Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33FH0 系列
Touch MCU 芯片

产品说明书

2025.5

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 **21**

图目录 **23**

1 产品综述 **30**

1.1 概述 30
1.2 芯片结构框图 32
1.3 产品型号列表 33

2 引脚和封装 **34**

2.1 封装和引脚排列 34
2.1.1 LQFP100 封装图（FM33FH0x10） 34
2.1.2 LQFP64 封装图（FM33FH0x6） 35
2.1.3 LQFP48 封装图（FM33FH0x5） 36
2.1.4 引脚功能定义（FM33FH0xx） 36
2.1.5 PA 引脚组数字外设功能 45
2.1.6 PB 引脚组数字外设功能 46
2.1.7 PC 引脚组数字外设功能 46
2.1.8 PD 引脚组数字外设功能 47
2.1.9 PE 引脚组数字外设功能 47
2.1.10 PF 引脚组数字外设功能 47
2.1.11 PG 引脚组数字外设功能 48
2.1.12 封装尺寸图 49

3 电参数 **54**

3.1 参数说明 54
3.2 参数测试条件 54
3.2.1 供电方案 54
3.3 极限参数 55
3.4 性能参数 55
3.4.1 典型工作条件 55
3.4.2 功耗参数 56
3.4.3 复位和电源监控 57
3.4.4 内部基准电压 58
3.4.5 低功耗模式唤醒时间 58
3.4.6 外部时钟源特性 59
3.4.7 内部时钟源特性 60
3.4.8 PLL 特性 61
3.4.9 ADC 特性 62
3.4.10 DAC 特性 65
3.4.11 温度传感器 65
3.4.12 模拟比较器特性 67
3.4.13 Flash 存储器特性 67
3.4.14 AC 参数 67
3.4.15 GPIO 特性 71

4 总线与存储 **75**

4.1 系统总线 75
4.2 存储空间分配 76

4.2.1 概述 76
4.2.2 外设模块寄存器地址分配 79
4.3 RAM 81
4.3.1 概述 81
4.3.2 奇偶校验 81
4.4 FLASH 82
4.4.1 概述 82
4.4.2 数据 flash 82
4.4.3 ECC 82
4.4.4 特殊信息扇区说明 83
4.4.5 Flash 编程 86
4.4.6 Flash 的内容保护 90
4.5 寄存器 93
4.5.1 Flash 读取控制寄存器（FLS_RDCR） 93
4.5.2 用户配置字寄存器（FLS_OPTBR） 94
4.5.3 ACLOCK 寄存器 1（FLS_ACLOCK1） 95
4.5.4 ACLOCK 寄存器 2（FLS_ACLOCK2） 96
4.5.5 ACLOCK 寄存器 3（FLS_ACLOCK3） 96
4.5.6 Flash 擦写控制寄存器（FLS_EPCR） 97
4.5.7 Flash Key 输入寄存器（FLS_KEY） 97
4.5.8 Flash 中断使能寄存器（FLS_IER） 98
4.5.9 Flash 标志寄存器（FLS_ISR） 98
4.5.10 Flash ECC 标志寄存器（FLS_ECCSR） 99
4.5.11 Flash ECC 中断使能寄存器（FLS_ECC_IER） 100
4.5.12 Flash ECC 错误地址寄存器（FLS_ECC_EADD） 101
4.5.13 Flash ECC 控制寄存器（FLS_ECCCR） 101
4.5.14 DATA Flash 擦写控制寄存器（DFLS_EPCR） 102
4.5.15 DATA Flash Key 输入寄存器（DFLS_KEY） 102
4.5.16 DATA Flash 中断使能寄存器（DFLS_IER） 103
4.5.17 DATA Flash 标志寄存器（DFLS_ISR） 104

5 电源管理单元（PMU） 106

5.1 芯片工作电源 106
5.1.1 电源域划分 106
5.1.2 电源结构图 107
5.1.3 ADC 基准电压 107
5.1.4 片内快速建立基准源（AVREF） 108
5.2 功耗模式 108
5.2.1 概述 108
5.2.2 功耗模式与系统频率 109
5.2.3 Active 模式 109
5.2.4 LP Run 模式 109
5.2.5 SLEEP 模式 110
5.2.6 DEEPSLEEP 模式 110
5.3 唤醒源 111
5.4 休眠唤醒后的时钟控制 112
5.5 寄存器 112
5.5.1 低功耗控制寄存器（PMU_CR） 112
5.5.2 唤醒时间控制寄存器（PMU_WKTR） 113
5.5.3 唤醒源标志查询寄存器（PMU_WKFR） 114
5.5.4 PMU 中断使能寄存器（PMU_IER） 115
5.5.5 PMU 中断标志寄存器（PMU_ISR） 115
5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR） 116
5.5.7 PTAT 控制寄存器（PMU_PTATCR） 117

6 处理器（CPU）.....118
6.1 概述.....118
6.1.1 处理器配置.....118
6.2 寄存器.....119
6.3 异常和中断.....119
6.3.1 中断向量表.....119
6.3.2 中断优先级.....120
6.3.3 错误处理.....120
6.3.4 锁定（Lockup）.....120
6.4 调试特性.....120
6.4.1 调试功能引脚.....121
6.4.2 调试状态下的看门狗控制.....121
6.4.3 DEBUG 的复位.....121
7 中断向量表（INTMUX）.....122
7.1 概述.....122
7.2 中断向量表.....122
7.3 中断向量多路选择.....123
7.4 寄存器.....124
7.4.1 中断源多路选择控制寄存器 1（INTMUX_CR1）.....124
7.4.2 中断源多路选择控制寄存器 2（INTMUX_CR2）.....125
8 复位管理单元（RMU）.....127
8.1 概述.....127
8.2 模块框图.....128
8.3 上下电复位.....128
8.4 软件复位.....130
8.5 NRST 引脚复位.....130
8.6 LOC 复位.....130
8.7 LOL 复位.....131
8.8 寄存器.....132
8.8.1 PDR 配置寄存器（PDR_CFG）.....133
8.8.2 复位配置寄存器（RMU_RSTCR）.....133
8.8.3 软件复位寄存器（RMU_SOFTRST）.....134
8.8.4 复位标志寄存器（RMU_RSTFR）.....134
8.8.5 AHB 外设复位寄存器.....135
8.8.6 APB 外设复位寄存器 1.....136
8.8.7 APB 外设复位寄存器 2.....138
9 独立看门狗（IWDT）.....140
9.1 概述.....140
9.2 结构框图.....140
9.3 IWDT 功能描述.....140
9.4 IWDT 窗口功能.....141
9.5 IWDT 冻结.....142
9.6 寄存器.....143
9.6.1 IWDT 清除寄存器（IWDT_SERV）.....143
9.6.2 IWDT 配置寄存器（IWDT_CR）.....143
9.6.3 IWDT 计数值寄存器（IWDT_CNT）.....144
9.6.4 IWDT 窗口寄存器（IWDT_WIN）.....145
9.6.5 IWDT 中断使能寄存器（IWDT_IER）.....145
9.6.6 IWDT 中断标志寄存器（IWDT_ISR）.....145
10 窗口看门狗（WWDT）.....147

10.1 功能描述.....147
10.2 结构框图.....147
10.3 WWDT 工作方式.....148
10.4 寄存器.....149
10.4.1 WWDT 控制寄存器（WWDT_CR）.....150
10.4.2 WWDT 配置寄存器（WWDT_CFGR）.....150
10.4.3 WWDT 计数寄存器（WWDT_CNT）.....151
10.4.4 WWDT 中断使能寄存器（WWDT_IER）.....151
10.4.5 WWDT 中断标志寄存器（WWDT_ISR）.....152
10.4.6 WWDT 预分频寄存器（WWDT_PSC）.....152

11 时钟管理单元（CMU）.....154

11.1 概述.....154
11.2 时钟架构.....154
11.2.1 时钟树.....154
11.2.2 SYSCLK 切换说明.....155
11.2.3 时钟安全性.....156
11.2.4 主要时钟说明.....156
11.2.5 外设模块的总线时钟和工作时钟.....157
11.2.6 休眠模式下的外设时钟.....158
11.2.7 LSCLK 切换逻辑.....158
11.2.8 RTC 时钟.....158
11.3 高频晶体振荡电路(XTHF).....159
11.3.1 概述.....159
11.3.2 工作方式.....159
11.3.3 停振检测（HFDET）.....159
11.3.4 XTHF 启动过程.....159
11.4 低频晶体振荡电路(XTLF).....160
11.4.1 概述.....160
11.4.2 工作方式.....160
11.4.3 停振检测（LFDET）.....160
11.4.4 XTLF 启动过程.....160
11.5 低功耗低频环振(RCLP).....161
11.5.1 概述.....161
11.6 锁相环(PLL).....161
11.6.1 概述.....161
11.6.2 锁定检测.....161
11.6.3 应用注意.....161
11.7 低功耗模式下的时钟源.....162
11.8 休眠唤醒的时钟处理.....162
11.9 寄存器.....163
11.9.1 系统时钟配置寄存器（CMU_SYSCLKCR）.....164
11.9.2 RCHF 时钟控制寄存器（CMU_RCHFCR）.....165
11.9.3 RCHF 调校寄存器（CMU_RCHFTR）.....166
11.9.4 PLL 控制寄存器（CMU_PLLCR）.....166
11.9.5 RCLP 控制寄存器（CMU_RCLPCR）.....167
11.9.6 RCLP 调校寄存器（CMU_RCLPTR）.....168
11.9.7 XTLF 控制寄存器（CMU_XTLFCR）.....168
11.9.8 LSCLK 选择寄存器（CMU_LSCLKSEL）.....169
11.9.9 XTHF 控制寄存器（CMU_XTHCR）.....170
11.9.10 CMU 中断使能寄存器（CMU_IER）.....171
11.9.11 CMU 中断标志寄存器（CMU_ISR）.....171
11.9.12 外设总线时钟控制寄存器 1（CMU_PCLKCR1）.....172

11.9.13 外设总线时钟控制寄存器 2（CMU_ PCLKCR2）.....173
11.9.14 外设总线时钟控制寄存器 3（CMU_ PCLKCR3）.....174
11.9.15 外设总线时钟控制寄存器 4（CMU_ PCLKCR4）.....175
11.9.16 外设工作时钟配置寄存器 1（CMU_OPCCR1）.....175
11.9.17 外设工作时钟配置寄存器 2（CMU_OPCCR2）.....177
11.9.18 外设工作时钟使能寄存器（CMU_OPCER1）.....178
11.9.19 AHB Master 控制寄存器（CMU_AMCR）.....179

12 时钟监控（CLM）.....180

12.1 概述.....180
12.2 结构框图.....181
12.3 功能描述.....181
12.3.1 工作模式.....181
12.3.2 使能与关闭.....181
12.3.3 时钟频率监控.....182
12.3.4 超时检测.....185
12.4 寄存器.....186
12.4.1 时钟监控控制寄存器（CLM_CR）.....187
12.4.2 时钟监控配置寄存器（CLM_CFGR）.....187
12.4.3 时钟监控计数值寄存器（CLM_CNTR）.....188
12.4.4 时钟监控中断标志寄存器（CLM_ISR）.....188
12.4.5 时钟监控比较高阈值寄存器（CLM_CMPH）.....189
12.4.6 时钟监控比较低阈值寄存器（CLM_CMPL）.....189
12.4.7 超时检测配置寄存器（CLM_TOCFGR）.....190

13 电源电压监测（SVD）.....191

13.1 概述.....191
13.2 结构框图.....191
13.3 引脚定义.....192
13.4 功能描述.....192
13.4.1 低压报警.....192
13.4.2 间歇使能模式.....194
13.4.3 欠压复位.....194
13.4.4 外部电源检测.....194
13.5 电源检测阈值.....195
13.6 寄存器.....199
13.6.1 SVD 配置寄存器（SVD_CFGR）.....199
13.6.2 SVD 控制寄存器（SVD_CR）.....200
13.6.3 SVD 中断使能寄存器（SVD_IER）.....200
13.6.4 SVD 状态和标志寄存器（SVD_ISR）.....201
13.6.5 SVD 参考电压选择寄存器（SVD_VSR）.....201

14 模拟比较器（COMPARATOR）.....203

14.1 概述.....203
14.2 结构框图.....203
14.3 功能描述.....204
14.3.1 基本功能.....204
14.3.2 内部比较基准选择和产生.....204
14.3.3 时钟和复位.....205
14.3.4 比较器输出滞回.....205
14.3.5 窗口功能.....205
14.3.6 输出数字滤波.....206
14.3.7 功耗和速度模式.....207
14.3.8 比较器中断.....207

14.3.9 比较器输出和触发信号输出 208
14.4 寄存器 209
14.4.1 COMP1 控制寄存器（COMP1_CR） 209
14.4.2 COMP2 控制寄存器（COMP2_CR） 211
14.4.3 COMP 中断配置寄存器（COMP_ICR） 212
14.4.4 COMP 中断标志寄存器（COMP_IF） 213
14.4.5 COMP 缓冲器控制寄存器（COMP_BUFCR） 214
15 双线串行总线主机（I²C MASTER） 215
15.1 概述 215
15.2 结构框图 215
15.3 时钟 216
15.4 接口时序 216
15.4.1 接口时序图 216
15.4.2 接口时序描述 217
15.5 I²C 工作模式 218
15.6 I²C 地址格式 218
15.7 I²C 初始化 219
15.7.1 主机波特率配置 219
15.8 I²C 主机功能 219
15.8.1 7bit 寻址 219
15.8.2 10bit 寻址 223
15.8.3 DMA 225
15.8.4 SCL 延展（Slave Clock Stretching） 229
15.8.5 超时机制 229
15.8.6 可编程时序 229
15.9 寄存器 230
15.9.1 I2C 主机配置寄存器（I2C_MSPCFGR） 231
15.9.2 I2C 主机控制寄存器（I2C_MSPCR） 231
15.9.3 I2C 主机中断使能寄存器（I2C_MSPIER） 232
15.9.4 I2C 主机中断标志寄存器（I2C_MSPISR） 233
15.9.5 I2C 主机状态寄存器（I2C_MSPSR） 234
15.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR） 235
15.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF） 235
15.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR） 236
15.9.9 I2C 主机超时寄存器（I2C_MSPTOR） 236
16 通用异步收发传输器（UART） 238
16.1 概述 238
16.2 结构框图 239
16.3 UART 类型区分 239
16.4 UART 字符描述 240
16.4.1 UART 模式 240
16.5 功能描述 241
16.5.1 时钟结构 241
16.5.2 同步策略 241
16.5.3 位接收采样 242
16.5.4 数据发送 243
16.5.5 数据接收 245
16.5.6 RS232 硬件流控 245
16.5.7 低功耗休眠唤醒（UART0/1） 247
16.5.8 使用 DMA 进行 UART 收发 248
16.5.9 DMA 模式下的发送完成中断 248
16.5.10 半双工单线通信 249

16.6 波特率发生.....249
16.6.1 波特率发生.....249
16.6.1 波特率自适应.....250
16.7 红外调制.....251
16.8 接收超时.....252
16.9 发送延迟.....252
16.10 LIN 总线通信功能.....252
16.10.1 概述.....252
16.10.2 LIN 主机操作.....255
16.10.3 LIN 从机操作.....255
16.10.4 LIN 接收采样与噪声检测.....263
16.10.5 LIN 错误状态.....264
16.10.6 LIN 自测试模式.....267
16.11 寄存器.....269
16.11.1 红外调制寄存器（UART_IRCR）.....270
16.11.2 UARTx 模式寄存器（UARTx_MCR）.....270
16.11.3 UARTx LIN 控制寄存器（UARTx_LINCR）.....271
16.11.4 UARTx LIN 波特率同步寄存器（UARTx_LINBSR）.....272
16.11.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR）.....272
16.11.6 UARTx LIN 发送时序寄存器（UARTx_LINTTR）.....273
16.11.7 UARTx LIN Pre-sync 寄存器（UARTx_LINPSR）.....274
16.11.8 UARTx LIN Break-len 寄存器（UARTx_LINBKR）.....274
16.11.9 UARTx 控制状态寄存器（UARTx_CSR）.....275
16.11.10 UARTx 中断使能寄存器（UARTx_IER）.....276
16.11.11 UARTx 中断标志寄存器（UARTx_ISR）.....278
16.11.12 UARTx 超时和延迟寄存器（UARTx_TODR）.....279
16.11.13 UARTx 接收缓冲寄存器（UARTx_RXBUF）.....279
16.11.14 UARTx 发送缓冲寄存器（UARTx_TXBUF）.....280
16.11.15 UATRx 波特率产生寄存器（UARTx_BGR）.....280
17 通用同异步收发传输器（USART）.....282
17.1 概述.....282
17.2 结构框图.....283
17.3 UART 类型区分.....283
17.4 USART 字符描述.....284
17.4.1 UART 模式.....284
17.4.2 智能卡模式.....284
17.5 功能描述.....285
17.5.1 时钟结构.....285
17.5.2 同步策略.....285
17.5.3 位接收采样.....286
17.5.4 数据发送.....287
17.5.5 数据接收.....290
17.5.6 RS232 硬件流控（new）.....291
17.5.7 低功耗休眠唤醒.....292
17.5.8 使用 DMA 进行 UART 收发.....292
17.5.9 DMA 模式下的发送完成中断.....292
17.5.10 半双工单线通信.....293
17.6 波特率发生.....293
17.6.1 波特率发生.....293
17.6.2 波特率自适应.....295
17.7 红外调制.....295
17.8 接收超时.....296

17.9 发送延迟.....296
17.10 寄存器.....297
17.10.1 红外调制寄存器（UART_IRCR）.....297
17.10.2 USARTx 模式寄存器（USARTx_MCR）.....298
17.10.3 USARTx 控制状态寄存器（USARTx_CSR）.....298
17.10.4 USARTx 中断使能寄存器（USARTx_IER）.....300
17.10.5 USARTx 中断标志寄存器（USARTx_ISR）.....301
17.10.6 USARTx 超时和延迟寄存器（USARTx_TODR）.....302
17.10.7 USARTx 接收缓冲寄存器（USARTx_RXBUF）.....302
17.10.8 USARTx 发送缓冲寄存器（USARTx_TXBUF）.....303
17.10.9 USATRx 波特率产生寄存器（USARTx_BGR）.....303
17.10.10 USARTx 智能卡状态标志寄存器（USARTx_SCISR）.....304
17.10.11 USARTx 智能卡中断使能寄存器（USARTx_SCIER）.....305
17.10.12 USARTx 智能卡帧格式控制寄存器（USARTx_FFCR）.....306
17.10.13 USARTx 智能卡额外保护时间寄存器（USARTx_EGTR）.....307
17.10.14 USARTx 智能卡输出时钟分频寄存器（USARTx_CODR）.....308
17.10.15 USARTx 智能卡控制寄存器（USARTx_SCCR）.....308

18 低功耗 UART（LPUART）.....310

18.1 概述.....310
18.2 结构框图.....311
18.3 工作时钟.....312
18.4 字符描述.....312
18.5 功能描述.....313
18.5.1 位接收采样和发送.....313
18.5.2 接收流程.....314
18.5.3 发送流程.....314
18.5.4 使用 DMA 进行 LPUART 收发.....314
18.5.5 休眠模式下的数据接收唤醒.....314
18.5.6 LPRUN 模式下的数据 DMA 收发.....315
18.5.7 DMA 模式下的发送完成中断.....315
18.6 寄存器.....316
18.6.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）.....316
18.6.2 LPUARTx 中断使能寄存器（LPUARTx_IER）.....318
18.6.3 LPUARTx 中断标志寄存器（LPUARTx_ISR）.....319
18.6.4 LPUARTx 波特率调制寄存器（LPUARTx_BMR）.....319
18.6.5 LPUARTx 接收数据寄存器（LPUARTx_RXBUF）.....320
18.6.6 LPUARTx 发送数据寄存器（LPUARTx_TXBUF）.....321
18.6.7 LPUARTx 数据匹配寄存器（LPUARTx_DMR）.....321

19 串行外设接口（SPI/I2S）.....322

19.1 概述.....322
19.2 SPI 功能描述.....323
19.2.1 SPI 结构框图.....323
19.2.2 引脚定义.....323
19.2.3 SPI 接口时序.....324
19.2.4 I/O 配置.....326
19.2.5 全双工数据通信.....327
19.2.6 TX-ONLY 模式.....329
19.2.7 RX-ONLY 模式.....329
19.2.8 主机 SSN 控制.....329
19.2.9 数据冲突.....330
19.2.10 使用 DMA 进行 SPI 收发.....330

19.3 I2S 功能描述.....332
19.3.1 I2S 结构框图.....332
19.3.2 引脚定义.....332
19.3.3 协议支持.....332
19.3.4 时钟产生.....336
19.3.5 I2S 主机功能.....337
19.3.6 I2S 从机功能.....339
19.3.7 帧格式错误.....340
19.3.8 DMA 支持.....340
19.4 寄存器.....341
19.4.1 SPI 控制寄存器 1（SPIx_CR1）.....341
19.4.2 SPI 控制寄存器 2（SPIx_CR2）.....342
19.4.3 SPI 控制寄存器 3（SPIx_CR3）.....344
19.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）.....345
19.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）.....345
19.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）.....347
19.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）.....347
19.4.8 I2S 配置寄存器（SPIx_I2SCR）.....348
19.4.9 I2S 预分频寄存器（SPIx_I2SPR）.....349

20 QUADSPI.....350

20.1 概述.....350
20.2 结构框图.....350
20.3 功能描述.....351
20.3.1 工作时钟.....351
20.3.2 工作模式.....351
20.3.3 总线配置.....356
20.3.4 帧格式.....358
20.3.5 延迟数据采样.....360
20.3.6 BUSY 位和 ABORT 功能.....361
20.3.7 nCS 控制.....361
20.4 高效的 XIP 实现.....362
20.4.1 应用举例.....363
20.5 寄存器.....365
20.5.1 QuadSPI 控制寄存器.....365
20.5.2 QuadSPI 配置寄存器.....367
20.5.3 QuadSPI 状态寄存器.....368
20.5.4 QuadSPI 数据长度寄存器.....368
20.5.5 QuadSPI 通信配置寄存器.....369
20.5.6 QuadSPI 地址寄存器.....370
20.5.7 QuadSPI alternate bytes 寄存器.....370
20.5.8 QuadSPI 数据寄存器.....371
20.5.9 QuadSPI 状态 MASK 寄存器.....372
20.5.10 QuadSPI 状态匹配寄存器.....372
20.5.11 QuadSPI 轮询间隔寄存器.....372
20.5.12 QuadSPI 超时寄存器.....373

21 单边沿半字节传输模块（SENT）.....374

21.1 概述.....374
21.2 消息格式.....374
21.2.1 Data Nibble.....374
21.2.2 同步脉冲（Sync Pulse）.....375
21.2.3 Status and communication nibble.....375
21.2.4 消息帧格式.....377

21.2.5 Optional Pause Pulse *377*
21.2.6 CRC 校验 *377*
21.3 结构框图 378
21.4 功能描述 378
21.4.1 发送 Tick 时间 *378*
21.4.2 发送模式 *379*
21.4.3 接收模式 *380*
21.4.4 接收消息诊断 *381*
21.4.5 输入数字滤波 *382*
21.5 寄存器 382
21.5.1 SENT 控制寄存器（SENT_CR） *382*
21.5.2 SENT 配置寄存器（SENT_CFGR） *383*
21.5.3 SENT 预分频寄存器（SENT_PSR） *384*
21.5.4 SENT 帧长度寄存器（SENT_FTR） *385*
21.5.5 SENT 状态和标志寄存器（SENT_SR） *385*
21.5.6 SENT 数据寄存器（SENT_DR） *387*
21.5.7 SENT 同步脉冲诊断寄存器（SENT_SYNC） *387*
21.5.8 SENT 同步校准寄存器（SENT_RCAL） *388*

22 直接存储访问控制器（DMA） **389**

22.1 概述 389
22.2 结构框图 389
22.3 功能描述 390
22.3.1 外设 DMA 传输 *390*
22.3.2 外设请求的处理 *390*
22.3.3 通道优先级仲裁 *391*
22.3.4 传输方向定义 *391*
22.3.5 地址指针 *392*
22.3.6 访问带宽 *392*
22.3.7 传输长度 *392*
22.3.8 循环模式 *392*
22.3.9 双缓冲区模式 *394*
22.4 通道控制 394
22.4.1 DMA 请求映射 *394*
22.4.2 Request Select *399*
22.5 寄存器 399
22.5.1 DMA 全局控制寄存器（DMA_GCR） *401*
22.5.2 DMA 软件请求寄存器寄存器（DMA_SWRR） *401*
22.5.3 DMA 中断状态标志寄存器（DMA_ISR） *402*
22.5.4 通道 x 控制寄存器（DMA_CHxCR） *402*
22.5.5 通道 x 配置寄存器（DMA_CHxCFGR） *403*
22.5.6 通道 x 外设地址寄存器（DMA_CHxPAR） *404*
22.5.7 通道 x 存储器地址寄存器 0（DMA_CHxMAR0） *405*
22.5.8 通道 x 存储器地址寄存器 1（DMA_CHxMAR1） *405*
22.5.9 通道 7 控制寄存器（DMA_CH7CR） *406*
22.5.10 通道 7 Flash 指针寄存器（DMA_CH7FLSAR） *407*
22.5.11 通道 7 RAM 指针寄存器（DMA_CH7RAMAR） *408*

23 循环冗余校验（CRC） **409**

23.1 概述 409
23.2 软件配置过程 410
23.3 GOLDEN 数据 411
23.4 DMA 接口 411
23.5 FLASH 数据完整性校验 412

23.6 寄存器 413
23.6.1 CRC 数据寄存器（CRC_DR） 413
23.6.2 CRC 控制状态寄存器（CRC_CR） 413
23.6.3 CRC LFSR 寄存器（CRC_LFSR） 415
23.6.4 CRC 输出异或寄存器（CRC_XOR） 415
23.6.5 CRC 多项式寄存器（CRC_POLY） 416
24 高级定时器（ATIM） 417
24.1 概述 417
24.2 主要特性 417
24.3 结构框图 418
24.4 管脚和内部信号连接 418
24.5 功能描述 422
24.5.1 定时单元 422
24.5.2 定时器工作模式 424
24.5.3 重复计数器 431
24.5.4 Preload 寄存器 432
24.5.5 计数器工作时钟 433
24.5.6 内部触发信号（ITRx） 438
24.5.7 捕捉/比较通道 439
24.5.8 输入捕捉模式 441
24.5.9 软件 Force 输出 443
24.5.10 输出比较模式 444
24.5.11 PWM 输出 445
24.5.12 非对称 PWM 模式 448
24.5.13 组合 PWM 模式 450
24.5.14 组合三相 PWM 模式 451
24.5.15 互补输出和死区插入 452
24.5.16 刹车功能 453
24.5.17 互补输出通道信号状态逻辑表 459
24.5.18 6-step PWM 输出 459
24.5.19 单脉冲输出 460
24.5.20 可重触发的单脉冲输出（retriggerable one-pulse mode） 462
24.5.21 外部事件清除 OCxREF 463
24.5.22 编码器接口模式（encoder interface mode） 464
24.5.23 定时器同步 479
24.5.24 触发信号输出 483
24.5.25 DMA 访问 485
24.5.26 DMA Burst 486
24.5.27 输入异或功能 486
24.5.28 霍尔传感器接口 486
24.5.29 Debug 模式 487
24.6 寄存器 487
24.6.1 ATIM 控制寄存器 1（ATIM_CR1） 488
24.6.2 ATIM 控制寄存器 2（ATIM_CR2） 489
24.6.3 ATIM 从机模式控制寄存器（ATIM_SMCR） 491
24.6.4 ATIM DMA 和中断使能寄存器（ATIM_DIER） 494
24.6.5 ATIM 状态寄存器（ATIM_ISR） 496
24.6.6 ATIM 事件产生寄存器（ATIM_EGR） 497
24.6.7 ATIM 捕捉/比较模式寄存器 1（ATIM_CCMR1） 498
24.6.8 ATIM 捕捉/比较模式寄存器 2（ATIM_CCMR2） 500
24.6.9 ATIM 捕捉/比较使能寄存器（ATIM_CCER） 502
24.6.10 ATIM 计数器寄存器（ATIM_CNT） 504

24.6.11 ATIM 预分频寄存器（ATIM_PSC）..... 504
24.6.12 ATIM 自动重载（auto-reload）寄存器（ATIM_ARR）..... 505
24.6.13 ATIM 重复计数寄存器（ATIM_RCR）..... 505
24.6.14 ATIM 捕捉/比较寄存器 1（ATIM_CCR1）..... 506
24.6.15 ATIM 捕捉/比较寄存器 2（ATIM_CCR2）..... 506
24.6.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）..... 507
24.6.17 ATIM 捕捉/比较寄存器 4（ATIM_CCR4）..... 507
24.6.18 ATIM 刹车和死区控制寄存器（ATIM_BDTR）..... 508
24.6.19 ATIM DMA 控制寄存器（ATIM_DCR）..... 510
24.6.20 ATIM DMA 访问寄存器（ATIM_DMAR）..... 511
24.6.21 ATIM 捕捉/比较模式寄存器 3（ATIM_CCMR3）..... 511
24.6.22 ATIM 比较寄存器 5（ATIM_CCR5）..... 512
24.6.23 ATIM 比较寄存器 6（ATIM_CCR6）..... 513
24.6.24 ATIM 编码器模式控制寄存器（ATIM_ECR）..... 513
24.6.25 ATIM 输入选择寄存器（ATIM_TISEL）..... 514
24.6.26 ATIM 辅助功能寄存器（ATIM_AFR）..... 515

25 定时器阵列单元（TAU）..... 517

25.1 概述..... 517
25.2 结构框图..... 518
25.3 功能描述..... 519
25.3.1 时钟和复位..... 519
25.3.2 工作模式..... 519
25.3.3 输入输出通道..... 519
25.3.4 16 位计数模式..... 520
25.3.5 触发信号输出（TRGO）..... 522
25.3.6 32 位级联模式..... 522
25.3.7 捕捉功能..... 527
25.3.8 预分频..... 531
25.3.9 16 位 PWM 输出..... 531
25.3.10 级联 PWM 输出..... 532
25.3.11 外部引脚输入数字滤波..... 536
25.4 寄存器..... 537
25.4.1 TAUy 控制寄存器（TAUy_CR）..... 539
25.4.2 TAUy 通道 x 配置寄存器（TAUy_TxCFGR）..... 539
25.4.3 TAUy 通道 x 模式寄存器（TAUy_TxMDR）..... 540
25.4.4 TAUy 重载寄存器（TAUy_TxARR）..... 541
25.4.5 TAUy 捕捉比较寄存器（TAUy_TxCCR）..... 542
25.4.6 TAU0 中断使能寄存器（TAU0_TxIER）..... 542
25.4.7 TAUy 中断标志寄存器（TAUy_TxISR）..... 543
25.4.8 TAUy 计数值寄存器（TAUy_TxCNTR）..... 543

26 16 位基本定时器（BSTIM16）..... 545

26.1 概述..... 545
26.2 主要特性..... 545
26.3 结构框图..... 545
26.4 功能描述..... 546
26.4.1 定时单元..... 546
26.4.2 定时器工作模式..... 548
26.4.3 计数器工作时钟..... 550
26.4.1 Debug 模式..... 551
26.5 寄存器..... 552
26.5.1 BSTIM 控制寄存器 1..... 552

26.5.2 BSTIM 控制寄存器 2 553
26.5.3 BSTIM 中断使能寄存器 554
26.5.4 BSTIM 状态寄存器 554
26.5.5 BSTIM 事件产生寄存器 555
26.5.6 BSTIM 计数器寄存器 555
26.5.7 BSTIMx 预分频寄存器 556
26.5.8 BSTIMx 自动重载（auto-reload）寄存器 556

27 16 位低功耗定时器（LPTIM16） 558

27.1 概述 558
27.2 结构框图 559
27.3 时钟和复位 559
27.4 定时器功能 560
27.4.1 普通定时器 560
27.4.2 外部脉冲触发计数 560
27.4.3 外部异步脉冲计数 561
27.4.4 Timeout 模式 561
27.5 捕捉比较功能 562
27.5.1 16bit PWM 562
27.5.2 输入捕捉 563
27.5.1 输入数字滤波 564
27.6 正交编码器 565
27.7 寄存器 567
27.7.1 LPTIM 配置寄存器（LPTIM16_CFGR） 567
27.7.2 LPTIM 计数值寄存器（LPTIM16_CNT） 569
27.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR） 569
27.7.4 LPTIM 目标值寄存器（LPTIM16_ARR） 571
27.7.5 LPTIM 中断使能寄存器（LPTIM16_IER） 571
27.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR） 572
27.7.7 LPTIM 控制寄存器（LPTIM16_CR） 573
27.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1） 574
27.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2） 574

28 实时时钟（RTC） 576

28.1 概述 576
28.2 结构框图 576
28.3 工作原理 577
28.3.1 时基计数器（LTBC） 577
28.3.2 LTBC 数字调校 578
28.3.3 BCD 时间 579
28.3.4 RTC 使能与停止 580
28.3.5 RTC 时间设置 580
28.3.6 RTC 时间读取 580
28.3.7 闰年判断 581
28.4 寄存器 582
28.4.1 RTC 写使能寄存器（RTC_WER） 582
28.4.2 RTC 中断使能寄存器（RTC_IER） 583
28.4.3 RTC 中断标志寄存器（RTC_ISR） 584
28.4.4 BCD 时间秒寄存器（RTC_BCDSEC） 586
28.4.5 BCD 时间分钟寄存器（RTC_BCDMIN） 586
28.4.6 BCD 时间小时寄存器（RTC_BCDHOUR） 587
28.4.7 BCD 时间天寄存器（RTC_BCDDAY） 587
28.4.8 BCD 时间星期寄存器（RTC_BCDWEEK） 587
28.4.9 BCD 时间月寄存器（RTC_BCDMONTH） 588

28.4.10 BCD 时间年寄存器（RTC_BCDYEAR）..... 588
28.4.11 闹钟寄存器（RTC_ALARM）..... 589
28.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）..... 589
28.4.13 LTBC 数值调整寄存器（RTC_ADJUST）..... 590
28.4.14 毫秒计数值寄存器（RTC_SBSCNT）..... 591
28.4.15 RTC 控制寄存器（RTC_CR）..... 591

29 模数转换器（ADC）..... 593

29.1 概述..... 593
29.2 结构框图..... 594
29.3 输入通道..... 594
29.4 单端输入..... 595
29.5 时钟与复位..... 596
29.5.1 时钟..... 596
29.5.2 复位..... 596
29.6 工作时序..... 596
29.7 功能描述..... 598
29.7.1 ADC 转换队列（ACQ）..... 598
29.7.2 注入 ACQ 转换..... 599
29.7.3 扫描 ACQ 转换..... 601
29.7.4 触发源..... 604
29.7.5 使用 VDDA 作为基准..... 605
29.7.6 温度传感器..... 605
29.7.7 可编程采样时间..... 606
29.7.8 采样通道分组..... 607
29.7.9 转换结果..... 608
29.7.10 转换模式..... 609
29.7.11 转换触发..... 613
29.7.12 转换停止..... 616
29.7.13 过采样和硬件平均..... 617
29.7.14 ADC 工作时钟..... 617
29.7.15 数据冲突和自动等待..... 618
29.7.16 DMA..... 619
29.7.17 模拟窗口看门狗（AWD）..... 625
29.7.18 ADC 校准..... 626
29.8 低功耗模式..... 627
29.9 寄存器..... 627
29.9.1 ACQ 中断和状态寄存器..... 628
29.9.2 ACQ 中断使能寄存器..... 628
29.9.3 通用中断和状态寄存器..... 629
29.9.4 通用中断使能寄存器..... 630
29.9.5 ADC 控制寄存器 1..... 631
29.9.6 ADC 控制寄存器 2..... 631
29.9.7 ADC 配置寄存器 1..... 632
29.9.8 ADC 配置寄存器 2..... 633
29.9.9 校准数据寄存器..... 634
29.9.10 ACQ 标志寄存器..... 635
29.9.11 Scan ACQ 通道控制寄存器..... 636
29.9.12 AWD 阈值配置寄存器..... 636
29.9.13 AWD 阈值配置寄存器..... 637
29.9.14 ACQx 控制寄存器..... 637
29.9.15 SCAN_ACQ 控制寄存器..... 638
29.9.16 通道完成标志寄存器..... 640

29.9.17 DMA 通道数据寄存器 641
29.9.18 采样通道 x 数据寄存器 641

30 数模转换器（DAC） 642

30.1 概述 642
30.2 结构框图 642
30.3 引脚定义 642
30.4 功能描述 643
30.4.1 工作时钟与信号时序 643
30.4.2 DAC 输出模式 643
30.4.3 DAC 触发源选择 645
30.4.4 DAC 输出电压 645
30.4.5 DAC 输出缓冲 645
30.5 寄存器 646
30.5.1 DAC 控制寄存器 1（DAC_CR1） 646
30.5.2 DAC 控制寄存器 2（DAC_CR2） 647
30.5.3 DAC 配置寄存器（DAC_CFGR） 647
30.5.4 DAC 软件触发寄存器（DAC_SWTRGR） 648
30.5.5 DAC 数据保持寄存器（DAC_DHR） 648
30.5.6 DAC 状态标志寄存器（DAC_ISR） 649
30.5.7 DAC 中断使能寄存器（DAC_IER） 649

31 电容触摸按键（TSI） 651

31.1 概述 651
31.2 自容模式工作原理 651
31.2.1 电容检测 651
31.2.2 信噪比（SNR） 651
31.2.3 按键触摸判定 652
31.2.4 接近检测 653
31.2.5 按键超时检测 653
31.2.6 按键互斥 653
31.3 互电容模式工作原理 653
31.3.1 互电容检测 653
31.3.2 Low EMI 驱动 654
31.4 SHIELDING 工作原理 654
31.4.1 防水（Liquide tolerant） 654
31.4.2 External Shielding Driver 656
31.4.3 Internal Shielding Driver 656
31.5 通道扫描 656
31.5.1 扫描模式 656
31.5.2 自电容扫描过程 657
31.5.3 互电容扫描过程 658
31.5.4 互电容扫描时的 IO 控制 660
31.5.5 扫描触发 660
31.5.6 扫描停止 662
31.5.7 扫描通道配置 662
31.5.8 DMA 支持 663
31.6 工作时钟 665
31.6.1 SSC 原理 667
31.7 多频点测量 667
31.8 滤波器和 BASELINE 跟踪 668
31.9 中断和状态标志 668
31.10 VDD 电压适配 669
31.11 设备调校 669

31.12 寄存器 670
31.12.1 按键中断标志寄存器（TSI_ISR） 670
31.12.2 按键中断使能寄存器（TSI_IER） 671
31.12.3 接收通道扫描状态寄存器（TSI_RXSR） 672
31.12.4 发射通道扫描状态寄存器（TSI_TXSR） 672
31.12.5 扫描配置寄存器（TSI_CFGR） 673
31.12.6 扫描控制寄存器（TSI_CR） 675
31.12.7 按键通道使能寄存器 0（TSI_CH_CR0） 675
31.12.8 按键通道使能寄存器 1（TSI_CH_CR1） 676
31.12.9 按键通道配置寄存器 0（TSI_CH_CFGR0） 676
31.12.10 按键通道配置寄存器 1（TSI_CH_CFGR1） 677
31.12.11 Raw Count 寄存器（TSI_RAW_CNTR） 677
31.12.12 时钟和周期控制寄存器（TSI_CCR） 678
31.12.13 DMA 控制寄存器（TSI_DMACR） 679
31.12.14 按键通道扫描参数配置寄存器（TSI_SP_CFGR） 680
31.12.15 IDAC 调校寄存器（TSI_IDAC_TR） 680
31.12.16 IDAC 模拟控制寄存器（TSI_ANA_CR） 681
31.12.17 TSC_DFT 寄存器（TSI_TEST） 682
31.12.18 测试电容通道使能寄存器（TSI_TESTRING0） 683
31.12.19 测试电容通道使能寄存器（TSI_TESTRING1） 684
31.12.20 ANA TEST 寄存器（TSI_ANA_TEST） 684
31.12.21 VREF 基准配置寄存器（TSI_VREFCR_TEST） 686
31.12.22 屏蔽通道增强寄存器（TSI_SHLDGND_CR0） 686
31.12.23 屏蔽通道增强寄存器（TSI_SHLDGND_CR1） 687
31.12.24 低辐射驱动配置寄存器（TSI_EMI_CFGR0） 687
31.12.25 低辐射驱动配置寄存器（TSI_EMI_CFGR1） 688
31.12.26 屏蔽通道使能寄存器（TSI_SHLD_CR0） 688
31.12.27 屏蔽通道使能寄存器（TSI_SHLD_CR1） 689
31.12.28 时钟管理寄存器（TSI_CKCR） 690
31.12.29 自容伪随机生成多项式寄存器（TSI_PRS_SW） 690
31.12.30 PLL 控制寄存器（TSI_PLL_CR） 691
31.12.31 互容 SSC 生成多项式寄存器（TSI_PRS_SSC） 692

32 I/O 端口（GPIO） 693

32.1 概述 693
32.2 引脚类型 693
32.2.1 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 694
32.2.2 NRST，固定上拉，NMOS 输出驱动 695
32.3 IO 端口功能定义 695
32.3.1 GPIO 输入 695
32.3.2 GPIO 输出 696
32.3.3 数字外设功能 696
32.3.4 模拟功能 697
32.4 NRST 引脚与 GPIO 复用 697
32.5 SWD 引脚 698
32.6 PA14/PA15 引脚 698
32.7 WKUPx 引脚 698
32.8 外部引脚中断（EXTI） 699
32.8.1 功能说明 699
32.8.2 应用指南 700
32.9 快速 GPIO 输出 701
32.10 寄存器 701
32.10.1 GPIO 输入使能寄存器（GPIOx_INEN） 706

32.10.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）..... 706
32.10.3 GPIO 开漏使能寄存器（GPIOx_ODEN）..... 707
32.10.4 GPIO 功能选择寄存器（GPIOx_FCR）..... 708
32.10.5 GPIO 输出数据寄存器（GPIOx_DO）..... 710
32.10.6 GPIO 输出数据置位寄存器（GPIOx_DSET）..... 710
32.10.7 GPIO 输出数据复位寄存器（GPIOx_DRST）..... 711
32.10.8 GPIO 输入数据寄存器（GPIOx_DIN）..... 712
32.10.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）..... 712
32.10.10 GPIO 驱动强度配置寄存器（GPIOx_DSR）..... 713
32.10.11 EXTI 边沿选择和使能寄存器 A~G（GPIO_EXTIEDSA/B/.../G）..... 714
32.10.12 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）..... 714
32.10.13 EXTI 中断标志寄存器（GPIO_EXTIF）..... 715
32.10.14 FOUT 配置寄存器（GPIO_FOUTSEL）..... 715
32.10.15 WKUP 控制寄存器（GPIO_PINWKEN）..... 716

33 安全功能（SFU）..... 718

33.1 概述..... 718
33.2 闪存 CRC 检查..... 718
33.3 RAM 奇偶校验..... 719
33.4 RAM 自检..... 719
33.5 RAM 写保护..... 719
33.6 关键寄存器写保护..... 720
33.7 非法地址访问检测..... 722
33.8 时钟频率检测..... 725
33.9 晶振停振检测..... 725
33.10 PLL 失锁检测..... 725
33.11 电源电压检测..... 725
33.12 IO 输出状态回读..... 725
33.13 LIN 自测试..... 726
33.14 CAN 自测试..... 726
33.15 ADC 测试..... 726
33.16 DAC 自测试..... 727
33.17 比较器自测试..... 727
33.18 温度检测..... 728
33.19 寄存器..... 729
33.19.1 RAM 奇偶校验控制寄存器（SFU_RAMPCR）..... 729
33.19.2 RAM 奇偶校验状态寄存器（SFU_RAMPSR）..... 729
33.19.3 存储访问保护控制寄存器（SFU_MAPCR）..... 730
33.19.4 存储访问保护状态寄存器（SFU_MAPSR）..... 732

34 专用编程接口..... 733

34.1 概述..... 733
34.2 编程器使用..... 733

35 调试支持..... 734

35.1 概述..... 734
35.2 DEBUG 引脚..... 735
35.2.1 SWD 引脚..... 735
35.2.2 上拉电阻..... 735
35.3 SWD 接口协议..... 735
35.3.1 协议简介..... 735
35.3.2 传输序列..... 735
35.3.3 SW-DP ID code..... 736
35.3.4 主机读操作..... 737

35.3.5 主机写操作 737
35.4 SWD-DP 寄存器 738
35.4.1 寄存器列表 738
35.5 CORE DEBUG 寄存器 738
35.6 低功耗调试支持 738
35.7 DEBUG 相关的配置项 739
35.8 寄存器 739
35.8.1 MCU DEBUG 配置寄存器 739
35.8.2 HardFault 查询寄存器 740
36 器件签名信息 742
36.1 唯一器件 ID 742
版本列表 743
上海复旦微电子集团股份有限公司销售及服务网点 744

表目录

表 1- 1FM33FH0 型号列表.....33
表 3- 1FM33FH0 极限参数.....55
表 3- 2FM33FH0 典型工作条件.....55
表 3- 3ACTIVE 电流参数.....56
表 3- 4 SLEEP 电流参数.....56
表 3- 5DEEPSLEEP 电流参数.....57
表 3- 6 复位和电源监控参数.....58
表 3- 7 内部基准电压参数.....58
表 3- 8 唤醒时间参数.....59
表 3- 9 低频晶体振荡器参数.....59
表 3- 10 高频晶体振荡器参数.....60
表 3- 11 内部 RC 振荡器参数.....61
表 3- 12 内部 RC 振荡器参数.....61
表 3- 13 PLL 参数.....61
表 3- 14 ADC 参数.....63
表 3- 15 ADC 采样时间.....65
表 3- 16 DAC 参数.....65
表 3- 17 ADC 采样时间.....65
表 3- 18 温度传感器参数.....66
表 3- 19 模拟比较器参数.....67
表 3- 20 FLASH 参数.....67
表 3- 21 SPI TIMING CHARACTERISTICS AT VDD=5V.....68
表 3- 22 SPI TIMING CHARACTERISTICS AT VDD=3.3V.....68
表 3- 23 QSPI TIMING CHARACTERISTICS AT VDD=5V.....70
表 3- 24 QSPI TIMING CHARACTERISTICS AT VDD=3.3V.....71
表 3- 25 普通 I/O 参数.....72
表 3- 26 NRST 引脚参数.....72
表 3- 27 引脚 AC 参数.....73
表 4- 1 外设模块总线地址列表.....81
表 6- 1FM33FH0 CPU 配置简表.....118
表 6- 2 CORTEX-M0 内核寄存器简表.....119
表 7- 1 中断向量表.....123
表 7- 2 中断源多路选择表.....124
表 9- 1 IWDT 溢出周期表.....141
表 10- 1 WWDT 溢出周期表.....149
表 14- 1 比较器工作模式.....207
表 15- 1 I^2C 接口时序要求.....218
表 15- 2 I^2C 保留地址定义.....219
表 16- 1 UART 类型列表.....240
表 16- 2 UART 数据帧格式.....240
表 16- 3 DMA 发送中断.....249
表 16- 4 常用时钟频率下波特率计算.....250
表 17- 1 UART 类型列表.....284
表 17- 2 UART 数据帧格式.....284
表 17- 3 DMA 发送中断.....293
表 17- 4 常用时钟频率下波特率计算.....294
表 18- 1LPUART 数据帧格式.....312
表 18- 2 LPUART 数据位调制系数.....313
表 18- 3 LPUART DMA 中断说明.....316

表 19-1SPI 引脚列表 324
表 19-2I2S 典型采样频率计算表 337
表 22-1PDMA 通道映射列表 399
表 24-1 外部引脚连接 419
表 24-2 内部信号连接 419
表 24-3ITR 信号连接 420
表 24-4ETR 信号连接 420
表 24-5OCREF_CLR 信号连接 420
表 24-6 刹车行为 458
表 24-7 ENCODER INTERFACE 计数方式（CC1P=CC2P=0） 465
表 24-8 DIRECTIONAL CLOCK MODE 计数方式 468
表 25-1 定时器级联触发选择 527
表 25-2 定时器 x 捕捉源选择 531
表 27-1 ENCODER INTERFACE 计数方式 565
表 29-1 ADC 输入通道 595
表 29-2 ADC 采样时间 606
表 29-3 DMA 配置与功能 620
表 30-1 DAC 相关引脚 643
表 30-2 DAC 触发源 645
表 32-1GPIO 功能逻辑定义表 694
表 32-2FCR 定义表 695
表 33-1 384K CODE FLASH 版本总线访问控制表 724
表 33-2 256K CODE FLASH 版本总线访问控制表 724
表 33-3 128K CODE FLASH 版本总线访问控制表 725

图目录

图 1-1 芯片结构框图 32
图 2-1FM33FH0x10 LQFP100 封装图 34
图 2-2FM33FH0x6 LQFP64 封装图 35
图 2-3 FM33FH0 LQFP48 封装图 36
图 2-4 LQFP100 封装尺寸图 49
图 2-5 LQFP64 封装尺寸图 51
图 2-6LQFP48 封装尺寸图 53
图 3-1 测试方案 54
图 3-2 ADC 参数说明 62
图 3-3 ADC 通道输入阻抗 64
图 3-4 温度传感器输出曲线 66
图 3-5SPI MASTER TIMING 69
图 3-6SPI SLAVE TIMING, CPHA=0 69
图 3-7SPI SLAVE TIMING, CPHA=1 70
图 3-8QSPI TIMING 71
图 3-9IO TIMING CHARATERISTICS 74
图 4-1 系统总线示意图 76
图 4-2FM33FH0 总线地址（384+32+16） 77
图 5-1 芯片电源结构图 107
图 8-1 芯片复位源框图 128
图 8-2 上下电复位示意图 129
图 8-3 POR, SVD, PDR 复位波形示意图 129
图 9-1 IWDT 结构框图 140
图 9-2 IWDT 窗口示意图 142
图 10-1WWDT 结构框图 147
图 10-2WWDT 窗口示意图 149
图 11-1 芯片时钟框图 155
图 12-1CLM 全局示意图 180
图 12-2CLM 电路框图 181
图 12-3 参考时钟计数和监控时钟计数 182
图 12-4 监控时钟频率下溢出 183
图 12-5 监控时钟频率上溢出 183
图 12-6 超时检测电路 185
图 12-7TIME-OUT 复位 186
图 13-1 低压检测电路框图 191
图 13-2 低压检测电路工作时序 192
图 13-3 电源检测电路间歇工作模式 193
图 14-1 比较器电路框图 203
图 14-2 比较器内建基准缓冲器 204
图 14-3 比较器输出滞回 205
图 14-4 窗口比较器框图 206
图 14-5 窗口比较器波形示意图 206
图 14-6 数字滤波（DFLEN=3）波形示意图 207
图 14-7 比较器中断产生 208
图 14-8 比较器输出逻辑 208
图 14-9 比较器输出上升沿产生触发输出，无滤波 209
图 14-10 比较器输出上升沿下降沿产生触发输出，无滤波 209
图 15-1 I^2C 主机模块框图 215
图 15-2I^2C 总线时序 216

图 15-3 数据有效时序 216
图 15-4 起始（START）与停止(STOP)命令定义 217
图 15-5 输出应答(ACK) 217
图 15-6 主机向 7 位地址从机写入数据时的帧格式 220
图 15-7 I²C 主机对 7 位地址从机发送数据流图 221
图 15-8 主机从 7 位地址从机读取数据时的帧格式 221
图 15-9 I²C 从 7 位地址从机读取数据流图 222
图 15-10 双向数据通信帧格式 223
图 15-11 10BIT 寻址，主机向从机写入数据 223
图 15-12 10BIT 寻址，主机从从机读取数据 224
图 15-13 I²C 软件发送数据流图 224
图 15-14 I²C 主机 DMA 发送流程图 226
图 15-15 I²C 主机 DMA 接收流程图 228
图 15-16 主机时序控制 230
图 16-1 UART 接口时序 239
图 16-2 UART 字符描述 240
图 16-3 起始位同步 242
图 16-4 位接收 16 倍采样 242
图 16-5 位接收 8 倍采样 242
图 16-6 UART 异步发送波形 2 244
图 16-7 UART 异步发送波形 3 245
图 16-8 两个 UART 之间的硬件流控 246
图 16-9NRTS 流控 246
图 16-10NCTS 流控 247
图 16-11 红外调制波形 251
图 16-12 UART 发送延迟 252
图 16-13LIN 总线拓扑 252
图 16-14LIN 总线帧结构 253
图 16-15LIN BREAK FIELD 253
图 16-16LINSYNC FIELD 253
图 16-17LIN 帧 ID 254
图 16-18LIN 总线唤醒信号 256
图 16-19LIN 唤醒检测，WKUP_LEN=5 257
图 16-20NORMAL BREAK 258
图 16-21BREAK ERROR 258
图 16-22BREAK 超时 259
图 16-23 连续 BREAK 后正常接收 260
图 16-24LIN 波特率自适应过程 261
图 16-25 波特率误差较大，无 PRE-SYNC 261
图 16-26LIN 从机接收流程 262
图 16-27 字节间隔和应答间隔 262
图 16-28LIN BIT SAMPLING 263
图 16-29LIN BIT ERROR 冲突检测 264
图 16-30LIN BIT ERROR 监测点 264
图 16-31PRE-SYNC 后发现 BREAK 长度不足，BREAKERROR 266
图 16-32SYNC 段数据错误 266
图 16-33SYNC 段 0XFF 或 0X00，无 PRE-SYNC 267
图 17-1 USART 模块框图 283
图 17-2 UART 字符描述 284
图 17-3 7816-3 帧结构图 285
图 17-4 起始位同步 286
图 17-5 位接收 16 倍采样 286
图 17-6 位接收 8 倍采样 287

图 17-7 UART 异步发送波形 1 288
图 17-8 UART 异步发送波形 2 289
图 17-9 UART 异步发送波形 3 289
图 17-10 7816 发送和重发 290
图 17-11 两个 UART 之间的硬件流控 291
图 17-12 nRTS 流控 291
图 17-13 nCTS 流控 292
图 17-14 红外调制波形 296
图 17-15 UART 发送延迟 296
图 18-1 LPUART 结构框图 311
图 18-2 字符描述 312
图 19-1 SPI 结构框图 323
图 19-2 SPI 数据/时钟时序图（CPHA=0） 324
图 19-3 SPI 数据/时钟时序图（CPHA=1） 325
图 19-44 线半双工写操作 325
图 19-5 4 线半双工读操作（无 DUMMY CYCLE） 326
图 19-6 4 线半双工读操作（有 DUMMY CYCLE） 326
图 19-7 SPI MASTER/SPI SLAVE 互连 327
图 19-8 SPI SSN 时序图（SSNM=1，CPHA=0） 329
图 19-9 SPI SSN 时序图（SSNM=0） 330
图 19-10I2S 结构框图 332
图 19-11PHILIPS 标准时序 333
图 19-1232/16-BITS MSB-JUSTIFIED 时序 334
图 19-1324-BITS MSB-JUSTIFIED 时序 335
图 19-14LSB-JUSTIFIED 时序 335
图 19-15PCM 标准时序 336
图 20-1 QSPI 结构框图 350
图 20-2 QPI 模式下的 PAGE PROGRAM 354
图 20-3 QSPI XIP 示意图 356
图 20-4 SINGLE-SPI CONFIGURATION 356
图 20-5 SINGLE-SPI MODE 0 AND MODE 3 357
图 20-6 DUAL-SPI CONFIGURATION 357
图 20-7 QUAD-SPI CONFIGURATION 357
图 20-8 单线发送指令 358
图 20-9 双线发送指令 359
图 20-10 四线发送指令 359
图 20-11 延迟数据采样 360
图 20-12 FIFO 结构 362
图 20-13 STANDARD SPI 模式下发送 WRITE ENABLE 指令 363
图 20-14 STANDARD SPI 模式下发送 WRITE STATUS REGISTER2 指令 364
图 20-15 STANDARD SPI 模式下发送 ENABLE QPI 指令 364
图 20-16QPI 模式下 FAST READ QUAD I/O 操作（前一条指令 M5-4=10） 364
图 21-1NIBBLE 波形示意图 375
图 21-2 数据 0 和数据 7 的 NIBBLE 波形 375
图 21-3 短消息格式 OPTION CHANNEL 376
图 21-4 加强消息格式 OPTION CHANNEL 377
图 21-5 消息帧格式 377
图 21-6SENT 模块结构框图 378
图 21-7SENT_RX 输入数字滤波示意图 382
图 22-1 DMA 结构框图 389
图 22-2 外设请求处理 391
图 22-3 活动寄存器和影子寄存器 392
图 22-4 循环模式下的传输长度寄存器 393

图 22-5 循环模式下的源/目的地址寄存器和存储器指针.....393
图 22-6 DOUBLE BUFFERING.....394
图 23-1CRC 运算流程图.....410
图 23-2 使用 DMA 对 RAM 中的数据进行 CRC 运算.....412
图 24-1 高级定时器结构框图.....418
图 24-2 预分频从 1 变为 2 的波形.....423
图 24-3 预分频从 1 变为 4 的波形.....423
图 24-4 向上计数波形，内部时钟不分频.....424
图 24-5 向上计数波形，内部时钟 2 分频.....425
图 24-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件.....425
图 24-7ARPE=1（ATIM_ARR 预装载）时的更新事件.....426
图 24-8 向下计数，内部时钟不分频.....427
图 24-9 向下计数，内部时钟 2 分频.....427
图 24-10 向下计数，内部时钟 2 分频.....428
图 24-11 向下计数，不使用重复计数时的更新事件.....428
图 24-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6.....429
图 24-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢).....430
图 24-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出).....430
图 24-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置.....431
图 24-16 ATIM 时钟源框图.....433
图 24-17 内部时钟源模式，时钟分频因子为 1.....433
图 24-18TI2 外部时钟连接例子.....434
图 24-19 外部时钟模式 1 下的时序.....434
图 24-20 外部时钟模式 1 下的时序.....435
图 24-21 外部触发输入框图.....436
图 24-22 外部时钟模式 2 下的时序 1.....436
图 24-23 外部时钟模式 2 下的时序 2.....437
图 24-24 捕获/比较通道(通道 1 输入部分).....439
图 24-25 捕获/比较通道 1 的主电路.....439
图 24-26 捕获/比较通道的输出部分(通道 1 至 3).....440
图 24-27 捕获/比较通道的输出部分(通道 4).....440
图 24-28 PWM 输入捕获模式时序.....441
图 24-29 输出比较模式，翻转 OC1.....444
图 24-30 边沿对齐的 PWM 波形(ARR=7, PWM MODE 1).....446
图 24-31 边沿对齐的 PWM 波形(PWM MODE 1).....446
图 24-32 中央对齐的 PWM 波形(APR=7, PWM MODE 1).....447
图 24-33 非对称 PWM 输出.....448
图 24-34 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 1）.....449
图 24-35 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（ALL PWM MODE 2）.....449
图 24-36 通道 1/3 组合 PWM 输出.....450
图 24-37 组合三相 PWM 输出.....451
图 24-38 带死区插入的互补输出.....452
图 24-39 死区波形延迟大于负脉冲.....452
图 24-40 死区波形延迟大于正脉冲.....452
图 24-41 死区插入后的互补输出波形与 CCXP 和 CCXNP 的关系.....453
图 24-42 模式与输出状态术语.....454
图 24-43 刹车输入结构.....455
图 24-44 响应刹车的输出.....457
图 24-45 产生六步 PWM，使用 COM 的例子(OSSR=1).....460
图 24-46 单脉冲模式的例子.....461
图 24-47 重触发 OPM 模式 1，向上计数，CCR=0.....462
图 24-48 重触发 OPM 模式 2，向上计数，CCR=0.....463
图 24-49 重触发 OPM 模式 1，向下计数，CCR=0.....463

图 24-50 重触发 OPM 模式 2，向下计数，CCR=0 463
图 24-51 ETR 信号清除 ATIM 的 OCxREF 464
图 24-52 编码器模式下的计数器操作实例 465
图 24-53 正交编码器不同边沿计数模式示例 466
图 24-54 时钟方向编码器示例 467
图 24-55 带方向的时钟编码器（CC1P=CC2P=0） 467
图 24-56 带方向的时钟编码器（CC1P=CC2P=1） 468
图 24-57 编码器 INDEX 定义 469
图 24-58IPOS[1:0]=11 时的 INDEX 识别 470
图 24-59IPOS[1:0]=11, INDEX GATED ON A, ARR=9 471
图 24-60IPOS[1:0]=00, INDEX UNGATED, ARR=9 471
图 24-61IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9 472
图 24-62IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LEADING CHANNEL B 473
图 24-63IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING CHANNEL B 474
图 24-64IPOS[1:0]=11, INDEX GATED ON A&B, ARR=9, INDEX LAGGING B, ZOOMED 474
图 24-65IPOS[1:0]=01, UNGATED INDEX 475
图 24-66IPOS[0]=1, CLOCK+DIRECTION MODE 476
图 24-67IPOS[0]=1, DIRECTIONAL CLOCK MODE 476
图 24-68 正交编码状态转移 477
图 24-69INDEX 错误撤销 477
图 24-70INDEX 错误置位 478
图 24-71 向下计数时 INDEX 错误置位 478
图 24-72COUNTER MODE PRELOAD 479
图 24-73 复位模式下的时序 480
图 24-74 门控模式下的时序 481
图 24-75 多 TIMER 同时使能 481
图 24-76 触发器模式下的时序 482
图 24-77 外部时钟模式 2＋触发模式下的时序 483
图 24-78 霍尔传感器接口 487
图 25-1TAU 结构框图 518
图 25-2TIMERX 结构框图 518
图 25-3 16BIT 连续定时功能 520
图 25-4 定时器内部时钟预分频计数 521
图 25-5 定时器外部事件上升沿递增时序 521
图 25-6 定时器外部事件上升下降沿递增，2 分频时序 521
图 25-7 定时器内部信号下降沿递增 522
图 25-8 定时器级联计数（TRIGGER TO INCREMENT） 523
图 25-9 定时器级联计数（TRIGGER TO INCREMENT），从机 PRESCALE1 = 1 523
图 25-10 定时器级联计数（TRIGGER TO START） 524
图 25-11 定时器级联计数（TRIGGER TO START），从机无法产生溢出 525
图 25-12 定时器级联计数（TRIGGER TO START），从机计数预分频 526
图 25-13 定时器捕捉外部输入信号上升沿 528
图 25-14 定时器 2 分频捕捉外部输入信号下降沿 528
图 25-15 定时器脉冲宽度高电平捕捉 529
图 25-16TI00 二分频计数 531
图 25-17TI00 二分频捕捉 531
图 25-1816 位 PWM 输出示意图 532
图 25-19 从机 ARR 大于主机的情况 534
图 25-20 非对称非交叠 PWM 输出示意图 535
图 25-21TIXX 输入数字滤波示意图 536
图 26-1 16 位基本定时器结构框图 545
图 26-2 预分频从 1 变为 2 的波形 547
图 26-3 预分频从 1 变为 4 的波形 547

图 26-4 向上计数波形，内部时钟不分频 548
图 26-5 向上计数波形，内部时钟 2 分频 549
图 26-6 ARPE=0（ARR 没有预装载）时的更新事件 549
图 26-7 ARPE=1（ARR 预装载）时的更新事件 550
图 26-8 内部时钟源模式，时钟分频因子为 1 550
图 27-1 LPTIM16 结构框图 559
图 27-2 外部 ETR 脉冲上升沿触发计数 560
图 27-3 外部 ETR 脉冲异步计数（下降沿） 561
图 27-4TIMEOUT 模式 562
图 27-5PWM 输出 563
图 27-6 输入信号边沿捕捉 564
图 27-7 通道输入数字滤波 564
图 27-8 编码器模式下的计数器操作实例 566
图 28-1RTC 结构框图 576
图 28-2LTBC 结构框图 577
图 28-3RTC 时间读取流程图 581
图 29-1 ADC 结构框图 594
图 29-2 单端输入 596
图 29-3 ADC 校准时序 596
图 29-4 ADC 采样转换时序 597
图 29-5 ADC 采样序列时序 597
图 29-6ACQ 队列控制器 598
图 29-7 注入时序 601
图 29-8SCAN_ACQ 单次转换全自动触发模式 602
图 29-9SCAN_ACQ 单次转换半自动触发模式 603
图 29-10SCAN_ACQ 连续转换模式 604
图 29-11 ACQ 触发源 605
图 29-12 ADC 输入通道示意图 607
图 29-13 ADC 单次转换全自动触发模式 610
图 29-14 ADC 单次转换半自动触发模式 611
图 29-15 ADC 连续转换模式 612
图 29-16 单次注入触发执行 2 次注入转换 613
图 29-17 ADC 硬件触发源 615
图 29-18 ADC 注入组硬件触发源 615
图 29-19 常规组转换 STOP 616
图 29-20 常规组和注入组转换 STOP 617
图 29-21 ADC 时钟示意图 618
图 29-22 ADC 自动等待 618
图 29-23 ADC 单次全自动触发+DMA 案例 1 621
图 29-24 ADC 单次全自动触发+DMA 案例 2 622
图 29-25 ADC 单次半自动触发+DMA 623
图 29-26 ADC 全自动触发+DMA 循环模式 624
图 29-27 ADC 连续模式+DMA 循环模式 625
图 29-28 ADC 模拟看门狗阈值示意图 626
图 30-1 DAC 结构框图 642
图 30-2 DAC 工作时序 643
图 30-3 DAC 连续输出模式 644
图 30-4 DAC 触发输出模式 645
图 31-1 按键触摸与释放 652
图 31-2 通道扫描周期 658
图 31-3WAIT=0，扫描 2 个通道 661
图 31-4WAIT=1，扫描 2 个通道 662
图 31-5 扫描停止 662

图 31-6TSI 工作时钟 665
图 31-7 开关时钟与调制器时钟 666
图 31-8SMPL_SIZE=2，扫描 4 个通道 669
图 32-1 普通 GPIO 结构框图 694
图 32-2WKUPx 功能结构框图 698
图 32-3 引脚输入数字滤波 699
图 32-4 EXTI 信号输入示意图 699
图 35-1 CORTEX-M0 调试系统示意图 734

1 产品综述

1.1 概述

FM33FH0是基于ARM Cortex-M0的低功耗MCU，支持最大384KB程序flash、16KB数据flash和最大32KB RAM；集成12bit SAR-ADC，LIN控制器，最大36通道电容式按键触摸控制器，比较器等丰富外设；具备超宽工作电压范围和优异的低功耗性能。

主要特性如下：

- 宽电压范围：2.2~5.5V
- 工作温度范围（T_A）：-40℃~+105℃
- 处理器内核
  - ARM Cortex-M0
  - 支持用户/特权模式
  - 支持中断向量表重定向（VTOR）
  - 最高主频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120uA/MHz@48Mhz
  - Sleep模式：5uA
  - DeepSleep模式，RTC走时+全部RAM保持+CPU内核保持：2.5uA
- 存储器
  - 最大384KB Code Flash w/ ECC，擦写寿命：100,000次
  - 16KB Data Flash w/ ECC，擦写寿命：500,000次
  - Flash数据保存时间：10年@85℃
  - 用户代码保护
  - 32KB RAM空间，支持parity
- 丰富的模拟外设
  - 超低功耗PDR电路
  - 可编程电源监测模块（SVD）
  - 2x低功耗模拟比较器
  - 12bit 2Msps SAR-ADC，最大26个外部输入通道
  - 10bitDAC

- 温度传感器，全温区线性度+/-2℃

- 电容触摸按键
  - 最大支持36个独立按键通道
  - 自电容检测和互电容检测
  - 防水功能
  - 硬件滤波
- 通用通信接口
  - UART*6，支持LIN通信
  - USART*2，支持智能卡接口
  - LPUART*1
  - SPI*3，主从模式
  - I2C主机*2，400K
  - SENT *2
  - QSPI *1，支持XIP
  - 7通道外设DMA
  - 可编程CRC校验模块
- 定时资源
  - TAU定时器阵列，包含16个独立通道
  - 高级定时器*1
  - 24-bit Systick*1
  - 16bit低功耗定时器*1
  - 16bit基本定时器*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
- 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%，8/16MHz全温区变化小于+/-2%
  - 低功耗32768Hz晶体振荡器，带有停振检测电路
  - 低功耗低速RC振荡器，32KHz，全温区+/-3%
  - 高频晶体振荡器，4~40MHz
  - PLL，最高输出72MHz
- 封装：LQFP100/LQFP64/LQFP48/QFN32

1.2 芯片结构框图

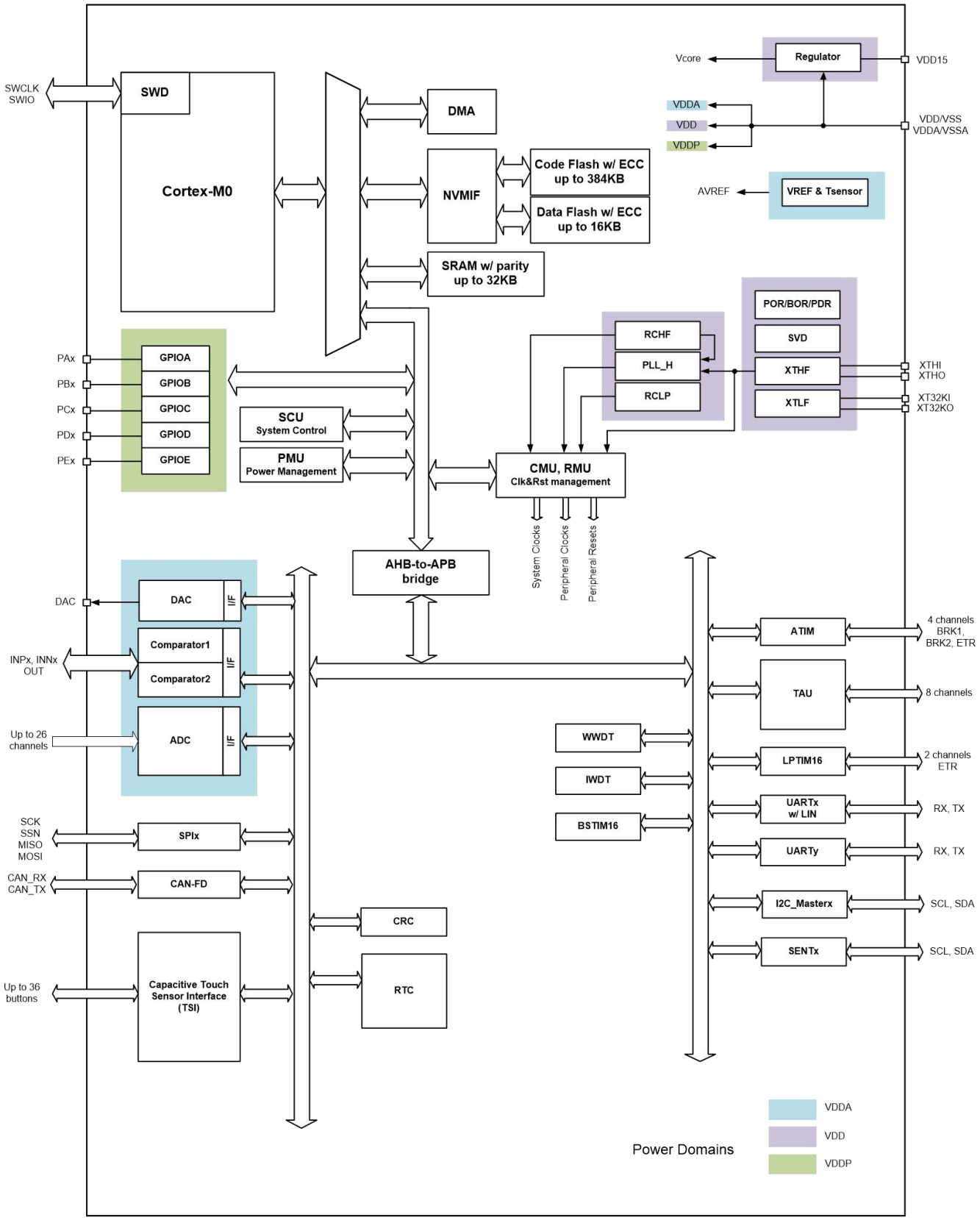


图 1-1 芯片结构框图

1.3 产品型号列表

型号	Code Flash (Kbytes）	Data Flash (Kbytes)	RAM (Kbytes)	封装
FM33FH0510	384	16	32	LQFP100
FM33FH0410	256	16	32	LQFP100
FM33FH056	384	16	32	LQFP64
FM33FH046	256	16	32	LQFP64
FM33FH026	128	16	16	LQFP64
FM33FH055	384	16	32	LQFP48
FM33FH045	256	16	32	LQFP48
FM33FH025	128	16	16	LQFP48

表 1-1FM33FH0 型号列表

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP100 封装图（FM33FH0x10）

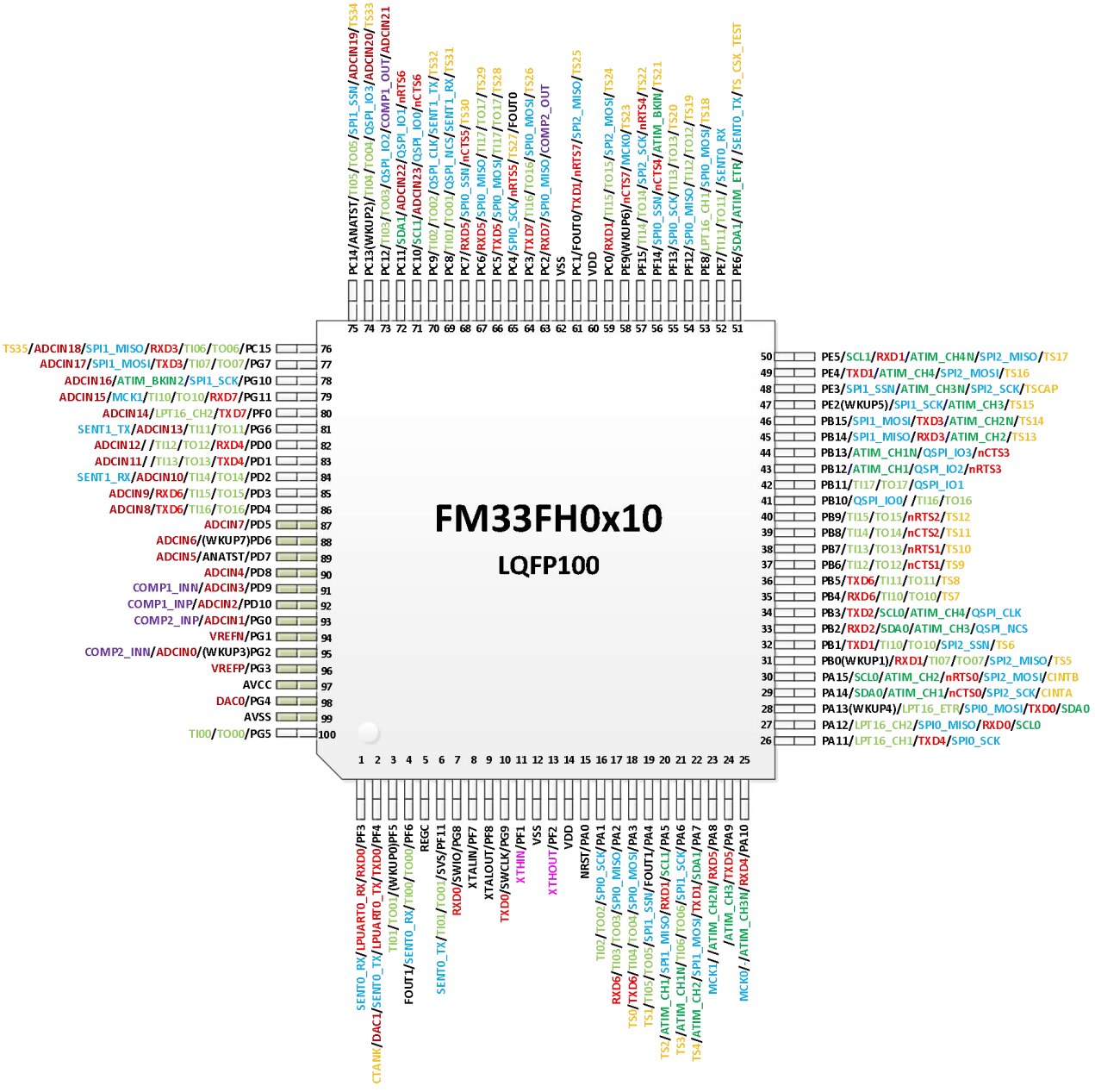


图 2-1FM33FH0x10 LQFP100 封装图

2.1.2 LQFP64 封装图（FM33FH0x6）

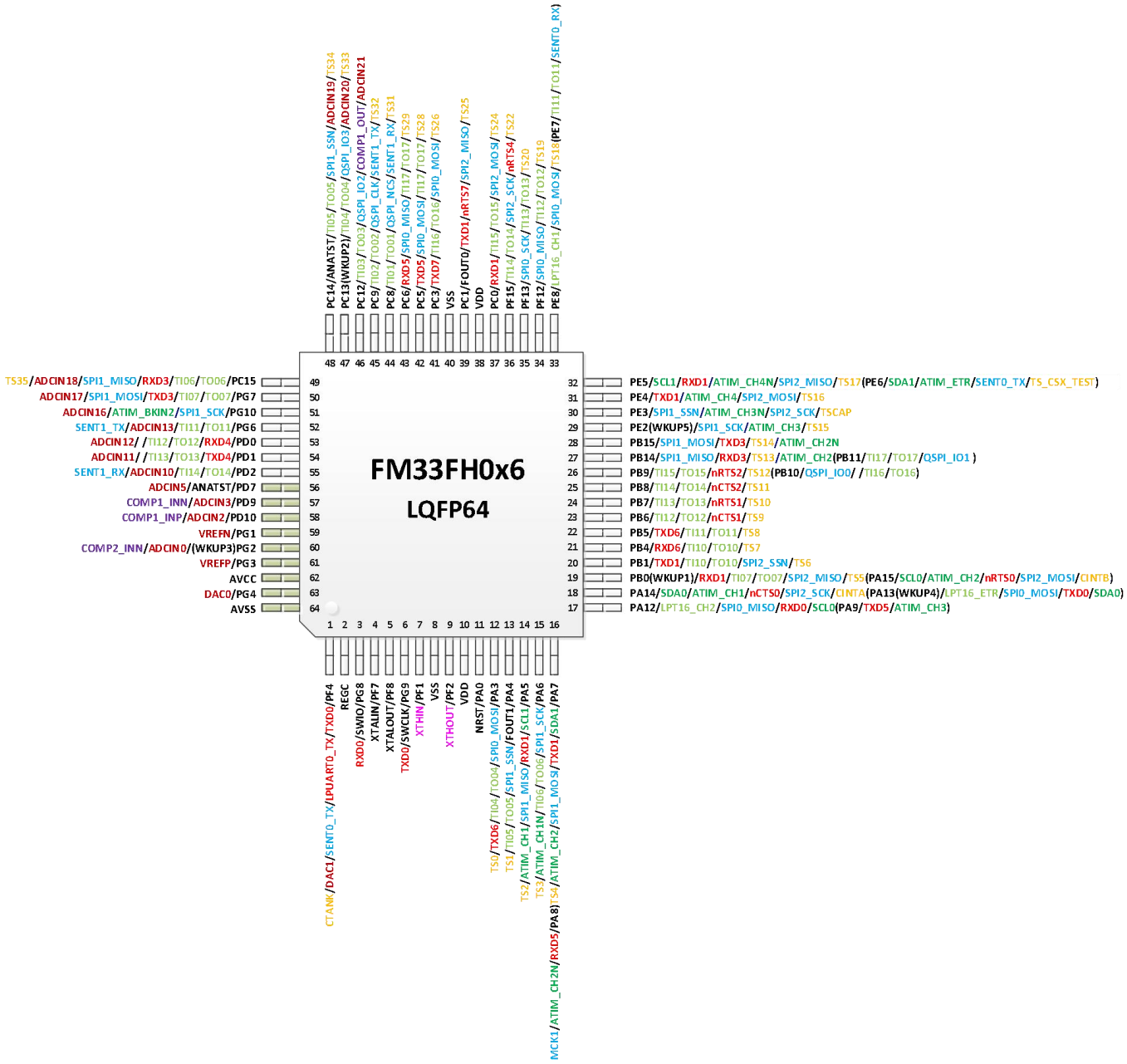


图 2-2FM33FH0x6 LQFP64 封装图

2.1.3 LQFP48 封装图（FM33FH0x5）

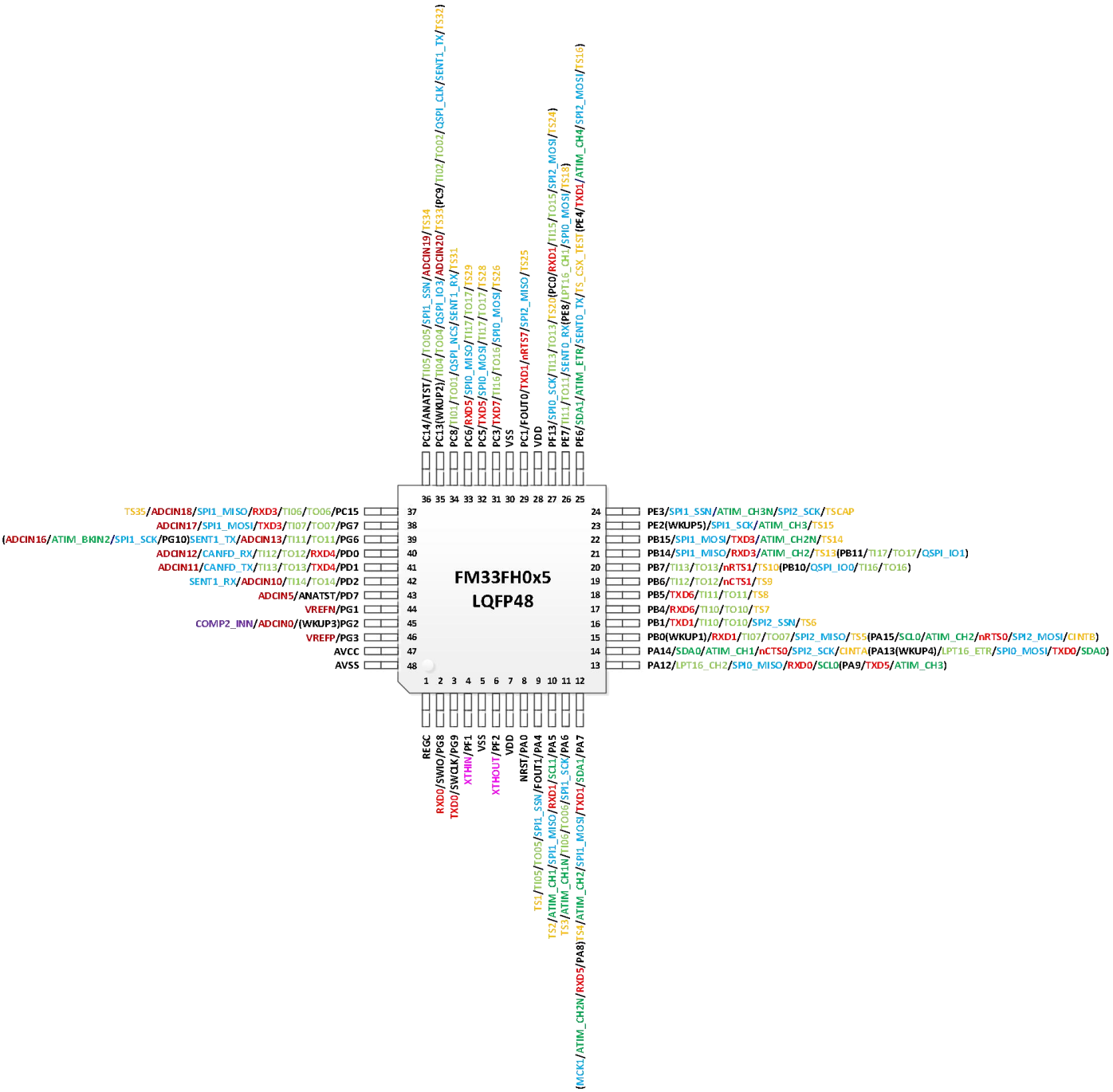


图 2-3 FM33FH0 LQFP48 封装图

2.1.4 引脚功能定义（FM33FH0xx）

注意 PA14/PA15 上电复位后默认为输入态、上拉使能。PG8/PG9 上电复位后默认为 SWD 功能、上拉使能。

Pin number			Pin Name	功能	描述
LQFP10	LQFP6	LQFP4			

0	**4**	**8**			
1			PF3	UART0_RX LPUART0_RX SENT0_RX	UART 接收 低功耗 UART 接收 SENT 接收
2	1		PF4	UART0_TX LPUART0_TX SENT0_TX DAC1 CTANK	UART 发送 低功耗 UART 发送 SENT 发送 DAC 输出（模拟功能） TSI 的 CTANK 电容引脚（模拟功能）
3			PF5	WKUP0 TO01 TI01	外部唤醒引脚 TAU 定时器输出 TAU 定时器输入
4			PF6	TO00 TI00 SENT0_RX FOUT1	TAU 定时器输出 TAU 定时器输入 SENT 接收 时钟频率测试输出管脚
5	2	1	REGC		LDO 输出，外接 1uF~4.7uF 电容到地
6			PF11	TO01 TI01 SENT0_TX SVS	TAU 定时器输出 TAU 定时器输入 SENT 发送 外部电源检测（模拟功能）
7	3	2	PG8	SWIO UART0_RX	SWD 接口数据 UART 接收
8	4		PF7	XTALIN	32768Hz 晶振输入脚（模拟功能）
9	5		PF8	XTALOUT	32768Hz 晶振输出脚（模拟功能）
10	6	3	PG9	SWCLK UART0_TX	SWD 接口时钟 UART 发送
11	7	4	PF1	XTHIN	高频晶振输入脚（模拟功能）
12	8	5	VSS		地
13	9	6	PF2	XTHOUT	高频晶振输出脚（模拟功能）
14	10	7	VDD		电源
15	11	8	PA0	NRST	全局复位引脚
16			PA1	SPI0_SCK TO02 TI02	SPI 时钟 TAU 定时器输出 TAU 定时器输入
17			PA2	SPI0_MISO TO03 TI03 UART6_RX	SPI 数据 TAU 定时器输出 TAU 定时器输入 UART 接收
18	12		PA3	SPI0_MOSI	SPI 数据

				TO04 TI04 UART6_TX TS0	TAU 定时器输出 TAU 定时器输入 UART 发送 触摸按键通道(模拟功能)
19	13	9	PA4	FOUT1 SPI1_SSN TO05 TI05 TS1	时钟频率测试输出管脚 SPI 片选 TAU 定时器输出 TAU 定时器输入 触摸按键通道(模拟功能)
20	14	10	PA5	SCL1 UART1_RX SPI1_MISO ATIM_CH1 TS2	I2C 时钟 UART 接收 SPI 数据 ATIM 外部通道 触摸按键通道(模拟功能)
21	15	11	PA6	SPI1_SCK TO06 TI06 ATIM_CH1N TS3	SPI 时钟 TAU 定时器输出 TAU 定时器输入 ATIM 外部通道 触摸按键通道(模拟功能)
22	16	12	PA7	SDA1 UART1_TX SPI1_MOSI ATIM_CH2 TS4	I2C 数据 UART 发送 SPI 数据 ATIM 外部通道 触摸按键通道(模拟功能)
23	16	12	PA8	UART5_RX ATIM_CH2N FDCAN_RX MCK1	UART 接收 ATIM 外部通道 FDCAN 接收 I2S 输出时钟
24	17	13	PA9	UART5_TX ATIM_CH3 FDCAN_TX	UART 发送 ATIM 外部通道 FDCAN 发送
25			PA10	UART4_RX ATIM_CH3N FDCAN_TX_DRATE MCK0	UART 接收 ATIM 外部通道 FDCAN 测试管脚 I2S 输出时钟
26			PA11	LPT_CH1 UART4_TX SPI0_SCK FDCAN_RX_DRATE	LPTIMER 通道 UART 发送 SPI 时钟 FDCAN 测试管脚

27	17	13	PA12	LPT_CH2 SPI0_MISO UART0_RX SCL0	LPTIMER 通道 SPI 数据 UART 接收 I2C 时钟
28	18	14	PA13	WKUP4 LPT_ETR SPI0_MOSI UART0_TX SDA0	外部唤醒引脚 LPTIMER 外部触发输入 SPI 数据 UART 发送 I2C 数据
29	18	14	PA14	SDA0 ATIM_CH1 nCTS0 SPI2_SCK CINTA	I2C 数据 ATIM 外部通道 URAT 的 CTS 流控 SPI 时钟 触摸按键互容 INTA 电容管脚(模拟功能)
30	19	15	PA15	SCL0 ATIM_CH2 nRTS0 SPI2_MOSI CINTB	I2C 时钟 ATIM 外部通道 URAT 的 RTS 流控 SPI 数据 触摸按键互容 INTB 电容管脚(模拟功能)
31	19	15	PB0	WKUP1 UART1_RX TI07 TO07 SPI2_MISO TS5	外部唤醒引脚 UART 接收 TAU 定时器输入 TAU 定时器输出 SPI 数据 触摸按键通道(模拟功能)
32	20	16	PB1	UART1_TX TI10 TO10 SPI2_SSN TS6	UART 发送 TAU 定时器输入 TAU 定时器输出 SPI 片选 触摸按键通道(模拟功能)
33			PB2	UART2_RX SDA0 ATIM_CH3 QSPI_NCS	UART 接收 I2C 数据 ATIM 外部通道 QSPI 片选
34			PB3	UART2_TX SCL0 ATIM_CH4 QSPI_CLK	UART 发送 I2C 时钟 ATIM 外部通道 QSPI 时钟
35	21	17	PB4	UART6_RX	UART 接收

				TI10 TO10 TS7	TAU 定时器输入 TAU 定时器输出 触摸按键通道(模拟功能)
36	22	18	PB5	UART6_TX TI11 TO11 TS8	UART 发送 TAU 定时器输入 TAU 定时器输出 触摸按键通道(模拟功能)
37	23	19	PB6	TI12 TO12 nCTS1 TS9	TAU 定时器输入 TAU 定时器输出 URAT 的 CTS 流控 触摸按键通道(模拟功能)
38	24	20	PB7	TI13 TO13 nRTS1 TS10	TAU 定时器输入 TAU 定时器输出 URAT 的 RTS 流控 触摸按键通道(模拟功能)
39	25		PB8	TI14 TO14 nCTS2 TS11	TAU 定时器输入 TAU 定时器输出 URAT 的 CTS 流控 触摸按键通道(模拟功能)
40	26		PB9	TI15 TO15 nRTS2 TS12	TAU 定时器输入 TAU 定时器输出 URAT 的 RTS 流控 触摸按键通道(模拟功能)
41	26	20	PB10	QSPI_IO0 FDCAN_RX TI16 TO16	QSPI 数据接口 FDCAN 接收 TAU 定时器输入 TAU 定时器输出
42	27	21	PB11	TI17 TO17 QSPI_IO1 FDCAN_TX	TAU 定时器输入 TAU 定时器输出 QSPI 数据接口 FDCAN 发送
43			PB12	ATIM_CH1 QSPI_IO2 nRTS3	ATIM 外部通道 QSPI 数据接口 URAT 的 RTS 流控
44			PB13	ATIM_CH1N QSPI_IO3 nCTS3	ATIM 外部通道 QSPI 数据接口 URAT 的 CTS 流控
45	27	21	PB14	SPI1_MISO UART3_RX ATIM_CH2 TS13	SPI 数据 UART 接收 ATIM 外部通道 触摸按键通道(模拟功能)

46	28	22	PB15	SPI1_MOSI UART3_TX ATIM_CH2N TS14	SPI 数据 UART 发送 ATIM 外部通道 触摸按键通道(模拟功能)
47	29	23	PE2	WKUP5 SPI1_SCK ATIM_CH3 TS15	外部唤醒引脚 SPI 时钟 ATIM 外部通道 触摸按键通道(模拟功能)
48	30	24	PE3	SPI1_SSN ATIM_CH3N SPI2_SCK TSCAP	SPI 片选 ATIM 外部通道 SPI 时钟 触摸按键自容 CAP 电容管脚(模拟功能)
49	31	25	PE4	UART1_TX ATIM_CH4 SPI2_MOSI TS16	UART 发送 ATIM 外部通道 SPI 数据 触摸按键通道(模拟功能)
50	32		PE5	SCL1 UART1_RX ATIM_CH4N SPI2_MISO TS17	I2C 时钟 UART 接收 ATIM 外部通道 SPI 数据 触摸按键通道(模拟功能)
51	32	25	PE6	SDA1 ATIM_ETR FDCAN_RX SENT0_TX TS_CSX_TEST	I2C 数据 ATIM 外部触发输入 FDCAN 接收 SENT 发送 触摸按键互容测试通道（模拟功能）
52	33	26	PE7	TI11 TO11 FDCAN_TX SENT0_RX	TAU 定时器输入 TAU 定时器输出 FDCAN 发送 SENT 接收
53	33	26	PE8	LPT_CH1 SPI0_MOSI TS18	LPTIMER 通道 SPI 数据 触摸按键通道(模拟功能)
54	34		PF12	SPI0_MISO TI12 TO12 TS19	SPI 数据 TAU 定时器输入 TAU 定时器输出 触摸按键通道(模拟功能)
55	35	27	PF13	SPI0_SCK TI13 TO13	SPI 时钟 TAU 定时器输入 TAU 定时器输出

				TS20	触摸按键通道(模拟功能)
56			PF14	SPI0_SSN nCTS4 ATIM_BKIN TS21	SPI 片选 URAT 的 CTS 流控 ATIM 刹车输入 触摸按键通道(模拟功能)
57	36		PF15	TI14 TO14 SPI2_SCK nRTS4 TS22	TAU 定时器输入 TAU 定时器输出 SPI 时钟 URAT 的 RTS 流控 触摸按键通道(模拟功能)
58			PE9	WKUP6 nCTS7 MCK0 TS23	外部唤醒引脚 URAT 的 CTS 流控 I2S 输出时钟 触摸按键通道(模拟功能)
59	37	27	PC0	UART1_RX TI15 TO15 SPI2_MOSI TS24	UART 接收 TAU 定时器输入 TAU 定时器输出 SPI 数据 触摸按键通道(模拟功能)
60	38	28	VDD		电源
61	39	29	PC1	FOUT0 UART1_TX nRTS7 SPI2_MISO TS25	时钟频率测试输出管脚 UART 发送 URAT 的 RTS 流控 SPI 数据 触摸按键通道(模拟功能)
62	40	30	VSS		地
63			PC2	UART7_RX SPI0_MISO COMP2_OUT	UART 接收 SPI 数据 比较器输出
64	41	31	PC3	UART7_TX TI16 TO16 SPI0_MOSI TS26	UART 发送 TAU 定时器输入 TAU 定时器输出 SPI 数据 触摸按键通道(模拟功能)
65			PC4	SPI0_SCK nRTS5 FOUT0 TS27	SPI 时钟 URAT 的 RTS 流控 时钟频率测试输出管脚 触摸按键通道(模拟功能)
66	42	32	PC5	UART5_TX SPI0_MOSI TI17	UART 发送 SPI 数据 TAU 定时器输入

				TO17 TS28	TAU 定时器输出 触摸按键通道(模拟功能)
67	43	33	PC6	UART5_RX SPI0_MISO TI17 TO17 TS29	UART 接收 SPI 数据 TAU 定时器输入 TAU 定时器输出 触摸按键通道(模拟功能)
68			PC7	UART5_RX SPI0_SSN nCTS5 TS30	UART 接收 SPI 片选 URAT 的 CTS 流控 触摸按键通道(模拟功能)
69	44	34	PC8	TI01 TO01 QSPI_NCS SENT1_RX TS31	TAU 定时器输入 TAU 定时器输出 QSPI 片选 SENT 接收 触摸按键通道(模拟功能)
70	45	35	PC9	TI02 TO02 QSPI_CLK SENT1_TX TS32	TAU 定时器输入 TAU 定时器输出 QSPI 时钟 SENT 发送 触摸按键通道(模拟功能)
71			PC10	SCL1 QSPI_IO0 nCTS6	I2C 时钟 QSPI 数据接口 URAT 的 CTS 流控
72			PC11	SDA1 QSPI_IO1 nRTS6	I2C 数据 QSPI 数据接口 URAT 的 RTS 流控
73	46		PC12	TI03 TO03 QSPI_IO2 COMP1_OUT ADCIN21	TAU 定时器输入 TAU 定时器输出 QSPI 数据接口 比较器输出 ADC 输入通道（模拟功能）
74	47	35	PC13	WKUP2 TI04 TO04 QSPI_IO3 ADCIN20 TS33	外部唤醒引脚 TAU 定时器输入 TAU 定时器输出 QSPI 数据接口 ADC 输入通道（模拟功能） 触摸按键通道(模拟功能)
75	48	36	PC14	TI05 TO05 SPI1_SSN	TAU 定时器输入 TAU 定时器输出 SPI 片选

				ADCIN19 TS34	ADC 输入通道（模拟功能） 触摸按键通道(模拟功能)
76	49	37	PC15	TO06 TI06 UART3_RX SPI1_MISO	TAU 定时器输出 TAU 定时器输入 UART 接收 SPI 数据
77	50	38	PG7	TO07 TI07 UART3_TX SPI1_MOSI ADCIN17	TAU 定时器输出 TAU 定时器输入 UART 发送 SPI 数据 ADC 输入通道（模拟功能）
78	51	39	PG10	SPI1_SCK ATIM_BKIN2 ADCIN16	SPI 时钟 ATIM 刹车输入 ADC 输入通道（模拟功能）
79			PG11	UART7_RX TO10 TI10 MCK1 ADCIN15	UART 接收 TAU 定时器输出 TAU 定时器输入 I2S 输出时钟 ADC 输入通道（模拟功能）
80			PF0	UART7_TX LPT_CH2 ADCIN14	UART 发送 LPTIMER 通道 ADC 输入通道（模拟功能）
81	52	39	PG6	TO11 TI11 SENT1_TX ADCIN13	TAU 定时器输出 TAU 定时器输入 SENT 发送 ADC 输入通道（模拟功能）
82	53	40	PD0	UART4_RX TO12 TI12 FDCAN_RX ADCIN12	UART 接收 TAU 定时器输出 TAU 定时器输入 FDCAN 接收 ADC 输入通道（模拟功能）
83	54	41	PD1	UART4_TX TO13 TI13 FDCAN_TX ADCIN11	UART 发送 TAU 定时器输出 TAU 定时器输入 FDCAN 发送 ADC 输入通道（模拟功能）
84	55	42	PD2	TO14 TI14 SENT1_RX ADCIN10	TAU 定时器输出 TAU 定时器输入 SENT 接收 ADC 输入通道（模拟功能）
85			PD3	TO15	TAU 定时器输出

				TI15 UART6_RX ADCIN9	TAU 定时器输入 UART 接收 ADC 输入通道（模拟功能）
86			PD4	TO16 TI16 UART6_TX ADCIN8	TAU 定时器输出 TAU 定时器输入 UART 发送 ADC 输入通道（模拟功能）
87			PD5	ADCIN7	ADC 输入通道（模拟功能）
88			PD6	WKUP7 ADCIN6	外部唤醒引脚 ADC 输入通道（模拟功能）
89	56	43	PD7	ADCIN5	ADC 输入通道（模拟功能）
90			PD8	ADCIN4	ADC 输入通道（模拟功能）
91	57		PD9	ADCIN3 COMP1_INN	ADC 输入通道（模拟功能） 比较器输入
92	58		PD10	ADCIN2 COMP1_INP	ADC 输入通道（模拟功能） 比较器输入
93			PG0	ADCIN1 COMP2_INP	ADC 输入通道（模拟功能） 比较器输入
94	59	44	PG1	VREFN	ADC 基准地
95	60	45	PG2	ADCIN0 COMP2_INN	ADC 输入通道（模拟功能） 比较器输入
96	61	46	PG3	WKUP3 VREFP	外部唤醒引脚 ADC 基准源
97	62	47	AVCC		电源
98	63		PG4	DAC0	DAC 输出（模拟功能）
99	64	48	AVSS		地
100			PG5	TO00 TI00	TAU 定时器输出 TAU 定时器输入

注：多个引脚使用相同pin number时会互相影响，请选择其中一个管脚作为功能使用。

2.1.5 PA 引脚组数字外设功能

GPIO	PADFS			
	00	01	10	11
PA0	NRST	-	-	-
PA1	SPI0_SCK	TO02	TI02	-
PA2	SPI0_MISO	TO03	TI03	UART6_RX
PA3	SPI0_MOSI	TO04	TI04	UART6_TX
PA4	FOUT1	SPI1_SSN	TO05	TI05
PA5	SCL1	UART1_RX	SPI1_MISO	ATIM_CH1
PA6	SPI1_SCK	TO06	TI06	ATIM_CH1N
PA7	SDA1	UART1_TX	SPI1_MOSI	ATIM_CH2
PA8	UART5_RX	ATIM_CH2N	-	MCK1

GPIO	PADFS			
	00	01	10	11
PA9	UART5_TX	ATIM_CH3	-	-
PA10	UART4_RX	ATIM_CH3N	-	MCK0
PA11	LPT_CH1	UART4_TX	SPI0_SCK	-
PA12	LPT_CH2	SPI0_MISO	UART0_RX	SCL0
PA13	LPT_ETR	SPI0_MOSI	UART0_TX	SDA0
PA14	SDA0	ATIM_CH1	nCTS0	SPI2_SCK
PA15	SCL0	ATIM_CH2	nRTS0	SPI2_MOSI

2.1.6 PB 引脚组数字外设功能

GPIO	PBDFS			
	00	01	10	11
PB0	UART1_RX	TI07	TO07	SPI2_MISO
PB1	UART1_TX	TI10	TO10	SPI2_SSN
PB2	UART2_RX	SDA0	ATIM_CH3	QSPI_NCS
PB3	UART2_TX	SCL0	ATIM_CH4	QSPI_CLK
PB4	UART6_RX	TI10	TO10	-
PB5	UART6_TX	TI11	TO11	-
PB6	TI12	TO12	nCTS1	-
PB7	TI13	TO13	nRTS1	-
PB8	TI14	TO14	nCTS2	-
PB9	TI15	TO15	nRTS2	-
PB10	QSPI_IO0	-	TI16	TO16
PB11	TI17	TO17	QSPI_IO1	-
PB12	ATIM_CH1	QSPI_IO2	nRTS3	-
PB13	ATIM_CH1N	QSPI_IO3	nCTS3	-
PB14	SPI1_MISO	UART3_RX	ATIM_CH2	-
PB15	SPI1_MOSI	UART3_TX	ATIM_CH2N	-

2.1.7 PC 引脚组数字外设功能

GPIO	PCDFS			
	00	01	10	11
PC0	UART1_RX	TI15	TO15	SPI2_MOSI
PC1	FOUT0	UART1_TX	nRTS7	SPI2_MISO
PC2	UART7_RX	SPI0_MISO	COMP2_OUT	-
PC3	UART7_TX	TI16	TO16	SPI0_MOSI
PC4	SPI0_SCK	nRTS5	FOUT0	-
PC5	UART5_TX	SPI0_MOSI	TI17	TO17
PC6	UART5_RX	SPI0_MISO	TI17	TO17
PC7	UART5_RX	SPI0_SSN	nCTS5	-
PC8	TI01	TO01	QSPI_NCS	SENT1_RX
PC9	TI02	TO02	QSPI_CLK	SENT1_TX
PC10	SCL1	QSPI_IO0	nCTS6	-
PC11	SDA1	QSPI_IO1	nRTS6	-

GPIO	PCDFS			
	00	01	10	11
PC12	TI03	TO03	QSPI_IO2	COMP1_OUT
PC13	TI04	TO04	QSPI_IO3	-
PC14	TI05	TO05	SPI1_SSN	-
PC15	TO06	TI06	UART3_RX	SPI1_MISO

2.1.8 PD 引脚组数字外设功能

GPIO	PDDFS			
	00	01	10	11
PD0	UART4_RX	TO12	TI12	-
PD1	UART4_TX	TO13	TI13	-
PD2	TO14	TI14	SENT1_RX	-
PD3	TO15	TI15	UART6_RX	-
PD4	TO16	TI16	UART6_TX	-
PD5				
PD6				
PD7				
PD8				
PD9				
PD10				

2.1.9 PE 引脚组数字外设功能

GPIO	PEDFS			
	00	01	10	11
PE2	SPI1_SCK	ATIM_CH3	-	-
PE3	SPI1_SSN	ATIM_CH3N	SPI2_SCK	-
PE4	UART1_TX	ATIM_CH4	SPI2_MOSI	-
PE5	SCL1	UART1_RX	ATIM_CH4N	SPI2_MISO
PE6	SDA1	ATIM_ETR	-	SENT0_TX
PE7	TI11	TO11	-	SENT0_RX
PE8	LPT_CH1	SPI0_MOSI	-	-
PE9	nCTS7	MCK0	-	-

2.1.10 PF 引脚组数字外设功能

GPIO	PFDFS			
	00	01	10	11
PF0	UART7_TX	LPT_CH2	-	-
PF1	-	-	-	-
PF2	-	-	-	-
PF3	UART0_RX	LPUART0_RX	SENT0_RX	-
PF4	UART0_TX	LPUART0_TX	SENT0_TX	-
PF5	TO01	TI01	-	-

GPIO	PFDFS			
	00	01	10	11
PF6	TO00	TI00	SENT0_RX	FOUT1
PF7	-	-	-	-
PF8	-	-	-	-
PF11	TO01	TI01	SENT0_TX	-
PF12	SPI0_MISO	TI12	TO12	-
PF13	SPI0_SCK	TI13	TO13	-
PF14	SPI0_SSN	nCTS4	ATIM_BKIN	-
PF15	TI14	TO14	SPI2_SCK	nRTS4

2.1.11 PG 引脚组数字外设功能

GPIO	PGDFS			
	00	01	10	11
PG0	-	-	-	-
PG1	-	-	-	-
PG2	-	-	-	-
PG3	-	-	-	-
PG4	-	-	-	-
PG5	TO00	TI00	-	-
PG6	TO11	TI11	SENT1_TX	-
PG7	TO07	TI07	UART3_TX	SPI1_MOSI
PG8	SWIO	UART0_RX	-	-
PG9	SWCLK	UART0_TX	-	-
PG10	SPI1_SCK	ATIM_BKIN2	-	-
PG11	UART7_RX	TO10	TI10	MCK1

2.1.12 封装尺寸图

2.1.12.1 LQFP100

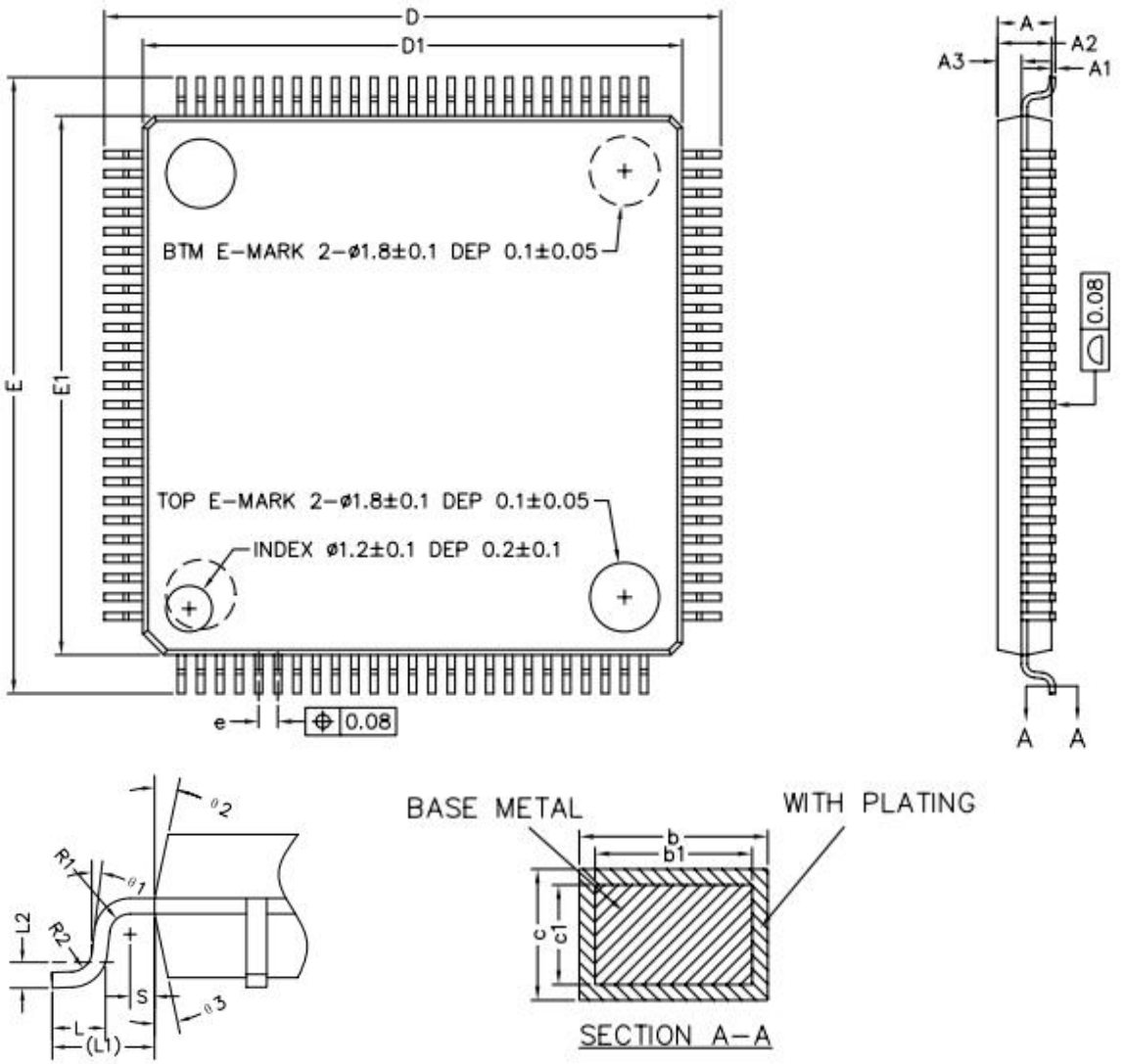


图 2-4 LQFP100 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10

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

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

2.1.12.2 LQFP64

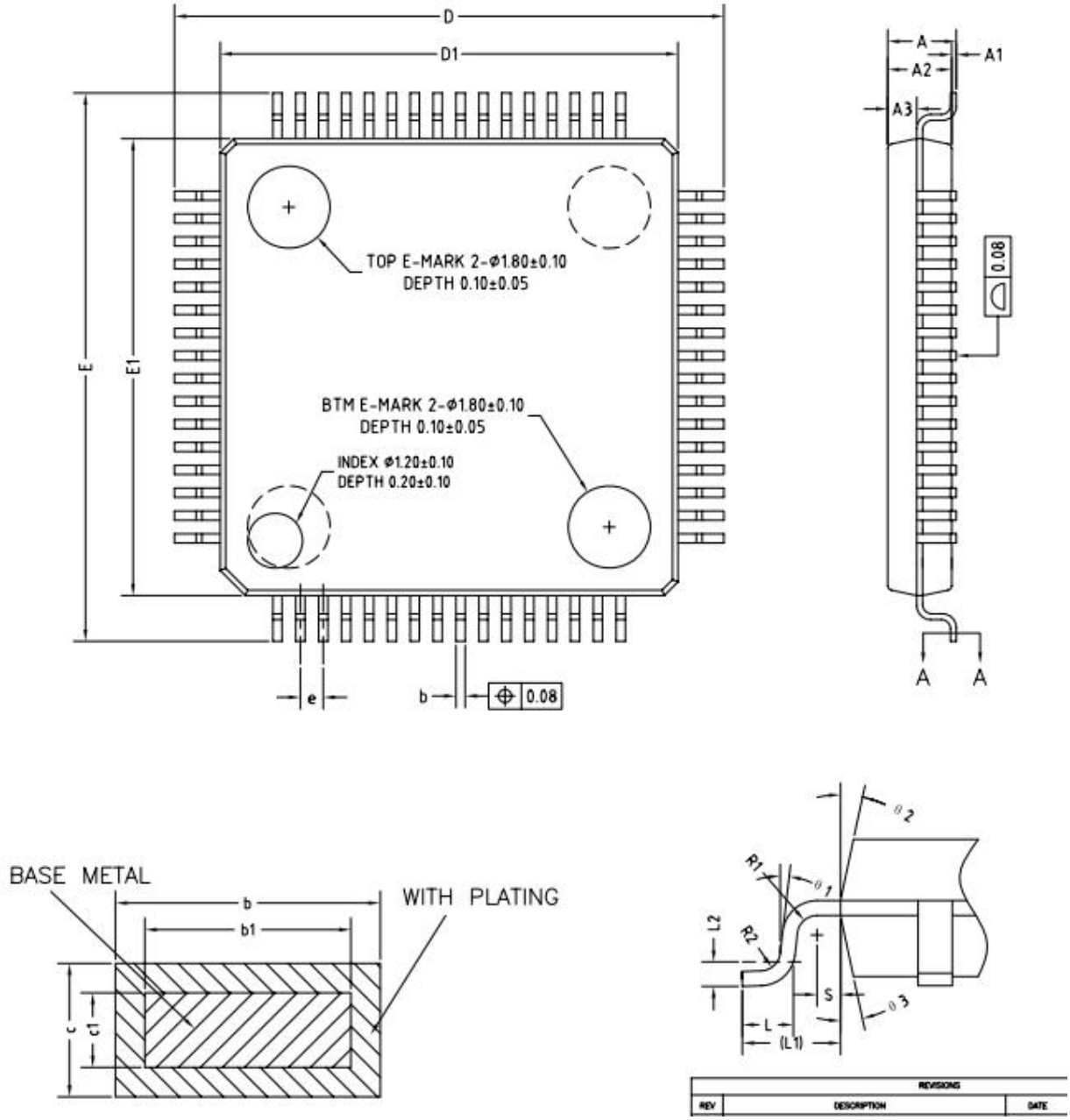


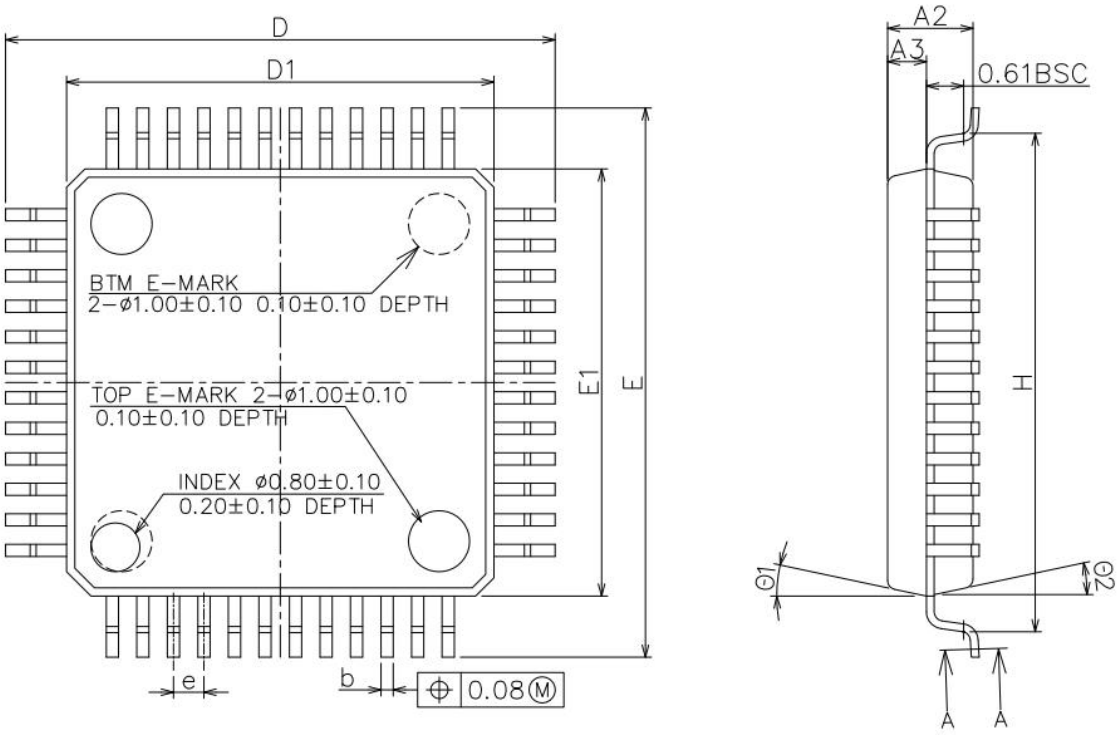
图 2-5 LQFP64 封装尺寸图

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

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

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

2.1.12.3 LQFP48



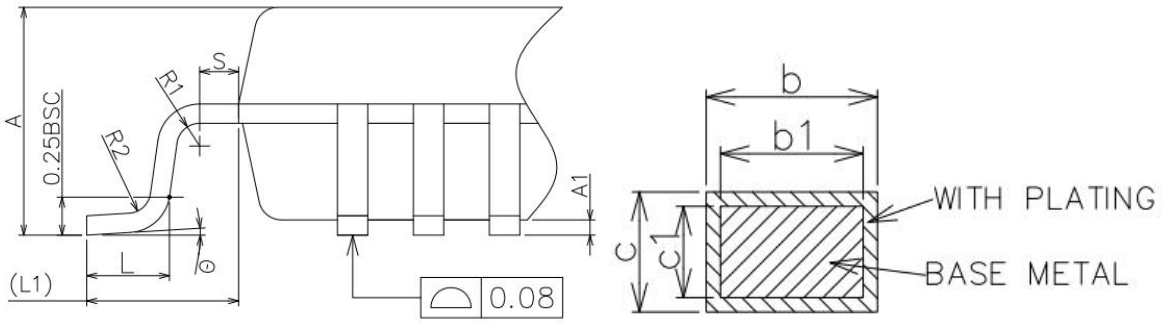


图 2-6LQFP48 封装尺寸图

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-026BDD.

3 电参数

3.1 参数说明

除非特别声明，本章节中所罗列的电参数在芯片量产测试过程中，在所标注的环境温度和供电电压下进行测试。

基于特征参数提取、设计仿真得到的参数，在表格备注中声明，这些参数不在量产测试中覆盖。

除非特别声明，typical 参数在 T_A=25C，VDD=5V 条件下，通过测试足够数量的样本获得，这些样本来自于标准量产工艺的晶圆批；typical 参数作为用户设计参考。

3.2 参数测试条件

3.2.1 供电方案

芯片量产 FT 测试时采用下图所示的电源供电方案，其中 VDD=5V。

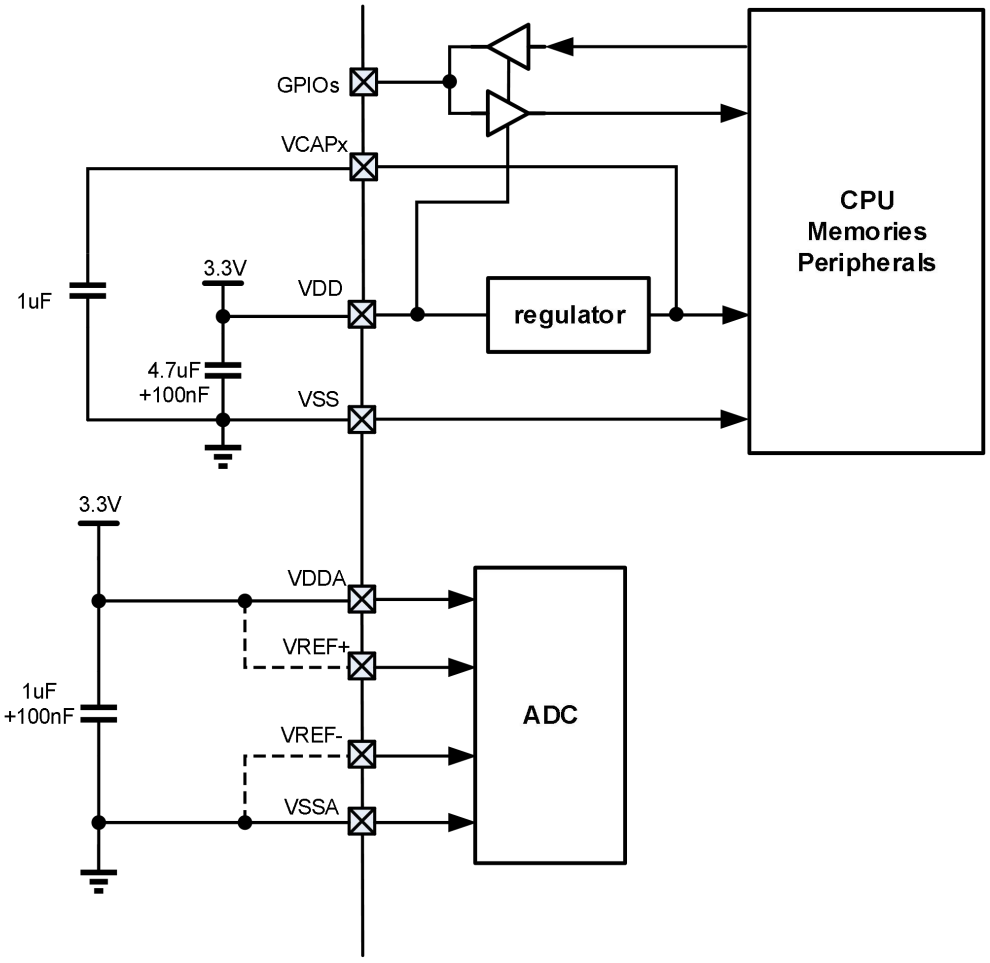


图 3-1 测试方案

注意：尽量靠近 REGC 引脚对地放置 1uF 稳压电容。

3.3 极限参数

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

Symbol	Parameter		min	max	unit
V_{DD}-V_{SS}	电源电压 （包含 VDD、VDDA）		-0.3	6.5	V
V_{PIN}	管脚电压	All others	V_{SS}-0.3	V_{DD}+0.3	V
		PB0/1	V_{SS}-0.3	6.5	V
$\|\triangle V_{DD}\|$	VDD 和 VREFP 之间的压差(1)		-	50	mV
$\|\triangle V_{SS}\|$	所有地引脚之间的压差		-	50	mV
T_A	工作温度		-40	105	℃
T_J	结温		-40	125	℃
T_{STG}	存储温度		-55	150	℃
HBM	ESD HBM 模式 T_A=25℃ 测试标准符合 JEDEC JS-001			+/-4000	V
CDM	ESD CDM 模式 T_A=25℃ 测试标准符合 JEDEC JS-002			+/-1000	V
LU	IO Latchup -(0.5VDD) < VI < (1.5VDD) T_A=25℃ 测试标准符合 JESD78E Class II			200	mA
$\sum I_{VDD}$	向芯片 VDD 流入的最大电流 (source)			100	mA
$\sum I_{VSS}$	从芯片 VSS 流出的最大电流(sink)			100	mA
$\sum I_{IO}$	所有 IO sink 的最大总和电流			100	mA
	所有 IO source 的最大总和电流			100	mA

表 3-1FM33FH0 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	T_A=-40~105℃	0	72	MHz
f_{PCLK}	APB 时钟频率		0	72	
VDD	典型工作电压范围		2.2	5.5	V
VDDA	模拟电路工作电压范围		2.2	5.5	V
Vcore	内核 LDO 输出电压 Active 模式	REGC 管脚对地连接 1uF 稳压电容	1.45	1.65	V

表 3-2FM33FH0 典型工作条件

TSI 模块工作电压范围为 2.7~5.5V.

3.4.2 功耗参数

芯片出厂时的功耗参数在环境温度下测试，高低温电流参数来自于特征参数提取。

测量功耗参数时，MCU 被配置为如下条件：

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

3.4.2.1 Active 模式功耗

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，Coremark	f_{AHB}=8MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	1.75	-	mA
			T_A=105℃	-	TBD	-	
		f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	3.23	-	
			T_A=105℃	-	TBD	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	T_A=25℃	-	4.7	-	
			T_A=105℃	-	TBD	-	
		f_{AHB}=64MHz (PLL) PLL on Flash 1 wait	T_A=25℃	-	10.1	-	
			T_A=105℃	-	TBD	-	

表 3-3ACTIVE 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_{sleep2}	Sleep 模式电流	SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持	T_A=25℃		5		uA
			T_A=105℃		55		

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	
$I_{deepsleep1}$	DeepSleep 模式电流	SVD 关闭 RTC 使用 XTLF 走时 CPU、RAM、外设数据保持	T_A=25℃		2.5		uA
			T_A=105℃		50		

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
$I_{deepsleep1}$	DeepSleep 模式电流	SVD 关闭 RTC 和 XTLF 关闭 CPU、RAM、外设数据保持	T_A=25℃		1.5		uA
			T_A=105℃		50		

表 3-5DEEPSLEEP 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			10		∞	us/V
	电源下降速度	PDR		200		∞	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]			2.0	2.1	2.2	V
V_{PDR}	低功耗下电复位电压[1] T_A=-40~105C	PDRCFG 使能		1.7	1.8	1.9	V
V_{SVD}	电压监测阈值电平 比较基准 1.0V	SVD[3:0]=0000	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0001	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0010	Fall		2.443		V
			Rise		2.543		
		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

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
			Rise		4.900		
		SVD[3:0]=1111	Fall		-		V
			Rise		-		

表 3-6 复位和电源监控参数

[1] 基于特征参数提取

3.4.4 内部基准电压

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

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

符号	参数说明	总线地址
REF_CAL	ADC 对 AVREF 输出的转换值 测试条件： T_A=30±1℃ VREFP=3.3V	0x1FFF_FB0C

内部基准源主要参数：

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
AV_{REF}	基准源输出电压[1]	-40℃≤T_A≤105℃		0.98	1.0	1.01	V
T_{setup}	内部基准源建立时间	-		-	5		us
V_{VREF_MEAS}	出厂时测量转换 AVREF 的 VREFP 电压	-		3.297	3.3	3.303	V
T_{coeff}	内部基准源温度系数	-40℃≤T_A≤105℃			25	100	ppm/℃
V_{coeff}	内部基准源电压系数	1.8≤VDDA≤5V			1	3	mV/V
T_{S_VREF}	ADC 测量 AVREF 时的采样时间	AVREF buffer 使能		10			us
T_{ADC_BUF}	驱动 ADC 输入的 VREF Buffer 的建立时间			5			us
I_{REF}	基准源工作电流	T_A = 25℃ VDDA = 3V	PTAT_EN=0		2		uA
			PTAT_EN=1		3		

表 3-7 内部基准电压参数

[1] 基于特征参数提取

3.4.5 低功耗模式唤醒时间

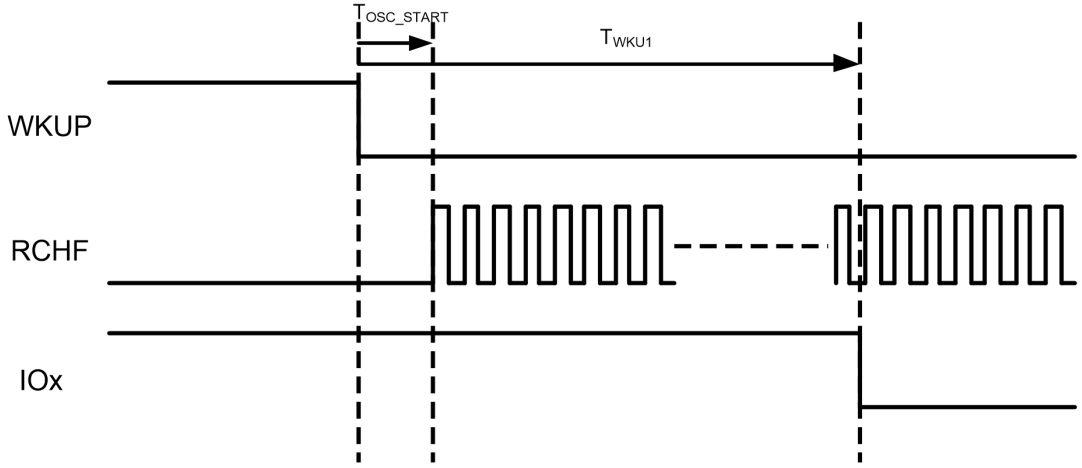
符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	

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	-	50	-	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，典型值小于 3us

T_{WKU1} 为唤醒事件到来，到程序运行后翻转 IO 的时间，典型值 5.5us。

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

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	

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率	VDD=3.3V	4	-	24	MHz
R_{fb}	反馈电阻	-	-	200	-	KΩ
VDD_{rise}	XTHF 最低工作电压 HF_CFG=0x0	8MHz	1.2	-	-	V
		16MHz,	TBD	-	-	
		24MHz	TBD	-	-	
	XTHF 最低工作电压 HF_CFG=0xF	8MHz	1.1	-	-	
		16MHz,	TBD	-	-	
		24MHz	TBD	-	-	
IDD	XTHF 工作电流 HF_CFG=0x0	8MHz	-	250	-	uA
		16MHz	-	350	-	
	XTHF 工作电流 HF_CFG=0x8	16MHz	-	560	-	
		24MHz	-	700	-	
		32MHz		900		
		40MHz		1050		
T_{start1}	XTHF 8M 起振时间	VDD=5V HF_CFG=0x0	-	11	-	ms
		VDD=5V HF_CFG=0x8	-	1.5	-	ms
T_{start2}	XTHF 16M 起振时间	VDD=5V, HF_CFG=0x8	-	5.2	-	ms
		VDD=5V, HF_CFG=0xF	-	3.3	-	ms
C_L	负载电容	-	5	-	25	pF

表 3-10 高频晶体振荡器参数

注：

[1]以上指标基于特征参数提取

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

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}	RCHF 振荡频率	VDD=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	
			FSEL==2'b11	31.84	32	32.16	
ACC_{RCHF}[1]	全温区 RCHF 变化范围	VDD=5V	FSEL==2'b00 T_A=-40~105℃	-1.5	-	1.5	%
			FSEL==2'b01 T_A=-40~105℃	-2	-	2	%
			FSEL==2'b10 T_A=-40~105℃	-3	-	3	%
			FSEL==2'b11 T_A=-40~105℃	-5	-	5	%

表 3-11 内部 RC 振荡器参数

[1]：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	T_A=25℃	30	32	34	KHz
I_{DD_RCLP}	RCLP 功耗	T_A=25℃	-	400	-	nA
ACC_{RCLP}[1]	全温区 RCLP 变化范围	T_A=-40~105℃	-10		10	%

表 3-12 内部 RC 振荡器参数

[1]：此项指标基于特征参数提取

3.4.8 PLL 特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{PLL}	PLL 输出频率	VDD=1.8~5.5V T_A=-40~105℃	32		72	MHz
I_{DD_PLL}	PLL 功耗	输出频率 32Mhz	-	700	-	uA
		输出频率 48Mhz	-	1000	-	
t_{LOCK}	PLL 锁定时间		-	65	-	us

表 3-13 PLL 参数

3.4.9 ADC 特性

3.4.9.1 参数说明

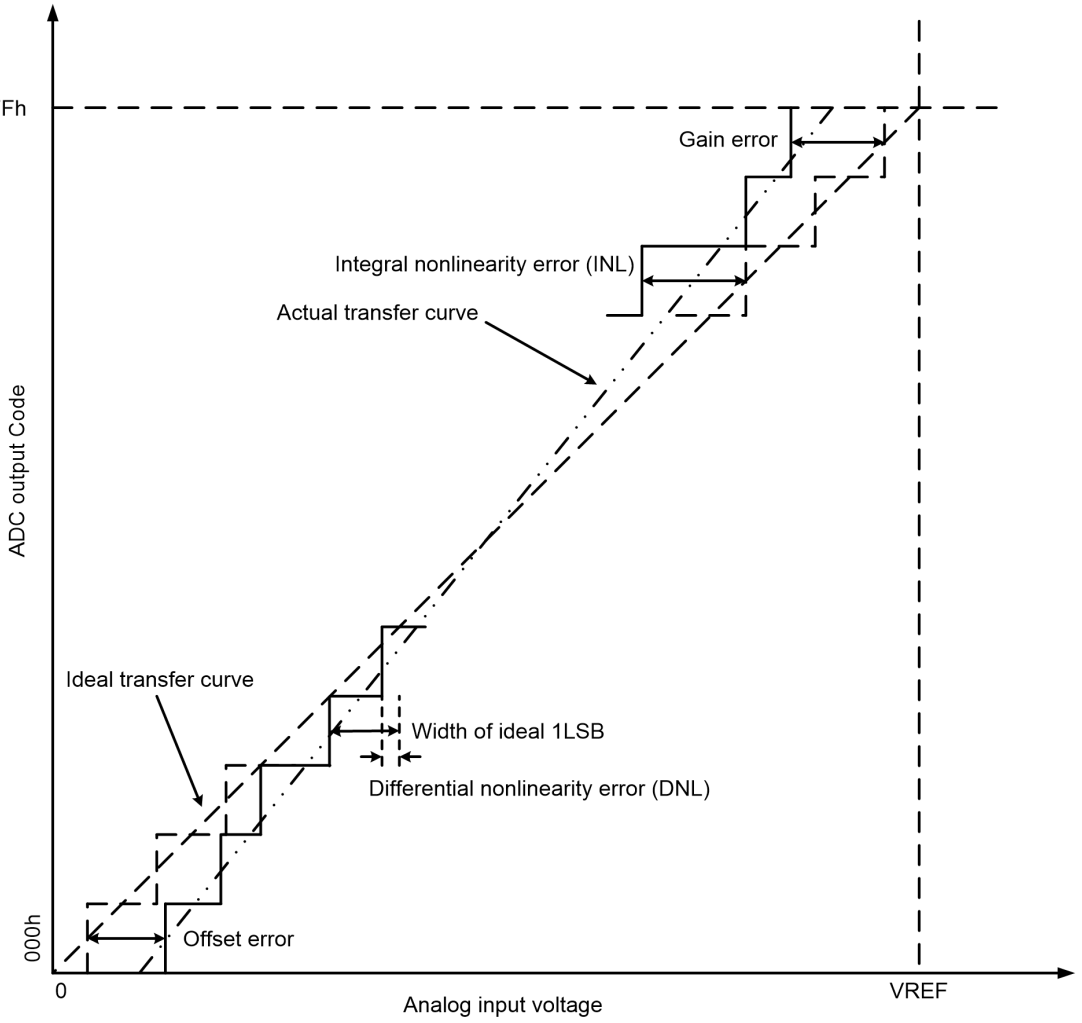


图 3-2 ADC 参数说明

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	-	2.2	-	5.5	V
VREF+	正参考电压	-	2.2	-	VDDA	V
VREF-	负参考电压	-	0	-	0.5	V
T_J	工作结温范围	-	-40	-	125	°C
V_{AIN}	输入电压范围		VREF-	-	VREF+	V
C_s	采样保持电容	-	-	5	-	pF
R_{AIN}	外部输入阻抗	-	-	-	2	KΩ
R_{ADC}	采样开关阻抗	-	-	300	-	Ω
F_{CLK}	ADC 工作时钟频率	-	-	-	32	MHz
F_S	ADC 采样频率		-	-	2	Msps
T_{SAMP}	采样保持时间	-	1	-	512	F_{CLK}
T_{CONV}	转换时间	-	-	14	-	F_{CLK}
ADC 动态性能[1]						
ENOB	VDDA=5V F_S=2Msps ADCCLK=XTHF32MHz T_{sample}=2*T_{ADCCLK} F_{AIN}=29KHz	VREF+=VDDA VREF-=VSSA	-	10.8	-	bits
	VDDA=3.3V F_S=2Msps ADCCLK=XTHF32MHz T_{sample}=2*T_{ADCCLK} F_{AIN}=29KHz	VREF+=VDDA VREF-=VSSA		10.7		bits
ADC 静态性能[1]						
DNL	差分非线性 Fs=2Msps ADCCLK=XTHF32MHz T_{sample}=2*T_{ADCCLK}	VDDA=2.5~5.5V	-0.5	-	1	LSB
INL	积分非线性 Fs=2Msps ADCCLK=XTHF32MHz T_{sample}=2*T_{ADCCLK}	VDDA=2.5~5.5V	-1.5	-	1.5	LSB
OffsetError	Offset error		-	+/-3	+/-5	LSB
GainError	Gain error	VDDA=5V	-0.2	-	0.1	%
TUE	Total unadjustable error	VDDA=5V	-	-	5.5	LSB

表 3-14 ADC 参数

[1]：此项指标基于特征参数提取

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

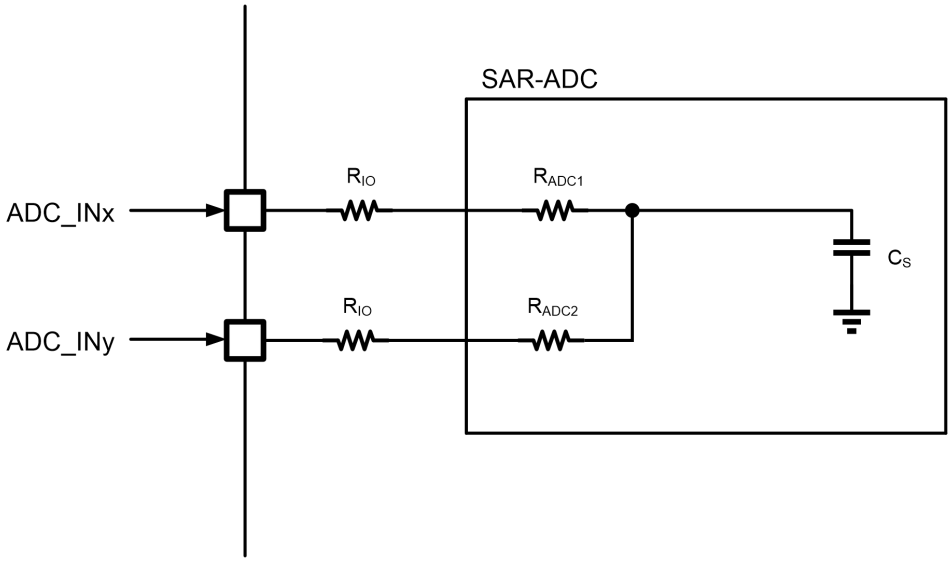


图 3-3 ADC 通道输入阻抗

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。

外部通道的 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。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
		105C	-	1250	1300	

符号	VDDA	温度	参数值			单位
			最小值	典型值	最大值	
		-40C	-	1125	1150	
	3.3V	25C	-	1250	1300	
		105C	-	1350	1450	
		-40C	-	1200	1240	
	2.5V	25C	-	1350	1450	
		105C	-	1500	1650	
		-40C	-	1250	1350	

表 3-15 ADC 采样时间

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 功耗[1] VDDA=VREF=5V 无负载	Middle code (512)		39		uA
		Max code (1023)		14		
V_{DAC}	输出电压范围[1]		0		VDDA	V
DNL	差分非线性[1]	Iload<10uA	-0.5		1	LSB
INL	积分非线性[1]		-1		1	LSB
OffsetError	失调误差[1] VDDA=VREF=5V	Code=0x000		3		mV
GainError	增益误差[1] VDDA=VREF=5V	Code=0xFFF		30		mV
t_{setup}	满幅输出建立时间	无负载			1	us

表 3-16 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 VDDA=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。

符号	参数说明	测试条件	数据保存地址
TS_CAL1	温度传感器标定值 1	VDDA=3.0V，T_A=30+/-1°C	

表 3-17 ADC 采样时间

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	

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	VDDA=5V	-	3.2	-	LSB/°C
Slope	输出斜率[1]	VDDA=5V	-	3.9	-	mV/°C
Linerity	全温区线性度[1]		-	+/-2		°C
I_{DDA}	温度传感器功耗（不含ADC）[2]			1	2	uA
t_{START}	温度传感器启动时间[2]				10	us
t_{SAMPLE}	ADC 采样温度传感器输出时要求的采样时间[2]		10	-	-	us

表 3-18 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。

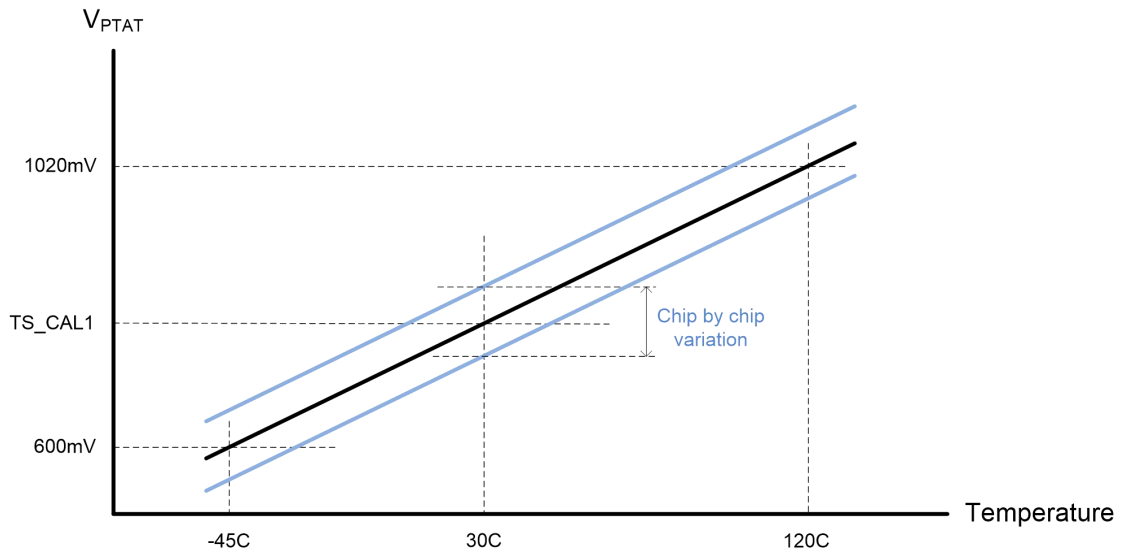


图 3-4 温度传感器输出曲线

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 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	-		2.2		5.5	V
VI_{comp1}	比较器输入电压范围	-		0	-	VDDA	V
I_{comp1}	比较器工作电流	VDDA=5V 50Khz 方波，±100mV overdrive	低功耗模式	-	-	2	uA
			高速模式	-	-	15	
$T_{propagation1}$	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	-	0.2	-	us
T_{setup1}	比较器建立时间	VDDA=5V	低功耗模式	-	2	-	us
			高速模式	-	1.5	-	

表 3-19 模拟比较器参数

3.4.13 Flash 存储器特性

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	Code Flash size			-	384K	bytes
	Data Flash size			-	16K	bytes
T_{PROG}	Byte Program Time		6	-	7.5	μs
T_{ERASE}	Sector/Block Erase		4	-	5	ms
	Chip Erase		30	-	40	ms
N_{ED}	Sector Endurance	code flash	-	100,000	-	Erase/Write cycles
		data flash	-	500,000	-	Erase/Write cycles
T_{DR}	Data Retention	$T_{A(average)}$=85℃	10			yrs

表 3-20 Flash 参数

3.4.14 AC 参数

3.4.14.1 SPI

除非特别声明，以下参数测试条件为：

- 测试温度：常温
- IO 驱动强度 DS=10
- 容性负载 C_L=25pF
- 信号翻转测量点：0.5VDD

- VDD=5V, or 3.3V

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tc(SCK)		Master mode	60	-	-	ns
tr(SCK)		Master mode	-	5	-	ns
tf(SCK)		Master mode	-	5	-	ns
tsu(SSN)		Slave mode	50	-	-	ns
th(SSN)		Slave mode	1	-	-	ns
tsu(MI)	Data input setup time	Master mode	21	-	-	ns
tsu(SI)		Slave mode	1	-	-	ns
th(MI)	Data input hold time	Master mode	0	-	-	ns
th(SI)		Slave mode	5	-	-	ns
ta(SO)	Data output access time	Slave mode	50	-	-	ns
tdis(SO)	Data output disable time	Slave mode	18	-	-	ns
tv(SO)	Data output valid time	Slave mode	-	-	50	ns
tv(MO)		Master mode	-	-	4	ns
th(SO)	Data output hold time	Slave mode	50	-	-	ns
th(MO)		Master mode	3	-	-	ns

表 3-21 SPI timing characteristics at VDD=5V

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tc(SCK)		Master mode	60	-	-	ns
tr(SCK)		Master mode	-	5	-	ns
tf(SCK)		Master mode	-	5	-	ns
tsu(SSN)		Slave mode	55	-	-	ns
th(SSN)		Slave mode	1	-	-	ns
tsu(MI)	Data input setup time	Master mode	21	-	-	ns
tsu(SI)		Slave mode	1	-	-	ns
th(MI)	Data input hold time	Master mode	0	-	-	ns
th(SI)		Slave mode	5	-	-	ns
ta(SO)	Data output access time	Slave mode	55	-	-	ns
tdis(SO)	Data output disable time	Slave mode	22	-	-	ns
tv(SO)	Data output valid time	Slave mode	-	-	55	ns
tv(MO)		Master mode	-	-	4	ns
th(SO)	Data output hold time	Slave mode	55	-	-	ns
th(MO)		Master mode	3	-	-	ns

表 3-22 SPI timing characteristics at VDD=3.3V

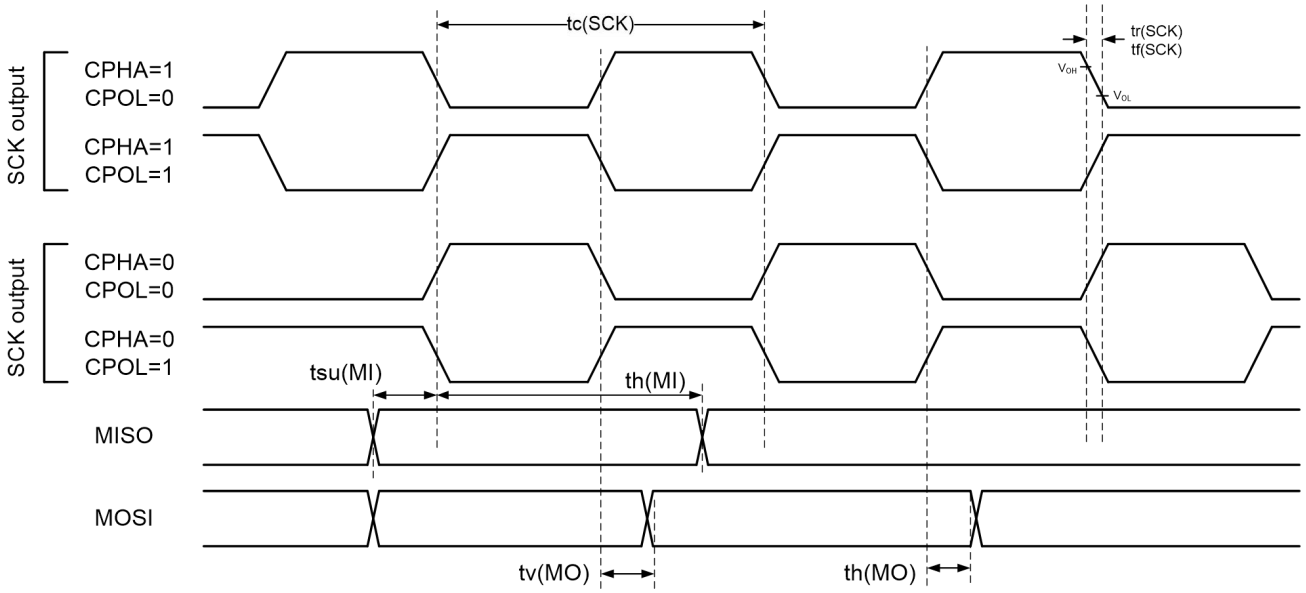


图 3-5SPI master timing

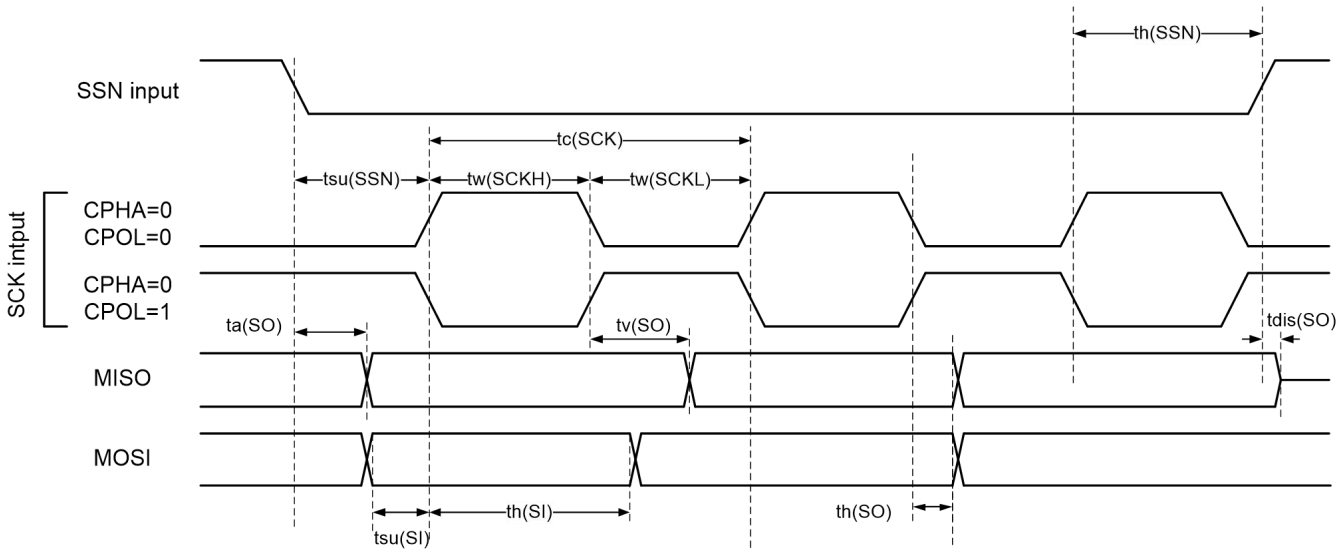


图 3-6SPI slave timing, CPHA=0

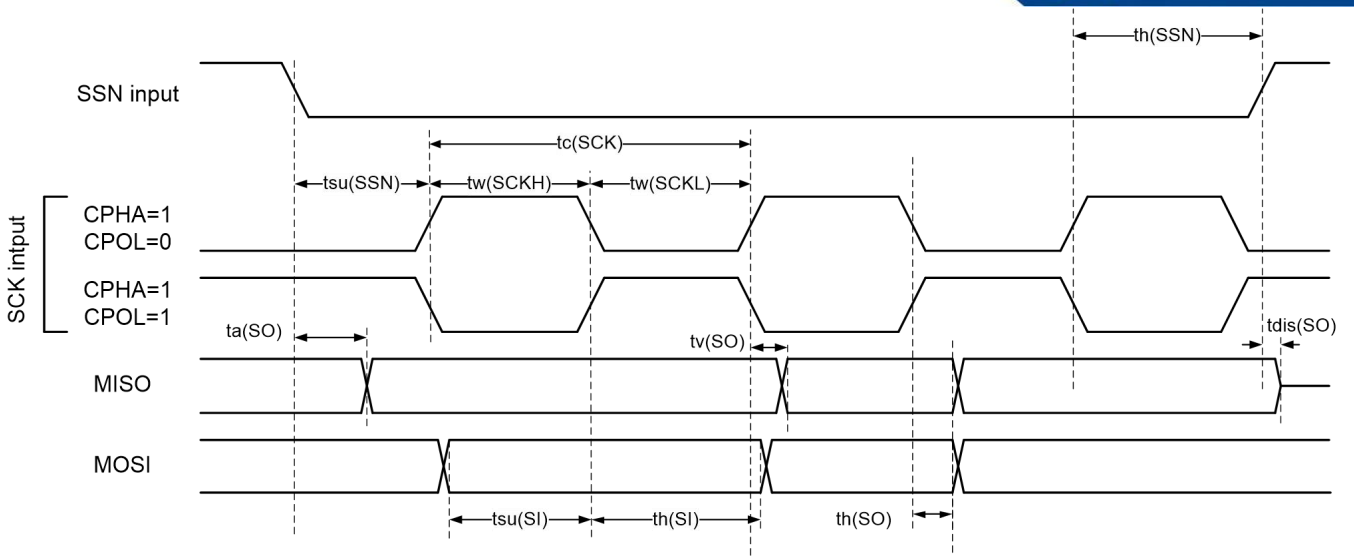


图 3-7SPI slave timing, CPHA=1

3.4.14.2 QSPI

除非特别声明，以下参数测试条件为：

- 测试温度：常温
- IO 驱动强度 DS=10
- 容性负载 C_L=25pF
- 信号翻转测量点：0.5VDD
- VDD=5V, or 3.3V

	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tc(SCK)			TBD	-	-	ns
tr(CLK)		10%VDD->90%VDD	-	TBD	-	ns
tf(CLK)		90%VDD->10%VDD	-	TBD	-	ns
tsu(IN)	Data input setup time		TBD	-	-	ns
th(IN)	Data input hold time		TBD	-	-	ns
tv(OUT)	Data output valid time		-	-	TBD	ns
th(OUT)	Data output hold time		TBD	-	-	ns

