

HC32L12x 系列

32 位 ARM[®] Cortex[®]-M0+ 微控制器

参考手册

Rev1.20
2026 年 8 月

说明事项

版权所有©2026 小华半导体有限公司。保留所有权利

本文件及附件包含的信息有关知识产权权益全部属于小华半导体有限公司（以下简称“XHSC”）；客户对本文件及附件包含的信息只享有内部使用权，未经 XHSC 书面允许，任何单位和个人不得擅自摘抄、复制、改动或以其他任何形式使用本文件的部分或全部内容，并不得以任何形式进行传播。

商标声明

 小华半导体有限公司  和其他商标均为 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。

注意事项

- XHSC 保留随时更改、更正、增强、修改产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- 重要提示：在使用本产品前，请务必查阅官方发布的最新勘误表，充分知悉产品已知修正信息，规避使用风险。
- 本通知中的信息取代并替换先前版本中的信息。

小华半导体有限公司

地址：	上海市浦东新区中科路 1867 号 A 座 4 楼
网址：	https://www.xhsc.com.cn
邮箱：	XHSC_MCU@xhsc.com.cn （商务咨询） XHSC_FAE@xhsc.com.cn （技术咨询）

前言

位/位域

寄存器包含位（一个位宽）或者位域（多个位宽）的字段，位域通常描述成[y:x]，从 x 位开始到 y 位结束。没有特殊说明的情况下，位域字段使用的是十进制数。
不建议使用没有指定的字段，否则可能导致不可预知后果。

寄存器地址

每个寄存器的地址可以通过系统地址划分的基地址以及寄存器的偏移地址（详见各模块的寄存器总表）获得。

寄存器属性

寄存器说明中使用以下缩写词：

缩写词	说明
RW	读写，可以读写该位。
RO	只读，只能读取该位。
WO	只写，只能写入该位。
RWA	读写，可以读写该位。 仅解锁或者安全模式下可写。
RWC	读写，可以读写该位。 写入任意值将该位清零。
RC	只读，只能读取该位。 读取后该位自动清零。
RS	只读，只能读取该位。 读取后该位自动置位。

数据格式

- 0x 前缀表示十六进制数据
- 0b 前缀表示二进制数据
- 数字没有前缀表示十进制数据

Reserved (Res.)

标记为 Reserved 的寄存器和位/位域是为了将来使用预留，必须保持复位值（寄存器另有说明的场景除外）。Reserved 位的读取值在不同设备或者特性中可能不同。

复位值 (Reset)

复位值表示寄存器重置之后的值。寄存器在使用前需要进行初始化重置，否则可能导致不可预知后果。

外设可用性说明

HC32L12x 系列 32 位 ARM® Cortex®-M0+ 微控制器所支持的外设以及其数目，请参见相应的《数据手册》。

安全声明

由于使用某个功能或者协议，可能会存在潜在的安全问题，需要进行声明，提醒用户慎用，规避安全风险。

目录

前言.....	iii
1 产品简介.....	25
2 存储与架构.....	26
2.1 概述.....	26
2.2 系统地址划分.....	27
2.3 存储器和模块地址分配.....	28
3 工作模式.....	29
3.1 概述.....	29
3.2 工作模式切换.....	29
3.3 运行模式 (Active Mode)	30
3.4 休眠模式 (Sleep Mode)	30
3.5 深度休眠模式 (DeepSleep Mode)	30
3.6 功能/模块工作状态.....	31
3.7 寄存器.....	33
3.7.1 系统控制寄存器 (SCB_SCR) (Cortex-M0+内核系统控制寄存器)	33
4 系统控制器 (SYSCTRL)	34
4.1 系统时钟介绍.....	34
4.1.1 时钟架构图.....	34
4.1.2 时钟类型.....	35
4.1.3 时钟启动过程.....	35
4.2 系统时钟切换.....	36
4.2.1 系统时钟切换简介.....	36
4.2.2 切换流程.....	36
4.2.3 切换示例.....	37
4.2.4 时钟切换示意图.....	38
4.3 时钟校准模块.....	38
4.4 中断唤醒控制.....	38
4.4.1 从深度休眠模式唤醒后执行中断服务.....	39
4.4.2 从深度休眠模式唤醒后不执行中断服务.....	39
4.4.3 使用退出休眠特性.....	40
4.5 复位控制.....	40
4.5.1 复位控制器概述.....	40
4.5.2 复位功能.....	41
4.6 寄存器.....	42
4.6.1 寄存器总表.....	42
4.6.2 系统控制寄存器 0 (SYSCTRL0)	43
4.6.3 系统控制寄存器 1 (SYSCTRL1)	45
4.6.4 系统控制寄存器 2 (SYSCTRL2)	46
4.6.5 RCH 控制寄存器 (RCH_CR)	47
4.6.6 XTH 控制寄存器 (XTH_CR)	48
4.6.7 RCL 控制寄存器 (RCL_CR)	50
4.6.8 XTL 控制寄存器 (XTL_CR)	51

4.6.9 复位标识寄存器 (RESET_FLAG)	53
4.6.10 外围模块时钟控制寄存器 0 (PERI_CLKEN0)	55
4.6.11 外围模块时钟控制寄存器 1 (PERI_CLKEN1)	58
4.6.12 外围模块复位控制寄存器 0 (PERI_RESET0)	59
4.6.13 外围模块复位控制寄存器 1 (PERI_RESET1)	61
4.6.14 外设调试功能有效性配置寄存器 (DEBUG_ACTIVE)	62
5 中断控制器 (NVIC)	63
5.1 简介.....	63
5.2 功能说明.....	63
5.2.1 中断优先级.....	63
5.2.2 中断向量表.....	64
5.2.3 中断输入和挂起行为.....	64
5.2.4 中断等待.....	66
5.2.5 中断源.....	66
5.2.6 中断结构图.....	67
5.3 操作示例.....	69
5.3.1 外部中断使能.....	69
5.3.2 NVIC 中断使能和清除使能.....	69
5.3.3 NVIC 中断挂起和清除挂起.....	69
5.3.4 NVIC 中断优先级.....	69
5.3.5 NVIC 中断屏蔽.....	69
5.4 寄存器.....	70
5.4.1 寄存器总表.....	70
5.4.2 中断使能设置寄存器 (NVIC_ISER)	71
5.4.3 中断使能清除寄存器 (NVIC_ICER)	71
5.4.4 中断挂起状态设置寄存器 (NVIC_ISPR)	72
5.4.5 中断挂起状态清除寄存器 (NVIC_ICPR)	72
5.4.6 中断优先级寄存器 (NVIC_IPR0)	73
5.4.7 中断优先级寄存器 (NVIC_IPR1)	73
5.4.8 中断优先级寄存器 (NVIC_IPR2)	74
5.4.9 中断优先级寄存器 (NVIC_IPR3)	74
5.4.10 中断优先级寄存器 (NVIC_IPR4)	75
5.4.11 中断优先级寄存器 (NVIC_IPR5)	75
5.4.12 中断优先级寄存器 (NVIC_IPR6)	76
5.4.13 中断优先级寄存器 (NVIC_IPR7)	76
5.4.14 中断屏蔽特殊寄存器 (PRIMASK)	77
6 Flash 控制器 (FLASH)	78
6.1 简介.....	78
6.2 功能说明.....	78
6.2.1 容量划分.....	78
6.2.2 读等待周期.....	78
6.2.3 Flash 安全保护.....	78
6.3 操作示例.....	79
6.3.1 页擦除 (Sector Erase)	80
6.3.2 全片擦除 (Chip Erase)	80

6.3.3 写操作 (Program)	80
6.3.4 连续写操作 (Cont-Program)	81
6.3.5 读操作 (Read)	83
6.4 寄存器.....	84
6.4.1 寄存器总表.....	84
6.4.2 CR 寄存器 (FLASH_CR)	85
6.4.3 IFR 寄存器 (FLASH_IFR)	86
6.4.4 ICLR 寄存器 (FLASH_ICLR)	86
6.4.5 BYPASS 寄存器 (FLASH_BYPASS)	87
6.4.6 SLOCK0 寄存器 (FLASH_SLOCK0)	87
6.4.7 SLOCK1 寄存器 (FLASH_SLOCK1)	88
6.4.8 保护状态寄存器 (FLASH_LOCK_ST)	89
7 RAM 控制器 (RAM)	90
7.1 简介.....	90
7.2 功能说明.....	90
7.2.1 RAM 地址范围.....	90
7.2.2 读写位宽.....	90
7.2.3 奇偶校验.....	90
7.3 寄存器.....	91
7.3.1 寄存器总表.....	91
7.3.2 控制寄存器 (RAM_CR)	92
7.3.3 奇偶校验出错地址寄存器 (RAM_ERRADDR)	92
7.3.4 出错中断标志寄存器 (RAM_IFR)	93
7.3.5 出错中断标志清除寄存器 (RAM_ICLR)	93
8 端口控制器 (GPIO)	94
8.1 简介.....	94
8.2 主要特性.....	94
8.3 功能说明.....	95
8.3.1 端口配置功能.....	95
8.3.2 端口的写入.....	98
8.3.3 端口的读取.....	98
8.3.4 端口数字复用功能.....	99
8.3.5 端口中断功能.....	101
8.3.6 端口输入滤波功能.....	101
8.3.7 端口复位状态.....	101
8.3.8 HSI2C 通信电平选择功能.....	101
8.4 操作示例.....	101
8.4.1 端口复用配置为模拟端口操作流程.....	101
8.4.2 端口复用配置为数字通用端口操作流程.....	101
8.4.3 端口复用配置为数字功能端口操作流程.....	102
8.4.4 端口复用配置为调试测试端口操作流程.....	102
8.4.5 端口复用配置为红外输出信号操作流程.....	102
8.4.6 端口高电平中断操作流程.....	102
8.4.7 端口低电平中断操作流程.....	102
8.4.8 端口上升沿中断操作流程.....	103

8.4.9 端口下降沿中断操作流程.....	103
8.4.10 端口上拉使能配置操作流程.....	103
8.4.11 端口下拉使能配置操作流程.....	103
8.4.12 端口增强驱动配置操作流程.....	103
8.4.13 端口开漏输出配置操作流程.....	103
8.4.14 端口位置位操作流程.....	103
8.4.15 端口位清零操作流程.....	103
8.4.16 端口位置位清零操作流程.....	104
8.4.17 端口输入滤波操作流程.....	104
8.5 寄存器.....	105
8.5.1 寄存器总表.....	105
8.5.2 端口通用寄存器.....	108
8.5.3 端口辅助寄存器.....	117
8.5.4 端口功能选择寄存器.....	124
9 DMA 控制器 (DMAC)	125
9.1 简介.....	125
9.2 主要特性.....	125
9.3 功能说明.....	126
9.3.1 功能框图.....	126
9.3.2 DMAC 传输模式.....	127
9.3.3 DMA 传输暂停.....	129
9.3.4 DMA 其它配置.....	130
9.3.5 DMA 中断.....	130
9.4 寄存器.....	132
9.4.1 寄存器总表.....	132
9.4.2 所有通道配置寄存器 (DMAC_CONF)	133
9.4.3 通道 x 配置 A 寄存器 (DMAC_CONFAX) (x=0~1)	134
9.4.4 通道 x 配置 B 寄存器 (DMAC_CONF BX) (x=0~1)	136
9.4.5 通道 x 传输源地址寄存器 (DMAC_SRCADR _x) (x=0~1)	138
9.4.6 通道 x 传输目的地址寄存器 (DMAC_DSTADR _x) (x=0~1)	138
10 时钟校准模块 (CTRIM)	139
10.1 简介.....	139
10.2 主要特性.....	139
10.3 功能说明.....	139
10.3.1 RCH 时钟校准模式.....	139
10.3.2 RCL 时钟校准模式.....	140
10.3.3 低功耗定时器模式.....	140
10.4 操作示例.....	141
10.4.1 RCH 时钟自动校准示例.....	141
10.4.2 RCL 时钟自动校准示例.....	141
10.5 寄存器.....	143
10.5.1 寄存器总表.....	143
10.5.2 自动装载寄存器 (CTRIM_ARR)	144
10.5.3 误差计数器寄存器 (CTRIM_CNT)	144
10.5.4 控制寄存器 0 (CTRIM_CR0)	145

10.5.5 控制寄存器 1 (CTRIM_CR1)	146
10.5.6 中断使能寄存器 (CTRIM_IER)	148
10.5.7 中断和状态寄存器 (CTRIM_ISR)	149
10.5.8 中断标志清除寄存器 (CTRIM_ICR)	150
10.5.9 误差结果寄存器 (CTRIM_FCAP)	150
10.5.10 TRIM 结果寄存器 (CTRIM_TVAL)	151
10.5.11 误差上限寄存器 (CTRIM_FLIM)	151
11 系统定时器 (SYSTICK)	152
11.1 简介.....	152
11.2 操作示例.....	152
11.2.1 设置 SysTick.....	152
11.3 寄存器.....	153
11.3.1 寄存器总表.....	153
11.3.2 SysTick 控制和状态寄存器 (CTRL)	154
11.3.3 SysTick 重载寄存器 (LOAD)	154
11.3.4 SysTick 当前值寄存器 (VAL)	155
11.3.5 SysTick 校准值寄存器 (CALIB)	155
12 低功耗定时器 (LPTIM)	156
12.1 简介.....	156
12.2 主要特性.....	156
12.3 功能说明.....	156
12.3.1 功能框图.....	156
12.3.2 工作模式.....	156
12.3.3 计数功能.....	158
12.3.4 定时功能.....	158
12.3.5 PWM 脉冲输出.....	158
12.3.6 GATE 互联.....	158
12.3.7 ETR 互联.....	159
12.4 寄存器.....	160
12.4.1 寄存器总表.....	160
12.4.2 计数器计数值寄存器 (LPTIM_CNT)	161
12.4.3 重载寄存器 (LPTIM_ARR)	161
12.4.4 比较寄存器 (LPTIM_CMP)	162
12.4.5 控制寄存器 (LPTIM_CR)	163
12.4.6 中断标志寄存器 (LPTIM_IFR)	165
12.4.7 中断标志清除寄存器 (LPTIM_ICLR)	165
13 通用定时器 (TIM0/1/2/3)	166
13.1 简介.....	166
13.2 主要特性.....	166
13.2.1 TIM0/1/2 主要特性.....	166
13.2.2 TIM3 主要特性.....	167
13.3 功能说明.....	168
13.3.1 定时器时钟.....	168
13.3.2 定时计数器.....	168
13.3.3 定时器预分频.....	168

13.3.4 模式 0 计数定时器功能.....	168
13.3.5 模式 1 脉宽测量 PWC.....	171
13.3.6 模式 2/3 比较捕获模式.....	174
13.3.7 模式 2/3 从模式.....	190
13.3.8 正交编码计数功能.....	192
13.3.9 Timer 触发 ADC.....	193
13.3.10 逐波限流和刹车控制.....	194
13.3.11 中断.....	195
13.3.12 Timer 互联.....	195
13.3.13 输入互联.....	196
13.3.14 DMA.....	196
13.4 操作示例.....	197
13.4.1 模式 0 计数定时器功能设置示例.....	197
13.4.2 模式 1 脉宽测量 PWC 设置示例.....	197
13.4.3 模式 2/3 比较捕获模式设置示例.....	198
13.4.4 互补 PWM 输出时 B 比较作为 ADC 触发功能.....	199
13.4.5 定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）示例.....	200
13.4.6 定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM1 的 UG 触发 TIM0/2）示例.....	200
13.4.7 DMA 设置示例.....	200
13.5 寄存器.....	202
13.5.1 寄存器总表.....	202
13.5.2 模式 0 计数定时器寄存器.....	204
13.5.3 模式 1 脉宽测量 PWC 寄存器.....	208
13.5.4 模式 2/3 比较捕获模式寄存器.....	216
14 实时时钟（RTC）	239
14.1 简介.....	239
14.2 主要特性.....	239
14.3 功能说明.....	240
14.3.1 功能框图.....	240
14.3.2 上电设定.....	240
14.3.3 时钟误差补偿.....	241
14.3.4 时间戳功能.....	242
14.3.5 唤醒定时器.....	242
14.3.6 RTC 中断.....	243
14.4 操作示例.....	243
14.4.1 RTC 计数开始设置.....	243
14.4.2 读出计数寄存器.....	243
14.4.3 写入计数寄存器.....	244
14.4.4 闹钟设定.....	244
14.4.5 1Hz 输出.....	244
14.5 寄存器.....	246
14.5.1 寄存器总表.....	246
14.5.2 控制寄存器 0（RTC_CR0）	247
14.5.3 控制寄存器 1（RTC_CR1）	249
14.5.4 秒计数寄存器（RTC_SEC）	250

14.5.5 分计数寄存器 (RTC_MIN)	250
14.5.6 时计数寄存器 (RTC_HOUR)	251
14.5.7 日计数寄存器 (RTC_DAY)	252
14.5.8 周计数寄存器 (RTC_WEEK)	253
14.5.9 月计数寄存器 (RTC_MON)	253
14.5.10 年计数寄存器 (RTC_YEAR)	254
14.5.11 时钟误差补偿寄存器 (RTC_COMPEN)	255
14.5.12 高速补偿精度参数寄存器 (RTC_HCOMP)	256
14.5.13 控制寄存器 2 (RTC_CR2)	257
14.5.14 闹钟 A 寄存器 (RTC_ALMA)	258
14.5.15 闹钟 B 寄存器 (RTC_ALMB)	259
14.5.16 时间戳时间寄存器 (RTC_TSTR)	260
14.5.17 时间戳日期寄存器 (RTC_TSDR)	261
14.5.18 唤醒定时器寄存器 (RTC_WUTR)	262
14.5.19 中断使能寄存器 (RTC_IER)	262
14.5.20 中断标志寄存器 (RTC_IFR)	263
14.5.21 中断清除寄存器 (RTC_ICR)	264
14.5.22 保护寄存器 (RTC_WPR)	264
15 独立看门狗 (IWDG)	265
15.1 简介.....	265
15.2 主要特性.....	265
15.3 功能说明.....	265
15.3.1 功能框图.....	265
15.3.2 窗口看门狗模式.....	265
15.3.3 独立看门狗模式.....	265
15.3.4 看门狗溢出处理.....	266
15.3.5 看门狗暂停.....	266
15.3.6 看门狗停止及再次运行.....	266
15.3.7 寄存器写保护.....	266
15.4 操作示例.....	266
15.4.1 配置 IWDG 为窗口看门狗.....	266
15.4.2 配置 IWDG 为独立看门狗.....	266
15.4.3 看门狗超时时间 (10kHz 的独立时钟)	267
15.5 寄存器.....	268
15.5.1 寄存器总表.....	268
15.5.2 键值寄存器 (IWDG_KR)	269
15.5.3 控制寄存器 (IWDG_CR)	270
15.5.4 重载寄存器 (IWDG_ARR)	271
15.5.5 状态寄存器 (IWDG_SR)	272
15.5.6 窗口寄存器 (IWDG_WINR)	273
16 窗口看门狗 (WWDG)	274
16.1 简介.....	274
16.2 主要特性.....	274
16.3 功能说明.....	274
16.3.1 功能框图.....	274

16.3.2 使能看门狗.....	274
16.3.3 配置看门狗超时.....	274
16.3.4 更新递减计数器.....	275
16.3.5 预溢出中断.....	275
16.4 操作示例.....	275
16.4.1 配置看门狗窗口溢出周期示例.....	275
16.5 寄存器.....	276
16.5.1 寄存器总表.....	276
16.5.2 控制寄存器 0 (WWDT_CR0)	277
16.5.3 控制寄存器 1 (WWDT_CR1)	278
16.5.4 状态寄存器 (WWDT_SR)	279
17 低功耗同步异步收发器 (LPUART)	280
17.1 简介.....	280
17.2 主要特性.....	280
17.3 功能说明.....	281
17.3.1 功能框图.....	281
17.3.2 时钟说明.....	281
17.3.3 工作模式.....	281
17.3.4 波特率生成.....	285
17.3.5 帧错误检测.....	288
17.3.6 多机通讯.....	288
17.3.7 硬件流控.....	289
17.3.8 DMA 传输.....	290
17.3.9 收发端缓存.....	290
17.4 操作示例.....	291
17.4.1 发送数据示例.....	291
17.4.2 接收数据示例.....	291
17.5 寄存器.....	292
17.5.1 寄存器总表.....	292
17.5.2 数据寄存器 (LPUART_SBUF)	293
17.5.3 控制寄存器 (LPUART_SCON)	294
17.5.4 地址寄存器 (LPUART_SADDR)	297
17.5.5 地址掩码寄存器 (LPUART_SADEN)	297
17.5.6 中断标志位寄存器 (LPUART_ISR)	298
17.5.7 中断标志位清除寄存器 (LPUART_ICR)	299
17.5.8 波特率寄存器 (LPUART_SCNT)	299
17.5.9 波特率产生寄存器 (LPUART_BSEL)	300
17.5.10 9600/4800 波特率调制器寄存器 (LPUART_MODU)	301
18 通用同步异步收发器 (USART)	303
18.1 简介.....	303
18.2 主要特性.....	303
18.2.1 异步串口特性.....	303
18.2.2 同步串口特性.....	303
18.2.3 智能卡接口特性.....	303
18.2.4 红外通信特性.....	303

18.2.5 LIN 特性.....	303
18.3 功能说明.....	304
18.3.1 USART 通用功能.....	304
18.3.2 异步串口模式.....	311
18.3.3 同步串口模式.....	315
18.3.4 智能卡 (7816) 模式.....	319
18.3.5 红外模式.....	322
18.3.6 LIN 模式.....	323
18.4 操作示例.....	327
18.4.1 异步串口模式发送数据编程示例.....	327
18.4.2 异步串口模式接收数据编程示例.....	327
18.4.3 红外模式 1: 发送数据设定.....	328
18.4.4 红外模式 1: 接收数据设定.....	328
18.5 寄存器.....	329
18.5.1 寄存器总表.....	329
18.5.2 控制寄存器 1 (USART_CR1)	330
18.5.3 控制寄存器 2 (USART_CR2)	333
18.5.4 控制寄存器 3 (USART_CR3)	335
18.5.5 波特率寄存器 (USART_BRR)	337
18.5.6 定时计数器寄存器 (USART_TOR)	337
18.5.7 状态寄存器 (USART_SR)	338
18.5.8 发送数据寄存器 (USART_TDR)	341
18.5.9 中断使能寄存器 (USART_IER)	342
18.5.10 中断标志清除寄存器 (USART_ICR)	344
18.5.11 接收数据寄存器 (USART_RDR)	345
18.5.12 地址寄存器 (USART_ADDR)	345
18.5.13 地址掩码寄存器 (USART_ADDMR)	346
18.5.14 波特率检出值寄存器 (USART_BDR)	346
18.5.15 红外调制寄存器 (USART_IR)	347
19 I2C 总线 (I2C)	348
19.1 简介.....	348
19.2 主要特性.....	348
19.3 协议描述.....	348
19.3.1 I2C 总线上数据传输.....	348
19.3.2 I2C 总线上的应答信号.....	349
19.3.3 I2C 总线上的仲裁.....	350
19.4 功能说明.....	351
19.4.1 功能框图.....	351
19.4.2 串行时钟发生器.....	351
19.4.3 输入滤波器.....	352
19.4.4 地址比较器.....	352
19.4.5 应答标志位.....	352
19.4.6 中断产生器.....	352
19.4.7 工作模式.....	352
19.4.8 状态码表述.....	357

19.5 操作示例.....	359
19.5.1 主机发送示例.....	359
19.5.2 主机接收示例.....	359
19.5.3 从机发送示例.....	360
19.5.4 从机接收示例.....	360
19.6 寄存器.....	361
19.6.1 寄存器总表.....	361
19.6.2 波特率发生器使能寄存器 (I2C_TMRUN)	362
19.6.3 波特率计数器配置寄存器 (I2C_TM)	362
19.6.4 配置寄存器 (I2C_CR)	363
19.6.5 数据寄存器 (I2C_DATA)	364
19.6.6 状态寄存器 (I2C_STAT)	364
19.6.7 从机地址 0 寄存器 (I2C_ADDR0)	365
19.6.8 从机地址 1 寄存器 (I2C_ADDR1)	365
19.6.9 从机地址 2 寄存器 (I2C_ADDR2)	366
19.6.10 从机地址匹配寄存器 (I2C_MATCH)	366
20 高速集成电路总线 (HSI2C)	367
20.1 简介.....	367
20.2 主要特性.....	367
20.3 功能说明.....	368
20.3.1 系统框图.....	368
20.3.2 协议描述.....	368
20.3.3 主机模式.....	369
20.3.4 从机模式.....	373
20.3.5 通信电平选择功能.....	375
20.4 操作示例.....	375
20.4.1 主机发送模式.....	375
20.4.2 主机接收模式.....	375
20.4.3 从机发送模式.....	376
20.4.4 从机接收模式.....	376
20.4.5 主机高速模式切换.....	377
20.5 寄存器.....	378
20.5.1 寄存器总表.....	378
20.5.2 主机控制寄存器 (I2C_MCR)	379
20.5.3 主机状态寄存器 (I2C_MSR)	380
20.5.4 主机中断使能寄存器 (I2C_MIER)	382
20.5.5 主机状态清除寄存器 (I2C_MSCR)	383
20.5.6 主机配置 1 寄存器 (I2C_MCFGR1)	384
20.5.7 主机配置 2 寄存器 (I2C_MCFGR2)	386
20.5.8 主机配置 3 寄存器 (I2C_MCFGR3)	387
20.5.9 主机时钟配置 0 寄存器 (I2C_MCCR0)	388
20.5.10 主机时钟配置 1 寄存器 (I2C_MCCR1)	389
20.5.11 主机发送数据寄存器 (I2C_MTDR)	390
20.5.12 主机接收数据寄存器 (I2C_MRDR)	391
20.5.13 主机数据配置寄存器 (I2C_MDMR)	391

20.5.14 主机 DMA 使能寄存器 (I2C_MDER)	392
20.5.15 从机控制 0 寄存器 (I2C_SCR0)	393
20.5.16 从机状态寄存器 (I2C_SSR)	394
20.5.17 从机中断使能寄存器 (I2C_SIER)	397
20.5.18 从机状态清除寄存器 (I2C_SSCR)	399
20.5.19 从机控制 1 寄存器 (I2C_SCR1)	400
20.5.20 从机控制 2 寄存器 (I2C_SCR2)	402
20.5.21 从机发送 ACK 寄存器 (I2C_STAR)	403
20.5.22 从机发送数据寄存器 (I2C_STDR)	404
20.5.23 从机接收数据寄存器 (I2C_SRDR)	405
20.5.24 从机 DMA 使能寄存器 (I2C_SDER)	406
20.5.25 从机地址寄存器 (I2C_SADDR)	407
20.5.26 从机地址状态寄存器 (I2C_SASR)	408
21 串行外设接口 (SPI)	409
21.1 简介.....	409
21.2 主要特性.....	409
21.3 功能说明.....	409
21.3.1 功能框图.....	409
21.3.2 通信格式和时序.....	410
21.3.3 从机选择管脚配置.....	411
21.3.4 全双工通信.....	411
21.3.5 单线半双工通信.....	412
21.3.6 单工通信.....	412
21.3.7 多机通信.....	413
21.3.8 主机模式延后采样.....	414
21.3.9 状态标志和错误标志.....	414
21.3.10 中断的产生和清除.....	417
21.4 操作示例.....	417
21.4.1 主机全双工收发数据示例.....	417
21.4.2 从机全双工收发数据示例.....	417
21.4.3 单线半双工发送接收数据示例.....	418
21.5 寄存器.....	419
21.5.1 寄存器总表.....	419
21.5.2 SPI 控制寄存器 0 (SPI_CR0)	420
21.5.3 SPI 控制寄存器 1 (SPI_CR1)	422
21.5.4 SPI 单线半双工输出使能寄存器 (SPI_HDOE)	423
21.5.5 SPI 内部从机选择寄存器 (SPI_SSI)	423
21.5.6 SPI 状态寄存器 (SPI_SR)	424
21.5.7 SPI 中断清除寄存器 (SPI_ICR)	426
21.5.8 SPI 数据寄存器 (SPI_DR)	426
22 模数转换器 (ADC)	427
22.1 简介.....	427
22.2 主要特性.....	427
22.3 功能说明.....	427
22.3.1 ADC 框图.....	427

22.3.2 转换时序及转换速度.....	428
22.3.3 单次转换模式.....	428
22.3.4 扫描转换模式.....	428
22.3.5 ADC 转换结果累加.....	430
22.3.6 ADC 转换外部触发源.....	431
22.3.7 ADC 转换结果比较.....	431
22.3.8 ADC 中断.....	432
22.3.9 使用温度传感器测量环境温度.....	432
22.4 操作示例.....	433
22.4.1 通过 ADC_SGLSTART.START 位启动 ADC 单次转换操作流程.....	433
22.4.2 通过外部触发启动 ADC 单次转换操作流程.....	433
22.4.3 通过 ADC_SQRSTART.START 位启动 ADC SQR 扫描转换操作流程.....	434
22.4.4 通过外部触发启动 ADC SQR 扫描转换操作流程.....	434
22.4.5 通过 ADC_JQSTART.START 位启动 ADC JQR 扫描转换操作流程.....	435
22.4.6 通过外部触发启动 ADC JQR 扫描转换操作流程.....	435
22.4.7 通过 ADC_SQRSTART.START 位启动 ADC 连续转换累加操作流程.....	436
22.4.8 通过外部触发启动 ADC 连续转换累加操作流程.....	436
22.4.9 通过 ADC 测量环境温度操作流程.....	437
22.5 寄存器.....	438
22.5.1 寄存器总表.....	438
22.5.2 BGR 控制寄存器 (ADC_BGR)	440
22.5.3 ADC 基本配置寄存器 0 (ADC_CR0)	441
22.5.4 ADC 基本配置寄存器 1 (ADC_CR1)	444
22.5.5 ADC SQR 扫描转换通道配置寄存器 0 (ADC_SQR0)	445
22.5.6 ADC SQR 扫描转换通道配置寄存器 1 (ADC_SQR1)	446
22.5.7 ADC JQR 扫描转换通道配置寄存器 (ADC_JQR)	447
22.5.8 ADC SQR 扫描转换通道 x 转换结果寄存器 (ADC_SQRRESULTx) (x = 0~5)	448
22.5.9 ADC JQR 扫描转换通道 x 转换结果寄存器 (ADC_JQRRESULTx) (x = 0~3)	448
22.5.10 ADC 转换结果寄存器 (ADC_RESULT)	449
22.5.11 ADC 转换结果累加值寄存器 (ADC_RESULTACC)	449
22.5.12 ADC 比较上阈值寄存器 (ADC_HT)	450
22.5.13 ADC 比较下阈值寄存器 (ADC_LT)	450
22.5.14 ADC 中断标志寄存器 (ADC_IFR)	451
22.5.15 ADC 中断清除寄存器 (ADC_ICR)	452
22.5.16 ADC 单次转换或 SQR 扫描转换外部中断触发源配置寄存器 (ADC_EXTTRIGGER0)	453
22.5.17 ADC JQR 扫描转换外部中断触发源配置寄存器 (ADC_EXTTRIGGER1)	454
22.5.18 ADC 单次转换启动控制寄存器 (ADC_SGLSTART)	455
22.5.19 ADC SQR 扫描转换启动控制寄存器 (ADC_SQRSTART)	456
22.5.20 ADC JQR 扫描转换启动控制寄存器 (ADC_JQSTART)	457
22.5.21 ADC 一直转换启动停止控制寄存器 (ADC_ALLSTART)	458
23 液晶控制器 (LCD)	459
23.1 简介.....	459
23.2 主要特性.....	459
23.3 功能说明.....	460
23.3.1 功能框图.....	460

23.3.2 帧频率.....	460
23.3.3 LCD 驱动波形.....	460
23.3.4 死区插入.....	464
23.3.5 LCD Bias 产生电压.....	464
23.3.6 DMA.....	466
23.3.7 中断.....	466
23.3.8 LCD 显示模式.....	467
23.4 操作示例.....	473
23.4.1 内部电阻模式操作流程.....	473
23.4.2 外部电容模式操作流程.....	473
23.4.3 外部电阻模式操作流程.....	473
23.5 寄存器.....	475
23.5.1 寄存器总表.....	475
23.5.2 配置寄存器 0 (LCD_CR0)	476
23.5.3 配置寄存器 1 (LCD_CR1)	478
23.5.4 状态寄存器 (LCD_SR)	480
23.5.5 中断清除寄存器 (LCD_INTCLR)	481
23.5.6 输出配置寄存器 0 (LCD_POEN0)	481
23.5.7 输出配置寄存器 1 (LCD_POEN1)	482
23.5.8 LCD_RAMx (x = 0~7)	484
23.5.9 LCD_RAMx (x = 8~F)	484
24 模拟电压比较器 (VC)	485
24.1 简介.....	485
24.2 主要特性.....	485
24.3 功能说明.....	485
24.3.1 功能框图.....	485
24.3.2 建立/响应时间.....	486
24.3.3 数字滤波.....	486
24.3.4 迟滞功能.....	487
24.3.5 比较结果极性配置.....	488
24.3.6 窗口比较功能.....	488
24.3.7 空白窗口.....	488
24.4 寄存器.....	489
24.4.1 寄存器总表.....	489
24.4.2 控制寄存器 0 (VCx_CR0) (x=0~1)	490
24.4.3 控制寄存器 1 (VCx_CR1) (x=0~1)	492
24.4.4 控制寄存器 2 (VCx_CR2) (x=0~1)	494
24.4.5 状态寄存器 (VCx_SR) (x=0~1)	495
24.4.6 偏移微调寄存器 (VC_TRIM)	496
25 低电压检测器 (LVD)	497
25.1 简介.....	497
25.2 主要特性.....	497
25.3 功能说明.....	497
25.3.1 LVD 框图.....	497
25.3.2 数字滤波.....	497

25.3.3 迟滞功能.....	498
25.4 操作示例.....	499
25.4.1 LVD 配置为低电压复位.....	499
25.4.2 LVD 配置为电压变化中断.....	499
25.5 寄存器.....	500
25.5.1 寄存器总表.....	500
25.5.2 LVD 配置寄存器 (LVD_CR)	501
25.5.3 LVD 中断标志寄存器 (LVD_IFR)	503
26 CRC 运算 (CRC)	504
26.1 简介.....	504
26.2 功能说明.....	504
26.2.1 功能框图.....	504
26.2.2 CRC 参数模型.....	504
26.2.3 CRC16 编码模式.....	505
26.2.4 CRC16 校验模式.....	505
26.2.5 CRC32 编码模式.....	505
26.2.6 CRC32 校验模式.....	505
26.3 寄存器.....	507
26.3.1 寄存器总表.....	507
26.3.2 控制寄存器 (CRC_CR)	508
26.3.3 结果寄存器 (CRC_RESULT)	509
26.3.4 数据寄存器 (CRC_DATA)	510
27 真随机数发生器 (TRNG)	511
27.1 简介.....	511
27.2 功能说明.....	511
27.2.1 真随机数产生原理.....	511
27.3 操作示例.....	511
27.3.1 生成 64bits 真随机数的操作流程 (上电第一次)	511
27.3.2 生成 64bits 真随机数的操作流程 (非上电第一次生成)	512
27.4 寄存器.....	513
27.4.1 寄存器总表.....	513
27.4.2 控制寄存器 (TRNG_CR)	514
27.4.3 模式寄存器 (TRNG_MODE)	515
27.4.4 数据寄存器 0 (TRNG_DATA0)	516
27.4.5 数据寄存器 1 (TRNG_DATA1)	516
28 高级加密标准模块 (AES)	517
28.1 简介.....	517
28.2 主要特性.....	518
28.3 功能说明.....	518
28.3.1 异常机制.....	518
28.3.2 运行时间说明.....	518
28.4 操作示例.....	518
28.4.1 加解密操作共同点.....	518
28.4.2 加密操作流程.....	518
28.4.3 解密操作流程.....	519

28.4.4 数据示例.....	519
28.5 寄存器.....	522
28.5.1 寄存器总表.....	522
28.5.2 控制寄存器 (AES_CR)	523
28.5.3 数据寄存器 (AES_DATAn) (n=0~3)	524
28.5.4 密钥寄存器 (AES_KEYn) (n=0~7)	525
28.5.5 控制寄存器映射地址 (AES_CRDM)	526
29 SWD 调试接口.....	527
29.1 简介.....	527
29.2 功能说明.....	527
29.2.1 调试系统框图.....	527
29.2.2 ARM® 参考文档.....	528
29.2.3 调试端口引脚.....	528
29.2.4 SWD 端口.....	529
29.2.5 内核调试.....	532
29.2.6 BPU (断点单元)	532
29.2.7 DWT (数据观察点)	532
29.2.8 MCU 调试组件 (DBG)	533
30 器件电子签名.....	534
30.1 简介.....	534
30.2 功能说明.....	534
30.2.1 产品唯一身份标识 (UID)	534
30.3 寄存器.....	535
30.3.1 产品唯一身份标识 (UID) 寄存器 (80 位)	535
30.3.2 产品型号寄存器.....	535
30.3.3 FLASH 容量寄存器.....	536
30.3.4 RAM 容量寄存器.....	536
30.3.5 管脚数量/Flash 扇区 Size 配置寄存器.....	537
A 术语表.....	538
修订记录.....	539

图目录

图 3-1 工作模式状态转换图.....	29
图 4-1 时钟架构图.....	34
图 4-2 晶振时钟启动示意图.....	36
图 4-3 时钟切换示意图.....	38
图 4-4 复位来源示意图.....	41
图 5-1 中断向量表.....	64
图 5-2 中断激活和挂起状态.....	65
图 5-3 中断挂起状态被清除然后被重新确认.....	65
图 5-4 中断退出时若中断请求保持高电平就会引起中断处理的再次执行.....	65
图 5-5 中断处理中产生的中断挂起也可以被确认.....	66
图 5-6 中断结构图.....	68
图 8-1 端口电路图.....	95
图 8-2 AHB/FASTIO 总线端口随系统时钟的变化.....	98
图 8-3 读取端口引脚数据同步图.....	98
图 8-4 GPIO 端口输入滤波功能时序图.....	101
图 9-1 DMA 控制器功能框图.....	126
图 9-2 DMA 软件 Block 传输模式.....	127
图 9-3 DMA 软件 Burst 传输模式.....	128
图 9-4 DMA 硬件 Block 传输模式.....	128
图 9-5 DMA 硬件 Burst 传输模式.....	129
图 10-1 RCH 时钟校准时工作框图.....	139
图 10-2 RCL 时钟校准时工作框图.....	140
图 10-3 通用低功耗定时器场景工作框图.....	141
图 12-1 LPTimer 框图.....	156
图 12-2 LPTimer 模式 1.....	157
图 12-3 LPTimer 模式 2.....	157
图 12-4 PWM 脉冲输出示意图.....	158
图 13-1 TIM0/1/2 框图.....	166
图 13-2 TIM3 框图.....	167
图 13-3 模式 0 计数定时器自由计数框图.....	169
图 13-4 模式 0 计数定时器重载计数框图.....	169
图 13-5 模式 0 计数定时器 16 位重载计数波形.....	170
图 13-6 模式 0 计数定时器 32 位自由计数波形.....	170
图 13-7 模式 0 计数定时器自由计数时序图.....	171
图 13-8 模式 0 计数定时器重载计数时序图（预分频设置为 2）.....	171
图 13-9 模式 1 脉宽测量 PWC 功能框图.....	172
图 13-10 模式 1 脉宽测量 PWC 高电平脉冲宽度测量.....	173
图 13-11 模式 1 脉宽测量 PWC 下降沿到下降沿周期测量.....	173
图 13-12 模式 1 脉宽测量 PWC 上升沿到上升沿周期测量.....	173
图 13-13 模式 1 脉宽测量 PWC 上升沿到上升沿周期测量单次模式.....	174
图 13-14 模式 2/3 比较捕获模式计数器框图.....	174
图 13-15 模式 2 比较捕获模式无预分频的向上计数.....	175
图 13-16 模式 2 比较捕获模式带预分频的向上计数.....	175
图 13-17 模式 2 比较捕获模式无预分频的向下计数.....	175
图 13-18 模式 2 比较捕获模式带预分频的向下计数.....	176
图 13-19 模式 2 边沿对齐计数器波形(M23CR.DIR=0).....	176
图 13-20 模式 2 边沿对齐计数器波形(M23CR.DIR=1).....	176
图 13-21 模式 3 比较捕获模式带预分频的上下计数.....	177
图 13-22 模式 3 中心对齐计数器波形.....	177
图 13-23 重复计数器产生更新时序（RCR.OV=0，RCR.UND=0）.....	178

图 13-24 上溢屏蔽时候重复计数器产生更新时序 (RCR.OV=1, RCR.UND=0)	178
图 13-25 下溢屏蔽时候重复计数器产生更新时序 (RCR.OV=0, RCR.UND=1)	178
图 13-26 模式 3 三角波模式缓存使能	179
图 13-27 模式 3 三角波模式缓存无效	179
图 13-28 模式 2 锯齿波模式上计数缓存使能	179
图 13-29 模式 2 锯齿波模式上计数缓存无效	180
图 13-30 模式 2 锯齿波模式下计数缓存使能	180
图 13-31 模式 2 锯齿波模式下计数缓存无效	180
图 13-32 模式 2 锯齿波模式下计数比较缓存使能	181
图 13-33 OCREF 输出框图	181
图 13-34 锯齿波计数单点比较 OCREF 输出波形 (OCMyx=0b111)	182
图 13-35 三角波计数单点比较 OCREF 输出波形 (OCMyx=0b111)	182
图 13-36 锯齿波计数双点比较 OCREF 输出 (OCMyx=0b111)	183
图 13-37 三角波计数双点比较 OCREF 输出 (OCMyx=0b111)	183
图 13-38 独立 PWM 输出框图	184
图 13-39 CCPy=0 时 PWM 输出波形	184
图 13-40 CCPy=1 时 PWM 输出波形	184
图 13-41 互补 PWM 输出框图	185
图 13-42 互补 PWM 输出波形图	185
图 13-43 互补 PWM 输出波形图 (有死区)	186
图 13-44 死区时间	186
图 13-45 三角波模式单脉冲计数	187
图 13-46 锯齿波上计数单脉冲模式	187
图 13-47 锯齿波下计数单脉冲模式	187
图 13-48 比较中断示意图	188
图 13-49 捕获功能框图	189
图 13-50 捕获时序图	189
图 13-51 TIM3 CH0A 捕获输入端口选择	189
图 13-52 CH0B 捕获输入端口选择	190
图 13-53 模式 2/3 从模式示意图	190
图 13-54 模式 2/3 从模式门控功能	191
图 13-55 模式 2/3 从模式触发和复位功能	192
图 13-57 正交编码计数模式 3 计数示意图	193
图 13-58 Timer 触发 ADC	194
图 13-59 DMA 请求触发源	196
图 14-1 RTC 框图	240
图 15-1 IWDG 功能框图	265
图 16-1 WWDG 功能框图	274
图 17-1 LPUART 功能框图	281
图 17-2 Mode0 发送数据	283
图 17-3 Mode0 接收数据	283
图 17-4 Mode1 发送数据	283
图 17-5 Mode1 接收数据	284
图 17-6 Mode2/Mode3 发送数据	284
图 17-7 Mode2/Mode3 接收数据	284
图 17-8 LPUART 硬件流控	289
图 17-9 RTS 硬件流控信号	289
图 17-10 CTS 硬件流控信号	289
图 17-11 接收缓存	290
图 17-12 发送缓存	290
图 18-1 异步串口模式硬件流控示例	308
图 18-2 CTS 硬件流控信号	309

图 18-3	RTS 硬件流控信号.....	309
图 18-4	异步串口功能框图.....	312
图 18-5	异步串口模式数据格式.....	312
图 18-6	发送数据示例.....	313
图 18-7	接收数据示例.....	313
图 18-8	多处理器通信数据格式.....	314
图 18-9	同步串口模式数据格式.....	315
图 18-10	同步串口模式发送数据图例 1 (LSB/无 CTS 功能) (使用内部时钟源)	316
图 18-11	同步串口模式发送数据图例 1 (LSB/无 CTS 功能) (使用外部时钟源)	316
图 18-12	同步串口模式发送数据图例 2 (LSB/CTS 功能)	317
图 18-13	同步串口模式接收数据图例 1 (LSB/无 RTS 功能)	318
图 18-14	同步串口模式接收数据图例 2 (LSB/RTS 功能)	318
图 18-15	同步串口模式注意事项.....	319
图 18-16	智能卡连接示意图.....	319
图 18-17	智能卡模式内部同步时序和采样时序图.....	320
图 18-18	智能卡模式数据格式.....	320
图 18-19	智能卡模式数据格式示例.....	320
图 18-20	智能卡模式发送数据图例.....	321
图 18-21	智能卡模式接收数据图例.....	322
图 18-22	红外数据调制.....	322
图 18-23	38K 红外调制.....	323
图 18-24	LIN 总线数据和帧结构.....	324
图 19-1	I2C 传输协议.....	348
图 19-2	START、repeated START 和 STOP.....	349
图 19-3	I2C 总线上位传输.....	349
图 19-4	I2C 总线上应答信号.....	349
图 19-5	I2C 总线上的仲裁.....	350
图 19-6	I2C 功能模块图.....	351
图 19-7	主发送模式数据同步图.....	353
图 19-8	I2C 主机发送状态图.....	353
图 19-9	主机接收模式数据同步图.....	354
图 19-10	I2C 主机接收状态图.....	354
图 19-11	从接收模式数据同步图.....	355
图 19-12	从机接收状态图.....	355
图 19-13	从机发送模式数据同步图.....	356
图 19-14	I2C 从机发送状态图.....	356
图 19-15	I2C 广播呼叫状态图.....	357
图 20-1	I2C 系统框图.....	368
图 20-2	I2C 总线的时序图.....	369
图 20-3	I2C 总线时序参数图.....	370
图 21-1	SPI 框图.....	409
图 21-2	SPI 通信格式时序图.....	410
图 21-3	从机选择 NSS 管脚配置.....	411
图 21-4	SPI 全双工通信.....	412
图 21-5	SPI 单线半双工通信.....	412
图 21-6	SPI 单工发送通信示意.....	413
图 21-7	SPI 单工接收通信示意.....	413
图 21-8	SPI 多机系统.....	414
图 21-9	SPI 主机模式一般标志位说明图.....	416
图 21-10	SPI 从机模式一般标志位说明图.....	416
图 22-1	ADC 示意框图.....	427
图 22-2	ADC 转换时序图.....	428

图 22-3 ADC SQR 扫描转换过程示例.....	429
图 22-4 ADC JQR 扫描转换过程示例.....	429
图 22-5 ADC SQR 扫描过程中进行 JQR 扫描示例.....	430
图 22-6 ADC 连续转换累加过程示例.....	431
图 22-7 ADC 转换结果比较示意图.....	431
图 23-1 LCD 功能框图.....	460
图 24-1 VC 功能框图.....	485
图 24-2 VC 滤波响应时间.....	487
图 24-3 VC 迟滞功能.....	487
图 24-4 VC 空白窗口.....	488
图 25-1 LVD 框图.....	497
图 25-2 LVD 滤波输出.....	498
图 25-3 LVD 迟滞响应.....	498
图 26-1 CRC 应用示意图.....	504
图 27-1 真随机数发生器功能框图.....	511
图 28-1 AES 的加解密示意图.....	517
图 28-2 AES 的加密流程图.....	517
图 29-1 调试支持框图.....	527

