Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33LR0 系列低功耗 MCU 芯片

产品说明书

2025-07

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(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

图片目录 24

表格目录 35

1 产品综述 38

1.1 概述 38

1.2 芯片结构框图 40

1.3 产品型号列表 40

2 引脚和封装 42

2.1 封装和引脚排列 42

2.1.1 LQFP48 封装图 42

2.1.2 LQFP44 封装图 43

2.1.3 LQFP32 封装图 44

*2.1.4 QFN32 封装图（5mm * 5mm） 45*

*2.1.5 QFN32 封装图（4mm * 4mm） 46*

*2.1.6 QFN20 封装图（3mm * 3mm） 47*

2.1.7 TSSOP24 封装图 48

2.1.8 PA 引脚组数字外设功能 48

2.1.9 PB 引脚组数字外设功能 48

2.1.10 PC 引脚组数字外设功能 49

2.1.11 PD 引脚组数字外设功能 49

2.1.12 PE 引脚组数字外设功能 50

2.1.13 引脚功能定义 50

2.2 封装尺寸图 54

2.2.1 LQFP48 54

2.2.2 LQFP44 55

2.2.3 LQFP32 57

2.2.4 QFN32（5mmx5mm） 59

2.2.5 QFN32（4mmx4mm） 61

2.2.6 QFN20（3mmx3mm） 63

2.3 焊接安装说明 64

2.4 MSL 等级 66

3 电参数 67

3.1 参数说明 67
3.2 极限参数 67
3.3 性能参数 67
3.3.1 *典型工作条件* *67*
3.3.2 *功耗参数* *68*
3.3.3 *复位和电源监控* *68*
3.3.4 *内部基准电压* *69*
3.3.5 *低功耗模式唤醒时间* *70*
3.3.6 *外部时钟源特性* *71*
3.3.7 *内部时钟源特性* *71*
3.3.8 *PLL 特性* *72*
3.3.9 *ADC 特性* *73*
3.3.10 *温度传感器* *74*
3.3.11 *模拟比较器特性* *75*
3.3.12 *Flash 存储器特性* *76*
3.3.13 *AC 参数* *76*
3.3.14 *GPIO 特性* *79*

4 总线与存储 82

4.1 系统总线 82
4.2 存储空间分配 83
4.2.1 *概述* *83*
4.2.2 *外设模块寄存器地址分配* *83*
4.3 RAM 85
4.3.1 *概述* *85*
4.3.2 *奇偶校验* *85*
4.4 Flash 86
4.4.1 *概述* *86*
4.4.2 *程序 flash* *86*
4.4.3 *数据 flash* *86*
4.4.4 *ECC* *86*
4.4.5 *NVR 扇区说明* *86*
4.4.6 *Flash 擦除与编程* *88*
4.4.7 *Flash 的内容保护* *90*

4.5 寄存器 92
4.5.1 *Flash 读取控制寄存器(FLS_RDCR)* *92*
4.5.2 *Flash 预取控制寄存器(FLS_PFTCON)* *93*
4.5.3 *用户配置字寄存器(FLS_OPTBR)* *93*
4.5.4 *Flash 擦写控制寄存器(FLS_EPCR)* *94*
4.5.5 *Flash Key 输入寄存器(FLS_KEY)* *95*
4.5.6 *Flash 中断使能寄存器(FLS_IER)* *95*
4.5.7 *Flash 标志寄存器(FLS_ISR)* *96*
4.5.8 *FLASH 检错中断使能寄存器(FLS_ERRIER)* *97*
4.5.9 *Flash ECC 寄存器(FLS_ECCSR)* *97*
4.5.10 *ACLOCK 寄存器 1(FLS_ACLOCK1)* *98*

5 电源管理单元（PMU） **100**

5.1 芯片工作电源 100
5.1.1 *电源域划分* *100*
5.1.2 *电源结构图* *100*
5.1.3 *片内快速建立基准源（AVREF）* *101*
5.2 功耗模式 101
5.2.1 *概述* *101*
5.2.2 *功耗模式与系统频率* *102*
5.2.3 *Active 模式* *102*
5.2.4 *LP Run 模式* *103*
5.2.5 *SLEEP 模式* *103*
5.2.6 *DEEPSLEEP 模式* *104*
5.3 唤醒源 104
5.4 寄存器 105
5.4.1 *低功耗控制寄存器(PMU_CR)* *105*
5.4.2 *唤醒时间控制寄存器(PMU_WKTR)* *106*
5.4.3 *休眠唤醒源标志寄存器(PMU_WKFR0)* *106*
5.4.4 *PMU 中断使能寄存器(PMU_IER)* *107*
5.4.5 *PMU 中断标志寄存器(PMU_ISR)* *108*
5.4.6 *模拟 BUFFER 控制寄存器(PMU_BUFCR)* *109*
5.4.7 *PTAT 控制寄存器(PMU_PTAT_CR)* *109*

6 处理器（CPU） **111**

6.1 概述 111
6.1.1 *处理器配置* *111*

6.2 寄存器 111
6.3 异常和中断 112
6.3.1 *中断向量表* *112*
6.3.2 *中断优先级* *112*
6.3.3 *错误处理* *112*
6.3.4 *锁定（Lockup）* *113*
6.4 调试特性 113
6.4.1 *调试功能引脚* *113*
6.4.2 *调试状态下的看门狗控制* *113*
6.4.3 *DEBUG 的复位* *113*
7 中断向量表（INTMUX） 114
7.1 中断向量表 114
7.2 中断向量多路选择 115
7.3 寄存器 115
7.3.1 *中断源选择寄存器(INTMUX_CR)* *116*
8 复位管理单元（RMU） 117
8.1 概述 117
8.2 模块框图 117
8.3 上下电复位 117
8.4 软件复位 119
8.5 NRST 引脚复位 119
8.6 寄存器 119
8.6.1 *BOR 配置寄存器(RMU_BORCR)* *120*
8.6.2 *复位配置寄存器(RMU_LKPCR)* *120*
8.6.3 *软件复位寄存器(RMU_SOFTRST)* *121*
8.6.4 *复位标志寄存器(RMU_RSTFR)* *121*
8.6.5 *外设复位使能寄存器(RMU_PRSTEN)* *122*
8.6.6 *AHB 外设复位寄存器(RMU_AHBRSTCR)* *123*
8.6.7 *APB 外设复位寄存器 1(RMU_APBRSTCR1)* *123*
8.6.8 *APB 外设复位寄存器 2(RMU_APBRSTCR2)* *125*
9 电源电压监测（SVD） 127
9.1 概述 127
9.2 结构框图 127

9.3 引脚定义 128
9.4 功能描述 128
9.4.1 *低压报警* *128*
9.4.2 *间歇使能模式* *129*
9.4.3 *欠压复位* *129*
9.5 电源检测阈值 130
9.6 寄存器 131
9.6.1 *SVD 配置寄存器(SVD_CFGR)* *131*
9.6.2 *SVD 控制寄存器(SVD_CR)* *132*
9.6.3 *SVD 中断使能寄存器(SVD_IER)* *132*
9.6.4 *SVD 状态和标志寄存器(SVD_ISR)* *133*
9.6.5 *SVD 参考电压选择寄存器(SVD_VSR)* *133*
10 独立看门狗（IWDT） 135
10.1 概述 135
10.2 结构框图 135
10.3 IWDT 功能描述 135
10.4 IWDT 窗口功能 136
10.5 IWDT 冻结 137
10.6 寄存器 137
10.6.1 *IWDT 清除寄存器(IWDT_SERV)* *137*
10.6.2 *IWDT 配置寄存器(IWDT_CR)* *138*
10.6.3 *IWDT 计数值寄存器(IWDT_CNT)* *139*
10.6.4 *IWDT 窗口寄存器(IWDT_WIN)* *139*
10.6.5 *IWDT 中断使能寄存器(IWDT_IER)* *140*
10.6.6 *IWDT 中断标志寄存器(IWDT_ISR)* *140*
11 窗口看门狗（WWDT） 142
11.1 功能描述 142
11.2 结构框图 142
11.3 WWDT 工作方式 142
11.4 寄存器 144
11.4.1 *WWDT 控制寄存器(WWDT_CR)* *144*
11.4.2 *WWDT 配置寄存器(WWDT_CFGR)* *145*
11.4.3 *WWDT 计数值寄存器(WWDT_CNT)* *146*
11.4.4 *WWDT 中断使能寄存器(WWDT_IER)* *146*

11.4.5 *WWDT 中断标志寄存器(WWDT_ISR)* *147*
11.4.6 *WWDT 预分频寄存器(WWDT_PSC)* *147*

12 时钟管理单元（CMU） **148**

12.1 概述 148
12.2 时钟架构 148
12.2.1 *时钟树* *148*
12.2.2 *SYSCLK 切换说明* *148*
12.2.3 *时钟安全性* *149*
12.2.4 *外设模块的总线时钟和工作时钟* *149*
12.2.5 *休眠模式下的外设时钟* *150*
12.3 高速片上环振（RCHF） 150
12.4 高频晶体振荡电路(XTHF) 150
12.4.1 *概述* *150*
12.4.2 *停振检测（HFDET）* *150*
12.5 低功耗 RC 振荡器(RCLP) 151
12.6 锁相环(PLL) 151
12.7 休眠唤醒的时钟处理 151
12.8 寄存器 151
12.8.1 *系统时钟配置寄存器(CMU_SYSCLKCR)* *151*
12.8.2 *RCHF 控制寄存器(CMU_RCHFCR)* *153*
12.8.3 *RCHF 调校寄存器(CMU_RCHFTR)* *154*
12.8.4 *PLL 控制寄存器(CMU_PLLCR)* *154*
12.8.5 *RCLP TRIM 寄存器(CMU_RCLPTR)* *156*
12.8.6 *XTHF 控制寄存器(CMU_XTHFCR)* *156*
12.8.7 *时钟控制中断使能寄存器(CMU_IER)* *157*
12.8.8 *时钟控制中断标志寄存器(CMU_ISR)* *158*
12.8.9 *外设总线时钟控制寄存器 1(CMU_PCLKCR1)* *158*
12.8.10 *外设总线时钟控制寄存器 2(CMU_PCLKCR2)* *159*
12.8.11 *外设总线时钟控制寄存器 3(CMU_PCLKCR3)* *160*
12.8.12 *外设总线时钟控制寄存器 4(CMU_PCLKCR4)* *161*
12.8.13 *外设工作时钟控制寄存器 1(CMU_OPCCR1)* *161*
12.8.14 *外设工作时钟控制寄存器 3(CMU_OPCCR3)* *162*
12.8.15 *AHB Master 控制寄存器(CMU_AHBMCR)* *163*

13 模拟比较器（Comparator） **165**

13.1 概述 165
13.2 结构框图 165
13.3 功能描述 165
13.3.1 *基本功能* *165*
13.3.2 *内部比较基准选择和产生* *165*
13.3.3 *时钟和复位* *166*
13.3.4 *引脚和内部信号连接* *166*
13.3.5 *输出数字滤波* *166*
13.3.6 *比较器中断* *167*
13.3.7 *比较器输出和触发信号输出* *167*
13.3.8 *比较器输出连接* *169*
13.4 寄存器 169
13.4.1 *COMP1 控制寄存器(COMP_CR1)* *169*
13.4.2 *COMP 中断配置寄存器(COMP_ICR)* *171*
13.4.3 *COMP 中断标志寄存器(COMP_ISR)* *171*
13.4.4 *COMP 缓冲器控制寄存器(COMP_BUFCR)* *172*

14 硬件除法器（DIVAS） 173

14.1 概述 173
14.2 工作流程 173
14.3 寄存器 173
14.3.1 *被除数寄存器(DIVAS_END)* *173*
14.3.2 *除数寄存器(DIVAS_SOR)* *174*
14.3.3 *商寄存器(DIVAS_QUOT)* *174*
14.3.4 *余数寄存器(DIVAS_REMD)* *175*
14.3.5 *状态寄存器(DIVAS_SR)* *175*

15 双线串行总线主机（I²C master） 177

15.1 概述 177
15.2 结构框图 177
15.3 引脚定义 177
15.4 时钟 177
15.5 接口时序 178
15.5.1 *接口时序图* *178*
15.5.2 *接口时序描述* *179*
15.6 I²C 工作模式 179

15.7 I²C 地址格式 179
15.8 I²C 初始化 180
15.8.1 *IO 配置* *180*
15.8.2 *主机波特率配置* *180*
15.9 I²C 主机功能 180
15.9.1 *7bit 寻址* *180*
15.9.2 *10bit 寻址* *185*
15.9.3 *SCL 延展（Slave Clock Stretching）* *189*
15.9.4 *超时机制* *189*
15.9.5 *可编程时序* *189*
15.10 寄存器 190
15.10.1 *I2C 主机配置寄存器(I2C_MSPCFGR)* *191*
15.10.2 *I2C 主机控制寄存器(I2C_MSPCR)* *191*
15.10.3 *I2C 主机中断使能寄存器(I2C_MSPIER)* *192*
15.10.4 *I2C 主机中断标志寄存器(I2C_MSPISR)* *193*
15.10.5 *I2C 主机状态寄存器(I2C_MSPSR)* *194*
15.10.6 *I2C 主机波特率设置寄存器(I2C_MSPBGR)* *195*
15.10.7 *I2C 主机收发缓冲寄存器(I2C_MSPBUF)* *196*
15.10.8 *I2C 主机时序控制寄存器(I2C_MSPTCR)* *196*
15.10.9 *I2C 主机超时寄存器(I2C_MSPTOR)* *197*
16 通用异步收发传输器（UART） 198
16.1 概述 198
16.2 结构框图 199
16.3 引脚定义 199
16.4 UART 类型区分 199
16.5 UART 字符描述 200
16.5.1 *UART 模式* *200*
16.5.2 *智能卡模式* *200*
16.6 功能描述 201
16.6.1 *时钟结构* *201*
16.6.2 *同步策略* *202*
16.6.3 *位接收采样* *202*
16.6.4 *数据发送* *203*
16.6.5 *数据接收* *204*
16.6.6 *RS232 硬件流控* *205*

16.6.7 *低功耗休眠唤醒* *206*
16.6.8 *DMA 模式下的发送完成中断* *207*
16.6.9 *半双工单线通信* *207*
16.7 波特率发生 208
16.7.1 *波特率计算* *208*
16.7.2 *波特率自适应* *209*
16.8 红外调制 209
16.9 接收超时 210
16.10 发送延迟 210
16.11 LIN 总线通信功能 210
16.11.1 *概述* *210*
16.11.2 *LIN 主机操作* *213*
16.11.3 *LIN 从机操作* *213*
16.11.4 *LIN 接收采样与噪声检测* *219*
16.11.5 *LIN 错误状态* *219*
16.12 寄存器 220
16.12.1 *UART 控制状态寄存器(UARTL_CSR)* *221*
16.12.2 *UART 中断使能寄存器(UARTL_IER)* *223*
16.12.3 *UART 中断标志寄存器(UARTL_ISR)* *224*
16.12.4 *UART 超时和延迟寄存器(UARTL_TODR)* *225*
16.12.5 *UART 接收缓冲寄存器(UARTL_RXBUF)* *226*
16.12.6 *UART 发送缓冲寄存器(UARTL_TXBUF)* *226*
16.12.7 *UART 波特率产生寄存器(UARTL_BGR)* *227*
16.12.8 *UART_LIN 模式寄存器(UARTL_MCR)* *227*
16.12.9 *UART_LIN 控制寄存器(UARTL_LINCR)* *228*
16.12.10 *UART_LIN 波特率同步寄存器(UARTL_LINBSR)* *229*
16.12.11 *UART_LIN 帧超时寄存器(UARTL_LINFTR)* *230*
16.12.12 *UART_LIN 发送时序寄存器(UARTL_LINTTR)* *230*
16.12.13 *UART_LIN Pre-sync 波特率寄存器(UARTL_LINPSR)* *231*
16.12.14 *UART_LIN break 段长度寄存器(UARTL_LINBKR)* *232*
16.12.15 *USART 控制状态寄存器(USART_CSR)* *232*
16.12.16 *USART 中断使能寄存器(USART_IER)* *234*
16.12.17 *USART 中断标志寄存器(USART_ISR)* *235*
16.12.18 *USART 超时和延迟寄存器(USART_TODR)* *236*
16.12.19 *USART 接收缓冲寄存器(USART_RXBUF)* *236*
16.12.20 *USART 发送缓冲寄存器(USART_TXBUF)* *237*

16.12.21 *USART 波特率产生寄存器(USART_BGR)* 237
16.12.22 *USART 模式寄存器(USART_MCR)* 238
16.12.23 *USART 智能卡状态标志寄存器(USART_SCISR)* 239
16.12.24 *USART 智能卡中断使能寄存器(USART_SCIER)* 240
16.12.25 *USART 智能卡帧格式控制寄存器(USART_FFCR)* 241
16.12.26 *USART 智能卡额外保护时间寄存器(USART_EGTR)* 242
16.12.27 *USART 智能卡输出时钟分频寄存器(USART_CODR)* 243
16.12.28 *红外调制配置寄存器(UART_IRCR)* 243

17 串行外设接口（SPI） **245**

17.1 概述 245
17.2 SPI 功能描述 245
17.2.1 *SPI 结构框图* 245
17.2.2 *引脚定义* 246
17.2.3 *SPI 接口时序* 247
17.2.4 *全双工数据通信* 250
17.2.5 *主机发送间隔* 251
17.2.6 *可编程数据字长* 251
17.2.7 *RX-ONLY 模式* 251
17.2.8 *主机 SSN 控制* 251
17.2.9 *数据冲突* 252
17.2.10 *使用 DMA 进行 SPI 收发* 252
17.3 寄存器 253
17.3.1 *SPI 控制寄存器 1(SPI_CR1)* 254
17.3.2 *SPI 控制寄存器 2(SPI_CR2)* 255
17.3.3 *SPI 控制寄存器 3(SPI_CR3)* 257
17.3.4 *SPI 中断控制寄存器(SPI_IER)* 257
17.3.5 *SPI 中断标志寄存器(SPI_ISR)* 258
17.3.6 *SPI 发送缓存寄存器(SPI_TXBUF)* 259
17.3.7 *SPI 接收缓存寄存器(SPI_RXBUF)* 260

18 直接存储访问控制器（DMA） **261**

18.1 概述 261
18.2 结构框图 261
18.3 工作原理 261
18.4 工作流程 262

18.5 访问带宽 264
18.6 通道控制 264
18.6.1 *DMA 请求映射* *264*
18.6.2 *通道优先级* *265*
18.6.3 *传输方向定义* *265*
18.6.4 *循环模式* *265*
18.7 寄存器 265
18.7.1 *DMA 全局控制器(DMA_GCR)* *266*
18.7.2 *通道 0 控制寄存器(DMA_CH0CR)* *266*
18.7.3 *通道 0 存储器指针寄存器(DMA_CH0MAD)* *267*
18.7.4 *通道 1 控制寄存器(DMA_CH1CR)* *268*
18.7.5 *通道 1 存储器指针寄存器(DMA_CH1MAD)* *269*
18.7.6 *通道 7 控制寄存器(DMA_CH7CR)* *270*
18.7.7 *通道 7 FLS 指针寄存器(DMA_CH7FLSAD)* *271*
18.7.8 *通道 7 RAM 指针寄存器(DMA_CH7RAMAD)* *271*
18.7.9 *DMA 中断状态标志寄存器(DMA_ISR)* *272*
18.7.10 *通道 0 传输剩余长度(DMA_CH0TFSADDR)* *273*
18.7.11 *通道 1 传输剩余长度(DMA_CH1TFSADDR)* *273*

19 循环冗余校验（CRC） 275

19.1 概述 275
19.2 软件配置过程 276
19.3 Golden 数据 277
19.4 DMA 接口 277
19.5 Flash 数据完整性校验 278
19.6 寄存器 278
19.6.1 *CRC 数据寄存器(CRC_DR)* *279*
19.6.2 *CRC 控制状态寄存器(CRC_CR)* *279*
19.6.3 *CRC 运算寄存器(CRC_LFSR)* *281*
19.6.4 *CRC 异或输出寄存器(CRC_XOR)* *281*
19.6.5 *CRC 多项式寄存器(CRC_POLY)* *282*

20 高级定时器（ATIM） 283

20.1 概述 283
20.2 主要特性 283
20.3 结构框图 284

20.4 管脚和内部信号连接 284
20.5 功能描述 286
20.5.1 *定时单元* 286
20.5.2 *定时器工作模式* 288
20.5.3 *重复计数器* 294
20.5.4 *Preload 寄存器* 295
20.5.5 *计数器工作时钟* 295
20.5.6 *内部触发信号（ITRx）* 300
20.5.7 *捕捉/比较通道* 300
20.5.8 *输入捕捉模式* 302
20.5.9 *软件 Force 输出* 303
20.5.10 *输出比较模式* 303
20.5.11 *PWM 输出* 304
20.5.12 *非对称 PWM 模式* 306
20.5.13 *组合 PWM 模式* 308
20.5.14 *组合三相 PWM 模式* 309
20.5.15 *互补输出和死区插入* 310
20.5.16 *刹车功能* 312
20.5.17 *互补输出通道信号状态逻辑表* 316
20.5.18 *6-step PWM 输出* 317
20.5.19 *单脉冲输出（one-pulse mode）* 318
20.5.20 *可重触发的单脉冲输出（retriggerable one-pulse mode）* 319
20.5.21 *外部事件清除 OCxREF* 321
20.5.22 *编码器接口模式（encoder interface）* 322
20.5.23 *定时器同步* 336
20.5.24 *触发信号输出* 339
20.5.25 *DMA 访问* 340
20.5.26 *DMA Burst* 341
20.5.27 *输入异或功能* 341
20.5.28 *霍尔传感器接口* 341
20.5.29 *Debug 模式* 342
20.6 寄存器 342
20.6.1 *ATIM 控制寄存器 1(ATIM_CR1)* 343
20.6.2 *ATIM 控制寄存器 2(ATIM_CR2)* 344
20.6.3 *ATIM 从机模式控制寄存器(ATIM_SMCR)* 347
20.6.4 *ATIM DMA 和中断使能寄存器(ATIM_DIER)* 350

20.6.5 *ATIM 状态寄存器(ATIM_SR)* *352*
20.6.6 *ATIM 事件产生寄存器(ATIM_EGR)* *353*
20.6.7 *ATIM 捕捉/比较模式寄存器 1(ATIM_CCMR1)* *354*
20.6.8 *ATIM 捕捉/比较模式寄存器 2(ATIM_CCMR2)* *358*
20.6.9 *ATIM 捕捉/比较使能寄存器(ATIM_CCER)* *360*
20.6.10 *ATIM 计数器寄存器(ATIM_CNT)* *361*
20.6.11 *ATIM 预分频寄存器(ATIM_PSC)* *362*
20.6.12 *ATIM 自动重载寄存器(ATIM_ARR)* *362*
20.6.13 *ATIM 重复计数寄存器(ATIM_RCR)* *363*
20.6.14 *ATIM 捕捉/比较寄存器 1(ATIM_CCR1)* *363*
20.6.15 *ATIM 捕捉/比较寄存器 2(ATIM_CCR2)* *364*
20.6.16 *ATIM 捕捉/比较寄存器 3(ATIM_CCR3)* *364*
20.6.17 *ATIM 捕捉/比较寄存器 4(ATIM_CCR4)* *365*
20.6.18 *ATIM 刹车和死区控制寄存器(ATIM_BDTR)* *366*
20.6.19 *ATIM 捕捉/比较模式寄存器 3(ATIM_CCMR3)* *368*
20.6.20 *ATIM 捕捉/比较寄存器 5(ATIM_CCR5)* *369*
20.6.21 *ATIM 捕捉/比较寄存器 6(ATIM_CCR6)* *370*
20.6.22 *ATIM 编码模式控制寄存器(ATIM_ECR)* *370*
20.6.23 *ATIM 输入选择寄存器(ATIM_TISEL)* *371*
20.6.24 *ATIM 辅助功能寄存器(ATIM_AFR)* *372*
20.6.25 *ATIM 刹车标志寄存器(ATIM_BRKFLAG)* *374*
20.6.26 *ATIM DMA 控制寄存器(ATIM_DCR)* *375*
20.6.27 *ATIM DMA 访问寄存器(ATIM_DMAR)* *375*

21 通用定时器（GPTIM） **377**

21.1 概述 377
21.2 主要特性 377
21.3 结构框图 377
21.4 管脚和内部连接 378
21.5 功能描述 381
21.5.1 *定时单元* *381*
21.5.2 *定时器工作模式* *382*
21.5.3 *计数器工作时钟* *387*
21.5.4 *内部触发信号（ITRx）的捕捉* *391*
21.5.5 *捕捉/比较通道* *392*
21.5.6 *输入捕捉模式* *393*

21.5.7 *软件 Force 输出* *394*
21.5.8 *输出比较模式* *394*
21.5.9 *PWM 输出* *395*
21.5.10 *非对称 PWM 模式* *397*
21.5.11 *组合 PWM 模式* *398*
21.5.12 *单脉冲输出* *399*
21.5.13 *外部事件清除 OCxREF* *400*
21.5.14 *编码器接口模式（encoder interface）* *401*
21.5.15 *GPTIM 从机模式* *415*
21.5.16 *DMA 访问* *417*
21.5.17 *DMA Burst* *418*
21.6 寄存器 418
21.6.1 *GPTIM 控制寄存器 1(GPTIM_CR1)* *419*
21.6.2 *GPTIM 控制寄存器 2(GPTIM_CR2)* *420*
21.6.3 *GPTIM 从机模式控制寄存器(GPTIM_SMCR)* *421*
21.6.4 *GPTIM DMA 和中断使能寄存器(GPTIM_DIER)* *424*
21.6.5 *GPTIM 状态寄存器(GPTIM_SR)* *426*
21.6.6 *GPTIM 事件产生寄存器(GPTIM_EGR)* *427*
21.6.7 *GPTIM 捕捉/比较模式寄存器 1(GPTIM_CCMR1)* *428*
21.6.8 *GPTIM 捕捉/比较模式寄存器 2(GPTIM_CCMR2)* *431*
21.6.9 *GPTIM 捕捉/比较使能寄存器(GPTIM_CCER)* *434*
21.6.10 *GPTIM 计数器寄存器(GPTIM_CNT)* *435*
21.6.11 *GPTIM 预分频寄存器(GPTIM_PSC)* *435*
21.6.12 *GPTIM 自动重载寄存器(GPTIM_ARR)* *436*
21.6.13 *GPTIM 捕捉/比较寄存器 1(GPTIM_CCR1)* *436*
21.6.14 *GPTIM 捕捉/比较寄存器 2(GPTIM_CCR2)* *437*
21.6.15 *GPTIM 捕捉/比较寄存器 3(GPTIM_CCR3)* *438*
21.6.16 *GPTIM 捕捉/比较寄存器 4(GPTIM_CCR4)* *438*
21.6.17 *GPTIM 编码器模式控制寄存器(GPTIM_ECR)* *439*
21.6.18 *GPTIM ITR 选择寄存器(GPTIM_TISEL)* *440*
21.6.19 *GTIM 辅助功能寄存器(GPTIM_AFR)* *441*
21.6.20 *GPTIM DMA 控制寄存器(GPTIM_DCR)* *441*
21.6.21 *GPTIM DMA 访问寄存器(GPTIM_DMAR)* *442*
22 定时器阵列单元（TAU） **444**
22.1 概述 444
22.2 结构框图 444

22.3 功能描述 445
22.3.1 *时钟和复位* *445*
22.3.2 *定时单元* *445*
22.3.3 *内部时钟计数* *446*
22.3.4 *外部输入信号计数* *446*
22.3.5 *内部捕捉输入信号计数* *447*
22.3.6 *级联模式* *448*
22.3.7 *捕捉功能* *451*
22.3.8 *捕捉源选择* *453*
22.3.9 *单个 PWM 输出* *455*
22.3.10 *级联 PWM 输出* *456*
22.3.11 *触发信号输出（TRGO）* *459*
22.4 寄存器 459
22.4.1 *TAU0 控制寄存器(TAU0_CR)* *460*
22.4.2 *TAU1 控制寄存器(TAU1_CR)* *461*
22.4.3 *TAU0 事件产生寄存器(TAU0_EGR)* *462*
22.4.4 *TAU1 事件产生寄存器(TAU1_EGR)* *462*
22.4.5 *TAU0 通道 x 配置寄存器(TAU0_TxCFGR x=[0, 7])* *463*
22.4.6 *TAU0 通道 x 模式寄存器(TAU0_TxMDR x=[0, 7])* *464*
22.4.7 *TAU0 通道 x 目标值寄存器(TAU0_TxARR x=[0, 7])* *465*
22.4.8 *TAU0 通道 x 捕捉比较寄存器(TAU0_TxCCR x=[0, 7])* *465*
22.4.9 *TAU0 通道 x 中断使能寄存器(TAU0_TxIER x=[0, 7])* *466*
22.4.10 *TAU0 通道 x 中断标志寄存器(TAU0_TxISR x=[0, 7])* *467*
22.4.11 *TAU0 通道 x 计数值寄存器(TAU0_TxCNT x=[0, 7])* *467*
22.4.12 *TAU1 通道 x 配置寄存器(TAU1_TxCFGR x=[0, 3])* *468*
22.4.13 *TAU1 通道 x 模式寄存器(TAU1_TxMDR x=[0, 3])* *469*
22.4.14 *TAU1 通道 x 目标值寄存器(TAU1_TxARR x=[0, 3])* *470*
22.4.15 *TAU1 通道 x 捕捉比较寄存器(TAU1_TxCCR x=[0, 3])* *471*
22.4.16 *TAU1 通道 x 中断使能寄存器(TAU1_TxIER x=[0, 3])* *471*
22.4.17 *TAU1 通道 x 中断标志寄存器(TAU1_TxISR x=[0, 3])* *472*
22.4.18 *TAU1 通道 x 计数值寄存器(TAU1_TxCNT x=[0, 3])* *472*
23 16 位基本定时器（BSTIM16） **474**
23.1 概述 474
23.2 主要特性 474
23.3 结构框图 474

23.4 功能描述 474
23.4.1 *定时单元* *474*
23.4.2 *定时器工作模式* *476*
23.4.3 *计数器工作时钟* *479*
23.4.4 *Debug 模式* *479*
23.5 寄存器 479
23.5.1 *BSTIM 控制寄存器 1(BSTIM16_CR1)* *479*
23.5.2 *BSTIM 控制寄存器 2(BSTIM16_CR2)* *480*
23.5.3 *BSTIM 中断使能寄存器(BSTIM16_IER)* *481*
23.5.4 *BSTIM 中断标志寄存器(BSTIM16_ISR)* *482*
23.5.5 *BSTIM 事件产生寄存器(BSTIM16_EGR)* *482*
23.5.6 *BSTIM 计数器寄存器(BSTIM16_CNT)* *483*
23.5.7 *BSTIM 预分频寄存器(BSTIM16_PSC)* *484*
23.5.8 *BSTIM 自动重载寄存器(BSTIM16_ARR)* *484*

24 16 位低功耗定时器（LPTIM16） 485

24.1 概述 485
24.2 结构框图 486
24.3 时钟和复位 486
24.4 定时器功能 486
24.4.1 *普通定时器* *486*
24.4.2 *外部脉冲触发计数* *487*
24.4.3 *外部异步脉冲计数* *487*
24.4.4 *Timeout 模式* *488*
24.5 捕捉比较功能 488
24.5.1 *16bit PWM* *488*
24.5.2 *输入捕捉* *489*
24.5.3 *输入数字滤波* *490*
24.6 正交编码器 490
24.7 寄存器 492
24.7.1 *LPTIM 配置寄存器(LPTIM16_CFGR)* *492*
24.7.2 *LPTIM 计数值寄存器(LPTIM16_CNT)* *494*
24.7.3 *LPTIM 捕捉比较控制和状态寄存器(LPTIM16_CCSR)* *494*
24.7.4 *LPTIM 目标值寄存器(LPTIM16_ARR)* *496*
24.7.5 *LPTIM 中断使能寄存器(LPTIM16_IER)* *497*
24.7.6 *LPTIM 中断标志寄存器(LPTIM16_ISR)* *498*

24.7.7 *LPTIM 控制寄存器(LPTIM16_CR)* *499*
24.7.8 *LPTIM 比较捕捉寄存器 1(LPTIM16_CCR1)* *499*
24.7.9 *LPTIM 比较捕捉寄存器 2(LPTIM16_CCR2)* *500*

25 模数转换器（ADC） **501**

25.1 概述 501
25.2 结构框图 501
25.3 输入通道 501
25.4 ADC 时钟和复位 502
25.5 功能描述 502
25.5.1 *使用 VDD 作为基准* *502*
25.5.2 *温度传感器* *503*
25.5.3 *可编程采样时间* *503*
25.5.4 *转换模式* *504*
25.5.5 *触发源* *507*
25.5.6 *过采样和硬件平均* *507*
25.5.7 *数据冲突和自动等待* *508*
25.5.8 *DMA* *508*
25.6 寄存器 509
25.6.1 *ADC 中断和状态寄存器(ADC_ISR)* *509*
25.6.2 *ADC 中断使能寄存器(ADC_IER)* *511*
25.6.3 *ADC 控制寄存器 1(ADC_CR1)* *511*
25.6.4 *ADC 控制寄存器 2(ADC_CR2)* *512*
25.6.5 *ADC 校准控制寄存器(ADC_CALCR)* *512*
25.6.6 *ADC 配置寄存器 1(ADC_CFGR1)* *513*
25.6.7 *ADC 配置寄存器 2(ADC_CFGR2)* *514*
25.6.8 *ADC 采样时间控制寄存器(ADC_SMTR)* *516*
25.6.9 *ADC 通道控制寄存器(ADC_CHER)* *516*
25.6.10 *ADC 数据寄存器(ADC_DR)* *517*

26 可编程胶合逻辑（PGL） **518**

26.1 概述 518
26.2 结构框图 518
26.3 功能描述 519
26.3.1 *LUT 真值表* *519*
26.3.2 *LUT 输入* *520*

26.3.3 *LUT 输出* *520*
26.3.4 *滤波和采样* *520*
26.3.5 *中断和触发* *521*
26.3.6 *低功耗模式* *521*
26.4 寄存器 521
26.4.1 *PGL 控制寄存器(PGL_CR)* *521*
26.4.2 *PGL 配置寄存器 0(PGL_CFGR0)* *522*
26.4.3 *PGL 配置寄存器 1(PGL_CFGR1)* *523*
26.4.4 *PGL 配置寄存器 2(PGL_CFGR2)* *524*
26.4.5 *PGL 配置寄存器 3(PGL_CFGR3)* *525*
26.4.6 *PGL 中断使能寄存器(PGL_IER)* *527*
26.4.7 *PGL 中断标志寄存器(PGL_ISR)* *527*
26.4.8 *LUT0 真值表寄存器(PGL_LUT0)* *528*
26.4.9 *LUT1 真值表寄存器(PGL_LUT1)* *528*
26.4.10 *LUT2 真值表寄存器(PGL_LUT2)* *529*
26.4.11 *LUT3 真值表寄存器(PGL_LUT3)* *529*
27 IO 端口（GPIO） 531
27.1 概述 531
27.2 引脚类型 531
27.2.1 *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* *531*
27.3 IO 端口功能定义 532
27.3.1 *GPIO 输入* *532*
27.3.2 *GPIO 输出* *532*
27.3.3 *数字外设功能* *533*
27.3.4 *模拟功能* *533*
27.4 特殊引脚 533
27.4.1 *晶体引脚* *533*
27.4.2 *SWD 引脚* *533*
27.4.3 *NRST 引脚* *533*
27.4.4 *WKUPx 引脚* *534*
27.5 外部引脚中断（EXTI） 534
27.5.1 *功能说明* *534*
27.5.2 *应用指南* *535*
27.6 快速 GPIO 输出 536
27.7 寄存器 536

27.7.1 *GPIOx 输入使能寄存器(GPIOx_INEN)* *537*
27.7.2 *GPIOx 上下拉使能寄存器(GPIOx_PUDEN)* *537*
27.7.3 *GPIOx 开漏使能寄存器(GPIOx_ODEN)* *538*
27.7.4 *GPIOx 功能选择寄存器(GPIOx_FCR)* *539*
27.7.5 *GPIOx 输出数据寄存器(GPIOx_DO)* *541*
27.7.6 *GPIOx 输出数据置位寄存器(GPIOx_DSET)* *541*
27.7.7 *GPIOx 输出数据复位寄存器(GPIOx_DRESET)* *542*
27.7.8 *GPIOx 输入数据寄存器(GPIOx_DIN)* *542*
27.7.9 *GPIOx 额外数字功能寄存器(GPIOx_DFS)* *543*
27.7.10 *GPIOx 压摆率驱动强度寄存器(GPIOx_SRDR)* *544*
27.7.11 *GPIOC 定时器通道选择寄存器(GPIOC_TMUX)* *544*
27.7.12 *EXTI 输入选择寄存器 0(EXTI_SEL0)* *545*
27.7.13 *EXTI 输入选择寄存器 1(EXTI_SEL1)* *548*
27.7.14 *EXTI 边沿选择和使能寄存器 0(EXTI_EDS0)* *549*
27.7.15 *EXTI 边沿选择和使能寄存器 1(EXTI_EDS1)* *551*
27.7.16 *EXTI 数字滤波控制寄存器(EXTI_DF)* *552*
27.7.17 *EXTI 中断标志寄存器(EXTI_IF)* *552*
27.7.18 *EXTI 输入信号寄存器(EXTI_DI)* *553*
27.7.19 *FOUT 配置寄存器(FOUTSEL)* *553*
27.7.20 *WKUP 使能寄存器(PINWKEN)* *554*

28 安全功能（SFU） **556**

28.1 概述 556
28.2 闪存 ECC 556
28.3 闪存 CRC 检查 556
28.4 RAM 奇偶校验 556
28.5 RAM 写保护 556
28.6 关键寄存器写保护 557
28.7 时钟频率检测 557
28.8 电源电压监控 557
28.9 程序流监控 557
28.10 IO 输出回读 557
28.11 通信数据 CRC 校验 557
28.12 非法地址访问检测 557
28.13 ADC 测试 558

28.14 温度检测 558
28.15 寄存器 558
28.15.1 *RAM 奇偶校验控制寄存器(SFU_RAMP_CR)* *559*
28.15.2 *RAM 奇偶校验状态寄存器(SFU_RAMP_SR)* *559*
28.15.3 *存储访问保护控制寄存器(SFU_MAP_CR)* *560*
28.15.4 *存储访问保护状态寄存器(SFU_MAP_SR)* *561*
29 专用编程接口 562
29.1 概述 562
29.2 编程器使用 562
30 调试支持 563
30.1 概述 563
30.2 Debug 引脚 563
30.2.1 *SWD 引脚* *563*
30.2.2 *上拉电阻* *563*
30.3 SWD 接口协议 564
30.3.1 *协议简介* *564*
30.3.2 *传输序列* *564*
30.3.3 *SW-DP ID code* *565*
30.3.4 *主机读操作* *565*
30.3.5 *主机写操作* *565*
30.4 SWD-DP 寄存器 566
30.4.1 *寄存器列表* *566*
30.5 Core debug 寄存器 566
30.6 低功耗调试支持 566
30.7 Debug 相关的配置项 566
30.8 寄存器 567
30.8.1 *MCU DEBUG 配置寄存器(DBGCFG)* *567*
30.8.2 *HardFault 查询寄存器(HDFFLAG)* *568*
31 器件签名信息 570
31.1 唯一器件 ID 570
版本列表 571
销售及服务网点 572

图片目录

图 1-1 芯片结构框图 40
图 2-1 FM33LR0x5 LQFP48 封装图 42
图 2-2 FM33LR0x4 LQFP44 封装图 43
图 2-3 FM33LR0x3 LQFP32 封装图 44
图 2-4 FM33LR0x3B QFN32 封装图 45
图 2-5 FM33LR0x3N QFN32 封装图 46
图 2-6 FM33LR0x2N QFN20 封装图 47
图 2-7 FM33LR0x2 TSSOP24 封装图 48
图 2-8 LQFP48 封装尺寸图 54
图 2-9 LQFP44 封装尺寸图 55
图 2-10 LQFP44 封装参数 56
图 2-11 LQFP32 封装尺寸图 57
图 2-12 LQFP32 封装参数图 58
图 2-13 QFN32_5x5 封装尺寸图 59
图 2-14 QFN32_5x5 封装参数 60
图 2-15 QFN32_4x4 封装尺寸图 61
图 2-16 QFN32_4x4 封装参数 62
图 2-17 QFN20_3x3 封装尺寸图 63
图 2-18 QFN20_3x3 封装参数 64
图 3-1 低功耗唤醒波形 70
图 3-2 ADC 参数说明 73
图 3-3 温度传感器输出曲线 75
图 3-4 SPI master timing 78
图 3-5 SPI slave timing, CPHA=0 78
图 3-6 SPI slave timing, CPHA=1 78
图 4-1 系统总线示意图 82

图 4-2 FM33LR0xx 总线地址（128+16） 83
图 4-3 RAM 奇偶校验 85
图 4-4 Flash 编程状态跳转图 88
图 5-1 芯片电源结构图 100
图 5-2 功耗模式与系统主频 102
图 8-1 芯片复位源框图 117
图 8-2 上下电复位示意图 118
图 8-3 POR, SVD, PDR 复位波形示意图 119
图 9-1 低压检测电路框图 127
图 9-2 低压检测电路工作时序 128
图 9-3 电源检测电路间歇工作模式 129
图 10-1 IWDT 结构框图 135
图 10-2 IWDT 窗口示意图 137
图 11-1 WWDT 结构框图 142
图 11-2 WWDT 窗口示意图 144
图 12-1 芯片时钟框图 148
图 13-1 比较器电路框图 165
图 13-2 比较器内建基准缓冲器 166
图 13-3 数字滤波（DFLEN=3）波形示意图 167
图 13-4 比较器中断产生 167
图 13-5 比较器输出逻辑 168
图 13-6 比较器输出上升沿产生触发输出，无滤波 168
图 13-7 比较器输出上升沿下降沿产生触发输出，无滤波 169
图 15-1 I^2C 主机模块框图 177
图 15-2 I^2C 总线时序 178
图 15-3 数据有效时序 178
图 15-4 起始（Start）与停止(Stop)命令定义 178
图 15-5 输出应答(ACK) 179
图 15-6 主机向 7 位地址从机写入数据时的帧格式 181

图 15-7 I²C 主机 7 位地址软件发送数据流图 182
图 15-8 I²C 主机对 7 位地址从机发送数据流图 183
图 15-9 主机从 7 位地址从机读取数据时的帧格式 183
图 15-10 I²C 软件 7 位地址接收数据流图 184
图 15-11 I²C 从 7 位地址从机读取数据流图 185
图 15-12 双向数据通信帧格式 185
图 15-13 10bit 寻址，主机向从机写入数据 186
图 15-14 I²C 软件 10 位地址发送数据流图 187
图 15-15 10bit 寻址，主机从从机读取数据 187
图 15-16 I²C 软件 10 位地址接收数据流图 188
图 15-17 I²C 软件重发数据流图 189
图 15-18 主机时序控制 190
图 16-1 UART 结构框图 199
图 16-2 UART 字符描述 200
图 16-3 7816 字符描述 201
图 16-4 位接收 16 倍采样 202
图 16-5 位接收 8 倍采样 202
图 16-6 UART 异步发送波形 1 203
图 16-7 UART 异步发送波形 2 204
图 16-8 UART 异步发送波形 3 204
图 16-9 nRTS 流控 205
图 16-10 nCTS 流控 206
图 16-11 UART 半双工单线通信连接示意图 208
图 16-12 红外调制波形 210
图 16-13 UART 发送延迟 210
图 16-14 LIN 总线拓扑 211
图 16-15 LIN 总线帧结构 211
图 16-16 LIN break field 211
图 16-17 LIN sync field 212

图 16-18 LIN 帧 ID 212
图 16-19 LIN 总线唤醒 214
图 16-20 Normal Break 215
图 16-21 Break Error 216
图 16-22 LIN 波特率自适应过程 217
图 16-23 波特率误差较大，无 pre-sync 217
图 16-24 LIN 从机接收流程 218
图 16-25 字节间隔和应答间隔 218
图 16-26 LIN bit error 冲突检测 219
图 16-27 LIN bit error 监测点 219
图 17-1 SPI 结构框图 246
图 17-2 SPI Master/SPI Slave 互连 247
图 17-3 SPI 数据/时钟时序图（CPHA=0） 248
图 17-4 SPI 数据/时钟时序图（CPHA=1） 248
图 17-5 4 线半双工写操作 249
图 17-6 4 线半双工读操作（无 dummy cycle） 249
图 17-7 4 线半双工读操作（有 dummy cycle） 250
图 17-8 SPI SSN 时序图（SSNM=1，CPHA=0） 252
图 17-9 SPI SSN 时序图（SSNM=0） 252
图 18-1 DMA 结构框图 261
图 18-2 DMA 寄存器配置 263
图 18-3 DMA 工作流程 264
图 19-1 CRC 运算流程图 276
图 19-2 使用 DMA 对 RAM 中的数据进行 CRC 运算 278
图 20-1 高级定时器结构框图 284
图 20-2 预分频从 1 变为 2 的波形 287
图 20-3 预分频从 1 变为 4 的波形 288
图 20-4 向上计数波形，内部时钟不分频 289
图 20-5 向上计数波形，内部时钟 2 分频 289

图 20-6 ATIM_CR1.ARPE=0（ATIM_ARR 没有预装载）时的更新事件 290
图 20-7 ATIM_CR1.ARPE=1（ATIM_ARR 预装载）时的更新事件 290
图 20-8 向下计数，内部时钟不分频 291
图 20-9 向下计数，内部时钟 2 分频 292
图 20-10 向下计数，不使用重复计数时的更新事件 292
图 20-11 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6 293
图 20-12 计数器时序图，ATIM_CR1.ARPE=1 时的更新事件(计数器下溢出) 293
图 20-13 计数器时序图，ATIM_CR1.ARPE=1 时的更新事件(计数器上溢出) 294
图 20-14 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置 295
图 20-15 ATIM 时钟源框图 296
图 20-16 内部时钟源模式，时钟分频因子为 1 296
图 20-17 TI2 外部时钟连接例子 297
图 20-18 外部时钟模式 1 下的时序 297
图 20-19 外部时钟模式 1 下的时序 298
图 20-20 外部触发输入框图 298
图 20-21 外部时钟模式 2 下的时序 1 299
图 20-22 外部时钟模式 2 下的时序 2 300
图 20-23 捕获/比较通道(通道 1 输入部分) 301
图 20-24 捕获/比较通道 1 的主电路 301
图 20-25 捕获/比较通道的输出部分(通道 1 至 4) 301
图 20-26 PWM 输入捕获模式时序 302
图 20-27 输出比较模式，翻转 OC1 303
图 20-28 边沿对齐的 PWM 波形(ARR=7, PWM mode 1) 304
图 20-29 边沿对齐的 PWM 波形(PWM mode 1) 305
图 20-30 中央对齐的 PWM 波形(APR=7, PWM mode 1) 306
图 20-31 非对称 PWM 输出 307
图 20-32 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 1） 307
图 20-33 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 2） 308
图 20-34 通道 1/3 组合 PWM 输出 309

图 20-35 组合三相 PWM 输出 310
图 20-36 带死区插入的互补输出 310
图 20-37 死区波形延迟大于负脉冲 311
图 20-38 死区波形延迟大于正脉冲 311
图 20-39 死区插入后的互补输出波形与 CCxP 和 CCxNP 的关系 312
图 20-40 模式与输出状态术语 313
图 20-41 刹车输入结构 313
图 20-42 响应刹车的输出（OSSI=1） 315
图 20-43 产生六步 PWM，使用 COM 的例子(OSSR=1) 318
图 20-44 单脉冲模式的例子 319
图 20-45 重触发 OPM 模式 1，向上计数，CCR=0 320
图 20-46 重触发 OPM 模式 2，向上计数，CCR=0 320
图 20-47 重触发 OPM 模式 1，向下计数，CCR=0 321
图 20-48 重触发 OPM 模式 2，向下计数，CCR=0 321
图 20-49 ETR 信号清除 ATIM 的 OCxREF 322
图 20-50 编码器模式下的计数器操作实例 323
图 20-51 正交编码器不同边沿计数模式示例 324
图 20-52 时钟方向编码器示例 325
图 20-53 带方向的时钟编码器（CC1P=CC2P=0） 325
图 20-54 带方向的时钟编码器（CC1P=CC2P=1） 326
图 20-55 编码器 index 定义 327
图 20-56 IPOS[1:0]=11 时的 index 识别 328
图 20-57 IPOS[1:0]=11, index gated on A, ARR=9 328
图 20-58 IPOS[1:0]=00, index ungated, ARR=9 329
图 20-59 IPOS[1:0]=11, index gated on A&B, ARR=9 329
图 20-60 IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B 330
图 20-61 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B 330
图 20-62 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed 331
图 20-63 IPOS[1:0]=01, ungated index 331

图 20-64 IPOS[0]=1, clock+direction mode 332
图 20-65 IPOS[0]=1, directional clock mode 333
图 20-66 正交编码状态转移 333
图 20-67 index 错误撤销 334
图 20-68 index 错误置位 334
图 20-69 向下计数时 index 错误置位 335
图 20-70 counter mode preload 335
图 20-71 复位模式下的时序 336
图 20-72 门控模式下的时序 337
图 20-73 多 timer 同时使能 337
图 20-74 触发器模式下的时序 338
图 20-75 外部时钟模式 2＋触发模式下的时序 339
图 20-76 霍尔传感器接口 342
图 21-1 通用定时器架构示意图 377
图 21-2 预分频从 1 变为 2 的波形 382
图 21-3 预分频从 1 变为 4 的波形 382
图 21-4 向上计数波形，内部时钟不分频 383
图 21-5 向上计数波形，内部时钟 2 分频 383
图 21-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件 384
图 21-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件 384
图 21-8 向下计数，内部时钟不分频 385
图 21-9 向下计数，内部时钟 2 分频 385
图 21-10 向下计数，不使用重复计数时的更新事件 386
图 21-11 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6 386
图 21-12 计数器时序图，ARPE=1 时的更新事件(计数器下溢) 387
图 21-13 计数器时序图，ARPE=1 时的更新事件(计数器溢出) 387
图 21-14 GPTIM 时钟源框图 388
图 21-15 内部时钟源模式，时钟分频因子为 1 388
图 21-16 TI2 外部时钟连接例子 388

图 21-17 外部时钟模式 1 下的时序 389
图 21-18 外部时钟模式 1 下的时序 389
图 21-19 外部触发输入框图 390
图 21-20 外部时钟模式 2 下的时序 1g 390
图 21-21 外部时钟模式 2 下的时序 2 391
图 21-22 捕获/比较通道(通道 1 输入部分 392
图 21-23 捕获/比较通道 1 的主电路 392
图 21-24 捕获/比较通道的输出部分 393
图 21-25 PWM 输入捕获模式时序 394
图 21-26 输出比较模式，翻转 OC1 395
图 21-27 边沿对齐的 PWM 波形(ARR=7, PWM mode 1) 396
图 21-28 中央对齐的 PWM 波形(APR=7, PWM mode 1) 396
图 21-29 非对称 PWM 输出 397
图 21-30 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 1） 398
图 21-31 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 2） 398
图 21-32 通道 1/3 组合 PWM 输出 399
图 21-33 单脉冲模式的例子 400
图 21-34 ETR 信号清除 GPTIM 的 OCxREF 401
图 21-35 编码器模式下的计数器操作实例 402
图 21-36 正交编码器不同边沿计数模式示例 403
图 21-37 时钟方向编码器示例 403
图 21-38 带方向的时钟编码器（CC1P=CC2P=0） 404
图 21-39 带方向的时钟编码器（CC1P=CC2P=1） 404
图 21-40 编码器 index 定义 406
图 21-41 IPOS[1:0]=11 时的 index 识别 407
图 21-42 IPOS[1:0]=11, index gated on A, ARR=9 407
图 21-43 IPOS[1:0]=00, index ungated, ARR=9 408
图 21-44 IPOS[1:0]=11, index gated on A&B, ARR=9 408
图 21-45 IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B 409

图 21-46 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B 409
图 21-47 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed 410
图 21-48 IPOS[1:0]=01, ungated index 410
图 21-49 IPOS[0]=1, clock+direction mode 411
图 21-50 IPOS[0]=1, directional clock mode 412
图 21-51 正交编码状态转移 412
图 21-52 index 错误撤销 413
图 21-53 index 错误置位 413
图 21-54 向下计数时 index 错误置位 414
图 21-55 counter mode preload 414
图 21-56 复位模式下的时序 415
图 21-57 门控模式下的时序 416
图 21-58 触发器模式下的时序 416
图 21-59 外部时钟模式 2＋触发模式下的时序 417
图 22-1 TAU 整体结构框图 444
图 22-2 TAU 定时器结构框图 445
图 22-3 定时器内部时钟预分频计数 446
图 22-4 定时器外部事件上升沿递增时序 447
图 22-5 定时器外部事件上升下降沿递增，2 分频时序 447
图 22-6 Tixx 输入数字滤波示意图 447
图 22-7 定时器内部信号下降沿递增 448
图 22-8 定时器级联计数（trigger to increment） 449
图 22-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1 449
图 22-10 定时器级联计数（trigger to start） 450
图 22-11 定时器级联计数（trigger to start），从机无法产生溢出 451
图 22-12 定时器捕捉外部输入信号上升沿 452
图 22-13 定时器 2 分频捕捉外部输入信号下降沿 452
图 22-14 定时器脉冲宽度高电平捕捉 453
图 22-15 TAU 的 PWM 输出示意图（ARR=7，输出极性取反） 456

图 22-16 非对称级联 PWM 输出示意图 457
图 22-17 从机 ARR 大于主机的情况 458
图 22-18 非对称非交叠 PWM 输出示意图 459
图 23-1 BTIM16 结构框图 474
图 23-2 BTIM16 预分频从 1 变为 2 的波形 475
图 23-3 BTIM16 预分频从 1 变为 4 的波形 476
图 23-4 BTIM16 向上计数不分频的波形 477
图 23-5 BTIM16 向上计数 2 分频的波形 477
图 23-6 BTIM16 无预装载(ARPE=0)波形 478
图 23-7 BTIM16 预装载(ARPE=1)波形 478
图 23-8 BTIM16 的 UG 初始化波形 479
图 24-1 LPTIM16 结构框图 486
图 24-2 LPTIM16 外部 ETR 脉冲上升沿触发计数 487
图 24-3 LPTIM16 外部 ETR 脉冲异步计数（下降沿） 487
图 24-4 LPTIM16_TimeOut 模式 488
图 24-5 LPTIM16_PWM 输出 489
图 24-6 LPTIM16 输入信号边沿捕捉 490
图 24-7 LPTIM16 通道输入数字滤波 490
图 24-8 LPTIM16 编码器模式下的计数器 491
图 25-1 ADC 结构框图 501
图 25-2 SAR ADC 时钟结构图 502
图 25-3 ADC 输入通道示意图 504
图 25-4 ADC 单次转换全自动触发模式 505
图 25-5 ADC 单次转换半自动触发模式 506
图 25-6 ADC 连续转换模式 507
图 25-7 ADC 自动等待 508
图 25-8 ADC_DMA 案例 509
图 26-1 LUT 结构框图 518
图 26-2 LUT 实现 2 输入 NAND 示意图 519

图 26-3 使用 LUT0 为 UART 输入做数字滤波 521
图 27-1 普通 GPIO 结构框图 531
图 27-2 引脚输入数字滤波 535
图 27-3 EXTI 信号输入示意图 535
图 30-1 Cortex-M0 调试系统示意图 563

表格目录

表 1-1 FM33LR0xx 型号列表 40
表 3-1 芯片极限参数 67
表 3-2 芯片典型工作条件 67
表 3-3 ACTIVE 电流参数 68
表 3-4 SLEEP 电流参数 68
表 3-5 复位和电源监控参数 69
表 3-6 内部基准电压参数 69
表 3-7 唤醒时间参数 70
表 3-8 高频晶体振荡器参数 71
表 3-9 内部高频 RC 振荡器参数 71
表 3-10 内部低频 RC 振荡器参数 72
表 3-11 PLL 参数 72
表 3-12 ADC 参数 74
表 3-13 温度传感器定标结果保存 75
表 3-14 温度传感器参数 75
表 3-15 模拟比较器参数 75
表 3-16 Flash 参数 76
表 3-17 SPI timing characteristics at VDD=5V 76
表 3-18 SPI timing characteristics at VDD=3.3V 77
表 3-19 普通 IO 参数 79
表 3-20 引脚 AC 参数 80
表 4-1 外设模块总线地址列表 83
表 6-1 CPU 配置简表 111
表 6-2 Cortex-M0 内核寄存器简表 111
表 7-1 中断向量表 114
表 7-2 中断源多路选择表 115

表 10-1 IWDT 溢出周期表 136
表 11-1 WWDT 溢出周期表 143
表 12-1 外设模块时钟列表 149
表 15-1 I²C 保留地址定义 179
表 15-2 I²C 寻址字节定义 180
表 16-1 UART 类型列表 199
表 16-2 DMA 发送中断 207
表 16-3 常用时钟频率下波特率计算 208
表 17-1 SPI 引脚功能定义 246
表 18-1 DMA 通道映射列表 264
表 19-1 CRC 的 Golden 数据表 277
表 20-1 外部引脚连接 284
表 20-2 内部信号连接 285
表 20-3 ITR 信号连接 285
表 20-4 ETR 信号连接 286
表 20-5 OCREF_CLR 信号连接 286
表 20-6 刹车行为 316
表 20-7 输出状态逻辑表 316
表 20-8 encoder interface 计数方式（CC1P=CC2P=0） 322
表 20-9 directional clock mode 计数方式 326
表 20-10 DMA 访问 340
表 21-1 GPTIM 输入输出引脚 378
表 21-2 GPTIM 内部输入输出信号 378
表 21-3 GPTIM 内部触发信号连接 378
表 21-4 GPTIM 外部触发信号连接 379
表 21-5 GPTIM 通道 1 信号连接 379
表 21-6 GPTIM 通道 2 信号连接 379
表 21-7 GPTIM 通道 3 信号连接 380
表 21-8 GPTIM 通道 4 信号连接 380

表 21-9 encoder interface 计数方式（CC1P=CC2P=0） 401
表 21-10 directional clock mode 计数方式 404
表 21-11 DMA 访问 417
表 22-1 定时器级联触发选择 448
表 22-2 TAU0_定时器捕捉源选择 453
表 22-3 TAU1_定时器捕捉源选择 455
表 24-1 encoder interface 计数方式 490
表 25-1 ADC 输入通道 501
表 25-2 ADC 采样时间 503
表 26-1 PGL 相关信号定义 520
表 27-1 GPIO 功能逻辑定义表 532
表 27-2 FCR 定义表 532
表 28-1 总线访问权限表 557

1 产品综述

1.1 概述

- 宽电压范围：2.2~5.5V
- 工作温度范围：-40℃~+105℃
- 处理器内核
  - ARM Cortex-M0
  - 支持用户/特权模式
  - 支持中断向量表重映射
  - 最高主频 48MHz
  - SWD 调试接口
  - 24bit Systick 定时器
- 低功耗技术平台
  - 典型运行功耗 150uA/MHz
  - Sleep 模式：5uA
  - DeepSleep 模式，全部 RAM 保持+CPU 内核保持：1uA
- 存储器
  - 128KB Flash 带 ECC
  - 14.5KB Data Flash
  - Flash 擦写寿命：100,000 次
  - Flash 数据保存时间：10 年@85℃
  - 用户代码保护
  - 16KB RAM 带 parity
- 丰富的模拟外设
  - 高可靠上下电复位电路
  - 可编程电源监测模块（SVD），支持中断和复位模式
  - 12bit 1MspsSAR-ADC
  - 1x 模拟比较器
  - 温度传感器，全温区线性度+/-5℃
- 通信接口
  - UART*4，其中 UART1 支持 LIN 通信
  - SPI*2，主从模式
  - I2C*2，400KHz，主机模式
  - 2 通道外设 DMA
  - 可编程 CRC 校验模块
- 定时资源
  - TAU 定时器阵列，包含 12 个独立通道
  - 16bit 4 通道高级定时器*1

  - 16bit 4 通道通用定时器*1
  - 16bit 基本定时器*1
  - 24-bit Systick*1
  - 16bit 2 通道低功耗定时器*1
  - 看门狗定时器*2
- 时钟发生电路
  - 片上 8/16/24/32MHz 高速 RC 振荡器，8MHz 出厂调校误差+/-0.5%，全温区变化小于+/-2%
  - 4~40MHz 高频晶体振荡器，带有停振检测电路
  - 低功耗低速 RC 振荡器，32KHz
  - PLL，输入 4MHz，输出 32~48MHz
- 封装：LQFP48/LQFP44/LQFP32/QFN32(5*5)(4*4)/QFN20(3*3)

1.2 芯片结构框图

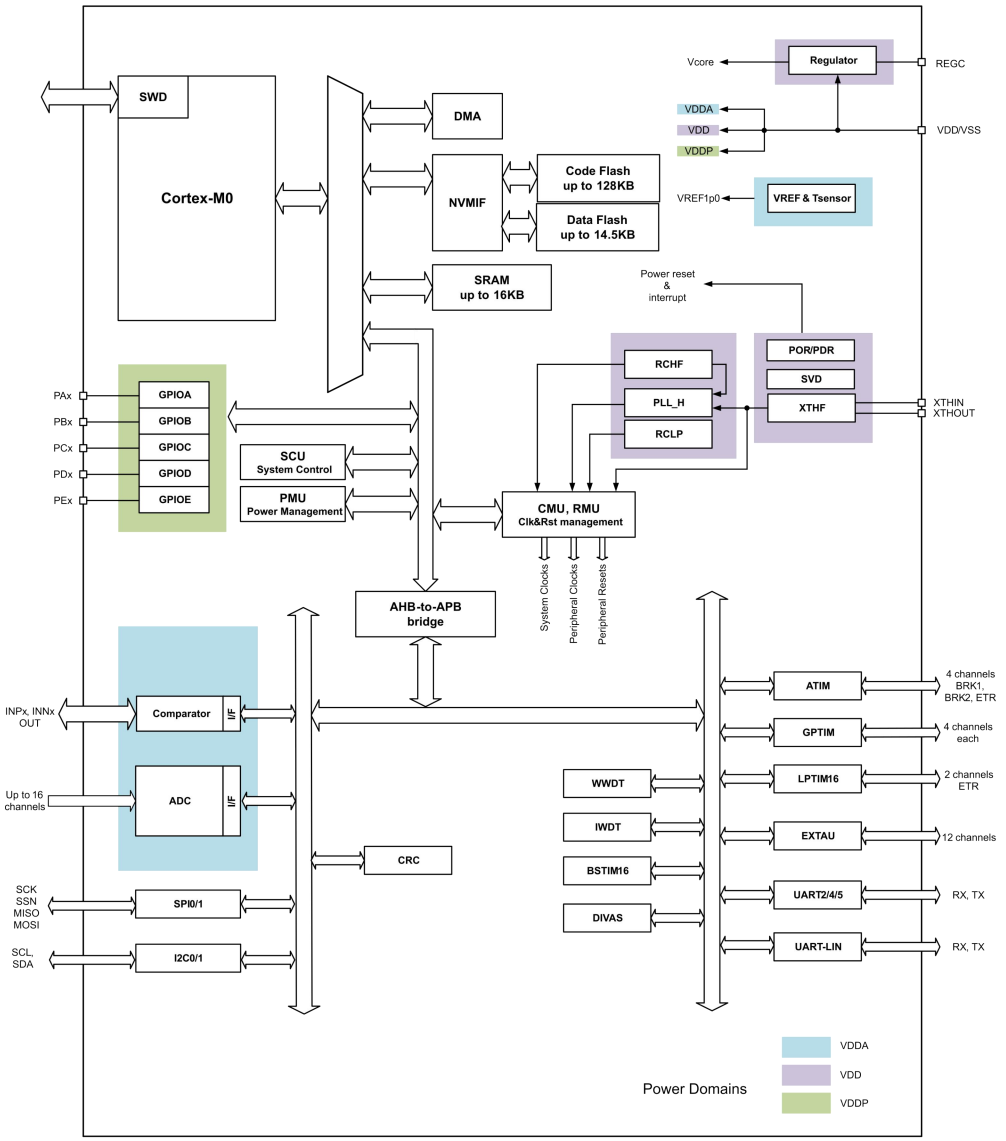


图 1-1 芯片结构框图

1.3 产品型号列表

表 1-1 FM33LR0xx 型号列表

型号	Code Flash (Kbytes)	Data Flash (Kbytes)	RAM (Kbytes)	封装
FM33LR025	128	14.5	16	LQFP48
FM33LR024	128	14.5	16	LQFP44
FM33LR023	128	14.5	16	LQFP32

型号	Code Flash (Kbytes)	Data Flash (Kbytes)	RAM (Kbytes)	封装
FM33LR023B	128	14.5	16	QFN32
FM33LR023N	128	14.5	16	QFN32
FM33LR022N	128	14.5	16	QFN20
FM33LR022	128	14.5	16	TSSOP24
FM33LR015	64	14.5	8	LQFP48
FM33LR014	64	14.5	8	LQFP44
FM33LR013	64	14.5	8	LQFP32
FM33LR013B	64	14.5	8	QFN32
FM33LR013N	64	14.5	8	QFN32
FM33LR012N	64	14.5	8	QFN20
FM33LR012	64	14.5	8	TSSOP24

2 引脚和封装

2.1 封装和引脚排列

2.1.1 LQFP48 封装图

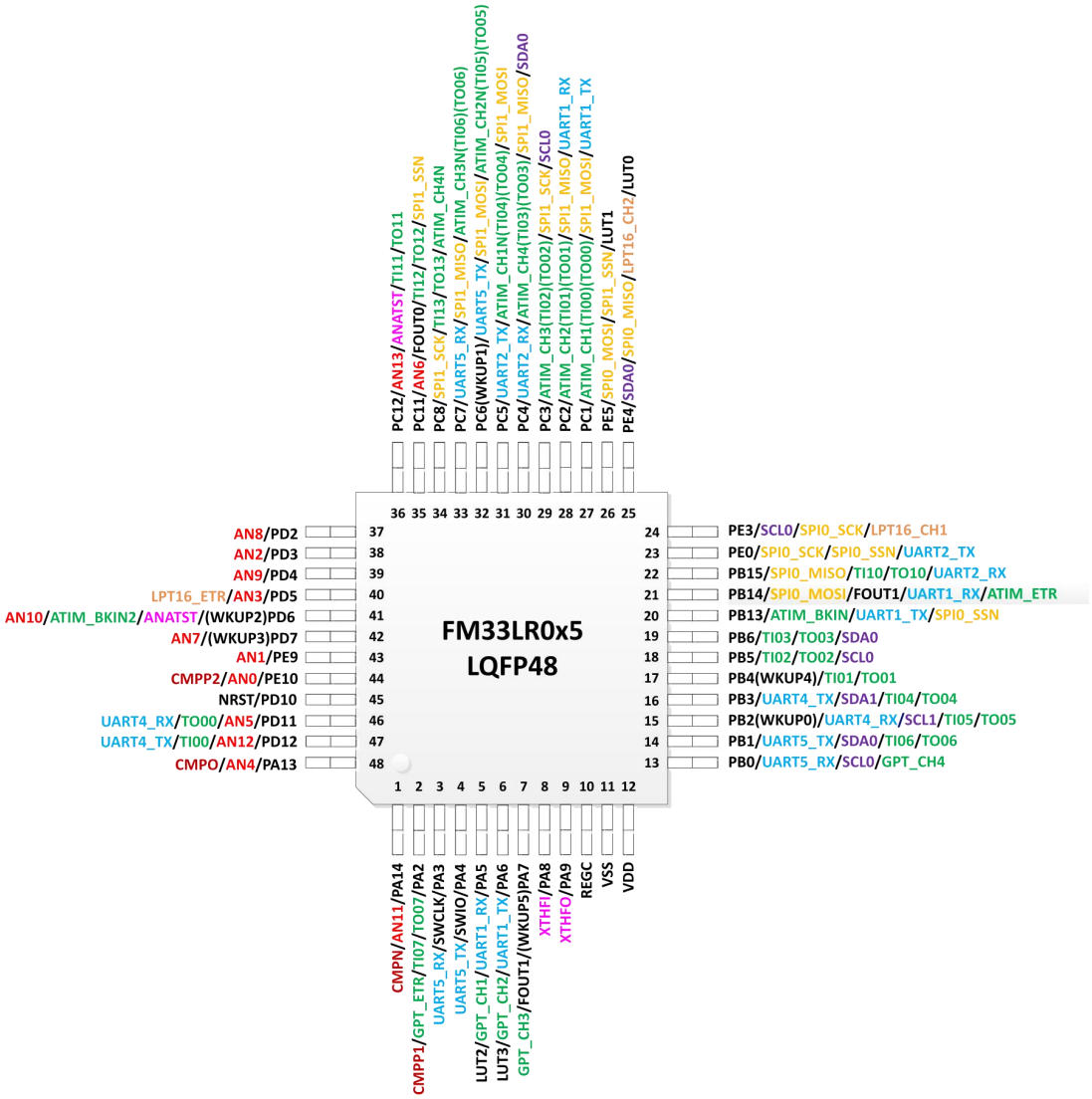


图 2-1 FM33LR0x5 LQFP48 封装图

2.1.2 LQFP44 封装图

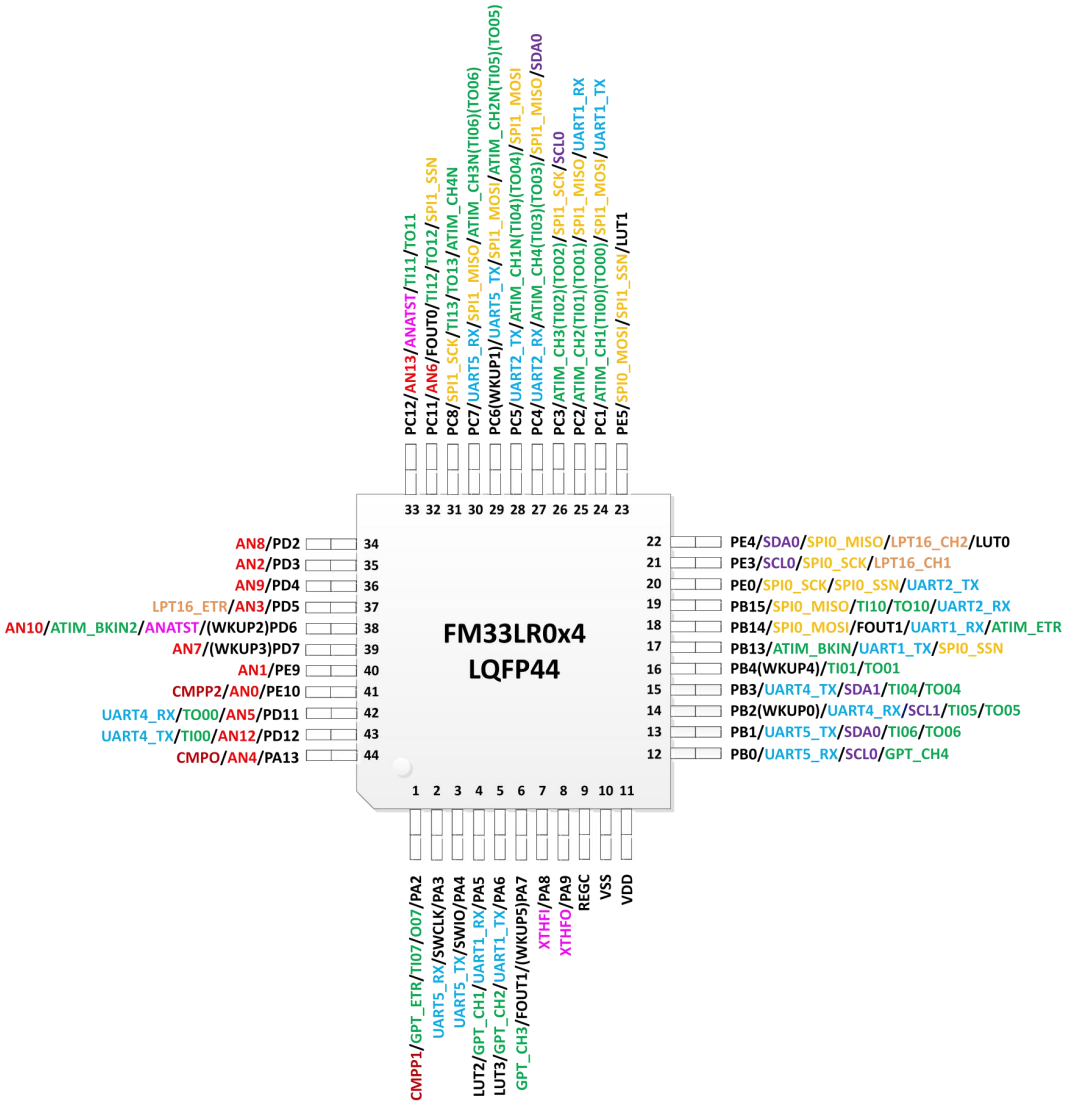


图 2-2 FM33LR0x4 LQFP44 封装图

2.1.3 LQFP32 封装图

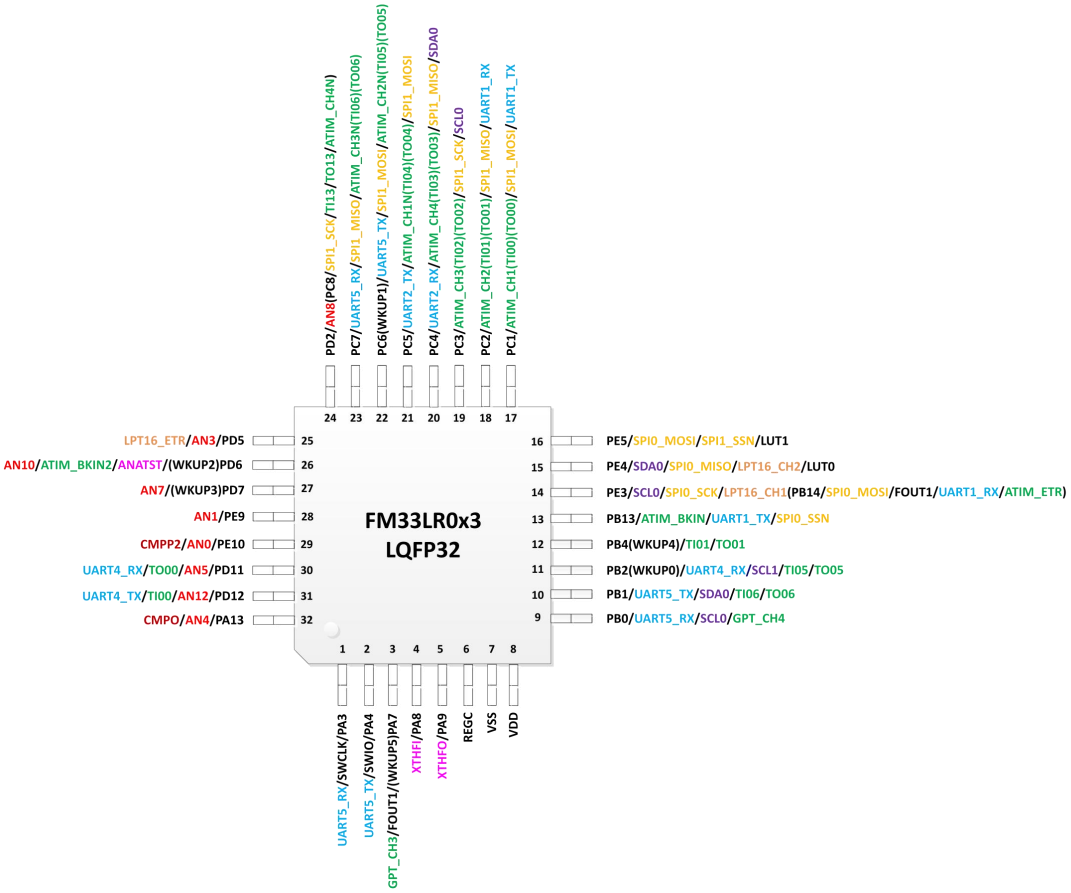


图 2-3 FM33LR0x3 LQFP32 封装图

2.1.4 QFN32 封装图（5mm * 5mm）

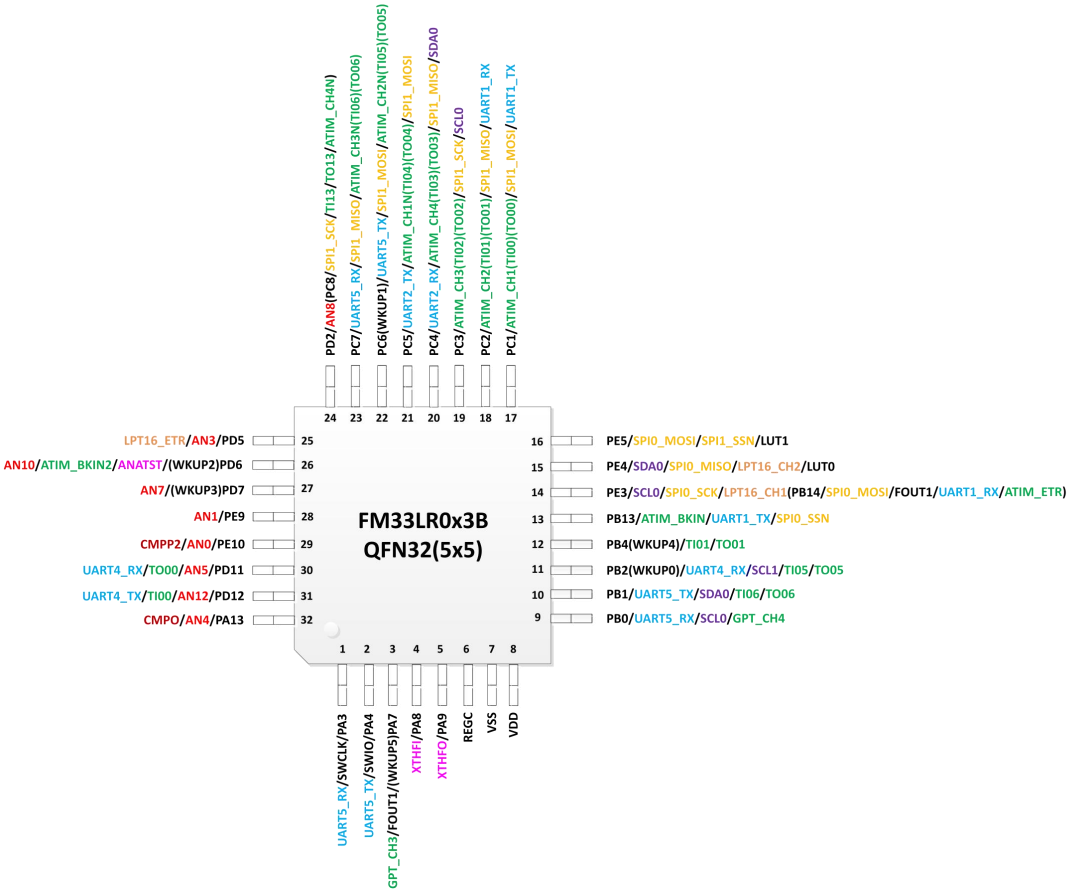


图 2-4 FM33LR0x3B QFN32 封装图

2.1.5 QFN32 封装图（4mm * 4mm）

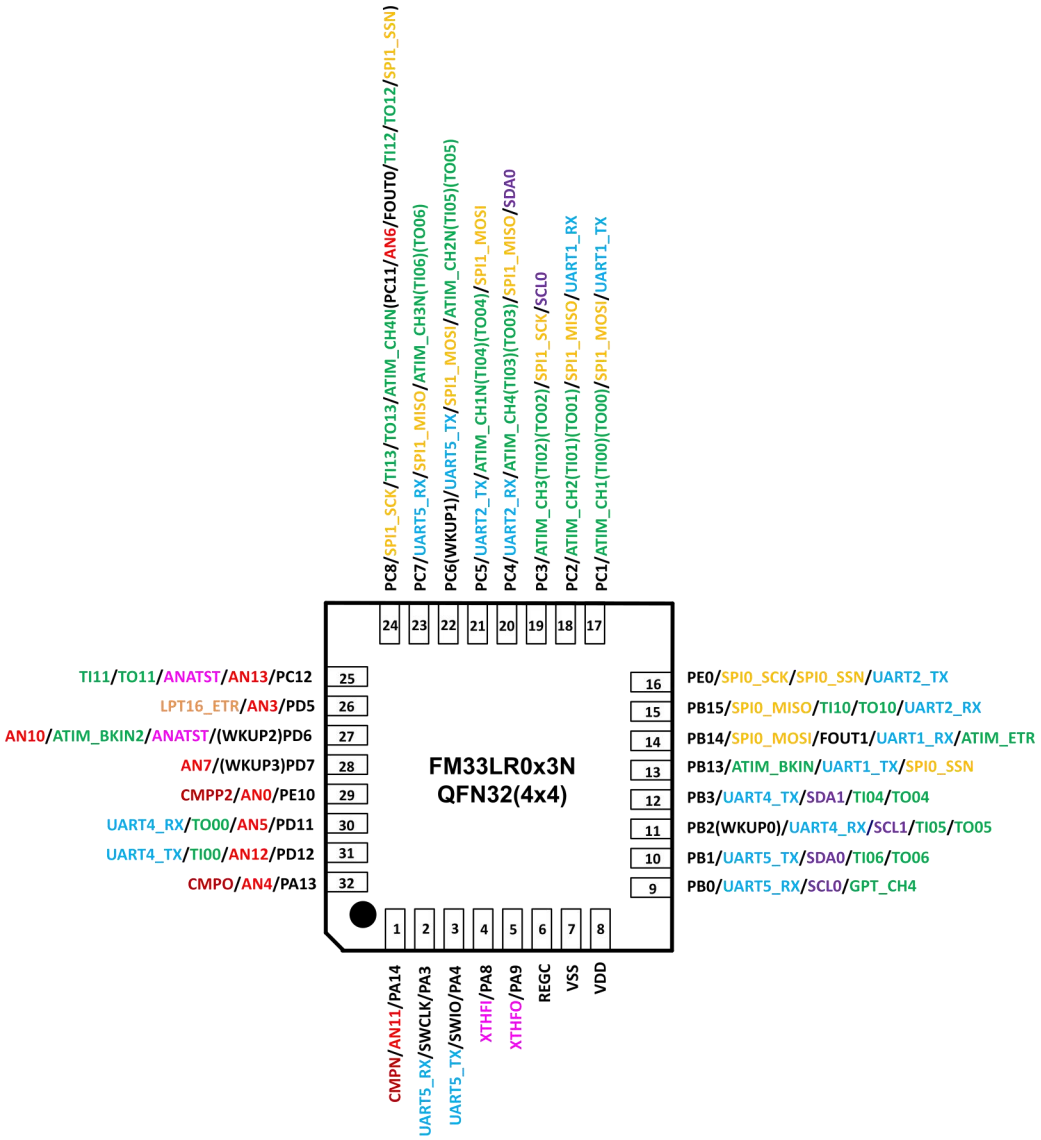


图 2-5 FM33LR0x3N QFN32 封装图

2.1.6 QFN20 封装图（3mm * 3mm）

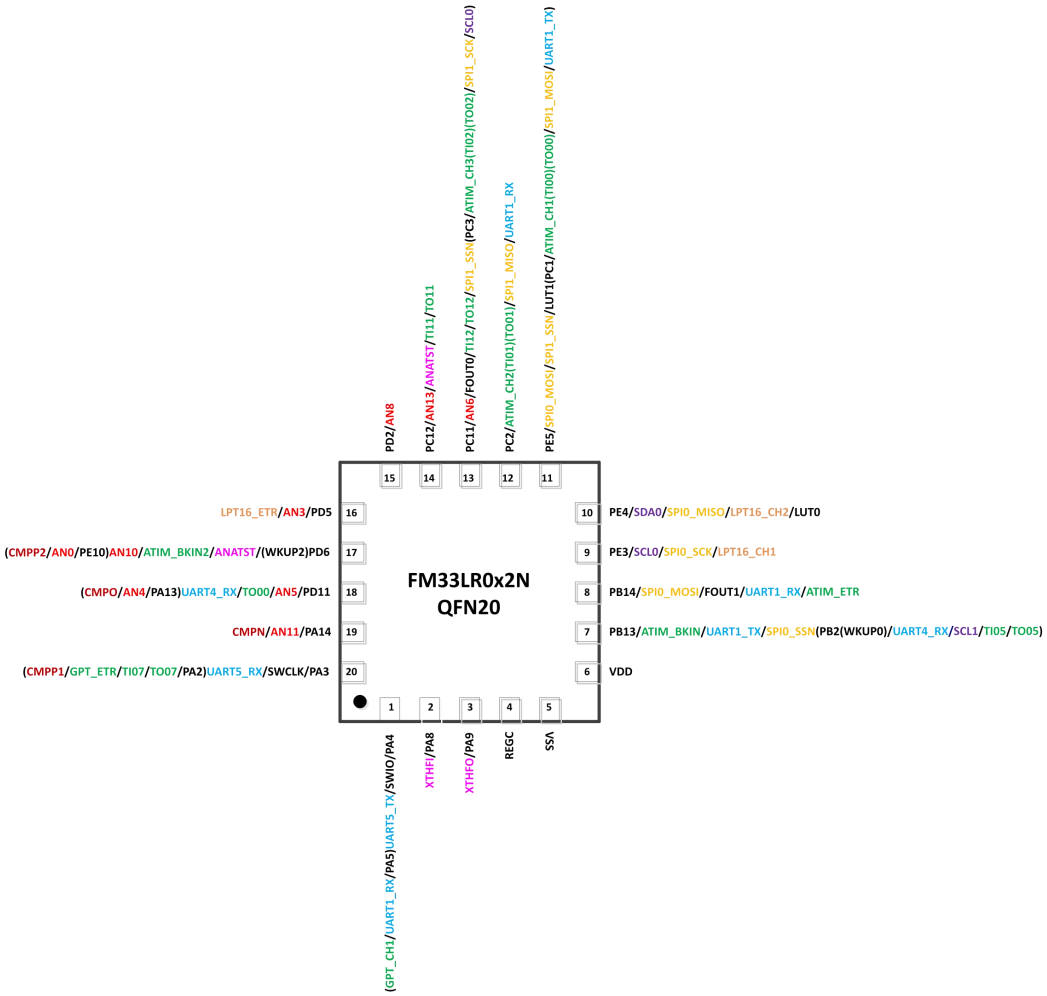


图 2-6 FM33LR0x2N QFN20 封装图

2.1.7 TSSOP24 封装图

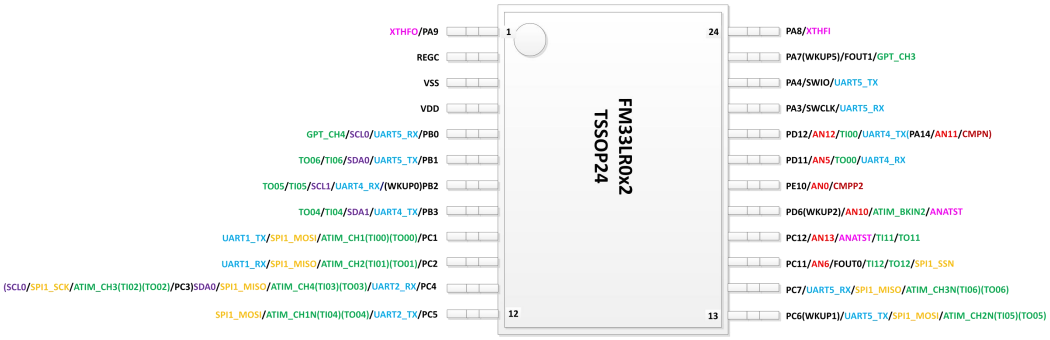


图 2-7 FM33LR0x2 TSSOP24 封装图

2.1.8 PA 引脚组数字外设功能

GPIO	PADFS	PADFS	PADFS	PADFS
	00	01	10	11
PA0	–	–	–	–
PA1	–	–	–	–
PA2	TO07	TI07	GPT_ETR	–
PA3	SWCLK	UART5_RX	–	–
PA4	SWIO	UART5_TX	–	–
PA5	UART1_RX	UART1_RX	GPT_CH1	LUT2
PA6	UART1_TX	UART1_TX	GPT_CH2	LUT3
PA7	FOUT1	FOUT1	GPT_CH3	–
PA8	–	–	–	–
PA9	–	–	–	–
PA10	–	–	–	–
PA11	–	–	–	–
PA12	–	–	–	–
PA13	CMPO	–	–	–
PA14	–	–	–	–
PA15	–	–	–	–

2.1.9 PB 引脚组数字外设功能

GPIO	PBDFS	PBDFS	PBDFS	PBDFS
	00	01	10	11
PB0	UART5_RX	SCL0	GPT_CH4	–
PB1	UART5_TX	SDA0	TI06	TO06
PB2	UART4_RX	SCL1	TI05	TO05
PB3	UART4_TX	SDA1	TI04	TO04
PB4	–	TI01	TO01	–
PB5	–	TI02	TO02	SCL0
PB6	–	TI03	TO03	SDA0
PB7	–	–	–	–
PB8	–	–	–	–
PB9	–	–	–	–
PB10	–	–	–	–

GPIO	PBDFS	PBDFS	PBDFS	PBDFS
PB11	–	–	–	–
PB12	–	–	–	–
PB13	ATIM_BKIN	ATIM_BKIN	UART1_TX	SPI0_SSN
PB14	ATIM_ETR	SPI0_MOSI	FOUT1	UART1_RX
PB15	UART2_RX	SPI0_MISO	TI10	TO10

2.1.10 PC 引脚组数字外设功能

GPIO	PCDFS	PCDFS	PCDFS	PCDFS
	00	01	10	11
PC0	–	–	–	–
PC1	ATIM_CH1 (TI00)(TO00)	SPI1_MOSI	UART1_TX	–
PC2	ATIM_CH1 (TI01)(TO01)	SPI1_MISO	UART1_RX	–
PC3	ATIM_CH3 (TI02)(TO02)	SPI1_SCK	SCL0	–
PC4	UART2_RX	ATIM_CH4 (TI03)(TO03)	SPI1_MISO	SDA0
PC5	UART2_TX	ATIM_CH1N (TI04)(TO04)	SPI1_MOSI	–
PC6	UART5_TX	ATIM_CH2N (TI05)(TO05)	SPI1_MOSI	–
PC7	UART5_RX	ATIM_CH3N (TI06)(TO06)	SPI1_MISO	–
PC8	TI13	TO13	SPI1_SCK	ATIM_CH4N
PC9	–	–	–	–
PC10	–	–	–	–
PC11	SPI1_SSN	FOUT0	TI12	TO12
PC12	TI11	TO11	–	–
PC13	–	–	–	–
PC14	–	–	–	–
PC15	–	–	–	–

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

2.1.11 PD 引脚组数字外设功能

GPIO	PDDFS	PDDDFS	PDDFS	PDDFS
	00	01	10	11
PD0	–	–	–	–
PD1	–	–	–	–
PD2	–	–	–	–
PD3	–	–	–	–
PD4	–	–	–	–
PD5	LPT16_ETR	LPT16_ETR	–	–
PD6	ATIM_BKIN2		–	–
PD7	–	–	–	–
PD8	–	–	–	–

GPIO	PDDFS	PDDDFS	PDDFS	PDDFS
PD9	–	–	–	–
PD10	–	–	NRST	–
PD11	TO00	TO00	UART4_RX	UART4_RX
PD12	TI00	TI00	UART4_TX	UART4_TX
PD13	–	–	–	–
PD14	–	–	–	–
PD15	–	–	–	–

2.1.12 PE 引脚组数字外设功能

GPIO	PEDFS	PEDFS	PEDFS	PEDFS
	00	01	10	11
PE0	UART2_TX	SPI0_SCK	SPI0_SSN	–
PE1	–	–	–	–
PE2	–	–	–	–
PE3	SCL0	SCL0	SPI0_SCK	LPT16_CH1
PE4	LUT0	SDA0	SPI0_MISO	LPT16_CH2
PE5	LUT1	SPI0_MOSI	SPI1_SSN	–
PE6	–	–	–	–
PE7	–	–	–	–
PE8	–	–	–	–
PE9	–	–	–	–
PE10	–	–	–	–
PE11	–	–	–	–
PE12	–	–	–	–
PE13	–	–	–	–
PE14	–	–	–	–
PE15	–	–	–	–

2.1.13 引脚功能定义

Pin Number					Pin Function	Descriptions
QFN20	QFN32(4x4)	LQFP32/QFN32(5x5)	LQFP44	LQFP48		
19	1	–	–	1	PA14	CMPN/AN11 比较器负端输入/ADC 外部输入通道 11
20	–	–	1	2	PA2	CMPP1/GPT_ETR/TI07/TO07 比较器正端输入 1/GPTIM 外部触发输入/TAU0 输入通道 7/TAU0 输出通道 7
20	2	1	2	3	PA3	UART5_RX/SWCLKUART5 接收/SWD 时钟
1	3	2	3	4	PA4	UART5_TX/SWIOUART5 发送/SWD 时钟

Pin Number					Pin Function	Descriptions
QFN20	QFN32(4x4)	LQFP32/QFN32(5x5)	LQFP44	LQFP48		
1	–	–	4	5	PA5	GPT_CH1/UART1_RXGPTIM 通道 1/UART1 接收
–	–	–	5	6	PA6	GPT_CH2/UART1_TXGPTIM 通道 2/UART1 发送
–	–	3	6	7	PA7	GPT_CH3/FOUT1/ (WKUP5)GPTIM 通道 3/FOUT 输出 /WKUP 唤醒 5
2	4	4	7	8	PA8	XTHFIXTHF 输入
3	5	5	8	9	PA9	XTHFOXTHF 输出
4	6	6	9	10	REGC	LDO 输出，外接 1UF 电容到地
5	7	7	10	11	VSS	地
6	8	8	11	12	VDD	电源
–	9	9	12	13	PB0	UART5_RX/SCL0/GPT_CH4UART 5 接收/I2C0 时钟/GPTIM 通道 4
–	10	10	13	14	PB1	UART5_TX/SDA0/TI06/TO06UART5 发送/I2C0 数据/TAU0 输入通道 6/TAU0 输出通道 6
7	11	11	14	15	PB2	(WKUP0)/UART4_RX/SCL1/TI05/ TO05WKUP 唤醒 0/UART4 接收 /I2C1 数据/TAU0 输入通道 5/TAU0 输出通道 5
–	12	–	15	16	PB3	UART4_TX/SDA1/TI04/TO04UART4 发送/I2C1 时钟/TAU0 输入通道 4/TAU0 输出通道 4
–	–	12	16	17	PB4	(WKUP4)/TI01/TO01WKUP 唤醒 4//TAU0 输入通道 1/TAU0 输出通道 1
–	–	–	–	18	PB5	TI02/TO02/SCL0TAU0 输入通道 2/TAU0 输出通道 2/I2C0 时钟
–	–	–	–	19	PB6	TI03/TO03/SDA0TAU0 输入通道 3/TAU0 输出通道 3/I2C0 数据
7	13	13	17	20	PB13	ATIM_BKIN/UART1_TX/SPI0_SSNATIM 刹车信号输入/UART4 发送 /SPI0 片选
8	14	14	18	21	PB14	SPI0_MOSI/FOUT1/UART1_RXSPI0 主机发送从机接收/FOUT 输出 /UART1 接收
–	15	–	19	22	PB15	SPI0_MISO/TI10/TO10SPI0 主机接收从机发送收/TAU1 输入通道 0/TAU1 输出通道 0
–	16	–	20	23	PE0	SPI0_SCK/SPI0_SSNSPI0 时钟 /SPI0 片选

Pin Number					Pin Function	Descriptions
QFN20	QFN32(4x4)	LQFP32/QFN32(5x5)	LQFP44	LQFP48		
9	–	14	21	24	PE3	SCL0/SPI0_SCK/LPT16_CH1I2C0 时钟/SPI0 时钟/LPTIM 通道 1
10	–	15	22	25	PE4	SDA0/SPI0_MISO/LPT16_CH2I2C0 数据/SPI0 主机接收从机发送收/LPTIM 通道 2
11	–	16	23	26	PE5	SPI0_MOSI/SPI1_SSNSPI0 主机发送从机接收/SPI1 片选
11	17	17	24	27	PC1	ATIM_CH1(TI00)(TO00)/SPI1_MOSI/UART1_TXATIM 通道 1（TAU0 输入通道 0）（TAU0 输出通道 0）/ SPI1 主机发送从机接收/UART1 发送
12	18	18	25	28	PC2	ATIM_CH2(TI01)(TO01)/SPI1_MISO/UART1_RXATIM 通道 2（TAU0 输入通道 1）（TAU0 输出通道 1）/ SPI1 主机接收从机发送/UART1 接收
13	19	19	26	29	PC3	ATIM_CH3(TI02)(TO02)/SPI1_SCK/SCL0ATIM 通道 3（TAU0 输入通道 2）（TAU0 输出通道 2）/SPI1 时钟/ I2C0 时钟
–	20	20	27	30	PC4	UART2_RX/ATIM_CH4(TI03)(TO03)/SPI1_MISO/SDA0UART2 接收/ ATIM 通道 4（TAU0 输入通道 3）（TAU0 输出通道 3）/SPI1 主机接收从机发送/I2C0 数据
–	21	21	28	31	PC5	UART2_TX/ATIM_CH1N(TI04)(TO04)/SPI1_MOSIUART2 发送/ ATIM 互补通道 1（TAU0 输入通道 4）（TAU0 输出通道 4）/SPI1 主机发送从机接收
–	22	22	29	32	PC6	(WKUP1)/UART5_TX/SPI1_MOSI/ ATIM_CH2N(TI05)(TO05)WKUP 唤醒 1/UART5 发送/SPI1 主机发送从机接收/ ATIM 互补通道 2（TAU0 输入通道 5）（TAU0 输出通道 5）
–	23	23	30	33	PC7	UART5_RX/SPI1_MISO/ATIM_CH3N(TI06)(TO06)UART5 接收/SPI1 主机接收从机发送/ ATIM 互补通道 3（TAU0 输入通道 6）（TAU0 输出通道 6）
–	24	24	31	34	PC8	SPI1_SCK/TI13/TO13/ATIM_CH4NSPI1 时钟/TAU1 输入通道

Pin Number					Pin Function	Descriptions
QFN20	QFN32(4x4)	LQFP32/QFN32(5x5)	LQFP44	LQFP48		
						3/TAU1 输出通道 3/ATIM 互补通道 4
13	24	–	32	35	PC11	AN6/FOUT0/TI12/TO12/SPI1_SSN ADC 外部输入通道 6/FOUT 输出/TAU1 输入通道 2/TAU1 输出通道 2/SPI1 片选
14	25	–	33	36	PC12	AN13/ANATST/TI11/TO11ADC 外部输入通道 13/模拟信号输出通道/TAU1 输入通道 1/TAU1 输出通道 1
15	–	24	34	37	PD2	AN8ADC 外部输入通道 8
–	–	–	35	38	PD3	AN2ADC 外部输入通道 2
–	–	–	36	39	PD4	AN9ADC 外部输入通道 9
16	26	25	37	40	PD5	LPT16_ETR/AN3LPTIM 外部触发输入/ADC 外部输入通道 3
17	27	26	38	41	PD6	AN10/ATIM_BKIN2/ANATST/(WKUP2)ADC 外部输入通道 10/ATIM 刹车信号输入 2/模拟信号输出通道/WKUP 唤醒 2
–	28	27	39	42	PD7	AN7/(WKUP3)ADC 外部输入通道 7/WKUP 唤醒 3
–	–	28	40	43	PE9	AN1ADC 外部输入通道 1
17	29	29	41	44	PE10	CMPP2/AN0 比较器正端输入 2/ADC 外部输入通道 0
–	–	–	–	45	PD10	NRST 复位引脚
18	30	30	42	46	PD11	UART4_RX/TO00/AN5UART4 接收/TAU0 输出通道 0/ADC 外部输入通道 5
–	31	31	43	47	PD12	UART4_TX/TI00/AN12UART4 发送/TAU0 输入通道 0/ADC 外部输入通道 12
18	32	32	44	48	PA13	CMPO/AN4 比较器输出/ADC 外部输入通道 4

2.2 封装尺寸图

2.2.1 LQFP48

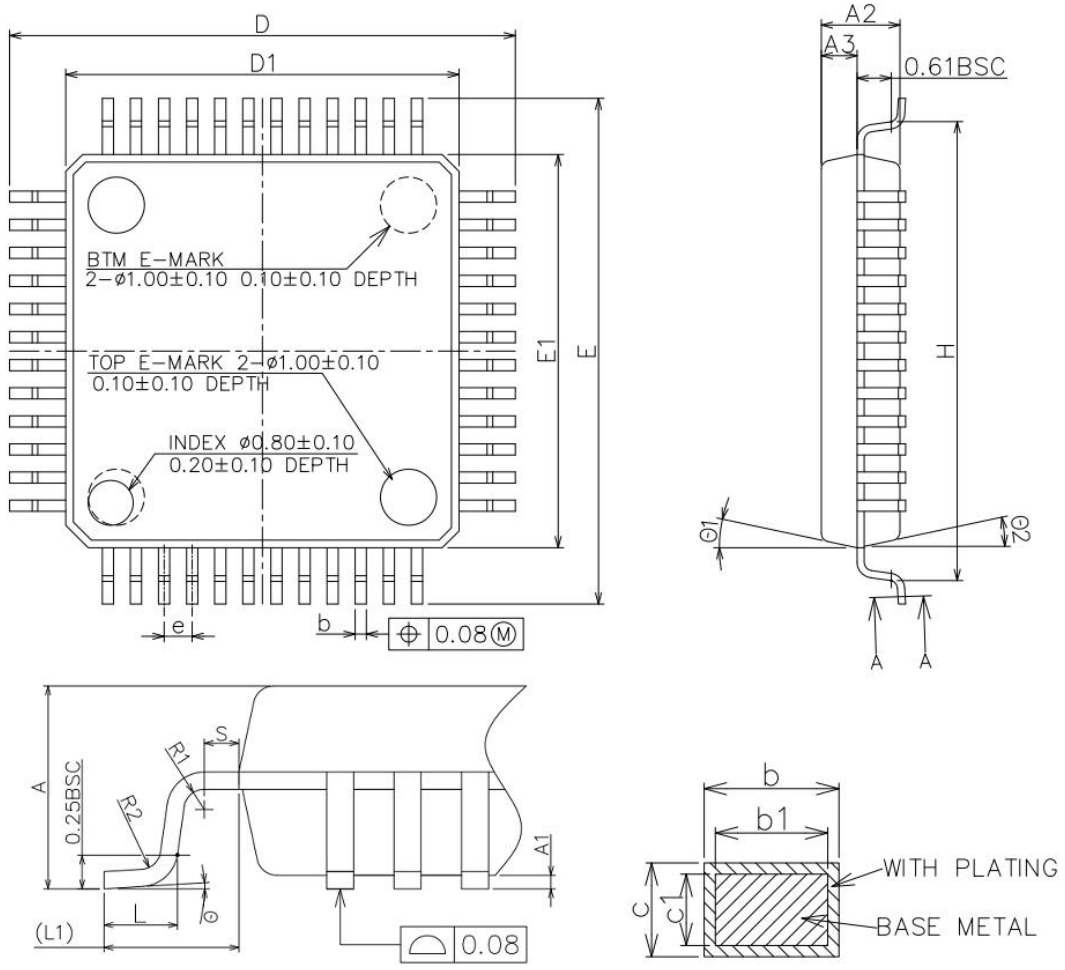


图 2-8 LQFP48 封装尺寸图

Symbol	MIN	NOM	MAX
A	–	–	1.6
A1	0.05	–	0.15
A2	1.35	1.4	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.2	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	8.8	9	9.2
D1	6.9	7	7.1
E	8.8	9	9.2
E1	6.9	7	7.1
e	0.50BSC		
L	0.45	0.6	0.75

Symbol	MIN	NOM	MAX
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.2
S	0.2	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

2.2.2 LQFP44

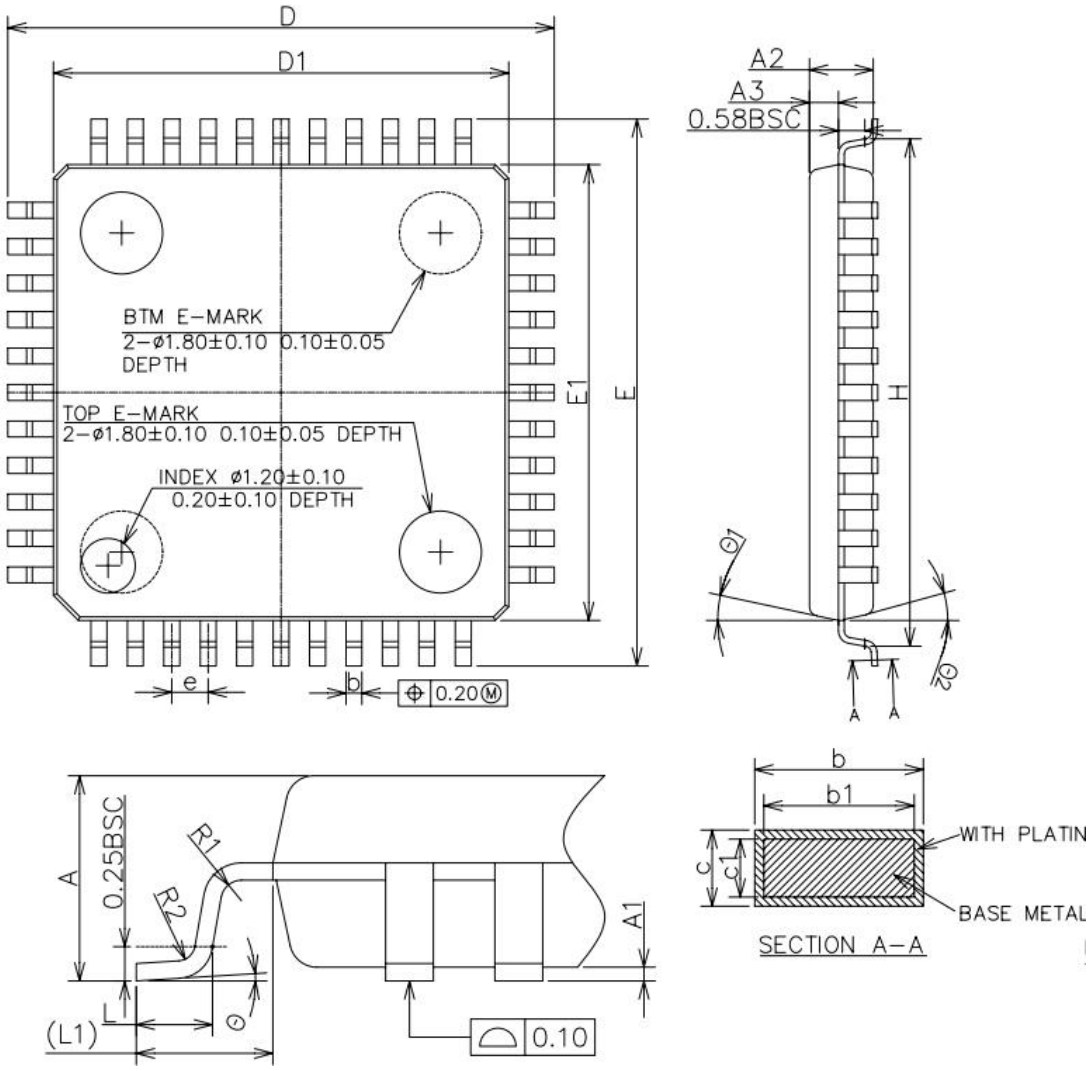


图 2-9 LQFP44 封装尺寸图

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

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

图 2-10 LQFP44 封装参数

2.2.3 LQFP32

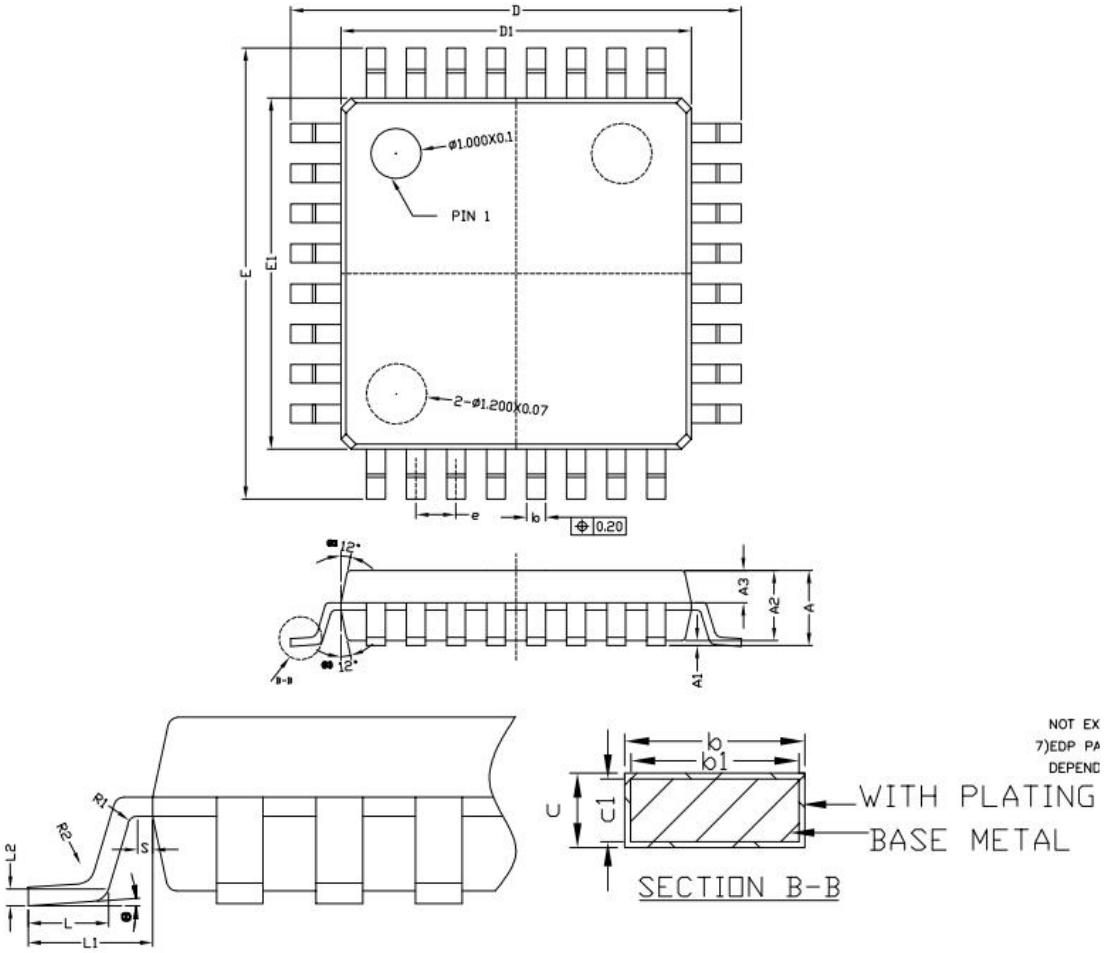


图 2-11 LQFP32 封装尺寸图

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

图 2-12 LQFP32 封装参数图

2.2.4 QFN32（5mmx5mm）

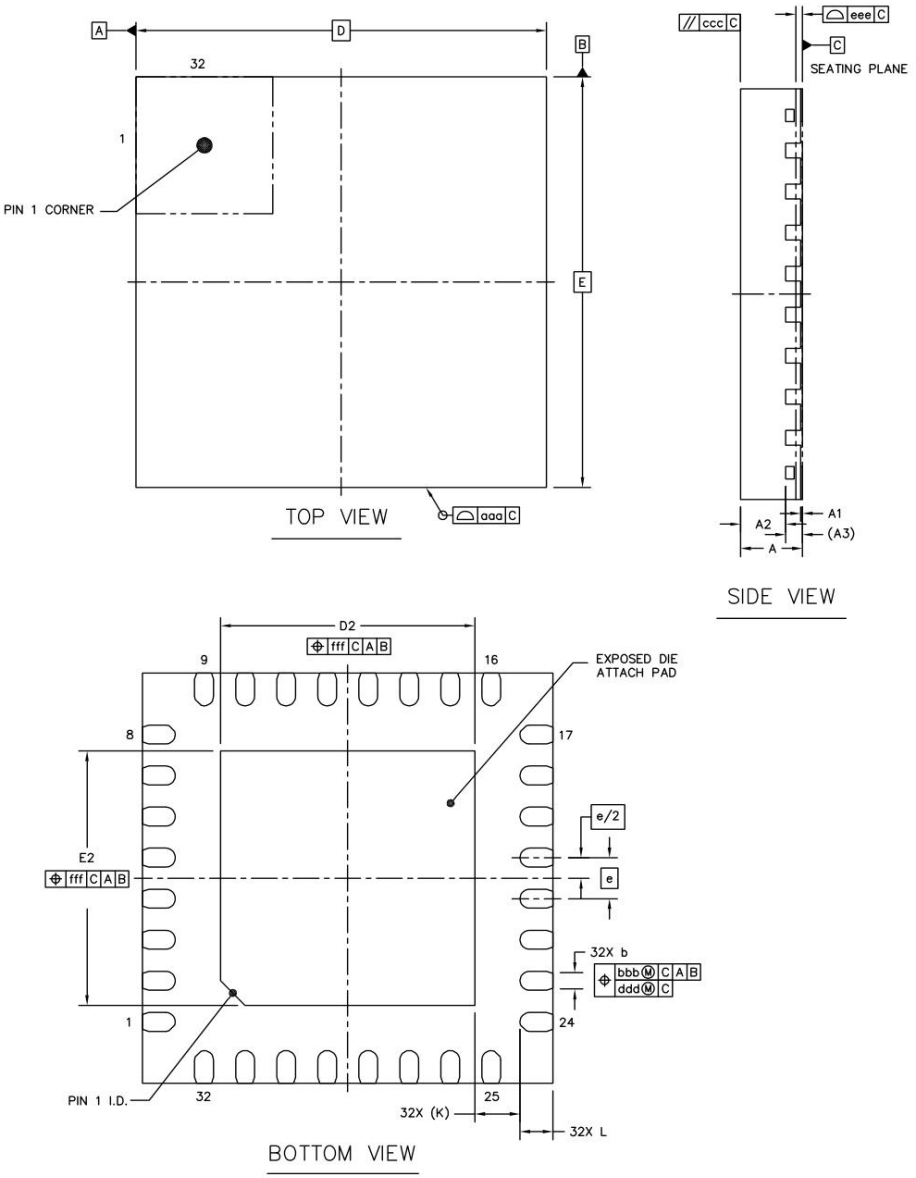


图 2-13 QFN32_5x5 封装尺寸图

		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D		5 BSC	
	Y	E		5 BSC	
LEAD PITCH		e		0.5 BSC	
EP SIZE	X	D2	3	3.1	3.2
	Y	E2	3	3.1	3.2
LEAD LENGTH		L	0.3	0.4	0.5
LEAD TIP TO EXPOSED PAD EDGE		K		0.55 REF	
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		ccc		0.1	
COPLANARITY		eee		0.08	
LEAD OFFSET		bbb		0.1	
		ddd		0.05	
EXPOSED PAD OFFSET		fff		0.1	

NOTES
1.REFER TO JEDEC MO-220;
2.COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD;
3.BAN TO USE THE LEVEL 1 ENVIRONMENT-RELATED SUBSTANCES OF JCET PRESCRIBING;
4.FINISH: Cu/EP·Sn8~20s

图 2-14 QFN32_5x5 封装参数

2.2.5 QFN32（4mmx4mm）

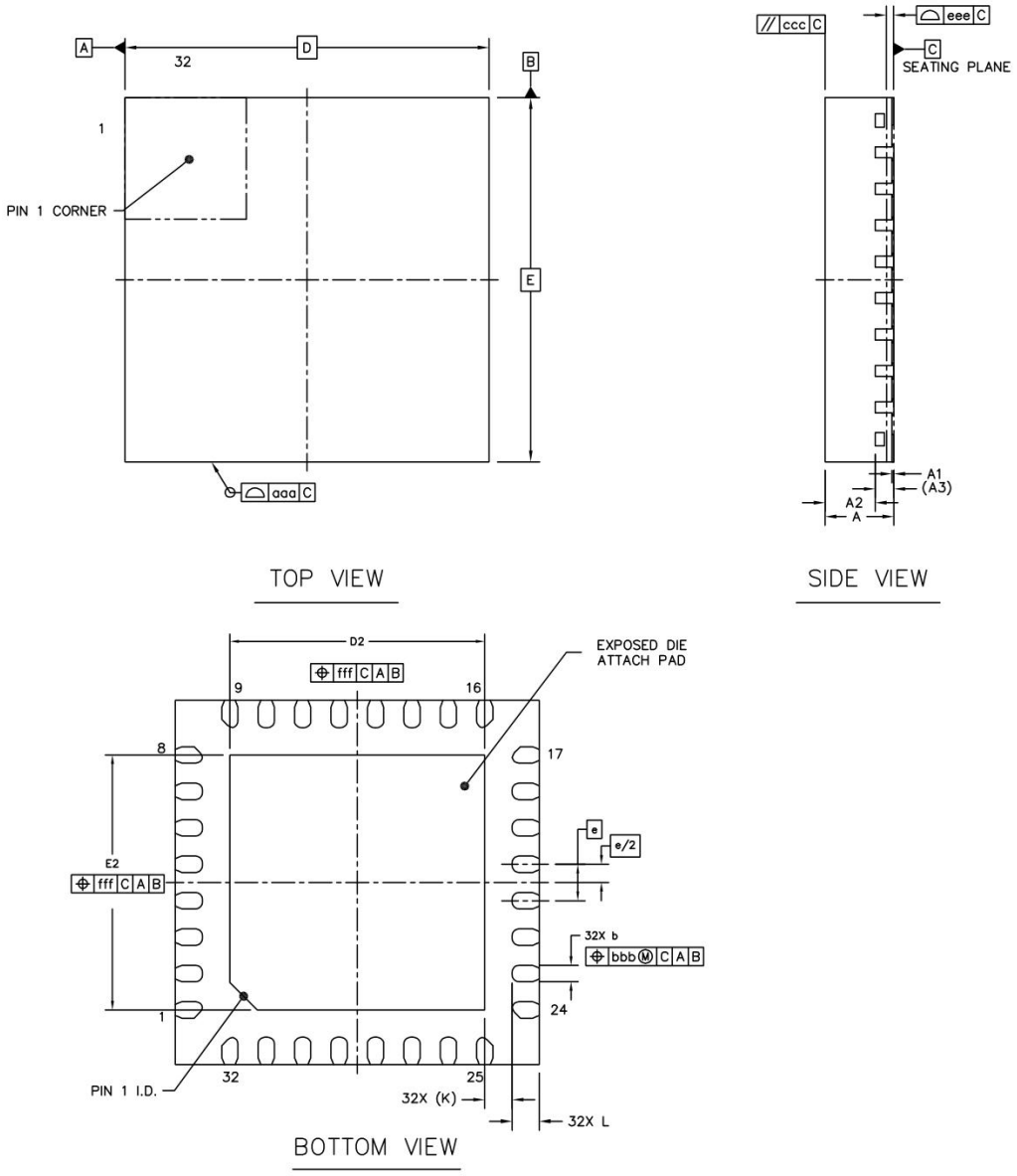


图 2-15 QFN32_4x4 封装尺寸图

		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D		4 BSC	
	Y	E		4 BSC	
LEAD PITCH		e		0.4 BSC	
EP SIZE	X	D2	2.7	2.8	2.9
	Y	E2	2.7	2.8	2.9
LEAD LENGTH		L	0.2	0.3	0.4
LEAD TIP TO EXPOSED PAD EDGE		K		0.3 REF	
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		ccc		0.1	
COPLANARITY		eee		0.08	
LEAD OFFSET		bbb		0.07	
EXPOSED PAD OFFSET		fff		0.1	

NOTES
1.REFER TO JEDEC MO-220;
2.COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD;
3.BAN TO USE THE LEVEL 1 ENVIRONMENT-RELATED SUBSTANCES OF JCET PRESCRIBING;
4.FINISH: Cu/EP·Sn8~20s