表 3-23 QSPI timing characteristics at VDD=5V

	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tc(SCK)			TBD	-	-	ns
tr(CLK)		10%VDD->90%VDD	-	TBD	-	ns
tf(CLK)		90%VDD->10%VDD	-	TBD	-	ns

	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tsu(IN)	Data input setup time		TBD	-	-	ns
th(IN)	Data input hold time		TBD	-	-	ns
tv(OUT)	Data output valid time		-	-	TBD	ns
th(OUT)	Data output hold time		TBD	-	-	ns

表 3-24 QSPI timing characteristics at VDD=3.3V

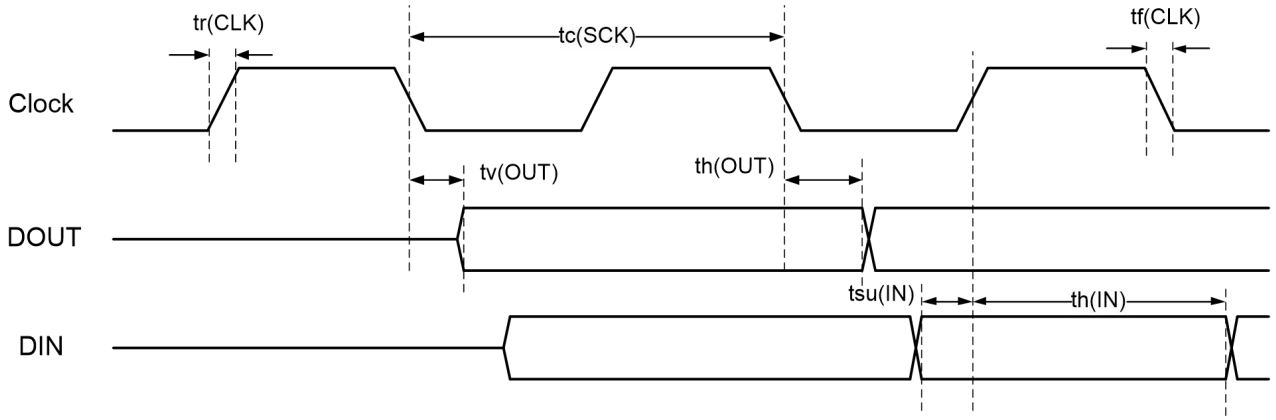


图 3-8QSPI timing

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		-0.1	-	0.1	μA
I_{IH}	输入高漏电	V_{IH}=5V		-0.1	-	0.1	μA
V_{OL}	输出低电平	V_{DD}=5V ISINK=5mA	DS=01	-	0.45	-	V
			DS=10		0.18		
			DS=11		0.15		
		V_{DD}=5V ISINK=10mA	DS=01	-	1.1	-	
			DS=10		0.37		
			DS=11		0.3		
		V_{DD}=5V ISINK=20mA	DS=01	-	禁止	-	
			DS=10		0.8		
			DS=11		0.65		
		V_{DD}=3.3V ISINK=5mA	DS=01	-	0.7	-	
			DS=10		0.24		

			DS=11		0.2		
		V_{DD}=3.3V ISINK=10mA	DS=01	-	禁止	-	
			DS=10		0.5		
			DS=11		0.4		
		V_{DD}=3.3V ISINK=20mA	DS=01	-	禁止	-	
			DS=10		禁止		
			DS=11		0.9		
V_{OH}	输出高电平	V_{DD}=5V ISINK=5mA	DS=01	-	4.3	-	V
			DS=10		4.75		
			DS=11		4.8		
		V_{DD}=5V ISINK=10mA	DS=01	-	3.3	-	
			DS=10		4.5		
			DS=11		4.6		
		V_{DD}=5V ISINK=20mA	DS=01	-	禁止	-	
			DS=10		3.9		
			DS=11		4		
		V_{DD}=3.3V ISINK=5mA	DS=01	-	2.2	-	
			DS=10		2.9		
			DS=11		3.05		
		V_{DD}=3.3V ISINK=10mA	DS=01	-	禁止	-	
			DS=10		2.55		
			DS=11		2.7		
		V_{DD}=3.3V ISINK=20mA	DS=01	-	禁止	-	
			DS=10		禁止		
			DS=11		2		
R_{PU}	弱上拉电阻			-	50	-	KΩ
R_{PD}	弱下拉电阻			-	50	-	KΩ

表 3-25 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
I_{IL}	输入低漏电	V_{IL}=0V	-0.1		0.1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-0.1		0.1	μA
R_{PU}	上拉电阻			10		KΩ
$T_{AFILTER}$	模拟滤波长度[1]	VDD=5V		100		ns
$T_{DFILTER}$	数字滤波长度[1]	VDD=1.8~5.5V -40℃≤T_A≤105℃	50		100	us

表 3-26 NRST 引脚参数

注：

[1] 此项参数基于特征参数提取

GPIO AC 特性

IO	符号	参数说明[1]	测试条件		typ	max	单位
GPIO	Fmax	Maximum frequency	C_L=15pF, VDD=5V	DS=01	-	20	MHz
				DS=10	-	35	
				DS=11	-	45	
			C_L=15pF, VDD=3.3V	DS=01	-	20	
				DS=10	-	35	
				DS=11	-	35	
			C_L=50pF, VDD=5V	DS=01	-	12	
				DS=10	-	30	
				DS=11	-	35	
			C_L=50pF, VDD=3.3V	DS=01	-	10	
				DS=10	-	30	
				DS=11	-	30	
	Tr	Output rise time	C_L=15pF, VDD=5V	DS=01	10	-	ns
				DS=10	2.9	-	
				DS=11	3.6	-	
			C_L=15pF, VDD=3.3V	DS=01	13.5	-	
				DS=10	5.2	-	
				DS=11	3.1	-	
			C_L=50pF, VDD=5V	DS=01	24.4	-	
				DS=10	9.5	-	
				DS=11	7.9	-	
			C_L=50pF, VDD=3.3V	DS=01	32.4	-	
				DS=10	10.5	-	
				DS=11	10.1	-	
	Tf	Output fall time	C_L=15pF, VDD=5V	DS=01	8.7	-	ns
				DS=10	2.8	-	
				DS=11	3.6	-	
			C_L=15pF, VDD=3.3V	DS=01	10.7	-	
				DS=10	4.2	-	
				DS=11	3.0	-	
			C_L=50pF, VDD=5V	DS=01	21.2	-	
				DS=10	8.5	-	
				DS=11	7.1	-	
			C_L=50pF, VDD=3.3V	DS=01	27.2	-	
				DS=10	9.8	-	
				DS=11	8.2	-	

表 3-27 引脚 AC 参数

注：

[1] 基于特征参数提取，不在量产测试中测试

[2] 测试条件：duty cycle 45%-55%, Tr+Tf < (2/3)*Tcycle, Tr/Tf 定义参见下图

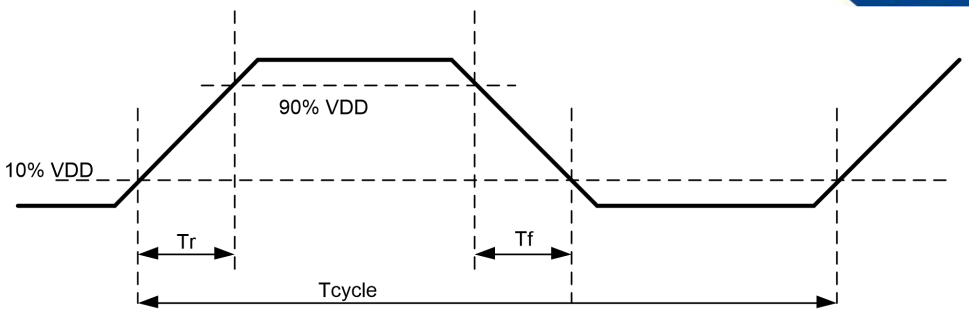


图 3-9IO timing charateristics

4 总线与存储

4.1 系统总线

FM33FH0xx 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

FM33FH0 的系统总线示意图如下，包含一条 AHB-Lite 总线、一条 APB 总线。

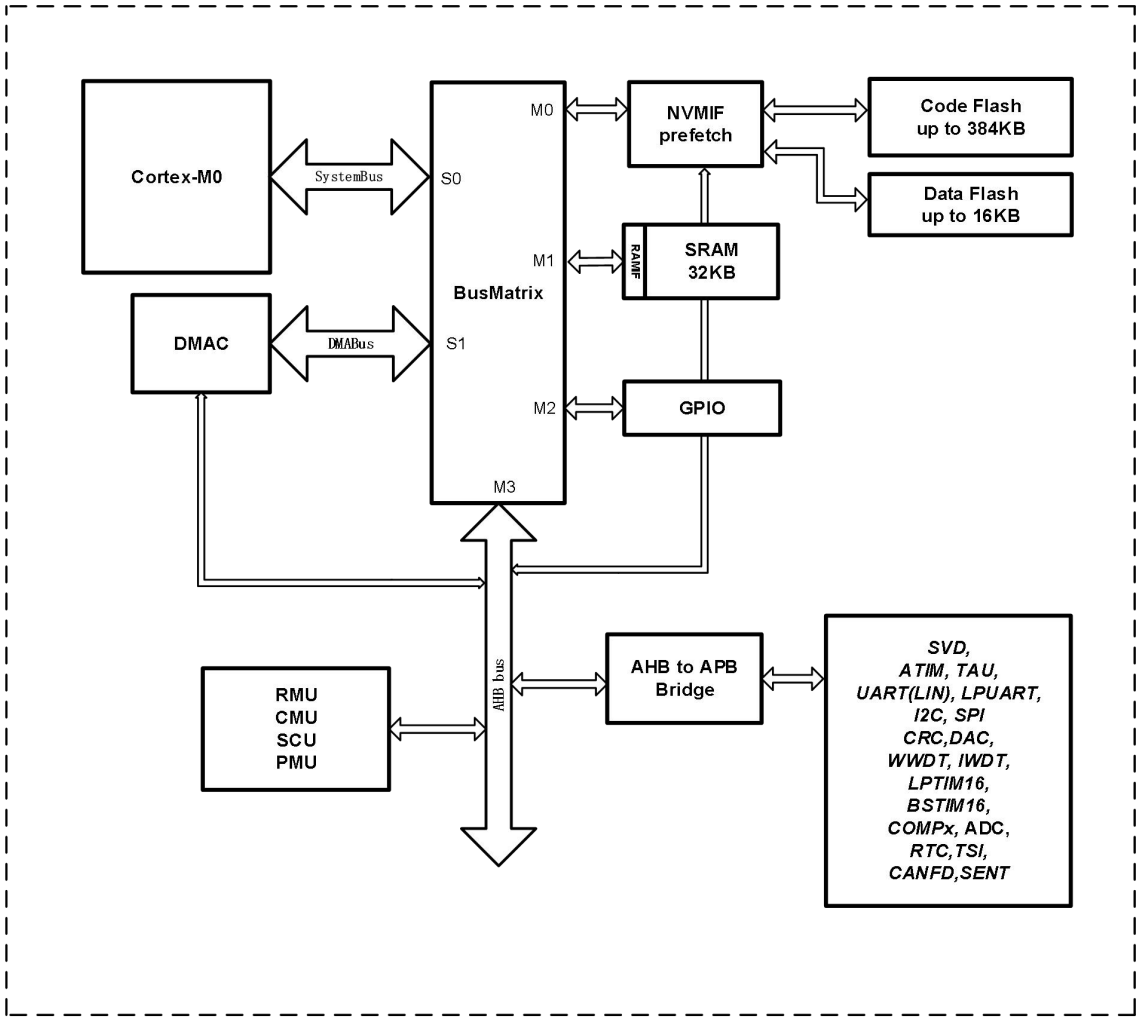


图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

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

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，FM33FH0 的地址空间分配如下图（384KB code Flash，16KB data flash, 32KB RAM）：

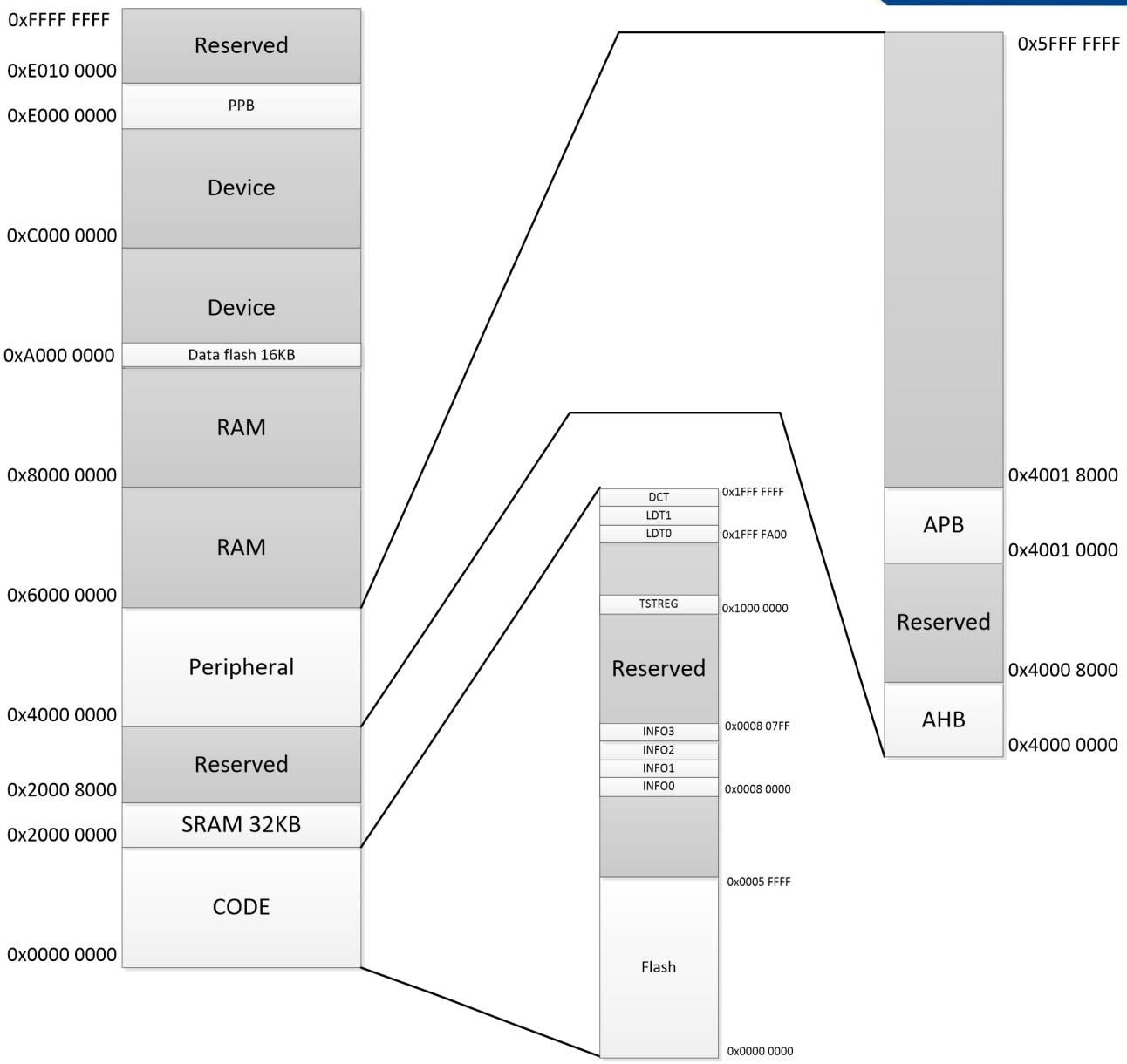


图 4-2FM33FH0 总线地址（384+32+16）

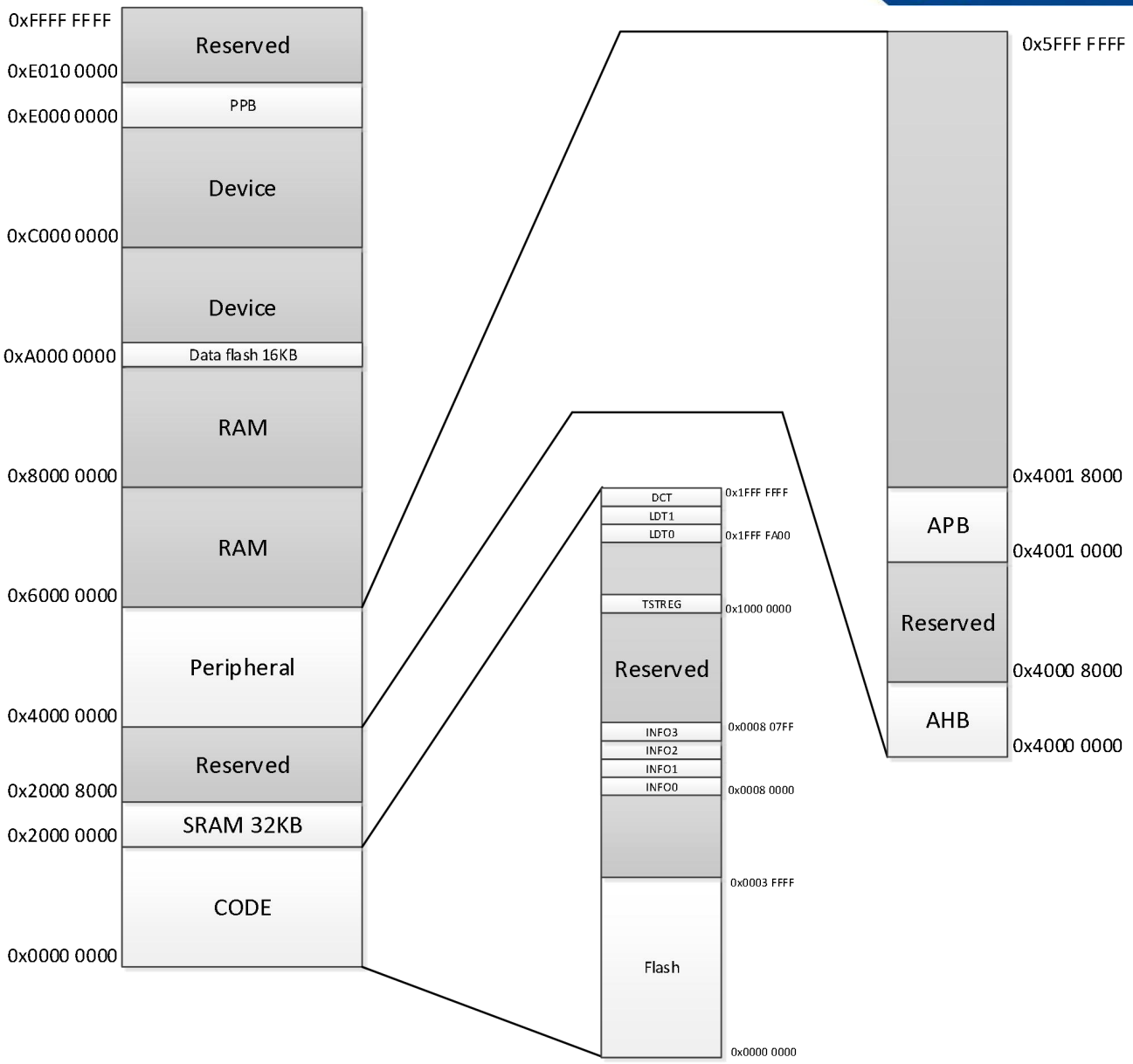


图 4-3FM33FH0 总线地址（256+32+16）

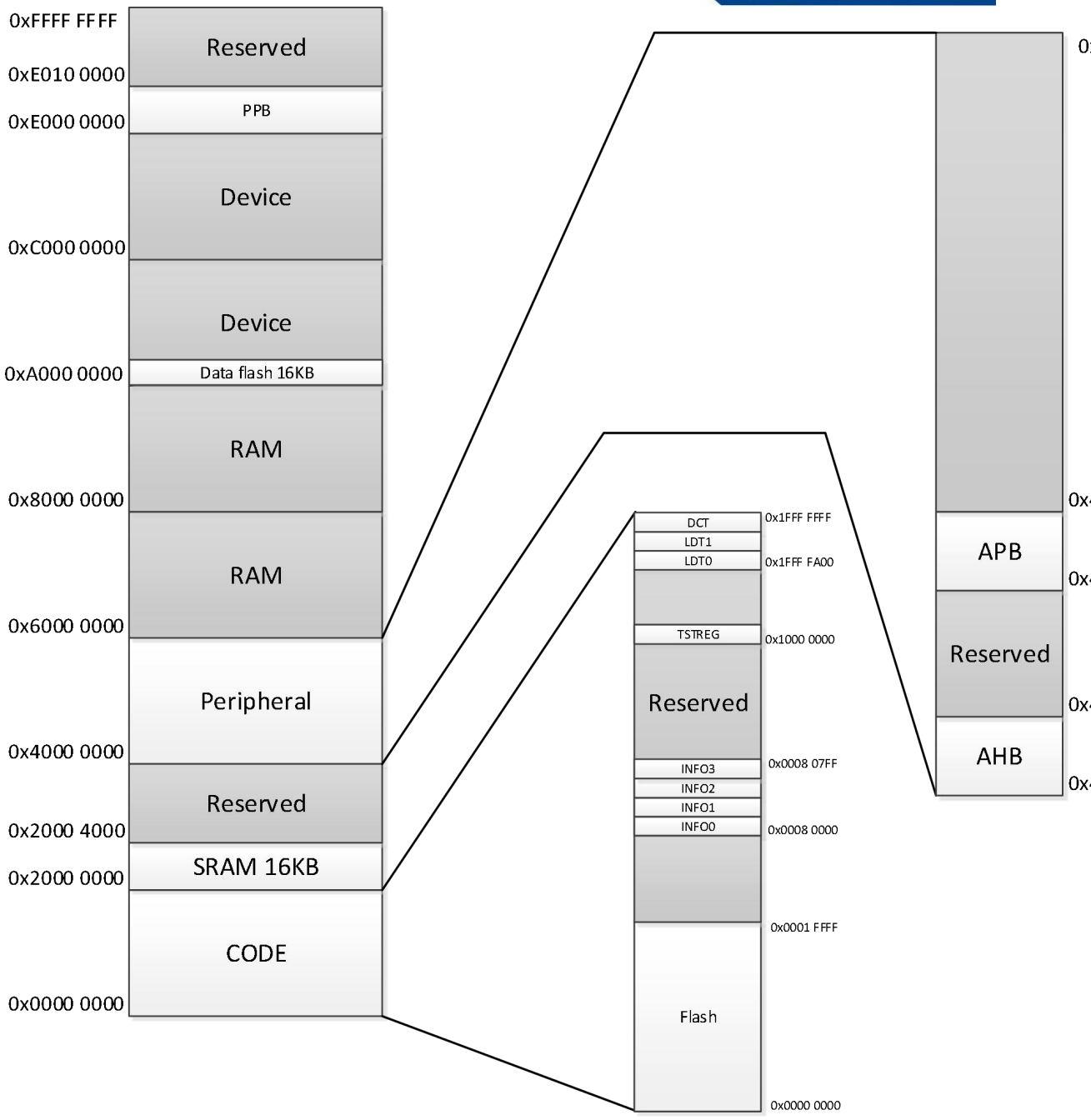


图 4-4FM33FH0 总线地址（128+16+16）

4.2.2 外设模块寄存器地址分配

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

总线	起始地址	空间	外设
AHB	0x0000_0000	384KB	Flash main array
	0x1FFF_F000	4KB	Flash Option cell array
	0x2000_0000	32KB	SRAM
	0x2000_8000	-	-
	0x4000_0000	1KB	SCU
	0x4000_0400	1KB	DMA

	0x4000_0800	1KB	QSPI_REG
	0x4000_0C00	1KB	GPIO
	0x4000_1000	1KB	NVMIF
	0x4000_1400	-	-
	0x4000_1800	-	-
	0x4000_1C00	1KB	SFU
	0x4000_2000	1KB	PMU
	0x4000_2400	1KB	CMU
	0x4000_2800	1KB	RMU
	0x4001_0000	64KB	APB
	-		
	-		
	0x9000_0000	256MB	QSPI-XIP space
	0xA000_0000	16KB	Data flash
APB1	0x4001_0000	1KB	TSI
	0x4001_0400	1KB	LPUART
	0x4001_0800	1KB	-
	0x4001_0C00	1KB	-
	0x4001_1000	1KB	RTC
	0x4001_1400	1KB	IWDT
	0x4001_1800	1KB	WWDT
	0x4001_1C00	1KB	UART0
	0x4001_2000	1KB	UART1
	0x4001_2400	1KB	-
	0x4001_2800	1KB	-
	0x4001_2C00	1KB	DAC
	0x4001_3000	1KB	-
	0x4001_3400	1KB	LPTIM
	0x4001_3800	1KB	-
	0x4001_3C00	1KB	TAU
	0x4001_4000	1KB	-
	0x4001_4400	1KB	
	0x4001_4800	1KB	SENT0
	0x4001_4C00	1KB	SENT1
	0x4001_5000	1KB	I2C0
	0x4001_5400	1KB	I2C1
	0x4001_5800	1KB	-
	0x4001_5C00	1KB	-
	0x4001_6000	1KB	INTMUX
	0x4001_6400	1KB	
	0x4001_6800	1KB	
	0x4001_6C00	1KB	
	0x4001_7000	1KB	
	0x4001_7400	1KB	UART2
	0x4001_7800	1KB	UART3
	0x4001_7C00	1KB	UARTIR
	0x4001_8000	1KB	CRC
	0x4001_8400	1KB	-
	0x4001_8800	1KB	-
	0x4001_8C00	1KB	SPI1
	0x4001_9000	1KB	SPI0
	0x4001_9400	1KB	-
	0x4001_9800	1KB	ANTEST
	0x4001_9C00	1KB	COMPARATOR
	0x4001_A000	1KB	UART4

	0x4001_A400	1KB	UART5
	0x4001_A800	1KB	SVD
	0x4001_AC00	1KB	ADC
	0x4001_B000	1KB	ATIM
	0x4001_B400	1KB	BSTIM
	0x4001_B800	1KB	UART6
	0x4001_BC00	1KB	UART7
	0x4001_C000	1KB	CLM
APB2	0x4002_0000	16KB	-
	0x4002_4000	16KB	-
	0x4002_8000	1KB	-
	0x4002_8400	1KB	
	0x4002_8800	1KB	

表 4-1 外设模块总线地址列表

4.3 RAM

4.3.1 概述

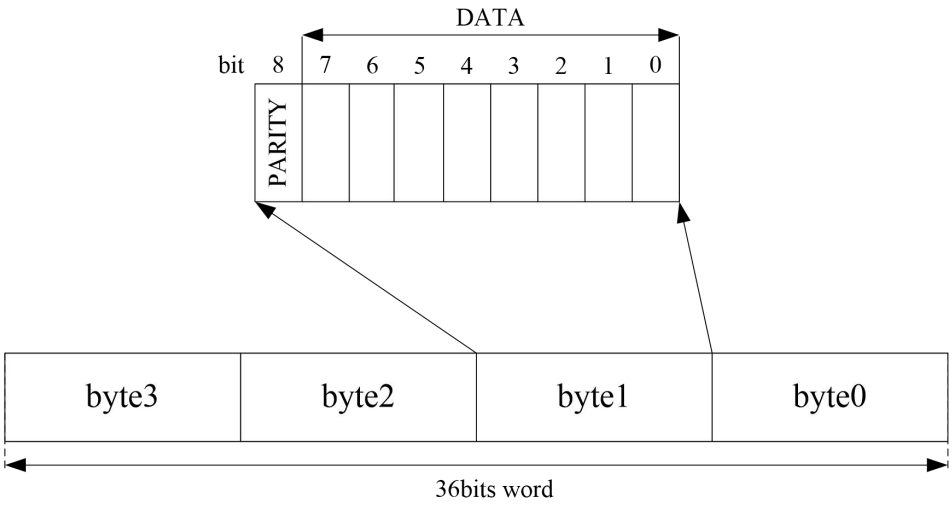
FM33FH0 含有一块 32KB RAM，SRAM 地址空间范围是 0x2000_0000~0x2000_7FFF，软件可以对 SRAM 进行字节、半字、字访问，CPU 和 DMA 都可以以最大系统频率对 SRAM 实现无等待的单周期读写。CPU 也可以从 SRAM 取指执行程序，因此在对程序效率要求高的场合，可以将部分代码导入 SRAM 中，实现最高频率下无等待的执行。

4.3.2 奇偶校验

FM33FH0 的 SRAM 支持硬件奇偶校验，每字节带有 1bit 校验位，以满足 IEC60730 标准要求的 RAM 检验功能。

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

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



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

4.4 Flash

4.4.1 概述

FM33FH0 集成两块独立的 flash：384KB 程序 flash 和 16KB 数据 flash

对程序 Flash 的擦写操作不会影响 data flash，可以实现全擦程序的同时不擦除用户数据。

4.4.2 数据 flash

16KB 数据 flash 与程序 flash 完全独立，擦写寿命可达 50 万次，读取周期为 80ns。CPU 可以同时访问 code flash 和 data flash。CPU 对 data flash 只能进行数据访问，不能执行取指操作（XN 属性），对 data flash 的取指操作将触发 hardfault。

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

注意 data flash 被逻辑映射到 0xA000 0000 位置。

4.4.3 ECC

Code 和 data flash 都支持 ECC，ECC 算法为纠 1 检 2（SEC-DED），以 32bit 数据为单位，能够自动纠正 1bit 错误，检测并报警 2bit 错误，因此可以有效避免偶发的单 bit 数据错误，特别适合需要功能安全（比如 IEC60730 SIL3），或者对 flash 数据可靠性要求高的场合。

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

4.4.4 特殊信息扇区说明

Flash IP 共有 4 个特殊扇区，说明如下

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

4.4.4.1 LDT0 page

LDT0为FMSH配置信息区，仅能在制造商模式下改写，用户模式下只读。

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

AHB地址	物理地址 LDTEN[0]=1	[31:16]	[15:0]	说明
0x1FFF_FB08	0x0042	~AVREFCAL	AVREFCAL	3V,30C 下ADC对AVREF的转换值
0x1FFF_FB10	0x0044	~TS_CAL	TS_CAL	3V, 30C下ADC对温度传感器的转换值
0x1FFF_FB20	0x0048	~RCLP_TRIM	RCLP_TRIM	RCLP调校值
0x1FFF_FB34	0x0053	~RCHF32TRIM	RCHF32TRIM	RCHF 32MHz调校值
0x1FFF_FB38	0x0054	~RCHF24TRIM	RCHF24TRIM	RCHF 24MHz调校值
0x1FFF_FB3C	0x0055	~RCHF16TRIM	RCHF16TRIM	RCHF 16MHz调校值
0x1FFF_FB40	0x0056	~RCHF8TRIM	RCHF8TRIM	RCHF 8MHz 调校值(auto-load)

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

详细参数格式定义如下：

助记符	[31:16]	[15:0]	功能描述
VREFCAL	{4'h0, ~VREFCAL}	{4'hF, VREFCAL}	VDDA=VREFP=3V+/-10mV, 30C+/-1C 下ADC对AVREF的转换值
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0]表示8bit调校值
RCHF32TRIM	{9'b0000_0000_0, ~RCHFtrim[6:0]}	{9'b1111_1111_1,RCHFtrim[6:0]}	RCHFtrim[6:0]表示7bit调校值
RCHF24TRIM			
RCHF16TRIM			
RCHF8TRIM			

4.4.4.2 LDT1 page

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

LDT1的总线地址是0x1FFF_FC00~0x1FFF_FDFF；

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

OPTBYTES 选项字节定义如下：

Bitfield	助记符	功能描述	出厂默认
31:24	IWDTEN	复位后 IWDT 默认状态 0x55：使能 IWDT 其他：关闭 IWDT	0xFF
23:20	IWDTSLP	配置 IWDT 在低功耗模式下是否允许停止计数 0xA：在 Sleep 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
7:4	DBRDPEN_D	data flash 调试接口访问保护使能 0xA：关闭调试接口保护 其他：使能调试接口保护	0xA
3:0	DBRDPEN_C	code flash 调试接口访问保护使能 0xA：关闭调试接口保护 其他：使能调试接口保护	0xA

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

LOCK 配置字节定义如下：

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

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

LOCK bit 和 Flash 权限锁定地址的对应关系如下表：

Address	LOCK bits
0x0000_0000 ~ 0x0000_1FFF	LOCK1[1:0]
0x0000_2000 ~ 0x0000_3FFF	LOCK1[3:2]
0x0000_4000 ~ 0x0000_5FFF	LOCK1[5:4]
0x0000_6000 ~ 0x0000_7FFF	LOCK1[7:6]
0x0000_8000 ~ 0x0000_9FFF	LOCK1[9:8]
0x0000_A000 ~ 0x0000_BFFF	LOCK1[11:10]
0x0000_C000 ~ 0x0000_DFFF	LOCK1[13:12]
0x0000_E000 ~ 0x0000_FFFF	LOCK1[15:14]
0x0001_0000 ~ 0x0001_1FFF	LOCK1[17:16]
0x0001_2000 ~ 0x0001_3FFF	LOCK1[19:18]
0x0001_4000 ~ 0x0001_5FFF	LOCK1[21:20]
0x0001_6000 ~ 0x0001_7FFF	LOCK1[23:22]
0x0001_8000 ~ 0x0001_9FFF	LOCK1[25:24]
0x0001_A000 ~ 0x0001_BFFF	LOCK1[27:26]
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]
0x0002_0000 ~ 0x0002_1FFF	LOCK2[1:0]
0x0002_2000 ~ 0x0002_3FFF	LOCK2[3:2]
0x0002_4000 ~ 0x0002_5FFF	LOCK2[5:4]
0x0002_6000 ~ 0x0002_7FFF	LOCK2[7:6]
0x0002_8000 ~ 0x0002_9FFF	LOCK2[9:8]
0x0002_A000 ~ 0x0002_BFFF	LOCK2[11:10]
0x0002_C000 ~ 0x0002_DFFF	LOCK2[13:12]
0x0002_E000 ~ 0x0002_FFFF	LOCK2[15:14]
0x0003_0000 ~ 0x0003_1FFF	LOCK2[17:16]
0x0003_2000 ~ 0x0003_3FFF	LOCK2[19:18]
0x0003_4000 ~ 0x0003_5FFF	LOCK2[21:20]
0x0003_6000 ~ 0x0003_7FFF	LOCK2[23:22]

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

4.4.4.3 Information1~3 page

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

IF1~3的总线地址是0x0008_0200 ~ 0x0008_07FF

4.4.4.4 Information0 page（OTP）

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

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

对IF0 page的读取没有任何限制。

4.4.5 Flash 编程

4.4.5.1 概述

FM33FH0 支持以下 Flash 编程方法：

- 在系统编程（ISP）：通过 FMSH 专用编程器或者 KEIL 用户界面实施芯片编程，使用 SWD 接

口

- 在应用编程（IAP）：通过 bootloader 代码实现芯片自编程，用户可定义任意串口，可用于实现程序在线升级

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

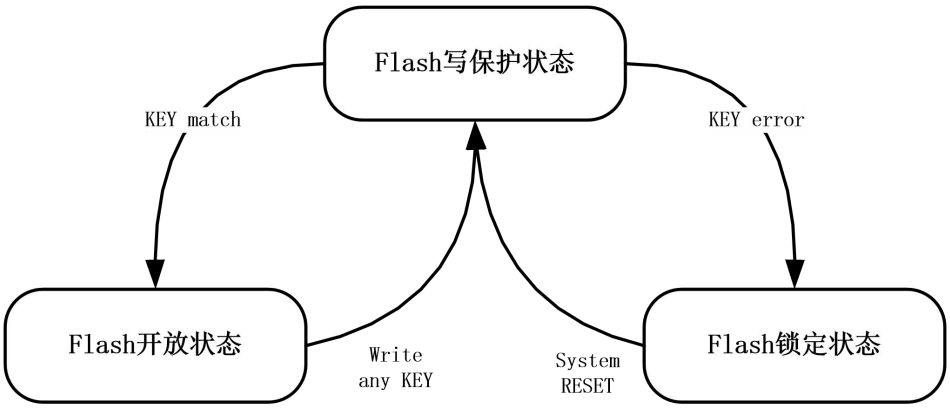
4.4.5.2 Flash 擦写时钟

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。

4.4.5.3 Flash 擦写方法

FM33FH0 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。

4.4.5.4 全擦操作（Matrix Erase）

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

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D

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

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

4.4.5.5 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 0x3C3C_3C3C
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

4.4.5.6 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

4.4.5.7 单次编程

单次编程由软件发起，操作流程如下：

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

注:相同的word地址不能被多次连续编程，如果待编程的地址已经被编程，编程前必须擦除。

4.4.5.8 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
- 置位连续编程使能寄存器（DMA 模式使能）
- 向 RAM 中写入 256 字节待编程数据
- 配置 DMA 存储器通道，设定传输方向、读地址和写地址
- 使能 DMA 存储器通道
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1
- 软件触发 DMA 存储器通道，DMA 连续 64 次读取 RAM 并对 Flash 编程
- NVMIF 检查被编程扇区是否被 ACLOCK 锁定，如果锁定则触发错误中断并通知 DMA 停止编程
- 256 字节完全编程结束后产生中断，释放 Flash 总线
- 软件确认编程结束后向 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。

4.4.6 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。

4.4.6.1 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 NVR0 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。

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

4.4.6.2 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 32 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，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。

4.4.6.3 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]	Last byte in page	SWD	Application
Main array	ON	00	x	-	R/E/W/F
		01	x	-	对应 Block 只能取指
		10/11	x	-	对应 Block 只能取指
	OFF	00	x	R/E/W	R/E/W/F
		01	x		对应 Block 只能取指
		10/11	x	对应 Block 无法访问	对应 Block 只能取指
DCT LDT0	x	x	x	R	R
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF1,2,3	x	x	x	R/E/W	R/E/W
IF0	x	x	55	R/E	R
			others	R/E/W	R

注：

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

[2] 进行 flash 全擦后可以擦除 LDT1

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

4.5 寄存器

模块起始地址：0x4000_10000x4000_1000

offset 地址	名称	符号
0x00	Flash 读取控制寄存器 （Flash Read Control Register）	FLS_RDCR
0x04	--	--
0x08	用户配置字寄存器 （Flash Option Bytes Register）	FLS_OPTBR
0x0C	--	--
0x10	--	--
0x14	Flash 擦写控制寄存器 （Flash Erase/Program Control Register）	FLS_EPCR
0x18	Flash Key 输入寄存器 （Flash Key Register）	FLS_KEY
0x1C	Flash 中断使能寄存器 （Flash Interrupt Enable Register）	FLS_IER
0x20	Flash 标志寄存器 （Flash Interrupt Status Register）	FLS_ISR
0x24	--	--
0x28	Flash ECC 标志寄存器	FLS_ECCSR
0x2C	Flash ECC 中断使能寄存器	FLS_ECC_IER
0x30	Flash ECC 错误地址寄存器	FLS_ECC_EADD
0x38	Flash ECC 控制寄存器	FLS_ECCCR
0x60	ACLOCK 寄存器 1 （Flash Application Code Lock Register1）	FLS_ACLOCK1
0x64	ACLOCK 寄存器 2 （Flash Application Code Lock Register1）	FLS_ACLOCK2
0x68	ACLOCK 寄存器 3 （Flash Application Code Lock Register1）	FLS_ACLOCK3
0x90	DATA Flash 擦写控制寄存器 （Flash Erase/Program Control Register）	DFLS_EPCR
0x94	DATA Flash Key 输入寄存器 （Flash Key Register）	DFLS_KEY
0x98	DATA Flash 中断使能寄存器 （Flash Interrupt Enable Register）	DFLS_IER
0x9C	DATA Flash 标志寄存器 （Flash Interrupt Status Register）	DFLS_ISR

4.5.1 Flash 读取控制寄存器（FLS_RDCR）

名称	FLS_RDCR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			DATA_WAIT			CODE_WAIT	
位权限	U-0			R/W-010			R/W-00	

位号	助记符	功能描述
31:5	-	RFU：未实现，读为 0
4:2	DATA_WAIT	Data Flash 读等待周期配置 000：无等待 001：1 wait cycle … 111：7 wait cycle Data flash 实际读周期为 DATA_WAIT+1 个 AHB 时钟周期 Data Flash 最小读周期为 80ns，软件应根据当前系统时钟频率合理配置 DATA_WAIT，确保读周期不短于 80ns 比如，当系统频率 AHBCLK 为 24MHz，对应时钟周期为 41.6ns，则读取数据 flash 至少需要 2 个周期，可以配置 DATA_WAIT 为大于等于 1 的数。
1:0	CODE_WAIT	Code Flash 读等待周期配置 (Wait Cycles) 00/11：0 wait cycle 01：1 wait cycle 10：2 wait cycles CPU 主频小于等于 24MHz 时，不需要开启 wait；主频大于 24MHz 小于等于 48MHz 时使能 1 wait

4.5.2 用户配置字寄存器（FLS_OPTBR）

名称	FLS_OPTBR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	IWDTEN	-					
位权限	R-0	R-1	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				ACLOCKEN		DBRDPEN	
位权限	U-0				R-01		R-01	

位号	助记符	功能描述
31	IWDTSLP	IWDT 在休眠模式下是否允许应用暂停计数 (IWDT Sleep) 1：允许应用在休眠模式下暂停 IWDT 计数 0：禁止应用在休眠模式下暂停 IWDT 计数
30	IWDTEN	IWDT 使能 1：上电后 IWDT 开始运行 0：上电后 IWDT 不运行
29:6	-	RFU：未实现，读为 0
5:4	DBRDPEN_D	DATA FLASH Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能
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_C	CODE FLASH Debug Port 读取保护使能 (Debug Read Protection Enable) 00/01/11：DBRDP 不使能 10：DBRDP 使能

4.5.3 ACLOCK 寄存器 1（FLS_ACLOCK1）

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

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

4.5.4 ACLOCK 寄存器 2（FLS_ACLOCK2）

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

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

4.5.5 ACLOCK 寄存器 3（FLS_ACLOCK3）

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

位号	助记符	功能描述
31:0	LOCK3	ACLOCK 配置寄存器 3，分别用于控制 Block47~Block32 的应用代码读写锁定。每个 Block 大小为 8KB，每个 Block 使用 2bit 进行权限控制。(Lock bits)

位号	助记符	功能描述
		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。

4.5.6 Flash 擦写控制寄存器（FLS_EPCR）

名称	FLS_EPCR							
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	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

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

4.5.7 Flash Key 输入寄存器（FLS_KEY）

名称	FLS_KEY							
offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

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

4.5.8 Flash 中断使能寄存器（FLS_IER）

名称	FLS_IER							
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	-				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)

4.5.9 Flash 标志寄存器（FLS_ISR）

名称	FLS_ISR

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

位号	助记符	功能描述
31:20	-	RFU：未实现，读为 0
19:17	KEYSTA	Flash 擦写 KEY 输入状态 (Flash Key Status) 000：Flash 写保护状态，未输入 KEY 001：全擦解锁状态 010：页擦解锁状态 011：编程解锁状态 100：KEY 错误锁定状态，需要复位才能解锁 101：扇区擦解锁状态 110/111：RFU
16:12	-	RFU：未实现，读为 0
11	OTPERR	OTP page 编程权限错误，硬件置位，软件写 1 清零 (OTP program Error) 1：尝试对已编程的 OTP 字节进行编程 0：无 OTP 编程错误
10	AUTHERR	Flash 读写权限错误，读取 LOCK 块数据或对 LOCK 块擦写时置位，软件写 1 清零。(Flash Authentication Error) 1：Flash 访问权限错误 0：Flash 访问没有发生权限错误
9	KEYERR	Flash KEY 错误，硬件置位，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

4.5.10 Flash ECC 标志寄存器（FLS_ECCSR）

名称	FLS_ECCSR							
offset	0x00000028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DD	CD	DC	CC	BE	-		-

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

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

4.5.11 Flash ECC 中断使能寄存器（FLS_ECC_IER）

名称	FLS_ECC_IER							
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	-				DDIE	CDIE	DCIE	CCIE

位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:4	-	RFU：未实现，读为 0
3	DDIE	Data flash ECC 检错中断使能，1 有效
2	CDIE	Data flash ECC 纠错中断使能，1 有效
1	DCIE	Code flash ECC 检错中断使能，1 有效
0	CCIE	Code flash ECC 纠错中断使能，1 有效

4.5.12 Flash ECC 错误地址寄存器（FLS_ECC_EADD）

名称	FLS_ECC_EADD							
offset	0x00000030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CFLS_ERR_ADD							
位权限	R-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CFLS_ERR_ADD				-			
位权限	R-0				U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CFLS_ERR_ADD							
位权限	R-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CFLS_ERR_ADD							
位权限	R-0							

位号	助记符	功能描述
31:20	DFLS_ERR_ADD	DATA FLASH ECC 错误地址，word 地址，只读
19:16	-	RFU：未实现，读为 0
15:0	CFLS_ERR_ADD	CODE FLASH ECC 错误地址，word 地址，只读

4.5.13 Flash ECC 控制寄存器（FLS_ECCCR）

名称	**FLS_ECCCR**							
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	CFECCCR							
位权限	R/W-0000 0001							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DFECCR							
位权限	R/W-0000 0001							

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

4.5.14 DATA Flash 擦写控制寄存器（DFLS_EPCR）

名称	DFLS_EPCR							
offset	0x00000090							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						ERTYPE	
位权限	U-0						R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PREQ	EREQ
位权限	U-0						R/W-0	R/W-0

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

4.5.15 DATA Flash Key 输入寄存器（DFLS_KEY）

名称	PFLS_KEY

offset	0x00000094							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY[7:0]							
位权限	W-0000 0000							

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

4.5.16 DATA Flash 中断使能寄存器（DFLS_IER）

名称	FLS_IER							
offset	0x00000098							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LVERRIE	-			OTPIE	AUTHIE	KEYIE	CKIE
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						PRDIE	ERDIE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU：未实现，读为 0
15	LVERRIE	低压擦写错中断使能，1 有效
14:12	-	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

位号	助记符	功能描述
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)

4.5.17 DATA Flash 标志寄存器（DFLS_ISR）

名称	FLS_ISR							
offset	0x0000009C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				KEYSTA			-
位权限	U-0				R-000			U-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LVERR	-			OTPERR	AUTHERR	KEYERR	CKERR
位权限	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						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	-	RFU：未实现，读为 0
15	LVERR	低压擦写错误，硬件置位，软件写 1 清零 1：使能 SVD 擦写保护时，SVD 报警的情况下启动擦写
14:12	-	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)

位号	助记符	功能描述
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

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 2.2~5.5V。

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 2.1V。

芯片下电时，复位阈值由 PDR 电路决定，固定值 1.8V。

芯片还可通过配置 SVD 为欠压模式复位模式实现不同档位的下电复位。

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

注意：在任何情况下不得同时关闭 SVD 欠压复位和 PDR 复位，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 2.2~5.5V，所有模拟模块在这个电压范围内都可以保证正常工作。

- REGC

内部 LDO 输出，外部靠近管脚对地连接 1uF 稳压电容。

5.1.2 电源结构图

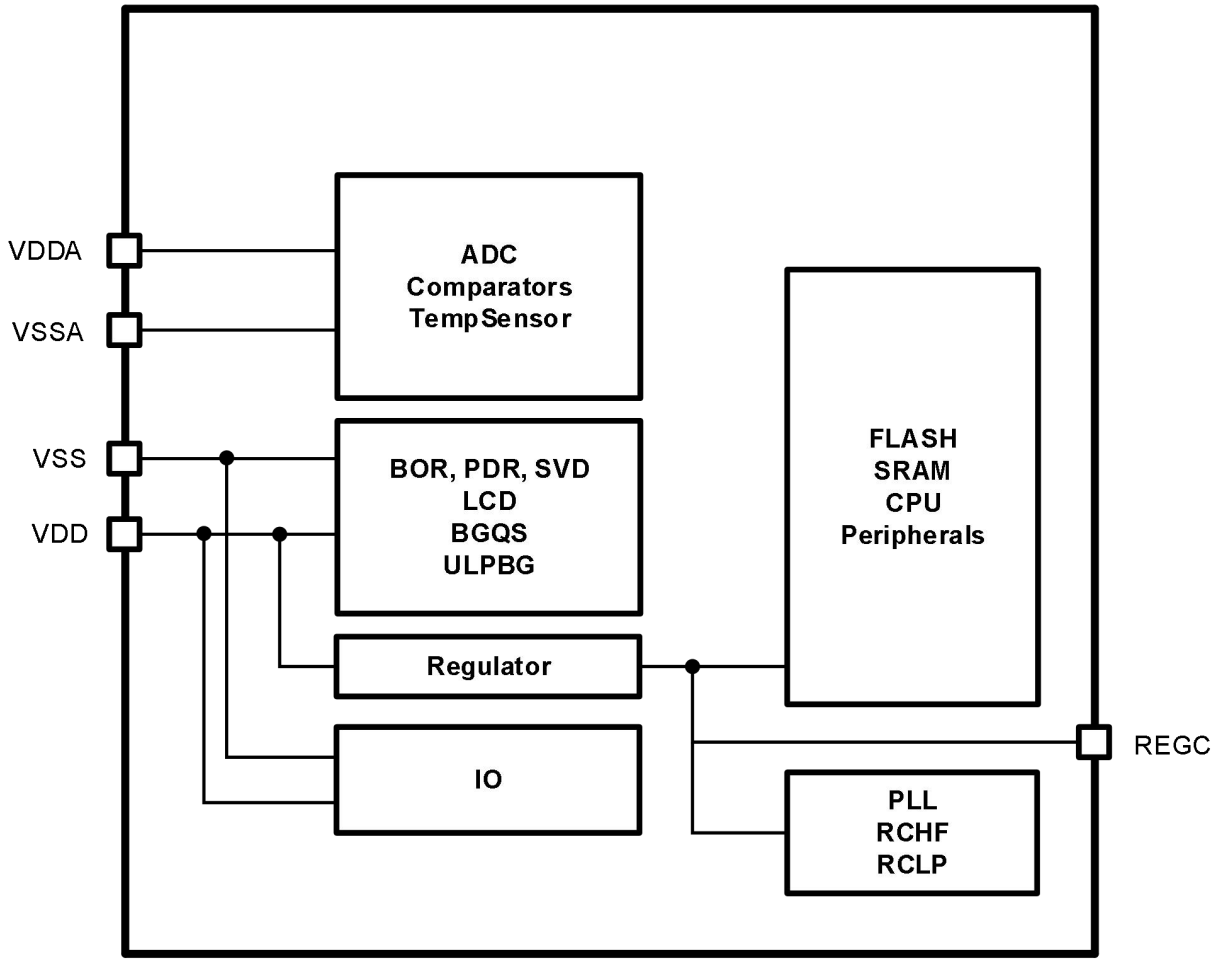


图 5-1 芯片电源结构图

5.1.3 ADC 基准电压

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

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

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

5.1.4 片内快速建立基准源（AVREF）

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

5.2 功耗模式

5.2.1 概述

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

芯片支持的功耗模式：

- ACTIVE 模式：正常运行
- LP Run 模式：LDO 工作在超低功耗模式下，CPU 和外设只能运行在较低频率下
- SLEEP 模式：CPU 停止，Flash 停止，LDO 工作在超低功耗模式下，仅部分外设可以运行
- DEEPSLEEP 模式：CPU 停止，Flash 停止，关闭基准电压，关闭 LDO，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	-
LP Run	30uA@32KHz	软件主动退出	低速工作	-
SLEEP	5uA	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	5us
DEEPSLEEP	1.5uA		CPU 休眠[2] 关闭 RCHF、PLL、ADC 等 关闭 BGQS，关闭 LDO15，RTC 走时	65us

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。

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	
LP Run	32khz	XTLF, RCLP	0	XTLF, RCLP

5.2.3 Active 模式

芯片正常工作模式。芯片上电复位完成后进入 Active 模式运行，默认的 CPU 频率是 8MHz，最高可以运行到 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，必须正确配置 flash wait。

5.2.4 LP Run 模式

当芯片需要低功耗低速运行时，可进入 LP RUN 模式，此时 LDO 进入低功耗模式，内核使用 LSCLK 运行，典型频率 32KHz。在需要高速运行时，软件可主动退出 LP RUN 进入 ACTIVE 模式，然后再将系统时钟切换到较高频率。

进入 LPRUN 模式

进入 LPRUN 的操作步骤：

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

LPRUN 模式下的硬件行为

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

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

退出 LPRUN 模式

按照以下步骤退出 LPRUN 模式：

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

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

注意：LPRUN 模式下如果发生 XTLF 停振，则系统无法运行，等待看门狗复位。

5.2.5 SLEEP 模式

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

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 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
- 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

5.2.6 DEEPSLEEP 模式

DEEPSLEEP 是芯片最低功耗模式，休眠功耗比 SLEEP 进一步降低大约 2uA。

进入 DEEPSLEEP 模式

软件按如下步骤进入 DEEPSLEEP 模式：

- 置位 SLPDP 寄存器

- 配置 PMOD 寄存器为 10
- 执行 WFI 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；

数字外设模块可以使用 XTLF、RCLP 等低速时钟继续工作。

注意，SVD、TSI、比较器等依赖内部基准的模块都不能工作。

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	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	√	√
LPTIM16	可屏蔽，用于定时唤醒	√	√
BSTIM16	可屏蔽，用于定时唤醒	√	√
I2C 从机	可屏蔽，用于从机接收唤醒	√	√
UARTx 唤醒	可屏蔽，检测到 UART0~3 下降沿唤醒信号时唤醒芯片	√	√
LIN 唤醒	可屏蔽，检测到 LIN 唤醒信号时唤醒芯片	√	√

通过Cortex-M0的PRIMASK功能，可以实现以上中断事件唤醒芯片，但是CPU不执行中断处理程序。

此时唤醒后CPU将继续从休眠前的指令之后开始运行。

注：芯片从休眠模式唤醒后，软件可以通过查询PMU.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寄存器复位，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，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：0x4000_2000

offset 地址	名称	符号
0x00	低功耗控制寄存器 （Power Management Control Register）	PMU_CR
0x04	唤醒时间控制寄存器 （Wakeup Time Register）	PMU_WKTR
0x08	唤醒源标志查询寄存器 （Wakeup Source Flags Register）	PMU_WKFR
0x0C	PMU 中断使能寄存器 （PMU Interrupt Enable Register）	PMU_IER
0x10	PMU 中断标志寄存器 （PMU Interrupt and Status Register）	PMU_ISR
0x3C	模拟 BUFFER 控制寄存器 （PMU Analog Buffer Control Register）	PMU_BUFCR
0x40	-	-
0x44	-	-
0x48	-	-
0x4C	-	-
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	RFUI	-					-	
位权限	R/W-0	U-0					U-0	

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				RFUI			
位权限	U-0				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				WKFSEL		SLPDP	-
位权限	U-0				R/W-00		R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				RFUI		PMOD	
位权限	U-0				R/W-00		R/W-00	

Bit	助记符	功能描述
31	**RFUI**	Dummy 寄存器
30:29	**--**	未实现，读为 0
28	**--**	未实现，读为 0
27:26	**--**	未实现，读为 0
25:24	**--**	未实现，读为 0
23:12	**--**	未实现，读为 0
11:10	**WKFSEL**	Sleep/DeepSleep 唤醒后的系统频率 00：RCHF-8MHz 01：RCHF-16MHz 10：RCHF-24MHz 11：RFU
9	**SLPDP**	DeepSleep 控制寄存器 1：DeepSleep 模式使能，下关闭基准电压源 0：常规 Sleep 模式 在 Sleep 下，如果置位了 SLPDP 位即为 DeepSleep 模式； 该位仅在 Sleep 下有效
8	**--**	未实现，读为 0
7:4	**--**	未实现，读为 0
3:2	**RFUI**	Dummy 寄存器
1:0	**PMOD**	低功耗模式配置寄存器 00：Active mode 01：LPRUN mode 10：Sleep mode / DeepSleep mode 11：RFU

5.5.2 唤醒时间控制寄存器（PMU_WKTR）

名称	**PMU_WKTR**							
offset	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						R/W-01	

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1:0	**T1A**	可编程额外唤醒延迟 ● 在 Sleep/DeepSleep 模式下，RCHF 时钟到来后，根据此寄存器配置等待额外延迟时间后，再读取 Flash 校验字 00：48us 01：64us 10：80us 11：96us

5.5.3 唤醒源标志查询寄存器（PMU_WKFR）

名称	PMU_WKFR							
offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART2WKF	UART1WKF	UART0WKF	RTCWKF	SVDWKF	LFDETWKF	UART3WKF	IOWKF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	I2C0WKF	I2C1WKF	UART5WKF	UART4WKF	LPU0WKF	UART7WKF	-	COMPWKF
位权限	R-0	R-0	R-0	R-0	R-0	R-0	U-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	UART6WKF	CFD0WKF	-	LPTWKF	-	BST16WKF	DBGWKF	-
位权限	R-0	R-0	U-0	R-0	U-0	R-0	R-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WKPxF							
位权限	R/W-00000000							

位	位名	功能描述
31	UART2WKF	UART2 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
30	UART1WKF	UART1 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
29	UART0WKF	UART0 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
28	RTCWKF	RTC 中断唤醒标志，中断撤销时硬件自动清零
27	SVDWKF	SVD 中断唤醒标志，中断撤销时硬件自动清零
26	LFDETWKF	32768Hz 晶体停振中断唤醒标志，中断撤销时硬件自动清零
25	UART3WKF	UART3 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
24	IOWKF	IO 中断唤醒标志，中断撤销时硬件自动清零
23	I2C1WKF	I2C_SMB1 中断唤醒标志，中断撤销时硬件自动清零
22	I2C0WKF	I2C_SMB0 中断唤醒标志，中断撤销时硬件自动清零

位	位名	功能描述
21	UART5WKF	UART5 中断唤醒标志，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
20	UART4WKF	UART4 中断唤醒标志(NEWKF)，中断撤销时硬件自动清零 包含 LINWKF 和 NEWKF
19	LPU0WKF	LPUART0 中断唤醒标志，中断撤销时硬件自动清零
18	UART7WKF	UART7 中断唤醒标志(NEWKF)，中断撤销时硬件自动清零
17	--	未实现，读为 0
16	COMPWKF	比较器中断唤醒标志，中断撤销时硬件自动清零
15	UART6WKF	UART6 中断唤醒标志(NEWKF)，中断撤销时硬件自动清零
14	--	未实现，读为 0
13	--	未实现，读为 0
12	LPTWKF	LPTIM16 中断唤醒标志，中断撤销时硬件自动清零
11	--	未实现，读为 0
10	BST16WKF	BSTIM16 中断唤醒标志，中断撤销时硬件自动清零
9	DBGWKF	CPU Debugger 唤醒标志，软件写 1 清零
8	--	未实现，读为 0
7:0	WKPxF	WKUP0~7 Pin 唤醒标志，软件写 1 清零

5.5.4 PMU 中断使能寄存器（PMU_IER）

名称	PMU_IER							
offset	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	-							LPREIE
位权限	U-0							R/W-0

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

5.5.5 PMU 中断标志寄存器（PMU_ISR）

名称	PMU_ISR							
offset	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	-							
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	-							LPREIF
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:2	--	未实现，读为 0
1	--	未实现，读为 0
0	**LPREIF**	LPRUN 错误中断标志，硬件置位，软件写 1 清零；软件进入 LPRUN 模式时如果触发了 LPREIF，则芯片仍将停留在 ACTIVE 模式 1：LPRUN Condition Error，即进入 LPRUN 时满足如下情况： 1） 系统时钟不是 LSCLK，或 2） RCHF 使能未关闭 0：LPRUN 正常进入

5.5.6 模拟 BUFFER 控制寄存器（PMU_BUFCR）

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

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

位号	助记符	功能描述
2	VPTATBUFFER_EN	Vptat Buffer 模块使能信号，高电平使能有效。(PTAT Buffer Enable)
1	-	RFU：未实现，读为 0
0	-	RFU：未实现，读为 0

5.5.7 PTAT 控制寄存器（PMU_PTATCR）

名称	PMU_PTATCR							
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 处理器（CPU）

6.1 概述

FM33FH0 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

6.1.1 处理器配置

Feature	Options	FM33FH0Config
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 6-1FM33FH0 CPU 配置简表

6.2 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 6-2 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。

6.3 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。

6.3.1 中断向量表

参见7.2中断向量表

6.3.2 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。

6.3.3 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

FM33FH0 的 HardFault 触发原因可以通过寄存器查询，以帮助软件开发者定位错误原因。

6.3.4 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。

6.4 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》

6.4.1 调试功能引脚

FM33FH0 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。

6.4.2 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。

6.4.3 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)。

7 中断向量表（INTMUX）

7.1 概述

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

7.2 中断向量表

CPU 中断向量表如下，

Position	Priority	Priority type	Acronym	Description	Address
0	-	-	MSP 初值	主栈指针初始化地址	0x0000_0000
1	-3	fixed	Reset	复位向量	0x0000_0004
2	-2	fixed	NMI	WKUPx 中断 PMU 中断 存储器非法访问中断 Flash ECC 检错中断	0x0000_0008
3	-1	fixed	HardFault	HardFault 中断向量	0x0000_000C
4-10	-	-	-	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall 系统服务请求	0x0000_002C
12-13	-	-	-	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	可挂起系统服务请求	0x0000_0038
15	6	settable	Systick	内部定时器中断向量	0x0000_003C
16	7	settable	MUX0	多路选择中断输入	0x0000_0040
17	8	settable	MUX1		0x0000_0044
18	9	settable	MUX2		0x0000_0048
19	10	settable	MUX3		0x0000_004C
20	11	settable	MUX4		0x0000_0050
21	12	settable	MUX5		0x0000_0054
22	13	settable	MUX6		0x0000_0058
23	14	settable	MUX7		0x0000_005C
24	15	settable	MUX8		0x0000_0060
25	16	settable	MUX9		0x0000_0064
26	17	settable	MUX10		0x0000_0068
27	18	settable	MUX11		0x0000_006C
28	19	settable	MUX12		0x0000_0070
29	20	settable	MUX13		0x0000_0074
30	21	settable	MUX14		0x0000_0078
31	22	settable	MUX15		0x0000_007C
32	23	settable	MUX16		0x0000_0080
33	24	settable	MUX17		0x0000_0084
34	25	settable	MUX18		0x0000_0088
35	26	settable	MUX19		0x0000_008C

Position	Priority	Priority type	Acronym	Description	Address
36	27	settable	MUX20		0x0000_0090
37	28	settable	MUX21		0x0000_0094
38	29	settable	MUX22		0x0000_0098
39	30	settable	MUX23		0x0000_009C
40	31	settable	MUX24		0x0000_00A0
41	32	settable	MUX25		0x0000_00A4
42	33	settable	MUX26		0x0000_00A8
43	34	settable	MUX27		0x0000_00AC
44	35	settable	MUX28		0x0000_00B0
45	36	settable	MUX29		0x0000_00B4
46	37	settable	MUX30		0x0000_00B8
47	38	settable	MUX31		0x0000_00BC

表 7-1 中断向量表

7.3 中断向量多路选择

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

	MUXNSEL=00	MUXNSEL=01	MUXNSEL=10	MUXNSEL=11
MUX0	WDT	TAU02	TAU05	TAU00
MUX1	SVD	TAU04	TAU02	TAU01
MUX2	RTC	TAU02	TAU05	TAU01
MUX3	-	未实现，读为 0	TAU14	TAU02
MUX4	QSPI	TAU15	TAU16	TAU03
MUX5	ADC	TAU11	TAU10	TAU04
MUX6	UART0	SENTx	LPTIM	I2C0
MUX7	UART1	TAU07	TAU10	TAU16
MUX8	UART2	ATIM	I2C0	QSPI
MUX9	UART3	TAU06	TAU07	TAU13
MUX10	UART4	TAU12	TAU13	SPI2
MUX11	UART5	TAU17	TAU02	TAU01
MUX12	I2C0	TAU03	TAU04	TAU05
MUX13	SPI0	TAU13	TAU14	TAU06
MUX14	SPI1	TAU16	TAU17	TAU07
MUX15	I2C1	TAU16	TAU17	TAU10
MUX16	UART6	TAU15	TAU16	TAU03
MUX17	UART7	TAU10	TAU16	LPTIM
MUX18	BSTIM	TAU10	TAU07	TAU06
MUX19	TSI	SPI2	TAU03	TAU00
MUX20	LPTIM	TAU13	TAU12	TAU11

MUX21	DMA	TAU12	TAU13	TAU11
MUX22	WKUPx	TAU15	TAU16	TAU12
MUX23	TAUxx	UART0	UART2	UART4
MUX24	DAC	TAU06	TAU07	TAU13
MUX25	COMPx	TAU06	TAU07	TAU14
MUX26	SENTx	TAU03	TAU04	TAU15
MUX27	NVMIF	TAU07	TAU10	TAU16
MUX28	ATIM	TAU14	TAU16	UART2
MUX29	LPUART	TAU07	TAU10	TAU11
MUX30	EXTI	TAU15	TAU17	TAU05
MUX31	RAMP/FDET/FLSEC C 纠错/CLM 中断	TAU12	TAU13	TAU17

表 7-2 中断源多路选择表

7.4 寄存器

模块起始地址：0x4001_6000

offset 地址	名称	符号
0x00	中断源多路选择控制寄存器 1（Interrupt Multiplexer Control Register）	INTMUX_CR1
0x04	中断源多路选择控制寄存器 2（Interrupt Multiplexer Control Register）	INTMUX_CR2

7.4.1 中断源多路选择控制寄存器 1（INTMUX_CR1）

名称	INTMUX_CR1							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MUX15SEL		MUX14SEL		MUX13SEL		MUX12SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MUX11SEL		MUX10SEL		MUX9SEL		MUX8SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MUX7SEL		MUX6SEL		MUX5SEL		MUX4SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MUX3SEL		MUX2SEL		MUX1SEL		MUX0SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

Bit	助记符	功能描述
31:30	MUX15SEL	寄存器定义参见表 7-2 中断源多路选择表

Bit	助记符	功能描述
29:28	MUX14SEL	
27:26	MUX13SEL	
25:24	MUX12SEL	
23:22	MUX11SEL	
21:20	MUX10SEL	
19:18	MUX9SEL	
17:16	MUX8SEL	
15:14	MUX7SEL	
13:12	MUX6SEL	
11:10	MUX5SEL	
9:8	MUX4SEL	
7:6	MUX3SEL	
5:4	MUX2SEL	
3:2	MUX1SEL	
1:0	MUX0SEL	

7.4.2 中断源多路选择控制寄存器 2（INTMUX_CR2）

名称	INTMUX_CR2							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MUX31SEL		MUX30SEL		MUX29SEL		MUX28SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MUX27SEL		MUX26SEL		MUX25SEL		MUX24SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MUX23SEL		MUX22SEL		MUX21SEL		MUX20SEL	
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	MUX19SEL		MUX18SEL		MUX17SEL		MUX16SEL	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

Bit	助记符	功能描述
31:30	MUX31SEL	寄存器定义参见表 7-2 中断源多路选择表
29:28	MUX30SEL	
27:26	MUX29SEL	
25:24	MUX28SEL	
23:22	MUX27SEL	
21:20	MUX26SEL	
19:18	MUX25SEL	
17:16	MUX24SEL	
15:14	MUX23SEL	
13:12	MUX22SEL	
11:10	MUX21SEL	
9:8	MUX20SEL	

Bit	助记符	功能描述
7:6	MUX19SEL	
5:4	MUX18SEL	
3:2	MUX17SEL	
1:0	MUX16SEL	

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（POR）典型释放电压 2.1V
- 下电复位（PDR）典型复位电压 1.8V
- SVD 可配置为低压报警或低压复位功能
- 上下电复位信号经过去抖动和延时，抗干扰能力强

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

8.2 模块框图

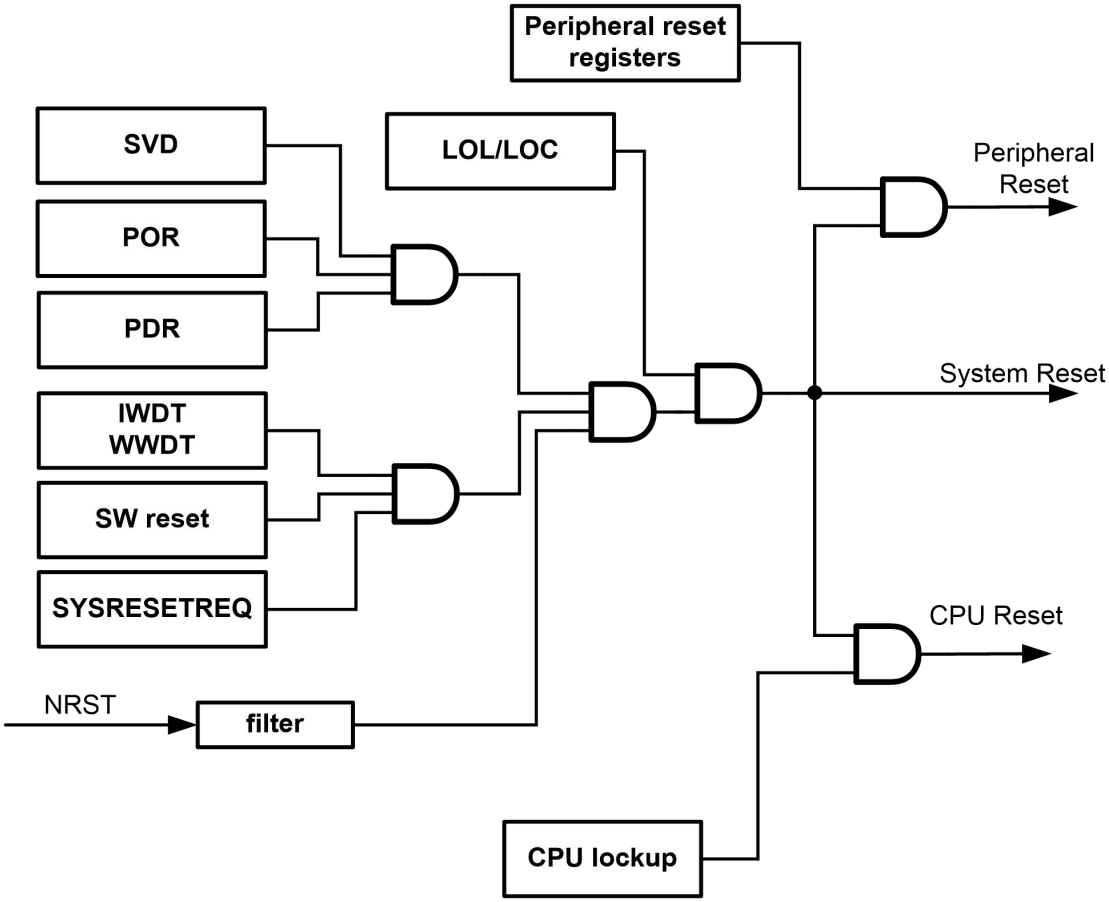


图 8-1 芯片复位源框图

8.3 上下电复位

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

- POR：提供上电复位
- SVD：提供精确的下电复位和报警
- 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。

芯片上电复位放开后，SVD默认是关闭的，软件可以开启。开启SVD之前，必须保证芯片电源电压已经高于SVD复位阈值，在电源电压低于SVD复位阈值时使能SVD复位功能，芯片将直接复位。

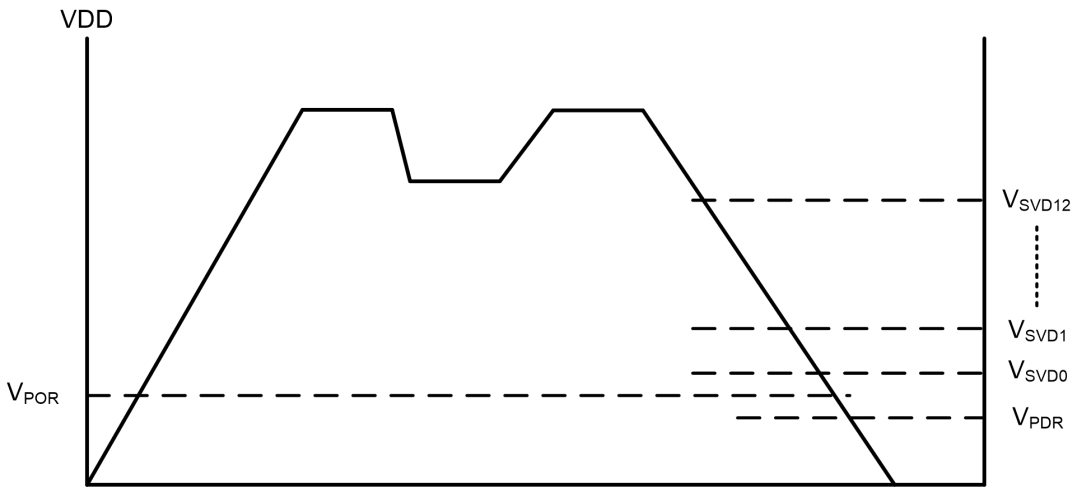


图 8-2 上下电复位示意图

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

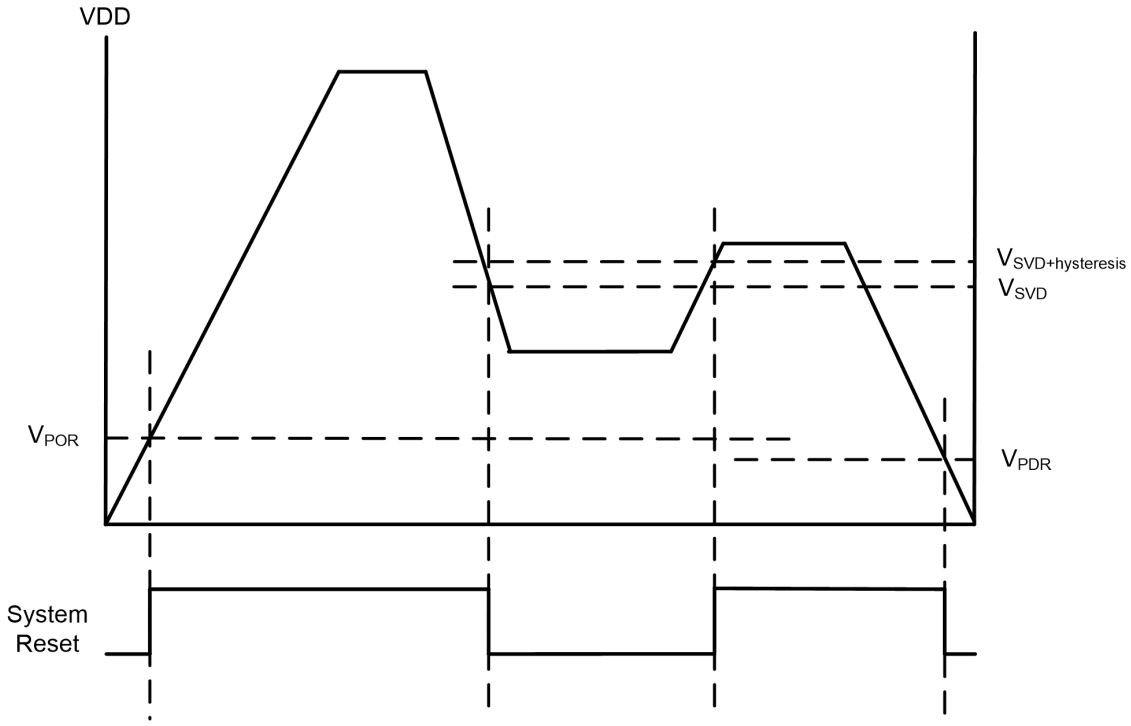


图 8-3 POR, SVD, PDR 复位波形示意图

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

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（PA0），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 LOC 复位

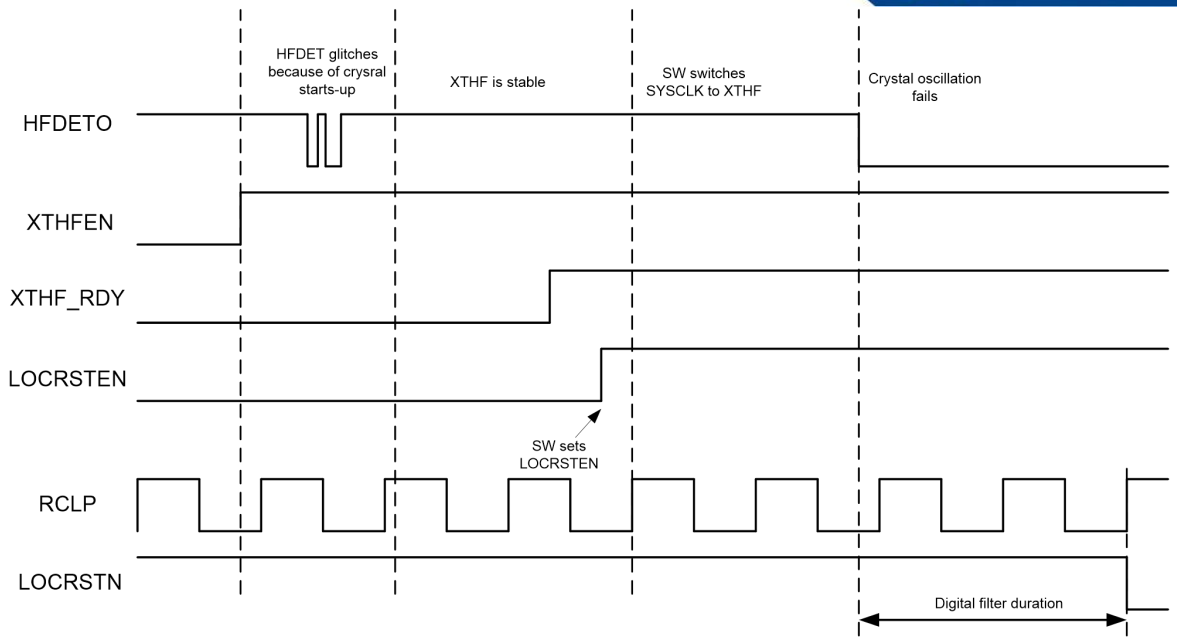
LOC（Loss Of Clock）复位在以下条件满足时复位全芯片：

- SYSCLK 源头选择为 XTHF
- 使能 LOC 复位功能（LOCRSTEN=1）
- HFDET 检测到 XTHF 停振

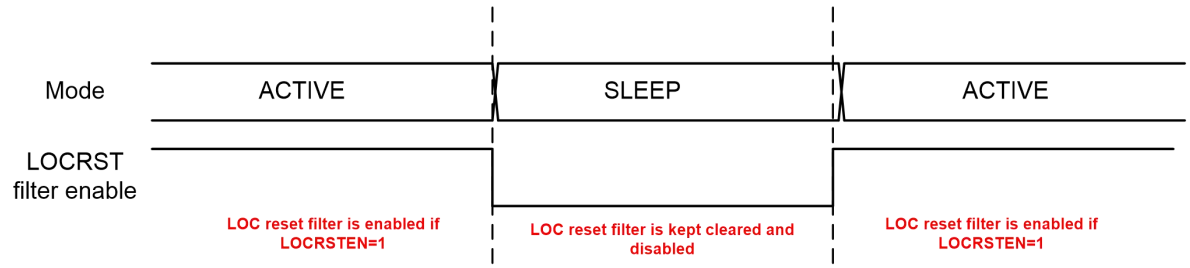
或者

- SYSCLK 源头选择为 XTLF
- 使能 LOC 复位功能（LOCRSTEN=1）
- LFDET 检测到 XTLF 停振

LOC 复位功能波形图如下：



LOCRSTEN 仅在 ACTIVE 模式下有效



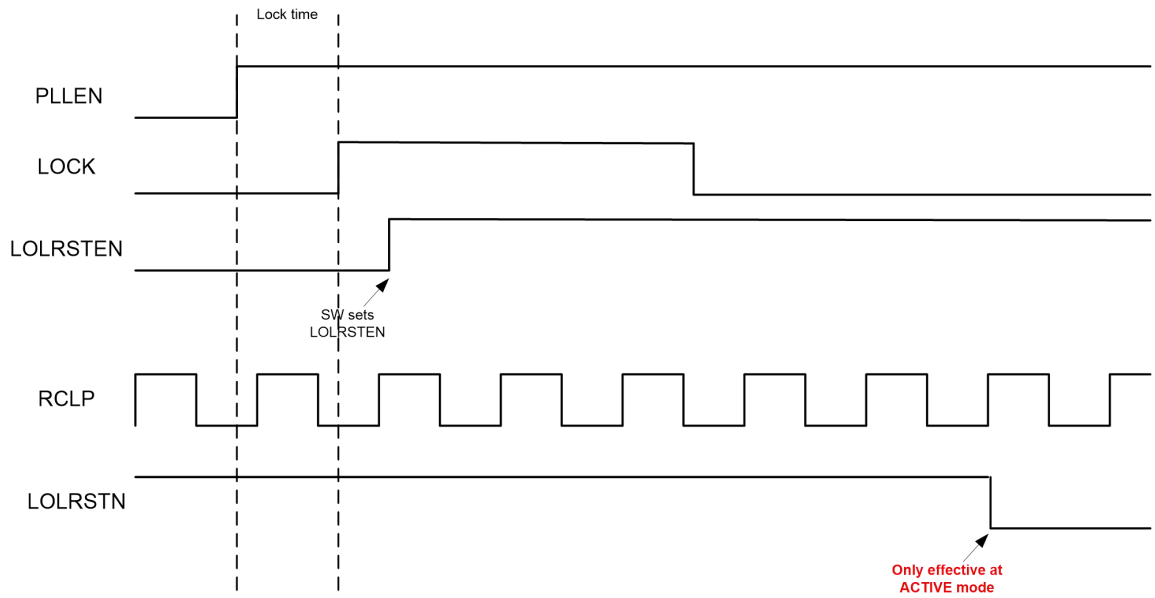
8.7 LOL 复位

Loss of Lock 复位在以下条件满足时复位全芯片：

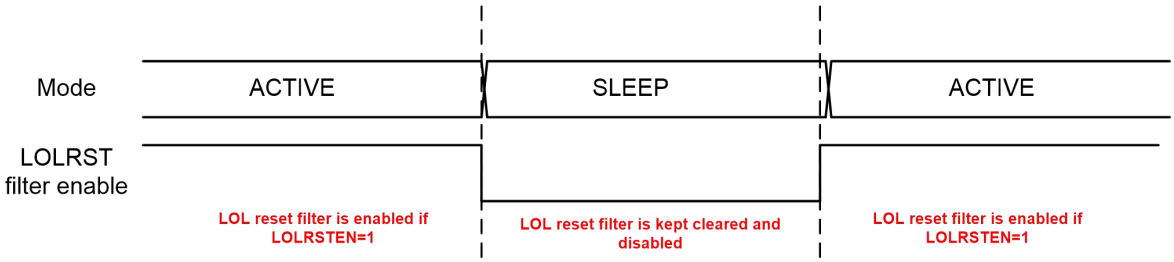
- SYSCLK 源头选择为 PLLH
- 使能 LOL 复位功能（LOLRSTEN=1）
- PLLH 失锁，LOCKED 标志清零

当 PLLH 参考时钟和输出反馈时钟之间的相位差大于设计阈值时，PLLH 输出的 LOCK 信号清零，表示 PLLH 失去锁定状态。

LOL 复位波形如下



注意：当芯片休眠后，PLLH 自动关闭，LOCK 标志清零，如果系统以 PLLH 为系统时钟进入休眠，会触发错误的复位。因此 LOL 复位功能仅在 Active 模式下有效，进入休眠后自动屏蔽 LOLRSTEN。



8.8 寄存器

模块起始地址：0x4000_2800

offset 地址	名称	符号
0x00	PDR 配置寄存器	PDR_CFG
0x04	-	-
0x08	复位配置寄存器	RMU_RSTCR
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

8.8.1 PDR 配置寄存器（PDR_CFG）

名称	PDR_CFG							
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	-					idac2_vdd_cfg	idac1_vdd_cfg	PDREN
位权限	U-0					R/W-0	R/W-0	R/W-1

Bit	助记符	功能描述
31:3	**--**	RFU：未实现，读为 0
2	**idac2_vdd_cfg**	TSI idac2 的 VDD 电压适配，VDD>4V 时建议配 1
1	**idac1_vdd_cfg**	TSI idac1 的 VDD 电压适配，VDD>4V 时建议配 1
0	**PDREN**	下电复位使能 0：关闭下电复位 1：使能下电复位

8.8.2 复位配置寄存器（RMU_RSTCR）

名称	**RSTCR**							
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	LOLRSTEN	LOCRSTEN	-				LKUPRST_EN	-
位权限	R/W-0	R/W-0	U-0				R/W-0	U-0

Bit	助记符	功能描述
31:8	**--**	RFU：未实现，读为 0
7	**LOLRSTEN**	PLL 失锁复位使能（仅 Active 模式有效）

Bit	助记符	功能描述
		1：当检测到 PLL 失锁事件，复位芯片 0：PLL 失锁不会复位芯片，仅产生报警
6	**LOCRSTEN**	晶振停振复位使能 1：当检测到晶振停振，复位芯片 0：晶振停振不会复位芯片，仅产生报警 注，LOCRSTEN=1 时产生 LOC 复位的条件： 1） 系统时钟为 XTHF，XTHF 停振 2） 系统时钟为 XTLF，XTLF 停振 3） 系统时钟为 PLL，PLL 参考时钟为 XTHF，XTHF 停振
5:2	--	RFU：未实现，读为 0
1	**LKUPRST_EN**	LOCKUP 复位使能 1：使能 CPU LOCKUP 复位 0：屏蔽 CPU LOCKUP 复位
0	--	RFU：未实现，读为 0

8.8.3 软件复位寄存器（RMU_SOFTRST）

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

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

8.8.4 复位标志寄存器（RMU_RSTFR）

名称	**RSTFLAG**							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							CLM2_TOUT_FLAG
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	CLM2_UDF_FLAG	CLM2_OVF_FLAG	CLM1_TOUT_FLAG	CLM1_UDF_FLAG	CLM1_OVF_FLAG	CLM0_TOUT_FLAG	CLM0_UDF_FLAG	CLM0_OVF_FLAG
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	TESTN_FLAG	RAMP_FLAG	MAP_FLAG	MDF_FLAG	NRSTN_FLAG	-	-	PODRN_FLAG
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOL_FLAG	LOC_FLAG	SOFTN_FLAG	IWDTN_FLAG	SVDN_FLAG	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
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
31:25	--	RFU：未实现，读为 0
24	**CLM2_TOUT_FLAG**	CLM2_TOUT 复位标志，高有效，软件写 1 清零
23	**CLM2_UDF_FLAG**	CLM2_UDF 复位标志，高有效，软件写 1 清零
22	**CLM2_OVF_FLAG**	CLM2_OVF 复位标志，高有效，软件写 1 清零
21	**CLM1_TOUT_FLAG**	CLM1_TOUT 复位标志，高有效，软件写 1 清零
20	**CLM1_UDF_FLAG**	CLM1_UDF 复位标志，高有效，软件写 1 清零
19	**CLM1_OVF_FLAG**	CLM1_OVF 复位标志，高有效，软件写 1 清零
18	**CLM0_TOUT_FLAG**	CLM0_TOUT 复位标志，高有效，软件写 1 清零
17	**CLM0_UDF_FLAG**	CLM0_UDF 复位标志，高有效，软件写 1 清零
16	**CLM0_OVF_FLAG**	CLM0_OVF 复位标志，高有效，软件写 1 清零
15	**TESTN_FLAG**	TESTN 引脚复位标志，高有效，软件写 1 清零
14	**RAMP_FLAG**	RAM 奇偶校验错误复位标志，高有效，软件写 1 清零
13	**MAP_FLAG**	非法地址访问复位标志，高有效，软件写 1 清零
12	**MDF_FLAG**	模式诊断超时复位标志，高有效，软件写 1 清零
11	**NRSTN_FLAG**	NRST 引脚复位标志，高有效，软件写 1 清零
10	--	RFU：未实现，读为 0
9	--	RFU：未实现，读为 0
8	**PODRN_FLAG**	上下电复位标志，高有效，软件写 1 清零
7	**LOL_FLAG**	PLL 失锁复位标志，高有效，软件写 1 清零
6	**LOC_FLAG**	晶振停振复位标志，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	**SVDN_FLAG**	SVD 复位标志，高有效，软件写 1 清零
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

8.8.5 AHB 外设复位寄存器

名称	AHBRST
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	-						QSPIRST	DMARST
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	QSPIRST	QSPI 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
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

8.8.6 APB 外设复位寄存器 1

名称	APBRST1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	UART3RST	UART2RST	UART1RST	UART0RST	UARTIRRST	-
位权限	R/W-0	R/W-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	INTMUXRST	-	-	-	-	-	UART7RST	UART6RST
位权限	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPI2RST	SPI1RST	SPI0RST	I2C1RST	I2C0RST	-	-	LPUARTRST
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	TSIRST	ATIMRST	TAU1RST	TAU0RST	LPT16RST	BT16RST
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

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

Bit	助记符	功能描述
		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 0：不复位 1：复位
26	UART0RST	UART0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
25	UARTIRRST	UART 红外调制模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
24	--	RFU：未实现，读为 0
23	INTMUXRST	INTMUX 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22	--	RFU：未实现，读为 0
21	--	RFU：未实现，读为 0
20	--	RFU：未实现，读为 0
19	--	RFU：未实现，读为 0
18	--	RFU：未实现，读为 0
17	UART7RST	UART7 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
16	UART6RST	UART6 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
15	SPI2RST	SPI2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
14	SPI1RST	SPI1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
13	SPI0RST	SPI0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
12	I2C1RST	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
11	I2C0RST	I2C0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
10	--	RFU：未实现，读为 0
9	--	RFU：未实现，读为 0

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

8.8.7 APB 外设复位寄存器 2

名称	APBRST2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						DACRST	ADCCRST
位权限	U-0						R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	CLM2RST	CLM1RST	CLM0RST	-		CRCRST	-
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0		R/W-0	U-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	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	-	-	-	-	SVDRST	COMPRST
位权限	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25	DACRST	DAC 控制器复位，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
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	CLM2RST	CLM2 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
21	CLM1RST	CLM1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	CLM0RST	CLM0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
19	--	RFU：未实现，读为 0
18	--	RFU：未实现，读为 0
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
16:12	--	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
9:8	--	RFU：未实现，读为 0
7	--	RFU：未实现，读为 0
6	--	RFU：未实现，读为 0
5	--	RFU：未实现，读为 0
4:2	--	RFU：未实现，读为 0
1	SVDRST	SVD 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
0	COMPRST	COMP 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

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。独立看门狗的使能由 OPTBYTES 控制，可配置为上电后自启动或者软件启动（默认关闭 IWDT 时，执行清狗操作软件开启 IWDT）。

任何情况下， IWDT 一旦启动就不可关闭，直到芯片复位。

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

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

为了便于调试，在以下情况下 IWDT 会停止运行：

- 芯片 CP 时，当 TESTN 有效，IWDT 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

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

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

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

9.2 结构框图

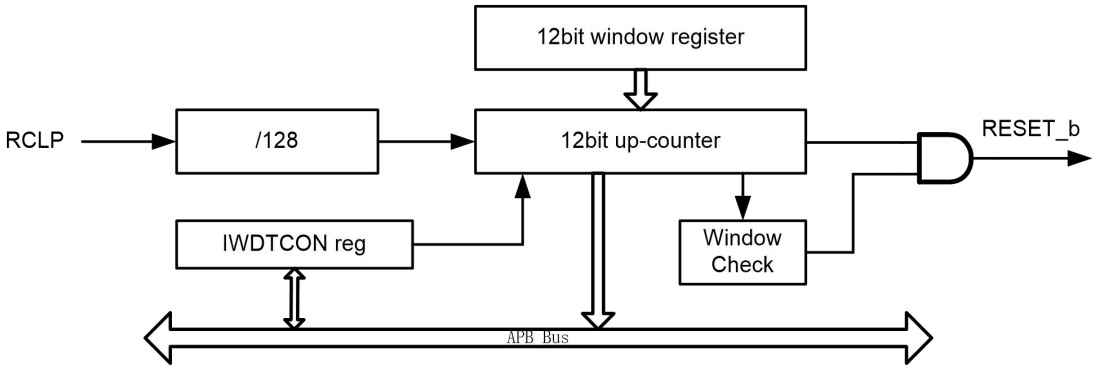


图 9-1 IWDT 结构框图

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

tWWDT = T_{RCLP} * 128 * IWDTOVP

RCLP 频率	溢出长度配置	溢出时间（ms）
32KHz	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。

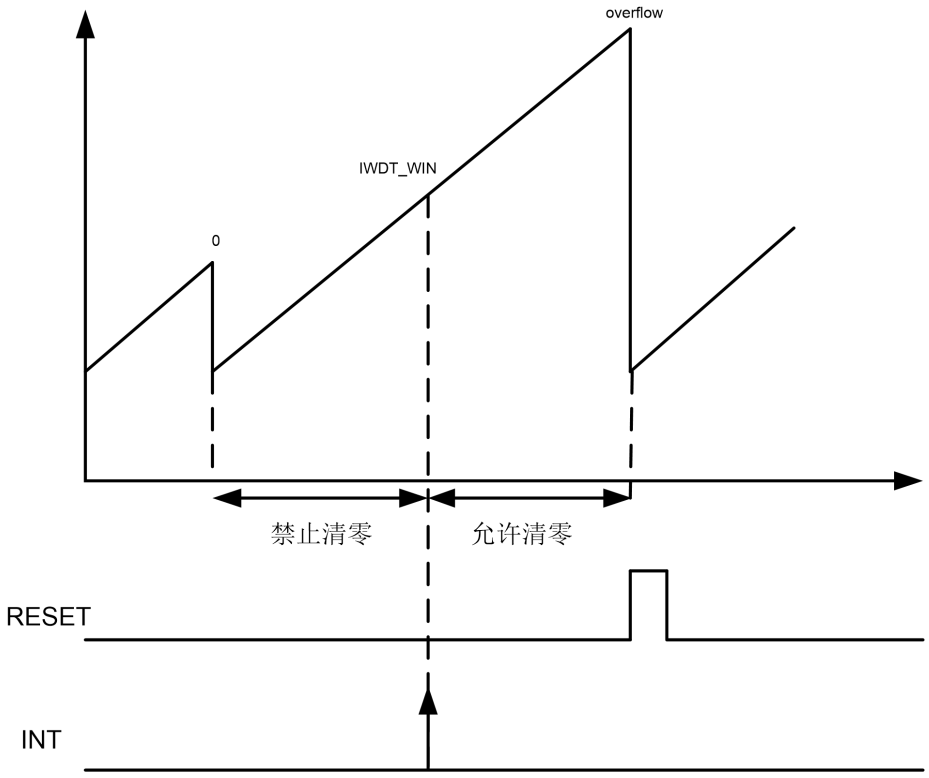


图 9-2 IWDT 窗口示意图

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**	上电后 IWDT 状态受 LDT1 区域 IWDTEN 控制。 IWDT 启动后，软件向此地址写入 0x1234_5A5A 时清狗 (IWDT Service Register,write only)

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 模式下冻结 IWDT 计数 0：Sleep 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

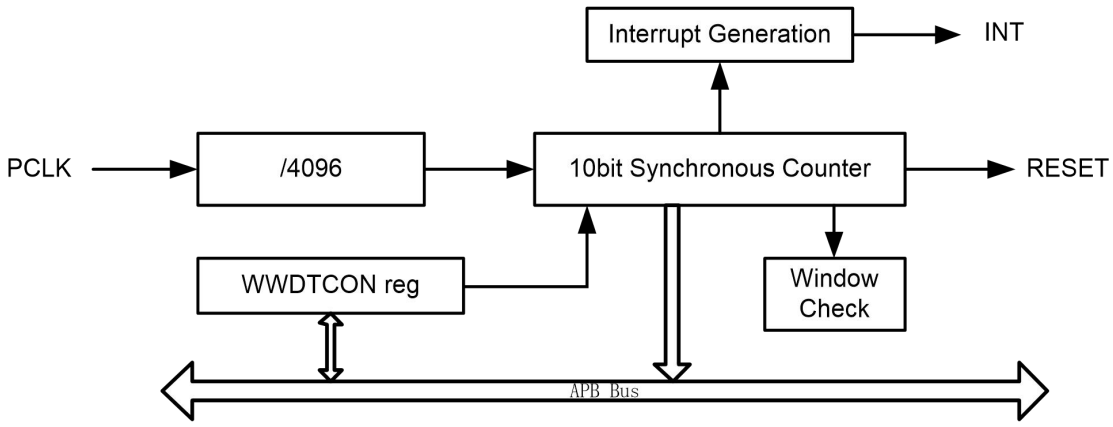


图 10-1WWDT 结构框图

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

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

下表为计算示例：

APBCLK 频率	溢出长度配置	溢出时间（ms）
48MHz	1	0.085
	4	0.341
	16	1.365
	64	5.461
	128	10.922
	256	21.845
	512	43.69
	1024	87.38
32MHz	1	0.128
	4	0.512
	16	2.048
	64	8.192
	128	16.384
	256	32.768
	512	64.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	129.072
	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。

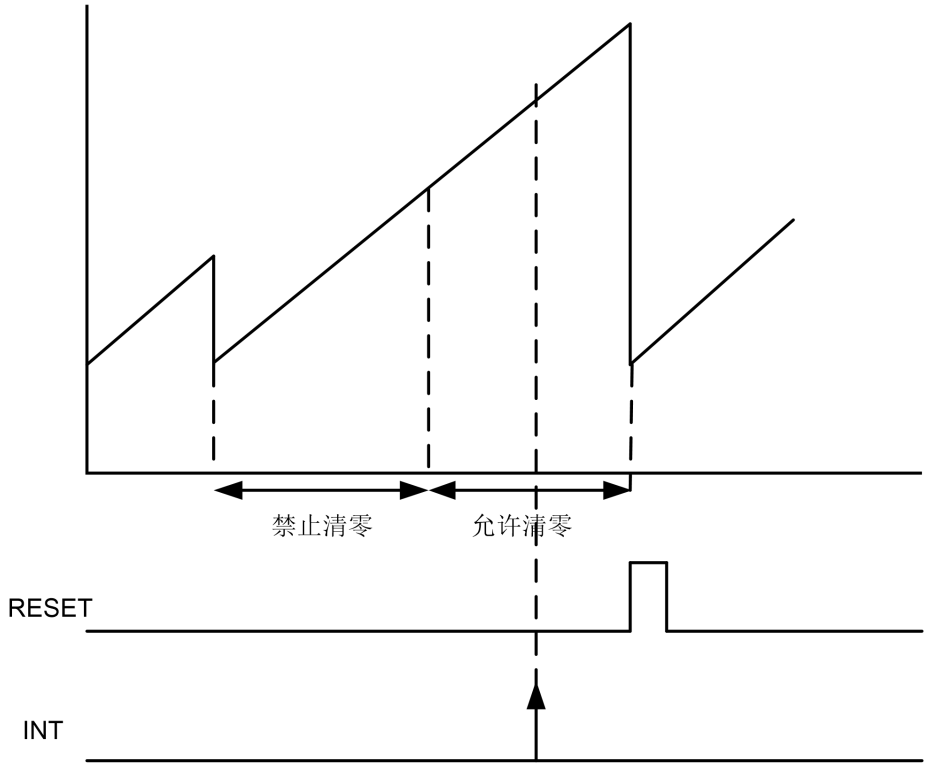


图 10-2WWDT 窗口示意图

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)、4~40MHz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（与 CPU 和总线时钟解耦）
- CPU 和总线最高频率 72MHz

11.2 时钟架构

11.2.1 时钟树

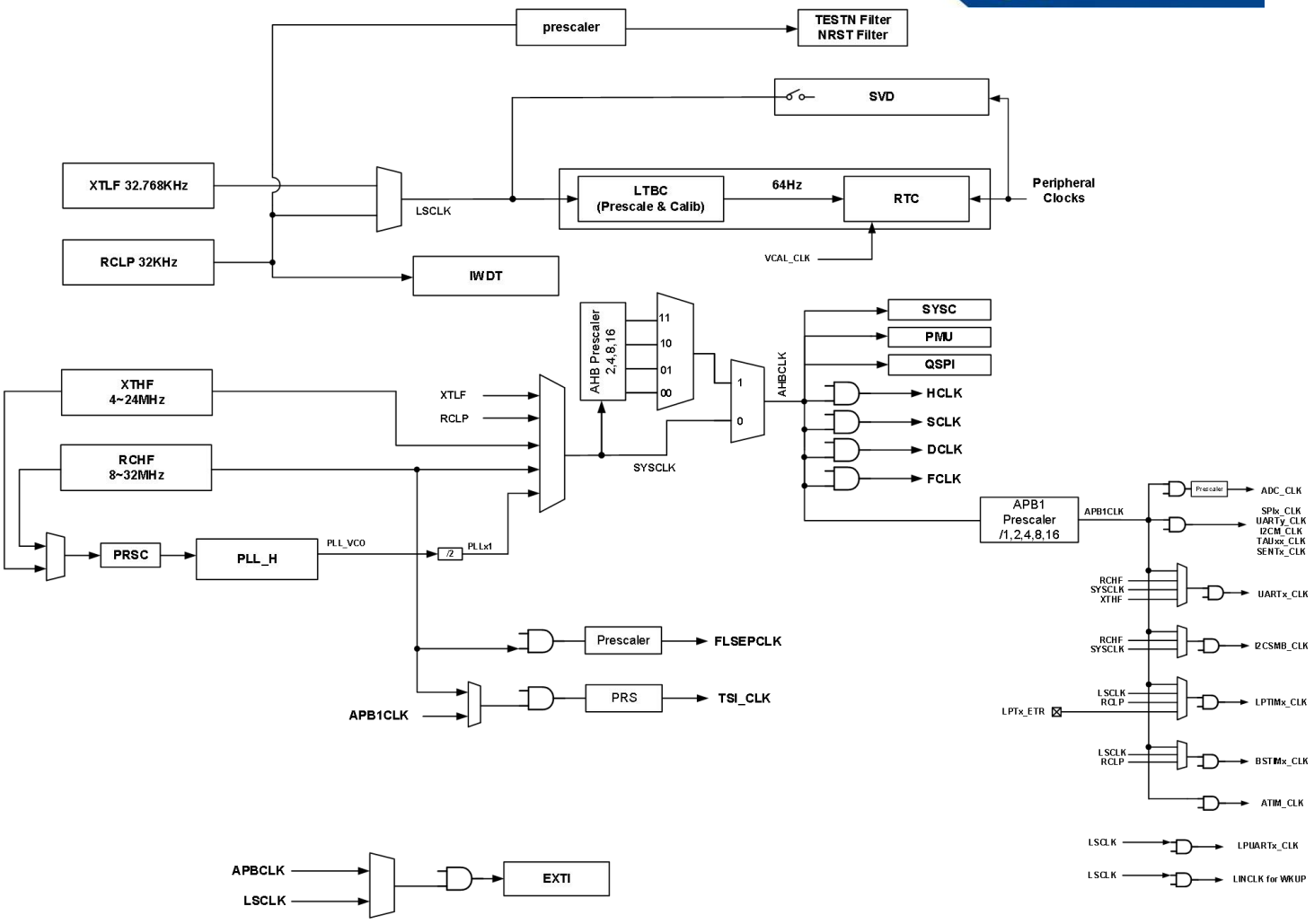


图 11-1 芯片时钟框图

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8MHzRCHF的不分频时钟作为系统主时钟，各外设模块的时钟可以分别独立控制。芯片工作时可以只打开需要工作的模块时钟，其他模块的时钟可关闭，以节省功耗。APB总线时钟APBCLK可以是AHBCLK的分频或同频时钟，用于驱动APB总线上的低速外设。

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	-

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。

在自动切换功能使能的情况下，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，同时复位AHBPRES寄存器、复位RCHF档位选择寄存器，确保系统AHB时钟为固定的RCHF不分频（RCHF频率选择寄存器FSEL是否复位由CKSAFE_CFG寄存器配置）。

- 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
- SYSCLK为PLL，PLL发生失锁，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	32KHz 低频系统时钟，晶体停振时可以切换到 RCLP 主要用于 RTC、IWDT、SVD、LCD
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF	32K~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

时钟	源头	说明
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	LSCLK
		RCHF
I2C_SMBUSx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
ATIM	APBCLK	APBCLK
LPTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		LPT32_ETR, LPT16_ETR
BSTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
ADC	APBCLK	APBCLK
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK

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

11.2.6 休眠模式下的外设时钟

Sleep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UARTx、LPUARTx、I2Cx、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 切换逻辑

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

11.2.8 RTC 时钟

RTC可以使用XTLF、RCLP工作。通常情况下，使用32768Hz的XTLF晶振来做RTC高精度计时。使用RCLP时，RTC走时精度无法保证。

11.3 高频晶体振荡电路(XTHF)

11.3.1 概述

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

XTHF可以适配4~40MHz晶体。软件可以通过XTHFEN寄存器使能或关闭XTHF时钟。

11.3.2 工作方式

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

11.3.3 停振检测（HFDET）

FM33FH0带有片上停振检测电路，与XTHF电路一起使能或关闭。停振检测使能后可以持续检测XTHF输出，当发现XTHF停振时，会产生报警中断，同时产生高级定时器刹车信号；如果XTHF正在被直接或者间接的用作系统工作时钟（直接指SYSCLK选为XTHF，间接指SYSCLK选为PLL同时PLL使用XTHF为输入参考时钟），则停振信号将自动使能RCHF并将SYSCLK切换到RCHF，以避免高频晶体意外停振导致系统异常。

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

HFDET停振检测阈值约200KHz，即XTHF时钟频率低于200KHz时触发停振报警；HFDET工作电流约1.2uA。

当晶振停振复位功能使能后，如果XTHF被用于系统时钟，并且检测到XTHF停振（LOC），系统将直接复位。

11.3.4 XTHF 启动过程

软件使能XTHF后，硬件等待XTHF输出稳定后自动置位RDY标志寄存器。

XTHF关闭或者HFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTHF后应等待RDY标志置位，再将XTHF作为系统时钟使用。

11.4 低频晶体振荡电路(XTLF)

11.4.1 概述

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

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

软件可以使能或关闭XTLF。为了提高抗干扰能力，采用4bit的XTLFEN控制位，4bit复位值为0101，必须改写为1010才能关闭XTLF，其他任何数据都会保持XTLF使能。

11.4.2 工作方式

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

注意：XTLF控制寄存器仅受POR和PDR复位影响，不受SVD、软件复位、NRST影响，以尽可能地延长RTC工作时间。所以即使使能了SVD，RTC也会保持工作直到发生PDR复位。

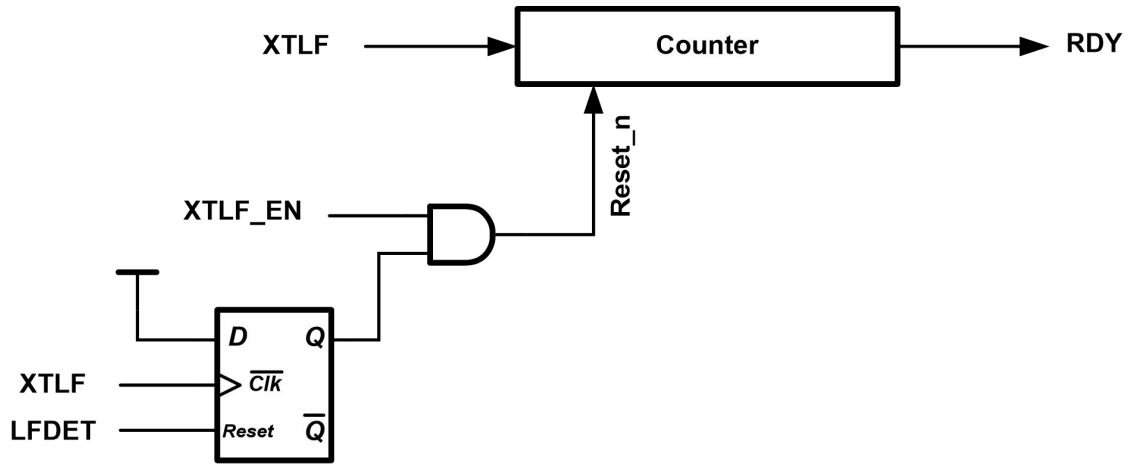
11.4.3 停振检测（LFDET）

芯片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，并且将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。

11.4.4 XTLF 启动过程

软件使能XTLF后，硬件等待XTLF输出稳定后自动置位RDY标志寄存器。



XTLF关闭或者LFDET检测到停振时，会自动清零计数器以及RDY标志。

软件使能XTLF后应等待RDY标志置位，再使用XTLF时钟。

11.5 低功耗低频环振(RCLP)

11.5.1 概述

RCLP是一个超低功耗内部环振，典型功耗仅400nA左右，典型输出频率32KHz。这个环振主要用作独立看门狗工作时钟，上电后自动启动，不可关闭。

11.6 锁相环(PLL)

11.6.1 概述

锁相环输入参考时钟可以是RCHF或XTHF分频，输入频率固定为1MHz，最高输出频率可达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。

11.6.2 锁定检测

PLL使能后，经过一段时间后输出锁定，此时PLL锁定标志置位。当PLL在工作中发生失锁，失锁标志置位（LOL），此时可以产生中断或者直接复位芯片。

11.6.3 应用注意

出于可靠性考虑，软件需注意以下几点：

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

配置PLL输出48MHz，并使系统以48MHz主频运行：

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

11.7 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	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	

11.8 休眠唤醒的时钟处理

当芯片从Sleep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。

11.9 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_XTHFCR
0x2C	CMU 中断使能寄存器（Interrupt Enable Register）	CMU_IER
0x30	CMU 中断标志寄存器（Interrupt Status Register）	CMU_ISR
0x34	外设总线时钟控制寄存器 1（Peripheral bus Clock Control Register1）	CMU_PCLKCR1
0x38	外设总线时钟控制寄存器 2（Peripheral bus Clock Control Register2）	CMU_ PCLKCR2
0x3C	外设总线时钟控制寄存器 3（Peripheral bus Clock Control Register3）	CMU_ PCLKCR 3
0x40	外设总线时钟控制寄存器 4（Peripheral bus Clock Control Register4）	CMU_ PCLKCR 4
0x44	外设工作时钟配置寄存器 1（Peripheral Clock Config Register1）	CMU_OPCCR1
0x48	外设工作时钟配置寄存器 2（Peripheral Clock Config Register 2）	CMU_OPCCR2
0x4C	外设工作时钟使能寄存器 1（Peripheral Clock Enable Register 1）	CMU_OPCER1
0x50	AHB Master 控制寄存器（AHB Master Control Register）	CMU_AHBMCR
-	-	-

11.9.1 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	SYSC_SW_LOC	SYSC_SW_LOL	-		LSCATS	-	SLP_ENEXTI	CKSAFE_CFG
位权限	R/W-0	R/W-0	U-0		R/W-1	U-0	R/W-1	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名			APB2PRES			APB1PRES		
位权限	U-0		R/W-000			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-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	STCLKSEL		-			SYSCLKSEL		
位权限	R/W-00		U-0			R/W-000		

Bit	助记符	功能描述
31	SYSC_SW_LOC	XTHF 停振时是否自动切换 SYSCLK 0：禁止自动切换到 RCHF 1：允许自动切换到 RCHF
30	SYSC_SW_LOL	PLL 失锁时是否自动切换 SYSCLK 0：禁止自动切换到 RCHF 1：允许自动切换到 RCHF
30:28	--	RFU：未实现，读为 0
27	LSCATS	LSCLK 自动切换使能 0：当检测到 XTLF 异常停振时，不会自动将 LSCLK 切换到 RCLP，软件可以通过写 LSCLKSEL 寄存器手动切换到 RCLP 1：当检测到 XTLF 异常停振时，自动使能 RCLP 并将 LSCLK 切换到 RCLP
26	--	RFU：未实现，读为 0
25	SLP_ENEXTI	Sleep/DeepSleep 模式下 EXTI 采样设置 1：Sleep/DeepSleep 模式下使能外部引脚中断采样（采样时钟为 LSCLK） 0：Sleep/DeepSleep 模式下禁止外部引脚中断采样（将无法产生 EXTI 中断）
24	CKSAFE_CFG	当系统时钟出现停振时，SYSCLK 自动切换的同时 RCHF FSEL 寄存器是否复位 0：复位 FSEL，RCHF 为 8MHz 1：不复位 FSEL，RCHF 保持为 FSEL 配置频率
24:22	--	RFU：未实现，读为 0
21:19	APB2PRES	APB2 时钟分频选择 0xx：不分频 1xx：2 分频
18:16	APB1PRES	APB1 时钟分频选择 0xx：不分频 100：2 分频

Bit	助记符	功能描述
		101：4 分频 110：8 分频 111：16 分频
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 工作时钟选择 00：SCLK 01：LSCLK 10：RFU 11：SYSCLK
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：RFU 110：RCLP 111：RCHF 注意：当系统时钟源发生停振时，自动复位此寄存器

11.9.2 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

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 0111：40MHz 1111：48MHz 其他：RFU
15:1	--	RFU：未实现，读为 0
0	EN	RCHF 使能寄存器 1：使能 RCHF 0：关闭 RCHF

11.9.3 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，7'h00 表示频率最低，7'h7F 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。

11.9.4 PLL 控制寄存器（CMU_PLLCR）

名称	CMU_PLLCR							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-						DB	
位权限	U-0						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	DB							
位权限	R/W-0001 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC			OSEL	-	INSEL	EN
位权限	R/Dy-0	R/W-000			R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25:16	DB	PLL 倍频比，输出时钟频率为 1M*(PLLDB+1) 0x1F：输出 32 倍频 0x2F：输出 48 倍频 0x3F：输出 64 倍频 0x47：输出 72 倍频
15:8	--	RFU：未实现，读为 0
7	LOCKED	PLL 锁定标志，软件通过查询此寄存器确认 PLL 已经处于锁定状态 1：PLL 已锁定 0：PLL 未锁定
6:4	REFPRSC	PLL 参考时钟预分频（目标是产生 1MHz 参考时钟给 PLL） 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：12 分频 101：16 分频 110：24 分频 111：32 分频
3	OSEL	DUMMY 寄存器
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

11.9.5 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	常使能，无效

11.9.6 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
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

11.9.7 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	ST_WAIT			XTLF_BYPASS	XTLFEN			
位权限	R/W-111			R/W-0	R/W-0			

位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RDY	-		GMCFG			ICFG	
位权限	R-0	U-00		R/W-100			R/W-01	

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
15:13	ST_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	XTLF_BYPASS	XTLF 外部输入使能： 0：使用晶振产生 XTLF 1：使用 XT32KI 引脚输入外部时钟
11:8	XTLFEN	XTLF 使能寄存器 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 发生停振时自动清零
6:5	--	RFU：未实现，读为 0
4:2	GMCFG	XTLF 的跨导配置寄存器
1:0	ICFG	XTLF 的电流源配置寄存器

11.9.8 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	-							LSSTA
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LSCLKSEL							
位权限	R/W-xxxx xxxx							

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0

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

11.9.9 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	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	ST_WAIT			HF_CFG				
位权限	R/W-111			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			XTLF_BYPASS	-		RDY	XTHFEN
位权限	U-0			R/W-0	U-0		R-0	R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30:19	--	RFU：未实现，读为 0
18:16	--	RFU：未实现，读为 0
15:13	ST_WAIT	XTHF 启动等待时间配置，单位是 XTHF 时钟周期 000：128 001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12:8	HF_CFG	XTHF 振荡强度配置 00000：最弱 11111：最强

Bit	助记符	功能描述
7:5	--	RFU：未实现，读为 0
4	XTHF_BYPASS	XTHF 外部输入使能： 0：使用晶振产生 XTHF 1：使用 XTHF_IN 引脚输入外部时钟
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 发生停振时自动清零
0	XTHFEN	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

11.9.10 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	-				LOL_IE	SYSCSE_IE	HFDET_IE	LFDET_IE
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	LOL_IE	PLL 失锁中断使能，1 有效
2	SYSCKE_IE	SYSCLK 时钟选择错误中断使能寄存器，1 有效
1	HFDET_IE	XTHF 高频检测报警中断使能，1 有效
0	LFDET_IE	XTLF 低频检测报警中断使能，1 有效

11.9.11 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-1	R-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				LOLIF	SYSCSE_IF	HFDETIF	LFDETIF
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
31:10	--	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
7:4	--	RFU：未实现，读为 0
3	LOLIF	PLL 失锁中断标志 在 PLL 锁定的情况下发生失锁，此标志置位。软件写 1 清零。
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

11.9.12 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	-			CLM2_PCE	CLM1_PCE	CLM0_PCE	-	
位权限	U-000			R/W-0	R/W-0	R/W-0	U-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-			SENT1_PCE	SENT0_PCE	-		
位权限	U-000			R/W-0	R/W-0	U-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						COMP_PCE	SVD_PCE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD_PCE	ANT_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RTC_PCE	-	LPT32_PCE
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

Bit	助记符	功能描述
31:29	--	RFU：未实现，读为 0
28	CLM2_PCE	CLM2 总线时钟使能，高使能
27	CLM1_PCE	CLM1 总线时钟使能，高使能
26	CLM0_PCE	CLM0 总线时钟使能，高使能
25:21	--	RFU：未实现，读为 0
20	SENT1_PCE	SENT1 总线时钟使能，高使能
19	SENT0_PCE	SENT0 总线时钟使能，高使能
18:10	--	RFU：未实现，读为 0
9	COMP_PCE	Comparator 总线时钟使能，高使能
8	SVD_PCE	SVD 总线时钟使能，高使能
7	PAD_PCE	PADCFG 总线时钟使能，高使能
6	ANT_PCE	模拟测试 buffer 总线时钟使能，高使能 此寄存器用于控制 BUF4TST 的总线时钟
5	IWDT_PCE	IWDT 总线时钟使能，高使能
4	SCU_PCE	SCU 总线时钟使能，高使能
3	PMU_PCE	PMU 总线时钟使能，高使能
2	RTC_PCE	RTC 总线时钟使能，高使能
1	--	RFU：未实现，读为 0
0	LPT_PCE	LPTIM 总线时钟使能，高使能

11.9.13 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	-				TSI_PCE	DAC_PCE	-	SARADC_PCE
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	WWDT_PCE	RAMBIST_PCE	NVM_PCE	DMA_PCE	-	-	-	CRC_PCE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11	TSI_PCE	TSI 总线时钟使能，高使能
10	DAC_PCE	DAC 总线时钟使能，高使能
9	--	RFU：未实现，读为 0
8	SARADC_PCE	SARADC 总线时钟使能，高使能
7	WWDT_PCE	WWDT 总线时钟使能，高使能
6	RAMBIST_PCE	RAMBIST 总线时钟使能，高使能

Bit	助记符	功能描述
5	NVM_PCE	NVMIF（Flash 擦写控制器）总线时钟使能，高使能
4	DMA_PCE	DMA 总线时钟使能，高使能
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0
0	CRC_PCE	CRC 总线时钟使能，高使能

11.9.14 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	-	-	-	-	-	-	I2C1_PCE	I2C0_PCE
位权限	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	-	-		-	UART7_PCE	UART6_PCE	LPUART0_PCE
位权限	U-0	U-0	U-0		U-0	R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	UCIR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	UART2_PCE	UART1_PCE	UART0_PCE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_PCE	-		-	-	SPI2_PCE	SPI1_PCE	SPI0_PCE
位权限	R/W-0	U-0		U-0	U-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31	--	RFU：未实现，读为 0
30	--	RFU：未实现，读为 0
29	--	RFU：未实现，读为 0
28	--	RFU：未实现，读为 0
27	-	RFU：未实现，读为 0
26	-	RFU：未实现，读为 0
25	I2C1_PCE	I2C1 总线时钟使能，高有效
24	I2C0_PCE	I2C0 总线时钟使能，高有效
23	-	RFU：未实现，读为 0
22	-	RFU：未实现，读为 0
21:20	-	RFU：未实现，读为 0
19	-	RFU：未实现，读为 0
18	UART7_PCE	UART7 总线时钟使能，高有效
17	UART6_PCE	UART6 总线时钟使能，高有效
16	LPUART0_PCE	LPUART 总线时钟使能，高有效
15	-	RFU：未实现，读为 0
14	UCIR_PCE	UART 红外调制工作时钟使能，高有效
13	UART5_PCE	UART5 总线时钟使能，高有效
12	UART4_PCE	UART4 总线时钟使能，高有效

位号	助记符	功能描述
11	UART3_PCE	UART3 总线时钟使能，高有效
10	UART2_PCE	UART2 总线时钟使能，高有效
9	UART1_PCE	UART1 总线时钟使能，高有效
8	UART0_PCE	UART0 总线时钟使能，高有效
7	QSPI_PCE	QSPI 总线时钟使能，高有效
6:3	-	RFU：未实现，读为 0
2	SPI2_PCE	SPI2 总线时钟使能，高有效
1	SPI1_PCE	SPI1 总线时钟使能，高有效
0	SPI0_PCE	SPI0 总线时钟使能，高有效

11.9.15 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	-							-
位权限	U-0							U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AT_PCE	-	TAU1_PCE	TAU0_PCE	-	-	-	BT32_PCE
位权限	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0

位号	助记符	功能描述
31:9	-	RFU：未实现，读为 0
8	-	RFU：未实现，读为 0
7	AT_PCE	高级定时器总线时钟使能，高有效
6	-	RFU：未实现，读为 0
5	TAU1_PCE	定时器阵列 1 总线时钟使能，高有效
4	TAU0_PCE	定时器阵列 0 总线时钟使能，高有效
3	-	RFU：未实现，读为 0
2	-	RFU：未实现，读为 0
1	-	RFU：未实现，读为 0
0	BT32_PCE	BSTIM32 总线时钟使能，高有效

11.9.16 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	-		-		LPUART0CKS	
位权限	U-0	R/W-0	U-0		U-0		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		-		UART5CKS		UART4CKS	
位权限	U-0		U-0		R/W-00		R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		BTCKS		-		LPTCKS	
位权限	U-0		R/W-00		U-0		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	UART3CKS		UART2CKS		UART1CKS		UART0CKS	
位权限	R/W-00		R/W-00		R/W-00		R/W-00	

位号	助记符	功能描述
31	-	RFU：未实现，读为 0
30	EXTICKS	EXTI 中断采样时钟选择 1：外部引脚中断使用 LSCLK 采样 0：外部引脚中断使用 AHBCLK 采样 *建议在关闭所有 EXTI 中断的情况下设置，设置完成后再使能 EXTI 中断
29:28	-	RFU：未实现，读为 0
27:26	-	RFU：未实现，读为 0
25:24	LPUART0CKS	LPUART0 工作时钟选择 00：LSCLK 01：RCHF 分频（根据 RCHF 档位自动分频到 32768Hz 附近） 10：RFU 11：RFU
23:20	-	RFU：未实现，读为 0
21:20	-	RFU：未实现，读为 0
19:18	UART5CKS	UART5 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
17:16	UART4CKS	UART4 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
15:14	-	RFU：未实现，读为 0
13:12	BTCKS	BSTIM 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：XTLF
11:10	-	RFU：未实现，读为 0
9:8	LPTCKS	LPTIM 工作时钟选择 00：APBCLK 01：LSCLK

位号	助记符	功能描述
		10：RCLP 11：XTLF
7:6	UART3CKS	UART3 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
5:4	UART2CKS	UART2 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
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

11.9.17 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-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTCPRSC		-	RTCCKS	-			
位权限	R/W-00		U-0	R/W-0	U-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	-							TSICKS
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:22	RTCPRSC	XTHF 的分频配置，时钟频率为 XTHF*(1+RTCPRSC)
21	--	RFU：未实现，读为 0
20	RTCCKS	RTC 工作时钟选择 0：LSCLK 1：XTHF 分频
19:1	--	RFU：未实现，读为 0
0	TSICKS	TSI 工作时钟选择

Bit	助记符	功能描述
		0：RCHF 1：APBCLK

11.9.18 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	-	-	LPUART0CKE	-		-
位权限	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0		U-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		-	-	-		-	TSICKE
位权限	U-0		U-0	U-0	U-0		U-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	UART5CKE	UART4CKE	LINCKE	UART3CKE	UART2CKE	UART1CKE	UART0CKE
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	-				-	BT32CKE	-	LPT32CKE
位权限	U-0				U-0	R/W-0	U-0	R/W-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
29	--	RFU：未实现，读为 0
28	--	RFU：未实现，读为 0
27	LPUART0CKE	LPUART0 工作时钟使能，高有效
26:25	--	RFU：未实现，读为 0
24	--	RFU：未实现，读为 0
23:22	--	RFU：未实现，读为 0
21	--	RFU：未实现，读为 0
20	--	RFU：未实现，读为 0
19:18	--	RFU：未实现，读为 0
17	--	RFU：未实现，读为 0
16	TSICKE	TSI 模块工作时钟使能，高有效
15	--	RFU：未实现，读为 0
14	UART5CKE	UART5 工作时钟使能，高有效
13	UART4CKE	UART4 工作时钟使能，高有效
12	LINCKE	LINCLK 使能，用于唤醒信号检测，高有效
11	UART3CKE	UART3 工作时钟使能，高有效
10	UART2CKE	UART2 工作时钟使能，高有效
9	UART1CKE	UART1 工作时钟使能，高有效
8	UART0CKE	UART0 工作时钟使能，高有效
7:4	--	RFU：未实现，读为 0
3	--	RFU：未实现，读为 0

Bit	助记符	功能描述
2	BTCKE	BSTIM 工作时钟使能，高有效
1	--	RFU：未实现，读为 0
0	LPTCKE	LPTIM 工作时钟使能，高有效

11.9.19 AHB Master 控制寄存器（CMU_AMCR）

名称	CMU_AMCR							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFUI	RFUI	-					
位权限	R/W-1	R/W-0	U-0					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							MPRIL
位权限	U-0							R/W-0

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

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3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。

3个CLM分别用于监控PLL、XTHF、RCHF

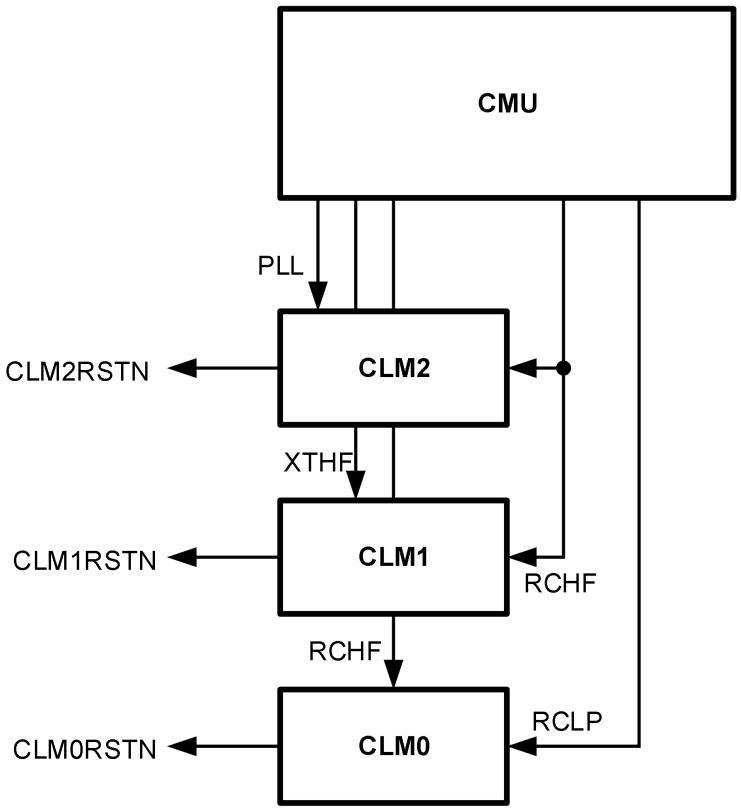


图 12-1CLM 全局示意图

12.2 结构框图

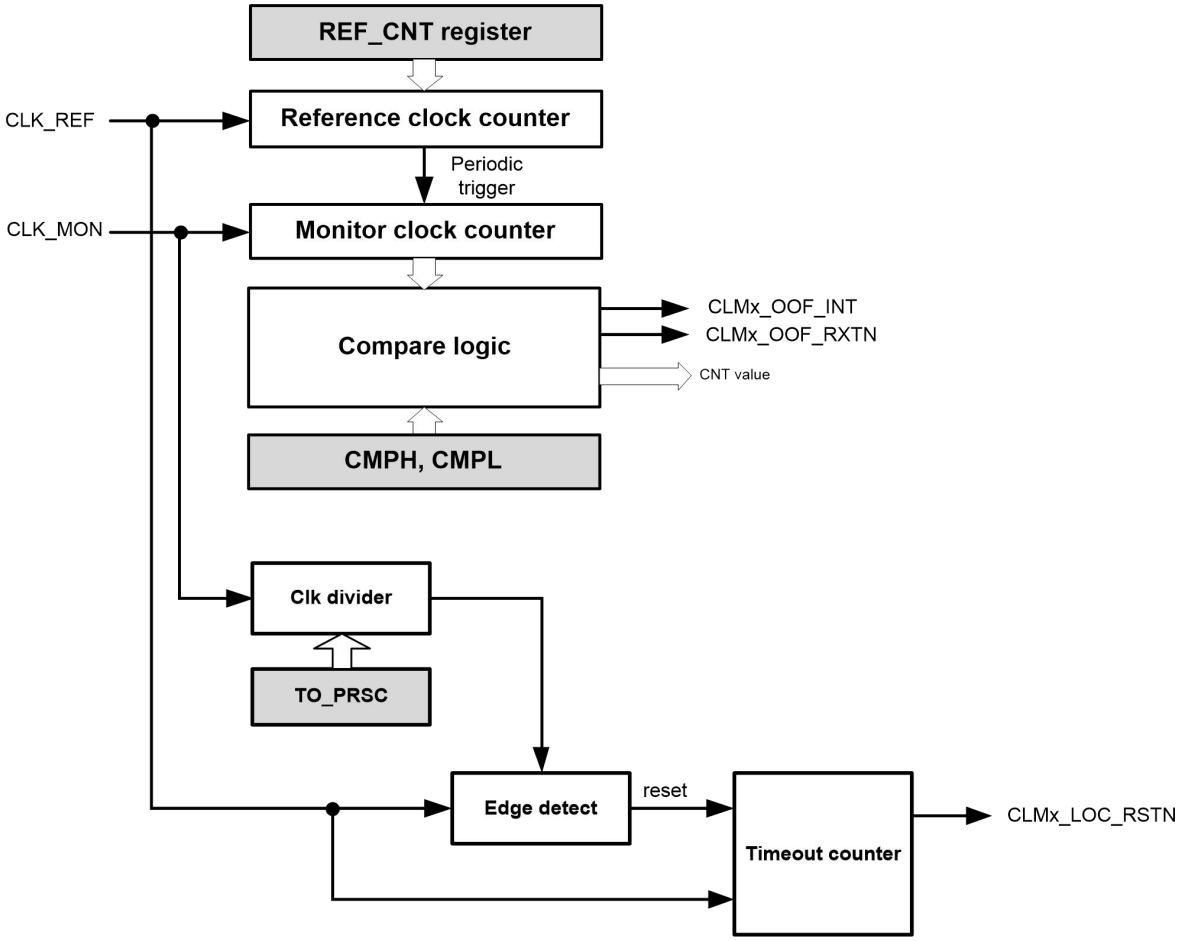


图 12-2CLM 电路框图

其中reference clock counter位宽16bit，monitor clock counter位宽24bit

每个CLM模块的CLK_REF和CLK_MON是固定的。

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，由软件完成配置后来使能，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

CLM与芯片低功耗模式

当芯片进入休眠模式或LPRUN模式时，由于CLK_MON的源头时钟都被自动关闭，此时CLM模块应停止工作，硬件自动关闭CLM，芯片回到active模式时，软件需要重新配置并使能CLM模块。

CLM不能在芯片唤醒时自动使能，因为休眠时RCHF、XTHF、PLL都会被硬件强制关闭，唤醒后软件可能修改时钟频率，CLK_MON频率可能与休眠前不同，甚至可能软件没有使能XTHF和PLL，如果CLM自动恢复休眠前的工作状态，则可能导致意外的频率报警。

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。

注意：频率监控不能检测CLK_MON完全停止的异常状态，这部分安全机制由超时计数器提供。

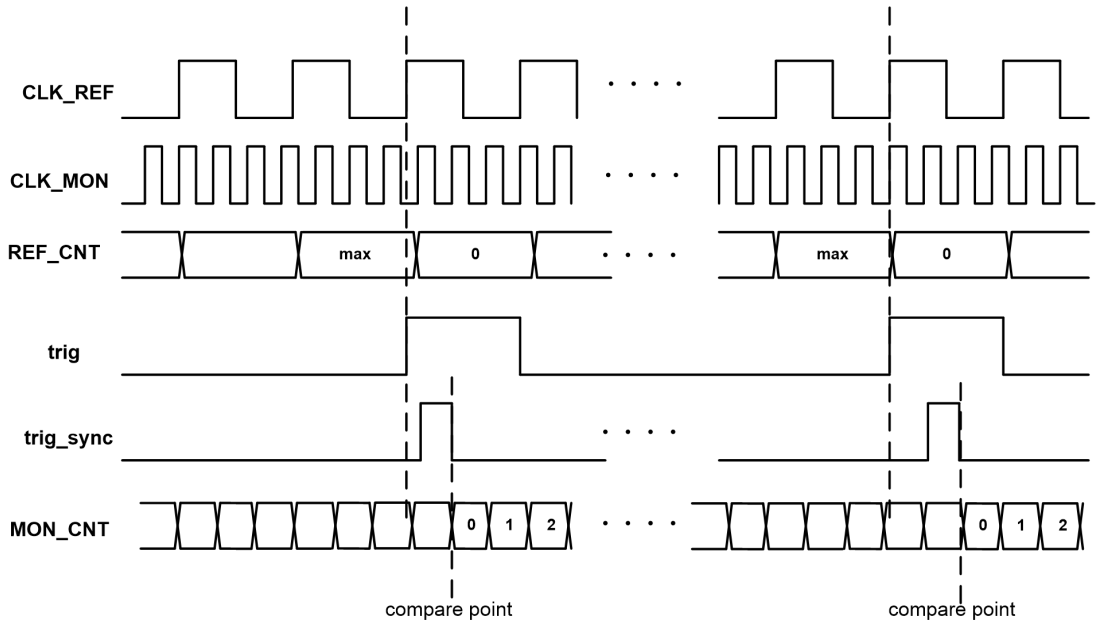


图 12-3 参考时钟计数和监控时钟计数

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。

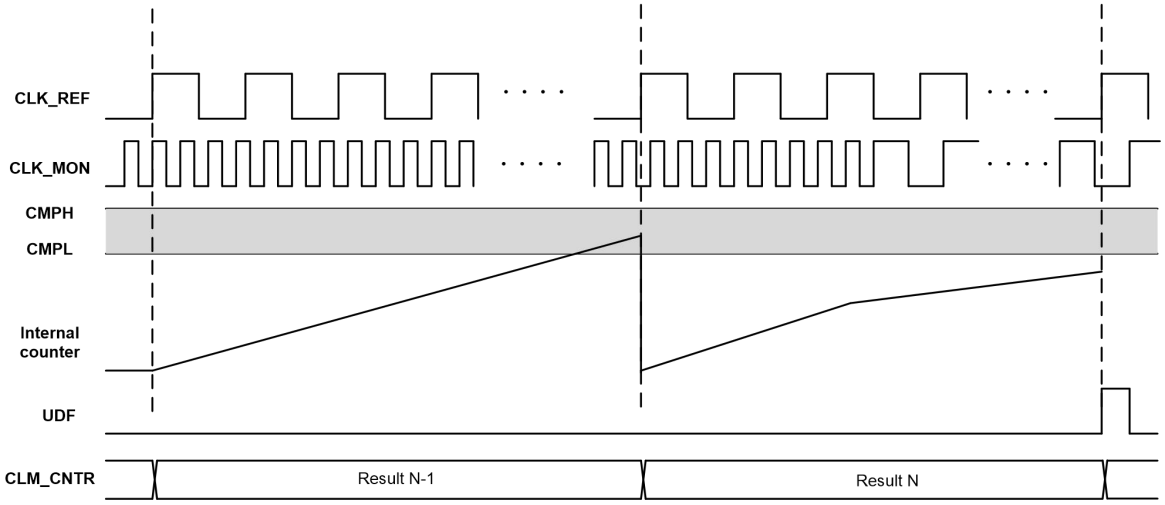


图 12-4 监控时钟频率下溢出

下图为监控时钟频率上溢出的例子。

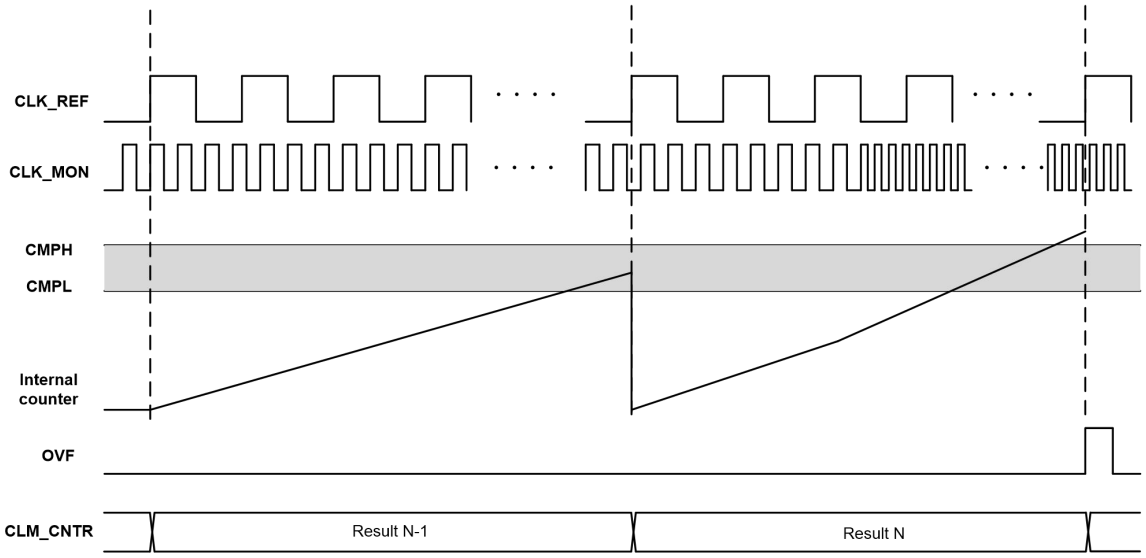


图 12-5 监控时钟频率上溢出

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：

例子1：

- 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RCHF，温度系数小于5%
- 检测PLL时钟丢失的最大故障响应时间（FTTI）是250us
- 配置RCHF为8Mhz，可设置REF_CLK计数周期为1440，即检测周期180us，考虑温度系数最大不超过200us
- 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

例子2：

- 系统使用PLL=120MHz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RCHF，温度系数小于5%
- 检测PLL时钟丢失的最大故障响应时间（FTTI）是120us
- 配置RCHF为8Mhz，可设置REF_CLK计数周期为800，即检测周期100us，考虑温度系数最大不超过110us
- 监控时钟在检测周期内的理想计数值应该是12000
- 设置CMPL=0.8*12000=9600，CMPH=1.2*11520=14400

例子3：

- 系统使用XTHF=40MHz为主时钟工作
- 监控时钟为XTHF，温度系数小于100ppm
- 参考时钟为RCHF，温度系数小于5%
- 检测XTHF时钟丢失的最大故障响应时间（FTTI）是250us
- 配置RCHF为8Mhz，可设置REF_CLK计数周期为1600，即检测周期200us，考虑温度系数最大不超过220us
- 监控时钟在检测周期内的理想计数值应该是8000
- 设置CMPL=0.8*8000=6400，CMPH=1.2*8000=9600

例子4：

- 系统使用RCHF=16MHz为主时钟工作
- 监控时钟为RCHF，温度系数小于5%
- 参考时钟为RCLP，温度系数小于10%
- 检测RCHF时钟丢失的最大故障响应时间（FTTI）是250us
- RCLP为32Khz，可设置REF_CLK计数周期为6，即检测周期187us，考虑温度系数最大不超过206us
- 监控时钟在检测周期内的理想计数值应该是2992
- 设置CMPL=0.8*2992=2394，CMPH=1.2*2992=3590

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。

超时计数器使用CLK_REF计数，溢出长度可配置；CLK_MON经过可配置比例的分频后，输入边沿检测电路，产生同步于CLK_REF的边沿脉冲，用于清零计数器。因此在CLK_MON正常翻转的情况下，清零脉冲周期性出现，超时计数器不会溢出。当CLK_MON长时间停止，超时计数器最终溢出，产生全局复位信号。

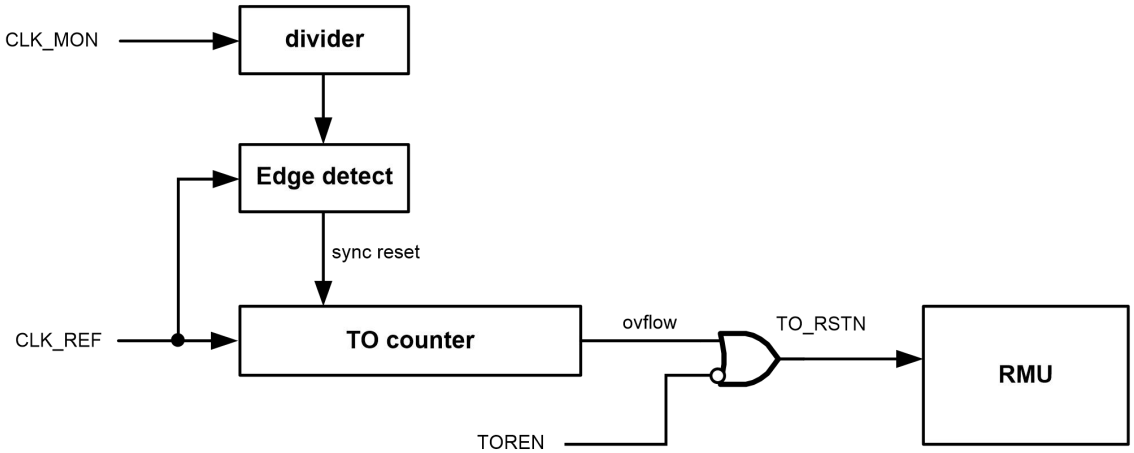


图 12-6 超时检测电路

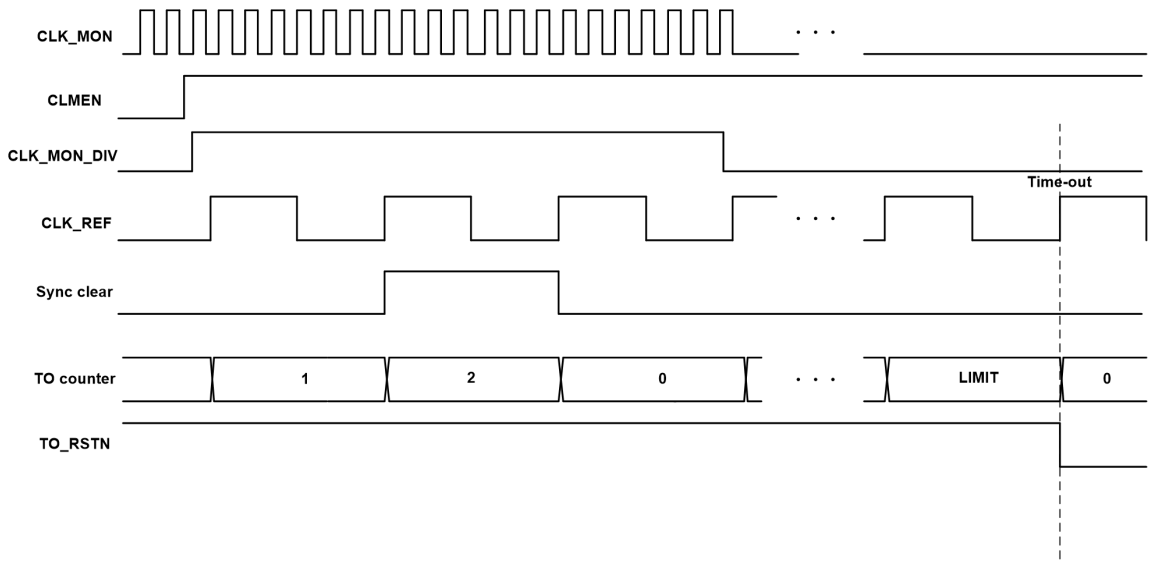


图 12-7Time-out 复位

注意：超时功能只有复位没有中断

使用超时检测需要正确配置CLK_MON分频和超时溢出上限。

例子1：

- CLK_MON=RCHF 8MHz，温度系数小于5%；CLK_REF=RCLP，温度系数小于10%
- 配置CLK_MON分频比为2000，得到被监控时钟4KHz，每隔125us左右产生一个复位沿
- TO溢出上限设为32，即溢出周期1000us左右
- 超过1000us没有检测到CLK_MON边沿，产生全局复位，因此最大FTTI=1ms

12.4 寄存器

CLM0模块起始地址：0x4001_C000

CLM1模块起始地址：0x4001_C400

CLM2模块起始地址：0x4001_C800

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
0x18	超时检测配置寄存器	CLM_TOCFGR

12.4.1 时钟监控控制寄存器（CLM_CR）

名称	CLM_CRx (x=0,1,2)							
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 *此寄存器主要用于 debug 目的
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 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 *当芯片退出 Active 模式时，EN 位自动清零，CLM 关闭并回到初始态；芯片唤醒后，软件需要根据系统时钟设置，重新配置并使能 CLM

12.4.2 时钟监控配置寄存器（CLM_CFGR）

名称	CLM_CFGRx (x=0,1,2)
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	REF_CNT							
位权限	R/W-10000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REF_CNT							
位权限	R/W-00000000							
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
31:24	--	RFU：未实现，读为 0
23:8	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	RFUI

12.4.3 时钟监控计数值寄存器（CLM_CNTR）

名称	CLM_CNTRx (x=0,1,2)							
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	CNT[23:16]							
位权限	R-0000 0000							
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
31:24	--	RFU：未实现，读为 0
23:0	CNT	时钟监控计数器计数值 (Clock calibration counter) 每个监控周期结束后自动更新，供软件查询

12.4.4 时钟监控中断标志寄存器（CLM_ISR）

名称	CLM_ISRx (x=0,1,2)							
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	CCL_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=0,1,2)							
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=0,1,2)							
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

12.4.7 超时检测配置寄存器（CLM_TOCFGR）

名称	CLM_TOCFGRx (x=0,1,2)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TO_LIMIT[15:8]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TO_LIMIT[7:0]							
位权限	R/W-1000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CLKMON_PRSC[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CLKMON_PRSC [7:0]							
位权限	R/W-1000 0000							

Bit	助记符	功能描述
31:16	TO_LIMIT	超时溢出上限，为 CLK_REF 的计数值
15:0	CLKMON_PRSC	超时检测 CLK_MON 预分频寄存器 超时检测对 CLK_MON 做停振检测前，对 CLK_MON 做(CLKMON_PRSC+1)分频

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，并给出中断信号或者复位。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中断
- 支持欠压复位功能
- 低压检测范围 2.3V~4.8V，13 级可编程阈值档位，档位间隔 0.214V
- 电压检测迟滞窗口 0.1V
- 可关断或间歇式工作（仅报警功能）
- 支持 1 个外部通道直接输入与内部基准电压源比较