表目录

表 2-1 地址划分表.....	28
表 3-1 不同工作模式下系统可响应的复位类型.....	29
表 3-2 各模式下功能/模块工作状态表.....	31
表 4-2 系统控制器 (SYSCTRL) 寄存器偏移地址.....	42
表 5-1 Cortex-M0+处理器中断一览表.....	63
表 5-4 中断控制器 (NVIC) 寄存器偏移地址.....	70
表 6-1 Flash 容量划分表.....	78
表 6-3 Flash 控制器 (FLASH) 寄存器偏移地址.....	84
表 7-1 RAM 地址映射.....	90
表 7-2 RAM 控制器 (RAM) 寄存器偏移地址.....	91
表 8-1 端口状态真值表.....	97
表 8-2 端口复用表.....	99
表 8-3 端口控制器 (GPIO) 寄存器偏移地址.....	105
表 9-1 DMAC 传输模式比对表.....	127
表 9-2 DMA 控制器 (DMAC) 寄存器偏移地址.....	132
表 10-1 时钟校准模块 (CTRIM) 寄存器偏移地址.....	143
表 11-2 SysTick 定时器寄存器偏移地址.....	153
表 12-2 低功耗定时器 (LPTIM) 寄存器偏移地址.....	160
表 13-8 CH0A 捕获输入信号.....	189
表 13-13 TIM0/1/2/3 中断列表.....	195
表 13-14 定时器互联信号列表.....	195
表 13-15 模式 0 定时器寄存器偏移地址.....	202
表 13-16 模式 1 脉宽测量 PWC 寄存器偏移地址.....	202
表 13-17 模式 2/3 比较捕获模式寄存器偏移地址.....	202
表 14-1 RTC 的基本特性.....	239
表 14-2 实时时钟 (RTC) 寄存器偏移地址.....	246
表 15-1 IWDG 超时时间 (10kHz 的独立时钟)	267
表 15-2 独立看门狗 (IWDG) 寄存器偏移地址.....	268
表 16-1 窗口看门狗 (WWDT) 寄存器偏移地址.....	276
表 17-1 LPUART Mode0/1/2/3 数据结构.....	282
表 17-2 B8 数据含义.....	282
表 17-3 SCLK=4MHz 波特率计算表.....	286
表 17-4 SCLK=8MHz 波特率计算表.....	286
表 17-5 SCLK=16MHz 波特率计算表.....	286
表 17-6 SCLK=24MHz 波特率计算表.....	287
表 17-7 SCLK=32MHz 波特率计算表.....	287
表 17-8 SCLK=48MHz 波特率计算表.....	288
表 17-9 低功耗同步异步收发器 (LPUART) 寄存器偏移地址.....	292
表 18-1 各模式采样计数和最高波特率.....	304
表 18-2 PCLK 为 4MHz 的波特率编程示例.....	305
表 18-3 PCLK 为 8MHz 的波特率编程示例.....	305
表 18-4 PCLK 为 16MHz 的波特率编程示例.....	305
表 18-5 PCLK 为 24MHz 的波特率编程示例.....	306
表 18-6 PCLK 为 32MHz 的波特率编程示例.....	306
表 18-7 PCLK 为 48MHz 的波特率编程示例.....	307
表 18-8 多功能定时计数器功能.....	310
表 18-9 各模式中断有效性.....	311
表 18-10 通用同步异步收发器 (USART) 寄存器偏移地址.....	329
表 19-1 I2C 时钟信号波特率.....	352
表 19-2 I2C 状态码表述.....	357

表 19-3 I2C 总线 (I2C) 寄存器偏移地址.....	361
表 20-1 主机时序参数.....	370
表 20-2 同步延迟.....	371
表 20-3 I2C 时序参数限制.....	371
表 20-4 I2C 示例时序配置.....	371
表 20-7 高速集成电路总线 (HSI2C) 寄存器偏移地址.....	378
表 21-1 SPI 中断.....	417
表 21-2 串行外设接口 (SPI) 寄存器偏移地址.....	419
表 22-1 插队行为状态表.....	430
表 22-2 ADC 中断.....	432
表 22-3 温度传感器校准值信息表.....	432
表 22-4 模数转换器 (ADC) 寄存器偏移地址.....	438
表 23-12 液晶控制器 (LCD) 寄存器偏移地址.....	475
表 23-15 COMy/SEGx PAD 在不同 duty 下功能配置.....	477
表 24-1 模拟电压比较器 (VC) 寄存器偏移地址.....	489
表 25-1 低电压检测器 (LVD) 寄存器偏移地址.....	500
表 26-1 CRC 参数模型.....	504
表 26-2 CRC 运算 (CRC) 寄存器偏移地址.....	507
表 27-1 真随机数发生器 (TRNG) 寄存器偏移地址.....	513
表 28-1 AES 加解密运行时间.....	518
表 28-2 128 位操作寄存器示例.....	519
表 28-3 192 位操作寄存器示例.....	520
表 28-4 256 位操作寄存器示例.....	521
表 28-5 高级加密标准模块 (AES) 寄存器偏移地址.....	522
表 28-11 寄存器填入数据.....	524
表 28-14 寄存器填入数据-128 位密钥.....	525
表 28-15 寄存器填入数据-192 位密钥.....	525
表 28-16 寄存器填入数据-256 位密钥.....	525
表 29-3 主机发送的数据包请求.....	529
表 29-4 目标发送的确认响应.....	529
表 29-5 主机或目标发送的数据传输.....	530

1 产品简介

HC32L12x 系列是一款旨在延长便携式测量系统的电池使用寿命的超低功耗、宽电压工作范围的 MCU。集成 12-bit 2Msps 高精度 SARADC，比较器、内置 PWM 定时器、LCD 显示、USART、SPI、I2C 等丰富的通讯外设，内建 AES、TRNG 等信息安全模块，具有高整合度、高抗干扰、高可靠性和超低功耗的特点。本产品内核采用 Cortex-M0+内核，主频最大到 48MHz，可配合成熟的 Keil、IAR、Eclipse 等主流开发调试软件，支持使用 C 语言及汇编语言开发。

超低功耗 MCU 典型应用

主要用于以下场景：

- 传感器应用、物联网应用
- 智能仪表、无线模块、温控器、货架标签
- 智能交通、报警系统
- 智能家居、医疗设备

关于本手册

本手册主要介绍芯片的功能、操作事项和使用方法。关于芯片的规格，请参阅对应的《数据手册》。

2 存储与架构

2.1 概述

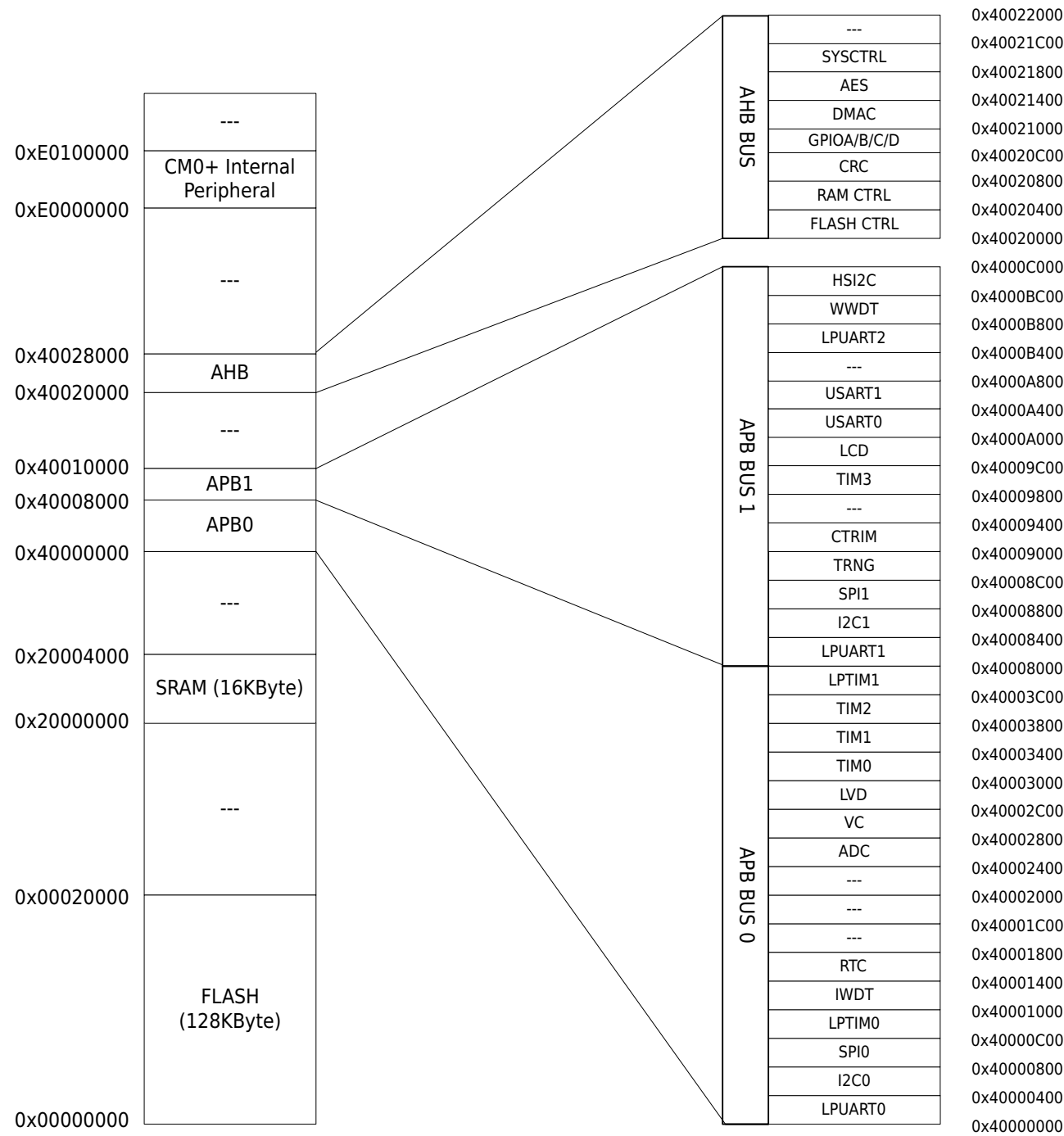
系统由以下部分组成：

- 2 个 AHB 总线 Master：
 - ▶ Cortex-M0+
 - ▶ DMA 控制器
- 多个 AHB 总线 Slaves：
 - ▶ Flash 存储器
 - ▶ SRAM 存储器
 - ▶ AHB to APB Bridge0，包含所有 APB0 接口外设
 - ▶ AHB to APB Bridge1，包含所有 APB1 接口外设
 - ▶ AHB 外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。

2.2 系统地址划分

系统的地址区域划分：



2.3 存储器和模块地址分配

表 2-1 地址划分表

类别	Boundary Address	Size	Memory Area
存储器	0x00000000 - 0x0001FFFF	128KByte	FLASH
	0x20000000 - 0x20003FFF	16KByte	SRAM
APB0 外设	0x40000000 - 0x40007FFF	32KByte	-
APB1 外设	0x40008000 - 0x4000FFFF	32KByte	-
AHB 外设	0x40020000 - 0x40027FFF	32KByte	-

3 工作模式

3.1 概述

本产品¹在电源管理模块的控制下可实现三种工作模式，各工作模式下可工作的功能模块及功耗各不相同。本产品支持以下工作模式：

- 1. 运行模式（Active Mode）：CPU 运行，周边功能模块运行。
- 2. 休眠模式（Sleep Mode）：CPU 停止，周边功能模块运行。
- 3. 深度休眠模式（DeepSleep Mode）：CPU 停止，高速时钟模块停止运行。

3.2 工作模式切换

本产品工作时可在三种工作模式之间自由切换。在运行模式下执行相关指令，可使本产品进入休眠模式或深度休眠模式。从休眠模式或深度休眠模式通过中断唤醒，回到运行模式。

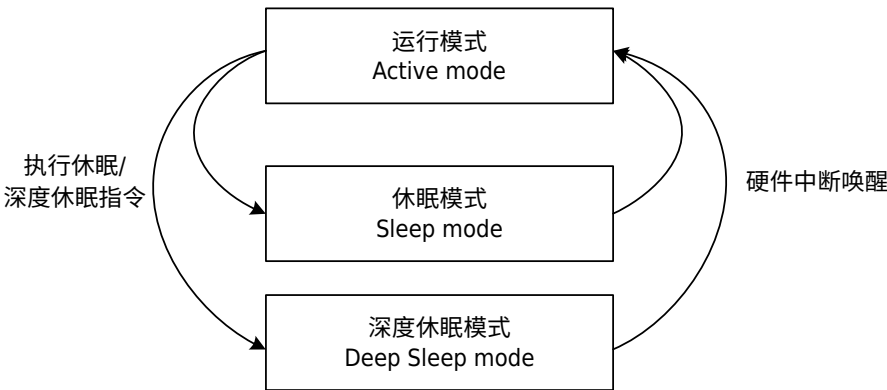


图 3-1 工作模式状态转换图

在各种工作模式下，CPU 可响应的中断详见《中断控制器（NVIC）》章节“[中断源](#)”。在不同工作模式下，系统可响应的复位类型如下：

表 3-1 不同工作模式下系统可响应的复位类型

编号	复位来源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR/PDR	✓	✓	✓
[1]	外部 Reset Pin 复位	✓	✓	✓
[2]	LVD 复位	✓	✓	✓
[3]	IWDT 复位	✓	✓	✓
[4]	Cortex-M0+ LOCKUP 硬件复位	✓	×	×
[5]	Cortex-M0+ SYSRESETREQ 软件复位	✓	×	×
[6]	WWDT 复位	✓	✓	×

3.3 运行模式 (Active Mode)

系统在电源上电复位后，或从各低功耗唤醒后，微控制器 MCU 处于运行状态。当 CPU 不需继续运行时，可以利用多种低功耗模式来节能，例如等待某个外部事件时。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

运行模式下，几种常用的降低芯片功耗的方法：

1. 在运行模式下，通过对预分频寄存器 (SYSCTRL0.HCLK_PRS、SYSCTRL0.PCLK_PRS) 进行编程，可以降低任意一个系统时钟 (HCLK、PCLK) 的速度。进入休眠模式前，也可以利用预分频器来降低外设的时钟。
2. 在运行模式下，关闭不使用外设的时钟 (PERI_CLKENx) 来减少功耗。
3. 在执行 WFI 指令前关闭不使用外设的时钟 (PERI_CLKENx)。
4. 在某些场景中，可以使用低功耗模式代替运行模式，因为本产品的唤醒时间极短，亦可满足系统的实时响应的需求。

3.4 休眠模式 (Sleep Mode)

使用 WFI 指令可以进入休眠模式。休眠模式下 CPU 停止运行，但时钟模块、系统时钟、NVIC 中断处理以及周边的功能模块仍都正常工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

进入休眠模式

通过执行 WFI 指令进入休眠状态。根据 Cortex-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择休眠模式进入机制：

- SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 被执行时，微控制器立即进入休眠模式。
- SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

退出休眠模式

如果执行 WFI 指令进入休眠模式，任意一个被高优先级嵌套向量中断控制器响应的外设中断都能将系统从休眠模式唤醒。

注意事项

- SLEEPONEXIT 位置 1，执行完中断自动进入 Sleep，程序不需要写 __wfi();
- SLEEPONEXIT 位清 0，main() 执行 __wfi() 后进入 Sleep，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 Sleep。等待后续中断触发。
- SLEEPONEXIT 位不影响 __wfi() 指令的执行。SLEEPONEXIT=0：main() 执行 wfi() 后进入 Sleep，中断触发且执行完中断程序返回 main() 后，继续往下执行；
- 若在中断中进入 Sleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。
- 执行 WFE 进入休眠和唤醒的内容，请查询 Cortex-M0+ 相关手册。

3.5 深度休眠模式 (DeepSleep Mode)

使用 SLEEPDEEP 配合 WFI 指令可以进入深度休眠模式。在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置是否运行，NVIC 中断处理仍可以工作。

- 当高速时钟作为系统时钟，系统进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度休眠前的状态。
- 当低速时钟作为系统时钟，系统进入深度休眠模式，由于低速时钟不会自动关闭，进入深度休眠模式时保持运行。只有 ARM Cortex-M0+ 不运行，其他模块都运行。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。

- 系统进入深度休眠状态，不会改变端口状态，在进入深度休眠前根据需要更改 IO 的状态为深度休眠下的状态。

进入深度休眠模式

首先设置 Cortex-M0+ 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入深度休眠状态。根据系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择深度休眠模式进入机制：

- SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 被执行时，微控制器立即进入深度休眠模式。
- SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入深度休眠模式。

退出深度休眠模式

如果执行 WFI 指令进入深度休眠模式，任意一个被高优先级嵌套向量中断控制器响应的外设中断（DeepSleep 下可运行的周边模块中断）都能将系统从深度休眠模式唤醒。

唤醒设置请参考《系统控制器（SYSCRTL）》章节“中断唤醒控制”相关描述。

注意事项

- SLEEPONEXIT 位置 1，执行完中断自动进入 DeepSleep，程序不需要写 __wfi()。
- SLEEPONEXIT 位清 0，main() 执行 __wfi() 后进入 DeepSleep，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 DeepSleep。等待后续中断触发。
- SLEEPONEXIT 位不影响 __wfi() 指令的执行。SLEEPONEXIT=0：main() 执行 wfi() 后进入 DeepSleep，中断触发且执行完中断程序返回 main() 后，继续往下执行。
- 若在中断中进入 DeepSleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。
- 执行 WFE 进入深度休眠和唤醒的内容，请查询 Cortex-M0+ 相关手册。

3.6 功能/模块工作状态

在三种工作模式下功能/模块的工作状态对比如下表所示：

表 3-2 各模式下功能/模块工作状态表

功能/模块	运行模式	休眠模式	深度休眠模式
Cortex-M0+	✓	×	×
XTH	✓	✓	×
XTL	✓	✓	✓
RCH	✓	✓	×
RCL	✓	✓	✓
RESET	✓	✓	✓
GPIO	✓	✓	✓
I2C0/1	✓	✓	×
HSI2C	✓	✓	✓ (3)
SPI0/1	✓	✓	×
CTRIM	✓	✓	✓ (2)
FLASH	✓	✓	×
RAM	✓	✓	×
DMAC	✓	✓	×
TIM0/1/2/3	✓	✓	×
LPTIM0/1	✓	✓	✓ (2)
RTC	✓	✓	✓ (2)
IWDT	✓	✓	✓

功能/模块	运行模式	休眠模式	深度休眠模式
WWDT	✓	✓	×
USART0/1	✓	✓	×
LPUART0/1/2	✓	✓	✓ ⁽²⁾
CRC	✓	✓	×
TRNG	✓	✓	×
AES	✓	✓	×
LCD	✓	✓	✓
ADC	✓	✓	×
VC0/1	✓	✓	✓ ⁽¹⁾
LVD	✓	✓	✓
SWD	✓	✓	×
POR/PDR	✓	✓	✓



注意

- 1. 深度休眠模式如果使用 VC 唤醒功能，VC 滤波时钟不能选择 PCLK。
- 2. 工作时钟为 PCLK/高速时钟时不支持深度休眠模式唤醒。
- 3. 只有从机模式支持 DeepSleep 唤醒。
- 4. 在执行 WFI 命令进入深度休眠模式之前，必须关闭所有不用于唤醒 MCU 的中断。

3.7 寄存器

3.7.1 系统控制寄存器（SCB_SCR）（Cortex-M0+内核系统控制寄存器）

基地址：0xE000ED10

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved SEVONPENDReservedSLEEPDEEPSLEEPONEXITReserved																															
Access	RW																															

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4	SEVONPEND	中断挂起标记 ● 0b1：每次新的中断挂起都会产生一个事件，如果使用了 WFE 进入休眠或深度休眠模式，可用于唤醒处理器	RW
3	Reserved	-	-
2	SLEEPDEEP	休眠模式选择 ● 0b0：休眠（Sleep）模式 ● 0b1：深度休眠（DeepSleep）模式	RW
1	SLEEPONEXIT	自动休眠标记 ● 0b0：该特性被禁用 ● 0b1：当退出异常处理并返回程序线程时，处理器自动进入休眠或深度休眠模式（WFI）	RW
0	Reserved	-	-

4 系统控制器 (SYSCTRL)

4.1 系统时钟介绍

时钟控制模块主要提供系统时钟、外设时钟和外设复位控制功能。

本产品提供以下几个不同的时钟源作为系统时钟：

- RCH-内部高速 RC 时钟（输出频率为 4/6/32/48MHz）
- RCL-内部低速 RC 时钟（38.4kHz 与 32.768kHz 可配置）
- XTH-外部高速晶振时钟
- XTL-外部低速晶振时钟



说明

1. 切换系统时钟的时钟源时，请严格按照操作步骤进行切换，请参考[系统时钟切换](#)。
2. XTL 可以不接晶振，直接从 PC14 管脚输入 32.768kHz 的时钟信号。XTH 可以不接晶振，直接从 PD00 管脚输入最大 32MHz 的时钟信号。

本产品还包含以下两个辅助时钟：

- RC10K-内部 10kHz 时钟，供 IWDT 模块使用
- RC256K-内部 256kHz 时钟，供 LVD 和 VC 模块使用

4.1.1 时钟架构图

本产品的时钟架构如下所示：

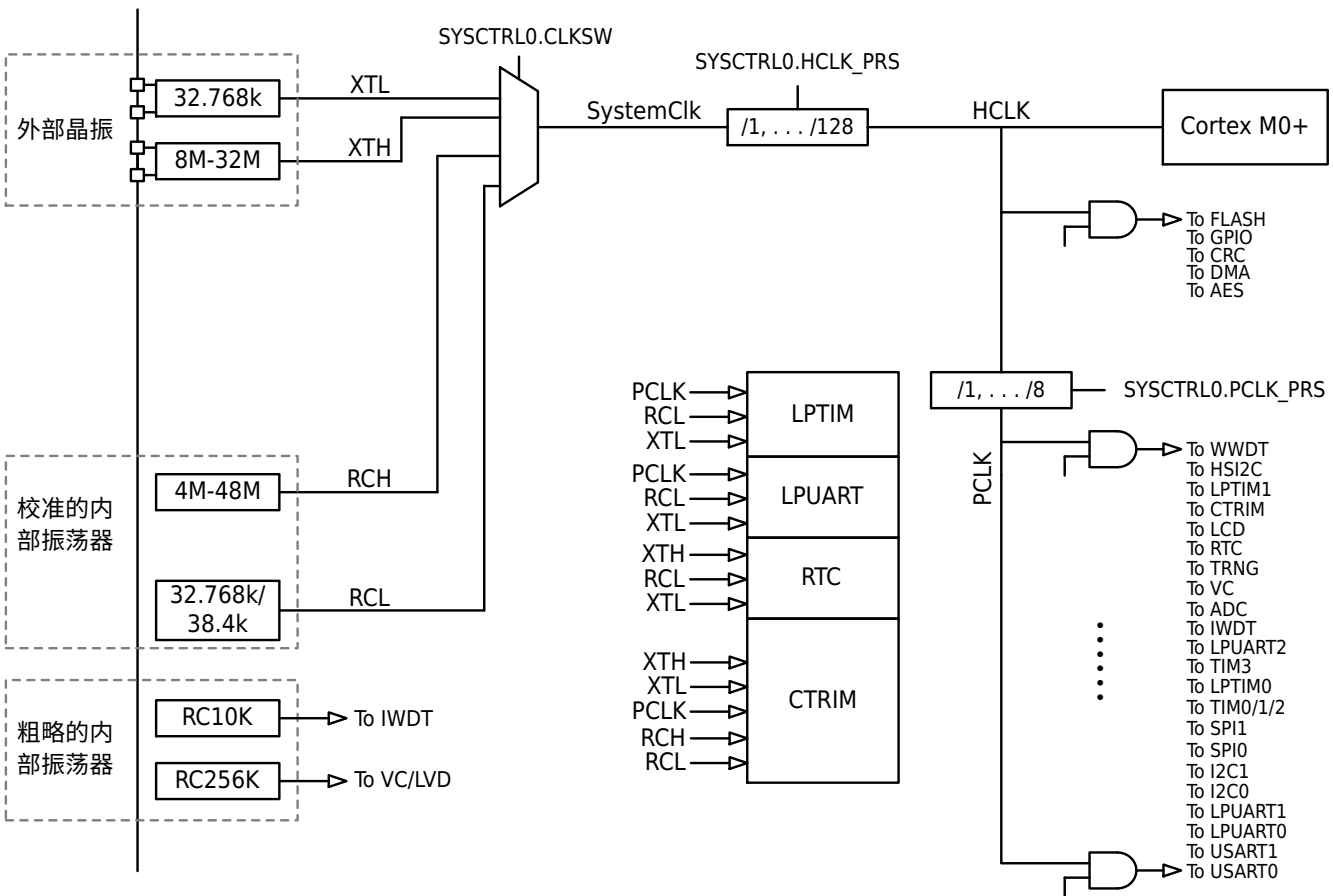


图 4-1 时钟架构图

4.1.2 时钟类型

4.1.2.1 RCH

RCH 是内部高速时钟振荡器，其频率可配置、可校准，也是芯片上电或部分复位后使用的默认时钟源，且默认频率为 4MHz。通过 RCH_CR 寄存器可以配置时钟频率。出厂时已预置校准好的 4 个频率配置值：4MHz、6MHz、32MHz、48MHz，使用时可直接加载配置；也可根据实际需求通过调整该寄存器的校准值改变至其它频率或进行更高精度的校准。

当系统进入 DeepSleep 模式时，此高速时钟会自动关闭。

更改 RCH 时钟源频率的操作请参见“[系统时钟切换](#)”章节相关描述。

内部高速时钟从启动到稳定用时较短。为了在深度休眠模式下能快速响应中断，建议进入深度休眠模式前将系统时钟源切换为 RCH。

4.1.2.2 RCL

RCL 是内部低速时钟，其频率可配置、可校准。

通过 RCL_CR 寄存器可以配置时钟频率，出厂时已预置校准好的 2 个频率配置值：32.768kHz、38.4kHz，使用时可直接加载配置；也可根据实际需求通过调整该寄存器的校准值改变至其它频率或进行更高精度的校准。

当系统进入 DeepSleep 模式时，该时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

4.1.2.3 XTL

XTL 外部低速晶振时钟需外接一个 32.768kHz 的低功耗晶振，具有超高精度以及超低功耗。当系统进入 DeepSleep 模式时，该时钟不会自动关闭。超低功耗模式下工作的外设模块可以选择 XTL 作为其时钟。XTL 支持外部有源时钟作为输入，可以不接晶体，直接从 PC14 管脚输入时钟信号。从 PC14 输入有源时钟信号的方法为：

- 配置 PC14 引脚为 GPIO 输入；
- 配置 XTL 稳定时间 XTL_CR.STARTUP；
- 设置 SYSCTRL1.EXTL_EN 为 1；
- 等待 XTL 稳定标志置起 XTL_CR.STABLE。



注意

晶体及其匹配电容及外部时钟源信号需符合《数据手册》电气特性中低速外部时钟 XTL 的相关要求。

4.1.2.4 XTH

外部高速晶振时钟需外接一个 8~32MHz 的高速晶振。当系统进入 DeepSleep 模式时，该时钟会自动关闭。XTH 支持外部有源时钟作为输入，可以不接晶体，直接从 PD00 管脚输入 4~32MHz 的时钟信号。从 PD00 输入时钟信号的方法为：

- 配置 PD00 引脚为 GPIO 输入；
- 配置 XTH 稳定时间 XTH_CR.STARTUP；
- 设置 SYSCTRL1.EXTH_EN 为 1；
- 等待 XTH 稳定标志置起 XTH_CR.STABLE。



注意

晶体及其匹配电容及外部时钟源信号需符合《数据手册》电气特性中高速外部时钟 XTH 的相关要求。

4.1.3 时钟启动过程

上述时钟源都需要启动稳定时间，以 XTH 为例说明时钟的启动稳定过程。

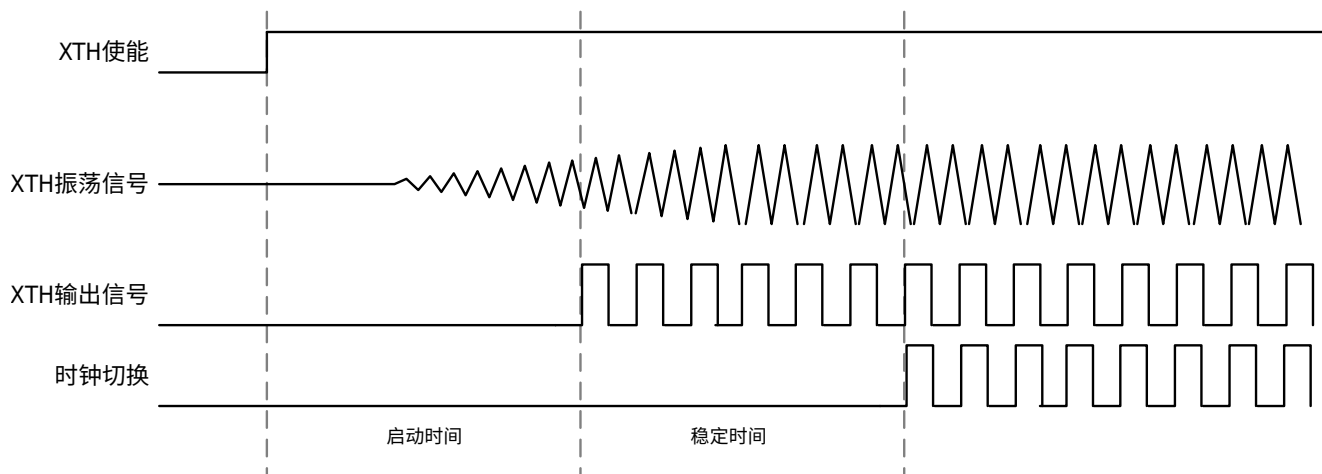


图 4-2 晶振时钟启动示意图

4.2 系统时钟切换

4.2.1 系统时钟切换简介

系统时钟的时钟源可通过 SYSCTRL0.CLKSW 在 RCH、RCL、XTH、XTL 之间进行切换。切换时 RCH、RCL、XTH、XTL 这几个时钟源中的任意两者可以相互切换。时钟切换操作必须按照下文所描述的时钟切换流程进行，否则可能出现异常。

系统时钟源经过系统分频后可作为 HCLK，可通过 SYSCTRL0.HCLK_PRS 设置合适的分频值。

时钟切换时需要根据当前及目标 HCLK 的频率，同步配置 FLASH_CR.WAIT：

- 当 $HCLK \leq 24MHz$ 时，应设置 FLASH_CR.WAIT 为 0；
- 当 $HCLK > 24MHz$ 时，应设置 FLASH_CR.WAIT 为 1；



说明

设置 FLASH_CR.WAIT 的具体操作方法，详见《FLASH 控制器》章节。

4.2.2 切换流程

4.2.2.1 标准的时钟切换流程

操作步骤

步骤 1. 如新时钟源需要外部引脚，则将该引脚设置为适当的模式。



说明

接外部晶振时需要模拟引脚；接外部时钟输入时需要 GPIO 输入并使能外部时钟输入。

步骤 2. 配置新时钟源的振荡参数。

步骤 3. 使能新时钟源的振荡器。

步骤 4. 如果新时钟源的频率大于原先时钟源频率，则根据新时钟源的频率，按照《FLASH 控制器》章节流程，切换前，先配置 FLASH_CR.WAIT。

步骤 5. 等待新时钟源输出稳定的频率。

步骤 6. 配置 SYSCTRL0.CLKSW，选择系统时钟的来源为新时钟源。

步骤 7. 如果新时钟源的频率小于原先时钟源频率，根据新时钟源的频率，按照《FLASH 控制器》章节流程，切换后，再配置 FLASH_CR.WAIT。

步骤 8. 关闭不再使用的时钟源。

4.2.2.2 RCH 不同振荡频率间切换流程

背景信息

根据使用场景选择以下方式之一：

操作步骤

- 根据目标频率选择加载：从出厂预设的目标频率校准值存放地址将数据直接加载至 RCH_CR。
- 根据实际需要直接设置：向 RCH_CR 写入校准值，调整 RCH 的目标输出频率。

4.2.3 切换示例

4.2.3.1 从其它时钟切换到 XTL 示例

操作步骤

- 步骤 1. 设置 PCADS.14 及 PCADS.15 为 1，配置 PC14/PC15 引脚为模拟端口。
- 步骤 2. 根据晶体及系统特性，配置 XTL_CR 寄存器中的参数。
- 步骤 3. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 4. 设置 SYSCTRL0.XTL_EN 为 1，使能晶振振荡电路。
- 步骤 5. 查询等待 XTL_CR.STABLE 标志变为 1，晶振输出稳定时钟。
- 步骤 6. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 7. 设置 SYSCTRL0.CLKSW 为 3，将系统时钟切换为 XTL。
- 步骤 8. 参考《FLASH 控制器》章节设置 FLASH_CR.WAIT 为 0。
- 步骤 9. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 10. 关闭不再使用的时钟源。

4.2.3.2 从其它时钟切换到 XTH 示例

操作步骤

- 步骤 1. 设置 PDADS.0 及 PDADS.1 为 1，配置 PD00/PD01 引脚为模拟端口。
- 步骤 2. 根据晶振特性，配置 XTH_CR[3:0]。
- 步骤 3. 设置 XTH_CR.STARTUP 为 3，选择合适的晶振稳定时间。
- 步骤 4. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 5. 设置 SYSCTRL0.XTH_EN 为 1，使能晶振振荡电路。
- 步骤 6. 根据当前时钟和 XTH 两者中较高的频率，参考《FLASH 控制器》章节配置 FLASH_CR.WAIT。
- 步骤 7. 查询等待 XTH_CR.STABLE 标志变为 1 后，晶振输出稳定时钟。
- 步骤 8. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 9. 设置 SYSCTRL0.CLKSW 为 1，将系统时钟切换为 XTH。
- 步骤 10. 根据 XTH 的频率，参考《FLASH 控制器》配置 FLASH_CR.WAIT。
- 步骤 11. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 12. 关闭不再使用的时钟源。

4.2.3.3 从其它时钟切换到 RCL 示例

操作步骤

- 步骤 1. 配置 RCL_CR.TRIM 及 RCL_CR.STARTUP。
- 步骤 2. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 3. 设置 SYSCTRL0.RCL_EN 为 1，使能 RCL 振荡电路。
- 步骤 4. 查询等待 RCL_CR.STABLE 标志变为 1，RCL 输出稳定时钟。
- 步骤 5. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 6. 设置 SYSCTRL0.CLKSW 为 2，将系统时钟切换为 RCL。

步骤 7. 参考《FLASH 控制器》章节设置 FLASH_CR.WAIT 为 0。

步骤 8. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

步骤 9. 关闭不再使用的时钟源。

4.2.3.4 从其它时钟切换到 RCH 示例

操作步骤

步骤 1. 根据当前时钟和将要配置的 RCH 时钟两者中较高的频率，按照《FLASH 控制器》章节流程配置 FLASH_CR.WAIT。

步骤 2. 向 RCH_CR 寄存器写入预设的值，配置需要的时钟频率。

步骤 3. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

步骤 4. 设置 SYSCTRL0.RCH_EN 为 1，使能 RCH 振荡电路。

步骤 5. 查询等待 RCH_CR.STABLE 标志变为 1，RCH 输出稳定时钟。

步骤 6. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

步骤 7. 设置 SYSCTRL0.CLKSW 为 0，将系统时钟切换为 RCH。

步骤 8. 根据 RCH 的频率，按照《FLASH 控制器》章节流程配置 FLASH_CR.WAIT。

步骤 9. 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

步骤 10. 关闭不再使用的时钟源。

4.2.4 时钟切换示意图

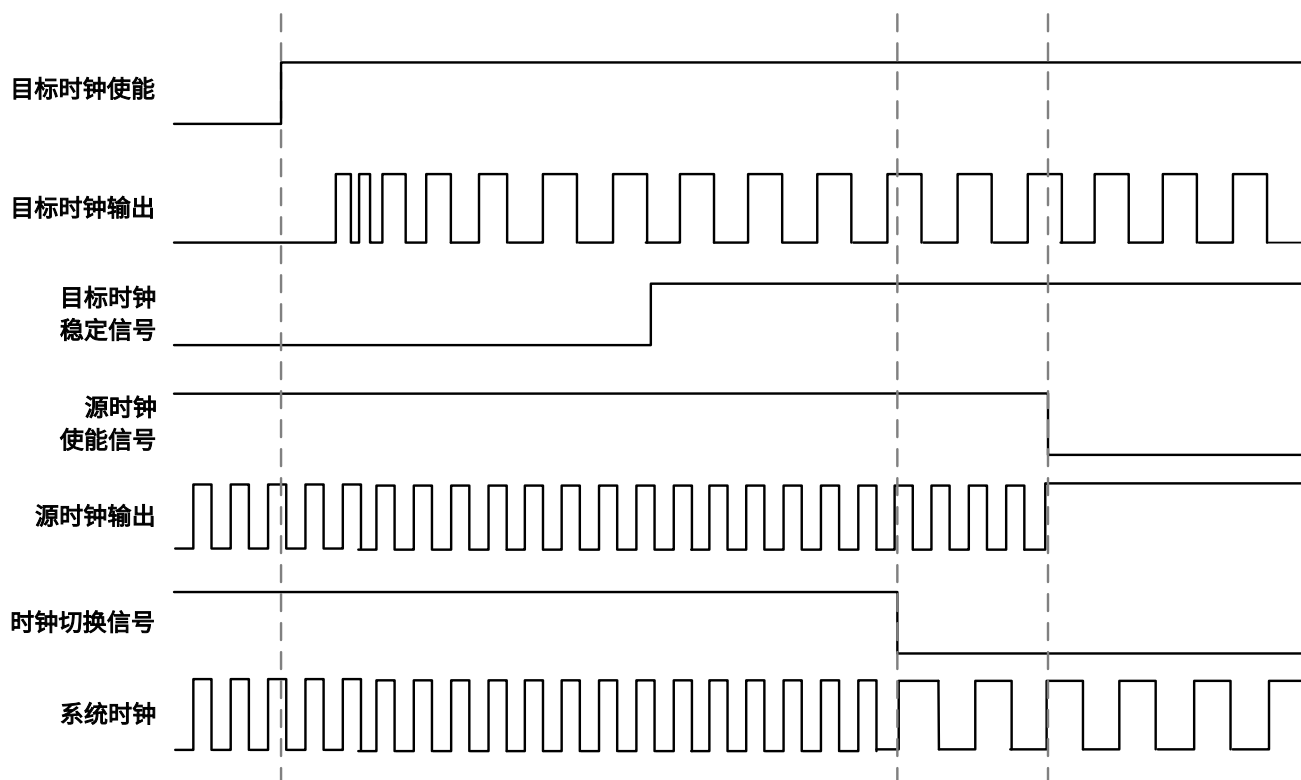


图 4-3 时钟切换示意图

4.3 时钟校准模块

本产品内嵌时钟校准电路 CTRIM 模块，具体使用可以参考相关章节描述。

4.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠/深度休眠状态时，会停止执行指令。在休眠/深度休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠/深度休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级>当前等级	Y	Y
0	IRQ 优先级≤当前等级	N	N
1	IRQ 优先级>当前等级	Y	N
1	IRQ 优先级≤当前等级	N	N

4.4.1 从深度休眠模式唤醒后执行中断服务

操作步骤

- 步骤 1. 使能需要唤醒处理器的模块所对应的 NVIC。
- 步骤 2. 使能需要唤醒处理器的模块所对应的中断。
- 步骤 3. 设置 SCB->SCR.SLEEPDEEP 为 1。
- 步骤 4. 执行 WFI 指令进入深度休眠模式。
- 步骤 5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务程序。

配置举例

```
SCB->SCR |= 0x00000004u;
while(1)
{
    __asm("WFI");
}
```

4.4.2 从深度休眠模式唤醒后不执行中断服务

操作步骤

- 步骤 1. 使能需要唤醒处理器的模块所对应的 NVIC。
- 步骤 2. 使能需要唤醒处理器的模块所对应的中断。
- 步骤 3. 设置 PRIMASK 为 1。
- 步骤 4. 设置 SCB->SCR.SLEEPDEEP 为 1。
- 步骤 5. 执行 WFI 指令进入深度休眠模式。
- 步骤 6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令。
- 步骤 7. 清除中断标志，清除中断挂起状态。
- 步骤 8. 执行用户定义的操作。

配置举例

```
__asm("CPSID I"); //Set PRIMASK
SCB->SCR |= 0x00000004u;
while(1)
{
    __asm("WFI");
    ... //清除唤醒中断的中断标志
    ... //清除唤醒中断的中断挂起状态
}
```



```
... //执行用户定义的操作
}
```

4.4.3 使用退出休眠特性

背景信息

退出休眠 (sleep-on-exit) 非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠或深度休眠模式。利用退出休眠特性，处理器可以尽可能多的处于休眠或深度休眠模式。

Cortex-M0+利用退出休眠特性进入休眠或深度休眠，这种情况与执行完异常退出后立即执行 WFI 的效果类似。需要注意的是，为了下次进入异常时不用再进行压栈操作，处理器不会执行出栈的过程。

操作步骤

- 步骤 1. 使能需要唤醒处理器的模块所对应的 NVIC。
- 步骤 2. 使能需要唤醒处理器的模块所对应的中断。
- 步骤 3. 设置 SCB->SCR.SLEEPDEEP 为 1。
- 步骤 4. 设置 SCB->SCR.SLEEPONEXIT 为 1。
- 步骤 5. 执行 WFI 指令进入深度休眠模式。
- 步骤 6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序。
- 步骤 7. 退出中断服务时自动进入深度休眠模式。

配置举例

```
SCB->SCR |= 0x00000004u;
SCB->SCR |= 0x00000002u;
while(1)
{
    __asm("WFI");
}
```

4.5 复位控制

4.5.1 复位控制器概述

本产品具有 7 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序会从复位向量处开始执行。

- 数字区域上电掉电复位 POR
- 外部 Reset PAD，低电平为复位信号
- WWDT 复位
- IWDT 复位
- LVD 低电压复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示。复位标志均由硬件置位，需要用户软件清零。芯片复位时，如果查询到 RESET_FLAG.POR15V 或 RESET_FLAG.POR5V 为 1 则为**上电复位**。**上电复位**时用户程序应当将寄存器 RESET_FLAG 清零，则下一次复位时可通过 RESET_FLAG 的相关比特判断复位来源。

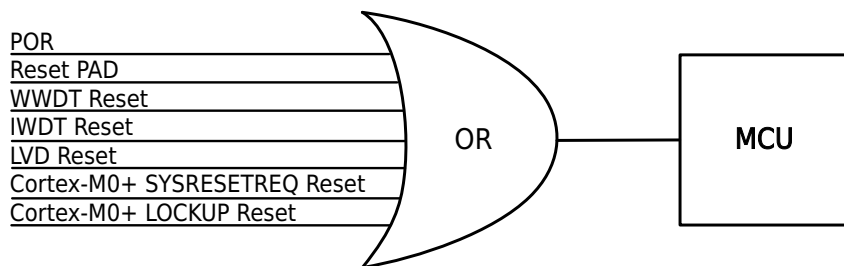


图 4-4 复位来源示意图

4.5.2 复位功能

4.5.2.1 上电下电复位 POR

本产品有两个供电区域：VCC 区域、VCAP 区域。所有的模拟模块及 IO 工作于 VCC 区域；其它模块工作于 VCAP 区域。

- VCC 区域上电时，当 VCC 电压低于 POR 阈值电压时，会产生 POR5V 信号；VCC 区域下电时，当 VCC 电压低于 PDR 阈值电压时，会产生 POR5V 信号。
- VCAP 区域上电时，当 VCAP 电压低于 POR 阈值电压时，会产生 POR15V 信号；VCAP 区域下电时，当 VCAP 电压低于 PDR 阈值电压时，会产生 POR15V 信号。

POR5V 信号和 POR15V 信号均会将芯片的寄存器复位到初始化状态。

4.5.2.2 外部复位管脚复位

当外部复位引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于一定时间宽度的毛刺信号，因此，加到复位引脚上的低电平信号必须大于该时间宽度，才能确保芯片可靠复位，具体时间参数请参见《数据手册》的“电气特性”章节“RESETB 引脚特性”。

4.5.2.3 看门狗复位

看门狗复位参见《独立看门狗 (IWDT) 》和《窗口看门狗 (WWDT) 》章节相关介绍。

4.5.2.4 LVD 低电压复位

LVD 低电压复位参见《低电压检测器 (LVD) 》章节相关介绍。

4.5.2.5 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位。

4.5.2.6 Cortex-M0+ LOCKUP 复位

当 Cortex-M0+ 遇到严重的异常时，会将自身 PC 指针停在当前地址处并锁死，在几个时钟周期延时之后复位整个 CORE 区域。SYSCTRL1.LOCKUP_EN 为 1 时，该复位功能有效。

4.6 寄存器

4.6.1 寄存器总表

基地址：0x40021800

表 4-2 系统控制器 (SYSCTRL) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	SYSCTRL0	系统控制寄存器 0	32
0x04	SYSCTRL1	系统控制寄存器 1	32
0x08	SYSCTRL2	系统控制寄存器 2	32
0x0C	RCH_CR	RCH 控制寄存器	32
0x10	XTH_CR	XTH 控制寄存器	32
0x14	RCL_CR	RCL 控制寄存器	32
0x18	XTL_CR	XTL 控制寄存器	32
0x1C	RESET_FLAG	复位标识寄存器	32
0x20	PERI_CLKEN0	外围模块时钟控制寄存器 0	32
0x24	PERI_CLKEN1	外围模块时钟控制寄存器 1	32
0x28	PERI_RESET0	外围模块复位控制寄存器 0	32
0x2C	PERI_RESET1	外围模块复位控制寄存器 1	32
0x38	DEBUG_ACTIVE	外设调试功能有效性配置寄存器	32

4.6.2 系统控制寄存器 0 (SYSCTRL0)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00008001																															
Name	Reserved																WAKEUP_BYRCH	Reserved		PCLK_PRS	HCLK_PRS		CLKSW		Reserved	XTL_EN	RCL_EN	XTH_EN	RCH_EN			
Access																	RW								RW	RW	RW		RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	WAKEUP_BYRCH	唤醒后系统时钟源控制位 0b0: 从 DeepSleep 模式唤醒后, 不改变系统时钟源 0b1: 从 DeepSleep 模式唤醒后, 系统时钟源为 RCH, 原时钟继续使能	RW
14:13	Reserved	-	-
12:11	PCLK_PRS	PCLK 时钟来源选择 0b00: HCLK 0b01: HCLK/2 0b10: HCLK/4 0b11: HCLK/8	RW
10:8	HCLK_PRS	HCLK 系统分频选择 0b000: SystemClk 0b001: SystemClk/2 0b010: SystemClk/4 0b011: SystemClk/8 0b100: SystemClk/16 0b101: SystemClk/32 0b110: SystemClk/64 0b111: SystemClk/128	RW
7:5	CLKSW	SystemClk 时钟来源选择 0b000: RCH 0b001: XTH 0b010: RCL 0b011: XTL - 其他: 禁止设置	RW
4	Reserved	-	-
3	XTL_EN	外部低速晶振 XTL 使能控制 0b0: 关闭 0b1: 使能  说明 需要将 PC14、PC15 设置成模拟端口。	RW

位/位域	标记	功能描述	读写
2	RCL_EN	内部低速时钟 RCL 使能控制 ● 0b0: 关闭 ● 0b1: 使能	RW
1	XTH_EN	外部高速晶振 XTH 使能控制 ● 0b0: 关闭 ● 0b1: 使能  说明 需要将 PD00、PD01 设置成模拟端口。 当系统进入 DeepSleep，此高速时钟会自动关闭。	RW
0	RCH_EN	内部高速时钟 RCH 使能控制 ● 0b0: 关闭 ● 0b1: 使能  说明 当系统进入 DeepSleep，此高速时钟会自动关闭。	RW



说明

每次改写 SYSCTRL0 或 SYSCTRL1 的值，均需要先对 SYSCTRL2 依次写入 0x5A5A、0xA5A5。这样的步骤可有效避免对 SYSCTRL0 或 SYSCTRL1 寄存器的误操作。

4.6.3 系统控制寄存器 1 (SYSCTRL1)

Offset	Bit Position																																									
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Reset	0x00000008																																									
Name	Reserved																								SWD_USE_IO		SPI_PCTRL		LOCKUP_EN		Reserved		CLOCK_FAULT_EN		XTL_ALWAYS_ON		EXTL_EN		EXTH_EN		Reserved	
Access																									RW		RW		RW				RW		RW		RW		RW			

位/位域	标记	功能描述	读写
31:9	Reserved	-	-
8	SWD_USE_IO	SWD 端口功能配置 ● 0b0: SWD 端口 ● 0b1: GPIO 端口	RW
7	SPI_PCTRL	SPI 端口控制选择 ● 0b0: SPI 控制 ● 0b1: GPIO 控制	RW
6	LOCKUP_EN	Cortex-M0+ LockUp 使能 ● 0b0: 关闭 ● 0b1: 使能	RW
5	Reserved	-	-
4	CLOCK_FAULT_EN	时钟错误检测使能 ● 0b0: 关闭 ● 0b1: 使能	RW
3	XTL_ALWAYS_ON	XTL 使能权限控制 ● 0b0: SYSCTRL0.XTL_EN 可置位可清零 ● 0b1: SYSCTRL0.XTL_EN 只可置位	RW
2	EXTL_EN	外部 XTL 时钟来源控制 ● 0b0: XTL 由晶体产生 ● 0b1: XTL 时钟源从 PC14 输入，当使用该时钟做系统时钟时，不能关闭 EXTL_EN（外部时钟源需符合《数据手册》电气特性中低速外部时钟 XTL 的相关要求）  说明 使用 PC14 输入时钟时，需设置 SYSCTRL0.XTL_EN 为 0。	RW

