_005C
24	15	settable	SPI1		0x0000_0060
25	16	settable	DAC	DAC 中断	0x0000_0064
26	17	settable	UART0	UART 中断	0x0000_0068
27	18	settable	UART1		0x0000_006C
28	19	settable	UART2		0x0000_0070
29	20	settable	UART3		0x0000_0074
30	21	settable	UART4		0x0000_0078
31	22	settable	FSCAN	FSCAN 中断	0x0000_007C
32	23	settable	LPUARTx	LPUARTx 中断	0x0000_0080
33	24	settable	I2Cx	I2Cx 中断	0x0000_0084
34	25	settable	CLM/ LUT	时钟监控中断 PGL 中断	0x0000_0088
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	SENTx	SENT0/1 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	GPT0	通用定时器 0 中断	0x0000_00A8
43	34	settable	GPT1	通用定时器 1 中断	0x0000_00AC
44	35	settable	ATIM	高级定时器中断	0x0000_00B0

Position	Priority	Priority type	Acronym	Description	Address
45	36	settable	TAU	TAU 中断	0x0000_00B4
46	37	settable	EXTI	外部引脚中断	0x0000_00B8
47	38	settable	LED/LCD	LED/LCD 中断	0x0000_00BC

表 7-3 FM33LF0xxA 中断向量表

其中WKUPx中断可以接到NMI，或者38#入口。通过PADCFG模块的WKUPEN.WKISEL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。

7.3.2 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。

7.3.3 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

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。

7.3.4 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。

7.4 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》

7.4.1 调试功能引脚

FM33LF0A 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。

7.4.2 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 MCUDBGCR 寄存器配置看门狗打开或关闭。

7.4.3 DEBUG 的复位

内核的 DEBUG 部分仅受上下电复位影响，其他系统复位源如看门狗、引脚复位、软件复位等，都不会复位 DAP 电路。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)。

8 复位管理单元（RMU）

8.1 概述

复位电路特点：

- 支持多个复位源，如上下电复位、SVD 复位、看门狗复位、软件复位、引脚复位等
- 上下电复位监控主电源供电
- 上电复位典型释放电压 1.55V
- 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

8.2 模块框图

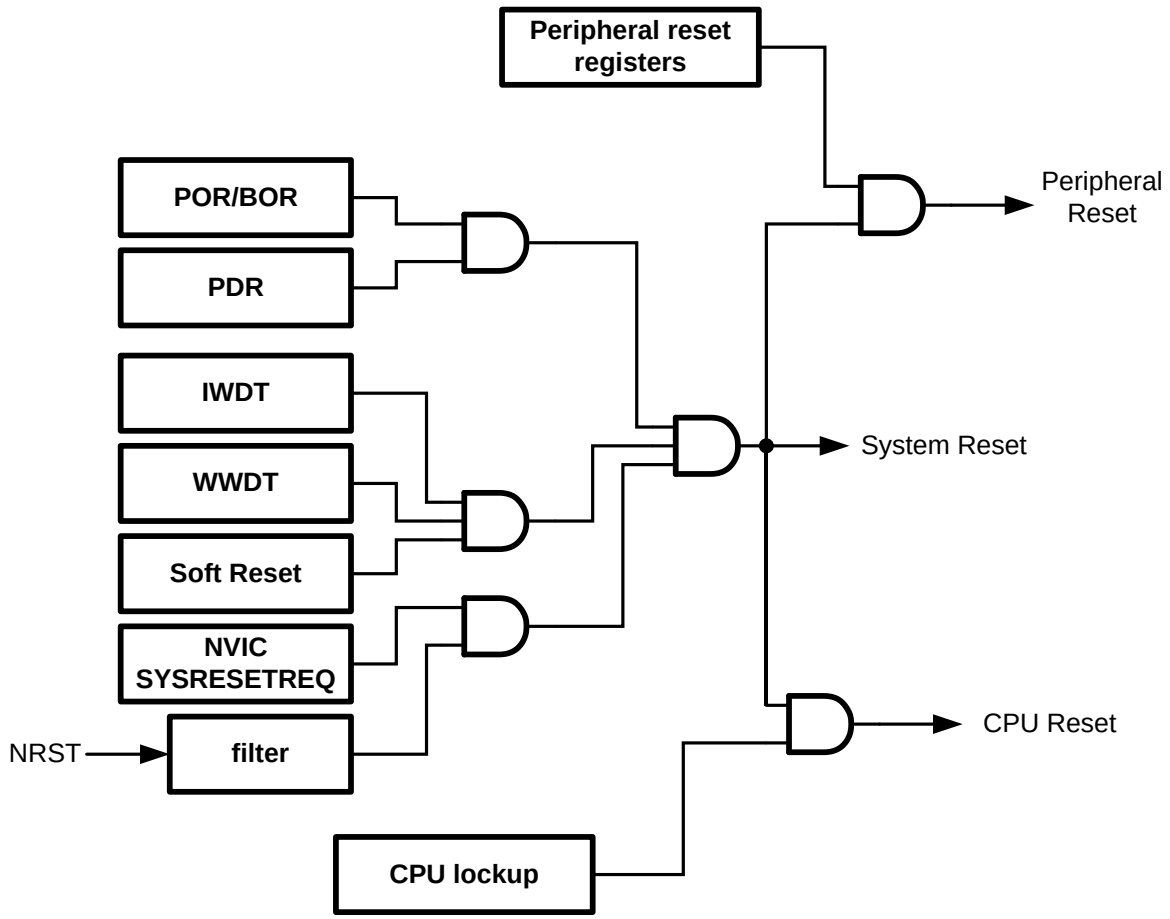


图 8-1 芯片复位源框图

8.3 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档位：1.8V、2.0V、2.2V、2.4V、2.7V、3.0V、3.6V、4V

芯片上电复位放开后，BOR复位默认是关闭的，软件可以开启。

注意：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。

8.4 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。

8.5 NRST 引脚复位

NRST 引脚可以复用为 GPIO（PC14），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。

8.6 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

8.6.1 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

8.6.2 BOR 配置寄存器（RMU_BORCR）

名称	RMU_BORCR

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	-							
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
31:4	--	RFU：未实现，读为 0
4:2	CFG	下电复位电压配置 000—1.8V 001—2.0V 010—2.2V 011—2.4V 100—2.7V 101—3.0V 110—3.6V 111—4.0V
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

8.6.3 复位配置寄存器（RMU_LKPCR）

名称	RMU_LKPCR							
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	-						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

Bit	助记符	功能描述
1	**EN**	LOCKUP 复位使能 1：使能 SC000 LOCKUP 复位 0：屏蔽 SC000 LOCKUP 复位
0	--	RFU：未实现，读为 0

8.6.4 软件复位寄存器(RMU_SOFTRST)

名称	RMU_SOFTRST							
地址	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

8.6.5 复位标志寄存器(RMU_ RSTFLAG)

名称	RMU_RSTFLAG							
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	-	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
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

Bit	助记符	功能描述
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

8.6.6 外设复位使能寄存器(RMU_PRSTEN)

名称	RMU_PRSTEN							
地址	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

8.6.7 AHB 外设复位寄存器(RMU_AHBRSTCR)

名称	RMU_AHBRSTCR							
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	-							
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

8.6.8 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	-	UART4RST	UART3R3ST	UART2RST	UART1RST	UART0RST	UARTIRRST	-
位权限	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	GPT1RST	GPT0RST	-	-	BT16RST	-	
位权限	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	SPI1RST	SPI0RST	-			LPUART1RST	LPUART0RST
位权限	U-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		PGLRST	-	I2CSMB1RST	I2CSMB0RST	LPT16RST	-
位权限	U-0		R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
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
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

Bit	助记符	功能描述
		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
24:23	--	RFU：未实现，读为 0
22	GPT1RST	GPTIM1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
21	GPT0RST	GPTIM0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20:19	--	RFU：未实现，读为 0
18	BT16RST	BSTIM16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
17:15	--	RFU：未实现，读为 0
14	SENT1RST	SENT1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
13	SENT0RST	SENT0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
12:10	--	RFU：未实现，读为 0
9	LPUART1RST	LPUART1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
8	LPUART0RST	LPUART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
7:6	--	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	--	RFU：未实现，读为 0
3	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
2	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
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	--	RFU：未实现，读为 0

8.6.9 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	LEDRST	LCDRST	CLM1RST	CLM0RST	TAU0RST	OPARST	DACRST	ADCCRST
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	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	-			DIVASRST	AESRST	CRCRST	RNGRST
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				SENT1RST	SENT0RST	-	
位权限	U-0				R/W-0	R/W-0	U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		FSCANRST	-		ATRST	SVDRST	COMPRST
位权限	U-0		R/W-0	U-0		R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	LEDRST	LED 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
30	LCDRST	LCD 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
29	CLM1RST	CLM1 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
28	CLM0RST	CLM0 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
27	TAU0RST	TAU0 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
26	OPARST	OPA 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
25	DACRST	DAC 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
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
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
22:20	--	RFU：未实现，读为 0
19	DIVASRST	DIVAS 模块复位，软件写 1 复位，写 0 撤销复位

Bit	助记符	功能描述
		0：不复位 1：复位
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
15:12	--	RFU：未实现，读为 0
11	SENT1RST	SENT1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
10	SENT0RST	SENT0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
9:6	--	RFU：未实现，读为 0
5	FSCANRST	FSCAN 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
4:3		
2	ATRST	ATIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
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

9 独立看门狗（IWDT）

9.1 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。根据 option byte 配置，独立看门狗在芯片上电后由软件启动，或者自动启动，启动后无法关闭，直到芯片发生复位。

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 MCUDBGCR 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
- TESTN=0，芯片处于 CP 状态

IWDT 核心是一个 12bit 向上计数器，复位后从 0 开始递增，计数到溢出长度后触发 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。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。

9.2 结构框图

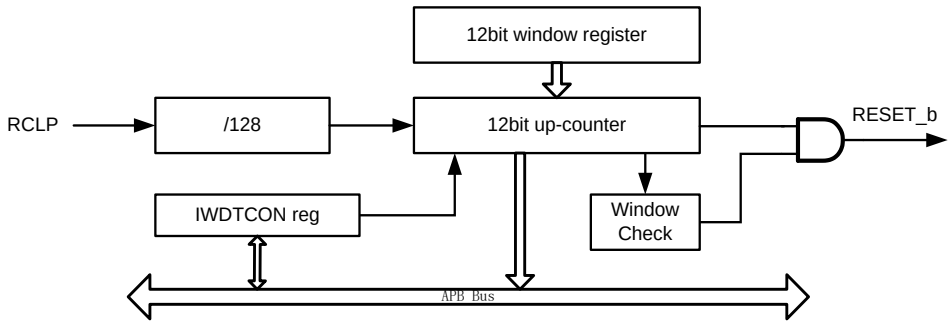


图 9-1 IWDT 结构框图

9.3 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 IWDTCFG 寄存器，选择合适的溢出周期
- 读 IWDTCFG，确保写入正确
- 溢出周期更新完毕，CPU 正常运行

IWDT 使用 LSCLK 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：

$tWWDT = T_{RCLP} * 128 * IWDTOVP$

LSCLK 频率	溢出长度配置	溢出时间（ms）
32768Hz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

表 9-1 IWDT 溢出周期表

9.4 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。

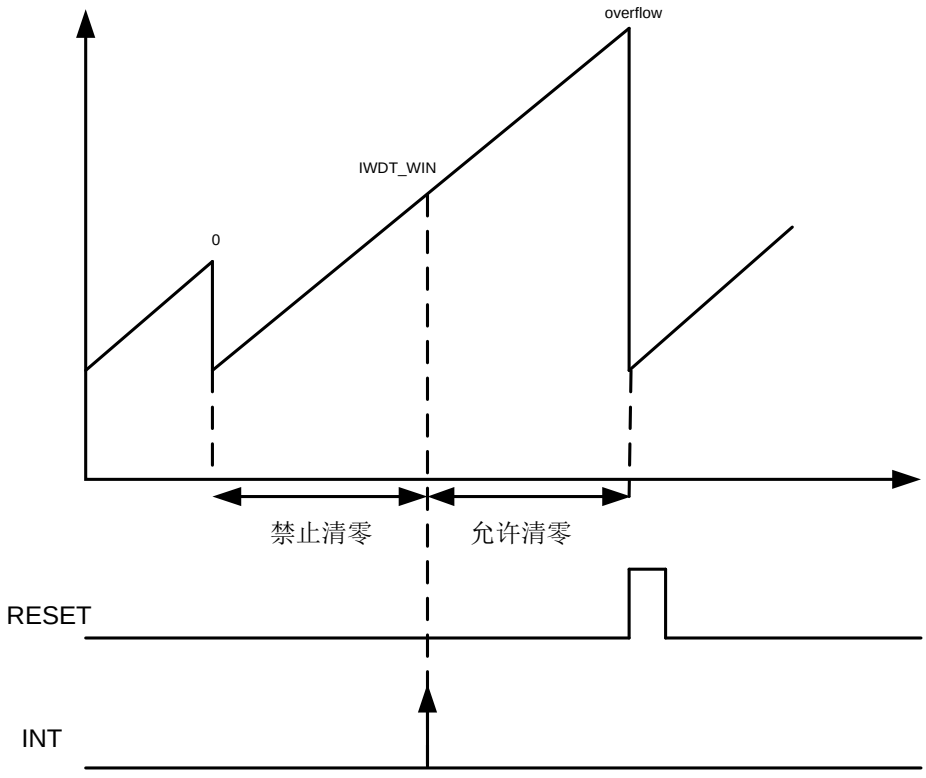


图 9-2 IWDT 窗口示意图

9.5 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）。

9.6 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

9.6.1 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**	上电复位后 IWDT 默认关闭，软件向此寄存器写入 0x1234_5A5A 后启动 IWDT，此后 IWDT 不可关闭直到下一次芯片复位。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

9.6.2 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

9.6.3 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

9.6.4 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)

9.6.5 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

9.6.6 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

10 窗口看门狗（WWDT）

10.1 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。

10.2 结构框图

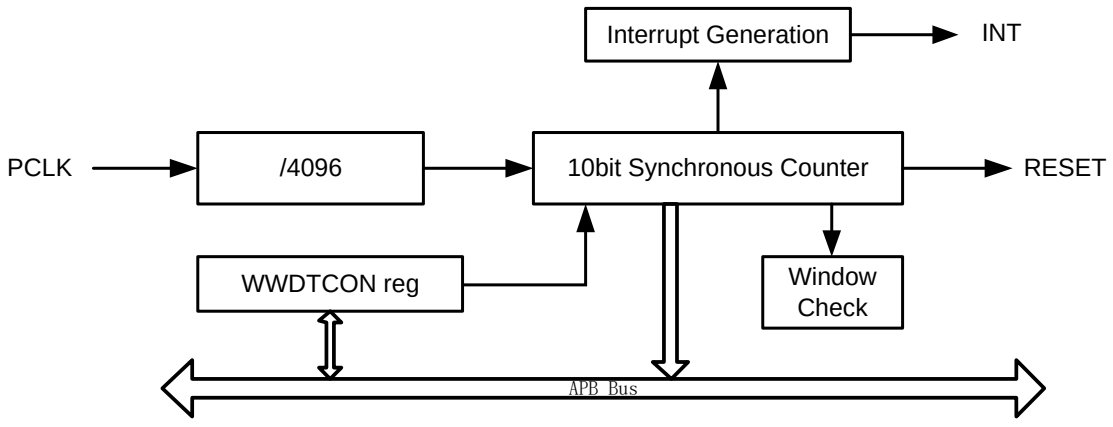


图 10-1WWDT 结构框图

10.3 WWDT 工作方式

WWDT 在芯片复位后默认关闭，软件需对 WWDTCON 寄存器写入 0x5A 来启动 WWDT。WWDT 启动后，如果软件在 open window 内对 WWDTCON 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 APBCLK 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	258.144

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

表 10-1 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。

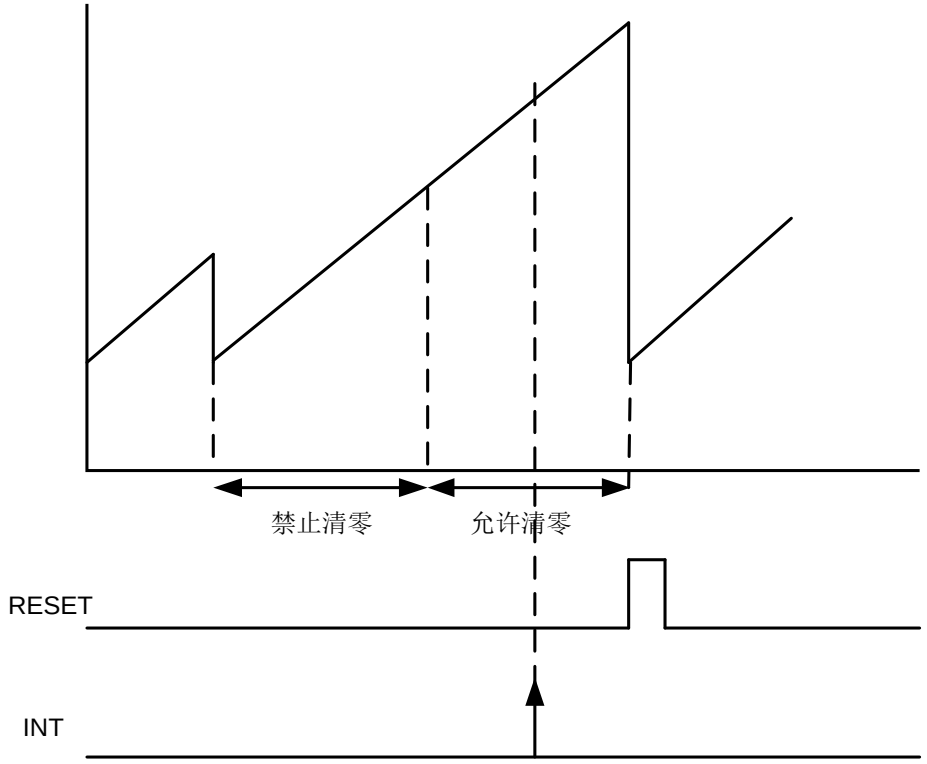


图 10-2WWDT 窗口示意图

10.4 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

10.4.1 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

10.4.2 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

位号	助记符	功能描述
		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

10.4.3 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)

10.4.4 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

位号	助记符	功能描述
0	IE	WWDT 中断使能 (WWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

10.4.5 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

10.4.6 WWDT 预分频寄存器（WWDT_PSC）

名称	WWDT_PSC							
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	-				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

位号	助记符	功能描述
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)

11 时钟管理单元（CMU）

11.1 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)、32KHz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
- CPU 和总线最高频率 72MHz

11.2 时钟架构

11.2.1 时钟树

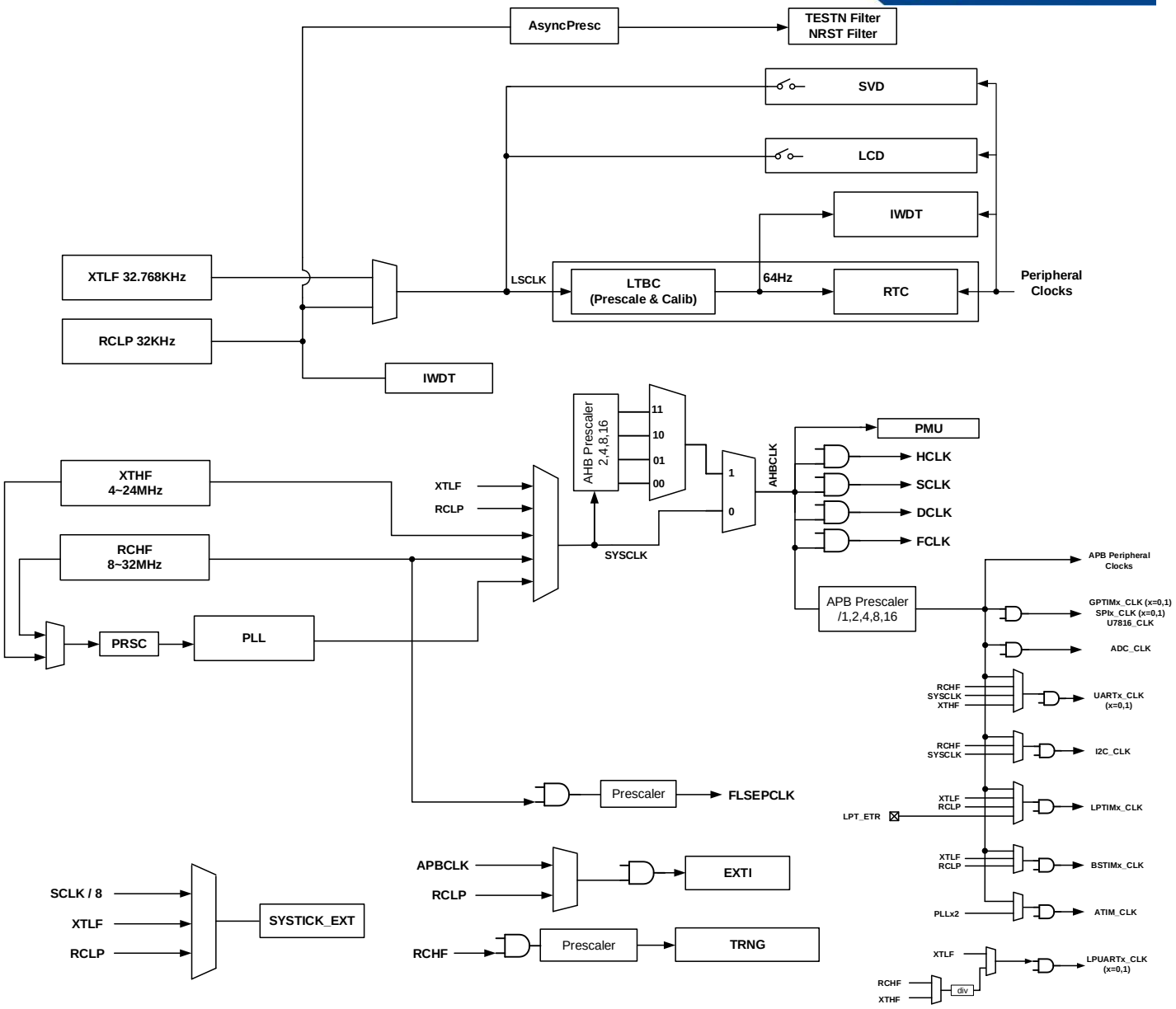


图 11-1 芯片时钟框图

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8MHz RCHF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。

11.2.2 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	RCLP 使能

11.2.3 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，AHBPRES寄存器不复位。

- 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。

11.2.4 主要时钟说明

时钟	源头	说明
LSCLK	XTLF, RCLP	32.768KHz 低频系统时钟，晶体停振时软件可以切换到 RCLP 主要用于 RTC、LCD、SVD
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF	32.768K~72MHz，经过分频后得到 AHBCLK
HCLK(AHBCLK)	SYSCLK	AHB 总线时钟，用于驱动 CPU、RAM、Flash 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

11.2.5 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
LPUARTx (x=0,1)	APBCLK	XTLF
		RCHF
		XTHF
I2C	APBCLK	APBCLK
		RCHF
		SYSCLK
LPTIM16	APBCLK	APBCLK
		XTLF
		RCLP
		LPT16_ETR
BSTIM16	APBCLK	APBCLK
		XTLF
		RCLP
ADC	APBCLK	XTHF
		RCHF
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		RCLP
TRNG	APBCLK	RCHF
IWDT	APBCLK	RCLP
LCD	APBCLK	LSCLK
RTC	APBCLK	LSCLK
非独立工作时钟外设		
PMU	AHBCLK	

模块	总线时钟	工作时钟
DMA	AHBCLK	
ATIM	APBCLK	
GPTIMx (x=0,1,2)	APBCLK	
UARTy (y=4,5)	APBCLK	
SPIx (x=0,1)	APBCLK	
7816	APBCLK	
AES	APBCLK	
CRC	APBCLK	
WWDT	APBCLK	
OPAx	APBCLK	
COMPx	APBCLK	

11.2.6 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LPUARTx、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。

11.2.7 LSCLK

LSCLK由XTLF和RCLP二选一产生，选择由软件操作LSCLKSEL寄存器完成。

11.2.8 SYSTICK 时钟

CPU的SYSTICK计数器时钟可以选择使用CPU的内部系统时钟和外部时钟，内部系统时钟即SCLK时钟，外部时钟有3个源可选，分别为SCLK的8分频，XTLF和RCLP时钟，由STCLKSEL寄存器选择。

11.3 高频 RC 振荡器(RCHF)

11.3.1 概述

高频RC振荡器典型振荡频率为8/16/24MHz，可用作系统主时钟，MCU在此频率下工作，可达到性能与功耗的平衡。为满足不同应用对MCU执行速度的需求，高频RC振荡器的输出频率可调，最高能达到32MHz。RCHF输出频率可以进行调校，出厂前调校到目标频率的+/-0.5%以内，8MHz输出全温区（-40~+125°C）频率变化小于+/-2.5%，16MHz输出全温区（-40~+125°C）频率变化小于+/-3%。

11.3.2 软件使用指南

芯片上电后默认使用RCHF 8MHz时钟工作，硬件电路会自动从LDT0读取8MHz校准值，保证8MHz频率常温下误差小于+/-0.5%。

如果软件需要使用其他频率，则按照以下步骤操作：

- 改写RCHFCR.FSEL
- 从LDT0读取频率调校值
- 将频率调校值写入RCHFTR寄存器，即可得到常温下目标频率误差小于+/-0.5%的时钟

11.4 低功耗 RC 振荡器(RCLP)

11.4.1 概述

RCLP是一个极低功耗的低频环振，典型频率是32.768Khz。为提高可靠性，在任意条件下都应保持RCLP开启状态。

RCLP出厂时经过测试校准，软件使用前可以从0x1FFF_FB20地址读取校准值，并写入RCLPTR寄存器，即可获得校准的低频时钟。

11.5 高频晶体振荡电路(XTHF)

11.5.1 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4~32MHz晶体。软件可以通过XTHFEN寄存器使能或关闭XTHF时钟。

11.5.2 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PC2和PC3引脚配置为模拟功能。

11.5.3 停振检测（HFDET）

FM33LF0A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，同时，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），配置LOC_SW_OFF寄存器为0，则停振信号将可以自动使能RCHF并将SYSCLK切换到RCHF，以避免高频晶体意外停振导致系统死机，在特定实时应用中，这

种时钟的切换是不希望的，则将LOC_SW_OFF保持为1，即使由于干扰发生了HFDET信号，也会继续保持当前时钟配置。

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。

11.6 低频晶体振荡电路(XTLF)

11.6.1 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，4bit复位值为1010，即上电默认关闭，必须改写为0101才能使能XTLF。如果写入的值不是0101或1010，则写入无效，不改变寄存器值。

11.6.2 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。

XTLF复用引脚PD9和PD10的控制寄存器复位默认是模拟通道。

11.6.3 停振检测

FM33LF0xxA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LSCLKSEL寄存器将LSCLK切换到RCLP。

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。

11.7 锁相环(PLL)

11.7.1 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72MHz。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。

11.7.2 软件使用指南

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的时钟源RCHF或XTHF必须分频到1MHz再倍频
- 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72MHz，并使系统以72MHz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2
- 使能PREFETCH（可选）
- 将AHB时钟选择为PLL输出

11.8 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/RTCBKP	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	

11.9 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并恢复到休眠前的频率输出；同时将SYSCLKSEL寄存器复位成00，将系统时钟选为RCHF，而AHBPRES寄存器不会被复位，保持休眠前的状态；因此芯片唤醒后默认将使用RCHF或者其分频时钟工作。

11.10 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
0x10	RCLP 控制寄存器（RCLP Control Register）	CMU_RCLPCR
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_XTHCR
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_PCLKCR3
0x40	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_PCLKCR4
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	外设工作时钟使能寄存器 1（Peripheral Clock Enable Register）	CMU_OPCER1
0x50	外设工作时钟使能寄存器 2（Peripheral Clock Enable Register）	CMU_OPCER2
0x54	AHB Master 控制寄存器（AHB Master Control Register）	CMU_AHBMCR

11.10.1 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	-						SLP_EN EXTI	-
位权限	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	RFUI		-			AHBPRES		
位权限	R/W-00		U-0			R/W-011		
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
31:26	--	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 RCLP） 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 101：4 分频 110：8 分频 111：16 分频 注意：当系统时钟源发生停振时，自动复位此寄存器
7:6	STCLKSEL	CPU 内核 systick 的外部工作时钟选择 00：SCLK8 分频 01：XTLF 10：RCLP 11：SCLK8 分频
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

Bit	助记符	功能描述
		111：RCHF 注意：当系统时钟源发生停振时，自动复位此寄存器

11.10.2 RCHF 时钟控制寄存器（CMU_RCHFCR）

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
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

11.10.3 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，8'h00 表示频率最低，8'hFF 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。

11.10.4 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	-						PLLDB	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PLLDB							
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			-		INSEL	EN
位权限	R/Dy-0	R/W-000			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	PLLDB	PLL 倍频比，输出时钟频率为 1M*(PLLDB+1) 0x1F：输出 32 倍频 ..... 0x47：输出 72 倍频 *注：请勿使用高于 72 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

Bit	助记符	功能描述
3:2	--	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

11.10.5 RCLP 控制寄存器（CMU_RCLPCR）

名称	CMU_RCLPCR							
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	-							ENB
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	ENB	RCLP 常使能，此寄存器保持为 0

11.10.6 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

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 1111 1111：频率最高

11.10.7 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-1010			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RDY	-		GMCFG			ICFG	
位权限	R-0	U-0		R/W-100			R/W-01	

Bit	助记符	功能描述
31:16	--	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 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	RDY	XTLF 稳定标志，硬件置位，只读 关闭 XTLF 或者 XTLF 停振时自动清零
6:5	--	RFU：未实现，读为 0
4:2	GMCFG	XTLF 跨导配置信号,使用默认值无需改动
1:0	ICFG	XTLF 输入电流源配置信号，使用默认值无需改动

11.10.8 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	-							
位权限	U-0							
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
31:8	--	RFU：未实现，读为 0
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；

11.10.9 XTHF 控制寄存器（CMU_XTHCR）

名称	CMU_XTHCR							
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	WAIT			-	CFG			
位权限	R/W-000			U-0	R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BYPASS	-		RDY	EN
位权限	U-0			R/W-0	U-0		R-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
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	--	RFU：未实现，读为 0
11:8	CFG	XTHF 振荡强度配置 0000：最弱 ..... 1111：最强
7:5	--	RFU：未实现，读为 0
4	BYPASS	XTHF BYPASS 寄存器 0：XTHF BYPASS 无效 1：XTHF BYPASS 有效
3:2	--	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 停振时自动清零
0	EN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

11.10.10 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	-					SYSCSE_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	SYSCSE_IE	SYSCLK 时钟选择错误中断使能寄存器，1 有效

Bit	助记符	功能描述
1	HFDET_IE	XTHF 高频检测报警中断使能，1 有效
0	LFDET_IE	XTLF 高频检测报警中断使能，1 有效

11.10.11 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，软件写 1 清零；只有在 FFDETO 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 FFDETO 不为 0 的情况下才能够清除此寄存器

11.10.12 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	-				CLM0	CLM1	LCD	LED
位权限	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	-			SENT1	SENT0			

位权限	U-0			R/W-0	R/W-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						COMP	SVD
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD	ANT	IWDT	SCU	PMU	RTC	-	LPT16
位权限	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27	CLM0	CLM0 总线时钟使能，高使能
26	CLM1	CLM1 总线时钟使能，高使能
25	LCD	LCD 总线时钟使能，高使能
24	LED	LED 总线时钟使能，高使能
23:21	--	RFU：未实现，读为 0
20	SENT1	SENT1 总线时钟使能，高使能
19	SENT0	SENT0 总线时钟使能，高使能
18:10	--	RFU：未实现，读为 0
9	COMP	Comparator 总线时钟使能，高使能
8	SVD	SVD 总线时钟使能，高使能
7	PAD	PADCFG 总线时钟使能，高使能
6	ANT	模拟测试总线时钟使能，高使
5	IWDT	IWDT 总线时钟使能，高使能
4	SCU	SCU 总线时钟使能，高使能
3	PMU	PMU 总线时钟使能，高使能
2	RTC	RTC 总线时钟使能，高使能
1	--	RFU：未实现，读为 0
0	LPT16	LPTIM16 总线时钟使能，高使能

11.10.13 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	-				PGL	OPA	DAC	ADC
位权限	U-0				R/W-0	U-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT	RAMBIST	NVM	DMA	-	AES	TRNG	CRC
位权限	R/W-1	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
31:12	--	RFU：未实现，读为 0
11	PGL	PGL 总线时钟使能，高使能
10	OPA	RFU：未实现，读为 0
9	DAC	DIVAS 总线时钟使能，高使能
8	ADC	ADC 总线时钟使能，高使能
7	WWDT	WWDT 总线时钟使能，高使能
6	RAMBIST	RAMBIST 总线时钟使能，高使能
5	NVM	NVMIF（Flash 擦写控制器）总线时钟使能，高使能
4	DMA	DMA 总线时钟使能，高使能
3	--	RFU：未实现，读为 0
2	AES	AES 总线时钟使能，高使能
1	TRNG	RNG 总线时钟使能，高使能
0	CRC	CRC 总线时钟使能，高使能

11.10.14 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	-		I2CSMB0	I2CSMB1	-			
位权限	U-0		R/W-0	R/W-0	U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	UCIR	-			FSCAN	-	LPUART1	LPUART0
位权限	R/W-0	U-0			R/W-0	U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名				UART4	UART3	UART2	UART1	UART0
位权限				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	-						SPI2	SPI1
位权限	U-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	I2CSMB1 总线时钟使能，高有效
28	I2CSMB0	I2CSMB0 总线时钟使能，高有效
27:24	-	RFU：未实现，读为 0
23	UCIR	串口红外总线时钟使能，高有效
22:20		
19	FSCAN	FSCAN 总线时钟使能，高有效
18	-	RFU：未实现，读为 0
17	LPUART1	LPUART1 总线时钟使能，高有效
16	LPUART0	LPUART0 总线时钟使能，高有效
15:13	-	RFU：未实现，读为 0
12	UART4	UART4 总线时钟使能，高有效
11	UART3	UART3 总线时钟使能，高有效

位号	助记符	功能描述
10	UART2	UART2 总线时钟使能，高有效
9	UART1	UART1 总线时钟使能，高有效
8	UART0	UART0 总线时钟使能，高有效
7:2	-	RFU：未实现，读为 0
1	SPI2	SPI2 总线时钟使能，高有效
0	SPI1	SPI1 总线时钟使能，高有效

11.10.15 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	-				DIVAS	TAU0	-	BT16
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	-			AT	-	GT1	GT0	-
位权限	U-0			R/W-0	U-0	R/W-0	R/W-0	U-0

位号	助记符	功能描述
31:12	-	RFU：未实现，读为 0
11	DIVAS	DIVAS 总线时钟使能，高有效
10	TAU0	TAU0 总线时钟使能，高有效
9	-	RFU：未实现，读为 0
8	BT16	BSTIM16 总线时钟使能，高有效
7:5	-	RFU：未实现，读为 0
4	AT	高级定时器总线时钟使能，高有效
3	-	RFU：未实现，读为 0
2	GT1	通用定时器 1 总线时钟使能，高有效
1	GT0	通用定时器 0 总线时钟使能，高有效
0	-	RFU：未实现，读为 0

11.10.16 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	-		LPUART1CKS		LPUART0CKS	
位权限	U-0	R/W-0	U-0		R/W-00		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		UART3CKS		LPT16CKS		UART2CKS	
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	-				UART1CKS		UART0CKS	
位权限	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 RCLP 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
29:28	-	RFU：未实现，读为 0
27:26	LPUART1CKS	LPUART1 工作时钟选择 00：XTLF 01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RCLP 11：RFU
25:24	LPUART0CKS	LPUART0 工作时钟选择 00：XTLF 01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RCLP 11：RFU
23:20	-	RFU：未实现，读为 0
19:18	I2CSMB1CKS	I2C1 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：RFU
17:16	I2CSMB0CKS	I2C0 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：XTLF 10：RCLP 11：RFU
13:12	UART3CKS	UART3 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
11:10	LPT16CKS	LPTIM16 工作时钟选择

位号	助记符	功能描述
		00：APBCLK 01：XTLF 10：RCLP 11：RFU
9:8	UART2CKS	UART2 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
7:4	-	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

11.10.17 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	-		FSCANCKS	
位权限	R/W-00		U-0	R/W-0	U-0		R/W-0	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LEDCKS		-	RNGPRSC			-	
位权限	R/W-00		U-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LEDPRSC			-				
位权限	R/W-000			U-0				

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	--	RFU：未实现，读为 0
20	RTCCKS	RTC 工作时钟选择 0：LSCLK 1：XTHF 分频

Bit	助记符	功能描述
19:18	--	RFU：未实现，读为 0
17:16	FSCANCKS	FSCAN 工作时钟选择 00：RCHF 01：XTHF 10：PLL 11：APBCLK
15:14	LEDCKS	LED 工作时钟选择 00：RCHF 01：XTHF 10：PLL 11：APBCLK
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 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110, 111：RFU
9:8	--	RFU：未实现，读为 0
7:5	LEDPRSC	LED 工作时钟分频 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
4:0	--	RFU：未实现，读为 0

11.10.18 外设工作时钟使能寄存器（CMU_OPCER1）

名称	CMU_OPCER1							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	NVMCKE	LPUART1CKE	LPUART0CKE	-		LEDCKE	RNGCKE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		I2C1SMBCKE	I2C0SMBCKE	-			
位权限	U-0		R/W-0	R/W-0	U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	ATCKE	-					
位权限	U-0	R/W-0	U-0					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	UART3CKE	UART2CKE	UART1CKE	UART0CKE	BT16CKE	-	LPT16CKE	-
位权限	R/W-0	R/W-0	R/W-0	R/W-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	NVMCKE	Flash 擦写时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
29	LPUART1CKE	LPUART1 工作时钟使能，高有效
28	LPUART0CKE	LPUART0 工作时钟使能，高有效
27:26	--	RFU：未实现，读为 0
25	LEDCKE	LED 工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
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	I2CSMB1CKE	I2CSMB1 工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
20	I2CSMB0CKE	I2CSMB0 工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
19:15	--	RFU：未实现，读为 0
14	ATCKE	高级定时器工作时钟使能，高有效 *注意：芯片休眠后此寄存器被复位*
13:8	--	RFU：未实现，读为 0
7	UART3CKE	UART3 工作时钟使能，高有效
6	UART2CKE	UART2 工作时钟使能，高有效
5	UART1CKE	UART1 工作时钟使能，高有效
4	UART0CKE	UART0 工作时钟使能，高有效
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

11.10.19 外设工作时钟使能寄存器（CMU_OPCER2）

名称	CMU_OPCER2							
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	-							FSCANCKE
位权限	U-0							R/W-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							
位权限	U-0							

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	FSCANCKE	FSCAN 工作时钟使能，高有效
7:0	--	RFU：未实现，读为 0

11.10.20 AHB Master 控制寄存器（CMU_AHBMCR）

名称	CMU_AHBMCR							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOC_SW_OFF	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	LOC_SW_OFF	时钟丢失切换开关，默认为 1 则表示默认在 XTHF 为系统时钟基础时，停振检测发现晶振出现停振时，也不会自动切换为 RCHF。如果这位寄存器写为 0，则会在检测到停振信号时自动将系统时钟切换到 RCHF。
30:1	--	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

12 时钟监控（CLM）

12.1 概述

时钟监控模块用于交叉监控两个时钟之间确定的频率关系，当相对频率出现明显异常时产生报警中断或者全局复位。

本芯片共有2个CLM模块。其主要作用是监控CLK_MON时钟频率范围并确认CLK_MON是否工作正常，以更好地实现系统功能安全设计。

12.2 结构框图

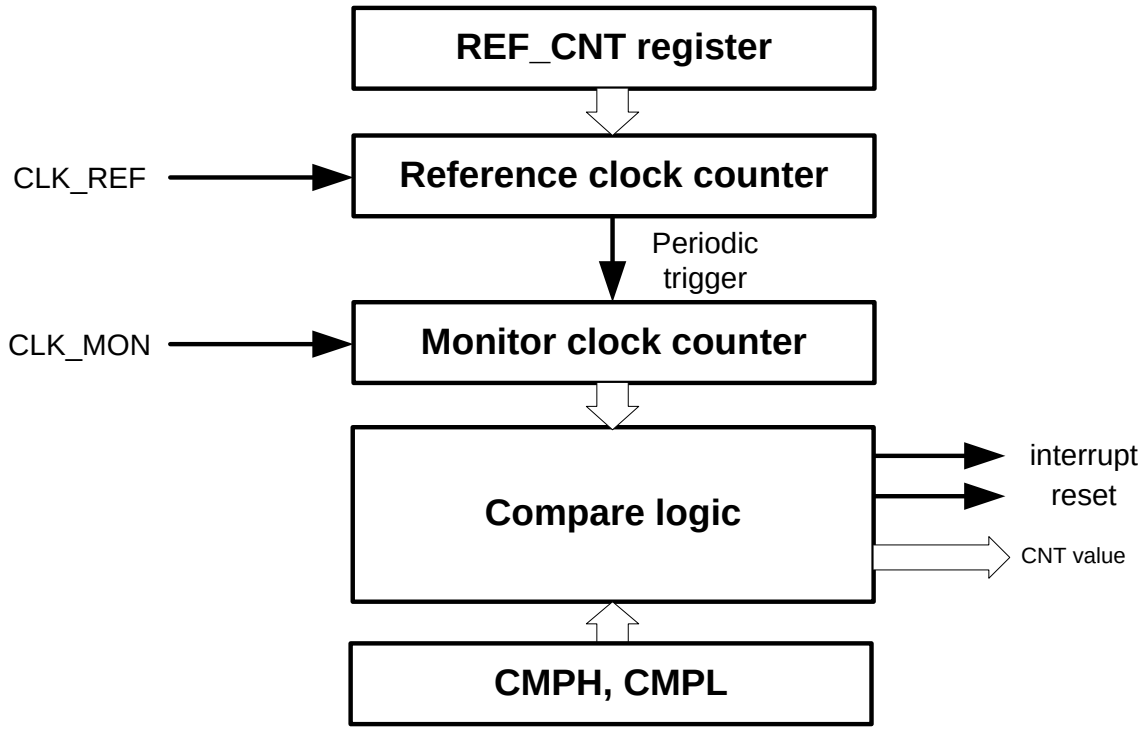


图 12-1 CLM 电路框图

其中reference clock counter位宽8bit，monitor clock counter位宽24bit

CLK_REF固定为RCLP，CLK_MON可以从多个高频时钟里选择（RCHF、XTHF、PLL）

12.3 功能描述

12.3.1 工作模式

CLM模块支持复位模式或中断模式。

在复位模式下（MODE=10），当计数值超过CMPH和CMPL设置的上下限，CLM输出全局复位信号。

在中断模式下（MODE=00/01/11），当计数值超过CMPH和CMPL设置的上下限，CLM输出中断信号。

12.3.2 使能与关闭

CLM模块上电后默认关闭，由软件完成配置后使能，由于涉及功能安全，CLM控制寄存器受到写保护，以防止寄存器被异常改写，参见SFU章节。建议软件使能CLM后，通过SFU模块启动CLM模块寄存器写保护功能。

当CLK_MON为XTHF和PLL时，由于时钟启动需要时间，软件必须注意在确认监控时钟正常启动后再使能CLM模块，避免CLM出现误动作。

- CLK_MON=XTHF：使能CLM前应查询CMU_XTHFCR.RDY
- CLK_MON=PLL：使能CLM前应查询CMU_PLLCR.LOCKED

当芯片进入休眠模式时，由于CLK_MON的源头时钟都被自动关闭，此时CLM模块自动停止工作。

12.3.3 时钟频率监控

CLM频率监控用于监视CLK_MON的频率是否处于系统允许的正常范围之内，其工作原理是2个独立工作的定时器，分别由CLK_REF和CLK_MON驱动。其中参考时钟计数器的溢出周期由REF_CNT寄存器定义，每次溢出时输出触发信号清零监控时钟计数器，并且在触发点检查监控时钟计数器的值是否处于规定范围之内。

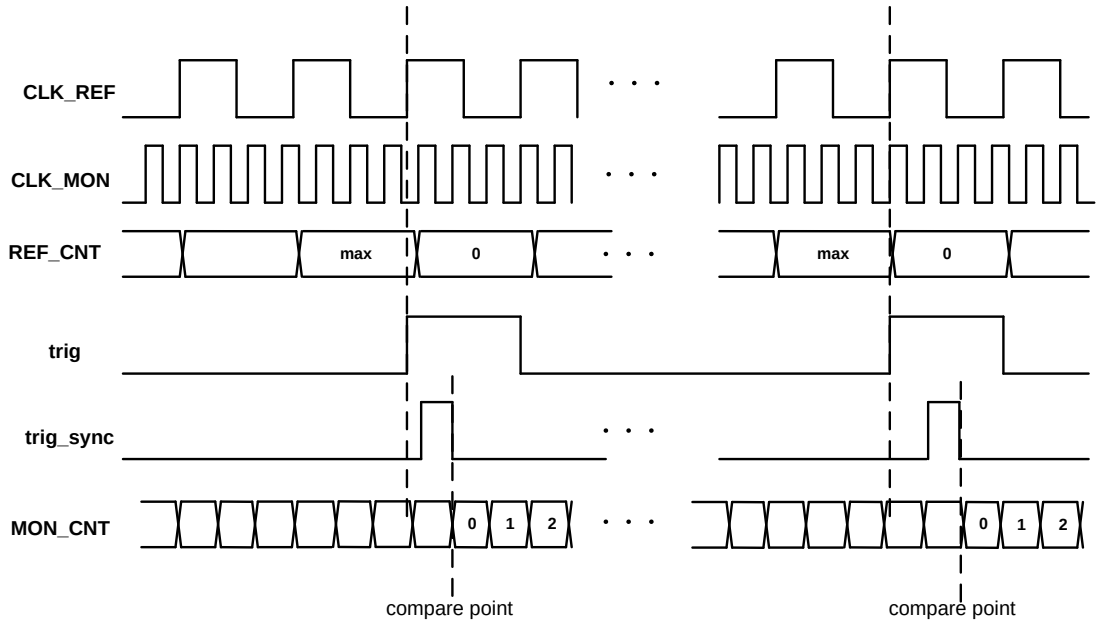


图 12-2 参考时钟计数和监控时钟计数

CMPH和CMPL定义了监控阈值的上下限，在一个监控周期内，如果计数值超过了上下阈值，则说明被监控的时钟频率异常，此时CLM产生异常事件，根据系统要求，此异常可以由软件处理或者直接产生系统复位。

下图中，CLK_MON频率异常，低于正常范围，导致UDF置位。

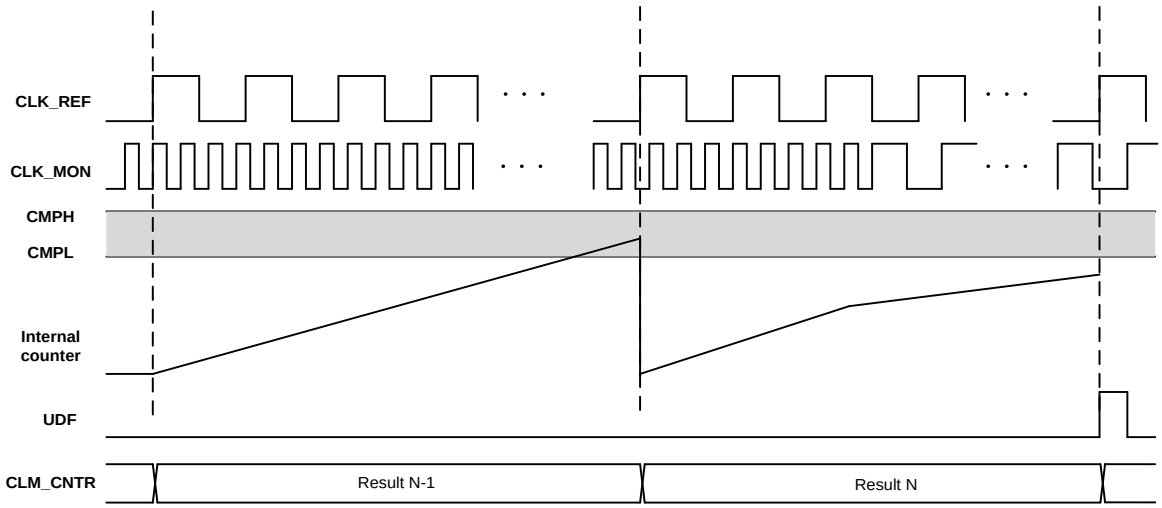


图 12-3 监控时钟频率下溢出

下图为监控时钟频率上溢出的例子。

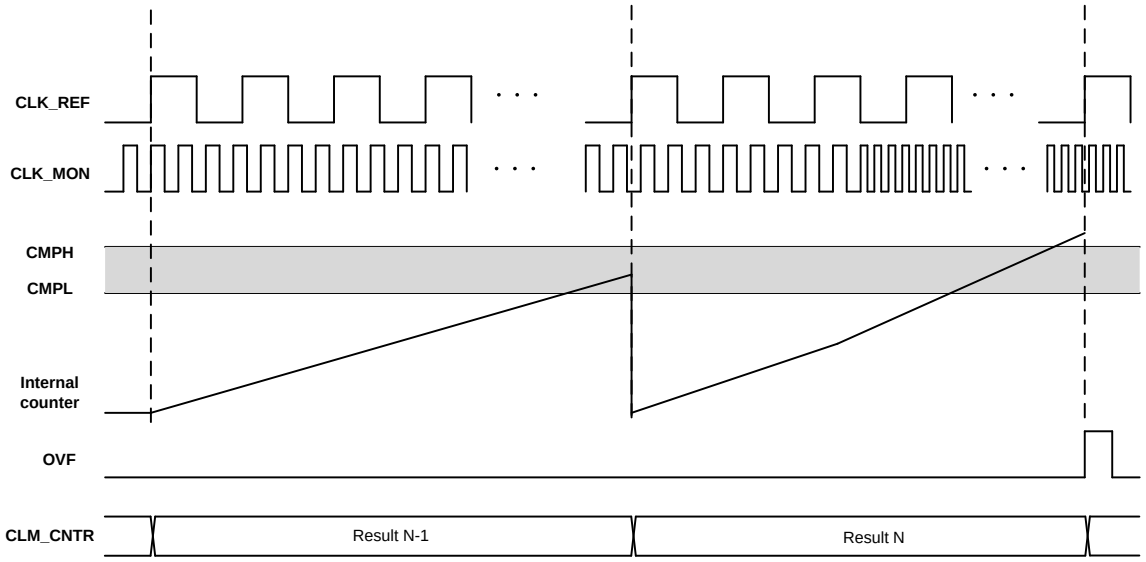


图 12-4 监控时钟频率上溢出

使用时钟监控功能需要注意以下几点：

- 为了避免监控计数器溢出，需要根据CLK_REF和CLK_MON之间的频率关系，合理设置参考时钟计数周期
- 设置CMPH和CMPL时要充分考虑时钟的温度系数，并留有足够余量；设置过小的阈值范围可能导致误报警
- 硬件不负责检查CMPH和CMPL的大小关系，只要计数值高于CMPH或低于CMPL就会报警
- 置位EN寄存器之前，先要完成参考时钟和监控时钟的选择

设置检测参数应考虑系统故障响应时间的要求。最差情况下，检测到系统时钟丢失需要REF_CLK一个完整监控周期，以下面的例子说明：

- 系统使用PLL=64MHz为主时钟工作
- 监控时钟为PLL，PLL的源时钟为XTHF，温度系数小于100ppm
- 参考时钟为RCLP，温度系数小于10%
- 假设检测PLL时钟丢失的最大故障响应时间是250us
- 考虑RCLP标称频率32KHz，可设置REF_CLK预分频系数为6，即检测周期约180us，考虑温度系数最大不超过200us，满足最大故障响应时间
- 监控时钟在检测周期内的理想计数值应该是11520
- 设置CMPL=0.8*11520=9200，CMPH=1.2*11520=13800

12.3.4 超时检测

时钟频率监控无法检测到监控时钟完全停止的异常情况，因此需要超时检测机制识别到监控时钟计数器长时间没有变化的情况。CLM使能后，TO计数器使用RCLP计数，溢出周期为15

注意：CLM超时检测的原理类似于IWDT，但是仅针对时钟丢失的异常，识别周期更短，能够满足系统故障响应时间的要求。

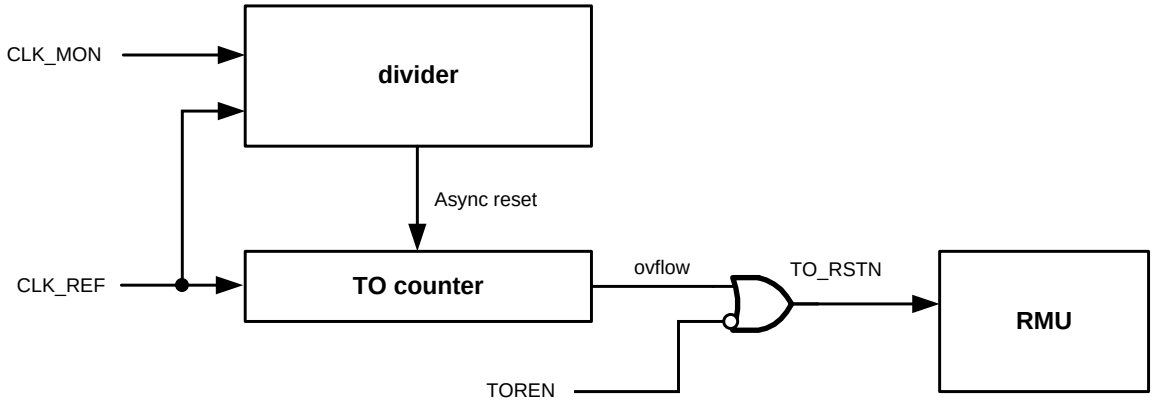


图 12-5 超时检测电路

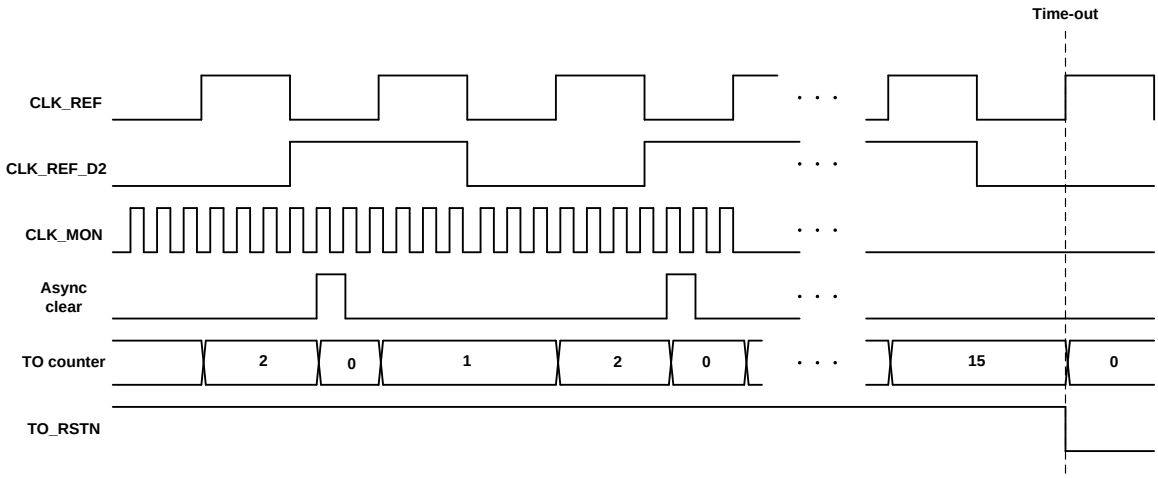


图 12-6Time-out 复位

实现原理如上图，对CLK_REF分频得到CLK_REF_D2，使用CLK_MON异步采样CLK_REF_D2得到异步复位信号，用这个异步复位信号清零超时计数器。只要CLK_MON不停振，TO计数器永远没有机会递增到3以上，而超时阈值固定为15，当TO计数器计数到15时，即产生复位。

12.4 寄存器

CLM0模块起始地址：0x4001_B000

CLM1模块起始地址：0x4001_B400

offset 地址	名称	符号
0x00	时钟监控控制寄存器	CLM_CR
0x04	时钟监控配置寄存器	CLM_CFGR
0x08	时钟监控计数值寄存器	CLM_CNTR
0x0C	时钟监控中断标志寄存器	CLM_ISR
0x10	时钟监控比较高阈值寄存器	CLM_CMPH
0x14	时钟监控比较低阈值寄存器	CLM_CMPL

12.4.1 时钟监控控制寄存器（CLM_CR）

名称	CLM_CRx (x=0,1)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOREN	-						
位权限	R/W-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	MODE		-		UDFIE	OVFIE	MONIE	EN
位权限	R/W-01		U-0		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31	TOREN	Time-out reset enable，仅在 EN=1 时有效 0：禁止超时复位输出 1：允许超时复位输出
30:8	--	RFU：未实现，读为 0
7:6	MODE	CLM 工作模式选择 10：复位模式 其他：中断模式
5:4	--	RFU：未实现，读为 0
3	UDFIE	时钟监控计数器下溢出中断使能 0：禁止中断 1：允许中断
2	OVFIE	时钟监控计数器上溢出中断使能 0：禁止中断 1：允许中断
1	MONIE	时钟监控周期中断使能

Bit	助记符	功能描述
		0：禁止中断 1：允许中断
0	EN	CLM 使能 1：启动 0：关闭，内部计数器复位

12.4.2 时钟监控配置寄存器（CLM_CFGR）

名称	CLM_CFGRx (x=0,1)							
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	REF_CNT							
位权限	R/W-00100000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						MONSEL	
位权限	U-0						R/W-00	

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	REF_CNT	参考时钟周期 检测周期=T_{REF_CLK}*(REF_CNT+1)
7:2	--	RFU：未实现，读为 0
1:0	MONSEL	监控时钟选择 (Monitor clock select) 00：RCHF 01：PLL 10：XTHF 11：XTHF

12.4.3 时钟监控计数值寄存器（CLM_CNTR）

名称	CLM_CNTRx (x=0,1)							
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[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	时钟监控计数器计数值 (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

12.4.4 时钟监控中断标志寄存器（CLM_ISR）

名称	CLM_ISRx (x=0,1)							
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	-					UDF	OVF	MON_IF
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	UDF	时钟监控下溢出标志，监控周期内计数值小于监控阈值下限 硬件置位，软件写 1 清零
1	OVF	时钟监控上溢出标志，监控周期内计数值大于监控阈值上限 硬件置位，软件写 1 清零
0	MON_IF	时钟监控周期中断标志 (Clock monitor interrupt flag) 每个时钟监控周期完成后硬件置位，软件写 1 清零

12.4.5 时钟监控比较高阈值寄存器（CLM_CMPH）

名称	CLM_CMPHx (x=0,1)							
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	CMPH[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMPH[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	CMPH	时钟监控比较值高阈值

12.4.6 时钟监控比较低阈值寄存器（CLM_CMPL）

名称	CLM_CMPLx (x=0,1)							
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	CMPL[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMPL[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	CMPL	时钟监控比较值低阈值

13 电源电压监测（SVD）

13.1 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或周期使能以节省功耗。

特点：

- 监测主电源，电压低于或高于设定的阈值时产生欠压和复压中断
- 低压检测范围 1.8V~4.8V，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
- 可间歇式工作
- 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

13.2 结构框图

下图是电源检测电路的模块框图。

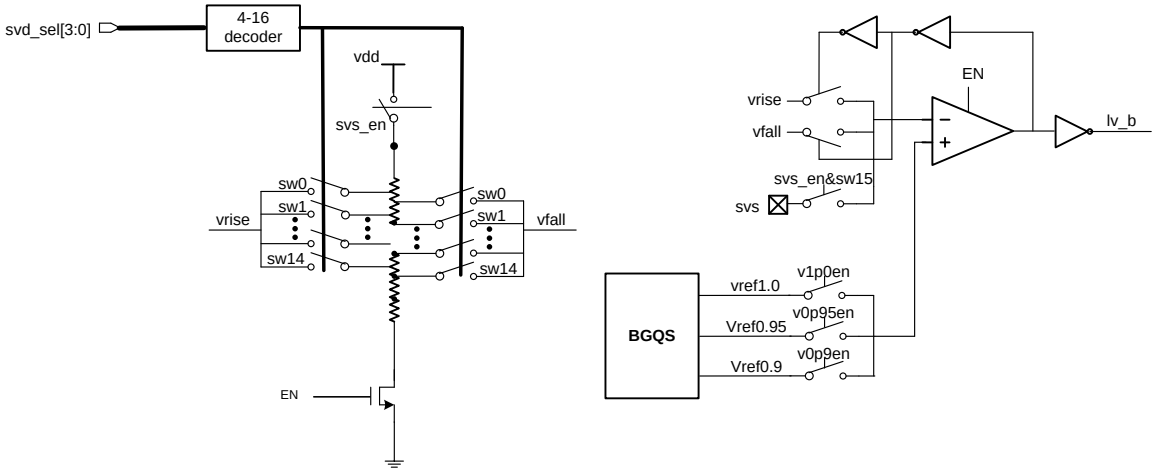


图 13-1 低压检测电路框图

SVD共有15个内部通道和1个外部通道，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：

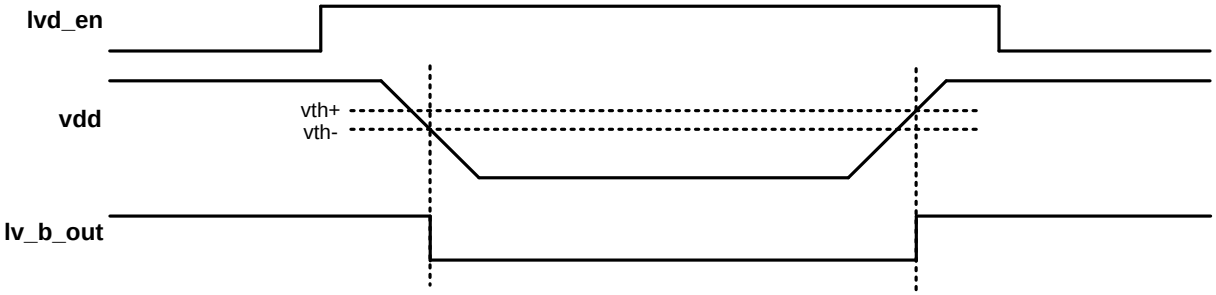


图 13-2 低压检测电路工作时序

13.3 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）。

13.4 时钟和复位

总线时钟为APBCLK。

工作时钟固定为RCLP，用于数字滤波、间歇定时。

13.5 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，检测范围1.8V~4.8V，每级相差0.214V；另外还支持1路外部输入检测通道。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。为节省功耗，使用低压报警功能时又可分为常使能和间歇工作两种模式。间歇工作时，可通过设置节DSEF寄存器设置开启时间间隔。

常使能条件下SVD从欠压到过压有0.1V回滞窗口，而间歇使能情况下没有回滞窗口；对于内部通道，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。而对于SVS通道，则需要特殊设计，由数字电路锁存上一次间歇窗口的判决结果，作为本次间歇窗口中的阈值选择依据，从而实现SVS的下降阈值和上升阈值选择。相应的，BG需要输出1.0V、0.9V、0.8V三个基准电压。

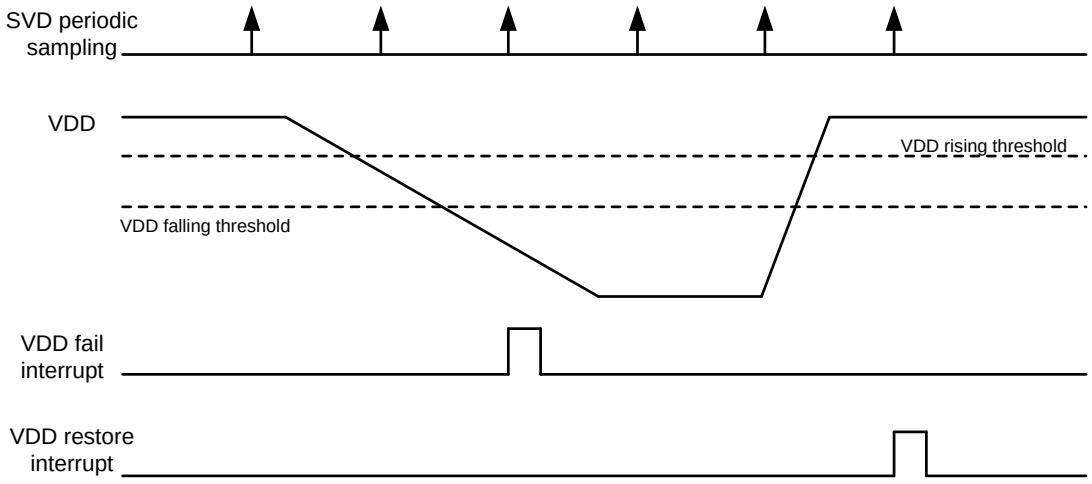


图 13-3 电源检测电路间歇工作模式

间歇工作时，当软件使能SVD的间隙使能后，SVD并不一定会立刻工作，而是要等待下一个开启窗口到来。而常使能情况下，软件开启SVD后经过一到两个RCLP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。
- 在间歇使能/内部通道模式下，在每次间歇使能启动时（即不使能到使能时）检测下降阈值，因此没有阈值窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将阈值档位恢复。
- 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、0.9V、0.8V，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。
- 在间歇使能/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、0.9V、0.8V，检测阈值没有窗口，需要软件配合，即在前次间歇使能检测到欠压时，软件将阈值档位调高一档；在前次间歇使能检测到非欠压时，软件将档位恢复。

13.5.1 间歇使能模式

在休眠模式下可以通过间歇使能来降低SVD的平均功耗。在DeepSleep模式下，AVREF默认关闭，

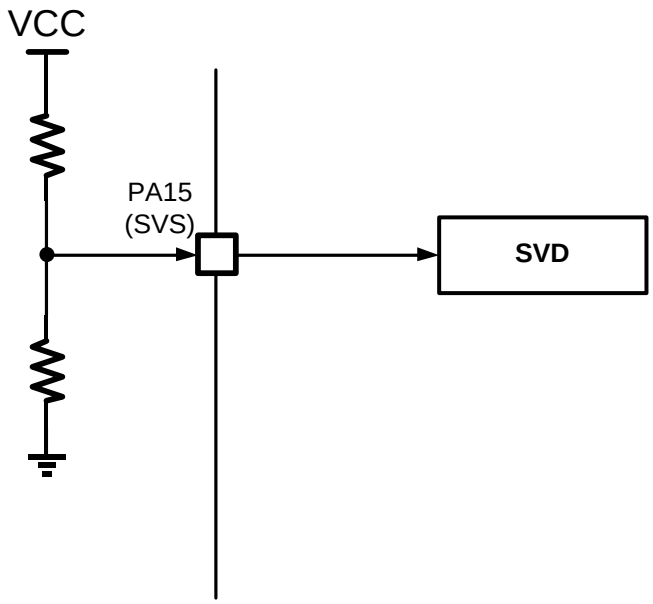
SVD开启窗口中，比较器加AVREF总功耗小于5uA。如果开启间隔设置为1s，则平均电流小于5nA。通过间歇使能模式，芯片可以保持对VDD电源供电情况的监视，同时几乎不会带来额外的休眠功耗增加。

13.5.2 外部电源检测

SVD除了可以检测芯片电源，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的输入经过外部电阻分压后，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：



寄存器配置方法如下表：

SVSEN	SVDLVL	说明
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	电源电压检测 分压后的档位参见后续章节描述

13.6 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。

电源检测：SVSxEN = 1，{VREF1P0EN, VREF0P95EN, VREF0P9EN} = 100，比较基准1.0V

SVDLVL	上升阈值（V）	下降阈值（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

电源检测：SVSxEN = 1，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 010，比较基准0.9V

SVDLVL	上升阈值（V）	下降阈值（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

电源检测：SVSxEN = 1，{ VREF1P0EN, VREF0P95EN, VREF0P9EN } = 001，比较基准0.8V

SVDLVL	上升阈值（V）	下降阈值（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

13.7 寄存器

模块偏移地址：0x4001_2800

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

13.7.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset								
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	ITVL	
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 13.6 电源检测阈值 (SVD threshold level)
3	DFEN	数字滤波使能(Digital Filter Enable) 1：启动 SVD 输出的数字滤波

Bit	助记符	功能描述
		0：关闭 SVD 输出的数字滤波
2	MOD	SVD 低压报警工作模式选择，仅 SVDEN=01 时有效 1：间歇使能模式 0：常使能模式 注意：间歇使能模式下必须开启数字滤波
1:0	ITVL	SVD Interval，SVD 低压报警间歇使能间隔 (SVD interval enable period) 00：62.5ms 01：256ms 10：1s 11：4s

13.7.2 SVD 控制寄存器（SVD_CR）

名称	SVD_CR							
offset								
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	FLS_EPGRD	-				SVS0EN	SVDEN	
位权限	R/W-0	U-0				R/W-0	R/W-00	

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	FLS_EPGRD	Flash 擦写保护功能，使用 SVD 的欠压信号对 flash 擦写操作进行保护，避免低压下 flash 擦写的不可靠。
6:3	--	RFU：未实现，读为 0
2	SVSEN	SVS 外部电源检测通道控制信号 (SVS external monitor channel enable) 0：SVS 通道关闭 1：SVS 通道使能 当 SVSEN=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。
1:0	SVDEN	SVD 使能 (SVD enable) 00：关闭 SVD 01：SVD 低压报警模式使能 10：RFU 11：关闭 SVD

13.7.3 SVD 中断使能寄存器（SVD_IER）

名称	SVD_IER							
offset								
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) 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

13.7.4 SVD 状态和标志寄存器（SVD_ISR）

名称	SVD_ISR							
offset								
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

Bit	助记符	功能描述
		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

13.7.5 SVD 参考电压选择寄存器（SVD_VSR）

名称	SVD_VSR							
offset								
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

14 AES 硬件运算单元（AES）

14.1 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

14.2 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_KEYRx寄存器中的密钥加密。

模式2：密钥扩展，把初始存储在AES_KEYRx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_KEYRx寄存器中的解密密钥（预计算的）解密。

模式4：用存储在AES_KEYRx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_DINR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_DOUTR寄存器中读4次共128bit的结果。

AES还支持DMA模式。通过配置DMAOUTEN=1和DMAINEN=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。

14.3 AES 数据流处理模式

AES有4种数据流处理模式：ECB，CBC，CTR，GCM。

14.3.1 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14-1和图14-2所示。

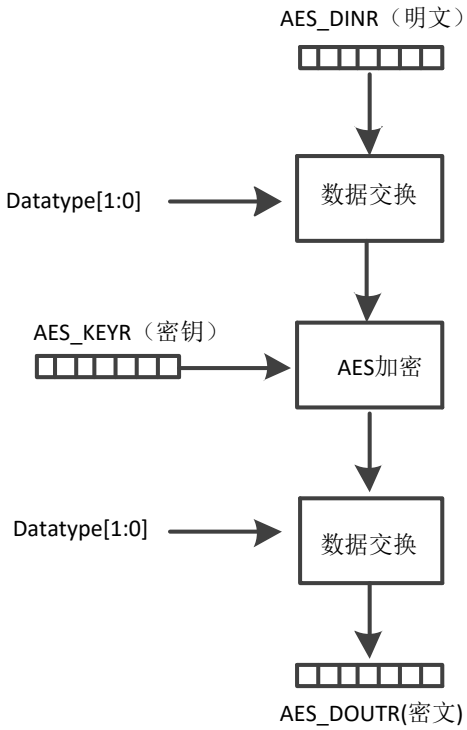


图 14-1 ECB 模式加密流程

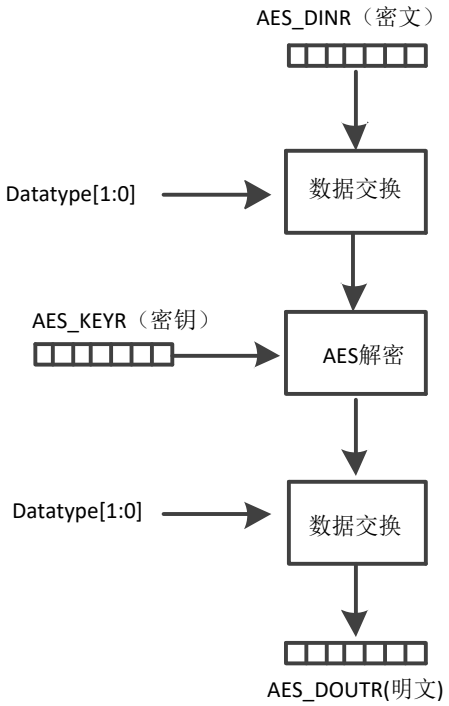


图 14-2 ECB 模式解密流程

14.3.2 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14-3和图14-4所示。

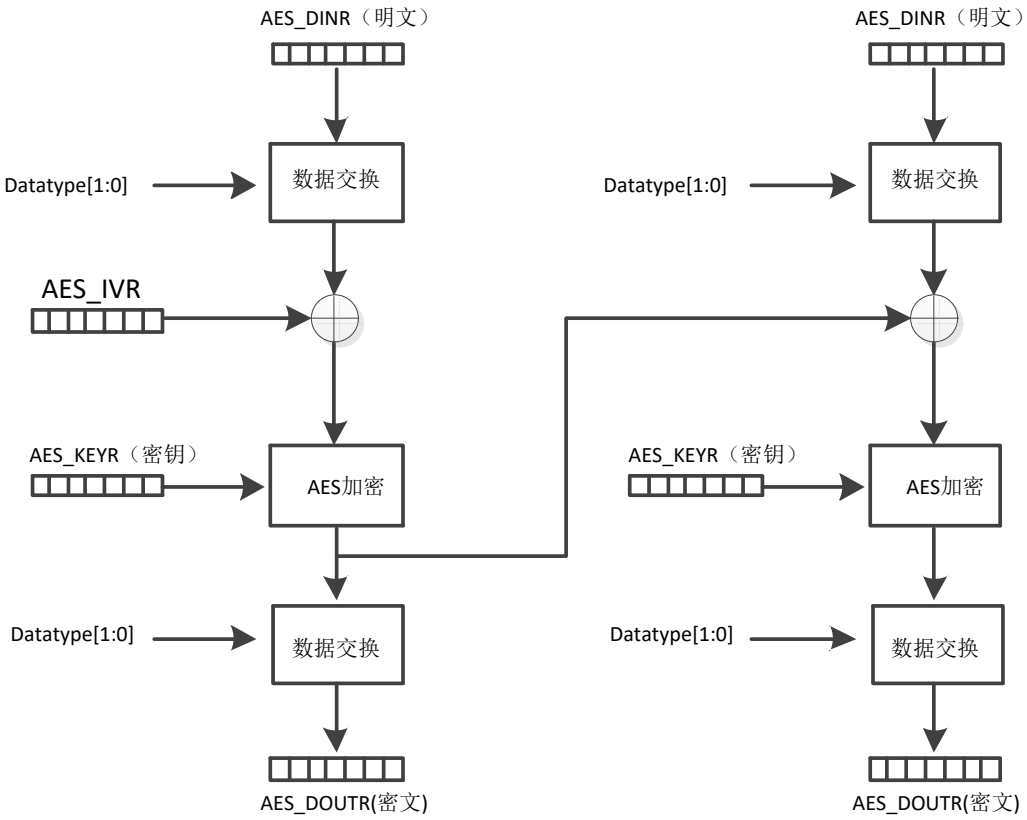


图 14-3 CBC 加密过程

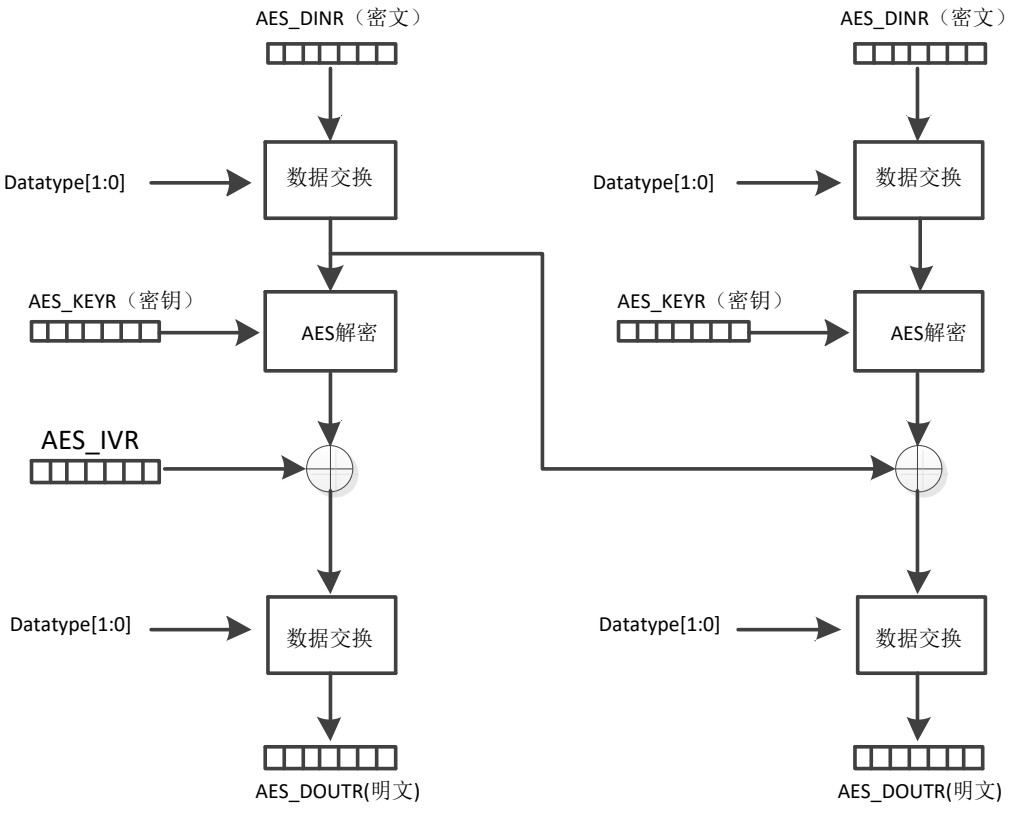


图 14-4 CBC 解密过程

注：在AES工作时读取AES_IVR寄存器的值为0x00000000

14.3.3 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14-5所示



图 14-5 暂停模式流程

14.3.4 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14-6和图14-7所示。

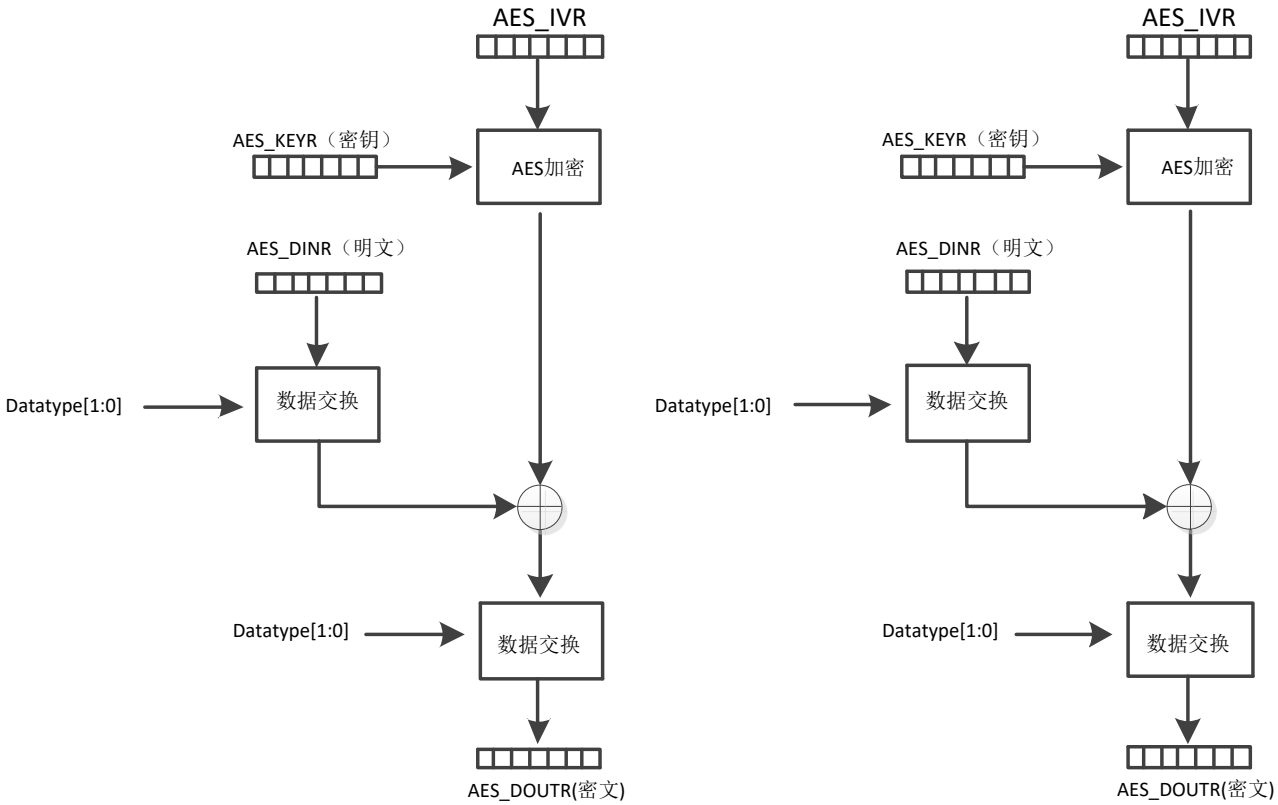


图 14-6 CTR 加密流程

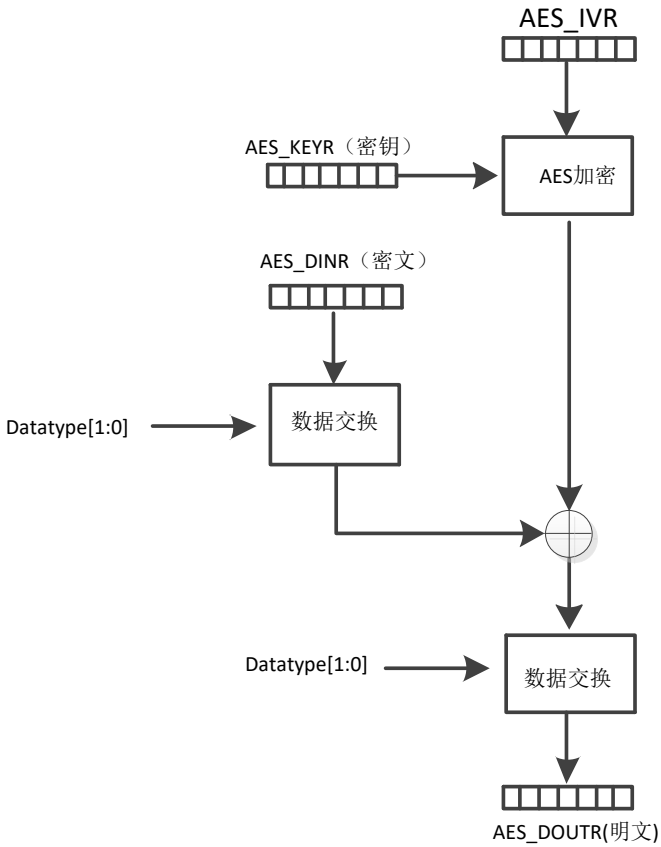


图 14-7 CTR 解密流程

随机数（nonce）和32位计数器存储在IV寄存器中，如图14-8所示

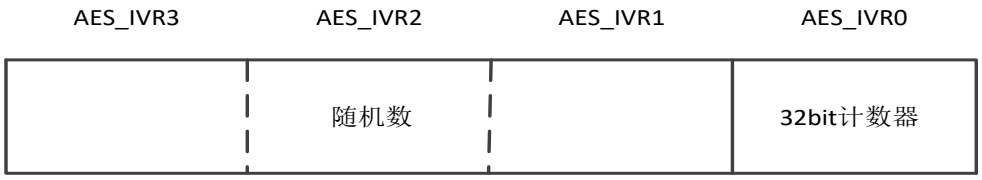


图 14-8 32 位计数器和随机数的存储方式

CTR模式下密钥扩展和解密模式没有意义。

14.3.5 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。

14.3.6 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：

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

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

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

GCM模式的加解密流程如图14-9，图14-10所示。

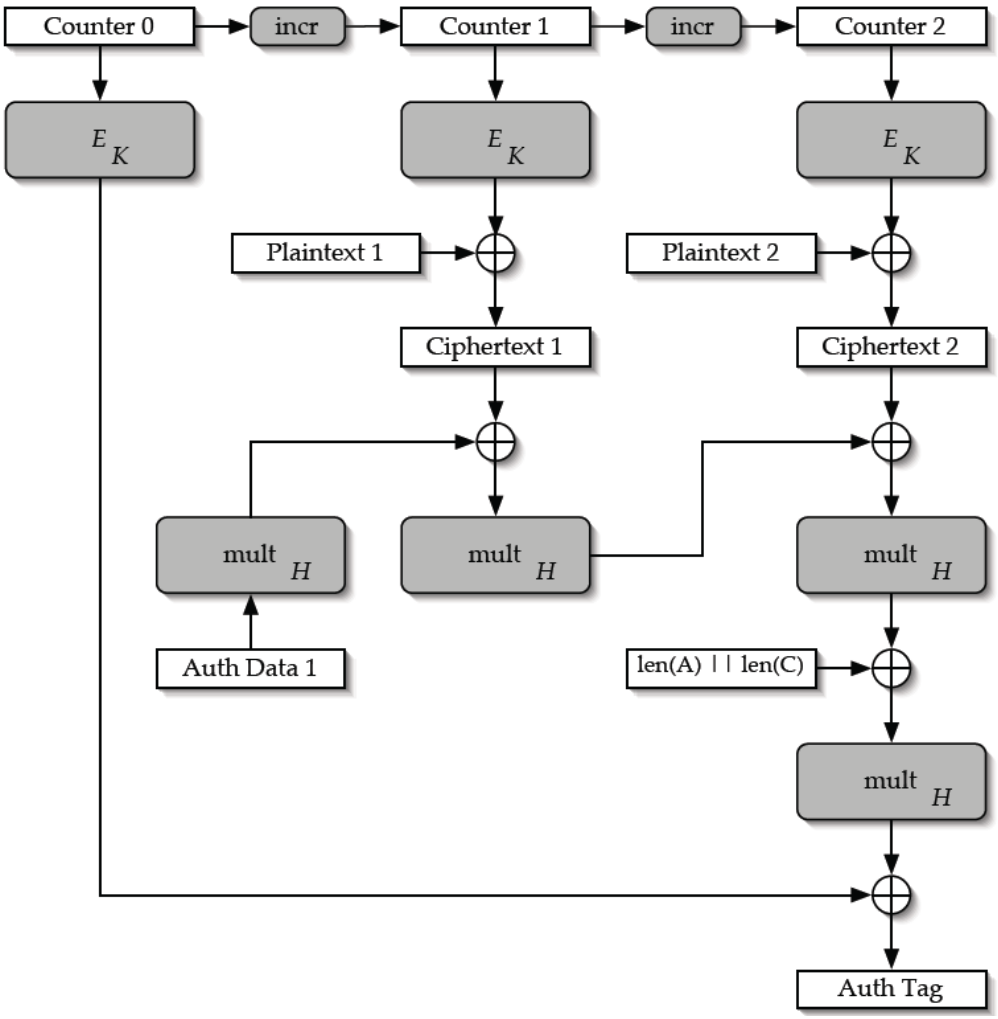


图 14-9 GCM 加密流程

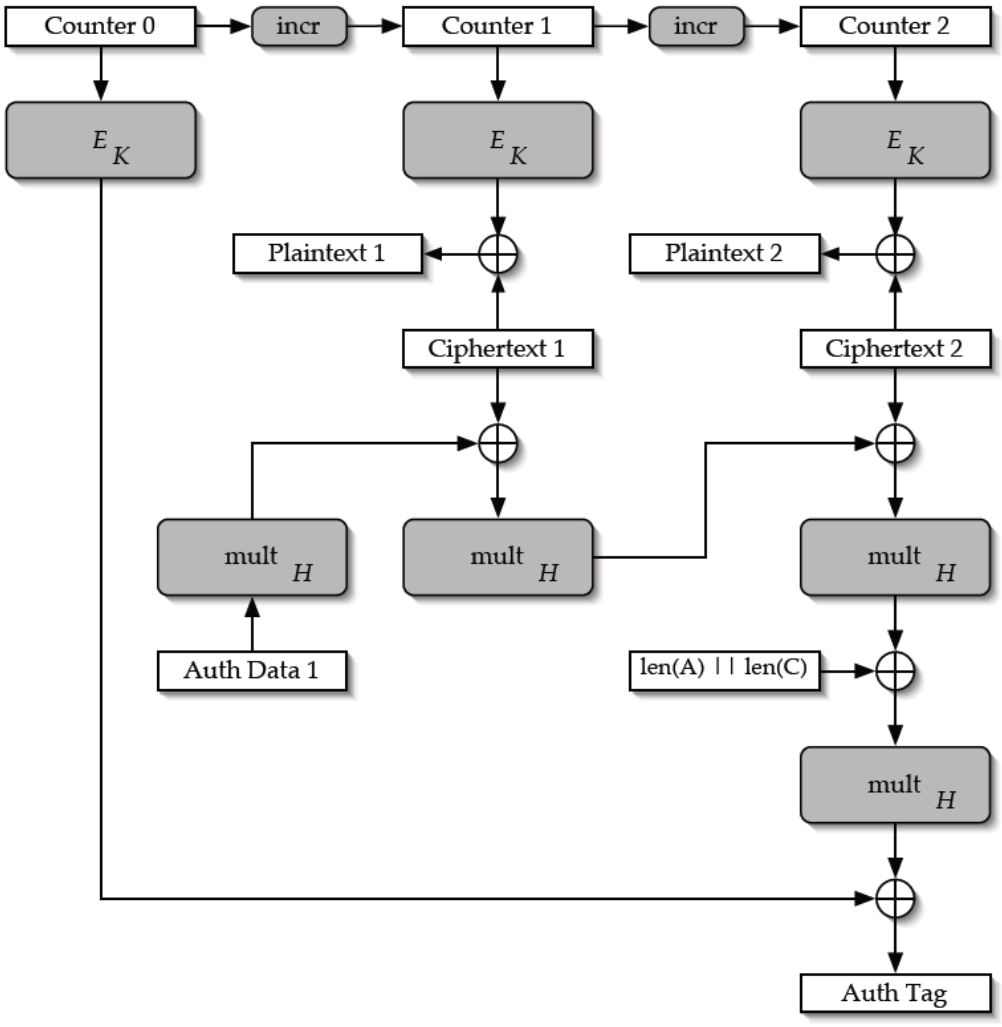


图 14-10 GCM 解密流程

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。

14.3.7 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14-11所示。

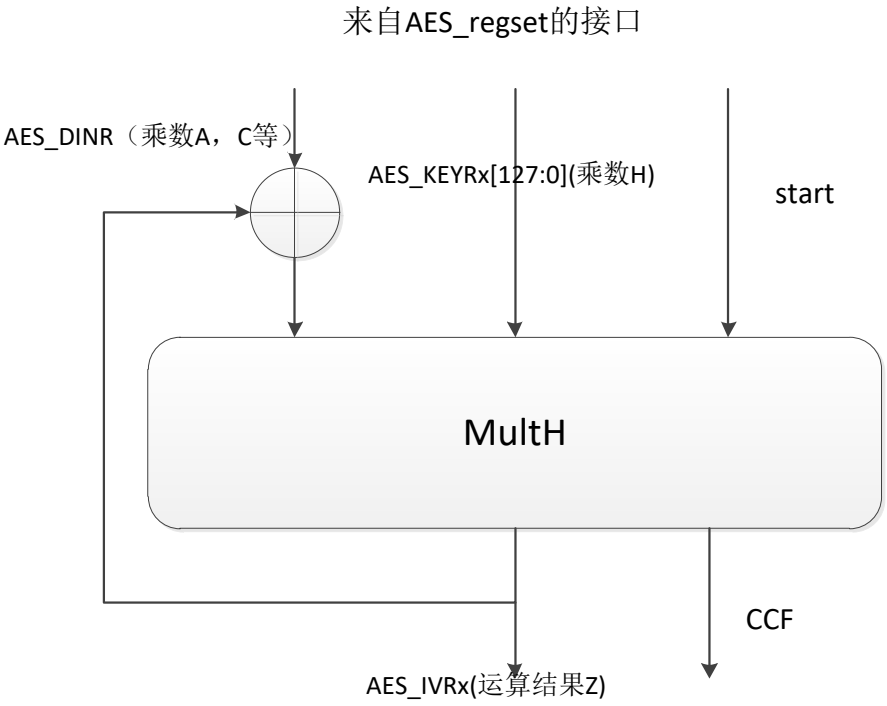


图 14-11 multH 模块框图

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。

14.3.8 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。

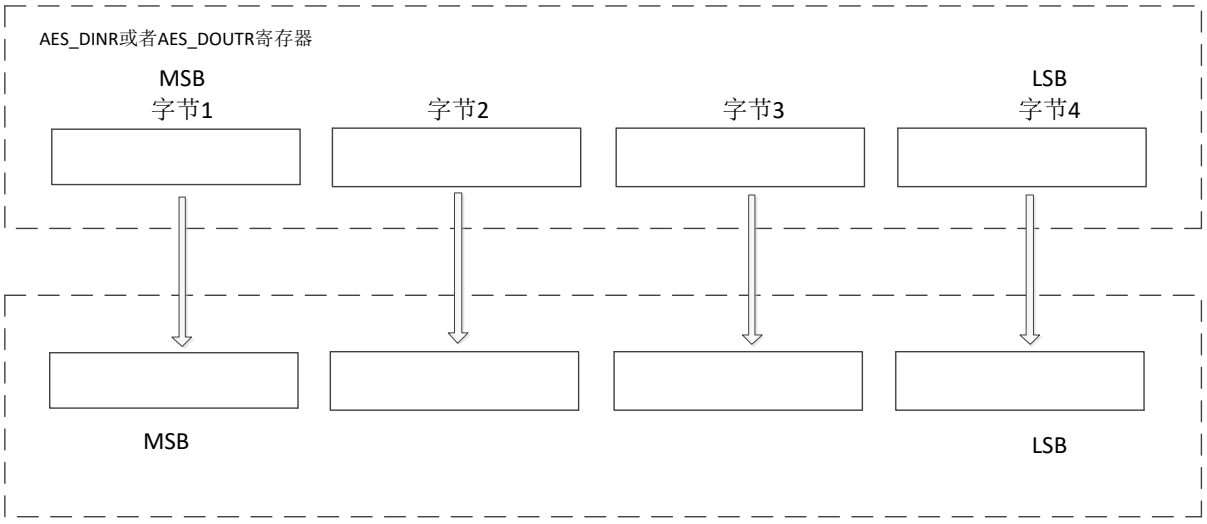
GCM模式的加解密过程和CTR模式相同。认证过程时仅使用MultH模块而不用AES加解密。

- 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。

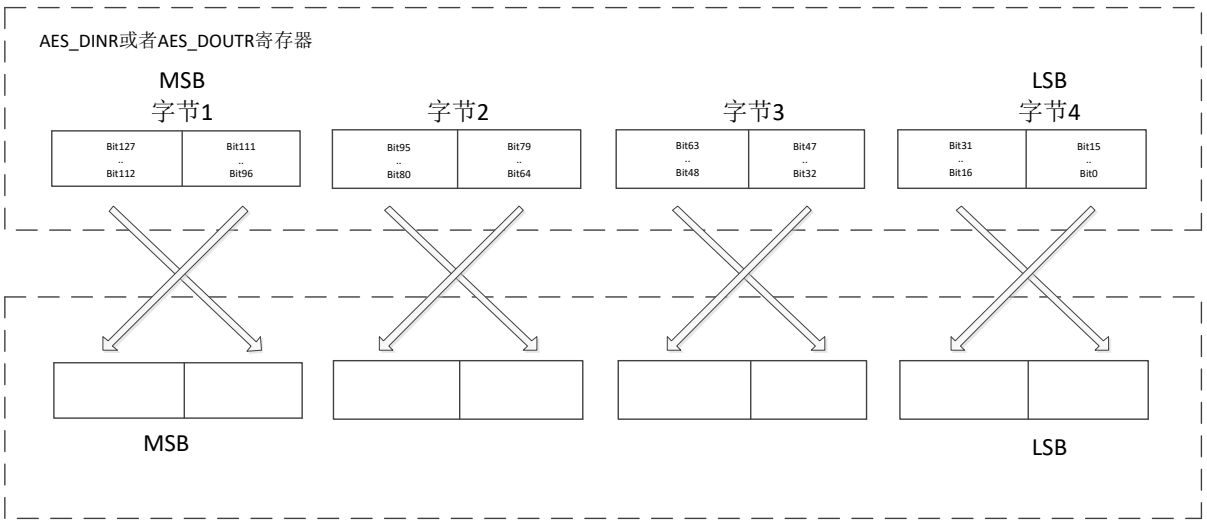
14.4 数据类型

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14-12所示。

Datatpye 2'b00 : 不交换



Datatpye 2'b01 : 半字交换



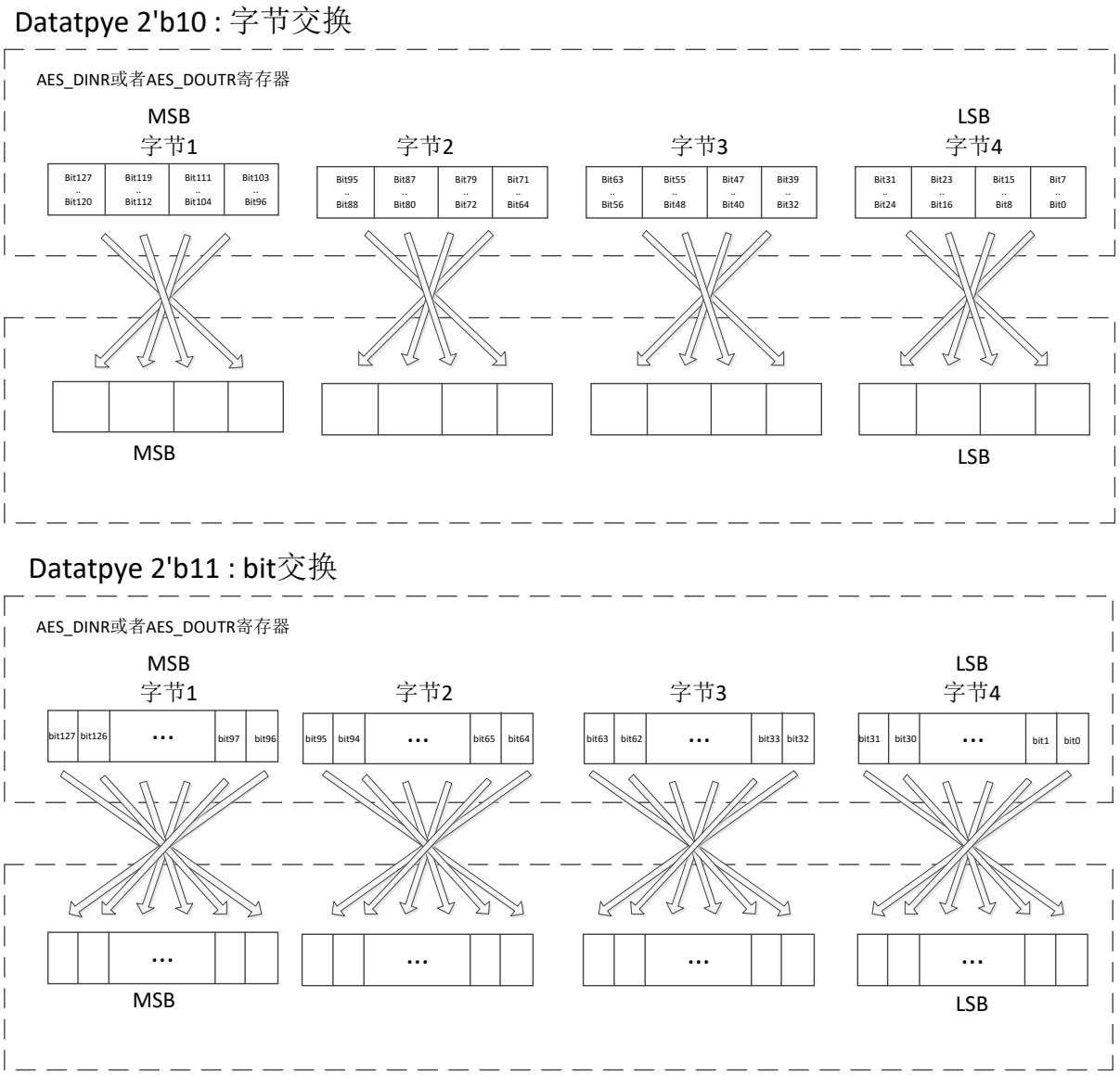


图 14-12 根据数据类型存储数据的示意图

14.5 工作流程

14.5.1 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_DINR 寄存器4次
- 等待CCF标志置起

- 从AES_DOUTR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。

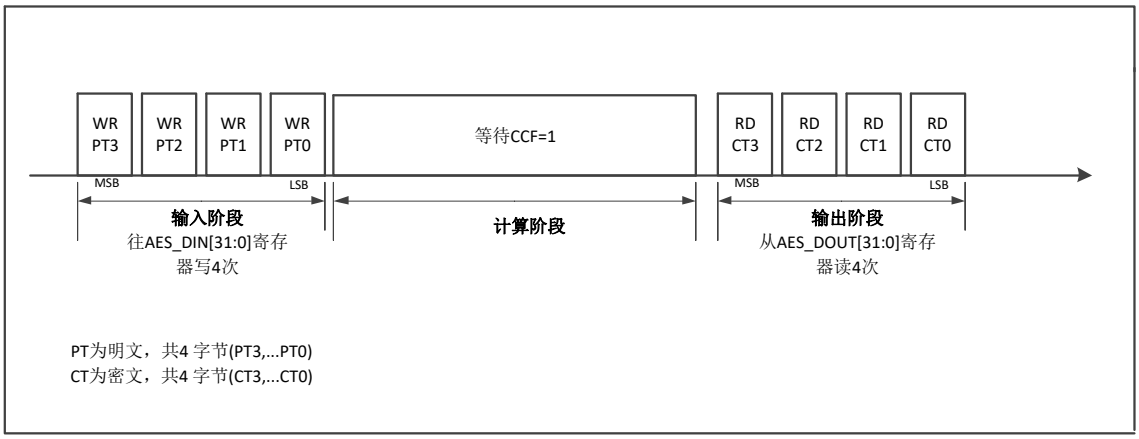


图 14-13 模式 1：加密流程

14.5.2 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_KEYRx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_KEYRx寄存器。想要重新计算扩展密钥，重复步骤3,4,5,6。

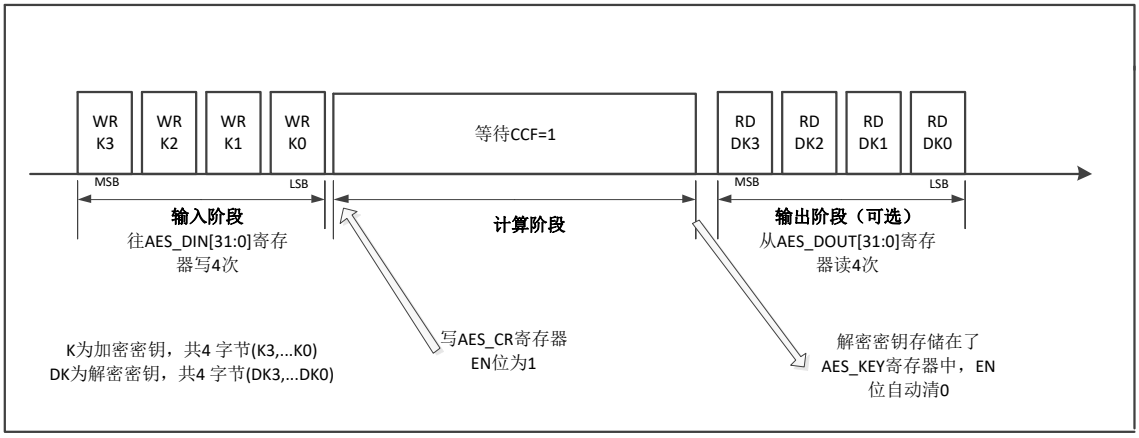


图 14-14 模式 2 示意图

14.5.3 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_KEYRx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_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR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

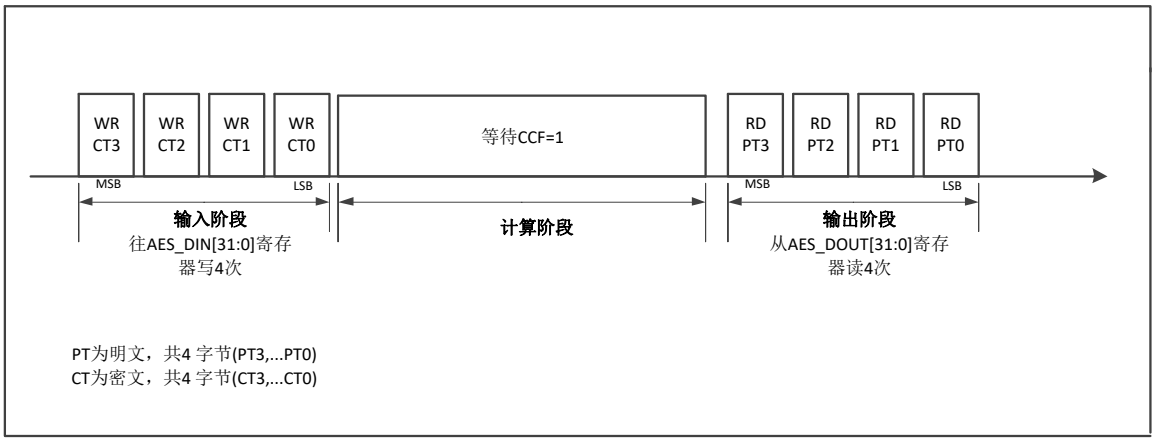


图 14-15 模式 3 示意图

14.5.4 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_KEYRx寄存器，CBC模式下写AES_IVRx寄存器。
- 写EN=1，使能AES
- 写AES_DINR 寄存器4次
- 等待CCF标志置起
- 从AES_DOUTR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_KEYRx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_KEYRx寄存器中。

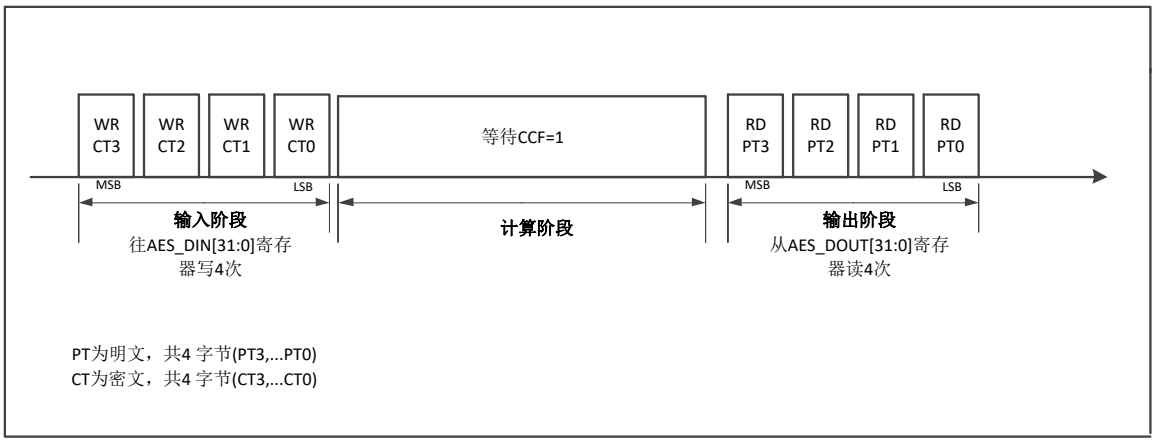


图 14-16 模式 4 示意图

14.5.5 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AESHx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_DINR 寄存器4次。MultH模块会把上一次的计算结果异或上AES_DINR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_DINR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。

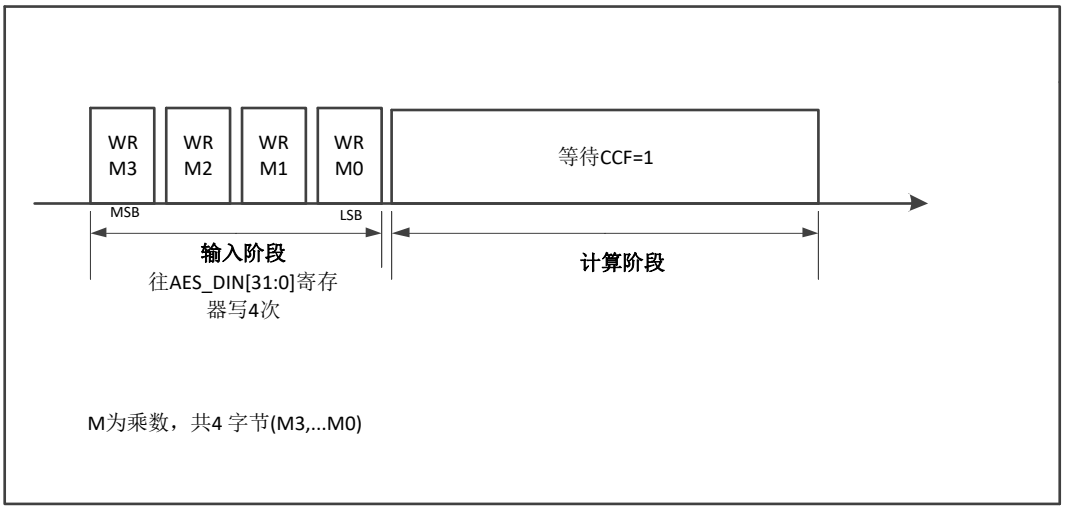


图 14-17 multH 模块使用流程示意图

14.6 DMA 接口

- 一个输入的请求通道：当DMAINEN为1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。
- 一个输入的请求通道：当DMAOUTEN为1时，每当AES在需要从AES_DOUTR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DMAOUTEN=1时，CCF标志可能为高。

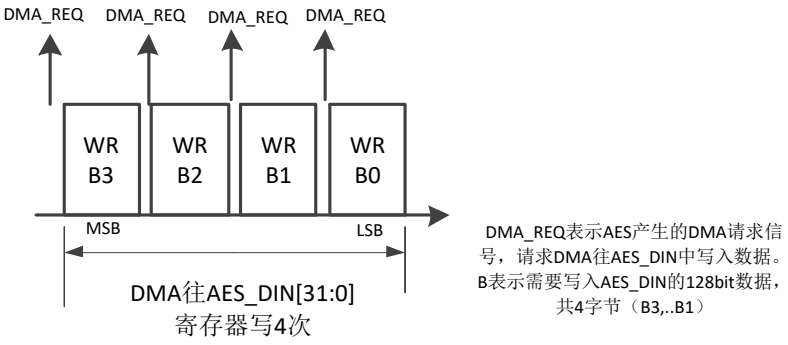


图 14-18 输入时 DMA 请求和数据传输示意图

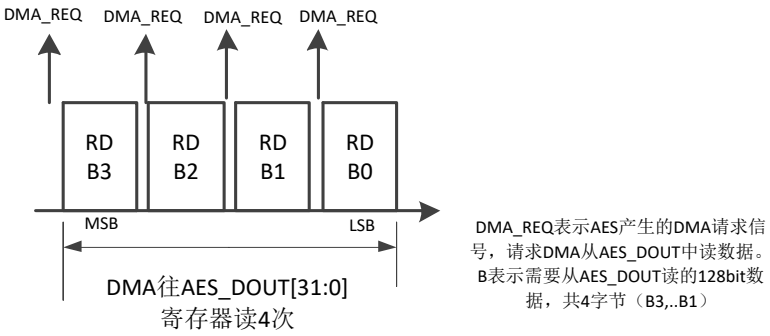


图 14-19 输出时 DMA 请求和数据传输示意图

14.6.1 MultH 模块与 DMA 间接口

MultH计算也可以通过DMA计算。当DMAINEN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DMAOUTEN=1无效AES不会产生DMA请求。

14.7 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。

14.8 寄存器

模块基地址：0x4001_3800

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

14.8.1 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

Bit	助记符	功能描述
		10：CTR 11：使用 MultH 模块
4:3	MODE	AES 工作模式，AESEN=1 时不可修改。 (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

14.8.2 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)

14.8.3 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/W1C-0	R/W1C-0	R/W1C-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

14.8.4 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

Bit	助记符	功能描述
		模式 3 和模式 4（解密）：把密文从 MSB 到 LSB 分 4 次写入。 MultH 模式：把乘数 A 或 C 从 MSB 到 LSB 分 4 次写入。

14.8.5 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_DOUTR 寄存器。

14.8.6 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	W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	KEYx[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	KEYx[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	KEYx[7:0]							
位权限	W-0000 0000							

Bit	助记符	功能描述

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，AESLKEY7 存放秘钥最高 32bit。(AES Key)

14.8.7 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)

14.8.8 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]

15 随机数发生器（TRNG）

15.1 概述

FM33LF0xxA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%。

15.2 功能描述

15.2.1 随机数产生

下图为真随机数发生器结构框图。

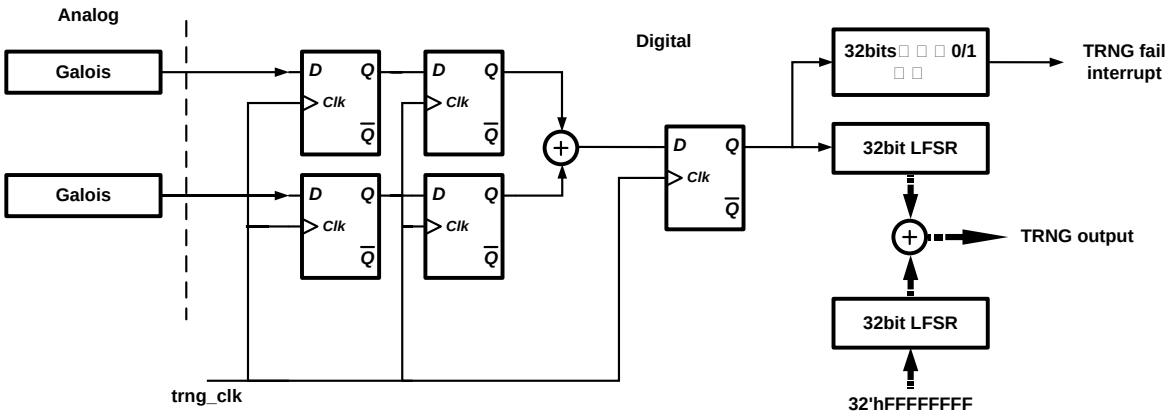


图 15-1 真随机数模块框图

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。

15.2.2 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：

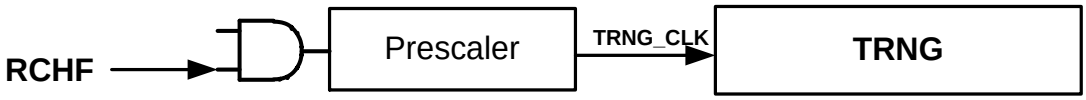


图 15-2 真随机数模块工作时钟

15.2.3 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的间隔不应小于8us。

15.2.4 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LFSR_BUSY，确认LFSR不在运行中
- 将待运算数据写入CRCDIR
- 置位CRCCR_EN
- 查询并等待CRC_EN被清零
- 从RNG_DOR读出运算结果

15.3 寄存器

模块基地址：0x4001_3C00

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	RNF 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

15.3.1 随机数控制寄存器(RNG_CR)

名称	RNG_CR							
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	-							RNGEN
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	RNGEN	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

15.3.2 随机数/CRC 结果输出寄存器(RNG_DOR)

名称	RNG_DOR							
地址	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RNGOUT[31:24]							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RNGOUT[23:16]							
位权限	R-00000000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RNGOUT[15:8]							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RNGOUT[7:0]							
位权限	R-00000000							

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

15.3.3 RNG 标志寄存器(RNG_SR)

名称	RNG_SR							
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	-						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

15.3.4 CRC 控制寄存器(RNG_CRCCR)

名称	RNG_CRCCR							
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	-							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

15.3.5 CRC 输入数据寄存器(RNG_CRCDIR)

名称	RNG_CRCDIR							
地址	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

15.3.6 CRC 标志寄存器(RNG_CRCSR)

名称	RNG_CRCSR							
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	-							
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
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

16 模拟比较器（Comparator）

16.1 概述

芯片集成4个模拟比较器，支持以下特性：

- 4个rail-to-rail快速比较器，支持轨到轨输入
- 可编程回滞窗口
- 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
  - DAC输出
  - OPA输出
- 中断事件可唤醒MCU
- 内建输入基准Buffer
- 输出信号可连接到GPIO，也可以作为触发源连接到定时器、ADC
- 可产生ATIM刹车或输出清零信号
- 可编程的输出数字滤波功能

16.2 结构框图

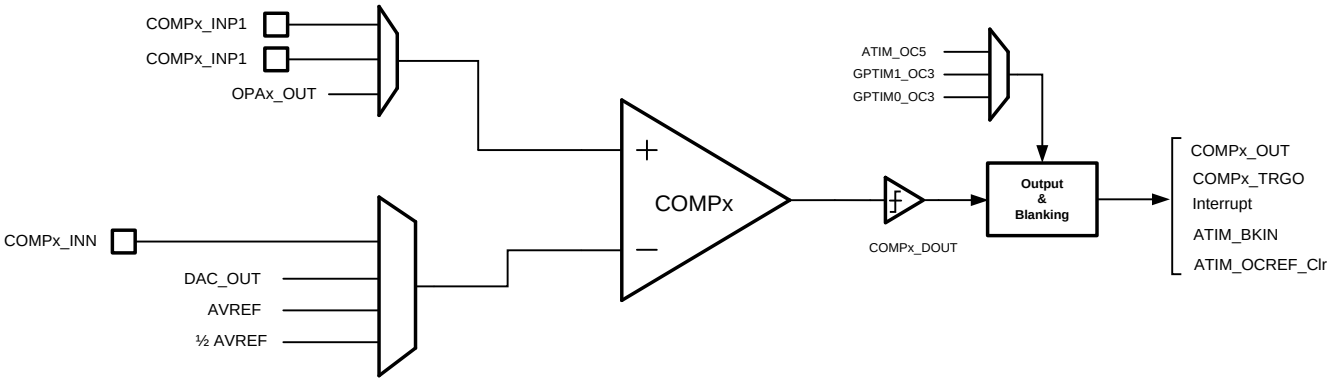


图 16-1 比较器电路框图

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。

比较器正端输入包括：管脚输入、OPA输出

比较器负端输入包括：管脚输入、DAC输出、基准电压

比较器输出包括：管脚输出、触发信号输出、中断、高级定时器刹车和清零

16.3 功能描述

16.3.1 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。

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。

16.3.2 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。

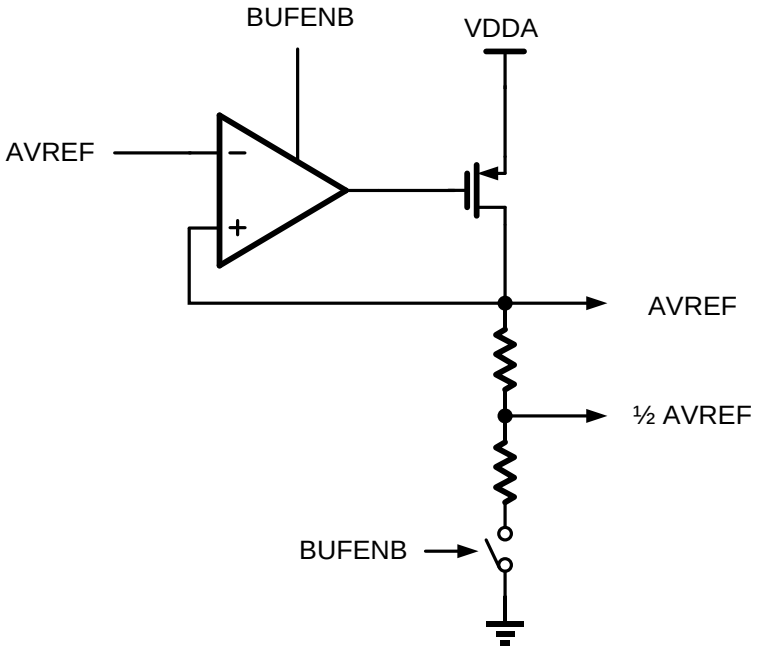


图 16-2 比较器内建基准缓冲器

16.3.3 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。

16.3.4 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	内部信号
COMP1_INP1	PD4	
COMP1_INP2	PD5	
COMP1_INP3		OPA1_OUT
COMP1_INN1	PA15	
COMP1_INN2		DAC_OUT
COMP1_INN3		AVREF
COMP1_INN4		1/2 AVREF
COMP1_OUT	PA10	-

表 16-1 比较器 1 引脚列表

COMP2	引脚	内部信号
COMP2_INP1	PD2	
COMP2_INP2	PA8	
COMP2_INP3		OPA2_OUT
COMP2_INN1	PD3	
COMP2_INN2		DAC_OUT
COMP2_INN3		AVREF
COMP2_INN4		1/2 AVREF
COMP2_OUT	PD3	-

表 16-2 比较器 2 引脚列表

COMP3	引脚	内部信号
COMP3_INP1	PD1	
COMP3_INP2	PA9	
COMP3_INP3		OPA1_OUT
COMP3_INN1	PB15	
COMP3_INN2		DAC_OUT
COMP3_INN3		AVREF
COMP3_INN4		1/2 AVREF
COMP3_OUT	PD4	-

表 16-3 比较器 3 引脚列表

COMP4	引脚	内部信号
COMP4_INP1	PD0	
COMP4_INP2	PA10	
COMP4_INP3		OPA2_OUT
COMP4_INN1	PC13	
COMP4_INN2		DAC_OUT
COMP4_INN3		AVREF
COMP4_INN4		1/2 AVREF
COMP4_OUT	PA15	-

表 16-4 比较器 4 引脚列表

16.3.5 比较器输出滞回

通过HYSTEN寄存器可以配置比较器是否使能回滞功能，实现原理如下图所示。注意当使能回滞功能时，反馈电阻可能对高阻抗输入信号造成影响，并且回滞窗口大小也与输入信号的内阻有关。

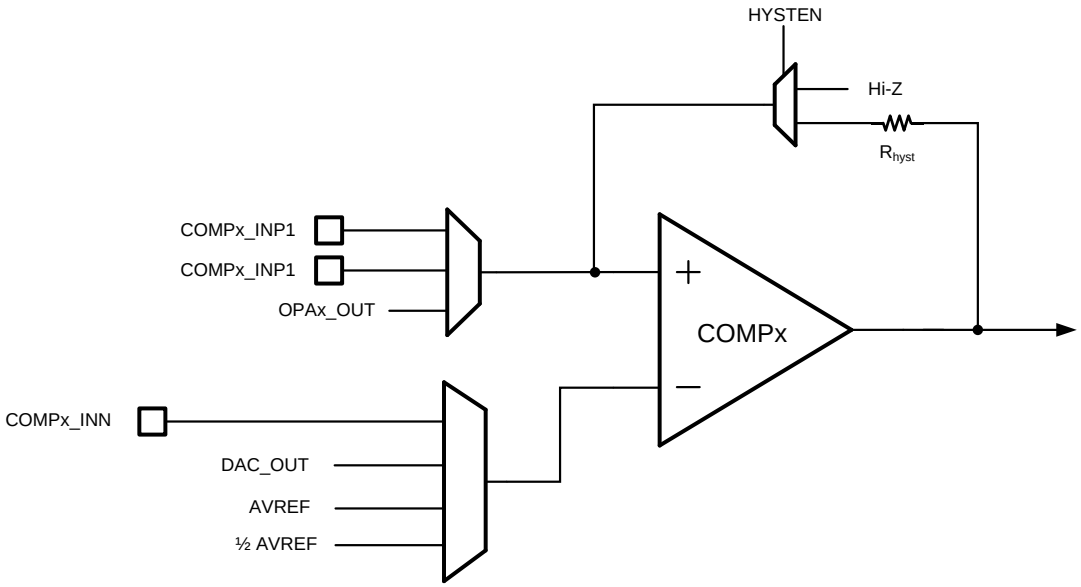


图 16-3 比较器输出滞回

16.3.6 输出屏蔽

输出屏蔽功能用于在特定PWM周期位置屏蔽比较器输出，用于防止开关短路时的瞬态电流脉冲导致比较器输出误动作。屏蔽信号使用部分定时器OC输出，连接关系参见寄存器定义。

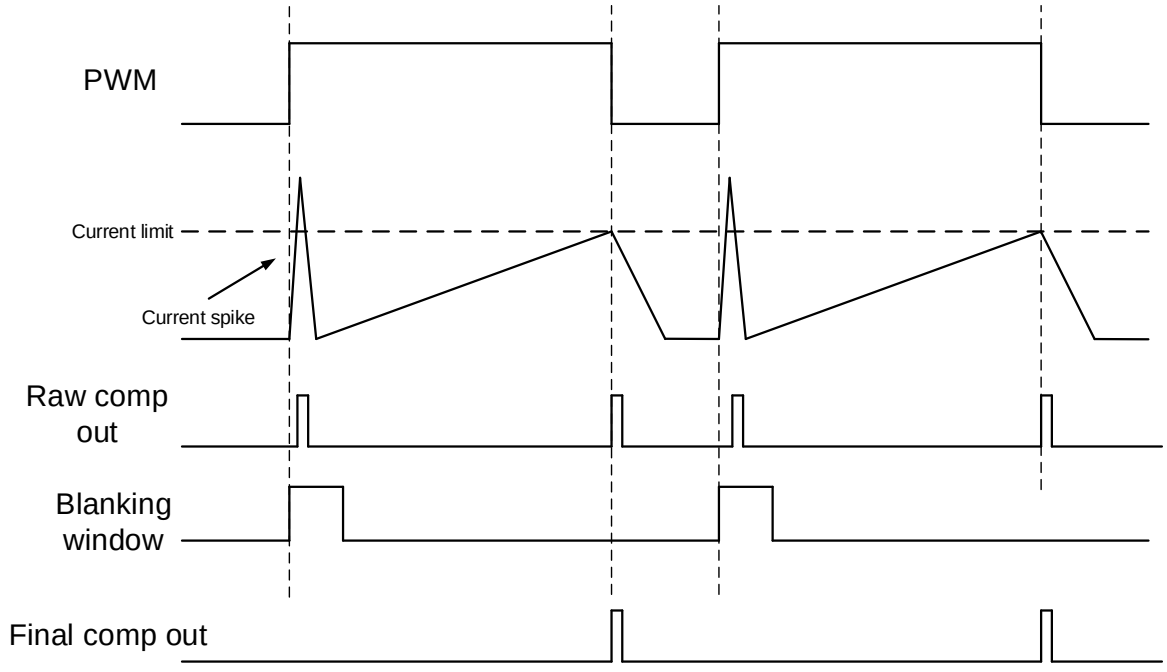


图 16-4 比较器输出屏蔽功能

上图中的blank window信号源由COMPx_CR.BLANKSEL寄存选择。

16.3.7 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，这个滤波长度最多可达32个APBCLK时钟周期。

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：

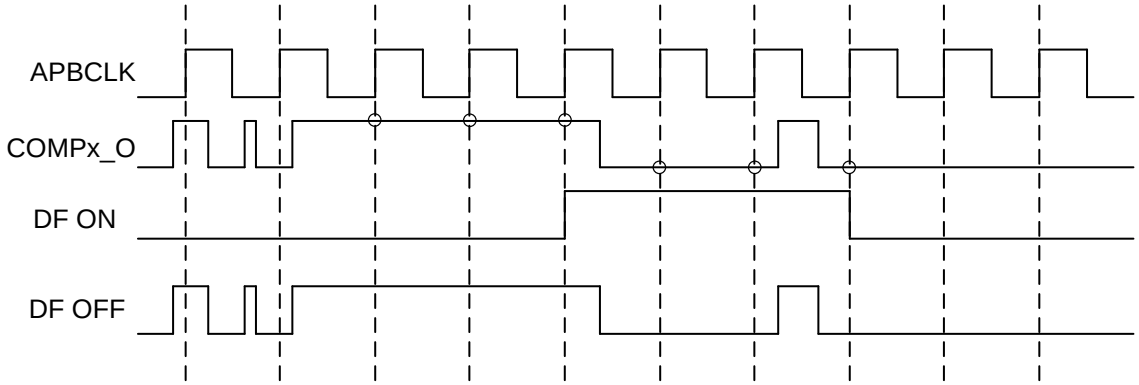


图 16-5 数字滤波（DFLEN=3）波形示意图

新增的滤波扩展配置功能DFLEN_EX寄存器，可将源滤波长度可选扩展2倍或者4倍，即扩展后的滤波最多可达128个APBCLK周期。

16.3.8 功耗和速度模式

比较器支持轨到轨输入，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
快速模式		0.2us	5us	
低功耗模式	<2uA	2us	15us	

表 16-5 比较器工作模式

16.3.9 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。CMPx_ICR寄存器可以使能或禁止中断输出。CMPx_IF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。

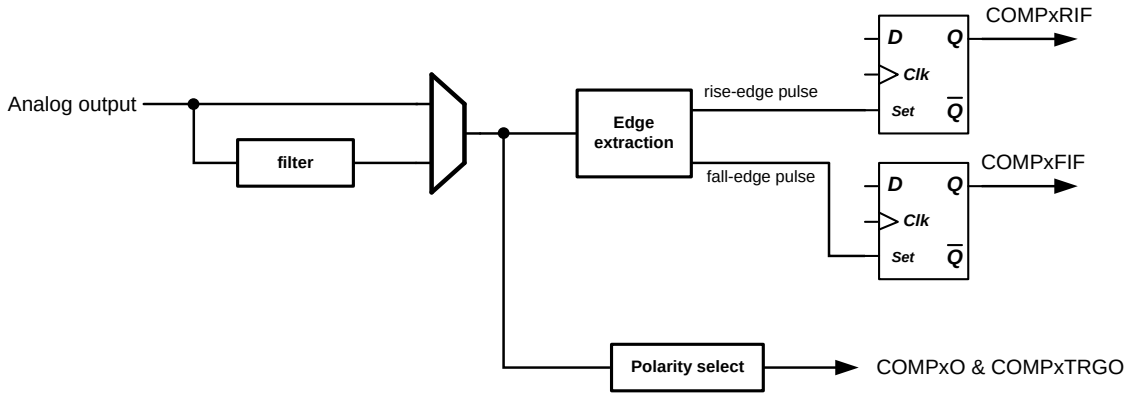


图 16-6 比较器中断产生

16.3.10 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。

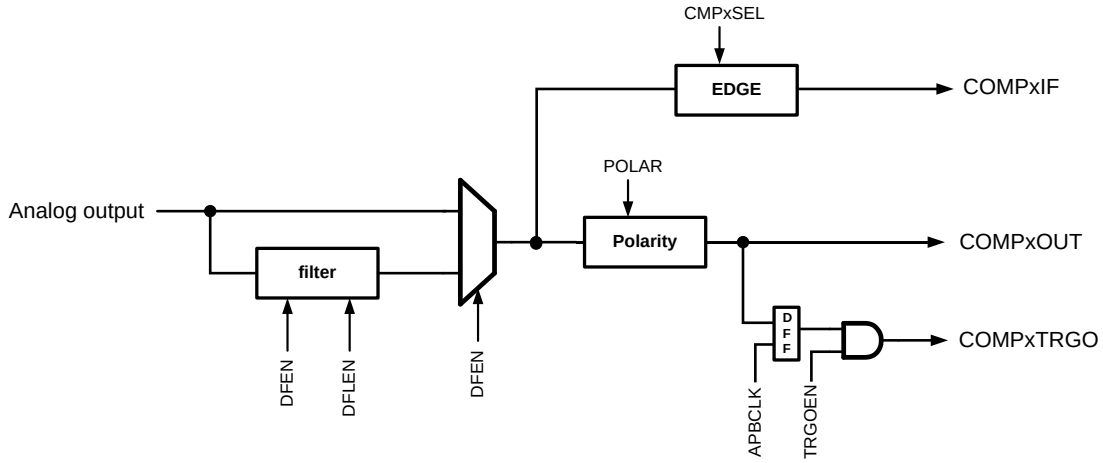


图 16-7 比较器输出逻辑

比较器输出

比较器比较结果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。

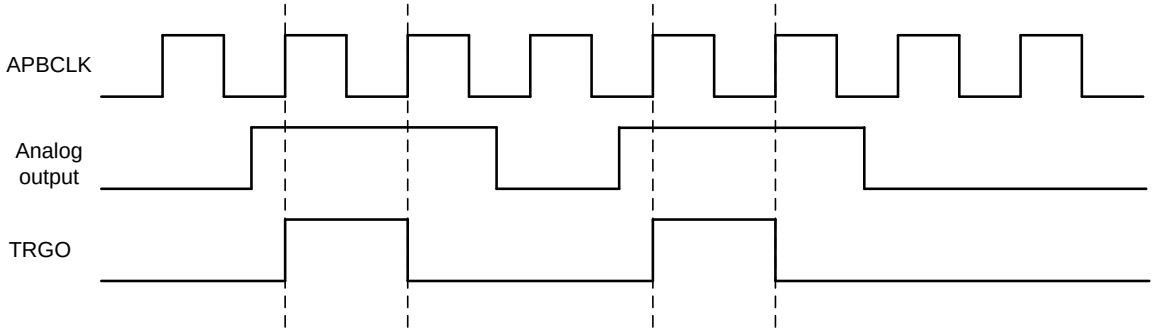


图 16-8 比较器输出上升沿产生触发输出，无滤波

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。

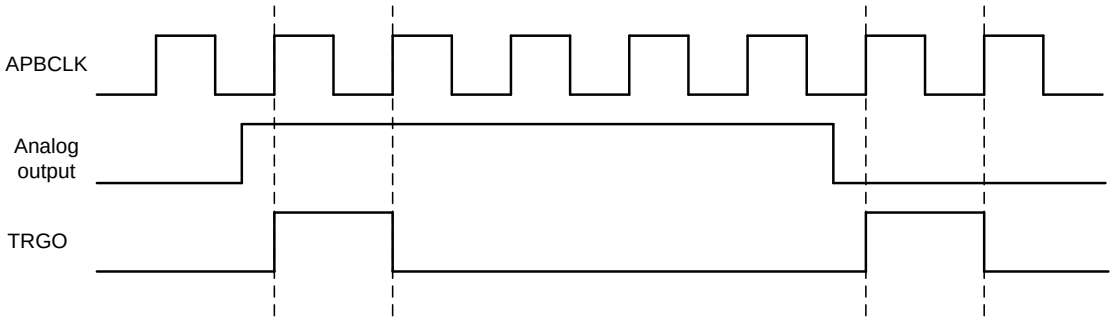


图 16-9 比较器输出上升沿下降沿产生触发输出，无滤波

16.4 寄存器

模块基地址：0x4001_5400

offset 地址	名称	符号
0x00	COMP1 控制寄存器	COMP1_CR
0x04	COMP2 控制寄存器	COMP2_CR
0x08	COMP3 控制寄存器	COMP3_CR
0x0C	COMP4 控制寄存器	COMP4_CR
0x10	COMP 中断配置寄存器	COMP_ICR
0x14	COMP 中断标志寄存器	COMP_ISR
0x18	COMP 缓冲器控制寄存器	COMP_BUFCR

16.4.1 COMP1 控制寄存器（COMP1_CR）

名称	COMP1_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK	-		DFLEN_EX		TRGOSEL		TRGOEN
位权限	R/W-0	U-0		R/W-00		R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					HYSTEN	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		-		BLANKSEL			CMPO
位权限	R/W-00		U-0		R/W-000			R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		PSEL		NSEL			EN
位权限	U-0		R/W-00		R/W-00			R/W-0

Bit	助记符	功能描述
31	LOCK	寄存器锁定。此 bit 置位后软件不能清零，只有系统复位才能清零。 0：COMP1_CR 寄存器可以读写 1：COMP1_CR 寄存器只读，无法改写
30:29	--	RFU：未实现，读为 0
28:27	DFLEN_EX	输出数字滤波长度扩展配置寄存器。滤波长度周期对原先 DFLEN 规定的长度进行扩展 00：原先 DFLEN 定义长度 01：原先 DFLEN 定义长度*2 10/11：原先 DFLEN 定义长度*4
26:25	TRGOSEL	比较器 1 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 1 触发信号输出使能

Bit	助记符	功能描述
		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
18	HYSTEN	回滞功能使能 0：关闭回滞窗口 1：使能回滞窗口
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	比较器功耗模式选择 01：低功耗模式 10：快速模式 00：RFU 11：RFU
13:12	--	RFU：未实现，读为 0
11:9	BLANKSEL	COMP1 输出屏蔽源控制 000：无屏蔽 001：ATIM_OC5 010：GPTIM0_OC3 011：GPTIM1_OC3 其他：RFU
8	CMPO	比较器 1 输出，软件只读
7:5	--	RFU：未实现，读为 0
4	PSEL	比较器 1 正极输入选择 00：COMP1_INP1 01：COMP1_INP2 10：OPA1_OUT 11：RFU
3:1	NSEL	比较器 1 负极输入选择 000：COMP1_INN 001：DAC_OUT 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	EN	比较器 1 使能位 0：关闭

Bit	助记符	功能描述
		1：使能

16.4.2 COMP2 控制寄存器（COMP2_CR）

名称	COMP2_CR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK	-		DFLEN_EX		TRGOSEL		TRGOEN
位权限	R/W-0	U-0		R/W-00		R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					HYSTEN	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		-		BLANKSEL			CMPO
位权限	R/W-00		U-0		R/W-000			R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		PSEL		NSEL			EN
位权限	U-0		R/W-00		R/W-00			R/W-0

Bit	助记符	功能描述
31	LOCK	寄存器锁定。此 bit 置位后软件不能清零，只有系统复位才能清零。 0：COMP2_CR 寄存器可以读写 1：COMP2_CR 寄存器只读，无法改写
30:29	--	RFU：未实现，读为 0
28:27	DFLEN_EX	输出数字滤波长度扩展配置寄存器。滤波长度周期对原先 DFLEN 规定的长度进行扩展 00：原先 DFLEN 定义长度 01：原先 DFLEN 定义长度*2 10/11：原先 DFLEN 定义长度*4
26:25	TRGOSEL	比较器 2 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 2 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（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	HYSTEN	回滞功能使能 0：关闭回滞窗口 1：使能回滞窗口
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	MODE	比较器功耗模式选择 01：低功耗模式 10：快速模式 00：RFU 11：RFU
13:12	--	RFU：未实现，读为 0
11:9	BLANKSEL	COMP2 输出屏蔽源控制 000：无屏蔽 001：ATIM_OC5 010：GPTIM0_OC3 011：GPTIM1_OC3 其他：RFU
8	CMPO	比较器 2 输出，软件只读
7:5	--	RFU：未实现，读为 0
4	PSEL	比较器 2 正极输入选择 00：COMP2_INP1 01：COMP2_INP2 10：OPA2_OUT 11：RFU
3:1	NSEL	比较器 2 负极输入选择 000：COMP2_INN 001：DAC_OUT 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	EN	比较器 2 使能位 0：关闭 1：使能

16.4.3 COMP3 控制寄存器（COMP3_CR）

名称	COMP3_CR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK	-		DFLEN_EX		TRGOSEL		TRGOEN
位权限	R/W-0	U-0		R/W-00		R/W-00		R/W-0

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					HYSTEN	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		-		BLANKSEL			CMPO
位权限	R/W-00		U-0		R/W-000			R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		PSEL		NSEL			EN
位权限	U-0		R/W-00		R/W-00			R/W-0

Bit	助记符	功能描述
31	LOCK	寄存器锁定。此 bit 置位后软件不能清零，只有系统复位才能清零。 0：COMP3_CR 寄存器可以读写 1：COMP3_CR 寄存器只读，无法改写
30:29	--	RFU：未实现，读为 0
28:27	DFLEN_EX	输出数字滤波长度扩展配置寄存器。滤波长度周期对原先 DFLEN 规定的长度进行扩展 00：原先 DFLEN 定义长度 01：原先 DFLEN 定义长度*2 10/11：原先 DFLEN 定义长度*4
26:25	TRGOSEL	比较器 3 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 3 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 3 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
18	HYSTEN	回滞功能使能 0：关闭回滞窗口 1：使能回滞窗口
17	POLAR	比较器 3 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 3 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	比较器功耗模式选择 01：低功耗模式 10：快速模式

Bit	助记符	功能描述
		00：RFU 11：RFU
13:12	--	RFU：未实现，读为 0
11:9	BLANKSEL	COMP3 输出屏蔽源控制 000：无屏蔽 001：ATIM_OC5 010：GPTIM0_OC3 011：GPTIM1_OC3 其他：RFU
8	CMPO	比较器 3 输出，软件只读
7:5	--	RFU：未实现，读为 0
4	PSEL	比较器 3 正极输入选择 00：COMP3_INP1 01：COMP3_INP2 10：OPA1_OUT 11：RFU
3:1	NSEL	比较器 3 负极输入选择 000：COMP3_INN 001：DAC_OUT 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	EN	比较器 3 使能位 0：关闭 1：使能

16.4.4 COMP4 控制寄存器（COMP4_CR）

名称	COMP4_CR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK	-		DFLEN_EX		TRGOSEL		TRGOEN
位权限	R/W-0	U-0		R/W-00		R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					HYSTEN	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		-		BLANKSEL			CMPO
位权限	R/W-00		U-0		R/W-000			R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		PSEL		NSEL			EN
位权限	U-0		R/W-00		R/W-00			R/W-0

Bit	助记符	功能描述
31	LOCK	寄存器锁定。此 bit 置位后软件不能清零，只有系统复位才能清零。 0：COMP4_CR 寄存器可以读写 1：COMP4_CR 寄存器只读，无法改写
30:29	--	RFU：未实现，读为 0
28:27	DFLEN_EX	输出数字滤波长度扩展配置寄存器。滤波长度周期对原先 DFLEN 规定的长度进行扩展 00：原先 DFLEN 定义长度 01：原先 DFLEN 定义长度*2 10/11：原先 DFLEN 定义长度*4
26:25	TRGOSEL	比较器 4 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 4 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 4 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
18	HYSTEN	回滞功能使能 0：关闭回滞窗口 1：使能回滞窗口
17	POLAR	比较器 4 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 4 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	比较器功耗模式选择 01：低功耗模式 10：快速模式 00：RFU 11：RFU
13:12	--	RFU：未实现，读为 0
11:9	BLANKSEL	COMP4 输出屏蔽源控制 000：无屏蔽 001：ATIM_OC5 010：GPTIM0_OC3 011：GPTIM1_OC3 其他：RFU
8	CMPO	比较器 4 输出，软件只读
7:5	--	RFU：未实现，读为 0

Bit	助记符	功能描述
4	PSEL	比较器 4 正极输入选择 00：COMP4_INP1 01：COMP4_INP2 10：OPA2_OUT 11：RFU
3:1	NSEL	比较器 4 负极输入选择 000：COMP4_INN 001：DAC_OUT 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	EN	比较器 4 使能位 0：关闭 1：使能

16.4.5 COMP 中断配置寄存器（COMP_ICR）

名称	COMP_ICR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				CMP4SEL		-	CMP4IE
位权限	U-0				R/W-00		U-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				CMP3SEL		-	CMP3IE
位权限	U-0				R/W-00		U-0	R/W-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
31:28	--	RFU：未实现，读为 0
27:26	CMP4SEL	比较器 4 中断源选择 00/11：比较器 4 输出上升或下降沿产生中断 01：比较器 4 输出上升沿产生中断 10：比较器 4 输出下降沿产生中断
25	--	RFU：未实现，读为 0
24	CMP4IE	比较器 4 中断使能，1 有效
23:20	--	RFU：未实现，读为 0
19:18	CMP3SEL	比较器 3 中断源选择 00/11：比较器 3 输出上升或下降沿产生中断 01：比较器 3 输出上升沿产生中断

Bit	助记符	功能描述
		10：比较器 3 输出下降沿产生中断
17	--	RFU：未实现，读为 0
16	CMP3IE	比较器 3 中断使能，1 有效
15:12	--	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。

16.4.6 COMP 中断标志寄存器（COMP_IF）

名称	COMP_IF							
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	-				CMP4IF	CMP3IF	CMP2IF	CMP1IF
位权限	U-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	CMP4IF	比较器 4 中断标志，硬件置位，软件写 1 清零
2	CMP3IF	比较器 3 中断标志，硬件置位，软件写 1 清零
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

16.4.7 COMP 缓冲器控制寄存器（COMP_BUFCR）

名称	COMP_BUFCR

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	-					BUFBYP	-	BUFENB
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	BUFBYP	缓冲器 bypass 使能 0：不 bypass 1：bypass
1	--	RFU：未实现，读为 0
0	BUFENB	缓冲器使能 0：关闭基准缓冲器 1：使能基准缓冲器

17 除法/开方运算加速器（DIVAS）

17.1 概述

DIVAS模块用于帮助软件加速除法和开方运算。该硬件电路包含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16bit余数；以及一个无符号整数开方电路，输入32bit无符号数，输出16bit开方运算结果。

基本特性：

- 有符号整数除法运算（二进制补码格式）
- 32bit被除数、16bit除数
- 32bit商和16bit余数
- 除以0警告
- 一次除法计算需要5 cycles@72MHz
- 无符号整数开方运算
- 32bit被开方数，16bit结果
- 一次开方运算需要16个时钟周期@72MHz

17.2 时钟和复位

DIVAS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DIVAS模块寄存器之前必须在RMU模块中清除DIVAS复位（DIVASRST），并在CMU中使能DIVAS时钟（DIVAS）。

17.3 硬件除法工作流程

软件按照如下步骤调用硬件除法器。

- 清零MODE寄存器
- 配置DIV_TIME寄存器
- 向OPRD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）
- 写入DIVSOR后硬件除法器自动开始工作，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

17.4 硬件开方工作流程

软件按照如下步骤调用硬件开方器。

- 置位MODE寄存器
- 配置SQR_TIME寄存器
- 向OPRD寄存器写入32bit被开方数
- 硬件开方器自动开始工作，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
- 读取ROOT寄存器中的根

17.5 计算时间配置

根据不同APBCLK频率，软件应配置合理的开方和除法计算时间（APBCLK时钟周期数）：

运算	APBCLK (MHz)	推荐时钟周期数
除法	8	1
	16	1
	24	2
	32	2
	48	3
	64	4
	72	5
开方	8	2
	16	4
	24	8
	32	8
	48	12
	64	16
	72	16

注意：软件在启动计算前，应先合理配置计算时间；当系统频率确定后，计算时间寄存器仅需设置一次。不合理的计算时间设置可能导致错误的计算结果。硬件计算开始后，DIVAS模块等待配置的计算时间，然后清零BUSY标志表示计算完成。

17.6 寄存器

模块基地址：0x40019C00

offset 地址	名称	符号
DIVAS(模块基地址：0x40019C00)		
0x00	操作数寄存器 （Operand Register）	DIVAS_OPRD
0x04	除数寄存器 （Divisor Regsiter）	DIVAS_DIVSOR
0x08	商寄存器 （Quotient Register）	DIVAS_QUOT
0x0C	余数寄存器 （Reminder Register）	DIVAS_REMD
0x10	根寄存器 （Root Register）	DIVAS_ROOT
0x14	状态标志寄存器 （Status Register）	DIVAS_SR
0x18	控制寄存器 （Control Register）	DIVAS_CR

17.6.1 操作数寄存器(DIVAS_OPRD)

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

位号	位名	说明
31:0	OPRD	操作数寄存器 在除法运算时保存 32bit 有符号被除数 在开方运算时保存 32bit 无符号被开方数

17.6.2 除数寄存器(DIVAS_DIVSOR)

名称	DIVAS_DIVSOR							
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

17.6.3 商寄存器(DIVAS_QUOT)

名称	DIVAS_QUOT							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QUOT[31:24]							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QUOT [23:16]							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QUOT [15:8]							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QUOT [7:0]							
位权限	R-00000000							

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

17.6.4 余数寄存器(DIVAS_REMD)

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-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REMD[7:0]							
位权限	R-00000000							

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，读取时直接返回除法模块输出）

17.6.5 根寄存器(DIVAS_ROOT)

名称	DIVAS_ROOT							
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	ROOT[15:8]							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ROOT[7:0]							
位权限	R-00000000							

位号	位名	说明
31:16	--	RFU：未实现，读为 0
15:0	ROOT	16bit 无符号开方根 （仅地址，无实际寄存器，读取时直接返回开方模块输出）

17.6.6 状态标志寄存器(DIVAS_SR)

名称	DIVAS_SR							
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	-						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 = DIVAS 在计算过程中，结果未就绪 0 = 计算完毕，结果就绪 软件在写入操作数或除数后，DIVAS 开始计算，软件应查询 BUSY 为低后再读取商、余数或根寄存器 **单次计算所需的时钟周期数由 SQR_TIME 和 DIV_TIME 寄存器定义**

17.6.7 控制寄存器(DIVAS_CR)

名称	DIVAS_CR							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SQR_TIME				DIV_TIME			
位权限	R/W-1111				R/W-0100			
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	-							MODE
位权限	U-0							R/W-0

位号	位名	说明
31:28	SQR_TIME	开方运算周期 0000:1 clock cycle 0001: 2 clock cycles .. 1111: 16 clock cycles
27:24	DIV_TIME	除法运算周期 0000:1 clock cycle

位号	位名	说明
		0001: 2 clock cycles .. 1111: 16 clock cycles
23:1	--	RFU：未实现，读为 0
0	MODE	工作模式控制 0：硬件除法器 1：硬件开方器

18 双线串行总线和系统管理总线（I²C-SMBus）

18.1 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。

18.2 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

18.3 模块框图

18.3.1 I2C-SMB 模块框图

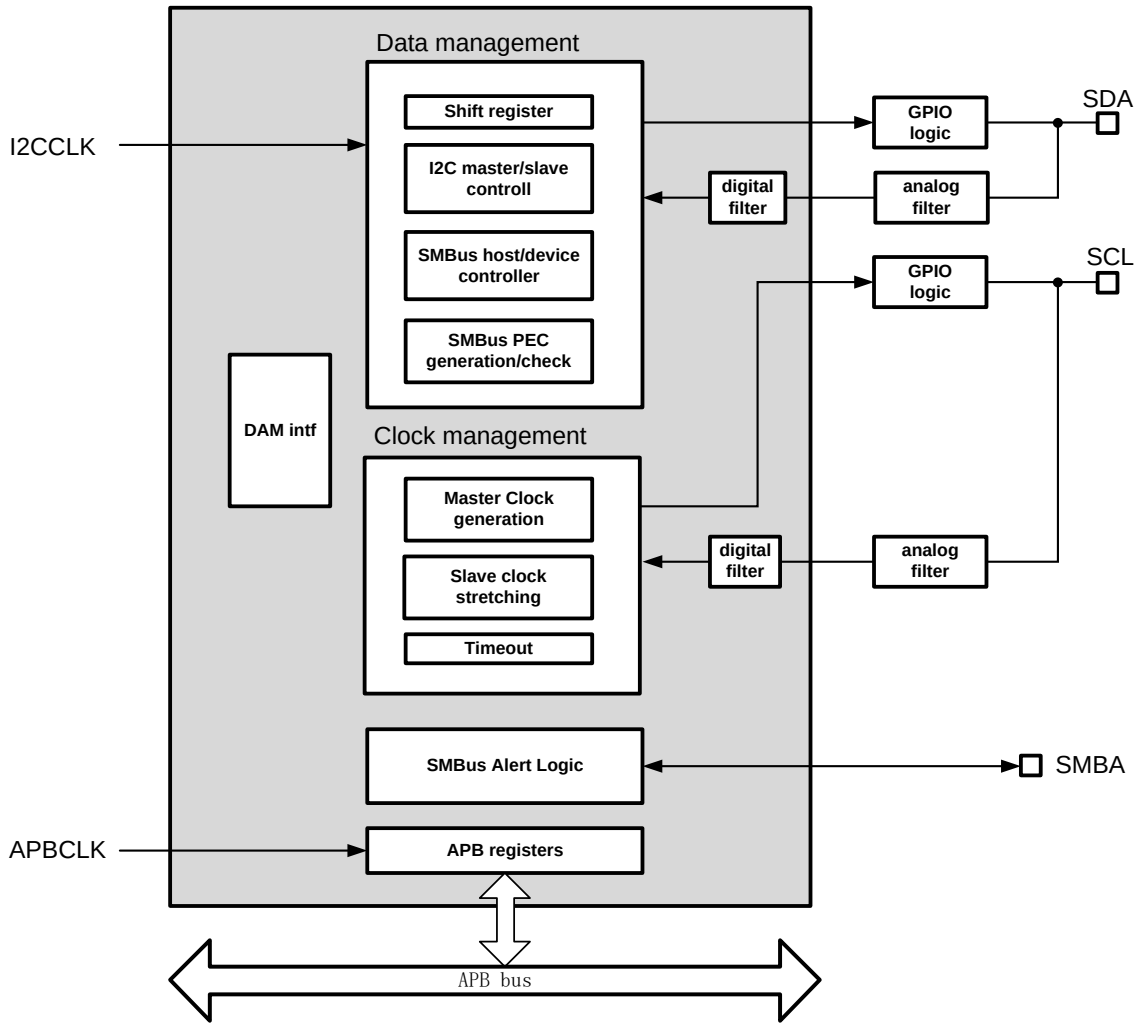


图 18-1 I²C-SMB 模块框图

18.4 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
PC4,PA11	I2C_SMB0	SCLA0	串行时钟
PA12,PC5		SDAA0	串行数据
PC6		SMB0_ALERTB	SMBus Alert#
PB15,PB0	I2C_SMB1	SCLA1	串行时钟
PB1,PC13		SDAA1	串行数据
PC7		SMB1_ALERTB	SMBus Alert#