13.2 结构框图

下图是电源检测电路的模块框图。

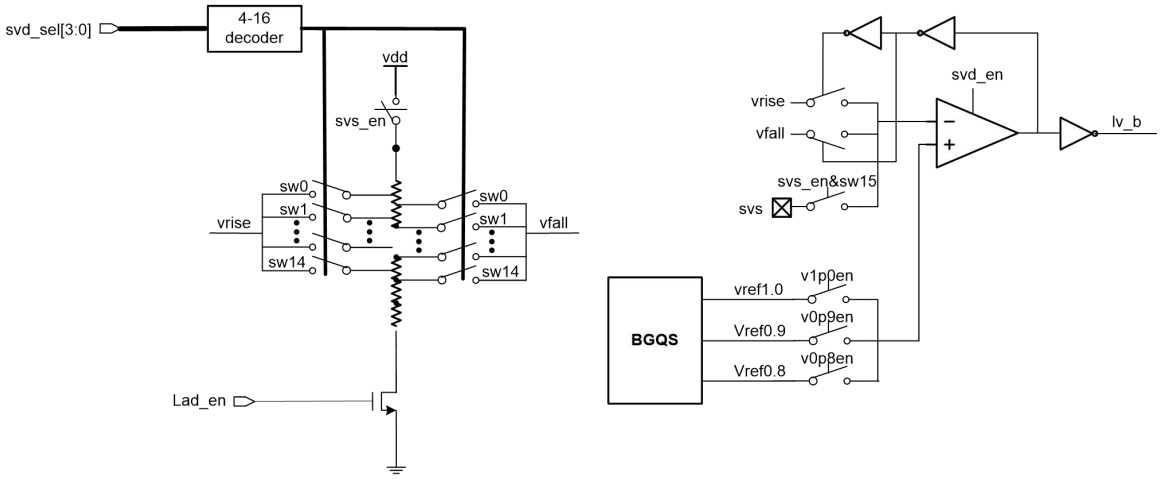


图 13-1 低压检测电路框图

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：

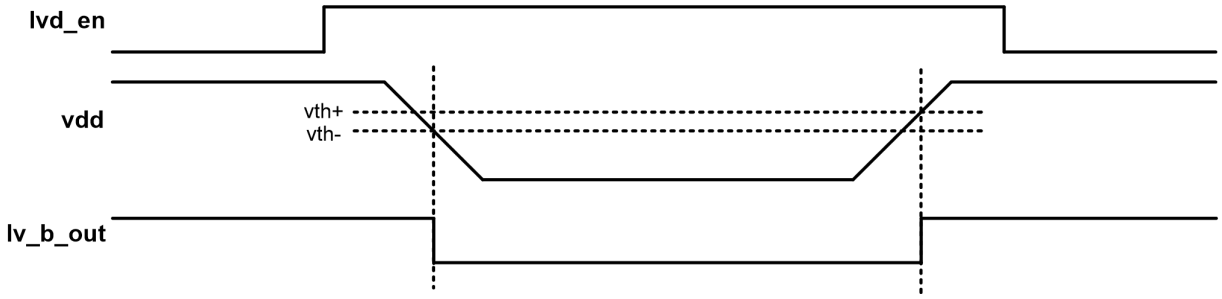


图 13-2 低压检测电路工作时序

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 功能描述

SVD支持低压报警和欠压复位功能，由软件配置和使能。

上电复位释放后，软件可以选择工作模式并使能SVD。注意使用低压复位功能时，软件需要保证电源电压高于SVD复位阈值，否则一旦使能SVD低压复位功能，芯片就将立即复位。

13.4.1 低压报警

SVDEN=01时，使能低压报警功能。

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13级检测电平，检测范围2.3V~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。

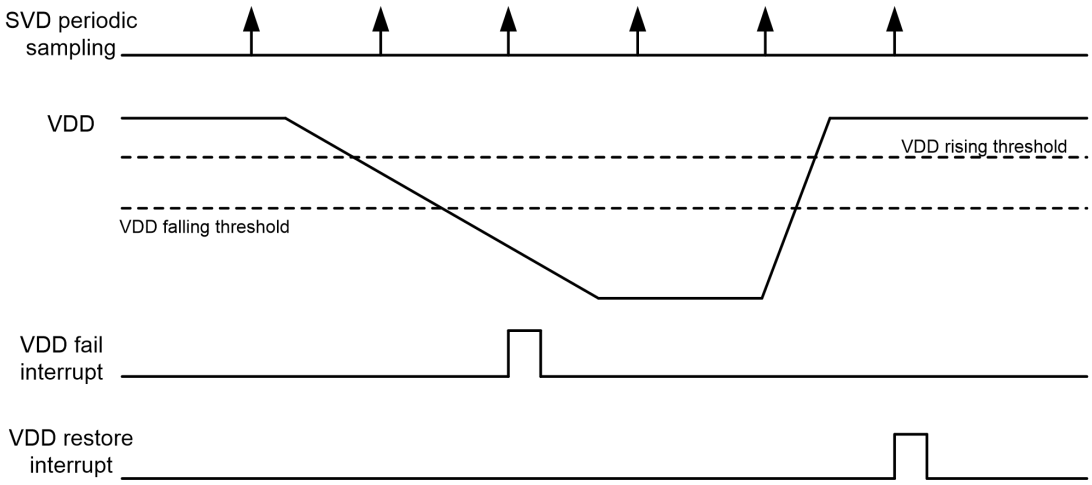


图 13-3 电源检测电路间歇工作模式

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LSCLK时钟同步周期后，SVD就会开始工作。SVD开启后到输出稳定建立大约需要100us时间，软件读取SVD输出时需要注意。

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

工作模式说明：

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

13.4.2 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，此时间歇启动SVD的同时还要启动AVREF，数字电路需要先等待AVREF建立，才能采样SVD输出。AVREF启动时间小于5us，SVD比较器建立时间不超过100us；在等待足够长时间后，数字电路才锁存SVD输出信号。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。

注意：间歇使能模式仅适用于低压报警功能，不能用于欠压复位功能。

13.4.3 欠压复位

SVDEN=10时，使能SVD欠压复位功能,此时功耗会增加约2uA。

欠压复位模式下，当电源电压低于LVL寄存器定义的阈值时，SVD将产生全局复位输出。

当电源电压高于LVL定义的阈值时，复位释放。

软件使能SVD复位前，须保证芯片电源高于复位阈值，以避免一使能SVD芯片立即复位。

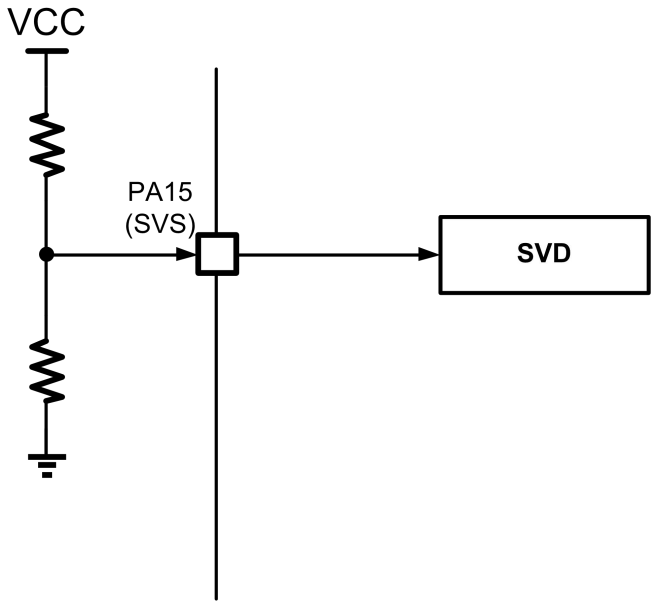
SVD欠压复位的滤波时长约4us，SVD欠压复位恢复的滤波时长约2ms。

13.4.4 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
0	X	外部电源检测通道关闭，仅检测内部电源电压
1	1111	外部电压输入不做内部分压，直接输入到比较器与内部基准电压比较
	0000~1110	外部电压输入先经过内部电阻分压，然后再输入到比较器与内部基准电压比较 分压后的档位参见后续章节描述

13.5 电源检测阈值

通过SVSEN和SVDLVL寄存器可以选择电压检测对象和检测阈值。

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.443	2.543
0001	2.443	2.543
0010	2.443	2.543
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043

1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	N/A	N/A

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.420	2.520
0001	2.420	2.520
0010	2.420	2.519
0011	2.415	2.520
0100	2.630	2.732
0101	2.845	2.945
0110	3.055	3.156
0111	3.270	3.367
1000	3.480	3.576
1001	3.690	3.787
1010	3.900	3.997
1011	4.105	4.205
1100	4.315	4.412
1101	4.520	4.619
1110	4.725	4.826
1111	N/A	N/A

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.405	2.508
0001	2.405	2.508
0010	2.410	2.509
0011	2.405	2.508
0100	2.620	2.720
0101	2.830	2.932
0110	3.045	3.144
0111	3.255	3.353
1000	3.465	3.563
1001	3.675	3.773
1010	3.880	3.982
1011	4.090	4.189
1100	4.295	4.396
1101	4.505	4.603
1110	4.710	4.807
1111	N/A	N/A

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.443	2.543
0001	2.443	2.543
0010	2.443	2.543
0011	2.443	2.543
0100	2.657	2.757
0101	2.871	2.971
0110	3.086	3.186
0111	3.300	3.400
1000	3.514	3.614
1001	3.729	3.829
1010	3.943	4.043
1011	4.157	4.257
1100	4.371	4.471
1101	4.586	4.686
1110	4.800	4.900
1111	1.0	1.0

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.420	2.520
0001	2.420	2.520
0010	2.420	2.519
0011	2.415	2.520
0100	2.630	2.732
0101	2.845	2.945
0110	3.055	3.156
0111	3.270	3.367
1000	3.480	3.576
1001	3.690	3.787
1010	3.900	3.997
1011	4.105	4.205
1100	4.315	4.412
1101	4.520	4.619
1110	4.725	4.826
1111	0.95	0.95

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

SVDLVL	下降阈值（V）	上升阈值（V）
0000	2.405	2.508
0001	2.405	2.508
0010	2.410	2.509
0011	2.405	2.508
0100	2.620	2.720
0101	2.830	2.932

0110	3.045	3.144
0111	3.255	3.353
1000	3.465	3.563
1001	3.675	3.773
1010	3.880	3.982
1011	4.090	4.189
1100	4.295	4.396
1101	4.505	4.603
1110	4.710	4.807
1111	0.9	0.9

13.6 寄存器

模块偏移地址：0x4001_A800

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

13.6.1 SVD 配置寄存器（SVD_CFGR）

名称	SVD_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LVL				DFEN	MOD	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 13.5 电源检测阈值 (SVD threshold level)
3	DFEN	数字滤波使能（SVDMODE=1 时必须置 1）(Digital Filter Enable)，仅低压报警模式下有效，滤波时长为 3 个 RCLP 周期。 1：启动 SVD 输出的数字滤波 0：关闭 SVD 输出的数字滤波
2	MOD	SVD 工作模式选择，配置模式后还要置位 SVDEN 才会启动 SVD (SVD Mode) 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 间歇使能间隔 (SVD interval enable period) 00：62.5ms 01：256ms 10：1s

Bit	助记符	功能描述
		11：4s

13.6.2 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	-					SVSEN	SVDEN	
位权限	U-0					R/W-0	R/W-0	

Bit	助记符	功能描述
31:9	--	RFU：未实现，读为 0
8	TE	SVD 保护 flash 擦写使能，仅 SVD 报警模式下有效 0：SVD 不保护 flash 擦写 1：SVD 保护 flash 擦写
7:3	--	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) 01：低压报警模式 10：欠压复位模式 00/11：关闭 SVD

13.6.3 SVD 中断使能寄存器（SVD_IER）

名称	SVD_IER							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

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

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

13.6.4 SVD 状态和标志寄存器（SVD_ISR）

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

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

13.6.5 SVD 参考电压选择寄存器（SVD_VSR）

名称	SVD_VSR

offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit0	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					V1P0EN	V0P9EN	V0P8EN
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	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
0	V0P8EN	0.8V 基准输入使能信号 (0.8V reference enable) 1：使能 0.8V 基准输入 0：关闭 0.8V 基准输入

14 模拟比较器（Comparator）

14.1 概述

芯片集成2个比较器，支持以下特性：

- 2个rail-to-rail比较器，支持轨到轨输入，多种功耗模式
- 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
- 中断事件可唤醒MCU
- 输出信号可连接到GPIO，也可以作为触发源连接到定时器、ADC
- 内建输入基准Buffer
- 窗口比较器功能
- 可编程的输出数字滤波功能

14.2 结构框图

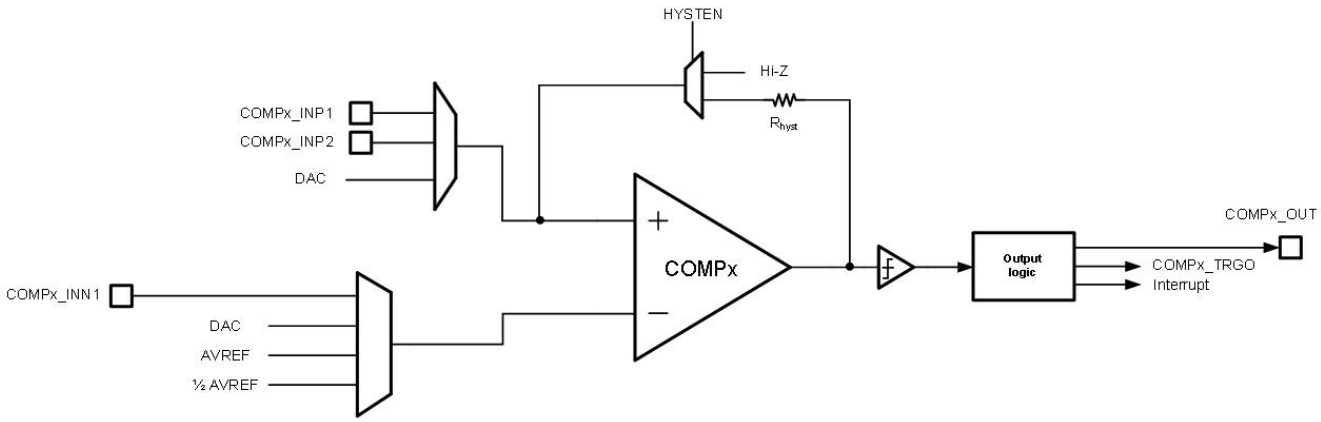


图 14-1 比较器电路框图

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。

比较器支持可配置的回滞功能。正端输入包括引脚、DAC，负端输入包括引脚、DAC、基准电压。

比较器输入电压范围0~VDD，建立时间小于10us，典型传输延迟小于200ns。

14.3 功能描述

14.3.1 基本功能

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

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

比较器输出的逻辑信号可以经过数字滤波和极性控制后输出，或产生中断信号。

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

14.3.2 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。

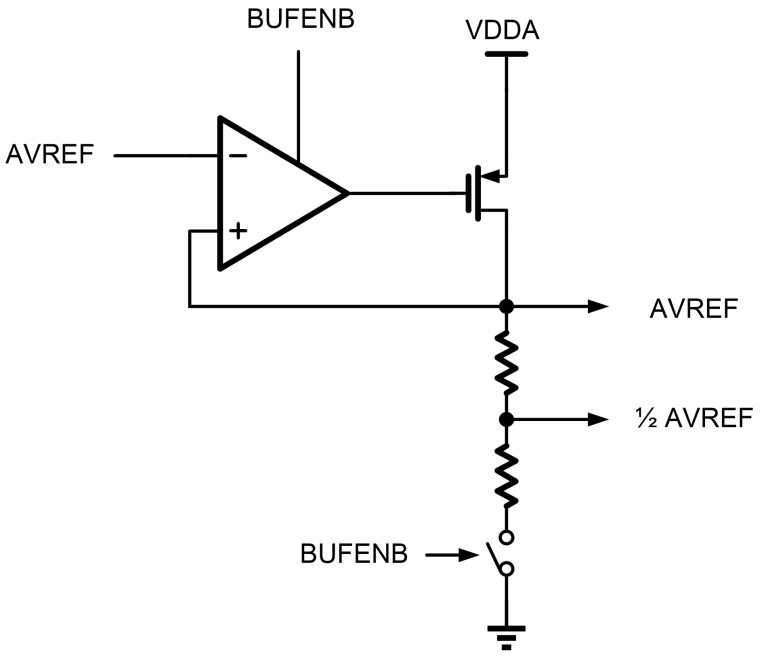


图 14-2 比较器内建基准缓冲器

14.3.3 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。

14.3.4 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。

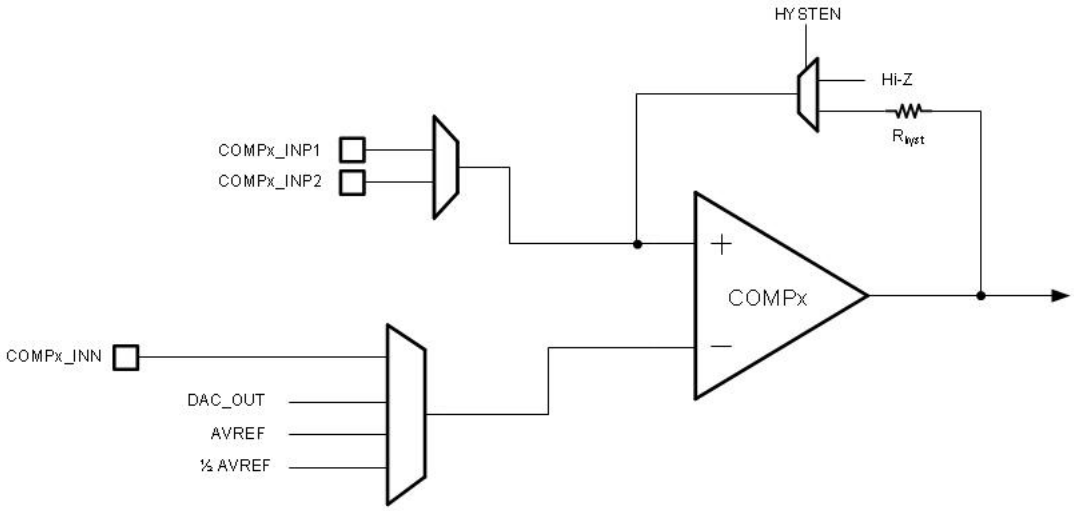


图 14-3 比较器输出滞回

14.3.5 窗口功能

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

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

窗口比较器结构如下图所示：

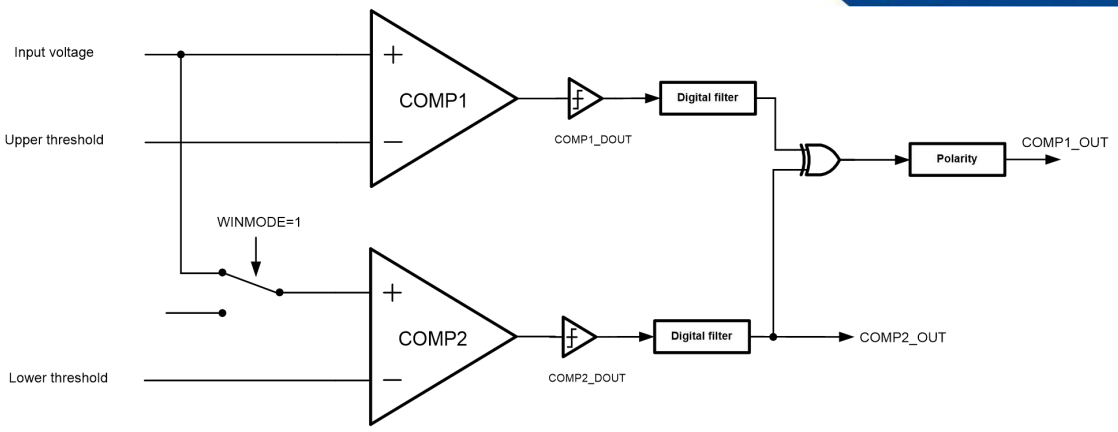


图 14-4 窗口比较器框图

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

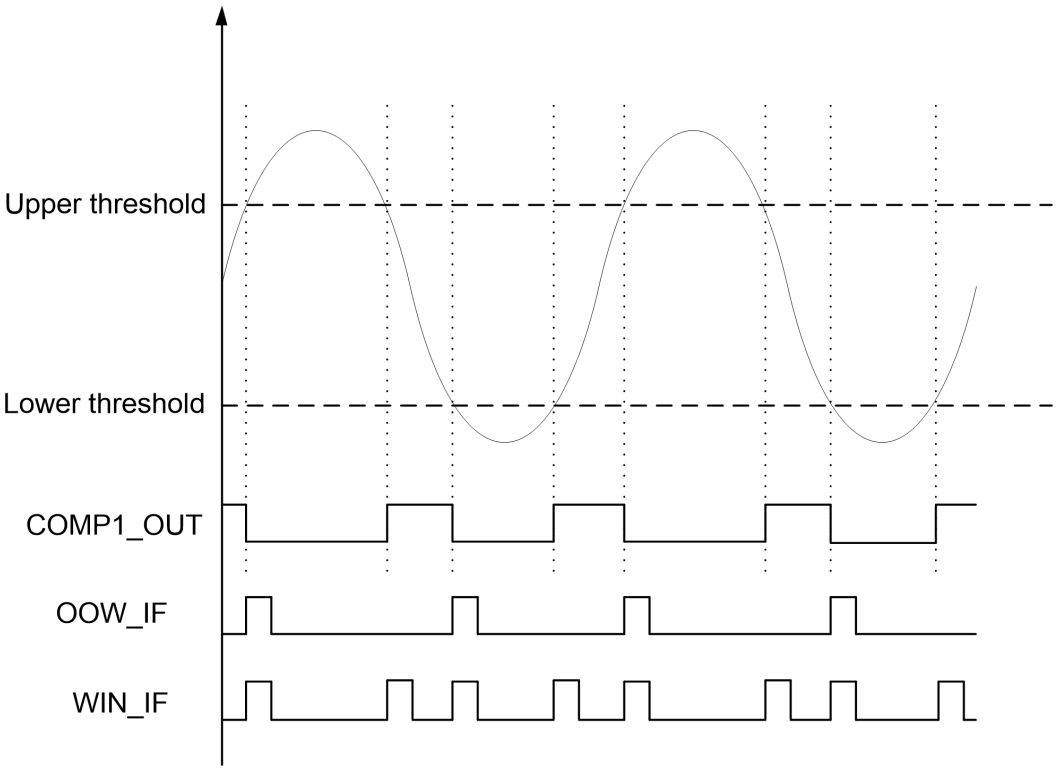


图 14-5 窗口比较器波形示意图

14.3.6 输出数字滤波

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

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

下图是DFLEN=3的示意图：

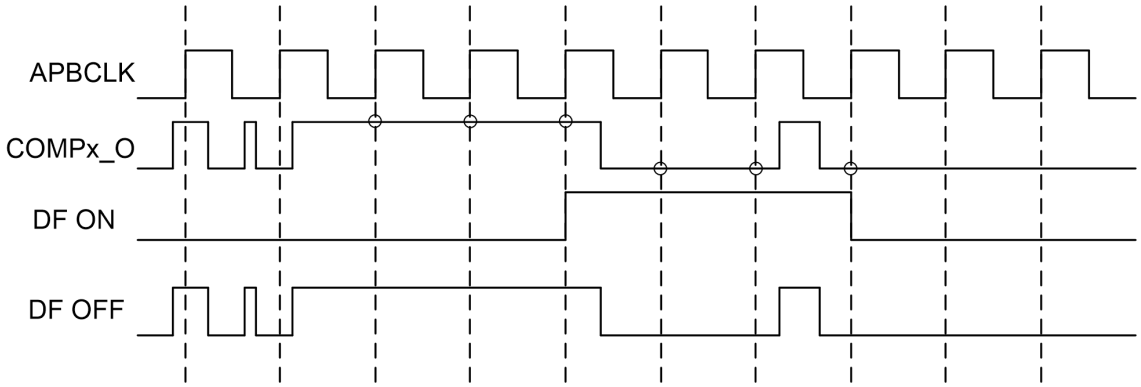


图 14-6 数字滤波（DFLEN=3）波形示意图

14.3.7 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
高速模式	50uA	200ns	5us	所有芯片功耗模式下都可以使用
低功耗模式	1uA	2us	15us	

表 14-1 比较器工作模式

14.3.8 比较器中断

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

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

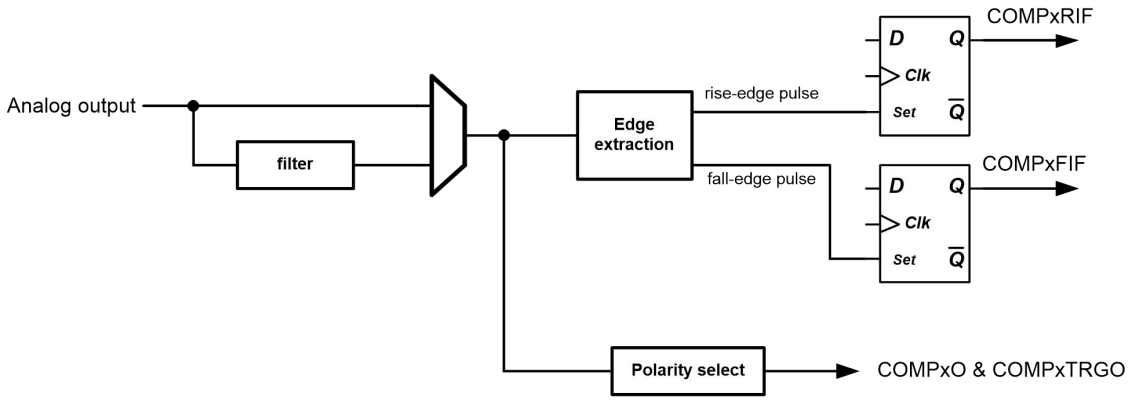


图 14-7 比较器中断产生

14.3.9 比较器输出和触发信号输出

比较器可以直接输出比较结果，还可以给其他外设电路输出触发信号。

输出逻辑示意图如下。

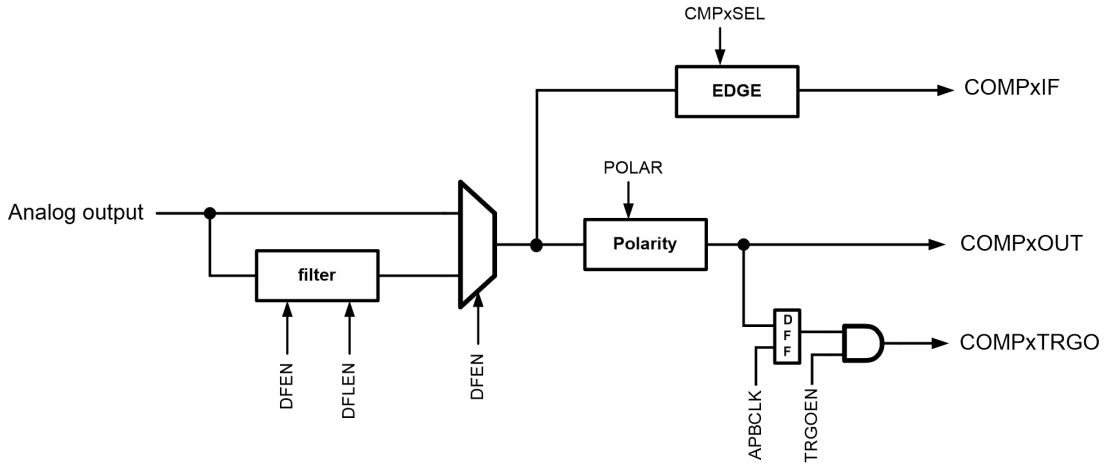


图 14-8 比较器输出逻辑

比较器输出

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

比较器触发信号输出

可以在比较器输出的上升沿、下降沿分别或同时产生触发信号输出。需要输出触发信号时，必须使能COMP总线时钟，当触发事件发生时，触发信号被同步到APBCLK上升沿输出。触发信号可以被连接到定时器的内部触发输入，或者ADC的内部触发输入。

TRGOEN用于使能或禁止触发信号输出。

下图为比较器输出上升沿产生触发信号的例子。

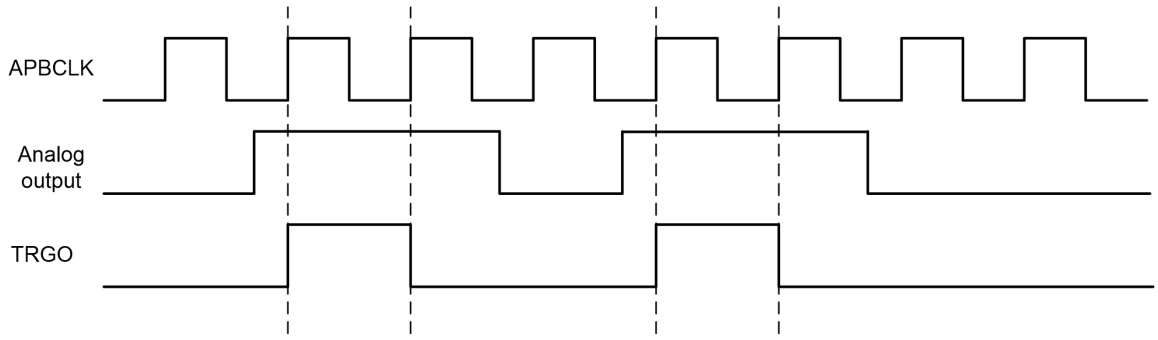


图 14-9 比较器输出上升沿产生触发输出，无滤波

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。

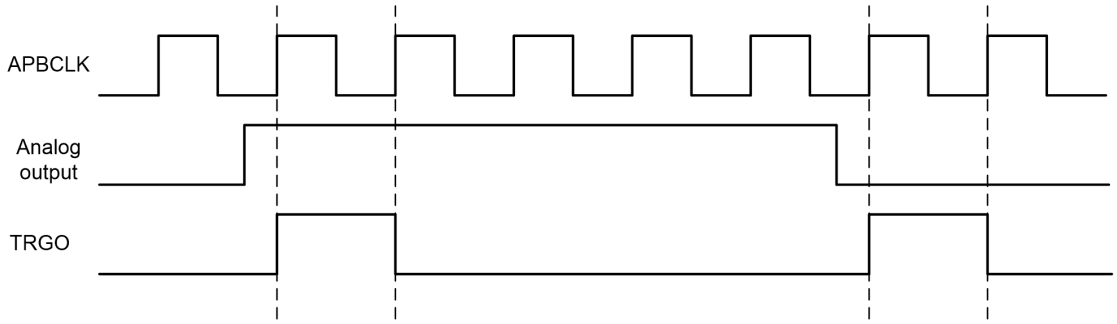


图 14-10 比较器输出上升沿下降沿产生触发输出，无滤波

比较器的触发输出可以被连接到GPTIMx的输入，以便定时器自动记录比较器输出翻转次数。

比较器的触发输出也可以作为启动ADC转换的触发信号，请参考29.7.11转换触发。

14.4 寄存器

模块基地址：0x4001_9C00

offset 地址	名称	符号
0x00	COMP 控制寄存器 1	COMP1_CR1
0x04	COMP 控制寄存器 2	COMP2_CR2
0x08	-	-
0x0C	COMP 中断配置寄存器	COMP_ICR
0x10	COMP 中断标志寄存器	COMP_ISR
0x14	COMP 缓冲器控制寄存器	COMP_BUFCR

14.4.1 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	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					WINMODE	POLAR	DFEN
位权限	R/W-00000					R/W-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-					CMP1O
位权限	R/W-00		U-0					R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HYSTEN	V1PSEL			V1NSEL			CMP1EN
位权限	R/W_0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 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	WINMODE	比较器窗口模式控制寄存器 0：禁止窗口模式 1：使能窗口模式（COMP1 和 COMP2 联动）
17	POLAR	比较器 1 输出极性控制 0：正向输出 1：取反输出
16	DFEN	比较器 1 输出数字滤波使能 0：禁止输出数字滤波 1：使能输出数字滤波
15:14	MODE	比较器 1 工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:9	--	RFU：未实现，读为 0
8	CMP1O	比较器 1 输出，软件只读
7:4	V1PSEL	比较器 1 正极输入选择 000：COMP1_INP1

Bit	助记符	功能描述
		001：COMP1_INP2 010：DAC 其他：RFU
3:1	V1NSEL	比较器 1 负极输入选择 000：COMP1_INN1 001：DAC 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	CMP1EN	比较器 1 使能位 0：关闭比较器 1 1：使能比较器 1

14.4.2 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	-					TRGOSEL		TRGOEN
位权限	U-0					R/W-00		R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					-	POLAR	DFEN
位权限	R/W-00000					U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MODE		-					CMP2O
位权限	R/W-00		U-0					R
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	V2PSEL			V2NSEL			CMP2EN
位权限	U-0	R/W-000			R/W-000			R/W-0

Bit	助记符	功能描述
31:27	--	RFU：未实现，读为 0
26:25	TRGOSEL	比较器触发信号输出选择 00：比较器输出的上升和下降沿都输出触发信号 01：比较器输出的上升沿输出触发信号 10：比较器输出的下降沿输出触发信号 11：比较器输出的上升和下降沿都输出触发信号
24	TRGOEN	比较器触发信号输出使能 0：禁止触发输出 1：允许触发输出
23:19	DFLEN	比较器 2 输出数字滤波长度配置寄存器。滤波长度周期是 DFLEN+1（最小滤波长度为 3） 00000：3 次 APBCLK 采样 00001：3 次 APBCLK 采样

Bit	助记符	功能描述
		00010：3 次 APBCLK 采样 00011：4 次 APBCLK 采样 …… 11111：32 次 APBCLK 采样
18	--	RFU：未实现，读为 0
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 2 工作模式 00/11：低功耗模式 01：RFU 10：高速模式
13:9	--	RFU：未实现，读为 0
8	CMP2O	比较器 2 输出，软件只读
7	--	RFU：未实现，读为 0
6:4	V2PSEL	比较器 2 正极输入选择 000：COMP2_INP1 001：COMP2_INP2 010：DAC 其他：RFU
3:1	V2NSEL	比较器 2 负极输入选择 000：COMP2_INN1 001：DAC 010：AVREF 011：1/2 AVREF 100：RFU 101：RFU 110：RFU 111：RFU
0	CMP2EN	比较器 2 使能位 0：关闭比较器 2 1：使能比较器 2

14.4.3 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	-						OOW_IE	WIN_IE
位权限	U-0						R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		-	-	-		-	-

位权限	U-0		U-0	U-0	U-0		U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CMP2SEL		-	CMP2IE	CMP1SEL		-	CMP1IE
位权限	R/W-00		U-0	R/W-0	R/W-00		U-0	R/W-0

Bit	助记符	功能描述
31:26	--	RFU：未实现，读为 0
25	OOW_IE	Out-Of-Window 中断使能
24	WIN_IE	Window 中断使能
23:16	--	RFU：未实现，读为 0
15:14	--	RFU：未实现，读为 0
13	--	RFU：未实现，读为 0
12	--	RFU：未实现，读为 0
11:10	--	RFU：未实现，读为 0
9	--	RFU：未实现，读为 0
8	--	RFU：未实现，读为 0
7:6	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
5	--	RFU：未实现，读为 0
4	CMP2IE	比较器 2 中断使能，1 有效
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。

14.4.4 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						U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			OOW_IF	WIN_IF	-	CMP2IF	CMP1IF
位权限	U-0			R/W-0	R/W-0	U-0	R/W-0	R/W-0

Bit	助记符	功能描述

Bit	助记符	功能描述
31:10	--	RFU：未实现，读为 0
9	--	RFU：未实现，读为 0
8	--	RFU：未实现，读为 0
7:5	--	RFU：未实现，读为 0
4	OOW_IF	Out-of-Window 中断标志，硬件置位，软件写 1 清零
3	WIN_IF	Window 中断标志，硬件置位，软件写 1 清零
2	--	RFU：未实现，读为 0
1	CMP2IF	比较器 2 中断标志，硬件置位，软件写 1 清零
0	CMP1IF	比较器 1 中断标志，硬件置位，软件写 1 清零

14.4.5 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

15 双线串行总线主机（I²C master）

15.1 概述

I²C 主机模块实现 MCU 与外部 I2C 接口器件之间的同步通信，硬件实现串并转换。支持 I²C 的主机模式，不支持多主机模式。

特点：

- 支持主机模式，不支持多主机模式，不支持从机模式
- 支持 7 位或 10 位从机地址
- 传输速度支持 standard mode(100Kbps), fast mode(400Kbps)

15.2 结构框图

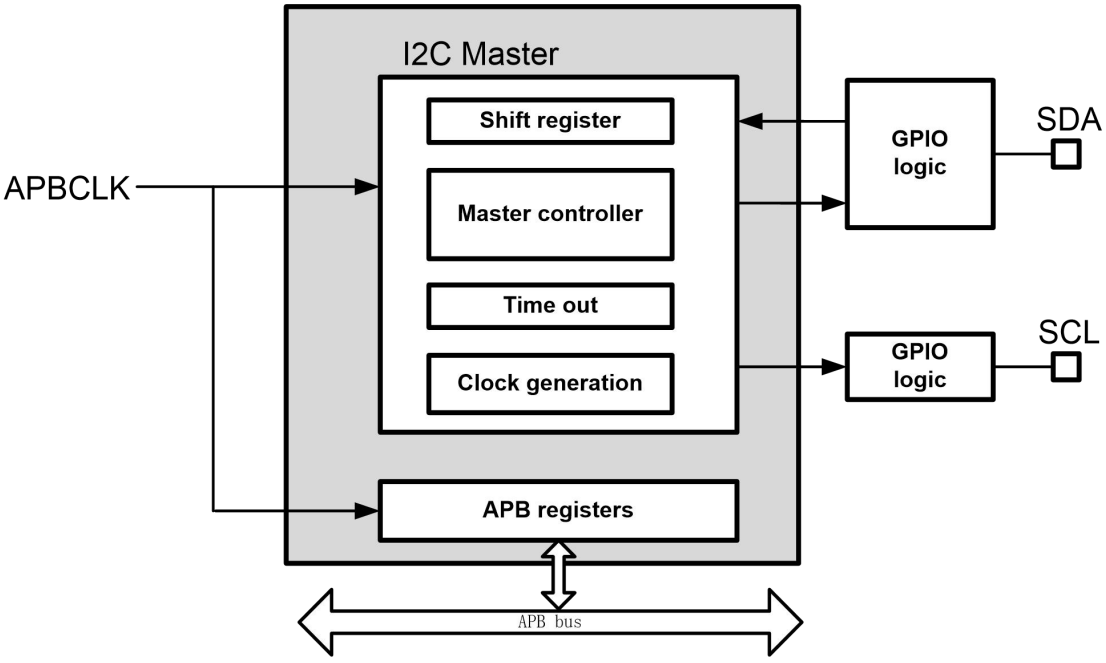


图 15-1 I²C 主机模块框图

15.3 时钟

I²C 主机使用 APBCLK 工作。时钟使能控制在 CMU 模块内完成，进行 I²C 通信前必须正确配置相应的 CMU 控制寄存器。

15.4 接口时序

15.4.1 接口时序图

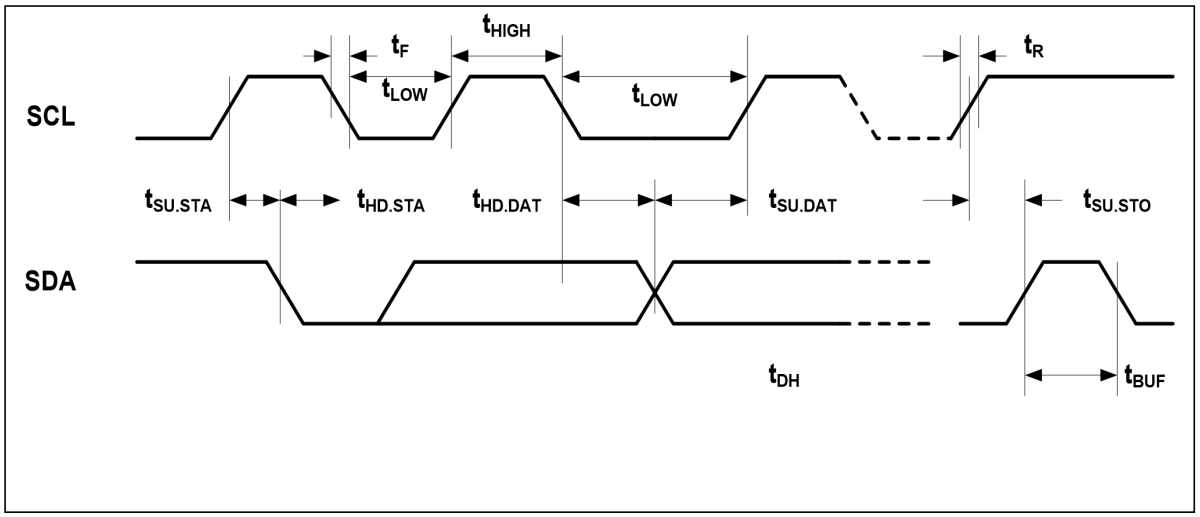


图 15-2I²C 总线时序

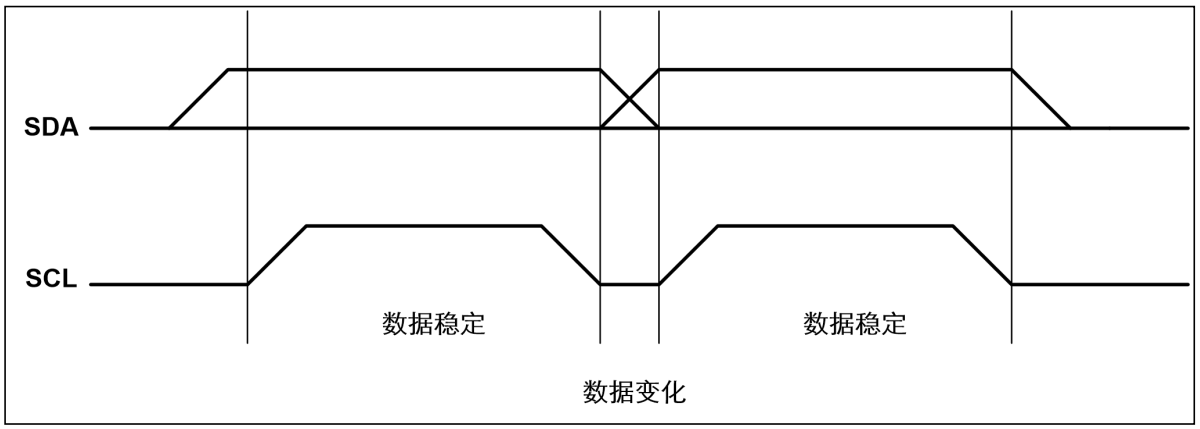


图 15-3 数据有效时序

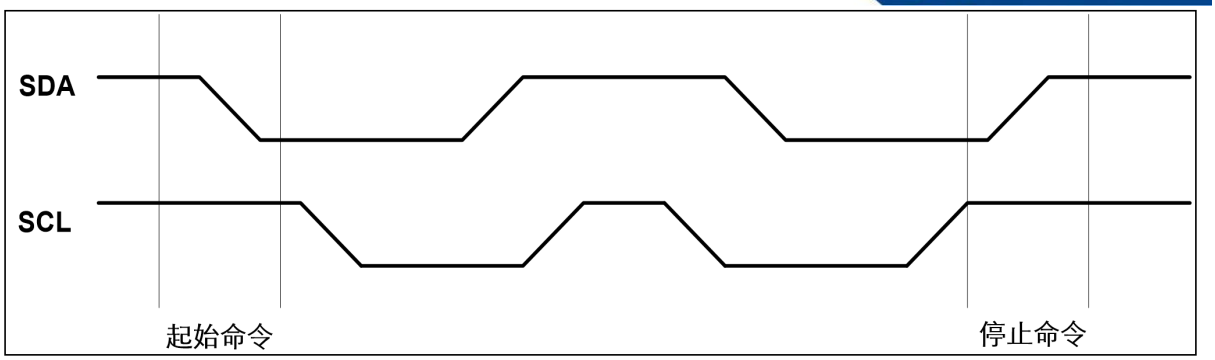


图 15-4 起始（Start）与停止(Stop)命令定义

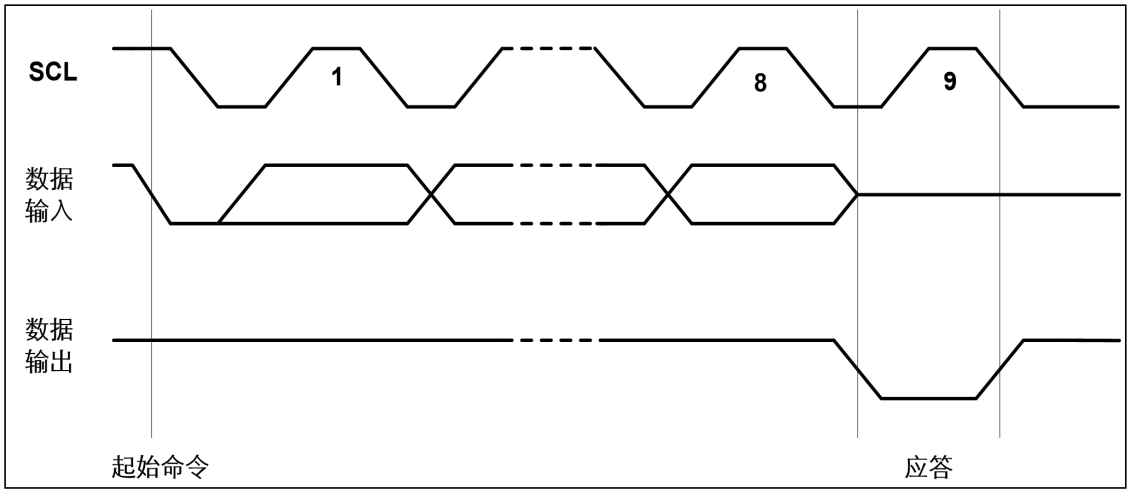


图 15-5 输出应答(ACK)

15.4.2 接口时序描述

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

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

停止命令：当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使 EEPROM 进入等待态低功耗模式。

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

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	

参数	符号	标准模式（100K）		快速模式(400K)		单位
		最小值	最大值	最小值	最大值	
SCL 时钟频率	F_{SCL}	0	100	0	400	kHz
启动条件建立时间	$T_{SU:STA}$	4.7	—	0.6	—	us
启动条件保持时间	$T_{HD:STA}$	4.0	—	0.6	—	us
时钟低电平时间	T_{LOW}	4.7	—	1.3	—	us
时钟高电平时间	T_{HIGH}	4.0	—	0.6	—	us
数据输入建立时间	$T_{SU:DAT}$	250	—	100(4)	—	ns
数据输入保持时间	$T_{HD:DAT}$	5.0 0(2)	— 3.45(3)	— 0(2)	— 0.9(3)	us us
SDA 和 SCL 上升时间	T_R	—	1000	20+0.1Cb(5)	300	ns
SDA 和 SCL 下降时间	T_F	—	300	20+0.1Cb(5)	300	ns
停止条件建立时间	$T_{SU:STO}$	4.0	—	0.6	—	us
总线空闲时间	T_{BUF}	4.7	—	1.3	—	us
总线的容性负载	Cb	—	400	—	400	Pf
噪声容限低值	V_{nL}	0.1V_{DD}	—	0.1V_{DD}	—	V
噪声容限高值	V_{nH}	0.2V_{DD}	—	0.2V_{DD}	—	V

表 15-1 I²C 接口时序要求

15.5 I²C 工作模式

I²C模块支持以下工作模式：

- 主机接收
- 主机发送

芯片上电后I²C模块默认关闭，主机和从机都不工作。软件需要根据应用选择模块工作模式，通过设置MSPEN来使能主机通信。

15.6 I²C 地址格式

I²C总线协议定义了以下保留地址，对其中多数保留地址，I²C硬件不做合法性判断，由软件保证地址的合法性。

从机地址	R/W_ bit	描述
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

表 15-2 I²C 保留地址定义

15.7 I²C 初始化

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

- 清零RCC模块的I2CRST寄存器，确保I²C模块不处于复位状态
- 置位RCC模块的I2C_APBEN寄存器，使能I²C模块寄存器总线接口时钟

15.7.1 主机波特率配置

I²C 主机需要在使能前配置通信波特率，而从机不需要配置。

MSPBRG[8:0]波特率配置寄存器用于产生通信波特率。MSPBRG 是 9 bit 波特率分频系数，波特率计算公式如下：

$T_{SCL}=2T_{BRG}$

$T_{BRG}=2\times T_{I2CCLK}\times(MSPBRG[8:0]+1)$；$T_{I2CCLK}$ 为 I²C 工作时钟周期，即：

$MSPBRG = F_{I2CCLK}/(4*F_{SCL}) - 1$

例如对于 100k 波特率，若 I²C 工作时钟为 8M，则 MSPBRG=19。

15.8 I²C 主机功能

FM33FH0的I²C主机模式不支持多主机总线，因此挂在总线上的其他设备都是从机。总线上总是由主机提供同步时钟SCL，SDA数据流方向可以是主机发送从机接收，或者从机发送主机接收。

I²C总线通信总是由主机发起，主机模式支持7bit或10bit寻址。

15.8.1 7bit 寻址

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

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

位描述：

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

主机向从机写入数据

典型的7bit寻址，主机向从机写入数据的帧结构如下图所示。

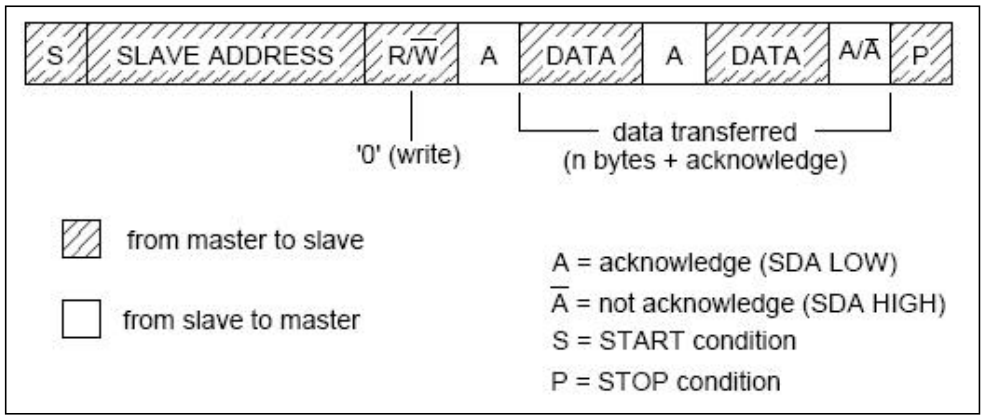


图 15-6 主机向 7 位地址从机写入数据时的帧格式

1、主机发起 START 时序

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

4、主机发送第一帧 8 位数据

5、主机在每次发送完 8 位数据后，会在第 9 个 SCL 判断是否检测到有效的 ACK，如果主机检测到 ACK 成功后，会继续输出下一字节数据

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

7、主机完成所有数据发送后，发送 STOP 时序

I2C主机对7位地址从机写入数据的波形示意图如下：

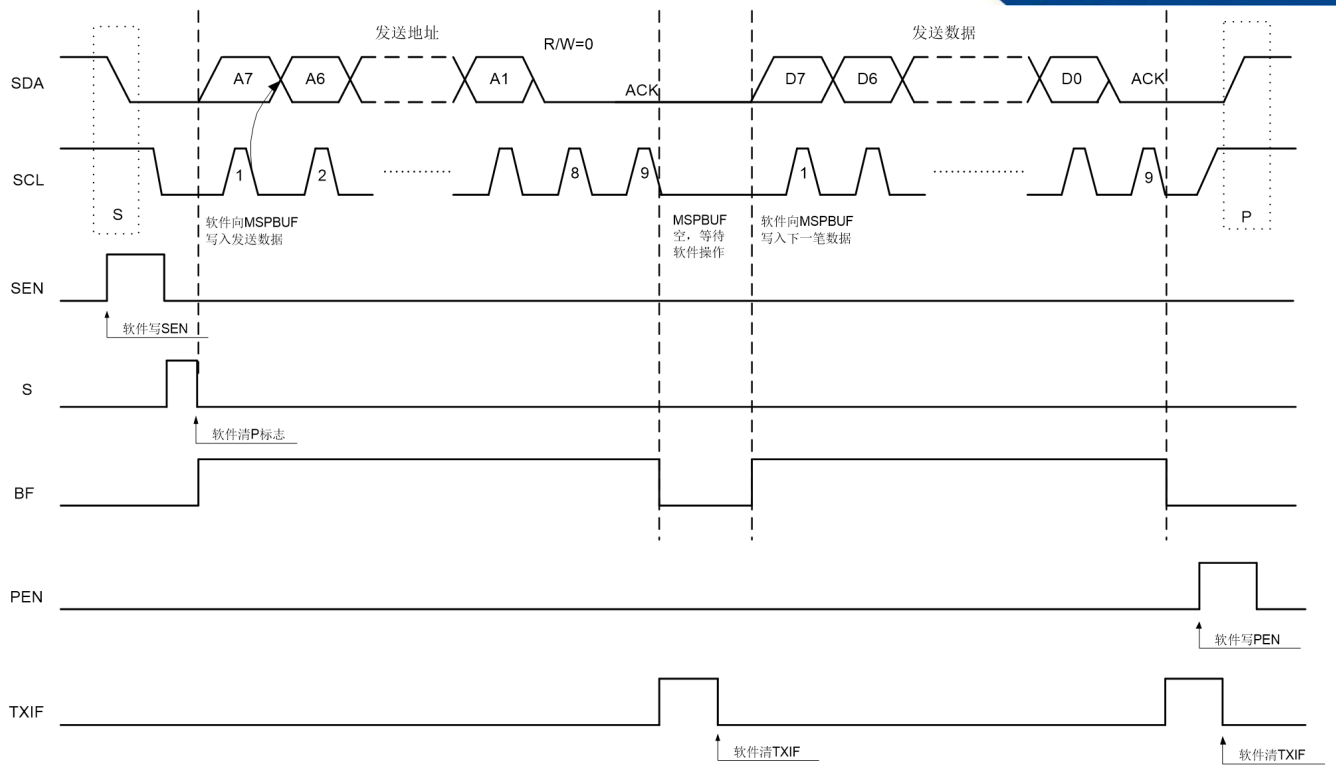


图 15-7 I²C 主机对 7 位地址从机发送数据流图

主机从从机读取数据

典型的7bit寻址，主机从从机读取数据的帧格式如下图所示。

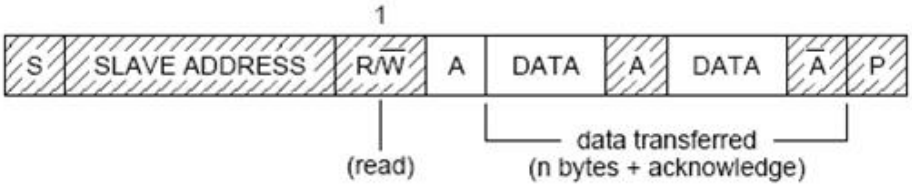


图 15-8 主机从 7 位地址从机读取数据时的帧格式

1、 主机发起 START 时序

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

3、 此时设置 MSPCON.RCEN 为 1，主机自动转为接收状态

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

5、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

6、 主机发送 STOP 时序终止读取

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：

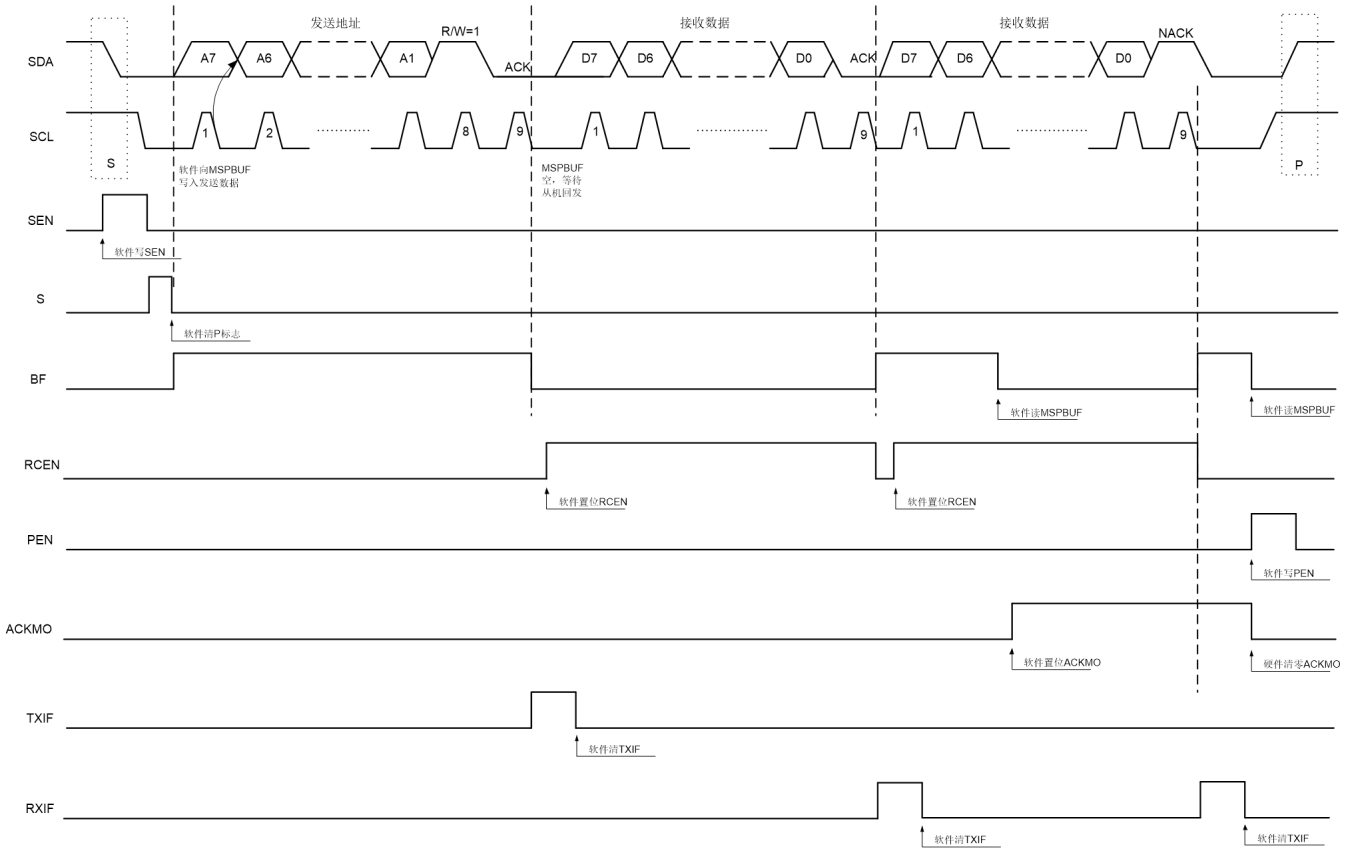


图 15-9 I²C 从 7 位地址从机读取数据流图

双向数据传输（组合模式）

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

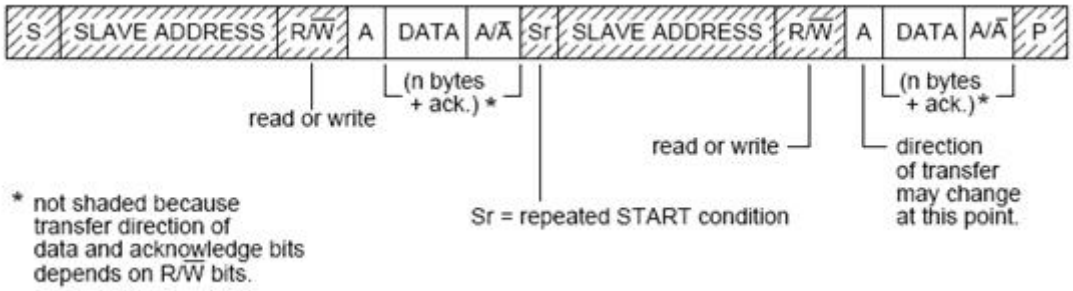


图 15-10 双向数据通信帧格式

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。

15.8.2 10bit 寻址

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

主机向从机写入数据

典型的10bit寻址，主机向从机写入数据的数据流图如下图所示。

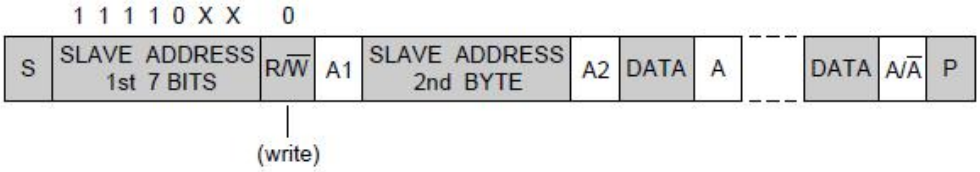


图 15-11 10bit 寻址，主机向从机写入数据

1、主机发起 START 时序

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

3、主机检查从机回发的 ACK

4、主机发送第二个从机地址字节，包含从机地址的低 8 位

5、主机检查从机回发的 ACK

6、主机继续向从机写入数据

7、主机完成所有数据发送后，发送 STOP 时序

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。

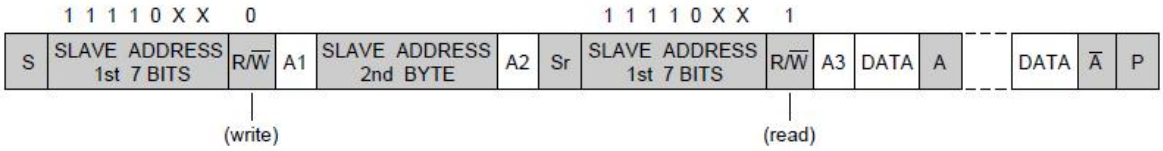


图 15-12 10bit 寻址，主机从从机读取数据

1、 主机发起 START 时序

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

3、 主机发送第二字节从机地址，包含低 8 位地址

4、 主机发送 ReSTART 时序

5、 主机再次发送第一字节从机地址，将 R/W 为改为 0

6、 此时设置 MSPCON.RCEN 为 1，主机转为接收状态

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

8、 主机读取最后一个字节后，在第 9 个 SCL 向从机发送 NACK

9、 主机发送 STOP 时序终止读取

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。

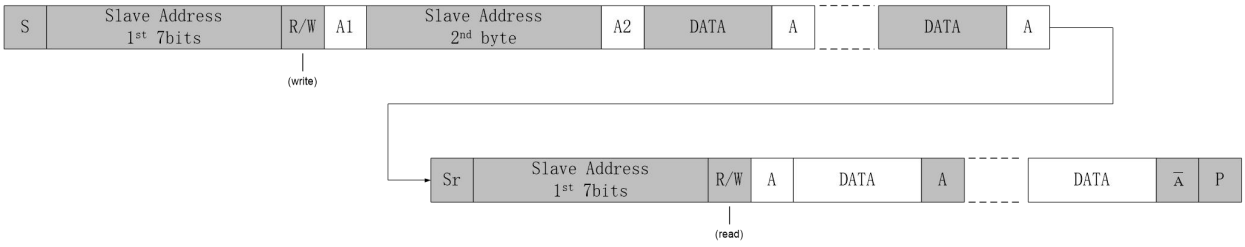


图 15-13 I²C 软件发送数据流图

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。

15.8.3 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，MSPEN 置位，如果数据缓存寄存器 MSPBUF 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 MSPBUF，同时 I2C 模块自动置位 SEN 产生 START 时序，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，此时可以又软件置位 PEN 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 PEN 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：

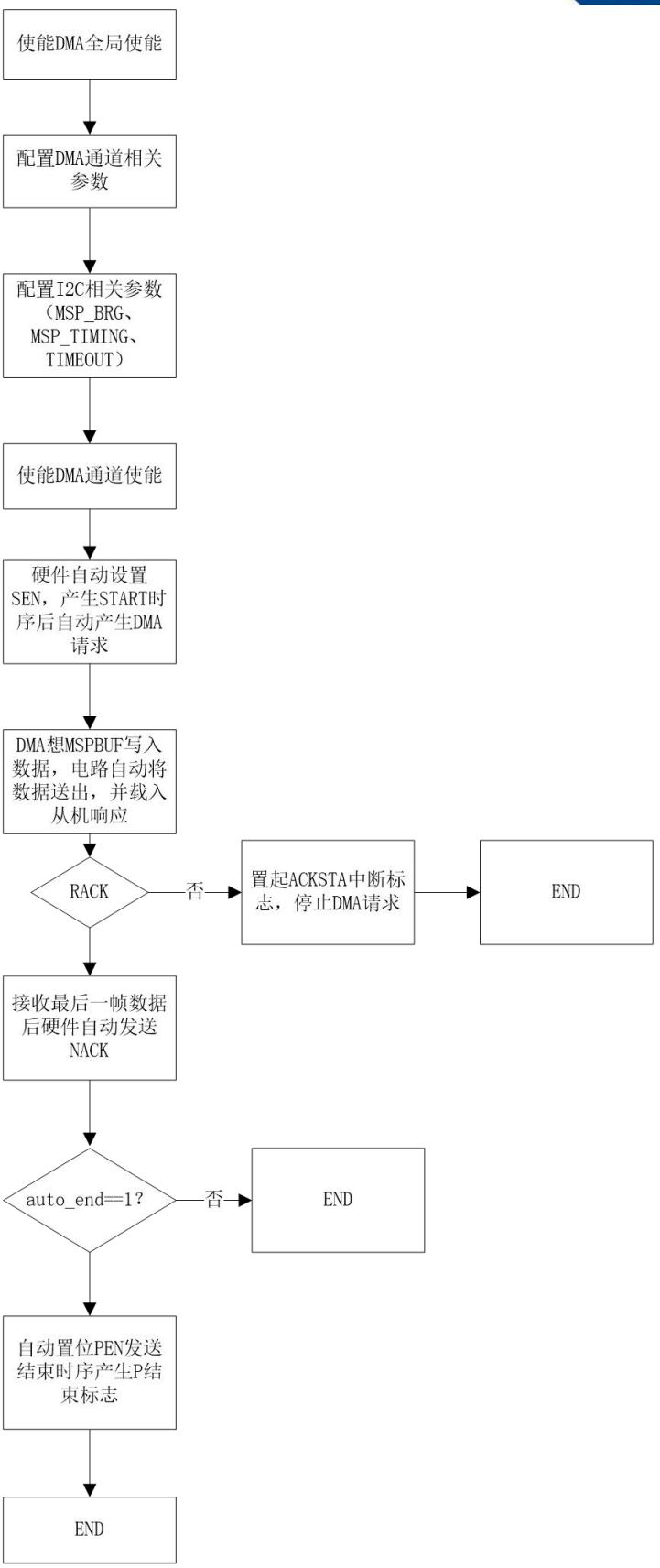


图 15-14 I²C 主机 DMA 发送流程图

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 MSP_DMAEN=1，然后使能对应的 DMA 通道，I2C 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 MSPBUF 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 I2C 回发 NACK。随后根据 AUTOEND 寄存器配置，可以由软件或硬件置位 PEN 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：

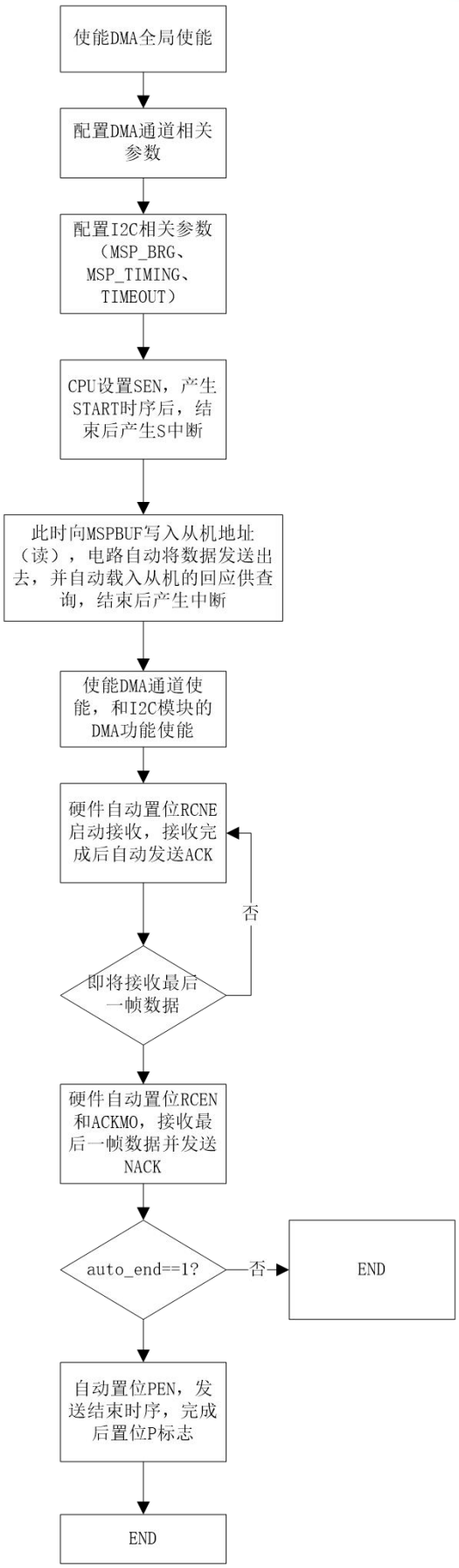


图 15-15 I²C 主机 DMA 接收流程图

15.8.4 SCL 延展（Slave Clock Stretching）

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

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。

15.8.5 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。

15.8.6 可编程时序

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

通过 MSPBRG 寄存器可以设置 SCL 的低电平和高电平宽度，通过 SDAHD 寄存器可以配置 SDA 数据相对 SCL 时钟脉冲的保持和建立保持时间长度。此功能对于发送数据和发送 ACK 都有效。

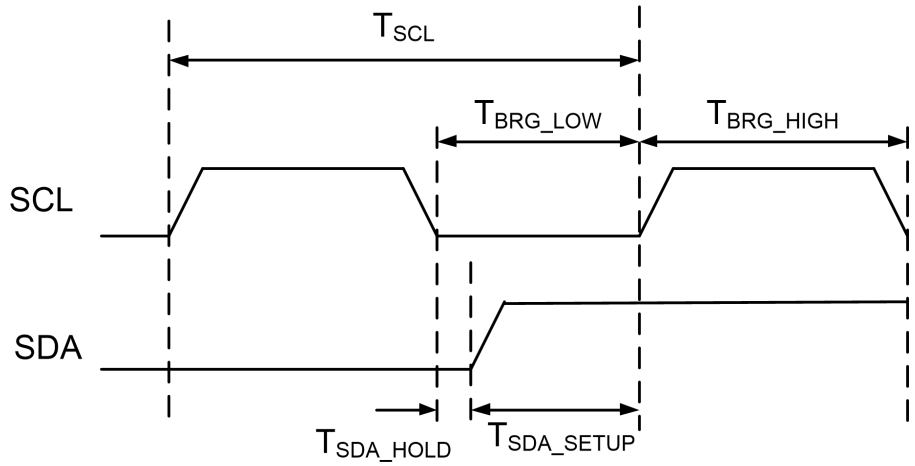


图 15-16 主机时序控制

上图中，T_{SCL}为通信波特率，各个参数可以由以下公式表达：

$T_{SCL} = T_{BRG_LOW} + T_{BRG_HIGH}$

$T_{SDA_SETUP} = T_{BRG_LOW} - T_{SDA_HOLD}$

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

MSPBRGH >= 2

MSPBRGL >= 2

MSPBRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

15.9 寄存器

I2C0模块基地址：0x40015000

I2C1模块基地址：0x40015400

offset 地址	名称	符号
0x00	I2C 主机配置寄存器（I2C Master Config Register）	I2C_MSPCFGR
0x04	I2C 主机控制寄存器（I2C Master Control Register）	I2C_MSPCR
0x08	I2C 主机中断使能寄存器（ I2C Master Intterupt Enable Register）	I2C_MSPIER
0x0C	I2C 主机中断标志寄存器（ I2C Master Interrupt Status Register）	I2C_MSPISR
0x10	I2C 主机状态寄存器	I2C_MSPSR

	（I2C Master Status Register）	
0x14	I2C 主机波特率寄存器 （I2C Master Baud rate Generator Register）	I2C_MSPBGR
0x18	I2C 主机收发缓存寄存器 （I2C Master transfer Buffer）	I2C_MSPBUF
0x1C	I2C 主机时序控制寄存器 （I2C Master Timing Control Register）	I2C_MSPTCR
0x20	I2C 主机超时寄存器 （I2C Master Time-Out Register）	I2C_MSPTOR

15.9.1 I2C 主机配置寄存器（I2C_MSPCFGR）

名称	I2C_MSPCFGR							
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	MSP_DMAEN
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						TOEN	MSPEN
位权限	U-0						R/W-0	R/W-0

位号	位名	说明
31:15	--	RFU：未实现，读为 0
17	AUTOEND	主机 DMA 自动终止 (DMA automatic transfer end) 1：DMA 指定长度传输完成后，自动发送 STOP 时序 0：DMA 指定长度传输完成后，等待软件接管
16	MSP_DMAEN	主机 DMA 使能 (Master DMA enable) 0：关闭 DMA 功能 1：使能 DMA 功能
15:2	-	RFU：未实现，读为 0
1	TOEN	SCL 拉低超时使能 (Time Out enable) 1：使能超时功能，超时周期由 MSPTO 寄存器定义 0：关闭超时功能
0	MSPEN	I2C 主机模块使能控制位 (Master enable) 1：I2C 主机使能 0：I2C 主机禁止

15.9.2 I2C 主机控制寄存器（I2C_MSPCR）

名称	I2C_MSPCR

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	-				RCEN	PEN	RSEN	SEN
位权限	U-0				R/W-0	R/W/Dy-0	R/W/Dy-0	R/W/Dy-0

位号	位名	说明
31:4	--	RFU：未实现，读为 0
3	RCEN	主机接收模式下，接收使能位 (Receive enable) 1：主机接收使能 0：接收禁止 主机通信中，软件在发送完地址字节后，通过置位 RCEN 将传输方向切换为主机接收，然后可以接收来自于从机的数据。RCNE 在接收过程中保持为 1，直到软件置位 PEN 发送 STOP 时序。
2	PEN	STOP 时序产生使能控制位，软件写 1 发送 STOP 时序，发送完成后硬件自动清零 (Stop Enable)
1	RSEN	Repeated START 时序产生使能控制位，软件写 1 发送 Repeated START 时序，发送完成后硬件自动清零 (Repeated Start Enable)
0	SEN	START 时序产生使能控制位，软件写 1 发送 START 时序，发送完成后硬件自动清零 (Start Enable)

15.9.3 I2C 主机中断使能寄存器（I2C_MSPIER）

名称	I2C_MSPIER							
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	-	WCOLE	OVTE	SE	PE	NACKE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOLE	WCOL 中断使能寄存器 (Write collision interrupt enable) 1：允许写冲突中断 0：禁止写冲突中断
5	OVTE	SCL 超时中断使能寄存器 (SCL overtime enable) 1：允许超时中断 0：禁止超时中断
4	SE	START 时序中断使能寄存器 (Start interrupt enable) 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	STOP 时序中断使能寄存器 (Stop interrupt enable) 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	主机发送模式下 NACK 中断使能寄存器 (Non-ACK interrupt enable) 1：允许收到 NACK 产生中断 0：禁止产生 NACK 中断
1	TXIE	I2C 主机发送完成中断使能(Trasnmit done interrupt enable) 1：允许发送完成中断 0：禁止发送完成中断
0	RXIE	I2C 主机接收完成中断使能(Receive done interrupt enable) 1：允许接收完成中断 0：禁止接收完成中断

15.9.4 I2C 主机中断标志寄存器（I2C_MSPISR）

名称	I2C_MSPISR							
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	-	WCOL	OVT	S	P	ACKSTA	TXIF	RXIF
位权限	U-0	R/W1C-0		R-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0

位号	位名	说明
31:7	--	RFU：未实现，读为 0
6	WCOL	写冲突检测位，MCU 只能在完成 START 时序或发送完成一帧读写之后才能写 MSPBUF，否则发生写冲突；硬件置位，软件写 1 清零(Write Collision Interrupt Flag,write 1 to clear) 1：发送写冲突 0：未发生冲突

位号	位名	说明
5	OVT	SCL 超时中断标志，仅在 TOEN 为 1 时工作(SCL OverTime Interrupt Flag) 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	START 时序发送完成中断标志，硬件置位，软件读取后清零(Start Interrupt flag)
3	P	STOP 时序发送完成中断标志，硬件置位，软件读取后清零(Stop interrupt flag)
2	ACKSTA	主控发送模式下，来自从机的回应信号；当主机发送后收到 NACK，此标志可以产生中断；硬件置位，软件写 1 清零。(Acknowledge Status Flag ,write 1 to clear) 1：从机回应 NACK 0：从机回应 ACK
1	TXIF	I2C 主机发送完成中断标志，硬件置位，软件写 1 清零(Trasnmit done interrupt flag,write 1 to clear) 此标志寄存器在主机接收完从机回发的 ACK 或 NACK 后置位。
0	RXIF	I2C 主机接收完成中断标志，硬件置位，软件写 1 清零(Receive done interrupt flag,write 1 to clear) 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

15.9.5 I2C 主机状态寄存器（I2C_MSPSR）

名称	I2C_MSPSR							
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	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BUSY	RW_	-	BF	-	ACKMO
位权限	U-0		R-0	R-0	U-0	R-0	U-0	R/W-0

位号	位名	说明
31:6	--	RFU：未实现，读为 0
5	BUSY	I2C 通信状态位 (Busy) 1：接口处于读写状态，正在进行数据传输， 0：已完成数据传输
4	RW	I2C 传输方向状态位 (Read/Write) 1：主机从从机读取数据 0：主机向从机写入数据
3	-	RFU：未实现，读为 0
2	BF	缓冲器满状态位 (Buffer Full) 接收：

位号	位名	说明
		1：接收完成，MSPBUF 满 0：接收未完成，MSPBUF 空 发送： 1：正在发送，MSPBUF 满 0：发送完成，MSPBUF 空
1	-	RFU：未实现，读为 0
0	ACKMO	主控接收模式下，主机回应信号的状态 (Ack Master output) 1：主机回发 NACK 0：主机回发 ACK ***注意：必须在 P 标志寄存器被清零的情况下，软件才能置位 ACKMO***

15.9.6 I2C 主机波特率设置寄存器（I2C_MSPBGR）

名称	I2C_MSPBGR							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							MSPBRGH[8]
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSPBRGH[7:0]							
位权限	R/W-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							MSPBRGL[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBRGL[7:0]							
位权限	R/W-00010011							

位号	位名	说明
31:25	--	RFU：未实现，读为 0
24:16	MSPBRGH	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 (Master SCL High level length)
15:9	-	RFU：未实现，读为 0
8:0	MSPBRGL	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 (Master SCL Low level length)

15.9.7 I2C 主机收发缓冲寄存器（I2C_MSPBUF）

名称	I2C_MSPBUF							
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	SSPBUF							
位权限	R/W-00000000							

位号	位名	说明
31:8	--	RFU：未实现，读为 0
7:0	MSPBUF	MSPBUF[7:0]：数据的读写通过对 MSPBUF 的操作完成。发送时，对 MSPBUF 执行写操作，同时也载入数据收发移位寄存器(MSPSR)；接收时，MSPBUF 与 MSPSR 组成双缓冲结构，读出数据为 MSPBUF 的数据。接收完一个字节的数据，MSPSR 将数据载入 MSPBUF，同时置位 I2CIF。MSPSR 不是直接寄存器，没有物理地址。 (Master data Buffer)

15.9.8 I2C 主机时序控制寄存器（I2C_MSPTCR）

名称	I2C_MSPTCR							
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	-							SDAHD[8]
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SDAHD[7:0]							
位权限	R/W-00001010							

位号	位名	说明
31:9	--	RFU：未实现，读为 0
8:0	SDAHD	定义 SDA 相对于 SCL 下降沿的保持时间参数，以 I2C 工作时钟计数(SDA hold delay) 注意：最小有效值为 1，最大有效值为 MSPBRGL

15.9.9 I2C 主机超时寄存器（I2C_MSPTOR）

名称	I2C_MSPTOR							
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					TIMEOUT[11:8]			
位权限					R/W-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUT[7:0]							
位权限	R/W-1111 1111							

位号	位名	说明
31:12	--	RFU：未实现，读为 0
11:0	TIMEOUT	定义从机 SCL 低电平延展超时周期，软件可以在 MSPEN=0 的情况下改写 (SCL stretching Time Out) $T_{SCL_STRETCHING_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

16 通用异步收发传输器（UART）

16.1 概述

UART串行通信模块特点如下

- 波特率软件可配置
- 支持全双工通信口
- 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
- 支持发送延迟功能

UART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

16.2 结构框图

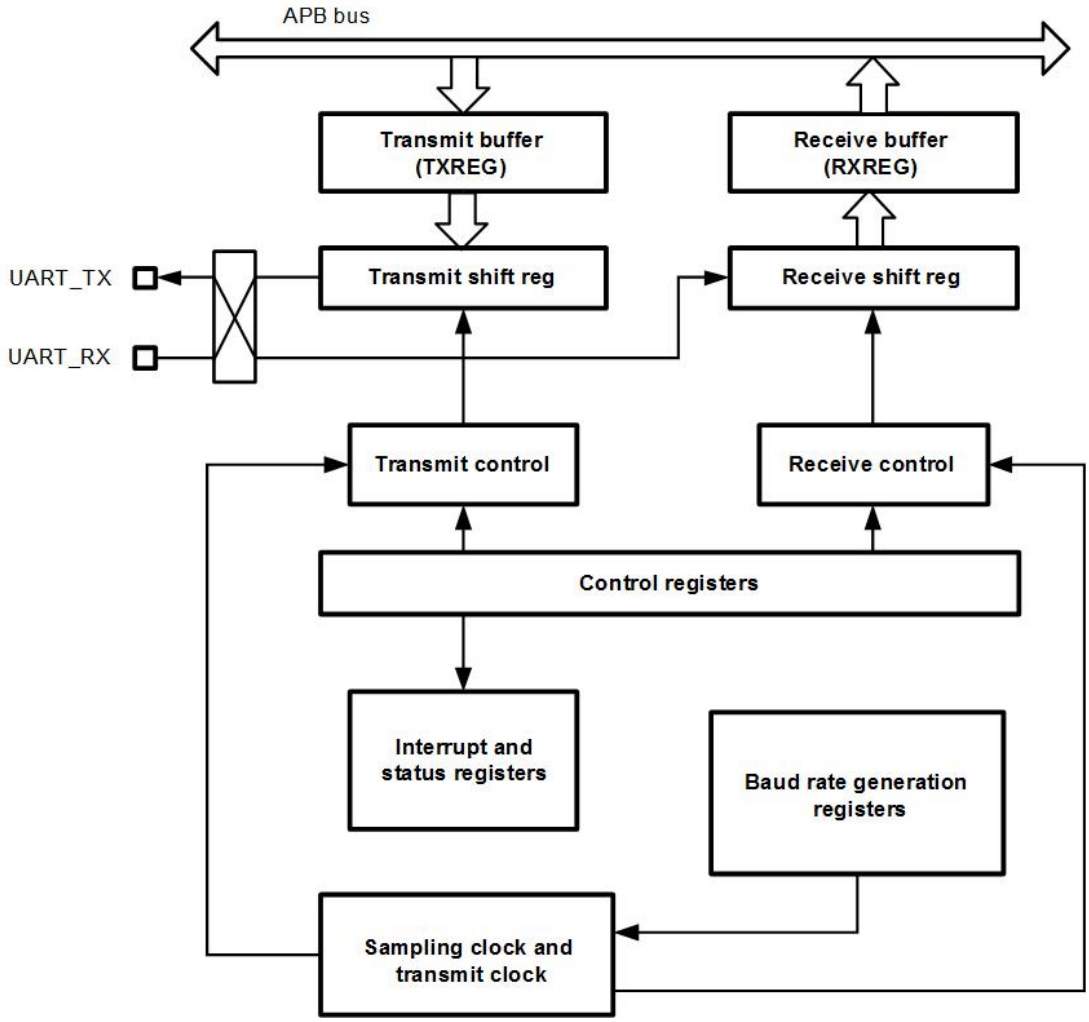


图 16-1 UART 接口时序

16.3 UART 类型区分

芯片集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART (UART0/1/2/3/4/5)	USART (UART6/7)	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
半双工单线通信	Y	Y	-
休眠唤醒	Y	Y	Y
接收超时	Y	Y	-

发送延迟	Y	Y	-
硬件流控	Y		
数据长度	6、7、8、9bits		
LIN support	Y	N	

表 16-1 UART 类型列表

16.4 UART 字符描述

16.4.1 UART 模式

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

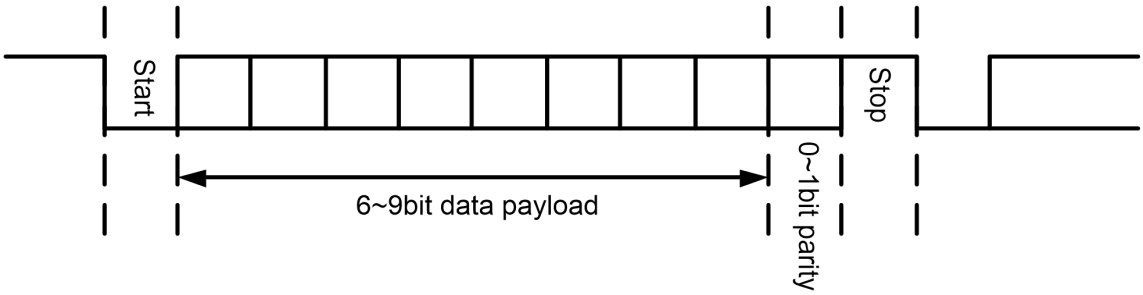


图 16-2 UART 字符描述

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 16-2 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】。

16.5 功能描述

16.5.1 时钟结构

UART 采用了多时钟结构：

- 总线寄存器时钟用 PCLK 表示，来源于 APBCLK。当 CPU 或者 DMA 需要访问 UART 内部寄存器时，必须使能 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，能够独立于 APBCLK 工作。必须使能 UCLK 才能进行数据收发。
- LINCLK 用于实现数据收发以外的 LIN 功能，比如唤醒检测/发送、间隔域检测/发送。LINCLK 使用 LSCLK 工作。

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 UART 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。

16.5.2 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。

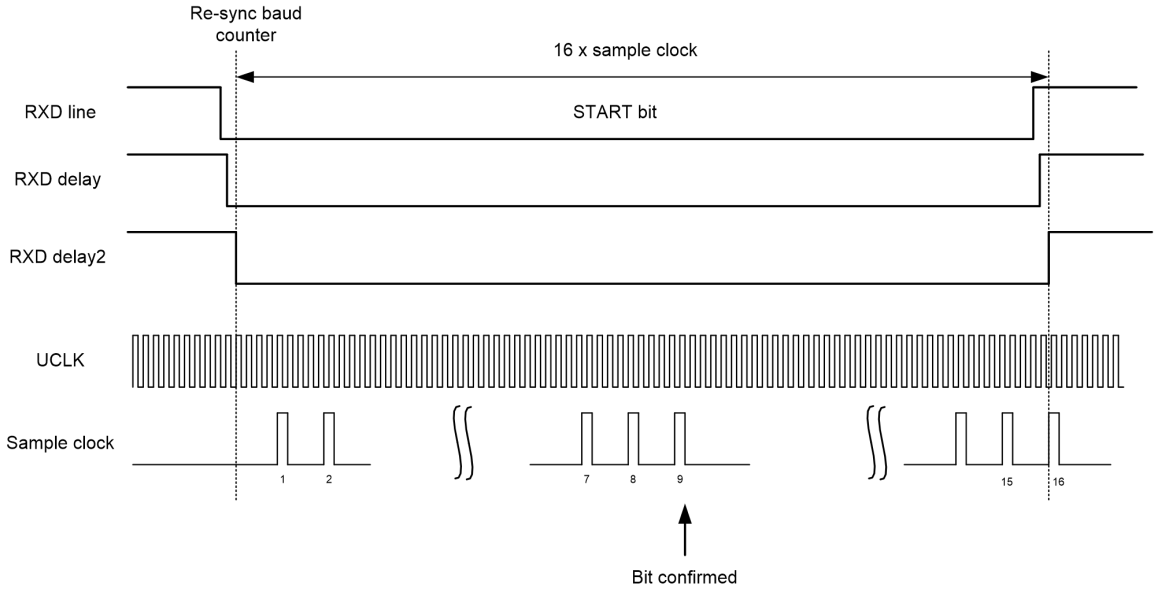


图 16-3 起始位同步

16.5.3 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。

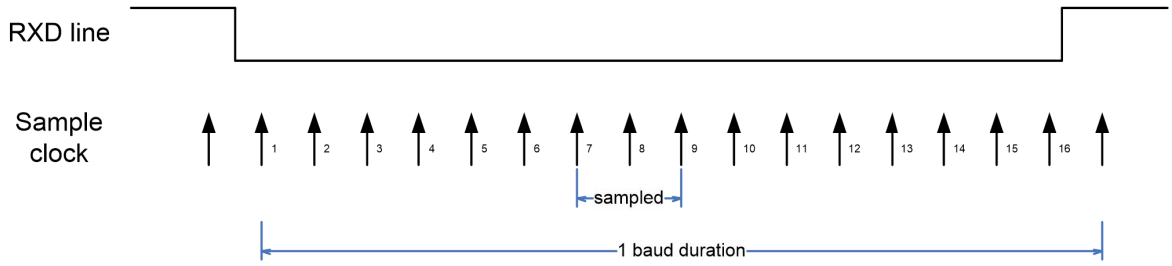


图 16-4 位接收 16 倍采样

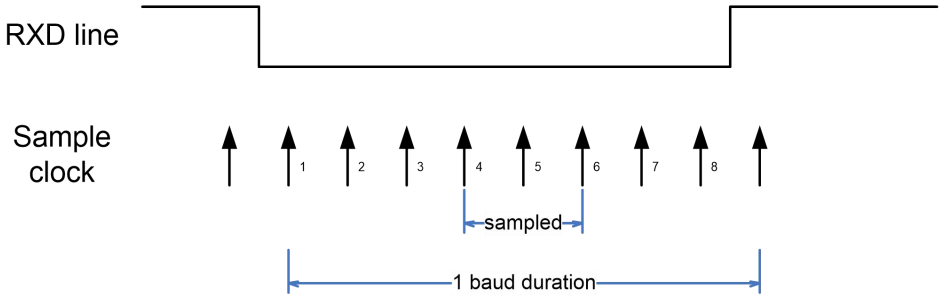


图 16-5 位接收 8 倍采样

接收移位寄存器收到的 bit 位是多数判决的结果。例如三次采样结果是 001，则判决为 0；如果是 011，

则判决为 1。

通过 OVSM 寄存器可以配置 UART 接收数据时的过采样倍数。如果 UART 对输入信号进行 16 倍过采样，要求 SPBRG 配置不能小于 16，即 UART 工作时钟必须至少是波特率的 16 倍。如果 UART 对输入信号进行 8 倍过采样，要求 SPBRG 配置不能小于 8，即 UART 工作时钟必须至少是波特率的 8 倍。

选择较小的过采样倍数时，可以获得较高的通信波特率。

16.5.4 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。

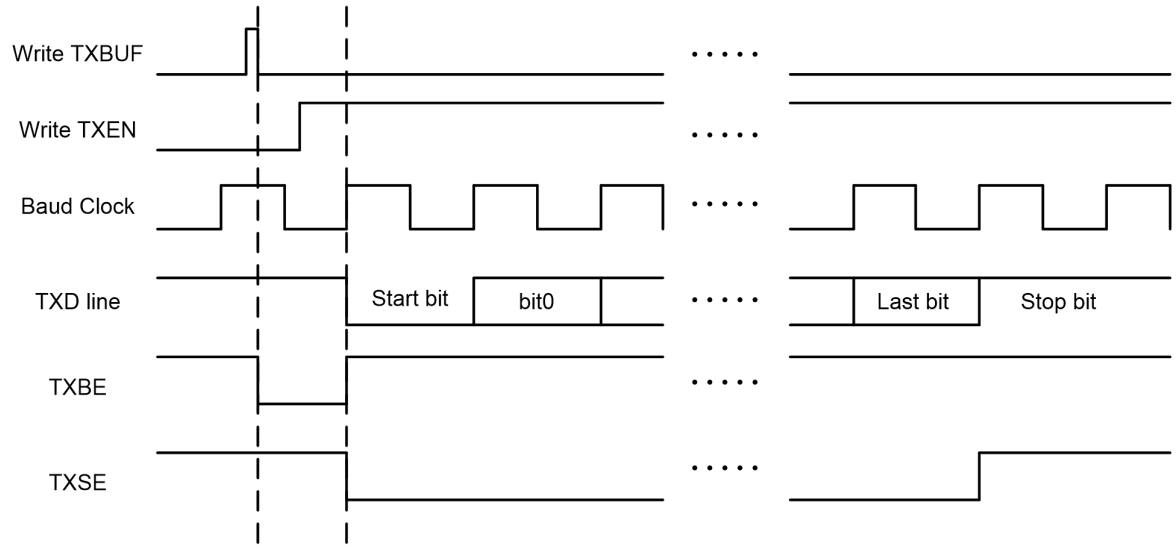


图 16-错误!未定义书签。 UART 异步发送波形 1

上图中推荐的操作步骤如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 TXSE_IE 或者 TXBE_IE
- 决定数据发送的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 如果希望发送的串行数据红外调制，向 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。

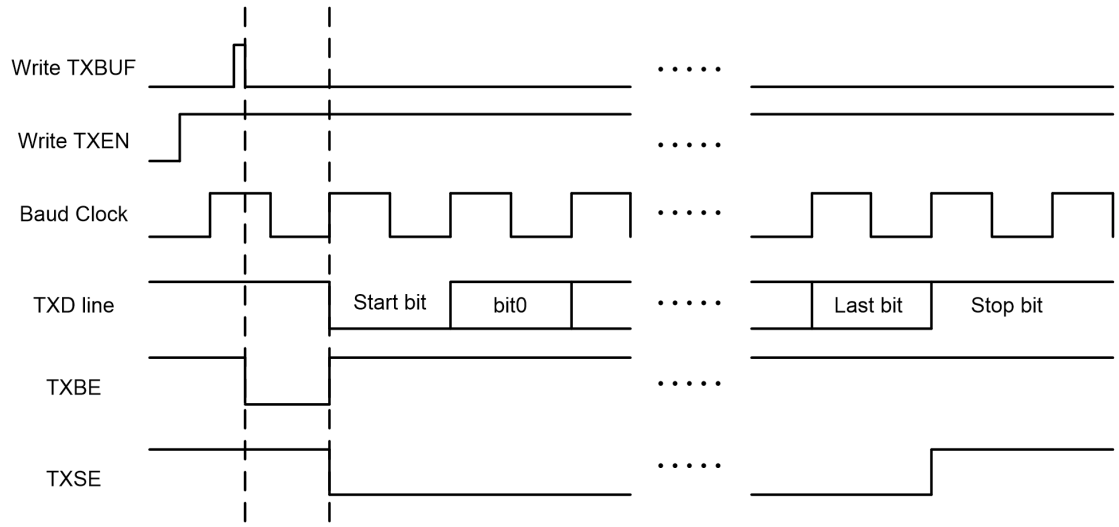


图 16-6 UART 异步发送波形 2

当TXBUF为空时，软件可以立即写入下一个待发送数据，以实现连续无间隔的数据发送。

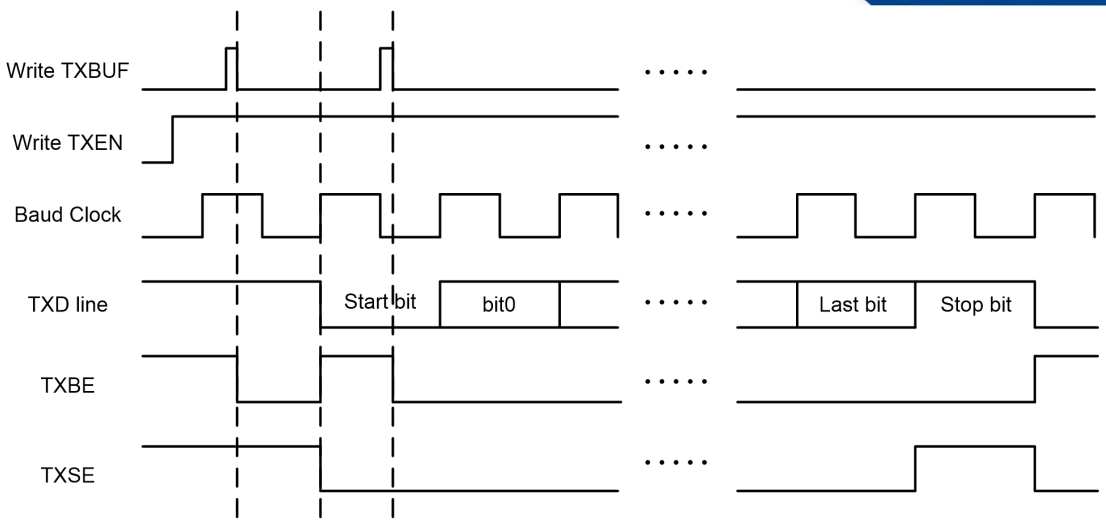


图 16-7 UART 异步发送波形 3

16.5.5 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

16.5.6 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：

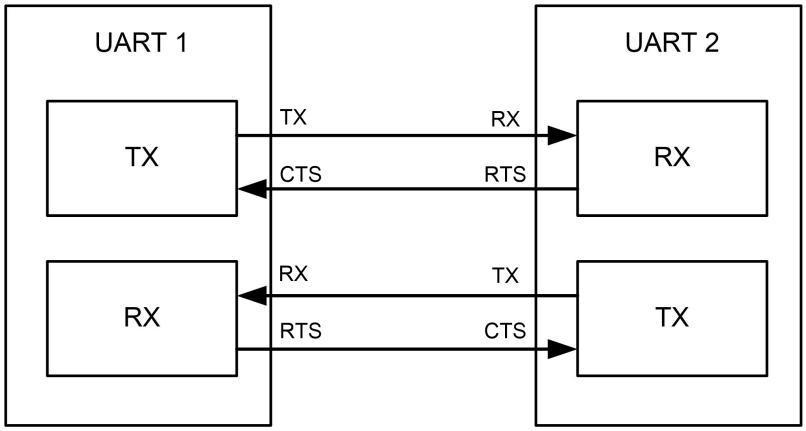


图 16-8 两个 UART 之间的硬件流控

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。

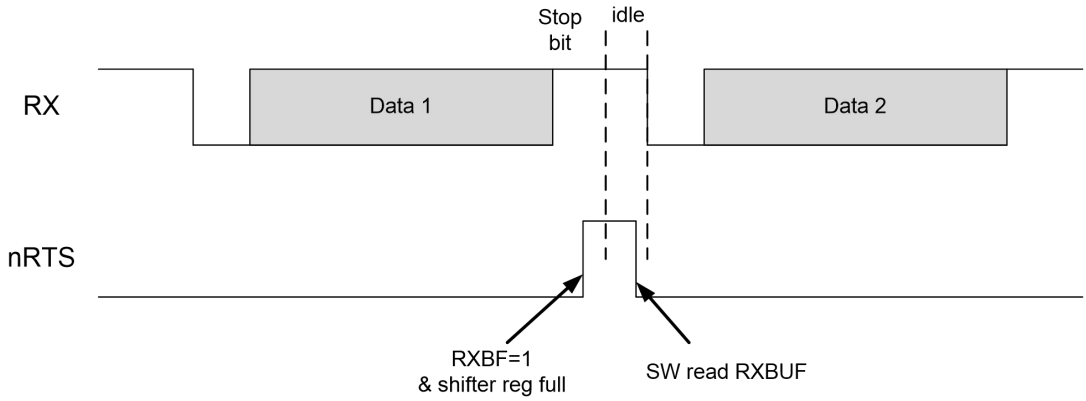


图 16-9nRTS 流控

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。

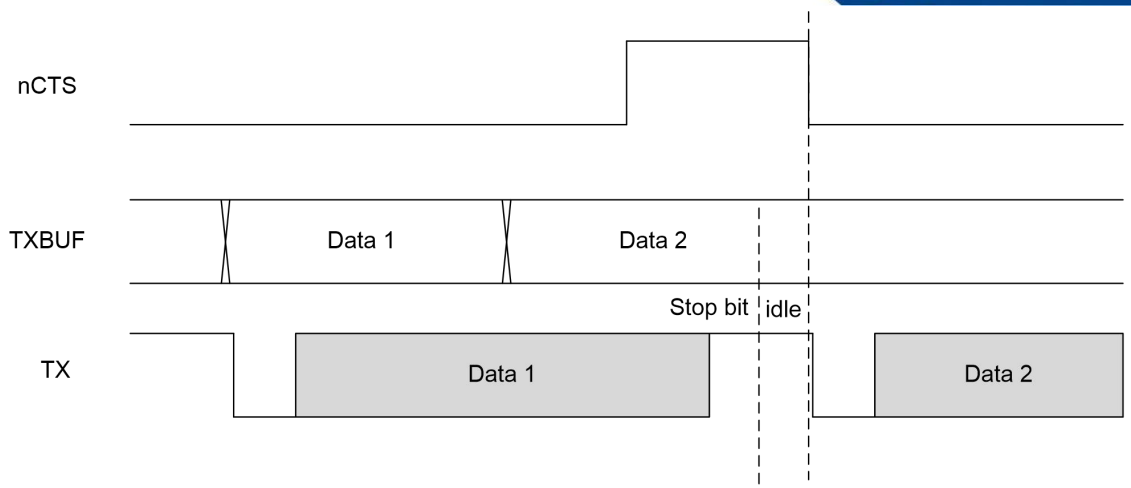


图 16-10nCTS 流控

16.5.7 低功耗休眠唤醒（UART0/1）

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

16.5.8 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 1 或 3 配置为 RXD0
- 设置对应通道参数：RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 UART 模块参数
- 使能 UART 模块接收使能，等待数据接收
- 收到数据后 UART 自动产生 DMA 请求
- DMA 响应请求，读取 UART 接收缓存寄存器，写入指定 RAM 地址

16.5.9 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

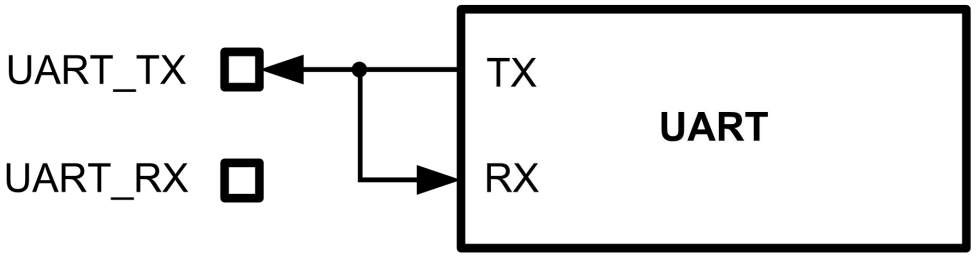
表 16-3 DMA 发送中断

16.5.10 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



16.6 波特率发生

16.6.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

UART 波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 UART4 和 UART5，F_{CLK} 就是 APBCLK；对于 UART0 和 UART1，F_{CLK} 是独立与 APBCLK 的工作时钟。

为了支持全双工通信，接收和发送波特率单独产生；

下表是常用系统时钟频率下的波特率：

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300	0	80000	299.9991	-0.00031	106667
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

表 16-4 常用时钟频率下波特率计算

16.6.1 波特率自适应

利用 Timer 的 Capture 功能，可以实现波特率自适应功能。可实现的一种方法为，外部 UART 设备按约定的数据内容(比如 0xF8)发送一帧，由 Timer 对该帧数据的高电平脉宽进行计数，MCU 读取 Timer 捕捉结果计算得到波特率因子，并写入波特率发生寄存器中，作为波特率发生的时钟分频计数值 X 使用。这时接收状态复位，重新等待起始位，以写入的波特率因子所产生的波特率接收数据。参考 Timer 章节。

16.7 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：

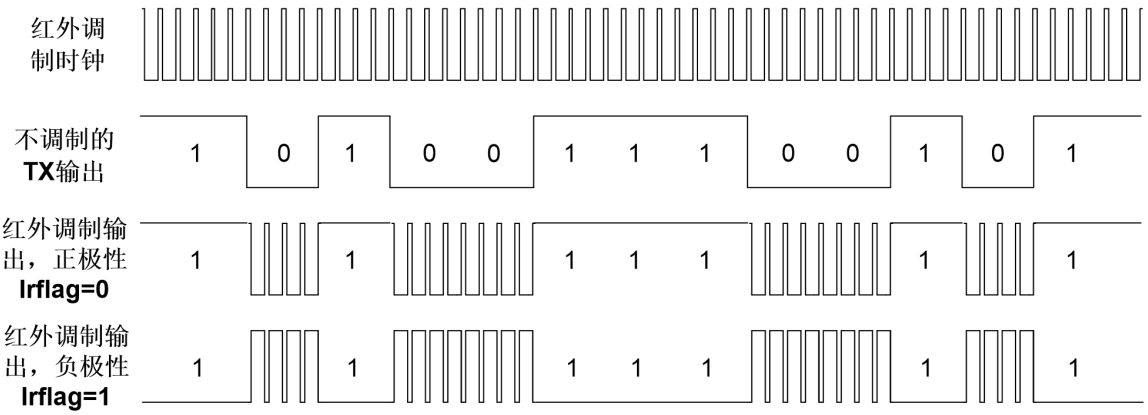


图 16-11 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

16.8 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。

16.9 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。

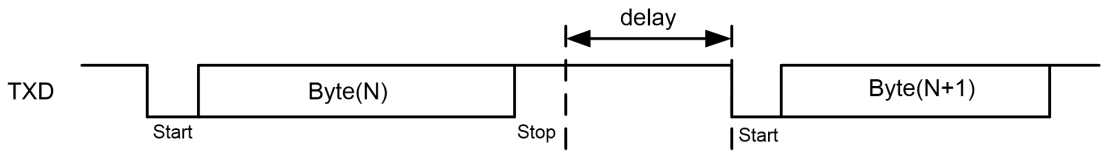


图 16-12 UART 发送延迟

16.10 LIN 总线通信功能

16.10.1 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。

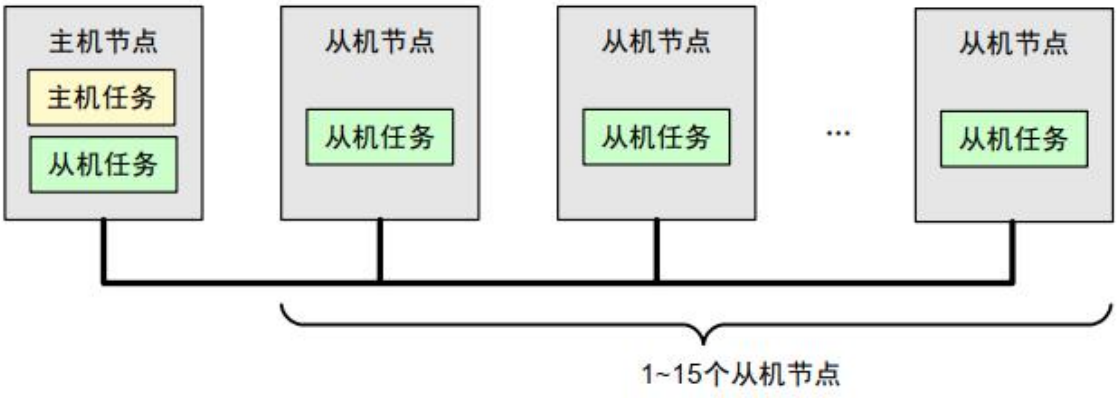


图 16-13LIN 总线拓扑

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。

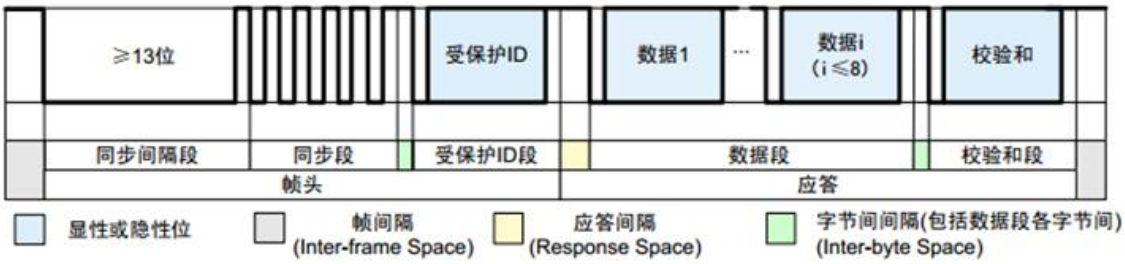


图 16-14LIN 总线帧结构

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)。

16.10.1.1 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 16-15LIN break field

16.10.1.2 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

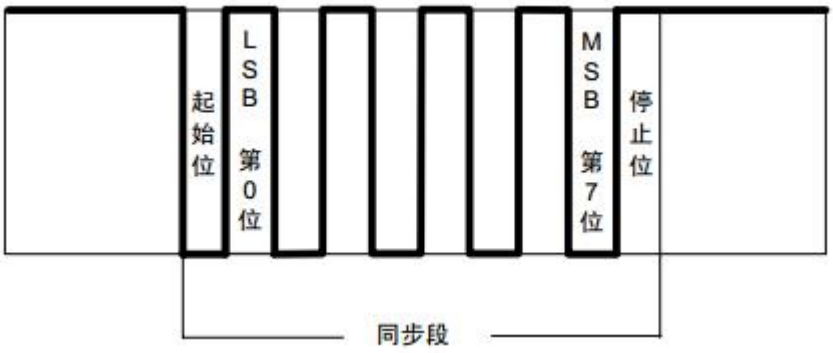


图 16-16LINsync field

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)。

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

同步段结束后，从机软件根据计数值完成通信波特率的自动调整。

16.10.1.3 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 16-17LIN 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。

16.10.1.4 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。

16.10.2 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。

16.10.2.1 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。

16.10.2.2 Break 和 Sync 发送

Break field发送固定为13bit，由软件置位TX_BREAK寄存器发送。

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。

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。

16.10.3 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。

16.10.3.1 唤醒检测

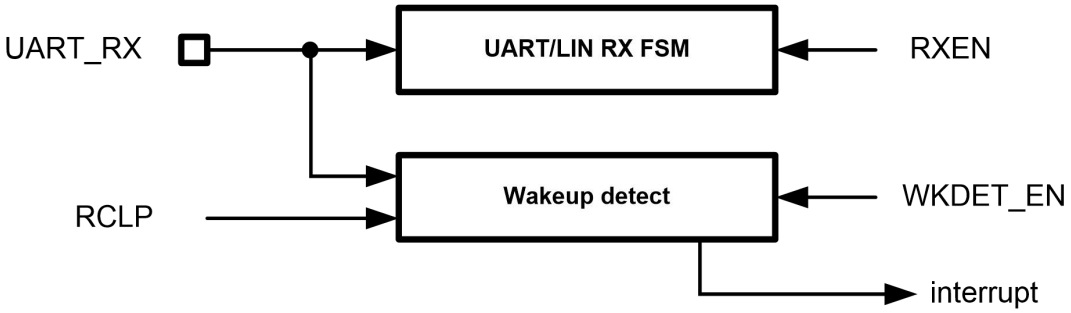
LIN总线处于休眠状态时，主/从机节点都可以向总线上发送唤醒信号，唤醒信号持续 250us～5ms。其余节点（除发送唤醒信号以外的节点）以大于150us为阈值判定唤醒信号。每个从机节点必须在唤醒信号显性脉冲的结束处算起100ms以内准备接收来自主机的命令(帧头)；主机节点也必须被唤醒，100ms之内主机节点发送帧头开始通信。



图 16-18LIN 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，RXD经过RCLP采样会导致1~2个RCLP时钟的延迟，假如设置WKUP_LEN=5，实际RXD低电平长度需达到6~7个RCLP周期，才会触发唤醒标志。

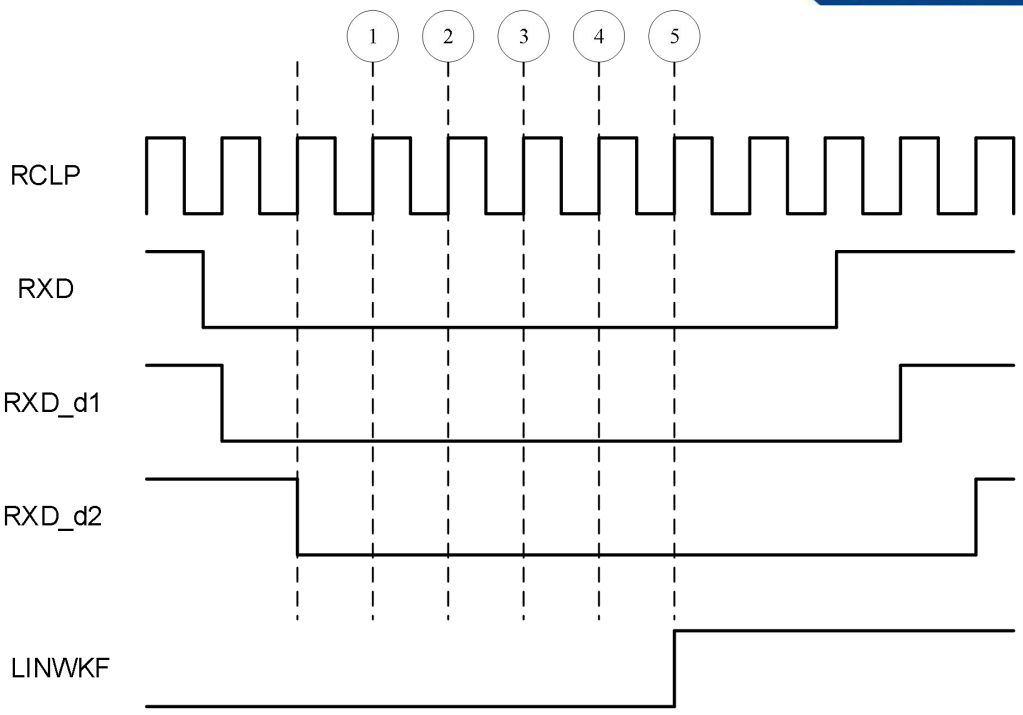


图 16- 19LIN 唤醒检测，WKUP_LEN=5

注意：

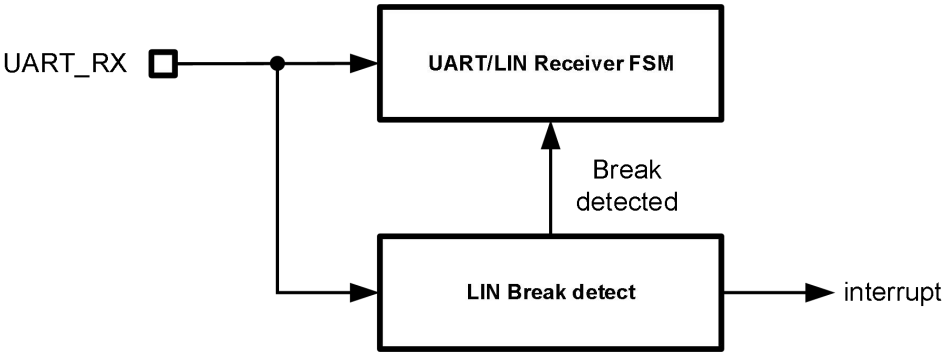
唤醒检测仅用于产生标志，当芯片处于休眠模式时，LIN唤醒检测标志可以用于唤醒MCU。

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。

16.10.3.2 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。

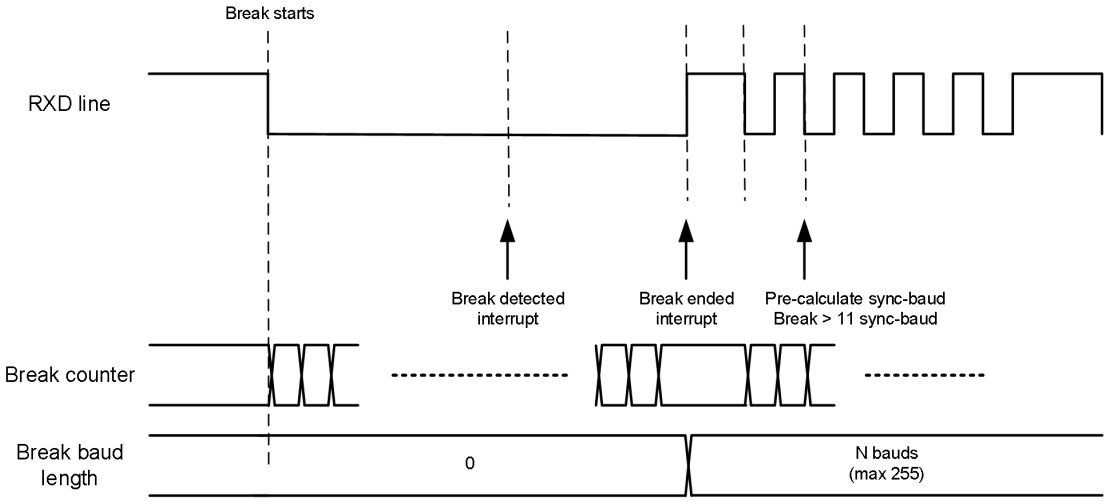


图 16-20Normal Break

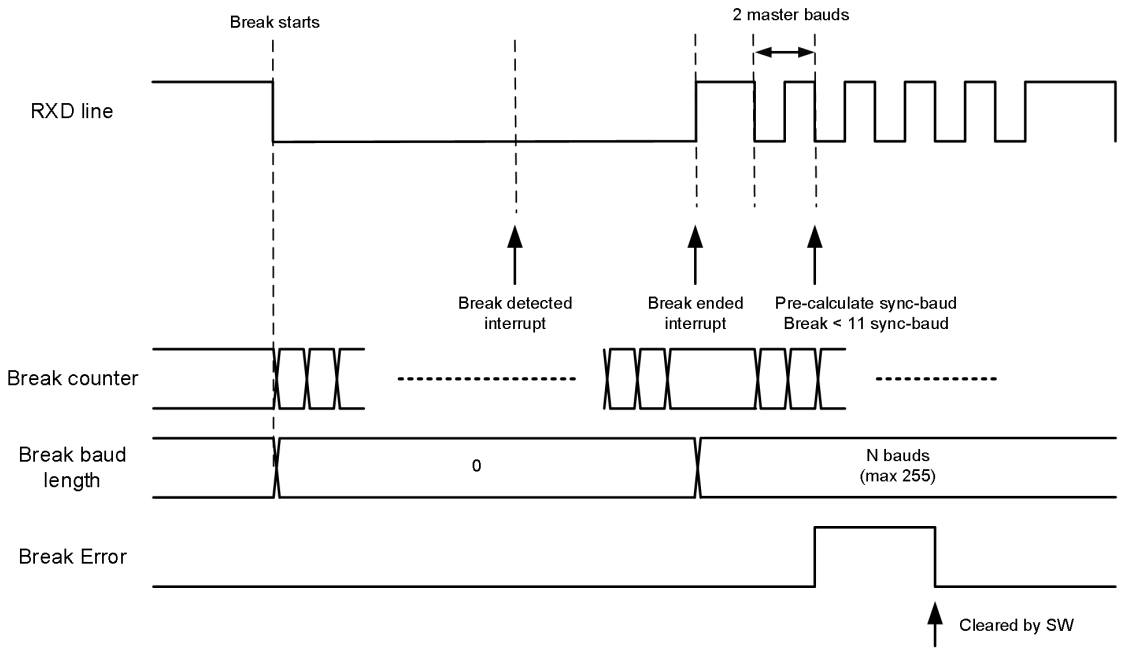


图 16-21Break Error

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。

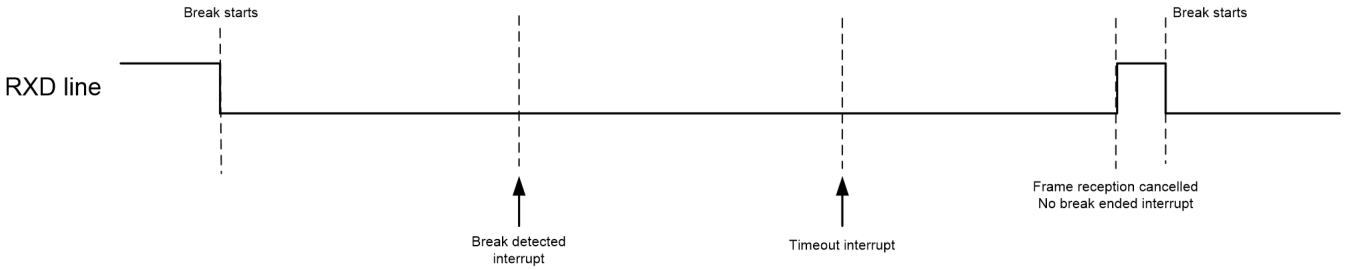


图 16-22Break 超时

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，停止位错误）

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。

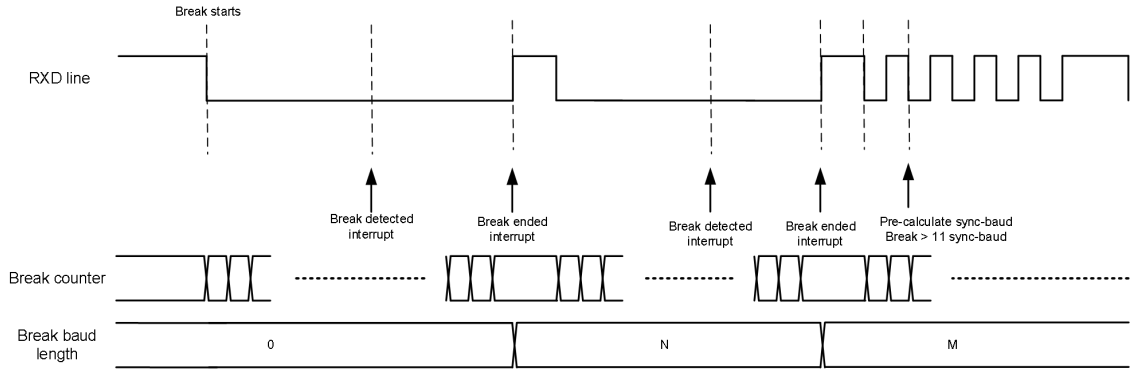


图 16-23 连续 Break 后正常接收

16.10.3.3 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*

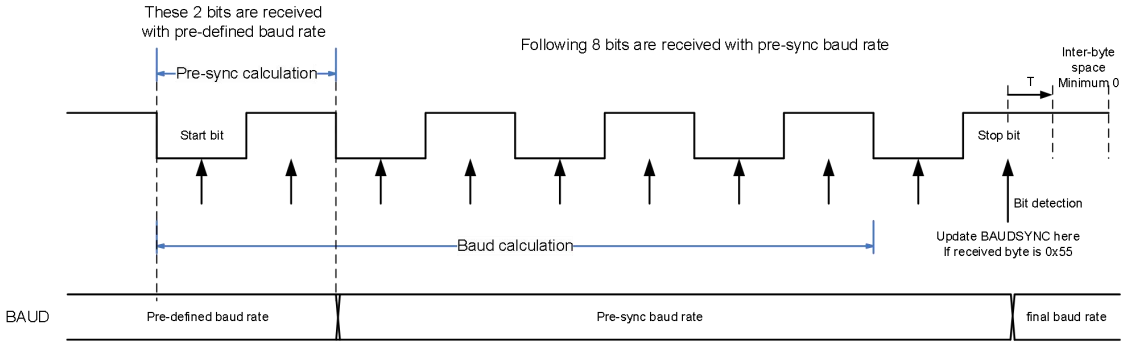


图 16-24LIN 波特率自适应过程

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）

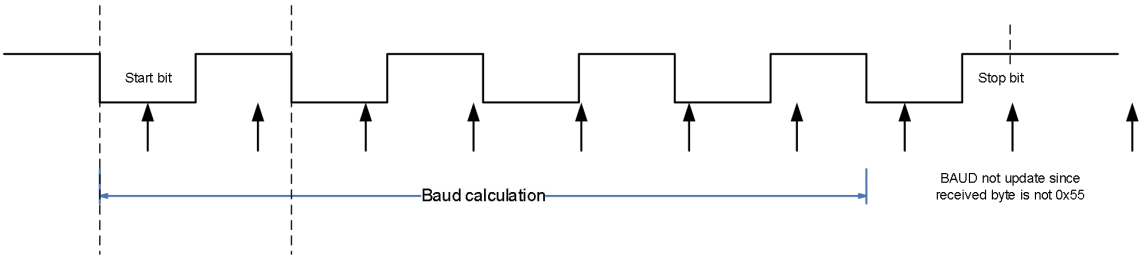


图 16-25 波特率误差较大，无 pre-sync

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。

16.10.3.4 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。

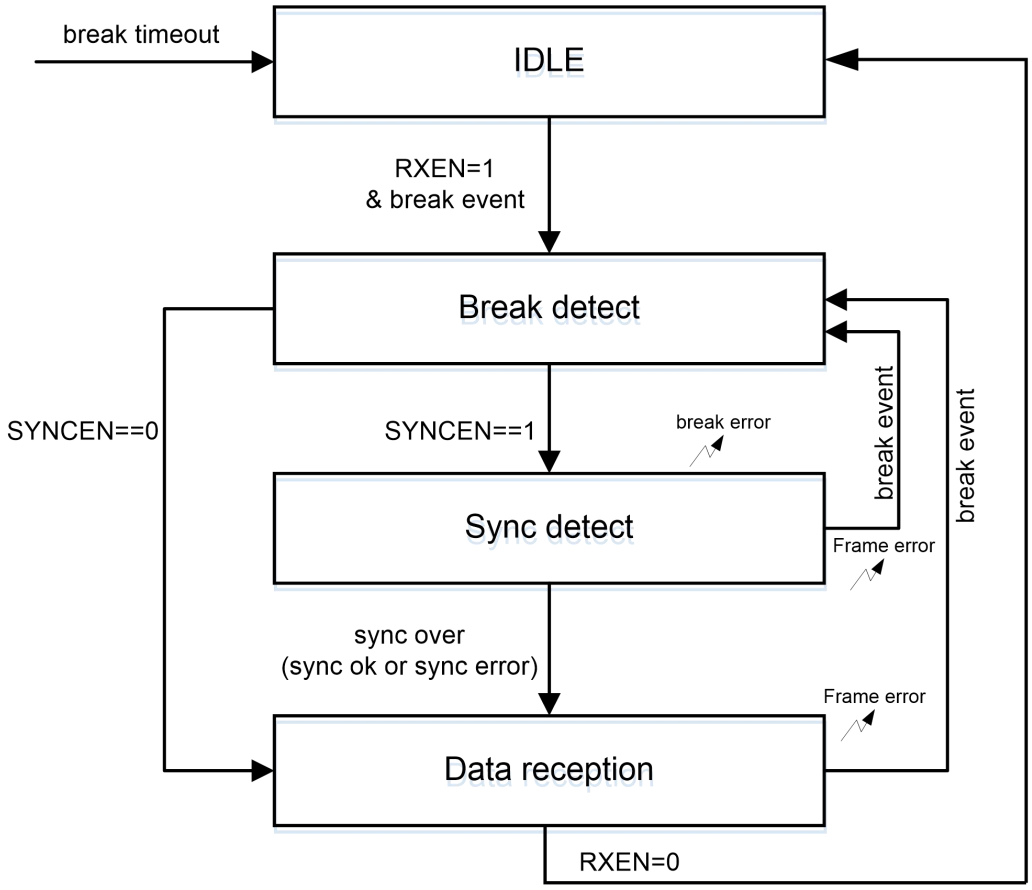


图 16-26LIN 从机接收流程

16.10.3.5 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。

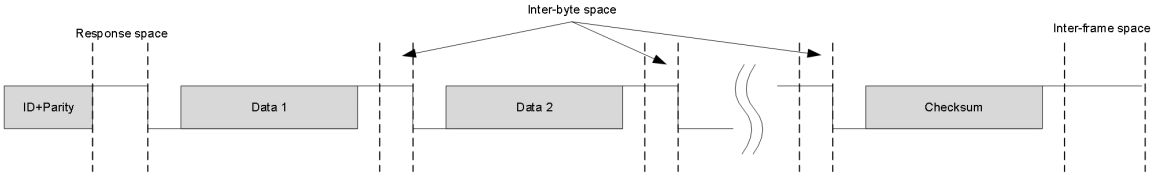


图 16-27 字节间隔和应答间隔

校验和由软件计算，校验和的发送与数据字节相同。

16.10.3.6 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。

16.10.4 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。

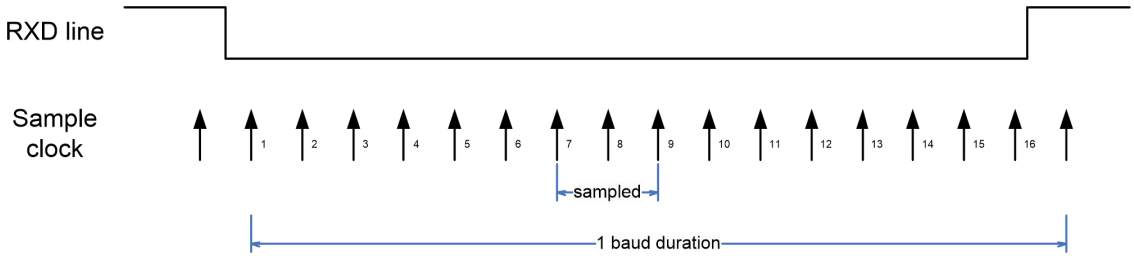


图 16-28LIN bit sampling

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.