位/位域	标记	功能描述	读写
1	EXTH_EN	外部 XTH 时钟来源控制 0b0: XTH 由晶体产生 0b1: XTH 时钟源从 PD00 输入  说明 使用 PD00 输入时钟时： 1. 必须设置 SYSCTRL0.XTH_EN 为 0。 2. 必须配置 SYSCTRL0.WAKEUP_BYRCH 为 1，否则低功耗模式下电流异常。	RW
0	Reserved	-	-

 **说明**

每次改写 SYSCTRL0 或 SYSCTRL1 的值，均需要先对 SYSCTRL2 依次写入 0x5A5A、0xA5A5。这样的步骤可有效避免对 SYSCTRL0 或 SYSCTRL1 寄存器的误操作。

4.6.4 系统控制寄存器 2 (SYSCTRL2)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																SYSCTRL2															
Access																	WO															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	SYSCTRL2	寄存器 SYSCTRL0 和 SYSCTRL1 保护控制寄存器 对 SYSCTRL2 先写 0x5A5A，再写 0xA5A5，启动对于寄存器 SYSCTRL0 和 SYSCTRL1 的写操作，只要对寄存器 SYSCTRL0 或 SYSCTRL1 进行了写操作，这个保护位自动恢复保护状态。后续再对 SYSCTRL0 或 SYSCTRL1 进行写操作时还需要重复先前的操作。	WO

4.6.5 RCH 控制寄存器 (RCH_CR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0X00XXXX																															
Name	STABLE	Reserved														FSEL	TRIMC						TRIMF						Reserved			
Access	RO															RW	RW						RW									

位/位域	标记	功能描述	读写
31	STABLE	RCH 时钟稳定标志位 ● 0b0: RCH 未稳定, 不可以被内部电路使用 ● 0b1: RCH 已经稳定, 可以被内部电路使用  说明 该位上电时复位值为 0, 在实际使用过程中, RCH 稳定后, 读出恒为 1。	RO
30:18	Reserved	-	-
17:16	FSEL	工作频率选择 该位无需单独设置, 参数选择详见说明 ● 0b00: 4MHz ● 0b01: 6MHz ● 0b10: 32MHz ● 0b11: 48MHz	RW
15:8	TRIMC	时钟频率粗调 参数选择详见说明	RW
7:2	TRIMF	时钟频率细调 参数选择详见说明	RW
1:0	Reserved	-	-

 **说明**

RCH 时钟频率调整, 更改该寄存器的数值即可调整 RCH 的输出频率。

Flash 中已保存了以下几组频率的校准值和配置参数, 将 Flash 内的指定地址的数据全部读出并写入整个 RCH_CR 寄存器即可获得精准的频率。

- 48MHz 校准值地址: 0x00101BB0 - 0x00101BB3
- 32MHz 校准值地址: 0x00101BB4 - 0x00101BB7
- 6MHz 校准值地址: 0x00101BB8 - 0x00101BBB
- 4MHz 校准值地址: 0x00101BBC - 0x00101BBF

4.6.6 XTH 控制寄存器 (XTH_CR)

Offset	Bit Position																																				
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000016																																				
Name	Reserved																					FAULT_STATE		DETEN		Reserved		DETIE		STABLE		STARTUP		XTH_FSEL		DRIVER	
RW																						RW				RW		RO		RW		RW		RW			
Access																						RW		RW				RW		RO		RW		RW		RW	

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	FAULT_STATE	XTH 工作标志 写 0 清除标志, 写 1 无效 读取: 0b0: 工作正常 0b1: 工作失效	RW
9	DETEN	停振检测使能 0b0: 关闭停振检测 0b1: 开启停振检测  说明 1. 需要设置 SYSCTRL0.XTH_EN 为 1, 读取 XTH_CR.STABLE 为 1 后, 才能开启停振检测。 2. 当进入 DeepSleep 中断时, 需将该功能关闭, 否则中断唤醒后, 会产生该中断请求。	RW
8	Reserved	-	-
7	DETIE	停振检测中断使能控制 0b0: 禁止 0b1: 使能	RW
6	STABLE	外部高速晶体振荡稳定标志位 0b0: XTH 未稳定, 不可以被内部电路使用 0b1: XTH 已经稳定, 可以被内部电路使用	RO
5:4	STARTUP	外部高速晶体振荡稳定时间选择 0b00: 8192 个周期 0b01: 32768 个周期 0b10: 131072 个周期 0b11: 262144 个周期  注意 如果 XTH 稳定时间不足, 在进行时钟切换时或从深度休眠唤醒时, 系统不能稳定工作。	RW

位/位域	标记	功能描述	读写
3:2	XTH_FSEL	外部晶振工作频率选择 ● 0b00: 8M ● 0b01: 8M~16M ● 0b10: 16M~24M ● 0b11: 24M~32M	RW
1:0	DRIVER	外部晶振驱动能力选择 ● 0b00: 最弱驱动能力 ● 0b01: 较弱驱动能力 ● 0b10: 较强驱动能力 (推荐值) ● 0b11: 最强驱动能力  说明 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。	RW

4.6.7 RCL 控制寄存器 (RCL_CR)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0X000XXX（上电复位后 TRIM 值为 32.768kHz 校准值地址存储的值）																															
Name	Reserved																				RO	STABLE	Reserved	MSEL	TRIM							
Access																					RO			RW	RW							

位/位域	标记	功能描述	读写
31:13	Reserved	-	-
12	STABLE	内部低速时钟 RCL 稳定标志位 0b0: RCL 未稳定, 不可以被内部电路使用 0b1: RCL 已经稳定, 可以被内部电路使用	RO
11	Reserved	-	-
10	MSEL	模式选择 0b0: 高性能模式 (HPM) 0b1: 低功耗模式 (LPM)	RW
9:0	TRIM	内部低速时钟频率调整 Flash 中保存了 4 组频率的校准值。将 Flash 内的校准值读出并写入 RCL_CR 即可获得精准的频率。 38.4kHz 校准值地址: - 高性能模式 (HPM): 0x00101BAC-0x00101BAF - 低功耗模式 (LPM): 0x00101BC8-0x00101BCB 32.768kHz 校准值地址: - 高性能模式 (HPM): 0x00101BA8-0x00101BAB - 低功耗模式 (LPM): 0x00101BC4-0x00101BC7	RW

4.6.8 XTL 控制寄存器 (XTL_CR)

Offset	Bit Position																																					
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000029																																					
Name	Reserved																						FAULT_STATE		DETEN		DETIME		DETIE		STABLE		STARTUP		AMP_SEL		DRIVER	
RW																							RW		RW		RW		RO		RW		RW		RW			
Access																																						

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	FAULT_STATE	XTL 工作状态标志 写 0 清除标志，写 1 无效 读取： ● 0b0：工作正常 ● 0b1：工作失效	RW
9	DETEN	停振检测使能 ● 0b0：关闭停振检测 ● 0b1：开启停振检测  说明 需要设置 SYSCTRL0.XTL_EN 为 1，读取 XTL_CR.STABLE 为 1 后，才能开启停振检测。	RW
8	DETIME	晶体失效检测间隔选择 ● 0b0：档位 1，2ms 检测间隔 ● 0b1：档位 2，8ms 检测间隔  说明 SYSCTRL0.XTL_EN 为 1 时禁止修改该位。	RW
7	DETIE	停振检测中断使能控制 ● 0b0：禁止 ● 0b1：使能	RW
6	STABLE	外部低速晶体振荡稳定标志位 ● 0b0：XTL 未稳定，不可以被内部电路使用 ● 0b1：XTL 已经稳定，可以被内部电路使用	RO
5:4	STARTUP	外部低速晶体振荡稳定时间选择 ● 0b00：4096 个周期 ● 0b01：8192 个周期 ● 0b10：16384 个周期（推荐值） ● 0b11：32768 个周期	RW

位/位域	标记	功能描述	读写
3:2	AMP_SEL	XTL 晶振振荡幅度的调整 ● 0b00：最小振幅 ● 0b01：正常振幅 ● 0b10：较大振幅（推荐值） ● 0b11：最大振幅	RW
1:0	DRIVER	XTL 晶振驱动能力选择 ● 0b00：最弱驱动能力 ● 0b01：较弱驱动能力（推荐值） ● 0b10：较强驱动能力 ● 0b11：最强驱动能力  说明 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大；驱动能力越弱则功耗越小。	RW

4.6.9 复位标识寄存器 (RESET_FLAG)



说明

上电复位时复位值为 0x03。程序判断本次复位源后应该清除该寄存器，以便下次发生复位后判断复位源。

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000003																															
Name	Reserved														WWDT	RSTB	SYSREQ	LOCKUP	Reserved	IWDT	LVD	POR15V	POR5V									
Access															RW	RW	RW	RW		RW	RW	RW	RW									

位/位域	标记	功能描述	读写
31:9	Reserved	-	-
8	WWDT	WWDT 复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
7	RSTB	RESETB 端口复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
6	SYSREQ	Cortex-M0+ CPU 软件复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
5	LOCKUP	Cortex-M0+ CPU Lockup 复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
4	Reserved	-	-
3	IWDT	IWDT 复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
2	LVD	LVD 复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW

位/位域	标记	功能描述	读写
1	POR15V	VCAP 域复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW
0	POR5V	VCC 电源域复位标志 写 0 清除，写 1 无效 读取： ● 0b0：未发生复位 ● 0b1：发生复位	RW

4.6.10 外围模块时钟控制寄存器 0 (PERI_CLKEN0)

Offset	Bit Position																																				
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x80800000																																				
Name	FLASH	HSI2C	DMA	GPIO	AES	CRC	Reserved				LCD	Reserved		RTC	Reserved		TRNG	VC	ADC	IWD	Reserved				TIM3	Reserved		LPTIM0	TIM012	SPI1	SPI0	I2C1	I2C0	LPUART1	LPUART0	USART1	USART0
Access	RW	RW	RW	RW	RW	RW				RW		RW			RW	RW	RW	RW				RW			RW	LPTIM0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

位/位域	标记	功能描述	读写
31	FLASH	FLASH 控制器模块时钟使能 关闭后 FLASH 配置寄存器不可写，FLASH 中的程序仍然可以运行。 ● 0b0：关闭 ● 0b1：使能	RW
30	HSI2C	HSI2C 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
29	DMA	DMAC 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
28	GPIO	GPIO 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
27	AES	AES 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
26	CRC	CRC 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
25:23	Reserved	-	-
22	LCD	LCD 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
21	Reserved	-	-
20	RTC	RTC 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
19	Reserved	-	-
18	TRNG	TRNG 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW

位/位域	标记	功能描述	读写
17	VC	VC、LVD 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
16	ADC	ADC、BGR 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
15	IWDT	IWDT 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
14:12	Reserved	-	-
11	TIM3	TIM3 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
10	Reserved	-	-
9	LPTIM0	LPTIM0 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
8	TIM012	TIM0/1/2 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
7	SPI1	SPI1 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
6	SPI0	SPI0 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
5	I2C1	I2C1 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
4	I2C0	I2C0 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
3	LPUART1	LPUART1 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
2	LPUART0	LPUART0 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW
1	USART1	USART1 模块时钟使能 ● 0b0: 关闭 ● 0b1: 使能	RW

位/位域	标记	功能描述	读写
0	USART0	USART0 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW

 **说明**

上述外设在使用之前，需将相应控制位设置为使能。

4.6.11 外围模块时钟控制寄存器 1 (PERI_CLKEN1)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																					WWDT	Reserved	LPUART2	Reserved			LPTIM1	Reserved	CTRIM	Reserved	
Access																						RW		RW				RW		RW		

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	WWDT	WWDT 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
9	Reserved	-	-
8	LPUART2	LPUART2 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
7:5	Reserved	-	-
4	LPTIM1	LPTIM1 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
3	Reserved	-	-
2	CTRIM	CTRIM 模块时钟使能 ● 0b0：关闭 ● 0b1：使能	RW
1:0	Reserved	-	-

 **说明**

上述外设在使用之前，需将相应控制位设置为使能。

4.6.12 外围模块复位控制寄存器 0 (PERI_RESET0)

Offset	Bit Position																																																				
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
Reset	0x7C570BFF																																																				
Name	Reserved	HSI2C		DMA		GPIO		AES		CRC		Reserved		LCD		Reserved		RTC		Reserved		TRNG		VC		ADC		Reserved		TIM3		Reserved		LPTIM0		TIM012		SPI1		SPI0		I2C1		I2C0		LPUART1		LPUART0		USART1		USART0	
Access		RW	RW	RW	RW	RW	RW			RW		RW			RW	RW	RW				RW			RW	LPTIM0	RW	TIM012	RW	SPI1	RW	SPI0	RW	I2C1	RW	I2C0	RW	LPUART1	RW	LPUART0	RW	USART1	RW	USART0										

位/位域	标记	功能描述	读写
31	Reserved	-	-
30	HSI2C	HSI2C 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
29	DMA	DMAC 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
28	GPIO	GPIO 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
27	AES	AES 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
26	CRC	CRC 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
25:23	Reserved	-	-
22	LCD	LCD 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
21	Reserved	-	-
20	RTC	RTC 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
19	Reserved	-	-
18	TRNG	TRNG 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
17	VC	VC 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW

位/位域	标记	功能描述	读写
16	ADC	ADC、BGR 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
15:12	Reserved	-	-
11	TIM3	TIM3 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
10	Reserved	-	-
9	LPTIM0	LPTIM0 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
8	TIM012	TIM0/1/2 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
7	SPI1	SPI1 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
6	SPI0	SPI0 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
5	I2C1	I2C1 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
4	I2C0	I2C0 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
3	LPUART1	LPUART0 模块复位使能 LPUART1 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
2	LPUART0	● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
1	USART1	USART1 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW
0	USART0	USART0 模块复位使能 ● 0b0: 处于复位状态 ● 0b1: 正常工作状态	RW

4.6.13 外围模块复位控制寄存器 1 (PERI_RESET1)

Offset	Bit Position																																	
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000514																																	
Name	Reserved																					RW	WWDT	Reserved	RW	LPUART2	Reserved	Reserved	RW	LPTIM1	Reserved	RW	CTRIM	Reserved
Access																						RW			RW				RW			RW		

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	WWDT	WWDT 模块复位使能 ● 0b0：处于复位状态 ● 0b1：正常工作状态	RW
9	Reserved	-	-
8	LPUART2	LPUART2 模块复位使能 ● 0b0：处于复位状态 ● 0b1：正常工作状态	RW
7:5	Reserved	-	-
4	LPTIM1	LPTIM1 模块复位使能 ● 0b0：处于复位状态 ● 0b1：正常工作状态	RW
3	Reserved	-	-
2	CTRIM	CTRIM 模块复位使能 ● 0b0：处于复位状态 ● 0b1：正常工作状态	RW
1:0	Reserved	-	-

4.6.14 外设调试功能有效性配置寄存器 (DEBUG_ACTIVE)



说明

仅在 SWD 调试模式下，此寄存器设置才起作用。

Offset	Bit Position																															
0x38	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																WWDT	LPTIM1	TIM3	Reserved	RTC	IWDT	Reserved				LPTIM0	TIM2	TIM1	TIM0		
Access																	RW	RW	RW		RW	RW					RW	RW	RW	RW		

位/位域	标记	功能描述	读写
31:14	Reserved	-	-
13	WWDT	SWD 调试界面下 WWDT 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
12	LPTIM1	SWD 调试界面下 LpTimer1 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
11	TIM3	SWD 调试界面下计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
10	Reserved	-	-
9	RTC	SWD 调试界面下 RTC 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
8	IWDT	SWD 调试界面下 IWDT 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
7:4	Reserved	-	-
3	LPTIM0	SWD 调试界面下 LPTIM0 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
2	TIM2	SWD 调试界面下 Timer2 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
1	TIM1	SWD 调试界面下 Timer1 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW
0	TIM0	SWD 调试界面下 Timer0 计数功能配置 ● 0b0: 正常计数 ● 0b1: 暂停计数	RW

5 中断控制器（NVIC）

5.1 简介

Cortex-M0+处理器内置了嵌套向量中断控制器（NVIC），支持最多 32 个中断请求（IRQ）输入，以及 1 个不可屏蔽中断（NMI）输入（在本产品系统中未使用）。同时，处理器也支持多个内部异常。

每个异常源都有一个单独的异常编号，每种异常类型都有对应的优先级，有些异常的优先级是固定的，而有些则是可编程的，详见下表。

表 5-1 Cortex-M0+处理器中断一览表

异常编号	异常类型	优先级	描述
1	复位	-3（最高）	复位
2	NMI	-2	不可屏蔽中断（在本产品系统中未使用）
3	硬件错误	-1	错误处理异常
4-10	保留	-	-
11	SVC	可编程	通过 SVC 指令调用管理程序
12-13	保留	-	-
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick 定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
...	...	...	...
47	中断#31	可编程	外部中断#31

本章节只对处理器的 32 个外部中断请求（中断#0~中断#31）进行介绍，处理器内部异常可参考内核相关说明指南等文档。同时，本章节只涉及处理器内核中 NVIC 的中断处理机制，不涉及外设模块自身的中断产生机制。

5.2 功能说明

5.2.1 中断优先级

每一个外部中断都对应一个优先级寄存器，每个优先级都是 2 位宽，并且使用中断优先级寄存器的最高两位，每个寄存器占 1 个字节（8 位）。在这个设定下，可以使用的优先级为 0x00（最高）、0x40、0x80 和 0xC0（最低）。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用		未使用，读出为 0					

当处理器运行中遇到多个中断时：

- 如果处理器已经在运行中断处理，而新中断的优先级较大，此时会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。
- 如果处理器正在运行的中断处理的优先级大于或者等于新的中断，那么新的中断会等待并且进入挂起状态。等到运行的中断执行完毕后，继续执行新的中断。
- 如果两个中断同时发生，并且优先级相同，此时中断编号较小的中断会先执行。例如，如果中断#0 和中断#1 使能且具有相同的优先级，同时被触发时，中断#0 先执行。

5.2.2 中断向量表

当 Cortex-M0+处理器要处理中断服务请求时，首先需要确定异常处理的起始地址，所需的信息叫做向量表，如下图所示。向量表存储在存储器空间的开始位置，包含了系统中可用异常（中断）的异常（中断）向量，以及主栈指针（MSP）的初始值。

存储器地址		异常编号
0x000000BC	中断#31向量	47
...	...	...
0x0000004C	中断#3向量	19
0x00000048	中断#2向量	18
0x00000044	中断#1向量	17
0x00000040	中断#0向量	16
0x0000003C	SysTick向量	15
0x00000038	PendSV向量	14
0x00000034	保留	13
0x00000030	保留	12
0x0000002C	SVC向量	11
0x00000028	保留	10
0x00000024	保留	9
0x00000020	保留	8
0x0000001C	保留	7
0x00000018	保留	6
0x00000014	保留	5
0x00000010	保留	4
0x0000000C	硬件错误异常	3
0x00000008	NMI向量	2
0x00000004	复位向量	1
0x00000000	MSP初始值	0

图 5-1 中断向量表

中断向量的存储顺序同中断编号一致，由于每个向量都是 1 个字（4 字节），中断向量的地址为中断编号乘 4，每个中断向量都是中断处理的起始地址。

5.2.3 中断输入和挂起行为

Cortex-M0+处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到

有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。处理过程如下图所示：

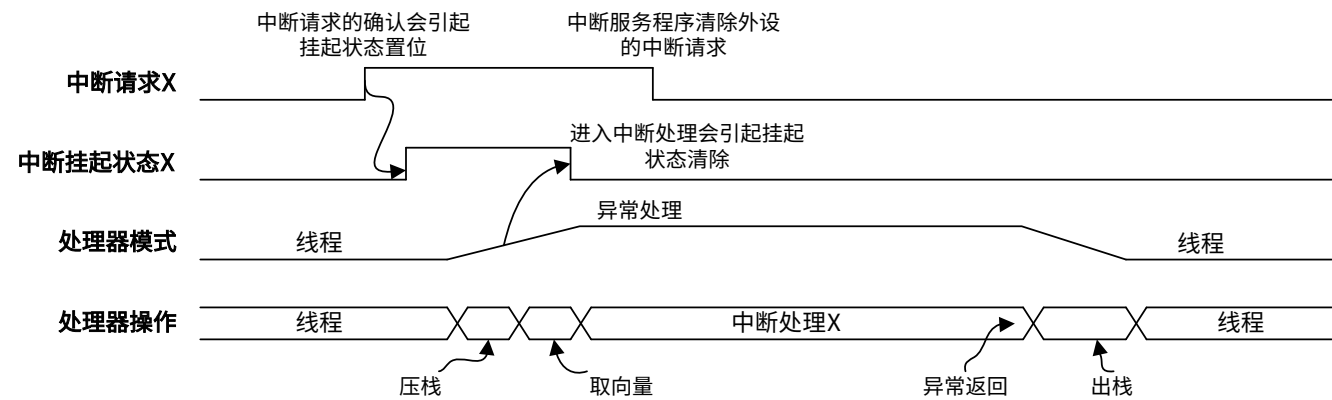


图 5-2 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除，处理器会忽略掉本次请求，不会执行中断处理。在设置外设时，可以通过写 NVIC_ICPR 寄存器来清除中断挂起状态，将外设在被设置之前产生的中断请求清除。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。处理过程如下图所示：

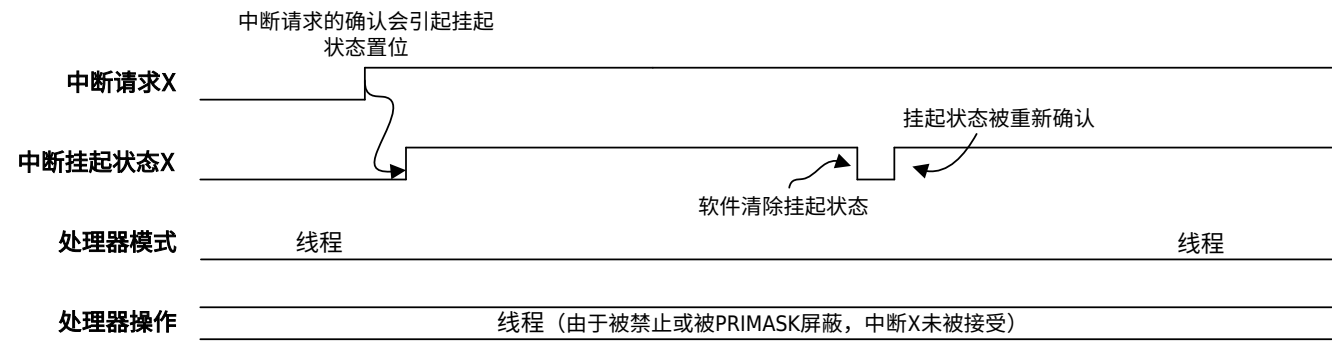


图 5-3 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。处理过程如下图所示：

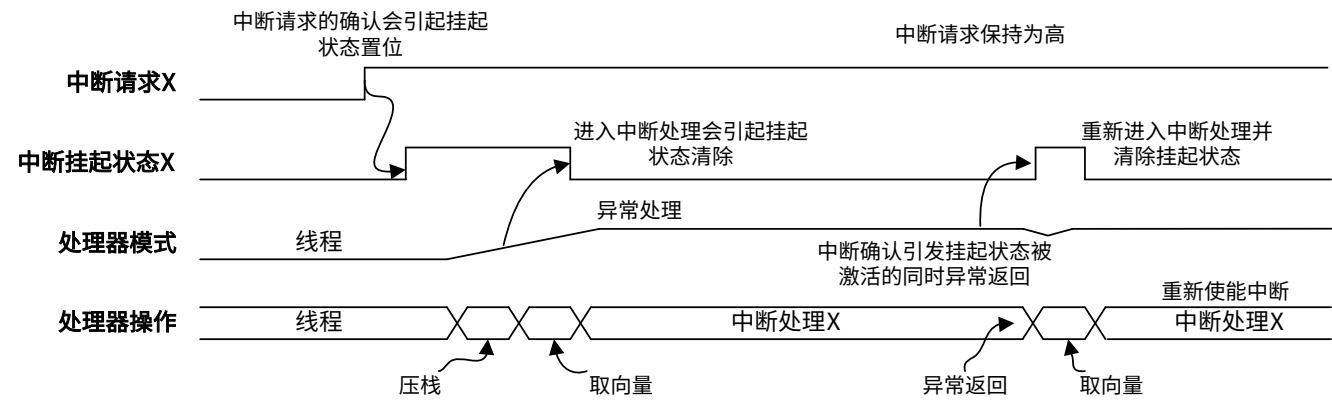


图 5-4 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在中断服务程序执行的过程中产生外设中断请求，该请求会被当作新的中断请求，并且在本次中断退出后，还会引起中断服务程序的再次执行。处理过程如下图所示：

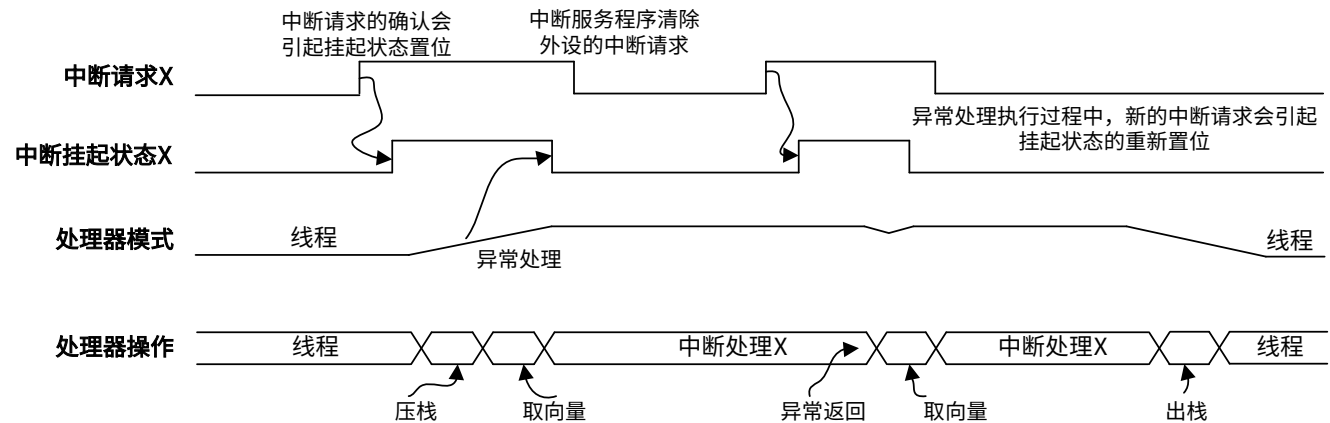


图 5-5 中断处理中产生的中断挂起也可以被确认

5.2.4 中断等待

通常情况下, NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理器时钟周期开始, 一直到中断处理开始执行结束。计算中断等待需要具备以下前提:

- 该中断使能并且没有被 PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态, 在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输, 如果存储器系统需要等待, 那么发生总线传输时产生的等待状态则可能使得中断延迟。

下面的几种情况可能会导致不同的中断等待:

- 中断的末尾连锁, 如果中断返回时产生了另外一个中断请求, 处理器就会跳过出栈和压栈过程, 这样就减少了中断等待时间。
- 延迟到达, 如果中断发生时, 另外一个低优先级的中断正在进行压栈处理, 由于延迟到达机制的存在, 高优先级的中断会首先执行, 这样也会导致高优先级中断的等待时间减小。

5.2.5 中断源

因为 Cortex-M0+处理器的 NVIC 支持最多 32 个外部中断, 而在本系统中外部中断源大于 32 个, 部分外部中断复用在同一个 NVIC 中断输入上, 其中 NMI (不可屏蔽中断) 并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示:

NVIC 中断输入	外部中断源	Active 模式	Sleep 模式	DeepSleep 模式
中断#0	PORTA	✓	✓	✓
中断#1	PORTB	✓	✓	✓
中断#2	PORTC	✓	✓	✓
中断#3	PORTD	✓	✓	✓
中断#4	DMAC	✓	✓	×
中断#5	TIM3	✓	✓	×
中断#6	USART0	✓	✓	×
中断#7	USART1	✓	✓	×
中断#8	LPUART0	✓	✓	✓
中断#9	LPUART1	✓	✓	✓

NVIC 中断输入	外部中断源	Active 模式	Sleep 模式	DeepSleep 模式
中断#10	SPI0	✓	✓	×
中断#11	SPI1	✓	✓	×
中断#12	I2C0	✓	✓	×
中断#13	I2C1	✓	✓	×
中断#14	TIM0	✓	✓	×
中断#15	TIM1	✓	✓	×
中断#16	TIM2	✓	✓	×
中断#17	LPTIM0/1	✓	✓	✓
中断#18	HSI2C	✓	✓	✓ ⁽²⁾
中断#19	LPUART2	✓	✓	✓
中断#20	-	-	-	-
中断#21	WWDT	✓	✓	×
中断#22	IWDT	✓	✓	✓
中断#23	RTC	✓	✓	✓
中断#24	ADC	✓	✓	×
中断#25	-	-	-	-
中断#26	VC0	✓	✓	✓
中断#27	VC1	✓	✓	✓
中断#28	LVD	✓	✓	✓
中断#29	LCD	✓	✓	✓
中断#30	FLASH/RAM	✓	✓	×
中断#31	CTRIM/CLKDET	✓	✓	✓

**注意**

1. 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。
2. 只有从机模式支持 DeepSleep 唤醒。

5.2.6 中断结构图

本系统的中断结构框如下图所示，需要注意：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ 有多于 1 个外设中断输入复用，必须分别读取这些外设的中断标志位来判定是哪个外设的中断。

- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 NVIC_ISER 置位与否，中断挂起寄存器 NVIC_ISPR 都会被置位，表示相应的外设中断源有中断产生。
- 只有中断使能寄存器 NVIC_ISER 置位，相应的中断 IRQ 才会给处理器相应，执行相应的中断程序。
- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 NVIC_ISPR 由硬件自动清除。
- 中断优先级寄存器 NVIC_IPR0-NVIC_IPR7 设置了 32 个中断源的优先级，0b00 优先级最高，0b11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

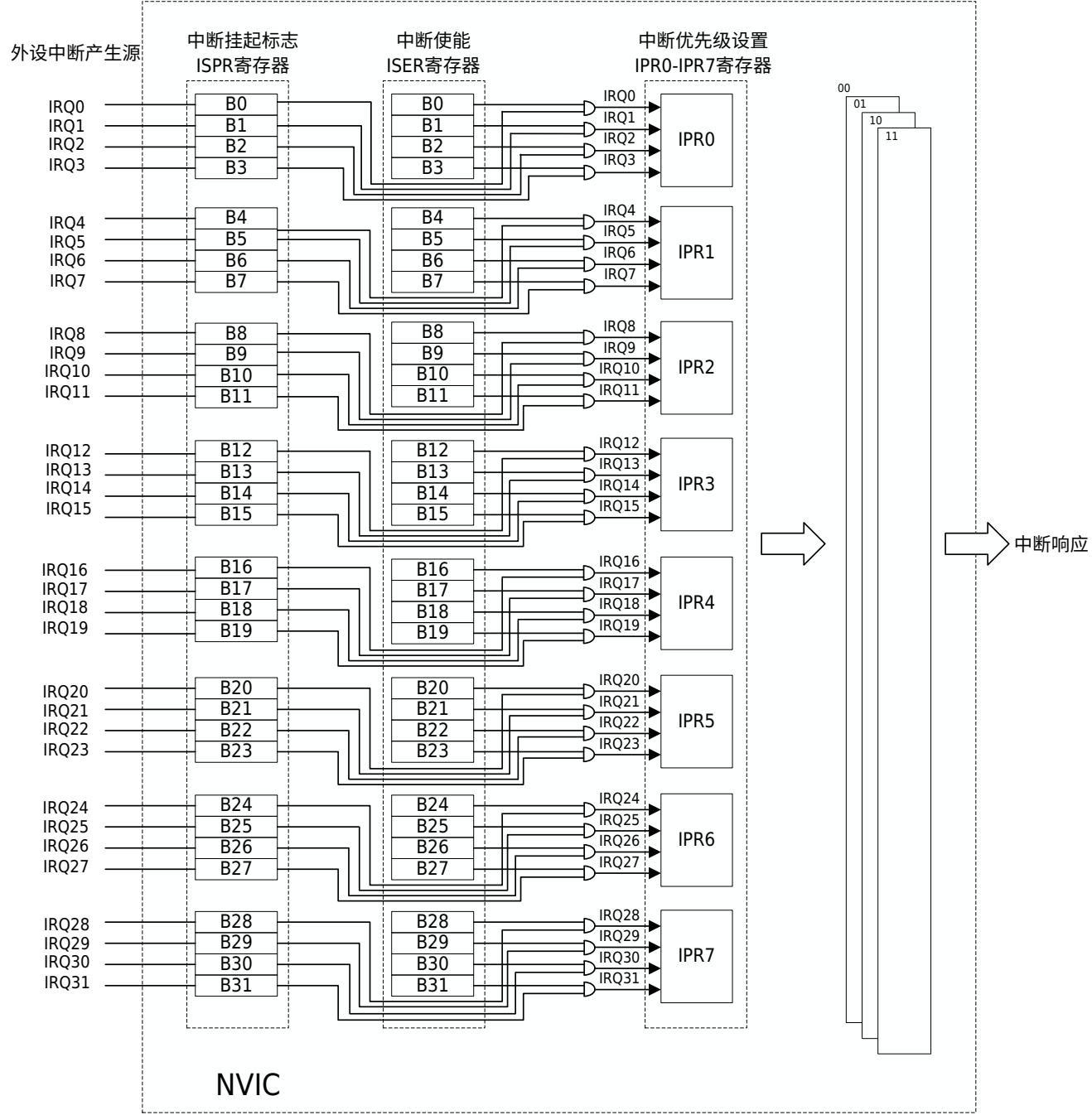


图 5-6 中断结构图

5.3 操作示例

5.3.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考相应外设模块的章节描述。

5.3.2 NVIC 中断使能和清除使能

Cortex-M0+处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零使能位，共计 32 位的中断使能寄存器 NVIC_ISER 和 32 位的清零使能寄存器 NVIC_ICER。如果需要使能某一个中断，则对 NVIC_ISER 寄存器的相应位置 1。如果需要清零某一个中断，则对 NVIC_ICER 寄存器的相应位置 1。此处提及的中断使能和清除使能操作仅仅针对处理器 NVIC，每个外设的中断生成与否，是由外设的中断控制寄存器决定，与 NVIC_ISER 和 NVIC_ICER 无关。

5.3.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持合法。

当处理器开始进入中断处理，硬件会自动引起挂起状态的清除。

可以通过操作中断设置挂起 NVIC_ISPR 和中断清除挂起 NVIC_ICPR 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

5.3.4 NVIC 中断优先级

设置 NVIC_IPR0-NVIC_IPR7 寄存器决定 IRQ0-IRQ31 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+处理器支持。

5.3.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 PRIMASK 实现。特殊寄存器 PRIMASK 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI（本系统不支持）和硬件错误异常处于使能。实际上，当 PRIMASK 设置为 1 后，处理器的当前优先级就变成了 0（可编程的最高优先级）。

可以通过多种方法编程 PRIMASK 寄存器，使用汇编语言，可以利用 MSR 指令设置和清除 PRIMASK 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 PRIMASK。

```
void_enable_irq(void); //清除 PRIMASK
void_disable_irq(void); //设置 PRIMASK
```

5.4 寄存器

5.4.1 寄存器总表

基地址：0xE000E000

表 5-4 中断控制器 (NVIC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x100	NVIC_ISER	中断使能设置寄存器	32
0x180	NVIC_ICER	中断使能清除寄存器	32
0x200	NVIC_ISPR	中断挂起状态设置寄存器	32
0x280	NVIC_ICPR	中断挂起状态清除寄存器	32
0x400	NVIC_IPR0	中断#0-中断#3 优先级寄存器	32
0x404	NVIC_IPR1	中断#4-中断#7 优先级寄存器	32
0x408	NVIC_IPR2	中断#8-中断#11 优先级寄存器	32
0x40C	NVIC_IPR3	中断#12-中断#15 优先级寄存器	32
0x410	NVIC_IPR4	中断#16-中断#19 优先级寄存器	32
0x414	NVIC_IPR5	中断#20-中断#23 优先级寄存器	32
0x418	NVIC_IPR6	中断#24-中断#27 优先级寄存器	32
0x41C	NVIC_IPR7	中断#28-中断#31 优先级寄存器	32
-	PRIMASK	中断屏蔽特殊寄存器	32

5.4.2 中断使能设置寄存器 (NVIC_I SER)

Offset	Bit Position																															
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RW SETENA																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	SETENA	中断使能设置标记 设置使能中断#0 到中断#31 ● 位 0: IRQ0 ● 位 1: IRQ1 ● 位 2: IRQ2 ● ... ● 位 31: IRQ31  说明 每一位写 1 置位，写 0 无效。	RW

5.4.3 中断使能清除寄存器 (NVIC_I CER)

Offset	Bit Position																															
0x180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RW CLRENA																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	CLRENA	中断清除使能 清除使能中断#0 到中断#31 ● 位 0: IRQ0 ● 位 1: IRQ1 ● 位 2: IRQ2 ● ... ● 位 31: IRQ31  说明 每一位写 1 清零，写 0 无效。	RW

5.4.4 中断挂起状态设置寄存器 (NVIC_ISPR)

Offset	Bit Position																															
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SETPEND																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	SETPEND	中断挂起状态设置标记 设置中断#0 到中断#31 的挂起状态 ● 位 0: IRQ0 ● 位 1: IRQ1 ● 位 2: IRQ2 ● ... ● 位 31: IRQ31  说明 每一位写 1 置位, 写 0 无效。	RW

5.4.5 中断挂起状态清除寄存器 (NVIC_ICPR)

Offset	Bit Position																															
0x280	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	CLRPEND																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	CLRPEND	中断挂起状态清除 清除中断#0 到中断#31 的挂起状态 ● 位 0: IRQ0 ● 位 1: IRQ1 ● 位 2: IRQ2 ● ... ● 位 31: IRQ31  说明 每一位写 1 清零, 写 0 无效。	RW

5.4.6 中断优先级寄存器 (NVIC_IPR0)

Offset	Bit Position																															
0x400	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR0																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR0	中断#0 到中断#3 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#3 的优先级 ● [23:22]：中断#2 的优先级 ● [15:14]：中断#1 的优先级 ● [7:6]：中断#0 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.7 中断优先级寄存器 (NVIC_IPR1)

Offset	Bit Position																															
0x404	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR1																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR1	中断#4 到中断#7 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#7 的优先级 ● [23:22]：中断#6 的优先级 ● [15:14]：中断#5 的优先级 ● [7:6]：中断#4 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.8 中断优先级寄存器 (NVIC_IPR2)

Offset	Bit Position																															
0x408	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR2																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR2	中断#8 到中断#11 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#11 的优先级 ● [23:22]：中断#10 的优先级 ● [15:14]：中断#9 的优先级 ● [7:6]：中断#8 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.9 中断优先级寄存器 (NVIC_IPR3)

Offset	Bit Position																															
0x40C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR3																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR3	中断#12 到中断#15 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#15 的优先级 ● [23:22]：中断#14 的优先级 ● [15:14]：中断#13 的优先级 ● [7:6]：中断#12 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.10 中断优先级寄存器 (NVIC_IPR4)

Offset	Bit Position																															
0x410	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR4																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR4	中断#16 到中断#19 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#19 的优先级 ● [23:22]：中断#18 的优先级 ● [15:14]：中断#17 的优先级 ● [7:6]：中断#16 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.11 中断优先级寄存器 (NVIC_IPR5)

Offset	Bit Position																															
0x414	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR5																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR5	中断#20 到中断#23 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#23 的优先级 ● [23:22]：中断#22 的优先级 ● [15:14]：中断#21 的优先级 ● [7:6]：中断#20 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.12 中断优先级寄存器 (NVIC_IPR6)

Offset	Bit Position																															
0x418	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR6																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR6	中断#24 到中断#27 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#27 的优先级 ● [23:22]：中断#26 的优先级 ● [15:14]：中断#25 的优先级 ● [7:6]：中断#24 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.13 中断优先级寄存器 (NVIC_IPR7)

Offset	Bit Position																															
0x41C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	IPR7																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	IPR7	中断#28 到中断#31 的优先级 各中断优先级对应的位如下： ● [31:30]：中断#31 的优先级 ● [23:22]：中断#30 的优先级 ● [15:14]：中断#29 的优先级 ● [7:6]：中断#28 的优先级 上述位设置 0b00 时优先级最高，0b11 时优先级最低。由于其他位没有使用，对其进行设置没有意义，读出始终为 0。	RW

5.4.14 中断屏蔽特殊寄存器 (PRIMASK)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															PRIMASK
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	PRIMASK	中断屏蔽标识 一般用于处理对时间敏感的应用的场景。该特殊寄存器可以通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。 0b0：所有的中断和异常处于允许状态 0b1：只有 NMI 和硬件错误异常处于使能	RW

6 Flash 控制器 (FLASH)

6.1 简介

本系统包含一块 128KB 容量的 Flash 存储器，共划分为 256 个页 (Sector)，每个页 (Sector) 的容量为 512 字节 (Byte)。

本控制器支持对 Flash 存储器的擦除、编程以及读操作，还支持对 Flash 存储器擦写保护，以及控制寄存器的写保护。

本控制器支持对 Flash 的 byte (8 位)、half-word (16 位) 和 word (32 位) 三种位宽的数据读写操作。注意，Byte 操作的目标地址必须按字节对齐，half-word 操作的目标地址必须按半字对齐 (地址最低位为 0b0)，word 操作的地址必须按字对齐 (地址最低两位为 0b00)。如果目标地址没有按照位宽对齐，则该操作无效，并且 CPU 会进入 Hard Fault 出错中断。

6.2 功能说明

6.2.1 容量划分

容量划分的地址和序号对应关系如下：

表 6-1 Flash 容量划分表

地址	序号	...	地址	序号
0x0E00 - 0x0FFF	Sector7	...	0x1FE00 - 0x1FFFF	Sector255
0x0C00 - 0x0DFF	Sector6	...	0x1FC00 - 0x1FDFF	Sector254
0x0A00 - 0x0BFF	Sector5	...	0x1FA00 - 0x1FBFF	Sector253
0x0800 - 0x09FF	Sector4	...	0x1F800 - 0x1F9FF	Sector252
0x0600 - 0x07FF	Sector3	...	0x1F600 - 0x1F7FF	Sector251
0x0400 - 0x05FF	Sector2	...	0x1F400 - 0x1F5FF	Sector250
0x0200 - 0x03FF	Sector1	...	0x1F200 - 0x1F3FF	Sector249
0x0000 - 0x01FF	Sector0	...	0x1F000 - 0x1F1FF	Sector248

6.2.2 读等待周期

本设备内置的 Flash 支持的最快取指频率为 24MHz。当 HCLK 频率超过 24MHz 小于等于 48MHz 时，必须为 Flash 的读取时间插入等待周期，即设置 FLASH_CR.WAIT 为 1。当插入等待周期时，Flash 每两个周期才会完成一次读取操作。

6.2.3 Flash 安全保护

页面擦写保护

整个 128KB Flash 存储器被划分为 256 个页，每 4 个页共用一个擦写保护位。当页被保护时，对该页进行的擦写操作均无效并产生报警标志位和中断信号。当 Flash 存储器中的任意页被保护时，对该 Flash 的全片擦写无效，并产生报警标志位和中断信号。

PC 地址擦写保护

CPU 在 Flash 中运行程序时，如果当前 PC 指针正好落在待擦写的页地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。

寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

- 需要通过写序列方式才能更改的寄存器
FLASH_CR、FLASH_SLOCK0、FLASH_SLOCK1
- 不需要通过写序列方式即可更改的寄存器
FLASH_ICLR、FLASH_BYPASS

通过写序列方式修改寄存器值的具体操作步骤如下所示：

- 步骤 1. 向 FLASH_BYPASS 寄存器写入 0x5A5A。
- 步骤 2. 向 FLASH_BYPASS 寄存器写入 0x5A55。
- 步骤 3. 对待修改的寄存器写入目标值。
- 步骤 4. 验证待修改的寄存器当前值是否与目标相同，如不同则跳转到第一步。
- 步骤 5. 执行其他操作。

 **注意**
写 0x5A5A、0x5A55、写目标寄存器，这三步写操作之间不可插入任何写操作（如：Flash、RAM、REG 等），否则无法改写目标寄存器的数值。如改写失败，需要重新进行这三步操作。

数据读出保护

本芯片 Flash 支持 4 级读保护，当前保护等级可通过 FLASH_LOCK_ST 寄存器读出。使用编程工具对芯片进行编程时，可配置 Flash 的读保护等级。

芯片在各保护等级下的功能如下表所示：

保护等级	芯片功能
Level0	可以通过 CPU、DMA、ISP 及 SWD 对 Flash 进行读写擦
Level1	数据不可从 SWD 读出 ● SWD 插入前，可通过 CPU 或 DMA 对 Flash 进行读写擦； ● SWD 插入后，Flash 拒绝任何访问。 可以通过 ISP 及 SWD 降级到 Level0，降级后 Flash 内容全为 FF
Level2	数据不可读出 可通过 CPU 或 DMA 对 Flash 进行读写擦 可以通过 ISP 降级到 Level0，降级后 Flash 内容全为 FF SWD 接口被禁止
Level3	数据不可读出 可通过 CPU 或 DMA 对 Flash 进行读写擦 ISP 及 SWD 接口均不可读，即芯片只能进行一次编程

6.3 操作示例

 **注意**

- 对 FLASH 进行擦除或写操作前，须将 FLASH_CR.RO 设置为 0，将 FLASH 配置为可读，可编程，可擦除。
- 当 FLASH 擦除或写操作完成后，建议将 FLASH_CR.RO 设置为 1，将 FLASH 配置为只读，不可编程或擦除。

6.3.1 页擦除 (Sector Erase)

背景信息

页擦除每次可以擦除用户指定的一个页 (Sector)。擦除操作完成后，页 (Sector) 内的数据均为 0xFF。如果该擦除操作是从 Flash 内执行，则 CPU 会停止取指，硬件自动等待该操作完成 (FLASH_CR.BUSY 变为 0)；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成 (FLASH_CR.BUSY 变为 0)。

操作步骤

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 为 2，设置 Flash 操作模式为 Sector 擦除。
- 步骤 3. 检查 FLASH_CR.OP 是否为 2，如不为 2 则跳转到步骤 1。
- 步骤 4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 5. 设置 FLASH_SLOCKx 相应的比特为 1，去除该 Sector 的擦写保护。
- 步骤 6. 检查 FLASH_SLOCKx 相应的比特是否为 1，如不为 1 则跳转到步骤 4。
- 步骤 7. 对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。

```
*((unsigned char *)0x00000200) = 0x00
```

- 步骤 8. 等待 FLASH_CR.BUSY 变为 0，Sector 擦除操作完成。
- 步骤 9. 如需擦除其它 Sector，重复步骤 4~步骤 8。

6.3.2 全片擦除 (Chip Erase)

背景信息

全片擦除可以一次性擦除全部的页 (Sector)。擦除操作完成后，所有页 (Sector) 内的数据均为 0xFF。如果该擦除操作是从 Flash 内执行，则该操作会被禁止。因为该操作会擦除当前 PC 所在的程序段。如发生这种情况，FLASH_IFR.PC 会被置起；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成 (FLASH_CR.BUSY 变为 0)。

操作步骤

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 为 3，设置 Flash 操作模式为 Chip 擦除。
- 步骤 3. 检查 FLASH_CR.OP 是否为 3，如不为 3 则跳转到步骤 1。
- 步骤 4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 5. 设置 FLASH_SLOCK0 为 0xFFFFFFFF，去除 Sector0~127 的擦写保护。
- 步骤 6. 检查 FLASH_SLOCK0 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到步骤 4。
- 步骤 7. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 8. 设置 FLASH_SLOCK1 为 0xFFFFFFFF，去除 Sector128~255 的擦写保护。
- 步骤 9. 检查 FLASH_SLOCK1 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到步骤 7。
- 步骤 10. 对待擦除的 Chip 内的任意地址进行写操作，触发 Chip 擦除。

```
*((unsigned char *)0x00000000) = 0x00
```

- 步骤 11. 等待 FLASH_CR.BUSY 变为 0，Chip 擦除操作完成。

6.3.3 写操作 (Program)

背景信息

写操作只能将 Flash 内的比特数据由 1 写成 0，故写入数据前应确保待写入的地址内的数据为 0xFF。支持 3 种写入方式：Byte (8bits)、Half-word (16bits)、Word (32bits)，写入的数据以小端模式存放在 Flash 中，即低地址存放数据的低字节。如果写操作是从 Flash 内执行，则 CPU 会停止取指，硬件自动等

待该操作完成 (FLASH_CR.BUSY 变为 0) ; 如果该写操作是从 RAM 内执行, 则 CPU 不会停止取指, 用户软件应等待该操作完成 (FLASH_CR.BUSY 变为 0) 。