表 18-1 I²C 引脚列表

18.5 时钟和复位

I²C 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、RCLF，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 I²C 复位寄存器（I2CRST）。

18.6 接口时序

18.6.1 通信过程

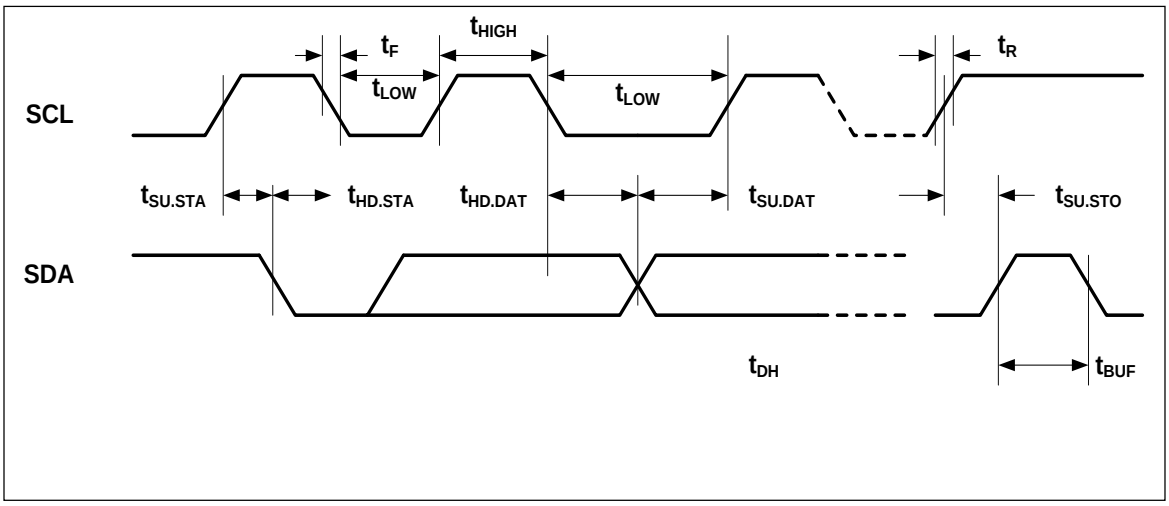


图 18-2 I²C 总线时序

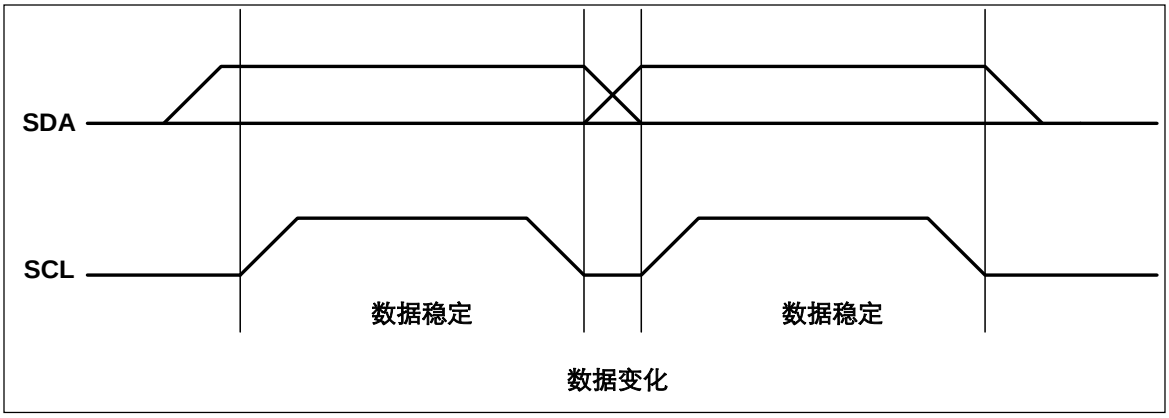


图 18-3 数据有效时序

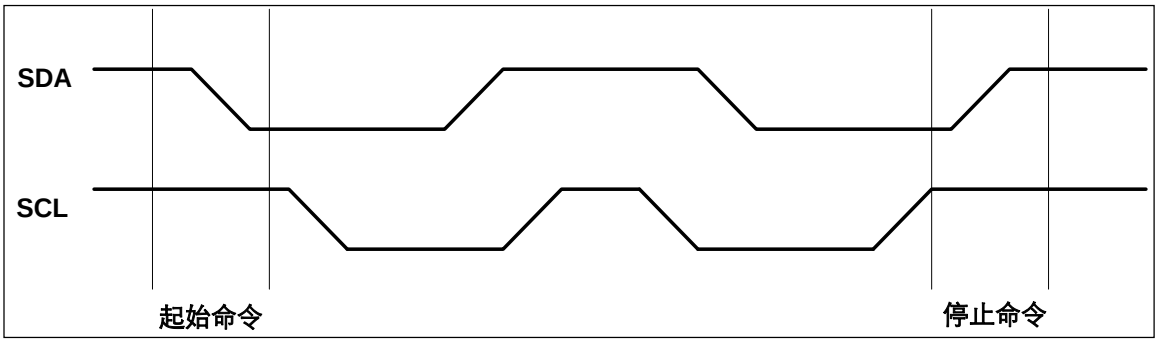


图 18-4 起始（Start）与停止(Stop)命令定义

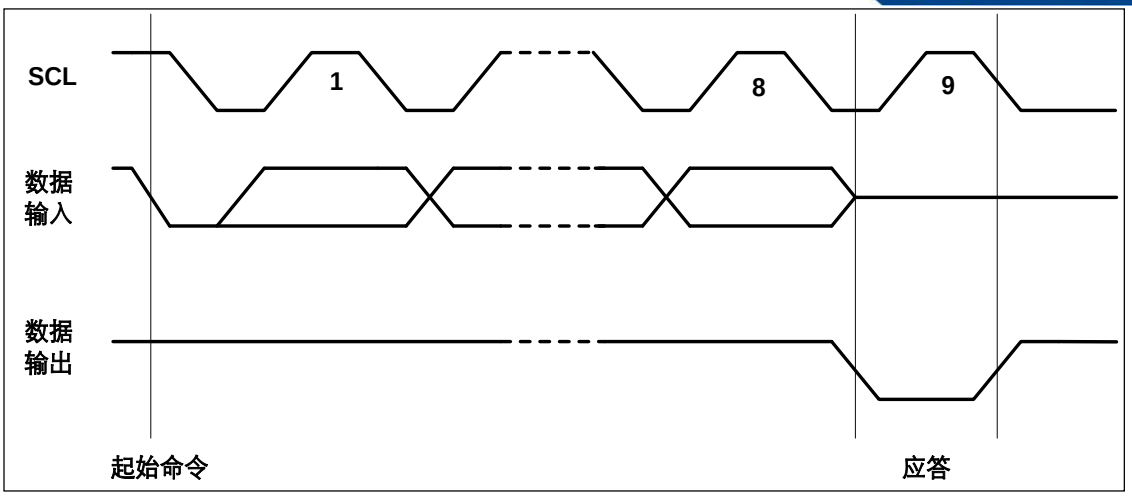


图 18-5 输出应答(ACK)

18.6.2 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 18-3）；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 18-4）。

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 18-4）。

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 18-5）。

18.6.3 总线时序参数表

I^2C 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 18-2 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。

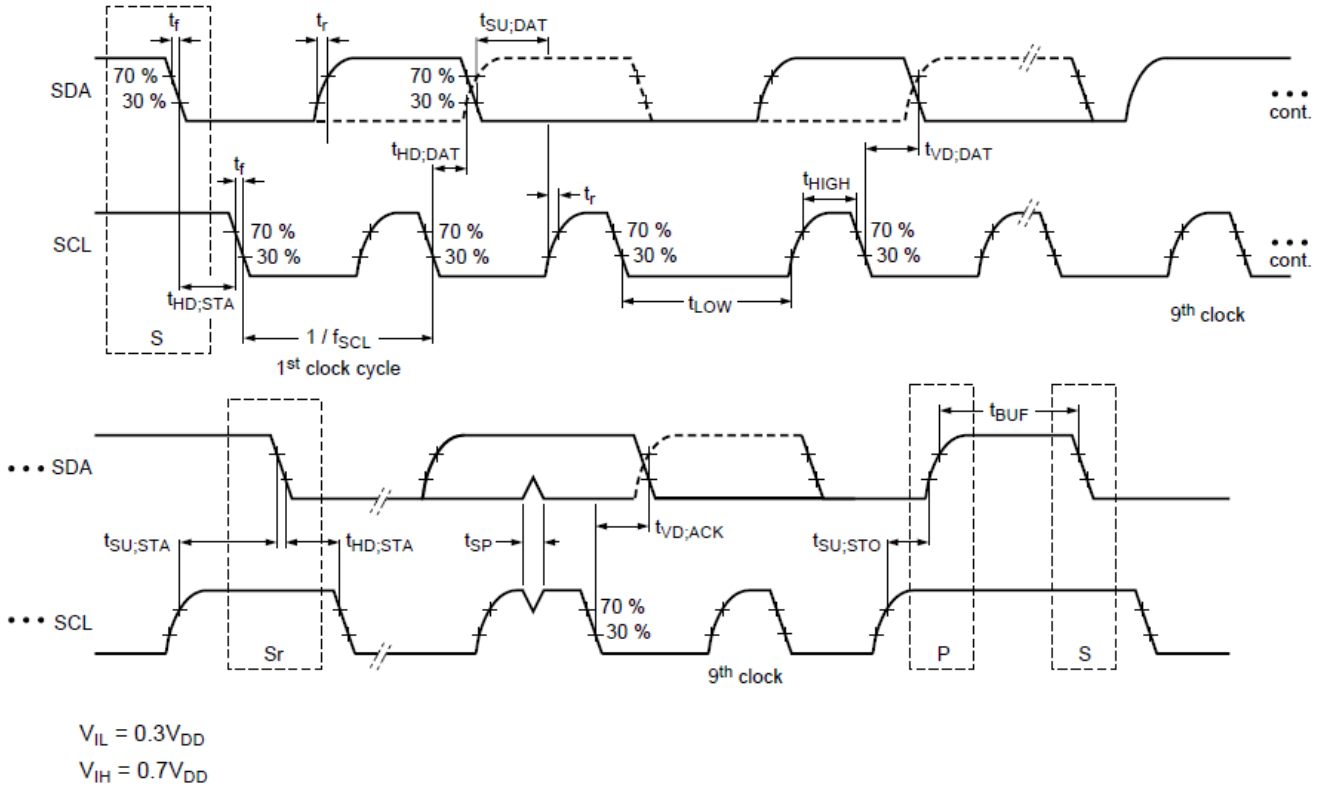


图 18-6I²C 时序参数图例

18.7 I2C 功能描述

I²C 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 I²C 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 I²C 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。

18.7.1 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 18-3 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。

18.7.2 I2C 初始化过程

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_APBEN寄存器，使能I²C模块寄存器总线接口时钟
- 配置CMU模块的I2C_CKSEL和I2C_CKEN寄存器，选择并使能I²C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DFEN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I2CCLK，因此无法在休眠模式下使用。如果从机希望不使用I2CCLK工作，则必须关闭数字滤波。

18.7.3 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

18.7.3.1 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。

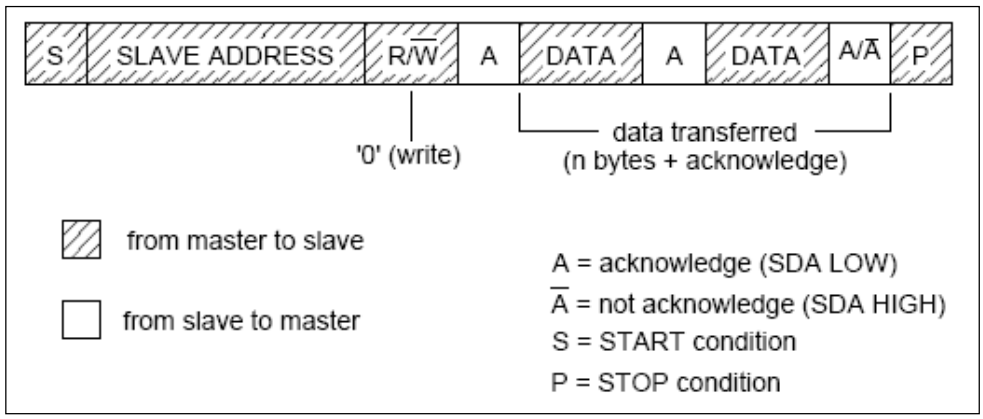


图 18-7 主机向 7 位地址从机写入数据时的帧格式

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：

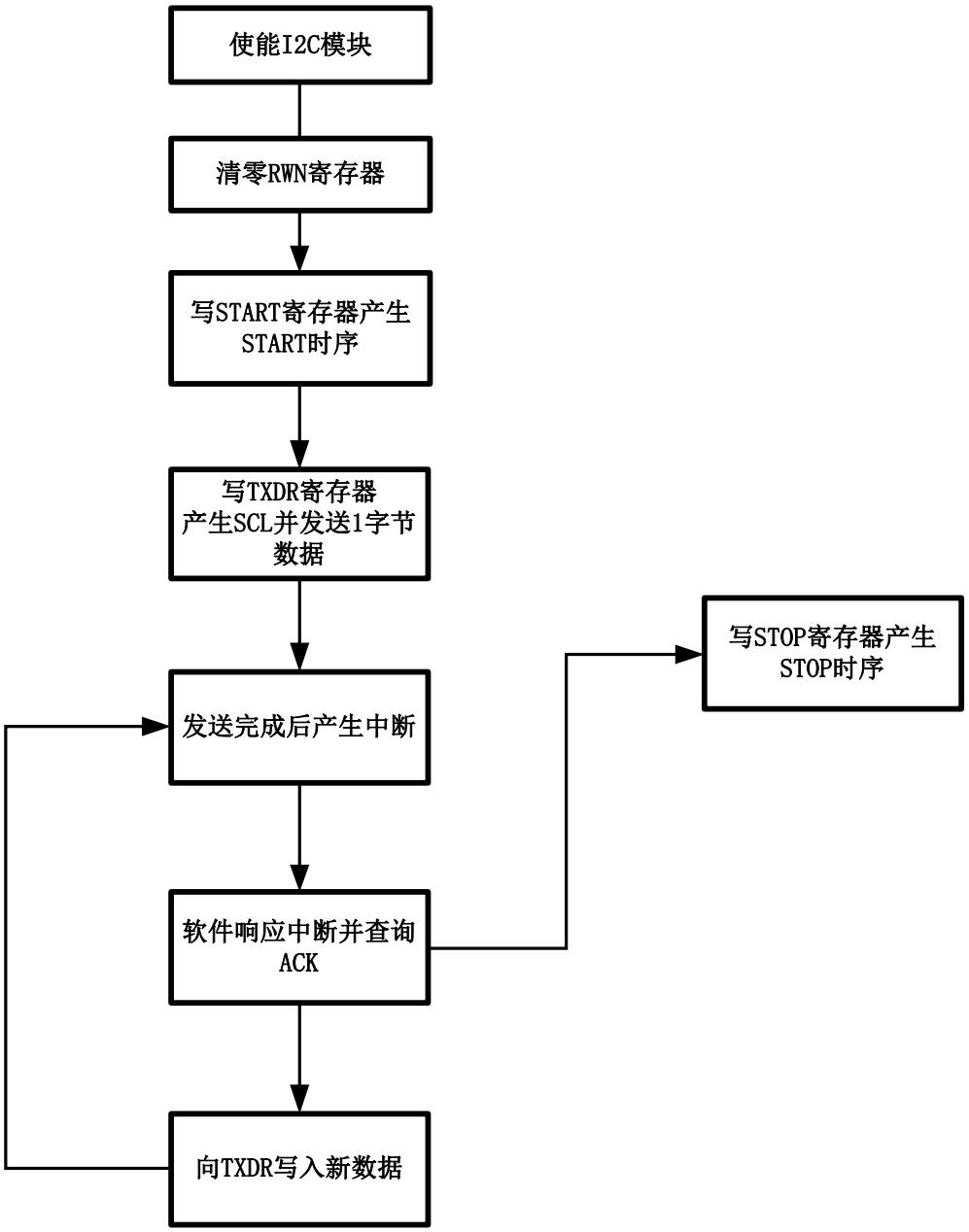


图 18-8I²C 主机软件发送数据流图

I²C主机对7位地址从机写入数据的波形示意图如下：

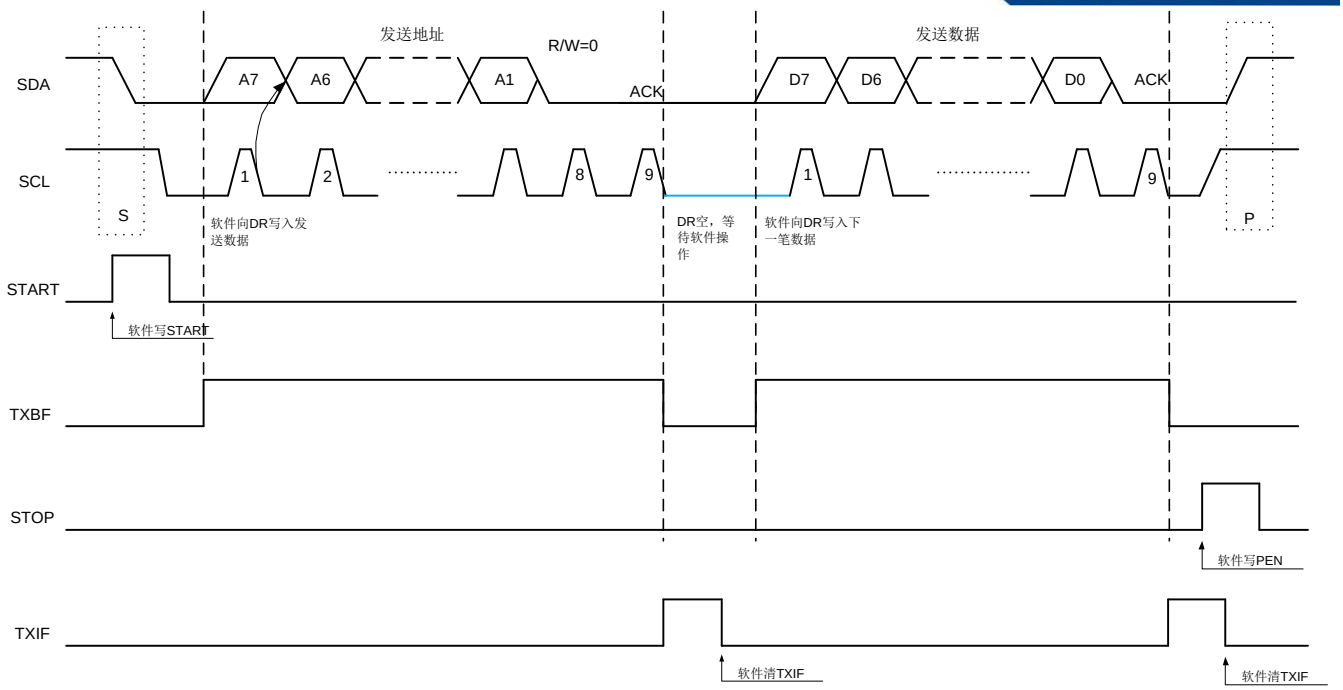


图 18-9 I²C 主机对 7 位地址从机发送数据流图

当数据寄存器DR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DR。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。

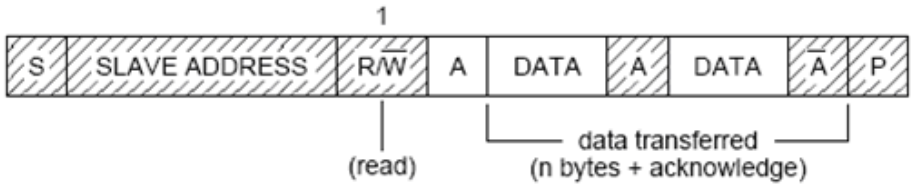


图 18-10 主机从 7 位地址从机读取数据时的帧格式

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：

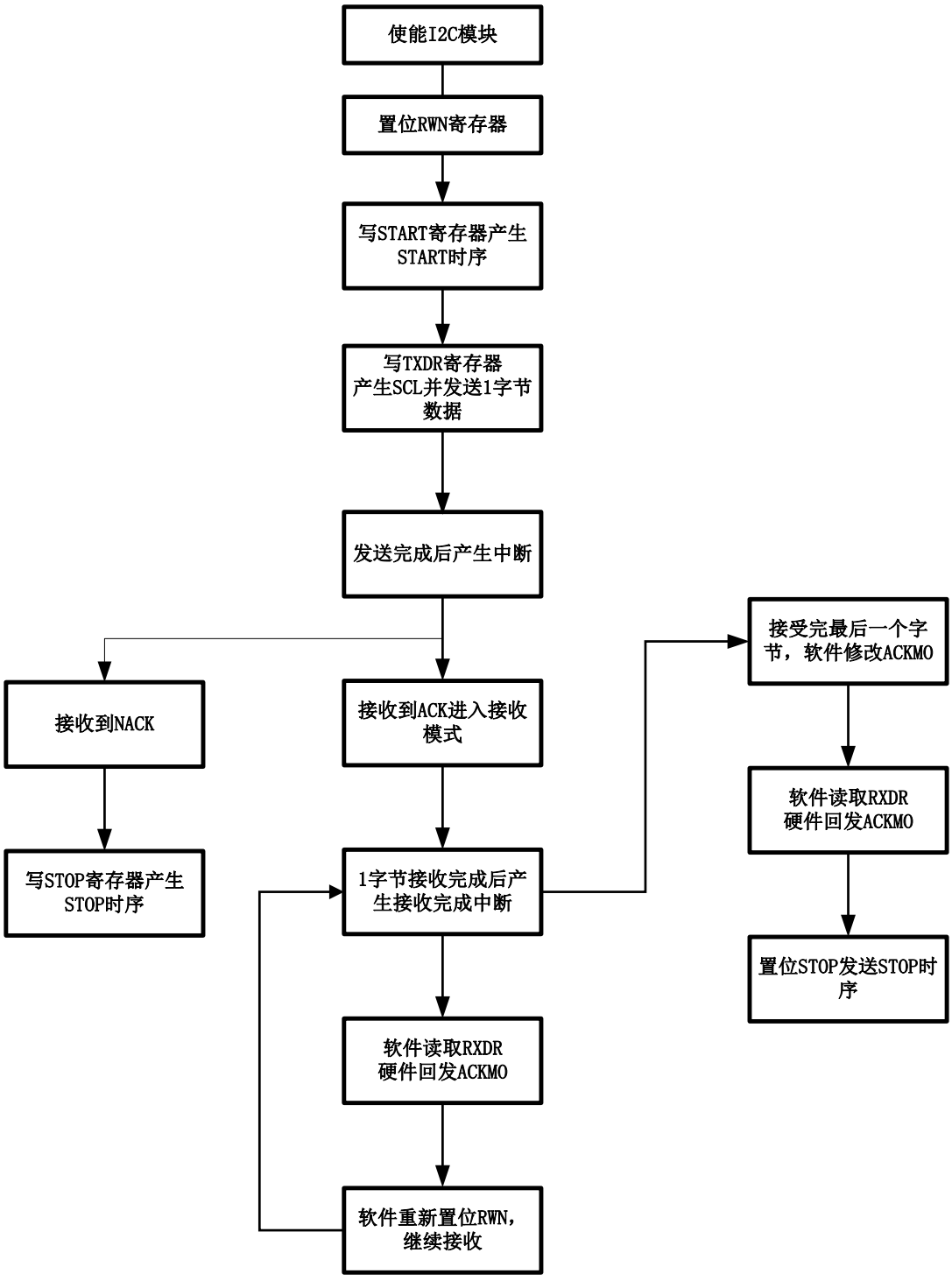


图 18-11 I²C 软件接收数据流图

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：

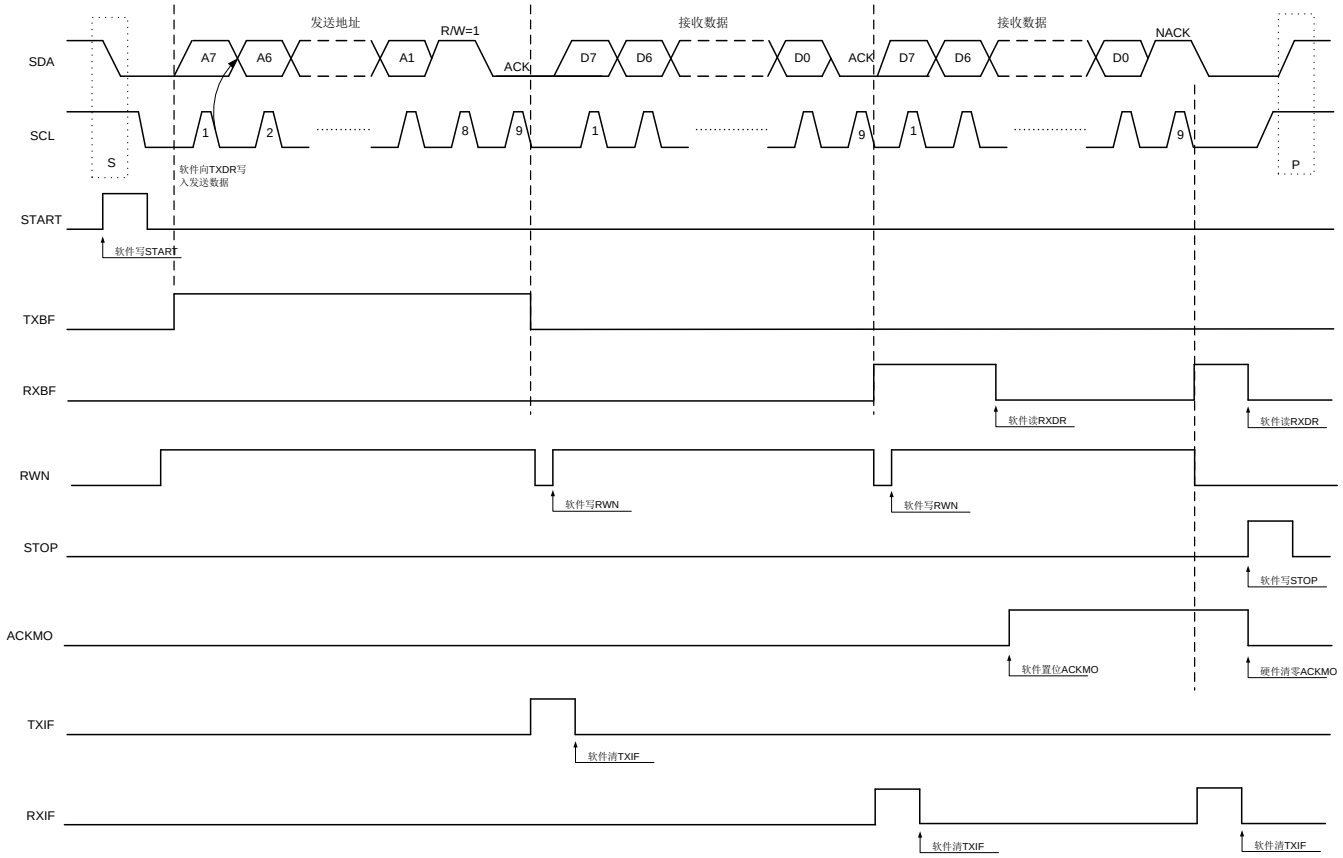


图 18-12 I²C 从 7 位地址从机读取数据流图

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。

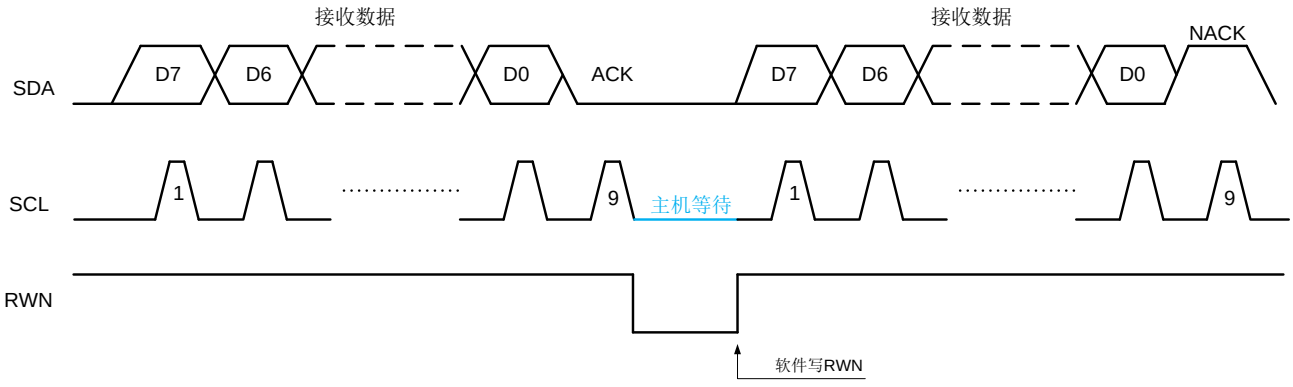


图 18-13 I²C 主机接收数据等待

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。

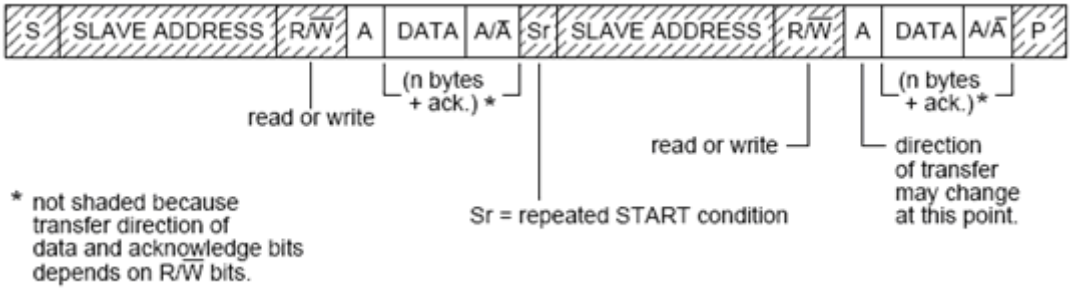


图 18-14 双向数据通信帧格式

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。

18.7.3.2 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。

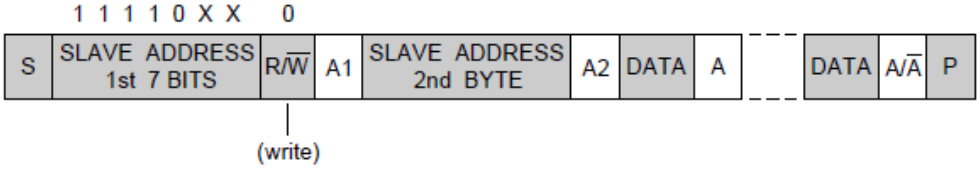


图 18-15 10bit 寻址，主机向从机写入数据

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：

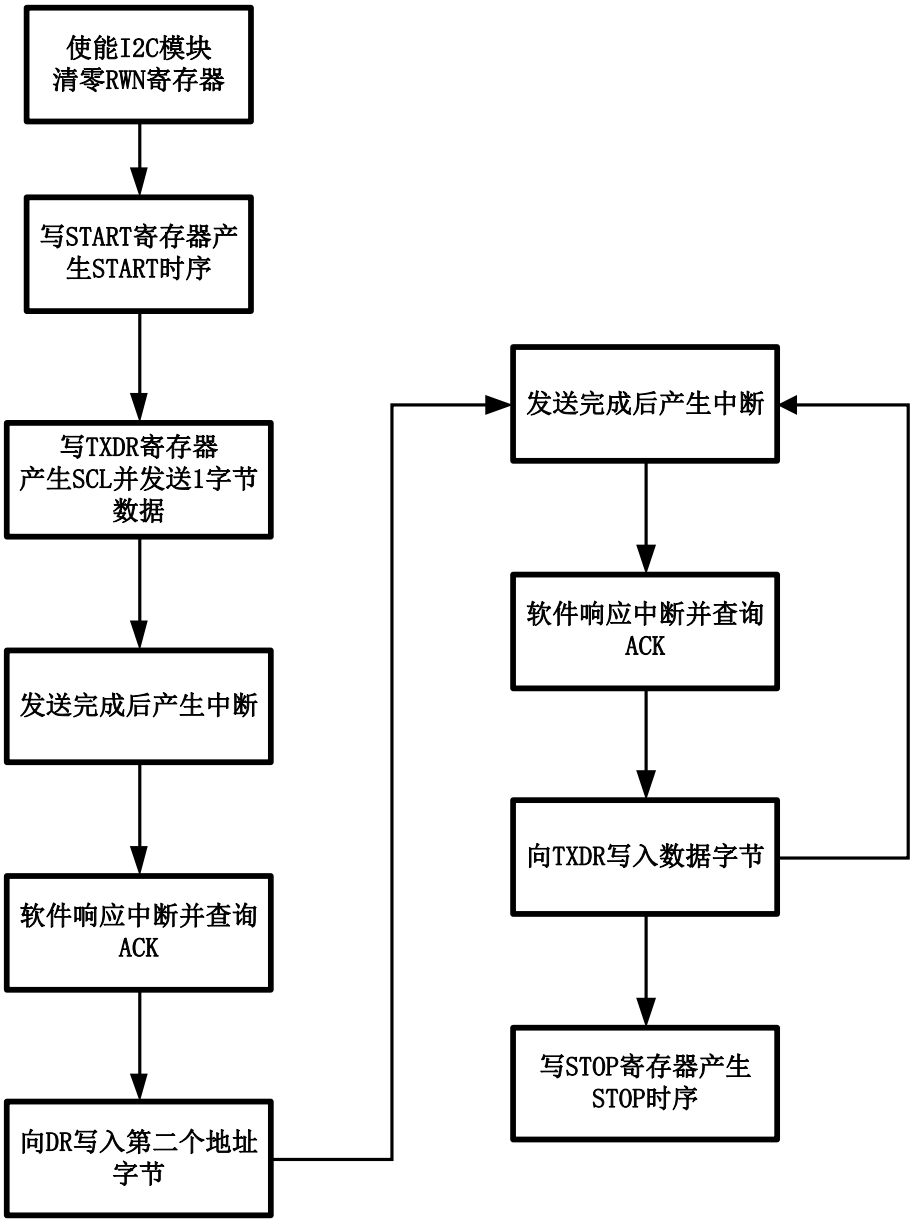


图 18-16I²C 软件发送数据流图

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。

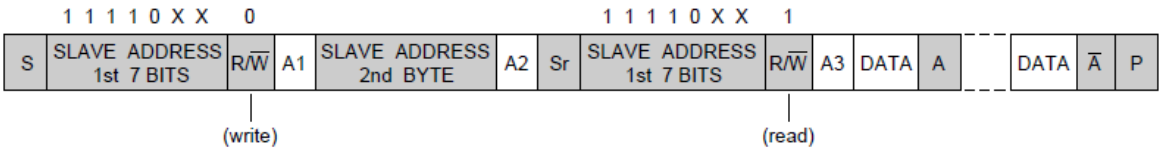


图 18-17 10bit 寻址，主机从从机读取数据

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：

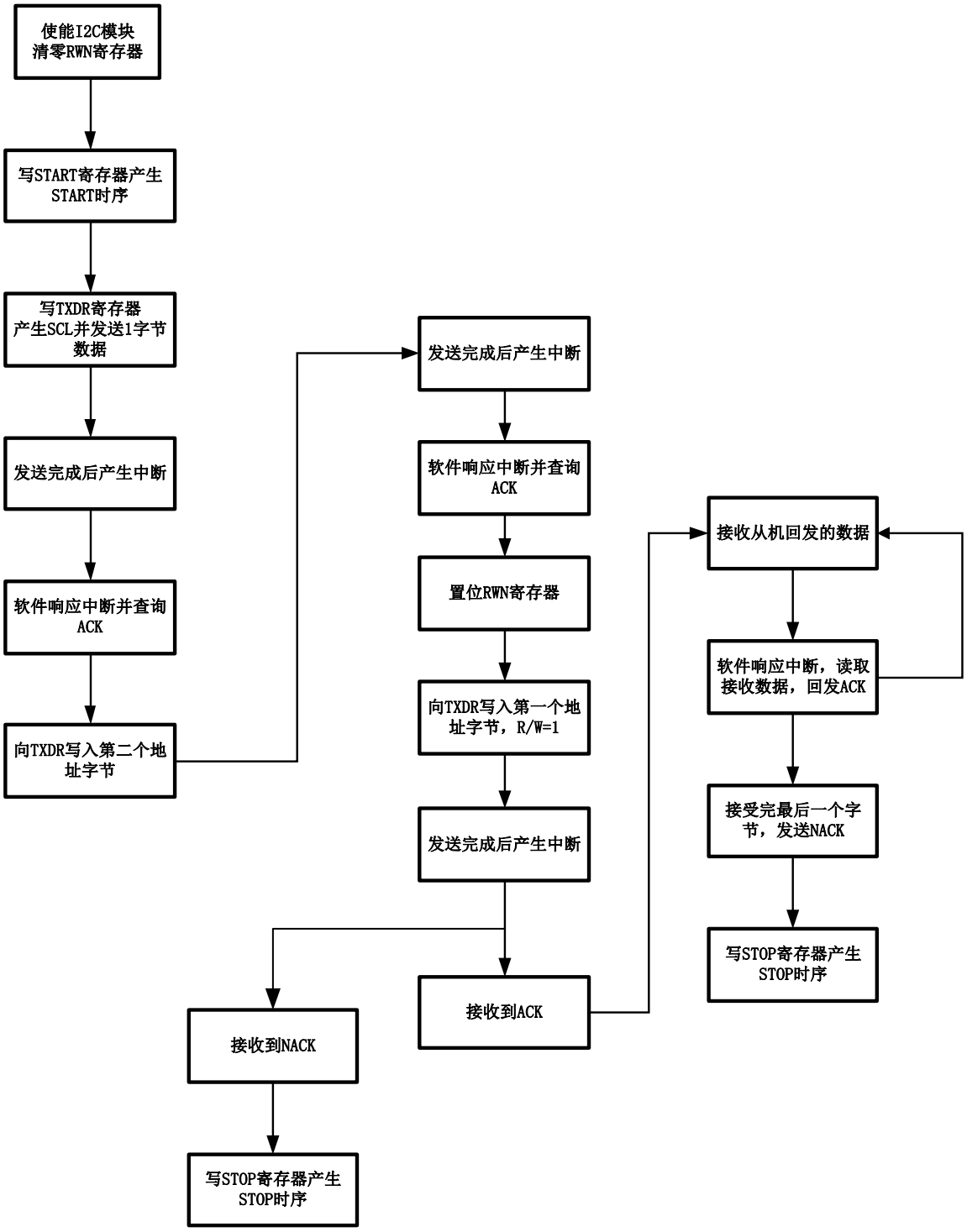


图 18-18 I²C 软件发送数据流图

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。

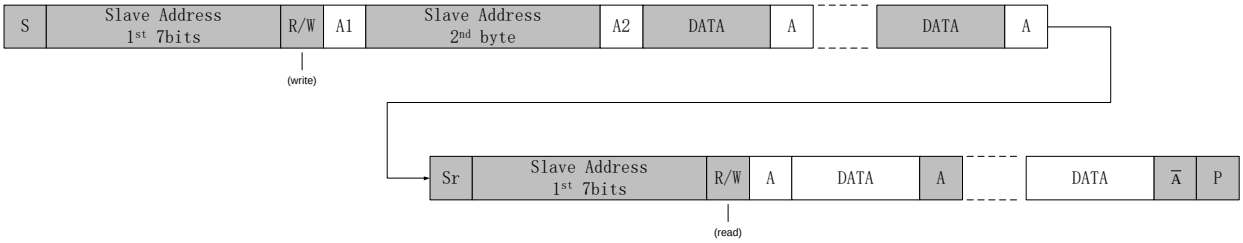


图 18-19 I²C 软件发送数据流图

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。

18.7.3.3 DMA

I²C 主机支持 DMA，需要注意的是，必须在 I²C 模块的总线时钟（APBCLK）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 I2C_TX。

在 DMAEN=1 的情况下，START 置位，如果数据缓存寄存器 TXDR 为空，I²C 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 TXDR，开始数据发送（第一个字节是从机地址）。DMA 发送模式下，I²C 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，I²C 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 I²C 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：

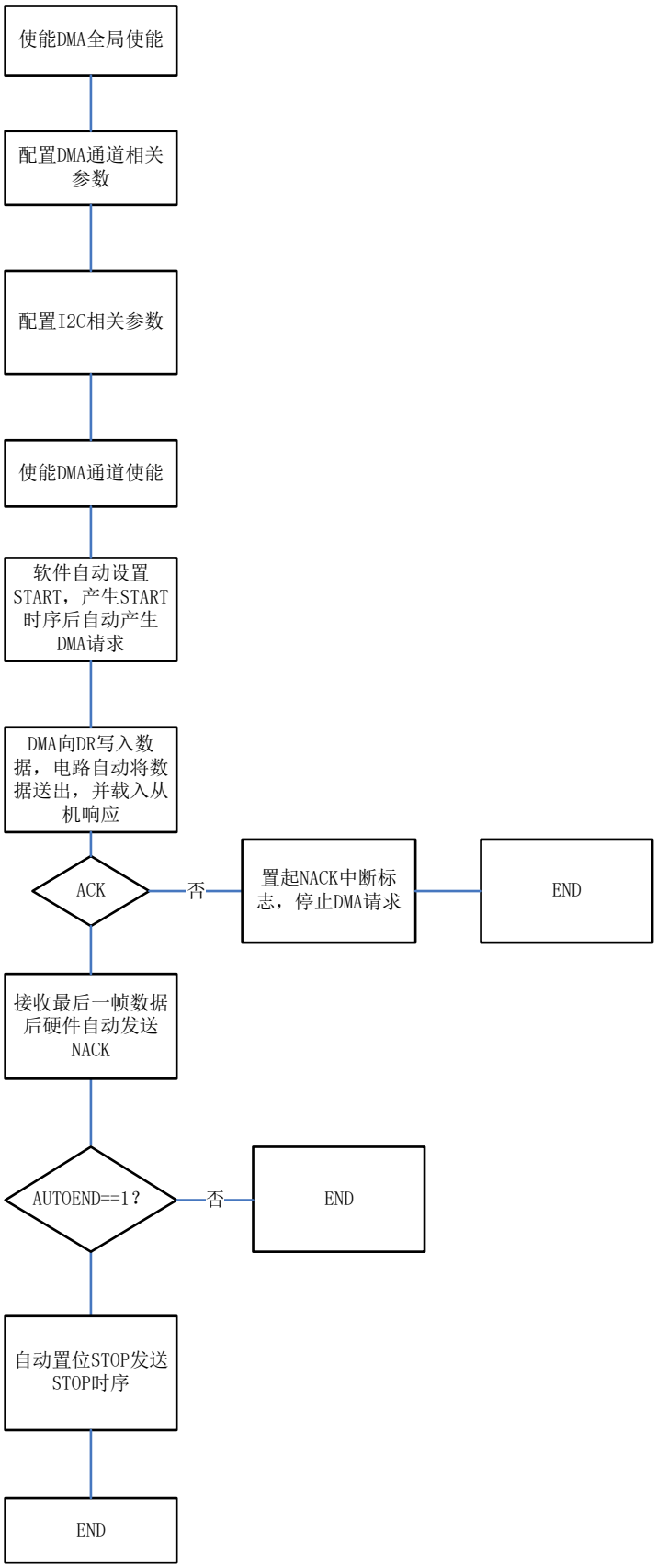


图 18-20 I²C 主机 DMA 发送流程图

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 I2C_RX。

软件首先发送完从机地址后，设置 DMAEN=1，然后使能对应的 DMA 通道为 I2C_RX，I^2C 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 RXDR 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 I^2C 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I2C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：

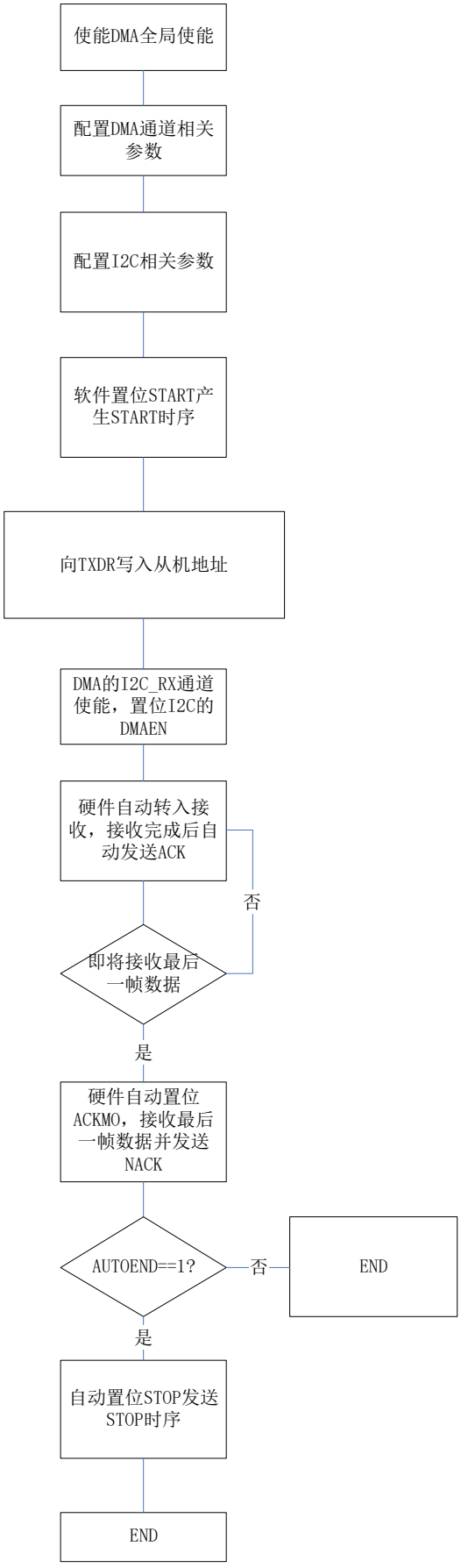


图 18-21 I²C 主机 DMA 接收流程图

18.7.3.4 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。

18.7.3.5 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。

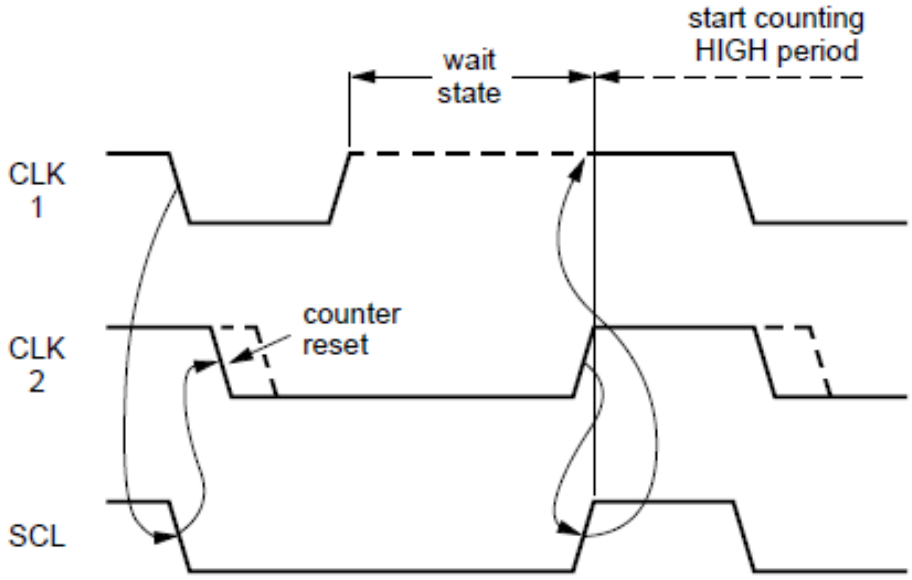


图 18-22 I²C 多主机时钟同步

18.7.3.6 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。

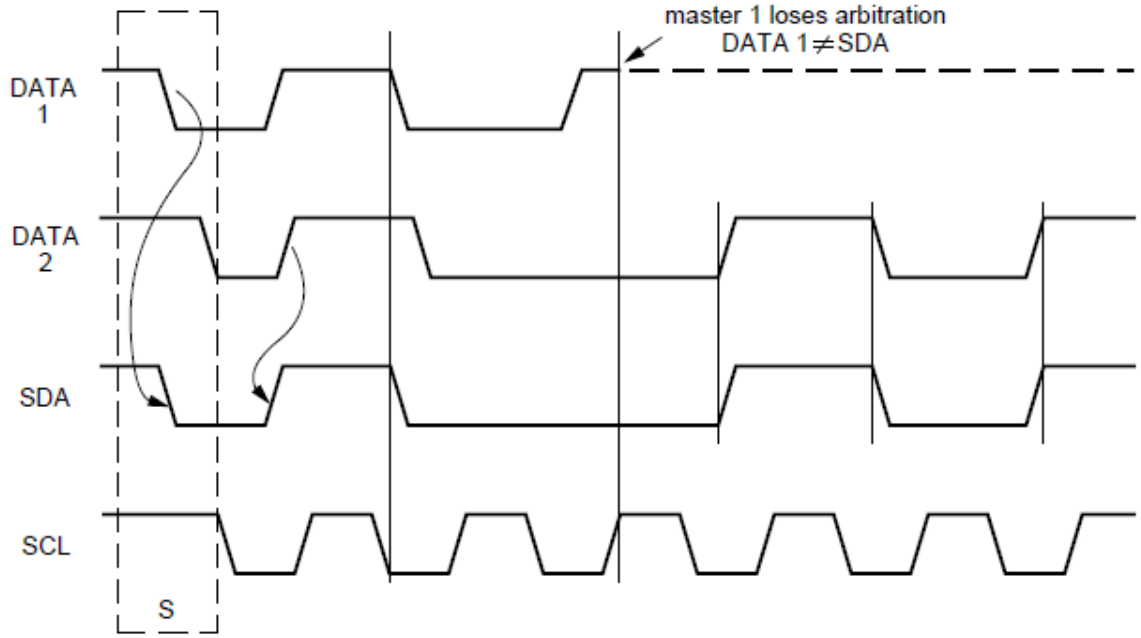


图 18-23 I²C 多主机数据仲裁

18.7.3.7 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。

18.7.3.8 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保持时间长度。

SDAHD 时序对于 SDA 数据位和 ACK/NACK 都同样有效。

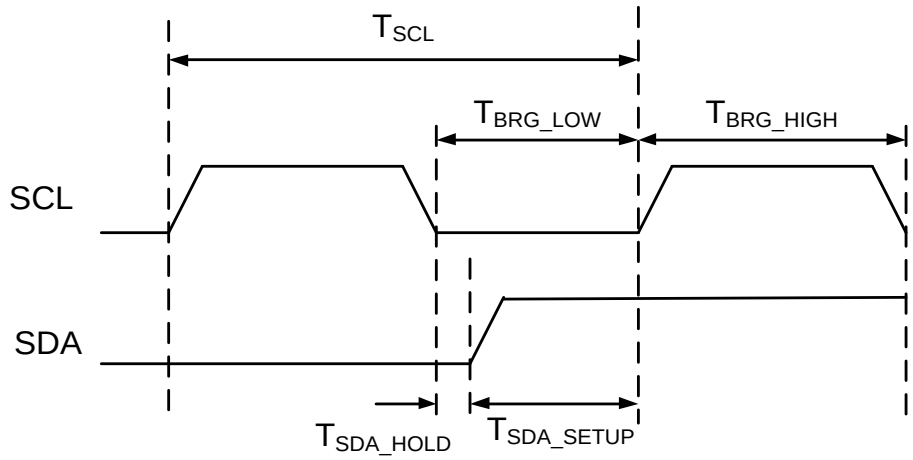


图 18-24 主机时序控制

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

18.7.4 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。

18.7.4.1 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 SLAVE_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。

18.7.4.2 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 I2CCLK。如果仅使用模拟滤波，则可以不需要 I2CCLK 工作。

在休眠模式下如果希望 I^2C 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 I^2C 模块无法检测总线电平变化。

18.7.4.3 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
- 由于R/W=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TXDR
- 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：

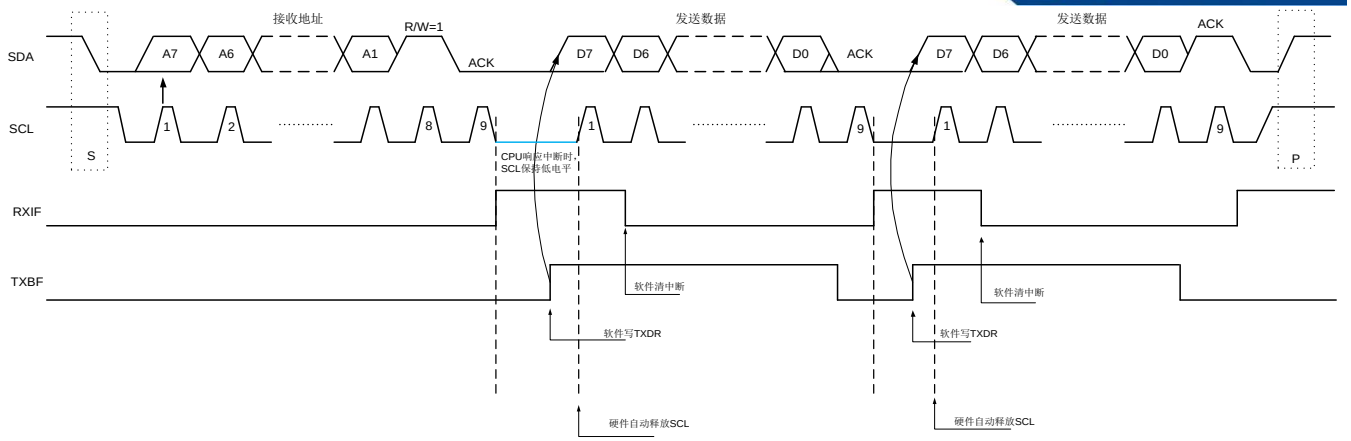


图 18-25 I²C 从机数据发送波形

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

18.7.4.4 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
- 由于R/W=0，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RXDR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RXBF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RXDR，硬件自动清零RXBF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）：

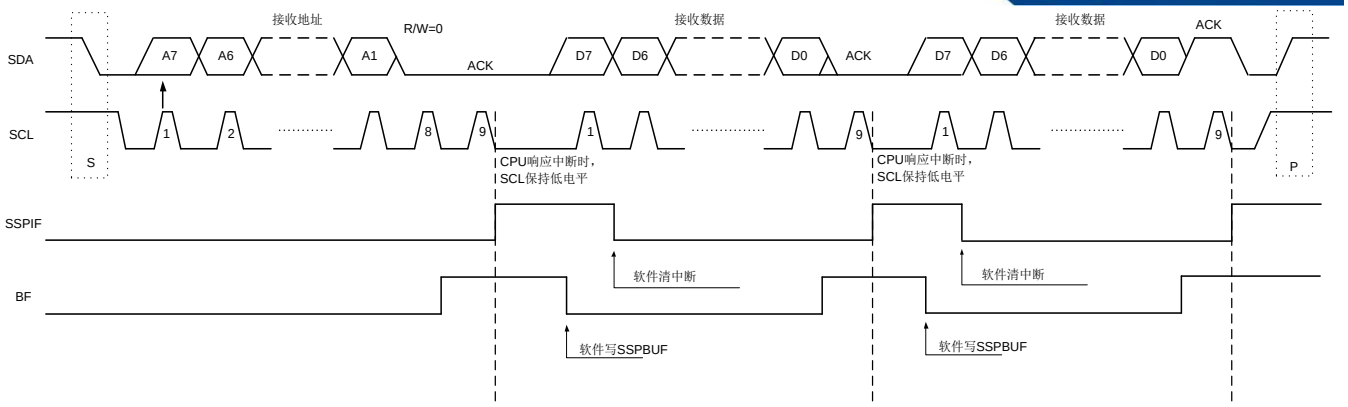


图18-26 I²C从机数据接收波形

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，ADM标志置位，地址字节将被写入RXDR并置位RXBF标志，然后硬件拉低SCL。当软件读取RXDR后，RXBF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RXDR并导致RXBF置位，软件需要读取RXDR来清零RXBF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寄存器清零，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RXBF标志。如果接收中出现了数据溢出，RXOV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；RXOV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BF标志。

接收数据溢出

当从机接收缓冲区满（BF=1）时，如果又收到新的数据，则发生接收溢出，RXOV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：

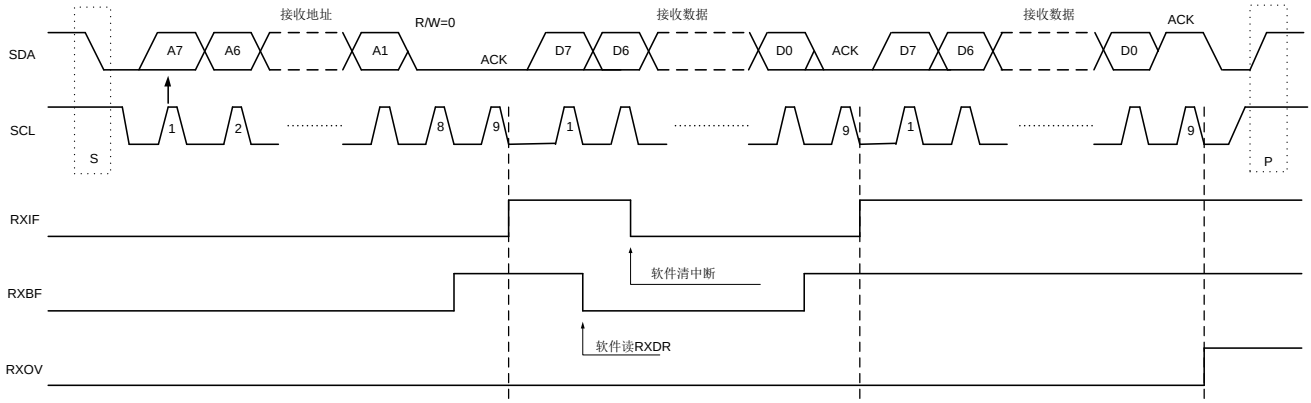


图18-27 I²C从机数据接收波形（SCLSEN=0，接收溢出）

18.7.4.5 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I2CCLK配置为RCHF，并关闭数字滤波功能。

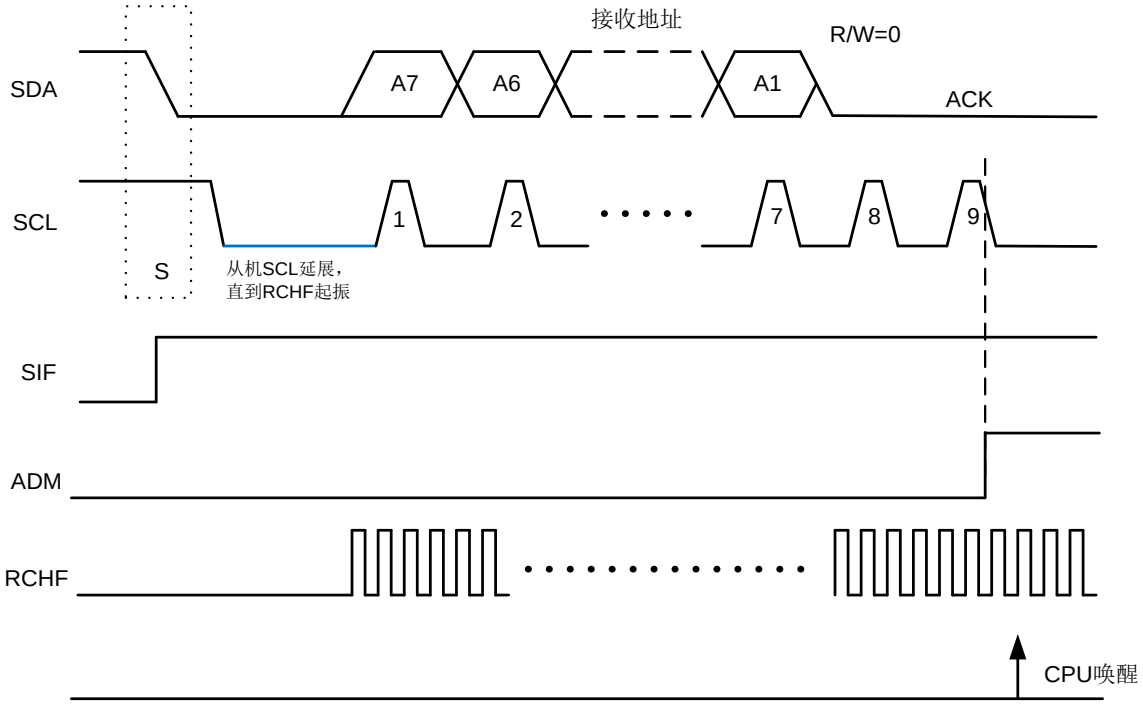


图 18-28 从机地址匹配唤醒

18.7.4.6 DMA

I²C 从机支持 DMA 操作，需要注意的是，必须在 I²C 的总线时钟（APBCLK）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 ADM 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 I2C_RX，并使能 I²C 从机的 DMAEN；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 RXDR。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 清零，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：

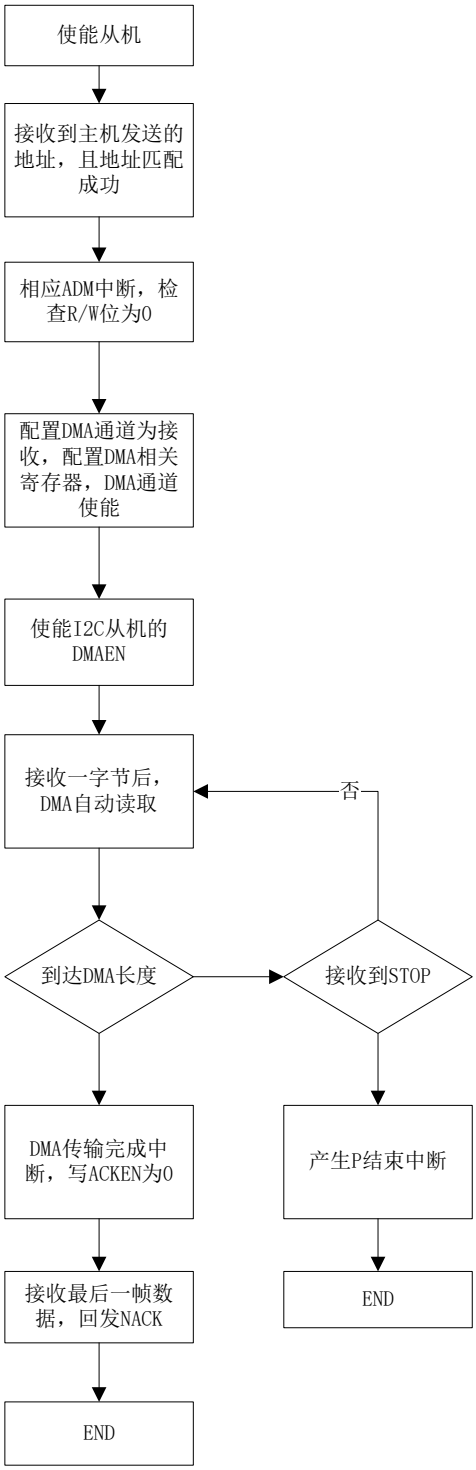


图18-29 I²C从机DMA接收流程图

从机使用DMA发送数据

当 I²C 从机接收到正确的地址后，产生 ADM 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 I2C_TX，并使能 I2C 从机的 DMAEN；随后当从机数据缓存 TXDR 为空时，将产生 DMA 请求，通知 DMA 写入 TXDR。

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：

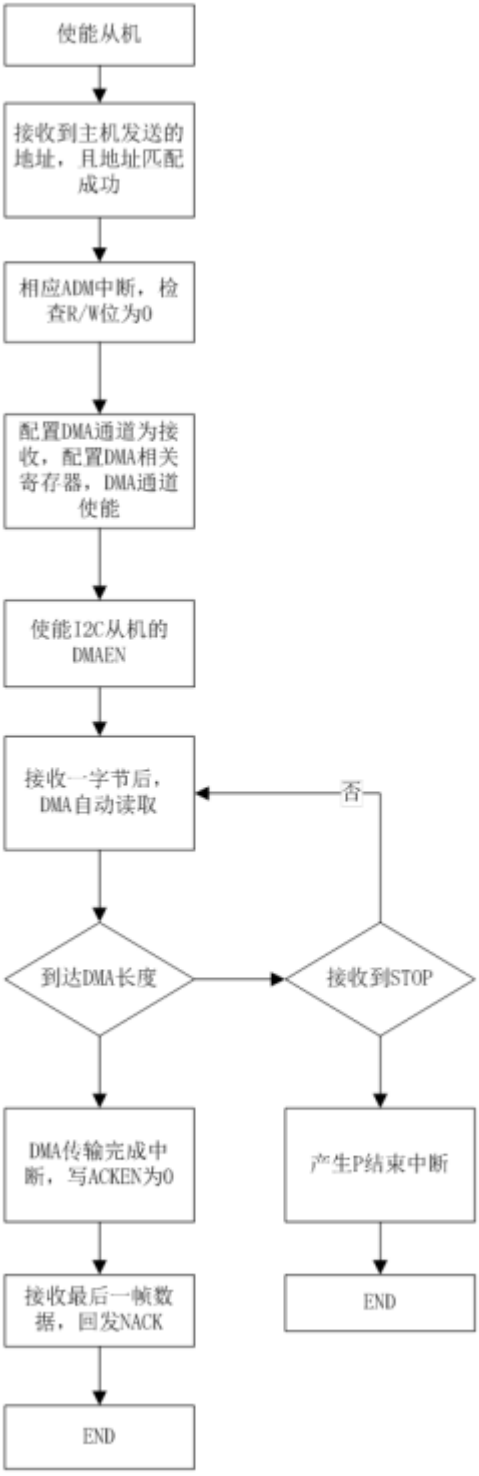


图18-30 I2C从机DMA发送流程图

18.7.4.7 SDA 输出延迟

由于从机的数据收发只使用 SCL 进行，因此需要一些模拟延迟来实现 SDA 的数据建立和保持时间控制，而 SCL 的时序完全由主机控制。

从机时序控制如下图所示。根据 I²C 协议要求，SDA 相对于 SCL 下降沿的数据保持时间最小为 0ns，即从机使用 SCL 下降沿发送数据即可满足要求。但是考虑到总线上 SCL 波形的实际下降时间，为了更好的覆盖保持时间要求，在 SDA 输出上额外加入大于 300ns 的 RC 延迟。这个延迟仅需施加在 I²C 从机的 SDA 输出上（SSP_SDAO）

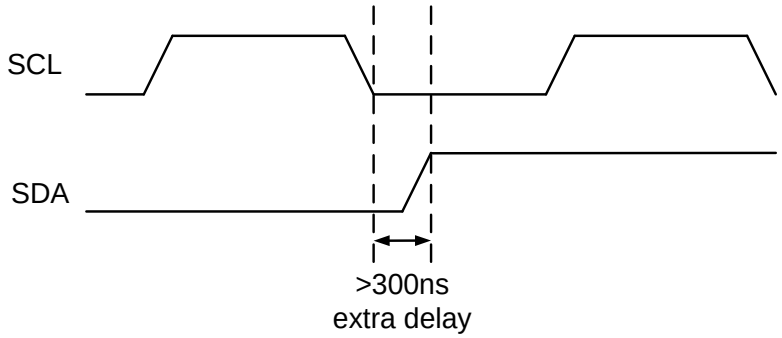


图 18-31 SDA 输出延迟波形

18.8 SMBus 功能描述

18.8.1 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 000），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RW_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 18-4SMBus 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

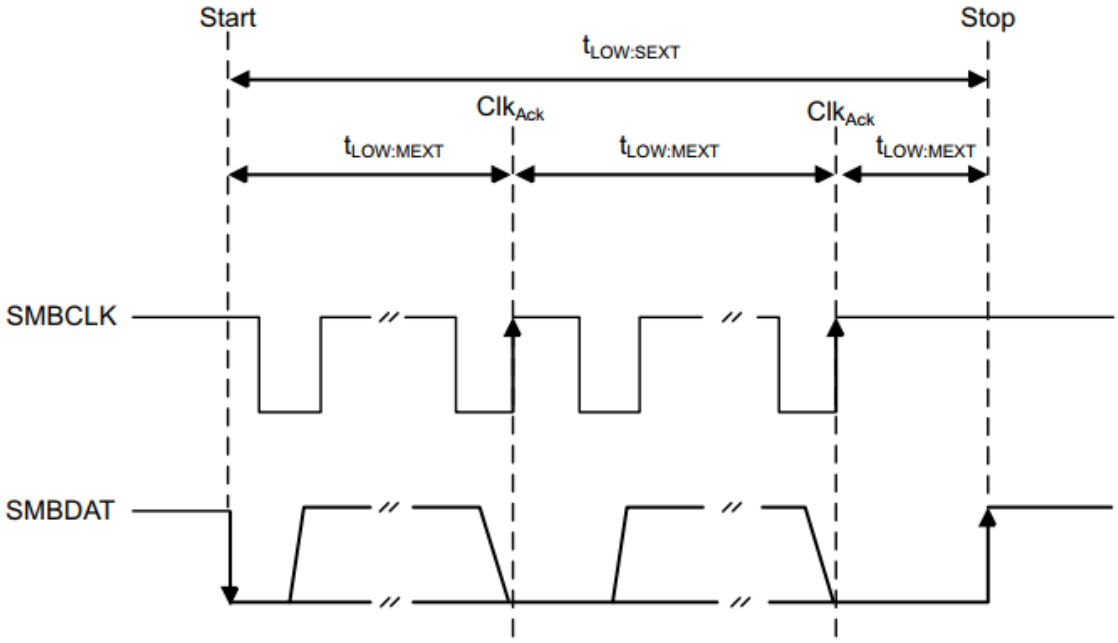


图 18-32$t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 波形

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。

18.8.2 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。

18.8.3 SMBus 主机功能

18.8.3.1 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，I2C_SMBUS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

18.8.3.2 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I2C_SMBUS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。

18.8.3.3 SMBus 从机发送

SMBus从机发送数据流程与I^2C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

18.8.3.4 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。

18.8.3.5 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。

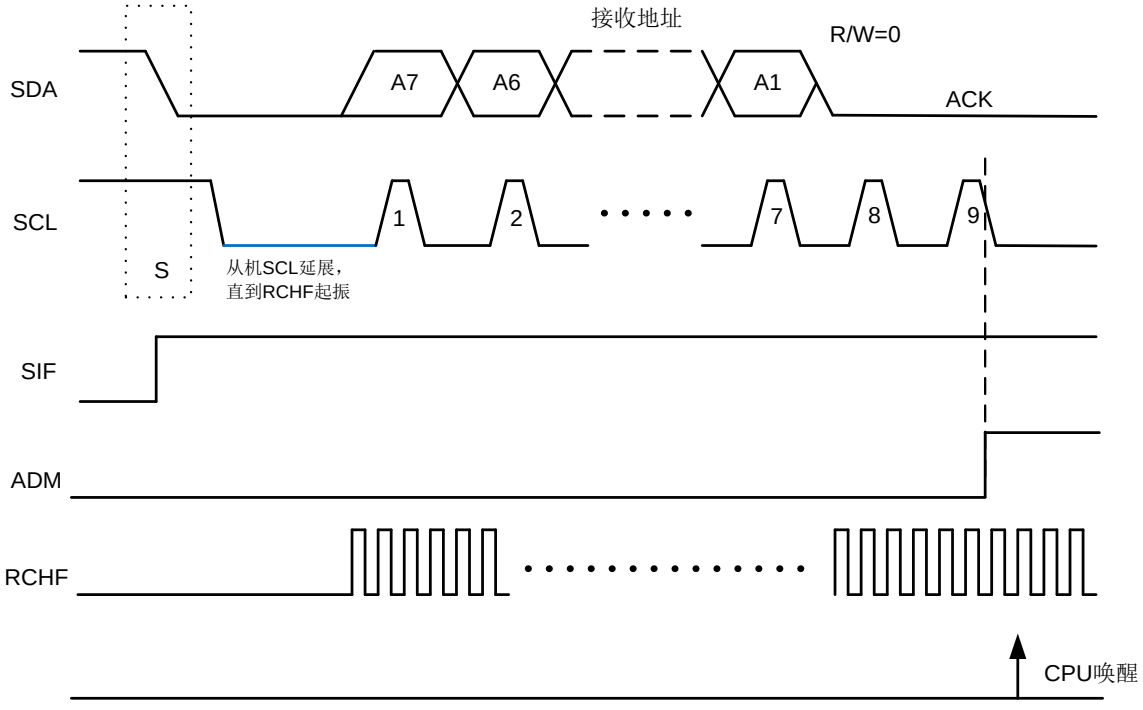


图 18-33 从机地址匹配唤醒

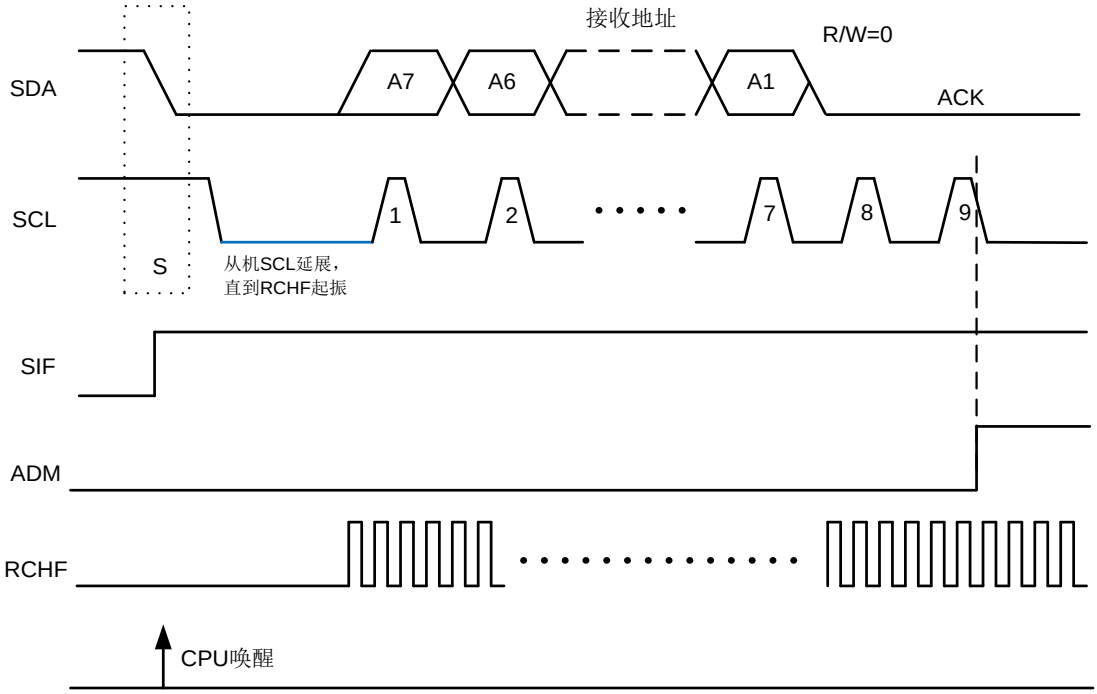


图 18-34 从机 START 唤醒

注意：使用SMBus从机唤醒功能时，必须将I2CCLK配置为RCHF

18.8.4 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：

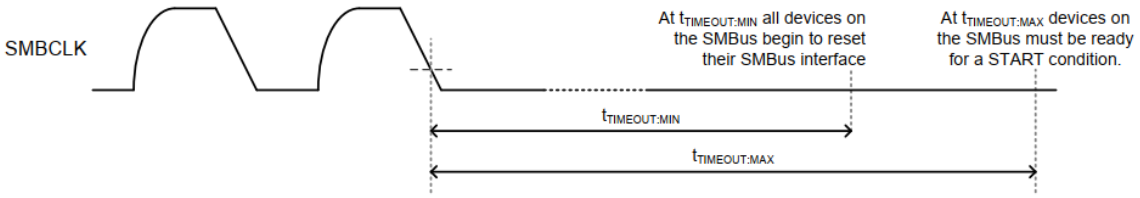


图 18-35SMBus 总线时钟超时事件

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。

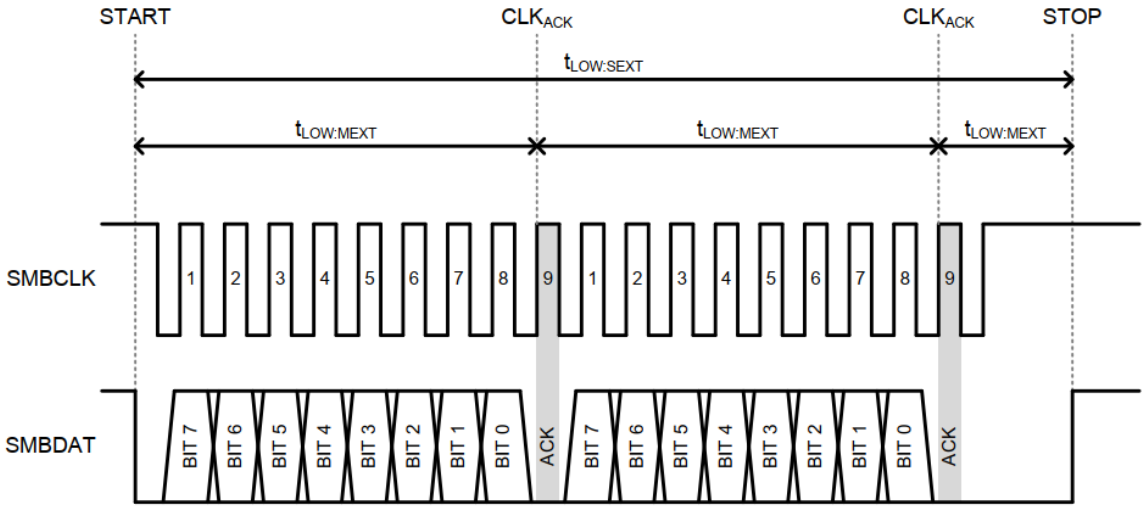


图 18-36 主机和从机超时事件

18.8.5 CRC-8

I2C_SMBUS 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。

18.9 寄存器

I2CSMB0模块基地址：0x40011C00

I2CSMB1模块基地址：0x40012400

offset 地址	名称	符号
	I2C_SMB	
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_TIMINGR
0x00000018	I2C_SMB 超时寄存器	I2CSMB_TOR
0x0000001C	I2C_SMB 接收数据寄存器	I2CSMB_RXDR
0x00000020	I2C_SMB 发送数据寄存器	I2CSMB_TXDR
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

18.9.1 I2C_SMB 控制寄存器 1（I2CSMB_CR1）

名称	I2CSMB_CR1							
Offset	0x00000000							
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-000			
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 I2CCLK 0010：数字滤波长度 2 个 I2CCLK …… 1111：数字滤波长度 15 个 I2CCLK
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

18.9.2 I2C_SMB 控制寄存器 2（I2CSMB_CR2）

名称	I2CSMB_CR2							
Offset	0x00000004							
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)

18.9.3 I2C_SMB 中断控制寄存器（I2CSMB_IER）

名称	I2CSMB_IER							
Offset	0x00000008							
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

位号	位名	说明
		0：禁止接收完成产生中断

18.9.4 I2C_SMB 状态和标志寄存器（I2CSMB_ISR）

名称	I2CSMB_ISR							
Offset	0x0000000C							
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 RXDR 或 TXDR 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)

位号	位名	说明
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 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)

18.9.5 I2C_SMB 波特率设置寄存器（I2CSMB_BGR）

名称	I2CSMB_BGR							
Offset	0x00000010							
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 I2C 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 I2C 工作时钟计数 (Master SCL Low level length)

18.9.6 I2C_SMB 主机时序控制寄存器（I2CSMB_TIMINGR）

名称	I2CSMB_TIMINGR							
Offset	0x00000014							
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	SCLDLY							
位权限	R/W-0000 1000							
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
31:24	--	RFU：未实现，读为 0
23:16	SCLDLY	从机模式下，SCL 延展后，软件写入 TXBUF 后到 SCL 释放的时间延迟，以 I2C 工作时钟计数，默认 8 个 clock
15:9	--	RFU：未实现，读为 0
8:0	SDAHD	主机发送时定义 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 BRGL

18.9.7 I2C_SMB 超时寄存器（I2CSMB_TOR）

名称	I2CSMB_TOR							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TEXTEN	-			TIMEOUTB			
位权限	R/W-0	U-0			R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TIMEOUTB							
位权限	R/W-0000 0000							
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-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUTA							
位权限	R/W-0000 0000							

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 $t_{LOW:SEXT}$ $t_{LOW:EXT}=(TIMEOUTB+1)\times2048\times t_{I2CCLK}$ 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， $t_{TIMEOUT}=(TIMEOUTA+1)\times2048\times t_{I2CCLK}$ TIDLE=1 时用于检测总线空闲， $t_{IDLE}=(TIMEOUTA+1)\times4\times t_{I2CCLK}$ 注意：禁止在 TOEN=1 的情况下改写此寄存器

18.9.8 I2C_SMB 接收数据寄存器（I2CSMB_RXDR）

名称	I2CSMB_RXDR							
Offset	0x0000001C							
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	RXDATA							
位权限	R-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	RXDATA	8bit 接收数据，只读

18.9.9 I2C_SMB 发送数据寄存器（I2CSMB_TXDR）

名称	I2CSMB_TXDR							
Offset	0x00000020							
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	TXDATA							
位权限	R/W-0000 0000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	TXDATA	8bit 发送数据

18.9.10 I2C_SMB 从机地址寄存器（I2CSMB_SADR）

名称	I2CSMB_SADR							
Offset	0x00000024							
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

18.9.11 CRC 数据寄存器（I2CSMB_CRC_DR）

名称	CRC_DR							
Offset	0x00000028							
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)

18.9.12 CRC LFSR 寄存器（I2CSMB_CRC_LFSR）

名称	CRC_LFSR							
Offset	0x0000002C							
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

18.9.13 CRC 多项式寄存器（I2CSMB_CRC_POLY）

名称	CRC_POLY							
Offset	0x00000030							
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-0010 0001							

位号	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**POLY**	CRC 运算多项式系数 (CRC Polynominals)

19 通用异步收发传输器（UART）

19.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 5路独立通道（UART0, UART1, UART2, UART3, UART4）
- 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, UART1, UART2, UART3支持LIN通信，包括以下特性：

- 唤醒信号的检测和发送
- 同步间隔域（Break Field）检测
- 同步域（Sync Field）检测和波特率自适应

UART4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

19.2 结构框图

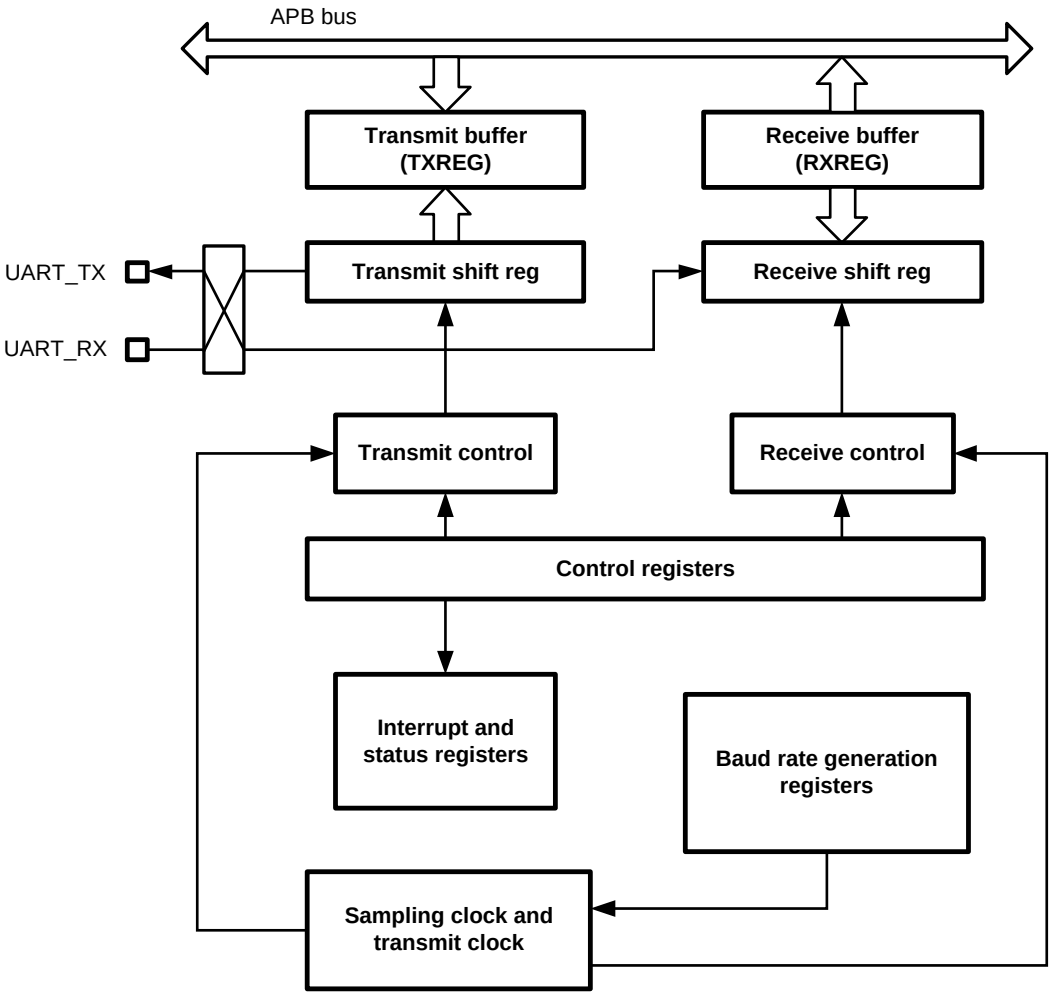


图 19-1 UART 接口时序

19.3 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 FM33LF0x6 的引脚映射关系

引脚		UARTx	符号	功能
PA13	PA2	UART0	UART0_RX	数据接收
PA14	PA3		UART0_TX	数据发送
PB13	PC2	UART1	UART1_RX	数据接收
PB14	PC3		UART1_TX	数据发送
PB0	PD7	UART2	UART2_RX	数据接收
PB1	PD8		UART2_TX	数据发送
PA0	PB2	UART3	UART3_RX	数据接收
PA1	PB3		UART3_TX	数据发送
PC4	PD0	UART4	UART4_RX	数据接收
PC5	PD1		UART4_TX	数据发送

表 19-1 UART 引脚列表

19.4 UART 类型区分

FM33LF0xxA集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART0/1/2/3	UART4	LPUART
DMA 支持	Y	Y	Y
半双工/全双工	Y	Y	Y
红外发射	Y	Y	-
双时钟域（工作时钟独立于总线）	Y	-	Y
休眠唤醒	Y	-	Y
接收超时	Y	-	-
发送延迟	Y	-	-
数据长度	6、7、8、9bits		
LIN support	Y	N	N
智能卡模式	N	Y	N

表 19-2 UART 类型列表

19.5 UART 字符描述

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。

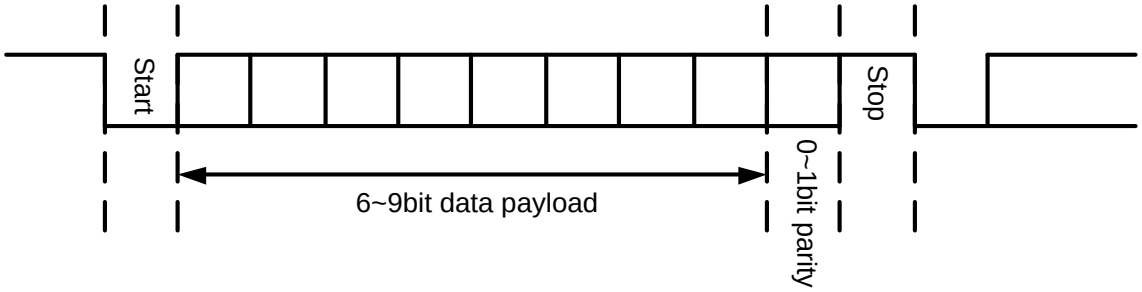


图 19-2 UART 字符描述

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 19-3 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】。

19.5.1 智能卡模式

ISO7816-3协议规定的字符格式如下图。

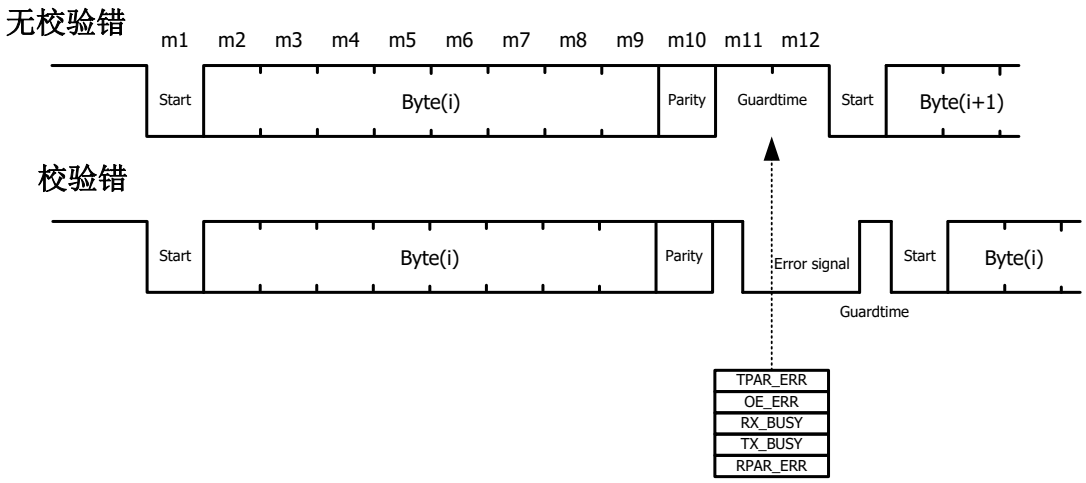


图 19-3 7816-3 帧结构图

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RX_BUSY无效并产生可能的OE_ERR标志，完成数据发送；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RPAR_ERR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。

19.6 功能描述

19.6.1 时钟结构

UART0~3 采用了双时钟结构：

- 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 PCLK
- 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 UART0~3 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。

UART4 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。

19.6.2 同步策略

UART 支持两种同步策略：起始位同步和重同步。

当 RESYNCEN=0，UART 只执行起始位同步，时钟频率误差将在字节内累积。

当 RESYNCEN=1，UART 执行起始位同步，并在字节内每个下降沿执行重同步。

注意，针对 LIN2.1，协议规定从机不执行重同步，只执行起始位同步，因此应设置 RESYNCEN=0。

通过 start bit 下降沿采样可以实现起始位重同步。使用 UCLK 采样 RXD，当检测到起始位下降沿时，重同步波特率计数器，以实现波特率同步于起始位检测，如下图所示。

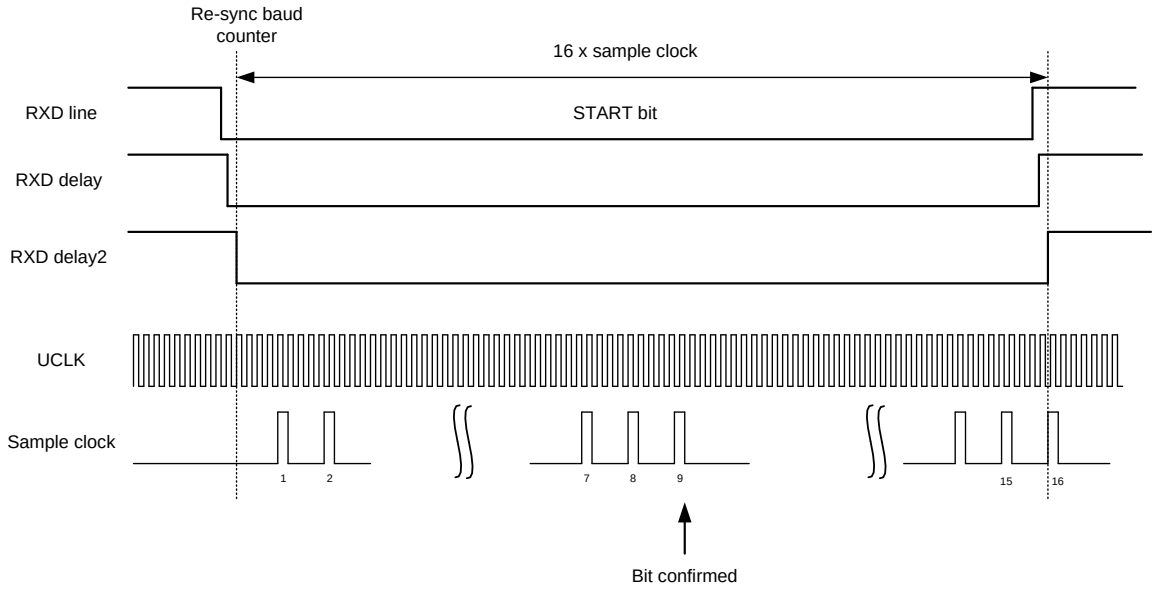


图 19-4 起始位同步

19.6.3 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。

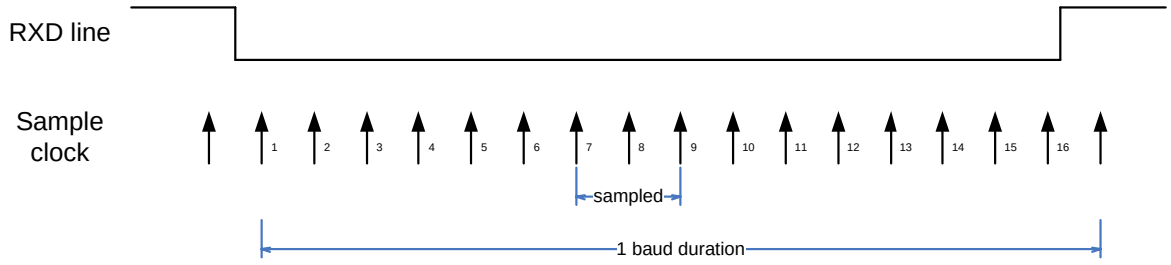


图 19-5 位接收 16 倍采样

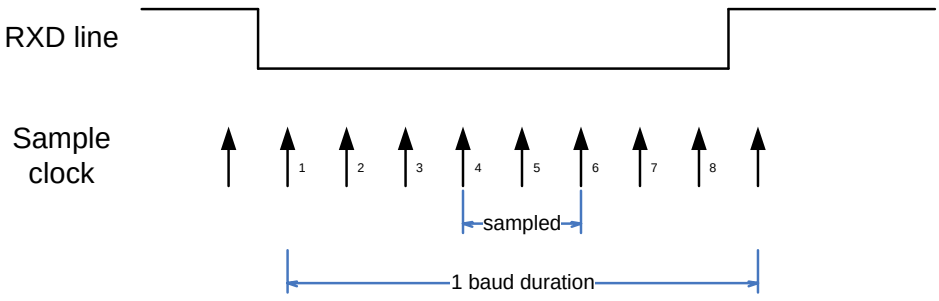


图 19-6 位接收 8 倍采样

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。

19.6.4 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。

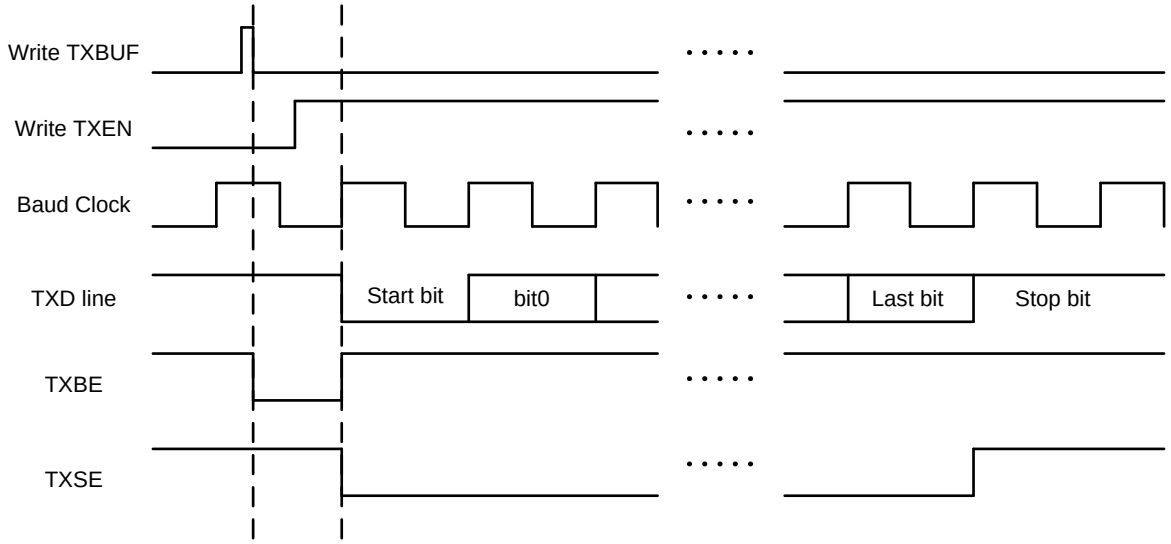


图 19-7 UART 异步发送波形 1

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 IRCON 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。

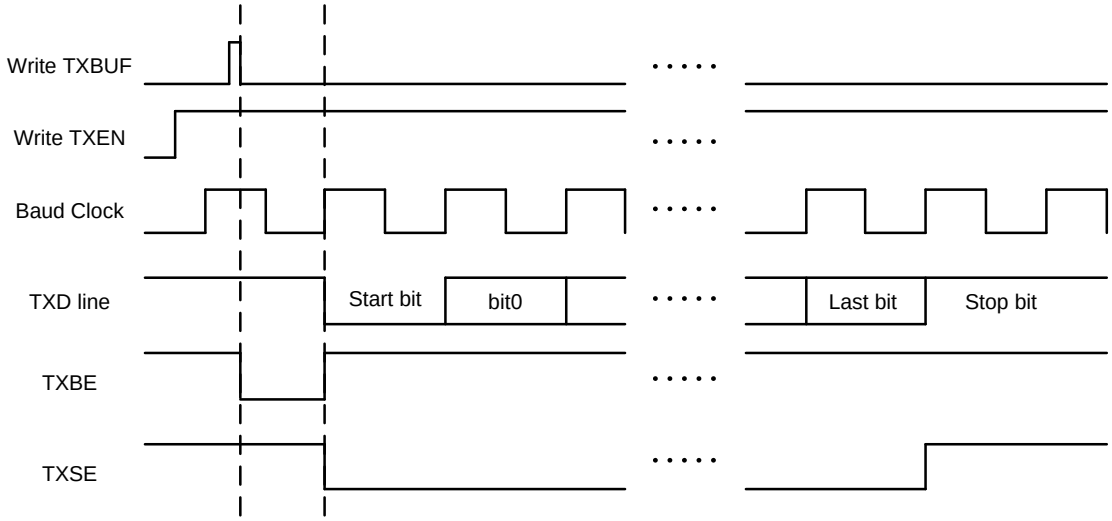


图 19-8 UART 异步发送波形 2

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。

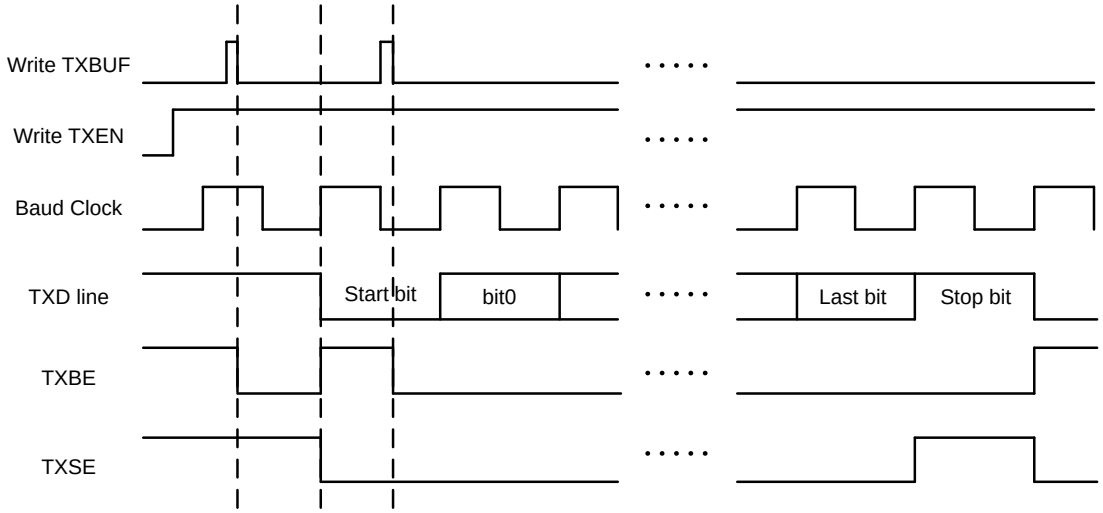


图 19-9 UART 异步发送波形 3

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。

注意：在错误重发过程中，TXBUSY保持高电平，直到重发结束；此时TXBUF中的数据也不会写入TXSR，TXIF不会置位，直到重发完成，才将TXBUF中的数据写入TXSR并且置位TXIF。

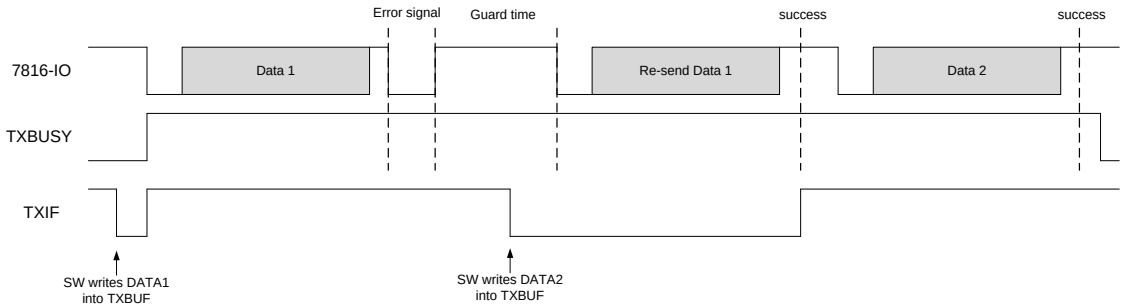


图 19-107816 发送和重发

19.6.5 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(RXBUFFER)，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。

19.6.6 RS232 硬件流控

UART模式下，通过nCTS输入和nRTS输出，可以实现两个设备之间的数据流控制，如下图：

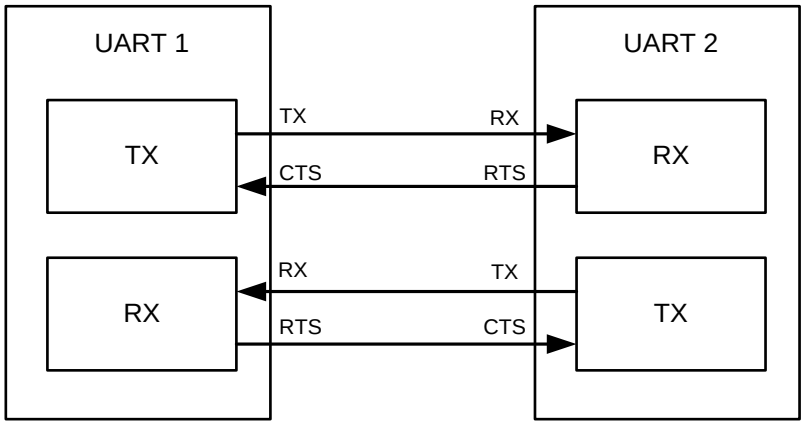


图 19-11 两个 UART 之间的硬件流控

RS232 RTS和CTS流控功能由RTSEN和CTSEN寄存器分别独立控制。

当RTSEN=1，nRTS在UART接收电路准备就绪时保持低电平，当UART数据缓冲区满且接收移位寄存器满，此时无法继续接收数据，nRTS信号拉高指示对方暂停发送新数据。当数据缓冲区被读取后，nRTS重新拉低通知对方可以发送新数据。

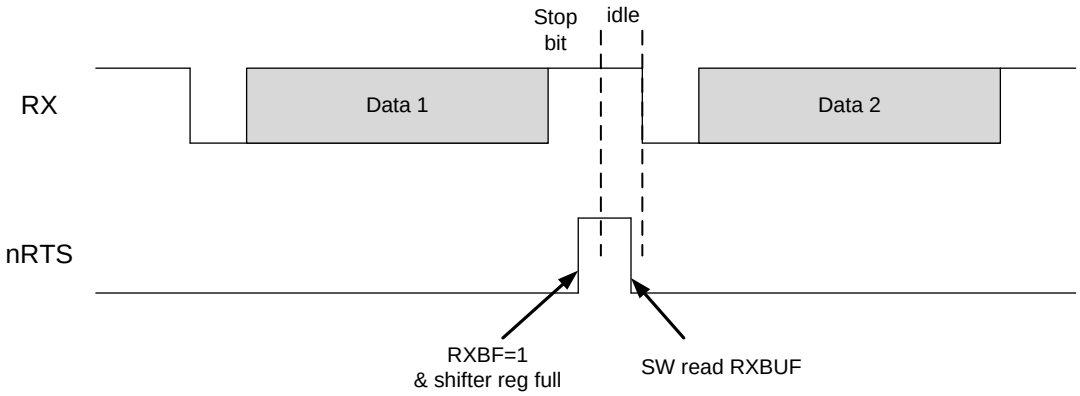


图 19-12nRTS 流控

当CTSEN=1，UART发送数据前先检查nCTS输入，如果nCTS为低电平，表示对方可以接收数据，此时UART正常发送；如果nCTS为高电平，表示对方暂时不能接收，则UART推迟发送直到nCTS变成低电平。在CTSEN=1的情况下，nCTS输入翻转时CTSIF标志自动置位，并且可以产生中断。

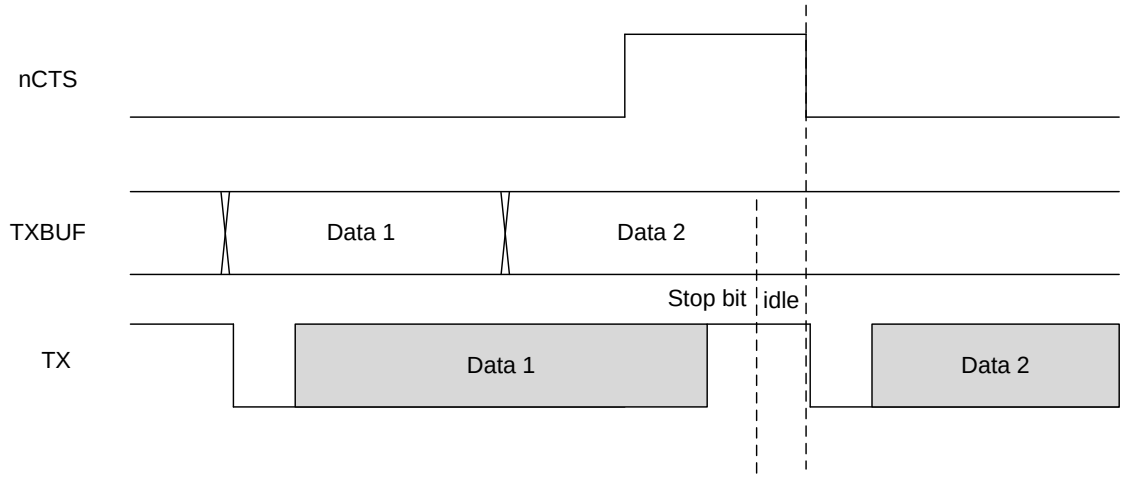


图 19-13nCTS 流控

19.6.7 低功耗休眠唤醒

UART支持两种唤醒模式

1）UART模式：RXD下降沿触发的芯片休眠唤醒。

UART模式下，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UART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
- 配置UART工作时钟为RCHF，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

2）LIN模式：从机支持总线唤醒信号检测。

LIN从机模式下，通过置位WKDET_EN寄存器，LIN模块持续检测总线信号，当出现隐性到显性电平转换，并保持显性电平时间长度符合LIN协议要求后，触发LIN唤醒事件。

软件配置方法如下：

- 配置UART寄存器，使能WKDET_EN
- 将对应GPIO配置为UART数据接收功能
- 关闭UART发送和接收使能
- 软件设置芯片进入休眠，等待LIN唤醒事件

19.6.8 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 通道配置为 UART0_RX
- 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

19.6.9 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

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	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

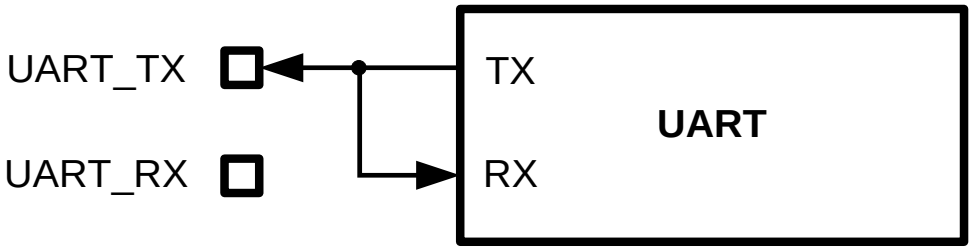
表 19-4 DMA 发送中断

19.6.10 半双工单线通信

当HDSEL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LIN和智能卡模式下使用，即置位HDSEL时必须保证MODE=0。

当HDSEL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



19.7 波特率发生

19.7.1 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。

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，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 19-5 常用时钟频率下波特率计算

19.7.2 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。

19.8 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：

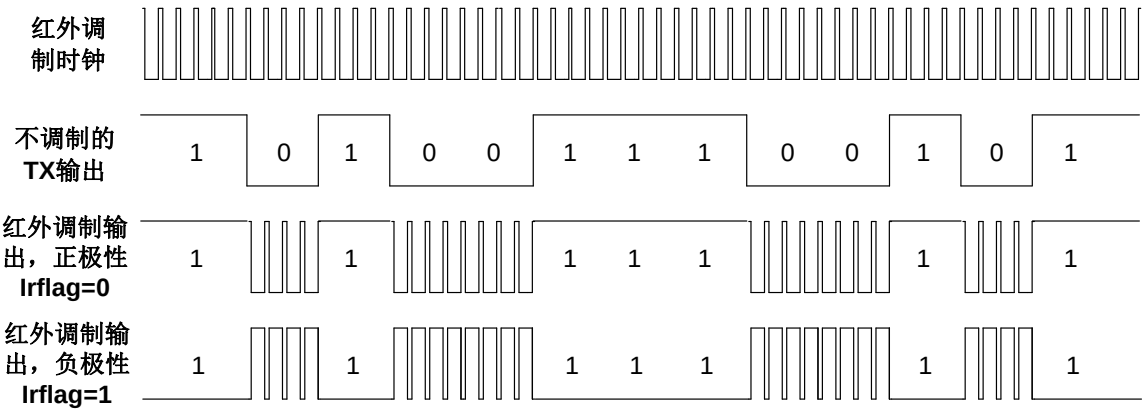


图 19-14 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

19.9 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。

19.10 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。

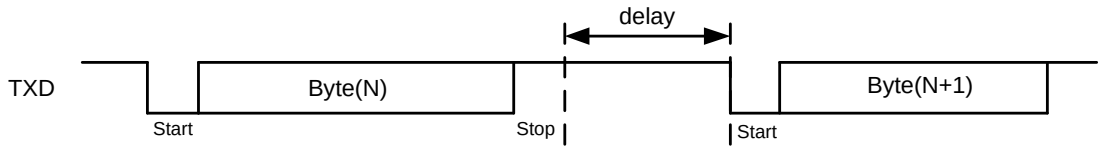


图 19-15 UART 发送延迟

19.11 LIN 总线通信功能

19.11.1 概述

LIN（Local Interconnect Network）是一个低速低成本串行通信协议，通信波特率1~20Kbps，支持单主机和最多15个从机互联。每个节点通过单线收发器接入总线，收发器与控制器之间通过类似UART的接口连接。

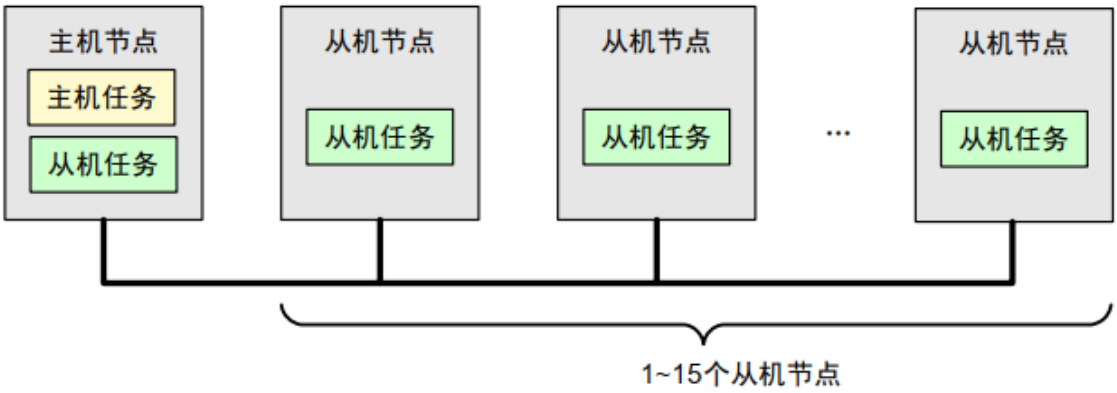


图 19-16LIN 总线拓扑

LIN总线的帧结构包含帧头和应答，帧头总是由主机发送，主机发送帧头后可以继续发送数据，也可以接收从机发送的应答数据。

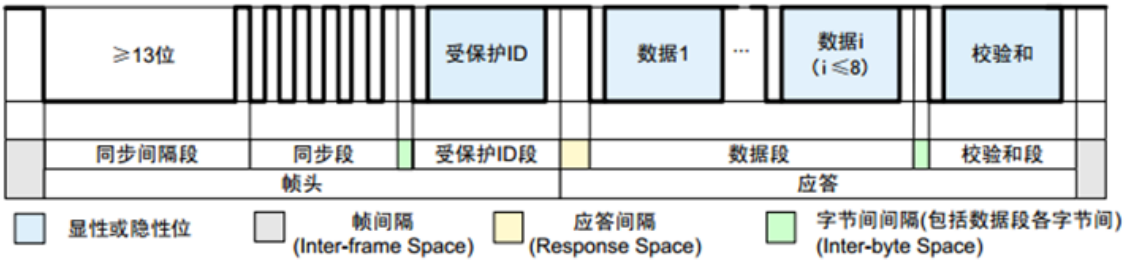


图 19-17LIN 总线帧结构

除了同步间隔段，其他各段都是通过标准UART数据传输格式的格式传输的。在 LIN 帧中，数据传输都是先发送LSB(Least Significant Bit，最低有效位)，最后发送 MSB(Most Significant Bit，最高有效位)。

19.11.1.1 Break Field

同步间隔段（Break Field）为至少13bit低电平和间隔字符组成。总线波特率总是由主机决定的，从机检测到总线上持续至少11bit的显性电平时，认为是帧的开始。



图 19-18LIN break field

当从机节点选择的时钟(精度不高的时钟)在容限范围内(±14%)时， (13 - 11.18) / 13= 14%，即是说当处于最差情况下(时钟相差 14%)时，从机任务按照自身时钟测量的主机节点发送的 13位显性电平不会低于 11.18 位，若识别阈值高于 11.18 位，那么当选用 14%的时钟时，就会出现主机发送同步间隔，而从机检测不到的情形。由于在除同步间隔段以外，帧中任何其余部分都不会发送超过 9 位的显性电平(可以参照本章后面几节的内容)， (10.26 - 9) / 9 = 14%，即是说判断阈值必须大于10.26位，否则可能把帧中其余部分误判作为同步间隔段。综上，识别阈值为 11 位显性电平。

19.11.1.2 Sync Field

同步段用于使从机完成波特率时钟同步，从而可以降低总线对从机时钟精度的要求。UART-LIN模块用作LIN从机时支持基于同步段的波特率自适应

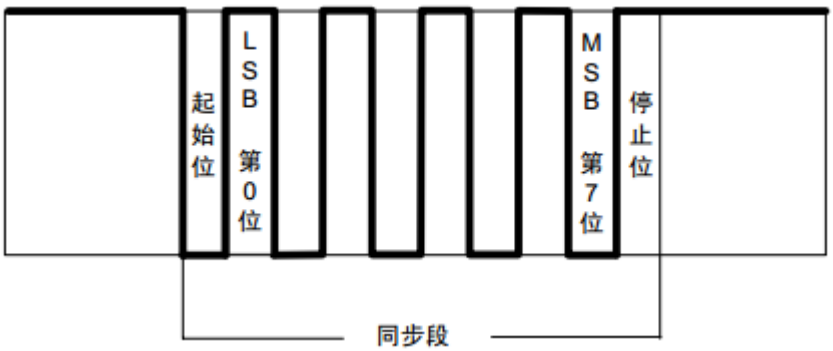


图 19-19LINsync field

LIN 同步以下降沿为判断标志，采用字节 0x55(转换为二进制为 01010101b)。

$$1位时间 = \frac{第7位的下降沿时刻 - 起始位的下降沿时刻}{8}$$

19.11.1.3 PID Field 和 Data Field

PID段的前 6 位叫作帧 ID(Frame ID)，加上两个奇偶校验位后称作Protected ID。

帧 ID 的范围在 0x00～0x3F 之间，共 64 个。帧 ID 标识了帧的类别和目的地。从机任务对于帧头作出的反应(接收/发送/忽略应答部分)都是依据帧 ID 判断的。如果帧 ID 传输错误，将会导致信号无法正确到达目的地，因此引入奇偶校验位。

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \neg (ID1 \oplus ID3 \oplus ID4 \oplus ID5)$$

由公式可以看出，PID 不会出现全 0 或全 1 的情况，因此，如果从机节点收到了“0xFF”或“0x00”，可判断为传输错误。

帧的类型		帧 ID
信号携带帧	无条件帧	0x00 ～ 0x3B
	事件触发帧	
	偶发帧	
诊断帧	主机请求帧	0x3C
	从机应答帧(注 1)	0x3D
保留帧		0x3E，0x3F

注：1. 从机应答帧是一个完整的帧，与帧结构中的“应答”(帧的一部分)不同，注意区别。

图 19-20LIN 帧 ID

节点发送的数据位于数据段，包含 1 到 8 个字节，先发送编号最低的字节，编号依次增加。

协议没有规定帧中的哪一部分显示数据长度码的信息，数据的内容和长度是由系统设计者根据帧 ID 事先约定好的。

一般情况下，对于一个帧中的应答，总线上只存在一个发送节点，否则就会出现错误。事件触发帧例外，可能存在零个、一个或多个发布节点。

PID和数据段的收发与标准UART相同，LIN模块无需特殊处理。

19.11.1.4 Checksum Field

校验段对帧中所传输的内容进行校验，校验对象包含：

- 经典校验（LIN 1.x）：所有数据段

- 加强校验（LIN 2.x）：所有数据段以及PID段

校验方法为将校验对象的各字节作带进位二进制加法，并将所得最终的和逐位取反，以该结果作为要发送的校验和。接收方根据校验和类型，对接收数据作相同的带进位二进制加法，最终的和不取反，并将该和与接收到的校验和作加法，如果结果为 0xFF，则校验和无误，这在一定程度上保证了数据传输的正确性。

校验功能由软件实现。

19.11.2 LIN 主机操作

在主机模式下，LIN模块能够向总线发送唤醒信号、间隔段、同步段、数据段。

19.11.2.1 唤醒发送

唤醒信号由软件置位TX_WKUP寄存器发送，信号长度为WKUP_LEN寄存器定义时钟周期数，默认9个LINCLK时钟，约合270us。最大支持256个时钟周期，约7.6ms。LIN协议规定的唤醒信号长度是250us~5ms。当唤醒信号发送完成后，TX_WKUP寄存器被自动清零，同时LINWKTF发送完成标志置位。

19.11.2.2 Break 和 Sync 发送

Break field由软件置位TX_BREAK寄存器发送，主机发送的break长度由BLT寄存器配置，支持13~28个baud。break发送完后，跟随发送break间隔符，长度由BDT寄存器配置，支持1~4个baud。

同步段由软件发送0x55实现，数据段和校验段的发送与标准UART数据发送相同。

为了实现无缝的数据发送，软件可以在LIN发送Break段的过程中，预先置位TXEN并向TXBUF写入0x55或其他数据，当硬件发送完Break段后，会自动启动后续数据的发送。

主机发送完Break和PID后也可以主动转入接收，此时数据接收与标准UART接收相同。

19.11.2.3 数据发送

LIN主机的数据发送与UART类似，区别是根据IBS寄存器的设置，硬件可能会在发送字节之间插入inter-byte space。根据RSS寄存器的设置，硬件也会在发送完帧头后，插入response space，然后再进行数据发送。

校验和由软件计算，校验和的发送与数据字节相同。

19.11.3 LIN 从机操作

从机模式下，LIN需要支持唤醒信号和间隔段接收，以及基于同步段的波特率自适应。使能LIN从机后唤醒信号和Break检测电路自动使能。

19.11.3.1 唤醒检测

LIN总线处于休眠状态时，主/从机节点都可以向总线上发送唤醒信号，唤醒信号持续 250us～5ms。其余节点（除发送唤醒信号以外的节点）以大于150us为阈值判定唤醒信号。每个从机节点必须在唤醒信号显性脉冲的结束处算起100ms以内准备接收来自主机的命令(帧头)；主机节点也必须被唤醒，100ms之内主机节点发送帧头开始通信。

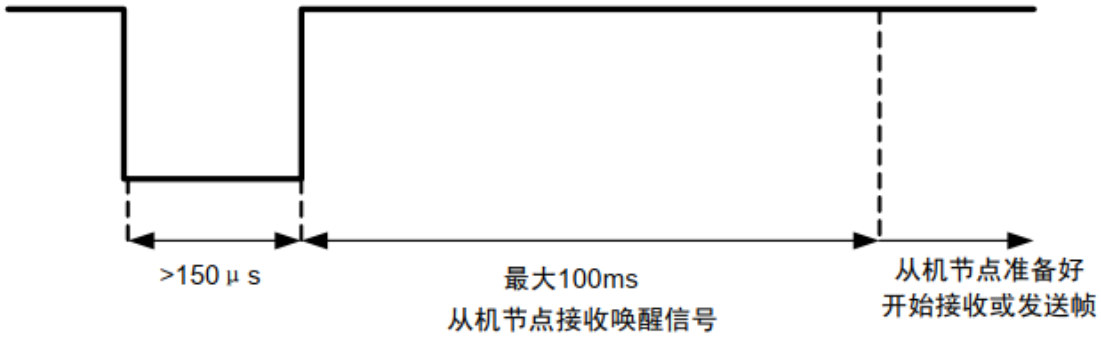
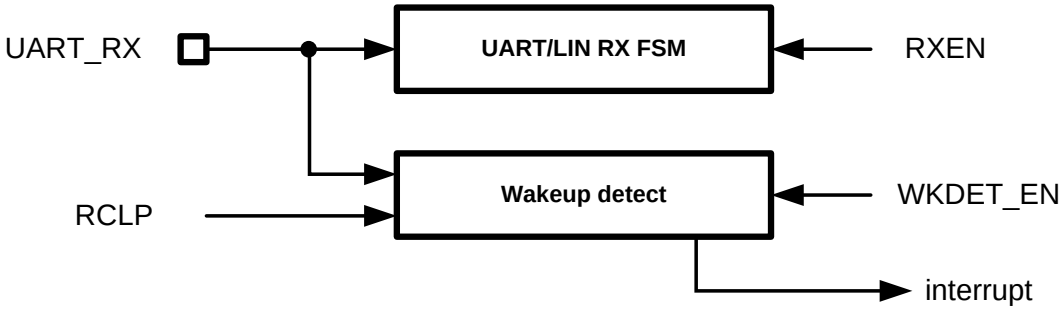


图 19-21LIN 总线唤醒信号

唤醒信号识别通过一个低速时钟驱动的计数器实现。当软件置位WKDET_EN寄存器后，计数器开始监视RXD输入，RXD为高电平情况下计数器保持复位，RXD下降沿启动计数器开始计数，当RXD保持低电平时间达到WKUP_LEN所定义的时钟个数时，认为唤醒信号有效，产生唤醒中断，并关闭计数器，直到下一个RXD下降沿到来。

唤醒计数器使用LINCLK工作，即RCLP，独立于APBCLK，因此这部分功能可以在休眠模式下使用。RCLP时钟周期大约为30us，软件可以通过WKUP_LEN寄存器设置唤醒计数器的阈值，满足最少150us即可。当RXD保持低电平长度超过WKUP_LEN设置后，LINWKF中断标志寄存器被置位，表示LIN模块识别到总线上有效的唤醒信号。



注意：

WKUP_LEN=N，实际能够唤醒从机的最短电平长度是N-2个RCLP时钟周期。

唤醒条件满足后，唤醒标志将在低电平采样值=N时置位。

唤醒检测仅用于产生标志，当芯片处于休眠模式时，LIN唤醒检测标志可以用于唤醒MCU。

在LIN模块处于正常通信时，应关闭唤醒检测功能（WKDETEN=0）。

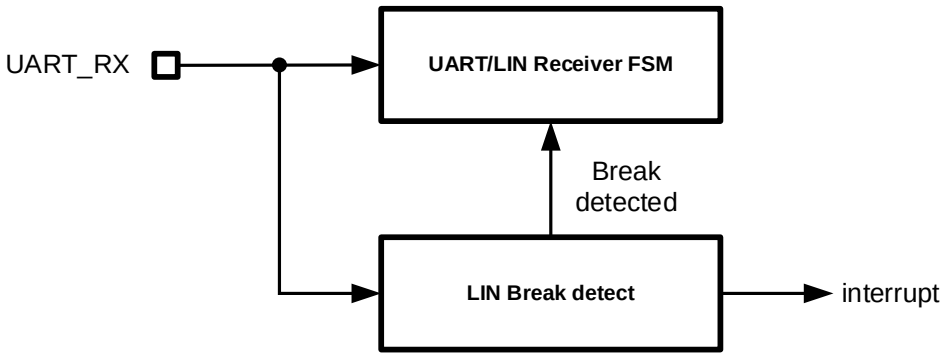
软件应仅在关闭UART收发使能的情况下，打开LIN唤醒检测。

举例：

应用希望检测大于150us的唤醒信号，相当于大约5个RCLP周期，此时需要将WKUP_LEN设置为7，则接收到WKUP低电平长度大于5个RCLP周期时即认为唤醒有效，而唤醒检测标志将在下降沿之后第7个RCLP处置位。

19.11.3.2 Break 检测

Break检测模块独立于LIN从机接收，因此在LIN通信的任何阶段，都能够识别主机发送的Break，并将接收状态机置于初始状态，开始新的一帧接收。



从机处于唤醒状态时（LIN_SLEEP=0），首先要接收Break段。从机根据设定的波特率采样RXD，当接收超过连续11bit低电平，认为检测到合法的Break并置位break检测中断，Break结束中断将在Break结束后置位。

Break低电平阶段使用软件预设的波特率来进行检测，低电平长度超过11 baud即置位break detected标志，检测到上升沿后置位break ended标志，同时记录收到的Break的波特数；随后在SYNC段中，根据前2个下降沿的时间差计算主机baud值（参见Pre-Sync），使用这个值重新验证收到的Break长度是否满足超过11 baud，即应满足：

preset-baud-rate * break-baud-count >sync-baud-rate * 11

如果上式不满足，则置位break error寄存器。

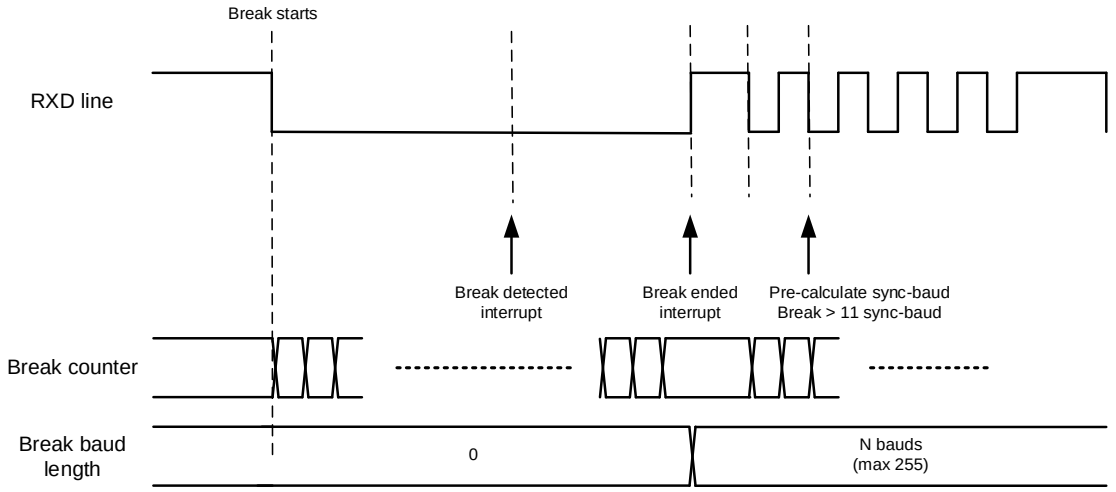


图 19-22Normal Break

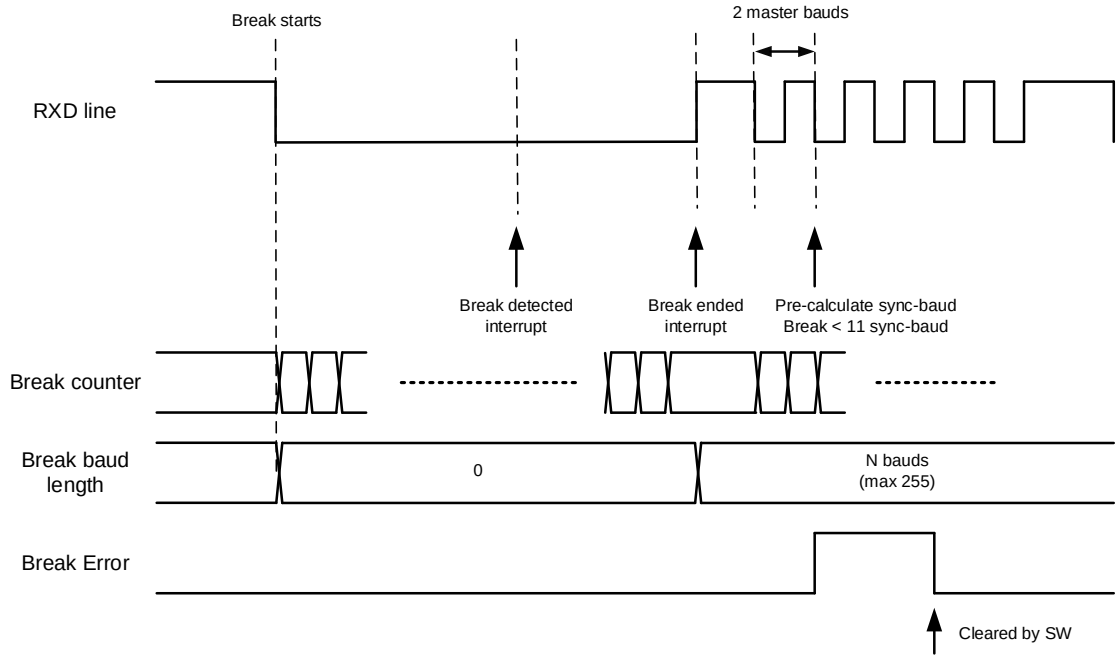


图 19-23Break Error

由于协议没有规定break长度上限，当break持续很长时间显性电平时，LIN模块支持timeout事件中断。timeout长度由寄存器定义（单位baud），通常规定：

timeout time = 48 + (number of data bytes + 1)*14 [baud]

timeout定时器在LIN模块检测到Break下降沿时开始计数，在收到校验和后清零。如果发生了timeout事件，LIN模块置位超时中断、停止当前frame接收，并回到IDLE等待下一个break的下降沿。

因此对于超长break段，LIN将产生超时中断，而不会进行后续的frame接收。发生Break超时后，LIN从机回到IDLE状态，等待下一个RXD下降沿到来。

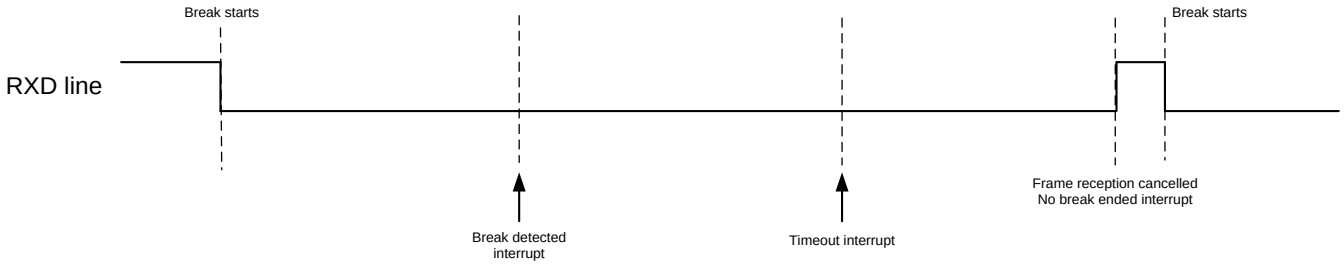


图 19-24Break 超时

与Break接收相关的错误状态：

- Break过长：报超时错误
- Break过短：不识别为Break，LIN从机状态机不动作
- 不发Break直接发送数据：效果等同于Break过短
- 在数据接收过程中收到Break：触发帧格式错误（stop=0）

与Break发送相关的错误状态：

- 发送电平与回环电平不一致：报biterror

特殊情况：

主机发送Break后没有发送SYNC，而是重新发送Break。为了应对这种特殊情况，LIN从机的Break检测应该是并行于接收状态机的，即从机可以在任何状态下识别新的Break并视为主机的一次重新发送过程，此时应终止当前接收过程，准备接收新的帧。

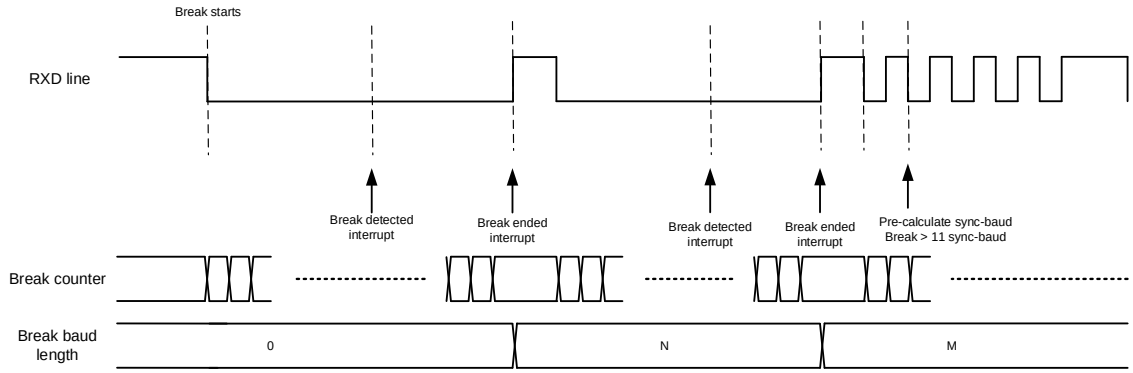


图 19-25 连续 Break 后正常接收

19.11.3.3 波特率同步

在波特率同步状态下，软件置位SYNCEN使能波特率自适应电路，波特率检测模块测量同步段的宽度，软件计算波特率，同步段结束后软件更新波特率寄存器。同步段接收过程中接收电路不工作，

不会将任何数据写入RX DATA寄存器。

如果软件清零SYNCEN，则LIN从机不识别Sync段，不做波特率计算，如果主机发送sync段，从机将作为数据接收。SYNCEN=0可以在必要时改用Timer来实现软件处理的波特率自适应，处理完成后再由LIN硬件进行数据收发。

由于LIN波特率最低1Kbps，考虑到系统主频较高的情况下，需要较长的计数器才能计算8bit总长度，将波特率同步计数器位宽扩展到20bit，这样在波特率计数器为16bit的情况下，足够容纳8个最长baud位的累加长度。

Pre-Sync操作

由于从机和主机之间可能存在较大的初始波特率误差，为了提高同步段识别的正确率，从机首先计算同步段前2个下降沿之间的时间间隔，得到一个初始波特率，后续同步段的接收识别使用这个初始波特率进行，如果识别到正确的0x55，从机将同步段内记录的平均波特率自动更新到BAUD寄存器。如果没有这个功能，当主机从机波特率误差较大，如10%（仍小于协议要求的14%），从机将无法正确接收0x55，此时会导致SYNC段错误。

Pre-sync阶段的2bit是使用预定义波特率接收的，这里的隐含假设是双方误差不会大到影响2bit的接收。而后续到Stop位的8个bit接收都是使用pre-sync后的波特率进行的。

注意：

- *在SYNCEN=1的情况下，如果break段后接收到sync段非0x55，则产生sync错误中断标志。*
- *波特率自适应仅在接收到Break之后执行*

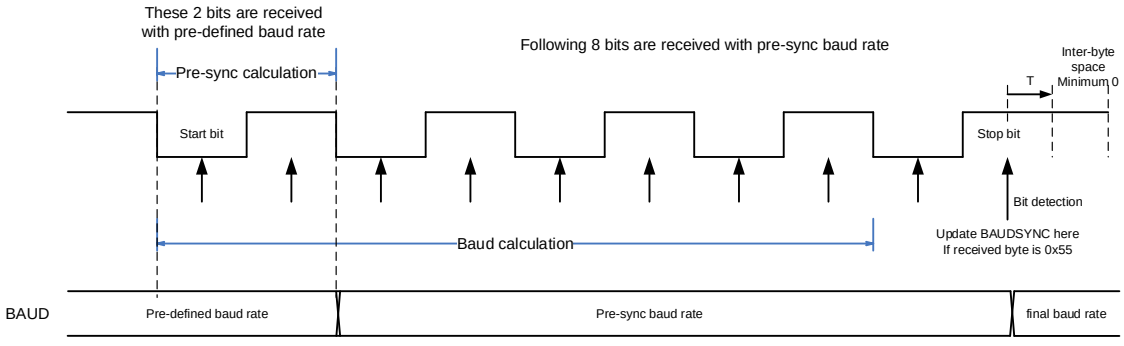


图 19-26LIN 波特率自适应过程

最终的波特率寄存器在stop位检测后更新，更新结果为第一个和第五个下降沿之间的时间间隔除以8得到。

下图举例说明了初始波特率误差过大时，如果没有pre-sync过程，0x55接收将出错（LIN不允许字节内re-sync）

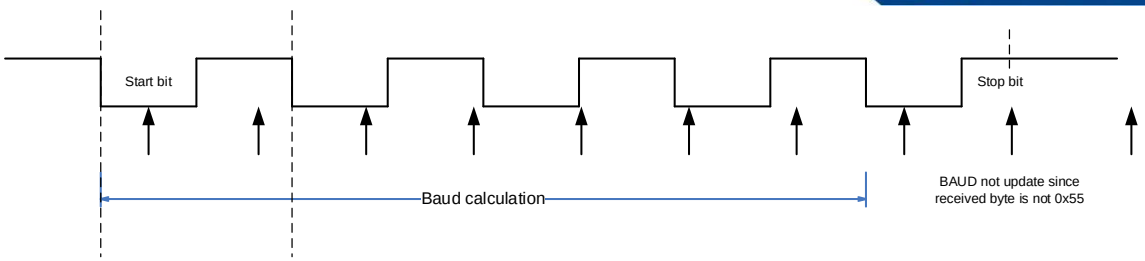


图 19-27 波特率误差较大，无 pre-sync

特殊情况：

主机在Break后发送了0xFF或者0x00，此时从机无法检测到第二个下降沿，这个时候从机仍是按照软件设置的预定义波特率进行接收，不会发生pre-sync，收到的数据也不是0x55，最终报syncerror。

19.11.3.4 从机接收

LIN从机置位RXEN后，状态机开始接收数据，首先做break检测。由于LIN模块并不知道总线数据帧有多少字节，这个信息是总线协议层面事先约定的，所以LIN模块完成break检测和sync检测后会一直保持数据接收状态。软件在收到协议层规定的数据字节和checksum之后，确认当前帧已经接收完毕，此时需要清零RXEN使状态机回到初始态，然后重新置位RXEN使状态机重新进入break检测状态。而wakeup检测是一个独立电路进行的，与LIN模块接收状态机无关。

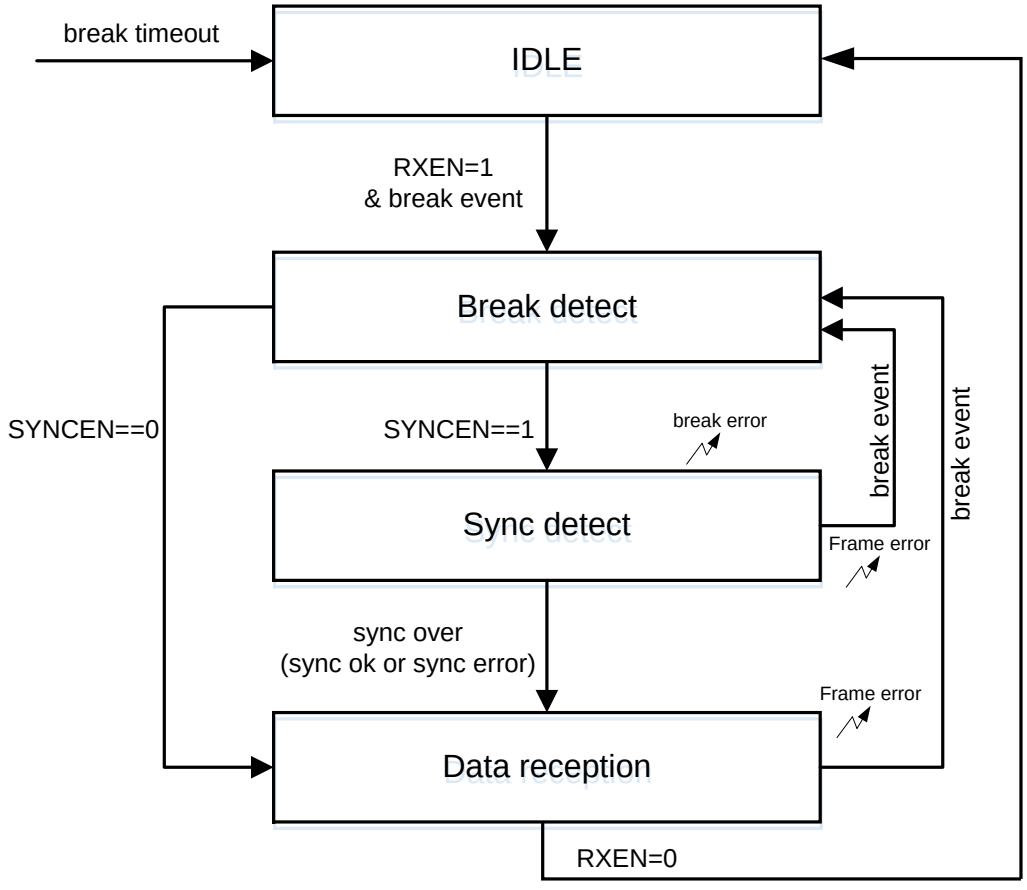


图 19-28 LIN 从机接收流程

19.11.3.5 应答发送

LIN从机的数据发送与UART类似，区别是根据IBS寄存器的设置，硬件可能会在发送字节之间插入inter-byte space。根据RSS寄存器的设置，硬件也会在发送完帧头后，插入response space，然后再进行数据发送。

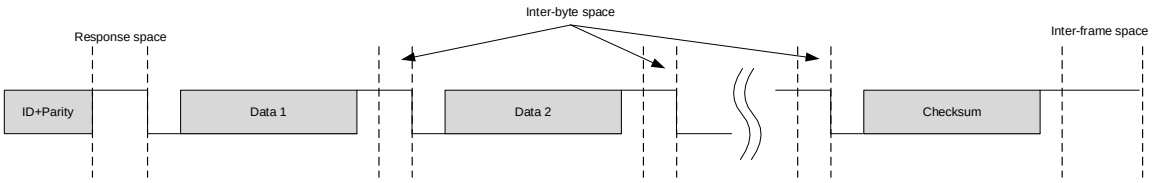


图 19-29 字节间隔和应答间隔

校验和由软件计算，校验和的发送与数据字节相同。

19.11.3.6 从节点休眠

根据LIN2.1协议，当主机保持总线为隐性电平或者显性电平4~10s，无电平转换时，从节点应进入休眠状态。

从节点休眠有两种进入方式：

1）LIN总线保持长时间隐性电平：由于长时间没有检测到Break，MCU主动进入休眠，进入休眠前需要置位WKDETEN寄存器，使能LINwakeup检测功能。

2）LIN总线保持长时间显性电平：从机会检测到Break超时错误，MCU主动进入休眠，进入休眠前需要置位WKDETEN寄存器，使能LINwakeup检测功能。

注意，为了避免休眠中产生错误的break检测，软件休眠前应清零RXEN，关闭数据接收功能。

19.11.4 LIN 接收采样与噪声检测

LIN模式下，数据接收时固定采用16倍过采样模式，OVSM寄存器无效。同时LIN模式支持总线噪声检测，当3个连续的数据采样点得到的采样数据不一致时，NF标志置位，表示检测到总线噪声。即使NF标志置位，不影响3中取2的有效数据判决逻辑。

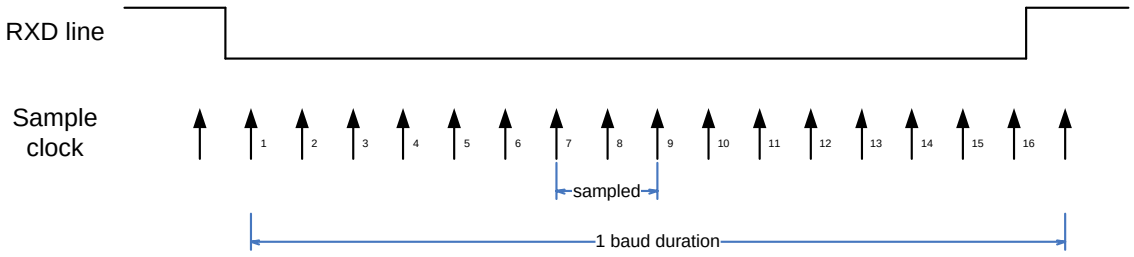


图 19-30LIN bit sampling

数据判决和噪声检测标志：

采样点 7、8、9	数据位判决	噪声标志 NF
000	0	N
001	0	Y
010	0	Y
011	1	Y
100	0	Y
101	1	Y
110	1	Y
111	1	N

注意，LIN2.1 协议中规定 LIN 通信不支持 re-synchronization，因此数据同步仅发生在每个字节的起始位，此时必须设置 RESYNCEN=0.