图 2-16 QFN32_4x4 封装参数

2.2.6 QFN20（3mmx3mm）

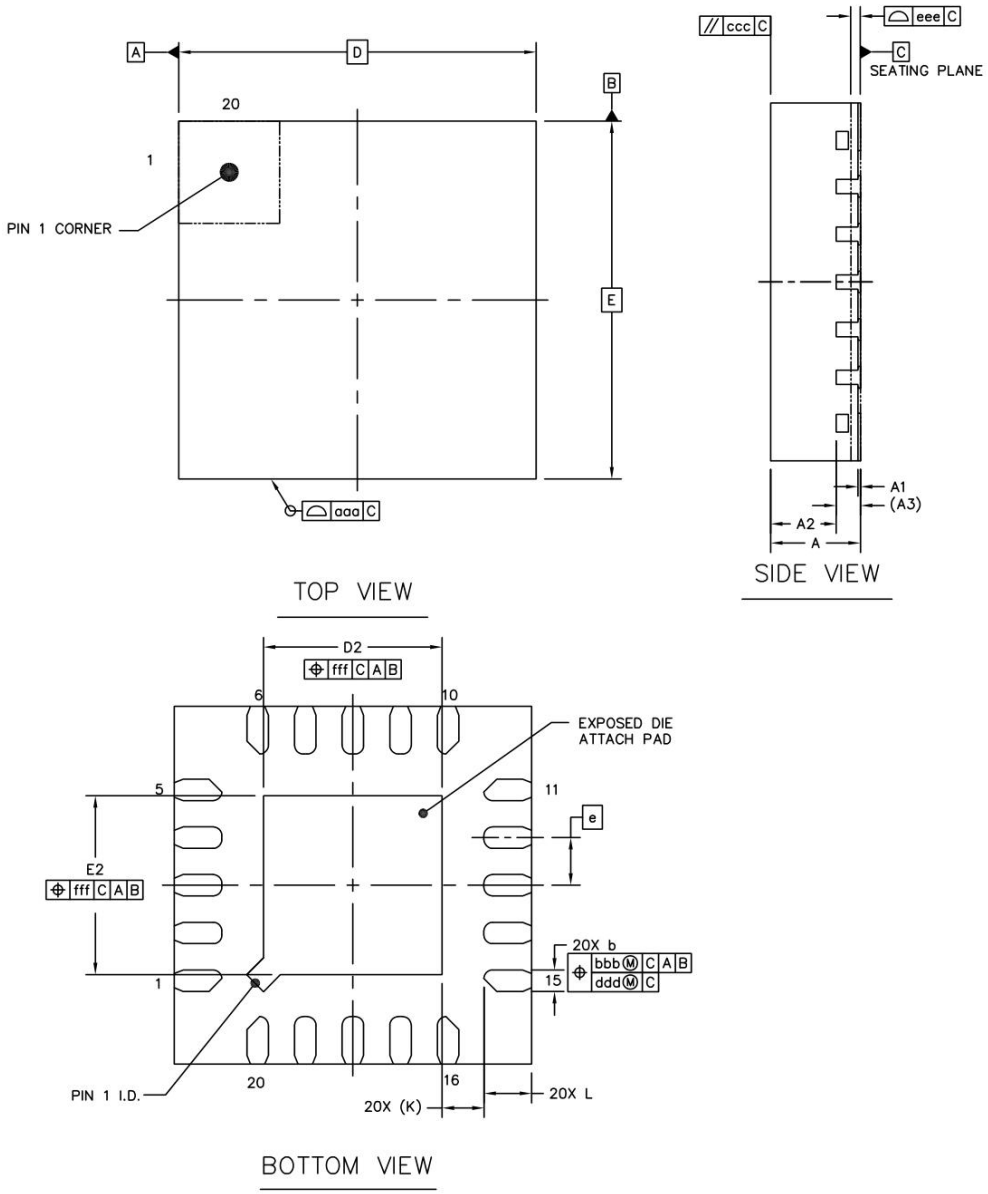


图 2-17 QFN20_3x3 封装尺寸图

		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D		3 BSC	
	Y	E		3 BSC	
LEAD PITCH		e		0.4 BSC	
EP SIZE	X	D2	1.4	1.5	1.6
	Y	E2	1.4	1.5	1.6
LEAD LENGTH		L	0.3	0.4	0.5
LEAD TIP TO EXPOSED PAD EDGE		K		0.35 REF	
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		ccc		0.1	
COPLANARITY		eee		0.08	
LEAD OFFSET		bbb		0.07	
		ddd		0.05	
EXPOSED PAD OFFSET		fff		0.1	

NOTES
1.REFER TO JEDEC MO-220;
2.COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD;
3.BAN TO USE THE LEVEL 1 ENVIRONMENT-RELATED SUBSTANCES OF JCET PRESCRIBING;
4.FINISH: Cu/EP · Sn8~20s

图 2-18 QFN20_3x3 封装参数

2.3 焊接安装说明

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

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

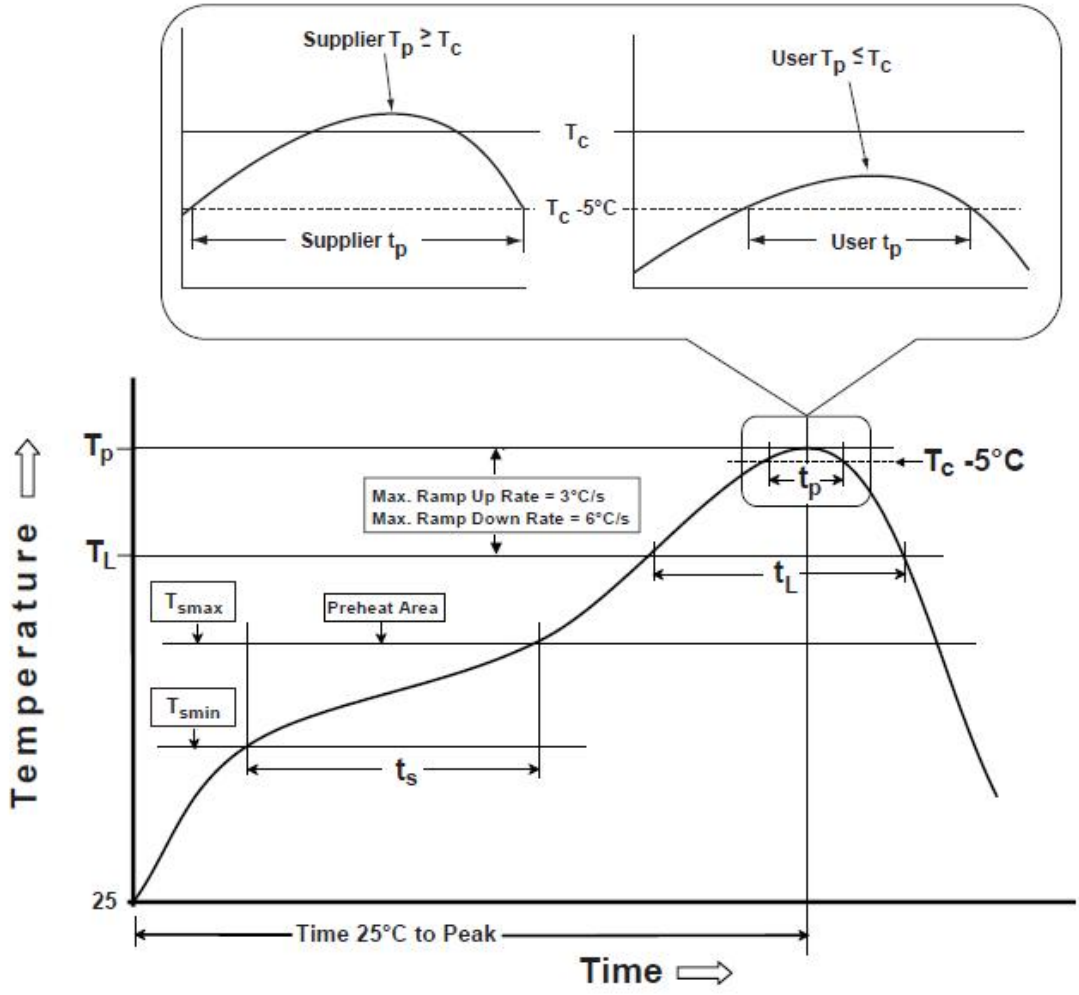
封装厚度	塑封体体积 mm^3 <350	塑封体体积 mm^3 350 - 2000	塑封体体积 mm^3 >2000
<1.6mm	260℃	260℃	260℃
1.6~2.5 mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

下表给出了各种封装形式的回流焊峰值温度：

封装类型	塑封体厚度 mm	塑封体体积 mm^3	回流焊峰值温度
LQFP80	1.4	201.6	260℃

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

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



特别声明：

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

2.4 MSL 等级

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

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 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。

表 3-1 芯片极限参数

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	VSS-0.3	6.5	V
T_A	工作温度	-40	105	℃
T_{STG}	存储温度	-55	150	℃
HBM	ESD HBM 模式 TA=25℃测试标准符合 JEDEC JS-001		+/-8000	V
CDM	ESD CDM 模式 TA=25℃测试标准符合 JEDEC JS-002		+/-1000	V
LU	IO Latchup-(0.5VDD) < VI < (1.5VDD)TA=25 ℃ 测试标准符合 JESD78E		+/-200	mA
$\sum I_{VDD}$	向芯片 VDD 流入的最大电流(source)		200	mA
$\sum I_{VSS}$	从芯片 VSS 流出的最大电流(sink)		200	mA
$\sum I_{IO}$	所有 IO sink 的最大总和电流		200	mA
	所有 IOsource 的最大总和电流		200	mA

3.3 性能参数

3.3.1 典型工作条件

表 3-2 芯片典型工作条件

Symbol	Parameter	Conditions	min	max	unit
f_{HCLK}	AHB 时钟频率	–	0	48	MHz
f_{PCLK}	APB 时钟频率	–	0	48	MHz
VDD	典型工作电压范围	–	2.2	5.5	V
T_J	结温	T_A=-40~+105C	-40	125	°C

3.3.2 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
- 高低温下电参数基于特征参数提取获得的均值加减 3sigma
- 除非特别声明，所有功耗数据在 VDD=5V 的条件下测试获得

3.3.2.1 Active 模式功耗

表 3-3 ACTIVE 电流参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
IDD_{RUN}	运行模式下的功耗，CPU 从 Flash 取指，CoremarkAll peripherals enabled	fAHB=16MHz (RCHF)PLL off Flash 0 wait	T_A=25℃	–	3.1	–	mA
			T_A=105℃	–	4.4	–	
		fAHB=48MHz (PLL)PLL on Flash 1 wait	T_A=25℃	–	5.6	–	
			T_A=105℃	–	6.5	–	

注：上表参数基于特征参数提取，不包含在量产测试中

3.3.2.2 SLEEP/DEEPSLEEP 模式功耗

表 3-4 SLEEP 电流参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
I_{sleep1}	Sleep 模式电流 1	SVD 关闭 CPU、RAM、外设数据保持	T_A=25℃	–	3	–	uA
			T_A=105℃	–	8.9	–	
$I_{dpsleep}$	Deepsleep 电流	SVD 关闭 CPU、RAM、外设数据保持	T_A=25℃	–	1.2	–	uA
			T_A=105℃	–	6.3	–	

3.3.3 复位和电源监控

芯片的复位和电源监控参数如下表。

表 3-5 复位和电源监控参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
t_{VDD}	电源上升速度			–		∞	us/V
	电源下降速度	SVD disabled		200		∞	us/V
		SVD enabled		20		∞	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			–	2.1	–	V
V_{PDR}	下电复位电压			–	1.9	–	V
V_{SVD}	电压监测阈值电平	SVD[3:0]=0000	Fall		2.4		V
			Rise		2.6		
		SVD[3:0]=0001	Fall		2.4		V
			Rise		2.6		
		SVD[3:0]=0010	Fall		2.4		V
			Rise		2.6		
		SVD[3:0]=0011	Fall		2.4		V
			Rise		2.6		
		SVD[3:0]=0100	Fall		2.7		V
			Rise		2.9		
		SVD[3:0]=0101	Fall		3		V
			Rise		3.2		
		SVD[3:0]=0110	Fall		3.6		V
			Rise		3.8		
		SVD[3:0]=0111	Fall		4		V
			Rise		4.2		

3.3.4 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： TA=30±1℃ VREFP=3.3V	0x1FFF_FD08

内部基准源主要参数：

表 3-6 内部基准电压参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
AV_{REF}	基准源输出电压	-40℃≤T_A≤105℃	0.97	1	1.03	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	
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℃	–	–	100	ppm/℃
V_{coeff}	内部基准源电压系数	2.2≤VDD≤5V	–	0.6	1	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		–	–	10	us

3.3.5 低功耗模式唤醒时间

表 3-7 唤醒时间参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
T_{WKU1}	Sleep 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}=16MHz	–	5.5	–	us

[1] 基于特征参数提取

典型唤醒事件波形图，仅供设计参考

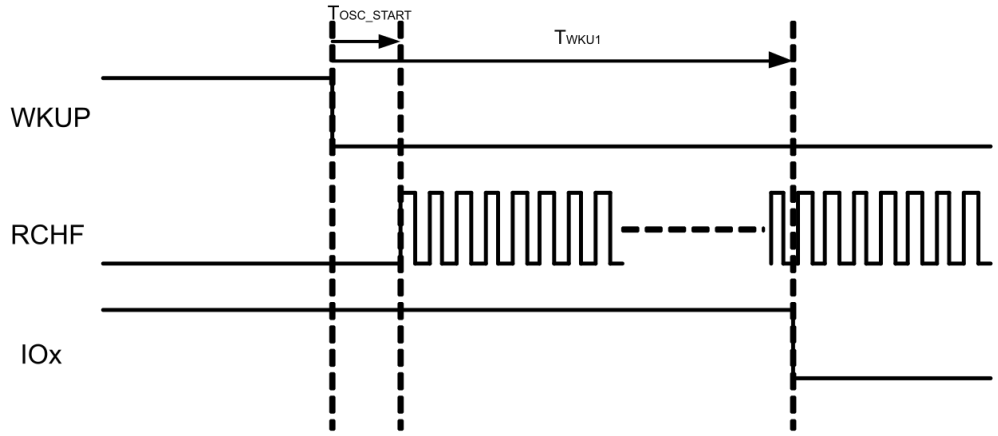


图 3-1 低功耗唤醒波形

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。

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

注意：以上时间评估使用 RCHF 8Mhz 为唤醒后的工作时钟。

3.3.6 外部时钟源特性

表 3-8 高频晶体振荡器参数

符号	参数说明	测试条件		参数值			单位
				最小值	典型值	最大值	
F_{XTHF}	XTHF 振荡频率[1]	VDD=5V		4	–	32	MHz
R_{fb}	反馈电阻[2]	–		–	200	–	KΩ
gm	跨导增益[2]	VDD=5V	Gmcfg =1000	15.8	17.04	–	mA/V
		VDD=3.3V		14.32	15.06	–	
IDD	晶振工作电流[1] VDD=5V	8MHz, CFG=1000		–	500	–	uA
		16MHz, CFG=1000		–	600	–	
		24MHz, CFG=1000		–	700	–	
T_{start1}	晶振 8M 起振时间[1]	VDD=5V, CFG=1000		–	0.5	–	ms
T_{start2}	晶振 16M 起振时间[1]	VDD=5V, CFG=1000		–	0.45	–	ms
T_{start3}	晶振 24M 起振时间[1]	VDD=5V, CFG=1000		–	0.4	–	ms
C_L	负载电容[2]	–		5	–	25	pF

注：

[1]指标基于特征参数提取

[2]指标基于仿真

[3]最低工作电压由设计保证到下电复位

3.3.7 内部时钟源特性

内部高频 RC 振荡器

表 3-9 内部高频 RC 振荡器参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{RCHF}	RCHF 振荡频率 VDD=2.2~5.5VTA=25C	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	31.7	32	32.2	
ACC_{RCHF}	全温区 RCHF 变化范围（相对 25 ℃ 频率的变	FSEL==2'b00 TA=-40~+105℃	-1	–	1	%
		FSEL==2'b01 TA=-40~+105℃	-2	–	2	%

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
	化）	FSEL==2'b10 TA=-40~+105℃	-3.5	–	3.5	%
		FSEL==2'b11 TA=-40~+105℃	-5	–	5	%

[注 1]：此项指标由量产测试保证

[注 2]：此项指标基于特征参数提取

内部低频 RC 振荡器

表 3-10 内部低频 RC 振荡器参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
f_{RCLP}[1]	RCLP 低功耗振荡频率	VDD=2.2~5.5VTA=25℃	31.68	32	32.32	KHz
ACC_{RCLP}[2]	全温区 RCLP 变化范围	VDD=2.2$_{5.5}$VTA=-$_{40}$+105℃	-5	–	5	%

[注 1]：此项指标由量产测试保证

[注 2]：此项指标基于特征参数提取

3.3.8 PLL 特性

表 3-11 PLL 参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
F_{PLL}	PLL 输出频率	VDD=2.2~5.5V TA=-40~105℃	32	–	48	MHz
t_{LOCK}	PLL 锁定时间			65		us

3.3.9 ADC 特性

3.3.9.1 参数说明

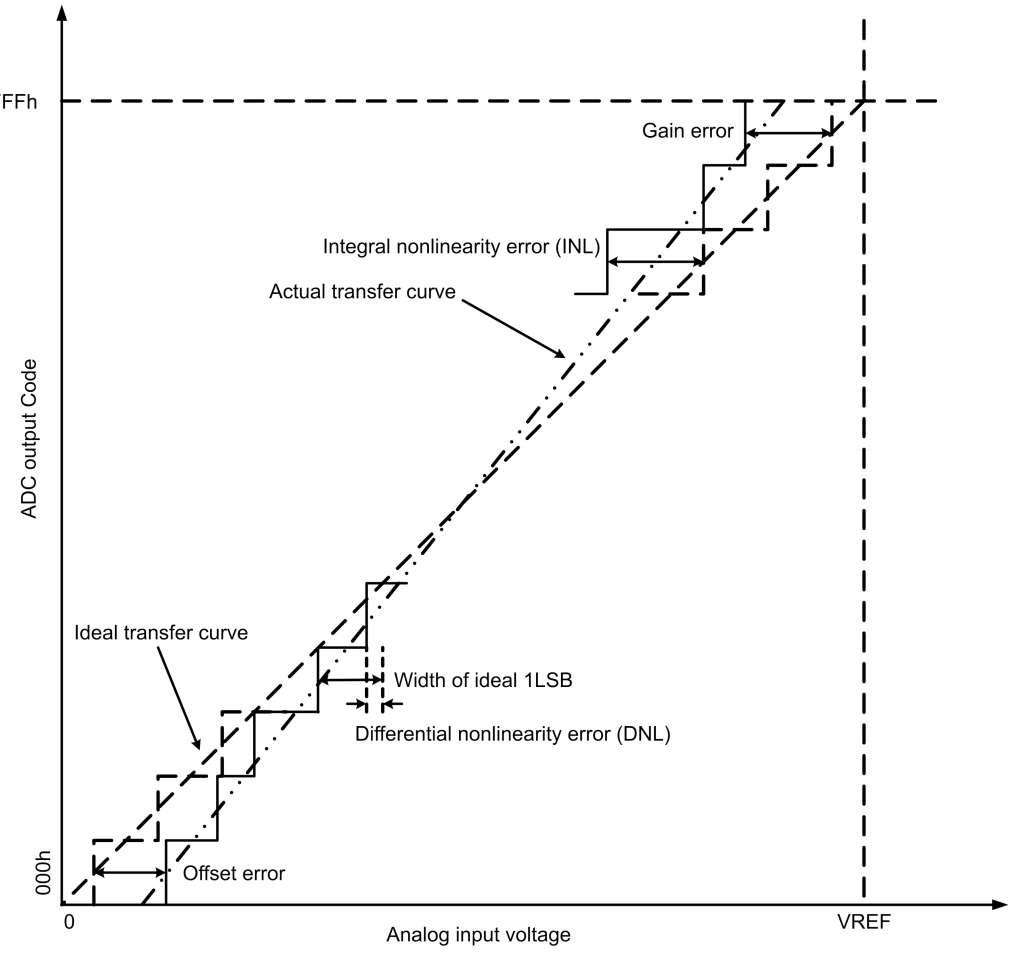


图 3-2 ADC 参数说明

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。

3.3.9.2 性能指标

表 3-12 ADC 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	
VDDA	工作电压范围	–	2.2		5.5	V
VREF+	正参考电压	–	1.5		VDDA	V
VREF-	负参考电压	–	0		0.5	V
T_J	工作结温范围	–	-40		125	°C
V_{AIN}	输入电压范围		VREF-		VREF+	V
C_s	采样保持电容	–		5		pF
R_{AIN}	模拟输入阻抗	VDDA=3V		1.6		KΩ
F_{CLK}	ADC 工作时钟频率	–			32	MHz
F_S	ADC 采样频率				1	Msps
T_{SAMP}	采样保持时间	–	4		512	F_{CLK}
T_{CONV}	转换时间	–		12		F_{CLK}
ADC 动态性能						
ENOB	VDD=3.3V VREF+=VDDA F_S=1Msps -40 ℃≤T_A≤85℃	单端模式 F_{AIN}=29KHz		11		bits
	VDD=5V VREF+=VDDA F_S=1Msps -40 ℃≤T_A≤85℃	单端模式 F_{AIN}=29KHz		11.2		bits
SFDR	无杂散动态范围 VDD=5V VREF+=VDDA F_S=1Msps F $_{AIN}$=1KHz -40℃≤T $_A$≤85℃	单端模式 F_{AIN}=29KHz		76		dB
ADC 静态性能						
DNL	差分非线性	VDD=3.3v/5V	-1	–	1	LSB
INL	积分非线性	VDD=3.3v/5V	-1	–	1	LSB
OffsetError	失调误差	VDD=3.3v/5V	-5	–	5	LSB
GainError	增益误差	VDD=3.3v/5V	-5	–	5	LSB

3.3.10 温度传感器

芯片出厂时经过温度定标，定标条件是 VDDA=3.0V，TA=30+/-1° C。在此条件下，使用 ADC 采样并转换温度传感器输出电压，将转换结果保存在 Flash 指定地址。

表 3-13 温度传感器定标结果保存

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	0x1FFFFD12

表 3-14 温度传感器参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
Reso	分辨率	VDDA=3.0V		0.3		°C
Slope	输出斜率[1]			3.02		mV/°C
Linerity	全温区线性度[1]	T=30~85C	–	+/-3	+/-5	°C
I_{DDA}	温度传感器功耗（不含 ADC）[2]	VDDA=3.0V		1		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

[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。

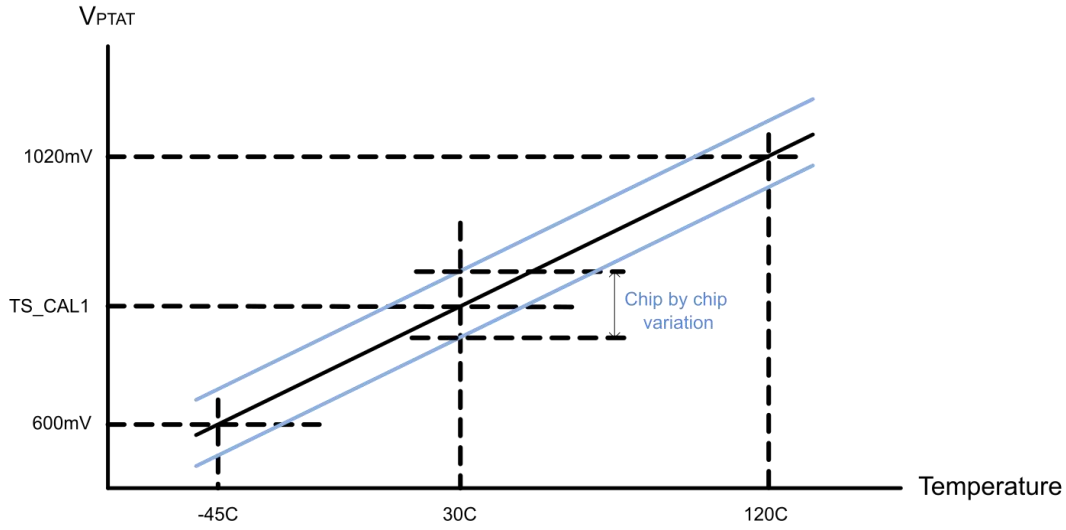


图 3-3 温度传感器输出曲线

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。

3.3.11 模拟比较器特性

表 3-15 模拟比较器参数

符号	参数说明	测试条件	参数值	单位

			最小值	典型值	最大值	
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		8		uA
$T_{propagation1}$	比较器传播延迟	VDDA=5V 200mV step 100mV overdrive		300		ns
T_{setup1}	比较器建立时间	VDDA=5V		300		ns

3.3.12 Flash 存储器特性

表 3-16 Flash 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	
	Flash size			–	128K	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		100000			Erase/ Write cycles
T_{DR}	Data Retention	T_A=85℃average	10			yrs

3.3.13 AC 参数

3.3.13.1 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

表 3-17 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	31.25	–	–	ns
tr(SCK)		Master mode	–	3	–	ns
tf(SCK)		Master mode	–	3	–	ns

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
tsu(SSN)		Slave mode	1	–	–	ns
th(SSN)		Slave mode	1	–	–	ns
tsu(MI)	Data input setup time	Master mode	3.5	–	–	ns
tsu(SI)		Slave mode	3	–	–	ns
th(MI)	Data input hold time	Master mode	1	–	–	ns
th(SI)		Slave mode	1	–	–	ns
ta(SO)	Data output access time	Slave mode	25	–	–	ns
tdis(SO)	Data output disable time	Slave mode	1	–	–	ns
tv(SO)	Data output valid time	Slave mode	–	–	25	ns
tv(MO)		Master mode	–	–	3	ns
th(SO)	Data output hold time	Slave mode	25	–	–	ns
th(MO)		Master mode	1	–	–	ns

表 3-18 SPI timing characteristics at VDD=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	31.25	–	–	ns
tr(SCK)		Master mode	–	3	–	ns
tf(SCK)		Master mode	–	3	–	ns
tsu(SSN)		Slave mode	1	–	–	ns
th(SSN)		Slave mode	1	–	–	ns
tsu(MI)	Data input setup time	Master mode	3.5	–	–	ns
tsu(SI)		Slave mode	3	–	–	ns
th(MI)	Data input hold time	Master mode	1	–	–	ns
th(SI)		Slave mode	1	–	–	ns
ta(SO)	Data output access time	Slave mode	25	–	–	ns
tdis(SO)	Data output disable time	Slave mode	1	–	–	ns
tv(SO)	Data output valid time	Slave mode	–	–	25	ns
tv(MO)		Master mode	–	–	3	ns
th(SO)	Data output hold time	Slave mode	25	–	–	ns
th(MO)		Master mode	1	–	–	ns

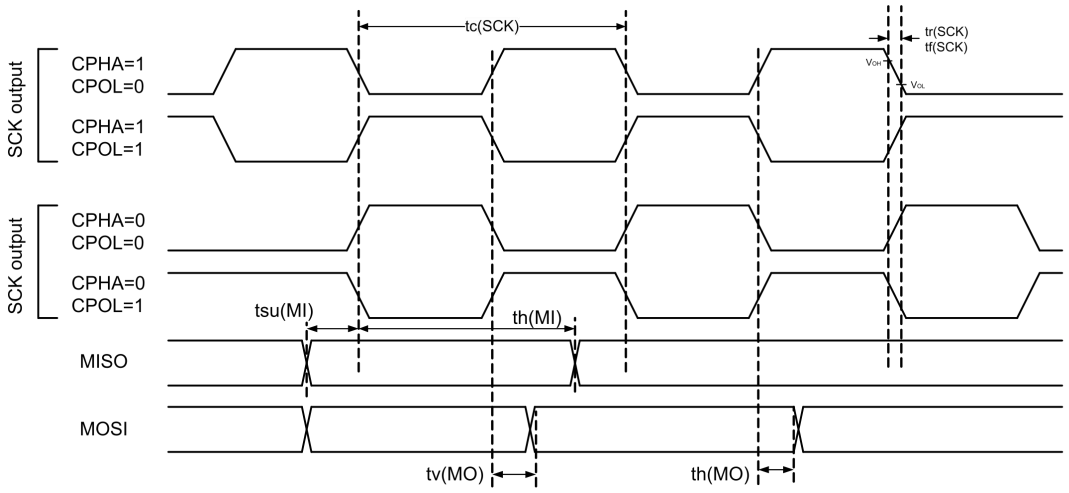


图 3-4 SPI master timing

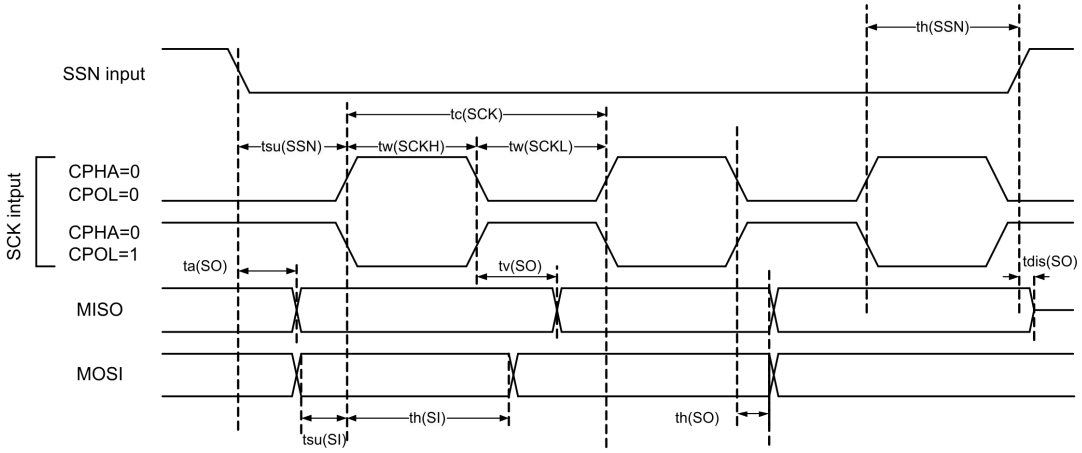


图 3-5 SPI slave timing, CPHA=0

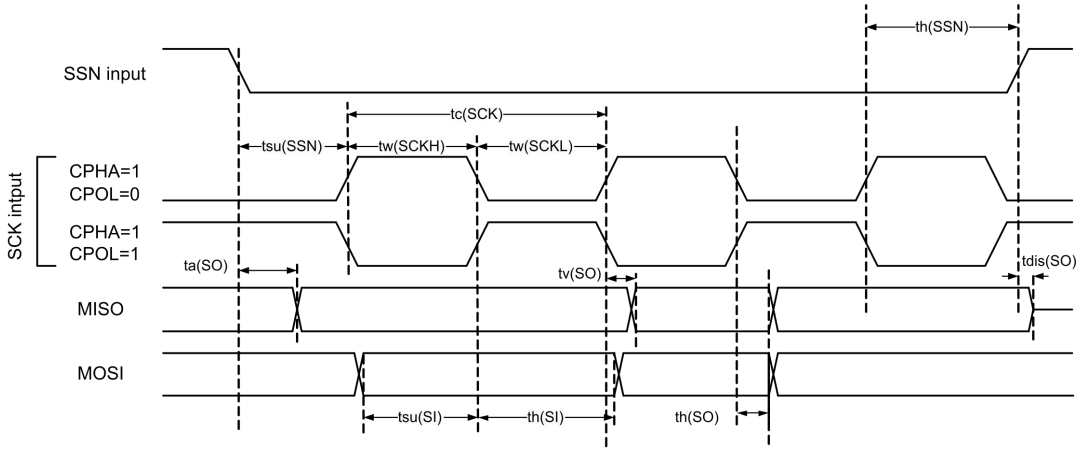


图 3-6 SPI slave timing, CPHA=1

3.3.13.2 I²C

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	—	0.6	—	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	—	0.6	—	us
数据输入建立时间	$T_{SU:DAT}$	250	—	100	—	ns
数据输入保持时间	$T_{HD:DAT}$	5	—	—	—	us
		0	3.45	0	0.9	us
SDA 和 SCL 上升时间	T_R	—	1000	—	300	ns
SDA 和 SCL 下降时间	T_F	—	300	—	300	ns
停止条件建立时间	$T_{SU:STO}$	4	—	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

3.3.14 GPIO 特性

普通 IO

表 3-19 普通 IO 参数

符号	参数说明	测试条件	参数值			单位
			最小值	典型值	最大值	
V_{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	-1		1	μA
I_{IH}	输入高漏电	V_{IH}=5V	-1		1	μA
V_{OL}	输出低电平	VDD=5V ISINK=10mA 强驱动		0.23		V
		VDD=5V ISINK=20mA 强驱动		0.46		
V_{OH}	输出高电平	VDD=5V ISOURCE=10mA 强驱动		4.75		V
		VDD=5V ISINK=20mA 强驱动		4.5		
R_{PU}	弱上拉电阻			100		KΩ

GPIO AC 特性

表 3-20 引脚 AC 参数

IO	符号	参数说明	测试条件		typ	max	单位
GPIO	Fmax	Maximum frequency	C_L=15pF, VDD=5V	SR=0,DR=0	–	48	MHz
				SR=0,DR=1	–	48	
				SR=1,DR=0	–	48	
				SR=1,DR=1		48	
			C_L=15pF, VDD=3.3V	SR=0,DR=0	–	48	
				SR=0,DR=1	–	48	
				SR=1,DR=0		48	
				SR=1,DR=1	–	48	
			C_L=50pF, VDD=5V	SR=0,DR=0	–	48	
				SR=0,DR=1	–	40	
				SR=1,DR=0		48	
				SR=1,DR=1	–	40	
			C_L=50pF, VDD=3.3V	SR=0,DR=0	–	48	
				SR=0,DR=1	–	35	
				SR=1,DR=0		48	
				SR=1,DR=1	–	35	
	Tr	Output rise time	C_L=15pF, VDD=5V	SR=0,DR=0	0.8	–	ns
				SR=0,DR=1	0.992	–	
				SR=1,DR=0	0.88		
				SR=1,DR=1	1.088	–	
			C_L=15pF, VDD=3.3V	SR=0,DR=0	0.96	–	
				SR=0,DR=1	1.264	–	
				SR=1,DR=0	1.104		
				SR=1,DR=1	1.392	–	
			C_L=50pF, VDD=5V	SR=0,DR=0	1.792	–	
				SR=0,DR=1	3.76	–	
				SR=1,DR=0	1.872		
				SR=1,DR=1	3.952	–	
			C_L=50pF, VDD=3.3V	SR=0,DR=0	2.816	–	
				SR=0,DR=1	5.056	–	
				SR=1,DR=0	2.928		
				SR=1,DR=1	5.168	–	
	Tf	Output fall time	C_L=15pF, VDD=5V	SR=0,DR=0	0.752	–	ns
				SR=0,DR=1	0.944	–	
				SR=1,DR=0	0.832		
				SR=1,DR=1	1.024	–	
			C_L=15pF, VDD=3.3V	SR=0,DR=0	0.896	–	
				SR=0,DR=1	1.152	–	
				SR=1,DR=0	1.008		
				SR=1,DR=1	1.328	–	
			C_L=50pF, VDD=5V	SR=0,DR=0	1.648	–	
				SR=0,DR=1	3.696	–	
				SR=1,DR=0	1.872		
				SR=1,DR=1	3.92	–	
			C_L=50pF, VDD=3.3V	SR=0,DR=0	2.576	–	
				SR=0,DR=1	5.056	–	
				SR=1,DR=0	2.72		
				SR=1,DR=1	5.088	–	

注:参数基于特征参数提取。

4 总线与存储

4.1 系统总线

本芯片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

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

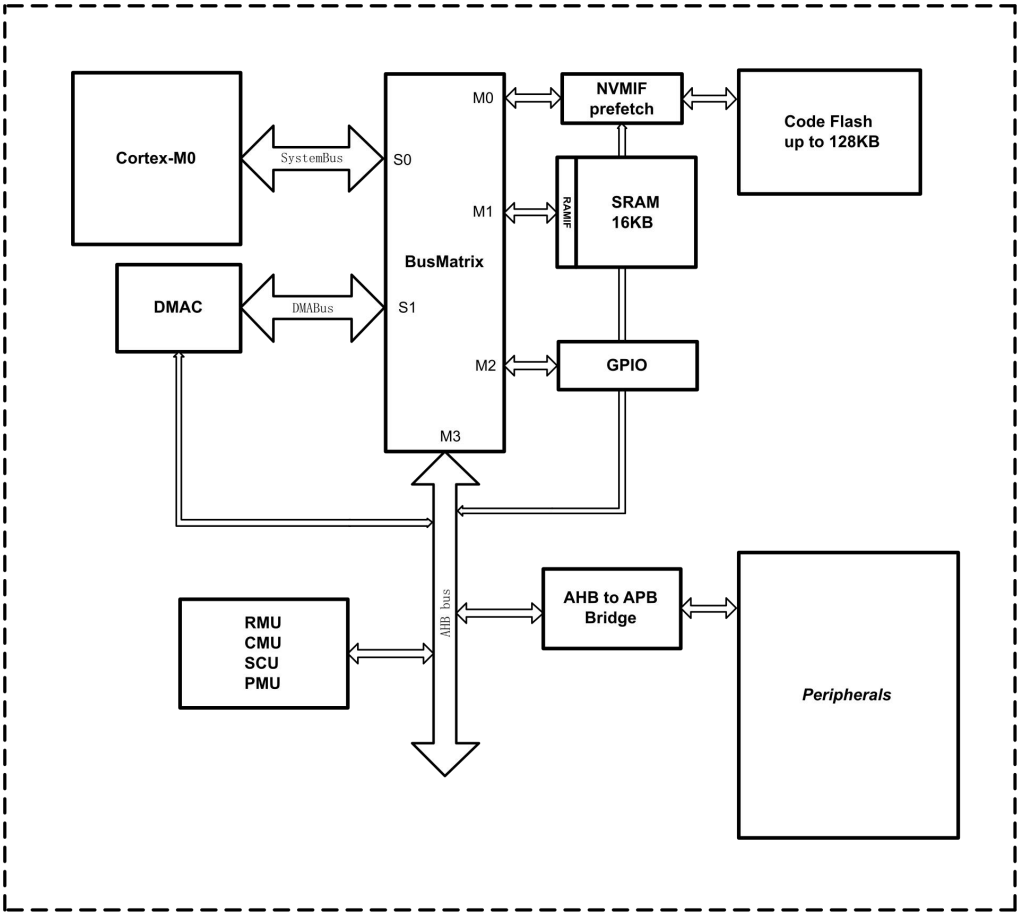


图 4-1 系统总线示意图

4.2 存储空间分配

4.2.1 概述

当芯片从 Flash 启动时，本芯片的地址空间分配如下图（128KB Flash，16KB RAM）：

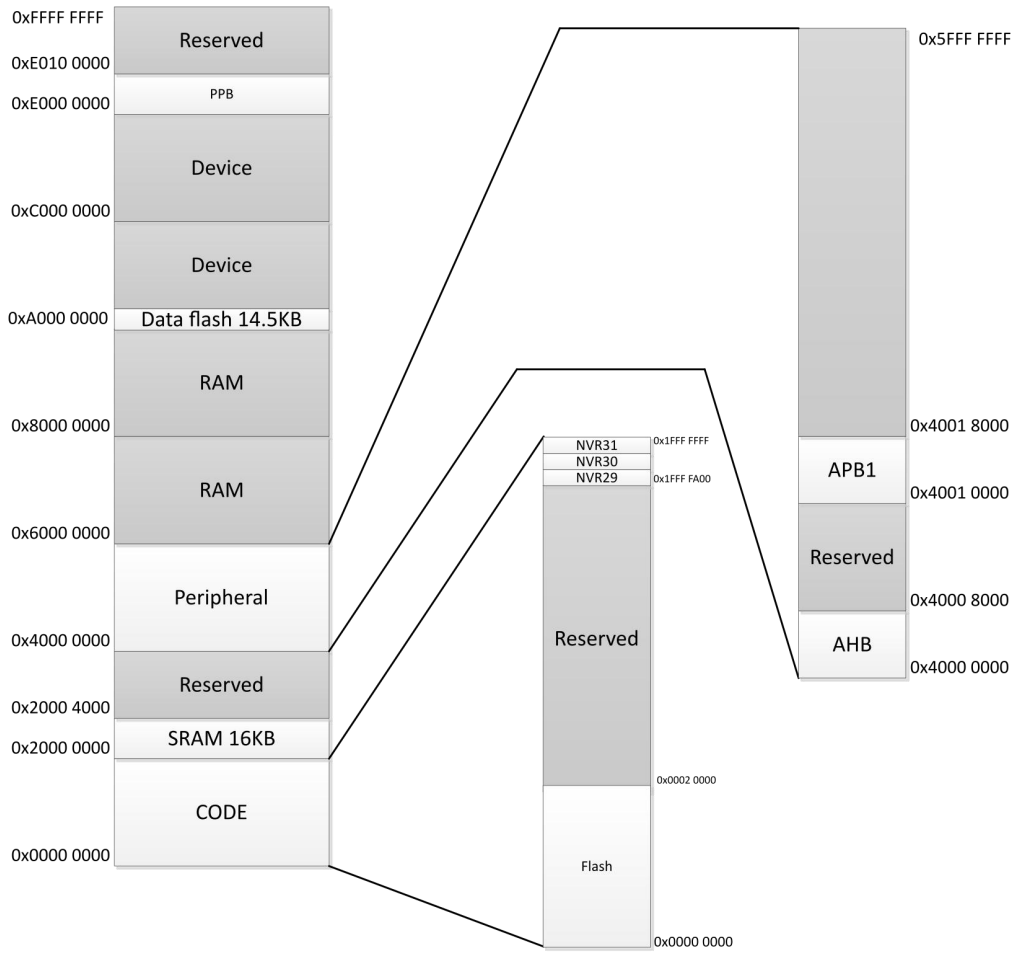


图 4-2 FM33LR0xx 总线地址（128+16）

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。

表 4-1 外设模块总线地址列表

总线	起始地址	空间	外设
AHB	0x0000_0000	128KB	Flash main array
	0x1FFF_F000	4KB	
	0x2000_0000	16KB	SRAM
	0x2000_8000	–	–
	0x4000_0000	1KB	SCU

总线	起始地址	空间	外设
	0x4000_0400	1KB	DMA
	0x4000_0800	1KB	RAMBIST
	0x4000_0C00	1KB	GPIO
	0x4000_1000	1KB	NVMIF
	0x4000_1400	–	–
	0x4000_1800	–	–
	0x4000_1C00	1KB	SFU
	0x4000_2000	1KB	PMU
	0x4000_2400	1KB	CMU
	0x4000_2800	1KB	RMU
	0x4001_0000	64KB	APB
APB	0x4001_0000	1KB	
	0x4001_0400	1KB	
	0x4001_0800	1KB	
	0x4001_0C00	1KB	
	0x4001_1000	1KB	
	0x4001_1400	1KB	IWDT
	0x4001_1800	1KB	WWDT
	0x4001_1C00	1KB	
	0x4001_2000	1KB	UART1
	0x4001_2400	1KB	
	0x4001_2800	1KB	
	0x4001_2C00	1KB	
	0x4001_3000	1KB	HDIV
	0x4001_3400	1KB	LPTIM16
	0x4001_3800	1KB	PGL
	0x4001_3C00	1KB	TAU
	0x4001_4000	1KB	
	0x4001_4400	1KB	
	0x4001_4800	1KB	
	0x4001_4C00	1KB	GPTIM
	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	
	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

总线	起始地址	空间	外设
	0x4001_9C00	1KB	COMPARATOR
	0x4001_A000	1KB	UART4
	0x4001_A400	1KB	UART5
	0x4001_A800	1KB	SVD
	0x4001_AC00	1KB	ADC
	0x4001_B000	1KB	ATIM
	0x4001_B400	1KB	BSTIM16
	0x4001_B800	1KB	
	0x4001_BC00	1KB	

4.3 RAM

4.3.1 概述

本芯片包含 16KB RAM（带 parity）

4.3.2 奇偶校验

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

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

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

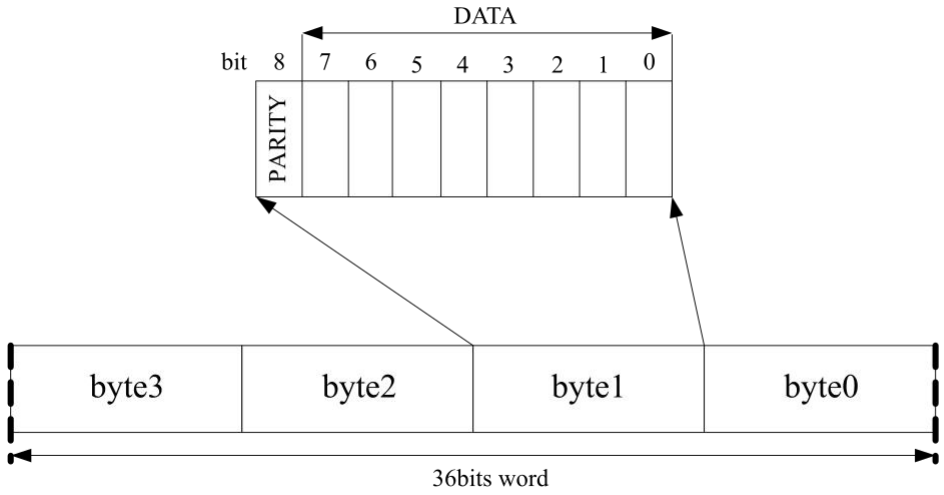


图 4-3 RAM 奇偶校验

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

本芯片使用的 Flash 容量为 128KB，阵列组织格式包含 page(512B)、sector(2KB)、mat(128KB)。

4.4.2 程序 flash

Mainarray128KB 全部为程序空间，强制 ECC 使能，由于 ECC 原因只支持 32bit 编程。

4.4.3 数据 flash

Flash 自带的 32 个 NVR 扇区中，除了 NVR30、NVR31 保留，NVR29 为选项字节，其余 29 个 NVR 扇区全部开放给用户作为数据 flash 使用，总可用容量为 14.5KB。

由于 NVR 中保存的数据不带 ECC 校验，因此数据 flash 也没有 ECC，但是支持 parity；对用户来说，数据 flash 支持字节编程。

Flash 全擦操作不会影响 NVR 扇区，因此数据 flash 内容不受全擦影响。

4.4.4 ECC

Flash ECC 中断：

- Code ECC detection 中断：main array 检测到 2bit 错误，无法纠错，中断可屏蔽
- Code ECC correction 中断：main array 纠正 1bit 错误，不影响正常运行，中断可屏蔽
- Dataflash parity 中断：NVR 区域检测到 parity 错误
- 错误产生时，ADDR_ECC 寄存器记录错误发生的 flash word 地址

4.4.5 NVR 扇区说明

Flash IP 共有 16 个特殊扇区，说明如下

区域	说明	用途
NVR31	Flash core trim 区	HHG 测试使用
NVR30	FMSH 数据区	保存 FMSH 的调校信息、模式字、测试数据等；软件只读
NVR29	用户选项数据区	用户选项字节（OPTBYTES）
NVR28~0	Data flash	作为 data flash 使用

4.4.5.1 NVR29

NVR29 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）。

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FA00	~OPTBYTES[15:0]	OPTBYTES[15:0]	用户选项字节低半字
0x1FFF_FA04	~OPTBYTES[31:16]	OPTBYTES[31:16]	用户选项字节高半字
0x1FFF_FA08	LOCK1	LOCK1	ACLOCK 配置字，控制最低 128KB flash
0x1FFF_FA0C			

AHB addr	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FA10			

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 其他：使能 IWDT 0x55：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
7:0	DBRDPEN	调试接口访问保护使能 0xAA：关闭调试接口保护 其他：使能调试接口保护	0xAA

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

LOCK 配置字节定义如下：

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

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]

Address	LOCK bits
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]

4.4.5.2 NVR28~0

这 29 个 NVR 扇区开放给用户作为数据区使用，SWD 和软件均可以读写，共 14.5KB。

4.4.6 Flash 擦除与编程

4.4.6.1 概述

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 支持的擦除操作：全擦、扇区擦

4.4.6.2 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。

擦写时钟独立于 CPU 时钟，两者之间作为异步时钟处理。

4.4.6.3 Flash 擦写方法

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：

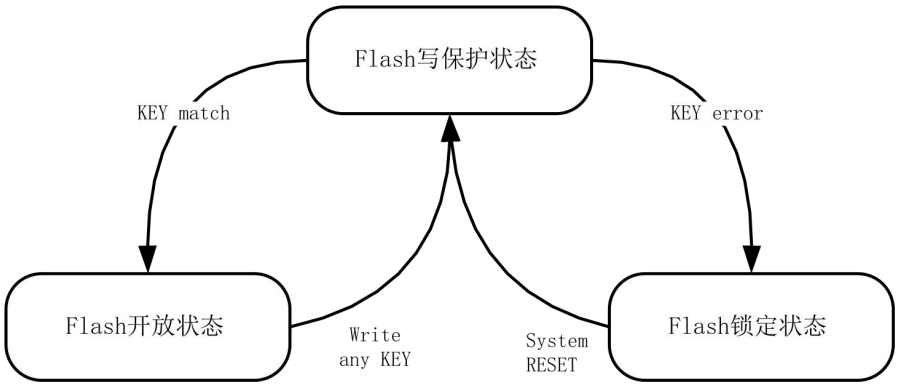


图 4-4 Flash 编程状态跳转图

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。

4.4.6.4 全擦操作（Chip 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 main array，不会擦除 NVR

SWD 可以在制造商或用户模式下启动全擦，操作流程如下：

- 编程器通过 SWD 配置 ERTYPE 寄存器为 10
- 编程器通过 SWD 清除 PREQ 寄存器，置位 EREQ 寄存器
- 编程器通过 SWD 写入 Flash 全擦 Key：0x9696_9696 和 0x7D7D_7D7D
- SWD 向 Flash 任意地址写擦除请求 0x1234_ABCD
- NVMIF 启动对 Flash 的全擦，并暂停任何 Master 对 Flash 的访问
- 全擦完成后置位中断标志和全擦标志（全擦标志表示 main array 全部擦除，任何对 main array 的编程将清除此标志）
- 在全擦标志有效的情况下，SWD 可以任意擦写 NVR，否则擦写 NVR 被禁止并触发错误中断
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

注意：全擦操作只擦除 Flash 的 main array，不会影响 NVR

4.4.6.5 扇区擦操作（Block Erase）

SWD 和应用代码都可以执行扇区擦，擦除单位 2K 字节。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 Block 内任意地址（以 2KB 为单位）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

4.4.6.6 页擦操作（Sector Erase）

SWD 和应用代码都可以执行扇区擦，擦除单位 512 字节。操作流程如下：

- 配置 ERTYPE 寄存器为 00
- 清除 PREQ 寄存器，置位 EREQ 寄存器
- 写入 Flash 块擦 Key：0x9696_9696 和 0xEAEA_EAEA
- 向需要擦除的 Page 内任意地址写擦除请求 0x1234_ABCD
- NVMIF 检查目标扇区是否属于被 ACLOCK 锁定的 Block，如果没有锁定则启动对目标扇区的擦除，如果被锁定则触发错误标志
- 扇区擦完成后置位中断标志
- 软件确认擦除结束后向 FlashKEY 寄存器写任意值恢复写保护

4.4.6.7 单次编程

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

- 清除 EREQ 寄存器，置位 PREQ 寄存器
- 清除连续编程使能寄存器
- 写入 Flash 编程 Key：0xA5A5_A5A5 和 0xF1F1_F1F1

- 向 Flash 目标地址写数据，如果目标地址被 ACLOCK 锁定，则触发错误标志，如果没有锁定，则执行编程
- 编程完成后置位中断标志，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。

4.4.6.8 连续编程

连续编程指通过 DMA 的 Memory 通道一次性向 Flash 写入多个 word，固定为 32 个 word 的数据，不允许跨 row 编程。

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
- 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。

4.4.7 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 NVR 中的 OPTBYTES 来控制。

4.4.7.1 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 由用户配置信息区内的 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。

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。

4.4.7.2 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 10 时，此 Block 处于写保护状态；当对应 LOCK 位为 01 时，此 Block 处于读写保护状态；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 NVR 区中配置关闭 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
- 写保护：指定 Block 允许 CPU 取指和读取，不允许 CPU 和 DMA 擦写，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	软件和 DMA 读取	软件取指	软件和 DMA 擦写	SWD 读取和擦写
11	允许	允许	允许	允许
10	允许	允许	禁止	允许
01	禁止	允许	禁止	允许
00	禁止	允许	禁止	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写 0，但是不能写 1（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。

4.4.7.3 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	0	x	–	F
		1	x	–	F

Flash area	DBRDP	LOCK bits (per Block)[3]	Last byte in page	SWD	Application
		10	x	–	R/F
		11	x	–	R/E/W/F
	OFF	0	x	–	F
		1	x	R/E/W	F
		10	x	R/E/W	R/F
		11	x	R/E/W	R/E/W/F
DCT	x	x	x	R	R
LDT0					
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF3	x	x	x	R/E/W	R/E/W
IF2,1,0	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

FLS 基地址 (0x40001000)

地址	符号	名称
0x0000	FLS_RDCR	Flash 读取控制寄存器
0x0004	FLS_PFTCON	Flash 预取控制寄存器
0x0008	FLS_OPTBR	用户配置字寄存器
0x0014	FLS_EPCR	Flash 擦写控制寄存器
0x0018	FLS_KEY	Flash Key 输入寄存器
0x001C	FLS_IER	Flash 中断使能寄存器
0x0020	FLS_ISR	Flash 标志寄存器
0x0024	FLS_ERRIER	FLASH 检错中断使能寄存器
0x0028	FLS_ECCSR	Flash ECC 寄存器
0x0060	FLS_ACLOCK1	ACLOCK 寄存器 1

4.5.1 Flash 读取控制寄存器(FLS_RDCR)

符号	FLS_RDCR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	FLS_RDCR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						WAIT	
权限	R-000000						RW-00	

位	位名	权限	功能描述
31:2	RFU	R	未实现：读为 0
1:0	WAIT	RW	Flash 读等待周期配置 00：0 wait cycle 01：1 wait cycle 10：2 wait cycles 11：3 wait cycles

4.5.2 Flash 预取控制寄存器(FLS_PFTCON)

符号	FLS_PFTCON							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							PFTPHASE_EN
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU	R	未实现：读为 0
0	PFTPHASE_EN	RW	指令预取指使能 1：使能 Prefetch 0：禁止 Prefetch

4.5.3 用户配置字寄存器(FLS_OPTBR)

符号	FLS_OPTBR
地址	0x0008

符号	FLS_OPTBR							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	IWDTSLP	IWDTEN	RFU					
权限	R-0	R-1	R-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				ACLOCKEN		DBRDPEN	
权限	R-0000				R-01		R-01	

位	位名	权限	功能描述
31	IWDTSLP	R	IWDT 在休眠模式下是否允许应用暂停计数 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	R	IWDT 使能 1：上电后 IWDT 开始运行 0：上电后 IWDT 不运行
29:4	RFU	R	未实现：读为 0
3:2	ACLOCKEN	R	应用代码权限锁定使能 00/01/11：ACLOCK 不使能 10：ACLOCK 使能
1:0	DBRDPEN	R	Debug Port 读取保护使能 00/01/11：DBRDP 不使能 10：DBRDP 使能

4.5.4 Flash 擦写控制寄存器(FLS_EPCR)

符号	FLS_EPCR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						ERTYPE	
权限	R-000000						RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						PREQ	EREQ
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:10	RFU	R	未实现：读为 0
9:8	ERTYPE	RW	Flash 擦除类型配置 00/11：page Erase 01：Sector Erase 10：Chip Erase(SWD only)
7:2	RFU	R	未实现：读为 0
1	PREQ	RW	Program Request 软件置位，硬件完成编程后自动清零
0	EREQ	RW	Erase Request 软件置位，硬件完成擦除后自动清零

4.5.5 Flash Key 输入寄存器(FLS_KEY)

符号	FLS_KEY							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	KEY							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	KEY							
权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	KEY							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	KEY							
权限	W-00000000							

位	位名	权限	功能描述
31:0	KEY	W	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。 空地址，物理上无寄存器实现。

4.5.6 Flash 中断使能寄存器(FLS_IER)

符号	FLS_IER							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	FLS_IER							
位名	RFU					AUTHIE	KEYIE	CKIE
权限	R-00000					RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						PRDIE	ERDIE
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:11	RFU	R	未实现：读为 0
10	AUTHIE	RW	Flash 读写权限错误中断使能，1 有效
9	KEYIE	RW	Flash KEY 错误中断使能，1 有效
8	CKIE	RW	擦写定时时钟错误中断使能，1 有效
7:2	RFU	R	未实现：读为 0
1	PRDIE	RW	编程完成标志中断使能，1 有效
0	ERDIE	RW	擦写完成标志中断使能，1 有效

4.5.7 Flash 标志寄存器(FLS_ISR)

符号	FLS_ISR							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				KEYSTA			RFU
权限	R-0000				R-000			R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU					AUTHERR	KEYERR	CKERR
权限	R-00000					W1C-0	W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						PRD	ERD
权限	R-000000						W1C-0	W1C-0

位	位名	权限	功能描述
31:20	RFU	R	未实现：读为 0
19:17	KEYSTA	R	Flash 擦写 KEY 输入状态 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
16:11	RFU	R	未实现：读为 0

位	位名	权限	功能描述
10	AUTHERR	W1C	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	W1C	Flash KEY 错误，硬件置位，软件写 1 清零
8	CKERR	W1C	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。
7:2	RFU	R	未实现：读为 0
1	PRD	W1C	Program Done，编程完成标志，硬件置位，软件写 1 清零
0	ERD	W1C	Erase Done，擦写完成标志，硬件置位，软件写 1 清零

4.5.8 FLASH 检错中断使能寄存器(FLS_ERRIER)

符号	FLS_ERRIER							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					IE	RFU	
权限	R-00000					W1C-0	R-00	

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	IE	W1C	FLASH ECC 检错中断使能，高有效
1:0	RFU	R	未实现：读为 0

4.5.9 Flash ECC 寄存器(FLS_ECCSR)

符号	FLS_ECCSR							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PERDD	RFU	ECCDC	ECCCC	ECCBE	RFU		ECC_IE
权限	W1C-0	R-0	W1C-0	W1C-0	W1C-0	R-00		RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	FLS_ECCSR							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	FLS_SEL	ECC_ADDR						
权限	R-0	R-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ECC_ADDR							
权限	R-00000000							

位	位名	权限	功能描述
31	PERDD	W1C	data flash parity 检错标志 (Parity error Detection in data flash)，硬件置位，软件写 1 清零
30	RFU	R	未实现：读为 0
29	ECCDC	W1C	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	ECCCC	W1C	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	ECCBE	W1C	ECC bit error 标志 当 ECC 检测发现数据正确但是 ECC 位错误时，此寄存器置位。 此时 ECC 也会产生纠错标志，但 32bit 数据本身仍然是正确的可以使用
26:25	RFU	R	未实现：读为 0
24	ECC_IE	RW	code flash 纠错中断使能，1 有效
23:16	RFU	R	未实现：读为 0
15	FLS_SEL	R	该 bit 为 0 指示错误地址发生在 code flash。该 bit 为 1 指示错误 地址发生在 data flash
14:0	ECC_ADDR	R	code flash ECC 检错、纠错和 data flash 出错时的错误 word 地 址

4.5.10 ACLOCK 寄存器 1(FLS_ACLOCK1)

符号	FLS_ACLOCK1							
地址	0x0060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LOCK1							
权限	RW-11111111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LOCK1							

符号	FLS_ACLOCK1							
权限	RW-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LOCK1							
权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCK1							
权限	RW-11111111							

位	位名	权限	功能描述
31:0	LOCK1	RW	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 10：当前 Block 允许 SWD 读写，禁止软件擦写，软件可以取指 或读取 01：当前 Block 允许 SWD 读写，禁止软件读写，软件可以取指 00：当前 Block 禁止 SWD 读写，禁止软件读写，软件可以取指 所有 bit 软件只能写 0，不能写 1。

5 电源管理单元（PMU）

5.1 芯片工作电源

5.1.1 电源域划分

- VDD

芯片的主电源（VDD）的典型工作电压范围是 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，如果使能了 SVD 复位功能，下电复位阈值由 SVD 电路决定，可以由软件配置获得 8 个阈值档位。如果没有使能 SVD 复位，则下电复位阈值由 PDR 电路决定，典型阈值 1.9V。

- REGC

芯片内部 LDO 产生的内核电压，系统上需要尽可能靠近 REGC 管脚对地布置 1uF 稳压电容。

5.1.2 电源结构图

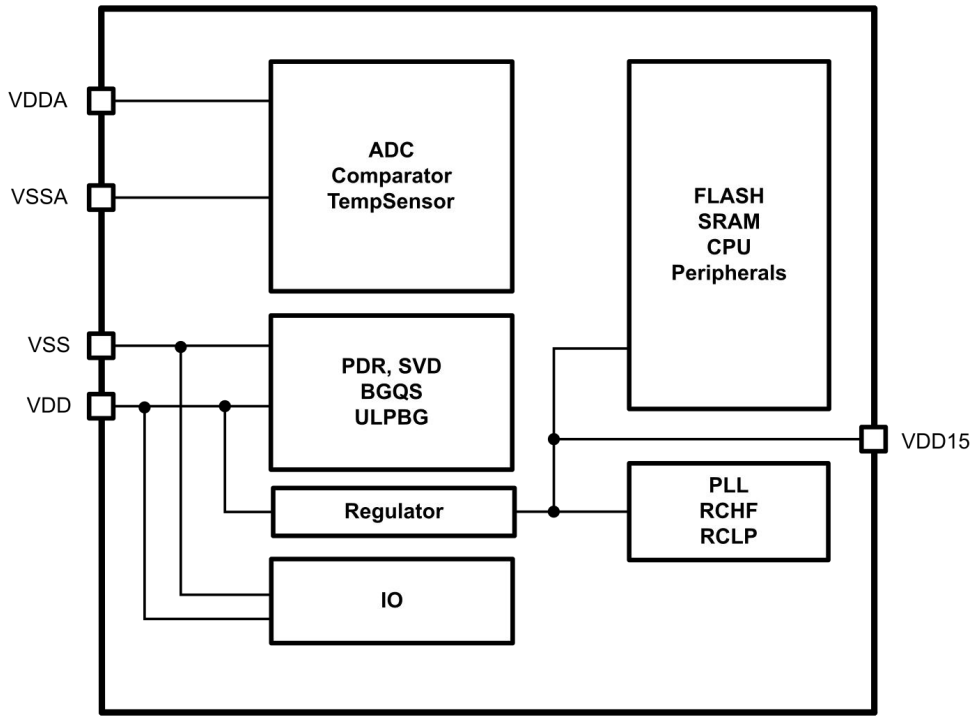


图 5-1 芯片电源结构图

5.1.3 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 和其他模拟电路提供基准电压，为 PLL 提供偏置电流。

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 关闭，内部基准打开，仅部分外设可以运行
- DEEPSLEEP 模式：CPU 停止，Flash 停止，LDO 关闭，内部基准关闭，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	70uA@32KHz	软件主动退出	低速工作	–
SLEEP	10uA	电源检测中断 比较器中断 IO 引脚中断 WKUPx 唤醒 低功耗定时器唤醒 看门狗复位 NRST 引脚复位	CPU 休眠 关闭 RCHF、PLL、ADC 等 内部基准保持开启，LDO15 关闭	32us
DEEPSLEEP	1uA	电源检测中断 比较器中断 IO 引脚中断 WKUPx 唤醒 低功耗定时器唤醒 看门狗复位	CPU 休眠 关闭 RCHF、PLL、ADC 等 内部基准关闭，LDO15 关闭	35us

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

在不同功耗模式下，CPU 主频的限制如下图所示：

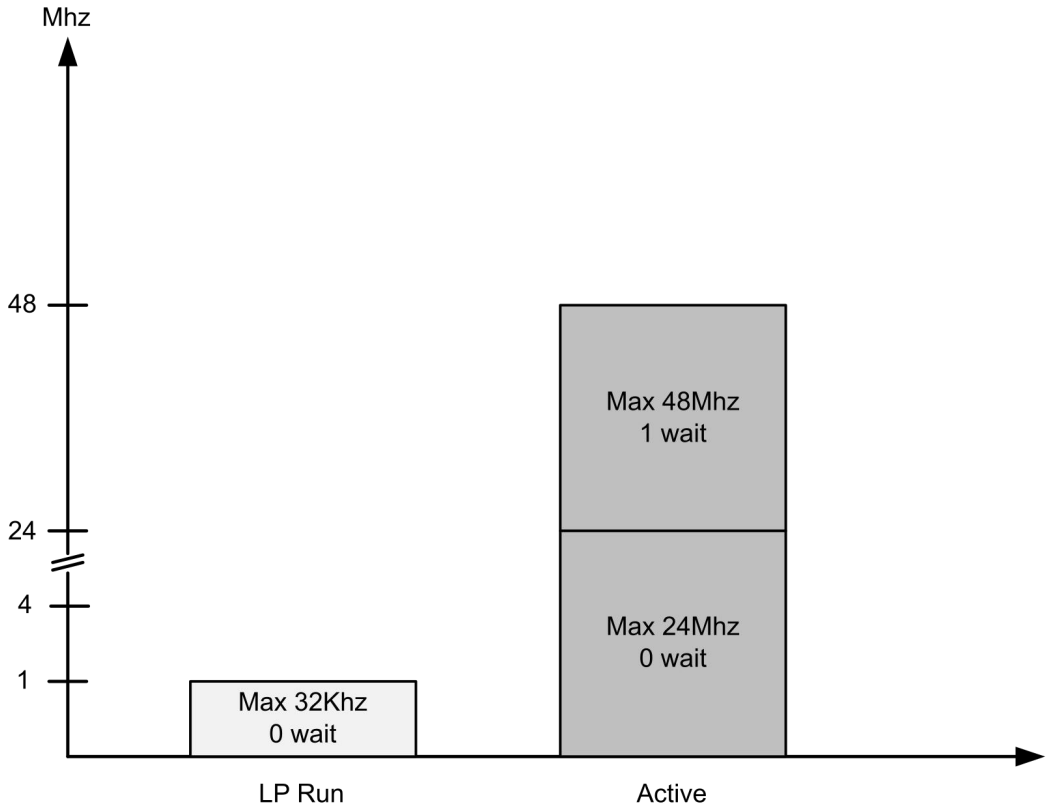


图 5-2 功耗模式与系统主频

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

功耗模式	CPU 频率	可用时钟源	Flash wait	外设工作时钟
ACTIVE	<=24Mhz	All	0	All
ACTIVE	>24Mhz, <=48Mhz	All	1	All
LP Run	32khz	RCLP	0	RCLP

5.2.3 Active 模式

芯片正常工作模式。芯片上电复位完成后进入 Active 模式运行，默认的 CPU 频率是 8MHz，最高可以运行到 48MHz。在 Active 模式下所有的数字和模拟外设都可以全速运行。

当主频高于 24MHz 时，必须配置 flash wait。

5.2.4 LP Run 模式

当芯片需要低功耗低速运行时，可进入 LP RUN 模式，此时 LDO 进入低功耗模式，内核使用 RCLP 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 RCLP
- 配置 PMOD 寄存器为 01
- 如果系统时钟配置不满足以上寄存器条件，则置位异常中断并且禁止进入 LPRUN

LPRUN 模式下的硬件行为

进入 LP Run 之后，硬件自动关闭 RCHF、XTHF、PLL、TRNG，随后 LDO 进入低功耗模式。SVD、比较器仍可以在 LPRUN 模式下工作。

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

退出 LPRUN 模式

按照以下步骤退出 LPRUN 模式：

- 软件将 PMOD 寄存器配置为 00
- 软件根据需要使能 RCHF 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/DEEPSLEEP 模式。如果返回 ACTIVE，硬件恢复 LDO 为正常模式，并解除对高速时钟模块的限制。

5.2.5 SLEEP 模式

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

进入 SLEEP 模式

软件按如下步骤进入 SLEEP 模式：

- 配置 PMOD 寄存器为 10
- 执行 WFI 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。BGSH 根据寄存器配置进入采保或保持正常工作。

数字外设模块可以使用 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，此模式下由于关闭了内部基准源，因此休眠功耗比 SLEEP 进一步降低。

进入 DEEPSLEEP 模式

软件按如下步骤进入 DEEPSLEEP 模式：

- 置位 VREFOFF 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；BGQS 关闭，LDO 关闭

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	Sleep	Deepsleep
停振检测	可屏蔽，XTHF 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	√	√
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	√	√
WKUPx 引脚	可屏蔽，用于外部输入唤醒	√	√
NRST	不可屏蔽，用于全局复位	√	√
LPTIM32	可屏蔽，用于定时唤醒	√	√
BSTIM16	可屏蔽，用于定时唤醒	√	√
LIN 唤醒	可屏蔽，检测到 LIN 唤醒信号时唤醒芯片	√	√

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

注：芯片从休眠模式唤醒后，软件可以通过查询 PMU.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 寄存器

PMU 基地址 (0x40002000)

地址	符号	名称
0x0000	PMU_CR	低功耗控制寄存器
0x0004	PMU_WKTR	唤醒时间控制寄存器
0x0008	PMU_WKFR0	休眠唤醒源标志寄存器
0x000C	PMU_IER	PMU 中断使能寄存器
0x0010	PMU_ISR	PMU 中断标志寄存器
0x003C	PMU_BUFCR	模拟 BUFFER 控制寄存器
0x0050	PMU_PTAT_CR	PTAT 控制寄存器

5.4.1 低功耗控制寄存器(PMU_CR)

符号	PMU_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	REV1							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	REV1							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REV1				WKFSEL		SLPDP	REV
权限	RW-0000				RW-00		RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REV						PMOD	
权限	RW-000000						RW-00	

位	位名	权限	功能描述
31:12	REV1	RW	预留寄存器，可读写，无功能
11:10	WKFSEL	RW	SLEEP/DEEPSLEEP 唤醒频率 00：RCHF-8MHz 01：RCHF-16MHz 10: RCHF-24MHz 11：RFU
9	SLPDP	RW	DeepSleep 控制寄存器(Sleep Deep) 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
8:2	REV	RW	预留寄存器，可读写，无功能
1:0	PMOD	RW	低功耗模式配置寄存器 00: ACTIVE mode 01: LPRUN

位	位名	权限	功能描述
			10: Sleep mode 11: RFU 注 1：本寄存器无法写入 01/11 注 2：从 SLEEP/DEEPSLEEP 返回 ACTIVE 时，由硬件自动改写为 00

5.4.2 唤醒时间控制寄存器(PMU_WKTR)

符号	PMU_WKTR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	T2A		RFU				T1A	
权限	RW-11		R-0000				RW-01	

位	位名	权限	功能描述
31:8	RFU	R	RFU: 未实现，读为 0
7:6	T2A	RW	可编程额外唤醒延迟 2 00: 20us 01: 30us 10: 40us 11: 50us
5:2	RFU	R	RFU:未实现，读为 0
1:0	T1A	RW	可编程额外唤醒延迟 1 00：10us 01：12us 10：16us 11：20us

5.4.3 休眠唤醒源标志寄存器(PMU_WKFR0)

符号	PMU_WKFR0							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5 WKF	UART1 WKF	RFU	RFU	SVDWK F	RFU	RFU	IOWKF
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

符号	PMU_WKFR0							
位名	UART4 WKF	UART2 WKF	RFU	RFU	RFU	RFU	RFU	CMPW KF
权限	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LIN1W KF	RFU	RFU	LPT16 WKF	RFU	BST16 WKF	DBGW KF	RFU
权限	R-0	R-0	R-0	R-0	R-0	R-0	W1C-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		WKPxF					
权限	R-00		W1C-000000					

位	位名	权限	功能描述
31	UART5WKF	R	UART5 中断唤醒标志，中断撤销时硬件自动清零
30	UART1WKF	R	UART1 中断唤醒标志，中断撤销时硬件自动清零
29	RFU	R	RFU:未实现，读为 0
28	RFU	R	RFU:未实现，读为 0
27	SVDWKF	R	SVD 中断唤醒标志，中断撤销时硬件自动清零
26	RFU	R	RFU:未实现，读为 0
25	RFU	R	RFU:未实现，读为 0
24	IOWKF	R	IO 中断唤醒标志，中断撤销时硬件自动清零
23	UART4WKF	R	UART4 中断唤醒标志，中断撤销时硬件自动清零
22	UART2WKF	R	UART2 中断唤醒标志，中断撤销时硬件自动清零
21	RFU	R	RFU:未实现，读为 0
20	RFU	R	RFU:未实现，读为 0
19	RFU	R	RFU:未实现，读为 0
18	RFU	R	RFU:未实现，读为 0
17	RFU	R	RFU:未实现，读为 0
16	CMPWKF	R	模拟比较器中断唤醒标志，中断撤销时硬件自动清零
15	LIN1WKF	R	UART1_LIN 中断唤醒标志，中断撤销时硬件自动清零
14	RFU	R	RFU:未实现，读为 0
13	RFU	R	RFU:未实现，读为 0
12	LPT16WKF	R	LPTIM16 中断唤醒标志，中断撤销时硬件自动清零
11	RFU	R	RFU:未实现，读为 0
10	BST16WKF	R	BSTIM16 中断唤醒标志，中断撤销时硬件自动清零
9	DBGWKF	W1C	CPU Debugger 唤醒标志，软件写 1 清零
8	RFU	R	RFU:未实现，读为 0
7:6	RFU	R	RFU:未实现，读为 0
5:0	WKPxF	W1C	NWKUPx Pin 唤醒标志，软件写 1 清零

5.4.4 PMU 中断使能寄存器(PMU_IER)

符号	PMU_IER
地址	0x000C

符号	PMU_IER							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU0							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU0							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU0							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU0							LPRUNEIE
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU0	R	RFU: 未实现，读为 0
0	LPRUNEIE	RW	LPRUN 错误中断标志，硬件置位，软件写 1 清零 1：在 PMOD=2’h1 后，系统时钟不符合 LPRUN 模式的定义时置位 0：在 PMOD=2’h1 后，系统时钟符合 LPRUN 模式的定义

5.4.5 PMU 中断标志寄存器(PMU_ISR)

符号	PMU_ISR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU0							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU0							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU0							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU0							LPRUNEIF
权限	R-0000000							W1C-0

位	位名	权限	功能描述
31:1	RFU0	R	RFU: 未实现，读为 0
0	LPRUNEIF	W1C	LPRUN 错误中断标志，硬件置位，软件写 1 清零 1：在 PMOD=2’h1 后，系统时钟不符合 LPRUN

位	位名	权限	功能描述
			模式的定义时置位 0：在 PMOD=2’h1 后，系统时钟符合 LPRUN 模式的定义

5.4.6 模拟 BUFFER 控制寄存器(PMU_BUFCR)

符号	PMU_BUFCR							
地址	0x003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	RFU	AVREFBUF_OUTEN	AVREFBUF_EN	VPTATBUFFER_OUTEN	VPTATBUFFER_EN	RFU	
权限	R-0	R-0	RW-0	RW-0	RW-0	RW-0	R-00	

位	位名	权限	功能描述
31:7	RFU	R	RFU：未实现，读为 0
6	RFU	R	RFU：未实现，读为 0
5	AVREFBUF_OUTEN	RW	AVREF 输出缓冲输出使能，ADC 采样 AVREF 时建议使能
4	AVREFBUF_EN	RW	AVREF 输出缓冲使能，ADC 采样 AVREF 时建议使能
3	VPTATBUFFER_OUTEN	RW	VPTAT BUFFER 模块开关通道输出使能信号，高电平使能有效
2	VPTATBUFFER_EN	RW	VPTAT BUFFER 模块使能信号，高电平使能有效
1:0	RFU	R	RFU：未实现，读为 0

5.4.7 PTAT 控制寄存器(PMU_PTAT_CR)

符号	PMU_PTAT_CR							
地址	0x0050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

符号	PMU_PTAT_CR							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							PTAT_EN
权限	R-0000000							RW-0

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

6 处理器（CPU）

6.1 概述

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

表 6-1 CPU 配置简表

Feature	Options	FM442 Config
Interrupts	1~32	32
Data endianness	little/big	little
SysTick Timer	Present or absent	absent
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

6.2 寄存器

主要内核寄存器列表

表 6-2 Cortex-M0 内核寄存器简表

名字	描述
R0-R12	通用寄存器
MSP (R13) PSP (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

名字	描述
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

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

参见后续章节中断向量表

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：

存储器相关:

- 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

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

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。当 NRST 引脚为低电平超过 8ms 后，看门狗自动关闭。当调试器不使用 NRST 引脚时，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。

7 中断向量表（INTMUX）

7.1 中断向量表

CPU 中断向量表如下，

表 7-1 中断向量表

Position	Priority	Priority type	Acronym	Description	Address
0	–	–	MSP 初值	主栈指针初始化地址	0x0000_0000
1	-3	fixed	Reset	复位向量	0x0000_0004
2	-2	fixed	NMI	WKUPx 中断 低功耗模式错误中断 存储器非法访问中断 Flash ECC 检错中断	0x0000_0008
3	-1	fixed	HardFault	HardFault 中断向量	0x0000_000C
4-10	–	–	–	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall 系统服务请求	0x0000_002C
12-13	–	–	–	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	可挂起系统服务请求	0x0000_0038
15	6	settable	Systick	内部定时器中断向量	0x0000_003C
16	7	settable	WDT	窗口看门狗或独立看门狗中断	0x0000_0040
17	8	settable	SVD	电源监测报警中断	0x0000_0044
18	9	settable			0x0000_0048
19	10	settable	MUX0	可配置多路中断	0x0000_004C
20	11	settable			0x0000_0050
21	12	settable	ADC	ADC 转换完成中断	0x0000_0054
22	13	settable			0x0000_0058
23	14	settable	MUX1	可配置多路中断	0x0000_005C
24	15	settable	MUX2		0x0000_0060
25	16	settable	MUX3		0x0000_0064
26	17	settable	MUX4		0x0000_0068
27	18	settable	MUX5		0x0000_006C
28	19	settable	MUX6		0x0000_0070
29	20	settable	MUX7		0x0000_0074
30	21	settable	MUX8		0x0000_0078
31	22	settable	I2CMx	I2C master 中断	0x0000_007C
32	23	settable			0x0000_0080
33	24	settable	GPTIM	GPTIM 中断	0x0000_0084
34	25	settable	MUX9	可配置多路中断	0x0000_0088
35	26	settable			0x0000_008C
36	27	settable	LPTIM	LPTIM16 中断	0x0000_0090
37	28	settable	DMA	DMA 中断	0x0000_0094

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

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

7.2 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。

表 7-2 中断源多路选择表

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

7.3 寄存器

INTMUX 基地址 (0x40016000)

地址	符号	名称
0x0000	INTMUX_CR	中断源选择寄存器

7.3.1 中断源选择寄存器(INTMUX_CR)

符号	INTMUX_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU				MUX13SEL		MUX12SEL	
权限	R-0000				RW-00		RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MUX11SEL		MUX10SEL		MUX9SEL		MUX8SEL	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MUX7SEL		MUX6SEL		MUX5SEL		MUX4SEL	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MUX3SEL		MUX2SEL		MUX1SEL		MUX0SEL	
权限	RW-00		RW-00		RW-00		RW-00	

位	位名	权限	功能描述
31:28	RFU	R	未实现
27:26	MUX13SEL	RW	中断源选择,详见中断向量章节
25:24	MUX12SEL	RW	中断源选择,详见中断向量章节
23:22	MUX11SEL	RW	中断源选择,详见中断向量章节
21:20	MUX10SEL	RW	中断源选择,详见中断向量章节
19:18	MUX9SEL	RW	中断源选择,详见中断向量章节
17:16	MUX8SEL	RW	中断源选择,详见中断向量章节
15:14	MUX7SEL	RW	中断源选择,详见中断向量章节
13:12	MUX6SEL	RW	中断源选择,详见中断向量章节
11:10	MUX5SEL	RW	中断源选择,详见中断向量章节
9:8	MUX4SEL	RW	中断源选择,详见中断向量章节
7:6	MUX3SEL	RW	中断源选择,详见中断向量章节
5:4	MUX2SEL	RW	中断源选择,详见中断向量章节
3:2	MUX1SEL	RW	中断源选择,详见中断向量章节
1:0	MUX0SEL	RW	中断源选择,详见中断向量章节

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、看门狗复位、软件复位、引脚复位等
- 上下电复位（POR/PDR）监控主电源供电
- 上电复位典型释放电压 2.1V
- 低功耗下电复位电路（PDR），典型阈值 1.9V
- SVD 可用于产生中断或复位，阈值可编程

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

8.2 模块框图

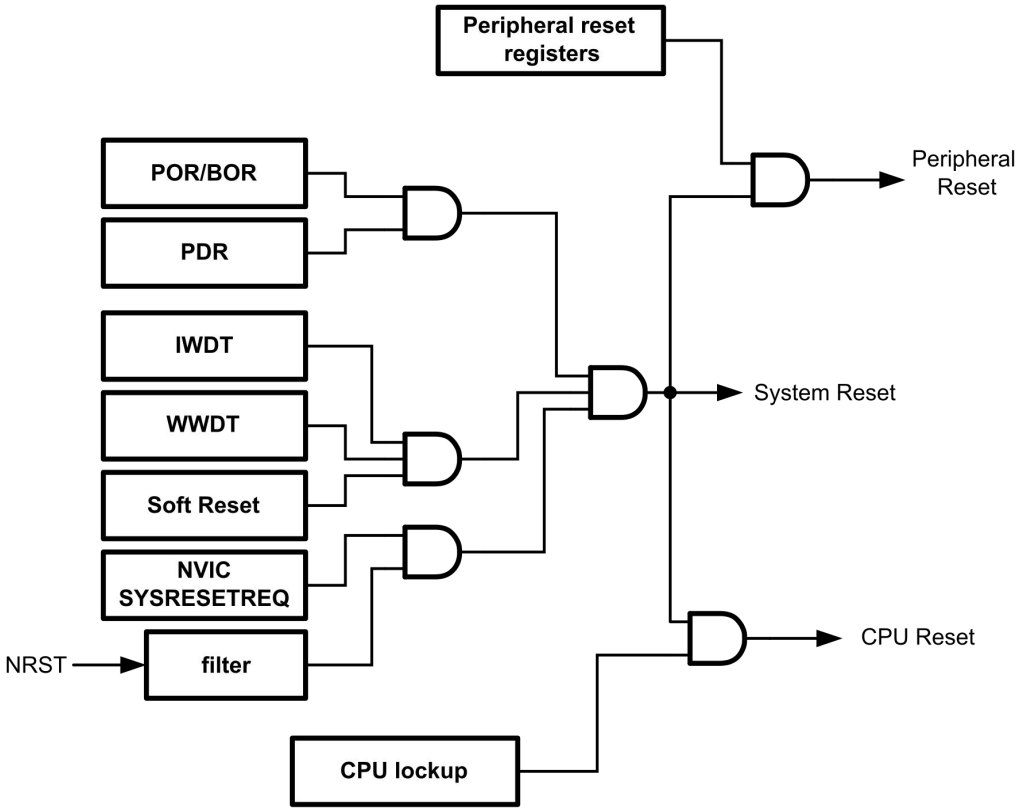


图 8-1 芯片复位源框图

8.3 上下电复位

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

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

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

SVD 下电复位阈值软件可配置，共 5 档：2.4V、2.7V、3V、3.6V、4V，注意电源回升时有 200mV 左右的回滞窗口。

芯片上电复位放开后，SVD 下电复位默认是关闭的，软件可以开启。

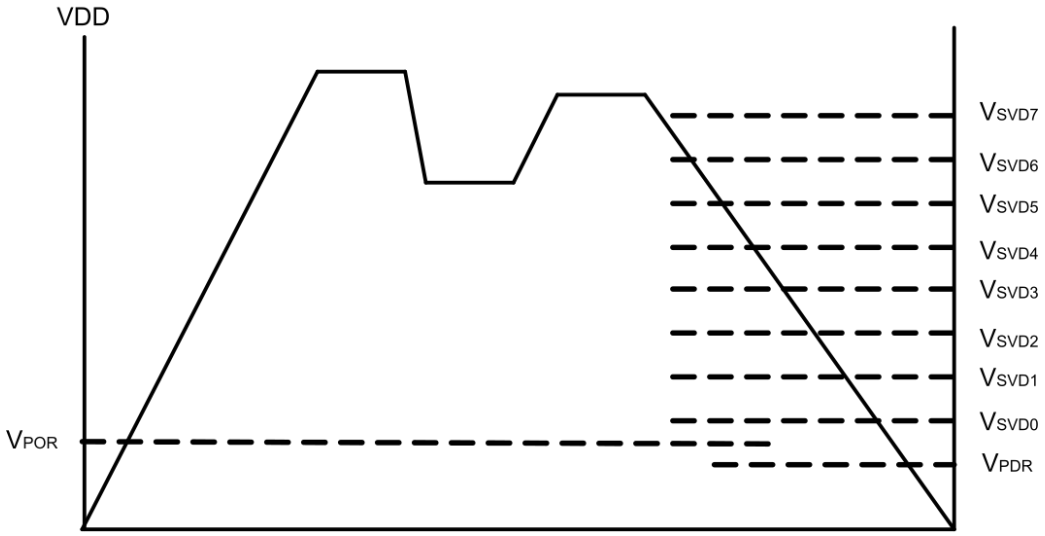


图 8-2 上下电复位示意图

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

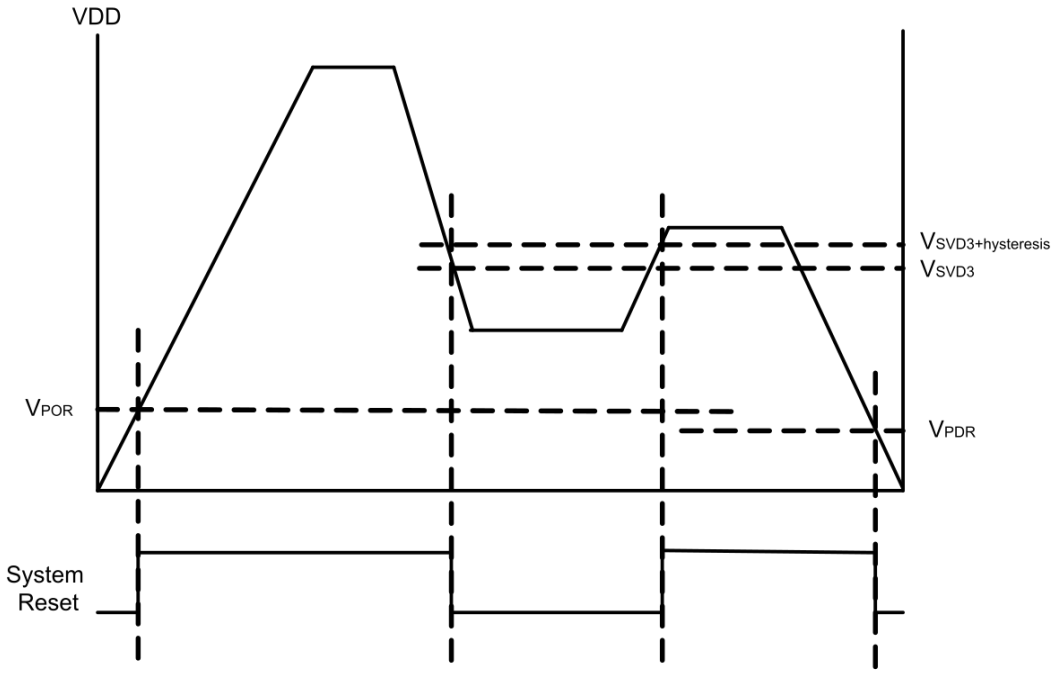


图 8-3 POR, SVD, PDR 复位波形示意图

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，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 或其他功能，此后 NRST 复位功能失效。详情参见”I/O 端口”章节。

8.6 寄存器

RMU 基地址 (0x40002800)

地址	符号	名称
0x0004	RMU_BORCR	BOR 配置寄存器
0x0008	RMU_LKPCR	复位配置寄存器
0x000C	RMU_SOFTRST	软件复位寄存器
0x0010	RMU_RSTFR	复位标志寄存器
0x0014	RMU_PRSTEN	外设复位使能寄存器
0x0018	RMU_AHBRSTCR	AHB 外设复位寄存器
0x001C	RMU_APBRSTCR1	APB 外设复位寄存器 1
0x0020	RMU_APBRSTCR2	APB 外设复位寄存器 2

8.6.1 BOR 配置寄存器(RMU_BORCR)

符号	RMU_BORCR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			BOR_CFG			RFU	BOR_ENB
权限	R-000			RW-000			R-0	RW-0

位	位名	权限	功能描述
31:5	RFU	R	RFU：未实现，读为 0
4:2	BOR_CFG	RW	BOR 复位电压配置 000：falling 2.4V reset, rising 2.6V release 001：falling 2.4V reset, rising 2.6V release 010：falling 2.4V reset, rising 2.6V release 011：falling 2.4V reset, rising 2.6V release 100：falling 2.7V reset, rising 2.9V release 101：falling 3.0V reset, rising 3.2V release 110：falling 3.6V reset, rising 3.8V release 111：falling 4.0V reset, rising 4.2V release
1	RFU	R	RFU：未实现，读为 0
0	BOR_ENB	RW	BOR 复位使能，上电后默认关闭 0：使能 BOR 下电复位 1：关闭 BOR 下电复位

8.6.2 复位配置寄存器(RMU_LKPCR)

符号	RMU_LKPCR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							

符号	RMU_LKPCR							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						LKUPRST_EN	RFU0
权限	R-000000						RW-0	R-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	LKUPRST_EN	RW	LOCKUP 复位使能 1：使能 CPU LOCKUP 复位 0：屏蔽 CPU LOCKUP 复位
0	RFU0	R	RFU：未实现，读为 0

8.6.3 软件复位寄存器(RMU_SOFTRST)

符号	RMU_SOFTRST							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SOFTRST							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SOFTRST							
权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SOFTRST							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SOFTRST							
权限	W-00000000							

位	位名	权限	功能描述
31:0	SOFTRST	W	软件写 0x5C5C_AABB 触发全局复位

8.6.4 复位标志寄存器(RMU_RSTFR)

符号	RMU_RSTFR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	RMU_RSTFR							
位名	RFU	RAMP_FLAG	MAP_FLAG	MDFN_FLAG	NRSTN_FLAG	RFU	RFU	PODRN_FLAG
权限	R-0	W1C-0	W1C-0	W1C-0	W1C-0	R-0	R-0	W1C-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SVDN_FLAG	TESTN_FLAG	SOFTN_FLAG	IWDTN_FLAG	RFU	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
权限	W1C-0	W1C-0	W1C-0	W1C-0	R-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:15	RFU	R	RFU：未实现，读为 0
14	RAMP_FLAG	W1C	RAM parity 错误复位标志，高有效，软件写 1 清零
13	MAP_FLAG	W1C	非法地址访问复位标志，高有效，软件写 1 清零
12	MDFN_FLAG	W1C	模式诊断超时复位标志，高有效，硬件置位，软件写 1 清零
11	NRSTN_FLAG	W1C	NRST 引脚复位标志，高有效，硬件置位，软件写 1 清零
10	RFU	R	RFU：未实现，读为 0
9	RFU	R	RFU：未实现，读为 0
8	PODRN_FLAG	W1C	上下电复位标志，高有效，硬件置位，软件写 1 清零
7	SVDN_FLAG	W1C	SVD 复位标志，高有效，硬件置位，软件写 1 清零
6	TESTN_FLAG	W1C	TESTN 引脚复位标志，高有效，硬件置位，软件写 1 清零
5	SOFTN_FLAG	W1C	软件复位标志，高有效，硬件置位，软件写 1 清零
4	IWDTN_FLAG	W1C	IWDT 复位标志，高有效，硬件置位，软件写 1 清零
3	RFU	R	RFU：未实现，读为 0
2	WWDTN_FLAG	W1C	WWDT 复位标志，高有效，硬件置位，软件写 1 清零
1	LKUPN_FLAG	W1C	LOOKUP 复位标志，高有效，硬件置位，软件写 1 清零
0	NVICN_FLAG	W1C	NVIC 复位标志，高有效，硬件置位，软件写 1 清零

8.6.5 外设复位使能寄存器(RMU_PRSTEN)

符号	RMU_PRSTEN							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PERHRSTEN							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PERHRSTEN							
权限	W-00000000							

符号	RMU_PRSTEN							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PERHRSTEN							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PERHRSTEN							
权限	W-00000000							

位	位名	权限	功能描述
31:0	PERHRSTEN	W	外设模块复位使能，物理实现上只有 1bit 软件对此地址写 0x1357_9BDF，使能外设复位功能，此后可以通过外设模块复位寄存器复位各个模块 软件对此地址写任意其他数据，将关闭外设复位功能 当软件读该寄存器时，最低位为物理寄存器有效值。

8.6.6 AHB 外设复位寄存器(RMU_AHBRSTCR)

符号	RMU_AHBRSTCR							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							DMARST
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	DMARST	RW	DMA 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

8.6.7 APB 外设复位寄存器 1(RMU_APBRSTCR1)

符号	RMU_APBRSTCR1

符号	RMU_APBRSTCR1							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UART5RST	UART4RST	RFU	UART2RST	UART1RST	RFU	UARTIRRST	RFU
权限	RW-0	RW-0	R-0	RW-0	RW-0	R-0	RW-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU	RFU	SPI1RST	SPI0RST	RFU	RFU	RFU	RFU
权限	R-0	R-0	RW-0	RW-0	R-0	R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	INTMUXRST	RFU	RFU	RFU	RFU	RFU	I2C1RST	I2C0RST
权限	RW-0	R-0	R-0	R-0	R-0	R-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	RFU	PGLRST	ATIMRST	TAU1RST	TAU0RST	LPT16RST	BT16RST
权限	R-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31	UART5RST	RW	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	RW	UART4 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
29	RFU	R	RFU：未实现，读为 0
28	UART2RST	RW	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	RW	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	RFU	R	RFU：未实现，读为 0
25	UARTIRRST	RW	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	R	RFU：未实现，读为 0
23	RFU	R	RFU：未实现，读为 0
22	RFU	R	RFU：未实现，读为 0
21	SPI1RST	RW	SPI1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
20	SPI0RST	RW	SPI0 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

位	位名	权限	功能描述
19	RFU	R	RFU：未实现，读为 0
18	RFU	R	RFU：未实现，读为 0
17	RFU	R	RFU：未实现，读为 0
16	RFU	R	RFU：未实现，读为 0
15	INTMUXRST	RW	INTMUX 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
14	RFU	R	RFU：未实现，读为 0
13	RFU	R	RFU：未实现，读为 0
12	RFU	R	RFU：未实现，读为 0
11	RFU	R	RFU：未实现，读为 0
10	RFU	R	RFU：未实现，读为 0
9	I2C1RST	RW	I2C1 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
8	I2C0RST	RW	I2C0 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	R	RFU：未实现，读为 0
6	RFU	R	RFU：未实现，读为 0
5	PGLRST	RW	PGL 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
4	ATIMRST	RW	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	RW	TAU1 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	TAU0RST	RW	TAU0 控制器复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
1	LPT16RST	RW	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	RW	BSTIM16 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位

8.6.8 APB 外设复位寄存器 2(RMU_APBRSTCR2)

符号	RMU_APBRSTCR2							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							ADCCRST

符号	RMU_APBRSTCR2							
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ADCRST	RFU					CRCRST	RFU
权限	RW-0	R-00000					RW-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	GPTIMRST	RFU			DIVASRST	RFU	SVDRST	COMPRST
权限	RW-0	R-000			RW-0	R-0	RW-0	RW-0

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	ADCCRST	RW	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	RW	ADC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
22:18	RFU	R	RFU：未实现，读为 0
17	CRCRST	RW	CRC 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
16:8	RFU	R	RFU：未实现，读为 0
7	GPTIMRST	RW	GPTIM 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
6:4	RFU	R	RFU：未实现，读为 0
3	DIVASRST	RW	DIVAS 模块复位，软件写 1 复位，写 0 撤销复位 0：不复位 1：复位
2	RFU	R	RFU：未实现，读为 0
1	SVDRST	RW	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	RW	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 电源电压监测（SVD）

9.1 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 2.4V~4.0V，5 级可编程阈值档位
- 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

9.2 结构框图

下图是电源检测电路的模块框图。本模块除了 SVD_IER，其余寄存器只在上电复位时复位。

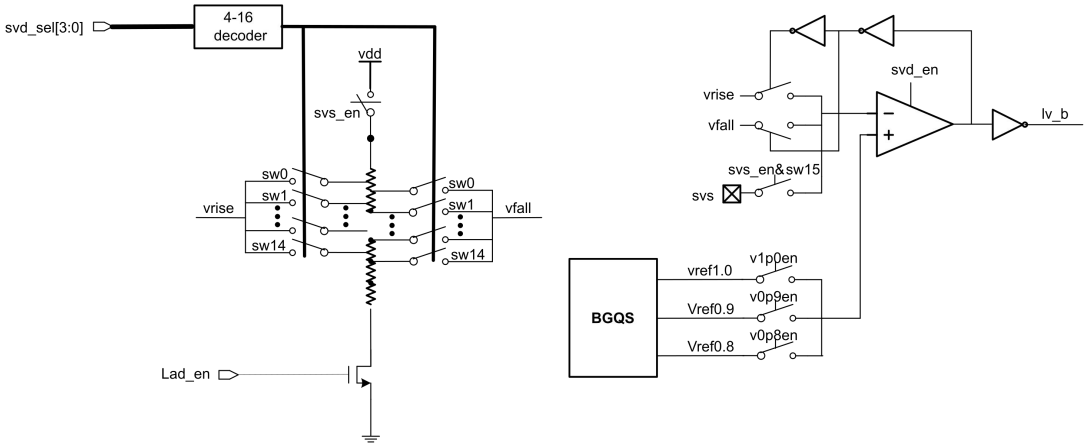


图 9-1 低压检测电路框图

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：

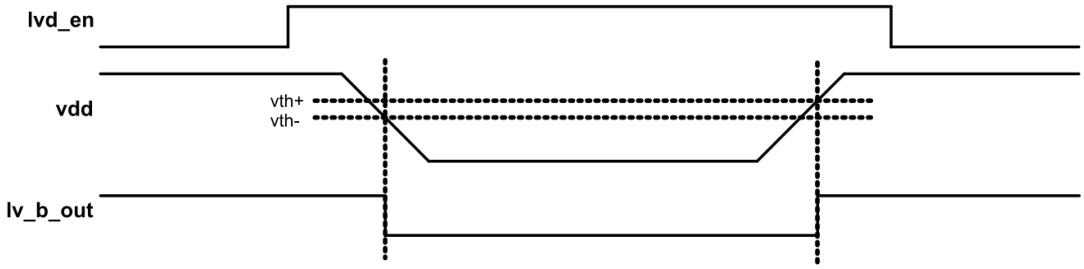


图 9-2 低压检测电路工作时序

9.3 引脚定义

SVD 模块直接检测芯片电源（VDD）。

9.4 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。

9.4.1 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。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。

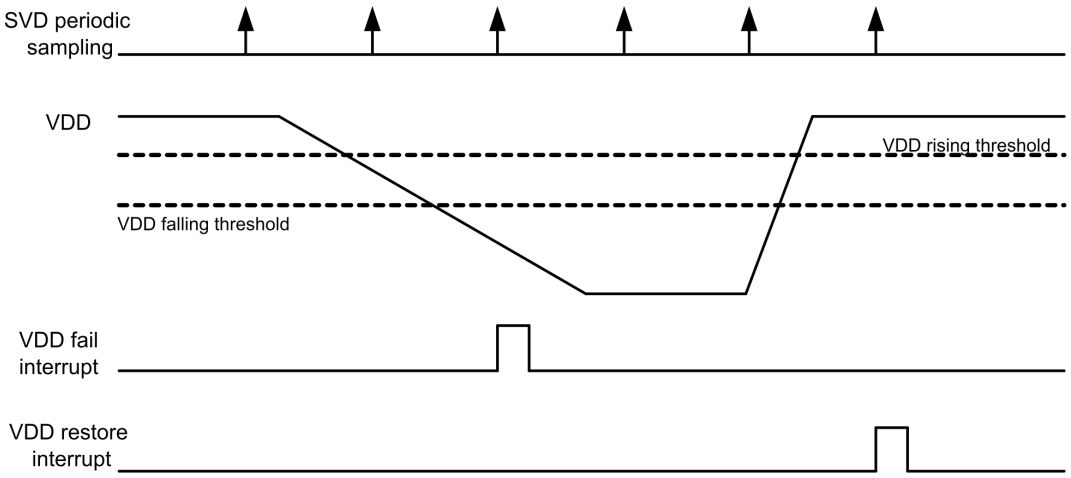


图 9-3 电源检测电路间歇工作模式

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。

9.4.2 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。

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。

9.4.3 欠压复位

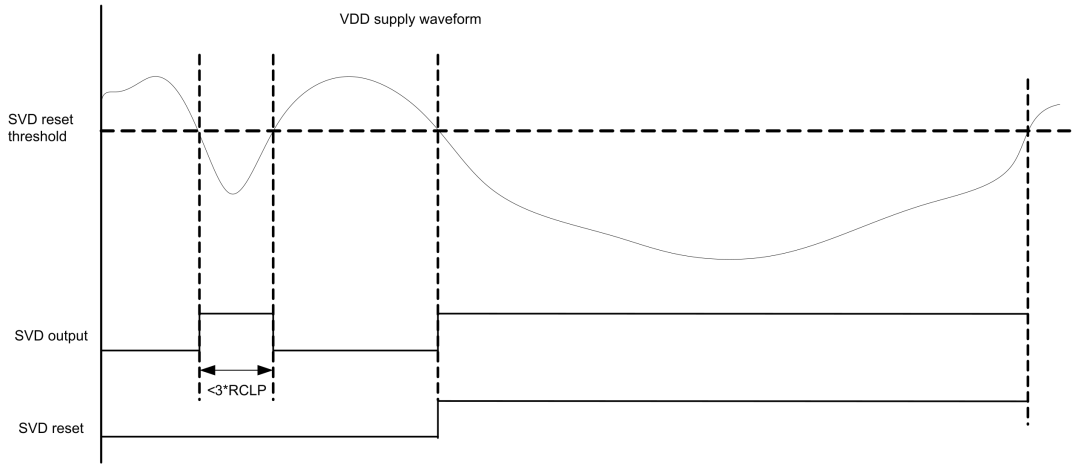
SVDEN=10 时，使能 SVD 欠压复位功能。使用欠压复位时，必须保持 AVREF 使能，即使进入 deepsleep 时硬件也不会关闭 AVREF，因此此时功耗会增加约 2uA。

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，注意这个复位不会复位 SVD 自己的控制寄存器

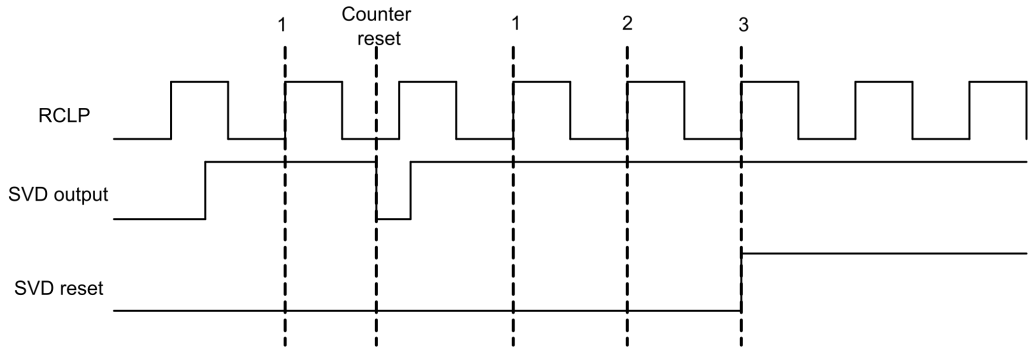
当电源电压高于 LVL 定义的阈值时，复位释放。

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。

为了增强抗干扰能力，可以使能数字滤波功能，连续三个 RCLP 时钟采样到 SVD 复位有效时，才会产生全局复位，在此过程中任何 SVD 复位无效信号都会清零计数器。



数字滤波示意图：



9.5 电源检测阈值

通过 SVDLVL 寄存器可以选择电压检测阈值。

SVDLVL	下降阈值（V）	上升阈值（V）
000	2.4	2.6
001	2.4	2.6
010	2.4	2.6
011	2.4	2.6
100	2.7	2.9
101	3.0	3.2
110	3.6	3.8
111	4.0	4.2

9.6 寄存器

SVD 基地址 (0x4001A800)

地址	符号	名称
0x0000	SVD_CFGR	SVD 配置寄存器
0x0004	SVD_CR	SVD 控制寄存器
0x0008	SVD_IER	SVD 中断使能寄存器
0x000C	SVD_ISR	SVD 状态和标志寄存器
0x0010	SVD_VSR	SVD 参考电压选择寄存器

9.6.1 SVD 配置寄存器(SVD_CFGR)

符号	SVD_CFGR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	LVL			DFEN	MOD	ITVL	
权限	R-0	RW-000			RW-1	RW-0	RW-00	

位	位名	权限	功能描述
31:7	RFU	R	RFU：未实现，读为 0
6:4	LVL	RW	SVD 报警阈值档位设置，参见电源检测阈值章节
3	DFEN	RW	数字滤波使能（SVDMODE=1 时必须置 1） 1：启动 SVD 输出的数字滤波 0：关闭 SVD 输出的数字滤波
2	MOD	RW	SVD 工作模式选择，配置模式后还要置位 SVDEN 才会启动 SVD 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	RW	SVD Interval，SVD 间歇使能间隔 00：62.5ms 01：256ms 10：1s 11：4s

9.6.2 SVD 控制寄存器(SVD_CR)

符号	SVD_CR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						EN	
权限	R-000000						RW-00	

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1:0	EN	RW	SVD 使能 (SVD enable) 00：关闭 SVD 01：SVD 低压报警模式使能 10：SVD 欠压复位模式使能 11：关闭 SVD

9.6.3 SVD 中断使能寄存器(SVD_IER)

符号	SVD_IER							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						PFIE	PRIE
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	PFIE	RW	电源跌落中断使能寄存器，1 有效 (Power Fall

位	位名	权限	功能描述
			interrupt enable)
0	PRIE	RW	电源恢复中断使能寄存器，1 有效 (Power Rise interrupt enable)

9.6.4 SVD 状态和标志寄存器(SVD_ISR)

符号	SVD_ISR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							SVDO
权限	R-0000000							R-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SVDR	RFU					PFF	PRF
权限	R-1	R-00000					W1C-0	W1C-0

位	位名	权限	功能描述
31:9	RFU	R	RFU：未实现，读为 0
8	SVDO	R	SVD 电源检测输出 (SVD output) 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	R	SVD 输出锁存信号，数字电路锁存的 SVD 状态 (SVD registered output)
6:2	RFU	R	RFU：未实现，读为 0
1	PFF	W1C	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 (Power fall flag)
0	PRF	W1C	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 (Power rise flag)

9.6.5 SVD 参考电压选择寄存器(SVD_VSR)

符号	SVD_VSR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	SVD_VSR							
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					V1P0EN	RFU	
权限	R-00000					RW-1	R-00	

位	位名	权限	功能描述
31:3	RFU	R	RFU：未实现，读为 0
2	V1P0EN	RW	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
1:0	RFU	R	RFU：未实现，读为 0

10 独立看门狗（IWDT）

10.1 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执行清狗动作启动。

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。

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。

10.2 结构框图

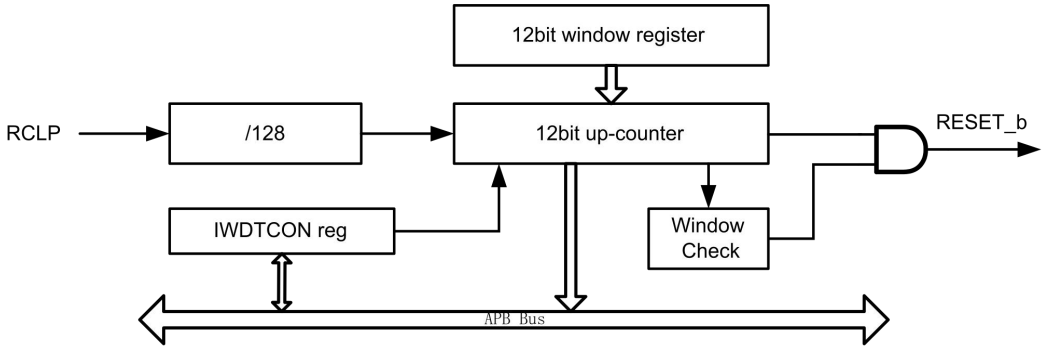


图 10-1 IWDT 结构框图

10.3 IWDT 功能描述

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

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

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

- 溢出周期更新完毕，CPU 正常运行

IWDT 使用 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 = 32KHz 时，不同溢出长度配置和溢出时间关系如下：

表 10-1 IWDT 溢出周期表

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

10.4 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。

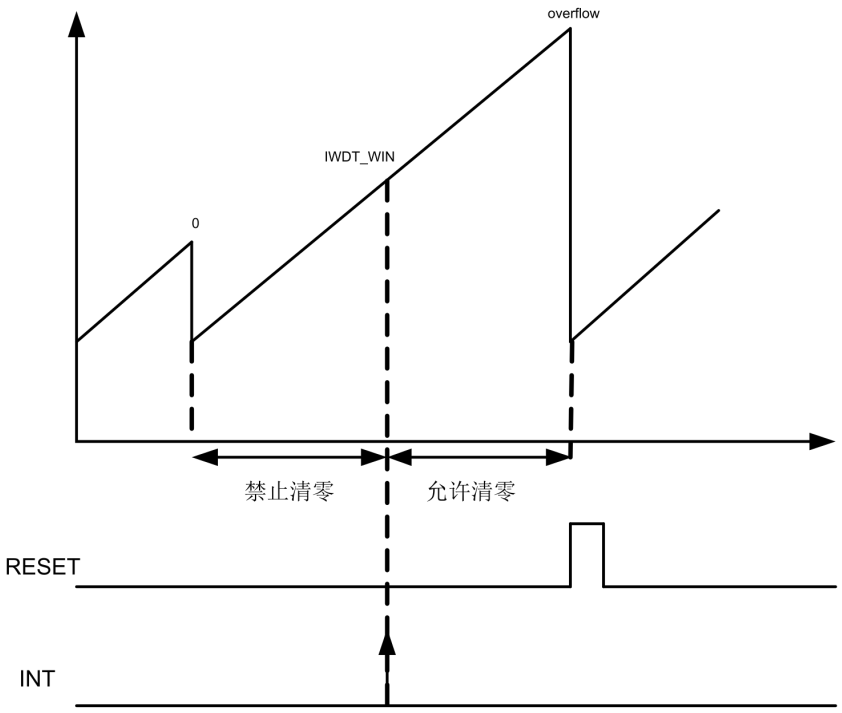


图 10-2 IWDT 窗口示意图

10.5 IWDT 冻结

用户可以通过 OPTBYTES 配置是否允许 IWDT 在休眠模式下冻结计数。

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

10.6 寄存器

IWDT 基地址 (0x40011400)

地址	符号	名称
0x0000	IWDT_SERV	IWDT 清除寄存器
0x0004	IWDT_CR	IWDT 配置寄存器
0x0008	IWDT_CNT	IWDT 计数值寄存器
0x000C	IWDT_WIN	IWDT 窗口寄存器
0x0010	IWDT_IER	IWDT 中断使能寄存器
0x0014	IWDT_ISR	IWDT 中断标志寄存器

10.6.1 IWDT 清除寄存器(IWDT_SERV)

符号	IWDT_SERV							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	IWDT_SERV							
位名	SERV							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	SERV							
权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SERV							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SERV							
权限	W-00000000							

位	位名	权限	功能描述
31:0	SERV	W	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)

10.6.2 IWDT 配置寄存器(IWDT_CR)

符号	IWDT_CR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU1							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU1							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU1				FREEZE	RFU0		
权限	R-0000				RW-0	R-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU0					OVP		
权限	R-00000					RW-001		

位	位名	权限	功能描述
31:12	RFU1	R	RFU：未实现，读为 0
11	FREEZE	RW	芯片休眠时冻结 IWDT，仅在 OPTBYTES 中 IWDTSLP 配置有效时起作用 1：Sleep/DeepSleep 模式下冻结 IWDT 计数 0：Sleep/DeepSleep 模式下保持 IWDT 运行
10:3	RFU0	R	RFU：未实现，读为 0
2:0	OVP	RW	配置看门狗溢出周期

位	位名	权限	功能描述
			000：125ms 001：250ms 010：500ms 011：1s 100：2s 101：4s 110：8s 111：16s

10.6.3 IWDT 计数值寄存器(IWDT_CNT)

符号	IWDT_CNT							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				CNT			
权限	R-0000				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:0	CNT	R	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

10.6.4 IWDT 窗口寄存器(IWDT_WIN)

符号	IWDT_WIN							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	IWDT_WIN							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				WIN			
权限	R-0000				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WIN							
权限	RW-00000000							

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:0	WIN	RW	IWDT 窗口寄存器

10.6.5 IWDT 中断使能寄存器(IWDT_IER)

符号	IWDT_IER							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							IE
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	IE	RW	IWDT 窗口中断使能 0：中断使能禁止 1：中断使能打开

10.6.6 IWDT 中断标志寄存器(IWDT_ISR)

符号	IWDT_ISR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	IWDT_ISR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							WINIF
权限	R-0000000							W1C-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	WINIF	W1C	IWDT 进入窗口中断标志，写 1 清零 0：无中断产生 1：计数值进入清狗窗口

11 窗口看门狗（WWDT）

11.1 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。

11.2 结构框图

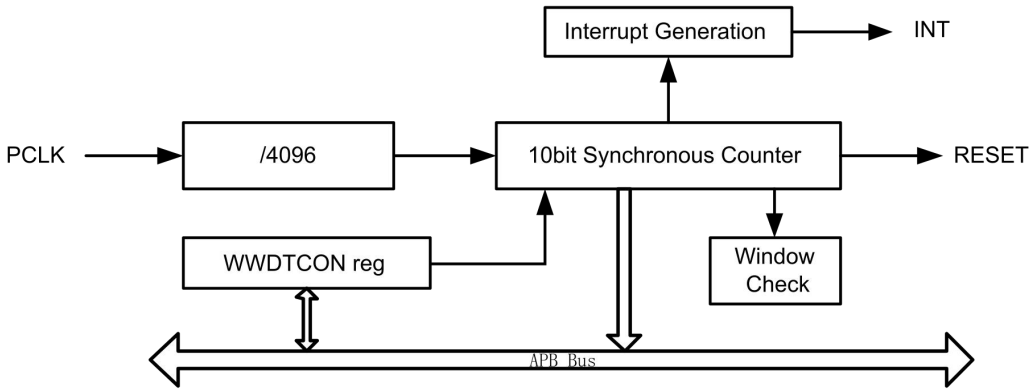


图 11-1 WWDT 结构框图

11.3 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：

$$t_{WWDT} = T_{APBCLK} * 4096 * N_{CFG}$$

下表为计算示例：

表 11-1 WWDT 溢出周期表

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
	1024	524.288

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。

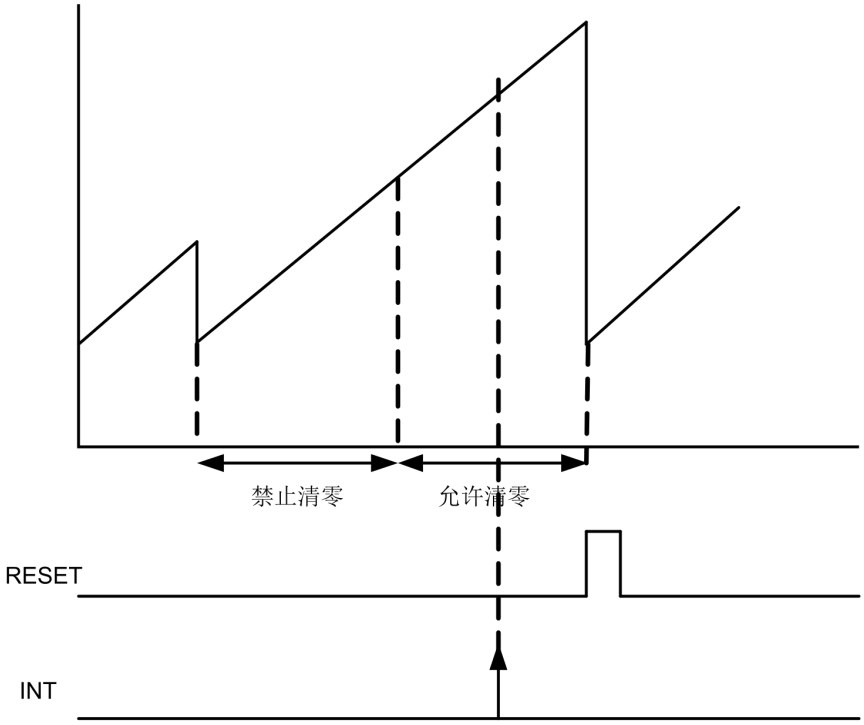


图 11-2 WWDT 窗口示意图

11.4 寄存器

WWDT 基地址 (0x40011800)

地址	符号	名称
0x0000	WWDT_CR	WWDT 控制寄存器
0x0004	WWDT_CFGR	WWDT 配置寄存器
0x0008	WWDT_CNT	WWDT 计数值寄存器
0x000C	WWDT_IER	WWDT 中断使能寄存器
0x0010	WWDT_ISR	WWDT 中断标志寄存器
0x0014	WWDT_PSC	WWDT 预分频寄存器

11.4.1 WWDT 控制寄存器(WWDT_CR)

符号	WWDT_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							

符号	WWDT_CR							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CON							
权限	W-00000000							

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:0	CON	W	当 CPU 向此地址写入 0x5A 时启动 WWDT 定时器 在启动 WWDT 后，当 CPU 向此地址写入 0xAC 时清零计数器

11.4.2 WWDT 配置寄存器(WWDT_CFGR)

符号	WWDT_CFGR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					CFG		
权限	R-00000					RW-011		

位	位名	权限	功能描述
31:3	RFU	R	RFU：未实现，读为 0
2:0	CFG	RW	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 000：T_{PCLK} * 4096 * 1 001：T_{PCLK} * 4096 * 4 010：T_{PCLK} * 4096 * 16 011：T_{PCLK} * 4096 * 64 100：T_{PCLK} * 4096 * 128 101：T_{PCLK} * 4096 * 256 110：T_{PCLK} * 4096 * 512 111：T_{PCLK} * 4096 * 1024

11.4.3 WWDT 计数值寄存器(WWDT_CNT)

符号	WWDT_CNT							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						CNT	
权限	R-000000						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:10	RFU	R	RFU：未实现，读为 0
9:0	CNT	R	WWDT 计数值寄存器值，软件可通过查询此寄存器了解 WWDT 计 时进度 (WWDT Counter value,read only)

11.4.4 WWDT 中断使能寄存器(WWDT_IER)

符号	WWDT_IER							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							IE
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	IE	RW	WWDT 中断使能 (WWDT Interrupt Enable) 0：中断使能禁止 1：中断使能打开

11.4.5 WWDT 中断标志寄存器(WWDT_ISR)

符号	WWDT_ISR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							NOVF
权限	R-0000000							W1C-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	NOVF	W1C	WWDT 75%计时中断标志，写 1 清零 0：无中断产生 1：中断标志置位 如果 IE=1，则寄存器置位将触发中断

11.4.6 WWDT 预分频寄存器(WWDT_PSC)

符号	WWDT_PSC							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				DIV_CNT			
权限	R-0000				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIV_CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:0	DIV_CNT	R	WWDT 的 4096 预分频计数器当前数值，只读

12 时钟管理单元（CMU）

12.1 概述

芯片内包含 4~40MHz 高频晶体振荡器、16MHz 高频 RC 振荡器(RCHF)、32KHz 低功耗内部环振（RCLP）、一个锁相环（PLL）。芯片内部的时钟产生模块整合这些时钟源，产生各个模块工作所需要的时钟。