操作步骤

● Byte 写操作

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 为 1, 设置 Flash 操作模式为写入。
- 步骤 3. 检查 FLASH_CR.OP 是否为 1, 如不为 1 则跳转到步骤 1。
- 步骤 4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 5. 设置 FLASH_SLOCK 相应的比特为 1, 去除擦写保护。
- 步骤 6. 检查 FLASH_SLOCK 相应的比特是否为 1, 如不为 1 则跳转到步骤 4。
- 步骤 7. 对待写入的目标地址进行 Byte 写操作, 触发写入操作。

```
*((unsigned char *)0x00001231) = 0x5A
```

- 步骤 8. 等待 FLASH_CR.BUSY 变为 0, 写入操作完成。
- 步骤 9. 如需写 Byte 到其它地址, 重复步骤 7~步骤 8。

● Half-word 写操作

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 为 1, 设置 Flash 操作模式为写入。
- 步骤 3. 检查 FLASH_CR.OP 是否为 1, 如不为 1 则跳转到步骤 1。
- 步骤 4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 5. 设置 FLASH_SLOCK 相应的比特为 1, 去除擦写保护。
- 步骤 6. 检查 FLASH_SLOCK 相应的比特是否为 1, 如不为 1 则跳转到步骤 4。
- 步骤 7. 对待写入的目标地址进行 Half-word 写操作, 触发写入操作。

```
*((unsigned short int *)0x00001232) = 0xABCD
```

- 步骤 8. 等待 FLASH_CR.BUSY 变为 0, 写入操作完成。
- 步骤 9. 如需写 Half-word 到其它地址, 重复步骤 7~步骤 8。

● Word 写操作

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 为 1, 设置 Flash 操作模式为写入。
- 步骤 3. 检查 FLASH_CR.OP 是否为 1, 如不为 1 则跳转到步骤 1。
- 步骤 4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55, 使能寄存器改写。
- 步骤 5. 设置 FLASH_SLOCK 相应的比特为 1, 去除擦写保护。
- 步骤 6. 检查 FLASH_SLOCK 相应的比特是否为 1, 如不为 1 则跳转到步骤 4。
- 步骤 7. 对待写入的目标地址进行 Word 写操作, 触发写入操作。

```
*((unsigned int *)0x00001234) = 0x55667788
```

- 步骤 8. 等待 FLASH_CR.BUSY 变为 0, 写入操作完成。
- 步骤 9. 如需写 Word 到其它地址, 重复步骤 7~步骤 8。

6.3.4 连续写操作 (Cont-Program)

背景信息

写操作只能将 Flash 内的比特数据由 1 写成 0, 故写入数据前应确保待写入的地址内的数据为 0xFF。支持 3 种写入方式: Byte (8bits)、Half-word (16bits)、Word (32bits), 写入的数据以小端模式存放在

Flash 中，即低地址存放数据的低字节。连续写操作只能从 RAM 内执行，并且禁止对 Flash 做“读操作”，用户软件应等待该操作完成（FLASH_CR.BUSY 变为 0）。

连续编程时建议 HCLK 不低于 2MHz。

操作步骤

● Byte 写操作

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 值为 0。
- 步骤 3. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 4. 配置 FLASH_CR.CONTP 为 1，设置 Flash 操作模式为连续写入。
- 步骤 5. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 0 和 1，如不为则跳转到步骤 1。
- 步骤 6. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 7. 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。
- 步骤 8. 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到步骤 6。
- 步骤 9. 对待写入的目标地址进行 Byte 写操作，触发写入操作。

```
*((unsigned char *)0x00001232) = 0xAB
```

- 步骤 10. 等待 FLASH_CR.BUSY 变为 0，写入操作完成。
- 步骤 11. 如需写 Byte 到其它地址，重复步骤 9~步骤 10。
- 步骤 12. 全部数据写完后执行下一步。
- 步骤 13. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 14. 配置 FLASH_CR.OP 为 1。
- 步骤 15. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 16. 配置 FLASH_CR.CONTP 为 0。
- 步骤 17. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 1 和 0，如不为则跳转到步骤 13。
- 步骤 18. 等待 FLASH_CR.BUSY 变为 0。
- 步骤 19. 后续 FLASH_CR.OP 可以改为其他模式的值完成其他功能操作。

● Half-word 写操作

- 步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 2. 配置 FLASH_CR.OP 值为 0。
- 步骤 3. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 4. 配置 FLASH_CR.CONTP 为 1，设置 Flash 操作模式为连续写入。
- 步骤 5. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 0 和 1，如不为则跳转到步骤 1。
- 步骤 6. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 7. 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。
- 步骤 8. 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到步骤 6。
- 步骤 9. 对待写入的目标地址进行 Half-word 写操作，触发写入操作。

```
*((unsigned short int *)0x00001232) = 0xABCD
```

- 步骤 10. 等待 FLASH_CR.BUSY 变为 0，写入操作完成。
- 步骤 11. 如需写 Half-word 到其它地址，重复步骤 9~步骤 10。
- 步骤 12. 全部数据写完后执行下一步。
- 步骤 13. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- 步骤 14. 配置 FLASH_CR.OP 为 1。

步骤 15. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 16. 配置 FLASH_CR.CONTP 为 0。

步骤 17. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 1 和 0，如不为则跳转到步骤 13。

步骤 18. 等待 FLASH_CR.BUSY 变为 0。

步骤 19. 后续 FLASH_CR.OP 可以改为其他模式的值完成其他功能操作。

● Word 写操作

步骤 1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 2. 配置 FLASH_CR.OP 值为 0。

步骤 3. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 4. 配置 FLASH_CR.CONTP 为 1，设置 Flash 操作模式为连续写入。

步骤 5. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 0 和 1，如不为则跳转到步骤 1。

步骤 6. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 7. 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

步骤 8. 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到步骤 6。

步骤 9. 对待写入的目标地址进行 Word 写操作，触发写入操作。

```
*((unsigned int *)0x00001234) = 0xABCDABCD
```

步骤 10. 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

步骤 11. 如需写 Word 到其它地址，重复步骤 9~步骤 10。

步骤 12. 全部数据写完后执行下一步。

步骤 13. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 14. 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

步骤 15. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0x5A55，使能寄存器改写。

步骤 16. 配置 FLASH_CR.CONTP 为 0。

步骤 17. 检查 FLASH_CR.OP 和 FLASH_CR.CONTP 是否为 1 和 0，如不为则跳转到步骤 13。

步骤 18. 等待 FLASH_CR.BUSY 变为 0。

步骤 19. 后续 FLASH_CR.OP 可以改为其他模式的值完成其他功能操作。

6.3.5 读操作 (Read)

背景信息

支持 3 种读取方式：Byte (8bits)、Half-word (16bits)、Word (32bits)，读出的数据为小端模式，即低地址存放数据的低字节。读操作无需操作步骤，任何时刻都可以读出 FLASH 内的数据。

操作步骤

● Byte 读操作

```
temp = *((unsigned char *)0x00001231)
```

● Half-word 读操作

```
temp = *((unsigned short int *)0x00001232)
```

● Word 读操作

```
temp = *((unsigned int *)0x00001234)
```

6.4 寄存器

6.4.1 寄存器总表

基地址：0x40020000

表 6-3 Flash 控制器 (FLASH) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x20	FLASH_CR	控制寄存器	32
0x24	FLASH_IFR	中断标志寄存器	32
0x28	FLASH_ICLR	中断标志清除寄存器	32
0x2C	FLASH_BYPASS	0x5A5A-0xA5A5 Bypass 序列寄存器	32
0x30	FLASH_SLOCK0	Sector0-127 擦写保护寄存器	32
0x34	FLASH_SLOCK1	Sector128-255 擦写保护寄存器	32
0x6C	FLASH_LOCK_ST	保护状态寄存器	32

6.4.2 CR 寄存器 (FLASH_CR)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0b00000000 00000000 0000X010 00000000																															
Name	Reserved																				RO	CONTP	DPSTB_EN	Reserved	IE	BUSY	WAIT	OP				
Access																					RW	RW	RW		RW	RO	RW	RW	RW			

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11	RO	只读使能位 ● 0b0: 可读, 可编程, 可擦除 ● 0b1: 只读, 不可编程或擦除	RW
10	CONTP	连续编程 ● 0b0: 单次编程 ● 0b1: 连续编程	RW
9	DPSTB_EN	FLASH dpstb 使能 Mask 位 ● 0b0: 当系统进入 DeepSleep 模式, FLASH 不进入低功耗模式 ● 0b1: 当系统进入 DeepSleep 模式, FLASH 进入低功耗模式	RW
8:7	Reserved	-	-
6:5	IE	FLASH 中断使能 IE[6]对应 FLASH 擦写被保护地址中断使能 IE[5]对应 FLASH 擦写 PC 值中断使能 ● 0b0: 不使能 ● 0b1: 使能	RW
4	BUSY	空闲/忙标志位 ● 0b0: 空闲状态 ● 0b1: 忙状态	RO
3:2	WAIT	读 FLASH 周期 ● 0b00: 1 个周期, 0~24MHz ● 0b01: 2 个周期, 24~48MHz ● 0b10: 禁止设置 ● 0b11: 禁止设置	RW
1:0	OP	FLASH 操作 ● 0b00: 读 (read) ● 0b01: 写 (program) ● 0b10: 页擦除 (sector erase) ● 0b11: 全片擦除 (chip erase)	RW

6.4.3 IFR 寄存器 (FLASH_IFR)

Offset	Bit Position																																	
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000010																																	
Name	Reserved																														RO		PROT	
Access																															RO		PC	

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	PROT	FLASH 擦写保护报警中断标志位	RO
0	PC	FLASH 擦写 PC 地址报警中断标志位	RO

6.4.4 ICLR 寄存器 (FLASH_ICLR)

Offset	Bit Position																																	
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x0000000F																																	
Name	Reserved																																WO	PROT
Access																																	WO	PC

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	PROT	清除保护报警中断标志位 写 0 清除，写 1 无效；读出恒为 1	WO
0	PC	清除 PC 地址报警中断标志位 写 0 清除，写 1 无效；读出恒为 1	WO

6.4.5 BYPASS 寄存器 (FLASH_BYPASS)

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																BYSEQ															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	BYSEQ	FLASH 模块保护控制寄存器 对 FLASH_BYPASS 先写 0x5A5A，再写 0xA5A5，启动对于 FLASH 模块寄存器的写操作，只要对 FLASH 模块寄存器进行了写操作，这个保护位自动恢复保护状态。后续再对 FLASH 模块寄存器进行写操作时还需要重复先前的操作。	RW

6.4.6 SLOCK0 寄存器 (FLASH_SLOCK0)

Offset	Bit Position																															
0x30	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000 (外设复位无效)																															
Name	SLOCK																															
Access	RW																															

位/位域	标记	功能描述	读写	
31:0	SLOCK	Sector 擦写保护位	RW	
		● 0b0: 不允许擦写		
		● 0b1: 允许擦写		
		Bit 位		序号
		0		Sector0~3
		1		Sector4~7
		2		Sector8~11
		3		Sector12~15
		...		...
31	Sector124~127			

6.4.7 SLOCK1 寄存器 (FLASH_SLOCK1)

Offset	Bit Position																															
0x34	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000（外设复位无效）																															
Name	SLOCK																															
Access	RW																															

位/位域	标记	功能描述	读写	
31:0	SLOCK	Sector 擦写保护位	RW	
		● 0b0: 不允许擦写		
		● 0b1: 允许擦写		
		Bit 位		序号
		0		Sector128~131
		1		Sector132~135
		2		Sector136~139
		3		Sector140~143
		...		...
31	Sector252~255			

6.4.8 保护状态寄存器 (FLASH_LOCK_ST)

Offset	Bit Position																															
0x6C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															LOCK_ST
Access																																RO

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1:0	LOCK_ST	FLASH 保护状态 ● 0b00: Level0, ISP 可读写, SWD 可读写 ● 0b01: Level1, ISP 可降级, SWD 可降级; 数据不可读出 ● 0b10: Level2, ISP 可降级, SWD 无功能; 数据不可读出 ● 0b11: Level3, ISP 无功能, SWD 无功能; 数据不可读出	RO

7 RAM 控制器 (RAM)

7.1 简介

本系统中包含一块容量为 16KB 的 SRAM，支持字节 (8bits)、半字 (16bits)、字 (32bits) 三种读写操作。可在系统时钟频率下进行读写操作，无须等待周期。此外，本控制器还支持奇偶校验，可对每个字节 (Byte) 的 SRAM 数据进行奇偶校验，并产生奇偶校验出错中断。

7.2 功能说明

7.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示：

表 7-1 RAM 地址映射

地址范围	大小	Memory 类型
0x20000000 - 0x20003FFF	16KByte	SRAM

7.2.2 读写位宽

本控制器支持字节 (8 位)、半字 (16 位)、字 (32 位) 三种位宽的读写操作：

- 字节操作的地址必须按字节对齐
- 半字操作的目标地址必须按半字对齐 (地址最低位为 0b0)
- 字操作的地址必须按字对齐 (地址最低两位为 0b00)

如果读写操作的目标地址没有按照位宽规定对齐，该操作无效，并且系统会产生 Hard Fault 出错中断。

7.2.3 奇偶校验

本控制器支持 SRAM 数据的奇偶校验：

- 对 SRAM 写数据时，对每个字节的数据做奇偶校验，并把 1bit 校验值和 8bits 数据一起存入 SRAM 中。
- 对 SRAM 读数据时，控制器会读取 8bits 数据和 1bit 校验值，并做奇偶校验。

如果校验出错，则置位奇偶校验出错标志位，在中断使能情况下，会产生出错中断。



注意

在奇偶校验使能时，在读取 SRAM 数据前，必须初始化 SRAM，否则可能会误触发奇偶校验报警标志位或者中断。

7.3 寄存器

7.3.1 寄存器总表

基地址：0x40020400

表 7-2 RAM 控制器 (RAM) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	RAM_CR	控制寄存器	32
0x04	RAM_ERRADDR	奇偶校验出错地址寄存器	32
0x08	RAM_IFR	出错中断标志寄存器	32
0x0C	RAM_ICLR	出错中断标志清除寄存器	32

7.3.2 控制寄存器 (RAM_CR)

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000001																																	
Name	Reserved																																IE	Reserved
Access																																	RW	

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	IE	奇偶校验出错报警中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
0	Reserved	-	-

7.3.3 奇偶校验出错地址寄存器 (RAM_ERRADDR)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ERRADDR															
Access																	RO															

位/位域	标记	功能描述	读写
31:14	Reserved	-	-
13:0	ERRADDR	奇偶校验出错 Byte 地址 中断标志位清除后, 该地址同时被清除	RO

7.3.4 出错中断标志寄存器 (RAM_IFR)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															ERR
Access																																RO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	ERR	奇偶校验出错标志位 ● 0b0: 读 RAM 时, 奇偶校验正确 ● 0b1: 读 RAM 时, 奇偶校验出错	RO

7.3.5 出错中断标志清除寄存器 (RAM_ICLR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	Reserved																															ERRCLR
Access																																WO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	ERRCLR	出错中断标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO

8 端口控制器 (GPIO)

8.1 简介

本系列最多有 56 个数字通用输入输出端口。模拟模块 ADC/VC/LVD/LCD 的输入输出信号、各功能模块（如 SPI/USART/I2C/Timer 等）的输入输出信号可以和数字通用输入输出端口复用。

每个端口都可以配置成内部上拉（pull up）/下拉（pull down）的输入，高阻输入（floating input），推挽输出（Push-pull output），开漏输出（Open-drain output），两档驱动能力输出。数字端口被配置成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口输入值寄存器的结果为“0”。

每个数字端口被配置为输入时，都可以提供外部中断，中断类型可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发 4 种，查询 Px_STAT[n]的中断标志位即可知道相应的中断触发端口。另外，每个数字端口的中断都可以把芯片从休眠模式/深度休眠模式唤醒到工作模式。

8.2 主要特性

端口控制器支持以下特性：

- 端口输入值/输出值寄存器支持 FAST IO/AHB 总线读写
- 其他寄存器支持 AHB 总线接口读写
- 模拟功能引脚/数字通用引脚/数字功能引脚复用
- 支持上拉/下拉/两档驱动/开漏输出功能选择
- 支持工作模式/休眠模式/深度休眠模式下中断
- 支持高电平/低电平/上升沿/下降沿触发中断
- 支持位置位、位清零、位置位清零功能
- 部分端口支持输入滤波功能



说明

FAST IO 介绍请参考《ARM Cortex-M0+_IntegrationAndImplementationManual.pdf》。

8.3 功能说明

8.3.1 端口配置功能

每一个端口都可以根据系统需求通过配置寄存器 (PxADS) 把端口配置成模拟端口或数字端口。当 PxADS 为 1 时, 端口配置成模拟端口, 当 PxADS 为 0 时, 端口配置成数字端口。

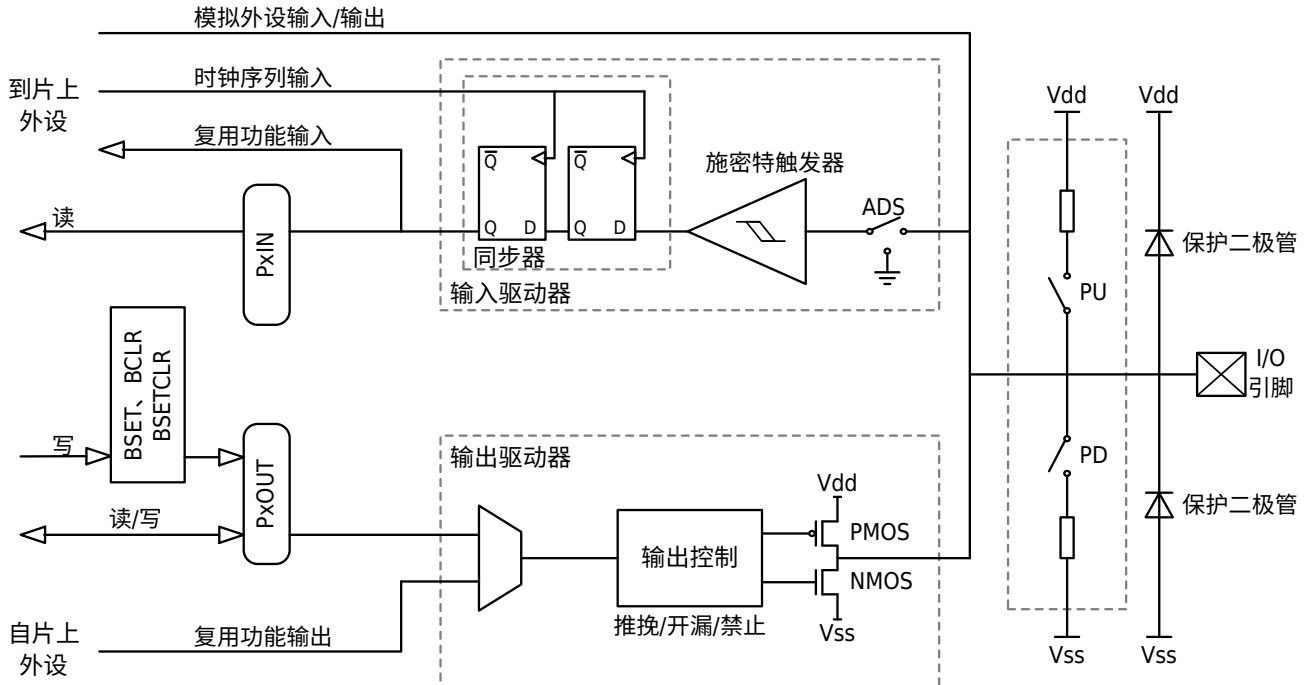


图 8-1 端口电路图



说明

输出/输入数据寄存器 (PxOUT/PxIN) 支持 AHB/FAST IO 总线访问, 其他寄存器支持 AHB 总线访问。

端口配置为数字端口时, 可以接受数字通用输入输出信号 (设置 Pxy_SEL 寄存器为 0), 也可以通过配置寄存器 Pxy_SEL 接受各功能模块 (如 SPI, USART, I2C, Timer 等) 的输入输出信号, 详见端口数字复用功能模块。

还可以配置相应的寄存器实现以下特性:

- 内部上拉 (PxPU) / 下拉 (PxPD)
上拉寄存器 (PxPU)、下拉寄存器 (PxPD) 分别对应端口上拉使能和端口下拉使能。当对应位为 1 时, 设置对应位引脚上拉/下拉使能; 为 0 时, 禁止对应位引脚上拉/下拉。
- 两档驱动输出 (PxDR)
能够通过 PxDR 寄存器改变驱动能力, PxDR 为 1 时为低驱动能力; PxDR 为 0 时为高驱动能力。
- 开漏输出 (PxOD)
通过 PxOD 寄存器设置管脚输出状态。当 PxOD 为‘1’时, 端口开漏输出使能, 为‘0’时, 端口推挽输出使能。开漏引脚不连接外部的上拉电阻时, 只能输出低电平, 如果需要同时具备输出高电平的功能时, 则需要上拉电阻。
- 方向选择 (PxDIR)
用于设置端口引脚的方向。PxDIR 为 0 时端口为输出, PxDIR 为 1 时端口为输入。
- 输入电平状态 (PxIN)
通过读取 PxIN 寄存器能够获得同步后的管脚电平。PxIN 为 1 时为高电平, PxIN 为 0 时为低电平。

- 输出高低电平选择 (PxOUT)

当端口引脚配置为输出时，如果 PxOUT 为 1，端口引脚输出为高电平；如果 PxOUT 为 0，输出低电平。

- 位置位 (PxBSET)、位清零 (PxBCLR)、位置位清零 (PxBSETCLR)

位置位与位清零适用于在不改变其他位的值的情况下，对用户想要改变的位进行设置，得到相应的值。位置位是将对应的值设置为 1，位清零是将对应的值设置为 0。

**说明**

以上特性在配置为模拟端口时无效。

端口的状态和寄存器配置的关系如下表：

表 8-1 端口状态真值表

IO 状态	IO 方向	PxADS	PxDIR	PxOUT	PxIN	PxBSET	PxBCLR	PxBSETCLR	PxPU	PxPD	PxOD	PxDR	Pxy_SEL
模拟	输入/输出	1	W	W	0	W	W	W	W	W	W	W	W
浮空	输入	0	1	W	X	W	W	W	0	0	W	W	0
下拉	输入	0	1	W	0	W	W	W	0	1	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	0	W	W	0
上拉	输入	0	1	W	1	W	W	W	1	1	W	W	0
1	输入	0	1	W	1	W	W	W	W	W	W	W	0
0	输入	0	1	W	0	W	W	W	W	W	W	W	0
1	输出	0	0	1	1	0	0	0	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	0	W	0
1	输出	0	0	W	1	1	0	0	W	W	0	W	0
0	输出	0	0	W	0	0	1	0	W	W	0	W	0
1 (SET)	输出	0	0	W	1 (SET)	0	0	1	W	W	0	W	0
0 (CLR)	输出	0	0	W	0 (CLR)	0	0	1	W	W	0	W	0
0	输出	0	0	0	0	0	0	0	W	W	1	W	0
Z	输出	0	0	1	X	0	0	0	0	0	1	W	0
0	输出	0	0	1	0	0	0	0	0	1	1	W	0
1	输出	0	0	1	1	0	0	0	1	0	1	W	0
1	输出	0	0	1	1	0	0	0	1	1	1	W	0

0 - Logic low, 1 - Logic high, W - Whatever 0 or 1, X - unknow state, Z - high impedance

8.3.2 端口的写入

端口输入值/输出值寄存器 (PxIN/PxOUT) 支持 AHB 总线和 FAST IO 总线的读写 (通过寄存器 GPIO_CTRL2.AHB_SEL 位控制)，对于其他的寄存器则只支持 AHB 总线的读写。对于这两种不同的总线，系统时钟 (HCLK) 对这两种总线的处理周期并不相同。下面两幅图为对于两种总线端口翻转的最快时序：

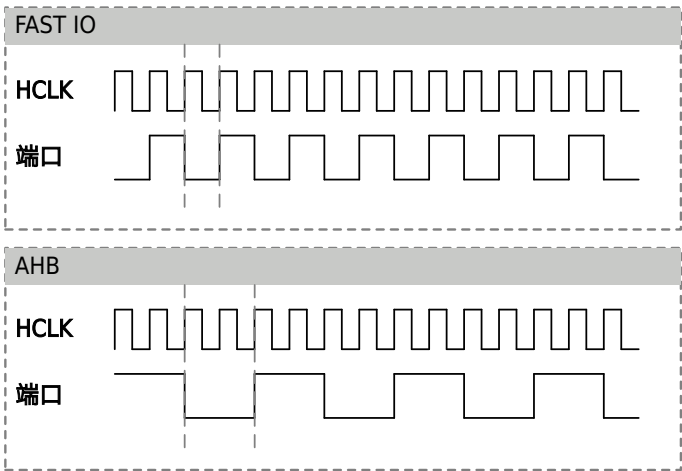


图 8-2 AHB/FASTIO 总线端口随系统时钟的变化

对于 AHB 总线，每两个 HCLK 周期，IO 翻转一次，而对于 FAST IO 总线，每个 HCLK 周期，IO 翻转一次。

8.3.3 端口的读取

每一个端口可以通过读取 PxIN 寄存器来获得端口引脚电平。如端口电路图所示，PxIN 寄存器的各个位与其前面的锁存器组成了一个同步器从而避免了系统时钟状态发生改变时在短时间内引脚电平变化而造成的信号不稳定，但是同时也引入了延迟。读取端口引脚数据的同步图如下：

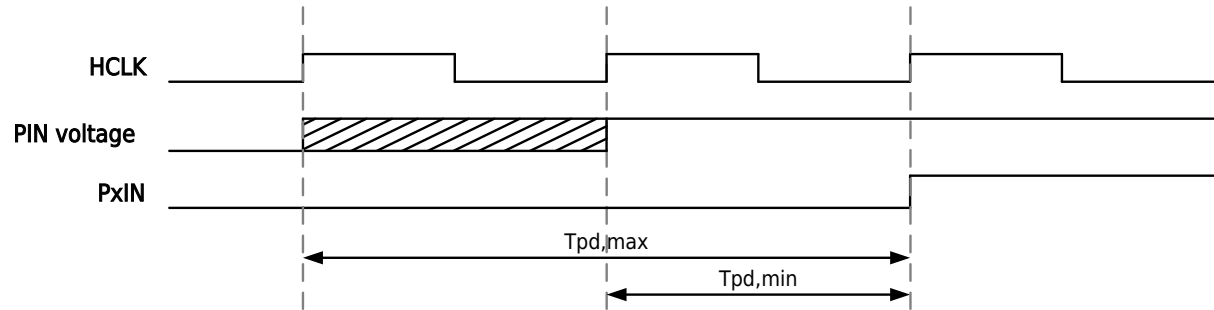


图 8-3 读取端口引脚数据同步图

在系统时钟上升沿之后的时钟时期，管脚电平信号会锁存在内部寄存器，如阴影部分所示，在下一次系统时钟上升沿之后，稳定的管脚电平信号能够被读取。之后再系统时钟上升沿时，数据锁存到 PxIN 寄存器中。信号装换延迟 Tpd 为 1-2 个系统时钟。

 **说明**

未连接引脚的处理：在使用过程中如果有未接引脚，为避免在其他数字输入使能模式时引脚没有确定的电平造成悬空消耗电流，推荐赋予引脚一个确定的电平。

8.3.4 端口数字复用功能

端口数字复用是端口控制器的主要功能之一。通过配置寄存器，可以灵活地把端口配置成数字通用端口/数字功能端口。
Pxy_SEL 寄存器用于数字通用端口/数字功能端口切换，每一个端口都可以独立配置成系统所需的功能端口。

表 8-2 端口复用表

AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA00	USART1_CTS	LPUART1_TXD	TIM0_ETR	VC0_OUT	TIM1_CHA	TIM3_ETR	TIM0_CHA
PA01	USART1_RTS	LPUART1_RXD	TIM0_CHB	TIM1_ETR	TIM1_CHB	HCLK_OUT	SPI1_MOSI
PA02	USART1_TXD	TIM0_CHA	VC1_OUT	TIM1_CHA	TIM2_CHA	PCLK_OUT	SPI1_MISO
PA03	USART1_RXD	TIM0_GATE	TIM1_CHB	TIM2_CHB	SPI1_CS	TIM3_CH1A	-
PA04	SPI0_CS	USART1_TXD	LPUART1_RTS	TIM2_ETR	-	LVD_OUT	TIM3_CH2B
PA05	SPI0_SCK	TIM0_ETR	LPUART1_CTS	TIM0_CHA	USART1_RTS	XTL_OUT	XTH_OUT
PA06	SPI0_MISO	LPTIM0_TOG	TIM3_BK	TIM1_CHA	VC0_OUT	TIM3_GATE	LPUART0_CTS
PA07	SPI0_MOSI	LPTIM0_TOGN	HCLK_OUT	TIM3_CH0B	TIM2_CHA	VC1_OUT	LPUART0_RXD
PA08	USART0_TXD	TIM3_CH0A	CTRIM_ETRTOG	LPTIM1_ETR	TIM1_GATE	-	TIM3_BK
PA09	USART0_TXD	TIM3_CH1A	TIM0_BK	I2C0_SCL	LPUART0_TXD	HCLK_OUT	TIM3_CH0A
PA10	USART0_RXD	TIM3_CH2A	TIM2_BK	I2C0_SDA	TIM2_GATE	PCLK_OUT	TIM0_CHB
PA11	USART0_CTS	TIM3_GATE	I2C1_SCL	HSI2C_SCL	VC0_OUT	SPI0_MISO	CTRIM_ETRTOG
PA12	USART0_RTS	TIM3_ETR	I2C1_SDA	HSI2C_SDA	VC1_OUT	SPI0_MOSI	-
PA13	IR_OUT	USART0_RXD	LVD_OUT	TIM3_ETR	RTC_1HZ	I2C1_SDA	USART1_SCK
PA14	USART1_TXD	USART0_TXD	TIM3_CH2A	LVD_OUT	RCH_OUT	RCL_OUT	I2C1_SCL
PA15	SPI0_CS	USART1_RXD	LPUART1_RTS	TIM0_ETR	TIM0_CHA	TIM3_CH1A	USART0_SCK
PB00	-	TIM3_CH1B	LPUART0_TXD	LPTIM1_TOG	RCH_OUT	RCL_OUT	-
PB01	-	PCLK_OUT	TIM3_CH2B	-	LPUART0_RTS	LPTIM1_ETR	TCLK_OUT
PB02	LPTIM0_TOG	-	LPUART1_TXD	-	TIM1_BK	TIM0_BK	TIM2_BK
PB03	SPI0_SCK	TIM0_CHB	TIM1_GATE	TIM3_CH0A	LPTIM0_GATE	XTL_OUT	XTH_OUT
PB04	SPI0_MISO	-	TIM2_BK	USART0_CTS	TIM2_GATE	TIM3_CH0B	LPTIM0_ETR
PB05	SPI0_MOSI	HSI2C_SCL	TIM1_BK	-	LPTIM0_GATE	-	USART0_RTS
PB06	I2C0_SCL	USART0_TXD	TIM1_CHB	TIM0_CHA	LPTIM0_ETR	TIM3_CH0A	LPTIM0_TOG
PB07	I2C0_SDA	USART0_RXD	TIM2_CHB	LPUART1_CTS	TIM0_CHB	LPTIM0_TOGN	-
PB08	I2C0_SCL	TIM1_CHA	USART1_SCK	TIM2_CHA	TIM0_GATE	TIM3_CH2A	USART0_TXD
PB09	I2C0_SDA	IR_OUT	SPI1_CS	TIM2_CHA	HSI2C_SDA	TIM2_CHB	USART0_RXD

AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PB10	I2C1_SCL	SPI1_SCK	TIM1_CHA	LPUART0_TXD	TIM3_CH1A	LPUART1_RTS	USART1_RTS
PB11	I2C1_SDA	TIM1_CHB	LPUART0_RXD	TIM2_GATE	-	LPUART1_CTS	USART1_CTS
PB12	SPI1_CS	TIM3_BK	LPUART0_TXD	TIM0_BK	USART0_SCK	LPUART0_RTS	-
PB13	SPI1_SCK	I2C1_SCL	TIM3_CH0B	LPUART0_CTS	TIM1_CHA	TIM1_GATE	-
PB14	SPI1_MISO	I2C1_SDA	TIM3_CH1B	TIM0_CHA	RTC_1HZ	LPUART0_RTS	TIM1_BK
PB15	SPI1_MOSI	TIM3_CH2B	TIM0_CHB	TIM0_GATE	-	-	LPUART1_RXD
PC00	LPTIM0_GATE	-	USART1_CTS	HSI2C_SCL	-	-	-
PC01	LPTIM0_TOG	-	USART1_RTS	HSI2C_SDA	-	-	-
PC02	SPI1_MISO	LPTIM0_TOGN	-	-	-	-	-
PC03	SPI1_MOSI	LPTIM0_ETR	LPTIM0_TOGN	-	-	-	-
PC04	LPUART0_TXD	TIM2_ETR	IR_OUT	-	-	-	-
PC05	LPUART0_RXD	-	-	-	-	-	-
PC06	-	-	TIM2_CHA	LPTIM1_GATE	-	LPUART2_RXD	-
PC07	-	-	TIM2_CHB	LPTIM1_ETR	-	LPUART2_TXD	-
PC08	-	-	TIM2_ETR	LPTIM1_TOG	-	LPUART2_CTS	-
PC09	-	-	TIM1_ETR	LPTIM1_TOGN	-	LPUART2_RTS	-
PC10	LPUART1_TXD	LPUART0_TXD	-	-	-	-	-
PC11	LPUART1_RXD	LPUART0_RXD	-	-	-	-	-
PC12	LPUART0_TXD	LPUART1_TXD	-	-	-	-	-
PC13	RTC_TAMP	RTC_1HZ	TIM3_CH1B	-	-	-	-
PC14	-	USART0_SCK	-	LPTIM1_TOGN	-	-	-
PC15	-	USART1_SCK	-	LPTIM1_GATE	-	-	-
PD00	I2C0_SDA	CTIM_ETRTOG	USART1_TXD	-	-	LPUART2_RXD	-
PD01	I2C0_SCL	-	USART1_RXD	-	-	LPUART2_TXD	-
PD02	-	LPUART0_RTS	TIM1_ETR	HSI2C_SCL	-	-	-
PD03	-	-	-	-	-	-	-
PD04	-	-	LPTIM1_TOGN	-	-	-	-
PD05	USART0_SCK	-	LPTIM1_TOG	HSI2C_SDA	-	-	-
PD06	I2C1_SCL	LPUART1_CTS	USART0_CTS	LPTIM1_GATE	-	LPUART2_CTS	-
PD07	I2C1_SDA	LPUART1_RTS	USART0_RTS	LPTIM1_TOGN	-	LPUART2_RTS	-

8.3.5 端口中断功能

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，分别对应的中断使能寄存器为高电平中断使能寄存器 (PxHIE) /低电平中断使能寄存器

(PxLIE) /上升沿中断使能寄存器 (PxRIE) /下降沿中断使能寄存器 (PxFIE)。当中断触发时，通过查询中断状态寄存器 (Px_STAT) 就可以判断是哪一个端口触发了中断，通过清零中断清除寄存器 (Px_ICLR) 就可以清除对应的中断状态标志位。

8.3.6 端口输入滤波功能

PA00、PA08、PB00 和 PB08 具备数字输入滤波功能。如果打开滤波功能，则 MCU 将接收经过滤波的 PAD 输入信号；否则 MCU 仍接收原始 PAD 输入信号。滤波时钟和分频系数均可通过滤波配置寄存器 (GPIO_FLTCFG) 进行配置。滤波功能时序图如下：

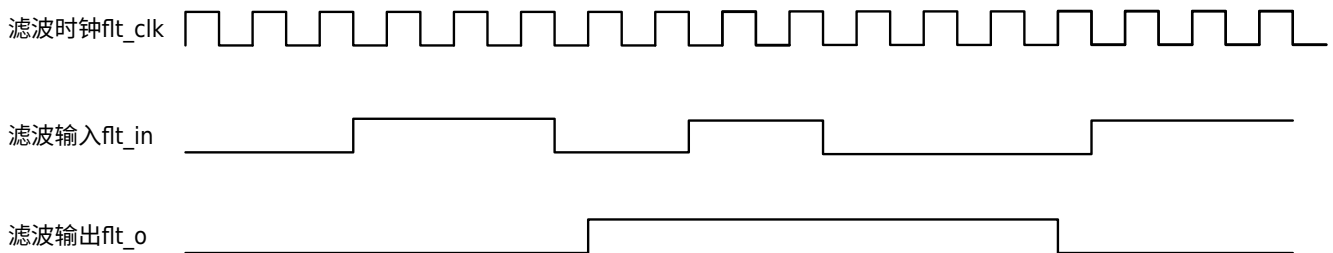


图 8-4 GPIO 端口输入滤波功能时序图



注意

如果需要低功耗唤醒，则要关闭滤波功能或者将滤波时钟切换到 RC256K。

8.3.7 端口复位状态

在复位期间及复位完成未配置 GPIO 寄存器之前，除下面几个 I/O 端口，其余端口被配置为模拟端口状态；特殊管脚的状态如下：

- SWCLK (PA14)：数字输入上拉
- SWDIO (PA13)：数字输入上拉
- PA10：数字输入下拉
- PA11：数字输入

8.3.8 HSI2C 通信电平选择功能

HSI2C 接口支持不同电压下电平信号通信，即低电压电平识别。且仅当 PAD 选择 HSI2C 功能时，支持此功能。通信电平电压识别选择通过配置 GPIO_CTRL1.HSI2C_LSEL 实现。GPIO_CTRL1.HSI2C_LSEL 配置为 0 时，HSI2C 和 GPIO 的输入电平正常识别；GPIO_CTRL1.HSI2C_LSEL 配置为 1 时，HSI2C 和 GPIO 的输入低电压电平识别。

8.4 操作示例

8.4.1 端口复用配置为模拟端口操作流程

操作步骤

设置寄存器 PxADS[n]为 1。

8.4.2 端口复用配置为数字通用端口操作流程

操作步骤

步骤 1. 设置寄存器 PxADS[n]为 0。

- 步骤 2. 设置寄存器 Pxy_SEL 为 0。
- 步骤 3. 设置寄存器 PxDIR[n]为 1：端口方向为输入，CPU 可以读取端口的状态 PxIN[n]。
- 步骤 4. 设置寄存器 PxDIR[n]为 0：端口方向为输出。
- 步骤 5. 设置寄存器 PxOUT[n]为 1：端口输出高电平。
- 步骤 6. 设置寄存器 PxOUT[n]为 0：端口输出低电平。

8.4.3 端口复用配置为数字功能端口操作流程

操作步骤

- 步骤 1. 设置寄存器 PxADS[n]为 0。
- 步骤 2. 根据系统需求，设置寄存器 Pxy_SEL 为 1~7，具体请参考[端口复用表](#)。
- 步骤 3. 根据系统需求，设置寄存器 PxDIR[n]。
- 步骤 4. 根据系统需求，设置寄存器 PxPU[n]/PxPD[n]/PxOD[n]。

8.4.4 端口复用配置为调试测试端口操作流程

背景信息

具体操作请参见《SWD 调试接口》章节相关介绍。

8.4.5 端口复用配置为红外输出信号操作流程

背景信息

端口 PA13、PB09、PC04 可以把频率为 38.4kHz 的内部时钟信号调制成红外信号输出。



说明

在开启 38.4kHz 的 RCL 时钟时进行操作才会有载波输出。

操作步骤

- 步骤 1. 设置寄存器 PAADS[13]/PBADS[9]/PCADS[4]为 0。
- 步骤 2. 设置寄存器 PA13_SEL=1/PB09_SEL=2/PC04_SEL=3。
- 步骤 3. 设置寄存器 PADIR[13]=0/PBDIR[9]=0/PCDIR[4]=0，端口方向为输出。
- 步骤 4. 设置寄存器 GPIO_CTRL1[14]选择红外信号输出极性。
- 步骤 5. 设置寄存器 PAOUT[13]=1/PBOUT[9]=1/PCOUT[4]=1，选择门控红外信号输出。

8.4.6 端口高电平中断操作流程

操作步骤

- 步骤 1. 设置寄存器 PxADS[n]为 0。
- 步骤 2. 设置寄存器 Pxy_SEL 为 0。
- 步骤 3. 设置寄存器 PxDIR[n]为 1。
- 步骤 4. 设置寄存器 PxHIE[n]为 1。
- 步骤 5. 中断触发后读取中断状态寄存器 Px_STAT[n]。
- 步骤 6. 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]。

8.4.7 端口低电平中断操作流程

操作步骤

- 步骤 1. 设置寄存器 PxADS[n]为 0。
- 步骤 2. 设置寄存器 Pxy_SEL 为 0。
- 步骤 3. 设置寄存器 PxDIR[n]为 1。
- 步骤 4. 设置寄存器 PxLIE[n]为 1。
- 步骤 5. 中断触发后读取中断状态寄存器 Px_STAT[n]。

步骤 6. 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]。

8.4.8 端口上升沿中断操作流程

操作步骤

步骤 1. 设置寄存器 PxADS[n]为 0。

步骤 2. 设置寄存器 Pxy_SEL 为 0。

步骤 3. 设置寄存器 PxDIR[n]为 1。

步骤 4. 设置寄存器 PxRIE[n]为 1。

步骤 5. 中断触发后读取中断状态寄存器 Px_STAT[n]。

步骤 6. 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]。

8.4.9 端口下降沿中断操作流程

操作步骤

步骤 1. 设置寄存器 PxADS[n]为 0。

步骤 2. 设置寄存器 Pxy_SEL 为 0。

步骤 3. 设置寄存器 PxDIR[n]为 1。

步骤 4. 设置寄存器 PxFIE[n]为 1。

步骤 5. 中断触发后读取中断状态寄存器 Px_STAT[n]。

步骤 6. 设置寄存器 Px_ICLR[n]为 0 清除中断状态寄存器 Px_STAT[n]。

8.4.10 端口上拉使能配置操作流程

操作步骤

设置寄存器 PxPU[n]为 1。

8.4.11 端口下拉使能配置操作流程

背景信息

当 PxPU[n]和 PxPD[n]同时置 1 时, PxPU[n]优先级高, PxPD[n]无效。

操作步骤

步骤 1. 设置寄存器 PxPU[n]为 0。

步骤 2. 设置寄存器 PxPD[n]为 1。

8.4.12 端口增强驱动配置操作流程

操作步骤

设置寄存器 PxDR[n]为 0。

8.4.13 端口开漏输出配置操作流程

操作步骤

设置寄存器 PxOD[n]为 1。

8.4.14 端口位置位操作流程

操作步骤

设置寄存器 PxBSET[n]为 1。

8.4.15 端口位清零操作流程

操作步骤

设置寄存器 PxBCLR[n]为 1。

8.4.16 端口位置位清零操作流程

操作步骤

设置寄存器 PxBSETCLR[n]为 1。

8.4.17 端口输入滤波操作流程

操作步骤

- 步骤 1. 设置寄存器 GPIO_FLTCFG.FCLK_SEL 为 0 选择 HCLK 为滤波时钟。
- 步骤 2. 设置寄存器 GPIO_FLTCFG.FCLK_DIV 为 0b01，选择滤波时钟分频系数为 4。
- 步骤 3. 设置寄存器 GPIO_FLTCFG.PA0FLT_EN 为 1，开启 PA00 的滤波功能。

8.5 寄存器

8.5.1 寄存器总表

基地址：0x40020C00

表 8-3 端口控制器 (GPIO) 寄存器偏移地址

偏移地址	寄存器	描述
0x00+0x4*y	PAy_SEL	端口 PAy 功能配置寄存器 (y=0~15)
0x40+0x4*y	PBy_SEL	端口 PBy 功能配置寄存器 (y=0~15)
0x80+0x4*y	PCy_SEL	端口 PCy 功能配置寄存器 (y=0~15)
0xC0+0x4*y	PDy_SEL	端口 PDy 功能配置寄存器 (y=0~7)
0x100	PADIR	端口 PA 输入输出配置寄存器
0x104	PAIN	端口 PA 输入值寄存器
0x108	PAOUT	端口 PA 输出值配置寄存器
0x10C	PAADS	端口 PA 数模配置寄存器
0x110	PABSET	端口 PA 位置位寄存器
0x114	PABCLR	端口 PA 位清零寄存器
0x118	PABSETCLR	端口 PA 位置位清零寄存器
0x11C	PADR	端口 PA 驱动能力配置寄存器
0x120	PAPU	端口 PA 上拉使能配置寄存器
0x124	PAPD	端口 PA 下拉使能配置寄存器
0x12C	PAOD	端口 PA 开漏输出配置寄存器
0x130	PAHIE	端口 PA 高电平中断使能配置寄存器
0x134	PALIE	端口 PA 低电平中断使能配置寄存器
0x138	PARIE	端口 PA 上升沿中断使能配置寄存器
0x13C	PAFIE	端口 PA 下降沿中断使能配置寄存器
0x200	PA_STAT	端口 PA 中断状态寄存器
0x210	PA_ICLR	端口 PA 中断清除寄存器
0x140	PBDIR	端口 PB 输入输出配置寄存器
0x144	PBIN	端口 PB 输入值寄存器
0x148	PBOUT	端口 PB 输出值配置寄存器
0x14C	PBADs	端口 PB 数模配置寄存器
0x150	PBBSET	端口 PB 位置位寄存器

偏移地址	寄存器	描述
0x154	PBBCLR	端口 PB 位清零寄存器
0x158	PBBSETCLR	端口 PB 位置位清零寄存器
0x15C	PBDR	端口 PB 驱动能力配置寄存器
0x160	PBPU	端口 PB 上拉使能配置寄存器
0x164	PBPD	端口 PB 下拉使能配置寄存器
0x16C	PBOD	端口 PB 开漏输出配置寄存器
0x170	PBHIE	端口 PB 高电平中断使能配置寄存器
0x174	PBLIE	端口 PB 低电平中断使能配置寄存器
0x178	PBRIE	端口 PB 上升沿中断使能配置寄存器
0x17C	PBFIE	端口 PB 下降沿中断使能配置寄存器
0x240	PB_STAT	端口 PB 中断状态寄存器
0x250	PB_ICLR	端口 PB 中断清除寄存器
0x180	PCDIR	端口 PC 输入输出配置寄存器
0x184	PCIN	端口 PC 输入值寄存器
0x188	PCOUT	端口 PC 输出值配置寄存器
0x18C	PCADS	端口 PC 数模配置寄存器
0x190	PCBSET	端口 PC 位置位寄存器
0x194	PCBCLR	端口 PC 位清零寄存器
0x198	PCBSETCLR	端口 PC 位置位清零寄存器
0x19C	PCDR	端口 PC 驱动能力配置寄存器
0x1A0	PCPU	端口 PC 上拉使能配置寄存器
0x1A4	PCPD	端口 PC 下拉使能配置寄存器
0x1AC	PCOD	端口 PC 开漏输出配置寄存器
0x1B0	PCHIE	端口 PC 高电平中断使能配置寄存器
0x1B4	PCLIE	端口 PC 低电平中断使能配置寄存器
0x1B8	PCRIE	端口 PC 上升沿中断使能配置寄存器
0x1BC	PCFIE	端口 PC 下降沿中断使能配置寄存器
0x280	PC_STAT	端口 PC 中断状态寄存器
0x290	PC_ICLR	端口 PC 中断清除寄存器
0x1C0	PDDIR	端口 PD 输入输出配置寄存器

偏移地址	寄存器	描述
0x1C4	PDIN	端口 PD 输入值寄存器
0x1C8	PDOUT	端口 PD 输出值配置寄存器
0x1CC	PDADS	端口 PD 数模配置寄存器
0x1D0	PDBSET	端口 PD 位置位寄存器
0x1D4	PDBCLR	端口 PD 位清零寄存器
0x1D8	PDBSETCLR	端口 PD 位置位清零寄存器
0x1DC	PDDR	端口 PD 驱动能力配置寄存器
0x1E0	PDPUP	端口 PD 上拉使能配置寄存器
0x1E4	PDPD	端口 PD 下拉使能配置寄存器
0x1EC	PDOD	端口 PD 开漏输出配置寄存器
0x1F0	PDHIE	端口 PD 高电平中断使能配置寄存器
0x1F4	PDLIE	端口 PD 低电平中断使能配置寄存器
0x1F8	PDRIE	端口 PD 上升沿中断使能配置寄存器
0x1FC	PDFIE	端口 PD 下降沿中断使能配置寄存器
0x2C0	PD_STAT	端口 PD 中断状态寄存器
0x2D0	PD_ICLR	端口 PD 中断清除寄存器
0x304	GPIO_CTRL1	端口辅助功能配置寄存器 1
0x308	GPIO_CTRL2	端口辅助功能配置寄存器 2
0x30C	GPIO_TIMGS	端口辅助功能定时器门控选择
0x310	GPIO_TIMES	端口辅助功能定时器 ETR 选择
0x314	GPIO_TIMCPS0	端口辅助功能定时器捕获输入选择 0
0x318	GPIO_TIMCPS1	端口辅助功能定时器捕获输入选择 1
0x330	GPIO_FLTCFG	端口滤波配置寄存器