19.11.5 LIN 错误状态

19.11.5.1 发送数据冲突检测（Bit Error）

配置 BERM 寄存器可以使能发送数据冲突检测功能（仅 LIN 模式下有效）。冲突检测原理是在发送数据时，同时采样 RXD 接收数据，并进行比对，如果收发数据不一致，则置位 bit 冲突中断 BERIF。

数据冲突检测针对主机发送的所有 bit 位，包含 break 段、sync 段、数据段和校验和段。

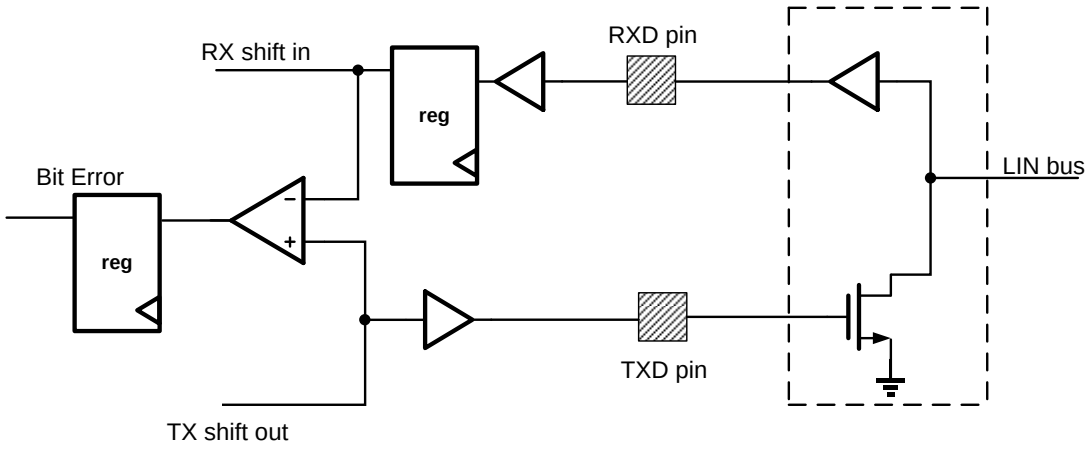


图 19-31LIN bit error 冲突检测

数据冲突检测可以发生在 bit 发送的第 9 个时钟或第 13 个时钟：

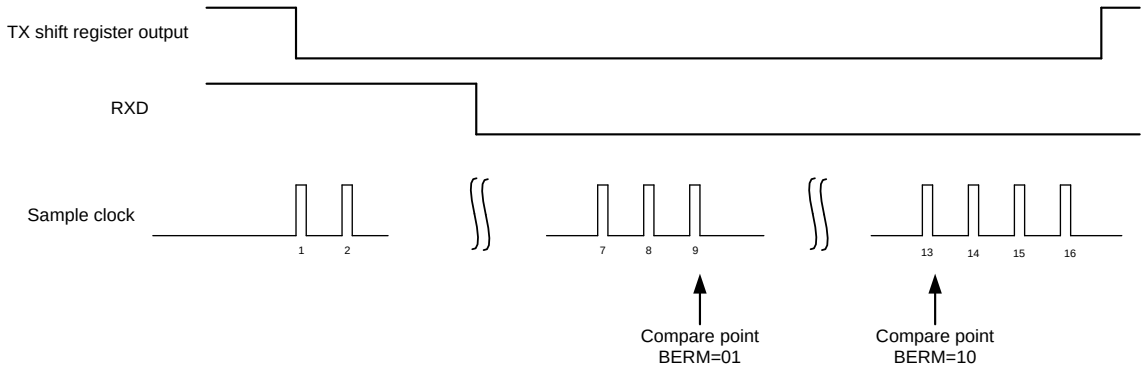


图 19-32LIN bit error 监测点

注意在 sync 段之前由于存在可能的波特率误差，BER 检测时间点可能存在偏差。WKUP 信号不做位冲突检测。

当 BERIF 置位时，LIN 模块采取以下动作：

- 当前发送仍会完成
- 软件重新发送前应先清零 BERIF

注意：由于 WKUP 发送完全独立于 UART 状态机，因此 WKUP 信号不参与发送冲突检测。为了避免 BERIF 误置位，建议在发送唤醒信号时不要使能 BERM 寄存器。

19.11.5.2 timeout 错误

timeout 错误检测 break 连续低电平长度（总线下降沿到上升沿的间隔）是否超出了 FTO 寄存器定义的总 baud 数。FTOEN 寄存器设置是否使能超时功能。

当总线检测到下降沿，LIN 内部超时计数器自动启动计数，检测到上升沿后定时器自动停止并清零。此功能用于检测总线异常的低电平锁死。

根据协议，frame 超时时间通常由下式定义：

frame timeout time = 48 + (number of data bytes + 1)*14 [baud]

这个帧超时检查需要由软件来完成。

19.11.5.3 帧格式错误

数据接收过程中，检测到 stop bit 为低电平。此时停止后续帧数据接收，并置位 FER 标志寄存器。如果主机在数据帧的中间发送 Break，从机将检测为帧格式错误。

19.11.5.4 sync 错误

在 SYNCEN=1 的情况下，LIN 从机预期会在 break 段后收到 sync 段。如果 break 段之后收到的字节不是 0x55，则从机认为是 sync 段错误，置位 SER 标志，并且可以产生中断。

SYNC 错误有两种可能性：

1） 主机发送的 SYNC 段波特率偏差过大

2）主机没有发送正确的 SYNC 段

针对错误 1），LIN 从机在接收到 SYNC 段的头两个下降沿时，会自动更新接收使用的波特率，并以新波特率接收后续数据，以校验是否是 0x55。但是即使 0x55 数据正确，波特率过低使得之前收到的 Break 长度不足 11 bit，也是违反协议的（波特率高不违反协议，因为协议只规定了 Break 长度最低值）。因此 LIN 从机需要记录收到的 Break 长度（单位 baud），并且在收到 SYNC 头两个下降沿后判决 Break 长度是否大于 11 个 baud，如不满足则置位 Break 错误标志。

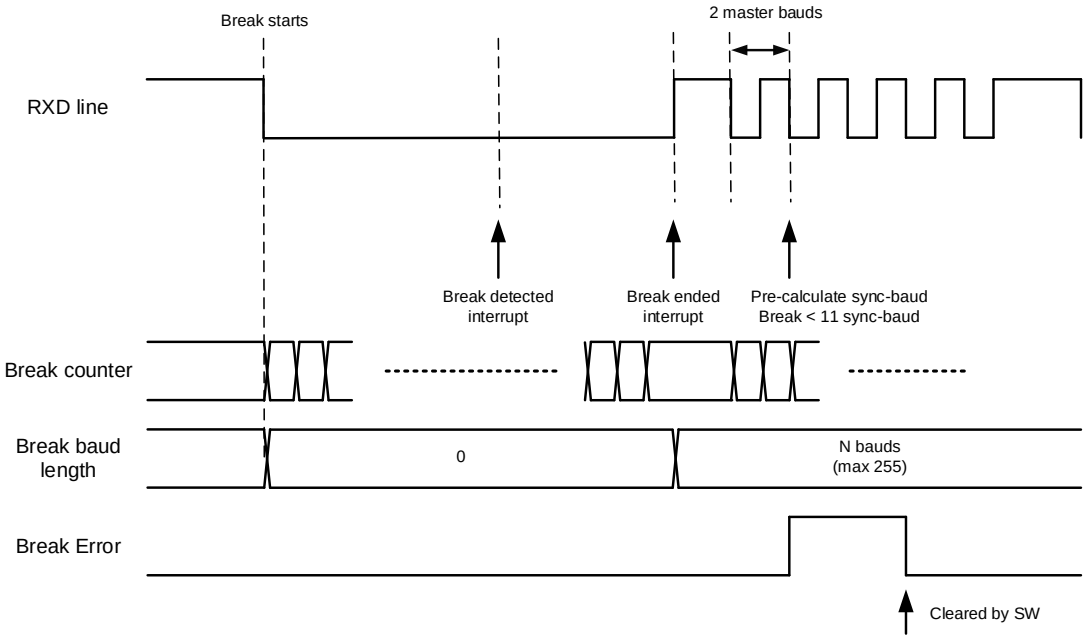


图 19-33Pre-sync 后发现 Break 长度不足，BreakError

针对错误 2），LIN 从机将识别出非 0x55 数据，此时从机置位 SYNC 错误标志。

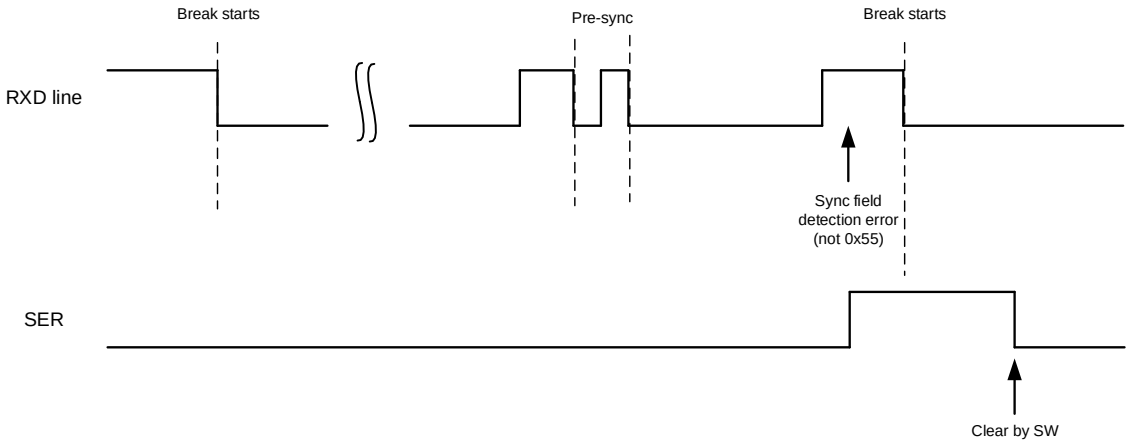


图 19-34SYNC 段数据错误

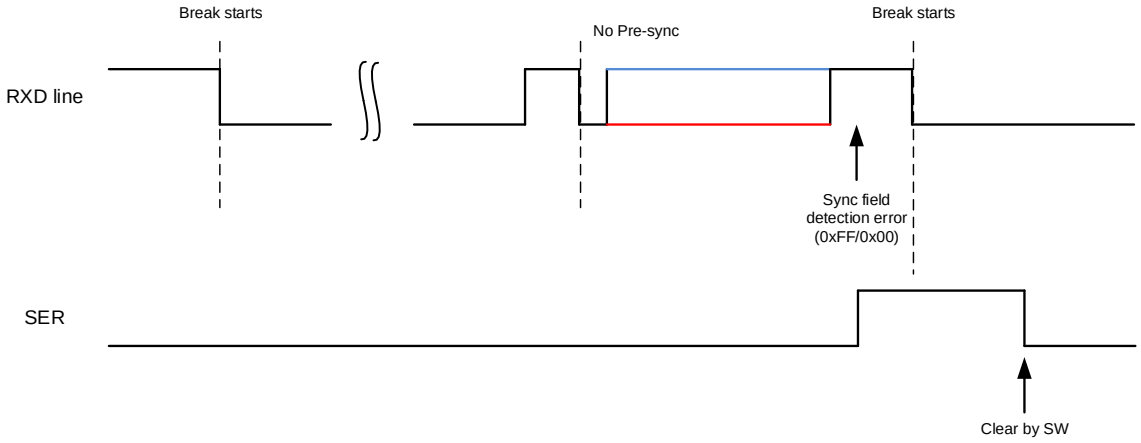
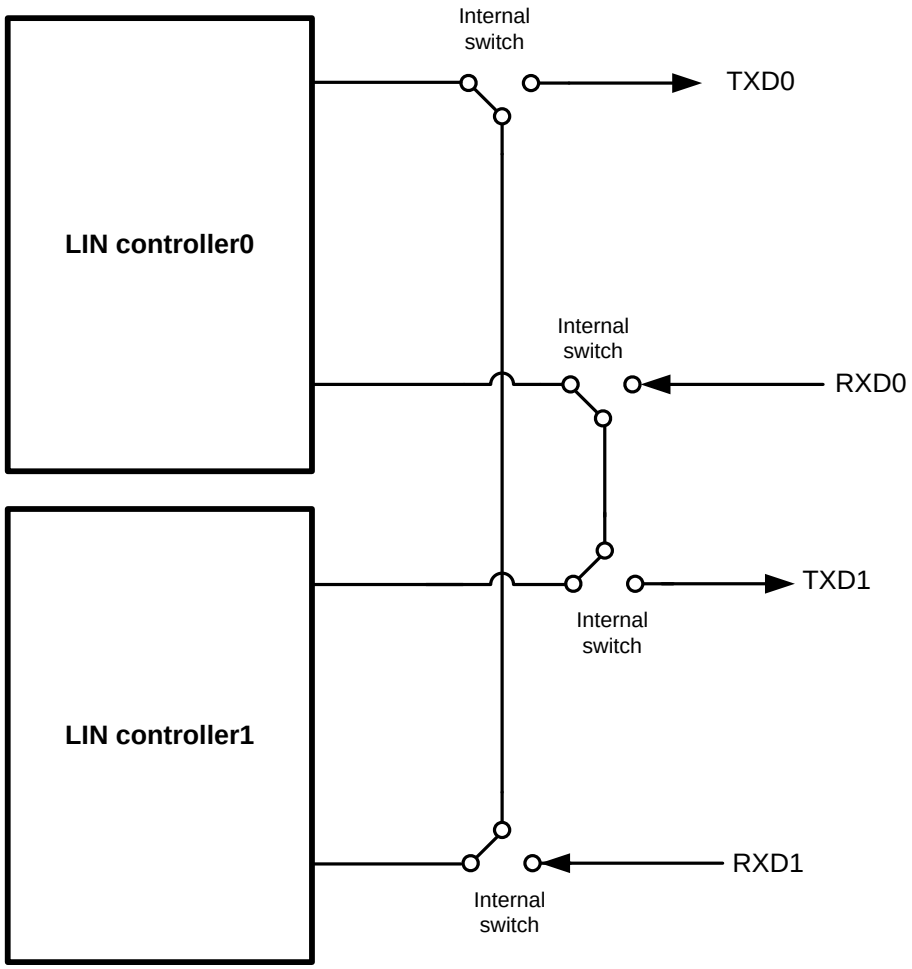


图 19-35SYNC 段 0xFF 或 0x00，无 pre-sync

19.11.6 LIN 自测试模式

LIN 自测试模式需要使用 2 个 LIN 模块实现，LIN 的数据收发与 IO 断开，在 LIN 模块之间实现内部互联：



自测试模式下，通常将 UART0 设置为 LIN 主机，其他设置为 LIN 从机；软件操作主机发送 Break、Sync、数据段、校验和，LIN 从机则自动接收 Break、Sync（自动更新 BAUDSYNC）、数据段和校验和。收发数据正确闭环说明模块功能正常。

模块间自测试互联的配置寄存器仅位于 UART0 模块。

自测试互联设置：

	LINSTC=00	LINSTC=01	LINSTC=10	LINSTC=11
UART0_RXD	self-test disabled	UART1_TXD	UART2_TXD	UART3_TXD
UART0_TXD				
UART1_RXD		UART0_TXD		
UART1_TXD				
UART2_RXD			UART0_TXD	
UART2_TXD				
UART3_RXD				UART0_TXD
UART3_TXD				

LIN 自测试之前，需要配置 LINSTC 寄存器完成内部开关互联，并将参与自测试的模块分别配置成主机和从机。

自测试总是由 UART0 参与和其他模块之间通信来完成。比如配置 LINSTC=01，使用 UART0 和 UART1 进行自测试，流程如下：

- 设置 LINSTC=01
- 设置 UART0 为 LIN 主机，UART1 为 LIN 从机
- UART0 发送 break、sync、data，UART1 完成接收，软件判断收发正常
- 设置 UART0 为 LIN 从机，UART1 为 LIN 主机
- UART1 发送 break、sync、data，UART0 完成接收，软件判断收发正常
- 自测试结束

19.12 寄存器

UART0模块基地址：0x4001 2000

UART1模块基地址：0x4001 6800

UART2模块基地址：0x4001 7000

UART3模块基地址：0x4001 7400

UART4模块基地址：0x4001 7800

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	UART0 LIN 控制寄存器 （UART0 LIN Control Register）	UART0_LINCR
0x28	UART0 LIN 波特率同步寄存器 （UART0 LIN Baud Sync Register）	UART0_LINBSR
0x2C	UART0 LIN 帧超时寄存器 （UART0 LIN Frame Timeout Register）	UART0_LINFTR
0x30	UART0 LIN 发送时序寄存器 （UART0 LIN Transmit Timing Register）	UART0_LINTTR
0x34	UART0 LIN Pre-sync 波特率寄存器 （UART0 LINPre-sync baud register）	UART0_LINPSR
0x38	UART0 LIN Break 长度寄存器 （UART0 LIN Break Length Register）	UART0_LINBKR
0x80	LIN 自测试开关控制寄存器 （LIN Self-test Cwitch Control Register）	UART0_LINSCCR
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_IER

offset 地址	名称	符号
	（UART1 Interrupt Enable Register）	
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 Transmit Buffer）	UART1_TXBUF
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	UART1 LIN 控制寄存器 （UART1 LIN Control Register）	UART1_LINCR
0x28	UART1 LIN 波特率同步寄存器 （UART1 LIN Baud Sync Register）	UART1_LINBSR
0x2C	UART1 LIN 帧超时寄存器 （UART1 LIN Frame Timeout Register）	UART1_LINFTR
0x30	UART1 LIN 发送时序寄存器 （UART1 LIN Transmit Timing Register）	UART1_LINTTR
0x34	UART1 LIN Pre-sync 波特率寄存器 （UART1 LINPre-sync baud register）	UART1_LINPSR
0x38	UART1 LIN Break 长度寄存器 （UART1 LIN Break Length Register）	UART1_LINBKR
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
0x24	UART2 LIN 控制寄存器 （UART2 LIN Control Register）	UART2_LINCR
0x28	UART2 LIN 波特率同步寄存器 （UART2 LIN Baud Sync Register）	UART2_LINBSR
0x2C	UART2 LIN 帧超时寄存器 （UART2 LIN Frame Timeout Register）	UART2_LINFTR
0x30	UART2 LIN 发送时序寄存器 （UART2 LIN Transmit Timing Register）	UART2_LINTTR
0x34	UART2 LIN Pre-sync 波特率寄存器	UART2_LINPSR

offset 地址	名称	符号
	（UART2 LINPre-sync baud register）	
0x38	UART2 LIN Break 长度寄存器 （UART2 LIN Break Length Register）	UART2_LINBKR
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
0x0C	UART3 超时和延迟寄存器 （UART3 Time-Out and Delay Register）	UART3_TODR
0x10	UART3 接收缓冲寄存器 （UART3 Receive Buffer）	UART3_RXBUF
0x14	UART3 发送缓冲寄存器 （UART3 Transmit 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
0x24	UART3 LIN 控制寄存器 （UART3 LIN Control Register）	UART3_LINCR
0x28	UART3 LIN 波特率同步寄存器 （UART3 LIN Baud Sync Register）	UART3_LINBSR
0x2C	UART3 LIN 帧超时寄存器 （UART3 LIN Frame Timeout Register）	UART3_LINFTR
0x30	UART3 LIN 发送时序寄存器 （UART3 LIN Transmit Timing Register）	UART3_LINTTR
0x34	UART3 LIN Pre-sync 波特率寄存器 （UART3 LINPre-sync baud register）	UART3_LINPSR
0x38	UART3 LIN Break 长度寄存器 （UART3 LIN Break Length Register）	UART3_LINBKR
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
0x0C	UART4 超时和延迟寄存器 （UART4 Time-Out and Delay Register）	UART4_TODR
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	UATR5 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

offset 地址	名称	符号
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

19.12.1 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)

19.12.2 UARTx 模式寄存器（UARTx_MCR）

名称	**UARTx_MCR(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	-							
位权限	U-0							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					HDSEL	LINMD	MODE
位权限	U-0					R/W-0	R/W-0	R/W-0

*UART0~3

位号	助记符	功能描述
31:3	-	未实现：读为0
2	**HDSEL**	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	**LINMD**	LIN 模式选择寄存器 0：LIN 主机 1：LIN 从机
0	**MODE**	模块控制寄存器 0：标准 UART 模式 1：LIN 模式

*UART4

位号	助记符	功能描述
31:3	-	未实现：读为0
2	**HDSEL**	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）

19.12.3 UARTx LIN 控制寄存器（UARTx_LINCR）

名称	UARTx_LINCR(x=0~3)							
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	WKUP_LEN							
位权限	R/W-0000 1000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SYNCEN	FTOEN	BERM		NDET_EN	WKDET_EN	TX_BF	TX_WKUP
位权限	R/W-1	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	未实现：读为0
15:8	**WKUP_LEN**	发送唤醒信号时，此寄存器定义 LIN 总线唤醒信号长度 接收唤醒信号时，此寄存器定义 LIN 唤醒信号识别阈值 单位是 LINCLK，即 RCLP；复位值为 8，表示 9 个时钟周期，大约 270us；最大 256 个时钟周期，大约 7.6ms
7	**SYNCEN**	LIN从机Sync段接收使能 0：不识别sync段，不进行波特率计算 1：识别sync段，进行波特率计算
6	**FTOEN**	LIN Frame timeout使能 0：禁止帧超时检测 1：使能帧超时检测，超时阈值由LINFTO寄存器定义
5:4	**BERM**	LIN总线发送冲突检测配置 00/11：关闭发送冲突检测 01：使能发送冲突检测，在发送第9bit处检测 10：使能发送冲突检测，在发送第13bit处检测
3	**NDET_EN**	总线噪声检测使能 (Noise detect enable) 0：关闭LIN总线噪声检测 1：使能LIN总线噪声检测，检测到噪声时NF标志置位
2	**WKDET_EN**	LIN唤醒信号检测使能 0：禁止检测唤醒信号 1：使能检测唤醒信号
1	**TX_BF**	发送 LIN 总线 Break 信号，软件置位，发送完成后硬件自动清零 仅在 LIN 主机模式下有效
0	**TX_WKUP**	发送 LIN 总线唤醒信号，软件置位，发送完成后硬件自动清零 LIN 主机和从机模式下都可以发送唤醒 *注意：TX_BF 和 TX_WKUP 不能同时置位；如果软件同时置位 TX_BF 和 TX_WKUP，硬件只会发送唤醒信号，不会发送 Break*

19.12.4 UARTx LIN 波特率同步寄存器（UARTx_LINBSR）

名称	**UARTx_LINBSR(x=0~3)**							
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	-				BAUD_SYNC			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BAUD_SYNC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BAUD_SYNC							
位权限	R-0000 0000							

位号	助记符	功能描述
31:20	-	未实现：读为0

位号	助记符	功能描述
19:0	**BAUD_SYNC**	同步段计数结果，计算同步段总共 10bit 的时钟个数，用于波特率自适应。 软件在检测到 SYNCF 标志后读取此寄存器，除以 8 后得到当前 LIN 波特率，可用于查询和诊断。 在 SYNCEN=1 的情况下，SYNC 段成功接收后，此寄存器结果被自动除以 8 后写入 UART 波特率产生寄存器，用以设定后续数据通信的波特率，实现波特率自适应。

19.12.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR）

名称	UARTx_LINFTR(x=0~3)							
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					FTO			
位权限					R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FTO							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:20	**-**	未实现：读为0
11:0	**FTO**	Frame timeout 设置 break 段波特率个数结果，此结果用于 break 接收结束后长度记录；软件可读取此寄存器用于查询和计算

19.12.6 UARTx LIN 发送时序寄存器（UARTx_LINTTR）

名称	UARTx_LINTTR(x=0~3)							
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	-		IBS		-	RSS		
位权限	U-0		R/W-00		U-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BDT		BLT			
位权限	U-0		R/W-00		R/W-0000			

位号	助记符	功能描述
31:14	-	未实现：读为0
13:12	**IBS**	inter-byte space，定义数据发送时的数据字节间隔，仅发送时有效 00: 0 baud 01: 1 baud 10: 2 baud 11: 3 baud
11	-	未实现：读为0
10:8	**RSS**	response space，定义 header 到发送数据 response 之间的间隔，仅发送时有效 000: 0 baud 001: 1 baud … 111: 7 baud
7:6	-	未实现：读为0
5:4	**BDT**	Break delimiter 长度 00: 1 baud 01: 2 baud 10: 3 baud 11: 4 baud
3:0	**BLT**	Break 长度 0000: 13 baud 0001: 14 baud ... 1110: 27 baud 1111: 28 baud

19.12.7 UARTx LIN 超时定时器清零寄存器（UARTx_LINTOCR）

名称	UARTx_LINPSR (x=0~3)							
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	-				PRE_SYNC			
位权限	U-0				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRE_SYNC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRE_SYNC							
位权限	R-0000 0000							

位号	助记符	功能描述
31:20	-	未实现：读为0
19:0	**PRE_SYNC**	同步段预同步（前 2bit）计数结果，此结果用于设定同步段后续 8bit 接收波特率；软件可读取此寄存器用于查询和诊断。

19.12.8 UARTx LIN Break 长度寄存器（UARTx_LINBKR）

名称	UARTx_LINBKR (x=0~3)							
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	-				BREAK_LEN			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BREAK_LEN							
位权限	R-0000 0000							

位号	助记符	功能描述
31:12	-	未实现：读为0
11:0	**BREAK_LEN**	Break 段的工作时钟个数，用于 Break 接收结束后的长度记录，供软件查询。

19.12.9 UARTx 控制状态寄存器（UARTx_CSR）

名称	UARTx_CSR (x=0~4)							
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	RTSEN	CTSEN	-				TXIREN	RXTOEN
位权限	R/W-0	R/W-0	U-0				R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESYNCEN	-	OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
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
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-00		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

位号	助记符	功能描述
23	**RTSEN**	RS232 RTS 流控使能，仅 UART 模式有效 0：关闭 RTS 流控 1：使能 RTS 流控
22	**CTSEN**	RS232 CTS 流控使能，仅 UART 模式有效 0：关闭 CTS 流控 1：使能 CTS 流控
21:18	-	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	**RESYNCEN**	字节内重同步使能 (Re-synchronization enable) 0：禁止重同步，仅作起始位同步 1：允许重同步 注意，LIN通信时保持此bit为0
14	-	未实现：读为0
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 *注意：在 LIN 模式下，固定为 16 倍过采样，此寄存器无效*
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)

19.12.10 UARTx 中断使能寄存器（UARTx_IER）

名称	UARTx_IER(x=0~4)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOE_IE	SER_IE	FER_IE	BER_IE	NF_IE	LINSYNC_IE	LINB_IE	LINWK_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	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	CTSIE	-			RXTO_IE	RXERR_IE	-	RXBF_IE
位权限	R/W-0	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	**TOE_IE**	LIN timeout error中断使能，1有效
30	**SER_IE**	LIN sync error 中断使能，1 有效
29	**FER_IE**	LIN framing error 中断使能，1 有效
28	**BER_IE**	LIN 发送冲突中断使能，1 有效
27	**NF_IE**	LIN 噪声检测中断使能，1 有效
26	**LINSYNC_IE**	LIN 同步段中断使能，1 有效
25	**LINB_IE**	LIN Break 段中断使能，1 有效
24	**LINWK_IE**	LIN 唤醒中断使能，1 有效
23:17	**-**	**未实现：读为0**
16	**-**	**未实现：读为0**
15	**CTSIE**	nCTS翻转中断使能，1有效
14:12	**-**	**未实现：读为0**

位号	助记符	功能描述
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)
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)

19.12.11 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0~4)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOER	SER	FER	BERIF	NF	LINSYNCF	LINBF	LINWKF
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	LINWKTF	LINBEF	-			PERR	FERR	OERR
位权限	R/W-0	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	CTSIF	-			RXTO	-		RXBF
位权限	R/W-0	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 -0	R/W-0

位号	助记符	功能描述
31	**TOER**	LIN timeout error 标志，硬件置位，软件写1清零
30	**SER**	LIN sync error 标志，硬件置位，软件写 1 清零
29	**FER**	LIN framing error 标志，硬件置位，软件写 1 清零
28	**BERIF**	LIN 发送冲突标志（bit error），硬件置位，软件写 1 清零
27	**NF**	LIN 总线噪声检测标志，硬件置位，软件写 1 清零
26	**LINSYNCF**	LIN 同步段接收完成中断标志，硬件置位，软件写 1 清零
25	**LINBF**	LIN 从机模式：LIN Break 段接收完成中断标志，硬件置位，软件写 1 清零 LIN 主机模式：LIN Break 段发送完成中断标志，硬件置位，软件写 1 清零
24	**LINWKF**	LIN 唤醒信号检测标志，识别到有效唤醒信号时硬件置位，软件写 1 清零
23	**LINWKTF**	LIN 唤醒信号发送完成中断标志，硬件置位，软件写 1 清零

位号	助记符	功能描述
22	**LINBEF**	LIN 从机模式：LIN Break 段结束中断标志，硬件置位，软件写 1 清零
21:19	-	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	**CTSIF**	CTS 信号翻转标志，当 nCTS 输入发生翻转时置位，软件写 1 清零
14:12	-	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) （仅 UART0 和 UART1 有效）
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)
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**	发送缓存空中断标志，硬件置位，软件写入 TXBUF 时清零 (Transmit Buffer Empty flag)
1	**TXBE**	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)
0	**TXSE**	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)

19.12.12 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(x=0~3)							
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	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)

19.12.13 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF(x=0~4)							
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]

19.12.14 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF(x=0~4)							
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

位号	助记符	功能描述
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]

19.12.15 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR(x=0~4)							
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) 当LIN模式下SYNCEN=1，在PreSync阶段以及成功接收SYNC段后，硬件自动更新此寄存器； 注意，即使 SYNCEN=1，软件仍可以改写 SPBRG

波特率计算详见波特率发生章节

注：当SPBRG <= 0x000F时，UARTDIV=16'H000F;

当SPBRG >0x000F时，UARTDIV=SPBRG;

19.12.16 UARTx 智能卡状态标志寄存器（UARTx_SCISR）

名称	UARTx_SCISR (x=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。即数据及校验位接收之后，无论是否需要重发，都需要及时清除该标志） 0：数据接收空闲
15:12	--	RFU：未实现，读为 0
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：接收数据时无奇偶校验错误
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

19.12.17 UARTx 智能卡中断使能寄存器（UARTx_SCIER）

名称	UARTx_SCIER (x=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

19.12.18 UARTx 智能卡帧格式控制寄存器（UARTx_FFCR）

名称	UARTx_FFCR (x=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-1	R/W-1	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); 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

19.12.19 UARTx 智能卡额外保护时间寄存器（UARTx_EGTR）

名称	UARTx_EGTR (x=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)

19.12.20 UARTx 智能卡输出时钟分频寄存器（UARTx_CODR）

名称	UARTx_CODR (x=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-00011				

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。

20 低功耗 UART（LPUART）

20.1 概述

LPUART 是增强型异步串行通信接口，其工作时钟可以选择 32768Hz 晶振时钟（XTLF）、高频环振时钟（RCHF）。LPUART 可以支持到最高 9600 波特率的数据接收，此时 LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

特点：

- 异步数据收发
- 2路独立LPUART
- 标准UART帧格式
  - 1bit起始位
  - 可配置数据长度，支持6、7、8、9bits
  - 奇校验、偶校验或无校验位
  - 1或2bit停止位
  - 与UARTx共享红外调制输出
- 可编程数据极性
- 当工作时钟为XTLF时，支持Sleep/DeepSleep模式下的数据收发
- 中断标志
  - 接收Buffer满
  - 接收Buffer溢出
  - 接收帧格式错误
  - 接收校验位错误
  - START检测
  - 数据匹配
  - 发送完成
- 休眠模式下唤醒芯片
  - RXD下降沿唤醒
  - 起始位检测唤醒
  - 1字节接收完成唤醒
  - 1字节数据匹配唤醒
- 支持DMA（Sleep/DeepSleep模式下不支持）

20.2 结构框图

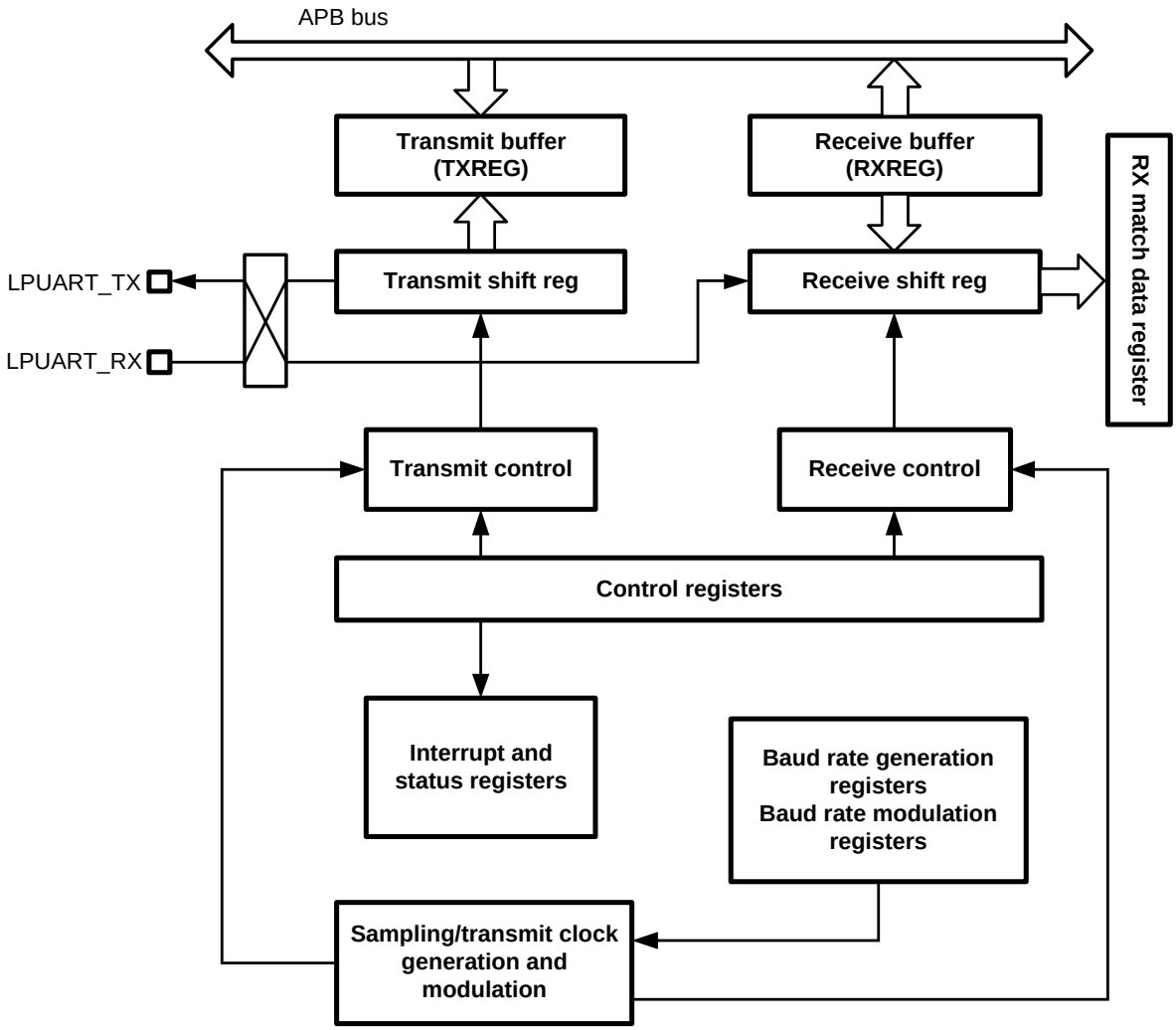


图 20-1 LPUART 结构框图

20.3 工作时钟

LPUART 使用独立于 APBCLK 的时钟进行数据收发，工作前需要在 CMU 模块中配置相关寄存器。

LPUART 使用 XTLF 或 RCLP 工作。由于 RCLP 精度不高，在使用 RCLP 进行 LPUART 通信前必须先进行时钟校准，将 RCLP 校准到+/-1%以内。

20.4 字符描述

LPUART 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。

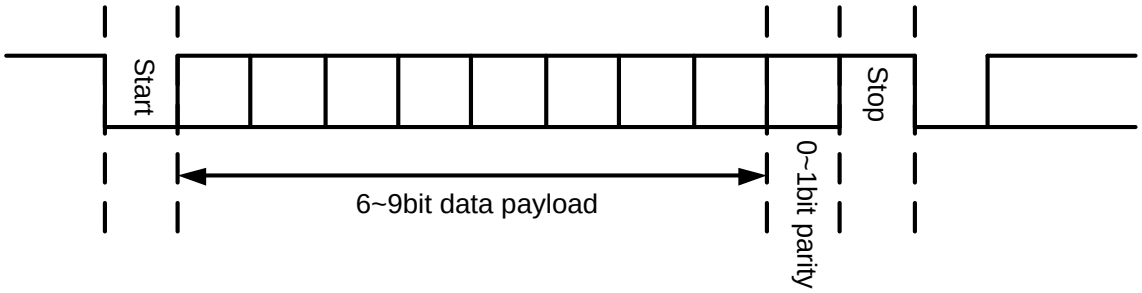


图 20-2 字符描述

LPUART 支持多种帧格式，由 LPUARTxCSR.PDSEL 寄存器和 LPUARTxCSR.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 20-1 LPUART 数据帧格式

[1]：Stop 位可能是 1bit 或者 2bit，根据 STOPCFG 寄存器决定

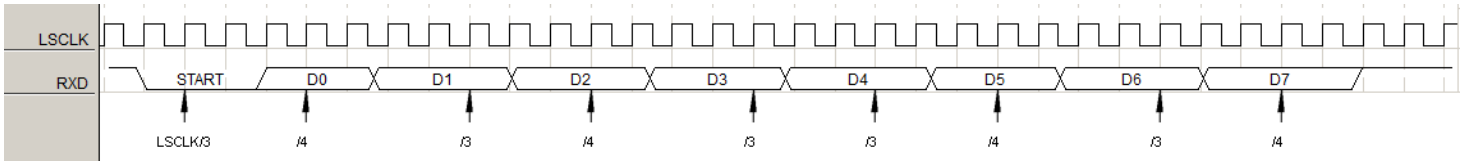
注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

20.5 功能描述

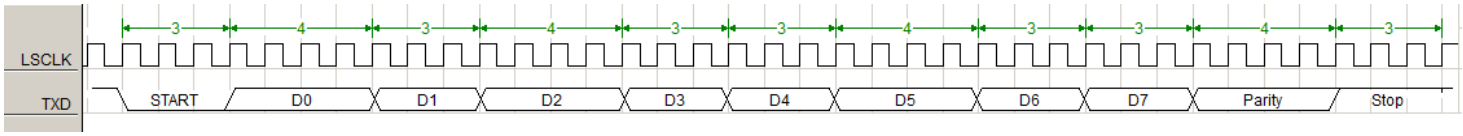
20.5.1 位接收采样和发送

当工作时钟选择为 LSCLK 或 RCHF 分频时，频率仅为 32Khz 左右，此时标准串口无法支持 9600bps 通信，因此需要引入 bit 调制设计。

由于 LPUART 工作时钟不是波特率的整数倍，采用固定分频系数的话会引入累积误差，在接收时采用 3、4 分频交替进行接收，确保在每个 bit 的中间位置采样，每个 bit 采样一次。每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：



类似于 LPUART 接收，LPUART 工作时钟不是波特率的整数倍，采用固定分频系数同样会引入累积误差，在发送时也采用了 3、4 分频交替进行发送，每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：



软件需要根据通信波特率的不同合理配置调制控制寄存器 MCTL，当工作时钟为 32768Hz 时，建议的配置参数表如下：

Baud	MCTL												
	Bit0（start）	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12
9600	0	1	0	0	1	0	1	0	1	0	1	0	0
4800	1	1	0	1	1	1	1	1	0	1	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0	1
1200	0	1	0	0	1	0	0	1	0	0	1	0	0
600	0	1	1	0	1	0	1	1	0	1	1	0	1
300	0	1	0	0	0	0	1	0	0	0	0	1	0

表 20-2 LPUART 数据位调制系数

20.5.2 接收流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开接收使能
- 等待中断事件

20.5.3 发送流程

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性、中断参数等
- 配置LPUEN寄存器打开发送使能
- 等待中断事件

20.5.4 使用 DMA 进行 LPUART 收发

当 LPUART 模块被使能后，LPUART 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 LPUART 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 LPUART 请求，并完成 RAM 和 LPUART 之间的数据搬运。

应用举例：使用 DMA 进行 LPUART1 接收

- 将 DMA 通道配置为 LPUART1_RX
- 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 LPUART1 模块参数
- 使能 LPUART1 模块接收使能 LPUEN.RXEN=1，等待数据接收
- 收到数据后 LPUART1 自动产生 DMA 请求
- DMA 响应请求，读取 LPUART1 接收缓存寄存器，写入指定 RAM 地址

20.5.5 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep、DeepSleep 模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

- 配置LPUBAUD寄存器决定波特率
- 根据波特率选择合适的调制参数，配置MCTL寄存器
- 配置LPUCON寄存器，选择帧格式、极性，通过LPUxCR.RXEV选择唤醒事件为START位、一帧接收完成、一帧数据匹配或RXD下降沿检测
- 配置LPUEN寄存器打开接收使能
- 软件进入Sleep/DeepSleep

20.5.6 DMA 模式下的发送完成中断

当 LPUART 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 LPUART 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LPUART发送
- 关闭DMA通道中断使能
- 置位LPUART TXBE_IE或TXSE_IE寄存器，允许中断产生
- 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
- LPUART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，LPUART产生TXBE或TXSE中断

下表假设 LPUART 通过 DMA 发送 N 个帧：

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	LPUART 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	不产生
	1	1~N-1	每帧发送完成后置位	不产生
		N	每帧发送完成后置位	产生

表 20-3 LPUART DMA 中断说明

20.6 寄存器

LPUART0模块基地址：0x40014000

LPUART1模块基地址：0x40014400

offset 地址	名称	符号
LPUART0 寄存器		
0x00	LPUART0 控制状态寄存器（LPUART0 Control Status Register）	LPUART0_CSR
0x04	LPUART0 中断使能寄存器（LPUART0 Interrupt Enable Register）	LPUART0_IER
0x08	LPUART0 中断标志寄存器（LPUART0 Interrupt Status Register）	LPUART0_ISR
0x0C	LPUART0 波特率调制寄存器（ LPUART0 Baud rate Modulation Register）	LPUART0_BMR
0x10	LPUART0 接收缓冲寄存器（LPUART0 Receive Buffer Register）	LPUART0_RXBUF
0x14	LPUART0 发送缓冲寄存器（LPUART0 Transmit Buffer Register）	LPUART0_TXBUF
0x18	LPUART0 数据匹配寄存器（LPUART0 data Matching Register）	LPUART0_DMR
LPUART1 寄存器		
0x00	LPUART1 控制状态寄存器（LPUART1 Control Status Register）	LPUART1_CSR
0x04	LPUART1 中断使能寄存器（LPUART1 Interrupt Enable Register）	LPUART1_IER
0x08	LPUART1 中断标志寄存器（LPUART1 Interrupt Status Register）	LPUART1_ISR
0x0C	LPUART1 波特率调制寄存器（ LPUART1 Baud rate Modulation Register）	LPUART1_BMR
0x10	LPUART1 接收缓冲寄存器（LPUART1 Receive Data Register）	LPUART1_RXBUF
0x14	LPUART1 发送缓冲寄存器（LPUART1 Transmit Data Register）	LPUART1_TXBUF
0x18	LPUART1 数据匹配寄存器（LPUART1 data Matching Register）	LPUART1_DMR

20.6.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）

名称	LPUARTx _CSR(x=0,1)							
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	-				WKBYTE_CFG	-	RXEV	
位权限	U-0				R/W-0	U-0	R/W-00	

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	DMATXIFCFG	BITORD	STOPCFG
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	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
位权限	R/W-00		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**	LPUART 通信标志，只读 (Busy) 1：LPUART 正在通信中 0：LPUART 空闲
23:20	-	未实现：读为0
19	**WKBYTE_CFG**	数据接收唤醒条件配置 (Wakeup Byte Config) 1：接收完1字节，并且奇偶校验和STOP位都正确，才触发唤醒中断 0：接收完1字节，不检查校验位和STOP位，直接触发唤醒中断
18	-	未实现：读为0
17:16	**RXEV**	唤醒中断事件配置，用于控制何种事件下向 CPU 提供唤醒中断 (Receive Wakeup Event) 00：START 位检测唤醒 01：1byte 数据接收完成 10：接收数据匹配成功 11：RXD 下降沿检测
15:12	-	未实现：读为0
11	**IOSWAP**	RX 和 TX 引脚交换 (IO swapping) 0：默认引脚顺序（与封装图一致） 1：交换引脚顺序
10	**DMATXIFCFG**	DMA发送完成中断使能，仅在LPUART通过DMA进行发送时有效 (DMA Transmit Interrupt Config) 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 (Parity) 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)

20.6.2 LPUARTx 中断使能寄存器（LPUARTx _IER）

名称	LPUARTx _IER(x=0,1)							
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	-			RXEV_IE	-	RXERR_IE	-	RXBF_IE
位权限	U-0			R/W-0	U-0	R/W-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE_IE	TXSE_IE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:13	**-**	未实现：读为0
12	**RXEV_IE**	接收唤醒事件中断使能，1 有效 (Receive Event Interrupt Enable)
11	**-**	未实现：读为0
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)
7:2	**-**	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 Interrupt Enable)

20.6.3 LPUARTx 中断标志寄存器（LPUARTx _ISR）

名称	LPUARTx _ISR(x=0,1)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							RXEVF
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				TXOV	PERR	FERR	OERR
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TXBE	TXSE
位权限	U-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	**RXEVF**	接收唤醒事件中断标志，硬件置位，软件写 1 清零 (Receive Event Interrupt Flag,write 1 to clear) 中断标志触发源由 LPUxCR.RXEV 寄存器配置。
23:20	-	未实现：读为0
19	**TXOV**	发送缓存溢出错误，硬件置位，软件写1清零 (Transmit Overflow Error flag,write 1 to clear) 当发送缓存中的数据还未进入移位寄存器发送时，软件向发送缓存写入新数据，将触发TXOV标志置位。
18	**PERR**	奇偶校验错误中断标志，硬件置位，软件写 1 清零 (Parity Error flag,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 清零 (Receive Buffer Overflow Error flag,write 1 to clear)
15:9	-	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)
7:2	-	未实现：读为 0
1	**TXBE**	发送缓存空中断标志，硬件置位，写入 TXBUF 时清零 (Transmit Buffer Empty flag,write 1 to clear)
0	**TXSE**	发送缓存空且发送移位寄存器空中断标志，硬件置位，软件写 1 或者发送数据被载入移位寄存器时清零 (Transmit Shift register Empty flag, write 1 to clear)

20.6.4 LPUARTx 波特率调制寄存器（LPUARTx _BMR）

名称	LPUARTx _BMR(x=0,1)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	MCTL_EN	-			MCTL[11:8]			
位权限	R/W-1	U-0			R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MCTL[7:0]							
位权限	R/W-0000 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	-					BAUD		
位权限	U-0					R/W-000		

位号	助记符	功能描述
31	**MCTL_EN**	波特率调制使能 (modulation control enable) 0：关闭波特率调制 1：使能波特率调制
30:28	**-**	未实现：读为0
27:16	**MCTL**	LPUART 每个 bit 的位宽调制控制信号
15:3	**-**	未实现：读为0
2:0	**BAUD**	波特率控制（bps） 000：9600 001：4800 010：2400 011：1200 100：600 101/110/111：300

20.6.5 LPUARTx 接收数据寄存器（LPUARTx _RXBUF）

名称	LPUARTx _RXBUF(x=0,1)							
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**

位号	助记符	功能描述
8:0	**RXBUF**	接收数据缓存寄存器(Receive Buffer)

20.6.6 LPUARTx 发送数据寄存器（LPUARTx _TXBUF）

名称	LPUARTx _TXBUF(x=0,1)							
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							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	**-**	未实现：读为0
8:0	**TXBUF**	发送数据缓存寄存器(Transmit Buffer)

20.6.7 LPUARTx 数据匹配寄存器（LPUARTx _DMR）

名称	LPUARTx _DMR(x=0,1)							
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	-							MATD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MATD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:9	**-**	**未实现：读为0**
8:0	**MATD**	第一帧接收比较数据，如果 RXEV=10，当接收到的第一帧数据与 MATD 相同时，触发 RXEVF 中断，可以用于休眠模式下的数据接收唤醒。(Matched Data)

21 串行外设接口（SPI/I2S）

21.1 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
- 2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（4~32bits）
- 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）

21.2 SPI 功能描述

21.2.1 SPI 结构框图

下图为 SPI 模块的结构示意图。

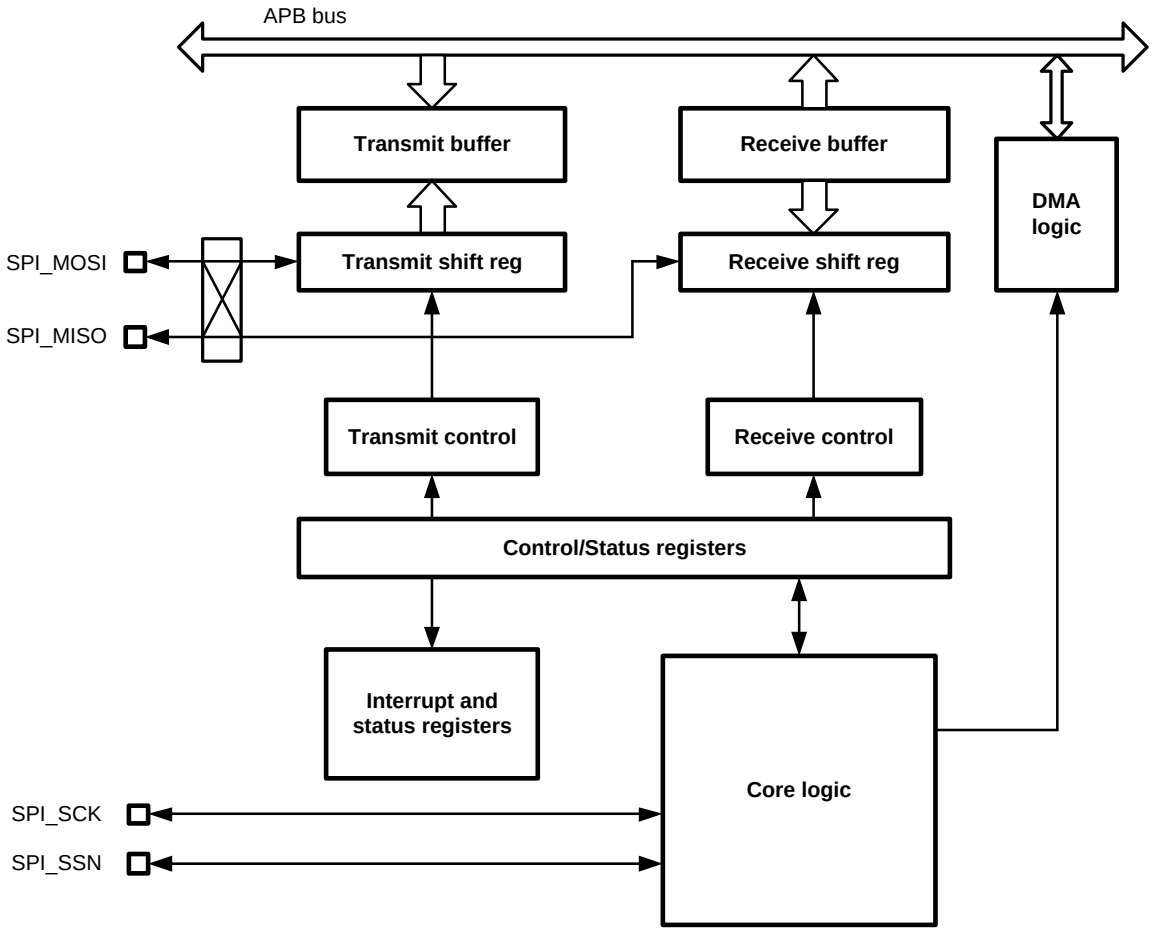


图 21-1 SPI 结构框图

21.2.2 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
PB8	PD2	SPI0	SSN	片选信号	SSN	片选信号

PB9	PD3		SCLK	时钟	SCLK	时钟
PB10	PD4		MISO	主机输入从机输出	DCN	命令/数据标识
PB11	PD5		MOSI	主机输出从机输入	SDATA	双向数据
PC7	PA11	SPI1	SSN	片选信号	SSN	片选信号
PC8	PA12		SCLK	时钟	SCLK	时钟
PC9	PB0		MISO	主机输入从机输出	DCN	命令/数据标识
PC10	PB1		MOSI	主机输出从机输入	SDATA	双向数据

表 21-1SPI 引脚列表

21.2.3 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（SPE）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

21.2.3.1 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。

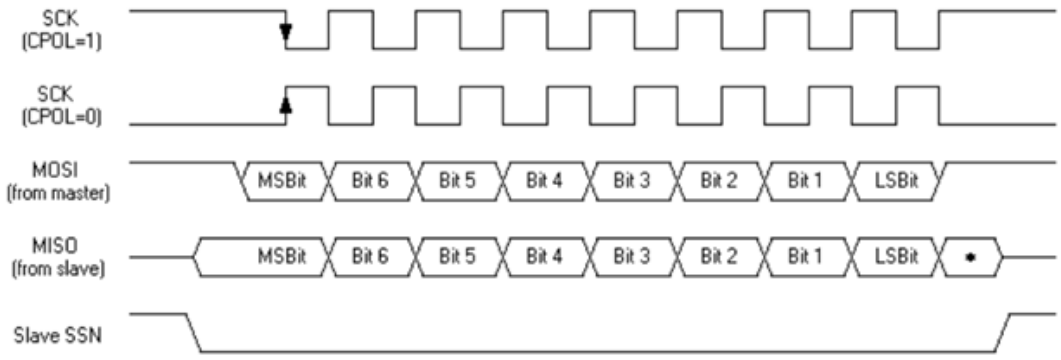


图 21-2 SPI 数据/时钟时序图（CPHA=0）

21.2.3.2 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。

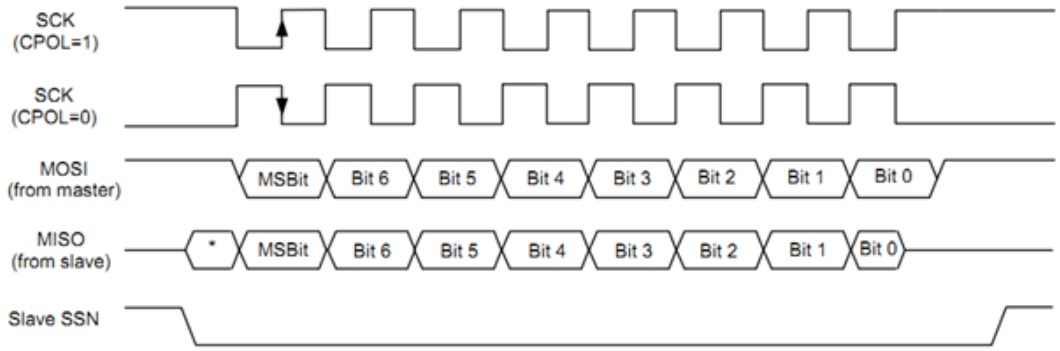


图 21-3 SPI 数据/时钟时序图（CPHA=1）

21.2.3.3 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。FM454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。

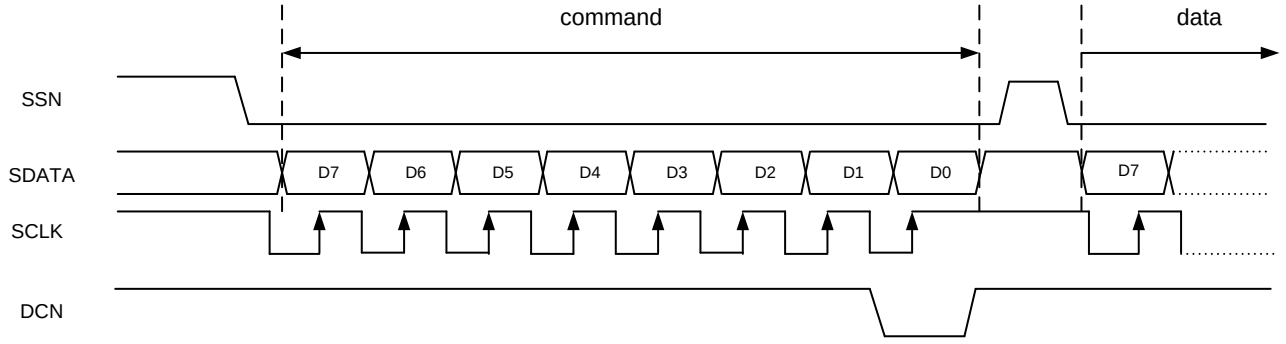


图 21-4 4 线半双工写操作

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。

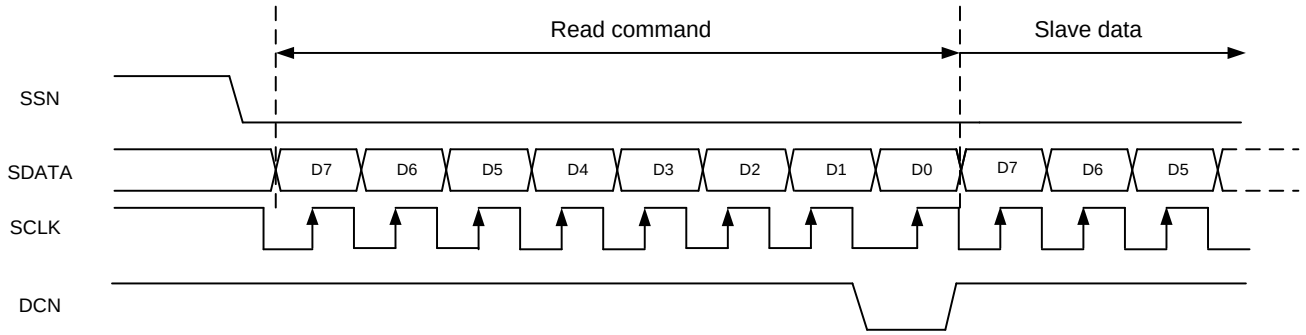


图 21-5 4 线半双工读操作（无 dummy cycle）

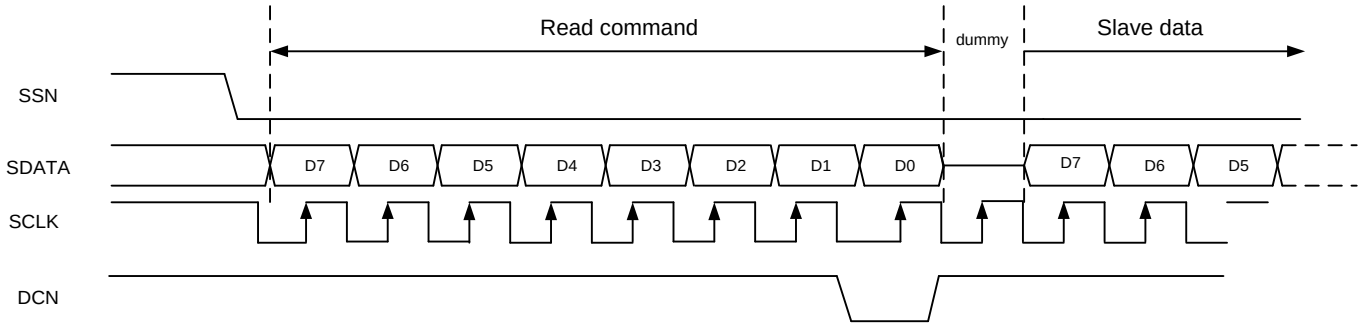


图 21-6 4 线半双工读操作（有 dummy cycle）

21.2.4 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，如图 21-2 所示，当 SPI 配置为主器件时，SSN 引脚必须接高电平，当 SPI 配置为从器件时，SSN 引脚必须接低电平。

SPI 主从器件的连接如下图所示：

主从器件的 MOSI、MISO 和 SCK 分别连在一起，主器件的 SSN 必须接高电平，从器件的 SSN 必须接低电平。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

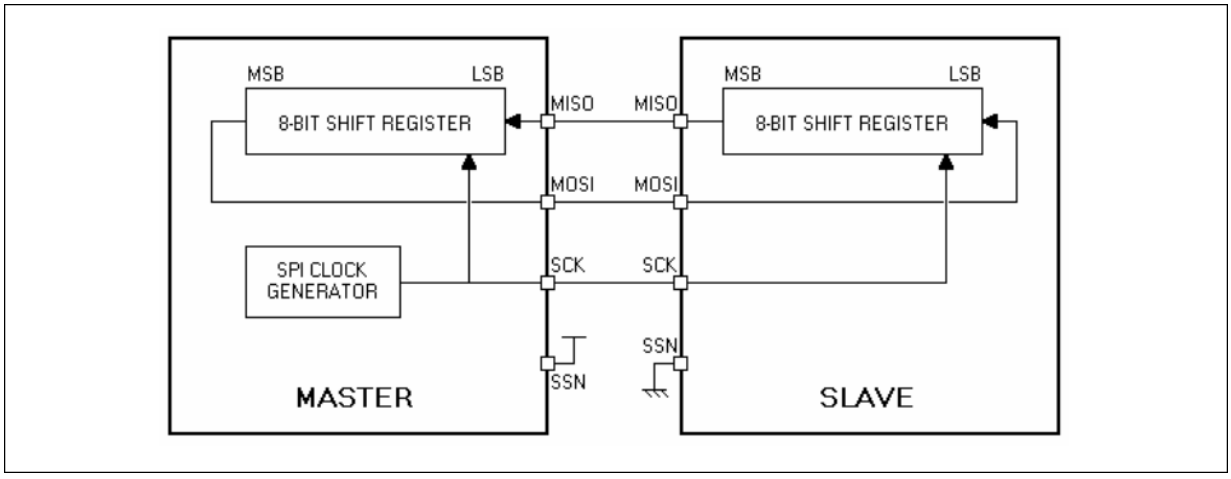


图 21-7 SPI Master/SPI Slave 互连

21.2.5 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21.2.11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，当新的数据从发送缓冲区转移至发送移位寄存器后，TXSE标志自动清零。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21.2.11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。

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。

21.2.6 主机发送间隔

通过SPIx_CR1.WAIT寄存器可以配置主机发送间隔，0~3个SCK时钟。当WAIT=0时，由于字节之间没有间隔，即使SSNM=1，SSN也不会拉高。

注意，当WAIT=0时，不能配置BAUD=f_{APBCLK}/2，否则在最后一个字节通信完成后，主机会多发一组SCK时钟。

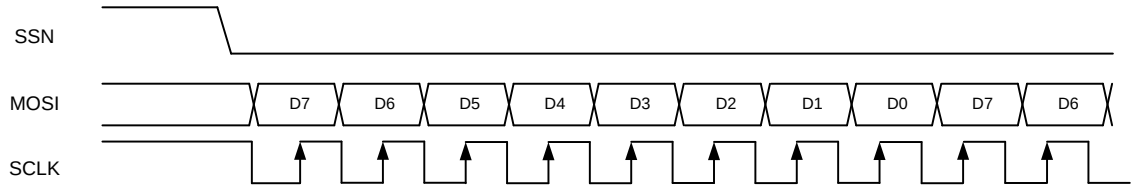


图 21-8 SPI Master 发送，WAIT=0

下图为WAIT=1的例子，SSNM=0，因此SSN不会拉高，但是SCK暂停一拍。

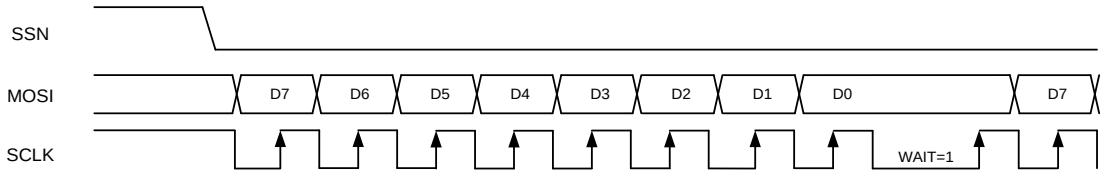


图 21-9 SPI Master 发送，WAIT=1

21.2.7 可编程数据字长

为了实现最大的协议灵活性，SPI支持4~32bit可编程数据字长，实际使用中常用的数据格式是8bit、16bit或32bit。

SPI发送缓冲和接收缓冲寄存器都是32bit的，当通信字长小于32bit时，数据在缓冲寄存器中以最低有效位保存；比如以8bit字长发送时，发送数据应写入TXBUF的bit7~0。数据缓冲寄存器不支持多帧缓冲，假设字长为8bit，不能一次性向TXBUF写入4帧数据，只能每次写入8bit，高于低8bit的数据将被丢弃。

21.2.8 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。

21.2.9 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。

21.2.10 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）；

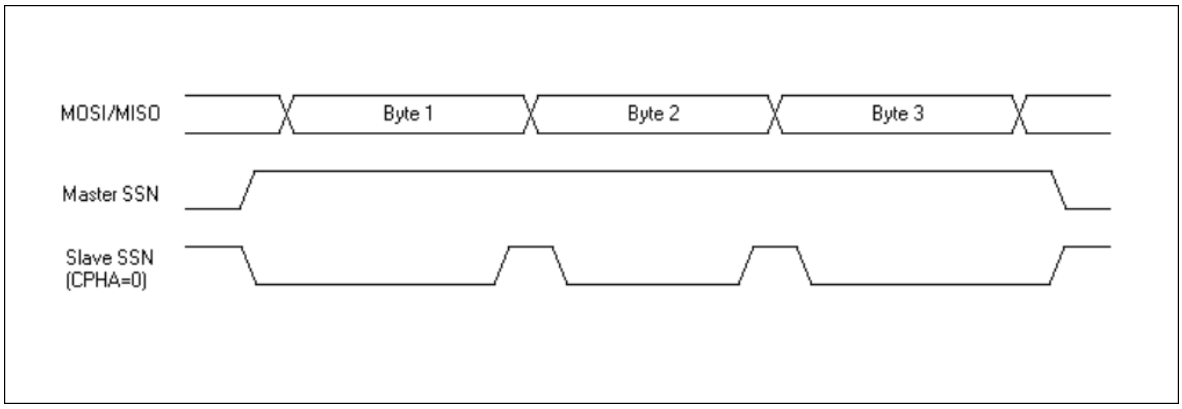


图 21-10 SPI SSN 时序图（SSNM=1，CPHA=0）

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。

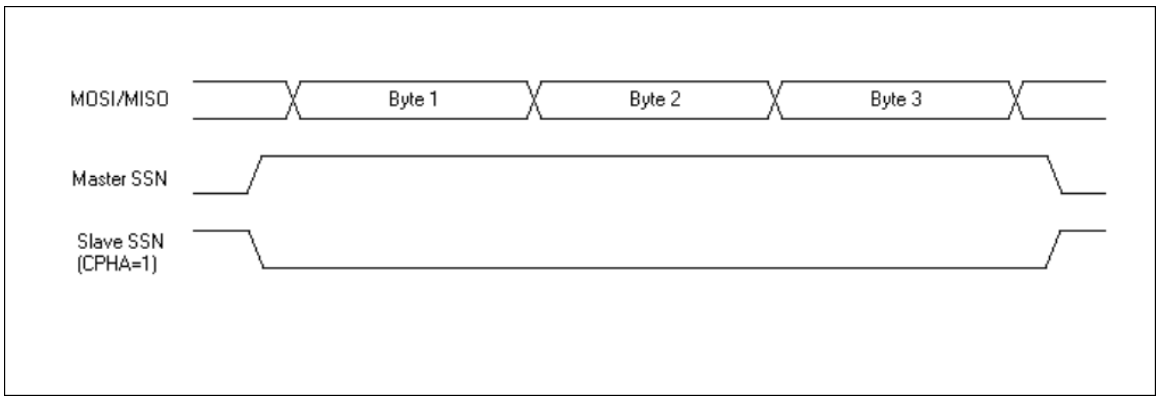


图 21-11 SPI SSN 时序图（SSNM=0）

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。

21.2.11 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。

21.2.12 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 }

21.3 I2S 功能描述

21.3.1 I2S 结构框图

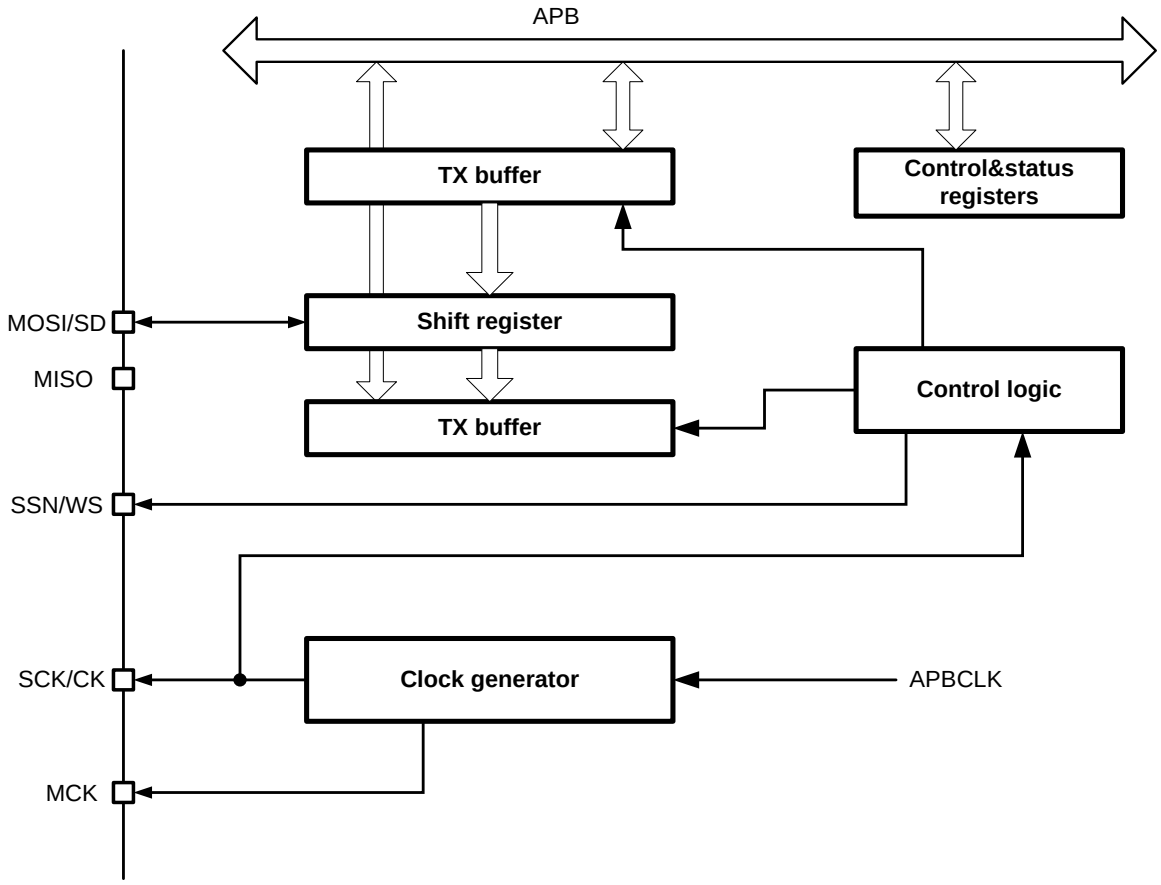


图 21-12I2S 结构框图

21.3.2 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

21.3.3 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。

21.3.3.1 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：

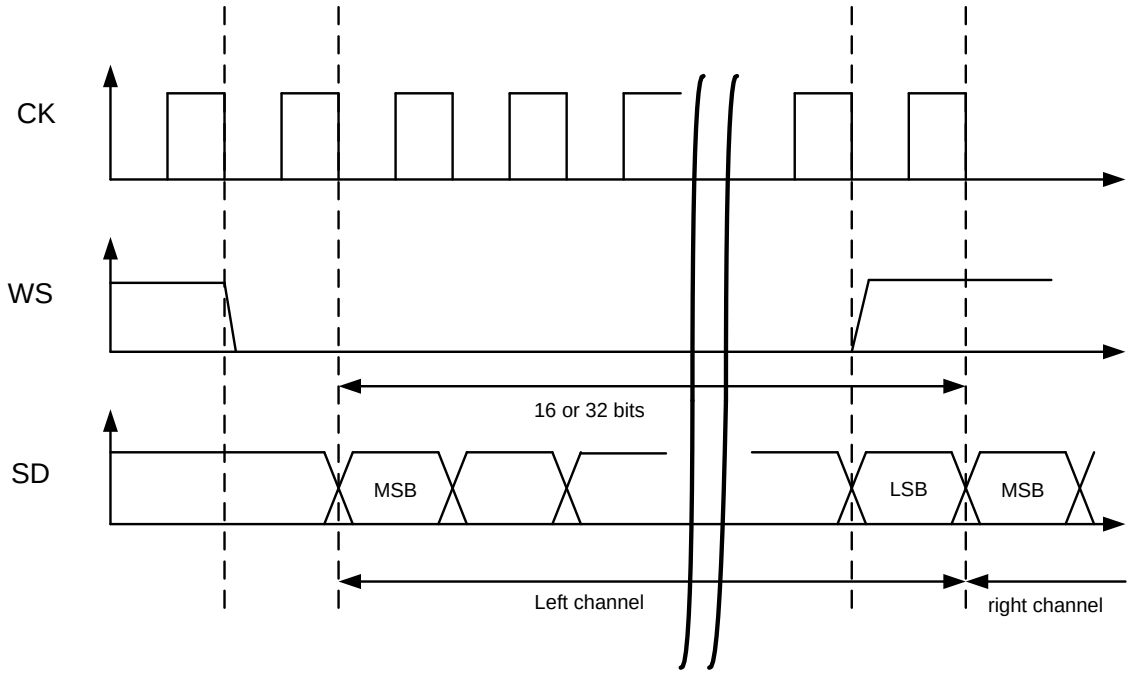


图 21-13Philips 标准时序

21.3.3.2 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。

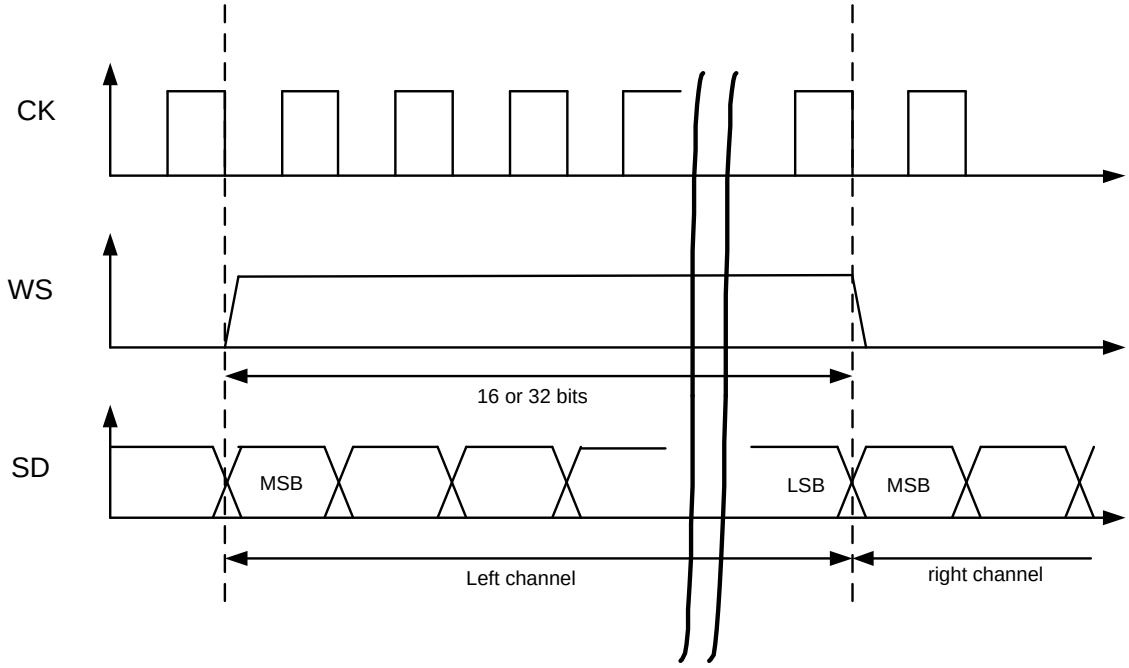


图 21-1432/16-bits MSB-justified 时序

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：

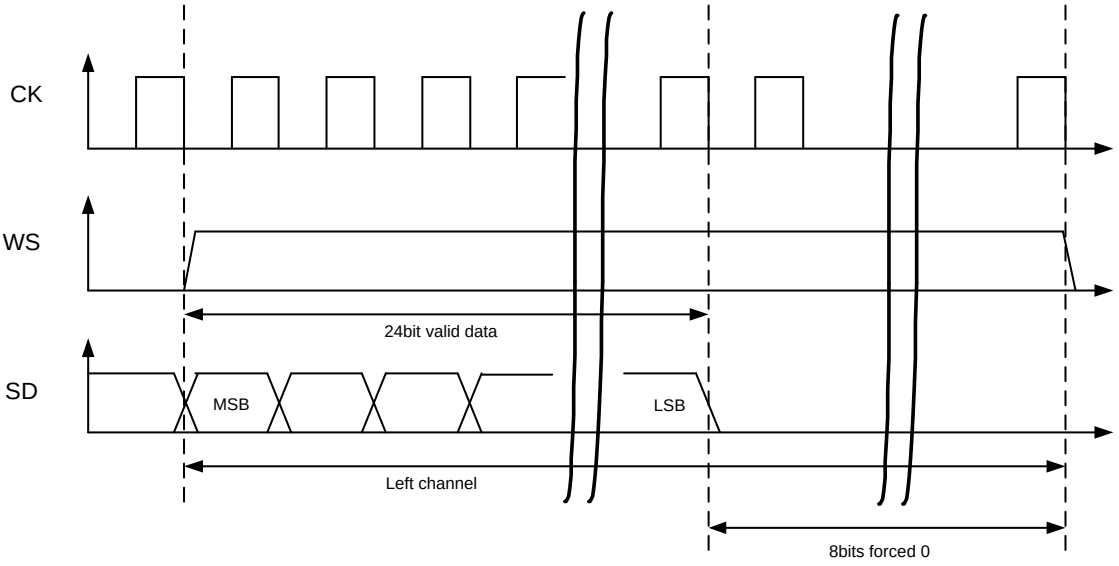


图 21-1524-bits MSB-justified 时序

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。

21.3.3.3 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：

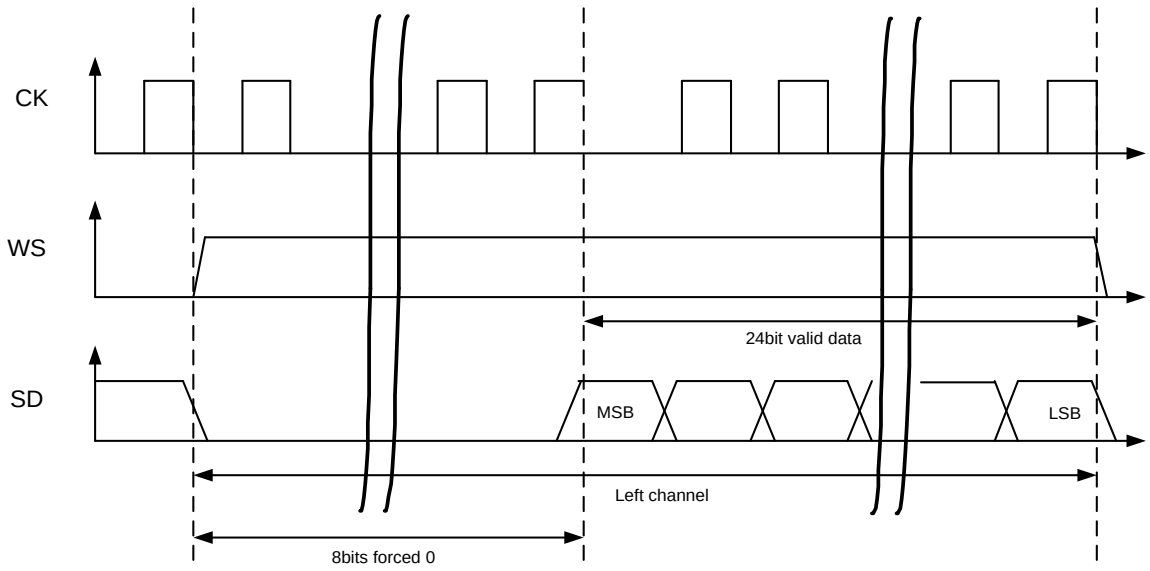


图 21-16LSB-justified 时序

21.3.3.4 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。

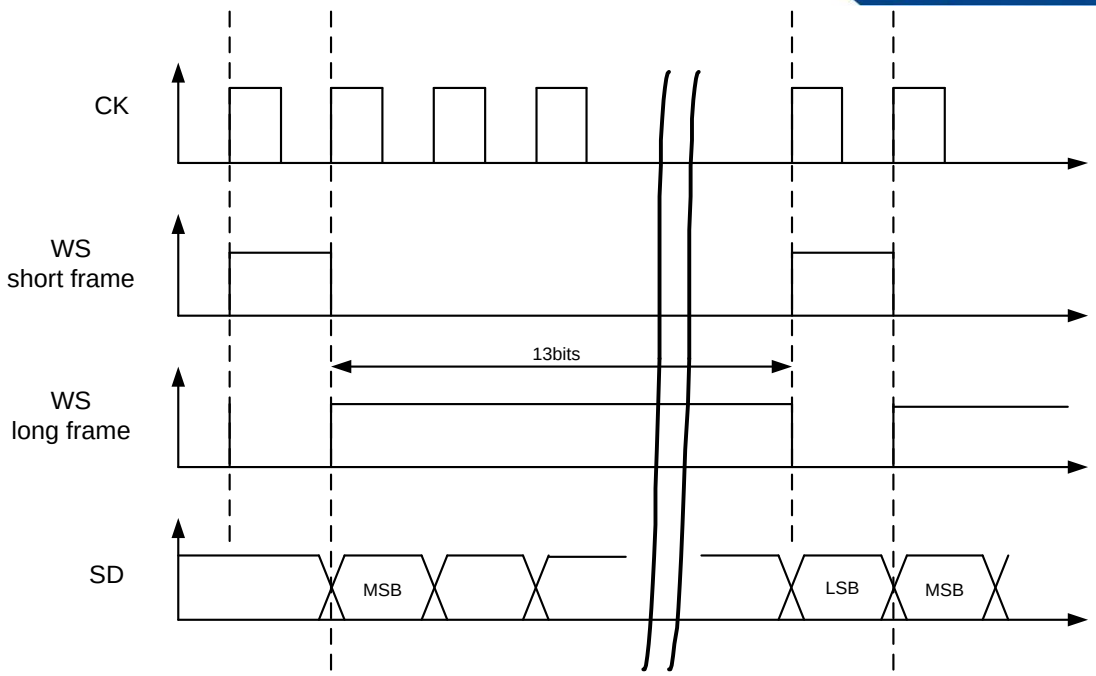


图 21-17PCM 标准时序

21.3.4 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×F_S

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 21-2I2S 典型采样频率计算表

21.3.5 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。

21.3.5.1 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 I2SCFG 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

21.3.5.2 发送流程

在 I2S 使能的情况下，软件或 DMA 写 TX BUFFER 将启动总线发送。TX Buffer 的宽度是 16bit，因此 24 或 32 位音频数据发送需要 2 次写操作。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 TX buffer 的数据会被立即载入移位寄存器进行发送，同时 TXE 中断标志寄存器置位，表示发送缓冲区空，软件可以在 TXE 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 buffer。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXE 置位并且 BUSY 清零。

21.3.5.3 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 RX buffer，同时置位 RXNE 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXNE 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 RX buffer 中的数据。如果新的数据到来时 RX buffer 非空，则 OVR 错误标志寄存器将被置位。

21.3.6 I2S 从机功能

21.3.6.1 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

21.3.6.2 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 TX buffer。如果主机发送时钟和 WS 时，TX buffer 和移位寄存器都空，则 I2S 模块将置位 UDR 错误标志，表示数据下溢出。

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，TX buffer 中的数据被载入移位寄存器开始发送，此时 TXE 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 TX buffer。

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 buffer。

软件在清零 I2SEN 寄存器关闭 I2S 模块前，必须等待 TXE 置位并且 BUSY 清零。

21.3.6.3 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。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 RX buffer，同时置位 RXNE 中断标志寄存器，表示接收缓冲区非空。软件可以在 RXNE 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 RX buffer 中的数据。如果新的数据到来时 RX buffer 非空，则 OVR 错误标志寄存器将被置位。

21.3.7 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。

21.3.8 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，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。

21.4 寄存器

SPI0模块基地址：0x40010400

SPI1模块基地址：0x40010800

offset 地址	名称	符号
SPIx 寄存器 (x=0,1)		
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

21.4.1 SPI 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1 (x=0,1)							
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	CPOL	CPHA
位权限	R/W-00		R/W-000			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) 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

Bit	助记符	功能描述
		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 (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。 当 WAIT=0 时，SPI 主机连续发送，字节之间没有间隔 当 WAIT=0 时，BAUD 不能配置为 $f_{APBCLK}/2$
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。

21.4.2 SPI 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2 (x=0,1)							
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	DLEN2			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	R/W-000			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	**DLEN2**	通信数据字长扩展配置寄存器，与 DLEN 组合起来形成最终配置（配置方式考虑既有产品的兼容性） {DLEN2, DLEN} 00000: 8bit 00001: 16bit 00010: 24bit 00011: 32bit 00100: 7bit 00101: 15bit 00110: 23bit 00111: 31bit 01000: 6bit 01001: 14bit 01010: 22bit 01011: 30bit 01100: 5bit 01101: 13bit 01110: 21bit 01111: 29bit 10000: 4bit 10001: 12bit 10010: 20bit 10011: 28bit 10100: RFU 10101: 11bit 10110: 19bit 10111: 27bit 11000: RFU 11001: 10bit 11010: 18bit 11011: 26bit 11100: RFU 11101: 9bit 11110: 17bit 11111: 25bit
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)

Bit	助记符	功能描述
		参见 DLEN2
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) 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

21.4.3 SPI 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3 (x=0,1)							
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	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	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 HSPISTA.MERR 寄存器
0	SERRC	Slave Error Clear，软件写 1 清除 HSPISTA.SERR 寄存器

21.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）

名称	SPIx_IER (x=0,1)							
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)

21.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR (x=0,1)							
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/Dy-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/W-0	R/W-0	R/W-0	R/W-0	R-1	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，软件写 1 清零 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，软件写 1 清零 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

Bit	助记符	功能描述
		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

21.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF (x=0,1)							
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
31:0	**TXBUF**	SPI 发送缓存(Transmit Buffer)

21.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF (x=0,1)							
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 接收缓存(Receive Buffer)

21.4.8 I2S 配置寄存器（SPIx_I2SCR）

名称	SPIx_I2SCR (x=0,1)							
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 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

Bit	助记符	功能描述
		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

21.4.9 I2S 预分频寄存器（SPIx_I2SPR）

名称	SPIx_I2SPR (x=0,1)							
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 0010							

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)

22 单边沿半字节传输模块（SENT）

22.1 概述

SENT（Single Edge Nibble Transmission）模块是基于SAE J2716协议的单向、单线串行通信接口，本接口主要用于车载传感器与ECU之间的数据通信。本芯片包含2个独立SENT模块。

SENT模块主要特性如下：

- 符合SAE J2716协议
- 可配置为发送或接收模式
- 自动数据波特率同步
- CRC计算和CRC检测功能
- 1个消息帧的数据缓存
- 可选数据长度
- 可选的暂停脉冲
- 帧格式错误检测

SENT通信时序基于预定义的时间单元tick，接收方和发送方都要在通信开始前预先配置相同的tick周期，支持的tick周期范围是3us到90us。

一个SENT消息帧从同步脉冲开始，同步脉冲用来让接收方计算发送方的数据波特率，SENT协议允许tick周期出现20%的变化，从而降低对通信双方的时钟精度要求。

22.2 消息格式

22.2.1 Data Nibble

SENT数据传输基于nibble形式，每个nibble总是用来编码4bit数据，编码形式是nibble的长度。每个nibble总是以4 tick长度的低电平开始，nibble最小长度为12 tick，代表数据0000，最大长度为27 tick，代表数据1111：

- Data = 4’b0000: nibble is 12 ticks
- Data = 4’b0001: nibble is 12+1=13 ticks
- ...
- Data = 4’b1111: nibble is 12+15=27 ticks

Nibble波形示意图如下：

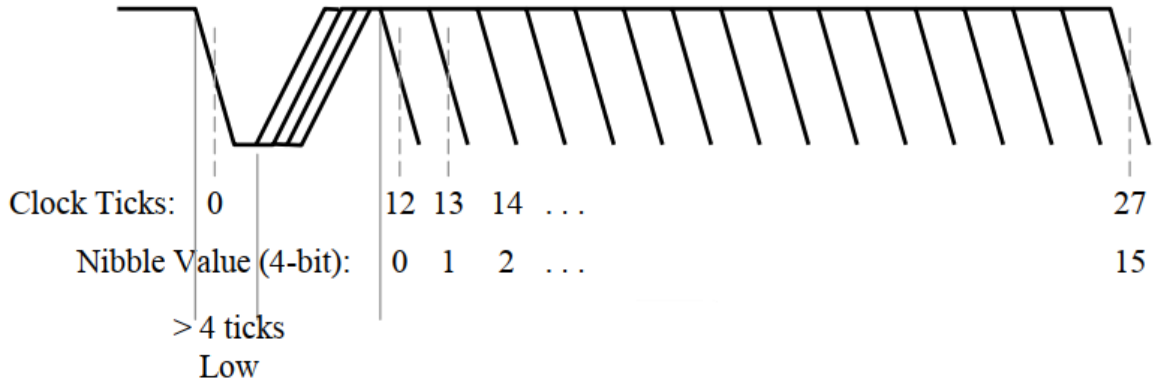


图 22-1nibble 波形示意图

发送数据0和发送数据7时的nibble波形如下所示：

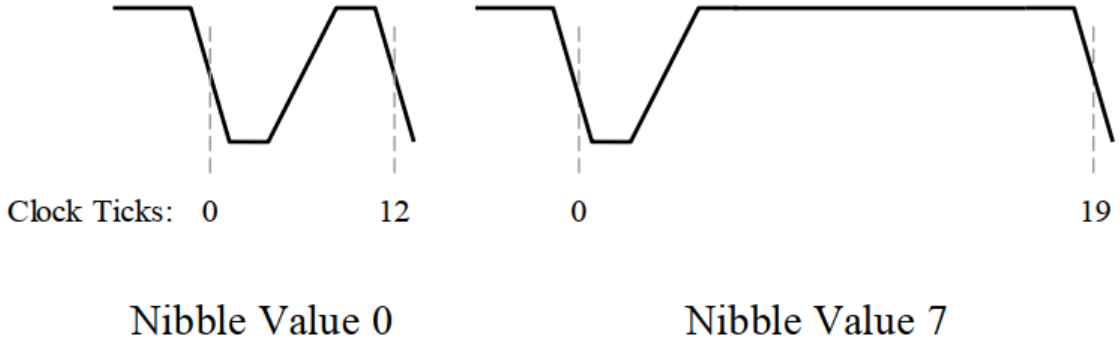


图 22-2 数据 0 和数据 7 的 nibble 波形

22.2.2 同步脉冲（Sync Pulse）

同步脉冲用于接收方波特率计算，名义长度是56 ticks，以4 ticks低电平为起始。

通信双方虽然事先约定了tick长度，但是考虑到时钟偏差，实际tick长度可能与理想值不同，通过检测sync pulse的实际长度，接收方可以计算实际的tick长度，并校准采样时钟。

22.2.3 Status and communication nibble

这个nibble在同步脉冲之后、数据nibble之前发送，用于传输特定辅助信息，其中最低2bit为保留位。

注意status and communication nibble不包含在帧CRC计算中。

Bit2和bit3可作为可选的消息通道使用，支持短消息格式（short serial message format）或加强消息格式（enhanced serial message format）。

注意：使用status and communication nibble用作option channel时，收发的数据和识别pattern注入、CRC计算都是由软件完成的。

22.2.3.1 Short serial message format

短消息格式中，使用status and communication nibble的bit 2来传输额外的16bit串行数据，由于每个帧用来传输数据的仅1bit，所以需要连续16帧完成短消息传输，并且连续16帧必须满足无数据格式

错误和CRC错误的条件。

作为短消息格式的识别，第一帧的status and communication nibble的bit 3为1，而后续15帧bit 3连续为0；16bit短消息中包含4bit消息ID、8bit数据字节、4bit CRC校验和。CRC校验包含消息ID和数据字节，校验算法与data nibble相同。

短消息发送格式如下图所示：

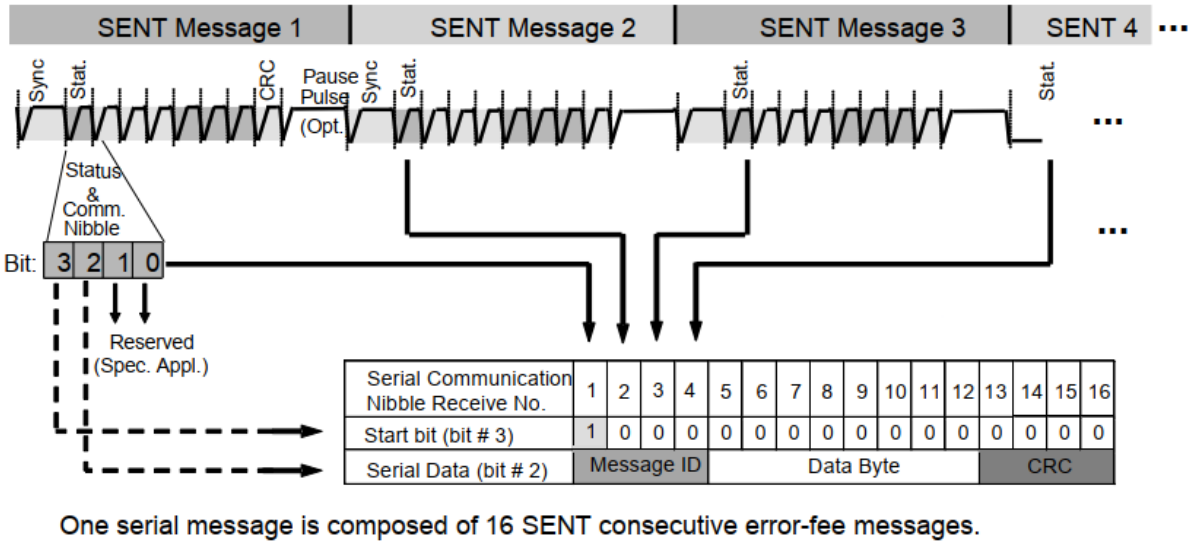


图 22-3 短消息格式 option channel

22.2.3.2 Enhanced serial message format

另一种option channel格式为加强消息格式。加强消息格式中，需要连续18个帧，并且同时使用status and communication nibble的bit 2和bit 3来做通信。连续18帧必须同时满足无数据格式错误和CRC错误的条件。

加强消息帧通过特殊的bit3 pattern来识别：“01111110”，这个pattern在消息帧的#18、#1~#7中分别发送。

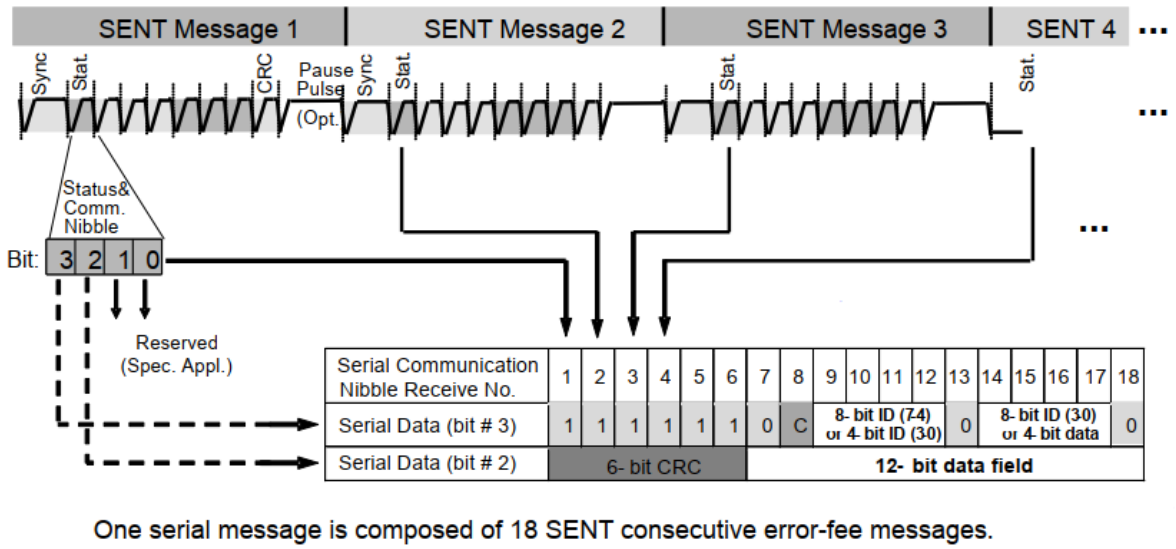


图 22-4 加强消息格式 option channel

使用加强消息格式时，option channel中传输数据的CRC算法与data nibble不同，具体请参见SAE J2716协议。

22.2.4 消息帧格式

一个完整的消息帧包含了同步脉冲，状态和通信nibble，数据nibble，CRC nibble，可选的暂停脉冲，如下图所示：

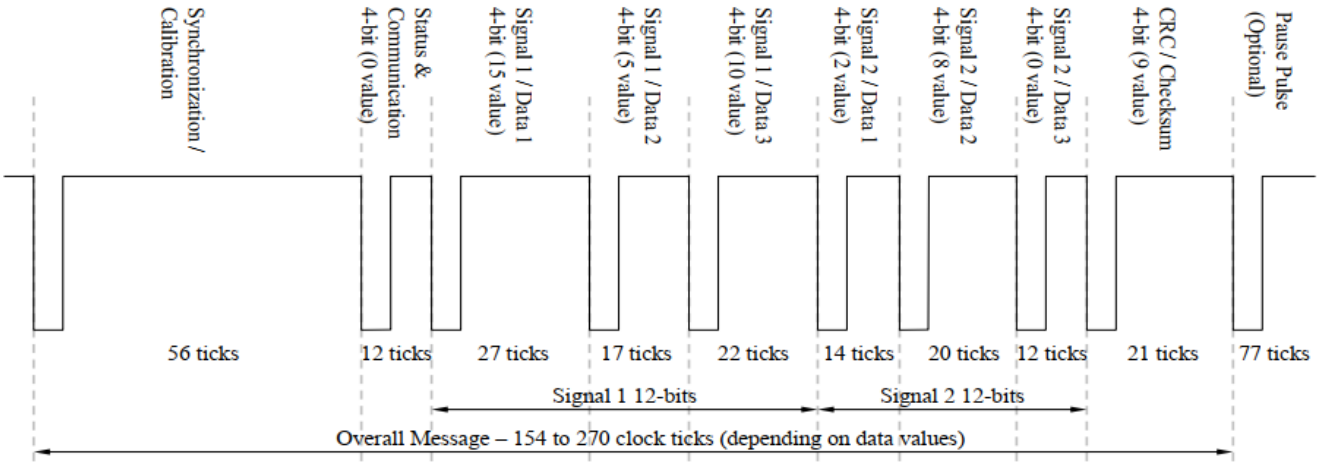


图 22-5 消息帧格式

消息帧中数据nibble个数是由应用决定的，不同应用使用预先定义的、固定的的nibble个数。比如传感器需要发送24bit数据给ECU，则nibble个数为6。本模块支持1~6 data nibbles。

22.2.5 Optional Pause Pulse

Pause pulse用于填充消息帧，使得所有的消息帧都可以有相同的长度（tick个数）。FRAMETIME寄存器用于定义整个消息帧的长度，虽然这是一个12bit寄存器，SAE J2716协议规定了Pause脉冲最长不超过768 ticks，因此整个消息帧长度不会超过1038 ticks（270+768），配置FRAMETIME大于1038是没有意义的。

$$FRAMETIME = T_{FRAME} / T_{tick}$$

22.2.6 CRC 校验

SENT通信仅针对data nibble使用4bit CRC，status&communication nibble不参与CRC计算。

CRC多项式：x^4+x^3+x^2+1，初始化种子0101

注意：新版标准建议发送时计算CRC需要在数据后加上一个0000的输入增强：参见SAE J2716 Appendix B

Table B.1-3 – Example 4-bit CRC calculation by polynomial division

		seed	nibble1	nibble2	nibble3	nibble4	nibble5	nibble6	augmentation
Seed/Aug	5/0	5							0
data (hex)	4AC4AC		4	A	C	4	A	C	
data (bin)		0 1 0 1	0 1 0 0	1 0 1 0	1 1 0 0	0 1 0 0	1 0 1 0	1 1 0 0	0 0 0 0

为了与legacy标准保持兼容，可以通过CRCM寄存器配置是否在发送模式计算CRC时加入0000增强。

22.3 结构框图

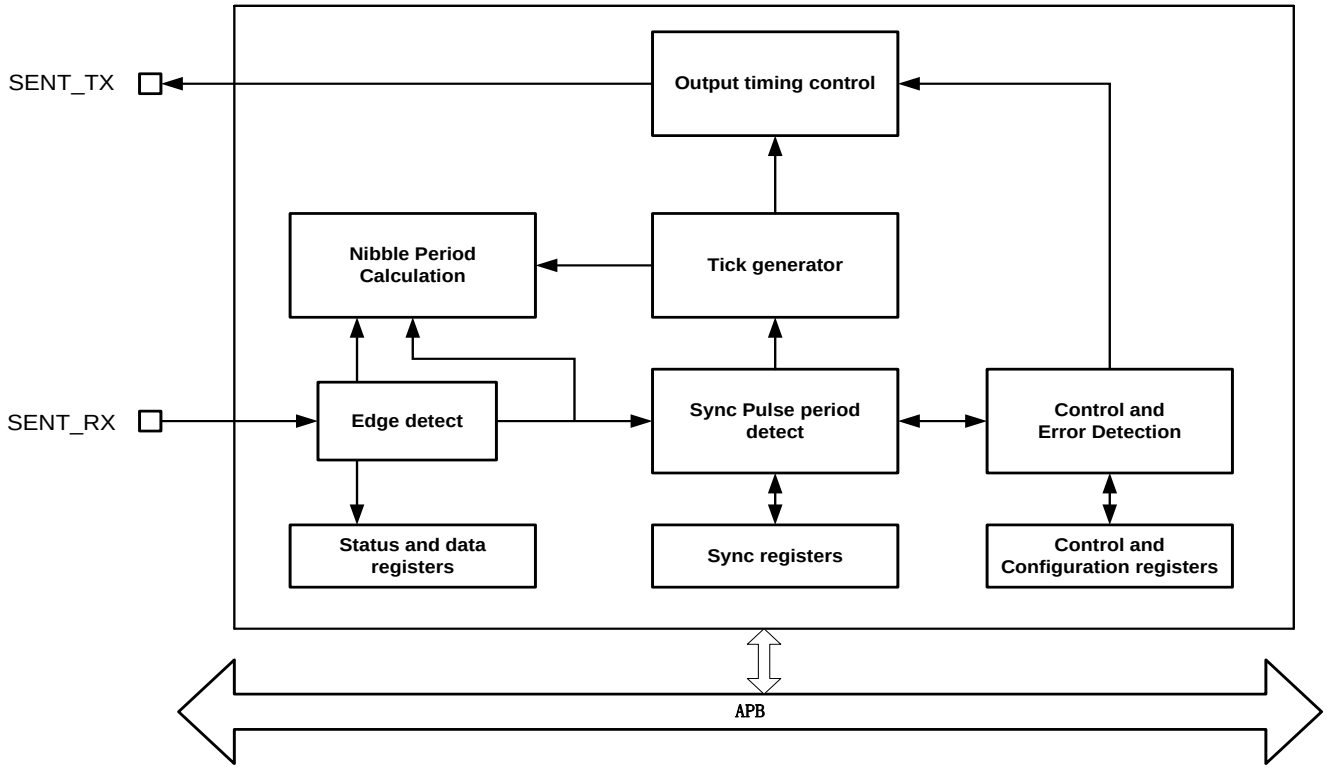


图 22-6SENT 模块结构框图

22.4 功能描述

22.4.1 发送 Tick 时间

SENT模块使用APBCLK工作。由于SENT协议规定的tick长度是3~90us，即tick时钟频率11.1KHz~333KHz，根据Prescaler寄存器设置，对APBCLK分频后得到工作时钟F_{SENT}，这个时钟固定为Tick频率的8倍。

$$F_{SENT} = F_{APBCLK} / (prescaler + 1)$$

$$T_{tick} = \frac{8}{F_{SENT}}$$

模块工作前，需要在CMU和RMU模块中配置相关寄存器，使能模块工作时钟并释放模块复位。

22.4.2 发送模式

SENT模块配置为发送模式时，可以支持连续发送或软件触发发送模式。

- 连续发送：软件使能发送后，SENT模块连续不断发送寄存器中的数据，软件应负责及时填充新的数据；如果数据更新不及时，SENT模块将重复发送已经发送过的数据
- 软件触发：SENT模块完成一帧发送后，会等待软件重新使能发送

发送初始化流程：

1. 清零MOD，进入发送模式
2. 配置TXM选择连续发送或软件触发模式
3. 配置Tick长度
4. 配置数据nibble个数
5. 配置CRCEN和CRCM，选择硬件计算CRC或软件计算CRC，以及是否进行CRC增强
6. 配置PPEN选择是否要发送pause pulse
7. 如果PPEN=1，向FRAMETIME寄存器写入帧长度
8. 向数据寄存器写入待发送的数据和Status&Communication nibble
9. 根据需要使能中断
10. 如果CRCEN=0，软件计算CRC并写入CRC寄存器
11. 如果CRCEN=1，软件通过置位CCRC寄存器触发硬件计算CRC
12. 置位发送使能，开始发送（CCRC和TXEN可以同时置位）

注意：在连续发送的模式下，每个data nibble发完都可以及时更新data寄存器，所有datanibble更新完毕后，通过置位CCRC更新CRCnibble；以上操作必须在下一个sync发送前完成。

注意：参与CRC计算的data nibble个数由NIBCNT指定。

22.4.3 接收 Tick 时间

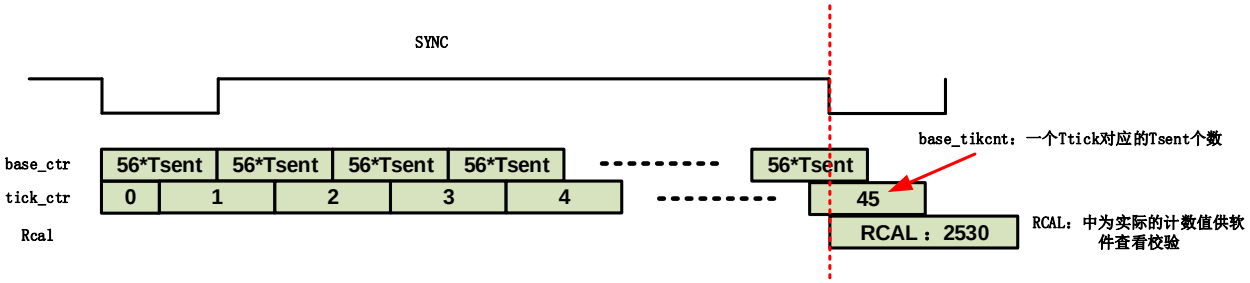
接收模式下，SENT模块自动计算sync pulse的宽度，采样到的sync pulse宽度处于SYNCMIN和SYNCMAX寄存器定义的范围之内时，模块开始正常数据接收。Sync脉冲宽度被自动写入RCAL寄存器中，用作后续数据接收时的校准参数。

在接收SYNC阶段，会在sent_rx的下降沿以 $F_{SENT} = F_{APBCLK} / (prescaler + 1)$ 的时钟频率计数按照每56计数加1一直到下一个下降沿，期间计算多少个56*T_{SENT}即可以得到当前的T_{TICK}，此处会进行round操作，每一轮计数过半才能加1，需要注意的是，为了减小误差到小于1/64，需要根据使用的T_{tick}调节pres分频选取合适的F_{SENT}操作，让tick_ctr的值大于32，避免短的T_{tick}使用较低频率。

所以推荐满足 $32<T_{tick}*F_{SENT}<64$,如 T_{tick} =12.5us ，F_{sent}=2.667M（8M的3分频）

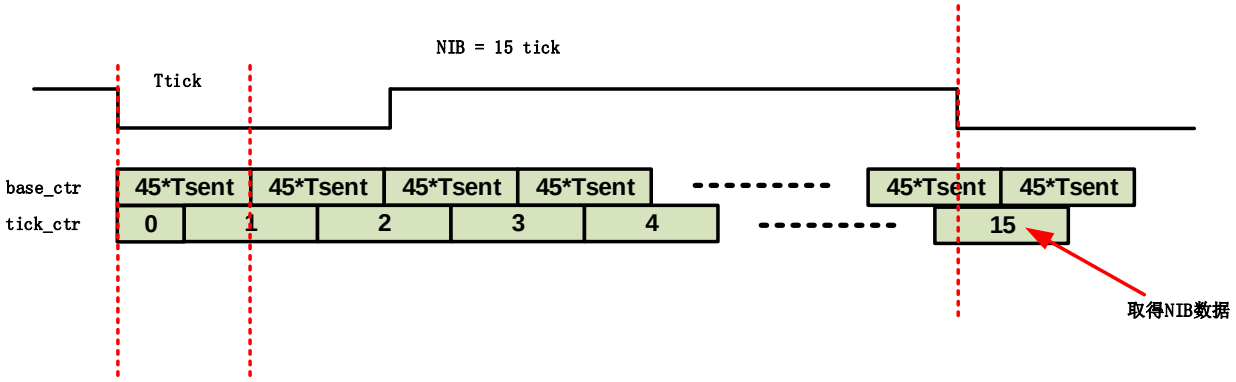
由于计数的tick使用整数，所以存在四舍五入的问题，如果Ttick*FSENT的小数部分正好在0.5附近，则会造成最大误差，极端情况也可能出错，所以建议Ttick*FSENT尽量接近整数，小数部分尽量远离0.5。

例如下图 sync中得到的base_tickcnt=45，则表示1*T_{TICK} =45*T_{SENT}



在接收数据阶段，base_ctr则按照SYNC过程中得到的对应于Tick的计数值进行计数，得到最终的NIB长度。

如下图



SYNCMIN和SYNCMAX用于接收诊断，因为协议允许+/-20%的误差容限，而当误差超过+/-20%时，应产生错误事件；

计算这两个值需要根据通信Ttick的时间，当前Fsent的频率，计算理论SYNC的周期数，SYNC = Ttick*56*Fsent

举例：Ttick = 6us，Fsent=8M，SYNC = 6*8*56 = 2688

SYNCMIN = 0.8* SYNC=2150

SYNCMAX = 1.2* SYNC=3225

超出这个范围的将会报错并继续接收新的SYNC过程。

22.4.4 接收模式

接收初始化流程：

1. 置位MOD，进入接收模式
2. 配置Tick长度
3. 配置数据nibble个数
4. 配置CRCEN，选择硬件计算CRC或软件计算CRC
5. 配置PPEN选择是否要接收pause pulse
6. 配置SYNCMAX和SYNCMIN
7. 根据需要使能中断
8. 置位接收使能，开始接收

SENT模块在接收data nibble过程中CRC模块自动计算校验和，并将结果与接收到的CRC nibble进行比较，如果校验和错误将置位CRCERR标志，并可以触发中断。

22.4.5 接收消息诊断

接收模式下，当以下情况发生时，认为遇到了错误，模块产生错误标志和中断。

- Sync pulse大于SYNCMAX或小于SYNCMIN
- 两个sync pulse之间的下降沿个数与预期不符（消息帧长度是预定义的，比如data nibble个数为6，PPEN=1，则两个syncpulse之间应出现9个下降沿，否则就是格式错误）
- CRC错误
- Data nibble计算结果<0（小于12ticks）或>15（大于27ticks）
- 连续两个sync pulse测量值误差超过了+/-1.5625%（1/64）

注意：即使检测到错误事件（CRC错误，和连续2个syncpulse测量误差超过了1/64），SENT模块仍会将解码的数据写入寄存器，软件应自行判断如何处理数据。（其他错误会导致数据无法写入，无法解码）

22.5 寄存器

SENT0模块起始地址：0x4001_A000

SENT1模块起始地址：0x4001_A800

offset 地址	名称	符号
0x00	SENT 控制寄存器	SENT_CR
0x04	SENT 配置寄存器	SENT_CFGR
0x08	SENT 预分频寄存器	SENT_PSR
0x0C	SENT 帧长度寄存器	SENT_FTR
0x10	SENT 状态和标志寄存器	SENT_SR
0x14	SENT 数据寄存器	SENT_DR
0x18	SENT 同步脉冲诊断寄存器	SENT_SYNC
0x1C	SENT 同步校准寄存器	SENT_RCAL

22.5.1 SENT 控制寄存器（SENT_CR）

名称	SENT_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	-					-	RXEN	TXEN
位权限	U-0					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	RXEN	SENT 接收使能，仅当 MOD=0 时可以置位 1：开始接收数据 0：关闭接收功能
0	TXEN	SENT 发送使能，软件置位启动发送，仅当 MOD=1 时可以置位 当 TXM=1，即连续发送模式时，TXEN 置位后保持为 1，直到软件清零，SENT 模块会不断发送数据，软件应保证及时更新数据寄存器 当 TXM=0，即软件触发模式时，TXEN 在当前帧发送完成后自动清零，SENT 模块发送完一帧后停止，等待下次软件置位 TXEN

22.5.2 SENT 配置寄存器（SENT_CFGR）

名称	SENT_CFGR							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DFLEN							
位权限	R/W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		FTXIE	NTXIE	SYNCIE	ERRIE	FRXIE	NRXIE
位权限	U-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	-					NIBCNT		
位权限	U-0					R/W-110		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CRCM	PPEN	CRCEN	TXPOL	TXM	MOD
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
31:24	DFLEN	RX 输入数字滤波，表示 APBCLK 采样滤波的长度，连续一致的电平才会被确定为输入电平 0x00:关闭数字滤波 0x01:采样一个 APBCLK 周期 … 0xFF:采样 255 个 APBCLK 周期
31:22	--	RFU：未实现，读为 0
21	FTXIE	Frame 发送完成中断使能
20	NTXIE	Nibble 发送完成中断使能
19	SYNCIE	Sync pulse 中断使能 1：syncpulse 正确接收（宽度在 SYNCMIN 和 SYNCMAX 之间）后产生中断 0：syncpulse 不产生中断
18	ERRIE	错误事件中断使能 1：错误事件发生时产生中断 0：不产生错误中断（标志寄存器仍会置位）
17	FRXIE	Frame 接收完成中断使能
16	NRXIE	Nibble 接收完成中断使能
10:8	NIBCNT	配置帧中 Data Nibbles 个数，发送和接收有效 000: forbidden 001: 1 nibble 010: 2 nibbles … 110: 6 nibbles 111: forbidden
7:6	--	RFU：未实现，读为 0
5	CRCM	CRC 模式选择，仅发送模式有效 0:recommended mode, augmented with 0000 1: legacy mode, no augment
4	PPEN	暂停脉冲（pause pulse）使能 1：帧发送完成后发送 PP，PP 结束后才能启动下一帧发送 0：帧发送完成后不发送 PP，直接启动下一帧发送

Bit	助记符	功能描述
3	CRCEN	硬件 CRC 使能 1：使能硬件 CRC 计算 0：禁止硬件 CRC 计算
2	TXPOL	发送空闲状态下的输出电平 0：发送空闲时 SENT_TX 输出高电平 1：发送空闲时 SENT_TX 输出低电平
1	TXM	发送模式选择 0：软件触发模式 1：连续发送模式
0	MOD	SENT 工作模式 0：接收模式 1：发送模式

22.5.3 SENT 预分频寄存器（SENT_PSR）

名称	SENT_PSR							
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	PRESCALER							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRESCALER							
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	PRESCALER	SENT 时钟预分频，根据这个寄存器的值对 APBCLK 进行预分频得到 SENT 工作时钟频率 实际分频值是 PRESCALER+1 $F_{SENT}=F_{APBCLK}/(PRESCALER+1)$

22.5.4 SENT 帧长度寄存器（SENT_FTR）

名称	SENT_FTR							
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					FRAMETIME			
位权限					R/W-000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FRAMETIME							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:0	FRAMETIME	发送帧长度，单位是 tick，包含 sync、status、data、CRC 和 pause 在内的总长度；仅发送模式且 PPEN=1 时有效，据此可以计算出 pause 脉冲的长度

22.5.5 SENT 状态和标志寄存器（SENT_SR）

名称	SENT_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	-		NIB			RXIDLE	PAUSE	SYNC
位权限	U-0		R-000			R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		TCOERR	SSERR	NBERR	CRCERR	NCNTERR	SYNCERR
位权限	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	-		FTXF	NTXF	SYNCF	-	FRXF	NRXF
位权限	U-0		R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:19	NIB	Nibble 收发状态指示 000：正在收发 status nibble 001：正在收发 1st data nibble 010：正在收发 2nd data nibble … 110：正在收发 6nddatanibble 111：正在收发 CRCnibble
18	RXIDLE	接收空闲状态 1：接收到高电平的时间已经超过了 SYNCMAX 0：采样到低电平，处于接收过程中
17	PAUSE	Pause 状态指示

Bit	助记符	功能描述
		1：正在接收/发送 pause 0：未在接收/发送 pause
16	SYNC	Sync 状态指示 1：正在接收/发送 sync 0：未在接收/发送 sync
15:14	--	RFU：未实现，读为 0
13	TCOERR	Tick counter overflow error tick 计数器溢出错误，此错误可能会在接收 SYNC 的时候出现，如果模块时钟和 pres 配置与 Tick 的长度不匹配，比如过长的 Tick 和过慢的时钟，则会出现溢出错误，推荐计算参考 1.6.3 接收 Tick 时间章节，软件写 1 清零 仅配置为接收工作时有效
12	SSERR	连续 Sync（Successive Sync）长度误差超范围，硬件置位，软件写 1 清零 当连续两个 sync 长度测量值误差超过 1/64 时置位
11	NBERR	Nibble 测量错误（<12ticks 或>27ticks）标志，硬件置位，软件写 1 清零
10	CRCERR	CRC 校验错误标志，硬件置位，软件写 1 清零
9	NCNTERR	两个 sync 脉冲之间的 nibble 个数与设定的帧格式不符 硬件置位，软件写 1 清零
8	SYNCERR	Sync 脉冲宽度小于 SYNCMIN 或大于 SYNCMAX，硬件置位，软件写 1 清零
7:6	--	RFU：未实现，读为 0
5	FTXF	Frame 发送完成标志，硬件置位，软件写 1 清零
4	NTXF	Nibble 发送完成标志，硬件置位，软件写 1 清零
3	SYNCF	Sync 脉冲接收完成标志，硬件置位，软件写 1 清零 此标志置位的同时更新 RCAL
2	--	RFU：未实现，读为 0
1	FRXF	Frame 接收完成标志，硬件置位，软件写 1 清零
0	NRXF	Nibble 接收完成标志，硬件置位，软件写 1 清零

22.5.6 SENT 数据寄存器（SENT_DR）

名称	SENT_DR0							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CRC				DATA6			
位权限	R/W-0000				R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DATA5				DATA4			
位权限	R/W-0000				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA3				DATA2			
位权限	R/W-0000				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA1				STATUS			
位权限	R/W-0000				R/W-0000			

Bit	助记符	功能描述
31:28	CRC	CRC nibble 发送模式： 当 CRCEN=1 时，此数据由硬件在发送过程中自动计算产生，软件只可读、不可写 当 CRCEN=0 时，此数据由软件在发送前写入 接收模式： 此数据为收到的 CRCnibble 值；当 CRCEN=1 时，硬件在接收过程中自动计算 CRC，并将结果与此数据比较；当 CRCEN=0，硬件不做 CRC 计算和比较，由软件自行校验
27:24	DATA6	Data nibble 6
23:20	DATA5	Data nibble 5
19:16	DATA4	Data nibble 4
15:12	DATA3	Data nibble 3
11:8	DATA2	Data nibble 2
7:4	DATA1	Data nibble 1 发送时由软件写入，接收时由硬件更新，软件读取
3:0	STATUS	Status and communication nibble 发送时由软件写入，接收时由硬件更新，软件读取

22.5.7 SENT 同步脉冲诊断寄存器（SENT_SYNC）

名称	SENT_SYNC							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				SYNCMAX			
位权限	U-0				R/W-1111			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SYNCMAX							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				SYNCMIN			
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SYNCMIN							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:28	--	RFU：未实现，读为 0
27:16	SYNCMAX	Sync 长度上限，检测到 sync 长度大于此值时产生错误事件 单位是 F_{SENT} 时钟周期，即 ticks/8
15:12	--	RFU：未实现，读为 0
11:0	SYNCMIN	Sync 长度下限，检测到 sync 长度小于此值时产生错误事件 单位是 F_{SENT} 时钟周期，即 ticks/8

22.5.8 SENT 同步校准寄存器（SENT_RCAL）

名称	SENT_RCAL
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	-		BASECNT					
位权限	U-0		R-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				RCAL			
位权限	U-0				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RCAL							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	BASECNT	保存同步过程中记录下的 Tick 时间对应的技术值，结合当前时钟可计算出模块得到的 tick 宽度 仅配置为接收工作时有效 单位是 F_{SENT} 时钟周期
15:12	--	RFU：未实现，读为 0
11:0	RCAL	SyncCalibration，只读，保存 syncpulse 测量结果，单位 ticks/8

23 经典控制器区域网络（FSCAN）

23.1 概述

FSCAN模块用于经典CAN总线数据收发，支持CAN2.0B协议。本芯片支持1路FSCAN模块。

FSCAN模块主要特性如下：

- 符合ISO11898-1协议，支持CAN2.0A和CAN2.0B标准
- 支持standard（11bit ID）和extended（29bit ID）帧
- 支持最高波特率1Mbps
- 8 messages接收FIFO
- 3个独立的发送buffer
- 支持错误或仲裁失败条件下的自动重发
- 最大32个扩展ID接收滤波器
- 支持loopback模式
- 支持异步唤醒
- 发送和接收错误计数器
- 自动bus-off恢复或软件控制的bus-off恢复

23.2 结构框图

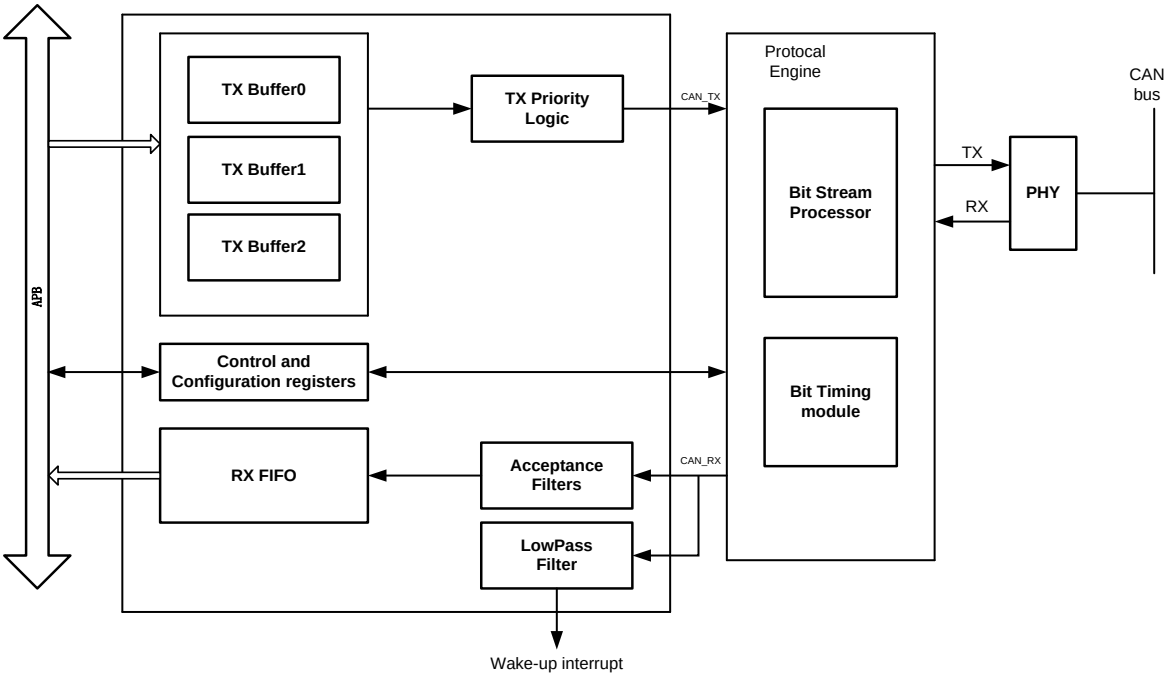


图 23-1 CAN 模块结构框图

FSCAN模块需要外接一个CAN PHY芯片用于实现物理层数据收发。

23.3 功能描述

23.3.1 时钟和复位

FSCAN模块有两个时钟：CAN_CLK和SYS_CLK。

FSCAN总线的数据收发使用CAN_CLK进行，APB总线的寄存器访问使用SYS_CLK进行。CAN_CLK和SYS_CLK之间没有频率约束关系。系统上SYS_CLK连接到APBCLK。

- CAN_CLK频率范围：8~24MHz
- SYS_CLK频率范围：8~80MHz

CAN_CLK可以选择使用芯片内部高频环振、外部晶振、PLL，使用时钟的频率精度需满足ISO11898-1规定的容差范围。

CAN模块的复位包括系统复位、RMU软件复位、模块内部软件复位。其中，通过对模块内部寄存器SRST写1，可以实现模块内部的自复位操作，此时模块内所有配置和控制寄存器，包括SRST本身，

会被复位成复位值。

23.3.2 Bit timing

CAN数据bit时序可以划分为4个部分：

- 同步段
- 传输段
- 相位段1
- 相位段2



图 23-2 CAN bit timing

每个段都是由一定数量的时间片组成，称为time quanta（tq），量子时钟周期决定tq时间长度，而量子时钟是由CAN模块内部对CAN_CLK分频得到的。通过设置波特率预分频寄存器BRPR可以配置量子时钟相对于CAN_CLK的分频系数。

传输段和相位段1组成了时间段1（TS1），相位段2组成时间段2（TS2）；通过Bit-Timing寄存器BTR可以设置TS1和TS2中tq的数量。同步段总是1个tq长度。

23.3.3 Bit stream processor

BSP模块用于实现CAN总线协议中的MAC/LLC功能。在数据发送中BSP负载完成以下工作：

- 并行数据串行化
- 根据协议要求插入stuff bits、CRC、和其他协议数据域

在发送时BSP同时监控接收数据以执行总线仲裁，当仲裁失败时，将执行重发任务。

在接收时BSP移除stuff bits、CRC位和其他协议数据域，将有效负载数据存入RX FIFO，并进行数据错误监控。

根据总线错误状态，BSP控制CAN控制器进入对应的错误状态：Error Active，Error Passive，Bus Off。

当TX或RX数据信号上发现传输错误事件，BSP模块根据ISO11898-1标准规定的规则更新总线错误计数器（CAN_ECR），并根据ECR的值使CAN控制器进入各种错误状态。

23.3.4 控制器工作模式

CAN控制器模块支持以下工作模式：

- Configuration
- Normal
- Sleep
- Loop Back

Configuration mode

当以下任意事件发生时，CAN控制器将进入Configuration mode：

- CEN寄存器写0
- SRST寄存器写1
- CAN控制器复位后默认进入configuration mode

此模式下CAN控制器处于如下状态：

- 停止与总线同步，并驱动固定的隐性位输出
- ECR和ESR寄存器复位
- BTR和BRPR寄存器可以改写
- CAN控制器不再接收任何消息
- TX FIFO和TX HPB中的消息不会被发送，但是会被保留；当进入Normal模式后，这些消息会被发送
- 可以读取RX FIFO中的消息
- 可以向TX FIFO和TX HPB写入消息
- 所有配置寄存器可以访问

在configuration mode下，置位CEN寄存器将使CAN控制器在等待11个连续隐性位之后退出此模式。退出configuration mode后，根据LBACK和SLEEP寄存器状态，CAN控制器会进入Normal、LoopBack或Sleep模式。

Normal mode

在normal mode下，CAN控制器参与总线通信，正常收发消息。从normal模式控制器可以进入Confiuration或Sleep模式。

Sleep mode

CAN控制器可以从Configuration或者Normal模式进入Sleep模式。

在Normal模式下，如果SLEEP寄存器置1，TX buffer中没有剩余消息，并且CAN总线IDLE，CAN控

制器将进入Sleep模式。

在Configuration模式下，如果LPBACK寄存器为0，SLEEP寄存器为1，且CEN寄存器为1，CAN控制器将进入Sleep模式。

当总线上其他节点发送消息，或者软件对TX buffer写入数据，CAN控制器将退出Sleep模式。控制器进入或退出Sleep模式时都能够产生中断。

在Sleep模式下，CAN控制器可以进入Configuration或者Normal模式。

Loop Back mode

在loop back模式下，CAN控制器不参与总线通信，仅向总线持续发送隐性位，也不接收CAN总线上的任何消息。同时内部发送的bit流直接环回到接收端，实现自发自收，用于测试和诊断。

控制器仅能从configuration模式进入loop back模式，通过设置LPBACK寄存器为1，并且CEN寄存器为1。

23.3.5 寄存器写权限与控制器模式

以下寄存器仅能在configuration模式下改写：

1、MSR寄存器中：

TPRI（发送优先级配置），WUPE（唤醒使能），AREN（自动重发），ABOM（busoff恢复配置），LPBACK

2、位时序相关寄存器：

CAN_BRPR, CAN_BTR

3、其他寄存器：

TXB_CR[TXIE]

23.3.6 消息帧存储和消息帧结构（Message storage and structure）

消息存储

CAN总线通信以消息帧（Message Frame）为基本组织形式。待发送的消息保存在TX buffer中，模块共包含3个独立的发送buffer，并具有固定的优先级：buffer0 > buffer1 > buffer2，即编号小的发送buffer优先级高。

CAN控制器接收的消息首先经过acceptance filter过滤，符合滤波器规则的消息才会被存入RX FIFO，否则将被丢弃。RX FIFO中可以保存2条消息。

消息的发送和接收遵从以下规则：

- 编号越小的发送buffer优先级越高
- 发送过程中出现仲裁失败或错误，CAN控制器将尝试重发当前消息；只有当前消息成功发送完

成后，才会发送后续消息，即使是HPB中的消息也不能打断当前消息的重发过程

- TX buffer、RX FIFO中的消息数据在CAN控制器处于Bus Off状态或Configuration模式时仍会保留

消息结构

CAN总线的每条消息长度为16字节，包含消息ID（4字节）、数据长度码DLC（4字节）、Data Word 1（4字节）和Data Word 2（4字节）。

消息ID（Identifier）

bit	31	30:13	12	11	10:0
field	RTR	ID[17:0]	IDE	SRR/RTR	ID[28:18]

数据长度码（Data Length Code – DLC）

bit	31:4	3:0
field	RFU	DLC

数据字1（Data Word 1）

bit	31:24	23:16	15:8	7:0
field	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]

数据字2（Data Word 2）

bit	31:24	23:16	15:8	7:0
field	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]

消息接收时，软件必须从RX FIFO读取完整16字节，即使接收的消息数据长度小于8字节；每个消息读取分4次32bit读操作完成，按照ID、DLC、DW1、DW2的顺序依次从FIFO读出。

消息发送时，软件必须向TX buffer写入完整的16字节，必须按照ID、DLC、DW1、DW2的顺序依次写入4个word，即使有效数据长度少于8字节，也必须完成4个word写入，DW中空余位应填充0。数据发送时，每个字节的MSb先发，数据字节按照DB0~DB7的顺序发送。

消息识别码（Identifier）

CAN控制器支持两种格式的消息识别码：

- 标准帧：ID长度为11bit，仅ID[28:18]、SRR/RTR和IDE为有效位，其中IDE为0，SRR/RTR根据数据帧和远程帧来选择。
- 扩展帧：ID长度为29bit，所有位都有效，SRR/RTR和IDE为1

Bits	Name	Descriptions
31	RTR	Remote Transmission Request

		仅扩展帧有效 1：remote frame 0：data frame
30:13	ID[18:0]	Extended Message ID 仅扩展帧有效 标准帧应为全0
12	IDE	Identifier Extension 1：使用扩展消息识别码，29位ID 0：使用标准消息识别码，11位ID
11	SRR/RTR	Substitute Remote Transmission Request 仅标准帧有效，扩展帧应保持1 1：表示当前消息帧是一个remote frame 0：表示当前消息帧是一个data frame
10:0	ID[28:18]	Standard Message ID 标准帧和扩展帧都有效

23.3.7 控制器初始化

芯片上电复位或系统复位后，在进行CAN总线通信前，需按照以下说明配置CAN总线寄存器。

- 选择操作模式

对于回环模式，置位MSR中的LPBACK位，清零MSR中的SLEEP位。

对于睡眠模式，置位MSR中的SLEEP位，清零MSR中的LPBACK位。

对于正常模式，清零MSR中的LPBACK和SLEEP位。

- 配置传输层配置寄存器

对波特率预分频寄存器BRPR和位时序寄存器BTR进行编程，使其与网络定时参数和系统的网络特性相对应。

- 配置接收滤波寄存器

清零CAN_AFR寄存器中的UAF位，并通过FSC和MSKM寄存器配置滤波器长度、个数和滤波策略

查询CAN_SR中的ACFBSY位直到返回0

将适当的掩码信息写入接收滤波掩码寄存器AFMRx

将适当的ID信息写入到接收滤波寄存器AFIRx

置位CAN_AFR寄存器中的UAF位

对每组需使能的接收滤波掩码寄存器AFMRx和接收滤波寄存器AFIRx对重复上述步骤

配置接收中断使能寄存器CAN_RX_IER以选择接收中断状态寄存器CAN_RX_ISR中可生成中断的标志位

置位CAN_CR寄存器中的CEN位来启用CAN

23.3.8 消息发送

消息发送采用队列形式，包含3个独立的发送buffer，每个发送buffer保存一个完整的消息帧。

软件可以直接操作所有的发送buffer，每个发送buffer有独立的发送请求（transmit request）、发送撤销请求（transmit abort request）、buffer空标志，并根据buffer编号具有固定优先级。

软件写buffer前需要根据TXE寄存器判断发送buffer是否空闲，对非空闲buffer写入的新数据将被丢弃。

23.3.8.1 发送优先级判决

如果多个发送buffer的发送请求同时有效，发送优先级判决将被执行。优先级判决可以基于以下两种方法：

- ID优先级（TPRI=0）
- Buffer编号优先级（TPRI=1）

当TPRI=0，发送优先级判决基于buffer中的消息ID。消息ID的优先级规则符合ISO11898-1规定的仲裁规则，经过内部仲裁后，ID值最小的消息帧将被首先发送。

当TPRI=1，发送优先级判决基于buffer编号。请求发送的多个buffer中编号最小的被最先发送。

在使能了自动重发功能后，如果出现发送错误或者仲裁失败，上一次赢得优先发送的buffer将被自动重发。

23.3.8.2 发送流程

- 查询TXE寄存器，确认当前发送buffer是否空闲
- 将待发送消息帧或远程帧写入目标发送buffer，写1清零对应的TXE寄存器，表示buffer中数据就绪可以发送
- 根据需要置位TXE中断使能
- FSCAN自动进行优先级判决并在总线空闲时启动发送
- 如果消息成功发送，TXE寄存器自动置位，TXSF[1:0]寄存器指示发送完成状态（成功发送或者abort），如果中断使能，则发起TXE中断

Buffer空寄存器TXE由软件清零，并在以下条件下自动置位：

- 发送成功完成
- 发送成功撤销
- 仲裁失败或发送出现错误，并且AREN=0（关闭自动重发）

23.3.8.3 发送撤销（transmit abort）

当某个发送buffer的发送请求寄存器被置位，软件可以通过置位TXAR寄存器来撤销发送。当发送撤销成功完成，TXE置位，TXSF指示当前发送被成功撤销。

正在发送中的消息不能被撤销。然而，如果一个被撤销但是已经在发送的消息输掉了仲裁或发生了错误，该消息不会被自动重发，而是直接结束发送并由TXSF指示当前发送被撤销。

不论发送成功完成还是成功撤销，TXAR都会被自动清零。

23.3.9 消息接收

接收到的消息如果成功通过了消息滤波器，将被存入深度为8的接收FIFO中，同时产生接收完成中断（RXOK）。消息长度为固定的16字节，软件必须从RX FIFO读取完整16字节，即使接收的消息数据长度小于8字节；每个消息读取分4次32bit读操作完成，按照ID、DLC、DW1、DW2的顺序依次从FIFO读出。

当接收FIFO已满的情况下收到新的消息，将导致FIFO溢出事件中断，新的消息帧将被丢弃。

每当接收到新消息并将其写入RX FIFO时，CAN_ISR中的RXNEMP位和RXOK位都会被置1。如果对空RX FIFO执行读取操作，则CAN_ISR中的RXUFLW位置1。

从RX FIFO读取消息：

1、轮询CAN_ISR中的RXOK或RXNEMP位。在中断模式下，读FIFO可以在CAN_ISR中的RXOK或RXNEMP位生成中断后发生执行。

-从接收 FIFO ID寄存器读取消息的ID

-从接收 FIFO DLC寄存器读取消息的数据长度

-从接收 FIFO DW1位置读取消息DW1

-从接收 FIFO DW2位置读取消息DW2

执行读取后，如果RX FIFO中有一条或多条消息，则置位CAN_ISR中的RXNEMP位。该位既可以被轮询，也可以生成中断。

重复此操作，直到FIFO为空。

23.3.10 消息滤波器（Acceptance Filter）

CAN控制器带有16组32bit消息滤波器，每组消息滤波器包含一个掩码寄存器（AFMR）和一个ID寄存器（AFIR）。

滤波器模式

每个消息滤波器组（AFIR+AFMR）支持两种模式，通过Filter Scale寄存器FSC[3:0]配置：

- 扩展模式（default）：包含11位base ID、18位ID extension、SRR、IDE、RTR
- 标准模式：包含11位base ID、RTR、IDE、ID extension的高3bit

16组消息滤波器由16bit使能寄存器UAF1~15使能或关闭。消息滤波器默认为扩展模式，当使用标准模式时，每组的2个16bit滤波器不能单独使能或关闭。掩码寄存器总是与ID寄存器一起使能和关闭。

掩码寄存器用作ID寄存器

每个Mask寄存器也可以用作ID寄存器，因此FSCAN最大可以支持32个32bit绝对ID滤波，由MSKM[3:0]寄存器配置。

例如，MSKM=1000，掩码寄存器CAN_AFMR4将被用作ID寄存器，格式定义与CAN_AFIR相同。

映射关系

下面以第一组消息滤波器为例说明消息滤波器寄存器和消息ID的映射关系

1）FSC[0]=0，MSKM=0，32bit滤波＋掩码

ID	**AFIR0[31:24]**	**AFIR0[23:16]**	**AFIR0[15:8]**	**AFIR0[7:0]**
MASK	**AFMR0[31:24]**	**AFMR0[23:16]**	**AFMR0[15:8]**	**AFMR0[7:0]**

mapping	ID[28:18] (EXID[17:7])	SRR	IDE	ID[17:11] (EXID[6:0])	ID[10:0] (SID[10:0])	RTR

这种模式下，共支持16组32bit带掩码的消息ID滤波。

2）FSC[0]=0，MSKM=1，2组32bit滤波

ID	**AFIR0[31:24]**	**AFIR0[23:16]**	**AFIR0[15:8]**	**AFIR0[7:0]**
ID	**AFMR0[31:24]**	**AFMR0[23:16]**	**AFMR0[15:8]**	**AFMR0[7:0]**

mapping	ID[28:18] (EXID[17:7])	SRR	IDE	ID[17:11] (EXID[6:0])	ID[10:0] (SID[10:0])	RTR

这种模式下，共支持32组32bit绝对消息ID滤波。

3）FSC[0]=1，MSKM=0，2组16bit滤波＋掩码

ID	**AFIR0[31:24]**	**AFIR0[23:16]**
MASK	**AFMR0[31:24]**	**AFMR0[23:16]**

ID	**AFIR0[15:8]**	**AFIR0[7:0]**
MASK	**AFMR0[15:8]**	**AFMR0[7:0]**

mapping	SID[10:0]	RTR	IDE	EXID[17:15]

这种模式下，共支持32组16bit带掩码的消息ID滤波。

注：上图中RTR在标准帧中代表RTR，在扩展帧中代表SRR；如果IDE=1，表示过滤扩展帧，低3bit作为扩展ID[17:15]参与比较，如果IDE=0，表示过滤标准帧，低3bit不参与比较。此时，32bit滤波器被展开成2组16bit滤波器使用。

4）FSC[0]=1，MSKM=1，4组16bit滤波

ID	**AFIR0[31:24]**	**AFIR0[23:16]**
ID	**AFMR0[31:24]**	**AFMR0[23:16]**

ID	**AFIR0[15:8]**	**AFIR0[7:0]**
ID	**AFMR0[15:8]**	**AFMR0[7:0]**

mapping	SID[10:0]	RTR	IDE	EXID[17:15]

这种模式下，共支持64组16bit绝对消息ID滤波。

注：上图中RTR在标准帧中代表RTR，在扩展帧中代表SRR；如果IDE=1，表示过滤扩展帧，低3bit作为扩展ID[17:15]参与比较，如果IDE=0，表示过滤标准帧，低3bit不参与比较。此时，32bit滤波器被展开成4组16bit滤波器使用。

消息滤波工作流程

- 接收到的消息ID先与Mask寄存器进行逻辑与（如果掩码寄存器被用作ID滤波则跳过这个步骤）
- 滤波器ID寄存器内容同样与Mask寄存器进行逻辑与（如果掩码寄存器被用作ID滤波则跳过这个步骤）

- 两个逻辑与操作结果进行比较（如果掩码寄存器被用作ID滤波则直接比较输入ID和ID寄存器）
- 如果两个结果相等，本消息通过滤波器，被存入RX FIFO
- 多个滤波器可以同时工作，接收消息只要通过其中任意滤波器，就会被存入RX FIFO
- 无法通过滤波器的消息帧被丢弃

注意：软件可以选择使能任意多个消息滤波器，或者关闭所有消息滤波器。如果没有一个消息滤波被使能，则所有接收到的消息都会存入RX FIFO。

23.3.11 错误管理

23.3.11.1 节点策略

FSCAN 包含发送错误计数器和接收错误计数器。当消息帧被正确的发送或接收，相应的错误计数器应被递减，而当在发送或接收中遇到错误，相应的错误计数器应被递增。

下面罗列了错误计数的详细规则：

- 当接收节点检测到错误，接收错误计数器+1 （例外：发送主动错误标志或过载标志时检测到的 bit 错误不会导致接收错误计数器递增）
- 接收节点发送主动错误标志后，检测到后续第一个 bit 是显性位，则接收错误计数器+8
- 发送节点发送错误标志，发送错误计数器+8；以下例外情况下发送错误计数器保持不变
  - 发送节点处于被动错误状态，检测到 ACK 错误，并且被动错误标志中没有出现显性位
  - 仲裁阶段检测到填充位错误，并且发送的是隐性填充位，实际检测到显性填充位
- 发送节点在发送主动错误标志或过载标志时检测到 bit 错误，发送错误计数器+8
- 当检测到总线上连续 14 个显性位，或者被动错误标志后检测到连续 8 个显性位，发送节点应将发送错误计数器+8，接收节点应将接收错误计数器+8
- 当一个消息帧被正确发送后，发送错误计数器-1，减到 0 停止
- 当一个消息帧被正确接收后，接收错误计数器如果在 1~127 之间，则应-1，如果是 0 则保持不变，如果大于 127 则被设置一个 119 到 127 之间的数

23.3.11.2 错误状态转移

FSCAN 退出 configuration 模式后，进入主动错误状态（error active）。此状态下，CAN 控制器检

测到错误后会主动发送错误帧。

当接收错误计数器或发送错误计数器的计数值大于 127，FSCAN 将进入被动错误状态（error passive）。此状态下，FSCAN 不再发送错误帧。

当被动错误节点的接收错误计数器和发送错误计数器的计数值都小于等于 127 时，FSCAN 回到主动错误状态。

当节点的发送错误计数器值大于 255，节点应进入总线关闭状态（bus-off）。

处于总线关闭状态的节点不能发送任何帧，也不能发送 ACK，对总线无影响。

节点错误状态转移可以符合下图所示的 ISO11898-1 协议规范。

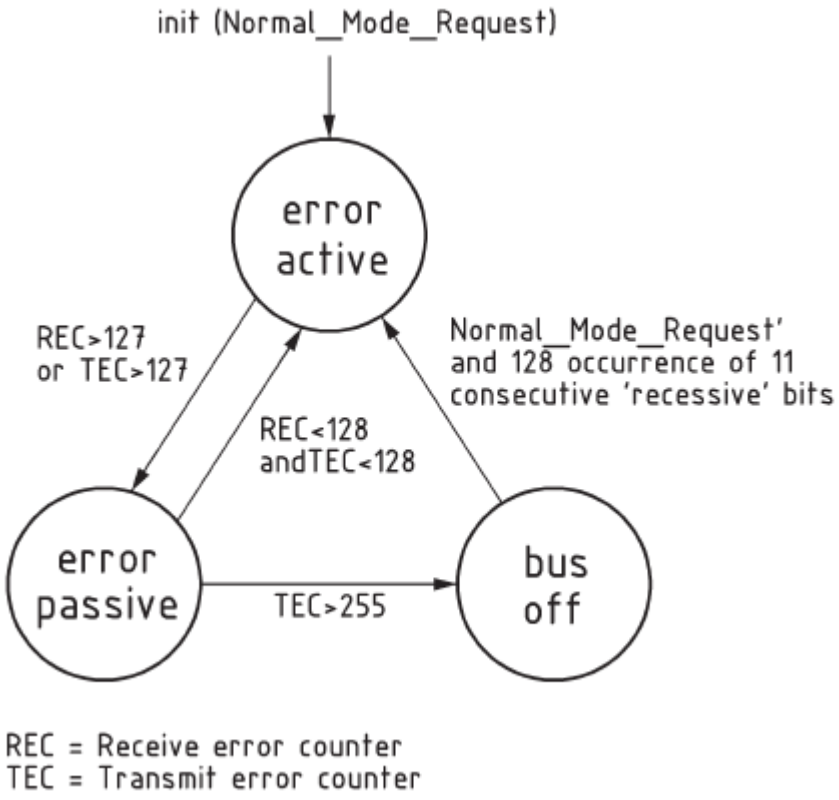


图 23-3 节点错误状态转移

FSCAN定义了以下错误状态：

- 正常接收、正常发送：TEC和REC都小于96
- 接收warning：REC大于等于96并且小于128
- 接收error：REC大于等于128
- 发送warning：TEC大于等于96并且小于128
- 发送error：TEC大于等于128并且小于256
- Bus-off：TEC大于等于256

当FSCAN内部状态机在以上状态之间发送切换时，ESCF中断标志寄存器置位，可以通知软件采取措施。同时，RSTS和TSTS寄存器用于指示当前CAN控制器所处的状态，软件也可以通过查询REC和TEC寄存器来确定控制器状态。

23.3.11.3 Bus-off recovery

FSCAN可以支持使用以下策略从总线关闭状态中恢复。

- ABOM=00/11：自动恢复模式

  这种模式下，FSCAN采用符合ISO11898协议规范的总线关闭恢复方式。当FSCAN连续检测到11bit隐性位超过128次，控制器自动回到error active状态。发送和接收错误计数器TEC和REC都自动清零，并且BORF中断标志置位。

- ABOM=01/10：软件恢复模式

  当ABOM=01，控制器退出bus-off由软件触发。软件置位RTBO后，控制器等待128次连续11位隐性位，然后退出bus-off state

  当ABOM=10，软件置位RTBO后，控制器立即退出bus-off state并进入integration state，在11个连续隐性位后，控制器重新加入CAN总线。

23.3.11.4 错误管理中断和标志

错误管理中断标志寄存器用于通知软件FSCAN内部错误状态发生了转换。

23.3.12 异步唤醒

FSCAN能够在没有CANCLK时钟的情况下监视CAN_RX信号的下降沿（隐性电平->显性电平），并产生异步唤醒中断，将处于休眠模式的MCU唤醒。下降沿检测带有一个模拟滤波电路，能够滤除小于100ns的毛刺信号，避免噪声干扰引起误唤醒。

借助此功能，可以在芯片休眠状态下实现CAN总线活动唤醒MCU。使用流程如下。

- 等待总线空闲
- 使能CAN_RX异步唤醒功能，并置位唤醒中断使能寄存器
- 配置FSCAN进入sleep mode
- 芯片进入休眠
- CAN总线下降沿置位唤醒中断，唤醒MCU
- MCU退出休眠，执行时钟初始化
- 软件配置FSCAN进入configuration或normal

注意：WUPF中断标志仅在FSCAN处于Sleepmode时才会置位。

23.4 寄存器

FSCAN模块起始地址：0x4001A400

offset 地址	名称	符号
0x00	CAN 控制寄存器 (CAN Control Register)	CAN_CR
0x04	CAN 模式选择寄存器 (CAN mode select register)	CAN_MSR
0x08	波特率预分频寄存器 (CAN Baud rate Prescaler Register)	CAN_BRPR
0x0C	CAN 位时序寄存器 (CAN Bit Timing Register)	CAN_BTR
0x10	CAN 错误计数寄存器 (CAN Error Counter Register)	CAN_ECR
0x14	CAN 错误状态寄存器 (CAN Error Status Register)	CAN_ESR
0x18	CAN 状态标志寄存器 (CAN Status Register)	CAN_SR
0x1C	CAN 接收中断标志寄存器 (CAN Interrupt Status Register)	CAN_RXISR
0x20	CAN 接收中断使能寄存器 (CAN Interrupt Enable Register)	CAN_RXIER
0x24		
0x28	CAN 发送缓冲控制寄存器 (CAN Transmit Buffer Control Register)	CAN_TXBCR
0x2C	CAN 发送中断状态寄存器 (CAN Transmit Buffer Status Register)	CAN_TXBSR
0x30	CAN 发送缓冲 0ID 寄存器 (CAN Transmit Buffer 0 ID Register)	CAN_TXB0IDR
0x34	CAN 发送缓冲 0DLC 寄存器 (CAN Transmit Buffer 0 DLC Register)	CAN_TXB0DLCR
0x38	CAN 发送缓冲 0 数据字寄存器 1 (CAN Transmit Buffer 0 DataWord1 Register)	CAN_TXB0DW1R
0x3C	CAN 发送缓冲 0 数据字寄存器 2 (CAN Transmit Buffer 0 DataWord2 Register)	CAN_TXB0DW2R
0x40	CAN 发送缓冲 1 ID 寄存器 (CAN Transmit Buffer 1 ID Register)	CAN_TXB1IDR
0x44	CAN 发送缓冲 1 DLC 寄存器 (CAN Transmit Buffer 1 DLC Register)	CAN_TXB1DLCR
0x48	CAN 发送缓冲 1 数据字寄存器 1 (CAN Transmit Buffer 1 DataWord1 Register)	CAN_TXB1DW1R
0x4C	CAN 发送缓冲 1 数据字寄存器 2 (CAN Transmit Buffer 1 DataWord2 Register)	CAN_TXB1DW2R
0x50	CAN 发送缓冲 2ID 寄存器	CAN_TXB2IDR

	(CAN Transmit Buffer 2 ID Register)	
0x54	CAN 发送缓冲 2 DLC 寄存器 (CAN Transmit Buffer 2 DLC Register)	CAN_TXB2DLCR
0x58	CAN 发送缓冲 2 数据字寄存器 1 (CAN Transmit Buffer 2 DataWord1 Register)	CAN_TXB2DW1R
0x5C	CAN 发送缓冲 2 数据字寄存器 2 (CAN Transmit Buffer 2 DataWord2 Register)	CAN_TXB2DW2R
0x60	CAN 接收滤波控制寄存器 (Acceptance Filter Control Register)	CAN_AFCR
0x64	CAN 接受滤波配置寄存器 (Acceptance Filter Config Register)	CAN_AFCFGR
0x80	CAN 接收滤波掩码寄存器 0 (Acceptance Filter Mask Register0)	CAN_AFMR0
0x84	CAN 接收滤波 ID 寄存器 0 (Acceptance Filter ID Register0)	CAN_AFIR0
0x88	CAN 接收滤波掩码寄存器 1 (Acceptance Filter Mask Register1)	CAN_AFMR1
0x8C	CAN 接收滤波 ID 寄存器 1 (Acceptance Filter ID Register1)	CAN_AFIR1
…	…	…
0xF8	CAN 接收滤波掩码寄存器 15 (Acceptance Filter Mask Register15)	CAN_AFMR15
0xFC	CAN 接收滤波 ID 寄存器 15 (Acceptance Filter ID Register15)	CAN_AFIR15
0x100	接收 FIFO ID 寄存器 (CAN RX FIFO ID Register)	CAN_RXFIDR
0x104	接收 FIFO DLC 寄存器 (CAN RX FIFO DLC Register)	CAN_RXFDLCR
0x108	接收 FIFO Data Word1 寄存器 (CAN RX FIFO Data Word1 Register)	CAN_RXFDW1R
0x10C	接收 FIFO Data Word2 寄存器 (CAN RX FIFO Data Word2 Register)	CAN_RXFDW2R

23.4.1 CAN 控制寄存器（CAN_CR）

名称	CAN_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	-						CEN	SRST

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	CEN	CAN 控制器使能 (CAN enable) 0：CAN 控制器处于 configuration mode 1：CAN 控制器根据 LPBACK 和 SLEEP 寄存器状态，进入 LoopBack、Sleep 或 Normal 模式
0	SRST	软件复位寄存器 (Software reset) 写 1 复位 CAN 控制器，读出总是 0

23.4.2 CAN 模式选择寄存器（CAN_MSR）

名称	CAN_MSR							
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	-		RX_PRESYNC_EN	TPRI	WUPE	AREN	ABOM	
位权限	U-0		R/W-0	R/W-1	R/W-0	R/W-1	R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RTBO	-					LPBACK	SLEEP
位权限	R/W-0	U-0					R/W-0	R/W-0

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13	RX_PRESYNC_EN	CAN 接收数据预采样同步使能 0：不经过预采样 1：经过预采样
12	TPRI	发送优先级模式选择 1：采用发送 buffer 编号优先级方式 0：采用消息 ID 优先级方式
11	WUPE	唤醒使能 1：允许 MCU 休眠时使用 CAN_RX 信号下降沿唤醒 0：禁止唤醒功能
10	AREN	自动重发使能 1：使能自动重发功能 0：关闭自动重发
9:8	ABOM	自动总线关闭管理(Automatic Bus Off Management) 此寄存器控制 CAN 在退出 bus-off 状态时的行为 00/11：检测到 128 次连续 11 个隐性位的情况下，硬件自动退出

Bit	助记符	功能描述
		bus-off state（兼容 ISO11898） 01：软件退出 bus-off：对 RTBO 置位后，控制器等待检测到 128 次连续 11 个隐性位的情况下退出 bus-off state 10：软件退出 bus-off：对 RTBO 置位后，控制器立即退出 bus-off state 并进入 integration state
7	RTBO	软件强制从 bus-off 退出 此寄存器仅在 ABOM=01/10 的情况下有效且可写 软件置位后，控制器在退出 bus-off state 后自动清零
6:2	--	RFU：未实现，读为 0
1	LPBACK	Loop Back mode 1：CAN 控制器处于 Loop Back 模式 0：CAN 控制器处于 Normal、Configuration 或 Sleep 模式
0	SLEEP	Sleep mode 1：CAN 控制器处于 Sleep 模式 0：CAN 控制器处于 Normal、Configuration 或 Loop Back 模式

23.4.3 CAN 波特率预分频寄存器（CAN_BRPR）

名称	CAN_BRPR							
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	BRP							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	BRP	波特率预分频 (Baud Rate Prescaler)，根据这个寄存器的值对 CAN_CLK 进行预分频得到量子时钟频率 实际分频值是 BPR+1 Tq=tCAN_CLK * (BPR+1)

23.4.4 CAN 位时序寄存器（CAN_BTR）

名称	CAN_BTR
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	-							SJW[1]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SJW[0]	TS2			TS1			
位权限	R/W-0	R/W-000			R/W-0000			

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8:7	SJW	Synchronization Jump Width，参见 CAN2.0 协议 tSJW = tq * (SJW + 1)
6:4	TS2	Time Segment2，定义 Phase Segment2 的长度 tTS2 = tq * (TS2 + 1)
3:0	TS1	Time Segment1，定义 Propagation Segment + Phase Segment1 的长度 tTS1 = tq * (TS1 + 1)

23.4.5 CAN 错误计数寄存器（CAN_ECR）

名称	CAN_ECR							
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	REC							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TEC							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	REC	接收错误计数器 (Receive Error Counter) 软件只读，每次发生接收错误时 REC 加 1
7:0	TEC	发送错误计数器 (Transmit Error Counter) 软件只读，每次发生发送错误时 TEC 加 1