特点：

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

12.2 时钟架构

12.2.1 时钟树

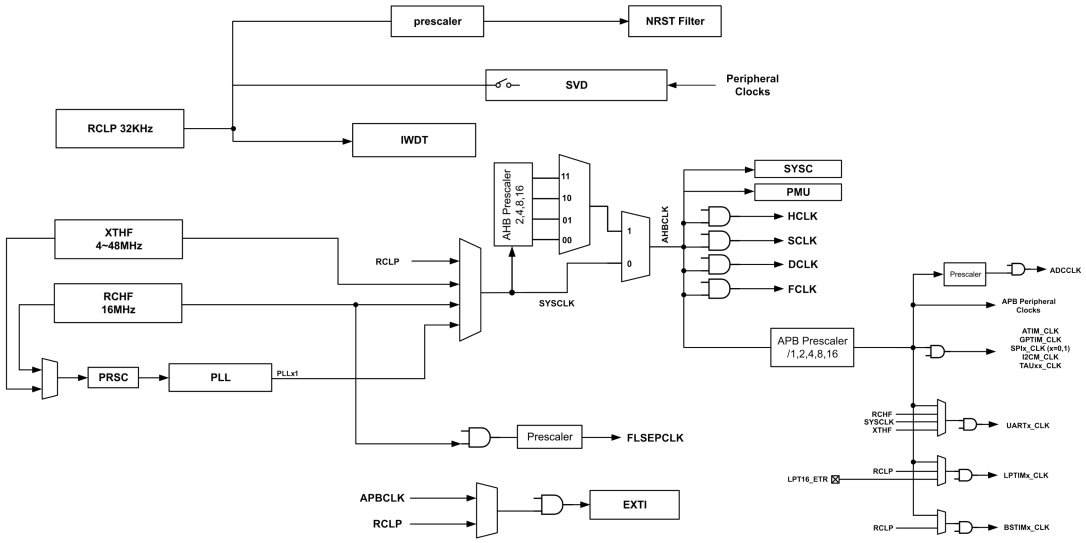


图 12-1 芯片时钟框图

系统主时钟(SYSCLK)可由 RCHF、XTHF、PLL、RCLP 及它们的分频时钟产生。上电默认使用 8MHz RCHF 的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。

12.2.2 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
RCLP	–

12.2.3 时钟安全性

当使用 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。专用的停振检测电路（HFDET）和 CLM 都可以用来监控系统时钟源的工作状态，并根据软件配置采取不同的策略。

SYSC_SWLOC 设置为允许自动切换到 RCHF 时，XTHF 如果发生时钟停振，停振检测输出将强制启动 RCHF，并将 SYSCLK 切换逻辑复位成 RCHF 通路。

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

以上切换发生的同时，停振检测电路会产生停振中断，通知软件处理异常。

SYSC_SWLOC 设置为禁止自动切换到 RCHF 时，XTHF 如果发生时钟停振，系统将没有时钟，等待 XTHF 起振。

12.2.4 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。

表 12-1 外设模块时钟列表

模块	总线时钟	工作时钟
PMU	AHBCLK	AHBCLK
DMA	AHBCLK	AHBCLK
NVMIF	AHBCLK	读:AHBCLK 擦写：RCHF
PAD	AHBCLK	AHBCLK/RCLP
IWDT	APBCLK	RCLP

模块	总线时钟	工作时钟
WWDT	APBCLK	APBCLK
HDIV	APBCLK	APBCLK
CRC	APBCLK	APBCLK
PGL	APBCLK	APBCLK
I2C	APBCLK	APBCLK
COMP	APBCLK	APBCLK
ADC	APBCLK	APBCLK
SPI	APBCLK	APBCLK
TAU	APBCLK	APBCLK
GPTIM	APBCLK	APBCLK
ATIM	APBCLK	APBCLK
BTIM	APBCLK	APBCLK/RCLP
LPTIM	APBCLK	APBCLK/RCLP
UART	APBCLK	LIN:APBCLK/RCHF/SYSCLK/XTHF 智能卡：APBCLK

12.2.5 休眠模式下的外设时钟

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

12.3 高速片上环振（RCHF）

RCHF 可配置输出频率为 8MHz、16MHz、24MHz 和 32MHz；出厂前，RCHF 在常温下被校准到标称频率的+/-0.5%以内，全温区频率变化量参见数据手册。

芯片复位后，默认使用 RCHF=8MHz 为系统时钟，并且自动装载 8MHz 的出厂校准参数；软件可以选择其他频率，并手动状态其他频率下的出厂校准参数，以获得常温下准确的环振频率。

软件修改 RCHF 输出频率的操作步骤：

- 改写 RCHF 输出频率寄存器
- 从 Flash 的 FMSH 数据区读取频率调校值（分别对应 16/24/32MHz）
- 将频率调校值写入 RCHFTRIM 寄存器，即可得到目标频率误差小于+/-0.5%（常温）的时钟

12.4 高频晶体振荡电路(XTHF)

12.4.1 概述

XTHF 适配 4~48MHz 晶体。软件可以通过 XTHFEN 寄存器使能或关闭 XTHF 时钟。

12.4.2 停振检测（HFDET）

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。

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。

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。

12.5 低功耗 RC 振荡器(RCLP)

RCLP 典型输出频率为 32KHz，不可关闭。LSCLK 时钟固定使用 RCLP。

12.6 锁相环(PLL)

锁相环输入参考时钟可以是 RCHF 或 XTHF 分频，输入频率固定为 4MHz，最高输出频率 48MHz。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。

12.7 休眠唤醒的时钟处理

当芯片从 Sleep/DeepSleep 模式下唤醒时，硬件自动打开 RCHF 并恢复到休眠前设置的频率输出；同时将 SYSCLKSEL 寄存器复位成 00，将系统时钟选为 RCHF。通过配置 AHBPRES_WKUPRSTEN 寄存器位，AHBPRES 寄存器可以在唤醒后复位或保持，使用 RCHF 或者其分频时钟工作。

12.8 寄存器

CMU 基地址 (0x40002400)

地址	符号	名称
0x0000	CMU_SYSCLKCR	系统时钟配置寄存器
0x0004	CMU_RCHFCR	RCHF 控制寄存器
0x0008	CMU_RCHFTR	RCHF 调校寄存器
0x000C	CMU_PLLCR	PLL 控制寄存器
0x0014	CMU_RCLPTR	RCLP TRIM 寄存器
0x0020	CMU_XTHFCR	XTHF 控制寄存器
0x002C	CMU_IER	时钟控制中断使能寄存器
0x0030	CMU_ISR	时钟控制中断标志寄存器
0x0034	CMU_PCLKCR1	外设总线时钟控制寄存器 1
0x0038	CMU_PCLKCR2	外设总线时钟控制寄存器 2
0x003C	CMU_PCLKCR3	外设总线时钟控制寄存器 3
0x0040	CMU_PCLKCR4	外设总线时钟控制寄存器 4
0x0044	CMU_OPCCR1	外设工作时钟控制寄存器 1
0x004C	CMU_OPCCR3	外设工作时钟控制寄存器 3
0x0050	CMU_AHBMCR	AHB Master 控制寄存器

12.8.1 系统时钟配置寄存器(CMU_SYSCLKCR)

符号	CMU_SYSCLKCR

符号	CMU_SYSCLKCR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	SYSC_SWLOC	RFU				RFU	SLP_ENEXTI	CKSAFE_CFG
权限	RW-0	R-0000				R-0	RW-1	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					APBPRES		
权限	R-00000					RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			AHBPRES_WKUPRSTEN	RFU	AHBPRES		
权限	R-000			RW-0	R-0	RW-011		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	STCLKSEL		RFU	RFU		SYSCLKSEL		
权限	RW-00		R-0	R-00		RW-000		

位	位名	权限	功能描述
31	SYSC_SWLOC	RW	SYSCLK 在 XTHF 停振时是否自动切换 0：禁止自动切换到 RCHF 1: 允许自动切换到 RCHF
30:27	RFU	R	RFU：未实现，读为 0
26	RFU	R	RFU：未实现，读为 0
25	SLP_ENEXTI	RW	SLEEP/DEEPSLEEP 模式下 EXTI 采样设置 1：SLEEP/DEEPSLEEP 模式下使能外部引脚中断采样（采样时钟为 RCLP） 0：SLEEP/DEEPSLEEP 模式下禁止外部引脚中断采样（将无法产生 EXTI 中断）
24	CKSAFE_CFG	RW	停振时 RCHF_FSEL 复位使能 0：停振时复位 1：停振时不复位 注意：当该位使能时，如果 RCHF_FSEL>24M，需要合理的配置 ahbpres，并设置对应 WKFSEL，否则可能导致唤醒时间不够，引起硬件故障
23:19	RFU	R	RFU：未实现，读为 0
18:16	APBPRES	RW	APB1 时钟分频选择 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
15:13	RFU	R	RFU：未实现，读为 0
12	AHBPRES_WKUPRSTEN	RW	唤醒后 AHB 时钟分频是否复位 1：不复位；0：复位

位	位名	权限	功能描述
11	RFU	R	RFU：未实现，读为 0
10:8	AHBPRES	RW	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
7:6	STCLKSEL	RW	CPU 内核 systick 工作时钟选择 00：SCLK 01：LSCLK 10：RFU 11：SYSCLK
5	RFU	R	RFU：未实现，读为 0
4:3	RFU	R	RFU：未实现，读为 0
2:0	SYSCLKSEL	RW	系统时钟源选择 000：RCHF 001：XTHF 010：PLL_DIV 011：RCHF 100：RCHF 101：RCHF 110：RCLP 111：RCHF

12.8.2 RCHF 控制寄存器(CMU_RCHFCR)

符号	CMU_RCHFCR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				FSEL			
权限	R-0000				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							EN
权限	R-0000000							RW-1

位	位名	权限	功能描述
31:20	RFU	R	RFU：未实现，读为 0
19:16	FSEL	RW	RCHF 频率选择寄存器 0000:8MHz

位	位名	权限	功能描述
			0001:16MHz 0010:24MHz 0011:32MHz 其他：禁止使用 在唤醒时，根据 WKFSEL 的设定异步复位、置位高两位不参与自动分频的判定逻辑，仅做备用且唤醒时异步清零
15:1	RFU	R	RFU：未实现，读为 0
0	EN	RW	RCHF 使能寄存器 1：使能 RCHF 0：关闭 RCHF

12.8.3 RCHF 调校寄存器(CMU_RCHFTR)

符号	CMU_RCHFTR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RCHFTRIM							
权限	RW-10000000							

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:0	RCHFTRIM	RW	RCHF 频率调校寄存器，8’h00 表示频率最低，8’hFF 表示频率最高 上电后芯片自动从 NVR1 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 NVR1 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。

12.8.4 PLL 控制寄存器(CMU_PLLCR)

符号	CMU_PLLCR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU					OPSC		

符号	CMU_PLLCR							
权限	R-00000					RW-000		
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU	DB						
权限	R-0	RW-0000111						
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ICFG				RTRIM			
权限	RW-0011				RW-1000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LOCKED	REFPRSC			RFU		INSEL	EN
权限	R-0	RW-000			R-00		RW-0	RW-0

位	位名	权限	功能描述
31:27	RFU	R	RFU：未实现，读为 0
26:24	OPSC	RW	PLL 分频，用于将分频后的输出作为主时钟的选项，默认不分频，分频数=PLLO_PSC+1
23	RFU	R	RFU：未实现，读为 0
22:16	DB	RW	PLL 倍频比，参考时钟频率为 Fin，则输出时钟频率为 Fout=4M*(PLLDB+1) 默认是 0x07
15:12	ICFG	RW	电流 cfg 信号，默认 0011
11:8	RTRIM	RW	电阻 trim 控制信号，默认 1000
7	LOCKED	R	PLL 锁定标志 1：PLL 已锁定 0：PLL 未锁定
6:4	REFPRSC	RW	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 010：3 分频 011：4 分频 100：6 分频 101：8 分频 110：10 分频 111：12 分频
3:2	RFU	R	未实现，读为 0
1	INSEL	RW	PLL 输入选择寄存器，在 PLL 已经使能后禁止修改 0：RCHF 1：XTHF
0	EN	RW	PLL 使能寄存器 1：使能 PLL 0：关闭 PLL 注意：当芯片进入 SLEEP/DEEPSLEEP 时，硬件关闭 PLL

12.8.5 RCLP TRIM 寄存器(CMU_RCLPTR)

符号	CMU_RCLPTR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIM							
权限	RW-10000000							

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:0	TRIM	RW	RCLP 调校值寄存器 0000 0000：频率最低 1111 1111：频率最高

12.8.6 XTHF 控制寄存器(CMU_XTHFCR)

符号	CMU_XTHFCR							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WAIT			RFU	CFG			
权限	RW-000			R-0	RW-1000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			BYPASS	RFU		RDY	EN
权限	R-000			RW-0	R-00		R-0	RW-0

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:13	WAIT	RW	XTHF 启动等待时间配置，单位是 XTHF 时钟周期 000：128

位	位名	权限	功能描述
			001：256 010：512 011：1024 100：2048 101：4096 110：8192 111：16384
12	RFU	R	RFU：未实现，读为 0
11:8	CFG	RW	XTHF 振荡强度配置 0000：最弱 1111：最强
7:5	RFU	R	RFU：未实现，读为 0
4	BYPASS	RW	XTHF BYPASS 寄存器 0：XTHF BYPASS 无效 1：XTHF BYPASS 有效 BYPASS 后可以从 XTHF_IN 管脚输入时钟，XTHF_OUT 可以配置为 GPIO 使用
3:2	RFU	R	RFU：未实现，读为 0
1	RDY	R	XTHF 稳定标志，硬件置位，只读 关闭 XTHF 或者 XTHF 发生停振时自动清零
0	EN	RW	XTHF 使能寄存器 0：关闭 XTHF 1：使能 XTHF

12.8.7 时钟控制中断使能寄存器(CMU_IER)

符号	CMU_IER							
地址	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					SYSCSE_IE	HFDET_IE	RFU
权限	R-00000					RW-0	RW-0	R-0

位	位名	权限	功能描述
31:3	RFU	R	RFU：未实现，读为 0
2	SYSCSE_IE	RW	SYSCLK 选择错误中断使能，1 有效

位	位名	权限	功能描述
1	HFDET_IE	RW	XTHF 频率检测报警中断使能，1 有效
0	RFU	R	RFU：未实现，读为 0

12.8.8 时钟控制中断标志寄存器(CMU_ISR)

符号	CMU_ISR							
地址	0x0030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						HFDETO	RFU
权限	R-000000						R-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					SYSCSE_IF	HFDET_IF	RFU
权限	R-00000					W1C-0	W1C-0	R-0

位	位名	权限	功能描述
31:10	RFU	R	RFU：未实现，读为 0
9	HFDETO	R	高频晶体停振检测模块输出 1：XTHF 未停振 0：XTHF 停振
8:3	RFU	R	RFU：未实现，读为 0
2	SYSCSE_IF	W1C	SYSCLK 选择错误中断标志，1 有效 当所选择的时钟源未处于开启状态，则置起
1	HFDET_IF	W1C	XTHF 频率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 HFDETO 不为 0 的情况下才能够清除此寄存器
0	RFU	R	RFU：未实现，读为 0

12.8.9 外设总线时钟控制寄存器 1(CMU_PCLKCR1)

符号	CMU_PCLKCR1							
地址	0x0034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

符号	CMU_PCLKCR1							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						COMP_PCE	SVD_PCE
权限	R-000000						RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PAD_PCE	ANT_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RFU	LPT_PCE	RFU
权限	RW-0	RW-0	RW-0	RW-1	RW-1	R-0	RW-0	R-0

位	位名	权限	功能描述
31:10	RFU	R	RFU：未实现，读为 0
9	COMP_PCE	RW	Comparator 总线时钟使能，高使能
8	SVD_PCE	RW	SVD 总线时钟使能，高使能
7	PAD_PCE	RW	PADCFG 总线时钟使能，高使能
6	ANT_PCE	RW	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	RW	IWDT 总线时钟使能，高使能
4	SCU_PCE	RW	SCU 总线时钟使能，高使能
3	PMU_PCE	RW	PMU 总线时钟使能，高使能
2	RFU	R	RFU：未实现，读为 0
1	LPT_PCE	RW	LPTIM 总线时钟使能，高使能
0	RFU	R	RFU：未实现，读为 0

12.8.10 外设总线时钟控制寄存器 2(CMU_PCLKCR2)

符号	CMU_PCLKCR2							
地址	0x0038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				PGL_PCE	RFU	DIVAS_PCE	ADC_PCE
权限	R-0000				RW-0	R-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDT_PCE	RAMBIST_PCE	NVM_PCE	DMA_PCE	RFU			CRC_PCE
权限	RW-0	RW-0	RW-0	RW-0	R-000			RW-0

位	位名	权限	功能描述

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11	PGL_PCE	RW	PGL 总线时钟使能，高使能
10	RFU	R	RFU：未实现，读为 0
9	DIVAS_PCE	RW	DIVAS 总线时钟使能，高使能
8	ADC_PCE	RW	ADC 总线时钟使能，高使能
7	WWDT_PCE	RW	WWDT 总线时钟使能，高使能
6	RAMBIST_PCE	RW	RAMBIST 总线时钟使能，高使能
5	NVM_PCE	RW	NVMIF（Flash 擦写控制器）总线时钟使能，高使能
4	DMA_PCE	RW	DMA 总线时钟使能，高使能
3:1	RFU	R	RFU：未实现，读为 0
0	CRC_PCE	RW	CRC 总线时钟使能，高使能

12.8.11 外设总线时钟控制寄存器 3(CMU_PCLKCR3)

符号	CMU_PCLKCR3							
地址	0x003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						I2C1_PCE	I2C0_PCE
权限	R-000000						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	UARTIR_PCE	UART5_PCE	UART4_PCE	RFU	UART2_PCE	UART1_PCE	RFU
权限	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						SPI1_PCE	SPI0_PCE
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	I2C1_PCE	RW	I2C1 总线时钟使能，高有效
24	I2C0_PCE	RW	I2C0 总线时钟使能，高有效
23:15	RFU	R	RFU：未实现，读为 0
14	UARTIR_PCE	RW	UARTIR 总线时钟使能，高有效
13	UART5_PCE	RW	UART5 总线时钟使能，高有效
12	UART4_PCE	RW	UART4 总线时钟使能，高有效
11	RFU	R	RFU：未实现，读为 0
10	UART2_PCE	RW	UART2 总线时钟使能，高有效
9	UART1_PCE	RW	UART1 总线时钟使能，高有效
8:2	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
1	SPI1_PCE	RW	SPI1 总线时钟使能，高有效
0	SPI0_PCE	RW	SPI0 总线时钟使能，高有效

12.8.12 外设总线时钟控制寄存器 4(CMU_PCLKCR4)

符号	CMU_PCLKCR4							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			GPTIM_PCE	RFU			BT16_PCE
权限	R-000			RW-0	R-000			RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	RFU	TAU1_PCE	TAU0_PCE	RFU	RFU	RFU	ATIM_PCE
权限	R-0	R-0	RW-0	RW-0	R-0	R-0	R-0	RW-0

位	位名	权限	功能描述
31:13	RFU	R	RFU：未实现，读为 0
12	GPTIM_PCE	RW	GTIMER 总线时钟使能，高有效
11:9	RFU	R	RFU：未实现，读为 0
8	BT16_PCE	RW	BSTIM 总线时钟使能，高有效
7	RFU	R	RFU：未实现，读为 0
6	RFU	R	RFU：未实现，读为 0
5	TAU1_PCE	RW	TAU1 总线时钟使能，高有效
4	TAU0_PCE	RW	TAU0 总线时钟使能，高有效
3	RFU	R	RFU：未实现，读为 0
2	RFU	R	RFU：未实现，读为 0
1	RFU	R	RFU：未实现，读为 0
0	ATIM_PCE	RW	ATIMER 总线时钟使能，高有效

12.8.13 外设工作时钟控制寄存器 1(CMU_OPCCR1)

符号	CMU_OPCCR1							
地址	0x0044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU	EXTICKS	RFU					
权限	R-0	RW-0	R-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	CMU_OPCCR1							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BTCKS		RFU		LPTCKS		RFU	
权限	RW-00		R-00		RW-00		R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				UART1CKS		RFU	
权限	R-0000				RW-00		R-00	

位	位名	权限	功能描述
31	RFU	R	RFU：未实现，读为 0
30	EXTICKS	RW	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
29:16	RFU	R	RFU：未实现，读为 0
15:14	BTCKS	RW	BSTIM 工作时钟源选择 00：APBCLK 01：LSCLK 10：RCLP 11：APBCLK
13:12	RFU	R	RFU：未实现，读为 0
11:10	LPTCKS	RW	LPTIM 工作时钟选择 00：APBCLK 01：LSCLK 10：RCLP 11：APBCLK
9:4	RFU	R	RFU：未实现，读为 0
3:2	UART1CKS	RW	UART1 工作时钟选择 00：APBCLK 01：RCHF 10：SYSCLK 11：XTHF
1:0	RFU	R	RFU：未实现，读为 0

12.8.14 外设工作时钟控制寄存器 3(CMU_OPCCR3)

符号	CMU_OPCCR3							
地址	0x004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTICKE	FLASHCKE	RFU					
权限	RW-0	RW-0	R-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	CMU_OPCCR3							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ATIM0CKE	RFU				LINCKE	UART1CKE	RFU
权限	RW-0	R-0000				RW-0	RW-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				BTCKE	RFU	LPTCKE	RFU
权限	R-0000				RW-0	R-0	RW-0	R-0

位	位名	权限	功能描述
31	EXTICKE	RW	外部引脚中断采样时钟使能，高有效
30	FLASHCKE	RW	Flash 擦写时钟使能，高有效
29:16	RFU	R	RFU：未实现，读为 0
15	ATIM0CKE	RW	ATIMER 工作时钟使能，高有效
14:11	RFU	R	RFU：未实现，读为 0
10	LINCKE	RW	LIN 工作时钟使能，高有效
9	UART1CKE	RW	UART1 工作时钟使能，高有效
8	RFU	R	RFU：未实现，读为 0
7:4	RFU	R	RFU：未实现，读为 0
3	BTCKE	RW	BSTIM 工作时钟使能，高有效
2	RFU	R	RFU：未实现，读为 0
1	LPTCKE	RW	LPTIM 工作时钟使能，高有效
0	RFU	R	RFU：未实现，读为 0

12.8.15 AHB Master 控制寄存器(CMU_AHBMCR)

符号	CMU_AHBMCR							
地址	0x0050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	REV		RFU					
权限	RW-10		R-000000					
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							MPRIL
权限	R-0000000							RW-1

位	位名	权限	功能描述

位	位名	权限	功能描述
31:30	REV	RW	保留位，无功能
29:1	RFU	R	RFU：未实现，读为 0
0	MPRIL	RW	AHB Master 优先级配置寄存器 0：DMA 优先 1：CPU 优先

13 模拟比较器（Comparator）

13.1 概述

芯片集成 1 个比较器，支持以下特性：

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

13.2 结构框图

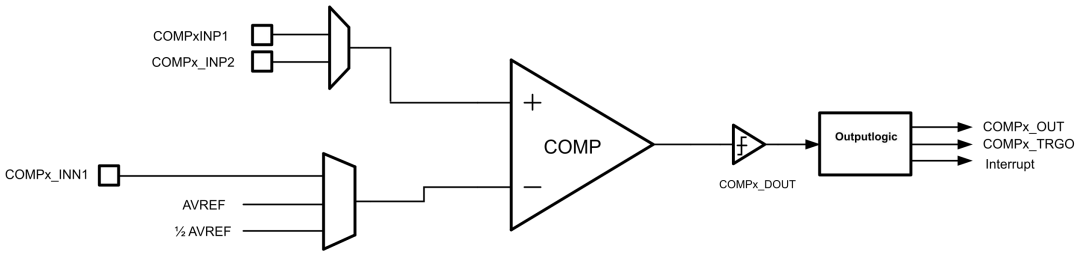


图 13-1 比较器电路框图

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。

13.3 功能描述

13.3.1 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。

13.3.2 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 时，必须使能 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。

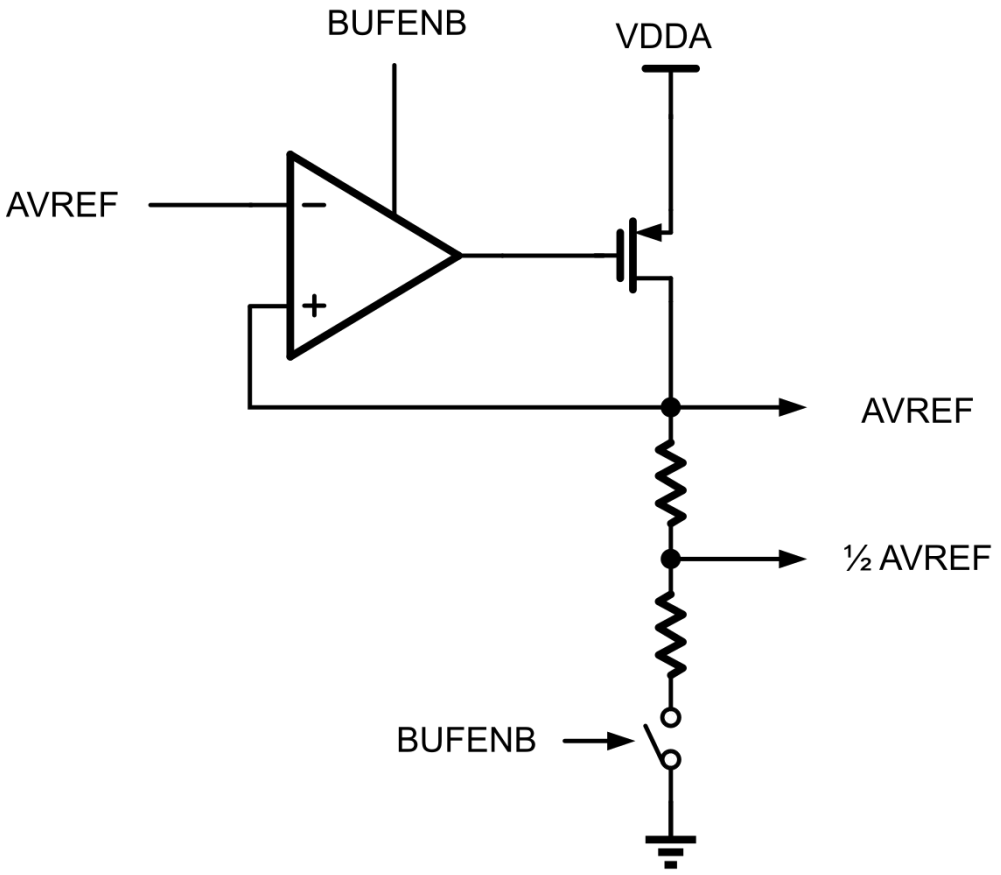


图 13-2 比较器内建基准缓冲器

13.3.3 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。

13.3.4 引脚和内部信号连接

使用外部引脚输入时需要将对应的 GPIO 配置为模拟通道功能，比较器的输入管脚映射参考引脚和封装。

13.3.5 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：

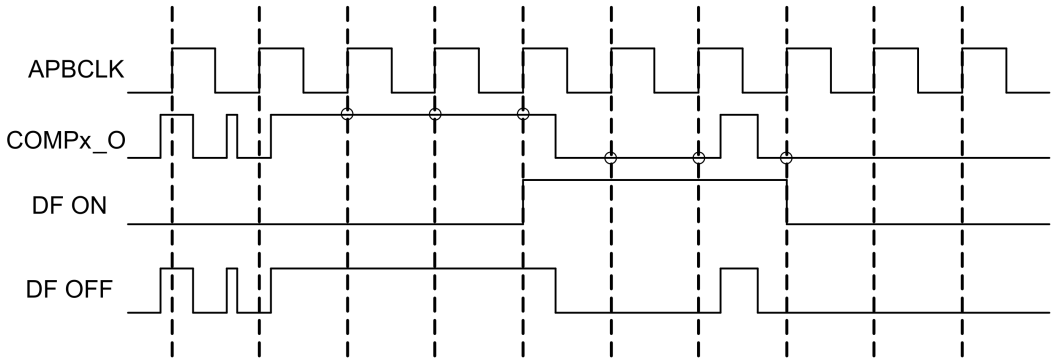


图 13-3 数字滤波（DFLEN=3）波形示意图

13.3.6 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。

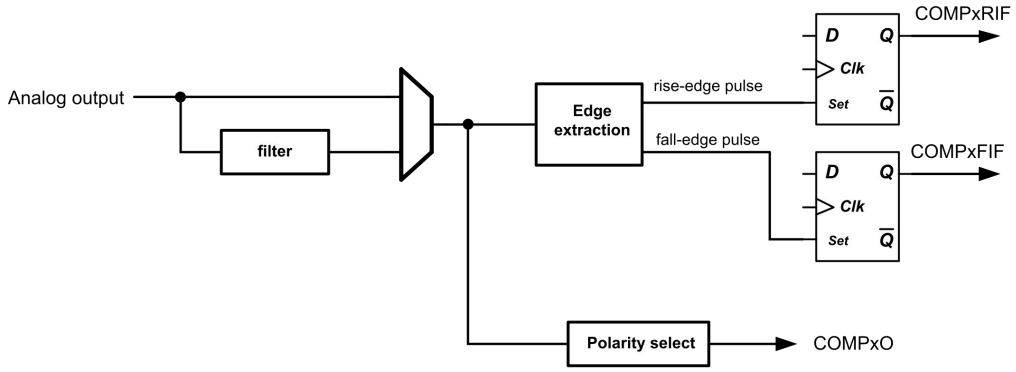


图 13-4 比较器中断产生

13.3.7 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。

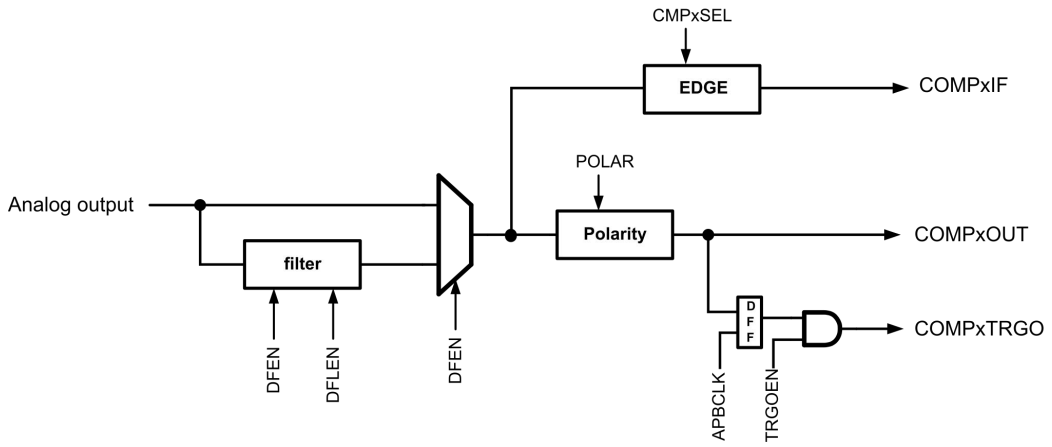


图 13-5 比较器输出逻辑

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。

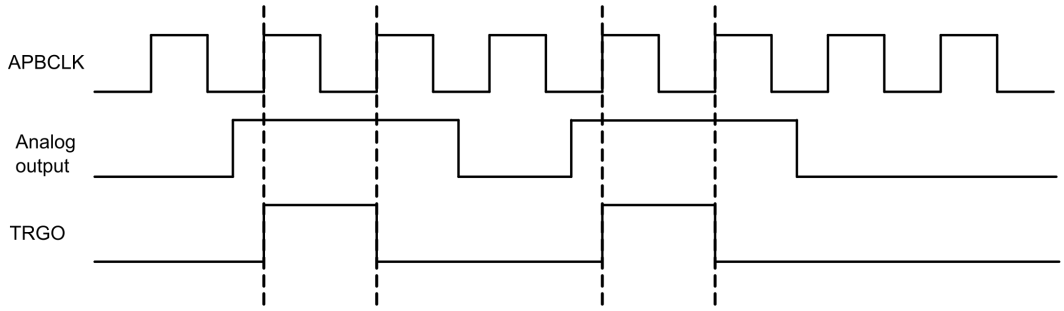


图 13-6 比较器输出上升沿产生触发输出，无滤波

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。

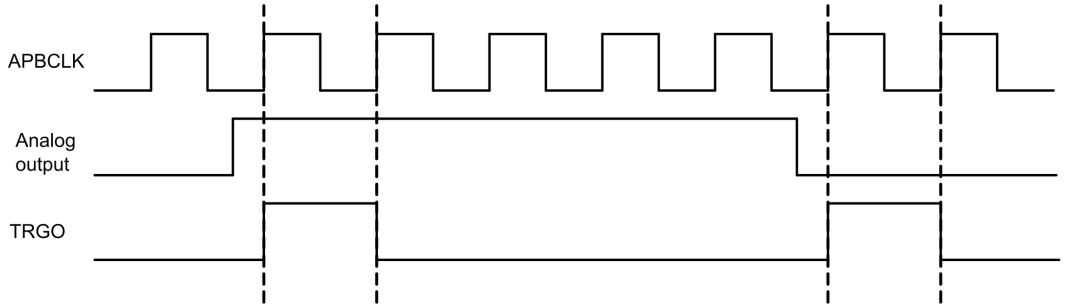


图 13-7 比较器输出上升沿下降沿产生触发输出，无滤波

13.3.8 比较器输出连接

比较器的输出可以被连接到 ATIMx 和 GPTIMx 的输入，以便定时器自动记录比较器输出翻转次数。具体连接关系请参考 GPTIM 和 ATIM 的引脚和内部连接章节。

比较器的输出也可以作为 PGL 的内部输入信号，请参考 PGL 的输入选择寄存器。

比较器的触发信号也可以作为启动 ADC 转换的触发信号，请参考 ADC 的触发源章节。

比较器的触发信号也可以作为 TAU 的触发信号，请参考 TAU 的捕捉源选择章节。

13.4 寄存器

COMP 基地址 (0x40019C00)

地址	符号	名称
0x0000	COMP_CR1	COMP1 控制寄存器
0x000C	COMP_ICR	COMP 中断配置寄存器
0x0010	COMP_ISR	COMP 中断标志寄存器
0x0014	COMP_BUFCR	COMP 缓冲器控制寄存器

13.4.1 COMP1 控制寄存器(COMP_CR1)

符号	COMP_CR1							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU					TRGOSEL		TRGOEN
权限	R-00000					RW-00		RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DFLEN					RFU	POLAR	DFEN
权限	RW-00000					R-0	RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							CMP1O
权限	R-0000000							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	V1PSEL			V1NSEL			CMP1EN
权限	R-0	RW-000			RW-000			RW-0

位	位名	权限	功能描述
31:27	RFU	R	RFU：未实现，读为 0
26:25	TRGOSEL	RW	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	RW	比较器 1 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	RW	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	RFU	R	RFU：未实现，读为 0
17	POLAR	RW	比较器 1 输出极性控制 0：正向输出 1：取反输出
16	DFEN	RW	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
15:9	RFU	R	RFU：未实现，读为 0
8	CMP1O	R	比较器 1 输出，软件只读
7	RFU	R	RFU：未实现，读为 0
6:4	V1PSEL	RW	比较器 1 正极输入选择 0：COMP1_INP1 1：COMP1_INP2 其他：RFU
3:1	V1NSEL	RW	比较器负极输入选择 000：COMP1_INN1 001：RFU 010：RFU 011：AVREF 011：AVREF/2 100：RFU 101：RFU 110：RFU 111：RFU
0	CMP1EN	RW	比较器 1 使能位 0：关闭比较器 1：使能比较器

13.4.2 COMP 中断配置寄存器(COMP_ICR)

符号	COMP_ICR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				CMP1SEL		RFU	CMP1IE
权限	R-0000				RW-00		R-0	RW-0

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3:2	CMP1SEL	RW	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	R	RFU：未实现，读为 0
0	CMP1IE	RW	比较器 1 中断使能，1 有效

13.4.3 COMP 中断标志寄存器(COMP_ISR)

符号	COMP_ISR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							CMP1IF
权限	R-0000000							W1C-0

位	位名	权限	功能描述

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	CMP1IF	W1C	比较器 1 中断标志，硬件置位，软件写 1 清零

13.4.4 COMP 缓冲器控制寄存器(COMP_BUFCR)

符号	COMP_BUFCR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					BYP	RFU	ENB
权限	R-00000					RW-0	R-0	RW-1

位	位名	权限	功能描述
31:3	RFU	R	RFU：未实现，读为 0
2	BYP	RW	缓冲器 bypass 使能 0：不 bypass 1：bypass
1	RFU	R	RFU：未实现，读为 0
0	ENB	RW	缓冲器使能 0：使能基准缓冲器 1：关闭基准缓冲器

14 硬件除法器（DIVAS）

14.1 概述

DIVAS 用于帮助软件加速除法运算。

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

乘法功能在 Cortex-M0 内核中完成。

特点：

- 有符号整数运算（二进制补码格式）
- 32bit 被除数、16bit 除数
- 输出 32bit 商和 32bit 余数
- 除以 0 警告
- 一次计算需要 8 个 24MHz 周期

14.2 工作流程

软件按照如下步骤调用硬件除法器。

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

14.3 寄存器

地址	符号	名称
0x0000	DIVAS_END	被除数寄存器
0x0004	DIVAS_SOR	除数寄存器
0x0008	DIVAS_QUOT	商寄存器
0x000C	DIVAS_REMD	余数寄存器
0x0010	DIVAS_SR	状态寄存器

14.3.1 被除数寄存器(DIVAS_END)

符号	DIVAS_END							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DIVEND							

符号	DIVAS_END							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DIVEND							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVEND							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVEND							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	DIVEND	RW	32bit 有符号被除数

14.3.2 除数寄存器(DIVAS_SOR)

符号	DIVAS_SOR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DIVSOR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DIVSOR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	DIVSOR	RW	16bit 有符号除数

14.3.3 商寄存器(DIVAS_QUOT)

符号	DIVAS_QUOT							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	QUOT							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	QUOT							

符号	DIVAS_QUOT							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	QUOT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	QUOT							
权限	R-00000000							

位	位名	权限	功能描述
31:0	QUOT	R	32bit 有符号商（仅地址，无实际寄存器，读取时直接返回 DW_div 模块输出）

14.3.4 余数寄存器(DIVAS_REMD)

符号	DIVAS_REMD							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	REMD							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REMD							
权限	R-00000000							

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

14.3.5 状态寄存器(DIVAS_SR)

符号	DIVAS_SR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	DIVAS_SR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						DIV_BY_0	BUSY
权限	R-000000						R-0	R-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	DIV_BY_0	R	除数为 0 标志 1 = 除数为 0 0 = 除数不为 0
0	BUSY	R	运算过程指示 1 = DIVAS 在计算过程中，结果未就绪 0 = 计算完毕，结果就绪 软件在写入操作数或除数后，DIVAS 开始计算，软件应查询 BUSY 为低后再读取商、余数寄存器

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：

- 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

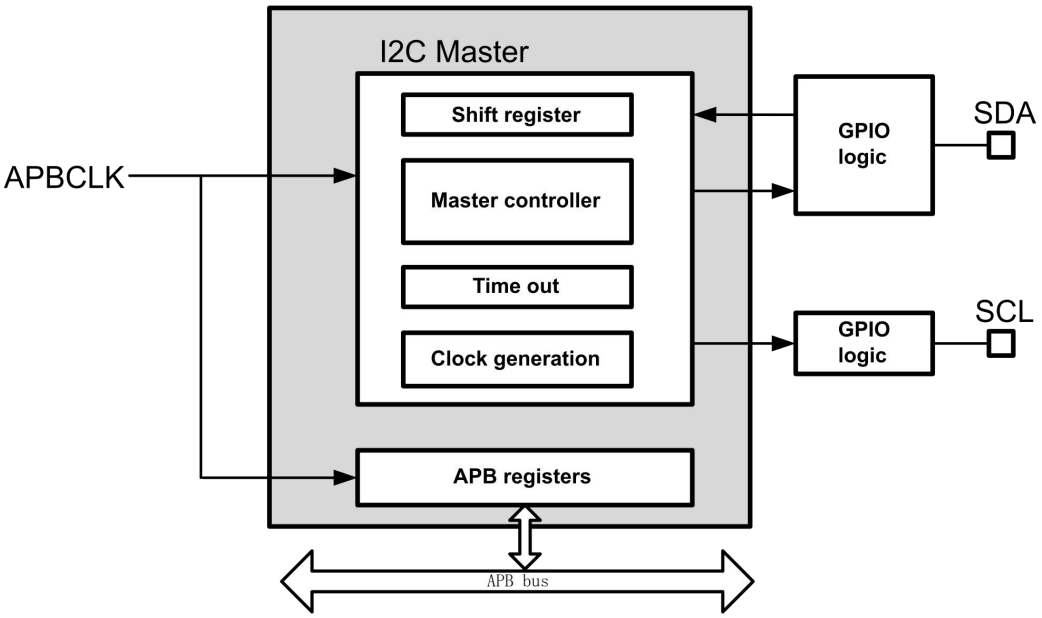


图 15-1 I²C 主机模块框图

15.3 引脚定义

I²C 主机模块使用 2 个引脚与外部器件通信，工作时需要配置为开漏模式，具体映射管脚参考引脚和封装。

15.4 时钟

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

15.5 接口时序

15.5.1 接口时序图

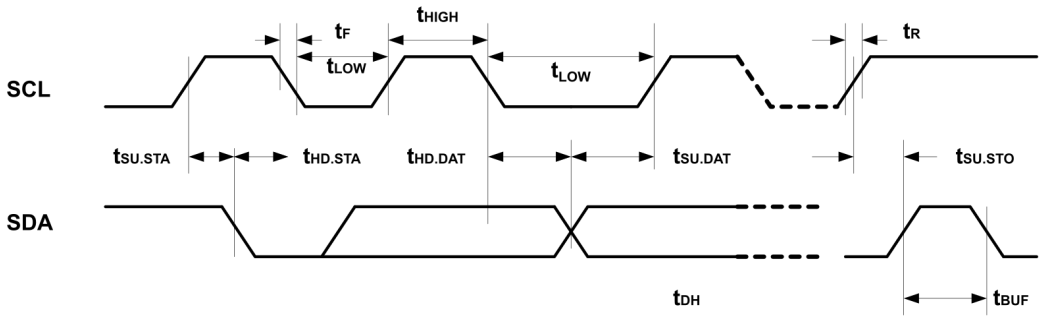


图 15-2 I²C 总线时序

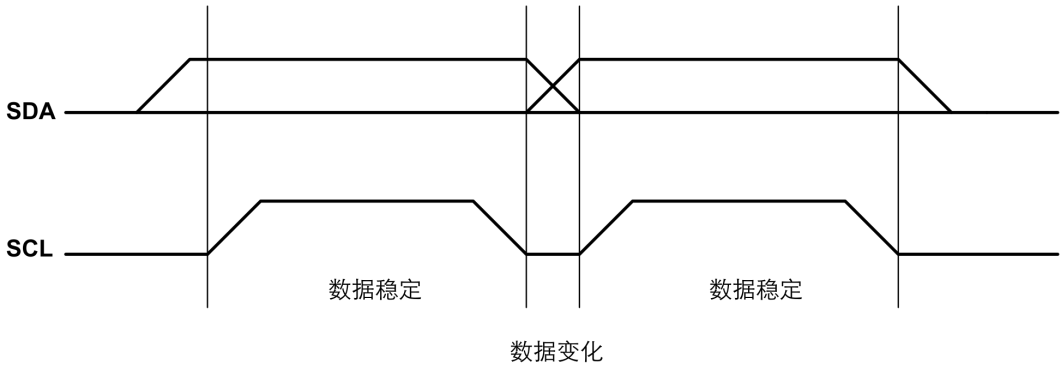


图 15-3 数据有效时序

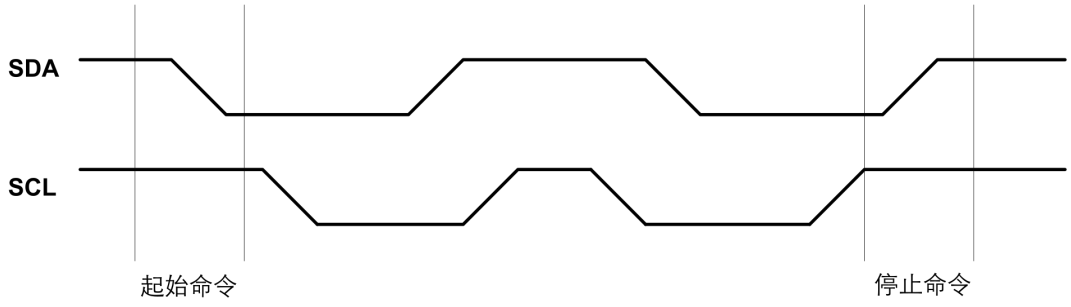


图 15-4 起始（Start）与停止(Stop)命令定义

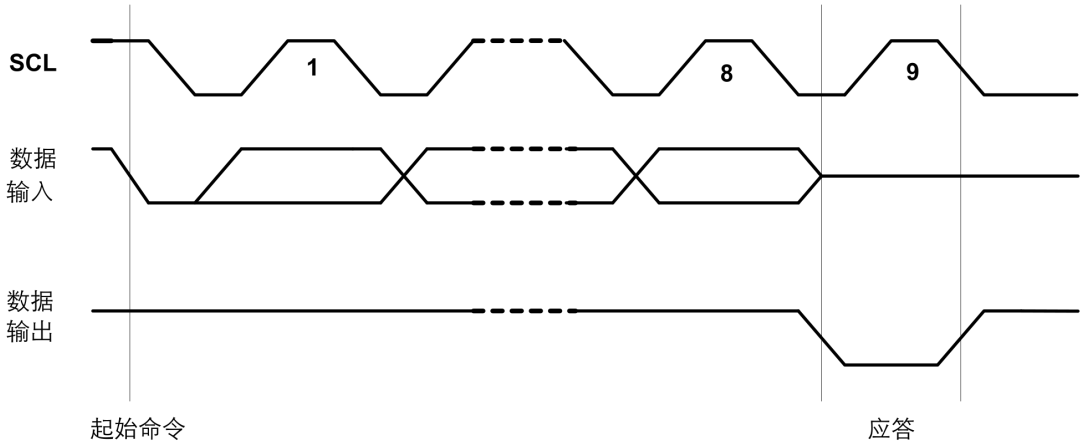


图 15-5 输出应答(ACK)

15.5.2 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。

I²C 接口时序要求参考电参数。

15.6 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。

15.7 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。

表 15-1 I²C 保留地址定义

从机地址	R/W_bit	描述

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

15.8 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

15.8.1 IO 配置

15.8.2 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。

15.9 I²C 主机功能

FM454 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。

15.9.1 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）。

表 15-2 I²C 寻址字节定义

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。

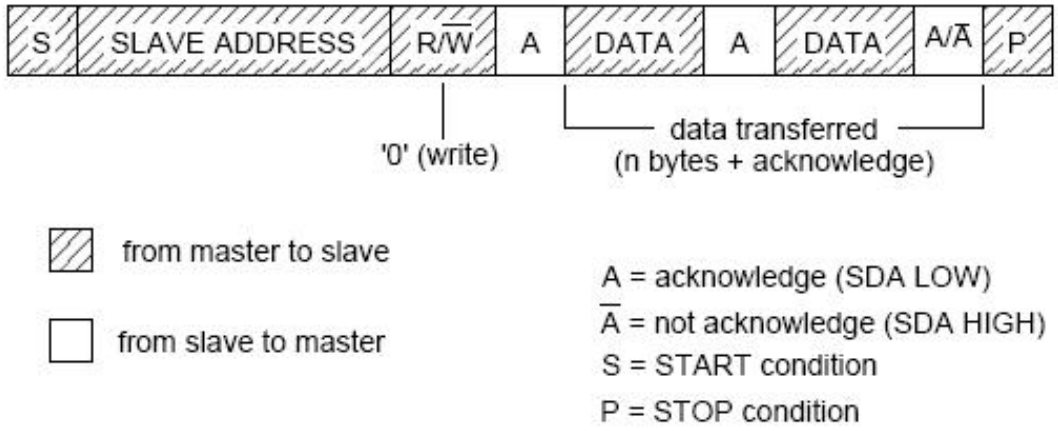


图 15-6 主机向 7 位地址从机写入数据时的帧格式

1. 主机清零 RWN 寄存器，并发起 START 时序
2. 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，发送数据时 R/W 位为 0
3. 主机发送第一帧 8 位数据
4. 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
5. 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
6. 主机完成所有数据发送后，发送 STOP 时序

软件启动 I2C 主机发送的操作流程如下图：

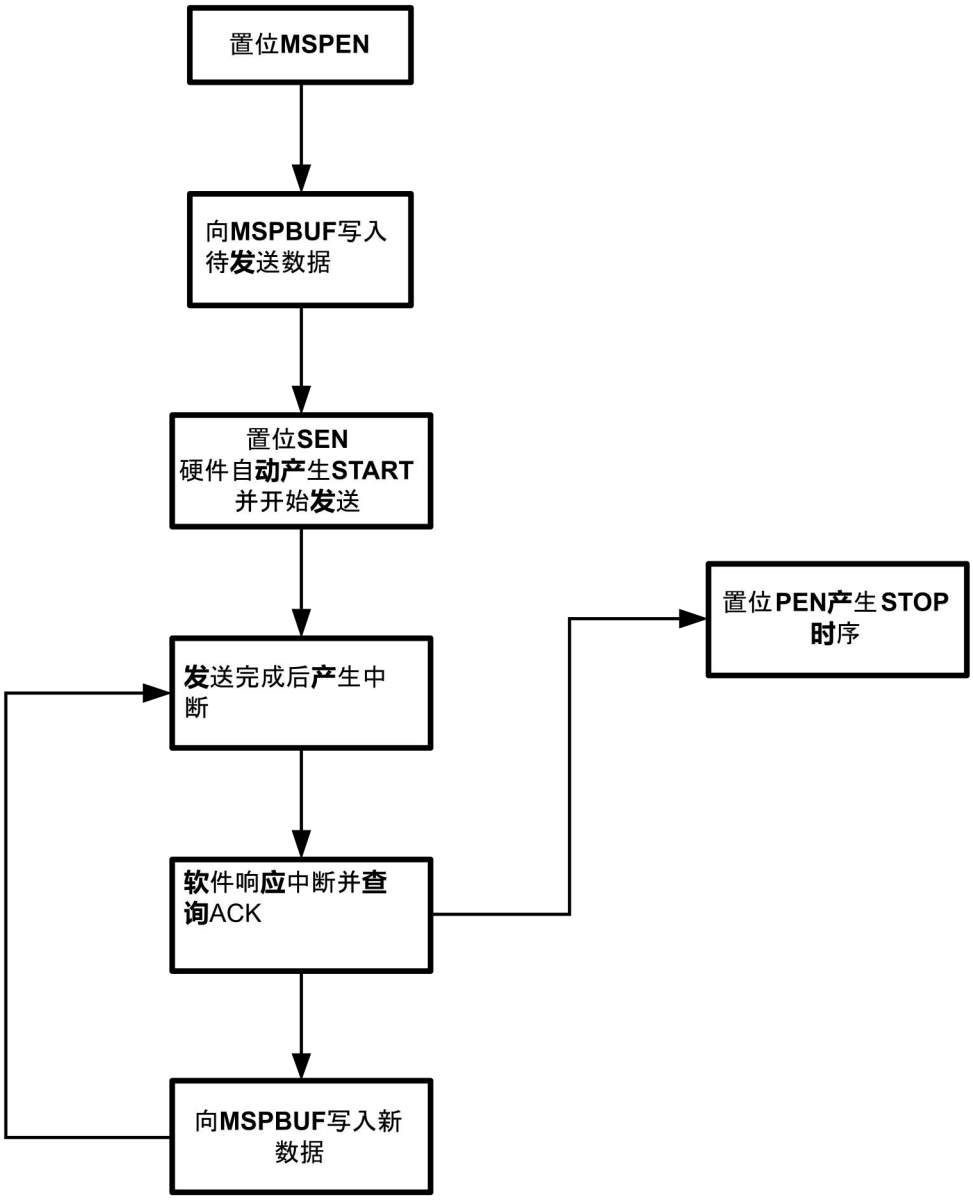


图 15-7 I²C 主机 7 位地址软件发送数据流图

I²C 主机对 7 位地址从机写入数据的波形示意图如下：

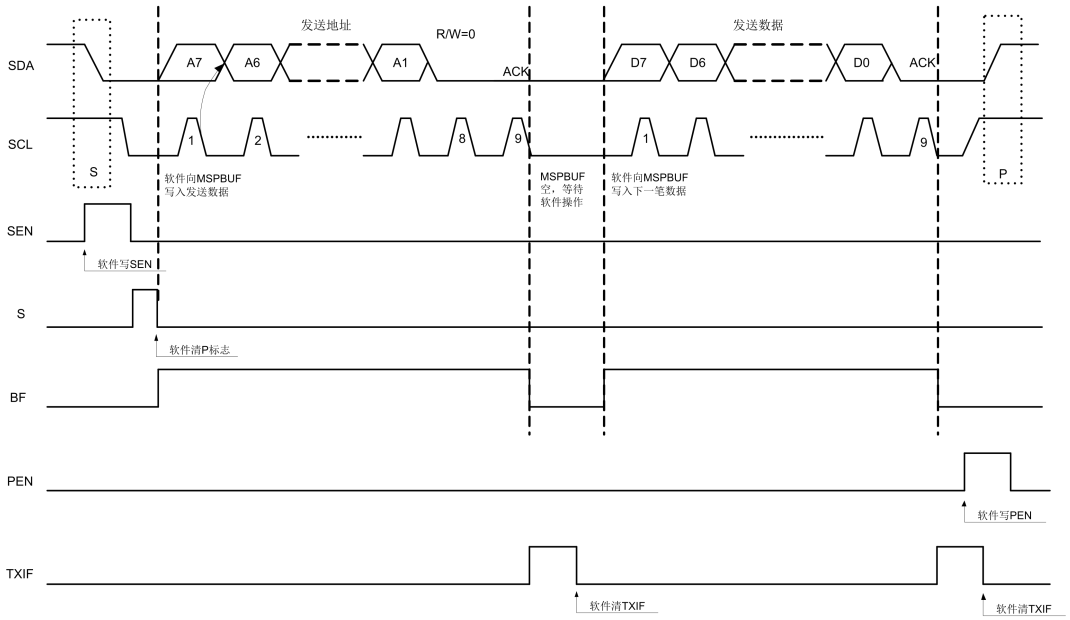


图 15-8 I²C 主机对 7 位地址从机发送数据流图

主机从从机读取数据

典型的 7bit 寻址，主机从从机读取数据的帧格式如下图所示。

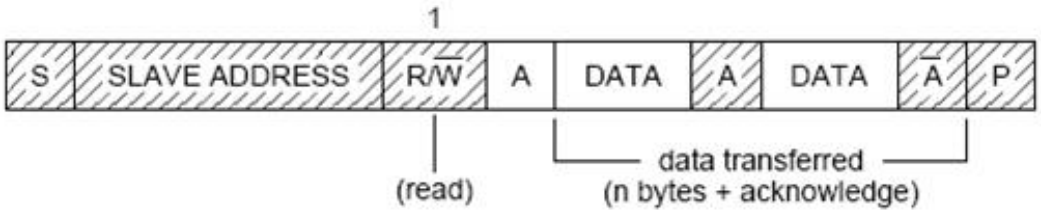


图 15-9 主机从 7 位地址从机读取数据时的帧格式

1. 主机发起 START 时序
2. 主机发送从机地址，从机地址包含 7 位从机地址和 1 位 R/W 标志位，数据读取时 R/W 位为 1
3. 此时设置 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

软件启动 I²C 接收的操作流程如下图：

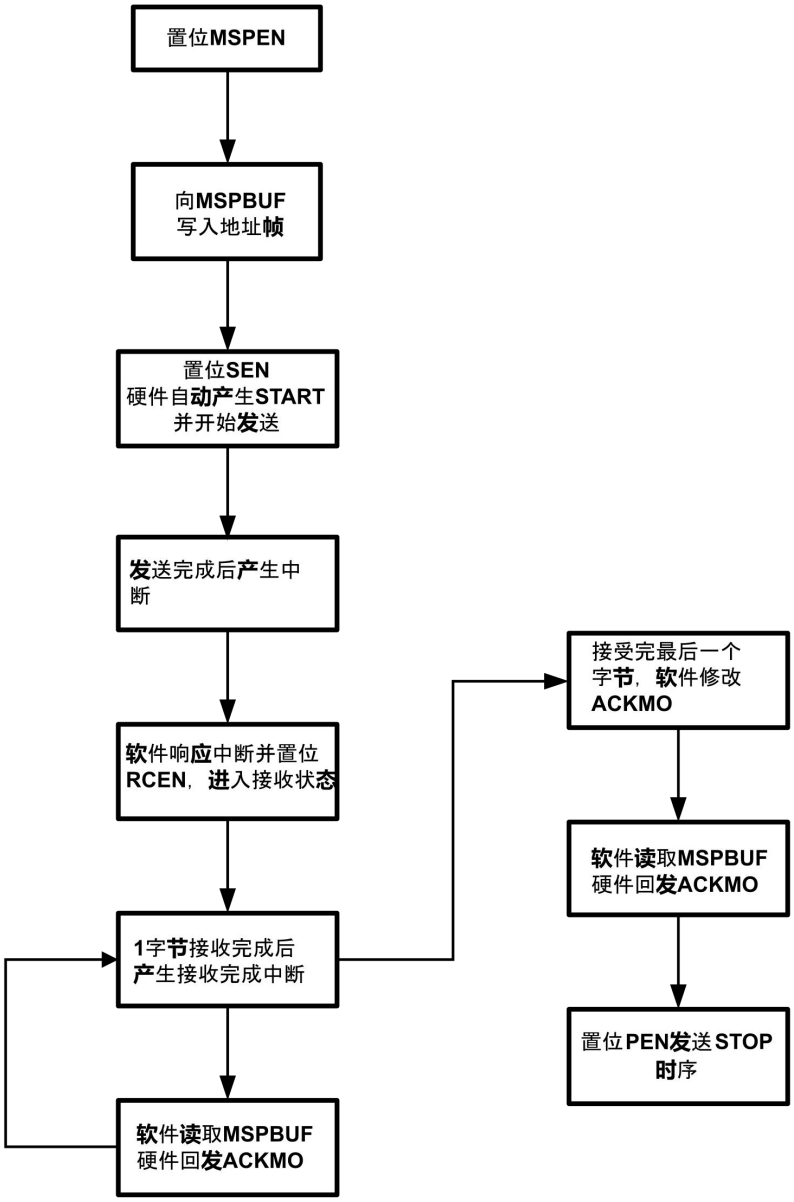


图 15-10 I²C 软件 7 位地址接收数据流图

主机每次接收完从机发送的数据后，根据 ACKMO 寄存器回发响应。ACKMO 复位值为 0，即默认状态下主机回发 ACK。如果软件希望主机在接收完成后回发 NACK，则需要在前一个字节接收完成中断中将 ACKMO 寄存器改写为 1。ACKMO 为 1 的情况下，主机在发送完响应后会自动清零 ACKMO。

I²C 主机从 7 位地址从机读出数据的波形示意图如下：

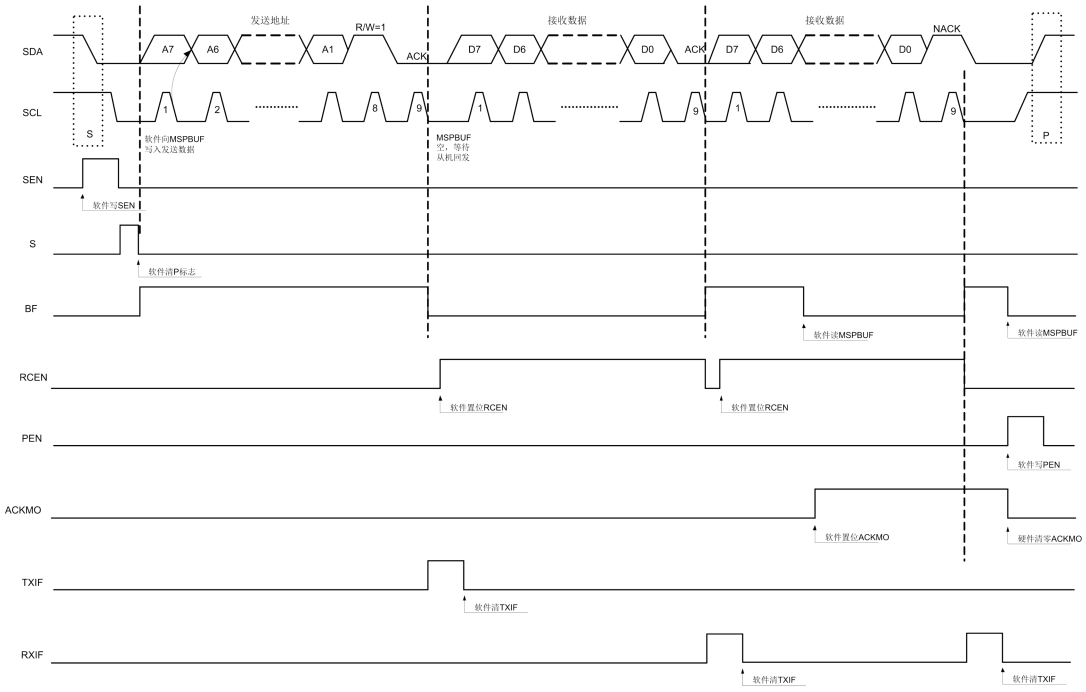


图 15-11 I²C 从 7 位地址从机读取数据流图

双向数据传输（组合模式）

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

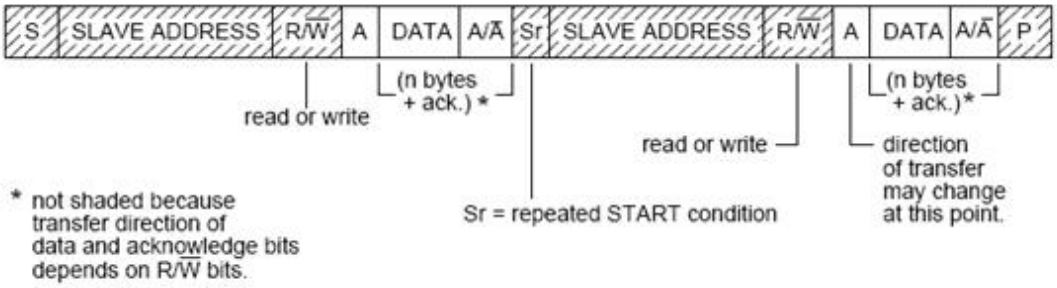


图 15-12 双向数据通信帧格式

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

15.9.2 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。

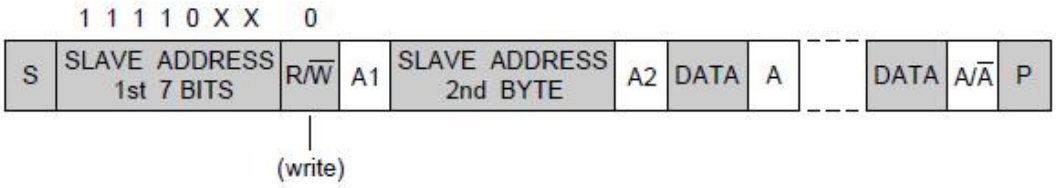


图 15-13 10bit 寻址，主机向从机写入数据

1. 主机清零 RWN 寄存器并发起 START 时序
2. 主机发送第一个从机地址字节，以 11110 开头，跟随 2bit 从机地址最高位，以及 R/W 标志位，发送数据时 R/W 位为 0
3. 主机检查从机回发的 ACK
4. 主机发送第二个从机地址字节，包含从机地址的低 8 位
5. 主机检查从机回发的 ACK
6. 主机继续向从机写入数据
7. 主机完成所有数据发送后，发送 STOP 时序

软件启动 I^2C 主机发送的操作流程如下图：

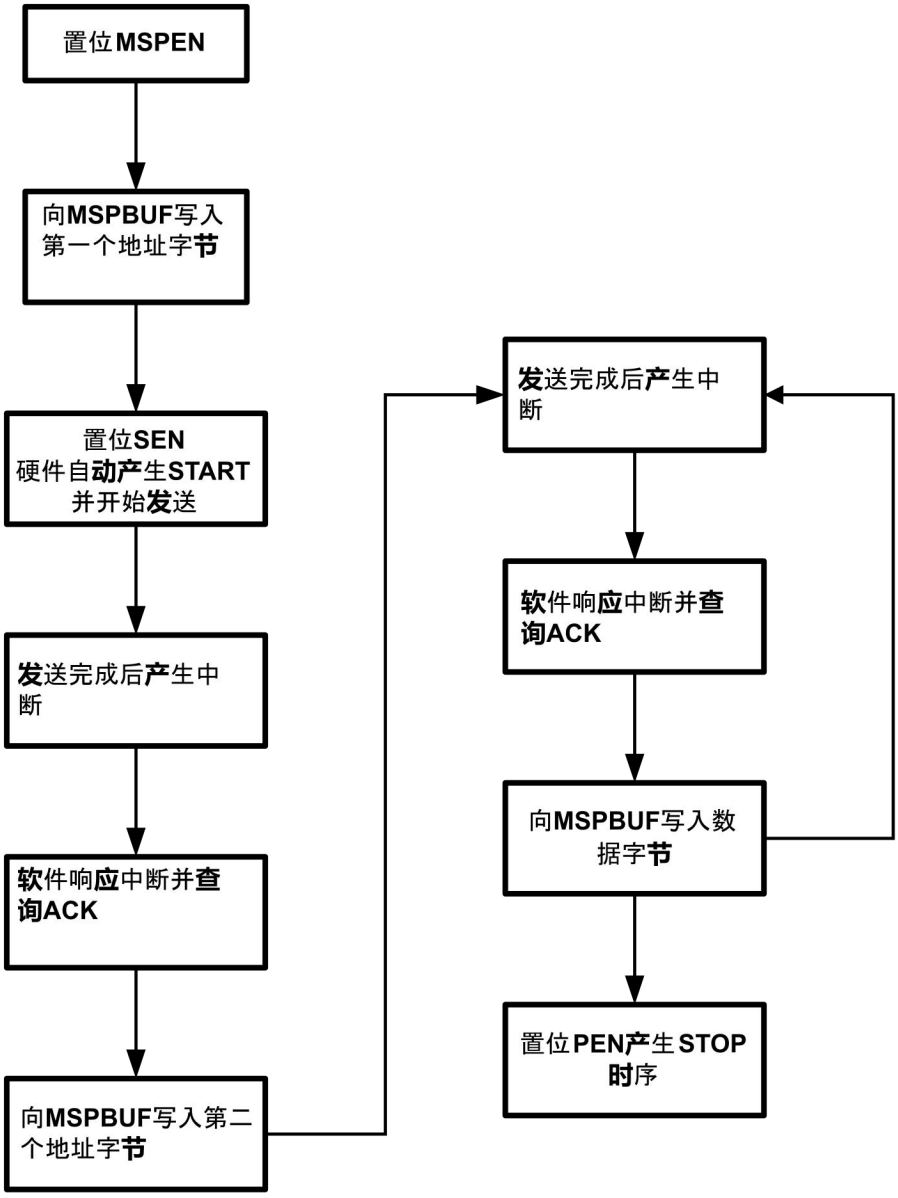


图 15-14 I²C 软件 10 位地址发送数据流图

主机从从机读取数据

典型的 10bit 寻址，主机从从机读取数据的数据流图如下图所示。

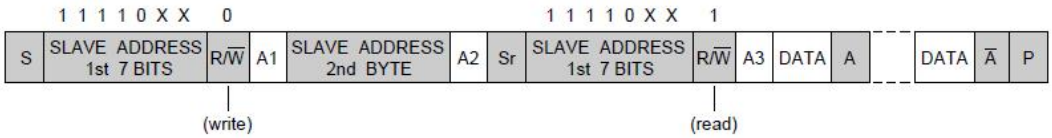


图 15-15 10bit 寻址，主机从从机读取数据

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

3. 主机发送第二字节从机地址，包含低 8 位地址
4. 主机发送 ReSTART 时序
5. 主机再次发送第一字节从机地址，将 R/W 为改为 0
6. 此时设置 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

软件启动 I²C 接收的操作流程如下图：

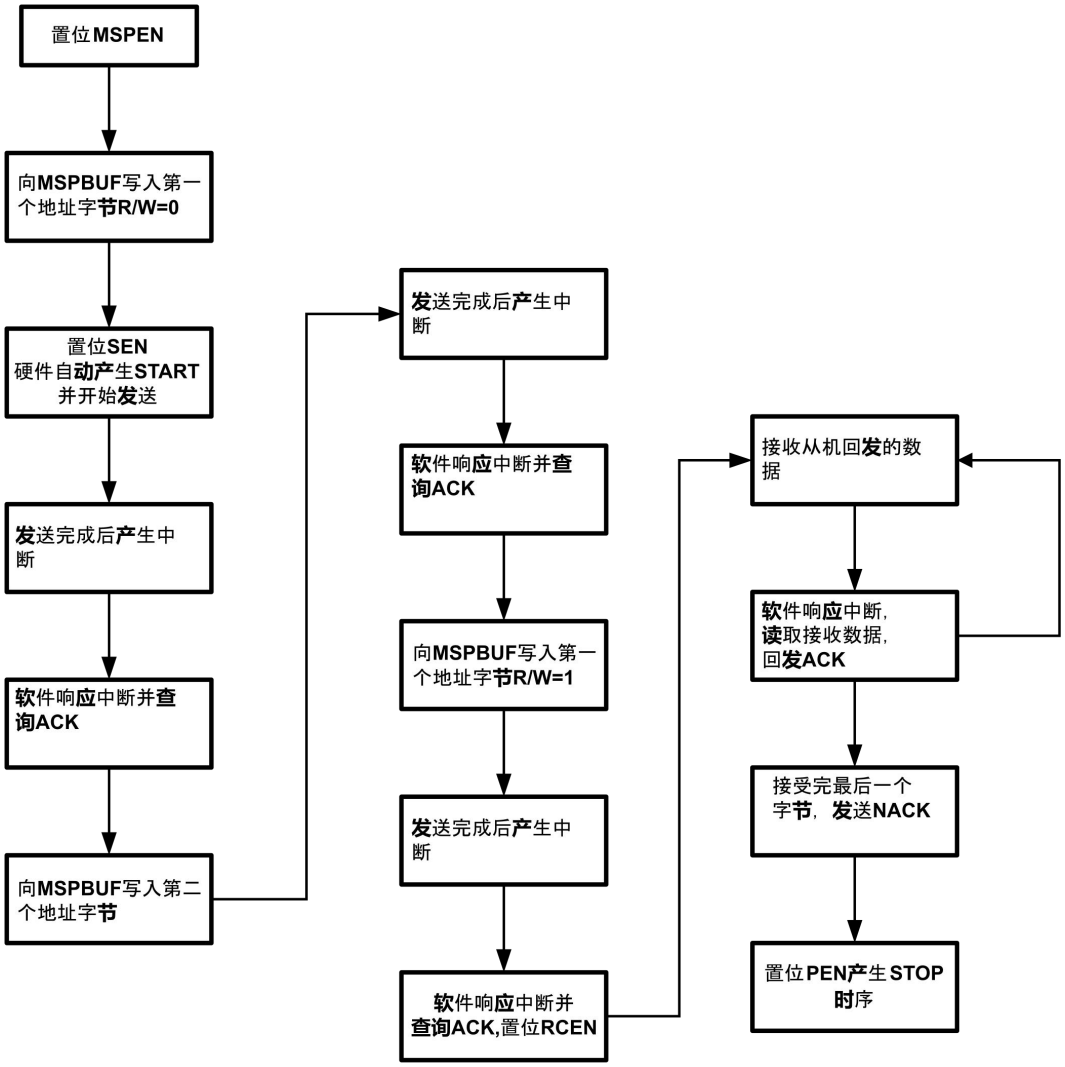


图 15-16 I²C 软件 10 位地址接收数据流图

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。

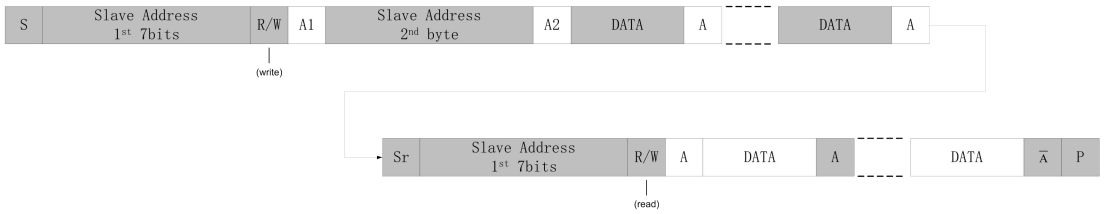


图 15-17 I²C 软件重发数据流图

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。

15.9.3 SCL 延展（Slave Clock Stretching）

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

注意：主机只在每字节收发的第一个 SCL 上升沿处进行 SCL 延展检查。

15.9.4 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。

15.9.5 可编程时序

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

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

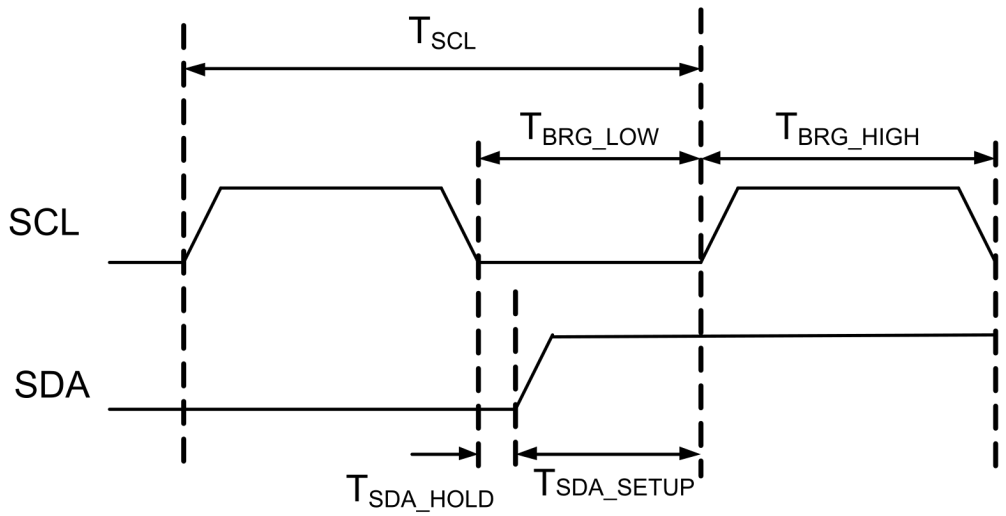


图 15-18 主机时序控制

上图中，T_{SCL} 为通信波特率，各个参数可以由以下公式表达：

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

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

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

MSPBRGH >= 2

MSPBRGL >= 2

MSPBRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

15.10 寄存器

I2C0 基地址 (0x40015000)

I2C1 基地址 (0x40015400)

地址	符号	名称
0x0000	I2C_MSPCFGR	I2C 主机配置寄存器
0x0004	I2C_MSPCR	I2C 主机控制寄存器
0x0008	I2C_MSPIER	I2C 主机中断使能寄存器
0x000C	I2C_MSPISR	I2C 主机中断标志寄存器
0x0010	I2C_MSPSR	I2C 主机状态寄存器
0x0014	I2C_MSPBGR	I2C 主机波特率设置寄存器
0x0018	I2C_MSPBUF	I2C 主机收发缓冲寄存器
0x001C	I2C_MSPTCR	I2C 主机时序控制寄存器
0x0020	I2C_MSPTOR	I2C 主机超时寄存器

15.10.1 I2C 主机配置寄存器(I2C_MSPCFGR)

符号	I2C_MSPCFGR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						TOEN	MSPEN
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	TOEN	RW	SCL 拉低超时使能 (Time Out enable) 1：使能超时功能，超时周期由 TIMEOUT 寄存器定义 0：关闭超时功能
0	MSPEN	RW	I2C 主机模块使能控制位 (Master enable) 1：I2C 主机使能 0：I2C 主机禁止

15.10.2 I2C 主机控制寄存器(I2C_MSPCR)

符号	I2C_MSPCR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				RCEN	PEN	RSEN	SEN
权限	R-0000				RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3	RCEN	RW	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	RW	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	RW	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	RW	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)

15.10.3 I2C 主机中断使能寄存器(I2C_MSPIER)

符号	I2C_MSPIER							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	WCOLE	OVTE	SE	PE	NACKE	TXIE	RXIE
权限	R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:7	RFU	R	RFU：未实现，读为 0
6	WCOLE	RW	WCOL 中断使能寄存器 1：允许写冲突中断 0：禁止写冲突中断
5	OVTE	RW	SCL 超时中断使能寄存器 1：允许超时中断 0：禁止超时中断

位	位名	权限	功能描述
4	SE	RW	START 时序中断使能寄存器 1：允许 START 时序中断 0：禁止 START 时序中断
3	PE	RW	STOP 时序中断使能寄存器 1：允许 STOP 时序中断 0：禁止 STOP 时序中断
2	NACKE	RW	从机无回应中断使能 1：允许从机无回应中断使能 0：禁止从机无回应中断使能
1	TXIE	RW	I2C 主机发送完成中断使能 1：允许发送完成中断 0：禁止发送完成中断
0	RXIE	RW	I2C 主机接收完成中断使能 1：允许接收完成中断 0：禁止接收完成中断

15.10.4 I2C 主机中断标志寄存器(I2C_MSPISR)

符号	I2C_MSPISR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	WCOL	OVT	S	P	ACKSTA	TXIF	RXIF
权限	R-0	W1C-0	W1C-0	R-0	R-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:7	RFU	R	RFU：未实现，读为 0
6	WCOL	W1C	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 SSPBUF，否则发生写冲突；硬件置位， 软件写 1 清零 1 = 发送写冲突 0 = 未发生冲突
5	OVT	W1C	SCL 超时中断标志，仅在 OVTEN 为 1 时工作 1：发生 SCL 超时 0：没有发生 SCL 超时
4	S	R	START 时序发送完成中断标志，硬件置位，软件读

位	位名	权限	功能描述
			取后清零
3	P	R	STOP 时序发送完成中断标志，硬件置位，软件读取后清零
2	ACKSTA	W1C	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	W1C	I2C 主机发送完成中断标志，硬件置位，软件写 1 清零 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	W1C	I2C 主机接收完成中断标志，硬件置位，软件写 1 清零 此标志寄存器在主机回发完 ACK 或 NACK 后置位。

15.10.5 I2C 主机状态寄存器(I2C_MSPSR)

符号	I2C_MSPSR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		BUSY	RW	RFU	BF	RFU	ACKMO
权限	R-00		R-0	R-0	R-0	R-0	R-0	RW-0

位	位名	权限	功能描述
31:6	RFU	R	RFU：未实现，读为 0
5	BUSY	R	I2C 通信状态位 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	R	I2C 传输方向状态位 1：主机从从机读取数据 0：主机向从机写入数据
3	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
2	BF	R	缓冲器满状态位 接收： 1 = 接收完成，SSPBUF 满 0 = 接收未完成，SSPBUF 空 发送： 1 = 正在发送，SSPBUF 满 0 = 发送完成，SSPBUF 空
1	RFU	R	RFU：未实现，读为 0
0	ACKMO	RW	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

15.10.6 I2C 主机波特率设置寄存器(I2C_MSPBGR)

符号	I2C_MSPBGR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							MSPBRGH
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MSPBRGH							
权限	RW-00010011							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							MSPBRGL
权限	R-0000000							RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBRGL							
权限	RW-00010011							

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24:16	MSPBRGH	RW	主机发送的 SCL 时钟高电平宽度，以 I2C 工作时钟计数 注：最小可配置为 0x1
15:9	RFU	R	RFU：未实现，读为 0
8:0	MSPBRGL	RW	主机发送的 SCL 时钟低电平宽度，以 I2C 工作时钟计数 注：最小可配置为 0x2

15.10.7 I2C 主机收发缓冲寄存器(I2C_MSPBUF)

符号	I2C_MSPBUF							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSPBUF							
权限	RW-00000000							

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:0	MSPBUF	RW	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，SSPSR 将数据载入 MSPBUF，同时置位 RXIF。MSPSR 不是直接寄存器，没有物理地址。

15.10.8 I2C 主机时序控制寄存器(I2C_MSPTCR)

符号	I2C_MSPTCR							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							SDAHD
权限	R-0000000							RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SDAHD							
权限	RW-00001010							

位	位名	权限	功能描述

位	位名	权限	功能描述
31:9	RFU	R	RFU：未实现，读为 0
8:0	SDAHD	RW	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 最小可配置为 1，需小于 MSPBRGL

15.10.9 I2C 主机超时寄存器(I2C_MSPTOR)

符号	I2C_MSPTOR							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				TIMEOUT			
权限	R-0000				RW-1111			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TIMEOUT							
权限	RW-11111111							

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:0	TIMEOUT	RW	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，不可配置为 0 $T_{SCL_STRETCHING_TIMEOUT}$=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
- 4 路独立通道（UART1, UART2, UART4, UART5）
- 全双工通信口
- 支持半双工单线通信
- 支持 RX-TX 端口交换
- UART 具有数据接收完成/接收错误中断，并提示错误类型
- 可配置数据长度，支持 6、7、8、9bits
- 可配置的停止位-支持 1 个停止位或 2 个停止位
- 可配置为红外调制输出功能，且载波频率可设置，及载波占空比可设置
- 支持 DMA
- 支持接收超时机制
- 支持发送延迟功能

UART1 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

UART2, UART4, UART5 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

16.2 结构框图

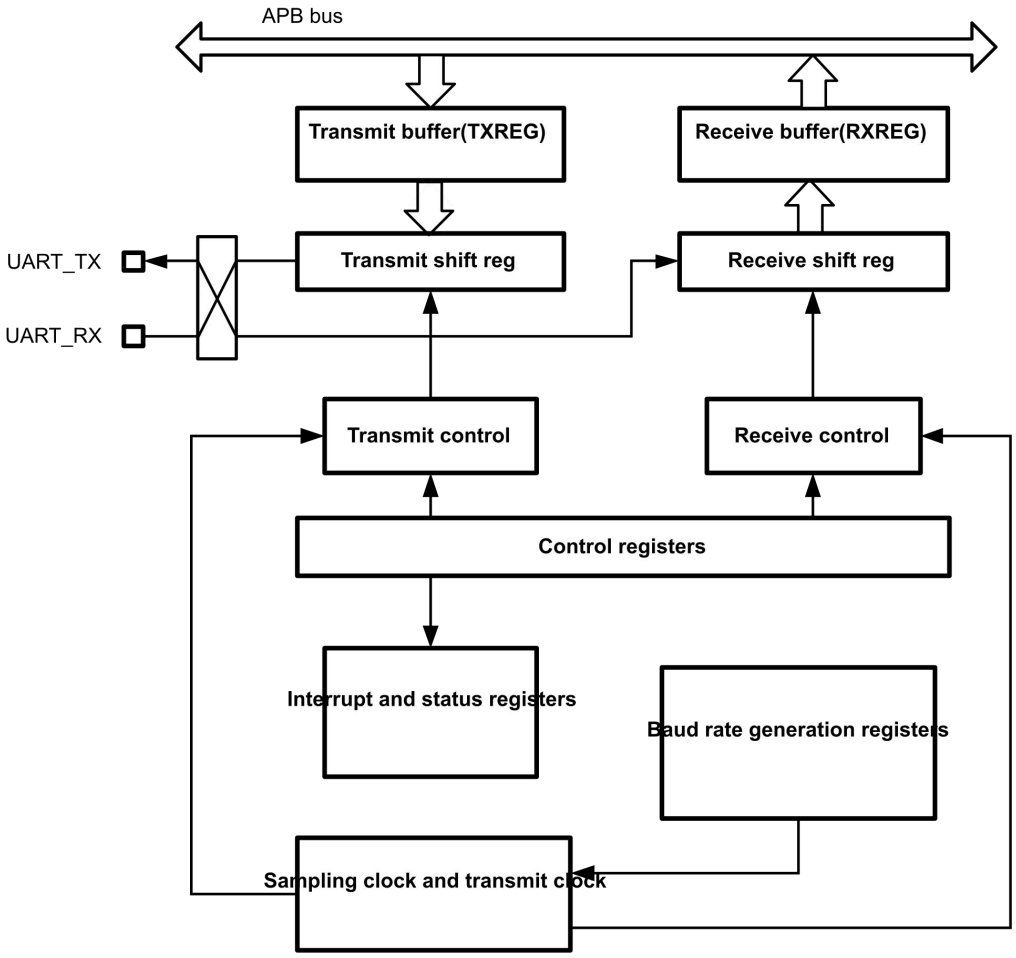


图 16-1 UART 结构框图

16.3 引脚定义

UART 模块使用 2 个引脚与外部器件通信，每个 UART 的收发信号可能被映射到不同的 GPIO 上，具体映射管脚参考引脚和封装。

智能卡模式下，UART_RX 复用为 7816 时钟输出，UART_TX 复用为 7816 数据信号线。

16.4 UART 类型区分

芯片集成了多种不同类型的 UART，其差异如下表所示：

表 16-1 UART 类型列表

UART 特性	UART1	UART2/4/5
DMA 支持	Y	Y
半双工/全双工	Y	Y
红外发射	Y	Y

UART 特性	UART1	UART2/4/5
双时钟域（工作时钟独立于总线）	Y	–
半双工单线通信	Y	Y
休眠唤醒	Y	Y
接收超时	Y	Y
发送延迟	Y	Y
数据长度	6、7、8、9bits	6、7、8、9bits
LIN support	Y	N
智能卡模式	N	Y

16.5 UART 字符描述

16.5.1 UART 模式

UART 支持多种帧格式，由 UARTxCSR.PDSEL 寄存器控制数据段长度为 6/7/8/9，由 UARTxCSR.PARITY 寄存器控制校验位为无校验/奇校验/偶校验，由 STOPCFG 寄存器决定 Stop 位为 1bit 或者 2bit。

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，通信帧长为【起始位+数据位+校验位+停止位】。

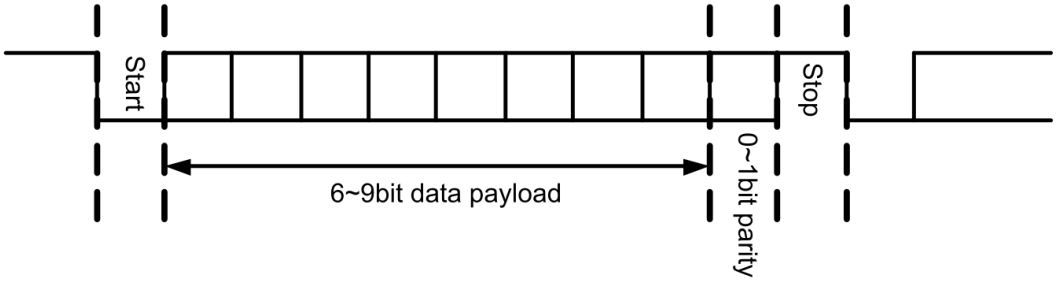


图 16-2 UART 字符描述

16.5.2 智能卡模式

ISO7816-3 协议规定的字符格式如下图。

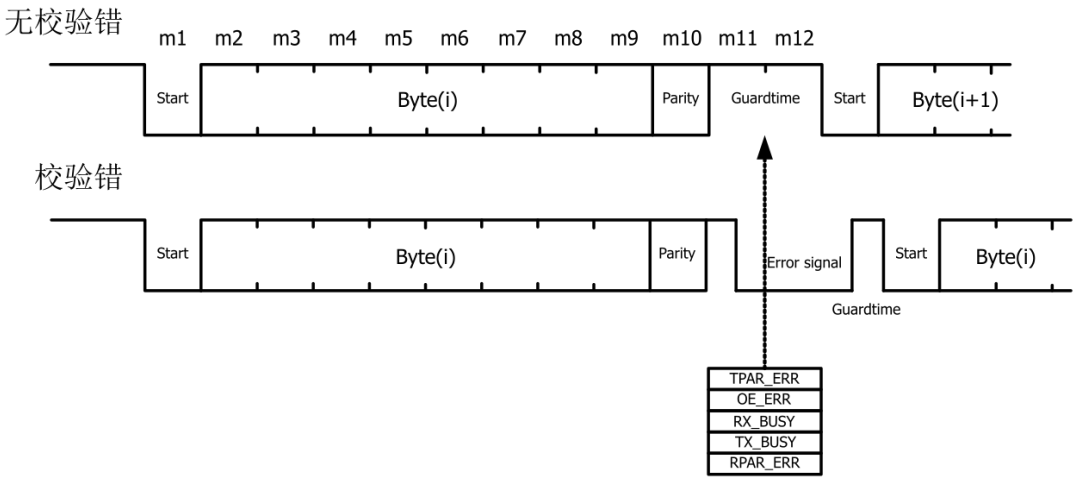


图 16-3 7816 字符描述

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。

16.6 功能描述

16.6.1 时钟结构

支持 LIN 的 UART 为异步 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、XTHF，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。

支持智能卡模式的 UART 为同步 UART,采用单时钟结构，此时 UCLK=PCLK，UART 的数据收发时钟也是来源于 APBCLK 的。

16.6.2 同步策略

UART 支持两种同步策略：起始位同步和重同步。

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

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

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

16.6.3 位接收采样

UART 对接收数据进行波特率的 8 倍或 16 倍过采样，并在每个 bit 的中间位置进行三中取二的多数判决，以提高对信号噪声的抑制能力。

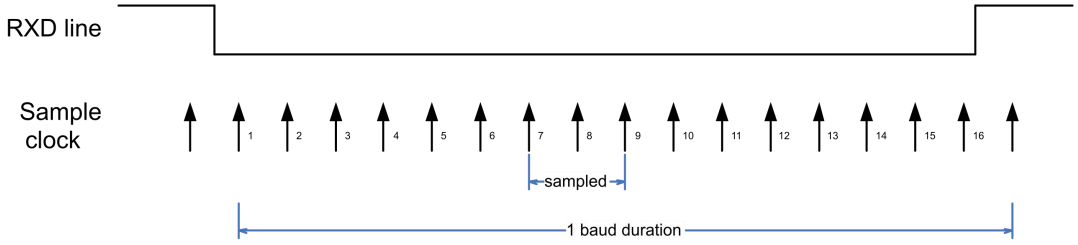


图 16-4 位接收 16 倍采样

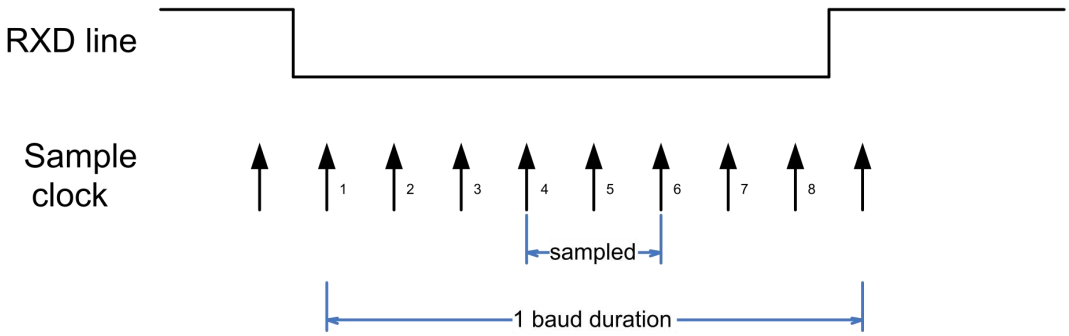


图 16-5 位接收 8 倍采样

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。

16.6.4 数据发送

在发送模式下，UART 的串行数据发送电路主要包括一个发送移位寄存器(TSR)，TSR 功能是将数据逐个移位送出。待发数据必须先写到发送缓冲区中。当软件置位 TXEN 寄存器后，如果发送缓冲区非空，UART 将缓冲区数据载入 TSR 并开始移位输出。

注：由于寄存器操作时钟和波特率时钟是异步关系，当发送开始时，需要等待波特率时钟到来，因此从 TXEN 置位到 UART 开始发送 Start 位之间，有最大 1 个 baud 的延迟。

TXBE 和 TXSE 是发送中断标志位，分别表示发送缓冲区空和 TSR 空，软件可以选择在合适的时间点产生发送完成中断。

一般情况下，一开始 TSR 寄存器是空的，数据的发送需先设定波特率 SPBRG，使能发送模块(设定 TXEN 为 1)，然后写入 TXBUF 寄存器开始发送。也可以在设定好波特率 SPBRG 后，先写入 TXBUF 寄存器，然后再设定 TXEN 使能发送模块来开始数据发送。如果在数据发送过程中将发送模块使能位 TXEN 清 0，那么数据发送工作就会被中断，发送模块也会被复位。

下图为 UART 异步发送的例子。这个示例中软件首先向 TXBUF 写入数据，然后通过置位 TXEN 启动发送。

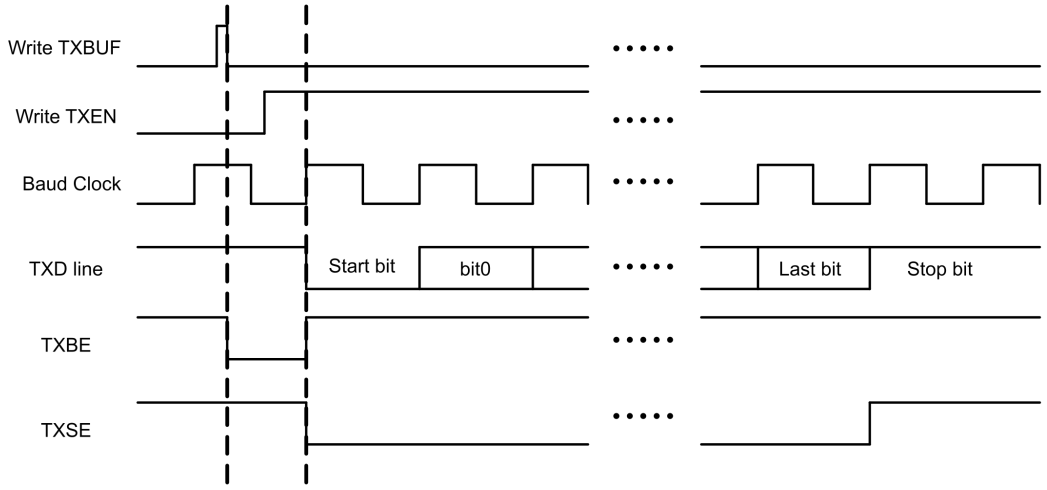


图 16-6 UART 异步发送波形 1

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。

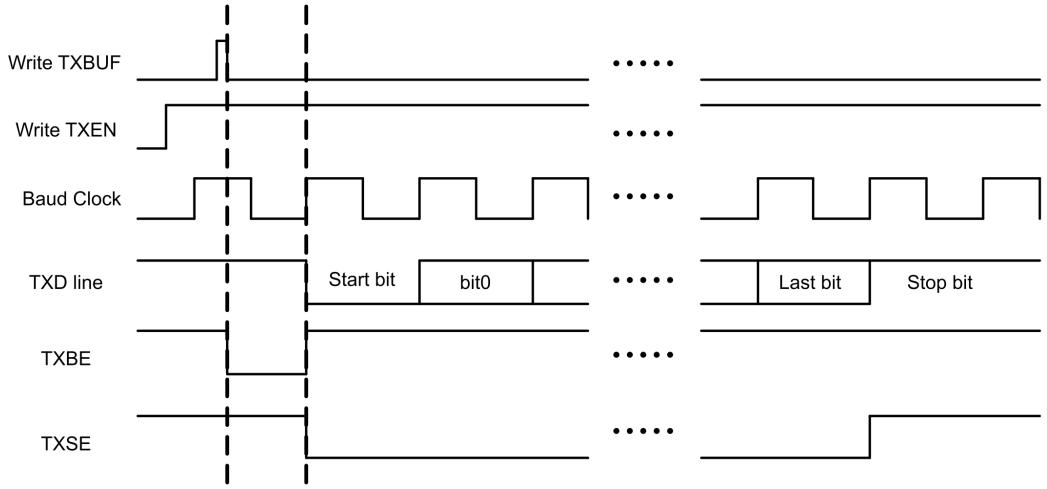


图 16-7 UART 异步发送波形 2

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。

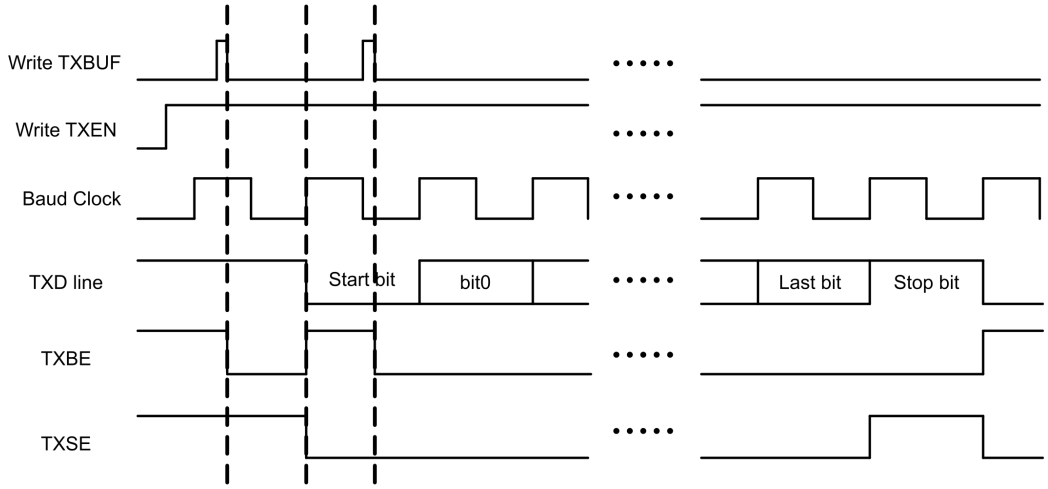


图 16-8 UART 异步发送波形 3

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。

16.6.5 数据接收

UART 的串行数据接收电路主要包括一个接收移位寄存器(RSR)。当接收到停止位后，RSR 就把接收到的数据送入接收缓冲区(RXBUFFER)，传送完成后，在每次接收数据送入接收缓冲区后将

中断标志 RXBF 置 1。当接收缓冲区已满时，RSR 接收到一帧数据后仍会将其写入接收缓冲区，即覆盖缓冲区中原有数据，并且再次置位 RXBF，同时发生接收溢出错误，OERR 被置 1；软件写 1 或者读取 RXBUF 都可以清除 OERR 标志。

接收过程中，如果没有检测到正确的停止位，则发生帧格式错，FERR 被置 1；如果发生奇偶校验错，标志位 PERR 被置 1。

推荐的异步接收操作如下：

- 选择合适的波特率，初始化 SPBRG
- 若需要中断，置位 RXBF_IE
- 设置数据接收的格式：设置 PDSEL 寄存器，决定发送数据长度；设置 PARITY 寄存器选择是否发送校验位以及校验类型，设置 STOPSEL 寄存器决定发送 1 位还是 2 位停止位
- 使能接收模块：置位 RXEN
- 在一帧接收完毕时，RXBF 位会置 1，如果 RXBF_IE 位为 1，将会产生中断
- 读取 PERR、FERR、OERR 寄存器，判断是否有数据错误或者溢出
- 读取 RXBUF 寄存器中的接收数据

智能卡模式

智能卡模式下，UART 完成数据和校验位接收后，自动判断校验位是否正确。如果校验位正确，则判决停止位，并在正确的停止位后置位接收完成中断，或在错误的停止位置位帧格式错误中断。如果校验位错误，则自动回发错误信号并等待对方重发；如果重发次数达到了寄存器设定的上限，仍没有接收到正确的帧，则停止回发错误信号，并置位接收错误中断。

16.6.6 RS232 硬件流控

UART 支持 RS232 硬件流控，可以通过 nCTS 输入和 nRTS 输出，实现两个设备之间的数据流控制。

注意：如果没有 CTS 和 RTS 端口，则对应 UART 无流控功能。

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。

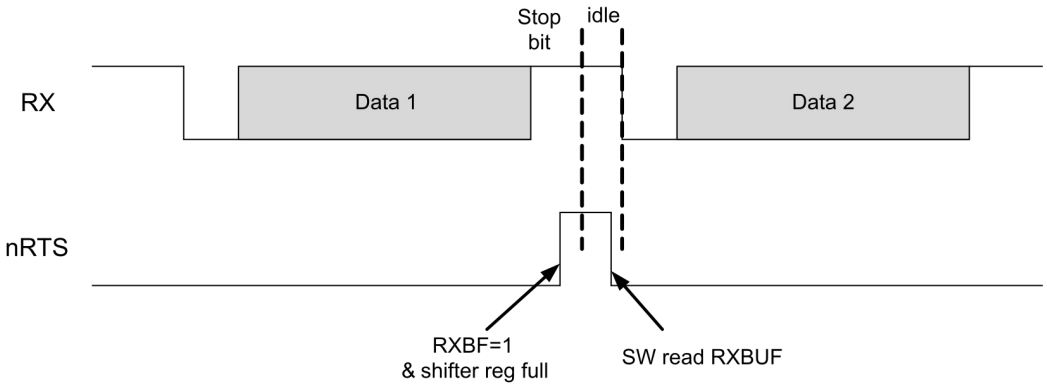


图 16-9 nRTS 流控

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。

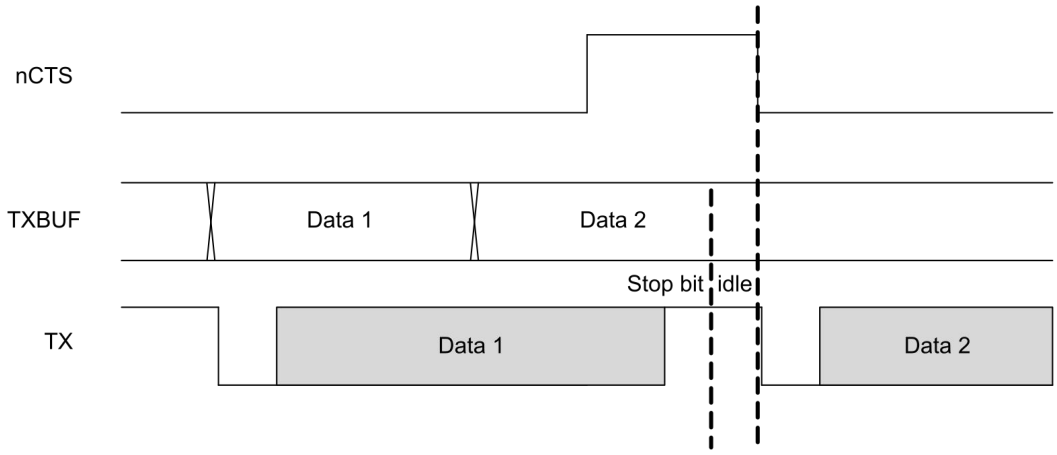


图 16-10 nCTS 流控

16.6.7 低功耗休眠唤醒

所有 UART 都支持 RXD 下降沿触发的芯片休眠唤醒。当置位了 NEWUP 寄存器之后，RXD 输入上的下降沿事件（低电平持续时间>100ns）将会使芯片从休眠模式下唤醒，借助这个功能，可以实现 UART 在休眠模式下唤醒后接收数据。

软件配置方法如下：

- 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
- 配置 UART 寄存器，使能 NEWUP
- 软件设置芯片进入休眠，等待 UART 接收事件

异步 UART 可以进行从机总线唤醒信号检测。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 ### 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

16.6.8 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：

表 16-2 DMA 发送中断

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

16.6.9 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

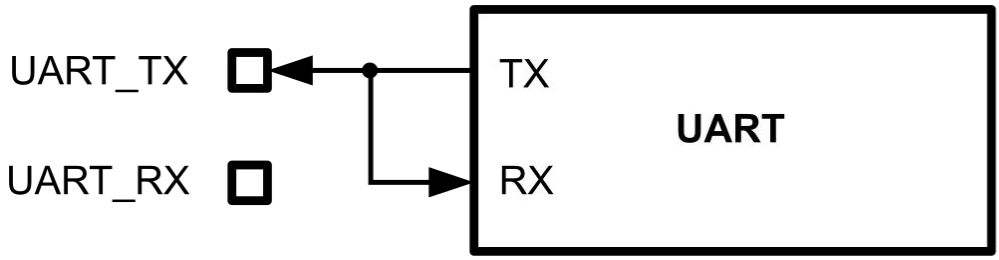


图 16-11 UART 半双工单线通信连接示意图

16.7 波特率发生

16.7.1 波特率计算

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，对于同步 UART，F_{CLK} 就是 APBCLK；对于异步 UART，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：

表 16-3 常用时钟频率下波特率计算

Baud	FCLK=16MHz			FCLK=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	FCLK=24MHz			FCLK=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

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。

$$F_{7816} = F_{APBCLK}/(CLKDIV + 1)$$

7816 通信波特率使用 7816 接口时钟产生：

$$Baud = F_{7816}/(SPBRG + 1)$$

智能卡模式下 SPBRG 最小可用值是 0x1，禁止配置 0x0

16.7.2 波特率自适应

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

16.8 红外调制

TZBRG 寄存器保存一个 11 位的分频系数 X，其值为 0~2047 之间的任一整数。所有 UART 共用一个红外调制频率发生器。

红外调制频率计算公式：

$$FIR = F_{APBCLK}/(TZBRG + 1)$$

红外调制的方式为：发送数据 0 时调制红外频率，发送数据 1 时不调制。为满足 PNP 和 NPN 两种红外驱动管的需求，寄存器 IRFLAG 位控制红外调制输出的极性。IRFLAG=0 时为正极性输出，适合 PNP 管驱动；IRFLAG=1 时为负极性输出，适合 NPN 管驱动。

TH 寄存器用于配置红外调制占空比

占空比：Y = (TZBRG[10:4] * TH) /(TZBRG+1)

当 TH=4’b0000 时，占空比为 Y = (TZBRG[10:1]+1)/(X+1)；

当 TZBRG[10:4]=7’h00 时，占空比为 Y = TH/(TZBRG[3:0]+ 1)；若此时 TH>TZBRG [3:0]，则红外调制时钟 IRCLK 为固定高电平。

当红外调制极性反向时（IRFLAG=1），占空比也为 1-Y

红外调制波形见下图：

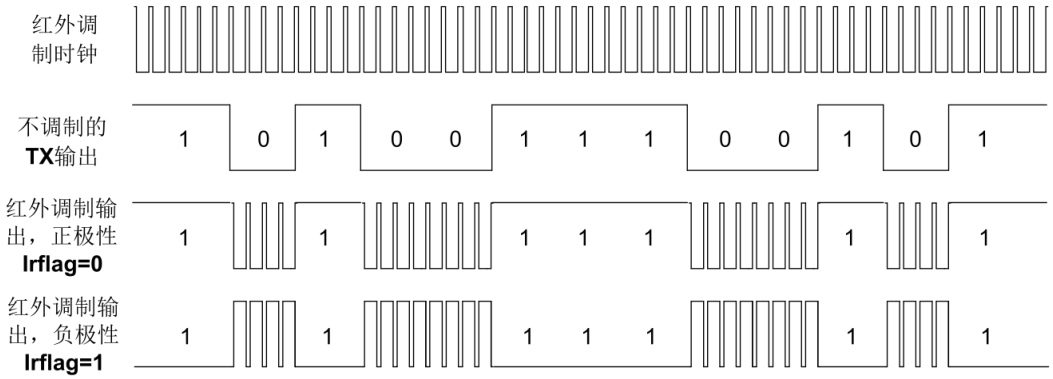


图 16-12 红外调制波形

无论有效电平是 0 还是 1，占空比定义为高电平长度/周期。

16.9 接收超时

针对 MODBUS 等时间敏感型应用，设计了接超时机制。当使能 RXTOEN 寄存器后，超时计数器以波特率时钟计数，当每次收到一个完整的数据帧，将清零超时计数器并重新开始计数。超时溢出的上限值可以由软件配置，最大 255 波特。

16.10 发送延迟

通过 TXDLY_LEN 寄存器，可以控制两个数据帧发送之间的间隔时间，单位是波特。发送延迟是从上一帧最后一个 STOP 位结束，到下一帧起始位之间的间隔。

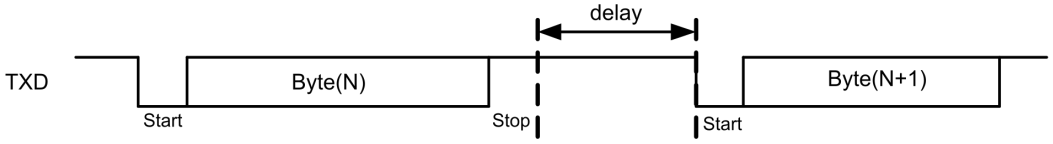


图 16-13 UART 发送延迟

16.11 LIN 总线通信功能

16.11.1 概述

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

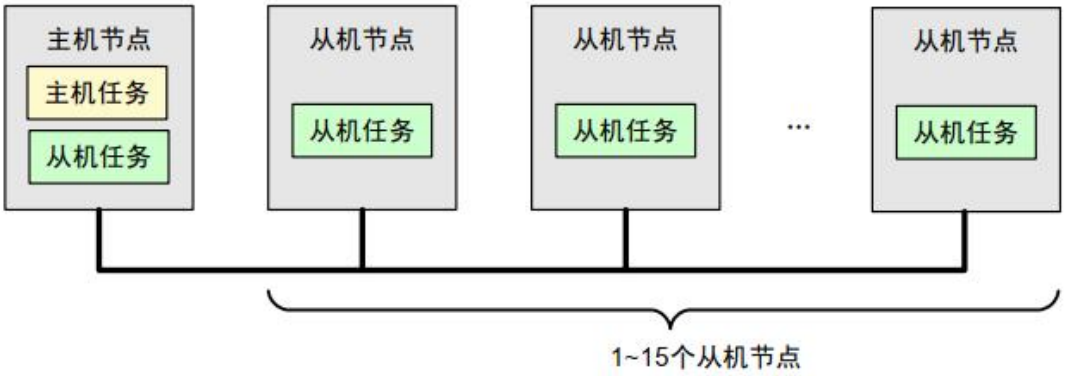


图 16-14 LIN 总线拓扑

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。

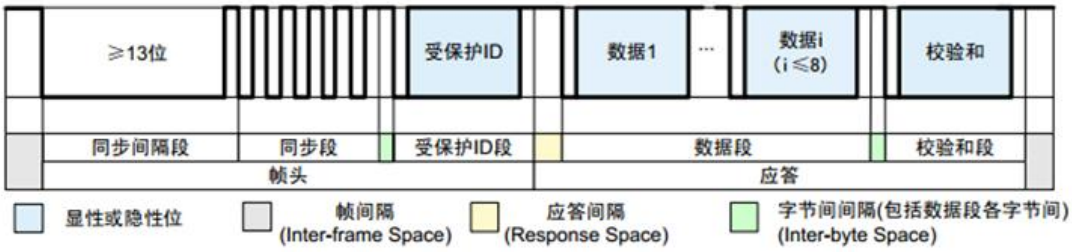


图 16-15 LIN 总线帧结构

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)。

16.11.1.1 Break Field

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



图 16-16 LIN break field

16.11.1.2 Sync Field

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

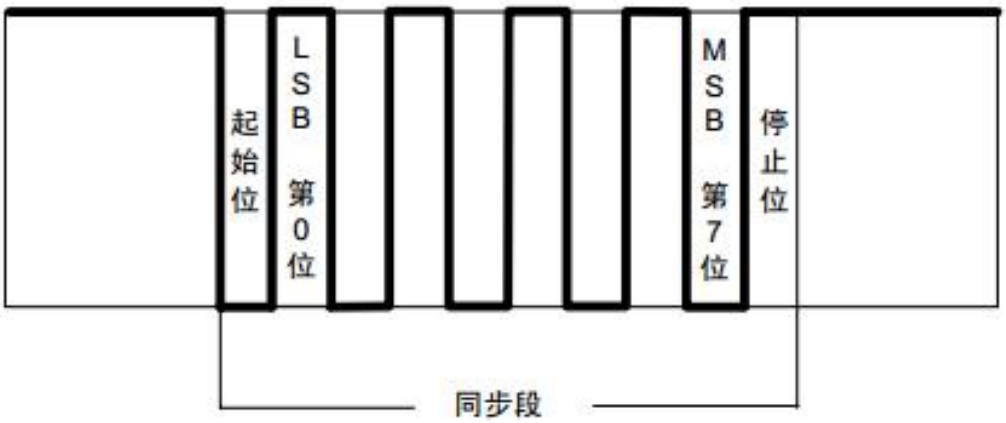


图 16-17 LIN sync field

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= （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。

16.11.1.3 PID Field 和 Data Field

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

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

$ P0=ID0 ID1 ID2 ID4 $

$ P1= $

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 16-18 LIN 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。

16.11.1.4 Checksum Field

校验段对帧中所传输的内容进行校验，校验对象包含：

- 经典校验（LIN 1.x）：所有数据段
- 加强校验（LIN 2.x）：所有数据段以及 PID 段

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

校验功能由软件实现。

16.11.2 LIN 主机操作

在主机模式下，LIN 模块能够向总线发送唤醒信号、间隔段、同步段、数据段。

16.11.2.1 唤醒发送

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

16.11.2.2 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。

16.11.3 LIN 从机操作

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

16.11.3.1 唤醒检测

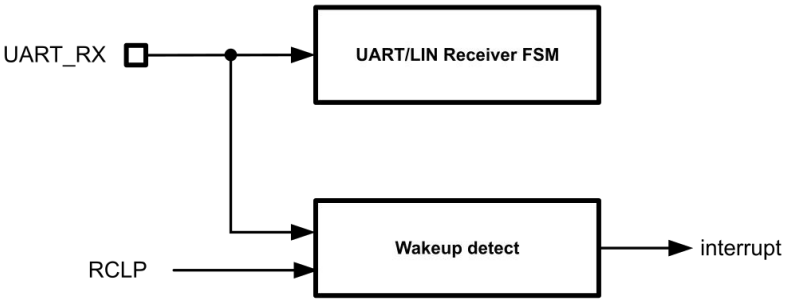
LIN 总线处于休眠状态时，主/从机节点都可以向总线上发送唤醒信号，唤醒信号持续 250us～5ms。其余节点（除发送唤醒信号以外的节点）以大于 150us 为阈值判定唤醒信号。每个从机节点必须在唤醒信号显性脉冲的结束处算起 100ms 以内准备接收来自主机的命令(帧头)；主机节点也必须被唤醒，100ms 之内主机节点发送帧头开始通信。



图 16-19 LIN 总线唤醒

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 变为低电平后计数器开始计数，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。

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。



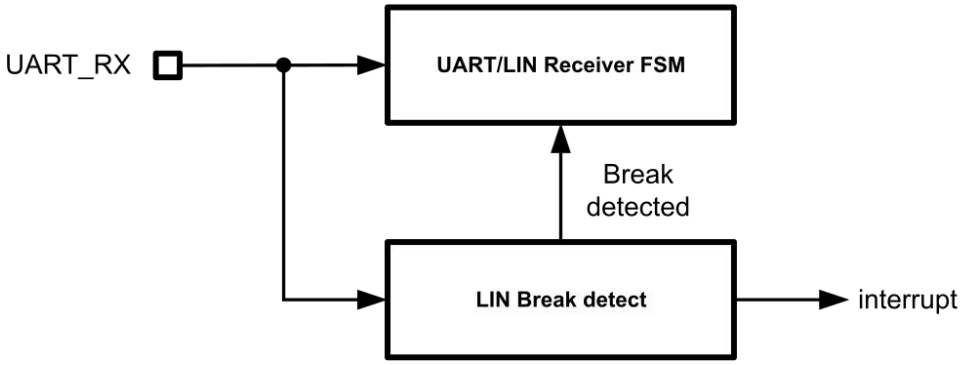
注意，RXD 经过 RCLP 采样会导致 1~2 个 RCLP 时钟的延迟，假如设置 WKUP_LEN=5，实际 RXD 低电平长度需达到 6~7 个 RCLP 周期，才会触发唤醒标志。

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。

16.11.3.2 Break 检测

Break 检测模块独立于 LIN 从机接收，因此在 LIN 通信的任何阶段，都能够识别主机发送的 Break，并将接收状态机置于初始状态，开始新的一帧接收。在数据接收过程中收到 Break 会触发帧格式错误（stop=0，停止位错误）。



从机接收到 Break 段，会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。

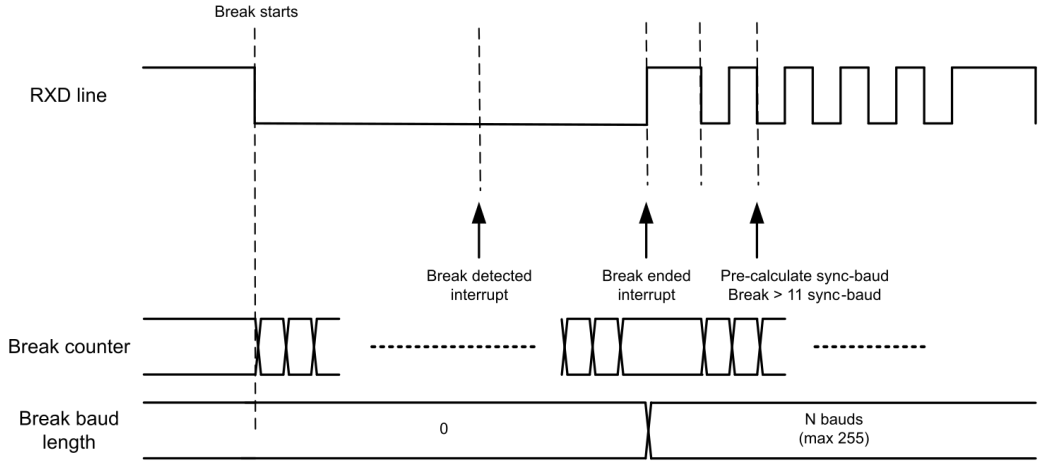


图 16-20 Normal Break

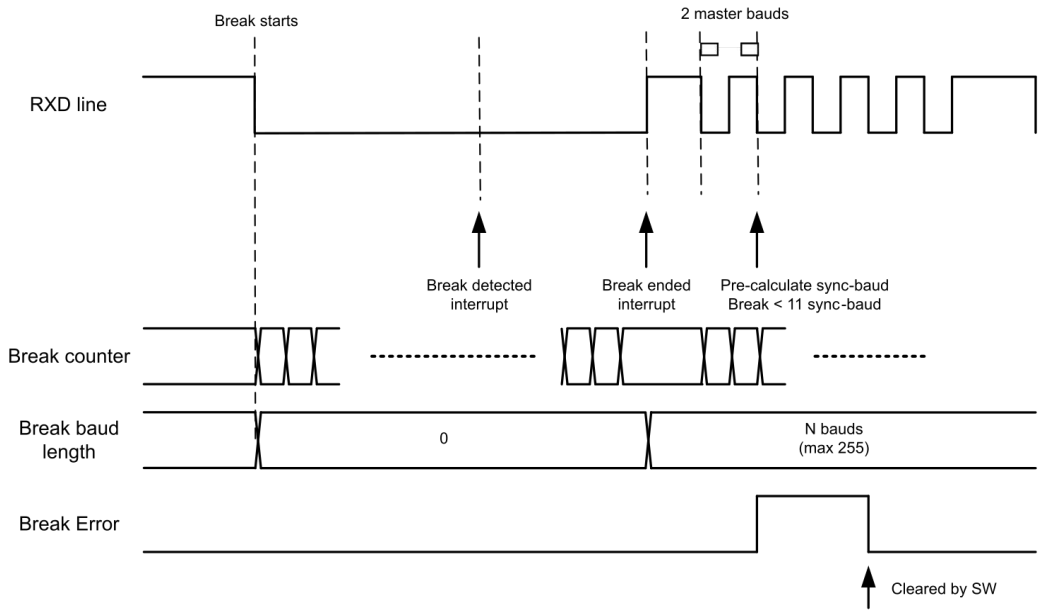


图 16-21 Break Error

16.11.3.3 超时检测

使能 FTOEN 寄存器后，LIN 模块开始计数，计数周期为波特率周期，计数到 FTO 后置起置位超时中断、停止当前 frame 接收，并回到 IDLE 状态。软件如果在帧结束时没有检测到帧超时标志，可通过关闭 FTOEN 把超时计数器清零。