8.5.2 端口通用寄存器

8.5.2.1 端口 Px 输入输出配置寄存器 (PxDIR) (x=A/B/C/D)

偏移地址 (Offset) : 0x100 (PA) , 0x140 (PB) , 0x180 (PC) , 0x1C0 (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x0000FFFF																																					
Name	Reserved																PxDIR																					
Access																	RW																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxDIR	端口 Px 输入输出配置寄存器 对应 Px15~Px00: ● 0b0: 配置成输出 ● 0b1: 配置成输入  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置, 例如: PxDIR[15]对应端口 Px15。	RW

8.5.2.2 端口 Px 输入值寄存器 (PxIN) (x=A/B/C/D)

偏移地址 (Offset) : 0x104 (PA) , 0x144 (PB) , 0x184 (PC) , 0x1C4 (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	-																																					
Name	Reserved																PxIN																					
Access																	RO																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxIN	端口 Px 输入值寄存器 对应 Px15~Px00: ● 0b0: 输入为低电平 ● 0b1: 输入为高电平  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RO

8.5.2.3 端口 Px 输出值配置寄存器 (PxOUT) (x=A/B/C/D)

偏移地址 (Offset) : 0x108 (PA) , 0x148 (PB) , 0x188 (PC) , 0x1C8 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																PxOUT															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxOUT	端口 Px 输出值寄存器 对应 Px15~Px00: ● 0b0: 输出为低电平 ● 0b1: 输出为高电平, 如果配置成开漏输出, 则需外部上拉电阻来拉高  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.4 端口 Px 数模配置寄存器 (PxADS) (x=A/B/C/D)

偏移地址 (Offset) : 0x10C (PA) , 0x14C (PB) , 0x18C (PC) , 0x1CC (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000093FF(PA);0x0000FFFF(PB/PC);0x000000FF(PD)																															
Name	Reserved																PxADS															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxADS	端口 Px 数模配置寄存器 对应 Px15~Px00: ● 0b0: 配置为数字端口 ● 0b1: 配置为模拟端口  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.5 端口 Px 位置位寄存器 (PxBSET) (x=A/B/C/D)

偏移地址 (Offset) : 0x110 (PA) , 0x150 (PB) , 0x190 (PC) , 0x1D0 (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																PxBSET																					
Access																	WO																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxBSET	端口 Px 位置位寄存器 对应 Px15~Px00, 写 1 置位, 写 0 对该端口的状态无影响  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	WO

8.5.2.6 端口 Px 位清零寄存器 (PxBCLR) (x=A/B/C/D)

偏移地址 (Offset) : 0x114 (PA) , 0x154 (PB) , 0x194 (PC) , 0x1D4 (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																PxBCLR																					
Access																	WO																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxBCLR	端口 Px 位清零寄存器 对应 Px15~Px00, 写 1 清零, 写 0 对该端口的状态无影响  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	WO

8.5.2.7 端口 Px 位置位清零寄存器 (PxBSETCLR) (x=A/B/C/D)

偏移地址 (Offset) : 0x118 (PA) , 0x158 (PB) , 0x198 (PC) , 0x1D8 (PD)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	PxBSET																PxBCLR															
Access	WO																WO															

位/位域	标记	功能描述	读写
31:16	PxBSET	端口 Px 位置位寄存器 对应 Px15~Px00, 写 1 置位, 写 0 对该端口的状态无影响 说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	WO
15:0	PxBCLR	端口 Px 位清零寄存器 对应 Px15~Px00, 写 1 清零, 写 0 对该端口的状态无影响 说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	WO

说明

PxBSET 和 PxBCLR 相同位同时置 1 时, PxBCLR 具有高优先级, 该端口被清零。

8.5.2.8 端口 Px 上拉使能配置寄存器 (PxPU) (x=A/B/C/D)

偏移地址 (Offset) : 0x120 (PA) , 0x160 (PB) , 0x1A0 (PC) , 0x1E0 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00006000(PA); 0x00000000(PB/PC/PD)																															
Name	Reserved																PxPU															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxPU	端口 Px 上拉使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.9 端口 Px 驱动能力配置寄存器 (PxDR) (x=A/B/C/D)

偏移地址 (Offset) : 0x11C (PA) , 0x15C (PB) , 0x19C (PC) , 0x1DC (PD)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PxDR															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxDR	端口 Px 驱动能力配置寄存器 对应 Px15~Px00: ● 0b0: 高驱动能力 ● 0b1: 低驱动能力  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.10 端口 Px 下拉使能配置寄存器 (PxPD) (x=A/B/C/D)

偏移地址 (Offset) : 0x124 (PA) , 0x164 (PB) , 0x1A4 (PC) , 0x1E4 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000400(PA); 0x00000000(PB/PC/PD)																															
Name	Reserved																PxPD															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxPD	端口 Px 下拉使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.11 端口 Px 开漏输出配置寄存器 (PxOD) (x=A/B/C/D)

偏移地址 (Offset) : 0x12C (PA) , 0x16C (PB) , 0x1AC (PC) , 0x1EC (PD)

xOffset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PxOD															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxOD	端口 Px 开漏输出配置寄存器 对应 Px15~Px00: ● 0b0: 将端口输出方式设置为推挽输出 ● 0b1: 将端口输出方式设置为开漏输出  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.12 端口 Px 高电平中断使能配置寄存器 (PxHIE) (x=A/B/C/D)

偏移地址 (Offset) : 0x130 (PA) , 0x170 (PB) , 0x1B0 (PC) , 0x1F0 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PxHIE															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxHIE	端口 Px 高电平中断使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.13 端口 Px 低电平中断使能配置寄存器 (PxLIE) (x=A/B/C/D)

偏移地址 (Offset) : 0x134 (PA) , 0x174 (PB) , 0x1B4 (PC) , 0x1F4 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																PxLIE															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxLIE	端口 Px 低电平中断使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.14 端口 Px 上升沿中断使能配置寄存器 (PxRIE) (x=A/B/C/D)

偏移地址 (Offset) : 0x138 (PA) , 0x178 (PB) , 0x1B8 (PC) , 0x1F8 (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																PxRIE																					
Access																	RW																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxRIE	端口 Px 上升沿中断使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.15 端口 Px 下降沿中断使能配置寄存器 (PxFIE) (x=A/B/C/D)

偏移地址 (Offset) : 0x13C (PA) , 0x17C (PB) , 0x1BC (PC) , 0x1FC (PD)

Offset	Bit Position																																					
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																PxFIE																					
Access																	RW																					

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	PxFIE	端口 Px 下降沿中断使能配置寄存器 对应 Px15~Px00: ● 0b0: 禁止 ● 0b1: 使能  说明 每个 Bit 对应一个端口, 请不要对不存在的端口进行配置。	RW

8.5.2.16 端口 Px 中断状态寄存器 (Px_STAT) (x=A/B/C/D)

偏移地址 (Offset) : 0x200 (PA) , 0x240 (PB) , 0x280 (PC) , 0x2C0 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																Px_STAT															
Access																	RO															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	Px_STAT	端口 Px 中断状态寄存器 对应 Px15~Px00: ● 0b0: 无中断触发 ● 0b1: 中断触发  说明 每个 Bit 对应一个端口，请不要对不存在的端口进行配置。	RO

8.5.2.17 端口 Px 中断清除寄存器 (Px_ICLR) (x=A/B/C/D)

偏移地址 (Offset) : 0x210 (PA) , 0x250 (PB) , 0x290 (PC) , 0x2D0 (PD)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																Px_ICLR															
Access																	WO															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	Px_ICLR	端口 Px 中断清除寄存器 对应 Px15~Px00, 写 0 清除标志, 写 1 无效; 读出恒为 1  说明 每个 Bit 对应一个端口，请不要对不存在的端口进行配置。	WO

8.5.3 端口辅助寄存器

8.5.3.1 端口辅助功能配置寄存器 1 (GPIO_CTRL1)

Offset	Bit Position																															
0x304	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																HSI2C_LSEL	IR_POL	HCLK_EN	PCLK_EN	HCLK_SEL	PCLK_SEL	Reserved									
Access																	RW	RW	RW	RW	RW	RW										

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	HSI2C_LSEL	HSI2C I/O 信号有效电平选择 0b0: 输入电平正常识别 0b1: 输入电平低电压识别  说明 具体数值参阅《数据手册》。	RW
14	IR_POL	IR 输出极性选择 0b0: 正向输出 0b1: 反向输出	RW
13	HCLK_EN	HCLK 输出门控 0b0: 门控 0b1: 输出	RW
12	PCLK_EN	PCLK 输出门控 0b0: 门控 0b1: 输出	RW
11:10	HCLK_SEL	HCLK 输出分频选择 0b00: HCLK 0b01: HCLK/2 0b10: HCLK/4 0b11: HCLK/8	RW
9:8	PCLK_SEL	PCLK 输出分频选择 0b00: PCLK 0b01: PCLK/2 0b10: PCLK/4 0b11: PCLK/8	RW
7:0	Reserved	-	-

8.5.3.2 端口辅助功能配置寄存器 2 (GPIO_CTRL2)

Offset	Bit Position																																	
0x308	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																AHB_SEL	Reserved								TCLK_DIV	TCLK_SEL				Reserved			
Access																	RW									RW	RW							

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	AHB_SEL	AHB 总线访问模式 0b0: 使用 FASTIO 总线访问 PxIN、PxOUT 寄存器 0b1: 使用 AHB 总线访问 PxIN、PxOUT 寄存器	RW
14:8	Reserved	-	-
7:6	TCLK_DIV	TCLK 输出分频选择 0b00: TCLK 0b01: TCLK/2 0b10: TCLK/4 0b11: TCLK/8	RW
5:4	TCLK_SEL	TCLK 输出信号源选择 0b00: RCH 0b01: RCL 0b10: XTH 0b11: XTL	RW
3:0	Reserved	-	-

8.5.3.3 端口辅助功能定时器门控选择 (GPIO_TIMGS)

Offset	Bit Position																																					
0x30C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																		LPTIM0_G				TIM3_G				TIM2_G				TIM1_G				TIM0_G			
Access																			RW				RW				RW				RW				RW			

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14:12	LPTIM0_G	LPTIM0 定时器 GATE 输入选择 参数选择详见下表	RW
11:9	TIM3_G	TIM3 定时器 GATE 输入选择 参数选择详见下表	RW
8:6	TIM2_G	TIM2 定时器 GATE 输入选择 参数选择详见下表	RW
5:3	TIM1_G	TIM1 定时器 GATE 输入选择 参数选择详见下表	RW
2:0	TIM0_G	TIM0 定时器 GATE 输入选择 参数选择详见下表	RW

取值	LPTIM0_G	TIM3_G	TIM2_G	TIM1_G	TIM0_G
0b000	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL
0b001	LPUART0_RXD	USART0_RXD	USART0_RXD	LPUART0_RXD	USART0_RXD
0b010	LPUART1_RXD	USART1_RXD	USART1_RXD	LPUART1_RXD	USART1_RXD
0b011	VC0_OUT	-	VC0_OUT	VC0_OUT	VC0_OUT
0b100	VC1_OUT	LPUART2_RXD	VC1_OUT	VC1_OUT	VC1_OUT
0b101	PB03	VC0_OUT	PA10	PA08	PA03
0b110	PB05	PA06	PB04	PB03	PB08
0b111	PC00	PA11	PB11	PB13	PB15

8.5.3.4 端口辅助功能定时器 ETR 选择 (GPIO_TIMES)

Offset	Bit Position																															
0x310	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																LPTIM0_E		TIM3_E		TIM2_E		TIM1_E		TIM0_E							
Access																	RW															

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14:12	LPTIM0_E	LPTimer0 定时器 ETR 输入选择 参数选择详见下表	RW
11:9	TIM3_E	TIM3 定时器 ETR 输入选择 参数选择详见下表	RW
8:6	TIM2_E	TIM2 定时器 ETR 输入选择 参数选择详见下表	RW
5:3	TIM1_E	TIM1 定时器 ETR 输入选择 参数选择详见下表	RW
2:0	TIM0_E	TIM0 定时器 ETR 输入选择 参数选择详见下表	RW

取值	LPTIM0_E	TIM3_E	TIM2_E	TIM1_E	TIM0_E
0b000	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL
0b001	-	USART0_RXD	LPUART0_RXD	USART0_RXD	LPUART0_RXD
0b010	LVD_OUT	USART1_RXD	LPUART1_RXD	USART1_RXD	LPUART1_RXD
0b011	VC0_OUT	VC1_OUT	VC0_OUT	VC1_OUT	VC0_OUT
0b100	VC1_OUT	-	-	LVD_OUT	LVD_OUT
0b101	PA08	PA00	PA04	PA01	PA00
0b110	PB01	PA12	PC04	PC09	PA05
0b111	PC07	PA13	PC08	PD02	PA15

8.5.3.5 端口辅助功能定时器捕获输入选择 0 (GPIO_TIMCPS0)

Offset	Bit Position																															
0x314	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																		TIM3_CB		TIM3_CA		TIM2_CA		TIM1_CA		TIM0_CA					
Access																			RW		RW		RW		RW		RW					

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14:12	TIM3_CB	TIM3 定时器 CH0B 输入选择 参数选择详见下表	RW
11:9	TIM3_CA	TIM3 定时器 CH0A 输入选择 参数选择详见下表	RW
8:6	TIM2_CA	TIM2 定时器 CHA 输入选择 参数选择详见下表	RW
5:3	TIM1_CA	TIM1 定时器 CHA 输入选择 参数选择详见下表	RW
2:0	TIM0_CA	TIM0 定时器 CHA 输入选择 参数选择详见下表	RW

取值	TIM3_CB	TIM3_CA	TIM2_CA	TIM1_CA	TIM0_CA
0b000	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL	Pxy_SEL
0b001	USART0_RXD	LPUART1_RXD	LPUART0_RXD	USART1_RXD	USART0_RXD
0b010	USART1_RXD	LPUART0_RXD	VC0_OUT	PA00	PA00
0b011	-	-	PA02	PA02	PA02
0b100	VC1_OUT	VC0_OUT	PA07	PA06	PA05
0b101	PA07	PA08	PB08	PB08	PA15
0b110	PB04	PB03	PB09	PB10	PB06
0b111	PB13	PB06	PC06	PB13	PB14

8.5.3.6 端口辅助功能定时器捕获输入选择 1 (GPIO_TIMCPS1)

Offset	Bit Position																															
0x318	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																LPTIM1_E		LPTIM1_G		Reserved											
Access																	RW		RW													

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:9	LPTIM1_E	LPTIM1 定时器 ETR 输入选择 参数选择详见下表	RW
8:6	LPTIM1_G	LPTIM1 定时器 GATE 输入选择 参数选择详见下表	RW
5:0	Reserved	-	-

取值	LPTIM1_E	LPTIM1_G
0b000	Pxy_SEL	Pxy_SEL
0b001	USART1_RXD	USART0_RXD
0b010	LPUART1_RXD	LPUART0_RXD
0b011	VC0_OUT	VC1_OUT
0b100	PC03	-
0b101	LPTIM0_OV	PD5
0b110	PB04	-
0b111	PB06	PC6

8.5.3.7 端口滤波配置寄存器 (GPIO_FLTCFG)

Offset	Bit Position																																	
0x330	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																					FCLK_SEL	FCLK_DIV	PA0FLT_EN	PA8FLT_EN	PB0FLT_EN	PB8FLT_EN	Reserved						
Access																						RW	RW	RW	RW	RW	RW							

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	FCLK_SEL	滤波时钟源选择 ● 0b0: HCLK ● 0b1: RC256K	RW
9:8	FCLK_DIV	滤波时钟分频配置 ● 0b00: FCLK ● 0b01: FCLK/4 ● 0b10: FCLK/16 ● 0b11: FCLK/64	RW
7	PA0FLT_EN	PA00 滤波使能 ● 0b0: 滤波关闭 ● 0b1: 滤波使能	RW
6	PA8FLT_EN	PA08 滤波使能 ● 0b0: 滤波关闭 ● 0b1: 滤波使能	RW
5	PB0FLT_EN	PB00 滤波使能 ● 0b0: 滤波关闭 ● 0b1: 滤波使能	RW
4	PB8FLT_EN	PB08 滤波使能 ● 0b0: 滤波关闭 ● 0b1: 滤波使能	RW
3:0	Reserved	-	-

8.5.4 端口功能选择寄存器

8.5.4.1 端口 Pxy 功能配置寄存器 (Pxy_SEL) (x=A/B/C/D)

偏移地址 (Offset) : 0x00+0x4*y (PA) (y=0~15) ;
0x40+0x4*y (PB) (y=0~15) ;
0x80+0x4*y (PC) (y=0~15) ;
0xC0+0x4*y (PD) (y=0~7) 。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												SEL			
Access																																
																													RW			

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2:0	SEL	端口 Pxy 功能选择 ● 0b000: GPIO ● 0b001: AF1 ● 0b010: AF2 ● 0b011: AF3 ● 0b100: AF4 ● 0b101: AF5 ● 0b110: AF6 ● 0b111: AF7  说明 AF1~AF7 的具体功能详见"端口数字复用功能"。	RW

9 DMA 控制器 (DMAC)

9.1 简介

直接存储器存取 (DMA) 用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输；传输过程中无须 CPU 参与，故 CPU 可同步进行其它操作。DMAC 具有 2 条独立的 DMA 通道，每个通道专门用来管理来自于外设或存储器访问的请求。还有一个仲裁器来协调各个 DMA 请求的优先权。

9.2 主要特性

DMA 控制器支持以下特性：

- 2 条独立 DMA 通道，支持优先级配置
- 3 种数据传输宽度：8-bit、16-bit、32-bit
- 4 种传输模式：软件 Block、软件 Burst、硬件 Block、硬件 Burst
- 2 种来源地址类型：外设、存储器
- 2 种目标地址类型：外设、存储器
- 2 种地址变化模式：固定、自增
- 传输地址寻址范围：0x00000000 - 0xFFFFFFFF
- 待传输数据块数量可配置：1~65536
- 待传输数据块大小可配置：1~16
- 支持地址及传输数量重装载功能

9.3 功能说明

9.3.1 功能框图

- DMAC

DMAC 具有 2 条 DMA 通道，当通道之间发生冲突时，由优先级控制器进行仲裁。

- Bus Matrix

CPU 和 DMAC 均连接到总线矩阵上。当 CPU 与 DMAC 访问不同的 AHB 总线设备或 AHB 桥时，数据传输可以同时进行。当 CPU 与 DMAC 访问同一总线设备或 AHB 桥时，CPU 的优先级高于 DMAC。即只有 CPU 释放 AHB 总线设备或 AHB 桥时，DMAC 才可访问该 AHB 总线设备或 AHB 桥。

- DMA requests

如果外设支持硬件 DMA request，则该 DMA 通道可以配置为硬件触发否则只能配置为软件触发。

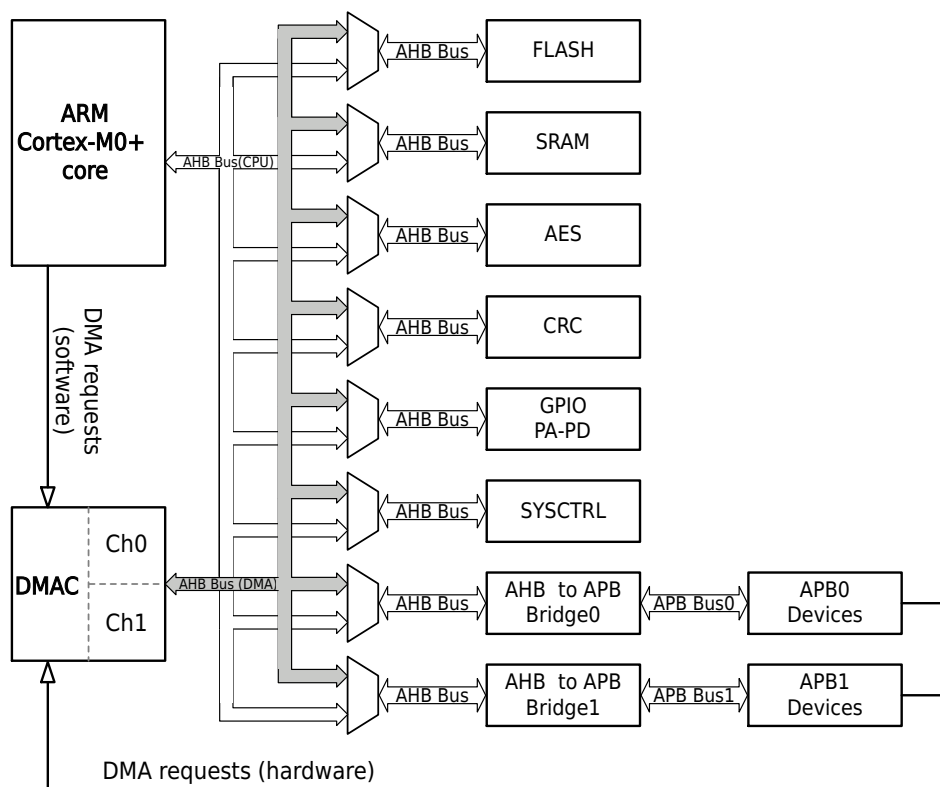


图 9-1 DMA 控制器功能框图

9.3.2 DMAC 传输模式

9.3.2.1 DMAC 传输模式概述

DMA 控制器支持 4 种传输模式，软件 Block 传输模式、软件 Burst 传输模式、硬件 Block 传输模式、硬件 Burst 传输模式。

表 9-1 DMAC 传输模式比对表

对比条目	软件 Block 传输	软件 Burst 传输	硬件 Block 传输	硬件 Burst 传输
是否可以被更高优先级 DMA 通道或 CPU 打断	可以	不可以	可以	不可以
触发传输启动的条件	写 DMAC 寄存器		外设中断标志	
触发一次传输的数据量	$(BC+1)*(TC+1)$		BC+1	$(BC+1)*(TC+1)$
配置后传输的总数据量	$(BC+1)*(TC+1)$			
主要应用场景	● 存储器到存储器 ● 存储器到无 DMA 请求的外设		● 存储器到有 DMA 请求的外设 ● 有 DMA 请求的外设到存储器	● 存储器到存储器 ● 存储器到无 DMA 请求的外设

9.3.2.2 软件 Block 传输模式

当配置 DMAC_CONFx.TRI_SEL=0，DMAC_CONFBx.MODE=0 时，DMAC 工作于软件 Block 传输模式。该模式下用户向 DMAC_CONFx.ST 写入 1，则触发 DAMC 启动软件 Block 传输模式，传输 $(BC+1) * (TC+1)$ 个数据后停止。每传输完成 BC+1 个数据之后，DMAC 插入一个传输间隙，优先级仲裁器在该间隙进行优先级仲裁。如在该间隙发现 CPU 或更高优先 DMA 通道请求总线，则总线占有权转移给 CPU 或更高优先 DMA 通道；当 CPU 或更高优先 DMA 通道释放总线后，未完成的数据传输将继续完成。

软件 Block 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 16-bit，地址自增。

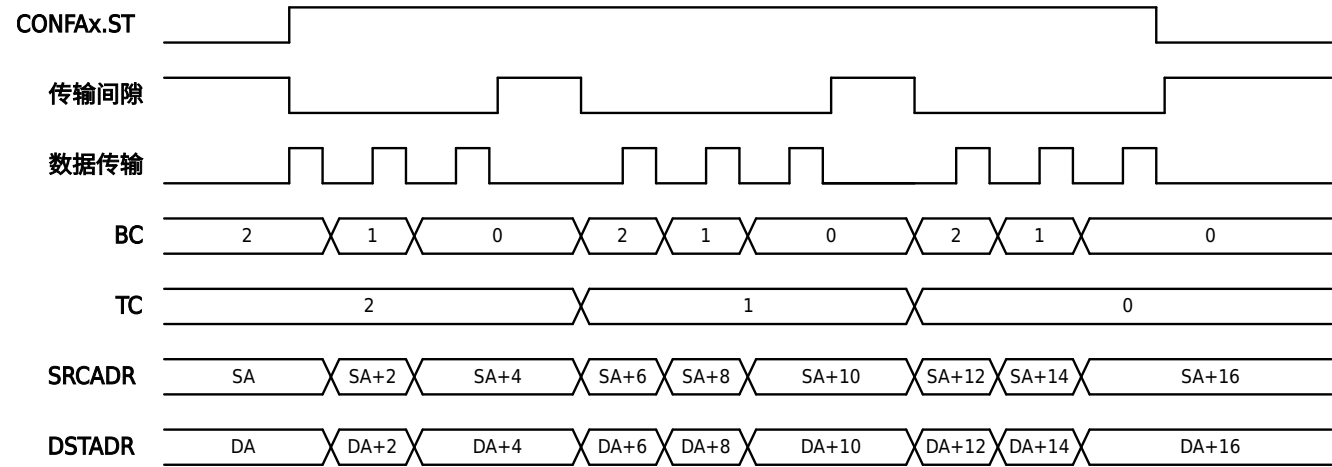


图 9-2 DMA 软件 Block 传输模式

9.3.2.3 软件 Burst 传输模式

当配置 DMAC_CONFx.TRI_SEL=0，DMAC_CONFBx.MODE=1 时，DMAC 工作于软件 Burst 传输模式。该模式下用户向 DMAC_CONFx.ST 写入 1，则触发 DMAC 启动软件 Burst 传输模式，传输 $(BC+1) * (TC+1)$ 个数据后停止。该传输没有传输间隙，所有数据传输完成之前，当前正在工作的 DMA 通道将一直占据总线，CPU 或更高优先 DMA 通道只能等待该操作完成后才能访问总线。建议每次不要传输太多的数据，否则将可能造成 CPU 在一段时间内完全不能工作。

软件 Burst 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 16-bit，地址自增。

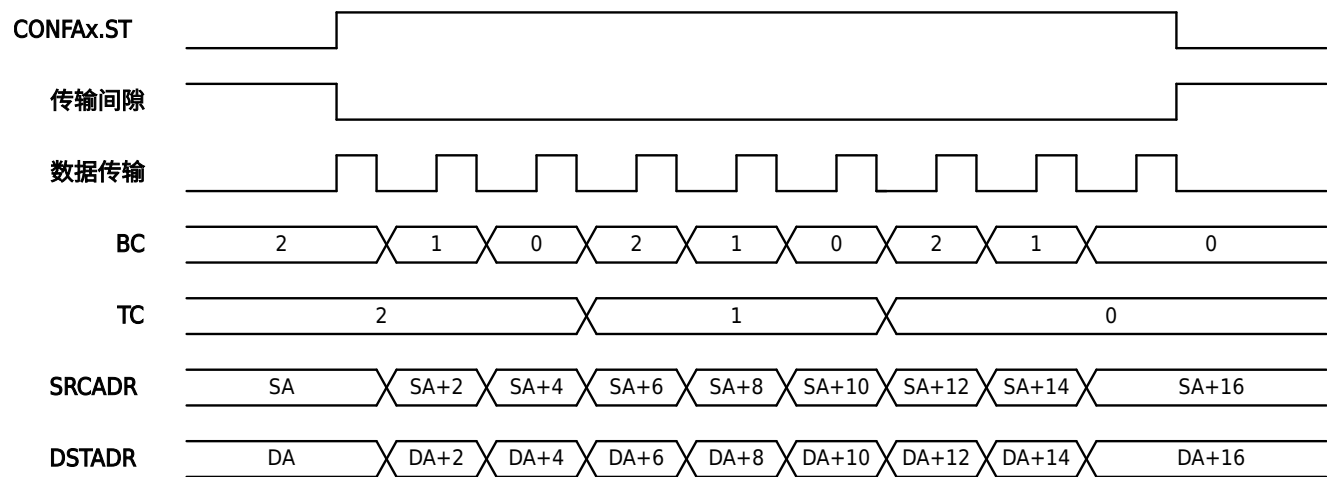


图 9-3 DMA 软件 Burst 传输模式

9.3.2.4 硬件 Block 传输模式

当配置 DMAC_CONFAx.TRI_SEL=非零, DMAC_CONFBx.MODE=0 时, DMAC 工作于硬件 Block 传输模式。该模式下每当外设 DMA request 信号出现一次, 则触发 DMAC 启动一次硬件 Block 传输, 传输(BC+1)个数据。需要出现(TC+1)次外设 DMA request 信号, 才能完成所有 DMA 数据的传输。每完成(BC+1)个数据之后, DMAC 插入一个传输间隙, 优先级仲裁器在该间隙进行优先级仲裁。如在该间隙发现 CPU 或更高优先 DMA 通道也在请求总线, 则总线占有权转移给 CPU 或更高优先 DMA 通道; 当 CPU 或更高优先 DMA 通道释放总线后, 未完成的数据传输将继续完成。



说明

在该模式下, 一般将 BC 设为 0, 以实现从外设获取一个数据或提供一个数据给外设。

DMA 支持的硬件 request 信号类型请参见 [DMAC_CONFAx.TRI_SEL](#) 位功能描述。
硬件 Block 传输示意如下所示, 其中 SA 代表源地址, DA 代表目标地址; 数据块大小为 2 (BC=1), 数据块数量为 3 (TC=2), 数据宽度为 8-bit, 目标地址自增, 源地址固定, 传输完成自动禁止。

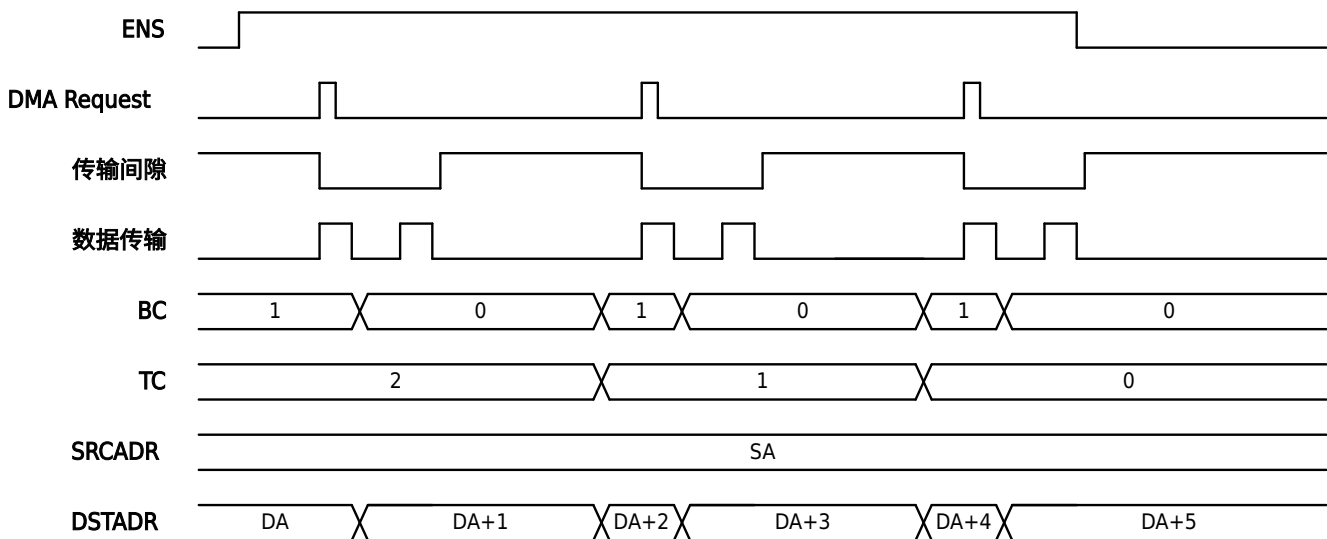


图 9-4 DMA 硬件 Block 传输模式

9.3.2.5 硬件 Burst 传输模式

当配置 DMAC_CONFAx.TRI_SEL=非零, DMAC_CONFBx.MODE=1 时, DMAC 工作于硬件 Burst 传输模式。该模式下当外设 DMA request 信号出现时, 则触发 DMAC 启动硬件 Burst 传输模式, 传输完成 (BC+1)*(TC+1)个数据后停止。该传输没有传输间隙, 所有数据传输完成之前, 当前正在工作的 DMA 通道将

一直占据总线，CPU 或更高优先 DMA 通道只能等待该操作完成后才能访问总线。建议每次不要传输太多的数据，否则将可能造成 CPU 在一段时间内完全不能工作。

DMA 支持的硬件 request 信号类型请参见 [DMAC_CONFx.TRI_SEL](#) 位功能描述。

硬件 Burst 传输示意如下所示，其中 SA 代表源地址，DA 代表目标地址；数据块大小为 3（BC=2），数据块数量为 3（TC=2），数据宽度为 8-bit，源地址自增，目标地址固定，传输完成自动禁止。

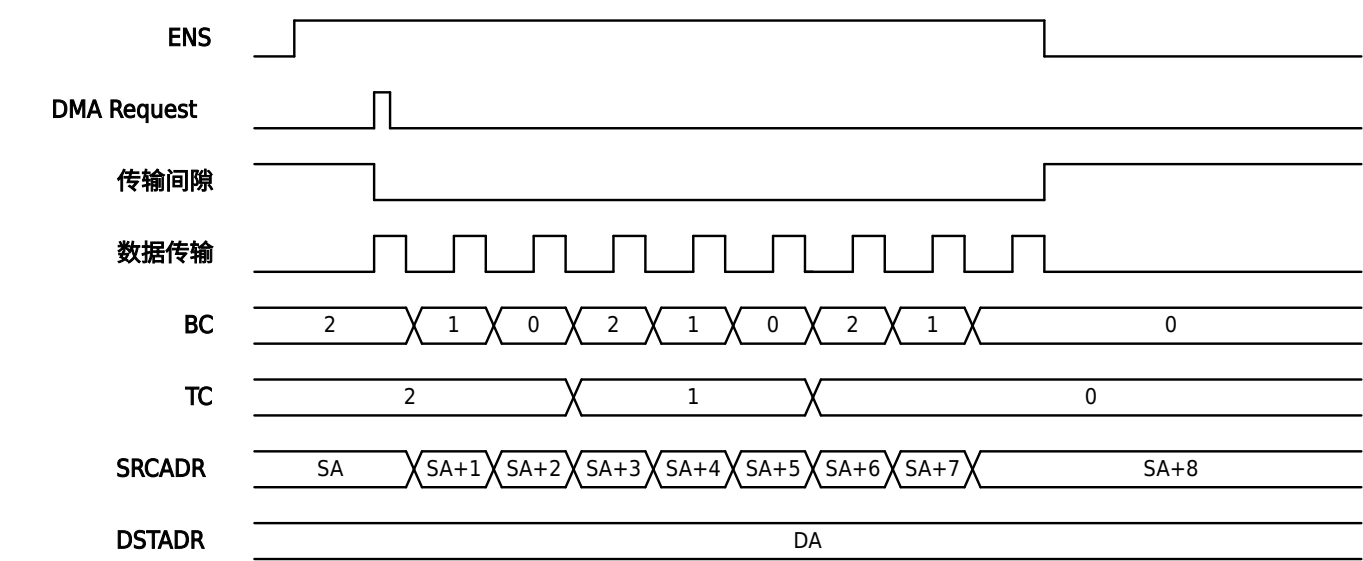


图 9-5 DMA 硬件 Burst 传输模式

9.3.3 DMA 传输暂停

软件 Block 传输模式时的传输暂停

当 DMAC 工作于软件 Block 传输模式时：

配置 DMAC_CONF.HALT 为非零值时，所有正在传输的通道都将在传输间隔时进入暂停状态，不再进行数据传输。配置 DMAC_CONF.HALT 为 0 后：之前正在传输的 DMAC 通道（DMAC_CONFx.ST 已经为 1）会继续刚才未完成的传输；之前未在传输的 DMAC 通道（DMAC_CONFx.ST 为 0），会等到 DMAC_CONFx.ST 为 1 后启动传输。

配置 DMAC_CONFx.PAS 为 1 时，x 相应的正在传输的通道将在传输间隔进入暂停状态，不再进行数据传输，配置 DMAC_CONFx.PAS 为 0 后，该通道会继续刚才未完成的传输。

软件 Burst 传输模式时的传输暂停

当 DMAC 工作于软件 Burst 传输模式且没有进行数据传输时：

配置 DMAC_CONF.HALT 为非零值，则 DMA 通道进入暂停状态，不再进行数据传输。如果在暂停期间某个 DMAC 通道软件触发使能（DMAC_CONFx.ST 为 1），则配置 DMAC_CONF.HALT 为 0 后，该通道会启动一次新的完整传输；如果在暂停期间某个 DMAC 通道软件触发未使能（DMAC_CONFx.ST 为 0），则配置 DMAC_CONF.HALT 为 0 后，待该通道软件触发使能（DMAC_CONFx.ST 为 1）后，会启动一次新的完整传输。

配置 DMAC_CONFx.PAS 为 1 时，则 DMA 通道进入暂停状态，不再进行数据传输。如果在暂停期间该通道软件触发使能（DMAC_CONFx.ST 为 1），则配置 DMAC_CONFx.PAS 为 0 后，该通道会启动一次新的完整传输；如果在暂停期间该通道软件触发未使能（DMAC_CONFx.ST 为 0），则配置 DMAC_CONFx.PAS 为 0 后，待该通道软件触发使能（DMAC_CONFx.ST 为 1）后，会启动一次新的完整传输。

当 DMAC 工作于软件 Burst 传输模式且正在进行数据传输时，向 DMAC_CONF.HALT 写入非 0 值或向 DMAC_CONFx.PAS 写入 1 均不能暂停正在进行的传输；当前传输完成后 DMA 通道也不会进入暂停态。

硬件 Block 传输模式时的传输暂停

当 DMAC 工作于硬件 Block 传输模式时：

配置 DMAC_CONF.HALT 为非零值时, 所有正在传输的通道都将在传输间隔时进入暂停状态, 不再进行数据传输。配置 DMAC_CONF.HALT 为 0 后, DMAC 各通道需要收到下一次触发信号才会继续刚才未完成的传输。配置 DMAC_CONFx.PAS 为 1 时, x 相应的正在传输的通道将在传输间隔进入暂停状态, 不再进行数据传输。配置 DMAC_CONFx.PAS 为 0 后, 该通道需要收到下一次触发信号才会继续刚才未完成的传输。

硬件 Burst 传输模式时的传输暂停

当 DMAC 工作于硬件 Burst 传输模式且没有进行数据传输时:

配置 DMAC_CONF.HALT 为非零值, 则 DMA 通道进入暂停状态, 不再进行数据传输。配置 DMAC_CONF.HALT 为 0 后, DMAC 收到下一次触发信号时会启动一次新的完整传输。

配置 DMAC_CONFx.PAS 为 1 时, 则 DMA 通道进入暂停状态, 不再进行数据传输。配置 DMAC_CONFx.PAS 为 0 后, DMAC 收到下一次触发信号时才会启动一次新的完整传输。

当 DMAC 工作于硬件 Burst 传输模式且正在进行数据传输时, 向 DMAC_CONF.HALT 写入非 0 值或向 DMAC_CONFx.PAS 写入 1 均不能暂停正在进行的传输; 当前传输完成后 DMA 通道也不会进入暂停态。

9.3.4 DMA 其它配置

数据宽度

配置 DMAC_CONFBx.WIDTH, 可以配置每个待传送的数据的宽度为 8-bit、16-bit、32-bit, 注意该位宽应与 DMA 的源地址与目标地址的位宽完全一致。

数据块大小

配置 DMAC_CONFx.BC, 可以配置每个数据块待传送的数据个数为 1~16。注意 BC 的值应当与源地址和目标地址所提供的连续操作次数相匹配。

- 如果源地址或目标地址一次只能提供或接收一个数据则 BC 只能配置为 0。例如 USART/LPUART/SPI 每次只能向 DMA 提供一个数据, USART/LPUART/SPI 每次只能从 DMA 接收一个数据, 所以 BC 只能配置为 0。
- 如果源地址和目标地址均能提供或接收多个数据, 则 BC 可以配置为更大的值, 以加快传输速度。例如从 FLASH 传输到 RAM 或从 FLASH 传输到 CRC, 源地址和目标地址都可以连续操作, 故可以配置 BC 为 15 以实现快速传输。

数据块个数

配置 DMAC_CONFx.TC, 可以配置每次 DMA 启动后, 传输出的数据块的个数为 1~65536。

通道优先级

- 当 DMAC_CONF.PRIO=0b0 时, DMAC 工作在固定优先模式下。此时 CH0 比 CH1 具有更高的优先级, 只有在高优先级通道不进行数据传输时, 低优先级才能进行数据传输。
- 当 DMAC_CONF.PRIO=0b1 时, DMAC 工作在循环优先模式下。此时 CH0 与 CH1 轮流具有更高的优先级, 两个通道将轮流占有总线以进行数据传输。

自动重载

如果使能自动重载功能则数据块大小、数据块个数、来源地址、目标地址在传输完成后均自动装载为上一次配置的数值, 不用每次进行配置。

9.3.5 DMA 中断

DMA 支持两种类型的中断: 传输出错和传输完成。当中断触发时, 通过查询状态寄存器 DMAC_CONFBx.STAT 就可以判断是什么条件触发了中断, 通过对寄存器 DMAC_CONFBx.STAT 清零就可以清除内置的中断状态标志位。

支持触发 DMA 中断的条件如下:

- 外设请求停止 DMA
- 访问传输来源地址出错

- 访问传输目的地址出错
- 传输完成

9.4 寄存器

9.4.1 寄存器总表

基地址：0x40021000

表 9-2 DMA 控制器 (DMAC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	DMAC_CONF	所有通道配置寄存器	32
0x10+0x10*x	DMAC_CONFAx	通道 x 配置 A 寄存器 (x=0~1)	32
0x14+0x10*x	DMAC_CONFBx	通道 x 配置 B 寄存器 (x=0~1)	32
0x18+0x10*x	DMAC_SRCADRx	通道 x 传输源地址寄存器 (x=0~1)	32
0x1C+0x10*x	DMAC_DSTADRx	通道 x 传输目的地址寄存器 (x=0~1)	32

9.4.2 所有通道配置寄存器 (DMAC_CONF)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EN	Reserved	PRI0	HALT	Reserved																											
Access	RW		RW	RW																												

位/位域	标记	功能描述	读写
31	EN	DMA 控制器使能 ● 0b0：禁止 ● 0b1：使能  注意 当禁止 DMA 控制器时，将强制停止正在进行的传输，造成不可预知的结果。	RW
30:29	Reserved	-	-
28	PRI0	DMA 通道优先级配置 ● 0b0：固定优先级模式，CH0>CH1 优先级 ● 0b1：循环优先级模式，CH0/CH1 轮流获取总线访问权	RW
27:24	HALT	DMA 全部通道传输暂停控制 ● 0b0000：恢复所有通道数据传输 ● 0b0001~0b1111：暂停所有通道数据传输	RW
23:0	Reserved	-	-

9.4.3 通道 x 配置 A 寄存器 (DMAC_CONFx) (x=0~1)

偏移地址 (Offset) : 0x10+0x10*x

Offset	Bit Position																																		
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	ENS	PAS	ST	TRI_SEL								Reserved				BC				TC															
Access	RW	RW	RW	RW												RW				RW															

位/位域	标记	功能描述	读写
31	ENS	DMA 通道使能控制 0b0: 禁止当前 DMA 通道 0b1: 使能当前 DMA 通道; 若 CONFBx.MSK 为 0, 则传输完成时该位自动清零  注意 - 当禁止 DMA 控制器时, 将强制停止正在进行的传输, 造成不可预知的结果。 - DMA 通道使能, 建议放到所有 DMA 配置完毕之后进行使能操作。	RW
30	PAS	DMA 通道数据传输暂停控制 0b0: 恢复 DMA 通道数据传输 0b1: 暂停 DMA 通道数据传输	RW
29	ST	DMA 通道软件触发控制 0b0: 停止 DMA 通道软件 Block/Burst 传输 0b1: 触发 DMA 通道启动软件 Block/Burst 传输, 传输完成时该位自动清零	RW
28:22	TRI_SEL	DMA 通道触发源配置 0x00: 软件触发 0x40: SPI0 接收 Buf 非空 0x41: SPI0 发送 Buf 空 0x42: SPI1 接收 Buf 非空 0x43: SPI1 发送 Buf 空 0x44: ADC JQR 转换完成 0x45: ADC SQR 转换完成 0x46: LCD 定时中断 0x47: 禁止设置 0x48: USART0 接收 Buf 非空 0x49: USART0 发送 Buf 空 0x4A: USART1 接收 Buf 非空 0x4B: USART1 发送 Buf 空 0x4C: LPUART0 接收 Buf 非空 0x4D: LPUART0 发送 Buf 空 0x4E: LPUART1 接收 Buf 非空 0x4F: LPUART1 发送 Buf 空 0x50: LPUART2 接收 Buf 非空	RW

位/位域	标记	功能描述	读写
28:22	TRI_SEL	0x51: LPUART2 发送 Buf 空 0x52: TIM0 通道 A, 发生捕获事件或比较事件 0x53: TIM0 通道 B, 发生捕获事件或比较事件 0x54: TIM1 通道 A, 发生捕获事件或比较事件 0x55: TIM1 通道 B, 发生捕获事件或比较事件 0x56: TIM2 通道 A, 发生捕获事件或比较事件 0x57: TIM2 通道 B, 发生捕获事件或比较事件 0x58: TIM3 通道 A, 发生捕获事件或比较事件 0x59: TIM3 通道 B, 发生捕获事件或比较事件 0x5A: HSI2C 主机或者从机接收 Buf 非空 0x5B: HSI2C 主机或者从机发送 Buf 空 - 其他: 禁止设置	RW
21:20	Reserved	-	-
19:16	BC	DMA 通道传输数据块大小 配置 DMA 通道待传输的数据块的大小为 BC+1  说明 当源地址或目标地址每隔一段时间才能提供或接收一个数据时, BC 值只可设为 0。	RW
15:0	TC	DMA 通道传输数据块数量 配置 DMA 通道待传输的数据块的数量为 TC+1 传输过程中, 每传输完成一个数据块, 该寄存器值自减 1  说明 DMA 传输的总数据量为(TC + 1)*(BC + 1)。	RW



说明

TIM0/1/2/3 的触发源:

- TIM0/1/2/3 通道 A: 比较捕获 A、外部触发
- TIM0/1/2/3 通道 B: 比较捕获 B、事件更新

9.4.4 通道 x 配置 B 寄存器 (DMAC_CONFBx) (x=0~1)

偏移地址 (Offset) : 0x14+0x10*x

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	MODE		WIDTH		FS	FD	RC	RS	RD	ERR_IE	FIS_IE	STAT	Reserved																MSK		
Access		RW		RW		RW	RW	RW	RW	RW	RW	RW	RW																		RW	

位/位域	标记	功能描述	读写
31:30	Reserved	-	-
29:28	MODE	DMA 通道传输模式配置 0b00: Block 传输 0b01: Burst 传输 0b10: 禁止设置 0b11: 禁止设置	RW
27:26	WIDTH	DMA 通道传输数据宽度配置 0b00: 8-bit 0b01: 16-bit 0b10: 32-bit 0b11: 禁止设置	RW
25	FS	DMA 通道源地址自增使能 0b0: 每传输完成一个数据, 源地址进行自增 0b1: 源地址在传输过程中保持不变  说明 数据宽度为 8-bit/16-bit/32-bit 所对应的自增量分别为 1/2/4。	RW
24	FD	DMA 通道目标地址自增使能 0b0: 每传输完成一个数据, 目标地址进行自增 0b1: 目标地址在传输过程中保持不变  说明 数据宽度为 8-bit/16-bit/32-bit 所对应的自增量分别为 1/2/4。	RW
23	RC	DMA 通道 BC 和 TC 重载使能 0b0: 禁止 BC/TC 重载, 传输完成后 BC/TC 的值均为 0 0b1: 使能 BC/TC 重载, 传输完成后 BC/TC 的值自动恢复为写入的初值	RW
22	RS	DMA 通道源地址重载使能 0b0: 禁止源地址重载, 传输完成后源地址的值不确定 0b1: 使能源地址重载, 传输完成后源地址的值为写入的初值	RW
21	RD	DMA 通道目标地址重载使能 0b0: 禁止目标地址重载, 传输完成后目标地址的值不确定 0b1: 使能目标地址重载, 传输完成后目标地址的值自动恢复为写入的初值	RW