注：在以下情况下ECR寄存器清零

- SRST寄存器写1
- CEN寄存器清零
- CAN控制器进入Bus Off
- Bus off恢复

23.4.6 CAN 错误状态寄存器（CAN_ESR）

名称	CAN_ESR							
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	-							OVLF
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BLF	ADERR	ACKER	B0ERR	B1ERR	STER	FMER	CRCER
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
31:9	--	RFU：未实现，读为 0
8	OVLF	Overload flag，总线上接收到过载帧时置位，软件写 1 清零 1：检测到 overload 0：没有检测到 overload
7	BLF	Bus Lock Flag，当总线上连续检测到 32 个显性位时置位，软件写 1 清零 1：bus locked 0：no bus lock
6	ADERR	ACK delimiter error 1：检测到 ACK delimiter 错误 0：无 ACK delimiter 错误 硬件置位，软件写 1 清零
5	ACKER	Acknowledge Error 1：检测到 ACK 错误 0：无 ACK 错误 硬件置位，软件写 1 清零
4	B0ERR	Bit0 Error 1：接收到的 bit 不是正在发送的 dominant bit 0：无 bit 错误 硬件置位，软件写 1 清零
3	B1ERR	Bit1 Error 1：接收到的 bit 不是正在发送的 recessive bit 0：无 bit 错误

Bit	助记符	功能描述
		硬件置位，软件写 1 清零
2	STER	Stuffing Error 1：数据填充位错误 0：无填充错误 硬件置位，软件写 1 清零
1	FMER	Form Error 1：消息帧中固定格式域出现错误 0：无错误 硬件置位，软件写 1 清零
0	CRCER	CRC Error 1：CRC 校验错误 0：CRC 校验正确 硬件置位，软件写 1 清零

23.4.7 CAN 状态标志寄存器（CAN_SR）

名称	CAN_SR							
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	-				ACFBSY	RXFE	RSTS	
位权限	U-0				R-0	R-1	R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TSTS		BBSY	BIDLE	NORMAL	SLEEP	LPBACK	CONFIG
位权限	R-00		R-0	R-0	R-0	R-0	R-0	R-1

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	ACFBSY	Acceptance Filter Busy 1：Acceptance Filter 工作，不能改写 Acceptance Filter Mask 和 ID 寄存器 0：Acceptance Filter 空闲，可以改写 Mask 和 ID 寄存器
10	RXFE	Receive FIFO empty 标志 1：接收 FIFO 空 0：接收 FIFO 非空
9:8	RSTS	FSCAN 内部接收错误状态指示 00：正常接收状态，REC<96 01：Warning 状态，96≦REC<128 10：Error 状态，128≦REC 11：Bus-off，这是一个冗余状态，因为接受错误并不会使 CAN

Bit	助记符	功能描述
		控制器进入 bus-off，这个状态与 TSTS=11 等价
7:6	TSTS	FSCAN 内部发送错误状态指示 00：正常发送状态，TEC<96 01：Warning 状态，96≦TEC<128 10：Error 状态，128≦TEC<256 11：Bus-off，256<TEC
5	BBSY	Bus Busy 标志，只读 1：CAN 控制器正在进行数据收发 0：CAN 控制器处于 configuration mode，或总线 IDLE
4	BIDLE	Bus IDLE 标志，只读 1：当前无总线通信 0:当前总线正在通信,或者 CAN 控制器处于 configuration mode
3	NORMAL	Normal mode 标志，只读 1：CAN 控制器处于 normal mode 0：CAN 控制器不处于 normal mode
2	SLEEP	Sleep mode 标志，只读 1：CAN 控制器处于 sleep mode 0：CAN 控制器不处于 sleep mode
1	LBACK	Loop Back mode 标志，只读 1：CAN 控制器处于 loop back mode 0：CAN 控制器不处于 loop back mode
0	CONFIG	Configuration mode 标志，只读 1：CAN 控制器处于 configuration mode 0：CAN 控制器不处于 configuration mode

23.4.8 CAN 接收中断标志寄存器（CAN_RXISR）

名称	CAN_RXISR							
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	-				RXFIFO_MC			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			BORF	WUPF	SLPF	ESCF	ERROR
位权限	U-0			R/W-0	R/W-0	RW-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXNEMP	RXOFLW	RXUFLW	RXOK	-			ARBLST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0			R/W-0

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
19:16	RXFIFO_MC	RXFIFO 中的 message count，有效数据为 0~8
15:13	--	RFU：未实现，读为 0

Bit	助记符	功能描述
12	BORF	Bus-off Recovery Flag，硬件置位，软件写 1 清零 1：CAN 控制器从 bus-off 恢复
11	WUPF	Wakeup 中断标志，硬件置位，软件写 1 清零 1：CAN 控制器在 Sleep 模式下检测到 RX 信号上的下降沿
10	SLPF	SLEEP 中断标志，硬件置位，软件写 1 清零 1：CAN 控制器进入 SLEEP 模式
9	ESCF	错误状态转换中断标志(Error Status Change Flag)，硬件置位，软件写 1 清零 1：CAN 控制器内部错误状态发生转换 此中断标志置位后，软件应查询 RSTS 和 TSTS 寄存器确认当前的错误状态，也可以读取 REC 和 TEC 错误状态转换事件包含进入或者退出以下状态： ● 接收 warning ● 接收 error ● 发送 warning ● 发送 error ● Bus-off
8	ERROR	错误中断标志，硬件置位，软件写 1 清零 1：消息收发过程中发生了错误 *当 CAN_ESR 寄存器中任意 bit 置位时，此寄存器置位；软件响应错误中断时，需要去 CAN_ESR 中查询具体的错误标志*
7	RXNEMP	Receive FIFO Not Empty 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 非空
6	RXOFLW	Receive FIFO Overflow 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 上溢出，即在接收 FIFO 满的情况下收到新的消息
5	RXUFLW	Receive FIFO Underflow 中断标志，硬件置位，软件写 1 清零 1：接收 FIFO 下溢出，即在接收 FIFO 空的情况下执行读操作
4	RXOK	Receive OK 中断标志，硬件置位，软件写 1 清零 1：表示 1 个消息帧被成功接收
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	ARBLST	Arbitration Lost 中断标志，硬件置位，软件写 1 清零 1：仲裁失败

23.4.9 CAN 接收中断使能寄存器（CAN_RXIER）

名称	CAN_RXIER							
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	-			EBORF	EWKUP	ESLP	EESCF	EERROR
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	ERXNEMP	ERXOFLW	ERXUFLW	ERXOK	-			EARBLST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0			R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
12	EBORF	Bus-off Recovery 中断使能，1 允许中断，0 禁止中断
11	EWKUP	Wakeup 中断使能，1 允许中断，0 禁止中断
10	ESLP	SLEEP 中断使能，1 允许中断，0 禁止中断
9	EESCF	错误转换中断使能
8	EERROR	错误中断使能，1 允许中断，0 禁止中断
7	ERXNEMP	Receive FIFO Not Empty 中断使能，1 允许中断，0 禁止中断
6	ERXOFLW	Receive FIFO Overflow 中断使能，1 允许中断，0 禁止中断
5	ERXUFLW	Receive FIFO Underflow 中断使能，1 允许中断，0 禁止中断
4	ERXOK	Receive OK 中断使能，1 允许中断，0 禁止中断
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	EARBLST	Arbitration Lost 中断使能，1 允许中断，0 禁止中断

23.4.10 CAN 发送缓冲控制寄存器（CAN_TXBCR）

名称	CAN_TXBCR							
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	-					TXAR		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					TXIE		
位权限	U-0					R/W-000		

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
10:8	TXAR	发送撤销请求（Transmit Abort request） 软件写 1 撤销发送请求，撤消后硬件自动清零。仅 TXR[y]=1 的

Bit	助记符	功能描述
		情况下对应 TXAR[y]位可写 TXAR[2] – TxBuffer2 TXAR[1] – TxBuffer1 TXAR[0] – TxBuffer0
7:3	--	RFU：未实现，读为 0
2:0	TXIE	TXE 中断使能寄存器 1：允许 TXE 产生中断 0：禁止 TXE 中断

23.4.11 CAN 发送中断标志寄存器（CAN_TXBSR）

名称	CAN_TXBSR							
offset								
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	-		TXRF2		TXRF1		TXRF0	
位权限	U-0		R-00		R-00		R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		TXS			TXE		
位权限	U-0		R -000			R/W-111		

Bit	助记符	功能描述
31:14	--	RFU：未实现，读为 0
13:12	TXRF2	发送结果标志（Transmit buffer 2 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
11:10	TXRF1	发送结果标志（Transmit buffer 1 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
9:8	TXRF0	发送结果标志（Transmit buffer 0 transmit result flag） 00：无发送请求或者还在发送中 01：发送撤销成功 10：发送成功（无撤销请求） 11：发送成功（有撤销请求，未能实现撤销）
7:6	--	RFU：未实现，读为 0
5:3	TXS	发送 buffer 状态标志 0：TxBuffer 不在发送中 1：TxBuffer 发送中

Bit	助记符	功能描述
		TXS[2] – TxBuffer2 TXS[1] – TxBuffer1 TXS[0] – TxBuffer0
2:0	TXE	发送 buffer 空标志，软件写 1 清零（写 0 无效），硬件置位 0：发送 buffer 非空，准备好发送 1：发送 buffer 空 TXE[2] – TxBuffer2 TXE[1] – TxBuffer1 TXE[0] – TxBuffer0 软件将待发消息写入 buffer 后，清零对应的 TXE 寄存器，通知硬件可以发送；发送完成后，控制器自动置位 TXE 并产生中断

23.4.12 CAN 发送缓冲 ID 寄存器（CAN_TXBxIDR, x=0,1,2）

名称	CAN_TXBxIDR, x=0,1,2							
offset	0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IDR[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	IDR[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDR[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDR[7:0]							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:0	IDR	发送消息识别码 (Identifier Register) 数据格式参见 23.3.6 消息帧存储和消息帧结构（Message storage and structure）

23.4.13 CAN 发送缓冲 DLC 寄存器（CAN_TXBxDLCR, x=0,1,2）

名称	CAN_TXBxDLCR, x=0,1,2							
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	-				DLC			
位权限	U-0				W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	DLC	数据长度码 (Data Length Code) 定义消息帧中的数据字节长度

23.4.14 CAN 发送缓冲 DataWord1 寄存器（CAN_TXBxDW1R, x=0,1,2）

名称	CAN_TXBxDW1R, x=0,1,2							
offset	0x48							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB3							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB2							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB1							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB0							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

23.4.15 CAN 发送缓冲 DataWord2 寄存器（CAN_TXBxDW2R, x=0,1,2）

名称	CAN_TXBxDW2R, x=0,1,2							
offset	0x4C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB7							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB6							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB5							
位权限	W-0000 0000							

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB4							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

23.4.16 CAN 接收滤波控制寄存器（CAN_AFCR）

名称	CAN_AFCR							
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	UAF[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	UAF[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	UAF	消息滤波器组使能 (Use Acceptance Filter) 1：使能滤波器 0：关闭滤波器 每个 bit 对应一组消息滤波器

23.4.17 CAN 接收滤波配置寄存器（CAN_AFCFGR）

名称	CAN_AFCFGR							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MSKM[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSKM[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FSC[15:8]							

位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FSC[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	MSKM	掩码寄存器使用模式，每个 bit 对应 1 个 AFMR 0：AFMR 用作掩码 1：AFMR 用作 ID 对应关系： MSKM[15] – AFMR15 … MSKM[3] – AFMR3 MSKM[2] – AFMR2 MSKM[1] – AFMR1 MSKM[0] – AFMR0
15:0	FSC	ID 寄存器扩展模式 0：32bit 扩展模式 1：16bit 标准模式 对应关系： FSC[15] – AFIR15 FSC[3] – AFIR3 FSC[2] – AFIR2 FSC[1] – AFIR1 FSC[0] – AFIR0

23.4.18 CAN 接收滤波掩码寄存器 x（CAN_AFMRx）

名称	CAN_AFMRx (x=0~15)							
offset	0x80 + (x)*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AMRTR	AMID[17:11]						
位权限	R/W-0	R/W-000 0000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AMID[10:3]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AMID[2:0]			AMIDE	AMSRR	AMID[28:26]		
位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AMID[25:18]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31	AMRTR	RTR Mask

Bit	助记符	功能描述
		1：RTR 位参与滤波器比较 0：RTR 位不参与滤波器比较
30:13	AMID[17:0]	ID Mask bit 17-0 1：对应位参与滤波器比较 0：对应位不参与滤波器比较
12	AMIDE	IDE Mask 1：IDE 位参与滤波器比较 0：IDE 位不参与滤波器比较
11	AMSRR	SRR Mask 1：SRR 位参与滤波器比较 0：SRR 位不参与滤波器比较
10:0	AMID[28:18]	ID Mask bit 28-18 1：对应位参与滤波器比较 0：对应位不参与滤波器比较

23.4.19 CAN 接收滤波 ID 寄存器 x（CAN_AFIRx）

名称	CAN_AFIRx (x=0~15)							
offset	0x84 + (x)*0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AIRTR	AIID[17:11]						
位权限	R/W-0	R/W-000 0000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AIID[10:3]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AIID[2:0]			AIIDE	AISRR	AIID[28:26]		
位权限	R/W-000			R/W-0	R/W-0	R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AIID[25:18]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31	AIRTR	滤波器 RTR (Acceptance ID RTR)
30:13	AIID[17:0]	滤波器 ID bit 17-0 (Acceptance ID IDR)
12	AIIDE	滤波器 IDE (Acceptance ID IDE)
11	AISRR	滤波器 SRR (Acceptance ID SRR)
10:0	AIID[28:18]	滤波器 ID bit 28-18 (Acceptance ID IDR)

23.4.20 CAN 接收 FIFO ID 寄存器（CAN_RXF_IDR）

名称	CAN_RXF_IDR							
offset	0x100							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IDR[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	IDR[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDR[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IDR[7:0]							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:0	IDR	发送消息识别码 (Identifier Register) 数据格式参见 23.3.6 消息帧存储和消息帧结构（Message storage and structure）

23.4.21 CAN 接收 FIFO DLC 寄存器（CAN_RXF_DLCR）

名称	CAN_RXF_DLCR							
offset	0x104							
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	-				DLC			
位权限	U-0				W-0000			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	DLC	数据长度码 (Data Length Code) 定义消息帧中的数据字节长度

23.4.22 CAN 接收 FIFO DataWord1 寄存器（CAN_RXF_DW1R）

名称	CAN_RXF_DW1R							
offset	0x108							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB3							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB2							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

位名	DB1							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB0							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

23.4.23 CAN 接收 FIFO DataWord2 寄存器（CAN_RXF_DW2R）

名称	CAN_RXF_DW2R							
offset	0x10C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DB7							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DB6							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DB5							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DB4							
位权限	W-0000 0000							

Bit	助记符	功能描述
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

24 直接存储访问控制器（DMA）

24.1 概述

- 4通道外设PDMA，支持Peripherals<>RAM传输和Peripherals<>Peripherals传输
- 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65536字节（64KB），支持byte/half-word/word传输
- 外设DMA通道支持双缓冲指针（double buffering）
- Flash->RAM通道最大传输长度8192字节，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）

24.2 结构框图

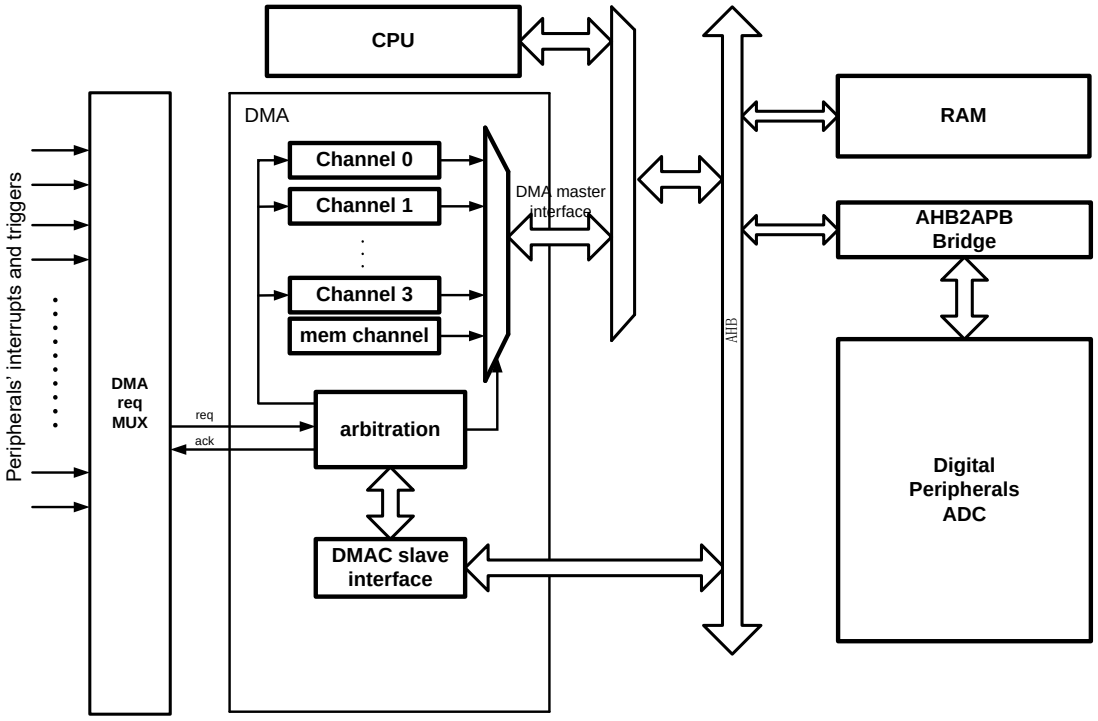


图 24-1 DMA 结构框图

24.3 功能描述

24.3.1 外设 DMA 传输

外设DMA传输以block为单位，block由特定长度的传输构成。当外设请求信号到来时，DMA控制器根据优先级判断响应外设请求，然后执行外设<>RAM或者外设<>外设的数据传输。

详细的外设请求-响应步骤可以表述为：

- 外设发送中断或trigger信号
- DMAMUX电路接收并锁存来自外设的中断或触发信号，产生DMA请求信号
- DMA控制器接收到请求（DMA request），判决优先级以决定是否要立即响应请求
- 当DMA控制器响应请求，ack信号立即被发送给DMAMUX
- DMAMUX接收到ack信号后立即释放req信号
- DMA控制器检测到req释放后，释放ack信号

当DMA响应外设请求后，数据传输开始进行：

- 一次数据传输封装了两个AHB通信
  - 一次从源地址的读操作，目标地址由外设地址寄存器或存储器地址寄存器提供（根据传

输方向），读取位宽由传输带宽寄存器定义

- 一次对目的地地址的写操作，目标地址由外设地址寄存器或存储器地址寄存器提供（根据传输方向），写入位宽由传输带宽寄存器定义

- 传输完成后，传输长度寄存器递减
- 传输完成后，根据寄存器配置，源、目的地址指针可以递增、递减、或者保持

每个外设通道包含两个指针寄存器：外设地址寄存器（PAR）和存储器地址寄存器（MAR）。

- DIR=0
  - 外设地址寄存器（PAR）作为源地址使用
  - 存储器地址寄存器（MAR）作为目的地地址使用
- DIR=1
  - 外设地址寄存器（PAR）作为目的地地址使用
  - 存储器地址寄存器（MAR）作为源地址使用

24.3.2 外设请求的处理

外设DMA请求可以是外设发出的中断信号或者trigger信号。DMAMUX电路负责锁存外设信号，并对DMA控制器发出请求。

外设中断信号可能是异步于系统时钟的，DMAMUX负责进行同步处理。

单个通道的请求-响应逻辑可以由下图表示，外设DMA共有4个相同的通道。

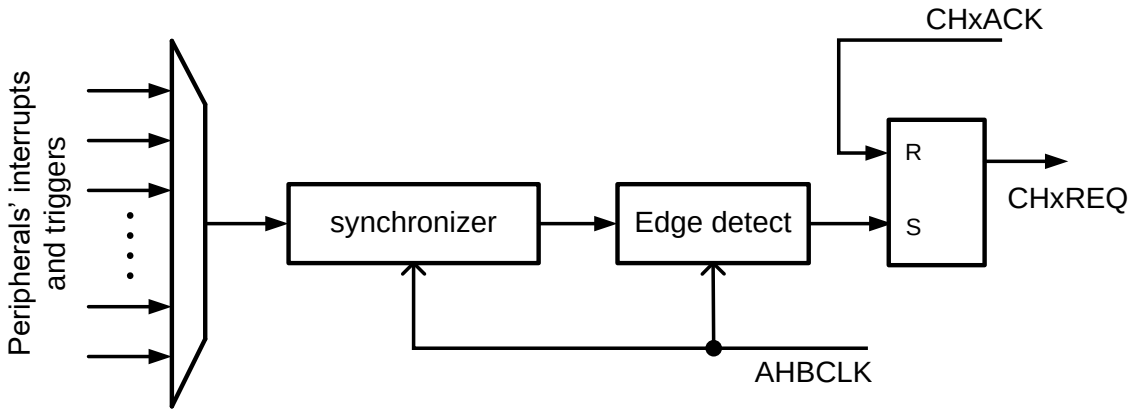


图 24-2 外设请求处理

24.3.3 通道优先级仲裁

DMA总共有5个通道，外设通道占据channel 0~3，存储器通道占据channel4。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。所有5个通道中被使能的通道同时参与优先级仲裁。

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。

24.3.4 传输方向定义

在DMA通道定义规则中，数据传输总是从源地址搬运到目的地址。因此软件配置时需要注意源和目的地址的正确性，DMA控制器不负责进行地址合法性检查。

源地址和目的地址都是32bit宽度，理论上可以寻址整个芯片地址空间。但是不合理的配置会导致错误的结果。

比如希望按一定的定时周期将RAM中的数据传输到UART发送，DIR=0：

- 选择timer输出作为请求信号
- 外设地址寄存器（PAR）指向RAM地址
- 存储器地址寄存器（MAR）指向UART发送buffer地址

以上例子中，如果DIR=1，则：

- 外设地址寄存器（PAR）指向UART发送buffer地址
- 存储器地址寄存器（MAR）指向RAM地址

24.3.5 地址指针

每个DMA通道包含有三个地址指针寄存器，分别是外设地址寄存器（PAR）、存储器地址寄存器0（MAR0）和存储器地址寄存器1（MAR1）；这些指针寄存器都采用双层结构，即包含active和shadow两组寄存器。软件对PAR、MAR的读是对active寄存器执行，写则同时对active和shadow寄存器执行。当DMA通道开始工作后，active寄存器可能会随着数据搬运进行实时增减，因此软件可以查询到当前DMA访问的地址，而shadow寄存器则保持软件写入值。在循环模式下，当TSIZE定义的传输长度完成后，active寄存器自动重载为shadow寄存器的值，实现指针回滚操作。

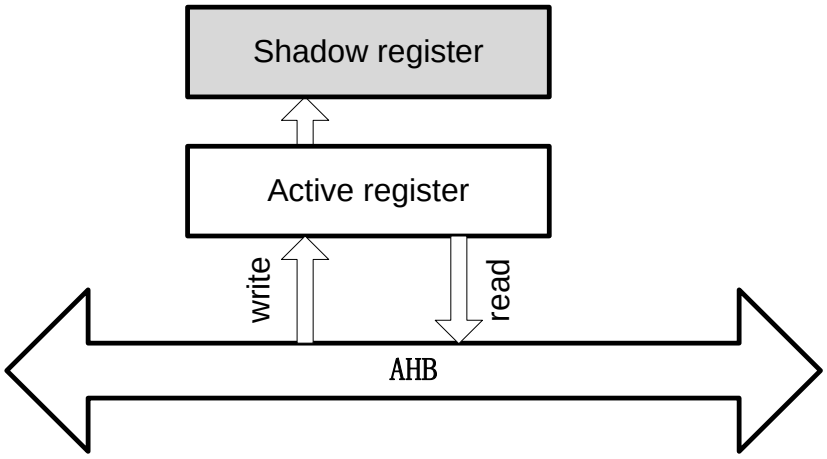


图 24-3 活动寄存器和影子寄存器

24.3.6 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。配置地址指针寄存器时，注意地址必须对齐访问带宽，比如在字访问配置下，地址指针必须对齐 32bit word，即最低 2bit 必须为 00。

24.3.7 传输长度

每个DMA通道的传输长度可以单独配置，最长为65536次传输。传输长度由CHxTSIZE寄存器定义。当传输开始后，每次传输后CHxTSIZE寄存器自动递减，直到递减到0。

TSIZE递减到一半时，半程中断标志置位；当TSIZE递减到0时，全程中断标志置位。

当TSIZE递减到0后，传输自动停止，通道使能寄存器自动清零。如果使能了循环模式，则TSIZE递减到0后会自动重载到最初的配置值，重新开始递减，通道使能寄存器保持置位。

24.3.8 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，而是返回寄存器定义的源地址和目的地址，继续传输，同时TSIZE自动回到最初的配置重新开始递减。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CHxCTRL.CIRC寄存器使能循环模式。

注意：存储DMA通道不支持循环模式。

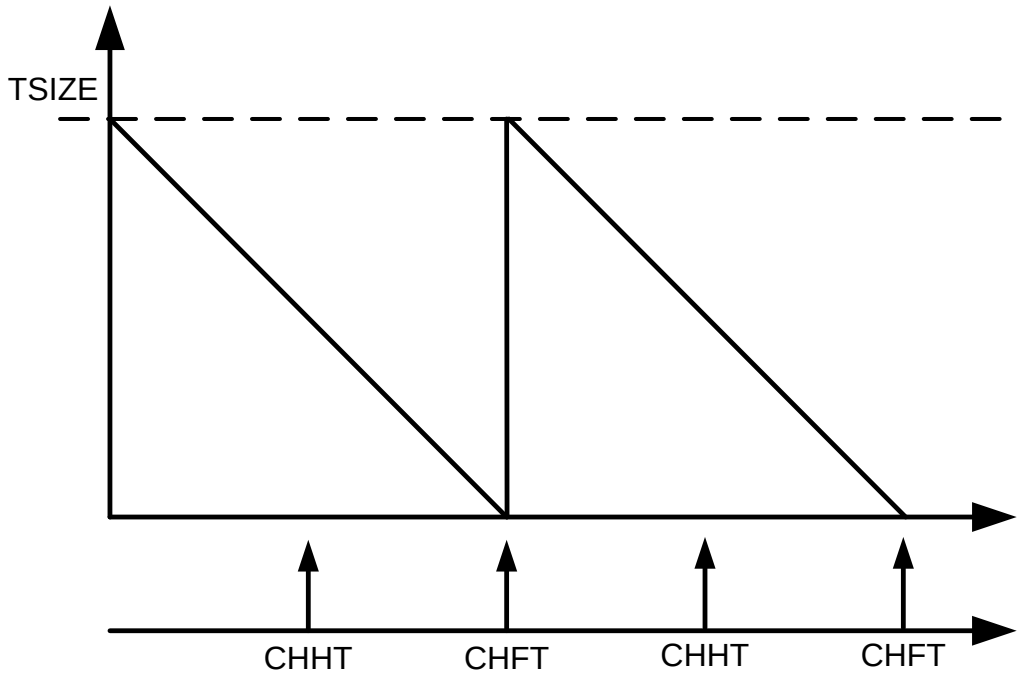


图 24-4 循环模式下的传输长度寄存器

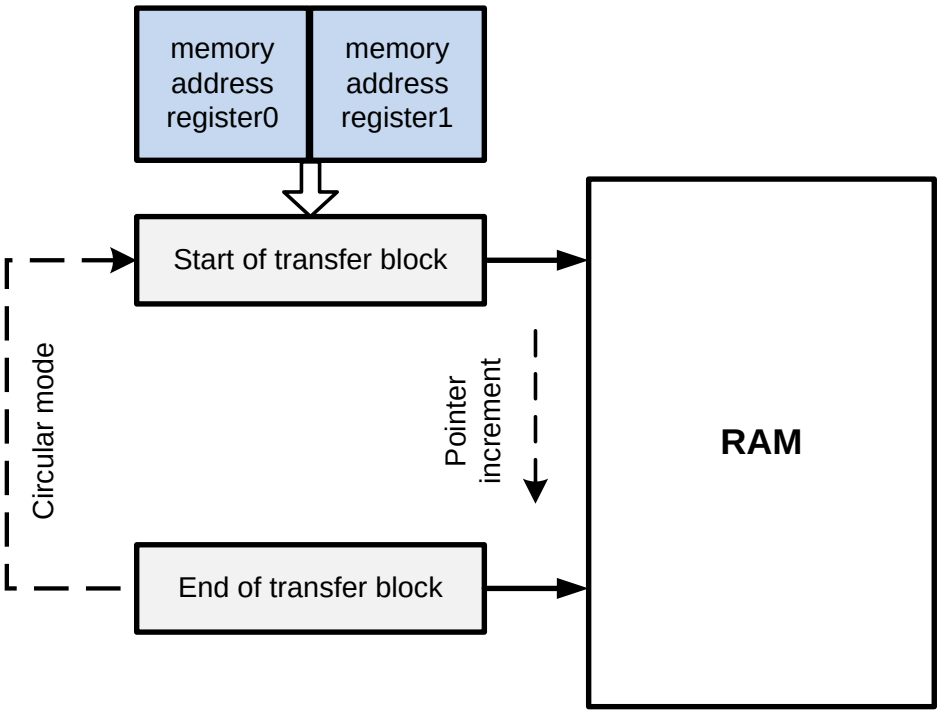


图 24-5 循环模式下的源/目的地址寄存器和存储器指针

正确停止循环模式，需要先关闭产生DMA请求的外设，然后再关闭DMA通道。

24.3.9 双缓冲区模式

外设通道支持double buffering，仅能在循环模式下使用。为了支持double buffering需要两组存储器目标地址寄存器，第一轮block transfer使用存储器地址寄存器0作为存储器起始地址，当第一轮block transfer结束后，下一轮block transfer自动切换到存储器地址寄存器1作为存储器起始地址，通过这种方式，循环使用两块存储器空间，实现乒乓操作，可以降低软件处理的实时性要求。

比如当DMA将外设数据搬运到RAM block0时，软件可以去处理RAM block1中保存的上次DMA传输的数据。

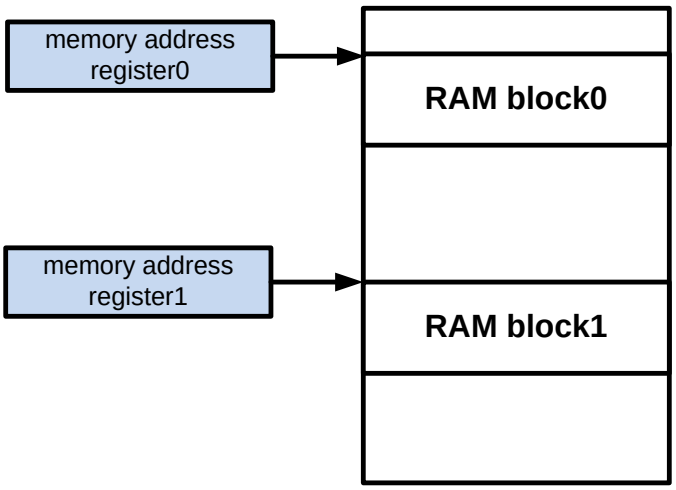


图 24-6 double buffering

同样的，也可以使用DMA将两块RAM中的数据搬运到外设，当DMA在搬运RAM block0的数据时，软件可以去填充RAM block1的数据，为下一次搬运做准备。

24.4 通道控制

24.4.1 DMA 请求映射

DMA 共有 4 个优先级可配的外设通道，每个通道均可独立选择所有外设请求，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	外设请求源
0	***SPI0_RX***
1	***SPI1_RX***
2	***SPI0_TX***
3	***SPI1_TX***
4	***UART0_RX***
5	***UART1_RX***
6	***UART2_RX***
7	***UART3_RX***
8	***UART4_RX***
9	***UART0_TX***
10	***UART1_TX***
11	***UART2_TX***
12	***UART3_TX***
13	***UART4_TX***
14	***LPUART0_RX***
15	***LPUART1_RX***
16	***LPUART0_TX***
17	***LPUART1_TX***
18	***I2C_SMB0_RX***
19	***I2C_SMB1_RX***
20	***I2C_SMB0_TX***
21	***I2C_SMB1_TX***
22	***AES_IN***
23	***AES_OUT***
24	***CRC***
25	***ATIM0_CH1***
26	***ATIM0_CH2***

27	***ATIM0_CH3***
28	***ATIM0_CH4***
29	***ATIM0_TRIG***
30	***ATIM0_COM***
31	***ATIM0_UEV***
32	***GPTIM0_CH1***
33	***GPTIM0_CH2***
34	***GPTIM0_CH3***
35	***GPTIM0_CH4***
36	***GPTIM0_TRIG***
37	***GPTIM0_UEV***
38	***GPTIM1_CH1***
39	***GPTIM1_CH2***
40	***GPTIM1_CH3***
41	***GPTIM1_CH4***
42	***GPTIM1_TRIG***
43	***GPTIM1_UEV***
44	***BSTIM0_TRIG***
45	***LPTIM0_TRIG***
46	***ADC_ACQ0***
47	***ADC_ACQ1***
48	***ADC_ACQ2***
49	***ADC_ACQ3***
50	***ADC_ACQ4***
51	**-**
52	**-**
53	**-**
54	**-**
55	**-**
56	**-**
57	**-**
58	**-**
59	**-**
60	***DAC***

61	***LED_PWMCR***
62	***LED_DOTDR***
63	***SWTRIG***

表 24-1 PDMA 通道映射列表

注意，外设请求和 DMA 传输的源/目的并没有必然联系，可以使用一个外设的中断或 trigger 信号来请求其他外设的 DMA 搬运，当然也可以由外设中断或 trigger 信号来请求本外设自己的数据搬运。

外设请求触发自身数据搬运

比如 ADC 某个 ACQ 通道，其转换完成中断可以连接到 DMA，触发 DMA 将自己的 result 寄存器数据搬运到 RAM。

外设请求触发其他外设数据搬运

比如以 ATIM0_CH1 为触发源，DMA 源地址选择为 RAM，目的地址为 UART0 TXBUF，则当触发信号到来后，DMA 将 RAM 中的数据搬运到 UART0 的发送 buffer，UART0 自动完成发送。

存储器通道只能由软件触发，当软件置位存储通道使能寄存器后，MDMA 就自动开始数据搬运，直到 TSIZE 定义的数据长度搬运结束。

24.4.2 Request Select

使用外设 DMA 前，每个通道通过 RSEL 寄存器，从 64 个请求源中选择 1 个作为本通道的传输触发信号。

24.5 寄存器

模块起始地址：0x4000_0400

offset 地址	名称	符号
0x00	通道 0 控制寄存器 （Channel 0 Control Register）	DMA_CH0CR
0x04	通道 0 配置寄存器 （Channel 0 Config Register）	DMA_CH0CFGR
0x08	通道 0 外设地址寄存器 （Channel 0 Peripheral Address Register）	DMA_CH0PAR
0x0C	通道 0 存储器地址寄存器 0 （Channel 0 Memory Address Register0）	DMA_CH0MAR0
0x10	通道 0 存储器地址寄存器 1 （Channel 0 Memory Address Register1）	DMA_CH0MAR1
0x20	通道 1 控制寄存器 （Channel 1 Control Register）	DMA_CH1CR
0x24	通道 1 配置寄存器 （Channel 1 Config Register）	DMA_CH1CFGR
0x28	通道 1 外设地址寄存器 （Channel 1 Peripheral Address Register）	DMA_CH1PAR
0x2C	通道 1 存储器地址寄存器 0 （Channel 1 Memory Address Register0）	DMA_CH1MAR0
0x30	通道 1 存储器地址寄存器 1 （Channel 1 Memory Address Register1）	DMA_CH1MAR1
0x40	通道 2 控制寄存器 （Channel 2 Control Register）	DMA_CH2CR
0x44	通道 2 配置寄存器 （Channel 2 Config Register）	DMA_CH2CFGR
0x48	通道 2 外设地址寄存器 （Channel 2 Peripheral Address Register）	DMA_CH2PAR
0x4C	通道 2 存储器地址寄存器 0 （Channel 2 Memory Address Register0）	DMA_CH2MAR0
0x50	通道 2 存储器地址寄存器 1 （Channel 2 Memory Address Register1）	DMA_CH2MAR1
0x60	通道 3 控制寄存器 （Channel 3 Control Register）	DMA_CH3CR
0x64	通道 3 配置寄存器 （Channel 3 Config Register）	DMA_CH3CFGR
0x68	通道 3 外设地址寄存器 （Channel 3 Peripheral Address Register）	DMA_CH3PAR
0x6C	通道 3 存储器地址寄存器 0 （Channel 3 Memory Address Register0）	DMA_CH3MAR0
0x70	通道 3 存储器地址寄存器 1 （Channel 3 Memory Address Register1）	DMA_CH3MAR1
0xE0	通道 7 控制寄存器 （Channel 7 Control Register）	DMA_CH7CR
0xE4	通道 7 Flash 指针寄存器 （Channel 7 Flash Address Register）	DMA_CH7FAR

offset 地址	名称	符号
0xE8	通道 7 RAM 指针寄存器 （Channel 7 RAM Address Register）	DMA_CH7RAR
0x200	DMA 全局控制寄存器 （DMA Global Control Register）	DMA_GCR
0x204	DMA 软件请求寄存器 （DMA Software Request Register）	DMA_SWRR
0x300	DMA 中断状态标志寄存器 （DMA Interrupt Status Register）	DMA_ISR

24.5.1 DMA 全局控制寄存器（DMA_GCR）

名称	DMA_GCR							
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	-						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 访问地址错误中断使能
0	**EN**	DMA 全局使能 (DMA enable) 1：DMA 使能 0：DMA 关闭

24.5.2 DMA 软件请求寄存器寄存器（DMA_SWRR）

名称	DMA_SWRR							
Offset	0x204							
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

名称	DMA_SWRR							
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SWTRIG[6:0]			
位权限	U-0				R/W-0			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	**SWTRIG**	外设通道软件触发请求，软件写 1 触发一次 DMA 请求，DMA 完成配置长度的搬运后将此请求硬件清零。使用软件触发功能前需要将通道请求映射配置为软件请求，否则软件触发请求将一直 pending SWTRIG[3] – Channel 3 SWTRIG[2] – Channel 2 … SWTRIG[0] – Channel 0

24.5.3 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
Offset								
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	U-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
31:18	-	RFU：未实现，读为 0
17	ADDRERR	DMA 访问地址错误
16	-	RFU：未实现，读为 0
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
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

24.5.4 通道 x 控制寄存器（DMA_CHxCR）

名称	DMA_CHxCR(x=0~3)							
Offset								
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	MINC		PINC		DBUF	-		EN
位权限	R/W-00		R/W-00		R/W-0	U-0		R/W-0

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:6	**MINC**	存储器指针（MAR）递增模式 00：存储器地址保持不变 01：存储器地址递增 10：存储器地址递减 11：RFU
5:4	**PINC**	外设指针（PAR）递增模式 00：外设地址保持不变 01：外设地址递增 10：外设地址递减 11：RFU
3	**DBUF**	双缓冲区模式使能（Double Buffering），仅在循环模式下有效 1：使能双缓冲区模式 0：禁止双缓冲区模式
2:1	--	RFU：未实现，读为 0
0	**EN**	Channelx 使能（Channelx Enable） 1：启动通道 0 0：关闭通道 0

24.5.5 通道 x 配置寄存器（DMA_CHxCFGR）

名称	DMA_CHxCFGR(x=0~3)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TSIZE[15:8]							
位权限	R/W-0000 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	SUSP	-	RSEL					

名称	DMA_CHxCFGR(x=0~3)							
位权限	R/W-0	U-0	R/W-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRI		BDW		CIRC	FTIE	HTIE	DIR
位权限	R/W-00		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	**TSIZE**	Channelx 传输长度，最大 65536 次传输，实际传输长度是 TSIZE+1 *注意：禁止将 TSIZE 配置为 0*
15	**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
14	--	RFU：未实现，读为 0
13:8	**RSEL**	Channelx 外设请求映射 (Channelx request Source Select) 每个通道可以接受 128 个外设请求，外设请求的映射参见通道控制章节
7:6	**PRI**	Channelx 优先级 (Channels Priority) 00：Low 01：Medium 10：High 11：Very High
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	**DIR**	通道传输方向 (Direction) 0：外设地址寄存器（PAR）为源指针，存储地址寄存器（MAR）为目的地指针 1：外设地址寄存器（PAR）为目的地指针，存储地址寄存器（MAR）为源指针

24.5.6 通道 x 外设地址寄存器（DMA_CHxPAR）

名称	DMA_CHxPAR (x=0~3)
Offset	

名称	DMA_CHxPAR (x=0~3)							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PAD[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PAD[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PAD[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**PAD**	Channelx 外设指针地址。(Channel x Peripheral Address Pointer) 此寄存器数值会随着指针移动递增或递减。在循环模式下，当 TSIZE 长度的传输完成后，此寄存器会重载为软件配置值。 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。

24.5.7 通道 x 存储器地址寄存器 0（DMA_CHxMAR0）

名称	DMA_CHxMAR0 (x=0~3)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MAD0[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MAD0[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MAD0[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MAD0[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**MAD0**	Channelx 存储器指针地址 0。(Channel x Memory Address pointer)

位号	助记符	功能描述
		此寄存器数值会随着指针移动递增或递减。在循环模式下，当TSIZE 长度的传输完成后，此寄存器会重载为软件配置值。 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。

24.5.8 通道 x 存储器地址寄存器 1（DMA_CHxMAR1）

名称	DMA_CHxMAR1 (x=0~3)							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MAD1[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MAD1[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MAD1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MAD1[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	**MAD1**	Channelx 存储器指针地址 1。(Channel x Memory Address pointer)此寄存器用于双缓冲区模式。 此寄存器数值会随着指针移动递增或递减。在循环模式下，当TSIZE 长度的传输完成后，此寄存器会重载为软件配置值。 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。

24.5.9 通道 7 控制寄存器（DMA_CH7CR）

名称	DMA_CH7CR							
Offset	0XE0							
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	RINC	FINC
位权限	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0

名称	DMA_CH7CR							
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	**RINC**	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	**FINC**	Channel7 Flash 地址增减设置，仅在 Flash->RAM 传输中有效 (Channel 7 Flash Incremental) 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 0 0：关闭通道 0

24.5.10 通道 7 Flash 指针寄存器（DMA_CH7FAR）

名称	DMA_CH7FAR							
Offset	0XE4							
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_CH7FAR							
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	FLSAD[14:8]						
位权限	U-0	R/W-000 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
31:15	-	RFU：未实现，读为 0
14:0	**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

24.5.11 通道 7 RAM 指针寄存器（DMA_CH7RAR）

名称	DMA_CH7RAR							
Offset	0XE8							
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[11:8]			
位权限	U-0				R/W-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
31:12	-	RFU：未实现，读为 0
11:0	**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

25 循环冗余校验（CRC）

25.1 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，FM33LF0xxA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

25.2 软件配置过程

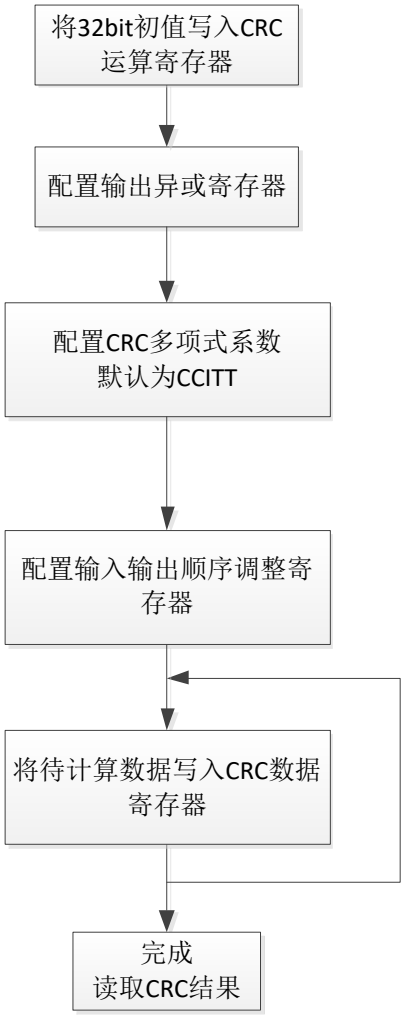


图 25-1CRC 运算流程图

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REFLECTIN处理使能；输出REFLECTOUT和XOROUT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。

25.3 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

25.4 DMA 接口

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。

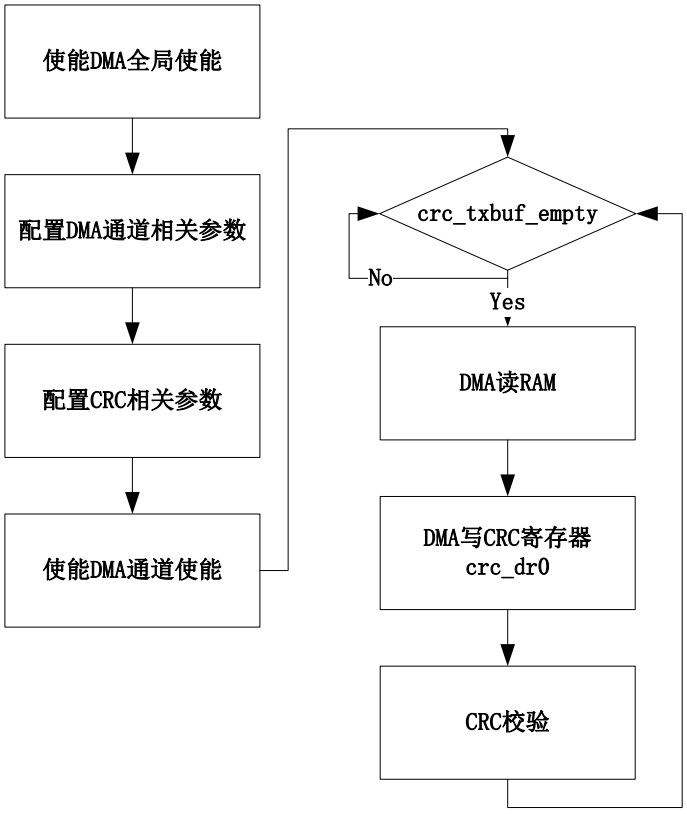


图 25-2 使用 DMA 对 RAM 中的数据进行 CRC 运算

25.5 Flash 数据完整性校验

为提升系统主频，降低critical path delay，删除这个局部总线功能。

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。

25.6 寄存器

模块起始地址：0x4001_0000

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

25.6.1 CRC 数据寄存器（CRC_DR）

名称	CRC_DR							
Offset	0x00000000							
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]。

25.6.2 CRC 控制状态寄存器（CRC_CR）

名称	CRC_CR							
Offset	0x00000004							
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	PARALLEL
位权限	U-0						R/W-0	R/W-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	**PARALLEL**	CRC 快速算法使能 0：CRC 快速算法使能关闭（1 个时钟处理 1bit 数据） 1：CRC 快速算法使能打开（1 个时钟处理 8bit 数据）
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

位号	助记符	功能描述
		10：8 位 11：7 位

25.6.3 CRC LFSR 寄存器（CRC_LFSR）

名称	CRC_LFSR							
Offset	0x00000008							
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

25.6.4 CRC 输出异或寄存器（CRC_XOR）

名称	CRC_XOR							
Offset	0x0000000C							
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。

25.6.5 CRC 多项式寄存器（CRC_POLY）

名称	CRC_POLY							
Offset	0x0000001C							
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）

26 高级定时器（ATIM）

26.1 概述

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。

26.2 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

26.3 结构框图

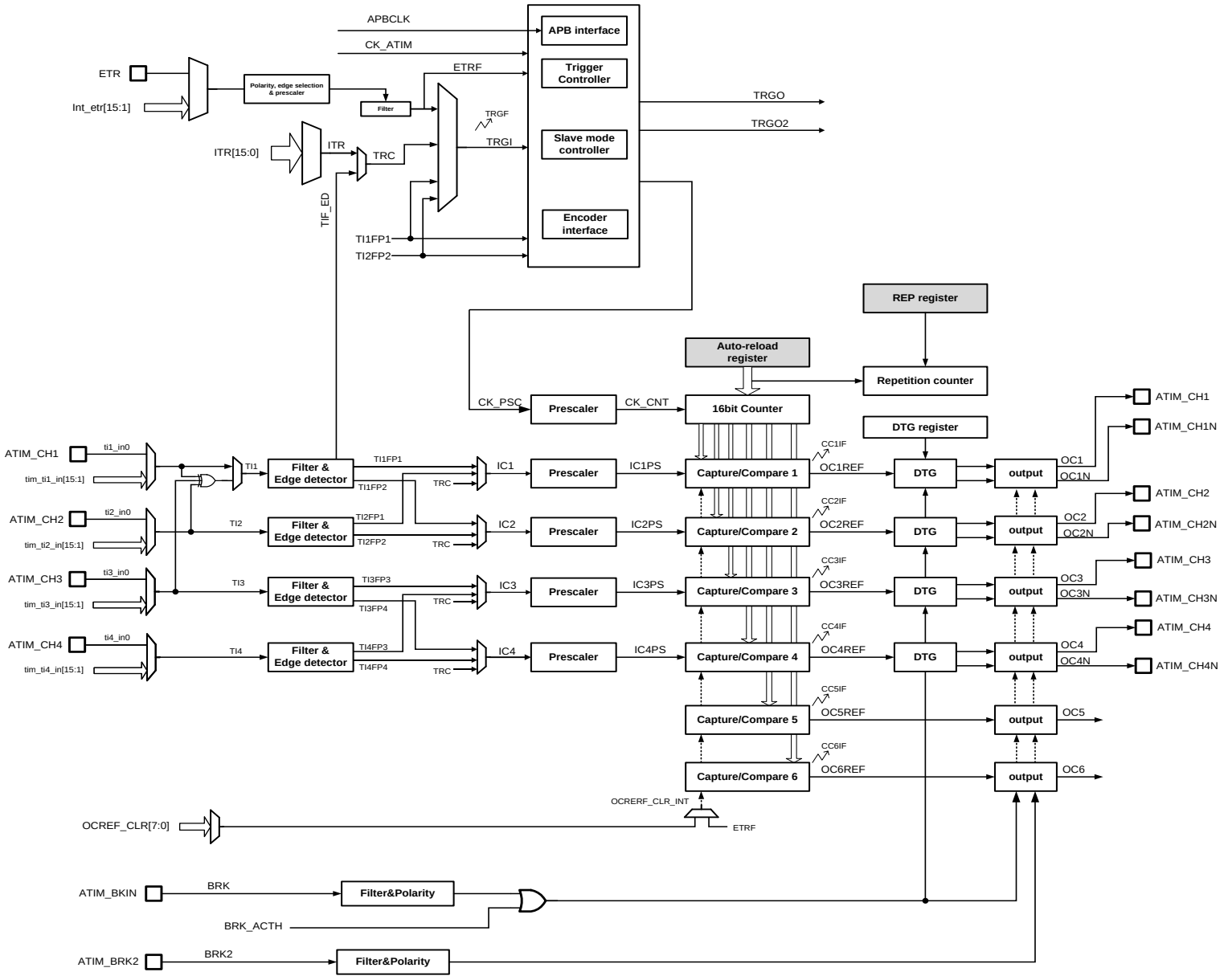
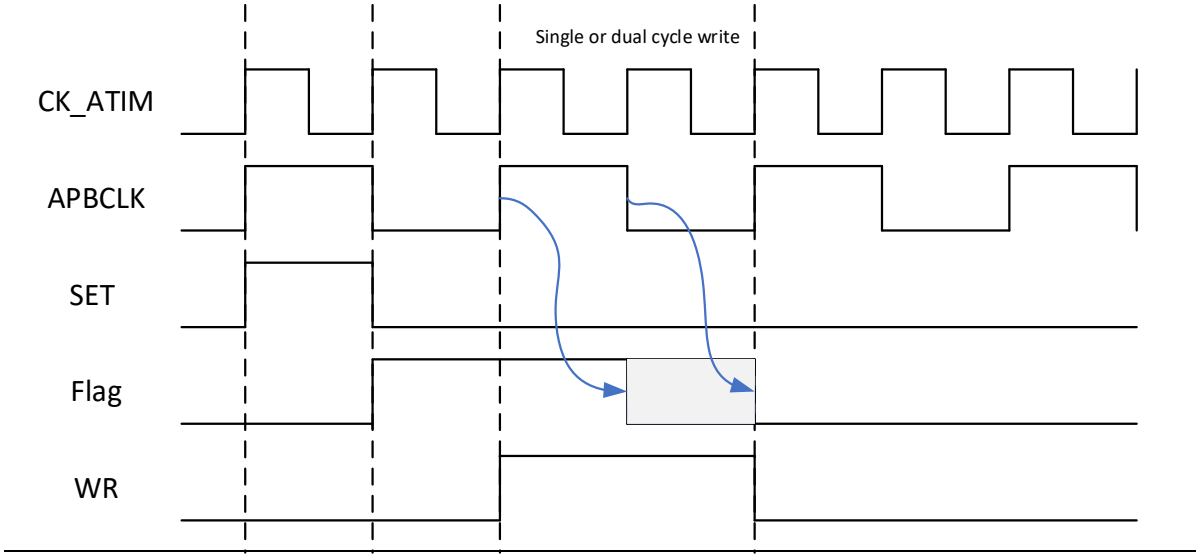


图 26-1 高级定时器结构框图

26.4 工作时钟

ATIM是全同步设计，包含两个时钟输入：APBCLK用于总线寄存器访问，CK_ATIM用于定时器工作。CK_ATIM=APBCLK。



26.5 管脚和内部信号连接

下表罗列了ATIM相关的输入输出引脚。

Pin name	type	Description
ATIM_CH1 ATIM_CH2 ATIM_CH3 ATIM_CH4	Input/output	定时器输入输出通道，可用于输入捕捉、PWM 输出 CH1 和 CH2 也可以用作外部计数时钟输入（频率不能高于 CK_ATIM 的 1/4）、外部触发、位置编码器输入 CH1、CH2、CH3 可以用于连接霍尔传感器
ATIM_CH1N ATIM_CH2N ATIM_CH3N ATIM_CH4N	Input/output	定时器互补输出通道，由 CHx 输出信号产生，能够产生死区插入
ATIM_ETR	Input	外部触发输入，可用于触发信号或者外部计数时钟 在使能了输入预分频的情况下，可以输入高于 CK_ATIM 频率的计数时钟
ATIM_BKIN ATIM_BKIN2	Input	两路刹车信号输入

表 26-1 外部引脚连接

ATIM与芯片内部信号连接：

signal name	type	Description
ITR[15:0]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_ATIM 的 1/4）
Int_etr[15:1]	input	外部触发输入，可用于触发信号或者外部计数时钟 在使能了输入预分频的情况下，可以输入高于 CK_ATIM 频率的计数时钟

TRGO TRGO2	output	内部触发信号输出，用于连接到其他 timer 或外设，比如 ADC、DAC
OCREF_CLR[7:0]	input	OCREF 清零信号输入，用于片内特殊信号异步清零 OCREF
APBCLK	input	寄存器总线时钟
CK_ATIM	Input	ATIM 工作时钟，与 APBCLK 同频
ATIM_CC1_DMA ATIM_CC2_DMA ATIM_CC3_DMA ATIM_CC4_DMA	output	4 个通道各自的 DMA 请求输出
ATIM_UEV_DMA	output	更新事件 DMA 请求
ATIM_TRG_DMA	output	触发信号 DMA 请求
ATIM_COM_DMA	output	COM 事件 DMA 请求

表 26-2 内部信号连接

ATIM的ITR[15:0]与片内信号连接：

ITRx	ATIM1
ITR0	-
ITR1	-
ITR2	-
ITR3	-
ITR4	GPTIM0_TRGO
ITR5	GPTIM1_TRGO
ITR6	-
ITR7	BSTIM16_TRGO
ITR8	LPTIM16_TRGO
ITR9	-
ITR10	-
ITR11	-
ITR12	-
ITR13	-
ITR14	-
ITR15	-

表 26-3ITR 信号连接

ATIM的ETR[15:0]与片内信号连接：

ETRx	ATIM1
ETR0	ATIM1_ETR
ETR1	Comp1_out
ETR2	Comp2_out
ETR3	Comp3_out
ETR4	-
ETR5	-
ETR6	-
ETR7	-
ETR8	-

ETR9	-
ETR10	-
ETR11	-
ETR12	-
ETR13	-
ETR14	-
ETR15	-

表 26-4ETR 信号连接

ATIM的TI1_IN[15:0]与片内信号连接：

TI1_INx	ATIM1
TI1_IN 0	ATIM1_CH1
TI1_IN 1	-
TI1_IN 2	-
TI1_IN 3	-
TI1_IN 4	-
TI1_IN 5	-
TI1_IN 6	-
TI1_IN 7	-
TI1_IN 8	-
TI1_IN 9	-
TI1_IN 10	-
TI1_IN 11	-
TI1_IN 12	-
TI1_IN 13	-
TI1_IN 14	-
TI1_IN 15	-

表 26-5 TI1_IN 信号连接

ATIM的TI2_IN[15:0]与片内信号连接：

TI2_INx	ATIM1
TI2_IN 0	ATIM1_CH2
TI2_IN 1	-
TI2_IN 2	-
TI2_IN 3	-
TI2_IN 4	-
TI2_IN 5	-
TI2_IN 6	-
TI2_IN 7	-
TI2_IN 8	-
TI2_IN 9	-
TI2_IN 10	-
TI2_IN 11	-
TI2_IN 12	-
TI2_IN 13	-
TI2_IN 14	-
TI2_IN 15	-

表 26-6 TI2_IN 信号连接

ATIM的TI3_IN[15:0]与片内信号连接：

TI3_INx	ATIM1
TI3_IN 0	ATIM1_CH3
TI3_IN 1	-
TI3_IN 2	-
TI3_IN 3	-
TI3_IN 4	-
TI3_IN 5	-
TI3_IN 6	-
TI3_IN 7	-
TI3_IN 8	-
TI3_IN 9	-
TI3_IN 10	-
TI3_IN 11	-
TI3_IN 12	-
TI3_IN 13	-
TI3_IN 14	-
TI3_IN 15	-

表 26-7 TI3_IN 信号连接

ATIM的TI4_IN[15:0]与片内信号连接：

TI4_INx	ATIM1
TI4_IN 0	ATIM1_CH4
TI4_IN 1	-
TI4_IN 2	-
TI4_IN 3	-
TI4_IN 4	-
TI4_IN 5	-
TI4_IN 6	-
TI4_IN 7	-
TI4_IN 8	-
TI4_IN 9	-
TI4_IN 10	-
TI4_IN 11	-
TI4_IN 12	-
TI4_IN 13	-
TI4_IN 14	-
TI4_IN 15	-

表 26-8 TI4_IN 信号连接

ATIM的OCREF_CLR与内部信号连接：

OCREF_CLRx	ATIM1
OCREF_CLR0	COMP0_OUT

OCREF_CLR1	COMP1_OUT
OCREF_CLR2	COMP2_OUT
OCREF_CLR3	-
OCREF_CLR4	-
OCREF_CLR5	-
OCREF_CLR6	-
OCREF_CLR7	-

表 26-9OCREF_CLR 信号连接

26.6 功能描述

26.6.1 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。

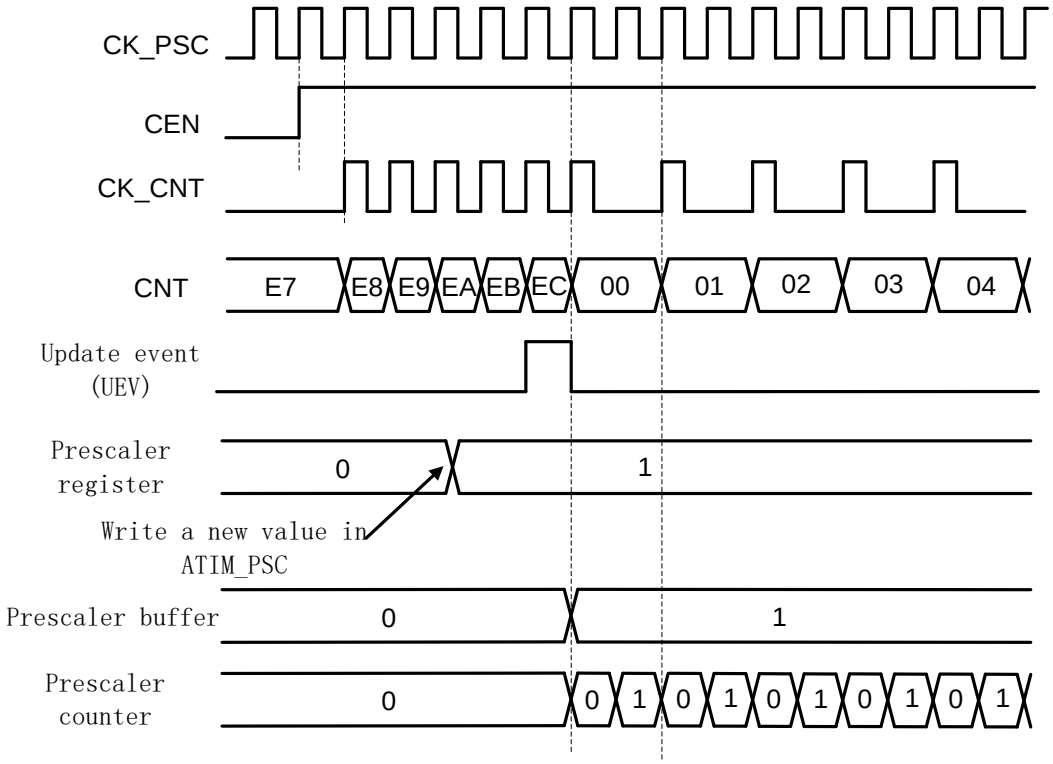


图 26-2 预分频从 1 变为 2 的波形

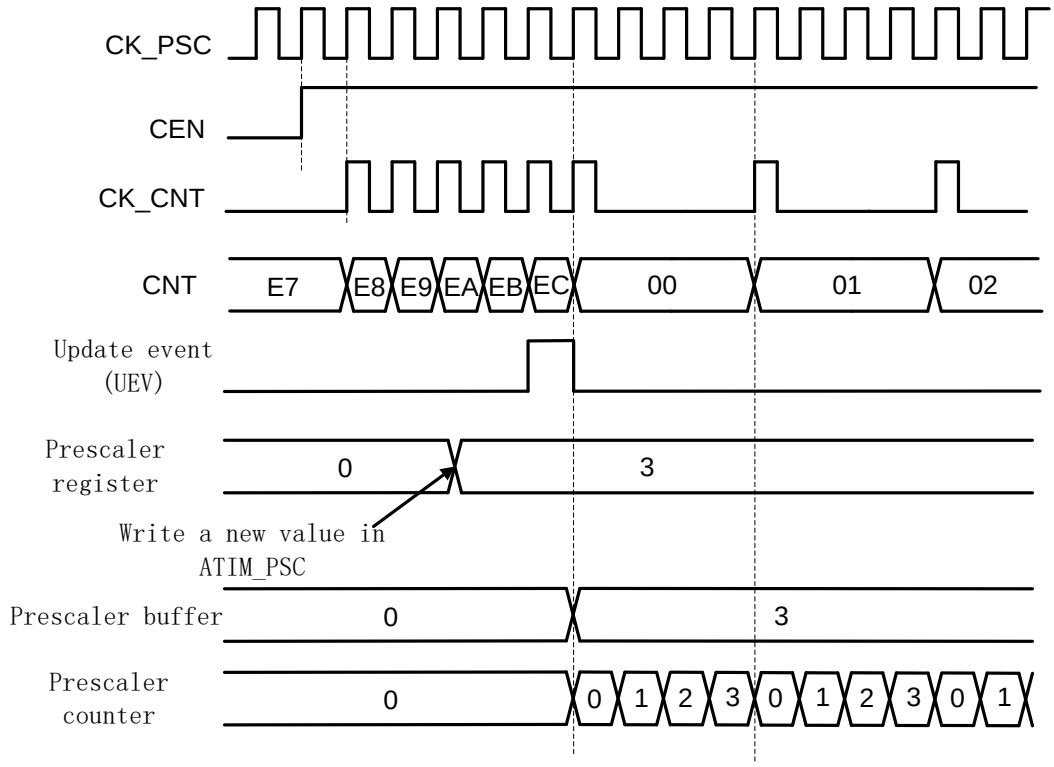


图 26-3 预分频从 1 变为 4 的波形

26.6.2 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

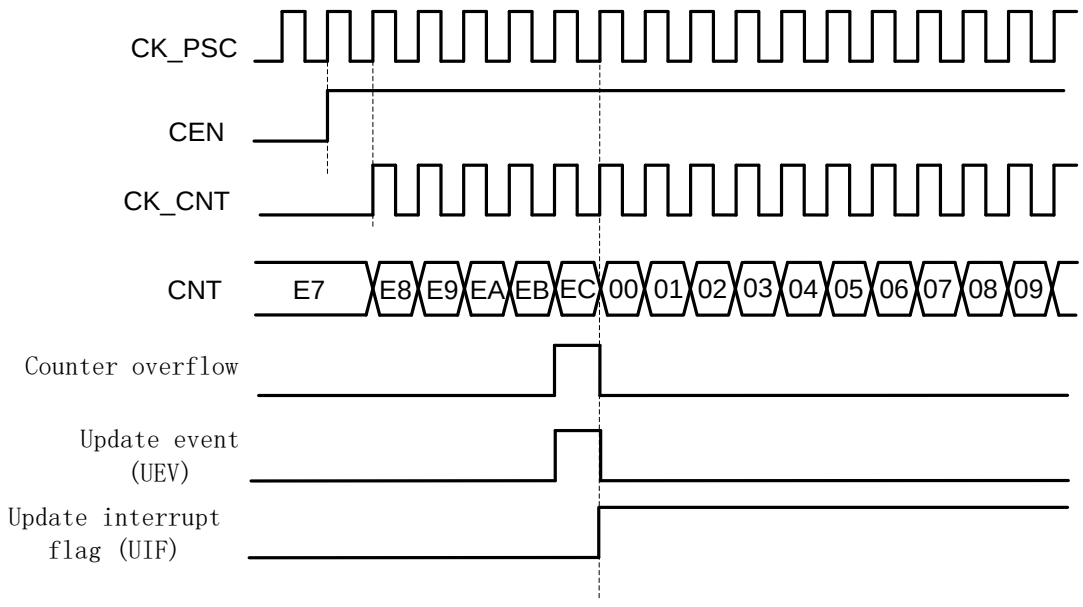


图 26-4 向上计数波形，内部时钟不分频

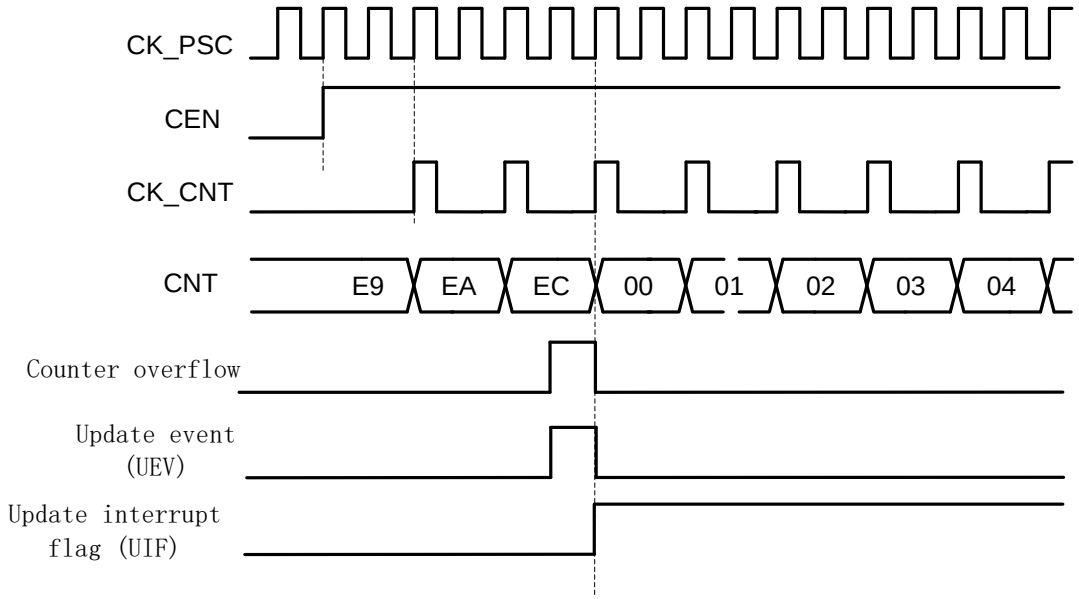


图 26-5 向上计数波形，内部时钟 2 分频

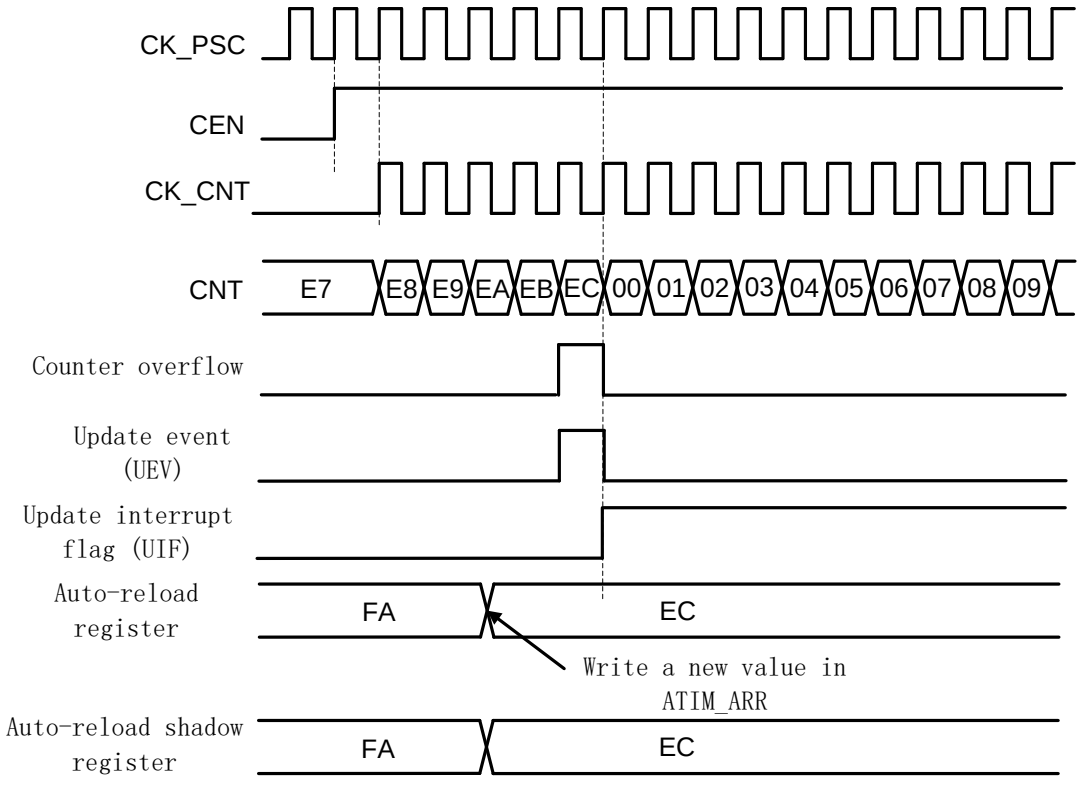


图 26-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

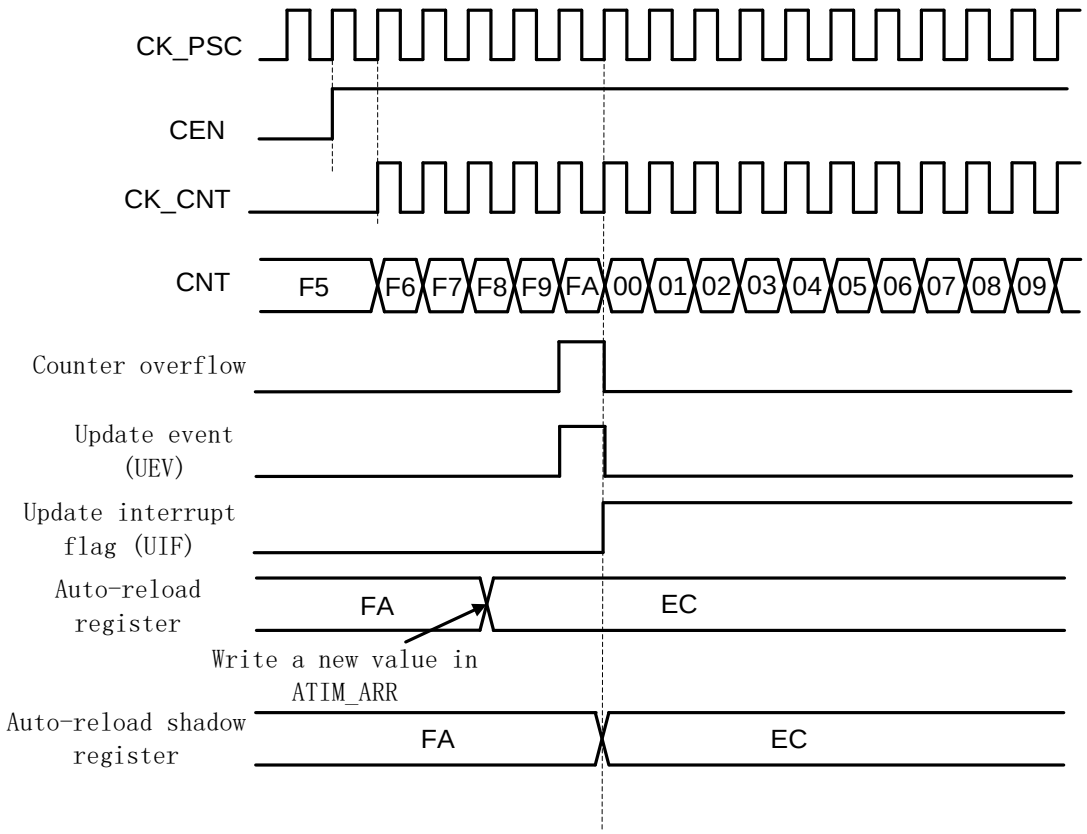


图 26-7ARPE=1（ATIM_ARR 预装载）时的更新事件

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

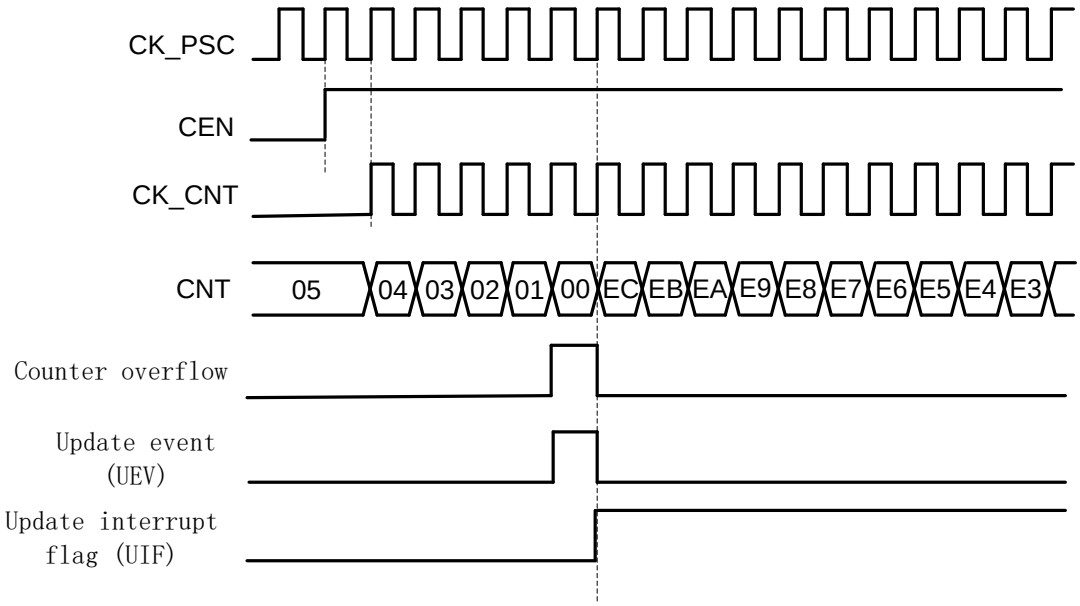


图 26-8 向下计数，内部时钟不分频

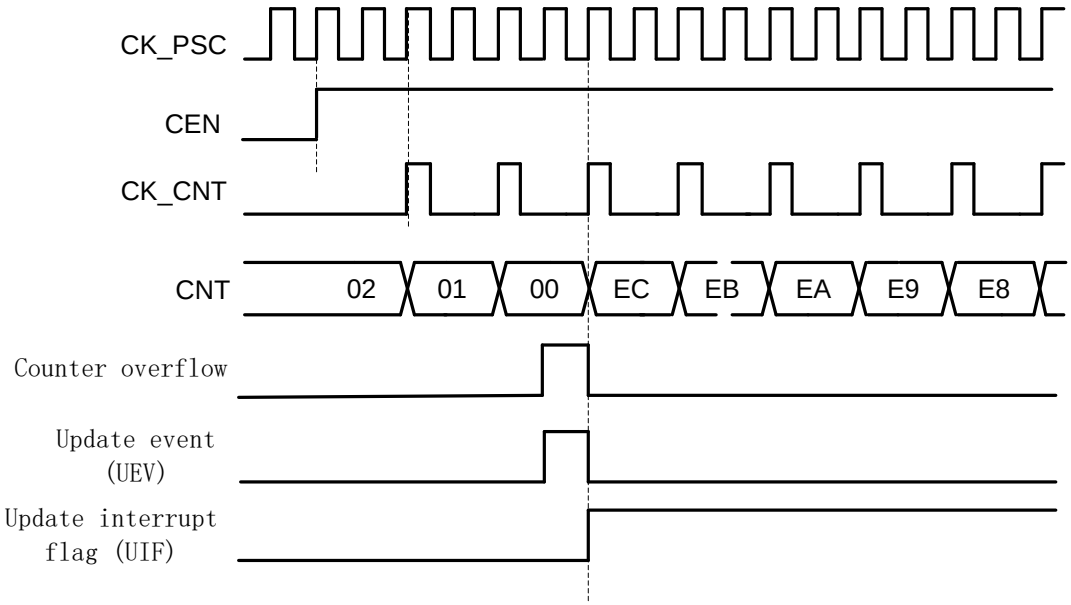


图 26-9 向下计数，内部时钟 2 分频

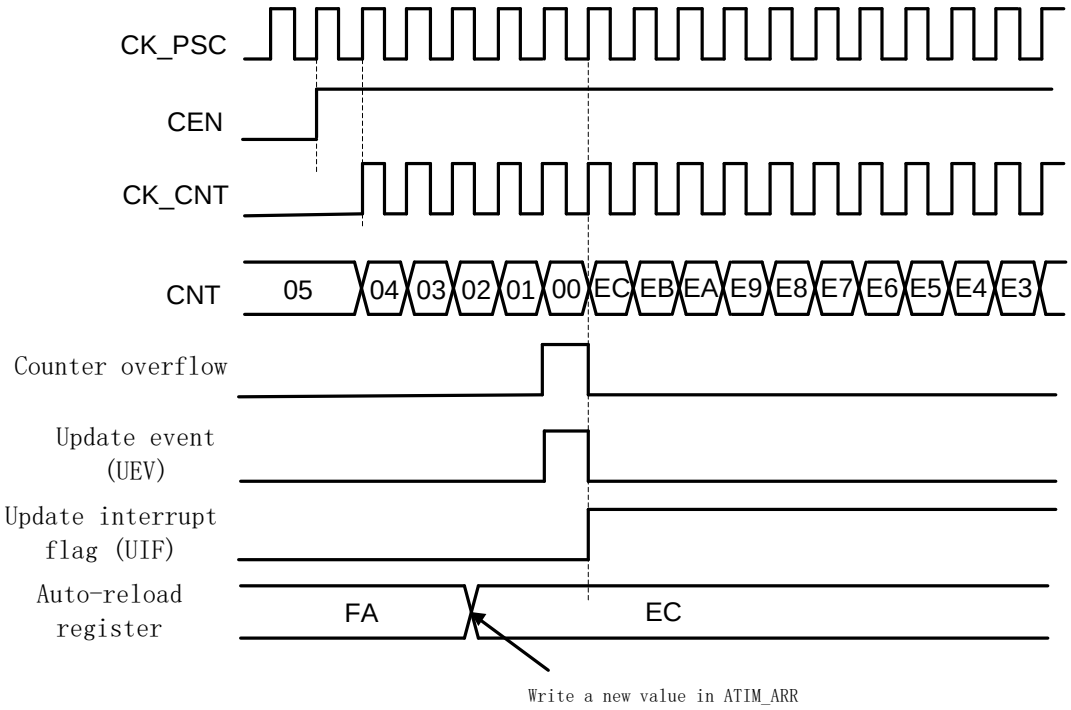


图 26-10 向下计数，内部时钟 2 分频

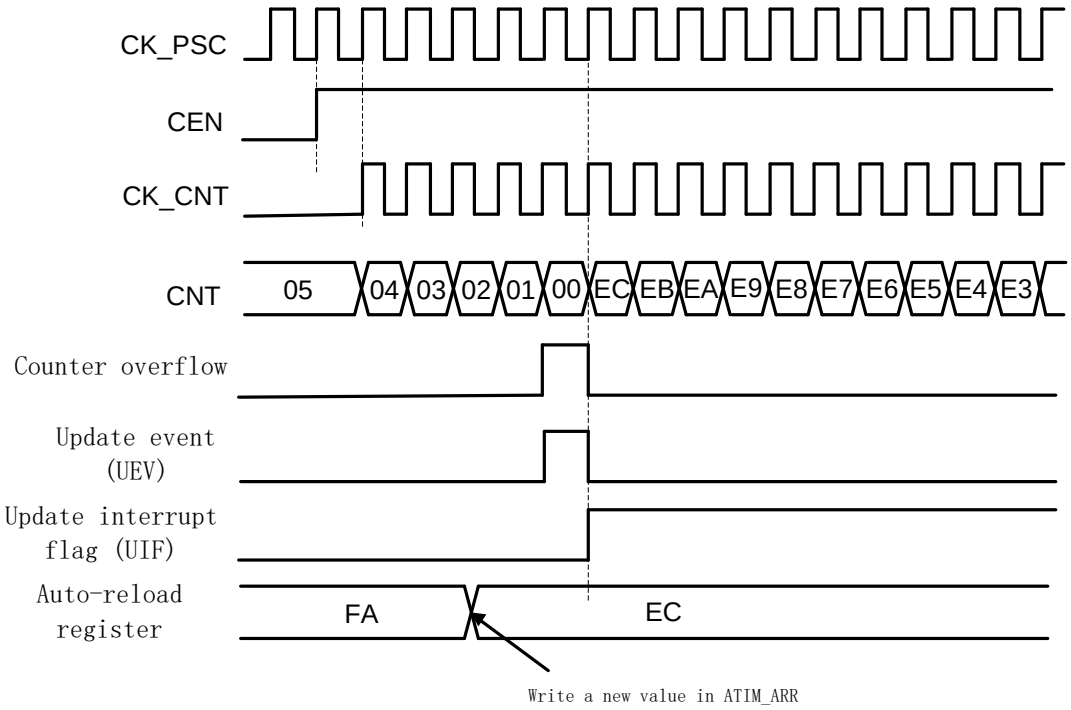


图 26-11 向下计数，不使用重复计数时的更新事件

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。

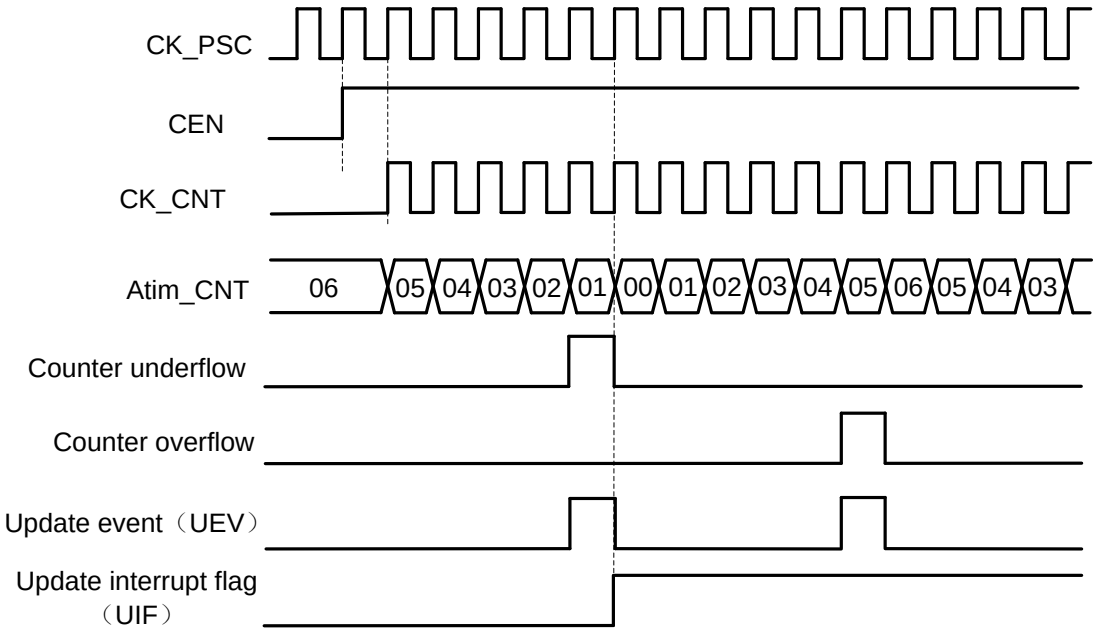


图 26-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

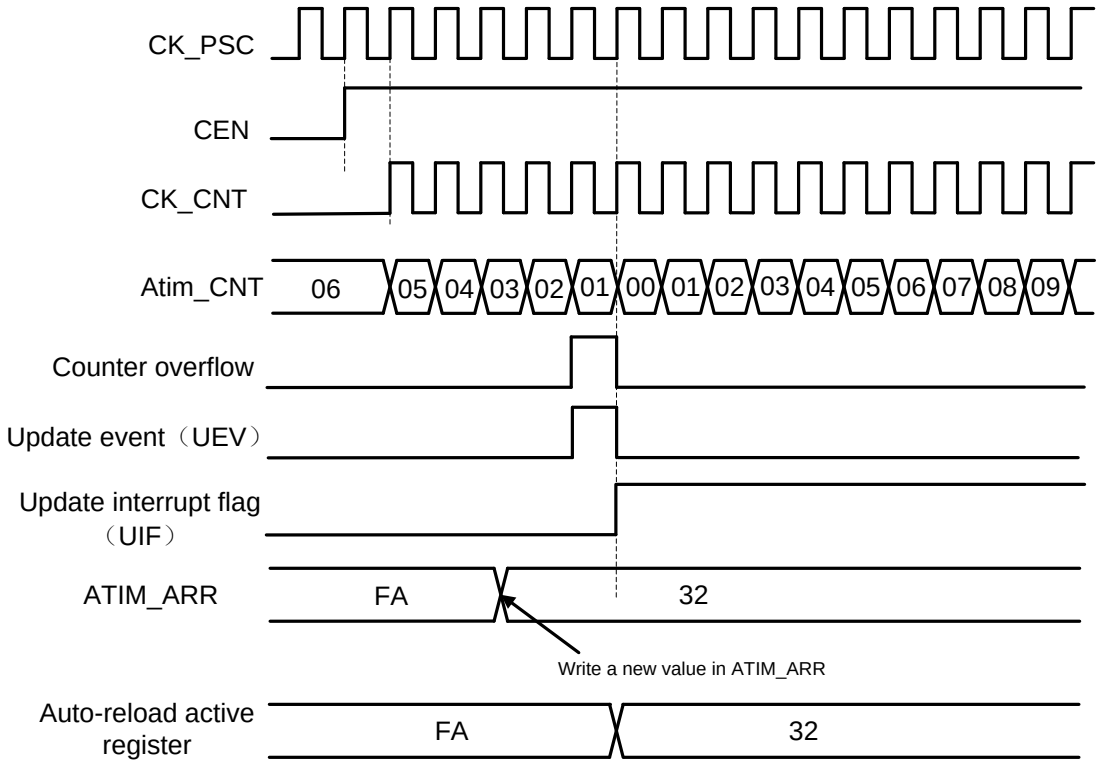


图 26-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

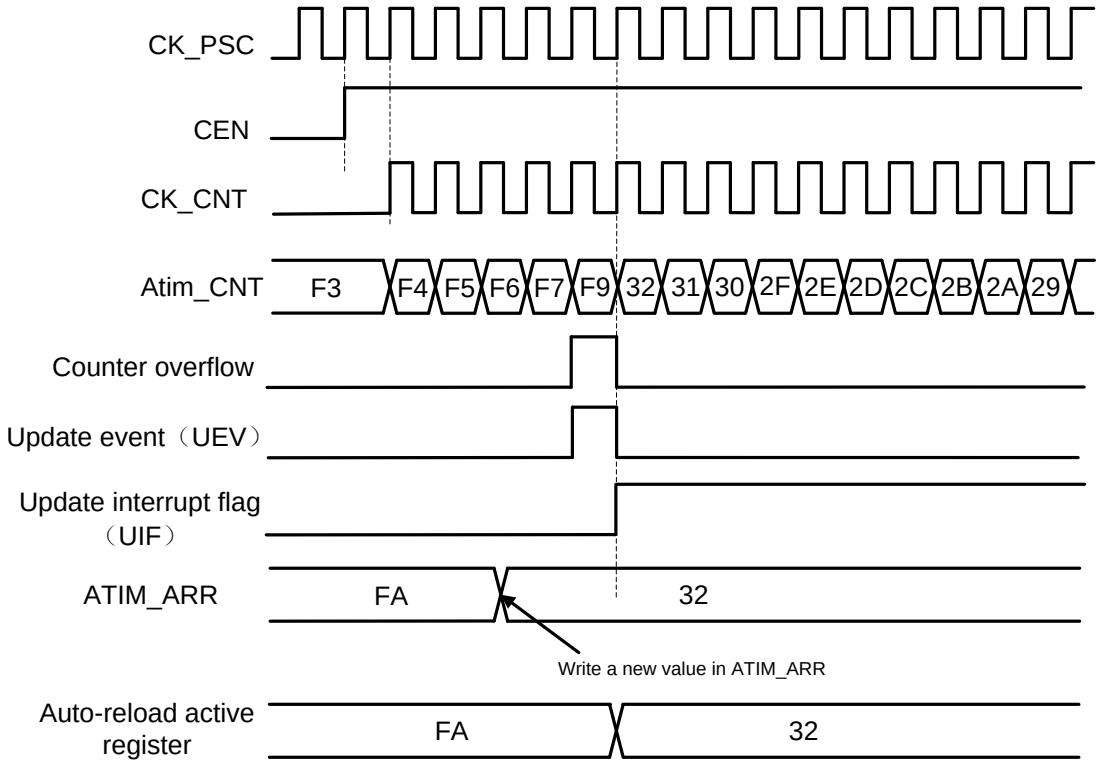


图 26-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

26.6.3 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。

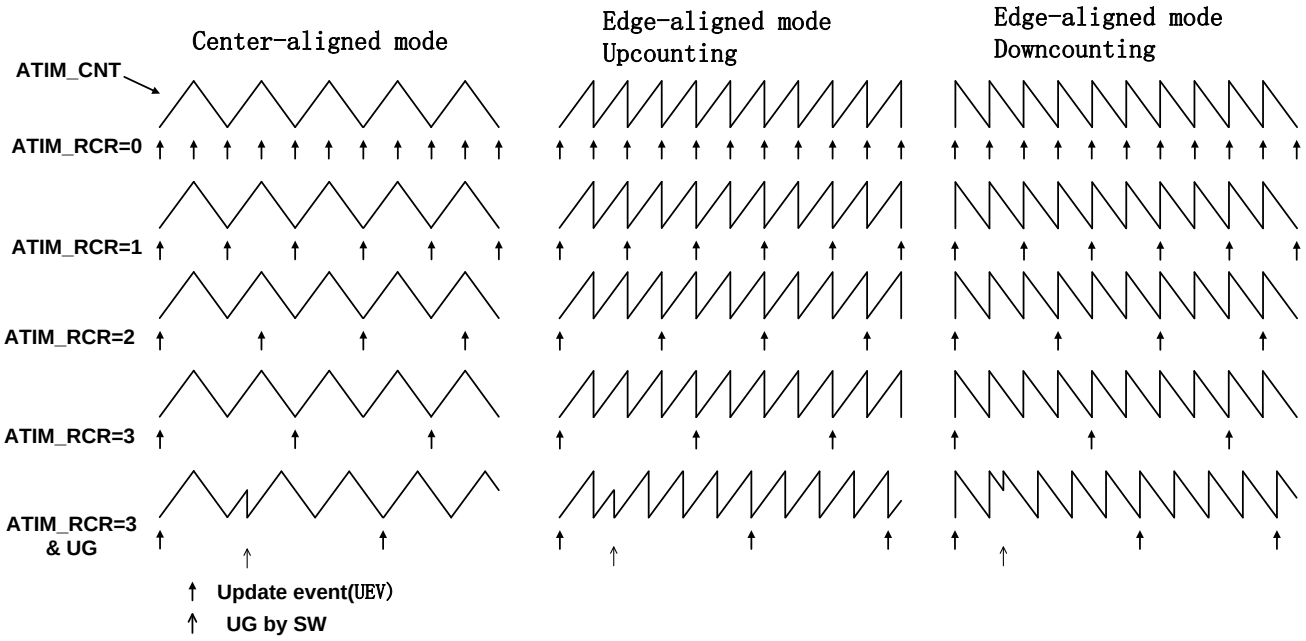


图 26-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

26.6.4 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

26.6.5 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

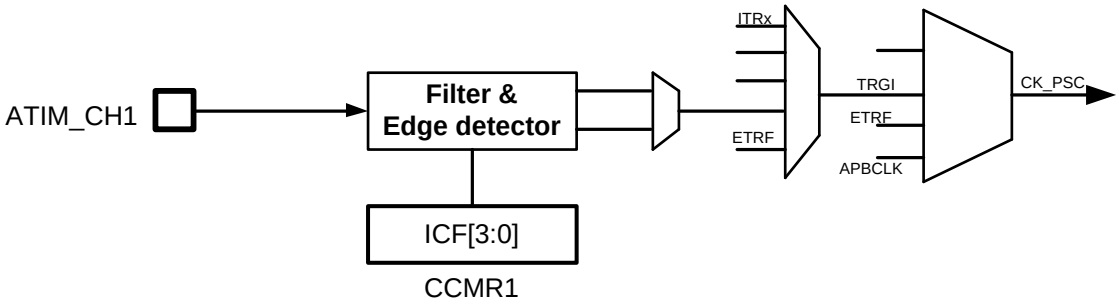


图 26-16 ATIM 时钟源框图

26.6.5.1 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。

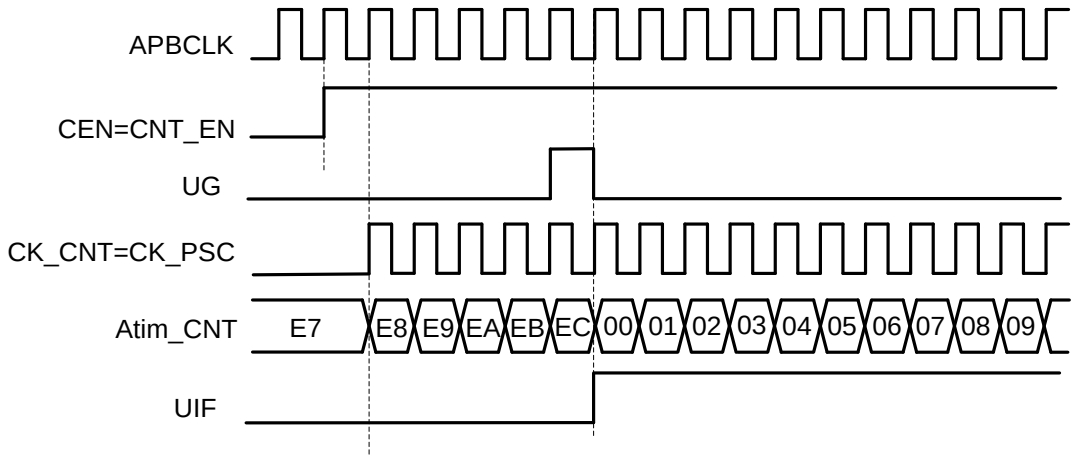


图 26-17 内部时钟源模式，时钟分频因子为 1

26.6.5.2 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。

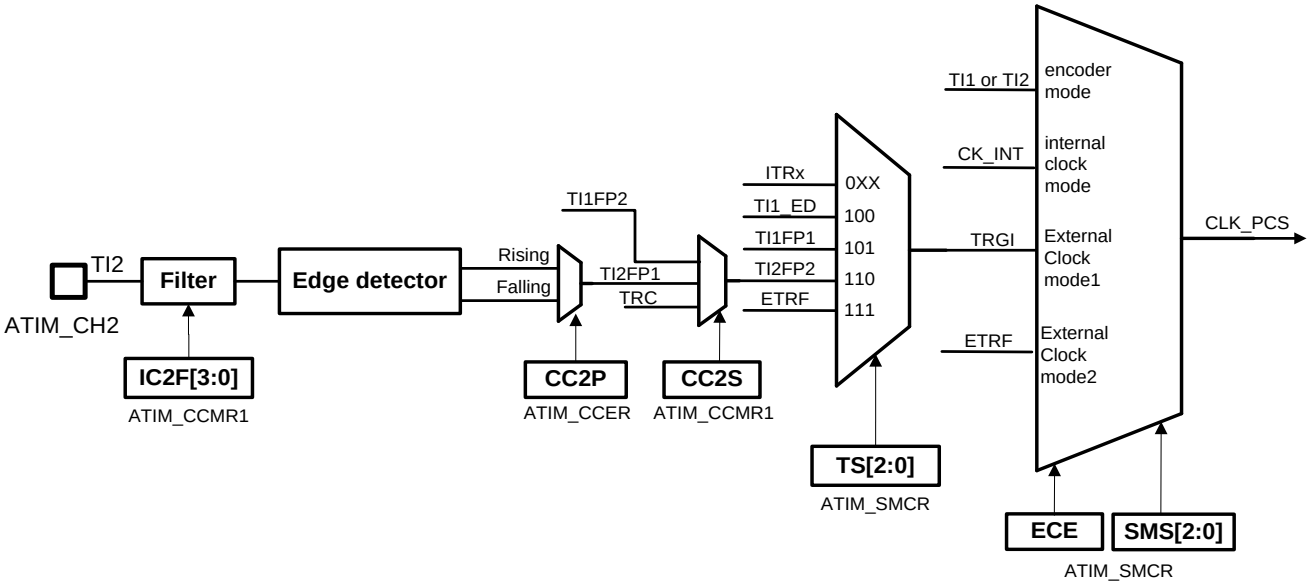


图 26-18TI2 外部时钟连接例子

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

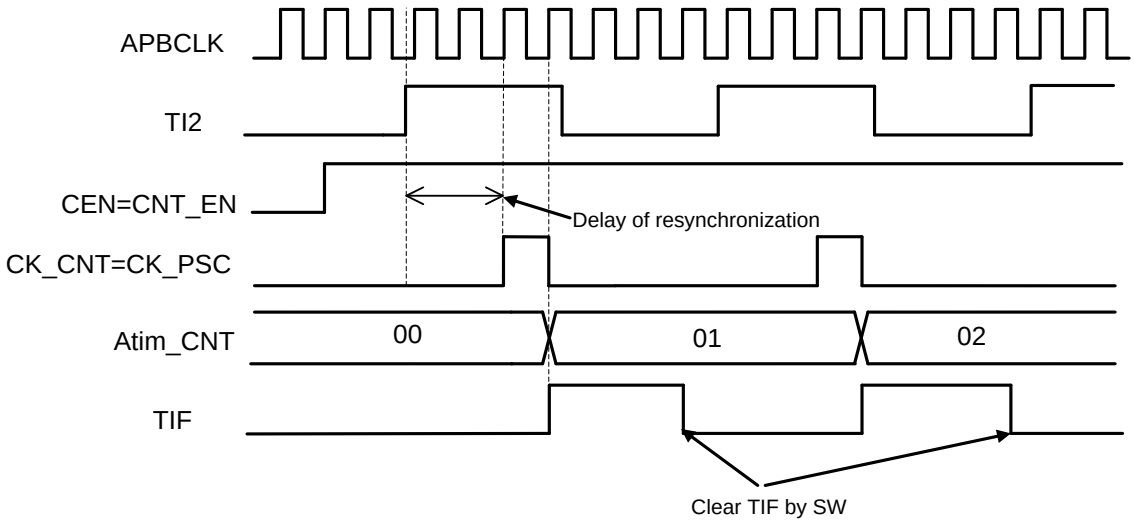


图 26-19 外部时钟模式 1 下的时序

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：

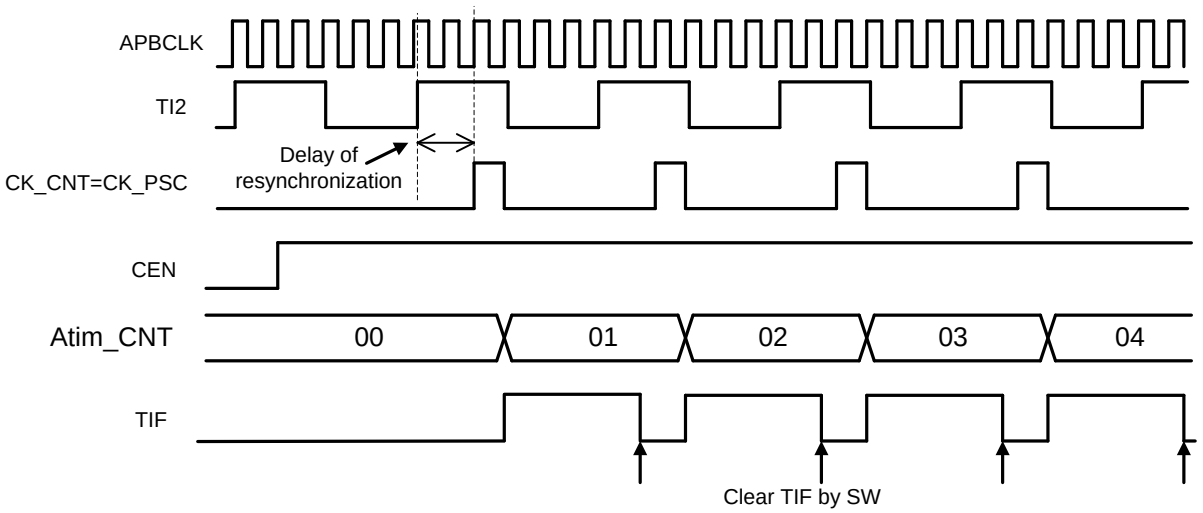


图 26-20 外部时钟模式 1 下的时序

26.6.5.3 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。

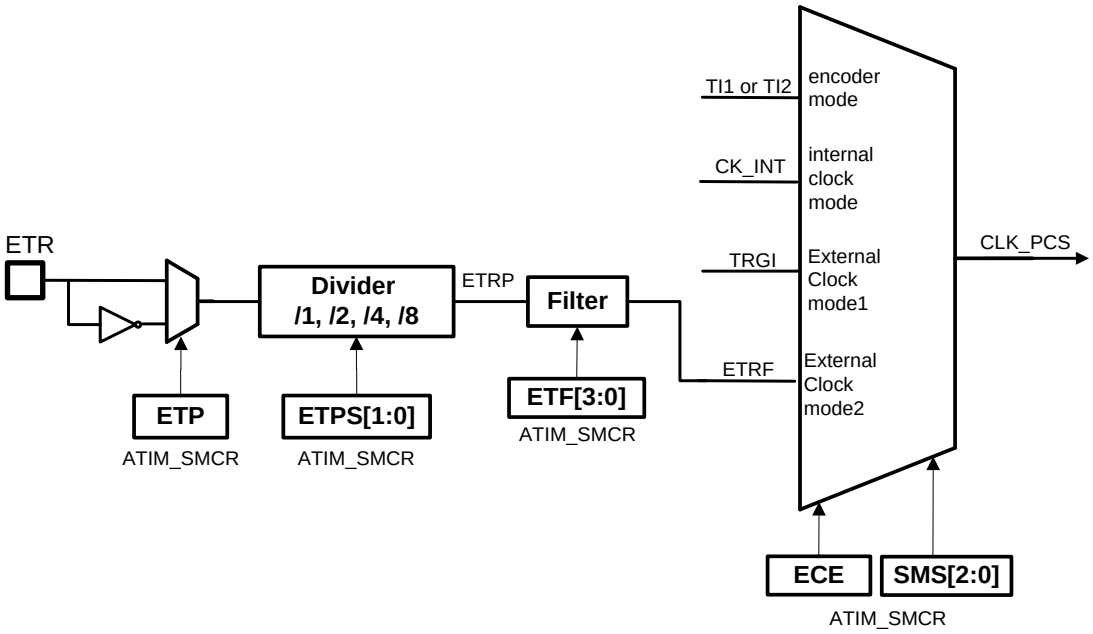


图 26-21 外部触发输入框图

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。

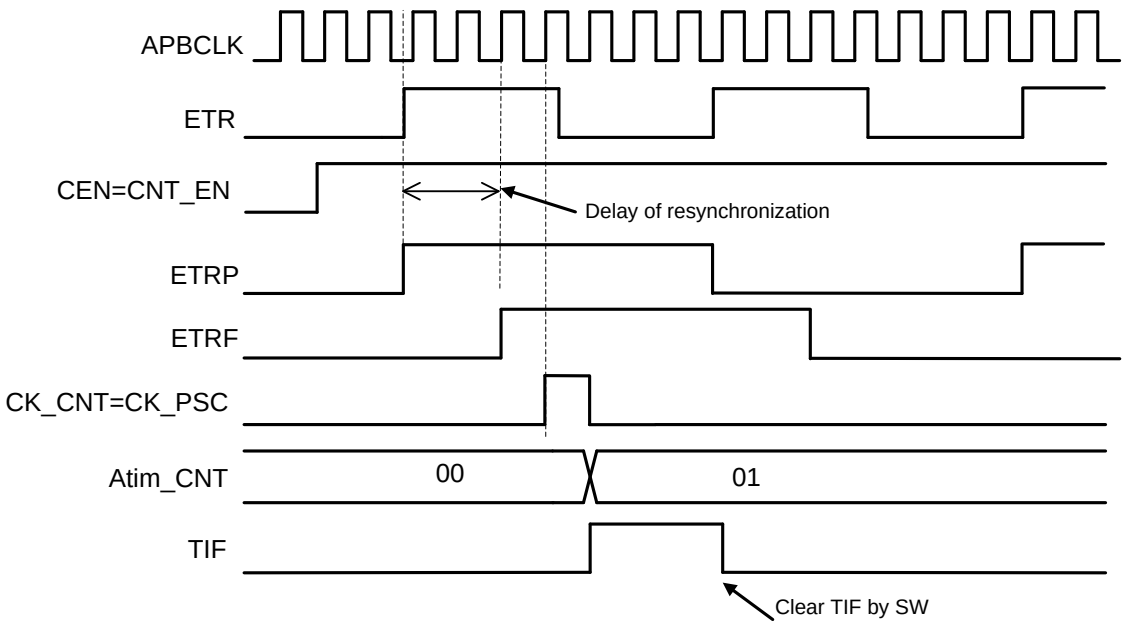


图 26-22 外部时钟模式 2 下的时序 1

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：

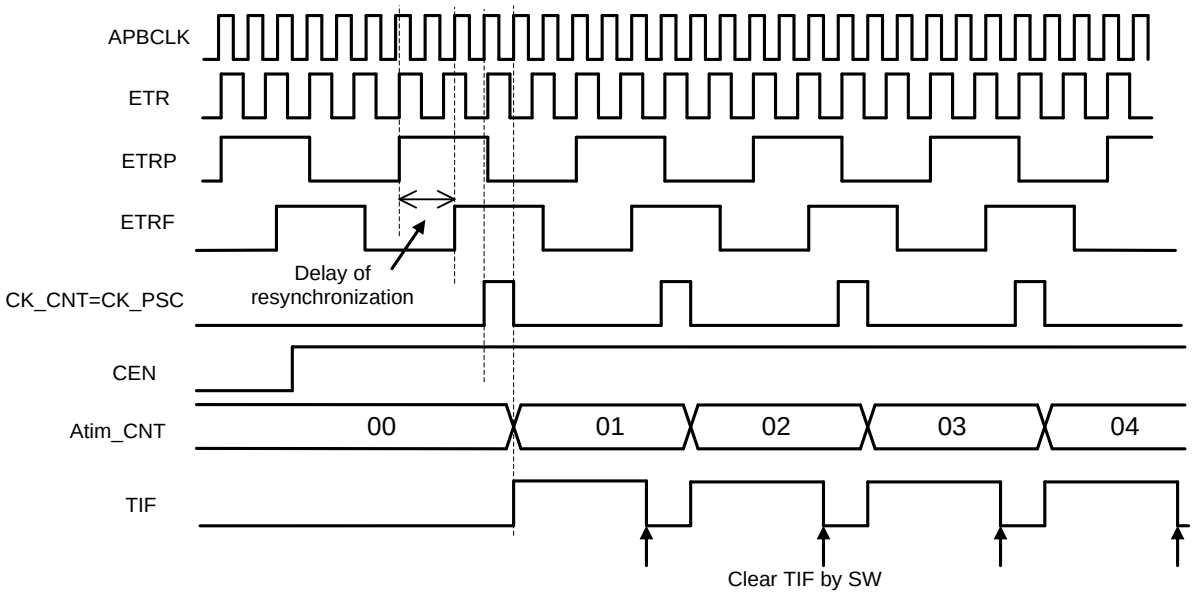


图26-23外部时钟模式2下的时序2

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。

26.6.6 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配置为用于选择ITR，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	ITR4(TS=1000)	ITR5(TS=1001)	ITR7(TS=1011)	ITR8(TS=1100)
ATIM	GPTIM0_TRGO	GPTIM1_TRGO	BSTIM16	LPTIM16

26.6.7 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。

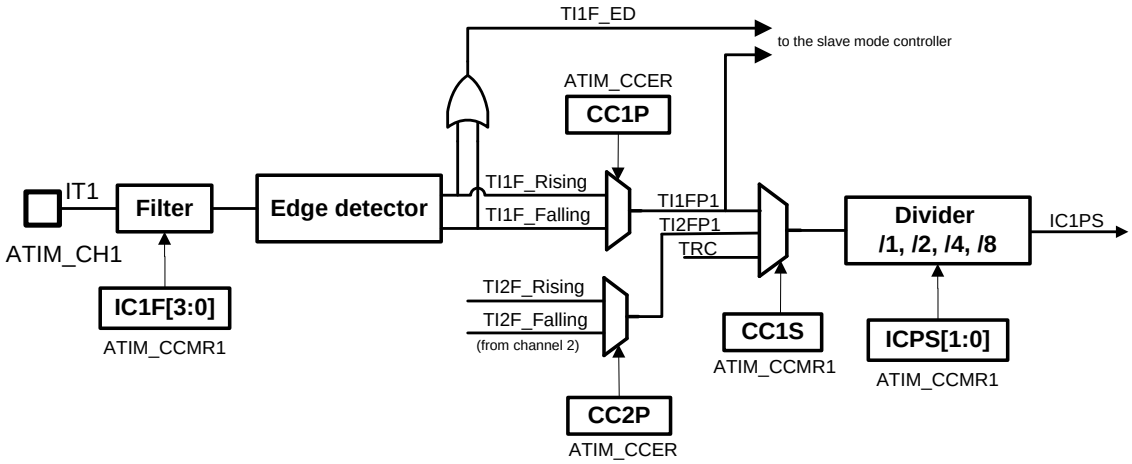


图26-24捕获/比较通道(通道1输入部分)

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。

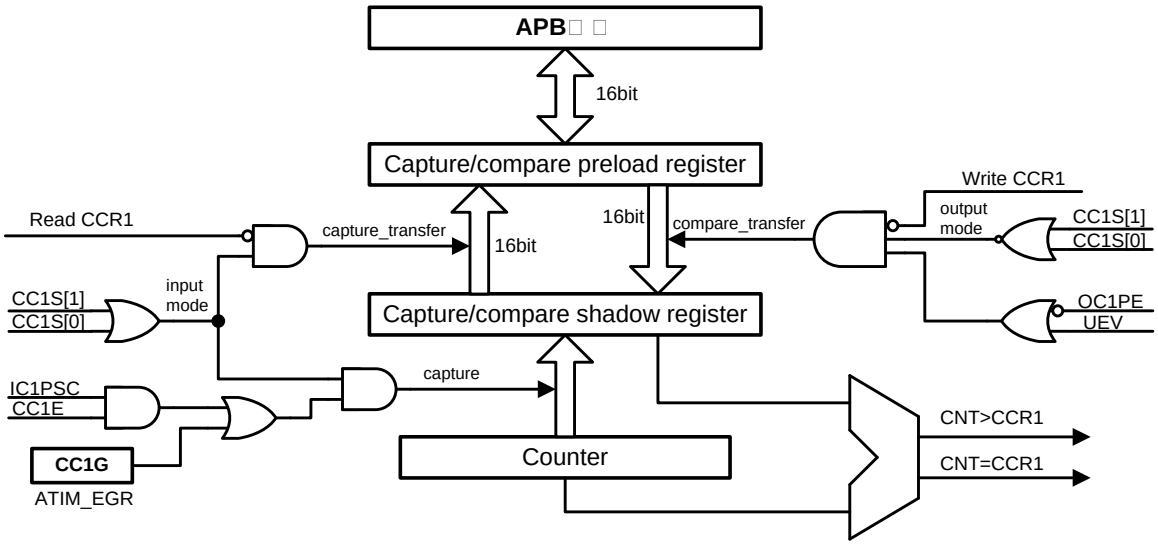


图26-25捕获/比较通道1的主电路

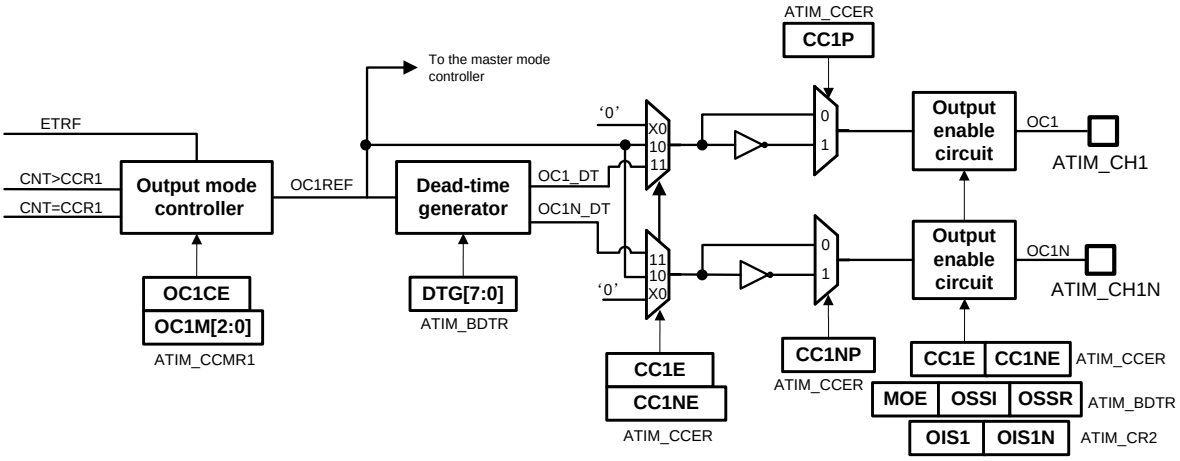


图26-26捕获/比较通道的输出部分(通道1至3)

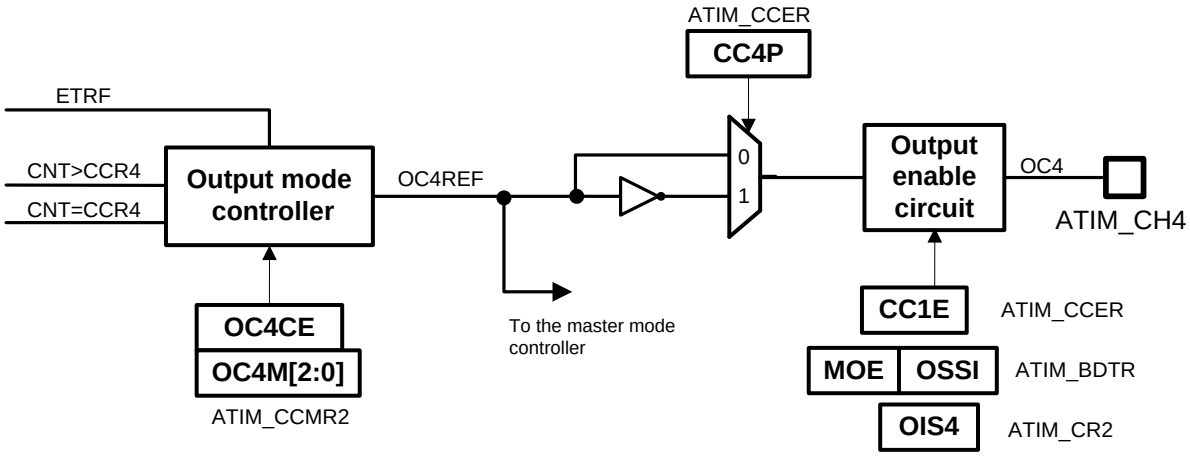


图26-27捕获/比较通道的输出部分(通道4)

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。

26.6.8 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

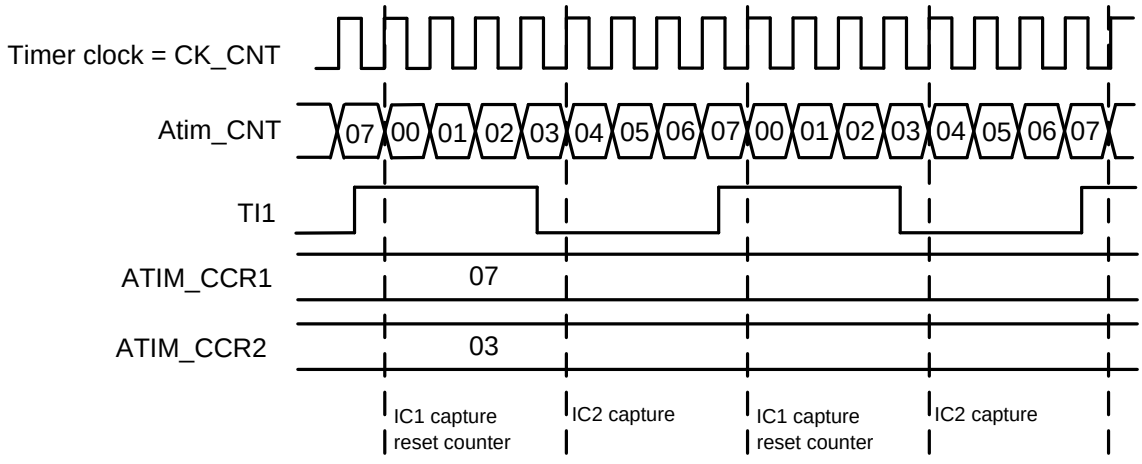


图26-28PWM输入捕获模式时序

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

26.6.9 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。

26.6.10 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

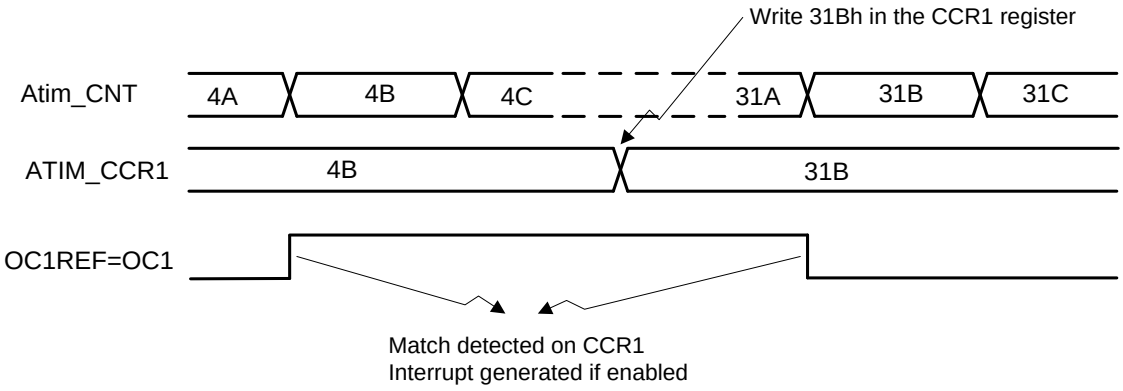


图26-29输出比较模式，翻转OC1

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。

26.6.11 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（通过配置CMS寄存器决定计数模式），PWM输出也支持边缘对齐和中央对齐模式。

PWM输出支持PWM模式1和PWM模式2，通过CCMR寄存器中的OCxM位配置。

PWM模式1：

- 向上计数时，CNT<CCR时OCxREF=1，CNT>=CCR时OCxREF=0
- 向下计数时，CNT>CCR时OCxREF=0，CNT<=CCR时OCxREF=1

PWM模式2：

- 向上计数时，CNT<CCR时OCxREF=0，CNT>=CCR时OCxREF=1
- 向下计数时，CNT>CCR时OCxREF=1，CNT<=CCR时OCxREF=0

26.6.11.1 PWM 边缘对齐模式

在向上计数的情况下（ATIM_CR1.DIR=0），下面的例子中配置为PWM模式1，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。

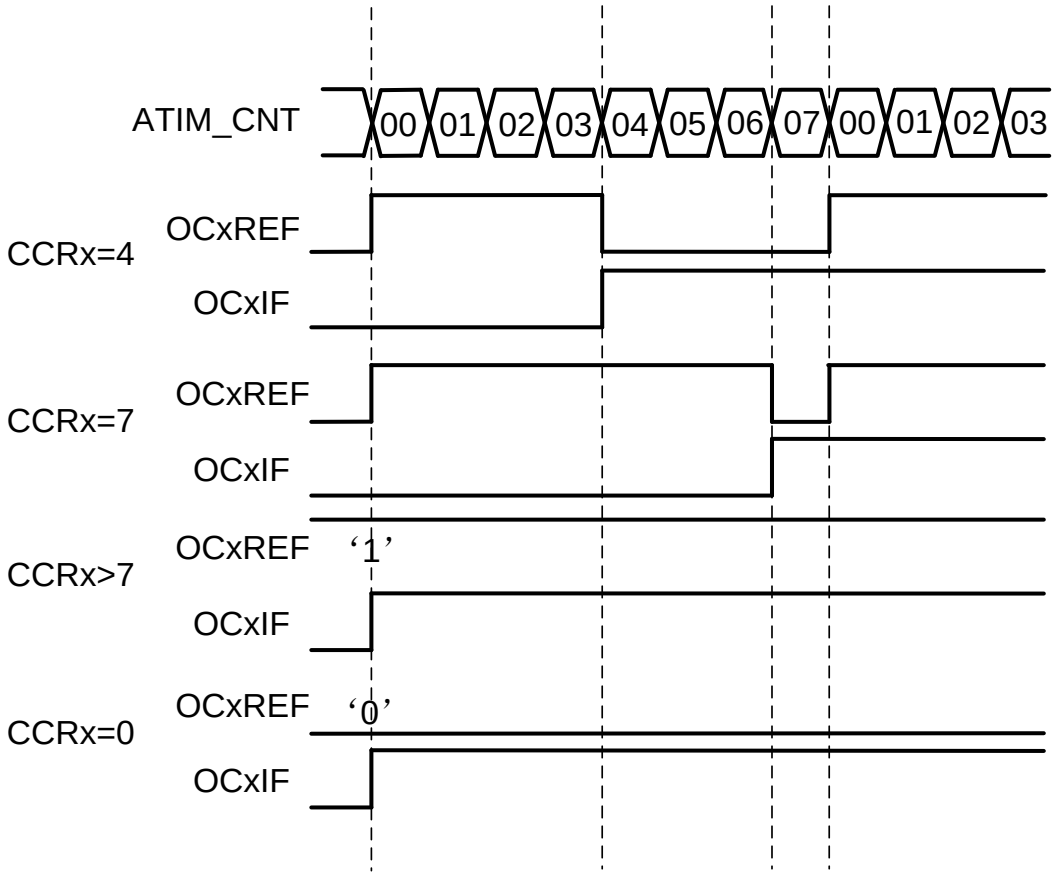


图26-30边沿对齐的PWM波形(ARR=7, PWM mode 1)

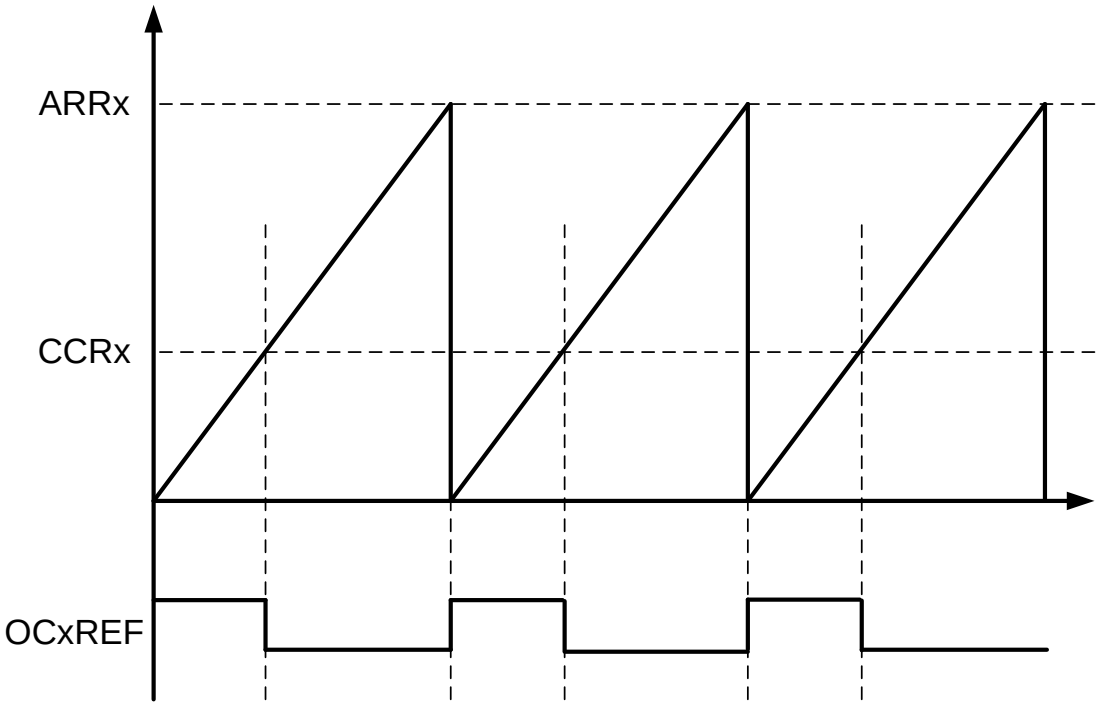


图26-31边沿对齐的PWM波形(PWM mode 1)

在向下计数时，OCxREF电平高低定义与向上计数时相同。

26.6.11.2 PWM 中心对齐模式

OCxREF电平定义与边缘对齐模式相同。此时DIR寄存器位由硬件根据当前计数方向自动更新，软件无法修改。CMS寄存器配置比较标志置位的时间点。下图是一个示例：

- ARR=7
- PWM mode 1
- OcxIF标志置位的位置由CMS寄存器配置

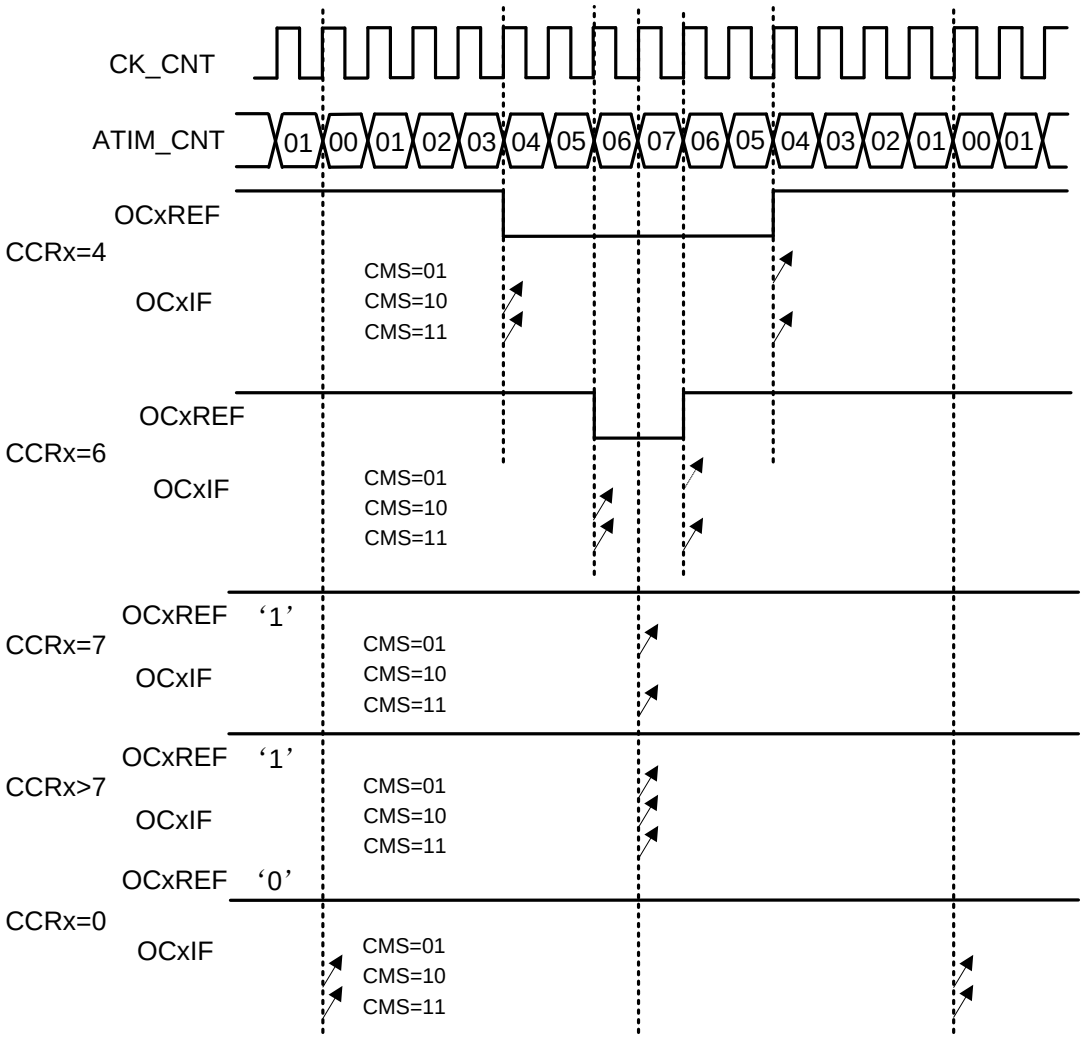


图26-32中央对齐的PWM波形(APR=7, PWM mode 1)

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。

26.6.12 非对称 PWM 模式

非对称PWM用于产生非对称的中心对齐PWM波形，需要同时使用两组CCR寄存器，由ARR寄存器决定PWM波形周期，一组CCR寄存器决定向上计数周期内的PWM信号变化点，另一组CCR寄存器决定向下计数周期内的PWM信号变化点。

使用非对称PWM模式时，只有OC1REF通道和OC3REF通道可以输出非对称波形，其中OC1REF由CCR1和CCR2寄存器控制，OC3REF由CCR3和CCR4寄存器控制。OCxM寄存器可以配置输出通道为非对称PWM模式1或者非对称PWM模式2。

下图为非对称相移PWM输出波形示例。

- ARR=7
- OC1REF为PWM mode 1，OC2REF为PWM mode 2

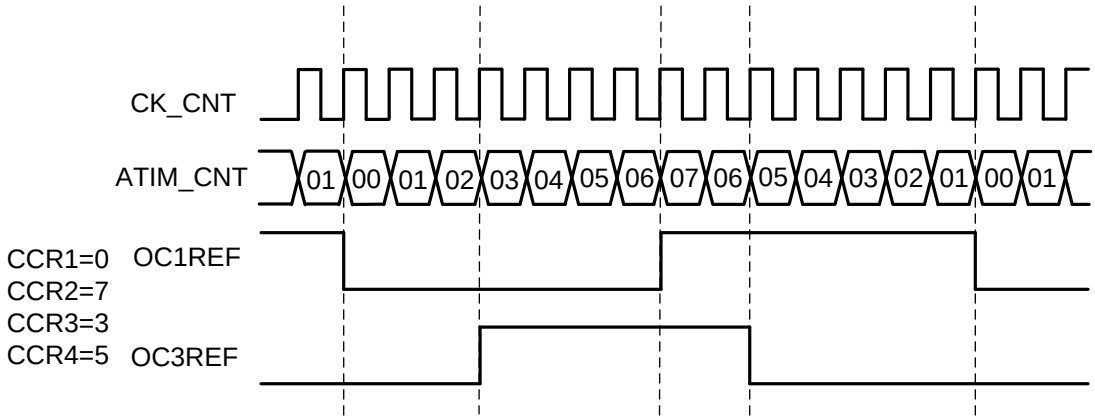


图26-33非对称PWM输出

使用非对称输出时，通道2和通道4仍可以单独输出普通PWM波形。

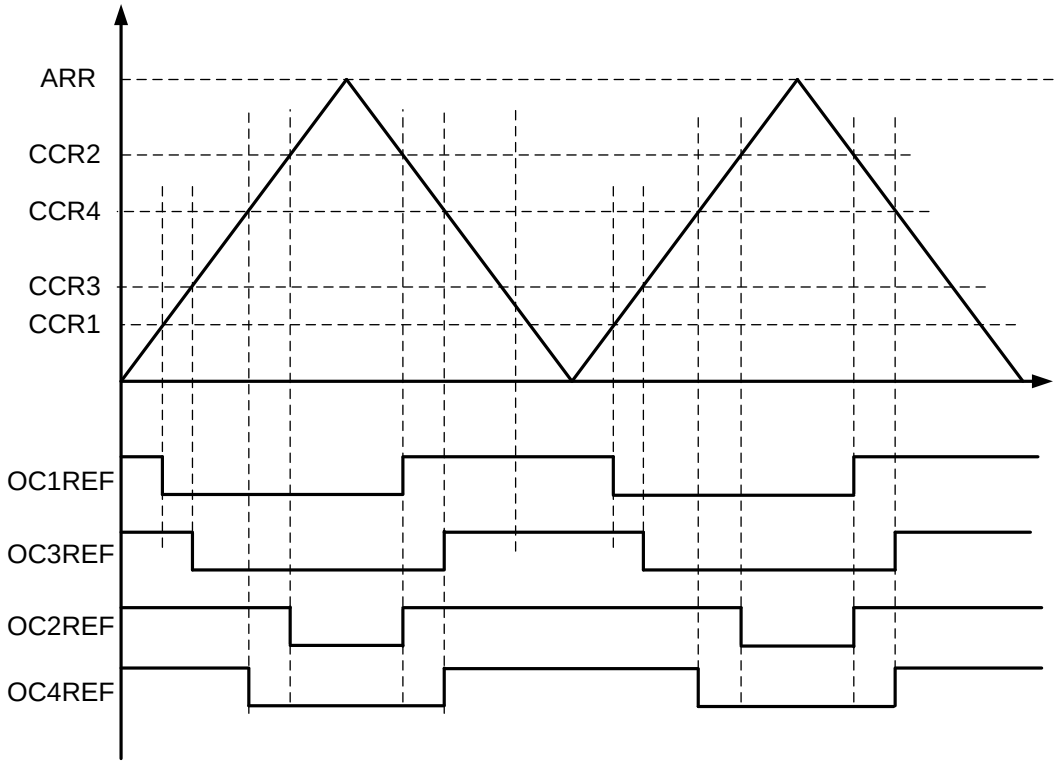


图26-34通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 1）

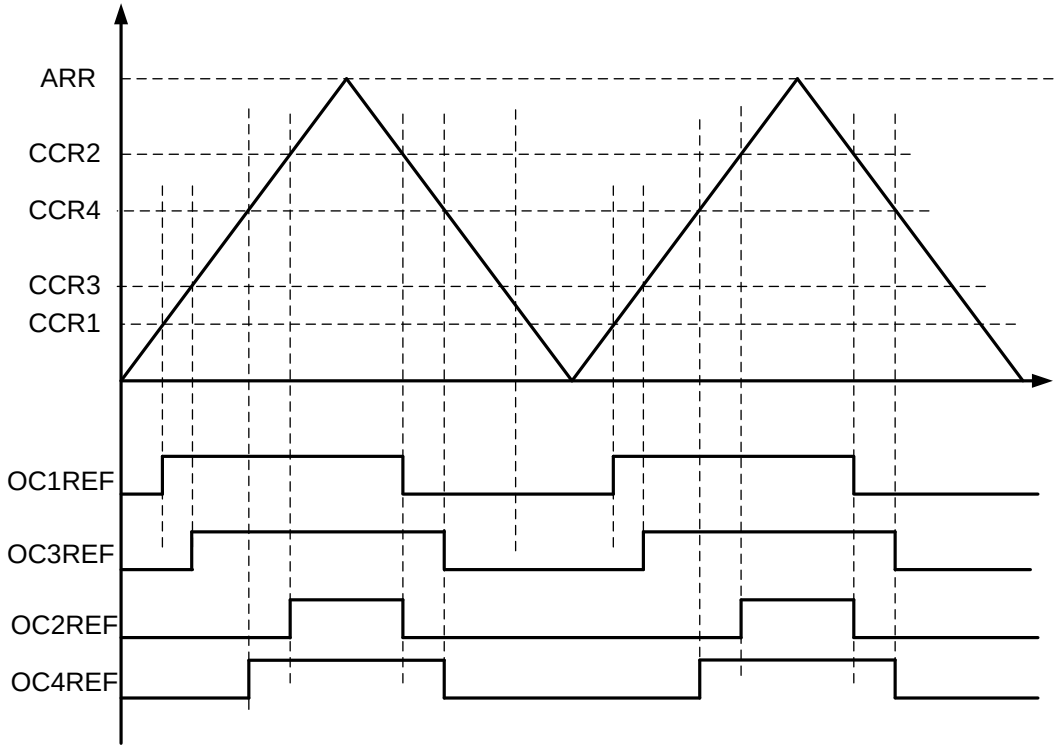


图26-35通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 2）

26.6.13 组合 PWM 模式

通道1和通道3可以配置为组合PWM模式，此模式下通道1的PWM波形输出由OC1REF和OC2REF组合产生，通道3的PWM波形输出由OC3REF和OC4REF组合产生，组合方式是OR或者AND，根据OCxM寄存器选择组合模式1（OR）或者组合模式2（AND）。

下图是组合PWM的例子：

- 通道1为组合PWM模式2
- 通道2为PWM模式1
- 通道3为组合PWM模式1
- 通道4为PWM模式2
- 通道1组合PWM输出

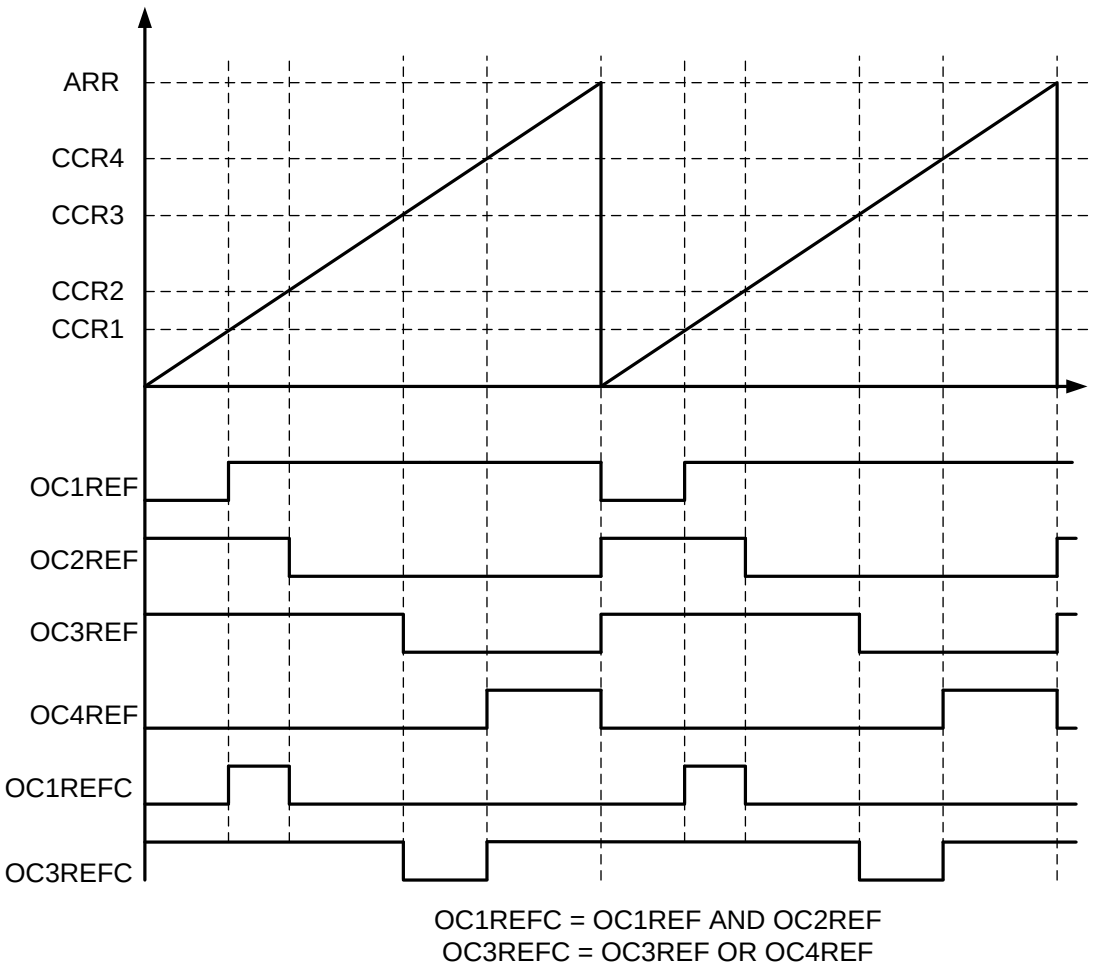


图26-36通道1/3组合PWM输出

26.6.14 组合三相 PWM 模式

组合三相PWM模式使用OC5REF信号与其他组合PWM信号相与的方式，来产生1~3路三相PWM信号。GC5C[3:1]寄存器用于选择OC5REF作用于OC1REF~OC3REF中的哪一路。

- 当GC5C1=1，OC1REF与OC5REF相与后输出
- 当GC5C2=1，OC2REF与OC5REF相与后输出
- 当GC5C3=1，OC3REF与OC5REF相与后输出

下图是一个三相组合输出的例子，通过preload寄存器，可以实现每个PWM周期结束时，自动更新GC5Cx寄存器，以实时控制每个PWM周期中OC5REF对哪路输出进行组合。

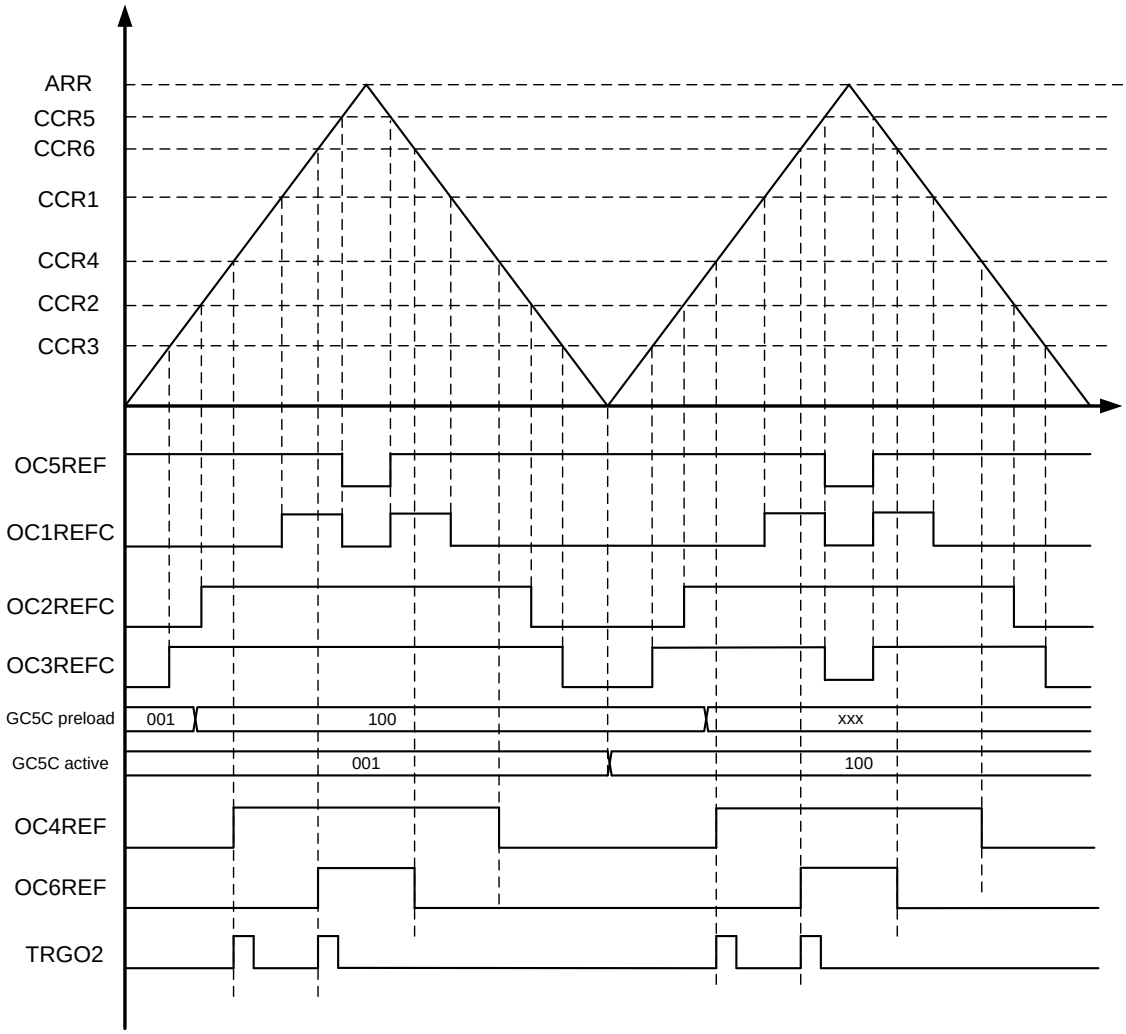


图26-37组合三相PWM输出

26.6.15 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。当通道的互补输出使能的情况下（CCxNE=1），死区插入功能才会有效。

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。如果延迟大于OCxREF高电平宽度，则输出脉冲不会产生。

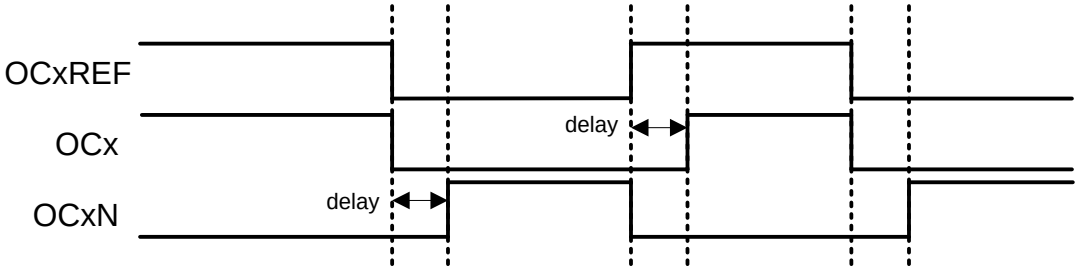


图26-38带死区插入的互补输出

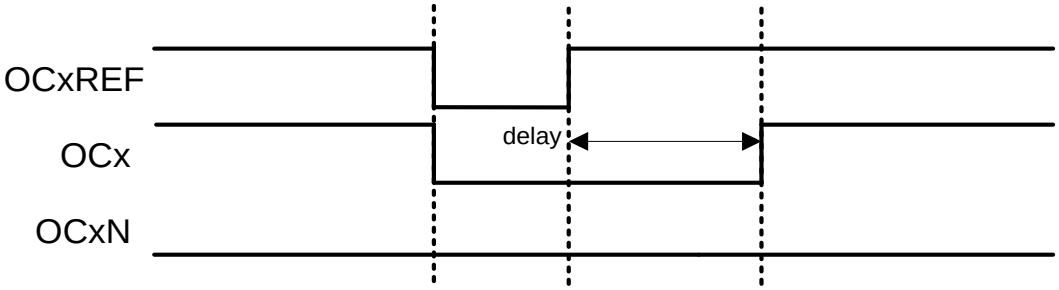


图26-39死区波形延迟大于负脉冲

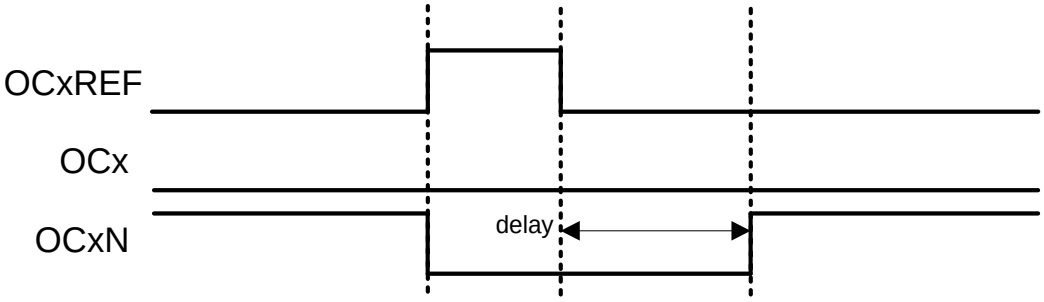


图26-40死区波形延迟大于正脉冲

注意，DTG定义的死区宽度对于同相输出和互补输出是同样有效的。由CCxP和CCxNP寄存器定义同相和互补输出的极性。

- CCxP=0: OCx active HIGH; CCxP=1: OCx active LOW
- CCxNP=0: OCxN active HIGH; CCxNP=1: OCxN active LOW

死区插入电路先对OCxREF信号的上升沿和下降沿分别进行延迟，得到OCx_DT和OCxN_DT信号，然后输出电路再根据CCxP和CCxNP的值，对OCx_DT和OCxN_DT进行极性处理后输出。

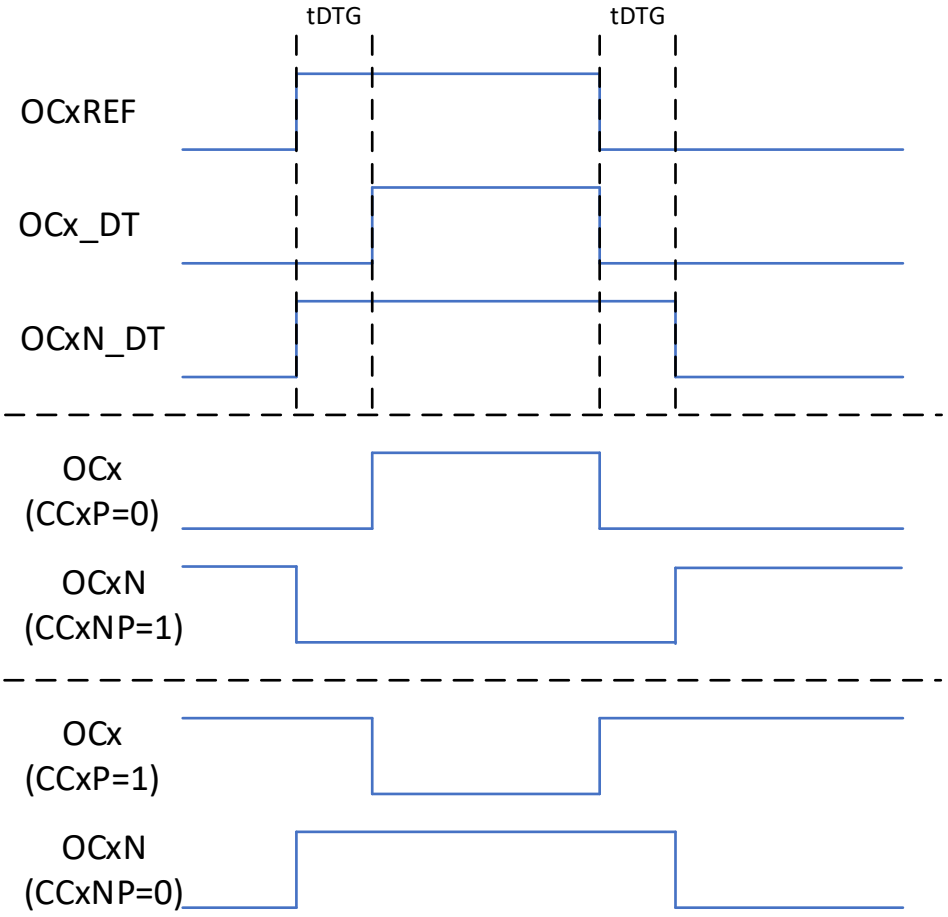


图26-41死区插入后的互补输出波形与CCxP和CCxNP的关系

26.6.16 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停振检测产生的报警输出、CPU内核LOCKUP；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。