16.11.3.4 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。

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*

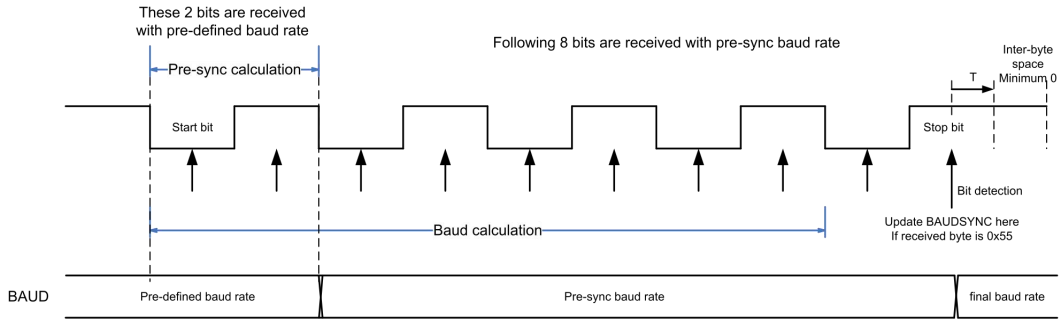


图 16-22 LIN 波特率自适应过程

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）

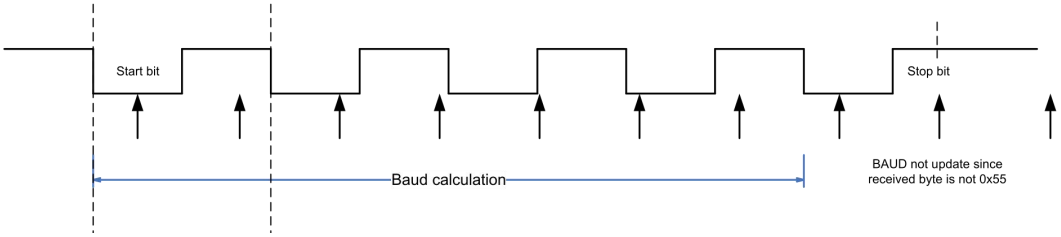


图 16-23 波特率误差较大，无 pre-sync

16.11.3.5 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。

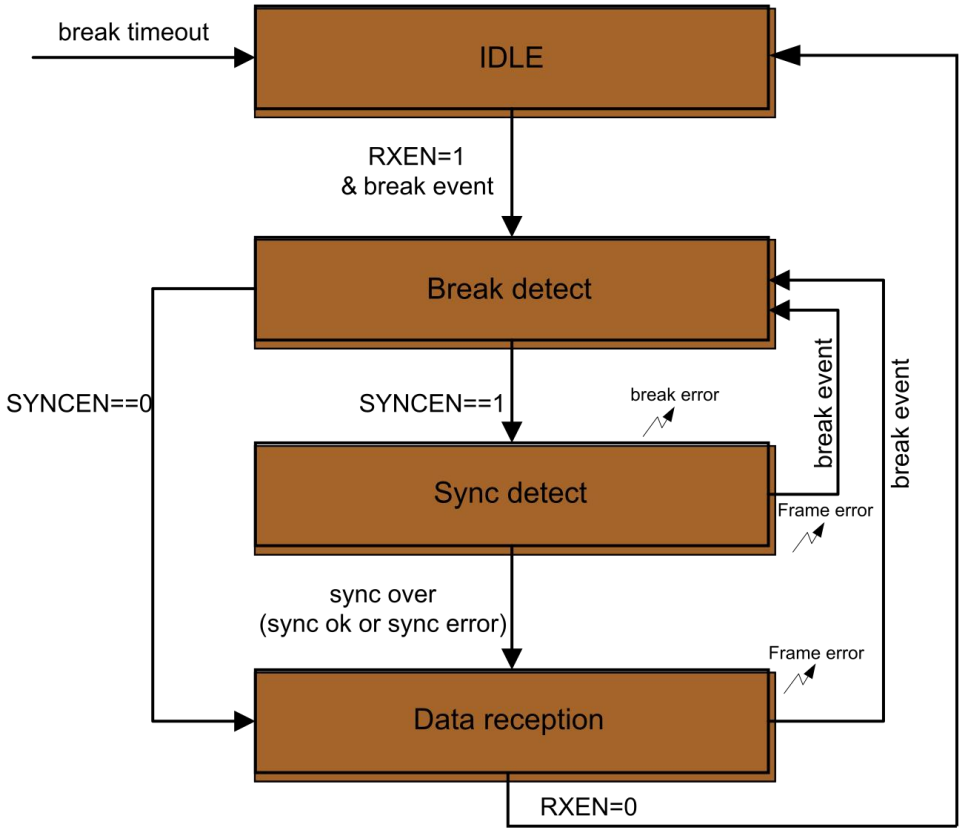


图 16-24 LIN 从机接收流程

16.11.3.6 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。

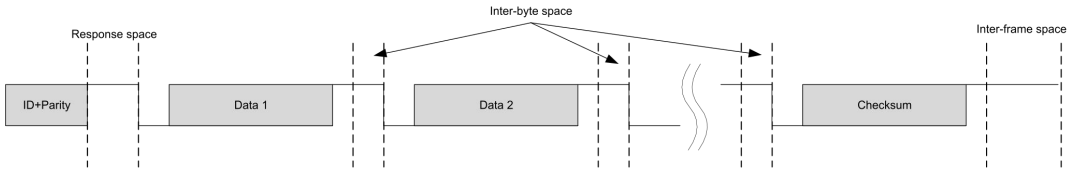


图 16-25 字节间隔和应答间隔

校验和由软件计算，校验和的发送与数据字节相同。

16.11.3.7 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。

16.11.4 LIN 接收采样与噪声检测

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

即使 NF 标志置位，不影响 3 中取 2 的有效数据判决逻辑。

16.11.5 LIN 错误状态

16.11.5.1 发送数据冲突检测（Bit Error）

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

数据冲突检测针对主机/从机发送的 break 段、sync 段、数据段和校验和段。

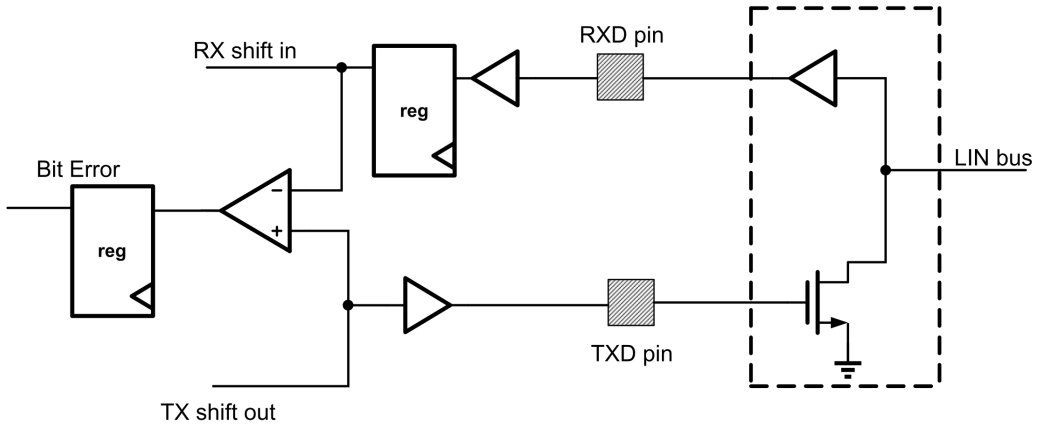


图 16-26 LIN bit error 冲突检测

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：

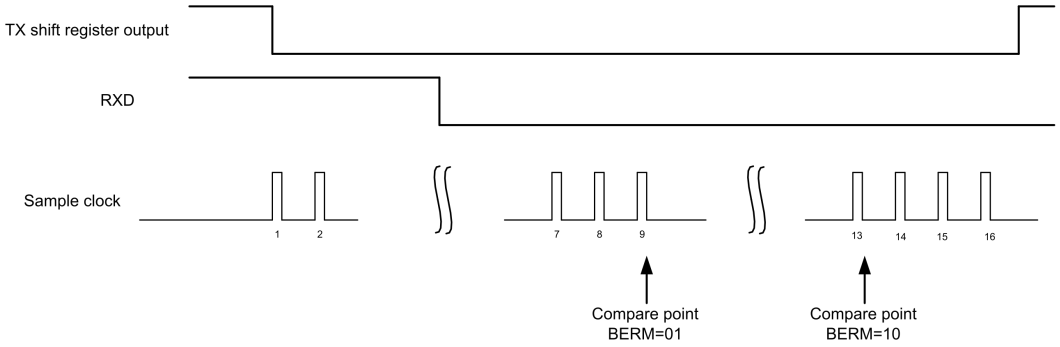


图 16-27 LIN bit error 监测点

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

16.11.5.2 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。

16.11.5.3 break 错误

检测到 break 段错误时 LINBER 标志会置位。LINBER 置位条件是：

[(SYNC 段前两个下降沿长度)/2 * 11] > [break 长度]

16.11.5.4 sync 错误

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

SYNC 错误有两种可能性：

1. 主机发送的 SYNC 段波特率偏差过大
2. 主机没有发送正确的 SYNC 段

16.12 寄存器

UART1 基地址 (0x40012000)

地址	符号	名称
0x0000	UARTL_CSR	UART 控制状态寄存器
0x0004	UARTL_IER	UART 中断使能寄存器
0x0008	UARTL_ISR	UART 中断标志寄存器
0x000C	UARTL_TODR	UART 超时和延迟寄存器
0x0010	UARTL_RXBUF	UART 接收缓冲寄存器
0x0014	UARTL_TXBUF	UART 发送缓冲寄存器
0x0018	UARTL_BGR	UART 波特率产生寄存器
0x0020	UARTL_MCR	UART_LIN 模式寄存器
0x0024	UARTL_LINCR	UART_LIN 控制寄存器
0x0028	UARTL_LINBSR	UART_LIN 波特率同步寄存器
0x002C	UARTL_LINFTR	UART_LIN 帧超时寄存器
0x0030	UARTL_LINTTR	UART_LIN 发送时序寄存器
0x0034	UARTL_LINPSR	UART_LIN Pre-sync 波特率寄存器
0x0038	UARTL_LINBKR	UART_LIN break 段长度寄存器

UART2 基地址 (0x40017400)

UART4 基地址 (0x4001A000)

UART5 基地址 (0x4001A400)

地址	符号	名称

地址	符号	名称
0x0000	USART_CSR	USART 控制状态寄存器
0x0004	USART_IER	USART 中断使能寄存器
0x0008	USART_ISR	USART 中断标志寄存器
0x000C	USART_TODR	USART 超时和延迟寄存器
0x0010	USART_RXBUF	USART 接收缓冲寄存器
0x0014	USART_TXBUF	USART 发送缓冲寄存器
0x0018	USART_BGR	USART 波特率产生寄存器
0x0020	USART_MCR	USART 模式寄存器
0x0040	USART_SCISR	USART 智能卡状态标志寄存器
0x0044	USART_SCIER	USART 智能卡中断使能寄存器
0x0048	USART_FFCR	USART 智能卡帧格式控制寄存器
0x004C	USART_EGTR	USART 智能卡额外保护时间寄存器
0x0050	USART_CODR	USART 智能卡输出时钟分频寄存器

UARTIR 基地址 (0x40017C00)

地址	符号	名称
0x0000	UART_IRCR	红外调制配置寄存器

16.12.1 UART 控制状态寄存器(UARTL_CSR)

符号	UARTL_CSR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							BUSY
权限	R-0000000							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						TXIREN	RXTOEN
权限	R-000000						RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RESYNCEN	RFU	OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
权限	RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
权限	RW-01		RW-00		RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:25	RFU	R	未实现：读为 0
24	BUSY	R	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23:18	RFU	R	未实现：读为 0
17	TXIREN	RW	发送红外调制使胜 1：使能红外调制发逿

位	位名	权限	功能描述
			0：关闭红外调剂
16	RXTOEN	RW	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	RW	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	RFU	R	未实现：读为 0
13	OVSM	RW	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	RW	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	RW	UART RX 下降沿唤醒功能使能寄存器(Negedge Wakeupenable) 1：使能 RX 下降沿唤醒 0：禁止 RX 下降沿唤醒
10	DMATXIFCFG	RW	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	RW	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	STOPCFG	RW	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	RW	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	RW	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

位	位名	权限	功能描述
			10：奇校验 11：RFU
3	RXPOL	RW	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	TXPOL	RW	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	RXEN	RW	接收使能，1 有效 (Receive Enable)
0	TXEN	RW	发送使能，1 有效 (Transmit Enable)

16.12.2 UART 中断使能寄存器(UARTL_IER)

符号	UARTL_IER							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOER_IE	SER_IE	RFU	BER_IE	NF_IE	LINSYNC_IE	LINB_IE	LINWK_IE
权限	RW-0	RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LINWKTF_IE	LINBEF_IE	LINBER_IE	RFU		PERR_IE	FERR_IE	OERR_IE
权限	RW-0	RW-0	RW-0	R-00		RW-0	RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				RXTO_IE	RXERR_IE	RFU	RXBF_IE
权限	R-0000				RW-0	RW-0	R-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWUP_IE	RFU					TXBE_IE	TXSE_IE
权限	RW-0	R-00000					RW-0	RW-0

位	位名	权限	功能描述
31	TOER_IE	RW	LIN timeouut 中断使能，1 有效
30	SER_IE	RW	lin sync error 中断使能，1 有效
29	RFU	R	未实现：读为 0
28	BER_IE	RW	lin 发送冲突中断使能，1 有效
27	NF_IE	RW	lin 噪声监测中断使能，1 有效
26	LINSYNC_IE	RW	LIN 同步段中断使能，1 有效
25	LINB_IE	RW	LIN Break 段中断使能，1 有效
24	LINWK_IE	RW	从机模式 LIN 唤醒检测标志中断使能，1 有效
23	LINWKTF_IE	RW	主机模式 LIN 唤醒发送完成标志中断使能，1 有效
22	LINBEF_IE	RW	从机模式 LIN Break 段结束标志中断使能，1 有效
21	LINBER_IE	RW	从机模式 LIN break error 中断使能，1 有效
20:19	RFU	R	未实现：读为 0
18	PERR_IE	RW	奇偶校验错误中断使能，1 有效

位	位名	权限	功能描述
17	FERR_IE	RW	帧格式错误中断使能，1 有效
16	OERR_IE	RW	接收缓存溢出错误使能，1 有效
15:12	RFU	R	未实现：读为 0
11	RXTO_IE	RW	接收超时中断使能，1 有效
10	RXERR_IE	RW	接收错误中断使能，1 有效
9	RFU	R	未实现：读为 0
8	RXBF_IE	RW	接收缓存满中断使能，1 有效
7	NEWUP_IE	RW	RX 下降沿异步检测中断使能，1 有效
6:2	RFU	R	未实现：读为 0
1	TXBE_IE	RW	发送缓存空中断使能，1 有效
0	TXSE_IE	RW	发送缓存空且发送移位寄存器空中断使能，1 有效

16.12.3 UART 中断标志寄存器(UARTL_ISR)

符号	UARTL_ISR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TOER	SER	RFU	BERIF	NF	LINSYNCF	LINBF	LINWKDF
权限	W1C-0	W1C-0	R-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LINWKTF	LINBEF	LINBER	RFU		PERR	FERR	OERR
权限	W1C-0	W1C-0	W1C-0	R-00		W1C-0	W1C-0	W1C-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	RFU			RXTO	RFU		RXBF
权限	R-0	R-000			W1C-0	R-00		W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWKF	RFU				TX_OERR	TXBE	TXSE
权限	W1C-0	R-0000				W1C-0	R-1	W1C-0

位	位名	权限	功能描述
31	TOER	W1C	LIN timeout error 标志，硬件置位，软件写 1 清零
30	SER	W1C	LIN sync error 标志，硬件置位，软件写 1 清零
29	RFU	R	未实现：读为 0
28	BERIF	W1C	LIN 发送冲突标志，硬件置位，软件写 1 清零
27	NF	W1C	LIN 总线噪声检测标志，硬件置位，软件写 1 清零 0
26	LINSYNCF	W1C	LIN 同步段接收完成中断标志，硬件置位，软件写 1 清零
25	LINBF	W1C	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

位	位名	权限	功能描述
24	LINWKDF	W1C	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	W1C	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	W1C	LIN 从机模式 break 段结束中断标志，硬件置位，软件写 1 清零
21	LINBER	W1C	LIN 从机模式：LIN Break error 标志，硬件置位，软件写 1 清零
20:19	RFU	R	未实现：读为 0
18	PERR	W1C	奇偶校验错误中断标志，硬件置位，软件写 1 清零
17	FERR	W1C	帧格式错误中断标志，硬件置位，软件写 1 清零
16	OERR	W1C	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。
15	RFU	R	未实现：读为 0
14:12	RFU	R	未实现：读为 0
11	RXTO	W1C	接收超时中断标志，硬件置位，软件写 1 清零
10:9	RFU	R	未实现：读为 0
8	RXBF	W1C	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
7	NEWKF	W1C	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零
6:3	RFU	R	未实现：读为 0
2	TX_OERR	W1C	发送缓存不为空时,写入新的 TXBUF 硬件置位,软件写 1 清零
1	TXBE	R	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
0	TXSE	W1C	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

16.12.4 UART 超时和延迟寄存器(UARTL_TODR)

符号	UARTL_TODR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	UARTL_TODR							
位名	TXDLY_LEN							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
权限	RW-11111111							

位	位名	权限	功能描述
31:16	RFU	R	未实现：读为 0
15:8	TXDLY_LEN	RW	发送延时长度，单位长度为波特率周期。
7:0	RXTO_LEN	RW	接收超时长度，单位长度为波特率周期。

16.12.5 UART 接收缓冲寄存器(UARTL_RXBUF)

符号	UARTL_RXBUF							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							RXBUF
权限	R-0000000							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
权限	R-00000000							

位	位名	权限	功能描述
31:9	RFU	R	未实现：读为 0
8:0	RXBUF	R	接收数据缓冲寄存器数据 7 位收发时，接收的 7bits 数据存入 RXBUF[6:0]

16.12.6 UART 发送缓冲寄存器(UARTL_TXBUF)

符号	UARTL_TXBUF							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	UARTL_TXBUF							
位名	RFU							TXBUF
权限	R-0000000							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
权限	W-00000000							

位	位名	权限	功能描述
31:9	RFU	R	未实现：读为 0
8:0	TXBUF	W	发送数据缓冲寄存器数据 7 位收发时，发送的 7bits 数据写入 TXBUF[6:0]

16.12.7 UART 波特率产生寄存器(UARTL_BGR)

符号	UARTL_BGR							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG							
权限	RW-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG							
权限	RW-01000001							

位	位名	权限	功能描述
31:16	RFU	R	未实现：读为 0
15:0	SPBRG	RW	波特率产生器寄存器值。 注：当 LIN 模式下且 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：即使 SYNCEN=1，软件仍可以改写 SPBRG

16.12.8 UART_LIN 模式寄存器(UARTL_MCR)

符号	UARTL_MCR							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	UARTL_MCR							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					HDSEL	LINMD	MODE
权限	R-00000					RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	HDSEL	RW	UART 半双工单线模式选择 0：双线 UART 1：单线半双工 UART 注意：此寄存器仅在 MODE=0 时有效
1	LINMD	RW	LIN 模式选择寄存器 0：LIN 主机 1：LIN 从机
0	MODE	RW	模块控制寄存器 0：标准 UART 模式 1：LIN 模式

16.12.9 UART_LIN 控制寄存器(UARTL_LINCR)

符号	UARTL_LINCR							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							BERS
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	WKUP_LEN							
权限	RW-00001000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SYNCEN	FTOEN	BERM		NDET_EN	WKDET_EN	TX_BF	TX_WKUP
权限	RW-1	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:17	RFU	R	未实现：读为 0
16	BERS	RW	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

位	位名	权限	功能描述
			1：检测到发送冲突时取消后续发送，直到 BERIF 清零
15:8	WKUP_LEN	RW	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（即超过此阈值后认为唤醒信号有效） 单位是 LINCLK，即 lsclk；复位值为 8，表示 9 个时钟周期，大约 279us；最大 256 个时钟周期，大约 7.6ms
7	SYNCEN	RW	lin 从机 sync 段接收使能
6	FTOEN	RW	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 FTO 寄存器定义
5:4	BERM	RW	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。
3	NDET_EN	RW	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	RW	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	RW	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	RW	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信号 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

16.12.10 UART_LIN 波特率同步寄存器(UARTL_LINBSR)

符号	UARTL_LINBSR							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				BAUD_SYNC			
权限	R-0000				R-0000			

符号	UARTL_LINBSR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BAUD_SYNC							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BAUD_SYNC							
权限	R-00000000							

位	位名	权限	功能描述
31:20	RFU	R	未实现：读为 0
19:0	BAUD_SYNC	R	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

16.12.11 UART_LIN 帧超时寄存器(UARTL_LINFTR)

符号	UARTL_LINFTR							
地址	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				FTO			
权限	R-0000				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FTO							
权限	RW-11111111							

位	位名	权限	功能描述
31:12	RFU	R	未实现：读为 0
11:0	FTO	RW	frame timeout 设置 定义超时的波特率个数

16.12.12 UART_LIN 发送时序寄存器(UARTL_LINTTR)

符号	UARTL_LINTTR							
地址	0x0030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	UARTL_LINTTR							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		IBS		RFU	RSS		
权限	R-00		RW-00		R-0	RW-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		BDT		BLT			
权限	R-00		RW-00		RW-0000			

位	位名	权限	功能描述
31:14	RFU	R	未实现：读为 0
13:12	IBS	RW	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	RFU	R	未实现：读为 0
10:8	RSS	RW	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 bau
7:6	RFU	R	未实现：读为 0
5:4	BDT	RW	Break delimiter 长度 00: 1 baud 01: 2 baud 10: 3 baud 11: 4 baud
3:0	BLT	RW	Break 长度 0000: 13 baud 0001: 14 baud … 1110: 27 baud 1111: 28 baud

16.12.13 UART_LIN Pre-sync 波特率寄存器(UARTL_LINPSR)

符号	UARTL_LINPSR							
地址	0x0034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	UARTL_LINPSR							
位名	RFU				PRE_SYNC			
权限	R-0000				R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRE_SYNC							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PRE_SYNC							
权限	R-00000000							

位	位名	权限	功能描述
31:20	RFU	R	未实现：读为 0
19:0	PRE_SYNC	R	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。

16.12.14 UART_LIN break 段长度寄存器(UARTL_LINBKR)

符号	UARTL_LINBKR							
地址	0x0038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU				BREAK_LEN			
权限	R-0000				R-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BREAK_LEN							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	BREAK_LEN							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BREAK_LEN							
权限	R-00000000							

位	位名	权限	功能描述
31:28	RFU	R	未实现：读为 0
27:0	BREAK_LEN	R	break 段长度计数结果，此结果用于 break 接收结束后长度记录， 使用工作时钟计数；软件可读取此寄存器用于查询和计算。

16.12.15 USART 控制状态寄存器(USART_CSR)

符号	USART_CSR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							BUSY

符号	USART_CSR							
权限	R-0000000							R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						TXIREN	RXTOEN
权限	R-000000						RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	CKOEN	OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
权限	R-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	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
权限	RW-01		RW-00		RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:25	RFU	R	未实现：读为 0
24	BUSY	R	UART 通信标志，只读 (Busy) 1：UART 正在通信中 0：UART 空闲
23:18	RFU	R	未实现：读为 0
17	TXIREN	RW	发送红外调制使胜 1：使能红外调制发逿 0：关闭红外调剜
16	RXTOEN	RW	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	RFU	R	未实现：读为 0
14	CKOEN	RW	U7816 时钟 CLK 输出使能控制位 (Clock output Enable) 1：7816 时钟输出使能 0：7816 时钟输出禁止
13	OVSM	RW	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	RW	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	RW	UART RX 下降沿唤醒功能使能寄存器(Negedge Wakeupenable) 1：使能 RX 下降沿唤醒 0：禁止 RX 下降沿唤醒
10	DMATXIFCFG	RW	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

位	位名	权限	功能描述
			中断信号输出 0：是否允许中断信号输出仅由 IE 决定
9	BITORD	RW	数据发送/接收时的位顺序 (Bit Order) 0：LSB first 1：MSB first
8	STOPCFG	RW	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	RW	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	RW	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	RW	接收数据极性配置 (Receive Polarity) 0：正向 1：取反
2	TXPOL	RW	发送数据极性配置 (Transmit Polarity) 0：正向 1：取反
1	RXEN	RW	接收使能，1 有效 (Receive Enable)
0	TXEN	RW	发送使能，1 有效 (Transmit Enable)

16.12.16 USART 中断使能寄存器(USART_IER)

符号	USART_IER							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				RXTO_IE	RXERR_IE	RFU	RXBF_IE
权限	R-0000				RW-0	RW-0	R-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

符号	USART_IER			
位名	NEWUP_IE	RFU	TXBE_IE	TXSE_IE
权限	RW-0	R-00000	RW-0	RW-0

位	位名	权限	功能描述
31:12	RFU	R	未实现：读为 0
11	RXTO_IE	RW	接收超时中断使能，1 有效
10	RXERR_IE	RW	接收错误中断使能，1 有效
9	RFU	R	未实现：读为 0
8	RXBF_IE	RW	接收缓存满中断使能，1 有效
7	NEWUP_IE	RW	RX 下降沿异步检测中断使能，1 有效
6:2	RFU	R	未实现：读为 0
1	TXBE_IE	RW	发送缓存空中断使能，1 有效
0	TXSE_IE	RW	发送缓存空且发送移位寄存器空中断使能，1 有效

16.12.17 USART 中断标志寄存器(USART_ISR)

符号	USART_ISR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					PERR	FERR	OERR
权限	R-00000					W1C-0	W1C-0	W1C-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	RFU			RXTO	RFU		RXBF
权限	R-0	R-000			W1C-0	R-00		W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	NEWKF	RFU				TX_OERR	TXBE	TXSE
权限	W1C-0	R-0000				W1C-0	R-1	W1C-0

位	位名	权限	功能描述
31:19	RFU	R	未实现：读为 0
18	PERR	W1C	奇偶校验错误中断标志，硬件置位，软件写 1 清零
17	FERR	W1C	帧格式错误中断标志，硬件置位，软件写 1 清零
16	OERR	W1C	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
15	RFU	R	未实现：读为 0

位	位名	权限	功能描述
14:12	RFU	R	未实现：读为 0
11	RXTO	W1C	接收超时中断标志，硬件置位，软件写 1 清零
10:9	RFU	R	未实现：读为 0
8	RXBF	W1C	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
7	NEWKF	W1C	RX 下降沿异步检测中断标志，硬件置位，软件写 1 清零
6:3	RFU	R	未实现：读为 0
2	TX_OERR	W1C	发送缓存不为空时,写入新的 TXBUF 硬件置位,软件写 1 清零
1	TXBE	R	发送缓存空中断标志，硬件置位，写入 TXBUF 时清零
0	TXSE	W1C	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

16.12.18 USART 超时和延迟寄存器(USART_TODR)

符号	USART_TODR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TXDLY_LEN							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXTO_LEN							
权限	RW-11111111							

位	位名	权限	功能描述
31:16	RFU	R	未实现：读为 0
15:8	TXDLY_LEN	RW	发送延时长度，单位长度为波特率周期。
7:0	RXTO_LEN	RW	接收超时长度，单位长度为波特率周期。

16.12.19 USART 接收缓冲寄存器(USART_RXBUF)

符号	USART_RXBUF							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							

符号	USART_RXBUF							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							RXBUF
权限	R-0000000							R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RXBUF							
权限	R-00000000							

位	位名	权限	功能描述
31:9	RFU	R	未实现：读为 0
8:0	RXBUF	R	接收数据缓冲寄存器数据 7 位收发时，接收的 7bits 数据存入 RXBUF[6:0]

16.12.20 USART 发送缓冲寄存器(USART_TXBUF)

符号	USART_TXBUF							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							TXBUF
权限	R-0000000							W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXBUF							
权限	W-00000000							

位	位名	权限	功能描述
31:9	RFU	R	未实现：读为 0
8:0	TXBUF	W	发送数据缓冲寄存器数据 7 位收发时，发送的 7bits 数据写入 TXBUF[6:0]

16.12.21 USART 波特率产生寄存器(USART_BGR)

符号	USART_BGR							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							

符号	USART_BGR							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	SPBRG							
权限	RW-00000011							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SPBRG							
权限	RW-01000001							

位	位名	权限	功能描述
31:16	RFU	R	未实现：读为 0
15:0	SPBRG	RW	波特率产生器寄存器值。 注：智能卡模式下 Baud = F7816/(SPBRG + 1)，F7816 为智能卡输出工作时钟频率，由 CODR 寄存器配置

16.12.22 USART 模式寄存器(USART_MCR)

符号	USART_MCR							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					HDSEL	RFU	MODE
权限	R-00000					RW-0	R-0	RW-0

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	HDSEL	RW	UART 半双工单线模式选择 0：双线 UART 1：单线半双工 UART 注意：此寄存器仅在 MODE=0 时有效
1	RFU	R	未实现：读为 0
0	MODE	RW	模块控制寄存器 0：标准 UART 模式 1：智能卡模式（ISO7816-3）

16.12.23 USART 智能卡状态标志寄存器(USART_SCISR)

符号	USART_SCISR							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					WAIT_RPT	TXBUSY	RXBUSY
权限	R-00000					R-0	R-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				TPARERR	RPARERR	FRERR	OVERR
权限	R-0000				W1C-0	W1C-0	W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TXSE	RXIF	TXIF	ERRIF
权限	R-0000				W1C-0	R-0	R-1	R-0

位	位名	权限	功能描述
31:19	RFU	R	未实现：读为 0
18	WAIT_RPT	R	U7816 接口发送了错误信号，正在等待对方重发数据; 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;
17	TXBUSY	R	发送数据忙标志。（发送完成后自动清零） 1 = 处于数据发送状态，发送移位寄存器正在发送数据。（开始发送起始位置 1，停止位中间清零） 0 = 数据发送空闲
16	RXBUSY	R	接收数据忙标志。（接收完成后自动清零） 1 = 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 ignal 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	R	未实现：读为 0
11	TPARERR	W1C	发送数据奇偶校验错误标志位。硬件置位，写 1 清零
10	RPARERR	W1C	接收数据奇偶校验错误标志位。硬件置位，写 1 清零
9	FRERR	W1C	接收帧格式错误标志位。硬件置位，写 1 清零 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	W1C	接收溢出错误标志位。硬件置位，写 1 清零

位	位名	权限	功能描述
			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
7:4	RFU	R	未实现：读为 0
3	TXSE	W1C	7816 数据发送完成标志
2	RXIF	R	接收完成标志，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	R	发送缓冲区空标志，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	R	错误标志，寄存器配置出错或传输过程中出错。硬件置位，清 U7816ERR 中相应错误标志清零 1 = 出错，错误类型见 U7816ERR 寄存器 0 = 无出错情况

16.12.24 USART 智能卡中断使能寄存器(USART_SCIER)

符号	USART_SCIER							
地址	0x0044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TXSE_IE	RXIE	TXIE	LSIE
权限	R-0000				RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:4	RFU	R	未实现：读为 0
3	TXSE_IE	RW	数据发送中断使能位。对应 TXSE 中断标志位 1 = 数据发送中断使能位 0 = 数据发送中断使能位

位	位名	权限	功能描述
2	RXIE	RW	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	RW	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	RW	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

16.12.25 USART 智能卡帧格式控制寄存器(USART_FFCR)

符号	USART_FFCR							
地址	0x0048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				SFREN	ERSW		ERSGD
权限	R-0000				RW-0	RW-00		RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BGTEN	REP_T	PAR		RFREN	TREPEN	RREPEN	DICONV
权限	RW-0	RW-0	RW-00		RW-0	RW-1	RW-1	RW-0

位	位名	权限	功能描述
31:12	RFU	R	未实现：读为 0
11	SFREN	RW	Guard Time 发送长度控制位 1 = Guard time 为 3etu 0 = Guard time 为 2etu
10:9	ERSW	RW	error signal 宽度选择 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	RW	error signal 后 guardtime 宽度选择（仅在发送时有效） 1 = error signal 后 guardtime 为 1~1.5etu。 0 = error signal 后 guardtime 为 2~2.5etu。 error signal 宽度为整数 etu 时 guardtime 为 1.5 或 2.5etu；error signal 宽度为 1.5etu 时 guardtime 为 1 或 2etu

位	位名	权限	功能描述
7	BGTEN	RW	BGT（block guard time）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 1 = BGT 使能，插入 Block guard time(22 etu); 0 = BGT 禁止，不插入 Block guard time(22 etu);
6	REP_T	RW	控制接收数据奇偶校验出错时自动重发次数 1 = 3 次 0 = 1 次
5:4	PAR	RW	奇偶校验类型选择 00：Even 01：Odd 10：Always 1 11：不校验，处理 注：智能卡模式下，必须配置为带校验位
3	RFREN	RW	Guard Time 长度控制位（发送时严格按照协议 2etu） 1 = Guard time 为 1 etu 0 = Guard time 为 2 etu
2	TREPEN	RW	缓发送数据奇偶校验错的处理方式选择 1 = 收到奇偶校验出错标志（error ignal），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	RW	接收数据奇偶校验错的处理方式选择 1 = 奇偶校验错，根据 T=0 协议自动回发 error ignal。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	RW	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

16.12.26 USART 智能卡额外保护时间寄存器(USART_EGTR)

符号	USART_EGTR							
地址	0x004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	USART_EGTR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TXEGT							
权限	RW-00000000							

位	位名	权限	功能描述
31:8	RFU	R	未实现：读为 0
7:0	TXEGT	RW	发送时插入的 EGT 时间数目（以 etu 为单位）

16.12.27 USART 智能卡输出时钟分频寄存器(USART_CODR)

符号	USART_CODR							
地址	0x0050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			CLKDIV				
权限	R-000			RW-00011				

位	位名	权限	功能描述
31:5	RFU	R	未实现：读为 0
4:0	CLKDIV	RW	U7816 时钟输出分频控制寄存器，控制 7816 工作时钟及 clk_out 引脚输出时钟分频数。 U7816 工作时钟 f7816 与系统主时钟 fmclk 的分频关系： f7816=fmclk/(CLK_DIV+1) 特殊情况：CLK_DIV 设置成 0 或 1 时，f7816=fmclk/2 注：7816 协议规定的工作时钟范围是 1~5MHz。

16.12.28 红外调制配置寄存器(UART_IRCR)

符号	UART_IRCR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	UART_IRCR							
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IRFLAG	TH				TZBRG		
权限	RW-0	RW-0000				RW-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TZBRG							
权限	RW-11010010							

位	位名	权限	功能描述
31:16	RFU	R	未实现：读为 0
15	IRFLAG	RW	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	RW	红外占空比调制参数
10:0	TZBRG	RW	红外调制频率

17 串行外设接口（SPI）

17.1 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。

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
- 写冲突错误标志
- 主模式错误检测、保护和中断标志
- 支持 DMA

17.2 SPI 功能描述

17.2.1 SPI 结构框图

下图为 SPI 模块的结构示意图。

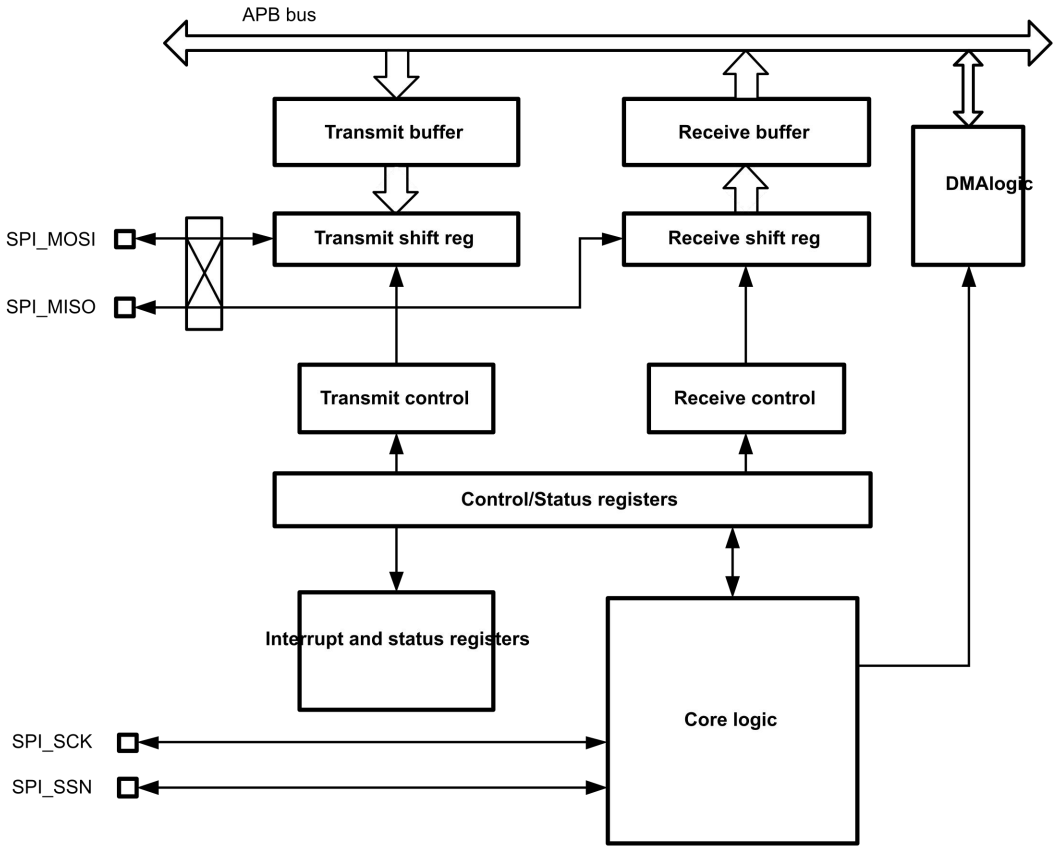


图 17-1 SPI 结构框图

17.2.2 引脚定义

SPI 模块使用 4 个引脚与外部器件通信，具体映射管脚参考引脚和封装。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：

表 17-1 SPI 引脚功能定义

全双工	功能	半双工	功能
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

主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。数据传输时 MSB 在前。

主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。数据传输时 MSB 在前。

串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，通常主机 SSN 输出与从机 SSN 输入直接相连。如果总线上只有一个从机，也可以通过将从机 SSN 接地使从机常使能，此时主机 SSN 输出可以用作 GPIO。

SPI 主从器件的连接如下图所示：

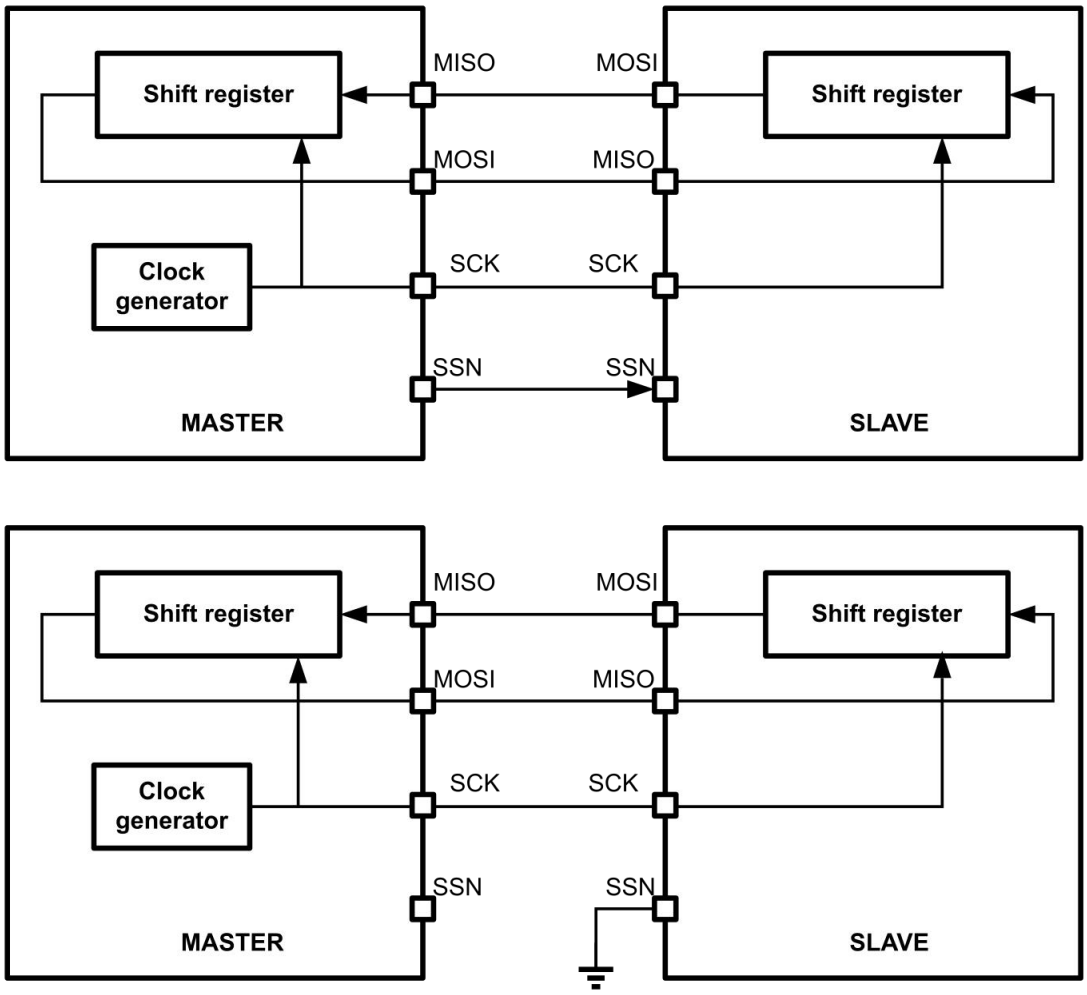


图 17-2 SPI Master/SPI Slave 互连

17.2.3 SPI 接口时序

为了兼容不同的 SPI 外设，SPI 串行时钟的时序可以通过时钟相位选择位（CPHA）和时钟极性选择位(CPOL)设置产生 4 种不同组合。为保证数据正确传输，主从器件的时序配置必需一致。

当处于从器件模式或 SPI 系统使能位（SPIEN）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

17.2.3.1 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。

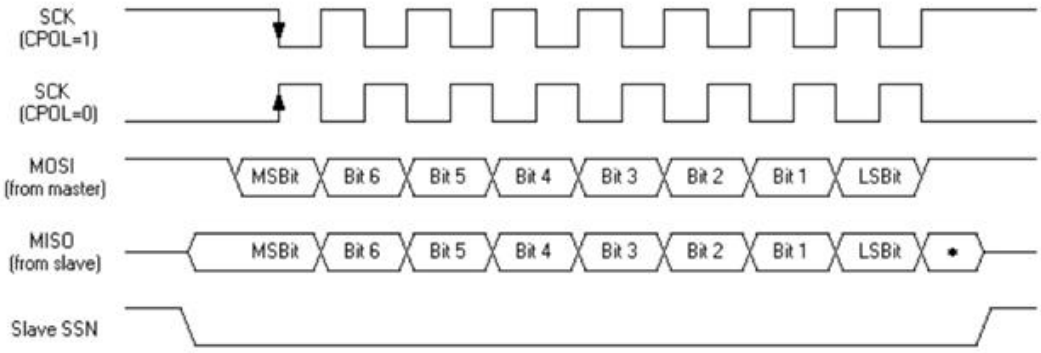


图 17-3 SPI 数据/时钟时序图（CPHA=0）

17.2.3.2 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。

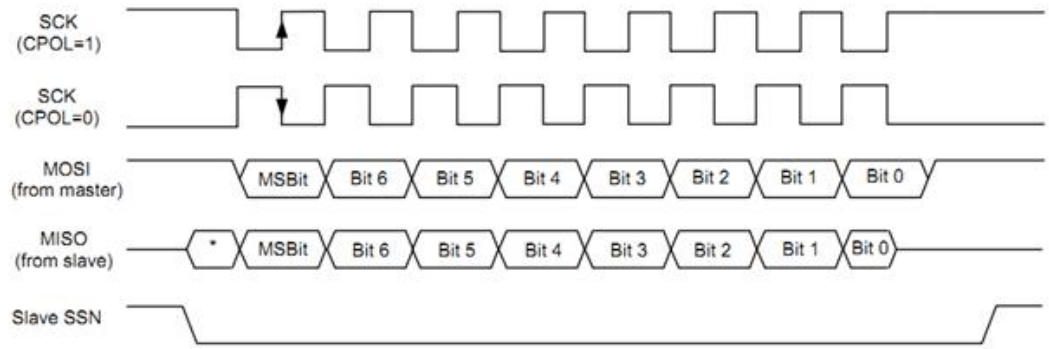


图 17-4 SPI 数据/时钟时序图（CPHA=1）

17.2.3.3 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。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。

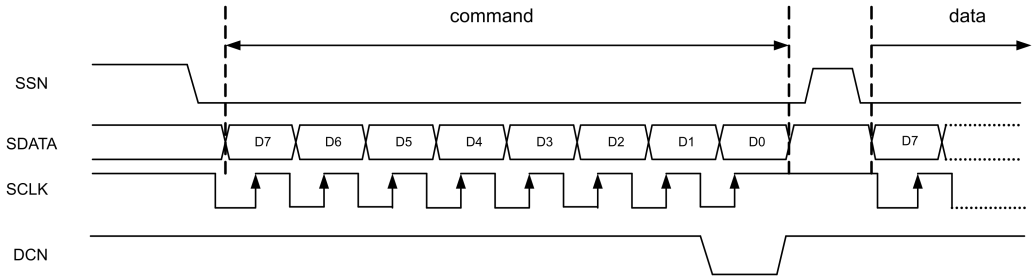


图 17-5 4 线半双工写操作

DCN 在第 8 个时钟上升沿采样判决，如果为 0，表示当前帧是命令帧。发送命令帧前，软件需要将 DCN 寄存器写 0，命令帧发送完成后硬件自动将 DCN 寄存器置位。

4 线半双工读操作

软件通过置位 HD_RW 寄存器，表示当前主机要发起读操作。

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

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

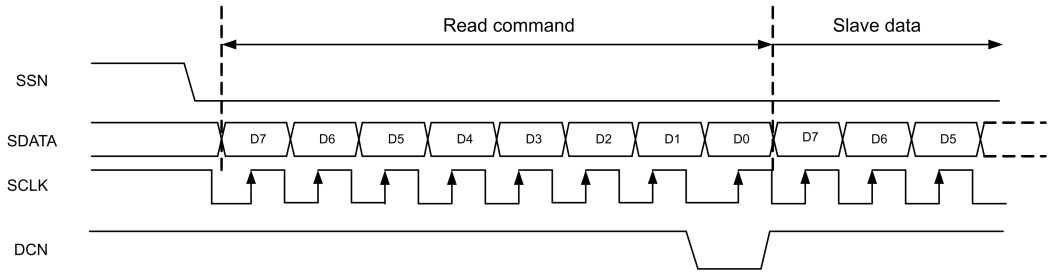


图 17-6 4 线半双工读操作（无 dummy cycle）

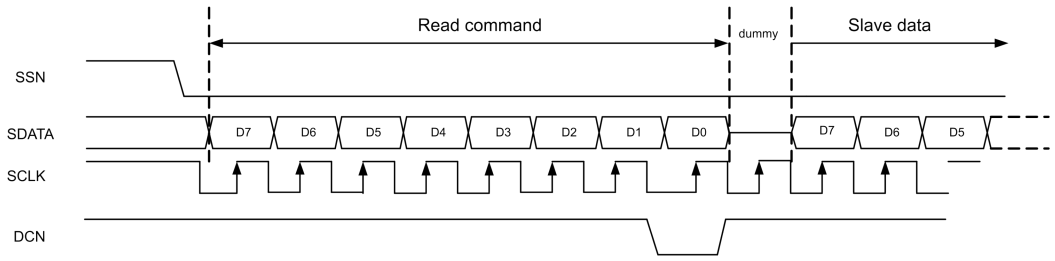


图 17-7 4 线半双工读操作（有 dummy cycle）

17.2.4 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。

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。

BUSY 标志

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

如何启动 SPI 通信

主机模式下，建议遵循以下步骤启动 SPI 通信：

- 应用配置 SPI 模块
- 置位 SPIEN
- 向 TXBUF 写入数据，SPI 模块自动开始发送 SCK 并进行数据收发

从机模式下，建议应用在主机开始发送 SCK 之前完成配置和使能，并将第一帧待发送数据写入 TXBUF，等待主机发送 SCK 开始通信。

如何结束 SPI 通信

主机全双工模式下，等待 RXBF 和 TXSE 标志置位代表最后一帧数据传输完毕。

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。

17.2.5 主机发送间隔

通过 SPIx_CR1.WAIT 寄存器可以配置主机发送两帧数据之间的间隔。当 WAIT=0 时，两帧之间没有间隔。当 WAIT=1 时，两帧之间暂停 WAIT+1 个 SCK 周期。

17.2.6 可编程数据字长

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

17.2.7 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。需注意 RX-ONLY 模式下每次读取数据后 SPI 会自动发送时钟，若希望 SPI 停止发送需在最后一帧数据读取前关闭 RXO 寄存器。

17.2.8 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）；

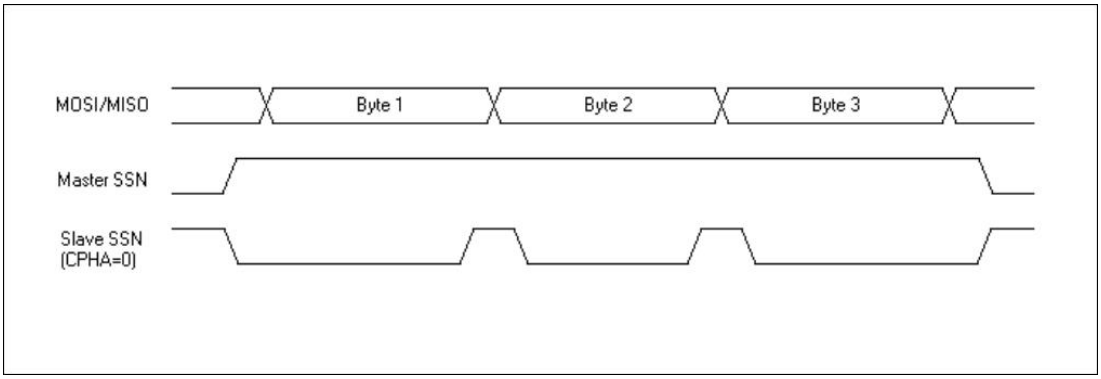


图 17-8 SPI SSN 时序图（SSNM=1，CPHA=0）

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。

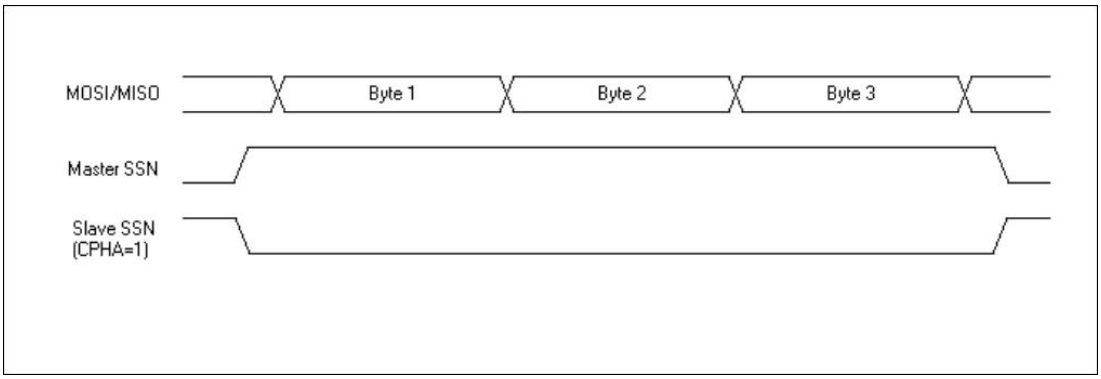


图 17-9 SPI SSN 时序图（SSNM=0）

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。

17.2.9 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。

17.2.10 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 }

17.3 寄存器

SPI1 基地址 (0x40018C00)

SPI0 基地址 (0x40019000)

地址	符号	名称
0x0000	SPI_CR1	SPI 控制寄存器 1
0x0004	SPI_CR2	SPI 控制寄存器 2
0x0008	SPI_CR3	SPI 控制寄存器 3
0x000C	SPI_IER	SPI 中断控制寄存器
0x0010	SPI_ISR	SPI 中断标志寄存器

地址	符号	名称
0x0014	SPI_TXBUF	SPI 发送缓存寄存器
0x0018	SPI_RXBUF	SPI 接收缓存寄存器

17.3.1 SPI 控制寄存器 1(SPI_CR1)

符号	SPI_CR1							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				IOSWAP	MSPA	SSPA	MM
权限	R-0000				RW-0	RW-0	RW-0	RW-1
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	CPOL	CPHA
权限	RW-00		RW-001			RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11	IOSWAP	RW	MOSI 和 MISO 引脚交换 0：默认引脚顺序 1：交换引脚顺序
10	MSPA	RW	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	RW	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	RW	Master/Slave 模式选择。 1：Master 模式 0：Slave 模式
7:6	WAIT	RW	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 硬件控制，并且 SSNM=1，则硬件会自动拉高 SSN。
5:3	BAUD	RW	Master 模式波特率配置位: 000： $f_{APBCLK}/2$

位	位名	权限	功能描述
			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	RW	帧格式 (Frame format) 0：先发送 MSB （Bit7） 1：先发送 LSB （Bit0） 注：当通信在进行时不能改变该位的值。
1	CPOL	RW	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	RW	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。

17.3.2 SPI 控制寄存器 2(SPI_CR2)

符号	SPI_CR2							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN	RFU			RXO	DLEN		HALFDUPLEX
权限	RW-0	R-000			RW-0	RW-00		RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
权限	RW-0	RW-1	RW-0	RW-1	RW-0	RW-1	RW-0	RW-0

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15	DUMMY_EN	RW	4 线半双工协议下是否在读操作中插入 dummy cycle

位	位名	权限	功能描述
			0：不插入 dummy cycle 1：在读命令之后插入一个 dummy cycle
14:12	RFU	R	RFU：未实现，读为 0
11	RXO	RW	RXONLY 控制位，此寄存器置位时，SPI 可以连续接收，无需软件写 TXBUF 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	RW	通信数据字长配置 (Data Length) 00: 8bit 01: 16bit 10: 24bit 11: 32bit
8	HALFDUPLEX	RW	通信模式选择 0：标准 SPI 模式，4 线全双工 1：DCN 模式，4 线半双工
7	HD_RW	RW	半双工模式下主机读写操作配置 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	RW	半双工模式下定义 command 帧长度 1：command 帧固定为 8bit 0：command 帧长度由 DLEN 定义
5	SSNM	RW	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	RW	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	RW	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	RW	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	RW	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	RW	SPI 使能 (SPI enable) 1：使能 SPI

位	位名	权限	功能描述
			0：关闭 SPI，清空发送接收缓存

17.3.3 SPI 控制寄存器 3(SPI_CR3)

符号	SPI_CR3							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TXBFC	RXBFC	MERRC	SERRC
权限	R-0000				W-0	W-0	W-0	W-0

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3	TXBFC	W	写 1 清除发送缓存。硬件自动回 0
2	RXBFC	W	写 1 清除接收缓存。硬件自动回 0
1	MERRC	W	写 1 清除 SPISTA 中 MASTER_ERROR 状态，硬件自动回 0
0	SERRC	W	写 1 清除 SPISTA 中 SLAVE_ERROR 状态，硬件自动回 0

17.3.4 SPI 中断控制寄存器(SPI_IER)

符号	SPI_IER							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		SSNRIE	SSNFIE	TXSEIE	ERRIE	TXBEIE	RXIE
权限	R-00		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:6	RFU	R	RFU：未实现，读为 0
5	SSNRIE	RW	从机 SSN 上升沿中断使能
4	SSNFIE	RW	从机 SSN 下降沿中断使能
3	TXSEIE	RW	发送移位寄存器空中断使能 0：关掉中断 1：使能中断
2	ERRIE	RW	错误中断使能，包括 RXCOL/TXCOL/MERR/SERR 0：关掉中断 1：使能中断
1	TXBEIE	RW	发送缓冲空中断使能 0：关掉中断 1：使能中断
0	RXIE	RW	接收完成中断使能 0：关掉中断 1：使能中断

17.3.5 SPI 中断标志寄存器(SPI_ISR)

符号	SPI_ISR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			DCN_TX	RFU	RXCOL	TXCOL	BUSY
权限	R-000			RW-1	R-0	W1C-0	W1C-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	MERR	SERR	SSNR	SSNF	TXSE	TXBE	RXBF
权限	R-0	R-0	R-0	W1C-0	W1C-0	W1C-0	R-1	R-0

位	位名	权限	功能描述
31:13	RFU	R	RFU：未实现，读为 0
12	DCN_TX	RW	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 onfig) 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

位	位名	权限	功能描述
			DCN_TX 1，即默认只会发送一个命令帧，后续都是数据帧
11	RFU	R	RFU：未实现，读为 0
10	RXCOL	W1C	接收缓存溢出，软件写 1 清零
9	TXCOL	W1C	发送缓存溢出，软件写 1 清零
8	BUSY	R	SPI 空闲标志，只读 (busy flag) 1：SPI 传输进行中 0：SPI 传输空闲 注意：此标志在 SPI 波特率时钟下置位，当软件使能 SPI 发送后，需等待波特率时钟到来才会置位
7	RFU	R	RFU：未实现，读为 0
6	MERR	R	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	R	Slave Error 标志(Slave Error flag)，硬件置位，软件写 1 清零 当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位
4	SSNR	W1C	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	W1C	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	W1C	Tx shift register empty 标志位 0：发送移位寄存器非空 1：发送移位寄存器空
1	TXBE	R	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	R	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

17.3.6 SPI 发送缓存寄存器(SPI_TXBUF)

符号	SPI_TXBUF
地址	0x0014

符号	SPI_TXBUF							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	txbuf							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	txbuf							
权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	txbuf							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	txbuf							
权限	W-00000000							

位	位名	权限	功能描述
31:0	txbuf	W	SPI 发送缓存(Transmit Buffer)

17.3.7 SPI 接收缓存寄存器(SPI_RXBUF)

符号	SPI_RXBUF							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	rxbuf							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	rxbuf							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	rxbuf							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	rxbuf							
权限	R-00000000							

位	位名	权限	功能描述
31:0	rxbuf	R	SPI 接收缓存(Receive Buffer)

18 直接存储访问控制器（DMA）

18.1 概述

- 2 通道外设 PDMA，支持 Peripherals<>RAM 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
- Flash->RAM 通道最大传输长度 8192 字节，只支持 word 传输
- 支持 Flash 连续编程（RAM->Flash），需要预先进行擦除，一次编程固定为 256 字节
- RAM 指针递增、递减
- 可产生半程中断和全程中断
- 通道优先级可配置（4 级优先级）

18.2 结构框图

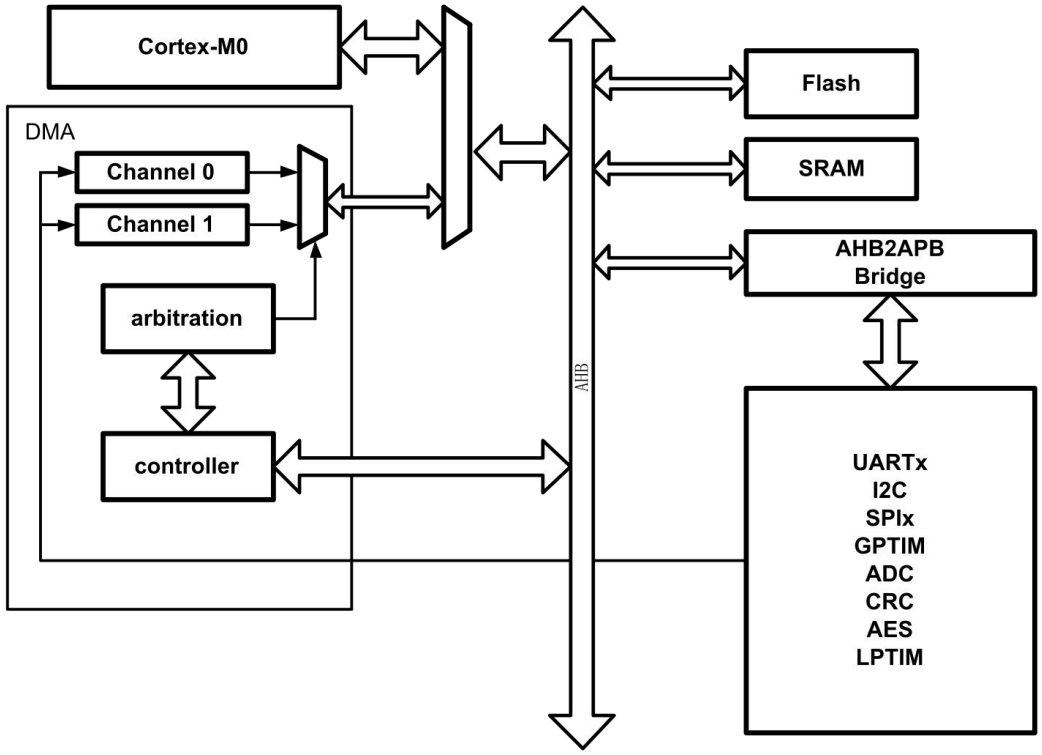


图 18-1 DMA 结构框图

18.3 工作原理

外设 DMA 为 Peripheral<>RAM 通道，采用外设请求触发方式进行数据传输，每个外设通道都可以支持外设->RAM 或者 RAM->外设的数据传输，并且根据目标外设类型的不同，自适应选择

byte/half-word/word 传输方式。DMA 作为 Master，在收到 request 后进行数据读写操作，外设目标地址根据通道接入选择自动定位，RAM 目标地址则根据寄存器配置定位。

每个 channel 可以从多个外设中选择一个作为 source 或 destination，同时软件可以设置通道优先级，当两个通道同时要访问 RAM 时，由优先级决定谁先访问，另一个通道将被挂起，直到优先通道访问完毕。

外设请求可以是准备发送（RAM/Flash->Peripheral）或接收完成（Peripheral->RAM），当 DMA 访问外设时，若 CPU 对同一个外设的访问将引起冲突，哪个 Master 访问被挂起取决于 BusMatrix 设置的仲裁优先级。

通过 DIR 寄存器可以配置每个通道的传输方向，软件必须保证传输方向配置与实际挂载到这个通道上的外设请求相一致。比如通道 1 当前挂载的外设请求是 UART0 接收，则必须将 DIR 寄存器配置为 0（数据从外设读出，写入 RAM），每次 UART0 接收完一帧数据，将发送 RXD0 请求给 DMA，DMA 响应请求后，从 UART0 接收缓存寄存器读取数据，如果 DIR 被错误的配置为 1，则 DMA 对 UART0 接收缓存寄存器的写操作将被 UART0 忽略。

软件可设置 DMA 的存储器指针，用于配置 DMA 传输的起始地址，可以选择指针递增或递减方式。另有 TRFLEN 寄存器配置传输次数，根据起始地址和传输次数，计算得到终止地址，当存储器指针指向终止地址时，本次传输结束，关闭通道。

当 channel 被使能后，DMA 就准备好接受通道所选中的外设请求。当配置传输长度一半的字节被传输后，一个 HTIF（Half transfer interrupt flag）中断置位；当配置传输长度全部完成后，TCIF（Transfer complete interrupt flag）中断置位。上述中断都可以被相应的中断使能寄存器屏蔽。

在 DMA 一个完整 transfer block 完成之前，软件随时可以关闭 channel 使能，此时 DMA 将被挂起，如果软件此后重新使能通道，则 DMA 继续执行之前挂起的操作。

18.4 工作流程

DMA 寄存器配置：



图 18-2 DMA 寄存器配置

DMA 对请求响应分成两部分处理：通道请求处理过程和数据搬运过程。

- 通道请求处理
  1. DMA 接受到请求，跳到步骤 2
  2. 判断是否有其他通道正在搬运数，若有，则停留在步骤 b 直至其他通道当次搬运完成；若无，进一步判断是否有其他同时置起的请求信号，若有，则判断当前通道优先级是否高于其他通道，若是，则跳到步骤 3 并向数据搬运过程发起请求，若否，则停留在步骤 2 直至其他通道当次搬运完成
  3. 并等待数据搬运完成响应信号，得到响应则，跳到步骤 4，否则停在步骤 3
  4. 数据搬运长度+1，判断是否达到设定长度，若是则产生通道使能关闭脉冲；判断请求是否释放，若是，则跳到步骤 1，若否，则停留在步骤 4 判断数据传输达到设定长度，否则跳到步骤 a

- 数据搬运
  1. 等待通道请求处理过程发起请求
  2. 向 HADDR 写源地址
  3. 向 HADDR 写目的地址，同时读取 HRDATA
  4. 将读到的 HRDATA 数据写到 HWDATA
  5. 向通道请求处理过程发出搬运完成响应，并跳到步骤 1

DMA 工作的流程如下图所示：

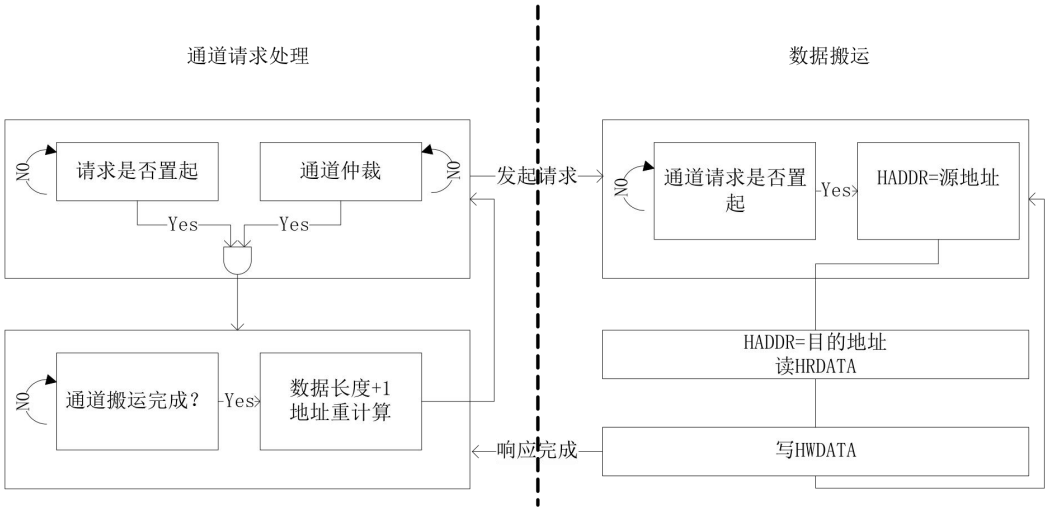


图 18-3 DMA 工作流程

18.5 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。

18.6 通道控制

18.6.1 DMA 请求映射

DMA 共有 2 个优先级可配的外设通道，每个通道可接受 8 个请求响应，通过 DMA_CHxCR.SSEL 寄存器选择其中一个请求送入各个通道。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。

表 18-1 DMA 通道映射列表

SSEL	外设	通道 0	通道 1
0	ADC	ADC	ADC
1	SPI0	SPI0_RX	SPI0_RX
2	SPI1	SPI1_RX	SPI1_TX
3	UART1	RXD1	TXD1
4	UART2	RXD2	TXD2
5	UART4	RXD4	TXD4
6	UART5	RXD5	TXD5

SSEL	外设	通道 0	通道 1
7	CRC	CRC	CRC

上表中从上到下分别表示 CHxSSEL=0~7 情况下有效的外设请求信号。比如针对通道 0，当 CH0SSEL=2 时，被选中的外设请求是 SPI1_RX，即 SPI1 的数据接收 DMA 请求被连接到 DMA 通道 0 的请求输入。

18.6.2 通道优先级

DMA 总共有 2 个外设通道，每个通道的优先级别可以通过寄存器配置为：very high,high,low,very low。当多个通道配置为相同优先级别时，通道序号越大，优先级别越低。

DMA 每搬运完一次数据都会重新进行通道请求选择，假设配置通道 0 传输长度为 3，通道 1 传输长度为 2，通道 1 的优先级高于 0。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。

18.6.3 传输方向定义

在 DMA 通道定义规则中，_RX 表示 DMA 从外设读取数据，写入 RAM，_TX 表示 DMA 从 RAM/Flash 读取数据，写入外设。

软件在配置每个通道的外设分配之后，还需要配置 CHx_DIR 寄存器设定通道传输方向，错误的方向设置会导致 DMA 无法正常工作。

18.6.4 循环模式

外设 DMA 通道支持循环模式（Circular mode）。循环模式下，当 CHxTSIZE 寄存器定义的传输长度完成后，DMA 不会自动停止，而是返回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。

存储 DMA 通道不支持循环模式。

18.7 寄存器

DMA 基地址 (0x40000400)

地址	符号	名称
0x0000	DMA_GCR	DMA 全局控制器
0x0004	DMA_CH0CR	通道 0 控制寄存器
0x0008	DMA_CH0MAD	通道 0 存储器指针寄存器
0x000C	DMA_CH1CR	通道 1 控制寄存器
0x0010	DMA_CH1MAD	通道 1 存储器指针寄存器
0x003C	DMA_CH7CR	通道 7 控制寄存器
0x0040	DMA_CH7FLSAD	通道 7 FLS 指针寄存器
0x0044	DMA_CH7RAMAD	通道 7 RAM 指针寄存器
0x0048	DMA_ISR	DMA 中断状态标志寄存器

地址	符号	名称
0x004C	DMA_CH0TFSADDR	通道 0 传输剩余长度
0x0050	DMA_CH1TFSADDR	通道 1 传输剩余长度

18.7.1 DMA 全局控制器(DMA_GCR)

符号	DMA_GCR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						ADDRERR_EN	EN
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	ADDRERR_EN	RW	DMA 地址错误中断使能
0	EN	RW	DMA 全局使能 1：DMA 使能 0：DMA 关闭

18.7.2 通道 0 控制寄存器(DMA_CH0CR)

符号	DMA_CH0CR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TSIZE							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		PRI		INC	SSEL		
权限	R-00		RW-00		RW-0	RW-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SUSP	DIR	BDW		CIRC	FTIE	HTIE	EN

符号	DMA_CH0CR						
权限	RW-0	RW-0	RW-00	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:16	TSIZE	RW	Channelx 传输长度，最大 65536 次传输，实际传输长度是 TSIZE+1
15:14	RFU	R	RFU：未实现，读为 0
13:12	PRI	RW	优先级 2'b00:Low 2'b01:Medium 2'b10:High 2'b11:Very high
11	INC	RW	1'b1：MEM 地址递增 1'b0：MEM 地址递减
10:8	SSEL	RW	3'b000 ADC→MEM 3'b001 SPI0→MEM 3'b010 SPI1→MEM 3'b011 UART1→MEM 3'b100 UART2→MEM 3'b101 UART4→MEM 3'b110 UART5→MEM 3'b111 MEM->CRC
7	SUSP	RW	通道暂停控制 1：暂停本通道传输，内部传输状态不复位 0：回复通道传输 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 情况下有效 USP 清零后 DMA 通道恢复传输
6	DIR	RW	通道传输方向，在每次传输前需正确配置传输方向。 0：从外设读取数据写入 MEM 1：从 MEM 读取数据写入外设
5:4	BDW	RW	传输带宽设置 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	CIRC	RW	循环缓冲模式 0：关闭循环模式 1：使能循环模式
2	FTIE	RW	通道 0 传输完毕中断使能
1	HTIE	RW	通道 0 半满中断使能
0	EN	RW	通道 0 使能

18.7.3 通道 0 存储器指针寄存器(DMA_CH0MAD)

符号	DMA_CH0MAD							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	DMA_CH0MAD							
位名	MEMAD							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MEMAD							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MEMAD							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	MEMAD	RW	Channelx 存储器指针地址，DMA 传输启动前软件向此寄存器写入存储器目标地址。(Channel x Memory Address ointer) 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。

18.7.4 通道 1 控制寄存器(DMA_CH1CR)

符号	DMA_CH1CR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TSIZE							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		PRI		INC	CH1_SSEL		
权限	R-00		RW-00		RW-0	RW-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	SUSP	DIR	BDW		CIRC	FTIE	HTIE	EN
权限	RW-0	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:16	TSIZE	RW	Channelx 传输长度，最大 65536 次传输，实际传输长度是 TSIZE+1
15:14	RFU	R	RFU：未实现，读为 0
13:12	PRI	RW	优先级 2’b00:Low 2’b01:Medium 2’b10:High 2’b11:Very high

位	位名	权限	功能描述
11	INC	RW	1’b1：MEM 地址递增 1’b0：MEM 地址递减
10:8	CH1_SSEL	RW	3’b000 ADC->MEM 3’b001 MEM->SPI0 3’b010 MEM->SPI1 3’b011 MEM->UART1 3’b100 MEM->UART2 3’b101 MEM->UART4 3’b110 MEM->UART5 3’b111 MEM->CRC
7	SUSP	RW	通道暂停控制 1：暂停本通道传输，内部传输状态不复位 0：回复通道传输 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 情况下有效 USP 清零后 DMA 通道恢复传输
6	DIR	RW	通道传输方向 0：从外设读取数据写入 MEM 1：从 MEM 读取数据写入外设
5:4	BDW	RW	传输带宽设置 00：字节，8bit 01：半字，16bit 10：字，32bit 11：RFU
3	CIRC	RW	循环缓冲模式 0：关闭循环模式 1：使能循环模式
2	FTIE	RW	通道 1 传输完毕中断使能
1	HTIE	RW	通道 1 半满中断使能
0	EN	RW	通道 1 使能

18.7.5 通道 1 存储器指针寄存器(DMA_CH1MAD)

符号	DMA_CH1MAD							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	MEMAD							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MEMAD							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MEMAD							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MEMAD							

符号	DMA_CH1MAD
权限	RW-00000000

位	位名	权限	功能描述
31:0	MEMAD	RW	Channelx 存储器指针地址，DMA 传输启动前软件向此寄存器写入存储器目标地址。(Channel x Memory Address ointer) 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。

18.7.6 通道 7 控制寄存器(DMA_CH7CR)

符号	DMA_CH7CR							
地址	0x003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU				TSIZE			
权限	R-0000				RW-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TSIZE							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		PRI		RFU	DIR	RI	FI
权限	R-00		RW-00		R-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					FTIE	HTIE	EN
权限	R-00000					RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:28	RFU	R	RFU：未实现，读为 0
27:16	TSIZE	RW	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 32 次传输
15:14	RFU	R	RFU：未实现，读为 0
13:12	PRI	RW	优先级 2’b00:Low 2’b01:Medium 2’b10:High 2’b11:Very high
11	RFU	R	RFU：未实现，读为 0
10	DIR	RW	1.FLS->RAM 0.RAM->FLS
9	RI	RW	仅在 FLS->RAM 有效，RAM->FLS 固定为递增 1.RAM 地址递增 0.RAM 地址递减
8	FI	RW	仅在 FLS->RAM 有效，RAM->FLS 固定为递增 1.FLSM 地址递增 0.FLS 地址递减
7:3	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
2	FTIE	RW	通道 7 传输完毕中断使能
1	HTIE	RW	通道 7 半满中断使能
0	EN	RW	通道 7 使能

18.7.7 通道 7 FLS 指针寄存器(DMA_CH7FLSAD)

符号	DMA_CH7FLSAD							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DFLS_SEL	RFU						
权限	RW-0	R-0000000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	FLSAD						
权限	R-0	RW-0000000						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FLSAD							
权限	RW-00000000							

位	位名	权限	功能描述
31	DFLS_SEL	RW	Flash 指针对象配置 0：Flash 地址指针指向 code flash 1：Flash 地址指针指向 data flash
30:15	RFU	R	RFU：未实现，读为 0
14:0	FLSAD	RW	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 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

18.7.8 通道 7 RAM 指针寄存器(DMA_CH7RAMAD)

符号	DMA_CH7RAMAD							
地址	0x0044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	DMA_CH7RAMAD							
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				RAMAD			
权限	R-0000				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RAMAD							
权限	RW-00000000							

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:0	RAMAD	RW	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 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

18.7.9 DMA 中断状态标志寄存器(DMA_ISR)

符号	DMA_ISR							
地址	0x0048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						ADDRERR	RFU
权限	R-000000						W1C-0	R-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CHFT7	RFU					CHFT1	CHFT0
权限	W1C-0	R-00000					W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHHT7	RFU					CHHT1	CHHT0
权限	W1C-0	R-00000					W1C-0	W1C-0

位	位名	权限	功能描述
31:18	RFU	R	RFU：未实现，读为 0
17	ADDRERR	W1C	DMA 传输地址错误标志，硬件置位，软件写 1 清零
16	RFU	R	RFU：未实现，读为 0
15	CHFT7	W1C	存储器通道传输完毕标志
14:10	RFU	R	RFU：未实现，读为 0
9	CHFT1	W1C	通道 1 传输完毕标志，硬件置位，软件写 1 清零
8	CHFT0	W1C	通道 0 传输完毕标志，硬件置位，软件写 1 清零

位	位名	权限	功能描述
7	CHHT7	W1C	存储器通道半满标志
6:2	RFU	R	RFU：未实现，读为 0
1	CHHT1	W1C	通道 1 半满标志，硬件置位，软件写 1 清零
0	CHHT0	W1C	通道 0 半满标志，硬件置位，软件写 1 清零

18.7.10 通道 0 传输剩余长度(DMA_CH0TFSADDR)

符号	DMA_CH0TFSADDR							
地址	0x004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TFSIZE_CNT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TFSIZE_CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	TFSIZE_CNT	R	Channe0 传输剩余长度

18.7.11 通道 1 传输剩余长度(DMA_CH1TFSADDR)

符号	DMA_CH1TFSADDR							
地址	0x0050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TFSIZE_CNT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TFSIZE_CNT							
权限	R-00000000							

位	位名	权限	功能描述

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	TFSIZE_CNT	R	Channe1 传输剩余长度

19 循环冗余校验（CRC）

19.1 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， CRC 计算单元为完全独立模块，通过软件控制可进行 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

19.2 软件配置过程

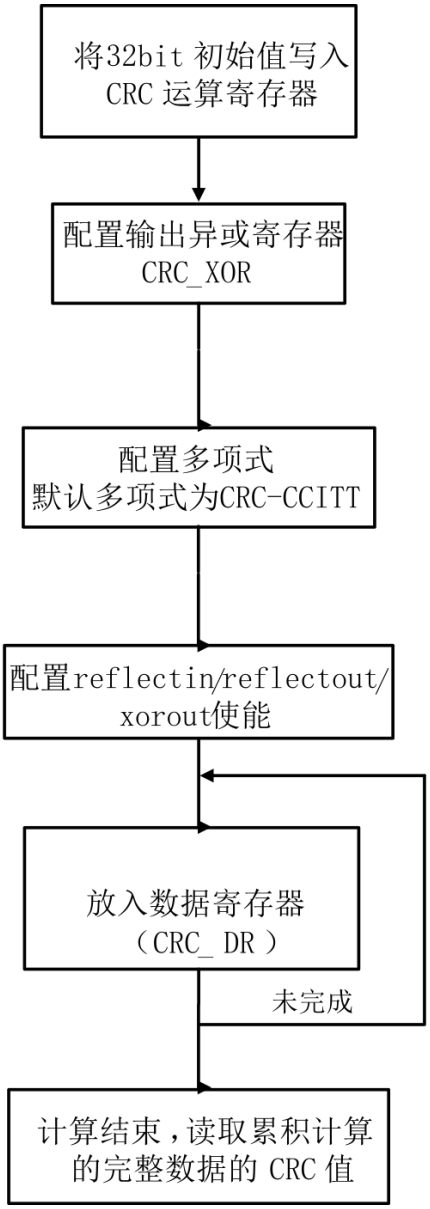


图 19-1 CRC 运算流程图

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。

19.3 Golden 数据

提供 Golden 数据表格供应用中测试及校验使用，下表格中得到的运算结果是输入输出不翻转、输出不异或的结果。

表 19-1 CRC 的 Golden 数据表

多项式	输入数据	初始值 0x0	初始值 0xffff	初始值 0x6363
0x0007	5A5A	0F	D8	C5
0x0007	1223344	F9	28	96
0x8005	5A5A	5DD9	DDD4	9696
0x8005	11223344	7D35	7D11	4698
0x1021	5A5A	1ACB	07C4	1877
0x1021	11223344	DD33	59F3	DD06

19.4 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。

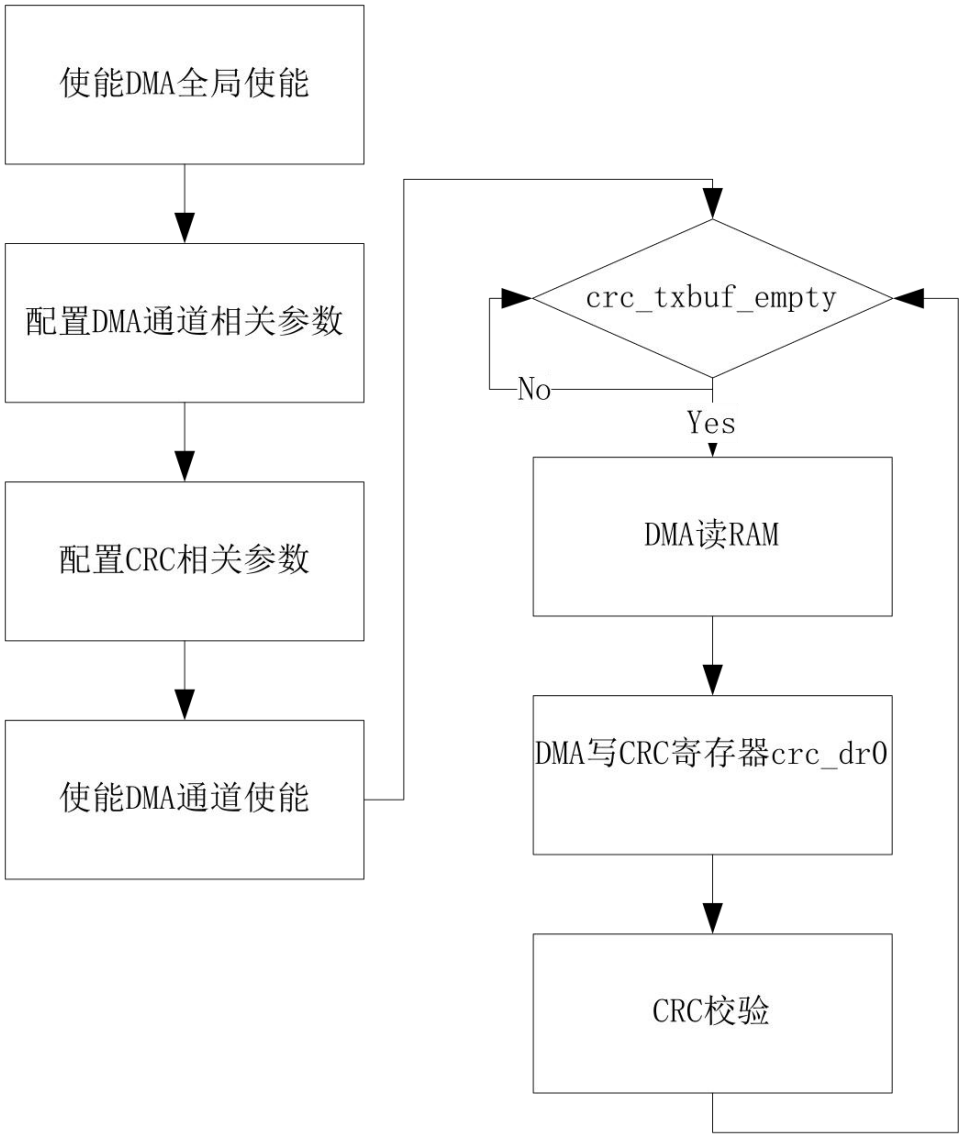


图 19-2 使用 DMA 对 RAM 中的数据进行 CRC 运算

19.5 Flash 数据完整性校验

如果需要 CRC 进行 Flash 内容的完整性校验，可以使用 DMA 将存储器指针直接指向 flash，DMA 可以直接将 Flash 数据搬运到 CRC 模块进行校验。

19.6 寄存器

CRC 基地址 (0x40018000)

地址	符号	名称
0x0000	CRC_DR	CRC 数据寄存器
0x0004	CRC_CR	CRC 控制状态寄存器

地址	符号	名称
0x0008	CRC_LFSR	CRC 运算寄存器
0x000C	CRC_XOR	CRC 异或输出寄存器
0x001C	CRC_POLY	CRC 多项式寄存器

19.6.1 CRC 数据寄存器(CRC_DR)

符号	CRC_DR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DR							
权限	RW-11111111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DR							
权限	RW-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DR							
权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DR							
权限	RW-11111111							

位	位名	权限	功能描述
31:0	DR	RW	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。 作为输入时：若为 Flash RC 计算或 word 操作使能，则对 crc_dr[31:0]进行计算，共 4 次 byte 运算（由低到高）；否则对 crc_dr[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 crc_dr[6:0]，若为 8 位多项式结果保存在 crc_dr[7:0]，若为 16 位多项式结果保存在 crc_dr[15:0]，若为 31 位多项式结果保存在 crc_dr[31:0]。

19.6.2 CRC 控制状态寄存器(CRC_CR)

符号	CRC_CR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	CRC_CR							
位名	RFU						OPWD	RFU
权限	R-000000						RW-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
权限	RW-00		RW-0	R-0	R-0	RW-0	RW-01	

位	位名	权限	功能描述
31:10	RFU	R	RFU：未实现，读为 0
9	OPWD	RW	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	R	RFU：未实现，读为 0
7:6	RFLTIN	RW	reflectin 位，表示输入数据是否反转 00：不反转 01：按字节反转 10：按半字反转 11：按字反转 例如： 计算数据为 11223344h， 如果 reflectin==00，则直接使用 11223344h 进行计算 如果 reflectin==01，则需要将数据变为 8844CC22h，再进行计算 如果 reflectin==10，则需要将数据变为 448822CCh，再进行计算 如果 reflectin==11，则需要将数据变为 22CC4488h，再进行计算
5	RFLTO	RW	CRC 输出反转控制 0：输出不反转 1：输出按字反转 例如： 如果 RFLTO==1，若当前计算的 CRC 结果为 1234h，则输出的结果为 2C48h 如果 RFLTO==0.则直接输出 1234h 注意：此结果不一定为最终输出结果，还需要看 bit2 是否为 1，详见本寄存器 bit2 说明
4	RES	R	CRC 结果标志位，只读 0：CRC 结果为 0 1：CRC 结果非全 0
3	BUSY	R	CRC 运算标志位，只读 0：CRC 运算结束 1：CRC 运算进行中
2	XOR	RW	输出异或使能 0：输出不异或 CRC_XOR 寄存器

位	位名	权限	功能描述
			1：输出异或 CRC_XOR 寄存器
1:0	SEL	RW	CRC 校验多项式选择 00：CRC32 01：CRC16 10：CRC8 11：CRC7

19.6.3 CRC 运算寄存器(CRC_LFSR)

符号	CRC_LFSR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	LFSR							
权限	RW-11111111							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	LFSR							
权限	RW-11111111							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	LFSR							
权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	LFSR							
权限	RW-11111111							

位	位名	权限	功能描述
31:0	LFSR	RW	CRC LFSR 寄存器，运算开始前可以由软件写入初始值

19.6.4 CRC 异或输出寄存器(CRC_XOR)

符号	CRC_XOR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	XOR							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	XOR							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	XOR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	XOR							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	XOR	RW	CRC 运算结果异或寄存器,当 CRC_CR.XOR 为 1 时，CRC 结果输出前将异或此寄存器的数据。

19.6.5 CRC 多项式寄存器(CRC_POLY)

符号	CRC_POLY							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	POLY							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	POLY							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	POLY							
权限	RW-00010000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	POLY							
权限	RW-00100001							

位	位名	权限	功能描述
31:0	POLY	RW	CRC 计算时使用的多项式

20 高级定时器（ATIM）

20.1 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。

本产品 ATIM 为同步设计，工作时钟就是总线时钟。

20.2 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，两个内部通道
- 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

20.3 结构框图

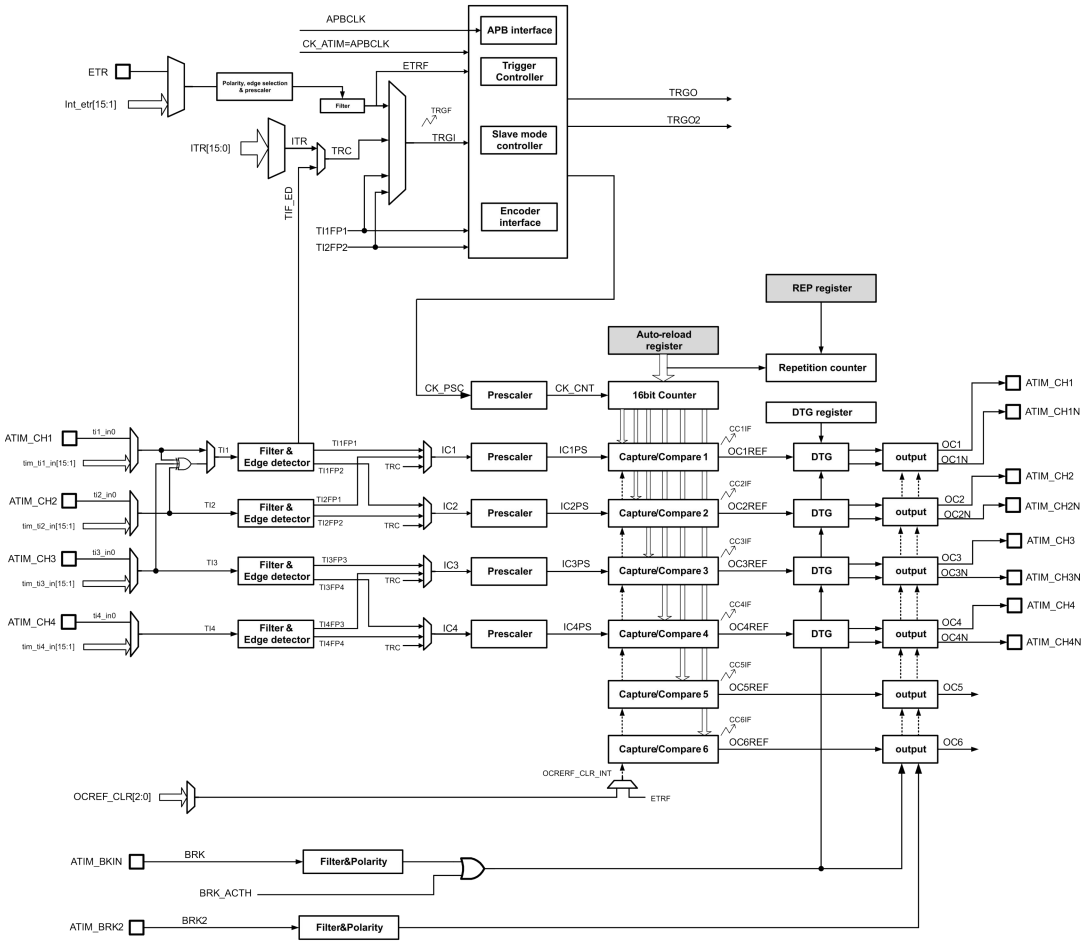


图 20-1 高级定时器结构框图

20.4 管脚和内部信号连接

下表罗列了 ATIM 相关的输入输出引脚。

表 20-1 外部引脚连接

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

Pin name	type	Description
		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

ATIM 与芯片内部信号连接：

表 20-2 内部信号连接

signal name	type	Description
ITR[15:0]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_ATIM 的 1/4）
Int_etr[15:1]	input	外部触发输入，可用于触发信号或者外部计数时钟 在使能了输入预分频的情况下，可以输入高于 CK_ATIM 频率的计数时钟
TRGO TRGO2	output	内部触发信号输出，用于连接到其他 timer 或外设，比如 ADC、DAC
OCREF_CLR[7:0]	input	OCREF 清零信号输入，用于片内特殊信号异步清零 OCREF
APBCLK	input	寄存器总线时钟
CK_ATIM	Input	ATIM 工作时钟，与 APBCLK 同频
ATIM_CC1_DMA ATIM_CC2_DMA ATIM_CC3_DMA ATIM_CC4_DMA	output	4 个通道各自的 DMA 请求输出
ATIM_UEV_DMA	output	更新事件 DMA 请求
ATIM_TRG_DMA	output	触发信号 DMA 请求
ATIM_COM_DMA	output	COM 事件 DMA 请求

ATIM 的 ITR[15:0]与片内信号连接：

表 20-3 ITR 信号连接

ITRx	ATIM0
ITR0	TAU10_TRGO
ITR1	TAU11_TRGO
ITR2	–
ITR3	–
ITR4	GTIM_TRGO
ITR5	BTIM_TRGO
ITR6	LPT_TRGO
ITR7	TAU01_TRGO
ITR8	TAU02_TRGO

ITRx	ATIM0
ITR9	TAU03_TRGO
ITR10	TAU04_TRGO
ITR11	TAU05_TRGO
ITR12	TAU06_TRGO
ITR13	TAU07_TRGO
ITR14	
ITR15	

ATIM 的 ETR[15:0]与片内信号连接：

表 20-4 ETR 信号连接

ETRx	ATIM0
ETR0	ATIM0_ETR
ETR1	COMP1_OUT
ETR2	–
ETR3	–
ETR4	–
ETR5	–
ETR6	–
ETR7	–
ETR8	–
ETR9	–
ETR10	–
ETR11	–
ETR12	–
ETR13	–
ETR14	–
ETR15	–

ATIM 的 OCREF_CLR 与内部信号连接：

表 20-5 OCREF_CLR 信号连接

OCREF_CLRx	ATIM0
OCREF_CLR0	COMP1_OUT
OCREF_CLR1	–
OCREF_CLR2	–

20.5 功能描述

20.5.1 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 ATIM_CR1.ARPE（Auto Reload Preload Enable）寄存器控制。当 ATIM_CR1.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 ATIM_CR1.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（ATIM_CR1.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。ATIM_PSC 寄存器同样被缓存(Preload 不可关闭)，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。

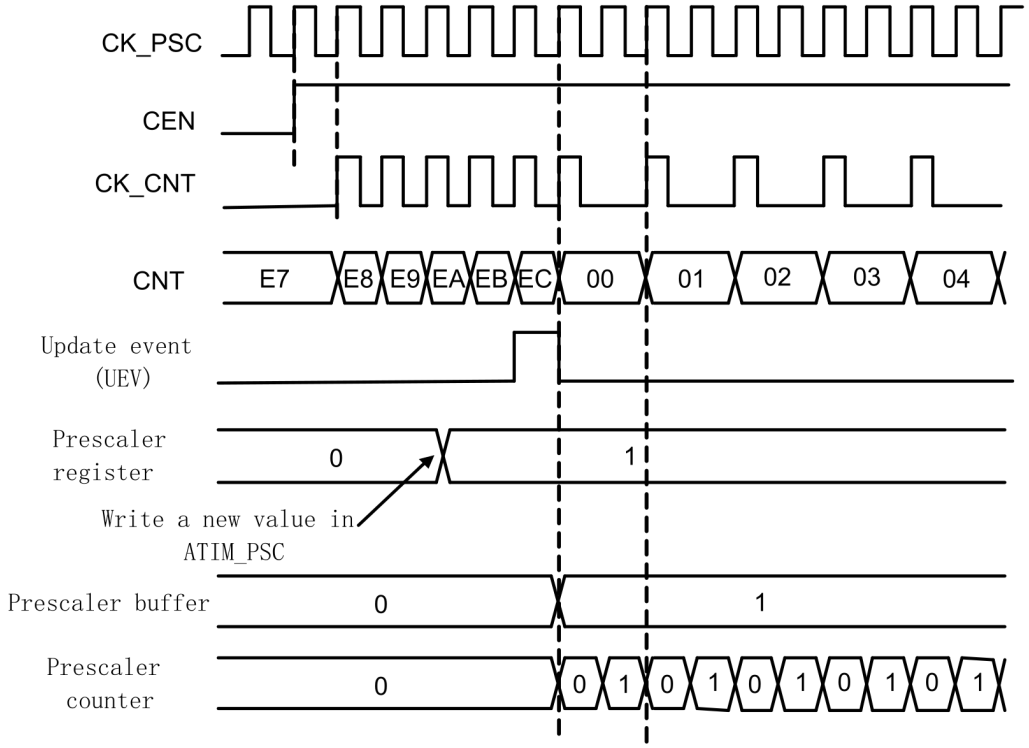


图 20-2 预分频从 1 变为 2 的波形

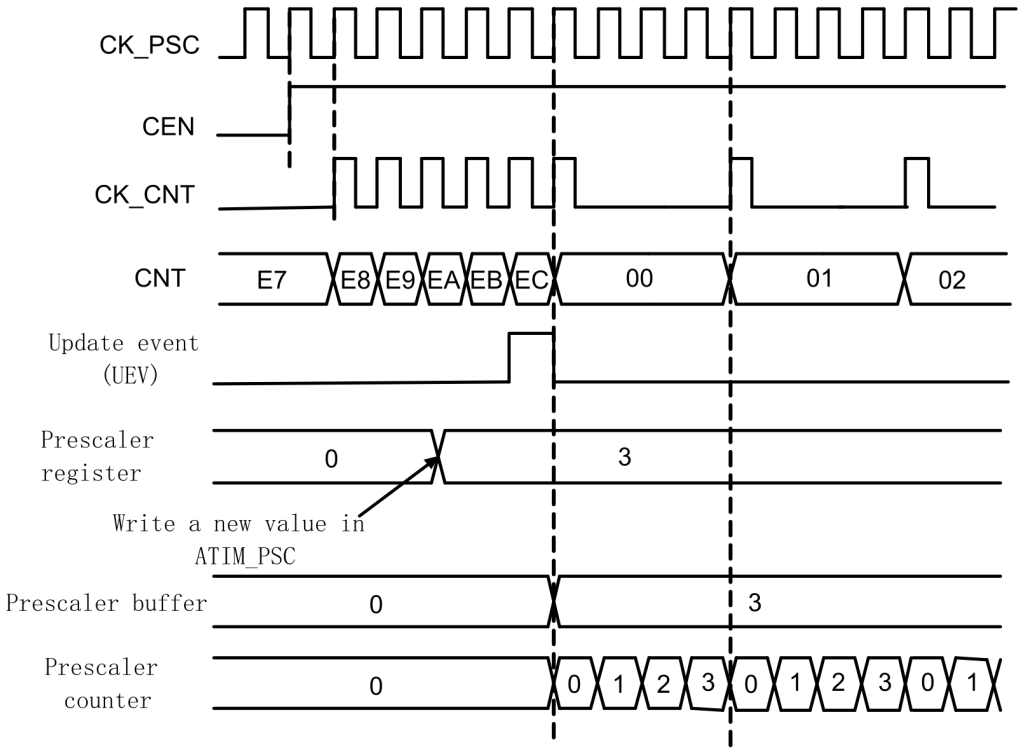


图 20-3 预分频从 1 变为 4 的波形

20.5.2 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 ATIM_EGR.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 ATIM_EGR.UG 寄存器是否触发 ATIM_ISR.UIF（Update Interrupt Flag）中断标志置位由 ATIM_CR1.URS 寄存器的设置决定。

通过设置 ATIM_CR1.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 ATIM_ISR.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

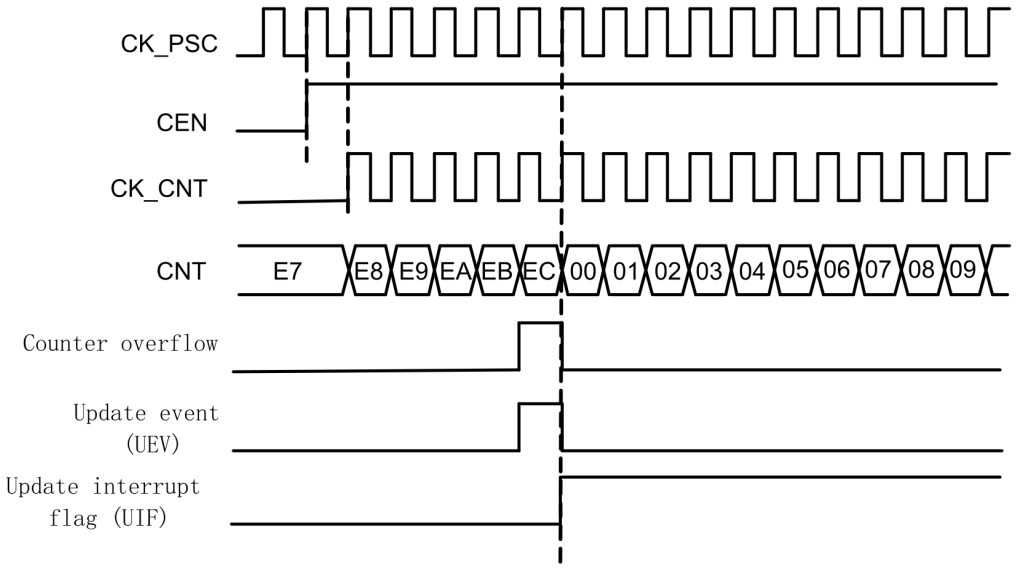


图 20-4 向上计数波形，内部时钟不分频

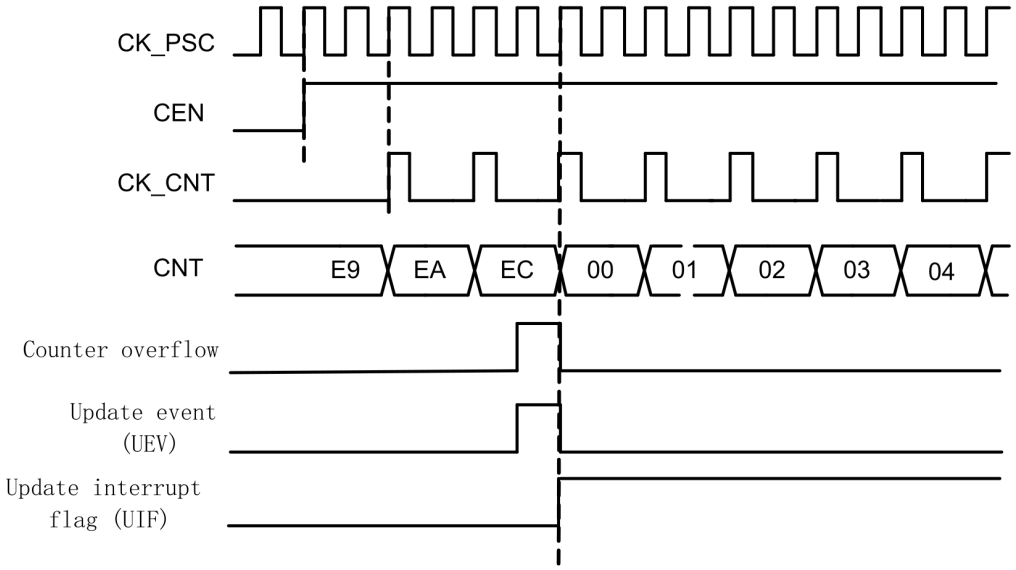


图 20-5 向上计数波形，内部时钟 2 分频

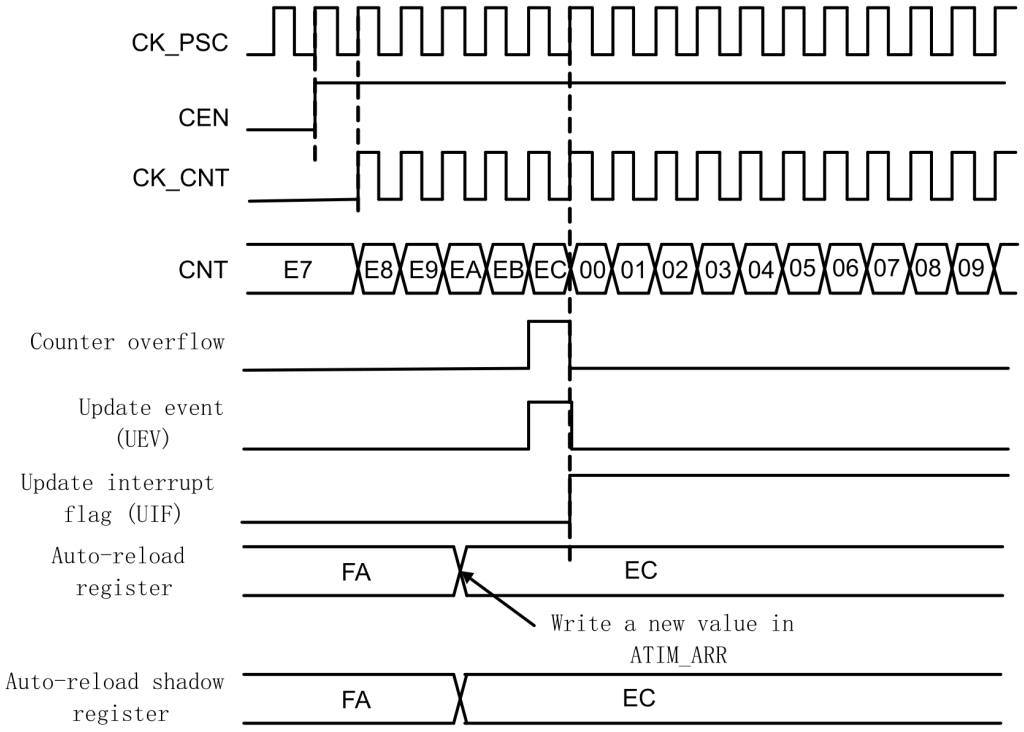


图 20-6 ATIM_CR1.ARPE=0（ATIM_ARR 没有预装载）时的更新事件

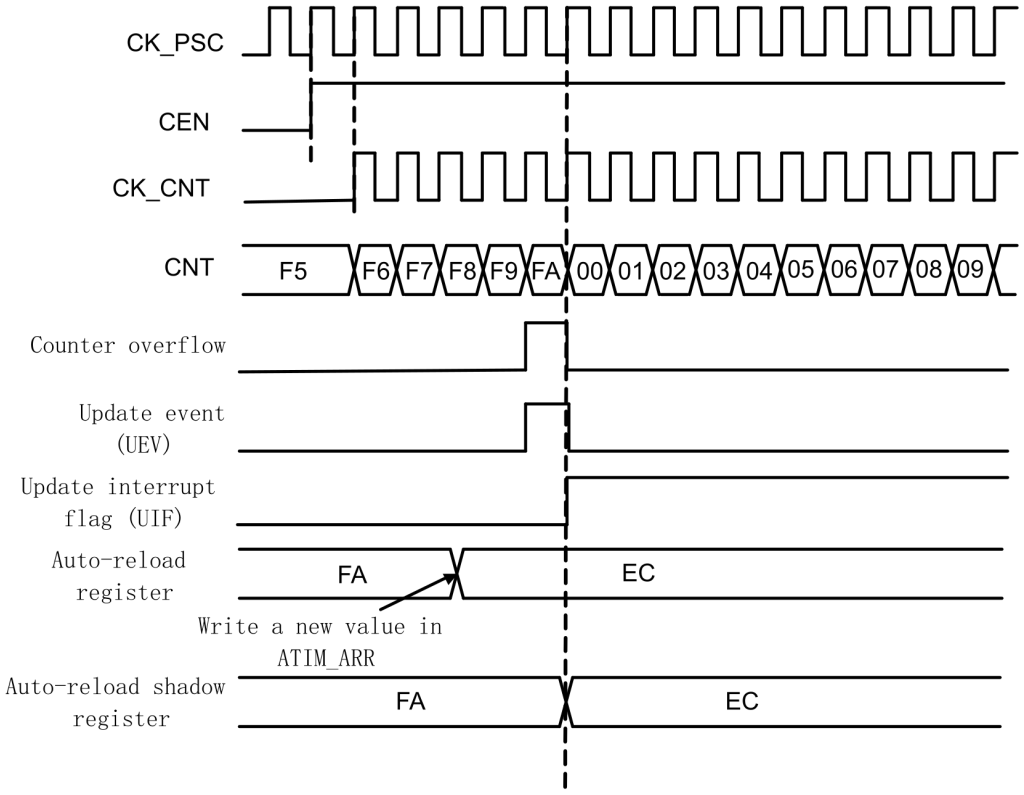


图 20-7 ATIM_CR1.ARPE=1（ATIM_ARR 预装载）时的更新事件

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 ATIM_EGR.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 ATIM_EGR.UG 寄存器是否触发 ATIM_ISR.UIF（Update Interrupt Flag）中断标志置位由 ATIM_CR1.URS 寄存器的设置决定。

通过设置 ATIM_CR1.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 ATIM_ISR.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

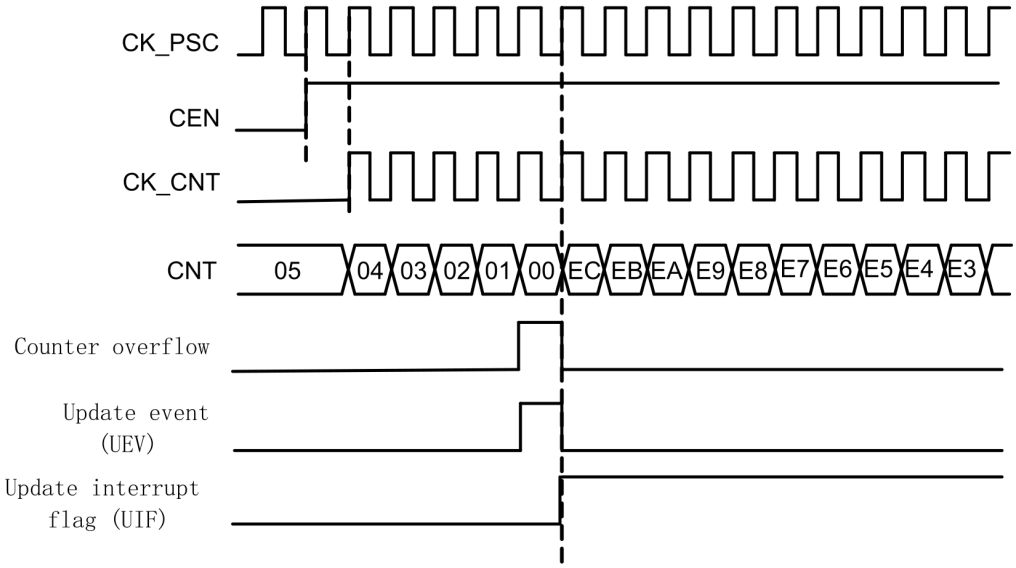


图 20-8 向下计数，内部时钟不分频

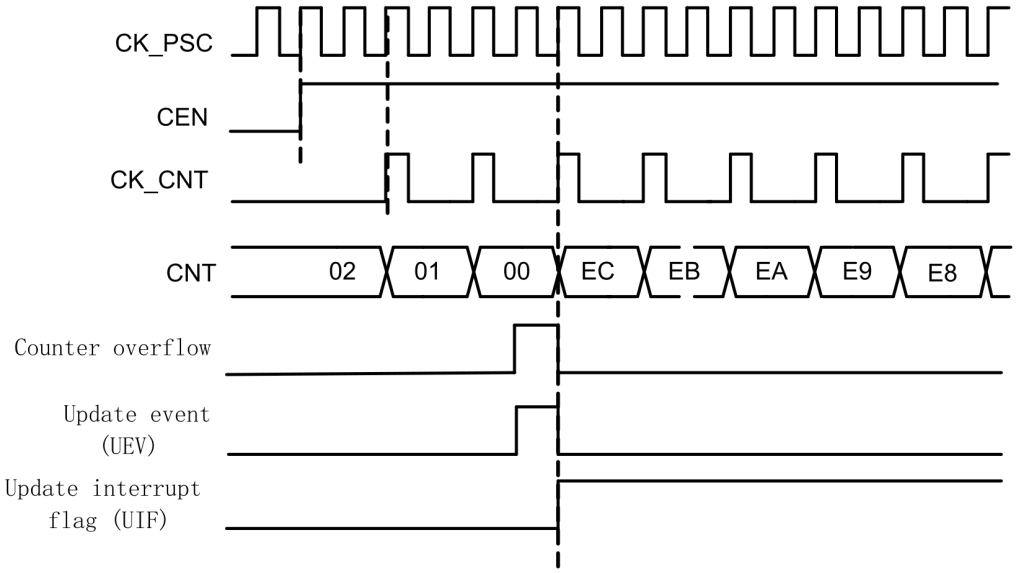


图 20-9 向下计数，内部时钟 2 分频

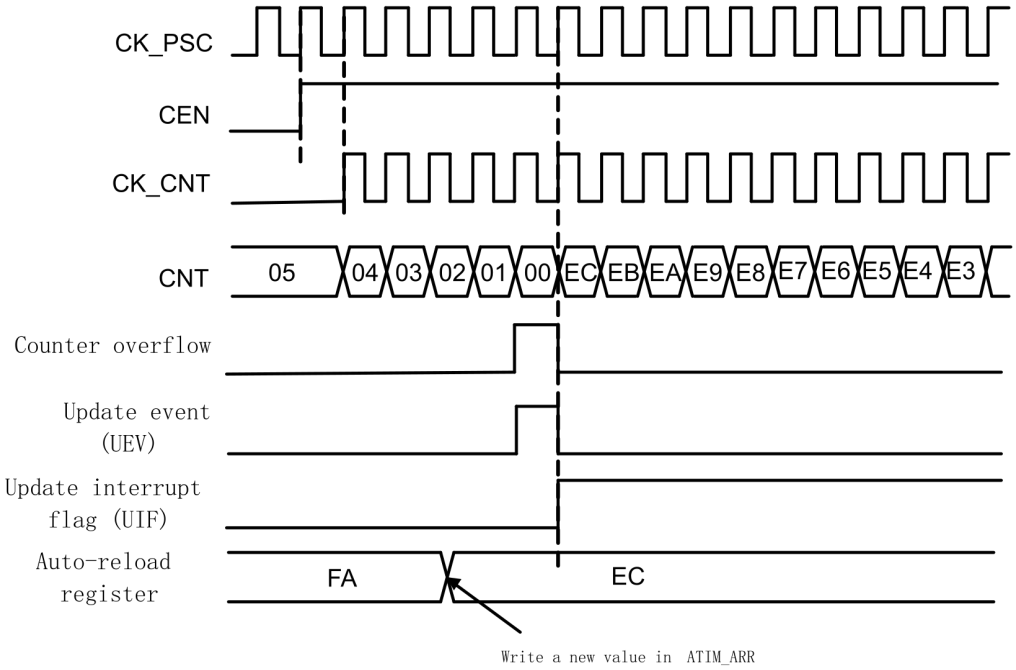


图 20-10 向下计数，不使用重复计数时的更新事件

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。

ATIM_CR1.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 ATIM_CR1.CMS!=00 时为中心对齐计数，当 ATIM_CR1.CMS=01 时，输出比较功能仅在向下计数时有效，当 ATIM_CR1.CMS=10 时，输出比较功能仅在向上计数时有效，当 ATIM_CR1.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。

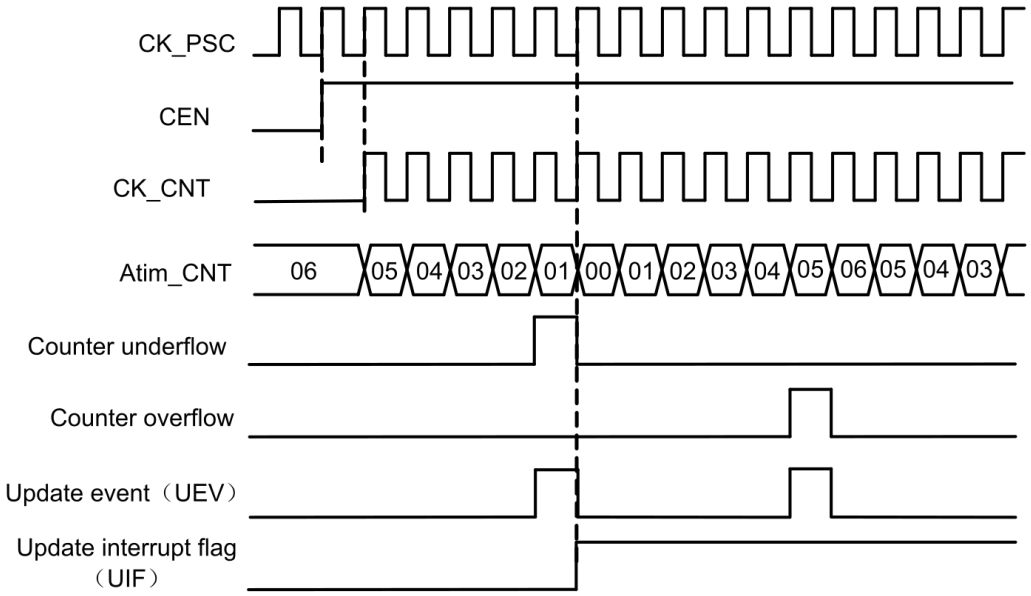


图 20-11 中心对齐计数器时序图，ATIM_PCS=0，ATIM_ARR=0x6

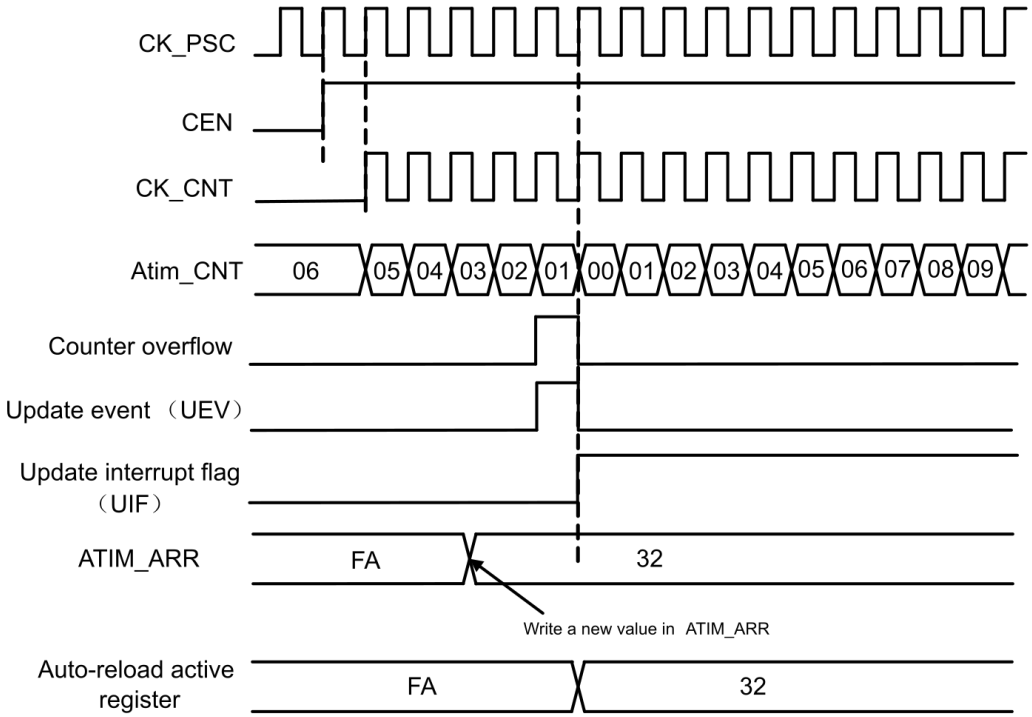


图 20-12 计数器时序图，ATIM_CR1.ARPE=1 时的更新事件(计数器下溢出)

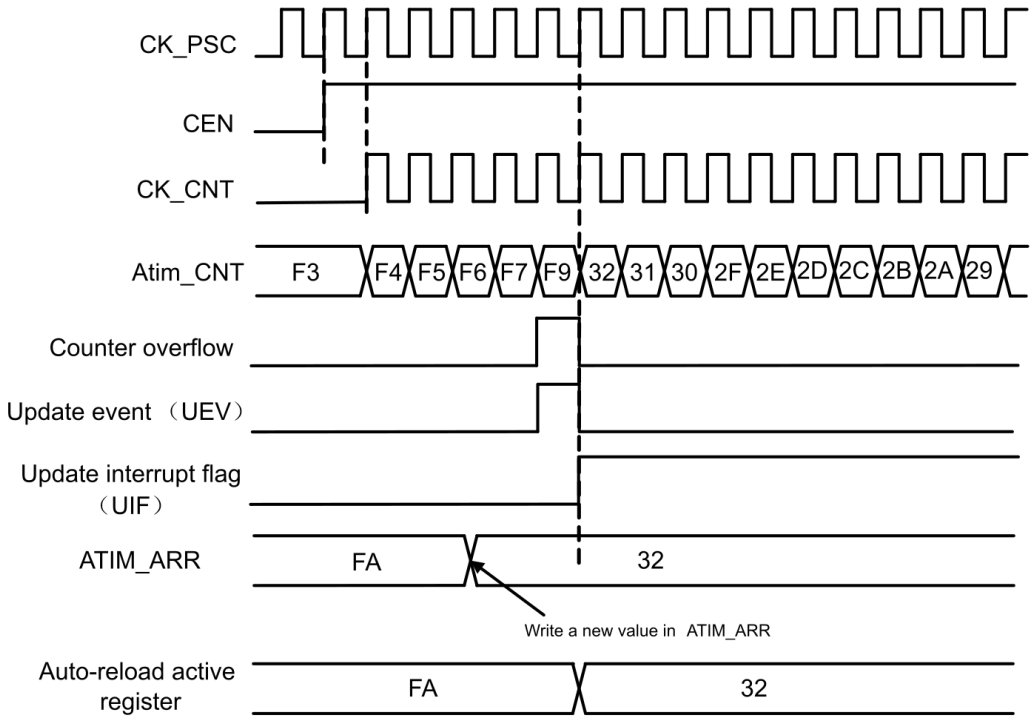


图 20-13 计数器时序图，ATIM_CR1.ARPE=1 时的更新事件(计数器上溢出)