位/位域	标记	功能描述	读写
20	ERR_IE	DMA 通道传输错误中断使能 ● 0b0: 禁止 ● 0b1: 使能  说明 在中断服务程序中，需要对 STAT 写入 0x00 以清除中断状态。	RW
19	FIS_IE	DMA 通道传输完成中断使能 ● 0b0: 禁止 ● 0b1: 使能  说明 在中断服务程序中，需要对 STAT 写入 0x00 以清除中断状态。	RW
18:16	STAT	DMA 通道当前传输状态 ● 0b000: 初始值 ● 0b001: 禁止设置 ● 0b010: 传输错误，外设请求停止 DMA ● 0b011: 传输错误，访问传输来源地址出错 ● 0b100: 传输错误，访问传输目的地址出错 ● 0b101: 传输完成 ● 0b110: 禁止设置 ● 0b111: 传输暂停	RW
15:1	Reserved	-	-
0	MSK	DMA 通道传输完成时自动禁止/使能配置 ● 0b0: DMA 通道传输完成时，CONFAX.ENS 自动清零 ● 0b1: DMA 通道传输完成时，CONFAX.ENS 保持不变	RW

9.4.5 通道 x 传输源地址寄存器 (DMAC_SRCADR_x) (x=0~1)

偏移地址 (Offset) : 0x18+0x10*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	SRCADR																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	SRCADR	DMA 通道源地址配置 可通过 CONFB _x .RS 配置其自动重载  注意 地址对齐方式应与数据宽度相匹配，否则会造成地址访问出错。	RW

9.4.6 通道 x 传输目的地址寄存器 (DMAC_DSTADR_x) (x=0~1)

偏移地址 (Offset) : 0x1C+0x10*x

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DSTADR																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	DSTADR	DMA 通道目标地址配置 可通过 CONFB _x .RD 配置其自动重载  注意 地址对齐方式应与数据宽度相匹配，否则会造成地址访问出错。	RW

10 时钟校准模块 (CTRIM)

10.1 简介

该模块工作于校准模式时可自动实时校准 RCH/RCL 的输出频率，使 RCH/RCL 输出频率的精度不再受环境变化影响。该模块工作于定时器模式时，具备通用定时功能且在 DeepSleep 模式下仍可正常工作。

10.2 主要特性

时钟校准支持的特性如下：

- 校准模式：支持 RCH/RCL 自动实时校准/单次校准
- 校准模式：多种校准步长，快速逼近目标
- 定时模式：DeepSleep 模式下可工作
- 定时模式：支持自动唤醒
- 定时模式：多种时钟信号可供选择
- 定时模式：翻转输出

10.3 功能说明

10.3.1 RCH 时钟校准模式

当配置 CTRIM_CR1.MD 为 0b00 时，该模块工作于 RCH 校准模式。该模式可自动实时校准 RCH 的输出频率，使 RCH 输出频率的精度不再受环境变化的影响。校准 RCH 时，需要向该模块提供一个精准的低频时钟信号；该精准的低频时钟信号可以是片内的 PCLK，也可以通过外部 ETR 管脚输入。注意，RCLK 频率应小于 RCH 的频率的 1/2。

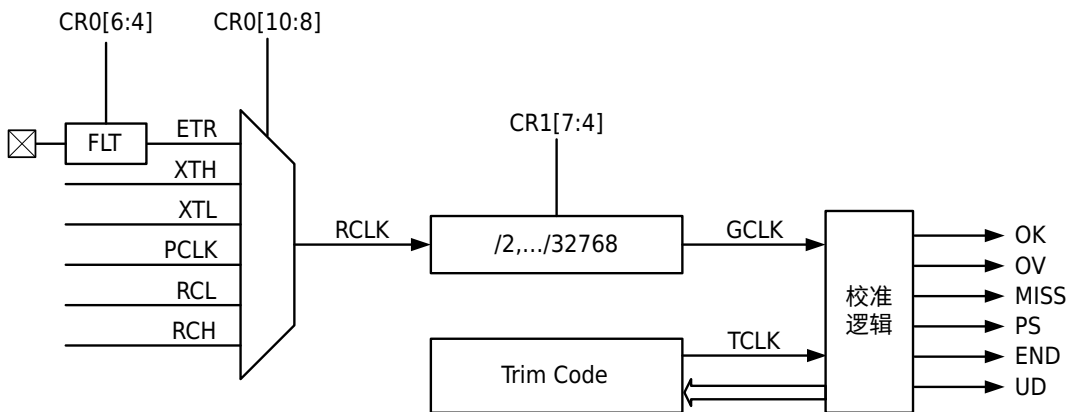


图 10-1 RCH 时钟校准时工作框图

当自动校准开始工作时，误差计数器在每个 GCLK 的周期内对 TCLK 信号（RCH 振荡频率）从 CTRIM_ARR 开始进行递减计数，计到 0 后开始递增计数。若 GCLK 周期内的误差计数器的计数值小于 CTRIM_FLIM 寄存器的值则置位 CTRIM_ISR.OK 以标识 RCH 输出频率已达到预期；若 GCLK 周期内的误差计数器的计数值大于 CTRIM_FLIM 寄存器的值则自动调整 RCH 的 TrimCode 并进行下一轮计数。自动调整规则为：若计数方向为减计数则对 RCH 的 TrimCode 增加 CTRIM_CR0.STEP；若计数方式为加计数则对 RCH 的 TrimCode 减少 CTRIM_CR0.STEP。校准逻辑会自动减小 CTRIM_CR0.STEP 的数值以实现 RCH 输出的频率逐次逼近目标值。当误差计数器计数值递减为零时，生成 UD 事件；当误差计数器计数值递加到 CTRIM_ARR 时，停止计数并生成 MISS 事件。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

● RCH 单次校准

设置合适的 STEP、初值、FLIM，方法同上。配置 CTRIM_CR1.OST=1，对 RCH 进行单次校准，STEP 变为 1 时结束。

10.3.2 RCL 时钟校准模式

当配置 CTRIM_CR1.MD 为 0b01 时，该模块工作于 RCL 校准模式。该模式可自动实时校准 RCL 的输出频率，使 RCL 输出频率的精度不再受环境变化的影响。校准 RCL 时，需要向该模块提供一个精准的高频时钟信号；该精准的高频时钟信号可以是片内的 PCLK，也可以通过外部 ETR 管脚输入。

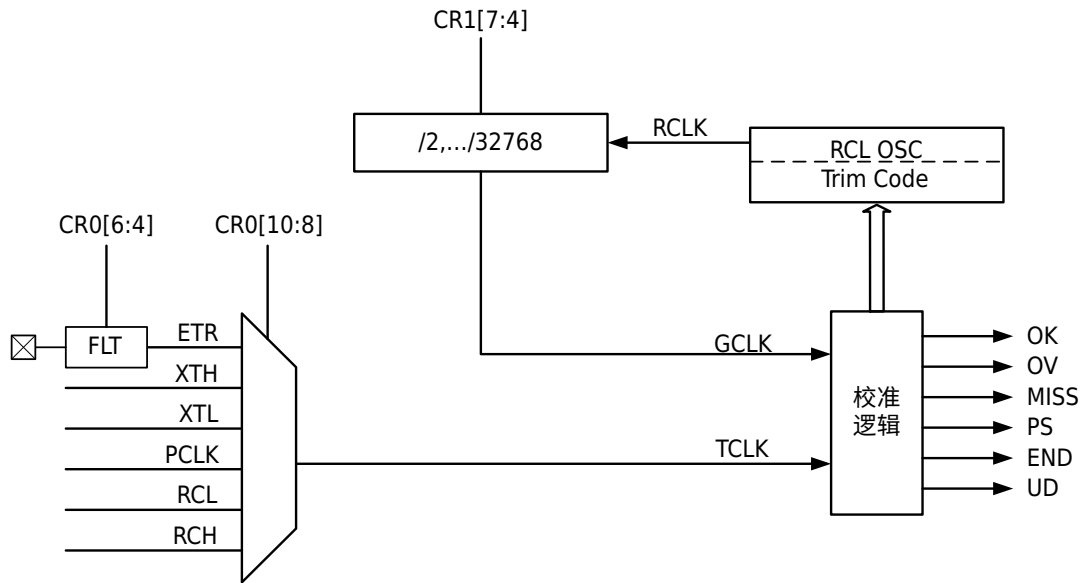


图 10-2 RCL 时钟校准时工作框图

当自动校准开始工作时，误差计数器在每个 GCLK（分频后的 RCL 振荡频率）的周期内对 TCLK 时钟信号从 CTRIM_ARR 开始进行递减计数，计到 0 后开始递增计数。若 GCLK 周期内的误差计数器的计数值小于 CTRIM_FLIM 寄存器的值则置位 CTRIM_ISR.OK 以标识 RCL 输出频率已达到预期；若 GCLK 周期内误差计数器的计数值大于 CTRIM_FLIM 寄存器的值则自动调整 RCL 的 TrimCode 并进行下一轮计数。自动调整规则为：若计数方向为减计数则对 RCL 的 TrimCode 减少 CTRIM_CR0.STEP；若计数方式为加计数则对 RCL 的 TrimCode 增加 CTRIM_CR0.STEP。校准逻辑会自动减小 CTRIM_CR0.STEP 的数值以实现 RCL 输出的频率逐次逼近目标值。

当误差计数器计数值递减为零时，生成 UD 事件；当误差计数器计数值递加到 CTRIM_ARR 时，停止计数并生成 MISS 事件。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

● RCL 单次校准

设置合适的 STEP、初值、FLIM。配置 CTRIM_CR1.OST=1，对 RCL 进行单次校准，STEP 变为 1 时结束。

10.3.3 低功耗定时器模式

当配置 CTRIM_CR1.MD 为 0b11 时，该模块工作于通用低功耗定时器模式。该模式下 CTRIM_CR0.SRC 需配置为低功耗模式下可用的时钟。该模式下 CTRIM 具有通用定时器的定时及中断功能，若定时器的时钟为 RCL 则其在 DeepSleep 模式下可正常工作，支持自动唤醒。

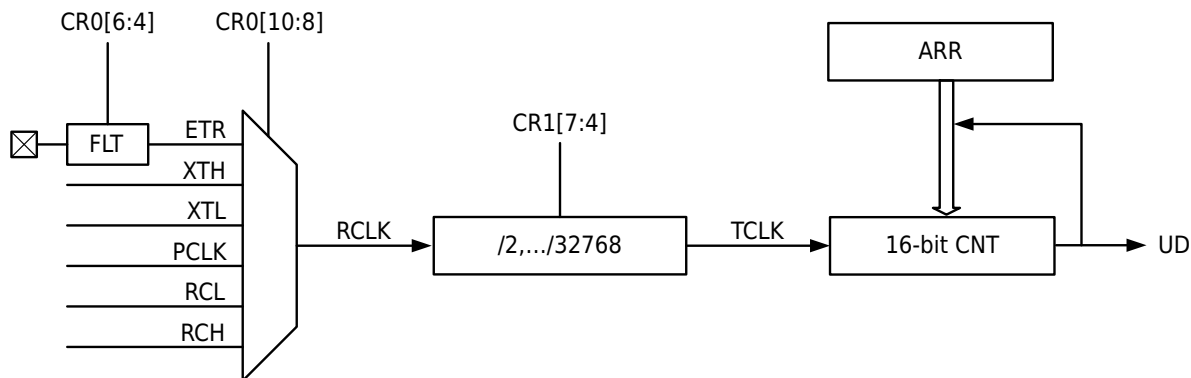


图 10-3 通用低功耗定时器场景工作框图

定时器计数值从 ARR 减计数，当计数到 0 后产生下溢出，如果使能中断会产生下溢出中断，计数周期为 $ARR+1$ 。

定时器每溢出一次 TOG 输出电平翻转一次；用户可通过 GPIO 的端口复用功能将 TOG 信号输出到管脚用于驱动外部电路。

10.4 操作示例

10.4.1 RCH 时钟自动校准示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节管脚数字复用功能的相关描述，配置相应管脚功能为 CTRIM.ETR，并向其提供 2kHz 的方波。
- 步骤 2. 设置 CTRIM_CR1.MD 为 0，选择 RCH 时钟校准模式。
- 步骤 3. 设置 CTRIM_CR0.SRC 为 0，外部 ETR 管脚输入信号作 RCLK。
- 步骤 4. 设置 CTRIM_CR1.PRS 为 1，对 RCLK 进行 2 分频，GCLK 频率为 1kHz。
- 步骤 5. 配置 CTRIM_ARR 为 47999，一个 GCLK 的计数周期内目标计数值为 48000。



说明

ETR 管脚输入信号为 2kHz 方波，GCLK 的周期为 1ms。1ms 内误差计数器对 RCH 输出频率的理论计数值应为 48000。

- 步骤 6. 设置 CTRIM_FLIM 为 240，选择 RCH 校正精度为 0.5%。
 $48000 \times 0.005 = 240$ 。
- 步骤 7. 设置 CTRIM_CR0.STEP 为 0，选择初始校准步长为 16。
- 步骤 8. 设置 CTRIM_CR1.AUTO 为 1，使能自动校准。
- 步骤 9. 设置 CTRIM_CR1.EN 为 1，启动时钟校准。
- 步骤 10. 等待 CTRIM_ISR.END 为 1，自动校准完成。

10.4.2 RCL 时钟自动校准示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节管脚数字复用功能的相关描述，配置相应管脚功能为 CTRIM.ETR，并向其提供 1MHz 的方波。
- 步骤 2. 设置 CTRIM_CR1.MD 为 1，选择 RCL 时钟校准模式。
- 步骤 3. 设置 CTRIM_CR0.SRC 为 0，外部 ETR 管脚输入信号作 TCLK。
- 步骤 4. 设置 CTRIM_CR1.PRS 为 0x0A，对 RCLK (RCL 输出频率) 进行 1024 分频，GCLK 目标频率为 32Hz。
- 步骤 5. 配置 CTRIM_ARR 为 31249，一个 GCLK 的计数周期内目标计数值为 31250。

**说明**

GCLK 的目标周期为 31.25ms，该时间内误差计数器对 TCLK (1MHz) 的理论计数值应为 31250。

- 步骤 6.** 设置 CTRIM_FLIM 为 156，选择 RCL 校正精度为 0.5%。
 $31250 \times 0.005 = 156$ 。
- 步骤 7.** 设置 CTRIM_CR0.STEP 为 0，选择初始校准步长为 16。
- 步骤 8.** 设置 CTRIM_CR1.AUTO 为 1，使能自动校准。
- 步骤 9.** 设置 CTRIM_CR1.EN 为 1，启动时钟校准。
- 步骤 10.** 等待 CTRIM_ISR.END 为 1，自动校准完成。

10.5 寄存器

10.5.1 寄存器总表

基地址：0x40009000

表 10-1 时钟校准模块 (CTRIM) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	CTRIM_ARR	自动装载寄存器	32
0x04	CTRIM_CNT	误差计数器寄存器	32
0x0C	CTRIM_CR0	控制寄存器 0	32
0x10	CTRIM_CR1	控制寄存器 1	32
0x14	CTRIM_IER	中断使能寄存器	32
0x18	CTRIM_ISR	中断和状态寄存器	32
0x1C	CTRIM_ICR	中断标志清除寄存器	32
0x20	CTRIM_FCAP	误差结果寄存器，GCLK 周期结束时计数器的值	32
0x24	CTRIM_TVAL	TRIM 结果寄存器	32
0x28	CTRIM_FLIM	误差上限寄存器	32

10.5.2 自动装载寄存器 (CTRIM_ARR)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																ARR															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	ARR	误差计数器重载值	RW

10.5.3 误差计数器寄存器 (CTRIM_CNT)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNT															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CNT	误差计数器计数值	RO

10.5.4 控制寄存器 0 (CTRIM_CR0)



说明

该寄存器只有在 CTRIM_CR1.EN 为 0 时才可以更改。

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																						RW SRC		Reserved	RW ETRFLT		Reserved	RW STEP			
Access																																

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10:8	SRC	时钟来源选择 ● 0b000: ETR ● 0b001: XTH ● 0b010: XTL ● 0b011: PCLK ● 0b100: RCL ● 0b101: RCH ● 0b110: 禁止设置 ● 0b111: 禁止设置	RW
7	Reserved	-	-
6:4	ETRFLT	外部输入管脚 (ETR) 滤波配置 ● 0b000: 无滤波 ● 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ ● 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ ● 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ ● 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ ● 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ ● 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ ● 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
3	Reserved	-	-
2:0	STEP	TRIM 初始步进量配置 ● 0b000: 0x10 ● 0b001: 0x02 ● 0b010: 0x08 ● 0b011: 0x20 ● 0b100: 0x80 ● 0b101: 0x100 ● 0b110: 0x200 ● 0b111: 0x400	RW

10.5.5 控制寄存器 1 (CTRIM_CR1)



说明

该寄存器只有在 CTRIM_CR1.EN 为 0 时才可以更改。

Offset	Bit Position																																
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																							OST		PRS			AUTO		MD		EN
Access																								RW	RW			RW	RW		RW		

位/位域	标记	功能描述	读写
31:9	Reserved	-	-
8	OST	校准模式配置 0b0: 实时校准模式, TRIM 过程持续进行 0b1: 单次校准模式, STEP 变为 1 时结束	RW
7:4	PRS	参考时钟预分频配置 0b0000: 禁止设置 0b0001: 2 分频 0b0010: 4 分频 0b0011: 8 分频 0b0100: 16 分频 0b0101: 32 分频 0b0110: 64 分频 0b0111: 128 分频 0b1000: 256 分频 0b1001: 512 分频 0b1010: 1024 分频 0b1011: 2048 分频 0b1100: 4096 分频 0b1101: 8192 分频 0b1110: 16384 分频 0b1111: 32768 分频	RW
3	AUTO	RCH/RCL 自动 TRIM 使能配置 0b0: 禁止 0b1: 使能 说明 实时校准模式下, 该位禁止配置为 0。	RW
2:1	MD	工作模式配置 0b00: RCH 校准模式, CR0.SRC 需选择低速精准时钟 0b01: RCL 校准模式, CR0.SRC 需选择高速精准时钟 0b10: 禁止设置 0b11: 定时器模式	RW

位/位域	标记	功能描述	读写
0	EN	定时器使能控制 ● 0b0：禁止 ● 0b1：使能	RW

10.5.6 中断使能寄存器 (CTRIM_IER)

Offset	Bit Position																																	
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																																	
Access																													RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5	OK	自动 Trim 结果在范围内中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
4	OV	TrimCode 溢出中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
3	MISS	计数失败中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
2	PS	误差计数器完成中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
1	END	自动 Trim 结束中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
0	UD	误差计数器下溢中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW

10.5.7 中断和状态寄存器 (CTRIM_ISR)

Offset	Bit Position																																					
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																																					
Access																											RO	DIR	RO	OK	RO	OV	RO	MISS	RO	PS	RO	END

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	DIR	误差计数器计数方向 0b0: 递增计数, RCH 输出频率高于目标频率, RCL 输出频率低于目标频率 0b1: 递减计数, RCH 输出频率低于目标频率, RCL 输出频率高于目标频率	RO
5	OK	Trim 精度标志 0b0: Trim 完成时精度不满足 FLIM 要求 0b1: Trim 完成时精度满足 FLIM 要求	RO
4	OV	TrimCode 溢出标志 0b0: RCH/RCL 的 TrimCode 不为 0x000 或 0x1FF 0b1: RCH/RCL 的 TrimCode 已调整为 0x000 或 0x1FF	RO
3	MISS	计数失败标志 0b0: 在 GCLK 的一个周期内, 误差计数器未计数到 ARR 0b1: 在 GCLK 的一个周期内, 误差计数器已计数到 ARR	RO
2	PS	误差计数器完成标志 0b0: 误差计数器正在计数中 0b1: 误差计数器已完成一次计数并停止	RO
1	END	自动 Trim 结束标志 0b0: 自动 Trim 进行中 0b1: 自动 Trim 已完成	RO
0	UD	误差计数器下溢标志 0b0: 误差计数器未计数到 0 0b1: 误差计数器已计数到 0	RO

10.5.8 中断标志清除寄存器 (CTRIM_ICR)

Offset	Bit Position																																							
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x0000003F																																							
Name	Reserved																																							
Access																																								

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5	OK	清除校准成功 OK 标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
4	OV	清除 TrimCode 溢出标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
3	MISS	清除计数失败标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
2	PS	清除误差计数器完成标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
1	END	清除自动 Trim 结束标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
0	UD	清除误差计数器下溢标志 写 0 清除标志, 写 1 无效; 读出恒为 1	WO

10.5.9 误差结果寄存器 (CTRIM_FCAP)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF																															
Name	Reserved																RO FCAP															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	FCAP	频率误差捕获值	RO

10.5.10 TRIM 结果寄存器 (CTRIM_TVAL)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000100																															
Name	Reserved																TVAL RO															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	TVAL	自动 TRIM 结果  说明 1. 自动 TRIM 结束需要将 TVAL 的值写入相应的 TRIM 寄存器中。 2. 如在 TRIM 过程中软件读，需要读两次相同有效。	RO

10.5.11 误差上限寄存器 (CTRIM_FLIM)

Offset	Bit Position																															
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000010																															
Name	Reserved																FLIM															
Access																																

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	FLIM	误差允许值	RW

11 系统定时器 (SYSTICK)

11.1 简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询计数标志。

11.2 操作示例

11.2.1 设置 SysTick

背景信息

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程。

操作步骤

- 步骤 1. 配置 SysTick->CTRL.ENABLE 为 0，禁止 SysTick。
 - 步骤 2. 配置 SysTick->CTRL.CLKSOURCE，选择 SysTick 的时钟源。
 - 步骤 3. 配置 SysTick->LOAD，选择 SysTick 的溢出周期。
 - 步骤 4. 向 SysTick->VAL 写入任意值，清零 SysTick->VAL 及 SysTick->CTRL.COUNTFLAG。
 - 步骤 5. 配置 SysTick->CTRL.TICKINT 为 1，使能 SysTick 中断。
 - 步骤 6. 配置 SysTick->CTRL.ENABLE 为 1，使能 SysTick。
 - 步骤 7. 在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。
- Systick 溢出周期为 SysTick->LOAD+1，配置示例如下：

时钟源	SysTick->LOAD	溢出周期
HCLK 4MHz	3999	1ms
XTL 32.768kHz	327	10.01ms

11.3 寄存器

11.3.1 寄存器总表

基地址：0xE000E010

表 11-2 SysTick 定时器寄存器偏移地址

偏移地址	寄存器	CMSIS 符号	描述	位宽
0x00	SYS_CSR	SysTick->CTRL	SysTick 控制和状态寄存器	32
0x04	SYS_RVR	SysTick->LOAD	SysTick 重装载寄存器	32
0x08	SYS_CVR	SysTick->VAL	SysTick 当前值寄存器	32
0x0C	SYS_CALIR	SysTick->CALIB	SysTick 校准值寄存器	32

11.3.2 SysTick 控制和状态寄存器 (CTRL)

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																COUNTFLAG	Reserved														CLKSOURCE	TICKINT	ENABLE
Access																	RO															RW	RW	RW

位/位域	标记	功能描述	读写
31:17	Reserved	-	-
16	COUNTFLAG	SysTick 定时器溢出标志 读该寄存器可清除 COUNTFLAG 标志 0b0: SysTick 定时器未发生溢出 0b1: SysTick 定时器发生下溢出	RO
15:3	Reserved	-	-
2	CLKSOURCE	SysTick 时钟源选择 0b0: 使用参考时钟 (XTL) 0b1: 使用内核时钟 (HCLK)  说明 当选择参考时钟作为 SysTick 的时钟源时，需要同步使能 XTL 时钟源。	RW
1	TICKINT	SysTick 中断使能 0b0: 禁止 0b1: 使能	RW
0	ENABLE	SysTick 定时器使能 0b0: 禁止 0b1: 使能	RW

11.3.3 SysTick 重载寄存器 (LOAD)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved								RW RELOAD																							
Access									RW																							

位/位域	标记	功能描述	读写
31:24	Reserved	-	-
23:0	RELOAD	SysTick 定时器重载值	RW

11.3.4 SysTick 当前值寄存器 (VAL)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved								CURRENT																							
Access																																

位/位域	标记	功能描述	读写
31:24	Reserved	-	-
23:0	CURRENT	SysTick 定时器的当前计数值 读寄存器获取 SysTick 定时器的当前计数值 写任意值到寄存器将会清零该寄存器及 COUNTFLAG	RW

11.3.5 SysTick 校准值寄存器 (CALIB)

Offset	Bit Position																																	
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0xFFFFFFFF																																	
Name	RO NOREF	RO SKEW	Reserved								TENMS																							
Access																																		

位/位域	标记	功能描述	读写
31	NOREF	SysTick 计数时钟是否支持参考时钟标志 ● 0b0: 支持使用参考时钟 (XTL) ● 0b1: 不支持使用参考时钟 (XTL)  说明 当该位为 1 时，CTRL.CLKSOURCE 必须设置为 1。	RO
30	SKEW	TENMS 精度指示 ● 0b0: TENMS 值代表精确的 10ms ● 0b1: TENMS 值代表粗略的 10ms	RO
29:24	Reserved	-	-
23:0	TENMS	10 毫秒校准值	RO

12 低功耗定时器 (LPTIM)

12.1 简介

LPTimer 是异步 16 位定时/计数器，在系统时钟关闭后仍然可以通过内部低速 RC、外部低速晶体振荡以及 ETR 外部输入计时/计数。通过中断在低功耗模式下唤醒系统。

本模块包含两个低功耗定时器，可以通过内部进行级联。

12.2 主要特性

- 16 位异步计数/定时器
- 可选计数/定时时钟源：RCL/PCLK/XTL/ETR
- 4 位分频配置寄存器，最大支持 32768 预分频
- PWM 脉冲输出
- ARR 重载可配置
- 溢出中断唤醒

12.3 功能说明

12.3.1 功能框图

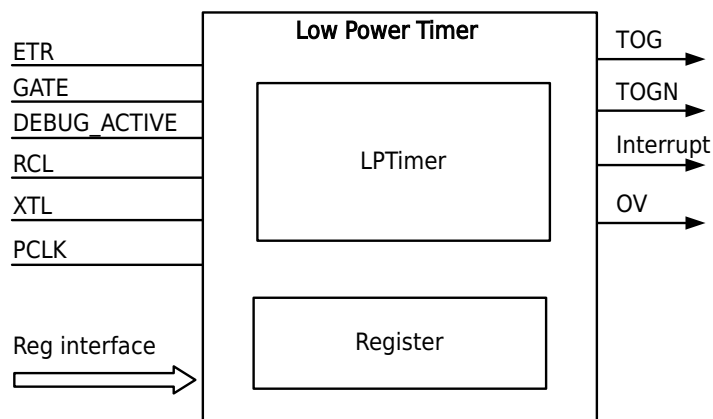


图 12-1 LPTimer 框图

12.3.2 工作模式

LPTimer 支持 2 种工作模式，每个定时器/计数器都有独立的控制启动信号，以及外部输入时钟，门控信号。

LPTimer 使用 ETR、GATE 来进行计数功能：

- ETR 用于计数器的外部输入时钟信号，也可以用于内部时钟计数。
- GATE 用于有效电平计数使能信号。

LPTimer 的定时器支持两种工作模式，通过设置 CR.MD 选择工作模式：模式 1 为 16 位自由计数模式，模式 2 是 16 位重载模式。

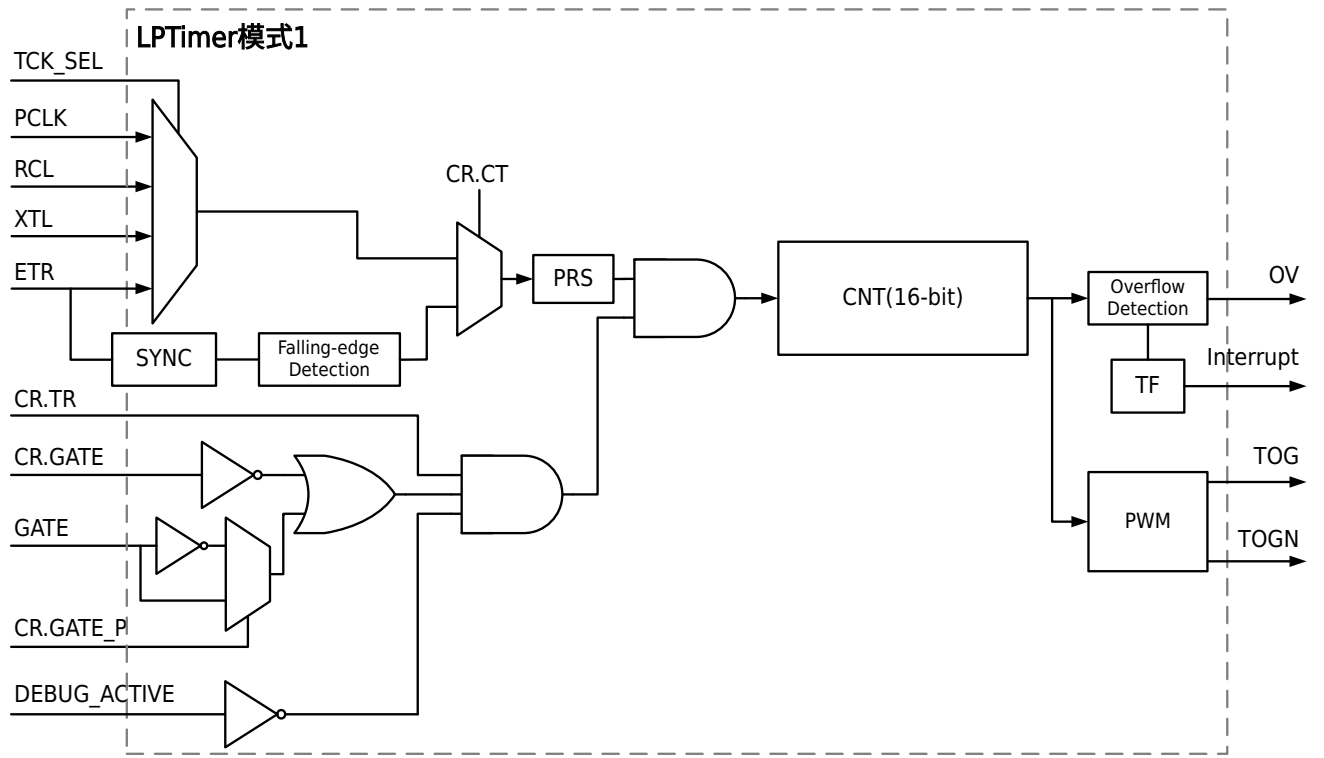


图 12-2 LPTimer 模式 1

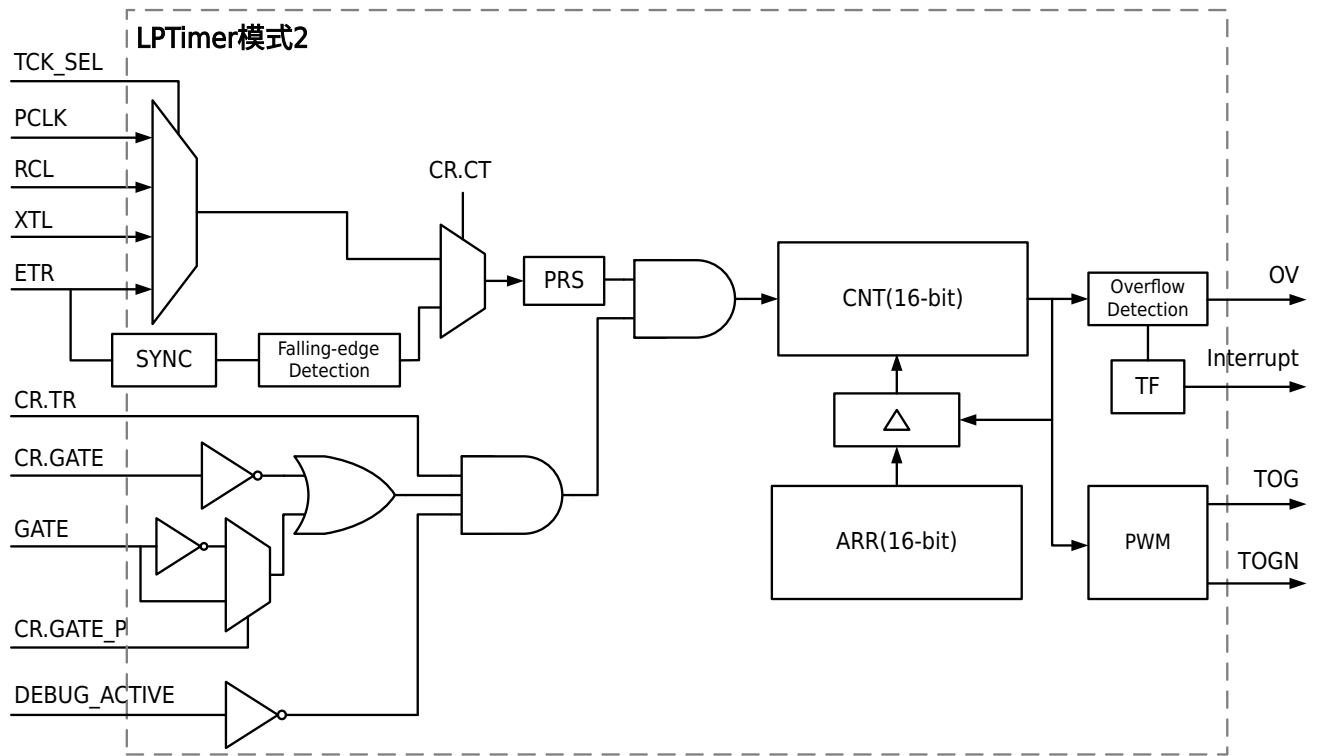


图 12-3 LPTimer 模式 2

LPTimer 启动时会自动装载重载寄存器 ARR 的值到计数器中。LPTimer 可选四种时钟作为定时器时钟，通过控制寄存器 CR.TCK_SEL 选择（默认选择 PCLK）。时钟选择如表：

TCK_SEL	定时器时钟	读定时器计数值
0b00	PCLK	读经过同步

TCK_SEL	定时器时钟	读定时器计数值
0b01	ETR	读经过同步
0b10	XTL	读经过同步
0b11	RCL	读经过同步

当设置对应定时器 TR 为 1 后定时器开始运行。计数器从设定的值开始计数，计数到最大 0xFFFF 后产生溢出中断。模式 1 产生中断后计数器清零继续向上计数，模式 2 产生中断后重载寄存器 ARR 的值到计数器中，向上计数。计数器 CNT 的初值在启动定时器时由 ARR 自动装载。

 **说明**

当配置 CR.CT=1 对 ETR 下降沿事件计数，此时计数时钟源不可以选择 ETR。

12.3.3 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的计数时钟采样，因此外部输入时钟频率不能超过系统的计数时钟。计数到最大值会溢出并且产生中断。

12.3.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器一个时钟累加一次，计数到最大值会溢出并且产生中断。

定时时钟周期数为(0xFFFF-ARR+0x0001)（开启重载模式时）。

12.3.5 PWM 脉冲输出

通过设定 LPTIM_CMP 的数值可以调整占空比：

- 开启重载模式（LPTIM_CR.MD=1）时，占空比=(0xFFFF-CMP)/(0xFFFF-ARR+0x0001)。
- 关闭重载模式（LPTIM_CR.MD=0）时，除定时器开启后第一个周期占空比=(0xFFFF-CMP)/(0xFFFF-ARR+0x0001)外，其他占空比=(0xFFFF-CMP)/(0xFFFF+0x0001)。

下图以定时器模式关闭 GATE 功能举例（LPTIM_CR.CT=0&LPTIM_CR.GATE=0）。TOG 会在达到 CMP 值后下个有效计数沿置起，计数器计数到最大值后清 0。其中，TOGN 为 TOG 信号取反逻辑。

 **说明**

使用内部时钟对 ETR 进行计数时，PWM 精度存在偏差可能性，与 ETR 频率&相位相关。

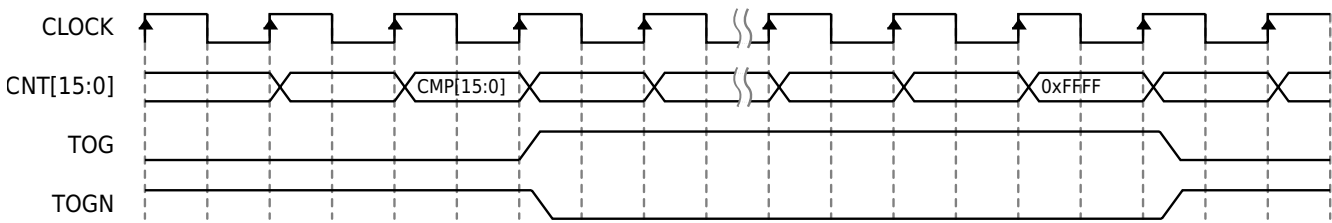


图 12-4 PWM 脉冲输出示意图

12.3.6 GATE 互联

当 LPTIMx_G=0b000 时，门控 GATE 输入为 Pxy_SEL 选择的端口输入；当 LPTIMx_G=0b001~0b111 时，连接其他模块的输入或输出。详细信息请参见《端口控制器（GPIO）》章节寄存器“端口辅助功能定时器门控选择（GPIO_TIMGS）”、“端口辅助功能定时器捕获输入选择 1（GPIO_TIMCPS1）”和“端口 Pxy 功能配置寄存器（Pxy_SEL）”相关描述。

12.3.7 ETR 互联

当 LPTIMx_E=0b000 时，ETR 输入为 Pxy_SEL 选择的端口输入；当 LPTIMx_E=0b001~0b111 时，连接其他模块的输入或输出。详细信息请参见《端口控制器 (GPIO)》章节寄存器“端口辅助功能定时器 ETR 选择 (GPIO_TIMES)”、“端口辅助功能定时器捕获输入选择 1 (GPIO_TIMCPS1)”和“端口 Pxy 功能配置寄存器 (Pxy_SEL)”相关描述。

设置 LPTIM1 定时器的外部输入 LPTIM1_E 为 0b101，选择 LPTIM0 的溢出输出 LPTIM0_OV 信号从芯片内部输出到 LPTIM1，作为 LPTIM1 的外部时钟输入，LPTIM1 设置为外部计数模式。这样两个定时器就可以级联使用。

12.4 寄存器

12.4.1 寄存器总表

- LPTimer0 基地址：0x40000C00
- LPTimer1 基地址：0x40003C00

表 12-2 低功耗定时器 (LPTIM) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	LPTIM_CNT	LPTimer 计数器计数值寄存器	32
0x004	LPTIM_ARR	LPTimer 重载寄存器	32
0x008	LPTIM_CMP	LPTimer 比较寄存器	32
0x00C	LPTIM_CR	LPTimer 控制寄存器	32
0x010	LPTIM_IFR	LPTimer 中断标志寄存器	32
0x014	LPTIM_ICLR	LPTimer 中断标志清除寄存器	32

12.4.2 计数器计数值寄存器 (LPTIM_CNT)

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNT															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CNT	定时器计数值	RO

12.4.3 重载寄存器 (LPTIM_ARR)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ARR															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	ARR	定时器重载寄存器 写 ARR 前需要读取 CR.WT_FLAG。当 WT_FLAG 为 1 时，才能写入数据。写 ARR 寄存器后 CR.WT_FLAG 被置 0。  说明 在定时器启动时或者重载时这个值会装载到定时器计数器中。定时器计数器值可通过 CR.SYNCCOUNTERST 软复位。不建议设置 ARR 为 0xFFFF，避免 count 计数器一直停留在 0xFFFF。	RW

12.4.4 比较寄存器 (LPTIM_CMP)

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CMP															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CMP	定时器比较寄存器 写 CMP 前需要读取 CR.WT_FLAG，当 WT_FLAG 为 1 时，才能写入数据。写 CMP 寄存器后 CR.WT_FLAG 被置 0。  说明 如果先配置 ARR 导致 CR.WT_FLAG 被置 0，可以不受约束直接配置更改。	RW

12.4.5 控制寄存器 (LPTIM_CR)

Offset	Bit Position																																				
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000088																																				
Name	Reserved																SYNCCOUNT	PRS				IE	GATE_P		GATE	WT_FLAG		Reserved	TCK_SEL		TOG_EN	CT	MD	TR			
Access																	RW						RW	RW	RW	RO		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	SYNCCOUNT RST	计数器同步复位 写 1 计数器同步复位，写 0 无效 读取： ● 0b0：空闲 ● 0b1：处于复位过程中 软件置 1 后，LPTIM_CNT 会被清 0（同步时间 2~3 计数器时钟）  说明 ● 由于 LPTimer 是异步定时器，软件配置同步到计数器时钟域需要经过 2~3 计数器时钟，故建议 TR 关闭时，若 LPTIM_CNT.CNT 大于等于 0xFFFC，在开启定时器前使用此寄存器对 CNT 计数器进行同步复位，避免不预期的 overflow 现象。在 TOG 输出使能有效 (LPTIM_CR.TOG_EN=1) 且定时器关闭时，若 LPTIM_CNT.CNT 在 LPTIM_CMP.CMP-3 到 LPTIM_CMP.CMP 区间范围内，在开启定时器前使用此寄存器对 CNT 计数器进行同步复位，避免不预期的 PWM 现象。 ● 定时器停止时，开启定时器 (LPTIM_CR.TR=1) 前需要确认该位为 0。	RW
14:11	PRS	预除频 在定时器运行时不可以更改 ● 0b0000: 1 ● 0b0001: 2 ● 0b0010: 4 ● 0b0011: 8 ● 0b0100: 16 ● 0b0101: 32 ● 0b0110: 64 ● 0b0111: 128 ● 0b1000: 256 ● 0b1001: 512 ● 0b1010: 1024 ● 0b1011: 2048 ● 0b1100: 4096 ● 0b1101: 8192 ● 0b1110: 16384 ● 0b1111: 32768	RW

位/位域	标记	功能描述	读写
10	IE	中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
9	GATE_P	GATE 极性控制 ● 0b0: 高电平有效 ● 0b1: 低电平有效	RW
8	GATE	定时器门控 ● 0b0: 无门控, TR=1 时定时器工作 ● 0b1: 只有门控为 1 并且 TR=1 时定时器工作	RW
7	WT_FLAG	WT 写同步标志 ● 0b0: 正在同步, 此时写 ARR/CMP 无效 ● 0b1: 同步完成, 此时可更改 ARR/CMP	RO
6	<i>Reserved</i>	-	-
5:4	TCK_SEL	LPTimer 时钟选择 ● 0b00: PCLK ● 0b01: ETR ● 0b10: XTL ● 0b11: RCL	RW
3	TOG_EN	PWM 输出使能 ● 0b0: TOG、TOGN 同时输出 0 ● 0b1: TOG、TOGN 输出相位相反的信号	RW
2	CT	计数器/定时器功能选择 ● 0b0: 定时器功能, 定时器使用 TCK_SEL 选择的时钟进行计数 ● 0b1: 计数器功能, 计数器使用外部输入的下降沿进行计数  说明 采样时钟使用 TCK_SEL 选择的时钟, 外部输入时钟要低于 1/2 采样时钟。	RW
1	MD	定时器工作模式 ● 0b0: 模式 1 无重载模式 16 位计数器/定时器 ● 0b1: 模式 2 自动重载 16 位计数器/定时器	RW
0	TR	定时器运行控制位 ● 0b0: 定时器停止 ● 0b1: 定时器运行	RW

12.4.6 中断标志寄存器 (LPTIM_IFR)

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															OV
Access																																RO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	OV	计数器溢出标志 ● 0b0: 未溢出 ● 0b1: 已溢出	RO

12.4.7 中断标志清除寄存器 (LPTIM_ICLR)

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															OV
Access																																WO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	OV	计数器溢出标志清除 写 0 清除标志，写 1 无效；读出恒为 0	WO

13 通用定时器 (TIM0/1/2/3)

13.1 简介

TIM0/1/2 是分别由 1 个计数单元和 2 个比较单元组成的定时器。每个定时器支持 2 个独立 PWM 输出或 1 对互补 PWM 输出和 1 个独立 PWM 输出，支持 2 个捕获输入。TIM0/1/2 可以组成 3 对互补 PWM 输出。

TIM3 是由 1 个计数单元和 8 个比较单元组成的定时器，支持 6 路独立 PWM 输出或 3 对互补 PWM 输出，支持 6 路捕获输入。另有 2 个专用比较单元，可以产生比较中断，并可以用于触发 ADC 和 DMA。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

13.2 主要特性

13.2.1 TIM0/1/2 主要特性

- 2 路独立 PWM 输出 CHA、CHB
- 1 路互补 PWM 输出 (CHA、CHB) + 1 路独立 PWM 输出 (gate)
- 1 路互补 PWM 输出 (CHA、CHB) + 1 路捕获功能(gate)
- 最多 2 路捕获输入
- 脉冲宽度测量
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发
- 死区控制
- 刹车控制
- 逐波限流
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出

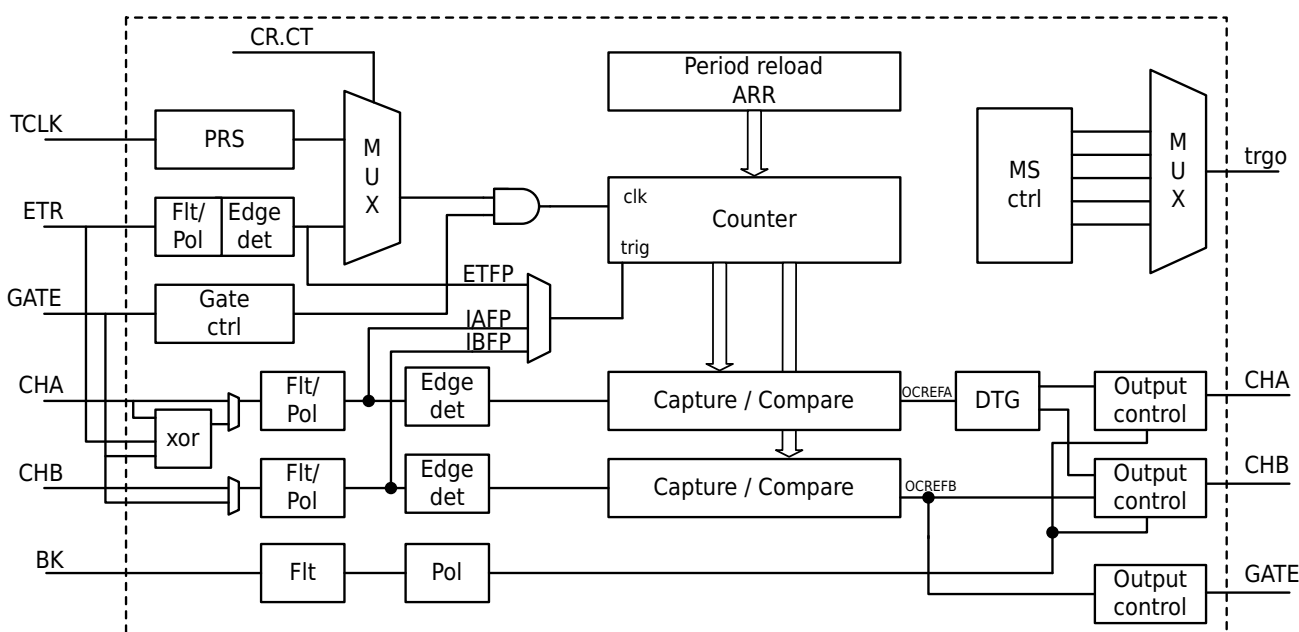


图 13-1 TIM0/1/2 框图

13.2.2 TIM3 主要特性

- 6 路独立 PWM 输出 CH0A、CH0B、CH1A、CH1B、CH2A、CH2B
- 3 路互补 PWM 输出 (CHxA、CHxB) + 1 路独立 PWM 输出 (gate)
- 3 路互补 PWM 输出 (CHxA、CHxB) + 1 路捕获功能 (gate)
- 2 路专用比较单元，触发 ADC 和 DMA
- 最多 6 路捕获输入
- 脉冲宽度测量
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能
- DMA 触发
- 死区控制
- 刹车控制
- 逐波限流
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出