16.10.5 LIN 错误状态

16.10.5.1 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。

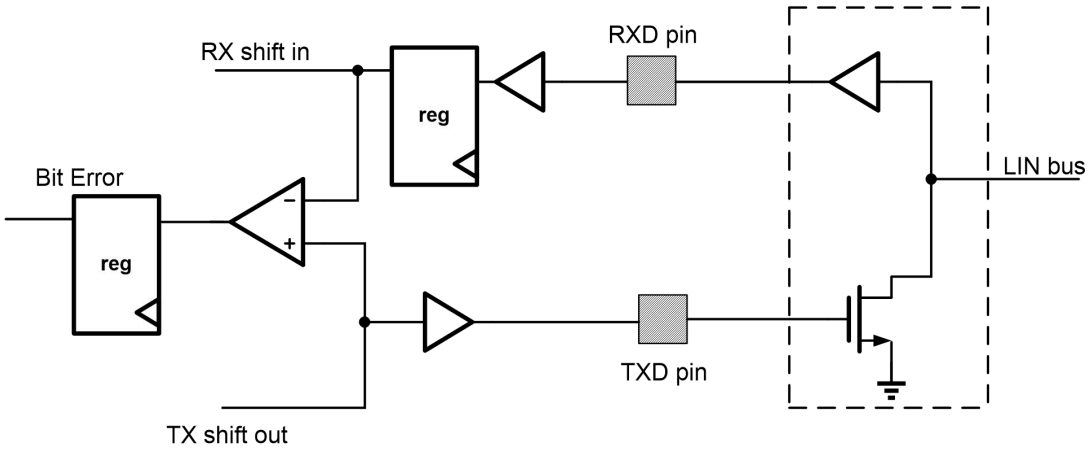


图 16-29LIN bit error 冲突检测

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：

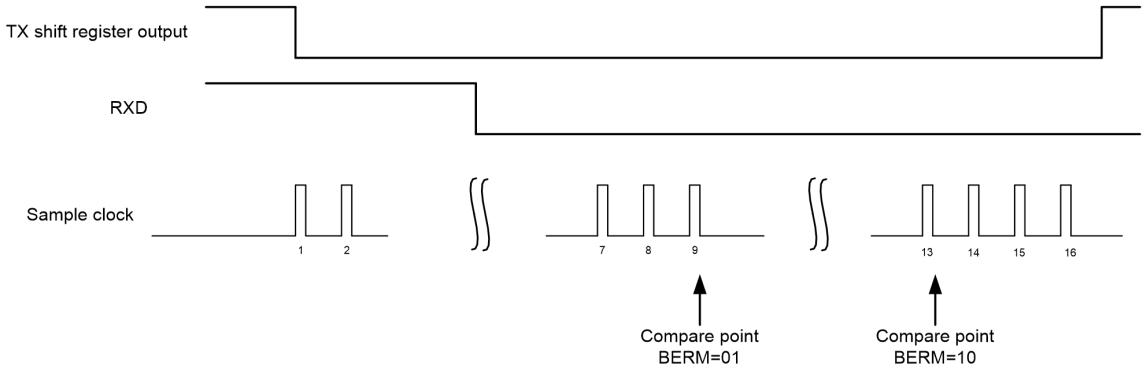


图 16-30LIN bit error 监测点

当 BERIF 置位时，LIN 模块采取以下动作：

- BERS=1
  - 下一个 bit 发送无效电平（当 TXPOL=0 时发送高电平，当 TXPOL=1 时发送低电平）
  - 当前发送取消，发送 buffer 数据清零
- BERS=0
  - 后续数据正常发送，不会发送无效电平，也不会清零发送 buffer
- 发送完成标志和 bit 错误标志 BERIF 同时置位
- 软件重新发送前应先清零 BERIF
  - BERS=1：不清零 BERIF 的情况下无法重新启动发送
  - BERS=0：不清零 BERIF 也可以启动新的发送

16.10.5.2 timeout 错误

timeout 错误检测 frame 中连续低电平长度（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。

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]

16.10.5.3 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。

16.10.5.4 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。

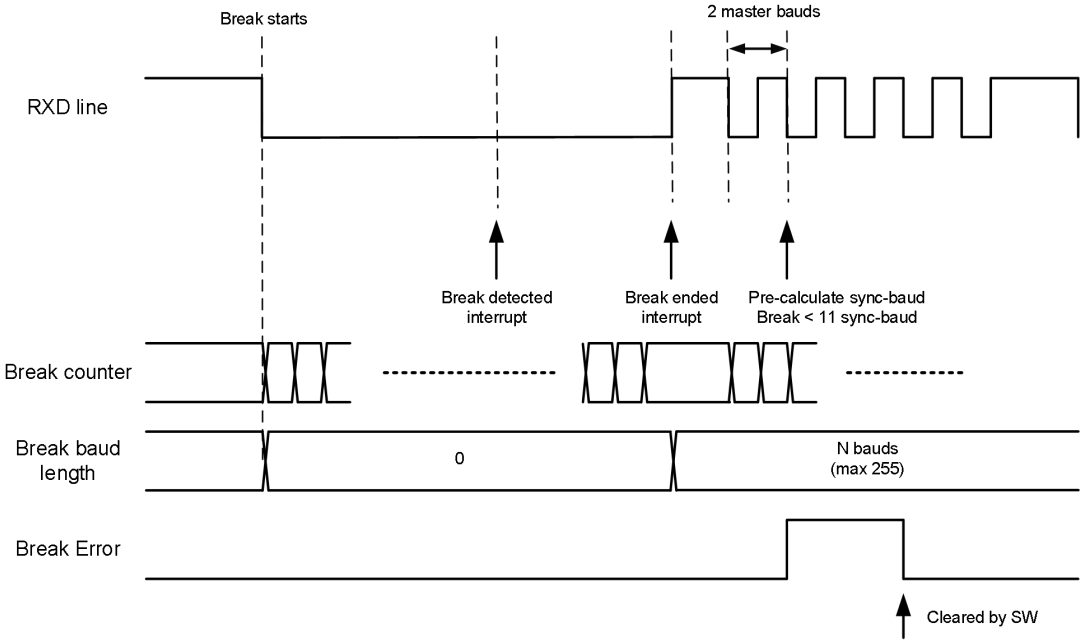


图 16-31Pre-sync 后发现 Break 长度不足，BreakError

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。

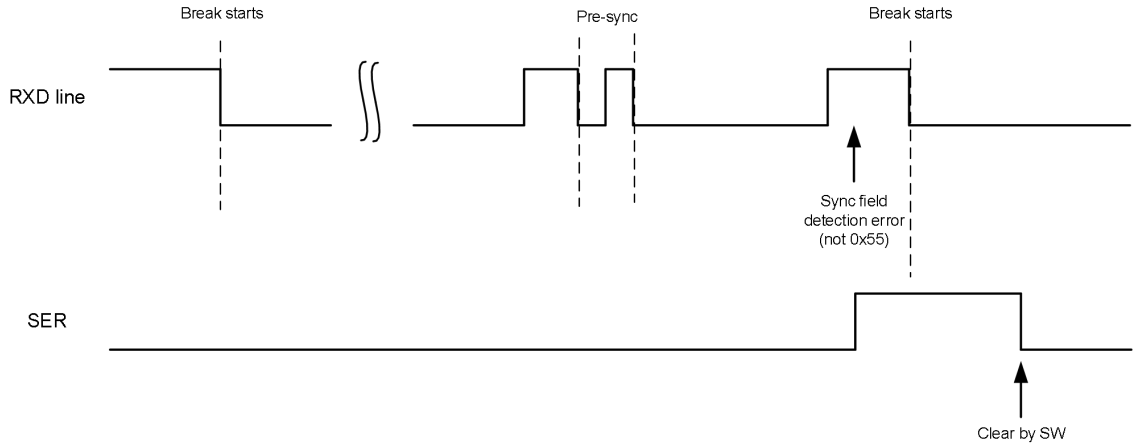


图 16-32SYNC 段数据错误

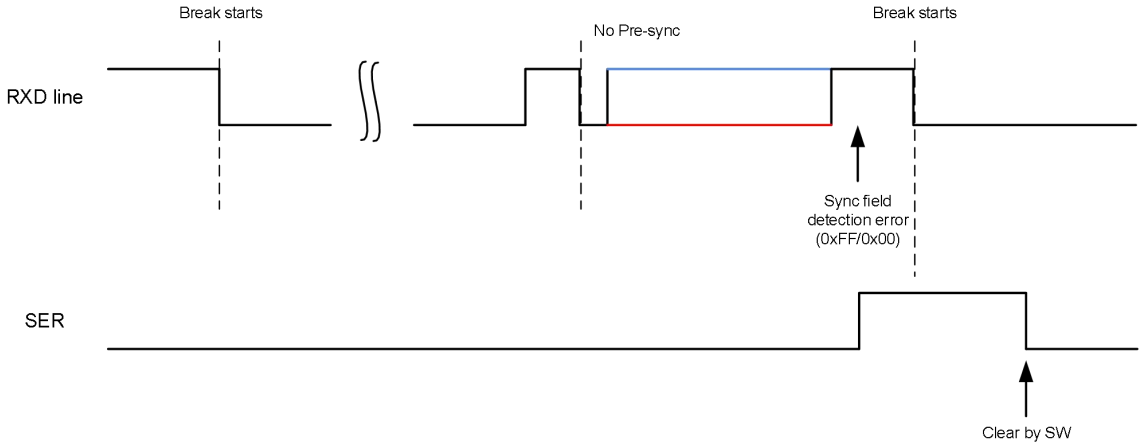
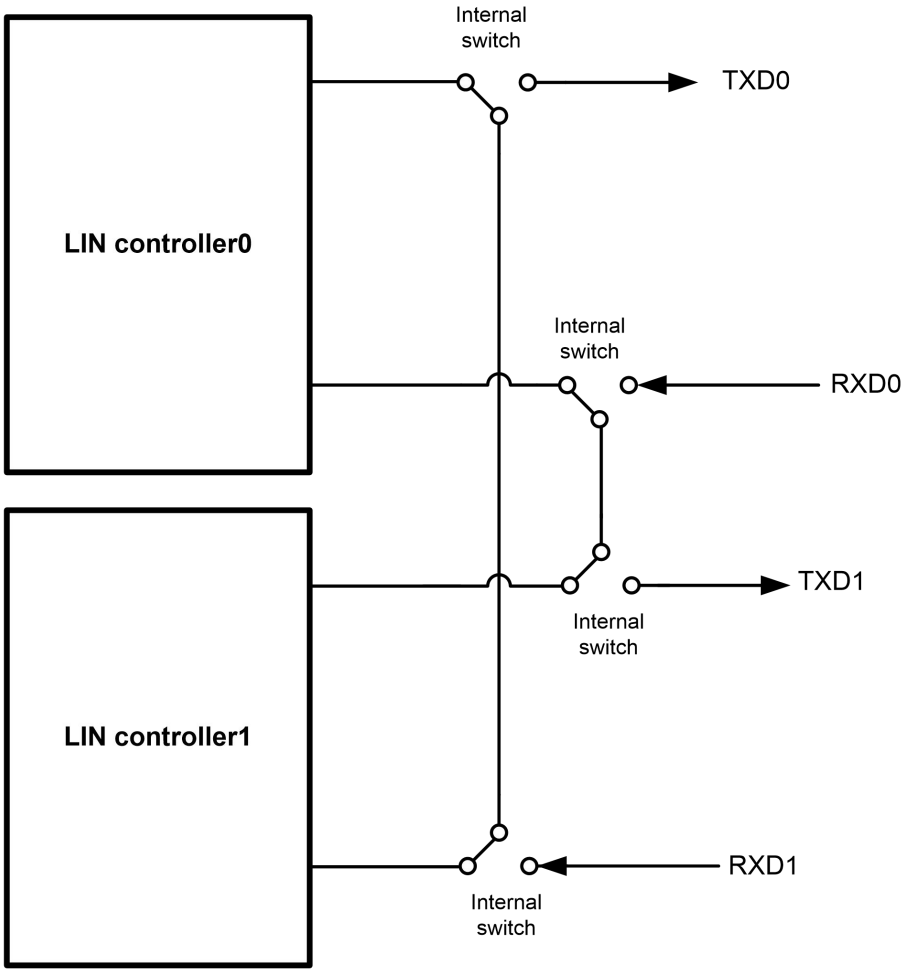


图 16- 33SYNC 段 0xFF 或 0x00，无 pre-sync

16.10.6 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

16.11 寄存器

UART0模块基地址：0x4001 1C00

UART1模块基地址：0x4001 2000

UART2模块基地址：0x4001 7400

UART3模块基地址：0x40017800

UART4模块基地址：0x4001 A000

UART5模块基地址：0x4001 A400

UARTIR模块基地址：0x40017C00

offset 地址	名称	符号
UARTIR 寄存器		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
UARTx 寄存器		
0x00	UART 控制状态寄存器 （UART Control Status Register）	UART_CSR
0x04	UART 中断使能寄存器 （UART Interrupt Enable Register）	UART_IER
0x08	UART 中断标志寄存器 （UART Interrupt Status Register）	UART_ISR
0x0C	UART 超时和延迟寄存器 （UART Time-Out and Delay Register）	UART_TODR
0x10	UART 接收缓冲寄存器 （UART Receive Buffer）	UART_RXBUF
0x14	UART 发送缓冲寄存器 （UART Transmit Buffer）	UART_TXBUF
0x18	UATR 波特率产生寄存器 （UART Baud rate Generator Register）	UART_BGR
0x20	UART 模式寄存器 （UART Mode Control Register）	UART_MCR
0x24	UART LIN 控制寄存器 （UART LIN Control Register）	UART_LINCR
0x28	UART LIN 波特率同步寄存器 （UART LIN Baud Sync Register）	UART_LINBSR
0x2C	UART LIN 帧超时寄存器 （UART LIN Frame Timeout Register）	UART_LINFTR
0x30	UART LIN 发送时序寄存器 （UART LIN Transmit Timing Register）	UART_LINTTR
0x34	UART LIN Pre-sync 波特率寄存器 （UARTLINPre-sync baud register）	UART_LINPSR
0x38	UART LIN Break 寄存器 （UART LIN Break register）	UART_LINBKR
0x80	LIN 自测试开关控制寄存器	UART_LINSCCR

offset 地址	名称	符号
	（LIN Self-test Cwitch Control Register）	

16.11.1 红外调制寄存器（UART_IRCR）

名称	UART_IRCR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG[10:8]		
位权限	R/W-0	R/W-0000				R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG[7:0]							
位权限	R/W-11010010							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15	**IRFLAG**	控制红外调制发送数据时的默认输出极性 (Infra Red) 0：正极性 1：负极性
14:11	**TH**	红外占空比调制参数 (Tranmission High Duty)
10:0	**TZBRG**	红外调制频率 (Transmission Baud Rate)

16.11.2 UARTx 模式寄存器（UARTx_MCR）

名称	**UARTx_MCR(x=0~5)**							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						LINMD	MODE
位权限	U-0						R/W-0	R/W-0

位号	助记符	功能描述

位号	助记符	功能描述
31:2	-	未实现：读为0
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

16.11.3 UARTx LIN 控制寄存器（UARTx_LINCR）

名称	UARTx_LINCR(x=0~5)							
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	-							BERS
位权限	U-0							R/W-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
31:17	-	未实现：读为0
16	**BERS**	LIN 总线发送冲突配置 0：检测到发送冲突时不会禁止后续发送 1：检测到发送冲突时取消后续发送，直到 BERIF 清零
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 注：只发送时才能使能该配置，否则在只接受时会误检测。

位号	助记符	功能描述
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*

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

名称	UARTx_LINBSR(x=0,1)							
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
19:0	**BAUD_SYNC**	同步段计数器，计算同步段的时钟个数，用于波特率同步 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

16.11.5 UARTx LIN 帧超时寄存器（UARTx_LINFTR）

名称	UARTx_LINFTR(x=0~5)							
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 定义低电平最大时间长度

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

名称	UARTx_LINTTR(x=0~5)							
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

位号	助记符	功能描述
3:0	**BLT**	Break 长度 0000: 13 baud 0001: 14 baud ... 1110: 27 baud 1111: 28 baud

16.11.7 UARTx LIN Pre-sync 寄存器（UARTx_LINPSR）

名称	UARTx_LINPSR(x=0~5)							
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-0
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。

16.11.8 UARTx LIN Break-len 寄存器（UARTx_LINBKR）

名称	UARTx_LINBKR(x=0~5)							
offset	0x38							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-				BREAK_LEN			
位权限	U-0000				R-1111			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BREAK_LEN							
位权限	R-1111 1111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BREAK_LEN							
位权限	R-1111 1111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BREAK_LEN							
位权限	R-1111 1111							

位号	助记符	功能描述

位号	助记符	功能描述
31:20	-	未实现：读为0
27:0	BREAK_LEN	break 段长度计数结果，此结果用于 braek 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。

16.11.9 UARTx 控制状态寄存器（UARTx_CSR）

名称	UARTx_CSR(x=0~5)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							BUSY
位权限	U-0							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	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
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

位号	助记符	功能描述
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
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)

16.11.10 UARTx 中断使能寄存器（UARTx_IER）

名称	UARTx_IER(x=0~5)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

位名	TOE_IE	SER_IE	-	BER_IE	NF_IE	LINSYNC_IE	LINB_IE	LINWK_IE
位权限	R/W-0	R/W-0	U-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_IE	LINBEF_IE	LINBER_IE	-	-	PERR_IE	FERR_IE	OERR_IE
位权限	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-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	**-**	**未实现：读为0**
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**	从机模式 LIN 唤醒检测标志中断使能，1 有效
23	**LINWKTF_IE**	主机模式LIN唤醒发送完成标志中断使能，1有效
22	**LINBEF_IE**	从机模式LIN Break段结束标志中断使能，1有效
21	**LINBER_IE**	从机模式LIN Break段和同步段不匹配标志中断使能，1有效
20:19	**-**	**未实现：读为0**
18	**PERR_IE**	奇偶校验错误中断使能，1有效
17	**FERR_IE**	framing error中断使能，1有效
16	**OERR_IE**	接收缓存溢出中断使能，1有效
15	**CTSIE**	nCTS翻转中断使能，1有效
14:12	**-**	**未实现：读为0**
11	**RXTO_IE**	接收超时中断使能，1 有效 (Receive Time-Out Interrupt Enable) （仅 UART0 和 UART1 有效）
10	**RXERR_IE**	接收错误中断使能，1 有效(Receive Error Interrupt Enable)
9	**-**	**未实现：读为0**
8	**RXBF_IE**	接收缓存满中断使能，1 有效 (Receive Buffer Full Interrupt Enable)
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)

16.11.11 UARTx 中断标志寄存器（UARTx_ISR）

名称	UARTx_ISR(x=0~5)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOER	SER	-	BERIF	NF	LINSYNCF	LINBF	LINWKF
位权限	R/W-0	R/W-0	U-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	LINBER	-		PERR	FERR	OERR
位权限	R/W-0	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	**-**	未实现：读为0
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
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	**LINBER**	LIN从机模式：LIN Break段错误中断标志，硬件置位，软件写1清零
20:19	**-**	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

位号	助记符	功能描述
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)

16.11.12 UARTx 超时和延迟寄存器（UARTx_TODR）

名称	UARTx_TODR(x=0~5)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
位权限	R/W-1111 1111							

位号	助记符	功能描述
31:16	**-**	**未实现：读为0**
15:8	**TXDLY_LEN**	发送延迟，最大 255baud (Transmit Delay Length)
7:0	**RXTO_LEN**	接收超时溢出长度，最大 255baud (Receive Time-Out Length)

16.11.13 UARTx 接收缓冲寄存器（UARTx_RXBUF）

名称	UARTx_RXBUF(x=0~5)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							RXBUF[8]
位权限	U-0							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:9	-	未实现：读为0
8:0	**RXBUF**	接收数据缓冲寄存器数据(Receive buffer)

7位收发时，接收的7bits数据存入RXBUF[6:0]

16.11.14 UARTx 发送缓冲寄存器（UARTx_TXBUF）

名称	UARTx_TXBUF(x=0~5)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-0000 0000							

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

7位收发时，发送的7bits数据写入TXBUF[6:0]

16.11.15 UATRx 波特率产生寄存器（UARTx_BGR）

名称	UARTx_BGR(x=0~5)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG[15:8]							
位权限	R/W-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG[7:0]							
位权限	R/W-01000001							

位号	助记符	功能描述
31:16	-	未实现：读为0
15:0	**SPBRG**	波特率产生器寄存器值(Serial Port Baud Rate Generation) 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;

17 通用同异步收发传输器（USART）

17.1 概述

USART串行通信模块特点如下

- 波特率软件可配置
- 支持全双工通信口
- 支持半双工单线通信
- 支持RX-TX端口交换
- USART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
- 支持发送延迟功能

USART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

17.2 结构框图

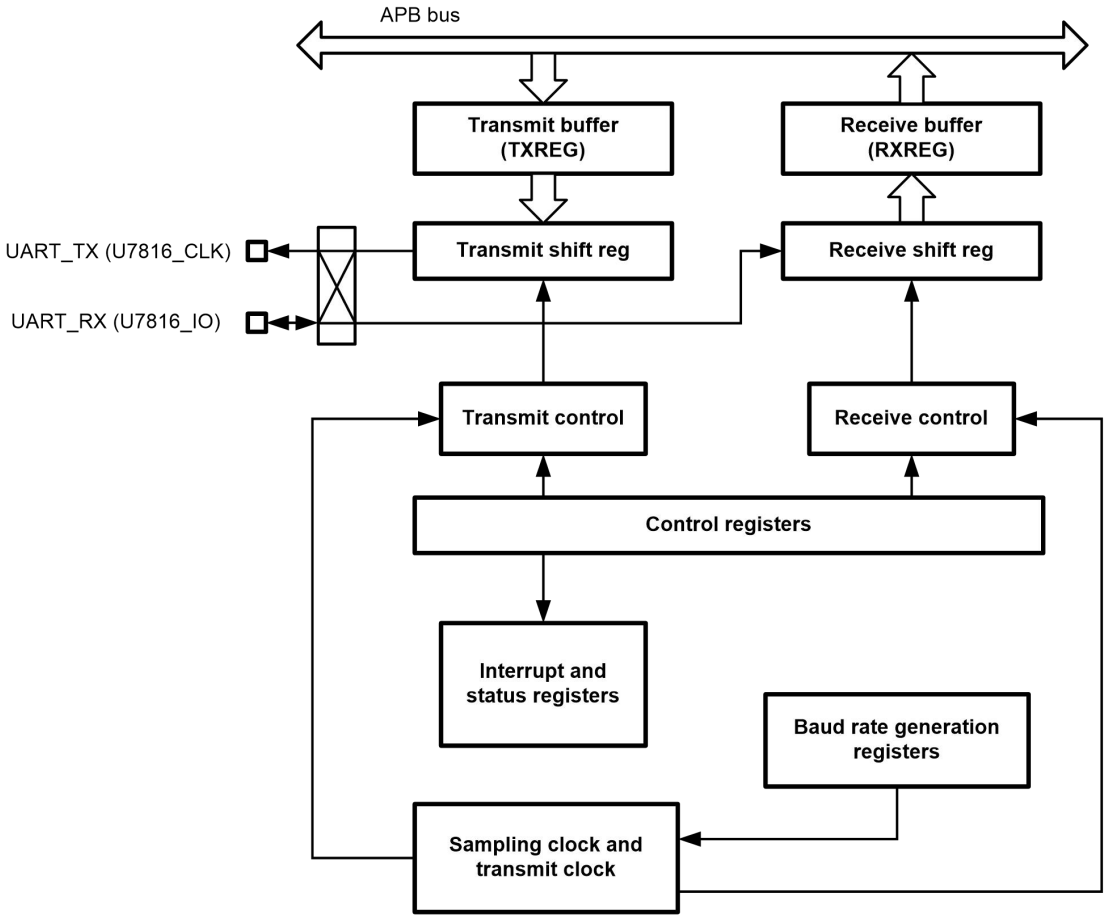


图 17-1 USART 模块框图

17.3 UART 类型区分

芯片集成了多种不同类型的UART（LPUART），其差异如下表所示：

UART 特性	UART (UART0/1/2/3/4/5)	USART (UART6/7)	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
半双工单线通信	Y	Y	-
休眠唤醒	Y	-	Y
接收超时	Y	Y	-
发送延迟	Y	Y	-
数据长度	6、7、8、9bits		

LIN support	Y	N	
智能卡模式	N	Y	N

表 17-1 UART 类型列表

17.4 USART 字符描述

17.4.1 UART 模式

UART 传输字符的基本时序如下图所示。每个字符帧包含至少 1bit START 位和至少 1bit STOP 位，数据长度可以配置为 6~9bits，并且可以选择有无校验位。

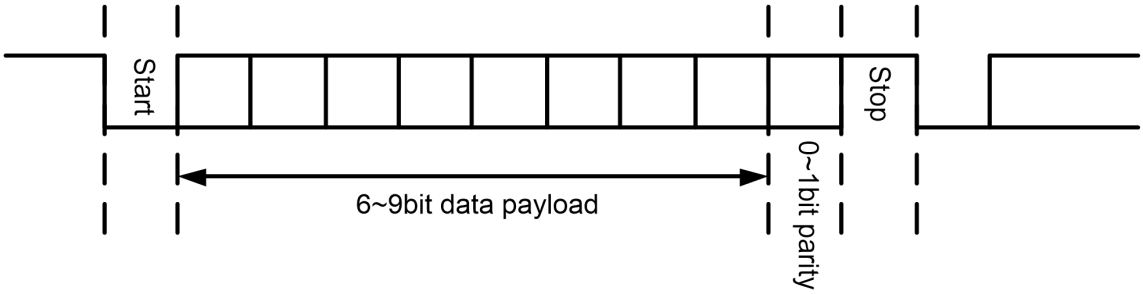


图 17-2 UART 字符描述

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 17-2 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】。

17.4.2 智能卡模式

ISO7816-3协议规定的字符格式如下图。

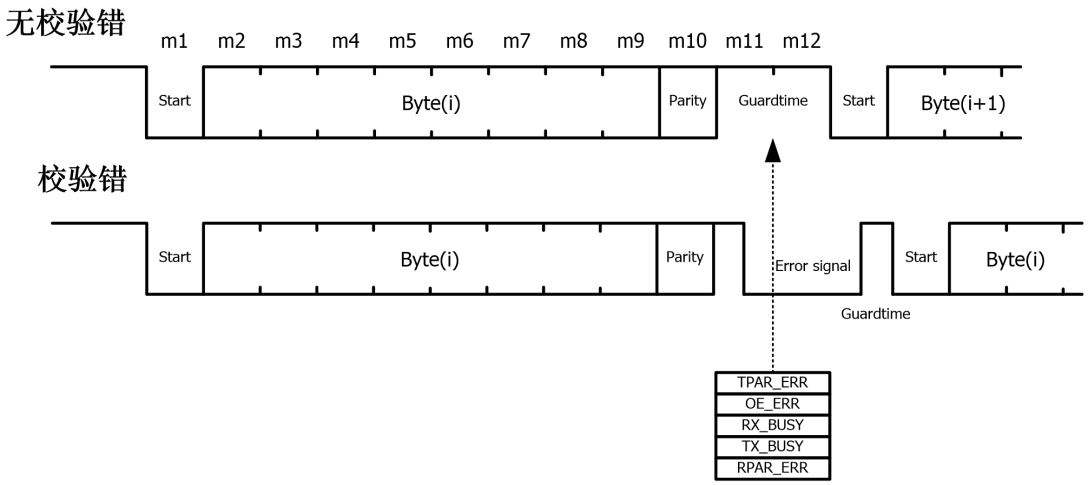


图 17-3 7816-3 帧结构图

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。

17.5 功能描述

17.5.1 时钟结构

USART 采用单时钟结构，此时 UCLK=PCLK，USART 的数据收发时钟也是来源于 APBCLK 的。

17.5.2 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。

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。

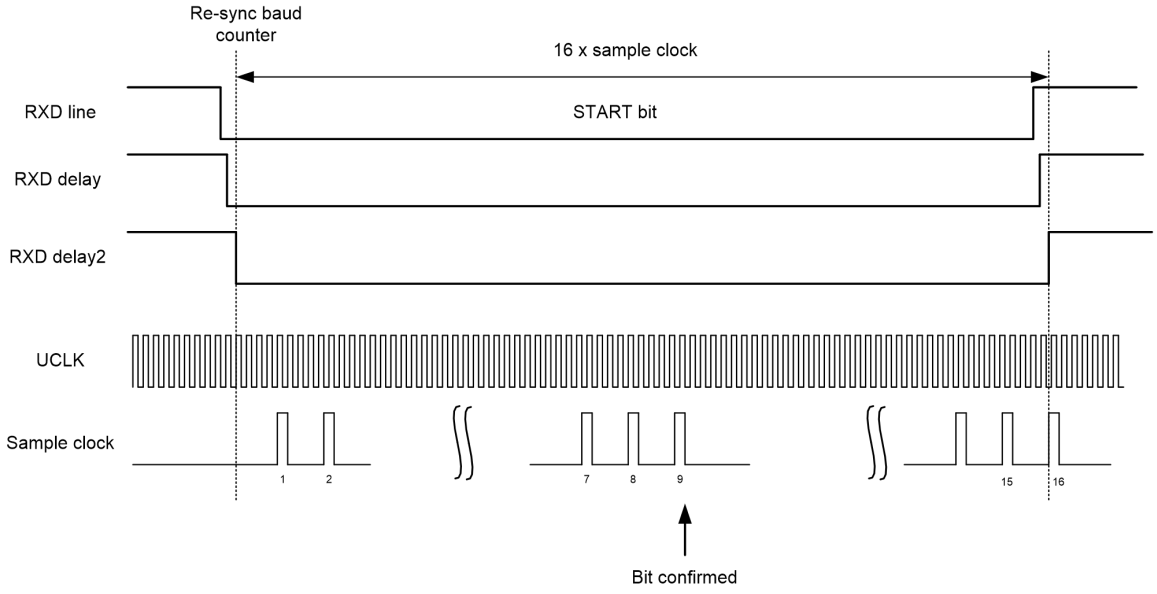


图 17-4 起始位同步

17.5.3 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。

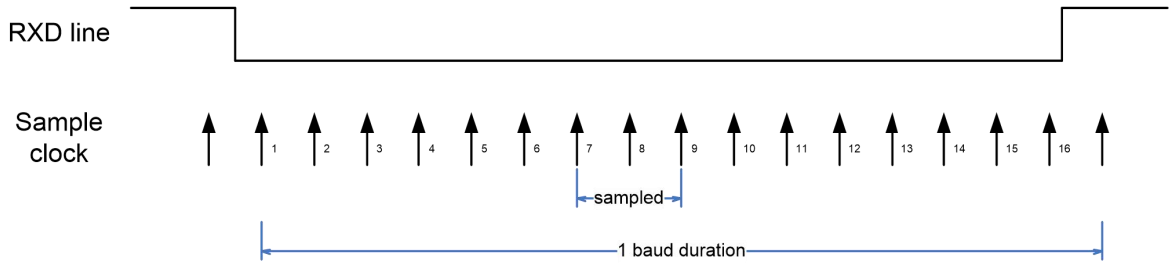


图 17-5 位接收 16 倍采样

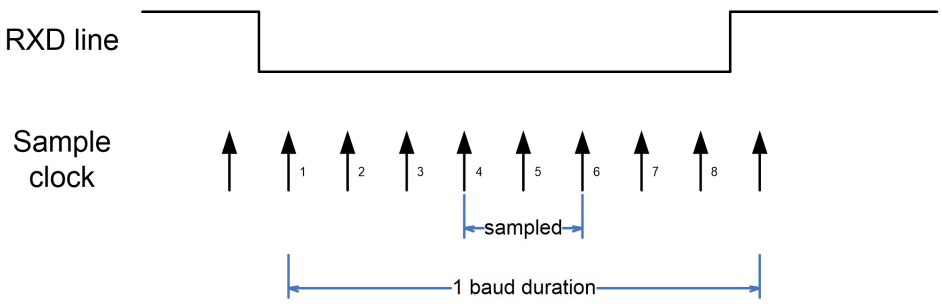


图 17-6 位接收 8 倍采样

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。

17.5.4 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(TXSR)，TXSR 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 TXSR 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 TXSR 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 TXSR 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。

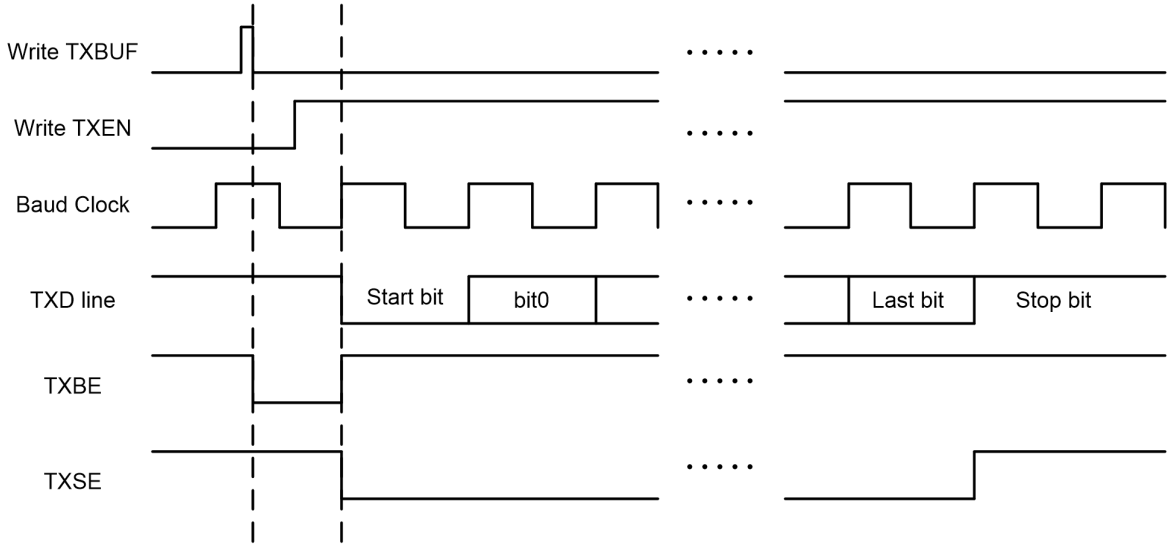


图 17-7 UART 异步发送波形 1

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。

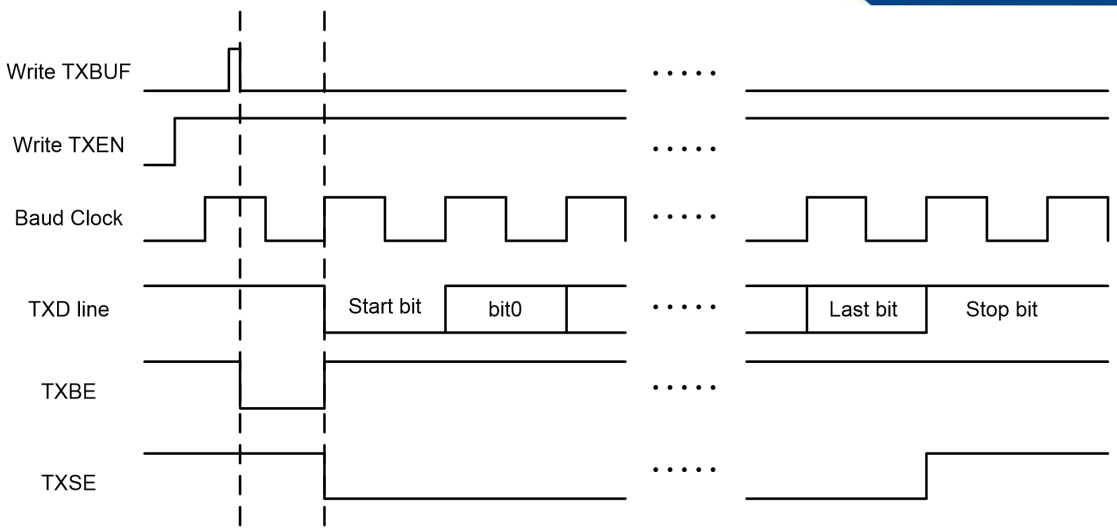


图 17-8 UART 异步发送波形 2

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。

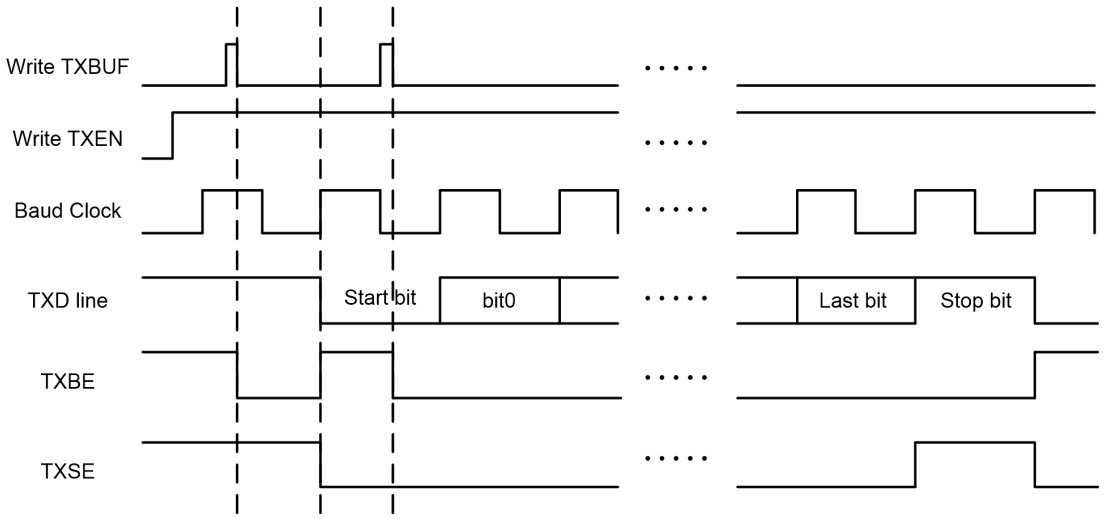


图 17-9 UART 异步发送波形 3

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，TXSE不会置位，直到重发结束；此时TXBUF中的数据也不会写入TXSR，

如果TXBF中有数据，则写TXBUF将触发ovflow错误。

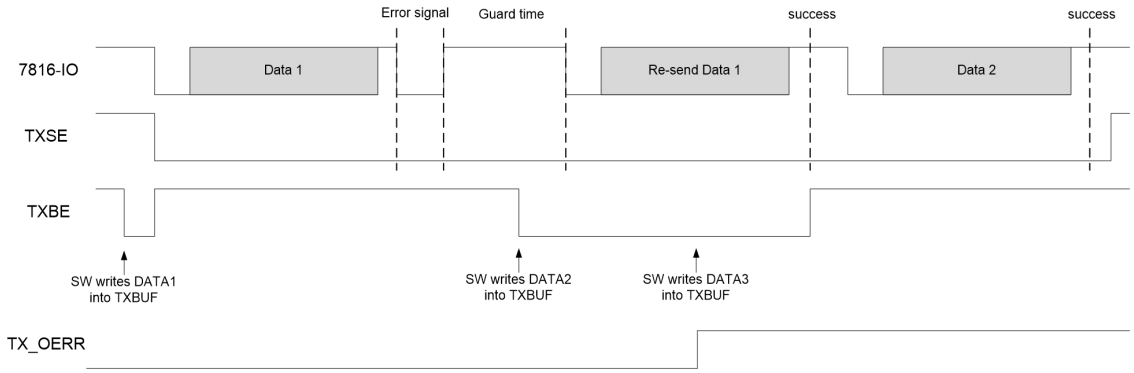


图 17-107816 发送和重发

17.5.5 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。

17.5.6 RS232 硬件流控（new）

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：

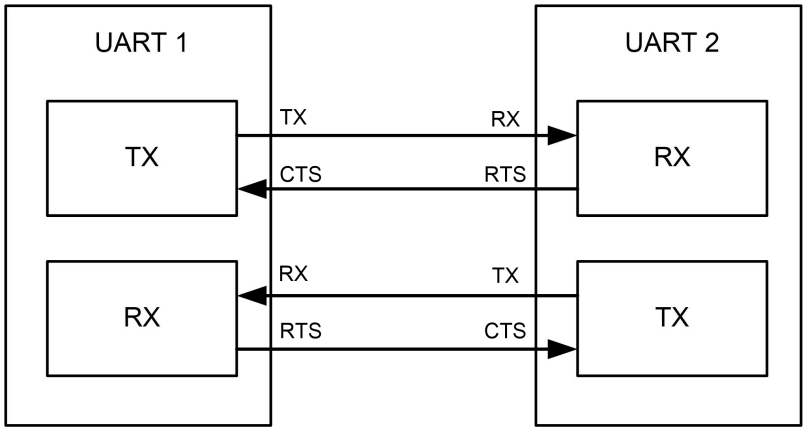


图 17-11 两个 UART 之间的硬件流控

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。

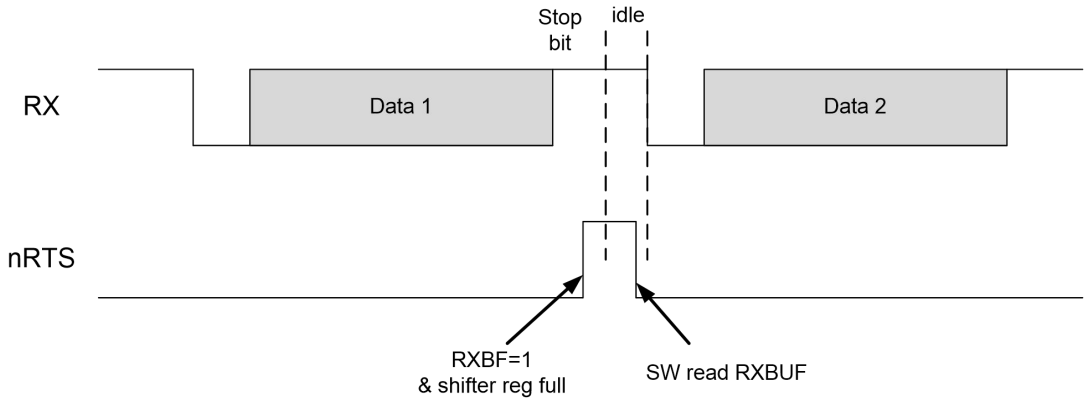


图 17-12nRTS 流控

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。

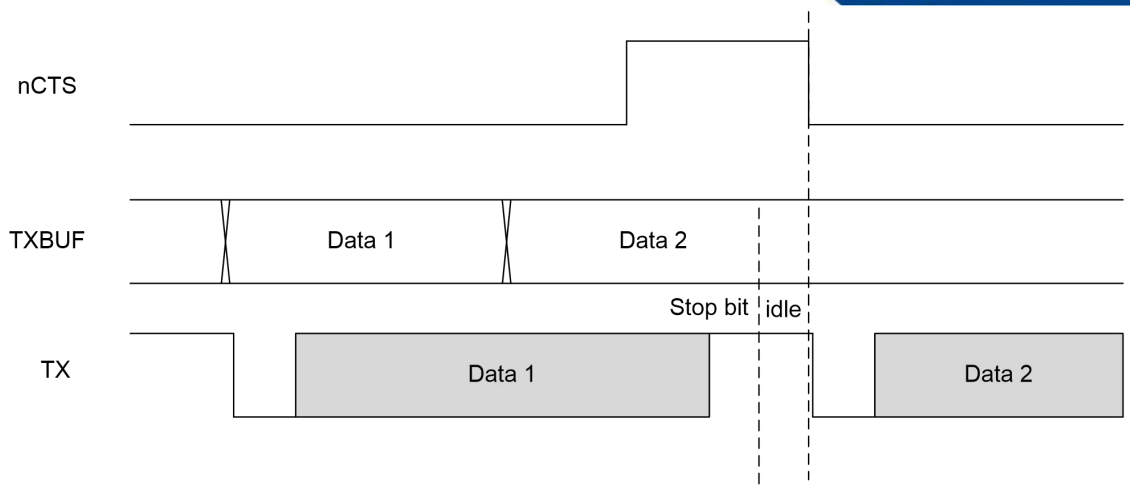


图 17-13nCTS 流控

17.5.7 低功耗休眠唤醒

USART不支持休眠唤醒

17.5.8 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。

17.5.9 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
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

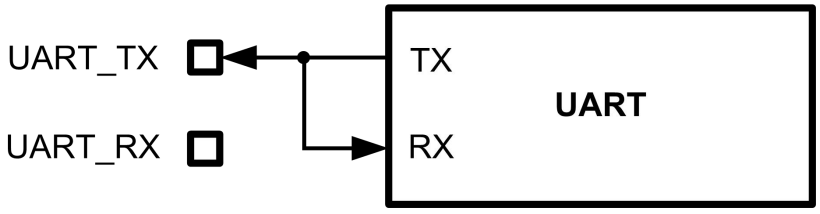
表 17-3 DMA 发送中断

17.5.10 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



17.6 波特率发生

17.6.1 波特率发生

波特率因子寄存器是一个 16 位的可读写的寄存器，其值 X 为 16—65535 之间的任一整数。

UART 波特率计算公式：

Baud = F_{CLK} / (SPBRG+1)；

注：F_{CLK} 在不同的 UART 中可以是不同的时钟，对于 USART，F_{CLK} 就是 APBCLK；对于 UART-LIN，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 17-4 常用时钟频率下波特率计算

智能卡模式下，通信波特率由 UARTx_CODR 和 UARTx_SPBRG 寄存器共同决定。

UARTx_CODR 用于对 APBCLK 预分频得到 7816 接口时钟，并且这个时钟可以输出给外部卡片。

F7816=FAPBCLK/(CLKDIV+1)

7816 通信波特率使用 7816 接口时钟产生：

Baud ＝ F7816/(SPBRG ＋ 1)

智能卡模式下 SPBRG 最小可用值是 0x1，禁止配置 0x0

17.6.2 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。

17.7 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：

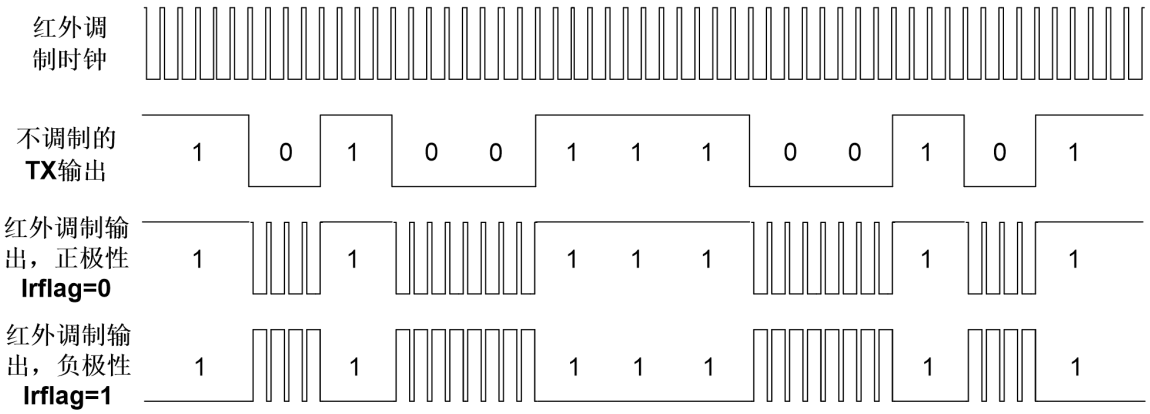


图 17-14 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

17.8 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。

17.9 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。

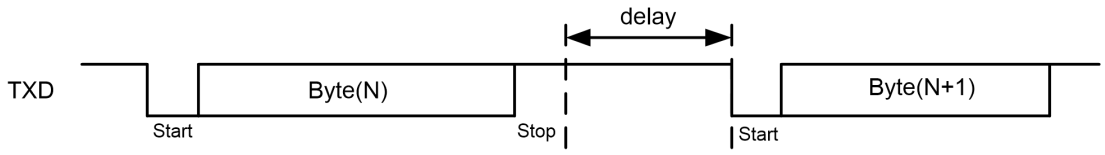


图 17-15 UART 发送延迟

17.10 寄存器

UART6模块基地址：0x4001 B800

UART7模块基地址：0x4001 BC00

UARTIR模块基地址：0x4001 7C00

offset 地址	名称	符号
UARTIR 寄存器（shared）		
0x00	红外调制寄存器 （Infrared modulation Control Register）	UART_IRCR
USARTx 寄存器		
0x00	USART 控制状态寄存器 （USART Control Status Register）	USART_CSR
0x04	USART 中断使能寄存器 （USART Interrupt Enable Register）	USART_IER
0x08	USART 中断标志寄存器 （USART Interrupt Status Register）	USART_ISR
0x0C	-	-
0x10	USART 接收缓冲寄存器 （USART Receive Buffer）	USART_RXBUF
0x14	USART 发送缓冲寄存器 （USART Transmit Buffer）	USART_TXBUF
0x18	USATR 波特率产生寄存器 （USART Baud rate Generator Register）	USART_BGR
0x20	USART 模式寄存器 （USART Mode Control Register）	USART_MCR
0x40	USART 智能卡状态标志寄存器 （USART Smart Card Interrupt Status Register）	USART_SCISR
0x44	USART 智能卡中断使能寄存器 （USART Smart Card Interrupt Enable Register）	USART_SCIER
0x48	USART 智能卡帧格式控制寄存器 （ USART Smart Card Frame Format Control Register）	USART_FFCR
0x4C	USART 智能卡额外保护时间寄存器 （USART Smart Card Extra Guard Time Register）	USART_EGTR
0x50	USART 智能卡输出时钟分频寄存器 （USART Smart Card Clock Output Register）	USART_CODR

17.10.1 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)

17.10.2 USARTx 模式寄存器（USARTx_MCR）

名称	USARTx_MCR							
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	-						-	MODE
位权限	U-0						U-0	R/W-0

位号	助记符	功能描述
31:1	-	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）

17.10.3 USARTx 控制状态寄存器（USARTx_CSR）

名称	USARTx_CSR(x=0~7)
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
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
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
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

位号	助记符	功能描述
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
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)

17.10.4 USARTx 中断使能寄存器（USARTx_IER）

名称	USARTx_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	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	-						TXBE_IE	TXSE_IE

位权限	U-0	R/W-0	R/W-0

位号	助记符	功能描述
31:17	-	**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**
11	**RXTO_IE**	接收超时中断使能，1 有效 (Receive Time-Out Interrupt Enable)
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	-	**未实现：读为 0**
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)

17.10.5 USARTx 中断标志寄存器（USARTx_ISR）

名称	USARTx_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	-					PERR	FERR	OERR
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	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	-	-				TX_OERR	TXBE	TXSE
位权限	U-0	U-0				R/W-0	R -0	R/W-0

位号	助记符	功能描述
31:19	-	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

位号	助记符	功能描述
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，软件读取 RXBUF 时清零 (Receive Buffer Full flag write 1 to clear)
7	-	未实现：读为 0
6:3	-	未实现：读为 0
2	**TX_OERR**	发送缓存溢出中断标志，硬件置位，软件写 1 清零 (Transmit Buffer Empty flag)
1	**TXBE**	发送缓存空中断标志，硬件置位，软件写发送缓存时清零 (Transmit buffer Empty flag)
0	**TXSE**	发送缓冲区空且发送移位寄存器空标志，硬件置位，待发送数据从 TXBUF 写入发送移位寄存器时清零 (Transmit Shift register Empty flag)

17.10.6 USARTx 超时和延迟寄存器（USARTx_TODR）

名称	**USARTx_TODR**							
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)

17.10.7 USARTx 接收缓冲寄存器（USARTx_RXBUF）

名称	**USARTx_RXBUF**							
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]

17.10.8 USARTx 发送缓冲寄存器（USARTx_TXBUF）

名称	**USARTx_TXBUF**							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							TXBUF[8]
位权限	U-0							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF[7:0]							
位权限	W-0000 0000							

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

7位收发时，发送的7bits数据写入TXBUF[6:0]

17.10.9 USATRx 波特率产生寄存器（USARTx_BGR）

名称	**USARTx_BGR**							
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)

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;

智能卡模式下，Baud ＝ F_{7816}/(SPBRG ＋ 1)，F_{7816}为智能卡输出工作时钟频率，由CODR寄存器配置

智能卡模式下SPBRG最小可用值是0x1，禁止配置0x0

17.10.10 USARTx 智能卡状态标志寄存器（USARTx_SCISR）

名称	USARTx_SCISR							
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）

Bit	助记符	功能描述
		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

17.10.11 USARTx 智能卡中断使能寄存器（USARTx_SCIER）

名称	USARTx_SCIER							
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

17.10.12 USARTx 智能卡帧格式控制寄存器（USARTx_FFCR）

名称	USARTx_FFCR							
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	-			DUMMY		ERSW		ERSGD
位权限	U-0			R/W-00		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。

Bit	助记符	功能描述
		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) 00：Even 01：Odd 10：Always 1 11：不校验
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

17.10.13 USARTx 智能卡额外保护时间寄存器（USARTx_EGTR）

名称	USARTx_EGTR							
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)

17.10.14 USARTx 智能卡输出时钟分频寄存器（USARTx_CODR）

名称	USARTx_CODR (x=6~7)							
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。

17.10.15 USARTx 智能卡控制寄存器（USARTx_SCCR）

名称	USARTx_SCCR							
offset	0x54							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							AWFR
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	AWFR	撤销等待自动重发 (Abort Waiting For Re-transmit) 当接收数据校验错误，回发 error signal 之后，U7816 进入等待对方重发状态，此时无法发送数据。软件置位 AWFR 可以撤销等待，回到正常收发流程中。 软件写 1 撤销等待，撤销成功后硬件自动清零。

18 低功耗 UART（LPUART）

18.1 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 模式下工作。

特点：

- 异步数据收发
- 1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
- 可编程数据极性
- 当工作时钟为XTLF时，支持Sleep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模式下不支持）

18.2 结构框图

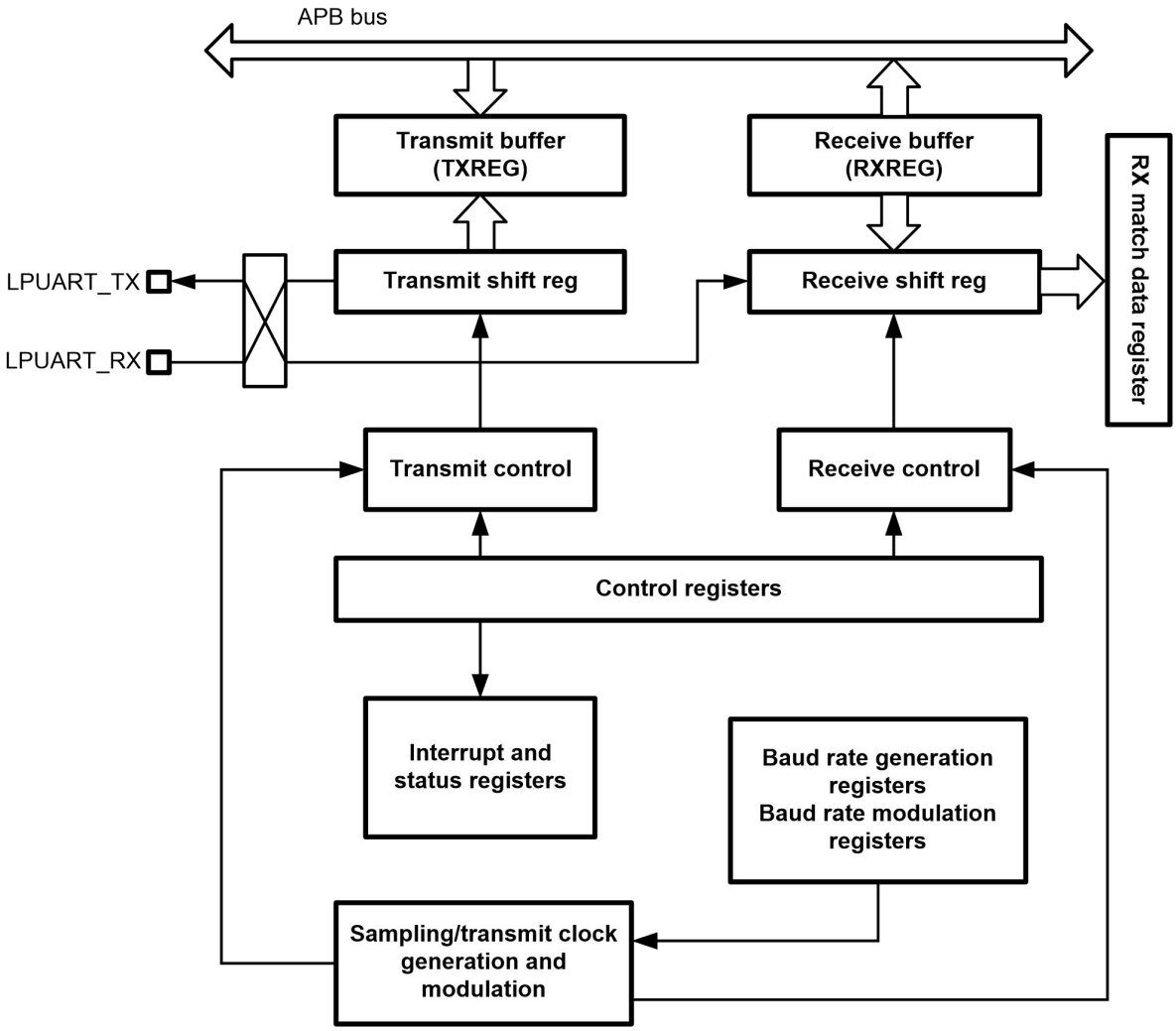


图 18-1 LPUART 结构框图

18.3 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。

18.4 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。

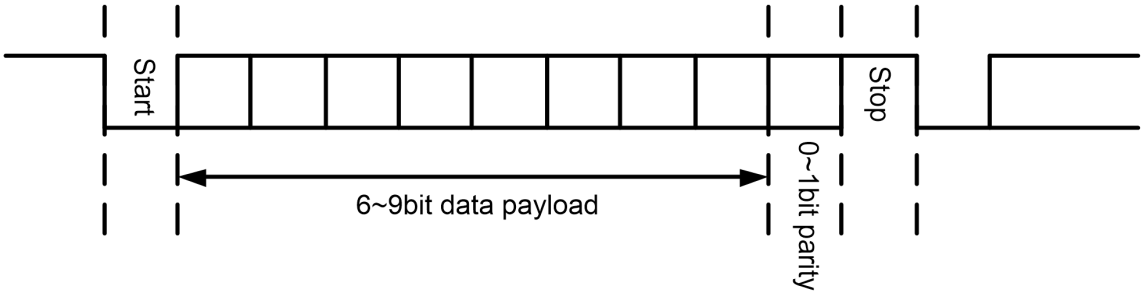


图 18-2 字符描述

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 18-1LPUART 数据帧格式

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

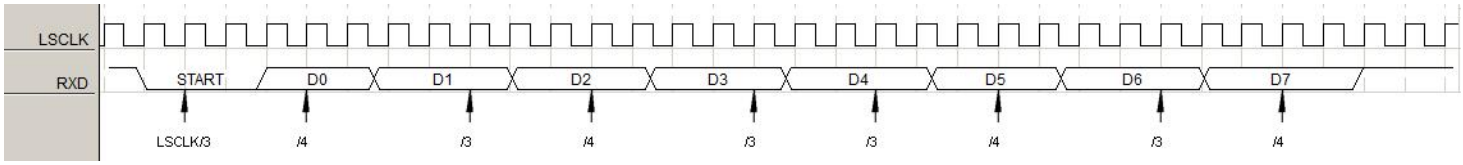
注意 PDSEL 寄存器用于配置帧的数据长度，通信帧长为【起始位+数据位+校验位+停止位】。

18.5 功能描述

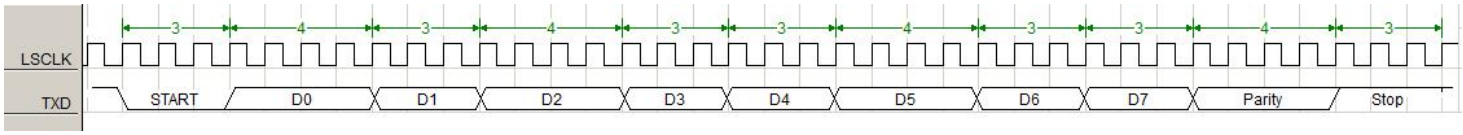
18.5.1 位接收采样和发送

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 18-2 LPUART 数据位调制系数

18.5.2 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

18.5.3 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

18.5.4 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 0 或 3 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

18.5.5 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep 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

18.5.6 LPRUN 模式下的数据 DMA 收发

通过 LPUART 和 DMA，软件可以实现 LPRUN 模式下一定数据量的 LPUART 自动收发，而无需 CPU 干预，同时保证典型条件下全芯片功耗小于 10uA。

- 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DMA通道控制寄存器，选择LPUART收发
- 如果需要发送数据，将待发数据写入RAM中指定位置
- 配置DMA数据收发长度和RAM指针
- 将系统主时钟选为LSCLK
- 软件进入LPRUN
- 配置LPUEN寄存器打开发送接收使能
- 如CPU无额外工作，可以主动进入WFI/WFE，等待中断唤醒

18.5.7 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 18-3 LPUART DMA 中断说明

18.6 寄存器

LPUART1模块基地址：0x40010400

offset 地址	名称	符号
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 Buffer Register）	LPUART1_RXBUF
0x14	LPUART1 发送缓冲寄存器（LPUART1 Transmit Buffer Register）	LPUART1_TXBUF
0x18	LPUART1 数据匹配寄存器（LPUART1 data Matching Register）	LPUART1_DMR

18.6.1 LPUARTx 控制状态寄存器（LPUARTx_CSR）

名称	LPUARTx _CSR(x=0)							
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)

18.6.2 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)

18.6.3 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)

18.6.4 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	MCTLEN	-			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	**MCTLEN**	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

18.6.5 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)

18.6.6 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)

18.6.7 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)

19 串行外设接口（SPI/I2S）

19.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过 4 线交换数据的串行同步通讯手段。SPI 接口模块可配置为主设备或从设备，实现与外部的 SPI 通信。

I2S 同样是一个同步串行通信外设，能够以主机或从机模式进行半双工通信，可以支持飞利浦 I2S 标准，MSB-LSB- justified，或 PCM 标准。

本模块可以配置为 SPI 或 I2S 功能。

SPI 特性：

- 全双工4线串行同步收发（SCLK, MOSI, MISO, SSN）
- MISO和MOSI可交换引脚顺序
- 半双工4线串行同步收发（SCLK, SDATA, SSN, DCN），用于TFT屏驱动
- 4路独立通道
- 主从模式
- 可编程时钟极性和相位
- 可编程比特速率
- 可编程数据字长（8/16/24/32bits）
- 最大波特率为$F_{APBCLK}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持DMA

I2S特性：

- 支持主机和从机模式，半双工通信
- 8bit 时钟预分频电路
- 支持 16/24/32bit 数据格式
- 音频通道数据帧长度为 16bit 或 32bit
- 可编程时钟极性
- 支持 I2S 协议：
  - ✓ Philips I2S 标准

- ✓ MSB-justified 标准
- ✓ LSB-justified 标准
- ✓ PCM 标准

- 支持 DMA
- 支持主机时钟输出（256×Fs，Fs 为音频采样频率）

19.2 SPI 功能描述

19.2.1 SPI 结构框图

下图为 SPI 模块的结构示意图。

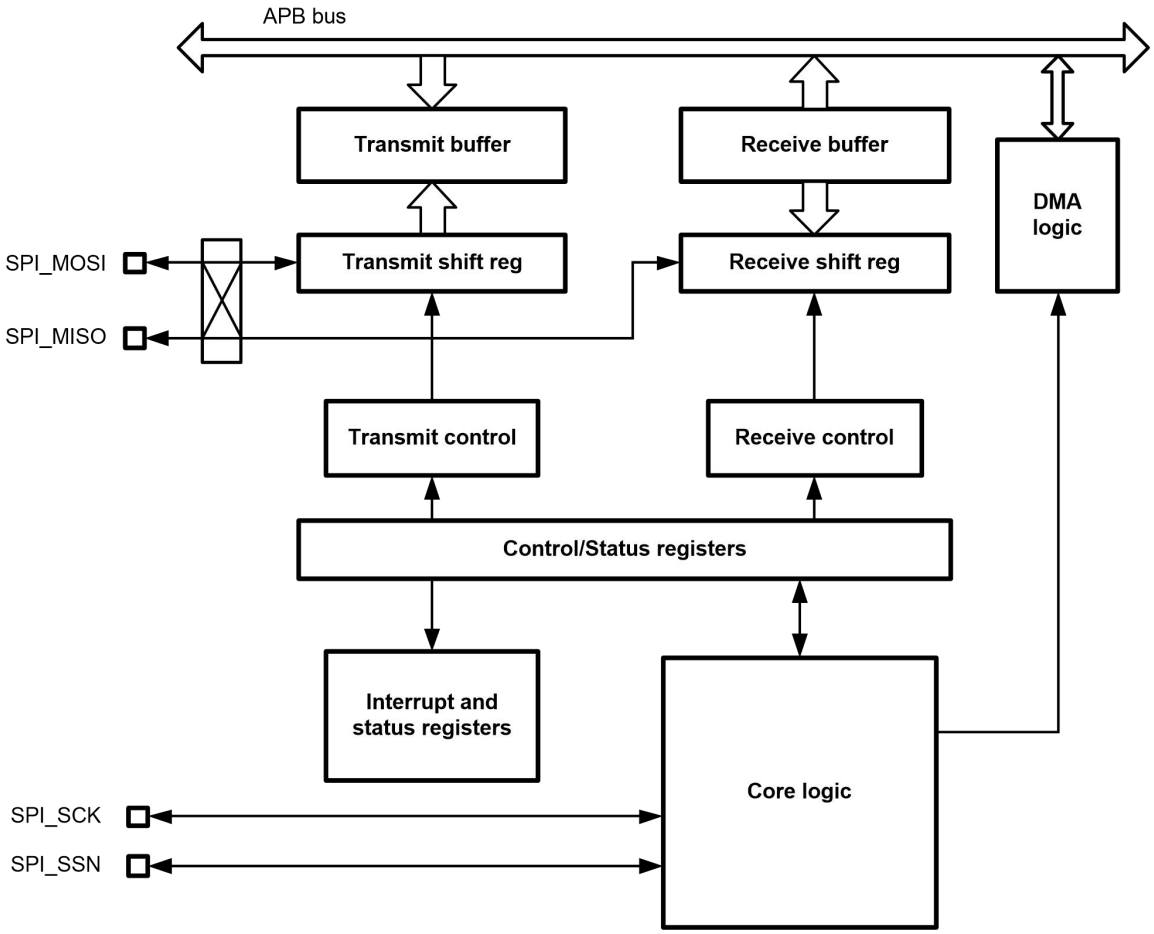


图 19-1 SPI 结构框图

19.2.2 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
	SPI0	SSN	片选信号	SSN	片选信号
		SCLK	时钟	SCLK	时钟
		MISO	主机输入从机输出	DCN	命令/数据标识
		MOSI	主机输出从机输入	SDATA	双向数据
	SPI1	SSN	片选信号	SSN	片选信号
		SCLK	时钟	SCLK	时钟
		MISO	主机输入从机输出	DCN	命令/数据标识
		MOSI	主机输出从机输入	SDATA	双向数据

表 19-1SPI 引脚列表

19.2.3 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。

19.2.3.1 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。

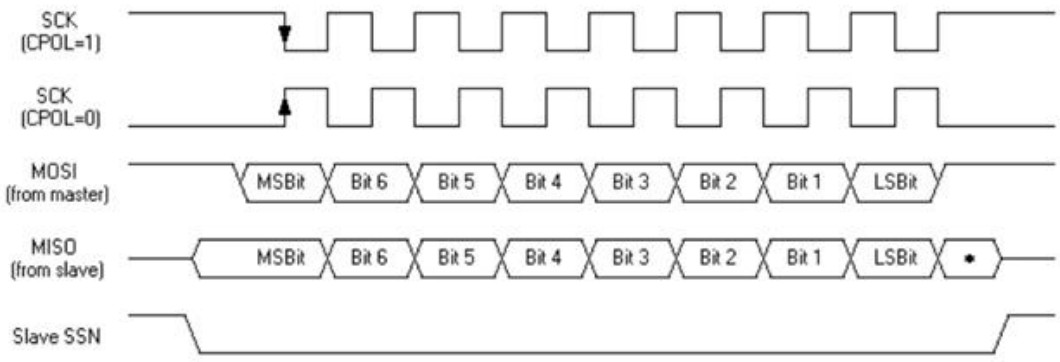


图 19-2 SPI 数据/时钟时序图（CPHA=0）

19.2.3.2 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。

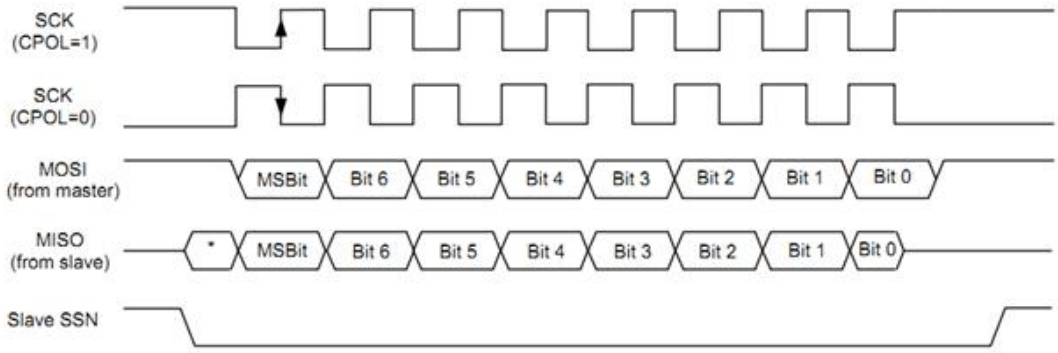


图 19-3 SPI 数据/时钟时序图（CPHA=1）

19.2.3.3 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。FM33FH0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。

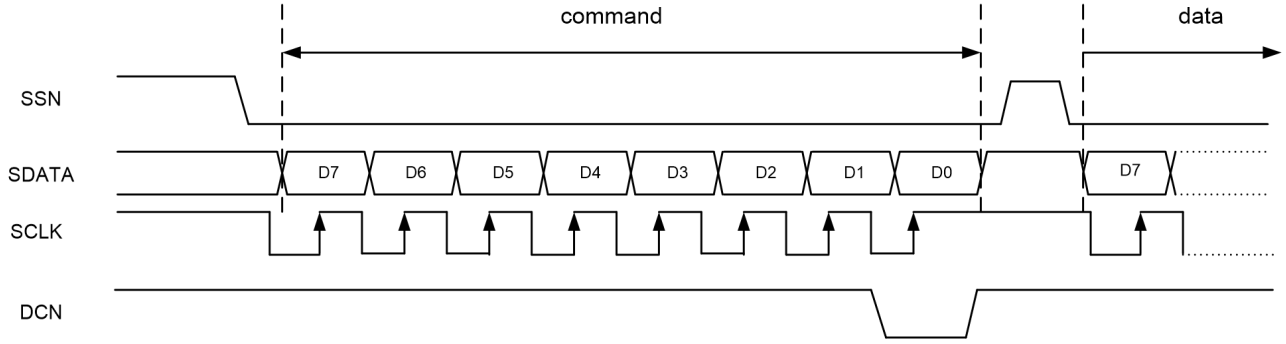


图 19-44 线半双工写操作

DCN在第8个时钟上升沿采样判决，如果为0，表示当前帧是命令帧。发送命令帧前，软件需要将DCN

寄存器写0，命令帧发送完成后硬件自动将DCN寄存器置位。

4线半双工读操作

软件通过置位HD_RW寄存器，表示当前主机要发起读操作。

4线半双工读操作支持8位、24位和32位读取。主机发起读操作时，首先发送读命令帧。当读命令帧发送完毕后，可以根据寄存器配置发送1个dummy cycle，在dummy cycle期间，SCLK时钟正常发送，但是主机不驱动SDATA，也不接受SDATA输入。

完成命令帧和dummy cycle（可选）后，4线半双工SPI自动进入接收状态，SDATA信号改由从机驱动，主机收到的数据帧将被写入接收缓冲区。每个数据帧接收完成后，将置位RXBF中断标志寄存器。软件应及时读取接收缓冲区中的数据，如果接收缓冲区和接收移位寄存器都处于满状态，硬件会停止SCLK发送，暂停从从机读取数据，直到软件或DMA读取了接收缓冲区。

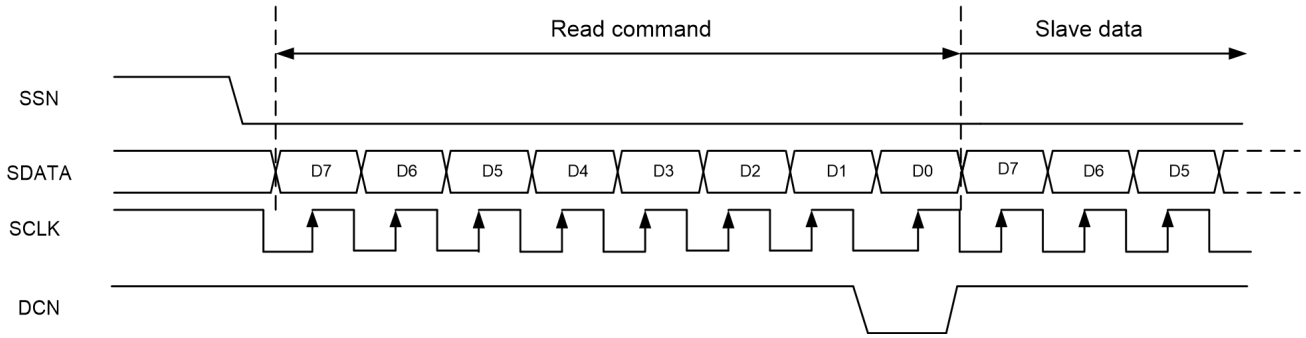


图 19-5 4 线半双工读操作（无 dummy cycle）

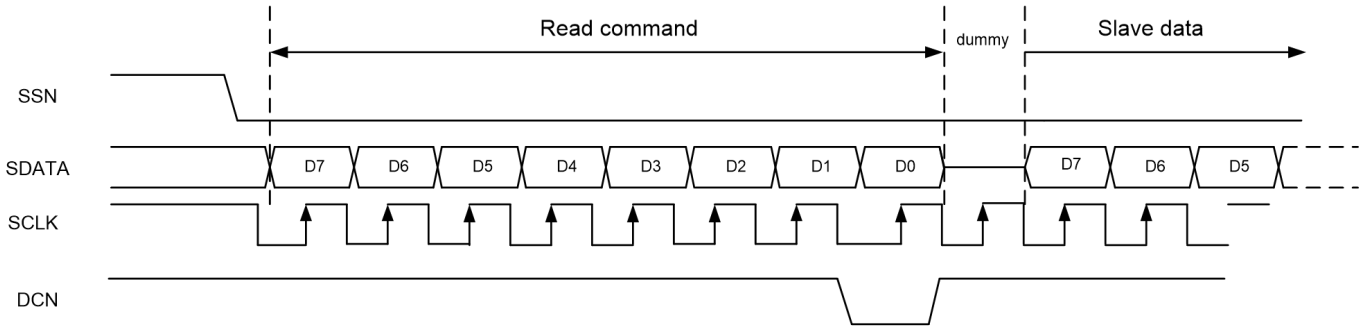


图 19-6 4 线半双工读操作（有 dummy cycle）

19.2.4 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 19-2 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**错误!未找到引用源。**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。

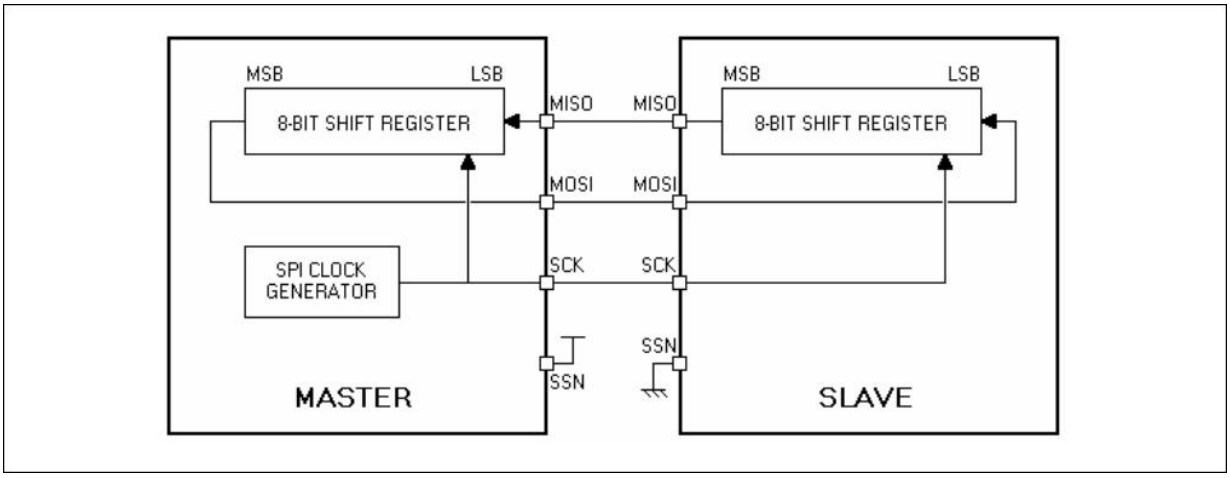


图 19-7 SPI Master/SPI Slave 互连

19.2.5 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19.2.9数据冲突。

发送移位寄存器中的数据被完全发送出去后，发送移位寄存器空标志TXSE置位，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19.2.9数据冲突。

BUSY标志

当SPI正在进行数据收发时，BUSY寄存器置位。此寄存器在某些场景下可以用来判断最后一帧数据是否传输完毕。比如TXBE只是表示数据已经进入移位发送，但是真正发送完成，需要等待BUSY标志清零。

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。

19.2.6 TX-ONLY 模式

某些时候SPI通信是半双工的，在主机仅需进行发送的情况下，通过置位TXO寄存器进入TX-ONLY模式，此时MISO收到的数据不会被写入RX Buffer中，相应的也不会置位RXBF中断标志。

通过置位TXO_AC，可以实现TXO自动清零功能。在TX-ONLY模式下，如果TX buffer空（TXBE置位）并且发送移位寄存器空，则TXO寄存器自动清零，退出TX-ONLY状态。

19.2.7 RX-ONLY 模式

SPI主机仅需进行接收的情况下，通过置位RXO寄存器进入RX-ONLY模式，此时SPI模块无需软件对TX Buffer进行写操作，即可进行连续不断的数据接收，此时MOSI将保持IDLE电平，并且不会置位TXBE中断标志寄存器。

19.2.8 主机 SSN 控制

SPI模块主机支持硬件或软件控制SSN信号。

当SSNSEN寄存器清零时，SSN由硬件电路控制；如果SSNM寄存器置位，则SPI每发完一帧数据后，将拉高SSN，SSN高电平时间由WAIT寄存器配置（若干个SCK时钟周期）；

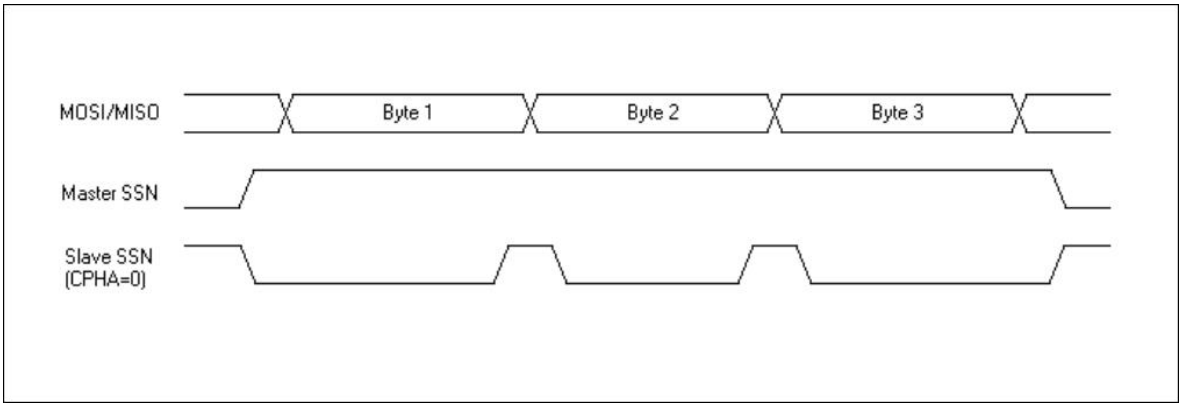


图 19-8 SPI SSN 时序图（SSNM=1，CPHA=0）

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。

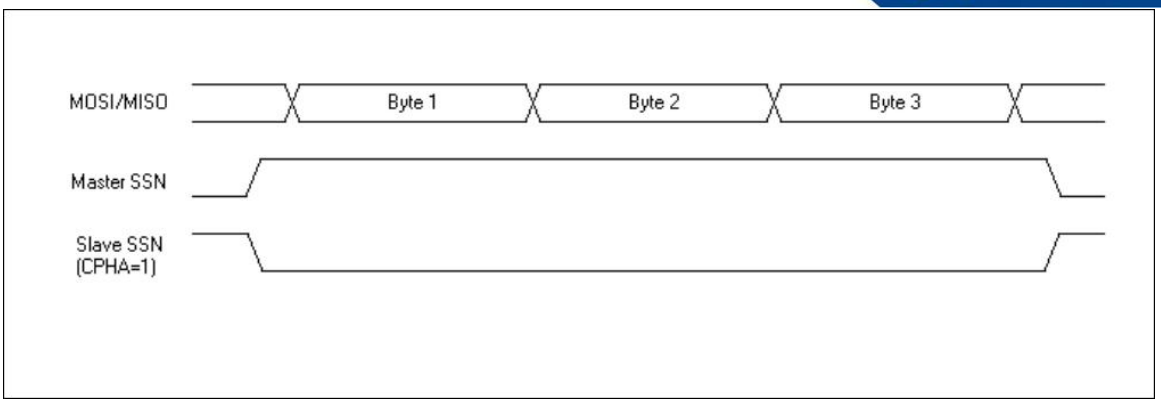


图 19-9 SPI SSN 时序图（SSNM=0）

当SSNSEN寄存器置位时，SSN直接由软件控制。软件通过写SPIxCR2.SSN寄存器位，可以直接操作SPI主机发送的SSN电平。

19.2.9 数据冲突

当 SPI 的 TX Buffer 数据尚未被读进移位寄存器，或者 SPI 的 RX Buffer 中的数据未被软件或 DMA 读取时，对 TX Buffer 或 RX Buffer 的写操作会产生对应的冲突错误，TXCOL/RXCOL 位会置起，产生中断。导致冲突的写入数据将被忽略。数据冲突错误在主从模式下都会产生。

对 TX Buffer 的写操作，由芯片内部的 Master 模块发起，包括 CPU、DMA 等等。对 RX Buffer 的写操作，则由外部 SPI 器件发起。

当数据冲突发生时，TX Buffer 和 RX Buffer 内原有数据不会被刷新，新写入的数据丢失。

19.2.10 使用 DMA 进行 SPI 收发

当 SPI 模块被使能后，SPI 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 SPI 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 SPI 请求，并完成 RAM 和 SPI 之间的数据搬运。

注意：如果使用 DMA 进行收发全双工通信，软件应先使能 DMA 发送通道，再使能 DMA 的接收通道；反之可能会导致 SPI 额外发送一个字节的 dummy 数据。

使用 DMA 进行 SPI 接收

- 将 DMA 通道 3 或 5 配置为 SPI_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道

- 配置 SPI 模块参数
- 使能 SPI 模块，等待数据接收
- 收到数据后 SPI 自动产生 DMA 请求
- DMA 响应请求，读取 SPI 接收缓存寄存器，写入指定 RAM 地址
- 当指定长度的 DMA 传输结束后，DMA 将忽略后续请求并产生传输完成中断，软件应处理中断并关闭 SPI
- 如果关闭 SPI 前又有数据被接收，软件可以通过写 RXBFC 清除 RXBUF

使用 DMA 进行 SPI 发送

DMA发送过程与上述接收过程类似，主要差别是，当指定长度的DMA传输结束后，软件不能立即关闭SPI，因为此时最后一帧数据还在移位发送中，因此软件需要查询BUSY标志直到移位发送结束，再关闭SPI模块。

数据帧长度与 RAM 数据组织方式

SPI 传输帧长度可以配置为 8、16、24、32bit。

当数据帧长度为 8bit 时，DMA 每次搬运 1byte，4 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data3，data2，data1，data0 }

当数据帧长度为 16bit 时，DMA 每次搬运 2bytes，2 次搬运填满 RAM 一个地址，字内采用小端存储：

RAM word: { data1，data0 }

当数据帧长度为 24bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址，但是有效数据仅占用 RAM 字内低 24bit：

RAM word: { 8'h0，data0 }

当数据帧长度为 32bit 时，DMA 每次搬运 1 word，1 次搬运填满 RAM 一个地址：

RAM word: { data0 }

19.3 I2S 功能描述

19.3.1 I2S 结构框图

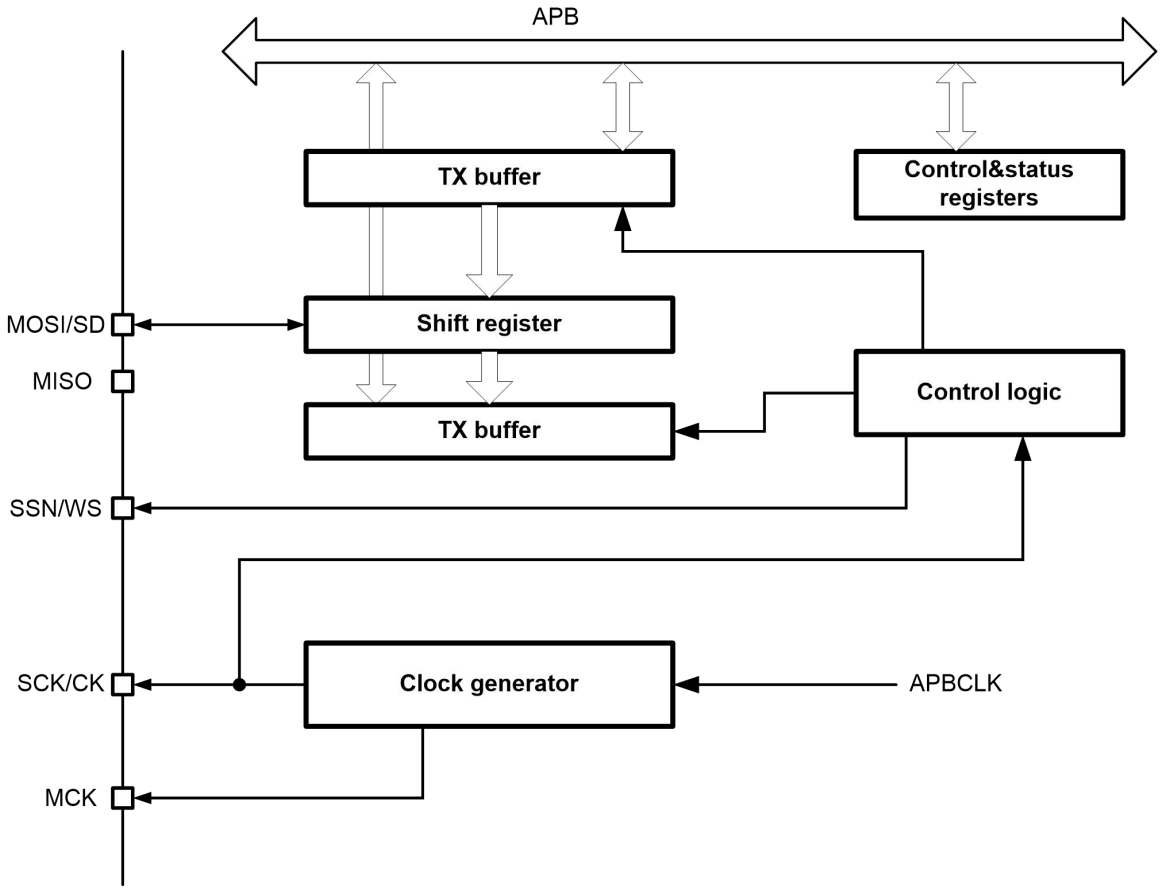


图 19-10I2S 结构框图

19.3.2 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

19.3.3 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。

19.3.3.1 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：

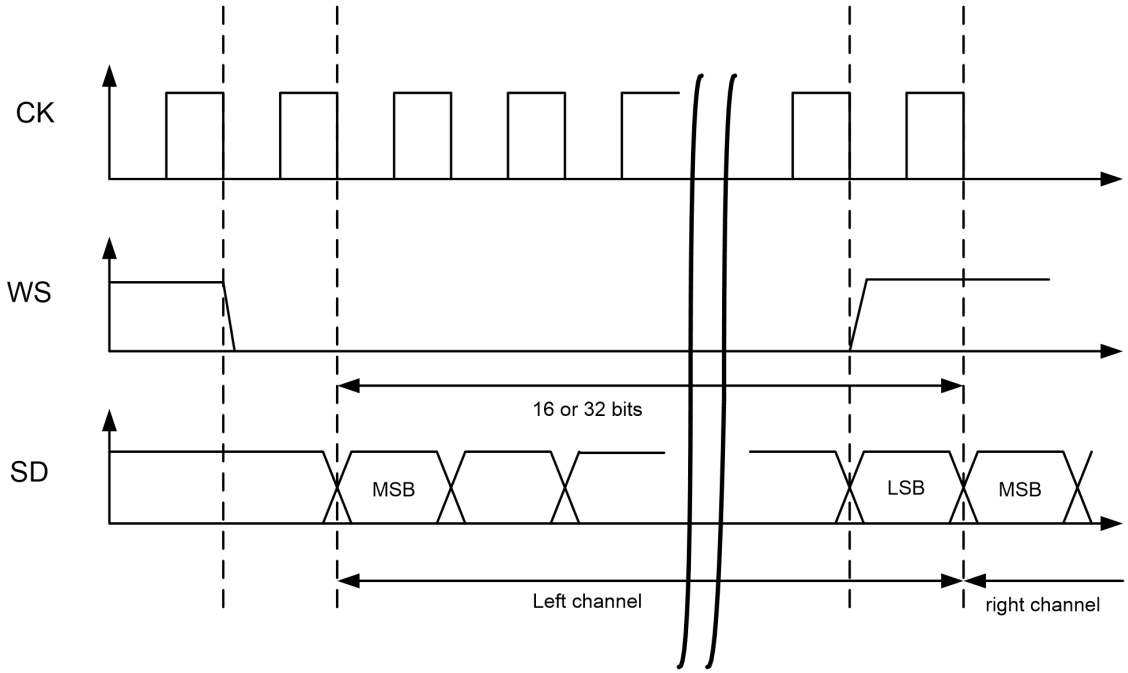


图 19-11Philips 标准时序

19.3.3.2 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。

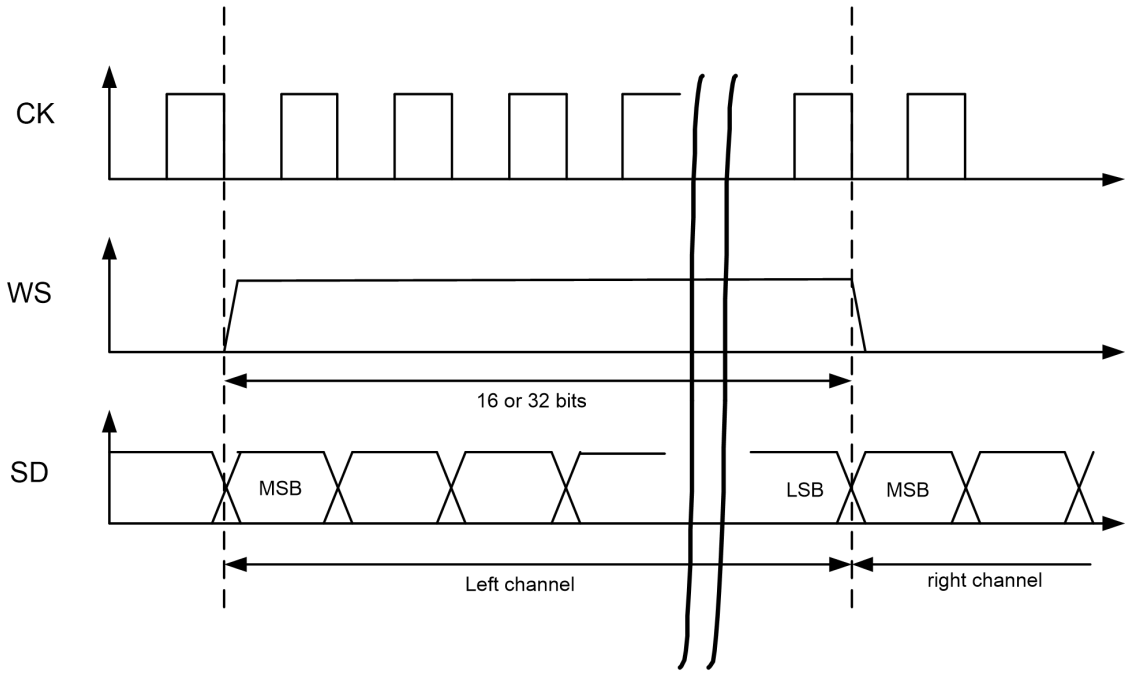


图 19-1232/16-bits MSB-justified 时序

在 32 位数据帧中传输 24 位数据时，MSB justified 格式将有效数据先发，最后补 0：

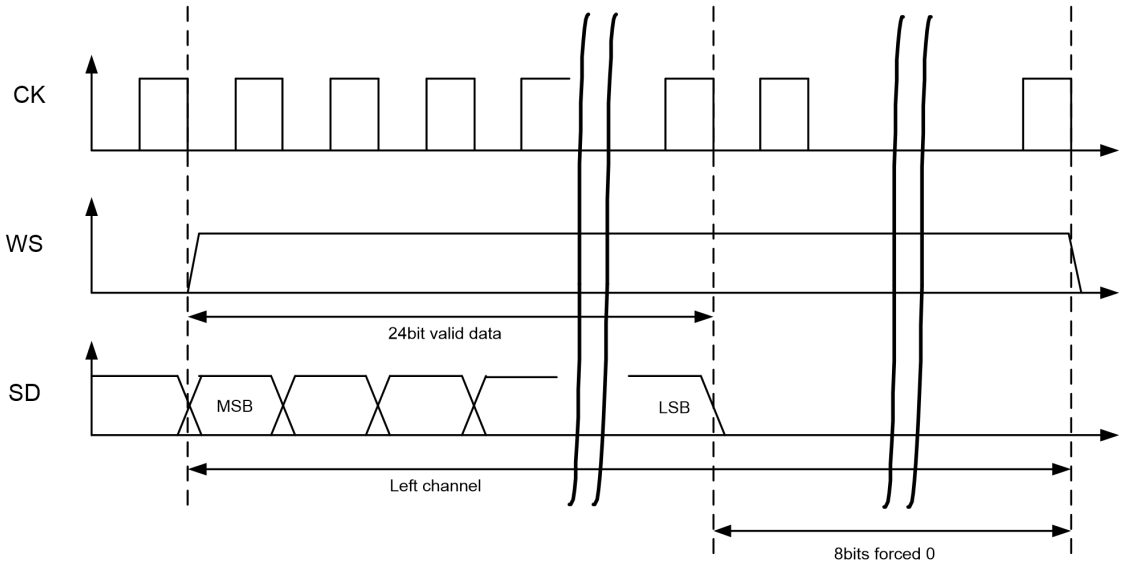


图 19-1324-bits MSB-justified 时序

在 32 位数据帧中传输 16 位数据时，MSB justified 格式同样是将有效数据先发，最后补 0。

19.3.3.3 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：

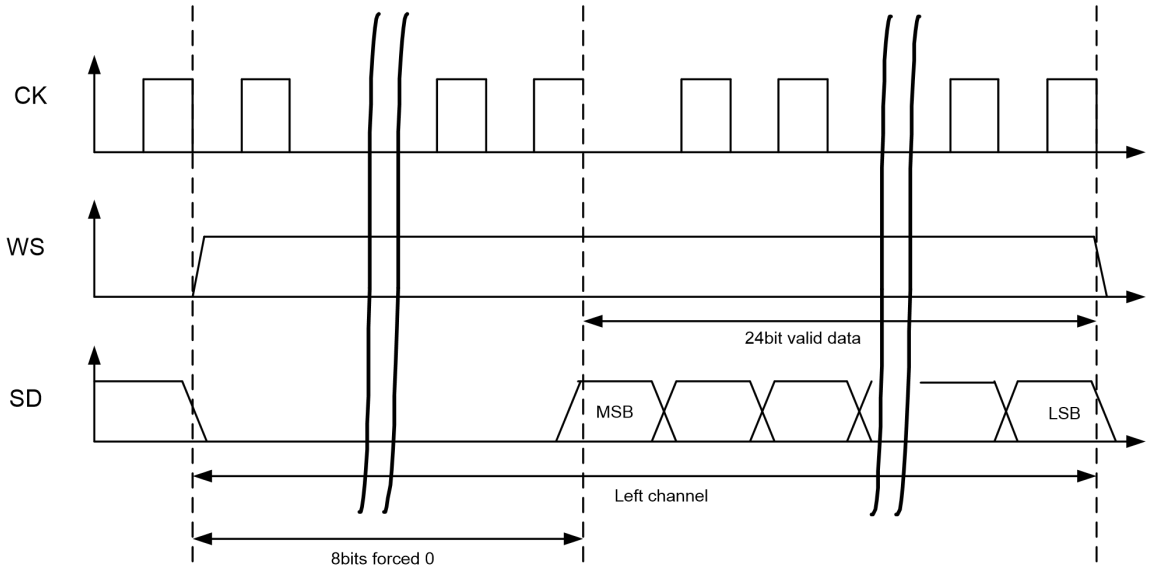


图 19-14LSB-justified 时序

19.3.3.4 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。

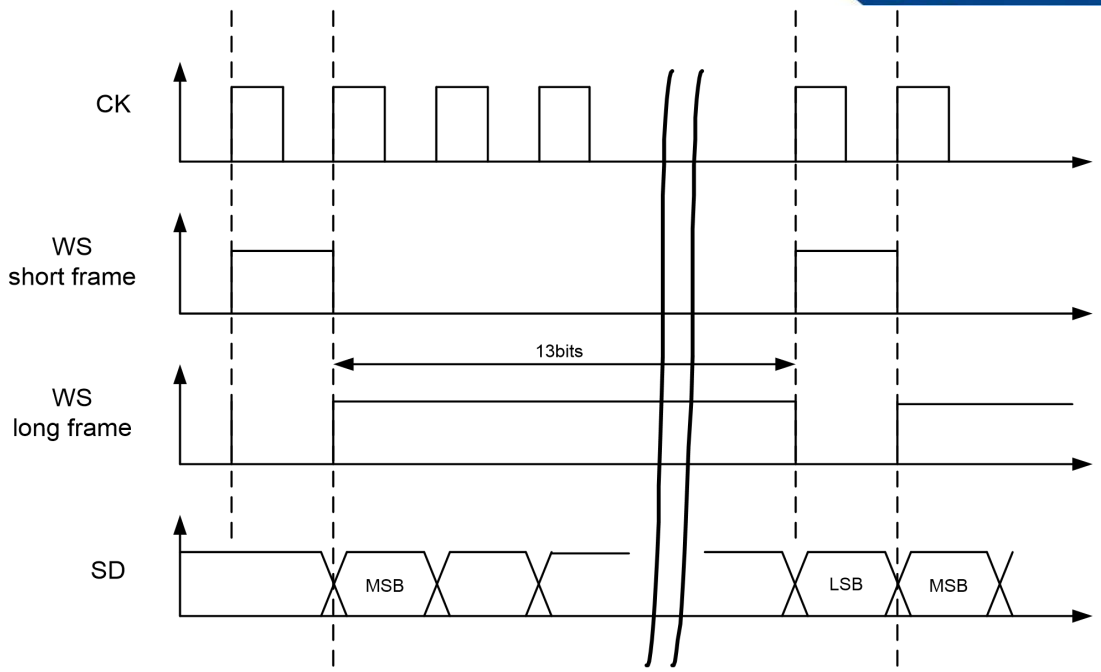


图 19-15PCM 标准时序

19.3.4 时钟产生

I2S 数据码率由下式决定：

I2S 码率 = 每声道数据帧位数×声道数×音频采样频率

假设采用 16 位音频帧格式，左右声道，音频采样率 F_S，则码率计算为：

I2S 码率 = 16×2×F_S

假设采用 32 位音频帧格式，左右声道，，音频采样率 F_S，则码率计算为：

I2S 码率 = 32×2×F_S

音频采样的频率包括：192Khz、96Khz、48Khz、44.1Khz、32Khz、22.05Khz、16Khz、11.025Khz、8Khz 等。为了获得以上目标频率，按照以下公式计算分频寄存器的值（CLKDIV 和 ODD 都是分频寄存器）：

如果输出 MCK：F_S = APBCLK / [(256)×((2×CLKDIV) + ODD)]，

如果不输出 MCK：F_S = APBCLK / [32×(CHLEN+1)×((2×CLKDIV) + ODD)]，

CHLEN=0：声道帧长 16bit

CHLEN=1：声道帧长 32bit

CLKDIV 寄存器定义了 50%占空比下时钟 CK 的高低电平宽度，ODD 寄存器用于调整高电平宽度，从而实现理论上的小数分频效果。

I2S 主机模式下可以通过 MCKOE 寄存器选择输出或者不输出 MCK，MCK 频率是 256×Fs

假设 APBCLK 为 48MHz，下表罗列了针对不同 Fs 的参考配置和实际采样频率误差。

APBCLK (MHz)	CHLEN	CLKDIV	ODD	MCK	Target Fs (Hz)	Actual Fs	Error
48	16	8	0	No	96000	93750	2.3438%
48	32	4	0	No	96000	93750	2.3438%
48	16	15	1	No	48000	48387.0968	0.8065%
48	32	8	0	No	48000	46875	2.3438%
48	16	17	0	No	44100	44117.6471	0.0400%
48	32	8	1	No	44100	44117.6471	0.0400%
48	16	23	1	No	32000	31914.8936	0.2660%
48	32	11	1	No	32000	32608.6957	1.9022%
48	16	34	0	No	22050	22058.8235	0.0400%
48	32	17	0	No	22050	22058.8235	0.0400%
48	16	47	0	No	16000	15957.4468	0.2660%
48	32	23	1	No	16000	15957.4468	0.2660%
48	16	68	0	No	11025	11029.4118	0.0400%
48	32	34	0	No	11025	11029.4118	0.0400%
48	16	94	0	No	8000	7978.7234	0.2660%
48	32	47	0	No	8000	7978.7234	0.2660%
48	16	2	0	Yes	48000	46875	2.3438%
48	32	2	0	Yes	48000	46875	2.3438%
48	16	2	0	Yes	44100	46875	6.2925%
48	32	2	0	Yes	44100	46875	6.2925%
48	16	3	0	Yes	32000	31250	2.3438%
48	32	3	0	Yes	32000	31250	2.3438%
48	16	4	1	Yes	22050	20833.3333	5.5178%
48	32	4	1	Yes	22050	20833.3333	5.5178%
48	16	6	0	Yes	16000	15625	2.3438%
48	32	6	0	Yes	16000	15625	2.3438%
48	16	8	1	Yes	11025	11029.4118	0.0400%
48	32	8	1	Yes	11025	11029.4118	0.0400%
48	16	11	1	Yes	8000	8152.17391	1.9022%
48	32	11	1	Yes	8000	8152.17391	1.9022%

表 19-2I2S 典型采样频率计算表

19.3.5 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。

19.3.5.1 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

19.3.5.2 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。

19.3.5.3 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。

19.3.6 I2S 从机功能

19.3.6.1 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

19.3.6.2 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。

19.3.6.3 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。

19.3.7 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。

19.3.8 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。

19.4 寄存器

SPI0模块基地址：0x40019000

SPI1模块基地址：0x40018C00

SPI2模块基地址：0x40010800

SPI3模块基地址：0x40014000

offset 地址	名称	符号
SPIx 寄存器 (x=0,1,2,3)		
0x00	SPIx 控制寄存器 1 （SPIx Control Register1）	SPIx_CR1
0x04	SPIx 控制寄存器 2 （SPIx Control Register2）	SPIx_CR2
0x08	SPIx 控制寄存器 3 （SPIx Control Register3）	SPIx_CR3
0x0C	SPIx 中断使能寄存器 （SPIx Interrupt Enable Register）	SPIx_IER
0x10	SPIx 中断状态寄存器 （SPIx Status Register）	SPIx_ISR
0x14	SPIx 发送数据缓冲寄存器 （SPIx Transmit Buffer）	SPIx_TXBUF
0x18	SPIx 接收数据缓冲寄存器 （SPIx Receive Buffer）	SPIx_RXBUF
0x20	I2S 配置寄存器 （SPIx I2S Control Register）	SPIx_I2SCR
0x24	I2S 预分频寄存器 （SPIx I2S Prescaler Register）	SPIx_I2SPR

19.4.1 SPI 控制寄存器 1（SPIx_CR1）

名称	SPIx_CR1 (x=0,1,2,3)							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				IOSWAP	MSPA	SSPA	MM
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	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)

Bit	助记符	功能描述
		0：默认引脚顺序 1：交换引脚顺序
10	**MSPA**	Master Sampling Position Adjustment，Master 对 MISO 信号的采样位置调整，用于高速通信时补偿 PCB 走线延迟 1：采样点延迟半个 SCK 周期 0：不调整
9	**SSPA**	Slave Sending Position Adjustment，Slave MISO 发送位置调整 1：提前半个 SCK 周期发送 0：不调整
8	**MM**	Master/Slave 模式选择。 1：Master 模式 0：Slave 模式
7:6	**WAIT**	Master 模式下，每发送完一帧后加入至少(1+WAIT)个 SCK cycle 等待时间，再传输下一帧的数据。如果 SSN 由硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。
5:3	**BAUD**	Master 模式波特率配置位: 000： $f_{APBCLK}/2$ 001： $f_{APBCLK}/4$ 010： $f_{APBCLK}/8$ 011： $f_{APBCLK}/16$ 100： $f_{APBCLK}/32$ 101： $f_{APBCLK}/64$ 110： $f_{APBCLK}/128$ 111： $f_{APBCLK}/256$ 当通信正在进行的时候，不能修改这些位。
2	**LSBF**	帧格式 (LSB First) 0：先发送 MSB 1：先发送 LSB 注：当通信在进行时不能改变该位的值。
1	**CPHOL**	时钟极性选择 (Clock Polarity) 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
0	**CPHA**	时钟相位选择 (Clock Phase) 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

19.4.2 SPI 控制寄存器 2（SPIx_CR2）

名称	SPIx_CR2 (x=0,1,2,3)							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN	DLEN2			RXO	DLEN		HALFDUPLEX
位权限	R/W-0	R/W-0			R/W-0	R/W-00		R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15	**DUMMY_EN**	4 线半双工协议下是否在读操作中插入 dummy cycle (Dummy cycle Enable) 0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	**DLEN2**	通信数据字长扩展配置寄存器，与 DLEN 组合形成最终配置 {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) 参见 DLEN2

Bit	助记符	功能描述
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

19.4.3 SPI 控制寄存器 3（SPIx_CR3）

名称	SPIx_CR3 (x=0,1,2,3)							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TXBFC	RXBFC	MERRC	SERRC
位权限	U-0				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

19.4.4 SPI/I2S 中断控制寄存器（SPIx_IER）

名称	SPIx_IER (x=0,1,2,3)							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SSNRIE	SSNFIE	TXSEIE	ERRIE	TXBEIE	RXIE
位权限	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5	SSNRIE	从机 SSN 上升沿中断使能
4	SSNFIE	从机 SSN 下降沿中断使能
3	TXSEIE	发送移位寄存器空中断使能(Transmit shift register empty interrupt enable)
2	ERRIE	SPI 错误中断使能 (Error Interrupt Enable)
1	TXBEIE	发送缓冲空中断使能 (Transmit buffer empty Interrupt Enable)
0	RXIE	接收完成中断使能 (Receive Interrupt Enable)

19.4.5 SPI/I2S 中断标志寄存器（SPIx_ISR）

名称	SPIx_ISR (x=0,1,2,3)							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DCN_TX	-	RXCOL	TXCOL	BUSY

位权限	U-0			R/W/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 0：发送移位寄存器非空

Bit	助记符	功能描述
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

19.4.6 SPI/I2S 发送缓存寄存器（SPIx_TXBUF）

名称	SPIx_TXBUF (x=0,1,2,3)							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF							
位权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TXBUF							
位权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXBUF							
位权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
位权限	W-00000000							

Bit	助记符	功能描述
31:0	**TXBUF**	SPI 发送缓存(Transmit Buffer)

19.4.7 SPI/I2S 发送缓存寄存器（SPIx_RXBUF）

名称	SPIx_RXBUF (x=0,1,2,3)							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF							
位权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RXBUF							
位权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RXBUF							
位权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
位权限	R-00000000							

Bit	助记符	功能描述
31:0	**RXBUF**	SPI 接收缓存(Receive Buffer)

19.4.8 I2S 配置寄存器（SPIx_I2SCR）

名称	SPIx_I2SCR (x=0,1,2,3)							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-				I2SSTD		CHLEN	CKPOL
位权限	U-0				R/W-00		R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DLEN		PCMSYNC	I2SMM		DLYEN	I2SMOD	I2SEN
位权限	R/W-00		R/W-0	R/W-0		R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:12	--	RFU：未实现，读为 0
11:10	**I2SSTD**	I2S 标准选择 00：Philips 标准 01：MSB-justified 10：LSB-justified 11：PCM 标准
9	**CHLEN**	声道长度（每个声道的 bit 数） 0：16bit 1：32bit 仅 DLEN=00 的情况下有效，否则 CHLEN 固定为 32bit
8	**CKPOL**	I2S 空闲时钟极性 0：CK 空闲时保持低电平 1：CK 空闲时保持高电平
7:6	**DLEN**	I2S 通信数据长度 00：16bit 01：24bit 10：32bit 11：RFU
5	**PCMSYNC**	PCM 帧同步 0：短帧同步 1：长帧同步 仅 I2SSTD=11 时有效
4:3	**I2SMM**	I2S Master Mode 00：从机接收 01：从机发送 10：主机接收 11：主机发送
2	**DLYEN**	WS 和 SD 数据延迟使能，仅 I2S 主机模式有效 0：关闭输出延迟 1：WS 和 SD 输出经过 100ns 模拟延迟

Bit	助记符	功能描述
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

19.4.9 I2S 预分频寄存器（SPIx_I2SPR）

名称	SPIx_I2SPR (x=0,1,2,3)							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						MCKOE	ODD
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CLKDIV							
位权限	R/W-0000 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)

20 QuadSPI

20.1 概述

QuadSPI是一个高速串行接口，可以使用最多6根信号线与外部QSPI存储器通信。QuadSPI接口支持传统SPI模式（4线半双工，不支持全双工）、Dual-SPI模式和Quad-SPI模式。通过QuadSPI接口，MCU可以有效扩展片内存储器，并且能够通过eXecution-In-Place（XIP）方式，直接从片外存储器取指令运行。

QuadSPI模块主要特点如下：

- QSPI主机
- 支持三种工作模式：外设模式、自动查询模式、存储器映射模式
- 灵活可编程的命令字节（instruction）
- 灵活可编程的帧格式
- 集成数据16字节FIFO
- 外设模式支持DMA访问

20.2 结构框图

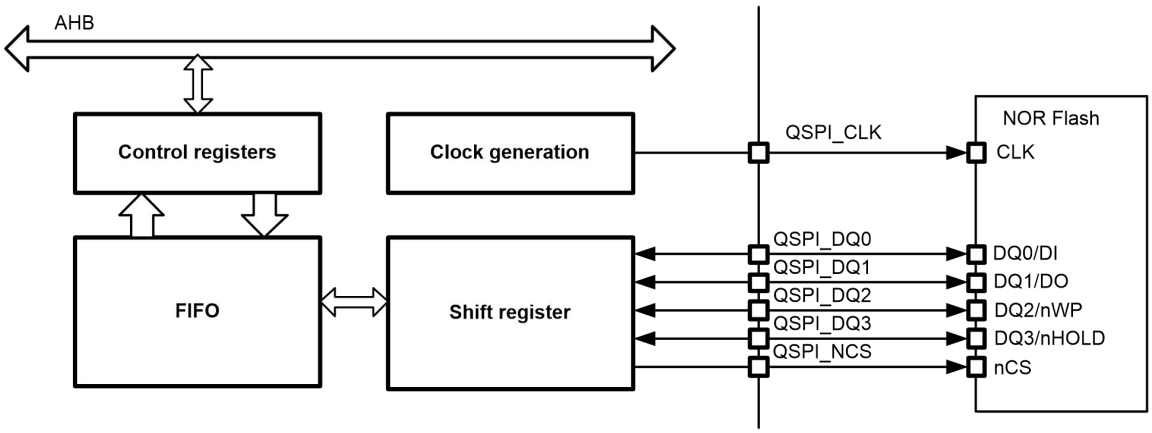


图 20-1 QSPI 结构框图

20.3 功能描述

20.3.1 工作时钟

QuadSPI模块工作时钟来自于AHBCLK，其对外输出的总线时钟可以是AHBCLK及其分频。用8bit的PRESCALER寄存器可以定义QSPI总线时钟相对于AHBCLK的分频比，支持2~256之间的任意分频，奇数分频时，占空比不是50%。

QSPI输出总线时钟频率由下式定义（最小分频比2）：

$F_{QSPI_CLK} = F_{AHB} / (PRESCALER + 1)$

20.3.2 工作模式

QuadSPI模块支持以下三种工作模式：

- 外设模式
- 自动查询模式
- 存储器映射模式

20.3.2.1 外设模式

外设模式是QuadSPI最基本的工作模式。这个模式下，QuadSPI模块与其他通信外设类似，CPU和DMA都可以通过访问模块寄存器的方式，来实现与外部QSPI从机的通信。通过OPMODE寄存器，可以配置外设读模式和外设写模式，分别用于对外部QSPI存储器读取数据和写入数据。

通过设置DATALEN寄存器，可以配置QSPI的通信字节长度，当通信开始后，QSPI_nCS引脚被拉低，只有DATALEN指定的数据长度传输完成后，nCS才会回到高电平，结束本次通信过程。

当OPMODE=00，QuadSPI处于外设写模式，即应用可以对外部QSPI存储器写入数据，此时FIFO用作写缓存，AHB总线master对QSPI_DATA寄存器的写操作都会触发对FIFO的PUSH，与此同时，FIFO非空会触发QuadSPI接口的指令和数据发送，从而完成对外部QSPI存储器的写操作。写操作过程中，如果FIFO空，而DATALEN定义的字节数还未发送完成，则QSPI总线时钟被暂停，直到软件或DMA向FIFO写入新数据；如果FIFO满，或者AHB准备写入的数据字节数大于FIFO空余字节数（比如FIFO剩余3字节空间时，AHB发起一个32bit写），则写操作被stall，直到FIFO有足够空间容纳本次需要写入的字节。

当OPMODE=01，QuadSPI处于外设读模式，即应用可以从外部QSPI存储器读取数据，此时FIFO作为读缓存，AHB总线master对QSPI_DATA寄存器的读操作都会触发对FIFO的POP，与此同时，FIFO非满的情况下QuadSPI接口会发送指令和地址并不断接收QSPI存储器返回的数据。读操作过程中，如果FIFO空，

或者FIFO中剩余字节数小于当前AHB读所请求的字节数，则AHB总线读出数据将被填充0；如果FIFO满，则QSPI总线读取外部存储器暂停，直到AHB完成一次FIFO读。

使用外设模式可以实现对外部Flash存储器的如下访问：

- 配置控制寄存器和查询状态寄存器
- 读取外部Flash存储器
- 编程或擦除外部Flash存储器

注意，只有外设模式下可以对片外Flash进行编程和擦除。

命令发送的触发条件

QuadSPI接口在合适的条件下自动开始发送指令，这里包含以下三种情况：

- 应用不需要提供地址（AMODE=00）和数据（DMODE=00或OPMODE=01），即只发送指令，应用写入INSTRUCTION寄存器后，QuadSPI自动开始发送指令
- 应用需要提供地址（AMODE != 00），但是不需要提供数据（DMODE=00或OPMODE=01），即只发送指令和地址，应用在写入QSPI_ADDR寄存器后，QuadSPI自动开始发送指令
- 应用在写入QSPI_DATA寄存器后，QuadSPI开始自动发送指令

当指令开始发送后，BUSY标志寄存器自动置位。

FIFO控制

外设模式下QuadSPI数据收发都是通过FIFO进行，FIFOLVL寄存器用于指示当前FIFO中保存的数据字节数。

在外设写模式下，AHB向FIFO PUSH数据，QuadSPI接口POP数据；在外设读模式下，QuadSPI向FIFO PUSH数据，AHB POP数据。AHB的字节、半字访问必须对齐QSPI_DATA寄存器的低地址。

FIFOTHR寄存器用于定义FIFO threshold。在外设读模式下，当FIFO中有效数据字节数大于FIFOTHR，FTF（FIFO threshold flag）置位，当DATALEN指定的数据长度读取完成后，TCF（Transfer Complete Flag）标志置位，软件应查询FIFO中剩余字节数；在外设写模式下，当FIFO中空字节数大于FIFOTHR时，FTF标志置位。当FIFO数据字节数不满足FTF置位标志时，FTF将自动清零。

当FTIE=1时，FTF标志置位将触发中断。

在外设模式下，当FIFO满时，QuadSPI接口暂停读取QSPI存储器，并且直到FIFO空字节数大于等于4（FIFOLVL<=12）时才会重新开始读取QSPI存储器。

FIFO数据的字节顺序遵循little endian规则，即bit[7:0]是首先被PUSH或POP的。

总线读FIFO时，首先POP出来的字节被对齐到总线32bit数据的低8位[7:0]，第二次POP的字节对齐总

线32位数据的[15:8]，以此类推。

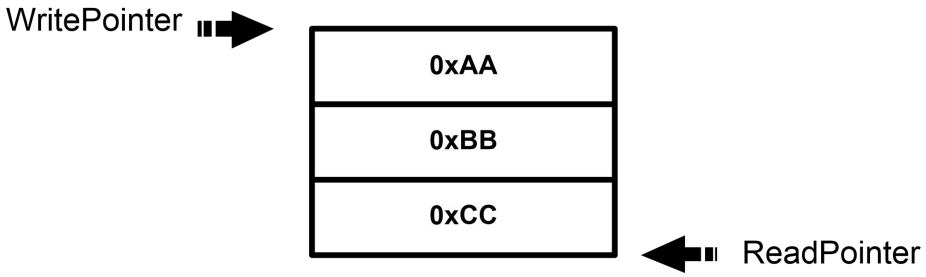
假设FIFO中有下图所示4字节数据，则软件或DMA发起word读时，总线上返回的数据是{0xAA,0xBB,0xCC,0xDD}；发起byte读时，总线返回{0x00,0x00,0x00,0xDD}；发起half-word读时，总线返回{0x00,0x00,0xCC,0xDD}



同理，当总线对FIFO写入数据时，bit[7:0]被首先PUSH，接下来是bit[15:8]，以此类推；假设总线发起32bit写操作，写入数据0xAABBCCDD，则经过4次PUSH后FIFO内数据为：



注意：当FIFO内字节数小于AHB总线读取位宽时，AHB返回数据高位将被补0。例如，FIFO中剩余3字节（参见下图），AHB总线发起一次word读，则QSPI向总线返回{0x00,0xAA,0xBB,0xCC}



DMA支持

使能DMA功能后，AHB总线读写由DMA发起，外设读写模式都可以通过DMA进行。

外设读模式下，软件首先配置指令、地址寄存器，将DMA传输长度和QuadSPI的DATALEN配置为相同

的值。QuadSPI开始读取QSPI存储器后，只要FIFO的FTF标志置位，QuadSPI就发送DMA请求，DMA从FIFO中取走当前字节。

外设写模式下，软件首先配置指令、地址寄存器，将DMA传输长度和QuadSPI的DATALEN配置为相同的值。使能DMA后DMA开始向FIFO写入数据，同时QuadSPI开始发送，只要FIFO的FTF标志置位，QuadSPI就发送DMA请求，DMA继续向FIFO写入数据。

应用举例：外设写模式

下面以复旦微电子公司的8M bits SPI串行NOR Flash芯片FM25Q08A为例，说明如何通过QuadSPI接口实现256字节的page program。

- 为了提升编程效率，首先使能QPI模式（详细方法参见FM25Q08A产品手册）
- 配置AMODE、DMODE、OPMODE
- 向INSTRUCTION寄存器写入”Page Program”指令代码02h
- 向地址寄存器写入编程目标地址
- 将DATALEN寄存器设置为256
- 根据QSPI总线时钟频率设置合适的dummy cycles
- Flush FIFO
- 向QSPI_DATA寄存器写入1st word，4字节数据被PUSH进入FIFO，QuadSPI接口自动拉低nCS发送指令、地址，以及后续4字节数据
- 软件继续向FIFO写入数据，同时注意FIFO水位和中断标志
- 全部256字节发送结束后，QuadSPI自动拉高nCS信号，完成本次传输，并置位传输完成标志

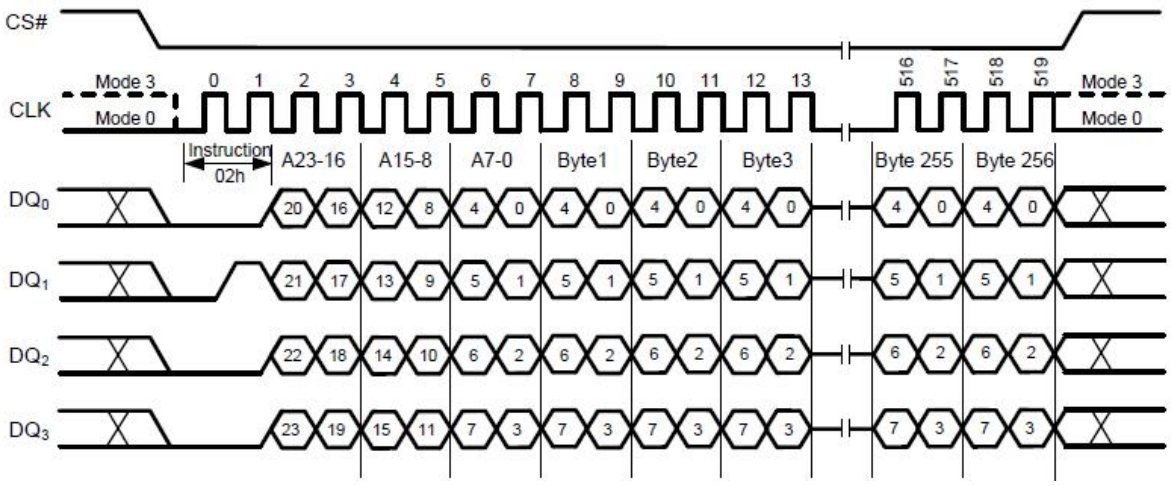


图 20-2 QPI 模式下的 page program

20.3.2.2 自动查询模式

在自动查询模式下（OPMODE=10），QuadSPI模块能够自动轮询外部Flash存储器的状态标志寄存器，而无需CPU参与；QuadSPI以配置的频率周期性读取Flash状态标志，并将返回的数据中的特定bit与期望值自动比较，当符合条件的状态匹配出现时，QuadSPI模块自动停止轮询，并产生中断，通知CPU状态匹配完成。

此模式在等待Flash完成某项操作时非常有效，比如在QuadSPI发送擦除命令后，以自动轮询方式等待Flash擦除完成，在这个过程中CPU可以继续执行其他任务，直到QuadSPI中断通知CPU轮询结束。

自动查询模式一次可以从QSPI存储器读取1~4字节数据，数据被写入QSPI_DATA寄存器，FIFO在这个模式下不起作用。读取的数据经过MASK寄存器屏蔽无关位，然后与MATCH寄存器比较，如果匹配成功，则产生中断标志，BUSY标志清零，QuadSPI停止工作。数据比较支持AND或OR模式，AND模式表示所有未mask位都要匹配，OR模式表示未mask位只要有1bit匹配即可。

20.3.2.3 存储器映射模式

存储器映射模式下，片外Flash被映射到AHB总线上，CPU可以将其视为片内存储器进行直接访问。通过存储映射，可以实现CPU在片外Flash上直接运行代码，即XIP功能。借助XIP，MCU能够有效扩展代码存储空间，而不完全受限于片内Flash容量大小，为应用提供更大的灵活性和扩展性。

在系统总线上，片外Flash将被映射到以下逻辑地址：0x9000_0000 ~ 0x9FFF_FFFF，总共256MB。因此存储器映射模式只支持最大256MB的片外Flash空间访问。

注意：存储器映射模式下只支持对片外Flash的读操作，不支持擦除和编程。

为了提高代码执行效率，QuadSPI采用了Prefetch策略，一旦CPU对片外Flash进行映射读取，QuadSPI会在总线空闲时连续读取后续地址数据，并将数据保存在QuadSPI内部的16字节FIFO中，这在逻辑上相当于实现了4个words的预取指buffer。指令预取会在以下两种情况下停止：

- FIFO满
- CPU取指发生了跳转

当CPU连续地址取指时，AHB总线接口会不断的从FIFO POP数据，每次总线32bit读操作会连续POP 4次，即返回4个字节。当FIFO中有效数据少于4字节时，AHB读访问将被stall，总线被暂停，直到FIFO中有了足够数据。

eXecution In Place (XIP)

XIP模式下CPU直接从外部Flash取指令运行，AHB总线直接访问外部Flash被映射的逻辑地址空间，QuadSPI自动将AHB总线访问转换成QSPI总线访问。指令预取通过FIFO实现，CPU在连续地址取指时，

AHB不断从FIFO POP数据直到FIFO空，与此同时QuadSPI不断向FIFO PUSH数据直到FIFO满。一旦取指出现跳转，则需要flush掉FIFO，重新开始新一轮读操作。

在连续取指过程中，如果CPU取指较慢，FIFO被QuadSPI填满的话，QSPI总线上将暂停时钟，但是nCS保持拉低，片外Flash处于等待访问的状态。

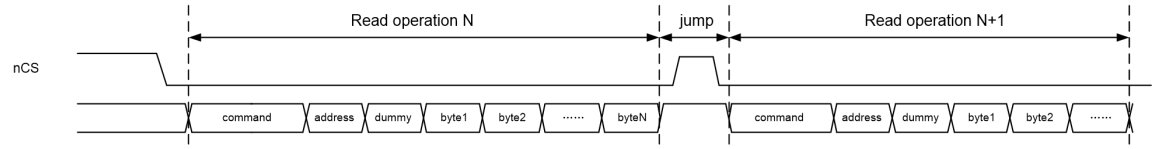


图 20-3 QSPI XIP 示意图

20.3.3 总线配置

QuadSPI支持如下总线配置形式，仅支持SDR格式，不支持DDR：

- Single-SPI
- Dual-SPI
- Quad-SPI

20.3.3.1 Single-SPI (classic SPI)

Single-SPI类似经典SPI总线，仅用到nCS、CLK、DQ0、DQ1四根信号线，其中DQ0作为单向数据输出（SO），DQ1作为单向数据输入（SI），不支持全双工通信。

在Single-SPI配置下，DQ2、DQ3信号是不需要的，可以用做其他功能。如果被连接到片外Flash的nWP和nHOLD接口上，则软件应将对应的GPIO输出固定为1。

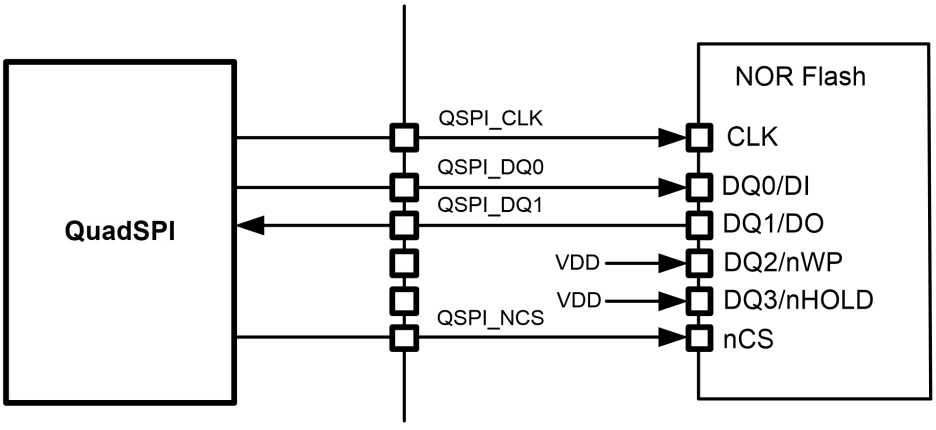


图 20-4 Single-SPI configuration

总线上的数据在CLK下降沿发送，在CLK上升沿被采样，支持SPI Mode0和Mode3，两种模式的差别仅在于总线IDLE时CLK信号的默认电平，参见下图：

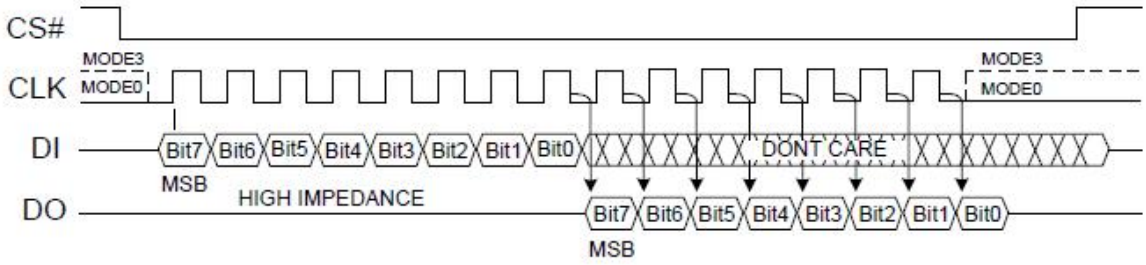


图 20-5 Single-SPI Mode 0 and Mode 3

QuadSPI模块上电后默认为single-SPI模式。

20.3.3.2 Dual-SPI

Dual-SPI配置下同样只用到4根信号线进行通信，与Single-SPI的主要区别是DQ0和DQ1同时参与数据的发送和接收，即DQ0和DQ1变成双向传输。Dual-SPI配置下DQ2和DQ3仍然是不需要的。

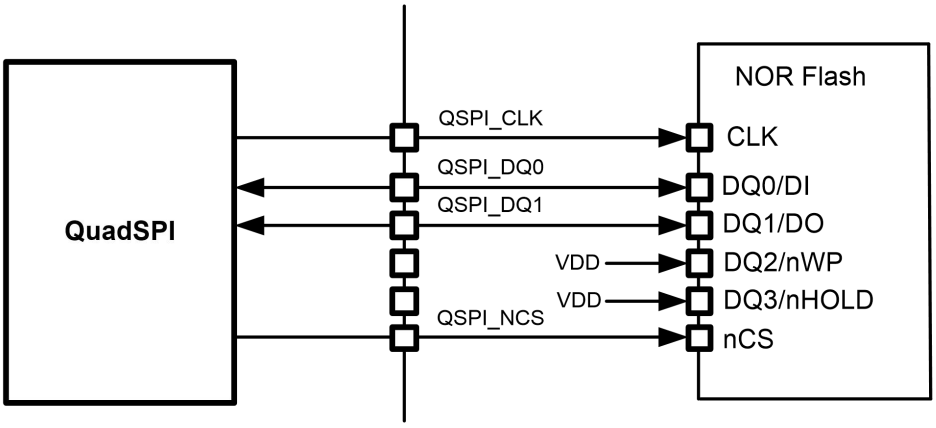


图 20-6 Dual-SPI configuration

20.3.3.3 Quad-SPI

Quad-SPI配置下需要用到6根信号线进行数据收发，DQ0/DQ1/DQ2/DQ3全部用于数据通信，用户可以灵活选择通信数据帧中的命令、地址、数据以单线、双线或四线方式收发。

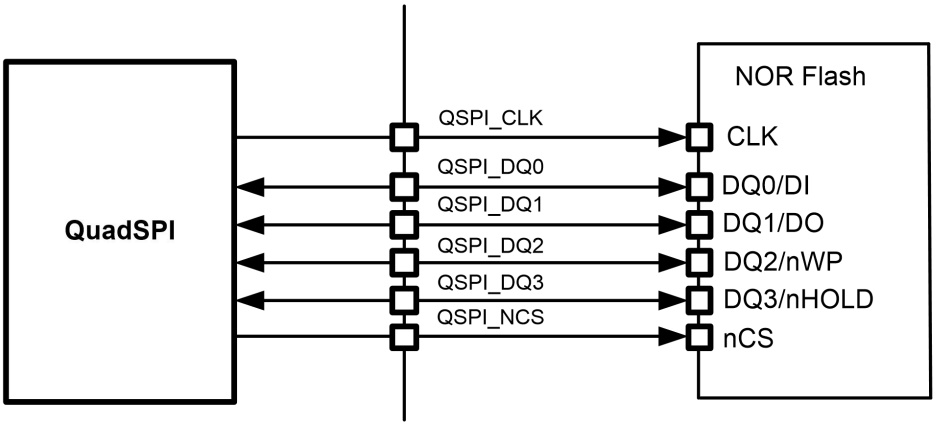


图 20-7 Quad-SPI configuration

20.3.4 帧格式

QSPI通信中，数据帧可以划分为5个部分：instruction, address, alternate, dummy, data

这5个部分根据配置可以存在或跳过，也可以选择以单线、双线或四线方式收发，QuadSPI模块在帧格式上提供了高度的灵活性。

20.3.4.1 Instruction phase

QSPI首次通信都以主机发送特定的指令开始，8bit指令由指令寄存器定义，可以发送任意内容。根据寄存器配置，指令可以以单线、双线或四线发送，某些场景下也可以跳过指令发送阶段。

通过IMODE寄存器，可以配置指令发送方式。

IMODE=00，跳过指令发送

IMODE=01，指令以单线方式发送

以下图为例，QuadSPI对片外Flash发送指令，仅使用DQ0：

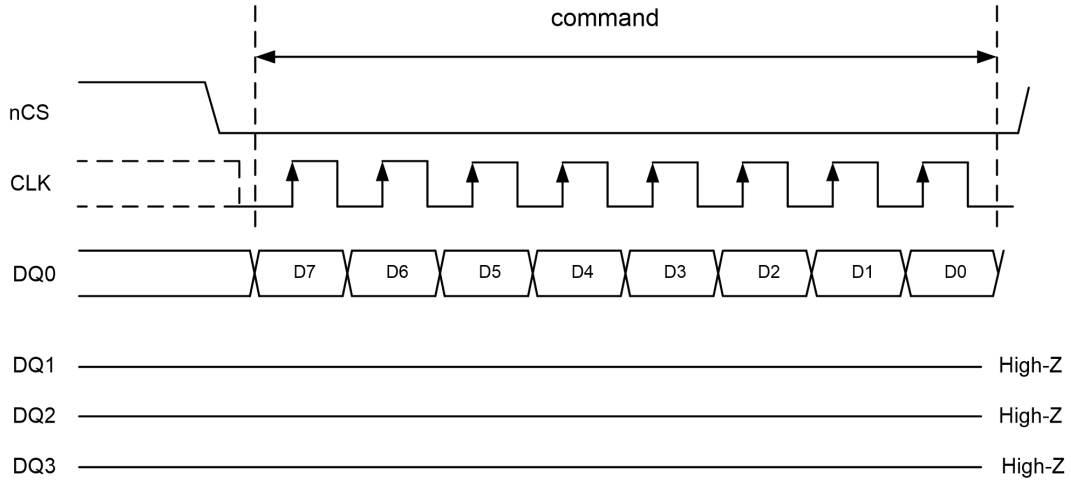


图 20-8 单线发送指令

IMODE=10，指令以双线方式发送

以下图为例，QuadSPI对片外Flash发送指令，使用DQ0和DQ1：

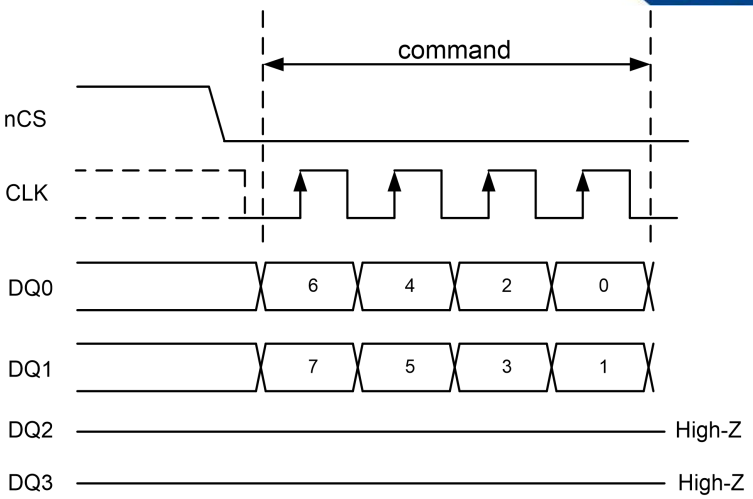


图 20-9 双线发送指令

IMODE=11，指令以四线方式发送

以下图为例，QuadSPI对片外Flash发送指令，使用DQ0、DQ1、DQ2、DQ3：

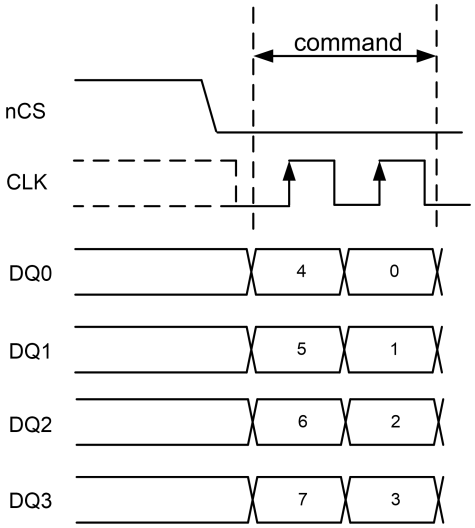


图 20-10 四线发送指令

20.3.4.2 address phase

QuadSPI在address phase向片外存储器发送访问地址，软件可以通过ADSIZE寄存器配置QuadSPI发送1、2、3、4字节地址，通过ADMODE寄存器配置地址以单线、双线、四线方式发送或跳过地址阶段。

在外设模式和自动查询模式下，QuadSPI发送地址由地址寄存器提供；在存储器映射模式下，发送地址直接由AHB总线提供。

20.3.4.3 alternate phase

alternate phase用于发送alternate bytes，某些NOR flash芯片可以使用alternate bytes来启动连续传输模式（比如FM25Q08A的” ontinuous read mode”）。通过ABSIZE寄存器，软件可以配置QuadSPI发送1、2、3、4字节的alternate bytes；通过ABMODE则可以配置以单线、双线、四线方式发送alternate bytes或跳过alternate阶段。QuadSPI发送的alternate bytes内容由寄存器提供。

20.3.4.4 dummy phase

某些QSPI存储器需要在数据帧中插入dummy cycle以提供双向数据口线的turn-around时间。通过DUMCYC寄存器可以设置dummy cycle个数，选择范围是0~31个CLK周期。在dummy phase中，DQ0、DQ1、DQ2、DQ3输出高阻态。

20.3.4.5 data phase

data phase中QuadSPI模块与外部QSPI存储器进行数据收发。

在外设模式和自动查询模式下，通过DATALEN寄存器配置传输数据的长度（bytes），当指定长度的数据被发送和接收完成后，硬件拉高nCS结束本次通信。

在存储器映射模式下，AHB直接通过QuadSPI的FIFO读取数据，DATALEN寄存器不起作用。此时QuadSPI只支持读操作，不支持写操作。AHB可以进行字节、半字和字读取，分别对应FIFO的POP指针+1、+2和+4。

所有读写操作都是通过QSPI_DR寄存器的访问来实现。

通过DMODE寄存器，可以配置数据收发以单线、双线、四线方式进行或者跳过数据阶段。

20.3.5 延迟数据采样

当PCB布线引入较大的时钟数据相位差时，有可能通过数据延迟采样来弥补。正常情况下数据再CLK下降沿发送，接收数据在CLK上升沿被采样；使能了延迟采样后，采样点被人为延迟半个时钟周期，即在下一个CLK的下降沿被采样。

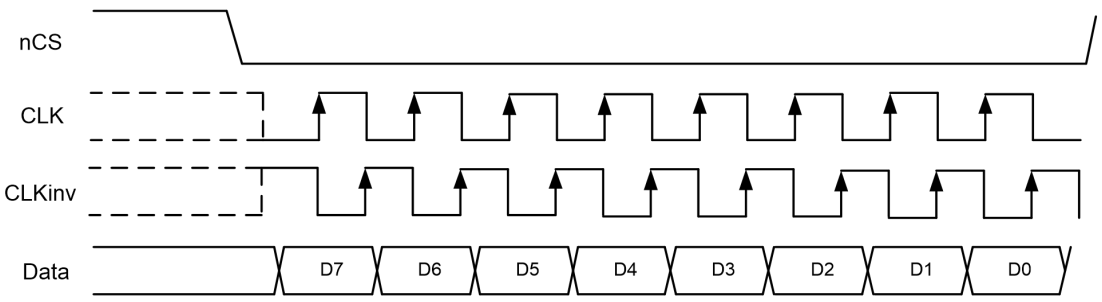


图 20-11 延迟数据采样

20.3.6 BUSY 位和 ABORT 功能

当QuadSPI开始访问QSPI存储器时，BUSY标志寄存器自动置位。

外设模式下，BUSY标志在数据传输完成并且FIFO空的情况下清零。

自动查询模式下，最近一次周期性查询结束并且匹配成功，或者应用执行了ABORT操作后，BUSY标志清零。

存储器映射模式下，BUSY保持高直到timeout或者ABORT。

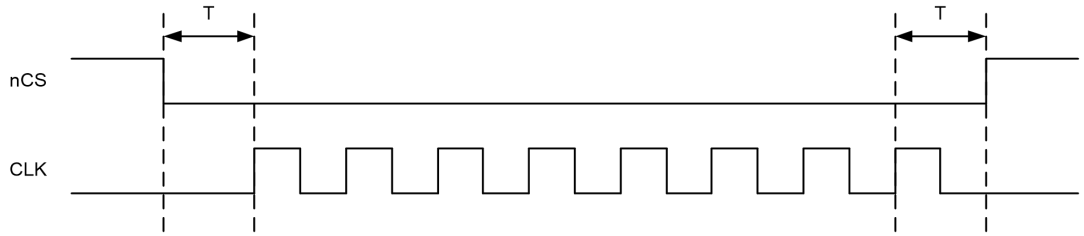
软件通过置位ABORT寄存器可以立即结束当前通信过程，ABORT置位后nCS被拉高，BUSY清零，FIFO被Flush，但是QuadSPI配置寄存器数据不会被清除，ABORT操作后软件可以随时重新发起新的通信。

20.3.7 nCS 控制

QuadSPI通信中，nCS信号完全由硬件控制。默认情况下nCS为高电平，当QSPI操作开始时nCS拉低，通信结束后nCS拉高。

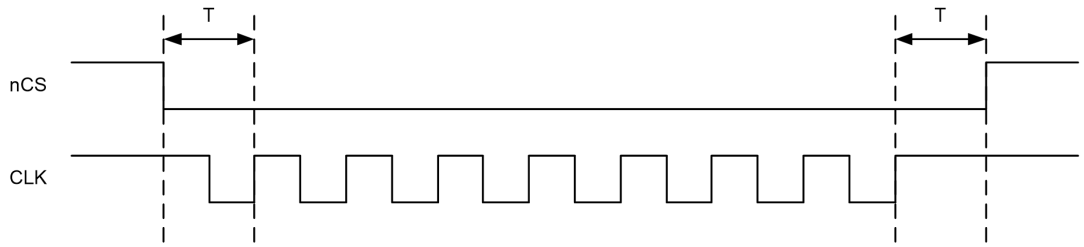
MODE 0

在mode 0配置下，QSPI_CLK与nCS的时序关系如下图：

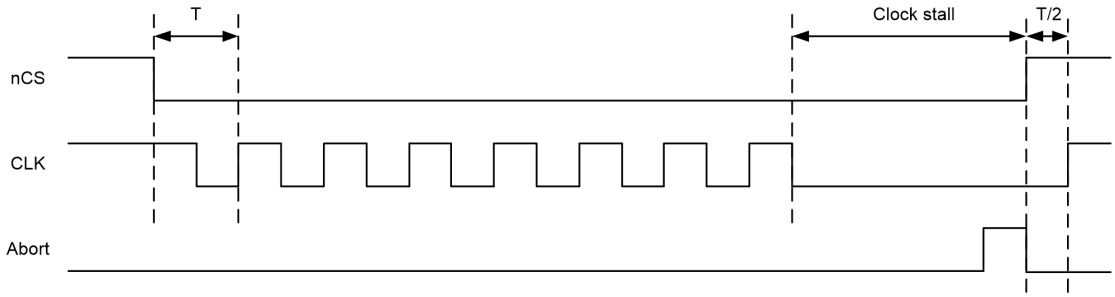


MODE 3

Mode3配置下的nCS与QSPI_CLK时钟的时序关系如下图：



如果在写操作中遇到FIFO空，或者读操作中遇到FIFO满，QSPI_CLK将被暂停，直到应用更新了FIFO状态。ABORT操作会导致nCS和QSPI_CLK回到IDLE状态。



存储器映射模式下，FIFO满后QuadSPI停止总线操作，此时如果使能了超时功能，超时计数器以QSPI_CLK频率开始计数，计数值达到QSPI_TO设定值之后，认为总线超时，拉高nCS。

20.4 高效的 XIP 实现

QuadSPI通过16字节的FIFO提升XIP效率，一旦AHB对0x9000_0000 ~ 0x9FFF_FFFF地址发起读操作，QuadSPI将自动发起对QSPI存储器的读取，并不断将返回数据PUSH到FIFO里。

FIFO的PUSH/POP操作是以字节为单位的，因为SPI Flash的最小访问单位是字节。当AHB发起32位总线取指时，FIFO控制器将执行4次POP，将4字节数据从FIFO读出后存入RDBUFF中，再一次性送出到AHB总线。因此针对QSPI的每次AHB取指需要4个AHBCLK完成。

由于SPI_CLK的最高频率是AHBCLK的2分频，QSPI读取NOR Flash 32bit数据需要8个SPI_CLK，所以QSPI PUSH 32bit进FIFO最短需要16个AHBCLK，而AHB取指需要4个AHBCLK，至少有4倍差距，因此认为4次POP的影响不是很大。

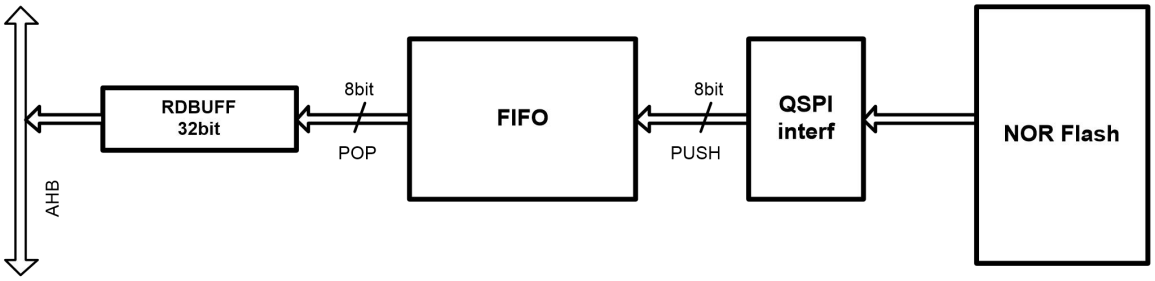
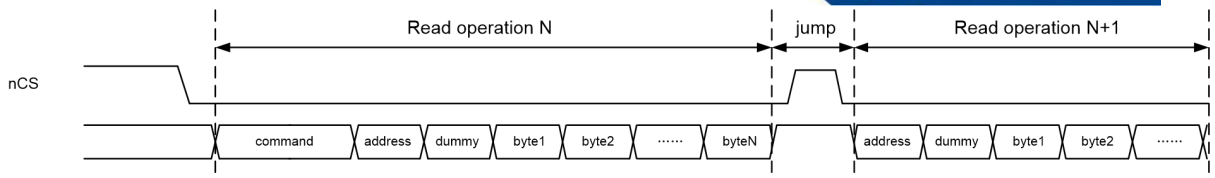


图 20-12 FIFO 结构

AHB触发XIP后，QuadSPI会在当前地址基础上不断递增读取后续地址数据，直到FIFO满才会停止。如果AHB发生任何jump，原有FIFO数据被Flush掉，重新从新地址开始读取。

最高效率的XIP通过与支持"QPI"和"Continuous Read Mode"的NOR Flash配合实现，通过置位CRM寄存器，QuadSPI仅在第一次读取时发送指令，后续读取新的地址时，仅需重新发送目标地址和alternate bytes，无需再发送指令；而支持"QPI"模式的Flash允许主机以四线方式发送指令、地址、alternate bytes和dummy bytes，从而实现最小的读取延迟。

以下图为例，使能了CRM后，发生地址跳转时，从新的地址读取不需要再次发送command。



20.4.1 应用举例

下面以复旦微电子公司的8M bits SPI串行NOR Flash芯片FM25Q08A为例，说明如何通过QuadSPI接口实现高效的XIP方案。

使能QPI模式

FM25Q08A上电后默认为标志SPI模式，首先要使能其QPI模式。

- 将FM25Q08A的Status Register 2中QE位置为1

上电后Status Register Protect (SRP1, SRP0) = 00，状态寄存器处于软件保护状态；在改写QE寄存器之前，需要首先发送Write Enable (06h)指令，解除软件保护。

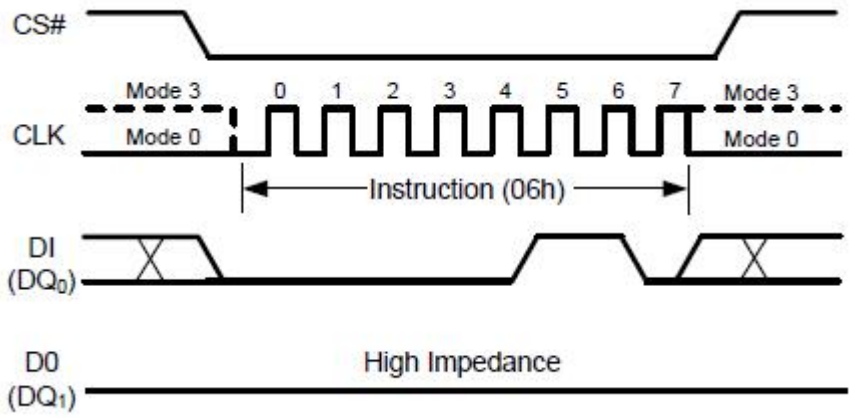


图 20-13 Standard SPI 模式下发送 Write Enable 指令

收到Write Enable指令之后，FM25Q08A的WEL寄存器将被置位，表示已解除写保护。此时，QuadSPI通过发送”Write Status Register2 (31h)”指令将QE位置为1。

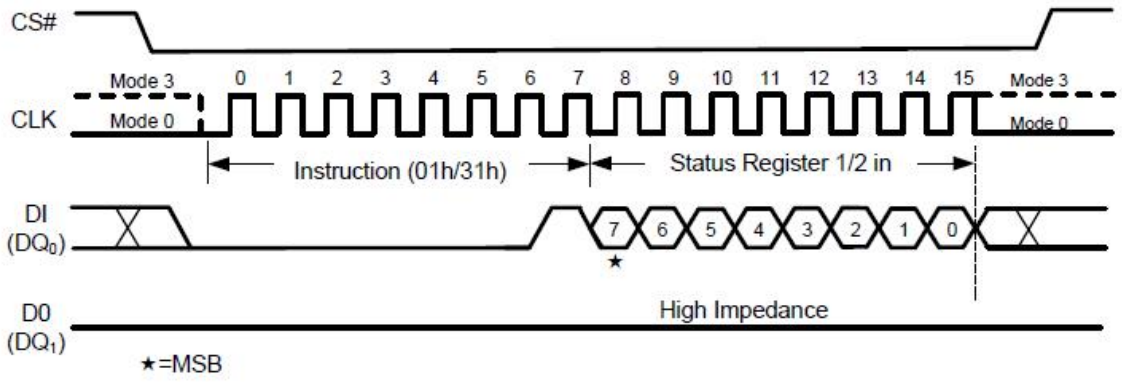


图 20-14 Standard SPI 模式下发送 Write Status Register2 指令

- 发送”Enable QPI (38h)”指令，使能QPI模式

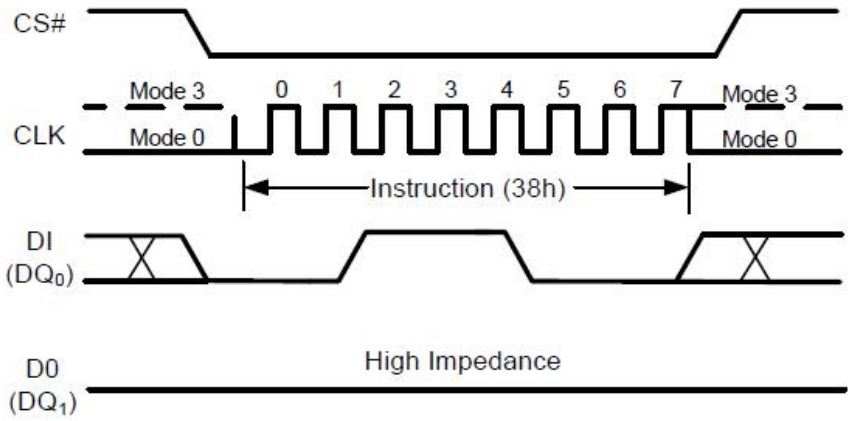


图 20-15 Standard SPI 模式下发送 Enable QPI 指令

完成以上操作后，FM25Q08A被置于QPI模式，后续通信可以采用四线方式进行。

QPI模式读取Flash

推荐使用”Fast Read Quad I/O (Ebh)”指令读取Flash，FM25Q08A的本指令可以支持”Continuous Read Mode”，只需要在第一次读操作时发送指令字节，后续读操作仅需发送地址。

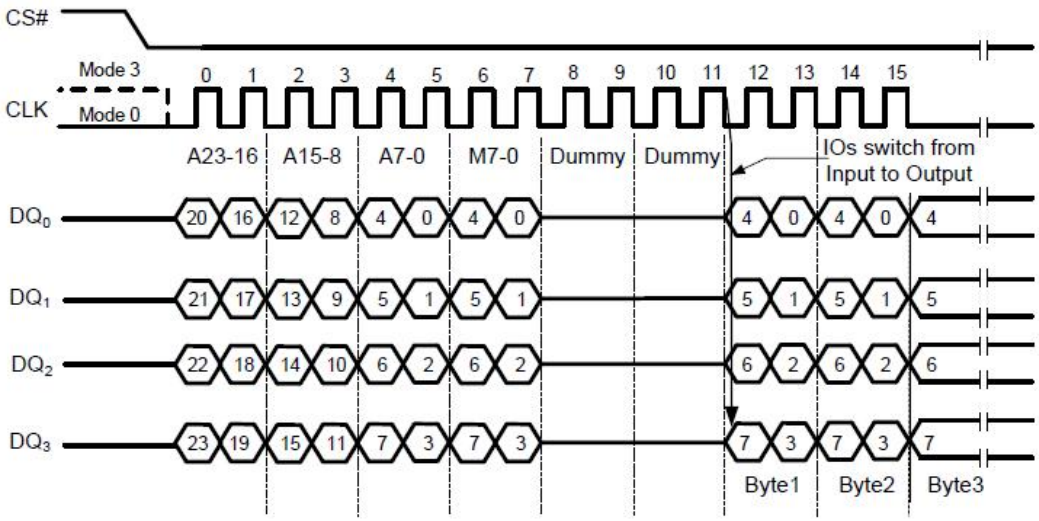


图 20-16QPI 模式下 Fast Read Quad I/O 操作（前一条指令 M5-4=10）

软件操作方法如下：

- 指令寄存器写入Ebh
- Alternate byte的bit[5:4]写入10，ABSIZE配置为1

- 根据CLK频率配置合适的dummy cycles
- 写入XIP起始地址
- IMODE、ADMODE、ABMODE都配置成11（四线方式）
- 软件跳转到XIP映射地址开始取指

20.5 寄存器

模块基地址：0x4000_0800

offset 地址	名称	符号
0x00	QSPI 控制寄存器 (QSPI Control Register)	QSPI_CR
0x04	QSPI 配置寄存器 (QSPI Config Register)	QSPI_CFG
0x08	QSPI 状态寄存器 (QSPI Status Register)	QSPI_SR
0x0C	QSPI 数据长度寄存器 (QSPI Data Length Register)	QSPI_DATALEN
0x10	QSPI 通信配置寄存器 (QSPI Communication Config Register)	QSPI_CCR
0x14	QSPI 地址寄存器 (QSPI Address Register)	QSPI_ADDR
0x18	QSPI alternate bytes 寄存器 (QSPI Alternate Bytes Register)	QSPI_ABR
0x1C	QSPI 数据寄存器 (QSPI Data Register)	QSPI_DR
0x20	QSPI 状态 MASK 寄存器 (QSPI Status Mask Register)	QSPI_SMSK
0x24	QSPI 状态匹配寄存器 (QSPI Status Match Register)	QSPI_SMAT
0x28	QSPI 轮询间隔寄存器 (QSPI Polling Interval Register)	QSPI_PITV
0x2C	QSPI 超时寄存器 (QSPI Time Out Register)	QSPI_TOR

20.5.1 QuadSPI 控制寄存器

名称	QSPI_CR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PRESCLAER							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PMM	-		TOIE	SMIE	FTIE	TCIE	TEIE
位权限	R/W-0	U-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	-				FIFOTHR			

名称	QSPI_CR							
位权限	U-0				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			SSHFT	TCEN	DMAEN	ABORT	EN
位权限	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	**PRESCALER**	QSPI 时钟预分频系数 0：$F_{QSPI_CLK} = F_{AHB}/2$ 1：$F_{QSPI_CLK} = F_{AHB}/2$ 2：$F_{QSPI_CLK} = F_{AHB}/3$ …… 255：$F_{QSPI_CLK} = F_{AHB}/256$ 注意：对于奇数分频，时钟占空比不是 50%，高电平比低电平长一个 AHBCLK 周期
23	**PMM**	轮询匹配模式 0：AND 模式，所有 bit 都匹配才置位 SMF 1：OR 模式，至少 1bit 匹配就会置位 SMF
22:21	--	RFU：未实现，读为 0
20	**TOIE**	超时中断使能 0：禁止超时中断 1：允许超时中断
19	**SMIE**	状态匹配中断使能 0：禁止状态匹配中断 1：允许状态匹配中断
18	**FTIE**	FIFO 水位中断使能 0：禁止 FIFO 水位中断 1：允许 FIFO 水位中断
17	**TCIE**	传输完成中断使能 0：禁止传输完成中断 1：允许传输完成中断
16	**TEIE**	传输错误中断使能 0：禁止传输错误中断 1：允许传输错误中断
15:12	--	RFU：未实现，读为 0
11:8	**FIFOTHR**	FIFO 水位寄存器，仅外设模式下起效 外设写模式： 0：FTF 在 FIFO 空字节大于等于 1 时置位 1：FTF 在 FIFO 空字节大于等于 2 时置位 …… 15：FTF 在 FIFO 空字节等于 16 时置位 外设读模式： 0：FTF 在 FIFO 中字节数大于等于 1 时置位 1：FTF 在 FIFO 中字节数大于等于 2 时置位 …… 15：FTF 在 FIFO 中字节数等于 16 时置位
7:5	--	RFU：未实现，读为 0
4	**SSHFT**	延迟采样使能 0：关闭延迟采样功能 1：使能延迟采样功能

Bit	助记符	功能描述
3	**TCEN**	总线超时使能，此寄存器仅在存储器映射模式下有效 当 BUSY 置位后，如果 QuadSPI 不发起对 QSPI 存储器的访问，超时寄存器开始计数，计数溢出长度由 TIMEOUT 寄存器定义。当 QSPI 总线长时间无动作，计数器溢出，nCS 被自动拉高，强制结束当前传输过程。 0：关闭超时功能 1：使能超时功能
2	**DMAEN**	DMA 使能 0：DMA 功能关闭，QuadSPI 不会发送 DMA 请求 1：DMA 功能开启，QuadSPI 在满足条件时发送 DMA 请求
1	**ABORT**	当前传输终止寄存器，软件写 1 终止传输，硬件拉高 nCS 后自动清零
0	**EN**	QuadSPI 模块使能 0：关闭 QuadSPI 1：使能 QuadSPI

20.5.2 QuadSPI 配置寄存器

名称	QSPI_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	-					CSHT		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CKMODE
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
10:8	**CSHT**	nCS 最小高电平时间，定义了连续两个帧之间 nCS 所需保持高电平的最短时间，以 QSPI_CLK 周期计数 0：至少 1 cycle 1：至少 2 cycles …… 7：至少 8 cycles
7:1	--	RFU：未实现，读为 0
0	**CKMODE**	SPI Clock Mode 寄存器 0：mode 0 1：mode 3

20.5.3 QuadSPI 状态寄存器

名称	QSPI_SR							
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	-			FIFOLVL				
位权限	U-0			R-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BUSY	TOF	SMF	FTF	TCF	~~TEF~~
位权限	U-0		R-0	R/W-0	R/W-0	R-0	R/W-0	~~R/W-0~~

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	FIFOLVL	FIFO 水位标志 此寄存器表示当前 FIFO 中保存的数据字节数，0 表示 FIFO 空，16 表示 FIFO 满 自动查询模式下此寄存器保持 0
7:6	--	RFU：未实现，读为 0
5	BUSY	1 表示 QuadSPI 传输进行中，通信结束后自动清零
4	TOF	超时标志，硬件置位，软件写 1 清零
3	SMF	自动查询模式下表征状态寄存器匹配成功，硬件置位，软件写 1 清零
2	FTF	FIFO threshold 标志，FIFO 水位高于设定阈值时自动置位，低于阈值时自动清零 自动查询模式下，每次读回一组状态值后都会自动置位 FTF，如果软件读取 QSPI_DATA 寄存器则 FTF 清零
1	TCF	传输完成标志，硬件置位，软件写 1 清零
0	--	RFU：未实现，读为 0

20.5.4 QuadSPI 数据长度寄存器

名称	QSPI_DATALEN							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_DATALEN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_DATALEN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_DATALEN[15:8]							
位权限	R/W-0000 0000							

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

Bit	助记符	功能描述
31:0	**QSPI_DATALEN**	传输数据长度为 DATALEN+1（bytes）

20.5.5 QuadSPI 通信配置寄存器

名称	QSPI_CCR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			CRM	OPMODE		DMODE	
位权限	U-0			R/W-0	R/W-00		R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	DUMCYC					ABSIZE	
位权限	U-0	R/W-00000					R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ABMODE		ADSIZE		ADMODE		IMODE	
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	INSTRUCTION							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:29	**--**	RFU：未实现，读为 0
28	**CRM**	Continuous Read Mode 0：每次通信发起时都要发送指令 1：只在第一次通信时发送指令
27:26	**OPMODE**	操作模式 00：外设写模式 01：外设读模式 10：自动查询模式 11：存储器映射模式
25:24	**DMODE**	数据通信模式（data phase） 00：无数据 01：单线 10：双线 11：四线
23	**--**	RFU：未实现，读为 0
22:18	**DUMCYC**	Dummy cycle 个数配置（以 QSPI_CLK 周期计算），0~31
17:16	**ABSIZE**	Alternate bytes 个数 00：8bits alternate bytes 01：16bits alternate bytes 10：24bits alternate bytes 11：32bits alternate bytes

Bit	助记符	功能描述
15:14	**ABMODE**	Alternate bytes 发送模式 00：无 alternate bytes 01：单线 10：双线 11：四线
13:12	**ADSIZE**	地址字节长度 00：8bits 地址 01：16bits 地址 10：24bits 地址 11：32bits 地址
11:10	**ADMODE**	地址字节发送模式 00：无地址字节 01：单线 10：双线 11：四线
9:8	**IMODE**	指令发送模式 00：无指令字节 01：单线 10：双线 11：四线
7:0	**INSTRUCTION**	QuadSPI 发送的指令字节

20.5.6 QuadSPI 地址寄存器

名称	QSPI_ADDR							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_ADDR[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_ ADDR[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_ ADDR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_ ADDR[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	**QSPI_ADDR**	发送给 QSPI 存储器的地址，在存储器映射模式下无效

20.5.7 QuadSPI alternate bytes 寄存器

名称	QSPI_ABR
offset	0x18

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

Bit	助记符	功能描述
31:0	**QSPI_ABR**	发送给 QSPI 存储器的 alternate bytes

20.5.8 QuadSPI 数据寄存器

名称	QSPI_DR							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_DATA[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_DATA[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_DATA[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_DATA[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	**QSPI_DATA**	QSPI 数据寄存器 外设模式写操作时，对 QSPI_DR 寄存器写入的数据将被 push FIFO，支持字节、半字、字写入，分别对 FIFO 压入 1、2、4 字节；如果写入字节数大于 FIFO 中空字节数，当前写操作被挂起，直到 FIFO 中有足够空间容纳当前写入数据。 外设模式读操作时，读取 QSPI_DR 寄存器将 pop FIFO，支持字节、半字、字读取，分别从 FIFO 弹出 1、2、4 字节；如果读取字节数大于 FIFO 中有效字节数，当前读操作被挂起，直到 FIFO 中有足够字节可以被读取，或者传输完成，后一种情况下只弹出最后几个实际有效字节。 对 QSPI_DR 的访问必须对齐低地址，即字节访问必须对齐 QSPI_DR[7:0]，半字访问必须对齐 QSPI_DR[15:0]

20.5.9 QuadSPI 状态 MASK 寄存器

名称	QSPI_SMSK							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_SMSK[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_SMSK[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_SMSK[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_SMSK[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	**QSPI_SMSK**	自动状态查询模式下的状态 mask 寄存器，对应 bit 写 0 屏蔽相应状态位

20.5.10 QuadSPI 状态匹配寄存器

名称	QSPI_SMAT							
offset	0x24							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QSPI_SMAT[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QSPI_SMAT[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_SMAT[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_SMAT[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	**QSPI_SMAT**	自动状态查询模式下的状态匹配寄存器 比较对象是 QSPI_DATA & QSPI_SMSK

20.5.11 QuadSPI 轮询间隔寄存器

名称	QSPI_PITV							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

名称	QSPI_PITV							
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QSPI_PITV[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_PITV[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	**QSPI_PITV**	自动状态查询模式下的轮询间隔（polling interval），定义为 QSPI_CLK 周期数

20.5.12 QuadSPI 超时寄存器

名称	QSPI_TOR							
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	QSPI_TO[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QSPI_TO[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	**QSPI_TO**	超时周期设置，定义为 QSPI_CLK 周期数，仅在存储器映射模式下有效 当 FIFO 满之后，QSPI 总线行为停止，超时计数器开始计数，计数值到达 QSPI_TO 设定值之后，拉高 nCS