20.5.3 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。

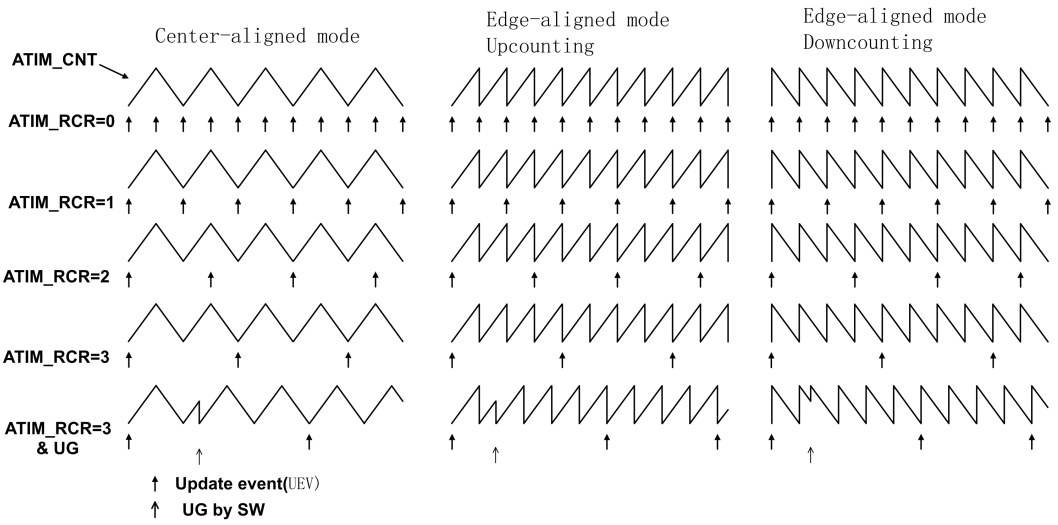


图 20-14 不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

20.5.4 Preload 寄存器

以下功能寄存器支持 preload 功能：

- 自动重载寄存器 ATIM_ARR
- 重复计数寄存器 ATIM_RCR
- 预分频寄存器 ATIM_PSC（不可关闭 preload 功能）
- 通道控制寄存器 ATIM_CCRx
- ATIM_CCER.CcxE 和 ATIM_CCER.CcxNE 控制寄存器
- ATIM_CCMRx.OcxM 控制寄存器

以上寄存器，除了 ATIM_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

20.5.5 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

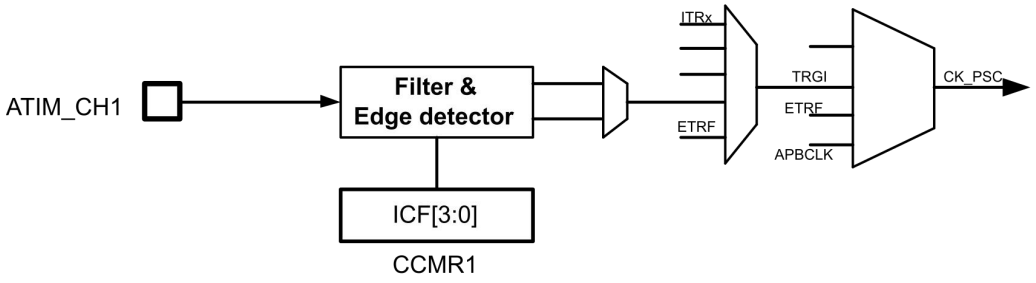


图 20-15 ATIM 时钟源框图

20.5.5.1 内部时钟模式

内部时钟模式下，禁止从机模式（ATIM_SMCR.SMS=000），CEN、DIR、UG 等寄存器位都是软件控制

软件操作 ATIM_EGR.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。

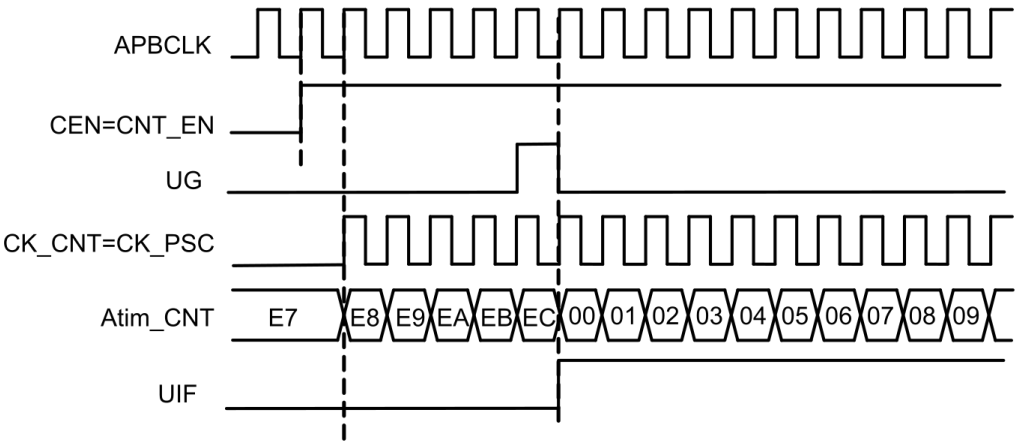


图 20-16 内部时钟源模式，时钟分频因子为 1

20.5.5.2 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 ATIM_SMCR.SMS=111，计数边沿可以配置为上升或下降沿。

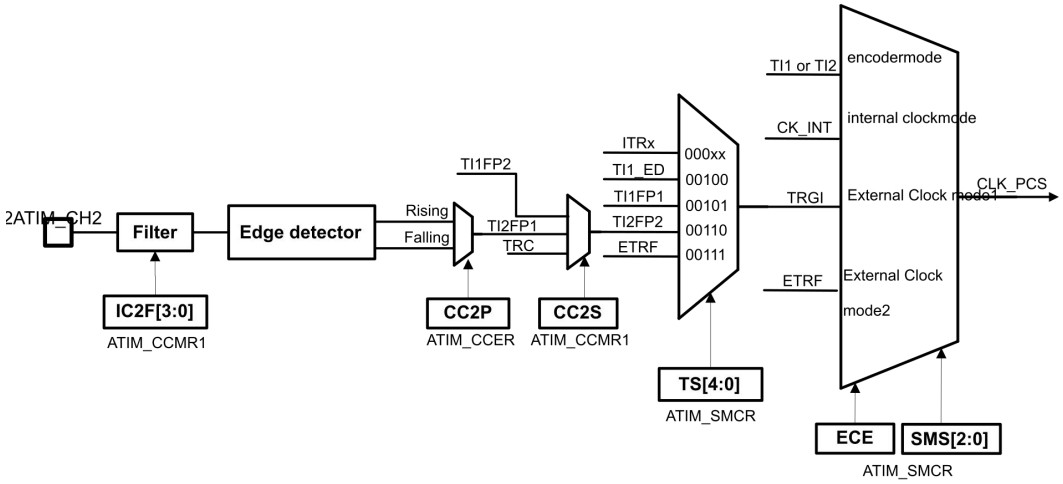


图 20-17 TI2 外部时钟连接例子

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

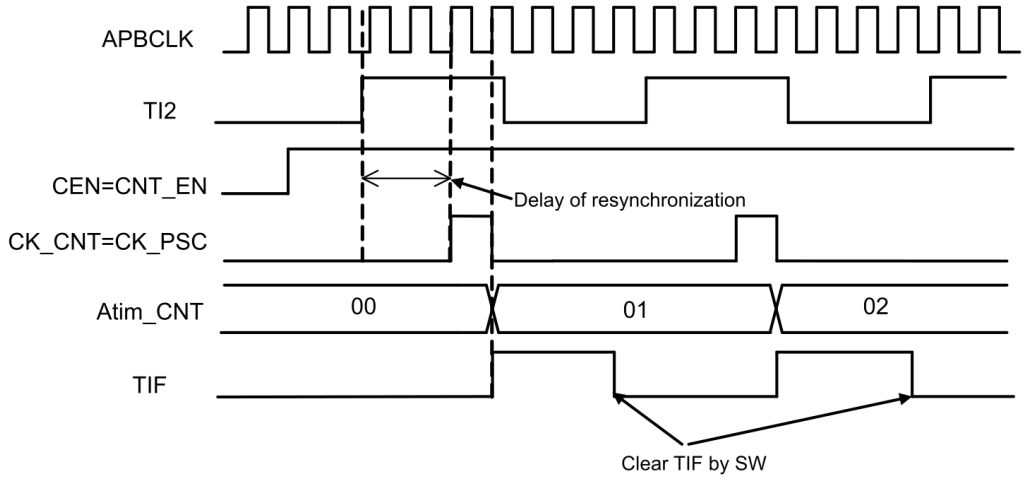


图 20-18 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能 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
- 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：

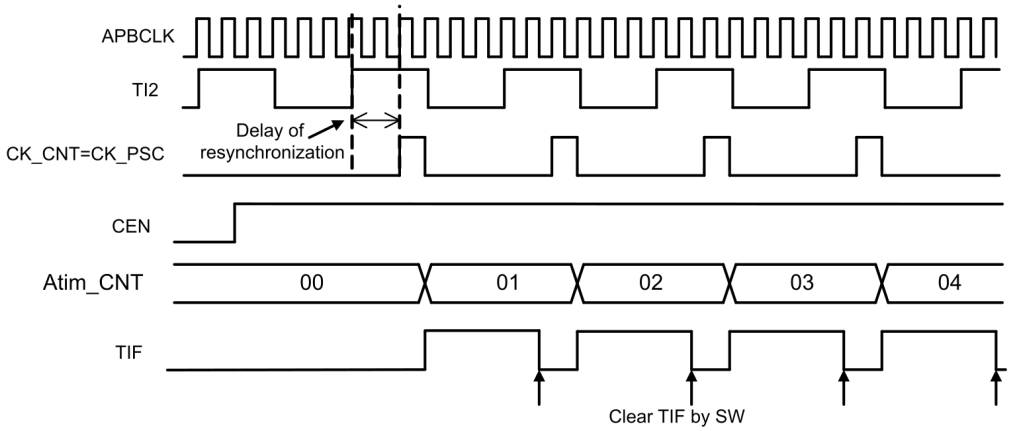


图 20-19 外部时钟模式 1 下的时序

20.5.5.3 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。

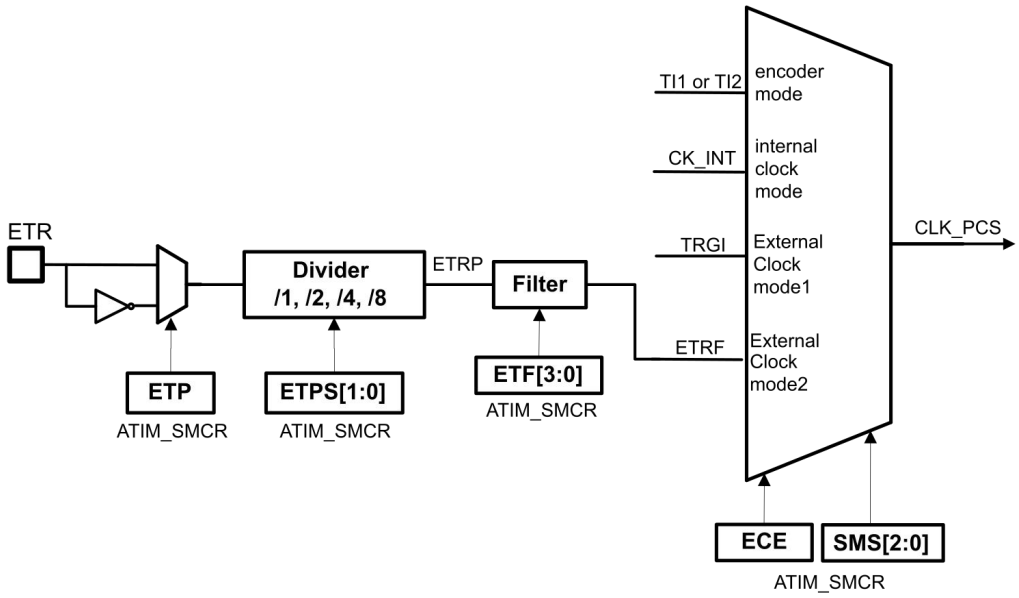


图 20-20 外部触发输入框图

下图是使用 ETR 二分频后的上升沿进行计数，其中实际计数发生时间因为内部时钟的同步过程而延迟于 ETR 输入上升沿。

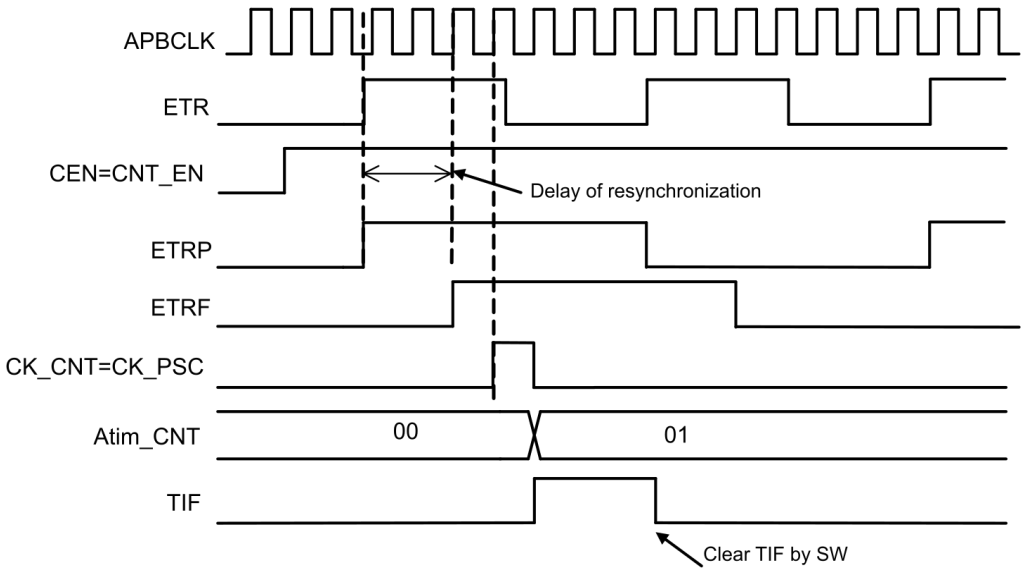


图 20-21 外部时钟模式 2 下的时序 1

与外部时钟模式 1 的主要差别是，ETR 输入直接被分频后再进行滤波，产生 CK_PSC 时钟，这意味着可以支持 ETR 输入频率高于 APB_CLK 的应用场景，这种情况下，需要首先对 ETR 输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在 GPIO 模块中，配置相应管脚为 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：

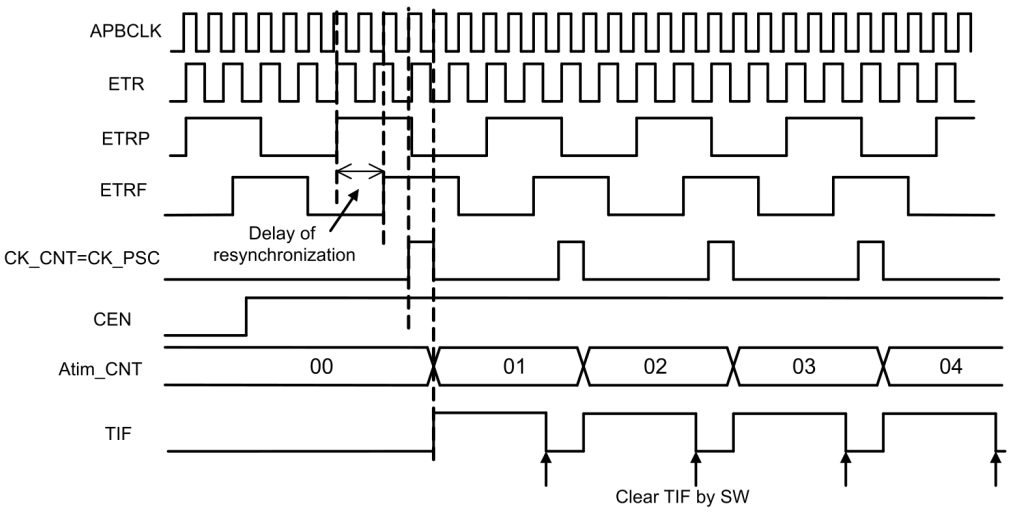


图 20-22 外部时钟模式 2 下的时序 2

在使用外部时钟模式 2 时，仍可以将 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。

20.5.6 内部触发信号（ITRx）

ATIM 支持最大 16 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 配置选择 ITR0~ITR15，并将 ATIM_CCMRx.CCxS 配置为 11，即将 TRC 选为捕捉信号。

每个 ITR 输入支持 16 个内部信号扩展，由 ATIM_SMCR.TS 寄存器配置。输入信号源参考”管脚和内部信号连接”章节。

20.5.7 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。

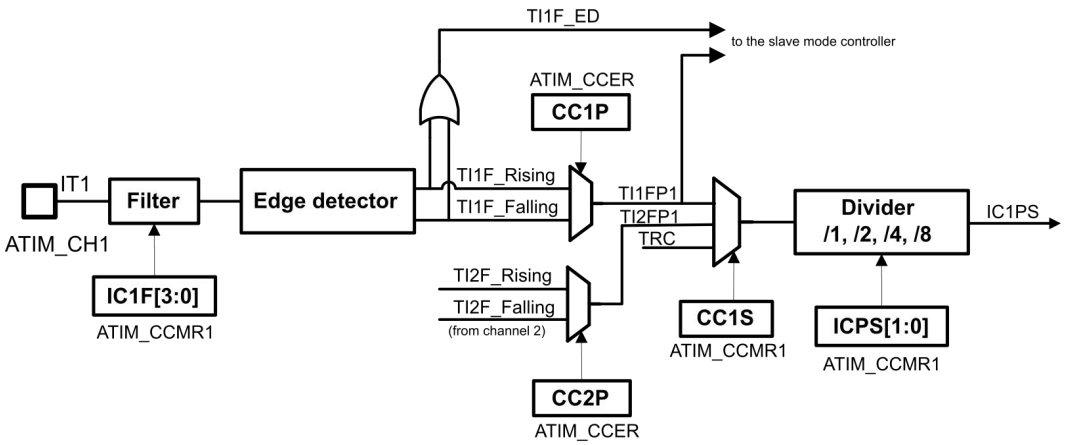


图 20-23 捕获/比较通道(通道 1 输入部分)

输出级电路会产生一个输出基准信号 OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道 1~4 支持互补输出和死区插入。

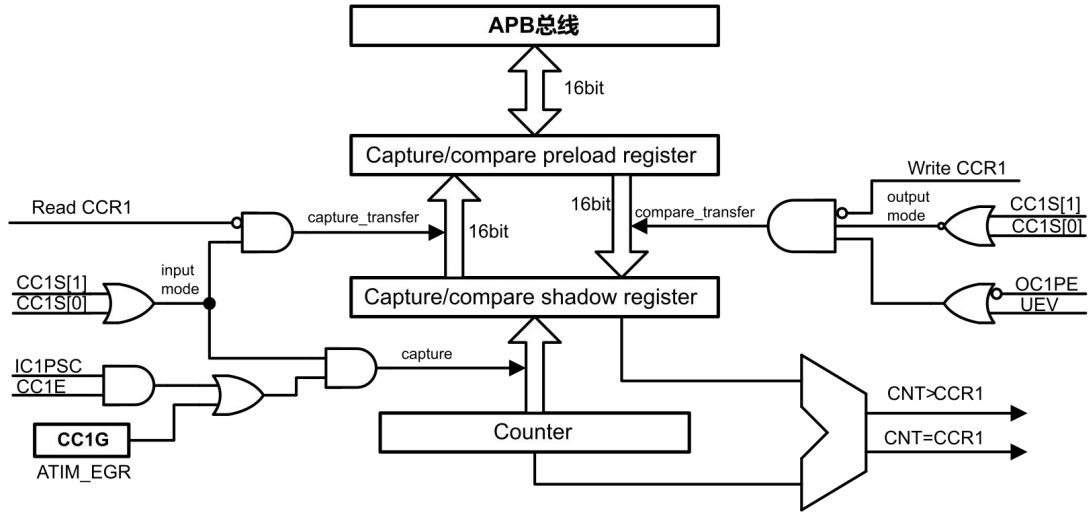


图 20-24 捕获/比较通道 1 的主电路

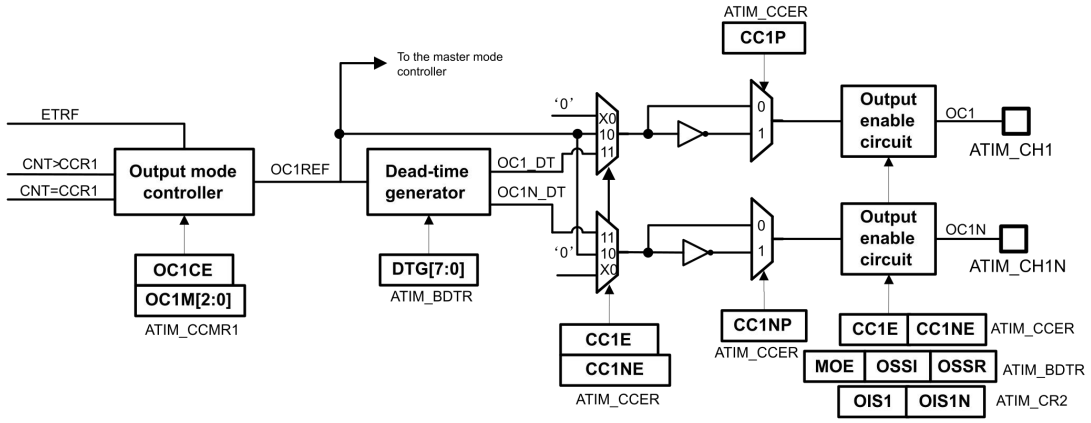


图 20-25 捕获/比较通道的输出部分(通道 1 至 4)

捕捉/比较寄存器（ATIM_CCRx）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。

20.5.8 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 和 ATIM_CCER.CC1NP，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

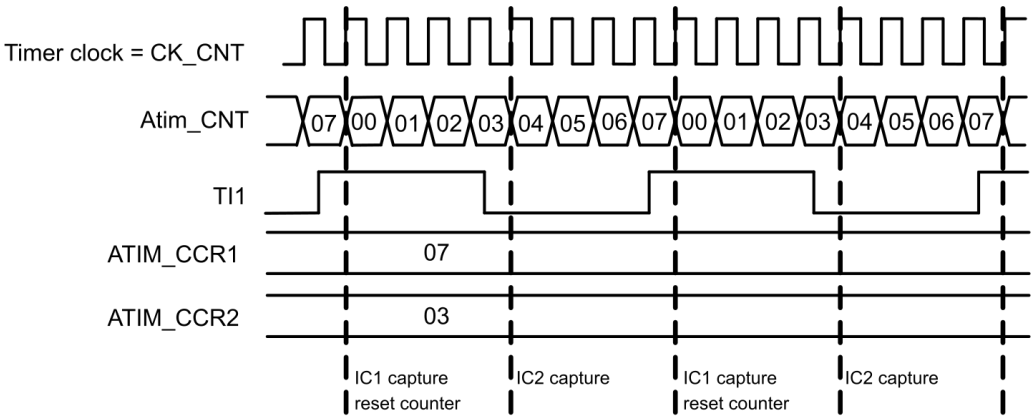


图 20-26 PWM 输入捕获模式时序

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.CC1NP=0；ATIM_CCER.CC2P=1，ATIM_CCER.CC2NP=0
- 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

20.5.9 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。

20.5.10 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

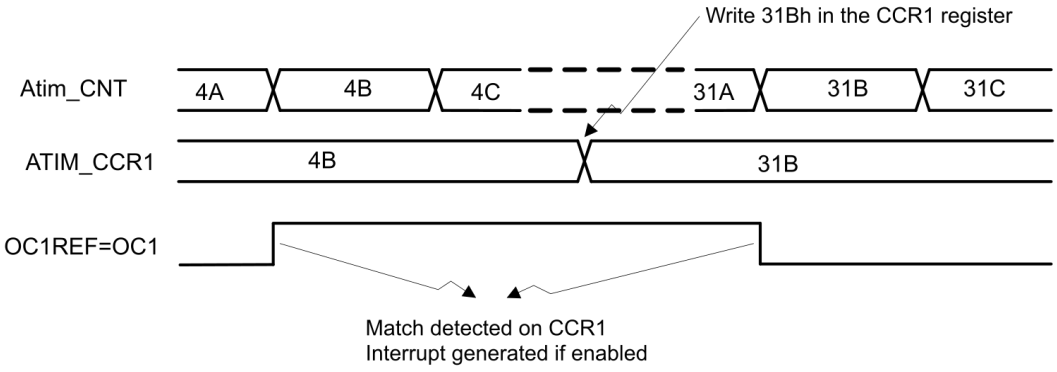


图 20-27 输出比较模式，翻转 OC1

在不使能 preload 的情况下，软件可以随时改写 CCR 寄存器实现对输出波形的实时控制。如果使能了 preload，则 CCR shadow 寄存器仅在下一次 update event 发生时更新为 preload 寄存器的内容。

20.5.11 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 ATIM_CR1.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

20.5.11.1 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。

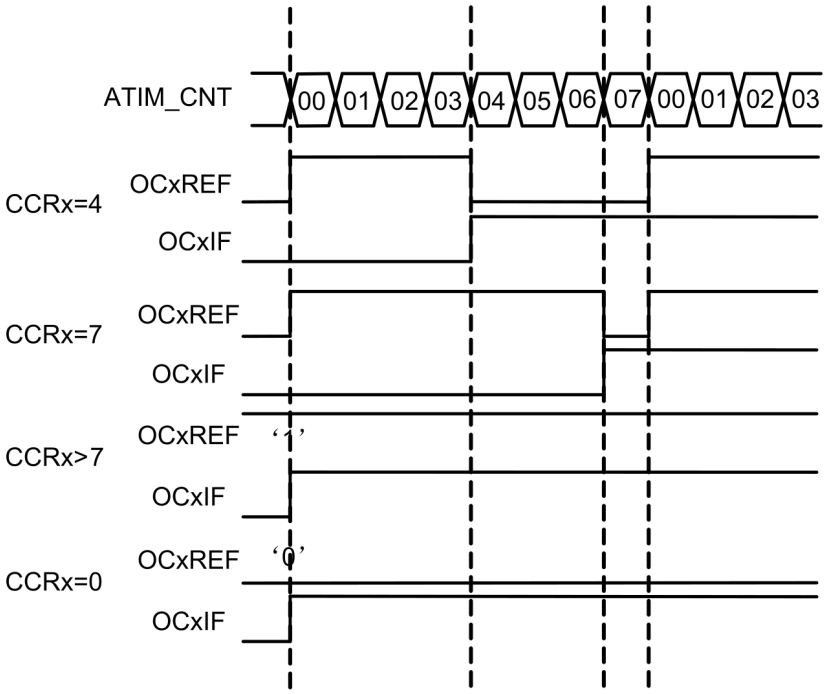


图 20-28 边沿对齐的 PWM 波形(ARR=7, PWM mode 1)

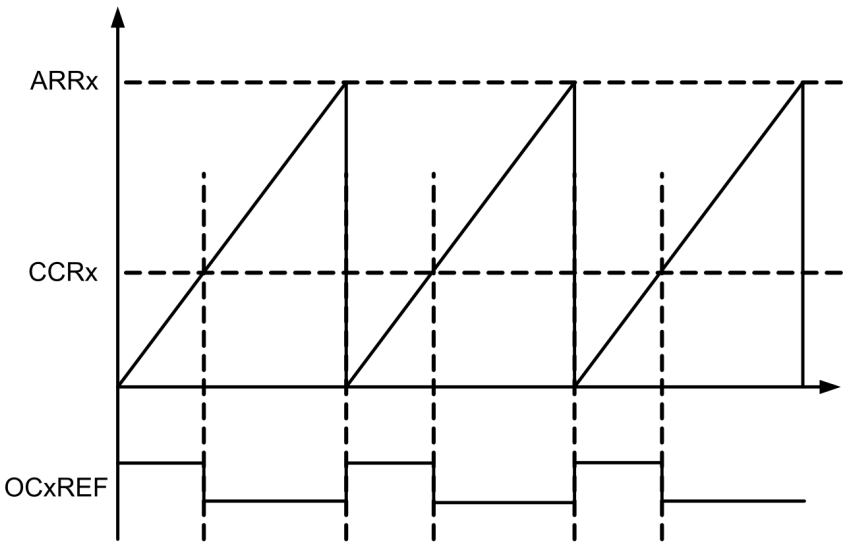


图 20-29 边沿对齐的 PWM 波形(PWM mode 1)

在向下计数时，OCxREF 电平高低定义与向上计数时相同。

20.5.11.2 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。ATIM_CR1.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 ATIM_CR1.CMS 寄存器配置

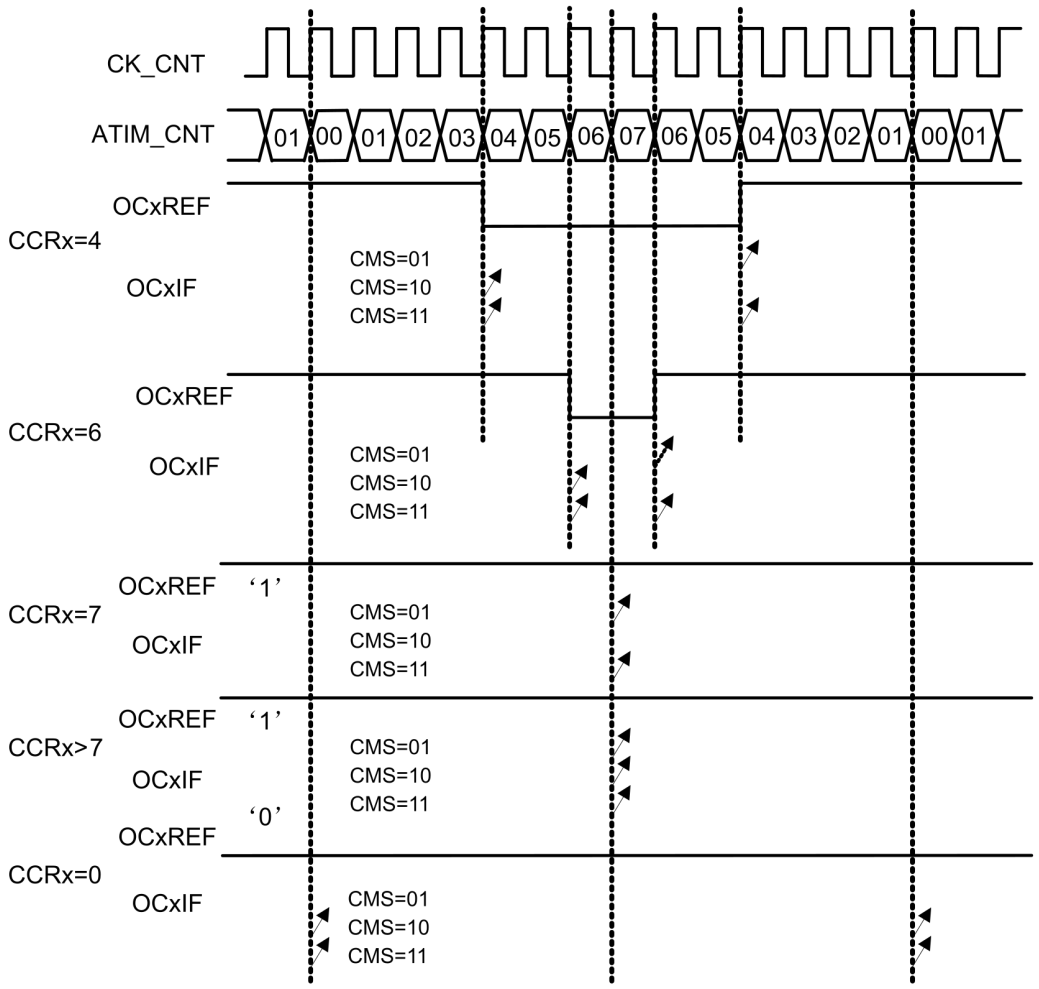


图 20-30 中央对齐的 PWM 波形(APR=7, PWM mode 1)

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 ATIM_EGR.UG 寄存器做一次 update，并且在计数过程中不要改写计数器。

20.5.12 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，OC1REF(OC2REF)由 ATIM_CCR1 和 ATIM_CCR2 寄存器控制，OC3REF(OC4REF)由 ATIM_CCR3 和 ATIM_CCR4 寄存器控制。ATIM_CCMRx.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，OC3REF 为 PWM mode 2

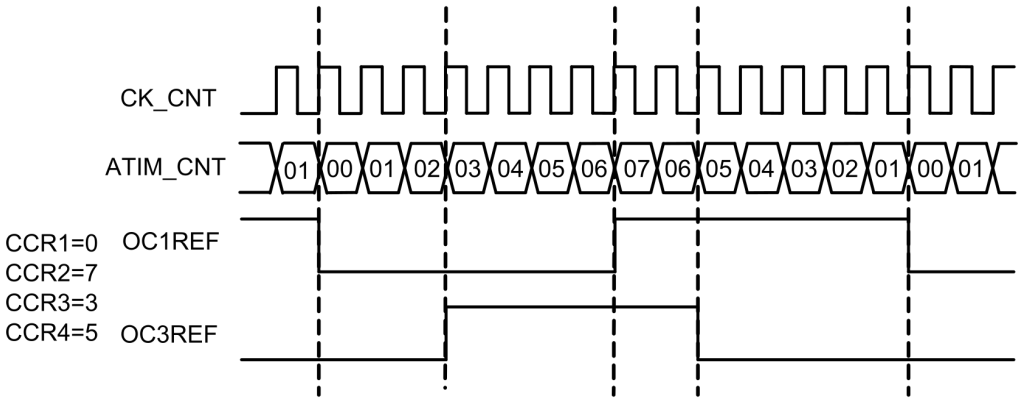


图 20-31 非对称 PWM 输出

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。

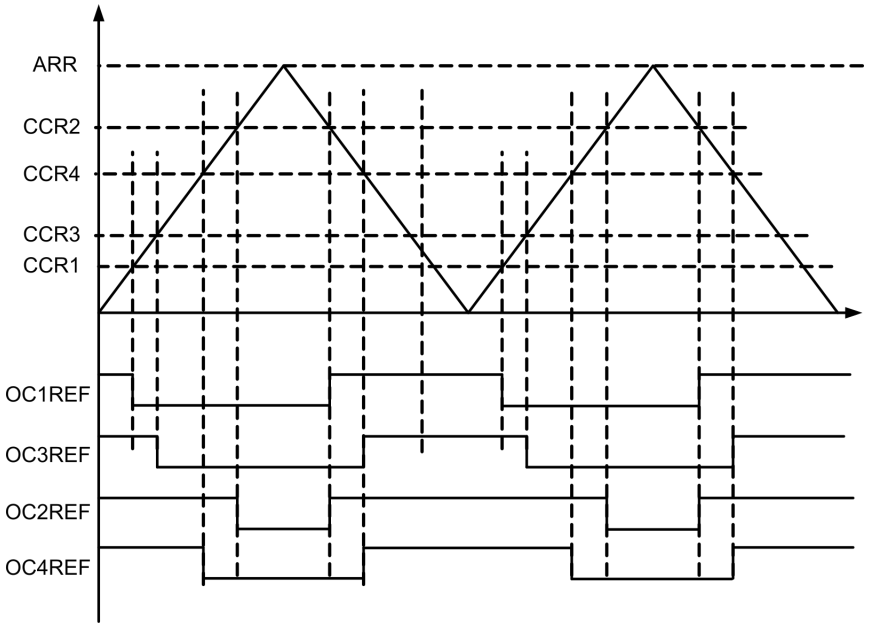


图 20-32 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 1）

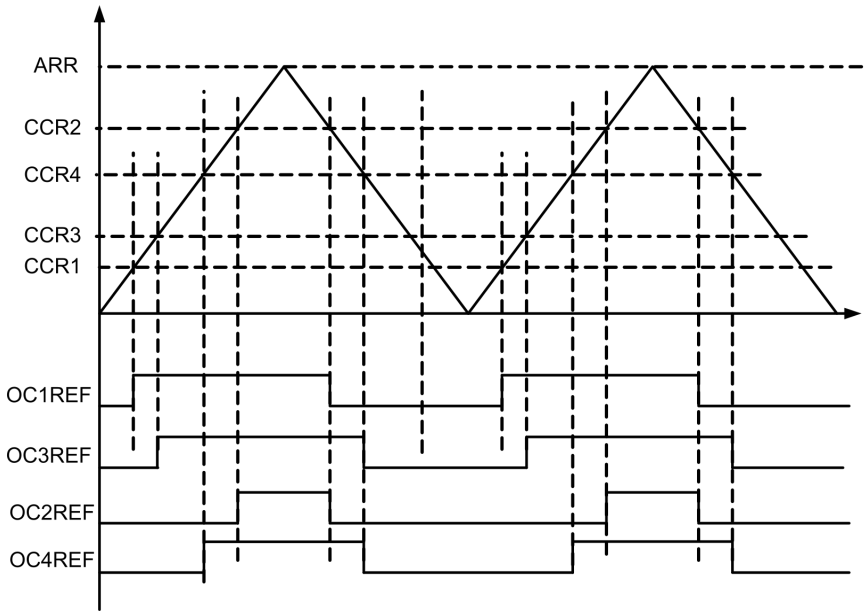


图 20-33 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 2）

20.5.13 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

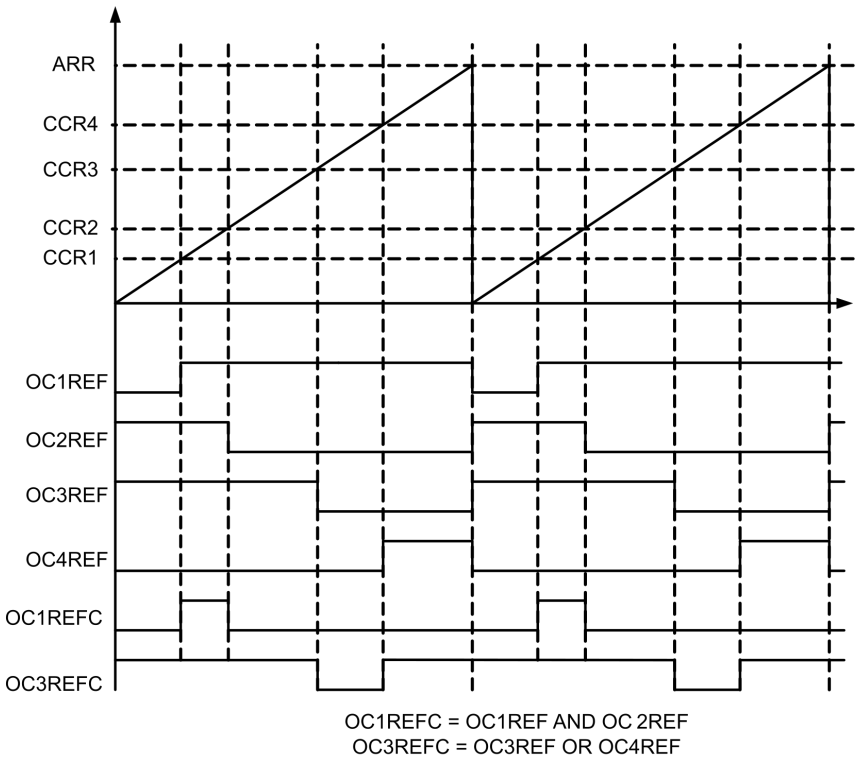


图 20-34 通道 1/3 组合 PWM 输出

20.5.14 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$_{3\text{ 路三相 PWM 信号。}}$ GC5C[3:1]寄存器用于选择 OC5REF 作用于 OC1REFOC3REF 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。

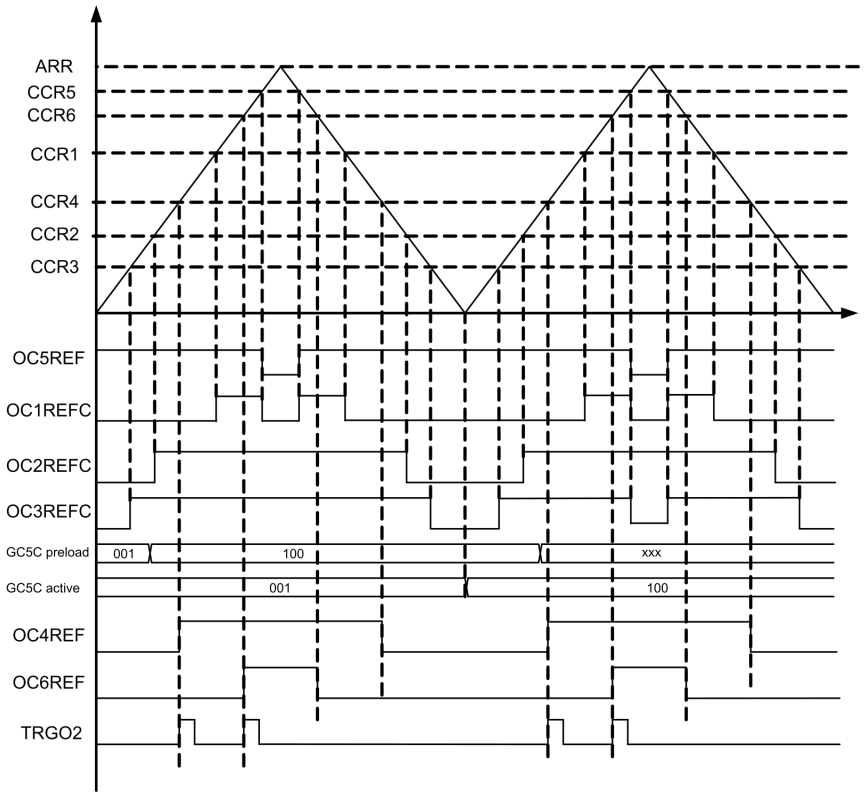


图 20-35 组合三相 PWM 输出

20.5.15 互补输出和死区插入

ATIM 的通道 1~4 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）。CCxp 和 CCxNP 为 0 时：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。

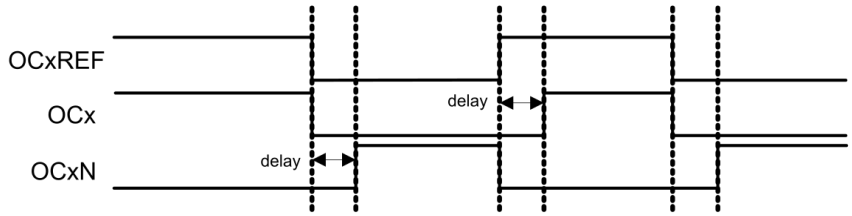


图 20-36 带死区插入的互补输出

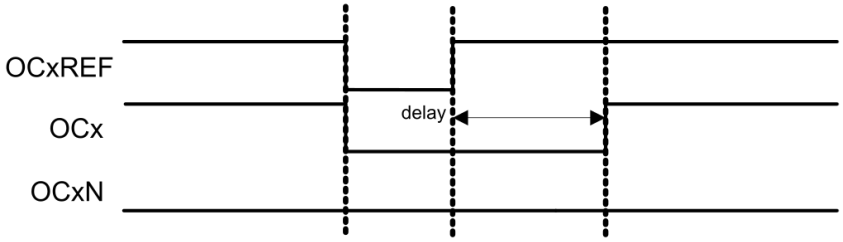


图 20-37 死区波形延迟大于负脉冲

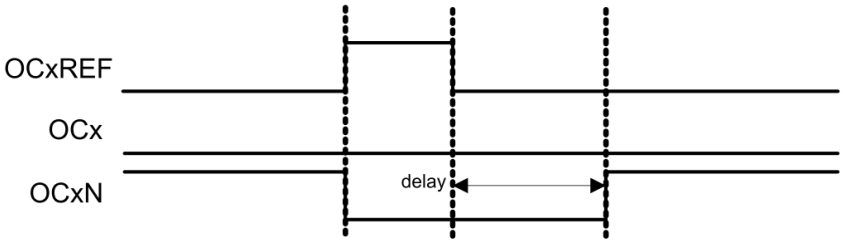


图 20-38 死区波形延迟大于正脉冲

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。

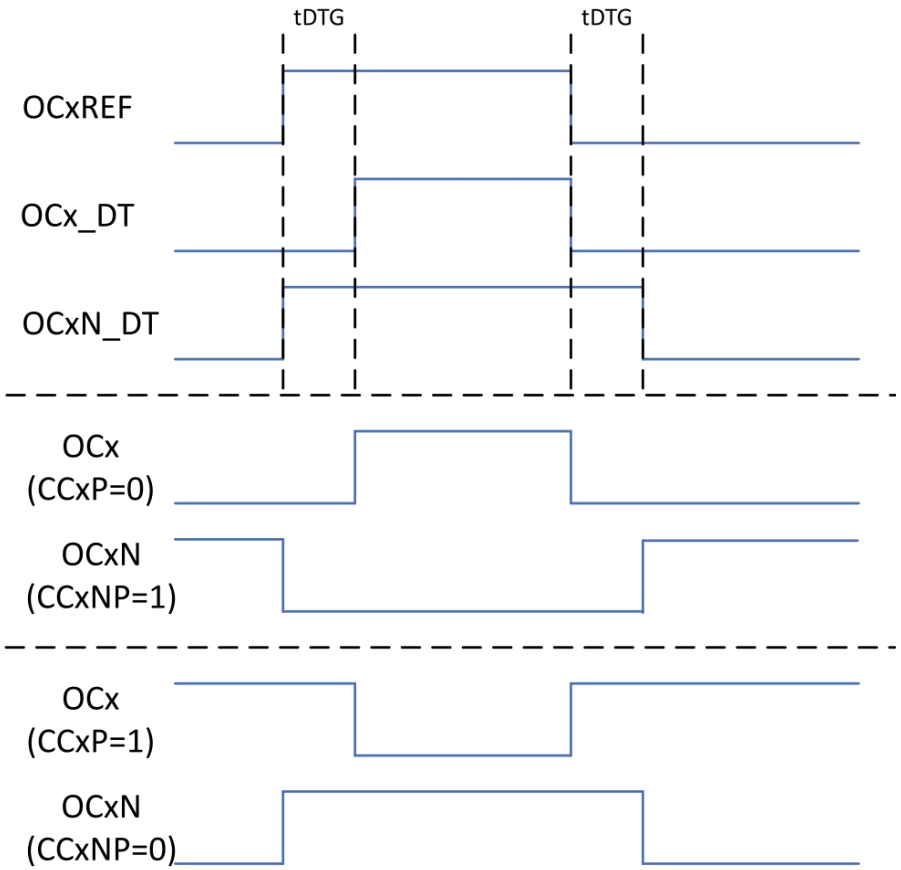


图 20-39 死区插入后的互补输出波形与 CCxP 和 CCxNP 的关系

20.5.16 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。刹车信号可以配置有效极性，以及数字滤波。

20.5.16.1 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（RUN mode）：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：

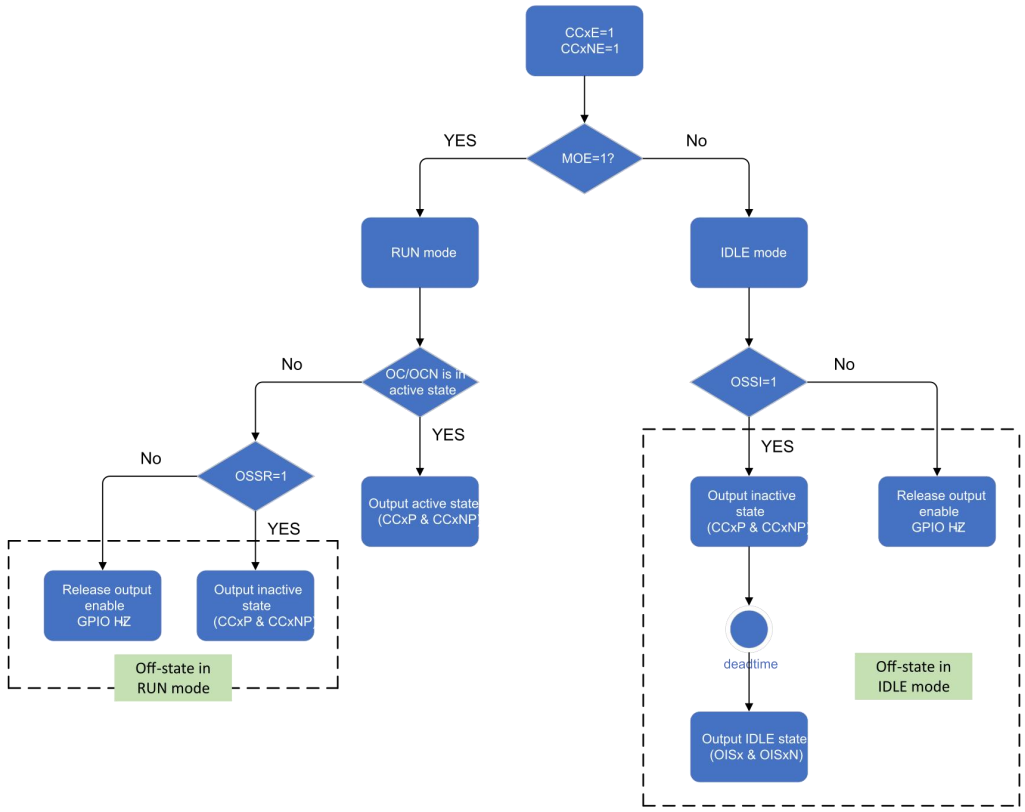


图 20-40 模式与输出状态术语

20.5.16.2 刹车逻辑

刹车输入控制逻辑如下图所示：

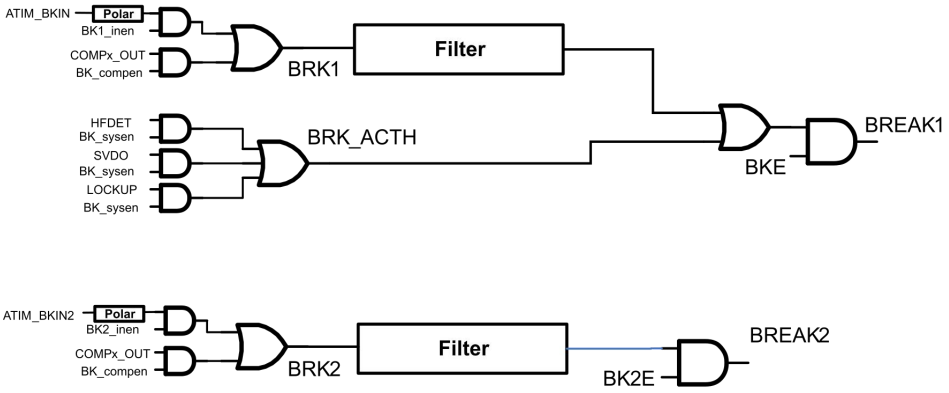


图 20-41 刹车输入结构

注意，刹车信号默认都是高电平有效，所以 SVD、HFDET 等报警信号输出低有效，取反后送给了高级定时器。BKIN 和 BKIN2 管脚输入信号支持极性选择，COMPx_OUT 送到 BRK1 之前也可以选择是否取反。

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
- 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。

20.5.16.3 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
- 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。

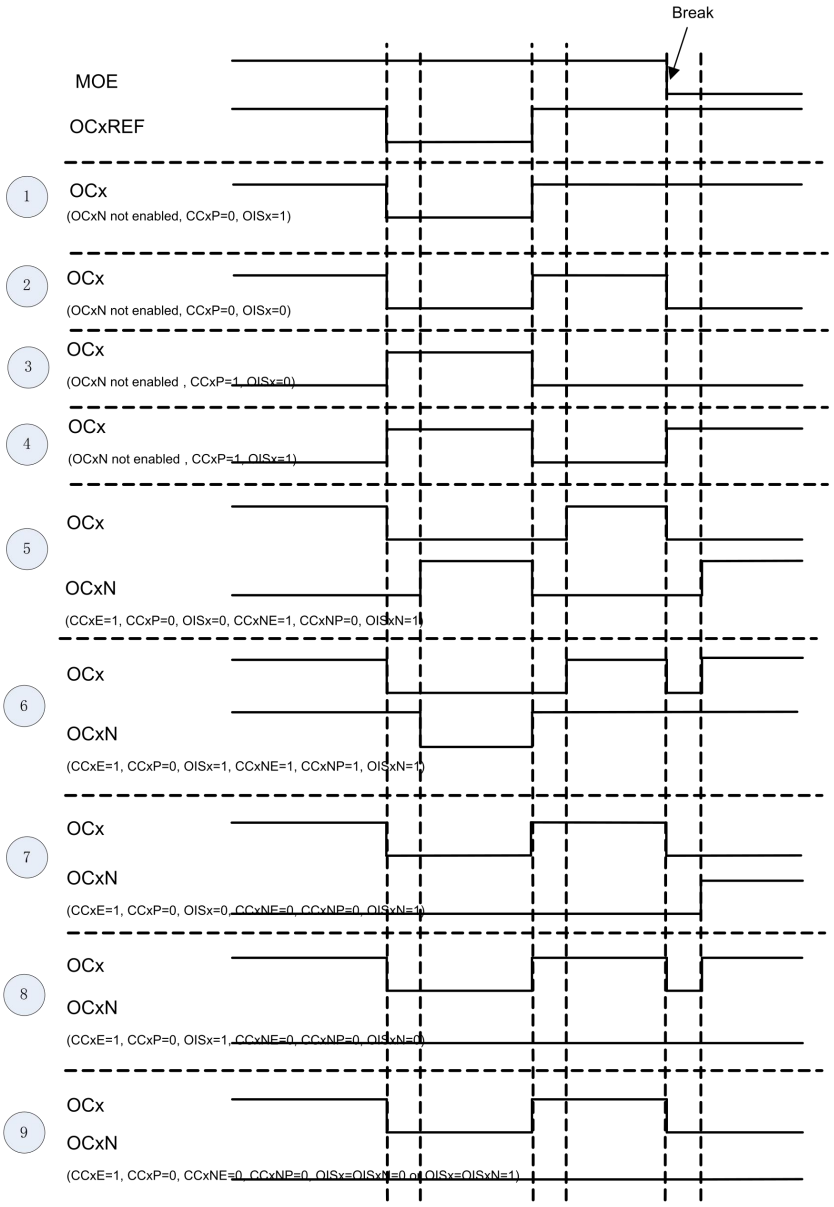


图 20-42 响应刹车的输出（OSSI=1）

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。

表 20-6 刹车行为

BREAK1	**BREAK2**	**ATIM 输出状态**	**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

20.5.17 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。

表 20-7 输出状态逻辑表

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） Hi-Z	
		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	Output Disabled (not

控制寄存器					输出状态	
					OCx=OCxREF xor CCxP, OCx_EN=1	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	X	X	输出关闭（不由 ATIM 驱动）	
			0	0	Hi-Z	
	1		0	1	Off-state（输出使能，并且输出 inactive state） 异步置位：OCx=CCxP, OCxN=CCxNP （BRK 或者 BRK2 事件触发） 接下来（仅 BRK 触发事件有效），如果工作时钟存在：经过死区时间后 OCx=OISx, OCxN=OISxN 前提是 OISx 和 OISxN 的配置不会使 OCx 和 OCxN 同时进入 active state） 注意：BRK2 仅能在 OSSR=OSSI=1 的情况下使用	
			1	0		
			1	1		

20.5.18 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。

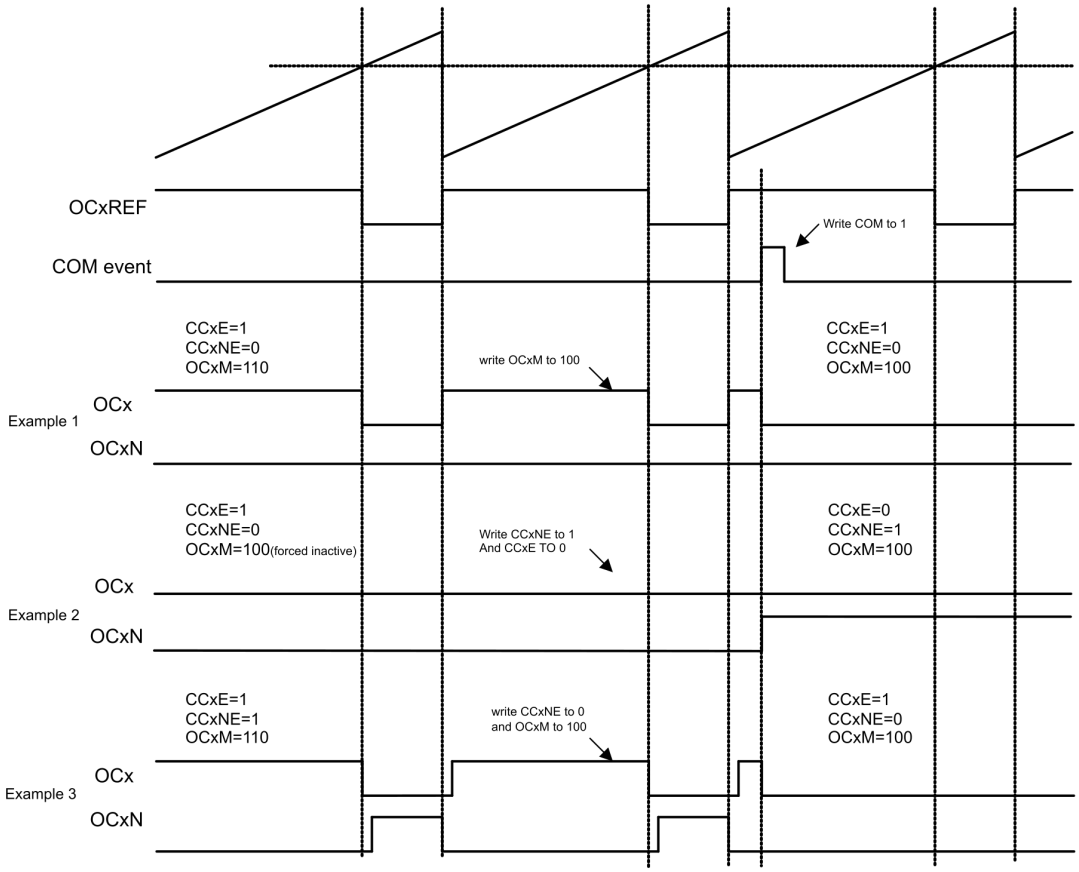


图 20-43 产生六步 PWM，使用 COM 的例子(OSSR=1)

20.5.19 单脉冲输出（one-pulse mode）

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

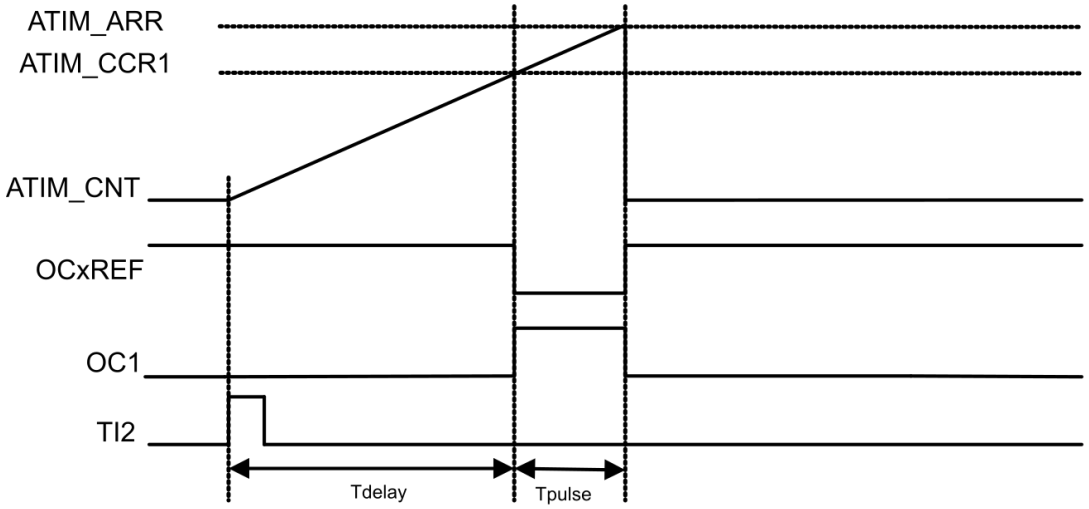


图 20-44 单脉冲模式的例子

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

20.5.20 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 ATIM_SMCR.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。

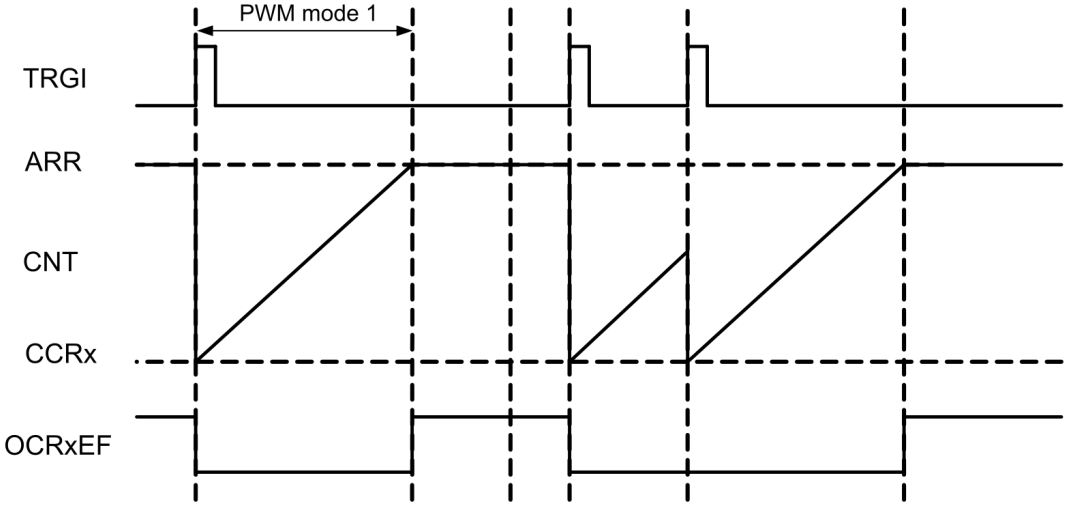


图 20-45 重触发 OPM 模式 1，向上计数，CCR=0

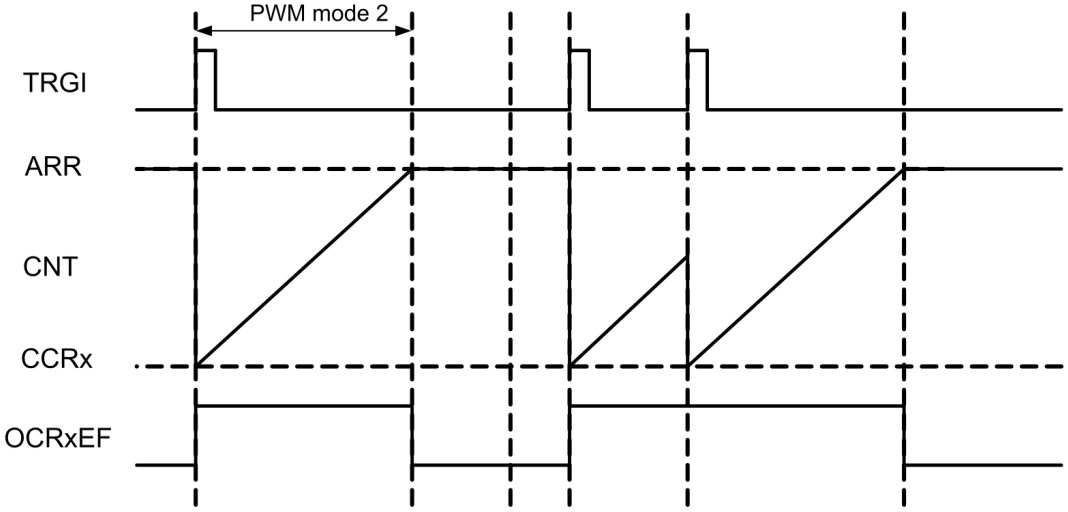


图 20-46 重触发 OPM 模式 2，向上计数，CCR=0

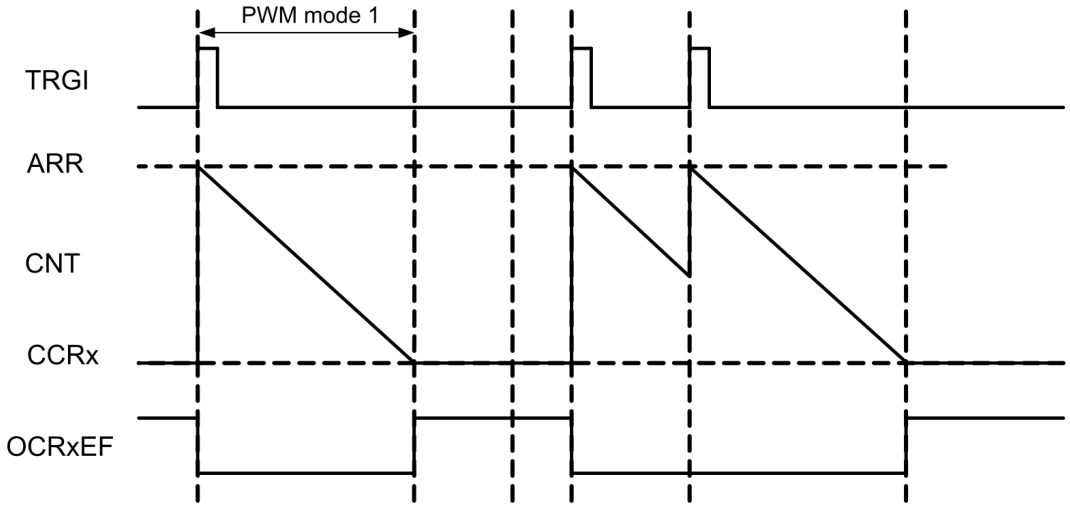


图 20-47 重触发 OPM 模式 1，向下计数，CCR=0

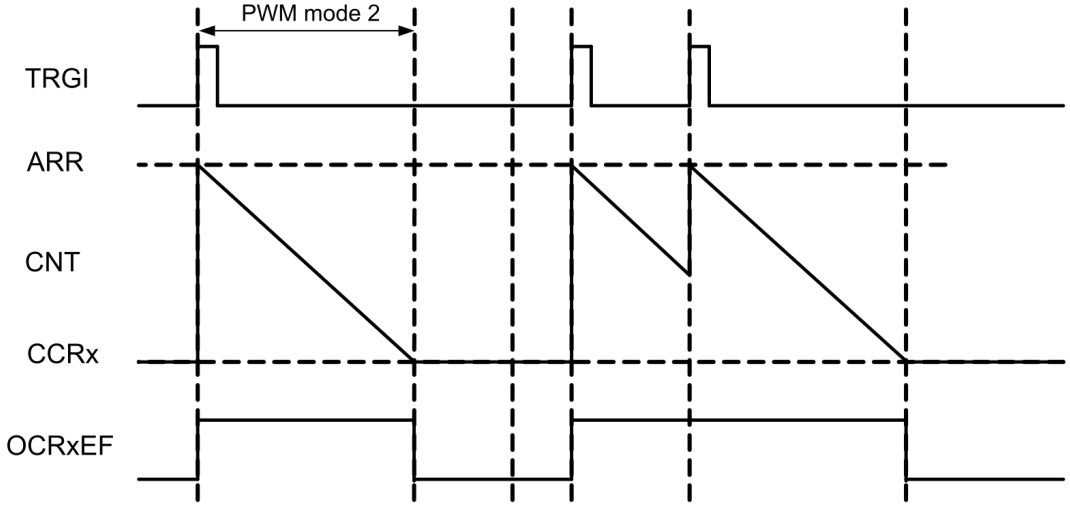


图 20-48 重触发 OPM 模式 2，向下计数，CCR=0

20.5.21 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。

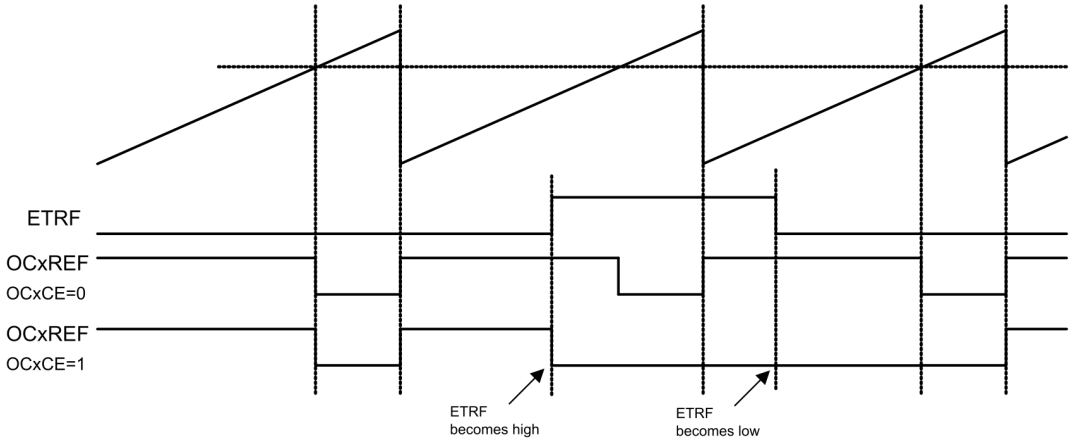


图 20-49 ETR 信号清除 ATIM 的 OCxREF

20.5.22 编码器接口模式（encoder interface）

20.5.22.1 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 ATIM_SMCR.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：

表 20-8 encoder interface 计数方式（CC1P=CC2P=0）

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 处计数 x1 mode	1110	高	不计数	递增	不计数	不计数
		低	不计数	递减	不计数	不计数
仅在 TI2 处计数 x1 mode	1111	高	不计数	不计数	不计数	递减
		低	不计数	不计数	不计数	递增

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。

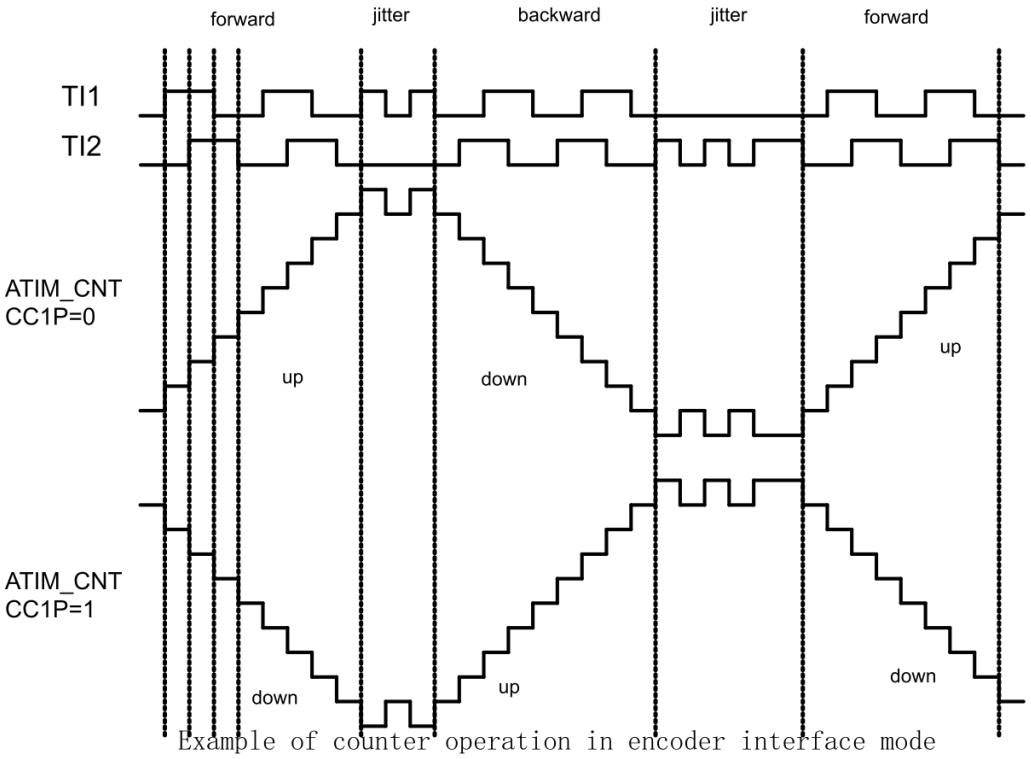


图 20-50 编码器模式下的计数器操作实例

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）。

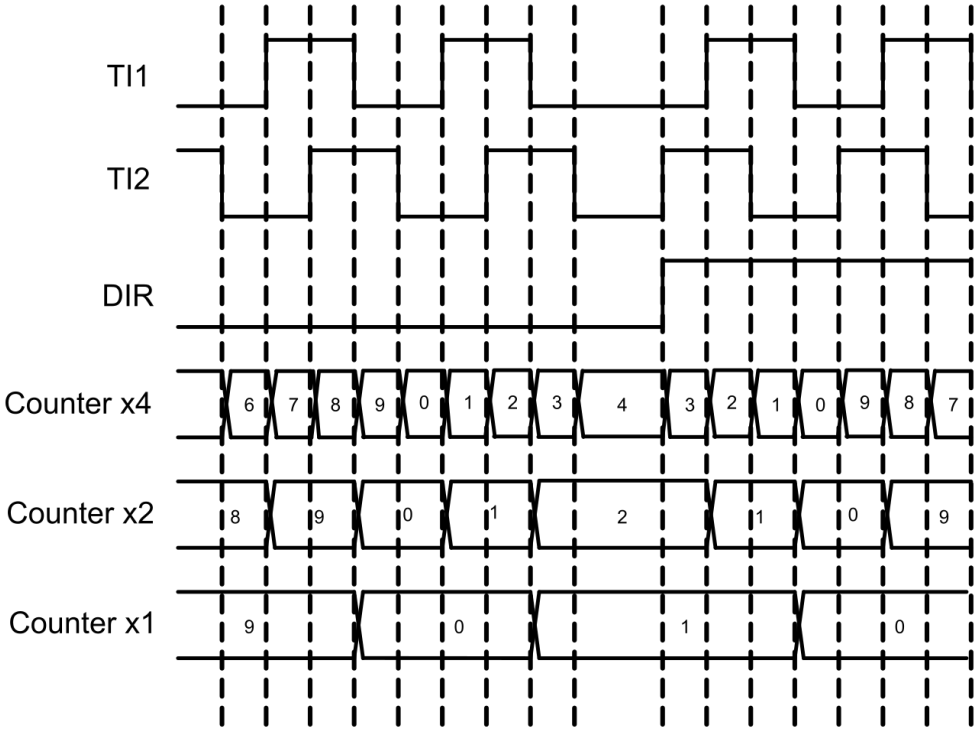


图 20-51 正交编码器不同边沿计数模式示例

当 ATIM_CR1 寄存器中的 UIFREMAP 位置位时，ATIM_ISR.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。

20.5.22.2 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 ATIM_SMCR.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。

- ATIM_SMCR.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
- ATIM_SMCR.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

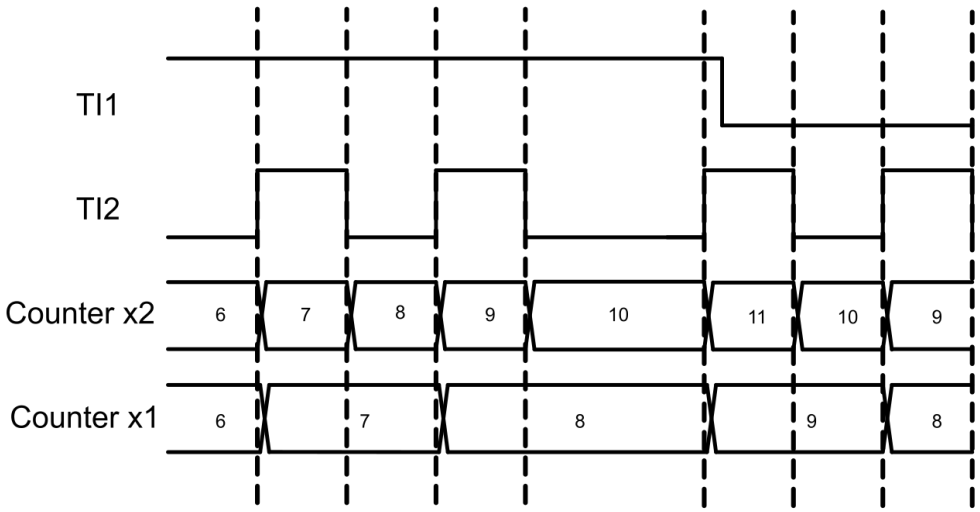


图 20-52 时钟方向编码器示例

20.5.22.3 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。

- ATIM_SMCR.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
- ATIM_SMCR.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）

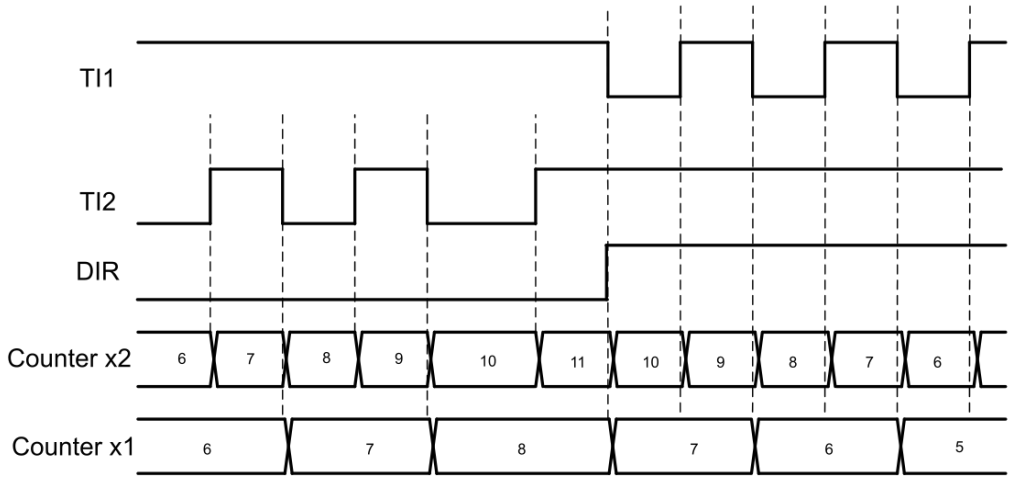


图 20-53 带方向的时钟编码器（CC1P=CC2P=0）

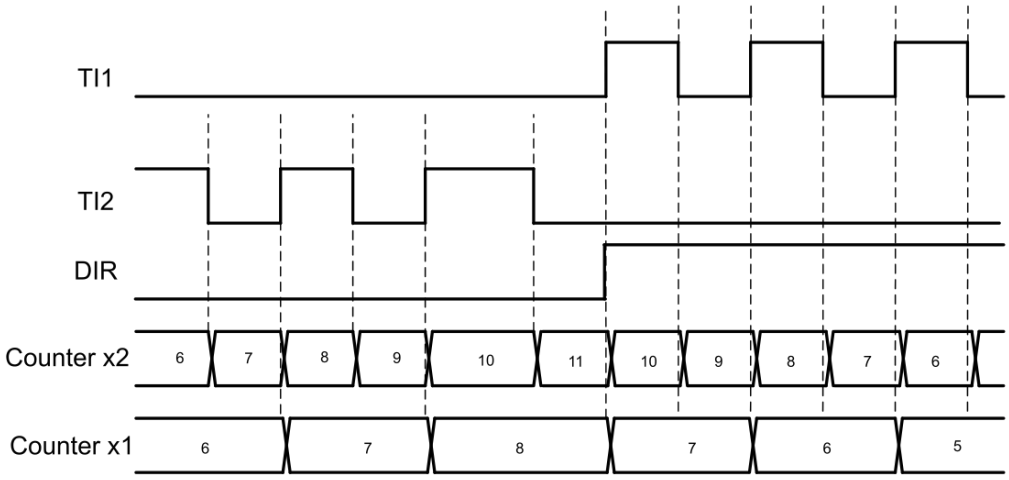


图 20-54 带方向的时钟编码器（CC1P=CC2P=1）

下表是计数方式与两路输入信号之间的关系：

表 20-9 directional clock mode 计数方式

带方向的时钟编码模式	SMS	对应信号的电平（TI1 对应 TI2，TI2 对应 TI1）	TI1FP1		TI2FP2	
			上升	下降	上升	下降
CCxP=0 x2 mode	1100	高	递减	递减	递增	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x2 mode	1100	高	不计数	不计数	不计数	不计数
		低	递减	递减	递增	递增
CCxP=0 x1 mode	1101	高	不计数	递减	不计数	递增
		低	不计数	不计数	不计数	不计数
CCxP=1 x1 mode	1101	高	不计数	不计数	不计数	不计数
		低	递减	不计数	递增	不计数

20.5.22.4 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 ATIM_SMCR.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：

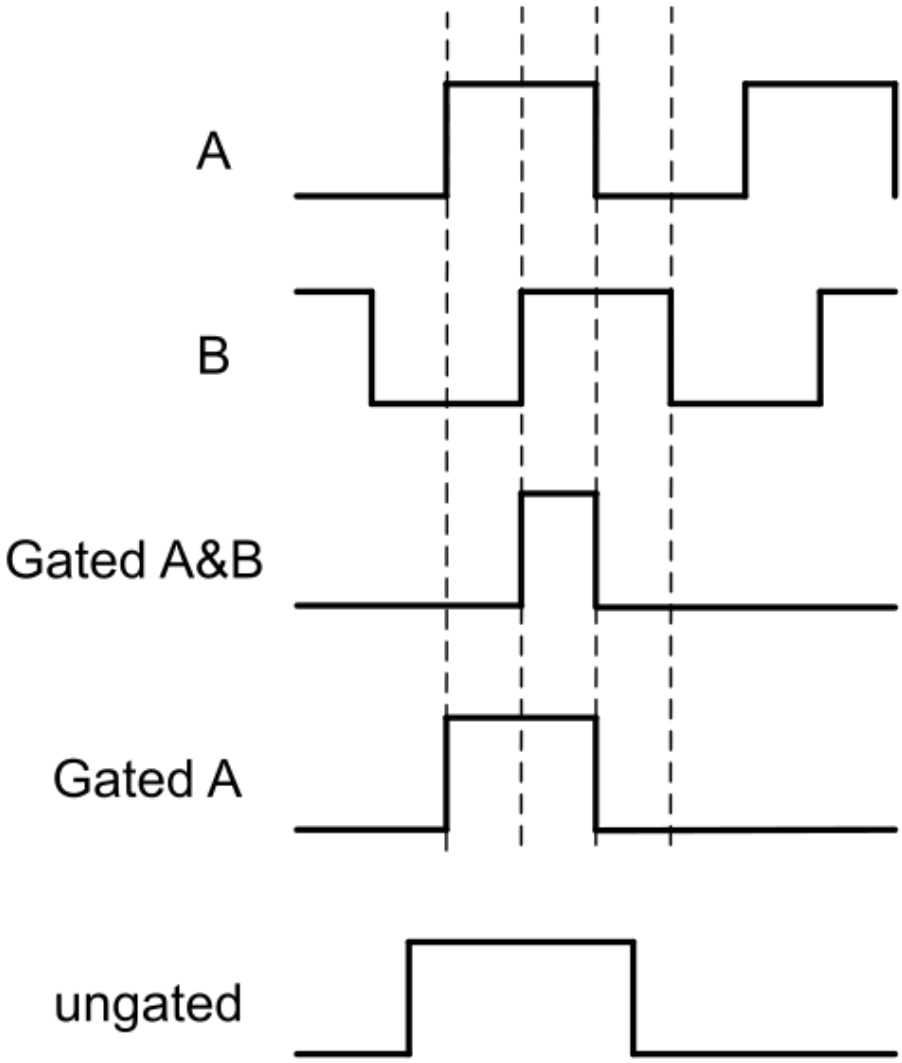


图 20-55 编码器 index 定义

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）。

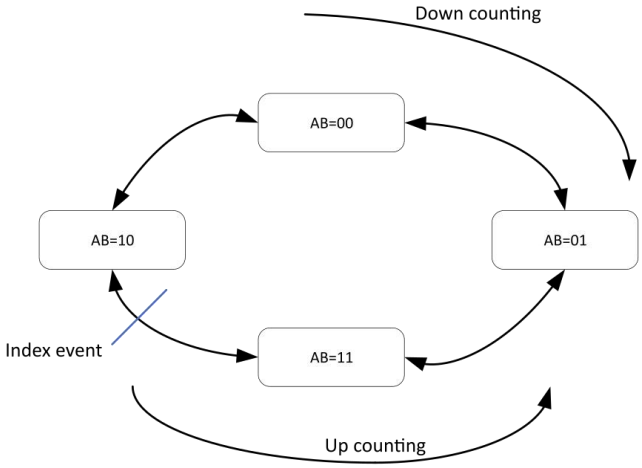


图 20-56 IPOS[1:0]=11 时的 index 识别

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。

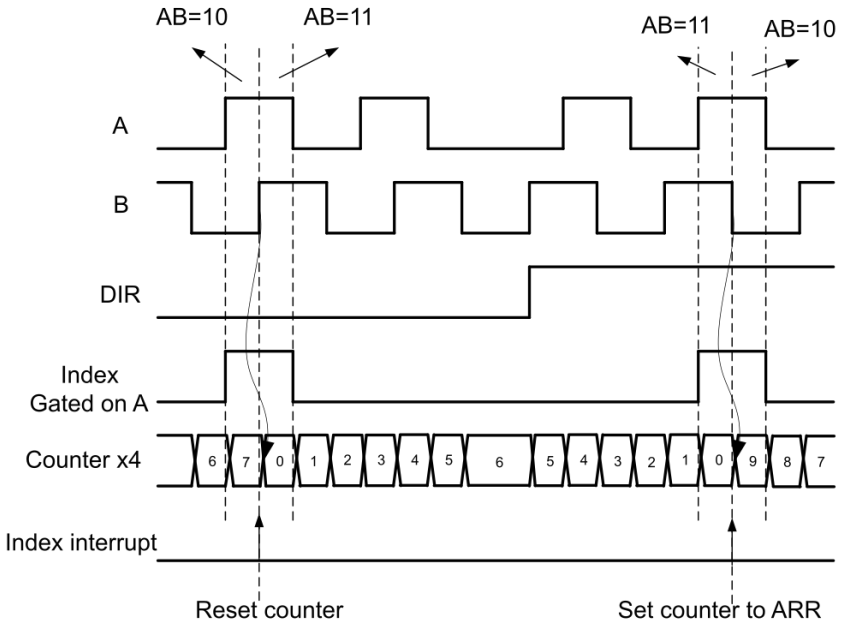


图 20-57 IPOS[1:0]=11, index gated on A, ARR=9

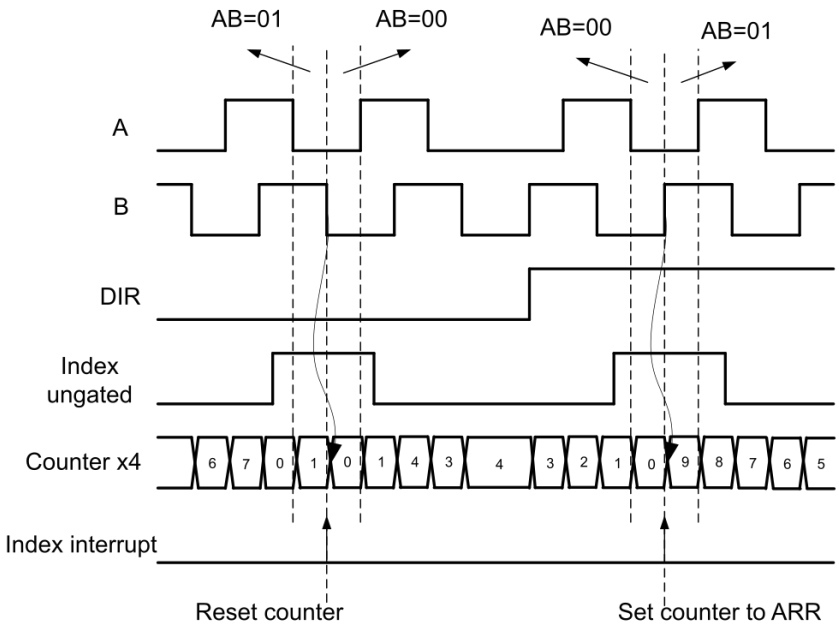


图 20-58 IPOS[1:0]=00, index ungated, ARR=9

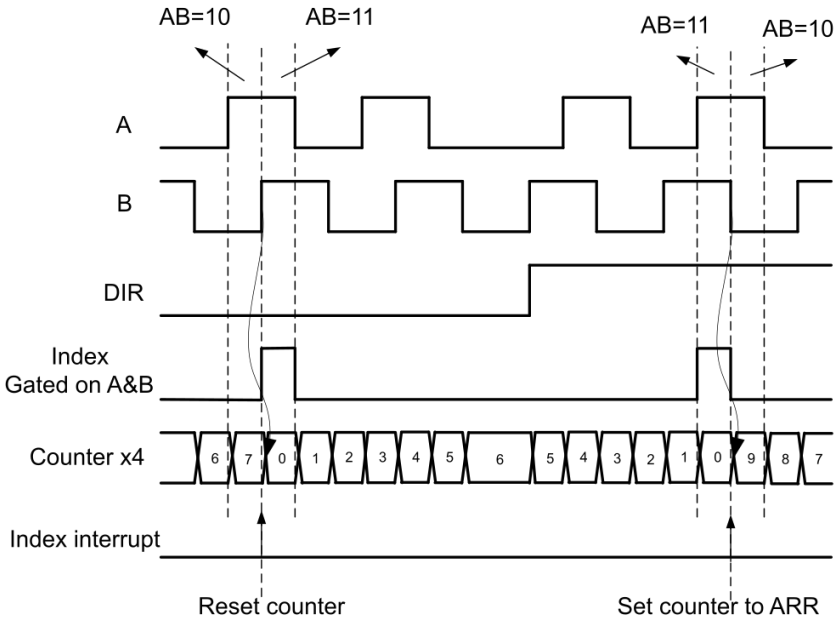


图 20-59 IPOS[1:0]=11, index gated on A&B, ARR=9

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：

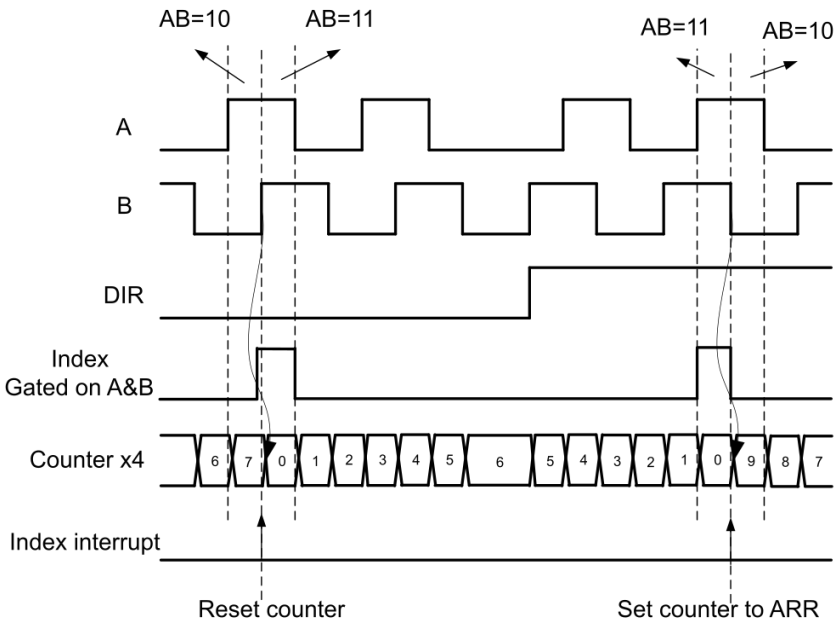


图 20-60 IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B

下图中则是 index 滞后 B 相输入，或者说出现一个窄 Z 相信号：

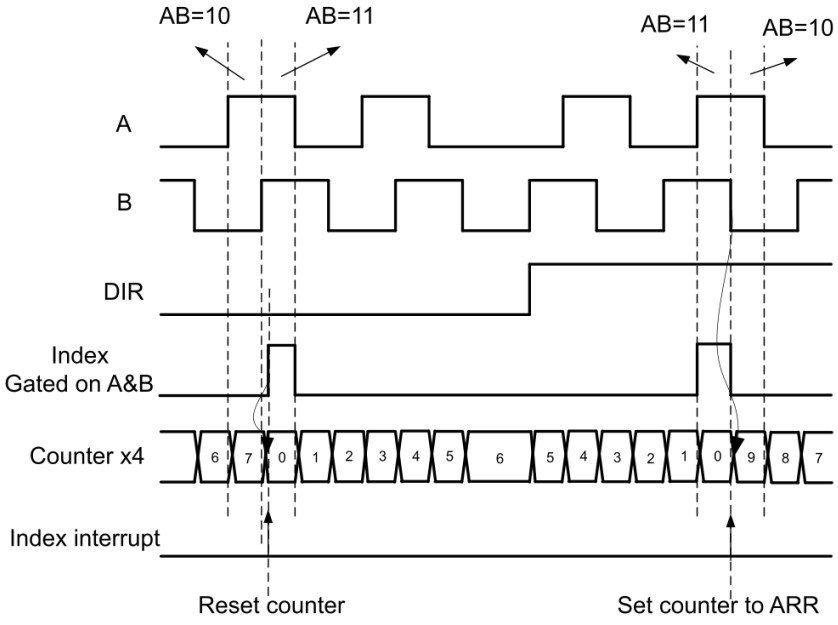


图 20-61 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B

上图中 index lagging 的区域放大图如下：

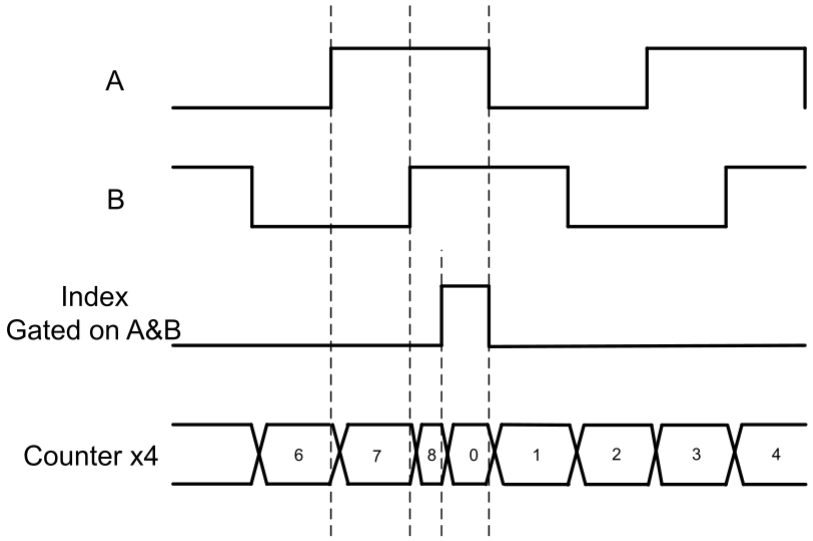


图 20-62 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed

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）：

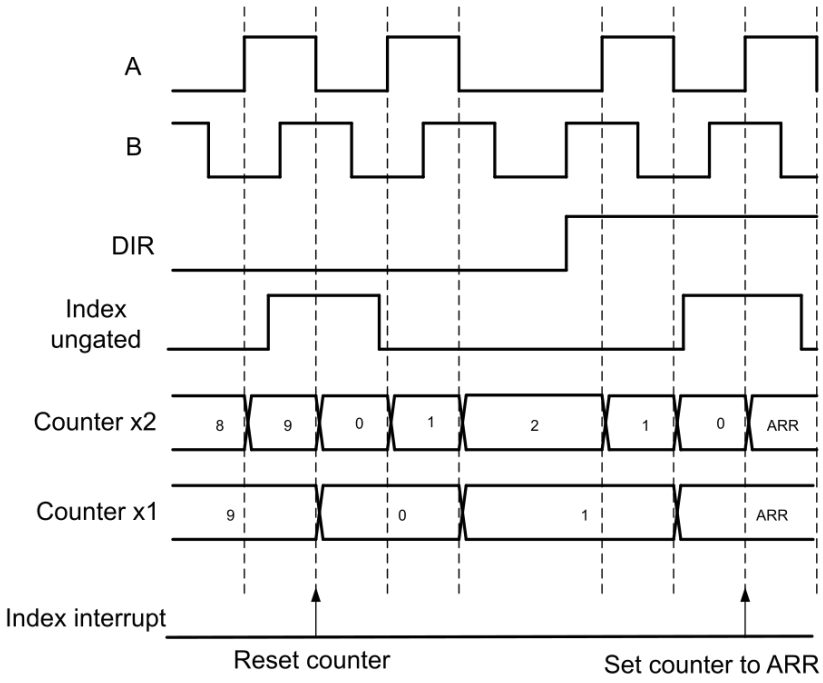


图 20-63 IPOS[1:0]=01, ungated index

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。

20.5.22.5 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。

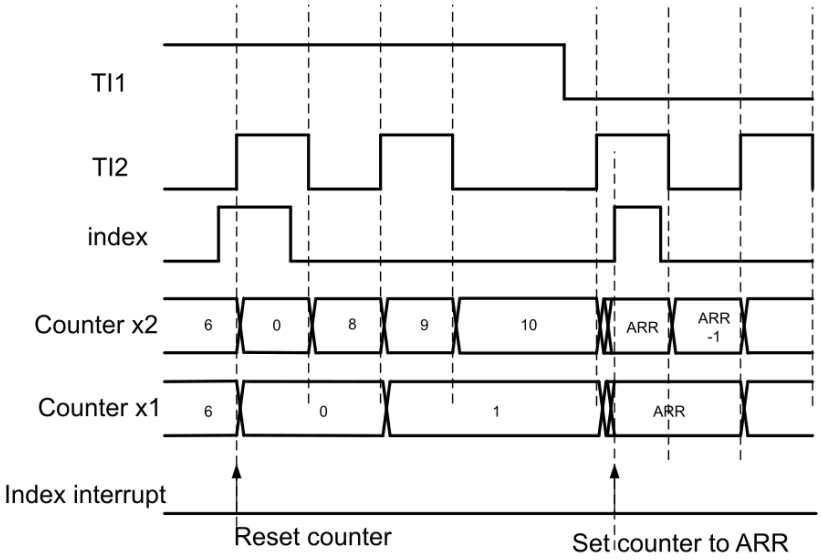


图 20-64 IPOS[0]=1, clock+direction mode

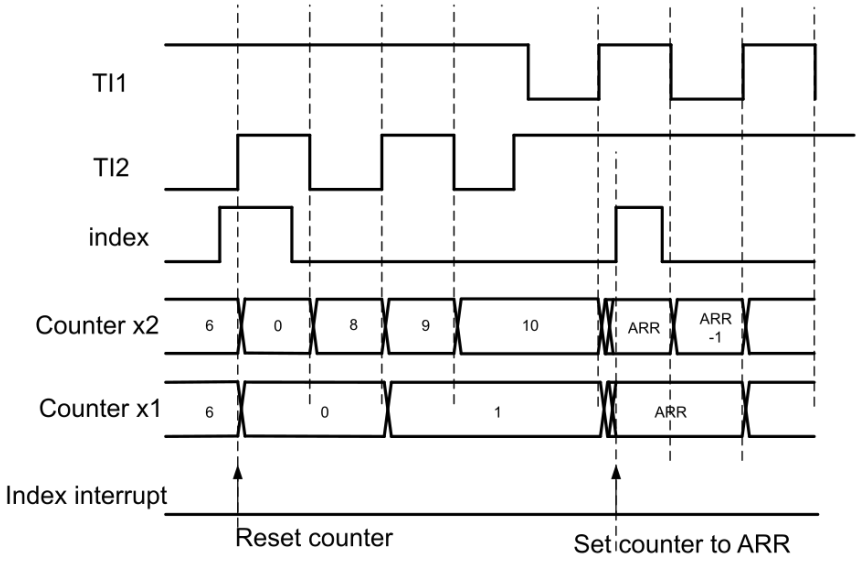


图 20-65 IPOS[0]=1, directional clock mode

20.5.22.6 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。

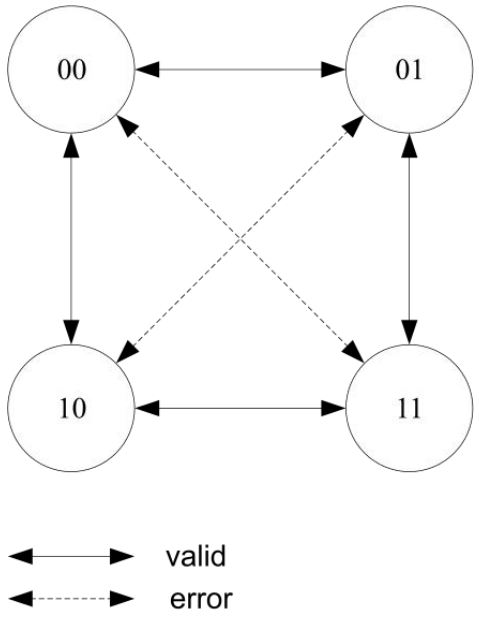


图 20-66 正交编码状态转移

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。

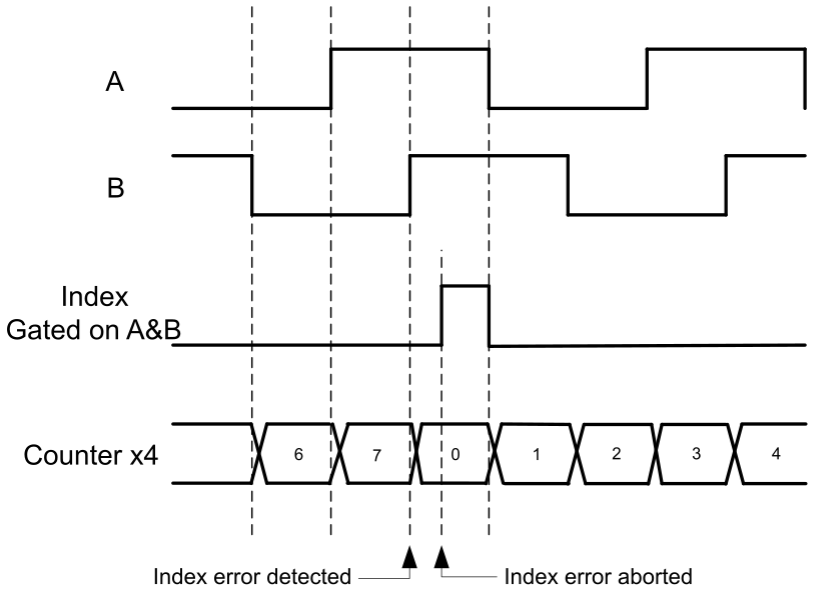


图 20-67 index 错误撤销

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。

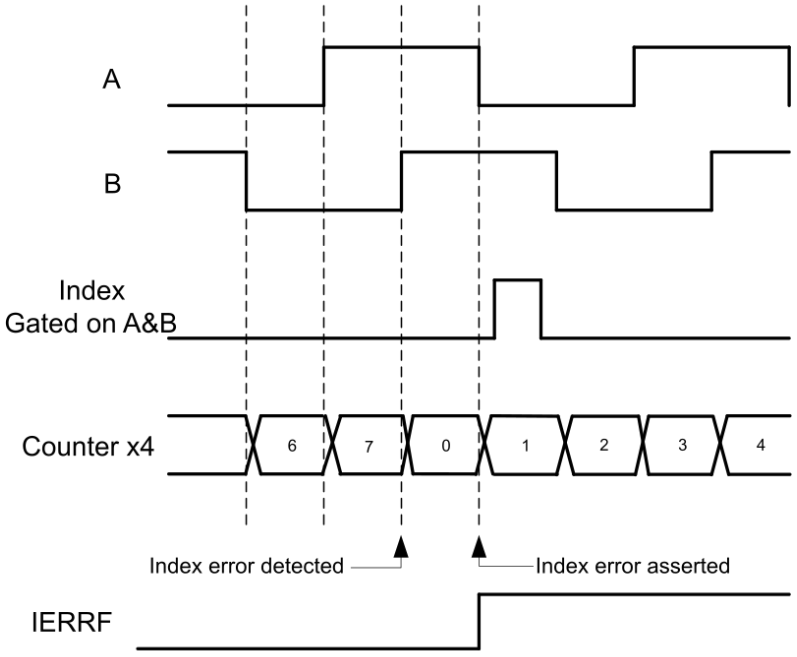


图 20-68 index 错误置位

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。

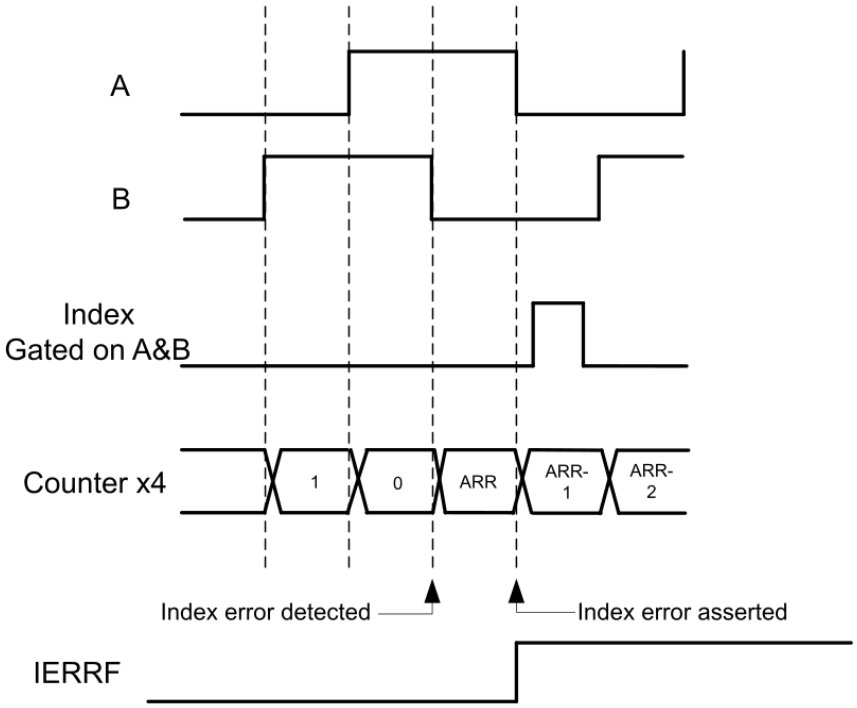


图 20-69 向下计数时 index 错误置位

20.5.22.7 Counter mode preload

某些应用中需要实时切换 counter mode（x4, x2, x1），可以通过 ATIM_SMCR.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

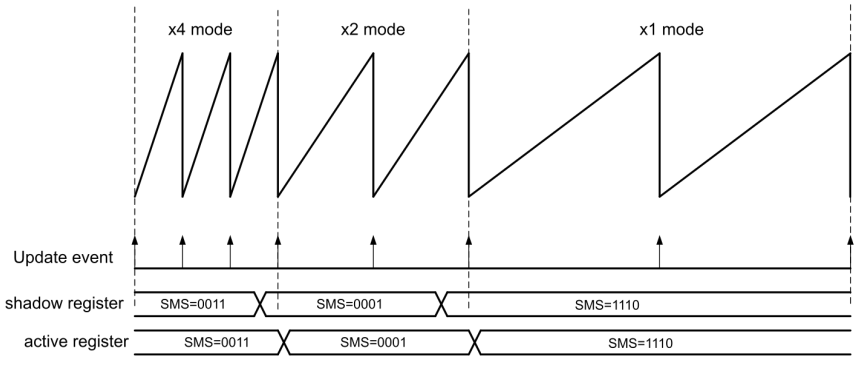


图 20-70 counter mode preload

20.5.22.8 编码器时钟输出

当 ATIM_SMCR.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。

20.5.23 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。

20.5.23.1 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

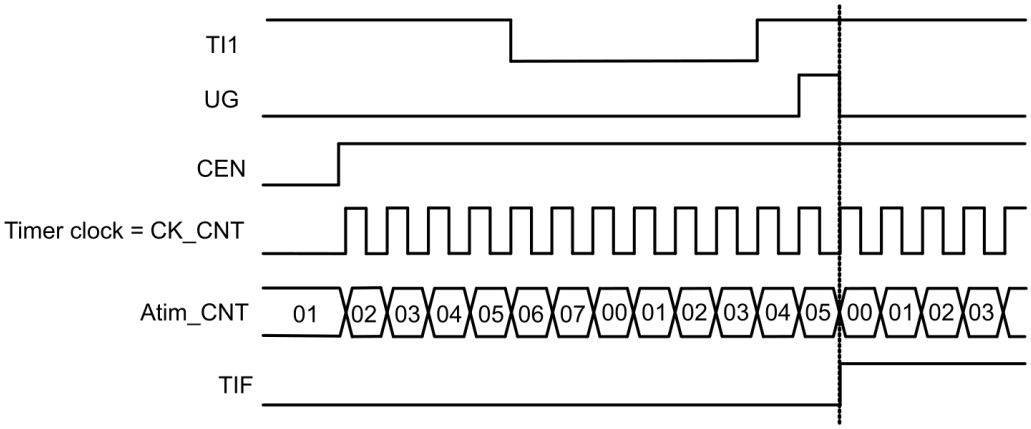


图 20-71 复位模式下的时序

20.5.23.2 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

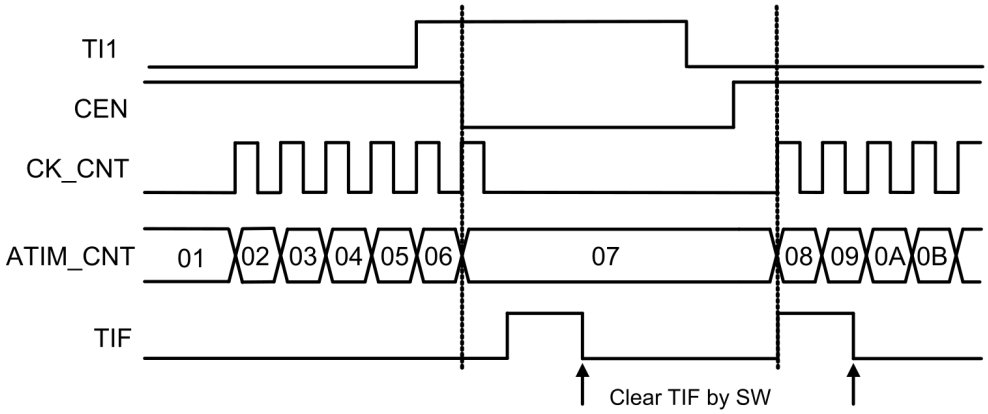


图 20-72 门控模式下的时序

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 ATIM_SMCR.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。

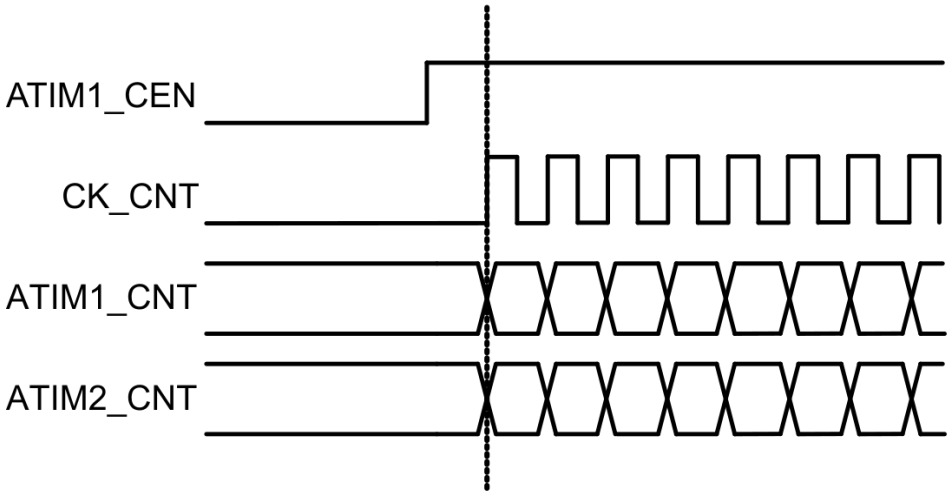


图 20-73 多 timer 同时使能

20.5.23.3 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

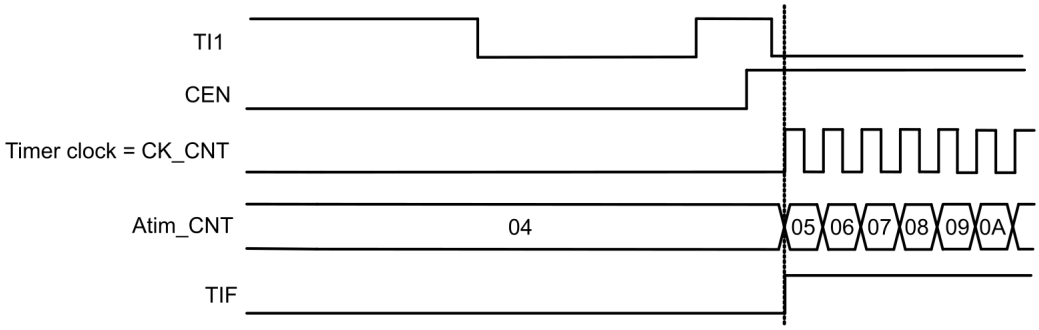


图 20-74 触发器模式下的时序

20.5.23.4 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

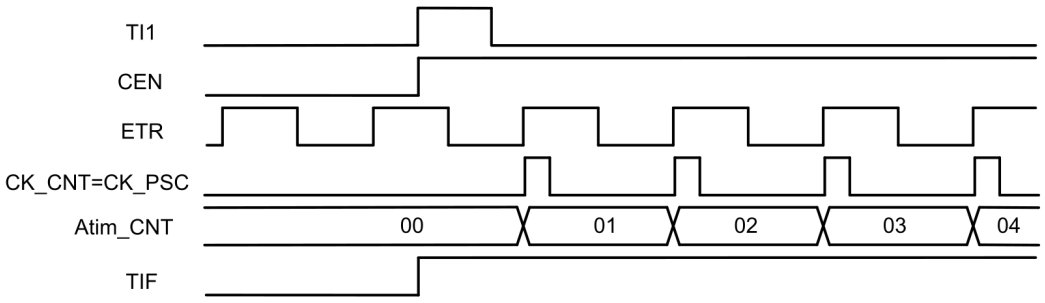


图 20-75 外部时钟模式 2＋触发模式下的时序

20.5.24 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_H,ATIM_CR2.MMS}和 ATIM_CR2.MMS2 寄存器选择触发源。

TRGO 输出组合配置：

{MMS_H,MMS}	TRGO
0000	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
0001	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】
0010	更新事件（UEV）用作 TRGO
0011	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
0100	OC1REFC 用作 TRGO
0101	OC2REFC 用作 TRGO
0110	OC3REFC 用作 TRGO
0111	OC4REFC 用作 TRGO
1000	，编码器模式的计数时钟信号从 TRGO 输出
1001	OC1REFC 的下降沿用作 TRGO
1010	OC2REFC 的下降沿用作 TRGO
1011	OC3REFC 的下降沿用作 TRGO
1100	OC4REFC 的下降沿用作 TRGO
1101	OC5REFC 的下降沿用作 TRGO
1110	OC6REFC 的下降沿用作 TRGO
1111	OC4REFC 的下降沿和 OC6REFC 的下降沿用作 TRGO

TRGO2 输出组合配置：

MMS2	TRGO2

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 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

20.5.25 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。

表 20-10 DMA 访问

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

20.5.26 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）。

20.5.27 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。

20.5.28 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）。

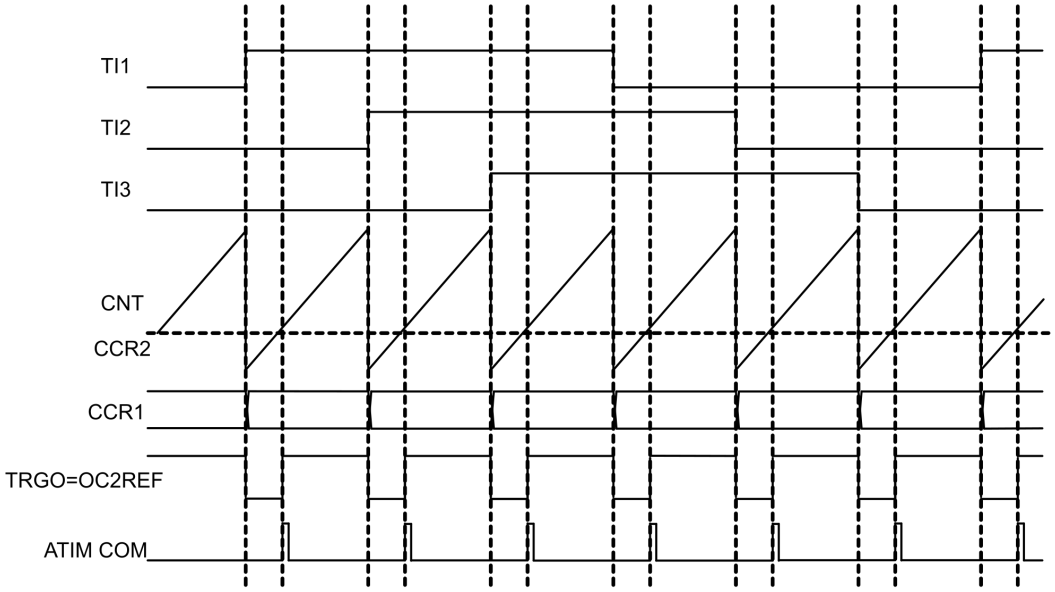


图 20-76 霍尔传感器接口

20.5.29 Debug 模式

当 CPU 进入 debug 模式后，定时器可以停止或继续工作，其行为由 DCU 模块的 DBG_TIMx_STOP 寄存器定义。

Debug 时当定时器被停止后，其输出会被禁止（MOE 清零），根据寄存器配置，此时的输出信号可以被 force 成 inactive 或由 GPIO 模块控制。DMA-Burst 模式下，DMA 所有访问都要指向 DMAR 虚拟寄存器，由 ATIM 自动根据访问来累加内部 offset 地址。DBA 寄存器用于指定 ATIM 内部首次 DMA 传输的目标地址，而 DBL 用于指定 Burst 长度

20.6 寄存器

ATIM 基地址 (0x4001B000)

地址	符号	名称
0x0000	ATIM_CR1	ATIM 控制寄存器 1
0x0004	ATIM_CR2	ATIM 控制寄存器 2
0x0008	ATIM_SMCR	ATIM 从机模式控制寄存器
0x000C	ATIM_DIER	ATIM DMA 和中断使能寄存器
0x0010	ATIM_SR	ATIM 状态寄存器
0x0014	ATIM_EGR	ATIM 事件产生寄存器
0x0018	ATIM_CCMR1	ATIM 捕捉/比较模式寄存器 1
0x001C	ATIM_CCMR2	ATIM 捕捉/比较模式寄存器 2
0x0020	ATIM_CCER	ATIM 捕捉/比较使能寄存器