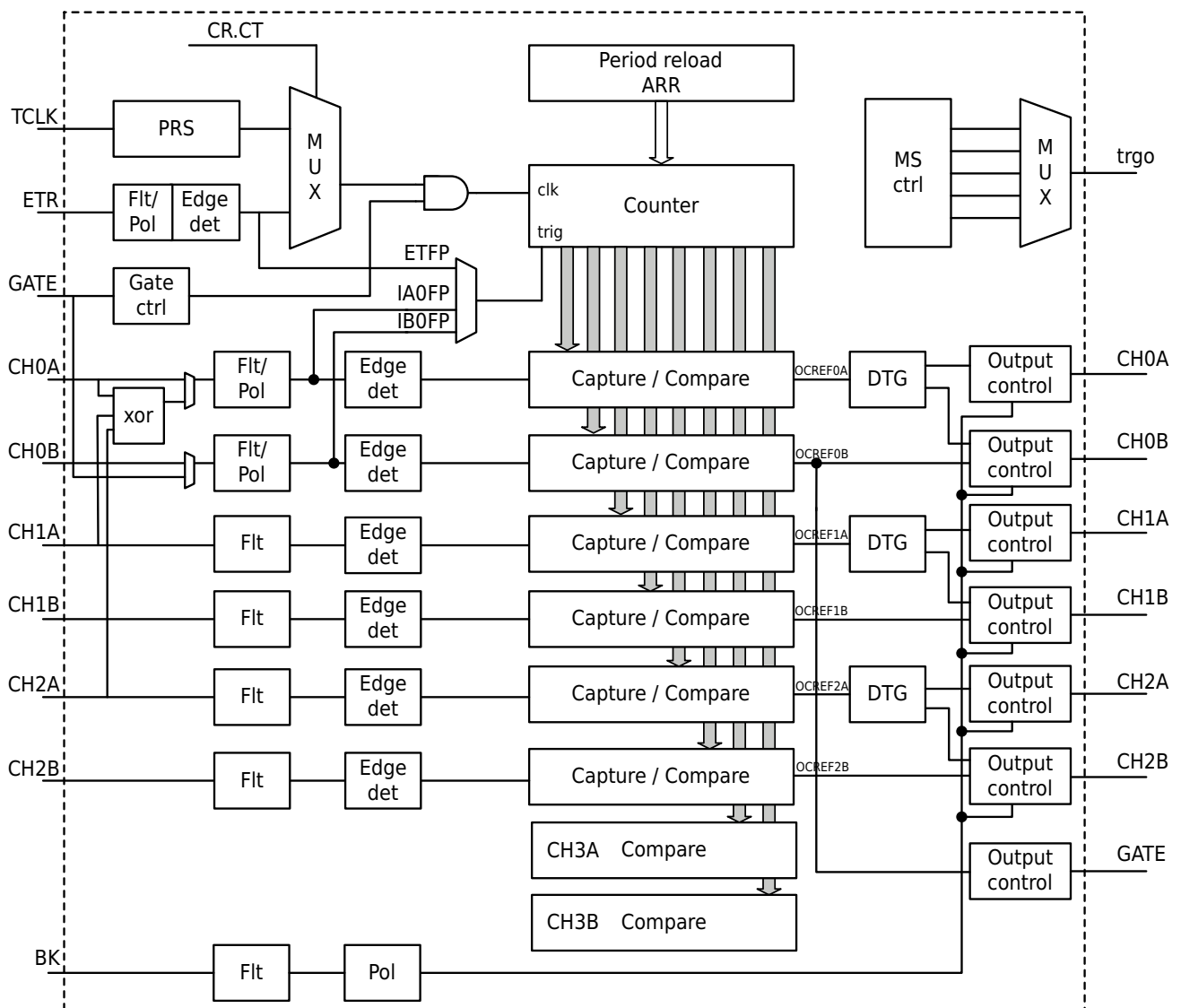


图 13-2 TIM3 框图

13.3 功能说明

13.3.1 定时器时钟

定时器使用 PCLK 作为定时器时钟，下文定时器时钟使用 TCLK 标示。

13.3.2 定时计数器

可编程通用定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

13.3.3 定时器预分频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	0b000	0b001	0b010	0b011	0b100	0b101	0b110	0b111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

13.3.4 模式 0 计数定时器功能

该模式计数器向上计数。计数器支持两种计数模式，重载模式与自由计数模式，可以选择外部时钟 ETR 计数或系统时钟计数。门控 Gate 可以控制计数屏蔽。计数到最大值溢出产生中断。翻转输出信号 TOG/TOGN 由端口 CHA/CHB 输出。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出，然后从 ARR 再开始计数，计数周期为 0xFFFF-ARR+1；自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

计数器可用通过外部门控功能控制计数器是否计数。控制关系如下：

M0CR.GATE	M0CR.GATEP	端口 GATE 输入	M0CR.CTEN	计数器
x	x	x	0	不计数
0	x	x	1	计数
1	0	高电平	1	计数
1	0	低电平	1	不计数
1	1	高电平	1	不计数
1	1	低电平	1	计数

功能框图

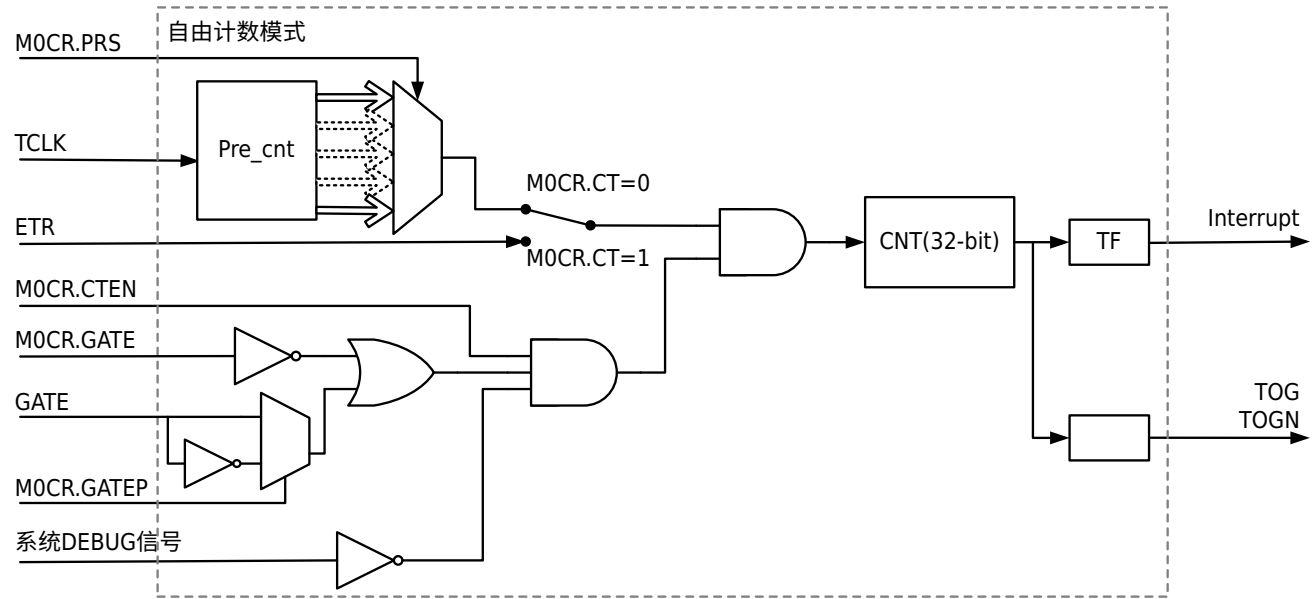


图 13-3 模式 0 计数定时器自由计数框图

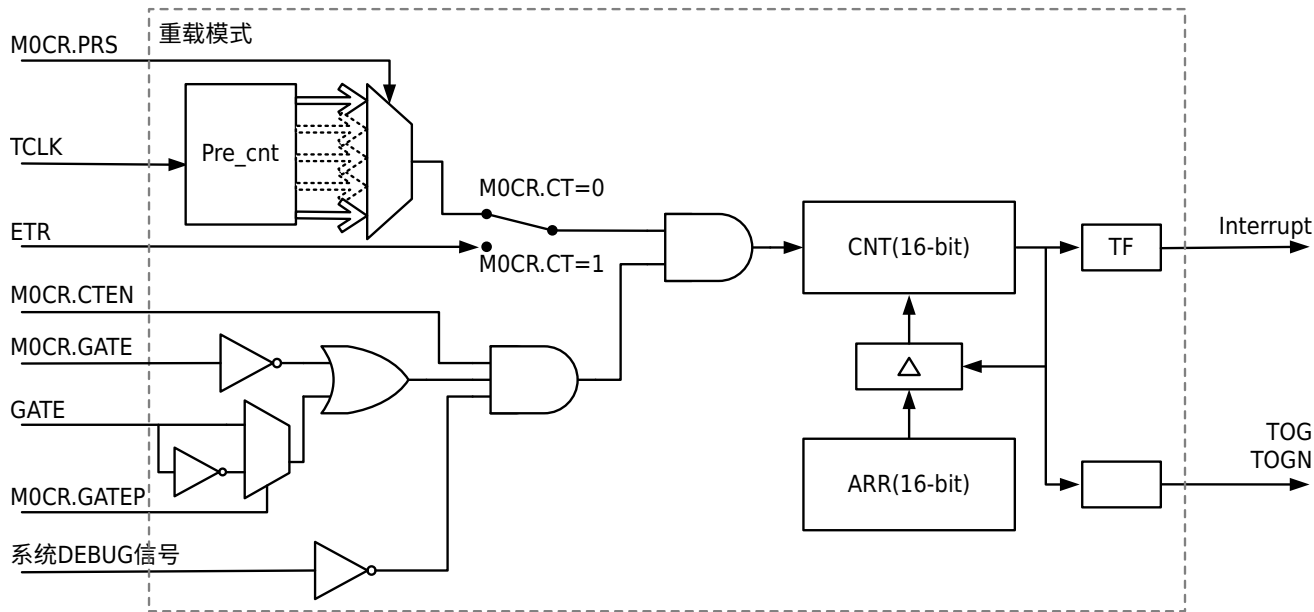


图 13-4 模式 0 计数定时器重载计数框图

 说明

框图中“系统 DEBUG 信号”来源为 DEBUG_ACTIVE。

计数波形

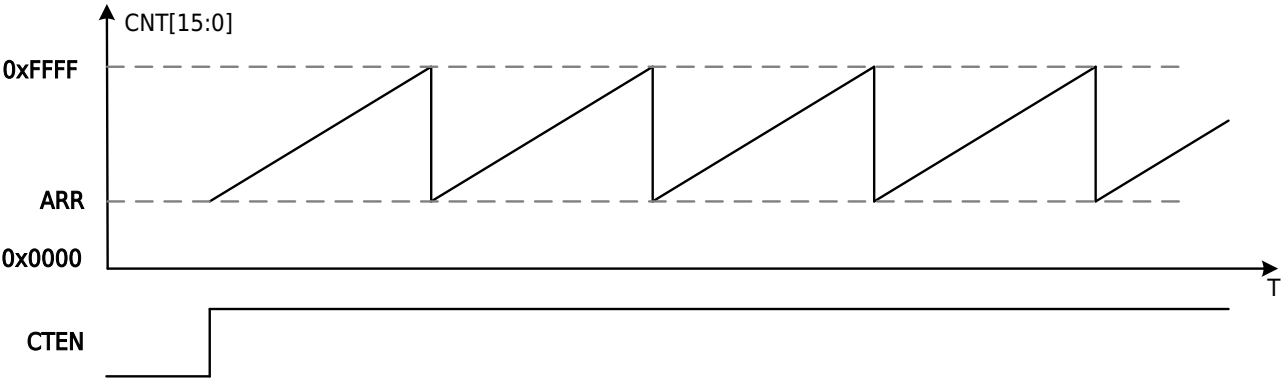


图 13-5 模式 0 计数定时器 16 位重载计数波形

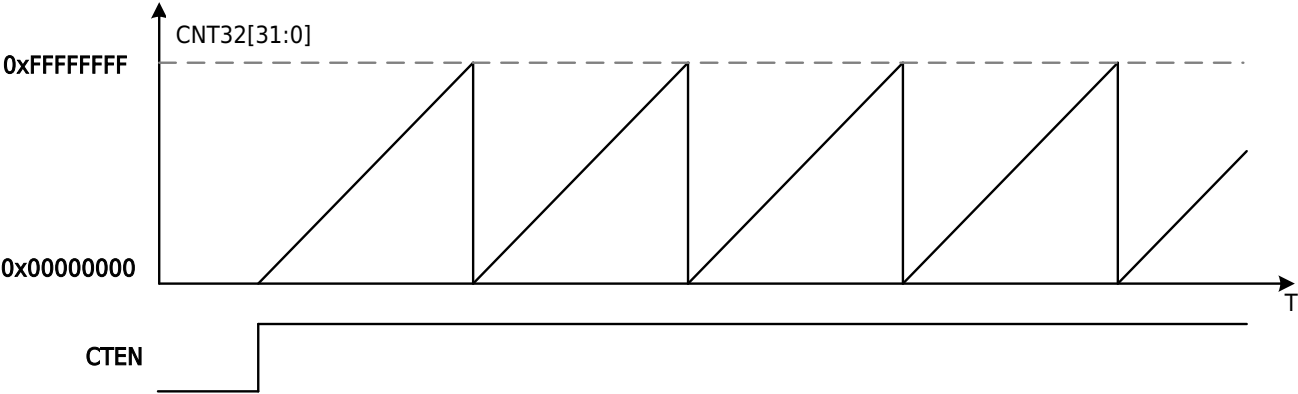


图 13-6 模式 0 计数定时器 32 位自由计数波形

计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 TCLK 采样，因此外部输入信号保持稳定的时间需要大于一个 TCLK 的时钟周期。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

**注意**
当 ETR 输入作为计数器的计数时钟时，ETR 的时钟频率不能大于 1/2 的 TCLK 时钟。

定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

时序图

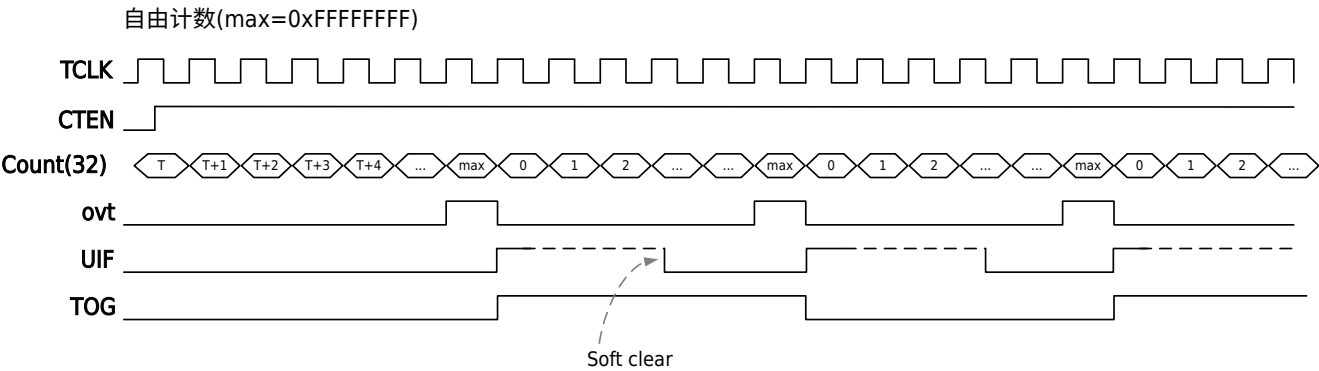


图 13-7 模式 0 计数定时器自由计数时序图

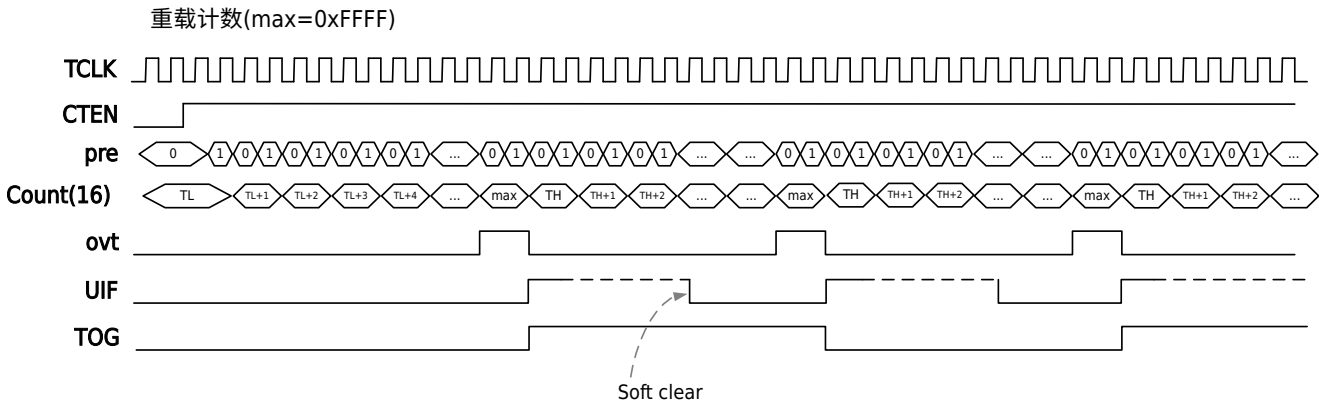


图 13-8 模式 0 计数定时器重载计数时序图 (预分频设置为 2)

Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 M0CR.TOGEN 为 0 可以同时设置端口 CHA、CHB 输出为 0。在计数时钟为 4MHz 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNT 初始值	ARR 重载值
1000Hz	0.5ms	2000	63536	0xF830	0xF830
2000Hz	0.25ms	1000	64536	0xFC18	0xFC18
4000Hz	0.125ms	500	65036	0xFE0C	0xFE0C

13.3.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。
第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CCR0A，并且产生捕获中断 CA0F，如果计数器发生溢出，会产生溢出标志。设置溢出中断使能会产生溢出中断。

M1CR.EDG1ST	0	0	1	1
M1CR.EDG2ND	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

PWC 功能框图

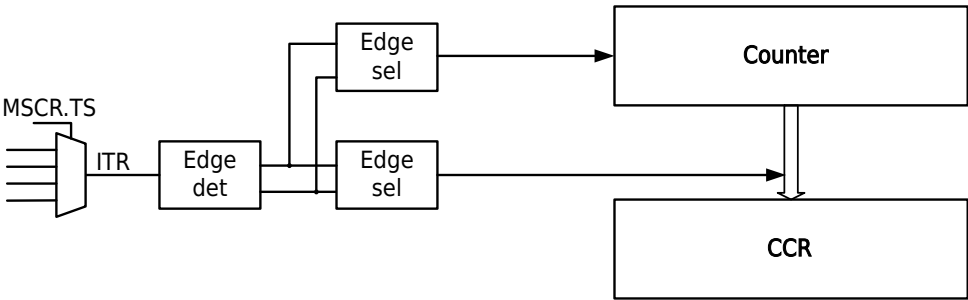


图 13-9 模式 1 脉宽测量 PWC 功能框图

通过寄存 MSCR.TS 选择测量信号源。

触发选择	取值	信号源
MSCR.TS	0b000	ETFP: ETR 外部输入滤波后的相位选择信号，可选择外部滤波与输入反向
	0b001	ITR0: Timer 内部互联信号 0，其他 timer 的 TRGO 输出
	0b010	ITR1: Timer 内部互联信号 1，其他 timer 的 TRGO 输出
	0b011	ITR2: Timer 内部互联信号 2，其他 timer 的 TRGO 输出
	0b100	ITR3: Timer 内部互联信号 3，其他 timer 的 TRGO 输出
	0b101	IA0ED: 无效
	0b110	IAFP: CH0A 外部输入滤波后的相位选择信号，可选择外部滤波与输入反向（极性选择在脉宽测量模式下无效）
	0b111	IBFP: CH0B 外部输入滤波后的相位选择信号，可选择外部滤波与输入反向（极性选择在脉宽测量模式下无效）

关于 TRGO 输出请参见相应寄存器描述。

定时器	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

PWC 波形测量时序图

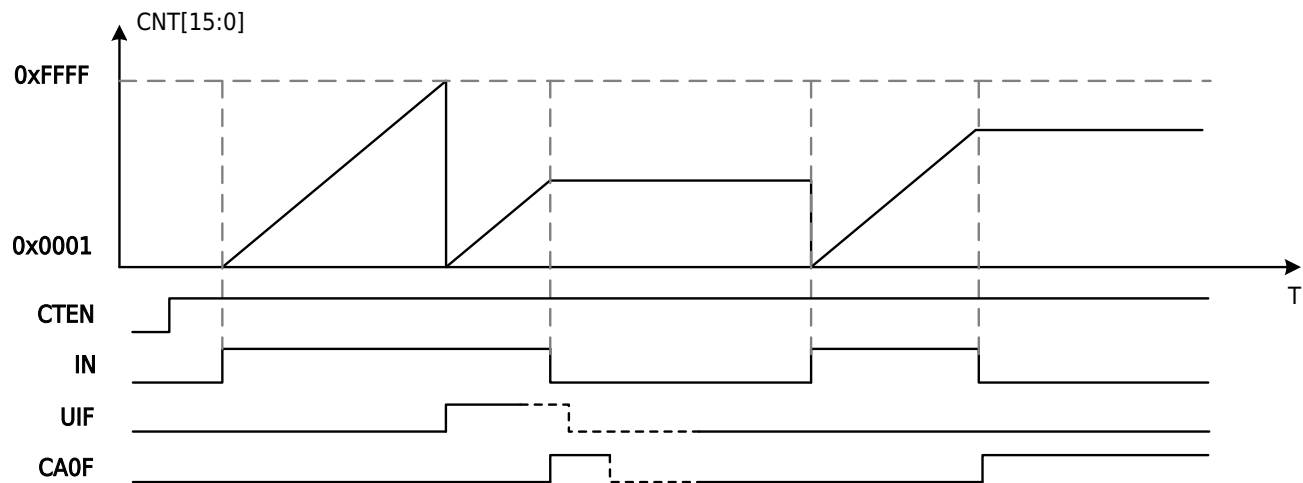


图 13-10 模式 1 脉宽测量 PWC 高电平脉冲宽度测量

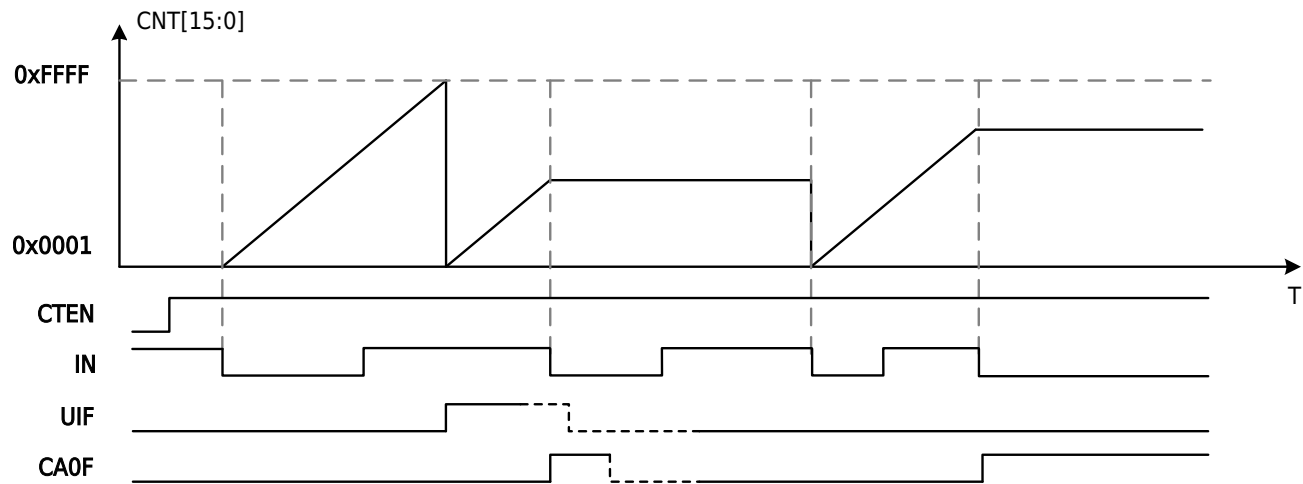


图 13-11 模式 1 脉宽测量 PWC 下降沿到下降沿周期测量

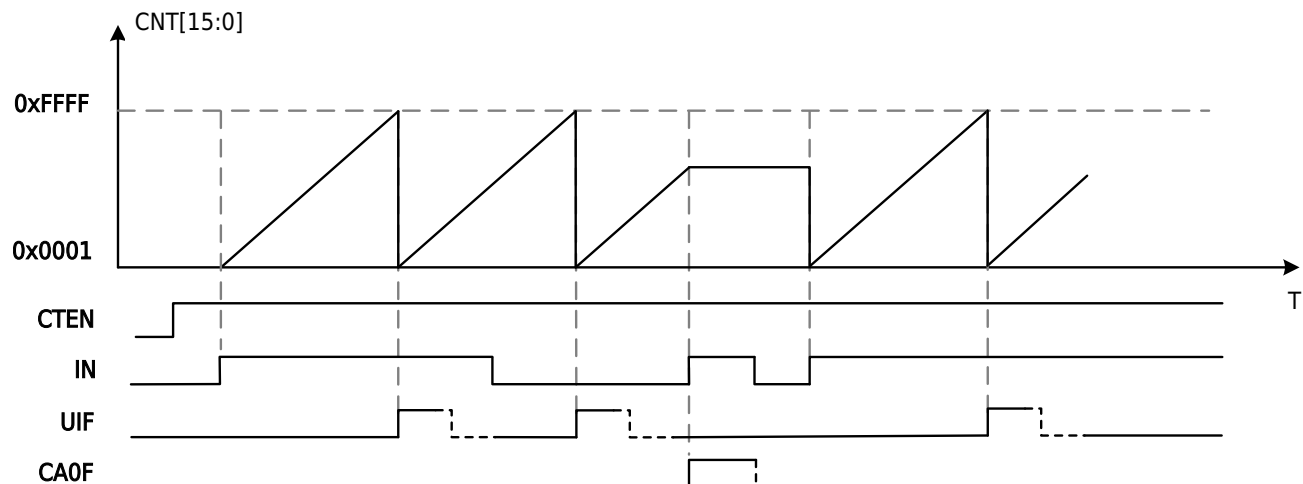


图 13-12 模式 1 脉宽测量 PWC 上升沿到上升沿周期测量

PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量，测量完成后 CTEN 将被清除。

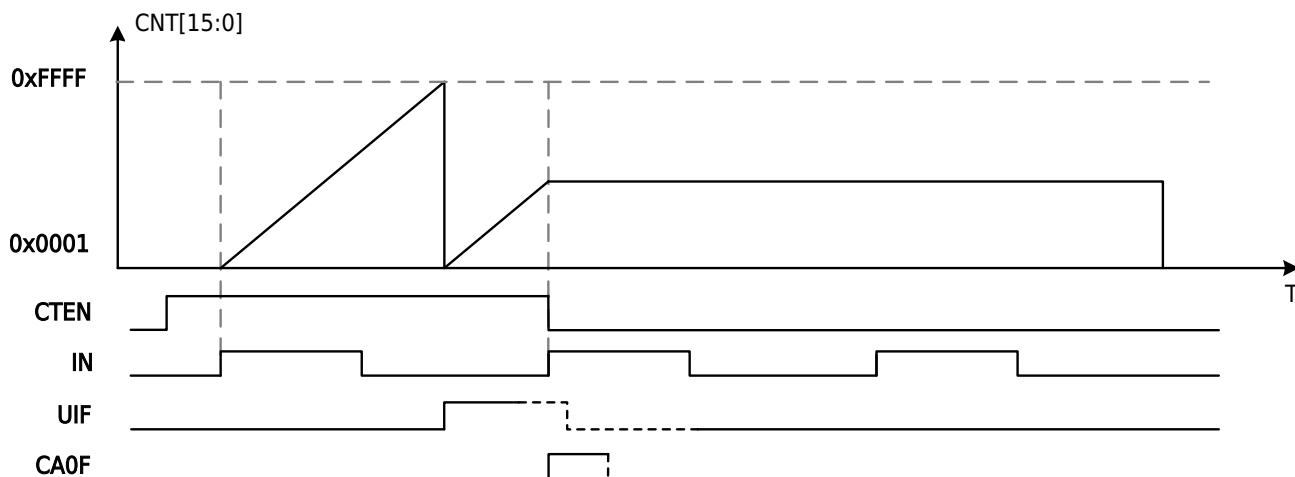


图 13-13 模式 1 脉宽测量 PWC 上升沿到上升沿周期测量单次模式

13.3.6 模式 2/3 比较捕获模式

13.3.6.1 计数器

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

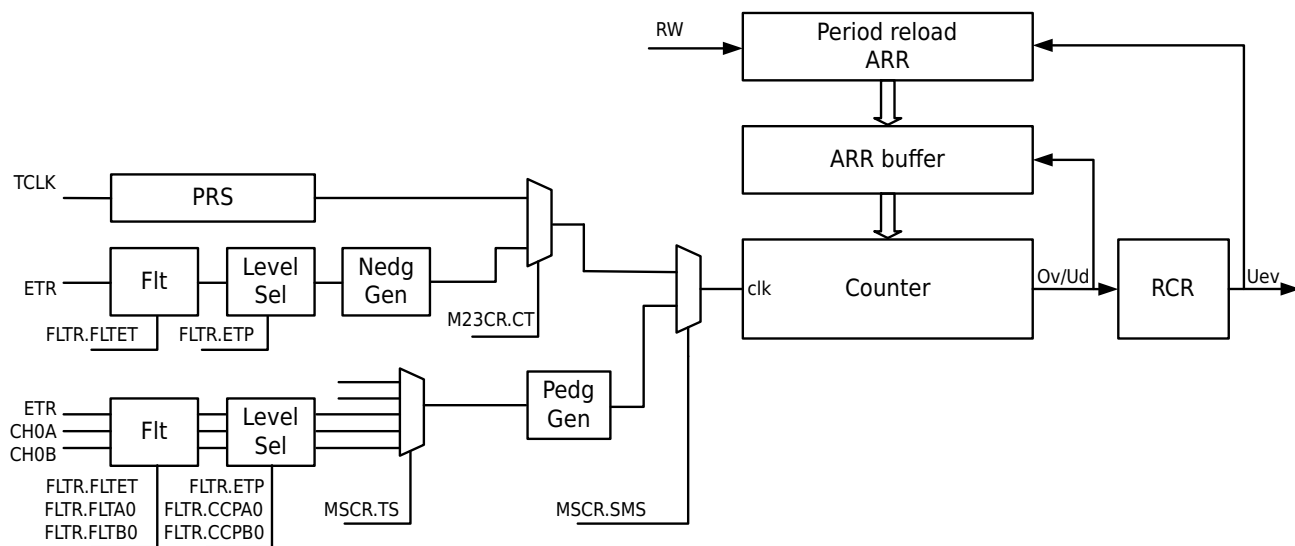


图 13-14 模式 2/3 比较捕获模式计数器框图

计数器基本单元包括：

- 计数器寄存器 CNT
- 预分频寄存器 M23CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLTR、M23CR、MSCR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并行缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才有自动装载寄存器更新到缓存寄存器。

时钟选择及门控功能，触发功能，复位功能请参见[模式 2/3 从模式](#)章节相关描述。

13.3.6.2 计数器波形

模式 2 锯齿波计数波形

模式 2 为锯齿波计数波形，通过设置 M23CR.DIR 可以更改计数方向。

设置 M23CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值 (TIMx_ARR)，然后重新从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIMx_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 M23CR.UG 位置 1 (通过软件或使用从模式控制器) 时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志 (TIMx_IFR.UIF 位) 置 1 (取决于 M23CR.URS 位)：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下几张图为 ARR=0x2C 时不同计数方向的计数器波形，计数溢出周期时钟个数为 ARR+1。

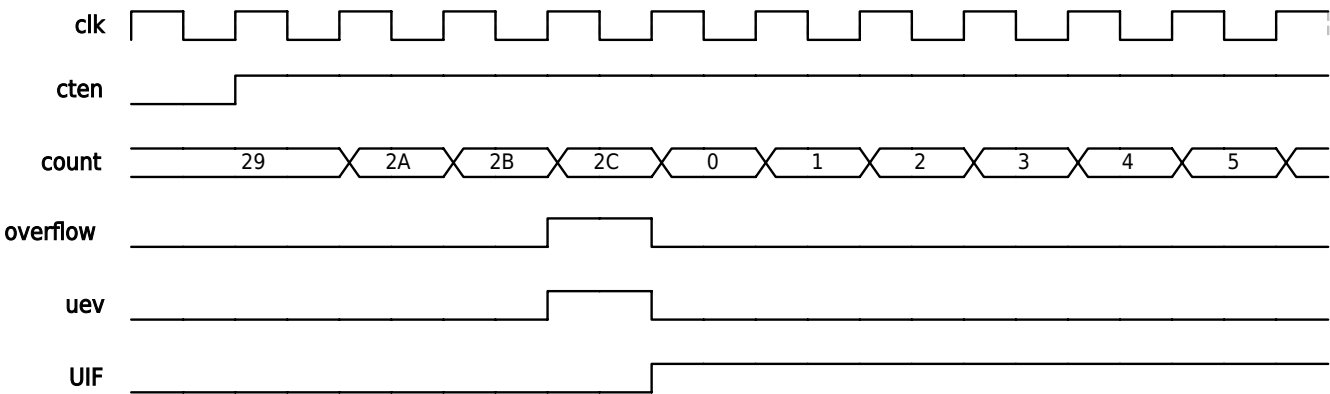


图 13-15 模式 2 比较捕获模式无预分频的向上计数

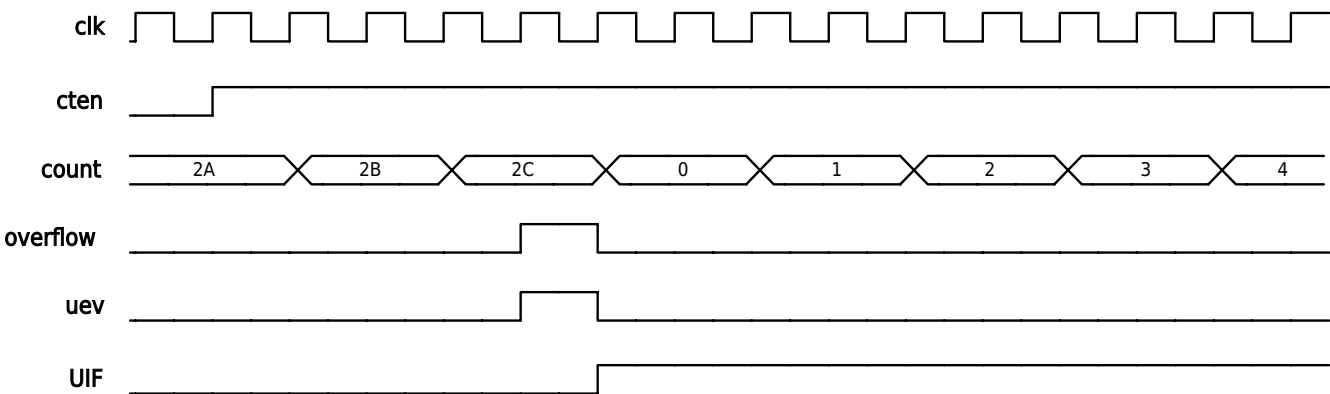


图 13-16 模式 2 比较捕获模式带预分频的向上计数

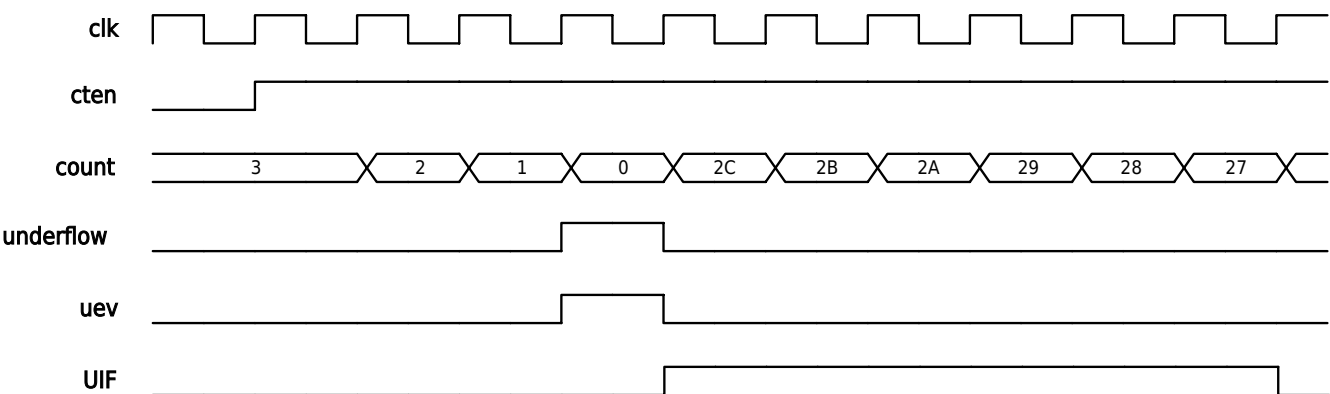


图 13-17 模式 2 比较捕获模式无预分频的向下计数

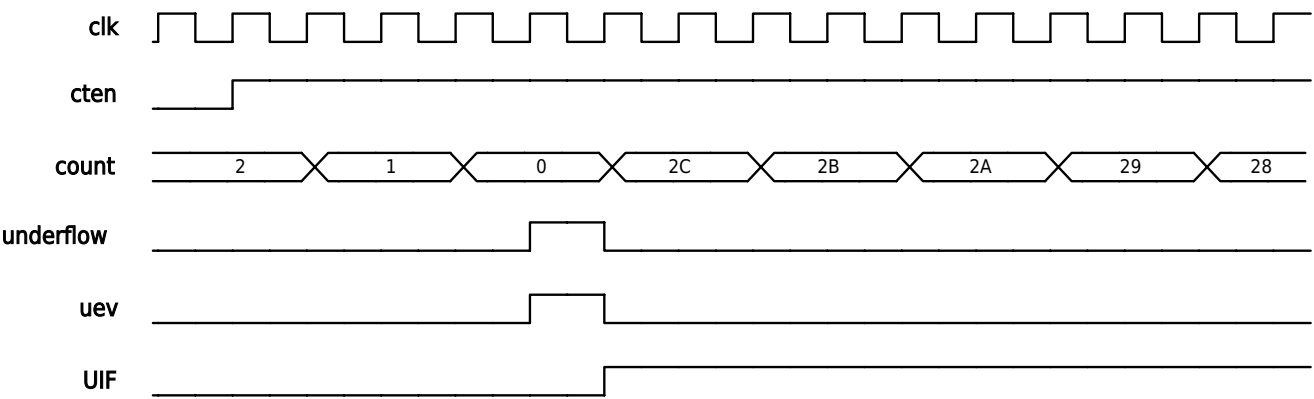


图 13-18 模式 2 比较捕获模式带预分频的向下计数

模式 2 的计数波形如下图所示：

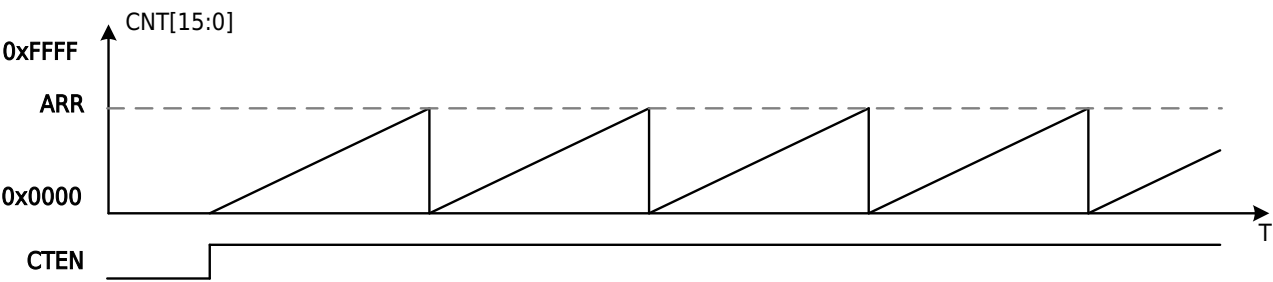


图 13-19 模式 2 边沿对齐计时器波形(M23CR.DIR=0)

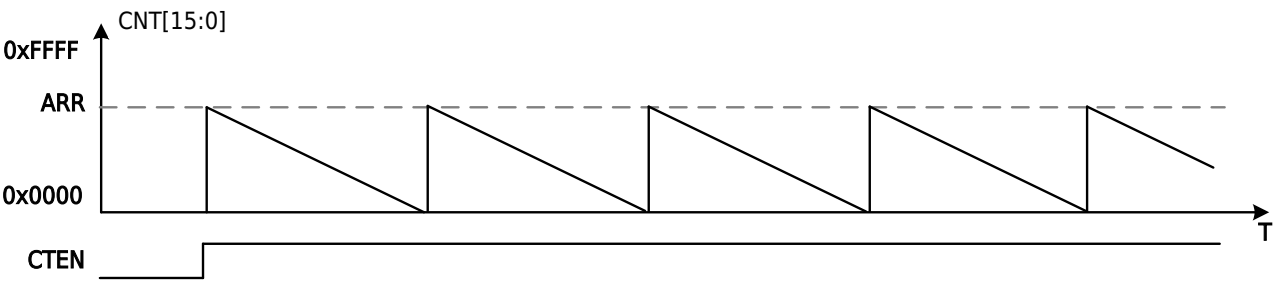


图 13-20 模式 2 边沿对齐计时器波形(M23CR.DIR=1)

模式 3 三角波计数波形

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 M23CR.DIR 方向位是只读的，写值无效。从其他模式切换到中心对齐模式 DIR 自动清 0。软件事件更新和从模式外部触发复位模式 DIR 自动清零。

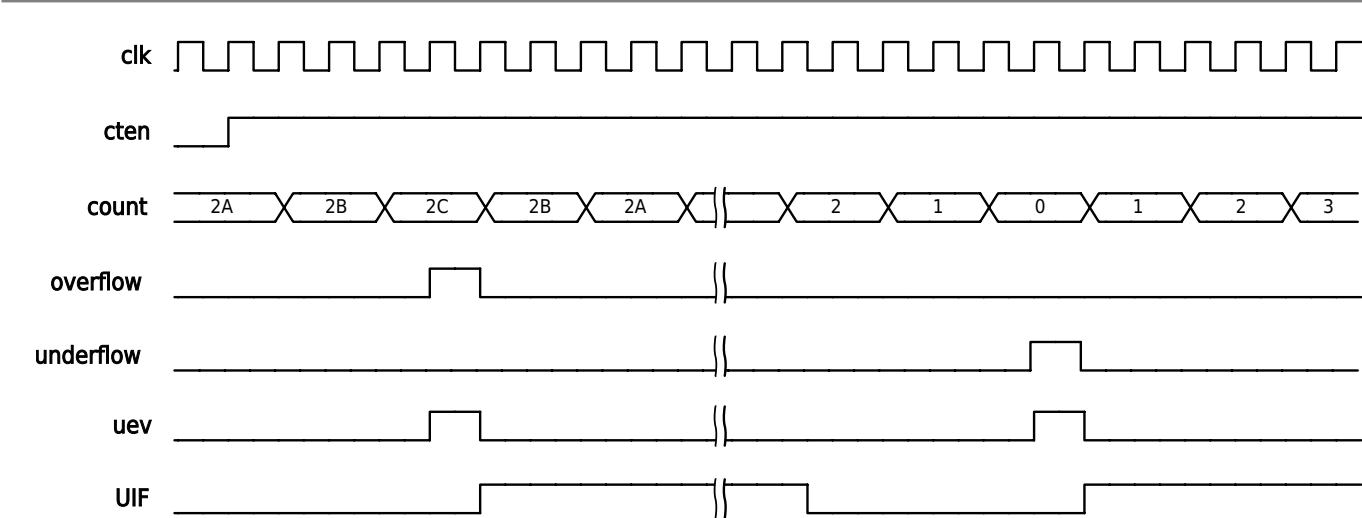


图 13-21 模式 3 比较捕获模式带预分频的上下计数

模式 3 的计数波形如下图所示：

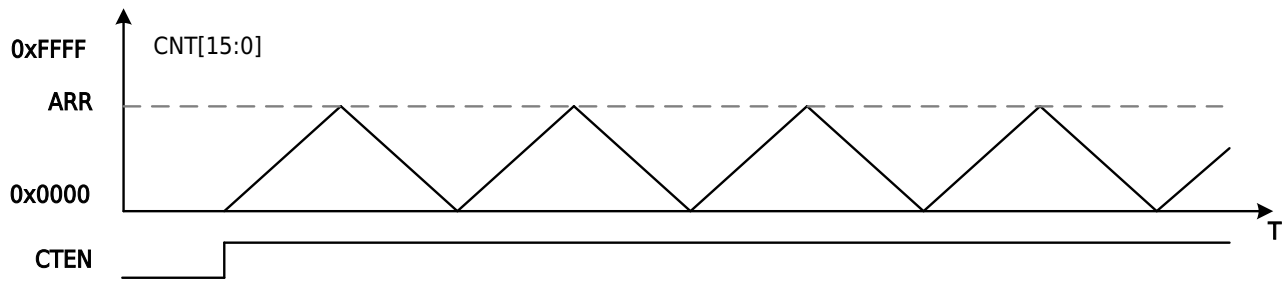


图 13-22 模式 3 中心对齐计数器波形

13.3.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存器的值更新到比较缓存寄存器中。

重复计数器在下面条件成立时递减（当 RCR.UND=0 且 RCR.OV=0）：

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

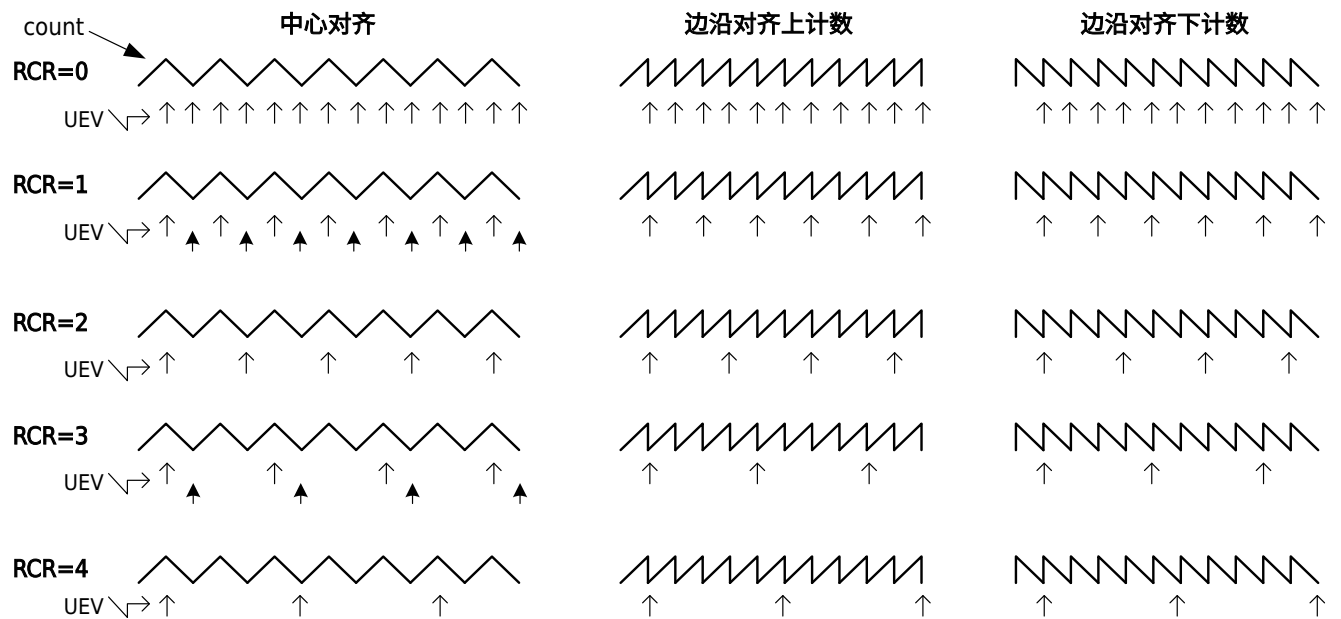


图 13-23 重复计数器产生更新时序 (RCR.OV=0, RCR.UND=0)