26.6.16.1 术语定义

首先明确几个基本概念：

- OCxREF信号为高称为有效电平（active state），为低称为无效电平（inactive state）
- OCx和OCxN互补输出的有效电平（active state）和无效电平（inactive state）并不是固定不

变的，而是由输出极性寄存器CCxP和CCxNP定义，参见“互补输出和死区插入”章节

- 运行模式（RUNmode）：CC通道正常工作，并且总输出使能MOE=1
- 空闲模式（IDLE mode）：相对运行模式而言，总输出使能MOE=0；导致MOE清零的原因可能是软件清零或者发生了刹车事件；MOE清零并不影响计数器和CC通道的工作
- 空闲模式输出状态（Output state in IDLE mode）：指IDLE模式下OCx和OCxN的输出状态，由OISx和OISxN寄存器定义
- Off-state：输出关闭状态，即OC/OCN是inactive，而GPIO可以是输出Hi-Z或者输出inactive state；在RUN mode下由OSSR寄存器定义，在IDLE模式下由OSSI寄存器定义

输出状态转移与寄存器之间的关系，可以用下图表示：

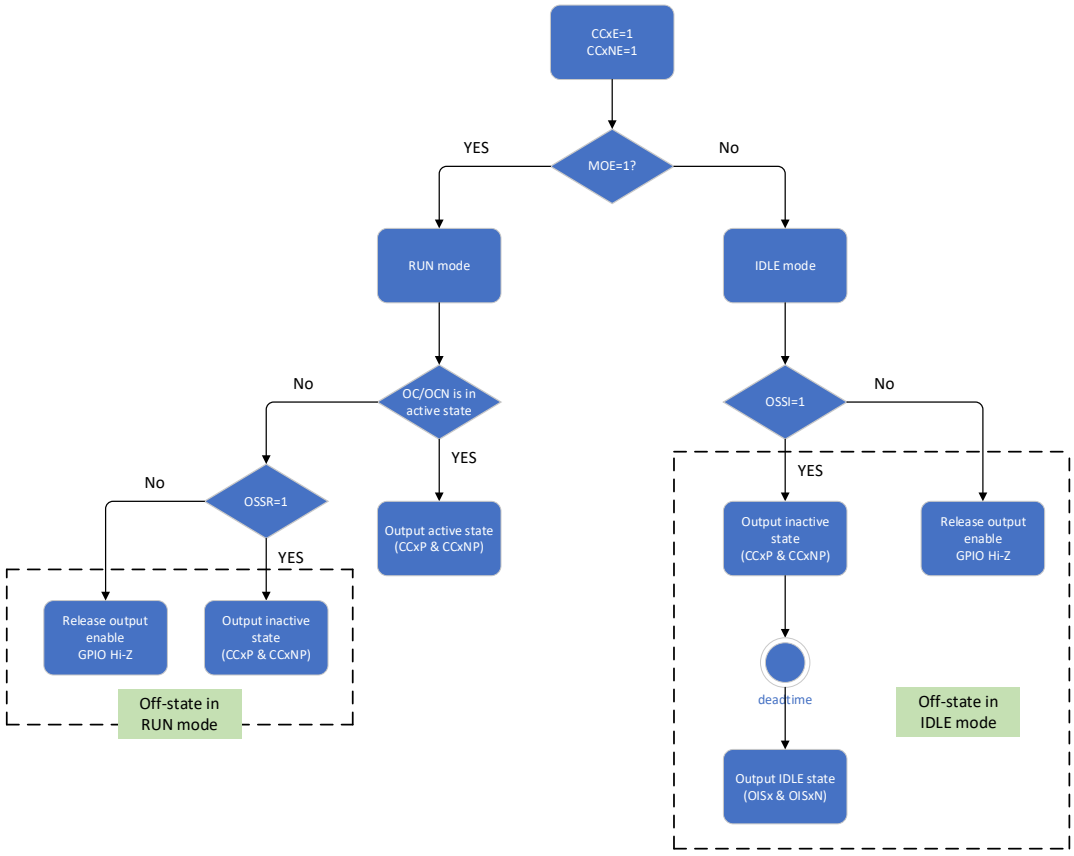


图26-42模式与输出状态术语

26.6.16.2 刹车逻辑

刹车输入控制逻辑如下图所示：

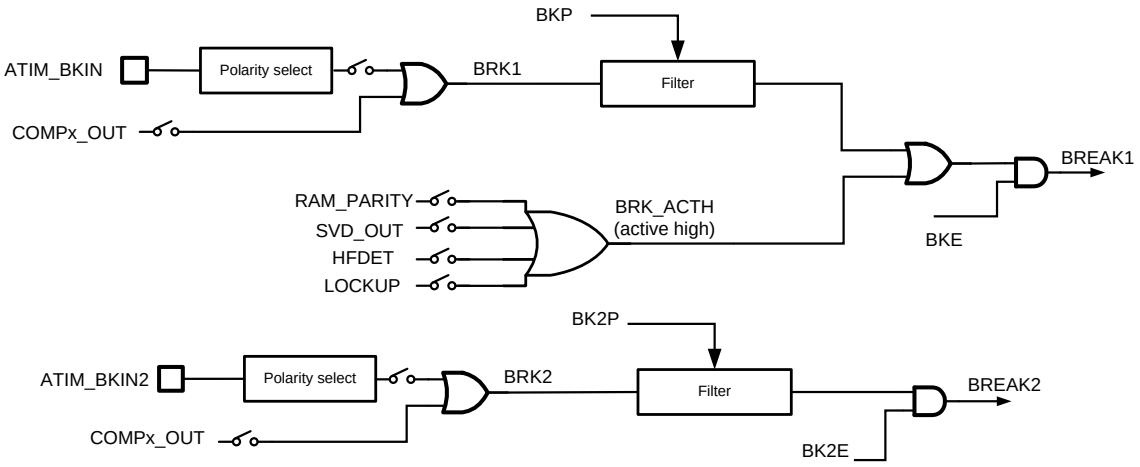


图26-43刹车输入结构

ATIM_BKIN和BKIN2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。BKE和BK2E用于使能或者关闭刹车输入，BKP和BK2P寄存器用于选择有效刹车的边沿，数字滤波功能仅针对BRK1和BRK2信号，对于芯片内部产生的BRK_ACTH无效。

高级定时器支持的刹车输入信号源包括：

- BRK1包括
  - 外部引脚输入BKIN
  - 比较器输出COMPx_OUT
- BRK_ACTH包括
  - SVD检测输出
  - 高频晶体停振
  - 内核锁定LOCKUP
  - SRAM校验位错误
- BRK2包括
  - 外部输入引脚BKIN2
  - 比较器输出COMPx_OUT

BRK1和BRK_ACTH组合成BREAK1送入内部，BRK2产生BREAK2送入内部。

26.6.16.3 刹车行为

当一个刹车事件发生时：

- 输出使能寄存器MOE被异步清零，即使MCU的时钟停振，此功能依然起作用（注意使用异步清零必须关闭刹车滤波）
- 如果没有使能互补输出
  - 如果OSSI=1：MOE清零后，每个输出通道立即被驱动为OISx寄存器定义的电平；
  - 如果OSSI=0：MOE清零后，ATIM将IO控制权释放（关闭输出使能），GPIO将输出hi-Z
- 如果使能了互补输出，且OSSI=1
  - 输出立即被异步置位成inactive状态
  - 如果timer时钟仍在工作，死区插入电路开始工作，在死区时间后驱动输出为OISx和OISxN所定义的电平；注意此时硬件禁止OC和OCN同时输出active状态，因此OISx和OISxN的配置要注意正确性；由于MOE下降沿异步置位OC/OCN，时钟重同步过程将导致死区时间比配置值长2个ck_tim时钟周期
- 如果使能了互补输出，且OSSI=0
  - ATIM释放IO控制权，输出使能清零，GPIO输出hi-Z
- 刹车标志寄存器置位（BIF或B2IF），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。

刹车输入控制和输出管理寄存器带有写保护功能，允许应用程序将部分配置参数锁定，避免误改写（死区长度、OCx/OCxN极性、OCxM配置、刹车使能和极性）。应用可以通过ATIMx_BDTR中的LOCK寄存器选择3个等级的保护。LOCK寄存器在芯片复位释放后仅能写一次，此后不能改写，直到芯片复位。

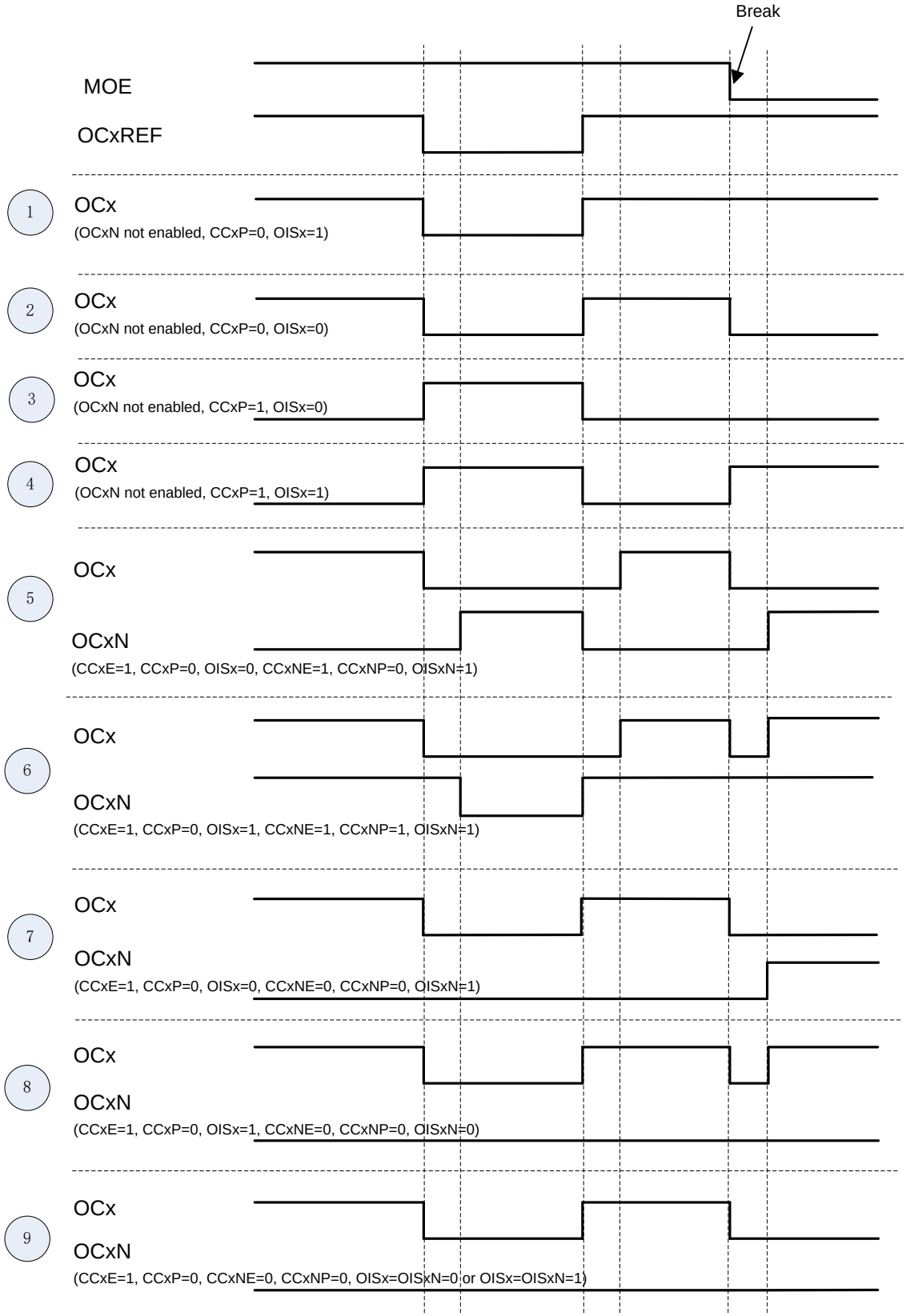


图26-44响应刹车的输出

上图中各组波形输出说明：

1. 关闭互补输出，OCx active high，idle state=1，刹车事件发生后OCx异步置位
2. 关闭互补输出，OCx active high，idle state=0，刹车事件发生后OCx异步清零
3. 关闭互补输出，OCx active low，idle state=0，刹车事件发生后OCx异步清零
4. 关闭互补输出，OCx active low，idle state=1，刹车事件发生后OCx异步置位
5. 使能互补输出，OCx active high，idle state=0；OCxN active high，idle state=1；死区插入；刹车事件发生后OCx异步清零，OCxN先异步清零（inactive），等待deadtime后置位
6. 使能互补输出，OCx active high，idle state=1；OCxN active low，idle state=1；死区插入；刹车事件发生后OCx先异步清零（inactive），等待deadtime后置位，OCxN异步置位
7. 关闭互补输出，OCx active high，idle state=0；刹车事件发生后OCx异步清零（OISx=0），等待deadtime后OCxN置位（OISxN=1）
8. 关闭互补输出，OCx active high，idle state=1；刹车事件发生后OCx异步清零（inactive），等待deadtime后OCx置位（OISx=1），OCxN保持低电平（OISxN=0）
9. 特殊情况：关闭互补输出，OCx active high，idle state=1；刹车事件发生后OCx异步清零（inactive），等待deadtime后OCx试图置位（OISx=1），但是由于OISxN=1，deadtime后将导致OCx和OCxN同时为1（同时active），这是禁止的，所以硬件会保持OCx和OCxN输出为0

两个刹车输入对于定时器输出有不同的作用效果：

- BREAK1可以将PWM输出置于inactive state，或置于预定义的安全状态
- BREAK2只能将PWM输出置于inactive state

BREAK1优先级高于BREAK2。

BREAK1	BREAK2	ATIM输出状态	典型使用场景	
			OCxN输出	OCx输出
Active	X	首先inactive，等待deadtime后强制为预定义状态 如果OSSI=0则输出关闭（控制权释放给GPIO模块）	ON after deadtime insertion	OFF
Inactive	Active	Inactive	OFF	OFF

表 26-10 刹车行为

26.6.17 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		

26.6.18 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。

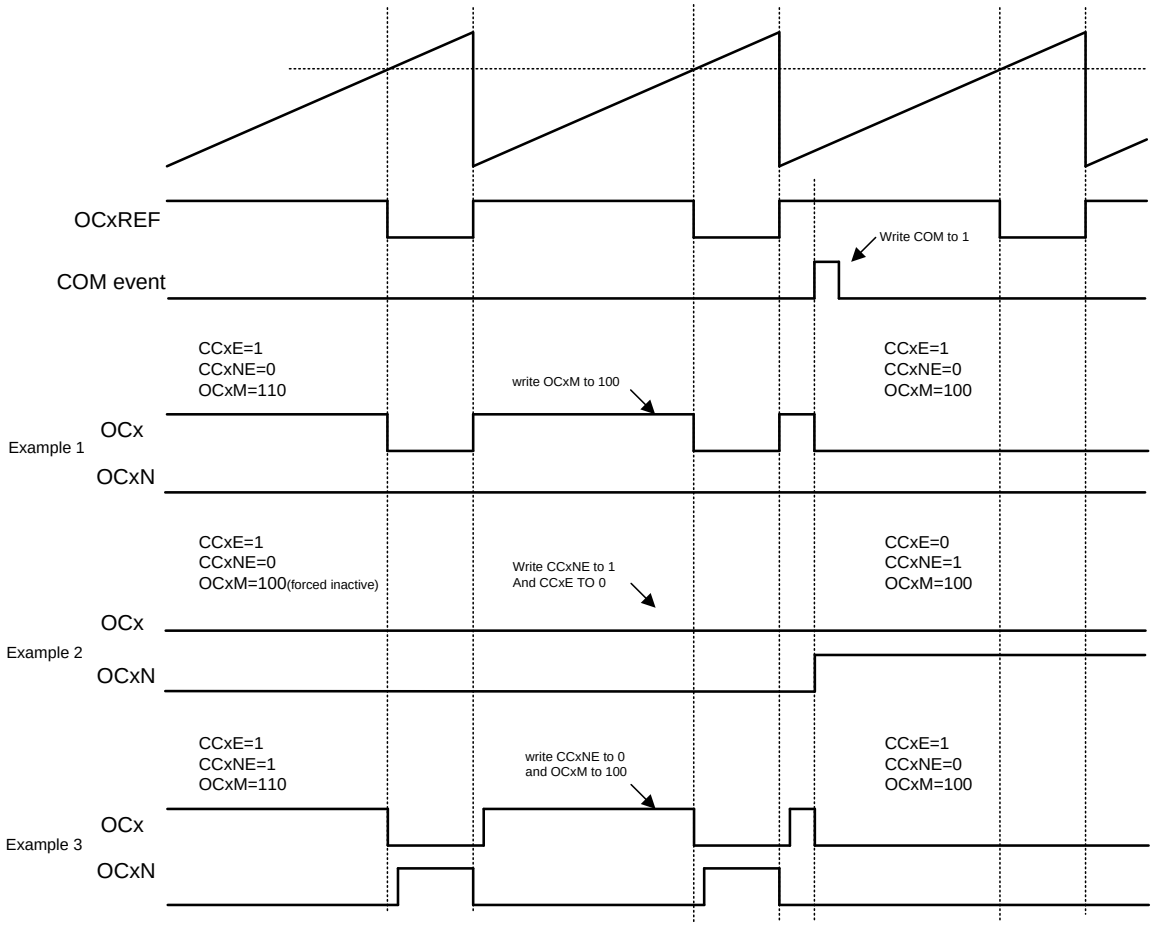


图26-45产生六步PWM，使用COM的例子(OSSR=1)

26.6.19 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

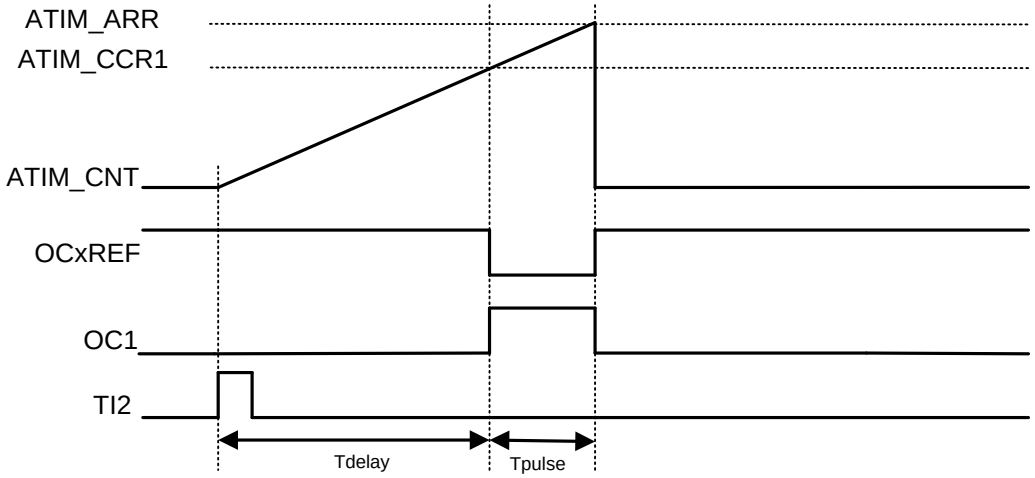


图26-46单脉冲模式的例子

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

26.6.20 可重触发的单脉冲输出（retriggerable one-pulse mode）

与OPM模式相比，可重触发的OPM模式的区别在于：

- 当触发事件发生时，立即输出脉冲（没有可编程延迟）
- 如果上一次OPM进程还未完成，新的触发事件将扩展输出脉冲宽度

使用此功能时，定时器必须是从机模式，配置SMS[3:0]=1000（组合的复位+触发模式），并且OCxM设置为1000或1001（可重触发OPM模式1或2）。

如果定时器配置为向上计数模式，CCRx必须设置为0，ARR用于定义脉冲宽度。如果定时器配置为向下计数模式，CCRx必须设置为大于等于ARR。

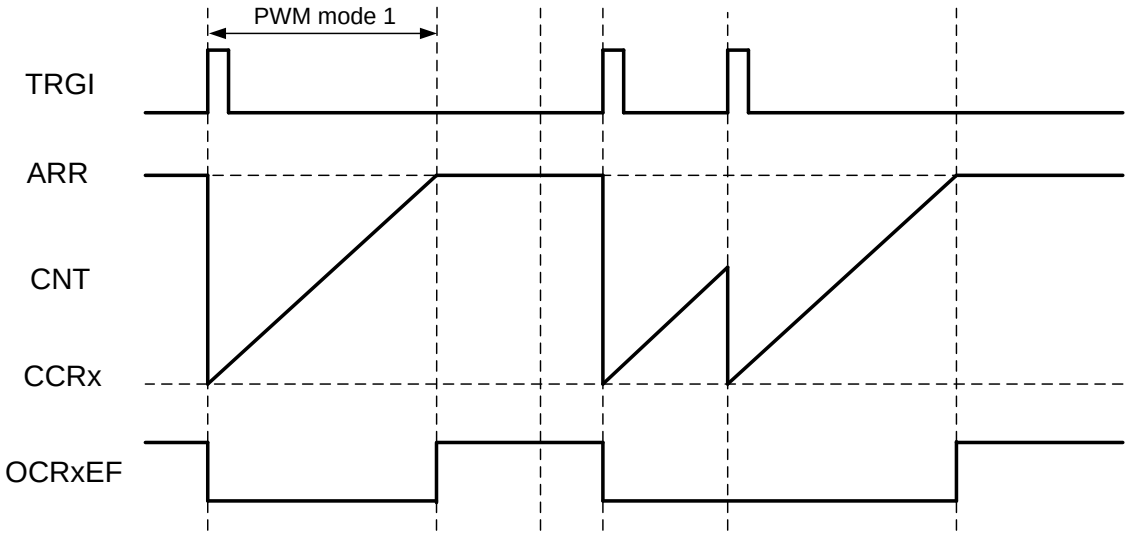


图26-47重触发OPM模式1，向上计数，CCR=0

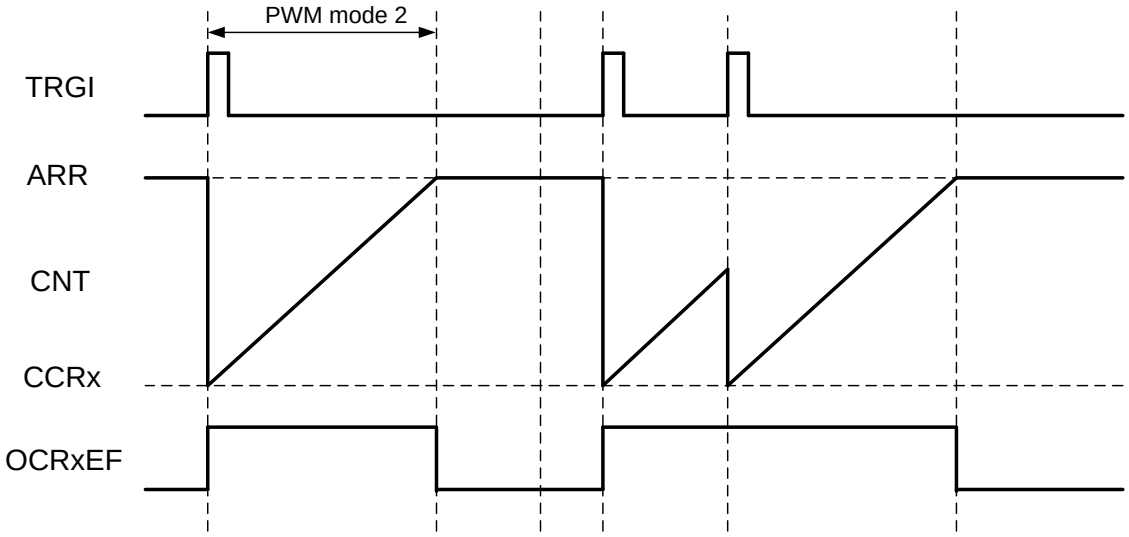


图26-48重触发OPM模式2，向上计数，CCR=0

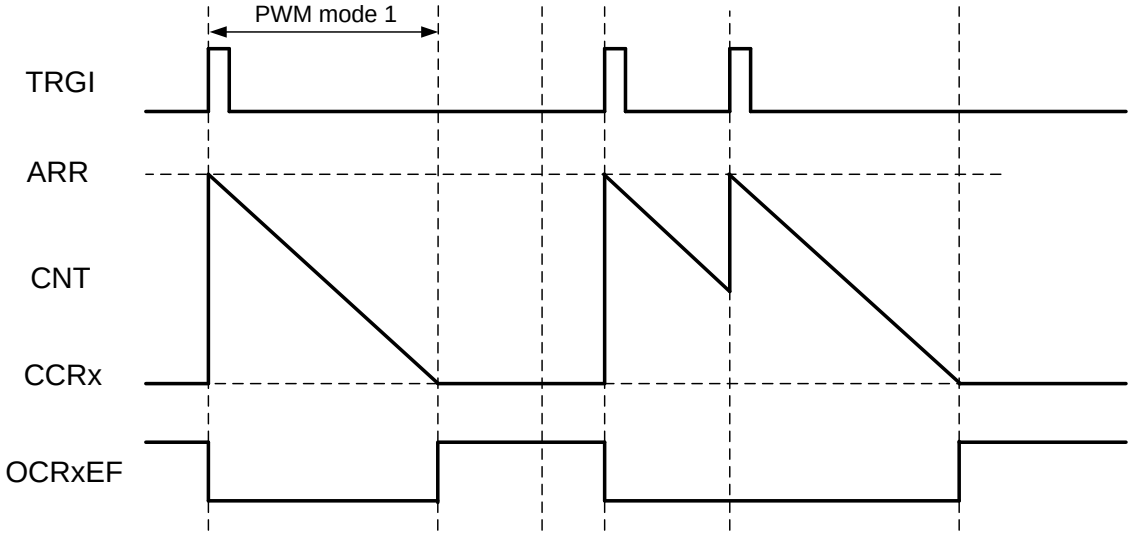


图26-49重触发OPM模式1，向下计数，CCR=0

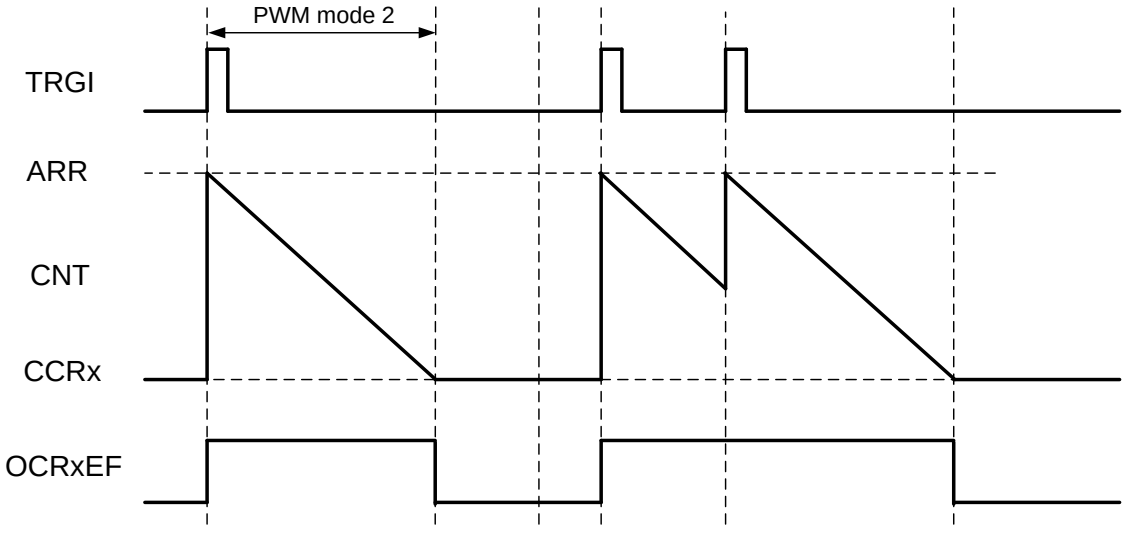


图26-50重触发OPM模式2，向下计数，CCR=0

26.6.21 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。

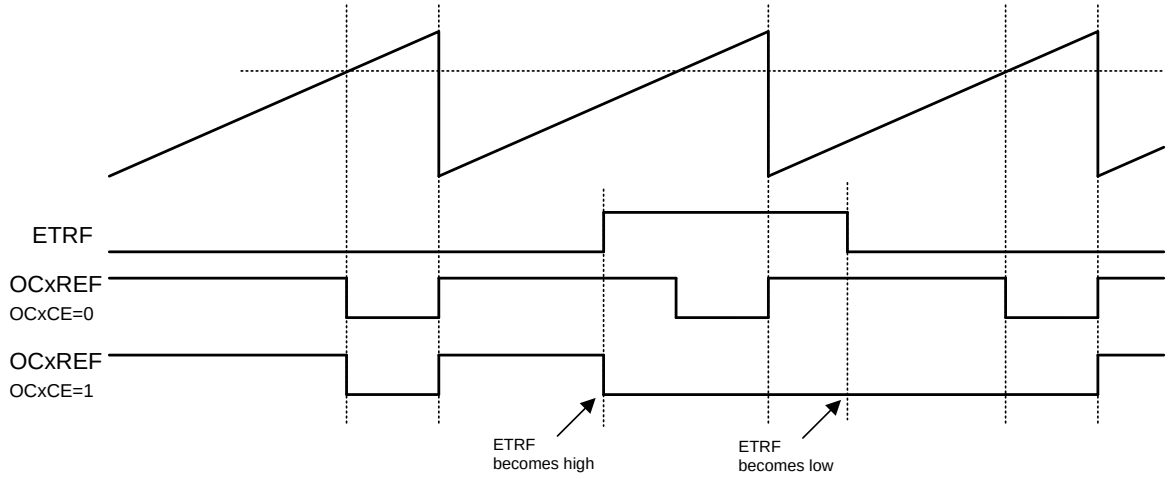


图26-51 ETR信号清除ATIM的OCxREF

26.6.22 编码器接口模式（encoder interface mode）

26.6.22.1 正交编码器

ATIM使用两个输入信号TI1和TI2来与正交编码器连接，输入信号首先经过数字滤波和极性选择，定时器再根据滤波和极性选择后的信号TI1FP1和TI2FP2的边沿进行计数。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，同时硬件自动更新DIR寄存器，向软件指示当前计数方向。根据SMS寄存器的配置，ATIM可以只对TI1的边沿计数、只对TI2的边沿计数、或者同时对TI1和TI2的边沿计数。

编码器接口模式启动后，计数器根据输入信号在0和ARR范围内连续计数，因此使能定时器之前必须配置ARR寄存器。

下表是计数方式与两路输入信号之间的关系：

有效沿	SMS	对应信号的电平（TI1 对应TI2，TI2 对应TI1）	TI1FP1		TI2FP2	
			上升	下降	上升	下降
仅在TI1处计数	0001	高	递减	递增	不计数	不计数
		低	递增	递减	不计数	不计数

x2 mode						
仅在TI2处计数 x2 mode	0010	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 x4 mode	0011	高	递减	递增	递增	递减
		低	递增	递减	递减	递增
仅在TI1处计数 x1 mode	1110	高	递减	递增	不计数	不计数
		低	不计数	不计数	不计数	不计数
仅在TI2处计数 x1 mode	1111	高	不计数	不计数	递增	递减
		低	不计数	不计数	不计数	不计数

表 26-11 encoder interface 计数方式（CC1P=CC2P=0）

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。

正交编码器的两路输出可以直接连接到MCU的输入管脚，同时，正交编码器的原位输出（index）信号可以连接到ETR输入，用于触发计数器复位。

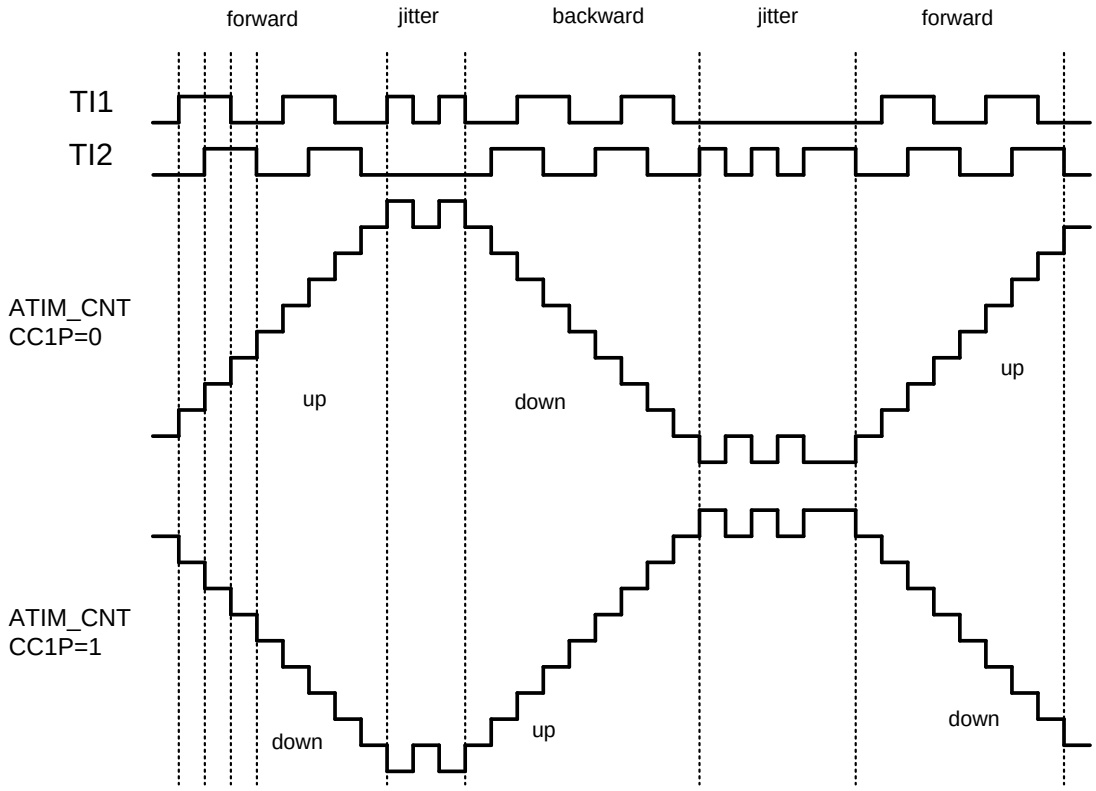


图26-52编码器模式下的计数器操作实例

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[3:0]=0011
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

编码器接口支持x4，x2，x1计数模式，其中x4模式在TI1和TI2的每个边沿计数，x2模式只在TI1或者TI2的边沿计数，x1模式则只在TI1或者TI2的某个边沿上计数（由CCxP选择有效沿）。

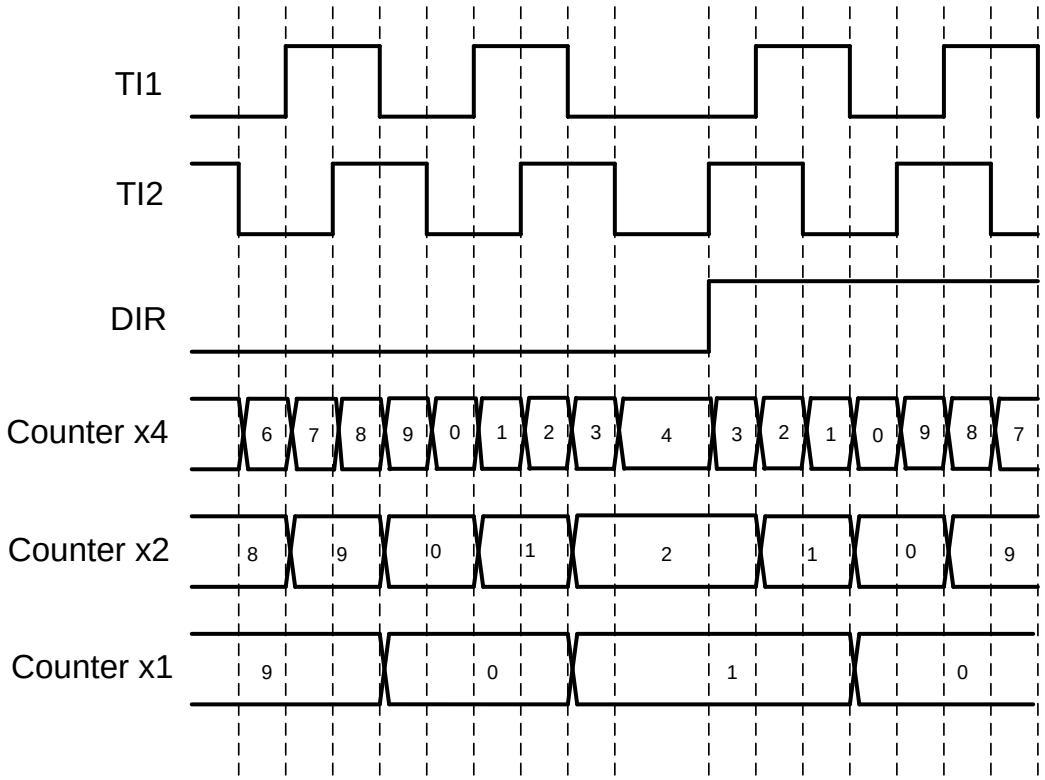


图26-53正交编码器不同边沿计数模式示例

当ATIM_CR1寄存器中的UIFREMAP位置位时，UIF中断标志寄存器被实时复制到ATIM_CNT[31]，供软件进行原子读取。

26.6.22.2 时钟方向编码器模式

除了正交编码器，另一种编码器提供“时钟”和“方向”两个输出信号。为支持这种编码器模式，需要将SMS配置为1010或1011，将时钟信号连接到TI2，将方向信号连接到TI1。

- SMS[3:0]=1010：x2模式，计数器在TI2输入信号的上升沿和下降沿都计数

- SMS[3:0]=1011：x1模式，计数器在TI2上升或者下降沿计数，由CC2P寄存器定义有效沿（CC2P=0：上升沿计数，CC2P=1：下降沿计数）

计数器计数方向由TI1输入的“方向”信号和CC1P寄存器决定。

- CC1P=0：TI1为高电平时向上计数，TI1为低电平时向下计数
- CC1P=1：TI1为高电平时向下计数，TI1为低电平时向上计数

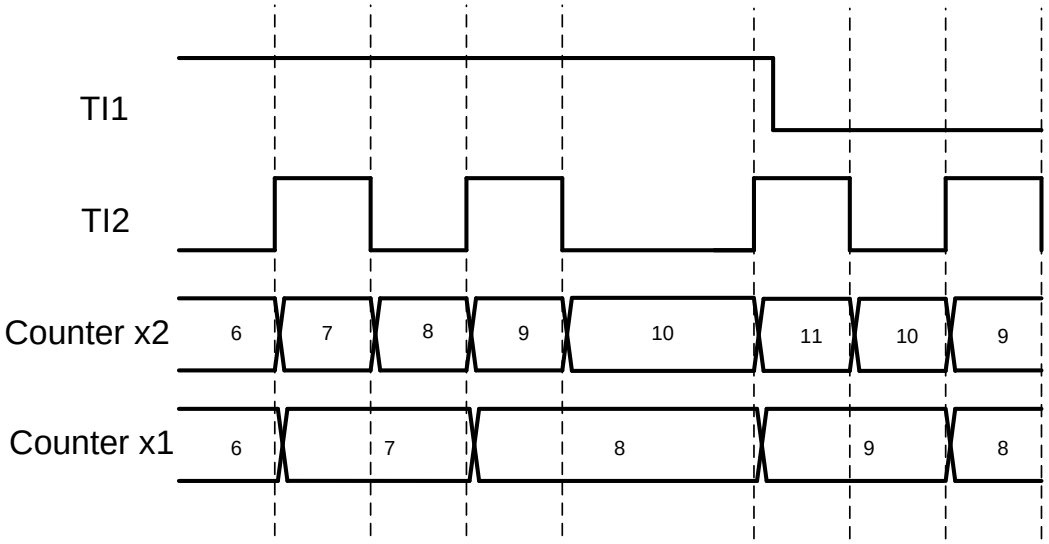


图26-54时钟方向编码器示例

26.6.22.3 带有方向信息的时钟编码器模式

这种编码器用时钟信号调制方向信息，输出两个信号，任意时刻只有其中一个信号带有时钟。

- SMS[3:0]=1100：x2模式，计数器在输入信号的上升沿和下降沿都计数。CC1P和CC2P用于定义时钟空闲状态，0表示高电平空闲，1表示低电平空闲
- SMS[3:0]=1101：x1模式，计数器在输入信号的上升或者下降沿计数，由CCxP寄存器定义有效沿（CCxP=0：下降沿计数，CCxP=1：上升沿计数）

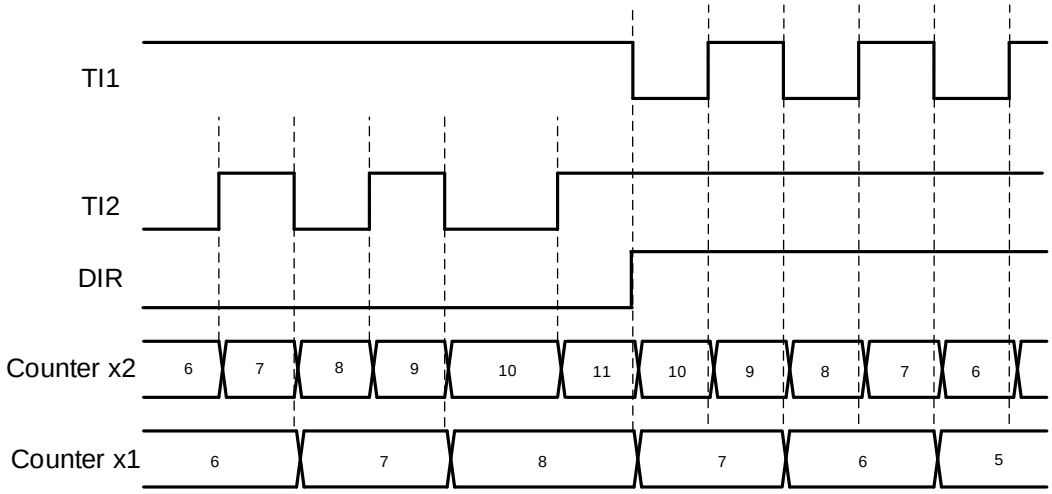


图26-55带方向的时钟编码器（CC1P=CC2P=0）

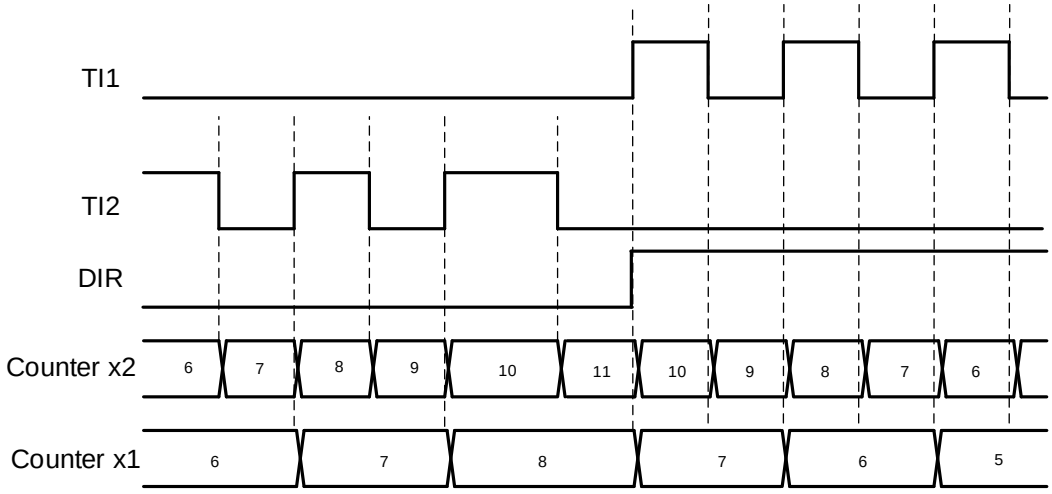


图26-56带方向的时钟编码器（CC1P=CC2P=1）

下表是计数方式与两路输入信号之间的关系：

带方向的时钟编码模式	SMS	对应信号的电平（TI1 对应 TI2，TI2 对应TI1）	TI1FP1		TI2FP2	
			上升	下降	上升	下降
CCxP=0 x2 mode	1100	高	递减	递增	递增	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x2 mode	1100	高	不计数	不计数	不计数	不计数
		低	递减	递增	递增	递增
CCxP=0 x1 mode	1101	高	不计数	递减	不计数	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x1 mode	1101	高	不计数	不计数	不计数	不计数
		低	递减	不计数	递增	不计数

表 26-12 directional clock mode 计数方式

26.6.22.4 Encoder index（Z 相输入）

正交编码器除了输出2路正交编码信号之外，通常还会输出1个周期归零指示信号（index）。这个零位信号可以用来复位计数器，需要将index信号连接到ATIM的ETR输入引脚。

Index功能通过ATIM_ECR寄存器中的IE寄存器使能，并且尽在encoder模式下有效，即SMS必须为0001/0010/0011/1010/1011/1100/1101/1110/1111

常见的编码器输出的零位信号有如下格式：

- Gated with A and B
- Gated with A or gated with B
- Ungated

不同零位信号的示意图如下：

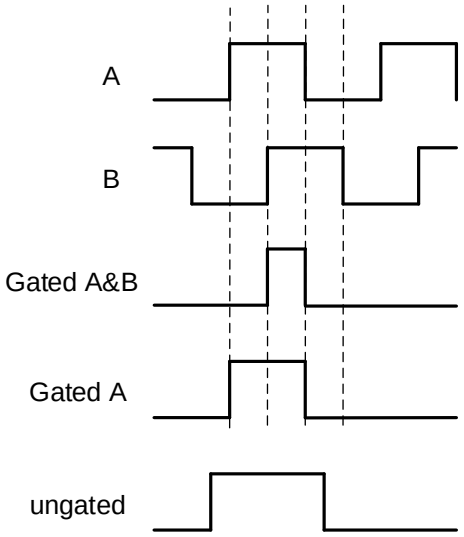


图26-57编码器index定义

使用中将编码器A相连接到TI1，B相连接到TI2，Z相连接到ETR。当ATIM检测到index输入后，产生index事件，执行对计数器的特定操作，注意这个操作是要被通道到特定状态上的，而不是异步即时发生的。

ATIM总是在检测index输入信号，然而检测到index后，触发计数器清零或置位的index事件将被同步到特定AB相状态上执行，通过ATIM_ECR中的IPOS[1:0]寄存器来定义index事件被同步执行的AB相位段。根据计数器计数方向，硬件电路采取不同的index事件执行策略：

- 向上计数时（DIR=0），index事件触发计数器清零

- 向下计数时（DIR=1），index事件触发计数器回到ARR

IPOS寄存器可以配置计数器在检测到index后，在什么AB相位下执行index事件，比如IPOS=11，计数器仅在A=1 &B=1的条件下执行index事件（虽然有可能在11状态之前就检测到index）。

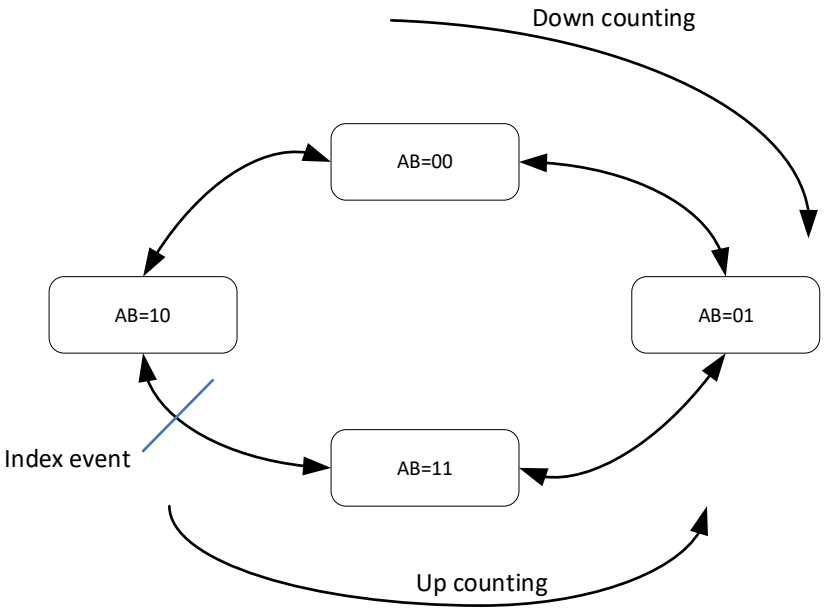


图26-58IPOS[1:0]=11时的index识别

下图显示了IPOS=11，定时器识别index事件，并且设置计数值的时序。

- 向上计数时（DIR=0），检测到index，当AB处于11相位计数值被清零（1次）
- 向下计数时（DIR=1），检测到index，当AB退出11相位时计数值设置为ARR（1次）

当index事件被执行时（而不是被检测到时），index中断标志置位。

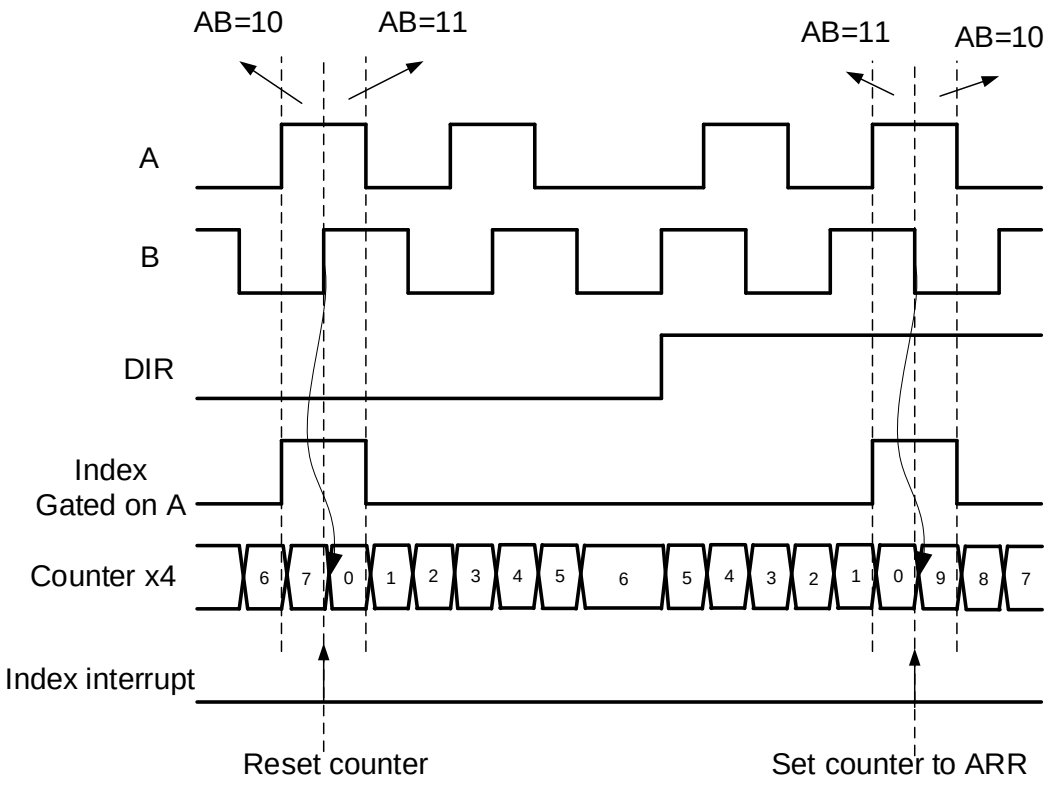


图26-59IPOS[1:0]=11, index gated on A, ARR=9

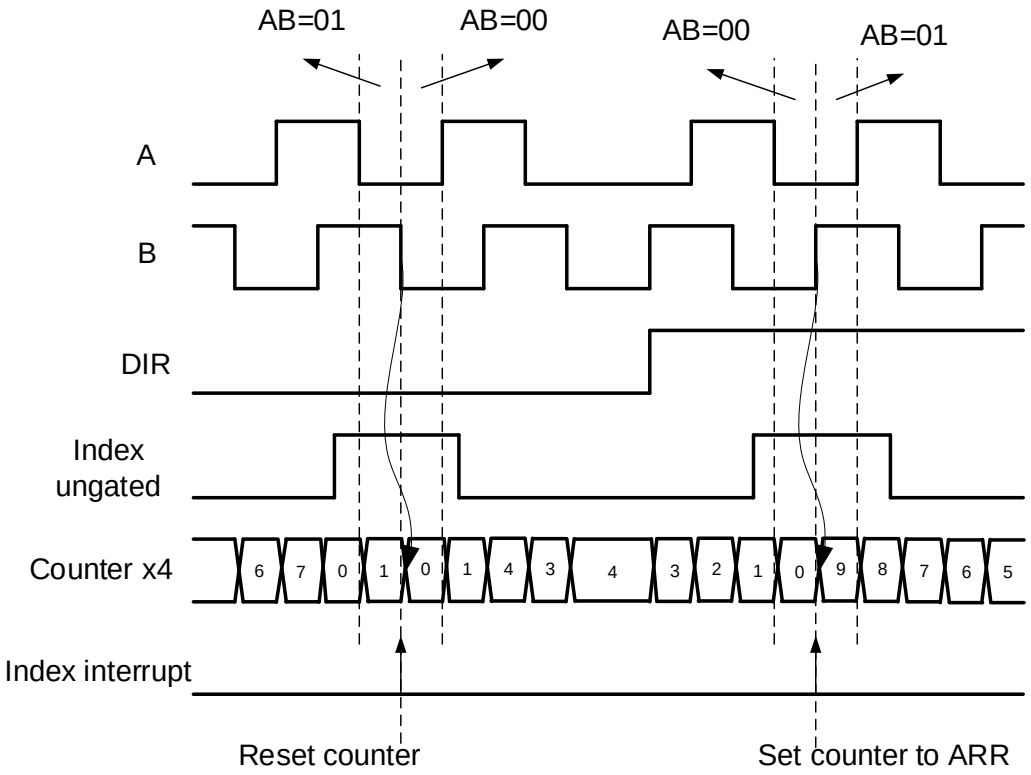


图26-60IPOS[1:0]=00, index ungated, ARR=9

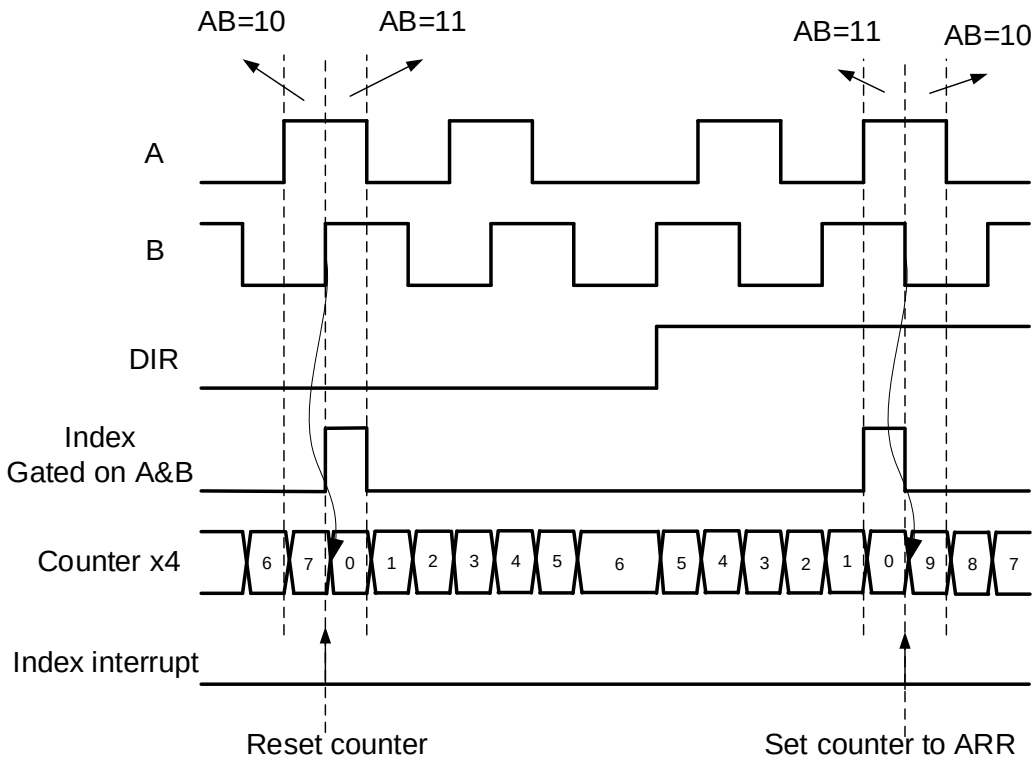


图26-61IPOS[1:0]=11, index gated on A&B, ARR=9

基于同步设计和状态转换记录，编码器模式的index检测允许一定的相位误差。比如下图中，index信号超前B相输入：

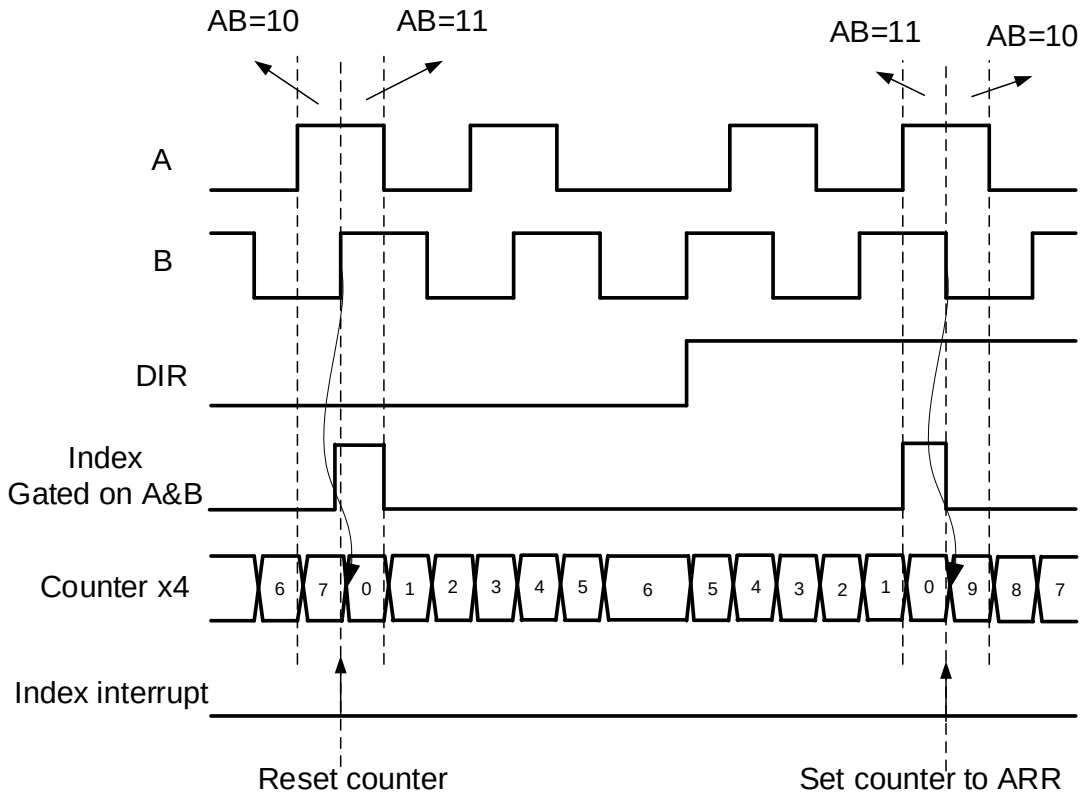


图26-62IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B

下图中则是index滞后B相输入，或者说出现一个窄Z相信号：

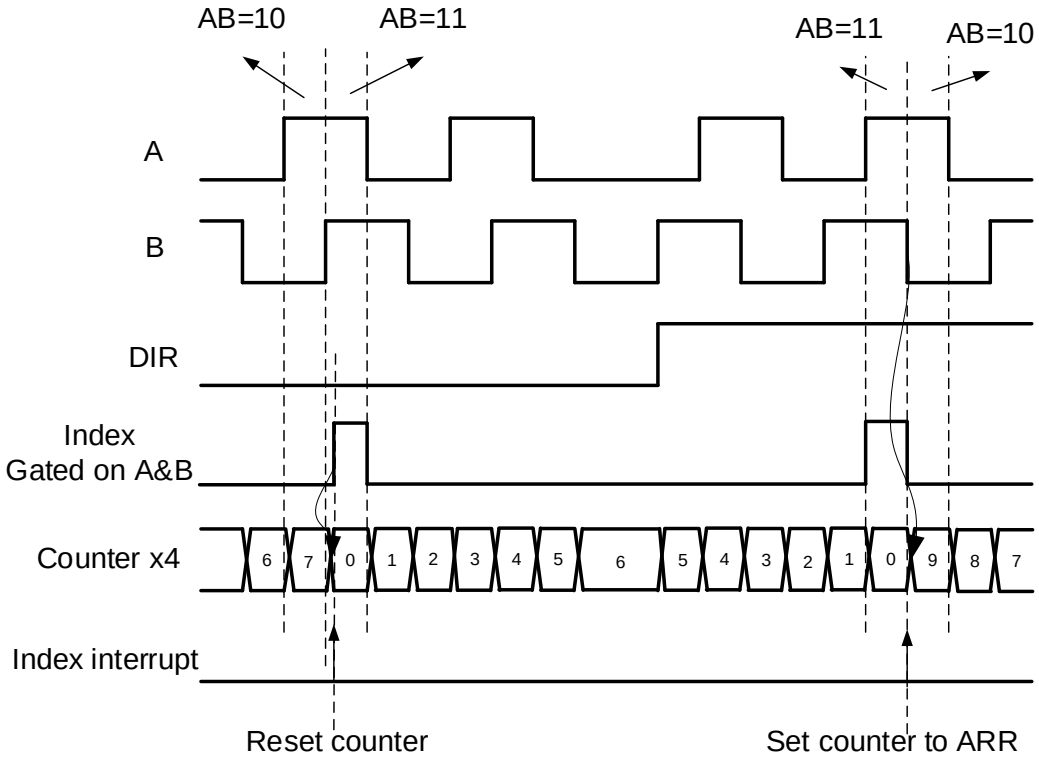


图26-63IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B

上图中index lagging的区域放大图如下：

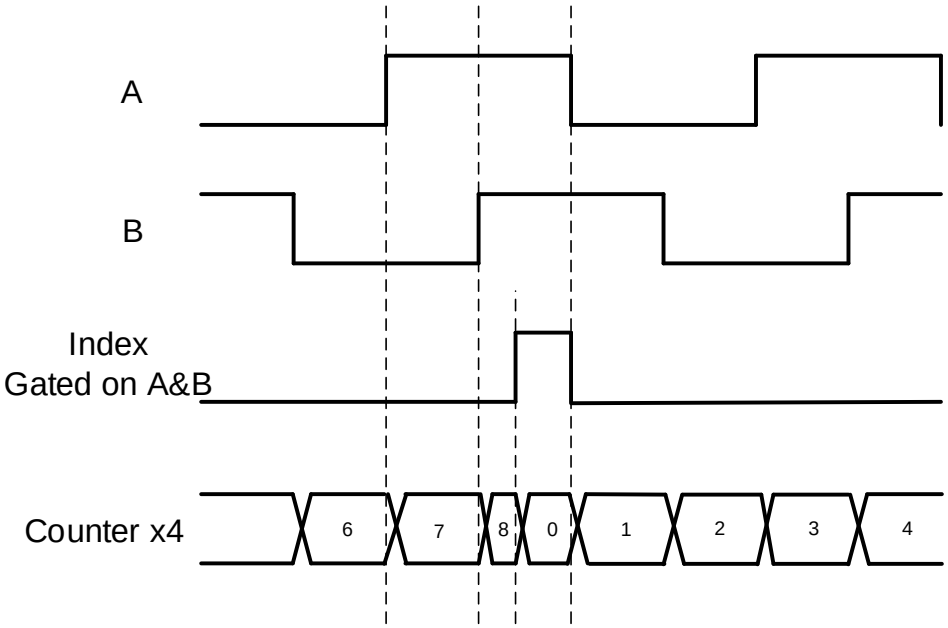


图26-64IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed

下图显示了counter mode x2 和x1的情况下，index如何影响计数值（IPOS[1:0]=01）：

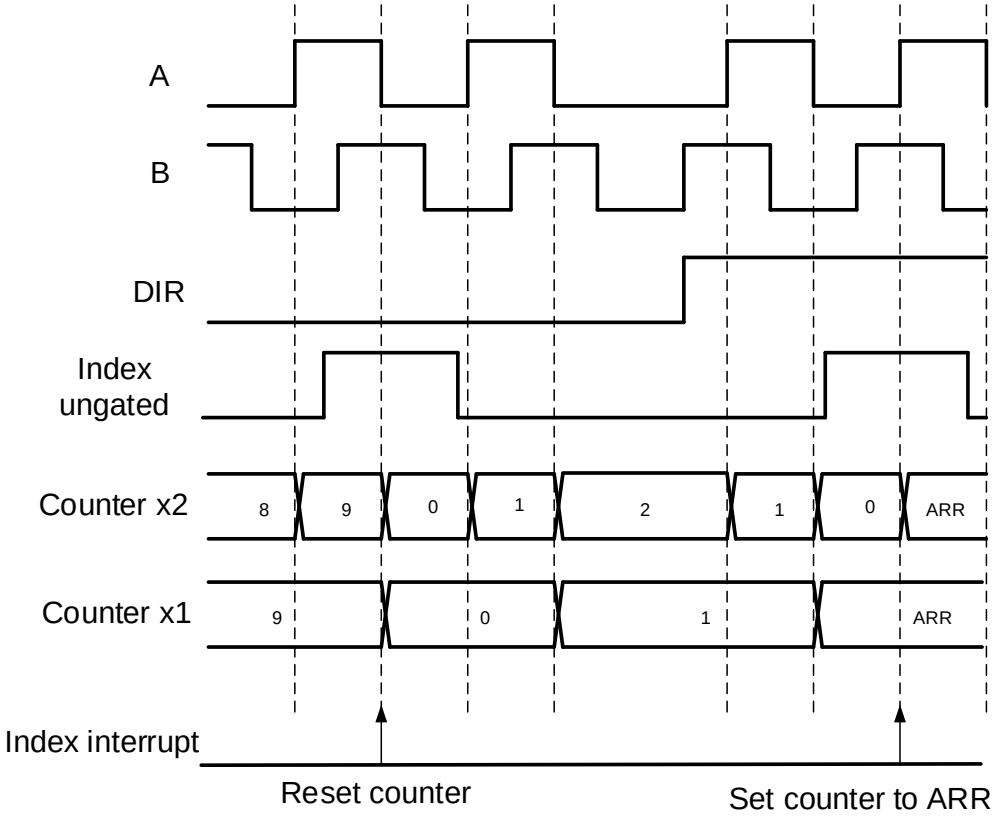


图26-65IPOS[1:0]=01, ungated index

Index的方向敏感性

可以通过ATIM_ECR寄存器中的IDIR[1:0]位配置定时器只在特定计数方向上响应index输入。

- IDIR=00：向上计数和向下计数时index都有效
- IDIR=01：向上计数时index有效
- IDIR=10：向下计数时index有效

注意：在时钟方向编码器和带方向信息的时钟编码器模式下，IDIR无效。

只响应第一个index

通过置位FIDX寄存器，可以使ATIM只响应第一个输入的index信号，后续index将被忽略，直到软件清零FIDX。

26.6.22.5 非正交编码模式下的 index 处理

当ATIM使用编码器接口模式，但是工作类型为非正交编码器，此时需要配置index信号在输入时钟信号的高电平处检测，或者在输入时钟信号的低电平处检测。通过IPOS[0]寄存器完成这个配置。

- IPOS[0]=0：在输入时钟低电平处检测index
- IPOS[0]=1：在输入时钟高电平处检测index

IPOS[1]此时不起作用。

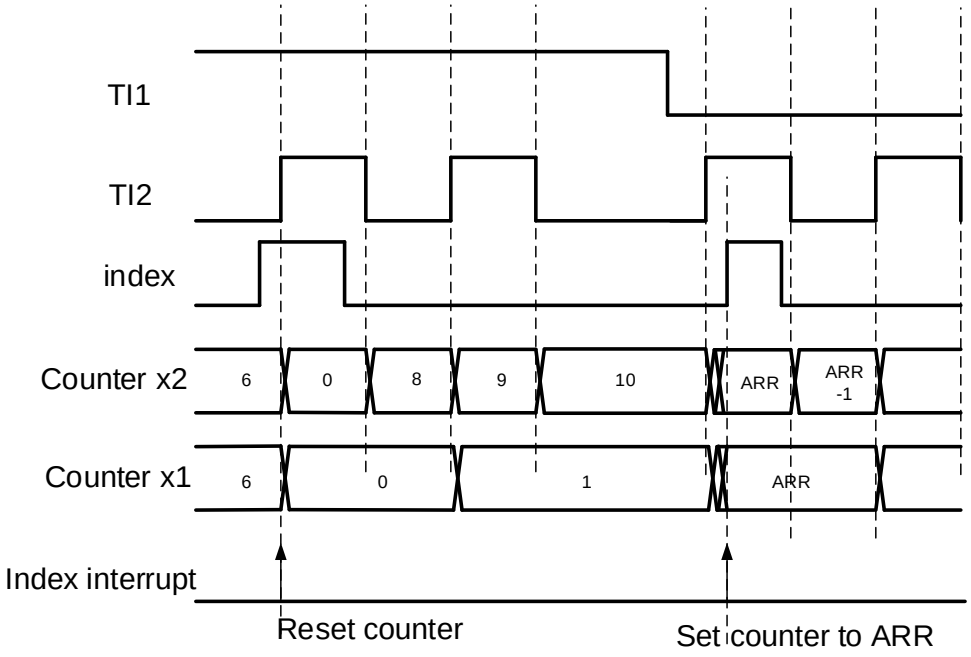


图26-66IPOS[0]=1, clock+direction mode

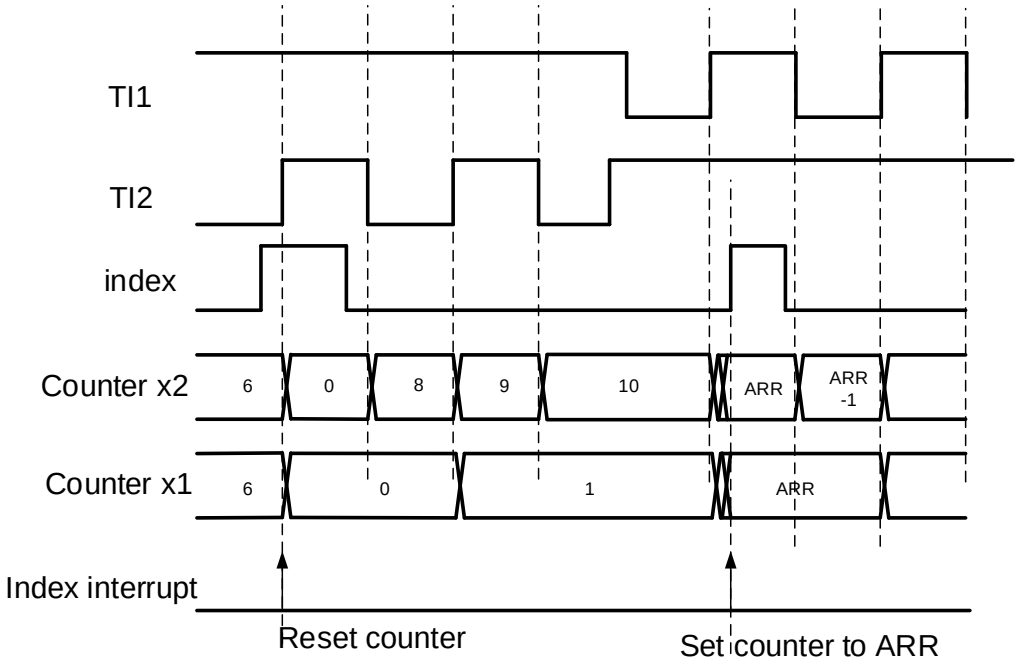


图26-67IPOS[0]=1, directional clock mode

26.6.22.6 正交编码模式错误管理

正交编码器输出的2路正交信号总是以格雷码形式变化，即相邻状态只会变化1bit。如果出现了2bit状态变化，则表示编码错误，此时TERRIF标志置位，并可以产生中断。

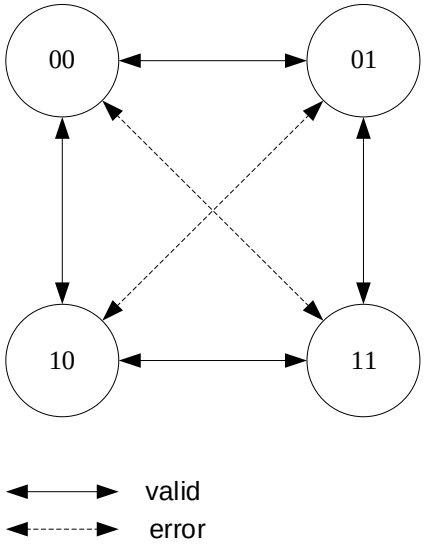


图26-68正交编码状态转移

当编码器计数溢出，而仍未检测到index事件时，表示出现了index错误。此时IERRF标志置位，并且可以产生中断。Index错误标志的置位要延迟到计数器roll-over，以适应窄Z相输入的情况，避免误报警。

下图的例子中，index在计数器等于0的周期内出现，错误标志不会置位。

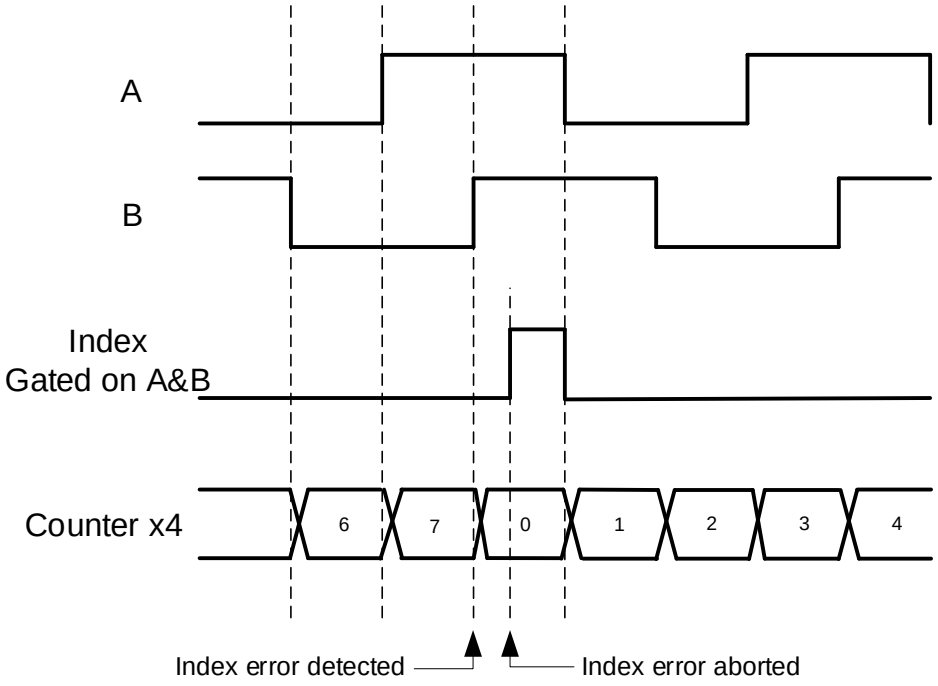


图26-69index错误撤销

下面的例子里，index直到计数器roll-over并且计数值大于0才到来，index错误标志置位。

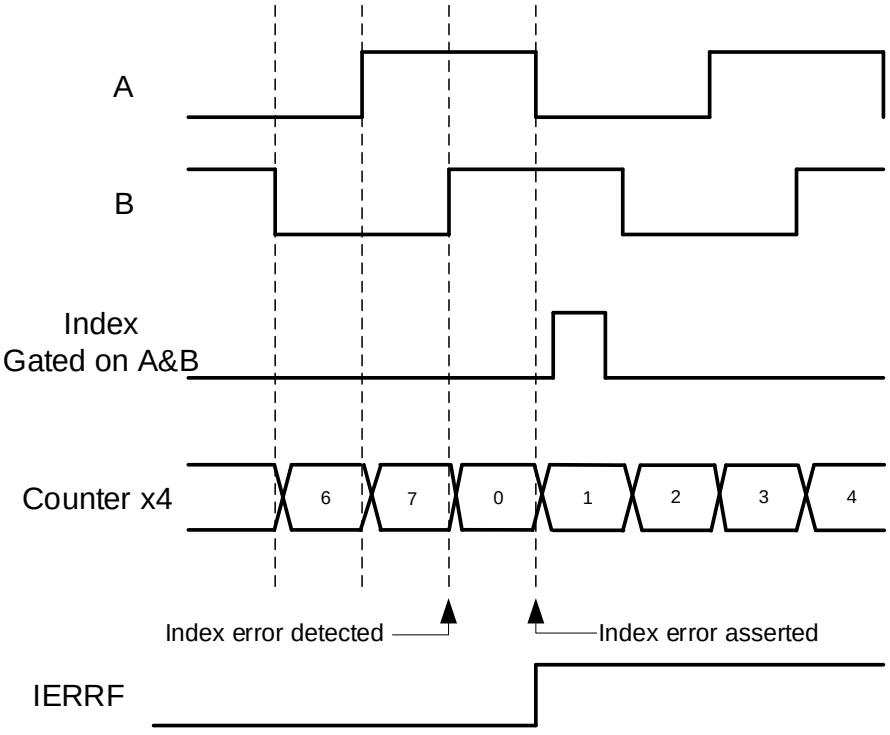


图26-70index错误置位

向下计数时，仅当计数值从1减到0，roll-over到ARR，然后递减时，才会置位index错误。

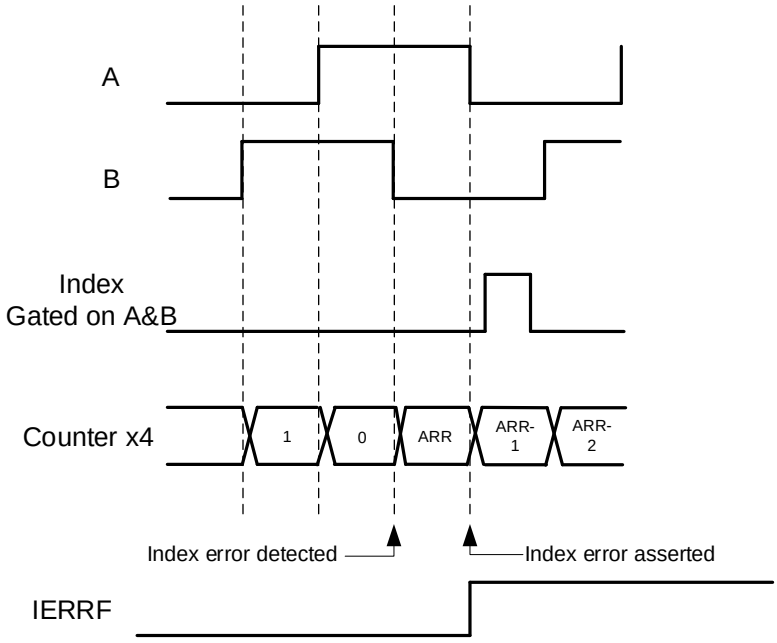


图26-71向下计数时index错误置位

26.6.22.7 Counter mode preload

某些应用中需要实时切换counter mode（x4, x2, x1），可以通过SMS寄存器的preload功能来实现。当SMSPE=1时，使能preload功能，并且根据SMSPS寄存器的值，选择在什么事件发生时进行preload：

- SMSPS=0：preload由update event(UEV)触发，即向上计数时计数器上溢出，或向下计数时计数器下溢出时触发，使用此配置时必须关闭index事件使能（IE=0）
- SMSPS=1：preload发生由index事件触发

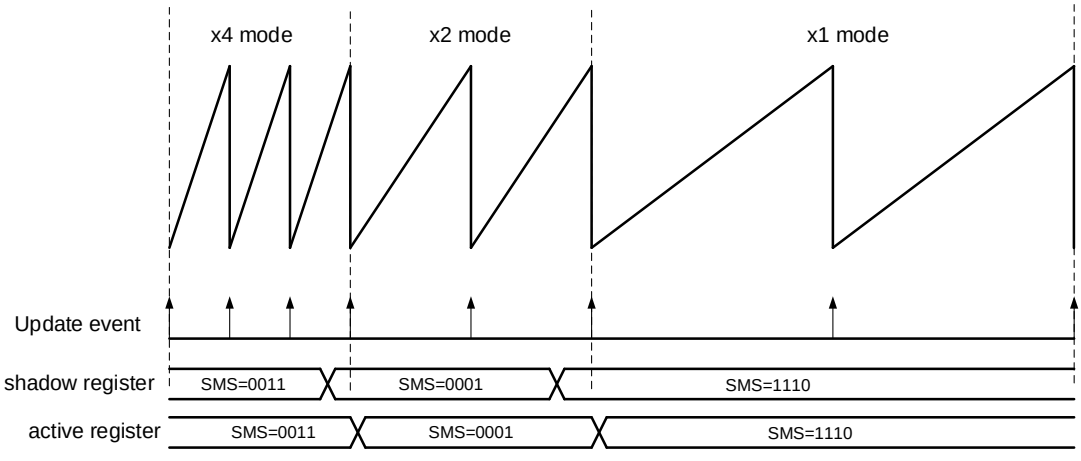


图26-72counter mode preload

26.6.22.8 编码器时钟输出

当SMS配置为编码器模式中的一种，配置MMS[3:0]=1000可以从TRGO输出被decode出来的编码器时钟。这个输出时钟是计数器实际计数频率，比如x4模式下这个时钟是正交编码信号频率的4倍。

26.6.23 定时器同步

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。

26.6.23.1 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

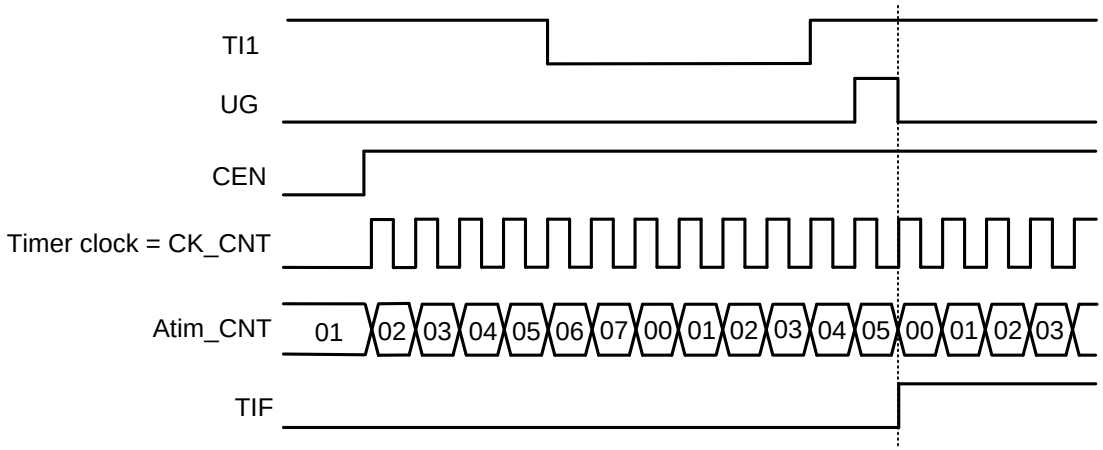


图26-73复位模式下的时序

26.6.23.2 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

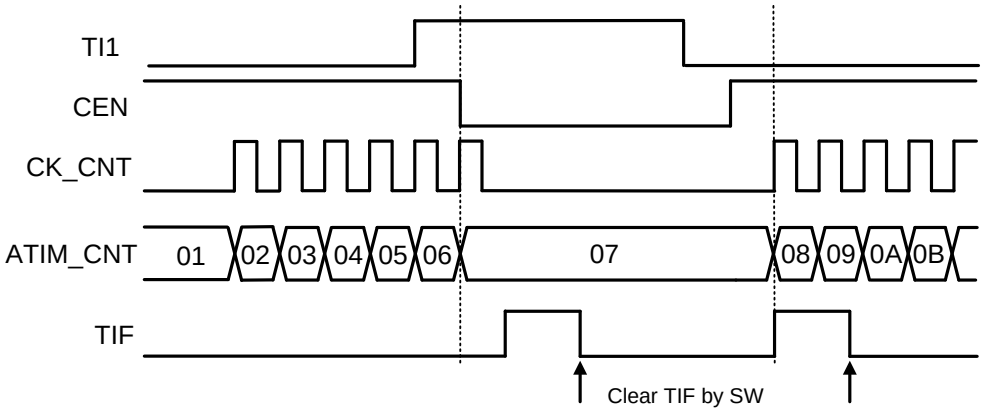


图26-74门控模式下的时序

使用门控模式可以实现软件同时启动多个定时器的计数，并且实现多个计数器计数值完全对齐（前提是不同定时器必须使用相同的工作时钟频率）。

配置master定时器的MMS=CEN，配置slave定时器的SMS=门控模式，并将slave定时器的ITR选择为master定时器的TRGO；此时，软件使能master定时器后，slave定时器也会被同时使能。

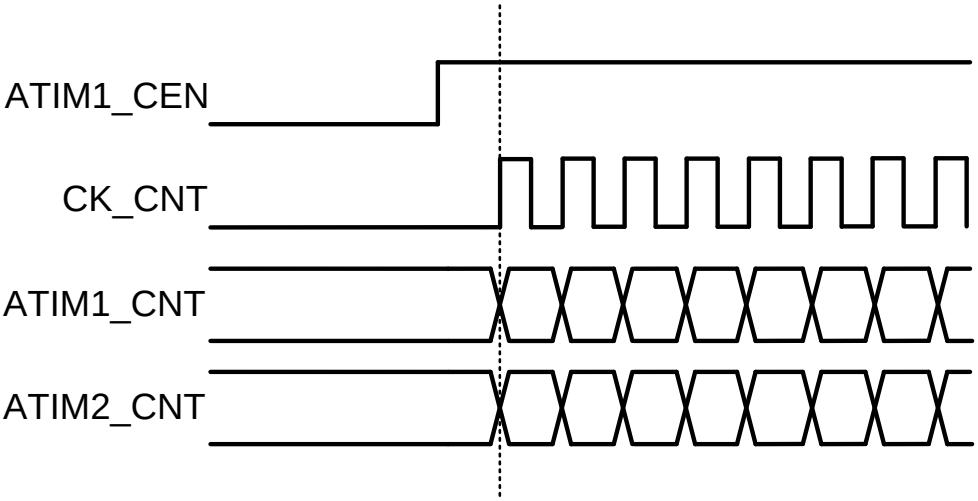


图26-75多timer同时使能

26.6.23.3 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

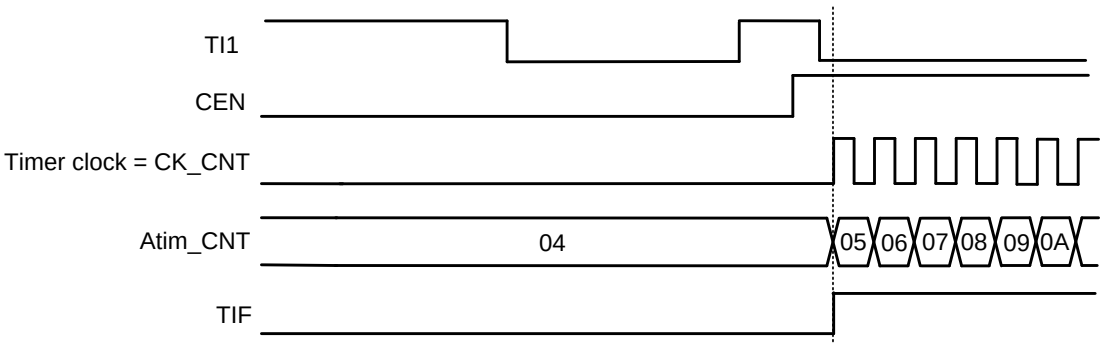


图26-76触发器模式下的时序

26.6.23.4 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

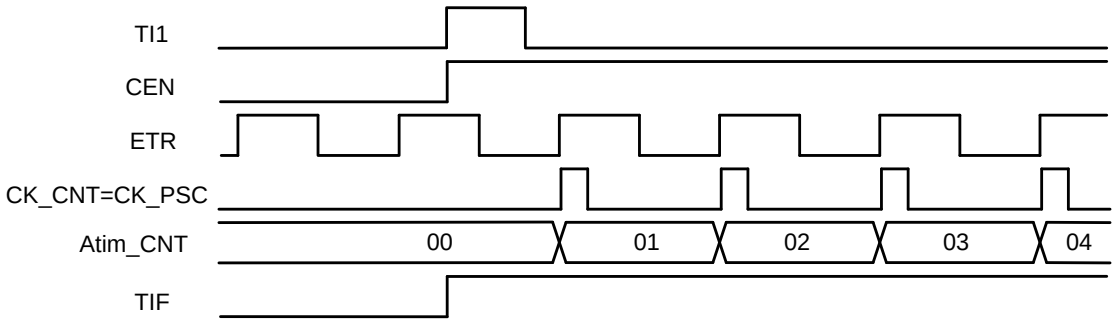


图26-77外部时钟模式2＋触发模式下的时序

26.6.24 触发信号输出

ATIM可以基于多种内部信号产生触发事件，包括定时器复位、定时器使能、或比较事件。同样有可能基于内部信号的边沿产生触发信号，甚至是多个内部信号边沿的组合。

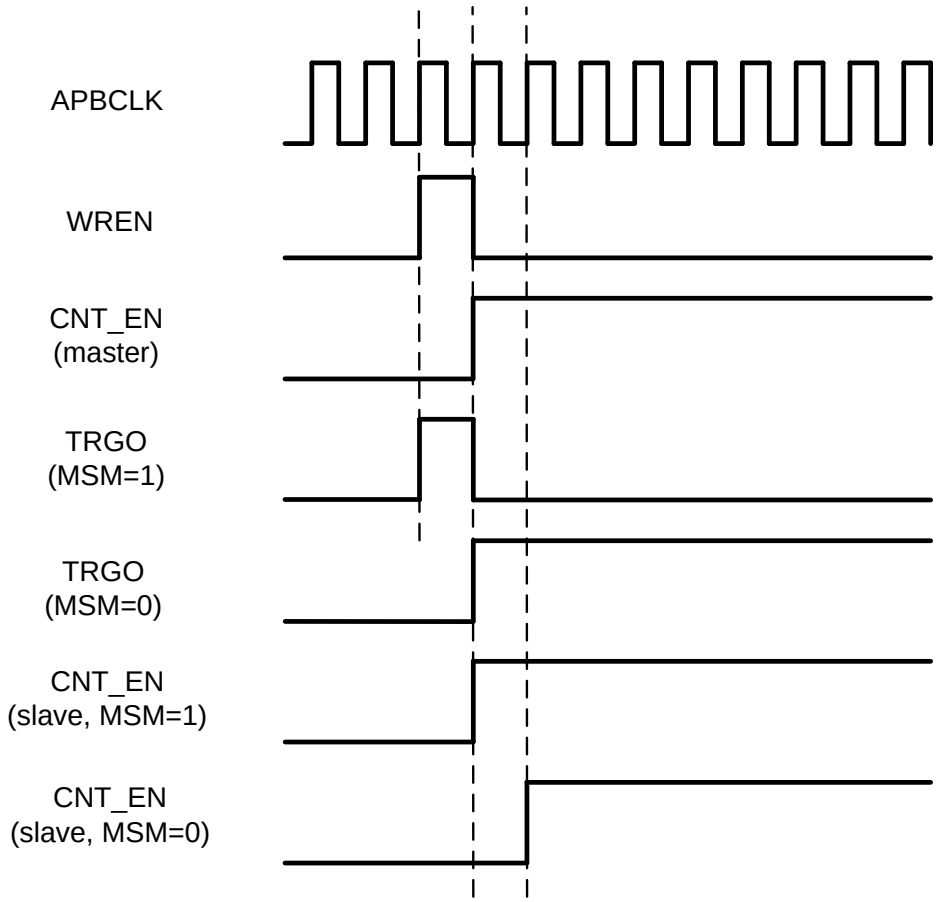
ADC触发信号从ATIM的TRGO和TRGO2端口输出，通过ATIM_CR2.MMS和ATIM_CR2.MMS2寄存器选择触发源。

TRGO输出组合配置：

MMS	TRGO

000	ATIM_EGR中的UG位被用作TRGO 当ATIM处于从机模式，且配置为reset mode时，触发信号输出导致计数器复位的同时，产生TRGO输出
001	ATIM的计数器使能信号被用作TRGO 此模式可用于同步启动多个定时器，将一个定时器配置为master，TRGO配置为使能信号，其他定时器配置为slave，当软件启动master定时器时，可以实现所有slave定时器同步启动【2】
010	更新事件（UEV）用作TRGO
011	CC1IF标志置位请求用作TRGO，注意即使CC1IF已经为高，置位脉冲仍可以输出TRGO
100	OC1REFC用作TRGO
101	OC2REFC用作TRGO
110	OC3REFC用作TRGO
111	OC4REFC用作TRGO

这里要特别注意主从定时器的严格同步设计。由于所有定时器EN位都是在APBCLK下，总线对主机EN的写操作，必须确保能够同时启动从机，这个功能要通过事先置位MSM寄存器实现



TRGO2输出组合配置：

MMS2	TRGO2

0000	ATIM_EGR中的UG位被用作TRGO2 当ATIM处于从机模式，且配置为reset mode时，触发信号输出导致计数器复位的同时，产生TRGO2输出
0001	ATIM的计数器使能信号被用作TRGO2 此模式可用于同步启动多个定时器，将一个定时器配置为master，TRGO2配置为使能信号，其他定时器配置为slave，当软件启动master定时器时，可以实现所有slave定时器同步启动
0010	更新事件（UEV）用作TRGO2
0011	CC1IF标志置位请求用作TRGO2，注意即使CC1IF已经为高，置位脉冲仍可以输出TRGO2
0100	OC1REFC用作TRGO2
0101	OC2REFC用作TRGO2
0110	OC3REFC用作TRGO2
0111	OC4REFC用作TRGO2
1000	OC5REFC用作TRGO2
1001	OC6REFC用作TRGO2
1010	OC4REFC的上升沿和下降沿用作TRGO2
1011	OC6REFC的上升沿和下降沿用作TRGO2
1100	OC4REFC和OC6REFC的上升沿用作TRGO2
1101	OC4REFC的上升沿和PC6TREFC的下降沿用作TRGO2
1110	OC5REFC和OC6REFC的上升沿用作TRGO2
1111	OC5REFC的上升沿和PC6TREFC的下降沿用作TRGO2

26.6.25 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	

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
	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	

26.6.26 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）。

26.6.27 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。

26.6.28 霍尔传感器接口

在高级定时器驱动马达的同时，可以通过使用另一个通用定时器作为接口定时器来连接霍尔传感器的输出。

接口定时器将3个输入信号XOR后送往通道1捕捉。接口定时器的slave mode controller设置为复位模式，将从机输入选为TI1F_ED，因此每当3个输入信号中任意一个发生翻转，定时器将重新从0开始计数。与此同时，将通道1配置为捕捉模式，捕捉信号为TRC，这样当输入信号翻转时，当前计数值被CC1通道捕捉，这个捕捉值记录了霍尔传感器信号翻转之间的间隔时间，可用于计算马达速度。

接口定时器的其他通道还可以配置为PWM输出模式，通过TRGO输出OCREF给高级定时器作为COM触发信号，可以用来触发高级定时器的PWM实时更新配置参数。

下图显示了GPTIM的CC1用于霍尔信号定时捕捉，CC2用于输出触发信号给高级定时器实现PWM参数（COM事件）。

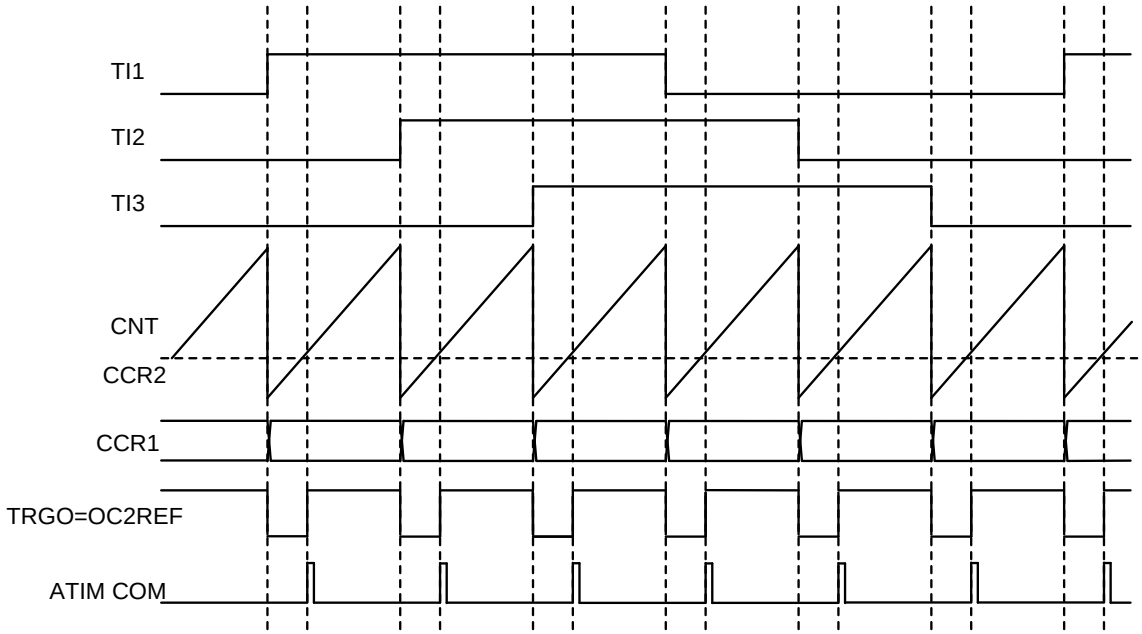


图26-78霍尔传感器接口

26.6.29 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DCU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

26.7 寄存器

ATIM模块基地址：0x40018000

offset 地址	名称	符号
0x00	ATIM 控制寄存器 1	ATIM_CR1
0x04	ATIM 控制寄存器 2	ATIM_CR2
0x08	ATIM 从机模式控制寄存器	ATIM_SMCR
0x0C	ATIM DMA 和中断使能寄存器	ATIM_IER
0x10	ATIM 状态寄存器	ATIM_ISR
0x14	ATIM 事件产生寄存器	ATIM_EGR
0x18	ATIM 捕捉/比较模式寄存器 1	ATIM_CCMR1
0x1C	ATIM 捕捉/比较模式寄存器 2	ATIM_CCMR2
0x20	ATIM 捕捉/比较使能寄存器	ATIM_CCER
0x24	ATIM 计数器寄存器	ATIM_CNT
0x28	ATIM 预分频寄存器	ATIM_PSC
0x2C	ATIM 自动重载寄存器	ATIM_ARR
0x30	ATIM 重复计数寄存器	ATIM_RCR
0x34	ATIM 捕捉/比较寄存器 1	ATIM_CCR1
0x38	ATIM 捕捉/比较寄存器 2	ATIM_CCR2
0x3C	ATIM 捕捉/比较寄存器 3	ATIM_CCR3
0x40	ATIM 捕捉/比较寄存器 4	ATIM_CCR4
0x44	ATIM 刹车和死区控制寄存器	ATIM_BDTR
0x48	ATIM 捕捉/比较模式寄存器 3	ATIM_CCMR3
0x4C	ATIM 捕捉/比较寄存器 5	ATIM_CCR5
0x50	ATIM 捕捉/比较寄存器 6	ATIM_CCR6
0x54	ATIM 编码模式控制寄存器	ATIM_ECR
0x58	ATIM 输入选择寄存器	ATIM_TISEL
0x5C	ATIM 辅助功能寄存器	ATIM_AFR
0x60	ATIM 刹车标志寄存器	ATIM_BRKFLAG
0x64	-	-
0x68	-	-
0x6C	-	-
0x70~0xF4	-	-
0xF8	ATIM DMA 控制寄存器	ATIM_DCR
0xFC	ATIM DMA 访问寄存器	ATIM_DMAR

26.7.1 ATIM 控制寄存器 1

名称	ATIM_CR1							
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	-				UIFREMAP	-	CKD	
位权限	U-0				RW-0	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
31:12	--	RFU：未实现，读为 0
11	**UIFREMAP**	UIF 标志位重映射 1：UIF 标志位被复制到 ATIM_CNT 寄存器的 bit31 0：无重映射功能
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 - 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

Bit	助记符	功能描述
		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

26.7.2 ATIM 控制寄存器 2

名称	ATIM_CR2							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						MMS[3]	-
位权限	U-0						RW-0	U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MMS2				-	OIS6	-	OIS5
位权限	RW-0000				U-0	RW-0	U-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
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	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
31:26	**--**	RFU：未实现，读为 0
25	**MMS[3]**	参见 MMS
24		
23:20	**MMS2**	主机模式选择 2，用于配置主机模式下向 ADC 发送的触发信号（TRGO2）源 0000：ATIM_EGR中的UG位被用作TRGO2；当ATIM处于从机模式，且配置为reset mode时，触发信号输出导致计数器复位的同时，产生TRGO2输出 0001：ATIM的计数器使能信号被用作TRGO2；此模式可用于同步启动多个定时器，将一个定时器配置为master，TRGO2配置为使能信号，其他定时器配置为slave，当软件启动master定时器时，可以实现所有slave定时器同步启动 0010：更新事件（UEV）用作 TRGO2 0011：CC1IF 标志置位请求用作 TRGO2，注意即使 CC1IF 已经为高，置位脉冲仍可以输出 TRGO2 0100：OC1REFC 用作 TRGO2 0101：OC2REFC 用作 TRGO2 0110：OC3REFC 用作 TRGO2 0111：OC4REFC 用作 TRGO2 1000：OC5REFC 用作 TRGO2 1001：OC6REFC 用作 TRGO2 1010：OC4REFC 的上升沿和下降沿用作 TRGO2 1011：OC6REFC 的上升沿和下降沿用作 TRGO2 1100：OC4REFC 和 OC6REFC 的上升沿用作 TRGO2

Bit	助记符	功能描述
		1101：OC4REFC 的上升沿和 OC6REFC 的下降沿用作 TRGO2 1110：OC5REFC 和 OC6REFC 的上升沿用作 TRGO2 1111：OC5REFC 的上升沿和 OC6REFC 的下降沿用作 TRGO2
19	--	RFU：未实现，读为 0
18	**OIS6**	参考 OIS1
17	--	RFU：未实现，读为 0
16	**OIS5**	参考 OIS1
15	**OIS4N**	参考 OIS1N
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**	与 MMS[3]组合成 4bit 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 0000：ATIM_EGR 的 UG 寄存器被用作 TRGO 0001：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 0010：UE（update event）信号被用作 TRGO 0011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 0100：OC1REFC 用作 TRGO 0101：OC2REFC 用作 TRGO 0110：OC3REFC 用作 TRGO 0111：OC4REFC 用作 TRGO 1000：编码器时钟输出，编码器模式的计数时钟信号从 TRGO 输出，仅当 SMS 配置为编码器模式中的一种时有效 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 1001：OC1REFC 下降沿用作 TRGO 1010：OC2REFC 下降沿用作 TRGO 1011：OC3REFC 下降沿用作 TRGO 1100：OC4REFC 下降沿用作 TRGO 1101：OC5REFC 下降沿用作 TRGO 1110：OC6REFC 下降沿用作 TRGO 1111：OC4REFC 的下降沿和 OC6REFC 的下降沿用作 TRGO

Bit	助记符	功能描述
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

26.7.3 ATIM 从机模式控制寄存器

名称	ATIM_SMCR							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						SMSPS	SMSPE
位权限	U-0						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		TS[4:3]		-			OCCS
位权限	U-0		RW-00		U-0			RW-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[2:0]			SMS			
位权限	RW-0	RW-000			RW-000			

Bit	助记符	功能描述
31:17	--	RFU：未实现，读为 0
25	**SMSPS**	SMS preload 触发源选择 0：SMS 更新由 updateevent 触发 1：SMS 更新由 indexevent 触发
24	**SMSPE**	SMSpreload 使能 0：禁止 SMSpreload 1：使能 SMSpreload
23:22	--	RFU：未实现，读为 0
21:20	**TS[4:3]**	参见 TS[2:0]
19:17	--	RFU：未实现，读为 0
16	**OCCS**	OCREF 清零选择 0：使用 OCREF_CLR 信号来清零 1：使用 ETRF 信号来清零
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[2:0]**	与 TS[4:3]组成 5bit 内部触发选择，用于选择同步计数器的触发源 00000：内部触发信号（ITR0） 00001：内部触发信号（ITR1） 00010：内部触发信号（ITR2） 00011：内部触发信号（ITR3） 00100：TI1 边沿检测（TI1F_ED） 00101：滤波后 TI1（TI1FP1） 00110：滤波后 TI2（TI2FP2） 00111：外部触发输入（ETRF） 01000：内部触发信号（ITR4） 01001：内部触发信号（ITR5） 01010：内部触发信号（ITR6） 01011：内部触发信号（ITR7） 01100：内部触发信号（ITR8） 01101：内部触发信号（ITR9） 01110：内部触发信号（ITR10） 01111：内部触发信号（ITR11）

Bit	助记符	功能描述
		10000：内部触发信号（ITR12） 10001：内部触发信号（ITR13） 10010：内部触发信号（ITR14） 10010：内部触发信号（ITR15） Others：RFU 注意：仅当 SMS=0000 即禁止从机模式的情况下，可以改写 TS 寄存器
3:0	**SMS**	从机模式选择 0000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 0001：Quadrature Encoder 模式 1（x1 mode）；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 0010：Quadrature Encoder 模式 2（x2 mode）；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 0011：Quadrature Encoder 模式 3（x4 mode）；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 0100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 0101：门控模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 0110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 0111：外部时钟模式 1；TRGI 上升沿驱动计数器 1000：组合的复位+触发模式；TRGI 上升沿初始化计数器，产生更新事件并启动计数 1001：组合的闸门+触发模式；TRGI 为高电平时，计数时钟使能，当 TRGI 变低电平时，计数器复位且停止 1010：Encoder 模式：时钟+方向，x2 1011：Encoder 模式：时钟+方向，x1 1100：Encoder 模式：带方向的时钟，x2 1101：Encoder 模式：带方向的时钟，x1 1110：Quadrature Encoder x1 mode，仅使用 TI1FP1 边沿计数，有效沿由 CC1P 选择 1111：Quadrature Encoder x1 mode，仅使用 TI2FP2 边沿计数，有效沿由 CC2P 选择 其他：RFU

26.7.4 ATIM DMA 和中断使能寄存器（ATIM_IER）

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	TERRIE	IERRIE	DIRIE	IDXIE	CC4BU RSTEN	CC3BU RSTEN	CC2BU RSTEN	CC1BU RSTEN
位权限	RW-0	RW-0	RW-0	RW-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
31:24	--	RFU：未实现，读为 0
23	**TERRIE**	状态转移错误中断使能，用于 encoder 模式，1 有效
22	**IERRIE**	Index 错误中断使能，用于 encoder 模式，1 有效
21	**DIRIE**	编码器方向变化中断使能，用于 encoder 模式，1 有效
20	**IDXIE**	Index 事件中断使能，用于 encoder 模式，1 有效
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）
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

Bit	助记符	功能描述
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

26.7.5 ATIM 状态寄存器（ATIM_ISR）

名称	ATIM_SR							
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	TERRIF	IERRIF	DIRIF	IDXIF	-		CC6IF	CC5IF
位权限	RW-0	RW-0	RW-0	RW-0	U-0		RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			CC4OF	CC3OF	CC2OF	CC1OF	B2IF
位权限	U-0			RW-0	RW-0	RW-0	RW-0	RW-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
31:24	**--**	RFU：未实现，读为 0
23	**TERRIF**	状态转移错误中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
22	**IERRIF**	Index 错误中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
21	**DIRIF**	方向变化中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
20	**IDXIF**	Index 事件中断标志，用于 encoder 模式，硬件置位，软件写

Bit	助记符	功能描述
		1 清零
19:18	--	RFU：未实现，读为 0
17	**CC6IF**	捕捉/比较通道 6 中断标志 参考 CC1IF
16	**CC5IF**	捕捉/比较通道 5 中断标志 参考 CC3IF
15:13	--	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	**B2IF**	刹车事件 2 中断标志，硬件置位，软件写 1 清零
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

26.7.6 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	-							B2G
位权限	U-0							W-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
31:9	--	RFU：未实现，读为 0
8	**B2G**	软件刹车 2，软件置位此寄存器产生刹车事件 2，硬件自动清零 置位此寄存器将导致 MOE 清零，B2IF 置位
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 置位此寄存器将导致 MOE 清零，BIF 置位
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。

26.7.7 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	-							OC2M[3]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							OC1M[3]
位权限	U-0							R/W-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
31:25	--	RFU：未实现，读为 0
24	**OC2M[3]**	参见 OC2M[2:0]
23:17	--	RFU：未实现，读为 0
16	**OC1M[3]**	参见 OC1M[1:0]
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE
14:12	**OC2M[2:0]**	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[2:0]**	与 OC1M[3]组成 4bit 输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 0000：OC1REF 冻结，输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 0001：CCR1=CNT 时，将 OC1REF 置高 0010：CCR1=CNT 时，将 OC1REF 置低 0011：CCR1=CNT 时，翻转 OC1REF 0100：OC1REF 固定为低（inactive） 0101：OC1REF 固定为高（active） 0110：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 0111：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 1000：可重触发的 OPM 模式 1 1001：可重触发的 OPM 模式 2 1010：RFU 1011：RFU 1100：组合 PWM 模式 1 1101：组合 PWM 模式 2 1110：非对称 PWM 模式 1 1111：非对称 PWM 模式 2
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

Bit	助记符	功能描述
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 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

Bit	助记符	功能描述
		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

26.7.8 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	-							OC4M[3]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							OC3M[3]
位权限	U-0							R/W-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
31:25	--	RFU：未实现，读为 0
24	**OC4M[3]**	参见 OC4M
23:17	--	RFU：未实现，读为 0
16	**OC3M[3]**	参见 OC3M
15	**OC4CE**	输出比较 4 清零使能，参考 OC1CE
14:12	**OC4M[2:0]**	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[2:0]**	与 OC3M[3]组成 4bit 输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为 请参考 OC1M[3:0]

Bit	助记符	功能描述
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

Bit	助记符	功能描述
		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	**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

26.7.9 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	-		CC6P	CC6E	-		CC5P	CC5E
位权限	U-0		RW-0	RW-0	U-0		RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
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	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
31:22	**--**	RFU：未实现，读为 0
21	**CC6P**	捕捉/比较 6 输出极性，参考 CC1P
20	**CC6E**	捕捉/比较 6 输出使能，参考 CC1E
19:18	**--**	RFU：未实现，读为 0
17	**CC5P**	捕捉/比较 6 输出极性，参考 CC1P
16	**CC5E**	捕捉/比较 6 输出使能，参考 CC1E
15	**CC4NP**	捕捉/比较 4 互补输出极性，参考 CC1NP
14	**CC4NE**	捕捉/比较 4 互补输出使能，参考 CC1NE
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

Bit	助记符	功能描述
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 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

26.7.10 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	UIFCPY	-						
位权限	R-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	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	**UIFCPY**	UIF copy 当 UIFREMAP=1，UIF 寄存器的值被实时复制到这里，只读
30:16	**--**	RFU：未实现，读为 0
15:0	**CNT**	计数器值

26.7.11 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

26.7.12 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

26.7.13 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	**RCR**	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

26.7.14 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

26.7.15 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

26.7.16 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

Bit	助记符	功能描述
		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

26.7.17 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

26.7.18 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	-						BK2P	BK2E
位权限	U-0						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BK2F				BKF			
位权限	RW-0000				RW-0000			
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
31:26	--	RFU：未实现，读为 0
25	**BK2P**	Break2 极性 0：BRK2 低有效 1：BRK2 高有效 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
24	**BK2E**	Break2 使能 0：BRK2 关闭 1：BRK2 使能 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
23:20	**BK2F**	Break2 滤波，定义 BRK2 信号滤波长度 0000：无数字滤波 0001：采样时钟=f_{CK_INT}, N=2 0010：采样时钟=f_{CK_INT}, N=4 0011：采样时钟=f_{CK_INT}, N=8 0100：采样时钟=f_{DTS}, N=12 0101：采样时钟=f_{DTS}, N=16 0110：采样时钟=f_{DTS}, N=24 0111：采样时钟=f_{DTS}, N=32 1000：采样时钟=f_{DTS}, N=48 1001：采样时钟=f_{DTS}, N=64 1010：采样时钟=f_{DTS}, N=80 1011：采样时钟=f_{DTS}, N=96 1100：采样时钟=f_{DTS}, N=128 1101：采样时钟=f_{DTS}, N=160 1110：采样时钟=f_{DTS}, N=192 1111：采样时钟=f_{DTS}, N=256 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
19:16	**BKF**	Break 滤波，定义 BRK 信号滤波长度 0000：无数字滤波 0001：采样时钟=f_{CK_INT}, N=2 0010：采样时钟=f_{CK_INT}, N=4 0011：采样时钟=f_{CK_INT}, N=8 0100：采样时钟=f_{DTS}, N=12 0101：采样时钟=f_{DTS}, N=16 0110：采样时钟=f_{DTS}, N=24 0111：采样时钟=f_{DTS}, N=32 1000：采样时钟=f_{DTS}, N=48 1001：采样时钟=f_{DTS}, N=64 1010：采样时钟=f_{DTS}, N=80 1011：采样时钟=f_{DTS}, N=96 1100：采样时钟=f_{DTS}, N=128 1101：采样时钟=f_{DTS}, N=160 1110：采样时钟=f_{DTS}, N=192 1111：采样时钟=f_{DTS}, N=256 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
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

Bit	助记符	功能描述
		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：当响应 break2 事件时，关闭 OC 和 OCN 输出； 当响应 break 事件时，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 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 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
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 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
12	**BKE**	刹车使能 0：禁止刹车输入 1：允许刹车输入 注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
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 注意：当 LOCK 寄存器配置为 level2 之后此寄存器不能改写
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 注意：当 LOCK 寄存器配置为 level2 之后此寄存器不能改写
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}

26.7.19 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。

26.7.20 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

26.7.21 ATIM 捕捉/比较模式寄存器 3（ATIM_CCMR3）

名称	ATIM_CCMR3							
地址	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							OC6M[3]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							OC5M[3]
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC6CE	OC6M			OC6PE	OC6FE	-	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC5CE	OC5M			OC5PE	OC5FE	-	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	U-0	

Bit	助记符	功能描述
31:25	**--**	RFU：未实现，读为 0
24	**OC6M[3]**	参见 OC6M
23:17	**--**	RFU：未实现，读为 0
16	**OC5M[3]**	参见 OC5M
15	**OC6CE**	输出比较 6 清零使能，参考 OC1CE
14:12	**OC6M[2:0]**	输出比较 6 模式配置，参考 OC1M
11	**OC6PE**	输出比较 6 预装载使能，参考 OC1PE
10	**OC6FE**	输出比较 6 快速使能，参考 OC1FE
9:8	**--**	RFU：未实现，读为 0
7	**OC5CE**	输出比较 5 清零使能 0：OC5REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC3REF
6:4	**OC5M[2:0]**	与 OC5M[3]组成 4bit 输出比较 5 模式配置，此寄存器定义 OC5REF 信号的行为 请参考 OC1M[3:0]
3	**OC5PE**	输出比较 5 预装载使能 0：CCR5 preload 寄存器无效，CCR5 可以直接写入 1：CCR5 preload 寄存器有效，针对 CCR5 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	**OC5FE**	输出比较 5 快速使能

Bit	助记符	功能描述
		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 OC5REF 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	--	RFU：未实现，读为 0

26.7.22 ATIM 比较寄存器 5（ATIM_CCR5）

名称	ATIM_CCR5							
地址	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GC5C3	GC5C2	GC5C1	-				
位权限	RW-0	RW-0	RW-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	CCR5[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR5[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31	**GC5C3**	Group Channel 5 and Channel 3 0：OC3REFC 不受 OC5REF 影响 1：OC3REFC=OC3REF& OC5REF
30	**GC5C2**	Group Channel 5 and Channel 2 0：OC2REFC 不受 OC5REF 影响 1：OC2REFC=OC2REF& OC5REF
29	**GC5C1**	Group Channel 5 and Channel 1 0：OC1REFC 不受 OC5REF 影响 1：OC1REFC=OC1REF& OC5REF
28:16	--	RFU：未实现，读为 0
15:0	**CCR5**	比较通道 5 寄存器 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 OC5 输出

26.7.23 ATIM 比较寄存器 6（ATIM_CCR6）

名称	ATIM_CCR6							
地址	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	CCR6[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR6[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	**CCR6**	比较通道 6 寄存器 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 OC6 输出

26.7.24 ATIM 编码器模式控制寄存器（ATIM_ECR）

名称	**ATIM_ECR**							
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								
位权限								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IPOS		FIDX			IDIR		IE
位权限	RW-00		RW-0			RW-00		RW-0

Bit	助记符	功能描述
31:25	--	RFU：未实现，读为 0
7:6	**IPOS**	Index positioning 定义正交编码模式时 index 事件复位或置位计数器的相位 仅 SMS=0001/0010/0011/1110/1111 时有效 00：操作相位 AB=00 01：操作相位 AB=01 10：操作相位 AB=10 11：操作相位 AB=11 非正交编码器模式时，IPOS[0]用于定义在什么时钟电平时复位计数器 x0：时钟低电平时复位计数器 x1：时钟高电平时复位计数器
5	**FIDX**	First Index 1：仅第一个 index 有效 0：所有 index 有效

Bit	助记符	功能描述
4:3		
2:1	**IDIR**	Index 方向 00：index 在任意计数方向时复位计数器 01：index 仅在向上计数时复位计数器 10：index 仅在向下计数时复位计数器 11：RFU
0	**IE**	Index 使能 0：禁止 index 功能 1：使能 index 功能

26.7.25 ATIM 输入选择寄存器（ATIM_TISEL）

名称	ATIM_TISEL							
地址	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				TI4SEL			
位权限	U-0				RW-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				TI3SEL			
位权限	U-0				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TI2SEL			
位权限	U-0				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TI1SEL			
位权限	U-0				RW-0000			

Bit	助记符	功能描述
31:28	**--**	RFU：未实现，读为 0
27:24	**TI4SEL**	ti4_in[15:0]输入选择 0000: ATIM_CH4 0001: ti4_in1 …. 1111: ti4_in15
23:20	**--**	RFU：未实现，读为 0
19:16	**TI3SEL**	ti3_in[15:0]输入选择 0000: ATIM_CH3 0001: ti3_in1 …. 1111: ti3_in15
15:12	**--**	RFU：未实现，读为 0
11:8	**TI2SEL**	ti2_in[15:0]输入选择 0000: ATIM_CH2 0001: ti2_in1 …. 1111: ti2_in15
7:4	**--**	RFU：未实现，读为 0
3:0	**TI1SEL**	ti1_in[15:0]输入选择

Bit	助记符	功能描述
		0000: ATIM_CH1 0001: ti1_in1 …. 1111: ti1_in15

26.7.26 ATIM 辅助功能寄存器（ATIM_AFR）

名称	ATIM_AFR							
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	ETRSEL				-	OCCLR_SEL		
位权限	R/W-0000				U-0	RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				BKSYS3E	BKSYS2E	BKSYS1E	BKSYS0E
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	-		-	BKCMP3E	BKCMP2E	BKCMP1E	BKIN2E	BKIN1E
位权限	U-0		U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:20	**ETRSEL**	ETR 输入信号选择 0000：ATIM_ETR 0001：ATIM_ETR_IN1 … 1111：ATIM_ETR_IN15
19	--	RFU：未实现，读为 0
18:16	**OCCLR_SEL**	OCREF_CLR 输入选择，用于选择使用什么片内信号产生 OCREF 清零信号 000：COMP0_OUT 001：COMP1_OUT 010：COMP2_OUT 其他：RFU
15:12	--	RFU：未实现，读为 0
11	**BKSYS3E**	系统刹车信号 3 使能，1 有效 接 RAM 校验结果输出
10	**BKSYS2E**	系统刹车信号 2 使能，1 有效 接 LOCKUP
9	**BKSYS1E**	系统刹车信号 2 使能，1 有效 接 SVD 输出
8	**BKSYS0E**	系统刹车信号 2 使能，1 有效 接 HFDET 输出
7:6	--	RFU：未实现，读为 0

Bit	助记符	功能描述
5	--	RFU：未实现，读为 0
4	**BKCMP3E**	比较器 3 输出刹车使能，1 有效
3	**BKCMP2E**	比较器 2 输出刹车使能，1 有效
2	**BKCMP1E**	比较器 1 输出刹车使能，1 有效
1	**BKIN2E**	管脚刹车信号 2 使能，1 有效
0	**BKIN1E**	管脚刹车信号 1 使能，1 有效

26.7.27 ATIM 刹车标志寄存器（ATIM_BRKFLAG）

名称	ATIM_BRKFLAG							
地址	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				BKSYS3_FLAG	BKSYS2_FLAG	BKSYS1_FLAG	BKSYS0_FLAG
位权限	U-0000				W1C-0	W1C-0	W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		BKCMP4_FLAG	BKCMP3_FLAG	BKCMP2_FLAG	BKCMP1_FLAG	BKIN2_FLAG	BKIN1_FLAG
位权限	U-00		W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

位	位名	功能描述
31:12	--	RFU，未实现，读为 0
11	**BKSYS3_FLAG**	系统刹车信号 3 标志，写 1 清零
10	**BKSYS2_FLAG**	LOCKUP 刹车信号标志，写 1 清零
9	**BKSYS1_FLAG**	SVD 刹车信号标志，写 1 清零
8	**BKSYS0_FLAG**	高频晶振停震刹车标志，写 1 清零
7:6	--	RFU，未实现，读为 0
5	**BKCMP4_FLAG**	比较器刹车信号 4 刹车标志，写 1 清零
4	**BKCMP3_FLAG**	比较器刹车信号 3 刹车标志，写 1 清零
3	**BKCMP2_FLAG**	比较器刹车信号 2 刹车标志，写 1 清零
2	**BKCMP1_FLAG**	比较器刹车信号 1 刹车标志，写 1 清零
1	**BKIN2_FLAG**	外部刹车信号 2 刹车标志，写 1 清零
0	**BKIN1_FLAG**	外部刹车信号 1 刹车标志，写 1 清零

26.7.28 DMA Burst 控制寄存器（ATIM_DCR）

名称	ATIM_DCR							
地址	0xF8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			DBL				
位权限	U-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			DBA				
位权限	U-000			RW-00000				

位	位名	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 DBL+1,注意不要超出寄存器地址范围
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。

26.7.29 DMA burst 访问寄存器（ATIM_DMAR）

名称	ATIM_DMAR							
地址	0xFC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR							
位权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR							
位权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR							
位权限	RW-00000000							

名称	ATIM_DMAR							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR							
位权限	RW-00000000							

位	位名	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

27 通用定时器（GPTIM）

27.1 概述

通用定时器包含一个16bit自动重载计数器及一个可编程预分频器。

通用定时器可以支持多种应用，包括如捕捉、输出比较、PWM。

27.2 主要特性

- 16bit向上、向下、双向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
- 4个独立通道可用于输入捕捉、输出比较、PWM（边缘或中心对齐模式）、单脉冲输出
- 支持与其他定时器级联
- 支持在以下事件发生时产生中断

  - 计数器溢出，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

27.3 结构框图

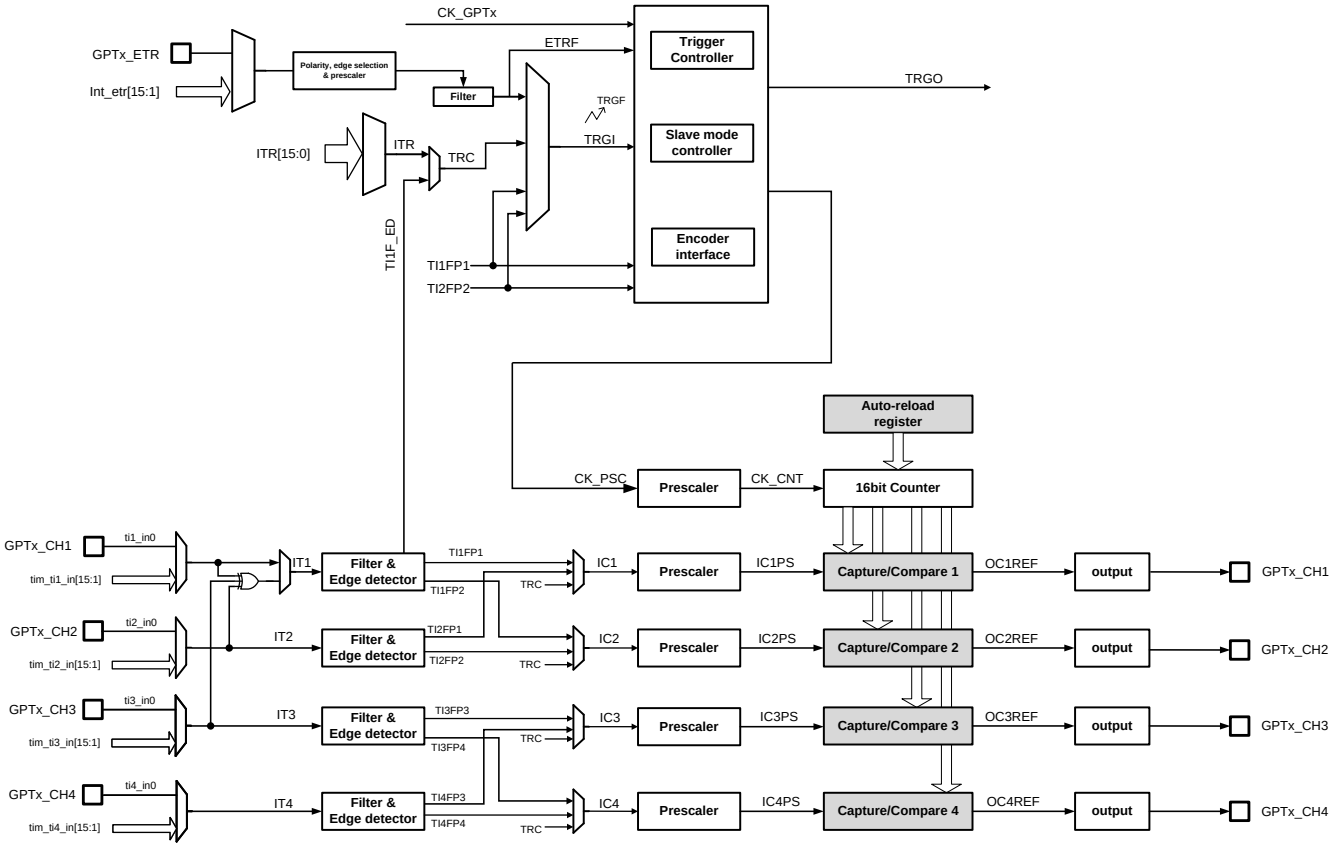


图 27-1 通用定时器架构示意图

27.4 引脚和内部连接

Pin name	direction	description
GPTx_CH1 GPTx_CH2 GPTx_CH3 GPTx_CH4	Input/output	GPTIM输入输出通道，可用于PWM输出或输入信号捕获 CH1和CH2也可用作外部计数时钟输入，外部触发或者正交编码器输入 CH1、CH2、CH3可以用作数字霍尔传感器接口
GPTx_ETR	input	外部触发信号输入，或外部计数时钟输入

表 27-1GPTIM 输入输出引脚

signal name	type	Description
ITR[15:0]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）

Int_etr[15:1]	input	外部触发输入，可用于触发信号或者外部计数时钟 在使能了输入预分频的情况下，可以输入高于 CK_GTIM 频率的计数时钟
Tim_ti1_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti2_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti3_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti4_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
TRGO	output	内部触发信号输出，用于连接到其他 timer 或外设
APBCLK	input	寄存器总线时钟
GTIM_CC1_DMA GTIM_CC2_DMA GTIM_CC3_DMA GTIM_CC4_DMA	output	4 个通道各自的 DMA 请求输出
GTIM_UEV_DMA	output	更新事件 DMA 请求

表 27-2GPTIM 内部输入输出信号

ITRx	GTIM0	GTIM1
ITR0	ATIM0_TRGO	ATIM0_TRGO
ITR1	ATIM0_TRGO2	ATIM0_TRGO2
ITR2	-	-
ITR3	-	-
ITR4	-	GPTIM0_TRGO
ITR5	GPTIM1_TRGO	-
ITR6	-	-
ITR7	BSTIM16_TRGO	BSTIM16_TRGO
ITR8	LPTIM16_TRGO	LPTIM16_TRGO
ITR9	-	-
ITR10	-	-
ITR11	-	-
ITR12	-	-
ITR13	-	-
ITR14	-	-
ITR15	-	-

表 27-3GPTIM 内部触发信号连接

ETRx	GTIM0	GTIM1
ETR0	GTIM0_ETR	GTIM1_ETR
ETR1	COMP1_OUT	COMP1_OUT
ETR2	COMP2_OUT	COMP2_OUT
ETR3	COMP3_OUT	COMP3_OUT
ETR4	COMP4_OUT	COMP4_OUT
ETR5	-	-
ETR6	-	-

ETR7	-	-
ETR8	-	-
ETR9	-	-
ETR10	-	-
ETR11	-	-
ETR12	-	-
ETR13	-	-
ETR14	-	-
ETR15	-	-

表 27-4GPTIM 外部触发信号连接

GTIMx_CH1	GTIM0	GTIM1
TI1_IN 0	GTIM0_CH1	GTIM1_CH1
TI1_IN 1	XTHF	XTHF
TI1_IN 2	RCLP	RCLP
TI1_IN 3	XTLF	XTLF
TI1_IN 4	RCHF	RCHF
TI1_IN 5	-	-
TI1_IN 6	-	-
TI1_IN 7	-	-
TI1_IN 8	-	-
TI1_IN 9	-	-
TI1_IN 10	-	-
TI1_IN 11	-	-
TI1_IN 12	-	-
TI1_IN 13	-	-
TI1_IN 14	-	-
TI1_IN 15	-	-

表 27-5GPTIM 通道 1 信号连接

GTIMx_CH2	GTIM0	GTIM1
TI2_IN 0	GTIM0_CH2	GTIM1_CH2
TI2_IN 1	BSTIM16_TRGO	BSTIM16_TRGO
TI2_IN 2	LPTIM16_TRGO	LPTIM16_TRGO
TI2_IN 3	ATIM0_TRGO	ATIM0_TRGO
TI2_IN 4	ATIM0_TRGO2	ATIM0_TRGO2
TI2_IN 5	-	GTIM0_TRGO
TI2_IN 6	GPTIM1_TRGO	-
TI2_IN 7	-	-
TI2_IN 8	-	-
TI2_IN 9	-	-
TI2_IN 10	-	-
TI2_IN 11	-	-
TI2_IN 12	-	-
TI2_IN 13	-	-
TI2_IN 14	-	-
TI2_IN 15	-	-

表 27-6GPTIM 通道 2 信号连接

GTIMx_CH3	GTIM0	GTIM1
TI3_IN 0	GTIM0_CH3	GTIM1_CH3
TI3_IN 1	UART0_RX	UART0_RX
TI3_IN 2	UART1_RX	UART1_RX
TI3_IN 3	UART2_RX	UART2_RX
TI3_IN 4	UART3_RX	UART3_RX
TI3_IN 5	UART4_RX	UART4_RX
TI3_IN 6	-	-
TI3_IN 7	-	-
TI3_IN 8	COMP1_OUT	COMP1_OUT
TI3_IN 9	COMP2_OUT	COMP2_OUT
TI3_IN 10	COMP3_OUT	COMP3_OUT
TI3_IN 11	COMP4_OUT	COMP4_OUT
TI3_IN 12	-	-
TI3_IN 13	-	-
TI3_IN 14	-	-
TI3_IN 15	-	-

表 27-7GPTIM 通道 3 信号连接

GTIMx_CH4	GTIM0	GTIM1
TI4_IN 0	GTIM0_CH4	GTIM1_CH4
TI4_IN 1	UART0_RX	UART0_RX
TI4_IN 2	UART1_RX	UART1_RX
TI4_IN 3	UART2_RX	UART2_RX
TI4_IN 4	UART3_RX	UART3_RX
TI4_IN 5	UART4_RX	UART4_RX
TI4_IN 6	-	-
TI4_IN 7	-	-
TI4_IN 8	-	-
TI4_IN 9	LUT0_OUT	LUT0_OUT
TI4_IN 10	LUT1_OUT	LUT1_OUT
TI4_IN 11	LUT2_OUT	LUT2_OUT
TI4_IN 12	LUT3_OUT	LUT3_OUT
TI4_IN 13	-	-
TI4_IN 14	-	-
TI4_IN 15	-	-

表 27-8GPTIM 通道 4 信号连接

27.5 功能描述

27.5.1 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（GPTIM_CNT）
- 预分频寄存器（GPTIM_PSC）
- 自动重载寄存器（GPTIM_ARR）

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（GPTIM_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）。