21 单边沿半字节传输模块（SENT）

21.1 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。

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。

21.2 消息格式

21.2.1 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：

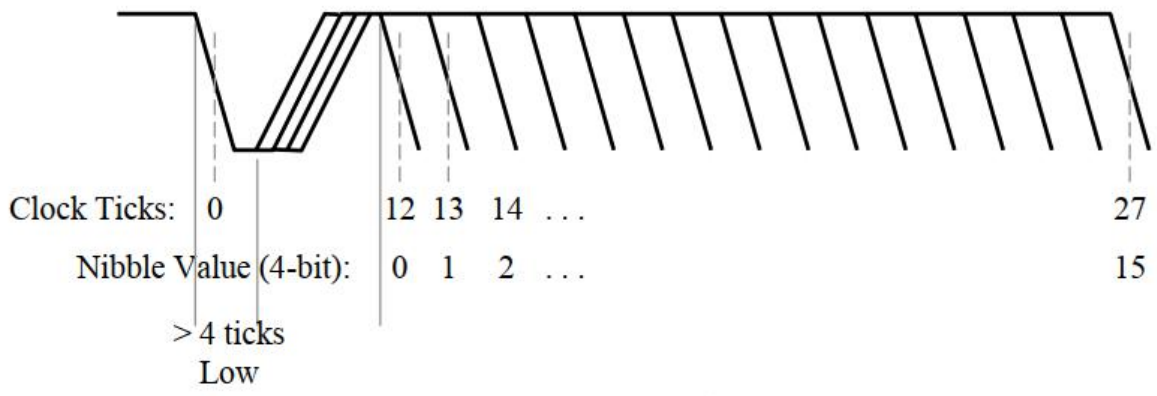


图 21-1nibble 波形示意图

发送数据0和发送数据7时的nibble波形如下所示：

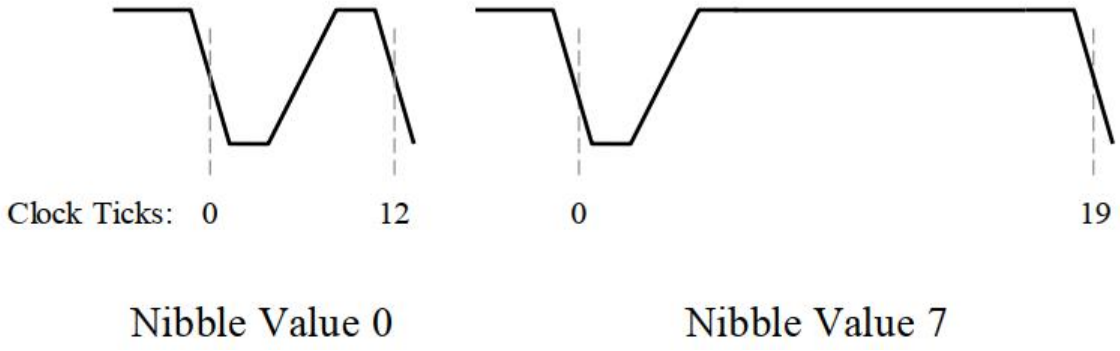


图 21-2 数据 0 和数据 7 的 nibble 波形

21.2.2 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。

21.2.3 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。

21.2.3.1 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：

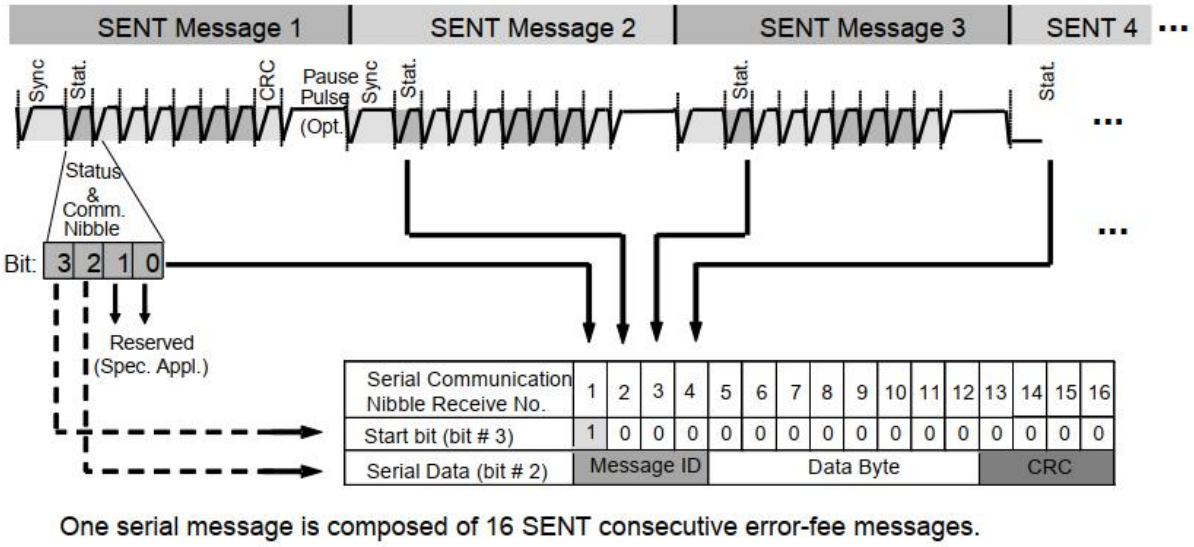


图 21-3 短消息格式 option channel

21.2.3.2 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。

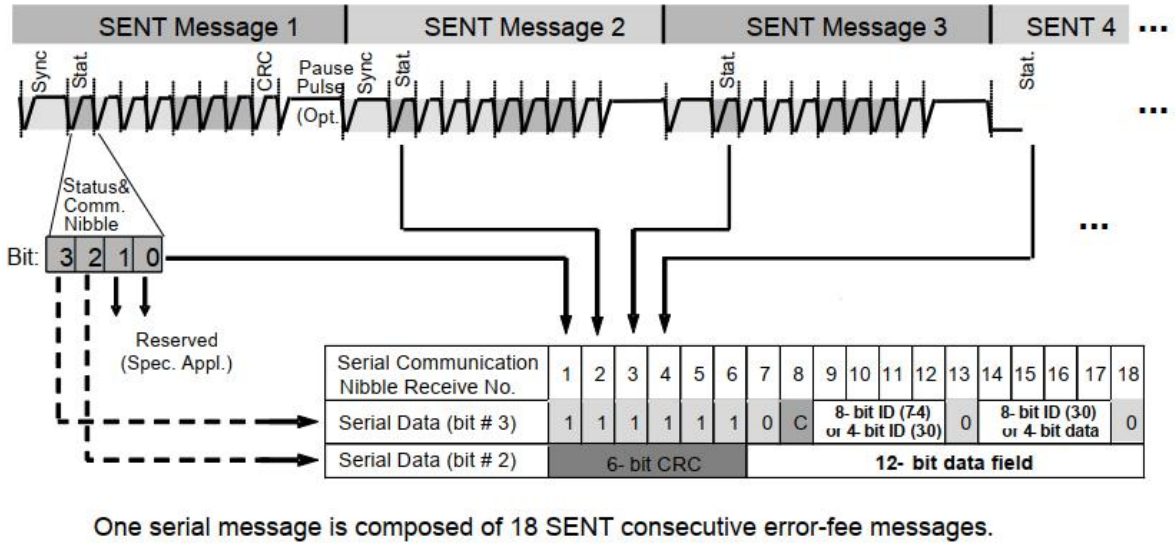


图 21-4 加强消息格式 option channel

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。

21.2.4 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：

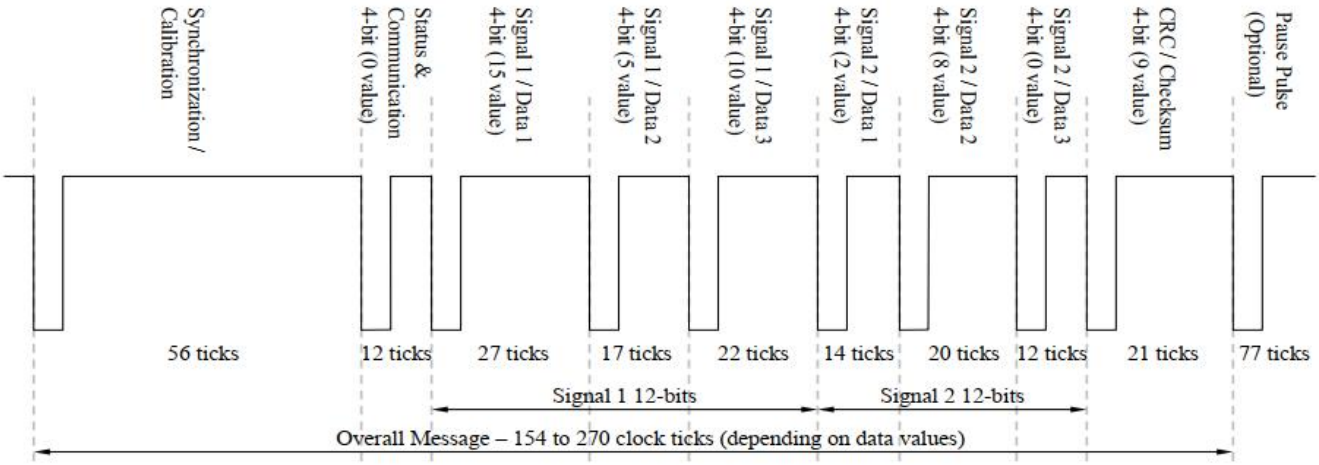


图 21-5 消息帧格式

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。

21.2.5 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。

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

21.2.6 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。

21.3 结构框图

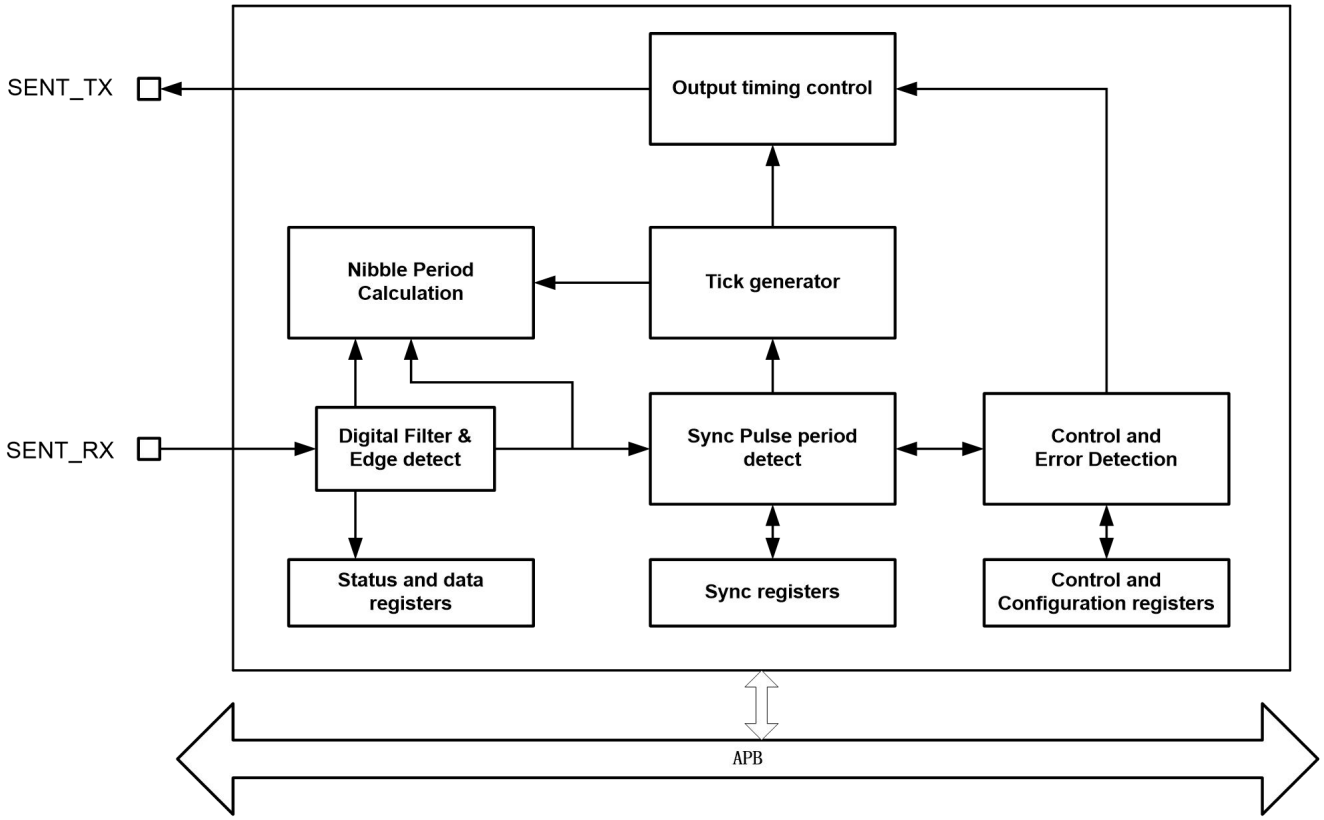


图 21-6SENT 模块结构框图

21.4 功能描述

21.4.1 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。

$$F_{SENT} = F_{APBCLK} / prescaler$$

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

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。

21.4.2 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；如果CRCEN=0则同时需写入软件计算的CRC值，否则CRC由硬件计算并自动更新。
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。

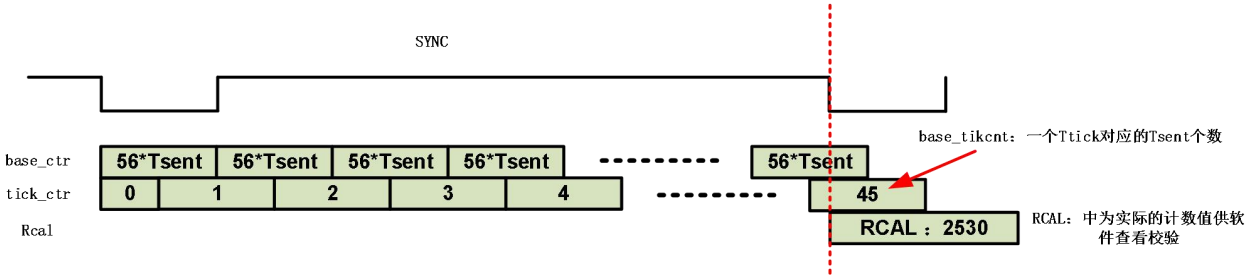
注意：参与CRC计算的data nibble个数由NIBCNT指定。

21.4.3 接收模式

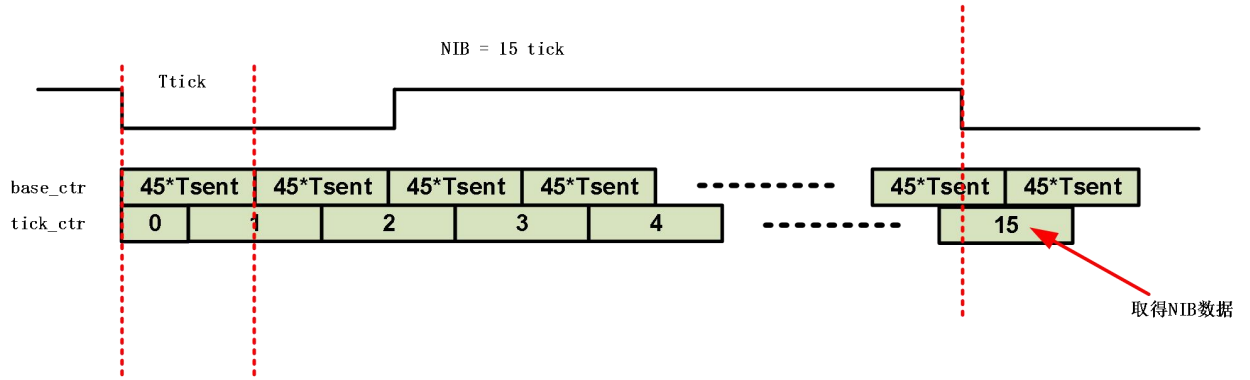
接收模式下，SENT模块自动计算sync pulse的宽度，采样到的sync pulse宽度处于SYNCMIN和SYNCMAX寄存器定义的范围之内时，模块开始正常数据接收。Sync脉冲宽度被自动写入RCAL寄存器中，用作后续数据接收时的校准参数。

在接收SYNC阶段，会在sent_rx的下降沿开始以 $F_{SENT} = F_{APBCLK} / (prescaler + 1)$ 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Ttick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Ttick使用较低频率。

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。

接收初始化流程：

1. 根据Ttick配置合适的模块频率与预分频值
2. 置位MOD，进入接收模式
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。

21.4.4 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）

21.4.5 输入数字滤波

SENT引脚输入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）：

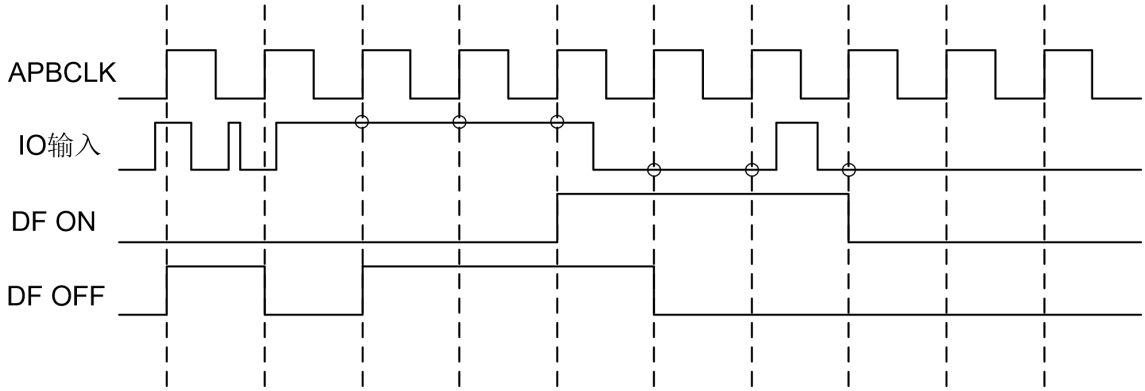


图 21-7SENT_RX 输入数字滤波示意图

21.5 寄存器

SENT0模块起始地址：0x4001 4800

SENT1模块起始地址：0x4001 4C00

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
0x20		
0x24		

21.5.1 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

21.5.2 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-0000 0000							
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 输入数字滤波控制，表示 APBCLK 采样数字滤波长度 0x00：关闭数字滤波 0x01：采样 1 个 APBCLK … 0xFF：采样 255 个 APBCLK
23:22	--	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

Bit	助记符	功能描述
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	帧中 Data Nibbles 个数 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
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

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

21.5.4 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

21.5.5 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	-		-	SSERR	NBERR	CRCERR	NCNTERR	SYNCERR
位权限	U-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	-		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 FRAMETIME 个 TICK 0：采样到低电平，处于接收过程中
17	PAUSE	Pause 状态指示 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	--	RFU：未实现，读为 0
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 仅配置为接收工作时有效
11	NBERR	Nibble 测量错误（<12ticks 或>27ticks）标志，硬件置位，软件写 1 清零 仅配置为接收工作时有效
10	CRCERR	CRC 校验错误标志，硬件置位，软件写 1 清零 仅配置为接收工作时有效
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 仅配置为接收工作时有效
8	SYNCERR	SYNC monitor 支持连续两个符合正确 sync 长度的下降沿长度存在（由于 pause 长度可以符合 sync 的长度），但是如果连续出现第三个符合 sync 长度的下降沿存在时会置起此标志，软件写 1 清零 仅配置为接收工作时有效
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

Bit	助记符	功能描述
		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

21.5.6 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

21.5.7 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

21.5.8 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

22 直接存储访问控制器（DMA）

22.1 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
- 地址指针递增、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）

22.2 结构框图

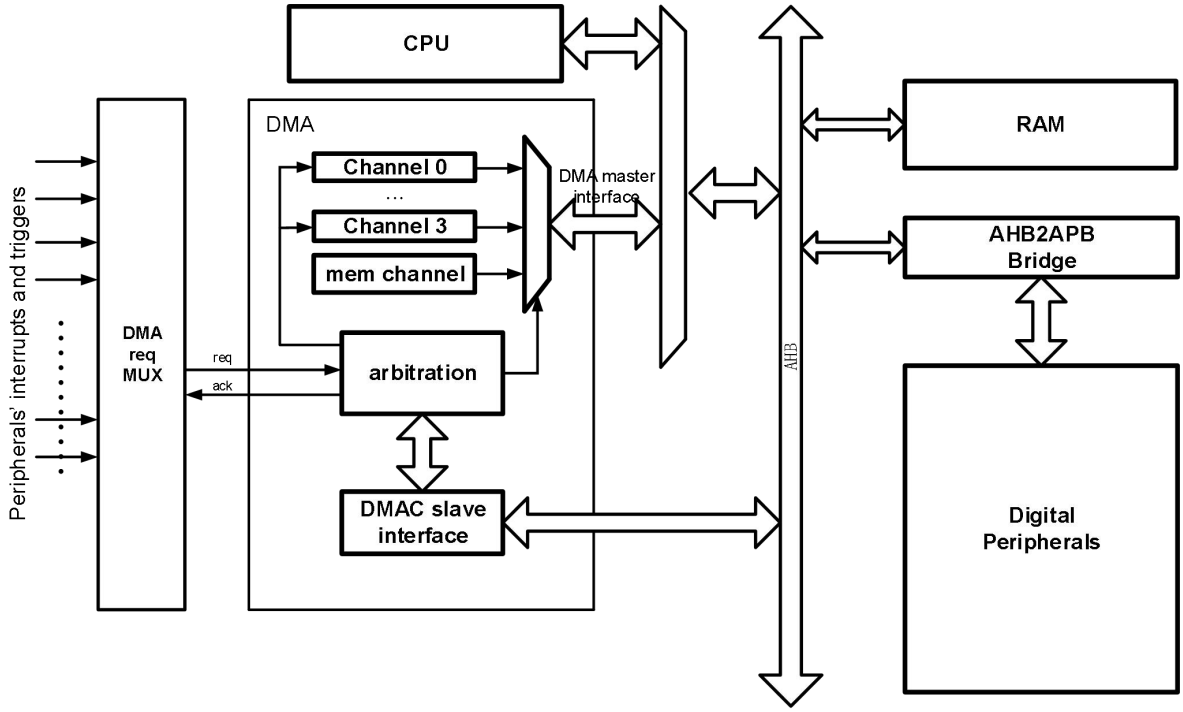


图 22-1 DMA 结构框图

22.3 功能描述

22.3.1 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

22.3.2 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7个相同的通道。

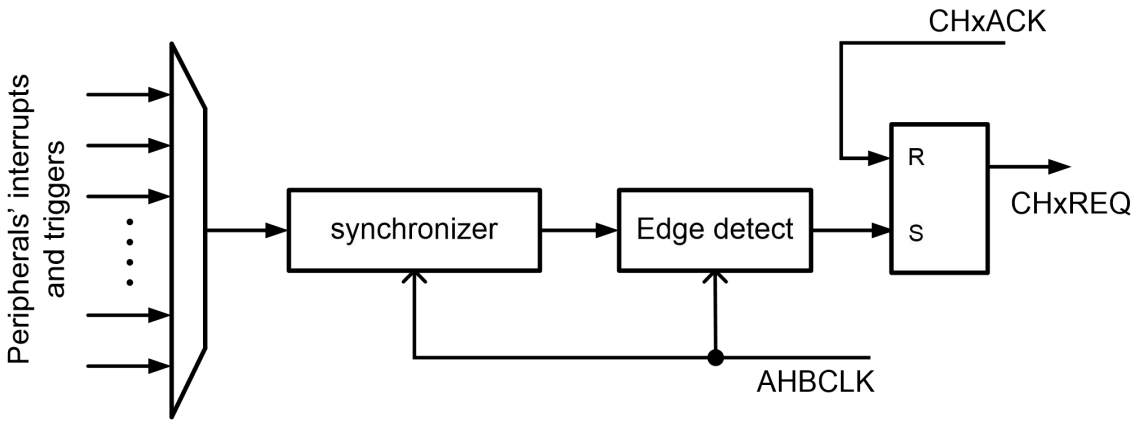


图 22-2 外设请求处理

22.3.3 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channel7。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。

22.3.4 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

22.3.5 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。

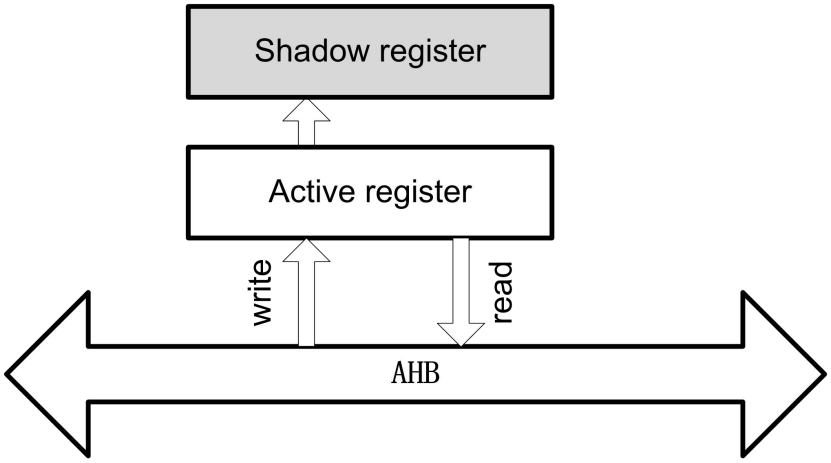


图 22-3 活动寄存器和影子寄存器

22.3.6 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。

22.3.7 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。

22.3.8 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。

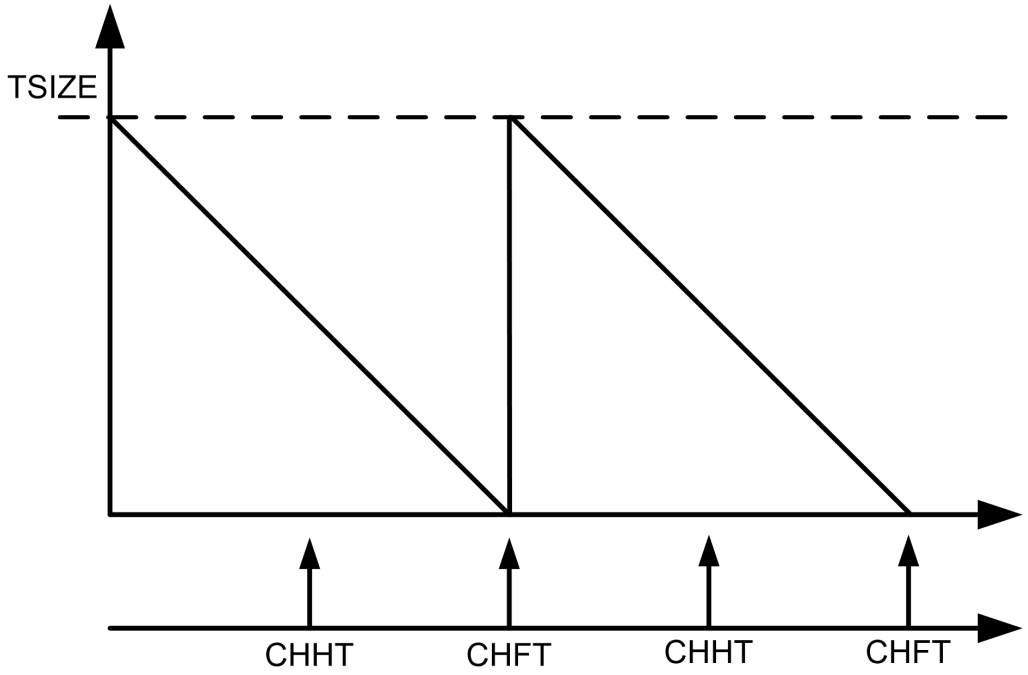


图 22-4 循环模式下的传输长度寄存器

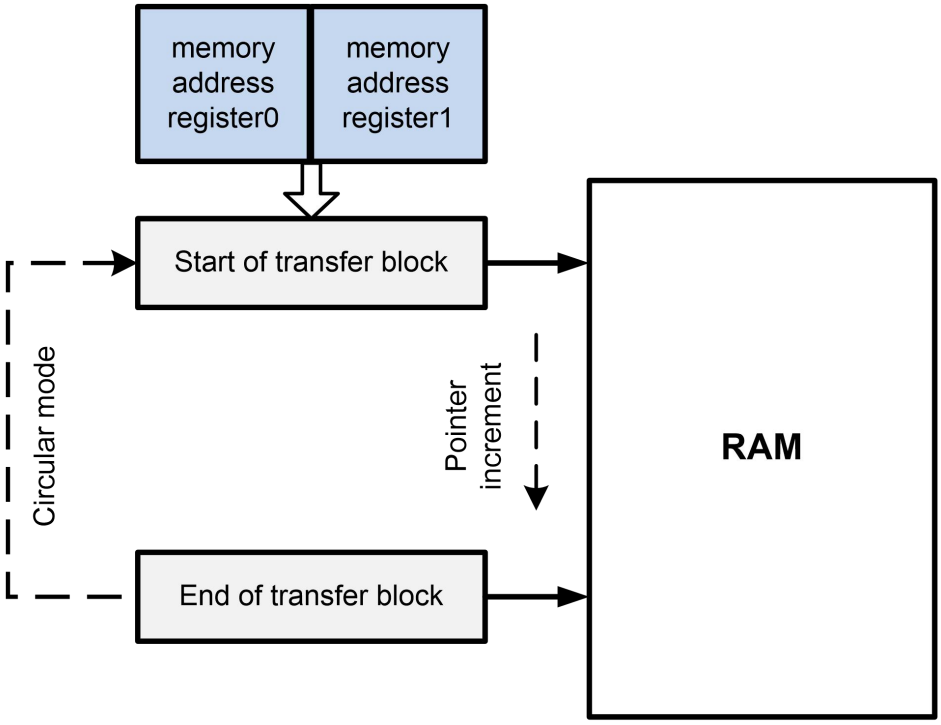


图 22-5 循环模式下的源/目的地址寄存器和存储器指针

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。

22.3.9 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。

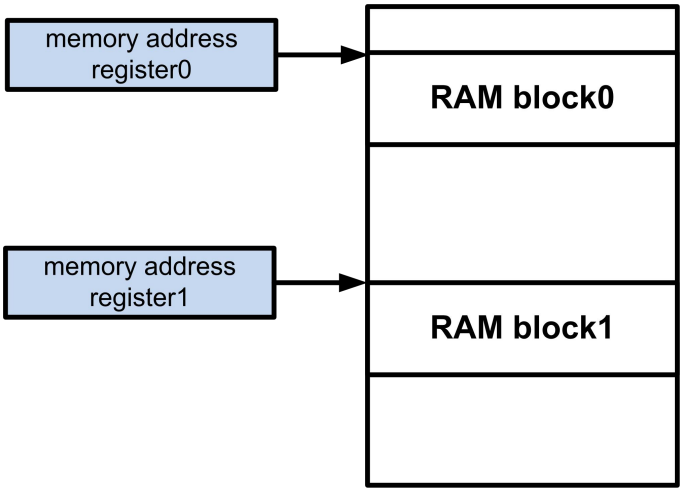


图 22-6 double buffering

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。

22.4 通道控制

22.4.1 DMA 请求映射

DMA 共有 4 个优先级可配的外设通道，每个通道可接受最多 128 个外设请求，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	***SPI2_RX***
3	**-**
4	**-**
5	**-**
6	***SPI0_TX***
7	***SPI1_TX***
8	***SPI1_TX***
9	**-**
10	**-**
11	**-**
12	***UART0_RX***
13	***UART1_RX***
14	***UART2_RX***
15	***UART3_RX***
16	***UART4_RX***
17	***UART5_RX***
18	***UART6_RX***
19	**-**
20	**-**
21	**-**
22	***UART7_RX***
23	***UART0_TX***
24	***UART1_TX***
25	***UART2_TX***
26	***UART3_TX***
27	***UART4_TX***
28	***UART5_TX***
29	***UART6_TX***
30	***UART7_TX***
31	**-**
32	**-**
33	**-**
34	***LPUART0_RX***

35	-
36	-
37	***LPUART0_TX***
38	-
39	-
40	-
41	-
42	-
43	-
44	-
45	-
46	-
47	-
48	-
49	-
50	-
51	***CRC***
52	-
53	***ATIM0_CH1***
54	***ATIM0_CH2***
55	***ATIM0_CH3***
56	***ATIM0_CH4***
57	***ATIM0_TRIG***
58	***ATIM0_COM***
59	***ATIM0_UEV***
60	-
61	-
62	-
63	-
64	-
65	-
66	-
67	-

68	-
69	-
70	-
71	-
72	-
73	-
74	-
75	-
76	-
77	-
78	-
79	-
80	***TAU00***
81	***TAU01***
82	***TAU02***
83	***TAU03***
84	***TAU04***
85	***TAU05***
86	***TAU06***
87	***TAU07***
88	***TAU10***
89	***TAU11***
90	***TAU12***
91	***TAU13***
92	***TAU14***
93	***TAU15***
94	***TAU16***
95	***TAU17***

96	
97	
98	***BSTIM_TRIG***
99	***LPTIM_TRIG***
100	**-**
101	**-**
102	**-**
103	***ADC***
104	
105	***QSPI***
106	***RTC***
107	***TSI_WR***
108	***TSI_RD***
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	***DAC***
120	
121	**-**
122	**-**
123	**-**
124	**-**
125	**-**
126	**-**
127	***SWTRIG***

表 22-1PDMA 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，其转换完成中断可以连接到 DMA，触发 DMA 将自己的数据寄存器数据搬运到 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。

22.4.2 Request Select

使用外设DMA前，每个通道通过RSEL寄存器，从最多128个请求源中选择1个作为本通道的传输触发信号。

22.5 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

offset 地址	名称	符号
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
0x80		
0x84		
0x88		
0x8C		
0x90		
0xA0		
0xA4		
0xA8		
0xAC		
0xB0		
0xC0		
0xC4		
0xC8		
0xCC		
0xD0		
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_CH7FLSAR
0xE8	通道 7 RAM 指针寄存器 （Channel 7 RAM Address Register）	DMA_CH7RAMAR
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
0x304	DMA 通道 0 传输剩余长度寄存器 （DMA Channel 0 Thransfer Size Register）	DMA_CH0TFSADDR

offset 地址	名称	符号
0x308	DMA 通道 1 传输剩余长度寄存器 （DMA Channel 1 Thransfer Size Register）	DMA_ CH1TFSADDR
0x30C	DMA 通道 2 传输剩余长度寄存器 （DMA Channel 2 Thransfer Size Register）	DMA_ CH2TFSADDR
0x310	DMA 通道 3 传输剩余长度寄存器 （DMA Channel 3 Thransfer Size Register）	DMA_ CH3TFSADDR

22.5.1 DMA 全局控制寄存器（DMA_GCR）

名称	DMA_GCR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						ADDRERR_EN	EN
位权限	U-0						R/W-0	R/W-0

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	**ADDRERR_EN**	DMA 错误地址中断使能 (DMA address error interrupt enable) 1：允许错误地址中断 0：禁止错误地址中断
0	**EN**	DMA 全局使能 (DMA enable) 1：DMA 使能 0：DMA 关闭

22.5.2 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
位名	-							
位权限	U-0							

名称	DMA_SWRR							
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
31:7	--	RFU：未实现，读为 0
6:0	**SWTRIG**	外设通道软件触发请求，软件写 1 触发一次 DMA 请求，DMA 响应请求后硬件自动清零。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[1] – Channel 1 SWTRIG[0] – Channel 0 others: RFU

22.5.3 DMA 中断状态标志寄存器（DMA_ISR）

名称	DMA_ISR							
Offset	0x300							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	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
31:17	-	RFU：未实现，读为 0
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

22.5.4 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							

名称	DMA_CHxCR(x=0~3)							
位	Bit23	Bit22	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

2.5.5 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						
位权限	R/W-0	R/W-0000000						
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
14:8	**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

22.5.6 通道 x 外设地址寄存器（DMA_CHxPAR）

名称	DMA_CHxPAR (x=0~3)							
Offset								
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

名称	DMA_CHxPAR (x=0~3)							
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。

22.5.7 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) 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。

22.5.8 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

名称	DMA_CHxMAR1 (x=0~3)							
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。

22.5.9 通道 7 控制寄存器（DMA_CH7CR）

名称	DMA_CH7CR							
Offset								
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
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)

位号	助记符	功能描述
		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

22.5.10 通道 7 Flash 指针寄存器（DMA_CH7FLSAR）

名称	DMA_CH7FLSAR							
Offset								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DFLS_SEL	-			FLSAD			
位权限	R/W-0	U-0			R/W-0000 0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	FLSAD							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FLSAD							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FLSAD							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31	**DFLS_SEL**	擦写 data flash 时置一，擦写 code flash 时置 0.

位号	助记符	功能描述
30:28	-	RFU：未实现，读为 0
27:0	**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 注：软件需确保 FLSAD 不能超过 flash 的最大 word 地址，否则可能会导致访问地址出错。

22.5.11 通道 7 RAM 指针寄存器（DMA_CH7RAMAR）

名称	DMA_CH7RAMAR							
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	RAMAD							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAMAD							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAMAD							
位权限	R/W-0000 0000							

位号	助记符	功能描述
31:24	-	RFU：未实现，读为 0
23:0	**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 注：软件需确保 RAMAD 不能超过 ram 的最大 word 地址，否则可能会导致访问地址出错。

23 循环冗余校验（CRC）

23.1 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，FM33FH0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

23.2 软件配置过程

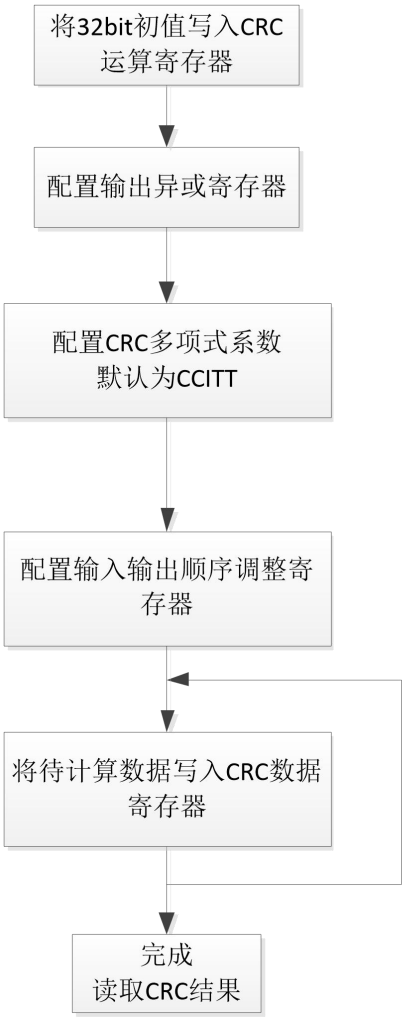


图 23-1CRC 运算流程图

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。

23.3 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

23.4 DMA 接口

CRC与DMA之间通道为单向的（RAM->CRC）。CRC模块可以通过DMA模块读取并校验RAM数据，其工作流程如图所示。CRC向DMA发起请求，DMA接收请求后，读取RAM并将数据写入CRC模块的CRCDR寄存器中。CRC模块接收到数据后，撤销DMA请求并开始计算校验值，校验完成后，CRC模块重新置起DMA请求。

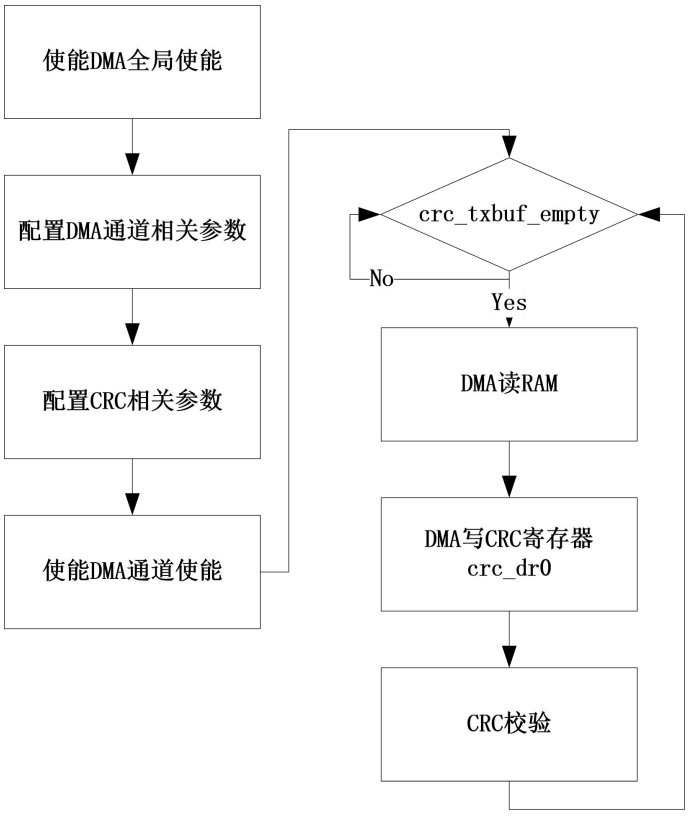


图 23-2 使用 DMA 对 RAM 中的数据进行 CRC 运算

23.5 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。

23.6 寄存器

模块起始地址：0x4001_8000

offset 地址	名称	符号
0x00000000	CRC 数据寄存器 （CRC Data Register）	CRC_DR
0x00000004	CRC 控制状态寄存器 （CRC Control Register）	CRC_CR
0x00000008	CRC LFSR 寄存器 （CRC Linear Feedback Shift Register）	CRC_LFSR
0x0000000C	CRC 输出异或寄存器 （CRC output XOR Register）	CRC_XOR
0x0000001C	CRC 多项式寄存器 （CRC Polynominal Register）	CRC_POLY

23.6.1 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]。

23.6.2 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	-
位权限	U-0						R/W-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
位权限	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-10	

位号	助记符	功能描述
31:10	-	RFU：未实现，读为 0
9	**OPWD**	WORD 操作使能 (Operation by Word) 0：字节操作，CRC 计算仅针对 CRCDR 最低字节进行 1：字操作，CRC 计算针对 CRCDR 全部 4 字节进行
8	-	RFU：未实现，读为 0
7:6	**RFLTIN**	CRC 输入反转控制 (Reflected Input) 00：输入不反转 01：输入按字节反转 10：输入按半字反转 11：输入按字反转 例如：计算数据为 0x11223344， 如果 RFLTIN==01，则将数据变为 0x8844CC22，再进行计算 如果 RFLTIN==10，则将数据变为 0x448822CC，再进行计算 如果 RFLTIN==11，则将数据变为 0x22CC4488，再进行计算
5	**RFLTO**	CRC 输出反转控制 (Reflected Output) 0：输入不反转 1：输入按字节反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 0x1234，则输出的结果为 0x2C48 如果 RFLTO==0，则直接输出 0x1234 注意：此结果不一定为最终输出结果，还需要看 XOR 是否为 1，详见本寄存器 bit2 说明
4	**RES**	CRC 结果标志位，只读 (Result Flag) 0：CRC 结果为 0 1：CRC 结果非全 0
3	**BUSY**	CRC 运算标志位，只读 (Busy) 0：CRC 运算结束 1：CRC 运算进行中
2	**XOR**	输出异或使能 (Output XORed with CRC_XOR register enable) 0：输出不异或 CRC_XOR 寄存器 1：输出异或 CRC_XOR 寄存器
1:0	**SEL**	CRC 多项式位宽选择 (Polynomial width Selection) 00：32 位 01：16 位 10：8 位 11：7 位

23.6.3 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

23.6.4 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。

23.6.5 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）

24 高级定时器（ATIM）

24.1 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。

24.2 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

24.3 结构框图

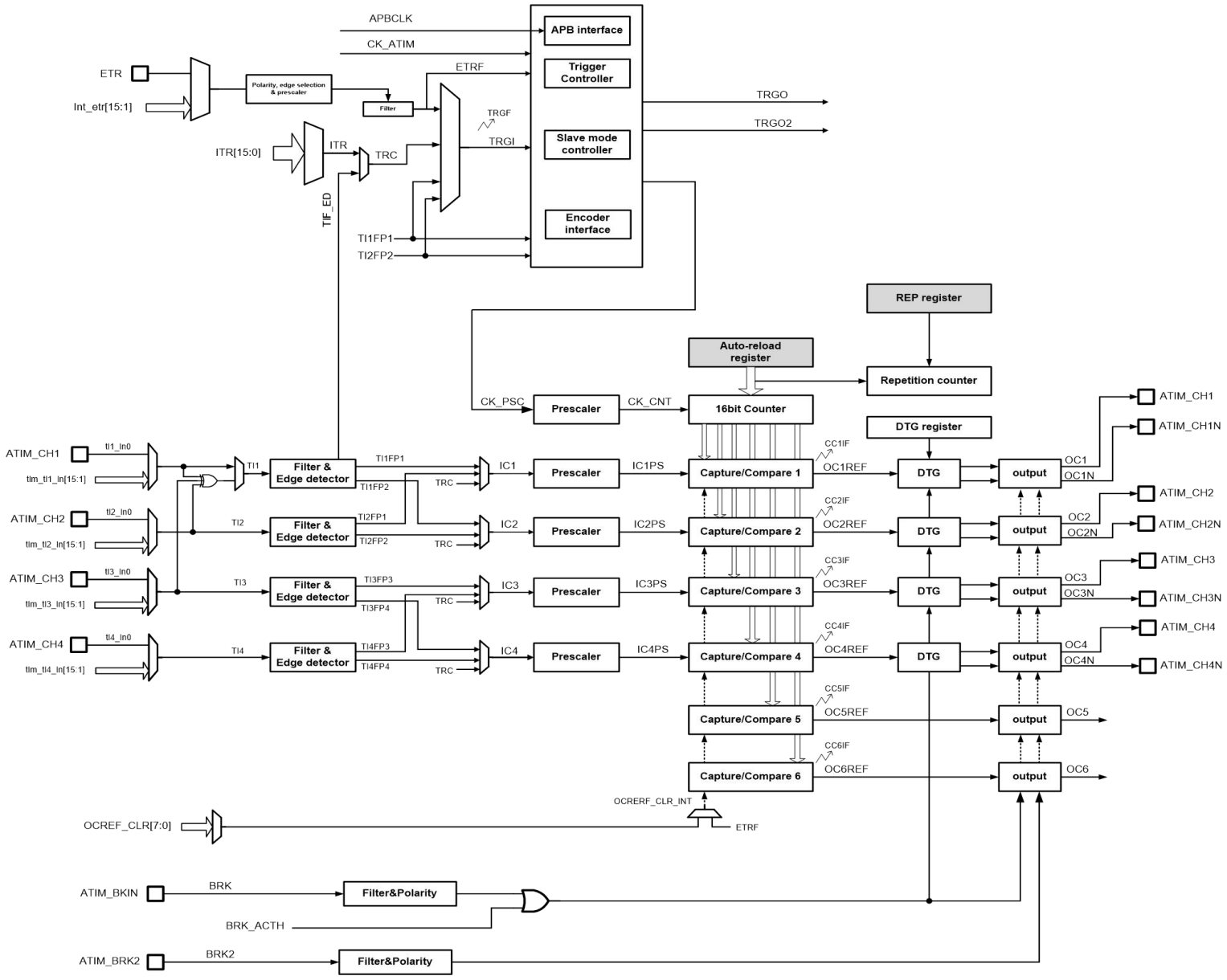


图 24-1 高级定时器结构框图

24.4 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 24-1 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 同频或者 2x
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 24-2 内部信号连接

ATIM的ITR[15:0]与片内信号连接：

ITRx	ATIM0
ITR0	TAU0_TRGO[0]
ITR1	TAU0_TRGO[1]
ITR2	TAU0_TRGO[2]
ITR3	TAU0_TRGO[3]
ITR4	TAU0_TRGO[4]
ITR5	TAU0_TRGO[5]
ITR6	TAU0_TRGO[6]
ITR7	TAU0_TRGO[7]
ITR8	BTIM_TRGO

ITR9	LPT_TRGO
ITR10	TAU1_TRGO[0]
ITR11	TAU1_TRGO[1]
ITR12	TAU1_TRGO[2]
ITR13	TAU1_TRGO[3]
ITR14	TAU1_TRGO[4]
ITR15	TAU1_TRGO[5]

表 24-3ITR 信号连接

ATIM的ETR[15:0]与片内信号连接：

ETRx	ATIM0
ETR0	ATIM0_ETR
ETR1	Comp1_out
ETR2	Comp2_out
ETR3	
ETR4	
ETR5	
ETR6	
ETR7	
ETR8	
ETR9	
ETR10	
ETR11	
ETR12	
ETR13	
ETR14	
ETR15	

表 24-4ETR 信号连接

ATIM的OCREF_CLR与内部信号连接：

OCREF_CLRx	ATIM0
OCREF_CLR0	COMP1_OUT
OCREF_CLR1	COMP2_OUT
OCREF_CLR2	
OCREF_CLR3	
OCREF_CLR4	
OCREF_CLR5	
OCREF_CLR6	
OCREF_CLR7	

表 24-5OCREF_CLR 信号连接

ATIM的CH1[15:0]与片内信号连接：

ATIMx_CH1	ATIM0
CH1 0	ATIM0_CH1
CH1 1~15	RFU

表 24-6 CH1 信号连接

ATIM的CH2[15:0]与片内信号连接：

ATIMx_CH2	ATIM0
CH2 0	ATIM0_CH2
CH2 1~15	RFU

表 24-7 CH2 信号连接

ATIM的CH3[15:0]与片内信号连接：

ATIMx_CH3	ATIM0
CH3 0	ATIM0_CH3
CH3 1	UART0_RX
CH3 2	UART1_RX
CH3 3	UART2_RX
CH3 4	UART3_RX
CH3 5	UART4_RX
CH3 6	UART5_RX
CH3 7	UART6_RX
CH3 8	UART7_RX
CH3 9	COMP1_OUT
CH3 10	COMP2_OUT
CH3 11	RFU
CH3 12	RFU
CH3 13	RFU
CH3 14	RFU
CH3 15	RFU

表 24-8 CH3 信号连接

ATIM的CH4[15:0]与片内信号连接：

ATIMx_CH4	ATIM0
CH4 0	ATIM0_CH4
CH4 1	UART0_RX
CH4 2	UART1_RX
CH4 3	UART2_RX
CH4 4	UART3_RX
CH4 5	UART4_RX
CH4 6	UART5_RX
CH4 7	UART6_RX
CH4 8	UART7_RX
CH4 9	COMP1_OUT
CH4 10	COMP2_OUT
CH4 11	RFU
CH4 12	RFU
CH4 13	RFU
CH4 14	RFU
CH4 15	RFU

表 24-9 CH4 信号连接

24.5 功能描述

24.5.1 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（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。

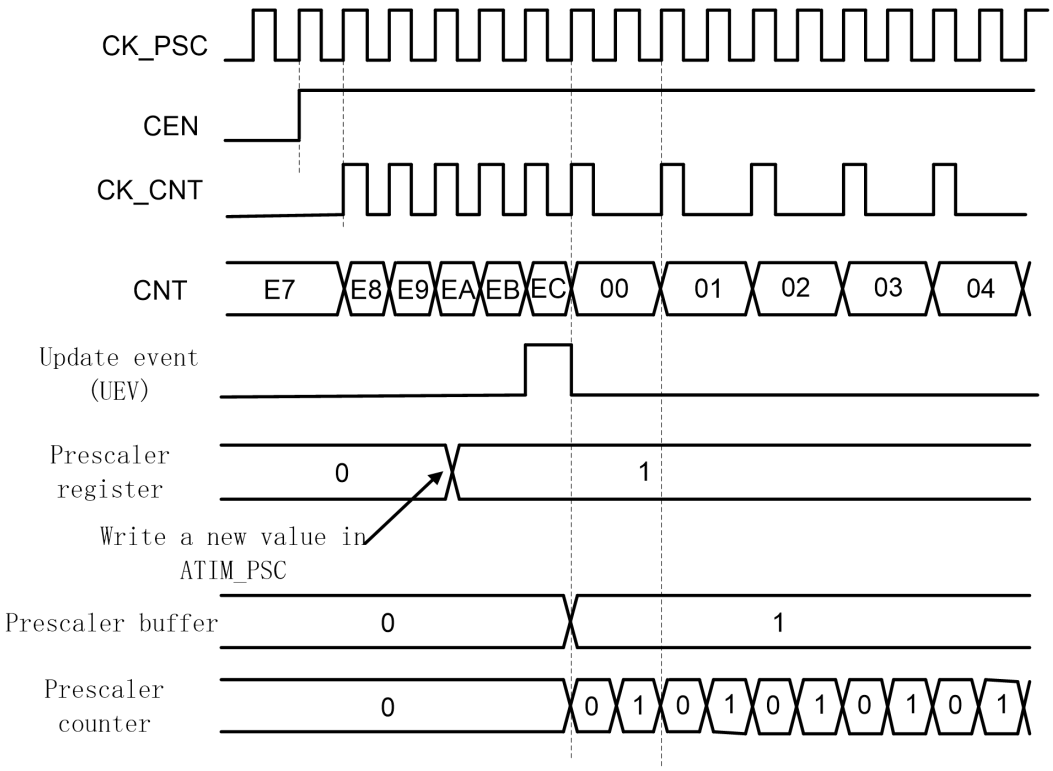


图 24-2 预分频从 1 变为 2 的波形

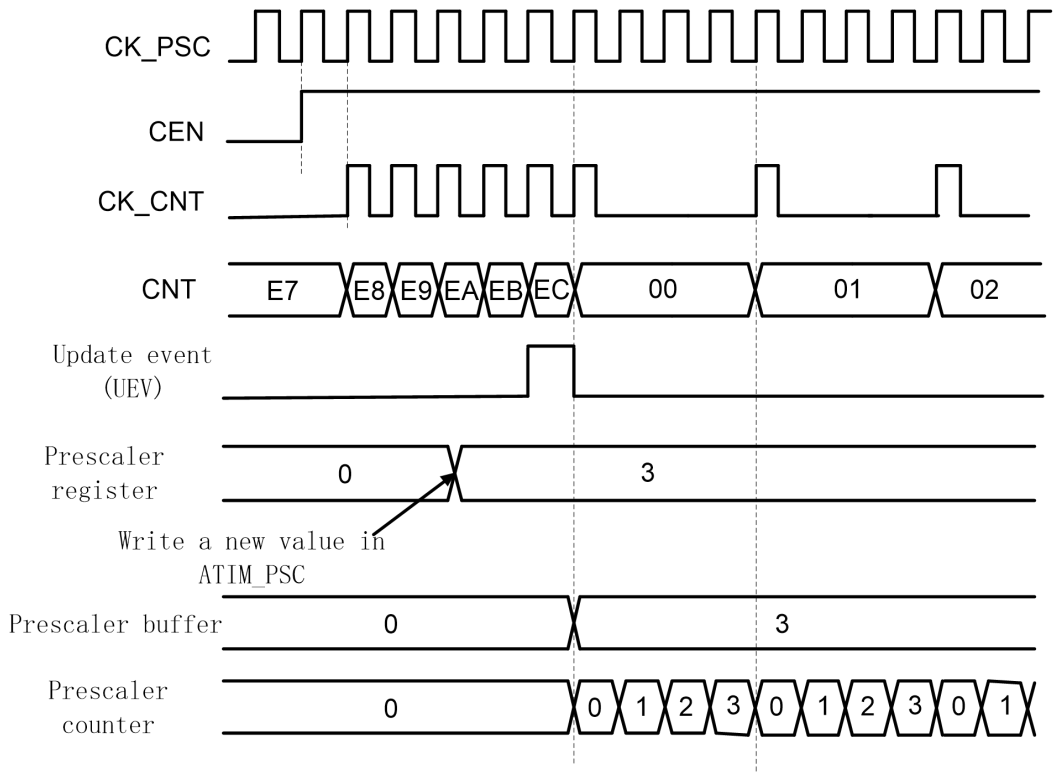


图 24-3 预分频从 1 变为 4 的波形

24.5.2 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。

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

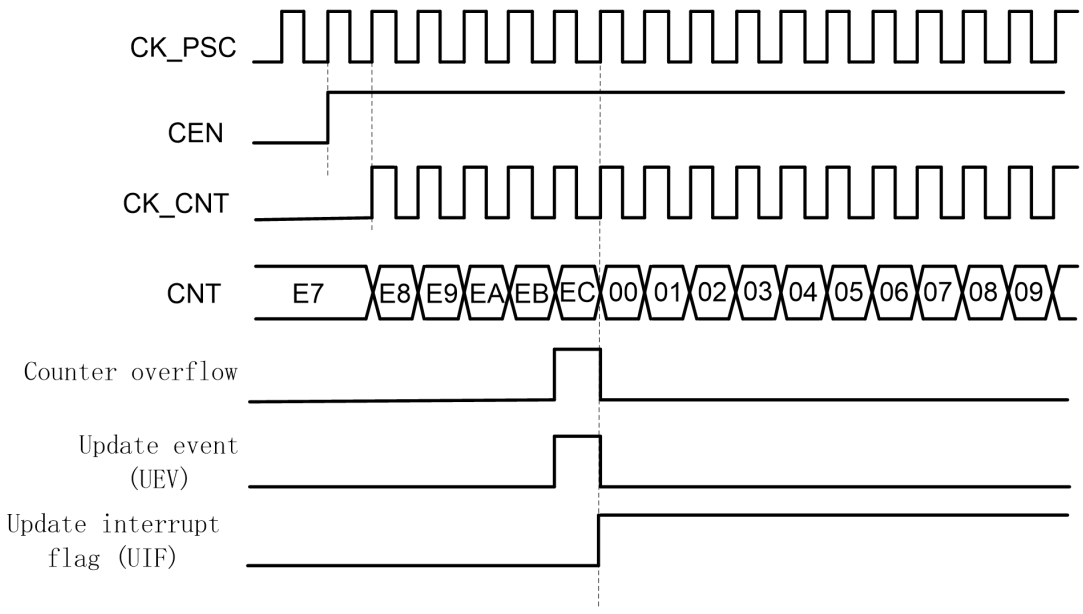


图 24-4 向上计数波形，内部时钟不分频

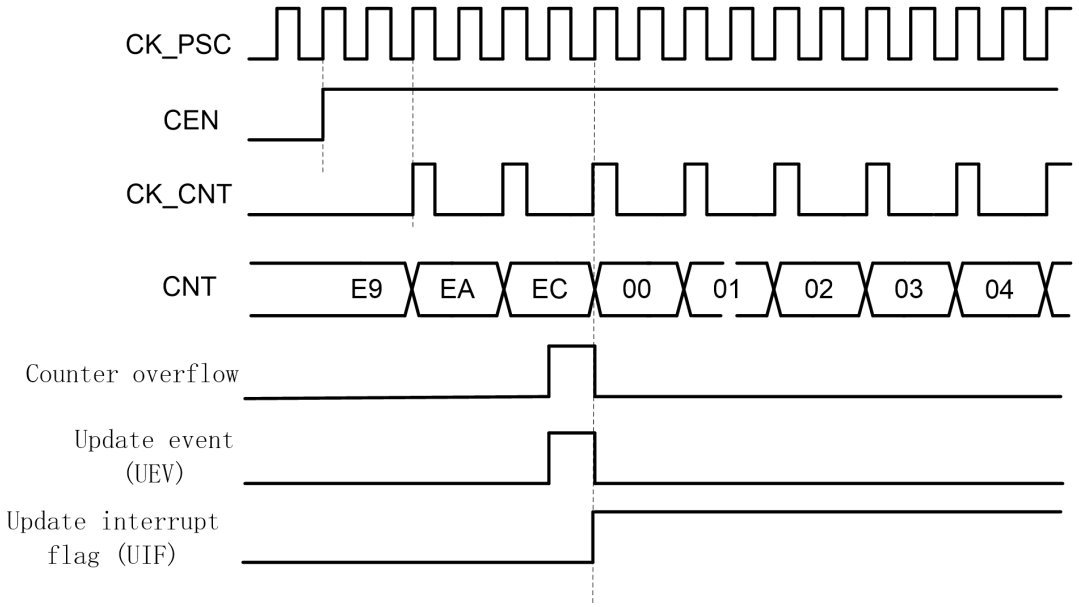


图 24-5 向上计数波形，内部时钟 2 分频

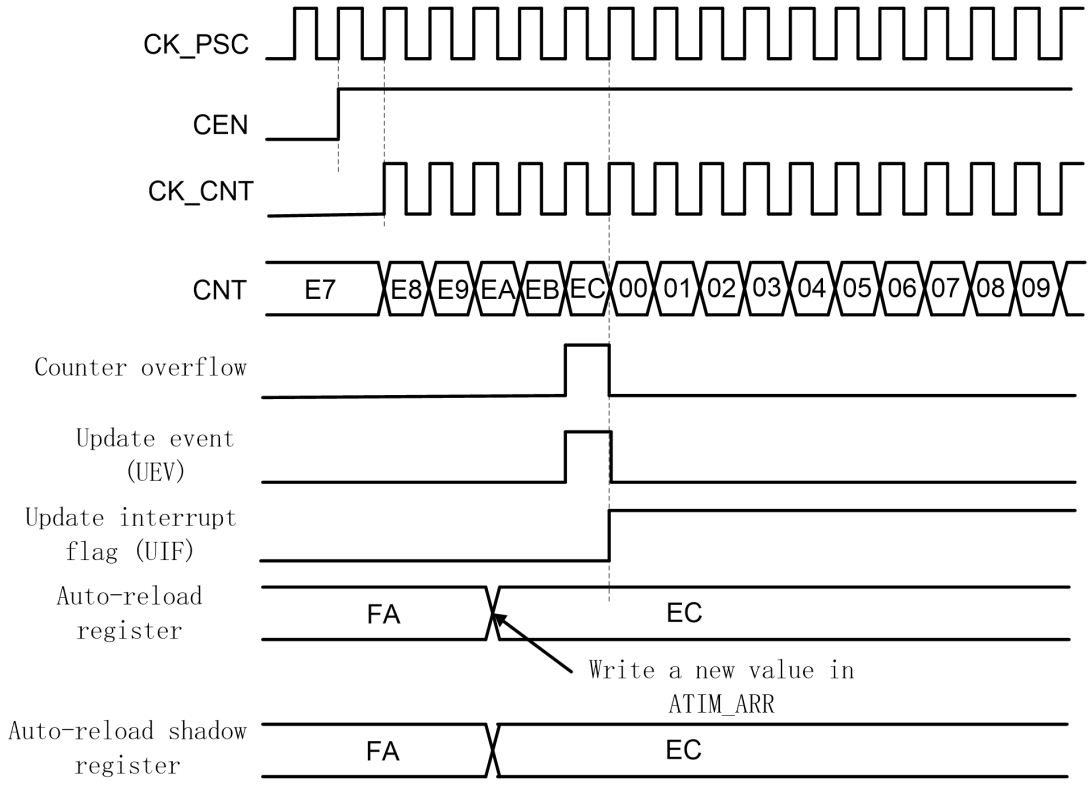


图 24-6ARPE=0（ATIM_ARR 没有预装载）时的更新事件

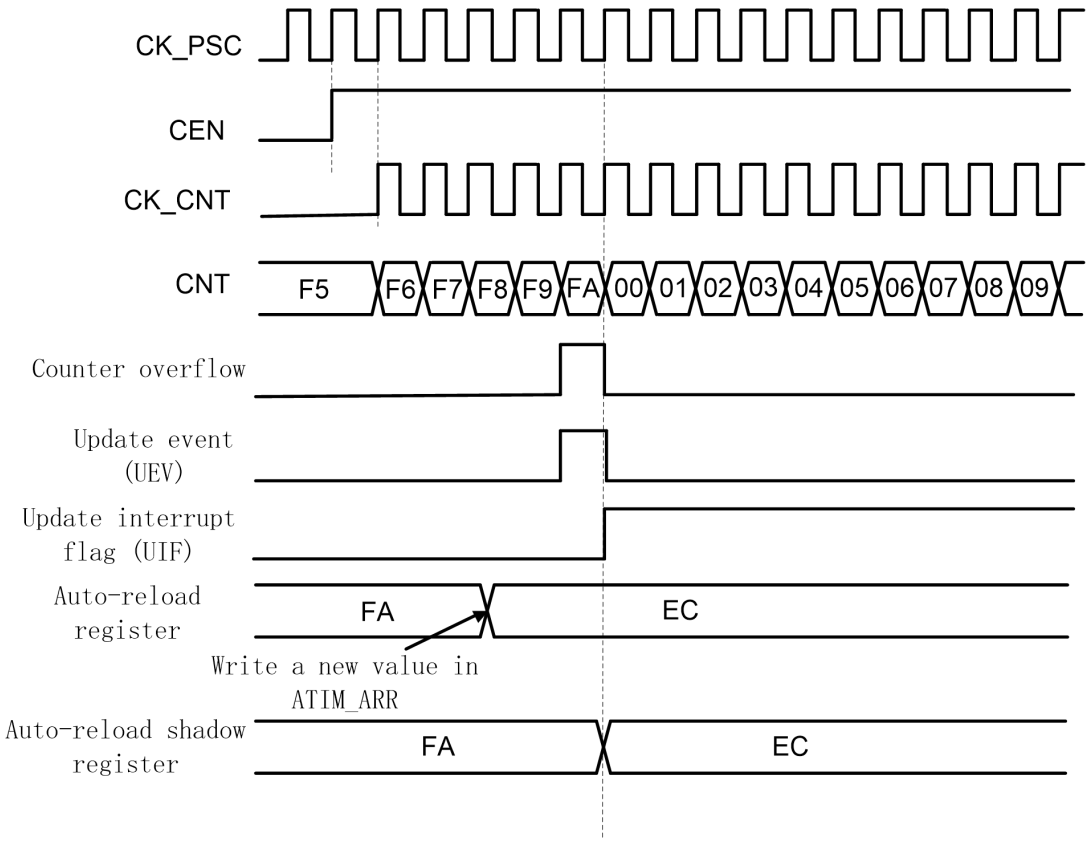


图 24-7ARPE=1（ATIM_ARR 预装载）时的更新事件

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

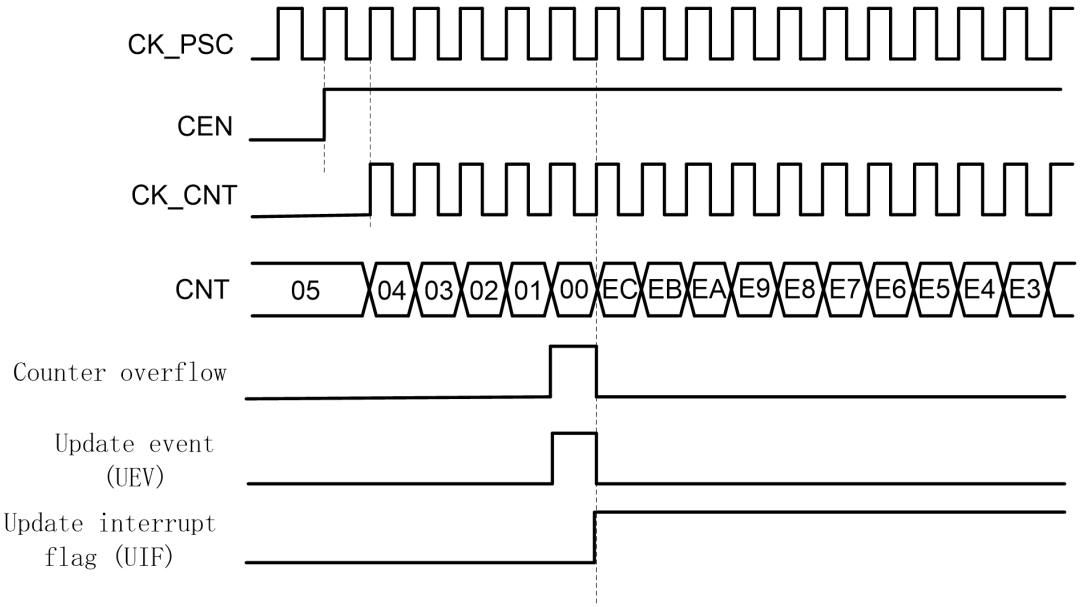


图 24-8 向下计数，内部时钟不分频

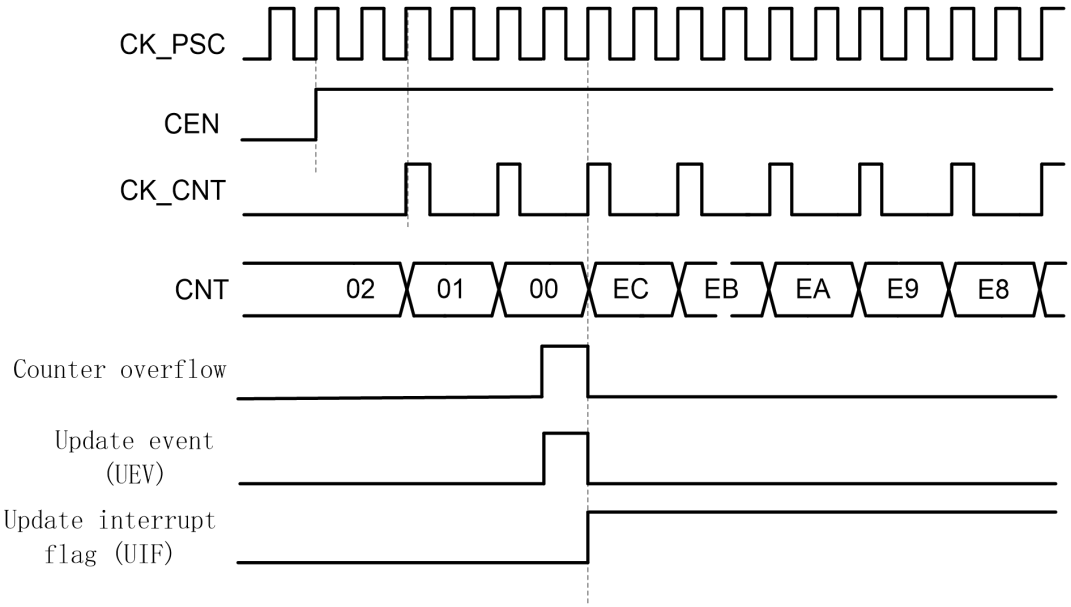


图 24-9 向下计数，内部时钟 2 分频

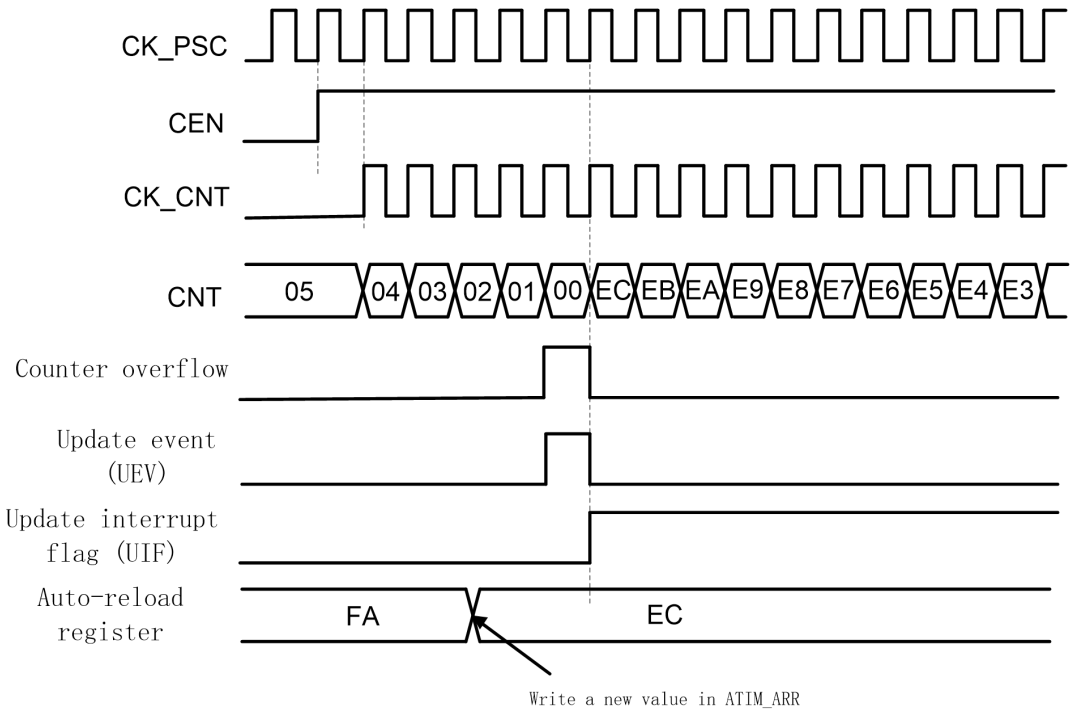


图 24-10 向下计数，内部时钟 2 分频

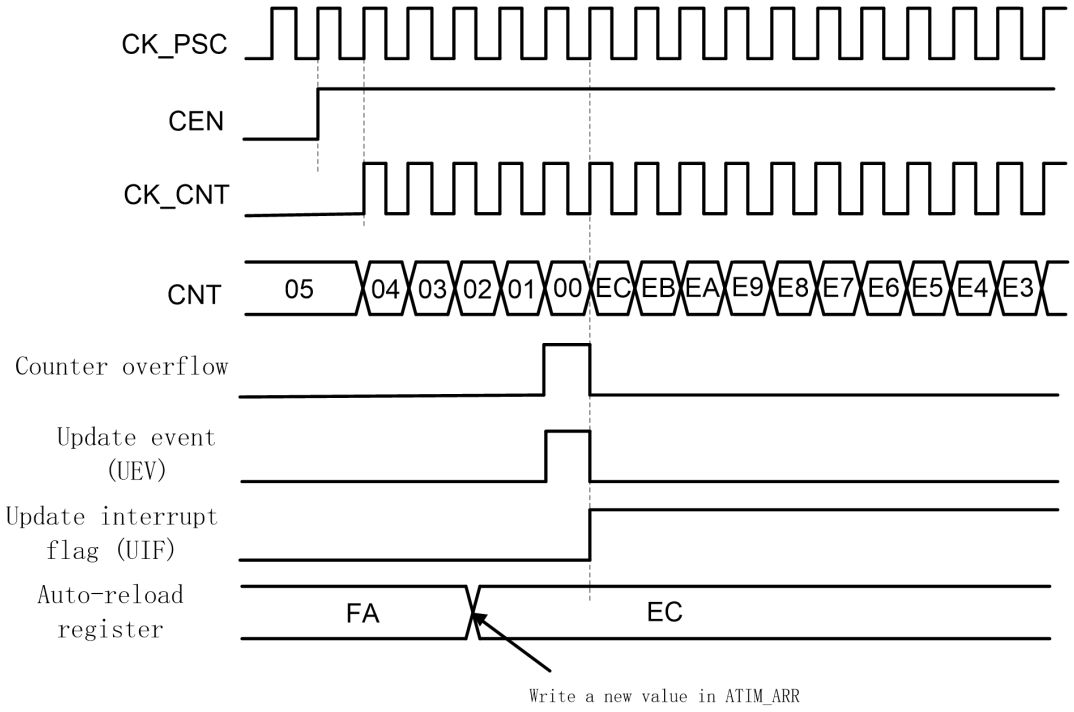


图 24-11 向下计数，不使用重复计数时的更新事件

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。

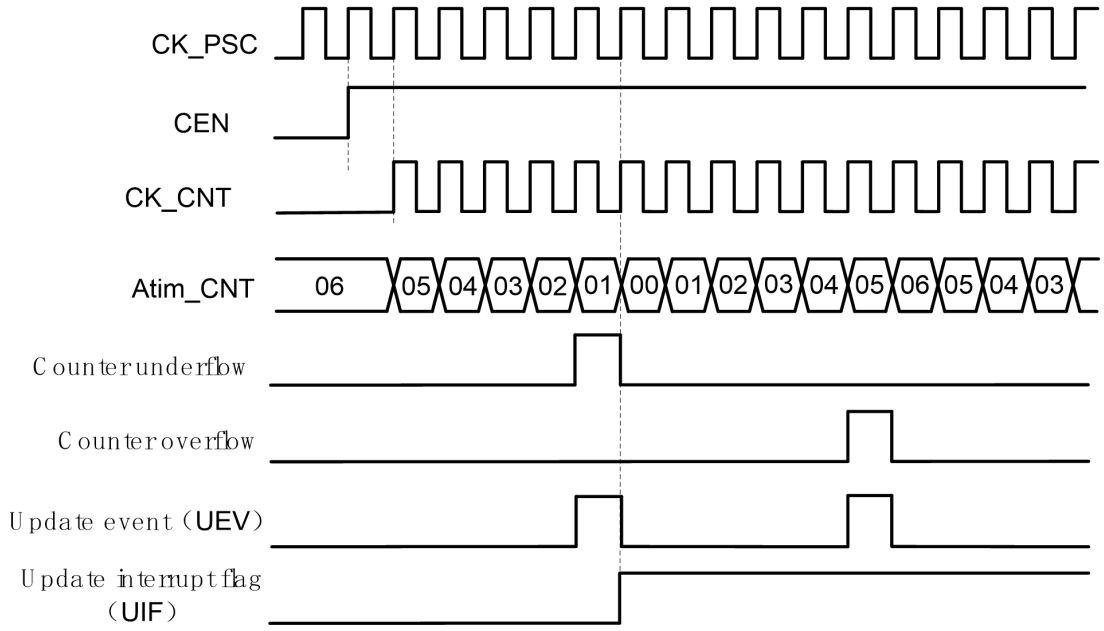


图 24-12 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

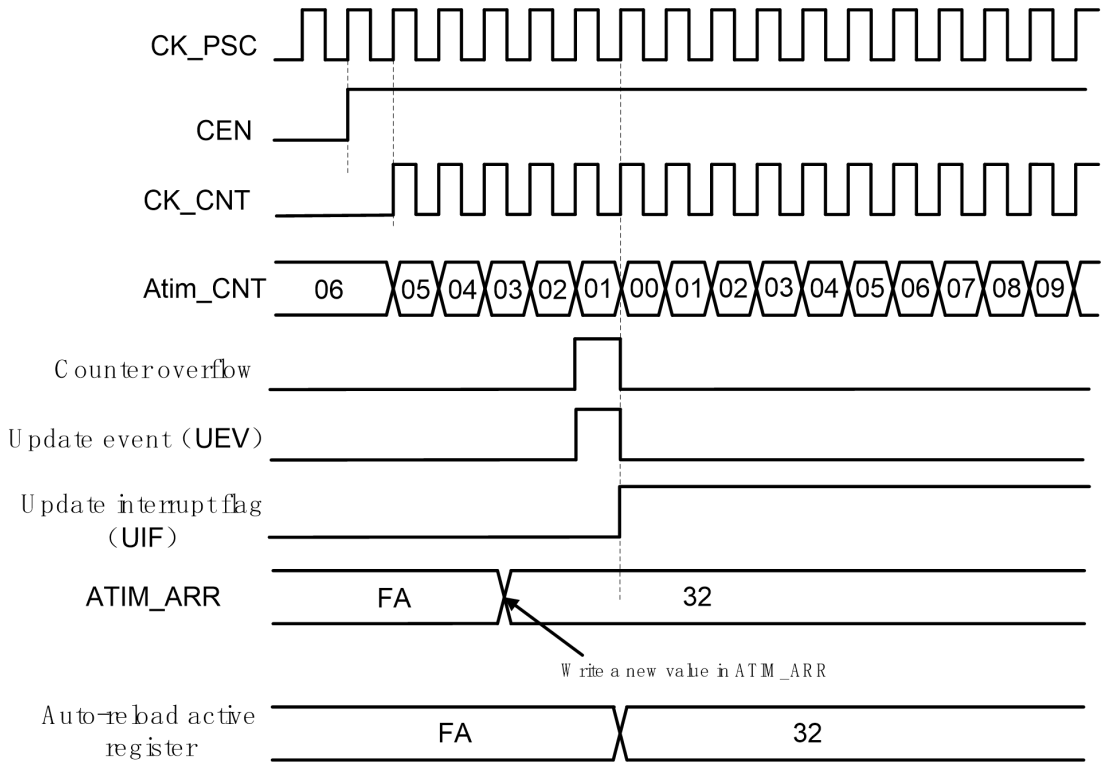


图 24-13 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

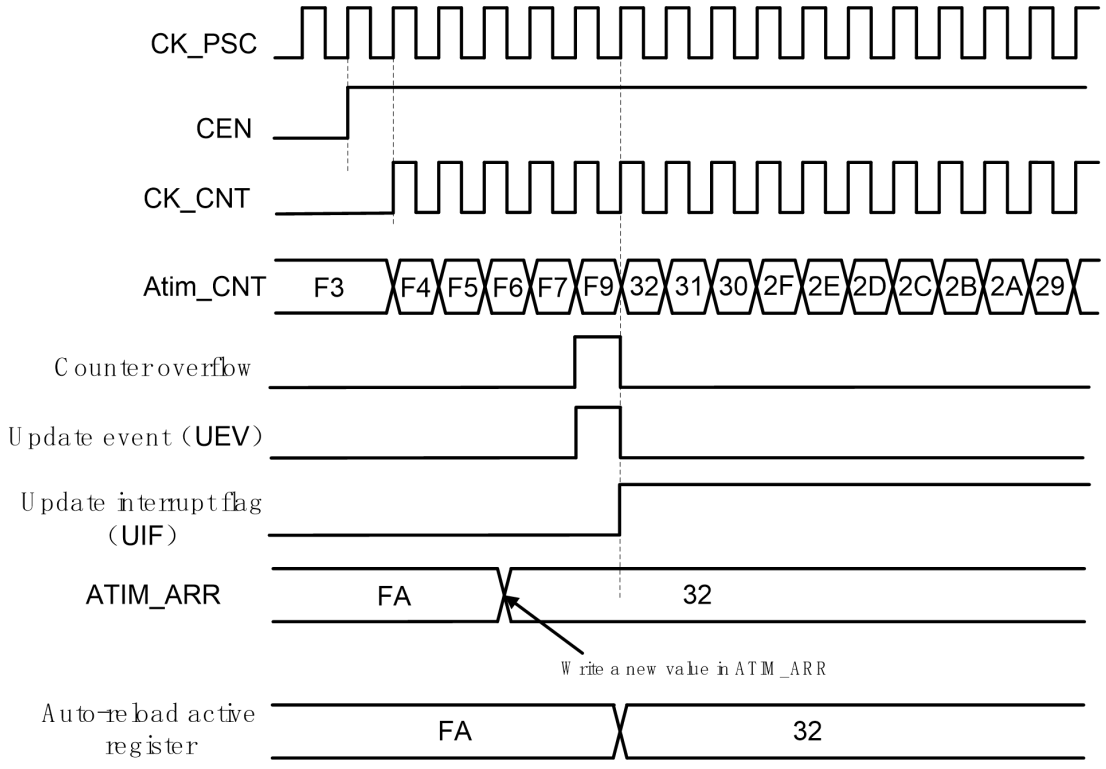


图 24-14 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

24.5.3 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。

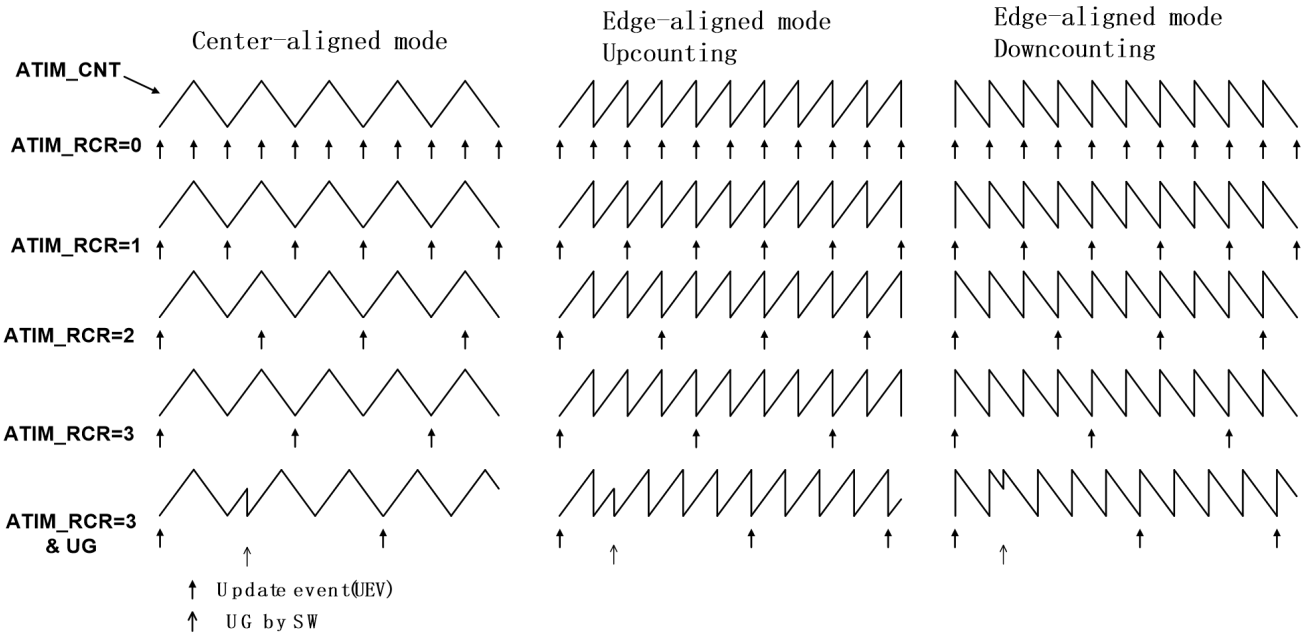


图 24-15 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

24.5.4 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

24.5.5 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

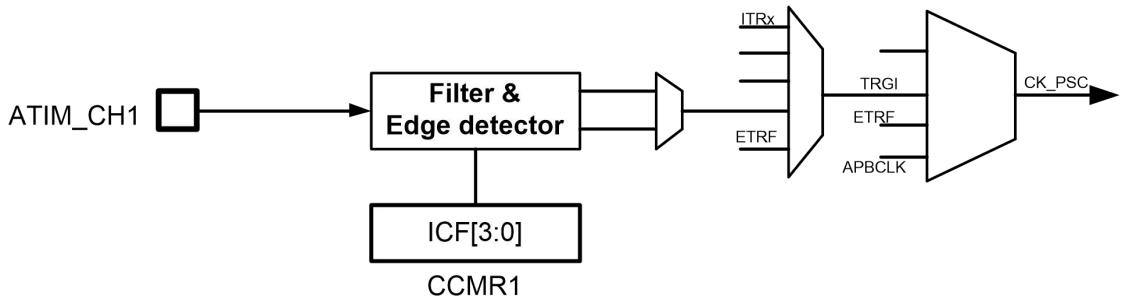


图 24-16 ATIM 时钟源框图

24.5.5.1 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。

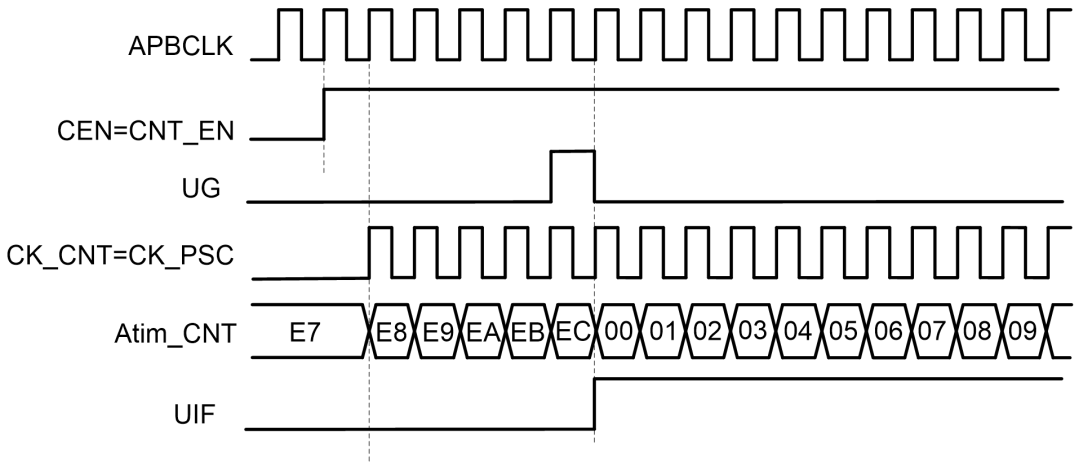


图 24-17 内部时钟源模式，时钟分频因子为 1

24.5.5.2 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。

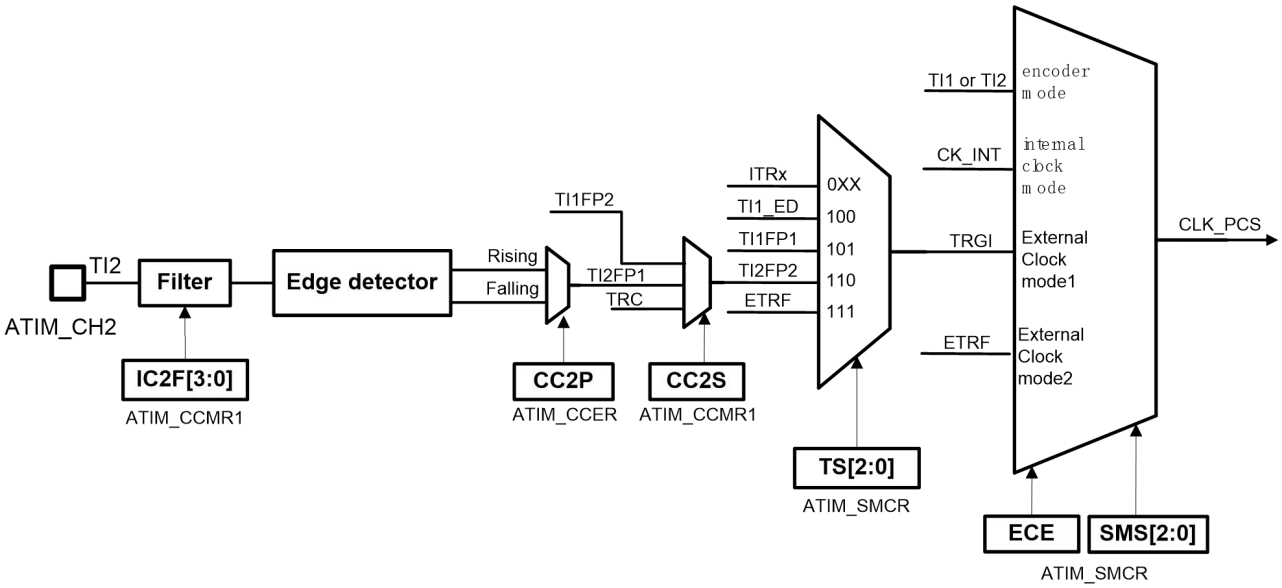


图 24-18TI2 外部时钟连接例子

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

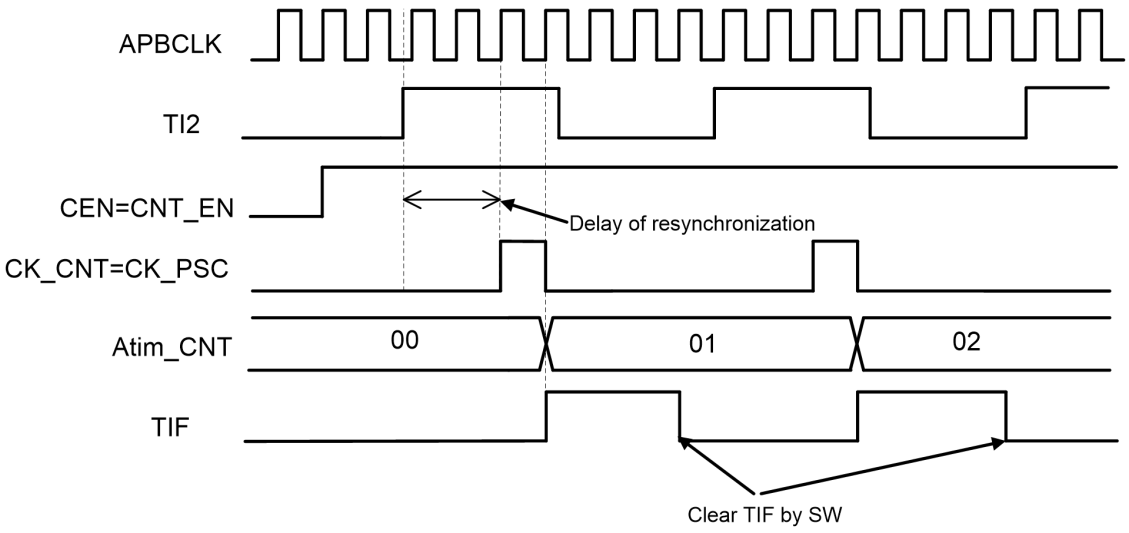


图 24-19 外部时钟模式 1 下的时序

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：

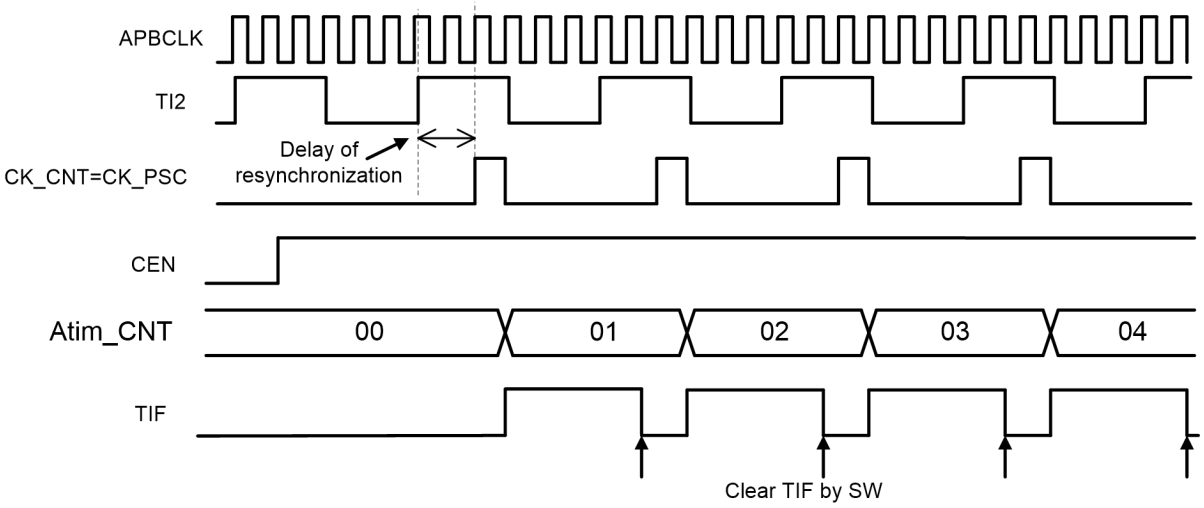


图 24-20 外部时钟模式 1 下的时序

24.5.5.3 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。

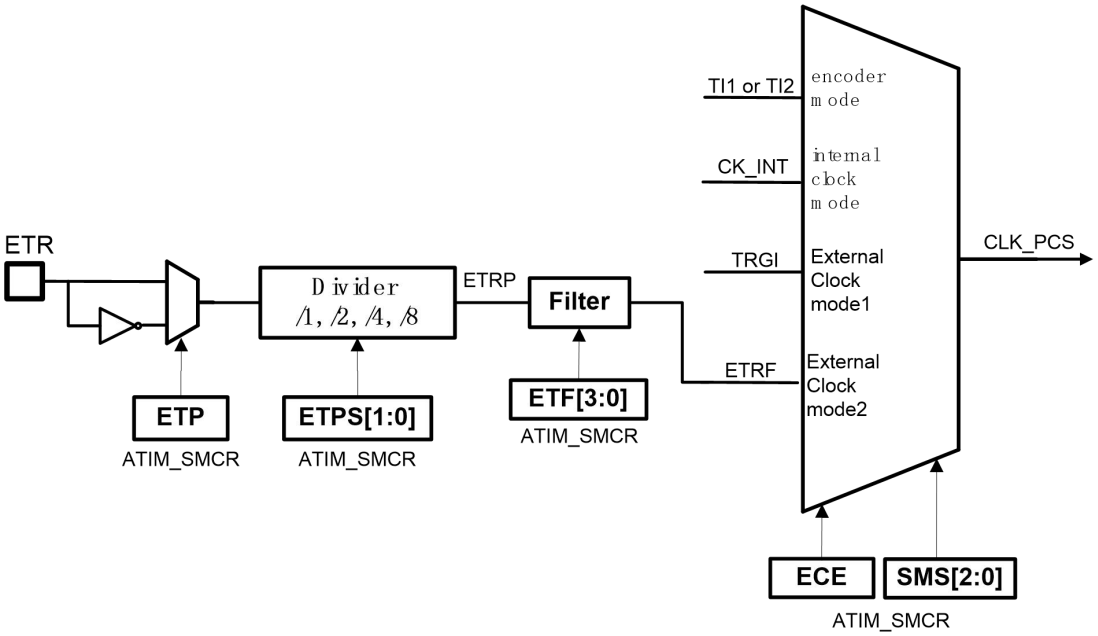


图 24-21 外部触发输入框图

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。

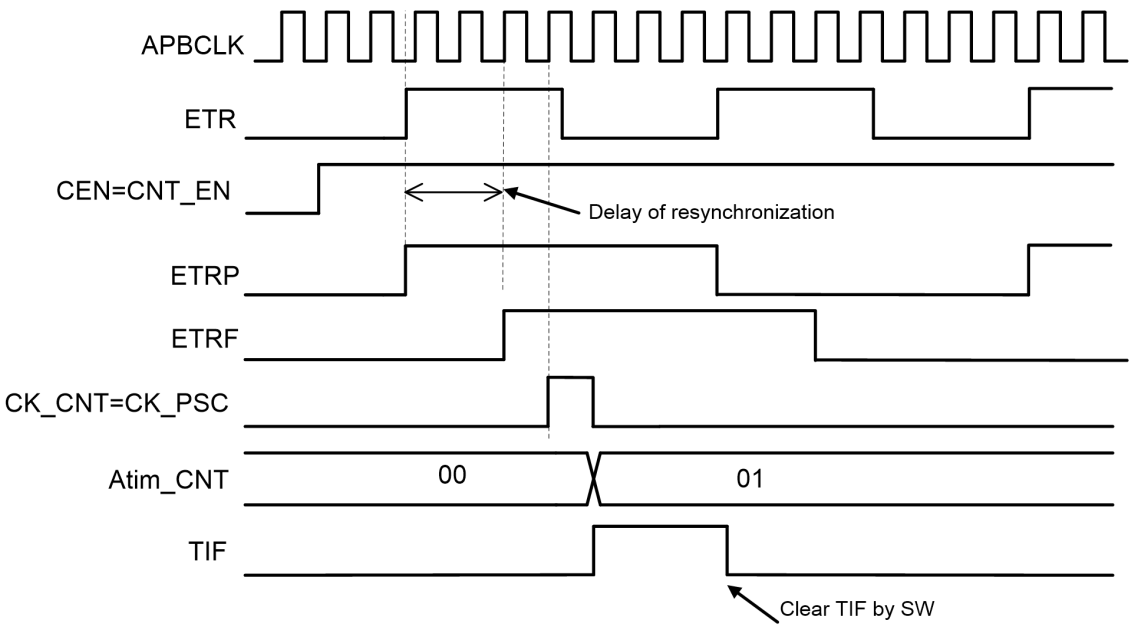


图 24-22 外部时钟模式 2 下的时序 1

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：

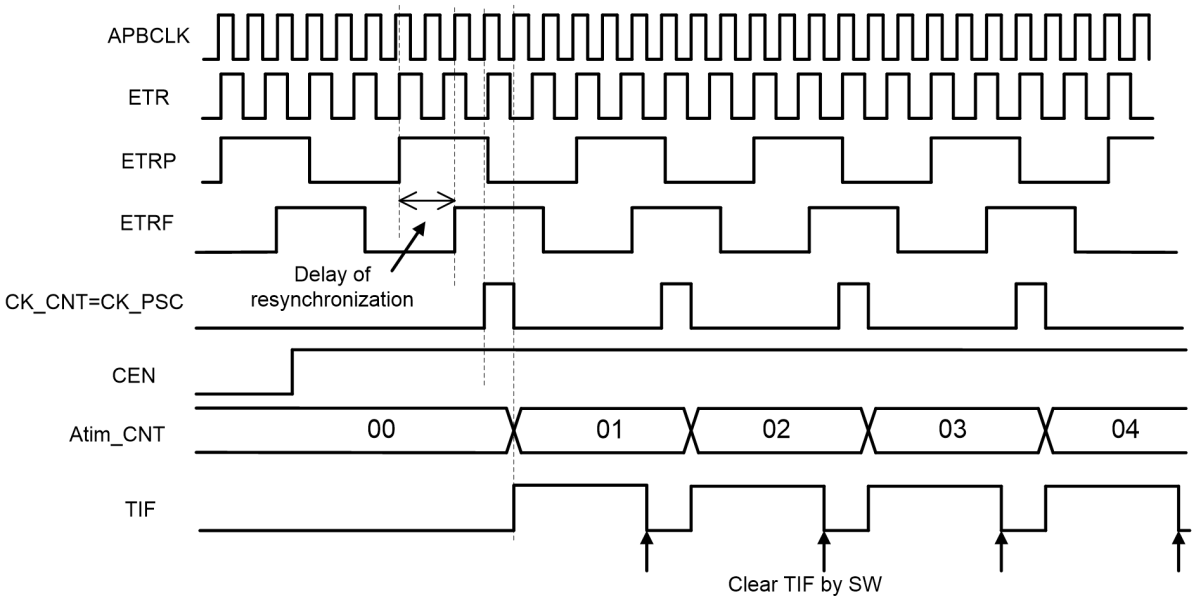


图 24-23 外部时钟模式 2 下的时序 2

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。

24.5.6 内部触发信号（ITRx）

ATIM支持4个ITR输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将TS配置为000~011用于选择ITR0~ITR3，并将CCxS配置为11，即将TRC选为捕捉信号。

每个ITR输入支持4个内部信号扩展，由ITRxSEL寄存器配置。输入信号源参考下表：

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIM	TAU10_TRGO	TAU11_TRGO	COMP1	COMP2

24.5.7 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。

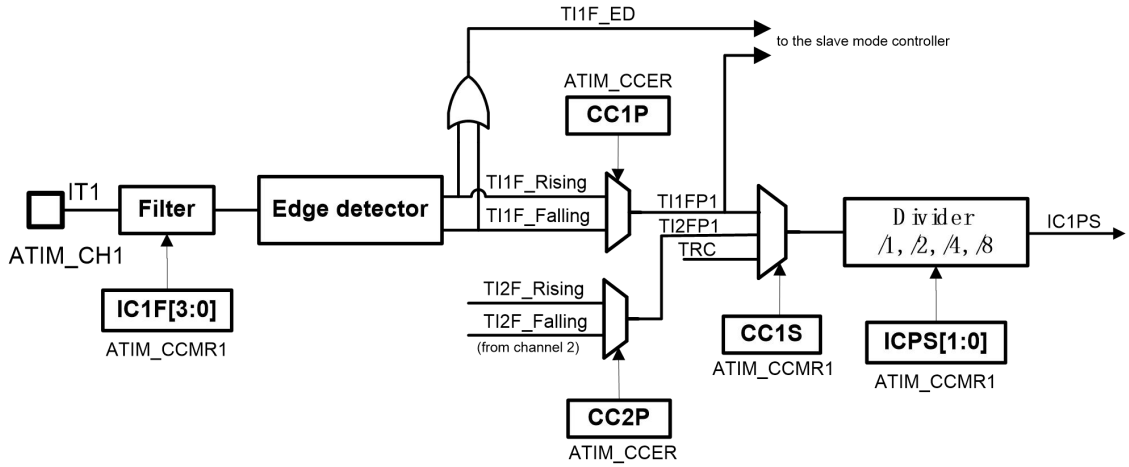


图 24-24 捕获/比较通道(通道 1 输入部分)

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。

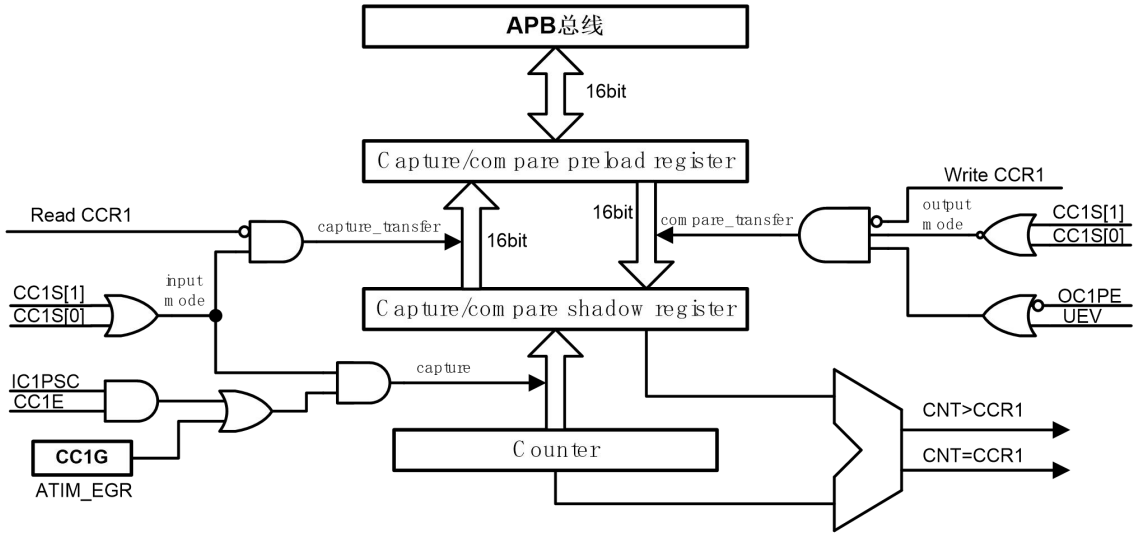


图 24-25 捕获/比较通道 1 的主电路

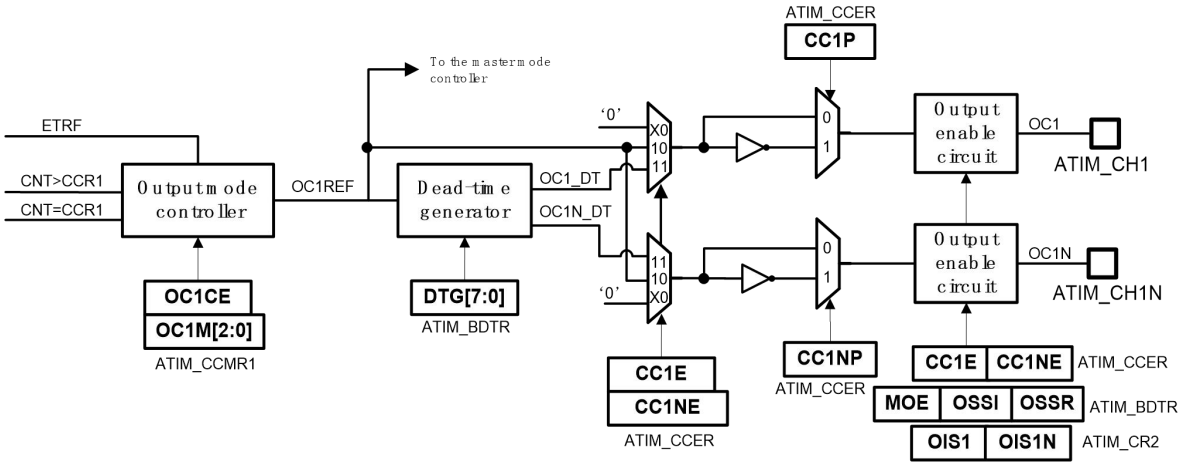


图 24-26 捕获/比较通道的输出部分(通道 1 至 3)

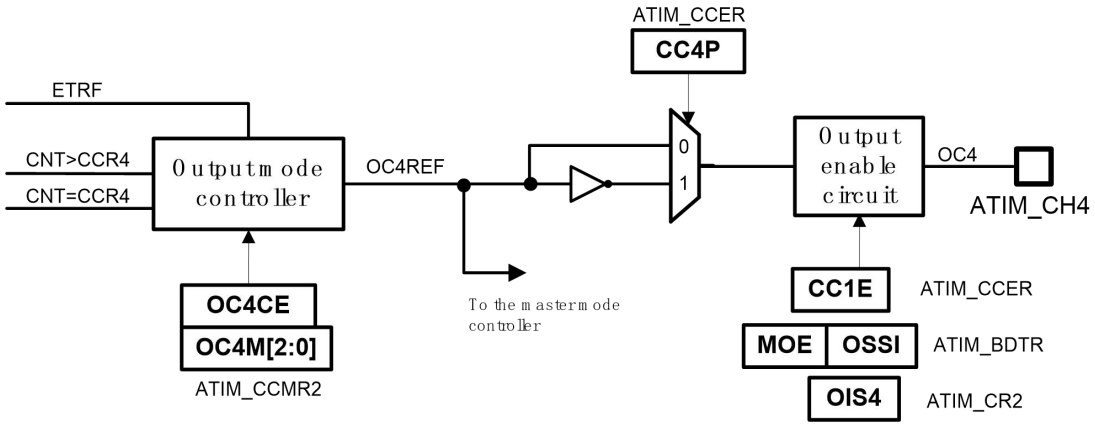


图 24-27 捕获/比较通道的输出部分(通道 4)

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。

24.5.8 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

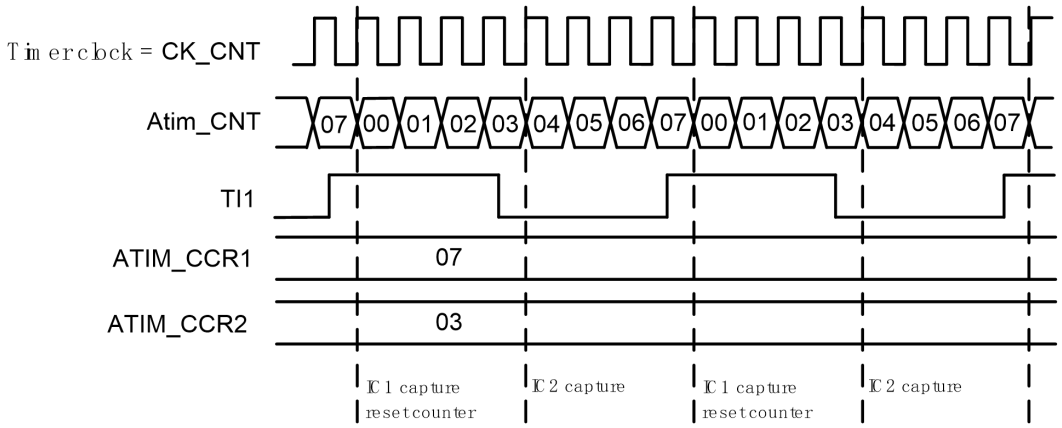


图 24-28 PWM 输入捕获模式时序

若想实现PWM输入捕获功能，需进行如下设置：

- 在GPIO模块中，配置相应管脚为ATIM_CH1功能
- 关闭通道使能，配置ATIM_CCER.CC1E=0，ATIM_CCER.CC2E=0确保之后通道配置成功

- 选择输入通道，两个通道IC1,IC2被映射到同一个TI1输入口，配置ATIM_CCMR1.CC1S=01, ATIM_CCMR1.CC2S=10
- 选择计数有效沿，两个通道IC1,IC2有效沿极性相反，配置ATIM_CCER.CC1P=0，ATIM_CCER.CC2P=1
- 配置输入滤波时间，配置ATIM_CCMR1.IC1F[3:0]，ATIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置ATIM_CCMR1.IC1PS[1:0] ，ATIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置ATIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置ATIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置ATIM_CCER.CC1E=1，ATIM_CCER.CC2E=1

24.5.9 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。

24.5.10 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

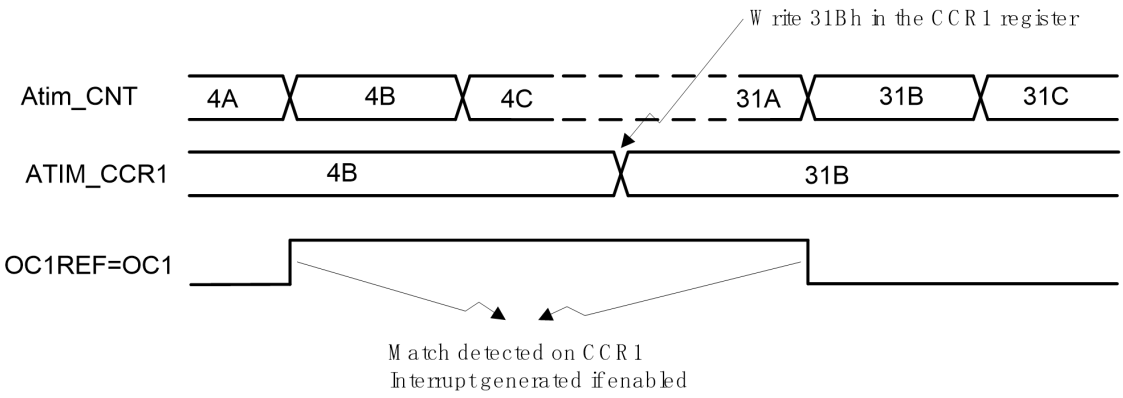


图 24-29 输出比较模式，翻转 OC1

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。

24.5.11 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

24.5.11.1 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。

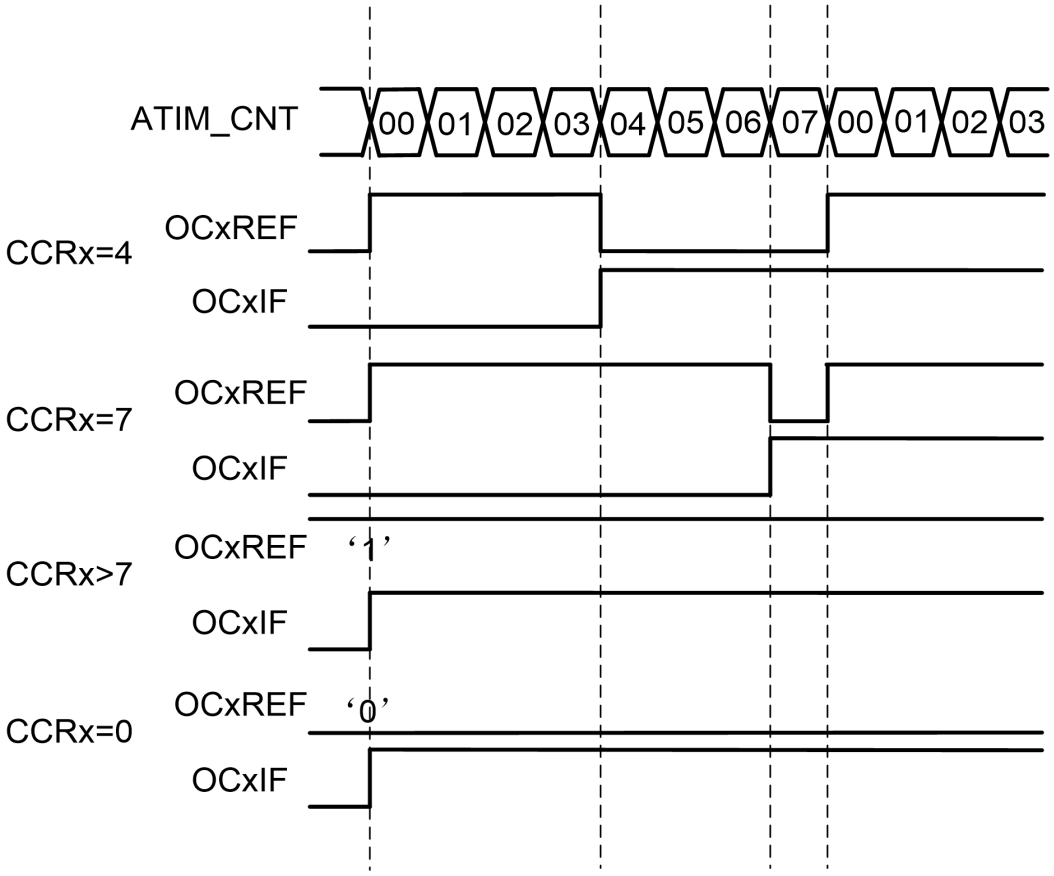


图24-30边沿对齐的PWM波形(ARR=7, PWM mode 1)

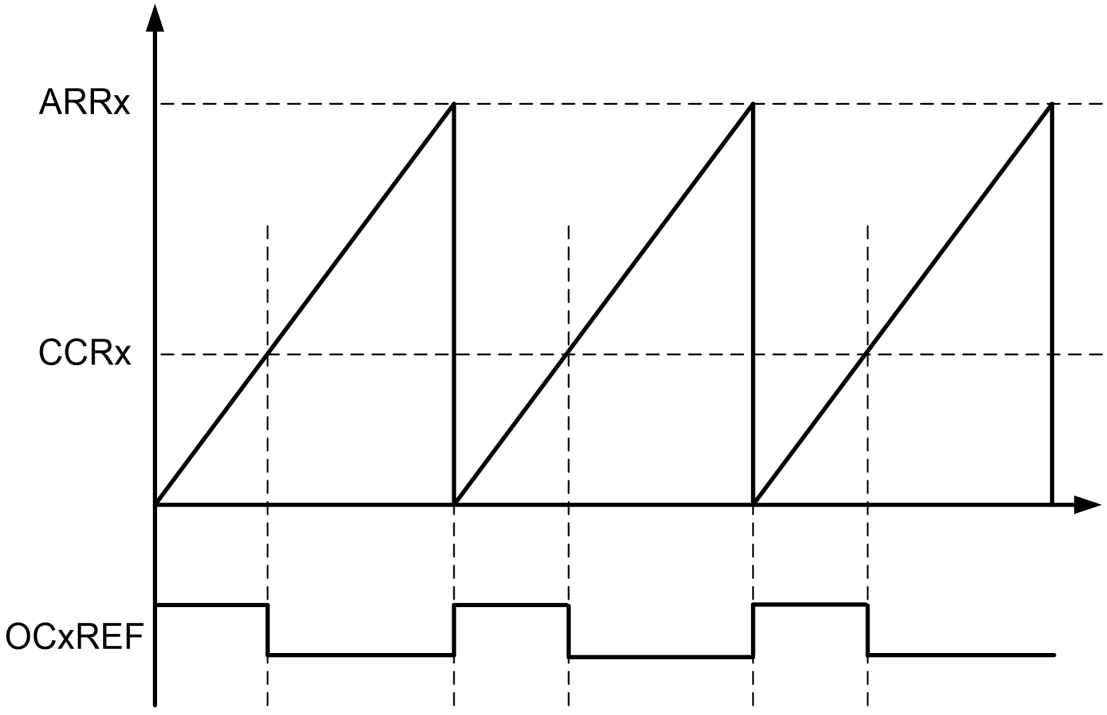


图24-31边沿对齐的PWM波形(PWM mode 1)

在向下计数时，OCxREF电平高低定义与向上计数时相同。

24.5.11.2 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

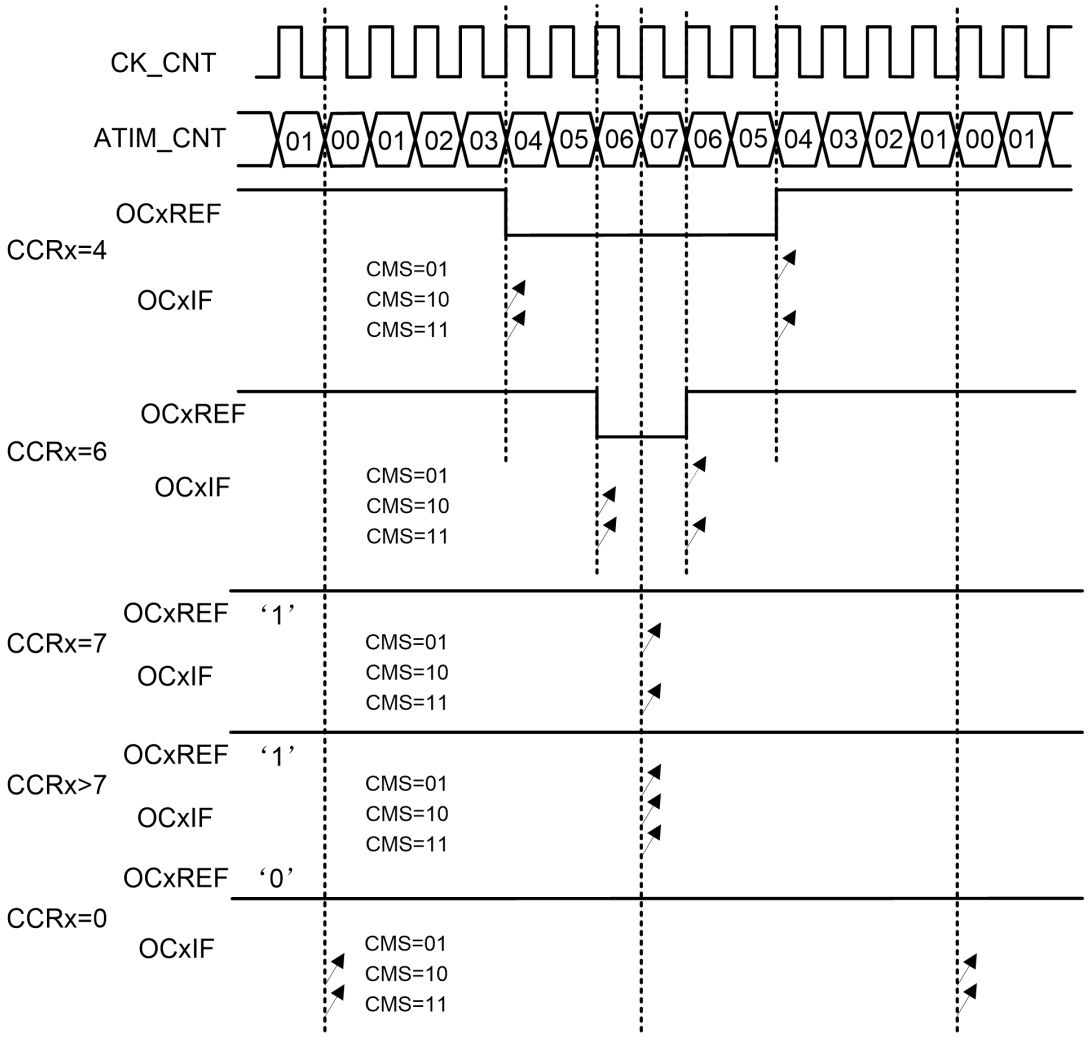


图24-32中央对齐的PWM波形(APR=7, PWM mode 1)

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。

24.5.12 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

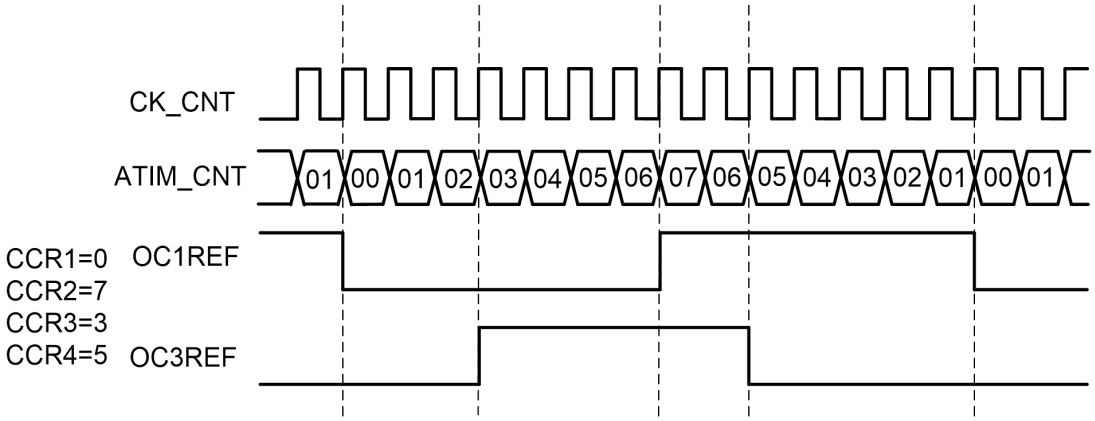


图24- 33非对称PWM输出

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。

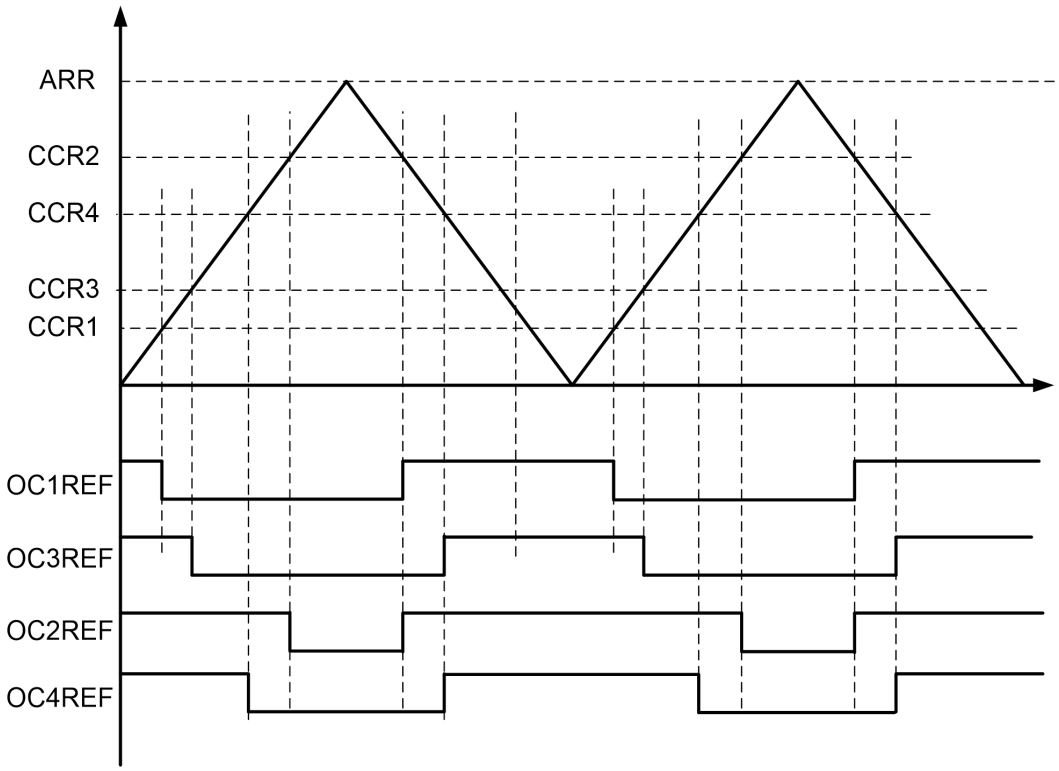


图24-34通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 1）

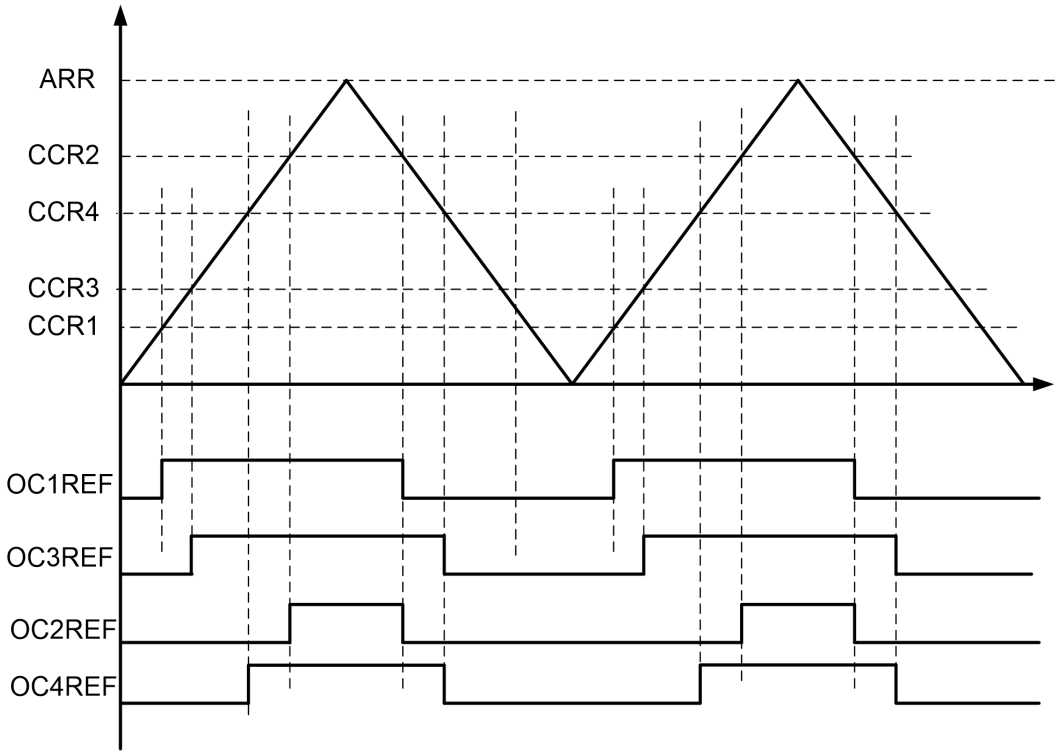


图24-35通道1/3非对称PWM输出，通道2/4常规输出（all PWM mode 2）

24.5.13 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

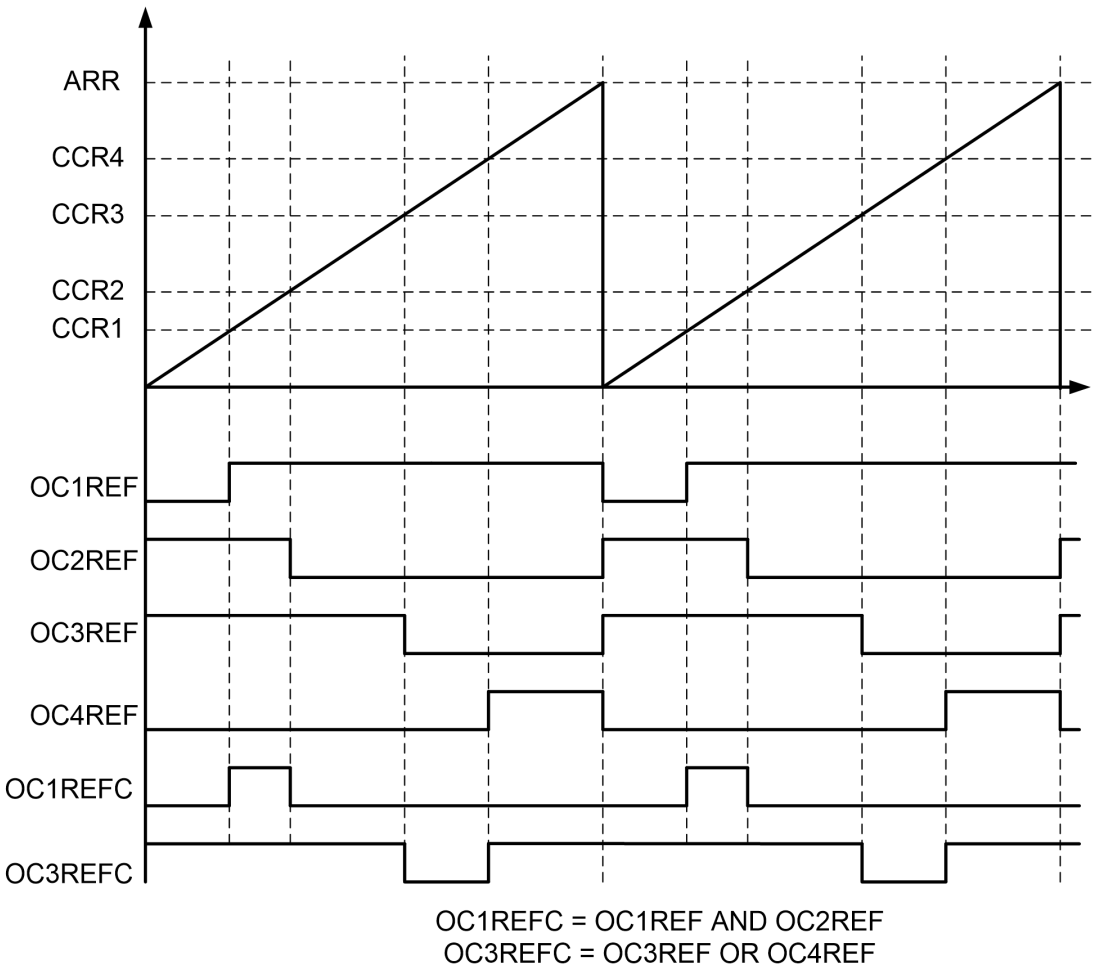


图24-36通道1/3组合PWM输出

24.5.14 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。

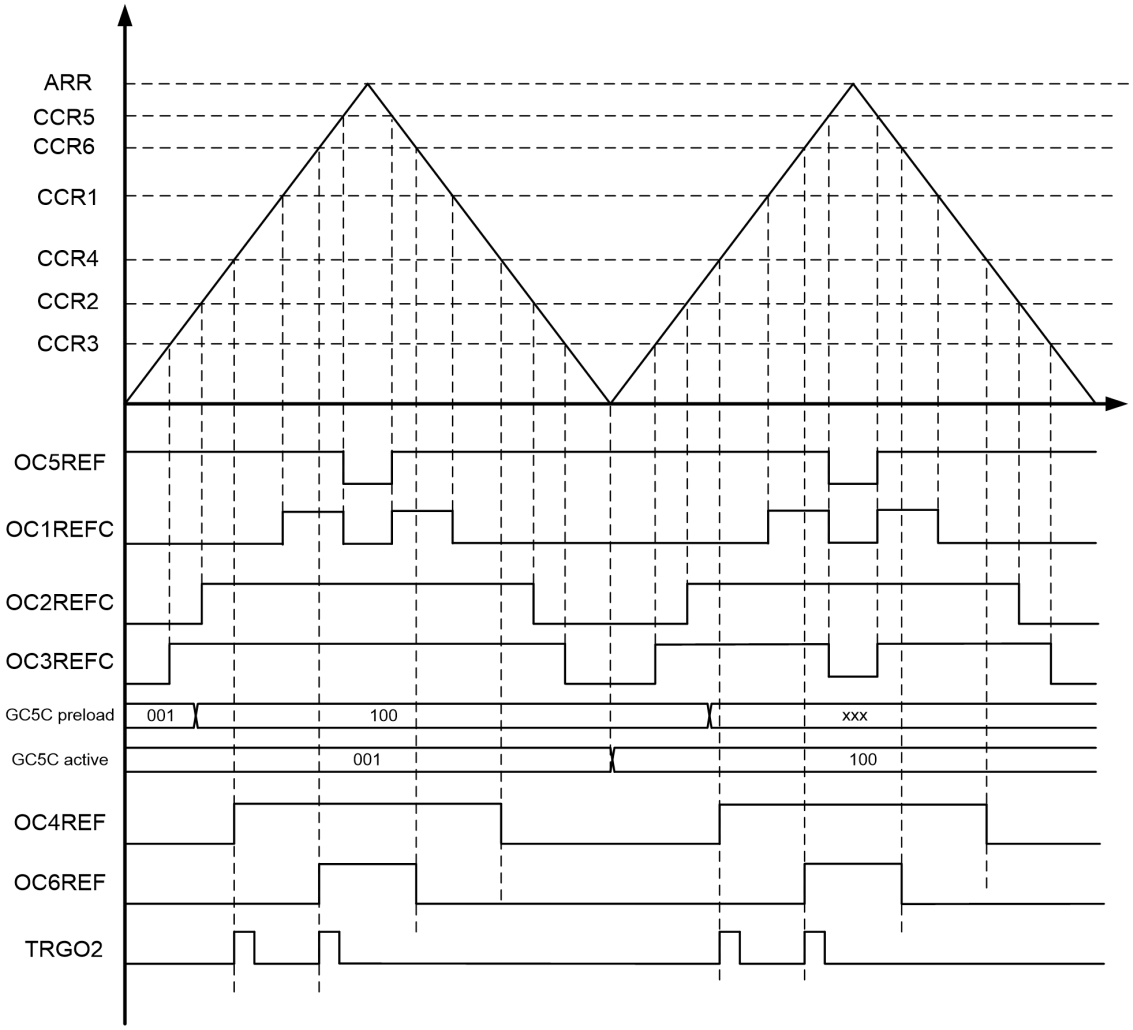


图24-37组合三相PWM输出

24.5.15 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。

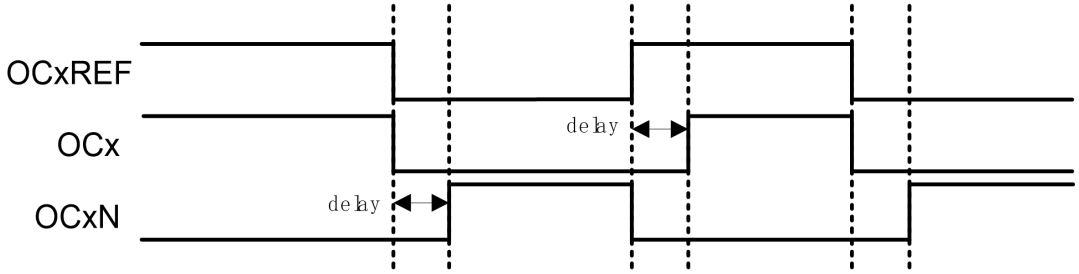


图24-38带死区插入的互补输出

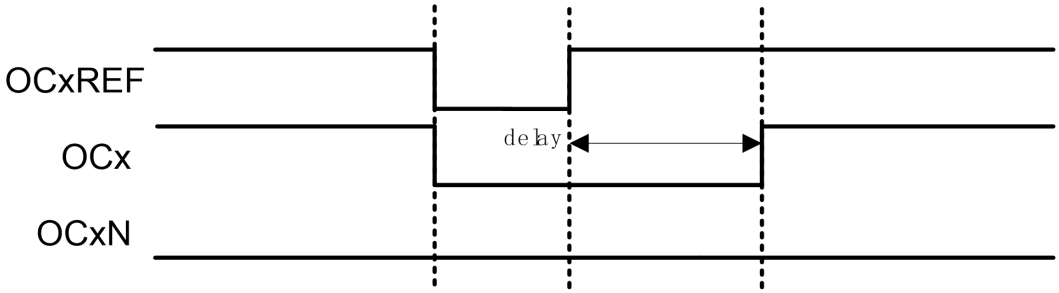


图24-39死区波形延迟大于负脉冲

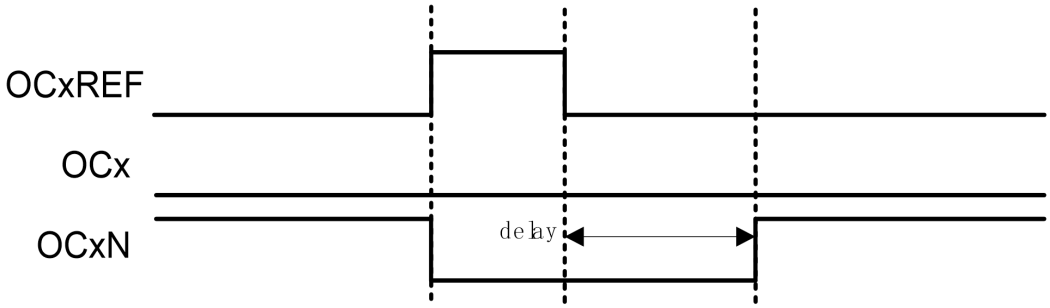


图24-40死区波形延迟大于正脉冲

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。

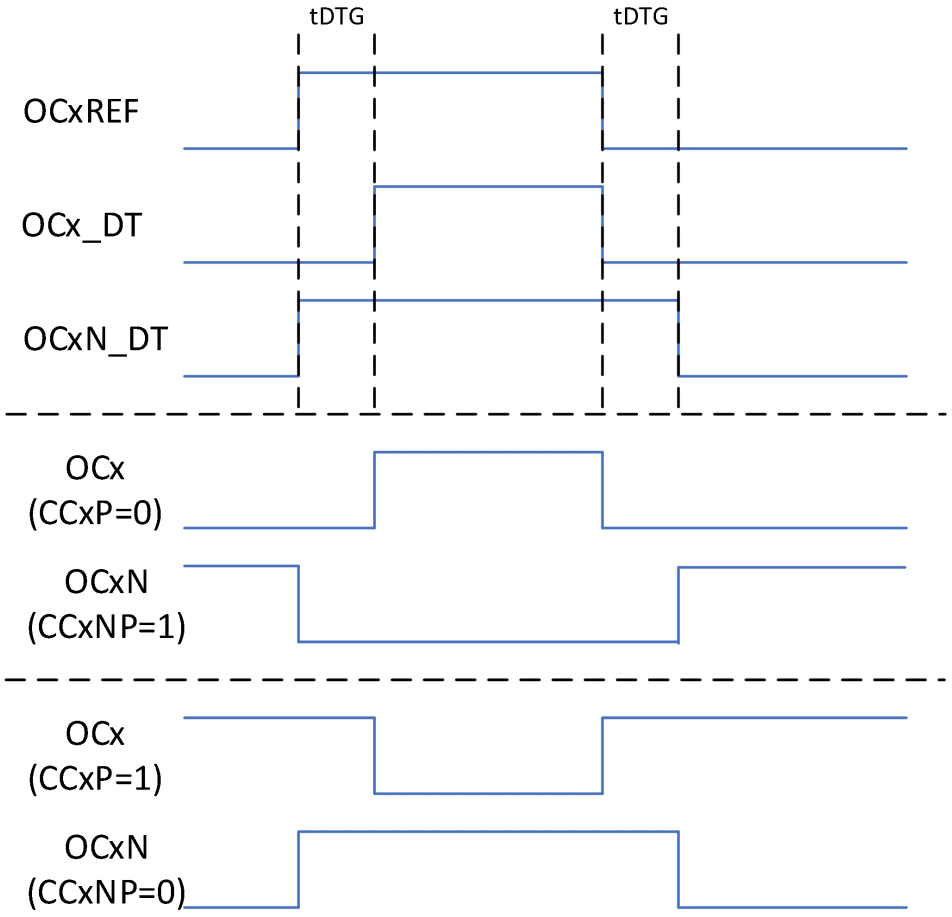


图24-41死区插入后的互补输出波形与CCxP和CCxNP的关系

24.5.16 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。

24.5.16.1 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：

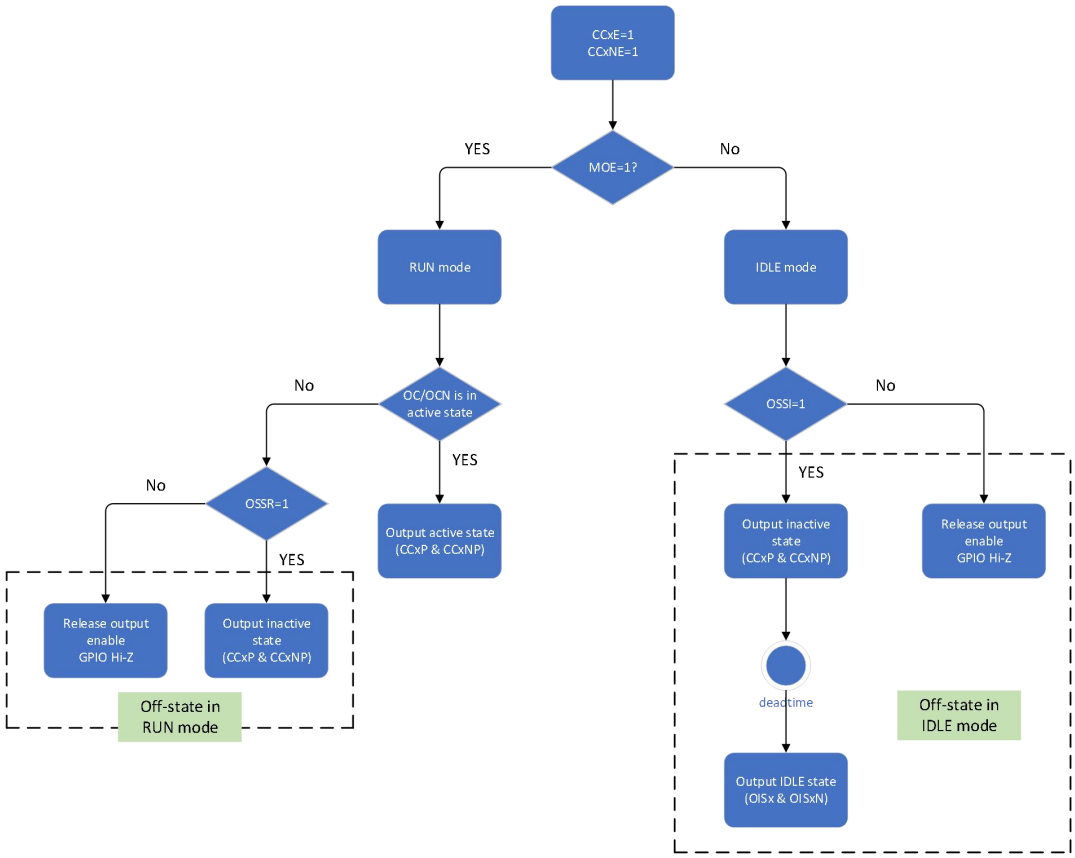


图24-42模式与输出状态术语

24.5.16.2 刹车逻辑

刹车输入控制逻辑如下图所示：

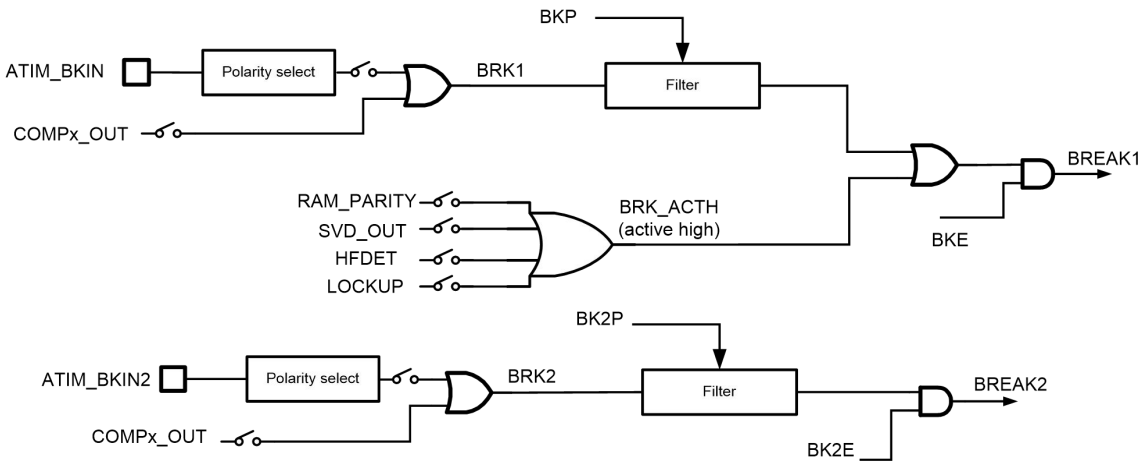


图24-43刹车输入结构

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。

24.5.16.3 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。

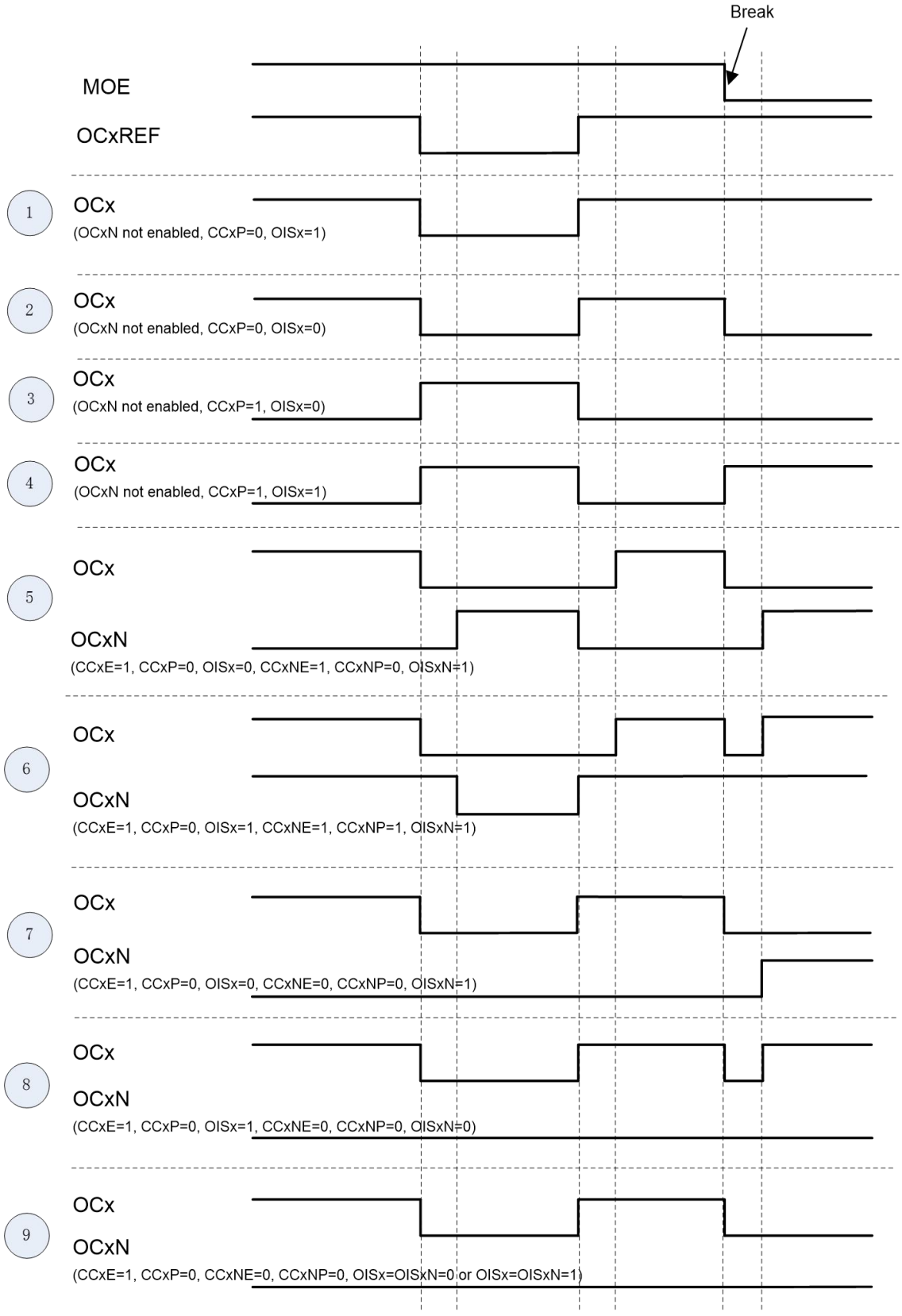


图24-44响应刹车的输出

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 24-6 刹车行为

24.5.17 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		

24.5.18 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。

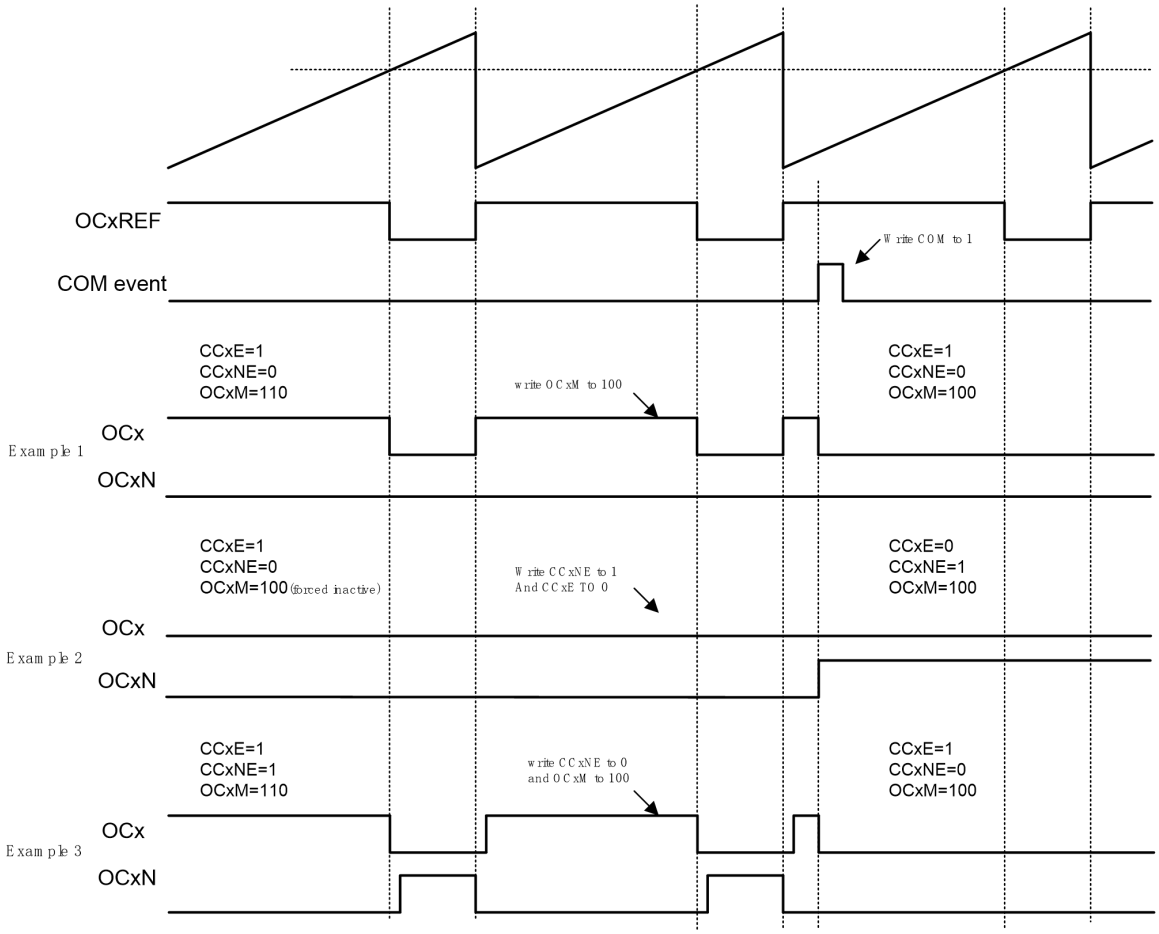


图24-45产生六步PWM，使用COM的例子(OSSR=1)