地址	符号	名称
0x0024	ATIM_CNT	ATIM 计数器寄存器
0x0028	ATIM_PSC	ATIM 预分频寄存器
0x002C	ATIM_ARR	ATIM 自动重载寄存器
0x0030	ATIM_RCR	ATIM 重复计数寄存器
0x0034	ATIM_CCR1	ATIM 捕捉/比较寄存器 1
0x0038	ATIM_CCR2	ATIM 捕捉/比较寄存器 2
0x003C	ATIM_CCR3	ATIM 捕捉/比较寄存器 3
0x0040	ATIM_CCR4	ATIM 捕捉/比较寄存器 4
0x0044	ATIM_BDTR	ATIM 刹车和死区控制寄存器
0x0048	ATIM_CCMR3	ATIM 捕捉/比较模式寄存器 3
0x004C	ATIM_CCR5	ATIM 捕捉/比较寄存器 5
0x0050	ATIM_CCR6	ATIM 捕捉/比较寄存器 6
0x0054	ATIM_ECR	ATIM 编码模式控制寄存器
0x0058	ATIM_TISEL	ATIM 输入选择寄存器
0x005C	ATIM_AFR	ATIM 辅助功能寄存器
0x0060	ATIM_BRKFLAG	ATIM 刹车标志寄存器
0x00F8	ATIM_DCR	ATIM DMA 控制寄存器
0x00FC	ATIM_DMAR	ATIM DMA 访问寄存器

20.6.1 ATIM 控制寄存器 1(ATIM_CR1)

符号	ATIM_CR1							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				UIFRE MAP	RFU	CKD	
权限	R-0000				RW-0	R-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
权限	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11	UIFREMAP	RW	UIF 位替换到 cnt 最高位
10	RFU	R	RFU：未实现，读为 0
9:8	CKD	RW	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：tDTS=tCK_INT 01：tDTS=2*tCK_INT

位	位名	权限	功能描述
			10：tDTS=4*tCK_INT 11：RFU，禁止使用
7	ARPE	RW	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	CMS	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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

20.6.2 ATIM 控制寄存器 2(ATIM_CR2)

符号	ATIM_CR2

符号	ATIM_CR2							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						MMS_H	RFU
权限	R-000000						RW-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	MMS2				RFU	OIS6	RFU	IOS5
权限	RW-0000				R-0	RW-0	R-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
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_L			CCDS	CCUS	RFU1	CCPC
权限	RW-0	RW-000			RW-0	RW-0	R-0	RW-0

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	MMS_H	RW	参考 MMS_L
24	RFU	R	RFU：未实现，读为 0
23:20	MMS2	RW	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 ode 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

位	位名	权限	功能描述
			TRGO2 1111：OC5REFC 的上升沿和 OC6REFC 的下降沿用作 TRGO2
19	RFU	R	RFU：未实现，读为 0
18	OIS6	RW	参考 OIS1
17	RFU	R	RFU：未实现，读为 0
16	IOS5	RW	参考 OIS1
15	OIS4N	RW	参考 OIS1N
14	OIS4	RW	参考 OIS1
13	OIS3N	RW	参考 OIS1N
12	OIS3	RW	参考 OIS1
11	OIS2N	RW	参考 OIS1N
10	OIS2	RW	参考 OIS1
9	OIS1N	RW	定义 OC1N 的输出 IDLE 状态 0：当 MOE=0 时，经过 dead time 后，OC1N=0 1：当 MOE=1 时，经过 dead time 后，OC1N=0
8	OIS1	RW	定义 OC1 的输出 IDLE 状态 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=0 1：当 MOE=1 时（如果使能了互补输出，需经过 dead time 后），OC1=0
7	TI1S	RW	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_L	RW	与 MMS_H 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：OC1REF 用作 TRGO 0101：OC2REF 用作 TRGO 0110：OC3REF 用作 TRGO 0111：OC4REF 用作 TRGO 1000：编码器时钟输出，编码器模式的计数时钟信号从 TRGO 输出，仅当 SMS 配置为编码器模式中的一种时有效 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO 1001：OC1REFC 的下降沿用作 TRGO 1010：OC2REFC 的下降沿用作 TRGO 1011：OC3REFC 的下降沿用作 TRGO 1100：OC4REFC 的下降沿用作 TRGO

位	位名	权限	功能描述
			1101：OC5REFC 的下降沿用作 TRGO 1110：OC6REFC 的下降沿用作 TRGO 1111：OC4REFC 的下降沿和 OC6REFC 的下降沿用作 TRGO
3	CCDS	RW	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	RW	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	RFU1	R	RFU：未实现，读为 0
0	CCPC	RW	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

20.6.3 ATIM 从机模式控制寄存器(ATIM_SMCR)

符号	ATIM_SMCR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						SMSPS	SMSPE
权限	R-000000						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		TS_H		RFU			OCCS
权限	R-00		RW-00		R-000			RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
权限	RW-0	RW-0	RW-00		RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS_L			SMS			
权限	RW-0	RW-000			RW-0000			

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	SMSPS	RW	SMS preload 触发源选择 0：SMS 更新由 update event 触发 1：SMS 更新由 index event 触发
24	SMSPE	RW	SMS preload 使能 0：禁止 SMS preload

位	位名	权限	功能描述
			1：使能 SMS preload
23:22	RFU	R	RFU：未实现，读为 0
21:20	TS_H	RW	参考 TS_L
19:17	RFU	R	RFU：未实现，读为 0
16	OCCS	RW	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	RW	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	RW	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	RW	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	RW	外部触发信号滤波时钟和长度选择 0000：无滤波 0001：fSAMPLING=fCK_INT, N=2 0010：fSAMPLING=fCK_INT, N=4 0011：fSAMPLING=fCK_INT, N=8 0100：fSAMPLING=fDTS/2, N=6 0101：fSAMPLING=fDTS/2, N=8 0110：fSAMPLING=fDTS/4, N=6 0111：fSAMPLING=fDTS/4, N=8 1000：fSAMPLING=fDTS/8, N=6 1001：fSAMPLING=fDTS/8, N=8 1010：fSAMPLING=fDTS/16, N=5 1011：fSAMPLING=fDTS/16, N=6 1100：fSAMPLING=fDTS/16, N=8 1101：fSAMPLING=fDTS/32, N=5 1110：fSAMPLING=fDTS/32, N=6 1111：fSAMPLING=fDTS/32, N=8
7	MSM	RW	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同步
6:4	TS_L	RW	与 TS_H 组成 5bit 内部触发选择，用于选择同步计数器的触发源 00000：内部触发信号（ITR0）

位	位名	权限	功能描述
			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） 10011：内部触发信号（ITR15） Others：RFU 注意：仅当 SMS=0000 即禁止从机模式的情况下，可以改写 TS 寄存器
3:0	SMS	RW	从机模式选择 0000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 0001：Quadrature Encoder 模式 1（x1 mode）；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 0010：Quadrature Encoder 模式 2（x2 mode）；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 0011：Quadrature Encoder 模式 3（x4 mode）；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 0100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 0101：门控模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 0110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 0111：外部时钟模式 1；TRGI 上升沿驱动计数器 1000：组合的复位+触发模式；TRGI 上升沿初始化计数器，产生更新事件并启动计数 1001：组合的闸门+触发模式；TRGI 为高电平时，计数时钟使能，当 TRGI 变低电平时，计数器复位且停止

位	位名	权限	功能描述
			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

20.6.4 ATIM DMA 和中断使能寄存器(ATIM_DIER)

符号	ATIM_DIER							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TERRIE	IERRIE	DIRIE	IDXIE	CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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	RFU1	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
权限	R-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
权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:24	RFU	R	RFU：未实现，读为 0
23	TERRIE	RW	状态转移错误中断使能，用于 encoder 模式，1 有效
22	IERRIE	RW	Index 错误中断使能，用于 encoder 模式，1 有效
21	DIRIE	RW	方向变化中断使能，用于 encoder 模式，1 有效
20	IDXIE	RW	Index 事件中断使能，用于 encoder 模式，1 有效
19	CC4BURSTEN	RW	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	RW	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	RW	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

位	位名	权限	功能描述
16	CC1BURSTEN	RW	捕捉比较通道 1 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	RFU1	R	RFU：未实现，读为 0
14	TDE	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	捕捉/比较通道 1 中断使能 0：禁止捕捉/比较 1 中断

位	位名	权限	功能描述
			1：允许捕捉/比较 1 中断
0	UIE	RW	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

20.6.5 ATIM 状态寄存器(ATIM_SR)

符号	ATIM_SR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TERRIF	IERRIF	DIRIF	IDXIF	RFU		CC6IF	CC5IF
权限	W1C-0	W1C-0	W1C-0	W1C-0	R-00		W1C-0	W1C-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			CC4OF	CC3OF	CC2OF	CC1OF	RFU1
权限	R-000			W1C-0	W1C-0	W1C-0	W1C-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
权限	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:24	RFU	R	RFU：未实现，读为 0
23	TERRIF	W1C	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	W1C	Index 错误中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
21	DIRIF	W1C	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	W1C	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	R	RFU：未实现，读为 0
17	CC6IF	W1C	捕捉/比较通道 6 中断标志 参考 CC1IF
16	CC5IF	W1C	捕捉/比较通道 5 中断标志 参考 CC1IF
15:13	RFU	R	RFU：未实现，读为 0
12	CC4OF	W1C	捕捉/比较通道 4 的 Overcapture 中断 参考 CC1OF
11	CC3OF	W1C	捕捉/比较通道 3 的 Overcapture 中断 参考 CC1OF
10	CC2OF	W1C	捕捉/比较通道 2 的 Overcapture 中断 参考 CC1OF
9	CC1OF	W1C	捕捉/比较通道 1 的 Overcapture 中断

位	位名	权限	功能描述
			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	RFU1	R	RFU：未实现，读为 0
7	BIF	W1C	刹车事件中断标志，硬件置位，软件写 1 清零
6	TIF	W1C	触发事件中断标志，硬件置位，软件写 1 清零
5	COMIF	W1C	COM 事件中断标志，硬件置位，软件写 1 清零
4	CC4IF	W1C	捕捉/比较通道 4 中断标志 参考 CC1IF
3	CC3IF	W1C	捕捉/比较通道 3 中断标志 参考 CC3IF
2	CC2IF	W1C	捕捉/比较通道 2 中断标志 参考 CC2IF
1	CC1IF	W1C	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	W1C	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

20.6.6 ATIM 事件产生寄存器(ATIM_EGR)

符号	ATIM_EGR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							BG2
权限	R-0000000							RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:9	RFU	R	RFU：未实现，读为 0
8	BG2	RW	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
7	BG	RW	软件刹车，软件置位此寄存器产生刹车事件，硬件自动清零
6	TG	RW	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	RW	软件 COM 事件，硬件置位，软件写 1 清零
4	CC4G	RW	捕捉/比较通道 4 软件触发，参考 CC1G
3	CC3G	RW	捕捉/比较通道 3 软件触发，参考 CC1G
2	CC2G	RW	捕捉/比较通道 2 软件触发，参考 CC1G
1	CC1G	RW	捕捉/比较通道 1 软件触发 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 ATIM_CCR1 寄存器，CC1IF 置位， 在使能的情况下可以产生相应的中断和 DMA 请求
0	UG	RW	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。

20.6.7 ATIM 捕捉/比较模式寄存器 1(ATIM_CCMR1)

输出比较模式

符号	ATIM_CCMR1							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							OC2M_H
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OC1M_H
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M_L			OC2PE	OC2FE	CC2S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M_L			OC1PE	OC1FE	CC1S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	

位	位名	权限	功能描述

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	OC2M_H	RW	参考 OC2M_L
23:17	RFU	R	RFU：未实现，读为 0
16	OC1M_H	RW	参考 OC1M_L
15	OC2CE	RW	输出比较 2 清零使能，参考 OC1CE
14:12	OC2M_L	RW	输出比较 2 模式配置，参考 OC1M_L
11	OC2PE	RW	输出比较 2 预装载使能，参考 OC1PE
10	OC2FE	RW	输出比较 2 快速使能，参考 OC1FE
9:8	CC2S	RW	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	RW	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	OC1M_L	RW	与 OC1M_H 组成 4bit 输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 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 置高，否则置低 1000：可重触发的 OPM 模式 1，向上计数时若计数器停止输出高（active）否则输出 pwm 模式 1 波形， 向下计数时若计数器停止输出低（inactive）否则输出 pwm 模式 1 波形，配合 sms 配置为复位+触发模式使用。 1001：可重触发的 OPM 模式 2，向上计数时若计数器停止输出低（inactive）否则输出 pwm 模式 2 波形， 向下计数时若计数器停止输出高（active）否则输出 pwm 模式 2 波形，配合 sms 配置为复位+触发模式使用。 1010：RFU 1011：RFU 1100：组合 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 或 oc2ref 的逻

位	位名	权限	功能描述
			辑， 通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1101：组合 PWM 模式 2，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 与 oc2ref 的逻辑， 通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1110：非对称 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形在向上计数时为 oc1ref 波形， 向下计数时为 oc2ref 波形，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1111：非对称 PWM 模式 2，oc1ref 波形与 pwm 模式 2 一致，oc1refc 波形在向上计数时为 oc1ref 波形， 向下计数时为 oc2ref 波形，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组）
3	OC1PE	RW	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 vent 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC1FE	RW	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	RW	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

符号	ATIM_CCMR1							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	ATIM_CCMR1							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IC2F				IC2PSC		CC2S	
权限	RW-0000				RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IC1F				IC1PSC		CC1S	
权限	RW-0000				RW-00		RW-00	

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:12	IC2F	RW	输入捕捉 2 滤波
11:10	IC2PSC	RW	输入捕捉 2 预分频
9:8	CC2S	RW	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
7:4	IC1F	RW	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 fDTS 采样 0001：fSAMPLING=fCK_INT, N=2 0010：fSAMPLING=fCK_INT, N=4 0011：fSAMPLING=fCK_INT, N=8 0100：fSAMPLING=fDTS/2, N=6 0101：fSAMPLING=fDTS/2, N=8 0110：fSAMPLING=fDTS/4, N=6 0111：fSAMPLING=fDTS/4, N=8 1000：fSAMPLING=fDTS/8, N=6 1001：fSAMPLING=fDTS/8, N=8 1010：fSAMPLING=fDTS/16, N=5 1011：fSAMPLING=fDTS/16, N=6 1100：fSAMPLING=fDTS/16, N=8 1101：fSAMPLING=fDTS/32, N=5 1110：fSAMPLING=fDTS/32, N=6 1111：fSAMPLING=fDTS/32, N=8
3:2	IC1PSC	RW	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
1:0	CC1S	RW	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

位	位名	权限	功能描述
			注意：CC1S 仅在通道关闭时（CC1E=0）可以写

20.6.8 ATIM 捕捉/比较模式寄存器 2(ATIM_CCMR2)

输出比较模式

符号	ATIM_CCMR2							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							OC4M_H
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OC3M_H
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M_L			OC4PE	OC4FE	CC4S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M_L			OC3PE	OC3FE	CC3S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	OC4M_H	RW	参考 OC4M_L
23:17	RFU	R	RFU：未实现，读为 0
16	OC3M_H	RW	参考 OC3M_L
15	OC4CE	RW	输出比较 4 清零使能，参考 OC1CE
14:12	OC4M_L	RW	输出比较 4 模式配置，参考 OC1M_L
11	OC4PE	RW	输出比较 4 预装载使能，参考 OC1PE
10	OC4FE	RW	输出比较 4 快速使能，参考 OC1FE
9:8	CC4S	RW	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	RW	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	OC3M_L	RW	输出比较 3 模式配置，参考 OC1M_L
3	OC3PE	RW	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

位	位名	权限	功能描述
			操作都是访问 preload 寄存器，当 update vent 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC3FE	RW	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	RW	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，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 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

符号	ATIM_CCMR2							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IC4F				IC4PSC		CC4S	
权限	RW-0000				RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IC3F				IC3PSC		CC3S	
权限	RW-0000				RW-00		RW-00	

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:12	IC4F	RW	输入捕捉 4 滤波
11:10	IC4PSC	RW	输入捕捉 4 预分频
9:8	CC4S	RW	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	RW	输入捕捉 3 滤波
3:2	IC3PSC	RW	输入捕捉 3 预分频

位	位名	权限	功能描述
1:0	CC3S	RW	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，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

20.6.9 ATIM 捕捉/比较使能寄存器(ATIM_CCER)

符号	ATIM_CCER							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		CC6P	CC6E	RFU		CC5P	CC5E
权限	R-00		RW-0	RW-0	R-00		RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
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
权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:22	RFU	R	RFU：未实现，读为 0
21	CC6P	RW	捕捉/比较 6 输出极性，参考 CC1P
20	CC6E	RW	捕捉/比较 6 输出使能，参考 CC1E
19:18	RFU	R	RFU：未实现，读为 0
17	CC5P	RW	捕捉/比较 5 输出极性，参考 CC1P
16	CC5E	RW	捕捉/比较 5 输出使能，参考 CC1E
15	CC4NP	RW	捕捉/比较 4 互补输出极性，参考 CC1NP
14	CC4NE	RW	捕捉/比较 4 互补输出使能，参考 CC1NE
13	CC4P	RW	捕捉/比较 4 输出极性，参考 CC1P
12	CC4E	RW	捕捉/比较 4 输出使能，参考 CC1E
11	CC3NP	RW	捕捉/比较 3 互补输出极性，参考 CC1NP
10	CC3NE	RW	捕捉/比较 3 互补输出使能，参考 CC1NE
9	CC3P	RW	捕捉/比较 3 输出极性，参考 CC1P
8	CC3E	RW	捕捉/比较 3 输出使能，参考 CC1E
7	CC2NP	RW	捕捉/比较 2 互补输出极性，参考 CC1NP
6	CC2NE	RW	捕捉/比较 2 互补输出使能，参考 CC1NE
5	CC2P	RW	捕捉/比较 2 输出极性，参考 CC1P
4	CC2E	RW	捕捉/比较 2 输出使能，参考 CC1E
3	CC1NP	RW	捕捉/比较 1 互补输出极性 0：OC1N 高电平为 active

位	位名	权限	功能描述
			1：OC1N 低电平为 active
2	CC1NE	RW	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	CC1P	RW	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 IC1 的上升沿 CC1NP=0，CC1P=1：捕捉 IC1 的下降沿 CC1NP=1：捕捉 IC1 的上升沿和下降沿
0	CC1E	RW	捕捉/比较 1 输出使能 CC1 通道配置为输出时 0：OC1 不 active 1：OC1 active CC1 通道配置为输入时 0：关闭捕捉功能 1：使能捕捉功能

20.6.10 ATIM 计数器寄存器(ATIM_CNT)

符号	ATIM_CNT							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UIFCPY	RFU						
权限	R-0	R-0000000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	RW-00000000							

位	位名	权限	功能描述
31	UIFCPY	R	UIF copy 当 UIFREMAP=1，UIF 寄存器的值被实时复制到这里，只读
30:16	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
15:0	CNT	RW	计数器值

20.6.11 ATIM 预分频寄存器(ATIM_PSC)

符号	ATIM_PSC							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PSC	RW	计数器时钟（CK_CNT）预分频值 fCK_CNT=fCK_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

20.6.12 ATIM 自动重载寄存器(ATIM_ARR)

符号	ATIM_ARR							
地址	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-11111111							

位	位名	权限	功能描述

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	ARR	RW	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

20.6.13 ATIM 重复计数寄存器(ATIM_RCR)

符号	ATIM_RCR							
地址	0x0030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	REP							
权限	RW-00000000							

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:0	REP	RW	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

20.6.14 ATIM 捕捉/比较寄存器 1(ATIM_CCR1)

符号	ATIM_CCR1							
地址	0x0034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR1	RW	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

20.6.15 ATIM 捕捉/比较寄存器 2(ATIM_CCR2)

符号	ATIM_CCR2							
地址	0x0038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR2	RW	捕捉/比较通道 2 寄存器 如果通道 2 配置为输出： 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 如果通道 2 配置为输入： CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

20.6.16 ATIM 捕捉/比较寄存器 3(ATIM_CCR3)

符号	ATIM_CCR3							
地址	0x003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

符号	ATIM_CCR3							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR3	RW	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

20.6.17 ATIM 捕捉/比较寄存器 4(ATIM_CCR4)

符号	ATIM_CCR4							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR4	RW	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

20.6.18 ATIM 刹车和死区控制寄存器(ATIM_BDTR)

符号	ATIM_BDTR							
地址	0x0044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						BK2P	BK2E
权限	R-000000						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	BK2F				BKF			
权限	RW-0000				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	
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							
权限	RW-00000000							

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	BK2P	RW	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	RW	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	RW	Break2 滤波，定义 BRK2 信号滤波长度 0000：无数字滤波 0001：采样时钟=fCK_INT, N=2 0010：采样时钟=fCK_INT, N=4 0011：采样时钟=fCK_INT, N=8 0100：采样时钟=fDTS, N=12 0101：采样时钟=fDTS, N=16 0110：采样时钟=fDTS, N=24 0111：采样时钟=fDTS, N=32 1000：采样时钟=fDTS, N=48 1001：采样时钟=fDTS, N=64 1010：采样时钟=fDTS, N=80 1011：采样时钟=fDTS, N=96 1100：采样时钟=fDTS, N=128 1101：采样时钟=fDTS, N=160 1110：采样时钟=fDTS, N=192 1111：采样时钟=fDTS, N=256

位	位名	权限	功能描述
			注意：当 LOCK 寄存器配置为 level1 之后此寄存器不能改写
19:16	BKF	RW	Break 滤波，定义 BRK 信号滤波长度 0000：无数字滤波 0001：采样时钟=fCK_INT, N=2 0010：采样时钟=fCK_INT, N=4 0011：采样时钟=fCK_INT, N=8 0100：采样时钟=fDTS, N=12 0101：采样时钟=fDTS, N=16 0110：采样时钟=fDTS, N=24 0111：采样时钟=fDTS, N=32 1000：采样时钟=fDTS, N=48 1001：采样时钟=fDTS, N=64 1010：采样时钟=fDTS, N=80 1011：采样时钟=fDTS, N=96 1100：采样时钟=fDTS, N=128 1101：采样时钟=fDTS, N=160 1110：采样时钟=fDTS, N=192 1111：采样时钟=fDTS, 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	RW	输出使能主控 此寄存器控制所有通道的输出使能，每个通道独立的输出使能还需要 CCxE 和 CCxNE 来控制。 MOE 由软件置位，或者在 AOE=1 的情况下硬件触发自动置位。当刹车输入有效时，MOE 被硬件异步清零。 0：关闭 OC 和 OCN 输出，具体 IO 输出状态由 OSSI 决定 1：使能 OC 和 OCN 输出（仍需各个通道的 CCxE 和 CCxNE 状态来决定是否输出）
14	AOE	RW	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
13	BKP	RW	刹车极性 0：刹车输入为低电平有效 1：刹车输入为高电平有效
12	BKE	RW	刹车使能 0：禁止刹车输入 1：允许刹车输入
11	OSSR	RW	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

位	位名	权限	功能描述
			无效状态
10	OSSI	RW	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 先驱动空闲状态，待死区时间结束后，启动无效状态
9:8	LOCK	RW	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	RW	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] * tDTS DTG[7:5]=10x：DT=(64+DTG[5:0]) * 2 * tDTS DTG[7:5]=110：DT=(32+DTG[4:0]) * 8 * tDTS DTG[7:5]=111：DT=(32+DTG[4:0]) * 16 * tDTS

20.6.19 ATIM 捕捉/比较模式寄存器 3(ATIM_CCMR3)

符号	ATIM_CCMR3							
地址	0x0048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							OC6M_H
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OC5M_H
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC6CE	OC6M_L			OC6PE	OC6FE	RFU	
权限	RW-0	RW-000			RW-0	RW-0	R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC5CE	OC5M_L			OC5PE	OC5FE	RFU	
权限	RW-0	RW-000			RW-0	RW-0	R-00	

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
24	OC6M_H	RW	参考 OC6M_L
23:17	RFU	R	RFU：未实现，读为 0
16	OC5M_H	RW	参考 OC5M_L
15	OC6CE	RW	输出比较 6 清零使能，参考 OC1CE
14:12	OC6M_L	RW	输出比较 6 模式配置，参考 OC1M_L
11	OC6PE	RW	输出比较 6 预装载使能，参考 OC1PE
10	OC6FE	RW	输出比较 6 快速使能，参考 OC1FE
9:8	RFU	R	RFU：未实现，读为 0
7	OC5CE	RW	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 OC5REF
6:4	OC5M_L	RW	输出比较 5 模式配置，参考 OC1M_L
3	OC5PE	RW	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 vent 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC5FE	RW	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 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	RFU	R	RFU：未实现，读为 0

20.6.20 ATIM 捕捉/比较寄存器 5(ATIM_CCR5)

符号	ATIM_CCR5							
地址	0x004C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	GC5C3	GC5C2	GC5C1	RFU				
权限	RW-0	RW-0	RW-0	R-00000				
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR5							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR5							
权限	RW-00000000							

位	位名	权限	功能描述
31	GC5C3	RW	Group Channel 5 and Channel 3 0：OC3REFC 不受 OC5REF 影响 1：OC3REFC=OC3REF & OC5REF，仅在 pwm 模式下有效
30	GC5C2	RW	Group Channel 5 and Channel 2 0：OC2REFC 不受 OC5REF 影响 1：OC2REFC=OC2REF & OC5REF，仅在 pwm 模式下有效
29	GC5C1	RW	Group Channel 5 and Channel 1 0：OC1REFC 不受 OC5REF 影响 1：OC1REFC=OC1REF & OC5REF，仅在 pwm 模式下有效
28:16	RFU	R	RFU：未实现，读为 0
15:0	CCR5	RW	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

20.6.21 ATIM 捕捉/比较寄存器 6(ATIM_CCR6)

符号	ATIM_CCR6							
地址	0x0050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR6							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR6							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR6	RW	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

20.6.22 ATIM 编码模式控制寄存器(ATIM_ECR)

符号	ATIM_ECR							
地址	0x0054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	ATIM_ECR							
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IPOS		FIDX	RFU		IDIR		INDEX_EN
权限	RW-00		RW-0	R-00		RW-00		RW-0

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:6	IPOS	RW	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	RW	First Index 1：仅第一个 index 有效 0：所有 index 有效
4:3	RFU	R	RFU：未实现，读为 0
2:1	IDIR	RW	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	INDEX_EN	RW	Index 使能 0：禁止 index 功能 1：使能 index 功能

20.6.23 ATIM 输入选择寄存器(ATIM_TISEL)

符号	ATIM_TISEL							
地址	0x0058							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	ATIM_TISEL							
位名	RFU				TI4SEL			
权限	R-0000				RW-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				TI3SEL			
权限	R-0000				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				TI2SEL			
权限	R-0000				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TI1SEL			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:28	RFU	R	RFU：未实现，读为 0
27:24	TI4SEL	RW	TI4 的输入选择信号 0000：tim_ti4_in0:TIMx_CH4 0001: tim_ti4_in1 … 1111: tim_ti4_in15
23:20	RFU	R	RFU：未实现，读为 0
19:16	TI3SEL	RW	TI3 的输入选择信号 0000：tim_ti3_in0:TIMx_CH3 0001: tim_ti3_in1 … 1111: tim_ti3_in15
15:12	RFU	R	RFU：未实现，读为 0
11:8	TI2SEL	RW	TI4 的输入选择信号 0000：tim_ti2_in0:TIMx_CH2 0001: tim_ti2_in1 … 1111: tim_ti2_in15
7:4	RFU	R	RFU：未实现，读为 0
3:0	TI1SEL	RW	TI1 的输入选择信号 0000：tim_ti1_in0:TIMx_CH1 0001: tim_ti1_in1 … 1111: tim_ti1_in15

20.6.24 ATIM 辅助功能寄存器(ATIM_AFR)

符号	ATIM_AFR							
地址	0x005C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

符号	ATIM_AFR							
位名	ETRSEL				RFU	OCC1R_SEL		
权限	RW-0000				R-0	RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				BKSYS3E	BKSYS2E	BKSYS1E	BKSYS0E
权限	R-0000				RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		BKCMP4E	BKCMP3E	BKCMP2E	BKCMP1E	BKIN2E	BKIN1E
权限	R-00		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:24	RFU	R	RFU，未实现，读为 0
23:20	ETRSEL	RW	ETR 的输入选择信号 0000：tim_etr_in0:TIMx_etr 0001: tim_etr_in1 … 1111: tim_etr_in15
19	RFU	R	RFU，未实现，读为 0
18:16	OCC1R_SEL	RW	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	R	RFU，未实现，读为 0
11	BKSYS3E	RW	系统刹车信号 3 使能 0：禁止系统刹车信号 3 1：使能系统刹车信号 3
10	BKSYS2E	RW	LOCKUP 刹车信号使能 0：禁止 LOCKUP 刹车信号 1：使能 LOCKUP 刹车信号
9	BKSYS1E	RW	SVD 刹车信号使能 0：禁止 SVD 刹车信号 1：使能 SVD 刹车信号
8	BKSYS0E	RW	高频晶振停震刹车信号使能 0：禁止高频晶振停震刹车信号 1：使能高频晶振停震刹车信号
7:6	RFU	R	RFU，未实现，读为 0
5	BKCMP4E	RW	比较器输出刹车信号 4 使能 0：禁止比较器刹车信号 1：使能比较器刹车信号
4	BKCMP3E	RW	比较器输出刹车信号 3 使能 0：禁止比较器刹车信号 1：使能比较器刹车信号
3	BKCMP2E	RW	比较器输出刹车信号 2 使能

位	位名	权限	功能描述
			0：禁止比较器刹车信号 1：使能比较器刹车信号
2	BKCMP1E	RW	比较器输出刹车信号 1 使能 0：禁止比较器刹车信号 1：使能比较器刹车信号
1	BKIN2E	RW	外部刹车信号 2 使能 0：禁止比较器刹车信号 1：使能比较器刹车信号
0	BKIN1E	RW	外部刹车信号 1 使能 0：禁止外部刹车信号 1：使能外部刹车信号

20.6.25 ATIM 刹车标志寄存器(ATIM_BRKFLAG)

符号	ATIM_BRKFLAG							
地址	0x0060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				BKSYS3_FLAG	BKSYS2_FLAG	BKSYS1_FLAG	BKSYS0_FLAG
权限	R-0000				W1C-0	W1C-0	W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		BKCMP4_FLAG	BKCMP3_FLAG	BKCMP2_FLAG	BKCMP1_FLAG	BKIN2_FLAG	BKIN1_FLAG
权限	R-00		W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:12	RFU	R	RFU，未实现，读为 0
11	BKSYS3_FLAG	W1C	系统刹车信号 3 标志，写 1 清零
10	BKSYS2_FLAG	W1C	LOCKUP 刹车信号标志，写 1 清零
9	BKSYS1_FLAG	W1C	SVD 刹车信号标志，写 1 清零
8	BKSYS0_FLAG	W1C	高频晶振停震刹车标志，写 1 清零
7:6	RFU	R	RFU，未实现，读为 0
5	BKCMP4_FLAG	W1C	比较器刹车信号 4 刹车标志，写 1 清零
4	BKCMP3_FLAG	W1C	比较器刹车信号 3 刹车标志，写 1 清零
3	BKCMP2_FLAG	W1C	比较器刹车信号 2 刹车标志，写 1 清零
2	BKCMP1_FLAG	W1C	比较器刹车信号 1 刹车标志，写 1 清零
1	BKIN2_FLAG	W1C	外部刹车信号 2 刹车标志，写 1 清零

位	位名	权限	功能描述
0	BKIN1_FLAG	W1C	外部刹车信号 1 刹车标志，写 1 清零

20.6.26 ATIM DMA 控制寄存器(ATIM_DCR)

符号	ATIM_DCR							
地址	0x00F8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			DBL				
权限	R-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU1			DBA				
权限	R-000			RW-00000				

位	位名	权限	功能描述
31:13	RFU	R	RFU：未实现，读为 0
12:8	DBL	RW	DMA Burst 长度 对 ATIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~23 00000：长度=1 00001：长度=2 …… 10110：长度=23 其他：无效值，禁止写入
7:5	RFU1	R	RFU：未实现，读为 0
4:0	DBA	RW	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。

20.6.27 ATIM DMA 访问寄存器(ATIM_DMAR)

符号	ATIM_DMAR							
地址	0x00FC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR							

符号	ATIM_DMAR							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DMAR							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	DMAR	RW	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

21 通用定时器（GPTIM）

21.1 概述

通用定时器包含一个 16bit 自动重载计数器及一个可编程预分频器。

通用定时器可以支持多种应用，包括如捕捉、输出比较、PWM。

21.2 主要特性

- 16bit 向上、向下、双向计数自动重载计数器
- 16bit 可编程预分频器，支持实时调整计数时钟分频
- 4 个独立通道可用于输入捕捉、输出比较、PWM（边缘或中心对齐模式）、单脉冲输出
- 支持与其他定时器级联
- 支持在以下事件发生时产生中断
  - 计数器溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕捉
  - 输出比较

21.3 结构框图

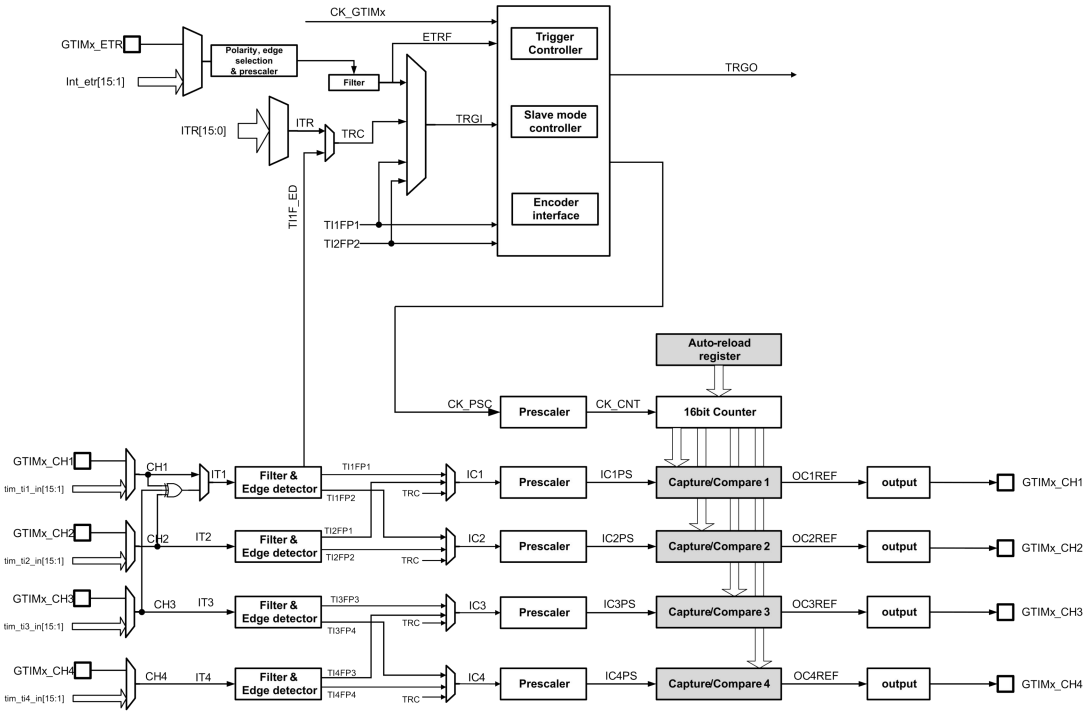


图 21-1 通用定时器架构示意图

21.4 管脚和内部连接

表 21-1 GPTIM 输入输出引脚

signal name	type	Description
ITR[15:0]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Int_etr[15:1]	input	外部触发输入，可用于触发信号或者外部计数时钟 在使能了输入预分频的情况下，可以输入高于 CK_GTIM 频率的计数时钟
Tim_ti1_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti2_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti3_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
Tim_ti4_in[15:1]	input	芯片内部触发信号输入，可用作触发、捕捉、计数时钟（频率不能高于 CK_GTIM 的 1/4）
TRGO	output	内部触发信号输出，用于连接到其他 timer 或外设
APBCLK	input	寄存器总线时钟
GTIM_CC1_DMA GTIM_CC2_DMA GTIM_CC3_DMA GTIM_CC4_DMA	output	4 个通道各自的 DMA 请求输出
GTIM_UEV_DMA	output	更新事件 DMA 请求 N/A

表 21-2 GPTIM 内部输入输出信号

Pin name	direction	description N/A
GPTx_CH1 GPTx_CH2 GPTx_CH3 GPTx_CH4	Input/output	GPTIM 输入输出通道，可用于 PWM 输出或输入信号捕获 CH1 和 CH2 也可用作外部计数时钟输入，外部触发或者正交编码器输入 CH1、CH2、CH3 可以用作数字霍尔传感器接口
GPTx_ETR	input	外部触发信号输入，或外部计数时钟输入

表 21-3 GPTIM 内部触发信号连接

ITRx	GTIM0
ITR0	ATIM_TRGO
ITR1	ATIM_TRGO2
ITR2	BTIM_TRGO
ITR3	LPTIM_TRGO
ITR4	TAU00_TRGO
ITR5	TAU01_TRGO
ITR6	TAU02_TRGO
ITR7	TAU03_TRGO
ITR8	TAU04_TRGO

ITRx	GTIM0
ITR9	TAU05_TRGO
ITR10	TAU06_TRGO
ITR11	TAU07_TRGO
ITR12	TAU10_TRGO
ITR13	TAU11_TRGO
ITR14	TAU12_TRGO
ITR15	TAU13_TRGO

表 21-4 GPTIM 外部触发信号连接

ETRx	GTIM0
ETR0	GTIM0_ETR
ETR1	COMP1_OUT
ETR2	RFU
ETR3	RFU
ETR4	RFU
ETR5	RFU
ETR6	RFU
ETR7	RFU
ETR8	RFU
ETR9	RFU
ETR10	RFU
ETR11	RFU
ETR12	RFU
ETR13	RFU
ETR14	RFU
ETR15	RFU

表 21-5 GPTIM 通道 1 信号连接

GTIMx_CH1	GTIM0
CH1 0	GTIM0_CH1
CH1 1	XTHF
CH1 2	RCLP
CH1 3	RFU
CH1 4	RCHF
CH1 5	RFU
CH1 6	RFU
CH1 7	RFU
CH1 8	RFU
CH1 9	LUT0_OUT
CH1 10	LUT1_OUT
CH1 11	LUT2_OUT
CH1 12	LUT3_OUT
CH1 13	RFU
CH1 14	RFU
CH1 15	RFU

表 21-6 GPTIM 通道 2 信号连接

GTIMx_CH2	GTIM0

GTIMx_CH2	GTIM0
CH2 0	GTIM0_CH2
CH2 1	BTIM_TRGO
CH2 2	LPT_TRGO
CH2 3	ATIM0_TRGO
CH2 4	ATIM0_TRGO2
CH2 5	RFU
CH2 6	RFU
CH2 7	RFU
CH2 8	RFU
CH2 9	RFU
CH2 10	RFU
CH2 11	RFU
CH2 12	RFU
CH2 13	RFU
CH2 14	RFU
CH2 15	RFU

表 21-7 GPTIM 通道 3 信号连接

GTIMx_CH3	GTIM0
CH3 0	GTIM0_CH3
CH3 1	RFU
CH3 2	UART1_RXD
CH3 3	UART2_RXD
CH3 4	RFU
CH3 5	UART4_RXD
CH3 6	UART5_RXD
CH3 7	RFU
CH3 8	RFU
CH3 9	COMP1_OUT
CH3 10	RFU
CH3 11	RFU
CH3 12	RFU
CH3 13	RFU
CH3 14	RFU
CH3 15	RFU

表 21-8 GPTIM 通道 4 信号连接

GTIMx_CH4	GTIM0
CH4 0	GTIM0_CH4
CH4 1	RFU
CH4 2	UART1_RXD
CH4 3	UART2_RXD
CH4 4	RFU
CH4 5	UART4_RXD
CH4 6	UART5_RXD
CH4 7	RFU
CH4 8	RFU
CH4 9	LUT0_OUT
CH4 10	LUT1_OUT

GTIMx_CH4	GTIM0
CH4 11	LUT2_OUT
CH4 12	LUT3_OUT
CH4 13	RFU
CH4 14	RFU
CH4 15	RFU

21.5 功能描述

21.5.1 定时单元

高级定时器的定时单元由一个 16 位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过 16 位预分频器对 APBCLK 进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（GPTIM_CNT）
- 预分频寄存器（GPTIM_PSC）
- 自动重载寄存器（GPTIM_ARR）

ARR 包含预装载功能，该功能通过 ARPE（Auto Reload Preload Enable）寄存器控制。当 ARPE=0 时，对 ARR 寄存器执行写入，写入数据将直接传入到影子寄存器；当 ARPE=1 时，对 ARR 寄存器执行写入的数据在 update event（GPTIM_CNT 上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发 ARR 更新（UEV）。

GPTIM_CNT 工作时钟由 GPTIM_PSC 产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT 才开始计数。当 CNT=ARR 时，本轮计数结束，发送 update event。

GPTIM_PSC 是一个同步预分频器，能够对 APBCLK 进行 1~65536 分频。PSC 寄存器同样被缓存，改写 PSC 实际不改写影子寄存器，只有当新的 update event 到来时，才会从 PSC 更新至影子寄存器。因此在 CNT 计数过程中，软件可以实时改写 PSC，而新的预分频比将在下一更新事件发生时被采用。

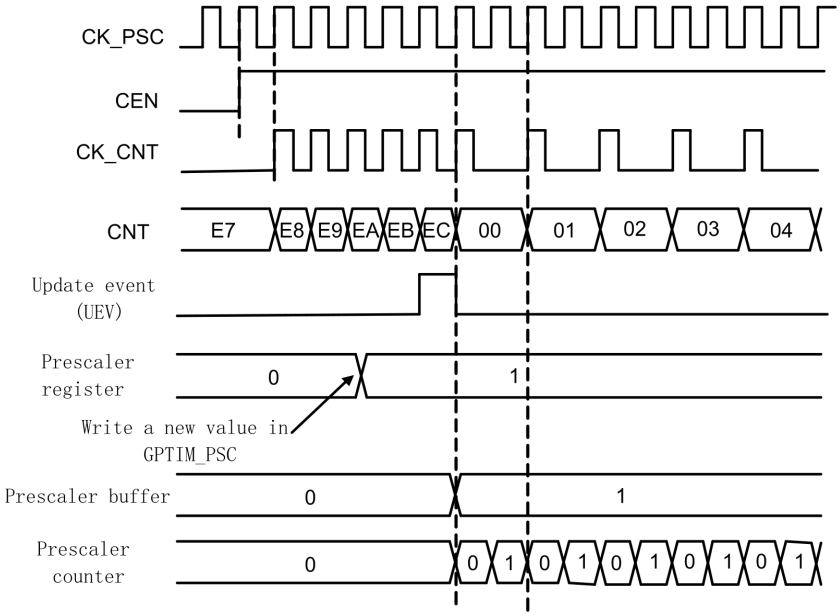


图 21-2 预分频从 1 变为 2 的波形

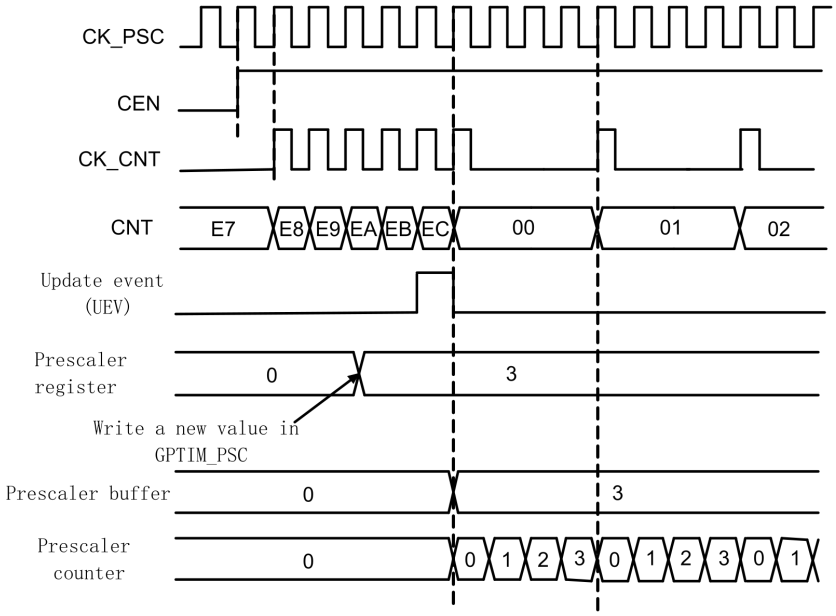


图 21-3 预分频从 1 变为 4 的波形

21.5.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

向上计数

此模式中，计数器使能后从 0 开始计数，直到 CNT=ARR，产生溢出事件，然后重新从 0 开始计数。

软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF（Update Interrupt Flag）中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- ARR 影子寄存器被更新为 GPTIM_ARR 内容
- PSC 影子寄存器被更新为 GPTIM_PSC 内容

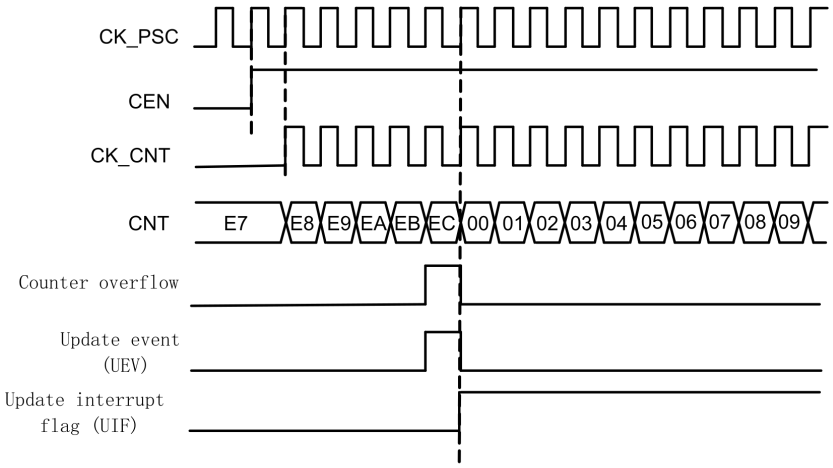


图 21-4 向上计数波形，内部时钟不分频

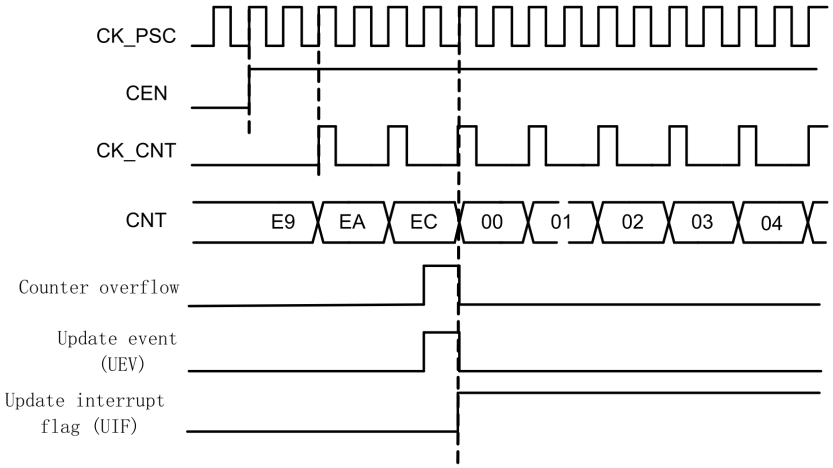


图 21-5 向上计数波形，内部时钟 2 分频

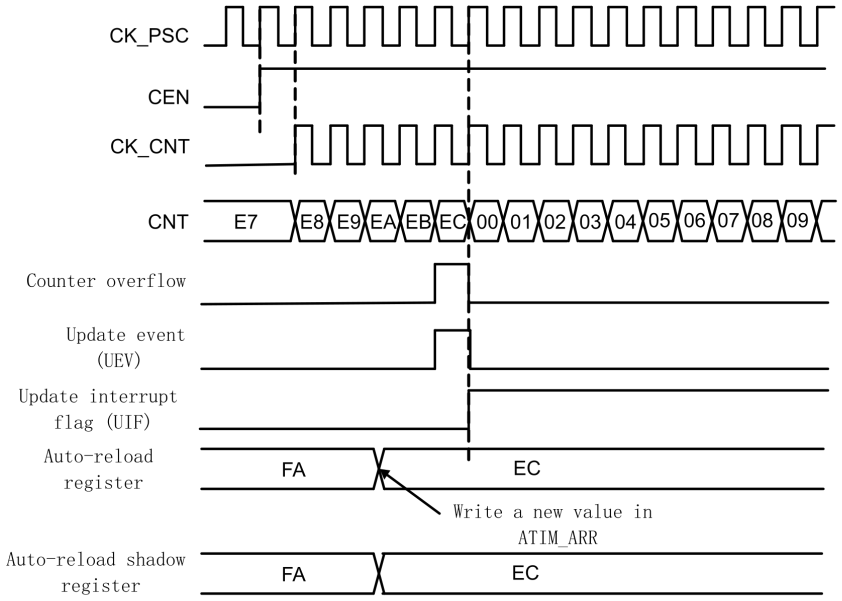


图 21-6 ARPE=0（GPTIM_ARR 没有预装载）时的更新事件

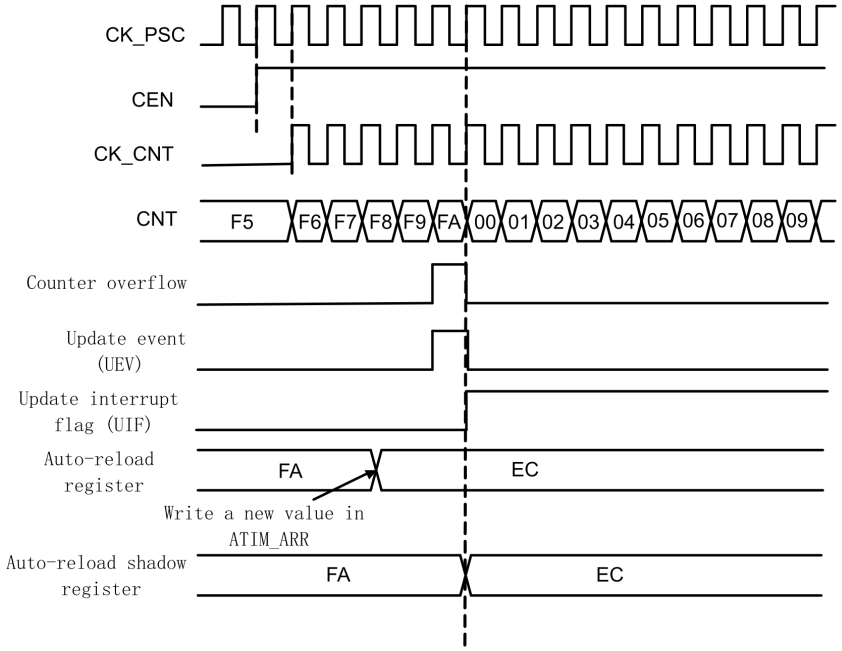


图 21-7 ARPE=1（GPTIM_ARR 预装载）时的更新事件

向下计数

向下计数模式中，计数器从 ARR 值开始递减，到 0 后产生下溢出事件，并且重新从 ARR 开始计数。

软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF（Update Interrupt Flag）中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- ARR 影子寄存器被更新为 GPTIM_ARR 内容
- PSC 影子寄存器被更新为 GPTIM_PSC 内容

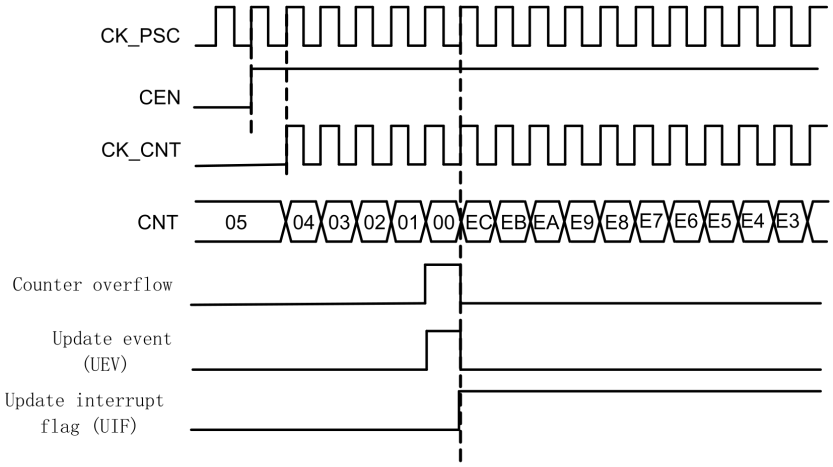


图 21-8 向下计数，内部时钟不分频

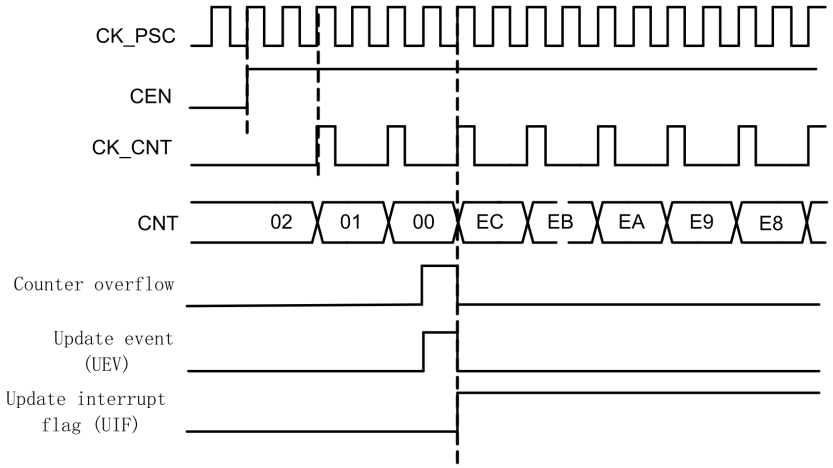


图 21-9 向下计数，内部时钟 2 分频

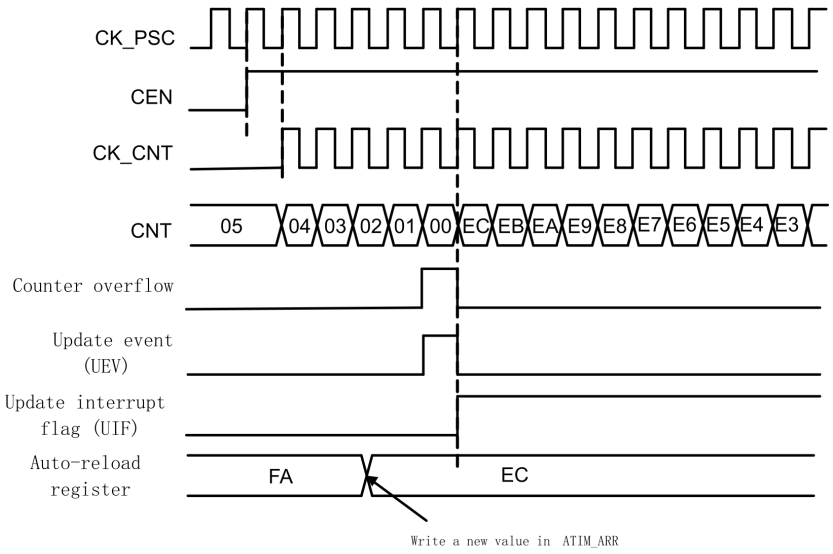


图 21-10 向下计数，不使用重复计数时的更新事件

中心对齐计数

在中心对齐模式下，计数器从 0 开始向上计数，到 ARR-1 产生上溢出事件，然后从 ARR 开始向下计数到 1，产生下溢出事件，再从 0 重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当 CMS!=00 时为中心对齐计数，当 CMS=01 时，输出比较功能仅在向下计数时有效，当 CMS=10 时，输出比较功能仅在向上计数时有效，当 CMS=11 时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR 寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在 overflow 和 underflow 的事件上都会更新 ARR、PSC 的影子寄存器。

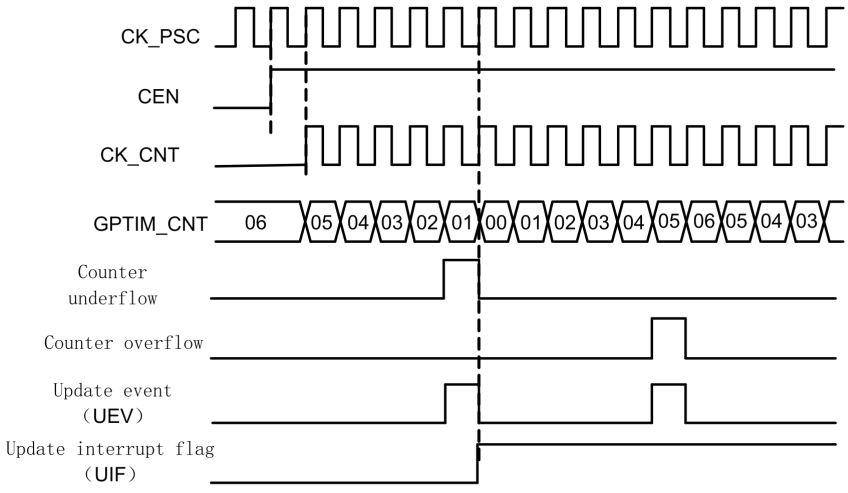


图 21-11 中心对齐计数器时序图，GPTIM_PCS=0，GPTIM_ARR=0x6

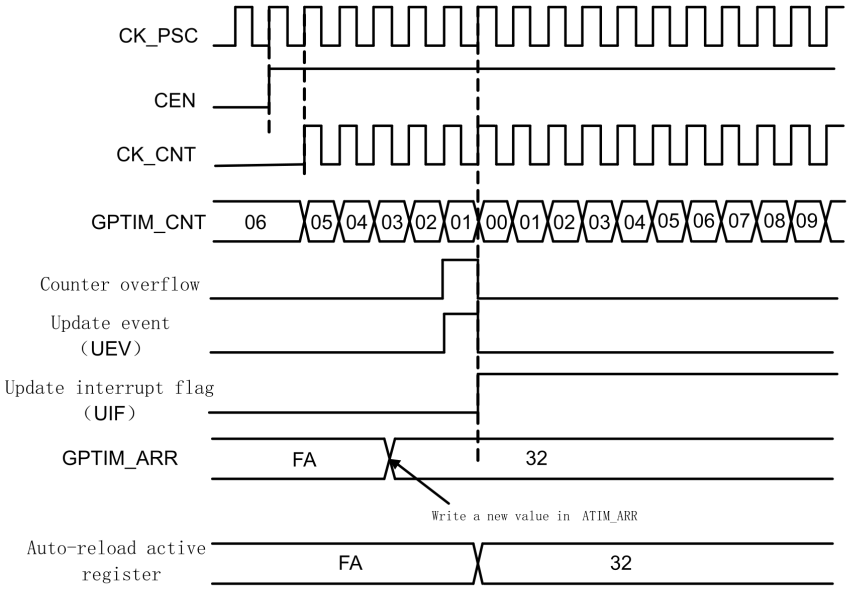


图 21-12 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

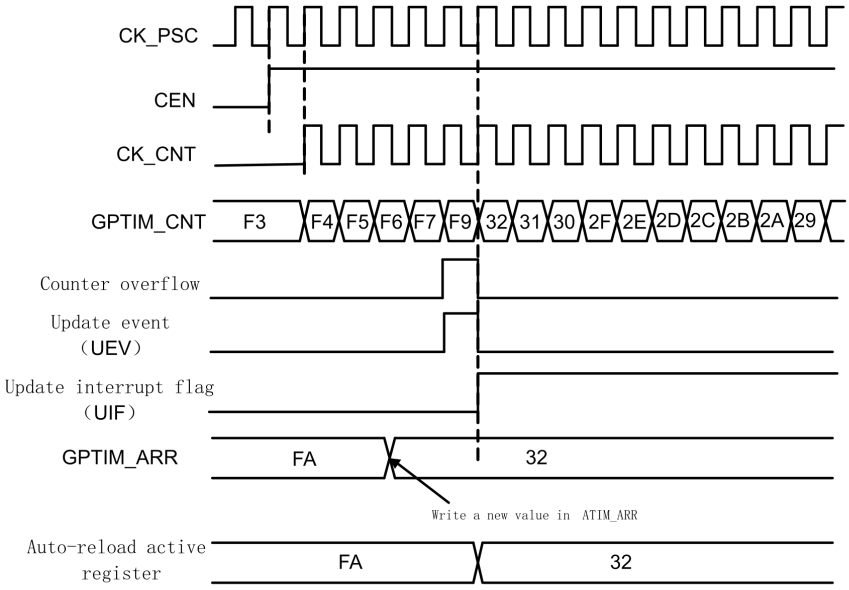


图 21-13 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

21.5.3 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK —- 内部时钟模式
- 外部引脚输入时钟（Tix）—- 外部时钟模式 1
- 外部引脚触发输入（ETR）—- 外部时钟模式 2
- 内部触发（ITRx）— - 使用一个 timer 的触发输出（TGO）作为计数时钟

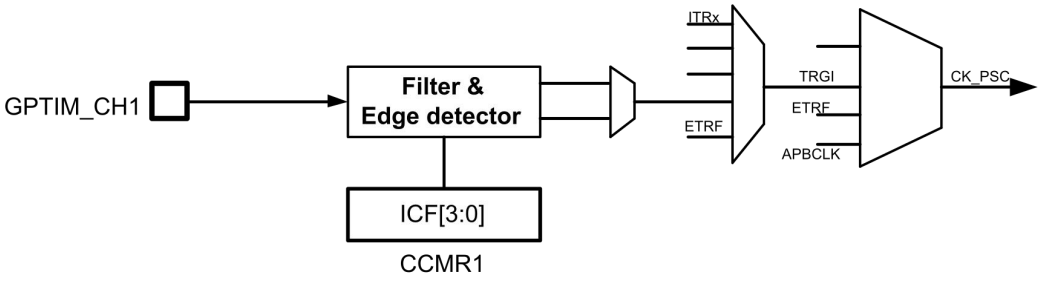


图 21-14 GPTIM 时钟源框图

21.5.3.1 内部时钟模式

内部时钟模式下，禁止从机模式（SMS=000），CEN、DIR、UG 等寄存器位都是软件控制软件操作 UG 寄存器后，update 信号经过 CLK_PSC 同步后，计数器值将被重新初始化。

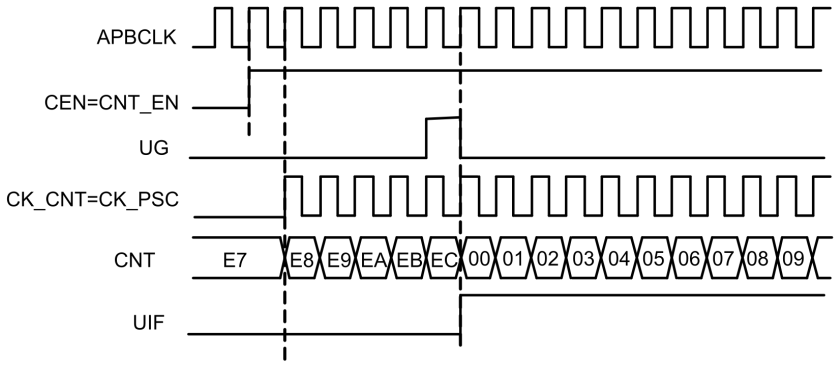


图 21-15 内部时钟源模式，时钟分频因子为 1

21.5.3.2 外部时钟模式 1

此模式下直接使用外部引脚输入信号作为计数时钟，配置 SMS=111，计数边沿可以配置为上升或下降沿。

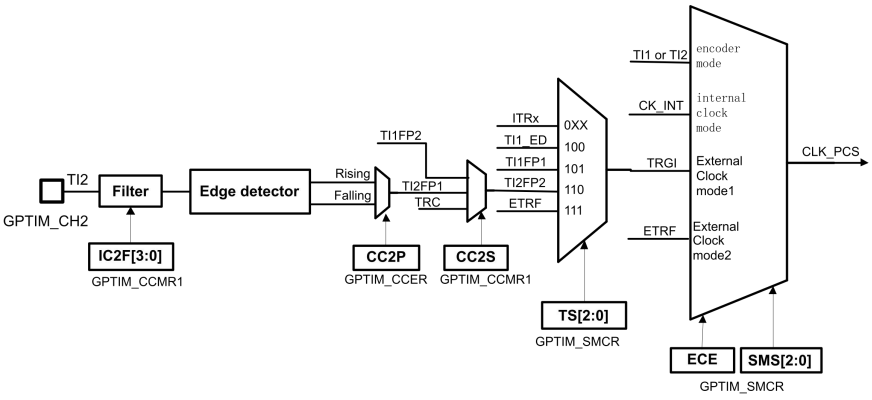


图 21-16 TI2 外部时钟连接例子

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

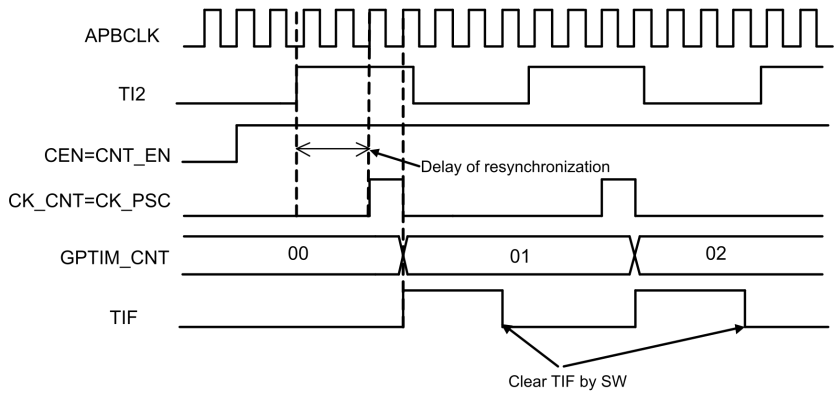


图 21-17 外部时钟模式 1 下的时序

使用外部时钟计数时，仍然要使能 GPTIM 的内部时钟（APBCLK），因为 GPTIM 要使用 APB_CLK 来对外部输入时钟进行同步和滤波。在外部时钟模式 1 下，外部输入时钟首先经过滤波和边沿选择，得到有效的计数沿，作为有效工作时钟（CLK_PSC）输入给预分频模块。

外部时钟同步采用简单的 2 级触发器结构，因此为了避免亚稳态，要求外部输入时钟宽度至少大于 2 个 APB_CLK 周期。

此模式下只有通道 1 和 2 的输入可以用做时钟输入，所需配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH2 功能
- 关闭通道使能，配置 GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC2S=01, IC2 映射到 TI2
- 选择计数有效沿，配置 GPTIM_CCER.CC2P=0，选择上沿或者下沿
- 配置输入滤波时间，配置 GPTIM_CCMR1.IC2F[3:0] (IC2F=0000，不进行输入滤波)
- 使能外部时钟模式 1，配置 GPTIM_SMCR.SMCR=111
- 选择触发输入源，配置 GPTIM_SMCR.TS=110,选定 TI2 作为触发输入源
- 打开通道使能，配置 GPTIM_CCER.CC2E=1
- 使能计数器，配置 GPTIM_CR1.CEN=1

下图是一个典型的外部时钟计数模式 1 的示例：

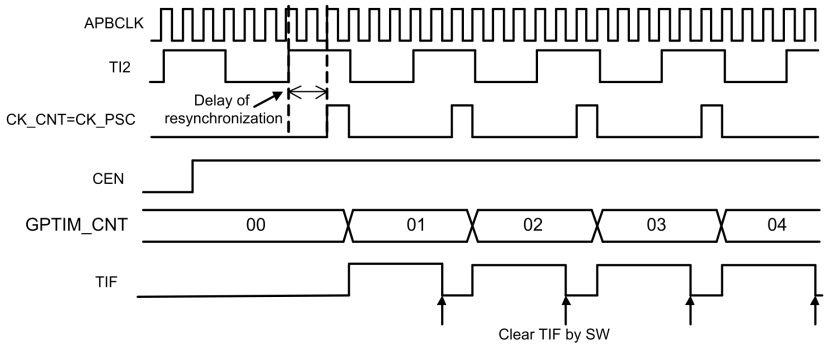


图 21-18 外部时钟模式 1 下的时序

21.5.3.3 外部时钟模式 2

此模式下使用 GPTIM_ETR 管脚输入信号的上升沿或下降沿（不支持双沿）来计数。

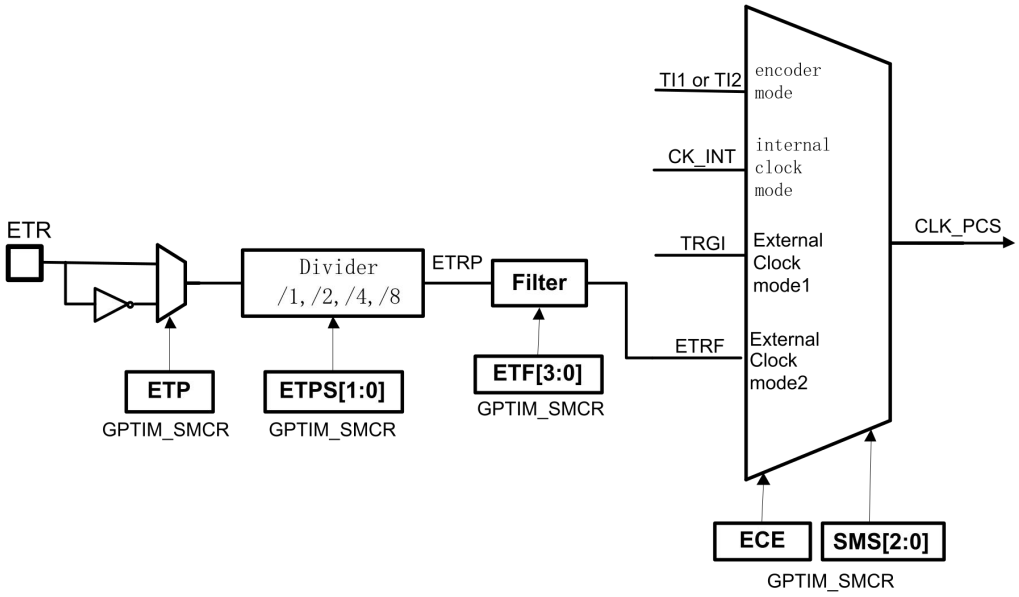


图 21-19 外部触发输入框图

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。

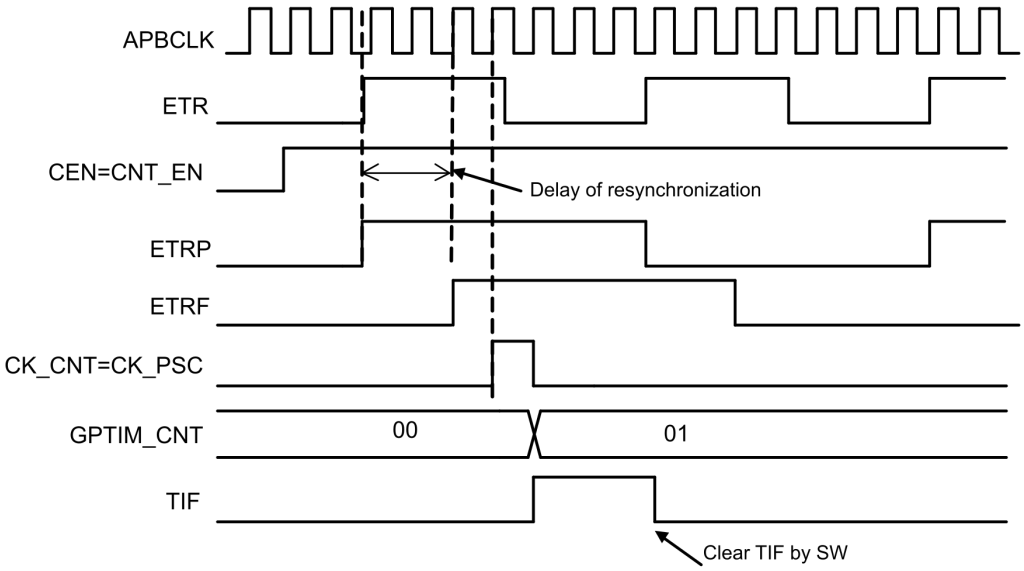


图 21-20 外部时钟模式 2 下的时序 1g

与外部时钟模式 1 的主要差别是，ETR 输入直接被分频后再进行滤波，产生 CK_PSC 时钟，这意味着可以支持 ETR 输入频率高于 APB_CLK 的应用场景，这种情况下，需要首先对 ETR 输入进行预分频，再用于驱动计数器。

此模式所需配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_ETR 功能
- 设置 ETP 进行沿选择，GPTIM_SMCR.ETP=0
- 设置 ETR 分频比，配置 GPTIM_SMCR.ETPS[1:0]=01
- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位 ECE 寄存器，使能外部时钟模式 2,GPTIM_SMCR.ECE=1，GPTIM_SMCR.SMS=000
- 使能计数器，配置 GPTIM_CR1.CEN=1

下图是一个典型的外部时钟模式 2 的示例：

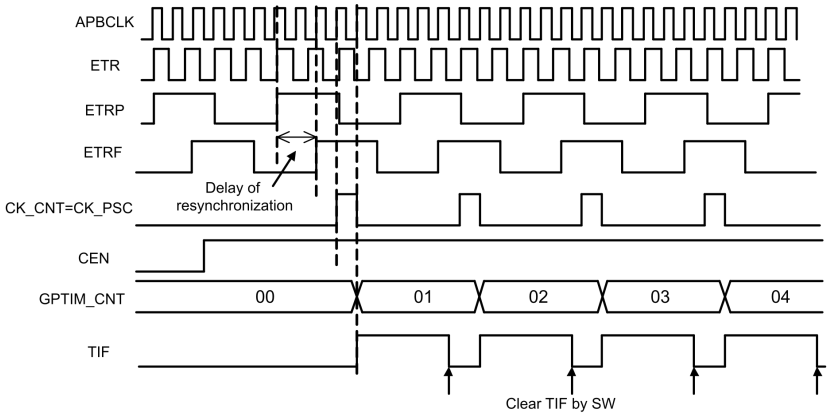


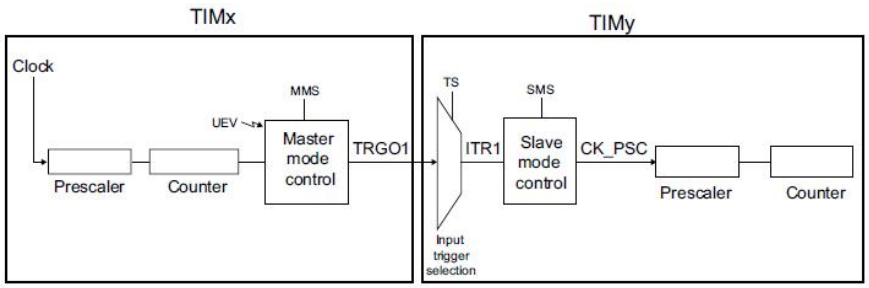
图 21-21 外部时钟模式 2 下的时序 2

在使用外部时钟模式 2 时，仍可以将 GPTIM 配置为 slave 模式：比如使用 ETR 输入计数，同时使用另一个 Timer 的 TRGO 作为触发信号，当触发事件到来时，复位计数器重新开始计数。

21.5.3.4 内部触发模式

GPTIM 支持最大 16 个 ITR 输入，可用于计数触发。每个 ITR 输入支持 16 个内部信号扩展，由 TS 寄存器配置。输入信号源参考”管脚和内部信号连接”章节。

当 ITR 选择为计数触发信号时，GPTIM 计数器将在每个 ITR 信号的高电平计数。通过内部触发模式可以实现 Timer 级联，下图是一个例子：



配置 TIMx 为 master mode 并周期性输出 TRGO 脉冲信号，TIMy 配置为 Slave mode 并将 TIMx 的 TRGO 设置为 ITR；当 TIMx.TRGO 脉冲到来时，TIMy 计数一次。

21.5.4 内部触发信号（ITRx）的捕捉

ITRx 当用于内部信号捕捉时，需要配置 TS 寄存器并将 CCxS 配置为 11，即将 TRC 选为捕捉信号。通过这个方法，Timer 可以捕捉各种芯片内部信号的周期或电平宽度。

其中，作为计数触发的 TRGO 信号，在到达 GPTIM 的 ITRx 之前都被处理成一个 APBCLK 宽度的同步使能信号。而用作周期或宽度捕捉的信号，则无需处理，直接由 GPTIM 进行捕捉。

软件应保证选择正确的信号用于正确的功能，错误的配置将导致完全错误的结果。比如将 ATIM_TRGO 用于宽度捕捉，则结果没有意义。

21.5.5 捕捉/比较通道

GPTIM 包含 4 个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样 Tix 输入并产生滤波后的信号 TixF，然后边沿检测和极性选择产生对应的 TixFPx 信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

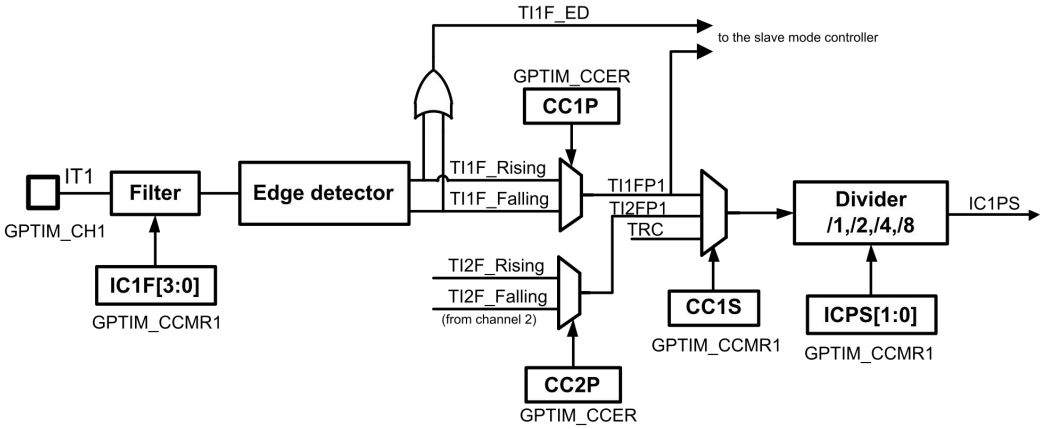


图 21-22 捕获/比较通道(通道 1 输入部分

输出级电路会产生一个输出基准信号 OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。GPTIM 输出通道不支持互补输出。

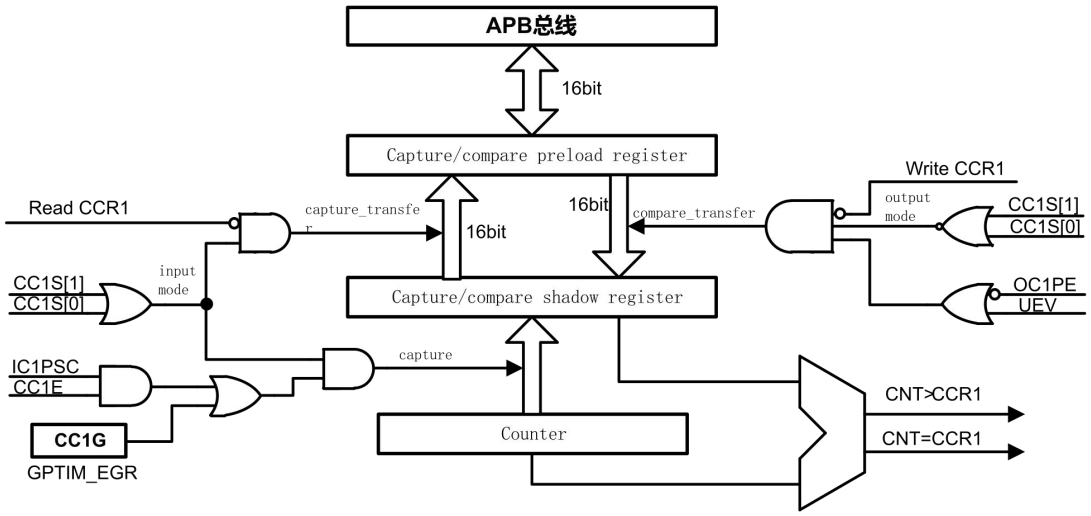


图 21-23 捕获/比较通道 1 的主电路

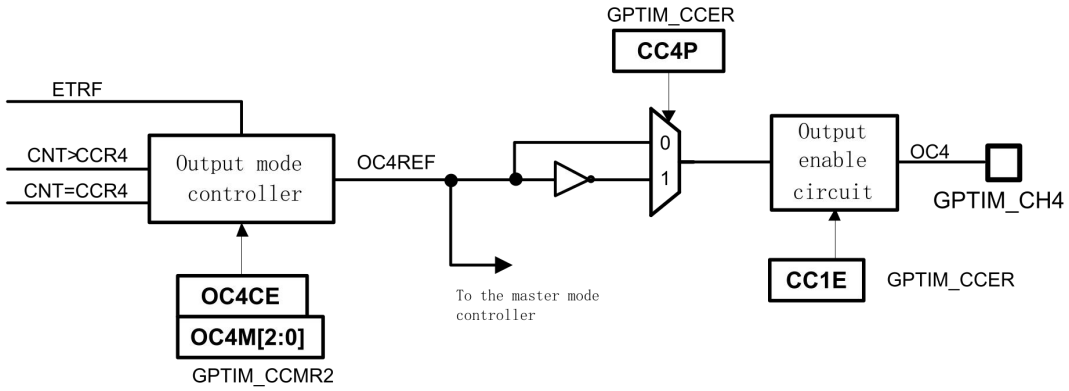


图 21-24 捕获/比较通道的输出部分

捕捉/比较寄存器（CCR）包含 preload 寄存器和 shadow 寄存器，软件读写总是访问 preload 寄存器。在捕捉模式下，捕捉值保存在 shadow 寄存器中并复制到 preload 寄存器。在比较模式下，preload 寄存器的值被拷贝到 shadow 寄存器用来与计数器比较。

21.5.6 输入捕捉模式

当 Icx 信号上出现预期的电平变换，将触发一次 capture，当前计数器值被锁存进 CCR，与此同时，CcxIF 中断标志置位，并且可以触发对应的中断或者 DMA 请求。如果一个捕捉事件在 CcxIF 为高的情况下出现，则捕捉数据冲突标志（CcxOF, Over-Capture）置位（CCR 中上次捕捉值被覆盖）。CcxIF 可以由软件清零，或者通过读取 CCR 寄存器自动清零。CcxOF 标志通过软件写 1 清零。

通过两个或更多通道配合，可以实现 PWM 信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从 TI1 引脚输入，芯片内部将滤波后的信号取上升沿得到 TI1FP1，将滤波后的信号取下降沿得到 TI1FP2，将 TI1FP1 输入给捕捉通道 1，将 TI1FP2 输入给捕捉通道 2，即可实现通道 1 对输入信号上升沿捕捉，同时通道 2 对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过 CCR1 和 CCR2 寄存器的值，即可计算输入信号的周期和占空比。

实现在 TI1 输入的上升沿捕获计数器的值到 GPTIM_CCR1 寄存器，配置步骤如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01, IC1 映射到 TI1
- 选择计数有效沿，配置 GPTIM_CCER.CC1P，选择上沿或者下沿
- 配置输入滤波时间，配置 GPTIM_CCMR1.IC1F[3:0]
- 配置输入预分频器，配置 GPTIM_CCMR1.IC1PS[1:0]
- 打开通道使能，配置 GPTIM_CCER.CC1E=1

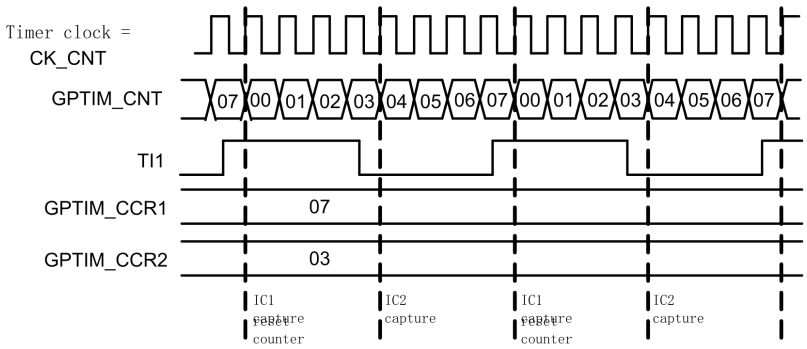


图 21-25 PWM 输入捕获模式时序

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 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0 确保之后通道配置成功
- 选择输入通道，两个通道 IC1,IC2 被映射到同一个 TI1 输入口，配置 GPTIM_CCMR1.CC1S=01, GPTIM_CCMR1.CC2S=10
- 选择计数有效沿，两个通道 IC1,IC2 有效沿极性相反，配置 GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=1
- 配置输入滤波时间，配置 GPTIM_CCMR1.IC1F[3:0]，GPTIM_CCMR1.IC2F[3:0]
- 配置输入预分频器，配置 GPTIM_CCMR1.IC1PS[1:0]，GPTIM_CCMR1.IC2PS[1:0]
- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=101
- 设定从模式控制器为复位模式，配置 GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置 GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

21.5.7 软件 Force 输出

在比较输出模式下，软件可以直接将 OCxREF force 成特定电平，而独立于 CCR 和计数器的比较结果。

软件通过写 OcxM=101 寄存器，可以直接将 OCxREF 强制为有效（OCxREF 固定为高有效），通过写 OcxM=100 可以直接将 OCxREF 强制为无效（低电平）。但是软件 force 操作不会取消比较过程，CCR 和计数器的比较还会一直进行。

21.5.8 输出比较模式

输出比较模式下，当 CCR 与计数器值相等，OCxREF 可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA 请求可以发送。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。

使用步骤：

1. 选择计数时钟（内部、外部、预分频等）
2. 向 ARR 和 CCR 寄存器写入期望数据
3. 根据需要设置中断使能和 DMA 使能
4. 选择输出模式

5. 使能计数器

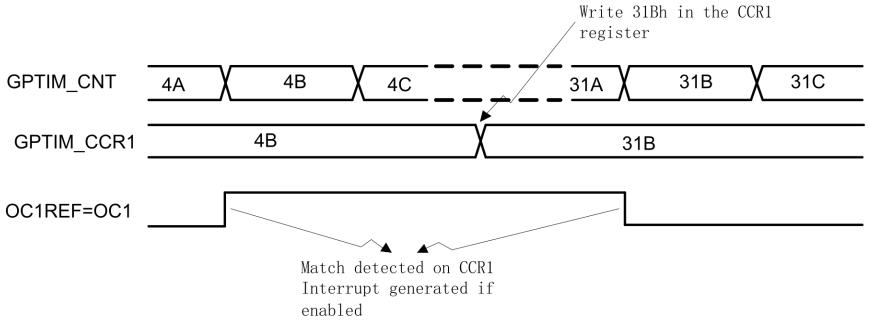


图 21-26 输出比较模式，翻转 OC1

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。

21.5.9 PWM 输出

PWM 模式可以输出脉宽调制信号，其周期由 ARR 寄存器决定，占空比由 CCR 寄存器决定。

输出信号的极性可以由 CCxP 寄存器配置。PWM 模式工作中，CNT 和 CCR 实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM 输出也支持边缘对齐和中央对齐模式。

PWM 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

PWM 边缘对齐模式

在向上计数的情况下，配置为 PWM 模式 1 时，OCxREF 信号在 CNT<CCR 时为高电平，否则为低电平。如果 CCR 值大于 ARR 值，则 OCxREF 被固定为 1；如果 CCR 为 0 则 OCxREF 被固定为 0.

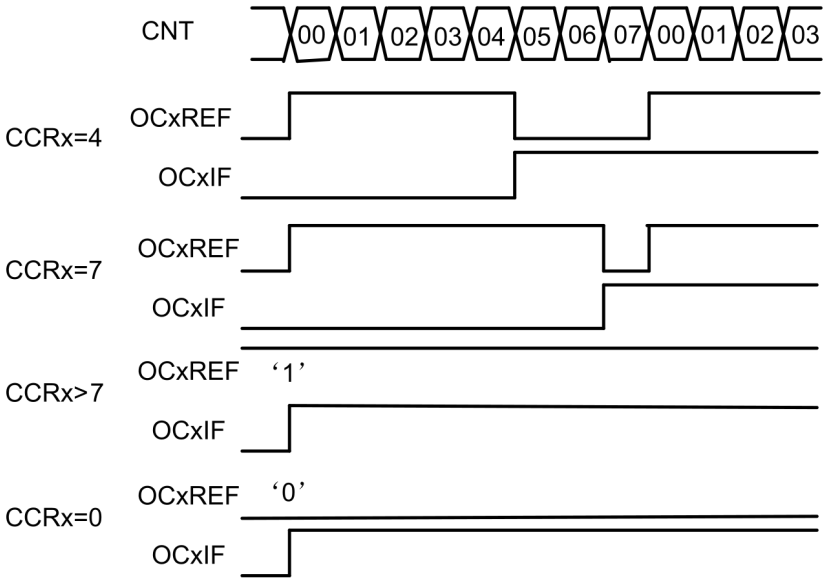


图 21-27 边沿对齐的 PWM 波形(ARR=7, PWM mode 1)

PWM 中央对齐模式

OCxREF 电平定义与边缘对齐模式相同。下图是一个示例：

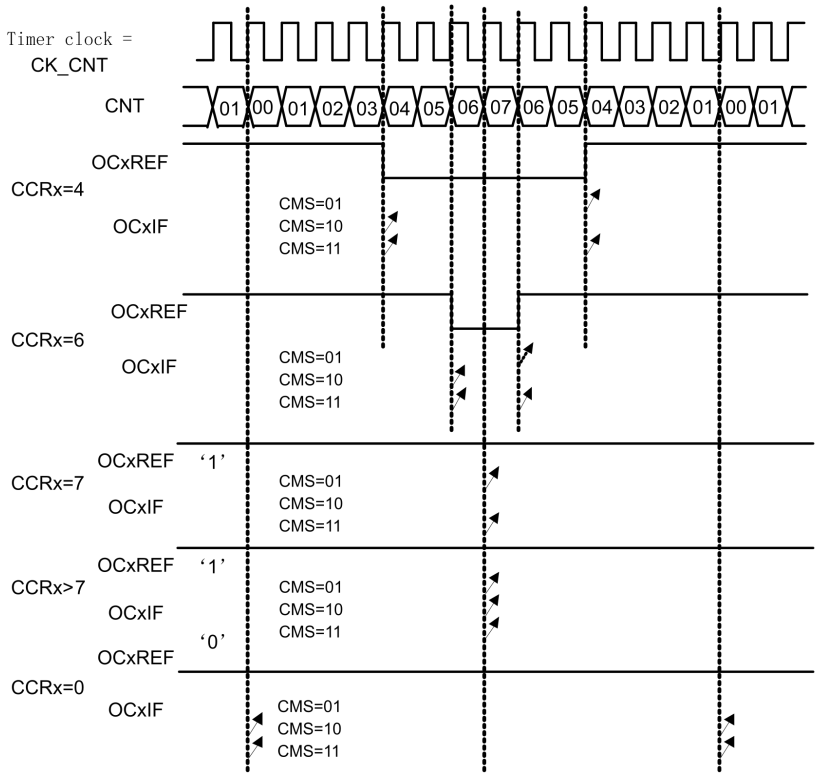


图 21-28 中央对齐的 PWM 波形(APR=7, PWM mode 1)

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。

21.5.10 非对称 PWM 模式

非对称 PWM 用于产生非对称的中心对齐 PWM 波形，需要同时使用两组 CCR 寄存器，由 ARR 寄存器决定 PWM 波形周期，一组 CCR 寄存器决定向上计数周期内的 PWM 信号变化点，另一组 CCR 寄存器决定向下计数周期内的 PWM 信号变化点。

使用非对称 PWM 模式时，OC1REF(OC2REF) 由 CCR1 和 CCR2 寄存器控制，OC3REF(OC4REF)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，OC3REF 为 PWM mode 2

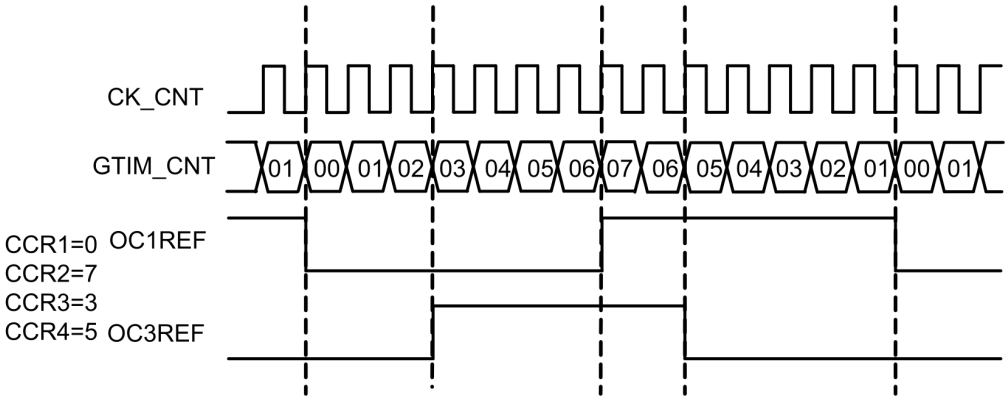


图 21-29 非对称 PWM 输出

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。

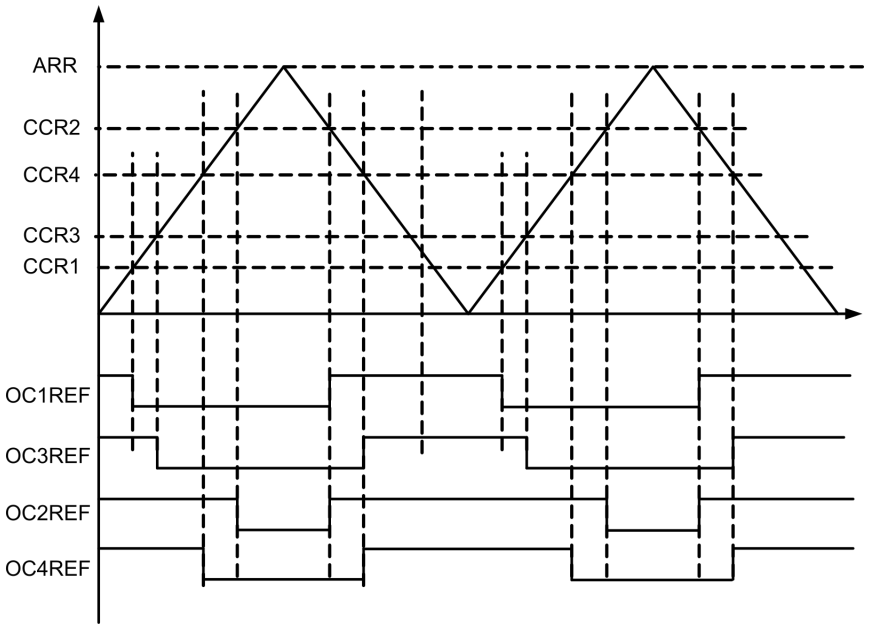


图 21-30 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 1）

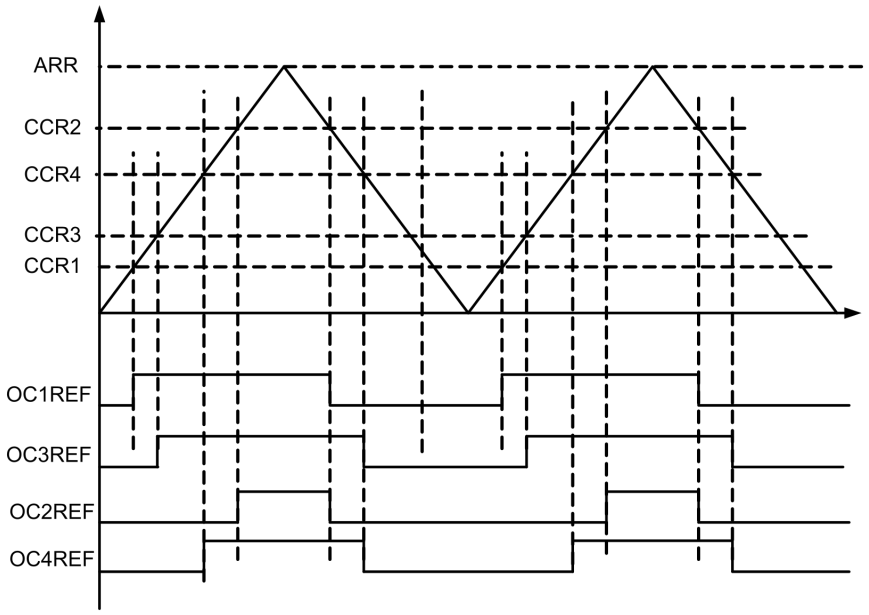


图 21-31 通道 1/3 非对称 PWM 输出，通道 2/4 常规输出（all PWM mode 2）

21.5.11 组合 PWM 模式

通道 1 和通道 3 可以配置为组合 PWM 模式，此模式下通道 1 的 PWM 波形输出由 OC1REF 和 OC2REF 组合产生，通道 3 的 PWM 波形输出由 OC3REF 和 OC4REF 组合产生，组合方式是 OR 或者 AND，根据 OCxM 寄存器选择组合模式 1（OR）或者组合模式 2（AND）。

下图是组合 PWM 的例子：

- 通道 1 为组合 PWM 模式 2
- 通道 2 为 PWM 模式 1
- 通道 3 为组合 PWM 模式 1
- 通道 4 为 PWM 模式 2

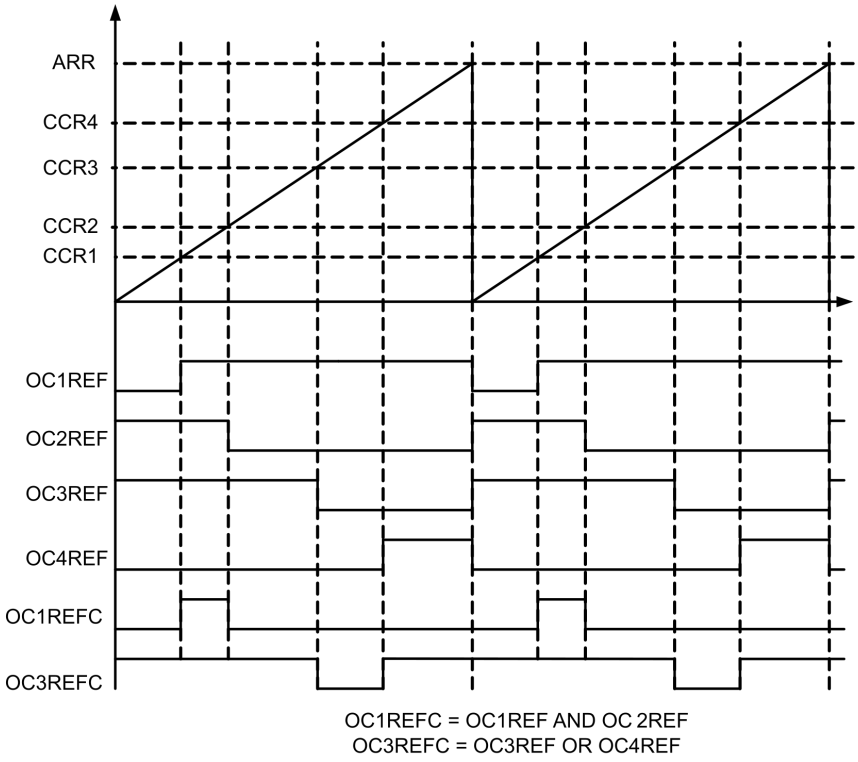


图 21-32 通道 1/3 组合 PWM 输出

21.5.12 单脉冲输出

单脉冲输出是比较输出模式的特殊情况，允许用户在某个事件发生后，经过可编程的延迟，输出一个可编程宽度的脉冲信号。

与其他输出模式不同的是，在下一次 update event 到来时，计数器会自动停止。只有当 CCR 和计数器初值不同时，脉冲才有可能正确输出。在向上计数时，要求 CNT < CCR ≤ ARR，在向下计数时，要求 CNT > CCR

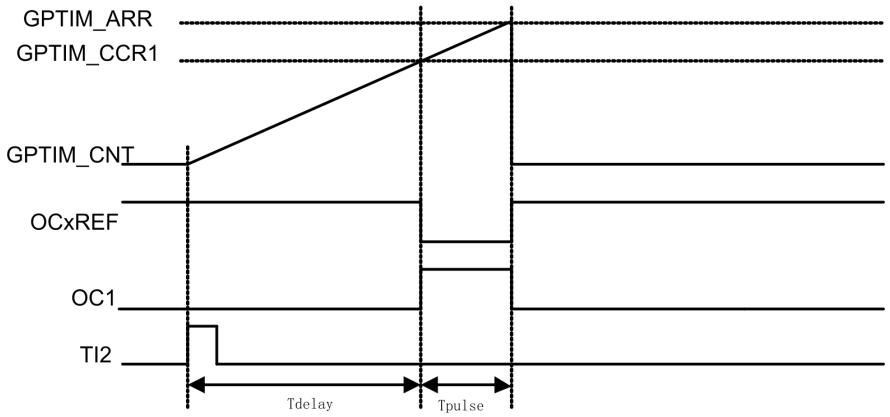


图 21-33 单脉冲模式的例子

上图是以 TI2 输入为计数器触发信号，计数值等于 CCR 后 OCxREF 输出低电平，计数到 ARR 后 OCxREF 回到高电平，并且计数器回滚到 0，停止计数。

实现上述功能 TI2 作为输入触发的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH2 功能
- 关闭通道使能，配置 GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC2P=0
- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=110，TI2FP2 作为 TRGI
- 设定从模式控制器为触发模式，配置 GPTIM_SMCR.SMS[2:0]=110,TI2FP2 用来启动计数器
- 打开通道使能，配置 GPTIM_CCER.CC2E=1

实现上述功能 OC1 作为输出的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0，确保之后通道配置成功
- 输出通道，配置 GPTIM_CCMR1.CC1S=00
- 选择计数有效沿，配置 GPTIM_ CCMR1.OC1M=111,PWM 模式 2
- 打开通道使能，配置 GPTIM_CCER.CC1E=1

OPM 波形产生时基的特殊设置：

- GPTIM_CCR1 的值决定了 Tdelay
- GPTIM_ARR 和 GPTIM_CCR1 的差值决定了 Tpulse（GPTIM_ARR-GPTIM_CCR1）
- 设置为单脉冲模式，配置 GPTIM_CR1.OPM=1

21.5.13 外部事件清除 OCxREF

OCxREF 的有效状态未高电平，通过对外部 ETR 引脚施加高电平，可以直接拉低 OCxREF，直到下一次 update event。此功能仅在输出比较和 PWM 模式下有效，无法在软件 force 模式下起作用。使能此功能需要将 OcxCE 置 1.

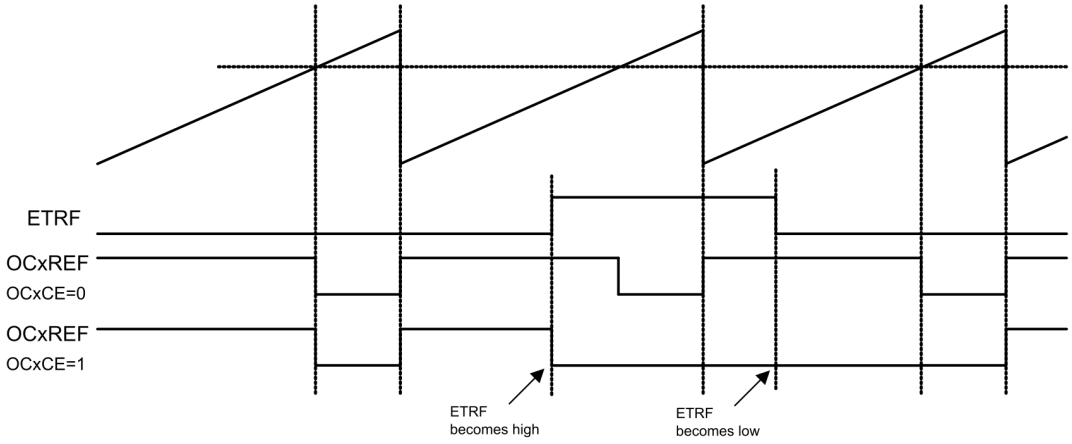


图 21-34 ETR 信号清除 GPTIM 的 OCxREF

21.5.14 编码器接口模式（encoder interface）

21.5.14.1 正交编码器

GPTIM 使用两个输入信号 TI1 和 TI2 来与正交编码器连接，输入信号首先经过数字滤波和极性选择，定时器再根据滤波和极性选择后的信号 TI1FP1 和 TI2FP2 的边沿进行计数。GPTIM 根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值，同时硬件自动更新 DIR 寄存器，向软件指示当前计数方向。根据 SMS 寄存器的配置，GPTIM 可以只对 TI1 的边沿计数、只对 TI2 的边沿计数、或者同时对 TI1 和 TI2 的边沿计数。

编码器接口模式启动后，计数器根据输入信号在 0 和 ARR 范围内连续计数，因此使能定时器之前必须配置 ARR 寄存器。

下表是计数方式与两路输入信号之间的关系：

表 21-9 encoder interface 计数方式（CC1P=CC2P=0）

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 处计数 x1 mode	1110	高	递减	递增	不计数	不计数
		低	不计数	不计数	不计数	不计数
仅在 TI2 处计数 x1 mode	1111	高	不计数	不计数	递增	递减
		低	不计数	不计数	不计数	不计数

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。

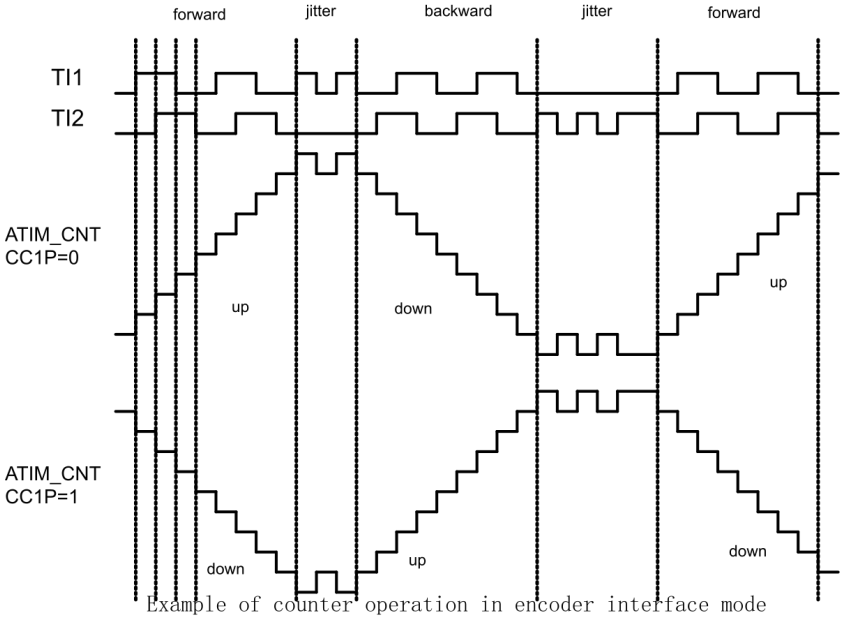


图 21-35 编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1，GPTIM_CH2 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0，GPTIM_CCER.CC2E=0，确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01，GPTIM_CCMR1.CC2S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC1P=0，GPTIM_CCER.CC2P=0
- 设定从模式控制器为编码模式 3，配置 GPTIM_SMCR.SMS[3:0]=0011
- 打开通道使能，配置 GPTIM_CCER.CC1E=1，GPTIM_CCER.CC2E=1

编码器接口支持 x4, x2, x1 计数模式，其中 x4 模式在 TI1 和 TI2 的每个边沿计数，x2 模式只在 TI1 或者 TI2 的边沿计数，x1 模式则只在 TI1 或者 TI2 的某个边沿上计数（由 CCxP 选择有效沿）。

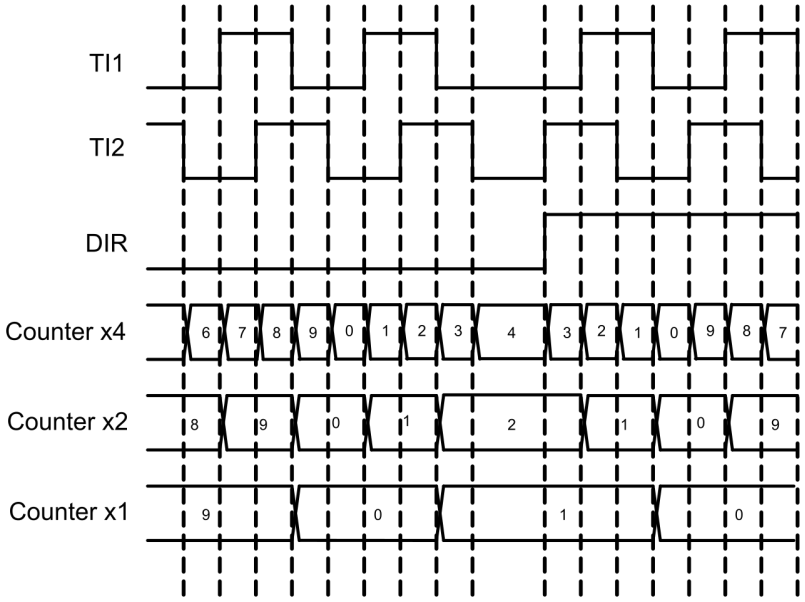


图 21-36 正交编码器不同边沿计数模式示例

当 GPTIM_CR1 寄存器中的 UIFREMAP 位置位时，UIF 中断标志寄存器被实时复制到 GPTIM_CNT[31]，供软件进行原子读取。

21.5.14.2 时钟方向编码器模式

除了正交编码器，另一种编码器提供”时钟”和”方向”两个输出信号。为支持这种编码器模式，需要将 SMS 配置为 1010 或 1011，将时钟信号连接到 TI2，将方向信号连接到 TI1。

- SMS[3:0]=1010：x2 模式，计数器在 TI2 输入信号的上升沿和下降沿都计数
- SMS[3:0]=1011：x1 模式，计数器在 TI2 上升或者下降沿计数，由 CC2P 寄存器定义有效沿（CC2P=0：上升沿计数，CC2P=1：下降沿计数）

计数器计数方向由 TI1 输入的”方向”信号和 CC1P 寄存器决定。

- CC1P=0：TI1 为高电平时向上计数，TI1 为低电平时向下计数
- CC1P=1：TI1 为高电平时向下计数，TI1 为低电平时向上计数

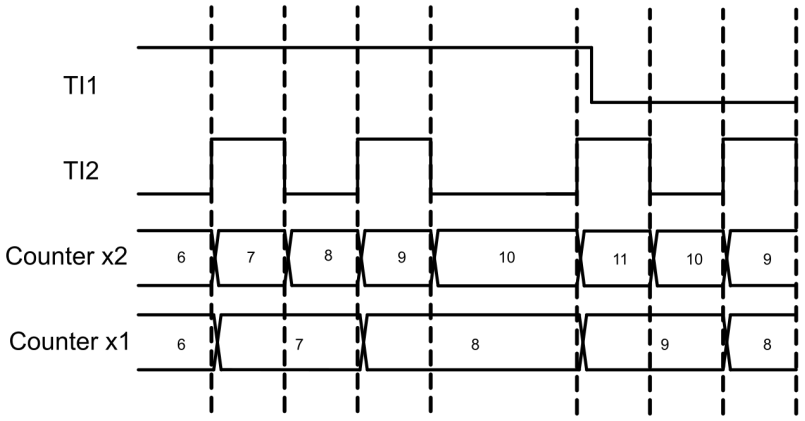


图 21-37 时钟方向编码器示例

21.5.14.3 带有方向信息的时钟编码器模式

这种编码器用时钟信号调制方向信息，输出两个信号，任意时刻只有其中一个信号带有时钟。

- SMS[3:0]=1100：x2 模式，计数器在输入信号的上升沿和下降沿都计数。CC1P 和 CC2P 用于定义时钟空闲状态，0 表示高电平空闲，1 表示低电平空闲
- SMS[3:0]=1101：x1 模式，计数器在输入信号的上升或者下降沿计数，由 CCxP 寄存器定义有效沿（CCxP=0：下降沿计数，CCxP=1：上升沿计数）

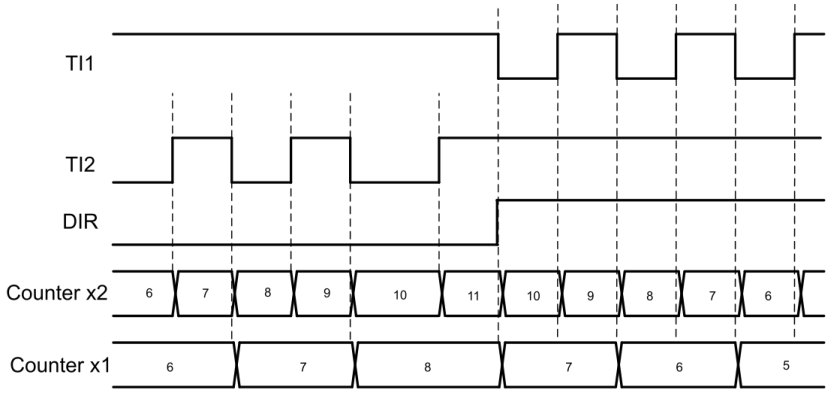


图 21-38 带方向的时钟编码器（CC1P=CC2P=0）

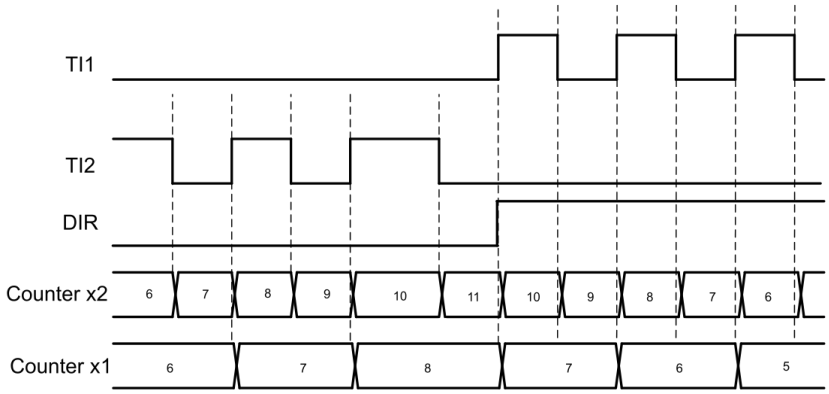


图 21-39 带方向的时钟编码器（CC1P=CC2P=1）

下表是计数方式与两路输入信号之间的关系：

表 21-10 directional clock mode 计数方式

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	1100	高	不计数	不计数	不计数	不计数

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
x2 mode		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

21.5.14.4 Encoder index（Z 相输入）

正交编码器除了输出 2 路正交编码信号之外，通常还会输出 1 个周期归零指示信号（index）。这个零位信号可以用来复位计数器，需要将 index 信号连接到 GPTIM 的 ETR 输入引脚。

Index 功能通过 GPTIM_ECR 寄存器中的 IE 寄存器使能，并且尽在 encoder 模式下有效，即 SMS 必须为 0001/0010/0011/1010/1011/1100/1101/1110/1111

常见的编码器输出的零位信号有如下格式：

- Gated with A and B
- Gated with A or gated with B
- Ungated

不同零位信号的示意图如下：

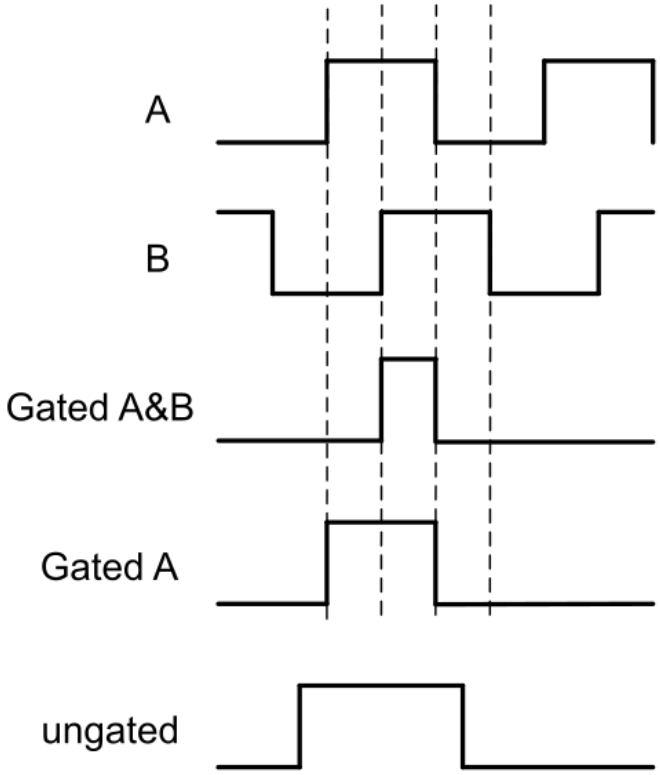


图 21-40 编码器 index 定义

使用中将编码器 A 相连接到 TI1，B 相连接到 TI2，Z 相连接到 ETR。当 GPTIM 检测到 index 输入后，产生 index 事件，执行对计数器的特定操作，注意这个操作是要被通道到特定状态上的，而不是异步即时发生的。

GPTIM 总是在检测 index 输入信号，然而检测到 index 后，触发计数器清零或置位的 index 事件将被同步到特定 AB 相状态上执行，通过 GPTIM_ECR 中的 IPOS[1:0]寄存器来定义 index 事件被同步执行的 AB 相位段。根据计数器计数方向，硬件电路采取不同的 index 事件执行策略：

- 向上计数时（DIR=0），index 事件触发计数器清零
- 向下计数时（DIR=1），index 事件触发计数器回到 ARR

IPOS 寄存器可以配置计数器在检测到 index 后，在什么 AB 相位下执行 index 事件，比如 IPOS=11，计数器仅在 A=1 & B=1 的条件下执行 index 事件（虽然有可能在 11 状态之前就检测到 index）。