GPTIM_CNT工作时钟由GPTIM_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。

GPTIM_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。

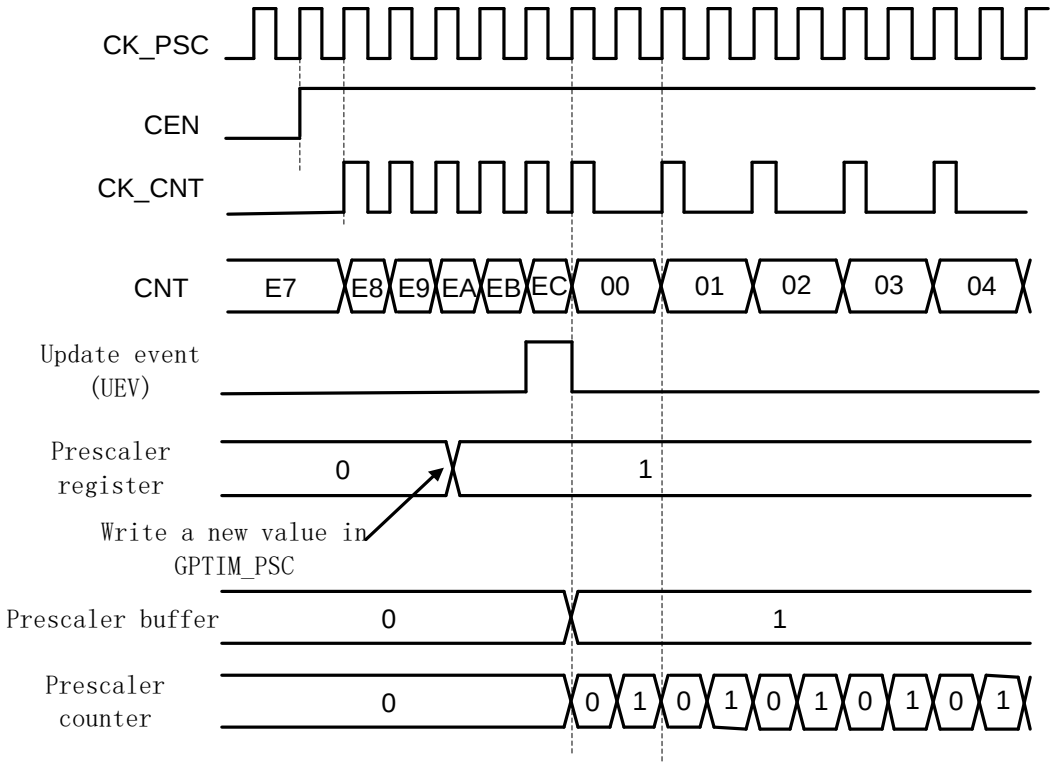


图 27-2 预分频从 1 变为 2 的波形

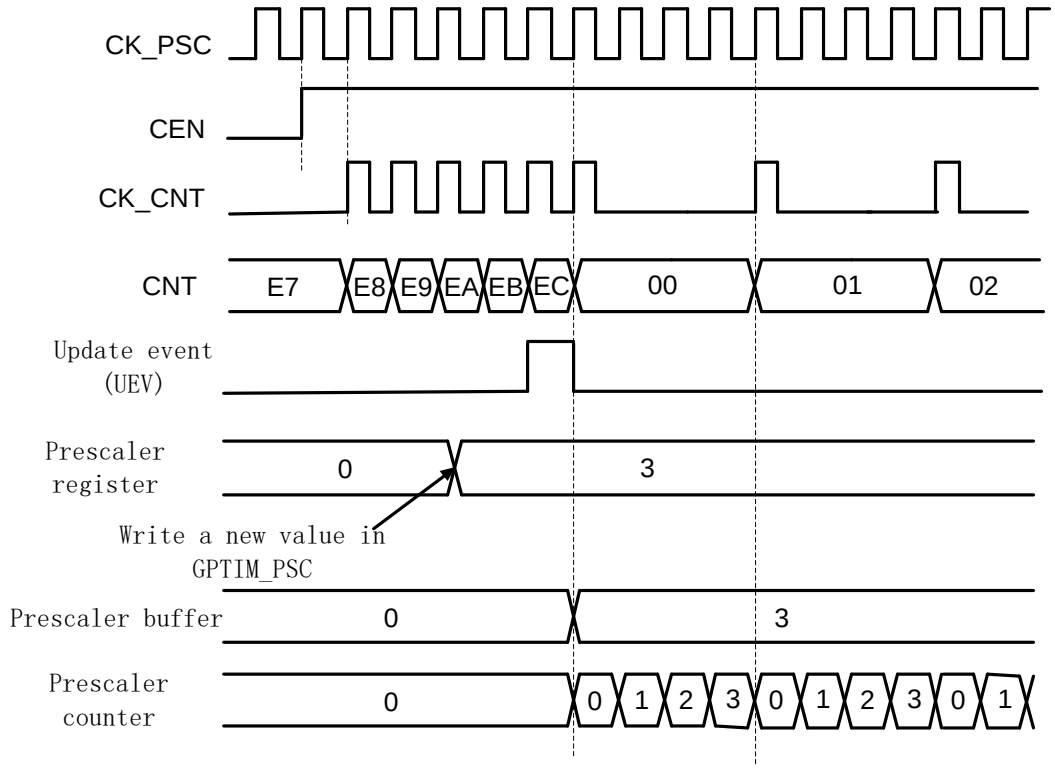


图 27-3 预分频从 1 变为 4 的波形

27.5.2 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。

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：

- ARR影子寄存器被更新为GPTIM_ARR内容
- PSC影子寄存器被更新为GPTIM_PSC内容

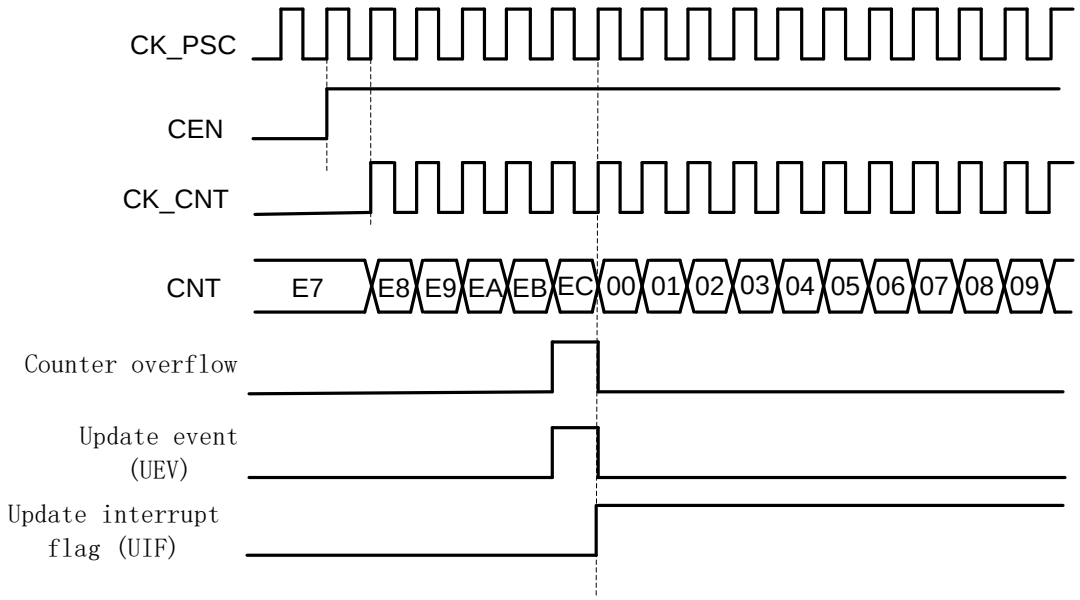


图 27-4 向上计数波形，内部时钟不分频

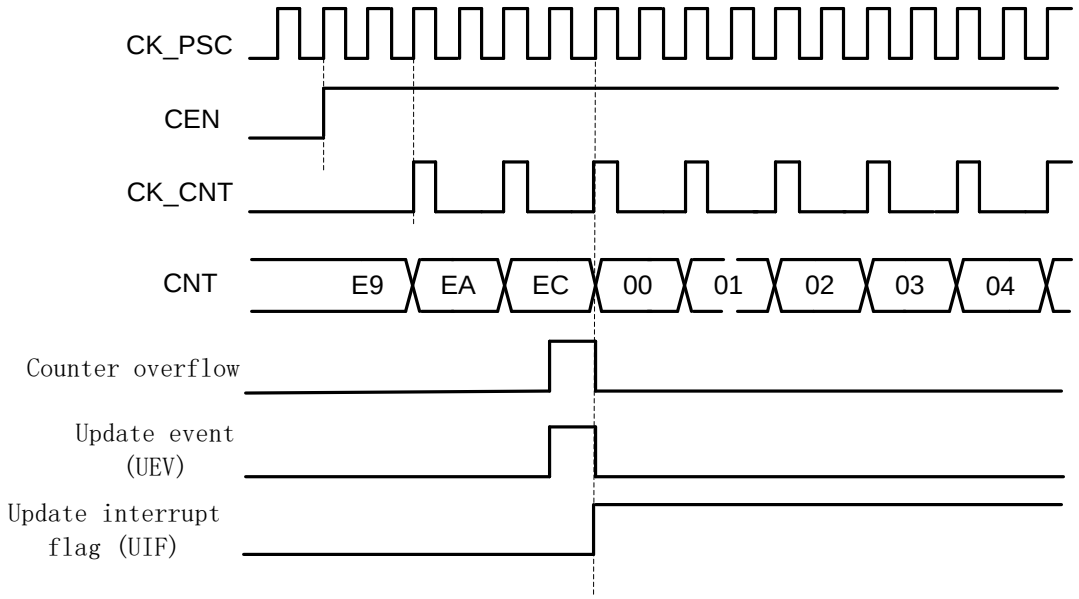


图 27-5 向上计数波形，内部时钟 2 分频

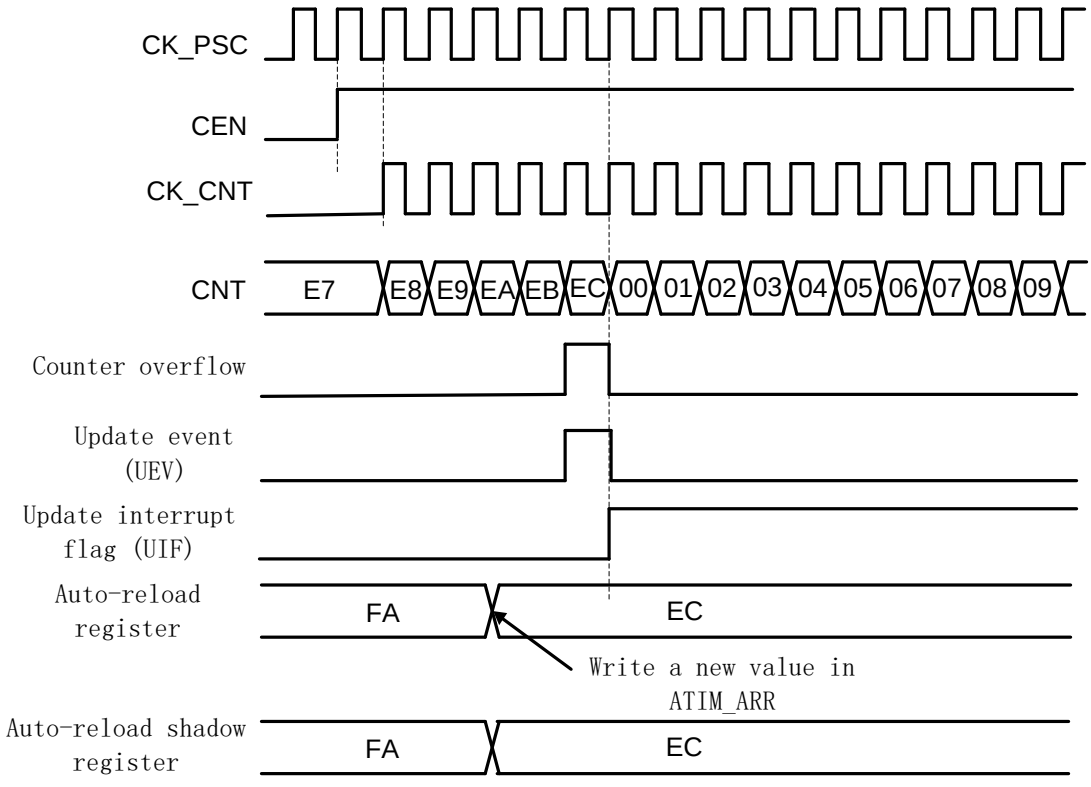


图 27-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件

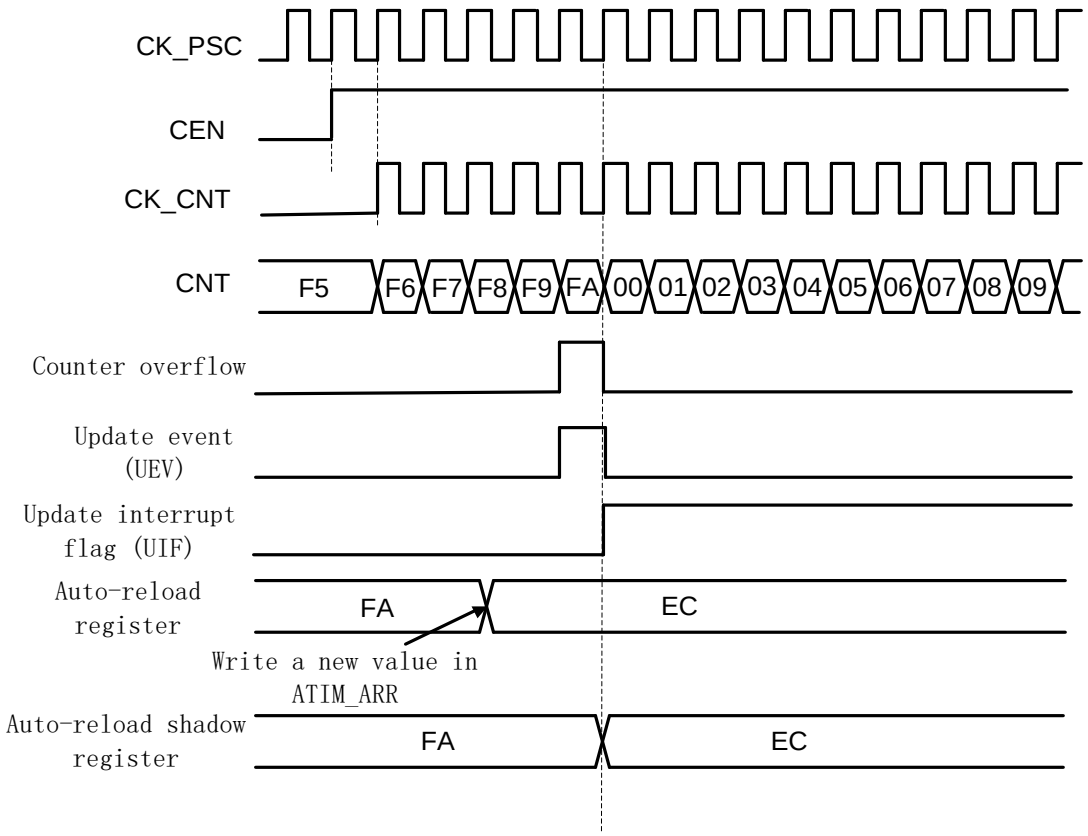


图 27-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件

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。

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：

- ARR影子寄存器被更新为GPTIM_ARR内容
- PSC影子寄存器被更新为GPTIM_PSC内容

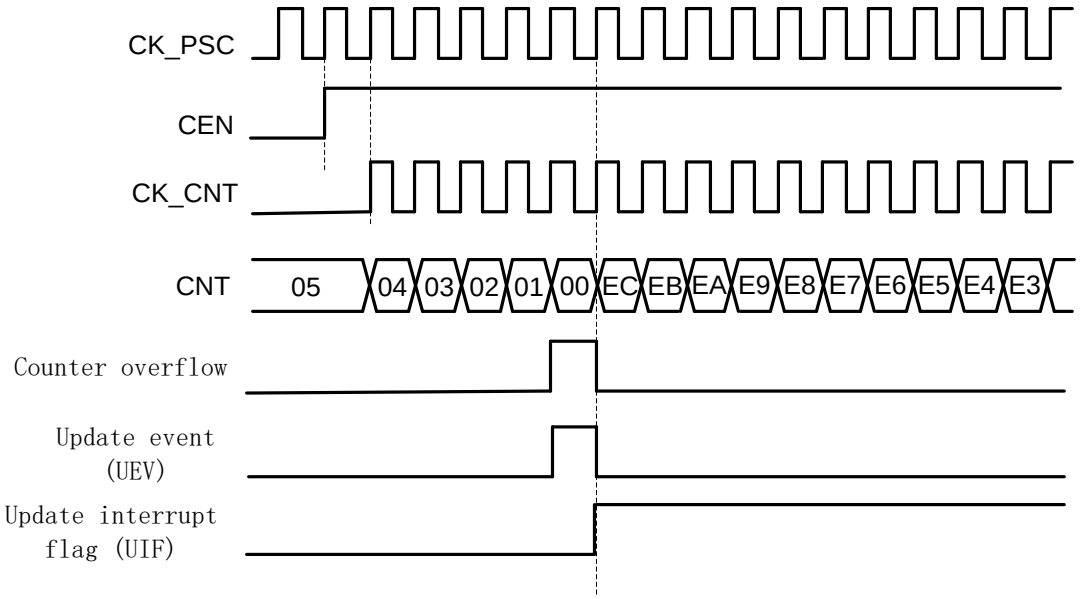


图 27-8 向下计数，内部时钟不分频

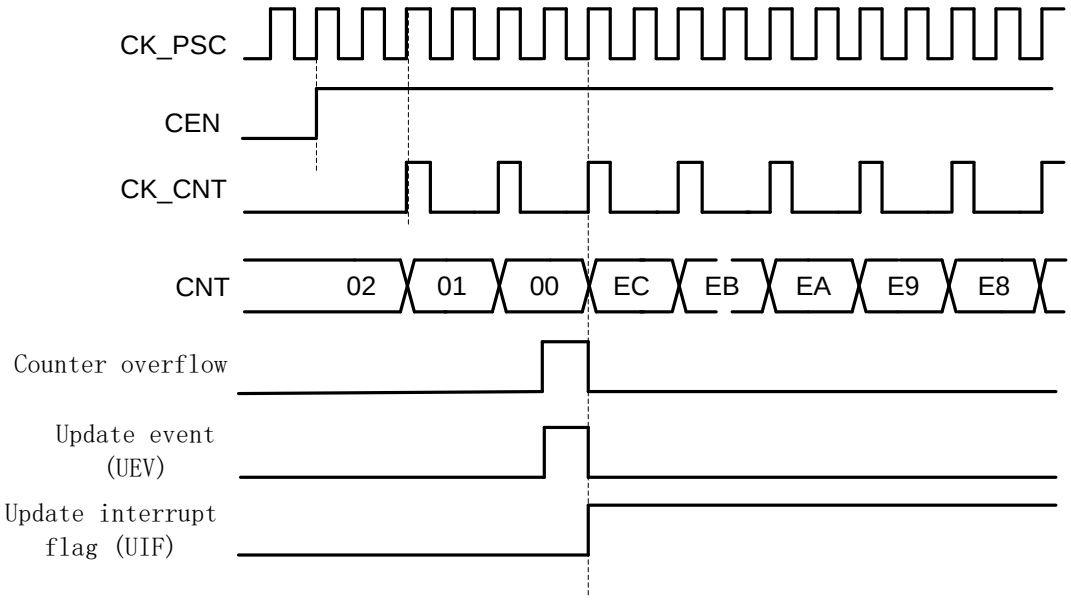


图 27-9 向下计数，内部时钟 2 分频

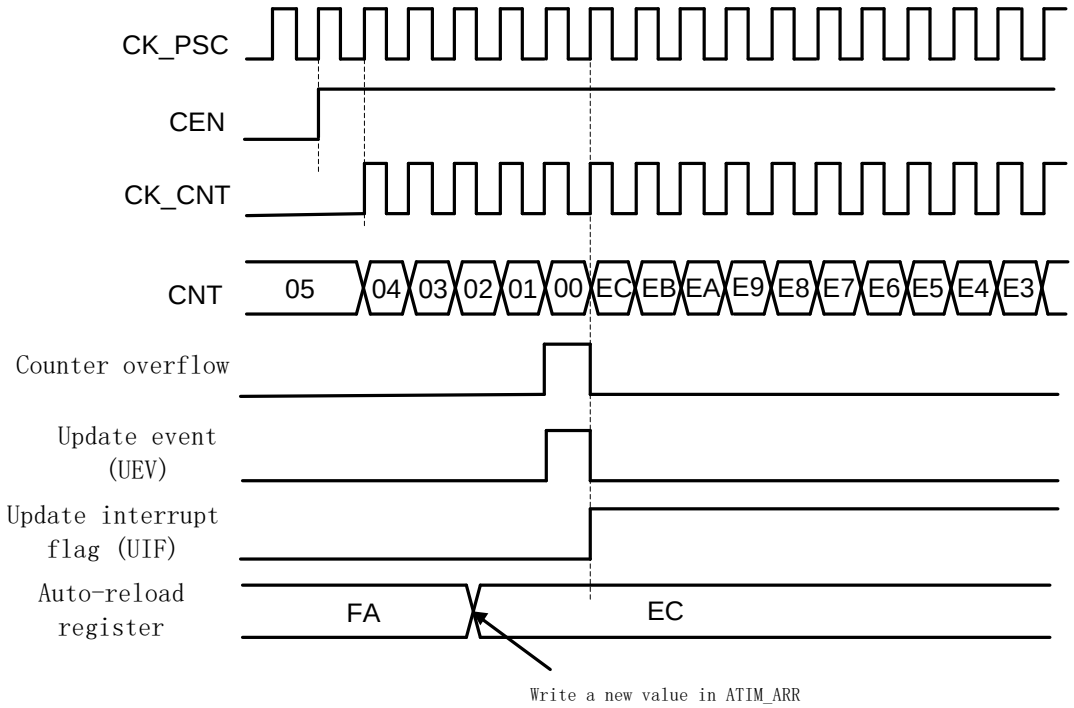


图 27-10 向下计数，内部时钟 2 分频

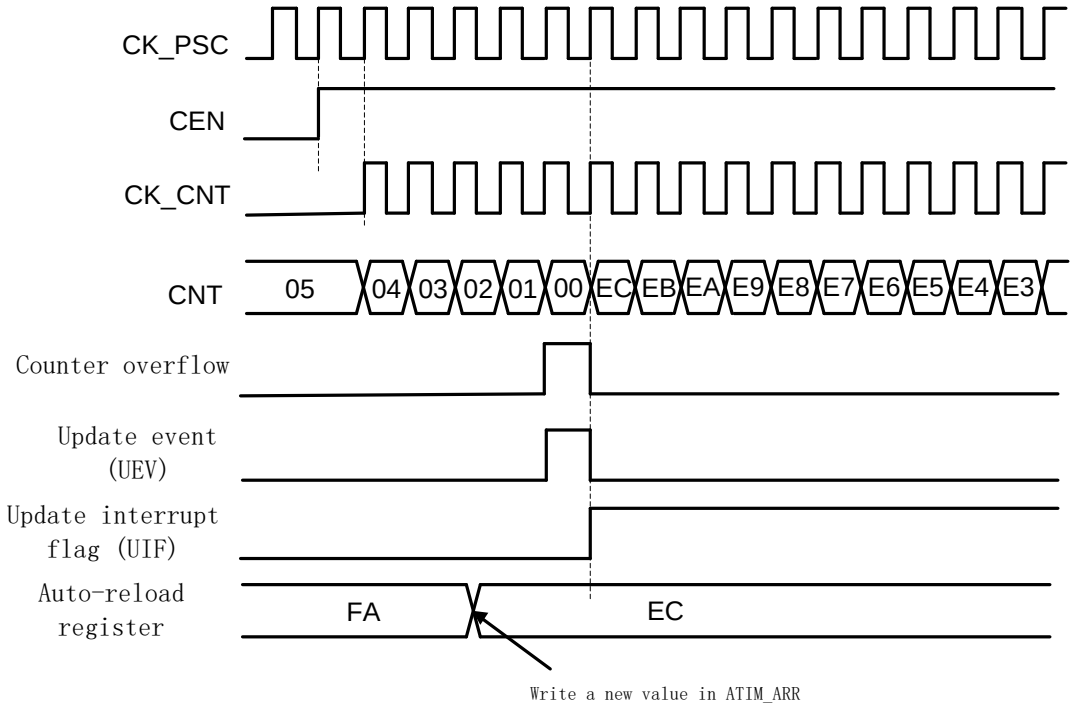


图 27-11 向下计数，不使用重复计数时的更新事件

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的影子寄存器。

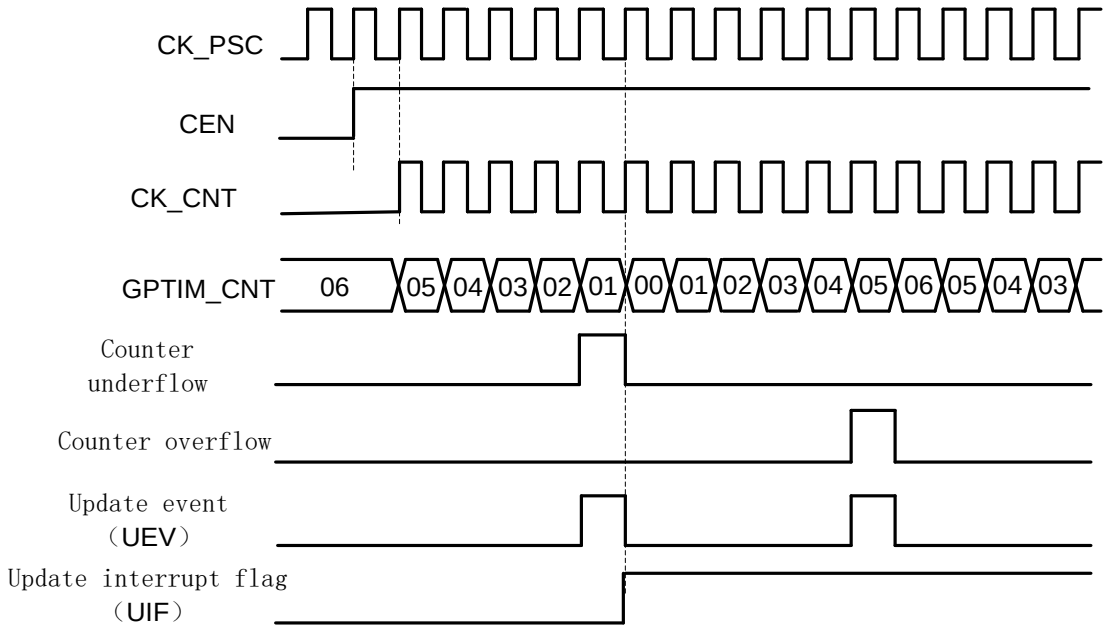


图 27-12 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6

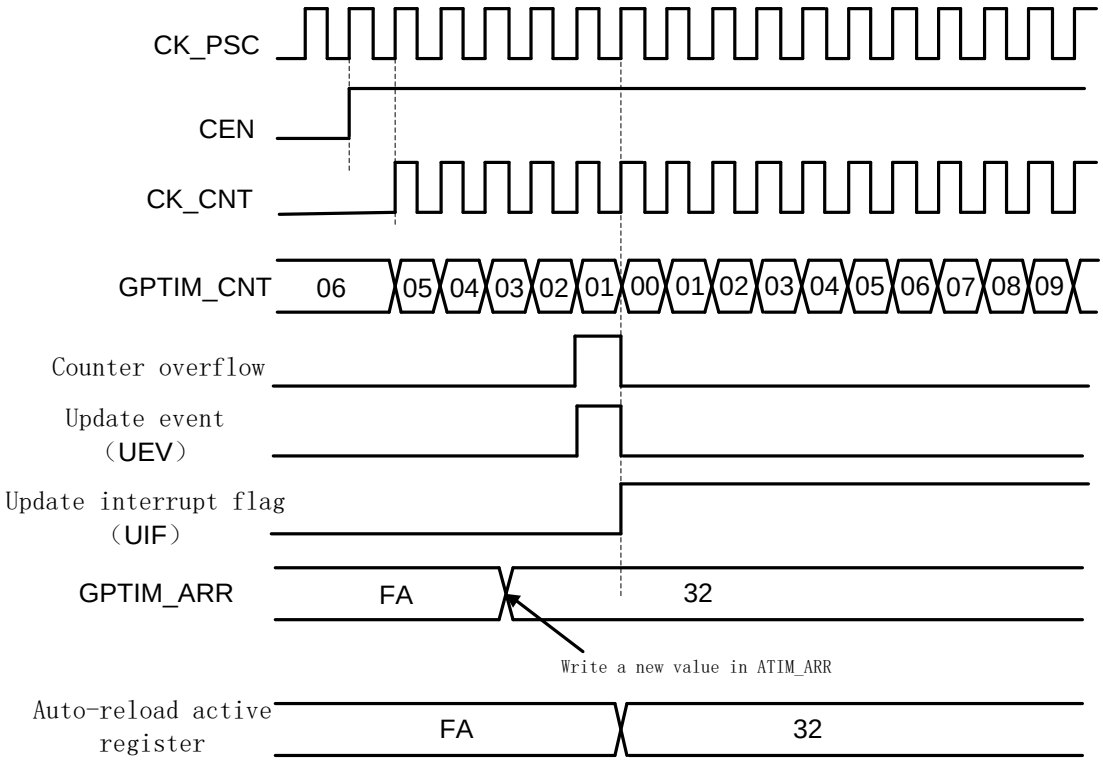


图 27-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

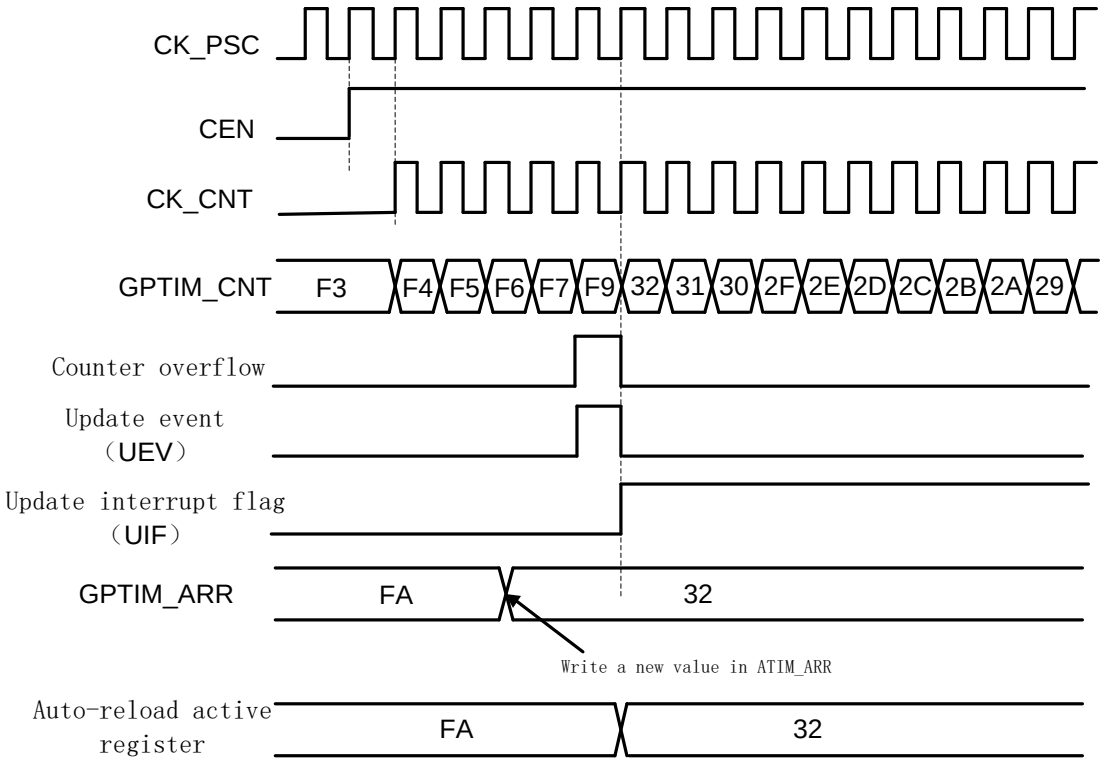


图 27-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

27.5.3 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

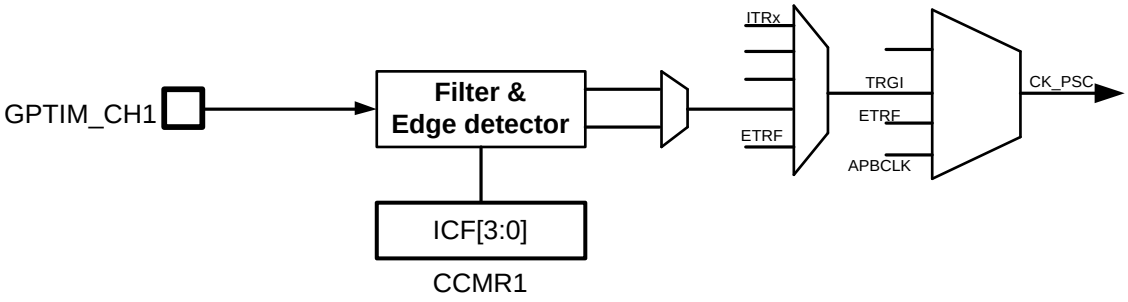


图 27-15 GPTIM 时钟源框图

27.5.3.1 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。

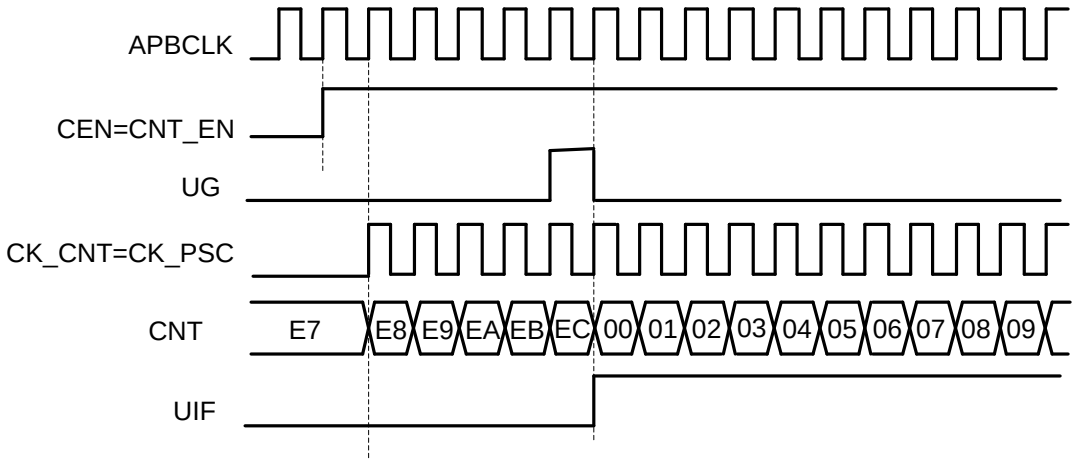


图 27-16 内部时钟源模式，时钟分频因子为 1

27.5.3.2 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。

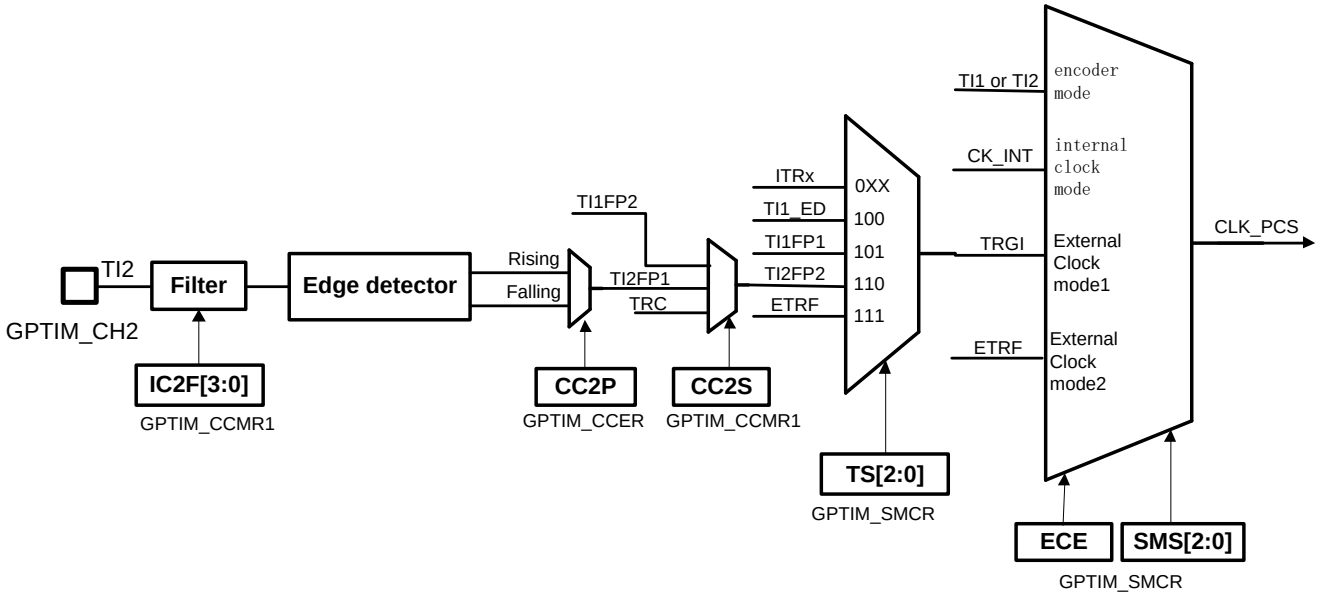


图 27-17TI2 外部时钟连接例子

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

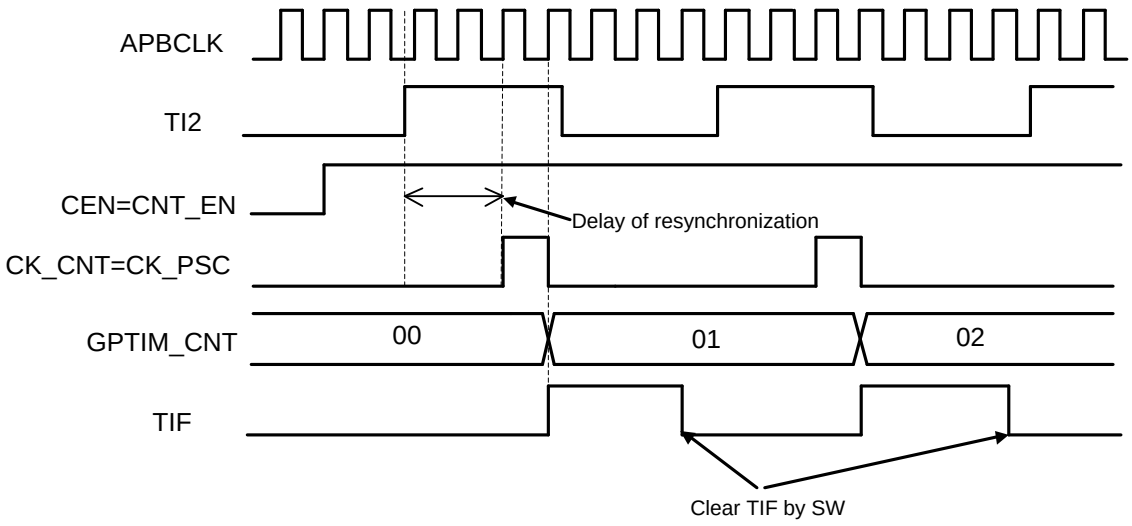


图 27-18 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能GPTIM的内部时钟（APBCLK），因为GPTIM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GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01, IC2映射到TI2
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC2F[3:0](IC2F=0000，不进行输入滤波)
- 使能外部时钟模式1，配置GPTIM_SMCR.SMCR=111
- 选择触发输入源，配置GPTIM_SMCR.TS=110,选定TI2作为触发输入源
- 打开通道使能，配置GPTIM_CCER.CC2E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式1的示例：

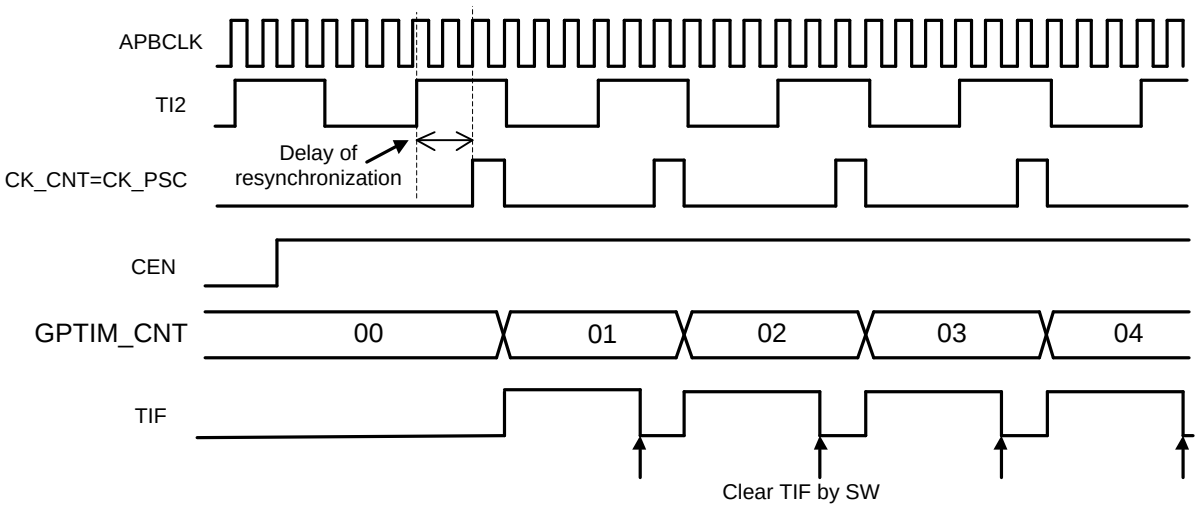


图 27-19 外部时钟模式 1 下的时序

27.5.3.3 外部时钟模式 2

此模式下使用GPTIM_ETR管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

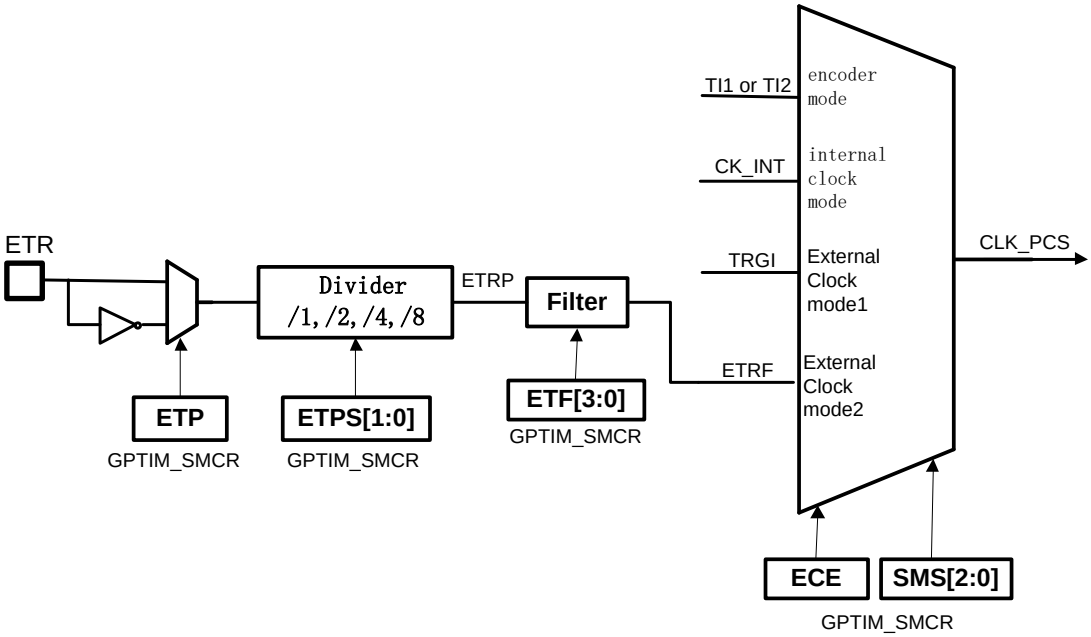


图 27-20 外部触发输入框图

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。

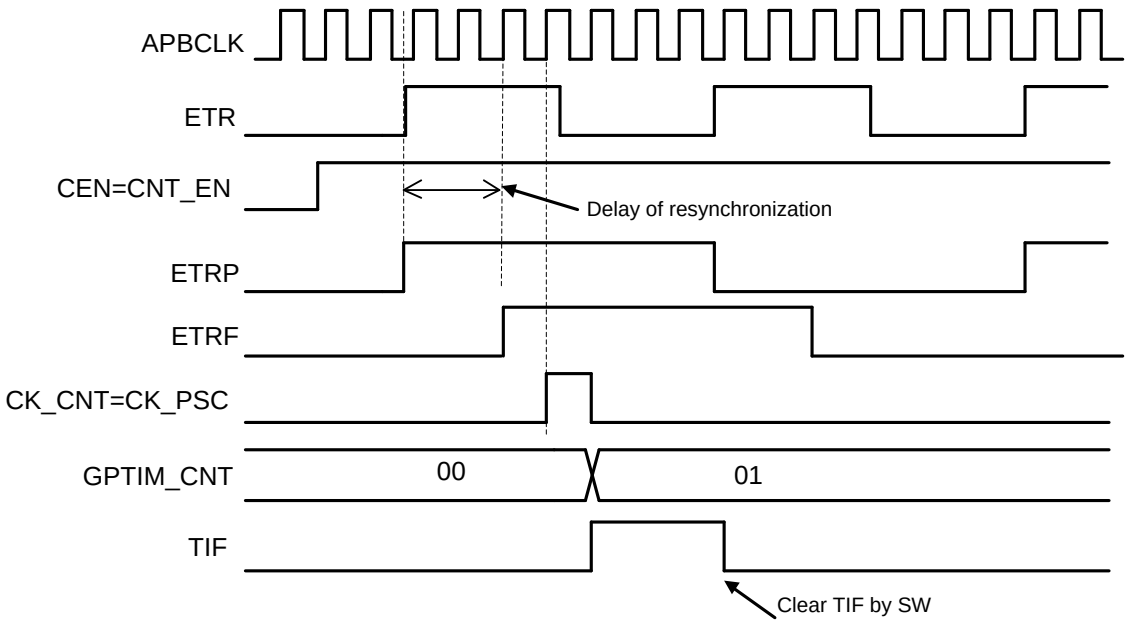


图 27-21 外部时钟模式 2 下的时序 1

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GPTIM_ETR功能
- 设置ETP进行沿选择，GPTIM_SMCR.ETP=0
- 设置ETR分频比，配置GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2,GPTIM_SMCR.ECE=1，GPTIM_SMCR.SMS=000
- 使能计数器，配置GPTIM_CR1.CEN=1

下图是一个典型的外部时钟模式2的示例：

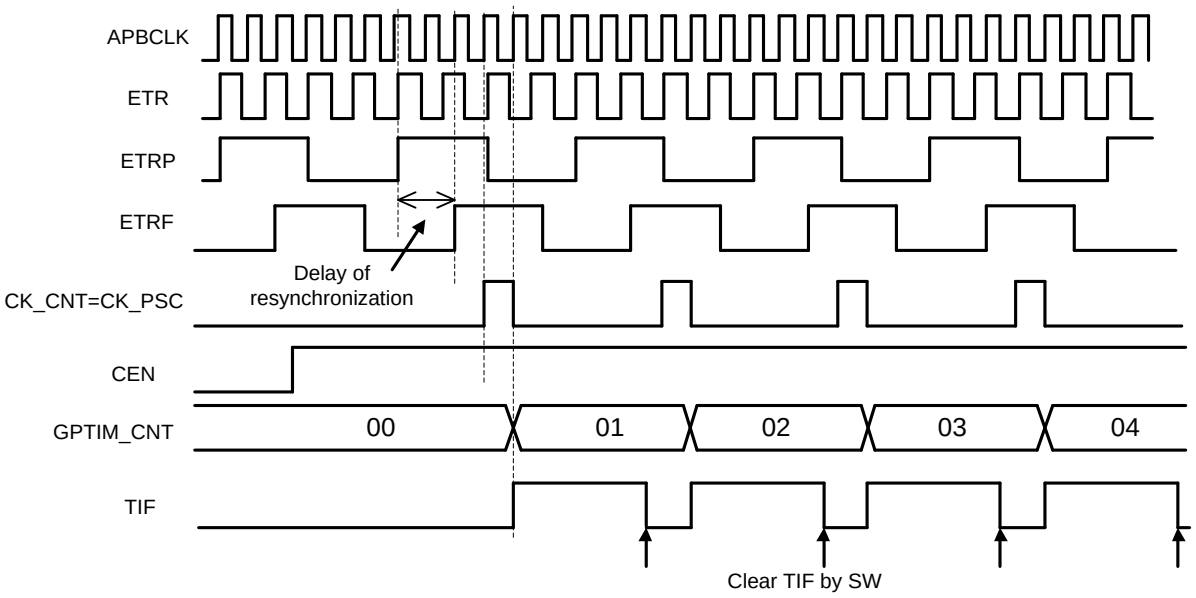
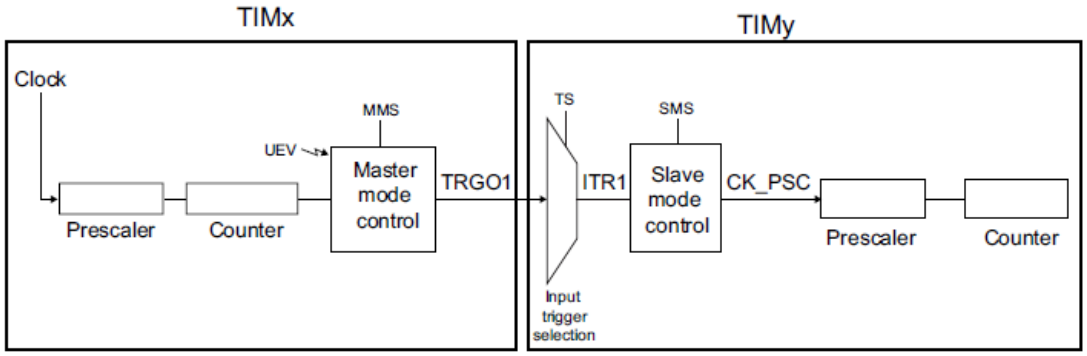


图 27-22 外部时钟模式 2 下的时序 2

在使用外部时钟模式2时，仍可以将GPTIM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。

27.5.3.4 内部触发模式

每个GPTIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当ITR选择为计数触发信号时，GPTIM计数器将在每个ITR信号的高电平计数。通过内部触发模式可以实现Timer级联，下图是一个例子：



配置TIMx为master mode并周期性输出TRGO脉冲信号，TIMy配置为Slave mode并将TIMx的TRGO设置为ITR；当TIMx.TRGO脉冲到来时，TIMy计数一次。

基于内部触发模式的timer级联有如下要求：

- TRGO信号设计为APBCLK单周期脉冲
- TIMx和TIMy都工作在APBCLK时钟域
- TRGO对于接收方来说是一个同步脉冲
- Master和Slave的工作时钟都必须使能

内部触发模式可以使用的触发信号除了其他定时器输出外，还可以是ADC_EOC或者比较器输出，为了满足以上要求，需要设计时将ADC和COMP输出的trigger信号处理成APBCLK同步脉冲。

27.5.4 内部触发信号（ITRx）的捕捉

每个GPTIM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。通过这个方法，Timer可以捕捉各种芯片内部信号的周期或电平宽度。

其中，作为计数触发的TRGO信号，在到达GPTIM的ITRx之前都被处理成一个APBCLK宽度的同步使能信号。而用作周期或宽度捕捉的信号，则无需处理，直接由GPTIM进行捕捉。

软件应保证选择正确的信号用于正确的功能，错误的配置将导致完全错误的结果。比如将ATIM_TRGO用于宽度捕捉，则结果没有意义。

27.5.5 捕捉/比较通道

GPTIM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。

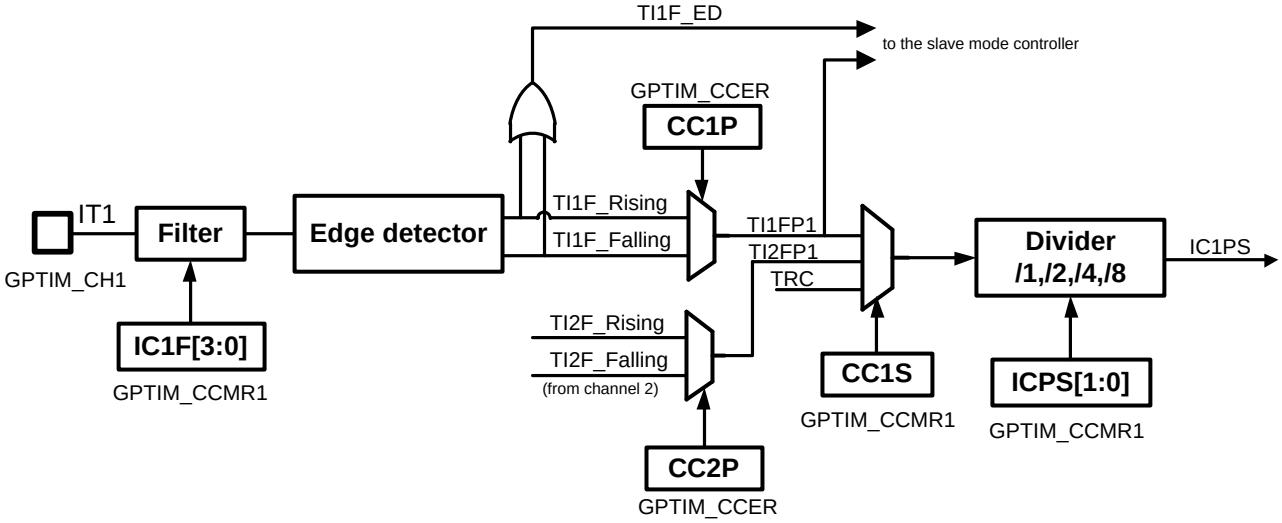


图 27-23 捕获/比较通道(通道 1 输入部分)

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。GPTIM输出通道不支持互补输出。

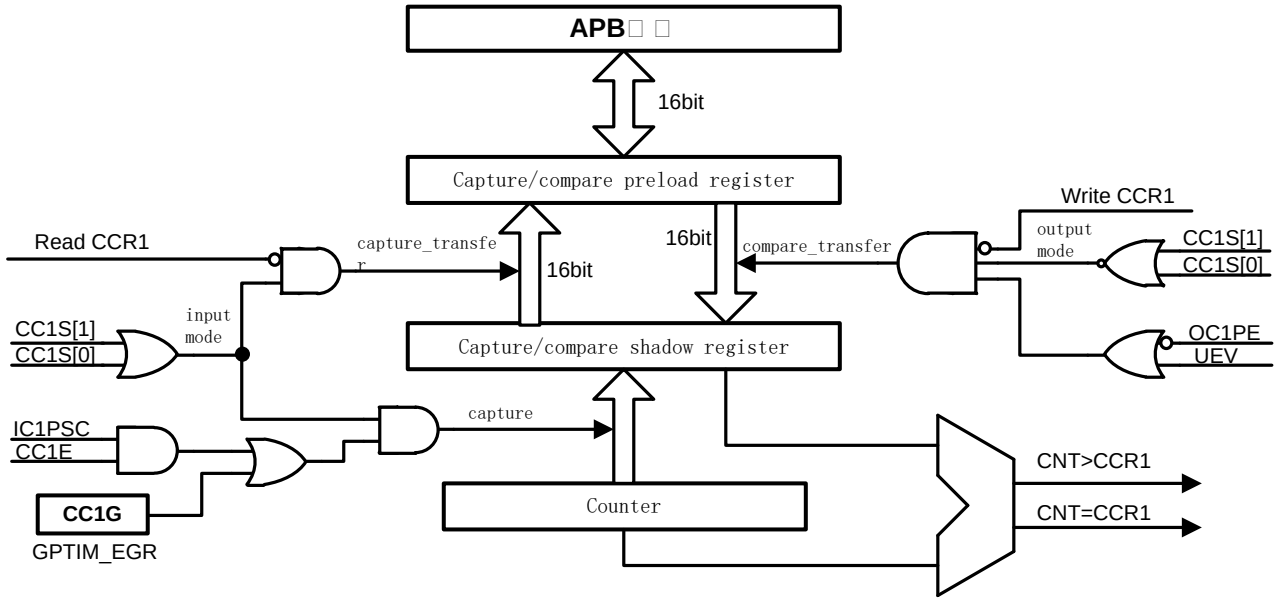


图 27-24 捕获/比较通道 1 的主电路

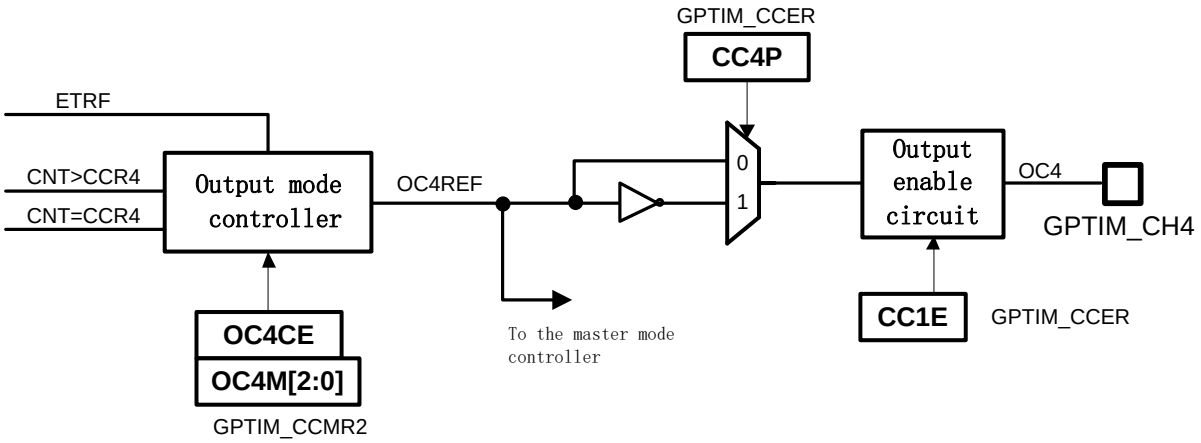


图 27-25 捕获/比较通道的输出部分

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。

27.5.6 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GPTIM_CCR1寄存器，配置步骤如下：

- 在GPIO模块中，配置相应管脚为GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01, IC1映射到TI1
- 选择计数有效沿，配置GPTIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置GPTIM_CCER.CC1E=1

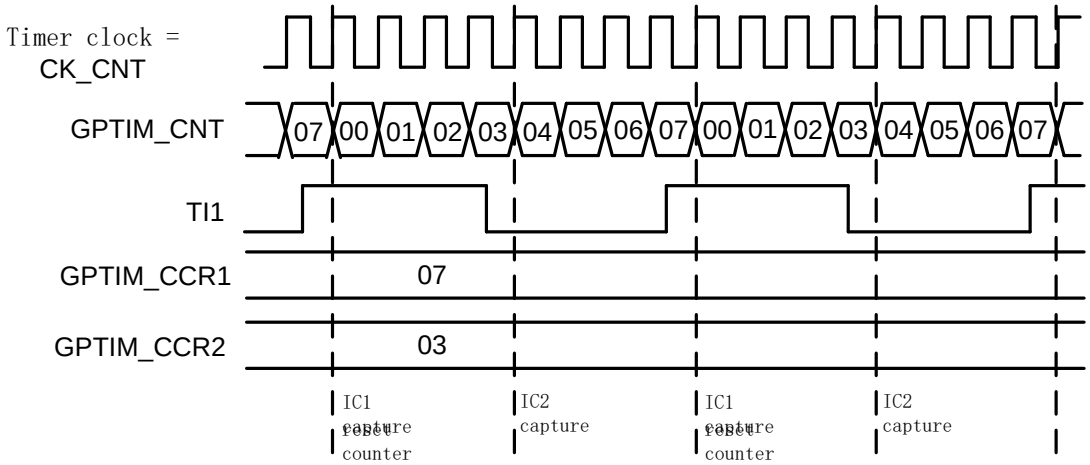


图 27-26PWM 输入捕获模式时序

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GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_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GPTIM_CCMR1.CC1S=01, GPTIM_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GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=1
- 配置输入滤波时间，配置GPTIM_CCMR1.IC1F[3:0]，GPTIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置GPTIM_CCMR1.IC1PS[1:0] ，GPTIM_CCMR1.IC2PS[1:0]

- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

27.5.7 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。

27.5.8 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：

1、选择计数时钟（内部、外部、预分频等）

2、向ARR和CCR寄存器写入期望数据

3、根据需要设置中断使能和DMA使能

4、选择输出模式

5、使能计数器

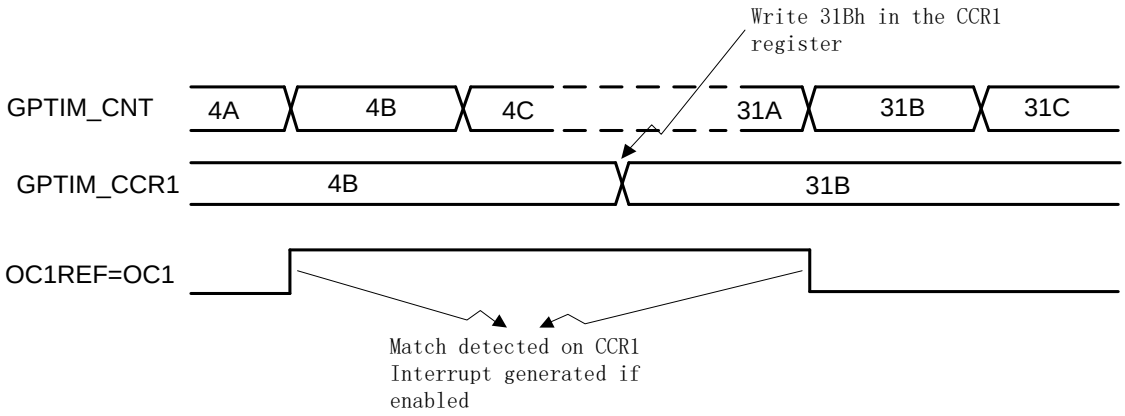


图 27-27 输出比较模式，翻转 OC1

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。

27.5.9 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.

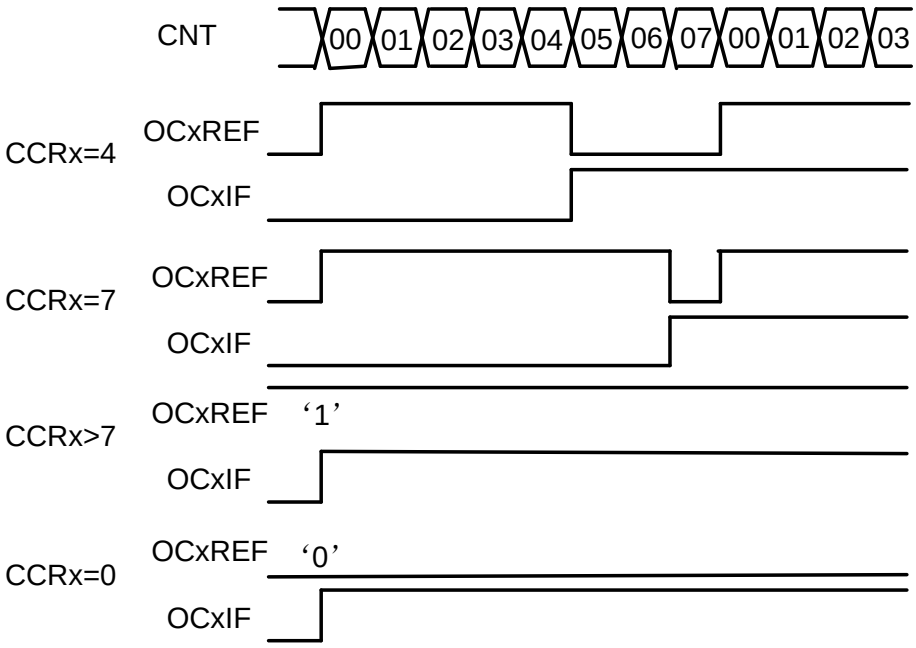


图 27-28 边沿对齐的 PWM 波形(ARR=7)

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：

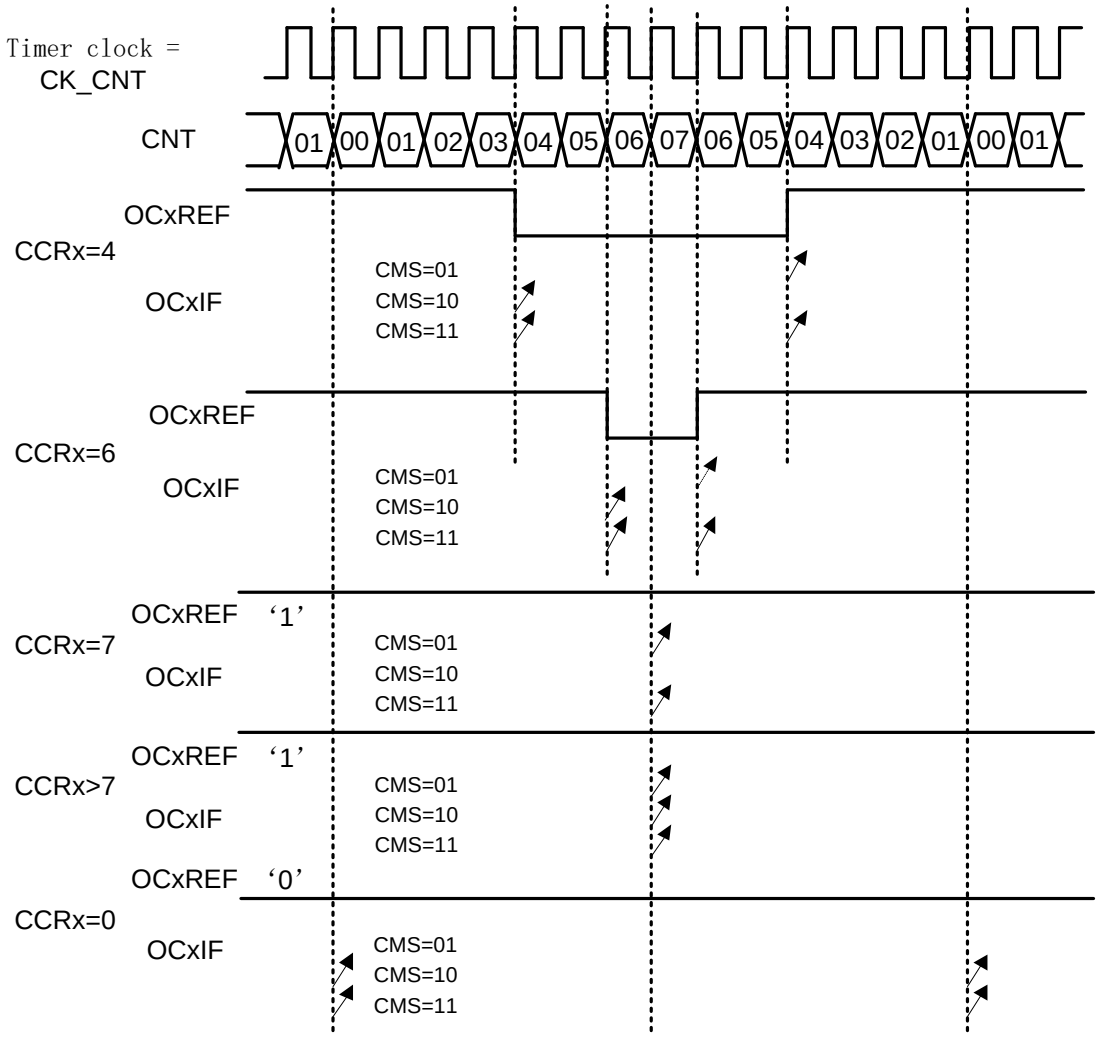


图 27-29 中央对齐的 PWM 波形(APR=7)

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。

27.5.10 非对称 PWM 模式

非对称PWM用于产生非对称的中心对齐PWM波形，需要同时使用两组CCR寄存器，由ARR寄存器决定PWM波形周期，一组CCR寄存器决定向上计数周期内的PWM信号变化点，另一组CCR寄存器决定向下计数周期内的PWM信号变化点。

使用非对称PWM模式时，只有OC1REF通道和OC3REF通道可以输出非对称波形，其中OC1REF由CCR1和CCR2寄存器控制，OC3REF由CCR3和CCR4寄存器控制。OCxM寄存器可以配置输出通道为非对称PWM模式1或者非对称PWM模式2。

下图为非对称相移PWM输出波形示例。

- ARR=7

- OC1REF为PWM mode 1，OC2REF为PWM mode 2

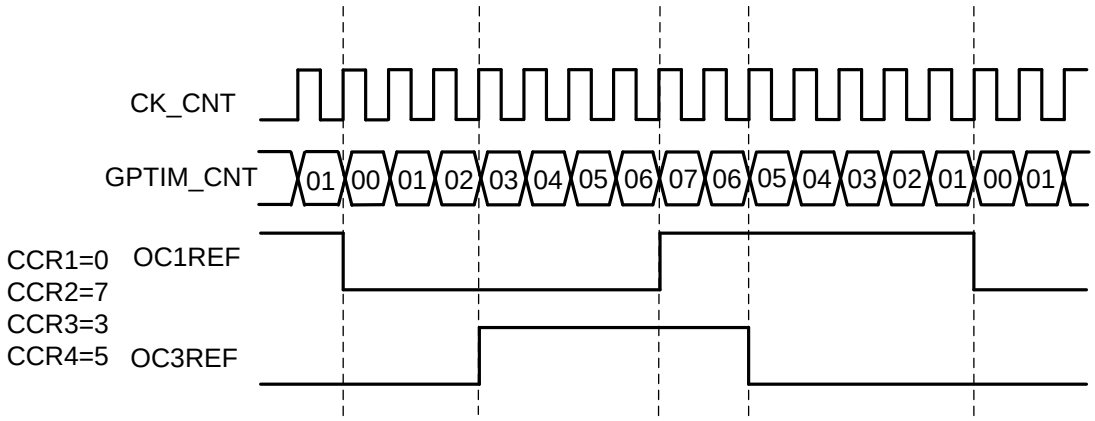


图27-30非对称PWM输出

使用非对称输出时，通道2和通道4仍可以单独输出普通PWM波形。

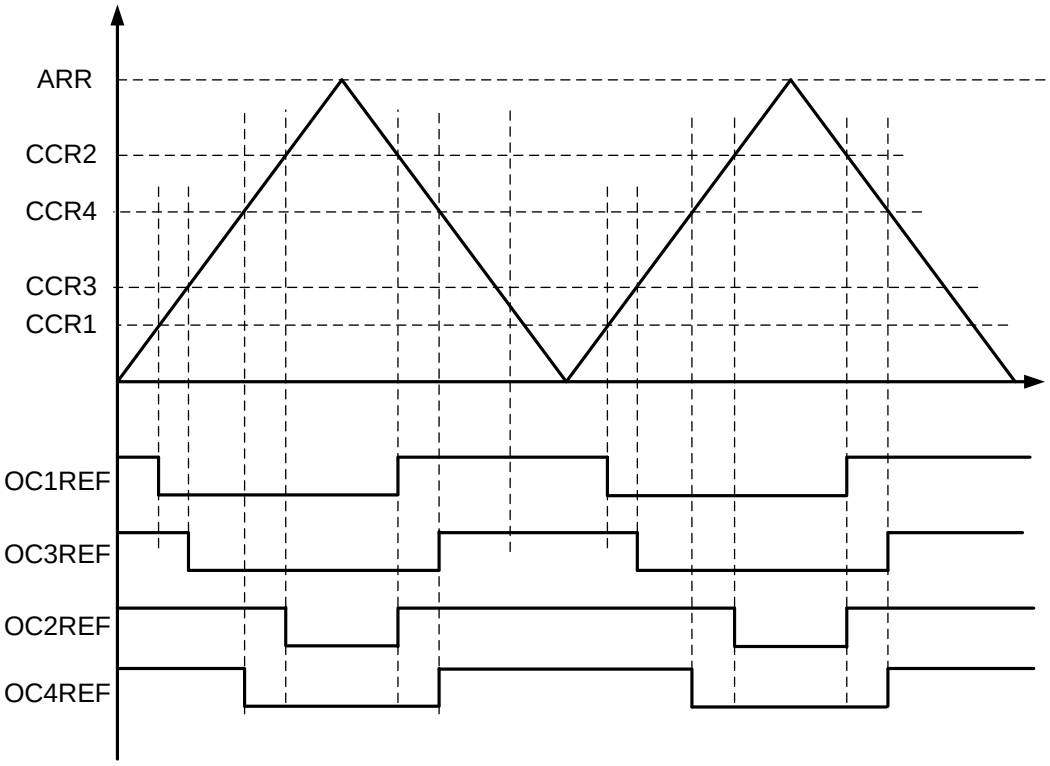


图27-31通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 1）

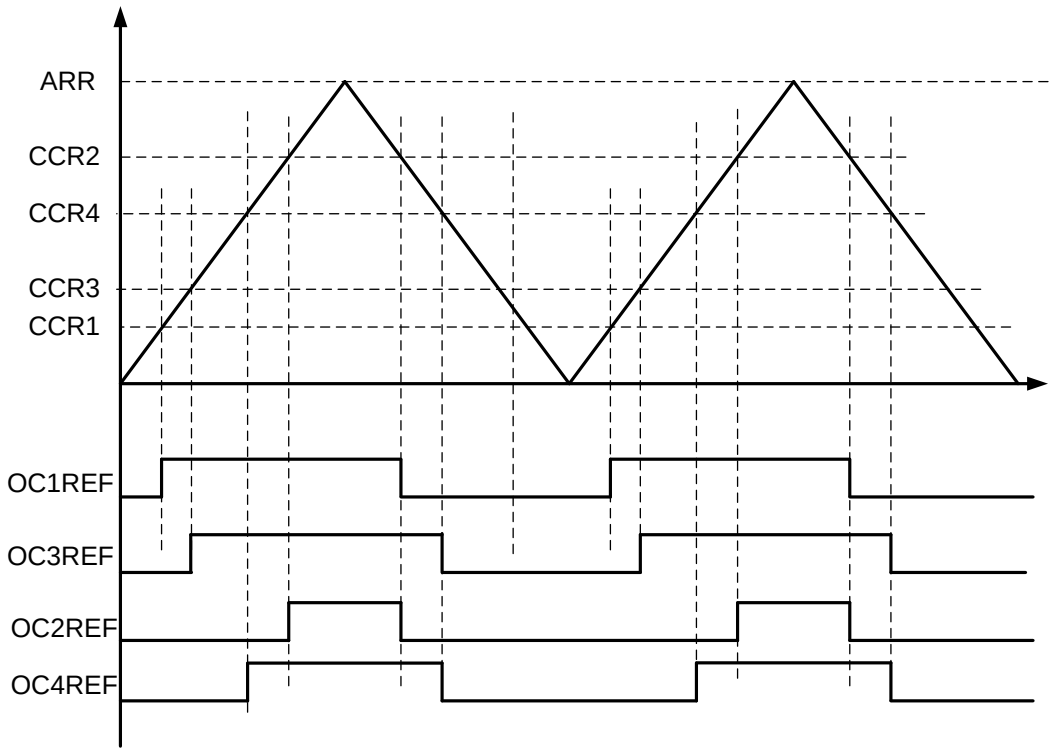


图27-32通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 2）

27.5.11 组合 PWM 模式

通道1和通道3可以配置为组合PWM模式，此模式下通道1的PWM波形输出由OC1REF和OC2REF组合产生，通道3的PWM波形输出由OC3REF和OC4REF组合产生，组合方式是OR或者AND，根据OCxM寄存器选择组合模式1（OR）或者组合模式2（AND）。

下图是组合PWM的例子：

- 通道1为组合PWM模式2
- 通道2为PWM模式1
- 通道3为组合PWM模式1
- 通道4为PWM模式2
- 通道1组合PWM输出

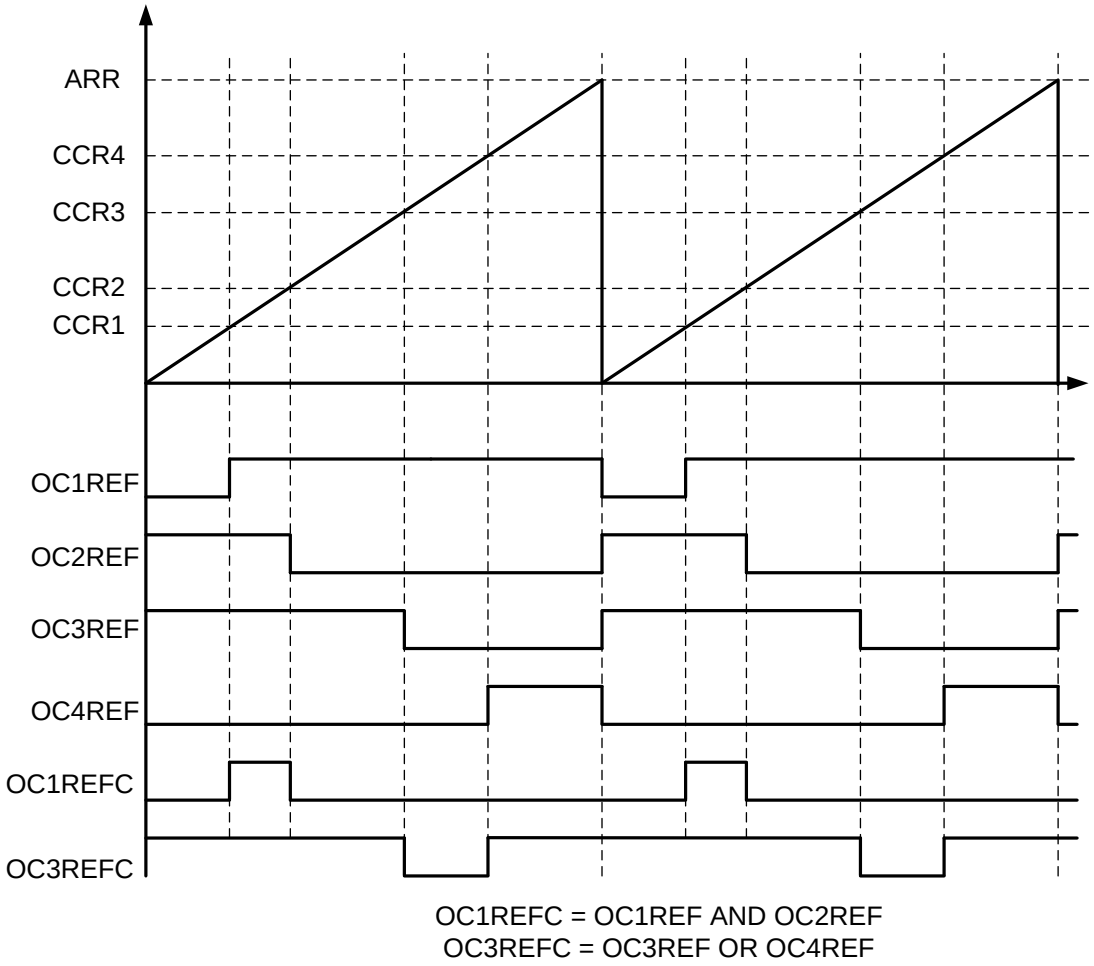


图27-33通道1/3组合PWM输出

27.5.12 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

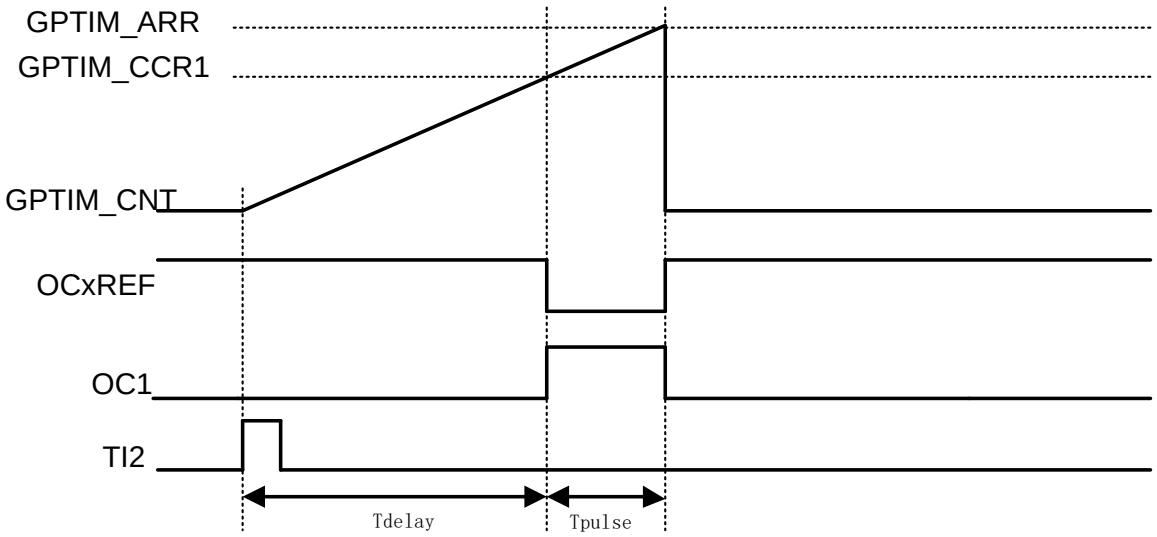


图 27-34 单脉冲模式的例子

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GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC2P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=110，TI2FP2作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110,TI2FP2用来启动计数器
- 打开通道使能，配置GPTIM_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GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置GPTIM_CCMR1.CC1S=00
- 选择计数有效沿，配置GPTIM_ CCMR1.OC1M=111,PWM模式2
- 打开通道使能，配置GPTIM_CCER.CC1E=1

OPM波形产生时基的特殊设置：

- GPTIM_CCR1的值决定了Tdelay
- GPTIM_ARR和GPTIM_CCR1的差值决定了Tpulse（GPTIM_ARR-GPTIM_CCR1）

- 设置为单脉冲模式，配置GPTIM_CR1.OPM=1

27.5.13 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，无法在软件force模式下起作用。使能此功能需要将OcxCE置1.

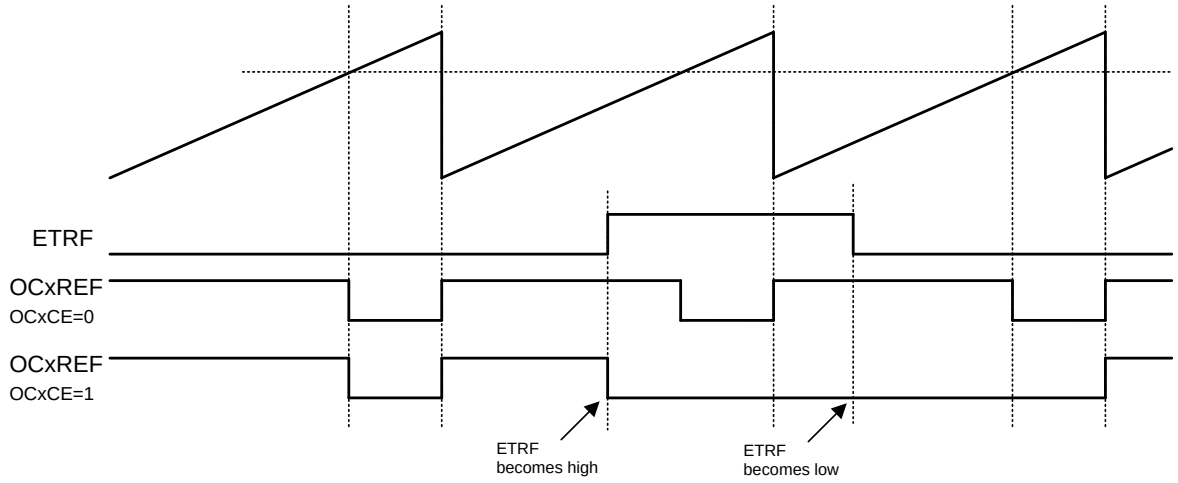


图 27-35 ETR 信号清除 GPTIM 的 OCxREF

27.5.14 编码器接口模式（encoder interface）

27.5.14.1 正交编码器

GPTIM使用两个输入信号TI1和TI2来与正交编码器连接，输入信号首先经过数字滤波和极性选择，定时器再根据滤波和极性选择后的信号TI1FP1和TI2FP2的边沿进行计数。GPTIM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，同时硬件自动更新DIR寄存器，向软件指示当前计数方向。根据SMS寄存器的配置，GPTIM可以只对TI1的边沿计数、只对TI2的边沿计数、或者同时对TI1和TI2的边沿计数。

编码器接口模式启动后，计数器根据输入信号在0和ARR范围内连续计数，因此使能定时器之前必须配置ARR寄存器。

下表是计数方式与两路输入信号之间的关系：

有效沿	SMS	对应信号的电平（TI1 对应TI2，TI2 对应TI1）	TI1FP1		TI2FP2	
			上升	下降	上升	下降
仅在TI1处计数	0001	高	递减	递增	不计数	不计数
		低	递增	递减	不计数	不计数

x2 mode						
仅在TI2处计数 x2 mode	0010	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 x4 mode	0011	高	递减	递增	递增	递减
		低	递增	递减	递减	递增
仅在TI1处计数 x1 mode	1110	高	递减	递增	不计数	不计数
		低	不计数	不计数	不计数	不计数
仅在TI2处计数 x1 mode	1111	高	不计数	不计数	递增	递减
		低	不计数	不计数	不计数	不计数

表 27-9 encoder interface 计数方式（CC1P=CC2P=0）

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。

正交编码器的两路输出可以直接连接到MCU的输入管脚，同时，正交编码器的原位输出（index）信号可以连接到ETR输入，用于触发计数器复位。

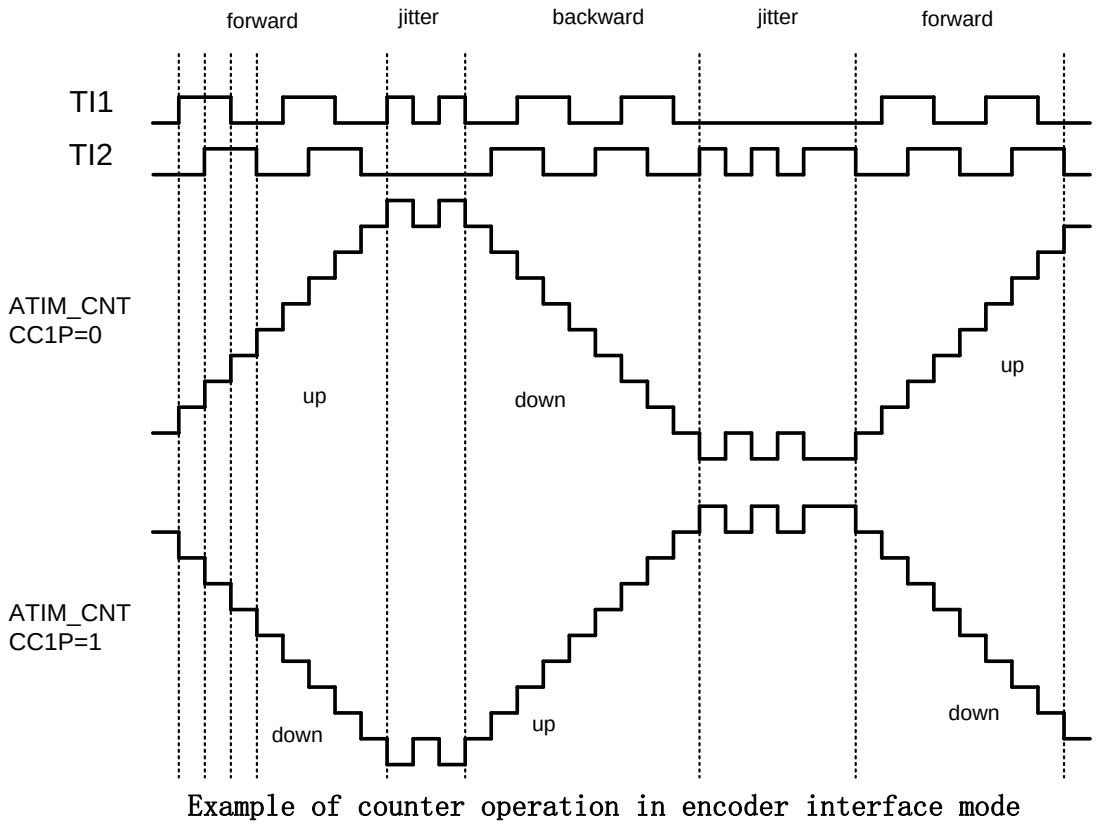


图27-36编码器模式下的计数器操作实例

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GPTIM_CH1，GPTIM_CH2功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01，GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=0
- 设定从模式控制器为编码模式3，配置GPTIM_SMCR.SMS[3:0]=0011
- 打开通道使能，配置GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

编码器接口支持x4，x2，x1计数模式，其中x4模式在TI1和TI2的每个边沿计数，x2模式只在TI1或者TI2的边沿计数，x1模式则只在TI1或者TI2的某个边沿上计数（由CCxP选择有效沿）。

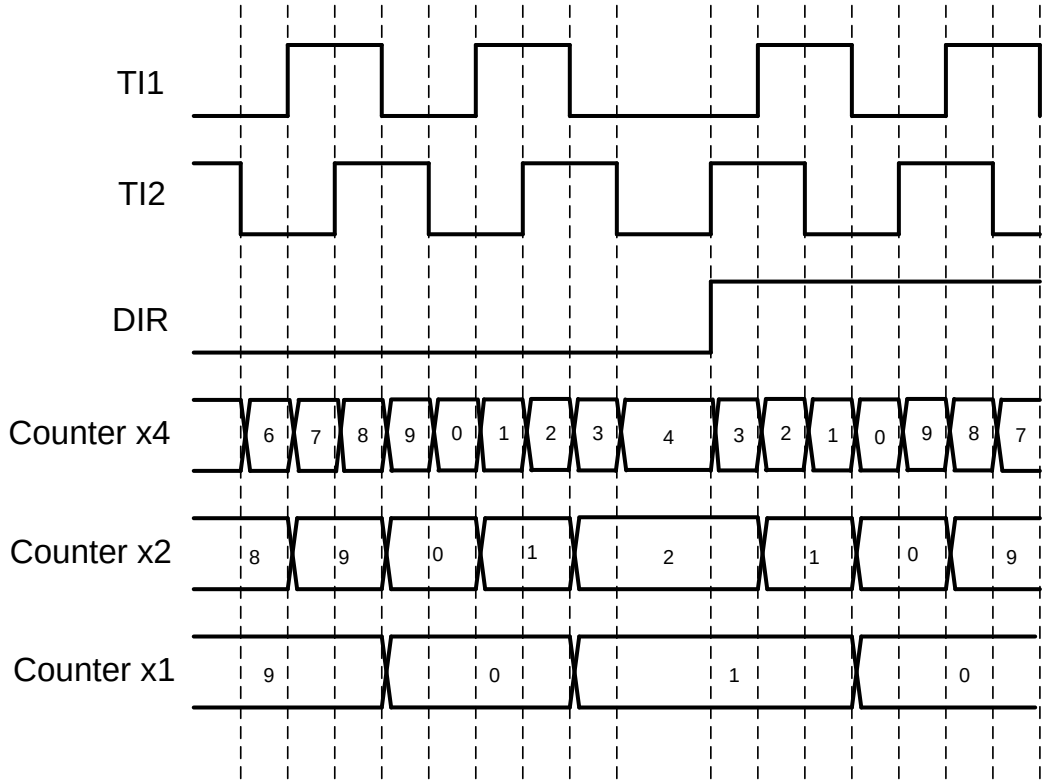


图27-37正交编码器不同边沿计数模式示例

当GPTIM_CR1寄存器中的UIFREMAP位置位时，UIF中断标志寄存器被实时复制到GPTIM_CNT[31]，供软件进行原子读取。

27.5.14.2 时钟方向编码器模式

除了正交编码器，另一种编码器提供“时钟”和“方向”两个输出信号。为支持这种编码器模式，需要将SMS配置为1010或1011，将时钟信号连接到TI2，将方向信号连接到TI1。

- SMS[3:0]=1010：x2模式，计数器在TI2输入信号的上升沿和下降沿都计数

- SMS[3:0]=1011：x1模式，计数器在TI2上升或者下降沿计数，由CC2P寄存器定义有效沿（CC2P=0：上升沿计数，CC2P=1：下降沿计数）

计数器计数方向由TI1输入的“方向”信号和CC1P寄存器决定。

- CC1P=0：TI1为高电平时向上计数，TI1为低电平时向下计数
- CC1P=1：TI1为高电平时向下计数，TI1为低电平时向上计数

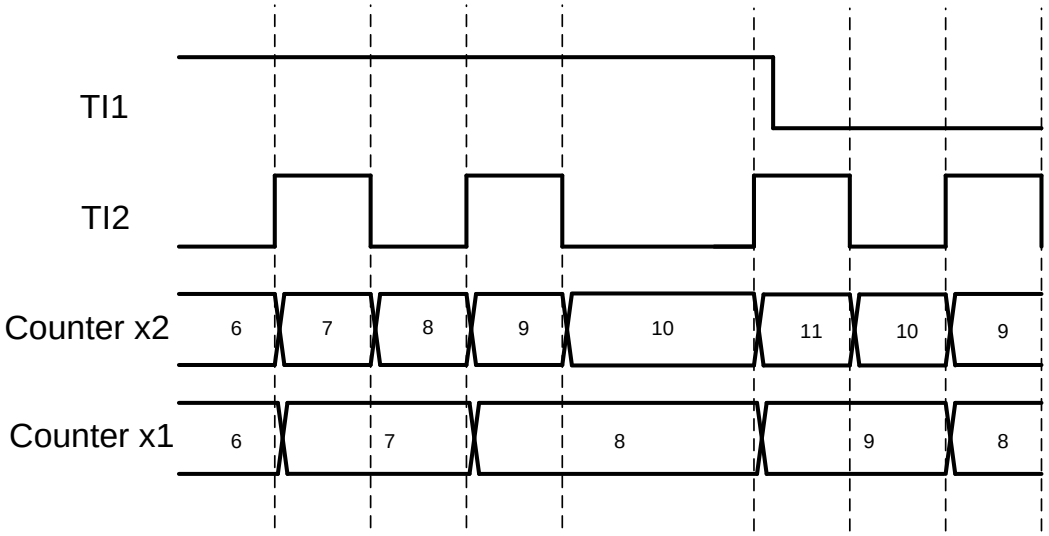


图27-38时钟方向编码器示例

27.5.14.3 带有方向信息的时钟编码器模式

这种编码器用时钟信号调制方向信息，输出两个信号，任意时刻只有其中一个信号带有时钟。

- SMS[3:0]=1100：x2模式，计数器在输入信号的上升沿和下降沿都计数。CC1P和CC2P用于定义时钟空闲状态，0表示高电平空闲，1表示低电平空闲
- SMS[3:0]=1101：x1模式，计数器在输入信号的上升或者下降沿计数，由CCxP寄存器定义有效沿（CCxP=0：下降沿计数，CCxP=1：上升沿计数）

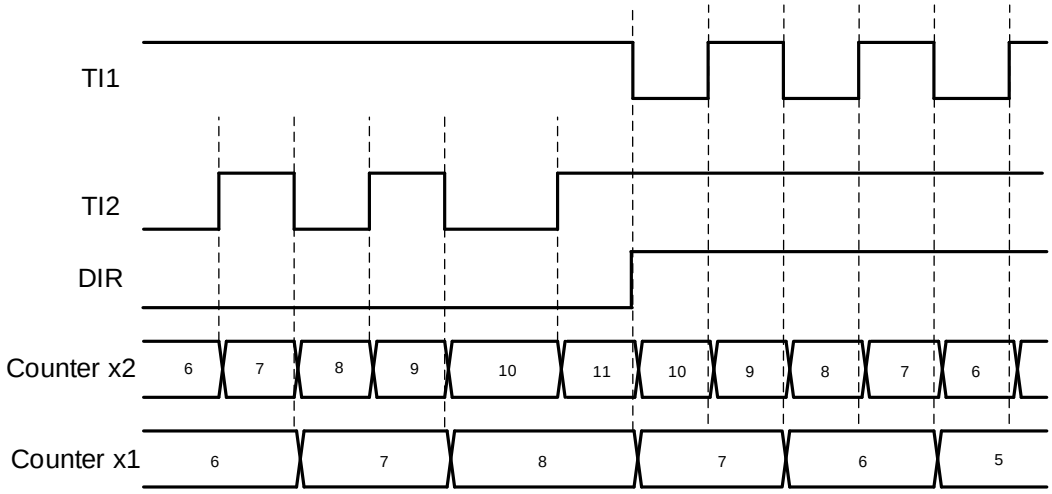


图27-39带方向的时钟编码器（CC1P=CC2P=0）

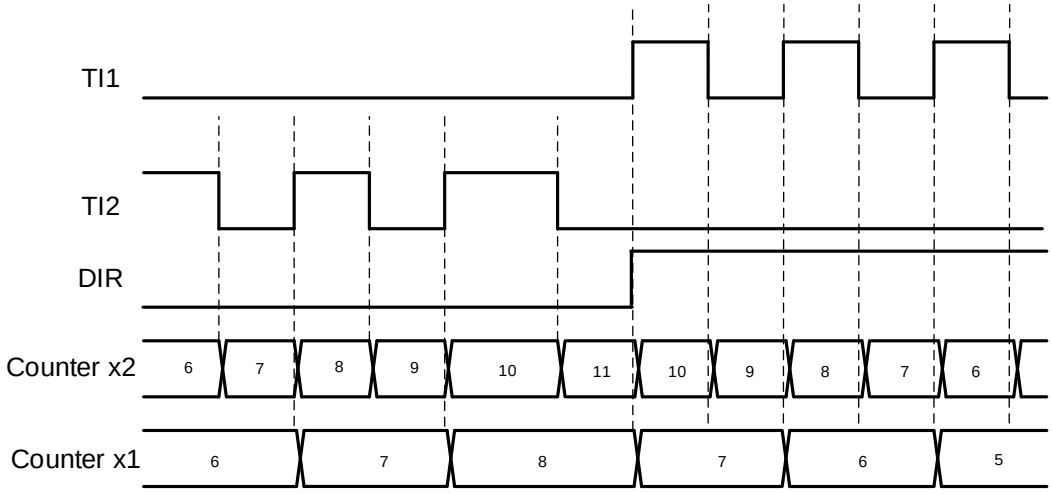


图27-40带方向的时钟编码器（CC1P=CC2P=1）

下表是计数方式与两路输入信号之间的关系：

带方向的时钟编码模式	SMS	对应信号的电平（TI1 对应 TI2，TI2 对应TI1）	TI1FP1		TI2FP2	
			上升	下降	上升	下降
CCxP=0 x2 mode	1100	高	递减	递减	递增	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x2 mode	1100	高	不计数	不计数	不计数	不计数
		低	递减	递减	递增	递增
CCxP=0 x1 mode	1101	高	不计数	递减	不计数	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x1 mode	1101	高	不计数	不计数	不计数	不计数
		低	递减	不计数	递增	不计数

表 27-10 directional clock mode 计数方式

27.5.14.4 Encoder index（Z 相输入）

正交编码器除了输出2路正交编码信号之外，通常还会输出1个周期归零指示信号（index）。这个零位信号可以用来复位计数器，需要将index信号连接到GPTIM的ETR输入引脚。

Index功能通过GPTIM_ECR寄存器中的IE寄存器使能，并且尽在encoder模式下有效，即SMS必须为0001/0010/0011/1010/1011/1100/1101/1110/1111

常见的编码器输出的零位信号有如下格式：

- Gated with A and B
- Gated with A or gated with B
- Ungated

不同零位信号的示意图如下：

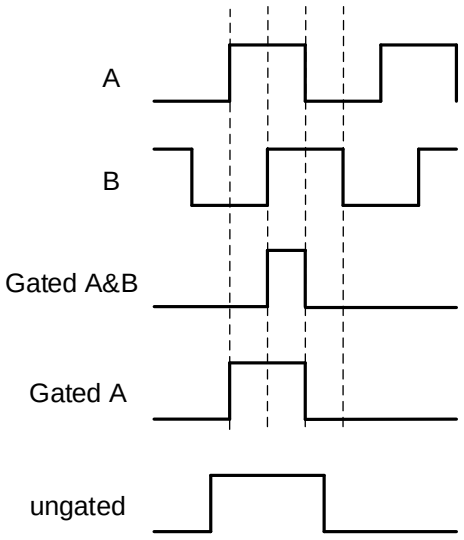


图27-41编码器index定义

使用中将编码器A相连接到TI1，B相连接到TI2，Z相连接到ETR。当GPTIM检测到index输入后，产生index事件，执行对计数器的特定操作，注意这个操作是要被通道到特定状态上的，而不是异步即时发生的。

GPTIM总是在检测index输入信号，然而检测到index后，触发计数器清零或置位的index事件将被同步到特定AB相状态上执行，通过GPTIM_ECR中的IPOS[1:0]寄存器来定义index事件被同步执行的AB相位段。根据计数器计数方向，硬件电路采取不同的index事件执行策略：

- 向上计数时（DIR=0），index事件触发计数器清零
- 向下计数时（DIR=1），index事件触发计数器回到ARR

IPOS寄存器可以配置计数器在检测到index后，在什么AB相位下执行index事件，比如IPOS=11，计数器仅在A=1 &B=1的条件下执行index事件（虽然有可能在11状态之前就检测到index）。

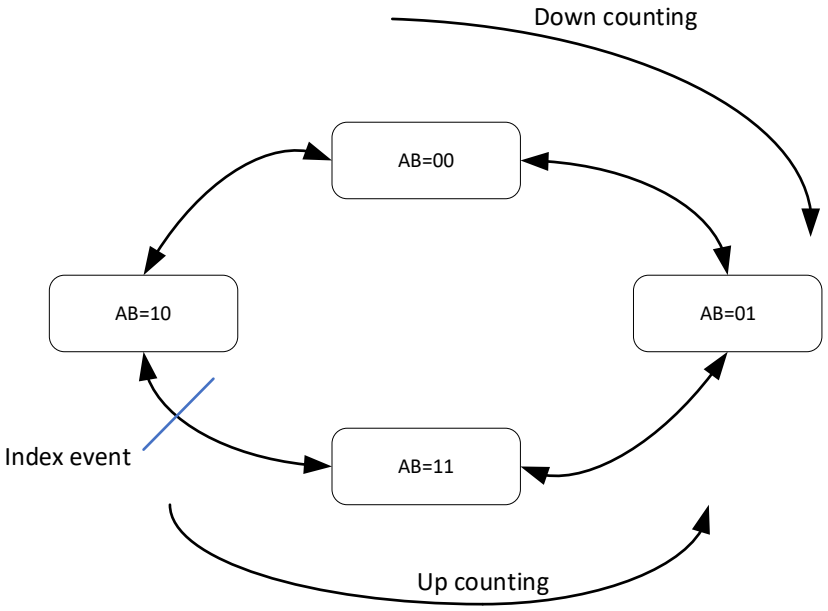


图27-42IPOS[1:0]=11时的index识别

下图显示了IPOS=11，定时器识别index事件，并且设置计数值的时序。

- 向上计数时（DIR=0），检测到index，当AB处于11相位计数值被清零（1次）
- 向下计数时（DIR=1），检测到index，当AB退出11相位时计数值设置为ARR（1次）

当index事件被执行时（而不是被检测到时），index中断标志置位。

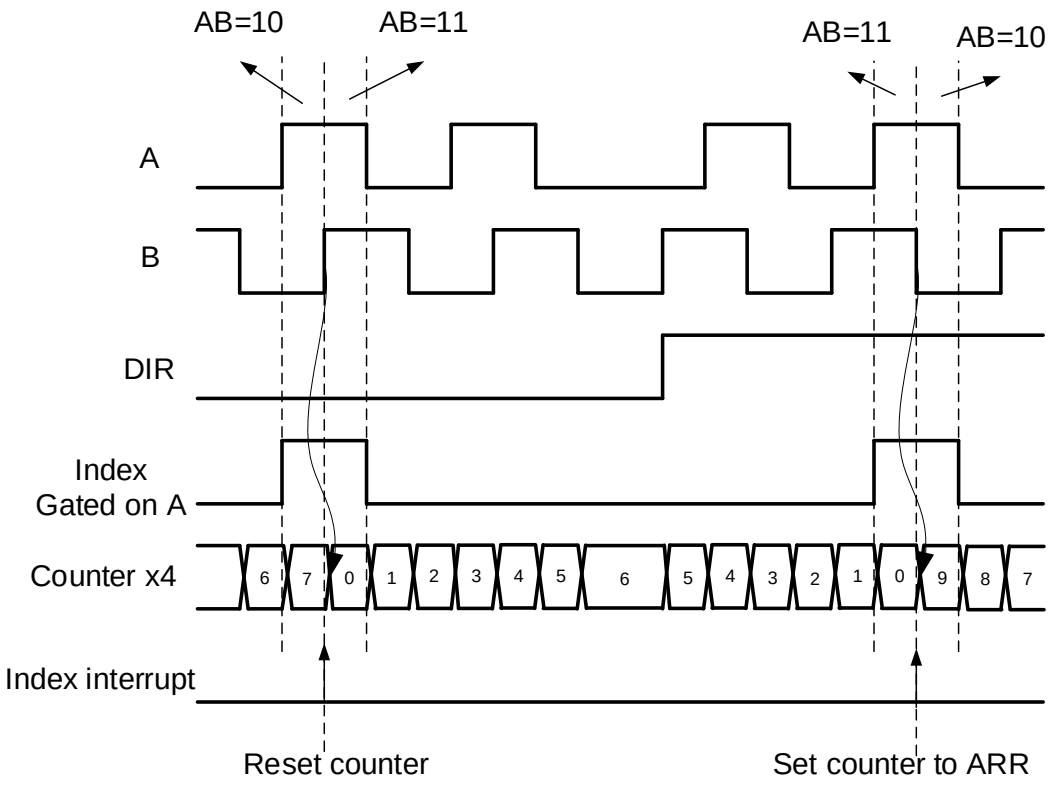


图27-43IPOS[1:0]=11, index gated on A, ARR=9

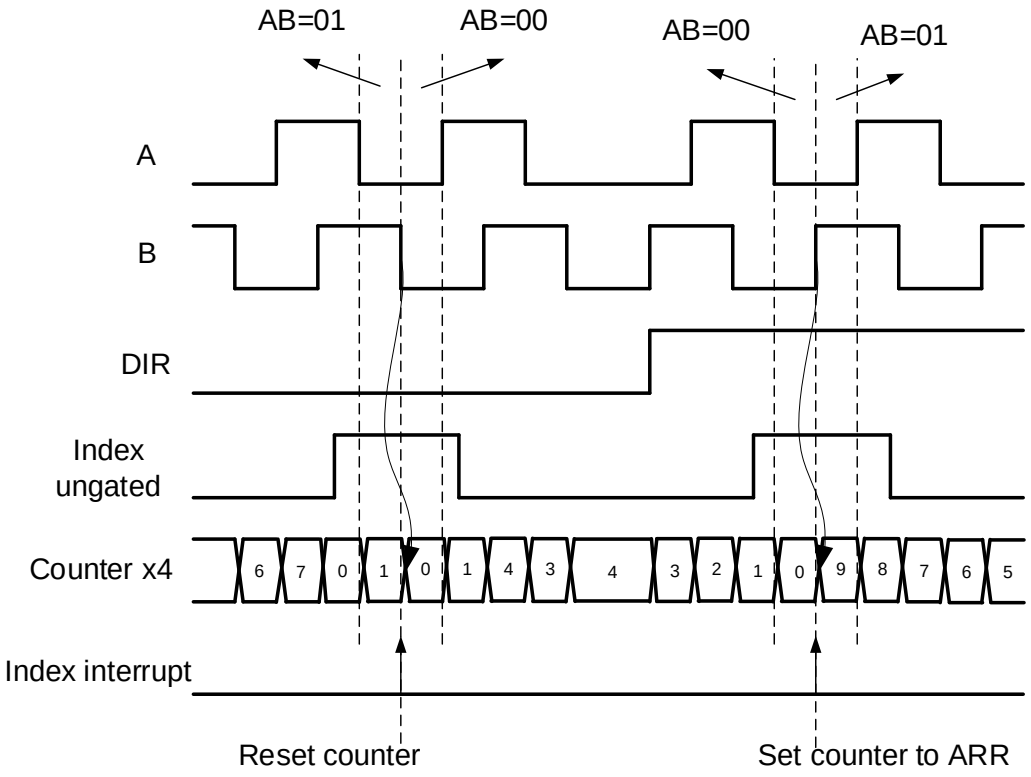


图27-44IPOS[1:0]=00, index ungated, ARR=9

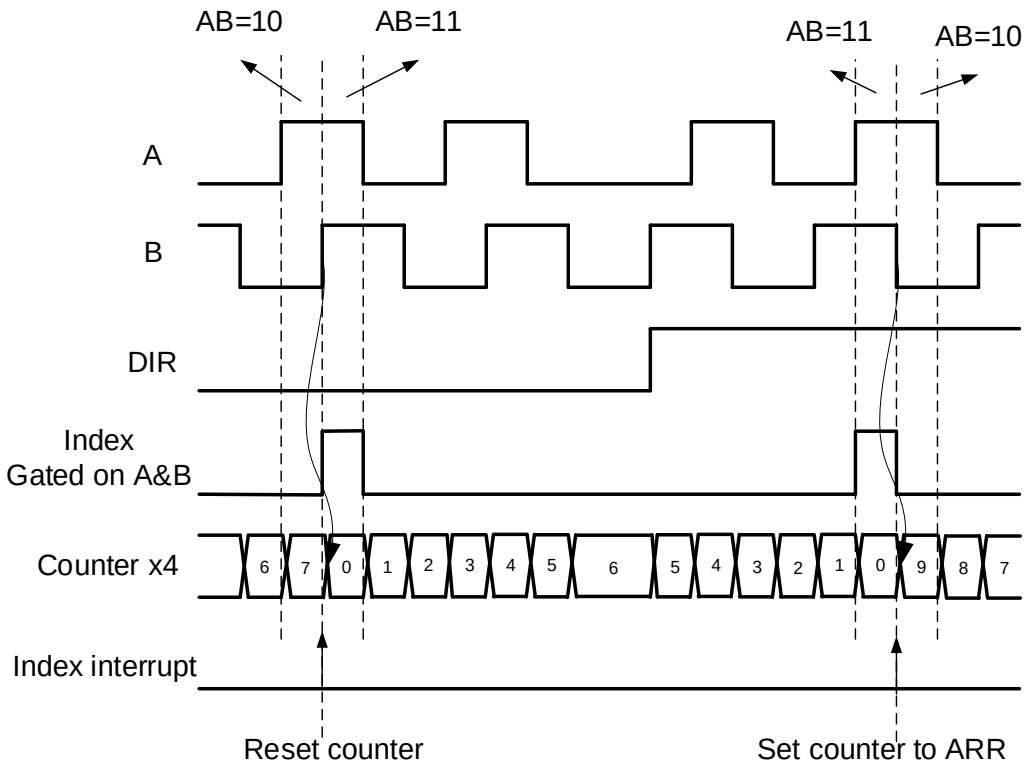


图27-45IPOS[1:0]=11, index gated on A&B, ARR=9

基于同步设计和状态转换记录，编码器模式的index检测允许一定的相位误差。比如下图中，index信号超前B相输入：

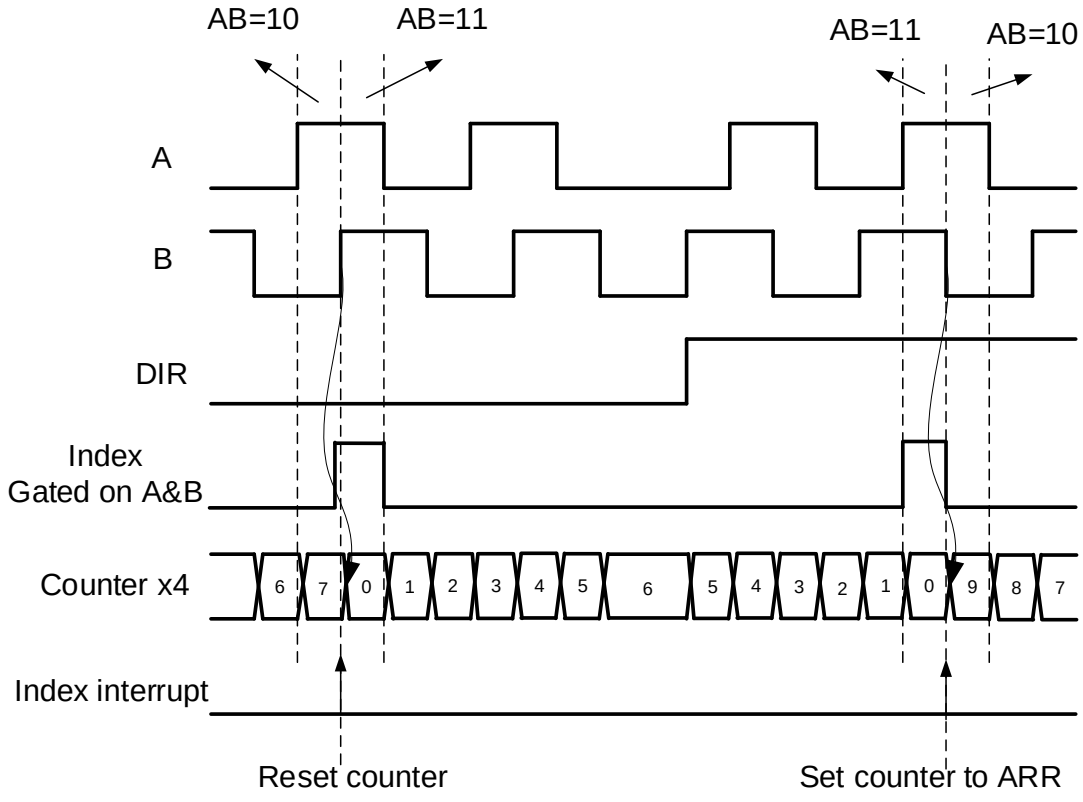


图27-46IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B

下图中则是index滞后B相输入，或者说出现一个窄Z相信号：

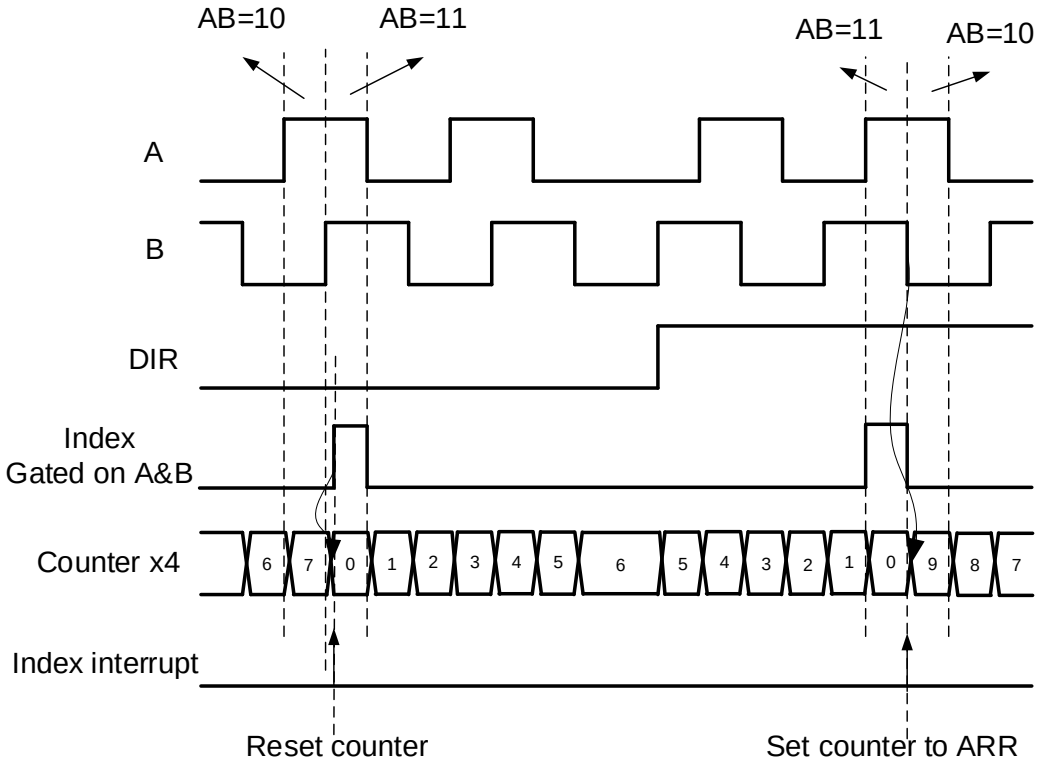


图27-47IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B

上图中index lagging的区域放大图如下：

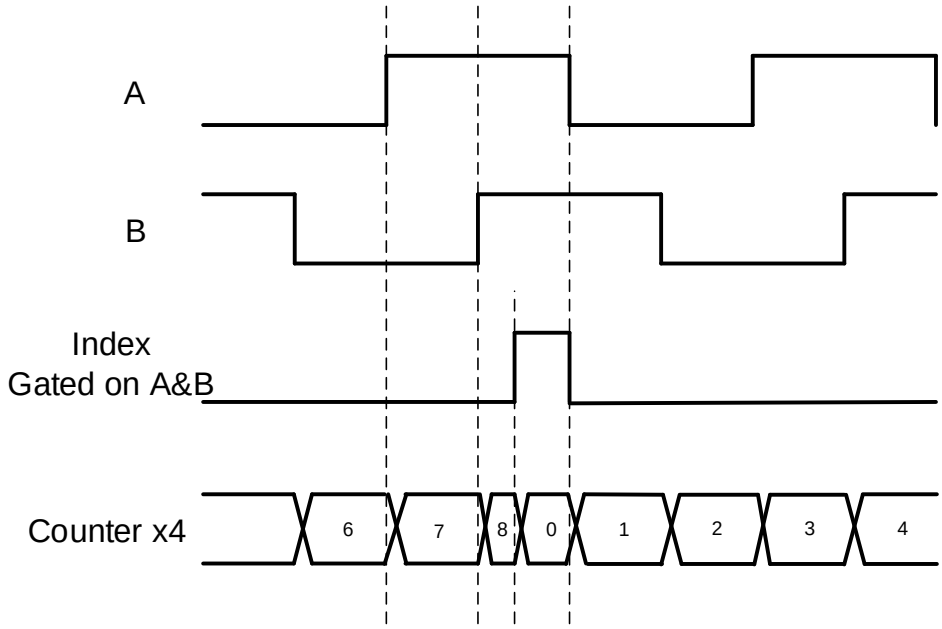


图27-48IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed

下图显示了counter mode x2 和x1的情况下，index如何影响计数值（IPOS[1:0]=01）：

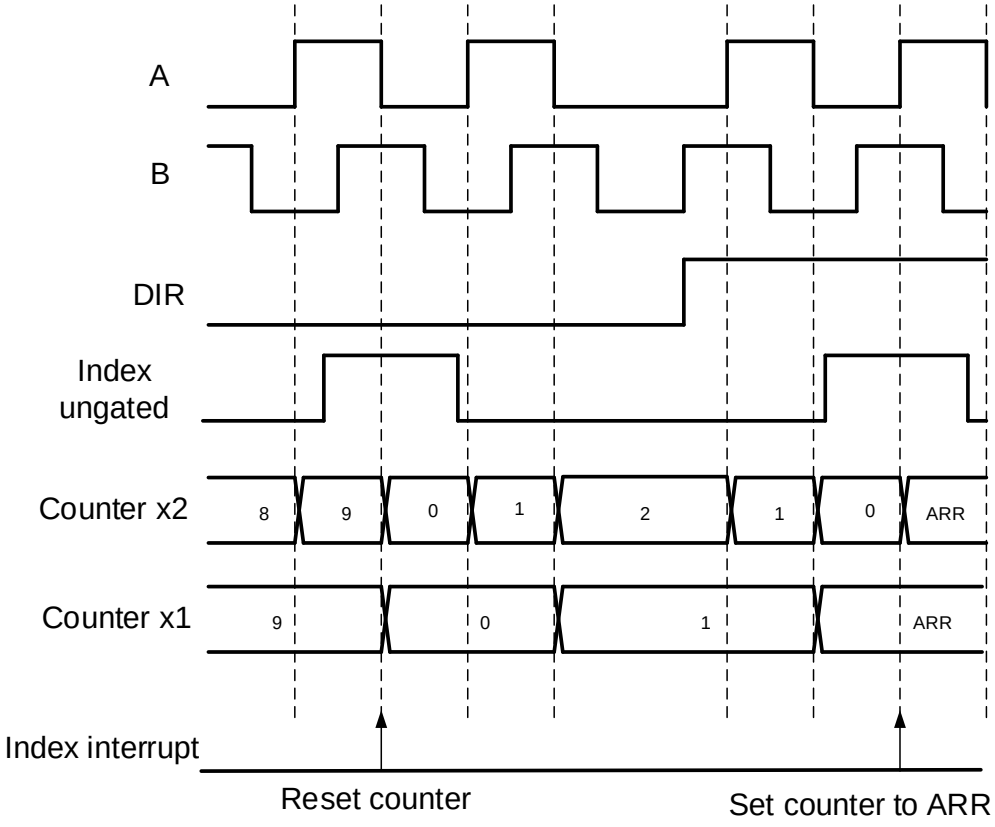


图27-49IPOS[1:0]=01, ungated index

Index的方向敏感性

可以通过GPTIM_ECR寄存器中的IDIR[1:0]位配置定时器只在特定计数方向上响应index输入。

- IDIR=00：向上计数和向下计数时index都有效
- IDIR=01：向上计数时index有效
- IDIR=10：向下计数时index有效

注意：在时钟方向编码器和带方向信息的时钟编码器模式下，IDIR无效。

只响应第一个index

通过置位FIDX寄存器，可以使GPTIM只响应第一个输入的index信号，后续index将被忽略，直到软件清零FIDX。

27.5.14.5 非正交编码模式下的 index 处理

当GPTIM使用编码器接口模式，但是工作类型为非正交编码器，此时需要配置index信号在输入时钟信号的高电平处检测，或者在输入时钟信号的低电平处检测。通过IPOS[0]寄存器完成这个配置。

- IPOS[0]=0：在输入时钟低电平处检测index
- IPOS[0]=1：在输入时钟高电平处检测index

IPOS[1]此时不起作用。

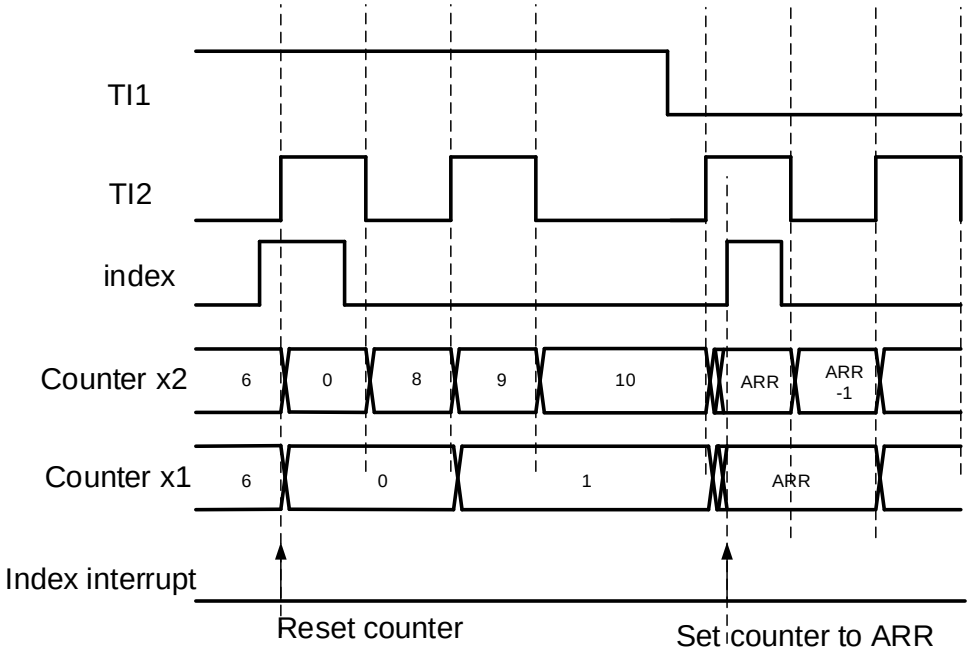


图27-50IPOS[0]=1, clock+direction mode

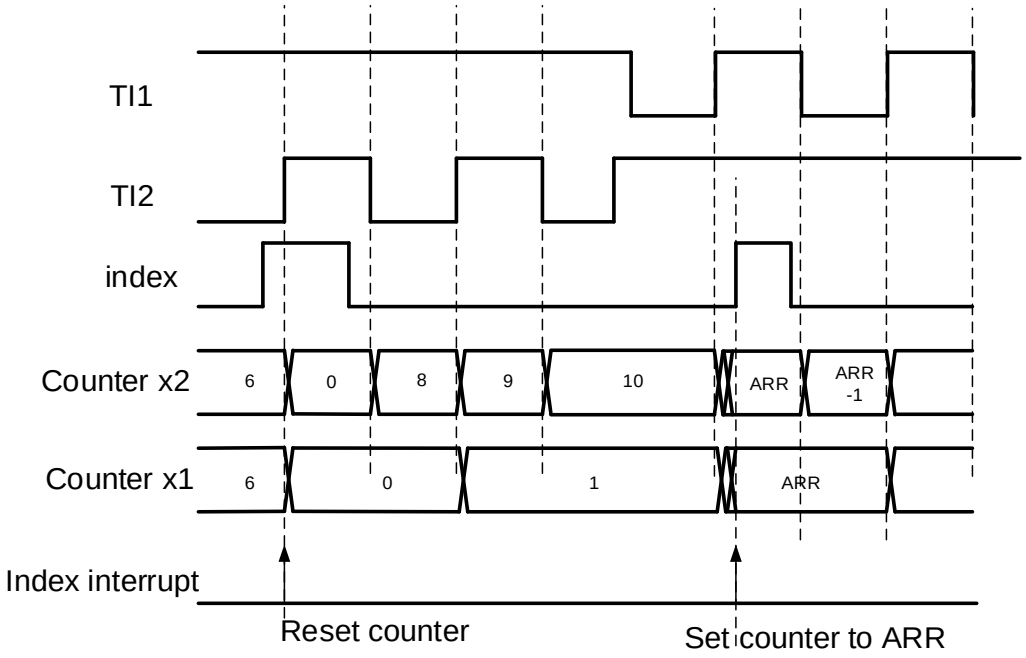


图27-51IPOS[0]=1, directional clock mode

27.5.14.6 正交编码模式错误管理

正交编码器输出的2路正交信号总是以格雷码形式变化，即相邻状态只会变化1bit。如果出现了2bit状态变化，则表示编码错误，此时TERRIF标志置位，并可以产生中断。

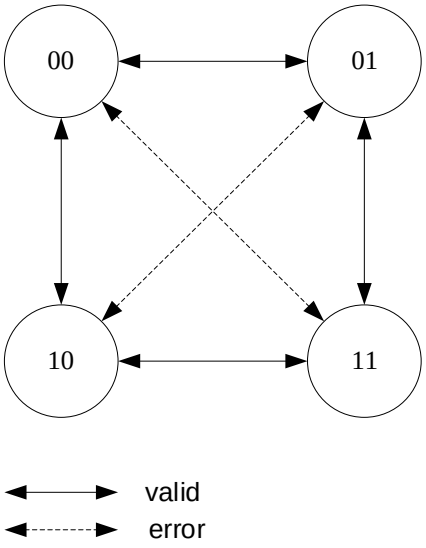


图27-52正交编码状态转移

当编码器计数溢出，而仍未检测到index事件时，表示出现了index错误。此时IERRF标志置位，并且可以产生中断。Index错误标志的置位要延迟到计数器roll-over，以适应窄Z相输入的情况，避免误报警。

下图的例子中，index在计数器等于0的周期内出现，错误标志不会置位。

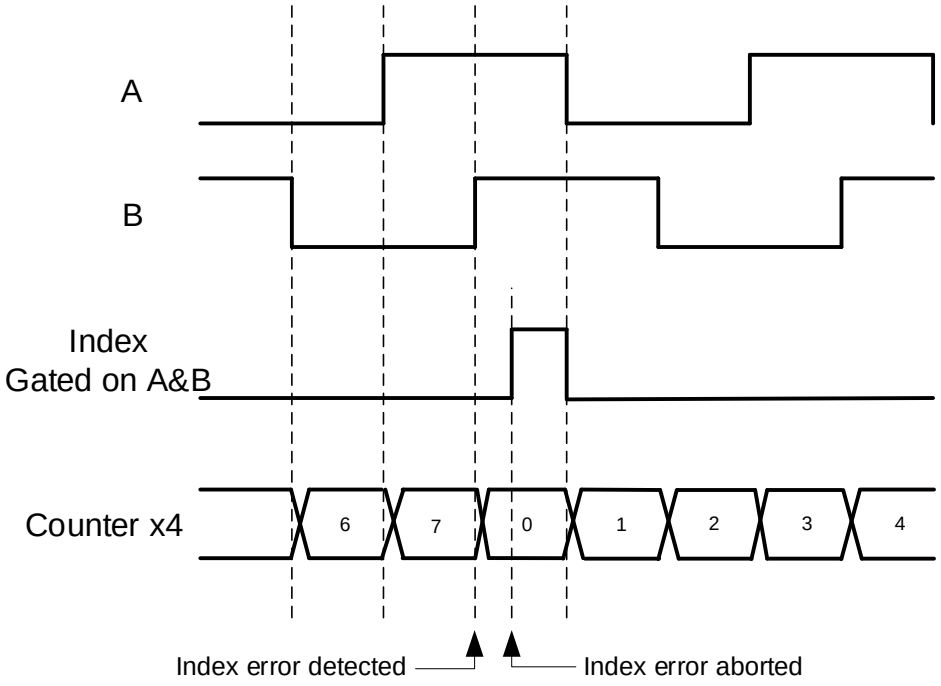


图27-53index错误撤销

下面的例子里，index直到计数器roll-over并且计数值大于0才到来，index错误标志置位。

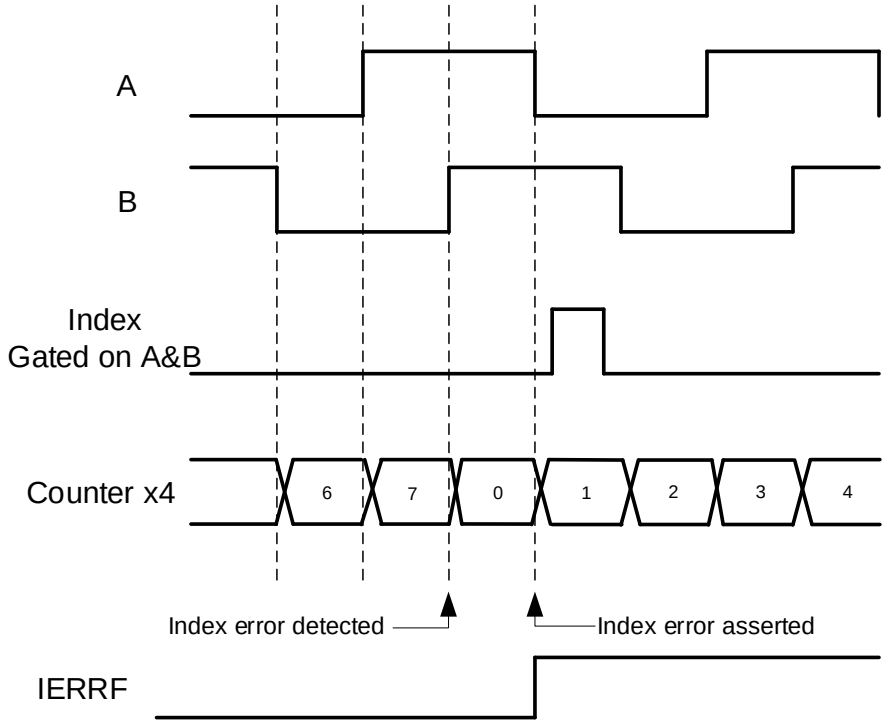


图27-54index错误置位

向下计数时，仅当计数值从1减到0，roll-over到ARR，然后递减时，才会置位index错误。

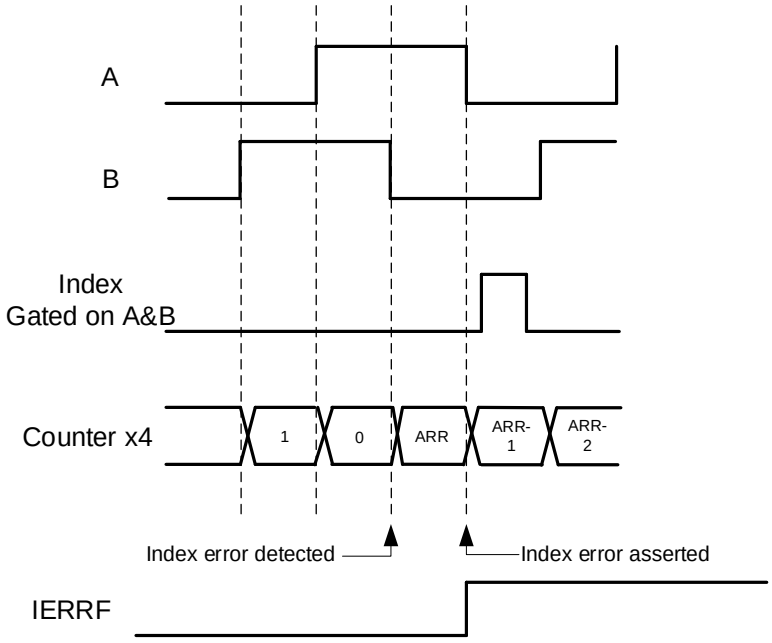


图27-55向下计数时index错误置位

27.5.14.7 Counter mode preload

某些应用中需要实时切换counter mode（x4, x2, x1），可以通过SMS寄存器的preload功能来实现。当SMSPE=1时，使能preload功能，并且根据SMSPS寄存器的值，选择在什么事件发生时进行preload：

- SMSPS=0：preload由update event(UEV)触发，即向上计数时计数器上溢出，或向下计数时计数器下溢出时触发，使用此配置时必须关闭index事件使能（IE=0）
- SMSPS=1：preload发生由index事件触发

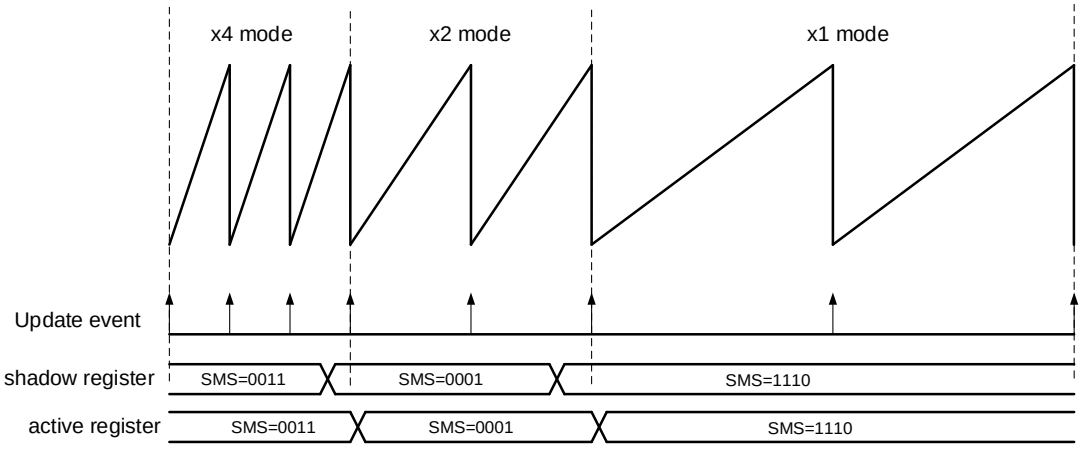


图27-56counter mode preload

27.5.14.8 编码器时钟输出

当SMS配置为编码器模式中的一种，配置MMS[3:0]=1000可以从TRGO输出被decode出来的编码器时钟。这个输出时钟是计数器实际计数频率，比如x4模式下这个时钟是正交编码信号频率的4倍。

27.5.15 GPTIM 从机模式

GPTIM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GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为复位模式，配置GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

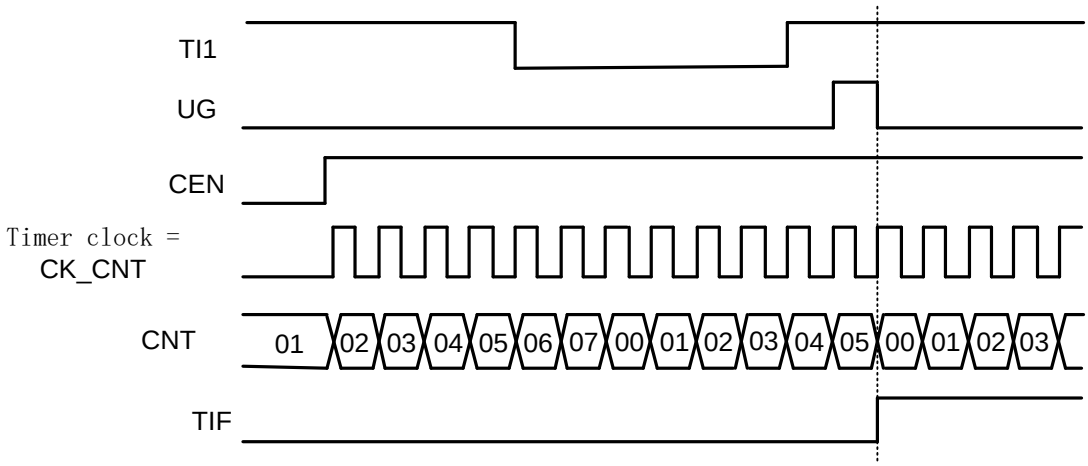


图 27-57 复位模式下的时序

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GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为门控模式，配置GPTIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置GPTIM_CCER.CC1E=1
- 使能计数器，配置GPTIM_CR1.CEN=1

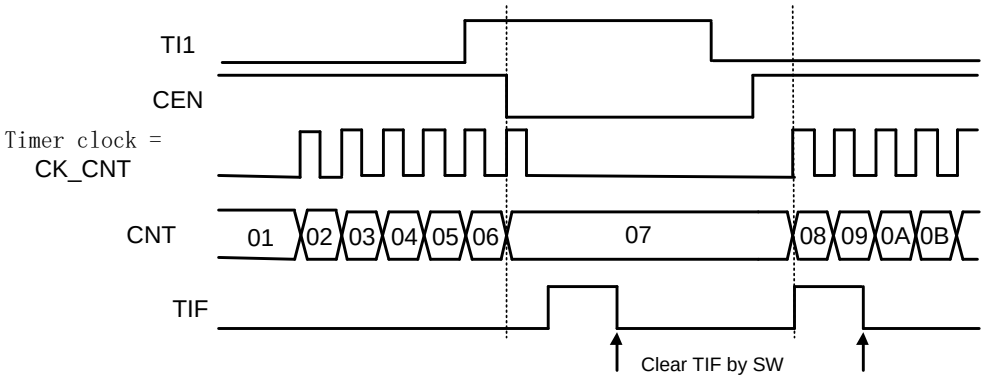


图 27-58 门控模式下的时序

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GPTIM_CH1功能
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置GPTIM_CCER.CC1E=1

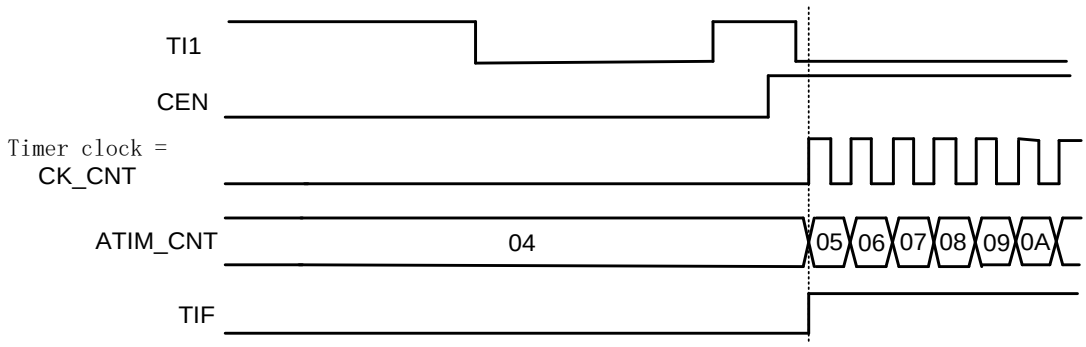


图 27-59 触发器模式下的时序

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GPTIM_CH1，ATIM_ETR功能
- 设置ETP进行沿选择，GPTIM_SMCR.ETP=0
- 设置ETR分频比，配置GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位ECE寄存器，使能外部时钟模式2，GPTIM_SMCR.ECE=1
- 关闭通道使能，配置GPTIM_CCER.CC1E=0确保之后通道配置成功
- 选择输入通道，配置GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置GPTIM_SMCR.TS[2:0]=101，TI1FP1作为TRGI
- 设定从模式控制器为触发模式，配置GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置GPTIM_CCER.CC1E=1

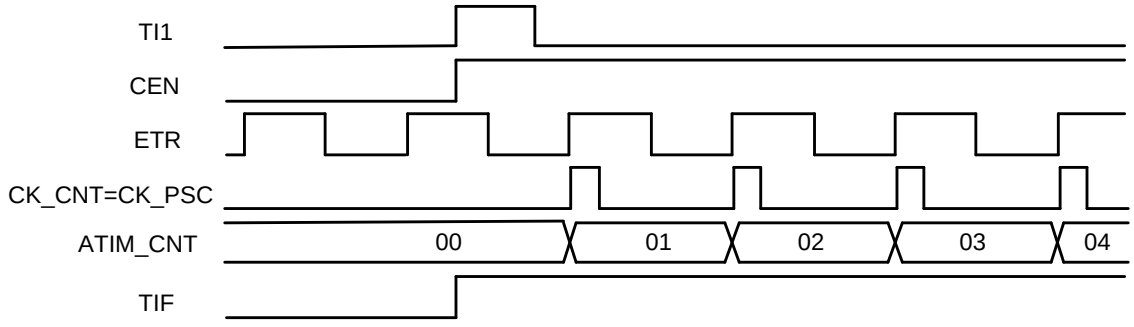


图 27-60 外部时钟模式 2＋触发模式下的时序

27.5.16 DMA 访问

GPTIM支持4种DMA请求，分别为4个CC通道请求。

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。

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
GTIMx_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	

27.5.17 DMA Burst

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GPTIM_DMAR。在特定的定时器事件发生时，GPTIM会连续发射多个DMA请求。每个DMA对GPTIM_DMAR的写操作都会被GPTIM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GPTIM内部的基地址（相对于GPTIM_CR的offset）。

27.5.18 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。

GPTIM_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。

27.5.19 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DCU模块的DBG_TIMx_STOP寄存器定义。

27.6 寄存器

GPTIM0模块起始地址：0x4001_4C00

GPTIM1模块起始地址：0x4001_6400

offset 地址	名称	符号
GPTIM0		
0x00000000	GPTIM0 控制寄存器 1（GPTIM0 Control Register1）	GPTIM0_CR1
0x00000004	GPTIM0 控制寄存器 2（GPTIM0 Control Register2）	GPTIM0_CR2
0x00000008	GPTIM0 从机模式控制寄存器（GPTIM0 Slave Mode Control Register）	GPTIM0_SMCR
0x0000000C	GPTIM0 DMA 和中断使能寄存器（GPTIM0 DMA and Interrupt Enable Register）	GPTIM0_DIER
0x00000010	GPTIM0 状态寄存器（GPTIM0 Interrupt Status Register）	GPTIM0_ISR
0x00000014	GPTIM0 事件产生寄存器（GPTIM0 Event Generation Register）	GPTIM0_EGR
0x00000018	GPTIM0 捕捉/比较模式寄存器 1（GPTIM0 Capture/Compare Mode Register1）	GPTIM0_CCMR1
0x0000001C	GPTIM0 捕捉/比较模式寄存器 2（GPTIM0 Capture/Compare Mode Register2）	GPTIM0_CCMR2
0x00000020	GPTIM0 捕捉/比较使能寄存器（GPTIM0 Capture/Compare Enable Register）	GPTIM0_CCER
0x00000024	GPTIM0 计数器寄存器（GPTIM0 Counter Register）	GPTIM0_CNT
0x00000028	GPTIM0 预分频寄存器（GPTIM0 Prescaler Register）	GPTIM0_PSC

offset 地址	名称	符号
0x0000002C	GPTIM0 自动重载寄存器（GPTIM0 Auto-Reload Register）	GPTIM0_ARR
0x00000034	GPTIM0 捕捉/比较寄存器 1（GPTIM0 Capture/Compare Register1）	GPTIM0_CCR1
0x00000038	GPTIM0 捕捉/比较寄存器 2（GPTIM0 Capture/Compare Register2）	GPTIM0_CCR2
0x0000003C	GPTIM0 捕捉/比较寄存器 3（GPTIM0 Capture/Compare Register3）	GPTIM0_CCR3
0x00000040	GPTIM0 捕捉/比较寄存器 4（GPTIM0 Capture/Compare Register4）	GPTIM0_CCR4
0x00000054	GPTIM0 编码器模式控制寄存器	GPTIM0_ECR
0x00000058	GPTIM0 输入选择寄存器	GPTIM0_TISEL
0x0000005C	GPTIM0 辅助功能寄存器	GPTIM0_AFR
0x000000F8	GPTIM0 DMA 控制寄存器（GPTIM0 DMA Control Register）	GPTIM0_DCR
0x000000FC	GPTIM0 DMA 访问寄存器（GPTIM0 DMA access Register）	GPTIM0_DMAR
GPTIM1		
0x00000000	GPTIM1 控制寄存器 1（GPTIM1 Control Register1）	GPTIM1_CR1
0x00000004	GPTIM1 控制寄存器 2（GPTIM1 Control Register2）	GPTIM1_CR2
0x00000008	GPTIM1 从机模式控制寄存器（GPTIM1 Slave Mode Control Register）	GPTIM1_SMCR
0x0000000C	GPTIM1 DMA 和中断使能寄存器（GPTIM1 DMA and Interrupt Enable Register）	GPTIM1_DIER
0x00000010	GPTIM1 状态寄存器（GPTIM1 Interrupt Status Register）	GPTIM1_ISR
0x00000014	GPTIM1 事件产生寄存器（GPTIM1 Event Generation Register）	GPTIM1_EGR
0x00000018	GPTIM1 捕捉/比较模式寄存器 1（GPTIM1 Capture/Compare Mode Register1）	GPTIM1_CCMR1
0x0000001C	GPTIM1 捕捉/比较模式寄存器 2（GPTIM1 Capture/Compare Mode Register2）	GPTIM1_CCMR2
0x00000020	GPTIM1 捕捉/比较使能寄存器（GPTIM1 Capture/Compare Enable Register）	GPTIM1_CCER
0x00000024	GPTIM1 计数器寄存器（GPTIM1 Counter Register）	GPTIM1_CNT
0x00000028	GPTIM1 预分频寄存器（GPTIM1 Prescaler Register）	GPTIM1_PSC
0x0000002C	GPTIM1 自动重载寄存器（GPTIM1 Auto-Reload Register）	GPTIM1_ARR
0x00000034	GPTIM1 捕捉/比较寄存器 1（GPTIM1 Capture/Compare Register1）	GPTIM1_CCR1
0x00000038	GPTIM1 捕捉/比较寄存器 2（GPTIM1 Capture/Compare Register2）	GPTIM1_CCR2
0x0000003C	GPTIM1 捕捉/比较寄存器 3	GPTIM1_CCR3

offset 地址	名称	符号
	（GPTIM1 Capture/Compare Register3）	
0x00000040	GPTIM1 捕捉/比较寄存器 4 （GPTIM1 Capture/Compare Register4）	GPTIM1_CCR4
0x00000054	GPTIM1 编码器模式控制寄存器	GPTIM1_ECR
0x00000058	GPTIM1 输入选择寄存器	GPTIM1_TISEL
0x0000005C	GPTIM1 辅助功能寄存器	GPTIM1_AFR
0x000000F8	GPTIM1 DMA 控制寄存器 （GPTIM0 DMA Control Register）	GPTIM1_DCR
0x000000FC	GPTIM1 DMA 访问寄存器 （GPTIM0 DMA access Register）	GPTIM1_DMAR

27.6.1 GPTIMx 控制寄存器 1（GPTIMx_CR1）

名称	GPTIMx_CR1(x=0,1)							
Offset	0x00000000							
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						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
位权限	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:10	-	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）(Counter 609lock Divider) 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT 10：tDTS=4*tCK_INT 11：RFU，禁止使用
7	ARPE	Auto-reload 预装载使能(Auto-Reload Preload Enable) 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(Counter Mode Selection) 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(counter Direction)

位号	助记符	功能描述
		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(One Pulse Mode) 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(Update Request Selection) 0：以下事件能够产生 update 中断 - 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
1	UDIS	禁止 update(Update Disable) 0：使能 update 事件；以下事件发生时产生 update 事件 - 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(Counter Enable) 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

27.6.2 GPTIMx 控制寄存器 2（GPTIMx_CR2）

名称	GPTIMx_CR2(x=0,1)							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						MMS_H	
位权限	U-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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS			CCDS	-		
位权限	R/W-0	R/W-000			R/W-0	U-0		

位号	助记符	功能描述
31:26	-	RFU，未实现，读为 0
25	MMS_H	参考 MMS_L
24:8	-	RFU，未实现，读为 0
7	TI1S	通道 1 输入源选择(Timer Input 1 Selection) 0：GPTIMx_CH1 输入通道 1

位号	助记符	功能描述
		1：GPTIMx_CH1, CH2, CH3 异或后输入通道 1
6:4	MMS_L	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(Master Mode Selection) 000：GPTIM_EGR 的 UG 寄存器被用作 TRGO 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 1000：编码器时钟输出，编码器模式的计数时钟信号从 TRGO 输出，仅当 SMS 配置为编码器模式中的一种时有效 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(Capture/Compare DMA Selection) 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
2:0	-	RFU，未实现，读为 0

27.6.3 GPTIMx 从机模式控制寄存器（GPTIMx_SMCR）

名称	GPTIMx_SMCR(x=0,1)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						SMSPS	SMSPE
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	ETP	ECE	ETPS		ETF			
位权限	R/W-0	R/W-0	R/W-00		R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS			SMS			
位权限	R/W-0	R/W-000			R/W-0000			

位号	助记符	功能描述
31:26	-	RFU：未实现，读为 0
25	**SMSPS**	SMS preload 触发源选择 0：SMS 更新由 update event 触发 1：SMS 更新由 index event 触发
24	**SMSPE**	SMS preload 使能 0：禁止 SMS preload 1：使能 SMS preload

位号	助记符	功能描述
23:22	**RFU**	RFU：未实现，读为 0
21:20	**TS_H**	参考 TS_L
19:17	**RFU**	RFU：未实现，读为 0
16	**OCCS**	OCREF 清零选择 0：使用 OCREF_CLR 信号来清零 1：使用 ETRF 信号来清零
15	**ETP**	外部触发信号极性配置(External Trigger Polarity) 0：高电平或上升沿有效 1：低电平或下降沿有效
14	**ECE**	外部时钟使能(External Clock Enable) 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(External Trigger Prescaler) 外部触发信号 ETRP 的频率最多只能是 GPTIM 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(External Trigger Filter) 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/从模式(Master Slave Mode) 0：无动作 1：TRGI 触发的动作被延迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于单个外部事件对多个定时器进行同步的情况。
6:4	**TS_L**	与 TS_H 组成 5bit 内部触发选择，用于选择同步计数器的触发源 00000：内部触发信号（ITR0） 00001：内部触发信号（ITR1） 00010：内部触发信号（ITR2） 00011：内部触发信号（ITR3） 00100：TI1 边沿检测（TI1F_ED） 00101：滤波后 TI1（TI1FP1） 00110：滤波后 TI2（TI2FP2）

位号	助记符	功能描述
		00111：外部触发输入（ETRF） 01000：内部触发信号（ITR4） 01001：内部触发信号（ITR5） 01010：内部触发信号（ITR6） 01011：内部触发信号（ITR7） 01100：内部触发信号（ITR8） 01101：内部触发信号（ITR9） 01110：内部触发信号（ITR10） 01111：内部触发信号（ITR11） 10000：内部触发信号（ITR12） 10001：内部触发信号（ITR13） 10010：内部触发信号（ITR14） 10011：内部触发信号（ITR15） Others：RFU 注意：仅当 SMS=0000 即禁止从机模式的情况下，可以改写 TS 寄存器
3:0	**SMS**	从机模式选择(Slave Mode Selection) 0000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 0001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 0010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 0011：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 0100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 0101：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 0110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 0111：外部时钟模式 1；TRGI 上升沿直接驱动计数器 1000：组合的复位+触发模式；TRGI 上升沿初始化计数器，产生更新事件并启动计数 1001：组合的闸门+触发模式；TRGI 为高电平时，计数时钟使能，当 TRGI 变低电平时，计数器复位且停止 1010：Encoder 模式：时钟+方向，x2 1011：Encoder 模式：时钟+方向，x1 1100：Encoder 模式：带方向的时钟，x2 1101：Encoder 模式：带方向的时钟，x1 1110：Quadrature Encoder x1 mode，仅使用 TI1FP1 边沿计数，有效沿由 CC1P 选择 1111：Quadrature Encoder x1 mode，仅使用 TI2FP2 边沿计数，有效沿由 CC2P 选择 其他：RFU

27.6.4 GPTIMx DMA 和中断使能寄存器（GPTIMx_DIER）

名称	GPTIMx_DIER(x=0,1)

Offset	0x0000000C							
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	TERRIE	IERRIE	DIRIE	IDXIE	CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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	-	TDE	-	CC4DE	CC3DE	CC2DE	CC1DE	UDE
位权限	U-0	R/W-0	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	-	TIE	-	CC4IE	CC3IE	CC2IE	CC1IE	UIE
位权限	U-0	R/W-0	U-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
31:20	-	RFU：未实现，读为 0
23	**TERRIE**	状态转移错误中断使能，用于 encoder 模式，1 有效
22	**IERRIE**	Index 错误中断使能，用于 encoder 模式，1 有效
21	-	RFU：未实现，读为 0
20	**IDXIE**	Index 事件中断使能，用于 encoder 模式，1 有效
19	**CC4BURSTEN**	捕捉比较通道 4 的 DMA 模式配置 (CC4 Burst Enable) 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 (CC3 Burst Enable) 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 (CC2 Burst Enable) 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 (CC1 Burst Enable) 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(Triggered DMA Enable) 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）
13	-	RFU：未实现，读为 0
12	**CC4DE**	捕捉比较通道 4 的 DMA 请求使能 (CC4 DMA Enable) 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 (CC3 DMA Enable) 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 (CC2 DMA Enable) 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 (CC1 DMA Enable)

位号	助记符	功能描述
		0：禁止 CC1 DMA 请求 1：允许 CC1 DMA 请求
8	**UDE**	Update Event DMA 请求使能 (Update event DMA Enable) 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	**-**	RFU：未实现，读为 0
6	**TIE**	触发事件中断使能 (Trigger event Interrupt Enable) 0：禁止触发事件中断 1：允许触发事件中断
5	**-**	RFU：未实现，读为 0
4	**CC4IE**	捕捉/比较通道 4 中断使能 (CC4 Interrupt Enable) 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 (CC3 Interrupt Enable) 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 (CC2 Interrupt Enable) 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 (CC1 Interrupt Enable) 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 (Update event Interrupt Enable) 0：禁止 Update 事件中断 1：允许 Update 事件中断

27.6.5 GPTIMx 状态寄存器（GPTIMx_ISR）

名称	GPTIMx_ISR(x=0,1)							
Offset	0x00000010							
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	TERRIF	IERRIF	DIRIF	IDXIF	-			
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	-		QDIR	CC4OF	CC3OF	CC2OF	CC1OF	-
位权限	U-0		R-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	TIF	QDCIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
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	助记符	功能描述
31:24	**-**	RFU：未实现，读为 0
23	**TERRIF**	状态转移错误中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
22	**IERRIF**	Index 错误中断标志，用于 encoder 模式，硬件置位，软件写 1

位号	助记符	功能描述
		清零
21	**DIRIF**	方向变化中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
20	**IDXIF**	Index 事件中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
19:13	**-**	RFU：未实现，读为 0
12	**CC4OF**	捕捉/比较通道 4 的 Overcapture 中断(Over-Capture Interrupt Flag for CC4, write 1 to clear) 参考 CC1OF
11	**CC3OF**	捕捉/比较通道 3 的 Overcapture 中断(Over-Capture Interrupt Flag for CC3, write 1 to clear) 参考 CC1OF
10	**CC2OF**	捕捉/比较通道 2 的 Overcapture 中断(Over-Capture Interrupt Flag for CC2, write 1 to clear) 参考 CC1OF
9	**CC1OF**	捕捉/比较通道 1 的 Overcapture 中断(Over-Capture Interrupt Flag for CC1, write 1 to clear) 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
8:7	**-**	RFU：未实现，读为 0
6	**TIF**	触发事件中断标志，硬件置位，软件写 1 清零 (Trigger event Interrupt Flag, write 1 to clear)
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	**CC4IF**	捕捉/比较通道 4 中断标志 (CC4 Interrupt Flag, write 1 to clear) 参考 CC1IF
3	**CC3IF**	捕捉/比较通道 3 中断标志 (CC3 Interrupt Flag, write 1 to clear) 参考 CC3IF
2	**CC2IF**	捕捉/比较通道 2 中断标志 (CC2 Interrupt Flag, write 1 to clear) 参考 CC2IF
1	**CC1IF**	捕捉/比较通道 1 中断标志 (CC1 Interrupt Flag, write 1 to clear) 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。(Update event Interrupt Flag,write 1 to clear) 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

27.6.6 GPTIMx 事件产生寄存器（GPTIMx_EGR）

名称	GPTIMx_EGR(x=0,1)							
Offset	0x00000014							
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	-	TG	-			CC2G	CC1G	UG
位权限	U-0	W-0	U-0			W-0	W-0	W-0

位号	助记符	功能描述
31:7	-	RFU：未实现，读为 0
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 (Trigger Generate)
5:3	-	RFU：未实现，读为 0
2	CC2G	捕捉/比较通道 2 软件触发，参考 CC1G (CC2 Generate)
1	CC1G	捕捉/比较通道 1 软件触发 (CC1 Generate) 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 (User Generate) 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。

27.6.7 GPTIMx 捕捉/比较模式寄存器 1（GPTIMx_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	GPTIMx_CCMR1(x=0,1)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							OC2M_H
位权限	U-0							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							OC1M_H
位权限	U-0							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M			OC2PE	OC2FE	CC2S	

	IC2F				IC2PSC		CC2S	
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	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
	IC1F				IC1PSC		CC1S	
位权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

输出比较模式

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24	**OC2M_H**	参考 OC2M_L
23:17	**RFU**	RFU：未实现，读为 0
16	**OC1M_H**	参考 OC1M_L
15	**OC2CE**	输出比较 2 清零使能，参考 OC1CE (OC2 Clear Enable)
14:12	**OC2M_L**	输出比较 2 模式配置，参考 OC1M_L (OC2 Mode)
11	**OC2PE**	输出比较 2 预装载使能，参考 OC1PE (OC2 Preload Enable)
10	**OC2FE**	输出比较 2 快速使能，参考 OC1FE (OC2 Fast Enable)
9:8	**CC2S**	捕捉/比较 2 通道选择 (CC2 channel Selection) 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(OC2 Clear Enable) 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC1M_L**	输出比较 1 模式配置，与 OC1M_H 组成 4bit 输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 (OC1 Mode) 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 1000：可重触发的 OPM 模式 1，向上计数时若计数器停止输出高（active）否则输出 pwm 模式 1 波形，向下计数时若计数器停止输出低（inactive）否则输出 pwm 模式 1 波形，配合 sms 配置为复位+触发模式使用。 1001：可重触发的 OPM 模式 2，向上计数时若计数器停止输出低（inactive）否则输出 pwm 模式 2 波形，向下计数时若计数器停止输出高（active）否则输出 pwm 模式 2 波形，配合 sms 配置为复位+触发模式使用。 1010：RFU

位号	助记符	功能描述
		1011：RFU 1100：组合 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 或 oc2ref 的逻辑，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1101：组合 PWM 模式 2，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 与 oc2ref 的逻辑，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1110：非对称 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形在向上计数时为 oc1ref 波形，向下计数时为 oc2ref 波形，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1111：非对称 PWM 模式 2，oc1ref 波形与 pwm 模式 2 一致，oc1refc 波形在向上计数时为 oc1ref 波形，向下计数时为 oc2ref 波形，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组）
3	**OC1PE**	输出比较 1 预装载使能 (OC1 Preload Enable) 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 (OC1 Fast Enable) 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 (CC1 channel Selection) 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

输入捕捉模式

位号	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 (IC2 Filter)
11:10	**IC2PSC**	输入捕捉 2 预分频 (IC2 Prescaler)
9:8	**CC2S**	捕捉/比较 2 通道选择 (Capture/Compare2 channel Selection) 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 (IC1 Filter) 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

位号	助记符	功能描述
		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 (IC1 Prescaler) 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 (Capture/Compare1 channel Selection) 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

27.6.8 GPTIMx 捕捉/比较模式寄存器 2（GPTIMx_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	GPTIMx_CCMR2(x=0,1)							
Offset	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							OC4M_H
位权限	U-0							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							OC3M_H
位权限	U-0							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC2F				IC2PSC		CC4S	
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		CC3S	
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

位号	助记符	功能描述
31:25	-	RFU：未实现，读为 0
24	**OC4M_H**	参考 OC4M_L
23:17	**RFU**	RFU：未实现，读为 0
16	**OC3M_H**	参考 OC3M_L
15	**OC4CE**	输出比较 4 清零使能，参考 OC3CE (OC4 Clear Enable)
14:12	**OC4M_L**	输出比较 4 模式配置，参考 OC3M (OC4 Mode)
11	**OC4PE**	输出比较 4 预装载使能，参考 OC3PE (OC4 Preload Enable)
10	**OC4FE**	输出比较 4 快速使能，参考 OC3FE (OC4 Fast Enable)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 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 1 清零使能(OC3 Clear Enable) 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	**OC3M_L**	输出比较 3 模式配置，参考 OC1M_L
3	**OC3PE**	输出比较 3 预装载使能 (OC3 Preload Enable) 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 (OC3 Fast Enable) 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 (CC3 channel Selection) 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

位号	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 (IC4 Filter)
11:10	**IC4PSC**	输入捕捉 4 预分频 (IC4 Prescaler)
9:8	**CC4S**	捕捉/比较 4 通道选择 (CC4 channel Selection) 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 3 滤波 (IC3 Filter) 此寄存器定义 TI3 的采样频率和滤波长度

位号	助记符	功能描述
		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(IC3 Prescaler) 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	**CC3S**	捕捉/比较 3 通道选择 (CC3 channel Selection) 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

27.6.9 GPTIMx 捕捉/比较使能寄存器（GPTIMx_CCER）

名称	GPTIMx_CCER(x=0,1)							
Offset	0x00000020							
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	-		CC3P	CC3E
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		CC2P	CC2E	-		CC1P	CC1E
位权限	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0

位号	助记符	功能描述
31:14	-	RFU：未实现，读为 0

位号	助记符	功能描述
13	**CC4P**	捕捉/比较 4 输出极性，参考 CC1P (CC4 Polarity)
12	**CC4E**	捕捉/比较 4 输出使能，参考 CC1E (CC4 output Enable)
11:10	-	RFU：未实现，读为 0
9	**CC3P**	捕捉/比较 3 输出极性，参考 CC1P (CC3 Polarity)
8	**CC3E**	捕捉/比较 3 输出使能，参考 CC1E (CC3 output Enable)
7:6	-	RFU：未实现，读为 0
5	**CC2P**	捕捉/比较 2 输出极性，参考 CC1P (CC2 Polarity)
4	**CC2E**	捕捉/比较 2 输出使能，参考 CC1E (CC2 output Enable)
3:2	-	RFU：未实现，读为 0
1	**CC1P**	捕捉/比较 1 输出极性 (CC1 Polarity) **CC1 通道配置为输出时：** 0：OC1 高有效 1：OC1 低有效 **CC1 通道配置为输入时：** CC1NP/CC1P 用于选择 TI1FP1 和 TI2FP1 的极性 00：非取反/上升沿 01：取反/下降沿 10：保留，不要使用 11：非取反，上下沿都有效
0	**CC1E**	捕捉/比较 1 输出使能 (CC1 output Enable) **CC1 通道配置为输出时** 0：OC1 输出关闭，Ocx=0，Ocx_EN=0 1：Ocx=OCxREF+极性选择，Ocx_EN=1 **CC1 通道配置为输入时** 0：关闭捕捉功能 1：使能捕捉功能

标准 Ocx 通道的输出控制位

CcxE 位	Ocx 输出状态
0	禁止输出（Ocx=0，Ocx_EN=0）
1	Ocx=OCxREF + 极性，Ocx_EN=1

27.6.10 GPTIMx 计数器寄存器（GPTIMx_CNT）

名称	GPTIMx_CNT(x=0,1)							
Offset	0x00000024							
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/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							

位权限	R/W-0000 0000

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**CNT**	计数器值(Counter)

27.6.11 GPTIMx 预分频寄存器（GPTIMx_PSC）

名称	GPTIMx_PSC(x=0,1)							
offset	0x00000028 + x*0x400							
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	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC[7:0]							
位权限	R/W-0000 0000							

位号	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(Counter Clock Prescaler) $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$ 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

27.6.12 GPTIMx 自动重载寄存器（GPTIMx_ARR）

名称	GPTIMx_ARR(x=0,1)							
Offset	0x0000002C							
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-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-1111 1111							

位号	助记符	功能描述

位号	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(Auto-Reload Register) 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

27.6.13 GPTIMx 捕捉/比较寄存器 1（GPTIMx_CCR1）

名称	GPTIMx_CCR1(x=0,1)							
Offset	0x00000034							
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	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(Capture/Compare channel 1 Register) **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

27.6.14 GPTIMx 捕捉/比较寄存器 2（GPTIMx_CCR2）

名称	GPTIMx_CCR2(x=0,1)							
Offset	0x00000038							
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	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(Capture/Compare channel 2 Register) **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

27.6.15 GPTIMx 捕捉/比较寄存器 3（GPTIMx_CCR3）

名称	GPTIMx_CCR3(x=0,1)							
Offset	0x0000003C							
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	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	R/W-0000 0000							

位号	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(Capture/Compare channel 3 Register) **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

27.6.16 GPTIMx 捕捉/比较寄存器 4（GPTIMx_CCR4）

名称	GPTIMx_CCR4(x=0,1)							
Offset	0x00000040							
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	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4[7:0]							
位权限	R/W-0000 0000							

位号	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(Capture/Compare channel 4 Register) **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

27.6.17 GPTIM 编码器模式控制寄存器（GPTIM_ECR）

名称	GPTIM_ECR							
地址	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								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IPOS		FIDX			IDIR		IE
位权限	RW-00		RW-0			RW-00		RW-0

Bit	助记符	功能描述
31:25	--	RFU：未实现，读为 0
7:6	**IPOS**	Index positioning 定义正交编码模式时 index 事件复位或置位计数器的相位 仅 SMS=0001/0010/0011/1110/1111 时有效 00：操作相位 AB=00 01：操作相位 AB=01 10：操作相位 AB=10 11：操作相位 AB=11 非正交编码器模式时，IPOS[0]用于定义在什么时钟电平时复位计数器 x0：时钟低电平时复位计数器

Bit	助记符	功能描述
		x1：时钟高电平时复位计数器
5	**FIDX**	First Index 1：仅第一个 index 有效 0：所有 index 有效
4:3		
2:1	**IDIR**	Index 方向 00：index 在任意计数方向时复位计数器 01：index 仅在向上计数时复位计数器 10：index 仅在向下计数时复位计数器 11：RFU
0	**IE**	Index 使能 0：禁止 index 功能 1：使能 index 功能

27.6.18 GPTIM 输入选择寄存器（GPTIM_TISEL）

名称	GPTIM_TISEL							
地址	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				TI4SEL			
位权限	U-0				RW-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				TI3SEL			
位权限	U-0				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				TI2SEL			
位权限	U-0				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TI1SEL			
位权限	U-0				RW-0000			

Bit	助记符	功能描述
31:28	**--**	RFU：未实现，读为 0
27:24	**TI4SEL**	ti4_in[15:0]输入选择 0000: ATIM_CH4 0001: ti4_in1 …. 1111: ti4_in15
23:20	**--**	RFU：未实现，读为 0
19:16	**TI3SEL**	ti3_in[15:0]输入选择 0000: ATIM_CH3 0001: ti3_in1 …. 1111: ti3_in15
15:12	**--**	RFU：未实现，读为 0
11:8	**TI2SEL**	ti2_in[15:0]输入选择 0000: ATIM_CH2

Bit	助记符	功能描述
		0001: ti2_in1 …. 1111: ti2_in15
7:4	--	RFU：未实现，读为 0
3:0	**TI1SEL**	ti1_in[15:0]输入选择 0000: ATIM_CH1 0001: ti1_in1 …. 1111: ti1_in15

27.6.19 GPTIM 辅助功能寄存器（GPTIM_AFR）

名称	GPTIM_AFR							
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	ETRSEL				-	OCCLR_SEL		
位权限	RW-0000				U-0	RW-000		
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	-							
位权限	U-0							

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:20	**ETRSEL**	ETR 输入信号选择 0000：GPTIM_ETR 0001：GPTIM_ETR_IN1 … 1111：GPTIM_ETR_IN15
19	--	RFU：未实现，读为 0
18:16	**OCCLR_SEL**	OCREF_CLR 输入选择，用于选择使用什么片内信号产生 OCREF 清零信号 000：COMP0_OUT 001：COMP1_OUT 010：COMP2_OUT 其他：RFU
15:0	--	RFU：未实现，读为 0

27.6.1 GPTIMx DMA 控制寄存器（GPTIMx_DCR）

名称	GPTIMx_DCR(x=0,1)
Offset	0x000000F8

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			R/W-0 0000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DBA				
位权限	U-0			R/W-0 0000				

位号	助记符	功能描述
31:13	-	RFU：未实现，读为 0
12:8	**DBL**	DMA Burst 长度 (DMA Burst Length) 对 GPTIM_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 00010：长度=3 00011：长度=4 00100：长度=5 00101：长度=6 00110：长度=7 00111：长度=8 01000：长度=9 01001：长度=10 01010：长度=11 01011：长度=12 01100：长度=13 01101：长度=14 01110：长度=15 01111：长度=16 10000：长度=17 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 (DMA Burst offset Address) 00000：GPTIM_CR1 00001：GPTIM_CR2 00010：GPTIM_SMCR …… 注意：当 DBA+DBL 超出了 GPTIM 寄存器地址范围，则实际 burst 传输到 GPTIM 最高寄存器地址后自动停止，即 burst 长度会缩短。