24.5.19 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

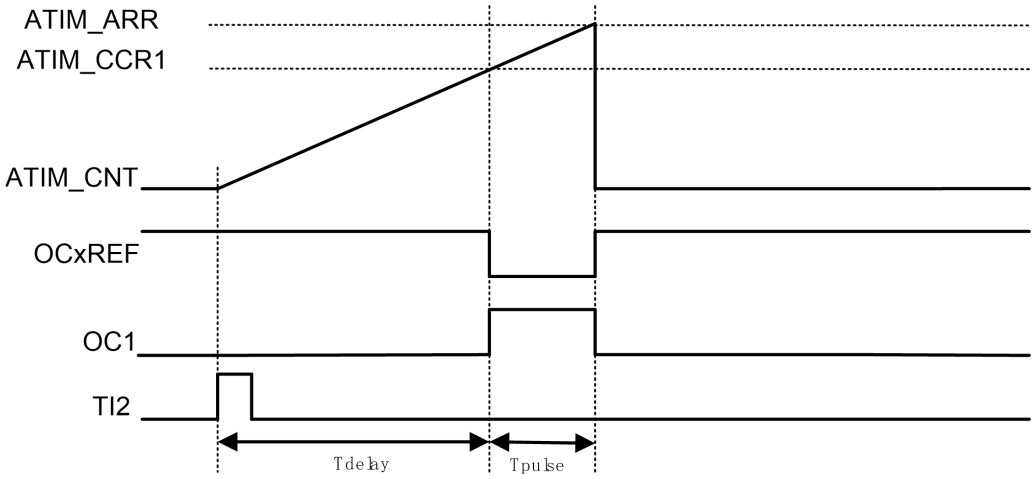


图24-46单脉冲模式的例子

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

24.5.20 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。

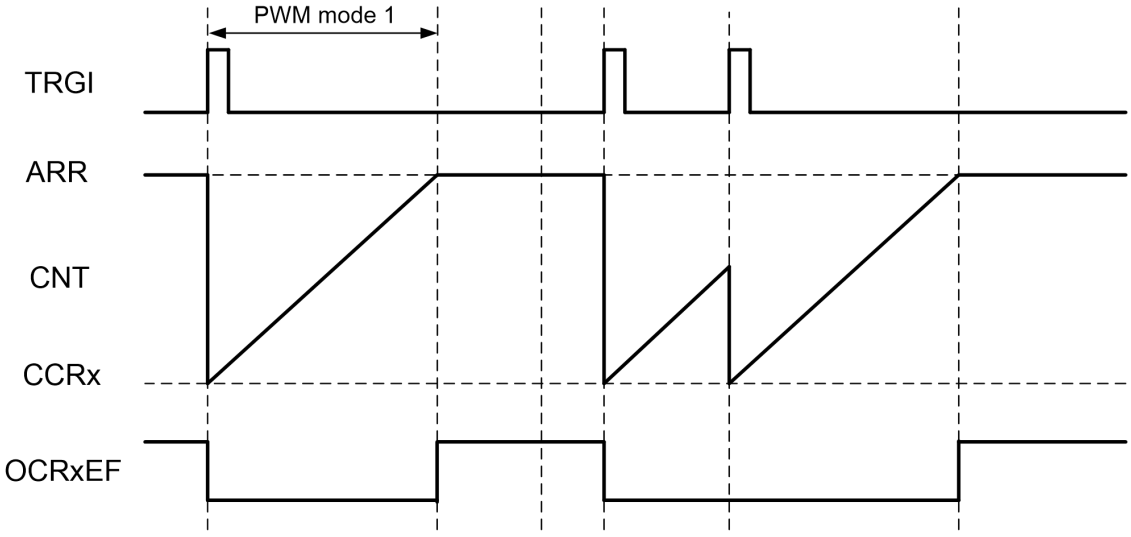


图24-47重触发OPM模式1，向上计数，CCR=0

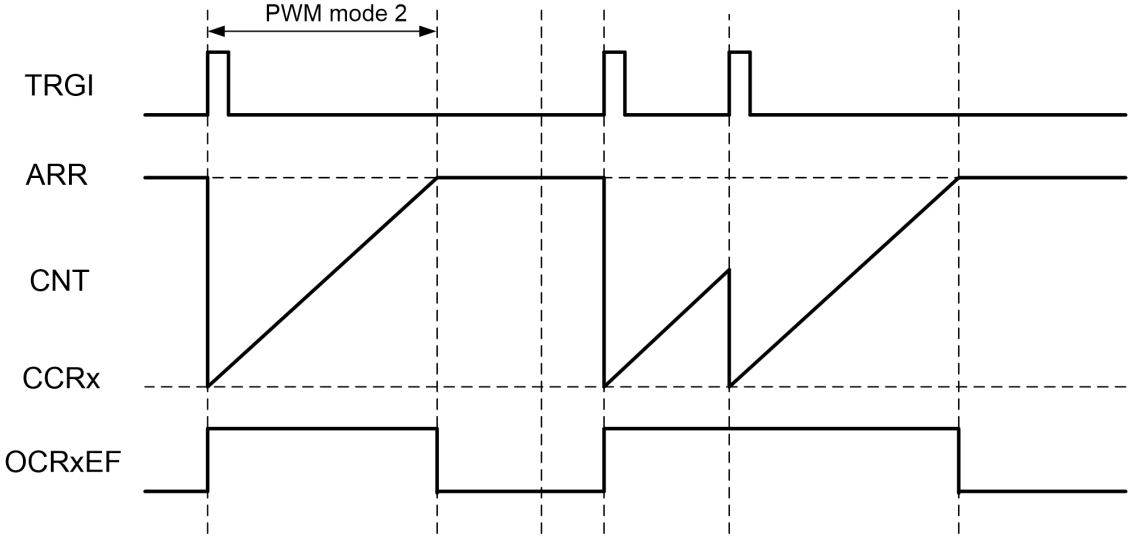


图24-48重触发OPM模式2，向上计数，CCR=0

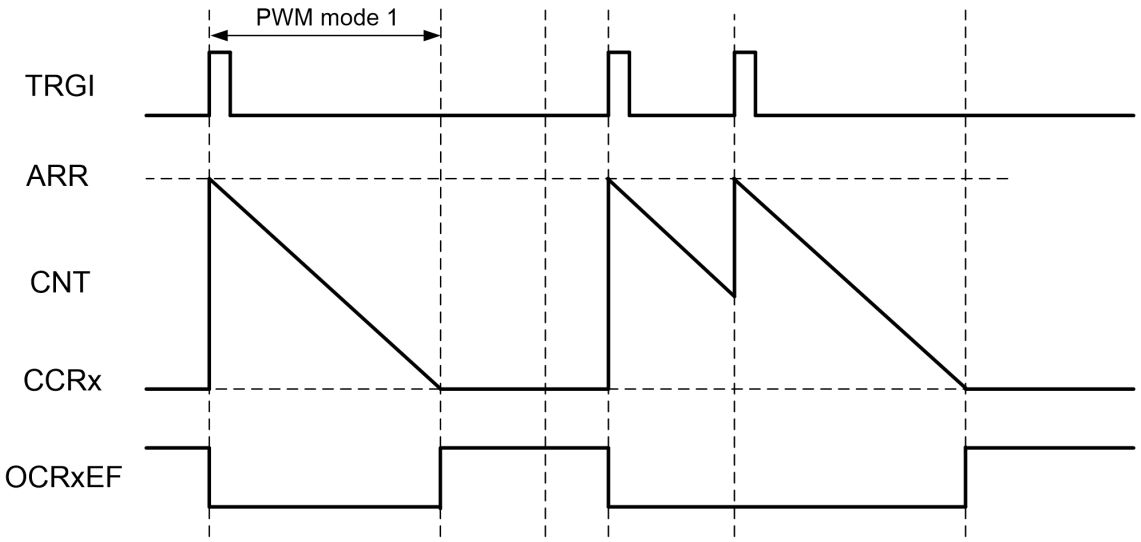


图24-49重触发OPM模式1，向下计数，CCR=0

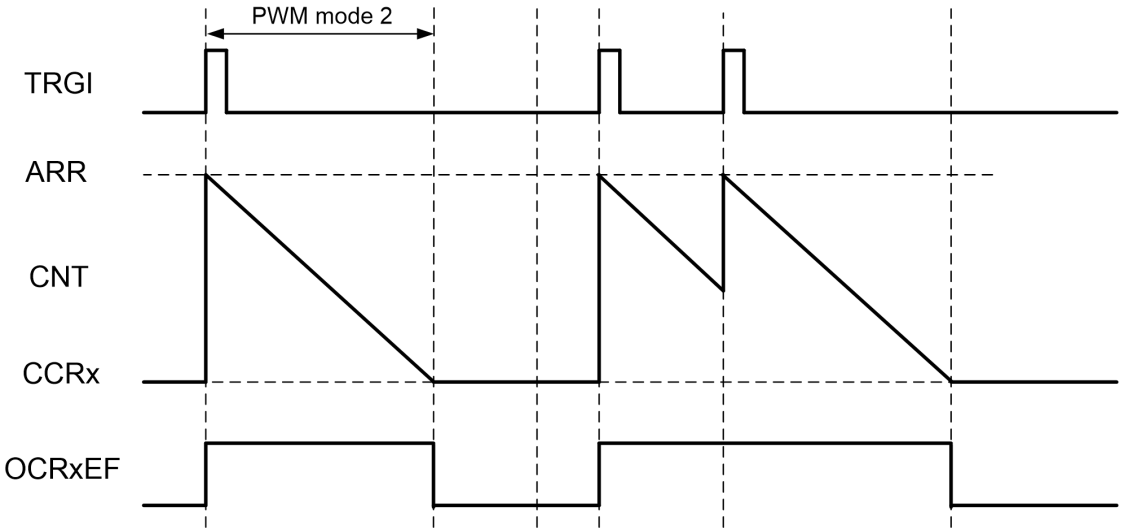


图24-50重触发OPM模式2，向下计数，CCR=0

24.5.21 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。

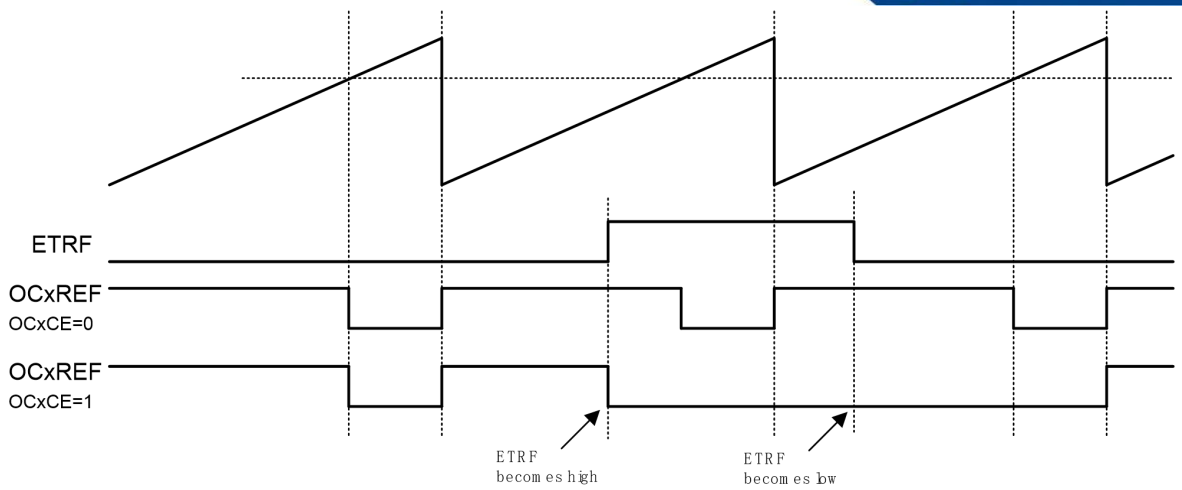


图24-51 ETR信号清除ATIM的OCxREF

24.5.22 编码器接口模式（encoder interface mode）

24.5.22.1 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 x2 mode	0001	高	递减	递增	不计数	不计数
		低	递增	递减	不计数	不计数
仅在TI2处计数 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	1110	高	递减	递增	不计数	不计数

处计数 x1 mode		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 24-7 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。

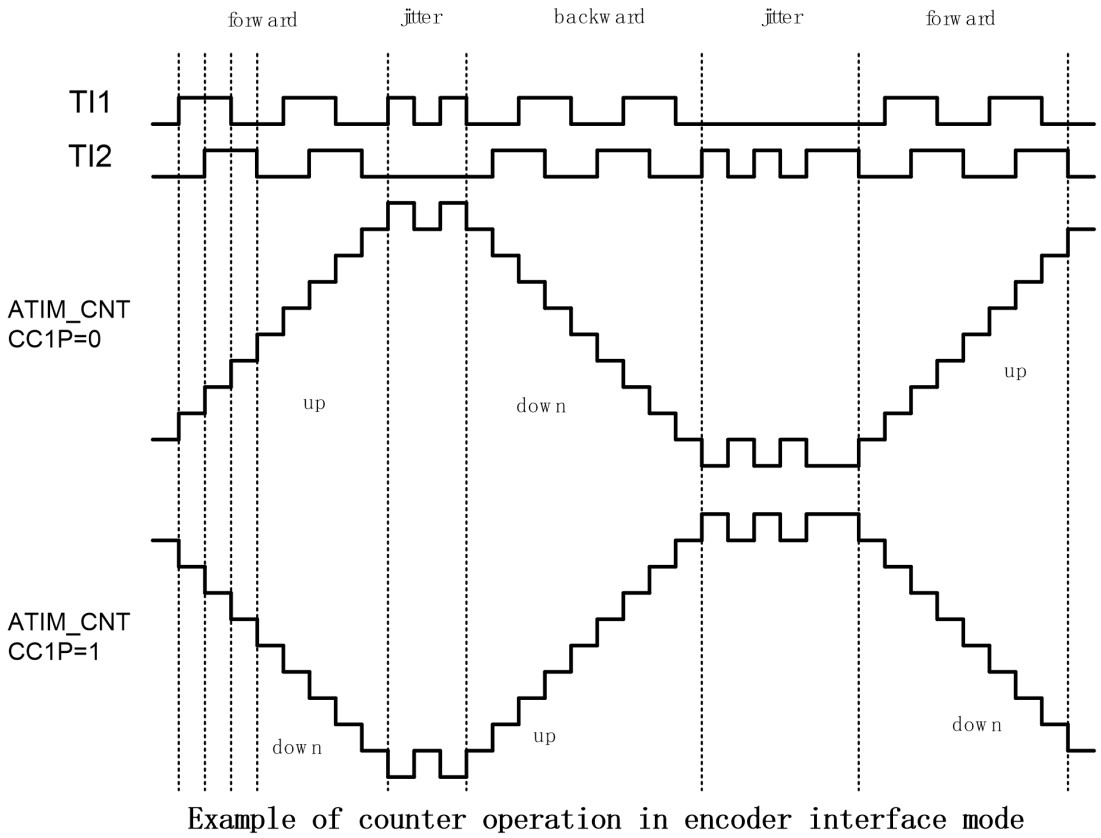


图24-52编码器模式下的计数器操作实例

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）。

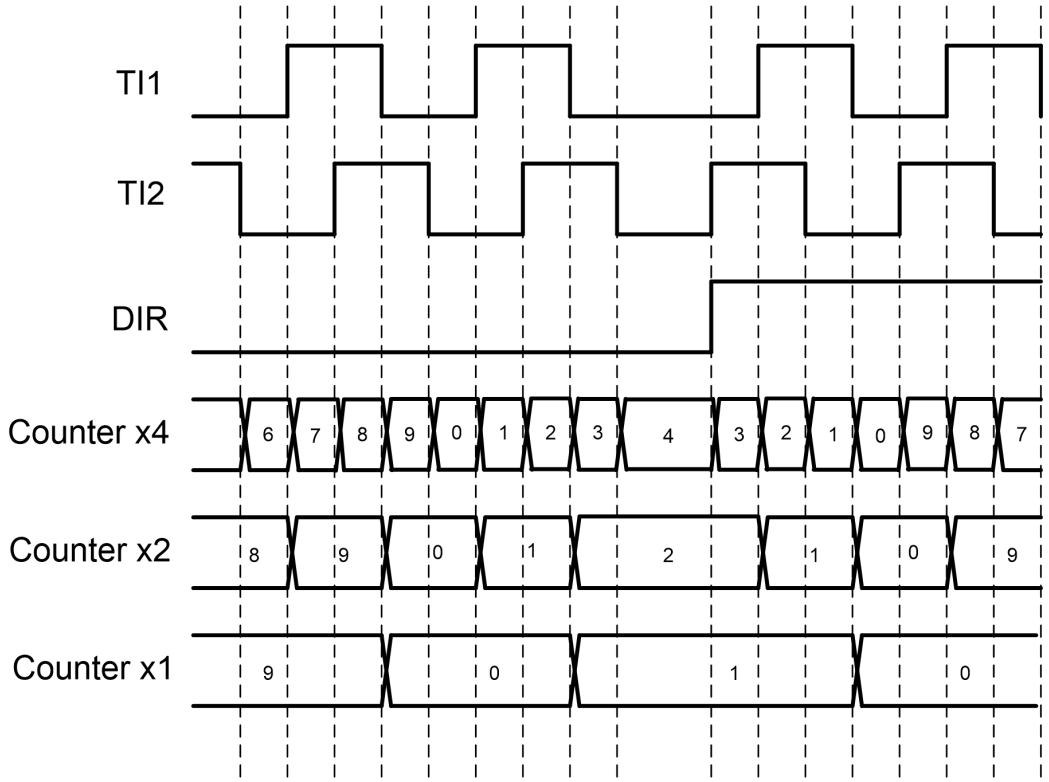


图24-53正交编码器不同边沿计数模式示例

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。

24.5.22.2 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

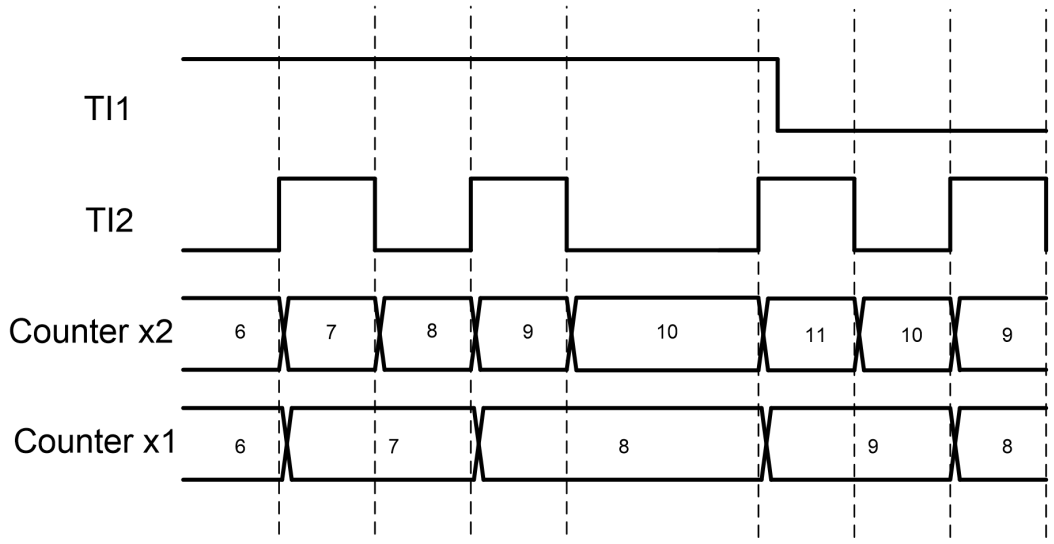


图24-54时钟方向编码器示例

24.5.22.3 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）

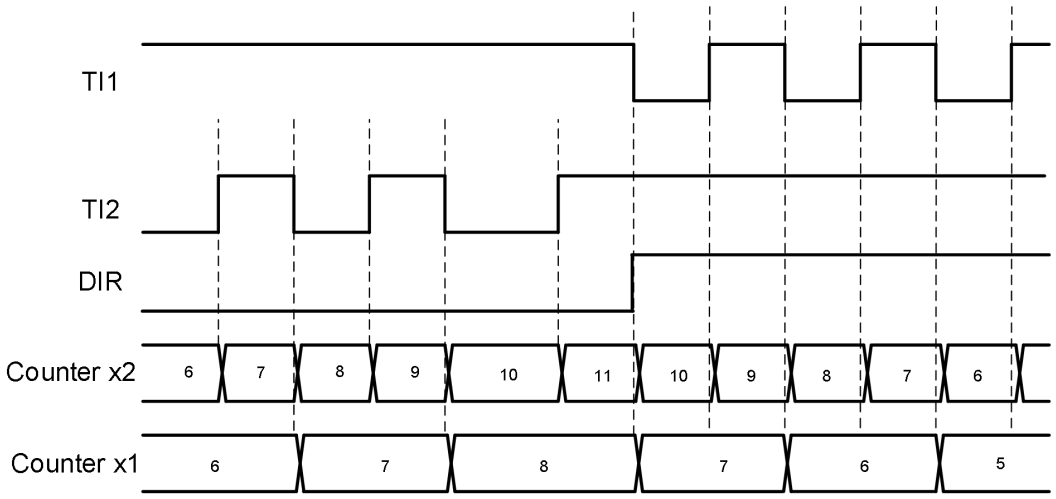


图24-55带方向的时钟编码器（CC1P=CC2P=0）

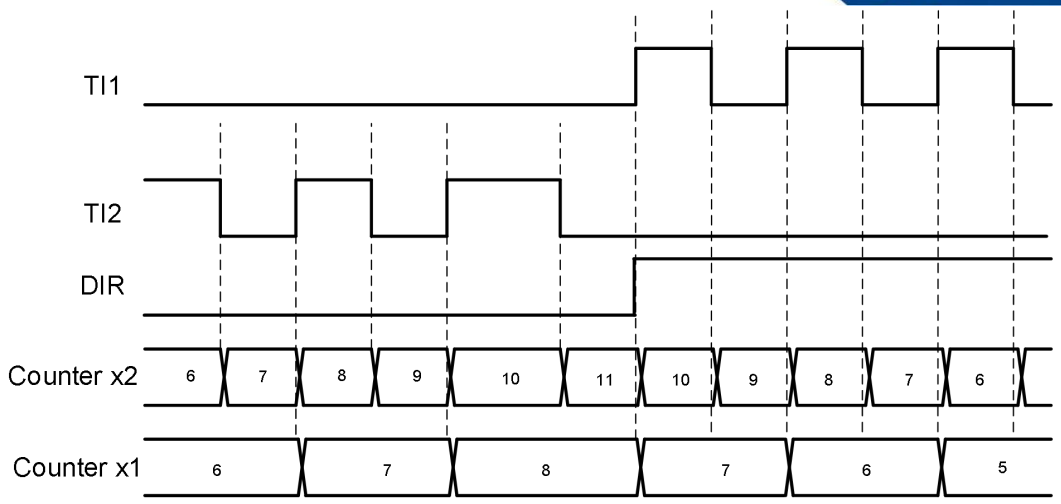


图24-56带方向的时钟编码器（CC1P=CC2P=1）

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 24-8 directional clock mode 计数方式

24.5.22.4 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：

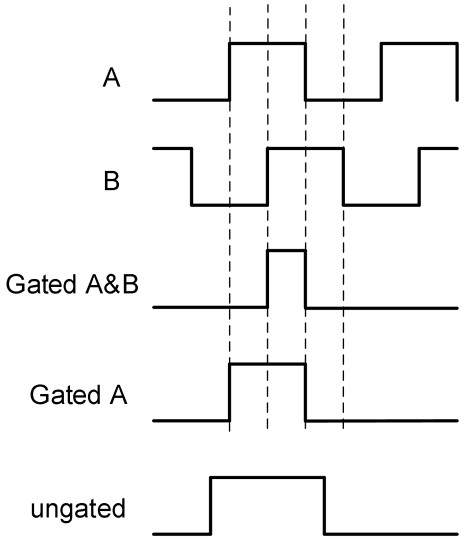


图24-57编码器index定义

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）。

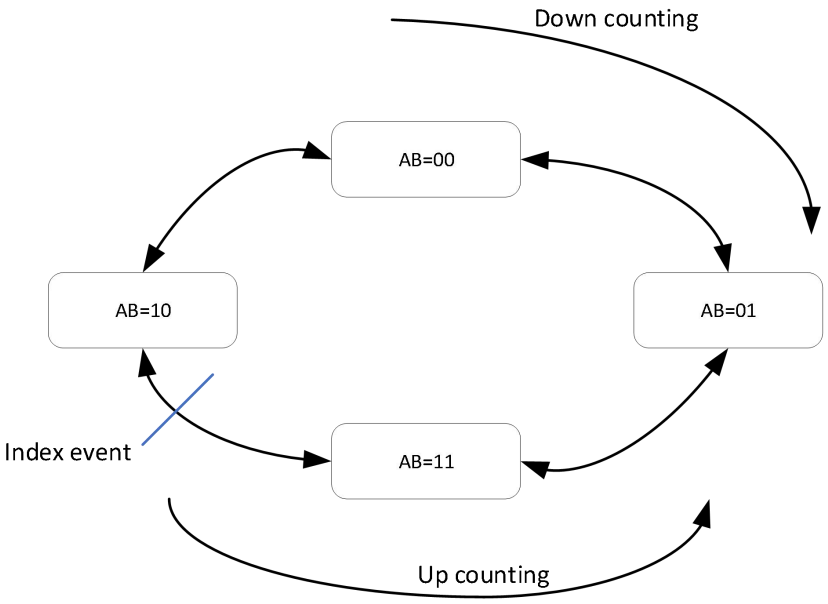


图24-58IPOS[1:0]=11时的index识别

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。

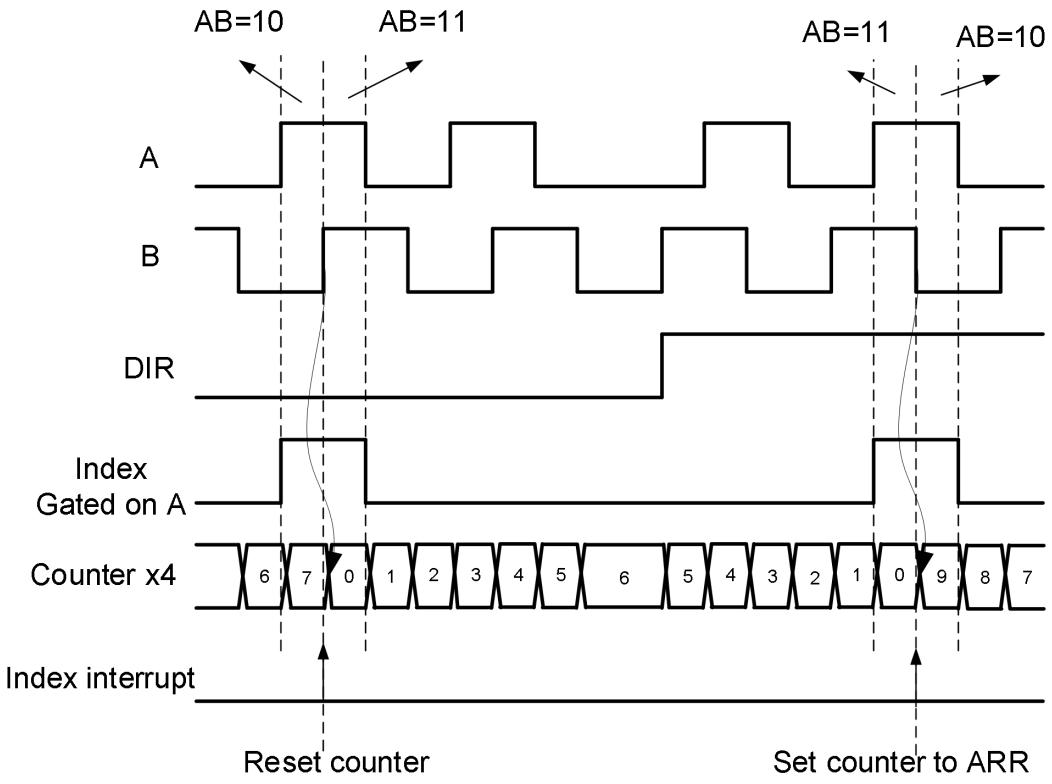


图24-59IPOS[1:0]=11, index gated on A, ARR=9

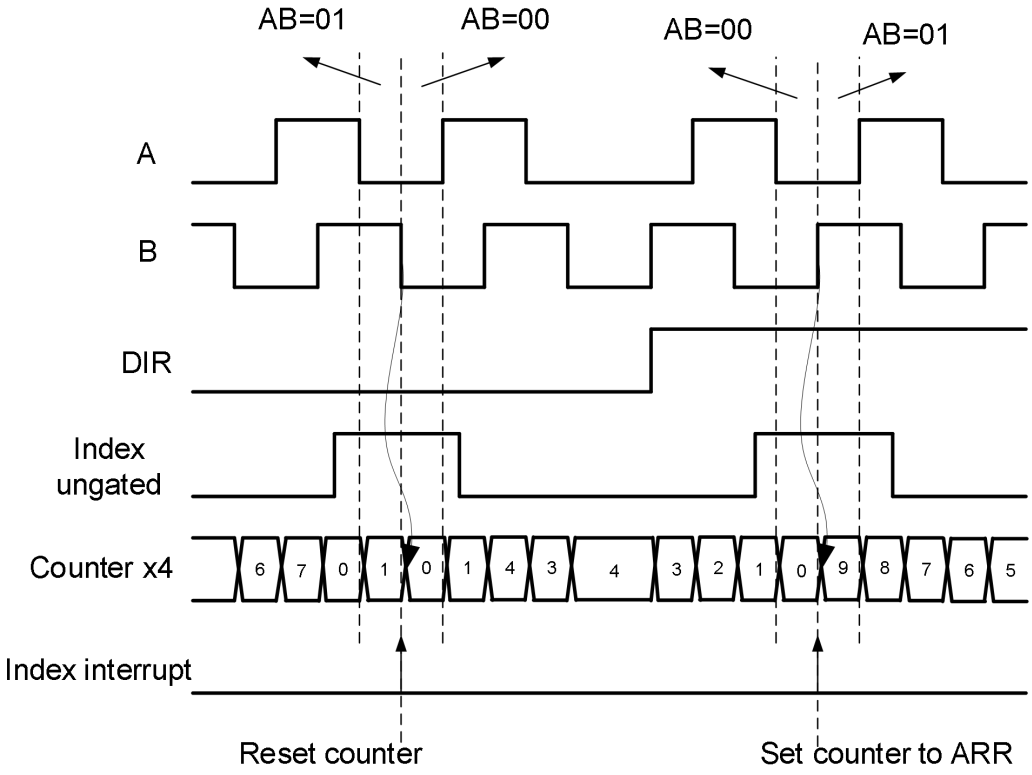


图24-60IPOS[1:0]=00, index ungated, ARR=9

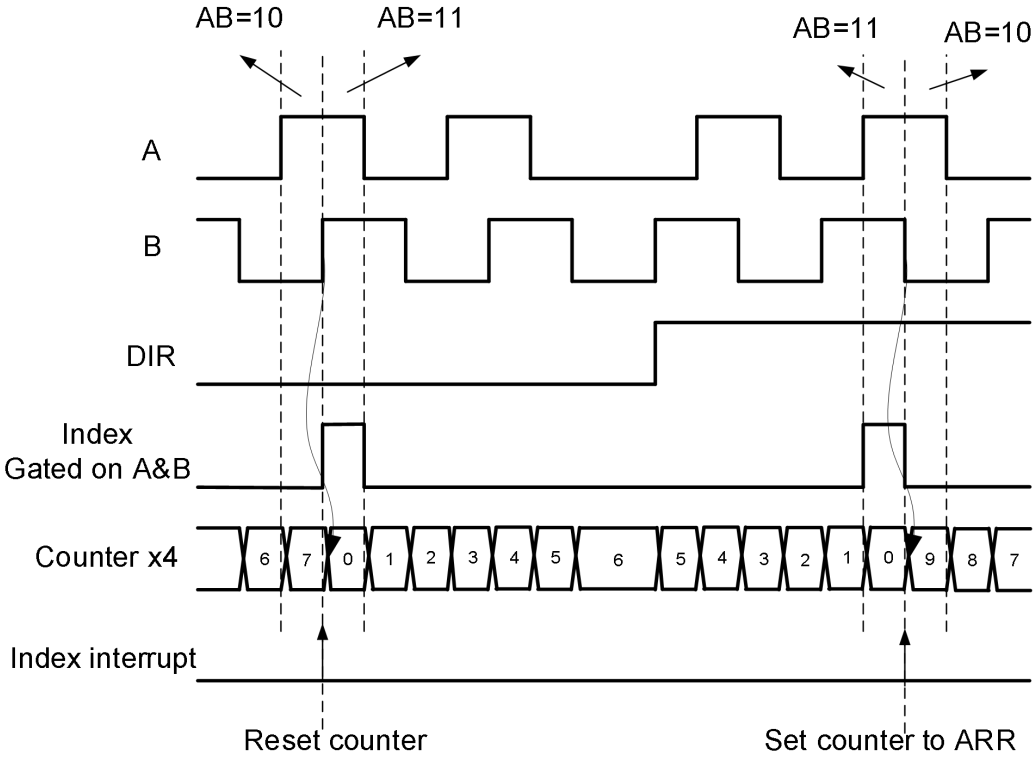


图24-61IPOS[1:0]=11, index gated on A&B, ARR=9

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：

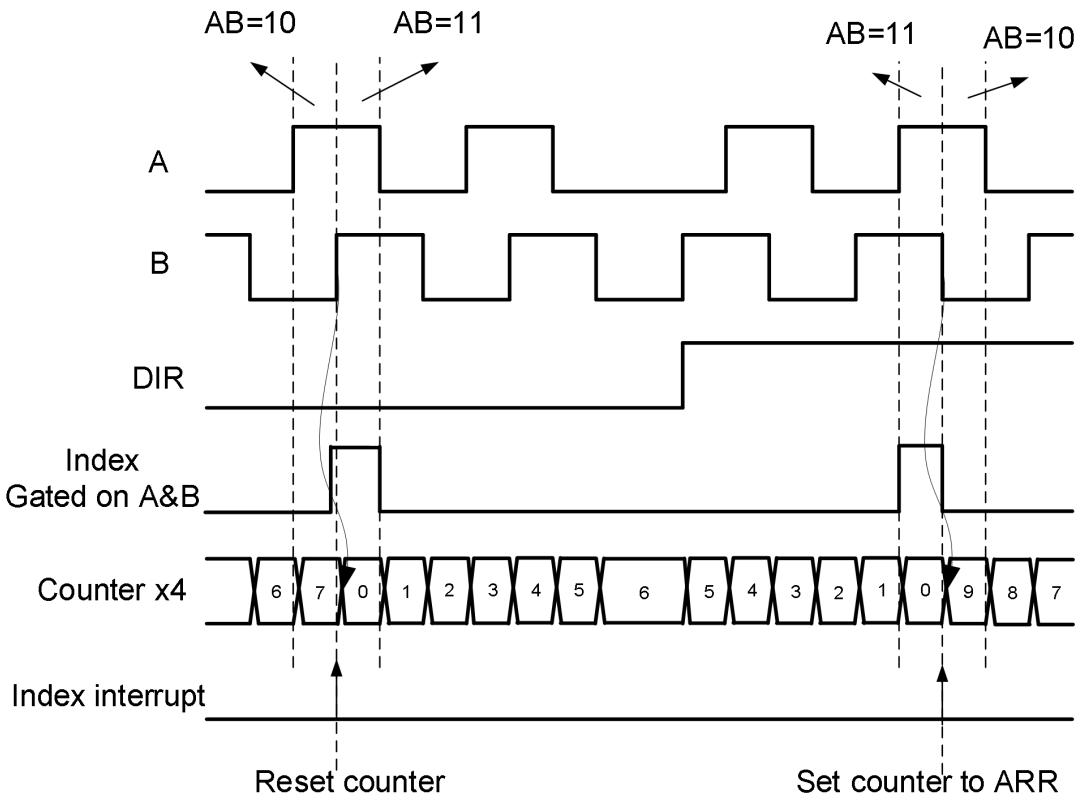


图24-62IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B

下图中则是index滞后B相输入，或者说出现一个窄Z相信号：

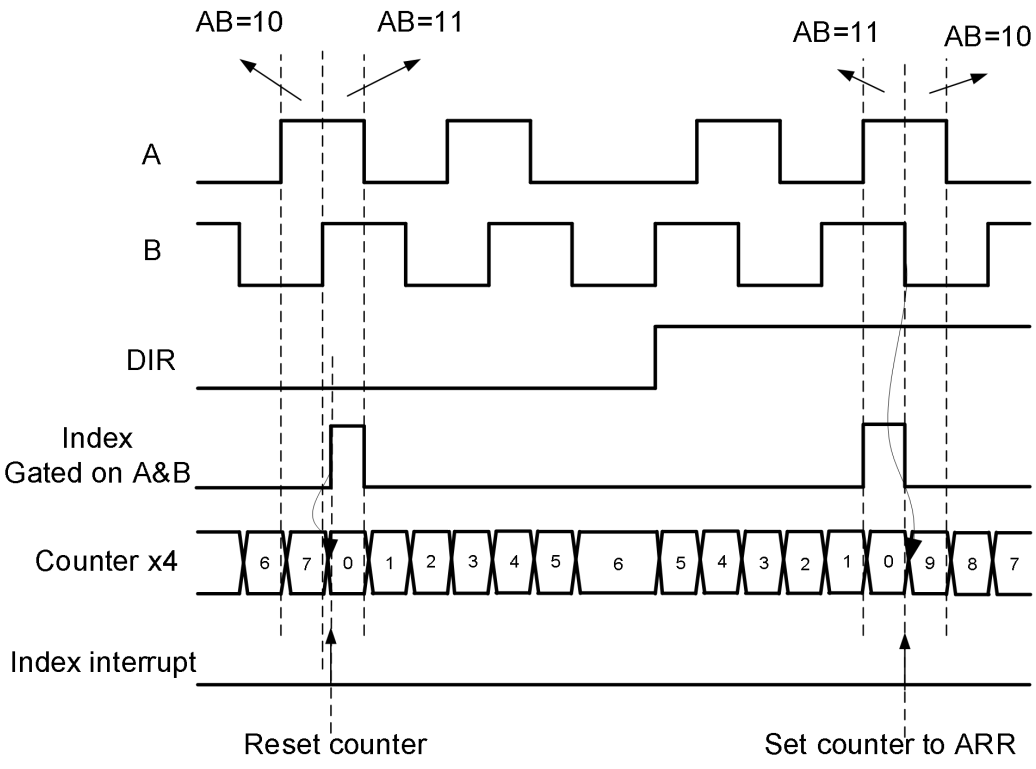


图24-63IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B

上图中index lagging的区域放大图如下：

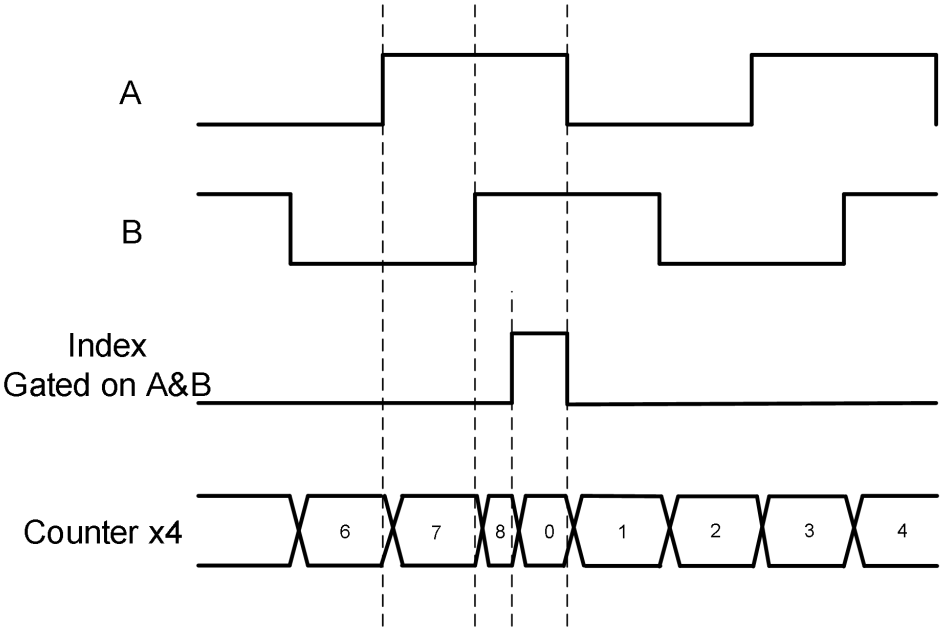


图24-64IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed

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）：

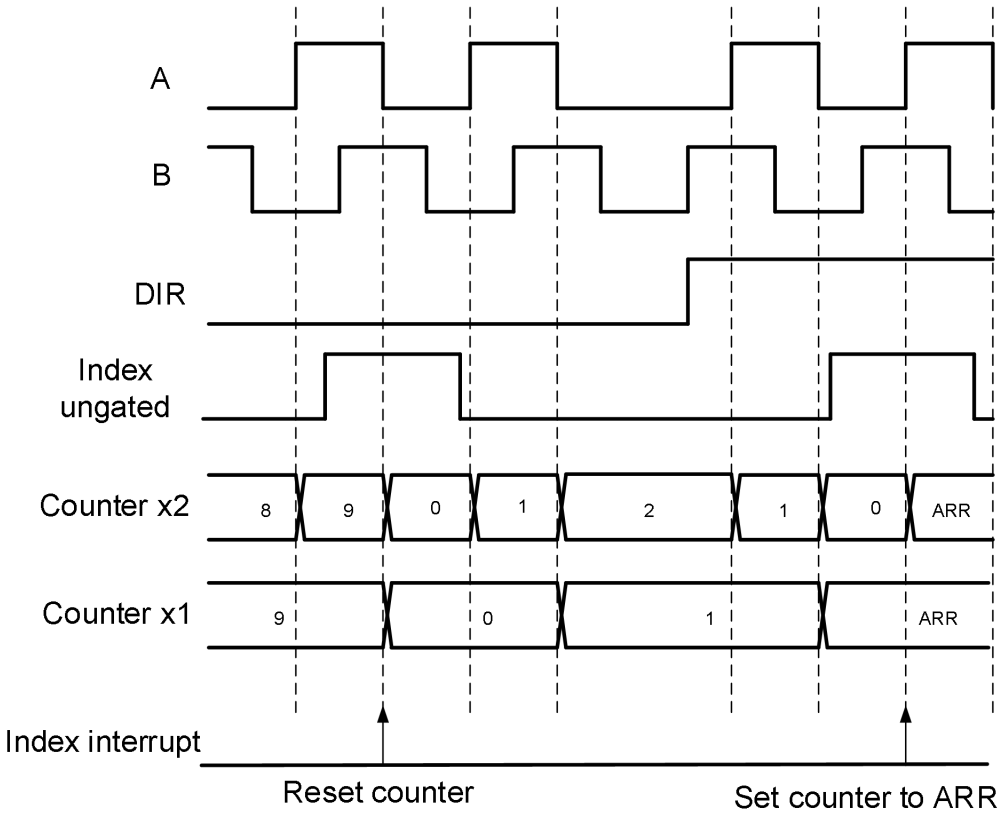


图24-65IPOS[1:0]=01, ungated index

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。

24.5.22.5 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。

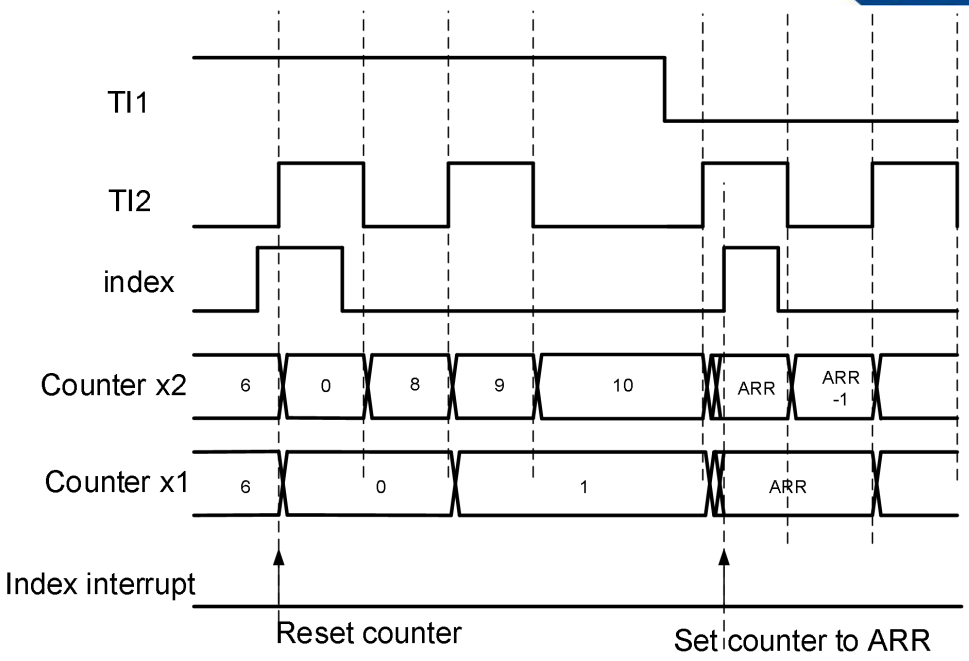


图24-66IPOS[0]=1, clock+direction mode

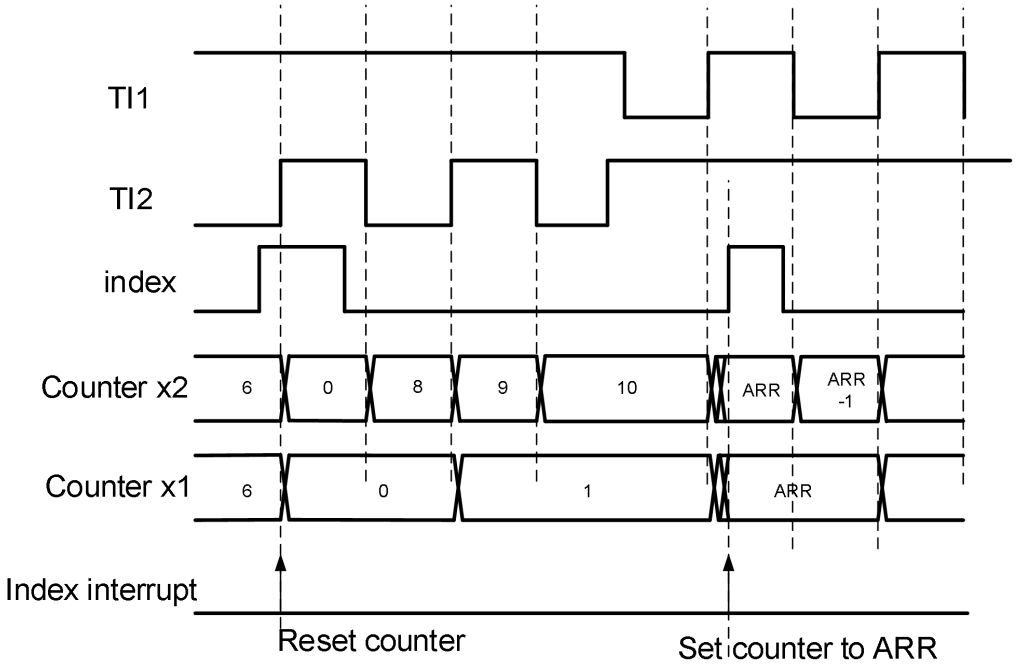


图24-67IPOS[0]=1, directional clock mode

24.5.22.6 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。

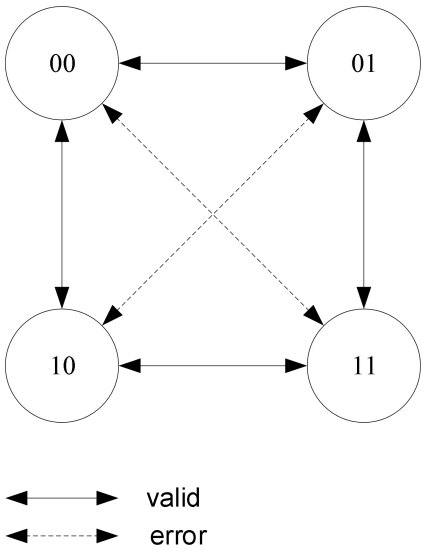


图24-68正交编码状态转移

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。

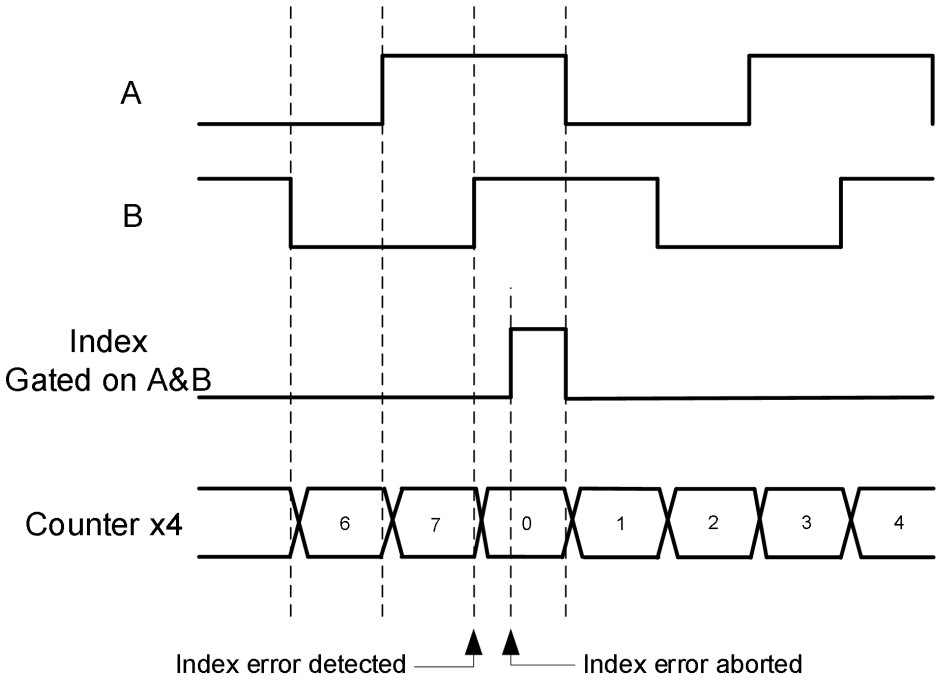


图24-69index错误撤销

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。

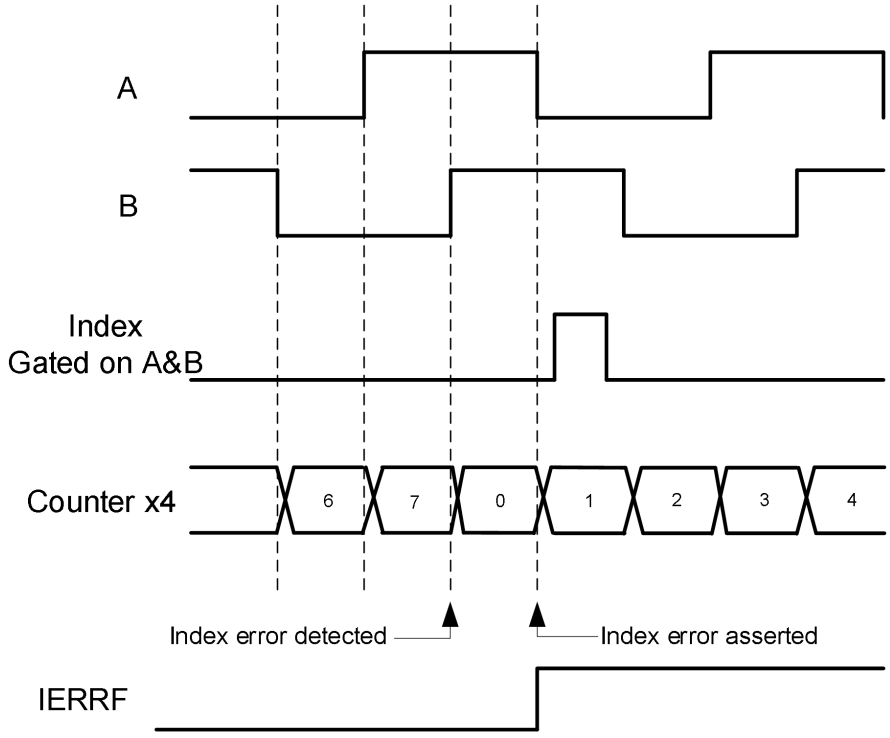


图24-70index错误置位

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。

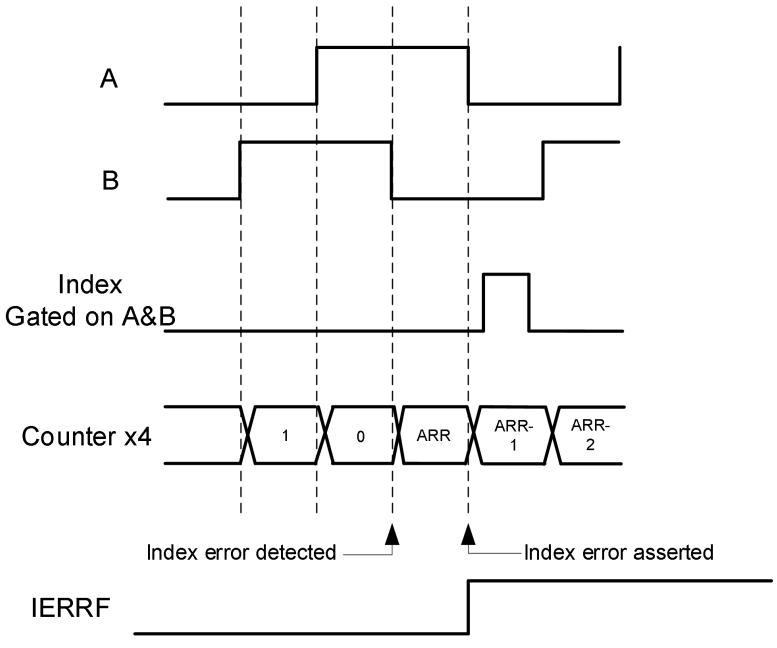


图24-71向下计数时index错误置位

24.5.22.7 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

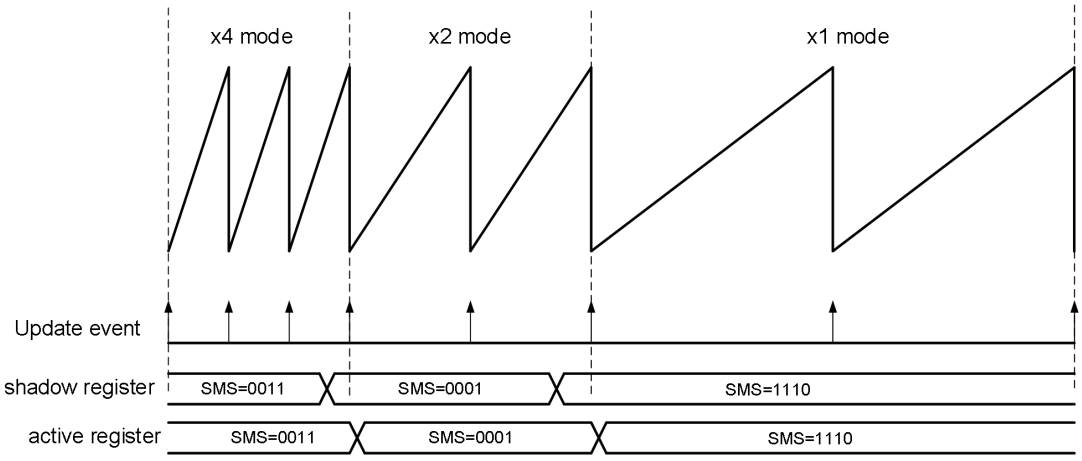


图24-72counter mode preload

24.5.22.8 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。

24.5.23 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。

24.5.23.1 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

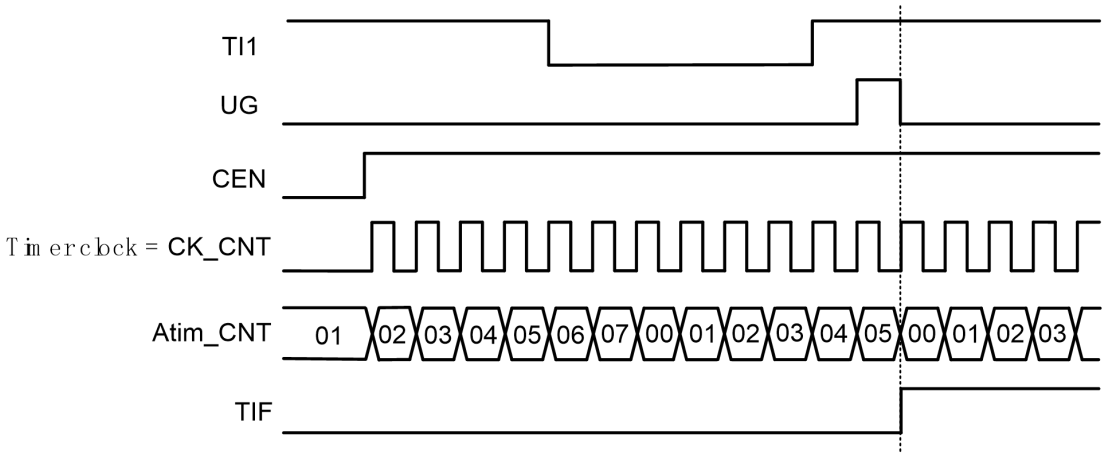


图24-73复位模式下的时序

24.5.23.2 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

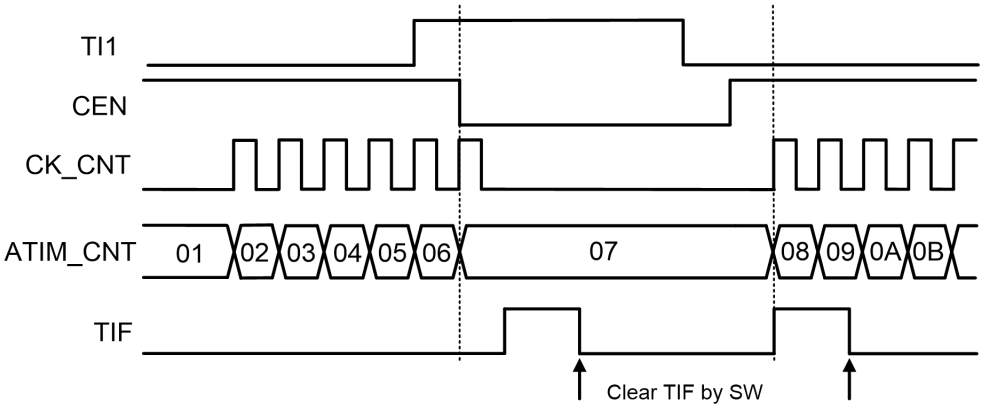


图24-74门控模式下的时序

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。

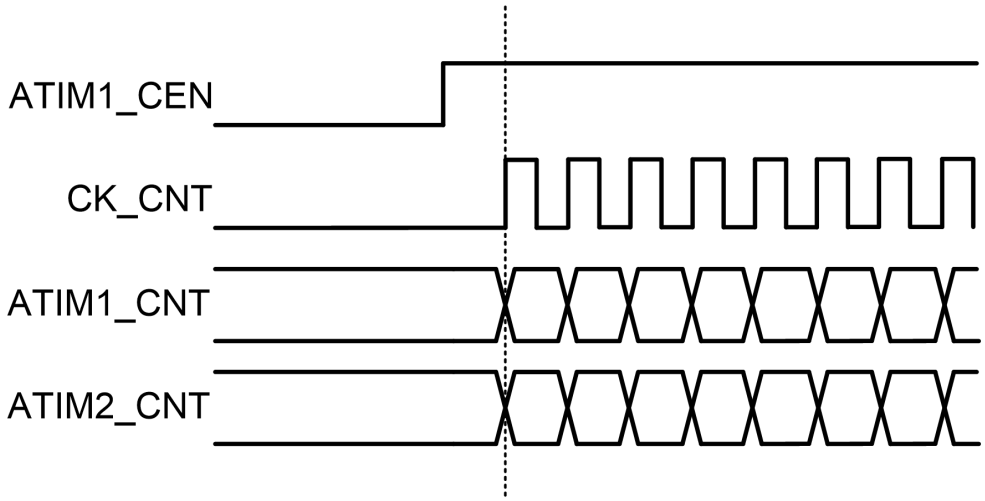


图24-75多timer同时使能

24.5.23.3 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

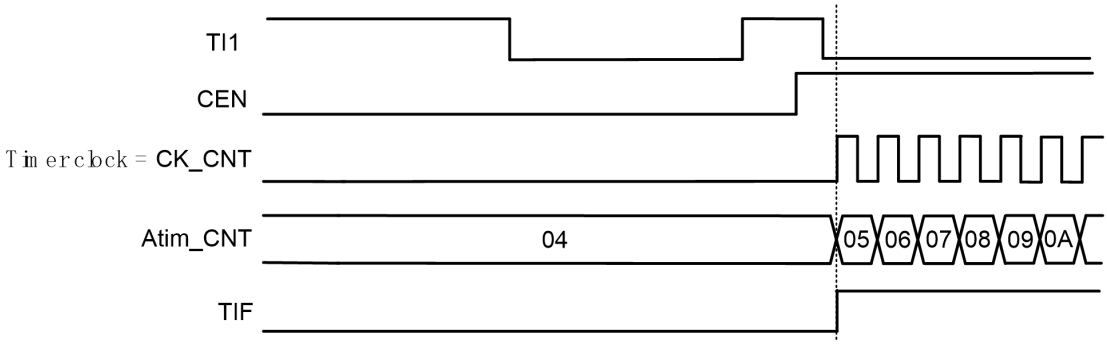


图24-76触发器模式下的时序

24.5.23.4 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

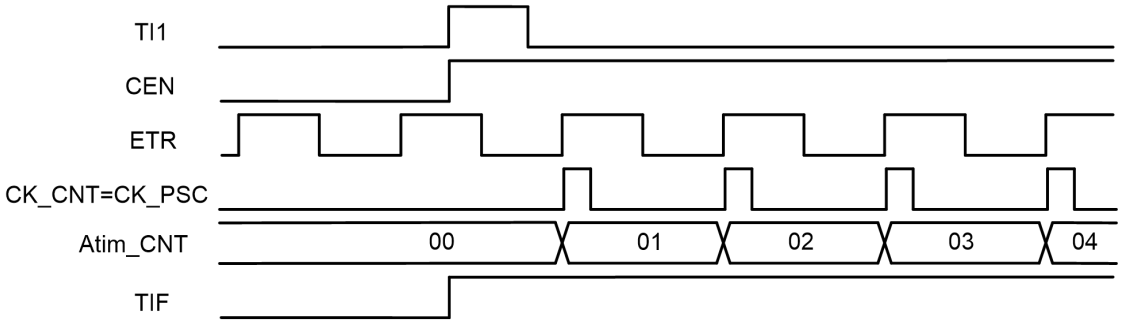


图24-77外部时钟模式2＋触发模式下的时序

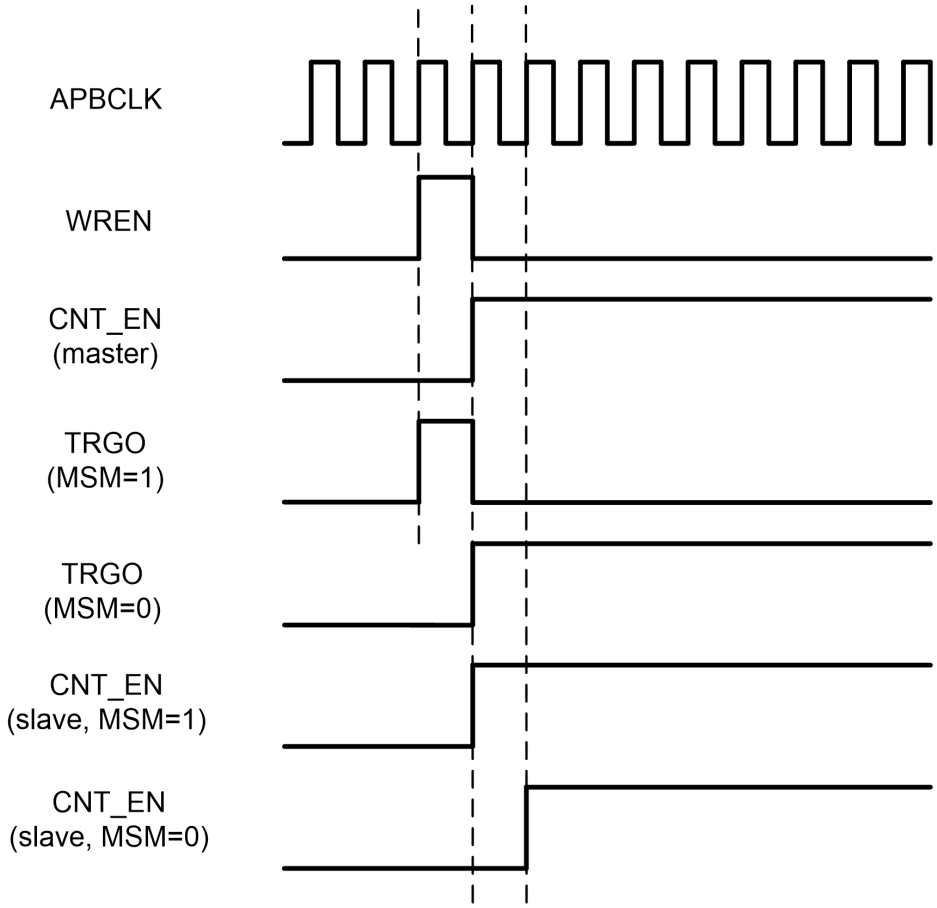
24.5.24 触发信号输出

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



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

只有当系统时钟是PLLx1时，才允许ATIM工作时钟选择PLLx2。因此可以认为ATIM输出的TRGO和TRGO2还是同步信号，只不过最高频率翻倍。为了保证APB slave可以正确采样到触发信号，当ATIM时钟为总线时钟的2倍时，必须将TRGO和TRGO2脉冲展宽到2个clock宽度（等效于1个APBCLK宽度）

24.5.25 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	
	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	

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

24.5.26 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）。

24.5.27 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。

24.5.28 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）。

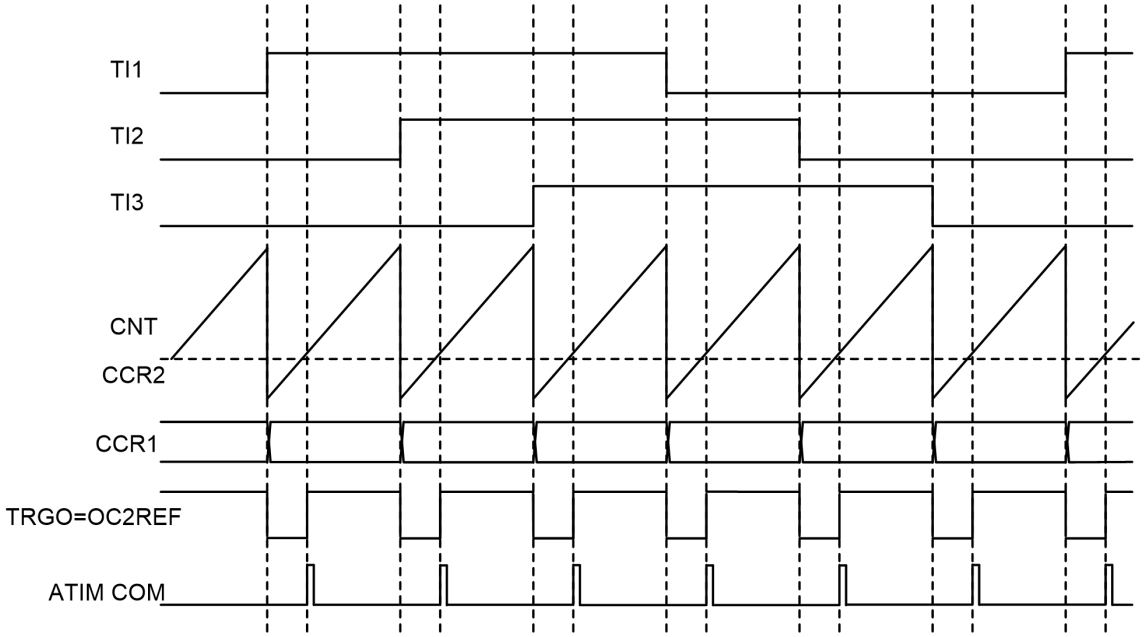


图24-78霍尔传感器接口

24.5.29 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DBG模块的DBG_AT_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。

24.6 寄存器

ATIM模块基地址：0x4001_B000

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_DIER
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	ATIM_BRKFLAG
0x64~0xF4	-	-
0xF8	ATIM DMA 控制寄存器	ATIM_DCR
0xFC	ATIM DMA 访问寄存器	ATIM_DMAR

24.6.1 ATIM 控制寄存器 1（ATIM_CR1）

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

Bit	助记符	功能描述
		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

24.6.2 ATIM 控制寄存器 2（ATIM_CR2）

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 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

Bit	助记符	功能描述
		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
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

24.6.3 ATIM 从机模式控制寄存器（ATIM_SMCR）

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
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

Bit	助记符	功能描述
		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） 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，

Bit	助记符	功能描述
		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

24.6.4 ATIM DMA 和中断使能寄存器（ATIM_DIER）

名称	ATIM_DIER							
地址	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TERRIE	IERRIE	DIRIE	IDXIE	CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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**	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

Bit	助记符	功能描述
		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
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

24.6.5 ATIM 状态寄存器（ATIM_ISR）

名称	ATIM_ISR							
地址	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	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 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

Bit	助记符	功能描述
		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

24.6.6 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

Bit	助记符	功能描述
		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。

24.6.7 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

Bit	助记符	功能描述
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
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

Bit	助记符	功能描述
		11：CC2 通道配置为输入，IC3 映射到 TRC 注意：CC2S 仅在通道关闭时（CC2E=0）可以写
7:4	**IC1F**	输入捕捉 1 滤波 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	输入捕捉 1 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
1:0	**CC1S**	捕捉/比较 1 通道选择 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S 仅在通道关闭时（CC1E=0）可以写

24.6.8 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]
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

Bit	助记符	功能描述
9:8	**CC4S**	捕捉/比较 4 通道选择 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S 仅在通道关闭时（CC4E=0）可以写
7:4	**IC3F**	输入捕捉 1 滤波 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 f_{DTS} 采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS/2}$, N=6 0101：$f_{SAMPLING}=f_{DTS/2}$, N=8 0110：$f_{SAMPLING}=f_{DTS/4}$, N=6 0111：$f_{SAMPLING}=f_{DTS/4}$, N=8 1000：$f_{SAMPLING}=f_{DTS/8}$, N=6 1001：$f_{SAMPLING}=f_{DTS/8}$, N=8 1010：$f_{SAMPLING}=f_{DTS/16}$, N=5 1011：$f_{SAMPLING}=f_{DTS/16}$, N=6 1100：$f_{SAMPLING}=f_{DTS/16}$, N=8 1101：$f_{SAMPLING}=f_{DTS/32}$, N=5 1110：$f_{SAMPLING}=f_{DTS/32}$, N=6 1111：$f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	输入捕捉 3 预分频 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC 寄存器在 CC1E=0 时复位
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

24.6.9 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
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，CC1P 控制输入取反 CC1P=0：输入不取反 CC1P=1：输入取反 CC1P 和 CC1NP 组合选择捕捉沿 CC1NP=0，CC1P=0：捕捉在 IC1 的上升沿进行 CC1NP=0，CC1P=1：捕捉在 IC1 的下降沿进行 CC1NP=1：捕捉在 IC1 的上升沿或下降沿进行
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

Bit	助记符	功能描述
		1：使能捕捉功能

24.6.10 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

24.6.11 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 $f_{CK_CNT}=f_{CK_PSC}/(PSC[15:0]+1)$

Bit	助记符	功能描述
		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

24.6.12 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

24.6.13 ATIM 重复计数寄存器（ATIM_RCR）

名称	ATIM_RCR							
地址	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REP[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	**REP**	重复计数值。REP 不为 0 时，每次 update 条件发生时 REP 递减，当 REP=0 时触发 update 事件

24.6.14 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

24.6.15 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

Bit	助记符	功能描述
		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

24.6.16 ATIM 捕捉/比较寄存器 3（ATIM_CCR3）

名称	ATIM_CCR3							
地址	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3[15:8]							
位权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3[7:0]							
位权限	RW-00000000							

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	**CCR3**	捕捉/比较通道 3 寄存器 **如果通道 3 配置为输出：** 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC3 输出 **如果通道 3 配置为输入：** CCR3 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR3 为只读

24.6.17 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

24.6.18 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

Bit	助记符	功能描述
		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

Bit	助记符	功能描述
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}

24.6.19 ATIM DMA 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	-							
位权限	U-0							
位	Bit23	Bit22	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

Bit	助记符	功能描述
		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。

24.6.20 ATIM DMA 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[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

24.6.21 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 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

24.6.22 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

24.6.23 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

24.6.24 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
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

24.6.25 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 0000: ATIM_CH1 0001: ti1_in1 …. 1111: ti1_in15

24.6.26 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	-		BKCMP4E	BKCMP3E	BKCMP2E	BKCMP1E	BKIN2E	BKIN1E
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	--	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
5	BKCMP4E	比较器 4 输出刹车使能，1 有效
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

25 定时器阵列单元（TAU）

25.1 概述

TAU是16bit定时器组成的阵列，支持以下功能

- 2组共32个独立的16bit向上计数器（TAU00~TAU07，TAU10~TAU17）
- 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。

25.2 结构框图

TAU整体结构示意图如下，包含TAU0和TAU1两个阵列。

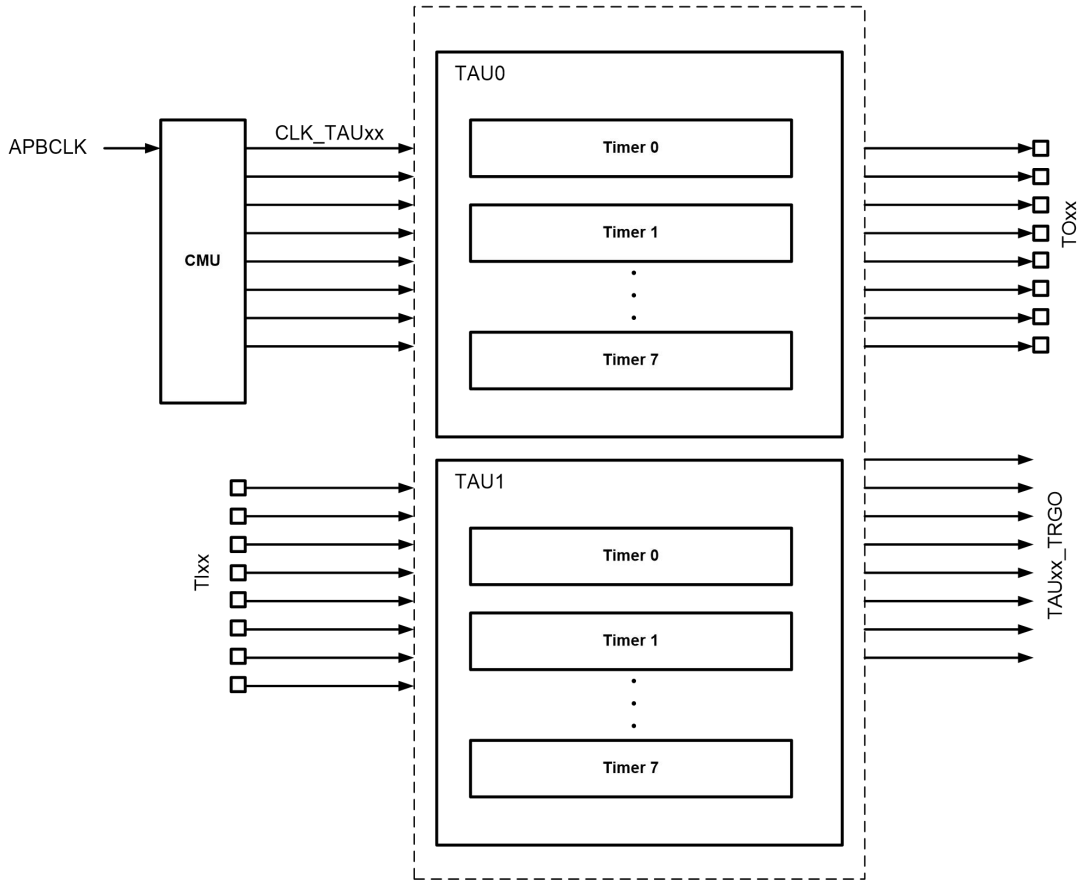


图 25-1TAU 结构框图

独立定时器结构框图如下：

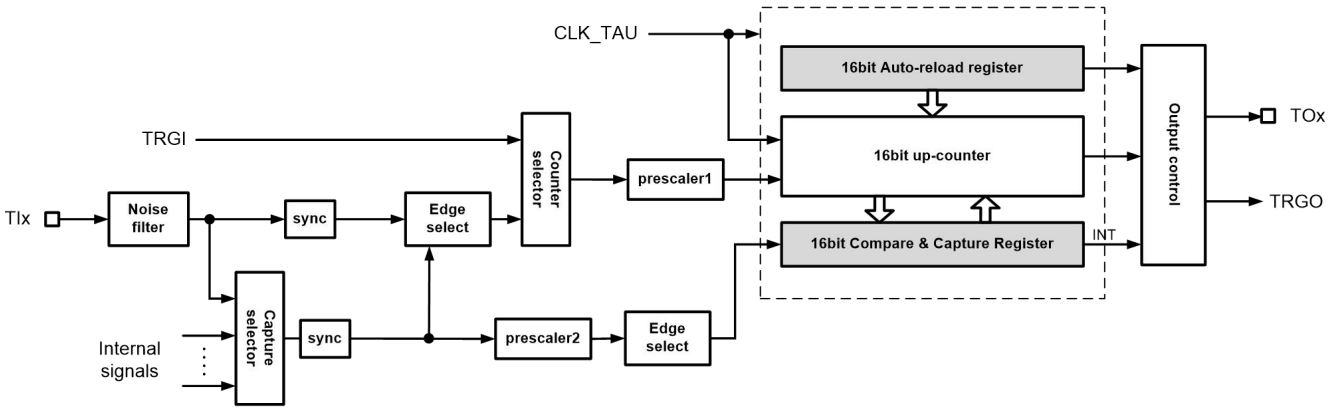


图 25-2TimerX 结构框图

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。

25.3 功能描述

25.3.1 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TAUxx_PCE寄存器，参见CMU章节。比如想要使能TAU0中的Timer0工作，应置位TAU00_PEC寄存器。

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。

25.3.2 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

25.3.3 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

25.3.4 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

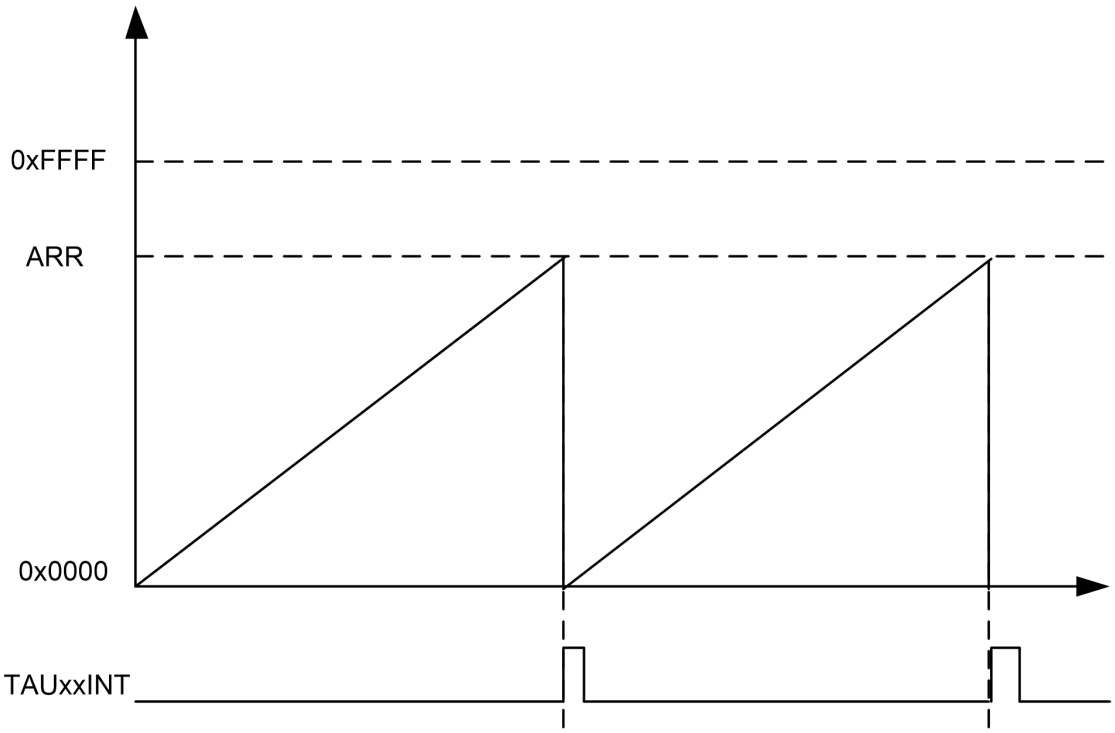


图 25-3 16bit 连续定时功能

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。

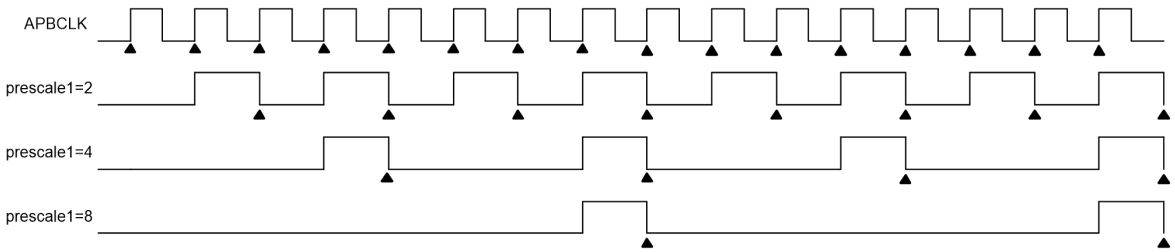


图 25-4 定时器内部时钟预分频计数

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。

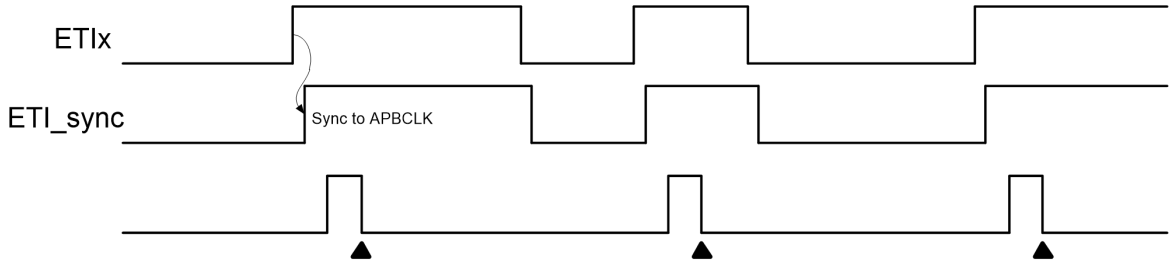


图 25-5 定时器外部事件上升沿递增时序

下图为ETIx上升下降沿事件计数，并且分频系数为2的例子。

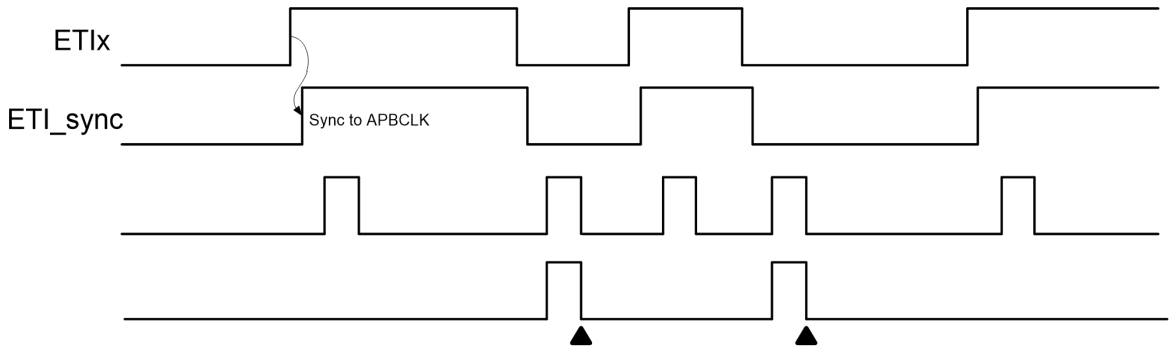


图 25-6 定时器外部事件上升下降沿递增，2 分频时序

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。

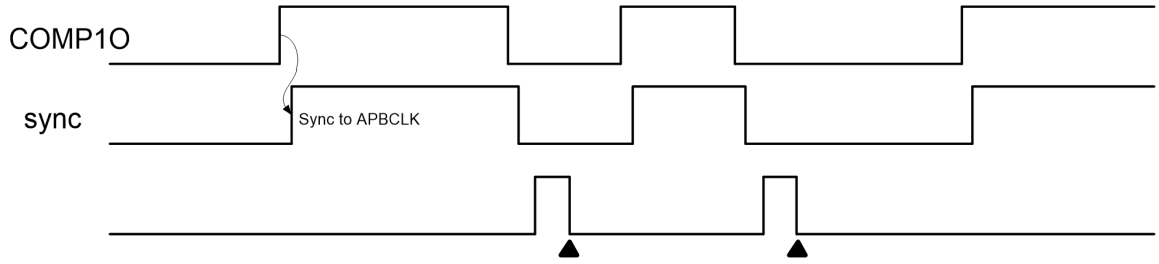


图 25-7 定时器内部信号下降沿递增

25.3.5 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。

25.3.6 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。

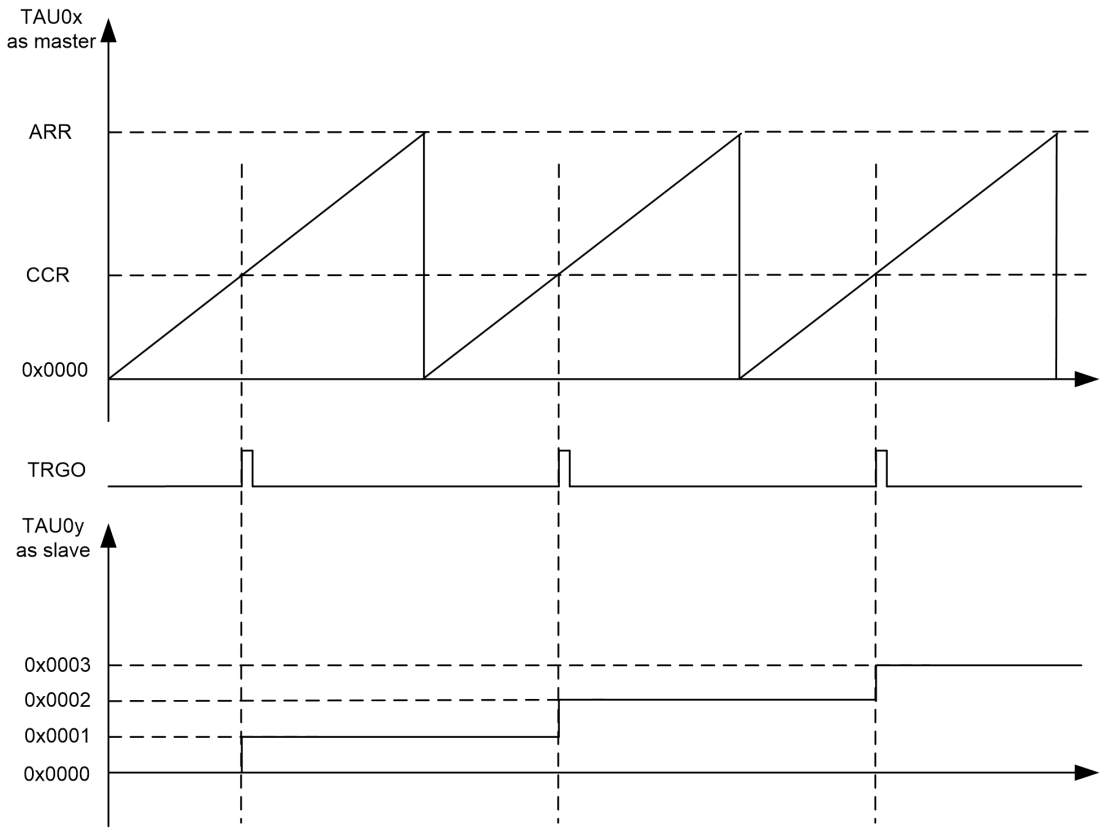


图 25-8 定时器级联计数（trigger to increment）

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。

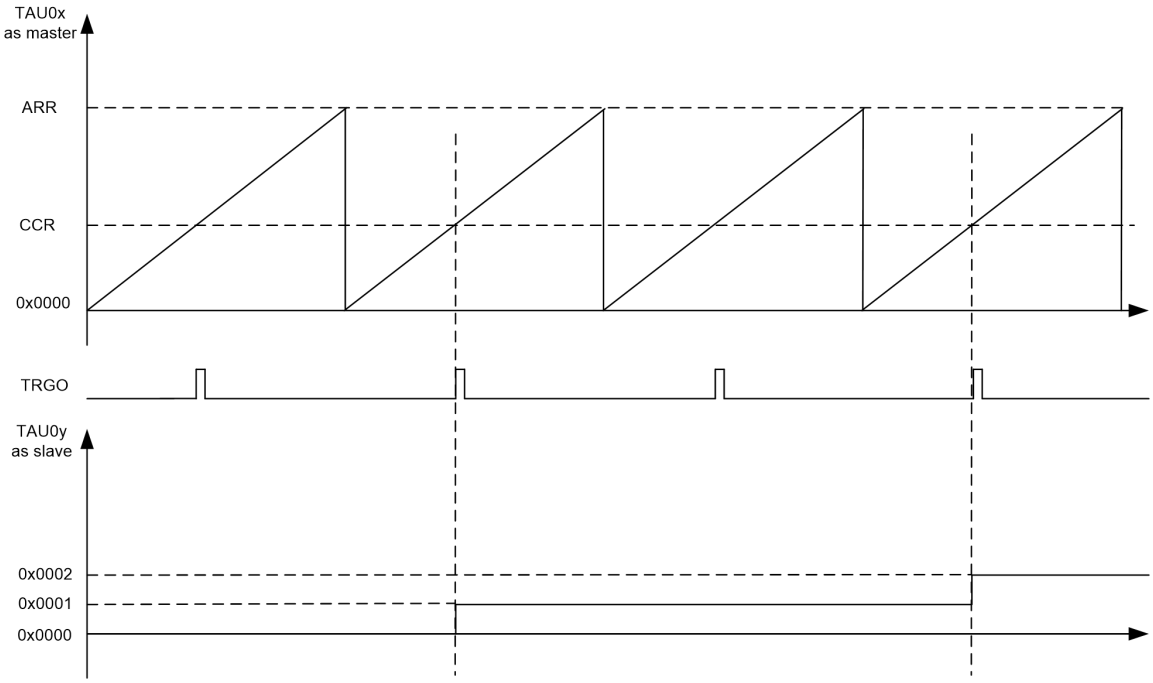


图 25-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1

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。

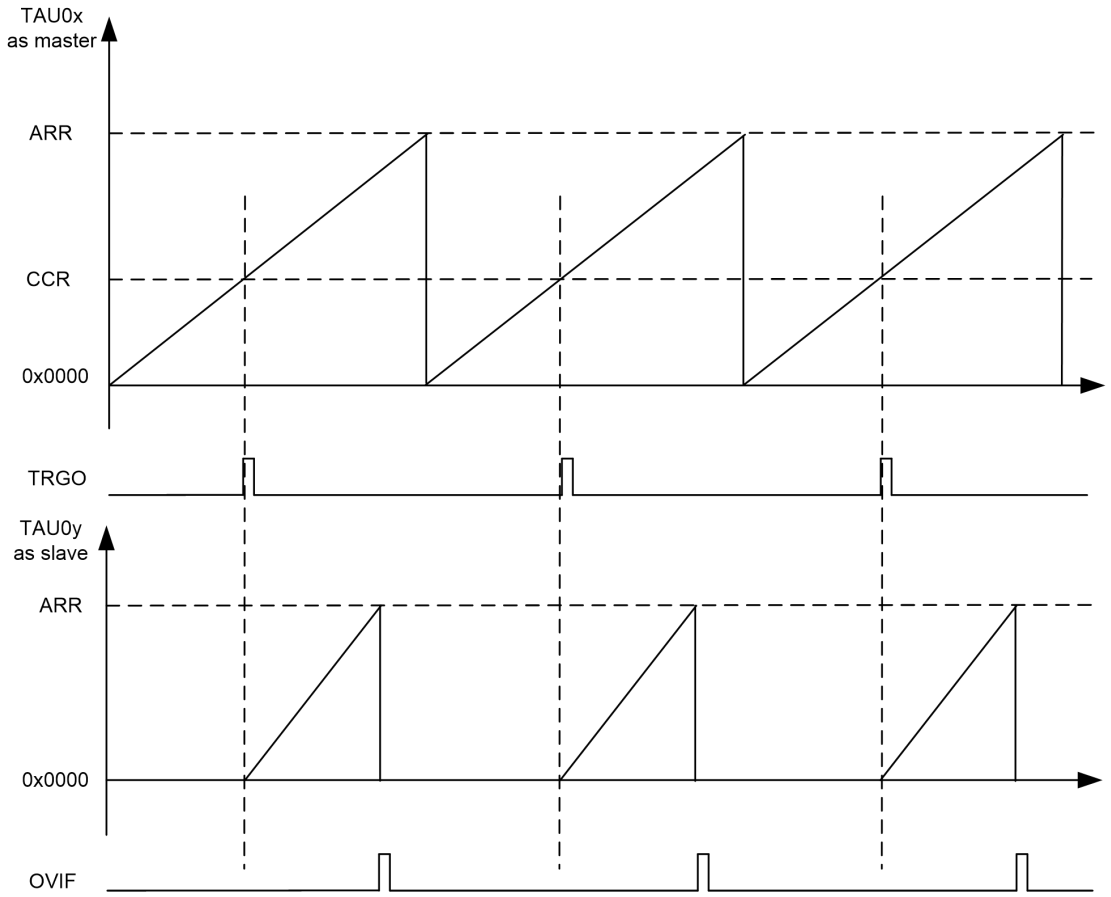


图 25-10 定时器级联计数（trigger to start）

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。

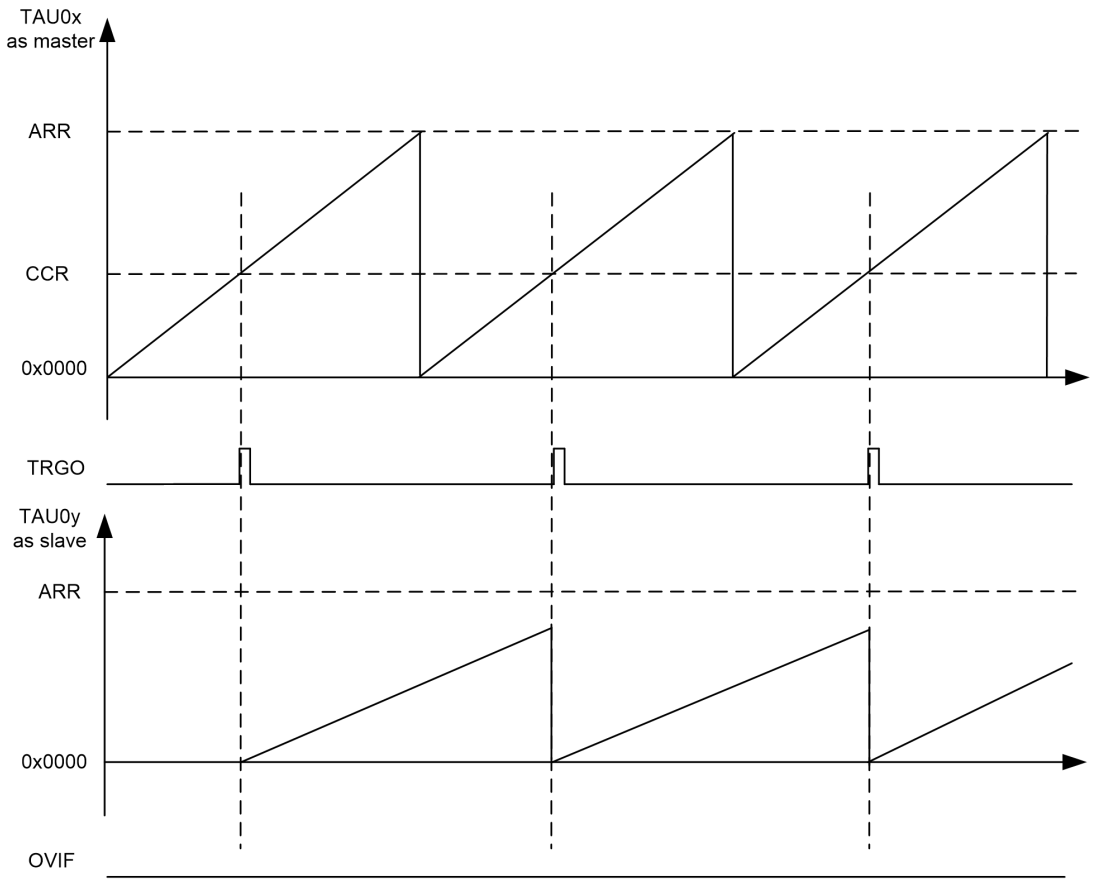


图 25-11 定时器级联计数（trigger to start），从机无法产生溢出

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。

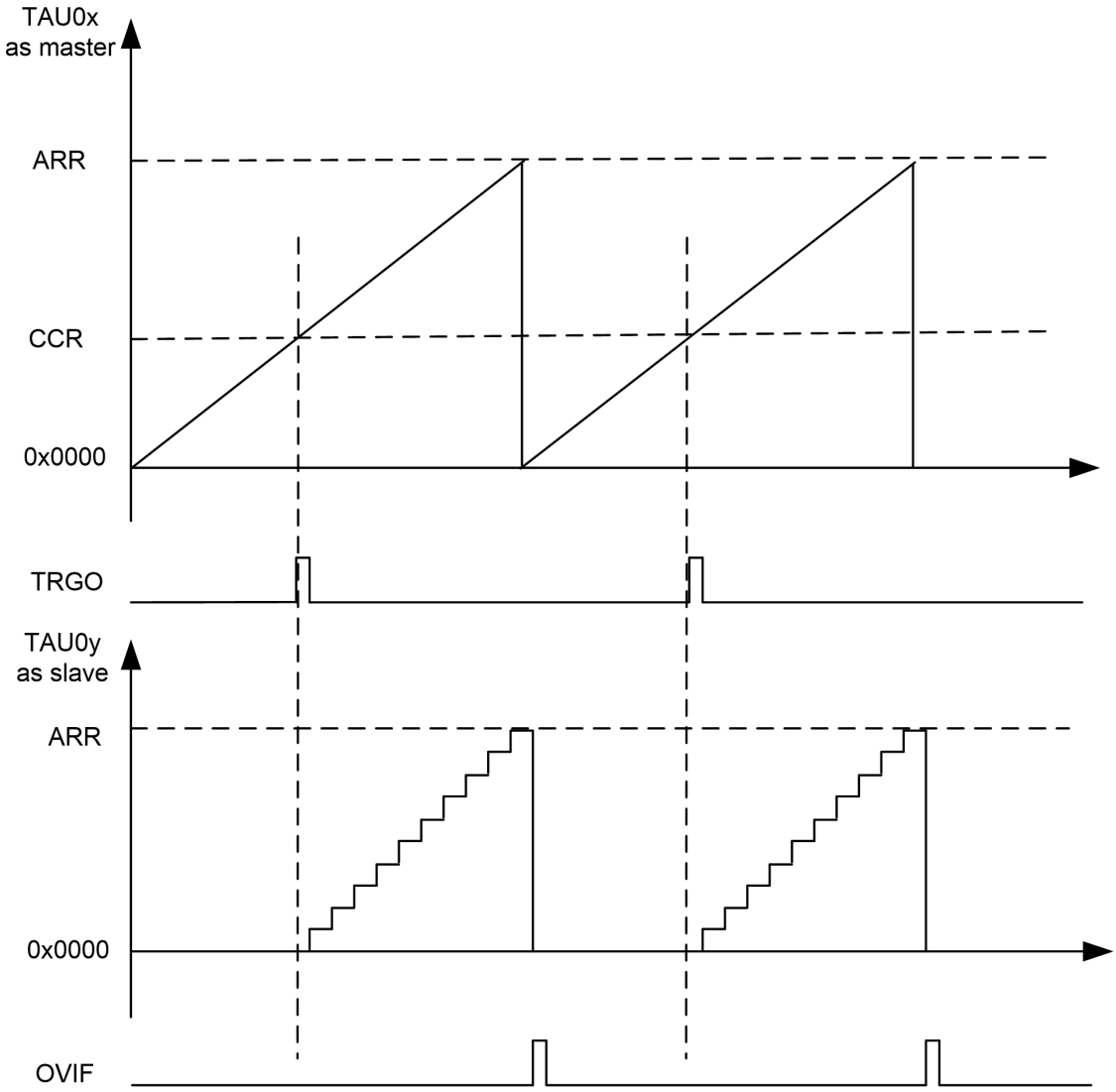


图 25-12 定时器级联计数（trigger to start），从机计数预分频

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 25-1 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。

25.3.7 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。

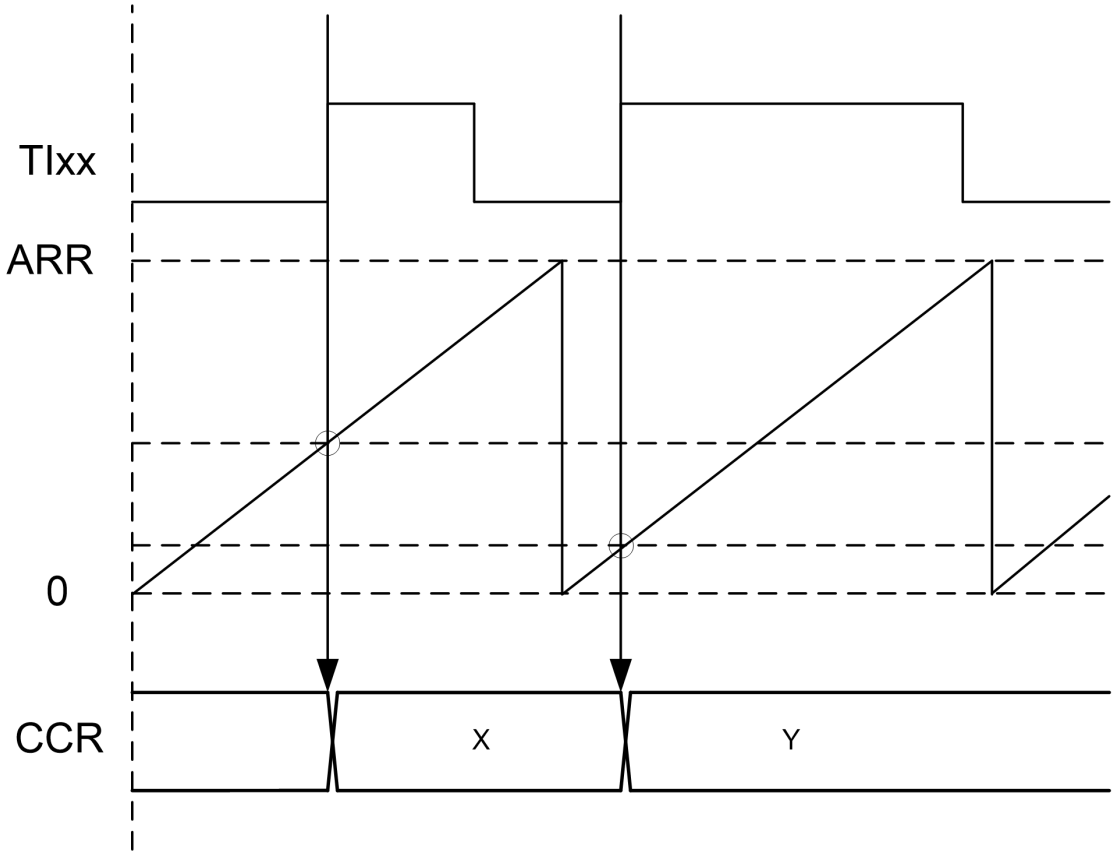


图 25-13 定时器捕捉外部输入信号上升沿

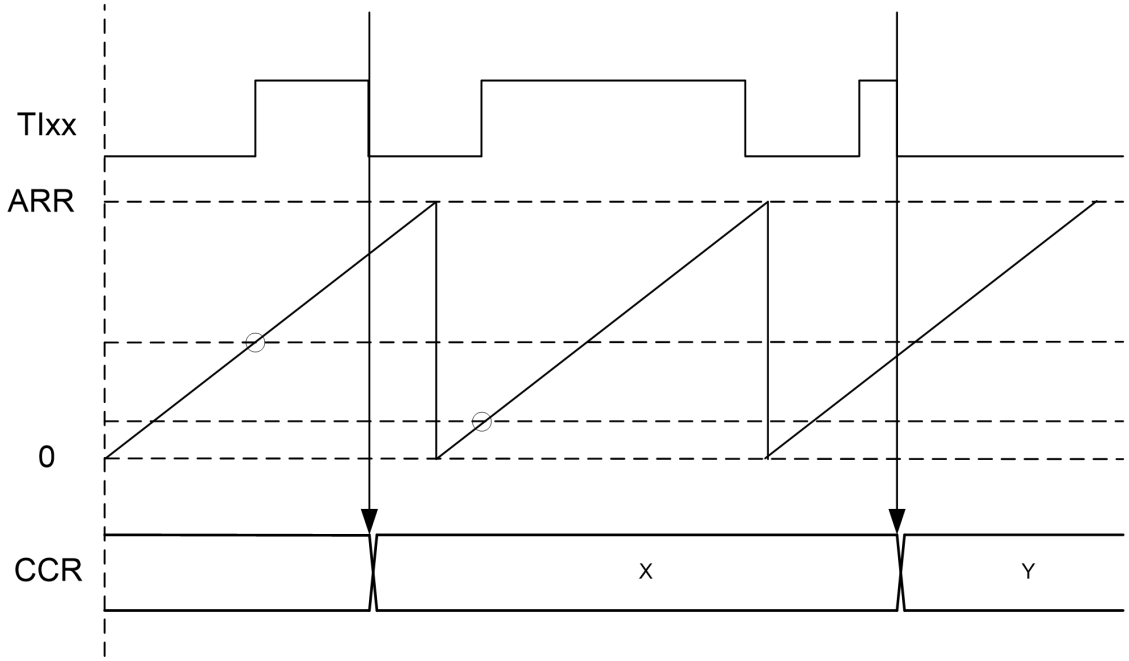


图 25-14 定时器 2 分频捕捉外部输入信号下降沿

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。

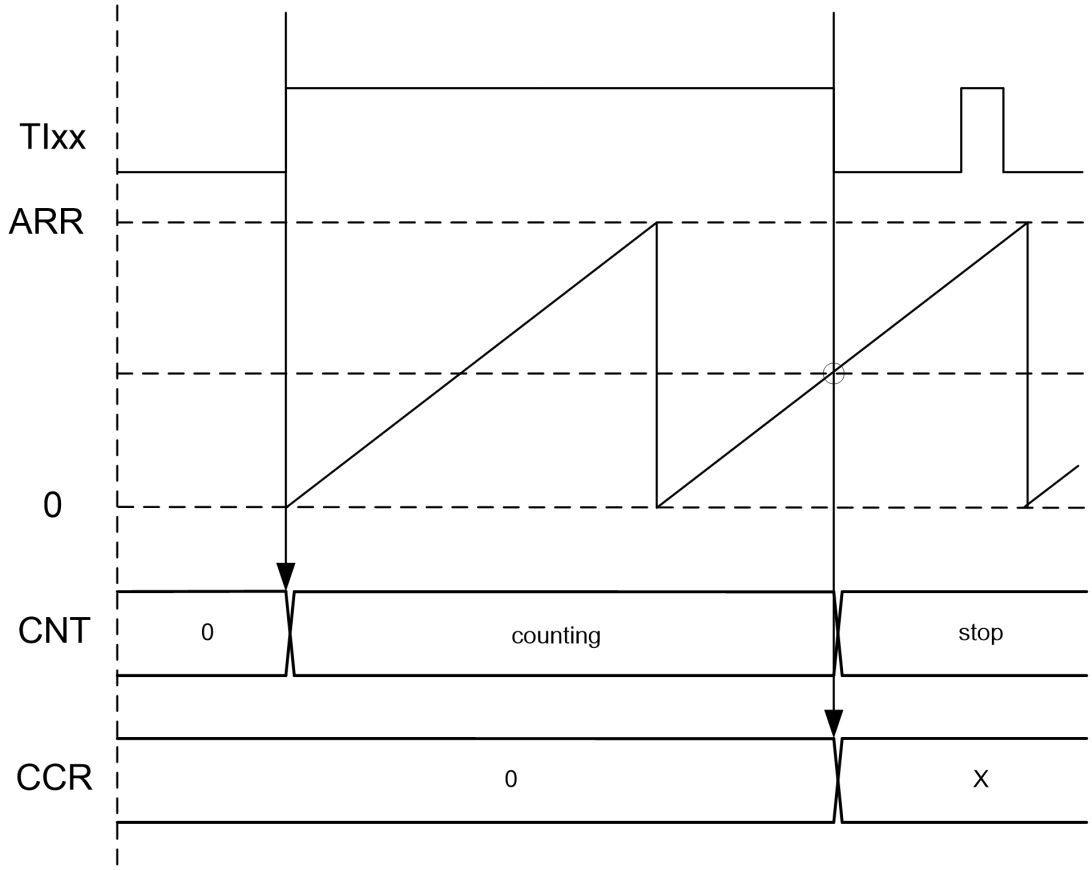


图 25-15 定时器脉冲宽度高电平捕捉

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。

TAUx定时器	CAPSEL	被捕捉信号
Timer0	000	TIx0
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP

Timer1	000	TIx1
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer2	000	TIx2
	001	UART0_RXD
	010	UART1_RXD
	011	XTLF
	100	UART2_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer3	000	TIx3
	001	UART3_RXD
	010	UART4_RXD
	011	XTLF
	100	UART5_RXD
	101	CMP1O
	110	CMP2O
	111	RCLP
Timer4	000	TIx4
	001	LPUART_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	UART6_RXD
	110	UART7_RXD
	111	-
Timer5	000	TIx5
	001	LPUART_RXD
	010	XTLF
	011	RCHF
	100	RCLP
	101	UART6_RXD
	110	UART7_RXD
	111	-
Timer6	000	TIx6
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-
	111	-
Timer7	000	TIx7
	001	-
	010	-
	011	-
	100	-
	101	-
	110	-

	111	-

表 25-2 定时器 x 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。

25.3.8 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：

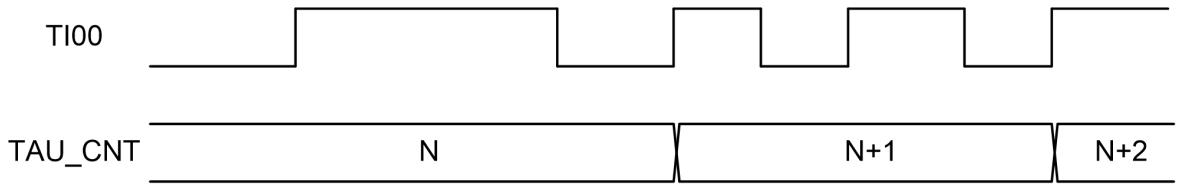


图 25-16TI00 二分频计数

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：

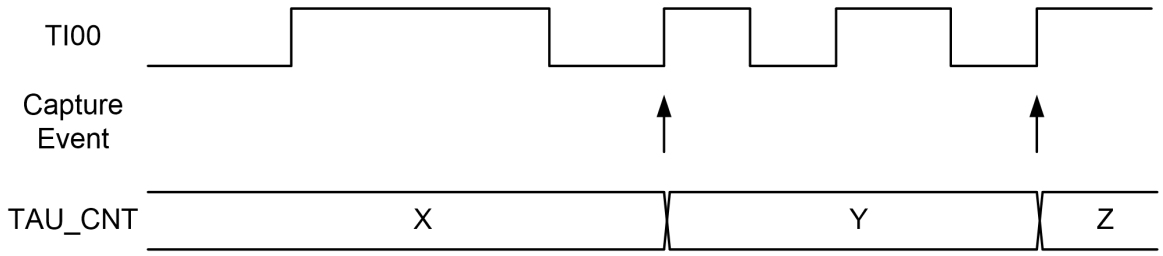


图 25-17TI00 二分频捕捉

25.3.9 16 位 PWM 输出

单个TAU支持独立16bitPWM输出，启动PWM之后定时器从初值开始计数，当计数值等于比较寄存器时，PWM输出置1，当计数值溢出时PWM输出置0，16bit PWM如下图所示：

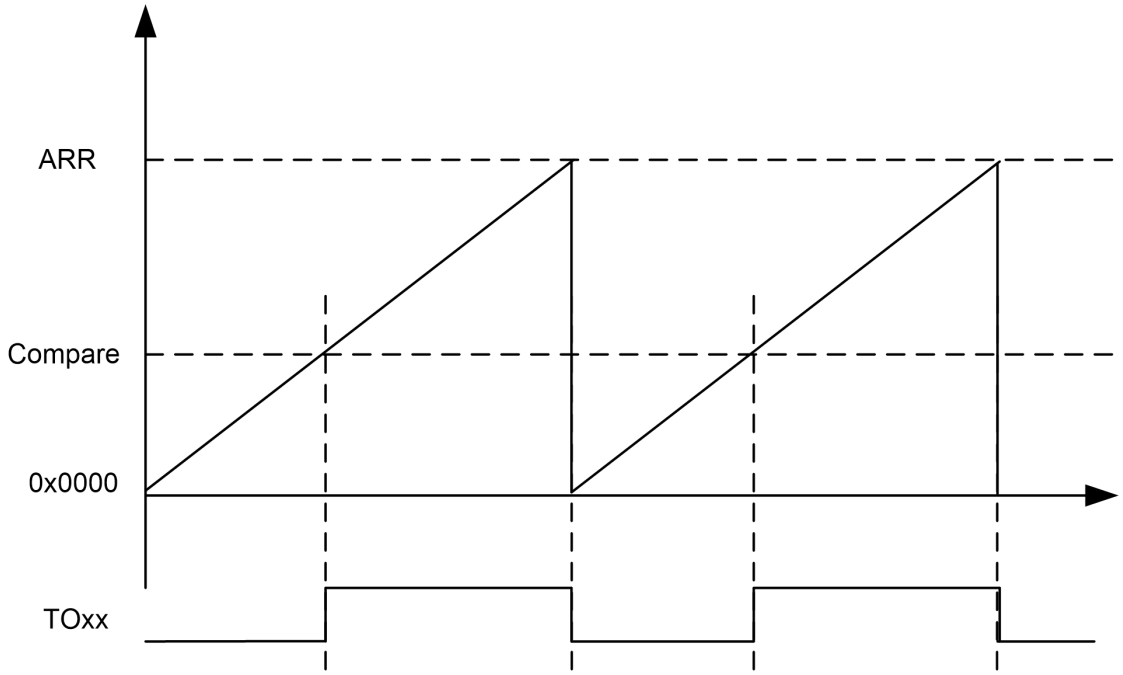


图 25-1816 位 PWM 输出示意图

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-$Duty_{PWM}$。

25.3.10 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。

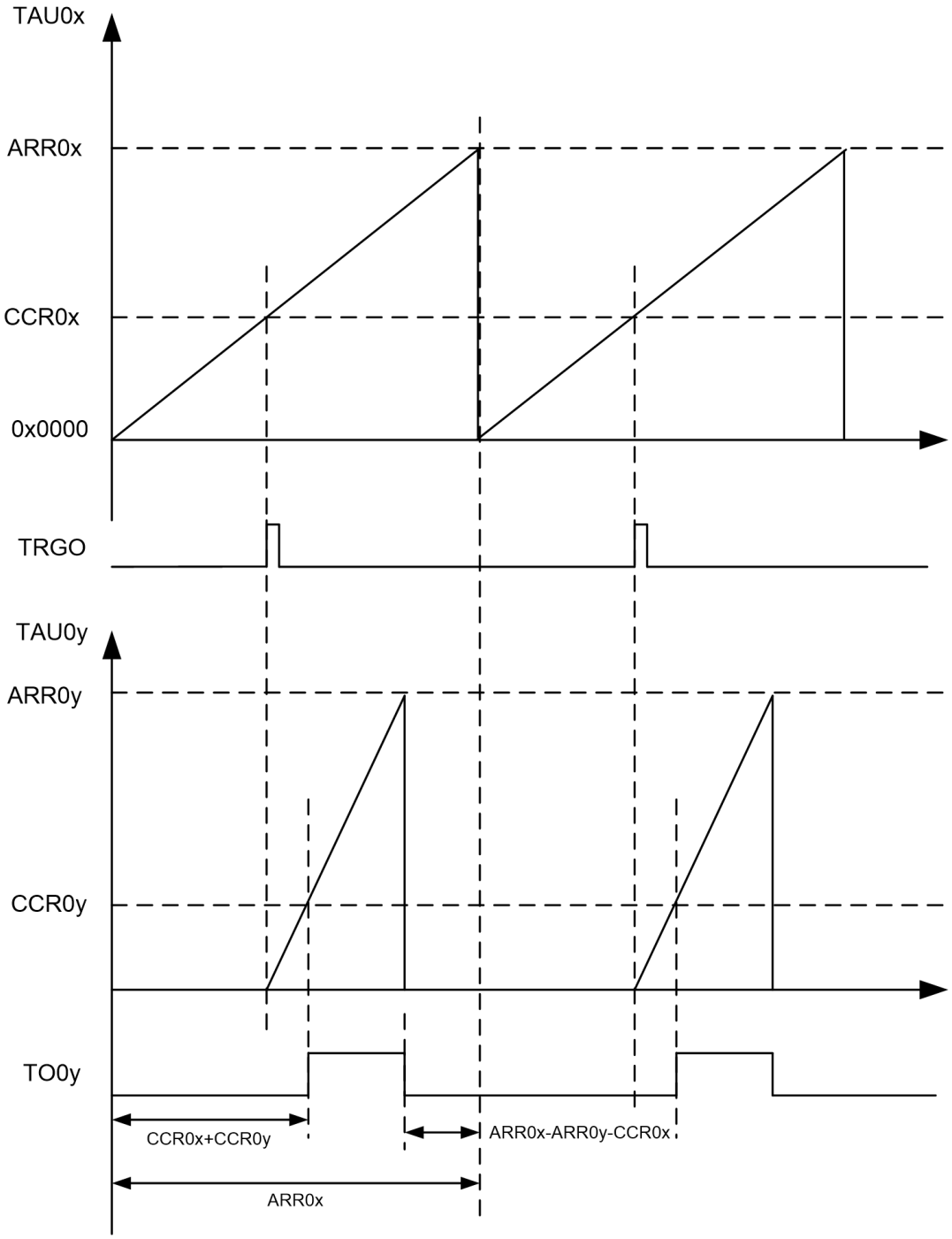


图 25- 错误!未定义书签。**非对称级联 PWM 输出示意图**

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。

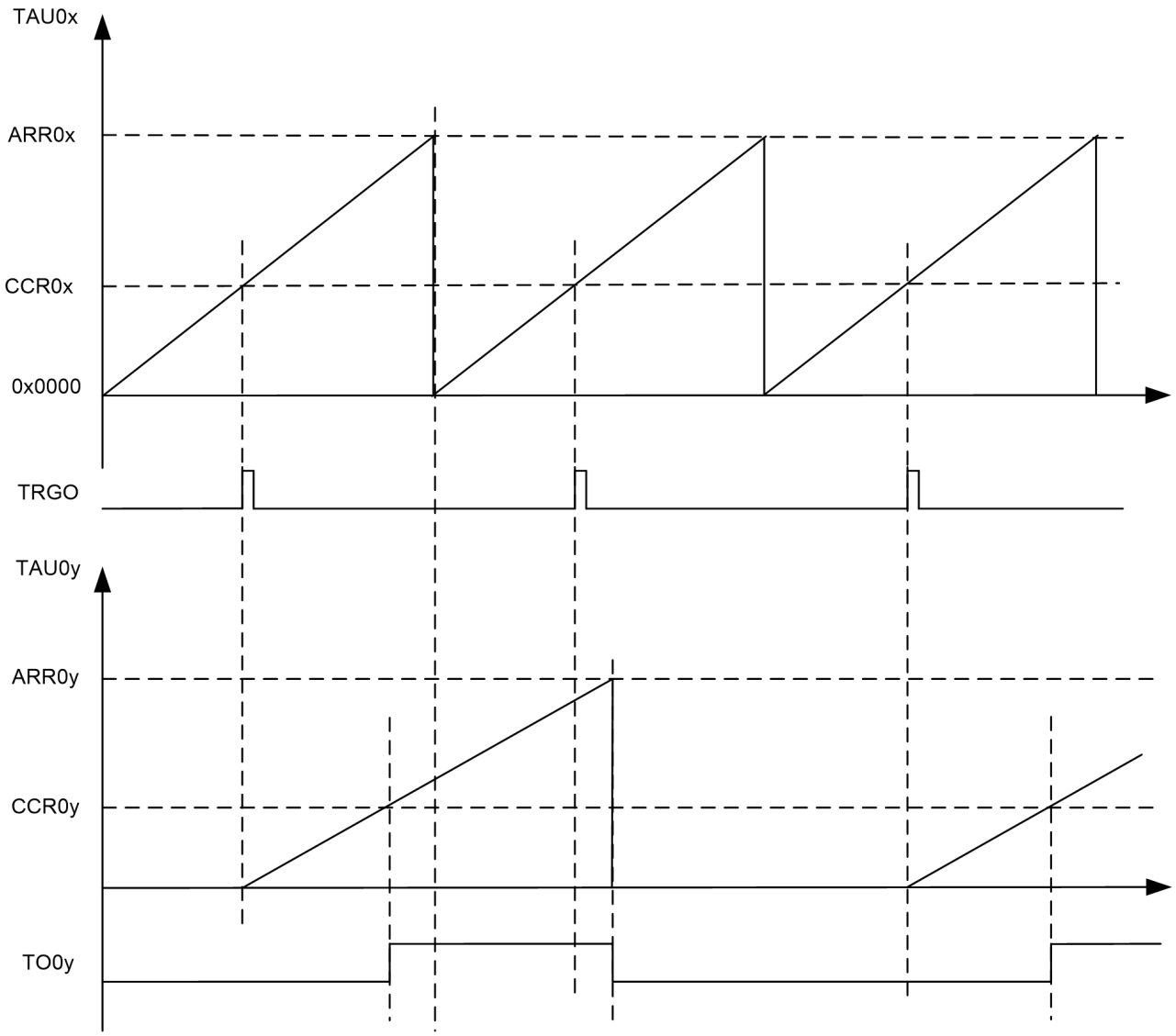


图 25-19 从机 ARR 大于主机的情况

级联定时器再搭配第三个独立定时器，可以输出两路非交叠PWM，如下图所示。

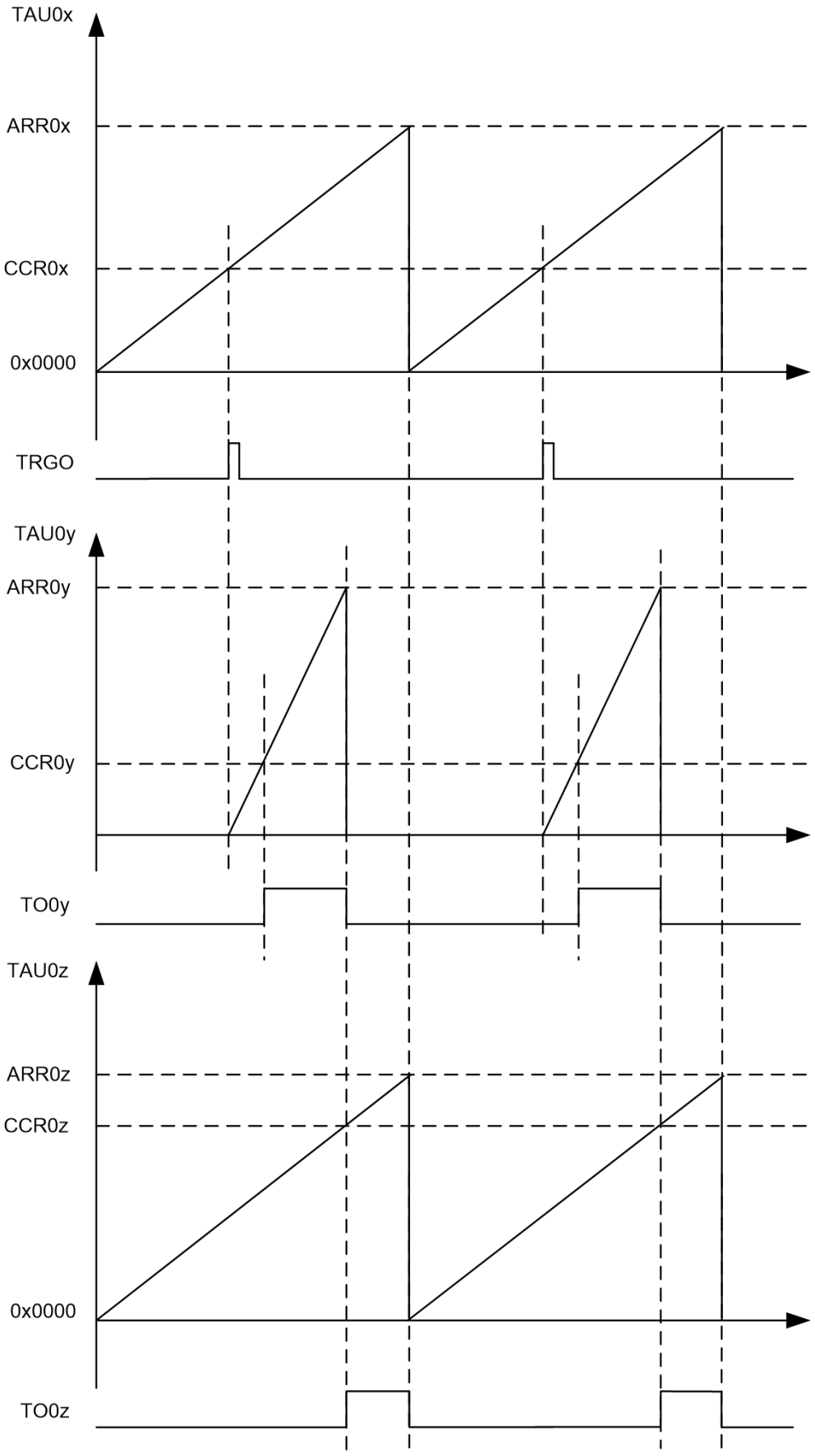


图 25-20 非对称非交叠 PWM 输出示意图

25.3.11 外部引脚输入数字滤波

扩展定时器的外部引脚输入TIxx可以选择是否使能数字滤波，以抑制信号噪声。使能数字滤波的情况下，使用APBCLK连续3次采样相同才认为是输入变化有效；不使能数字滤波时，扩展定时器直接对管脚输入进行采样。

数字滤波示意图如下：

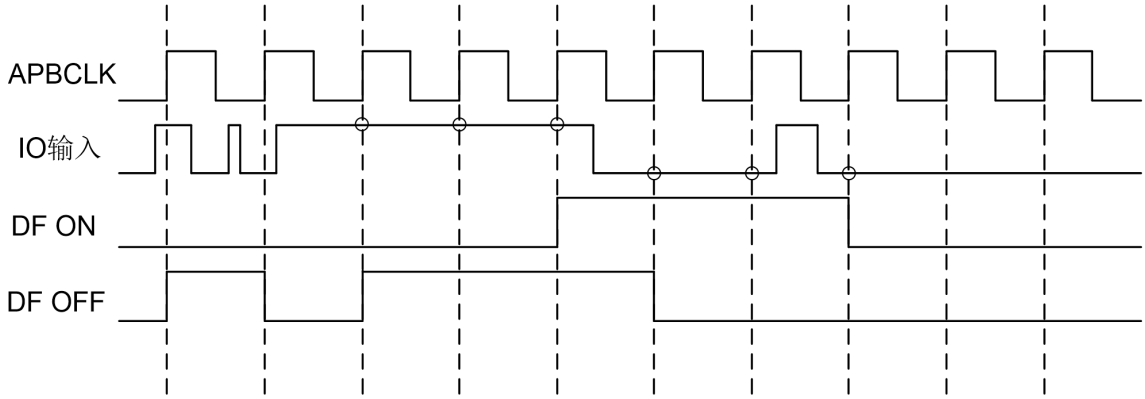


图 25-21TIxx 输入数字滤波示意图

25.4 寄存器

TAU模块起始地址：0x4001_3C00

offset 地址	名称	符号
0x00	TAU0 控制寄存器	TAU0_CR
0x04	TAU1 控制寄存器	TAU1_CR
0x08		
0x0C		
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 计数值寄存器	TAU0_TxCNT
0xD4		
0xD8		
0xDC		
0xE0		
0xE4		
0xE8		
0xEC		
0xF0	TAU1 通道 x 配置寄存器	TAU1_TxCFGR
0xF4		
0xF8		
0xFC		
0x100	TAU1 通道 x 模式寄存器	TAU1_TxMDR
0x104		
0x108		
0x10C		
0x110	TAU1 重载寄存器	TAU1_TxARR
0x114		
0x118		
0x11C		
0x120	TAU1 捕捉比较寄存器	TAU1_TxCCR
0x124		
0x128		
0x12C		
0x130	TAU1 中断使能寄存器	TAU1_TxIER
0x134		
0x138		
0x13C		
0x140	TAU1 中断标志寄存器	TAU1_TxISR
0x144		
0x148		
0x14C		
0x150	TAU1 中断标志寄存器	TAU1_TxCNT
0x154		
0x158		
0x15C		

25.4.1 TAUy 控制寄存器（TAUy_CR）

名称	TAUy_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。*

25.4.2 TAUy 通道 x 配置寄存器（TAUy_TxCFGR）

名称	TAUy_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25.3.632位级联模式
4:2	CAPSEL	捕捉源选择寄存器 000：外部引脚输入（TIxx） 其他：参见25.3.7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）

25.4.3 TAUy 通道 x 模式寄存器（TAUy_TxMDR）

名称	TAUy_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）

25.4.4 TAUy 重载寄存器（TAUy_TxARR）

名称	TAUy_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。

25.4.5 TAUy 捕捉比较寄存器（TAUy_TxCCR）

名称	TAUy_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-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR[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	CCR	捕捉比较寄存器 捕捉模式下，当捕捉

25.4.6 TAU0 中断使能寄存器（TAU0_TxIER）

名称	TAU0_TxIER (x=0~7)							
地址	0x90~0xAC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	---					CMPIE	CAPIE	OVIE
位权限	U-0					R/W-0	R/W-0	R/W-0

位号	位名	说明
31:3	---	未实现：读为 0
2	CMPIE	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	定时器捕捉中断使能 1 = 使能 0 = 禁止

位号	位名	说明
0	OVIE	定时器溢出中断使能 1 = 使能 0 = 禁止

25.4.7 TAUy 中断标志寄存器（TAUy_TxISR）

名称	TAUy_TxISR (x=0~7)							
地址	0xB0~0xCC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				EDGESTA	CMPIF	CAPIF	OVIF
位权限	U-0				R/W-0	R/W-0	R/W-0	R/W-0

位号	位名	说明
31:4	---	未实现：读为 0
3	EDGESTA	捕捉沿状态标志，软件写 1 清 0 1 = 脉冲宽度捕捉模式时表示捕捉到下沿 0 = 脉冲宽度捕捉模式时表示捕捉到上沿
2	CMPIF	比较状态，软件写 1 清 0 1 = 当前计数器的值大于等于比较寄存器的值 0 = 当前计数器的值小于比较寄存器的值
1	CAPIF	定时器捕捉产生信号，软件写1清0 1 = 捕捉到指定的沿 0 = 未捕捉到指定的沿
0	OVIF	定时器溢出信号，当计数器的值达到ARR时将置位，软件写1清0 1 = 产生计数溢出 0 = 未产生溢出

25.4.8 TAUy 计数值寄存器（TAUy_TxCNTR）

名称	TAUy_TxCNTR (x=0~7)							
地址	0xD0~0xEC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							

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

位号	位名	说明
31:16	---	未实现：读为 0
15:0	CNT	计数器当前计数值，只读

26 16 位基本定时器（BSTIM16）

26.1 概述

FM33FH0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。

26.2 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

26.3 结构框图

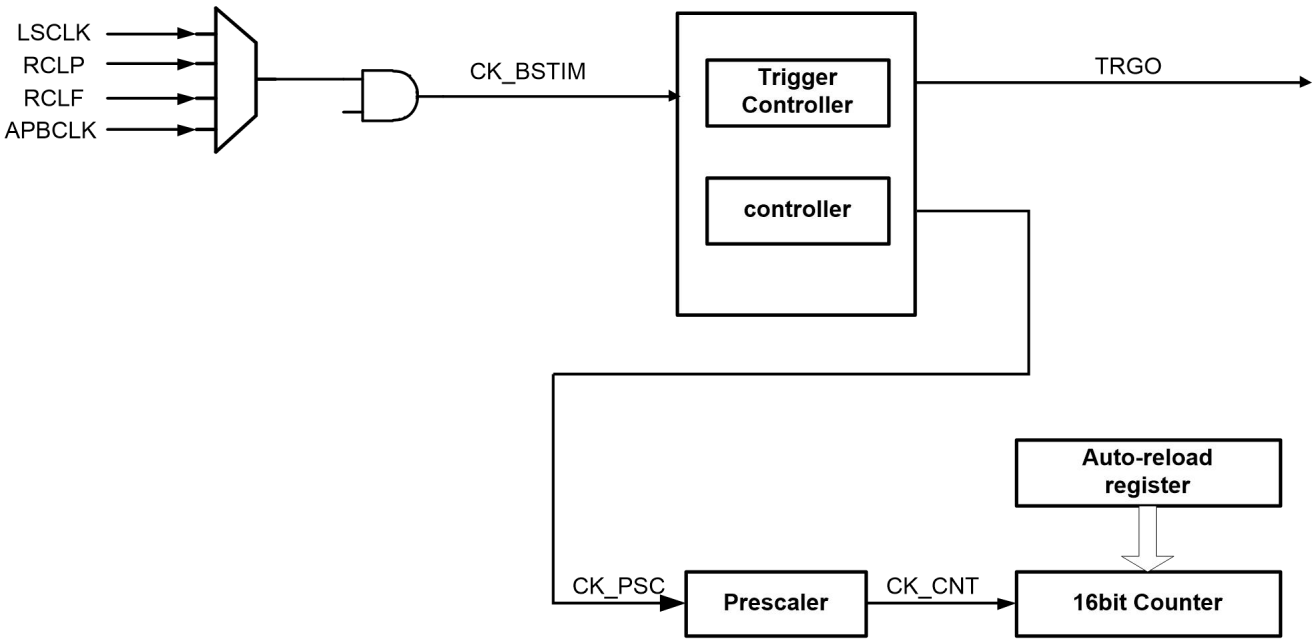


图 26-1 16 位基本定时器结构框图

26.4 功能描述

26.4.1 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。

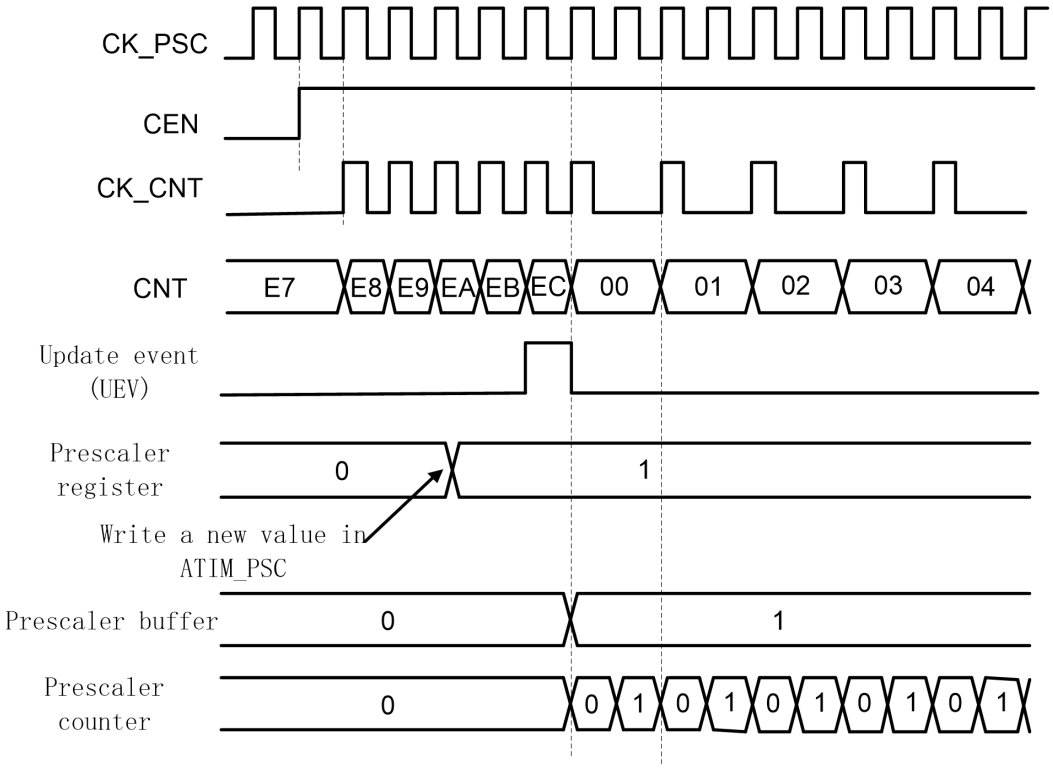


图 26-2 预分频从 1 变为 2 的波形

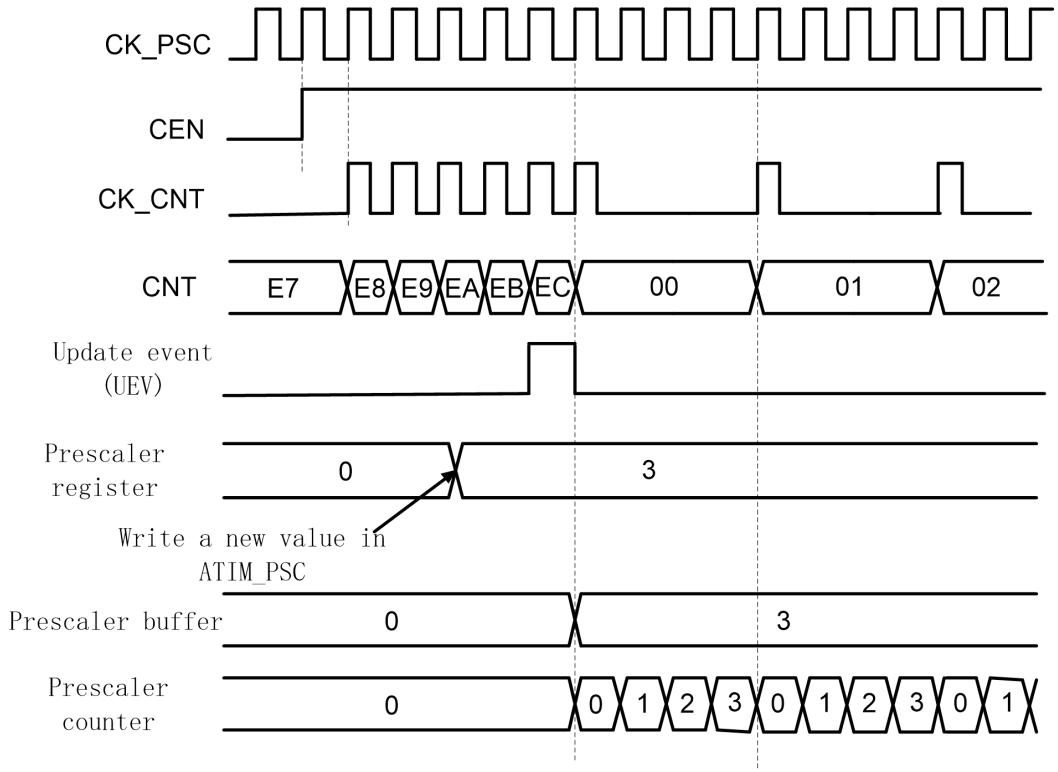


图 26-3 预分频从 1 变为 4 的波形

26.4.2 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

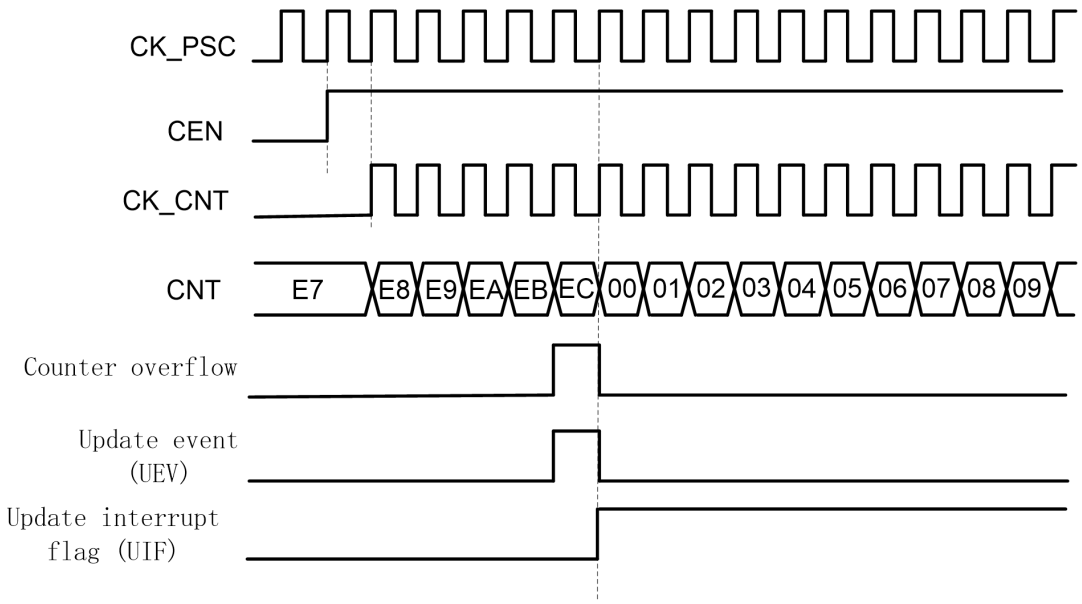


图 26-4 向上计数波形，内部时钟不分频

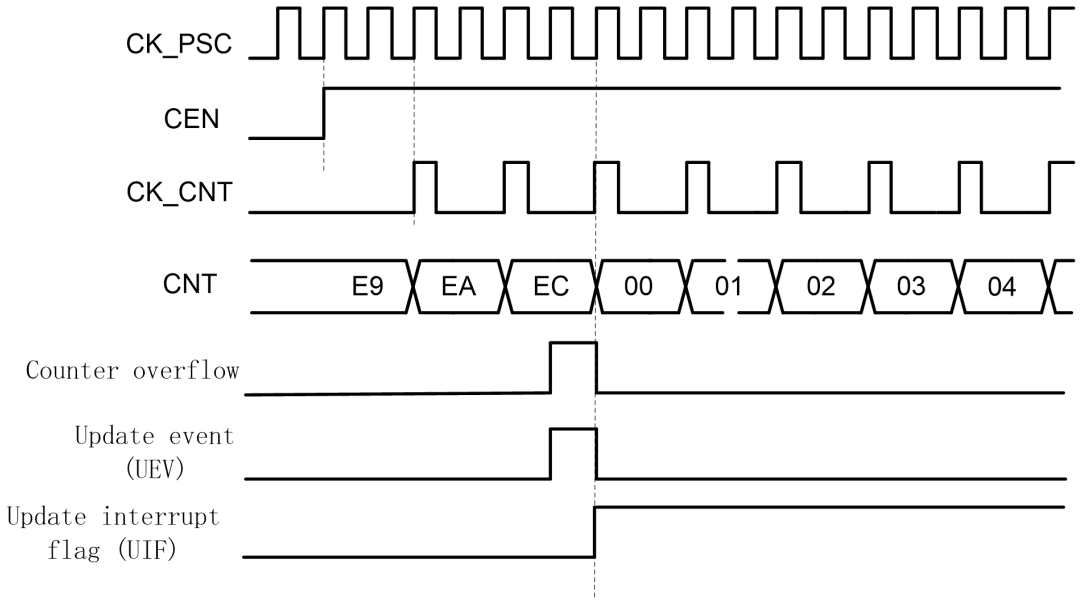


图 26-5 向上计数波形，内部时钟 2 分频

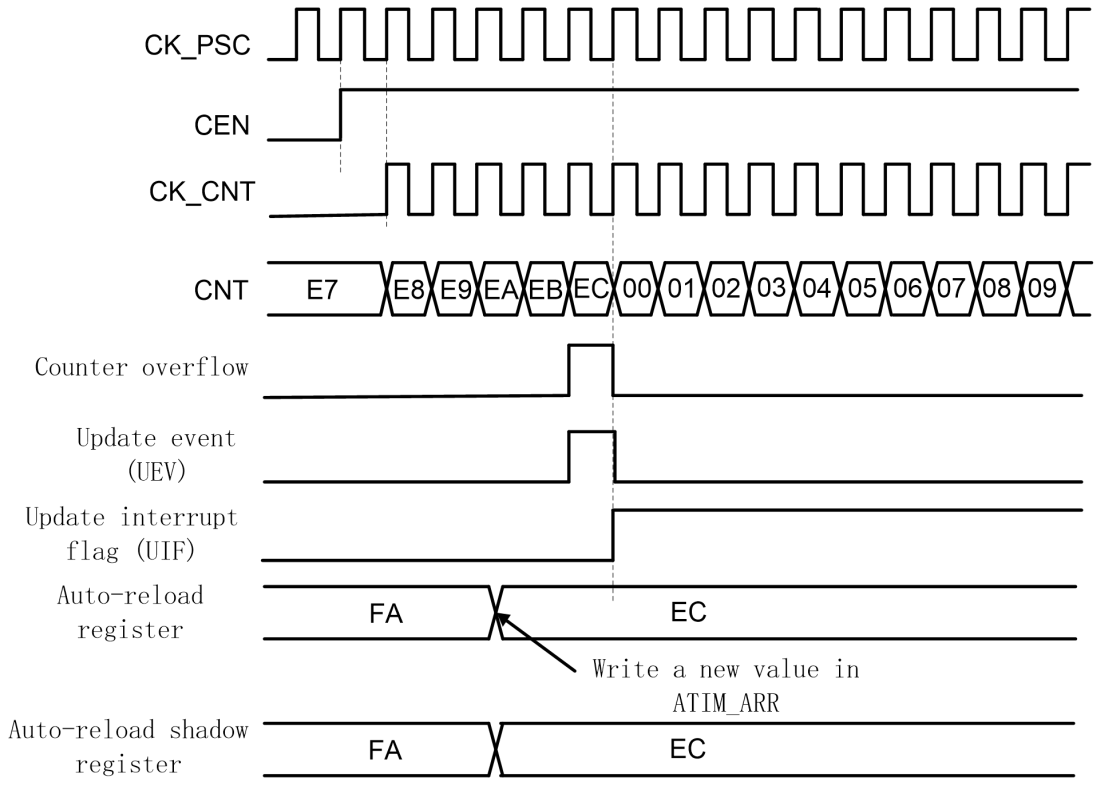


图 26-6 ARPE=0（ARR 没有预装载）时的更新事件

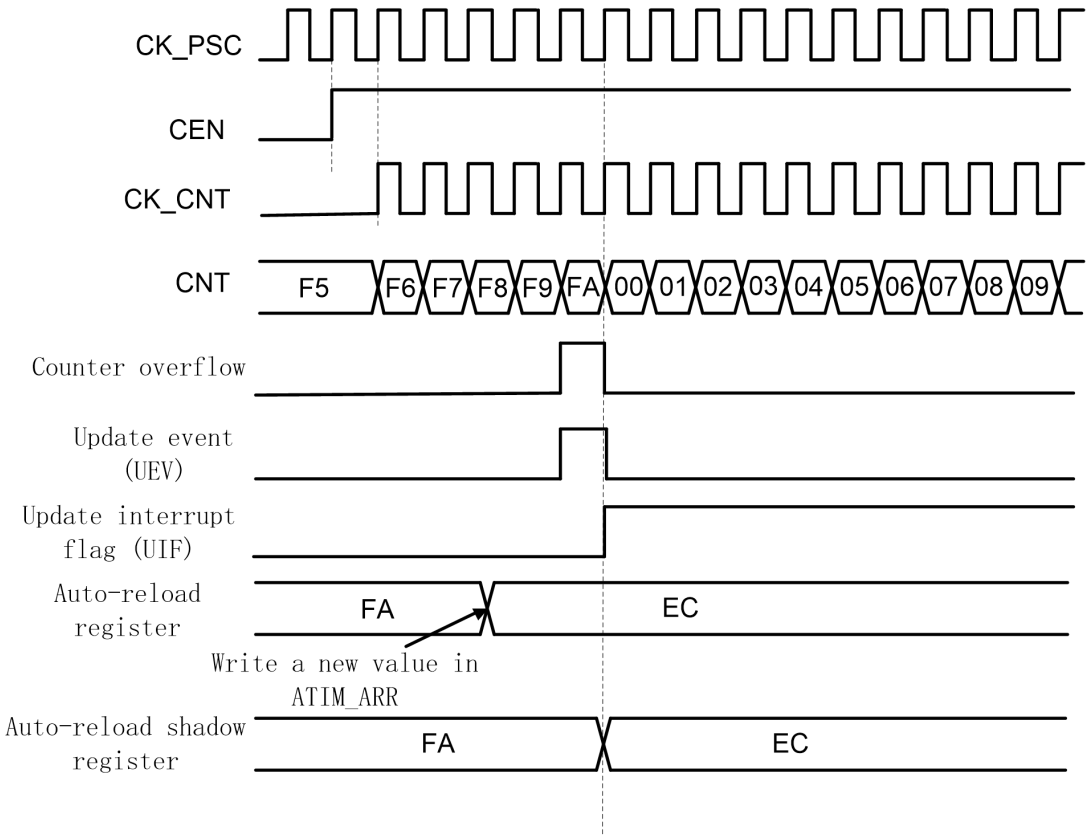


图 26-7 ARPE=1（ARR 预装载）时的更新事件

26.4.3 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。

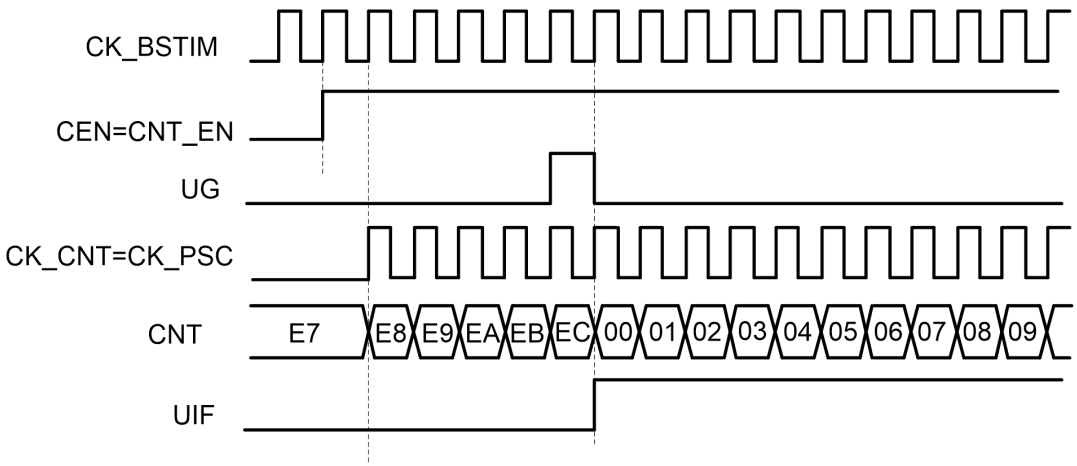


图 26-8 内部时钟源模式，时钟分频因子为 1

26.4.1 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。

26.5 寄存器

BSTIM16模块起始地址：0x4001_B400

offset 地址	名称	符号
0x00000000	BSTIM 控制寄存器 1 （BSTIM Control Register1）	BSTIM16_CR1
0x00000004	BSTIM 控制寄存器 2 （BSTIM Control Register2）	BSTIM16_CR2
0x0000000C	BSTIM 中断使能寄存器 （BSTIM Interrupt Enable Register）	BSTIM16_IER
0x00000010	BSTIM 中断标志寄存器 （BSTIM Interrupt Status Register）	BSTIM16_ISR
0x00000014	BSTIM 事件产生寄存器 （BSTIM Event Generation Register）	BSTIM16_EGR
0x00000024	BSTIM 计数器寄存器 （BSTIM Counter Register）	BSTIM16_CNT
0x00000028	BSTIM 预分频寄存器 （BSTIM Prescaler Register）	BSTIM16_PSC
0x0000002C	BSTIM 自动重载寄存器 （BSTIM Auto-Reload Register）	BSTIM16_ARR