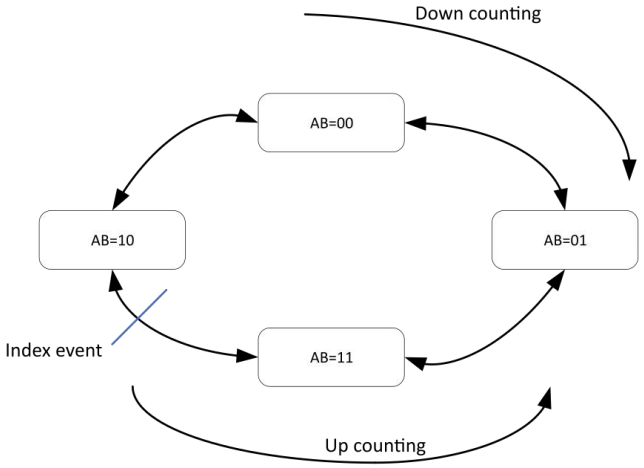


图 21-41 IPOS[1:0]=11 时的 index 识别

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。

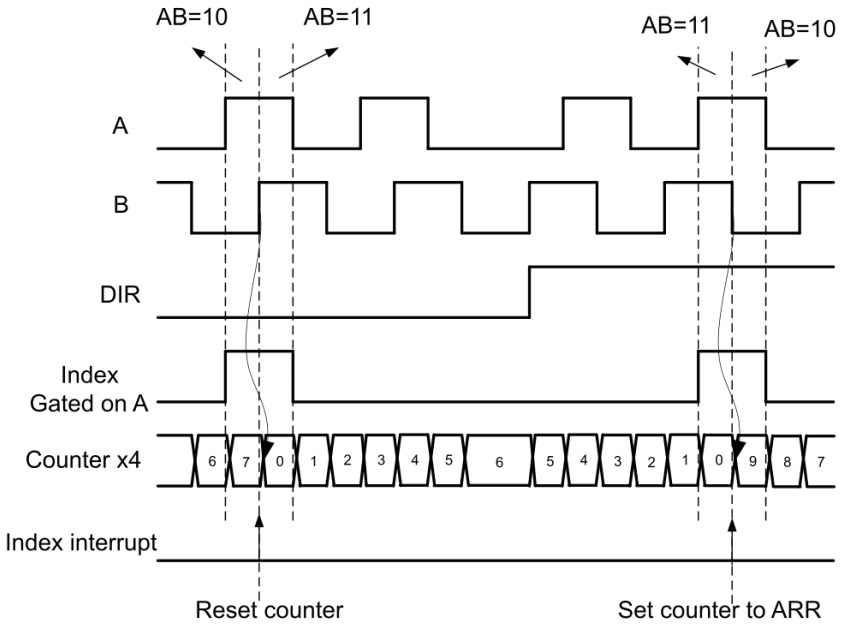


图 21-42 IPOS[1:0]=11, index gated on A, ARR=9

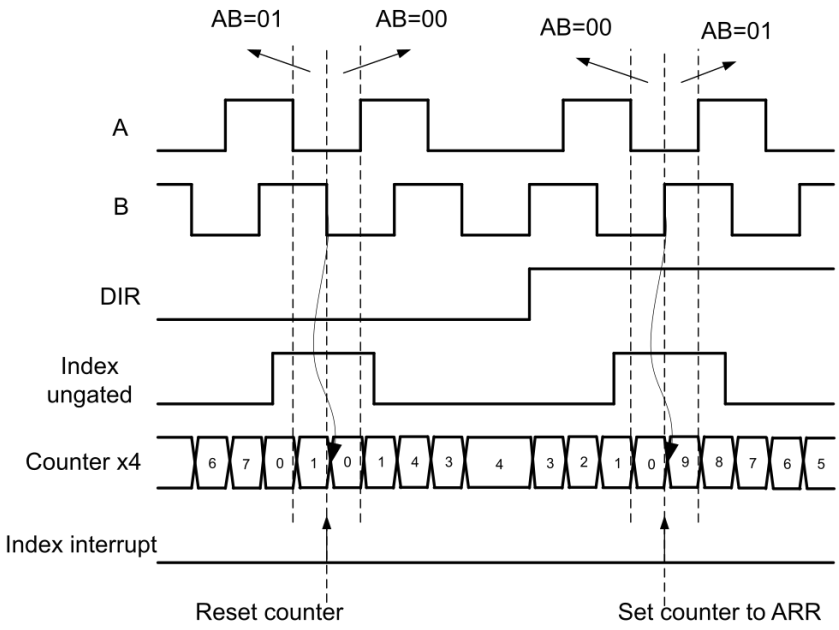


图 21-43 IPOS[1:0]=00, index ungated, ARR=9

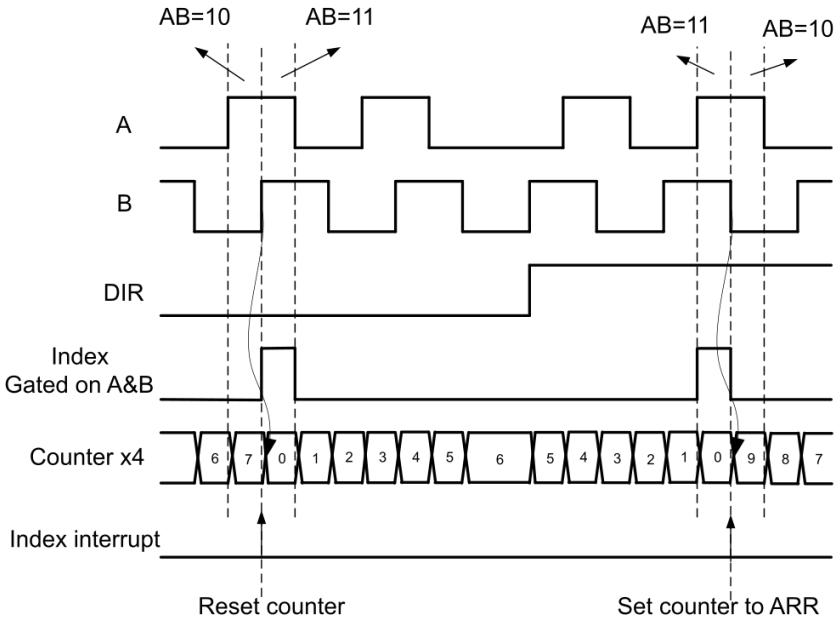


图 21-44 IPOS[1:0]=11, index gated on A&B, ARR=9

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：

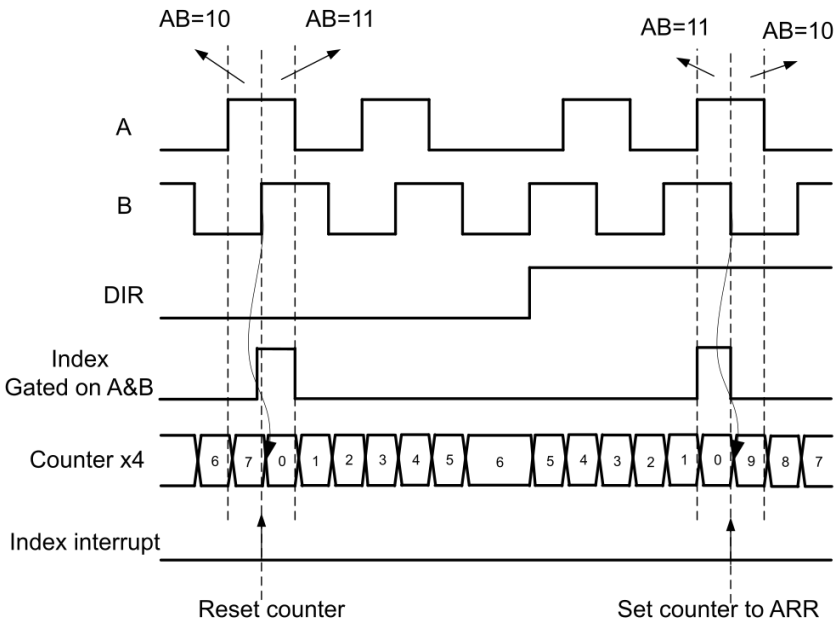


图 21-45 IPOS[1:0]=11, index gated on A&B, ARR=9, index leading channel B

下图中则是 index 滞后 B 相输入，或者说出现一个窄 Z 相信号：

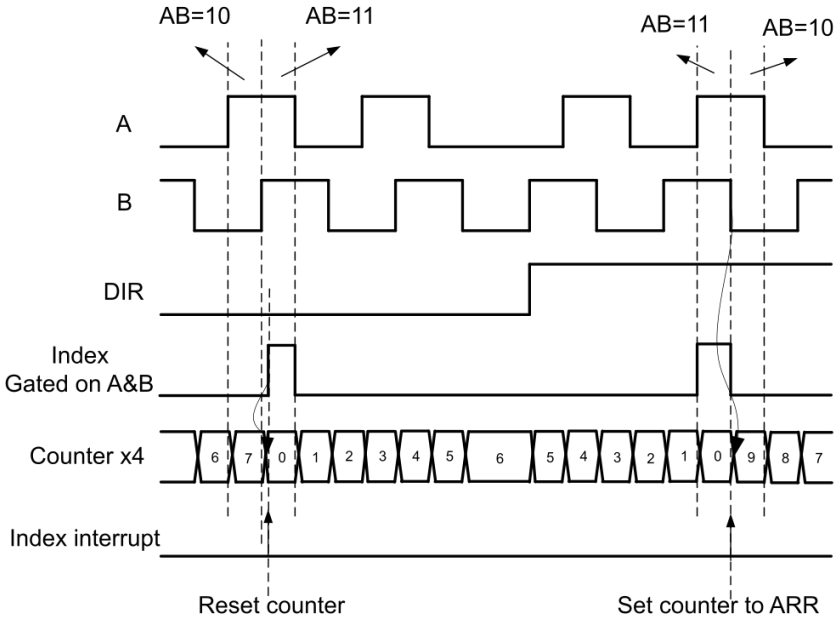


图 21-46 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging channel B

上图中 index lagging 的区域放大图如下：

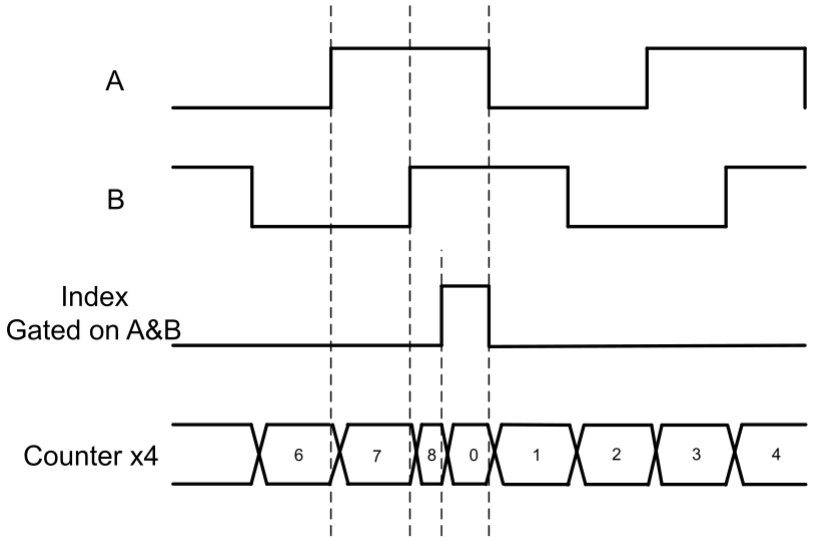


图 21-47 IPOS[1:0]=11, index gated on A&B, ARR=9, index lagging B, zoomed

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）：

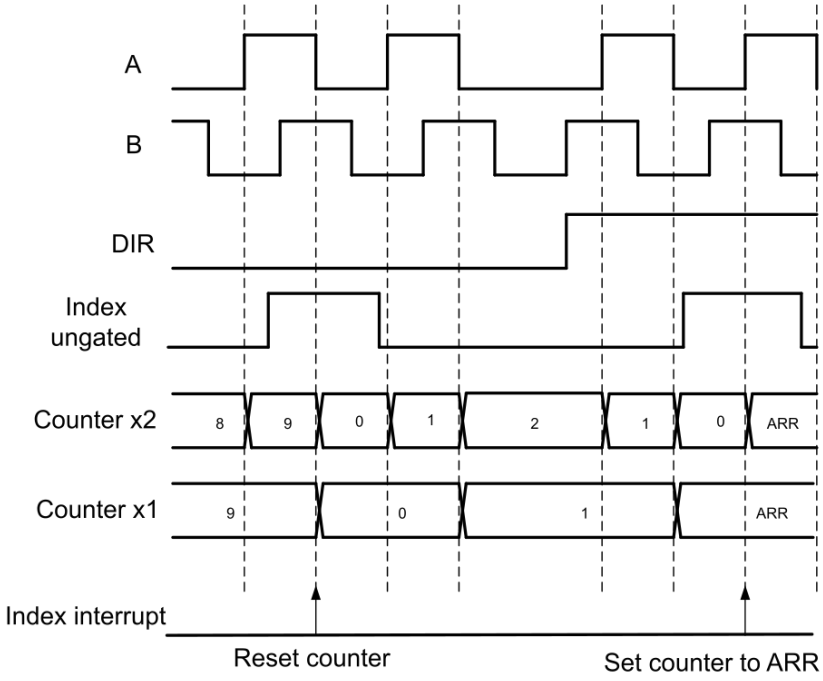


图 21-48 IPOS[1:0]=01, ungated index

Index 的方向敏感性

可以通过 GPTIM_ECR 寄存器中的 IDIR[1:0]位配置定时器只在特定计数方向上响应 index 输入。

- IDIR=00：向上计数和向下计数时 index 都有效
- IDIR=01：向上计数时 index 有效
- IDIR=10：向下计数时 index 有效

注意：在时钟方向编码器和带方向信息的时钟编码器模式下，IDIR 无效。

只响应第一个 index

通过置位 FIDX 寄存器，可以使 GPTIM 只响应第一个输入的 index 信号，后续 index 将被忽略，直到软件清零 FIDX。

21.5.14.5 非正交编码模式下的 index 处理

当 GPTIM 使用编码器接口模式，但是工作类型为非正交编码器，此时需要配置 index 信号在输入时钟信号的高电平处检测，或者在输入时钟信号的低电平处检测。通过 IPOS[0]寄存器完成这个配置。

- IPOS[0]=0：在输入时钟低电平处检测 index
- IPOS[0]=1：在输入时钟高电平处检测 index

IPOS[1]此时不起作用。

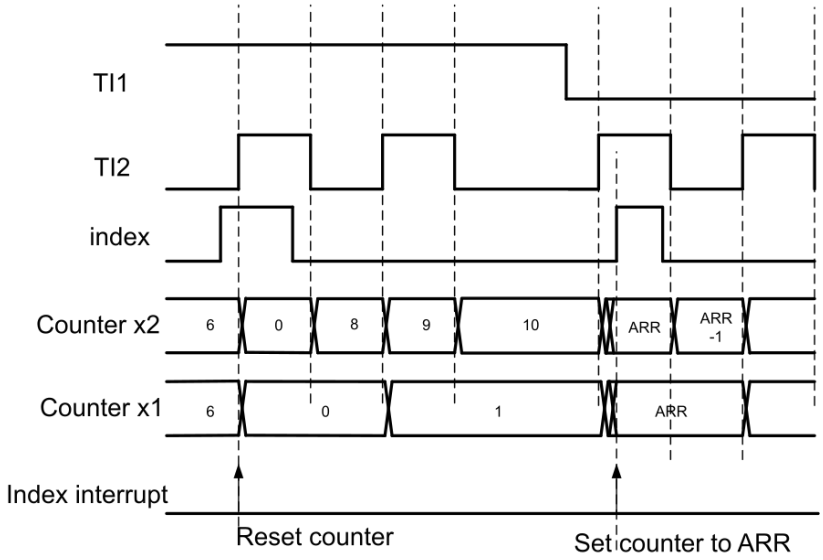


图 21-49 IPOS[0]=1, clock+direction mode

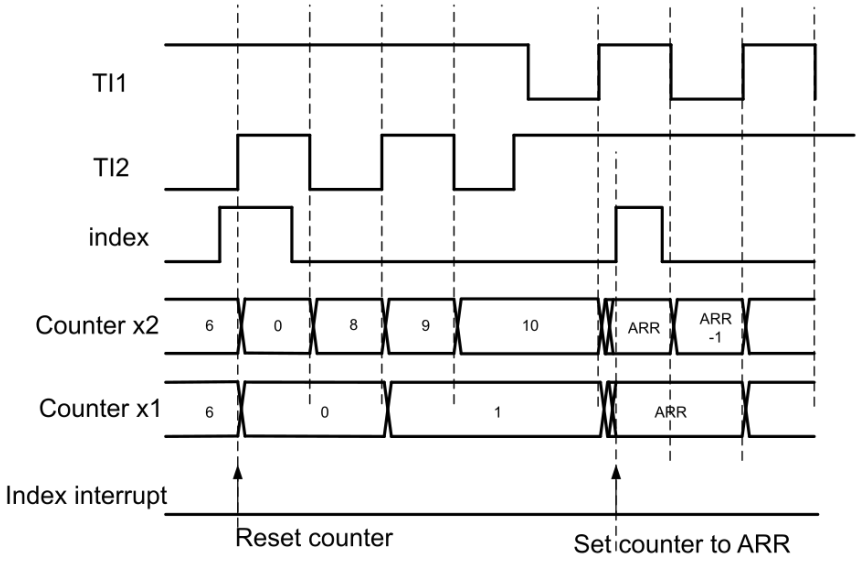


图 21-50 IPOS[0]=1, directional clock mode

21.5.14.6 正交编码模式错误管理

正交编码器输出的 2 路正交信号总是以格雷码形式变化，即相邻状态只会变化 1bit。如果出现了 2bit 状态变化，则表示编码错误，此时 TERRIF 标志置位，并可以产生中断。

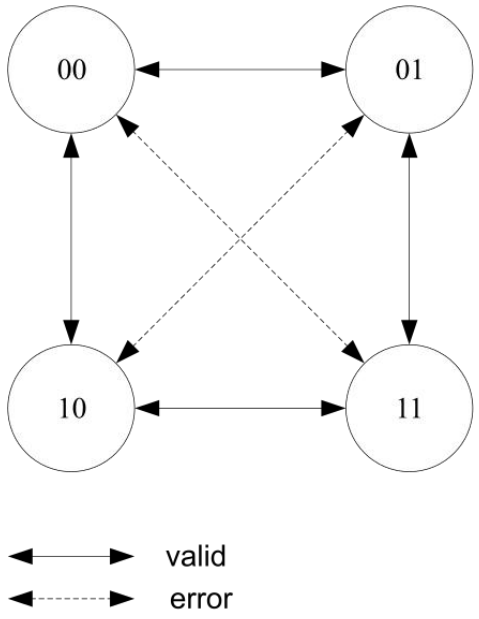


图 21-51 正交编码状态转移

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。

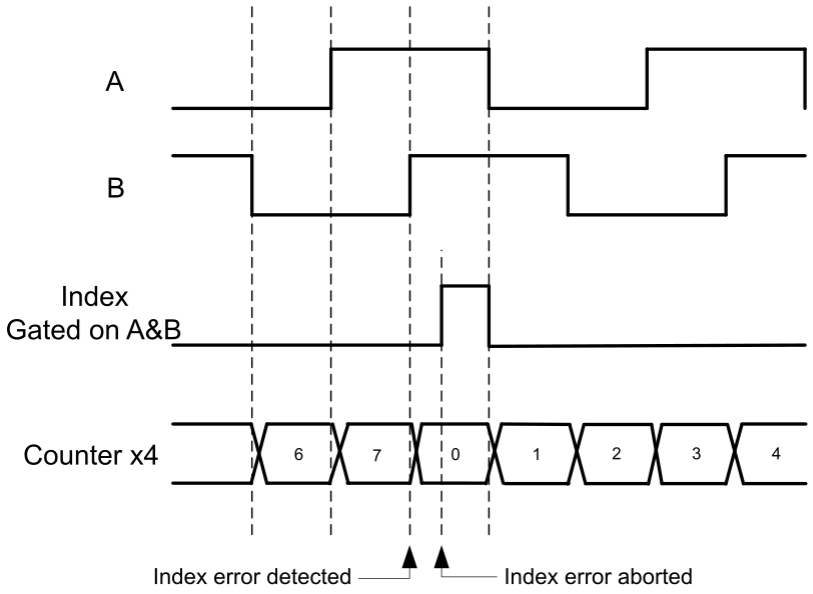


图 21-52 index 错误撤销

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。

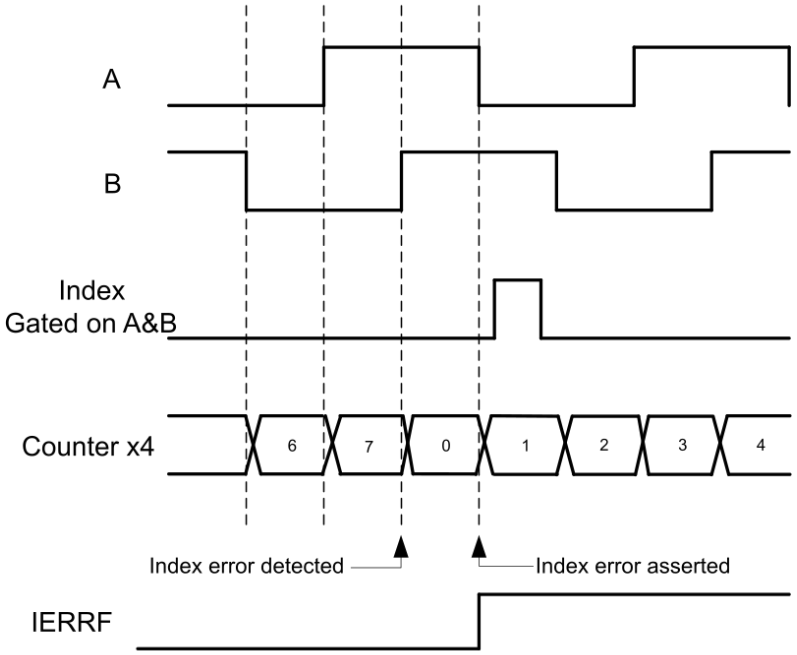


图 21-53 index 错误置位

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。

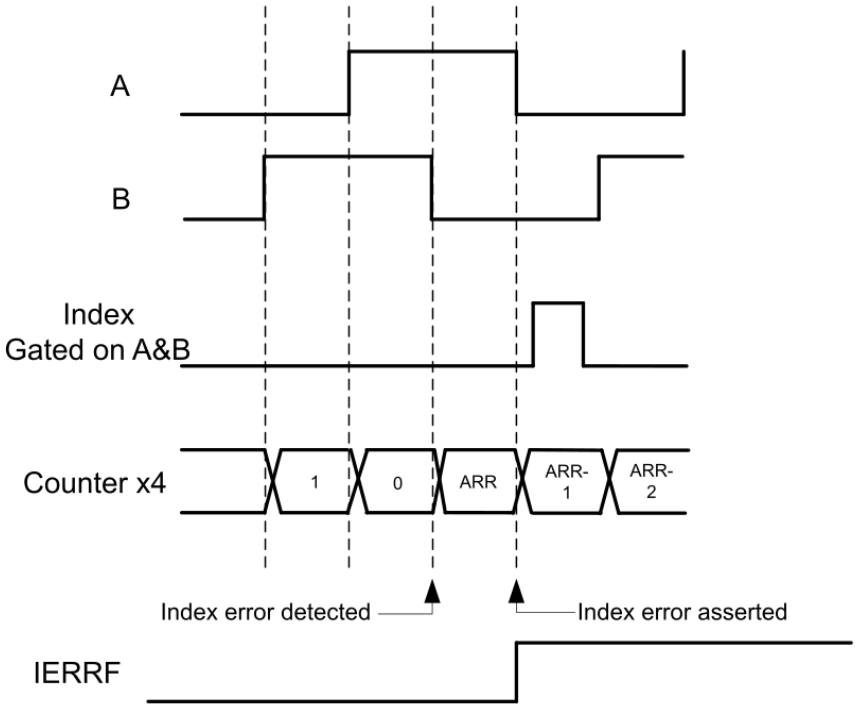


图 21-54 向下计数时 index 错误置位

21.5.14.7 Counter mode preload

某些应用中需要实时切换 counter mode（x4, x2, x1），可以通过 SMS 寄存器的 preload 功能来实现。当 SMSPE=1 时，使能 preload 功能，并且根据 SMSPS 寄存器的值，选择在什么事件发生时进行 preload：

- SMSPS=0：preload 由 update event(UEV)触发，即向上计数时计数器上溢出，或向下计数时计数器下溢出时触发，使用此配置时必须关闭 index 事件使能（IE=0）
- SMSPS=1：preload 发生由 index 事件触发

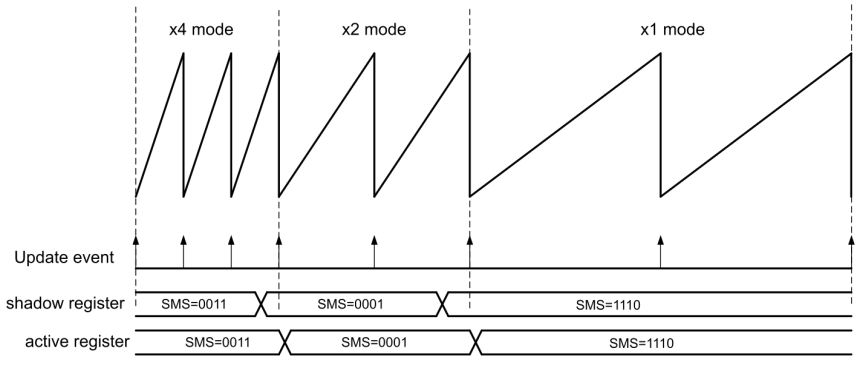


图 21-55 counter mode preload

21.5.14.8 编码器时钟输出

当 SMS 配置为编码器模式中的一种，配置 MMS[3:0]=1000 可以从 TRGO 输出被 decode 出来的编码器时钟。这个输出时钟是计数器实际计数频率，比如 x4 模式下这个时钟是正交编码信号频率的 4 倍。

21.5.15 GPTIM 从机模式

GPTIM 作为 slave 时（外部事件触发），可配置为三种工作模式：复位模式、门控模式、触发模式。

复位模式

此模式下，外部输入的事件将导致 TIM 内部所有 preload 寄存器重新初始化，CNT 回到 0 开始计数。以下图为例，计数器正常计数，外部 TI1 输入上升沿时，触发计数器清零，重新开始计数。

下图例中的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0 确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI
- 设定从模式控制器为复位模式，配置 GPTIM_SMCR.SMS[2:0]=100
- 打开通道使能，配置 GPTIM_CCER.CC1E=1
- 使能计数器，配置 GPTIM_CR1.CEN=1

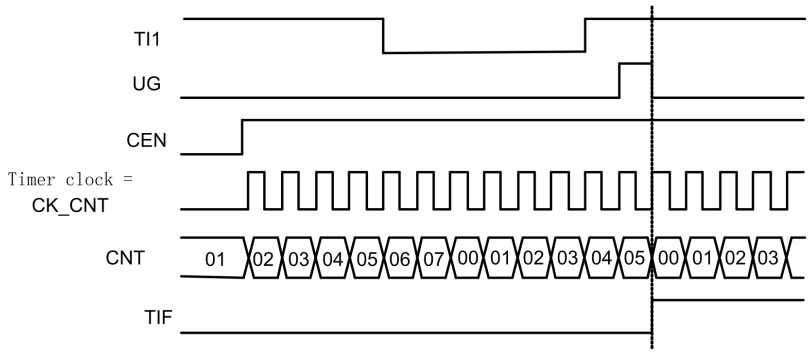


图 21-56 复位模式下的时序

门控模式

此模式下，计数器仅在输入信号为特定电平时工作。电平变换导致计数器开始或停止计数时，都会触发中断标志。

下图例中的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0 确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC1P=0

- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI
- 设定从模式控制器为门控模式，配置 GPTIM_SMCR.SMS[2:0]=101
- 打开通道使能，配置 GPTIM_CCER.CC1E=1
- 使能计数器，配置 GPTIM_CR1.CEN=1

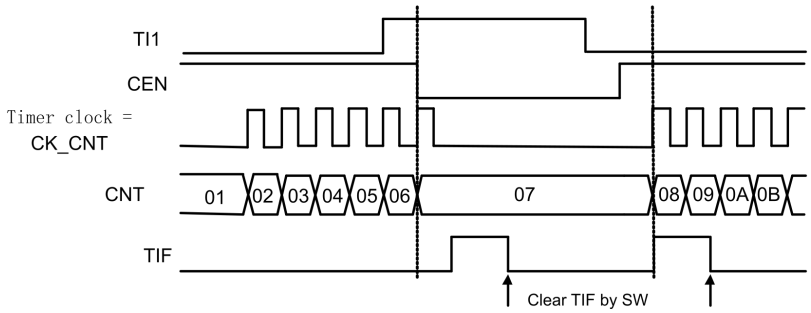


图 21-57 门控模式下的时序

触发模式

计数器在外部输入的某个事件到来后才开始计数。

下图例中的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1 功能
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0 确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI
- 设定从模式控制器为触发模式，配置 GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置 GPTIM_CCER.CC1E=1

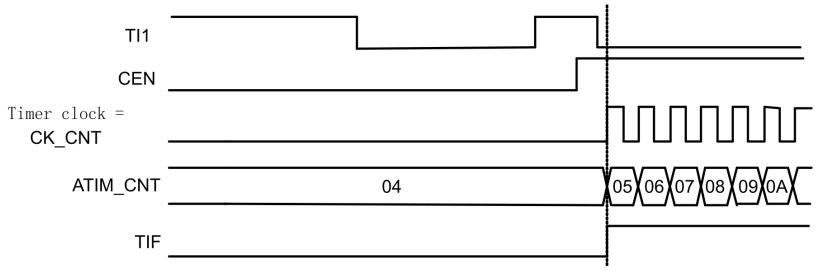


图 21-58 触发器模式下的时序

外部事件触发的外部时钟计数模式

可以将 ETR 设置为计数时钟，同时使用另一个外部输入作为计数器启动触发信号。比如在检测到 TI1 的上升沿之后，计数器开始以 ETR 输入的上升沿计数。

下图例中的配置如下：

- 在 GPIO 模块中，配置相应管脚为 GPTIM_CH1，ATIM_ETR 功能
- 设置 ETP 进行沿选择，GPTIM_SMCR.ETP=0
- 设置 ETR 分频比，配置 GPTIM_SMCR.ETPS[1:0]=01

- 配置输入滤波时间，GPTIM_SMCR.ETF[3:0]=0000
- 置位 ECE 寄存器，使能外部时钟模式 2，GPTIM_SMCR.ECE=1
- 关闭通道使能，配置 GPTIM_CCER.CC1E=0 确保之后通道配置成功
- 选择输入通道，配置 GPTIM_CCMR1.CC1S=01
- 选择计数有效沿，配置 GPTIM_CCER.CC1P=0
- 选择触发输入信号，配置 GPTIM_SMCR.TS[2:0]=101，TI1FP1 作为 TRGI
- 设定从模式控制器为触发模式，配置 GPTIM_SMCR.SMS[2:0]=110
- 打开通道使能，配置 GPTIM_CCER.CC1E=1

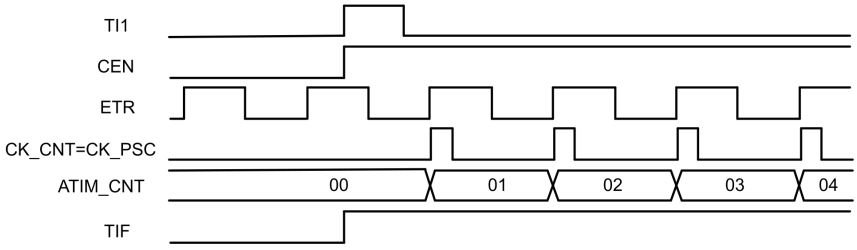


图 21-59 外部时钟模式 2＋触发模式下的时序

21.5.16 DMA 访问

GPTIM 支持 6 种 DMA 请求，分别为 4 个 CC 通道请求、外部触发请求和用户软件触发请求。

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，外部触发事件和软件触发事件也可以产生 DMA 请求，当这两种请求发生时，会启动 DMA Burst 传输，向 GPTIM 内部 1 个或多个寄存器写入数据，或者从 GPTIM 读取 1 个或多个寄存器值。

表 21-11 DMA 访问

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
GTIMx_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH4	0	0	Read CCR4	1
		1	Write CCR4	

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	

21.5.17 DMA Burst

DMA-Burst 支持一个事件触发连续多次 DMA 请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA 控制器需将外设目标地址指向一个虚拟寄存器 GPTIM_DMAR。在特定的定时器事件发生时，GPTIM 会连续发射多个 DMA 请求。每个 DMA 对 GPTIM_DMAR 的写操作都会被 GPTIM 重新定向到实际的功能寄存器上。

DBL 寄存器用于设置 DMA burst 长度，DBA 寄存器用于设置 DMA 访问 GPTIM 内部的基地址（相对于 GPTIM_CR 的 offset）。

21.6 寄存器

GPTIM 基地址 (0x40014C00)

地址	符号	名称
0x0000	GPTIM_CR1	GPTIM 控制寄存器 1
0x0004	GPTIM_CR2	GPTIM 控制寄存器 2
0x0008	GPTIM_SMCR	GPTIM 从机模式控制寄存器
0x000C	GPTIM_DIER	GPTIM DMA 和中断使能寄存器
0x0010	GPTIM_SR	GPTIM 状态寄存器
0x0014	GPTIM_EGR	GPTIM 事件产生寄存器
0x0018	GPTIM_CCMR1	GPTIM 捕捉/比较模式寄存器 1
0x001C	GPTIM_CCMR2	GPTIM 捕捉/比较模式寄存器 2
0x0020	GPTIM_CCER	GPTIM 捕捉/比较使能寄存器
0x0024	GPTIM_CNT	GPTIM 计数器寄存器
0x0028	GPTIM_PSC	GPTIM 预分频寄存器
0x002C	GPTIM_ARR	GPTIM 自动重载寄存器
0x0034	GPTIM_CCR1	GPTIM 捕捉/比较寄存器 1
0x0038	GPTIM_CCR2	GPTIM 捕捉/比较寄存器 2
0x003C	GPTIM_CCR3	GPTIM 捕捉/比较寄存器 3
0x0040	GPTIM_CCR4	GPTIM 捕捉/比较寄存器 4
0x0054	GPTIM_ECR	GPTIM 编码器模式控制寄存器
0x0058	GPTIM_TISEL	GPTIM ITR 选择寄存器
0x005C	GPTIM_AFR	GTIM 辅助功能寄存器
0x00F8	GPTIM_DCR	GPTIM DMA 控制寄存器
0x00FC	GPTIM_DMAR	GPTIM DMA 访问寄存器

21.6.1 GPTIM 控制寄存器 1(GPTIM_CR1)

符号	GPTIM_CR1							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				UIFREMAP	RFU	CKD	
权限	R-0000				RW-0	R-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
权限	RW-0	RW-00		RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11	UIFREMAP	RW	UIF 位替换到 cnt 最高位
10	RFU	R	RFU：未实现，读为 0
9:8	CKD	RW	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：tDTS=tCK_INT 01：tDTS=2*tCK_INT 10：tDTS=4*tCK_INT 11：RFU，禁止使用
7	ARPE	RW	Auto-reload 预装载使能 0：ARR 寄存器不使能 preload 1：ARR 寄存器使能 preload
6:5	CMS	RW	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	RW	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	RW	单脉冲输出模式 0：Update Event 发生时计数器不停止

位	位名	权限	功能描述
			1：Update Event 发生时计数器停止（自动清零 CEN）
2	URS	RW	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	RW	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	RW	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

21.6.2 GPTIM 控制寄存器 2(GPTIM_CR2)

符号	GPTIM_CR2							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						MMS_H	RFU
权限	R-000000						RW-0	R-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TI1S	MMS_L			CCDS	RFU		
权限	RW-0	RW-000			RW-0	R-000		

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	MMS_H	RW	参考 MMS_L
24:8	RFU	R	RFU：未实现，读为 0
7	TI1S	RW	GTIM 输入 TI1 选择

位	位名	权限	功能描述
			0：GTIM_CH1 引脚连接到 TI1 输入 1：GTIM_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入
6:4	MMS_L	RW	与 MMS_H 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：GTIM_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：OC1REF 用作 TRGO 0101：OC2REF 用作 TRGO 0110：OC3REF 用作 TRGO 0111：OC4REF 用作 TRGO 1000：编码器时钟输出，编码器模式的计数时钟信号从 TRGO 输出，仅当 SMS 配置为编码器模式中的一种时有效 注意：从机定时器或 ADC 必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	CCDS	RW	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
2:0	RFU	R	RFU：未实现，读为 0

21.6.3 GPTIM 从机模式控制寄存器(GPTIM_SMCR)

符号	GPTIM_SMCR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU						SMSPS	SMSPE
权限	R-000000						RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		TS_H		RFU			OCCS
权限	R-00		RW-00		R-000			RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ETP	ECE	ETPS		ETF			
权限	RW-0	RW-0	RW-00		RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSM	TS_L			SMS			
权限	RW-0	RW-000			RW-0000			

位	位名	权限	功能描述
31:26	RFU	R	RFU：未实现，读为 0
25	SMSPS	RW	SMS preload 触发源选择

位	位名	权限	功能描述
			0：SMS 更新由 update event 触发 1：SMS 更新由 index event 触发
24	SMSPE	RW	SMS preload 使能 0：禁止 SMS preload 1：使能 SMS preload
23:22	RFU	R	RFU：未实现，读为 0
21:20	TS_H	RW	参考 TS_L
19:17	RFU	R	RFU：未实现，读为 0
16	OCCS	RW	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	RW	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	RW	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	RW	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 GTIM 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	RW	外部触发信号滤波时钟和长度选择 0000：无滤波 0001：fSAMPLING=fCK_INT, N=2 0010：fSAMPLING=fCK_INT, N=4 0011：fSAMPLING=fCK_INT, N=8 0100：fSAMPLING=fDTS/2, N=6 0101：fSAMPLING=fDTS/2, N=8 0110：fSAMPLING=fDTS/4, N=6 0111：fSAMPLING=fDTS/4, N=8 1000：fSAMPLING=fDTS/8, N=6 1001：fSAMPLING=fDTS/8, N=8 1010：fSAMPLING=fDTS/16, N=5 1011：fSAMPLING=fDTS/16, N=6 1100：fSAMPLING=fDTS/16, N=8 1101：fSAMPLING=fDTS/32, N=5 1110：fSAMPLING=fDTS/32, N=6 1111：fSAMPLING=fDTS/32, N=8
7	MSM	RW	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

位	位名	权限	功能描述
			将当前定时器和从机定时器完美同步起来（？？）
6:4	TS_L	RW	与 TS_H 组成 5bit 内部触发选择，用于选择同步计数器的触发源 00000：内部触发信号（ITR0） 00001：内部触发信号（ITR1） 00010：内部触发信号（ITR2） 00011：内部触发信号（ITR3） 00100：TI1 边沿检测（TI1F_ED） 00101：滤波后 TI1（TI1FP1） 00110：滤波后 TI2（TI2FP2） 00111：外部触发输入（ETRF） 01000：内部触发信号（ITR4） 01001：内部触发信号（ITR5） 01010：内部触发信号（ITR6） 01011：内部触发信号（ITR7） 01100：内部触发信号（ITR8） 01101：内部触发信号（ITR9） 01110：内部触发信号（ITR10） 01111：内部触发信号（ITR11） 10000：内部触发信号（ITR12） 10001：内部触发信号（ITR13） 10010：内部触发信号（ITR14） 10011：内部触发信号（ITR15） Others：RFU 注意：仅当 SMS=0000 即禁止从机模式的情况下，可以改写 TS 寄存器
3:0	SMS	RW	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

位	位名	权限	功能描述
			计数器，产生更新事件并启动计数 1001：组合的闸门+触发模式；TRGI 为高电平时，计数时钟使能，当 TRGI 变低电平时，计数器复位且停止 1010：Encoder 模式：时钟+方向，x2 1011：Encoder 模式：时钟+方向，x1 1100：Encoder 模式：带方向的时钟，x2 1101：Encoder 模式：带方向的时钟，x1 1110：Quadrature Encoder x1 mode，仅使用 TI1FP1 边沿计数，有效沿由 CC1P 选择 1111：Quadrature Encoder x1 mode，仅使用 TI2FP2 边沿计数，有效沿由 CC2P 选择 其他：RFU

21.6.4 GPTIM DMA 和中断使能寄存器(GPTIM_DIER)

符号	GPTIM_DIER							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TERRIE	IERRIE	DIRIE	IDXIE	CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
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	RFU1	TDE	RFU	CC4DE	CC3DE	CC2DE	CC1DE	UDE
权限	R-0	RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIE	TIE	RFU	CC4IE	CC3IE	CC2IE	CC1IE	UIE
权限	RW-0	RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:24	RFU	R	RFU：未实现，读为 0
23	TERRIE	RW	状态转移错误中断使能，用于 encoder 模式，1 有效
22	IERRIE	RW	Index 错误中断使能，用于 encoder 模式，1 有效
21	DIRIE	RW	方向变化中断使能，用于 encoder 模式，1 有效
20	IDXIE	RW	Index 事件中断使能，用于 encoder 模式，1 有效
19	CC4BURSTEN	RW	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	RW	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

位	位名	权限	功能描述
17	CC2BURSTEN	RW	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	RW	捕捉比较通道 1 的 DMA 模式配置 0：Single 模式，仅访问 CCR 1：Burst 模式，通过 DCR 配置访问的地址和长度
15	RFU1	R	RFU：未实现，读为 0
14	TDE	RW	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	RFU	R	RFU：未实现，读为 0
12	CC4DE	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RW	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	RFU	R	RFU：未实现，读为 0
4	CC4IE	RW	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	RW	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	RW	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	RW	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

位	位名	权限	功能描述
0	UIE	RW	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

21.6.5 GPTIM 状态寄存器(GPTIM_SR)

符号	GPTIM_SR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	TERRIF	IERRIF	DIRIF	IDXIF	RFU			
权限	RW-0	RW-0	RW-0	RW-0	R-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			CC4OF	CC3OF	CC2OF	CC1OF	RFU1
权限	R-000			RW-0	RW-0	RW-0	RW-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BIF	TIF	RFU	CC4IF	CC3IF	CC2IF	CC1IF	UIF
权限	RW-0	RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:24	RFU	R	RFU：未实现，读为 0
23	TERRIF	RW	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	RW	Index 错误中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
21	DIRIF	RW	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	RW	Index 事件中断标志，用于 encoder 模式，硬件置位，软件写 1 清零
19:13	RFU	R	RFU：未实现，读为 0
12	CC4OF	RW	捕捉/比较通道 4 的 Overcapture 中断 参考 CC1OF
11	CC3OF	RW	捕捉/比较通道 3 的 Overcapture 中断 参考 CC1OF
10	CC2OF	RW	捕捉/比较通道 2 的 Overcapture 中断 参考 CC1OF
9	CC1OF	RW	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	RFU1	R	RFU：未实现，读为 0
7	BIF	RW	刹车事件中断标志，硬件置位，软件写 1 清零

位	位名	权限	功能描述
6	TIF	RW	触发事件中断标志，硬件置位，软件写 1 清零
5	RFU	R	RFU：未实现，读为 0
4	CC4IF	RW	捕捉/比较通道 4 中断标志 参考 CC1IF
3	CC3IF	RW	捕捉/比较通道 3 中断标志 参考 CC3IF
2	CC2IF	RW	捕捉/比较通道 2 中断标志 参考 CC2IF
1	CC1IF	RW	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 GTIM_CCR1 自动清零。
0	UIF	RW	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

21.6.6 GPTIM 事件产生寄存器(GPTIM_EGR)

符号	GPTIM_EGR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
权限	R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:7	RFU	R	RFU：未实现，读为 0
6	TG	RW	软件触发，软件置位此寄存器产生触发事件，硬件自动清零

位	位名	权限	功能描述
5	COMG	RW	软件 COM 事件，硬件置位，软件写 1 清零
4	CC4G	RW	捕捉/比较通道 4 软件触发，参考 CC1G
3	CC3G	RW	捕捉/比较通道 3 软件触发，参考 CC1G
2	CC2G	RW	捕捉/比较通道 2 软件触发，参考 CC1G
1	CC1G	RW	捕捉/比较通道 1 软件触发 如果 CC1 通道配置为输出：CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC1 通道配置为输入：当前计数值被捕捉到 GTIM_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	RW	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。

21.6.7 GPTIM 捕捉/比较模式寄存器 1(GPTIM_CCMR1)

输出比较模式

符号	GPTIM_CCMR1							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							OC2M_H
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OC1M_H
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC2CE	OC2M_L			OC2PE	OC2FE	CC2S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC1CE	OC1M_L			OC1PE	OC1FE	CC1S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	OC2M_H	RW	参考 OC2M_L
23:17	RFU	R	RFU：未实现，读为 0
16	OC1M_H	RW	参考 OC1M_L
15	OC2CE	RW	输出比较 2 清零使能，参考 OC1CE
14:12	OC2M_L	RW	输出比较 2 模式配置，参考 OC1M_L
11	OC2PE	RW	输出比较 2 预装载使能，参考 OC1PE
10	OC2FE	RW	输出比较 2 快速使能，参考 OC1FE
9:8	CC2S	RW	捕捉/比较 2 通道选择

位	位名	权限	功能描述
			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	RW	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	OC1M_L	RW	与 OC1M_H 组成 4bit 输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为 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 置高，否则置低 1000：可重触发的 OPM 模式 1，向上计数时若计数器停止输出高（active）否则输出 pwm 模式 1 波形， 向下计数时若计数器停止输出低（inactive）否则输出 pwm 模式 1 波形，配合 sms 配置为复位+触发模式使用。 1001：可重触发的 OPM 模式 2，向上计数时若计数器停止输出低（inactive）否则输出 pwm 模式 2 波形， 向下计数时若计数器停止输出高（active）否则输出 pwm 模式 2 波形，配合 sms 配置为复位+触发模式使用。 1010：RFU 1011：RFU 1100：组合 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 或 oc2ref 的逻辑， 通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1101：组合 PWM 模式 2，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形为 oc1ref 与 oc2ref 的逻辑， 通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1110：非对称 PWM 模式 1，oc1ref 波形与 pwm 模式 1 一致，oc1refc 波形在向上计数时为 oc1ref 波形， 向下计数时为 oc2ref 波形，通道输出

位	位名	权限	功能描述
			oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组） 1111：非对称 PWM 模式 2，oc1ref 波形与 pwm 模式 2 一致，oc1refc 波形在向上计数时为 oc1ref 波形， 向下计数时为 oc2ref 波形，通道输出 oc1refc 波形（1 通道和 2 通道为一组，3 通道和 4 通道为一组）
3	OC1PE	RW	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 vent 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC1FE	RW	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	RW	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

符号	GPTIM_CCMR1							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IC2F				IC2PSC		CC2S	
权限	RW-0000				RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IC1F				IC1PSC		CC1S	
权限	RW-0000				RW-00		RW-00	

位	位名	权限	功能描述

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:12	IC2F	RW	输入捕捉 2 滤波
11:10	IC2PSC	RW	输入捕捉 2 预分频
9:8	CC2S	RW	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
7:4	IC1F	RW	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 fDTS 采样 0001：fSAMPLING=fCK_INT, N=2 0010：fSAMPLING=fCK_INT, N=4 0011：fSAMPLING=fCK_INT, N=8 0100：fSAMPLING=fDTS/2, N=6 0101：fSAMPLING=fDTS/2, N=8 0110：fSAMPLING=fDTS/4, N=6 0111：fSAMPLING=fDTS/4, N=8 1000：fSAMPLING=fDTS/8, N=6 1001：fSAMPLING=fDTS/8, N=8 1010：fSAMPLING=fDTS/16, N=5 1011：fSAMPLING=fDTS/16, N=6 1100：fSAMPLING=fDTS/16, N=8 1101：fSAMPLING=fDTS/32, N=5 1110：fSAMPLING=fDTS/32, N=6 1111：fSAMPLING=fDTS/32, N=8
3:2	IC1PSC	RW	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
1:0	CC1S	RW	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

21.6.8 GPTIM 捕捉/比较模式寄存器 2(GPTIM_CCMR2)

输出比较模式

符号	GPTIM_CCMR2							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	GPTIM_CCMR2							
位名	RFU							OC4M_H
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							OC3M_H
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	OC4CE	OC4M_L			OC4PE	OC4FE	CC4S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OC3CE	OC3M_L			OC3PE	OC3FE	CC3S	
权限	RW-0	RW-000			RW-0	RW-0	RW-00	

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	OC4M_H	RW	参考 OC4M_L
23:17	RFU	R	RFU：未实现，读为 0
16	OC3M_H	RW	参考 OC3M_L
15	OC4CE	RW	输出比较 4 清零使能，参考 OC1CE
14:12	OC4M_L	RW	输出比较 4 模式配置，参考 OC1M_L
11	OC4PE	RW	输出比较 4 预装载使能，参考 OC1PE
10	OC4FE	RW	输出比较 4 快速使能，参考 OC1FE
9:8	CC4S	RW	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	RW	输出比较 1 清零使能 0：OC1REF 不受 ETRF 影响 1：检测到 ETRF 高电平时，自动清零 OC1REF
6:4	OC3M_L	RW	输出比较 3 模式配置，参考 OC1M_L
3	OC3PE	RW	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 vent 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中
2	OC3FE	RW	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

位	位名	权限	功能描述
			此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效
1:0	CC3S	RW	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，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 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

符号	GPTIM_CCMR2							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IC4F				IC4PSC		CC4S	
权限	RW-0000				RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IC3F				IC3PSC		CC3S	
权限	RW-0000				RW-00		RW-00	

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:12	IC4F	RW	输入捕捉 4 滤波
11:10	IC4PSC	RW	输入捕捉 4 预分频
9:8	CC4S	RW	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	RW	输入捕捉 3 滤波
3:2	IC3PSC	RW	输入捕捉 3 预分频
1:0	CC3S	RW	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，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 映射到 TRC 注意：CC3S 仅在通道关闭时（CC3E=0）可以写

21.6.9 GPTIM 捕捉/比较使能寄存器(GPTIM_CCER)

符号	GPTIM_CCER							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CC4NP	RFU	CC4P	CC4E	CC3NP	RFU	CC3P	CC3E
权限	RW-0	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CC2NP	RFU	CC2P	CC2E	CC1NP	RFU	CC1P	CC1E
权限	RW-0	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15	CC4NP	RW	捕捉/比较 4 互补输出极性，参考 CC1NP
14	RFU	R	RFU：未实现，读为 0
13	CC4P	RW	捕捉/比较 4 输出极性，参考 CC1P
12	CC4E	RW	捕捉/比较 4 输出使能，参考 CC1E
11	CC3NP	RW	捕捉/比较 3 互补输出极性，参考 CC1NP
10	RFU	R	RFU：未实现，读为 0
9	CC3P	RW	捕捉/比较 3 输出极性，参考 CC1P
8	CC3E	RW	捕捉/比较 3 输出使能，参考 CC1E
7	CC2NP	RW	捕捉/比较 2 互补输出极性，参考 CC1NP
6	RFU	R	RFU：未实现，读为 0
5	CC2P	RW	捕捉/比较 2 输出极性，参考 CC1P
4	CC2E	RW	捕捉/比较 2 输出使能，参考 CC1E
3	CC1NP	RW	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	RFU	R	RFU：未实现，读为 0
1	CC1P	RW	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 IC1 的上升沿 CC1NP=0，CC1P=1：捕捉 IC1 的下降沿 CC1NP=1：捕捉 IC1 的上升沿和下降沿

位	位名	权限	功能描述
0	CC1E	RW	捕捉/比较 1 输出使能 CC1 通道配置为输出时 0：OC1 不 active 1：OC1 active CC1 通道配置为输入时 0：关闭捕捉功能 1：使能捕捉功能

21.6.10 GPTIM 计数器寄存器(GPTIM_CNT)

符号	GPTIM_CNT							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	UIFCPY	RFU						
权限	R-0	R-0000000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	RW-00000000							

位	位名	权限	功能描述
31	UIFCPY	R	UIF copy 当 UIFREMAP=1，UIF 寄存器的值被实时复制到这里，只读
30:16	RFU	R	RFU：未实现，读为 0
15:0	CNT	RW	计数器值

21.6.11 GPTIM 预分频寄存器(GPTIM_PSC)

符号	GPTIM_PSC							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC							
权限	RW-00000000							

符号	GPTIM_PSC							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PSC	RW	计数器时钟（CK_CNT）预分频值 fCK_CNT=fCK_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

21.6.12 GPTIM 自动重载寄存器(GPTIM_ARR)

符号	GPTIM_ARR							
地址	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-11111111							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-11111111							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	ARR	RW	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

21.6.13 GPTIM 捕捉/比较寄存器 1(GPTIM_CCR1)

符号	GPTIM_CCR1							
地址	0x0034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	GPTIM_CCR1							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR1	RW	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

21.6.14 GPTIM 捕捉/比较寄存器 2(GPTIM_CCR2)

符号	GPTIM_CCR2							
地址	0x0038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR2	RW	捕捉/比较通道 2 寄存器 如果通道 2 配置为输出： 这是一个 preload 寄存器，其内容被载入 shadow 寄存器后用于与计数器比较产生 OC2 输出 如果通道 2 配置为输入： CCR2 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR2 为只读

21.6.15 GPTIM 捕捉/比较寄存器 3(GPTIM_CCR3)

符号	GPTIM_CCR3							
地址	0x003C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR3							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR3							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR3	RW	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

21.6.16 GPTIM 捕捉/比较寄存器 4(GPTIM_CCR4)

符号	GPTIM_CCR4							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR4							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR4							
权限	RW-00000000							

位	位名	权限	功能描述

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR4	RW	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

21.6.17 GPTIM 编码器模式控制寄存器(GPTIM_ECR)

符号	GPTIM_ECR							
地址	0x0054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	IPOS		FIDX	RFU		IDIR		INDEX_EN
权限	RW-00		RW-0	R-00		RW-00		RW-0

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:6	IPOS	RW	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	RW	First Index 1：仅第一个 index 有效 0：所有 index 有效
4:3	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
2:1	IDIR	RW	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	INDEX_EN	RW	Index 使能 0：禁止 index 功能 1：使能 index 功能

21.6.18 GPTIM ITR 选择寄存器(GPTIM_TISEL)

符号	GPTIM_TISEL							
地址	0x0058							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU				TI4SEL			
权限	R-0000				RW-0000			
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				TI3SEL			
权限	R-0000				RW-0000			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				TI2SEL			
权限	R-0000				RW-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TI1SEL			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:28	RFU	R	RFU：未实现，读为 0
27:24	TI4SEL	RW	TI4 的输入选择信号 0000：tim_ti4_in0:TIMx_CH4 0001: tim_ti4_in1 … 1111: tim_ti4_in15
23:20	RFU	R	RFU：未实现，读为 0
19:16	TI3SEL	RW	TI3 的输入选择信号 0000：tim_ti3_in0:TIMx_CH3 0001: tim_ti3_in1 … 1111: tim_ti3_in15
15:12	RFU	R	RFU：未实现，读为 0
11:8	TI2SEL	RW	TI4 的输入选择信号 0000：tim_ti2_in0:TIMx_CH2 0001: tim_ti2_in1 … 1111: tim_ti2_in15
7:4	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
3:0	TI1SEL	RW	TI1 的输入选择信号 0000：tim_ti1_in0:TIMx_CH1 0001: tim_ti1_in1 … 1111: tim_ti1_in15

21.6.19 GTIM 辅助功能寄存器(GPTIM_AFR)

符号	GPTIM_AFR							
地址	0x005C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ETRSEL				RFU	OCC1R_SEL		
权限	RW-0000				R-0	RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							
权限	R-00000000							

位	位名	权限	功能描述
31:24	RFU	R	RFU，未实现，读为 0
23:20	ETRSEL	RW	ETR 的输入选择信号 0000：GPTIM_ETR 0001：GPTIM_ETR_IN1 … 1111：GPTIM_ETR_IN15
19	RFU	R	RFU，未实现，读为 0
18:16	OCC1R_SEL	RW	OCREF_CLR 输入选择，用于选择使用什么片内信号产生 OCREF 清零信号 000：COMP0_OUT 001：COMP1_OUT 010：COMP2_OUT 其他：RFU
15:0	RFU	R	RFU，未实现，读为 0

21.6.20 GPTIM DMA 控制寄存器(GPTIM_DCR)

符号	GPTIM_DCR							
地址	0x00F8							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							

符号	GPTIM_DCR							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			DBL				
权限	R-000			RW-00000				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU1			DBA				
权限	R-000			RW-00000				

位	位名	权限	功能描述
31:13	RFU	R	RFU：未实现，读为 0
12:8	DBL	RW	DMA Burst 长度 对 GPTIM_DMAR 寄存器的读写将触发 burst DMA 操作，burst 长度为 1~23 00000：长度=1 00001：长度=2 …… 10110：长度=23 其他：无效值，禁止写入
7:5	RFU1	R	RFU：未实现，读为 0
4:0	DBA	RW	DMA 基地址，定义指向寄存器的偏移地址 00000：GTIM_CR1 00001：GTIM_CR2 00010：GTIM_SMCR …… 注意：当 DBA+DBL 超出了 GTIM 寄存器地址范围，则实际 burst 传输到 GTIM 最高寄存器地址后自动停止， 即 burst 长度会缩短。

21.6.21 GPTIM DMA 访问寄存器(GPTIM_DMAR)

符号	GPTIM_DMAR							
地址	0x00FC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	DMAR							
权限	W-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	DMAR							
权限	W-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DMAR							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

符号	GPTIM_DMAR
位名	DMAR
权限	W-00000000

位	位名	权限	功能描述
31:0	DMAR	W	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 GTIM_DMAR，GTIM 会根据 DBL 的值产生多次 DMA 请求

22 定时器阵列单元（TAU）

22.1 概述

TAU 是 16bit 定时器组成的阵列，支持以下功能

- 2 组共 12 个独立的 16bit 向上计数器（TAU00~TAU07，TAU10~TAU13）
- Auto-reload 定时
- 外部事件计数
- 输入边沿捕捉（事件触发、自由计数）
- 输入预分频
- 脉冲宽度或周期捕捉（PWC 模式）
- 脉冲宽度调制（PWM 模式）
- 可以级联成 32bit 定时器

22.2 结构框图

TAU 整体结构示意图如下，包含 TAU0 和 TAU1 两个阵列。

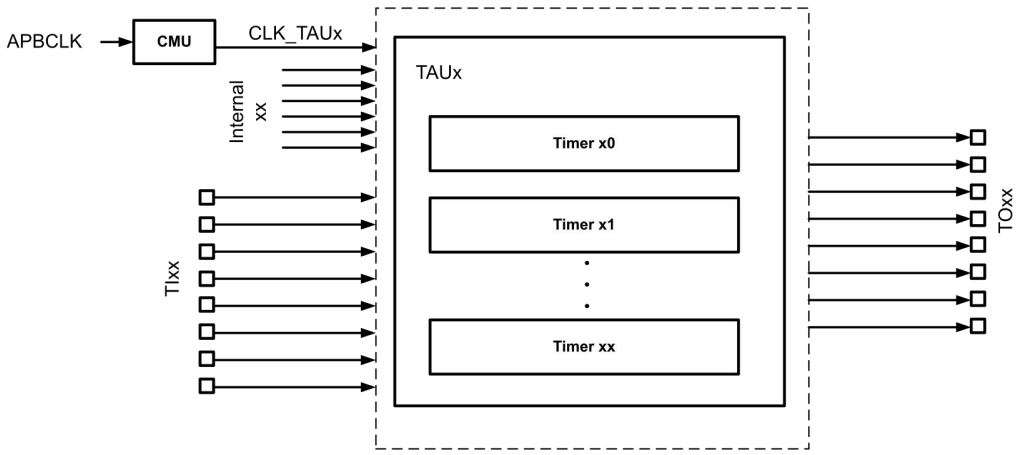


图 22-1 TAU 整体结构框图

独立定时器结构框图如下：

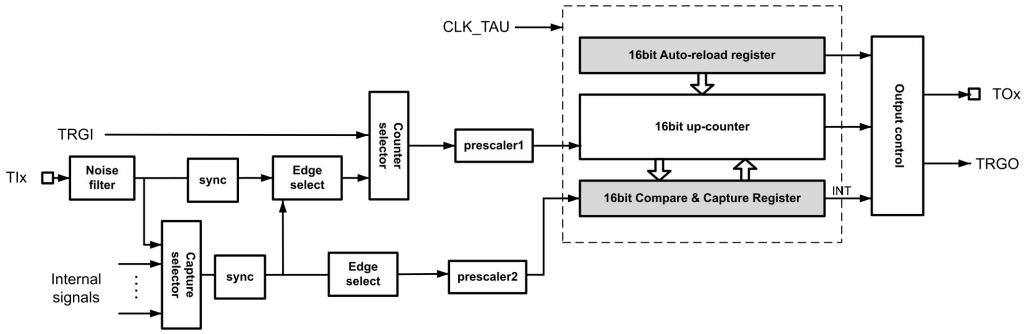


图 22-2 TAU 定时器结构框图

每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。

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。

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。

22.3 功能描述

22.3.1 时钟和复位

TAU 使用 APBCLK 工作，在使能 TAU 开始工作前，必须使能其总线时钟，即置位 TAUx_PCE 寄存器，参见 CMU 章节。比如想要使能 TAU0 中的 Timerx 工作，应置位 TAU0_PEC 寄存器。

通过 TAUx_RST 寄存器可以实现每组定时器的独立复位，参见 RMU 章节。如果想要复位 TAU0 中的 Timerx，应使能 TAU0_RST 寄存器。

22.3.2 定时单元

TAU 的定时单元由计数器和自动重载寄存器组成，计数器向上计数。计数时钟通过 16 位预分频器对计数源进行分频后得到。计数值 CNT 只读，关闭定时器使能 TEN 时计数器清零。

以下功能寄存器支持 preload 功能，软件通过配置 PE 寄存器使能或者禁止 preload 功能：

- 自动重载寄存器 ARR
- 捕捉比较寄存器 CCR

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

更新事件（updateevent）包括：

- 计数器溢出
- 软件写 UEV 寄存器

通过使用 preload 功能，可以实现当前 PWM 周期结束后，自动更新下一个 PWM 周期的占空比和长度配置，放松软件操作的时序要求。

TAU 以 APBCLK 时钟为工作时钟，触发计数器变化的计数源信号有以下四个来源：

- APBCLK
- 外部引脚输入 TIxx
- 内部捕捉信号输入
- 从机模式下主机的 trgo 信号

22.3.3 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。

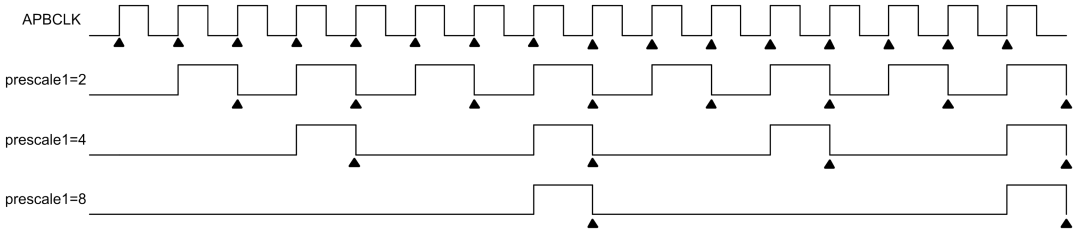


图 22-3 定时器内部时钟预分频计数

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。

22.3.4 外部输入信号计数

定时器通过 CNTSEL 寄存器选择外部输入信号计数时，可以配置 CNTEDGE 寄存器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 TIxx 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。

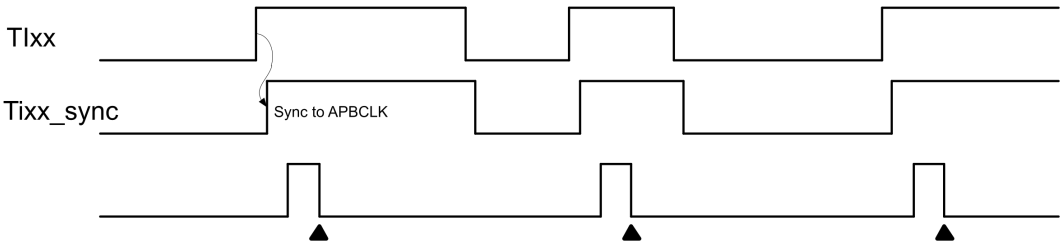


图 22-4 定时器外部事件上升沿递增时序

下图为 TIxx 上升下降沿事件计数，并且分频系数为 2 的例子。

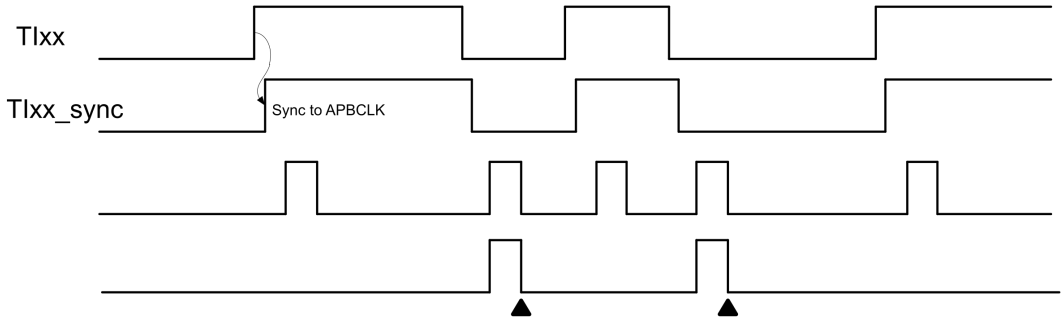


图 22-5 定时器外部事件上升下降沿递增，2 分频时序

定时器的外部引脚输入 Tixx 可以选择是否使能数字滤波，以抑制信号噪声。使能数字滤波的情况下，使用 APBCLK 连续 N 次采样相同才认为是输入变化有效；不使能数字滤波时，扩展定时器直接对管脚输入进行采样。数字滤波的长度由 DFLEN 寄存器定义，DFLEN=0 表示关闭数字滤波，最大滤波长度是 255 次采样。

数字滤波示意图如下（DFLEN=3）：

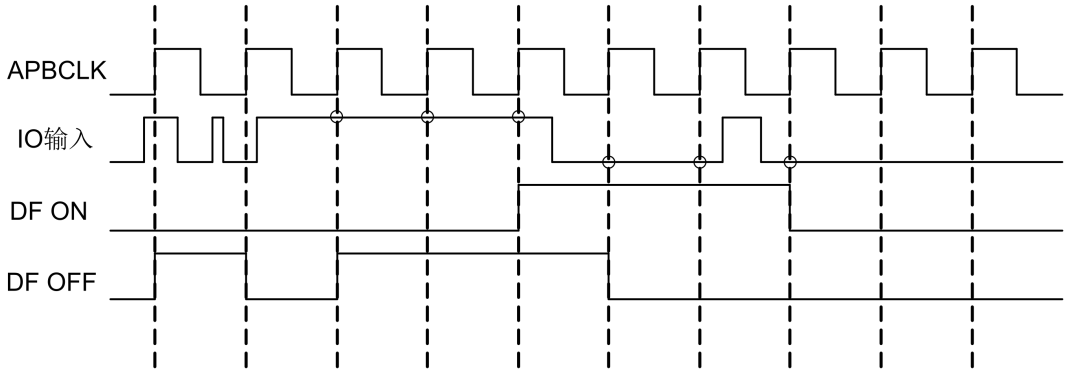


图 22-6 Tixx 输入数字滤波示意图

22.3.5 内部捕捉输入信号计数

通过配置 CNTSEL 寄存器和 CAPSEL 寄存器可以实现对内部捕捉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。

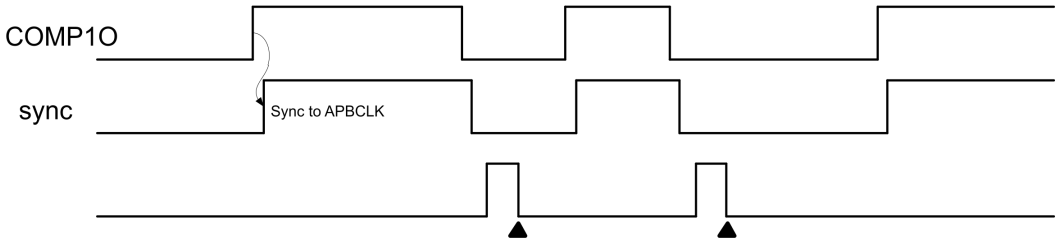


图 22-7 定时器内部信号下降沿递增

22.3.6 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，计数器的工作时钟仍然受 CNTSEL 控制。

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。

表 22-1 定时器级联触发选择

TAU0 定时器	TS=00	TS=01	TS=10	TS=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
7	Timer0	Timer4	Timer5	Timer6

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。

注意：只有 TAU0 支持级联计数模式，TAU1 不支持级联计数。

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。

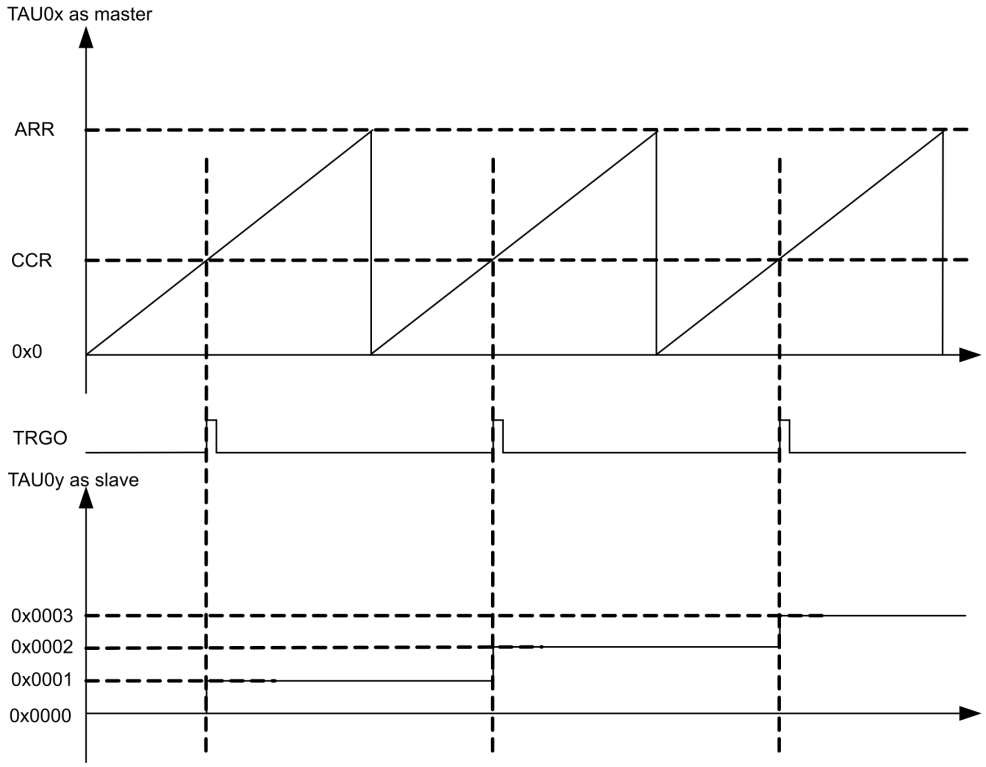


图 22-8 定时器级联计数（trigger to increment）

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。

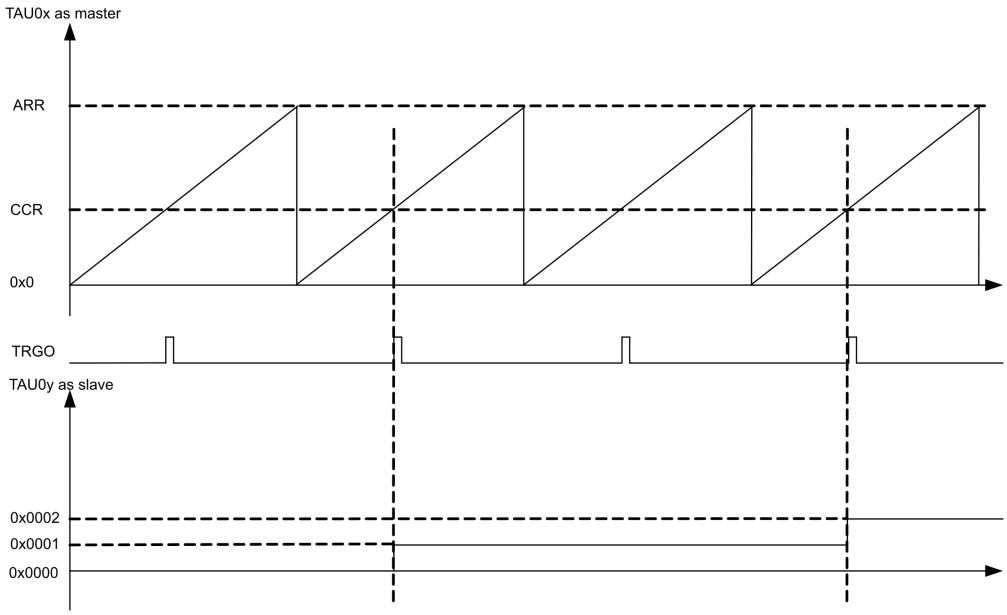


图 22-9 定时器级联计数（trigger to increment），从机 Prescale1 = 1

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。

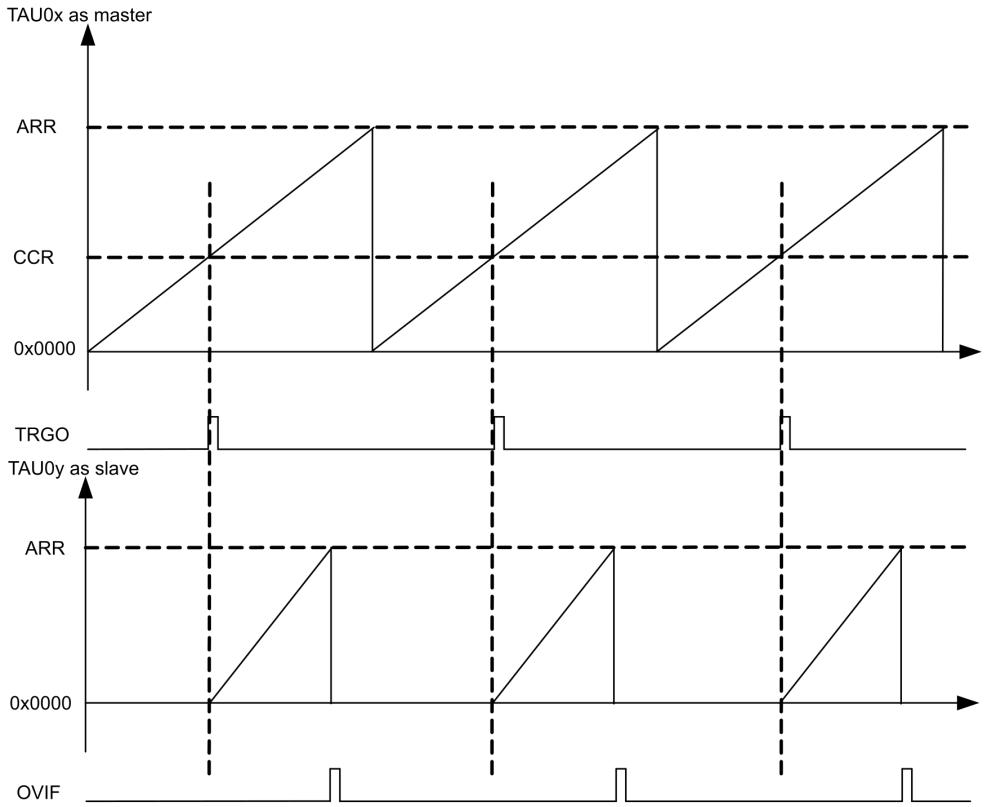


图 22-10 定时器级联计数（trigger to start）

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。

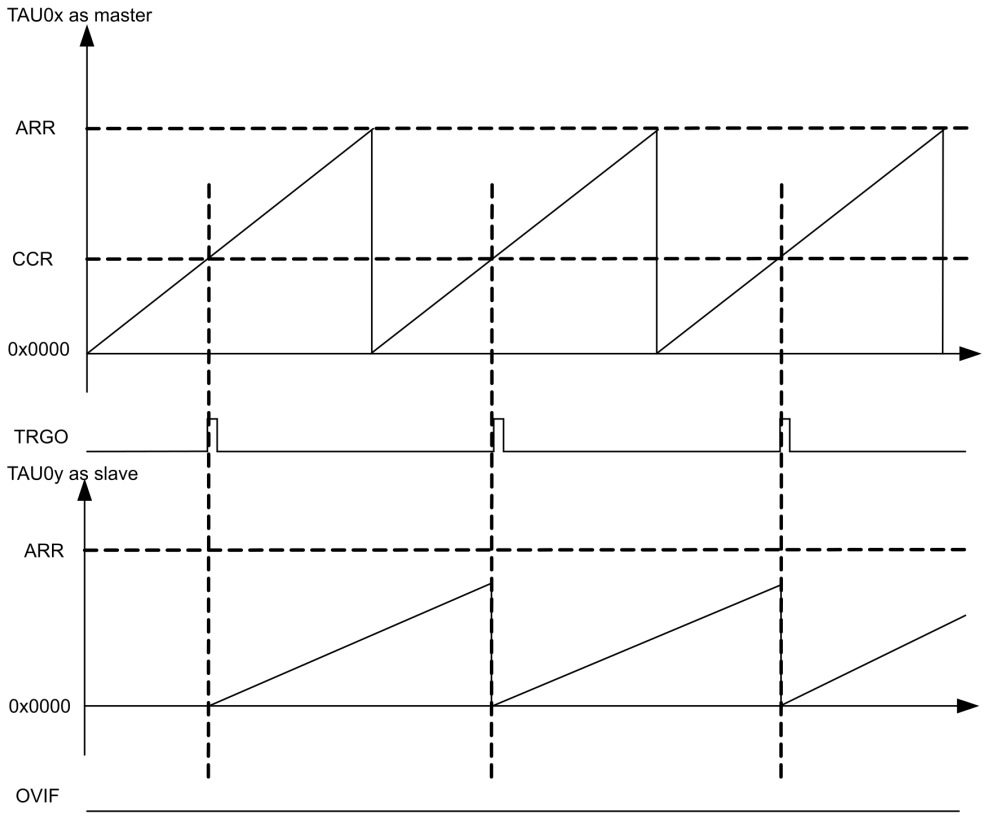


图 22-11 定时器级联计数（trigger to start），从机无法产生溢出

22.3.7 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。

每个捕捉模式下可通过 CAPONCE 寄存器选择单次捕捉模式或连续捕捉模式。 - 当 CAPONCE=0 时，定时器连续工作，直到软件关闭。 - 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。

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。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。

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。

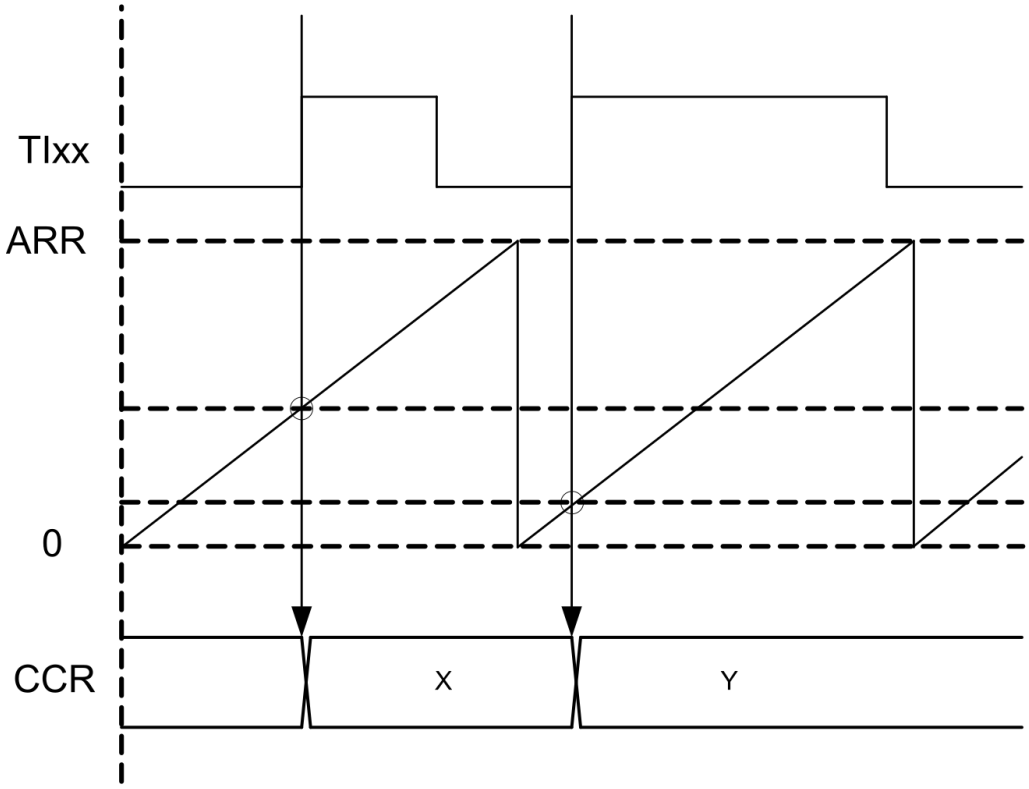


图 22-12 定时器捕捉外部输入信号上升沿

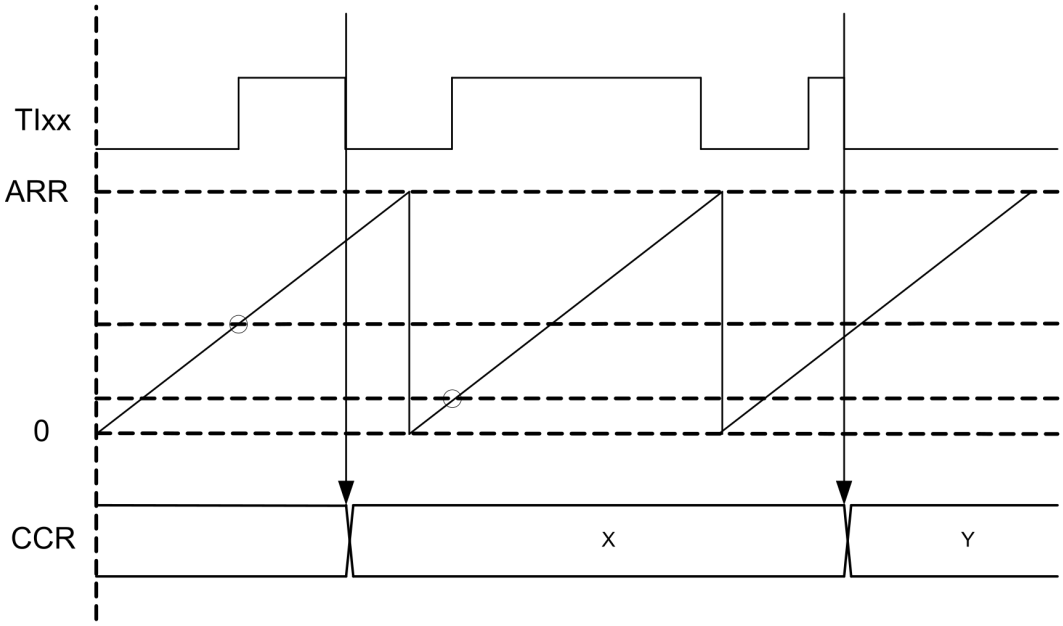


图 22-13 定时器 2 分频捕捉外部输入信号下降沿

脉冲宽度捕捉（PWC）功能用于测量输入信号的高电平宽度或者低电平宽度时，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。

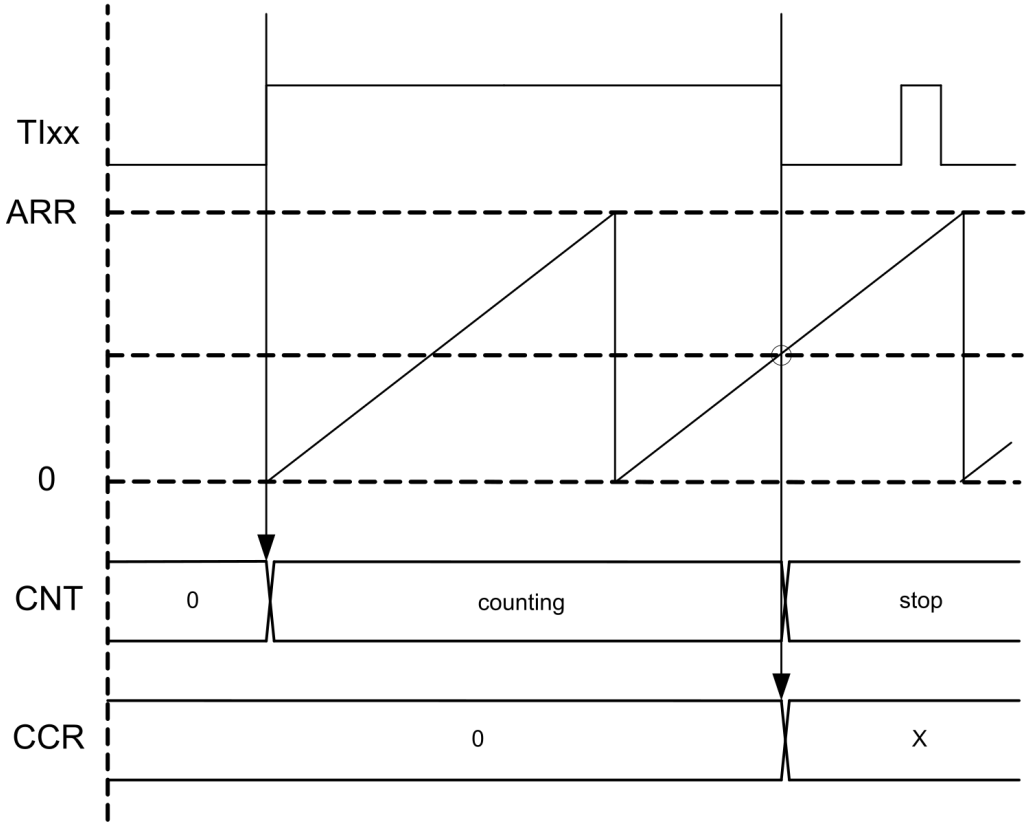


图 22-14 定时器脉冲宽度高电平捕捉

22.3.8 捕捉源选择

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。

表 22-2 TAU0_定时器捕捉源选择

TAU0 定时器	CAPSEL	被捕捉信号
Timer0	000	TI00
	001	–
	010	UART1_RXD
	011	–
	100	UART2_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer1	000	TI01
	001	–
	010	UART4_RXD

TAU0 定时器	CAPSEL	被捕捉信号
	011	–
	100	UART5_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer2	000	TI02
	001	–
	010	UART1_RXD
	011	–
	100	UART2_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer3	000	TI03
	001	–
	010	UART4_RXD
	011	–
	100	UART5_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer4	000	TI04
	001	–
	010	–
	011	RCHF
	100	RCLP
	101	–
	110	–
	111	–
Timer5	000	TI05
	001	–
	010	–
	011	RCHF
	100	RCLP
	101	–
	110	–
	111	–
Timer6	000	TI06
	001	–
	010	–
	011	–
	100	–
	101	–
	110	–
	111	–
Timer7	000	TI07
	001	–
	010	–
	011	–
	100	–
	101	–

TAU0 定时器	CAPSEL	被捕捉信号
	110	–
	111	–

表 22-3 TAU1_定时器捕捉源选择

TAU1 定时器	CAPSEL	被捕捉信号
Timer0	000	TI10
	001	–
	010	UART1_RXD
	011	–
	100	UART2_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer1	000	TI11
	001	–
	010	UART4_RXD
	011	–
	100	UART5_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer2	000	TI12
	001	–
	010	UART1_RXD
	011	–
	100	UART2_RXD
	101	CMP1O
	110	–
	111	RCLP
Timer3	000	TI13
	001	–
	010	UART4_RXD
	011	–
	100	UART5_RXD
	101	CMP1O
	110	–
	111	RCLP

22.3.9 单个 PWM 输出

每个 TAU 支持独立 PWM 输出，通过引脚 TOxx 输出波形。启动 TAU 之后定时器从初值开始计数，当计数值等于比较寄存器（CCR）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：

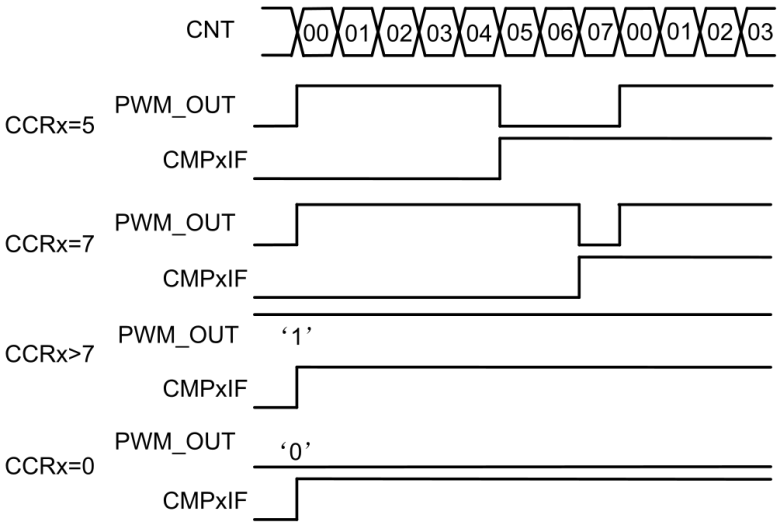


图 22-15 TAU 的 PWM 输出示意图（ARR=7，输出极性取反）

PWM 周期由 ARR 寄存器决定：

$$Period_{PWM} = T_{CNT} \times (ARR + 1)$$

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-$Duty_{PWM}$。

22.3.10 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。

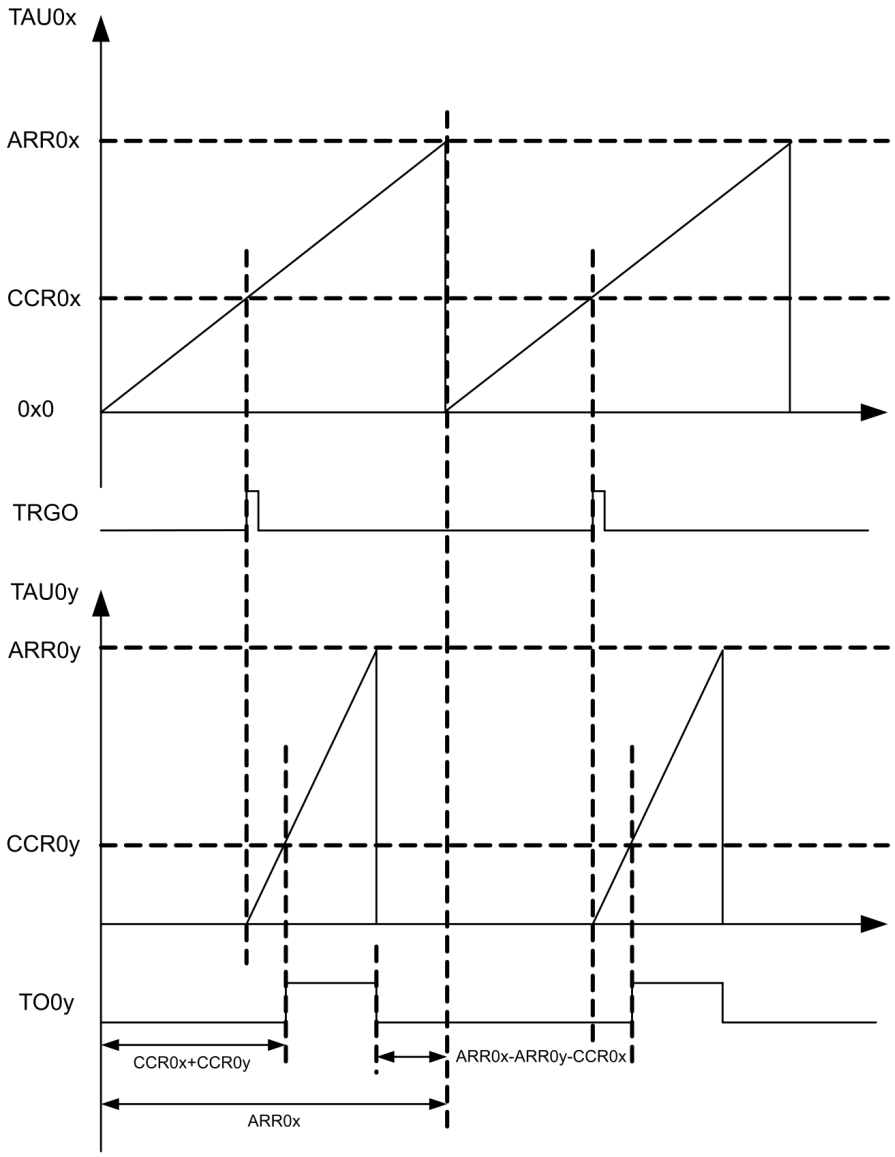


图 22-16 非对称级联 PWM 输出示意图

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。

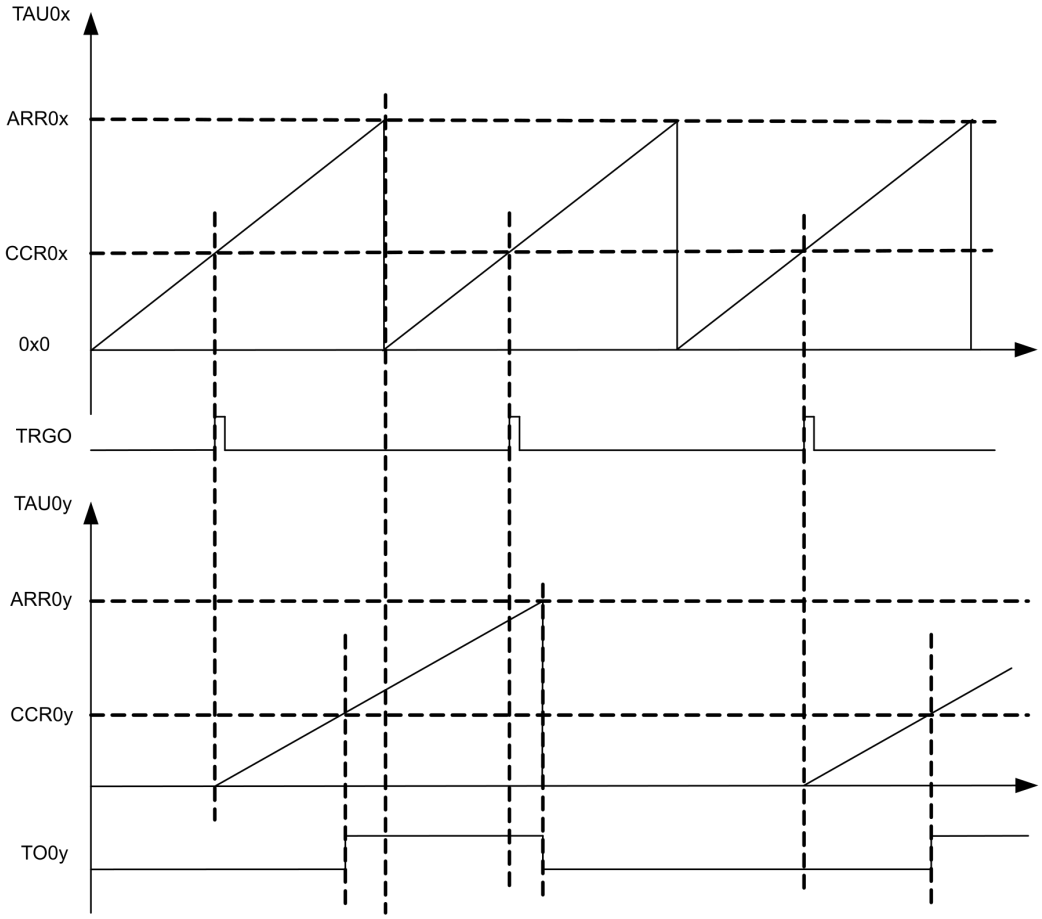


图 22-17 从机 ARR 大于主机的情况

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。

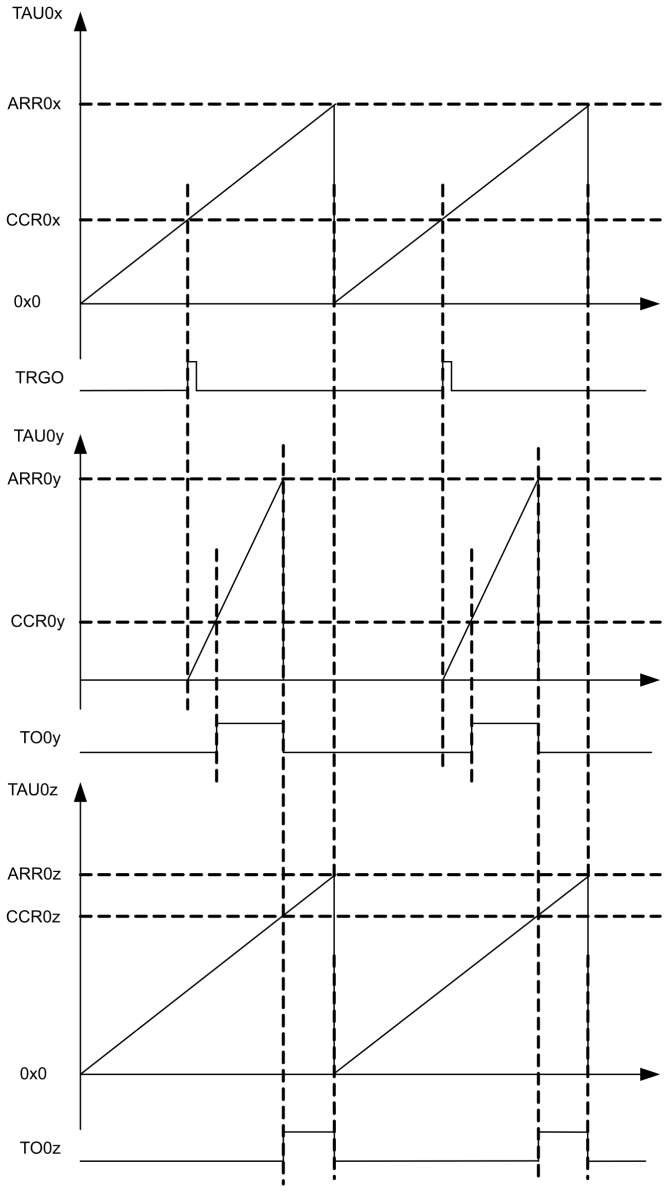


图 22-18 非对称非交叠 PWM 输出示意图

22.3.11 触发信号输出（TRGO）

每个 TAU 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 TAUxx_TRGO，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。

22.4 寄存器

TAU_COMMON 基地址 (0x40013C00)

地址	符号	名称
0x0000	TAU0_CR	TAU0 控制寄存器

地址	符号	名称
0x0004	TAU1_CR	TAU1 控制寄存器
0x0008	TAU0_EGR	TAU0 事件产生寄存器
0x000C	TAU1_EGR	TAU1 事件产生寄存器

TAU0 基地址 (0x40013C10)

地址	符号	名称
0x0000 + 0x04*x	TAU0_TxCFGR x=[0, 7]	TAU0 通道 x 配置寄存器
0x0020 + 0x04*x	TAU0_TxMDR x=[0, 7]	TAU0 通道 x 模式寄存器
0x0040 + 0x04*x	TAU0_TxARR x=[0, 7]	TAU0 通道 x 目标值寄存器
0x0060 + 0x04*x	TAU0_TxCCR x=[0, 7]	TAU0 通道 x 捕捉比较寄存器
0x0080 + 0x04*x	TAU0_TxIER x=[0, 7]	TAU0 通道 x 中断使能寄存器
0x00A0 + 0x04*x	TAU0_TxISR x=[0, 7]	TAU0 通道 x 中断标志寄存器
0x00C0 + 0x04*x	TAU0_TxCNT x=[0, 7]	TAU0 通道 x 计数值寄存器

TAU1 基地址 (0x40013CF0)

地址	符号	名称
0x0000 + 0x04*x	TAU1_TxCFGR x=[0, 3]	TAU1 通道 x 配置寄存器
0X0010 + 0x04*x	TAU1_TxMDR x=[0, 3]	TAU1 通道 x 模式寄存器
0x0020 + 0x04*x	TAU1_TxARR x=[0, 3]	TAU1 通道 x 目标值寄存器
0x0030 + 0x04*x	TAU1_TxCCR x=[0, 3]	TAU1 通道 x 捕捉比较寄存器
0x0040 + 0x04*x	TAU1_TxIER x=[0, 3]	TAU1 通道 x 中断使能寄存器
0x0050 + 0x04*x	TAU1_TxISR x=[0, 3]	TAU1 通道 x 中断标志寄存器
0x0060 + 0x04*x	TAU1_TxCNT x=[0, 3]	TAU1 通道 x 计数值寄存器

22.4.1 TAU0 控制寄存器(TAU0_CR)

符号	TAU0_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	TAU0_CR							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TEN							
权限	RW-00000000							

位	位名	权限	功能描述
31:8	RFU	R	未实现：读为 0
7:0	TEN	RW	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。

22.4.2 TAU1 控制寄存器(TAU1_CR)

符号	TAU1_CR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				TEN			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:4	RFU	R	未实现：读为 0
3:0	TEN	RW	定时器使能，每个 bit 单独控制一路定时器，1 使能 Bit0：Timer0 Bit1：Timer1 …… Bit3：Timer3 注意，应在完成所有配置寄存器写入后，最后置位 TEN 寄存器。TEN 置位后，定时器立即开始工作。

22.4.3 TAU0 事件产生寄存器(TAU0_EGR)

符号	TAU0_EGR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	UG							
权限	W-00000000							

位	位名	权限	功能描述
31:8	RFU	R	未实现：读为 0
7:0	UG	W	用户更新事件，每个 bit 对应一个寄存器，软件写 1 产生更新事件，在 PE=1 的情况下， 会将 preload 寄存器内容更新到影子寄存器中，更新完成后硬件自动清零。 Bit0：Timer0 Bit1：Timer1 …… Bit7：Timer7

22.4.4 TAU1 事件产生寄存器(TAU1_EGR)

符号	TAU1_EGR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				UG			
权限	R-0000				W-0000			

位	位名	权限	功能描述
31:4	RFU	R	未实现：读为 0
3:0	UG	W	用户更新事件，每个 bit 对应一个寄存器，软件写 1 产生更新事件，在 PE=1 的情况下， 会将 preload 寄存器内容更新到影子寄存器中，更新完成后硬件自动清零。 Bit0：Timer0 Bit1：Timer1 …… Bit3：Timer3

22.4.5 TAU0 通道 x 配置寄存器(TAU0_TxCFGR x=[0, 7])

符号	TAU0_TxCFGR x=[0, 7]							
地址	0x0000 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARPE	CAPCLR	CAPONCE	NF	CAPEDGE		CNTEDGE	
权限	RW-0	RW-0	RW-0	RW-0	RW-00		RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	TS		CAPSEL			CNTSEL	
权限	RW-0	RW-00		RW-000			RW-00	

位	位名	权限	功能描述
31	ARPE	RW	预装载使能 0：ARR 和 CCR 寄存器不使能 preload 1：ARR 和 CCR 寄存器使能 preload
30	CAPCLR	RW	清零捕捉模式 0：自有计数捕捉模式 1：清零捕捉模式
29	CAPONCE	RW	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	RW	噪声滤波 0：关闭数字滤波 1：打开数字滤波
27:26	CAPEDGE	RW	捕捉源边沿选择 00/11：上升沿和下降沿 01：上升沿

位	位名	权限	功能描述
			10：下降沿
25:24	CNTEDGE	RW	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	RW	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	RW	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	RW	输出极性选择 0：输出不取反 1：输出取反
6:5	TS	RW	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级联模式章节
4:2	CAPSEL	RW	捕捉源选择寄存器 000：外部引脚输入（TIxx） 其他：参见捕捉源选择章节
1:0	CNTSEL	RW	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）

22.4.6 TAU0 通道 x 模式寄存器(TAU0_TxMDR x=[0, 7])

符号	TAU0_TxMDR x=[0, 7]							
地址	0x0020 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		URS	SLVMD	TOEN	SLV	MD	
权限	R-00		RW-0	RW-0	RW-0	RW-0	RW-00	

位	位名	权限	功能描述
31:6	RFU	R	未实现，读为 0
5	URS	RW	更新事件选择寄存器 0：计数器溢出或软件置位 UEV 寄存器产生更新事件 1：只有计数器溢出产生更新事件
4	SLVMD	RW	Slave 模式选择，仅 SLV=1 的情况下有小

位	位名	权限	功能描述
			0：触发递增模式，每个触发信号导致从机计数器加一 1：触发启动模式，每个触发信号导致从机开始计数，直到 ARR 停止。
3	TOEN	RW	TO 输出使能 0：关闭 TO 输出 1：使能 TO 输出
2	SLV	RW	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	RW	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）

22.4.7 TAU0 通道 x 目标值寄存器(TAU0_TxARR x=[0, 7])

符号	TAU0_TxARR x=[0, 7]							
地址	0x0040 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	ARR	RW	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。

22.4.8 TAU0 通道 x 捕捉比较寄存器(TAU0_TxCCR x=[0, 7])

符号	TAU0_TxCCR x=[0, 7]							
地址	0x0060 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	TAU0_TxCCR x=[0, 7]							
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	CCR	RW	捕捉比较寄存器 捕捉模式下，当捕捉

22.4.9 TAU0 通道 x 中断使能寄存器(TAU0_TxIER x=[0, 7])

符号	TAU0_TxIER x=[0, 7]							
地址	0x0080 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					CMPIE	CAPIE	OVIE
权限	R-00000					RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	CMPIE	RW	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	RW	定时器捕捉中断使能 1 = 使能 0 = 禁止
0	OVIE	RW	定时器溢出中断使能 1 = 使能

位	位名	权限	功能描述
			0 = 禁止

22.4.10 TAU0 通道 x 中断标志寄存器(TAU0_TxISR x=[0, 7])

符号	TAU0_TxISR x=[0, 7]							
地址	0x00A0 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				EDGESTA	CMPIF	CAPIF	OVIF
权限	R-0000				W1C-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:4	RFU	R	未实现：读为 0
3	EDGESTA	W1C	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	W1C	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	W1C	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	W1C	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

22.4.11 TAU0 通道 x 计数值寄存器(TAU0_TxCNT x=[0, 7])

符号	TAU0_TxCNT x=[0, 7]							
地址	0x00C0 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

符号	TAU0_TxCNT x=[0, 7]							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	CNT	R	计数器当前计数值，只读

22.4.12 TAU1 通道 x 配置寄存器(TAU1_TxCFGR x=[0, 3])

符号	TAU1_TxCFGR x=[0, 3]							
地址	0x0000 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARPE	CAPCLR	CAPONCE	NF	CAPEDGE		CNTEDGE	
权限	RW-0	RW-0	RW-0	RW-0	RW-00		RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PRESCALE1							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PRESCALE2							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	OPOL	TS		CAPSEL			CNTSEL	
权限	RW-0	RW-00		RW-000			RW-00	

位	位名	权限	功能描述
31	ARPE	RW	预装载使能 0：ARR 和 CCR 寄存器不使能 preload 1：ARR 和 CCR 寄存器使能 preload
30	CAPCLR	RW	清零捕捉模式 0：自有计数捕捉模式 1：清零捕捉模式
29	CAPONCE	RW	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	RW	噪声滤波 0：关闭数字滤波 1：打开数字滤波

位	位名	权限	功能描述
27:26	CAPEDGE	RW	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	RW	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	RW	计数源预分频，分频值=PRESCALE1 + 1
15:8	PRESCALE2	RW	捕捉源预分频，分频值=PRESCALE2 + 1
7	OPOL	RW	输出极性选择 0：输出不取反 1：输出取反
6:5	TS	RW	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级联模式章节
4:2	CAPSEL	RW	捕捉源选择寄存器 000：外部引脚输入（TIxx） 其他：参见捕捉源选择章节
1:0	CNTSEL	RW	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）

22.4.13 TAU1 通道 x 模式寄存器(TAU1_TxMDR x=[0, 3])

符号	TAU1_TxMDR x=[0, 3]							
地址	0X0010 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		URS	SLVMD	TOEN	SLV	MD	
权限	R-00		RW-0	RW-0	RW-0	RW-0	RW-00	

位	位名	权限	功能描述
31:6	RFU	R	未实现，读为 0
5	URS	RW	更新事件选择寄存器 0：计数器溢出或软件置位 UEV 寄存器产生更新事

位	位名	权限	功能描述
			件 1：只有计数器溢出产生更新事件
4	SLVMD	RW	Slave 模式选择，仅 SLV=1 的情况下有小 0：触发递增模式，每个触发信号导致从机计数器加一 1：触发启动模式，每个触发信号导致从机开始计数，直到 ARR 停止。
3	TOEN	RW	TO 输出使能 0：关闭 TO 输出 1：使能 TO 输出
2	SLV	RW	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	RW	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）

22.4.14 TAU1 通道 x 目标值寄存器(TAU1_TxARR x=[0, 3])

符号	TAU1_TxARR x=[0, 3]							
地址	0x0020 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	ARR	RW	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。

22.4.15 TAU1 通道 x 捕捉比较寄存器(TAU1_TxCCR x=[0, 3])

符号	TAU1_TxCCR x=[0, 3]							
地址	0x0030 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	CCR	RW	捕捉比较寄存器 捕捉模式下，当捕捉

22.4.16 TAU1 通道 x 中断使能寄存器(TAU1_TxIER x=[0, 3])

符号	TAU1_TxIER x=[0, 3]							
地址	0x0040 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					CMPIE	CAPIE	OVIE
权限	R-00000					RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	CMPIE	RW	定时器比较中断使能 1 = 使能 0 = 禁止
1	CAPIE	RW	定时器捕捉中断使能

位	位名	权限	功能描述
			1 = 使能 0 = 禁止
0	OVIE	RW	定时器溢出中断使能 1 = 使能 0 = 禁止

22.4.17 TAU1 通道 x 中断标志寄存器(TAU1_TxISR x=[0, 3])

符号	TAU1_TxISR x=[0, 3]							
地址	0x0050 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				EDGESTA	CMPIF	CAPIF	OVIF
权限	R-0000				W1C-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:4	RFU	R	未实现：读为 0
3	EDGESTA	W1C	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	W1C	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	W1C	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	W1C	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

22.4.18 TAU1 通道 x 计数值寄存器(TAU1_TxCNT x=[0, 3])

符号	TAU1_TxCNT x=[0, 3]							
地址	0x0060 + 0x04*x							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	TAU1_TxCNT x=[0, 3]							
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:16	RFU	R	未实现，读为 0
15:0	CNT	R	计数器当前计数值，只读

23 16 位基本定时器（BSTIM16）

23.1 概述

本芯片包含 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。

23.2 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

23.3 结构框图

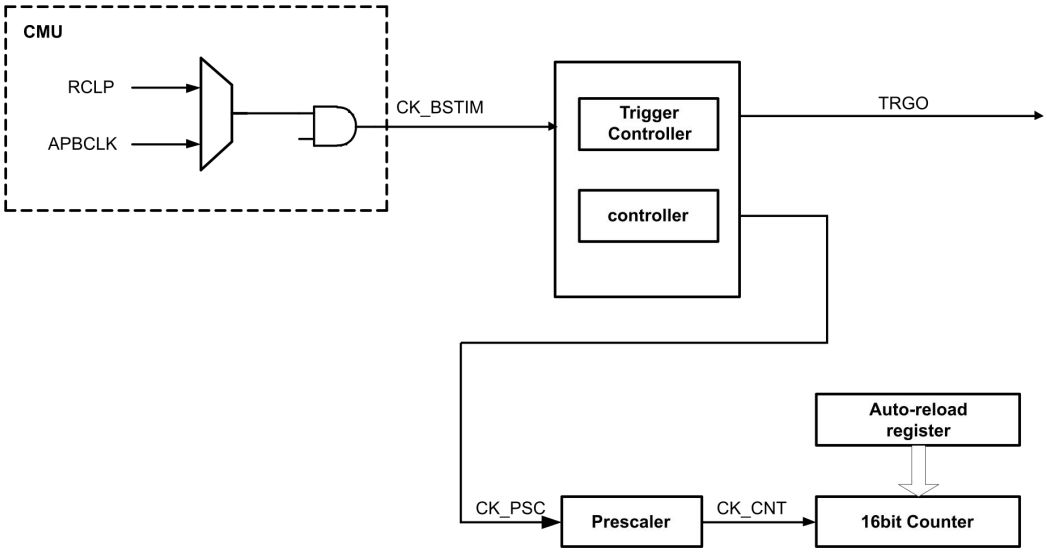


图 23-1 BTIM16 结构框图

23.4 功能描述

23.4.1 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 位预分频器对工作时钟进行分频后得到。

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。

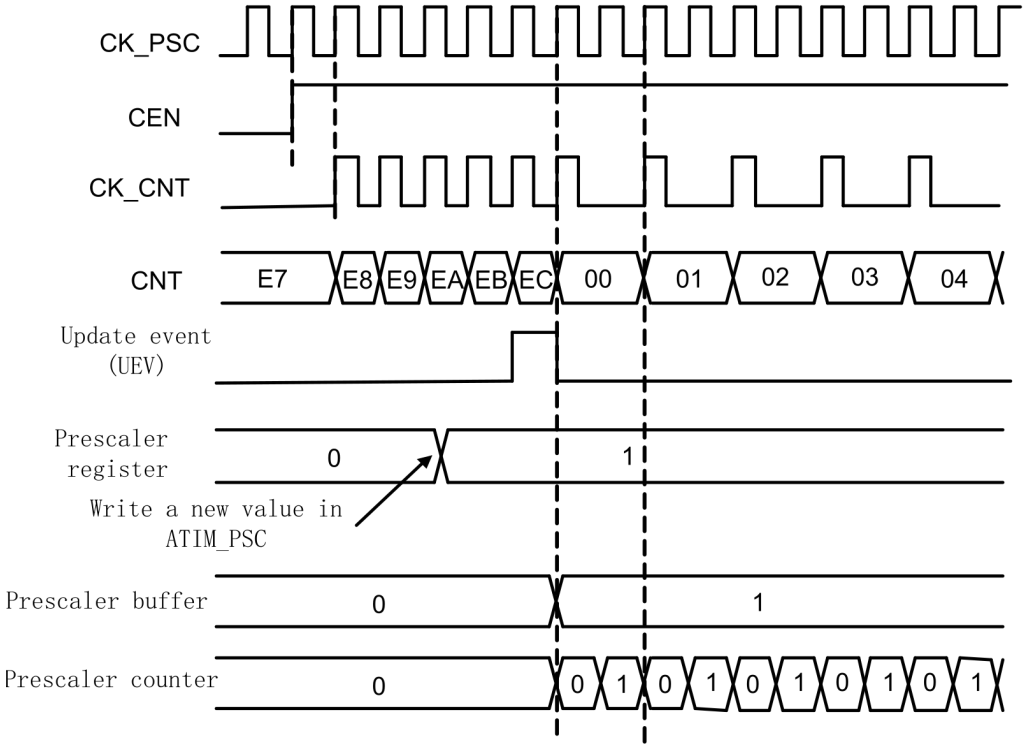


图 23-2 BTIM16 预分频从 1 变为 2 的波形

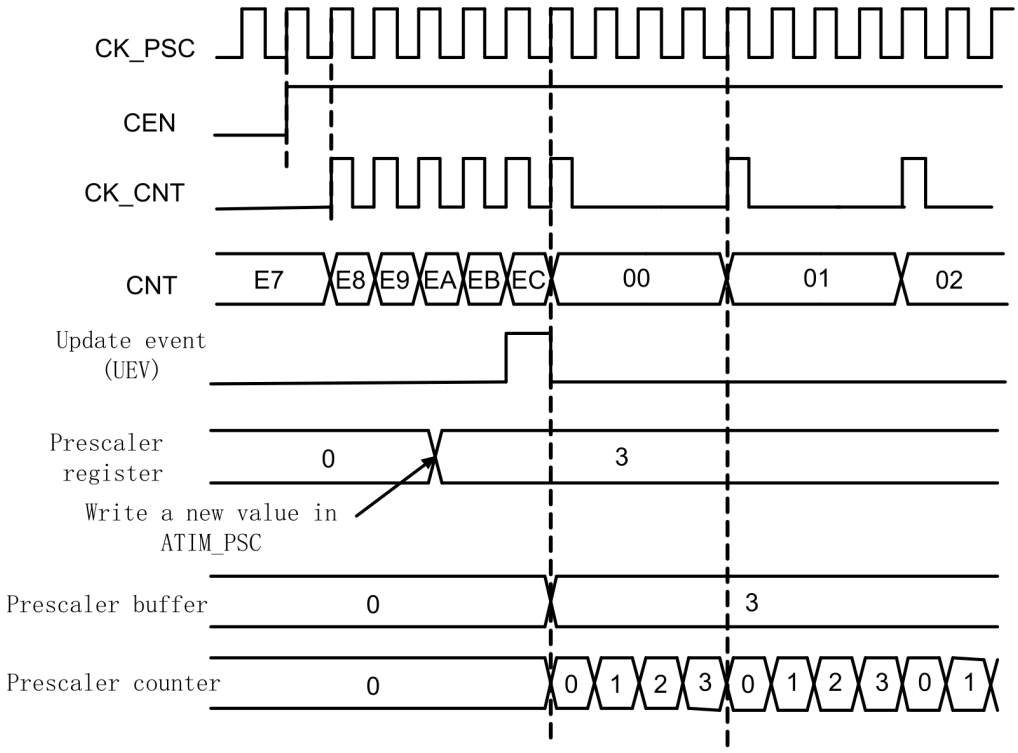


图 23-3 BTIM16 预分频从 1 变为 4 的波形

23.4.2 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 寄存器是否触发 UIF 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：

- BSTIM_ARR 更新为缓存中的值
- BSTIM_PSC 更新为缓存中的值

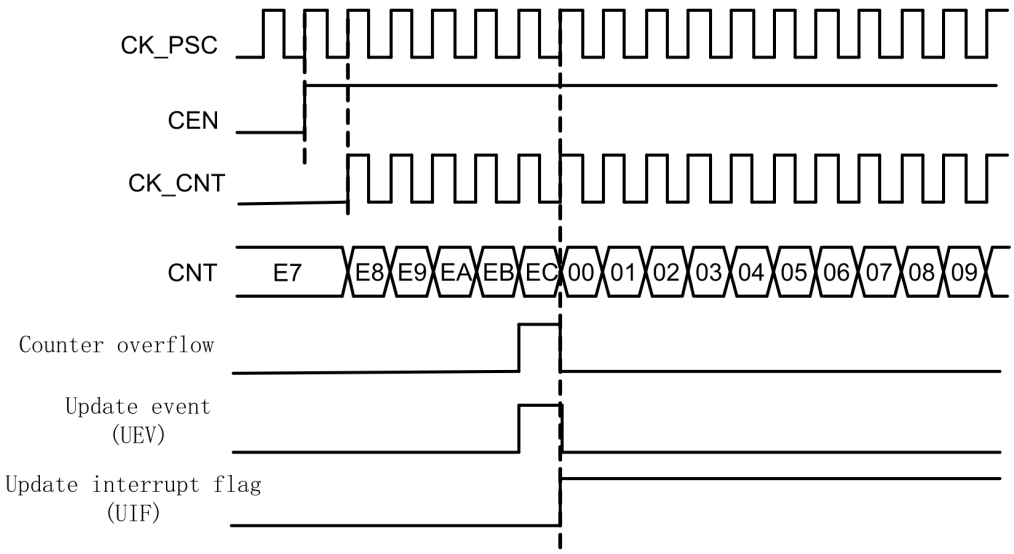


图 23-4 BTIM16 向上计数不分频的波形

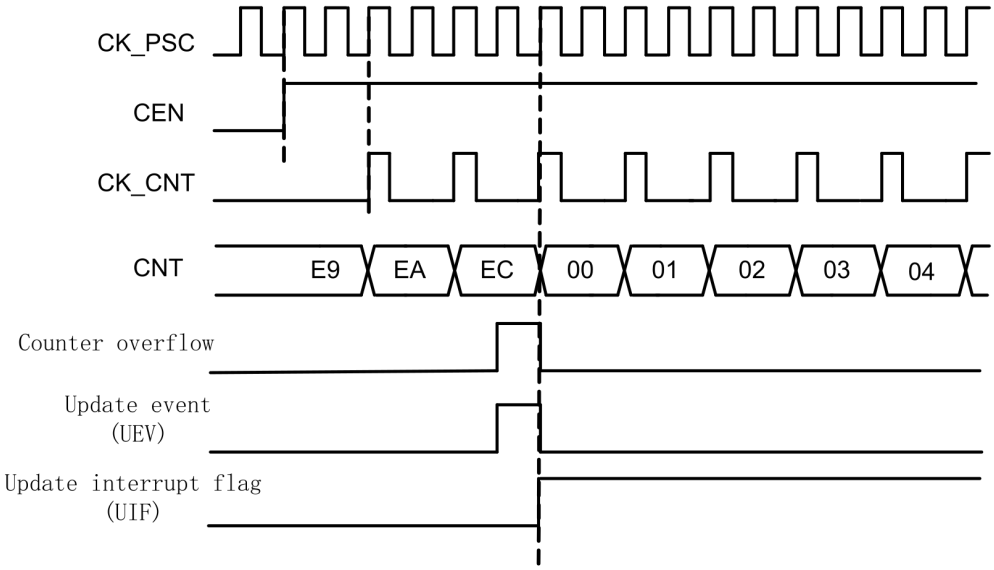


图 23-5 BTIM16 向上计数 2 分频的波形

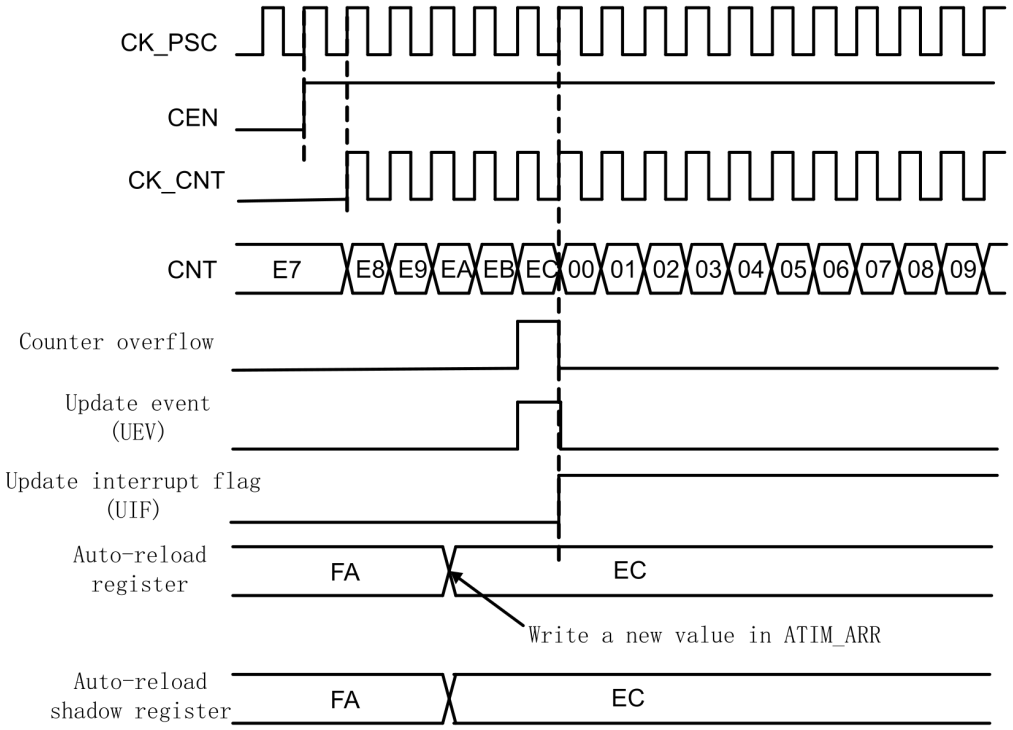


图 23-6 BTIM16 无预装载(ARPE=0)波形

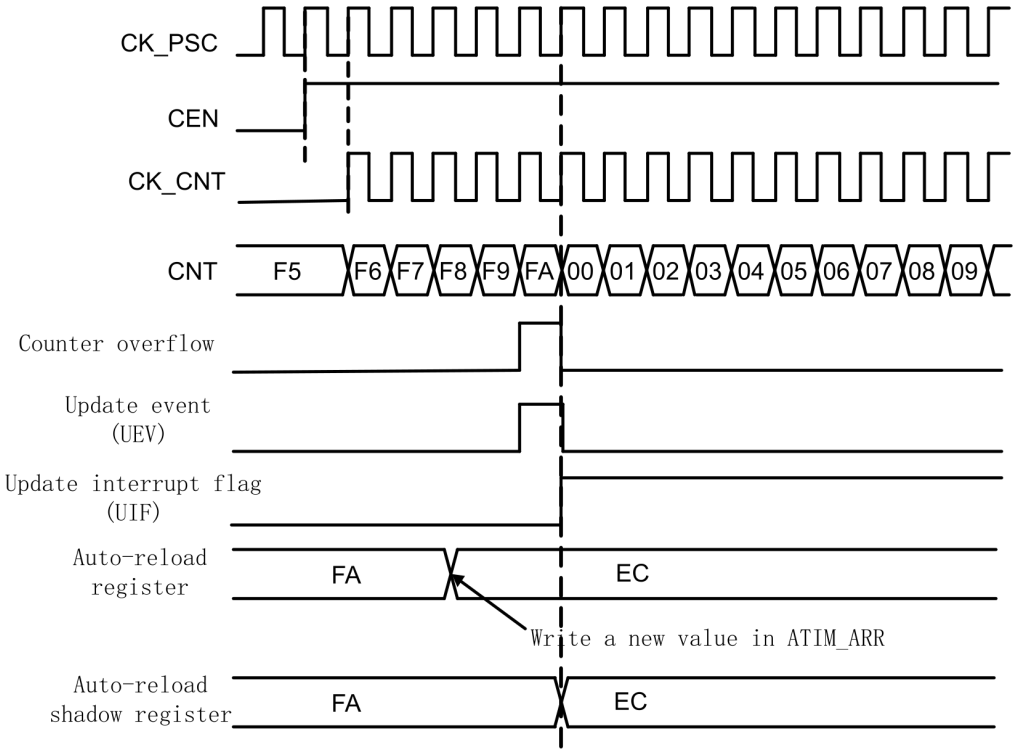


图 23-7 BTIM16 预装载(ARPE=1)波形

23.4.3 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 信号经过内部时钟同步后，计数器值将被重新初始化。