27.6.2 GPTIMx DMA 访问寄存器（GPTIMx_DMAR）

名称	GPTIMx_DMAR(x=0,1)							
Offset	0x000000FC							
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	DMAR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15:0	**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 GPTIM_DMAR，GPTIM 会根据 DBL 的值产生多次 DMA 请求

28 定时器阵列单元（TAU）

28.1 概述

TAU是16bit定时器组成的阵列，支持以下功能

- 共8个独立的16bit向上计数器（TAU00~TAU07）
- 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。

28.2 结构框图

TAU整体结构示意图如下

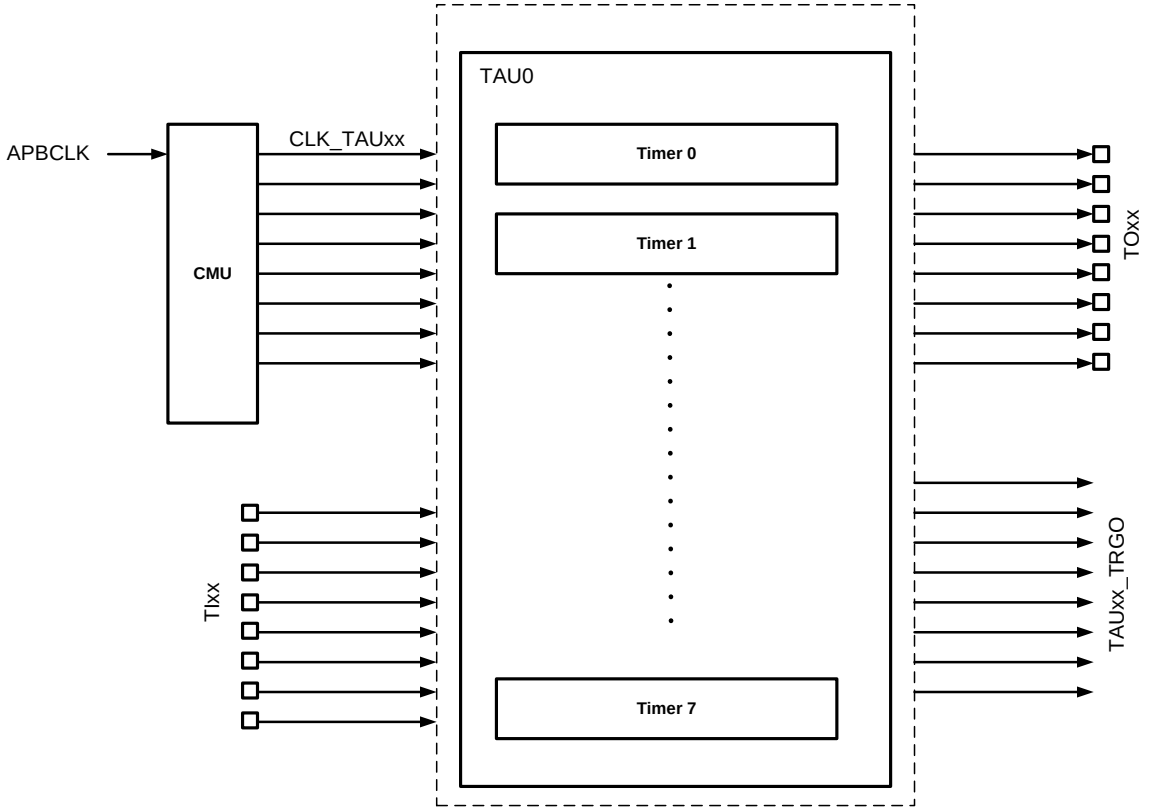


图 28-1TAU 结构框图

独立定时器结构框图如下：

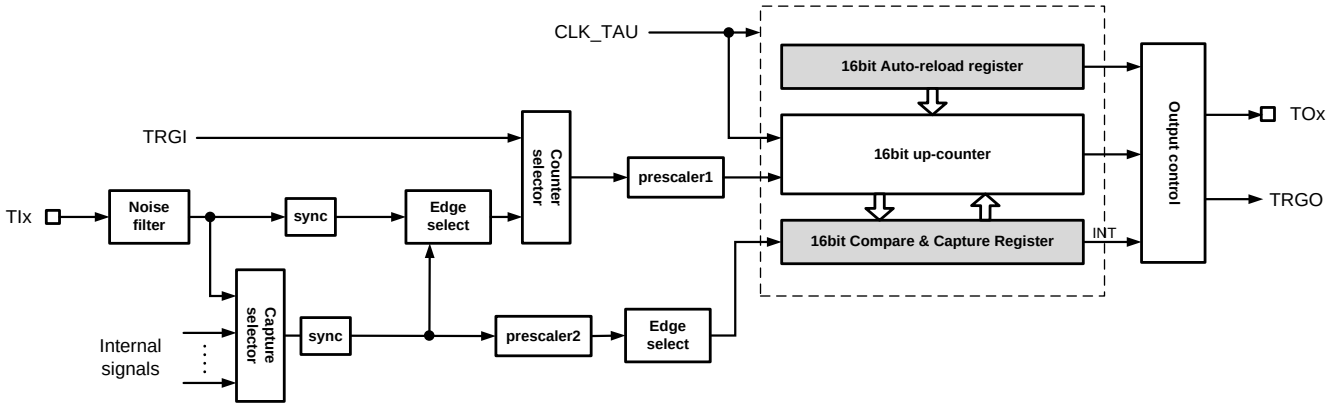


图 28-2TimerX 结构框图

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。

28.3 功能描述

28.3.1 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，即置位外设总线时钟TAU寄存器，参见CMU章节。

通过TAUxx_RST寄存器可以实现每个定时器的独立复位，参见RMU章节。如果想要使能TAU0中的Timer3工作，应清零TAU03_RST寄存器。

28.3.2 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

28.3.3 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

28.3.4 16 位计数模式

定时器计数模式启动和计数溢出时将初值寄存器的值加载到计数器中。计数值达到0xFFFF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

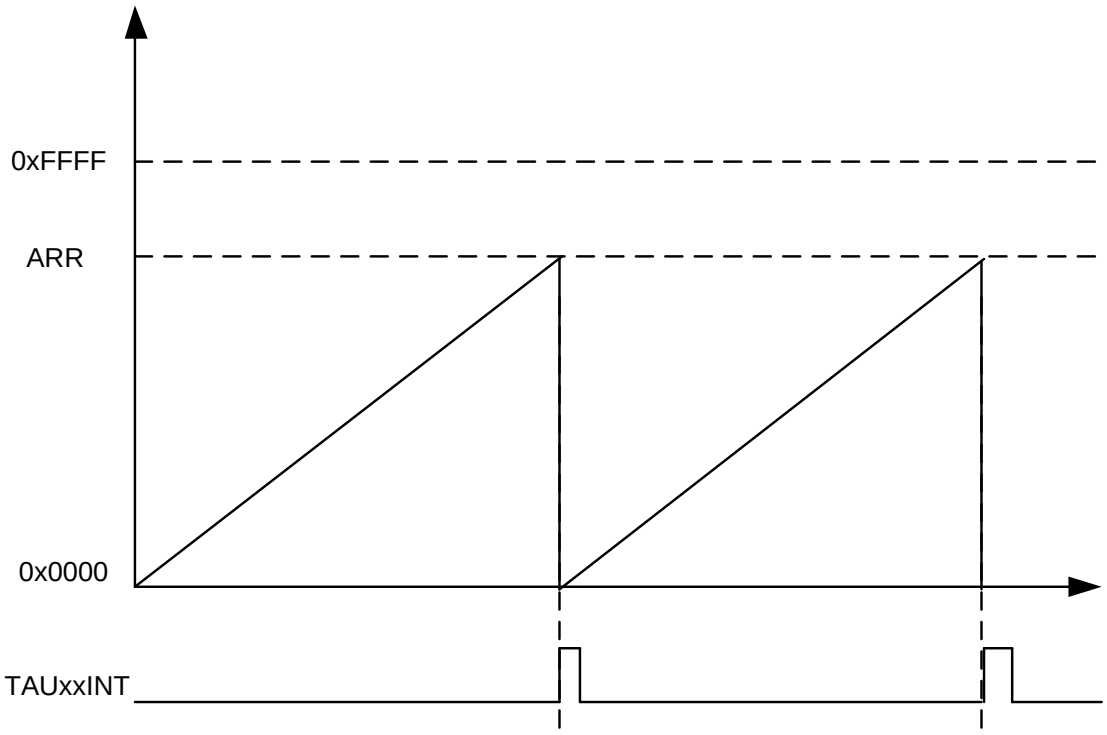


图 28-3 16bit 连续定时功能

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。

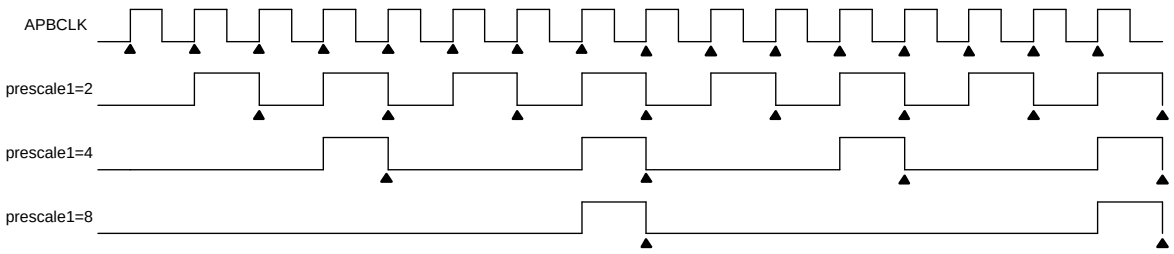


图 28-4 定时器内部时钟预分频计数

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。

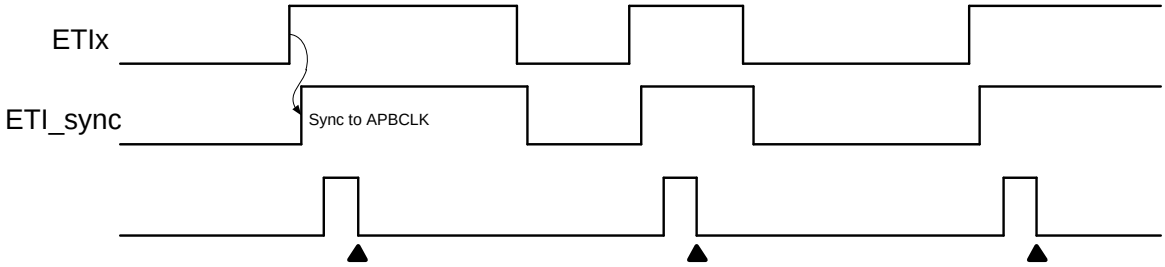


图 28-5 定时器外部事件上升沿递增时序

下图为ETIx上升下降沿事件计数，并且分频系数为2的例子。

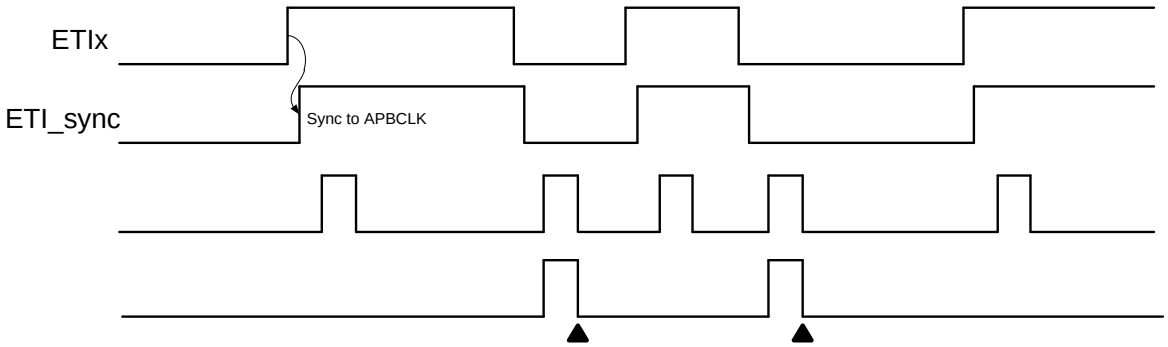


图 28-6 定时器外部事件上升下降沿递增，2 分频时序

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。

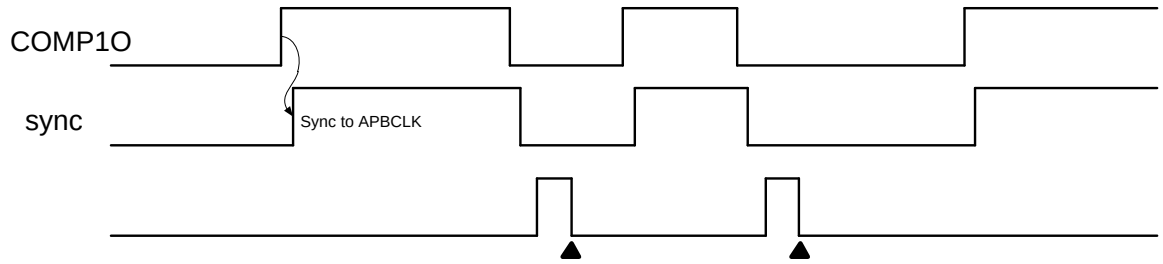


图 28-7 定时器内部信号下降沿递增

28.3.5 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。

28.3.6 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。

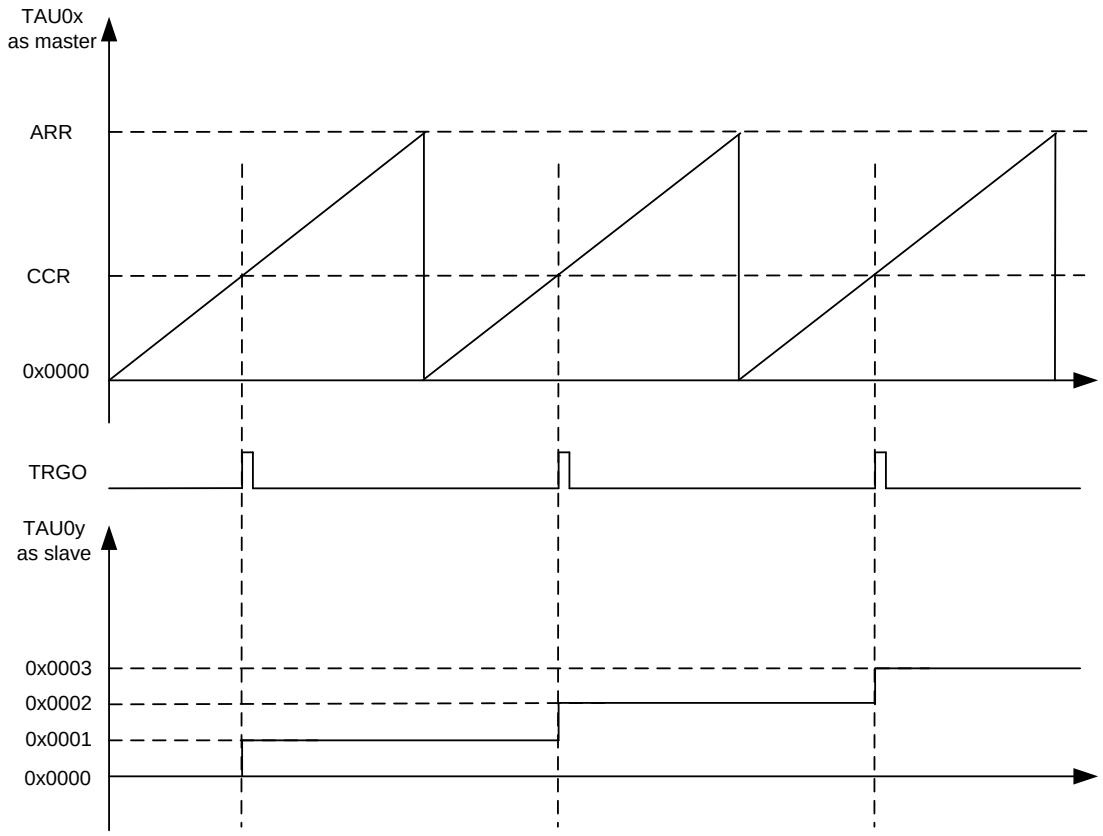


图 28-8 定时器级联计数（trigger to increment）

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。

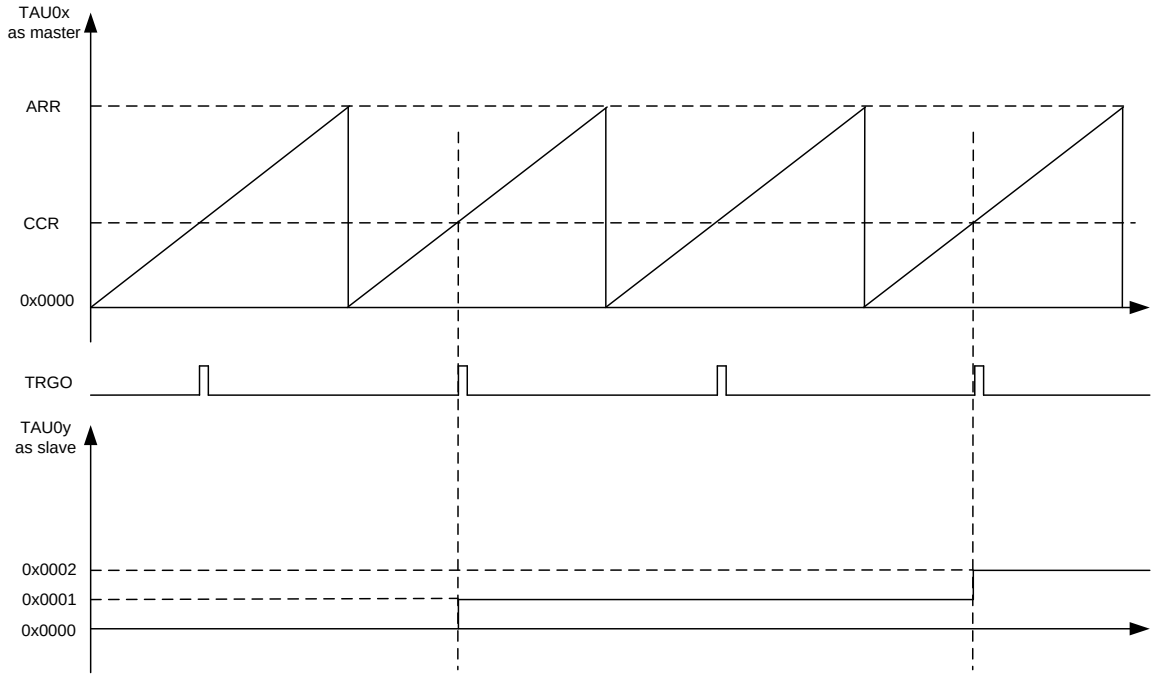


图 28-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1

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。

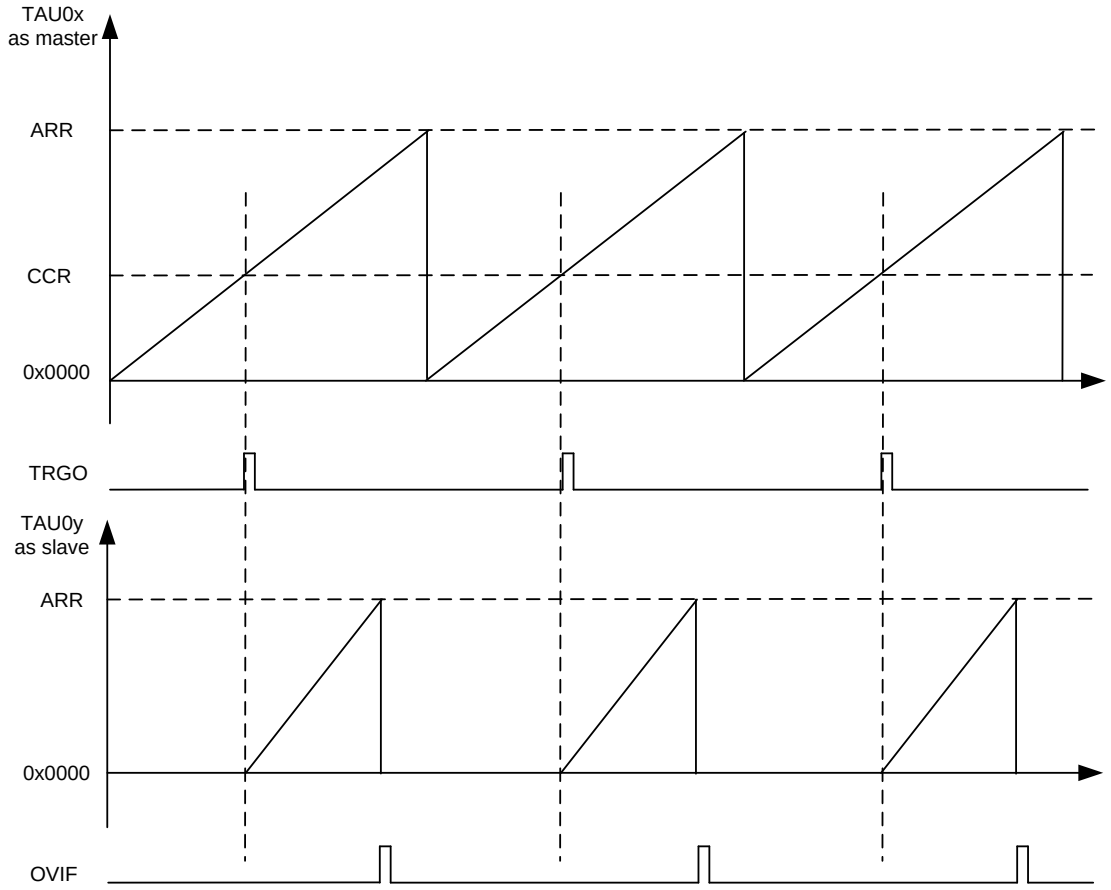


图 28-10 定时器级联计数（trigger to start）

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。

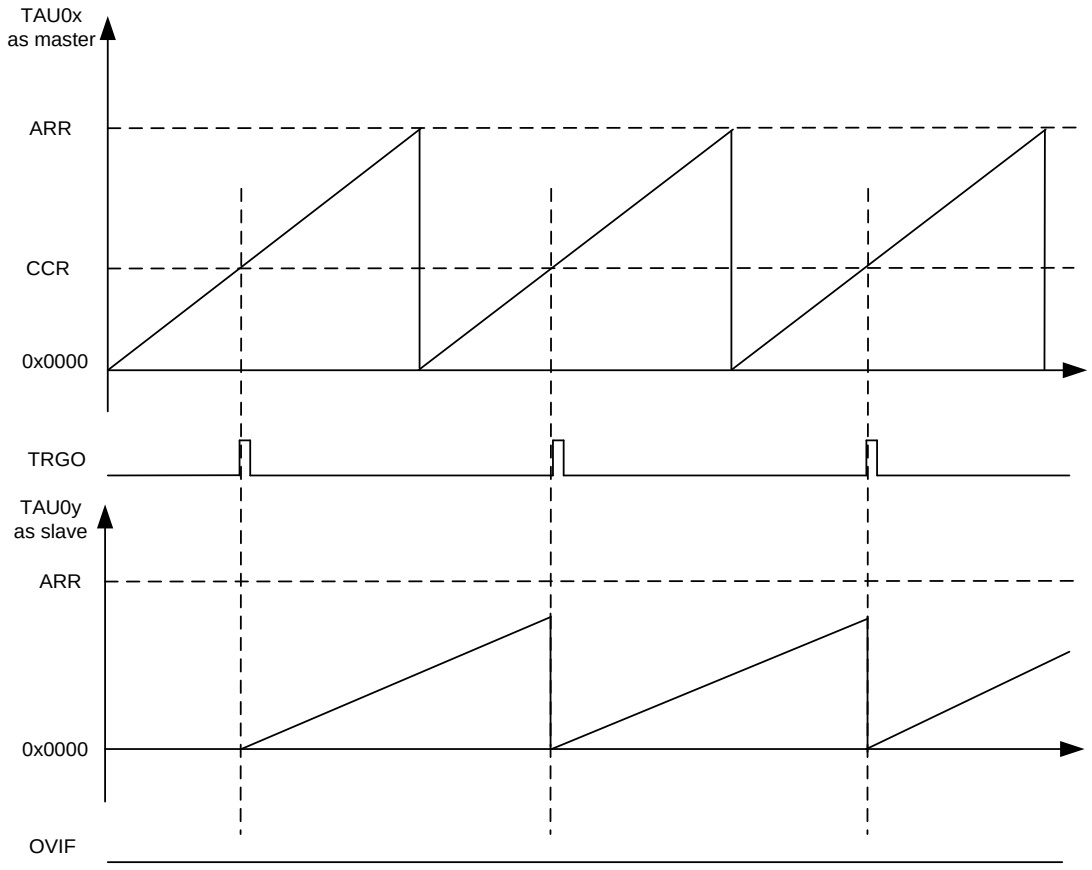


图 28-11 定时器级联计数（trigger to start），从机无法产生溢出

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。

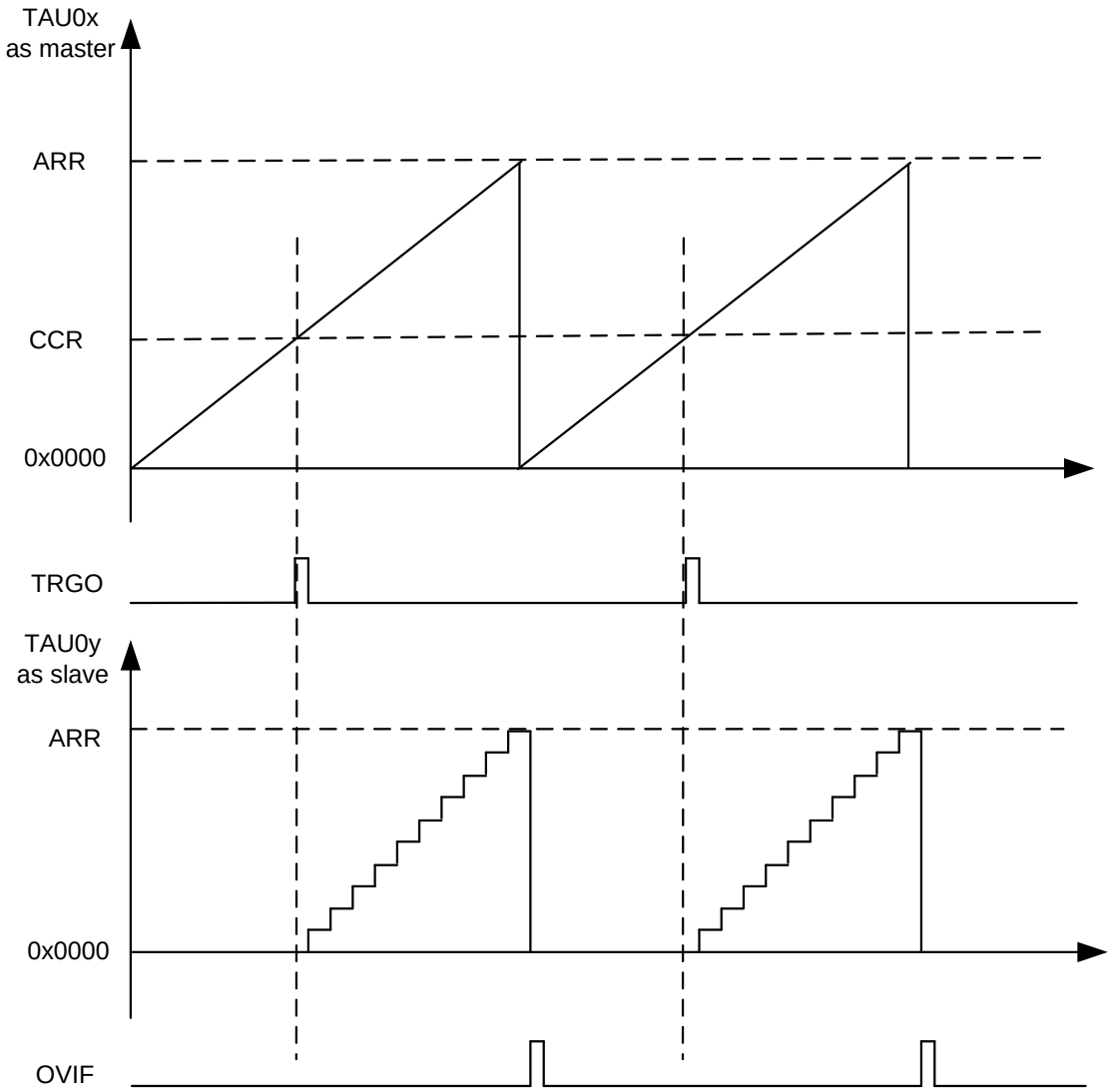


图 28-12 定时器级联计数（trigger to start），从机计数预分频

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
6	Timer0	Timer3	Timer4	Timer5

TAU0 定时器#	TS			
	00	01	10	11
7	Timer0	Timer4	Timer5	Timer6

表 28-1 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。

28.3.7 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_CCRy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。

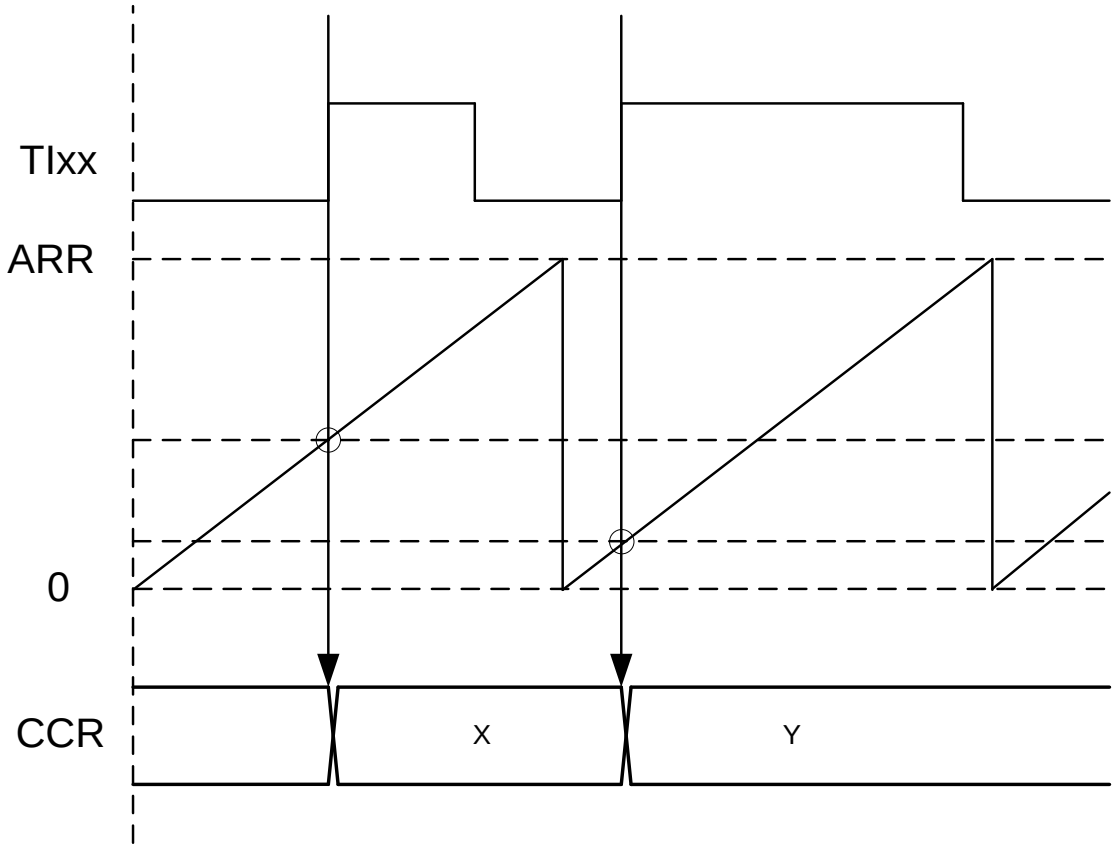


图 28-13 定时器捕捉外部输入信号上升沿

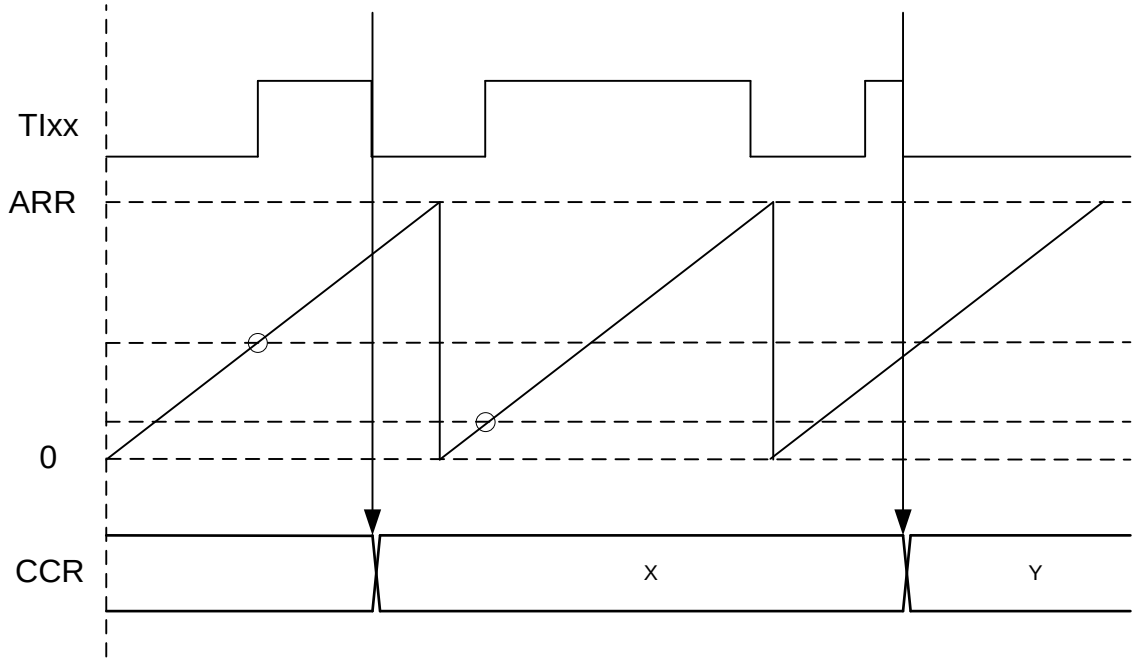


图 28-14 定时器 2 分频捕捉外部输入信号下降沿

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。

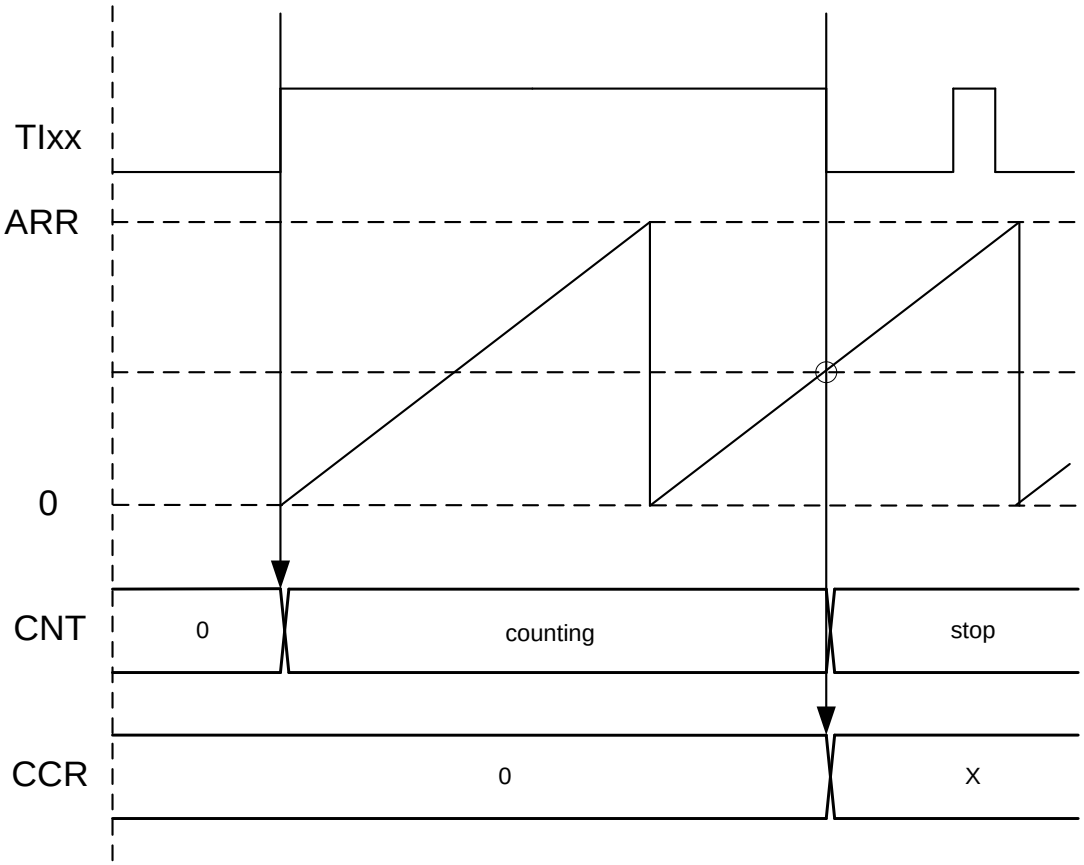


图 28-15 定时器脉冲宽度高电平捕捉

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	-
	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	-
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer4	000	TI04
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	UART0_RXD
	110	UART1_RXD
	111	UART2_RXD
Timer5	000	TI05
	001	LPUART1_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	SENT0_RX
	110	SENT1_RX
	111	UART3_RXD
Timer6	000	TI06
	001	UART0_RXD
	010	UART1_RXD
	011	UART2_RXD
	100	UART3_RXD
	101	SENT0_RX
	110	SENT1_RX
	111	-
Timer7	000	TI07
	001	UART0_RXD
	010	UART1_RXD
	011	UART2_RXD
	100	UART3_RXD
	101	SENT0_RX
	110	SENT1_RX

	111	-

表 28-2 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。

28.3.8 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：

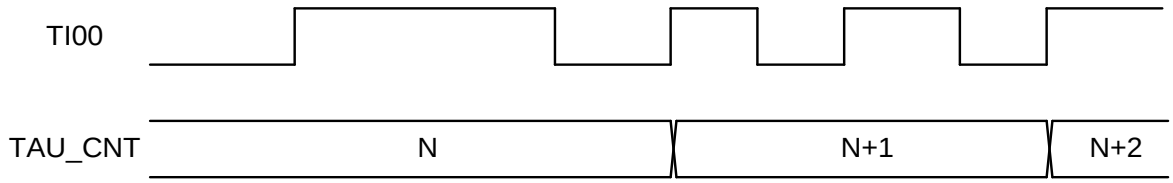


图 28-16TI00 二分频计数

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：

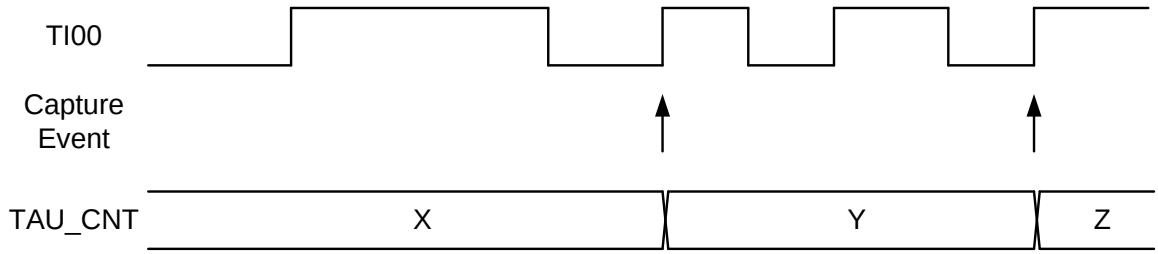


图 28-17TI00 二分频捕捉

28.3.9 16 位 PWM 输出

单个TAU支持独立16bit PWM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：

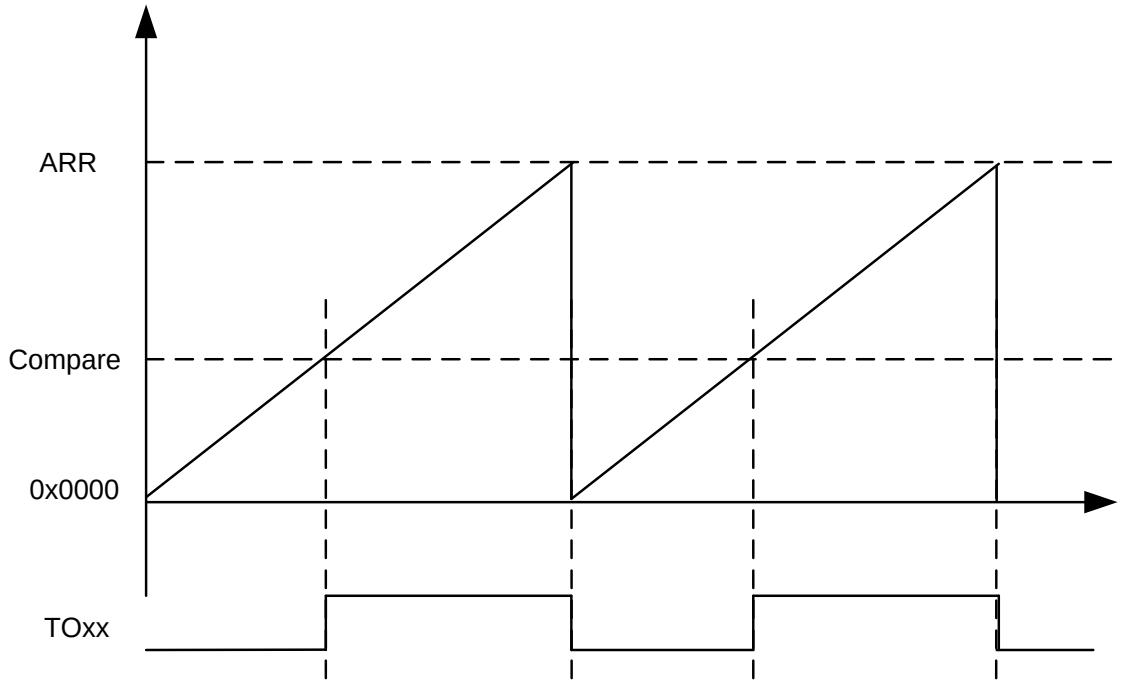


图 28-1816 位 PWM 输出示意图

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-$Duty_{PWM}$。

28.3.10 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。

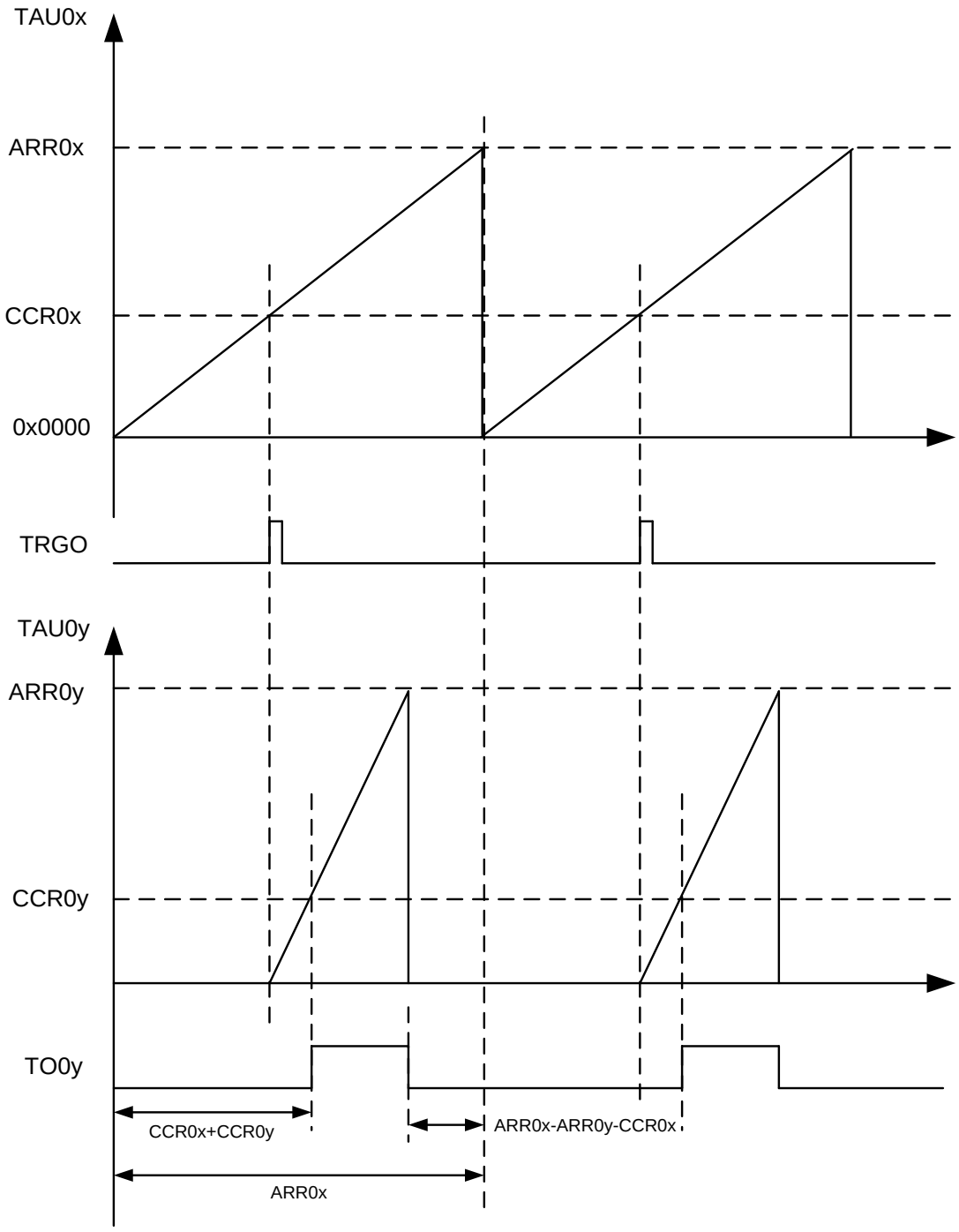


图 28-19 非对称级联 PWM 输出示意图

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。

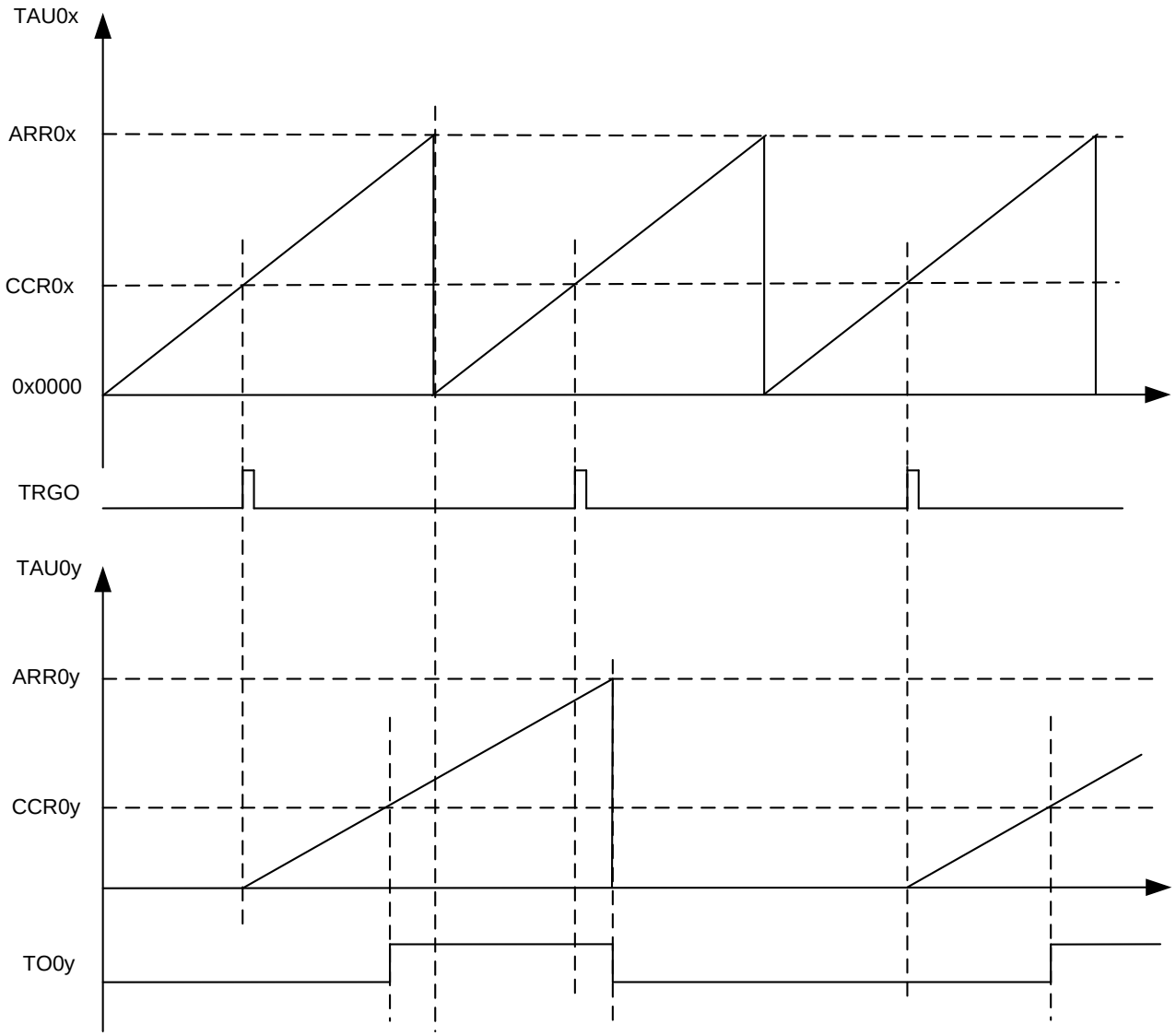


图 28-20 从机 ARR 大于主机的情况

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。

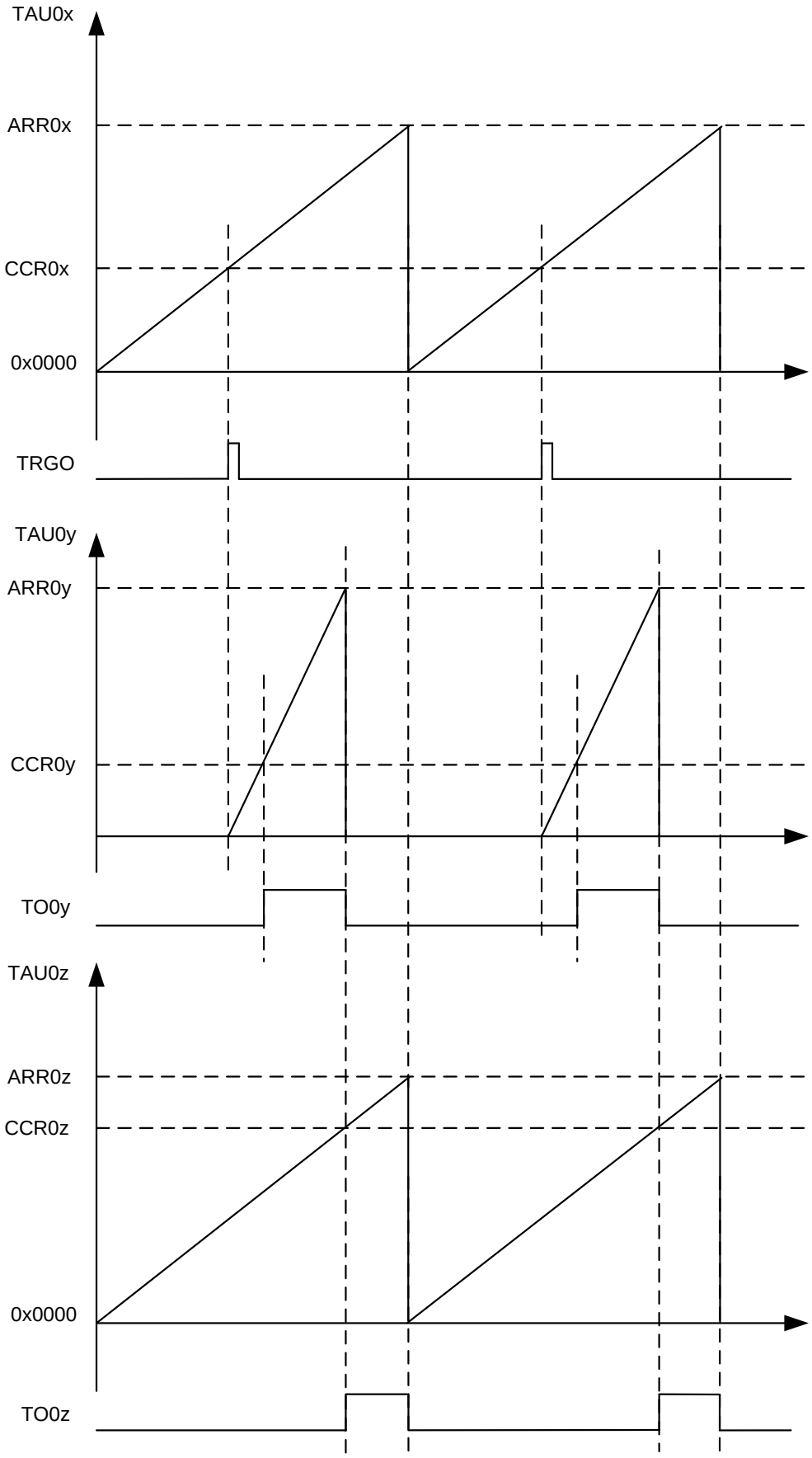

TAU0x
ARR0x
CCR0x
0x0000
TRGO
TAU0y
ARR0y
CCR0y
TO0y
TAU0z
ARR0z
CCR0z
0x0000
TO0z

图 28-21 非对称非交叠 PWM 输出示意图

28.3.11 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N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。数字滤波的长度由DFLEN寄存器定义，DFLEN=0表示关闭数字滤波，最大滤波长度是255次采样。

数字滤波示意图如下（DFLEN=3）：

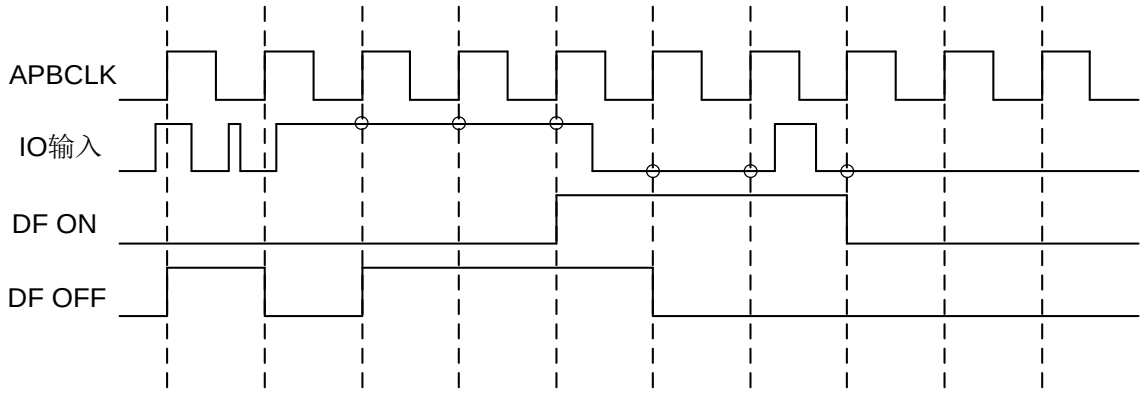


图 28-22TIxx 输入数字滤波示意图

28.4 寄存器

TAU模块起始地址：0x40013000

offset 地址	名称	符号
0x00	TAU0 控制寄存器	TAU0_CR
0x04		
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		

28.4.1 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	TEN[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	TEN	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 TEN 寄存器。TEN 置位后，定时器立即开始工作。*

28.4.2 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28.3.632位级联模式
4:2	CAPSEL	捕捉源选择寄存器

位号	位名	说明
		000：外部引脚输入（TIxx） 其他：参见28.3.7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）

28.4.3 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：Timer0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）

28.4.4 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。

28.4.5 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 捕捉模式下，当捕捉

28.4.6 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
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

28.4.7 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

位号	位名	说明
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

28.4.8 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

29 16 位基本定时器（BSTIM16）

29.1 概述

FM33LF0xxA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。

29.2 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

29.3 结构框图

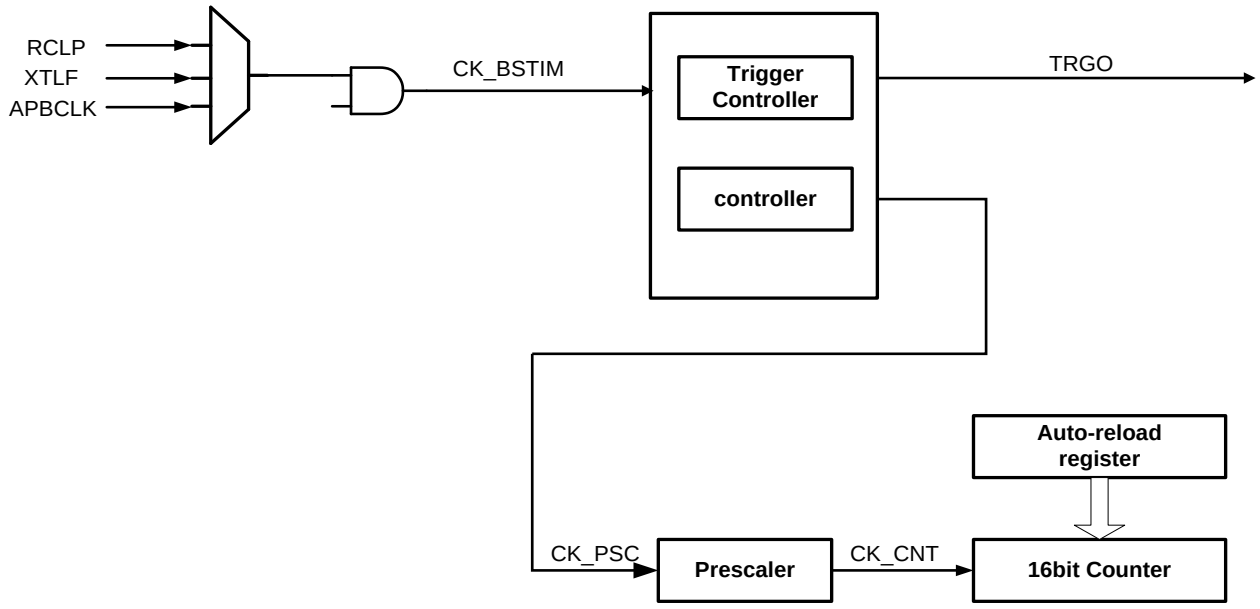


图 29-1 16 位基本定时器结构框图

29.4 功能描述

29.4.1 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。

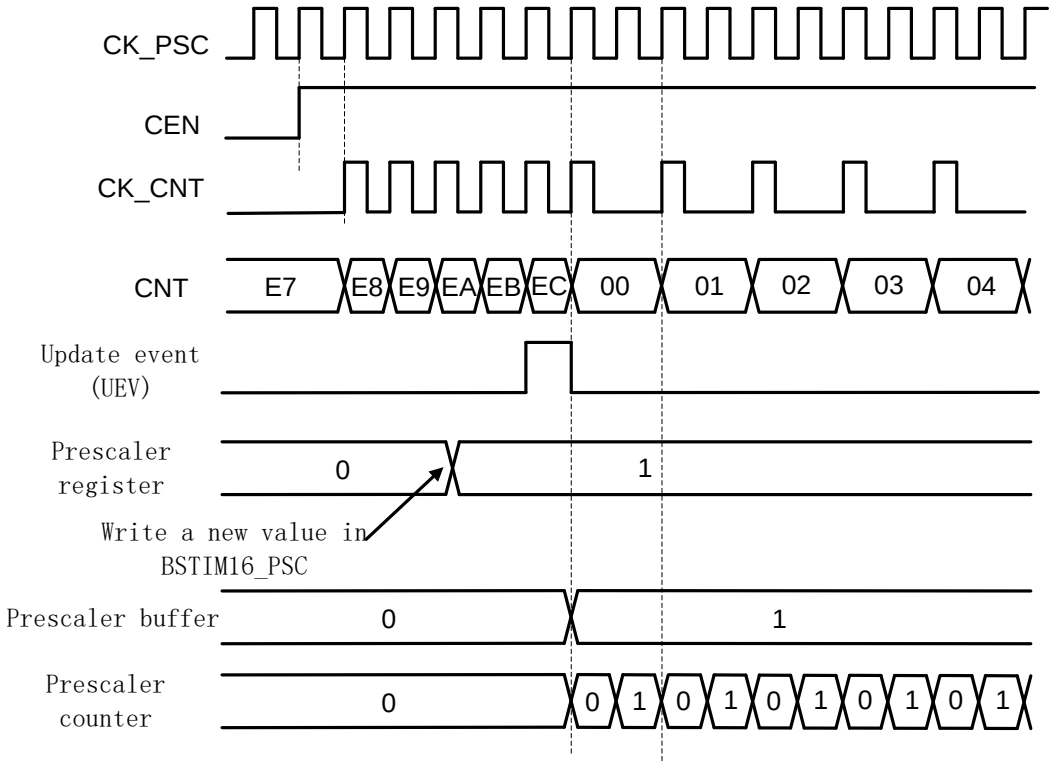


图 29-2 预分频从 1 变为 2 的波形

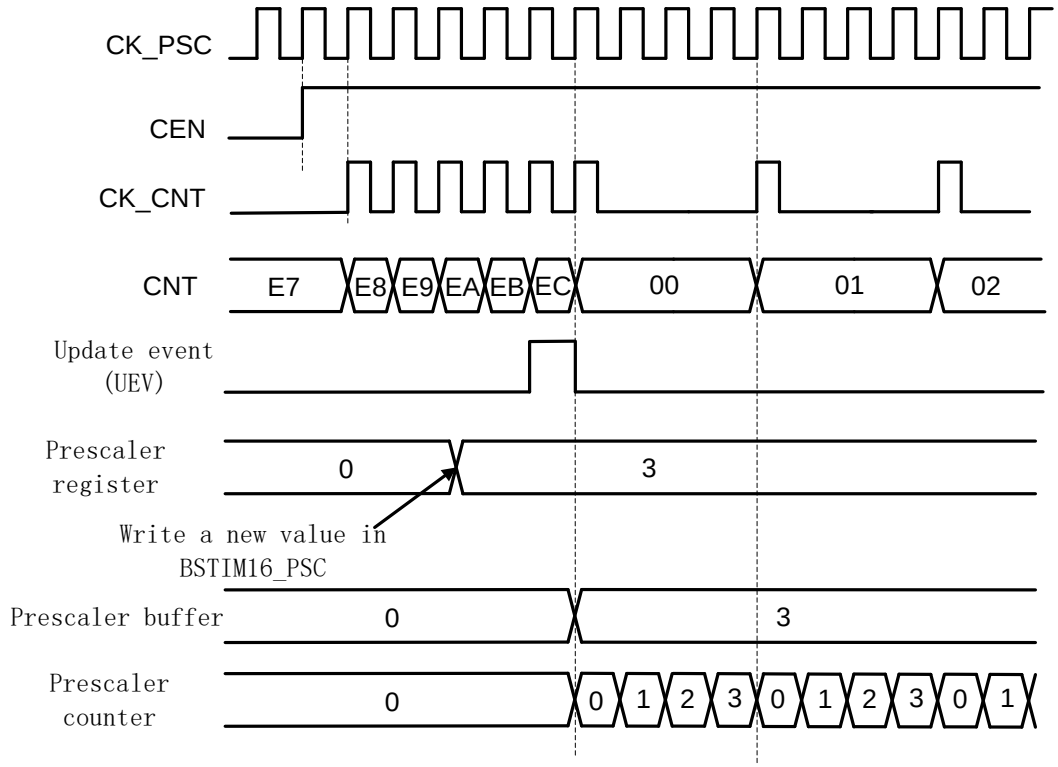


图 29-3 预分频从 1 变为 4 的波形

29.4.2 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和预分频计数器自动清零。设置UG寄存器不会触发UIF（Update Interrupt Flag）中断标志置位。

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

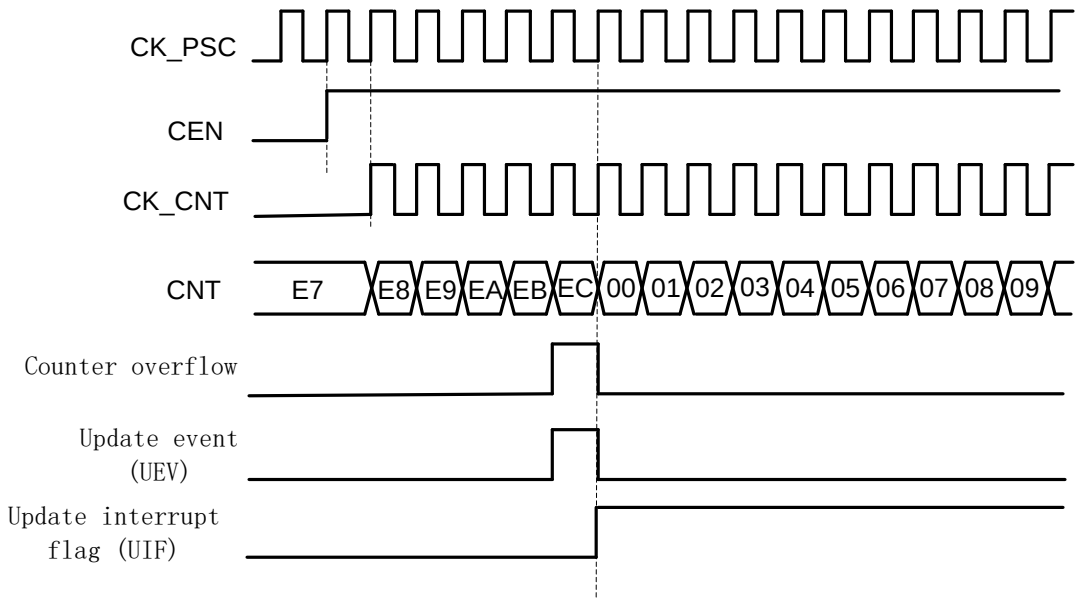


图 29-4 向上计数波形，内部时钟不分频

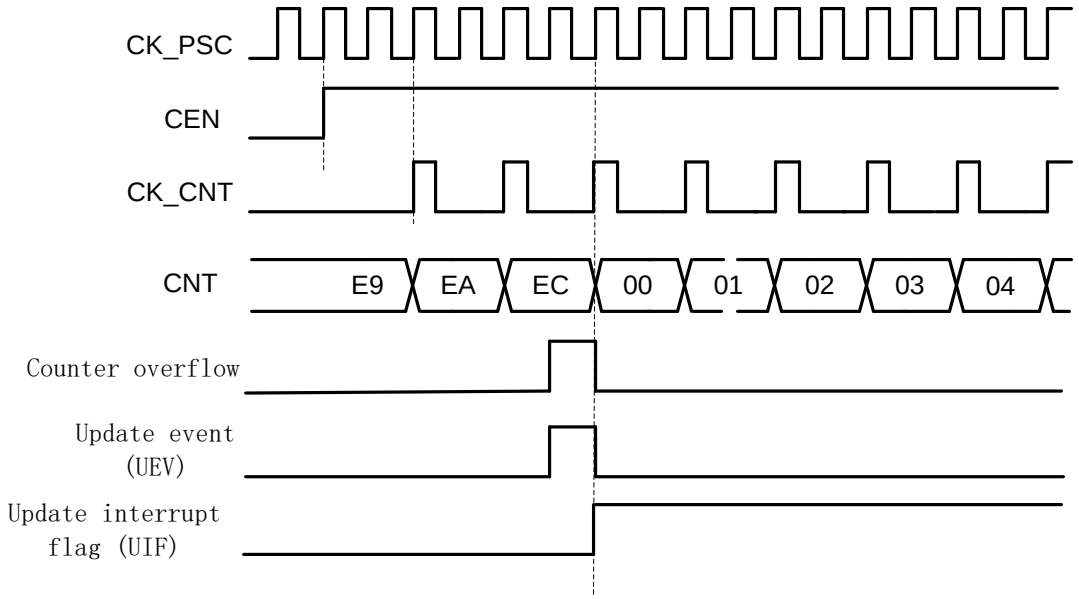


图 29-5 向上计数波形，内部时钟 2 分频

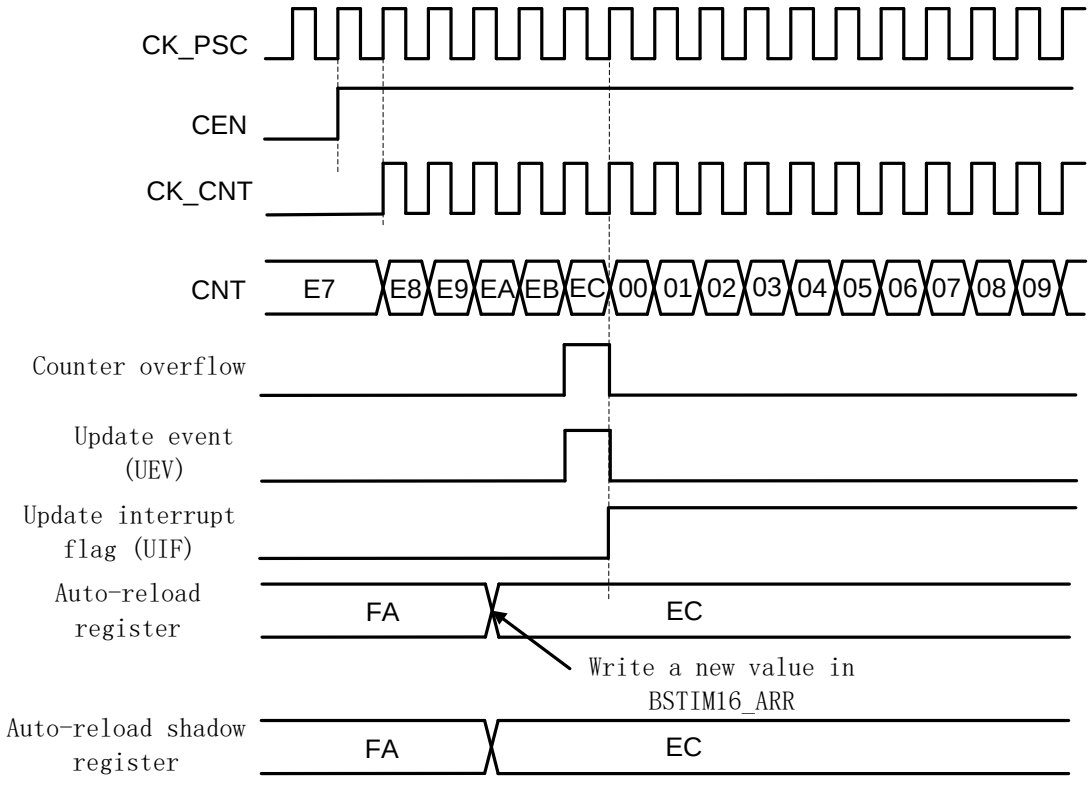


图 29-6 ARPE=0（ARR 没有预装载）时的更新事件

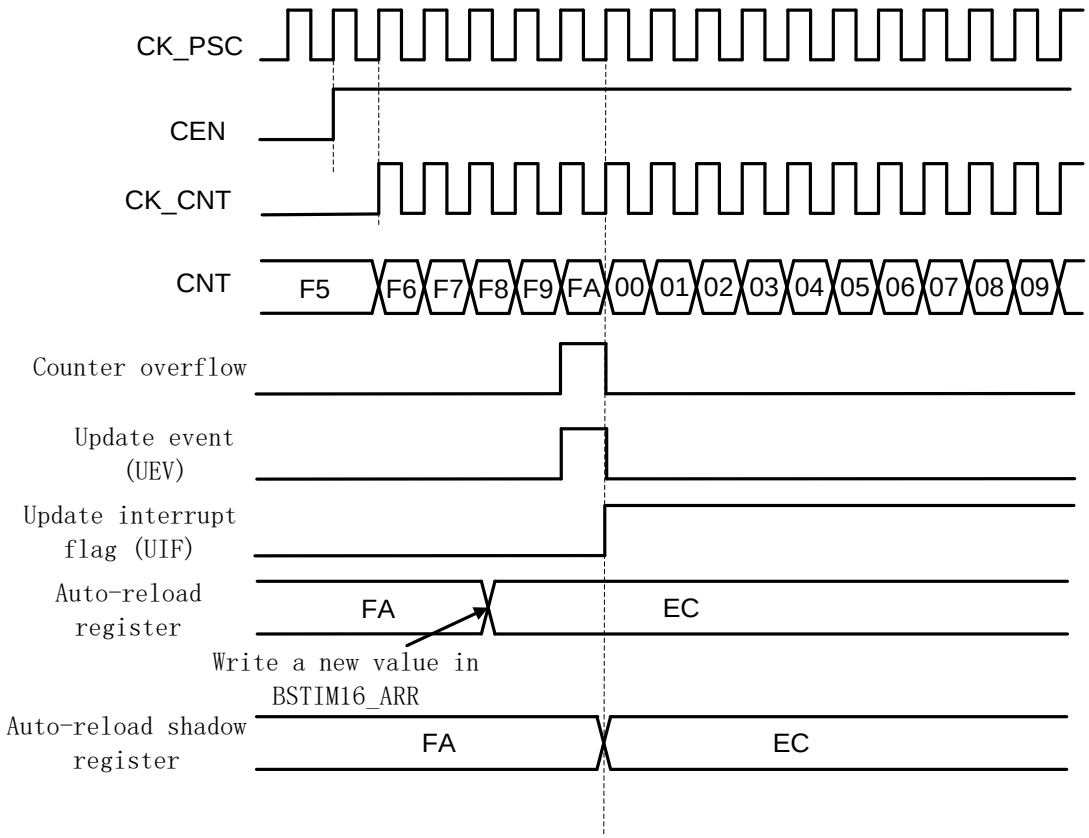


图 29-7 ARPE=1（ARR 预装载）时的更新事件

29.4.3 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。

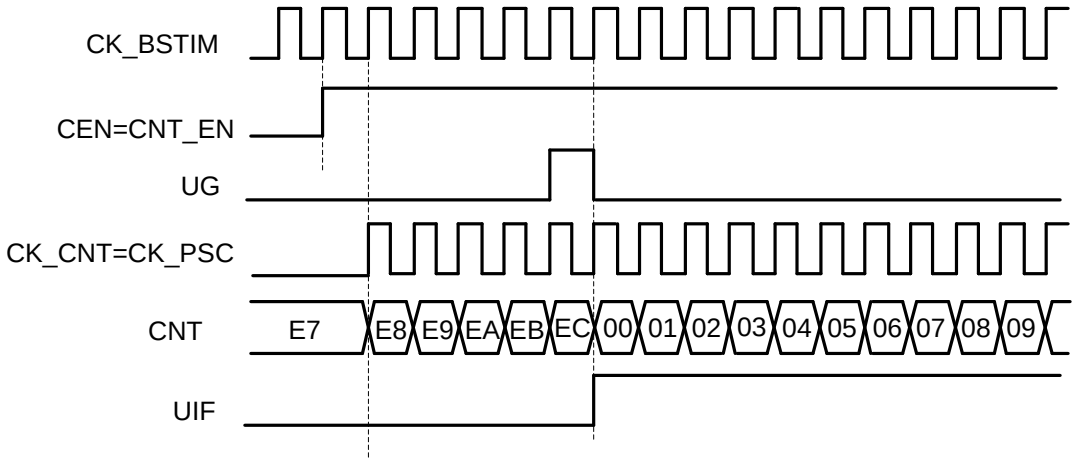


图 29-8 内部时钟源模式，时钟分频因子为 1

29.4.1 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DCU模块的DBG_TIMx_STOP寄存器定义。

29.5 寄存器

BSTIM16模块起始地址：0x4001_8C00

offset 地址	名称	符号
0x00000000	BSTIM16 控制寄存器 1 （BSTIM16 Control Register1）	BSTIM16_CR1
0x00000004	BSTIM16 控制寄存器 2 （BSTIM16 Control Register2）	BSTIM16_CR2
0x0000000C	BSTIM16 中断使能寄存器 （BSTIM16 Interrupt Enable Register）	BSTIM16_IER
0x00000010	BSTIM16 中断标志寄存器 （BSTIM16 Interrupt Status Register）	BSTIM16_ISR
0x00000014	BSTIM16 事件产生寄存器 （BSTIM16 Event Generation Register）	BSTIM16_EGR
0x00000024	BSTIM16 计数器寄存器 （BSTIM16 Counter Register）	BSTIM16_CNT
0x00000028	BSTIM16 预分频寄存器 （BSTIM16 Prescaler Register）	BSTIM16_PSC
0x0000002C	BSTIM16 自动重载寄存器 （BSTIM16 Auto-Reload Register）	BSTIM16_ARR

29.5.1 BSTIM16 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 - 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

29.5.2 BSTIM16 控制寄存器 2(BSTIM16_CR2)

名称	BSTIM16_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	-							
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 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 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

Bit	助记符	功能描述
3:0	-	RFU，未实现，读为 0

29.5.3 BSTIM16 中断使能寄存器(BSTIM16_DIER)

名称	BSTIM16_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	-							
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

29.5.4 BSTIM16 状态寄存器(BSTIM16_SR)

名称	BSTIM16_SR							
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

Bit	助记符	功能描述
		-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

29.5.5 BSTIM16 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，预分频计数器被清零。

29.5.6 BSTIM16 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

Bit	助记符	功能描述
15:0	**CNT**	计数器值

29.5.7 BSTIM16 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

29.5.8 BSTIM16 自动重载寄存器(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

Bit	助记符	功能描述
		入 shadow 寄存器

30 16 位低功耗定时器（LPTIM16）

30.1 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：XTLF、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

30.2 结构框图

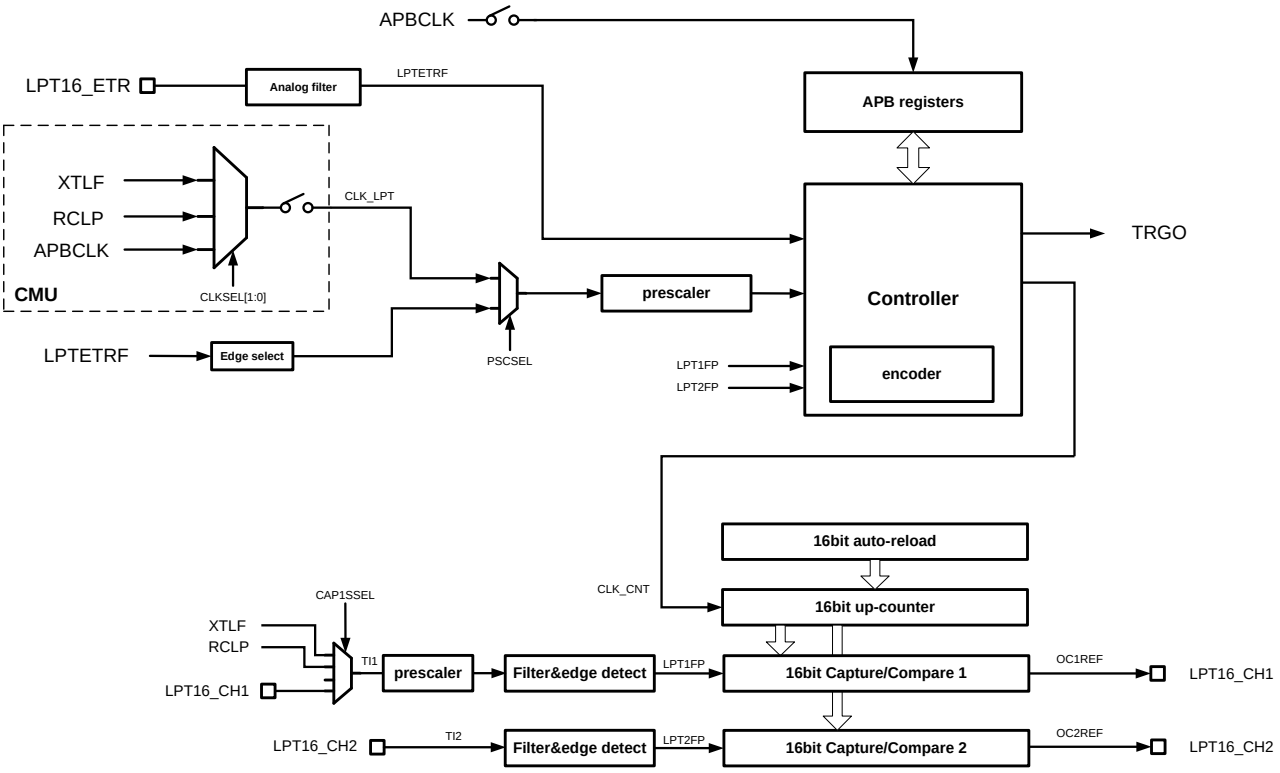


图 30-1 LPTIM16 结构框图

30.3 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11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中外设公式时钟配置寄存器，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模块外设复位寄存器。

30.4 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。

30.4.1 普通定时器

当LPTIM16_CFGR.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PCER.LPTCKS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LPTIM16_CR.EN。

30.4.2 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LPTIM16_ETR引脚输入的信号作为触发信号使用。LPTIM16_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LPTIM16_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LPTIM16_ETR的哪个边沿计数。

下图举例说明了LPTIM16_ETR触发计数，上升沿有效的情况。

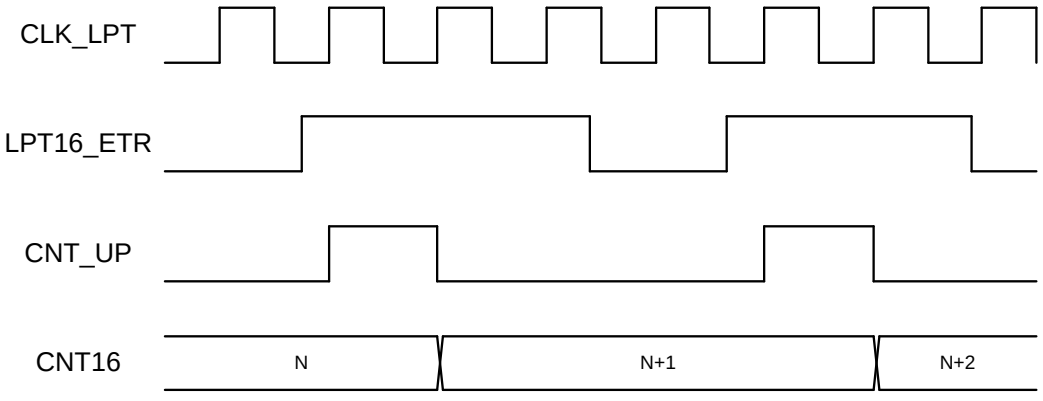


图 30-2 外部 ETR 脉冲上升沿触发计数

30.4.3 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LPTIM16_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LPTIM16_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。

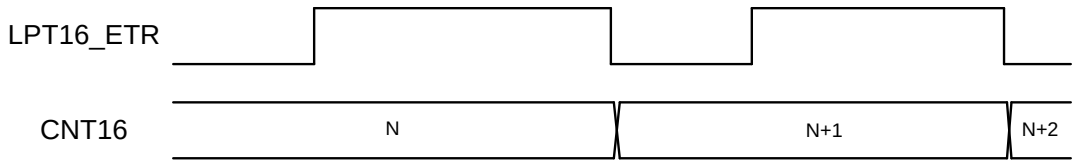


图 30-3 外部 ETR 脉冲异步计数（下降沿）

30.4.4 Timeout 模式

Timeout模式（LPTIM16_CFGR.TMODE=11）下，LPTIM16将LPTIM16_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LPTIM16_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LPTIM16_CR.EN结束计数过程。

LPTIM16_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LPTIM16_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LPTIM_CFGR.TRIGCFG寄存器设置LPTIM16对LPTIM16_ETR的哪个边沿计数。

下图是timeout模式下使用LPTIM16_ETR上升沿清零，并最终溢出的例子。

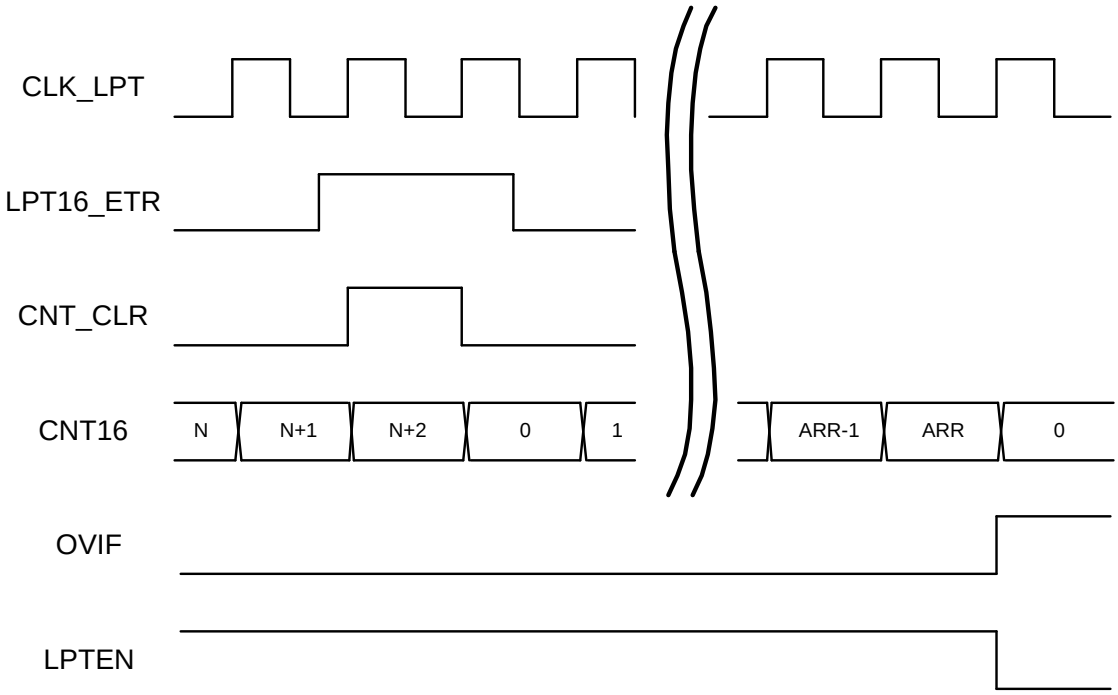


图 30-4TimeOut 模式

使用TimeOut模式，并使能LPTIM16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LPTIM16_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，LPTIM16超时溢出中断将唤醒芯片。

30.5 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。

30.5.1 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0x0000_0000开始计数，计数值等于比较值（CCRx）时输出置高，计数值等于目标值寄存器（ARR）时输出变低；PWM周期由ARR寄存器决定，占空比由CCRx寄存器决定。LPTIM16_CFGR.POLARITY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_CFGR.CCxS配置为10，此时LPTIM16_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，POLARITY=1的例子。

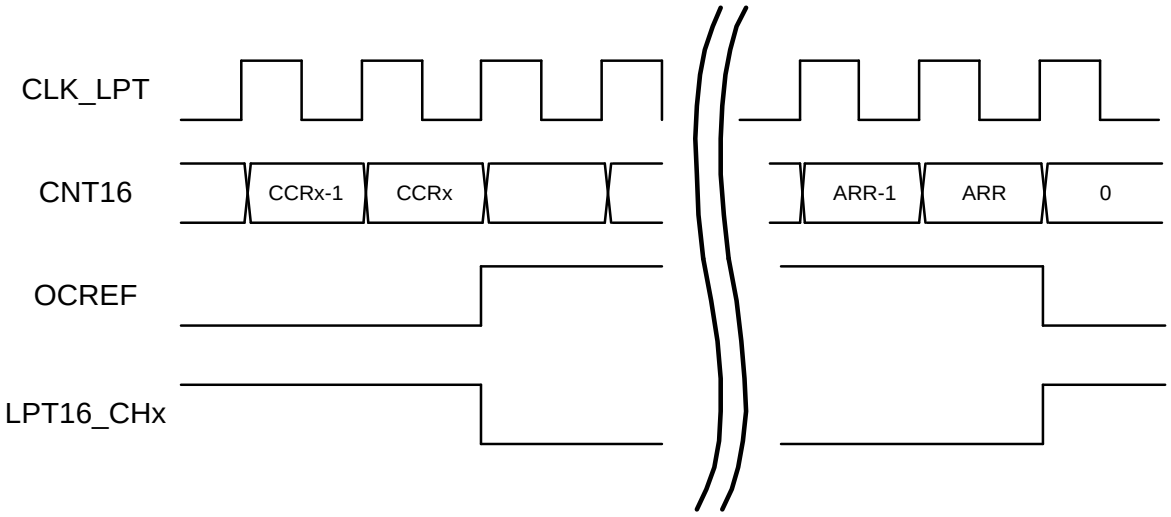


图 30-5PWM 输出

30.5.2 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。输入信号捕捉功能可以配合DMA使用，实现多次连续捕捉结果的自动搬运。

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。

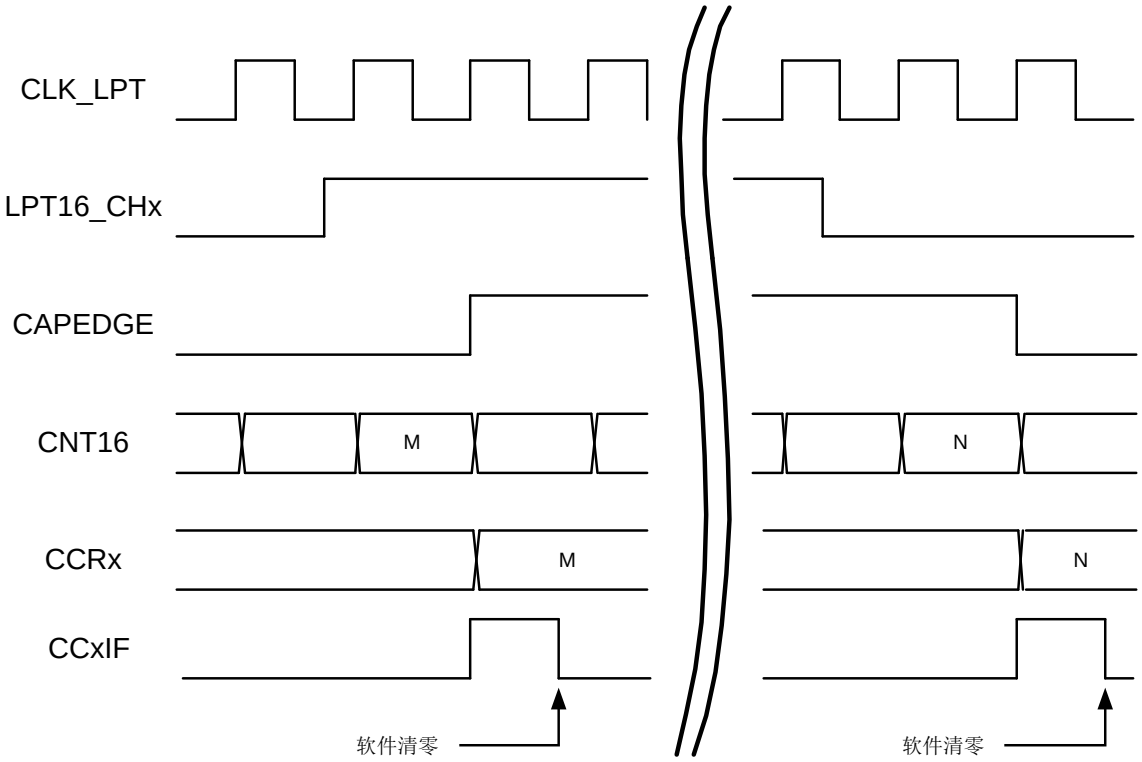


图 30-6 输入信号边沿捕捉

30.5.1 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。

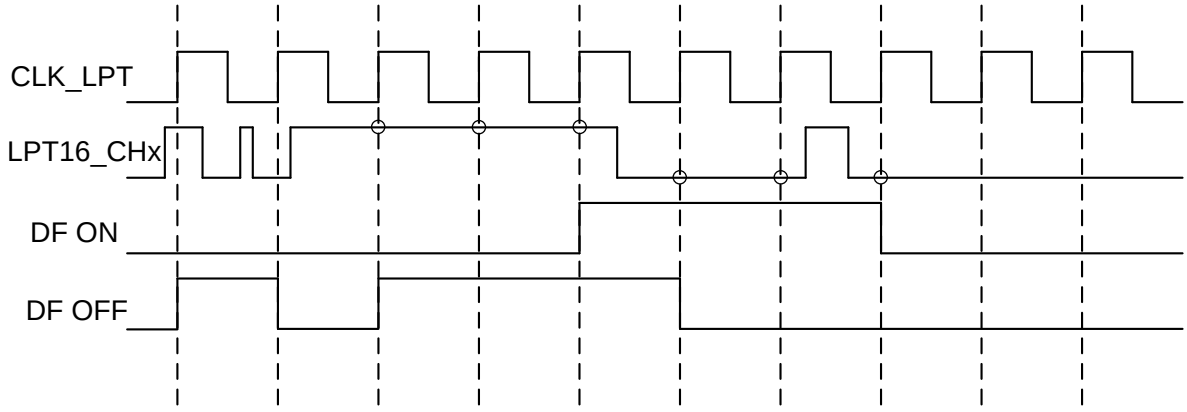


图 30-7 通道输入数字滤波

30.6 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 30-1 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。

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LPTIM16_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。

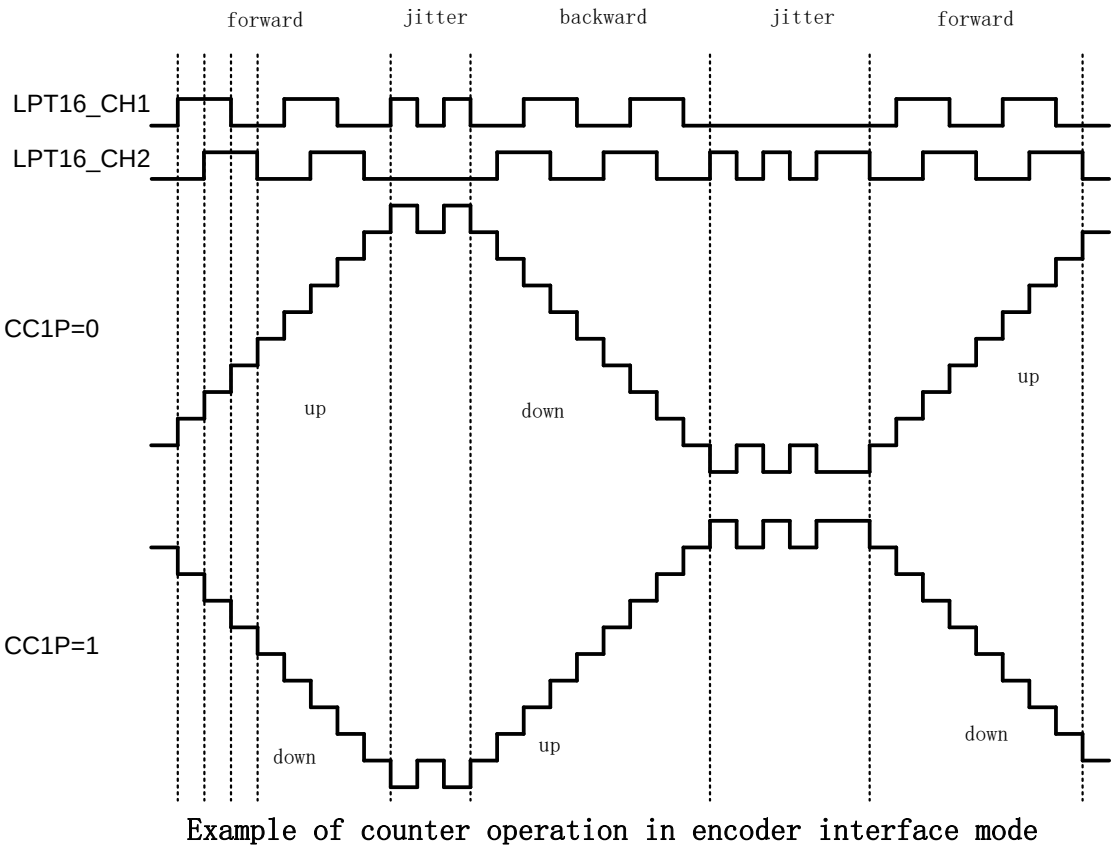


图 30-8 编码器模式下的计数器操作实例

正交编码信号使用LPTIM16_CHx输入，可以选择对输入信号是否进行数字滤波。注意LPTIM16_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LPTIM16_CH1，LPTIM16_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LPTIM16_CR.EN

30.7 寄存器

模块基地址：0x4001_8800

offset 地址	名称	符号
0x00	LPTIM16 配置寄存器 (LPTIM16 Config Register)	LPTIM16_CFGR
0x04	LPTIM16 计数寄存器 （LPTIM16 Counter Register）	LPTIM16_CNT
0x08	LPTIM16 捕捉比较控制和状态寄存器 （LPTIM16 Capture/Compare Control and Status Register）	LPTIM16_CCSR
0x0C	LPTIM16 目标值寄存器 （LPTIM16 Auto-Reload Register）	LPTIM16_ARR
0x10	LPTIM16 中断使能寄存器 （LPTIM16 Interrupt Enable Register）	LPTIM16_IER
0x14	LPTIM16 中断标志寄存器 （LPTIM16 Interrupt Status Register）	LPTIM16_ISR
0x18	LPTIM16 控制寄存器 （LPTIM16 Control Register）	LPTIM16_CR
0x20	LPTIM16 捕捉比较寄存器 1 （LPTIM16 Capture/Compare Register1）	LPTIM16_CCR1
0x24	LPTIM16 捕捉比较寄存器 2 （LPTIM16 Capture/Compare Register2）	LPTIM16_CCR2

30.7.1 LPTIM16 配置寄存器（LPTIM16_CFGR）

名称	LPTIM16_CFGR							
地址								
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	TMOD	
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: 外部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

30.7.2 LPTIM16 计数值寄存器（LPTIM16_CNT）

名称	LPTIM16_CNT							
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	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)

30.7.3 LPTIM16 捕捉比较控制和状态寄存器（LPTIM16_CCSR）

名称	LPTIM16_CCSR							
地址								
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：128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
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

位号	位名	说明
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）

30.7.4 LPTIM16 目标值寄存器（LPTIM16_ARR）

名称	LPTIM16_ARR							
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	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

30.7.5 LPTIM16 中断使能寄存器（LPTIM16_IER）

名称	LPTIM16_IER
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	-						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

30.7.6 LPTIM16 中断标志寄存器（LPTIM16_ISR）

名称	**LPTIM16_ISR**							
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			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

30.7.7 LPTIM16 控制寄存器（LPTIM16_CR）

名称	LPTIM16_CR
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	-							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

30.7.8 LPTIM16 捕捉比较寄存器 1（LPTIM16_CCR1）

名称	LPTIM16_CCR1							
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	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)

30.7.9 LPTIM16 捕捉比较寄存器 2（LPTIM16_CCR2）

名称	LPTIM16_CCR2							
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	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)

31 实时时钟（RTC）

31.1 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

31.2 结构框图

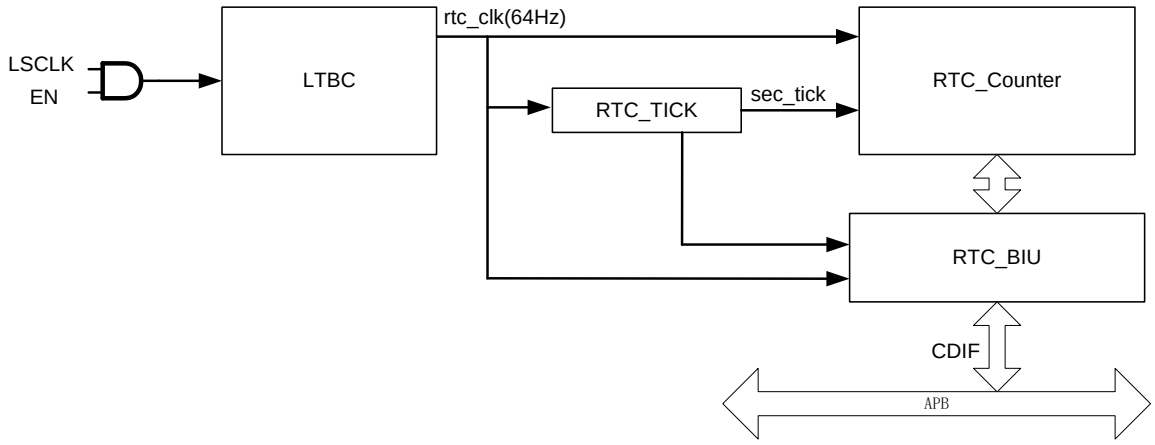


图 31-1RTC 结构框图

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。

31.3 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。

31.3.1 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 与 WDT 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+/-0.477ppm
- PLL 虚拟调校可得到精确秒时标
- 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

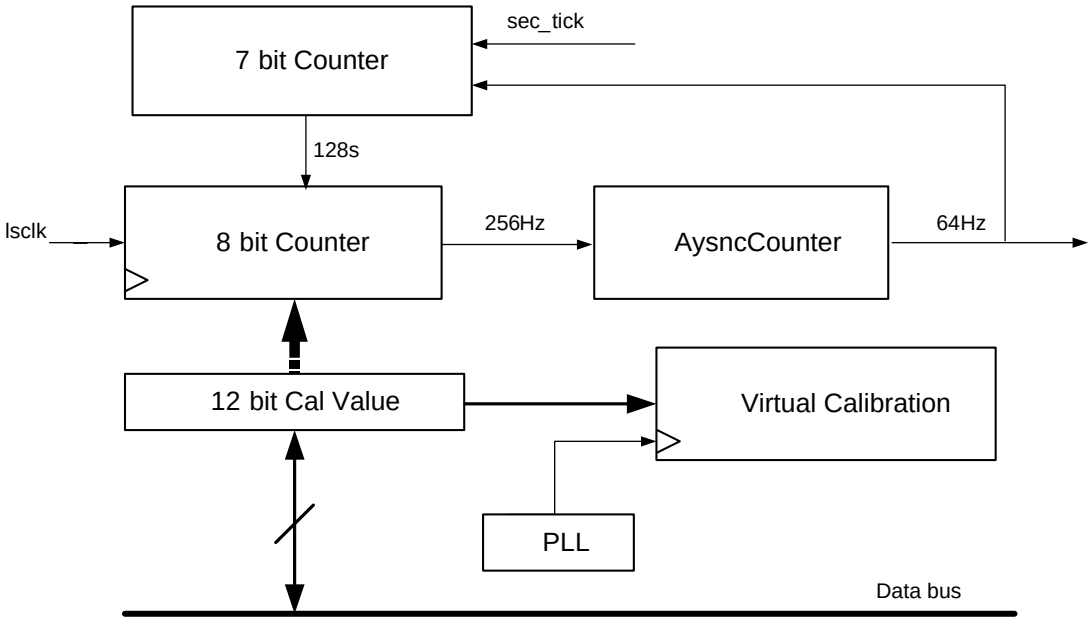


图 31-2LTBC 结构框图

31.3.2 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。FM33LF0xxA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 I	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

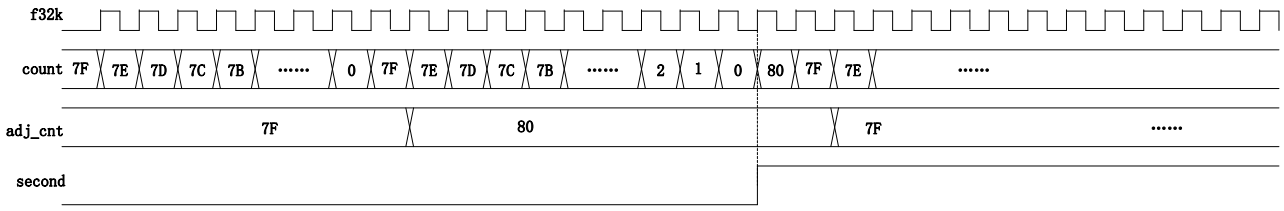
通过以上方式，可以得到平滑调整的秒周期，秒和秒之间最大只差30.5ppm，可以避免集中式调整

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。见下图：



31.3.3 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

31.3.4 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。

31.3.5 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 FM33LF0xxA 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，FM33LF0xxA 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。

31.3.6 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。

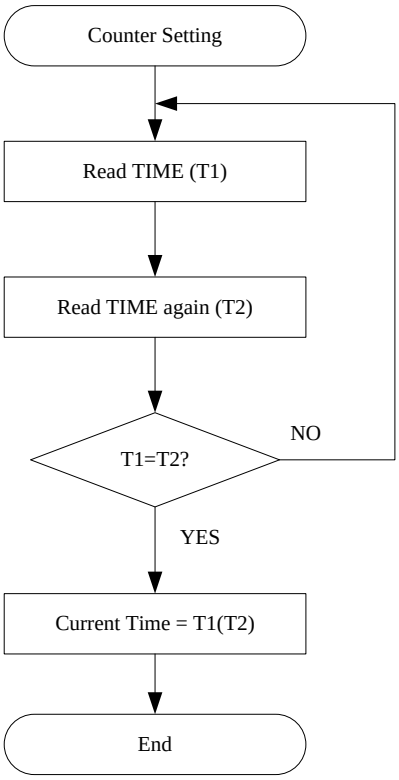


图 31-3RTC 时间读取流程图

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。

31.3.7 闰年判断

FM33LF0xxA 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。

31.4 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
0x00000034	LTBC 数值调整方向寄存器（RTC time Adjust Sign Register）	RTC_ADSIGN
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

31.4.1 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 RTCWE 置 1；当 CPU 向 RTCWE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 RTCWE 清 0。

31.4.2 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

位号	助记符	功能描述
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
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

31.4.3 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

位号	助记符	功能描述
		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 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

31.4.4 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)

31.4.5 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)

31.4.6 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)

31.4.7 BCD 时间天寄存器（RTC_BCDDAY）

名称	RTC_BCDDAY							
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	-		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)

31.4.8 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)

31.4.9 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)

31.4.10 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)

31.4.11 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)

31.4.12 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

31.4.13 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 1：表示减少计数初值

位号	助记符	功能描述
8:0	ADJUST	LTBC 补偿调整数值 (Time Adjust)

31.4.14 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)

31.4.15 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_A（内部走时时钟关闭） 1：使能 RTC_A

32 段码液晶驱动（LCD）

32.1 概述

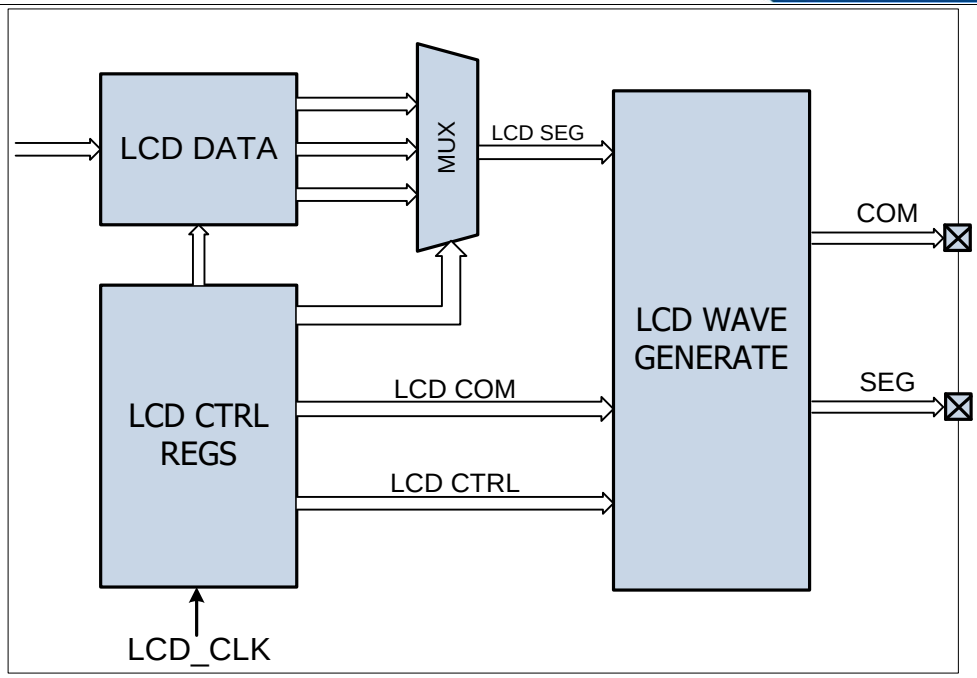
LCD 显示驱动模块用于驱动段码式液晶屏，能够支持 1、2、3、4、6、8COM，最大显示段数分别为 32 段（1COM）、64 段（2COM）、96 段（3COM）、128 段（4COM）、180 段（6COM）和 224 段（8COM）。

主要特点：

- 最大支持 8×28、6×30、4×32、3×32、2×32、1×32 的显示段数
- 1/3bias、1/4bias
- 16 级灰度可调
- LCD 驱动支持片内电阻型
- 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

32.2 结构框图

LCD 显示驱动模块结构框图如图 32-1 所示：



片内 **BUFFER** 驱动模式

图 32-1 LCD 显示控制模块结构框图

32.3 IO 配置

LCD驱动电路工作时最多会占用36个GPIO，在使用LCD前，需要将用到的引脚设置为模拟功能（PxyFCR=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。

32.4 功能说明

32.4.1 工作时钟和显示帧频率

LCD驱动电路使用LSCLK工作，其典型频率在32KHz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 32-1 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 32-2 典型帧频率和 DF 的关系

32.4.2 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。

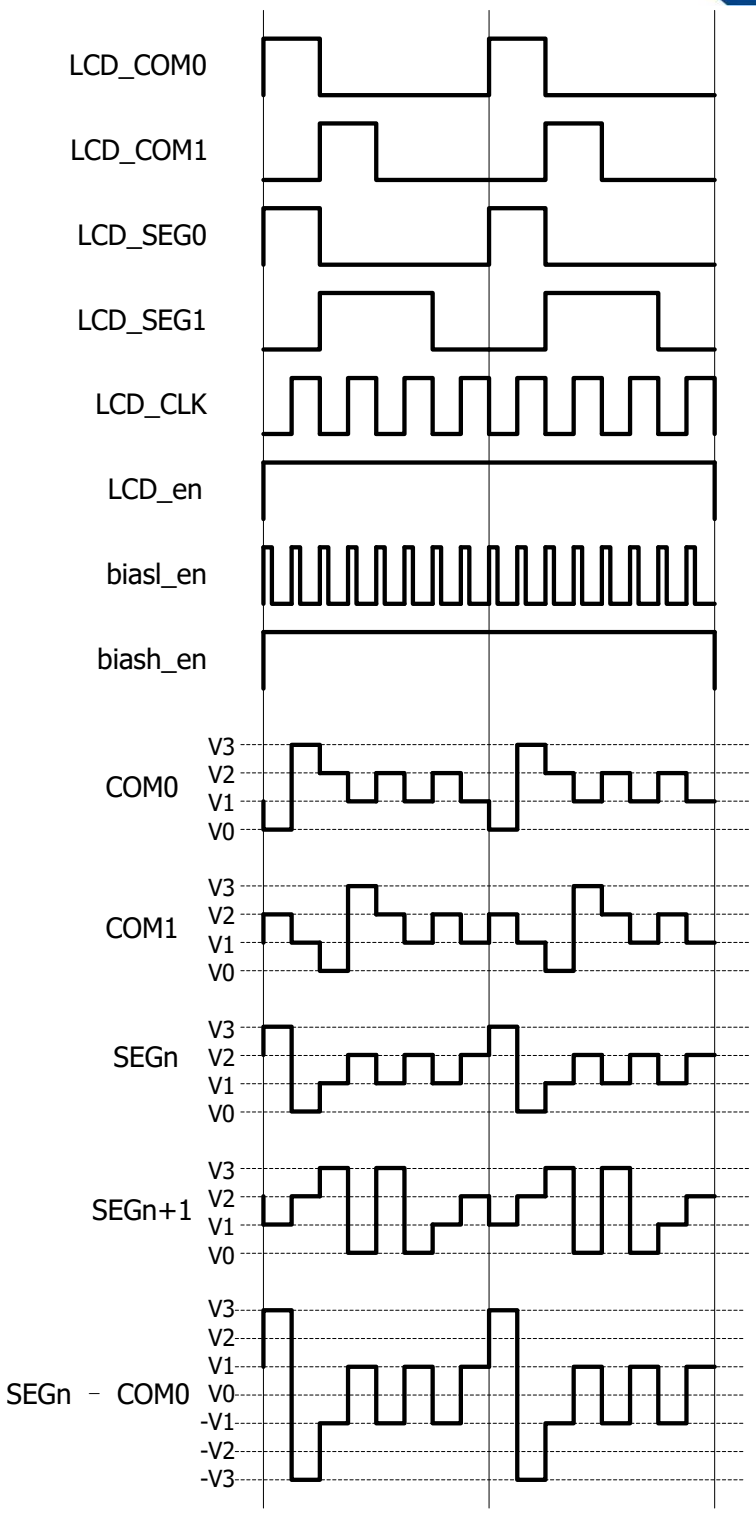


图 32-2LCD 驱动波形(1/4 duty, 1/3 bias, type A)

32.4.3 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。

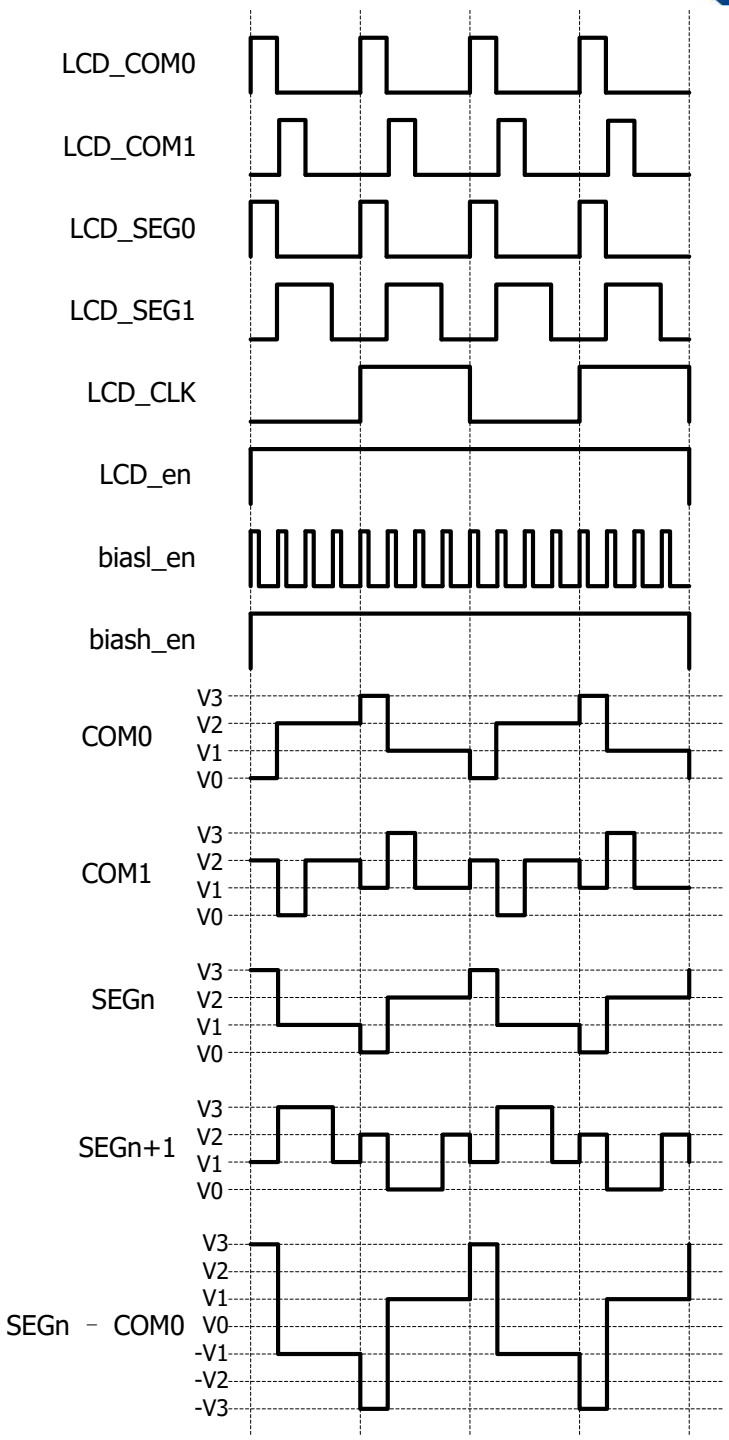


图 32-3 LCD 驱动波形(1/4 duty, 1/3 bias, type B)

32.4.4 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：

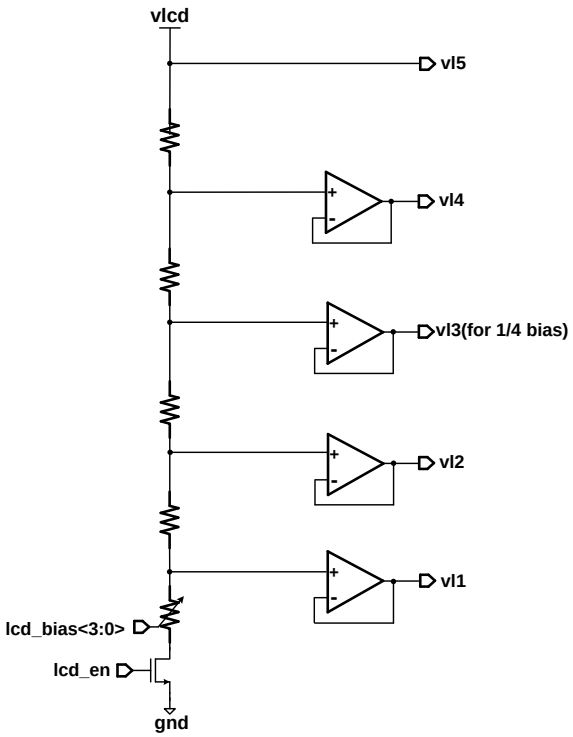


图 32-4 LCD 片内电阻 buffer 型驱动电路

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。

32.4.5 显示闪烁功能

软件可以设置显示控制寄存器 DISPCTRL 中的 FLICKER 位为 1，来使能显示闪烁。FLICKER 使能后，根据 TON 和 TOFF 寄存器值确定闪烁频率。在使能 FLICKER 功能之前应先设置 TON/TOFF 并设置 MD 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，FLICKER 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。

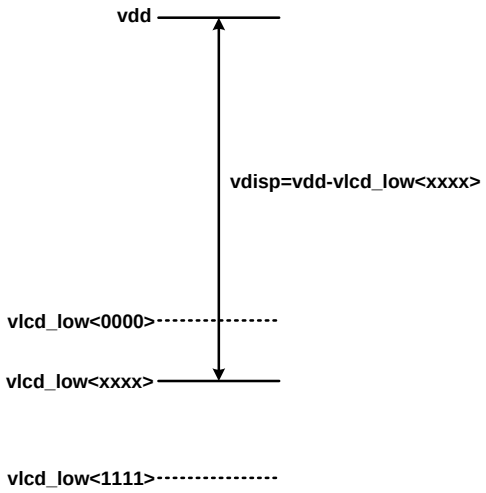
32.4.6 偏置电压调整

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 30.6.12 中的表格进行配置。

32.5 寄存器

模块起始地址：0x4001_0C00

offset 地址	名称	符号
0x00000000	显示控制寄存器（LCD Control Register）	LCD_CR
0x00000004	显示测试控制寄存器（LCD test Register）	LCD_TEST
0x00000008	显示频率控制寄存器（LCD Frequency Control Register）	LCD_FCR
0x0000000C	闪烁时间寄存器（LCD Flick Time Register）	LCD_FLKT
0x00000014	显示中断使能寄存器（LCD Interrupt Enable Register）	LCD_IER
0x00000018	显示中断标志寄存器（LCD Interrupt Status Register）	LCD_ISR
0x00000024	显示数据缓存寄存器 0（LCD data buffer registers 0）	LCD_DATA0
0x00000028	显示数据缓存寄存器 1（LCD data buffer registers 1）	LCD_DATA1
0x0000002C	显示数据缓存寄存器 2（LCD data buffer registers 2）	LCD_DATA2
0x00000030	显示数据缓存寄存器 3（LCD data buffer registers 3）	LCD_DATA3
0x00000034	显示数据缓存寄存器 4（LCD data buffer registers 4）	LCD_DATA4
0x00000038	显示数据缓存寄存器 5（LCD data buffer registers 5）	LCD_DATA5
0x0000003C	显示数据缓存寄存器 6（LCD data buffer registers 6）	LCD_DATA6
0x00000040	显示数据缓存寄存器 7（LCD data buffer registers 7）	LCD_DATA7
0x00000050	COM 使能控制寄存器（LCD COM Enable Register）	LCD_COMEN
0x00000054	SEG 使能控制寄存器 0（LCD SEG Enable Register0）	LCD_SEGEN0

32.5.1 显示控制寄存器（LCD_CR）

名称	LCD_CR							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			LMUX_2	-	BIASL_CFG		
位权限	U-0			R/W-0	U-0	R/W-000		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名		SCFSEL			-		IC_CTRL	

位权限		R/W-000			U-0		R/W-01	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ENMODE	FLICK	-		LCDBIAS			
位权限	R/W-0	R/W-0	U-0		R/W-1110			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BIASMD	ANTIPOLAR	WFT	LMUX		LCDEN
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

Bit	助记符	功能描述
31:29	--	RFU：未实现，读为 0
28	**LMUX_2**	COM 数量选择，与本寄存器 bit1，bit2 组合为 LMUX[2:0]后选择 000：4COM 001：6COM 010：8COM 011：8COM 100：1COM 101：2COM 110：3COM 111：8COM
27	--	RFU：未实现，读为 0
26:24	**BIASL_CFG**	BIASL输出配置 000：BIASL 始终输出 0 001：BIASL 输出持续 1+1 个 LSCLK 010：BIASL 输出持续 1+2 个 LSCLK 011：BIASL 输出持续 1+4 个 LSCLK 100：BIASL 输出持续 1+8 个 LSCLK 101：BIASL 始终输出 1
22:20	**SCFSEL**	电容驱动频率选择 000 = 频率为帧频*COM 数 001 =频率为 LSCLK/8 010 =频率为 LSCLK/16 011 =频率为 LSCLK/32 100 =频率为 LSCLK/64 101 =频率为 LSCLK/128 110 =频率为 LSCLK/256 111=频率为 LSCLK /512 注意：当选择 110 或 111 档位时，如果频率低于帧频*2*COM，则输出结果与 000 档位相同；当帧频率设置较高时，使用 000 档位可能导致驱动脉冲频率太高无法正常输出。
19:18	**--**	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 1 = 片内电阻型驱动
14	**FLICK**	显示闪烁使能位

Bit	助记符	功能描述
		1：显示闪烁，闪烁频率由 TON 和 TOFF 寄存器设置 0：关闭闪烁
13:12	--	RFU：未实现，读为 0
11:8	**LCDBIAS**	LCD 偏置电平选择位，用于显示灰度控制
7:6	--	RFU：未实现，读为 0
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，与本寄存器 bit28 组合为 LMUX[2:0]后选择 000：4COM 001：6COM 010：8COM 011：8COM 100：1COM 101：2COM 110：3COM 111：8COM
0	**LCDEN**	LCD 显示使能位 1：启动 LCD 显示 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

32.5.2 显示测试控制寄存器（LCD_TEST）

名称	LCD_TEST							
Offset	0x00000004							
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
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

32.5.3 测试模式下引脚输出数据寄存器

这组寄存器只在测试模式下有效，与相关显示寄存器共享存储空间。

名称	LCD 测试模式下引脚输出数据寄存器 TDISPDATA							
地址								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TDISPDATA0	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
TDISPDATA1	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
TDISPDATA2	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
TDISPDATA3	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
TDISPDATA4								
TDISPDATA5								

名称	LCD 测试模式下引脚输出数据寄存器 TDISPDATA							
地址								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TDISPDATA6								
TDISPDATA7	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
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

32.5.4 显示频率控制寄存器（LCD_FCR）

名称	LCD_FCR							
Offset	0x00000008							
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(Display Frequency)

32.5.5 闪烁时间寄存器（LCD_FLKT）

名称	LCD_FLKT							
offset	0x0000000C							
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

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

32.5.6 显示中断使能寄存器（LCD_IER）

名称	LCD_IER							
Offset	0x00000014							
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

32.5.7 显示中断标志寄存器（LCD_ISR）

名称	LCD_ISR							
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	---	---	---	---	---	---	DONF	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

32.5.8 显示数据寄存器（LCD_DATAx）

LCD 显示模块内有 8 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=0,1,2,3,4,5,6,7)**							
Offset	0x00000024 + x*0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DSDA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DSDA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DSDA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DSDA[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:0	DSDA	LCD 显示数据 (Display Data)

32.5.8.1 4COM 显示数据寄存器

名称	**4com 显示数据寄存器**							
Offset	0x00000024 ~ 0x00000038							
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

名称	4com 显示数据寄存器							
	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
	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
	-	-	-	-	-	-	-	-
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	-	-	-	-	-	-	-	-
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	-	-	-	-	-	-	-	-
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	-	-	-	-	-	-	-	-
DISPDATA5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	-	-	-	-	-	-	-	-
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	-	-	-	-	-	-	-	-
	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
	-	-	-	-	-	-	-	-

32.5.8.2 6COM 显示数据寄存器

名称	6com 显示数据寄存器							
Offset	0x00000024 ~ 0x00000040							
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
	-	-	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
	-	-	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
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	-	-	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

名称	6com 显示数据寄存器							
	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
	-	-	SEG29 COM3	SEG28 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
	-	-	SEG29 COM4	SEG28 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
	-	-	SEG29 COM5	SEG28 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
DISPDATA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	-	-	-	-	-	-	-	-
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	-	-	-	-	-	-	-	-
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	-	-	-	-	-	-	-	-
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	-	-	-	-	-	-	-	-
DISPDATA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	-	-	-	-	-	-	-	-
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	-	-	-	-	-	-	-	-
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	-	-	-	-	-	-	-	-
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	-	-	-	-	-	-	-	-

32.5.8.3 8COM 显示数据寄存器

名称	8com 显示数据寄存器							
Offset	0x00000024 ~ 0x00000040							
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
	-	-	-	-	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
	-	-	-	-	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
	-	-	-	-	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
	-	-	-	-	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
DISPDATA4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	8com 显示数据寄存器							
	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
	-	-	-	-	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
	-	-	-	-	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
	-	-	-	-	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
	-	-	-	-	SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7

32.5.9 LCD COM 使能控制寄存器（LCD_COMEN）

名称	LCD_COMEN

Offset	0x00000050							
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 输出使能控制 1: COM 输出使能 0: COM 输出禁止

32.5.10 LCD SEG 使能控制寄存器 0（LCD_SEGEN0）

名称	LCD_SEGEN0							
Offset	0x00000054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SEG31_ COM7_E N	SEG30_ COM6_E N	SEG29_ COM5_E N	SEG28_ COM4_E N	SEGEN[27:24]			
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SEGEN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SEGEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SEGEN[7:0]							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	SEG31_COM7_EN	LCD SEG 和 COM 输出使能控制 (SEG31 or COM7 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
30	SEG30_COM6_EN	LCD SEG 和 COM 输出使能控制 (SEG30 or COM6 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
29	SEG29_COM5_EN	LCD SEG 和 COM 输出使能控制 (SEG29 or COM5 enable) 1：SEG 或 COM 输出使能

位号	助记符	功能描述
		0：SEG 或 COM 输出禁止
28	SEG28_COM4_EN	LCD SEG 和 COM 输出使能控制 (SEG28 or COM4 enable) 1：SEG 或 COM 输出使能 0：SEG 或 COM 输出禁止
27:0	SEGEN	LCD SEG 输出使能控制 (SEG Enable) 每个 bit 对应一个特定的 SEG 1：SEG 输出使能 0：SEG 输出禁止

33 LED 驱动（LED）

33.1 概述

LED 显示驱动模块支持最大 10COM*16DIG 共 160 点的 LED 驱动扫描，同时每个显示点支持 8bitPWMdimming 调光功能。

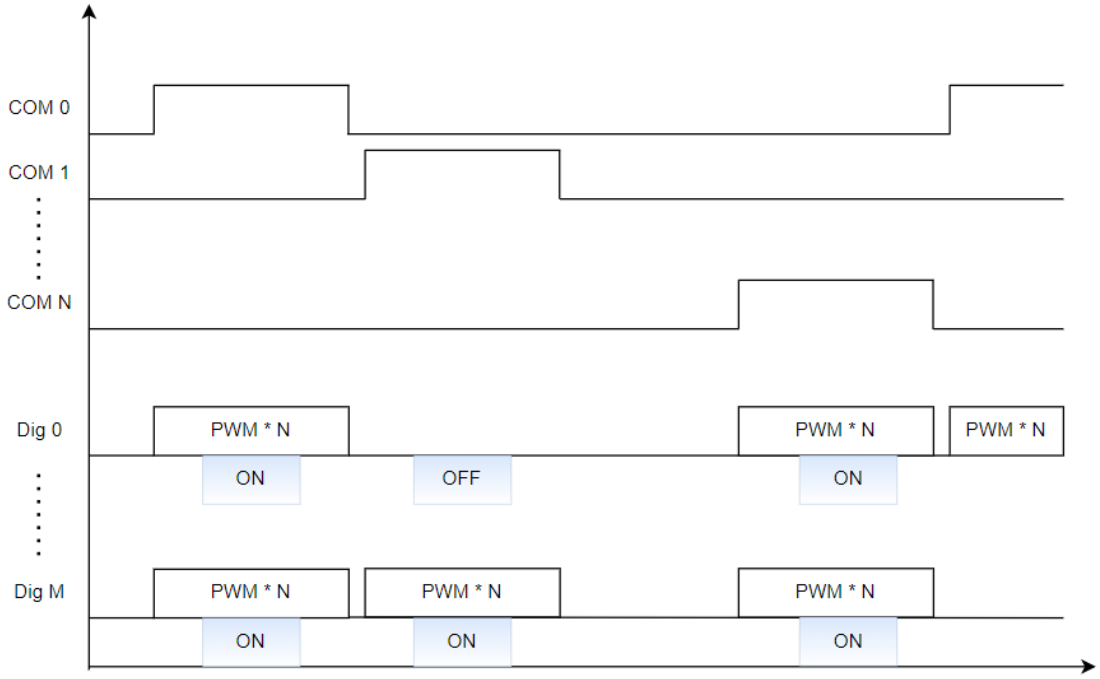
33.2 IO 配置

LED使用GPIO的数字输出功能，最大占用26个GPIO

33.3 功能说明

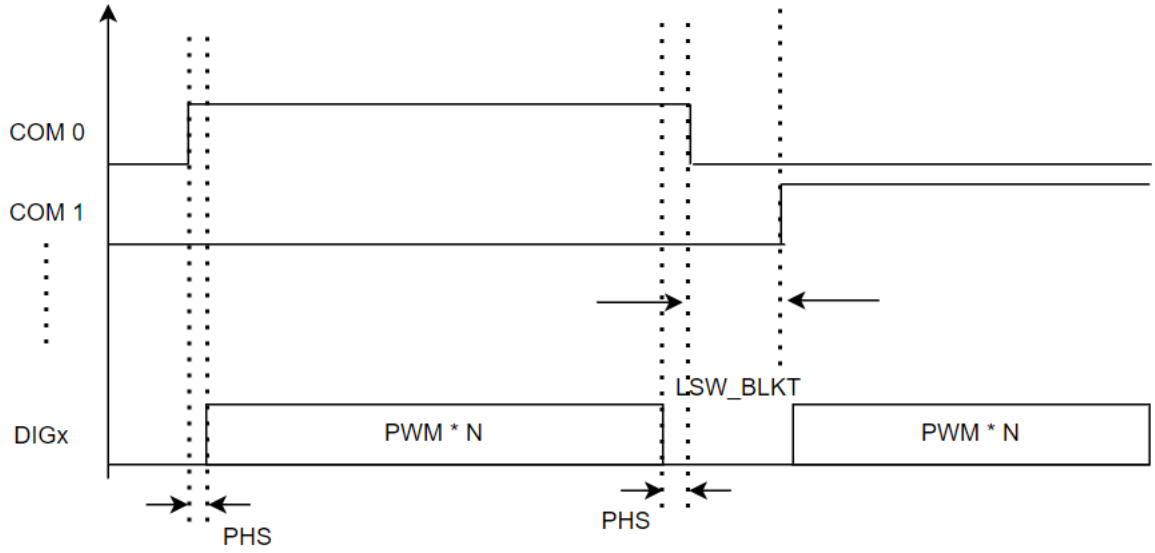
33.3.1 扫描波形

LED扫描波形如下图所示。COMx为行扫描信号，DIGx为列扫描信号。当COMx和DIGx同时有效时，对应的LED点亮。本模块最大支持10个行扫描，通过寄存器配置最大行数，扫描总是从COM0开始连续进行，不支持跳行扫描。下图中ON表示点亮的LEDdot，OFF表示熄灭的LEDdot。



每个行使能之间，有一个死区时间，这个时间最短是8*SCAN_CLK，最长是32*SCAN_CLK，在SCAN_CLK为8MHz的情况下，为1~32us，由LSW_BLKT寄存器定义。COM使能，到DIG输出（PWM

或者静态）之间，还有一个相移时间，由PHS寄存器定义，单位是SCAN_CLK，即最小相移步长125ns。

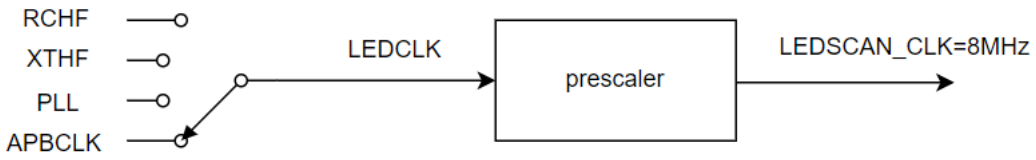


综上，每行显示时间：$T_{line} = N*T_{PWM} + 2*T_{PHS} + T_{blank}$

总的帧扫描时间：$T_{Frame} = T_{line} * Line_count$

33.3.2 工作时钟

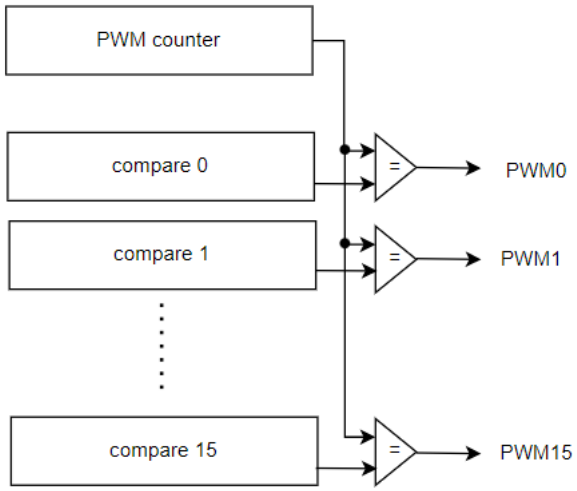
LED模块有寄存器总线时钟和扫描时钟两个时钟。其中总线时钟为APBCLK，扫描时钟为CMU输出时钟的分频，时钟源可以是RCHF、XTHF、PLL等。LED扫描以固定的8MHz时钟工作，使用前必须通过预分频寄存器得到8MHz的工作时钟。



33.3.3 PWM 调光

本模块支持数字调光（PWMdimming）功能，当PWM使能后，DIG输出波形不再是固定高电平，而是PWM波形，PWM计数器长度为8bit，占空比调整的最小步长为125ns（8MHz周期），PWM频率为8MHz/256=31.25KHz

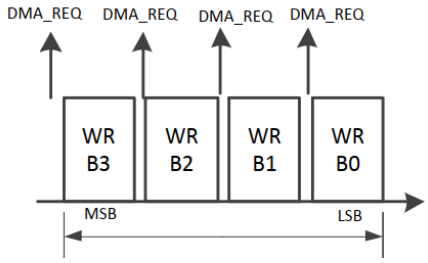
LED驱动需要最大16路PWM波形输出，使用同一个8bit计数器作为定时时基，16个8bit比较寄存器作为占空比比较基准。



PWM比较值的更新可以由软件执行，或者由DMA执行；在一个COM扫描完成后，会产生LSIF中断，可以选择软件跟新下一条COM对应的16个LED 点的PWM值，或者使用DMA来更新PWM值，完成16个8bit比较寄存器的自动更新，这个更新需要若干个APBCLK完成，为了不影响显示效果，DMA搬运必须在line switch blank time内完成：

- 如果blank time结束DMA更新仍未完成，则下一个行扫描的启动将被delay，直到DMA更新完成
- 如果DMA更新在blank time结束前完成，则下一行扫描在blank time结束后立即开始

为了保证显示效果，软件应将LED的DMA通道设置为最高优先级。



33.3.4 扫描时序

每个COM的行点亮时间由COM_LENGTH寄存器定义，为PWM频率的整数倍。

扫描的最大行数，由MAX_LINE寄存器定义。

则LED扫描周期频率定义为：

$F_{SCAN} = F_{PWM} / (COM_LENGTH * MAX_LINE)$

假设扫描10行，每行点亮32个PWM周期，则：

F_{SCAN} = 31.25KHz / (32*10) = 97Hz

如果取COM_LENGTH=26，则可以得到扫描帧频率为120Hz

33.3.5 点数据

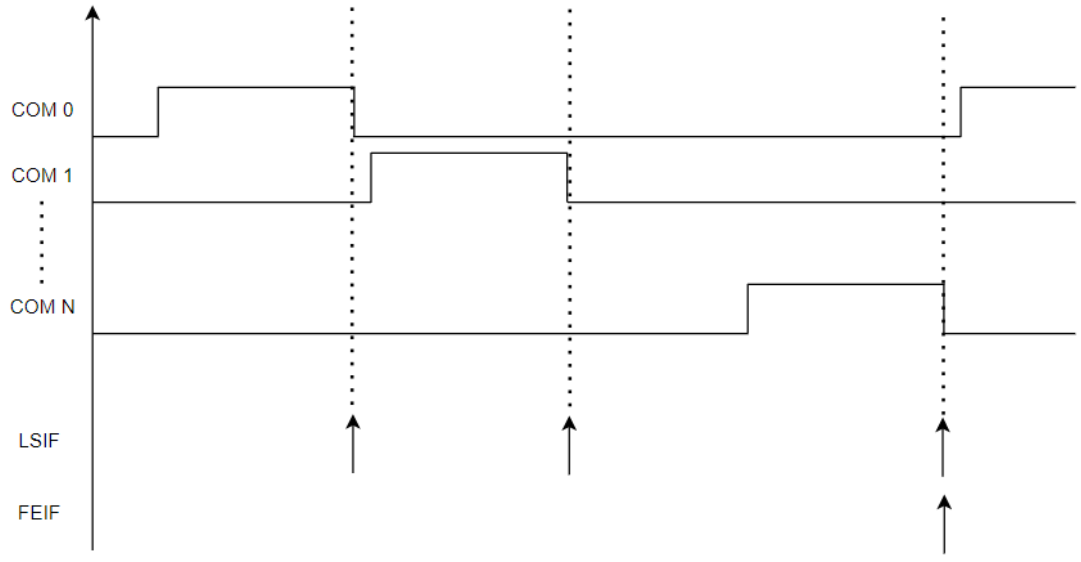
LED最大支持160点，160bit的点数据寄存器用于保存当前COM每个单点的显示是ON还是OFF。没一条COM值对应与16个DIG点，在所有COM扫描完成（一帧）后，FEIF中断标志寄存器置位，此时软件可以更新下一帧的dot数据，也可以通过使能DMA来自动完成帧数据的更新。为便于软件处理，点数据寄存器按照每行对应16bit来组织。

33.3.6 输出电平

通过寄存器可以配置扫描时输出的有效电平，默认配置下，COM和DIG输出电平为高有效，用户可以分别设置COM和DIG的输出有效电平为高或低，并且可以配置COM在空闲时为无效电平或者浮空态。

33.3.7 中断

LED模块可以在COM切换和帧扫描结束时产生中断。



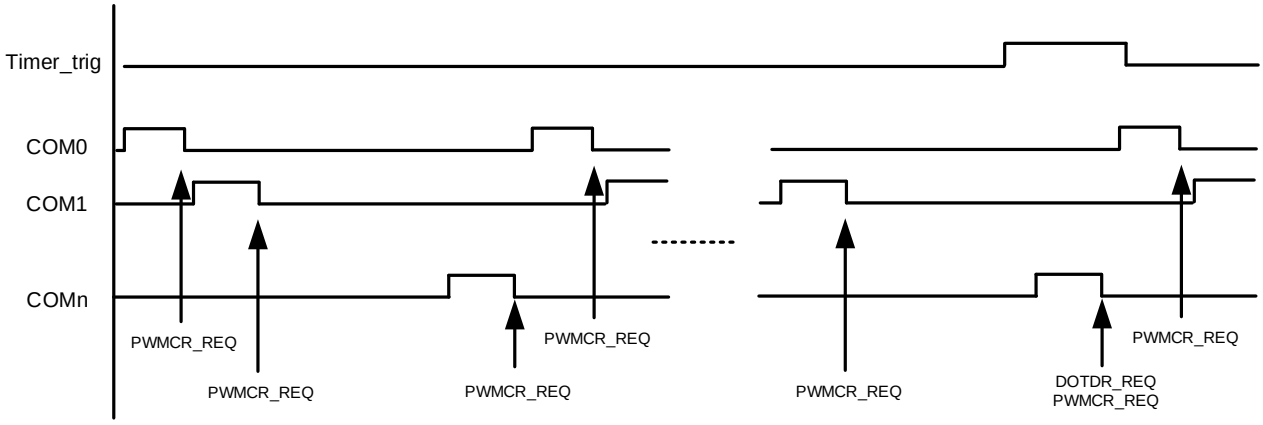
33.3.8 DMA 支持

LED模块可发出两种DMA的请求，如果都是用需分别使用不同的通道处理，一个是COM线切换时触发的PWM比较值的更新，一个是帧结束后触发的DOT数据更新。

LED模块可以在DMA的帮助下，完全自主地更新每个点的PWM值以及每个点的ON OFF值，可以实现图案自动显示功能。PWM以及DOT值对应的DMA功能均可以开关。

如下如所示DMA的请求产生的点，每次请求产生，会根据DMACR的配置数量搬运数据。所以实际RAM中需要按照显示图案需求进行组织。并维护PWM数据表以及DOT图案表。

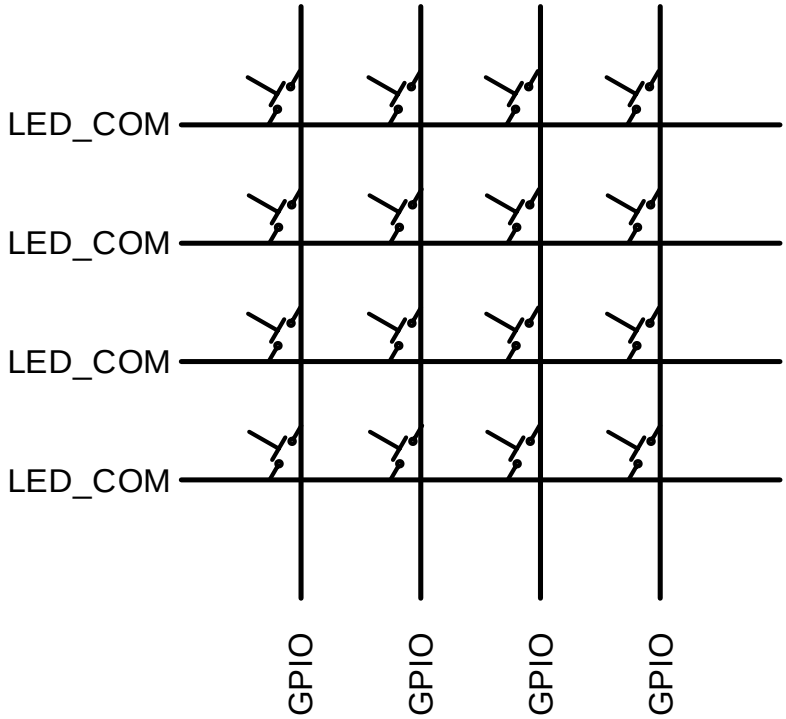
DOT DMA功能增加了外部trig同步的功能，如果使能外部trig同步功能后，将在外部trig来到后再触发请求所有dot数据的更新，更新完之后需再次等待trig同步，应对与需要按照特定时间进行数据刷新的场景。



33.3.9 按键扫描

LED驱动电路还兼具按键扫描功能。

以 4x4 键盘扫描为例，连接关系参见下图。



LED 按照 COM 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 LED_CCOM 寄存器查询当前 COM 脚的状态，为高电平的 COM 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。

33.4 寄存器

LED模块基地址：0x4001AC00

offset 地址	名称	符号
0x00	LED 控制寄存器	CR
0x04	LED 配置寄存器	CFGR
0x0C	LED 中断使能寄存器	IER
0x10	LED 中断状态寄存器	ISR
0x20	LED PWM 比较值 A	PWMCRA
0x24	LED PWM 比较值 B	PWMCRB
0x28	LED PWM 比较值 C	PWMCRC
0x2C	LED PWM 比较值 D	PWMCRD
0x40	LED 点数据 L0	DOT0DR
0x44	LED 点数据 L1	DOT1DR
0x48	LED 点数据 L2	DOT2DR
0x4C	LED 点数据 L3	DOT3DR

0x50	LED 点数据 L4	DOT4DR
0x54	LED 点数据 L5	DOT5DR
0x58	LED 点数据 L6	DOT6DR
0x5C	LED 点数据 L7	DOT7DR
0x60	LED 点数据 L8	DOT8DR
0x64	LED 点数据 L9	DOT9DR
0x80	DMA 控制寄存器	DMACR
0x84	PWMCRx 的 DMA 访问寄存器	DPCAR
0x88	DOTxDR 的 DMA 访问寄存器	DDRAR
0x8C	LED 扫描当前 COM 线	CCOM

33.4.1 LED 控制寄存器

名称	LED_CR							
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								
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名								
位权限								
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				DDMAEN	PDMAEN	PWMEN	LEDEN
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
30:4	--	RFU：未实现，读为 0
3	**DDMAEN**	Dot DMA使能位 0：关闭DotDMA功能 1：使能DotDMA功能
2	**PDMAEN**	PWM DMA使能位，仅在PWMEN=1的情况下有效 0：关闭PWMDMA功能 1：使能PWMDMA功能
1	**PWMEN**	PWMdimming 使能位 0：关闭 PWM 功能，静态驱动 1：使能 PWM 功能，占空比驱动
0	**LEDEN**	LED 显示使能位 1：启动 LED 显示 0：关闭 LED 显示

33.4.2 LED 配置寄存器

名称	LED_CFGR							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	PHS							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LSW_BLKT					COM_FLT	DIG_POLAR	COM_POLAR
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	COM_LENGTH							
位权限	R/W-0001 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SCAN_PRES				MAX_LINE			
位权限	R/W-0000				R/W-1010			

位号	位名	说明
31:24	**PHS**	DIG相对COM的相移，单位T_{SCAN_CLK} 相移范围0~255个T_{SCAN_CLK}
23:19	**LSW_BLKT**	Line switch blank time = (LSW_BLKT +1) * 8 * T_{SCAN_CLK} 通常 SCAN_CLK=8MHz，此时： 00000: 1us 00001: 2us … 11111: 32us
18	**COM_FLT**	COM 在空闲时的状态 0：空闲时输出无效电平 1：空闲时浮空
17	**DIG_POLAR**	DIG 有效电平极性 0：高有效 1：低有效
16	**COM_POLAR**	COM 有效电平极性 0：高有效 1：低有效
15:8	**COM_LENGTH**	行驱动时间，单位是 PWM 周期 0x00: 1 个 PWM 周期 0x01: 2 个 PWM 周期 … 0xFF: 256 个 PWM 周期
7:4	**SCAN_PRES**	扫描时钟预分频系数 F_{SCAN}=LED_CLK/(SCAN_PRES+1)
3:0	**MAX_LINE**	最大扫描行数 0000：RFU 0001：COM0 0010：COM0~1 …… 1010：COM0~9 1011：RFU 1100：RFU 1101：RFU 1110：RFU 1111：RFU

33.4.3 LED 中断使能寄存器

名称	LED_IER							
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	-						FEIE	LSIE
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:2	---	**未实现，读为0**
1	**FEIE**	Frame end 中断使能
0	**LSIE**	Line switch 中断使能

33.4.4 LED 中断标志寄存器

名称	LED_ISR							
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							FEIF	LSIF
位权限							R/W-0	R/W-0

位号	位名	说明
31:2	---	**未实现，读为0**
1	**FEIF**	Frame end 中断标志，硬件置位，软件写 1 清零
0	**LSIF**	Line switch 中断标志，硬件置位，软件写 1 清零

33.4.5 PWM 比较值寄存器 A

名称	LED_PWMCRA							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CR3							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CR2							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CR1							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CR0							
位权限	R/W-1111 1111							

位号	位名	说明
31:24	**CR3**	DIG3 PWM比较值寄存器
23:16	**CR2**	DIG2 PWM比较值寄存器
15:8	**CR1**	DIG1 PWM比较值寄存器
7:0	**CR0**	DIG0 PWM比较值寄存器 0x00: duty cycle = 0/256 0x01: duty cycle = 1/256 … 0xFF: duty cycle = 255/256

33.4.6 PWM 比较值寄存器 B

名称	LED_PWMCRB							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CR7							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CR6							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CR5							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CR4							
位权限	R/W-1111 1111							

位号	位名	说明
31:24	**CR7**	DIG7 PWM比较值寄存器
23:16	**CR6**	DIG6 PWM比较值寄存器
15:8	**CR5**	DIG5 PWM比较值寄存器

位号	位名	说明
7:0	**CR4**	DIG4PWM比较值寄存器 0x00: duty cycle = 0/256 0x01: duty cycle = 1/256 … 0xFF: duty cycle = 255/256

33.4.7 PWM 比较值寄存器 C

名称	LED_PWMCRC							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CR11							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CR10							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CR9							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CR8							
位权限	R/W-1111 1111							

位号	位名	说明
31:24	**CR11**	DIG11 PWM比较值寄存器
23:16	**CR10**	DIG10 PWM比较值寄存器
15:8	**CR9**	DIG9 PWM比较值寄存器
7:0	**CR8**	DIG8 PWM比较值寄存器 0x00: duty cycle = 0/256 0x01: duty cycle = 1/256 … 0xFF: duty cycle = 255/256

33.4.8 PWM 比较值寄存器 D

名称	LED_PWMCRD							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CR15							
位权限	R/W-1111 1111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CR14							
位权限	R/W-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CR13							
位权限	R/W-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	CR12
位权限	R/W-1111 1111

位号	位名	说明
31:24	**CR15**	DIG15 PWM比较值寄存器
23:16	**CR14**	DIG14 PWM比较值寄存器
15:8	**CR13**	DIG13 PWM比较值寄存器
7:0	**CR12**	DIG12 PWM比较值寄存器 0x00: duty cycle = 0/256 0x01: duty cycle = 1/256 … 0xFF: duty cycle = 255/256

33.4.9 LED 点数据寄存器 x (x=0~9)

名称	LED_DOTx							
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	Dot[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Dot[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:0	**---**	**未实现，读为0**
15:0	**dot**	COMx当前对应的DIG点数据，bit0对应DIG0，bit1对应DIG1…

33.4.10 DMA 控制寄存器

名称	LED_DMACR							
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	-							DDTRIG
位权限	U-0							RW-0

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DOTDR_LEN				-	PWMCR_LEN		
位权限	RW-1010				U-0	RW-100		