26.5.1 BSTIM 控制寄存器 1

名称	BSTIM16_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	-							
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		
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

26.5.2 BSTIM 控制寄存器 2

名称	BSTIM16_CR2							
地址								
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MMS			-			
位权限	U-0	R/W-000			U-0			

Bit	助记符	功能描述
31:7	-	RFU，未实现，读为 0
6:4	MMS	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：BSTIM_EGR 的 UG 寄存器被用作 TRGO 001：计数器使能信号 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

26.5.3 BSTIM 中断使能寄存器

名称	BSTIM16_DIER							
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	-							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

26.5.4 BSTIM 状态寄存器

名称	BSTIM16_SR							
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	-							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

26.5.5 BSTIM 事件产生寄存器

名称	BSTIM16_EGR							
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	-							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。 *注意：由于 BSTIM 计数器工作时钟与系统总线时钟 APBCLK 是互相独立的，当软件置位 UG 寄存器后，UIF 并不会立即置位，而是需要经过 CK_BSTIM 同步后才置位，两者之间的延迟与 APBCLK 和 CK_BSTIM 之间的频率及相位关系有关，并非确定值。*

26.5.6 BSTIM 计数器寄存器

名称	BSTIM16_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	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	CNT[7:0]
位权限	RW-00000000

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	CNT	计数器值

26.5.7 BSTIMx 预分频寄存器

名称	BSTIM16_PSC							
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	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

26.5.8 BSTIMx 自动重载（auto-reload）寄存器

名称	BSTIM16_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	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

27 16 位低功耗定时器（LPTIM16）

27.1 概述

LPTIM16是16bits低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM16在在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM16甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。LPTIM16的主要特性有：

- 1 个独立的 16bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK、RCLP、APBCLK、SYSCLK
  - 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

27.2 结构框图

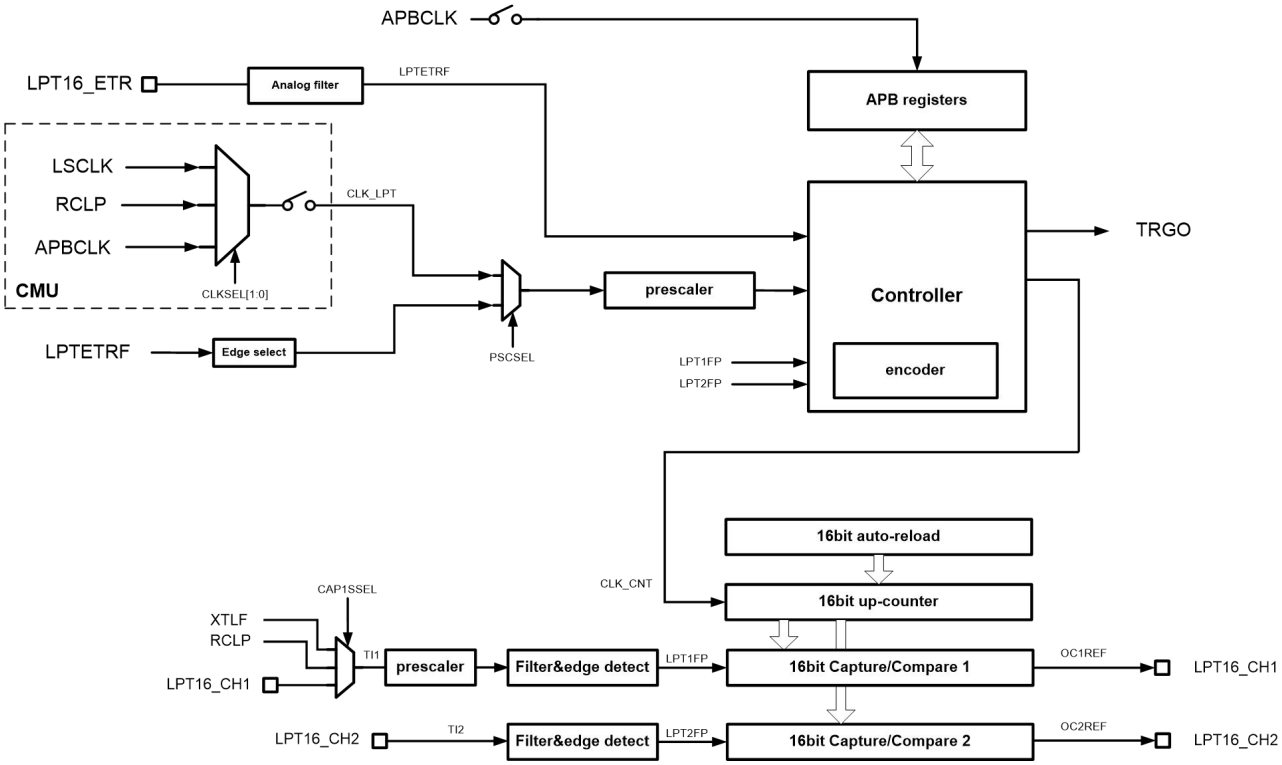


图 27-1 LPTIM16 结构框图

27.3 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寄存器可以选择LPTIM16的计数时钟源。为了保证定时器工作稳定可靠，禁止在EN为1的情况下修改计数时钟。

在选择了合适的计数时钟之后，还可以通过DIVSEL寄存器对其进行预分频，以获得更低的工作时钟频率。同样，必须在EN为0的情况下修改DIVSEL。

LPTIM16模块可以通过操作LPT16RST寄存器来复位和撤销复位，详情参见RMU章节。

27.4 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。

27.4.1 普通定时器

当LPTCFG.TMODE=00时，LPTIM16为普通定时器工作模式

- 使用多路选择后的CLK_LPT时钟计数
- 需要配置CMU模块中的OPCCON2.LPTCKS寄存器，选择合适的计数时钟
- LPTEN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LPTARR

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（LPTARR）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（LPTARR）后回到0，并自动停止，产生溢出中断，同时硬件自动清除LPTEN。

27.4.2 外部脉冲触发计数

外部脉冲触发计数模式（LPTCFG.TMODE=01）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用。LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发定时器递增。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图举例说明了LPT16_ETR触发计数，上升沿有效的情况。

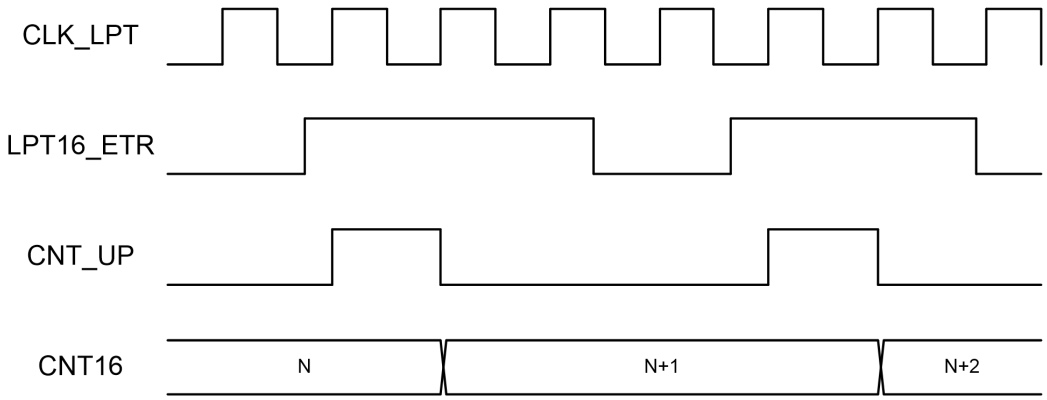


图 27-2 外部 ETR 脉冲上升沿触发计数

27.4.3 外部异步脉冲计数

外部异步脉冲计数模式（LPTCFG.TMODE=10）下，LPTIM16将LPT16_ETR引脚输入的信号直接作为计数时钟使用。这种情况下，LPTIM16全异步工作，不需要使能任何内部时钟。软件可以通过LPTCFG.EDGESEL来选择定时器使用ETR上升沿还是下降沿计数。由于这种模式下LPT16_ETR引脚上的任何干扰信号都有可能引起定时器误动作，因此推荐使能ETR输入模拟滤波功能，能够滤除大约100ns以内的glitch信号。

下图举例说明了外部异步脉冲计数，下降沿有效的情况。

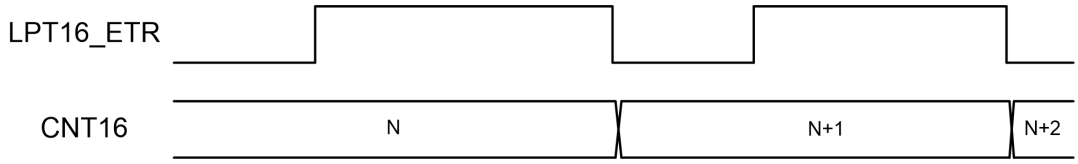


图 27-3 外部 ETR 脉冲异步计数（下降沿）

27.4.4 Timeout 模式

Timeout模式（LPTCFG.TMODE=11）下，LPTIM16将LPT16_ETR引脚输入的信号作为触发信号使用，定时器使用内部时钟CLK_LPT工作。Timeout模式下定时器启动后，不会立即开始计数，而是等待第一个LPT16_ETR信号的有效沿到来。当第一个有效沿到来后，触发定时器开始自由计数，此后每个新的ETR有效沿都会清零计数器，并重新开始计数。根据外部输入ETR信号的实际频率，合理配置计数器工作时钟和溢出上限（LPTARR），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LPTEN结束计数过程。

LPT16_ETR信号首先经过LPTIM16工作时钟采样、同步后，可以在其上升沿、下降沿或上升下降沿触发计数器清零重新计数。由于需要使用CLK_LPT采样并识别LPT16_ETR信号的变化沿，这里要求ETR输入信号有效电平宽度必须大于CLK_LPT周期的2倍。软件可以通过LPTCFG.TRIGCFG寄存器设置LPTIM16对LPT16_ETR的哪个边沿计数。

下图是timeout模式下使用LPT16_ETR上升沿清零，并最终溢出的例子。

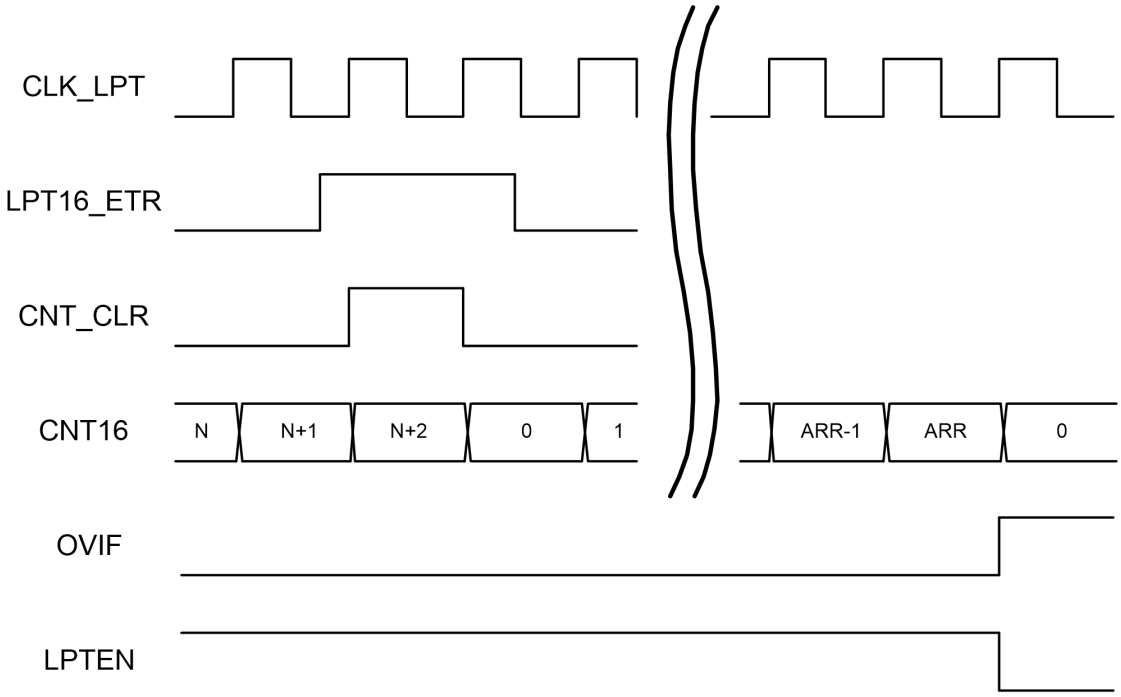


图 27-4TimeOut 模式

使用TimeOut模式，并使能LPTIM中断，在芯片休眠时可以实现外部信号触发的超时唤醒功能。此时只要LPT16_ETR管脚上有周期性信号输入，就能使芯片保持休眠，而一旦超过规定时间内没有新的触发信号到来，LPTIM超时溢出中断将唤醒芯片。

27.5 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。

27.5.1 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（LPTARR）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。LPTCFG.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LPTCFG.CCxS配置为10，此时LPT_CHx成为输出通道，相应的GPIO自动使能输出功能（软件需将GPIO配置为数字外设功能）。

下图是PWM输出，POLARITY=1的例子。

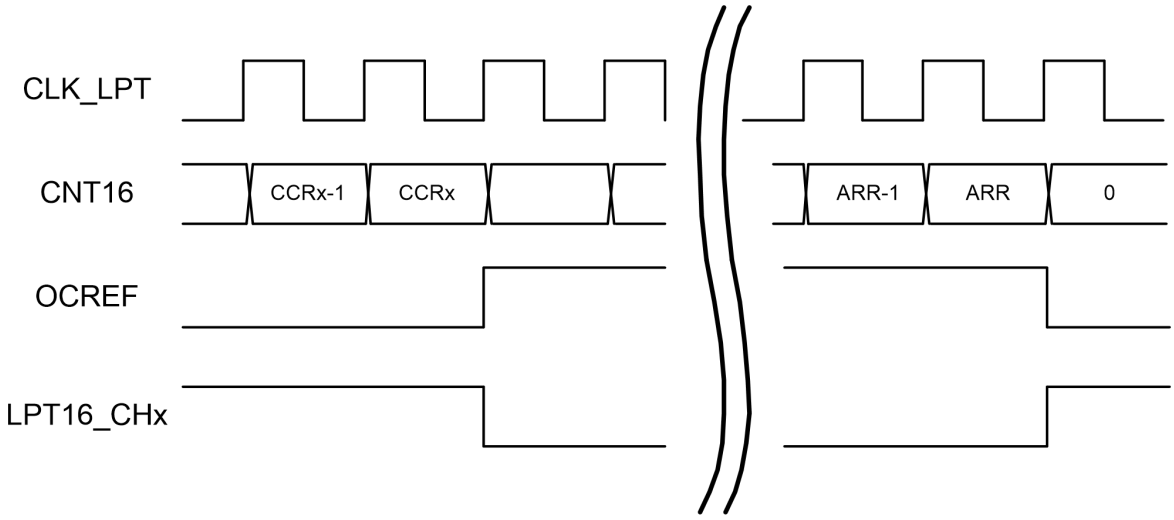


图 27-5PWM 输出

27.5.2 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。

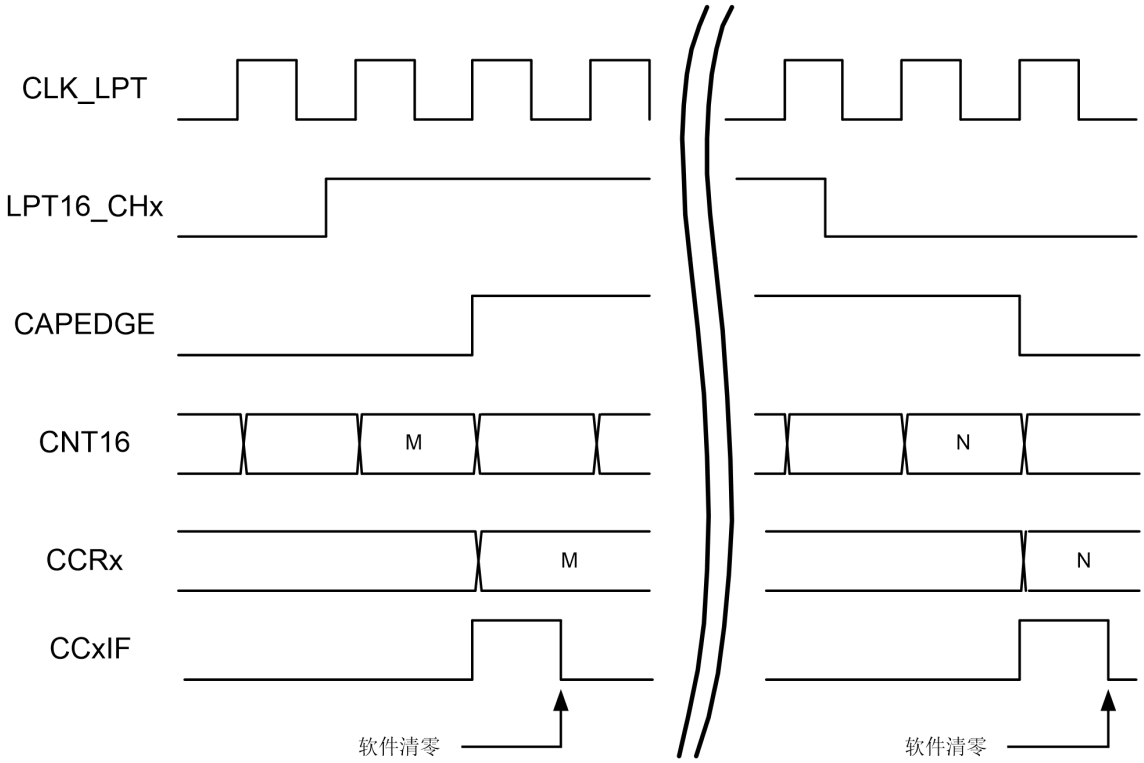


图 27-6 输入信号边沿捕捉

27.5.1 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。

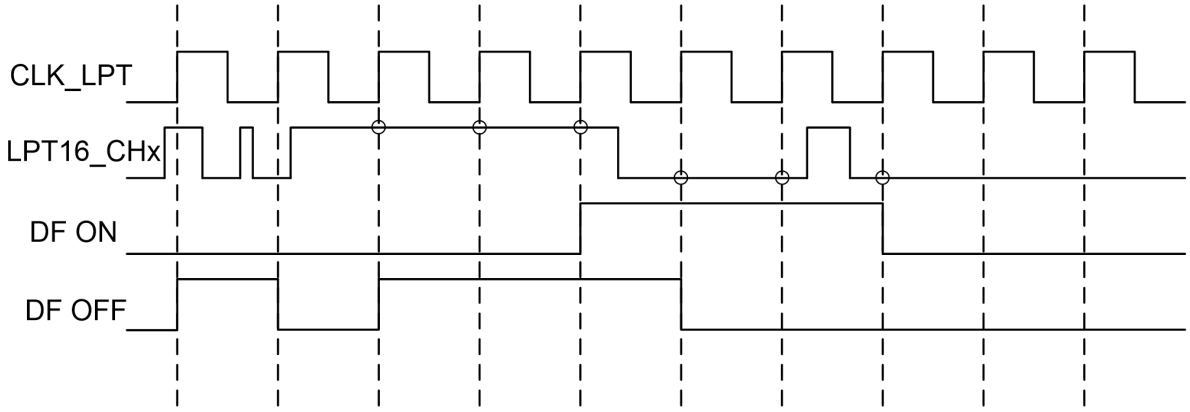


图 27-7 通道输入数字滤波

27.6 正交编码器

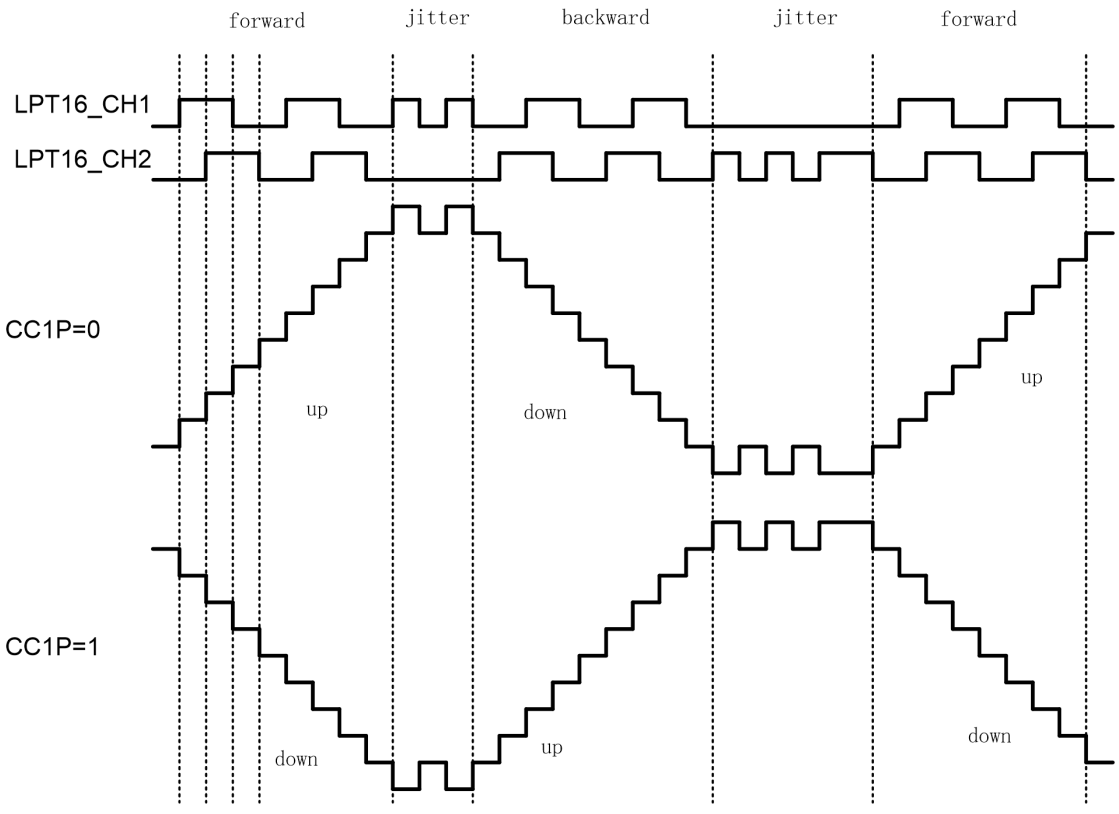
编码器接口模式涉及到两个外部输入信号，LPTIM16根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

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 27-1 encoder interface 计数方式

比如在计数器以LPT1FP信号为时钟计数时，如果LPT1FP上升沿采样到LPT2FP为高电平，则计数器递减；如果LPT1FP下降沿采样到LPT2FP为高电平，则计数器递增。（同步计数，CLK_LPT采样取沿后进行计数，参考ATIM设计）

下图为编码器在LPT1FP和LPT2FP边沿同时计数的例子。由于CC1P能够控制输入信号LPT16_CH1的极性，所以当CC1P为1时输入取反，相应的计数器增减方向也会反向。



Example of counter operation in encoder interface mode

图 27-8 编码器模式下的计数器操作实例

正交编码信号使用LPT16_CHx输入，可以选择对输入信号是否进行数字滤波。注意LPT16_CH1通道上还有预分频电路，在进行正交编码计数时应关闭预分频。

QDIR寄存器用于表示当前输入信号相位关系，当CH1相位领先于CH2时，QDIR=0，反之QDIR=1，此寄存器只读。

当CH1和CH2发生相位翻转时，QDCIF中断标志置位，表示出现换向，如果QDCIE寄存器置位，则产生中断事件。

编码模式输入通道需进行如下设置：

- 在GPIO模块中，配置相应管脚为LPT16_CH1，LPT16_CH2功能
- 选择输入通道，配置LPTIM_CCSR.CC1S=01，LPTIM_CCSR.CC2S=01
- 选择计数有效沿（输入信号极性），配置LPTIM_CCSR.CC1P和CC2P
- 设定编码器模式，配置LPTIM_CFGR.QEMD寄存器
- 使能LPTIM16，置位LPTIM_CR.EN

27.7 寄存器

模块基地址：0x4001_3400

offset 地址	名称	符号
0x00	LPTIM 配置寄存器 (LPTIM Config Register)	LPTIM16_CFGR
0x04	LPTIM 计数寄存器 （LPTIM Counter Register）	LPTIM16_CNT
0x08	LPTIM 捕捉比较控制和状态寄存器 （ LPTIM Capture/Compare Control and Status Register）	LPTIM16_CCSR
0x0C	LPTIM 目标值寄存器 （LPTIM Auto-Reload Register）	LPTIM16_ARR
0x10	LPTIM 中断使能寄存器 （LPTIM Interrupt Enable Register）	LPTIM16_IER
0x14	LPTIM 中断标志寄存器 （LPTIM Interrupt Status Register）	LPTIM16_ISR
0x18	LPTIM 控制寄存器 （LPTIM Control Register）	LPTIM16_CR
0x20	LPTIM 捕捉比较寄存器 1 （LPTIM Capture/Compare Register1）	LPTIM16_CCR1
0x24	LPTIM 捕捉比较寄存器 2 （LPTIM Capture/Compare Register2）	LPTIM16_CCR2

27.7.1 LPTIM 配置寄存器（LPTIM16_CFGR）

名称	LPTIM16_CFGR							
offset	0x00							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							ETR_AFEN
位权限	U-0							R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-					MMS		
位权限	U-0					R/W-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	PSCSEL	-	DIVSEL			-	
位权限	U-0	R/W-0	U-0	R/W-000			U-0	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		QEMD		ONST	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

27.7.2 LPTIM 计数值寄存器（LPTIM16_CNT）

名称	LPTIM16_CNT							
offset	0x04							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT[7:0]							
位权限	R-0000 0000							

位号	位名	说明
31:16	**--**	未实现：读为0
15:0	**CNT**	16bit 计数器当前计数值(Counter 16bits-wide)

27.7.3 LPTIM 捕捉比较控制和状态寄存器（LPTIM16_CCSR）

名称	LPTIM16_CCSR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CAP1PSC						CAP1SSEL	
位权限	R/W-00 0000						R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		CAP2EDGE	CAP1EDGE	-	-	CC2P	CC1P
位权限	U-0		R-0	R-0	U-0	U-0	R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-		CC2DF	CC1DF	CAPCFG2		CAPCFG1	
位权限	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	CC2S	CC1S
位权限	U-0	R/W-00	R/W-00

位号	位名	说明
31:26	**CAP1PSC**	通道1输入预分频 (Capture channel 1 prescaler) 0x00：不分频 0x3F：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：下降沿 1：上升沿
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：下降沿 1：上升沿
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）

27.7.4 LPTIM 目标值寄存器（LPTIM16_ARR）

名称	**LPTIM16_ARR**							
offset	0x0C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	未实现：读为0
15:0	**ARR**	自动重载目标寄存器(Auto-Reload Register) 当计数器计数值等于 ARR 时，计数器回到初值重新开始向上计数

27.7.5 LPTIM 中断使能寄存器（LPTIM16_IER）

名称	**LPTIM16_IER**
offset	0x10

位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-						OVR2IE	OVR1IE
位权限	U-0						R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	QDCIE	-			CC2IE	CC1IE
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:10	--	**未实现：读为0**
9	**OVR2IE**	通道 2 捕捉溢出中断使能(Channel 2 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
8	**OVR1IE**	通道 1 捕捉溢出中断使能(Channel 1 Over-Capture Interrupt Enable) 1：允许中断 0：禁止中断
7	**TRIGIE**	外部触发到来中断使能位(External Trigger Interrupt Enable) 1：外部触发到来中断使能 0：外部触发到来中断禁止
6	**OVIE**	计数器溢出中断使能位(Counter Over-Flow Interrupt Enable) 1：计数器溢出中断使能 0：计数器溢出中断禁止
5	**QDCIE**	Quadrature Decoder 方向改变中断使能 (Quad-decoder Interrupt Enable) 1：使能 QDCIF 中断 0：禁止 QDCIF 中断
4:2	--	**未实现：读为0**
1	**CC2IE**	捕捉/比较通道 2 中断使能位(Capture/Compare channel 2 Interrupt Enable) 1：捕捉/比较通道 2 中断使能 0：捕捉/比较通道 2 中断禁止
0	**CC1IE**	捕捉/比较通道 1 中断使能位(Capture/Compare channel 1 Interrupt Enable) 1：捕捉/比较通道 1 中断使能 0：捕捉/比较通道 1 中断禁止

27.7.6 LPTIM 中断标志寄存器（LPTIM16_ISR）

名称	**LPTIM16_ISR**							
offset	0x14							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名			QDIR	-			CAP2OVR	CAP1OVR
位权限			R-0	U-0			R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	QDCIF	-			CC2IF	CC1IF
位权限	R/W-0	R/W-0	R/W-0	U-0			R/W-0	R/W-0

位号	位名	说明
31:14	--	**未实现：读为0**
13	**QDIR**	Quadrature Decoder 方向寄存器 (Quad-decoder Direction) 0：CH1 领先于 CH2 1：CH1 滞后于 CH2
12:10	--	**未实现：读为0**
9	**CAP2OVR**	通道 2 捕捉溢出，硬件置位，软件写 1 清零(Channel 2 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC2IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
8	**CAP1OVR**	通道 1 捕捉溢出，硬件置位，软件写 1 清零(Channel 1 Over-Capture Interrupt Flag) 1：输入捕捉模式下，CC1IF 为 1 时出现新的捕捉，发生 overrun 0：没有发生 overrun
7	**TRIGIF**	外部触发到来中断标志位，写 1 清零(External Trigger Interrupt Flag) 1：外部触发到来中断产生 0：无中断产生
6	**OVIF**	计数器溢出中断使能位，写 1 清零(Counter Over-Flow Interrupt Flag) 1：计数器溢出中断产生 0：无中断产生
5	**QDCIF**	Quadrature Decoder 方向改变中断，硬件置位，软件写 1 清零(Quad-decoder Direction Change Interrupt Flag) 0：没有改变方向 1：方向改变
4:2	--	**未实现：读为0**
1	**CC2IF**	捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 2 Interrupt Flag) 1：计数器值和比较值 2 匹配，或者发生捕捉事件 0：无中断产生
0	**CC1IF**	捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零(Capture/Compare channel 1 Interrupt Flag) 1：计数器值和比较值 1 匹配，或者发生捕捉事件 0：无中断产生

27.7.7 LPTIM 控制寄存器（LPTIM16_CR）

名称	LPTIM16_CR
offset	0x18

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

位号	位名	说明
31:1	--	**未实现：读为0**
0	**EN**	LPTIM 使能位(LPTIM Enable) 1：使能计数器计数 0：禁止计数器计数

27.7.8 LPTIM 捕捉比较寄存器 1（LPTIM16_CCR1）

名称	LPTIM16_CCR1							
offset	0x1C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	--	**未实现：读为0**
15:0	**CCR1**	捕捉/比较值寄存器 1 (Channel1 Capture/Compare Register)

27.7.9 LPTIM 捕捉比较寄存器 2（LPTIM16_CCR2）

名称	LPTIM16_CCR2							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2[7:0]							
位权限	R/W-0000 0000							

位号	位名	说明
31:16	**--**	**未实现：读为0**
15:0	**CCR2**	捕捉/比较值寄存器 2 (Channel2 Capture/Compare Register)

28 实时时钟（RTC）

28.1 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

28.2 结构框图

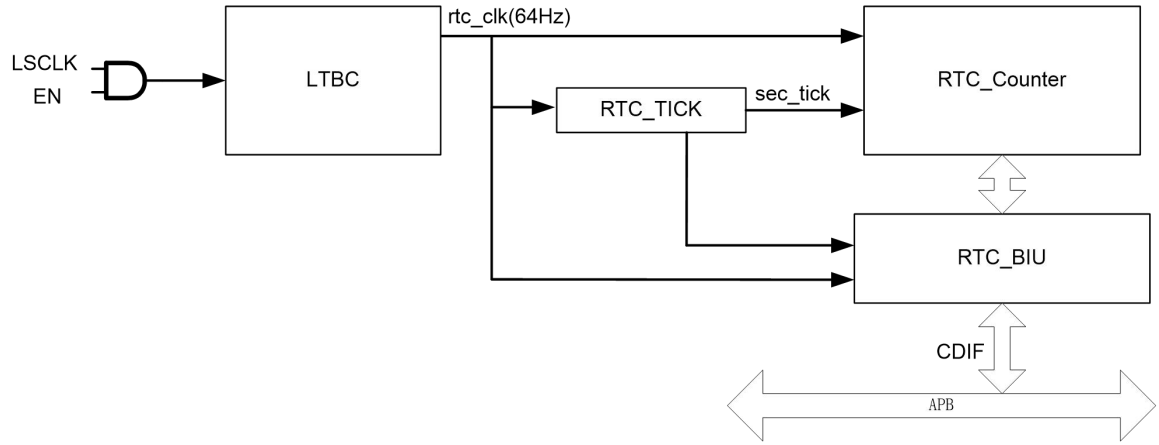


图 28-1RTC 结构框图

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。

28.3 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。

28.3.1 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

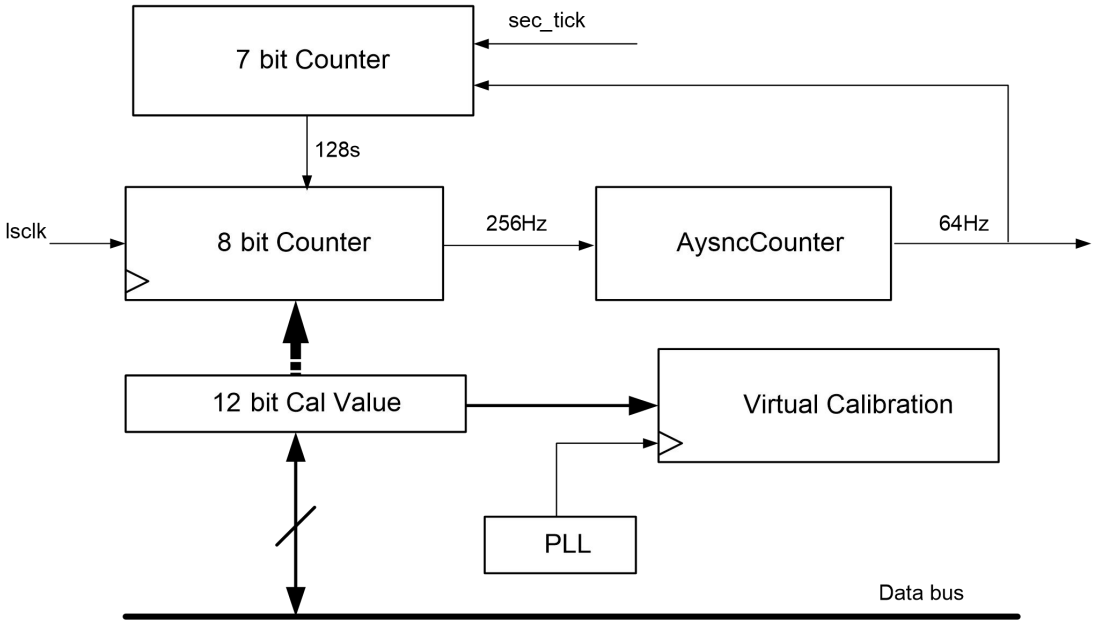


图 28-2LTBC 结构框图

28.3.2 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。FM33FH0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。

28.3.3 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

28.3.4 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。

28.3.5 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 FM33FH0 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，FM33FH0 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。

28.3.6 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。

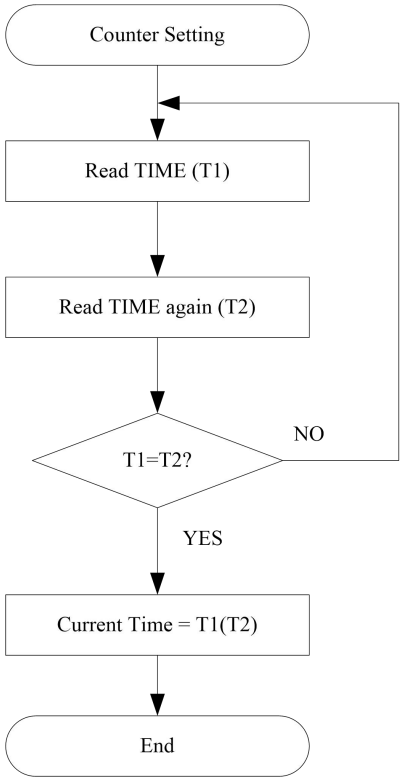


图 28-3RTC 时间读取流程图

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。

28.3.7 闰年判断

FM33FH0 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。

28.4 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

28.4.1 RTC 写使能寄存器（RTC_WER）

名称	RTC_WER							
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	-							

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。

28.4.2 RTC 中断使能寄存器（RTC_IER）

名称	RTC_IER							
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	-			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

28.4.3 RTC 中断标志寄存器（RTC_ISR）

名称	RTC_ISR							
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	-			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

28.4.4 BCD 时间秒寄存器（RTC_BCDSEC）

名称	RTC_BCDSEC							
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	-	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)

28.4.5 BCD 时间分钟寄存器（RTC_BCDMIN）

名称	RTC_BCDMIN							
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	-							
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)

28.4.6 BCD 时间小时寄存器（RTC_BCDHOUR）

名称	RTC_BCDHOUR							
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	-		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)

28.4.7 BCD 时间天寄存器（RTC_BCDDAY）

名称	RTC_BCDDAY							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAY					
位权限	U-0		R/W-xx xxxx					

位号	助记符	功能描述
31:6	-	RFU：未实现，读为 0
5:0	DAY	天数值，BCD 格式。(Binary-Coded Decimal format Date Register)

28.4.8 BCD 时间星期寄存器（RTC_BCDWEEK）

名称	RTC_BCDWEEK
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	-					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)

28.4.9 BCD 时间月寄存器（RTC_BCDMONTH）

名称	RTC_BCDMONTH							
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	-			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)

28.4.10 BCD 时间年寄存器（RTC_BCDYEAR）

名称	RTC_BCDYEAR							
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	-							
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)

28.4.11 闹钟寄存器（RTC_ALARM）

名称	RTC_ALARM							
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	-		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)

28.4.12 RTC 时间信号输出寄存器（RTC_TMSEL）

名称	RTC_TMSEL							
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	-				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

28.4.13 LTBC 数值调整寄存器（RTC_ADJUST）

名称	RTC_ADJUST							
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	-						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
8:0	ADJUST	LTBC 补偿调整数值（Time Adjust）

28.4.14 毫秒计数值寄存器（RTC_SBSCNT）

名称	RTC_SBSCNT							
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	-							
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)

28.4.15 RTC 控制寄存器（RTC_CR）

名称	RTC_CR							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							EN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	EN	RTC 使能寄存器 (RTC enable) 0：关闭 RTC（内部走时时钟关闭） 1：使能 RTC

29 模数转换器（ADC）

29.1 概述

FM33FH0 带有最高 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 24 个外部输入通道，单端输入
- 内部采样通道
  - 温度传感器
  - 内部基准测量
  - DAC 输出测量
  - TSI LDO 输出测量
  - VREFP 和 VREFN 自检通道
- 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

29.2 结构框图

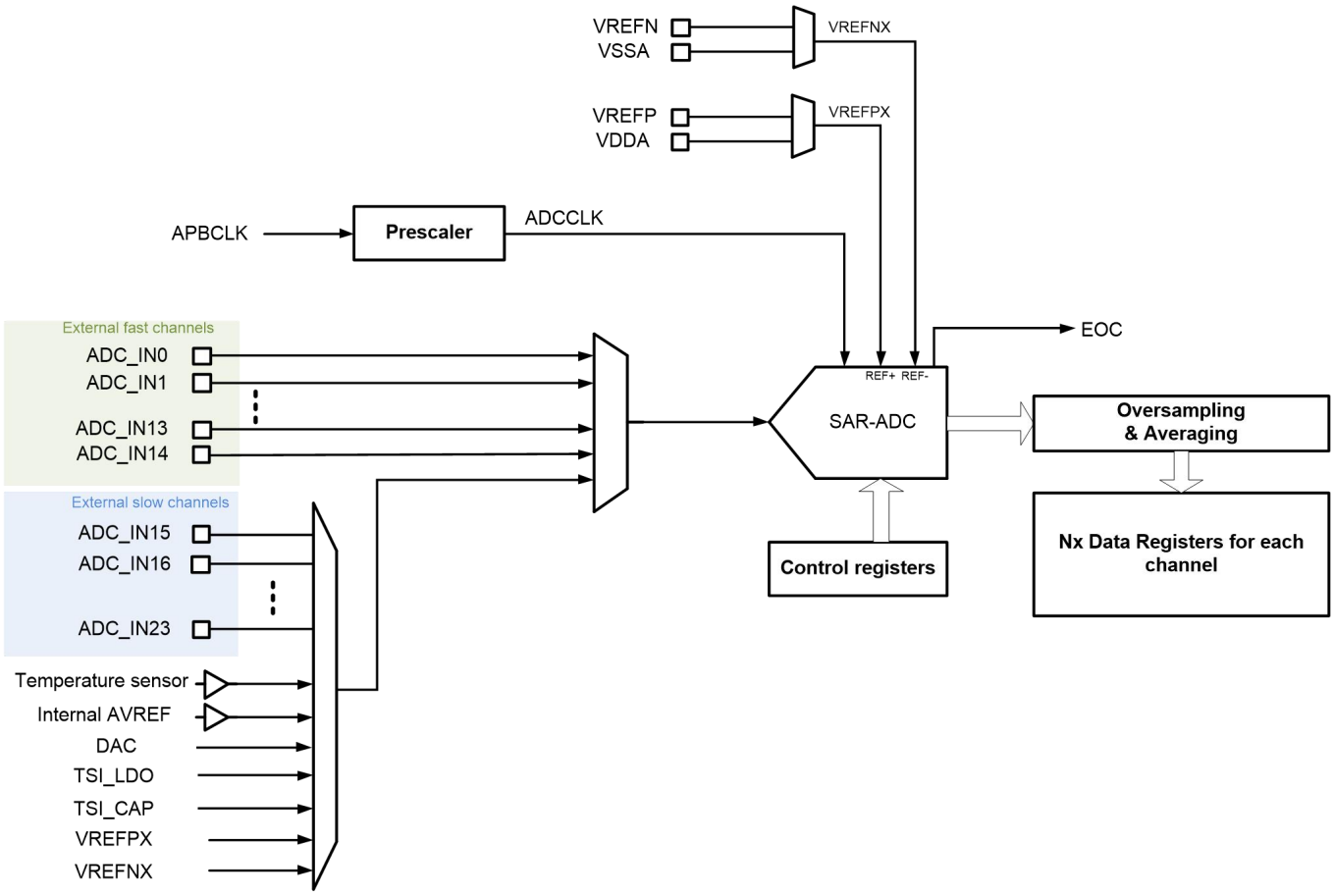


图 29-1 ADC 结构框图

29.3 输入通道

ADC 最大支持 3 个内部通道和 26 个外部通道（不支持差分输入）。

通道	IO	CHID	说明
ADC_IN0	PG2	0	外部快速通道，单端输入
ADC_IN1	PG0	1	
ADC_IN2	PD10	2	
ADC_IN3	PD9	3	
ADC_IN4	PD8	4	
ADC_IN5	PD7	5	
ADC_IN6	PD6	6	
ADC_IN7	PD5	7	
ADC_IN8	PD4	8	
ADC_IN9	PD3	9	
ADC_IN10	PD2	10	
ADC_IN11	PD1	11	
ADC_IN12	PD0	12	
ADC_IN13	PG6	13	

通道	IO	CHID	说明	
ADC_IN14	PF0	14		
ADC_IN15	PG11	15	外部慢速通道，单端输入 ***注意，当使用 ADC_IN24 和 ADC_IN25 通道时，必须将 ADC 基准源选择为 VDDA 和 VSSA***	
ADC_IN16	PG10	16		
ADC_IN17	PG7	17		
ADC_IN18	PC15	18		
ADC_IN19	PC14	19		
ADC_IN20	PC13	20		
ADC_IN21	PC12	21		
ADC_IN22	PC11	22		
ADC_IN23	PC10	23		
TempSensor	N/A	24	温度传感器采样通道	内部慢速通道
AVREF		25	内部 1.0V 基准源采样通道	
DAC		26	DAC 输出	
TSI_CAP		27	TSI 电容测量通道，可以测量 TSCAP 、 CTANK 、 CINTA、CINTB 上的充放电波形	
-		28	-	
-		29	-	
-		30	-	
VREFPX		29	内部校准通道，快速	
VREFNX		30	内部校准通道，快速	
VREFNX		31	内部校准通道，慢速	

表 29-1 ADC 输入通道

29.4 单端输入

ADC 仅支持单端输入模式。

单端模式下 ADC 转换的是单个输入引脚对地的电压值，输入幅度范围是 0~VREF+(VDDA)。为了避免可能的波形削顶风险导致输入失真，一般建议输入信号最大幅值不超过 0.95*VREF+。

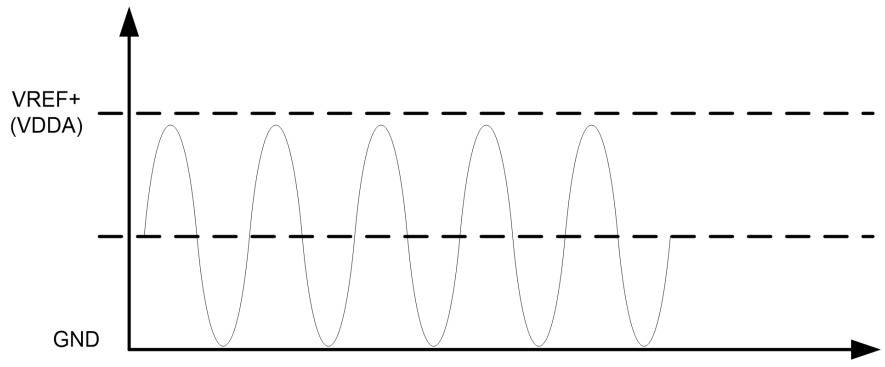


图 29-2 单端输入

29.5 时钟与复位

29.5.1 时钟

ADC 控制器为同步设计，模块只有一个 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，在奇数分频后会产生半周期时序。

在 APBCLK 时钟高于 32M 时，必须使用 2 及以上分频。

29.5.2 复位

模块使用 RMU 送来的外设复位信号，但是和 ADC 工作相关的寄存器，会加入 ADC_EN 寄存器进行复位，即 ADC_EN 无效时，这些寄存器也会被复位。同时 ADC_EN 也会对 ADC 模拟电路进行复位。

29.6 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前台校准，以获得更好的性能。校准后无需重新校准，除非芯片发生全局复位，或者电源电压、温度发生较大变化。Offset 校准通过软件置位 CALEN 启动，校准过程包含多个采样转换周期，周期数由 OSCAL_CYCLE 寄存器配置，一般推荐配置为 8，即完成校准需要 128 个 ADCCLK 周期。校准完成后置位 EOCAL 标志寄存器。

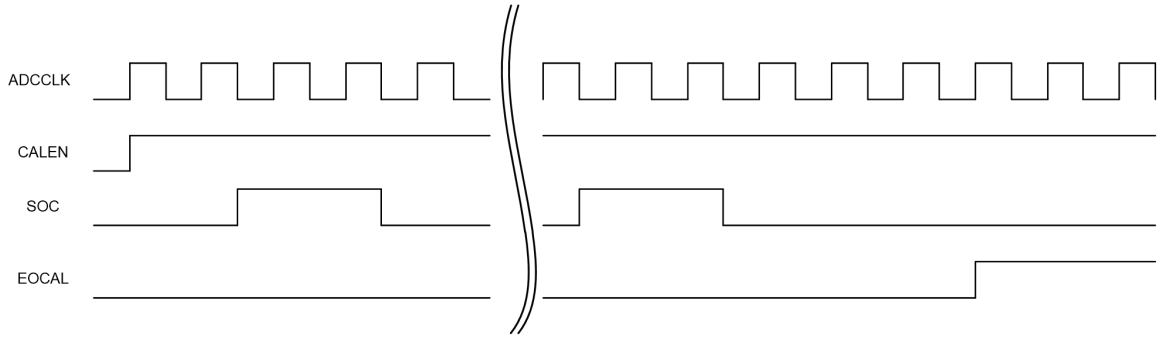


图 29-3 ADC 校准时序

采样转换时，通过 SOC 信号启动 ADC 采样，SOC 高电平宽度控制了 ADC 采样时间，SOC 变低后启动转换，转换周期为 14 个 ADC_CLK 周期，采样时间可配置，最短 2 个 ADC_CLK，最长 512 个 ADC_CLK。

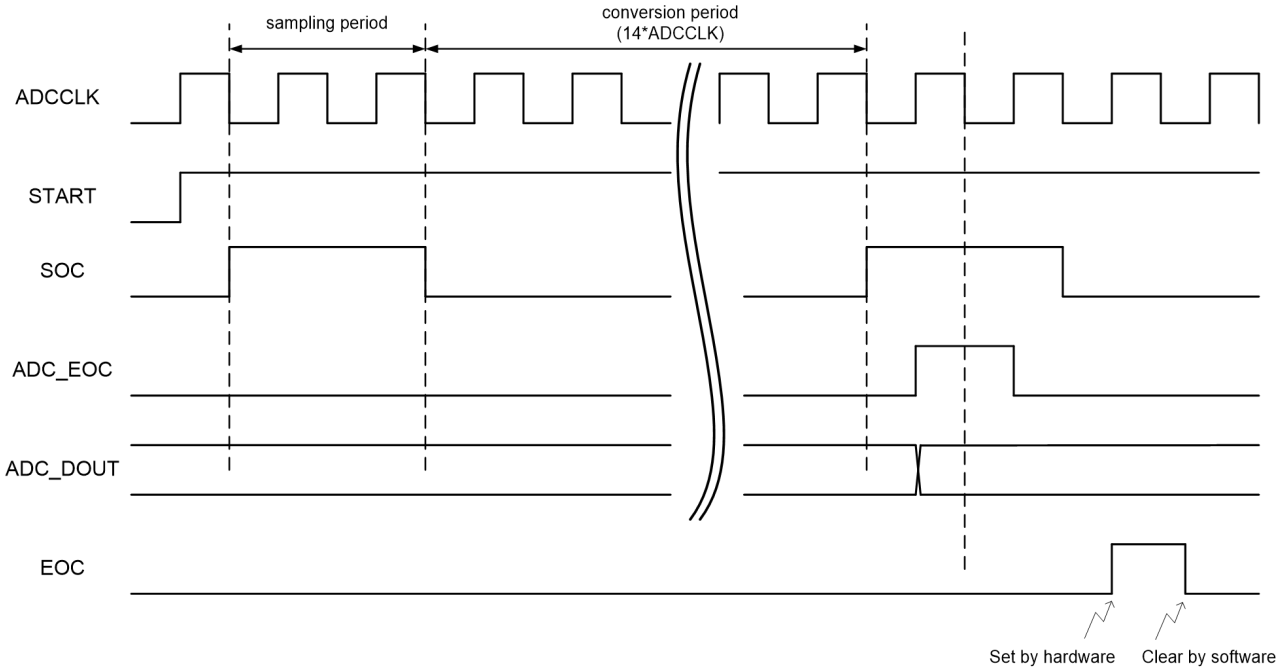


图 29-4 ADC 采样转换时序

多通道转换序列的时序示意图如下：

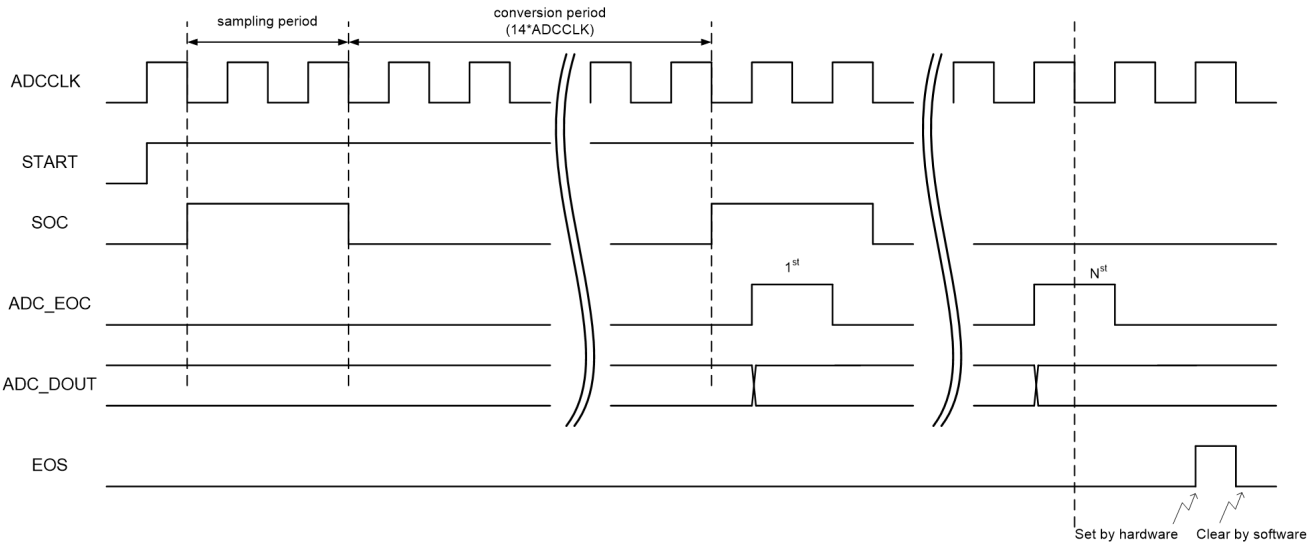


图 29-5 ADC 采样序列时序

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。

29.7 功能描述

29.7.1 ADC 转换队列（ACQ）

29.7.1.1 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，

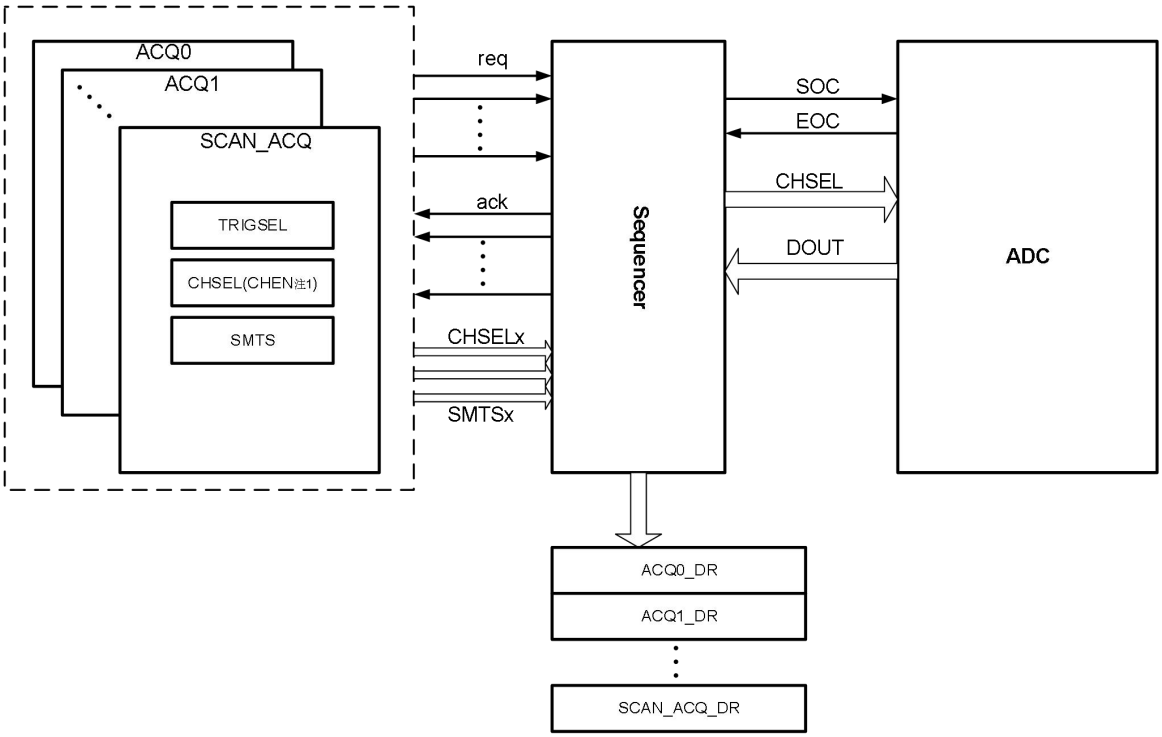


图 29-6ACQ 队列控制器

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

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，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。

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。

29.7.1.2 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。

29.7.1.3 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

29.7.2 注入 ACQ 转换

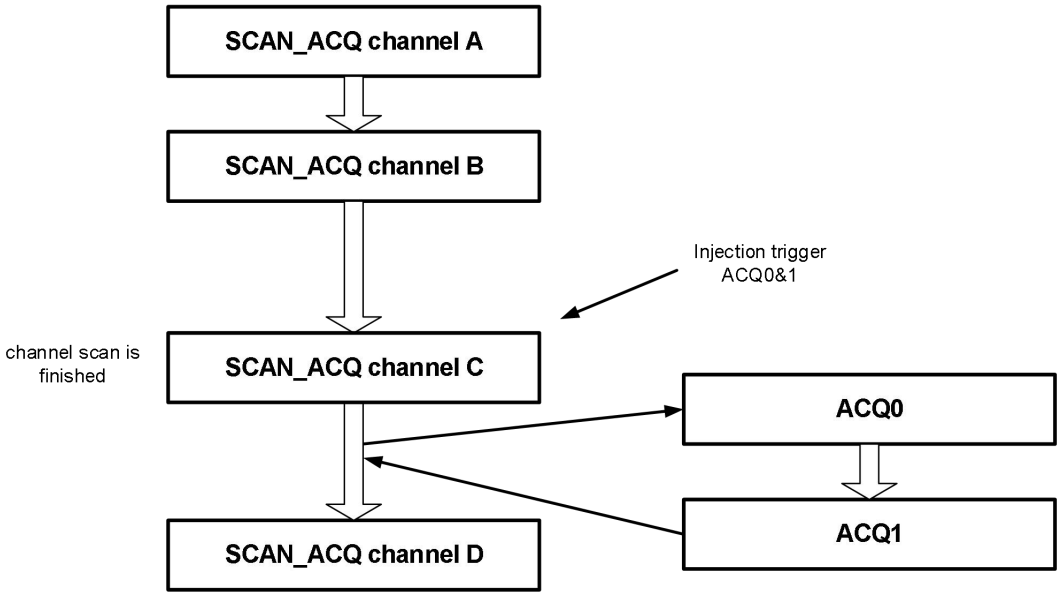
注入 ACQ 可以定义自己的采样转换配置，每个注入 ACQ 有自己的配置和使能，可以选取自己的触发进行自由触发，触发后进行优先转换，转换完成后有自己独立的标志寄存器，转换结果写入对应的通道数据寄存器中。注入 ACQ 仅响应自己的触发，并且一次触发转换一次，如果需要周期性转换注入 ACQ，需选取周期性的触发源。

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。

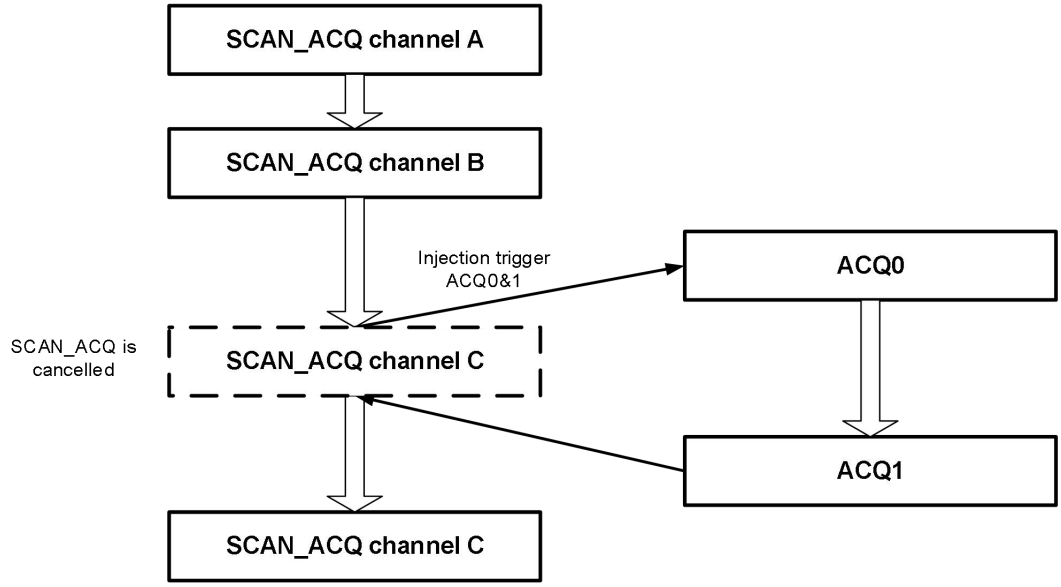
延迟注入模式，SCAN_ACQ 会执行完当前的采样转换再响应注入 ACQ 的工作，注入 ACQ 完成后，SCAN_ACQ 会继续工作，由于没有打断，看起来就像注入 ACQ 插入了 SCAN_ACQ 的工作流程中。

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：

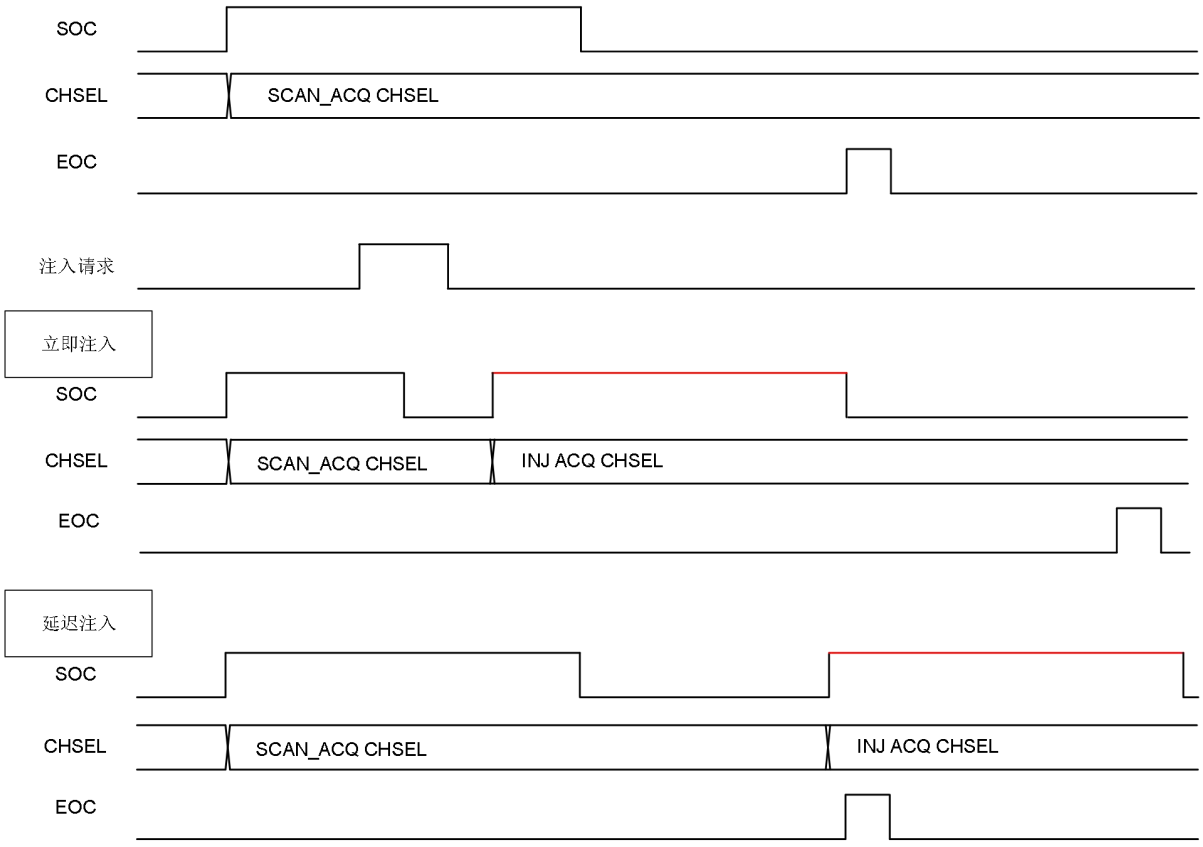


图 29-7 注入时序

29.7.3 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。

29.7.3.1 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

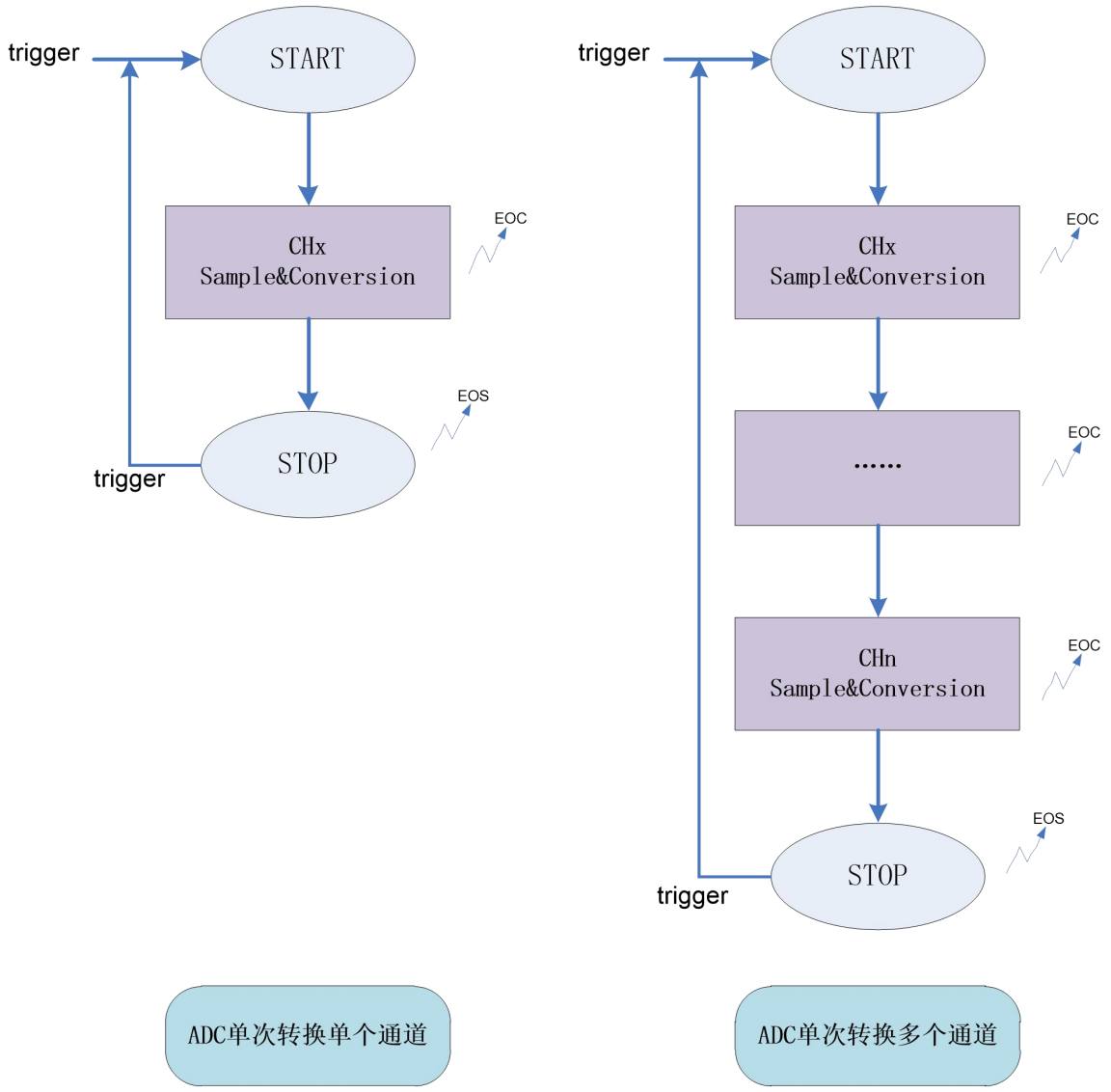


图 29-8SCAN_ACQ 单次转换全自动触发模式

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

……

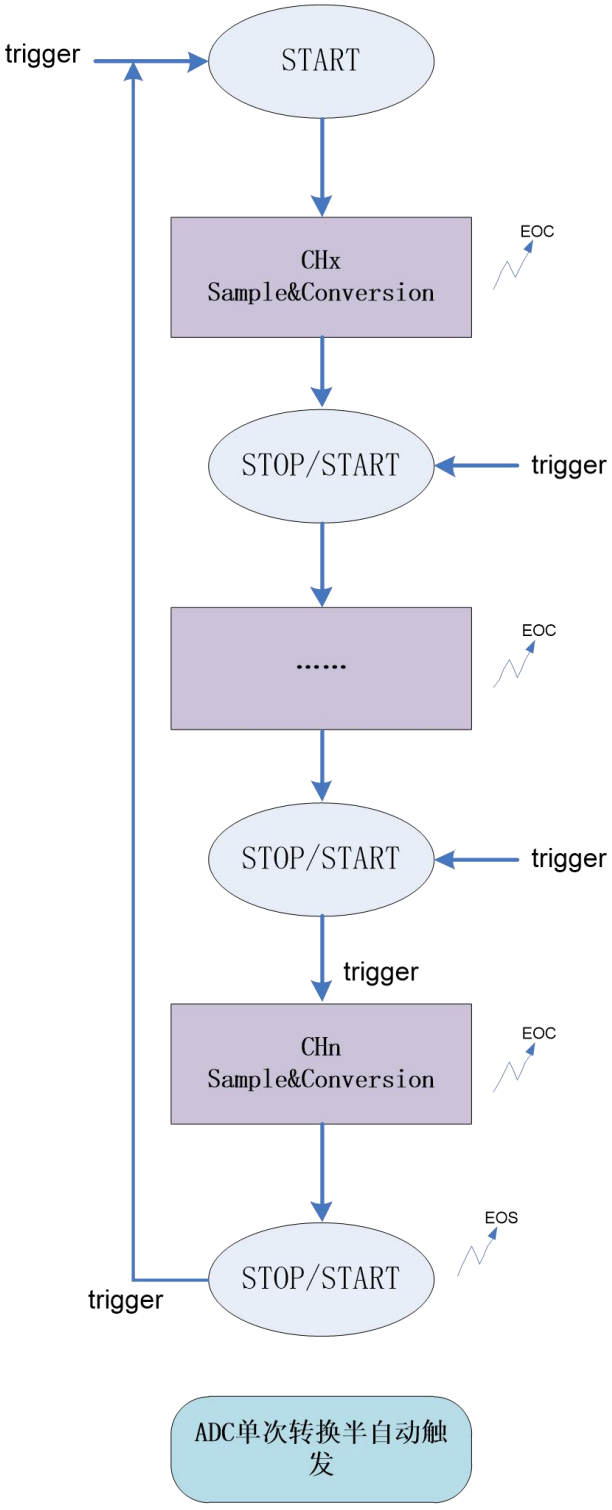


图 29-9SCAN_ACQ 单次转换半自动触发模式

29.7.3.2 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。

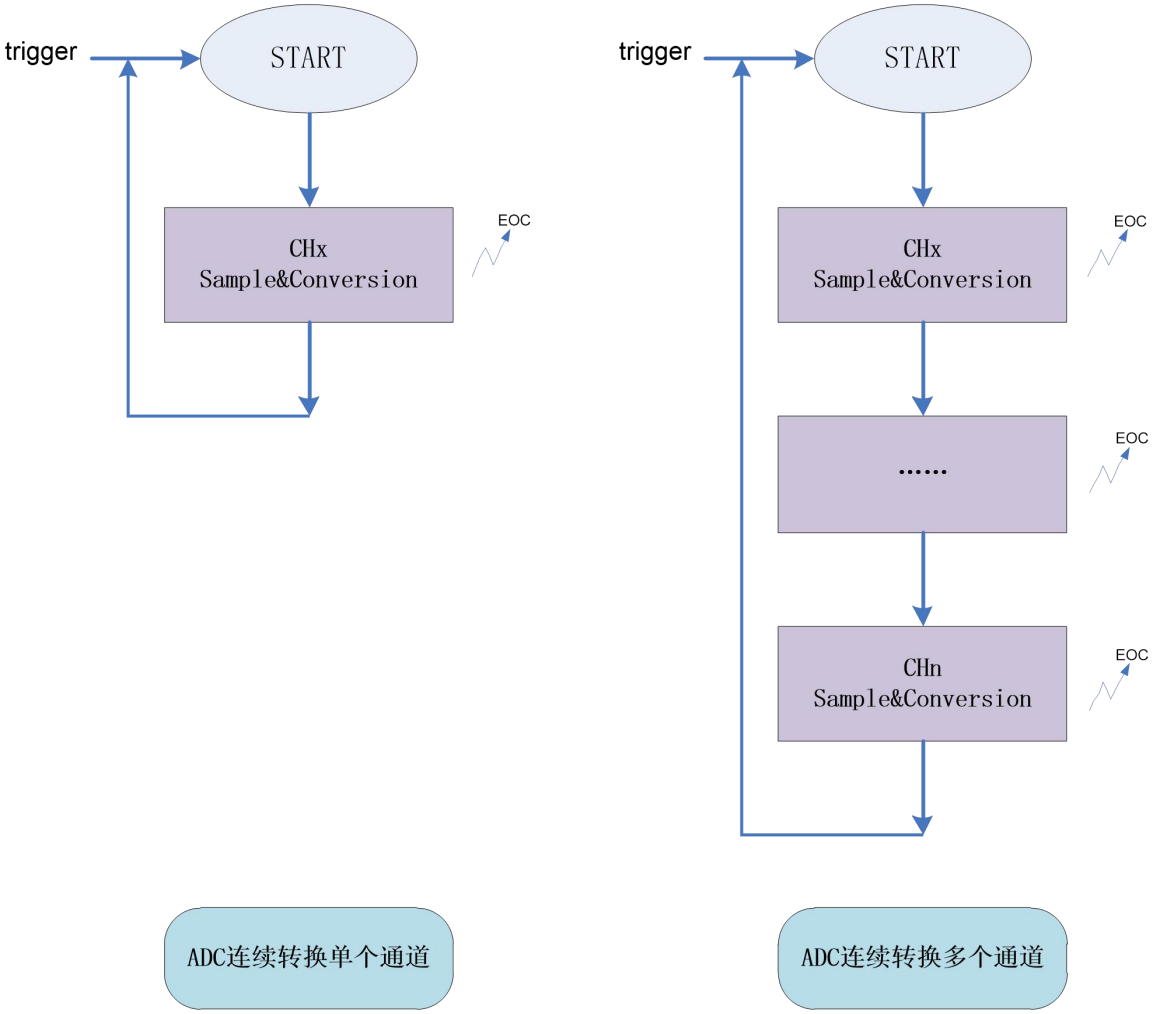


图 29-10SCAN_ACQ 连续转换模式

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。

29.7.4 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	BT0_TRGO	
0x17	LPT0_TRGO	
0x16	COMP4_TRGO	
0x15	COMP3_TRGO	
0x14	COMP2_TRGO	
0x13	COMP1_TRGO	

0x12~0xB	-	无触发
0xA	GT1_TRGO	
0x9	GT0_TRGO	
0x8	-	无触发
0x7	AT0_TRGO2	
0x6	-	无触发
0x5	AT0_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

图 29-11 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。

29.7.5 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，可以使用以下解决方案。

使用快速内部基准作为参考信号

芯片内部有一个快速建立的内部基准源，其输出电压是 1.0V，并且每颗芯片出厂时都经过校准，常温下精度为 1.0V+/-0.5%。

- ADC 采样并转换快速基准输出，并通过以下公式换算 VDDA 电压（假设 ADC 采样快速基准源的 12 位转换结果是 AVREF_DATA）：

$$VDDA = \frac{4095}{AVREF_DATA} \times 1.0V$$

- 假设 ADC 对某个输入通道的采样值为 ADC_DATA，通过以下公式可以得到当前某个输入通道的实际电压（12bit 输出）

$$V_{CHANNEL} = \frac{ADC_DATA}{AVREF_DATA} \times 1.0V$$

29.7.6 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：

1) 由以下公式计算当前温度传感器输出的绝对电压值

$$\text{VPTAT} = \frac{VREFIN_CAL \times TS_DATA}{VREFINT_DATA \times 4095} \times 3V$$

2) 由以下公式计算温度标定时（30°C）温度传感器输出的绝对电压值

$$\text{VPTAT_30C} = \frac{TS_CAL1}{4095} \times 3V$$

3) 根据温度传感器输出斜率计算当前绝对温度

$$\text{Temperature} = \frac{VPTAT - VPTAT_30C}{3.06mV/C} + 30C$$

其中，TS_DATA 是 ADC 采样当前温度传感器输出的转换值；由于不知道当前 VDDA 的准确电平，因此这个转换值需要根据 VREFINT 的转换结果进行比例缩放；TS_CAL1 是芯片生产时在 30C+/-1C、VDDA=3.0V 的条件下进行温度定标的转换结果，这个数据保存在 flash 中。

如果 ADC 当前工作参考电压是 VREFP，则当前温度可以通过下式计算：

$$Temperature = \frac{TS_DATA \times \frac{VREFP}{3.0V} - TS_CAL30}{slope} + 30C$$

29.7.7 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 SMTS1 和 SMTS2 寄存器可以选择采样时间。注意，当 ADCCLK=32MHz 时，采样时间不应小于 2*T_{ADCCLK}

SMTSx	Sampling cycles (T_{ADCCLK})
0000	1
0001	2
0010	4
0011	8
0100	12
0101	16
0110	32
0111	64
1000	80
1001	96
1010	128
1011	160
1100	192
1101	256
1110	320
1111	512

表 29-2 ADC 采样时间

实际 ADC 的采样转换时间：

T_{CONV} = (Sampling Cycles + 14) * T_{ADCCLK}

ADC 采样时间主要由采样电容、被采样信号的输出阻抗、芯片内部输入通道阻抗和所需达到的采样精度共同决定。

下图是单端输入通道的电路结构示意：

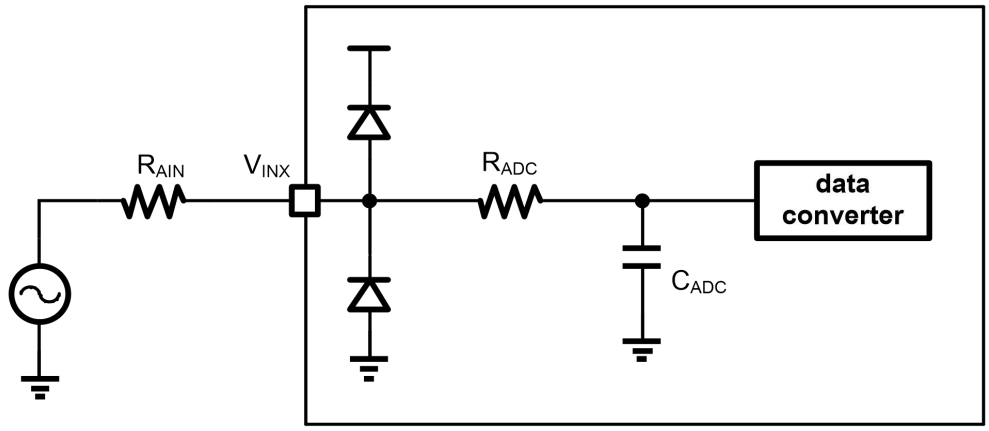


图 29-12 ADC 输入通道示意图

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。

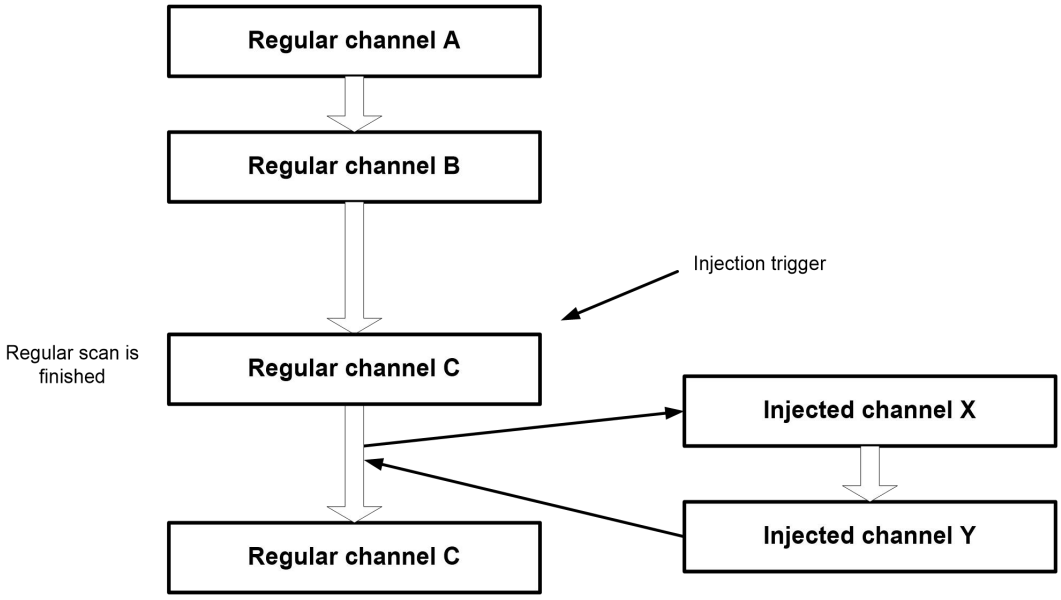
29.7.8 采样通道分组

ADC 支持常规组（regular group）和注入组（injection group）两类采样通道，其中注入组具有更高的优先级，能够在常规组扫描序列中插入注入组转换。注入组最大支持 4 个高优先级通道，通过 JSEL0、JSEL1、JSEL2、JSEL3 寄存器可选择任意 4 个通道为注入组，注入组通道转换结果仍保存在各个通道各自的结果寄存器中。

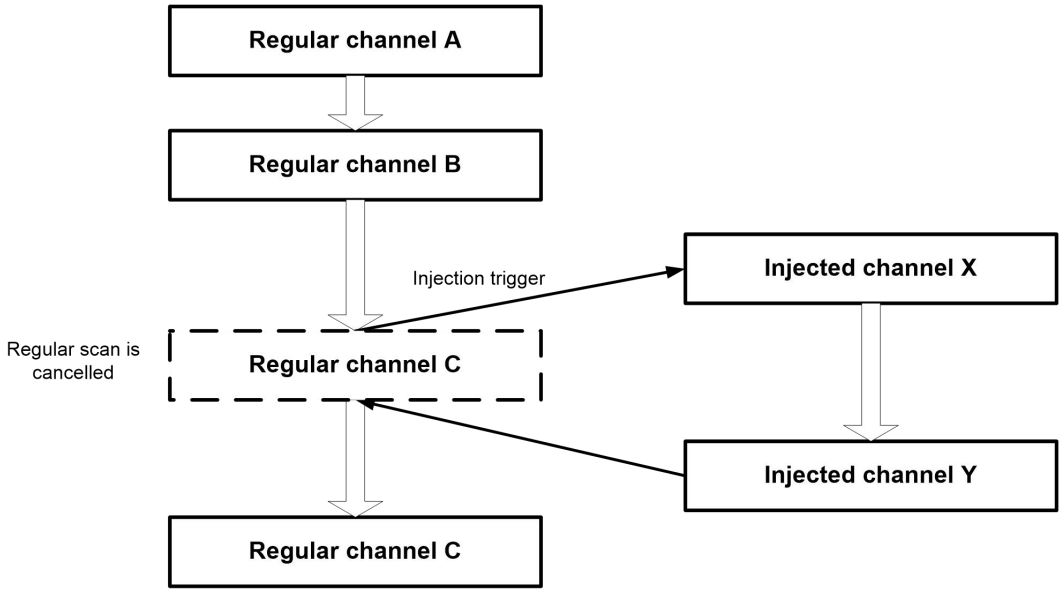
注入组支持半自动和自动模式。当 JSEMI=1，注入组采用半自动触发模式，每个触发信号将启动一次注入组转换；当 JSEMI=0，注入组采用全自动触发模式，每个触发信号将启动 JL 寄存器定义的注入组转换次数（最大 4 次）

注入组优先级高于常规组，因此当注入发生时，常规组扫描序列被打断。通过 JMOD 寄存器，可以配置注入中断模式：

- JMOD=0，立即注入——注入事件发生时，当前常规组采样转换被立即取消



- JMOD=1，延迟注入——注入事件发生时，等待当前常规组采样转换完成后再启动注入转换



29.7.9 转换结果

ADC 每个外部通道都有独立的转换结果寄存器，每个结果寄存器位宽为 12bit，经过过采样平均后的结果仍为 12bit。使能 DMA 的情况下，数据被搬运到 RAM 的同时，仍会写入对应的结果寄存器中。

ADC 每个内部通道共用一个转换结果寄存器，寄存器位宽 12bit，经过过采样平均后的结果仍为 12bit。使能 DMA 的情况下，数据被搬运到 RAM 的同时，仍会写入对应的结果寄存器中。为了便于软件区分内部通道数据寄存器中的值属于哪个内部信号，寄存器中同时包含通道编码。

29.7.10 转换模式

ADC 支持以下转换模式：

- 单次转换
  - 半自动触发（SEMI-AUTOMATIC）
  - 全自动触发（AUTOMATIC）
- 连续转换

转换启动可以由软件或事件触发，通过寄存器选择多个事件触发源。

29.7.10.1 单次转换模式

支持半自动触发和全自动触发两种模式。

全自动触发模式：软件或硬件触发事件启动 ADC 转换后，ADC 会顺序采样所有被使能的通道，单个通道采样完成后，EOC（End of Conversion）标志置位，所有通道采样完成后，EOS（End of Sequence）标志置位，本次转换结束。假设通道 0、3、5 被使能

- 1st 触发事件：通道 0、3、5 被顺序采样，过程中产生三次 EOC，最终产生 EOS
- 2nd 触发事件：重复上述过程

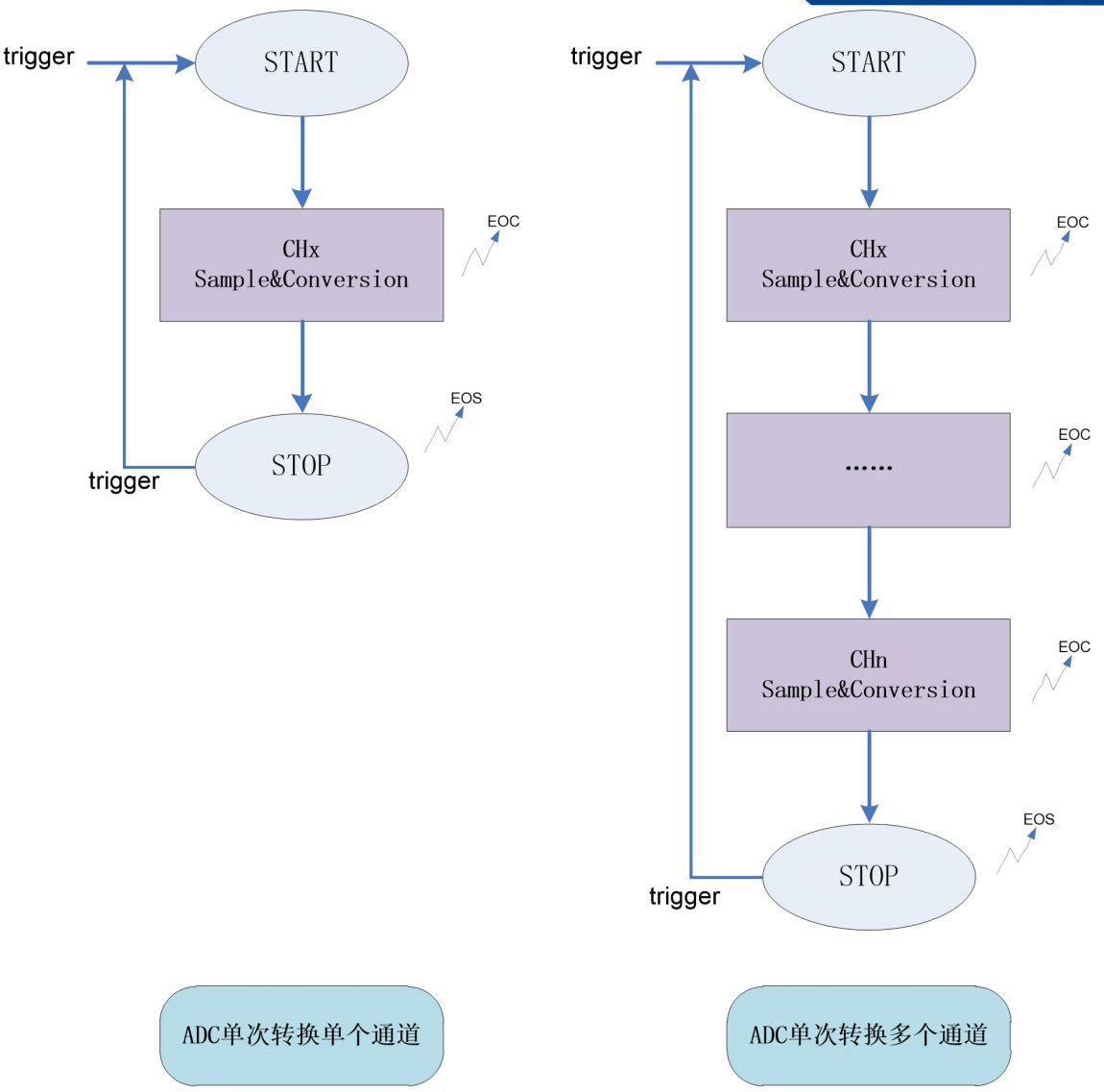


图 29-13 ADC 单次转换全自动触发模式

半自动触发模式：软件或硬件触发事件只会让 ADC 启动一次，转换一个使能通道。比如通道 0、3、5 被使能

- 1st 触发事件：通道 0 被采样，产生 EOC
- 2nd 触发事件：通道 3 被采样，产生 EOC
- 3rd 触发事件：通道 5 被采样，产生 EOC 和 EOS
- 4th 触发事件：通道 0 被采样，产生 EOC
- 5th 触发事件：通道 3 被采样，产生 EOC

……

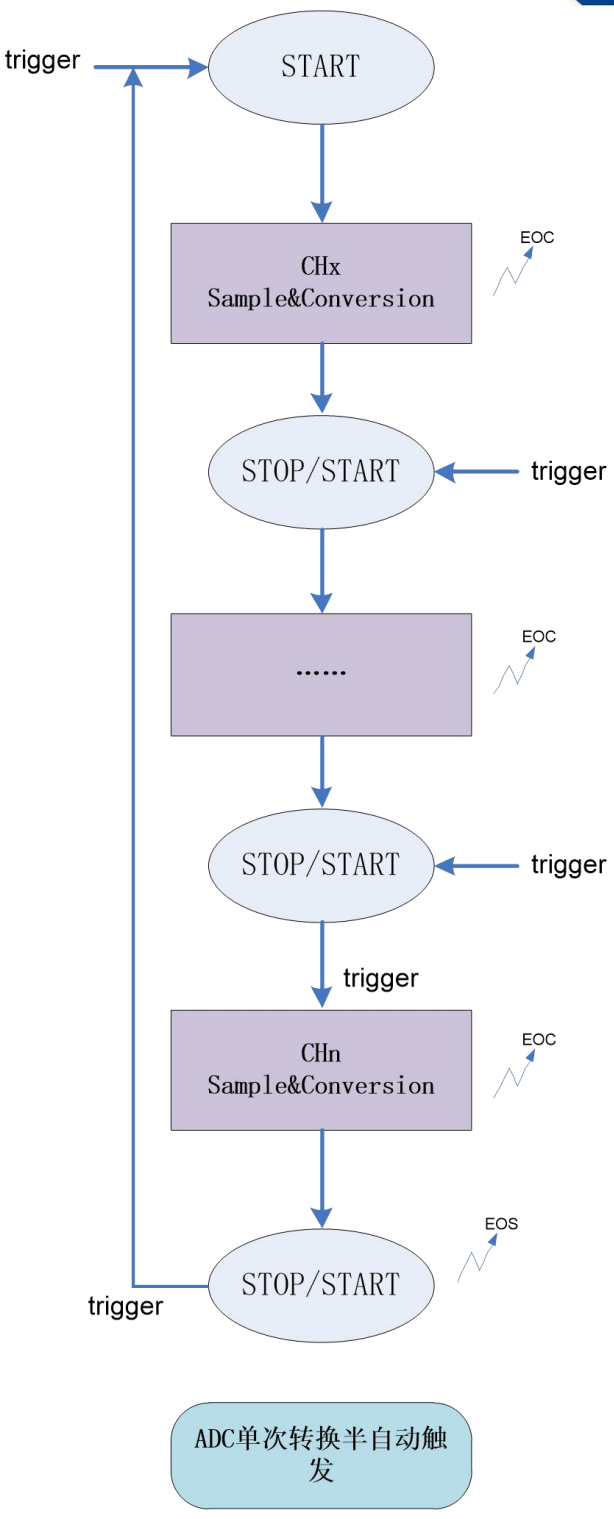


图 29-14 ADC 单次转换半自动触发模式

29.7.10.2 连续转换模式

触发事件到来后，所有使能通道被采样，并且 ADC 不会自动停止，而是循环采样，直到软件停止 ADC。

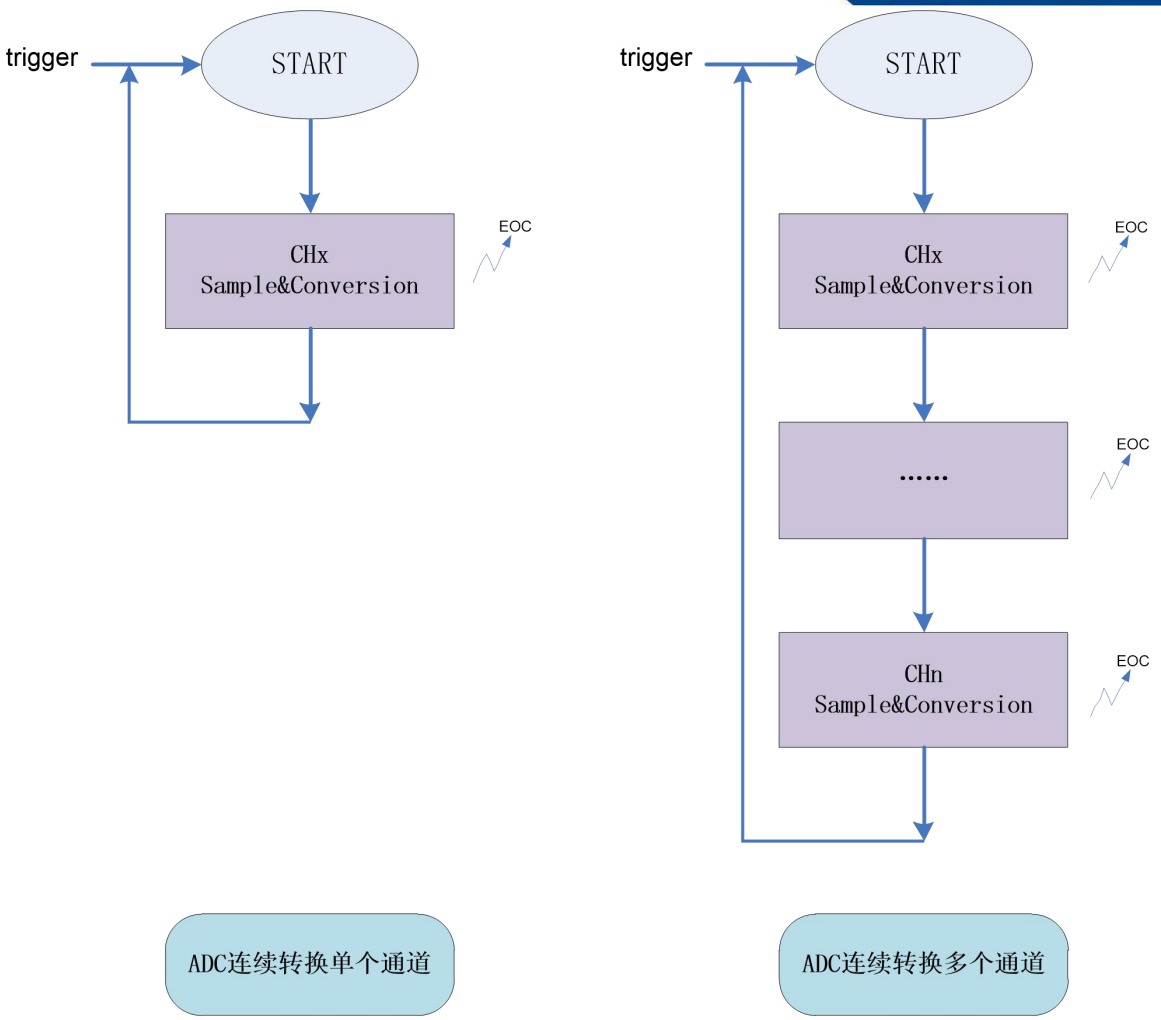


图 29-15 ADC 连续转换模式

每个通道被采样后，数据保存在 ADC_DATA 寄存器中，软件要在下次转换前及时读走数据，或者通过 DMA 进行数据搬移。如果不能及时取走数据，将引起 Overrun，置位 overrun 标志，并可以发出中断。

29.7.10.3 注入通道转换

注入组可以执行 1~4 个通道的采样转换，由 JL 寄存器定义。每个注入通道转换完成后，JEOC 标志置位，当 JL 定义的转换通道全部完成后，JEOS 标志置位。注入通道总是按 0->3 的顺序执行：

- JL=1，只执行注入组 0
- JL=2，执行注入组 0、1
- JL=3，执行注入组 0、1、2
- JL=4，执行注入组 0、1、2、3
- JL=0 或者非 1~4，不执行任何注入组

下图示例显示了 JL=2，JSEMI=0，注入触发后执行 2 次注入转换：

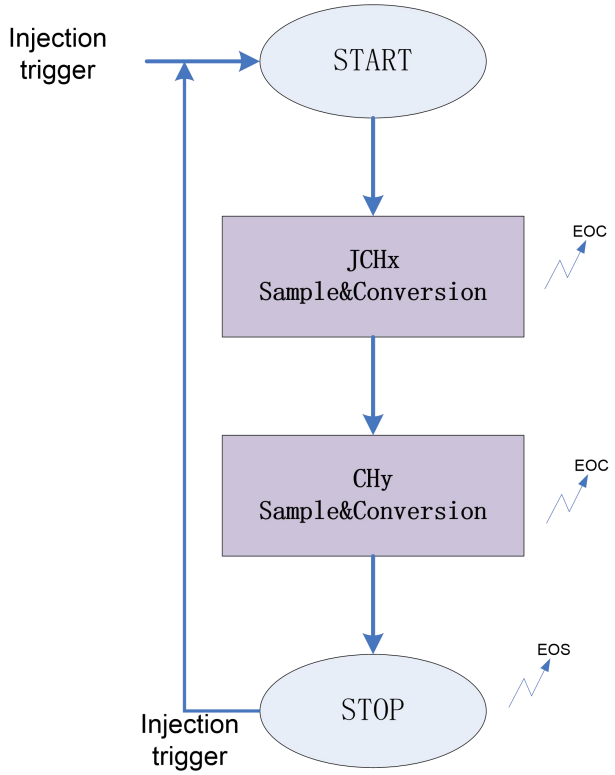


图 29-16 单次注入触发执行 2 次注入转换

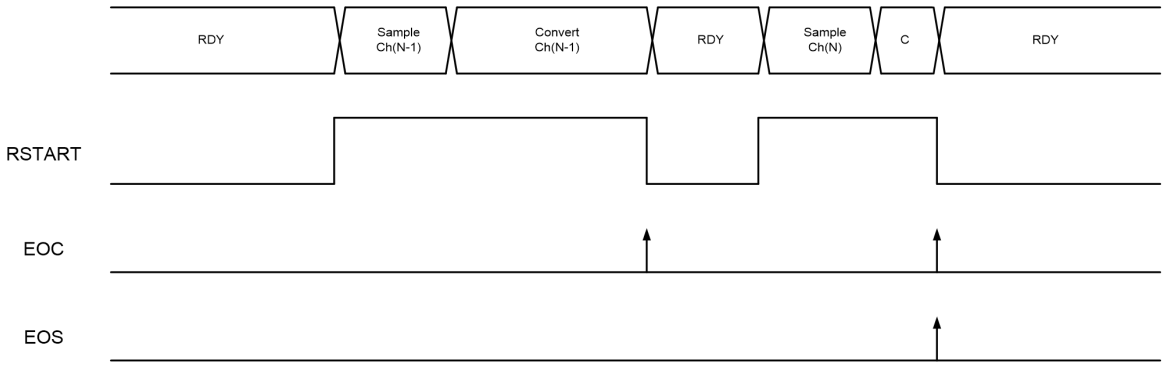
29.7.11 转换触发

ADC 使能后，转换触发支持软件或硬件事件触发。常规组通过 GTRGCFG 寄存器配置触发方式，注入组通过 JTRGCFG 寄存器配置触发方式。

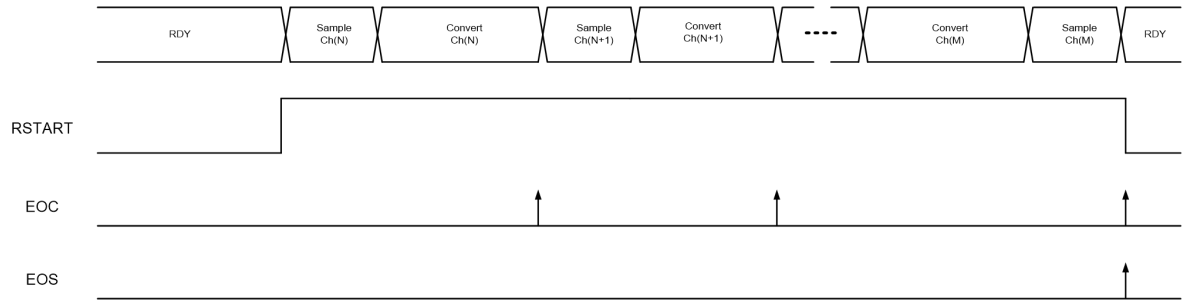
软件触发（xTRGCFG=00）

软件通过直接置位 RSTART 寄存器启动转换。

下图是软件半自动触发的单次转换，软件每次置位 RSTART 启动一次转换：



下图是软件全自动触发的单次转换，软件置位 RSTART 后完成全部使能通道转换：

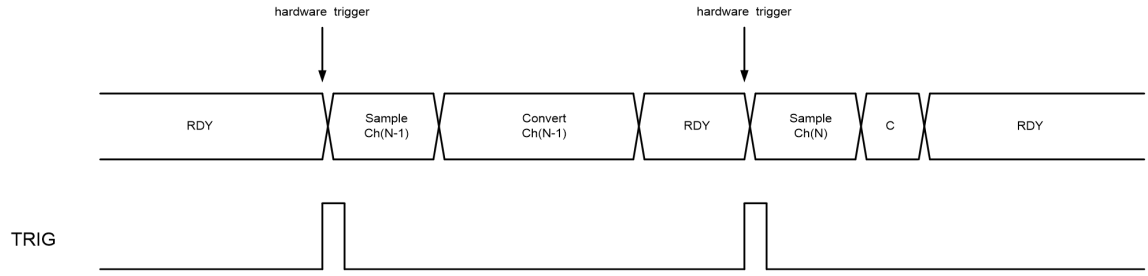


硬件触发（xTRGCFG=01/10/11）

此时 ADC 转换由硬件触发源启动，可以配置在系统触发信号的上升沿、下降沿触发、或者同时在上升沿下降沿触发转换。

如果 ADC 正处于转换过程中，此时到来的触发信号会被忽略。

下图为硬件半自动触发：



硬件触发源示意图：

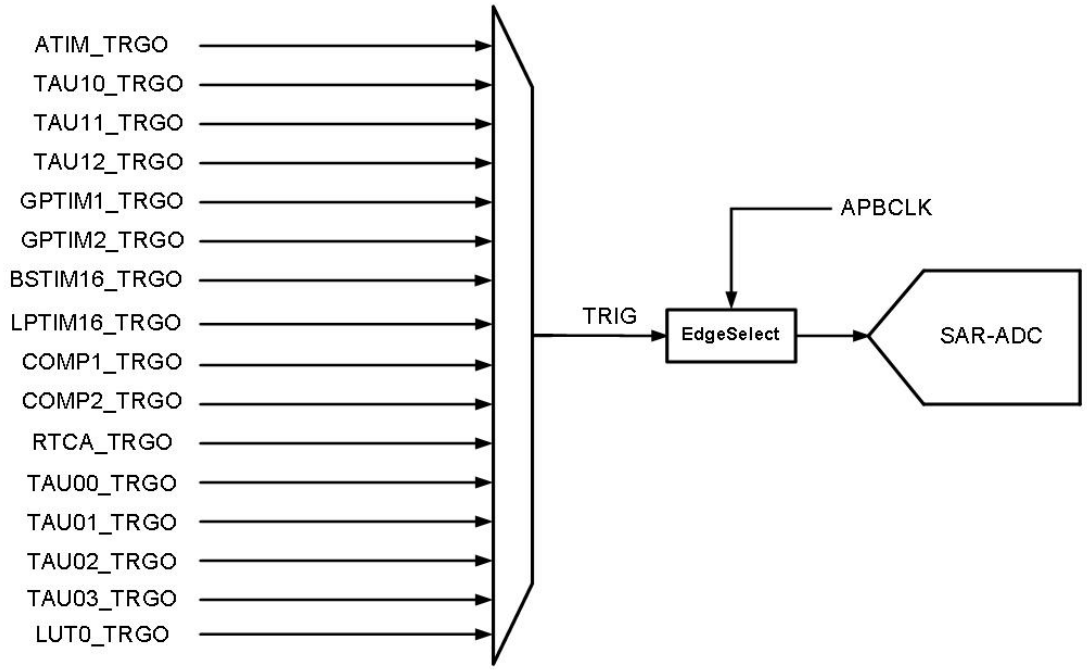


图 29-17 ADC 硬件触发源

注意：硬件触发源输入后经过 APBCLK 同步采样，因此使用硬件触发功能时必须使能 ADC 总线时钟（置位 CMU 模块中的 ADC_PCE 寄存器）。

ADC 注入组硬件触发源如下表。

JEXTS	触发信号	说明
0000	ATIM_TRGO	
0001	ATIM_TRGO2	
0010	GPTIM1_TRGO	
0011	GPTIM0_TRGO	
0100	BSTIM16_TRGO	
0101	LPTIM16_TRGO	
0110	COMP1_TRGO	
0111	COMP2_TRGO	
1000	COMP3_TRGO	
1001	COMP4_TRGO	
1010	ATIM_OC4	
1011		
1100		
1101	TAU11_TRGO	
1110	TAU12_TRGO	
1111	TAU13_TRGO	

图 29-18 ADC 注入组硬件触发源

触发源时序特征分类。

触发源	时钟域	当 ADC 工作时钟选择为 APBCLK 时的触发延迟	说明

		(T_{APBCLK})	
软件触发	APBCLK	3.5	可实现同步触发，确定延迟
LUTx_TRGO		4.5	
GPTIMx_TRGO		4.5	
COMPx_TRGO		4.5	
ATIM_TRGO	其他	2~3	异步时钟触发
BSTIM16_TRGO			
LPTIM16_TRGO			
RTC_TRGO			

29.7.12 转换停止

软件通过置位 RSTOP 和 JSTOP 寄存器，可以停止当前正在进行的常规组和注入组转换。常规组停止和注入组停止互不影响：

- 在注入组转换时置位 JSTOP，ADC 立即停止注入组转换并返回常规组转换序列
- 在注入组转换时置位 RSTOP，ADC 注入组转换仍继续执行，注入组转换完成后，由于常规组已经停止，不会再回到常规组转换序列
- RSTOP 会停止当前常规组序列，此后如果新的触发到来，则常规组转换序列将从头开始

下图示例显示了在常规组半自动硬件触发模式下，软件置位 RSTOP：

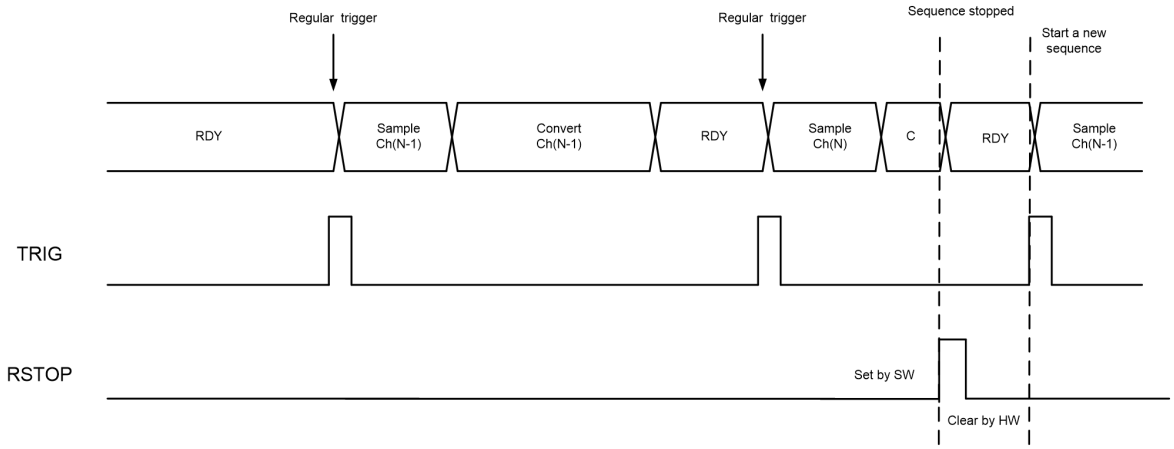


图 29-19 常规组转换 STOP

下图示例显示了常规组和注入组分别 STOP 的情况：常规组由软件触发后连续转换，其中插入软件触发的注入转换，注入转换被 JSTOP 终止，状态机回到常规组，随后常规组被 RSTOP 终止。

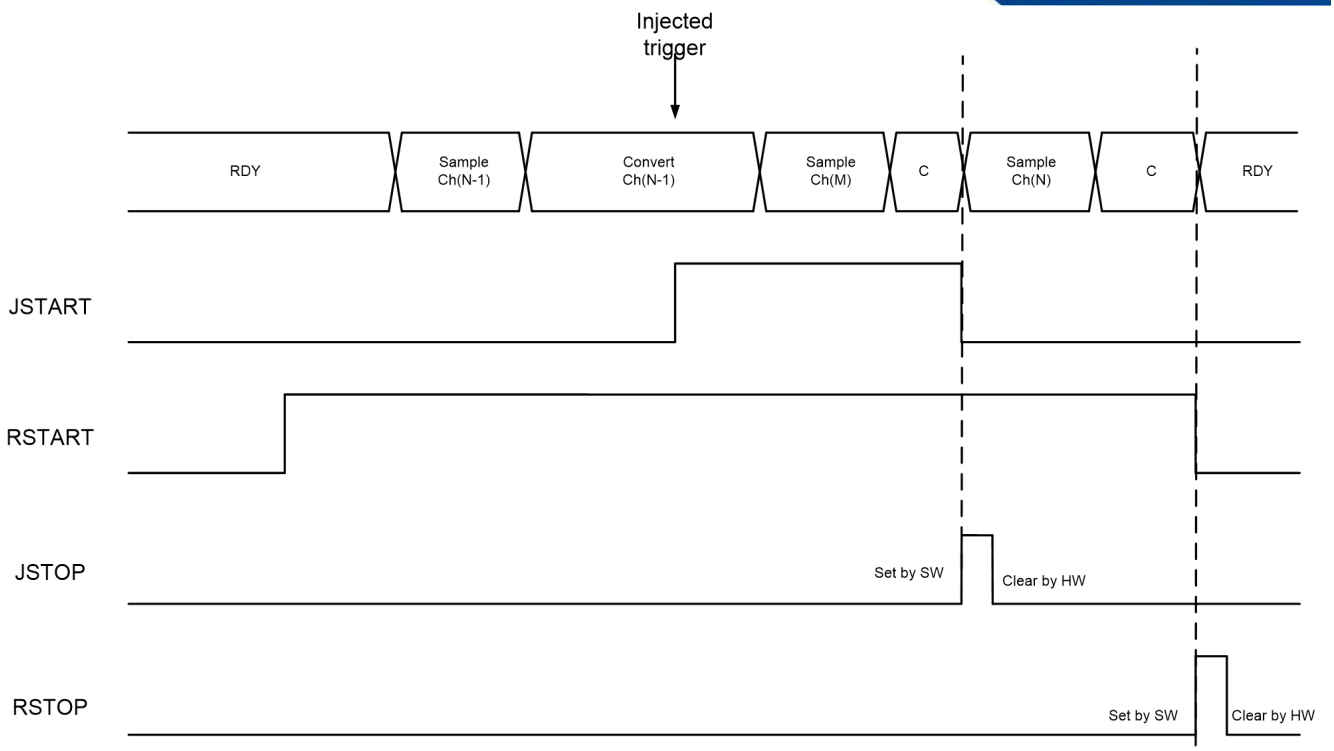


图 29-20 常规组和注入组转换 STOP

29.7.13 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。

29.7.14 ADC 工作时钟

ADC 工作时钟为 APB1CLK 或者其分频。

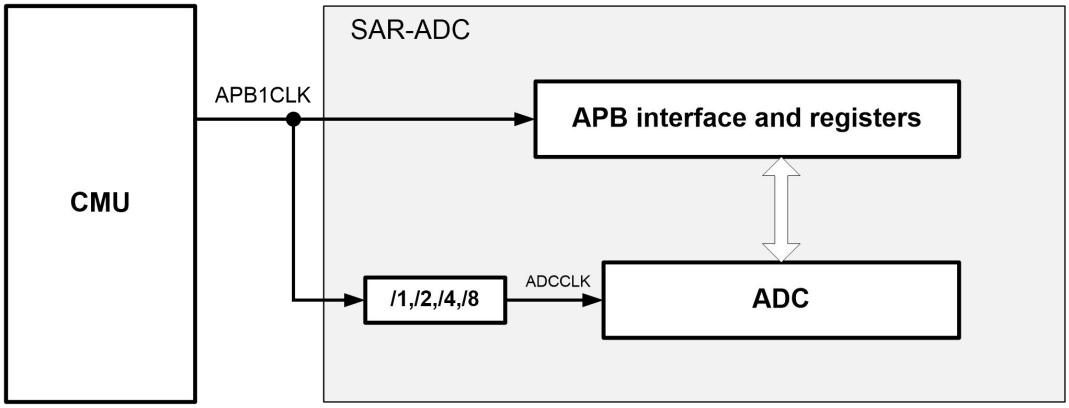


图 29-21 ADC 时钟示意图

29.7.15 数据冲突和自动等待

每次转换完成后 EOC 标志会置位，软件或 DMA 读取 ADC_DATA 寄存器后会自动清除 EOC，也可以由软件写 1 清除。当 EOC 标志没有被清除的情况下，新的转换数据到来，就会导致 data overrun；有两种 overrun 模式：

OVRMOD=0：保持旧的数据，新数据丢弃

OVRMOD=1：写数据写入覆盖旧数据

当使用 DMA 时出现 overrun，则不会发起新的 DMA request，直到 OVR 标志被软件清零

ADC 控制器还支持自动等待，如果 WAIT 寄存器被软件置位，那么在 ADC_DATA 寄存器被读取之前，ADC 控制器不会发起新的转换；在等待状态中到来的硬件触发事件也会被忽略。

下图是软件触发连续模式的情况下，使能了自动等待的示意图：

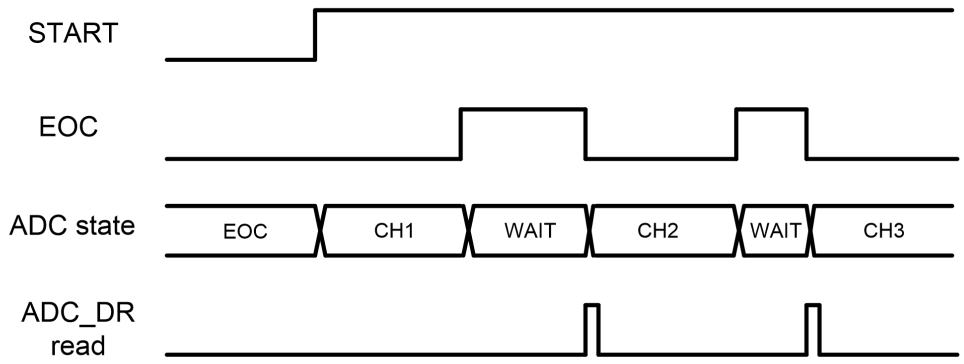


图 29-22 ADC 自动等待

注意：DMA 循环模式下请保持 WAIT=0

29.7.16 DMA

在多通道转换或连续转换时，使用 DMA 进行转换结果搬移是高效的解决方案。在使能了 DMAEN 的情况下，当每次转换完成后（EOC），ADC controller 模块会产生一个 DMA 请求，通知 DMA 将数据寄存器中的结果搬运到指定的 SRAM 地址。ADC 转换由触发事件启动，触发可以来自于软件触发或硬件触发。DMA 模式下支持一次触发后连续执行多次转换（自动模式），或者每次触发执行一次转换的方式（半自动模式）。

DMA 搬运的除了数据还有通道 ID，便于软件识别数据所属通道。由于通道 ID 占据 6bit，数据占据 12bit，因此 DMA 搬运必须以 word 为单位。数据组织如下：

00	ID (6bits)	0000 0000 0000	Data (12bits)

在自动模式下（ADC_CFGR2.SEMI=0）和半自动模式下（ADC_CFGR2.SEMI=1），ADC 的 DMA 接口都可以支持单次模式和循环模式：

单次模式

转换完成后发起数据搬运，此过程会一直重复，直到软件配置的 DMA 传输长度完成，然后 ADC 控制器会自动停止转换（通过接收 DMA 的传输完成中断标志信号），关闭 ADC，不再向 DMA 发起请求。此模式主要用于对特定模拟信号进行一定长度的采样。

循环模式

与 DMA 的循环模式相配合，ADC 不断循环转换并发起 DMA 请求，直到软件停止转换。此模式可以用于处理连续不断模拟信号采样。

通过 ADC_CFGR2.SEMI 寄存器可以配置 DMA 模式下的自动或半自动转换，通过 DMA 模块的 CHxCIRC 配置循环模式或单次模式，通过 DMA 模块的 CHxTSIZE 配置传输长度。寄存器配置组合实现的效果如下表所示。

ADC_CFGR2.CONT	ADC_CFGR2.SEMI	DMA CHxCIRC	DMA CHxTSIZE	说明
0	0	0	N	单次全自动模式，触发事件到来后，依次对使能输入通道转换 N 次并将数据搬运到 SRAM；所有使能输入通道都被采样过，ADC 自动停止并等待下一次触发；如果数据搬运次数达到 N，DMA 自动关闭。
0	0	1	N	单次全自动模式（循环存储），触发事件到来后，

				依次对使能输入通道不断转换并搬运数据，直到所有使能输入通道都采样过，ADC 自动停止并等待下一次触发；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到 DMA 通道指针指向的起始地址，并覆盖原数据。
0	1	0	N	单次半自动模式，每个触发事件启动一次转换，依次采样所有被使能的输入通道；数据搬运次数达到 N，DMA 自动关闭。
0	1	1	N	单次半自动模式（循环存储），每个触发事件启动一次转换，依次采样所有被使能的输入通道；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到 DMA 通道指针指向的起始地址，并覆盖原数据。
1	x	0	N	连续模式，触发后 ADC 不断采样使能通道，DMA 不断搬运数据，直到完成 N 次数据搬运，DMA 自动关闭。
1	x	1	N	连续模式（循环存储），触发后 ADC 不断采样使能通道，DMA 不断搬运数据；RAM 数据空间长度为 N，当搬运数据长度超过 N 之后回到起始地址覆盖原数据；直到软件关闭 DMA。

表 29-3 DMA 配置与功能

在 DMA 使能情况下，如果发生 overrun，则 ADC 控制器不再发送 DMA 请求，直到 OVR 标志被清除。

注意，在单次和连续转换模式下，都可以支持 DMA 传输；DMA 传输长度以 EOC 的次数定义，而不是 EOS，即 DMA 只关心搬运多少次 ADC_DATA。

DMA 循环模式下应保持 WAIT=0

29.7.16.1 DMA 使用案例

使用案例 1：

ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据，然后 ADC 和 DMA 暂停工作，等待第二次触发。第二次

触发后，ADC 依次采样 3 个通道，DMA 搬运 3 次数据。6 次 DMA 传输完成，DMA 通道关闭。

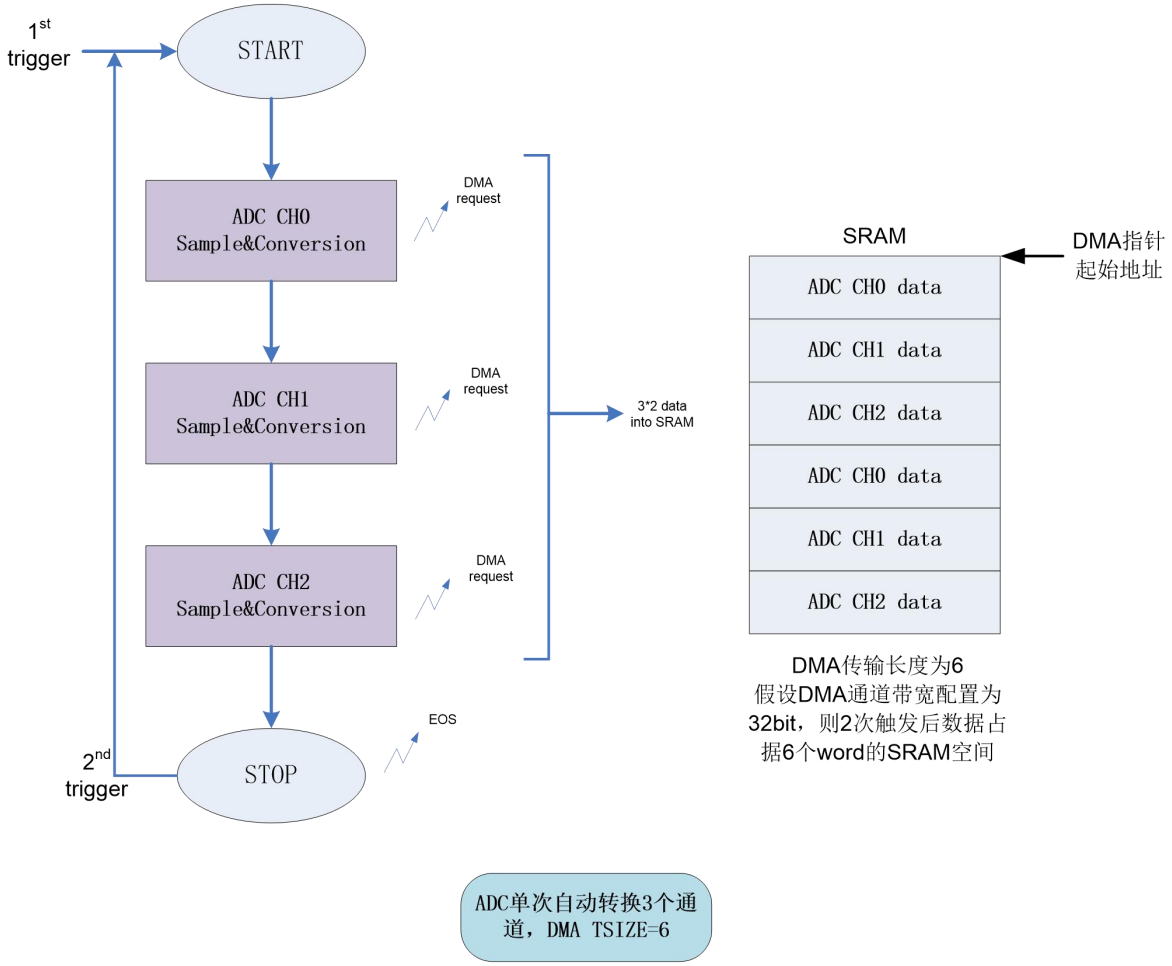


图 29-23 ADC 单次全自动触发+DMA 案例 1

使用案例 2：

ADC 配置为单次全自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 2。第一次触发后，ADC 依次采样 3 个通道，DMA 搬运 2 次数据后 DMA 和 ADC 自动关闭，不会对第三个输入通道进行采样。

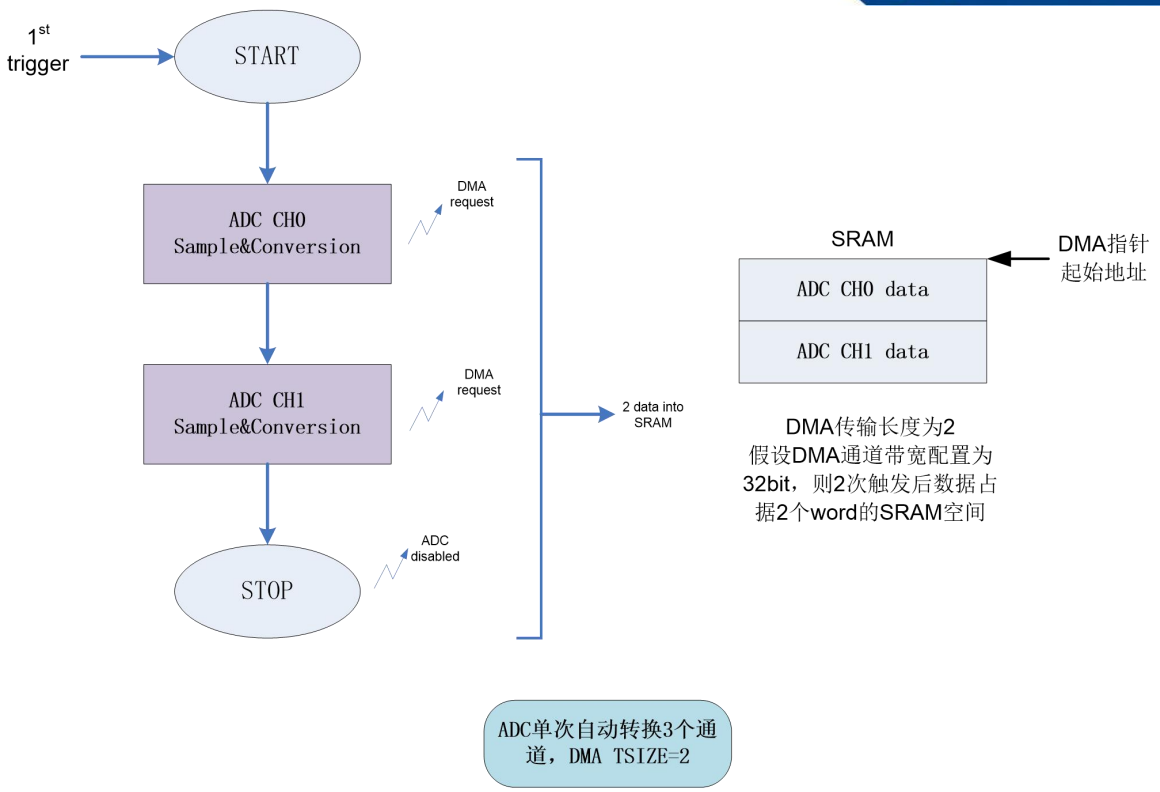


图 29-24 ADC 单次全自动触发+DMA 案例 2

使用案例 3：

ADC 配置为单次半自动模式，使能 3 个 ADC 输入通道，DMA TSIZE 配置为 6。每次触发后，DMA 搬运一个转换数据，总共 6 个数据搬运完毕后，自动关闭 ADC 和 DMA。

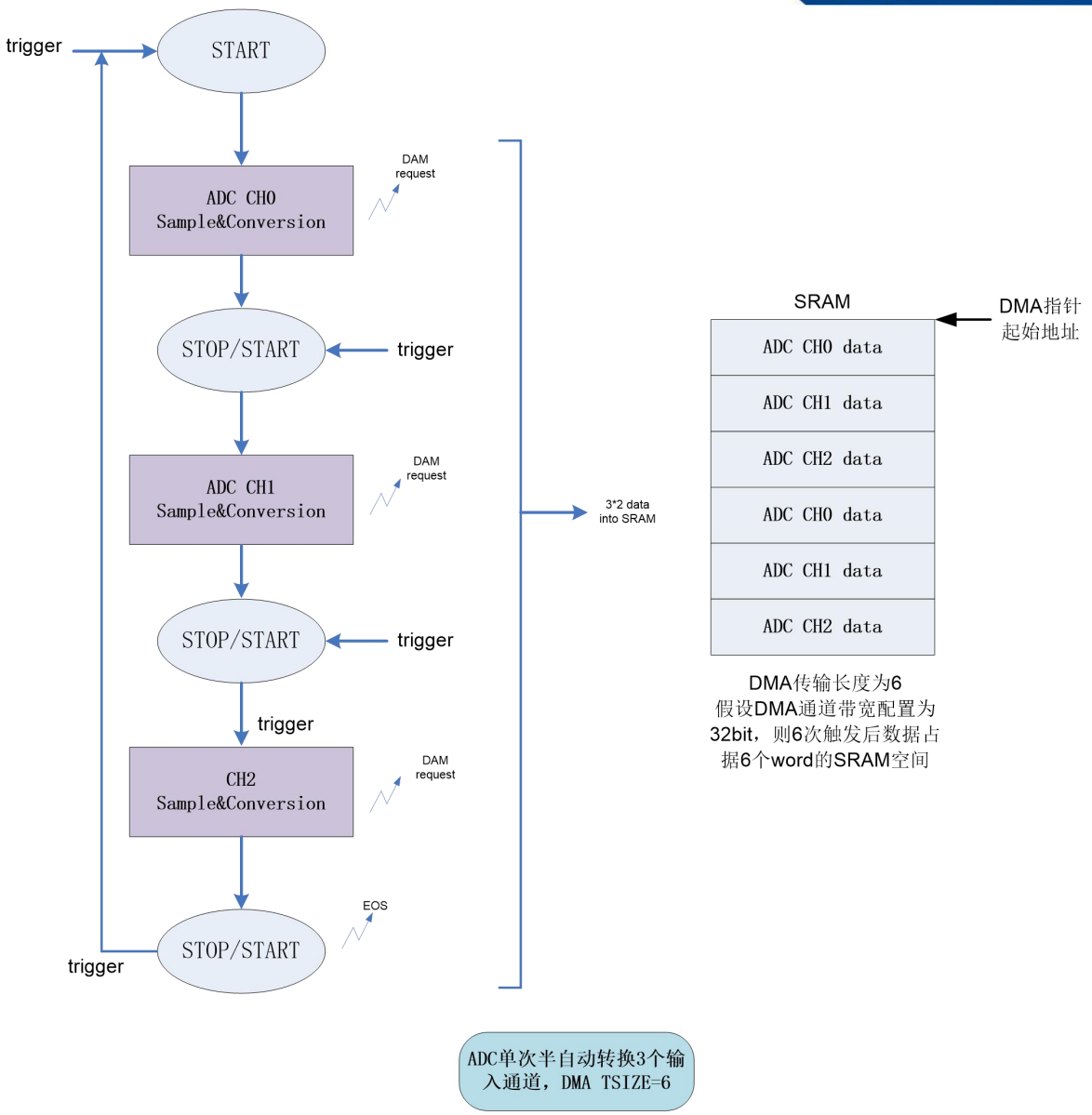


图 29-25 ADC 单次半自动触发+DMA

使用案例 4：

ADC 配置为全自动模式（SEMI=0），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 执行 3 次采样转换，DMA 搬运 3 次数据到 RAM 的 x/x+1/x+2 地址；第二次触发后，ADC 执行 3 次采样，DMA 搬运 3 次数据到 RAM 的 x+3/x+4/x+5 地址；第三次触发后，ADC 执行 3 次采样，DMA 搬运 3 个数据到 RAM 的 x/x+1/x+2 地址，循环往复直到软件关闭 DMA。

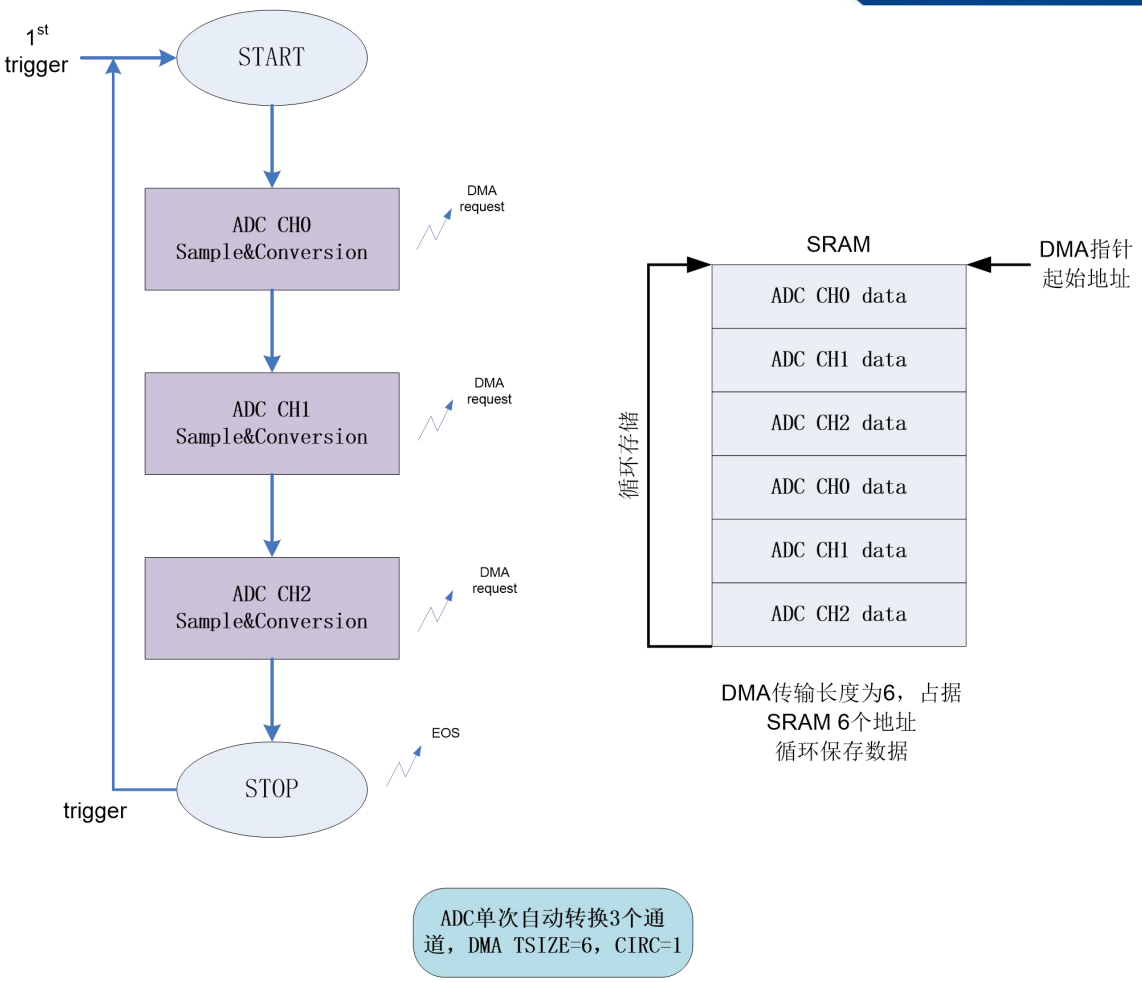


图 29-26 ADC 全自动触发+DMA 循环模式

使用案例 5：

ADC 配置为连续模式（CONT=1），使能 3 个 ADC 输入通道（ADC_IN0/1/2），DMA TSIZE 配置为 6，DMA 指针指向 RAM 地址 x，DMA 配置为循环模式（CIRC=1）。第一次触发后，ADC 对使能通道连续不间断采样转换，DMA 连续搬运数据到 RAM 的 x/x+1/x+2/x+3/x+4/x+5 地址；循环往复直到软件关闭 DMA。

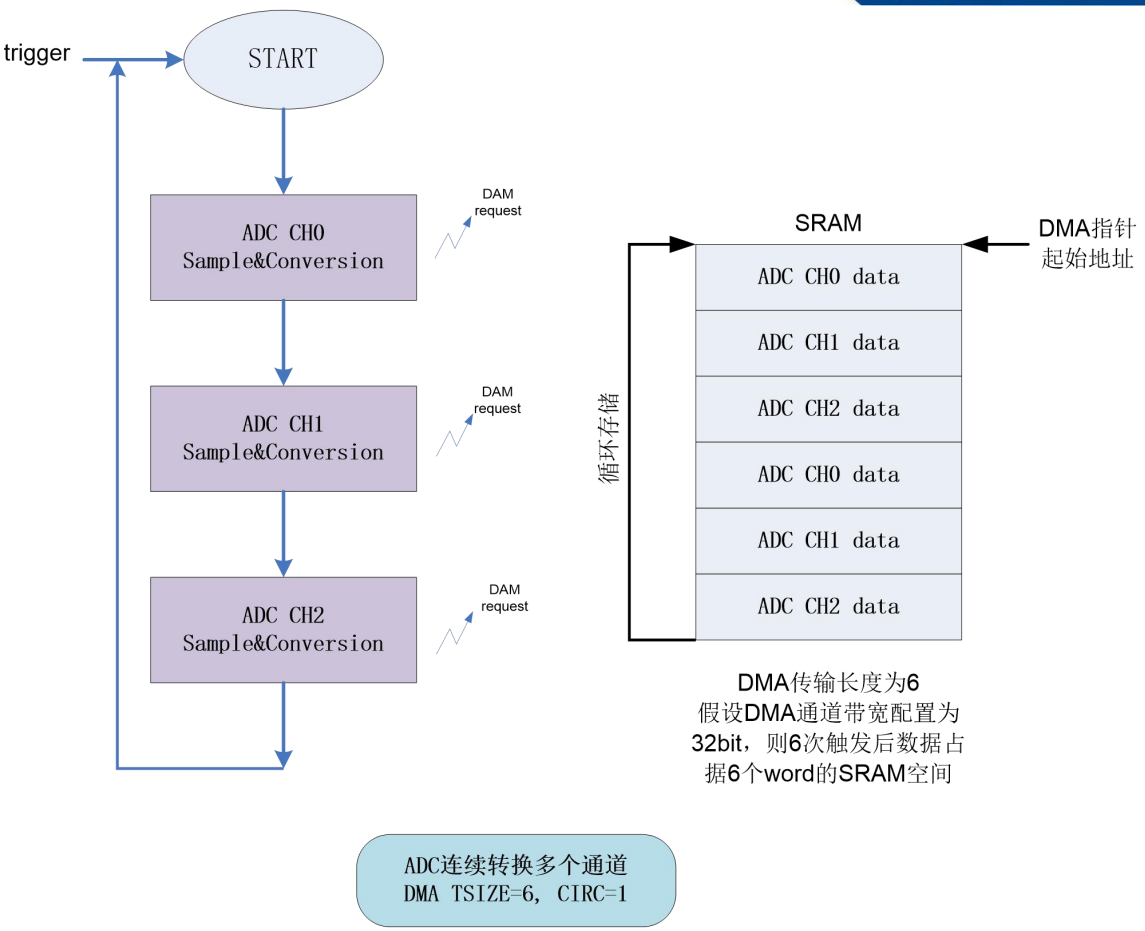


图 29-27 ADC 连续模式+DMA 循环模式

29.7.17 模拟窗口看门狗（AWD）

AWD 功能用于监视某个模拟输入通道或所有输入通道的输入信号电平是否处于寄存器设置的幅值范围之内。当 ADC 转换值高于 AWD_HT 或者低于 AWD_LT 时，都会置位中断标志寄存器。标志寄存器由软件写 1 清零。

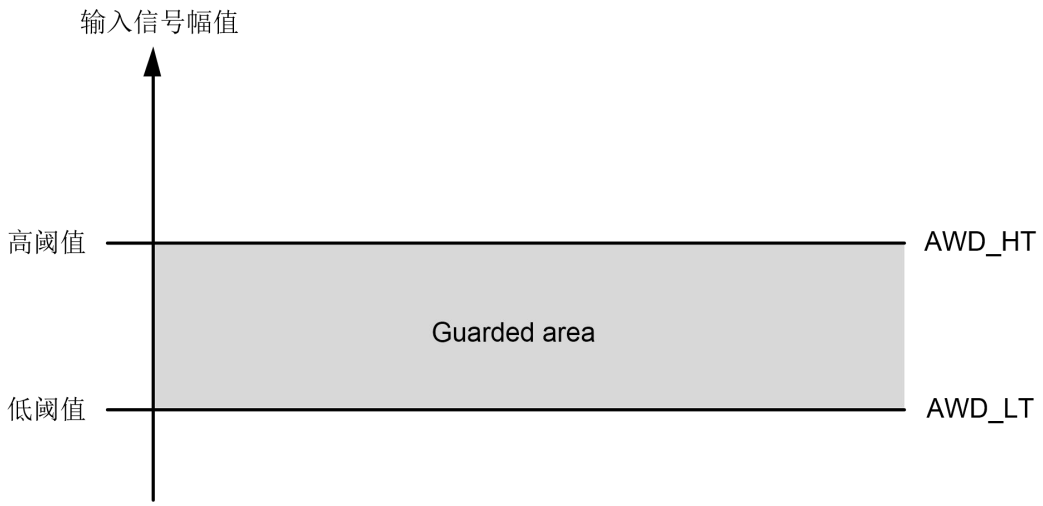


图 29-28 ADC 模拟看门狗阈值示意图

通过 AWDEN 寄存器使能模拟窗口看门狗功能，通过 AWDSC 寄存器配置单通道监视或全部通道监视。

AWD 阈值设置寄存器的位宽是 16bit。

29.7.18 ADC 校准

ADC 支持 offset 自校准，建议在芯片上电后，首先进行一次校准操作，以获得更好的精度。

ADC offset 校准采用以下方法：

使 ADC_OSR.CALI_EN 寄存器写 1 使能较准模式，此时从每个 ACQ_DR 寄存器读出的是 ADC core 输出经过转换后得到的 13bit（已经考虑冗余权重但是未经过饱和处理的数据）；校准时软件启动对 VREFPX 和 VREFNX 的转换，得到 2 个 13bit 结果 D0 和 D1，软件需要根据 D0 和 D1 计算得到 offset 值（13bit），并将 Doffset 写入 offset 寄存器：

$$D_{offset13} = \frac{D0 + D1 - 4095}{2}$$

通过将 13bit 结果（14bit->13bit 转换参见“ADC 结果转换”）与 Doffset 计算后，得到最终的 13bit 结果：

$$D_{cal13} = D_{out13} - D_{offset13}$$

在不做校准的情况下，Doffset 默认值为 136

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
- 结果写入 ADC_OSR，同时 CALI_EN 位写入 0
- 校准完成

29.8 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。

29.9 寄存器

模块起始地址：0x4001_AC00

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	GISR
0x0000000C	通用中断使能寄存器	GIER
0x00000010	控制寄存器 1	CR1
0x00000014	控制寄存器 2	CR2
0x0000001C	配置寄存器 1	CFGR1
0x00000020	配置寄存器 2	CFGR2
0x0000002C	校准数据寄存器	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	HLTR
0x0000003C	AWD 阈值配置寄存器	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	通道完成标志寄存器	CHISR
0x00000084	DMA 数据寄存器	CHDMA_DR
0x00000088~0x00000100	采样通道 x 数据寄存器	CHx_DR
0x00000304	Debug 数据寄存器	DBG_DR
0x00000308	Debug 控制寄存器	DBG_CR

29.9.1 ACQ 中断和状态寄存器

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_ID					OVR_IF
位权限	U-00		R-00000					W1C-0
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
21:17	OVR_ID	数据溢出时对应的通道 ID
16	OVR_IF	数据溢出标志
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 … EOCIF[0] – ACQ0

29.9.2 ACQ 中断使能寄存器

名称	ACQ_IER							
地址	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							

名称	ACQ_IER							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OVRIE
位权限	U-0000000							RW-0
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
31:17	RFU	RFU：未实现，读为 0
16	OVRIE	数据溢出中断使能
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

29.9.3 通用中断和状态寄存器

名称	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	R-0	R-0	U-0	R-0	R-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，本寄存器仅在 ACQ_ID 为 SCAN_ACQ 的时候有效，如果当前工作的 ACQ 不是 SCAN_ACQ,则显示为上一次 SCAN_ACQ 扫描过的通道。
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

29.9.4 通用中断使能寄存器

名称	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							
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	DUMMY
位权限	U-0	RW-0	RW-0	U-0	U-0	RW-0	RW-0	RW-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

位	位名	功能描述
2	EOCNT_IE	EOCNT 标志中断使能
1	EOSIE	EOS 标志中断使能
0	DUMMY	DUMMY 寄存器

29.9.5 ADC 控制寄存器 1

名称	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
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

29.9.6 ADC 控制寄存器 2

名称	CR2							
地址	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
位权限	U-00000000							

名称	CR2							
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

29.9.7 ADC 配置寄存器 1

名称	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
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

位	位名	功能描述
		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

29.9.8 ADC 配置寄存器 2

名称	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
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-1	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

位	位名	功能描述
		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 10：使能过采样，但是过采样会保留所有数据，供软件查看计算,每一个数据都会产生 EOCIF 标志
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
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

29.9.9 校准数据寄存器

名称	OSR

名称	OSR							
地址	0x0000002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CALI_EN	RFU		OS_CALI_SLOW				
位权限	RW-0	U-00		R/W-00001				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OS_CALI_SLOW							
位权限	R/W-10001001							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			OS_CALI_FAST				
位权限	U-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OS_CALI_FAST							
位权限	RW-10001001							

位	位名	功能描述
31	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
30:29	RFU	RFU:未实现，读为 0
28:16	OS_CALI_SLOW	慢速通道 offset 校准值
15:13	RFU	RFU:未实现，读为 0
12:0	OS_CALI_FAST	快速通道 offset 校准值

29.9.10 ACQ 标志寄存器

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	RFU							
位权限	U-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			REQ				
位权限	U-000			R-00000				