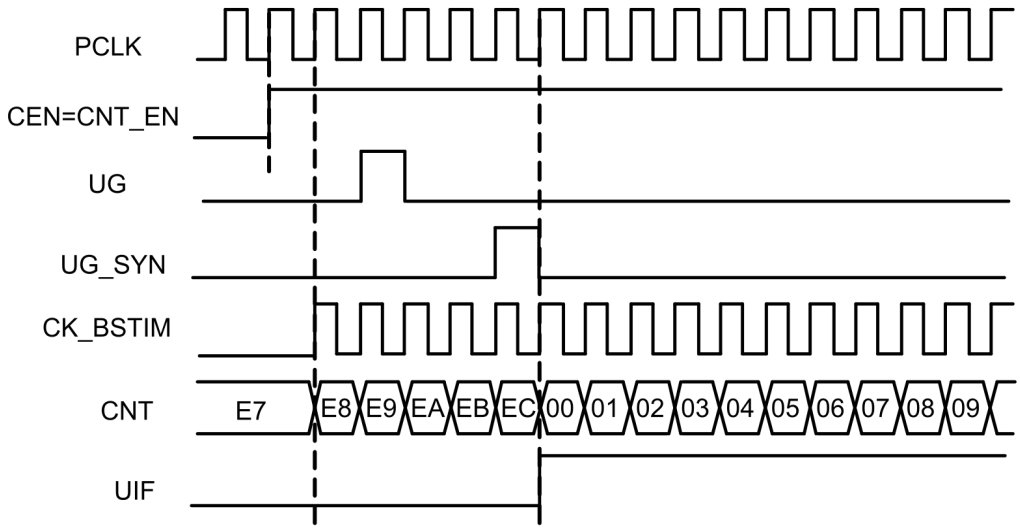


图 23-8 BTIM16 的 UG 初始化波形

23.4.4 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。

23.5 寄存器

BSTIM16 基地址 (0x4001B400)

地址	符号	名称
0x0000	BSTIM16_CR1	BSTIM 控制寄存器 1
0x0004	BSTIM16_CR2	BSTIM 控制寄存器 2
0x000C	BSTIM16_IER	BSTIM 中断使能寄存器
0x0010	BSTIM16_ISR	BSTIM 中断标志寄存器
0x0014	BSTIM16_EGR	BSTIM 事件产生寄存器
0x0024	BSTIM16_CNT	BSTIM 计数器寄存器
0x0028	BSTIM16_PSC	BSTIM 预分频寄存器
0x002C	BSTIM16_ARR	BSTIM 自动重载寄存器

23.5.1 BSTIM 控制寄存器 1(BSTIM16_CR1)

符号	BSTIM16_CR1							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

符号	BSTIM16_CR1							
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARPE	RFU1			OPM	URS	UDIS	CEN
权限	RW-0	R-000			RW-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:8	RFU	R	RFU，未实现，读为 0
7	ARPE	RW	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	RFU1	R	RFU，未实现，读为 0
3	OPM	RW	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	RW	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
1	UDIS	RW	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	RW	计数器使能 0：计数器关闭 1：计数器使能

23.5.2 BSTIM 控制寄存器 2(BSTIM16_CR2)

符号	BSTIM16_CR2
地址	0x0004

符号	BSTIM16_CR2							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	MMS			RFU1			
权限	R-0	RW-000			R-0000			

位	位名	权限	功能描述
31:7	RFU	R	RFU，未实现，读为 0
6:4	MMS	RW	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
3:0	RFU1	R	RFU，未实现，读为 0

23.5.3 BSTIM 中断使能寄存器(BSTIM16_IER)

符号	BSTIM16_IER							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							UIE
权限	R-0000000							RW-0

位	位名	权限	功能描述

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	UIE	RW	Update 事件中断使能 0：禁止 Update 事件中断 1：允许 Update 事件中断

23.5.4 BSTIM 中断标志寄存器(BSTIM16_ISR)

符号	BSTIM16_ISR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							UIF
权限	R-0000000							W1C-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	UIF	W1C	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

23.5.5 BSTIM 事件产生寄存器(BSTIM16_EGR)

符号	BSTIM16_EGR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	BSTIM16_EGR							
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							UG
权限	R-0000000							R-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	UG	R	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。

23.5.6 BSTIM 计数器寄存器(BSTIM16_CNT)

符号	BSTIM16_CNT							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CNT							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	CNT							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	CNT	RW	计数器值

23.5.7 BSTIM 预分频寄存器(BSTIM16_PSC)

符号	BSTIM16_PSC							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PSC							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PSC							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PSC							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PSC							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	PSC	RW	计数器时钟（CK_CNT）预分频值 $f_{CK_CNT}=f_{CK_PSC}/(PSC[31:0]+1)$ 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

23.5.8 BSTIM 自动重载寄存器(BSTIM16_ARR)

符号	BSTIM16_ARR							
地址	0x002C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	ARR							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	ARR							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-00000000							

位	位名	权限	功能描述
31:0	ARR	RW	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 16 位低功耗定时器（LPTIM16）

24.1 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

24.2 结构框图

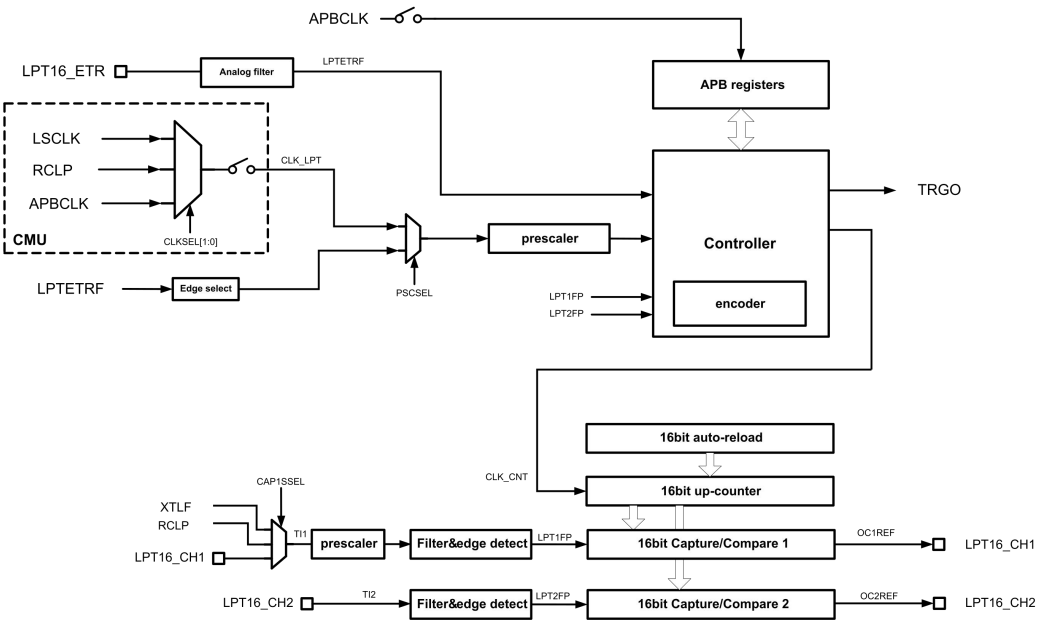


图 24-1 LPTIM16 结构框图

24.3 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 7 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。

24.4 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。

24.4.1 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。

24.4.2 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。

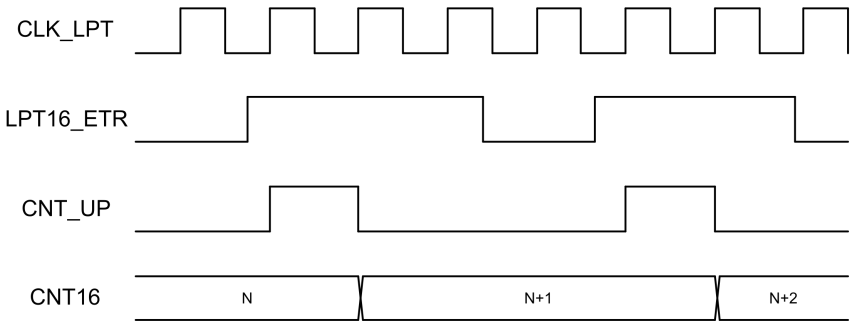


图 24-2 LPTIM16 外部 ETR 脉冲上升沿触发计数

24.4.3 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。

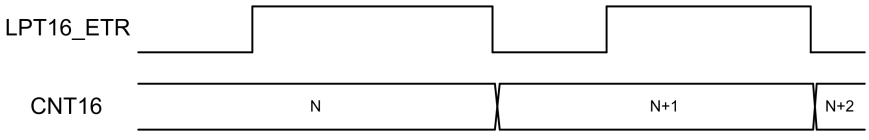


图 24-3 LPTIM16 外部 ETR 脉冲异步计数（下降沿）

24.4.4 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。

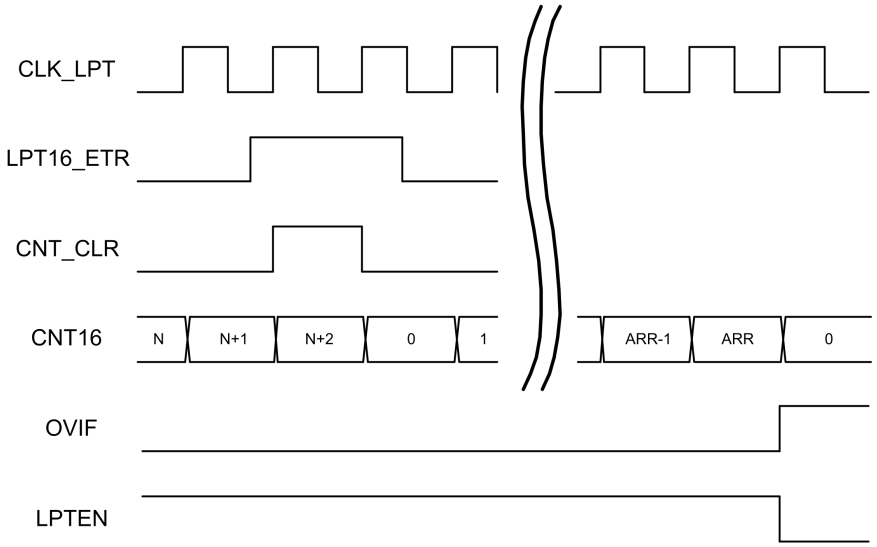


图 24-4 LPTIM16_TimeOut 模式

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。

24.5 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。

24.5.1 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。

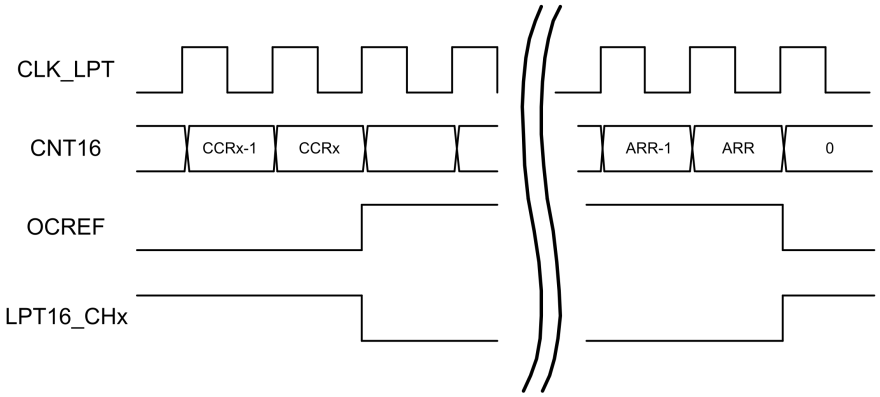


图 24-5 LPTIM16_PWM 输出

24.5.2 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。

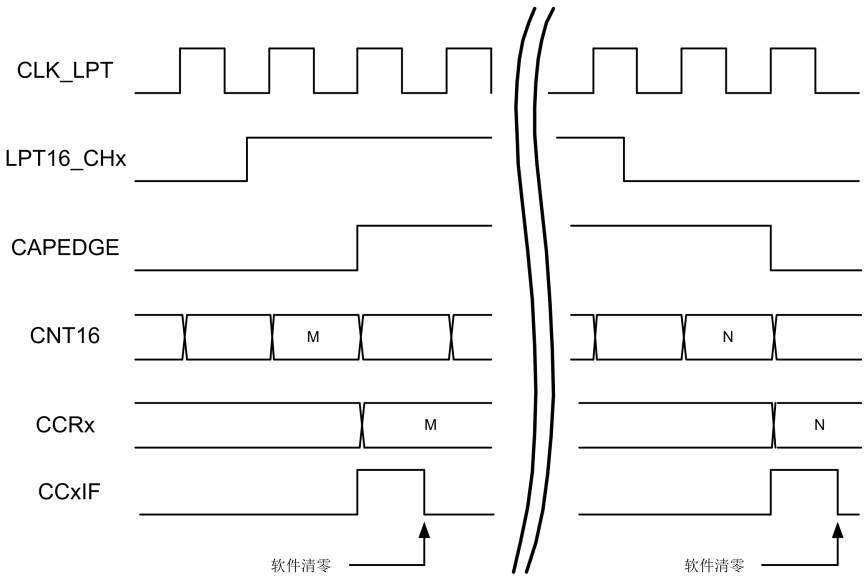


图 24-6 LPTIM16 输入信号边沿捕捉

24.5.3 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。

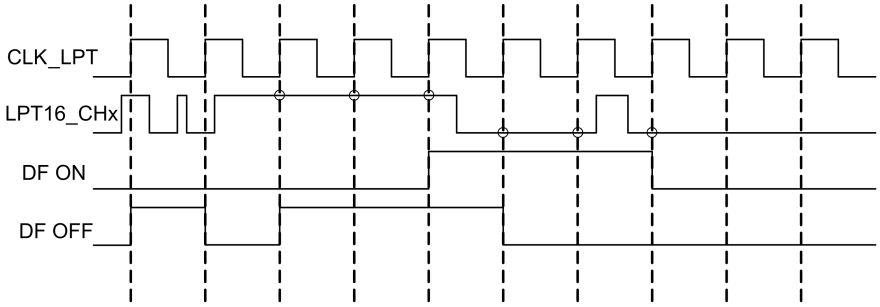


图 24-7 LPTIM16 通道输入数字滤波

24.6 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：

表 24-1 encoder interface 计数方式

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	高	递减	递增	不计数	不计数

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
处计数	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

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。

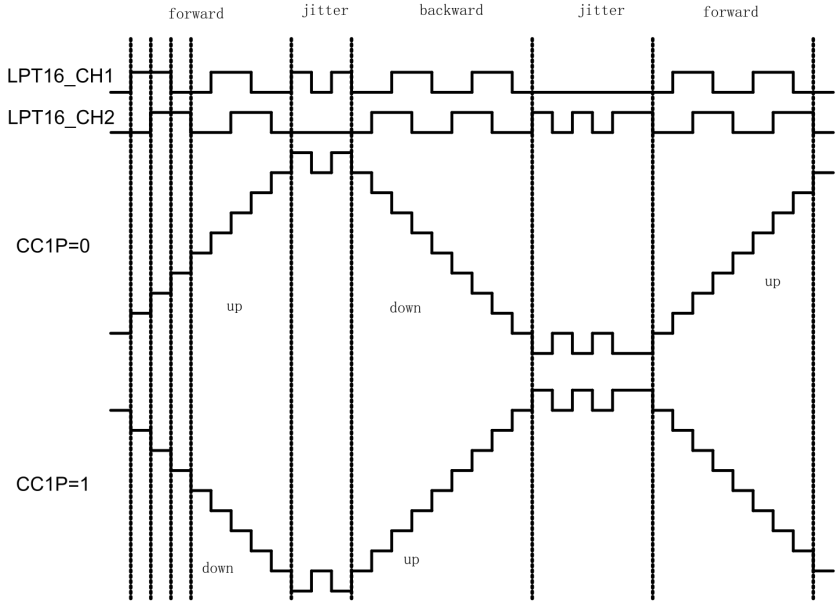


图 24-8 LPTIM16 编码器模式下的计数器

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

24.7 寄存器

LPTIM16 基地址 (0x40013400)

地址	符号	名称
0x0000	LPTIM16_CFGR	LPTIM 配置寄存器
0x0004	LPTIM16_CNT	LPTIM 计数值寄存器
0x0008	LPTIM16_CCSR	LPTIM 捕捉比较控制和状态寄存器
0x000C	LPTIM16_ARR	LPTIM 目标值寄存器
0x0010	LPTIM16_IER	LPTIM 中断使能寄存器
0x0014	LPTIM16_ISR	LPTIM 中断标志寄存器
0x0018	LPTIM16_CR	LPTIM 控制寄存器
0x0020	LPTIM16_CCR1	LPTIM 比较捕捉寄存器 1
0x0024	LPTIM16_CCR2	LPTIM 比较捕捉寄存器 2

24.7.1 LPTIM 配置寄存器(LPTIM16_CFGR)

符号	LPTIM16_CFGR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							ETR_AFEN
权限	R-0000000							RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					MMS		
权限	R-00000					RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	PSCSEL	RFU	DIVSEL			RFU	
权限	R-0	RW-0	R-0	RW-000			R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EDGESEL	TRIGCFG		QEMD		ONST	TMODE	
权限	RW-0	RW-00		RW-00		RW-0	RW-00	

位	位名	权限	功能描述
31:25	RFU	R	RFU：未实现，读为 0
24	ETR_AFEN	RW	LPT16_ETR 输入模拟滤波使能 0：关闭模拟滤波 1：使能模拟滤波，滤波宽度约 100ns
23:19	RFU	R	RFU：未实现，读为 0
18:16	MMS	RW	主机模式选择，用于配置主机模式下向从机发送的

位	位名	权限	功能描述
			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	RFU	R	RFU：未实现，读为 0
14	PSCSEL	RW	时钟预分频输入选择 0：CLKSEL 选择的时钟 1：LPTINSEL 选择的信号
13	RFU	R	RFU：未实现，读为 0
12:10	DIVSEL	RW	计数时钟分频选择 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	RFU	R	RFU：未实现，读为 0
7	EDGESEL	RW	LPT16_ETR 输入边沿选择 0：LPT16_ETR 的上升沿计数 1：LPT16_ETR 的下降沿计数
6:5	TRIGCFG	RW	外部触发边沿选择（需使用内部时钟同步采样 LPT16_ETR） 00: LPT16_ETR 输入信号上升沿触发 01: LPT16_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	RW	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 LPT161FP 电平高低来计数 10：Encoder 模式 2；计数器使用 LPT1FP 边沿，根据 LPT162FP 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
2	ONST	RW	单次计数模式使能 0：连续计数模式：计数器被触发后保持运行，直

位	位名	权限	功能描述
			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	RW	工作模式选择 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

24.7.2 LPTIM 计数值寄存器(LPTIM16_CNT)

符号	LPTIM16_CNT							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CNT							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CNT							
权限	R-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CNT	R	计数器当前计数值

24.7.3 LPTIM 捕捉比较控制和状态寄存器(LPTIM16_CCSR)

符号	LPTIM16_CCSR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	CAP1PSC						CAP1SSEL	
权限	RW-000000						RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		CAP2EDGE	CAP1EDGE	RFU		CC2P	CC1P
权限	R-00		R-0	R-0	R-00		RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		CC2DF	CC1DF	CAPCFG2		CAPCFG1	
权限	R-00		RW-0	RW-0	RW-00		RW-00	

符号	LPTIM16_CCSR							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				CC2S		CC1S	
权限	R-0000				RW-00		RW-00	

位	位名	权限	功能描述
31:26	CAP1PSC	RW	通道 1 输入预分频 (Capture channel 1 prescaler)，2*CAP1PSC 分频 0x00：不分频 0x3F：128 分频
25:24	CAP1SSEL	RW	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：RFU 10：RCLP 11：RFU
23:22	RFU	R	RFU：未实现，读为 0
21	CAP2EDGE	R	通道 2 当前被捕捉的边沿，在 CC2IF 置位时更新 (Channel 2 Captured Edge) 0：上升沿 1：下降沿
20	CAP1EDGE	R	通道 1 当前被捕捉的边沿，在 CC1IF 置位时更新 (Channel 1 Captured Edge) 0：上升沿 1：下降沿
19:18	RFU	R	RFU：未实现，读为 0
17	CC2P	RW	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	RW	通道 1 配置为比较功能时：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	RFU	R	RFU：未实现，读为 0
13	CC2DF	RW	通道 2 输入数字滤波 (Channel 2 input digital filter)

位	位名	权限	功能描述
			1：使能数字滤波 0：关闭数字滤波
12	CC1DF	RW	通道 1 输入数字滤波 (Channel 1 input digital filter) 1：使能数字滤波 0：关闭数字滤波
11:10	CAPCFG2	RW	通道 2 捕捉边沿选择 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	RW	通道 1 捕捉边沿选择 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：RFU
7:4	RFU	R	RFU：未实现，读为 0
3:2	CC2S	RW	通道 2 捕捉/比较功能使能 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	RW	通道 1 捕捉/比较功能使能 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）

24.7.4 LPTIM 目标值寄存器(LPTIM16_ARR)

符号	LPTIM16_ARR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ARR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ARR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	ARR	RW	自动重载目标寄存器

位	位名	权限	功能描述
			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

24.7.5 LPTIM 中断使能寄存器(LPTIM16_IER)

符号	LPTIM16_IER							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU						OVR2IE	OVR1IE
权限	R-000000						RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIE	OVIE	QDCIE	RFU			CC2IE	CC1IE
权限	RW-0	RW-0	RW-0	R-000			RW-0	RW-0

位	位名	权限	功能描述
31:10	RFU	R	RFU：未实现，读为 0
9	OVR2IE	RW	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	RW	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	RW	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	RW	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	RW	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	RFU	R	RFU：未实现，读为 0
1	CC2IE	RW	捕捉/比较通道 2 中断使能位(Capture/Compare channel 2 Interrupt Enable)

位	位名	权限	功能描述
			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	RW	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

24.7.6 LPTIM 中断标志寄存器(LPTIM16_ISR)

符号	LPTIM16_ISR							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU		QDIR	RFU			CAP2OVR	CAP1OVR
权限	R-00		R-0	R-000			W1C-0	W1C-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRIGIF	OVIF	QDCIF	RFU			CC2IF	CC1IF
权限	W1C-0	W1C-0	W1C-0	R-000			W1C-0	W1C-0

位	位名	权限	功能描述
31:14	RFU	R	RFU：未实现，读为 0
13	QDIR	R	Quadrature Decoder 方向方向寄存器(Quad-decoder Direction) 0：CH1 领先于 CH2 1：CH1 滞后于 CH2
12:10	RFU	R	RFU：未实现，读为 0
9	CAP2OVR	W1C	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	W1C	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	W1C	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

位	位名	权限	功能描述
6	OVIF	W1C	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	W1C	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	RFU	R	RFU：未实现，读为 0
1	CC2IF	W1C	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	W1C	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

24.7.7 LPTIM 控制寄存器(LPTIM16_CR)

符号	LPTIM16_CR							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							EN
权限	R-0000000							RW-0

位	位名	权限	功能描述
31:1	RFU	R	RFU：未实现，读为 0
0	EN	RW	LPTIM 使能位 1：使能计数器计数 0：禁止计数器计数

24.7.8 LPTIM 比较捕捉寄存器 1(LPTIM16_CCR1)

符号	LPTIM16_CCR1

符号	LPTIM16_CCR1							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR1							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR1							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR1	RW	比较捕捉值寄存器 1 作为比较值寄存器（CC1S！=0x1）时，可读写 作为捕捉值寄存器（CC1S==0x1）时，只读

24.7.9 LPTIM 比较捕捉寄存器 2(LPTIM16_CCR2)

符号	LPTIM16_CCR2							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	CCR2							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CCR2							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	CCR2	RW	比较捕捉值寄存器 2 作为比较值寄存器（CC2S！=0x1）时，可读写 作为捕捉值寄存器（CC2S==0x1）时，只读

25 模数转换器（ADC）

25.1 概述

本芯片带有一个 1Msps 12bit SAR-ADC，可实现温度、电池电压或其他直流信号的测量功能。主要特点为：

- 工作电压 1.8～5.5V
- 输入信号幅度 0~VDD
- 最高采样率 1Msps（ADCCLK=16MHz）
- 16 个输入通道，包含内部基准电压、温度传感器、14 个外部通道
- 可配置的采样保持时间
- 支持单次转换和连续转换
- 支持 DMA
- 支持过采样硬件平均，最高 16bit 输出（256 次硬件平均）

25.2 结构框图

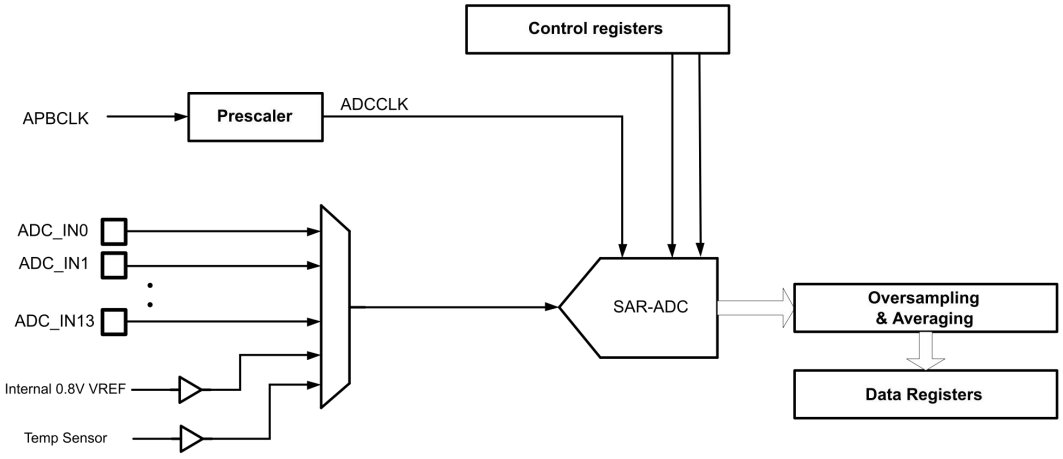


图 25-1 ADC 结构框图

25.3 输入通道

ADC 支持 2 个内部通道和 14 个外部通道。

表 25-1 ADC 输入通道

通道	说明
AN0~13	PAD，单端输入，来自于 PAD，具体引脚功能参见引脚和封装
AN14	AVREF,内部 1.0V 基准源采样通道
AN15	TempSensor，温度传感器采样通道

25.4 ADC 时钟和复位

ADC 使用 APBCLK 工作， ADCCLK 由 APBCLK 经过分频后产生。注意分频后的 ADCCLK 最高频率不能超过 16MHz。

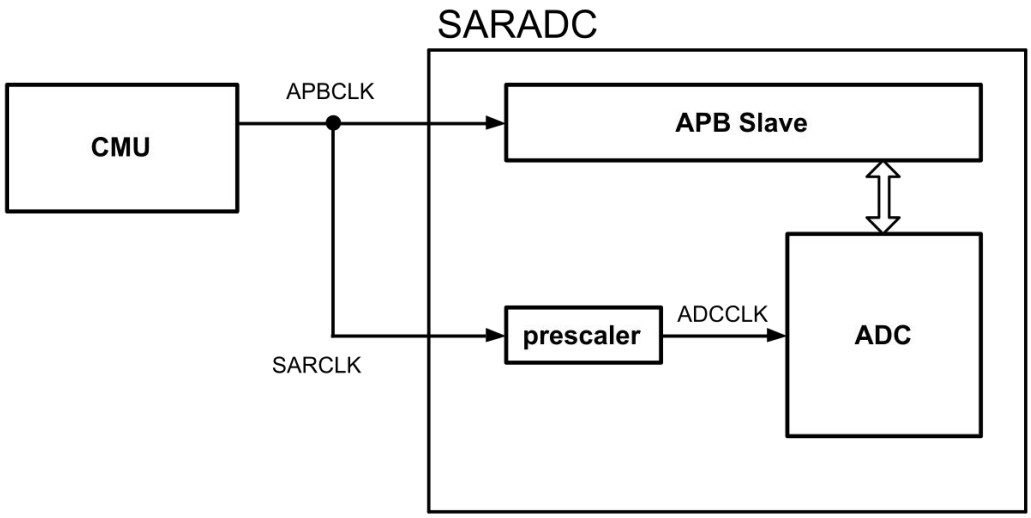


图 25-2 SAR ADC 时钟结构图

ADC 数字模块受 RMU 模块的 ADCRST 寄存器复位，模拟模块受 RMU 模块的 ADCCRST 寄存器复位。

25.5 功能描述

25.5.1 使用 VDD 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。

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%。快速基准源的启动速度很快，典型条件下可以在 5us 内完成启动和输出电压建立。

- 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）：

$$VDD = \frac{4095}{AVREF_DATA} \times 1.0V$$

- 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）

$$V_{CHANNEL} = \frac{ADC_DATA}{AVREF_DATA} \times 1.0V$$

25.5.2 温度传感器

根据温度传感器采样值计算当前温度

芯片测量温度前，需要置位 PTAT_EN 和 VPTATBUFFER_EN 寄存器。

ADC 内部通道选择温度传感器通道，得到转换数据 TS_DATA，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

$$VPTAT = \frac{VREFIN_CAL \times TS_DATA}{VREFINT_DATA \times 4095} \times 3V$$

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

$$VPTAT_30C = \frac{TS_CAL1}{4095} \times 3V$$

3. 根据温度传感器输出斜率计算当前绝对温度

$$Temperature = \frac{VPTAT - VPTAT_30C}{3.06mV/C} + 30C$$

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。

Slope 表示温度传感器输出斜率，典型值为 3.06mV/°C，温度传感器输出的详细特性参见电参数的温度传感器章节。

25.5.3 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 SMTS 寄存器可以选择采样时间：

表 25-2 ADC 采样时间

SMTS	Sampling cycles
0000	4
0001	6
0010	8
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160
1011	192
1100	256
1101	320
1110	384
1111	512

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：

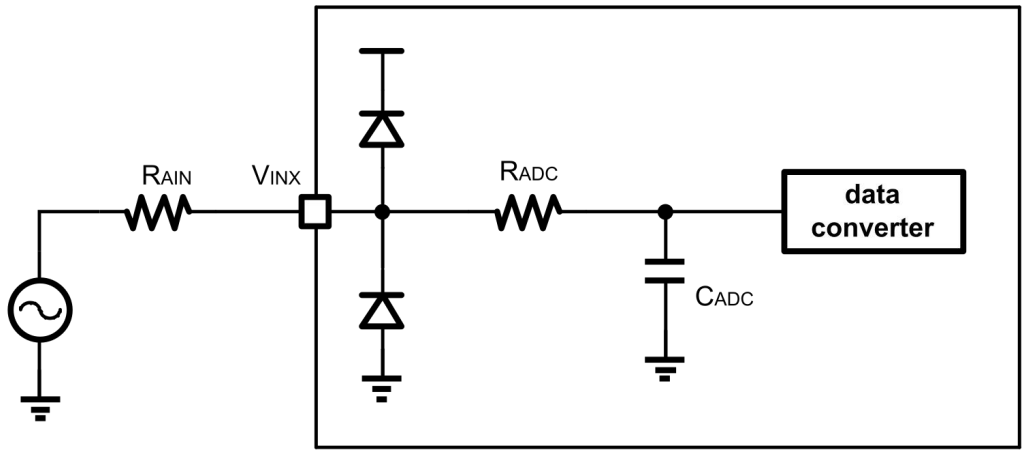


图 25-3 ADC 输入通道示意图

要求的采样时间可以根据下式估算：

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

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。

25.5.4 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。

25.5.4.1 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

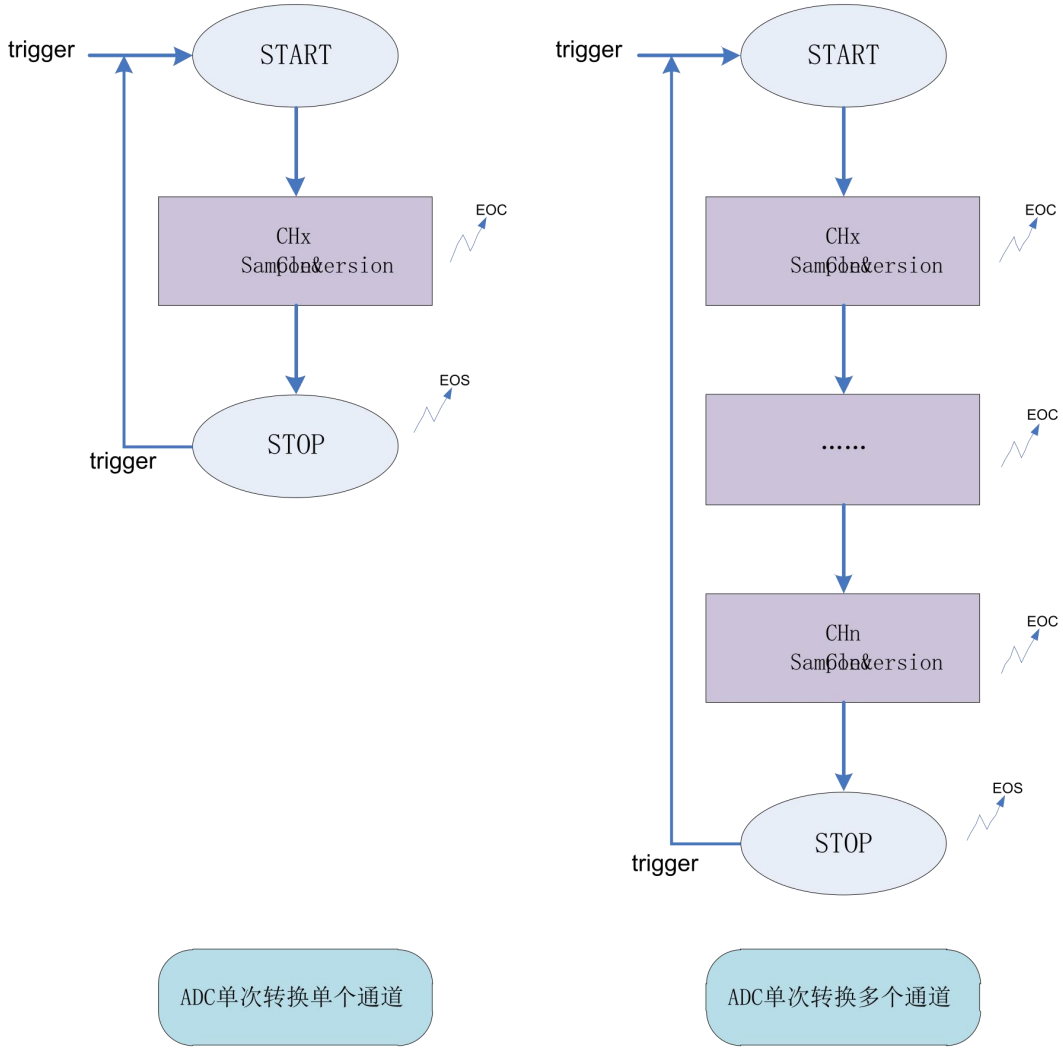


图 25-4 ADC 单次转换全自动触发模式

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

……

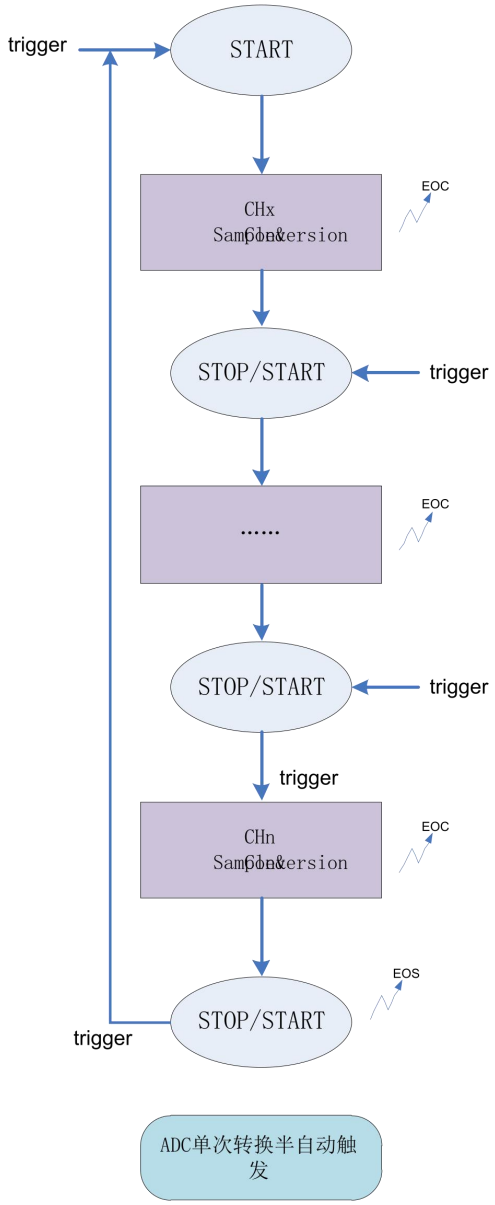


图 25-5 ADC 单次转换半自动触发模式

25.5.4.2 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、CONT_MOD 写 0。

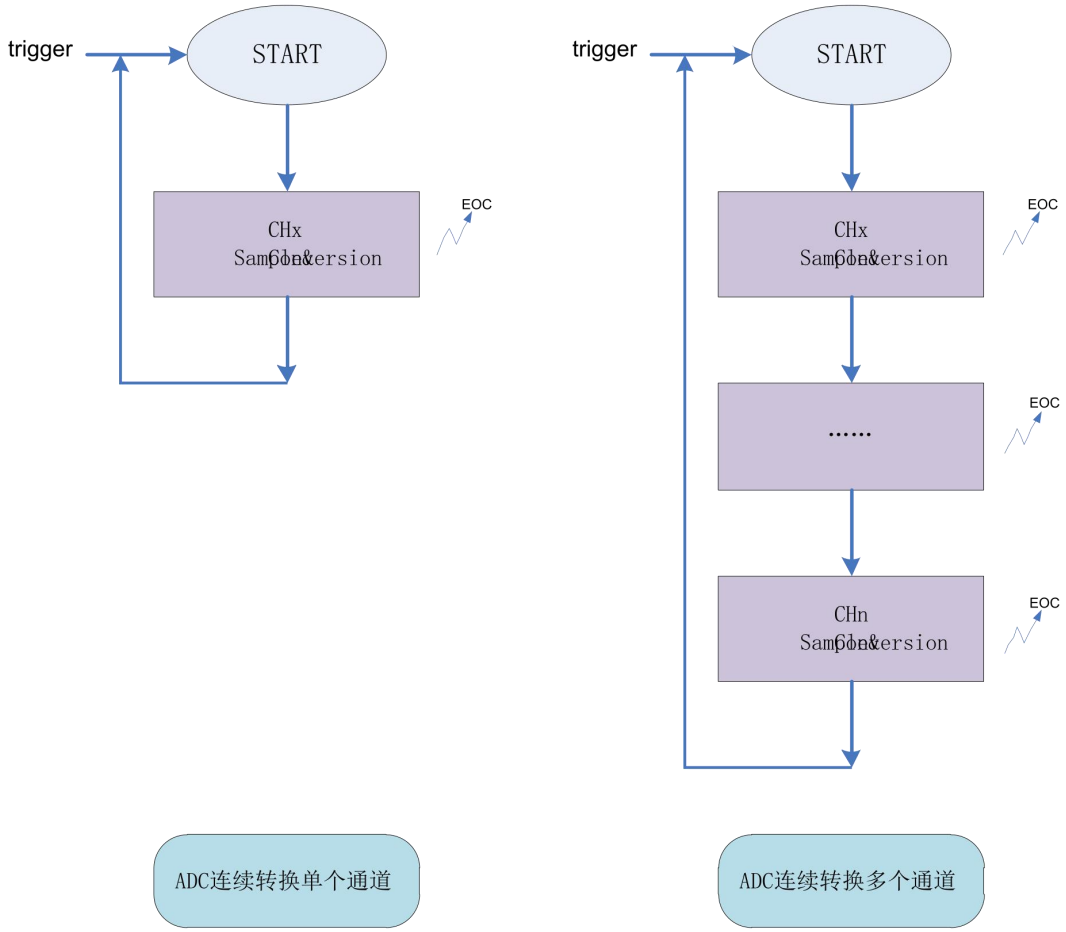


图 25-6 ADC 连续转换模式

每个通道被采样后，数据保存在 ADC_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。

RPT_LEN=0 时则连续转换模式不会自动停止，直到 ADC_CFGR2.CONT_MOD 写 0 则停止。

25.5.5 触发源

ADC 使能后，转换触发支持软件或硬件事件触发，受 TRGCFG 寄存器控制。从 AD 被触发启动到最后一个通道开始采样，期间忽略新的触发事件。

硬件触发源由 EXTS 寄存器选择，触发沿受 TRGCFG 寄存器控制，参见寄存器章节。

25.5.6 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，应用程序和 DMA 无需对同一个通道做多次数据处理。

25.5.7 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_DR 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

ADC 支持自动等待，如果 WAIT 寄存器被软件置位，那么在 ADC_DR 寄存器被读取之前，ADC 不会发起新的转换。

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：

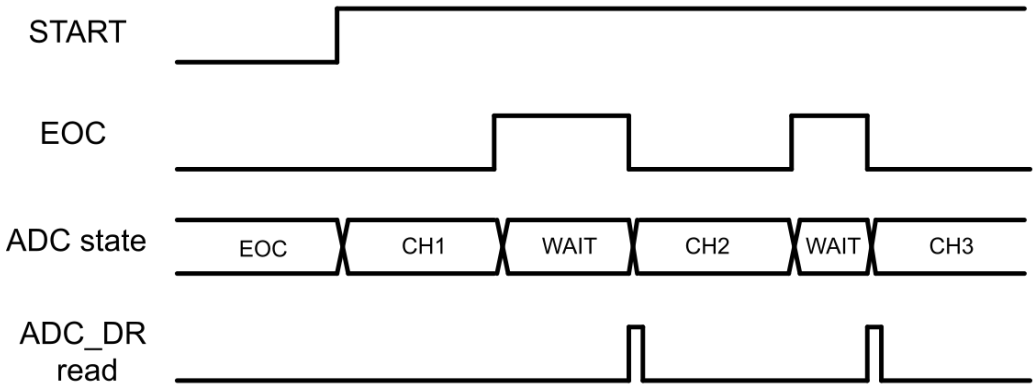


图 25-7 ADC 自动等待

25.5.8 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。

注意，DMAEN 使能后软件不可进入 EOC 中断读取 ADC_DR 寄存器，否则可能会导致 DMA 少搬运一次数据。

25.5.8.1 DMA 使用案例

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。

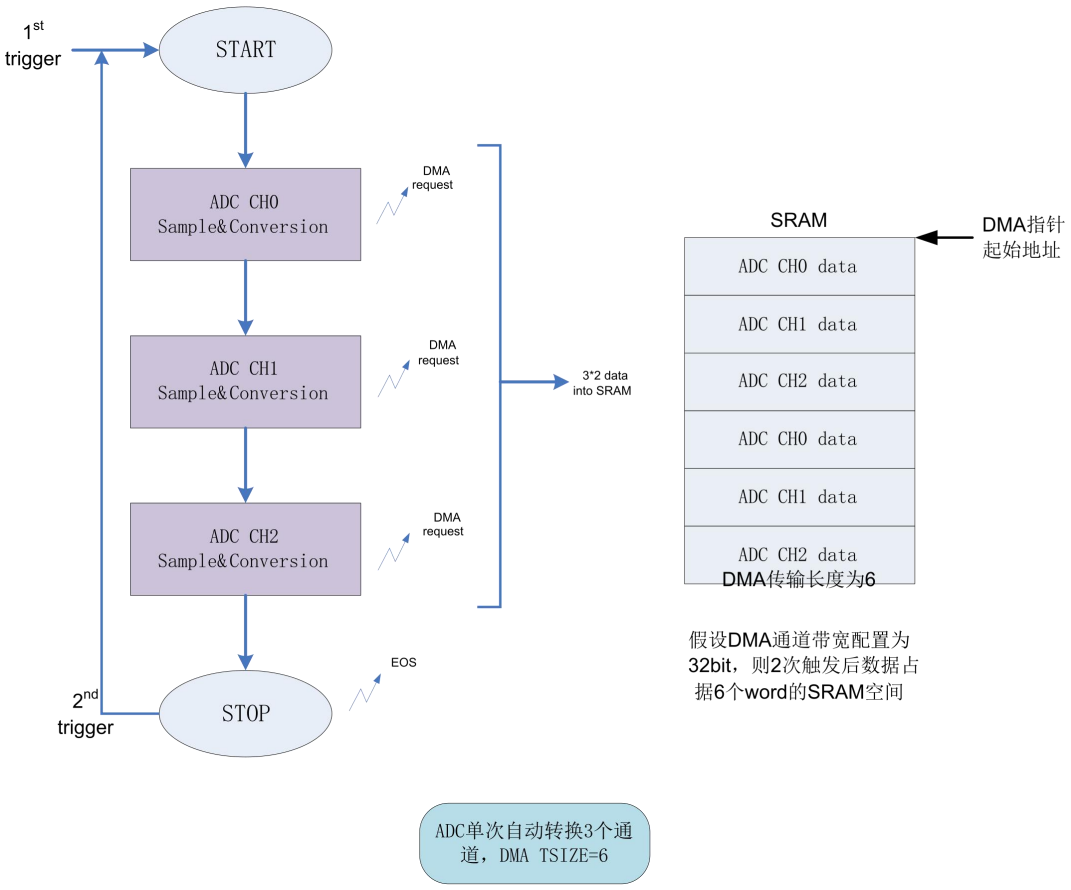


图 25-8 ADC_DMA 案例

由于 DMA 仅负责搬移数据，所以使用的时候需要考虑 ADC 的配置和 DMA 的配置相匹配。

25.6 寄存器

ADC 基地址 (0x4001AC00)

地址	符号	名称
0x0000	ADC_ISR	ADC 中断和状态寄存器
0x0004	ADC_IER	ADC 中断使能寄存器
0x0008	ADC_CR1	ADC 控制寄存器 1
0x000C	ADC_CR2	ADC 控制寄存器 2
0x0010	ADC_CALCR	ADC 校准控制寄存器
0x0014	ADC_CFGR1	ADC 配置寄存器 1
0x0018	ADC_CFGR2	ADC 配置寄存器 2
0x001C	ADC_SMTR	ADC 采样时间控制寄存器
0x0020	ADC_CHER	ADC 通道控制寄存器
0x0028	ADC_DR	ADC 数据寄存器

25.6.1 ADC 中断和状态寄存器(ADC_ISR)

符号	ADC_ISR

符号	ADC_ISR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU				DRCH			
权限	R-0000				R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RDY	RFU		EOCAL	BUSY	OVR	EOSIF	EOCIF
权限	W1C-0	R-00		W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

位	位名	权限	功能描述
31:12	RFU	R	RFU：未实现，读为 0
11:8	DRCH	R	通道标号
7	RDY	W1C	ADC ready 标志, 软件清零 PWD 寄存器后，等待 32 个 ADCCLK 后 RDY 自动置位，软件写 1 清零 当软件关闭 ADC 后 RDY 自动清零 此标志仅用于查询，不会产生中断 注意：在 RDY 没有置位的情况下触发 ADC 转换，ADC 将自动延迟等待直到 RDY=1 再开始转换， 这样可以使得用户在使用中无需关心 RDY 标志
6:5	RFU	R	RFU：未实现，读为 0
4	EOCAL	W1C	dummy
3	BUSY	W1C	ADC 忙标志 (ADC busy) 1：ADC 正在校准、采样或转换过程中 0：ADC 空闲
2	OVR	W1C	数据冲突标志，硬件置位，软件写 1 清零。(Over-Run) 当 ADC_DATA 寄存器中的上一次转换结果还未被读取，新的转换结果又到来时，硬件置位 OVR 标志。 0：没有数据冲突 1：出现数据冲突
1	EOSIF	W1C	转换序列结束（End Of Sequence） 所有使能通道都转换完成后，置位 EOS，软件写 1 清零。
0	EOCIF	W1C	单次转换结束（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。

25.6.2 ADC 中断使能寄存器(ADC_IER)

符号	ADC_IER							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU			EOCALIE	RFU	OVRIE	EOSIE	EOCIE
权限	R-000			RW-0	R-0	RW-0	RW-0	RW-0

位	位名	权限	功能描述
31:5	RFU	R	RFU：未实现，读为 0
4	EOCALIE	RW	dummy
3	RFU	R	RFU：未实现，读为 0
2	OVRIE	RW	数据冲突标志使能
1	EOSIE	RW	转换序列结束使能
0	EOCIE	RW	单次转换结束（End Of Conversion）中断使能寄存器 0：禁止 EOC 中断 1：允许 EOC 中断

25.6.3 ADC 控制寄存器 1(ADC_CR1)

符号	ADC_CR1							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						SWTRIG	ADEN
权限	R-000000						RW-0	RW-0

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1	SWTRIG	RW	ADC 启动转换寄存器（软件触发），软件写 1 启动，硬件自动清零。
0	ADEN	RW	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

25.6.4 ADC 控制寄存器 2(ADC_CR2)

符号	ADC_CR2							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						TRGCFG	
权限	R-000000						W-00	

位	位名	权限	功能描述
31:2	RFU	R	RFU：未实现，读为 0
1:0	TRGCFG	W	触发信号使能和极性选择。(Trigger Config) 当配置为非软件触发时，禁止软件写 START 寄存器。 00：软件触发 01：硬件上升沿触发 10：硬件下降沿触发 11：硬件上升、下降沿都触发

25.6.5 ADC 校准控制寄存器(ADC_CALCR)

符号	ADC_CALCR							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OSCAL_CYCLE							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU			OFFL_	RFU			

符号	ADC_CALCR							
				EN				
权限	R-000			RW-0	R-0000			
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	VCM_CTRL	VCM_MOD	CMPRDY_DLY		CMP_MOD	CKDIG_DLY		CALEN
权限	RW-0	RW-0	RW-00		RW-0	RW-00		RW-0

位	位名	权限	功能描述
31:24	RFU	R	RFU:未实现，读为 0
23:16	OSCAL_CYCLE	RW	dummy
15:13	RFU	R	RFU:未实现，读为 0
12	OFFL_EN	RW	dummy
11:8	RFU	R	RFU:未实现，读为 0
7	VCM_CTRL	RW	dummy
6	VCM_MOD	RW	dummy
5:4	CMPRDY_DLY	RW	dummy
3	CMP_MOD	RW	dummy
2:1	CKDIG_DLY	RW	dummy
0	CALEN	RW	dummy

25.6.6 ADC 配置寄存器 1(ADC_CFGR1)

符号	ADC_CFGR1							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RPT_LEN							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RPT_LEN							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							REFNSEL
权限	R-0000000							RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PCLK_PSC			RFU		CLKSEL	REFPSEL	
权限	RW-000			R-00		RW-0	RW-00	

位	位名	权限	功能描述
31:16	RPT_LEN	RW	连续采样模式下的重复转换次数，rpt_len 为 0 时会一直转换，非 0 时转换(rpt_len+1)次
15:9	RFU	R	RFU：未实现，读为 0
8	REFNSEL	RW	dummy
7:5	PCLK_PSC	RW	APBCLK 用作 ADC 工作时钟前的预分频 (APBCLK prescaler)

位	位名	权限	功能描述
			000：不分频 001：2 分频 010：4 分频 011：8 分频 101：3 分频 110：6 分频 111：5 分频 其他：RFU
4:3	RFU	R	RFU：未实现，读为 0
2	CLKSEL	RW	dummy
1:0	REFPSEL	RW	dummy

25.6.7 ADC 配置寄存器 2(ADC_CFGR2)

符号	ADC_CFGR2							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU	AWDCH					AWDSC	AWDEN
权限	R-0	RW-00000					RW-0	RW-0
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	OVSS				OVSR			OVSEN
权限	RW-0000				RW-000			RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU	IOTRFEN	RFU		SEMI	WAIT_F	CONT_MOD	OVRM
权限	R-0	RW-0	R-00		RW-0	RW-0	RW-0	RW-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTS				RFU	SCAN_DIR	RFU	DMAEN
权限	RW-0000				R-0	RW-0	R-0	RW-0

位	位名	权限	功能描述
31	RFU	R	RFU：未实现，读为 0
30:26	AWDCH	RW	dummy
25	AWDSC	RW	dummy
24	AWDEN	RW	dummy
23:20	OVSS	RW	过采样移位控制寄存器 0000：不移位 0001：右移 1bit 0010：右移 2bit 0011：右移 3bit 0100：右移 4bit 0101：右移 5bit 0110：右移 6bit 0111：右移 7bit

位	位名	权限	功能描述
			1000：右移 8bit Others：RFU
19:17	OVSR	RW	过采样率控制 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x
16	OVSEN	RW	过采样使能， 0：禁止过采样 1：使能过采样
15	RFU	R	RFU：未实现，读为 0
14	IOTRFEN	RW	dummy
13:12	RFU	R	RFU：未实现，读为 0
11	SEMI	RW	单次转换半自动模式，仅在单次转换（CONT=0）时有效，参见“转换模式”章节 0：自动模式 1：半自动模式
10	WAIT_F	RW	等待模式控制 0：无等待，如果上次转换数据没有及时读取，则可能出现 Overrun 1：等待模式，在上次转换数据被读取前，不会启动下一次转换
9	CONT_MOD	RW	连续转换模式使能 0：单次转换 1：连续转换
8	OVRM	RW	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
7:4	EXTS	RW	硬件触发源选择 0000： LUT0_TRGO 0001： LUT1_TRGO 0010： LUT2_TRGO 0011： ATIM_TRGO 0100： TAU01_TRGO 0101： ATIM_TRGO2 0110： BSTIM16_TRGO 0111： LPTIM16_TRGO 1000： COMP1_TRGO 1001： RFU 1010： RFU 1011： LUT3_TRGO

位	位名	权限	功能描述
			1100： TAU00_TRGO 1101： GPTIM_TRGO 1110： RFU 1111： RFU 注意：必须在 TRGCFG=00 的情况下修改 EXTS 寄存器
3	RFU	R	RFU：未实现，读为 0
2	SCAN_DIR	RW	扫描 ACQ 的通道扫描方向 0：正向扫描，如 CH1->CH2… 1：反向扫描，如 CH15->CH14…
1	RFU	R	RFU：未实现，读为 0
0	DMAEN	RW	DMA 使能 0：禁止发出 DMA 请求 1：允许发出 DMA 请求

25.6.8 ADC 采样时间控制寄存器(ADC_SMTR)

符号	ADC_SMTR							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				SMTS			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:8	RFU	R	RFU：未实现，读为 0
7:4	RFU	R	RFU：未实现，读为 0
3:0	SMTS	RW	采样时间控制（*ADC 工作时钟周期)，详见可编程采样时间章节

25.6.9 ADC 通道控制寄存器(ADC_CHER)

符号	ADC_CHER							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU					AVREF	TS	RFU
权限	R-00000					RW-0	RW-0	R-0

符号	ADC_CHER							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							ECH7
权限	R-0000000							RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	ECH1	ECH0	CHEN					
权限	RW-0	RW-0	RW-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	CHEN							
权限	RW-00000000							

位	位名	权限	功能描述
31:27	RFU	R	RFU：未实现，读为 0
26	AVREF	RW	AVREF 通道，写 1 使能
25	TS	RW	温度传感器通道，写 1 使能
24:17	RFU	R	RFU：未实现，读为 0
16	ECH7	RW	ADC_IN7 测量通道，写 1 使能
15	ECH1	RW	ADC_IN1 测量通道，写 1 使能
14	ECH0	RW	ADC_IN0 测量通道，写 1 使能
13:0	CHEN	RW	ADC_IN0~13 测量通道，写 1 使能

25.6.10 ADC 数据寄存器(ADC_DR)

符号	ADC_DR							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DATA							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	DATA							
权限	R-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	DATA	R	ADC 转换结果 在没有使能过采样平均的情况下，结果为低 12bit； 在使能过采样平均的情况下，结果为 12~16bit

26 可编程胶合逻辑（PGL）

26.1 概述

可编程胶合逻辑（Programmable Glue Logic）是基于查找表（LUT）的简单可编程逻辑，其输入输出可以连接到芯片引脚、内部信号，实现一些简单的胶合逻辑，在一些应用中可以帮助系统设计减少 PCB 上的逻辑器件。

每个 LUT 包含 4 个输入，1 个输出，1 个真值表，可选的同步/滤波电路。用户通过对真值表编程，可以获得期望的组合逻辑输出表达式。每个输入信号都可以被单独屏蔽。

PGL 的基本特性如下：

- 实现简单胶合逻辑，简化 PCB 设计
- 4 个 4 输入查找表
- 可以通过真值表编程实现 AND,NAND,OR,NOR,XOR,NOT 等逻辑表达式
- 时序同步或滤波
- 灵活的 LUT 输入选择：IO，内部信号，其他 LUT 输出
- 输出可以被连接到 IO 或其他外设触发

26.2 结构框图

PGL 模块共包含了 4 个 LUT，每个 LUT 的结构如下图所示。

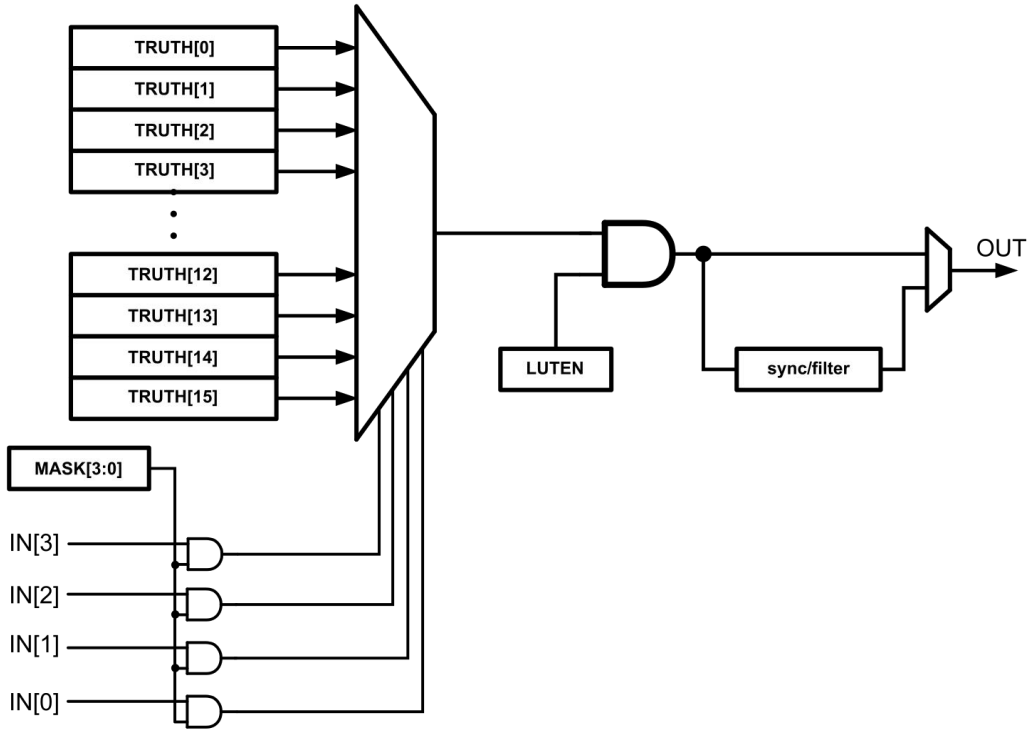


图 26-1 LUT 结构框图

26.3 功能描述

26.3.1 LUT 真值表

通过输入状态组合查询 LUT 真值表中的数据，可以获得想要的组合逻辑功能。

举例来说，希望通过 LUT0 实现一个 2 输入 NAND 逻辑功能，可以采用以下配置：

- 将 LUT0 的输入 2 和输入 3 屏蔽为 0
- LUT0 的输入 0 和输入 1 连接到 GPIO
- LUT0 真值表寄存器低 4bit 写入 0111

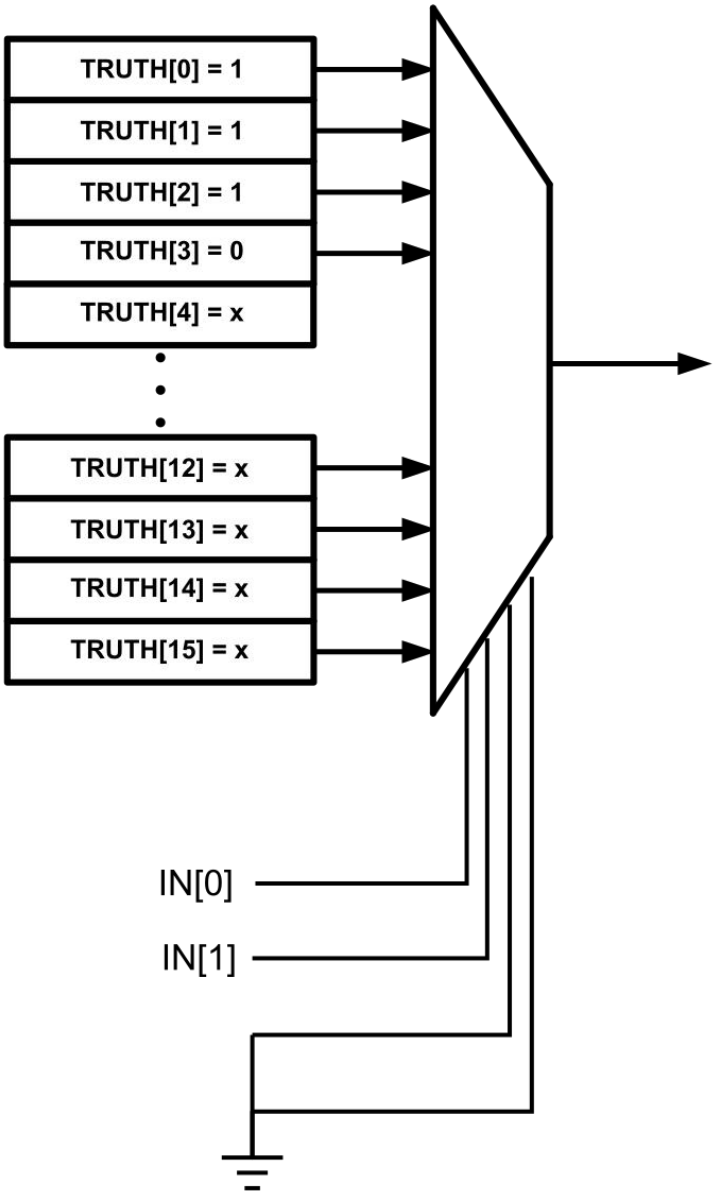


图 26-2 LUT 实现 2 输入 NAND 示意图

26.3.2 LUT 输入

每个 LUT 有 4 个输入，通过 INxSEL 寄存器配置选择输入来自于引脚、芯片外设、或其他 LUT 输出。

当使用引脚输入时，需将对应引脚配置为 GPIO 输入，清除输入的 MASK 寄存器，并使能 LUT。

表 26-1 PGL 相关信号定义

信号名	信号来源	说明
GPIO	PAD	来源于 GPIO 直接输入，具体引脚功能参见引脚和封装。
ATx_OCy	ATIM 模块	来源于 ATIMx 的 PWM 输出通道 y，详情见 ATIM 章节
GTx_OCy	GPTIM 模块	来源于 GTIMx 的 PWM 输出通道 y，详情见 GPTIM 章节
COMPx_OUT	COMP 模块	来源于比较器的数字输出，详见比较器章节
LPTx_OCy	LPTIM 模块	来源于 LPTIMx 模块的 PWM 输出通道 y，详见 LPTIM 章节
TAUx_OUT	TAU 模块	来源于 TAU 模块的 PWM 输出通道 y，详见 TAU 章节
LUTx_OUT	PGL 模块	来源于本模块 LUTx 的输出，输出既可以送往引脚，也可以送入别的 LUT 进一步信号处理

26.3.3 LUT 输出

当使用 LUT 引脚输出时，需将对应引脚配置为数字功能，并通过数字功能选择寄存器将引脚配置为 LUT 输出，然后使能 LUT。

LUT 的输出可以连接到引脚、或其他 LUT 输入，具体引脚功能参见引脚和封装。

26.3.4 滤波和采样

LUT 支持对输出的滤波和同步采样，滤波使能时当 APBCLK 连续采样到 3 个相同电平，则认为合法电平，否则滤波输出不会改变。

根据寄存器配置，应用可以选择 LUT 的输出为组合逻辑输出、APBCLK 同步采样后的输出、或者为采样后经过数字滤波的输出。

经过同步采样后的信号，经过边沿产生电路，可以产生 1 个 APBCLK 周期宽度的上升沿或下降沿脉冲信号。这个同步脉冲信号可用于其它外设的 trigger 信号，或者产生中断。

应用上可以使用 PGL 对外部引脚输入信号进行数字滤波后，再提供给通信外设输入。

比如，使用 LUT0 给 UART0 输入做数字滤波：

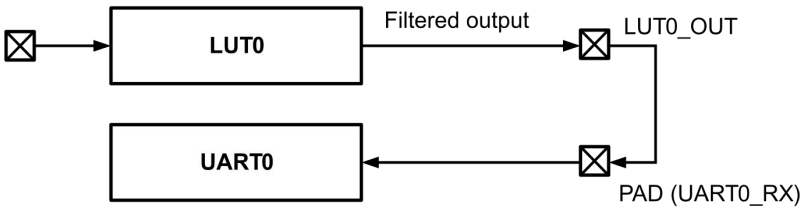


图 26-3 使用 LUT0 为 UART 输入做数字滤波

26.3.5 中断和触发

LUT 输出经过采样和边沿选择后，可以产生中断标志或者输出触发信号。

必须使能 LUT 外设时钟，才能产生中断或触发。

26.3.6 低功耗模式

PGL 可以在低功耗模式下使用。但是要注意滤波和采样功能需要 APB 时钟，因此无法在休眠模式下使用。

如果只是使用 LUT 的组合逻辑输出功能，则应用可以在配置完寄存器之后关闭 PGL 外设模块时钟，以节省功耗。

26.4 寄存器

PGL 基地址 (0x40013800)

地址	符号	名称
0x0000	PGL_CR	PGL 控制寄存器
0x0004	PGL_CFGR0	PGL 配置寄存器 0
0x0008	PGL_CFGR1	PGL 配置寄存器 1
0x000C	PGL_CFGR2	PGL 配置寄存器 2
0x0010	PGL_CFGR3	PGL 配置寄存器 3
0x0014	PGL_IER	PGL 中断使能寄存器
0x0018	PGL_ISR	PGL 中断标志寄存器
0x001C	PGL_LUT0	LUT0 真值表寄存器
0x0020	PGL_LUT1	LUT1 真值表寄存器
0x0024	PGL_LUT2	LUT2 真值表寄存器
0x0028	PGL_LUT3	LUT3 真值表寄存器

26.4.1 PGL 控制寄存器(PGL_CR)

符号	PGL_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							

符号	PGL_CR							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				LUTEN			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3:0	LUTEN	RW	LUT 使能寄存器，分别控制 LUT0~3 1：使能 LUT[x] 0：关闭 LUT[x]

26.4.2 PGL 配置寄存器 0(PGL_CFGR0)

符号	PGL_CFGR0							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				EDGESEL		OUTSEL	
权限	R-0000				RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				MASK			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:20	RFU	R	RFU：未实现，读为 0
19:18	EDGESEL	RW	LUT0 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	RW	LUT0 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	RW	LUT0 输入 3 选择寄存器

位	位名	权限	功能描述
			00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	RW	LUT0 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	RW	LUT0 输入 1 选择寄存器 00：GPIO 01：RFU 10：COMP2_OUT 11：TAU00_OUT
9:8	IN0SEL	RW	LUT0 输入 0 选择寄存器 00：GPIO 01：RFU 10：COMP1_OUT 11：TAU01_OUT
7:4	RFU	R	RFU：未实现，读为 0
3:0	MASK	RW	LUT0 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

26.4.3 PGL 配置寄存器 1(PGL_CFGR1)

符号	PGL_CFGR1							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				EDGESEL		OUTSEL	
权限	R-0000				RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				MASK			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:20	RFU	R	RFU：未实现，读为 0
19:18	EDGESEL	RW	LUT1 有效边沿选择 00：上升沿产生中断和 TRGO

位	位名	权限	功能描述
			01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	RW	LUT1 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	RW	LUT1 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	RW	LUT1 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	RW	LUT1 输入 1 选择寄存器 00：GPIO 01：RFU 10：COMP2_OUT 11：TAU02_OUT
9:8	IN0SEL	RW	LUT1 输入 0 选择寄存器 00：GPIO 01：LUT0_OUT 10：COMP1_OUT 11：TAU03_OUT
7:4	RFU	R	RFU：未实现，读为 0
3:0	MASK	RW	LUT1 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

26.4.4 PGL 配置寄存器 2(PGL_CFGR2)

符号	PGL_CFGR2							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				EDGESEL		OUTSEL	
权限	R-0000				RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
权限	RW-00		RW-00		RW-00		RW-00	

符号	PGL_CFGR2							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				MASK			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:20	RFU	R	RFU：未实现，读为 0
19:18	EDGESEL	RW	LUT2 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	RW	LUT2 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	RW	LUT2 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	RW	LUT2 输入 2 选择寄存器 00：GPIO 01：RFU 10：RFU 11：RFU
11:10	IN1SEL	RW	LUT2 输入 1 选择寄存器 00：GPIO 01：LUT1_OUT 10：COMP2_OUT 11：TAU04_OUT
9:8	IN0SEL	RW	LUT2 输入 0 选择寄存器 00：GPIO 01：LUT0_OUT 10：COMP1_OUT 11：TAU05_OUT
7:4	RFU	R	RFU：未实现，读为 0
3:0	MASK	RW	LUT2 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

26.4.5 PGL 配置寄存器 3(PGL_CFGR3)

符号	PGL_CFGR3
地址	0x0010

符号	PGL_CFGR3							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU				EDGESEL		OUTSEL	
权限	R-0000				RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				MASK			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:20	RFU	R	RFU：未实现，读为 0
19:18	EDGESEL	RW	LUT3 有效边沿选择 00：上升沿产生中断和 TRGO 01：下降沿产生中断和 TRGO 10：上升下降沿都产生中断和 TRGO 11：不产生中断和 TRGO
17:16	OUTSEL	RW	LUT3 输出选择 00：组合逻辑输出 01：同步采样输出 10：数字滤波输出 11：RFU
15:14	IN3SEL	RW	LUT3 输入 3 选择寄存器 00：GPIO 01：LPTIM16.OC1REF 10：BSTIM16.TRGO 11：ATIM.OC1REF
13:12	IN2SEL	RW	LUT3 输入 2 选择寄存器 00：GPIO 01：LUT2_OUT 10：RFU 11：RFU
11:10	IN1SEL	RW	LUT3 输入 1 选择寄存器 00：GPIO 01：LUT1_OUT 10：COMP2_OUT 11：TAU06_OUT
9:8	IN0SEL	RW	LUT3 输入 0 选择寄存器 00：GPIO 01：LUT0_OUT 10：COMP1_OUT 11：TAU07_OUT

位	位名	权限	功能描述
7:4	RFU	R	RFU：未实现，读为 0
3:0	MASK	RW	LUT3 输入屏蔽寄存器 1：屏蔽对应的输入（即固定为 0） 0：不屏蔽输入

26.4.6 PGL 中断使能寄存器(PGL_IER)

符号	PGL_IER							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				LUTIE			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3:0	LUTIE	RW	LUT 中断使能寄存器，分别控制 LUT0~3 1：使能 LUT[x]中断 0：禁止 LUT[x]中断

26.4.7 PGL 中断标志寄存器(PGL_ISR)

符号	PGL_ISR							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU				LUTIF			
权限	R-0000				RW-0000			

位	位名	权限	功能描述
31:4	RFU	R	RFU：未实现，读为 0
3:0	LUTIF	RW	LUT 中断标志寄存器，硬件置位，软件写 1 清零

26.4.8 LUT0 真值表寄存器(PGL_LUT0)

符号	PGL_LUT0							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	TRUTH	RW	LUT0 真值表，软件可以写入任意值以获得需要的逻辑组合

26.4.9 LUT1 真值表寄存器(PGL_LUT1)

符号	PGL_LUT1							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
15:0	TRUTH	RW	LUT1 真值表，软件可以写入任意值以获得需要的逻辑组合

26.4.10 LUT2 真值表寄存器(PGL_LUT2)

符号	PGL_LUT2							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	TRUTH	RW	LUT2 真值表，软件可以写入任意值以获得需要的逻辑组合

26.4.11 LUT3 真值表寄存器(PGL_LUT3)

符号	PGL_LUT3							
地址	0x0028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	TRUTH							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	TRUTH							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0

位	位名	权限	功能描述
15:0	TRUTH	RW	LUT3 真值表，软件可以写入任意值以获得需要的逻辑组合

27 IO 端口（GPIO）

27.1 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

27.2 引脚类型

主要有三种类型的 GPIO 引脚，其中大部分引脚支持输入输出、数字外设功能、模拟外设通道、可控上拉下拉电阻、可控开漏输出功能。

27.2.1 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

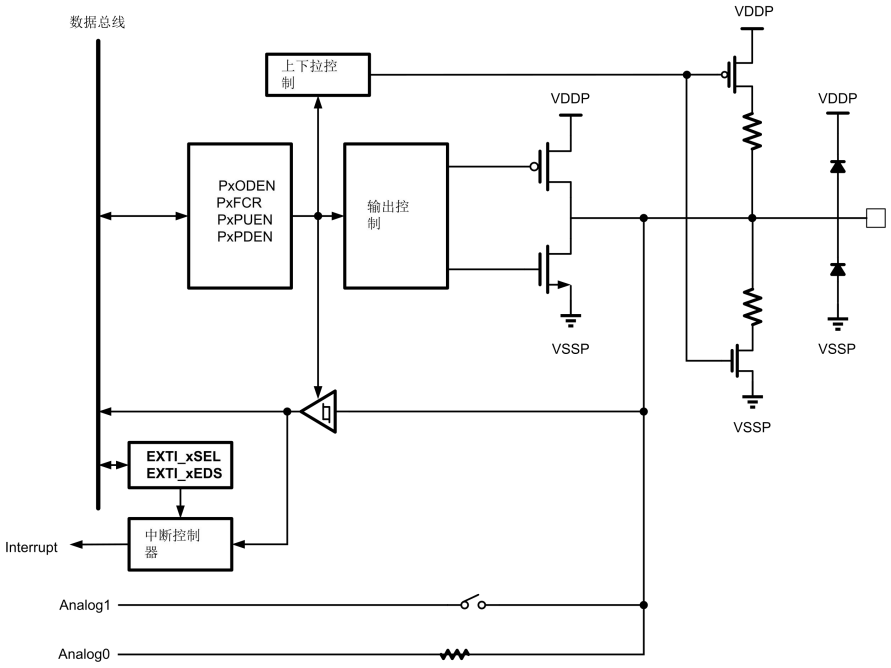


图 27-1 普通 GPIO 结构框图

控制逻辑定义如下：

表 27-1 GPIO 功能逻辑定义表

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
0	0	x	0/1	x	0	0	0/1
	1				1		
1	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

27.3 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：

表 27-2 FCR 定义表

FCR: Function Control Register	PAD function
00	GPIO input
01	GPIO output
10	Digital Function（数字外设功能）
11	Analog

27.3.1 GPIO 输入

当某个 GPIO 被配置成输入功能（FCR=00），并且对应的输入使能寄存器（INEN=1）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

27.3.2 GPIO 输出

当某个 GPIO 被配置成输出功能（FCR=01）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
- 下拉电阻由 PxPDEN 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
- 驱动能力可配置，解决驱动能力和 EMI 之间的平衡

27.3.3 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
- 下拉电阻由 PxPDEN 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。具体引脚功能参见引脚和封装。

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 E->A、15->0 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 PB0 数据被送到外设，PA0 输入信号无效。

27.3.4 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

27.4 特殊引脚

27.4.1 晶体引脚

支持外接 4~16MHz 高频晶体，作为晶体功能时相应管脚应当配置为模拟功能。

如果不使用晶体，软件可以将相应引脚配置为 GPIO 使用。

27.4.2 SWD 引脚

ARM SWD 信号 SWCLK 和 SWIO 复用 GPIO 功能，这两个 PAD 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。

27.4.3 NRST 引脚

NRST 引脚复位后默认为 NRST 功能，作为 NRST 功能时该管脚有 10K 上拉电阻。

如果不使用 NRST 功能，软件可以将相应引脚配置为 GPIO 使用。

NRST 带有数字滤波，滤波采用 RCLP 的 4 分频时钟工作，连续采样 16 拍为低电平后产生全局复位信号，直到变为高电平。

27.4.4 WKUPx 引脚

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 引脚输入上升沿或下降沿（软件配置）能够将芯片从休眠模式下唤醒。每个支持 WKUP 功能的 IO 都带有大约 100ns 的片内模拟滤波，能够滤除输入信号上的毛刺，避免误触发。Sleep/DeepSleep 模式下，使能了的 WKUPx 引脚上任何大于 100ns 的脉冲都会触发芯片唤醒。

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，配置上下拉使能。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。

27.5 外部引脚中断（EXTI）

27.5.1 功能说明

芯片的每组 GPIO（A~E）产生 4 个 EXTI 中断标志，EXTI_xSEL 寄存器用于选择某个 IO 接入 EXTI 通道，最终所有的 EXTI 中断汇总到 CPU 的 NVIC 中断入口，详情见 CPU 章节的中断向量表。

中断标志和引脚对应关系如下表：

GPIO	EXTI 输入选择	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]
PA8~PA11	EXTI_ASEL[5:4]	EXTI[2]
PA12~PA15	EXTI_ASEL[7:6]	EXTI[3]
PB0~PB3	EXTI_BSEL[1:0]	EXTI[4]
PB4~PB7	EXTI_BSEL[3:2]	EXTI[5]
PB8~PB11	EXTI_BSEL[5:4]	EXTI[6]
PB12~PB15	EXTI_BSEL[7:6]	EXTI[7]
PC0~PC3	EXTI_CSEL[1:0]	EXTI[8]
PC4~PC7	EXTI_CSEL[3:2]	EXTI[9]
PC8~PC11	EXTI_CSEL[5:4]	EXTI[10]
PC12~PC13	EXTI_CSEL[7:6]	EXTI[11]
PD0~PD3	EXTI_DSEL[1:0]	EXTI[12]
PD4~PD7	EXTI_DSEL[3:2]	EXTI[13]
PD8~PD11	EXTI_DSEL[5:4]	EXTI[14]
PD12~PD15	EXTI_DSEL[7:6]	EXTI[15]
PE0~PE3	EXTI_ESEL[1:0]	EXTI[16]
PE4~PE7	EXTI_ESEL[3:2]	EXTI[17]
PE8~PE11	EXTI_ESEL[5:4]	EXTI[18]

EXTI 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。

EXTI 模块可以配置是否对输入信号进行数字滤波。数字滤波使能后，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。

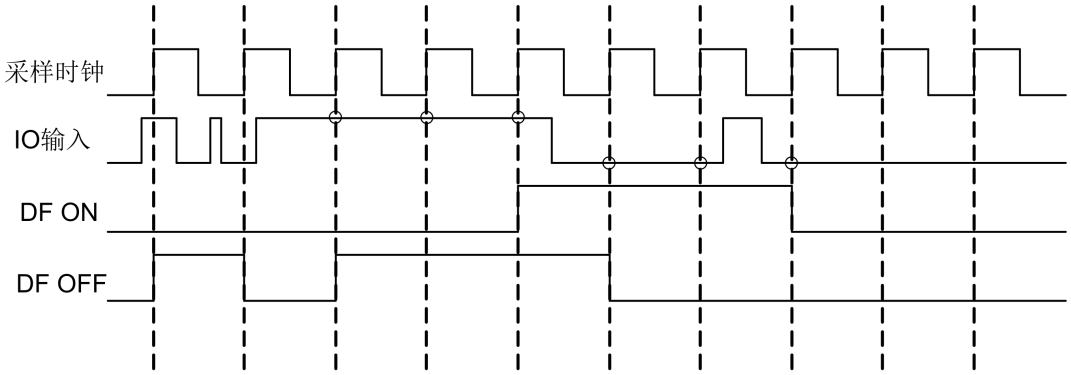


图 27-2 引脚输入数字滤波

软件可以选择数字滤波的采样时钟为 APBCLK 或者 RCLP。

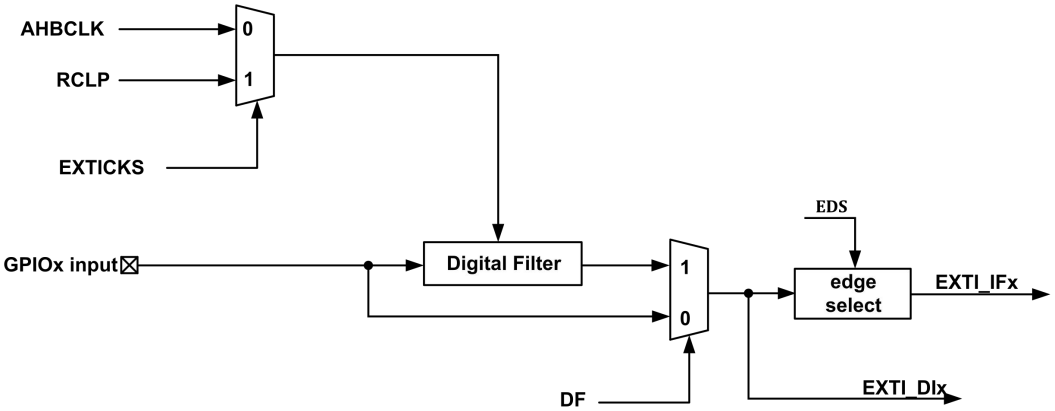


图 27-3 EXTI 信号输入示意图

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。

27.5.2 应用指南

如需在 Sleep/DeepSleep 模式下启动 EXTI 中断唤醒功能，推荐按照如下步骤进行操作：

- 关闭所有 EXTI 使能
- 配置 CMU.SYSCLKSEL 寄存器的 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 RCLP 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 AHBCLK。如果用户使用系统时钟产生 EXTI，推荐流程如下：

- 配置 CMU.PCCR1 寄存器的 EXTICKS 位为 0，选择 AHBCLK 进行 EXTI 采样
- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 配置 CMU.PCCR1 寄存器的 EXTICKE 位为 1，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 RCLP 来产生 EXTI，推荐流程如下：

- 配置 CMU.PCCR1 寄存器的 EXTICKS 位为 1，选择 RCLP 进行 EXTI 采样
- 打开数字滤波使能（如果需要）
- 配置 GPIO 为输入
- 配置 CMU.PCCR1 寄存器的 EXTICKE 位为 1，打开 EXTI 采样时钟使能
- 等待至少 4 个 RCLP 时钟周期
- 配置 EXTI_EDS 触发边沿，使能所需的 EXTI 中断

27.6 快速 GPIO 输出

用户可以通过 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。

27.7 寄存器

GPIOA 基地址 (0x40000C00)

GPIOB 基地址 (0x40000C40)

GPIOC 基地址 (0x40000C80)

GPIOD 基地址 (0x40000CC0)

GPIOE 基地址 (0x40000D00)

地址	符号	名称
0x0000	GPIOx_INEN	GPIOx 输入使能寄存器
0x0004	GPIOx_PUDEN	GPIOx 上下拉使能寄存器
0x0008	GPIOx_ODEN	GPIOx 开漏使能寄存器
0x000C	GPIOx_FCR	GPIOx 功能选择寄存器
0x0010	GPIOx_DO	GPIOx 输出数据寄存器
0x0014	GPIOx_DSET	GPIOx 输出数据置位寄存器
0x0018	GPIOx_DRESET	GPIOx 输出数据复位寄存器
0x001C	GPIOx_DIN	GPIOx 输入数据寄存器
0x0020	GPIOx_DFS	GPIOx 额外数字功能寄存器

地址	符号	名称
0x0024	GPIOx_SRDR	GPIOx 压摆率驱动强度寄存器

GPIOC_TMUX 基地址 (0x40000CB0)

地址	符号	名称
0x0000	GPIOC_TMUX	GPIOC 定时器通道选择寄存器

GPIO_COMMON 基地址 (0x40000DC0)

地址	符号	名称
0x0000	EXTI_SEL0	EXTI 输入选择寄存器 0
0x0004	EXTI_SEL1	EXTI 输入选择寄存器 1
0x0008	EXTI_EDS0	EXTI 边沿选择和使能寄存器 0
0x000C	EXTI_EDS1	EXTI 边沿选择和使能寄存器 1
0x0010	EXTI_DF	EXTI 数字滤波控制寄存器
0x0014	EXTI_IF	EXTI 中断标志寄存器
0x0018	EXTI_DI	EXTI 输入信号寄存器
0x0040	FOUTSEL	FOUT 配置寄存器
0x0140	PINWKEN	WKUP 使能寄存器

27.7.1 GPIOx 输入使能寄存器(GPIOx_INEN)

符号	GPIOx_INEN							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxINEN							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxINEN							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxINEN	RW	GPIO 输入使能开关 0: 关闭输入使能 1: 打开输入使能

27.7.2 GPIOx 上下拉使能寄存器(GPIOx_PUDEN)

符号	GPIOx_PUDEN
地址	0x0004

符号	GPIOx_PUDEN							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxPUD							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxPUD							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxPUDEN							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxPUDEN							
权限	RW-00011000							

位	位名	权限	功能描述
31:16	PxPUD	RW	GPIO 上下拉使能开关 0: 选择上拉 1: 选择下拉
15:0	PxPUDEN	RW	GPIO 上下拉使能开关 0: 关闭上下拉使能 1: 打开上下拉使能

27.7.3 GPIOx 开漏使能寄存器(GPIOx_ODEN)

符号	GPIOx_ODEN							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxODEN							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxODEN							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxODEN	RW	GPIO 开漏使能开关 0: 关闭开漏使能 1: 打开开漏使能

27.7.4 GPIOx 功能选择寄存器(GPIOx_FCR)

符号	GPIOx_FCR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15FCR		Px14FCR		Px13FCR		Px12FCR	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Px11FCR		Px10FCR		Px9FCR		Px8FCR	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Px7FCR		Px6FCR		Px5FCR		Px4FCR	
权限	RW-00		RW-00		RW-00		RW-10	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Px3FCR		Px2FCR		Px1FCR		Px0FCR	
权限	RW-10		RW-00		RW-00		RW-00	

位	位名	权限	功能描述
31:30	Px15FCR	RW	GPIOx_15 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
29:28	Px14FCR	RW	GPIOx_14 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
27:26	Px13FCR	RW	GPIOx_13 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
25:24	Px12FCR	RW	GPIOx_12 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
23:22	Px11FCR	RW	GPIOx_11 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
21:20	Px10FCR	RW	GPIOx_10 引脚功能选择 00：GPIO 输入 01：GPIO 输出

位	位名	权限	功能描述
			10：数字功能 11：模拟功能
19:18	Px9FCR	RW	GPIOx_9 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
17:16	Px8FCR	RW	GPIOx_8 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
15:14	Px7FCR	RW	GPIOx_7 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
13:12	Px6FCR	RW	GPIOx_6 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
11:10	Px5FCR	RW	GPIOx_5 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
9:8	Px4FCR	RW	GPIOx_4 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
7:6	Px3FCR	RW	GPIOx_3 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
5:4	Px2FCR	RW	GPIOx_2 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能
3:2	Px1FCR	RW	GPIOx_1 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能

位	位名	权限	功能描述
			11：模拟功能
1:0	Px0FCR	RW	GPIOx_0 引脚功能选择 00：GPIO 输入 01：GPIO 输出 10：数字功能 11：模拟功能

27.7.5 GPIOx 输出数据寄存器(GPIOx_DO)

符号	GPIOx_DO							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDO							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDO							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxDO	RW	GPIO 输出数据寄存器

27.7.6 GPIOx 输出数据置位寄存器(GPIOx_DSET)

符号	GPIOx_DSET							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxSET							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxSET							
权限	W-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxSET	W	GPIO 输出数据置位寄存器 举例：向 GPIOx_DSET 写 0x0000_8000，则 GPIOx_DO[15]置位，其余位保持不变。

27.7.7 GPIOx 输出数据复位寄存器(GPIOx_DRESET)

符号	GPIOx_DRESET							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxRESET							
权限	W-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxRESET							
权限	W-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxRESET	W	GPIO 输出数据复位寄存器 举例：向 GPIOx_DRESET 写 0x0000_8000，则 GPIOx_DO[15]清零，其余位保持不变

27.7.8 GPIOx 输入数据寄存器(GPIOx_DIN)

符号	GPIOx_DIN							
地址	0x001C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDIN							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDIN							
权限	R-00011000							

位	位名	权限	功能描述
31:16	RFU	R	RFU：未实现，读为 0
15:0	PxDIN	R	GPIOx 输入数据寄存器 软件读此寄存器直接返回引脚输入信号

27.7.9 GPIOx 额外数字功能寄存器(GPIOx_DFS)

符号	GPIOx_DFS							
地址	0x0020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	Px15DFS		Px14DFS		Px13DFS		Px12DFS	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	Px11DFS		Px10DFS		Px9DFS		Px8DFS	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	Px7DFS		Px6DFS		Px5DFS		Px4DFS	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	Px3DFS		Px2DFS		Px1DFS		Px0DFS	
权限	RW-00		RW-00		RW-00		RW-00	

位	位名	权限	功能描述
31:30	Px15DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
29:28	Px14DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
27:26	Px13DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
25:24	Px12DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
23:22	Px11DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
21:20	Px10DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
19:18	Px9DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
17:16	Px8DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
15:14	Px7DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
13:12	Px6DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
11:10	Px5DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
9:8	Px4DFS	RW	数字功能选择位

位	位名	权限	功能描述
			，详见数字外设功能描述章节
7:6	Px3DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
5:4	Px2DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
3:2	Px1DFS	RW	数字功能选择位 ，详见数字外设功能描述章节
1:0	Px0DFS	RW	数字功能选择位 ，详见数字外设功能描述章节

27.7.10 GPIOx 压摆率驱动强度寄存器(GPIOx_SRDR)

符号	GPIOx_SRDR							
地址	0x0024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	PxSR							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	PxSR							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PxDR							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PxDR							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	PxSR	RW	PortX 压摆率控制 0：高压摆率 1：低压摆率
15:0	PxDR	RW	PortX 驱动强度控制 0：高驱动强度 1：低驱动强度

27.7.11 GPIOC 定时器通道选择寄存器(GPIOC_TMUX)

符号	GPIOC_TMUX							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							

符号	GPIOC_TMUX							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PC7TMUX		PC6TMUX		PC5TMUX		PC4TMUX	
权限	RW-00		RW-00		RW-00		RW-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PC3TMUX		PC2TMUX		PC1TMUX		RFU	
权限	RW-00		RW-00		RW-00		R-00	

位	位名	权限	功能描述
31:16	RFU	R	RFU: 未实现，读为 0
15:14	PC7TMUX	RW	PC7 引脚定时器通道选择 00,11：ATIM_CH3N 01：TI06 10：TO06
13:12	PC6TMUX	RW	PC6 引脚定时器通道选择 00,11：ATIM_CH2N 01：TI05 10：TO05
11:10	PC5TMUX	RW	PC5 引脚定时器通道选择 00,11：ATIM_CH1N 01：TI04 10：TO04
9:8	PC4TMUX	RW	PC4 引脚定时器通道选择 00,11：ATIM_CH4 01：TI03 10：TO03
7:6	PC3TMUX	RW	PC3 引脚定时器通道选择 00,11：ATIM_CH3 01：TI02 10：TO02
5:4	PC2TMUX	RW	PC2 引脚定时器通道选择 00,11：ATIM_CH2 01：TI01 10：TO01
3:2	PC1TMUX	RW	PC1 引脚定时器通道选择 00,11：ATIM_CH1 01：TI00 10：TO00
1:0	RFU	R	RFU: 未实现，读为 0

27.7.12 EXTI 输入选择寄存器 0(EXTI_SEL0)

符号	EXTI_SEL0							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI_DSEL							

符号	EXTI_SEL0							
权限	RW-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI_CSEL							
权限	RW-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_BSEL							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_ASEL							
权限	RW-00000000							

位	位名	权限	功能描述
31:24	EXTI_DSEL	RW	PD EXTI 中断输入选择 EXTI_DSEL[7:6] 00：PD12 01：RFU 10：RFU 11：RFU EXTI_DSEL[5:4] 00：PD8 01：PD9 10：PD10 11：PD11 EXTI_DSEL[3:2] 00：PD4 01：PD5 10：PD6 11：PD7 EXTI_DSEL[1:0] 00：PD0 01：PD1 10：PD2 11：PD3
23:16	EXTI_CSEL	RW	PC EXTI 中断输入选择 EXTI_CSEL[7:6] 00：RFU 01：RFU 10：RFU 11：RFU EXTI_CSEL[5:4] 00：PC8 01：PC9 10：PC10 11：PC11 EXTI_CSEL[3:2] 00：PC4

位	位名	权限	功能描述
			01：PC5 10：PC6 11：PC7 EXTI_CSEL[1:0] 00：PC0 01：PC1 10：PC2 11：PC3
15:8	EXTI_BSEL	RW	PB EXTI 中断输入选择 EXTI_BSEL[7:6] 00：PB12 01：PB13 10：PB14 11：PB15 EXTI_BSEL[5:4] 00：PB8 01：PB9 10：PB10 11：PB11 EXTI_BSEL[3:2] 00：PB4 01：PB5 10：PB6 11：PB7 EXTI_BSEL[1:0] 00：PB0 01：PB1 10：PB2 11：PB3
7:0	EXTI_ASEL	RW	PA EXTI 中断输入选择 EXTI_ASEL[7:6] 00：PA12 01：PA13 10：PA14 11：PA15 EXTI_ASEL[5:4] 00：PA8 01：PA9 10：PA10 11：PA11 EXTI_ASEL[3:2] 00：PA4 01：PA5 10：PA6 11：PA7 EXTI_ASEL[1:0] 00：PA0

位	位名	权限	功能描述
			01：PA1 10：PA2 11：PA3

27.7.13 EXTI 输入选择寄存器 1(EXTI_SEL1)

符号	EXTI_SEL1							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_FSEL							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_ESEL							
权限	RW-00000000							

位	位名	权限	功能描述
31:16	RFU	R	RFU: 未实现，读为 0
15:8	EXTI_FSEL	RW	PB EXTI 中断输入选择 EXTI_BSEL[7:6] 00：RFU 01：RFU 10：RFU 11：RFU EXTI_BSEL[5:4] 00：PF8 01：RFU 10：RFU 11：RFU EXTI_BSEL[3:2] 00：PF4 01：PF5 10：PF6 11：PF7 EXTI_BSEL[1:0] 00：PF0 01：PF1 10：PF2 11：PF3
7:0	EXTI_ESEL	RW	PE EXTI 中断输入选择

位	位名	权限	功能描述
			EXTI_ESEL[7:6] 00：- 01：- 10：- 11：- EXTI_ESEL[5:4] 00：- 01：- 10：- 11：- EXTI_ESEL[3:2] 00：PE4 01：PE5 10：- 11：- EXTI_ESEL[1:0] 00：PE0 01：- 10：- 11：PE3

27.7.14 EXTI 边沿选择和使能寄存器 0(EXTI_EDS0)

符号	EXTI_EDS0							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	EXTI15_EDS		EXTI14_EDS		EXTI13_EDS		EXTI12_EDS	
权限	RW-11		RW-11		RW-11		RW-11	
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	EXTI11_EDS		EXTI10_EDS		EXTI9_EDS		EXTI8_EDS	
权限	RW-11		RW-11		RW-11		RW-11	
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI7_EDS		EXTI6_EDS		EXTI5_EDS		EXTI4_EDS	
权限	RW-11		RW-11		RW-11		RW-11	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI3_EDS		EXTI2_EDS		EXTI1_EDS		EXTI0_EDS	
权限	RW-11		RW-11		RW-11		RW-11	

位	位名	权限	功能描述
31:30	EXTI15_EDS	RW	EXTI[15] 边缘触发选择 00: rising 01: falling 10: both 11: disable
29:28	EXTI14_EDS	RW	EXTI[14] 边缘触发选择 00: rising

位	位名	权限	功能描述
			01: falling 10: both 11: disable
27:26	EXTI13_EDS	RW	EXTI[13] 边缘触发选择 00: rising 01: falling 10: both 11: disable
25:24	EXTI12_EDS	RW	EXTI[12] 边缘触发选择 00: rising 01: falling 10: both 11: disable
23:22	EXTI11_EDS	RW	EXTI[11] 边缘触发选择 00: rising 01: falling 10: both 11: disable
21:20	EXTI10_EDS	RW	EXTI[10] 边缘触发选择 00: rising 01: falling 10: both 11: disable
19:18	EXTI9_EDS	RW	EXTI[9] 边缘触发选择 00: rising 01: falling 10: both 11: disable
17:16	EXTI8_EDS	RW	EXTI[8] 边缘触发选择 00: rising 01: falling 10: both 11: disable
15:14	EXTI7_EDS	RW	EXTI[7] 边缘触发选择 00: rising 01: falling 10: both 11: disable
13:12	EXTI6_EDS	RW	EXTI[6] 边缘触发选择 00: rising 01: falling 10: both 11: disable
11:10	EXTI5_EDS	RW	EXTI[5] 边缘触发选择 00: rising 01: falling 10: both 11: disable
9:8	EXTI4_EDS	RW	EXTI[4] 边缘触发选择 00: rising

位	位名	权限	功能描述
			01: falling 10: both 11: disable
7:6	EXTI3_EDS	RW	EXTI[3] 边缘触发选择 00: rising 01: falling 10: both 11: disable
5:4	EXTI2_EDS	RW	EXTI[2] 边缘触发选择 00: rising 01: falling 10: both 11: disable
3:2	EXTI1_EDS	RW	EXTI[1] 边缘触发选择 00: rising 01: falling 10: both 11: disable
1:0	EXTI0_EDS	RW	EXTI[0] 边缘触发选择 00: rising 01: falling 10: both 11: disable

27.7.15 EXTI 边沿选择和使能寄存器 1(EXTI_EDS1)

符号	EXTI_EDS1							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		EXTI18_EDS		EXTI17_EDS		EXTI16_EDS	
权限	R-00		RW-11		RW-11		RW-11	

位	位名	权限	功能描述
31:6	RFU	R	RFU：未实现，读为 0
5:4	EXTI18_EDS	RW	EXTI[18] 边缘触发选择 00: rising 01: falling 10: both

位	位名	权限	功能描述
			11: disable
3:2	EXTI17_EDS	RW	EXTI[17] 边缘触发选择 00: rising 01: falling 10: both 11: disable
1:0	EXTI16_EDS	RW	EXTI[16] 边缘触发选择 00: rising 01: falling 10: both 11: disable

27.7.16 EXTI 数字滤波控制寄存器(EXTI_DF)

符号	EXTI_DF							
地址	0x0010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					EXTI_DF		
权限	R-00000					RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DF							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DF							
权限	RW-00000000							

位	位名	权限	功能描述
31:19	RFU	R	RFU：未实现，读为 0
18:0	EXTI_DF	RW	EXTI 输入数字滤波功能使能 0：关闭 EXTI 数字滤波 1：使能 EXTI 数字滤波

27.7.17 EXTI 中断标志寄存器(EXTI_IF)

符号	EXTI_IF							
地址	0x0014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					EXTI		
权限	R-00000					RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

符号	EXTI_IF							
位名	EXTI							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI							
权限	RW-00000000							

位	位名	权限	功能描述
31:19	RFU	R	RFU：未实现，读为 0
18:0	EXTI	RW	外部引脚中断标志寄存器，共可以产生引脚中断 硬件置位，软件写 1 清零

27.7.18 EXTI 输入信号寄存器(EXTI_DI)

符号	EXTI_DI							
地址	0x0018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU					EXTI_DI		
权限	R-00000					RW-000		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	EXTI_DI							
权限	RW-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	EXTI_DI							
权限	RW-00000000							

位	位名	权限	功能描述
31:19	RFU	R	RFU：未实现，读为 0
18:0	EXTI_DI	RW	EXTI 输入信号只读寄存器，软件读取此寄存器可以观察 EXTI 的输入信号的当前状态 注：当使能了数字滤波后，软件可以从这个寄存器读取到某个 IO 输入信号滤波后的状态。

27.7.19 FOUT 配置寄存器(FOUTSEL)

符号	FOUTSEL							
地址	0x0040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU1							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU1							

符号	FOUTSEL							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU1							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	FOUT1SEL				FOUT0SEL			
权限	RW-0000				RW-0000			

位	位名	权限	功能描述
31:8	RFU1	R	RFU：未实现，读为 0
7:4	FOUT1SEL	RW	FOUT1 输出选择 0000: ADC_ANACLK 0001: RCLP 0010: PCLK 0011: RFU 0100: ADC_EOC 0101: RFU 0110: PLLO/64 0111: ADC_SOC 1000: APBCLK/64 1001: PLLO 1010: RFU 1011: RCHF 1100: XTHF/64 1101: ADCCLK/64 1110: CLK8K 1111: XTHF
3:0	FOUT0SEL	RW	FOUT0 输出选择 0000: ADC_SOC 0001: RCLP 0010: RCHF/64 0011: ADC_ANACLK 0100: AHBCLK/64 0101: RFU 0110: PLLO/64 0111: ADC_EOC 1000: APBCLK/64 1001: PLLO 1010: RFU 1011: RCHF 1100: XTHF/64 1101: RFU 1110: CLK8K 1111: PCLK

27.7.20 WKUP 使能寄存器(PINWKEN)

符号	PINWKEN
地址	0x0140

符号	PINWKEN							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	WKISEL	RFU						
权限	RW-0	R-0000000						
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU		PINWKSEL					
权限	R-00		RW-000000					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PINWKSEL						RFU	
权限	RW-000000						R-00	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	PINWKEN							
权限	RW-00000000							

位	位名	权限	功能描述
31	WKISEL	RW	WKUPx 中断入口选择 0：NMI 中断 1：#39 入口 【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。
30:22	RFU	R	RFU：未实现，读为 0
21:10	PINWKSEL	RW	WKUPx 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、下降沿唤醒 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
9:8	RFU	R	RFU：未实现，读为 0
7:0	PINWKEN	RW	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

28 安全功能（SFU）

28.1 概述

为了对应 IEC60730 和 IEC61508 安全标准，芯片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 ECC
- 闪存内容 CRC 检查
- RAM 数据奇偶校验功能
- RAM 数据写保护功能
- 关键寄存器写保护
- 时钟频率检测功能
- 电源电压检测
- 非法地址访问检测功能
- ADC 测试功能
- 温度检测

28.2 闪存 ECC

代码闪存支持 ECC，可以实现 SEC-DED

28.3 闪存 CRC 检查

使用 DMA 实现对闪存内容的自检。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，用于检查程序完整性。

28.4 RAM 奇偶校验

SRAM 支持硬件奇偶校验，每字节带有 1bit 校验位，以满足 IEC60730 标准要求的 RAM 检验功能，参见总线与存储章节。

注意：SRAM 奇偶校验功能默认关闭，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，可以配置校验错误中断或复位是否使能。

28.5 RAM 写保护

为了满足 IEC61508 的技术要求，RAM 写保护功能用于防止在 CPU 失控时改写 RAM 中的关键数据。RAM 写保护最多覆盖 1KB RAM 空间，以 256B 为颗粒度，被保护的空间不允许写入，只允许读出。

注意：RAM 写保护使能后，CPU、DMA、RAMBIST 都不能改写被保护地址。

28.6 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。

参见存储访问保护控制寄存器（SFU_MAP_CR）

28.7 时钟频率检测

IEC60730 标准强制要求检查系统时钟振荡频率是否正常。

通过多个定时器或者时钟校准模块都可以实现此功能。比如使用 LPTIM16，将其工作时钟配置为 APBCLK，将通道 1 输入捕获目标设置为 RCLP（32Khz），通过捕捉 RCLP 的周期，可以测量系统时钟频率是否正常。

GPTIM、时钟校准电路，也都可以实现类似功能，详情参见相关章节。

28.8 电源电压监控

通过 SVD 和 ADC 都可以实现电源监控功能。

28.9 程序流监控

通过 IWDT 和 WWDT 可以实现程序流监控。

28.10 IO 输出回读

软件可以回读到 IO 输出的信号，实现 IO 输出监控。

28.11 通信数据 CRC 校验

使用芯片的 CRC 模块，可以实现通信数据的 CRC 校验保护。

28.12 非法地址访问检测

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。

不同地址区域具有不同的访问权限控制，非法地址禁止任何形式的访问（禁止读、写、取指），详情参见总线访问权限表。注意，此非法地址访问检测不区分 CPU 或 DMA 发起的访问。

表 28-1 总线访问权限表

属性	地址范围（min）	地址范围（max）	Read	Write	Fetch
Code Flash	0x00000000	0x0001FFFF	O	O	O
Data Flash	0xA0000000	0xA00039FF	O	O	X
SRAM	0x20000000	0x20003FFF	O	O	O

属性	地址范围（min）	地址范围（max）	Read	Write	Fetch
Peripheral AHB	0x40000000	0x40007FFF	O	O	X
Peripheral APB	0x40010000	0x4001FFFF	O	O	X
Others	–	–	X	X	X

28.13 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。

28.14 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。

28.15 寄存器

SFU 基地址 (0x40001C00)

地址	符号	名称
0x0000	SFU_RAMP_CR	RAM 奇偶校验控制寄存器
0x0004	SFU_RAMP_SR	RAM 奇偶校验状态寄存器
0x0008	SFU_MAP_CR	存储访问保护控制寄存器
0x000C	SFU_MAP_SR	存储访问保护状态寄存器

28.15.1 RAM 奇偶校验控制寄存器(SFU_RAMP_CR)

符号	SFU_RAMP_CR							
地址	0x0000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU					RAMP_RE	RAMP_IE	RFU1
权限	R-00000					RW-0	RW-0	R-0

位	位名	权限	功能描述
31:3	RFU	R	未实现：读为 0
2	RAMP_RE	RW	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	RW	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	RFU1	R	未实现：读为 0

28.15.2 RAM 奇偶校验状态寄存器(SFU_RAMP_SR)

符号	SFU_RAMP_SR							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU						RAMP_IF	RFU1
权限	R-000000						RW-0	R-0

位	位名	权限	功能描述
31:2	RFU	R	未实现：读为 0
1	RAMP_IF	RW	RAM 奇偶校验中断标志 (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	RFU1	R	未实现：读为 0

28.15.3 存储访问保护控制寄存器(SFU_MAP_CR)

符号	SFU_MAP_CR							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MAP_SEL	RAMPP	SVDP	CCP	GPIOP	RAMWP		MAP_EN
权限	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00		RW-0

位	位名	权限	功能描述
31:8	RFU	R	未实现：读为 0
7	MAP_SEL	RW	非法地址访问发生时的系统行为配置 (Memory Access Protection Select) 1：非法地址访问发生时，复位芯片 0：非法地址访问发生时，产生 NMI
6	RAMPP	RW	RAM 写保护 (RAM Write Protection) 00：无写保护 01：RAM 起始地址开始的 256 字节禁止写入 10：RAM 起始地址开始的 512 字节禁止写入 11：RAM 起始地址开始的 1K 字节禁止写入
5	SVDP	RW	SVD/BOR 控制寄存器保护 (SVD and BOR Protection) 1：SVD 和 BOR 控制寄存器不可改写 0：无保护
4	CCP	RW	CMU 控制寄存器保护 (Clock Control Protection) 1：CMU 模块的 SYSCLKCR、RCHFCR、RCHFTR、PLLCR、RCLPTR、XTHFCR 寄存器不可改写 0：无保护

位	位名	权限	功能描述
3	GPIOP	RW	GPIO 寄存器保护 1：GPIO 寄存器不可改写
2:1	RAMWP	RW	RAM 写保护 (RAM Write Protection) 00：无写保护 01：RAM 起始地址开始的 256 字节禁止写入 10：RAM 起始地址开始的 512 字节禁止写入 11：RAM 起始地址开始的 1024 字节禁止写入
0	MAP_EN	RW	存储器访问保护使能 (Memory Access Protection Enable) 1：使能非法地址访问检测功能 0：不使能非法地址访问检测 注意：此寄存器一旦置位，软件不能对其清零。

28.15.4 存储访问保护状态寄存器(SFU_MAP_SR)

符号	SFU_MAP_SR							
地址	0x000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU							MAPIF
权限	R-0000000							W1C-0

位	位名	权限	功能描述
31:1	RFU	R	未实现：读为 0
0	MAPIF	W1C	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

29 专用编程接口

29.1 概述

本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。

29.2 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。

30 调试支持

30.1 概述

本芯片基于 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 接口与芯片互联，并实现仿真调试。

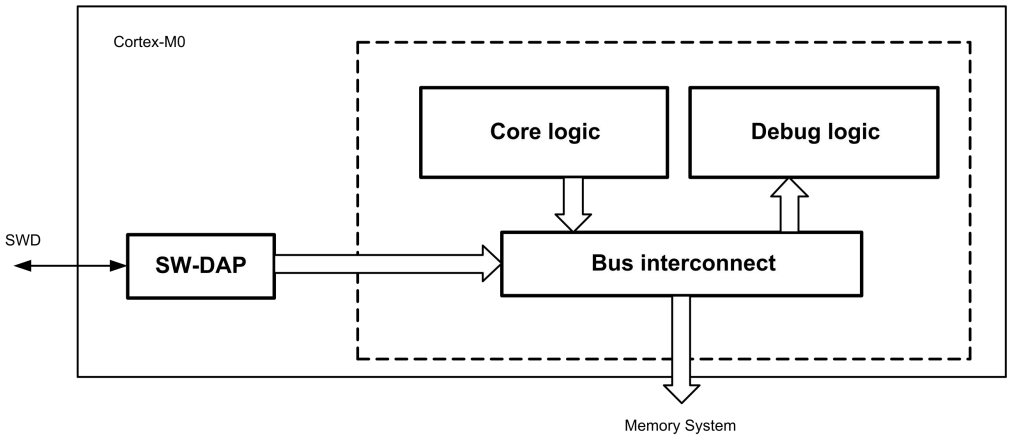


图 30-1 Cortex-M0 调试系统示意图

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。

30.2 Debug 引脚

30.2.1 SWD 引脚

MCU 的 SWD 引脚描述如下表：

SWD pins	Debug 功能
SWDIO	SWD 数据输入/输出
SWCLK	SWD 时钟输入

注意：芯片复位后 SWDIO 和 SWCLK 都默认为输入状态，与大部分 GPIO 不同。

30.2.2 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）。

30.3 SWD 接口协议

30.3.1 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。

30.3.2 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

Bit	Name	描述
32	Parity	32bit 数据的校验位

30.3.3 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。

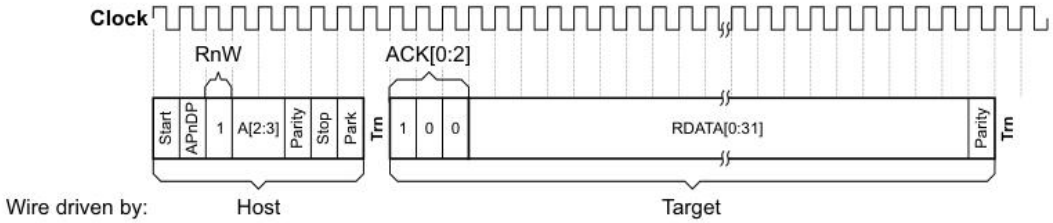
- 芯片复位，或者 SWIO 拉高 50 个 SWCLK 周期后，SW-DP 处于 RESET 状态
- 拉低 SWIO 至少 2 个 SWCLK 周期后，SW-DP 进入 IDLE 状态
- 当 SW-DP 处于 RESET，主机必须先使其进入 IDLE，然后对 ID code 寄存器进行读操作，才能激活 SW-DP。否则从机会对主机的通信回应 FAULT 响应。

30.3.4 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。

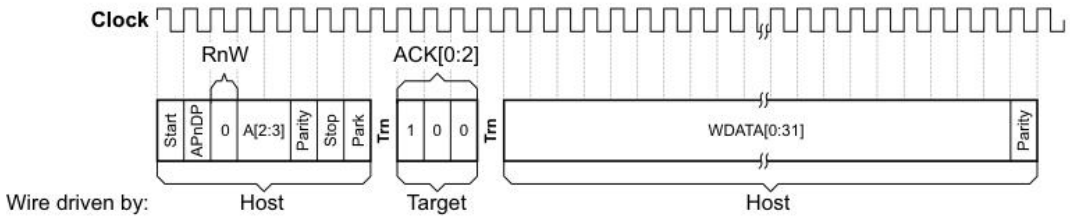


30.3.5 主机写操作

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。



30.4 SWD-DP 寄存器

30.4.1 寄存器列表

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
00	x	ABORT	WO
01	0x0	CTRL/STAT	RW
01	0x1	DLCR	RW
10	x	RESEND	RO
10	x	SELECT	WO
11	x	RDBUFF	RO

关于寄存器的详细说明，请参考 Cortex-M0 Technical Reference Manual.

30.5 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

30.6 低功耗调试支持

通常情况下，当芯片进入 Sleep/DeepSleep 模式时，CPU 的 FCLK 和 HCLK 都会被关闭，这样会导致 host 无法保持与芯片的 debugger 连接。为了支持低功耗模式下的 debug 调试，当 CPU 连接 debugger 时，FCLK 和 HCLK 必须保持运行，即此时芯片不会真正进入休眠，调试中应注意这一点。

30.7 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 DBGCFG 寄存器。

30.8 寄存器

SCU 基地址 (0x40000000)

地址	符号	名称
0x0004	DBGCFG	MCU DEBUG 配置寄存器
0x0008	HDFFLAG	HardFault 查询寄存器

30.8.1 MCU DEBUG 配置寄存器(DBGCFG)

符号	DBGCFG							
地址	0x0004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU						DBG_RCEN	DBG_SLEEP
权限	R-000000						RW-0	RW-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DBG_LPT_STOP	DBG_BT_STOP	RFU					
权限	RW-1	RW-1	R-000000					
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU		DBG_GPTIM_STOP	DBG_ATIM_STOP	DBG_TAU1_STOP	DBG_TAU0_STOP	DBG_WWDT_STOP	DBG_IWDT_STOP
权限	R-00		RW-1	RW-1	RW-1	RW-1	RW-1	RW-1

位	位名	权限	功能描述
31:18	RFU	R	未实现，读为 0
17	DBG_RCEN	RW	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	RW	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_LPT_STOP	RW	Debug 状态下 LPTIM16 使能控制位 1：Debug 时关闭 LPTIM16 0：Debug 时保持 LPTIM16 原来状态
14	DBG_BT_STOP	RW	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
13:6	RFU	R	未实现，读为 0

位	位名	权限	功能描述
5	DBG_GPTIM_STOP	RW	Debug 状态下 GPTIM 使能控制位 1：Debug 时关闭 GPTIM 0：Debug 时保持 GPTIM 原来状态
4	DBG_ATIM_STOP	RW	Debug 状态下 ATIM 使能控制位 1：Debug 时关闭 ATIM 0：Debug 时保持 ATIM 原来状态
3	DBG_TAU1_STOP	RW	Debug 状态下 TAU1 使能控制位 1：Debug 时关闭 TAU1 0：Debug 时保持 TAU1 原来状态
2	DBG_TAU0_STOP	RW	Debug 状态下 TAU0 使能控制位 1：Debug 时关闭 TAU0 0：Debug 时保持 TAU0 原来状态
1	DBG_WWDT_STOP	RW	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	RW	Debug 状态下 IWDT 使能控制位 1：Debug 时关闭 IWDT 0：Debug 时保持 IWDT 开启

30.8.2 HardFault 查询寄存器(HDFFLAG)

符号	HDFFLAG							
地址	0x0008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RFU							
权限	R-00000000							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RFU							
权限	R-00000000							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RFU							
权限	R-00000000							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RFU	dabort_addr_flag	dabort_resp_flag	RFU	bkpt_flag	tbit_flag	RFU	hdf_request_flag
权限	R-0	RW-0	RW-0	R-0	RW-0	RW-0	R-0	RW-0

位	位名	权限	功能描述
31:7	RFU	R	未实现，读为 0
6	dabort_addr_flag	RW	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	RW	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

位	位名	权限	功能描述
			产生错误 0：未访问非法地址
4	RFU	R	未实现，读为 0
3	bkpt_flag	RW	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	RW	Thumb-State 标志，写 1 清零 1：切换到 ARM 状态 0：处于 Thumb-State
1	RFU	R	未实现，读为 0
0	hdf_request_flag	RW	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

31 器件签名信息

每一颗 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。

31.1 唯一器件 ID

唯一器件 ID 保存在 Flash 中，可以通过查询 0x1FFF_FC20 起始地址获取数据。包含 12 个 byte，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	更改说明
v2.0	2025.04	更新为在线手册
v2.1	2025.06	更新 TSSOP24 封装

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