位号	位名	说明
31:8	---	**未实现，读为0**
8	DDTRIG	DOTDR 的 DMA 请求将于外部 trig 同步。外部 trig 可以是 timer 定时发出。 1：与外部 trig 同步后的一次 FRAME 扫描完成产生 DOT DMA 请求，下一次需要再次等待 trig。 0：每一次 FRAME 扫描完成后都产生 DOTDMA 请求
7:4	DOTDR_LEN	一次 FRAME（所有使能的 com 线扫描完毕）扫描后，如果使能了 DDMAEN，DMA 会搬入的 DOTDR 数据个数，每个 DOTDR 数据对应与特定的 COM 线，默认为 10 个，如果没有使用所有的 COM，则可以根据实际情况搬入数据。 范围：1~10
2:0	PWMCR_LEN	一次 COM 线扫描完毕后，如果使能了 PDMAEN,DMA 会搬入 PWMCR 寄存器的个数，最大 4 个对应了 PWMCRA/B/C/D.可根据使用到的最大 DIG 数增减。 范围：1~4

33.4.11 PWMCRx 的 DMA 访问寄存器

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

位号	位名	说明
31:0	DPCAR	DMA向PWMCR寄存器搬数据的统一入口，仅支持word搬入，每次COM扫描结束后更新PWMCR寄存器的个数由PWMCR_LEN控制，例如PWMCR_LEN=4，则会一次更新PWMCRA~PWMCRD

33.4.12 DOTxDR 的 DMA 访问寄存器

名称	DDRAR
地址	

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

位号	位名	说明
31:0	DDRAR	DMA向DOTxDR寄存器搬数据的统一入口，仅支持word搬入，每次帧扫描结束后更新DOTxDR寄存器的个数由DOTDR_LEN控制，例如DOTDR_LEN=10，则会一次更新DOT0DR~DOT9DR

33.4.13 LED 扫描当前 COM 线

名称	LED_CCOM							
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	-						ACT_COM	
位权限	U-0						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ACT_COM							
位权限	R-00000000							

位号	位名	说明
31:4	---	**未实现，读为0**
9:0	ACT_COM	LED 模块当前正在扫描的 COM 线，每 bit 对应于一个条 COM 线

34 模数转换器（ADC）

34.1 概述

FM33LF0xxA 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 2.5～5.5V
- 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 12 个外部输入通道
- 7 个内部采样通道
  - 温度传感器
  - 内部基准测量
  - OPA1
  - OPA2
  - VREFP
  - VREFN
- 可配置的采样保持时间
- 支持单次转换和连续转换
- 采样通道支持常规组和注入组
- 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

34.2 结构框图

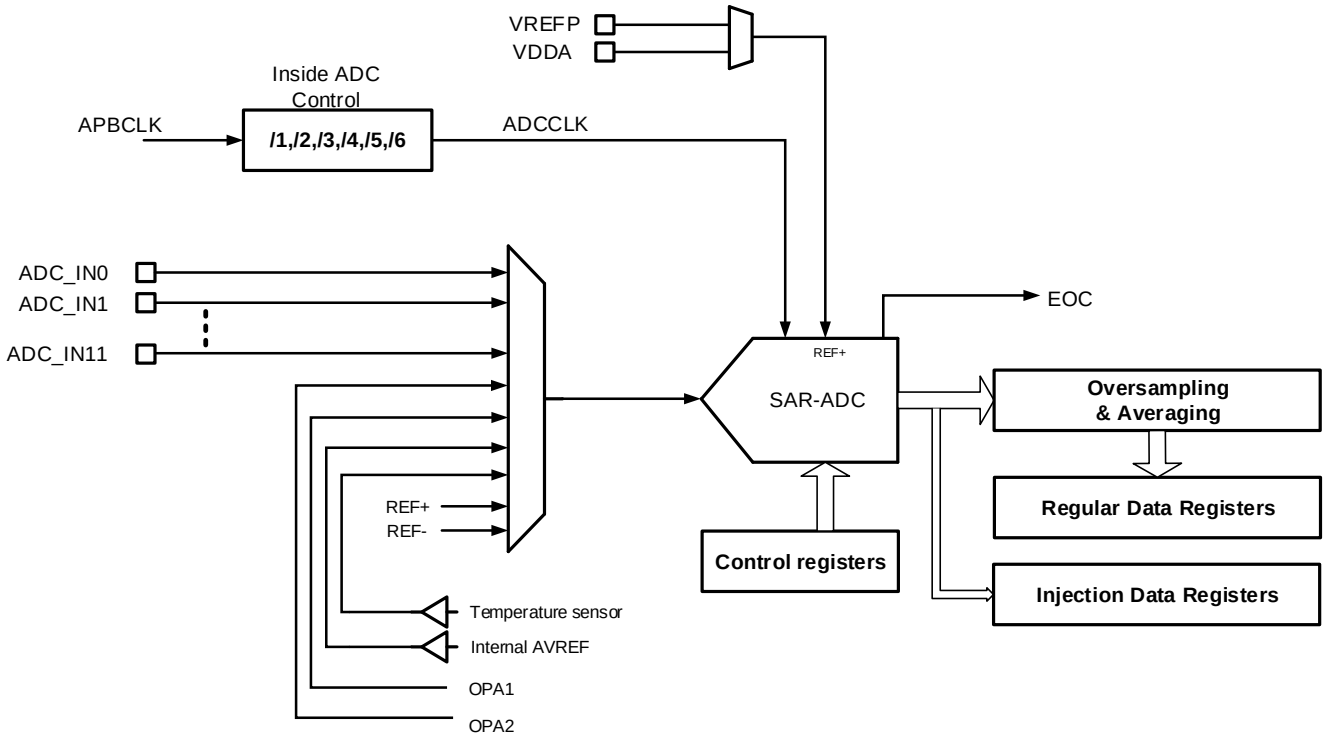


图 34-1 ADC 结构框图

34.3 输入通道

ADC 最大支持 4 个内部通道和 12 个外部通道。

ACQx_CR.CHSEL[13:9]	**通道**	**IO**	**说明**
0	ADC_IN0	PD11	外部快速通道，单端输入
1	ADC_IN1	PD0	
2	ADC_IN2	PD1	
3	ADC_IN3	PD2	
4	ADC_IN4	PA13	
5	ADC_IN5	PA14	
6	ADC_IN6	PA15	
7	ADC_IN7	PC10	
8	ADC_IN8	PC9	
9	ADC_IN9	PC8	
10	ADC_IN10	PC7	
11	ADC_IN11	PC6	
12	TS	N/A	温度传感器采样通道
13	AVREF		内部 1.0V 基准源采样通道
14	OPA1		运放输出
15	OPA2		运放输出

ACQx_CR.CHSEL[13:9]	通道	IO	说明
16	VREFP		基准电压
17	VREFN		基准地

表 34-1 ADC 输入通道

34.4 时钟与复位

34.4.1 时钟

ADC 控制器只有一个 APBCLK 输入，通过控制内部寄存器配置分频产生工作时钟 ADC_CLK。

分频电路产生 1，2，3，4，5，6 分频可选择。

在 APBCLK 时钟高于 32M 时，必须使用 2 及以上分频。

34.4.2 复位

模块使用 RMU 送来的外设复位信号 ADCCRST，会将整个数字模块进行复位，。ADC_EN 复位不仅复位数字状态机，同时也会对 ADC 模拟电路进行复位。

34.5 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 offset 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。校准方法参见“ADC 校准”章节。

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。

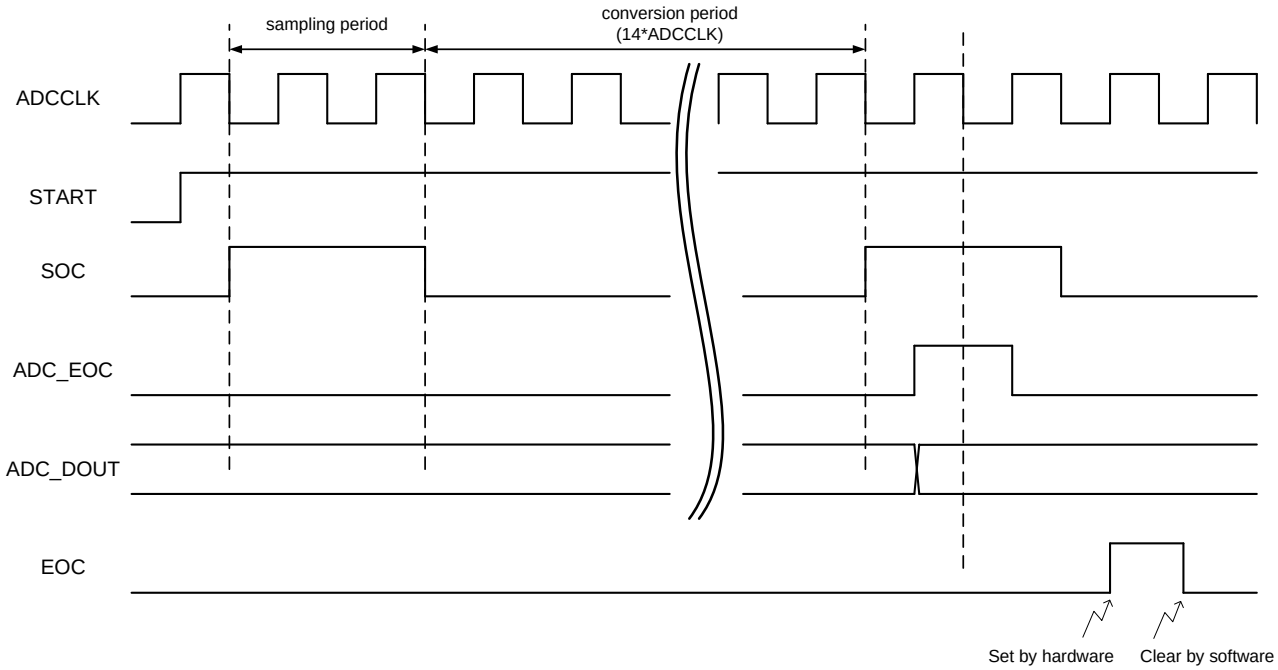


图 34-2 ADC 采样转换时序

多通道转换序列的时序示意图如下：

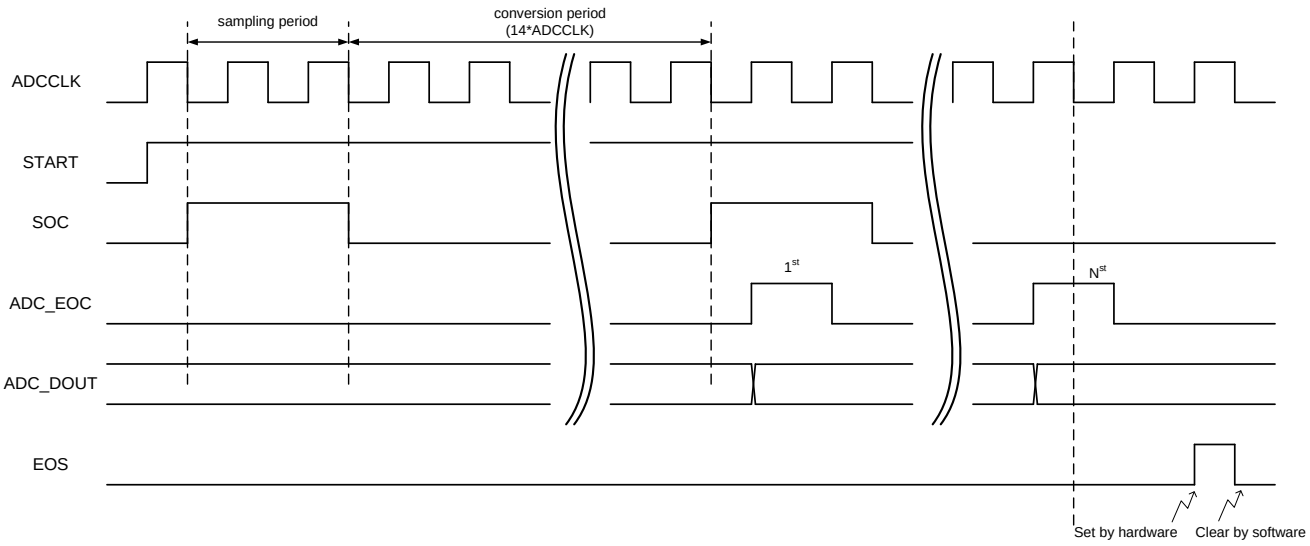


图 34-3 ADC 采样序列时序

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。

34.6 功能描述

34.6.1 ADC 转换队列（ACQ）

34.6.1.1 ACQ 描述

ADC 采样转换的配置以队列 ACQ（ADC Conversion Queue）形式组织，以获得最大的灵活性。ADC 共支持 4 个注入转换队列（ACQ0-3）以及一个扫描转换队列 SCAN_ACQ 组成，

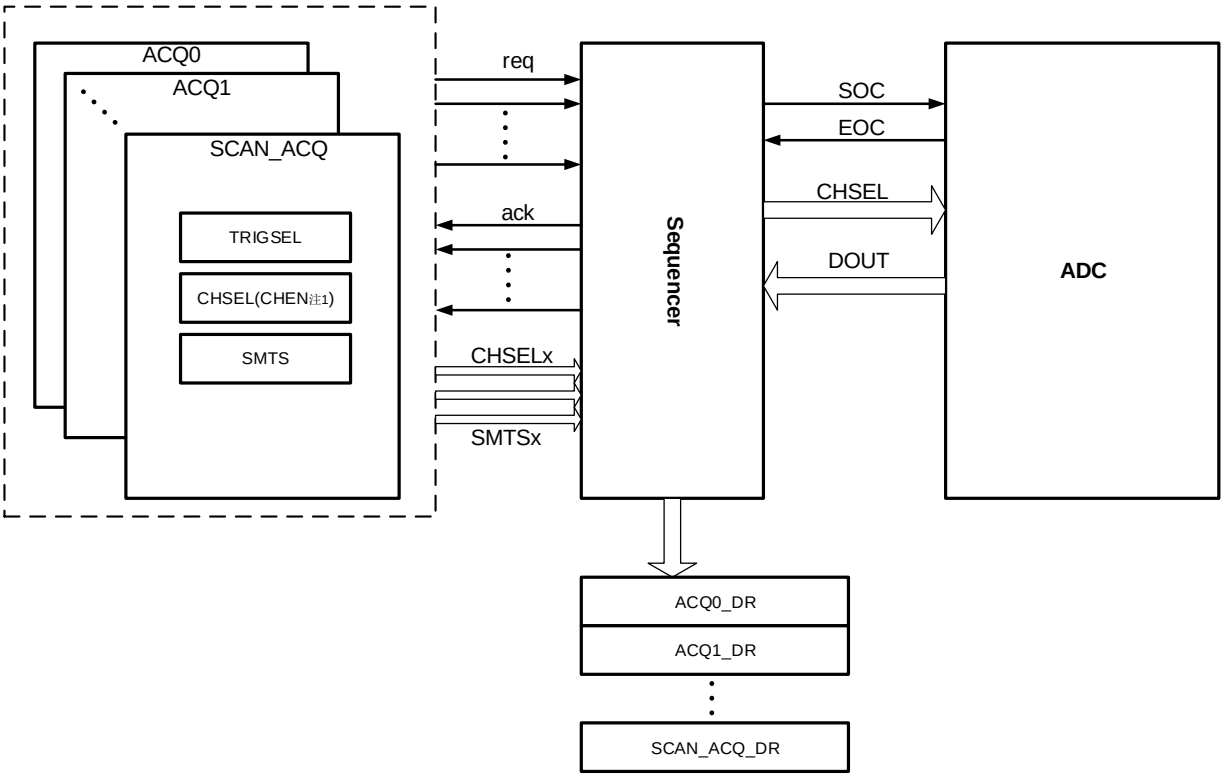


图 34-4ACQ 队列控制器

注 1：CHEN 为扫描转换队列 SCAN_ACQ 特有，注入转换队列 ACQ0~3 使用 CHSEL 配置；即注入 ACQ 通过 CHSEL 寄存器配置选择一个通道进行转换，而扫描 ACQ 可以通过 CHEN 寄存器配置多个通道进行轮询扫描转换。

由于 ACQ 之间互相独立，应用中会出现多个 ACQ 同时请求转换的情况（比如多个 ACQ 使用相同的触发源，或者多个不同触发源同时到来），此时 ADC 的 sequencer 电路通过内在优先级判决逐一顺序完成请求响应。ACQ 收到触发信号后将对 sequencer 发起 SOC 请求，sequencer 根据优先级判断逻辑选择响应最高优先级的 ACQ、发起 SOC，转换结束后再根据优先级处理下一个 ACQ 的 SOC 请求，直到所有 ACQ 请求被处理完毕。

ACQ 的工作逻辑认为注入 ACQ 是处理高实时性的转换工作，而扫描 ACQ 处理非实时需求的转换工作，所以注入转换队列拥有高优先级（ACQ0~3 彼此之间的优先级为按照编号 0~3 数字越大优先级依次降低），注入 ACQ 可以打断普通扫描 ACQ 的工作，被打断的工作将在所有触发并生效的注

入 ACQ 工作完成后继续扫描 ACQ 工作。

34.6.1.2 注入 ACQ

ACQ0~3 为注入转换队列，每个注入队列可以独立配置以下参数：

- 输入通道
- 触发条件
- 采样窗口长度
- 中断和 DMA 请求使能

每个 ACQ 独立配置，并且可以任意选择以上参数；多个 ACQ 可以配置位选择相同的通道、触发条件等。每个 ACQ 包含以下输出：

- 最大 16bit 转换结果寄存器
- 转换开始及完成标志、数据溢出标志

注入 ACQ0~3 彼此之间的优先级为按照编号 0~3 数字越大优先级依次降低，也就是说如 ACQ0,ACQ1 同时处于触发激活状态下,则优先响应 ACQ0。

34.6.1.3 扫描 ACQ

SCAN_ACQ 为扫描转换队列，此注入队列可以独立配置以下参数：

- 输入通道，通过 ACQ_SACR.CHEN 寄存器配置多个扫描通道
- 触发条件
- 采样窗口长度
- 中断和 DMA 请求使能

扫描 SCAN_ACQ 包含以下输出：

- 最大 16bit 转换结果寄存器
- CHID 寄存器，表示当前 DR 数据为哪个通道转换得来
- 转换开始及完成标志、数据溢出标志、以及扫描完成标志 EOS
- AWD 模拟比较器结果标志

34.6.2 注入 ACQ 转换

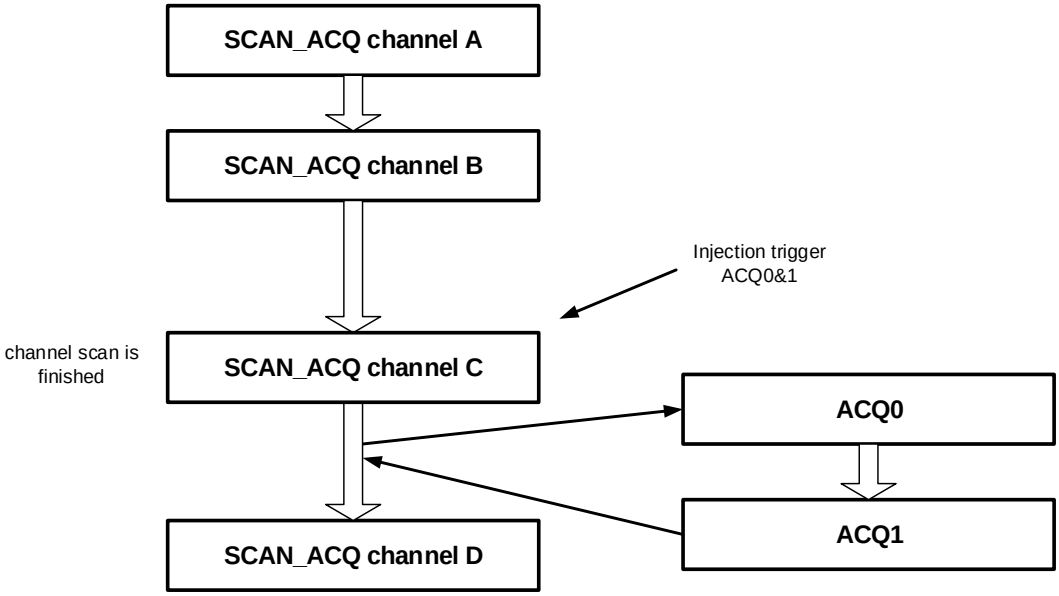
注入 ACQ 可以定义自己的采样转换配置，每个注入 ACQ 有自己的配置和使能，可以选取自己的触发进行自由触发，触发后进行优先转换，转换完成后有自己独立的标志寄存器以及 DR 寄存器。注入 ACQ 仅响应自己的触发，并且一次触发转换一次，如果需要周期性转换注入 ACQ，需选取周期性的触发源。

注入模式可以选择立即注入与延迟注入，这两种选择均为当前有 SCAN_ACQ 工作的时候注入 ACQ 的响应模式配置。

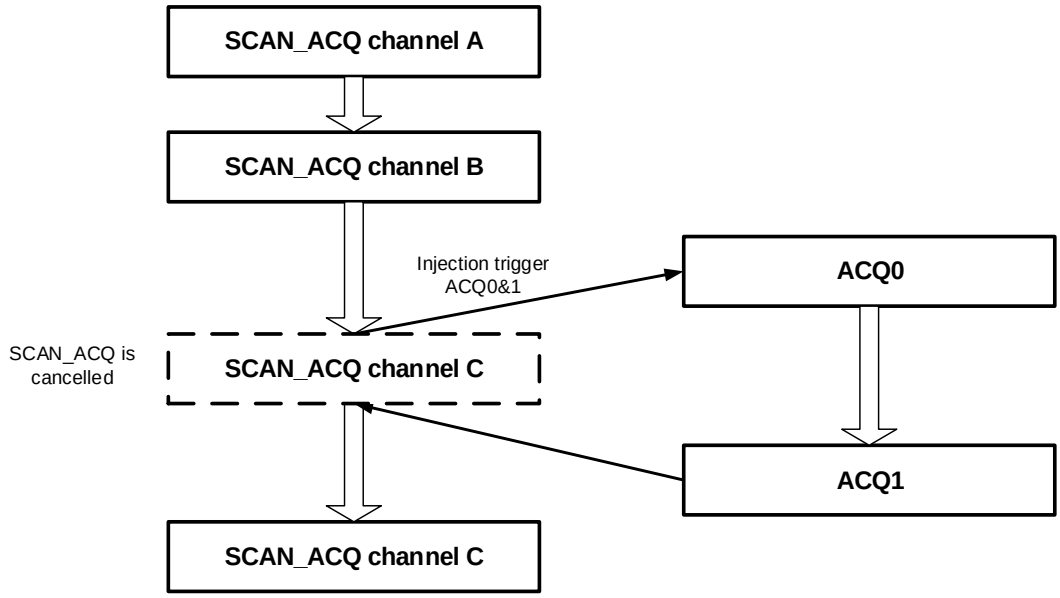
延迟注入模式，SCAN_ACQ 会执行完当前的采样转换再响应注入 ACQ 的工作，注入 ACQ 完成后，SCAN_ACQ 会继续工作，由于没有打断，看起来就像注入 ACQ 插入了 SCAN_ACQ 的工作流程中。

立即注入模式，注入 ACQ 会打断当前 SCAN_ACQ 的工作，立即执行注入 ACQ 的工作，当注入 ACQ 执行完成后回到 SCAN_ACQ 打断时的工作重新执行，此模式可以最高程度地提高 ACQ 触发响应的实时性。

- ADC_CFGR.INJ_MODE=1，延迟注入——注入事件发生时，当前工作的 SCAN_ACQ 采样转换完成后响应注入 ACQ 工作。



- ADC_CFGR.INJ_MODE=0，立即注入——注入事件发生时，理解结束当前工作的 SCAN_ACQ 采样转换工作，并应注入 ACQ 工作，完成后回到先前 SCAN_ACQ 工作。



工作时序如下：

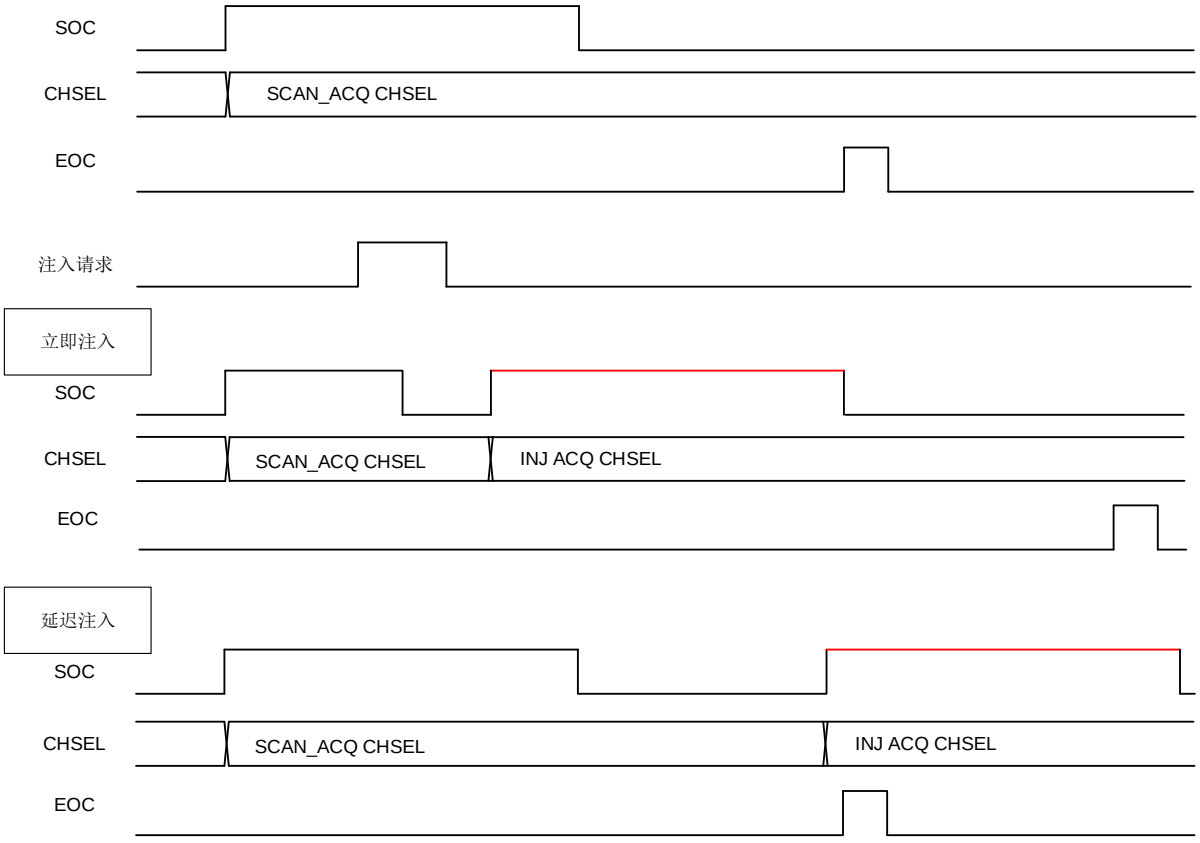


图 34-5 注入时序

34.6.3 扫描 ACQ 转换

SCAN_ACQ 支持以下转换模式：

- 单次转换 (ADC_CFGR2.CONT_MOD=0)
  - 半自动触发（SEMI-AUTOMATIC） (ADC_CFGR2.AUTO_SCAN=0)
  - 全自动触发（AUTOMATIC） (ADC_CFGR2.AUTO_SCAN=1)
- 连续转换 (ADC_CFGR2.CONT_MOD=1)

转换启动可以由软件或事件触发，通过寄存器选择事件触发源。

34.6.3.1 单次转换模式

配置 ADC_CFGR2.CONT_MOD=0 时，SCAN_ACQ 将进行单次转换模式。

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

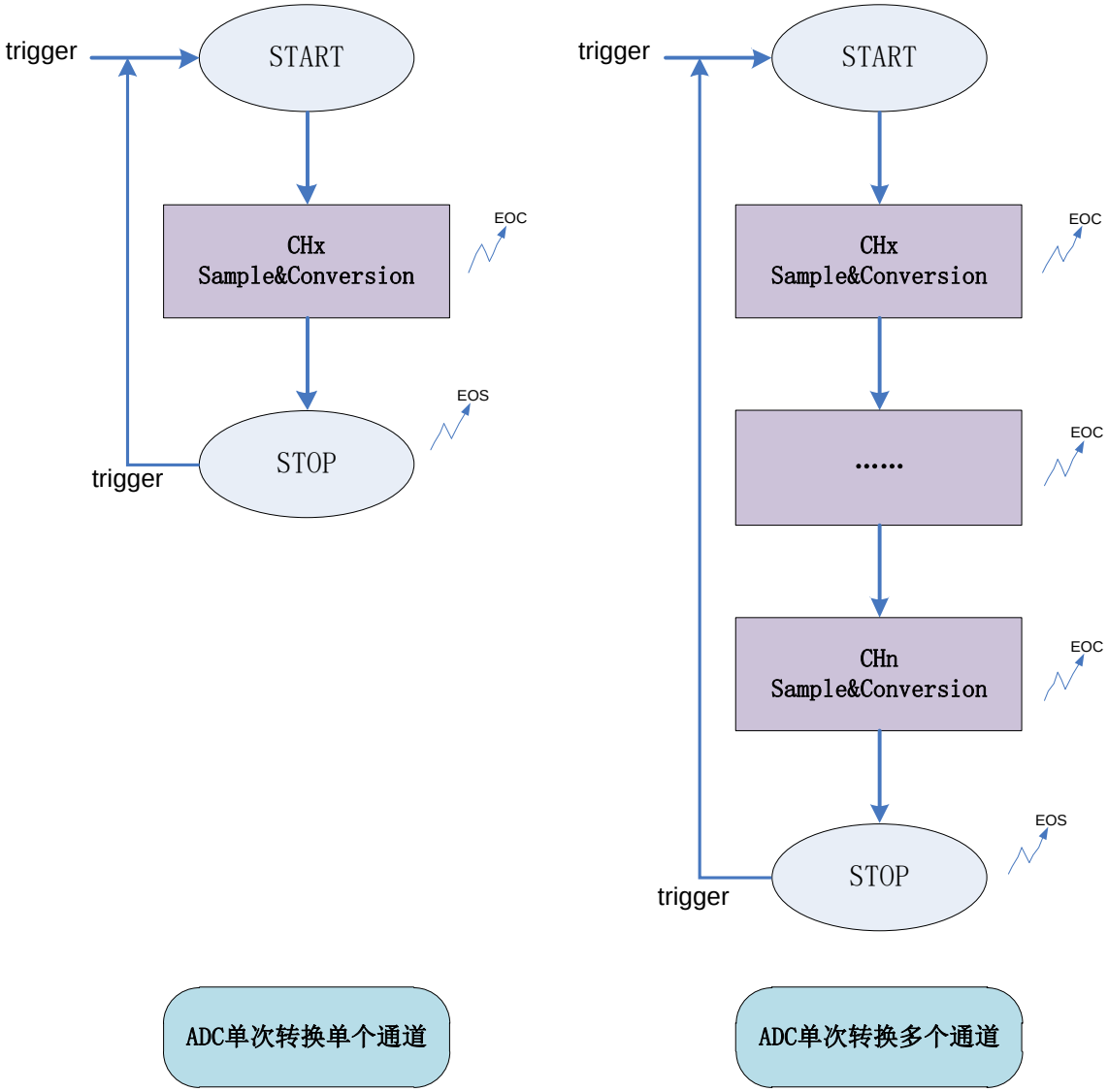


图 34-6SCAN_ACQ 单次转换全自动触发模式

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

……

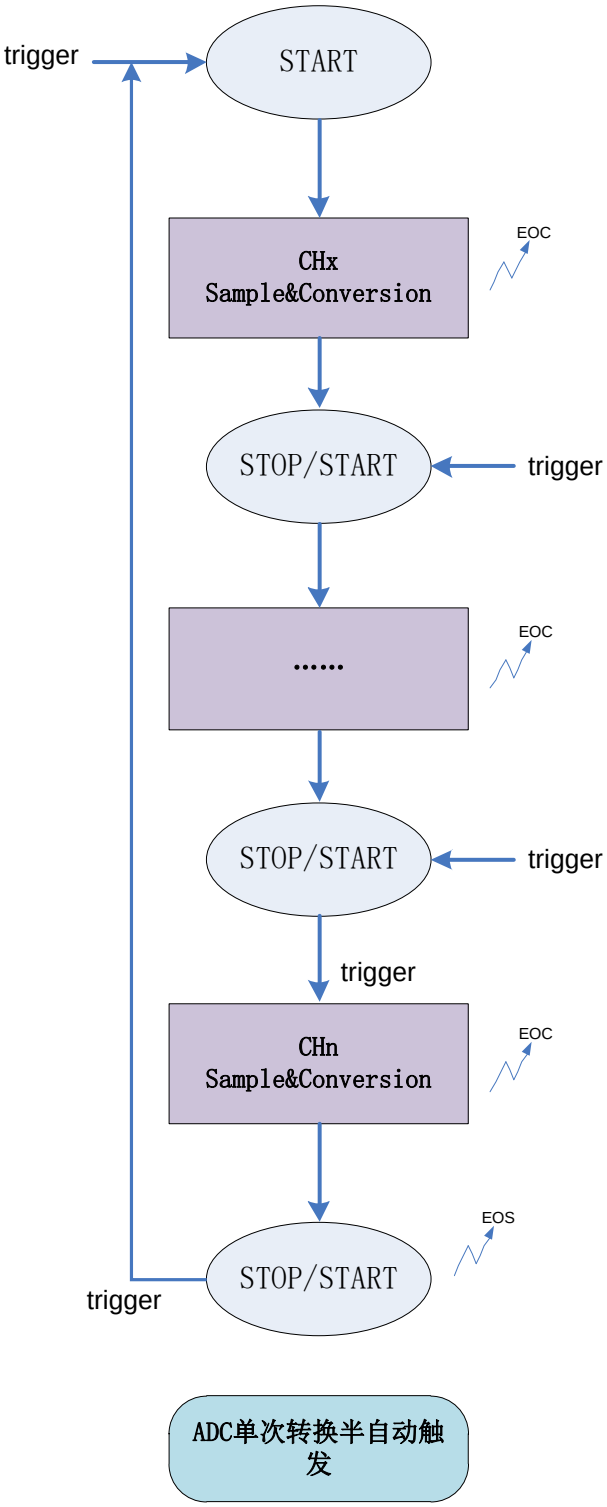


图 34-7SCAN_ACQ 单次转换半自动触发模式

34.6.3.2 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，直到以下两种情

况下停止：1、扫描转换次数达到 RPTLEN。2、ADC_CFGR2.CONT_MOD 写 0。

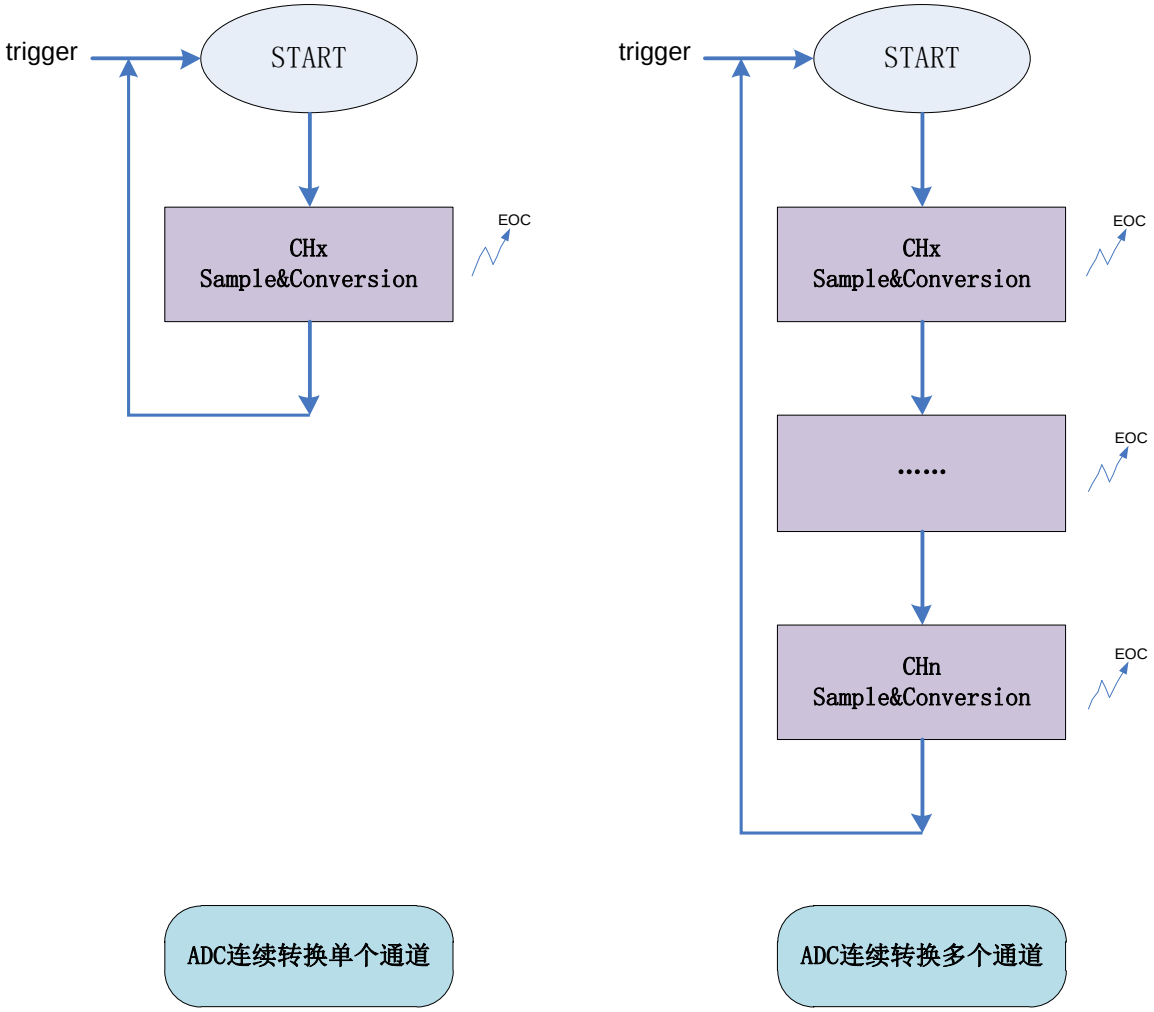


图 34-8SCAN_ACQ 连续转换模式

每个通道被采样后，数据保存在 SCAN_ACQ_DR 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。

RPTLEN=0 时则连续转换模式不会自动停止，直到 ADC_CFGR2.CONT_MOD 写 0 则停止。

34.6.4 触发源

不管是注入 ACQ 还是 SCAN_ACQ，都有各自的触发源选择（TRIGSEL），触发源如下表

TRIGSEL	触发信号	说明
0x1F~0x19		无触发
0x18	BSTIM_TRGO	
0x17	LPTIM_TRGO	
0x16	COMP4_TRGO	
0x15	COMP3_TRGO	
0x14	COMP2_TRGO	

0x13	COMP1_TRGO	
0x12~0xB		无触发
0xA	GPTIM1_TRGO	
0x9	GPTIM0_TRGO	
0x8		无触发
0x7	ATIM_TRGO2	
0x6		无触发
0x5	ATIM _TRGO	
0x4	LUT3_TRGO	
0x3	LUT2_TRGO	
0x2	LUT1_TRGO	
0x1	LUT0_TRGO	
0x0	SW_TRIG	软件写 CR2 对应 bit 寄存器

表 34-2 ACQ 触发源

由于 ADC 为同步设计，所以 ACQ 的触发源信号不需要同步，仅对上升沿的进行触发操作。

34.6.5 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，可以采用以下方案。

使用快速内部基准作为参考信号

芯片内部还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，AVREF 在量产测试中的测量电压是 AVREF_MEASURE）：

$$VDDA = \frac{4095}{AVREF_DATA} \times AVREF_MEASURE$$

如果对精度要求不高，也可以用以下简化公式代替：

$$VDDA = \frac{4095}{AVREF_DATA} \times 1.0V$$

- 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）

$$V_{CHANNEL} = \frac{ADC_DATA}{AVREF_DATA} \times AVREF_MEASURE$$

如果对精度要求不高，也可以用以下简化公式代替：

$$V_{CHANNEL} = \frac{ADC_DATA}{AVREF_DATA} \times 1.0V$$

34.6.6 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（其中 VREF）：

1) 由以下公式计算当前温度传感器输出的绝对电压值

$$V_{PTAT} = \frac{TS_DATA}{AVREF_DATA} \times AVREF_MEASURE$$

2) 由以下公式计算温度标定时（30°C，VDDA=3V）温度传感器输出的绝对电压值，其中 TS_CAL30 是保存在 flash 中的标定数据

$$V_{PTAT_30C} = \frac{TS_CAL30}{4095} \times 3.0V$$

3) 根据温度传感器输出斜率计算当前绝对温度，其中 slope 为温度传感器输出斜率，参见电参数章节

$$Temperature = \frac{V_{PTAT} - V_{PTAT_30C}}{slope} + 30C$$

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 AVREF 的转换结果进行比例缩放；TS_CAL30 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。

如果已知 ADC 当前工作参考电压是 VREFP，则当前温度可以通过下式计算：

$$Temperature = \frac{TS_DATA \times \frac{VREFP}{3.0V} - TS_CAL30}{slope} + 30C$$

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。

34.6.7 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 ACQX_CR.SMTS 寄存器可以选择采样

时间：

SMTS	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 34-3 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：

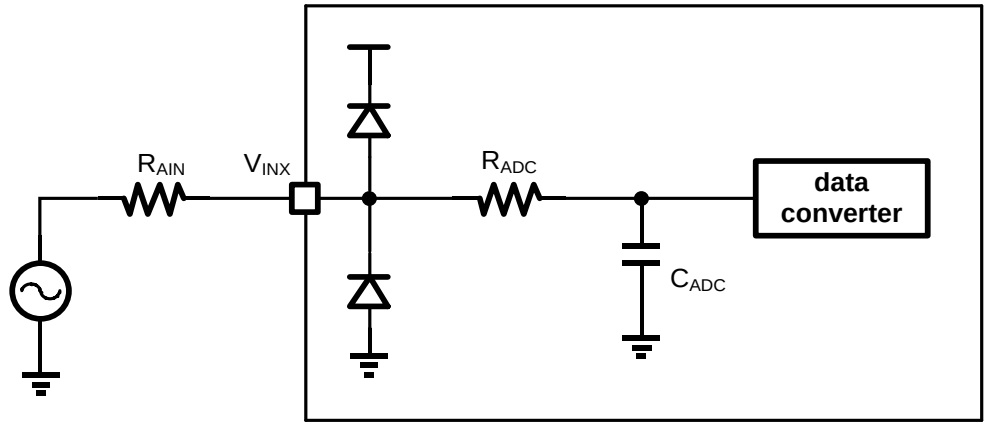


图 34-9 ADC 输入通道示意图

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。

34.6.8 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。

34.6.9 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 读取对应 ACQ 的 DR 寄存器后会自动清除对应的 EOC 标志，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

ADC 控制器还支持自动等待，如果 ADC_CFGR2.WAIT 寄存器被软件置位，那么当 EOC 标志没有被清除的情况下，此 ACQ 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，等待状态仅暂停当前 ACQ 的工作，其他 ACQ 不受影响。

34.6.10 DMA

所有的 ACQ 都可以独立产生 DMA 请求信号。在 ADC_CFGR2.DMAEN 寄存器置 1 后，DMA 请求信号同 EOCIF 产生逻辑相同，在 EOC 产生同时发出 DMA 请求，DMA 响应做数据搬移操作，而

DMA 仅做数据搬运动作，同时，数据搬移会使对应的 ACQ_ISR.EOCIFx 标志清零。

对于注入 ACQ，由于 ACQ_CR.CHSEL 的通道选择是配置为固定的，所以 DR 寄存器对应的就是配置的通道，不会有混淆的可能，所以 DMA 搬运的 DR 原本数据。对于 SCAN_ACQ 来说，其通道可以是使能了的任意一个通道，所以在 DMA 搬运的时候会在 SCAN_ACQ_DR 寄存器的数据里增加通道 CHID 来标识当前搬运出的数据属于哪一个通道的转换值。

由于 DMA 仅负责搬移数据，所以使用的时候只需要根据需求配置 ADC 的行为，DMA 的操作不会对 ADC 的操作产生，所以使用 ADC 的 DMA 功能时，需要同时考虑 ADC 的配置和 DMA 的配置相匹配。

34.6.11 模拟窗口看门狗（AWD）

AWD 功能用于监视某些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，AWD 功能仅对 SCAN_ACQ 有效，并且通过配置 ADC_AWDCR.AWD_CHEN 寄存器使能需要监控的通道，可以一次性使能多个通道，则每个通道的都会被监视，如果没有通道使能，则 AWD 功能关闭。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。

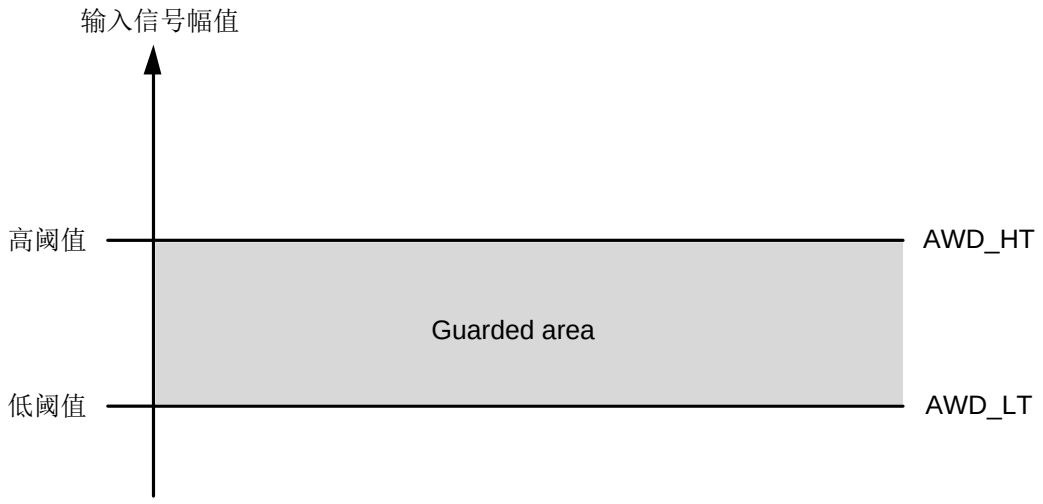


图 34-10 ADC 模拟看门狗阈值示意图

AWD 阈值设置寄存器的位宽是 16bit，应对与过采样时可能得到的 16bit 数据。

34.6.12 ADC 校准

ADC 支持比较器 offset 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。

ADC offset 校准采用以下方法：

校准流程：

- ADC_OSR.CALI_EN 寄存器写入 1
- 使能 ACQ0，采样通道选择 VREFP，使能过采样 32 次，结果右移 5bit，选择软件触发
- 使能 ACQ1，采样通道选择 VREFN，使能过采样 32 次，结果右移 5bit，选择软件触发
- 写软件启动触发寄存器 CR2 同时触发 ACQ0 和 ACQ1 工作等待 ACQ1 的 EOCIF 标志
- 从 ACQ0_DR 得到 D0，从 ACQ1_DR 得到 D1，计算 Doffset

$$D_{offset13} = \frac{D0 + D1 - 4095}{2}$$

在不做校准的情况下，Doffset 默认值为 136

- 结果写入 ADC_OSR，同时 CALI_EN 位写入 0
- 校准完成

34.7 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。

34.8 寄存器

模块起始地址：0x4001_5C00

offset 地址	名称	符号
0x00000000	ACQ 中断和状态寄存器	ACQ_ISR
0x00000004	ACQ 中断使能寄存器	ACQ_IER
0x00000008	通用中断和状态寄存器	ADC_GISR
0x0000000C	通用中断使能寄存器	ADC_GIER
0x00000010	控制寄存器 1	ADC_CR1
0x00000014	控制寄存器 2	ADC_CR2
0x0000001C	配置寄存器 1	ADC_CFGR1
0x00000020	配置寄存器 2	ADC_CFGR2
0x0000002C	校准数据寄存器	ADC_OSR
0x00000030	ACQ 标志寄存器	ACQ_SR
0x00000034	Scan ACQ 控制寄存器	ACQ_SACR
0x00000038	AWD 阈值配置寄存器	ADC_HLTR
0x0000003C	AWD 阈值配置寄存器	ADC_AWDCR
0x00000040	ACQ0 控制寄存器	ACQ0_CR
0x00000044	ACQ1 控制寄存器	ACQ1_CR
0x00000048	ACQ2 控制寄存器	ACQ2_CR
0x0000004C	ACQ3 控制寄存器	ACQ3_CR
0x00000050	SCAN_ACQ 控制寄存器	SCAN_ACQ_CR
0x00000080	ACQ0 数据寄存器	ACQ0_DR
0x00000084	ACQ1 数据寄存器	ACQ1_DR
0x00000088	ACQ2 数据寄存器	ACQ2_DR

offset 地址	名称	符号
0x0000008C	ACQ3 数据寄存器	ACQ3_DR
0x00000090	SCAN_ACQ 数据寄存器	SCAN_ACQ_DR

34.8.1 ACQ 中断和状态寄存器（ACQ_ISR）

名称	ACQ_ISR							
地址	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		OVR					
位权限	U-00		W1C-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			SOCIF				
位权限	U-000			W1C-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			EOCIF				
位权限	U-000			W1C-00000				

位	位名	功能描述
31:22	RFU	RFU：未实现，读为 0
21:16	OVR	ACQ 数据溢出标志，每个 bit 对应一个 ACQ： OVR[4] – SCAN_ACQ OVR[3] – ACQ3 … OVR[0] – ACQ0 0：没有发生 over-run 1：发送 over-run 此标志表明软件没能计数处理转换的数据，产生的条件为对应 EOCIF 有效时新的转换完成。
15:13	RFU	RFU：未实现，读为 0
12:8	SOCIF	ACQ 转换开始（Start Of Conversion） 每个 ACQ 转换开始后，自动置位 SOCIF，软件写 1 清零。用以节约程序进入中断的压栈动作，进入中断后查询 EOCIF 判断数据是否准备好。如果开启立即注入模式，请勿使用 SCAN_ACQ 的 SOCIF 标志，因为可能转换会被打断。此标志可根据 SOCIF_CFG 配置选择是在 SOC 上升沿还是下降沿触发。 SOCIF[4] – SCAN_ACQ SOCIF[3] – ACQ3 … SOCIF[0] – ACQ0
7:5	RFU	RFU：未实现，读为 0
4:0	EOCIF	ACQ 转换结束（End Of Conversion） 每个 ACQ 转换完成后，自动置位 EOCIF，软件写 1 清零，或者读取 DR 寄存器时自动清零。 EOCIF[4] – SCAN_ACQ EOCIF[3] – ACQ3 …

位	位名	功能描述
		EOCIF[0] – ACQ0

34.8.2 ACQ 中断使能寄存器（ACQ_IER）

名称	ACQ_IER							
地址	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		OVRIE					
位权限	U-00		RW-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			SOCIE				
位权限	U-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			EOCIE				
位权限	U-000			RW-00000				

位	位名	功能描述
31:22	RFU	RFU：未实现，读为 0
21:16	OVRIE	ACQ 数据溢出中断使能，每个 bit 对应一个 ACQ： OVRIE[4] – SCAN_ACQ OVRIE[3] – ACQ3 … OVRIE[0] – ACQ0 0：禁止 OVR 中断 1：允许 OVR 中断
15:13	RFU	RFU：未实现，读为 0
12:8	SOCIE	ACQ 转换开始中断使能（Start Of Conversion） SOCIE[4] – SCAN_ACQ SOCIE[3] – ACQ3 … SOCIE[0] – ACQ0 0：禁止 SOC 中断 1：允许 SOC 中断
7:5	RFU	RFU：未实现，读为 0
4:0	EOCIE	ACQ 转换结束中断使能（End Of Conversion） EOCIE[4] – SCAN_ACQ EOCIE[3] – ACQ3 … EOCIE[0] – ACQ0 0：禁止 EOC 中断 1：允许 EOC 中断

34.8.3 通用中断和状态寄存器（ADC_GISR）

名称	ADC_GISR							
地址	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		SCAN_CH					
位权限	U-00		R-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			ACQ_ID				
位权限	U-000			R-11111				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	AWD_AH	AWD_UL	RFU	BUSY	EOCNT	EOS	RFU
位权限	U-0	RW-0	RW-0	U-0	R-0	RW-0	R-0	U-0

位	位名	功能描述
31:22	RFU	RFU：未实现，读为 0
21:16	SCAN_CH	当前 SCAN_ACQ 正在扫描的通道，只读，本寄存器尽在 ACQ_ID 为 SCAN_ACQ 的时候有效。
15:13	RFU	RFU：未实现，读为 0
12:8	ACQ_ID	当前工作的 ACQ 编号，只读，ADC 工作时为当前 ACQ 的 ID，无 ACQ 工作时则显示 0x1F,表示没有 ACQ 激活。
7	RFU	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	RFU	RFU：未实现，读为 0
3	BUSY	ADC 忙标志 1：ADC 正在采样转换或者正在校准 0：ADC 空闲
2	EOCNT	SCAN_ACQ 连续模式完成，即 SCAN_ACQ 累积转换次数达到 RLEN 值
1	EOS	SCAN_ACQ 扫描序列转换结束（End Of Scan）中断标志 按照扫描方向扫描完最后一个通道后置起此标志，硬件置 1，软件写 1 清零
0	RFU	RFU：未实现，读为 0

34.8.4 通用中断使能寄存器（ADC_GIER）

名称	ADC_GIER							
地址	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

名称	ADC_GIER							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	AWD_AHIE	AWD_ULIE	RFU	RFU	EOCNT_IE	EOSIE	RFU
位权限	U-0	RW-0	RW-0	U-0	U-0	RW-0	RW-0	U-0

位	位名	功能描述
31:7	RFU	RFU：未实现，读为 0
6	AWD_AHIE	AWD_AH 标志中断使能
5	AWD_ULIE	AWD_UL 标志中断使能
4	RFU	RFU：未实现，读为 0
3	RFU	RFU：未实现，读为 0
2	EOCNT_IE	EOCNT 标志中断使能
1	EOSIE	EOS 标志中断使能
0	RFU	RFU：未实现，读为 0

34.8.5 控制寄存器 1（ADC_CR1）

名称	ADC_CR1							
地址	0x00000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		ACQEN					
位权限	U-00		RW-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					SCH_SWRST	RR_SWRST	ADEN
位权限	U-00000					WO-0	WO-0	RW-0

位	位名	功能描述
31:22	RFU	RFU：未实现，读为 0
21:16	ACQEN	ACQ 使能（ACQ Enable） ACQEN[4] – SCAN_ACQ ACQEN[3] – ACQ3(INJ) ACQEN[2] – ACQ2(INJ) ACQEN[1] – ACQ1(INJ) ACQEN[0] – ACQ0(INJ) 0：使能对应 ACQ 1：关闭对应 ACQ
15:3	RFU	RFU：未实现，读为 0

位	位名	功能描述
2	SCH_SWRST	SCAN ACQ 的转换扫描通道复位重置，让扫描从最低的通道号开始，如果扫描方向反向，则从最高的通道号开始。
1	RR_SWRST	普通转换组的 round-robin 状态复位，可以重置优先级，让 ACQ0 为最高优先级状态
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

34.8.6 控制寄存器 2（ADC_CR2）

名称	ADC_CR2							
地址	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			SWTRIG				
位权限	U-000			WO-00000				

位	位名	功能描述
31:5	RFU	RFU：未实现，读为 0
4:0	SWTRIG	ACQ 转换软件触发 每个 ACQ 选择软件触发后，写此位会触发此 ACQ 转换。 BIT4 – SCAN_ACQ BIT3 – ACQ3 … BIT0 – ACQ0

34.8.7 配置寄存器 1（ADC_CFGR1）

名称	ADC_CFGR1							
地址	0x0000001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RPTLEN							
位权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RPTLEN							
位权限	RW-00110000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							SOCIF_CFG
位权限	U-0000000							RW-1

名称	ADC_CFGR1							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PCLK_PSC			VCM_CTRL	INJ_MODE	REFNSEL	RFU	REFPSEL
位权限	RW-001			RW-0	RW-0	RW-0	U-0	RW-0

位	位名	功能描述
31:16	RPTLEN	连续模式下重复转换次数，仅对 SCAN_ACQ 有效，在开启了连续模式后，转换会不停止地进行直到达到此次数后停止
15:9	RFU	RFU：未实现，读为 0
8	SOCIF_CFG	SOCIF 中断标志置起点配置 0：SOC 上升沿出置起 SOCIF 标志 1：SOC 下降沿处置起 SOCIF 标志
7:5	PCLK_PSC	APBCLK 分频作为 ADC 工作时钟的预分频 (APBCLK prescaler) 001：不分频 010：2 分频 011：3 分频 100：4 分频 101：5 分频 110：6 分频 other：不分频
4	VCM_CTRL	VCM 控制选择
3	INJ_MODE	注入 ACQ 的注入模式 1：立即注入转换，立即中断当前转换，开始注入 ACQ 转换 0：延迟注入转换，当前 ACQ 转换完成后注入
2	REFNSEL	ADC 负基准选择 (Negative Reference Select) 0：VSSA 1：VREFN
1	RFU	RFU：未实现，读为 0
0	REFPSEL	ADC 正基准选择 (Positive Reference Select) 0：使用 VDDA 作为基准 1：使用 VREFP 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

34.8.8 配置寄存器 2（ADC_CFGR2）

名称	ADC_CFGR2							
地址	0x00000020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							AVGR_OVSS
位权限	U-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AVGR_OVSS			AVGR_OVSR			AVGEN	
位权限	RW-000			RW-000			RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				WAIT_F	AUTO_SCAN	CONT_MOD	OVRM

名称	ADC_CFGR2							
位权限	U-0000				RW-0	RW-1	RW-0	RW-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					SCAN_DIR	RFU	DMAEN
位权限	U-00000					RW-0	U-0	RW-0

位	位名	功能描述
31:25	RFU	RFU：未实现，读为 0
24:21	AVGR_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
20:18	AVGR_OVSR	过采样率控制 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x
17:16	AVGEN	过采样使能， 00/11：禁止过采样 01：使能过采样，并且自动得到平均数，等过采样完成后得到一个数据以及一个 EOCIF 标志的产生
15:12	RFU	RFU：未实现，读为 0
11	WAIT_F	等待模式使能，对所有 ACQ 有效，如果同一个 ACQ 转换完成后没有将数据取走，新的触发则会等待数据取走后继续转换，如果期间发生多次触发，仅当一次处理
10	AUTO_SCAN	自动扫描模式开启，仅当 CONT_MOD 关闭时有效,仅对 SCAN_ACQ 有效 0：一次 SCAN_ACQ 的触发仅转换一个扫描通道，下次触发会转换下一个使能了的通道。 1：一次 SCAN_ACQ 的触发会转换所有使能了的扫描通道
9	CONT_MOD	连续模式使能，仅对 SCAN_ACQ 有效，一次触发会对扫描 ACQ 使能的扫描通道进行不间断的扫描转换，直到 1：扫描转换次数达到 RPTLEN。 2：CONT_MOD 写 0。 0：关闭连续模式 1：开启连续模式
8	OVRM	Over-run 模式控制 0：当 over-run 发生时，保持上次数据，丢弃本次转换值 1：当 over-run 发生时，覆盖上次数据
7:3	RFU	RFU：未实现，读为 0

位	位名	功能描述
2	SCAN_DIR	扫描 ACQ 的通道扫描方向，仅对 SCAN_ACQ 有效 0：反向扫描，如 CH15->CH14... 1：正向扫描，如 CH1->CH2...
1	RFU	RFU：未实现，读为 0
0	DMAEN	DMA 使能 0：禁止发出 DMA 请求 1：允许发出 DMA 请求

34.8.9 校准数据寄存器（ADC_OSR）

名称	ADC_OSR							
地址	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU			RFU				
位权限	U-000			U-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							CALI_EN
位权限	U-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			OS_CALI				
位权限	U-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OS_CALI							
位权限	RW-10001000							

位	位名	功能描述
31:29	RFU	RFU:未实现，读为 0
28:17	RFU	RFU:未实现，读为 0
16	CALI_EN	通道较准使能，使能后，ADC 转换结果不会根据 OS_CALI 值进行调整，为 ADC 的输出的 rawdata，注意，仅用于较准过程，常规转换时必须配置为 0
15:13	RFU	RFU:未实现，读为 0
12:0	OS_CALI	通道校准值

34.8.10 ACQ 标志寄存器（ACQ_SR）

名称	ACQ_SR							
地址	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REQ							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	ACQ_SR
位名	REQ
位权限	R-00000000

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	REQ	ACQ 请求 pending 标志 1：对应 ACQ 有请求待处理 0：对应 ACQ 的采样请求已经被响应，或者无请求

34.8.11 Scan ACQ 控制寄存器（ACQ_SACR）

名称	ACQ_SACR							
地址	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						CHEN	
位权限	U-000000						RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHEN							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHEN							
位权限	RW-00000001							

位	位名	功能描述
31:18	RFU	RFU：未实现，读为 0
17:0	CHEN	scan ACQ 的通道使能，每一 bit 按照顺序对应于 ADC 的通道，为 1 则代表会扫描对应的通道。此寄存器仅对 SCAN_ACQ 工作时有效 如 CHEN=0x0005，表示 scan_ACQ 会扫描 0 通道和 2 通道

34.8.12 AWD 阈值配置寄存器（ADC_HLTR）

名称	ADC_HLTR							
地址	0x00000038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	AWD_HT							
位权限	RW-00001000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	AWD_HT							
位权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AWD_LT							
位权限	RW-00000001							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AWD_LT							

名称	ADC_HLTR
位权限	RW-00000000

位	位名	功能描述
31:16	AWD_HT	AWD 高阈值
15:0	AWD_LT	AWD 低阈值

34.8.13 AWD 阈值配置寄存器（ADC_AWDCR）

名称	ADC_AWDCR							
地址	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	AWD_CHEN							
位权限	RW-00000001							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AWD_CHEN							
位权限	RW-00000000							

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	AWD_CHEN	监控通道配置，和 SACR 寄存器中的 CHEN 一一对应，对应通道使能后，就会监控对应的通道，此寄存器仅对 SCAN_ACQ 有效 如 AWD_CHEN=0x0005，表示 scan_ACQ 扫描时，AWD 会检测 0 通道和 2 通道

34.8.14 ACQ0 控制寄存器（ACQ0_CR）

名称	ACQ0_CR							
地址	0x00000040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		CHSEL					TRIGSEL
位权限	U-00		RW-00000					RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGSEL				SMTS			
位权限	RW-0000				RW-0000			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0
13:9	CHSEL	通道选择，参见“输入通道”章节
8:4	TRIGSEL	触发源选择,5bit 配置参考“触发源”章节
3:0	SMTS	采样时间配置:ADC_CLK 周期数(禁止配置 0x0) 0x1 ： 2 0x2 ： 4 0x3 ： 8 0x4 ： 12 0x5 ： 16 0x6 ： 32 0x7 ： 64 0x8 ： 80 0x9 ： 96 0xa ： 128 0xb ： 160 0xc ： 192 0xd ： 256 0xe ： 320 0xf ： 512 其他：RFU。注意，转换周期最少 2 个 ADC_CLK

34.8.15 ACQ1 控制寄存器（ACQ1_CR）

名称	ACQ1_CR							
地址	0x00000044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		CHSEL					TRIGSEL
位权限	U-00		RW-00000					RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGSEL				SMTS			
位权限	RW-0000				RW-0000			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0
13:9	CHSEL	通道选择，参见“输入通道”章节
8:4	TRIGSEL	触发源选择,5bit 配置参考“触发源”章节
3:0	SMTS	采样时间配置:ADC_CLK 周期数(禁止配置 0x0) 0x1 ： 2 0x2 ： 4 0x3 ： 8

位	位名	功能描述
		0x4 ： 12 0x5 ： 16 0x6 ： 32 0x7 ： 64 0x8 ： 80 0x9 ： 96 0xa ： 128 0xb ： 160 0xc ： 192 0xd ： 256 0xe ： 320 0xf ： 512 其他：RFU。注意，转换周期最少 2 个 ADC_CLK

34.8.16 ACQ2 控制寄存器（ACQ2_CR）

名称	ACQ2_CR							
地址	0x00000048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		CHSEL					TRIGSEL
位权限	U-00		RW-00000					RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGSEL				SMTS			
位权限	RW-0000				RW-0000			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0
13:9	CHSEL	通道选择，参见“输入通道”章节
8:4	TRIGSEL	触发源选择,5bit 配置参考“触发源”章节
3:0	SMTS	采样时间配置:ADC_CLK 周期数(禁止配置 0x0) 0x1 ： 2 0x2 ： 4 0x3 ： 8 0x4 ： 12 0x5 ： 16 0x6 ： 32 0x7 ： 64 0x8 ： 80 0x9 ： 96 0xa ： 128 0xb ： 160

位	位名	功能描述
		0xc ： 192 0xd ： 256 0xe ： 320 0xf ： 512 其他：RFU。注意，转换周期最少 2 个 ADC_CLK

34.8.17 ACQ3 控制寄存器（ACQ3_CR）

名称	ACQ3_CR							
地址	0x0000004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		CHSEL					TRIGSEL
位权限	U-00		RW-00000					RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGSEL				SMTS			
位权限	RW-0000				RW-0000			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0
13:9	CHSEL	通道选择，参见“输入通道”章节
8:4	TRIGSEL	触发源选择,5bit 配置参考“触发源”章节
3:0	SMTS	采样时间配置:ADC_CLK 周期数(禁止配置 0x0) 0x1 ： 2 0x2 ： 4 0x3 ： 8 0x4 ： 12 0x5 ： 16 0x6 ： 32 0x7 ： 64 0x8 ： 80 0x9 ： 96 0xa ： 128 0xb ： 160 0xc ： 192 0xd ： 256 0xe ： 320 0xf ： 512 其他：RFU。注意，转换周期最少 2 个 ADC_CLK

34.8.18 SCAN_ACQ 控制寄存器（SCAN_ACQ_CR）

名称	SCAN_ACQ_CR							
地址	0x00000050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		RFU					TRIGSEL
位权限	U-00		U-00000					RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGSEL				SMTS			
位权限	RW-0000				RW-0000			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0
13:9	RFU	RFU：未实现，读为 0
8:4	TRIGSEL	触发源选择,5bit 配置参考“触发源”章节
3:0	SMTS	采样时间配置:ADC_CLK 周期数(禁止配置 0x0) 0x1 ： 2 0x2 ： 4 0x3 ： 8 0x4 ： 12 0x5 ： 16 0x6 ： 32 0x7 ： 64 0x8 ： 80 0x9 ： 96 0xa ： 128 0xb ： 160 0xc ： 192 0xd ： 256 0xe ： 320 0xf ： 512 其他：RFU。注意，转换周期最少 2 个 ADC_CLK

34.8.19 ACQ0 数据寄存器（ACQ0_DR）

名称	ACQ0_DR							
地址	0x00000080							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

名称	ACQ0_DR							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESULT							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RESULT							
位权限	R-00000000							

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	RESULT	ADC 转换结果

34.8.20 ACQ1 数据寄存器（ACQ1_DR）

名称	ACQ1_DR							
地址	0x00000084							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESULT							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RESULT							
位权限	R-00000000							

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	RESULT	ADC 转换结果

34.8.21 ACQ2 数据寄存器（ACQ2_DR）

名称	ACQ2_DR							
地址	0x00000088							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESULT							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

名称	ACQ2_DR
位名	RESULT
位权限	R-00000000

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	RESULT	ADC 转换结果

34.8.22 ACQ3 数据寄存器（ACQ3_DR）

名称	ACQ3_DR							
地址	0x0000008C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
位权限	U-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESULT							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RESULT							
位权限	R-00000000							

位	位名	功能描述
31:16	RFU	RFU：未实现，读为 0
15:0	RESULT	ADC 转换结果

34.8.23 SCAN_ACQ 数据寄存器（SCAN_ACQ_DR）

名称	SCAN_ACQ_DR							
地址	0x00000090							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU			CHID				
位权限	U-000			R-00000				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESULT							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RESULT							
位权限	R-00000000							

位	位名	功能描述
31:21	RFU	RFU：未实现，读为 0

位	位名	功能描述
20:16	CHID	ADC 转换结果对应的通道 ID
15:0	RESULT	ADC 转换结果

35 运算放大器（OPA）

35.1 概述

本芯片集成2个运算放大器，可用于放大微弱输入信号，或用于弱驱动信号阻抗匹配。

基本特性如下：

- 工作电压范围2.5~5.5V
- 共模输入电压范围rail-to-rail
- 输出电压范围VSSA+100mV~VDDA-100mV
- 典型GBW 10MHz
- 典型压摆率>10V/us
- 最大驱动电流500uA，最大容性负载20pF
- 支持standalone模式，Buffer模式
- 典型输入offset小于 +/-3mv，支持用户校准
- OPA输出可连接ADC，用于输入信号预放大和阻抗匹配
- DAC输出可以连接到OPA作为输出缓冲

35.2 结构框图

下图是OPA的结构框图：

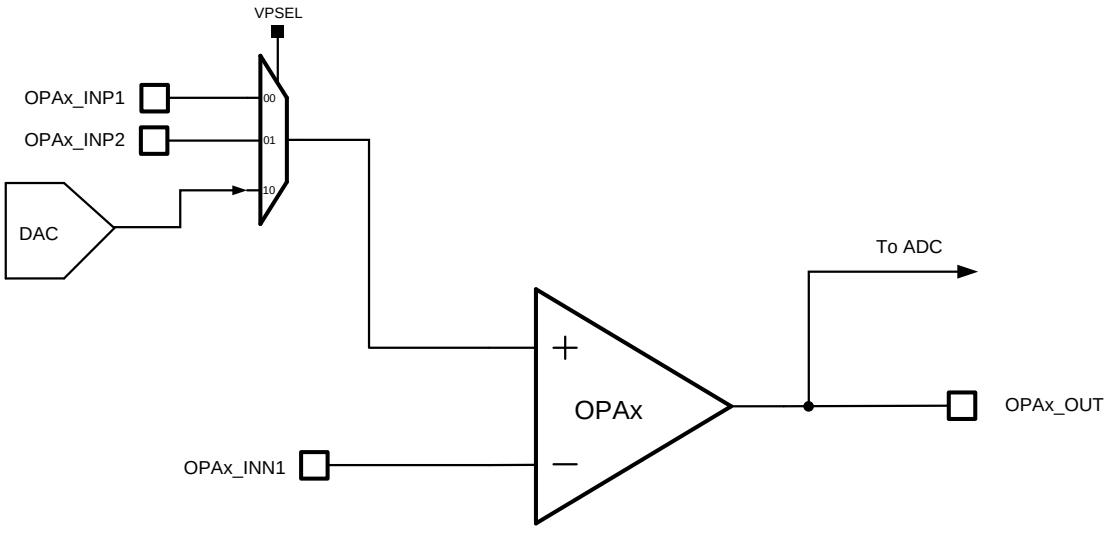


图 35-1 OPA 电路框图

35.3 引脚定义

OPA模块有多个模拟输入输出端口，被复用到多个GPIO上。

引脚	OPAx	符号	功能
PB13	OPA1	OPA1_INN	运放负端输入
PB14		OPA1_INP1	运放正端输入
PB11		OPA1_INP2	
PB12		OPA1_OUT	运放输出
PC0	OPA2	OPA2_INN	运放负端输入
PC1		OPA2_INP1	运放正端输入
PC4		OPA2_INP2	
PC5		OPA2_OUT	运放输出

表 35-1 OPA 引脚列表

35.4 功能描述

35.4.1 时钟和复位

OPA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OPA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。

OPA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OPA的寄存器配置。

35.4.2 Standalone 模式（non-inverting amplifier）

此模式下OPA的输入和输出都直接连接到芯片GPIO的电阻通道上，通过在片外连接反馈电阻，用户可以灵活调节运放的负反馈增益，如下图所示：

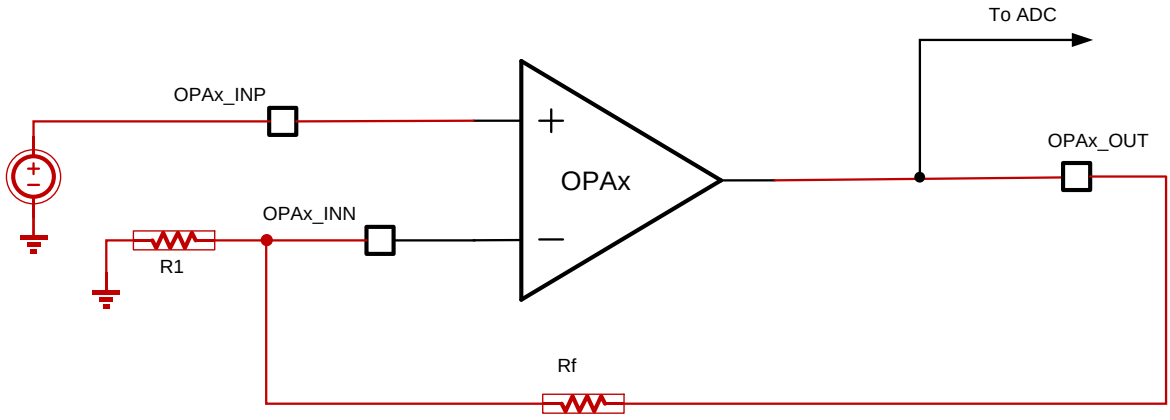


图 35-2 OPA 非反相放大

输出信号Vout由下式定义：

$$V_{out} = (1 + \frac{Rf}{R1}) \times V_{in}$$

35.4.3 Standalone 模式（inverting amplifier）

此模式下OPA的输入和输出都直接连接到芯片GPIO的电阻通道上，通过在片外连接反馈电阻，用户可以灵活调节运放的负反馈增益，如下图所示：

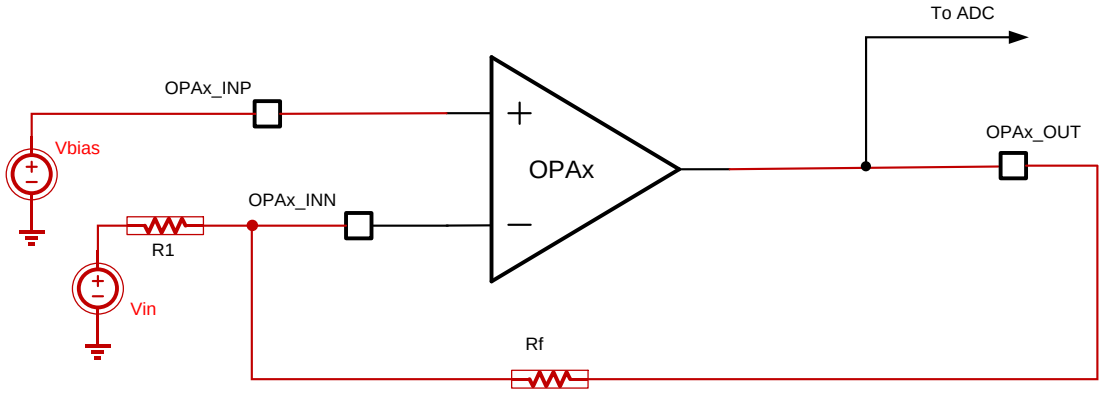


图 35-3 OPA 反相放大

输出信号Vout由下式定义：

$$V_{out} = V_{bias} + (V_{bias} - V_{in}) \times \frac{Rf}{R1}$$

35.4.4 Buffer 模式

buffer模式下OPA可用于为ADC输入提供阻抗调整，当输入信号频率与OPA的GBW相适应时，配置为buffer模式的OPA可以增强ADC输入信号的驱动能力。

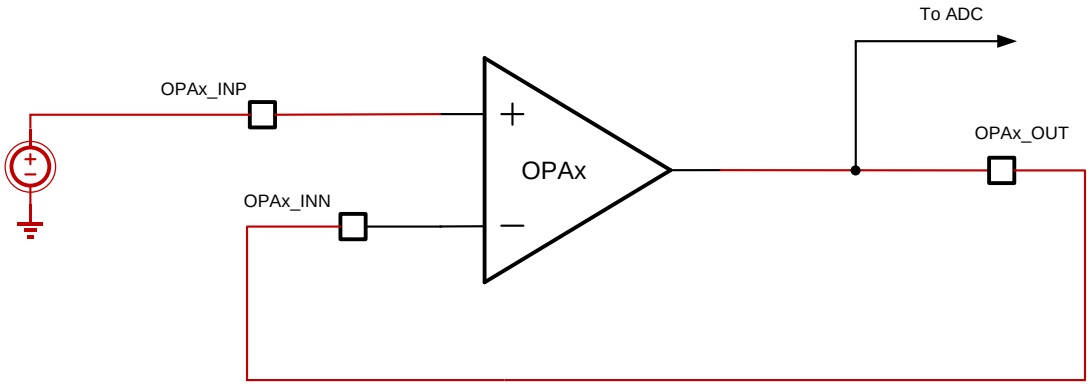


图 35-4 OPA 缓冲器模式

OPA也可以为DAC输出提供缓冲，以增强DAC输出的驱动能力。

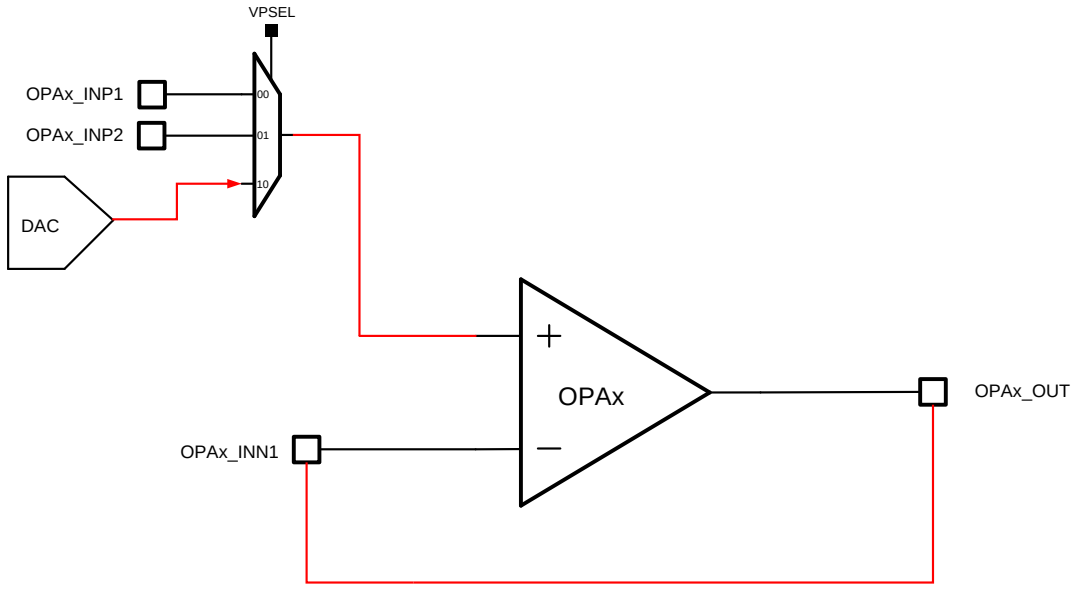


图 35-5 OPA 为 DAC 提供输出缓冲

35.4.5 Offset 校准

校准功能用于抵消运放输入offset电压。为了避免封装和reflow应力影响，在对运放offset要求较高的场合，建议用户在reflow焊接后进行板级校准。校准完全由软件操作实现，软件通过调整输入差分对镜像电流的大小来补偿固有输入offset电压，并通过读取OPA输出来实现校准。

Offset校准支持自动校准和外部校准两种模式，其中自动校准模式下由芯片内部自行产生输入偏置电压，外部校准模式下需要用户提供外部偏置输入。

自动校准模式

用户需按照以下步骤，分别对运放的N输入差分对和P输入差分对进行校准。

N差分输入校准步骤：

- 配置VPSEL=11
- 置位TRIM_MODE
- 置位NCAL_EN
- 置位EN
- 软件读取OUT寄存器

- 如果OUT=1，将NCAL寄存器写为000000并逐步递增，直到OUT从1变成0
- 如果OUT=0，将NCAL寄存器写为100000并逐步递增，直到OUT从0变成1

- 校准完成，并保存NCAL数据

P差分输入校准步骤：

- 配置VPSEL=11
- 置位TRIM_MODE
- 置位PCAL_EN
- 置位EN
- 软件读取OUT寄存器
  - 如果OUT=1，将PCAL寄存器写为000000并逐步递增，直到OUT从1变成0
  - 如果OUT=0，将PCAL寄存器写为100000并逐步递增，直到OUT从0变成1
- 校准完成，并保存PCAL数据

部校准模式

校准原理与自动校准模式类似，仅需清零TRIM_MODE，并在OPA外部输入管脚上提供所需的偏置电压。偏置电压可以根据实际应用中的共模电压选择。

当应用共模电压小于GND+500mV时，将INN和INP端短接并输入共模电平，此时仅校准PCAL。

当应用共模电压大于VDDA-500mV时，将INN和INP端短接并输入共模电平，此时仅校准NCAL。

当应用共模电压介于GND+500mV和VDDA-500mV之间时，将INN和INP端短接并输入共模电平，分两次校准PCAL和NCAL。

35.5 寄存器

模块起始地址：0x4001_5800

offset 地址	名称	符号
0x00	OPA1 控制寄存器 （OPA1 Control Register）	OPA1_CR
0x04	OPA1 校准寄存器 （OPA1 Calibration Register）	OPA1_CALR
0x08	OPA1 校准输出寄存器 （OPA1 Calibration Output Register）	OPA1_COR
0x20	OPA2 控制寄存器 （OPA2 Control Register）	OPA2_CR
0x24	OPA2 校准寄存器 （OPA2 Calibration Register）	OPA2_CALR
0x28	OPA2 校准输出寄存器 （OPA2 Calibration Output Register）	OPA2_COR

35.5.1 OPAx 控制寄存器（OPAx_CR）

名称	OPAx_CR (x=1,2)							
Offset	0x00, 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	-						VPSEL	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9:8	VPSEL	OPA 正端输入选择 (OPA Positive Input Select) 00：OPA_INP1 01：OPA_INP2 10：DAC 11：RFU
7:1	-	RFU：未实现，读为 0
0	EN	OPA 使能寄存器 (OPA enable) 0：关闭 OPA 1：使能 OPA

35.5.2 OPAx 校准寄存器（OPAx_CALR）

名称	OPA1_CALR (x=1,2)							
Offset	0x04, 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	NCAL							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PCAL							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					NCAL_EN	PCAL_EN	TRIM_MODE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:16	NCAL	OPA1 负输入端校准 trim 信号，最高位为符号位 (OPA1 negative input calibration) 01111111：输出电压减小最大 00000001：输出电压减小最小 00000000：输出电压不变 10000000：输出电压不变 10000001：输出电压增加最小 11111111：输出电压增加最大
15:8	PCAL	OPA1 正输入端校准 trim 信号，最高位为符号位 (OPA1 positive input calibration) 01111111：输出电压减小最大 00000001：输出电压减小最小 00000000：输出电压不变 10000000：输出电压不变 10000001：输出电压增加最小 11111111：输出电压增加最大
7:3	-	RFU：未实现，读为 0
2	NCAL_EN	N 差分对校准使能 0：关闭 N 校准 1：使能 N 校准
1	PCAL_EN	P 差分对校准使能 0：关闭 P 校准 1：使能 P 校准
0	TRIM_MODE	校准模式选择 0：外部校准模式 1：自动校准模式

35.5.3 OPAx 校准输出寄存器（OPAx_COR）

名称	OPAx_COR (x=1,2)

Offset	0x08, 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	-							OUT
位权限	U-0							R-x

位号	助记符	功能描述
31:1	-	RFU：未实现，读为 0
0	OUT	校准模式下 OPA 的输出。软件通过轮询这个信号来判断 offset 校准是否成功。

36 数模转换器（DAC）

36.1 概述

本芯片集成1个10位电压输出型数模转换器。DAC使用VREFP引脚电压作为参考基准，基准电压范围0~VDD。

DAC没有输出驱动buffer，但是可以通过OPA缓冲输出以提供驱动能力。

DAC可以为高速比较器提供比较基准。

DAC的主要特性如下：

- 工作电压 1.8～5.5V
- 输出信号幅度 0~VREFP
- 最高输出转换率 1Msps
- 支持 DMA
- DAC 输出可以连接到比较器输入

36.2 结构框图

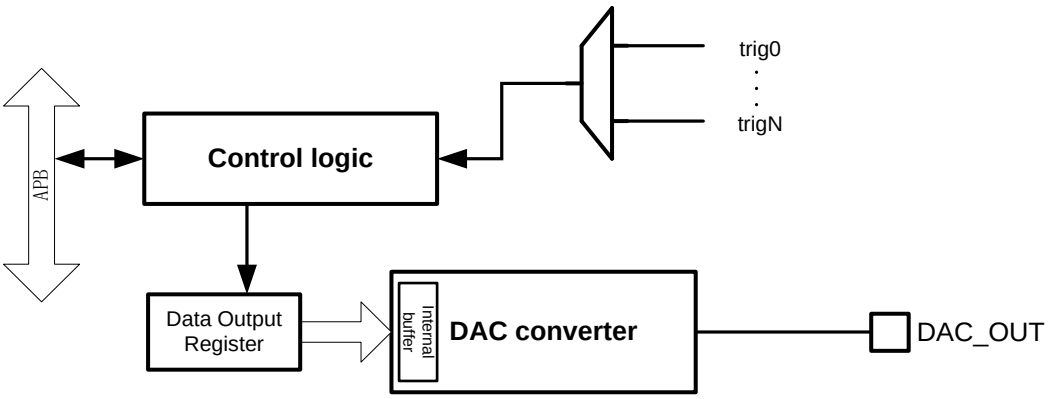


图 36-1 DAC 结构框图

36.3 引脚定义

DAC输出引脚与普通GPIO复用，使用DAC输出功能前需要将相关引脚配置为模拟功能。

引脚	类型	说明
VREFP	输入基准电压	DAC 工作基准，等于或低于 VDDA
VDDA	模拟电源	DAC 工作电源
VSSA	模拟地	DAC 的参考地

引脚	类型	说明
DAC_OUT	模拟输出	DAC 输出信号

表 36-1 DAC 相关引脚

36.4 功能描述

36.4.1 工作时钟与信号时序

DAC转换器使用APB时钟工作，数字码更新频率最高不能超过1MHz，否则无法保证良好的输出动态性能。DAC工作时钟来自于CMU模块，使用DAC之前必须正确配置时钟。

当DAC使能后，DAC有一个启动时间，经过t_{WAKEUP}时间之后，DAC输出电压才会开始建立，t_{WAKEUP}的具体指标参见3电参数。

软件或DMA将需要输出的数字码写入DHR寄存器。在连续输出模式下，DHR寄存器中的数字码将直接复制到DAC内部缓存中，并驱动DAC输出电平。在触发输出模式下，DHR寄存器中的数字码等待触发事件到来后，再更新到DAC内部缓存中。一旦数字码更新完成，DAC工作不再需要时钟。

DAC工作时序示意图如下：

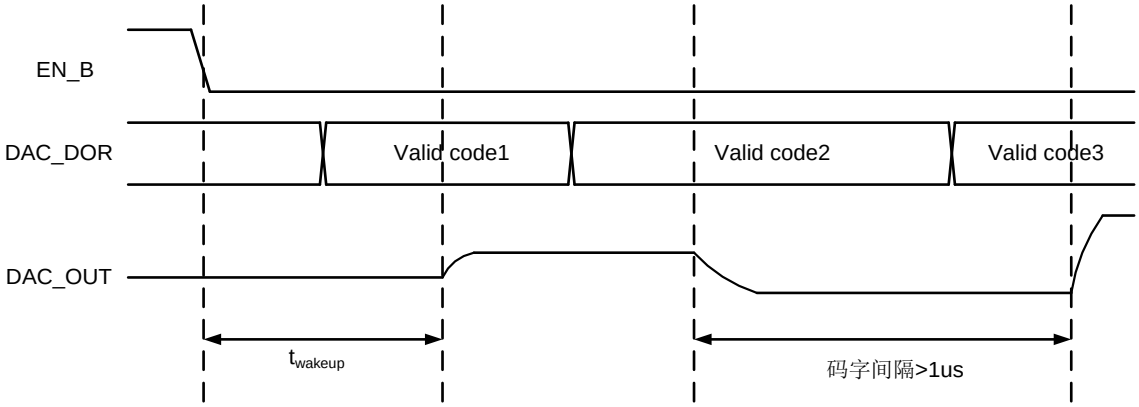


图 36-2 DAC 工作时序

36.4.2 DAC 输出模式

DAC输出可以采用触发方式或连续方式。

在TRGEN（Trigger Enable）为1的情况下，当trigger信号到来时，DAC_DHR寄存器中的数据被传输到DAC内部缓存中，DAC才会更新DAC_OUTx输出电压。

如果TRGEN=0，则软件对DAC_DHR寄存器的写操作将同时被映射到DAC内部缓存中，即DAC_OUTx的输出电平将被实时改变。

下图为DAC连续输出模式。在连续模式下，软件更新DAC_DHR寄存器的频率不能高于1Mhz，否则

输出无法保证正确建立。

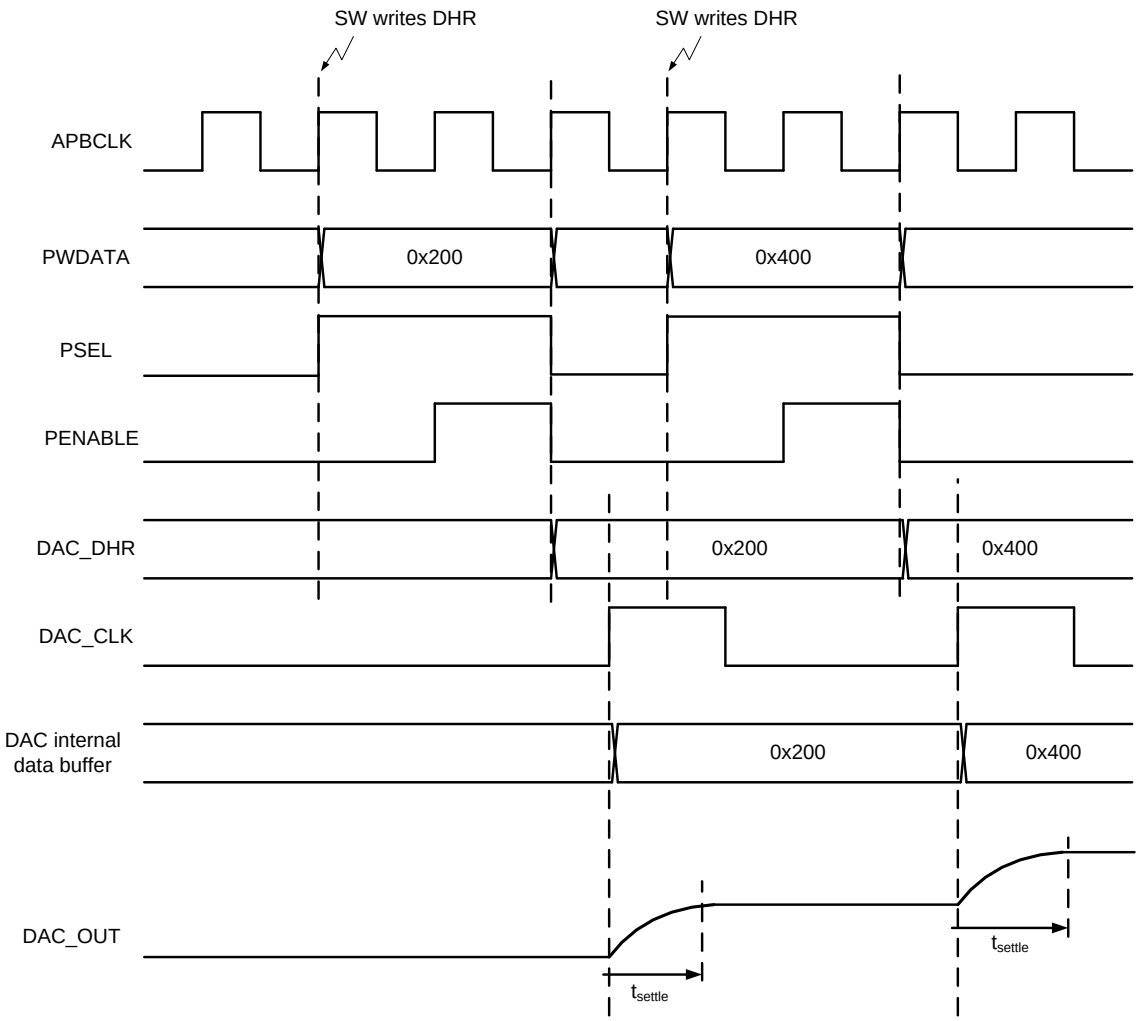


图 36-3 DAC 连续输出模式

下图为DAC触发输出模式。触发输出模式下，触发事件到来的频率不能超过1Mhz，否则输出无法保证正确建立。

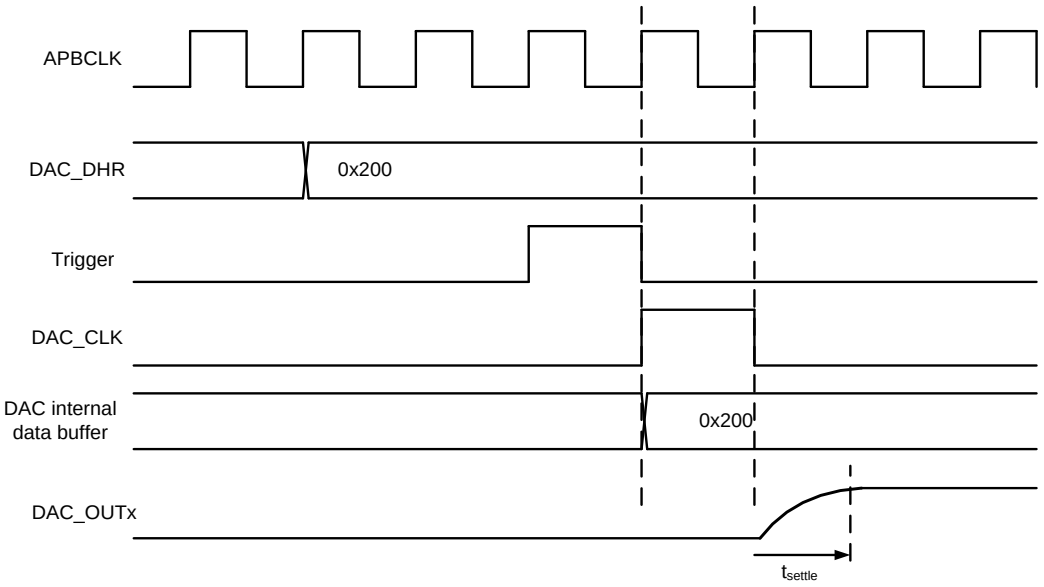


图 36-4 DAC 触发输出模式

DAC内部数据buffer被更新后，到DAC_OUTx输出电压稳定，有一个settling time，这个时间与电源电压、输出负载和DAC输出驱动能力有关。

36.4.3 DAC 触发源选择

DAC触发模式支持硬件触发或者软件触发。

通过TRGSEL寄存器，可以选择DAC使用的触发源，默认使用软件触发，软件可以通过对SWTRG寄存器写1实现DAC触发更新。

当选择其他触发信号源时，DAC将检测输入触发信号的上升沿，并将DAC_DHR中的数据更新到DAC_DOR寄存器中。

下表定义了TRGSEL不同配置对应的触发源。

触发源	类型	TRGSEL[3:0]
SWTRG	软件触发	0000
ATIM_TRGO	内部定时器触发信号	0001
RFU		0010
GPTIM0_TRGO	内部定时器触发信号	0011
GPTIM1_TRGO		0100
RFU		0101
BSTIM_TRGO	内部定时器触发信号	0110
LPTIM_TRGO		0111
RFU		1000
		1001
EXTI	引脚中断触发	1010

表 36-2 DAC 触发源

36.4.4 DAC 输出电压

DAC输出电压是在0~VREF+范围内对输入数字码的线性转换。

DAC输出电压理论值由下式决定：

$$DAC_OUT = VERF \times \frac{DAC_DOR}{1023}$$

36.4.5 DAC 输出缓冲

DAC本身输出无驱动能力，不能驱动阻性负载。在需要驱动能力的场合下，可以使用OPA2作为DAC输出缓冲，此时需要将OPA2连接为buffer模式，详情参见OPA章节。

注意，由于OPA本身限制，经过缓冲后的DAC输出不能实现轨到轨输出，具体输出范围和驱动能力参见电参数章节中的OPA部分。

36.4.6 DMA

通过DMA可以实现自动定时刷新DAC输出数据，DMA传输方向为存储器（RAM或者Flash）->DAC.DHR。比如DMA通道配置源地址为RAM，目标地址为DAC.DHR，通道触发信号为定时器的TRGO，DMA带宽为half-word，每次定时器产生TRGO时，触发DMA自动从RAM中搬运16bit数据到DHR，由于DHR位宽为10bit，高6bit自动丢弃。

在DMA配合下，可以很方便的实现定时自动刷新DAC输出。

36.5 寄存器

模块起始地址：0x40016000

offset 地址	名称	符号
0x00	DAC 控制寄存器 1 （DAC Control Register）	DAC_CR1
0x04	DAC 控制寄存器 2 （DAC Control Register）	DAC_CR2
0x08	DAC 配置寄存器 （DAC Config Register）	DAC_CFGR
0x0C	DAC 软件触发寄存器 （DAC Software Trigger Register）	DAC_SWTRGR
0x10	DAC 数据保持寄存器 （DAC Data Holding Register）	DAC_DHR
0x14	DAC 状态标志寄存器 （DAC Interrupt Status Register）	DAC_ISR
0x18	DAC 中断使能寄存器 （DAC Interrupt Enable Register）	DAC_IER
0x1C		

36.5.1 DAC 控制寄存器 1（DAC_CR1）

名称	DAC_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	-							
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	DAC 使能（enable） 0：关闭 DAC 1：使能 DAC

36.5.2 DAC 控制寄存器 2（DAC_CR2）

名称	DAC_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	-						DMAEN	TRGEN
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	DMAEN	DMA 使能 (DMA enable) 0：禁止 DMA 功能 1：使能 DMA 功能
0	TRGEN	触发模式使能 (Trigger enable) 0：禁止触发模式 1：使能触发模式

36.5.3 DAC 配置寄存器（DAC_CFGR）

名称	DAC_CFGR							
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	-		TRGSEL				-	SWIEN
位权限	U-0		R/W-0000				U-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:2	TRGSEL	触发源选择 (Trigger select)，必须在 TRGEN 为 0 的情况下修改触发源选择 参见 36.4.3DAC 触发源选择

Bit	助记符	功能描述
1	--	RFU：未实现，读为 0
0	SWIEN	DAC 反馈开关使能 (Switch Enable) 0：断开开关 1：使能开关

36.5.4 DAC 软件触发寄存器（DAC_SWTRGR）

名称	DAC_SWTRGR							
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	-							SWTRIG
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	SWTRIG	DAC 软件触发寄存器 (Software Trigger)，软件置位，当 DAC_DHR 中数据载入 DAC_DOR 寄存器后硬件自动清零。

36.5.5 DAC 数据保持寄存器（DAC_DHR）

名称	DAC_DHR							
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	-						DHR	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DHR							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0

Bit	助记符	功能描述
9:0	DHR	DAC 数据保持寄存器 (Data Holding Register)

36.5.6 DAC 状态标志寄存器（DAC_ISR）

名称	DAC_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	-				DMAERR	-		DOU
位权限	U-0				R/W-0	U-0		R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	DMAERR	DMA 错误标志 (DMA Error) 仅 DMAEN=1 时有效，软件写 1 清零
2:1	--	RFU：未实现，读为 0
0	DOU	数据输出寄存器更新标志 (Data Output Updated) 当 DHR 寄存器内容被写入 DOR 时，此标志置位，软件写 1 清零。 仅在触发方式下（TRGEN=1）有效。

36.5.7 DAC 中断使能寄存器（DAC_IER）

名称	DAC_IER							
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	-				DMAE_IE	-		DOU_IE

位权限	U-0	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3	DMAE_IE	DMA 错误中断使能 (DMA Error interrupt enable) 1：允许 DMAERR 中断 0：禁止 DMAERR 中断
2:1	--	RFU：未实现，读为 0
0	DOU_IE	数据输出寄存器更新标志 (Data Output Updated Interrupt Enable) 1：允许 DOU 中断 0：禁止 DOU 中断

37 可编程胶合逻辑（PGL）

37.1 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

37.2 结构框图

LUT的结构如下图所示。

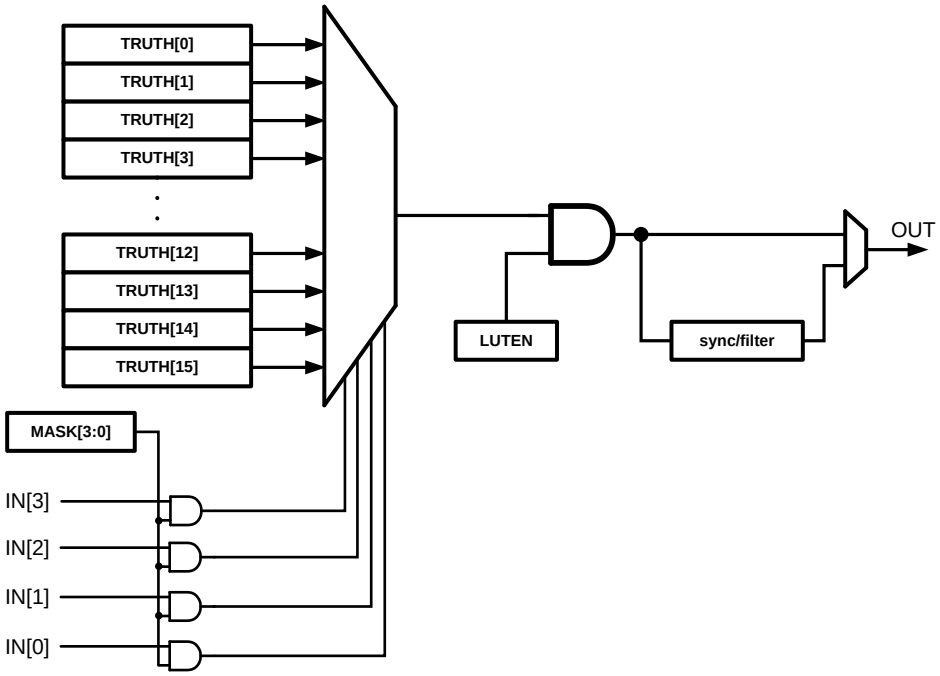


图 37-1 LUT 结构框图

PGL模块共包含了4个LUT。

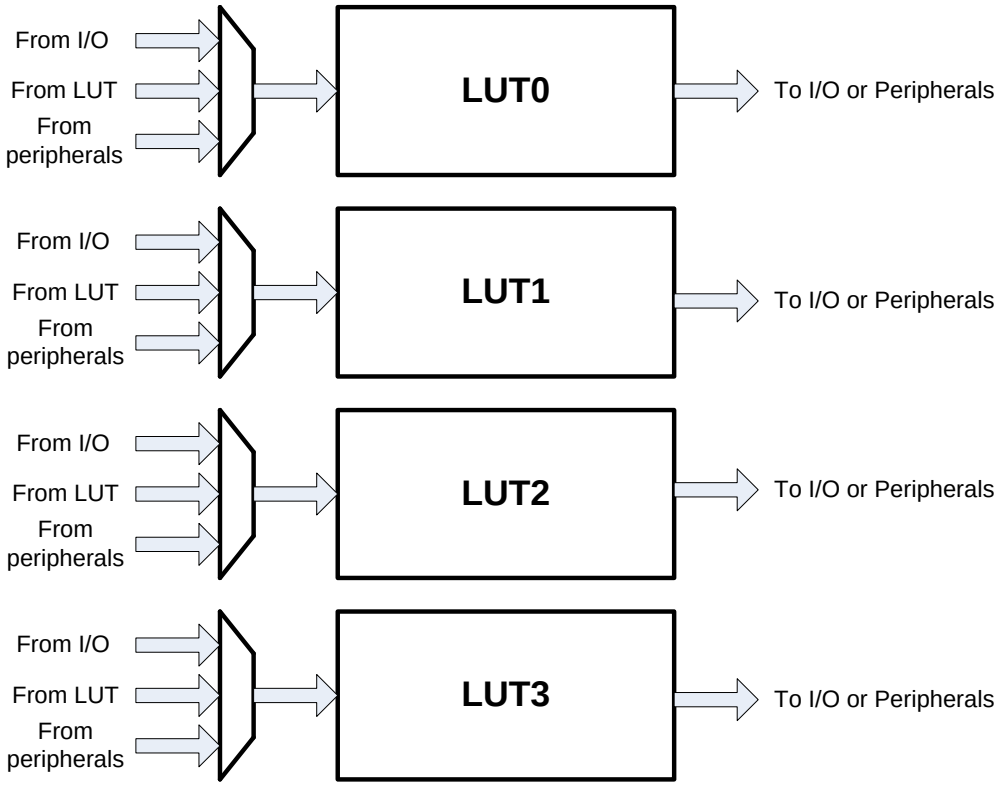


图 37-2 PGL 结构框图

37.3 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~PC5	GPIO 输入	LUT0 引脚输入通道
PC6~PC9	GPIO 输入	LUT1 引脚输入通道
PB7~PB10	GPIO 输入	LUT2 引脚输入通道
PB11~PB14	GPIO 输入	LUT3 引脚输入通道
PD2	额外数字功能	LUT0 输出
PD3		LUT1 输出
PD4		LUT2 输出
PD5		LUT3 输出

表 37-1 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。

37.4 功能描述

37.4.1 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

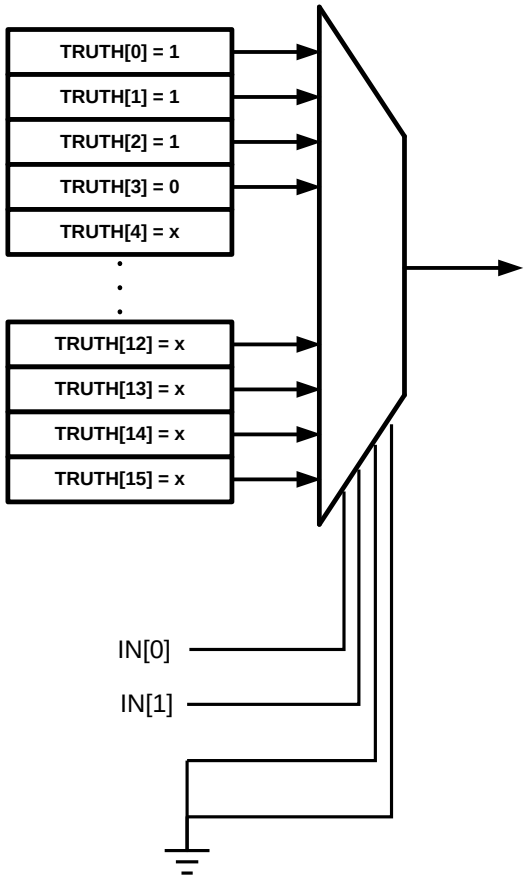


图 37-3 LUT 实现 2 输入 NAND 示意图

37.4.2 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。

37.4.3 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	PD2	LUT1,LUT2,LUT3
LUT1 输出	PD3	LUT2,LUT3
LUT2 输出	PD4	LUT3
LUT3 输出	PD5	-

表 37-2 LUT 输出连接

37.4.4 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。

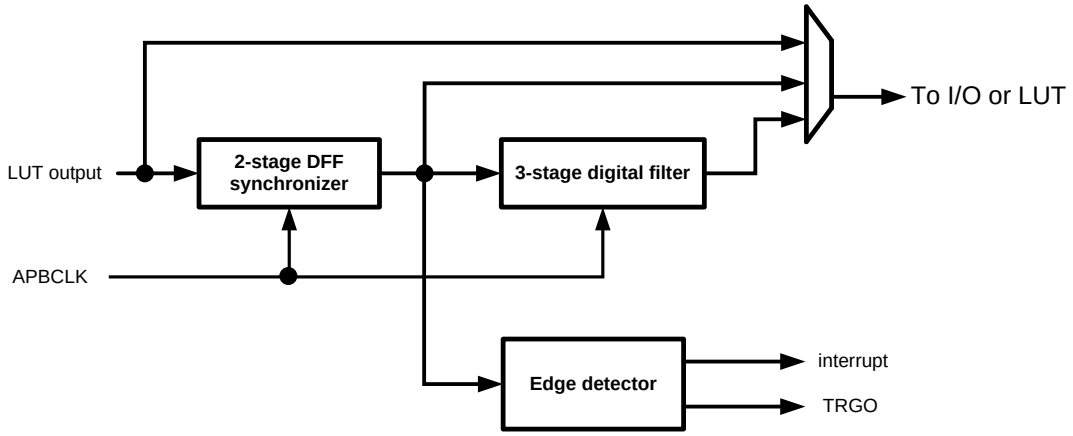


图 37-4 输出滤波和采样

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。

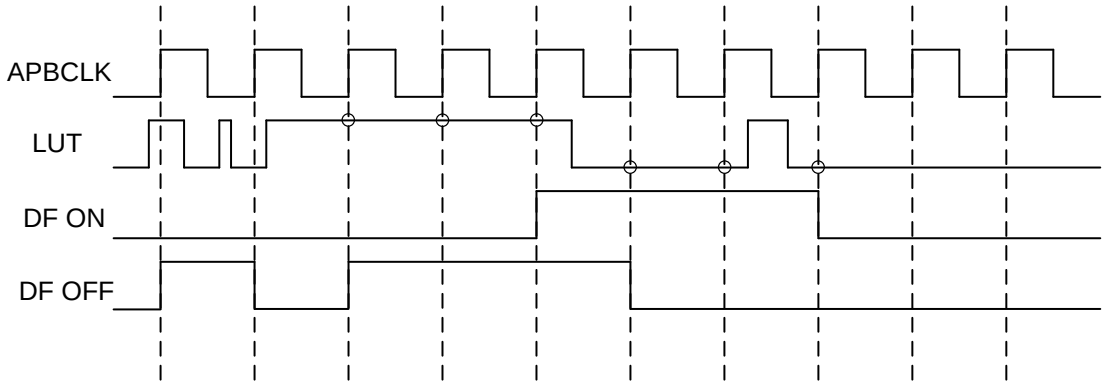


图 37-5 数字滤波

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。

应用上可以使用PGL对外部引脚输入信号进行数字滤波后，再提供给通信外设输入。

比如，使用LUT0给UART0输入做数字滤波：

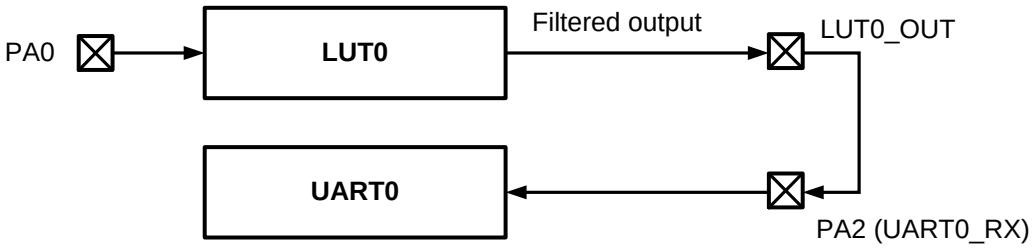


图 37-6 使用 LUT0 为 UART 输入做数字滤波

37.4.5 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。

37.4.6 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。

37.5 寄存器

模块起始地址：0x4001_6C00

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

37.5.1 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			
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]

37.5.2 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			
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：PC5 01：LPTIM16.OC1REF 10：BSTIM16.OCREF 11：ATIM.OC1REF
13:12	IN2SEL	LUT0 输入 2 选择寄存器 00：PC4 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT0 输入 1 选择寄存器 00：PC3 01：RFU 10：COMP2_OUT 11：RFU
9:8	IN0SEL	LUT0 输入 0 选择寄存器 00：PC2 01：RFU 10：COMP1_OUT 11：RFU
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

37.5.3 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			
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：PC9 01：LPTIM16.OC1REF 10：BTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT1 输入 2 选择寄存器 00：PC8 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT1 输入 1 选择寄存器 00：PC7 01：RFU 10：COMP2_OUT 11：RFU
9:8	IN0SEL	LUT1 输入 0 选择寄存器 00：PC6 01：LUT0 output 10：COMP1_OUT 11：RFU
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

37.5.4 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			
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：PB10 01：LPTIM16.OC1REF 10：BTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT2 输入 2 选择寄存器 00：PB9 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT2 输入 1 选择寄存器 00：PB8 01：LUT1 output 10：COMP2_OUT

Bit	助记符	功能描述
		11：RFU
9:8	IN0SEL	LUT2 输入 0 选择寄存器 00：PB7 01：LUT0 output 10：COMP1_OUT 11：RFU
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

37.5.5 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			
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：PB14 01：LPTIM16.OC1REF 10：BTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	LUT3 输入 2 选择寄存器 00：PB13

Bit	助记符	功能描述
		01：RFU 10：RFU 11：RFU
11:10	IN1SEL	LUT3 输入 1 选择寄存器 00：PB12 01：LUT1 output 10：COMP2_OUT 11：RFU
9:8	IN0SEL	LUT3 输入 0 选择寄存器 00：PB11 01：LUT0 output 10：COMP1_OUT 11：RFU
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

37.5.6 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			
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

37.5.7 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			
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

37.5.8 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

37.5.9 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

37.5.10 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

37.5.11 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

38 I/O 端口（GPIO）

38.1 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)

38.2 引脚类型

FM33LF0xxA 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；

38.2.1 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

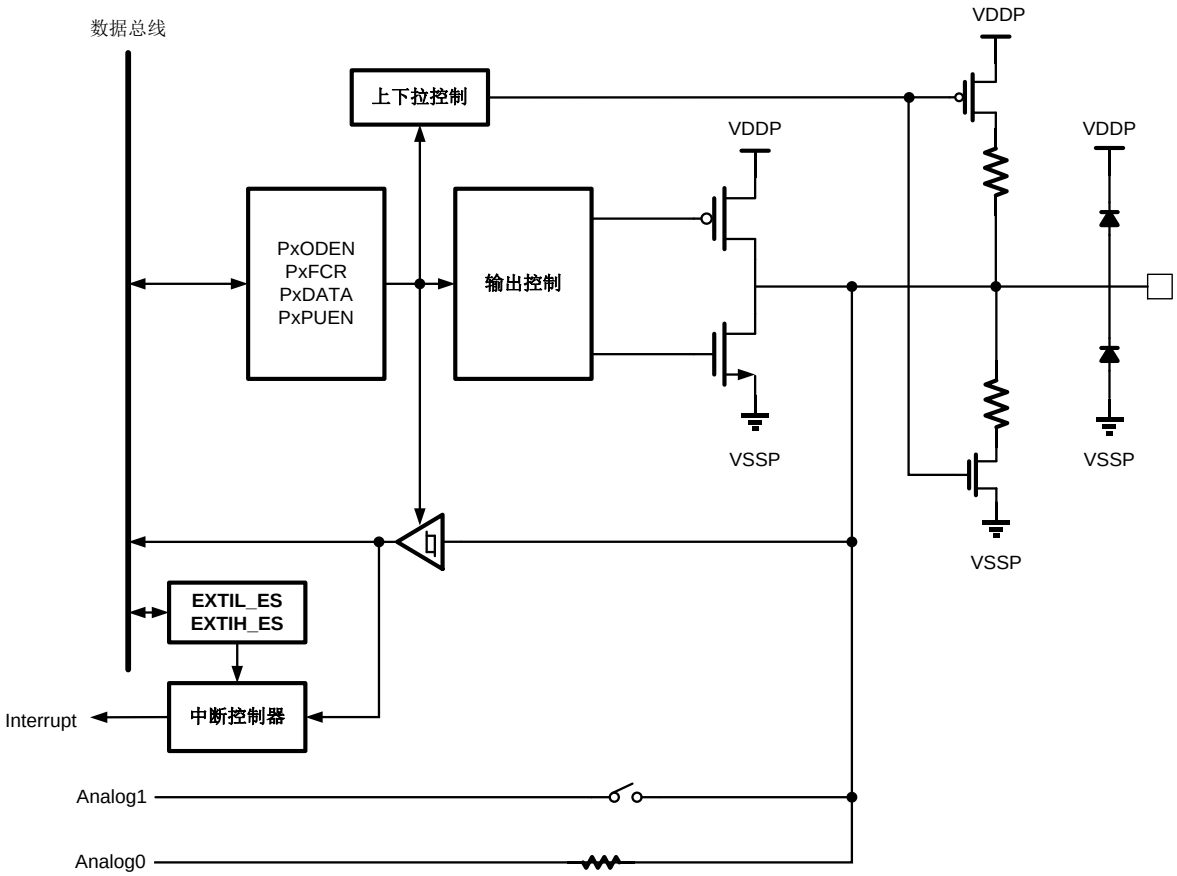


图 38-1 普通 GPIO 结构框图

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	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	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 38-1GPIO 功能逻辑定义表

上下拉电阻各50K欧姆

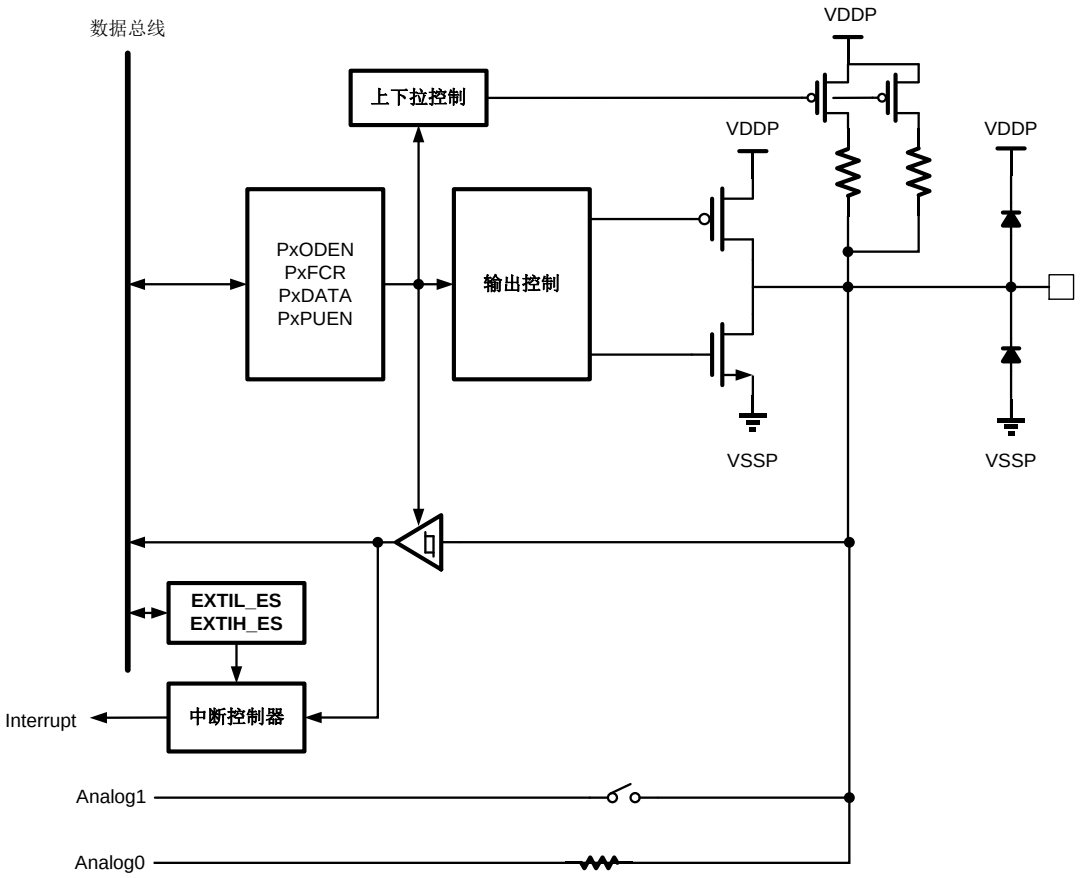
4级驱动能力：

X1 – 4mA

X2 – 8mA

X3 – 12mA

38.2.2 PC14(NRST)，仅上拉



作为NRST时，固定10K上拉电阻，不可控

作为GPIO时，支持50K上拉，不支持下拉电阻，不支持可配置驱动能力

38.3 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 38-2 FCR 定义表

38.3.1 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
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 下拉电阻由 PxPDEN 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

38.3.2 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
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 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 2 级驱动能力可配置，解决驱动能力和 EMI 之间的平衡

38.3.3 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
- 上拉电阻由 PxPUEN 寄存器控制使能或关闭
- 软件读取 PxDIN 寄存器能够获得 IO 上的电平状态
- 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 AFSELx 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。

38.3.4 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

每个 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 ANAEN 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 DISP 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	模块内部采用T型开关隔离crosstalk
OPAx_INN/INP	电阻	
OPAx_OUT	电阻	减小输出内阻
ANATST	开关	FCR=11 时使能开关
VCINx	开关	FCR=11 时使能开关
VDISPx	开关	FCR=11 并且 ANEN=1 时使能开关

表 38-3PAD 通道配置

38.3.5 使用外部晶体引脚

FM33LF0xxA 支持外接 32768Hz 晶体和 4~32MHz 高频晶体。

其中 PC2 和 PC3 默认为 GPIO，配置为模拟功能之后作为 XTHFIN 和 XTHFOUT 外接高频晶体。

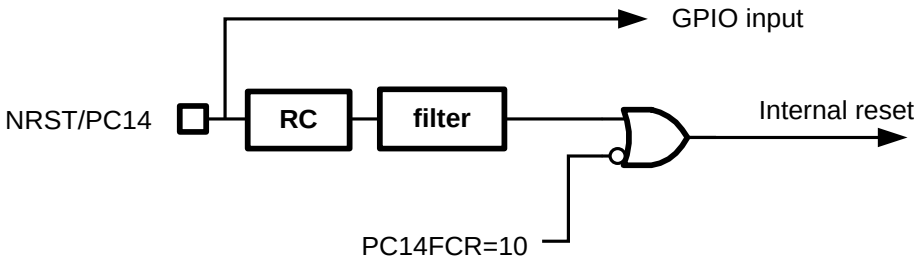
XT32KI 和 XT32KO 为模拟功能，外接 32768Hz 晶体，无法配置为 GPIO 使用。

如果要使用外部 32K 时钟输入，可从 XT32KI 灌入 32K 时钟。

38.4 NRST 引脚与 GPIO 复用

为了提高管脚利用率，NRST 引脚也可以复用为 GPIO。

PC14FCR 寄存器复位值为 10，即复位默认该引脚为 NRST 功能，并且 PCPUEN14 复位值为 1，即默认上拉使能。软件可以通过将 PC14FCR 修改为 00 或 01 来使能其 GPIO 功能，此时 NRST 复位功能无效。



NRST 引脚有数字滤波，滤波采用 RCLP 时钟工作。

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。

38.5 SWD 引脚

ARM SWD 引脚复用 PD7(SWCLK)和 PD8(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，以省去外部上拉。

38.6 WKUPx 引脚

FM33LF0xxA 有 9 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。

9 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

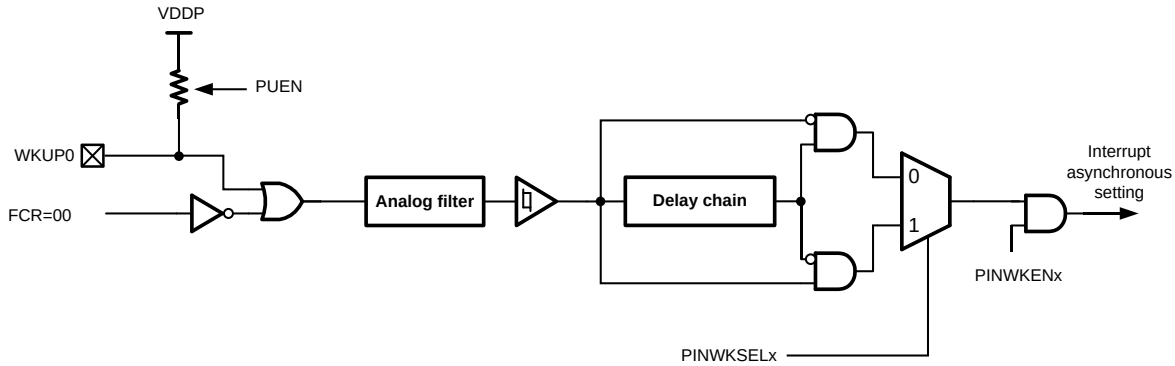


图 38-2 WKUPx 功能结构框图

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。

38.7 外部引脚中断（EXTI）

38.7.1 功能说明

FM33LF0xxA 的 4 组 GPIO（A~D）最多可以产生 16 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]

EXTI_xSEL 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。

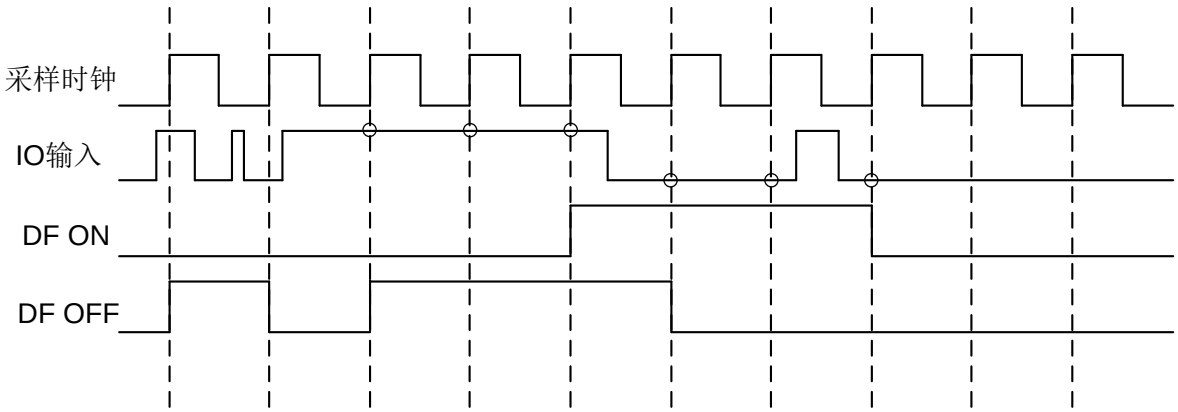


图 38-3 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

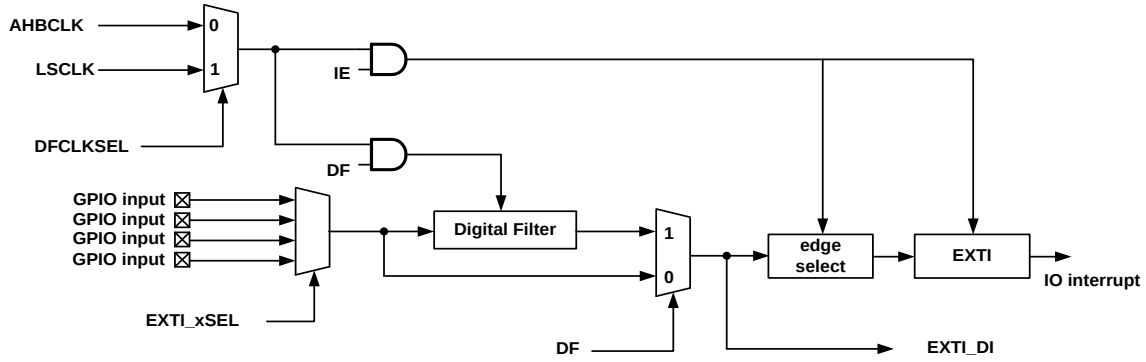


图 38-4 EXTI 信号输入示意图

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EXTI_EDS寄存器配置。

38.7.2 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 SYSCLKSEL 寄存器（0x0x4000020C）的 EXTICKSEL 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 EXTI_SEL 寄存器选择对应的 IO
- 置位 OPCCON1.EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 LSCLK 周期
- 配置 EXTI_EDS 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 OPCCON1.EXTICKE，打开 EXTI 工作时钟使能
- 等待至少 4 个 APBCLK 周期
- 配置 EXTI_EDS 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 OPCCON1.EXTICKE，打开 EXTI 采样时钟使能
- 等待至少 4 个 LSCLK 时钟周期
- 配置 EXTI_EDS 触发边沿，使能所需的 EXTI 中断

38.8 快速 GPIO 输出

FM33LF0xxA 可以通过 set-reset 功能快速改变每个 GPIO 的输出数据（bitwise operation），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。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 set 寄存器特定 bit 写 1 可以置位对应的数据寄存器的 bit 位，对 reset 寄存器特定地址写 1 可以清除对应的数据寄存器的 bit 位。

38.9 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-DownEnable 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	-	-
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_PUDEN
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	-	-
0x6C	GPIOB 驱动能力配置寄存器	GPIOB_DSR

offset 地址	名称	符号
	（GPIOB Driver Strength Register）	
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_PUDEN
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	-	-
0xAC	GPIOC 驱动能力配置寄存器 （GPIOC Driver Strength Register）	GPIOC_DSR
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_PUDEN
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	-	-
0xE8	-	-
0xEC	GPIOD 驱动能力配置寄存器 （GPIOD Driver Strength Register）	GPIOD_DSR

offset 地址	名称	符号
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

38.9.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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	PxINEN6	PxINEN5	PxINEN4	PxINEN3	PxINEN2	PxINEN1	PxINEN0
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

Bit	助记符	功能描述
		0：关闭输入使能 1：打开输入使能

38.9.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）

名称	GPIOx_PUDEN (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 0x04 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxPDEN [15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	PxPDEN [7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	PxPUEN [15:8]							
位权限	R/W-0000 0100							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	PxPUEN [7:0]							
位权限	R/W-00011000							

Bit	助记符	功能描述
31:16	PxPDEN	GPIO 下拉使能开关 0: 关闭下拉使能 1: 打开下拉使能
15:0	PxPUEN	GPIO 上拉使能开关 0: 关闭上拉使能 1: 打开上拉使能 *注：PCPUEN[14]复位值为1，默认上拉（NRST）* *注： PDPUEN[8]复位值为1，默认弱上拉（SWIO）*

38.9.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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

38.9.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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 *注：PC14FCR 复位值为 10（NRST 功能）*
27:26	Px13FCR	Px[13]引脚功能选择

Bit	助记符	功能描述
		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
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
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

Bit	助记符	功能描述
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
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

38.9.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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

38.9.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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。

38.9.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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

38.9.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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

38.9.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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 2.1.52.1.62.1.72.1.8

38.9.10 GPIO 驱动强度配置寄存器（GPIOx_DSR）

名称	GPIOx_DSR (x=A~D)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 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）

38.9.11 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 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

Bit	助记符	功能描述
		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] – 00：PA0 01：PA1 10：PA2 11：PA3

38.9.12 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

38.9.13 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	-							
位权限	U-0							
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
31:16	--	RFU：未实现，读为 0
15:0	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

38.9.14 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	-							
位权限	U-0							
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
31:16	--	RFU：未实现，读为 0
15:0	EXTI	外部引脚中断标志寄存器，共可以产生 16 个引脚中断 硬件置位，软件写 1 清零

38.9.15 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 的 16 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。*

38.9.16 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	FOUT1 输出选择 0x0: XTLF 0x1: RCLP 0x2: ADCCLK 0x3: LSCLK 0x4: ADC_EOC 0x5: RTCTM 0x6: PLLO_DIV64 0x7: - 0x8: APBCLK_DIV64 0x9: PLLO 0xA: - 0xB: RCHF 0xC: XTHF_DIV64 0xD: - 0xE: CLK8K 0xF: XTHF
3:0	FOUT0SEL	FOUT0 输出选择 0x0: XTLF 0x1: RCLP 0x2: RCHF_DIV64 0x3: LSCLK 0x4: AHBCLK_DIV64 0x5: RTCTM 0x6: PLLO_DIV64 0x7: RTCCLK64 0x8: APBCLK_DIV64 0x9: PLLO 0xA: ADC_SOC 0xB: RCHF 0xC: XTHF_DIV64 0xD: RCHF 0xE: CLK8K 0xF: -

38.9.17 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-0000 0000						U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN[7:0]							
位权限	R/W-0000 0000							

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 38 号中断 【注】如果需要实现唤醒后不进入 ISR，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。
30:26	--	RFU：未实现，读为 0
25:10	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：RFU 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
9:8	--	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

39 安全功能（SFU）

39.1 概述

为了对应 ISO26262 和 IEC61508 安全标准，FM33LF0A 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 自检
- RAMparity 校验
- RAM 数据写保护
- 关键寄存器写保护
- 非法地址访问检测
- 时钟频率检测
- 晶振停振检测
- 电源电压检测
- IO 输出状态回读
- LIN 自测试
- CAN 自测试
- ADC 自测试
- DAC 自测试
- 温度检测

39.2 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 64KB 闪存的全空间校验。

39.3 RAM 自检

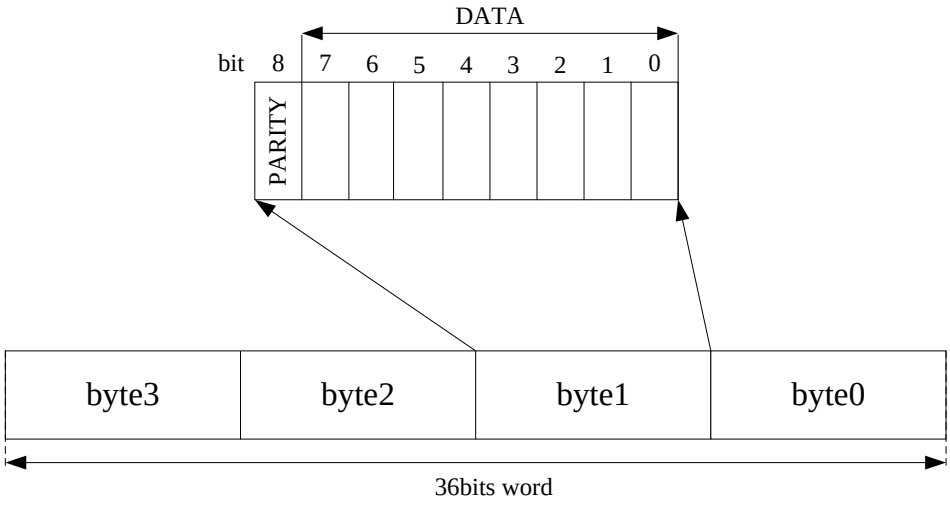
系统 RAM 可以通过 RAMBIST 电路实现自检。

39.4 RAM 奇偶校验

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。

39.5 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。

39.6 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。

支持写保护的模块寄存器：

- 比较器控制寄存器
- SVD 控制寄存器和配置寄存器
- CMU 系统时钟配置寄存器、外设时钟配置寄存器
- RMU 配置寄存器
- RAM 奇偶校验控制寄存器
- GPIO 控制寄存器
- IWDT 配置寄存器和窗口寄存器
- WWDT 配置寄存器和预分频寄存器
- Flash 控制寄存器

参见存储访问保护控制寄存器（SFU_MAP_CR）

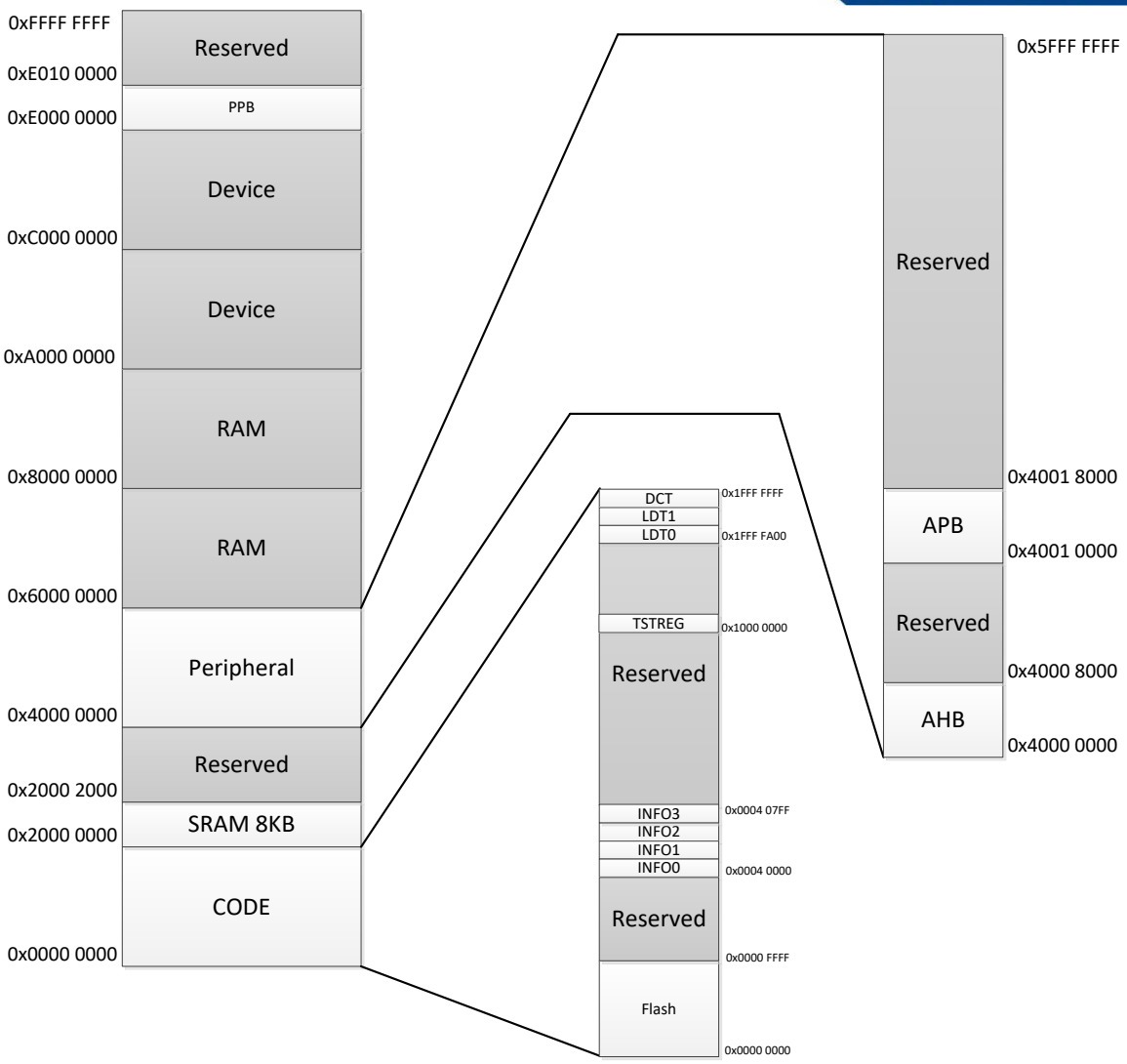
保护控制寄存器位	受保护模块	受保护寄存器
CMPP	模拟比较器	COMP1_CR COMP2_CR COMP3_CR COMP4_CR COMP_ICR COMP_BUFCR
WWP	窗口看门狗	WWDT_CFGR
IWP	独立看门狗	IWDT_CR IWDT_WIN
NVMP	Flash 控制寄存器	FLS_RDCR
RCP	复位管理单元	RMU_PDRCFG RMU_BORCFG
SVDP	电源电压监测	SVD_CFGR SVD_CR SVD_IER SVD_VSR
CCP	时钟管理单元	CMU_SCCR CMU_RCHFCR CMU_RCHTR CMU_PLLCR CMU_RCLPTR CMU_XTLFCR CMU_XTHFCR

		CMU_IER
CLM0P	时钟监控模块 0	CLM0_CR CLM0_CFGR CLM0_CMPH CLM0_CMPL
CLM1P	时钟监控模块 1	CLM1_CR CLM1_CFGR CLM1_CMPH CLM1_CMPL
GPIOAP	GPIOA 组	GPIOA_INEN GPIOA_PUDEN GPIOA_ODEN GPIOA_FCR GPIOA_DFS GPIOA_DSR
GPIOBP	GPIOB 组	GPIOB_INEN GPIOB_PUDEN GPIOB_ODEN GPIOB_FCR GPIOB_DFS GPIOB_DSR
GPIOCP	GPIOC 组	GPIOC_INEN GPIOC_PUDEN GPIOC_ODEN GPIOC_FCR GPIOC_DFS GPIOC_DSR
GPIODP	GPIOD 组	GPIOD_INEN GPIOD_PUDEN GPIOD_ODEN GPIOD_FCR GPIOD_DFS GPIOD_DSR

39.7 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（FM33LF01xA）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0000FFFF	O	O	O
Data Flash	0x00040000	0x000407FF	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 39-164K flash 版本总线访问控制表

地址访问权限表（FM33LF00xA）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x00007FFF	O	O	O
Data Flash	0x00040000	0x000407FF	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 39-2 32K flash 版本总线访问控制表

39.8 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。

CLM 模块用于实现时钟频率互检，参见 CLM 章节。

通过多个定时器也可以实现类似功能。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。

39.9 晶振停振检测

芯片的高频晶振和低频晶振各自带有停振检测电路，当晶振停振时，自动产生报警中断。

同时，如果芯片正在使用晶振作为系统时钟，停振检测电路发现晶体停振时，芯片会自动将系统时钟切换为内部环振，避免系统死机。

39.10 电源电压检测

芯片可以通过 SVD 电路、BOR 电路和 PDR 电路实现多重电源检测和欠压复位保护。

请参见 SVD、RMU 章节。

39.11 IO 输出状态回读

GPIO 在输出信号时（外设输出或者 IO 输出），软件可以从输入数据寄存器回读 IO 状态。

39.12 LIN 自测试

参见 UART 章节中的 LIN 子章节

39.13 CAN 自测试

FSCAN 支持环回测试（loopback），参见 FSCAN 章节。

39.14 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。

39.15 DAC 自测试

DAC 输出可以连接到 ADC 输入（通过片外连接），通过 ADC 采样 DAC 输出，可以判断 DAC 是否工作正常。

39.16 温度检测

芯片内置温度传感器，通过 ADC 采样温度传感器输出，可以得到芯片当前的工作结温，以此大致判断芯片所处的环境温度。

当结温临界，甚至高于芯片允许的工作温度范围时，软件应及时进入安全状态，措施包括但不限于：降低 CPU 工作频率，关闭非必须外设，关闭引脚输出避免抽灌电流，芯片主动停止工作等。

39.17 寄存器

安全功能模块基地址：0x4000_2C00

offset 地址	名称	符号
0x00	RAM 奇偶校验控制寄存器 (RAM Parity Control Register)	SFU_RAMP_CR
0x04	RAM 奇偶校验状态寄存器 (RAM Parity Status Register)	SFU_RAMP_SR
0x08	存储器访问保护控制寄存器 (Memory Access Protection Control Register)	SFU_MAP_CR
0x0C	存储器访问保护状态寄存器 (Memory Access Protection Control Register)	SFU_MAP_SR

39.17.1 RAM 奇偶校验控制寄存器（SFU_RAMP_CR）

名称	SFU_RAMP_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	-					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

39.17.2 RAM 奇偶校验状态寄存器（SFU_RAMP_SR）

名称	SFU_RAMP_SR
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

39.17.3 存储访问保护控制寄存器（SFU_MAP_CR）

名称	SFU_MAP_CR							
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					GPIODP	GPIOCP	GPIOBP	GPIOAP
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	-		CLM1P	CLM0P	CMPP	WWP	IWP	NVMP
位权限	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	MAP_SEL	RCP	SVDP	CCP	-	RAMWP		MAP_EN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-00		R/W-0

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19	GPIODP	GPIO D 组寄存器写保护
18	GPIOCP	GPIO C 组寄存器写保护
17	GPIOBP	GPIO B 组寄存器写保护
16	GPIOAP	GPIO A 组寄存器写保护
15:14	--	RFU：未实现，读为 0
13	CLM1P	CLM1 寄存器写保护

Bit	助记符	功能描述
		1：CLM1 控制寄存器不可改写 0：无保护
12	CLM0P	CLM0 寄存器写保护 1：CLM0 控制寄存器不可改写 0：无保护
11	CMPP	比较器寄存器保护 1：比较器控制寄存器不可改写 0：无保护
10	WWP	WWDT 看门狗寄存器保护 1：WWDT 配置寄存器不可改写 0：无保护
9	IWP	IWDT 看门狗寄存器保护 1：IWDT 配置寄存器不可改写 0：无保护
8	NVMP	Flash 控制寄存器保护 1：Flash 控制寄存器不可改写 0：无保护
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	RCP	RMU 控制寄存器保护(Reset Control Protection) 1：RMU 模块的配置寄存器不可改写 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	--	RFU：未实现，读为 0
2:1	RAMWP	RAM 写保护(RAM Write Protection) 00：无写保护 01：RAM 起始地址开始的 1K 字节禁止写入 10：RAM 起始地址开始的 2K 字节禁止写入 11：RAM 起始地址开始的 4K 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。*

39.17.4 存储访问保护状态寄存器（SFU_MAP_SR）

名称	SFU_MAP_SR							
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

40 专用编程接口

40.1 概述

FM33LF0A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。

40.2 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。

41 调试支持

41.1 概述

FM33LF0A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FM33LF0xxA芯片互联，并实现仿真调试。

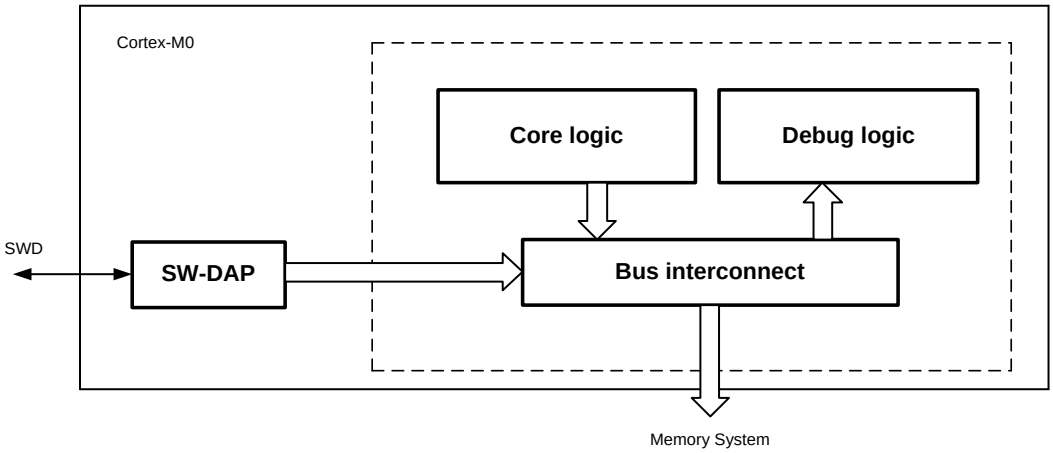


图 41-1 Cortex-M0 调试系统示意图

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。

41.2 Debug 引脚

41.2.1 SWD 引脚

FM33LF0xxA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	PD8
SWCLK	SWD时钟输入	PD7

注意：芯片复位后PD7和PD8都默认为输入状态，与大部分GPIO不同。

41.2.2 上拉电阻

芯片复位后，SWDIO引脚默认使能内部上拉（约100K欧姆），SWCLK引脚默认不使能内部上拉电阻，推荐用户在PCB上外接100K上拉或下拉电阻，以防止输入引脚浮空导致漏电增加。

41.3 SWD 接口协议

41.3.1 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。

41.3.2 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

41.3.3 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。

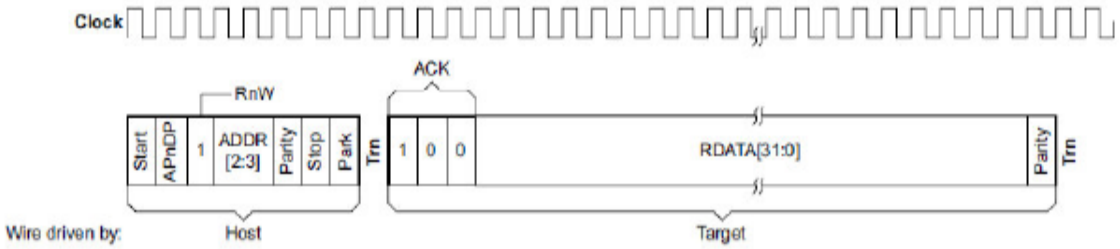
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

41.3.4 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。

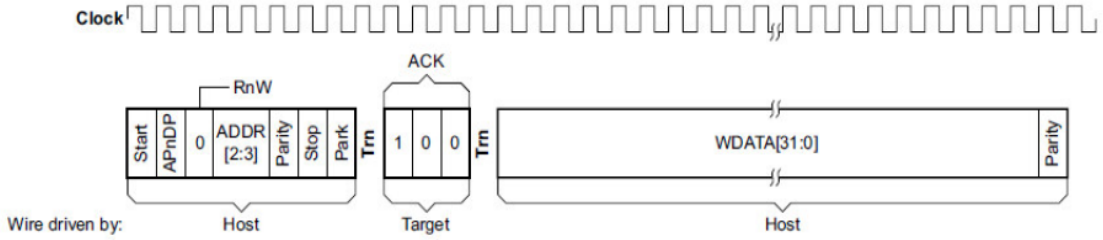


41.3.5 主机写操作

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。



41.4 SWD-DP 寄存器

41.4.1 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.

41.5 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

41.6 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。

41.7 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MCU DEBUG配置寄存器。

41.8 寄存器

地址	名称	符号
0x40000004	MCU Debug 配置寄存器	DBG_CR

41.8.1 MCU DEBUG 配置寄存器

FM33LF0xxA 扩展了 MCUDBGCR 寄存器，用于配置 Debug 状态下的看门狗和定时器。MCUDBGCR 寄存器可以由 SWD 接口或软件改写。

名称	DBGCFG							
地址	0x40000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						DBG_RCEN	DBG_SLEEP
位权限	U-000000						RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT0_STOP	DBG_BT0_STOP	RFU	RFU	DBG_ATIM0_STOP	RFU	DBG_GTIM1_STOP	DBG_GTIM0_STOP
位权限	RW-1	RW-1	U-0	U-0	RW-1	U-0	RW-1	RW-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						DBG_WWDT_STOP	DBG_IWDT_STOP
位权限	U-000000						RW-1	RW-1

位	位名	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

位	位名	功能描述
15	DBG_LPT0_STOP	Debug 状态下 LPTIM0 计数停止控制位 1：Debug 时若内核停止，LPTIM0 停止计数 0：Debug 时保持 LPTIM0 原来状态
14	DBG_BT0_STOP	Debug 状态下 BSTIM0 计数停止控制位 1：Debug 时若内核停止，BSTIM0 停止计数 0：Debug 时保持 BSTIM0 原来状态
13	RFU	未实现，读为 0
12	RFU	未实现，读为 0
11	DBG_ATIM0_STOP	Debug 状态下 ATIM0 计数停止控制位 1：Debug 时若内核停止，ATIM0 停止计数 0：Debug 时保持 ATIM0 原来状态
10	RFU	未实现，读为 0
9	DBG_GTIM1_STOP	Debug 状态下 GTIM1 计数停止控制位 1：Debug 时若内核停止，GTIM1 停止计数 0：Debug 时保持 GTIM1 原来状态
8	DBG_GTIM0_STOP	Debug 状态下 GTIM0 计数停止控制位 1：Debug 时若内核停止，GTIM0 停止计数 0：Debug 时保持 GTIM0 原来状态
7:2	RFU	未实现，读为 0
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

42 器件签名信息

每一颗FM33LF0A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。

42.1 唯一器件 ID

唯一器件ID保存在Flash中，地址0x1FFF_FA10开始的12个字节。

42.2 存储器容量查询

通过查询SYSCON寄存器的bit2，可以获得器件Flash容量信息。

名称	SYSCON							
地址	0x40000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	VERIFAIL	RDPROTFAIL	-					
位权限	R/Dy-0	R/Dy-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	-					FLSCFG	-	
位权限	U-0					R/Dy-00	U-0	

Bit	助记符	功能描述
31	VERIFAIL	Flash 配置信息，即 LDT0 区，校验错误标志 1：某项配置信息校验出现错误，错误项将保持默认值 0：校验正确
30	RDPROTFAIL	OPTBYTE/ACLOCK，即 LDT1 区，校验错误标志 1：校验失败，Debug 接口及 ACLOCK 保护相关寄存器置 1 0：校验通过
29:3	--	RFU：未实现，读为 0
2	FLSCFG	Flash 大小配置 0：64KB 1：32KB （RAM 都是 8KB）
1:0	--	RFU：未实现，读为 0

版本列表

版本号	发布日期	更改说明
1.0	2024.8.15	初版
1.1	2024.11.25	更改QFN32封装图章节笔误，SPI2/1->SPI1/0
1.2	2024.12.15	更新描述错误
1.3	2025.02.10	更新供电3.2.1供电方案描述，增加使用要求
1.4	2025.02.21	HBM性能更新为4KV
1.5	2025.03.26	修正LCD描述
1.6	2025.06.11	修正DBG_CR、FOUT选择寄存器描述

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/