位	位名	功能描述
31:5	RFU	RFU：未实现，读为 0
4:0	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

29.9.11 Scan ACQ 通道控制寄存器

名称	ACQ_SACR							
地址	0x00000034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHEN							
位权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHEN							
位权限	RW-00000000							
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-00000000							

位	位名	功能描述
31:0	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 有效 如 CHEN=0x0005，表示 scan_ACQ 会扫描 0 通道和 2 通道 其中，bit0~23 控制外部通道使能 bit24：VREFNX bit25：VREFPX bit26：温度传感器 bit27：AVREF bit28：DAC bit29：TSI LDO bit30：TSI CAP bit31：RFU

29.9.12 AWD 阈值配置寄存器

名称	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							
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

29.9.13 AWD 阈值配置寄存器

名称	AWDCR							
地址	0x0000003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU		AWD_CHEN					
位权限	U-00		RW-000000					
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-00000000							
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
31:30	RFU	RFU：未实现，读为 0
29:0	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

29.9.14 ACQx 控制寄存器

名称	ACQx_CR（x=0~3）							
地址	0x00000040+x*0x4							
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-0001			

位	位名	功能描述
31:14	RFU	RFU：未实现，读为 0

位	位名	功能描述
13:9	CHSEL	ACQ0~ACQ3 通道选择，参见《ADC 通道选择》章节
8:4	TRIGSEL	触发源选择,5bit 0：软件触发 1：atim_trgo 2：atim_trgo2 3：btim_trgo 4：lpt_trgo 5：rtc_trgo 6：comp1_dout 7：comp2_dout 8：tau0_trgo[0] 9：tau0_trgo[1] 10：tau0_trgo[2] 11：tau0_trgo[3] 12：tau0_trgo[4] 13：tau0_trgo[5] 14：tau0_trgo[6] 15：tau0_trgo[7] 16：tau1_trgo[0] 17：tau1_trgo[1] 18：tau1_trgo[2] 19：tau1_trgo[3] 20：tau1_trgo[4] 21：tau1_trgo[5] 22：tau1_trgo[6] 23：tau1_trgo[7] others RFU
3:0	SMTS	采样时间配置:ADC_CLK 周期数 0x0 ： 1 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

29.9.15 SCAN_ACQ 控制寄存器

名称	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				SMTS_FAST			
位权限	U-0000				R/W-0001			
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_SLOW			
位权限	RW-0000				RW-0001			

位	位名	功能描述
31:20	RFU	RFU：未实现，读为 0
19:16	SMTS_FAST	快速通道采样时间配置：ADC_CLK 周期数 0x0 ： 1 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
15:9	RFU	RFU：未实现，读为 0
8:4	TRIGSEL	触发源选择,5bit 0：软件触发 1：atim_trgo 2：atim_trgo2 3：btim_trgo 4：lpt_trgo 5：rtc_trgo 6：comp1_dout 7：comp2_dout 8：tau0_trgo[0] 9：tau0_trgo[1] 10：tau0_trgo[2] 11：tau0_trgo[3]

位	位名	功能描述
		12：tau0_trgo[4] 13：tau0_trgo[5] 14：tau0_trgo[6] 15：tau0_trgo[7] 16：tau1_trgo[0] 17：tau1_trgo[1] 18：tau1_trgo[2] 19：tau1_trgo[3] 20：tau1_trgo[4] 21：tau1_trgo[5] 22：tau1_trgo[6] 23：tau1_trgo[7] others RFU
3:0	SMTS_SLOW	慢速通道采样时间配置:ADC_CLK 周期数 0x0 ： 1 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

29.9.16 通道完成标志寄存器

名称	CHISR							
地址	0x00000080							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CH_SR							
位权限	W1C-000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CH_SR							
位权限	W1C-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CH_SR							
位权限	W1C-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CH_SR							
位权限	W1C-00000000							

位	位名	功能描述
31:0	CH_SR	ADC 对应通道转换完成标志，写 1 或读对应通道的数据清零

29.9.17 DMA 通道数据寄存器

名称	CHDMA_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			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
20:16	CHID	转换结果对应的通道 ID
15:0	RESULT	ADC 转换结果，每一个转换结果都会被复制到这个寄存器供 DMA 搬运，也可以软件读取

29.9.18 采样通道 x 数据寄存器

名称	CHx_DR（x=0~30）							
地址	0x00000088+x*0x4							
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
31:12	RFU	RFU：未实现，读为 0
11:0	RESULT	ADC 转换结果

30 数模转换器（DAC）

30.1 概述

本芯片集成1个10位电压输出型数模转换器。DAC使用VDDA引脚电压作为参考基准，基准电压范围0~VDDA。

DAC可以为高速比较器提供比较基准。

DAC的主要特性如下：

- 输出信号幅度 VSSA~VDDA
- 最高输出转换率 1Msps（F_{DAC}=16Mhz）
- DAC 输出可以连接到比较器输入和 ADC 输入

30.2 结构框图

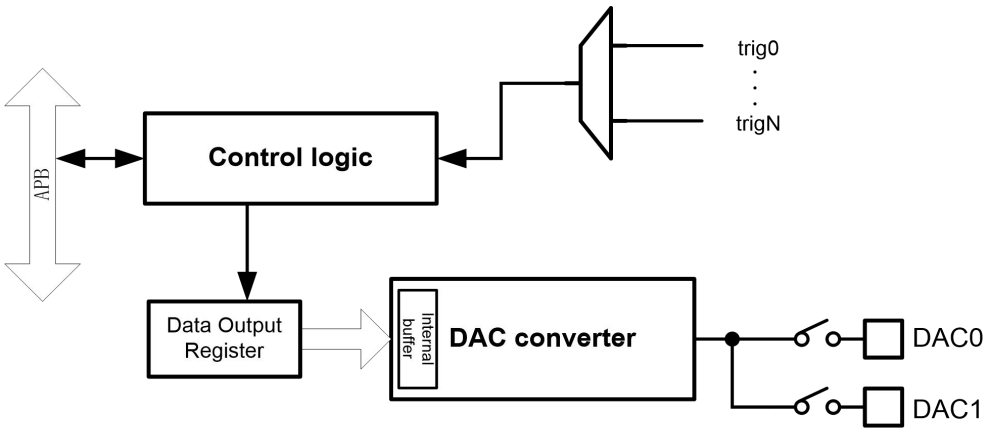


图 30-1 DAC 结构框图

30.3 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
VDDA	模拟电源	DAC 工作电源
VSSA	模拟地	DAC 的参考地
DAC0	模拟输出	DAC 输出信号
DAC1	模拟输出	DAC 输出信号

表 30-1 DAC 相关引脚

DAC输出连接到两个GPIO上，为了保证信号隔离，软件需要配置DAC_CFGR寄存器的OUTEN寄存器，使能相应的模拟开关才能输出模拟电平到GPIO。

30.4 功能描述

30.4.1 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：

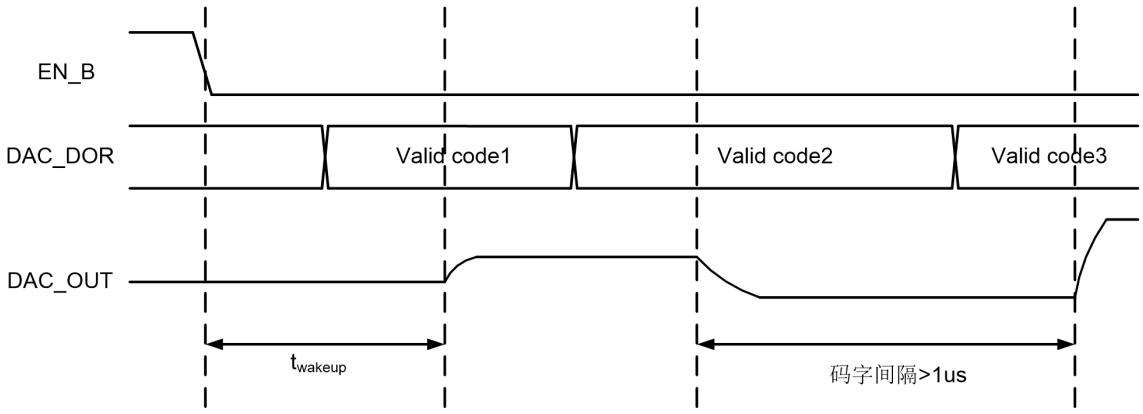


图 30-2 DAC 工作时序

30.4.2 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。

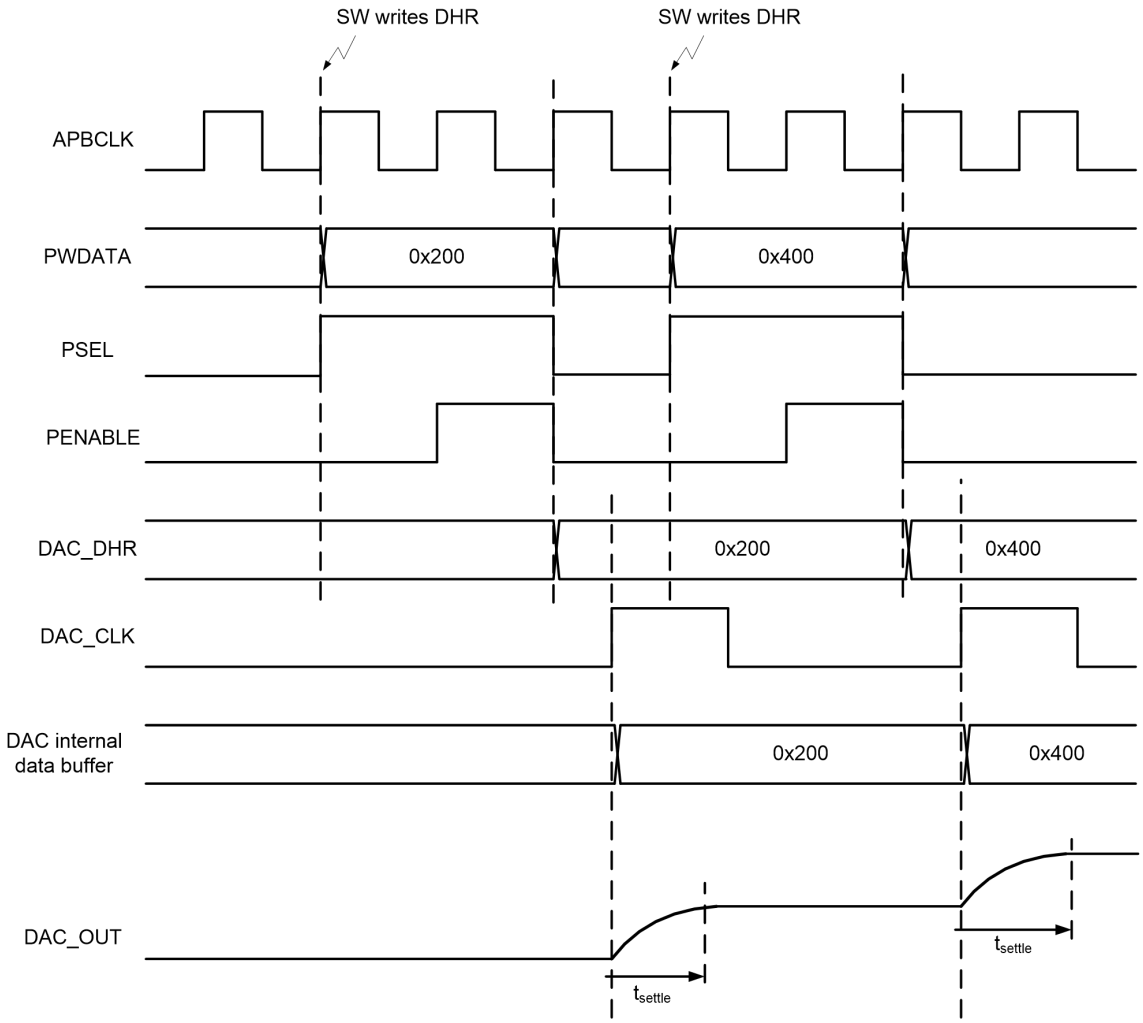


图 30-3 DAC 连续输出模式

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。

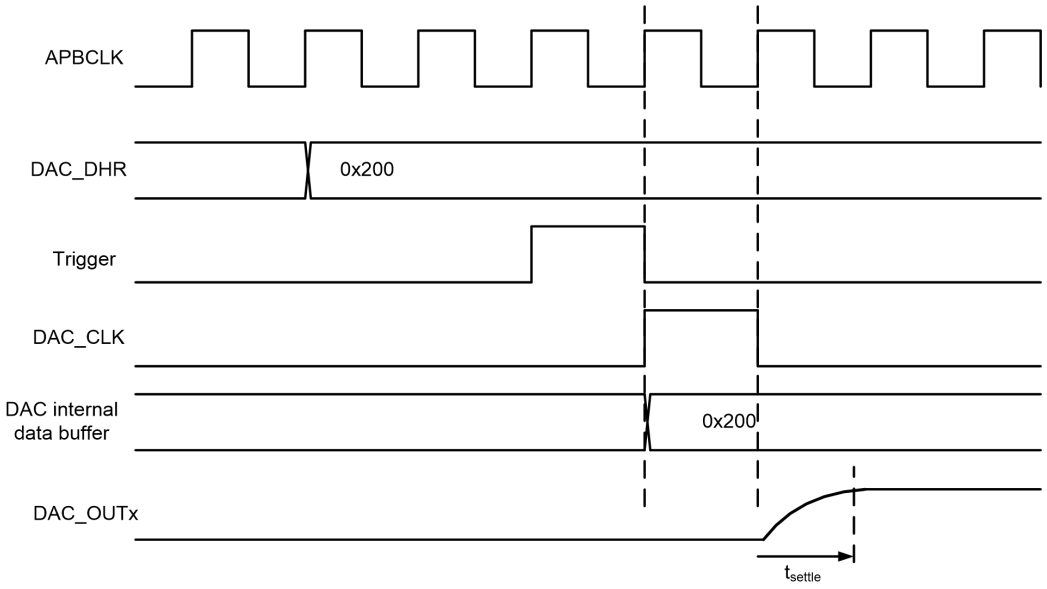


图 30-4 DAC 触发输出模式

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。

30.4.3 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
TAU01_TRGO		0010
TAU02_TRGO		0011
BSTIM_TRGO		0100
LPTIM_TRGO		0101
TAU00_TRGO		0110
TAU03_TRGO		0111
TAU10_TRGO		1000
TAU11_TRGO		1001
TAU12_TRGO		1010
TAU13_TRGO		1011
PA12	引脚中断触发	1100
PA8		1101
PA4		1110
PA0		1111

表 30-2 DAC 触发源

30.4.4 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：

$$DAC_OUT = VDDA \times \frac{DAC_DOR}{4096}$$

30.4.5 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，请在片外连接buffer。

30.5 寄存器

模块起始地址：0x40019800

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		

30.5.1 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

30.5.2 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	-							TRGEN
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	TRGEN	触发模式使能 (Trigger enable) 0：禁止触发模式 1：使能触发模式

30.5.3 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				OUTEN1	OUTEN0
位权限	U-0		R/W-0000				R/W-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 30.4.3DAC 触发源选择
1	OUTEN1	DAC_OUT 连接到 DAC1 的模拟开关使能信号 0：断开开关 1：使能开关

Bit	助记符	功能描述
0	OUTEN0	DAC_OUT 连接到 DAC0 的模拟开关使能信号 0：断开开关 1：使能开关

30.5.4 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。

30.5.5 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)

30.5.6 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	-							DOU
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	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。

30.5.7 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	-							DOU_IE
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

Bit	助记符	功能描述
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

31 电容触摸按键（TSI）

31.1 概述

TSI 模块用于电容式触摸检测和接近检测功能，其主要特性如下。

- 工作电压 2.7～5.5V
- 最大36个独立的按键传感器通道
- 支持自容和互容模式
- 高电容检测灵敏度
- 支持的按键电容范围：5~50pF
- 噪声抑制
  - Sigma-Delta电容检测
  - 多级可配置数字滤波器滤除高频噪声
  - 低辐射发散
- 基准线自动调校，补偿外界环境引起的按键电容缓慢变化
- 防水功能
  - 支持湿手触摸
  - 屏蔽电极驱动
  - 保护传感器
  - 互容模式

31.2 自容模式工作原理

31.2.1 电容检测

按键触摸功能基于电容检测的原理实现。当手指接近触摸按键传感器时，引起寄生电容变化，芯片通过检测电容变化实现按键事件判决。

31.2.2 信噪比（SNR）

在触摸检测应用中，信号定义为按键触发（ON）与非触发（OFF）状态下采样值的变化量。

非触发状态下传感器寄生电容的采样值可以表示为：

$$X(t)=X0+N0(t)$$

- X0是X(t)的均值

- N0(t)是非触发状态下的噪声成分

触发状态下的传感器采样值表示为：

$$X(t)=X1+N1(t)$$

- X1是X(t)的均值
- N1(t)是触发状态下的噪声成分

我们定义X0为传感器采样值的基线（baseline），表示传感器非触发条件下的平均输出，而X1和X0的差值定义为信号：

$$S=X1-X0$$

噪声成分N0和N1是相似的但是并不相同，比如当手指触摸传感器时，由于人体对于50Hz工频信号表现为天线，因此传感器会通过手指耦合到更多高频噪声，而这些高频噪声是没有手指接近时不存在的。我们定义系统噪声为按键OFF状态下的峰值噪声：

$$N=\max(N0(t))=\max(X0(t))-\min(X0(t))$$

因此，系统信噪比定义为SDR=S/N

为了实现可靠的按键检测，通常要求系统SNR达到5:1以上。

31.2.3 按键触摸判定

当处于OFF状态的按键信号强度超过触摸阈值（touch_thrsh+hys）后，判定为按键触摸事件；当处于ON状态的按键信号强度低于释放阈值（touch_thrsh-hys）后，判定为按键释放事件。防弹跳（debounce）寄存器定义了连续多次超过触摸阈值才能被判定为按键触摸事件，以及连续多次低于释放阈值才能被判定为按键释放事件。

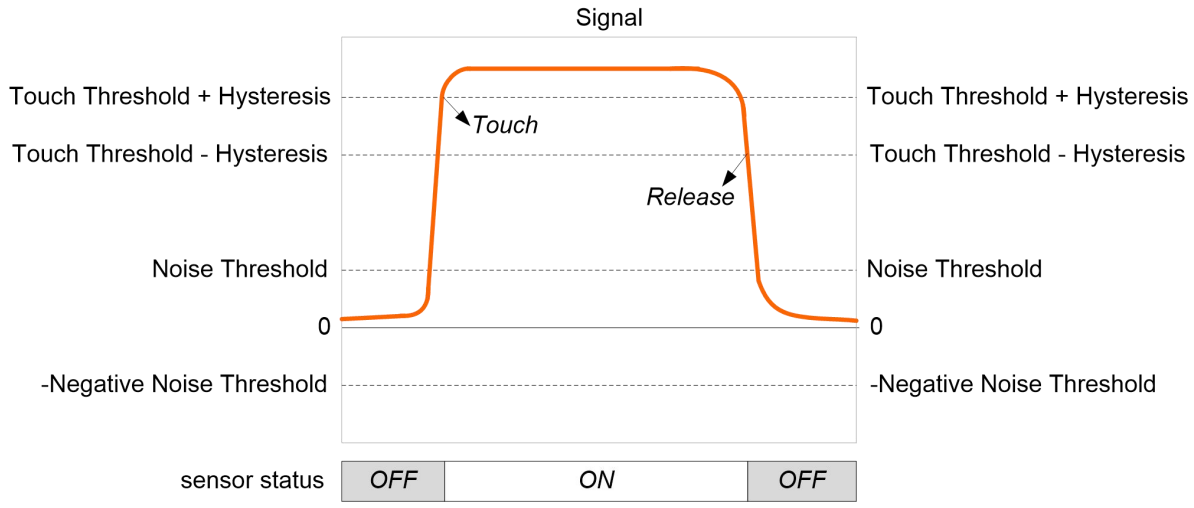


图 31-1 按键触摸与释放

注意到按键判定是基于信噪比，而信噪比是跟baseline相关的，在实际使用中受环境和噪声因素影响，必须要考虑baseline随时间变化的情况，因此应用中需要实现baseline跟踪算法。

31.2.4 接近检测

36个按键通道中的任意一个可以配置为接近检测通道，接近检测用于检测远距离接近事件，通常用于接近唤醒功能。接近检测传感器与按键触摸检测相独立。

31.2.5 按键超时检测

按键超时检测功能用于防止因导体长时间放置或靠近传感器而导致传感器一直处于ON状态。当传感器处于ON状态，超时计数器工作；当处于OFF状态，超时计数器停止工作，并清零。一旦计数值达到阈值，认为检测到超时事件，传感器状态强制切换为OFF状态，同时产生超时中断通知应用程序。

31.2.6 按键互斥

当开启按键互斥功能时，TSI最多同时只允许一个按键被判定为有效的触摸。

31.3 互电容模式工作原理

31.3.1 互电容检测

互容模式需要两个引脚连接到传感器，一个作为发射节点，一个作为接收节点。发射节点对传感器发射特定频率的方波，对传感器互容节点充放电，接收节点通过IDAC吸收或补充电荷，最终使传感器稳定到基准电压上，sigma-delta调制器通过这个方式计算传感器容值变化。

实现互容方案需要两个额外电容：CINTA和CINTB，容值都是470pF

互容模式下raw count可以通过下式计算

$$RawCount = \frac{2V_{TX}F_{TX}C_{M}MaxCount}{IDAC}$$

$$MaxCount = \frac{F_{Mod}N_{Sub}}{F_{TX}}$$

其中：

IDAC=IDAC current

C_M=TX-RX之间电极的互容

V_{TX}=TX信号幅度

F_{TX}=TX时钟频率

Fmod=调制器时钟频率

Nsub=sub-conversion个数

TX时钟频率要求：$F_{TX} < \frac{1}{10R_{seriesTx}C_{PTx}}$

当手指触摸到互容传感器时，传感器的互容减少，导致raw count减少。

31.3.2 Low EMI 驱动

TS24~TS35作为TX节点时，支持Low EMI驱动。这12个PAD的输出驱动可配置slew rate，在需要低辐射的场景下，应优先选用这些IO作为TX管脚。

31.4 Shielding 工作原理

31.4.1 防水（Liquide tolerant）

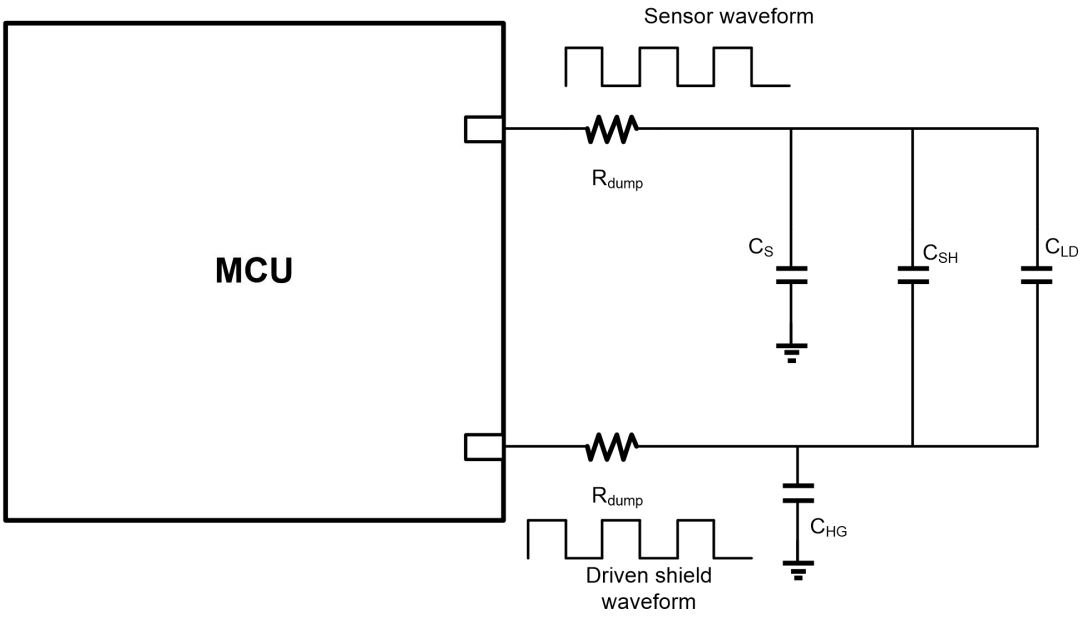
防水检测包含软件baseline跟踪和Guard sensor

- 在水汽和潮湿环境下，基于软件的baseline跟踪可以补偿由水汽引起的缓慢的基线变化，避免潮湿环境下的误触发。
- 当传感器被水流覆盖时，通过Guard sensor（自容模式）可以检测水流，并停止其他传感器的工作以防止误触发。
- 通过shielding节点可以补偿水滴对自容和互容的影响

31.4.1.1 水滴对自容的影响

当电极被水滴覆盖时，对地寄生电容增加，因此在没有触摸的情况下raw count也会增加，可能导致误触发。

为了补偿水滴对自容的影响，应将hatch fill连接到shield驱动上，作为shielding电极使用



在自容shield防水模式下，仅需要选择一个通道作为shield驱动发射端，来驱动hatch fill网络。当hatch fill面积很大，本身寄生电容较大时，对shield驱动能力要求较高，此时可能需要使用external shield driver模式，参见31.4.2

31.4.1.2 Guard sensor

当水流存在于传感器表面，会导致寄生电容大幅度增加，甚至超过手指触摸的效果。此时环绕在按键周围的guard sensor会被触发，软件此时应识别水流存在并关闭扫描，这个时候自容按键不能工作。guard sensor可以使用任意自容通道实现。

31.4.1.3 水滴对互容的影响

互容设计中通常会使用接地的hatch fill来提高噪声免疫。当水滴同时落在sensor和hatch上时，互容减少，效果类似于手指触摸；当水滴仅落在sensor上时，寄生电容增加，与手指触摸影响相反，但是可能导致软件进行baseline重置。

31.4.1.4 同时使用自容和互容扫描来避免水滴影响

结合自容模式和互容模式，软件可以设计出相关逻辑来识别水滴的影响

结合互容和自容的防水设计中，需要软件配置先做互容扫描，然后再对RX通道做自容扫描，软件将两次扫描的结果结合起来判断是否有水滴影响。

31.4.1.5 Shield 通道

软件可以选择任意按键通道作为shield通道，被选中的通道在不被测量时将对外发射屏蔽波形。
被选为shield通道的按键通道仍可以作为接收通道使用，接收通道工作时shield发射自动关闭，而其他通道被测量时，输出shield波形。此功能可以实现多个自容节点在不测量时被驱动shield波形，实现大面积水滴防护。

比如配置TS1和TS2为自容接收通道，同时置位SHLDEN1和SHLDEN2，则TSI在检测TS1时将从TS2发射屏蔽波形，检测TS2时将从TS1发射屏蔽波形。
需要对大面积ground hatch做shield驱动时，应配置一个单独通道，并禁止这个通道的检测功能。

配置说明举例：假设使用TS5驱动ground hatch，使用TS0~4作为自容检测，并且非活跃通道输出shield驱动波形，自电容多次扫描（multi-scan）。

- SHLDEN[5]=1, SHLDEN[4:0]=11111
- SCAN_MODE=01
- CHEN[5:0]=011111 （TS5不参与扫描）
- CHCFG[5:0]=000000
- TSEN=1

31.4.2 External Shielding Driver

此模式下使用GPIO开关驱动Shield节点，同时需要额外的C_{TANK}电容。
C_{TANK}用于增强shield驱动能力，适用于大面积shielding电极设计。

31.4.3 Internal Shielding Driver

此模式下使用CSD模块内的low-EMI开关驱动Shield节点，不需要C_{TANK}，适用于小面积shielding电极设计。

31.5 通道扫描

31.5.1 扫描模式

TSI支持以下扫描模式

- 自电容单次模式（single-scan）
- 自电容多次模式（multi-scan）
- 互电容模式（full-scan）

自电容单次模式（single-scan）

单次模式下TSI只进行一次扫描，可以扫描一个通道或者同时扫描多个通道（用于接近检测功能）。TSI_CH_CR寄存器中被使能的通道会被同时扫描，扫描完成后TSI自动停止。

自电容多次模式（multi-scan）

多次模式下，TSI会按照升序逐个扫描TSI_CH_CR中使能的通道，不支持多个通道同时扫描。WAIT寄存器用于配制通道之间是否需要更新通道参数：

WAIT=0：通道之间不修改参数，所有通道使用相同参数扫描

WAIT=1：通道之间修改参数，开始下一个通道扫描前需等待软件或者DMA更新TSI_CH_CFGR；软件更新后需要置位CONTINUE寄存器使TSI继续扫描，DMA更新后硬件自动执行后续扫描

互电容模式（full-scan）

互电容模式下，TSI按照升序扫描接收通道，同时也按照升序激活对应的发射通道。WAIT寄存器用于配制通道之间是否需要更新通道参数：

WAIT=0：通道之间不修改参数，所有通道使用相同参数扫描

WAIT=1：通道之间修改参数，开始下一个通道扫描前需等待软件或者DMA更新TSI_CH_CFGR；软件更新后需要置位CONTINUE寄存器使TSI继续扫描，DMA更新后硬件自动执行后续扫描

31.5.2 自电容扫描过程

TSI最大支持36个检测通道，TSI按照编号从小到大的规则逐个扫描使能的按键通道（不使能的通道跳过）。

SDM分辨率决定了每个通道单次采样时间，滤波器的配置参数和SMPL_SIZE寄存器决定了每个通道在一轮扫描中被采样的次数。比如某个通道使能了16阶均值滤波的情况下，对此通道连续采样16次后输出一个结果，实际采样次数由 SMPL_SIZE * AVG_ORGER决定。在没有使能滤波器的情况下，实际采样次数就是SMPL_SIZE。

每个通道扫描完成后，会产生中断，此时软件或者DMA可以读取raw count，并且软件或DMA可以在下一个通道扫描之前更新IDAC和滤波器配置参数。所有通道扫描完成后，CSD sequencer停止，等待下一次触发。

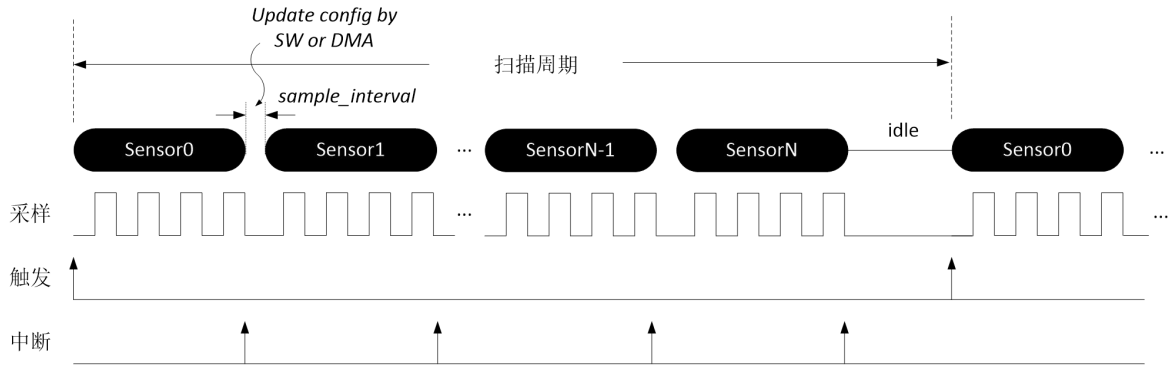


图 31-2 通道扫描周期

使能扫描后，TSI模块将以升序方式逐个扫描被使能的通道。比如使能了通道0、3、5、6，则扫描顺序为TS0->TS3->TS5->TS6：

Receive Channels							
Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
(1)			(2)		(3)	(4)	

31.5.3 互电容扫描过程

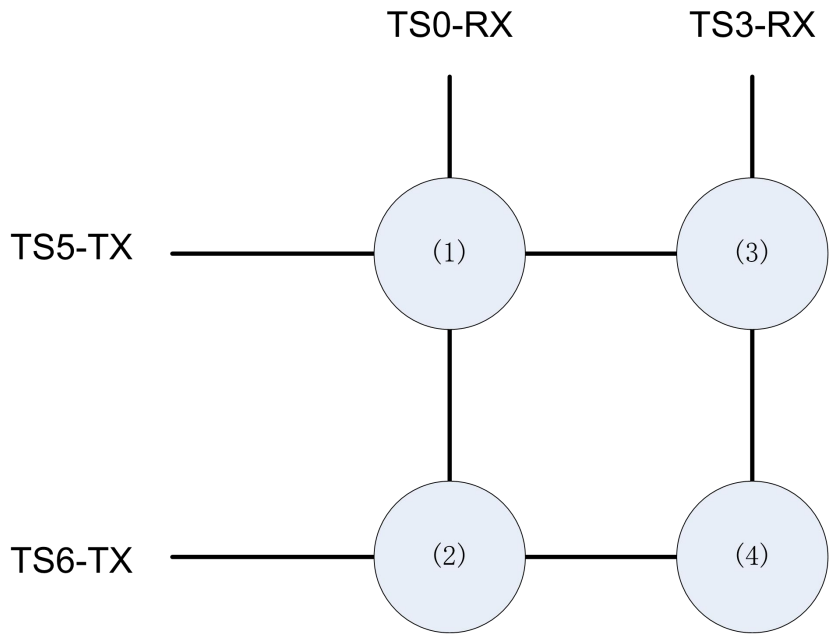
互电容扫描除了需要使能通道外，还要分配通道属性：发射或接收。

发射通道和接收通道都按照升序执行，在每个接收通道完成采样后，会产生中断，此时软件或者DMA可以读取raw count，并且软件或DMA可以在下一个通道扫描之前更新IDAC和滤波器配置参数。所有接收通道扫描完成后，CSD sequencer停止，等待下一次触发。

比如配置通道0、3为接收，通道5、6为发射，则扫描顺序是 (TS5-TS0) -> (TS5-TS3) -> (TS6-TS0) -> (TS6-TS3)：

		Receive channels			
		Channel 0	Channel 1	Channel 2	Channel 3
Transmit channels	Channel 4				
	Channel 5	(1)			(3)
	Channel 6	(2)			(4)
	Channel 7				

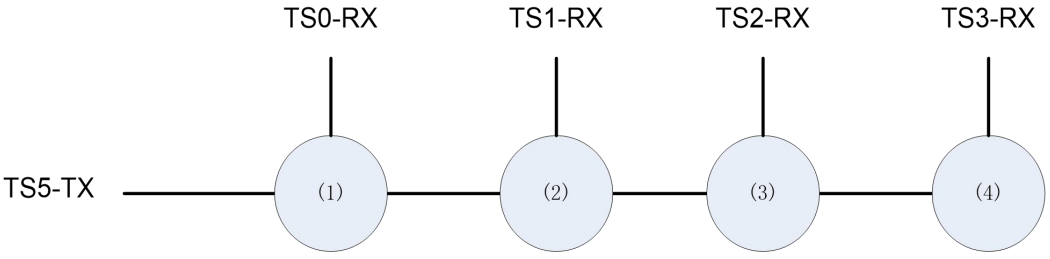
以上配置可以形成2x2互容扫描阵列，并以下图编号顺序逐个扫描互容节点：



使用这种方式可以轻松实现一个TX连续扫描多个RX：

		Receive channels			
		Channel 0	Channel 1	Channel 2	Channel 3
Transmit channels	Channel 4				
	Channel 5	(1) →	(2) →	(3) →	(4)
	Channel 6				
	Channel 7				

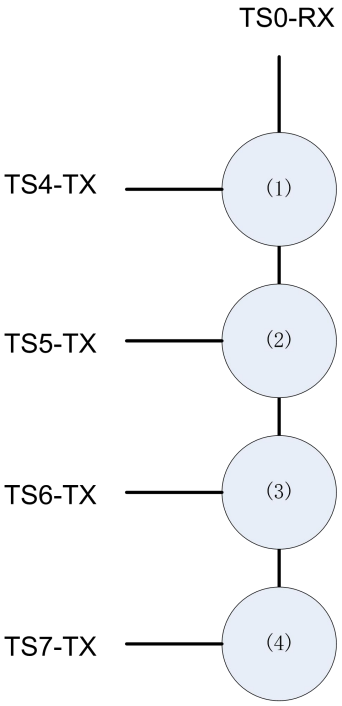
以上配置可以形成1x4互容行扫描阵列：



多个TX连续扫描一个RX：

		Receive channels			
		Channel 0	Channel 1	Channel 2	Channel 3
Transmit channels	Channel 4	(1)			
	Channel 5	(2)			
	Channel 6	(3)			
	Channel 7	(4)			

以上配置可以形成4x1互容列扫描阵列：



互容发射驱动只有一个脉冲生成数字电路，可以连接到所有TS引脚的数字输出；其中TS0~TS7支持低压buffer驱动，连接到IO的模拟通道上。

31.5.4 互电容扫描时的 IO 控制

当IO配置为TSI功能时，GPIO控制器将对应IO设置为模拟功能，关闭数字输入输出。

但是当某个TSI通道作为互容发射脚使用时，实际是发射数字波形。此时，当前被激活的TX引脚由TSI模块强制控制为数字输出功能，以便发射TX波形。

实际应用中，软件仅需将TSI引脚配置为模拟功能，TSI模块会自动在TX发射时将当前发射引脚控制成数字输出，这部分功能不需要通过GPIO配置寄存器来实现。当TX引脚闲置时，TSI模块将其配置为HiZ或者推挽输出0的状态，由配置寄存器决定。

当使能的RX引脚闲置时，可以配置为HiZ或者推挽输出0的状态。

31.5.5 扫描触发

TSI模块使能后，需等待触发信号到来才开始扫描。按键扫描可以通过软件触发或者外设trigger信号触发。

如果使用软件触发，START置位后TSI开始扫描，直到扫描结束后START自动清零，并置位序列完成中断。

如果使用硬件外设触发，START寄存器由硬件置位。由于外设trigger信号是系统时钟域的，使用硬

件触发时必须保持TSI的总线时钟使能。

TSI模块支持以下触发信号输入：

- RTC TRGO
- ATIM TRGO
- LPTIM TRGO
- BSTIM TRGO
- TAU TRGOx

WAIT寄存器用于控制每个通道扫描完成后是否等待软件或DMA更新通道配置参数：

- WAIT=0，当前通道扫描完成后，等待SCAN_INTV时间后，立即启动下一个通道扫描
- WAIT=1，当前通道扫描完成后，需要等待软件（或者DMA）置位CONTINUE寄存器，否则即使SCAN_INTV时间后也不会启动下一个通道扫描
- CONTINUE寄存器可以由软件置位，DMA模式下则在DMA更新完配置参数后硬件自动置位，当下一个通道扫描开始后CONTINUE自动清零；WAIT=0时CONTINUE无法写入

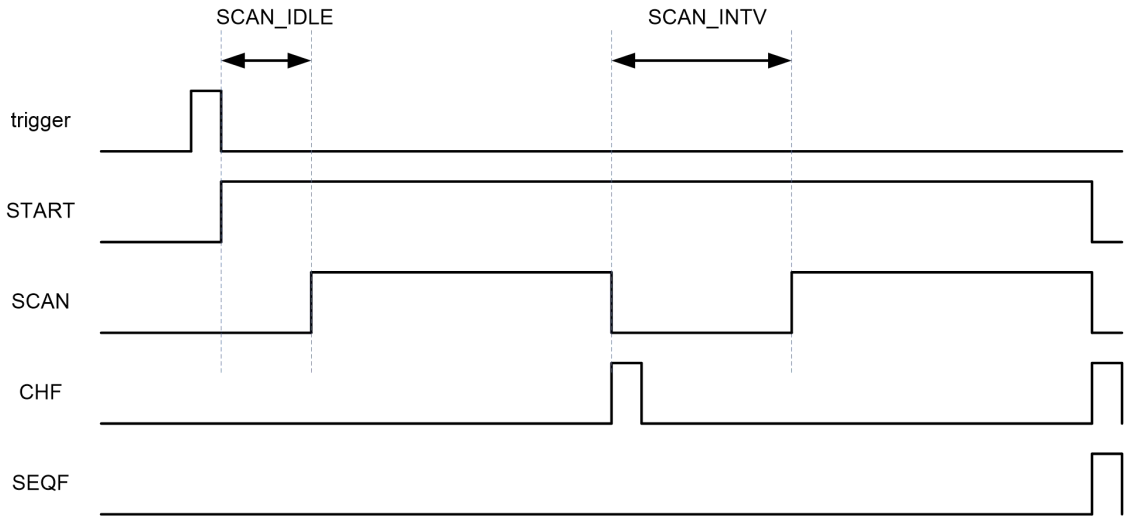


图 31-3WAIT=0，扫描 2 个通道

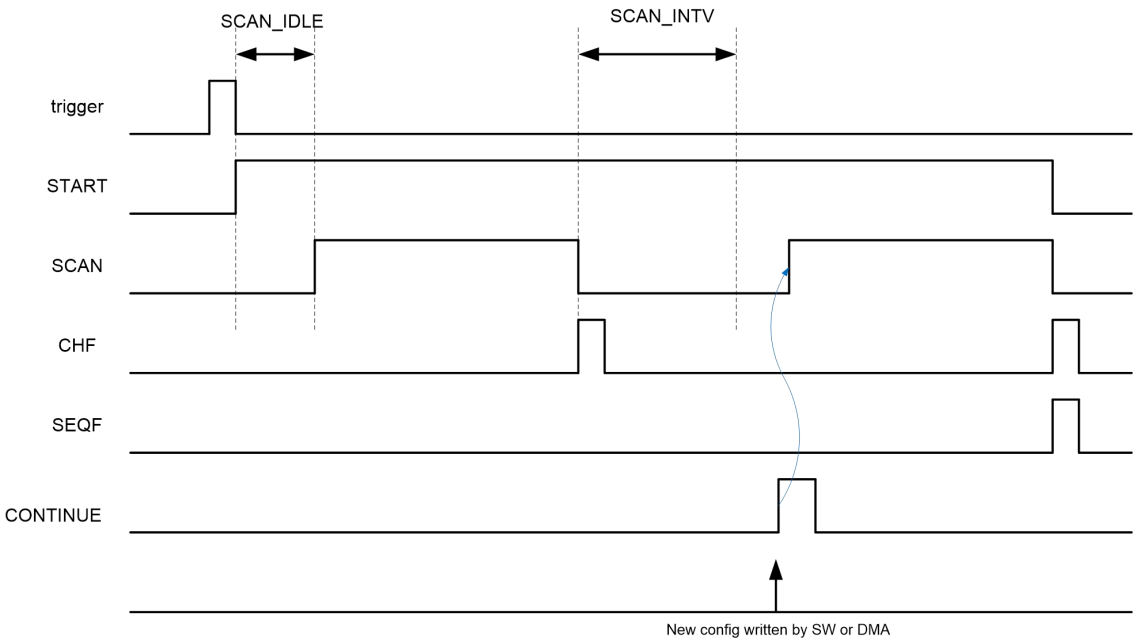


图 31-4WAIT=1，扫描 2 个通道

31.5.6 扫描停止

当START置位后，CSD sequencer会持续工作直到所有使能的通道扫描完成。在此过程中，如果用户希望终止扫描，可以置位STOP寄存器。

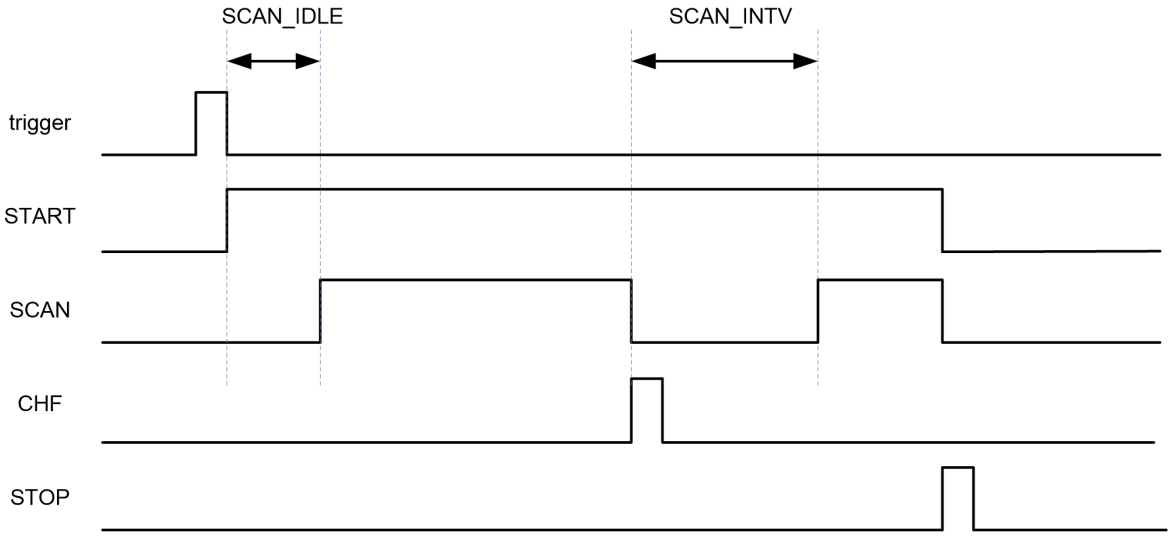


图 31-5 扫描停止

31.5.7 扫描通道配置

自电容配置：

- 时钟配置
- 使能被扫描的通道
- 触发源
- 扫描间隔时间
- 是否需要等待通道参数更新
- 是否使用DMA用于结果传输和参数更新

互电容配置：

- 时钟配置
- 使能被扫描的通道
- 配置通道属性：发射或接收
- 触发源
- 扫描间隔时间
- 是否需要等待通道参数更新
- 是否使用DMA用于结果传输和参数更新

31.5.8 DMA 支持

TSI支持两种DMA传输：读传输（DMA_RDEN）和写传输（DMA_WREN）。

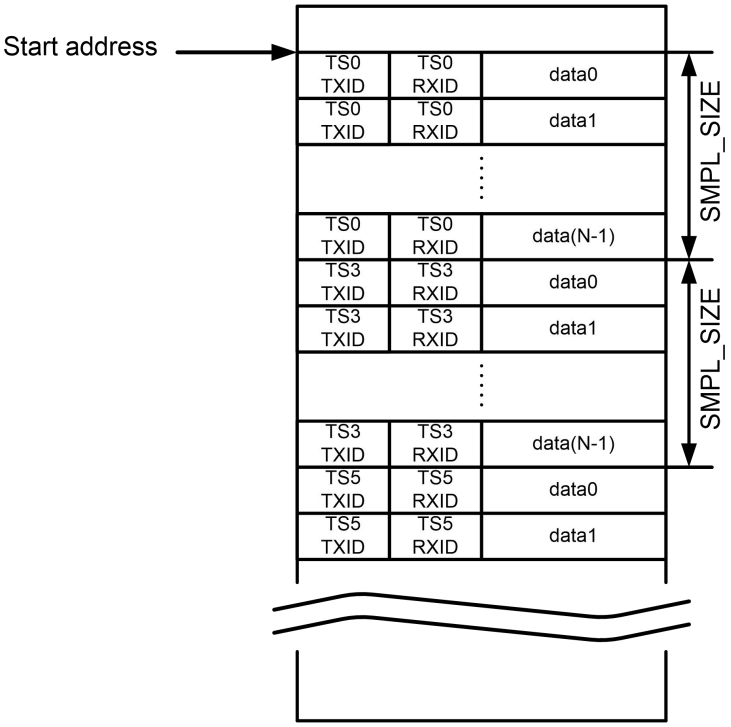
DMA READ

应用中可以使用DMA将TSI输出的raw count或者FIR滤波结果搬运到RAM指定地址。raw count和滤波输出最大都是16bit结果，同时搬运的还有raw count对应的通道ID（发射和接收），总数据位宽32bit，因此DMA搬运为32bit单位，每个结果都占据一个word。在不使能FIR功能时，DMA搬运的是raw count，在使能FIR功能时，DMA搬运的是滤波后的结果。

不同通道的数据在RAM中连续存储，软件可以通过通道ID来识别数据来自于哪个通道。

DMA READ配合DMA循环模式使用，可以将同一通道数据映射到固定RAM地址。

下图例子中使能了TS0、TS3、TS5，每个通道采样次数由SMPL_SIZE定义，数据依次排列：

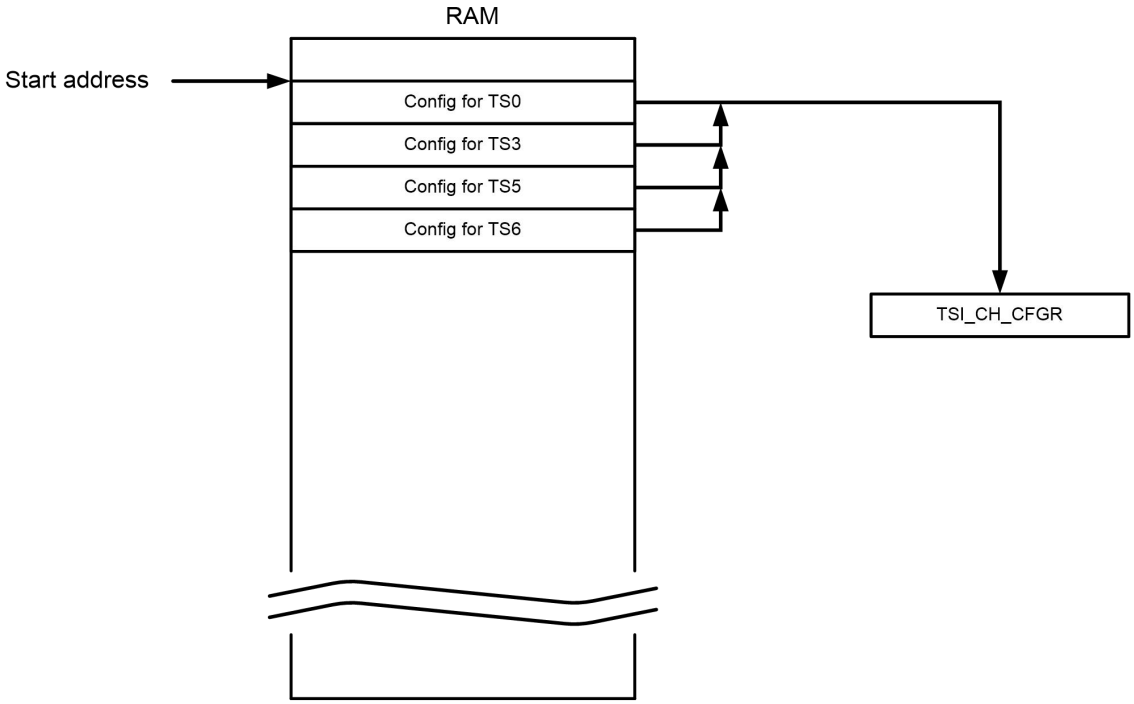


DMA WRITE

此功能用于在多次扫描过程中更新通道参数。当WAIT=1，当前通道扫描完成后TSI发出DMA请求，DMA将RAM中保存的通道参数写入TSI_CH_CFGR，自动完成下一通道的参数配置。

DMA WRITE配合DMA循环模式使用，可实现对每个通道使用独立的参数。

假设使能了TS0、TS3、TS5、TS6，他们的配置参数在RAM中连续存储，循环使用，软件应保证RAM中的存储顺序与实际使用通道顺序保持一致。



31.6 工作时钟

TSI工作时钟结构如下图所示。

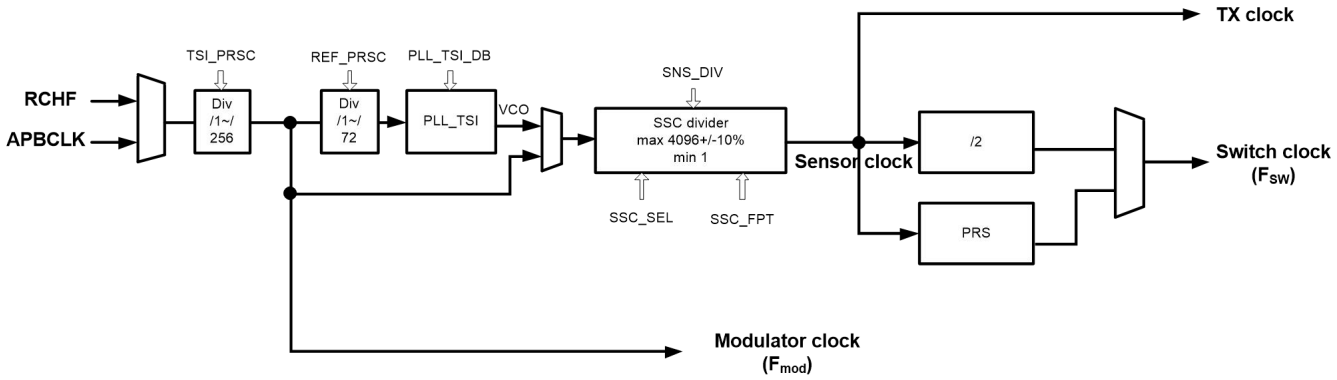


图 31-6TSI 工作时钟

其中PLL_TSI是一个独立的PLL，用于TSI实现跳频（多频点测量）和扩频（SSC）功能。在自容应用中，F_{SW}可以是PRS时钟或者non-PRS时钟；但是在互容应用中，TX clock不能使用PRS，如果需要降低发射EMI，需要使用SSC时钟。

PLL_TSI与系统PLL相同，中心频率80M，VCO中心值160M；由于这是TSI内部模块，控制寄存器位于TSI模块内。

TSI工作时钟经过CMU选择和预分频，相关寄存器在CMU模块内。

使用场景：

1）自容应用：Fsw是Fmod的分频，Fmod可以是RCHF或者APBCLK，两种配置都可以实现跳频，依靠PRS实现EMI抑制

2）互容应用：Fmod可以是RCHF或者APBCLK；TX clock可以选RCHF、APBCLK、PLL或者PLL SSC分频

为了实现更好的EMI抑制，TSI支持2种PRS，生成多项式长度分别为8和12，两种PRS的多项式都是软件可配置的。软件可以选择使用PRS8还是PRS12，同时软件可以任意配置多项式。

其中，SSC divider是一个带有LFSR调制的同步分频器，参见SSC原理章节

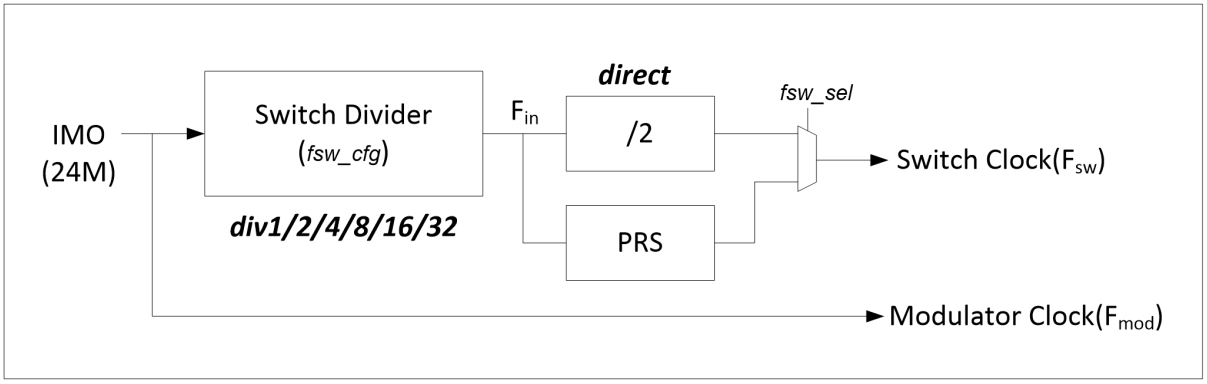


图 31-7 开关时钟与调制器时钟

当使用PRS时，平均的开关频率可以表示为：

$$F_{SW_AVERAGE} = \frac{F_{IN}}{4}$$

在配置TSI工作时钟时，需注意满足以下条件：

- 开关频率F_{SW}必须保证按键电容C_S的“满充满放”，即

  $$T_{SW} \geq 10 \cdot R_{SerialsTot\ al} \cdot C_S$$

- 如果是能PRS时钟，则开关频率是变化的，其最高频率需满足上述条件
- 一次完整的按键扫描周期必须大于等于PRS周期，即

  $$\frac{2^{N_PRS}-1}{F_{\text{in}}} \leq \frac{2^{N}-1}{F_{\text{mod}}}$$

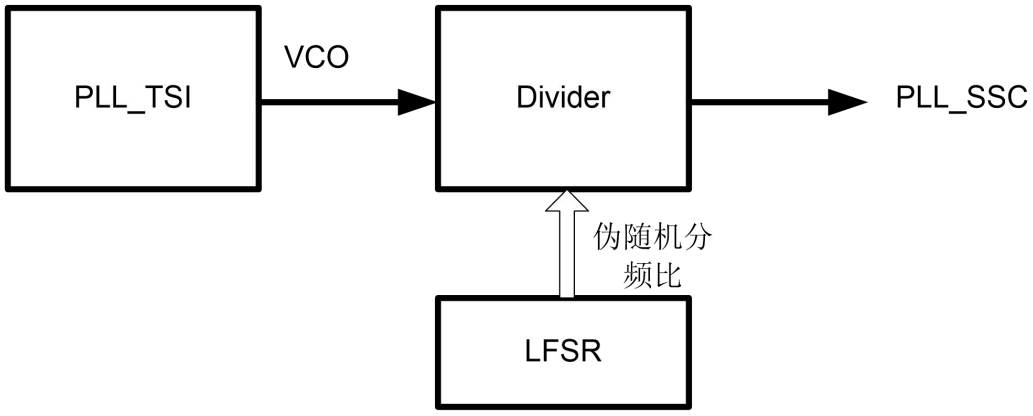
其中N_PRS=8，即PRS循环周期为255个Fin时钟。N是sigma-delta调制器分辨率，即Raw data数据位宽。Fmod是调制器工作时钟，也就是raw data积分时钟，用于采样并积分SDM输出码流。

31.6.1 SSC 原理

互容模式下，不能使用PRS时钟，因此需要SSC时钟实现EMI抑制，通过LFSR调整PLL_TSI输出时钟的分频比，实现中心分频比上下10%的调整范围。

比如中心分频比160，LFSR调制后的分频比为144~176，在这个范围内获得32个平均分布的时钟频率。在使能SSC的情况下，分频器的分频比被4种LFSR调制，实现扩频功能；在禁止SSC的情况下，分频器的分频比不做调制，直接输出中心频率。

PLL_TSI输出160MHz的情况下，SSC divider支持的中心频率输出为80M~39KHz。如果需要使用扩频功能，为了保证最佳的频谱扩散效果（多频点在中心频率左右均匀分布），必须将分频比配置为160的整数倍。



LFSR支持四种多项式长度：SSC6 SSC7 SSC9 SSC10

使用更长的多项式可以获得更高的分频比，比如320+/-10%、1024+/-10%，相应得到的modulator clock频率会更低。SSC支持的分频比是160、320、640、1280.

31.7 多频点测量

TSI可以实现跳频功能，以抑制同步噪声对按键测量的影响。多频点测量通过修改TSI工作时钟频率来改变F_{SW}驱动频率，并在不同驱动频率下完成多次测量，根据多数表决来判断是否有按键动作。

跳频功能有两种实现方案：

1）TSI工作时钟使用RCHF，此时软件可以通过修改RCHF trim值实现跳频；注意此时系统时钟不能是RCHF，否则将影响通信和系统定时。

2）TSI工作时钟使用PLL_TSI，此时软件可以通过修改PLL_TSI倍频比实现跳频。

一般推荐跳频范围是+/-15%

使用方案2）时，建议将PLL_TSI输出设置为48MHz，二分频后得到24M的IMO；进行跳频时，修改PLL_TSI倍频比寄存器改变PLL_TSI输出。比如三次测量使用倍频比为48 42 54，得到PLL_TSI输出频率为48M、42M、54M，F_{SW}采用16*2=32分频，则开关频率为1.5M、1.3125M、1.6875M，跳频范围+/-12.5%

31.8 滤波器和 baseline 跟踪

TSI数字滤波器整体架构包括按键触摸滤波器组和接近检测滤波器组。每组滤波器包含primary FIR滤波器（硬件均值滤波）和后续的secondary滤波器（软件IIR、中值滤波、抖动滤波）。所有按键通道复用同一组FIR滤波器，secondary滤波器为软件实现的辅助功能。

复旦微电子提供完整的软件库函数供用户使用。

关于滤波器实现函数和基线跟踪，请参考复旦微电子应用手册或联系复旦微电子公司销售代表。

31.9 中断和状态标志

TSI模块在扫描过程中会产生多种中断事件：

- Raw count更新事件
- 通道扫描完成事件
- 序列扫描完成事件

在每次采样完成后，硬件更新raw count数值到TSI_RAW_CNTR寄存器中，同时置位RCUF标志寄存器，在RCU_IE=1的情况下同时产生中断，通知软件读取新的raw count。如果使能了滤波器功能，则TSI_RAW_CNTR寄存器中保存的是滤波后的数据，更新频率降低为（采样频率/滤波器阶数），RUIF标志寄存器在滤波器更新输出时置位。RCUF标志由软件写1清零，或者软件/DMA读取TSI_RAW_CNTR时自动清零。

每个通道的采样次数由SMPL_SIZE寄存器配置。当完成SMPL_SIZE次采样后，当前通道扫描结束，CHF标志寄存器置位，在CH_IE=1的情况下同时产生中断。如果使能了滤波器，则SMPL_SIZE定义的是滤波器输出个数。CHF标志由软件写1清零。

当所有被使能通道都扫描完成后，SEQF标志寄存器置位，表示一轮扫描结束，在SEQ_IE=1的情况下同时产生中断。SEQF标志由软件写1清零。

下图是使能4个按键通道，每个通道采样2次的中断标志示意图：

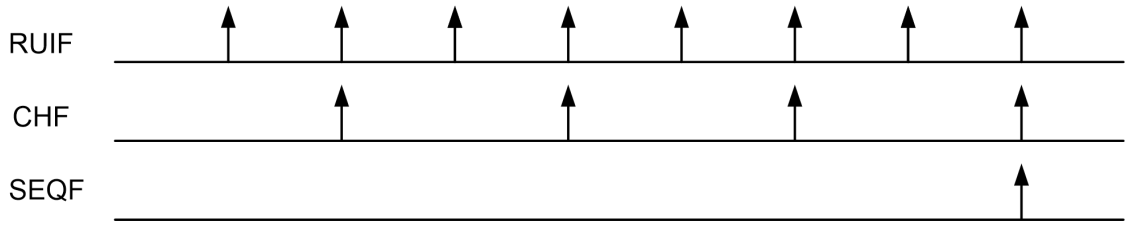


图 31-8SMPL_SIZE=2，扫描 4 个通道

31.10 VDD 电压适配

VDD在4~5V变动时会导致raw count变化，通过配置0x4000_2800地址的PDR_CFG寄存器相关bit可以改善该现象。

31.11 设备调校

关于设备调校方法，请参考复旦微电子应用手册或联系复旦微电子公司销售代表。

31.12 寄存器

模块起始地址：0x4001_0000

offset 地址	名称	符号
0x00	按键中断标志寄存器	TSI_ISR
0x04	按键中断使能寄存器	TSI_IER
0x08	接收扫描状态寄存器	TSI_RXSR
0x0C	发送扫描状态寄存器	TSI_TXSR
0x10	扫描配置寄存器	TSI_CFGR
0x14	扫描控制寄存器	TSI_CR
0x18	按键通道使能寄存器 0	TSI_CH_CR0
0x1C	按键通道使能寄存器 1	TSI_CH_CR1
0x20	按键通道配置寄存器 0	TSI_CH_CFGR0
0x24	按键通道配置寄存器 1	TSI_CH_CFGR1
0x28	RawCount 寄存器	TSI_RAW_CNTR
0x2C	时钟和周期控制寄存器	TSI_CCR
0x30	DMA 控制寄存器	TSI_DMACR
0x34	按键通道扫描参数配置寄存器	TSI_SP_CFGR
0x38	IDAC 调校寄存器	TSI_IDAC_TR
0x3C	IDAC 模拟控制寄存器	TSI_ANA_CR
0x40	DFT 寄存器	TSI_TEST
0x44	测试电容通道使能寄存器 0	TSI_TESTRING0
0x48	测试电容通道使能寄存器 1	TSI_TESTRING1
0x4C	Analog test 寄存器	TSI_ANA_TEST
0x50	VREF 基准配置寄存器	TSI_VREFCR_TEST
0x54		
0x58	屏蔽通道增强寄存器 0	TSI_SHLDGND_CR0
0x5C	屏蔽通道增强寄存器 0	TSI_SHLDGND_CR1
0x60	低辐射驱动配置寄存器 0	TSI_EMI0_CFGR
0x64	低辐射驱动配置寄存器 1	TSI_EMI1_CFGR
0x68	屏蔽通道使能寄存器 0	TSI_SHLD_CR0
0x6C	屏蔽通道使能寄存器 1	TSI_SHLD_CR1
0x70	时钟管理寄存器	TSI_CKCR
0x74	自容伪随机生成多项式寄存器	TSI_PRS_SW
0x78	PLL 控制寄存器	TSI_PLL_CR
0x7C	互容 SSC 生成多项式寄存器	TSI_PRS_SSC

31.12.1 按键中断标志寄存器（TSI_ISR）

名称	TSI_ISR
offset	0x00

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

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	SEQF	扫描序列完成 (Sequence Finished)，所有通道完成一轮扫描后置位，软件写 1 清零
1	CHF	当前通道扫描完成 (Channel Finished)，硬件置位，软件写 1 清零
0	RCUF	Raw count 更新标志寄存器 每完成一次采样，产生新的 raw count 时置位，软件写 1 清零，或者软件读取 raw count 寄存器时自动清零

31.12.2 按键中断使能寄存器（TSI_IER）

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

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	SEQ_IE	扫描序列完成中断使能 0：禁止中断 1：使能中断
1	CH_IE	当前通道扫描完成中断使能 0：禁止中断 1：使能中断

Bit	助记符	功能描述
0	RCU_IE	Raw count 更新中断使能 0：禁止中断 1：使能中断

31.12.3 接收通道扫描状态寄存器（TSI_RXSR）

名称	TSI_RXSR							
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	-		RXSR					
位权限	U-0		R-000000					

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:0	RXSR	接收通道扫描状态寄存器，表示当前正在扫描的接收通道（自容模式下就是当前活跃通道，互容模式下是当前接收通道） 000000：没有通道正在扫描 000001：正在扫描通道 0 000010：正在扫描通道 1 …… 100000：正在扫描通道 31 100001：正在扫描通道 32 100010：正在扫描通道 33 100011：正在扫描通道 34 100100：正在扫描通道 35 others：RFU

31.12.4 发射通道扫描状态寄存器（TSI_TXSR）

名称	TSI_TXSR							
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	-		TXSR					
位权限	U-0		R-000000					

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:0	TXSR	发射通道扫描状态寄存器，表示当前正在工作的发射通道（仅互容模式下有效，受 sequncer 控制） 000000：没有通道正在发射 000001：正在发射通道 0 000010：正在发射通道 1 …… 100000：正在发射通道 31 100001：正在发射通道 32 100010：正在发射通道 33 100011：正在发射通道 34 100100：正在发射通道 35 others：RFU

31.12.5 扫描配置寄存器（TSI_CFGR）

名称	TSI_CFGR							
offset	0x10							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SMPL_SIZE				-			
位权限	R/W-0000				U-0			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SCAN_MODE		RFUI	WAIT	-			
位权限	R/W-00		R/W-0	R/W-0	U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	HWTRG_SEL				AVG_ORDER			
位权限	R/W-0000				R/W-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AVG_EN	-	TMOD	SHIELD_EN	-	-	-	TSEN
位权限	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0

Bit	助记符	功能描述
31:28	SMPL_SIZE	每个通道的采样次数配置。在使能滤波器的情况下，SMPL_SIZE 指的是滤波输出的个数；在没有使能滤波器的情况下，SMPL_SIZE 是 raw count 输出个数。 0000：1 次 0001：2 次 …… 1111：16 次

Bit	助记符	功能描述
27:24	--	RFU：未实现，读为 0
23:22	SCAN_MODE	扫描模式寄存器 00:single scan 01: multi-scan 10: full scan 11: RFU
21	RFUI	Dummy 寄存器
20	WAIT	通道间是否等待参数更新，仅 multi-scan 和 full-scan 模式下有效 0：不等待参数更新 1：等待参数更新
19:16	--	RFU：未实现，读为 0
15:12	HWTRG_SEL	硬件触发信号选择 0000：RTC TRGO 0001：ATIM TRGO 0010：LPTIM16 TRGO 0011：BSTIM16 TRGO 0100：TAU00 TRGO 0101：TAU01 TRGO 0110：TAU02 TRGO 0111：TAU03 TRGO 1000：TAU04 TRGO 1001：TAU05 TRGO 1010：TAU06 TRGO 1011：TAU07 TRGO 1100：TAU10 TRGO 1101：TAU11 TRGO 1110：TAU12 TRGO 1111：TAU13 TRGO
11:8	AVG_ORDER	按键检测硬件均值滤波器阶数控制，仅 BUTTON_AVG_EN=1 时有效 0000：2 阶 0001：4 阶 0010：8 阶 0011：16 阶 Others：RFU
7	AVG_EN	按键检测硬件均值滤波器使能 0：关闭硬件均值滤波器 1：使能硬件均值滤波器 注意：硬件均值滤波器使能后对所有按键通道有效
6	--	RFU：未实现，读为 0
5	TMOD	触发模式选择 (Trigger mode) 0：软件触发，软件写 START 寄存器启动扫描 1：硬件触发，外设触发事件启动扫描
4	SHIELD_EN	屏蔽电极功能使能，用于防水（小水滴）功能 0：不使能 1：使能
3	--	RFU：未实现，读为 0
2	--	RFU：未实现，读为 0
1	--	RFU：未实现，读为 0

Bit	助记符	功能描述
0	TSEN	扫描模块使能 0：关闭 TSI 1：使能 TSI

31.12.6 扫描控制寄存器（TSI_CR）

名称	TSI_CR							
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	-					STOP	CONTINUE	START
位权限	U-0					WO-0	WO-0	R/W-0

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	STOP	对该寄存器写 1，扫描立刻停止
1	CONTINUE	多通道扫描 WAIT=1 时，软件置位此寄存器执行下一个扫描 软件可以在修改配置参数后再启动下一个通道的扫描
0	START	软件写 1 启动扫描，所有使能通道扫描结束后硬件自动清零 仅 SEN=1 时可写 软件写 0 无效

31.12.7 按键通道使能寄存器 0（TSI_CH_CR0）

名称	TSI_CH_CR0							
offset	0x18							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHEN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHEN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	CHEN[7:0]
位权限	R/W-0000 0000

Bit	助记符	功能描述
31:0	CHEN	通道使能寄存器，写 1 使能，清零禁止 bit0: TS0 bit1: TS1 … bit31: TS31

31.12.8 按键通道使能寄存器 1（TSI_CH_CR1）

名称	TSI_CH_CR1							
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	-				CHEN[35:32]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	CHEN	通道使能寄存器，写 1 使能，清零禁止 bit0: TS32 bit1: TS33 bit2: TS34 bit3: TS35

31.12.9 按键通道配置寄存器 0（TSI_CH_CFGR0）

名称	TSI_CH_CFGR0							
offset	0x20							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CHCFG[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CHCFG[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHCFG[15:8]							
位权限	R/W-0000 0000							

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

Bit	助记符	功能描述
31:0	CHCFG	通道配置寄存器 0：receiver 1：transmitter 仅互容模式下有效，自容模式下所有通道都是 receiver bit0: TS0 bit1: TS1 … bit31: TS31

31.12.10 按键通道配置寄存器 1（TSI_CH_CFGR1）

名称	TSI_CH_CFGR1							
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	-				CHCFG[35:32]			
位权限	U-0				R/W-0000			

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	CHCFG	通道配置寄存器 0：receiver 1：transmitter 仅互容模式下有效，自容模式下所有通道都是 receiver bit0: TS32 bit1: TS33 bit2: TS34 bit3: TS35

31.12.11 Raw Count 寄存器（TSI_RAW_CNTR）

名称	TSI_RAW_CNTR							
offset	0x28							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-		TXID					
位权限	U-0		R-11 1111					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

位名	-		RXID					
位权限	U-0		R-11 1111					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RAW_COUNT[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAW_COUNT[7:0]							
位权限	R-0000 0000							

Bit	助记符	功能描述
31:30	--	RFU：未实现，读为 0
29:24	TXID	发射通道 ID 识别寄存器，用于指示当前 Raw count 是使用哪个发射通道，随着 raw count 同步更新；仅互容模式有效。 000000：TS0 通道 000001：TS1 通道 000010：TS2 通道 …… 011111：TS31 通道 100000：TS32 通道 …… 100101：TS35 通道 others：无效
23:22	--	RFU：未实现，读为 0
21:16	RXID	接收通道 ID 识别寄存器，用于指示当前 Raw count 是来自于哪个接收通道的，随着 raw count 同步更新；互容和自容模式有效。 000000：TS0 通道 000001：TS1 通道 000010：TS2 通道 …… 011111：TS31 通道 100000：TS32 通道 …… 100101：TS35 通道 others：无效
15:0	RAW_COUNT	TSI 模块输出 raw count，有效位宽由 Resolution 寄存器决定，最大 16bit，最小 8bit

31.12.12 时钟和周期控制寄存器（TSI_CCR）

名称	TSI_CCR							
offset	0x2C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-	SCAN_ZERO						
位权限	U-0	R/W-0000 000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SMPL_IDLE							
位权限	R/W-0000_0000							

位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SCAN_INTV							
位权限	R/W-0000_0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SMPL_WAIT			PRSCLK_EN
位权限	U-0				R/W-000			R/W-0

Bit	助记符	功能描述
31	--	RFU：未实现，读为 0
30:24	SCAN_ZERO	Auto zero 使能时间配置，计数时钟为 APBCLK，共 32*(SCAN_ZERO+1)个周期
23:16	SMPL_IDLE	每个扫描周期启动后，首次扫描前的准备时间配置，计数时钟为 APBCLK，共 32*(SCAN_IDLE +1)个周期
15:8	SCAN_INTV	连续两个 sensor 扫描之间的采样间隔配置，计数时钟为 APBCLK，共 32*(SCAN_INTV+1)个周期
7:4	--	RFU：未实现，读为 0
3:1	SMPL_WAIT	采样等待配置，计数时钟为工作时钟，扫描准备时间结束后等待（2^SMPL_WAIT）个计数时钟后开始采样
0	PRSCLK_EN	伪随机时钟序列使能 0：FSW 采用固定频率 1：FSW 采用伪随机频率，有 PRS_SW 寄存器控制

31.12.13 DMA 控制寄存器（TSI_DMACR）

名称	TSI_DMACR							
offset	0x30							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						DMA_WREN	DMA_RDEN
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	DMA_WREN	DMA 写配置寄存器使能，1 有效
0	DMA_RDEN	DMA 读 raw count 使能，1 有效

31.12.14 按键通道扫描参数配置寄存器（TSI_SP_CFGR）

名称	TSI_SP_CFGR							
offset	0x34							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SNS_DIV[11:4]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SNS_DIV[3:0]				RES			
位权限	R/W-0000				R/W-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDAC_MODE	IDAC2_CFG						
位权限	R/W-0	R/W-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	IDAC1_CFG						
位权限	U-0	R/W-0000000						

Bit	助记符	功能描述
31:20	SNS_DIV[11:0]	sense clock 分频系数，共 12bit 此寄存器用于配置 sense clock 分频比，支持 1~4096 分频
19:16	RES	通道 x 扫描精度配置（N 值） 0000：8 位 0001：9 位 … 0111：15 位 禁止选择其他档位
15	IDAC_MODE	通道 x 电流源模式配置 0：single IDAC 1：dual IDAC
14:8	IDAC2_CFG	通道 x 的 IDAC2 电流配置
7	--	RFU：未实现，读为 0
6:0	IDAC1_CFG	通道 x 的 IDAC1 电流配置

注意：这个寄存器在每次当前通道扫描完成、下一通道开始扫描之前，由软件或者DMA更新参数；连续扫描时不更新参数，所有使能通道使用相同参数。

31.12.15 IDAC 调校寄存器（TSI_IDAC_TR）

名称	TSI_IDAC_TR							
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	-			IDAC2_TRIM				
位权限	U-0			R/W-10000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			IDAC1_TRIM				
位权限	U-0			R/W-10000				

Bit	助记符	功能描述
31:13	--	RFU：未实现，读为 0
12:8	IDAC2_TRIM	IDAC2 的校准值
7:5	--	RFU：未实现，读为 0
4:0	IDAC1_TRIM	IDAC1 的校准值

31.12.16 IDAC 模拟控制寄存器（TSI_ANA_CR）

名称	TSI_ANA_CR							
offset	0x3C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	HSCMP_RC_CFG				HSCMP_RC_BYP	-	LOWREF_SEL	
位权限	R/W-0000				R/W-1	U-0	R/W-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CMP_AZ_BYP	HSCMP_AZ_BYP	LES_PCFG		LES_NCFG		IDAC2_MODE_CTRL	IDAC1_MODE_CTRL
位权限	R/W-0	R/W-1	R/W-00		R/W-00		R/W-1	R/W-1
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IDAC2_DOUBLE	IDAC1_DOUBLE	IDAC2_RANGE		IDAC1_RANGE		SHLD_MODE	
位权限	R/W-0	R/W-0	R/W-10		R/W-10		R/W-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RC_BYP	CMP_RC				CMPHS	VREFBUF_CFG	
位权限	R/W-0	R/W-0010				R/W-0	R/W-10	

Bit	助记符	功能描述
31:28	HSCMP_RC_CFG	HS 比较器 RC 滤波配置信号
27	HSCMP_RC_BYP	HS 比较器 RC 滤波 bypass 使能信号
26	--	RFU：未实现，读为 0
25:24	LOWREF_SEL	低压基准选择信号
23	CMP_AZ_BYP	比较器 AutoZero bypass
22	HSCMP_AZ_BYP	高速比较器 AutoZero bypass
21:20	LES_PCFG	Low EMI Switch PMOS 配置
19:18	LES_NCFG	LowEMISwitchNMOS 配置
17	IDAC2_MODE_CTRL	IDAC2 模式配置：自动模式需配置为 1，互容模式建议配置为 0
16	IDAC1_MODE_CTRL	IDAC1 模式配置：自动模式需配置为 1，互容模式建议配置为 1
15	IDAC2_DOUBLE	IDAC2 电流翻倍模式

Bit	助记符	功能描述
		0：不翻倍 1：翻倍
14	IDAC1_DOUBLE	IDAC1 电流翻倍模式 0：不翻倍 1：翻倍
13:12	IDAC2_RANGE	IDAC2 档位配置 00: 37.5nA 01: 300nA 10: 1.2uA 11: 2.4uA
11:10	IDAC1_RANGE	IDAC1 档位配置 00: 37.5nA 01: 300nA 10: 1.2uA 11: 2.4uA
9:8	SHLD_MODE	Shield 模式： 00，11:454 模式 01：内部 buffer 模式 10：外部电容模式
7	RC_BYP	比较器输入端 RC 滤波 bypass 使能，为 1 时 bypass 滤波网络
6:3	CMP_RC	比较器输入端 RC 滤波带宽配置
2	CMPHS	比较器高速模式使能 0：不使能； 1：使能，偏置电流翻倍
1:0	VREFBUF_CFG	Reference buffer 的充电电流配置 00：不使能 01：弱驱动能力 10：中等驱动能力 11：强驱动能力

31.12.17 TSC_DFT 寄存器（TSI_TEST）

名称	TSI_TEST							
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	Sense_cfg				Ch_idle_idle	-	R2D_cfg	
位权限	R/W-0000				R/W-00	U-0	R/W-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	Sense_valid	-					
位权限	U-0	R/W-0	U-000000					
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	Sense_cfg	灵敏度测试档位配置，仅灵敏度测试时使用。正常扫描时保持全0.setp:0.1pF 0000:无效 0001:0.1pF 0010:0.2pF … 1111:1.5pF
19	--	RFU：未实现，读为 0
18	Ch_idle_cfg	被使能的接收通道，不活跃时的连接配置 0：浮空 1：强制接地
17:16	R2d_cfg	接地大电阻的配置 00:100kΩ 01:150kΩ 10:200kΩ 11:250kΩ
15	--	RFU：未实现，读为 0
14	Sensi_valid	Test_ring 的连接配置 0：test_ring 接大电阻 1：test_ring 接电容阵列
13:0	--	RFU：未实现，读为 0

31.12.18 测试电容通道使能寄存器（TSI_TESTRING0）

名称	TSI_TESTRING0							
offset	0x44							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TESTRING_EN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TESTRING_EN [23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TESTRING_EN [15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TESTRING_EN [7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	TESTRING_EN	各个通道的连接 test_ring 使能 bit0: TS0 bit1: TS1 … bit31: TS31

31.12.19 测试电容通道使能寄存器（TSI_TESTRING1）

名称	TSI_TESTRING1							
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	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				TESTRING_EN [35:32]			
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	TESTRING_EN	各个通道的连接 test_ring 使能 bit0：TS32 bit1：TS33 bit2：TS34 bit3：TS35

31.12.20 ANA TEST 寄存器（TSI_ANA_TEST）

名称	TSI_ANA_TEST							
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	Vcomp_dflt							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Vcomp_dfen	Idac1_chg_swt	Excap_sel		Excap_testen	Ibias_outen	Vcomp_fb	Idac_comp_mode
位权限	R/W-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	Vref_buf_ent	Comp_ent	Idac2_ent	Idac1_ent	Sigma_delta_ent	Debug_mode
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

Bit	助记符	功能描述
31:24	--	RFU：未实现，读为 0
23:16	Vcomp_dflt	比较器反馈数字滤波宽度配置，数字对 vcomp 连续采样 dflt

Bit	助记符	功能描述
		个 pll_vco 周期，结果一致才反馈给模拟 0 表示不做采样 1 表示采样 1 次 2 表示连续采样 2 次，相同电平才改变输出 ……
15	Vcomp_dfen	比较器反馈数字滤波使能
14	Idac1_chg_swt	仅在外部电容检测时使用，控制 IDAC1 的开关使能，置位后开始对 ABUS 充电
13:12	Excap_sel	外部电容检测选择 00：检测 CINTA 01：检测 CINTB 10：检测 CTANK 11：检测 CMOD
11	Excap_testen	外部电容检测使能，仅 debug_mode=1 时有效
10	Ibias_outen	IDAC1 的电流测试输出 0：不输出 1：输出
9	Vcomp_fb	比较器输出使用 Fmod 采样后反馈给模拟时，选择采样时钟沿 0：上升沿 1：下降沿
8	Idac_comp_mode	IDAC2 的补偿模式控制，非 debug 模式下起作用。在 debug 模式下，idac1_chg_sw 信号为比较器输出信号和 Idac1_ent 信号的与，idac2_chg_sw 信号为比较器输出信号和 Idac2_ent 信号的与 0：常开模式（非 debug 模式下，idac1_chg_sw 信号为比较器输出信号，idac2_chg_sw 信号一直为 1） 1：互补模式（非 debug 模式下，idac1_chg_sw 信号为比较器输出信号，idac2_chg_sw 信号为比较器输出信号的反）
7	--	RFU：未实现，读为 0
6	--	RFU：未实现，读为 0
5	Vref_buf_ent	0：debug 模式下 vref_buf_en、vref_buf_cona、vref_buf_conb 受内部状态机控制 1：debug 模式下 vref_buf_en 为 1，vref_buf_cona 为 1，vref_buf_conb 为 0，cmod_cona 信号为 1，cmod_conb 信号为 0.
4	Comp_ent	0：debug 模式下 comp_en 受模块内部状态机控制 1：debug 模式下 comp_en 为 1
3	Idac2_ent	0：debug 模式下 idac2_en 为 0 1：debug 模式下 idac2_en 为 1
2	Idac1_ent	0：debug 模式下 idac1_en 为 0 1：debug 模式下 idac1_en 为 1
1	Sigma_delta_ent	0：debug 模式下 delta_bias_en 受模块内部状态机控制 1：debug 模式下 delta_bias_en 为 1
0	Debug_mode	Debug 模式控制器，会影响 idac1_en、idac2_en、idac1_chg_sw、idac2_chg_sw、vref_buf_en、vref_buf_cona、vref_buf_conb、cmod_cona、cmod_conb、comp_en、delta_bias_en、s0_swen 信号。 0：所有信号受模块内部状态机控制 1：所有信号受特殊控制，详见上述各个 bit 的描述。

31.12.21 VREF 基准配置寄存器（TSI_VREFCR_TEST）

名称	TSI_VREFCR_TEST							
offset	0x50							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-			ref_sel		shield_bypbuf	vref_bypbuf	csd_bypbuf
位权限	U-000			R/W-00		R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-	vref_csdcmp_trim						
位权限	U-0	R/W-1000000						
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	vref_shield_trim						
位权限	U-0	R/W-1000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	vref_buf_trim						
位权限	U-0	R/W-1000000						

Bit	助记符	功能描述
31:29	--	RFU：未实现，读为 0
28:27	ref_sel	基准档位选择信号
26	shield_bypbuf	shield 基准 buffer bypass 使能
25	vref_bypbuf	vref 基准 buffer bypass 使能
24	csd_bypbuf	csd 基准 buffer bypass 使能
23	--	RFU：未实现，读为 0
22:16	vref_csdcmp_trim	csd 基准 buffer 的 trim 配置
15	--	RFU：未实现，读为 0
14:8	vref_shield_trim	shield 基准 buffer 的 trim 配置
7	--	RFU：未实现，读为 0
6:0	vref_buf_trim	vref 基准 buffer 的 trim 配置

31.12.22 屏蔽通道增强寄存器（TSI_SHLDGND_CR0）

名称	TSI_SHLDGND_CR0							
offset	0x58							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SHLD_CONGND_EN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SHLD_CONGND_EN[23:16]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SHLD_CONGND_EN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SHLD_CONGND_EN[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述

Bit	助记符	功能描述
31:0	SHLD_CONGND_EN	shield 通道接地时通过 pad 输出 0 使能寄存器，写 1 使能，清零禁止 bit0: TS0 bit1: TS1 … bit31: TS31 通过 Ch_idle_shield 寄存器可配置 shield 通道。

31.12.23 屏蔽通道增强寄存器（TSI_SHLDGND_CR1）

名称	TSI_SHLDGND_CR1							
offset	0x5C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SHLD_CONGND_EN[35:32]			
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	SHLD_CONGND_EN	shield 通道接地时通过 pad 输出 0 使能寄存器，写 1 使能，清零禁止 bit0: TS32 bit1: TS33 bit2: TS34 bit3: TS35 通过 Ch_idle_shield 寄存器可配置 shield 通道。

31.12.24 低辐射驱动配置寄存器（TSI_EMI_CFGR0）

名称	TSI_EMI_CFGR0							
offset	0x60							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SR_CFG0							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SR_CFG0							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SR_CFG0							

位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SR_CFG0							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	SR_CFG0	TSxx slew rate control，每 2bit 对应一个 pad，从低到高对应 PE6~PE9，PF12~PF15.高 16 位为 dummy 寄存器。 00: default 01: slow 10: RFU 11: RFU

31.12.25 低辐射驱动配置寄存器（TSI_EMI_CFGR1）

名称	TSI_EMI_CFGR1							
offset	0x64							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SR_CFG1							
位权限	R/W-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SR_CFG1							
位权限	R/W-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SR_CFG1							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SR_CFG1							
位权限	R/W-0							

Bit	助记符	功能描述
31:0	SR_CFG1	TSxx slew rate control，每 2bit 对应一个 pad，从低到高对应 PC0~PC14.高 2 位为 dummy 寄存器。 00: default 01: slow 10: RFU 11: RFU

31.12.26 屏蔽通道使能寄存器（TSI_SHLD_CR0）

名称	TSI_SHLD_CR0							
offset	0x68							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SHLDEN[31:24]							
位权限	R/W-0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SHLDEN[23:16]							
位权限	R/W-0000 0000							

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

Bit	助记符	功能描述
31:0	SHLDEN	shield 使能寄存器，写 1 使能，清零禁止 bit0: TS0 bit1: TS1 … bit31: TS31 置位的 bit 对应的 TS 通道，在自容模式下作为 shield 通道使用，通过 Ch_idle_shield 寄存器可配置 shield 通道

31.12.27 屏蔽通道使能寄存器（TSI_SHLD_CR1）

名称	TSI_SHLD_CR1							
offset	0x6C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				SHLDEN[35:32]			
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	SHLDEN	shield 使能寄存器，写 1 使能，清零禁止 bit0: TS32 bit1: TS33 bit2: TS34 bit3: TS35 置位的 bit 对应的 TS 通道在自容模式下作为 shield 通道使用，通过 Ch_idle_shield 寄存器可配置 shield 通道

31.12.28 时钟管理寄存器（TSI_CKCR）

名称	TSI_CKCR							
offset	0x70							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MOD_PRSC							
位权限	R/W-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		SSC_FPT		-	SSCEN	-	SNSSEL
位权限	U-00		R/W-00		U-0	R/W-0	U-0	R/W-0

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:8	MOD_PRSC	mod clock 时钟预分频控制，分频系数为 MOD_PRSC+1
7:6	--	RFU：未实现，读为 0
5:4	SSC_FPT	扩频频点数量选择，调制多项式位宽为 6 时仅支持 32、64 点，调制多项式位宽为 7 时仅支持 32、64、128 点。 00：32 点 01：64 点 10：128 点 11：256 点
3	--	RFU：未实现，读为 0
2	SSCEN	扩频时钟使能 0：关闭扩频 1：使能扩频
1	--	RFU：未实现，读为 0
0	SNSSEL	sense clock 时钟源选择 0：Fmod 1：PLL_TSI 输出的 VCO 频率

31.12.29 自容伪随机生成多项式寄存器（TSI_PRS_SW）

名称	TSI_PRS_SW							
offset	0x74							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				PRS_SW_WIDTH			
位权限	U-0				R/W-1000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			PRS_SW				

位权限	U-000			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRS_SW							
位权限	R/W-10001110							

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:16	PRS_SW_WIDTH	PRS_SW 生成多项式的位宽，需和多项式抽头系数匹配
15:13	--	RFU：未实现，读为 0
12:0	PRS_SW	PRS_SW 的生成多项式抽头系数，推荐多项式： width=8,prs_sw=0x8e width=12,prs_sw=0x829 width=13,prs_sw=0x1ab1

31.12.30 PLL 控制寄存器（TSI_PLL_CR）

名称	TSI_PLL_CR							
offset	0x78							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PLL_RTRIM					-	PLLDB	
位权限	R/W-10000					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	PLL_ICFG				-			LOCKED
位权限	R/W-0011				U-000			R/Dy-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REFPRSC							PLLEN
位权限	R/W-00 0000							R/W-0

Bit	助记符	功能描述
31:27	PLL_RTRIM	PLL 电阻配置
26	--	RFU：未实现，读为 0
25:16	PLLDB	PLL 倍频比，输出时钟频率为 2M*(PLLDB+1) 0x1F：输出 32 倍频 0x2F：输出 48 倍频 0x3F：输出 64 倍频 0x47：输出 72 倍频
15:12	PLL_ICFG	PLL 电流配置
11:9	--	RFU：未实现，读为 0
8	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
1:1	REFPRSC	PLL 参考时钟预分频（目标是产生 1MHz 参考时钟给 PLL），分频系数为 REFPRSC+1
0	PLLEN	PLL 使能寄存器

Bit	助记符	功能描述
		1：使能 PLL 0：关闭 PLL

31.12.31 互容 SSC 生成多项式寄存器（TSI_PRS_SSC）

名称	TSI_PRS_SSC							
offset	0x7C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-				PRS_SSC_WIDTH			
位权限	U-0				R/W-1000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			PRS_SSC				
位权限	U-000			R/W-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRS_SSC							
位权限	R/W-10001110							

Bit	助记符	功能描述
31:20	--	RFU：未实现，读为 0
19:16	PRS_SSC_WIDTH	PRS_SW 生成多项式的位宽，需和多项式抽头系数匹配
15:13	--	RFU：未实现，读为 0
12:0	PRS_SSC	PRS_SW 的生成多项式抽头系数，推荐多项式： width=6,prs_sw=0x21 width=7,prs_sw=0x6A width=9,prs_sw=0x108 width=10,prs_sw=0x204

32 I/O 端口（GPIO）

32.1 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）

32.2 引脚类型

FM33FH0 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；

32.2.1 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

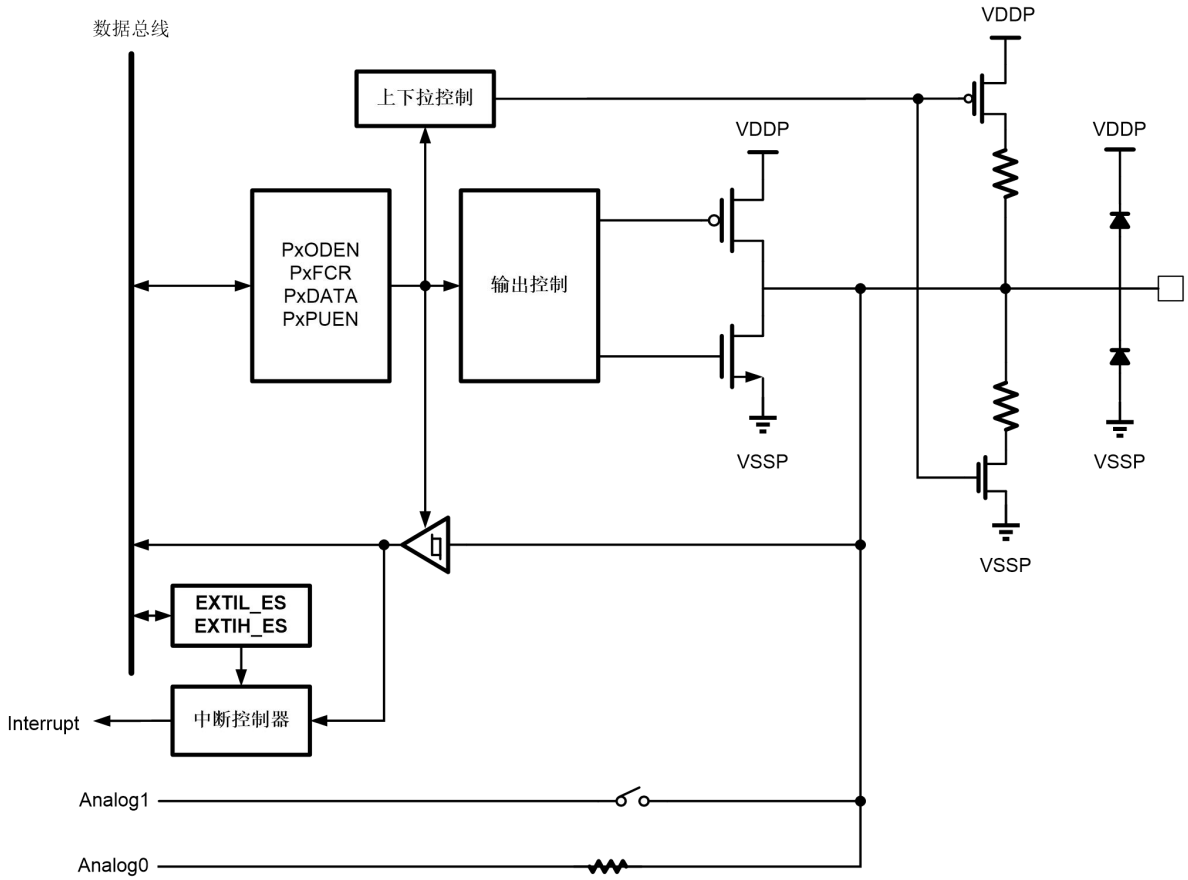


图 32-1 普通 GPIO 结构框图

控制逻辑定义如下：

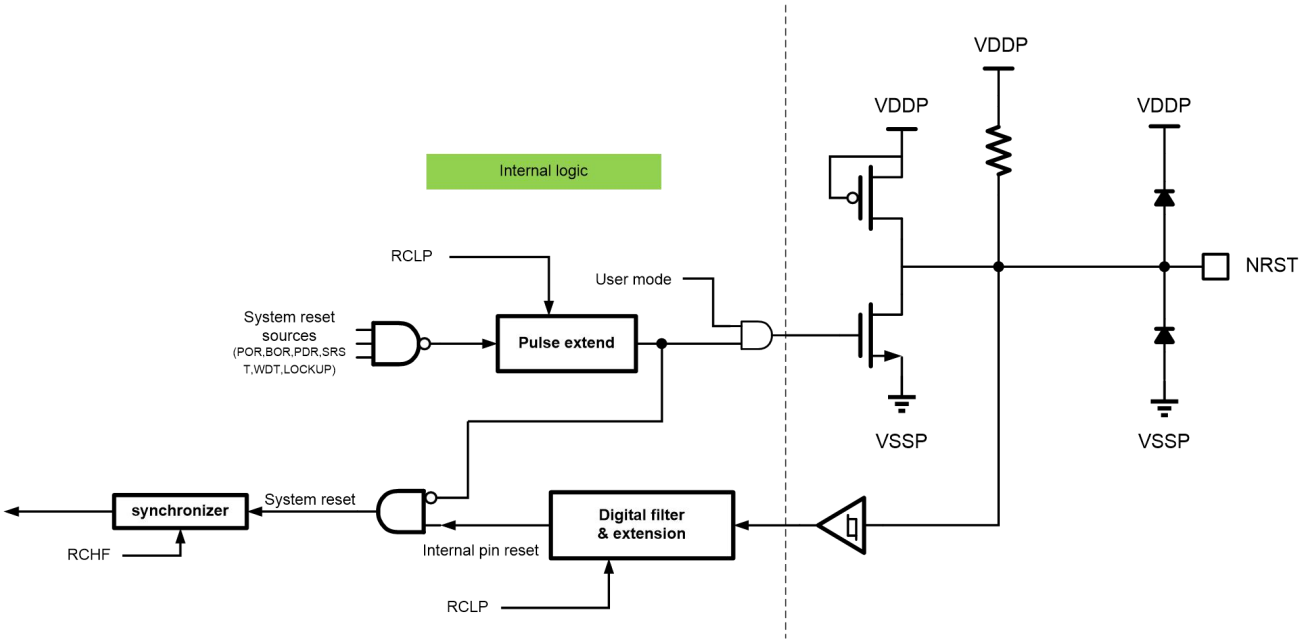
Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	外设输入功能	1	0	0/1
	x	0		外设推挽输出功能	0	1	
	x	1		外设开漏输出 0	0	1	
				外设开漏输出 1	0	0	
11	x	x	x	x	0	0	0

表 32-1GPIO 功能逻辑定义表

上下拉电阻各50K欧姆

驱动能力参考3.4.15章节的参数。

32.2.2 NRST，固定上拉，NMOS 输出驱动



32.3 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 32-2FCR 定义表

32.3.1 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

32.3.2 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

32.3.3 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。

32.3.4 模拟功能

当某个 GPIO 被配置成模拟功能：

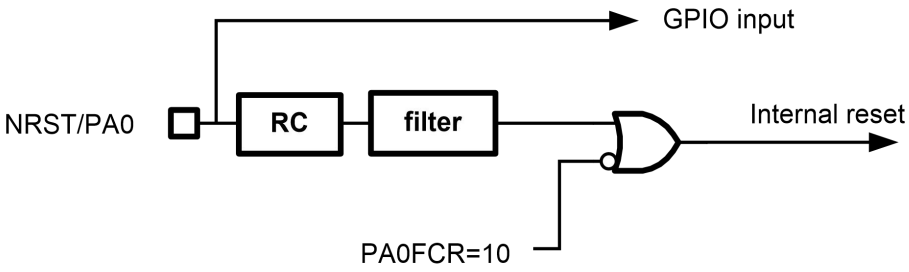
- 输出缓冲器关闭
- 数字输入功能关闭
- 上拉电阻关闭
- 软件读取 PxDIN 返回 0
- IO 模拟通道被连接到特定的模拟外设上
- 如果一个 IO 同时连接到多个模拟外设，则多个模拟外设在同一时刻只能使能其中一个

32.4 NRST 引脚与 GPIO 复用

NRST 为全局复位引脚，带有数字滤波，滤波采用 RCLP 时钟工作，RCLP 保持常开，无法关闭。

为了提高管脚利用率，用户模式下的 NRST 引脚也可以复用为 GPIO(PA0)。

PA0FCR 寄存器复位值为 10，即复位默认该引脚为 NRST 功能，并且 PAPUEN0 复位值为 1，即默认上拉使能。软件可以通过将 PA0FCR 修改为 00 或 01 来使能其 GPIO 功能，此时 NRST 复位功能无效。PA0 作为 GPIO 时只有 50K 上拉电阻，无下拉电阻，作为 NRST 时固定为 10K 上拉电阻。



NRST 数字滤波

数字滤波采用 RCLP 连续采样 64 拍，以及输入信号高电平异步复位的方式，即确保 64 个 RCLP 周期内 NRST 信号保持稳定的低电平。由于输入信号与 RCLP 为异步关系，实际有效滤波长度可能是 63~64 个 RCLP 周期之间，即 1.95ms 的典型范围，随着温度变化可能会有 10%左右的变化。

32.5 SWD 引脚

ARM SWD 引脚复用 PG9(SWCLK)和 PG8(SWIO)，这两个 GPIO 上电复位后默认为 SWD 功能，并且默认使能内部上拉电阻(50K)，以省去外部上拉。

32.6 PA14/PA15 引脚

PA14/PA15 上电复位后默认为输入态、上拉使能，使用时需注意。

32.7 WKUPx 引脚

FM33FH0 有 8 个 WKUP 引脚，能够将芯片从 Sleep 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 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。

由于原来的方案中，当一路 WKUPx 保持有效电平后，会屏蔽其他 WKUPx 引脚的边沿检测，所以考虑将所有 WKUPx 的模拟滤波和边沿检测分开实现。

8 路 WKUPx 电路结构完全独立，下图显示了其中一路 WKUP 功能的结构框图

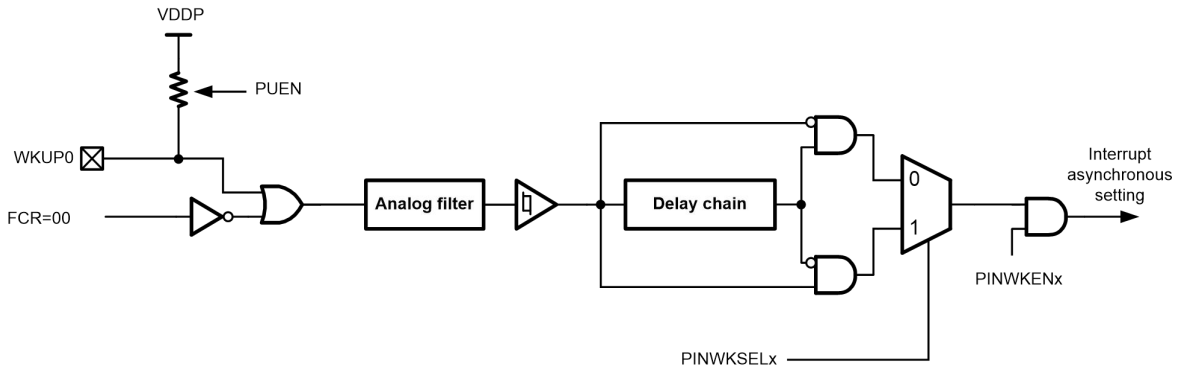


图 32-2WKUPx 功能结构框图

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。

32.8 外部引脚中断（EXTI）

32.8.1 功能说明

芯片每个引脚都可以独立产生 EXTI 中断，最终所有的 EXTI 中断汇总到 NVIC 的#46 入口。

总共可以产生多达 93 个引脚中断事件。软件可以单独配置每个管脚的中断事件为上升沿、下降沿、上升下降沿、或不产生中断。

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。

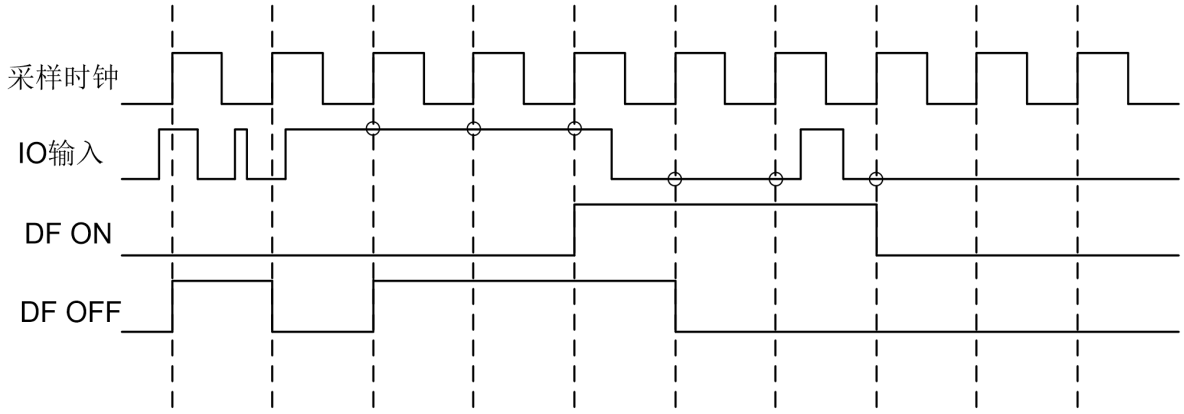


图 32-3 引脚输入数字滤波

当软件可以选择数字滤波的采样时钟为 APBCLK 或者 LSCLK。

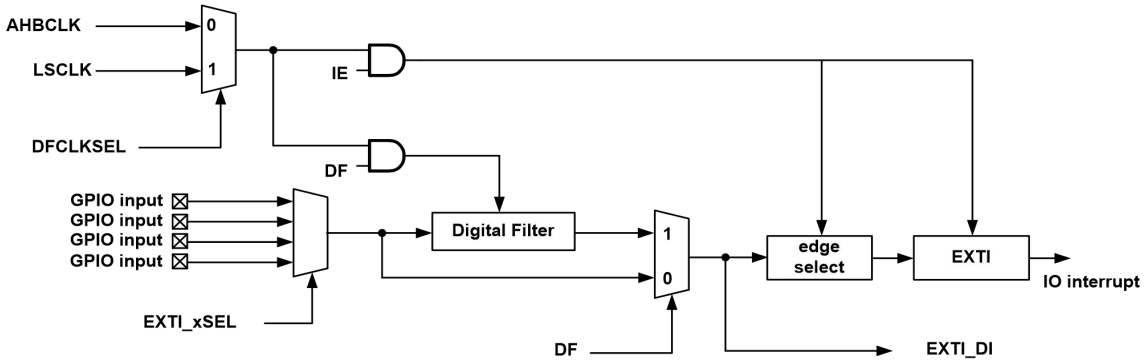


图 32-4 EXTI 信号输入示意图

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。

32.8.2 应用指南

如需在 Sleep 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（0x4000020C）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 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 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 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

32.9 快速 GPIO 输出

FM33FH0 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（atomic）。方法是每个 GPIO 组的输出数据寄存器都有 2 组 set-reset 映射虚拟地址，对 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。

32.10 寄存器

GPIO模块基地址：0x4000_0C00

offset 地址	名称	符号
0x00	GPIOA 输入使能寄存器（GPIOA Input Enable Register）	GPIOA_INEN
0x04	GPIOA 上下拉使能寄存器（GPIOA Pull-Up Pull-Down Enable Register）	GPIOA_PUDEN
0x08	GPIOA 开漏使能寄存器（GPIOA Open-Drain Enable Register）	GPIOA_ODEN
0x0C	GPIOA 功能选择寄存器（GPIOA Function Control Register）	GPIOA_FCR
0x10	GPIOA 输出数据寄存器（GPIOA Data Output Register）	GPIOA_DO
0x14	GPIOA 输出数据置位寄存器（GPIOA Data Set Register）	GPIOA_DSET
0x18	GPIOA 输出数据复位寄存器（GPIOA Data Reset Register）	GPIOA_DRST
0x1C	GPIOA 输入数据寄存器（GPIOA Data Input Register）	GPIOA_DIN
0x20	GPIOA 额外数字功能寄存器（GPIOA Digital Function Select）	GPIOA_DFS
0x24		
0x28		
0x2C	GPIOA 驱动能力配置寄存器（GPIOA Driver Strength Register）	GPIOA_DSR
0x40	GPIOB 输入使能寄存器（GPIOB Input Enable Register）	GPIOB_INEN
0x44	GPIOB 上拉使能寄存器（GPIOB Pull-Up Enable Register）	GPIOB_PUEN
0x48	GPIOB 开漏使能寄存器（GPIOB Opeb-Drain Enable Register）	GPIOB_ODEN
0x4C	GPIOB 功能选择寄存器（GPIOB Function Control Register）	GPIOB_FCR
0x50	GPIOB 输出数据寄存器	GPIOB_DO

offset 地址	名称	符号
	（GPIOB Data Output Register）	
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 Driver Strength Register）	GPIOB_DSR
0x80	GPIOC 输入使能寄存器 （GPIOC Input Enable Register）	GPIOC_INEN
0x84	GPIOC 上拉使能寄存器 （GPIOC Pull-Up Enable Register）	GPIOC_PUEN
0x88	GPIOC 开漏使能寄存器 （GPIOC Opeb-Drain Enable Register）	GPIOC_ODEN
0x8C	GPIOC 功能选择寄存器 （GPIOC Function Control Register）	GPIOC_FCR
0x90	GPIOC 输出数据寄存器 （GPIOC Data Output Register）	GPIOC_DO
0x94	GPIOC 输出数据置位寄存器 （GPIOC Data Set Register）	GPIOC_DSET
0x98	GPIOC 输出数据复位寄存器 （GPIOC Data Reset Register）	GPIOC_DRST
0x9C	GPIOC 输入数据寄存器 （GPIOC Data Input Register）	GPIOC_DIN
0xA0	GPIOC 额外数字功能寄存器 （GPIOC Digital Function Select）	GPIOC_DFS
0xA4	-	-
0xA8		
0xAC	GPIOC 驱动能力配置寄存器 （GPIOC Driver Strength Register）	GPIOC_DSR
0xB0	GPIOC 定时器通道选择寄存器 （GPIOC Timer Mux Register）	GPIOC_TMUXR
0xC0	GPIOD 输入使能寄存器 （GPIOD Input Enable Register）	GPIOD_INEN
0xC4	GPIOD 上拉使能寄存器 （GPIOD Pull-Up Enable Register）	GPIOD_PUEN
0xC8	GPIOD 开漏使能寄存器 （GPIOD Opeb-Drain Enable Register）	GPIOD_ODEN
0xCC	GPIOD 功能选择寄存器 （GPIOD Function Control Register）	GPIOD_FCR
0xD0	GPIOD 输出数据寄存器 （GPIOD Data Output Register）	GPIOD_DO

offset 地址	名称	符号
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
0xF0	GPIOE 输入使能寄存器 （GPIOE Input Enable Register）	GPIOE_INEN
0xF4	GPIOE 上拉使能寄存器 （GPIOE Pull-Up Enable Register）	GPIOE_PUEN
0xF8	GPIOE 开漏使能寄存器 （GPIOE Opeb-Drain Enable Register）	GPIOE_ODEN
0xFC	GPIOE 功能选择寄存器 （GPIOE Function Control Register）	GPIOE_FCR
0x100	GPIOE 输出数据寄存器 （GPIOE Data Output Register）	GPIOE_DO
0x104	GPIOE 输出数据置位寄存器 （GPIOE Data Set Register）	GPIOE_DSET
0x108	GPIOE 输出数据复位寄存器 （GPIOE Data Reset Register）	GPIOE_DRST
0x10C	GPIOE 输入数据寄存器 （GPIOE Data Input Register）	GPIOE_DIN
0x110	GPIOE 额外数字功能寄存器 （GPIOE Digital Function Select）	GPIOE_DFS
0x114	-	-
0x118		
0x11C	GPIOE 驱动能力配置寄存器 （GPIOE Driver Strength Register）	GPIOE_DSR
0x120	GPIOF 输入使能寄存器 （GPIOF Input Enable Register）	GPIOF_INEN
0x124	GPIOF 上拉使能寄存器 （GPIOF Pull-Up Enable Register）	GPIOF_PUEN
0x128	GPIOF 开漏使能寄存器 （GPIOF Opeb-Drain Enable Register）	GPIOF_ODEN
0x12C	GPIOF 功能选择寄存器 （GPIOF Function Control Register）	GPIOF_FCR
0x130	GPIOF 输出数据寄存器 （GPIOF Data Output Register）	GPIOF_DO
0x134	GPIOF 输出数据置位寄存器 （GPIOF Data Set Register）	GPIOF_DSET

offset 地址	名称	符号
0x138	GPIOF 输出数据复位寄存器 （GPIOF Data Reset Register）	GPIOF_DRST
0x13C	GPIOF 输入数据寄存器 （GPIOF Data Input Register）	GPIOF_DIN
0x140	GPIOF 额外数字功能寄存器 （GPIOF Digital Function Select）	GPIOF_DFS
0x144	-	-
0x148		
0x14C	GPIOF 驱动能力配置寄存器 （GPIOF Driver Strength Register）	GPIOF_DSR
0x150	GPIOG 输入使能寄存器 （GPIOG Input Enable Register）	GPIOG_INEN
0x154	GPIOG 上拉使能寄存器 （GPIOG Pull-Up Enable Register）	GPIOG_PUEN
0x158	GPIOG 开漏使能寄存器 （GPIOG Opeb-Drain Enable Register）	GPIOG_ODEN
0x15C	GPIOG 功能选择寄存器 （GPIOG Function Control Register）	GPIOG_FCR
0x160	GPIOG 输出数据寄存器 （GPIOG Data Output Register）	GPIOG_DO
0x164	GPIOG 输出数据置位寄存器 （GPIOG Data Set Register）	GPIOG_DSET
0x168	GPIOG 输出数据复位寄存器 （GPIOG Data Reset Register）	GPIOG_DRST
0x16C	GPIOG 输入数据寄存器 （GPIOG Data Input Register）	GPIOG_DIN
0x170	GPIOG 额外数字功能寄存器 （GPIOG Digital Function Select）	GPIOG_DFS
0x174	-	-
0x178		
0x17C	GPIOG 驱动能力配置寄存器 （GPIOG Driver Strength Register）	GPIOG_DSR
0x180	GPIOH 输入使能寄存器 （GPIOH Input Enable Register）	GPIOH_INEN
0x184	GPIOH 上拉使能寄存器 （GPIOH Pull-Up Enable Register）	GPIOH_PUEN
0x188	GPIOH 开漏使能寄存器 （GPIOH Opeb-Drain Enable Register）	GPIOH_ODEN
0x18C	GPIOH 功能选择寄存器 （GPIOH Function Control Register）	GPIOH_FCR
0x190	GPIOH 输出数据寄存器 （GPIOH Data Output Register）	GPIOH_DO
0x194	GPIOH 输出数据置位寄存器 （GPIOH Data Set Register）	GPIOH_DSET
0x198	GPIOH 输出数据复位寄存器 （GPIOH Data Reset Register）	GPIOH_DRST
0x19C	GPIOH 输入数据寄存器	GPIOH_DIN

offset 地址	名称	符号
	（GPIOH Data Input Register）	
0x1A0	GPIOH 额外数字功能寄存器 （GPIOH Digital Function Select）	GPIOH_DFS
0x1A4	-	-
0x1A8		
0x1AC	GPIOH 驱动能力配置寄存器 （GPIOH Driver Strength Register）	GPIOH_DSR
0x1C0	EXTI 边沿选择和使能寄存器 A （External Interrupt Edge Select and Enable RegisterA）	GPIO_EXTIEDSA
0x1C4	EXTI 边沿选择和使能寄存器 B （External Interrupt Edge Select and Enable RegisterB）	GPIO_EXTIEDSB
0x1C8	EXTI 边沿选择和使能寄存器 C （External Interrupt Edge Select and Enable RegisterC）	GPIO_EXTIEDSC
0x1CC	EXTI 边沿选择和使能寄存器 D （External Interrupt Edge Select and Enable RegisterD）	GPIO_EXTIEDSD
0x1D0	EXTI 边沿选择和使能寄存器 E （External Interrupt Edge Select and Enable RegisterE）	GPIO_EXTIEDSE
0x1D4	EXTI 边沿选择和使能寄存器 F （External Interrupt Edge Select and Enable RegisterF）	GPIO_EXTIEDSF
0x1D8	EXTI 边沿选择和使能寄存器 G （External Interrupt Edge Select and Enable RegisterG）	GPIO_EXTIEDSG
0x1DC	EXTI 边沿选择和使能寄存器 H （External Interrupt Edge Select and Enable RegisterH）	GPIO_EXTIEDSH
0x1E0	EXTI 数字滤波控制寄存器 （External Interrupt Digital Filter Register）	GPIO_EXTIDF
0x1E4	EXTI 中断标志寄存器 AB （External Interrupt and Status Register）	GPIO_EXTIFAB
0x1E8	EXTI 中断标志寄存器 CD （External Interrupt and Status Register）	GPIO_EXTIFCD
0x1EC	EXTI 中断标志寄存器 EF （External Interrupt and Status Register）	GPIO_EXTIFEF
0x1F0	EXTI 中断标志寄存器 GH （External Interrupt and Status Register）	GPIO_EXTIFGH
0x200	FOUT 配置寄存器 （Frequency Output Select Register）	GPIO_FOUTSEL

offset 地址	名称	符号
0x300	WKUP 使能寄存器 （Wakeup Enable Register）	GPIO_PINWKEN

32.10.1 GPIO 输入使能寄存器（GPIOx_INEN）

名称	GPIOx_INEN(x=A~G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 0：关闭输入使能 1：打开输入使能

32.10.2 GPIO 上下拉使能寄存器（GPIOx_PUDEN）

名称	GPIOx_PUDEN (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxPDEN[7:0]							
位权限	R/W-0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxPUEN[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxPUEN[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:16	PxPDEN	GPIO 上下拉选择 0：关闭下拉 1：使能下拉 下拉电阻 50K 欧姆
15:0	PxPUEN	GPIO 上下拉使能 0：关闭上拉 1：使能上拉 上拉电阻 50K 欧姆

32.10.3 GPIO 开漏使能寄存器（GPIOx_ODEN）

名称	GPIOx_ODEN (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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

Bit	助记符	功能描述
		0：关闭开漏输出 1：使能开漏输出

32.10.4 GPIO 功能选择寄存器（GPIOx_FCR）

名称	GPIOx_FCR (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 *注：PD15FCR 复位值为 10（NRST 功能）*
29:28	Px14FCR	Px[14]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
27:26	Px13FCR	Px[13]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function 11：Analog function
25:24	Px12FCR	Px[12]引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：Digital function

Bit	助记符	功能描述
		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
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

Bit	助记符	功能描述
		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

32.10.5 GPIO 输出数据寄存器（GPIOx_DO）

名称	GPIOx_DO (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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

32.10.6 GPIO 输出数据置位寄存器（GPIOx_DSET）

名称	GPIOx_DSET (x=A~ G)

offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 0x14 + y*0x40							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxSET15	PxSET14	PxSET13	PxSET12	PxSET11	PxSET10	PxSET9	PxSET8
位权限	W	W	W	W	W	W	W	W
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxSET7	PxSET6	PxSET5	PxSET4	PxSET3	PxSET2	PxSET1	PxSET0
位权限	W	W	W	W	W	W	W	W

Bit	助记符	功能描述
31:16	--	RFU：未实现，读为 0
15:0	PxSET	GPIO output data set register 举例：向 PADSET 写 0x0000_8000，则 PADO[15]置位，其余位保持不变。

32.10.7 GPIO 输出数据复位寄存器（GPIOx_DRST）

名称	GPIOx_DRST (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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

32.10.8 GPIO 输入数据寄存器（GPIOx_DIN）

名称	GPIOx_DIN (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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

32.10.9 GPIO 额外数字功能选择寄存器（GPIOx_DFS）

名称	GPIOx_DFS (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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

32.10.10 GPIO 驱动强度配置寄存器（GPIOx_DSR）

名称	GPIOx_DSR (x=A~ G)							
offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 PF, y=5 PG, y=6 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 01：驱动最弱 X1

Bit	助记符	功能描述
		10：驱动次强 X3 11：驱动最强 X4 上电后默认所有引脚驱动为次弱（X3）

注：PA0作为GPIO复用输出时，驱动强度固定，需配置为01或11。

32.10.11 EXTI 边沿选择和使能寄存器 A~G（GPIO_EXTIEDSA/B/…/G）

名称	GPIO_EXTIEDSA/B/…/G							
offset	0x240~0x258							
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

32.10.12 EXTI 数字滤波控制寄存器（GPIO_EXTIDF）

名称	GPIO_EXTIDFA/B/…/G							
offset	0x264~0x270							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DFB[15:8]							
位权限	R/W -0000 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_DFB[7:0]							
位权限	R/W -0000 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DFA[15:8]							

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

Bit	助记符	功能描述
31:16	EXTI_DFB	EXTIB 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
15:0	EXTI_DFA	EXTIA 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

32.10.13 EXTI 中断标志寄存器（GPIO_EXTIF）

名称	GPIO_EXTIFA/B/…/G							
offset	0x278~0x284							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI[31:24]							
位权限	R/W-00 0000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI[23:16]							
位权限	R/W-00 0000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI[15:8]							
位权限	R/W-0000 0000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI[7:0]							
位权限	R/W-0000 0000							

Bit	助记符	功能描述
31:0	EXTI_IF	外部引脚中断标志寄存器，共可以产生 32 个引脚中断 硬件置位，软件写 1 清零 高 16bit 为 GPIOB 组中断 低 16bit 为 GPIOA 组中断

32.10.14 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	PB12 输出选择 0000: XTLF 0001: RCLP 0010: ADCCLK 0011: LSCLK 0100: EOC 0101: RTCTM 0110: PLLO/64 0111: RFU 1000: APBCLK/64 1001: PLLO 1010: tsi_vcomp 1011: RCHF 1100: XTHF/64 1101: ADCCLK/64 1110: CLK8K 1111: XTHF
3:0	FOUT0SEL	PD11 输出选择 0000: XTLF 0001: RCLP 0010: RCHF/64 0011: LSCLK 0100: AHBCLK/64 0101: RTCTM 0110: PLLO/64 0111: RTCCLK64Hz 1000: APBCLK/64 1001: PLLO 1010: can_txclk 1011: RCHF 1100: XTHF/64 1101: ADC_SOC 1110: CLK8K 1111: ADC_CLK

32.10.15 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	-	REV1				PINWKSEL	
位权限	R/W-0	U-0	R/W-0000				R/W-00	
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						REV0	
位权限	R/W-0000 0000						R/W-00	
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 39 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	--	RFU：未实现，读为 0
29:26	REV1	DUMMY 寄存器
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：对应的 WKUP 引脚为上升、下降沿都能唤醒 寄存器对应顺序： Bit[11:10]对应 WKUP0 Bit[13:12]对应 WKUP1 Bit[15:14]对应 WKUP2 Bit[17:16]对应 WKUP3 Bit[19:18]对应 WKUP4 Bit[21:20]对应 WKUP5 Bit[23:22]对应 WKUP6 Bit[25:24]对应 WKUP7
9:8	REV0	DUMMY 寄存器
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

33 安全功能（SFU）

33.1 概述

为了对应 IEC60730 和 IEC61508 安全标准，FM33FH0 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
- 代码闪存 ECC，SEC-DED
- 数据闪存 ECC，SEC-DED
- 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
- 比较器自测试
- 温度检测

33.2 闪存 CRC 检查

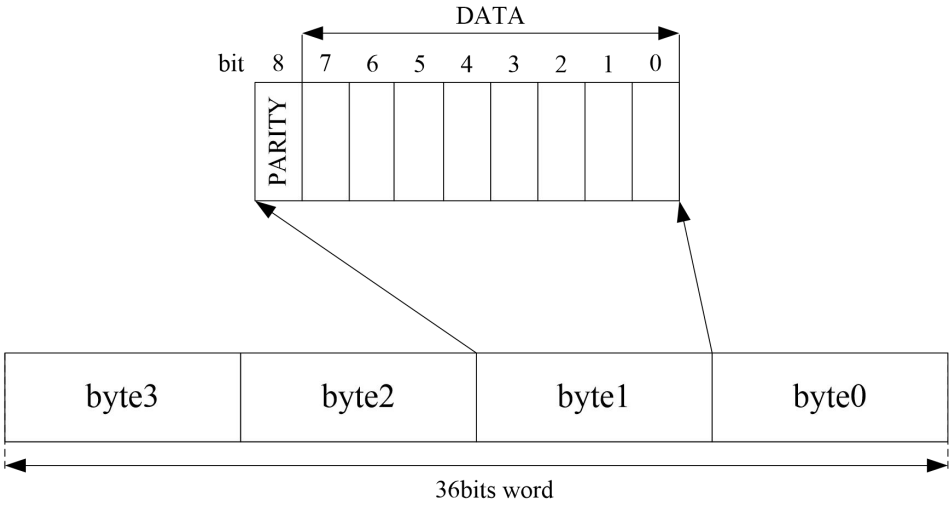
使用 DMA 实现对闪存内容的自检。为了不影响程序执行，启动闪存自检前应在 RAM 中执行代码，然后将 DMA 通道触发配置为 CRC，将源地址指向代码闪存，DMA 启动后，硬件自动将闪存内的数据搬运至 CRC 模块进行校验，完成全部闪存检验后，程序读取 CRC 结果并与预期值比较，以判断闪存内容是否存在问题。闪存检查的长度由 DMA 的 CHxTSIZE 寄存器控制，最大支持 384KB 闪存的全空间校验，用于检查程序完整性。

33.3 RAM 奇偶校验

FM33FH0 的 SRAM 支持硬件奇偶校验，每字节带有 1bit 校验位，以满足 IEC60730 标准要求的 RAM 检验功能。

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

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



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

上电复位后，SRAM 奇偶校验功能默认关闭，软件应完成 RAM 初始化之后再开启校验功能，并且可以配置校验错误中断或复位是否使能。

33.4 RAM 自检

系统 RAM 可以通过 RAMBIST 电路实现自检。

CAN-FD 的 messageRAM 在上电后执行硬件自检，参见 CAN-FD 章节。

33.5 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 4KB RAM 空间，以 1KB 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。

33.6 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
- RAMParity 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_MAPCR）

保护控制寄存器位	受保护模块	受保护寄存器
CMP1P	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
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

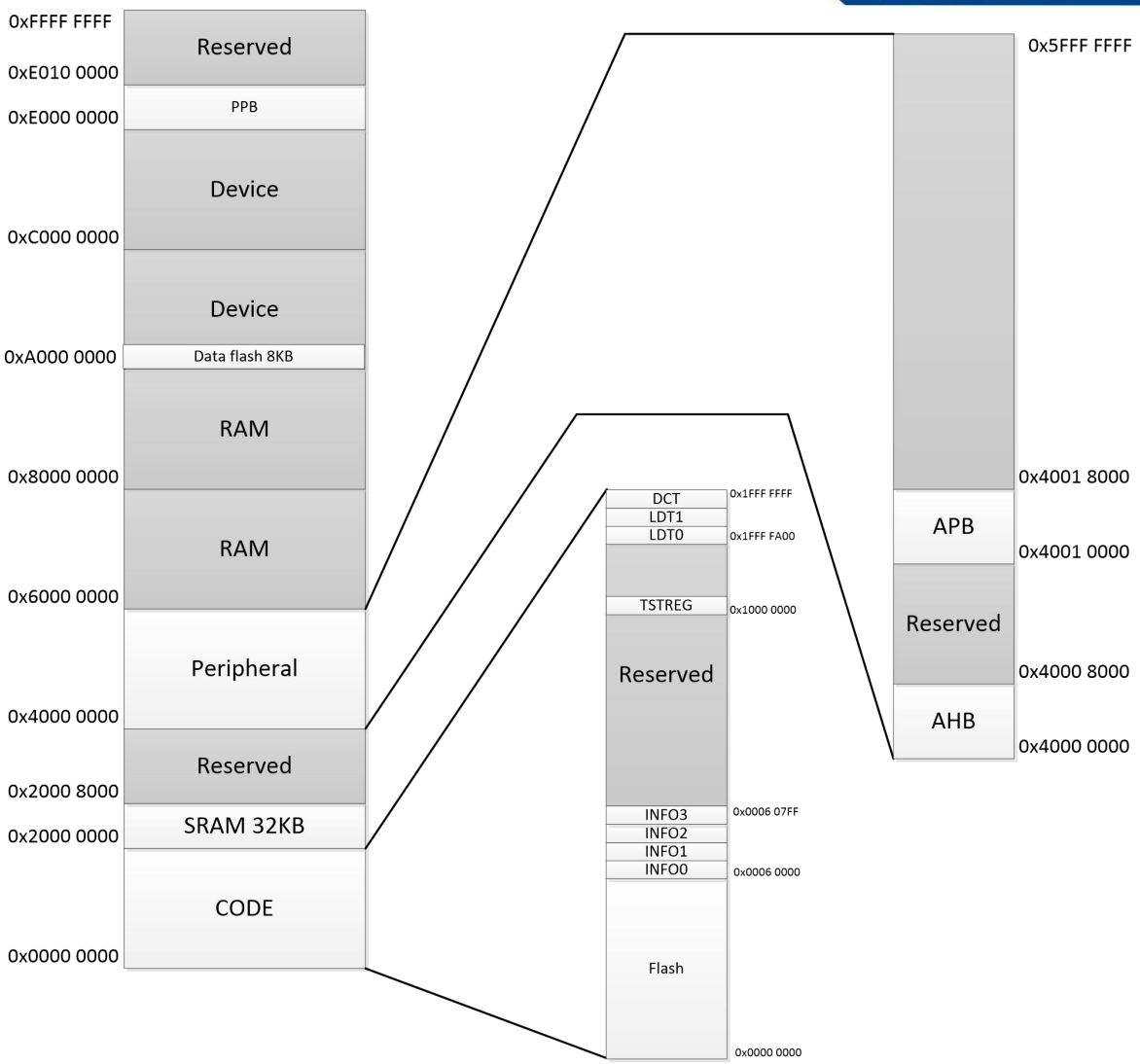
CLM0P	时钟监控模块 0	CLM0_CR CLM0_CFGR CLM0_CMPH CLM0_CMPL
~~CLM1P~~	~~时钟监控模块 1~~	~~CLM1_CR~~ ~~CLM1_CFGR~~ ~~CLM1_CMPH~~ ~~CLM1_CMPL~~
GPIOAP	GPIOA 组	GPIOA_INEN GPIOA_PUDEN GPIOA_ODEN GPIOA_FCR GPIOA_DFS GPIOA_DSR
GPIOBP	GPIOB 组	GPIOB_INEN GPIOB_PUDEN GPIOB_ODEN GPIOB_FCR GPIOB_DFS GPIOB_DSR
GPIOCP	GPIOC 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
GPIOEP	GPIOE 组	GPIOE_INEN GPIOE_PUDEN GPIOE_ODEN GPIOE_FCR GPIOE_DFS GPIOE_DSR
GPIOFP	GPIOF 组	GPIOF_INEN GPIOF_PUDEN GPIOF_ODEN GPIOF_FCR GPIOF_DFS GPIOF_DSR
GPIOGP	GPIOG 组	GPIOG_INEN GPIOG_PUDEN GPIOG_ODEN GPIOG_FCR GPIOG_DFS GPIOG_DSR
GPIOHP	GPIOH 组	GPIOH_INEN GPIOH_PUDEN GPIOH_ODEN GPIOH_FCR GPIOH_DFS GPIOH_DSR
GPIOIP	GPIOI 组	GPIOI_INEN

		GPIOI_PUDEN GPIOI_ODEN GPIOI_FCR GPIOI_DFS GPIOI_DSR
EXTIP	EXTI 配置寄存器	GPIO_EXTIA/B/…/H GPIO_EXTIDF
WKP	WKUP 控制寄存器	GPIO_PINWKEN

33.7 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（FM33FH05x(A)）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0005FFFF	O	O	O
Data Flash	0xA0000000	0xA0003FFF	O	O	X
Info Flash	0x00080000	0x000807FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20007FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB1	0x40010000	0x4001FFFF	O	O	X
Peripheral	0x40020000	0x4002FFFF	O	O	X

属性	地址范围		Read	Write	Fetch
	min	Max			
APB2					
Others	-	-	X	X	X

表 33-1 384K code flash 版本总线访问控制表

地址访问权限表（FM33FH04x(A)）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0003FFFF	O	O	O
Data Flash	0xA0000000	0xA0003FFF	O	O	X
Info Flash	0x00080000	0x000807FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20007FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB1	0x40010000	0x4001FFFF	O	O	X
Peripheral APB2	0x40020000	0x4002FFFF	O	O	X
Others	-	-	X	X	X

表 33-2 256K code flash 版本总线访问控制表

地址访问权限表（FM33FH02x(A)）

属性	地址范围		Read	Write	Fetch
	min	Max			
Code Flash	0x00000000	0x0001FFFF	O	O	O
Data Flash	0xA0000000	0xA0003FFF	O	O	X
Info Flash	0x00080000	0x000807FF	O	O	X
Vendor Reserved Flash	0x1FFFFA00	0x1FFFFFFF	O	X	X
SRAM	0x20000000	0x20003FFF	O	O	O
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB1	0x40010000	0x4001FFFF	O	O	X
Peripheral APB2	0x40020000	0x4002FFFF	O	O	X
Others	-	-	X	X	X

表 33-3 128K code flash 版本总线访问控制表

33.8 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。

33.9 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，可根据配置产生报警中断或者复位。

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。如果使能了停振（Lost Of Clock）复位功能，则直接复位芯片。

33.10 PLL 失锁检测

芯片的 PLL 带有锁定检测电路，当 PLL 失锁时，可根据配置产生报警中断或者复位。

同时，如果芯片正在使用 PLL 作为系统时钟，锁定检测电路发现 PLL 失锁时，芯片会自动将系统时钟切换为内部环振，避免系统死机。如果使能了失锁（Lost Of Lock）复位功能，则直接复位芯片。

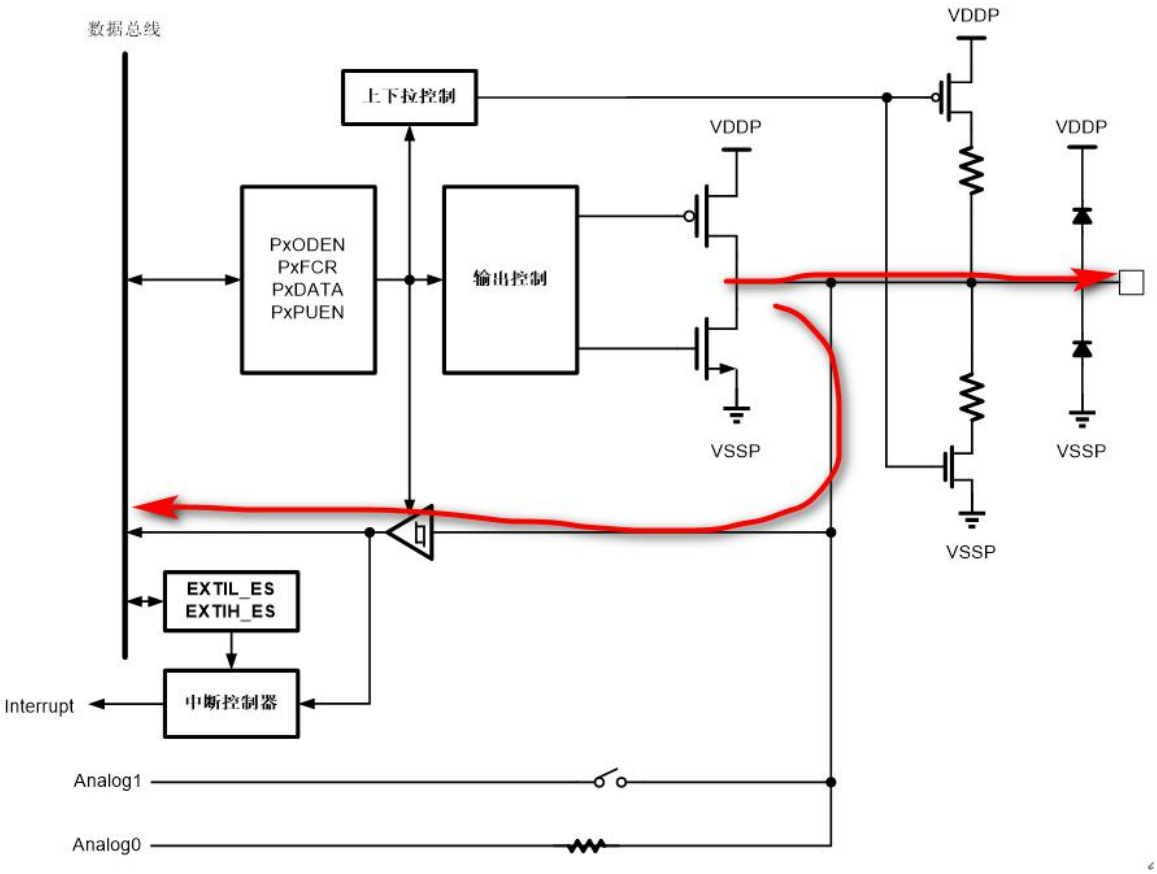
33.11 电源电压检测

芯片可以通过 SVD 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。

33.12 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，如下图所示。



33.13 LIN 自测试

参见 UART 章节中的 LIN 子章节

33.14 CAN 自测试

CAN-FD 支持环回测试（loopback），参见 CAN-FD 章节。

33.15 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。

33.16 DAC 自测试

DAC 输出可以连接到 ADC 输入，通过 ADC 采样 DAC 输出，可以判断 DAC 是否工作正常。

33.17 比较器自测试

通过以下方法实现比较器自测试：

- 比较器正端连接 DAC 输出
- 比较器负端连接内部基准电压
- 软件配置 DAC 输出一个低于基准的电压，此时比较器输出为 0
- 软件配置 DAC 输出一个高于基准的电压，此时比较器输出为 1
- 由于 DAC 的参考电压是 VDD，所以执行测试前，需要明确当前芯片的 VDD 电压，才能正确计算 DAC 的合理码值

33.18 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。

33.19 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_RAMPCR
0x04	RAM 奇偶校验状态寄存器 (RAM Parity Status Register)	SFU_RAMPSR
0x08	存储器访问保护控制寄存器 (Memory Access Protection Control Register)	SFU_MAPCR
0x0C	存储器访问保护状态寄存器 (Memory Access Protection Control Register)	SFU_MAPSR

33.19.1 RAM 奇偶校验控制寄存器（SFU_RAMPCR）

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

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2	RAMP_RE	RAM 奇偶校验复位使能 (RAM parity reset enable) 1：奇偶校验错误发生时复位芯片 0：奇偶校验错误发生时不复位芯片
1	RAMP_IE	RAM 奇偶校验中断使能 (RAM parity interrupt enable) 1：奇偶校验错误发生时产生中断 0：奇偶校验错误发生时不产生中断
0	--	RFU：未实现，读为 0

33.19.2 RAM 奇偶校验状态寄存器（SFU_RAMPSR）

名称	SFU_RAMPSR
offset	0x04

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

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	RAMP_IF	RAM 奇偶校验错误标志 (RAM parity interrupt flag)，硬件置位，软件写 1 清零 1：发生了奇偶校验错误 0：未发生奇偶校验错误
0	--	RFU：未实现，读为 0

33.19.3 存储访问保护控制寄存器（SFU_MAPCR）

名称	SFU_MAP_CR							
offset	0x08							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-					WKP	EXTIP	GPIOIP
位权限	U-0					R/W-0	R/W-0	R/W-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	GPIOHP	GPIOGP	GPIOFP	GPIOEP	GPIODP	GPIOCP	GPIOBP	GPIOAP
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	-	CLM2P	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	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
31:27	--	RFU：未实现，读为 0
26	WKP	WKUP 配置寄存器写保护
25	EXTIP	EXTI 配置寄存器写保护
24	GPIOIP	GPIO I 组寄存器写保护
23	GPIOHP	GPIO H 组寄存器写保护
22	GPIOGP	GPIO G 组寄存器写保护
21	GPIOFP	GPIO F 组寄存器写保护

Bit	助记符	功能描述
20	GPIOEP	GPIO E 组寄存器写保护
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	--	RFU：未实现，读为 0
14	CLM2P	CLM2 寄存器写保护 1：CLM2 控制寄存器不可改写 0：无保护
13	CLM1P	CLM1 寄存器写保护 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 控制寄存器保护(SVD Protection) 1：SVD 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

Bit	助记符	功能描述
		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。*

33.19.4 存储访问保护状态寄存器（SFU_MAPSR）

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

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	MAP_IF	存储器访问保护中断标志 (Memory Access Protection Interrupt flag)，硬件置位，软件写 1 清零。此标志置位后会触发 NMI 中断。 1：发生了非法地址访问 0：未发生非法地址访问

34 专用编程接口

34.1 概述

FM33FH0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。

34.2 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。

35 调试支持

35.1 概述

FM33FH0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FM33FH0芯片互联，并实现仿真调试。

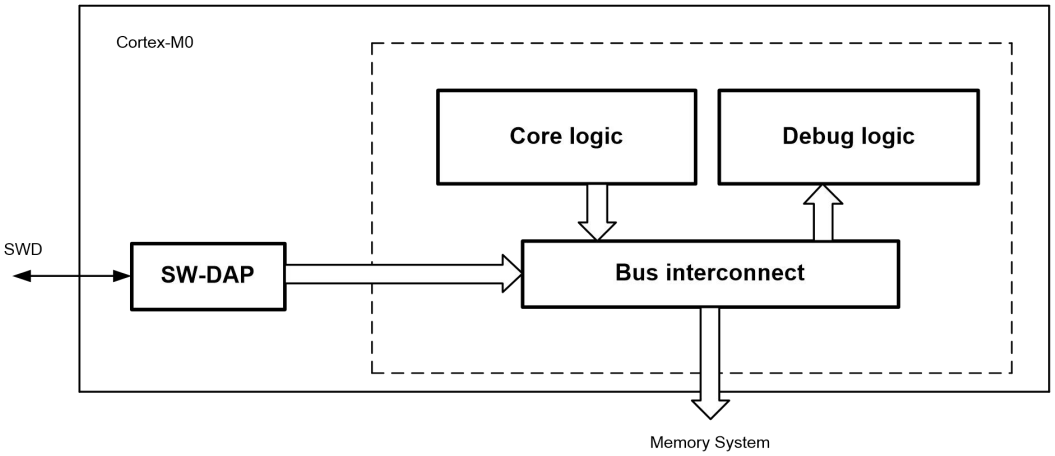


图 35-1 Cortex-M0 调试系统示意图

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。

35.2 Debug 引脚

35.2.1 SWD 引脚

FM33FH0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	PG8
SWCLK	SWD时钟输入	PG9

注意：芯片复位后PG8和PG9都默认为输入状态，与大部分GPIO不同。

35.2.2 上拉电阻

芯片复位后，SWDIO和SWCLK引脚默认使能内部上拉（约50K欧姆），用户无需在PCB上外接上拉电阻。

35.3 SWD 接口协议

35.3.1 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。

35.3.2 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

35.3.3 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。

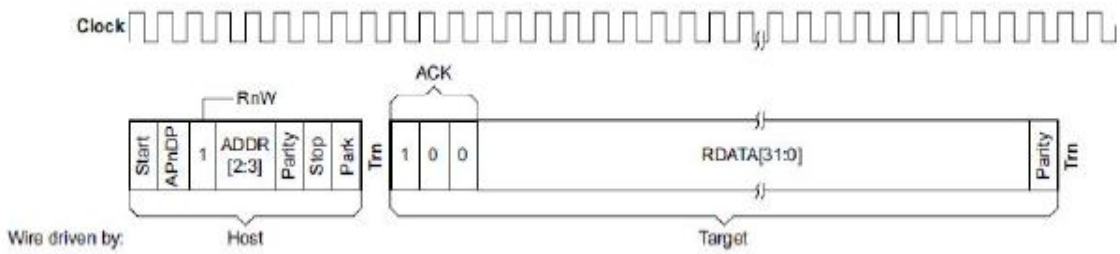
- 芯片复位，或者SWIO拉高50个SWCLK周期后，SW-DP处于RESET状态
- 拉低SWIO至少2个SWCLK周期后，SW-DP进入IDLE状态
- 当SW-DP处于RESET，主机必须先使其进入IDLE，然后对ID code寄存器进行读操作，才能激活SW-DP。否则从机会对主机的通信回应FAULT响应。

35.3.4 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。

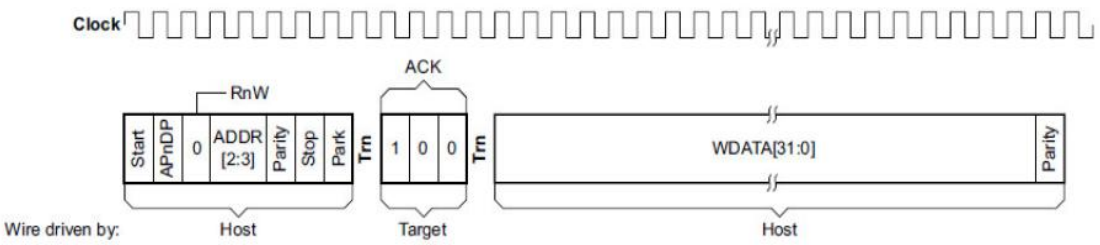


35.3.5 主机写操作

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。



35.4 SWD-DP 寄存器

35.4.1 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.

35.5 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

35.6 低功耗调试支持

通常情况下，当芯片进入Sleep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。

35.7 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。

35.8 寄存器

地址	名称	符号
0x40000004	MCU Debug 配置寄存器	DBG_CR
0x40000008	HARDFAULT 查询寄存器	HDFR

35.8.1 MCU DEBUG 配置寄存器

FM33FH0 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	DBG_CR							
地址	0x40000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-						DBG_RCEN	DBG_SLEEP
位权限	U-0						R/W-0	R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT16_STOP	DBG_BT16_STOP	DBG_TAU_STOP	-	DBG_ATIM_STOP	-	-	-
位权限	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	U-0	U-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	-	-	-	-	-	DBG_WWDT_STOP	DBG_IWDT_STOP
位权限	U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1

Bit	助记符	功能描述
31:18	--	RFU：未实现，读为 0
17	DBG_RCEN	Debug 模式下是否支持标志寄存器读清零操作 0：debug 禁止标志读清零操作 1：debug 允许标志读清零操作
16	DBG_SLEEP	休眠模式下的 debug 配置 0：正常进入休眠，关闭 HCLK 和 FCLK，debugger 无法保持连接 1：不关闭 HCLK 和 FCLK，debugger 能够在休眠模式下保持连接
15	DBG_LPT1	Debug 状态下 LPTIM16 使能控制位

Bit	助记符	功能描述
	6_STOP	1：Debug 时关闭 LPTIM16 0：Debug 时保持 LPTIM16 原来状态
14	DBG_BT16_STOP	Debug 状态下 BSTIM16 使能控制位 1：Debug 时关闭 BSTIM16 0：Debug 时保持 BSTIM16 原来状态
13	DBG_TAU_STOP	Debug 状态下 TAU 使能控制位 1：Debug 时关闭 TAU 0：Debug 时保持 TAU 原来状态
12	--	RFU：未实现，读为 0
11	DBG_ATIM_STOP	Debug 状态下 ATIM 使能控制位 1：Debug 时关闭 ATIM 0：Debug 时保持 ATIM 原来状态
10:2	--	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

35.8.2 HardFault 查询寄存器

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

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6	DABORT_ADDR_FLAG	地址非对齐访问错误标志，写 1 清零 1：地址非对齐访问错误 0：未进行地址非对齐访问
5	DABORT_RESP_FLAG	非法地址访问错误标志，写 1 清零 1：总线传输中访问了非法地址导致 HRESP 为高产生错误 0：未访问非法地址
4	SVCUNDE	SVC instructions 未定义标志，写 1 清零

Bit	助记符	功能描述
	F_FLAG	if the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set, the core should treat SVC instructions as though theywere UNDEFINED。
3	BKPT_FLAG	执行 BKPT 指令标志，写 1 清零 1：执行了 BKPT 指令 0：未执行 BKPT 指令
2	TBIT_FLAG	Thumb-State 标志，写 1 清零 1：切换到 ARM 状态 0：处于 Thumb-State
1	DFLS_FLAG	DFLASH 取指标志，写 1 清零 1：执行了特殊指令代码，如试图在 DFLASH 区域内取指 0：无指令代码试图在 DFLASH 被执行
0	HDF_REQUEST_FLAG	hardfault 标志位，任何类型的 hardfault 都会导致该位置位，写 1 清零 1：hardfault 请求 0：无 hardfault 请求

36 器件签名信息

每一颗FM33FH0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。

36.1 唯一器件 ID

唯一器件ID保存在Flash中，包含4个word，分别保存了芯片的Lot ID、Wafer ID、芯片在wafer上的坐标位置。通过这个数据，保证每颗芯片都有独立不重复的识别码。

具体使用方法请咨询复旦微电子。

版本列表

版本号	发布日期	页数	章节或图表	更改说明
1.0	2024.1.23			first release
1.1	2024.1.26			更新部分功耗参数
1.2	2024.3.5			增加GPIOAC参数
1.3	2024.5.6			修正nvmif寄存器
1.4	2024.8			修正LQFP64封装图PC12和PG7 TAU定义
1.5	2024.9			修正LQFP64封装图PG7和PG10 SPI定义
1.6	2024.10			1 删除BOR相关描述 2 修改上下电电压值
1.7	2024.11			更新VOL/VOH表格
1.8	2024.11			更新SVD复位描述，更新PA0的驱动强度配置，更新dataflash非法地址描述，DAC outen描述
1.9	2024.12			更新引脚功能定义，SWD和PA14/PA15管脚描述，更新外设qspi地址描述，更新pdr_cfg寄存器和TSI相关章节描述，删除XTLF外部输入描述
2.0	2025.2			更新SPI的AC参数，更新VOL/VOH表格
2.1	2025.3			修改TSI的CINTA和CINTB容值，更新IWDTEN的OPTBYTES描述
2.1.1	2025.4			更新DEBUG配置寄存器
2.1.2	2025.5		4.4.5.7	更新单次编程描述
2.1.3	2025.5		16.2	更新uart框图
2.1.4			3.4	更新电参数章节
2.1.5	2025.5		2.1	更新64封装图，更新单次编程流程

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/