RCR.OV 为重复计数控制上溢屏蔽功能位，如果 RCR.OV 为 1，则在执行重复计数时候，会忽略上溢事件。

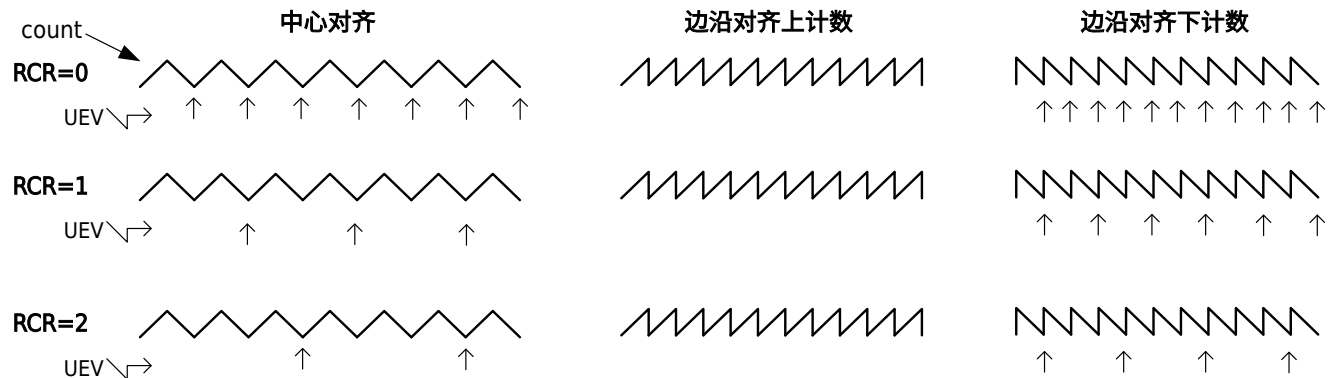


图 13-24 上溢屏蔽时候重复计数器产生更新时序 (RCR.OV=1, RCR.UND=0)

RCR.UND 为重复计数控制下溢屏蔽功能位，如果 RCR.UND 为 1，则在执行重复计数时候，会忽略下溢事件。

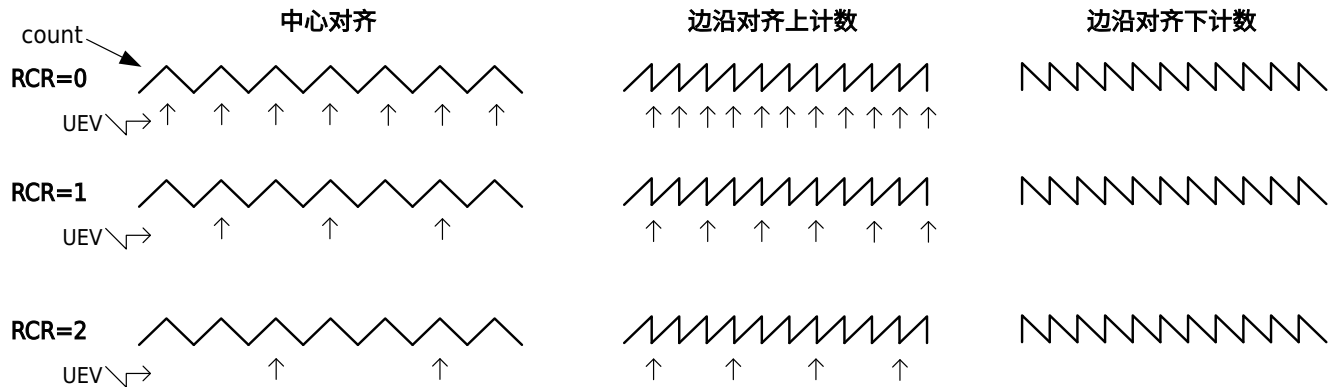


图 13-25 下溢屏蔽时候重复计数器产生更新时序 (RCR.OV=0, RCR.UND=1)

除了上下溢出通过重复计数器可以产生事件更新 UEV 外，还可以通过写寄存器 M23CR.UG 产生软件及从模式复位事件更新 UEV；这时需要配置 M23CR.URS。

13.3.6.4 数据缓存

自动重载数据 ARR 与比较寄存器 CCRxy 都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCRxy 才会生效。

自动重载值在不同计数模式下的更新时序图如下：

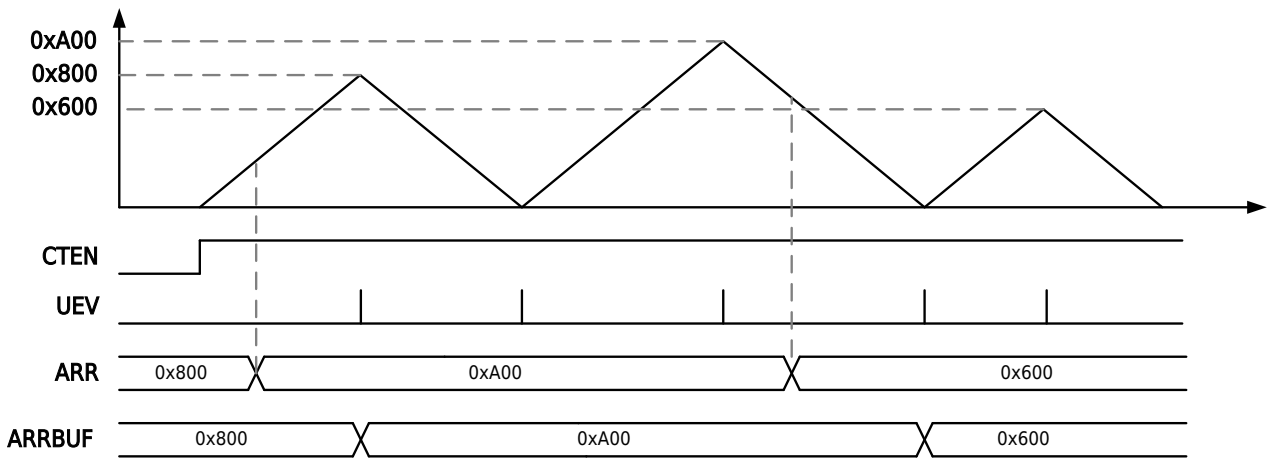


图 13-26 模式 3 三角波模式缓存使能

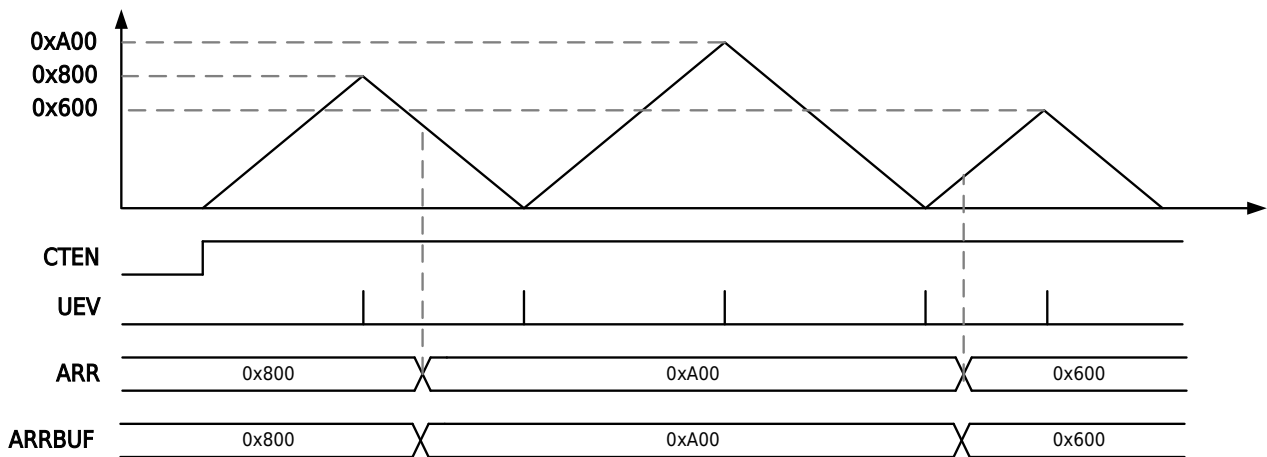


图 13-27 模式 3 三角波模式缓存无效

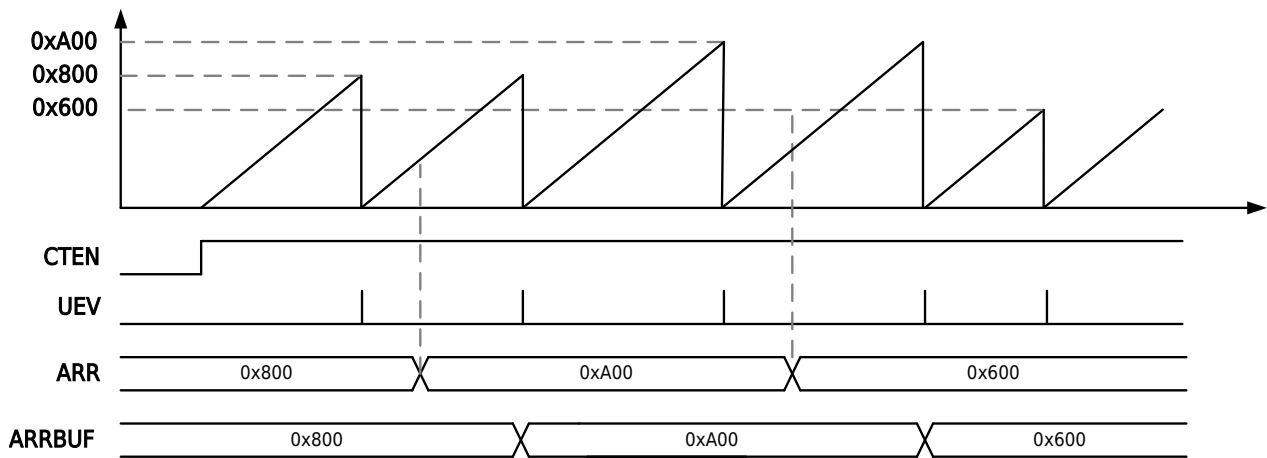


图 13-28 模式 2 锯齿波模式上计数缓存使能

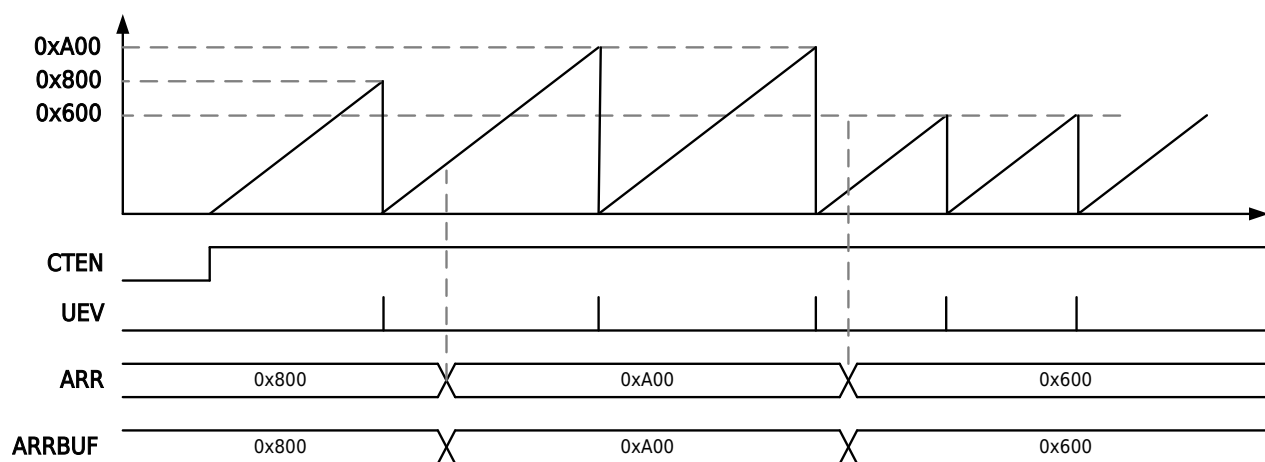


图 13-29 模式 2 锯齿波模式上计数缓存无效

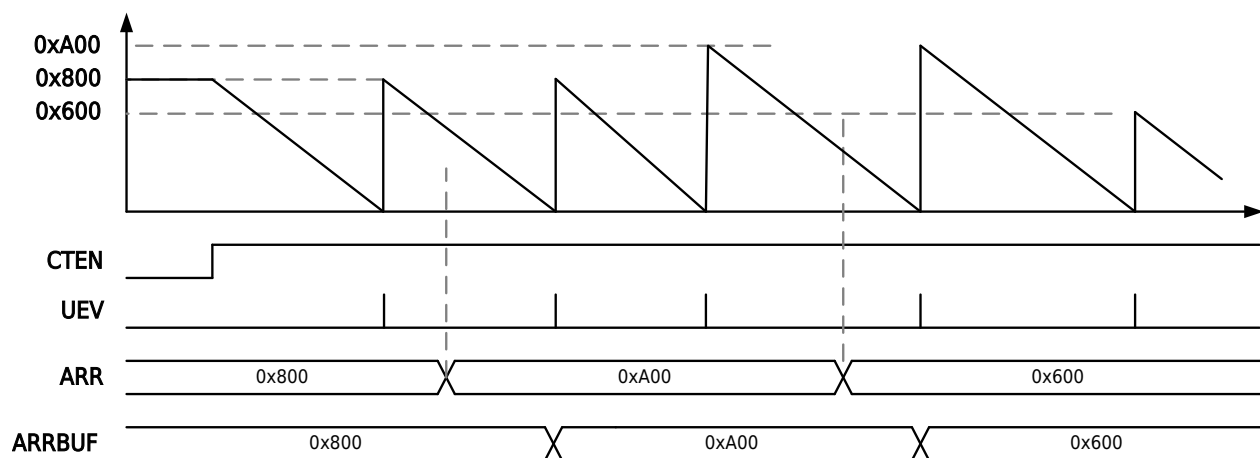


图 13-30 模式 2 锯齿波模式下计数缓存使能

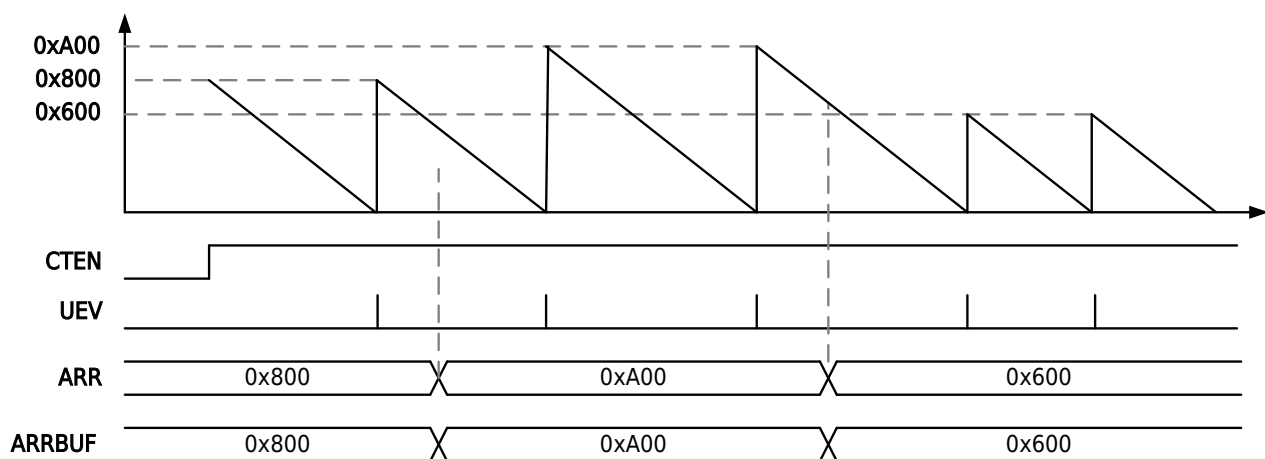


图 13-31 模式 2 锯齿波模式下计数缓存无效



说明

在三角波模式与锯齿波上计数模式时，如果缓存不使能，更改的 ARR 时，当前计数器的值要小于要更改的 ARR 周期值，否则当前周期会计数到 0xFFFF。

比较缓存与周期缓存更新状态一致，这里不一一列出时序图。

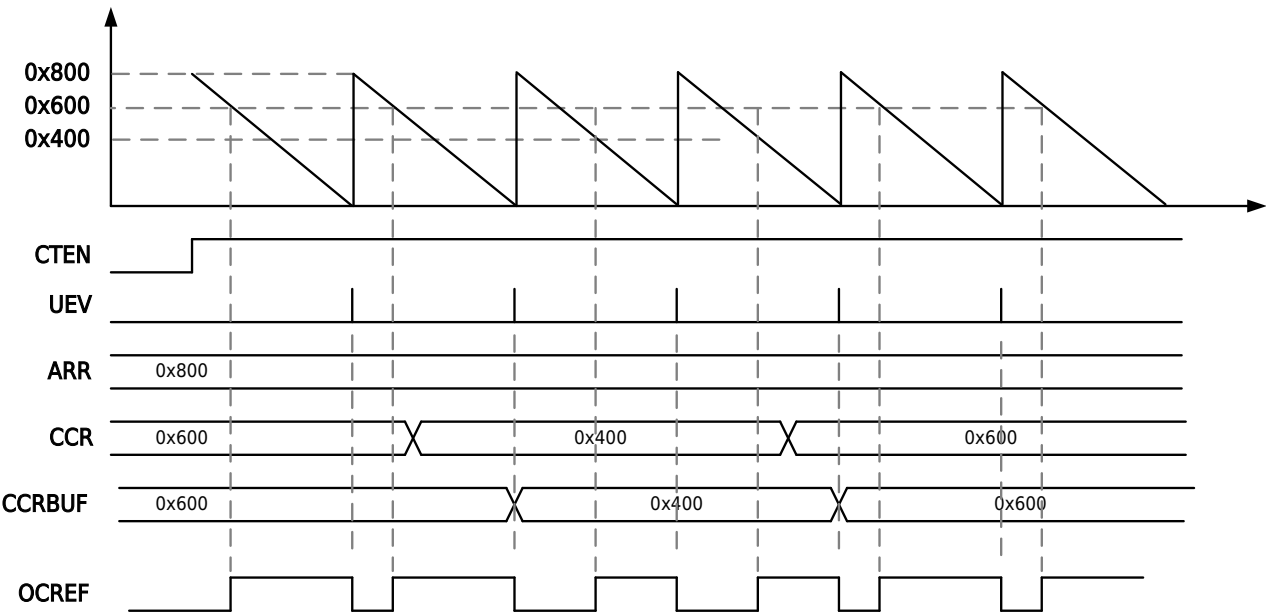


图 13-32 模式 2 锯齿波模式下计数比较缓存使能

13.3.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA、CCRB 一起控制 OCREFA 的比较输出。OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

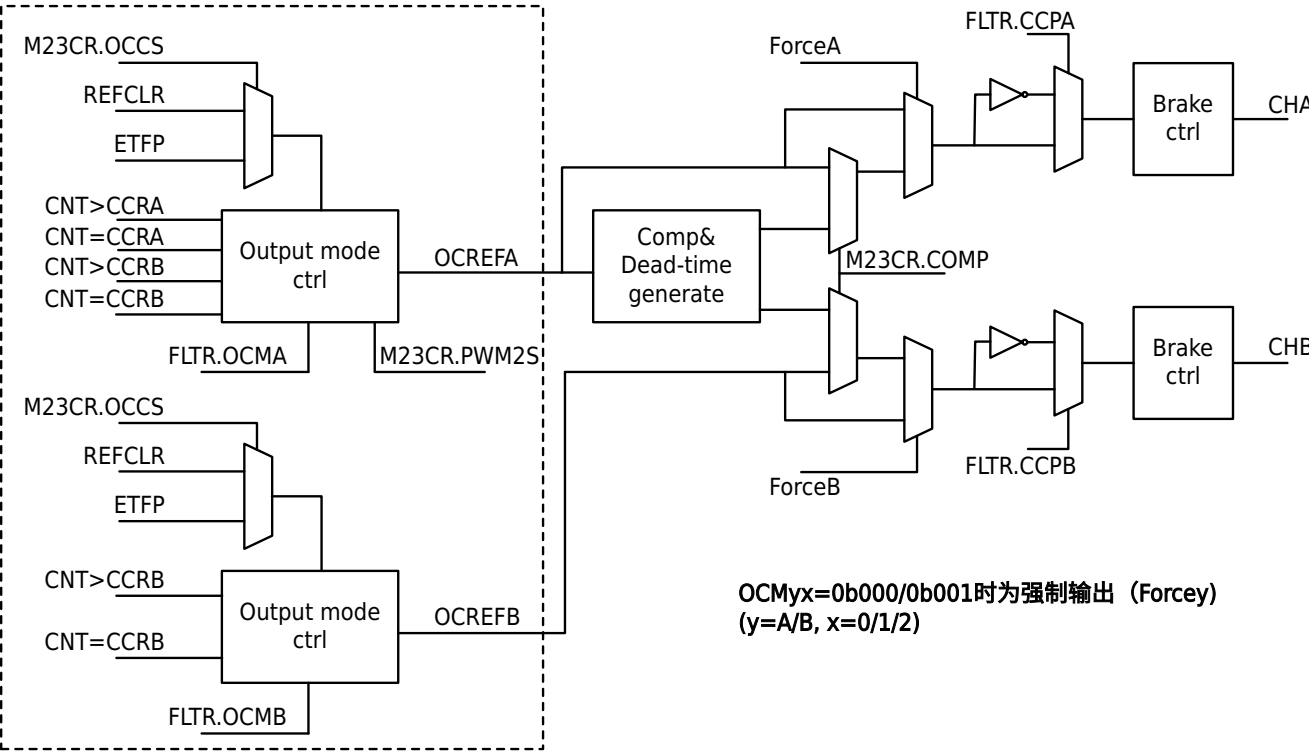


图 13-33 OCREF 输出框图

OCREF 输出使用 TIMx_FLTR.OCMyx 选择：

- 0b000：强制为 0
- 0b001：强制为 1
- 0b010：比较匹配时强制为 0
- 0b011：比较匹配时强制为 1

- 0b100: 比较匹配时翻转
- 0b101: 比较匹配时输出一个计数周期的高电平
- 0b110: PWM 模式 1
 - ▶ 单点比较
上计数时 $CNT < CCR_{xy}$ 输出为高电平, 下计数时 $CNT > CCR_{xy}$ 输出为低电平
 - ▶ 双点比较
 - ◆ 锯齿波上计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为低电平
 - ◆ 锯齿波下计数 $CCR_{xA} < CNT \leq CCR_{xB}$ 输出为高电平
 - ◆ 三角波上计数 $CNT < CCR_{xA}$ 输出为高电平, 下计数 $CNT > CCR_{xB}$ 输出为低电平
- 0b111: PWM 模式 2
 - ▶ 单点比较
上计数时 $CNT < CCR_{xy}$ 输出为低电平, 下计数时 $CNT > CCR_{xy}$ 输出为高电平
 - ▶ 双点比较
 - ◆ 锯齿波上计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为高电平
 - ◆ 锯齿波下计数 $CCR_{xA} \leq CNT < CCR_{xB}$ 输出为低电平
 - ◆ 三角波上计数 $CNT < CCR_{xA}$ 输出为低电平, 下计数 $CNT > CCR_{xB}$ 输出为高电平



说明

强制输出有高优先级, 当强制输出有效时, 互补输出控制无效。

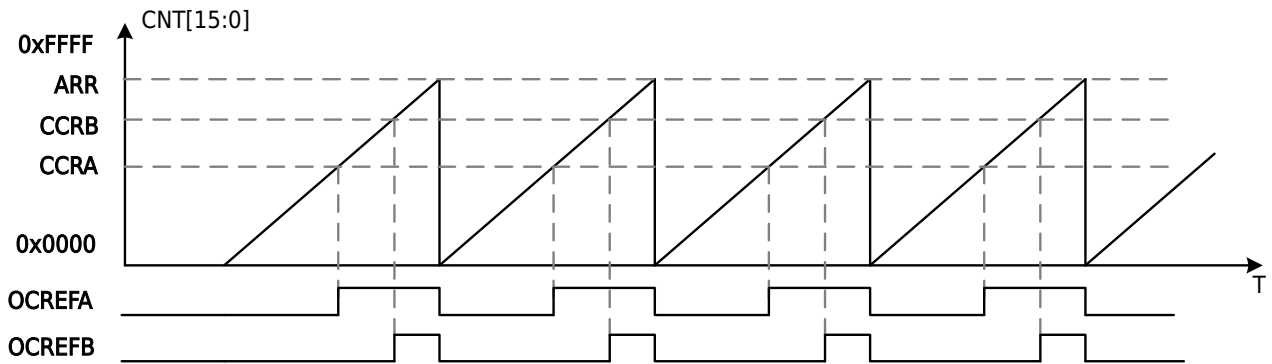


图 13-34 锯齿波计数单点比较 OCREF 输出波形 (OCMyx=0b111)

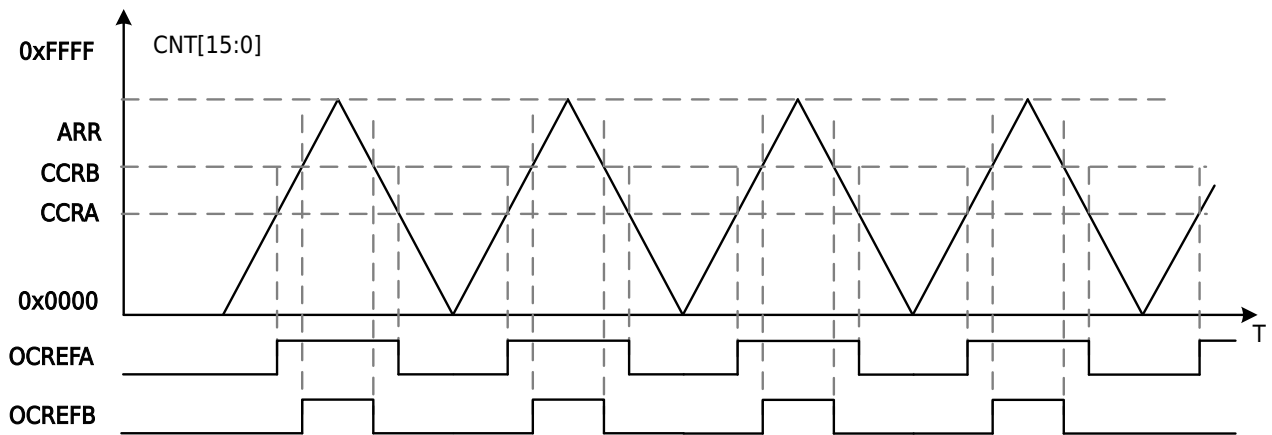


图 13-35 三角波计数单点比较 OCREF 输出波形 (OCMyx=0b111)

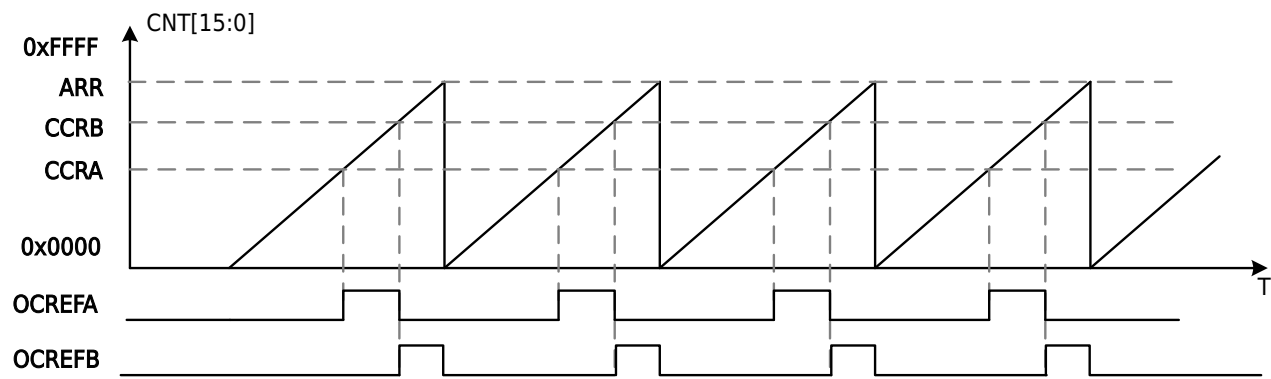


图 13-36 锯齿波计数双点比较 OCREF 输出 (OCMyx=0b111)

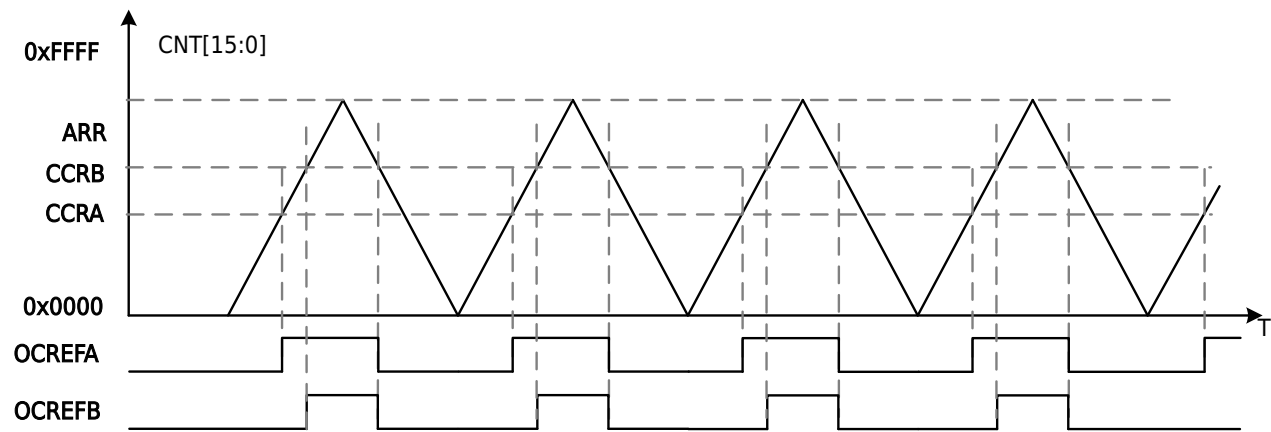


图 13-37 三角波计数双点比较 OCREF 输出 (OCMyx=0b111)

13.3.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFB 控制 CHB 的输出。通过 FLTR.CCPA、FLTR.CCPB 可以控制 CHA、CHB 输出的反向。

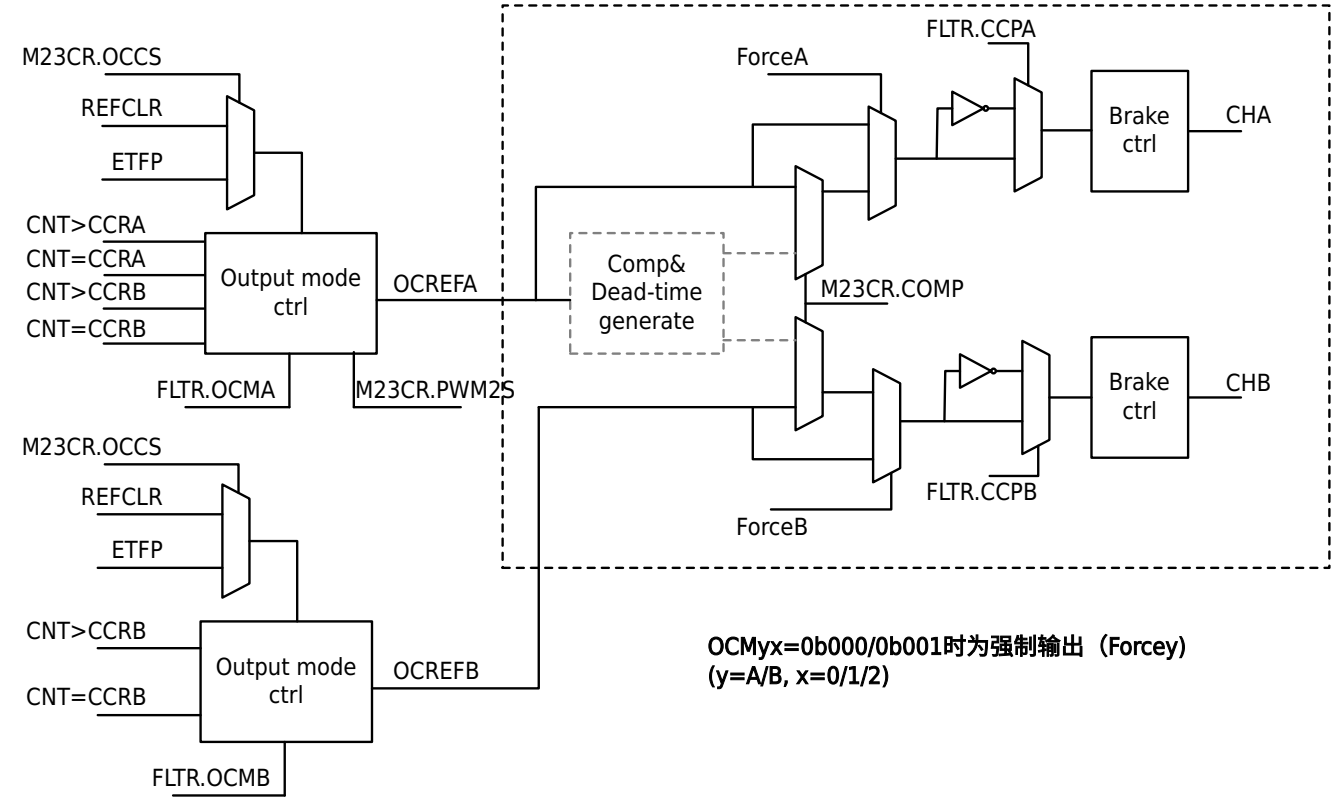


图 13-38 独立 PWM 输出框图

PWM 输出与 OCREF 关系如下图所示：

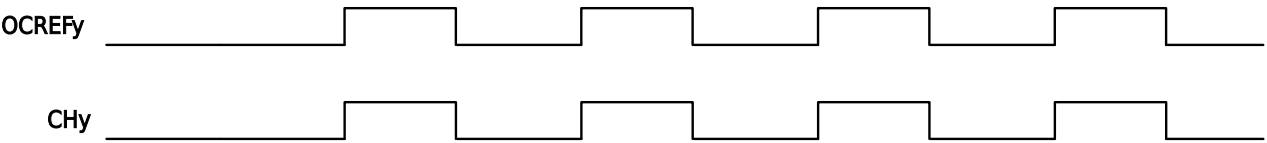


图 13-39 CCPy=0 时 PWM 输出波形

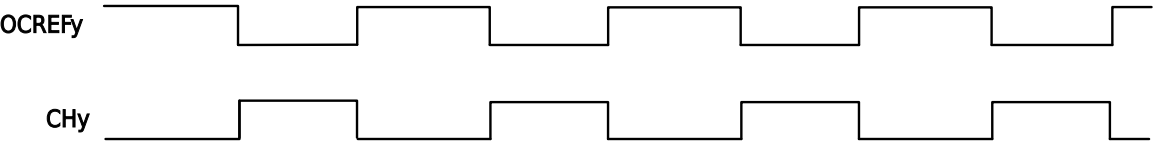


图 13-40 CCPy=1 时 PWM 输出波形

13.3.6.7 互补 PWM 输出

由 OCREFA 控制 CHA 的输出，OCREFA 同时控制 CHB 的输出。比较寄存器 CCRxB 可以作为专用比较控制 ADC 触发。

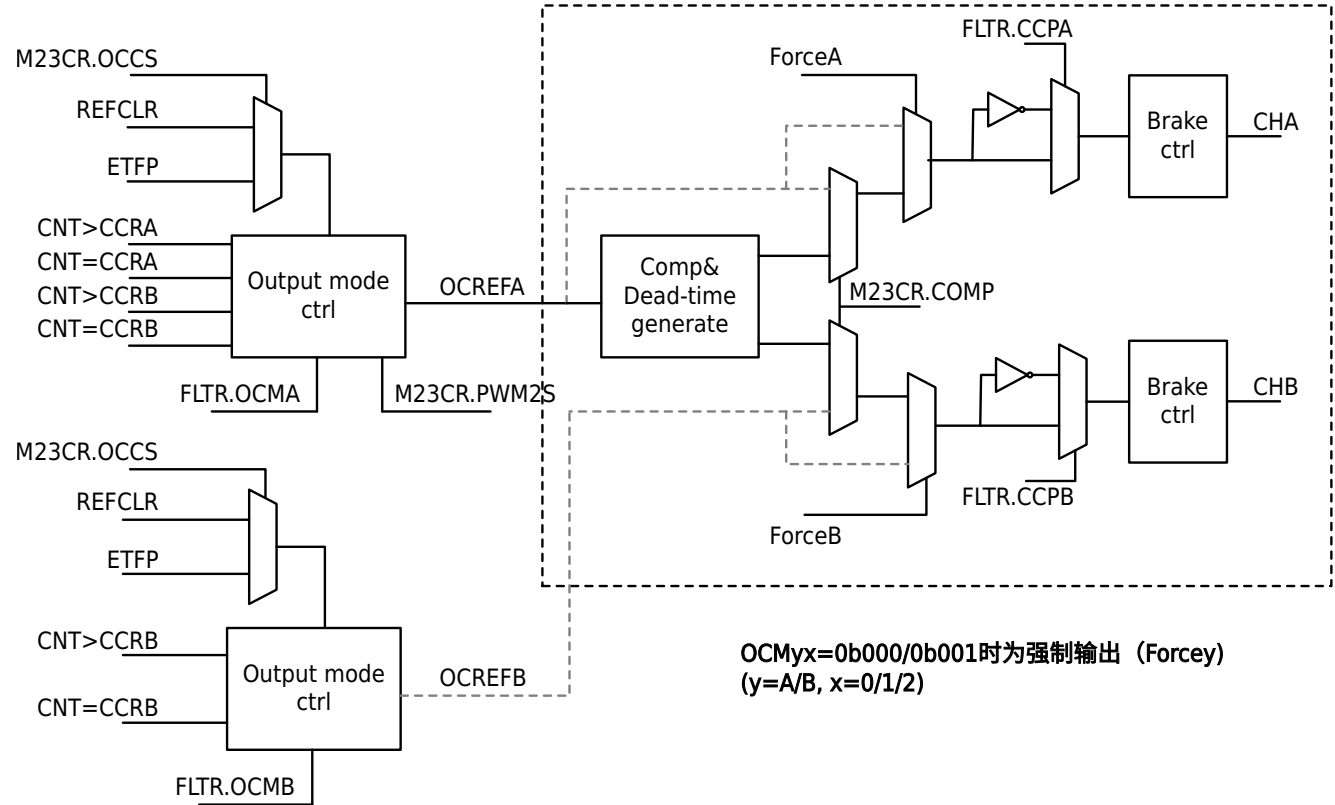


图 13-41 互补 PWM 输出框图

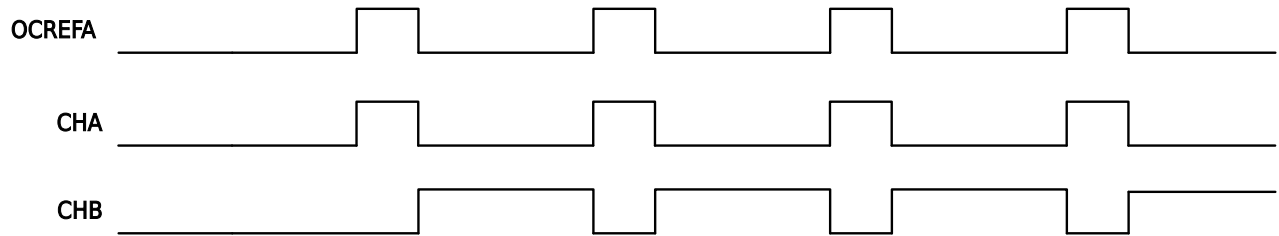


图 13-42 互补 PWM 输出波形图

13.3.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

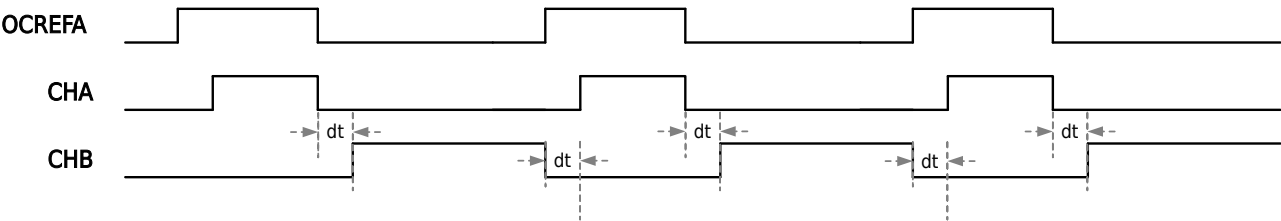


图 13-43 互补 PWM 输出波形图（有死区）

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下：

DTR	T	dt	step
DTR[7]=0b0	DTR[6:0]+2	2~129	1
DTR[7:6]=0b10	{DTR[5:0]+64}*2+2	130~256	2
DTR[7:5]=0b110	{DTR[4:0]+32}*8+2	258~506	8
DTR[7:5]=0b111	{DTR[4:0]+32}*16+2	514~1010	16

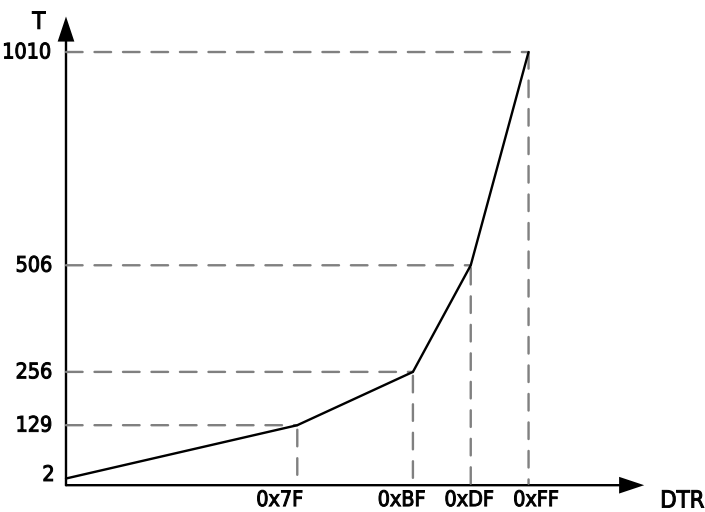


图 13-44 死区时间

说明

当 TCLK 为 48MHz 且 M23CR.PRS 设置为 0 的时候，死区时间由 5.33μs 调整为 21.04μs。

13.3.6.9 单脉冲输出

单脉冲模式（ONESHOT）是上述 PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，产生一个脉宽可编程的脉冲，如下图所示。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_M23CR.ONESHOT 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0，上计数模式计数值不要设置到大于等于 ARR。只有当比较值与计数器初始值不同时，才能正确产生输出脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时: $CNT < CCRxy \leq ARR$ (特别注意 $CCRxy > 0$)
- 递减计数时: $CNT > CCRxy$

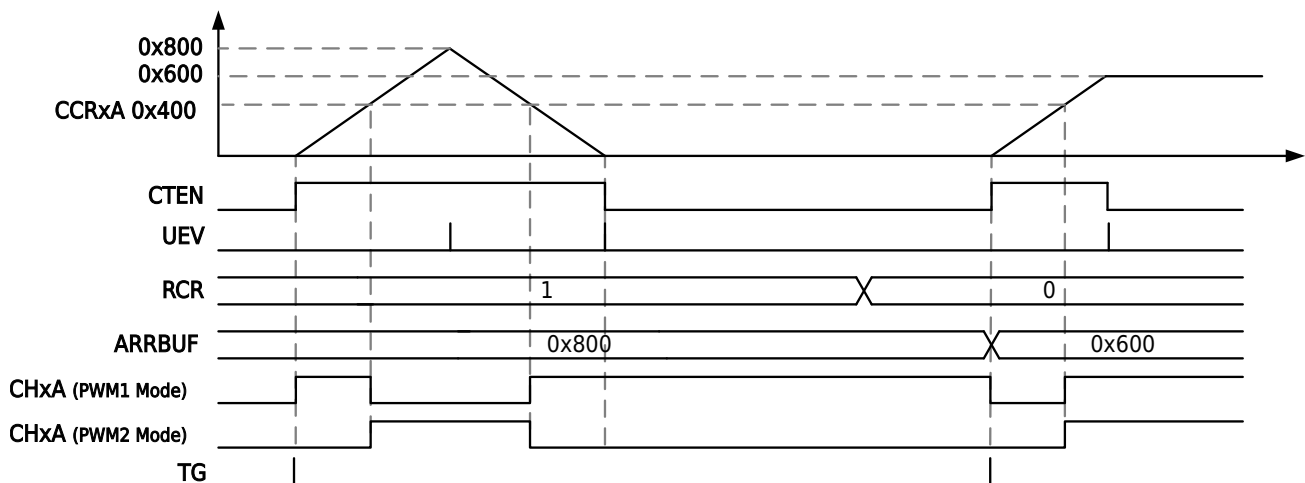


图 13-45 三角波模式单脉冲计数

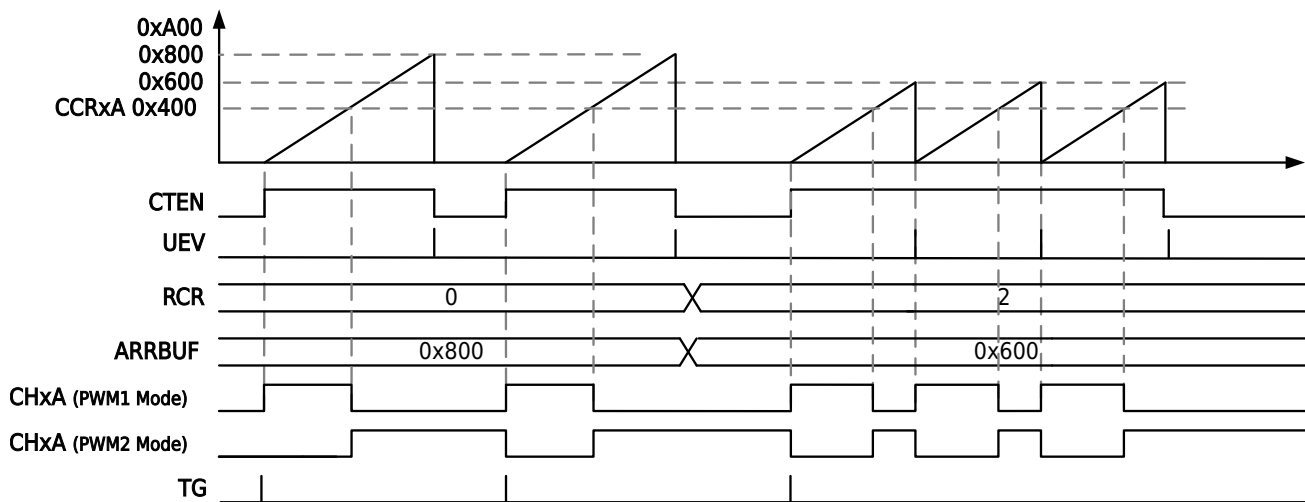


图 13-46 锯齿波上计数单脉冲模式

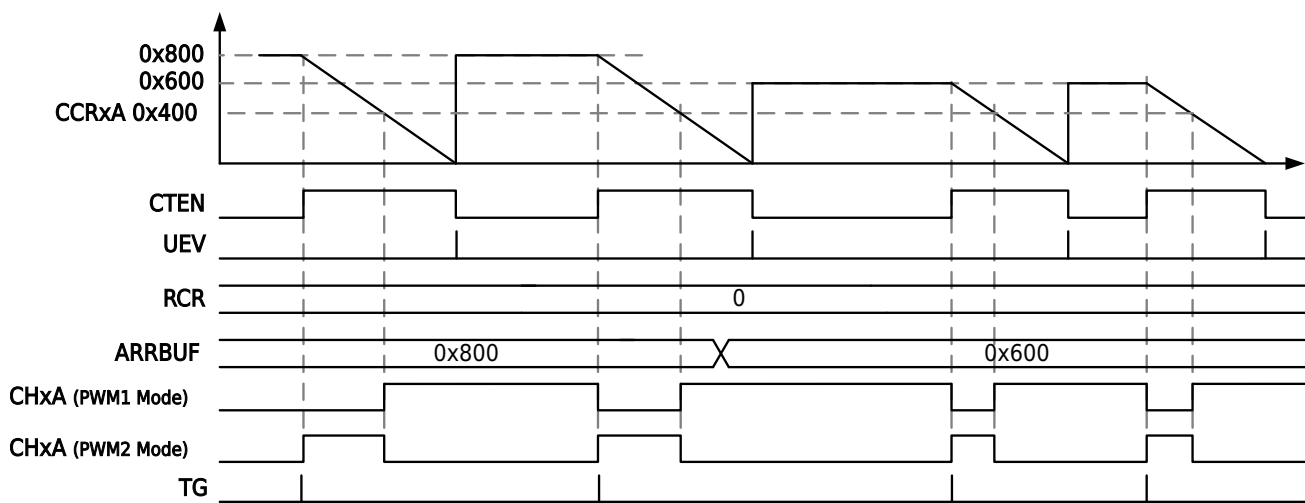


图 13-47 锯齿波下计数单脉冲模式



注意

上计数时，配置为 PWM2 模式时，由于通道初始化输出为低电平，所以第一次触发会产生一个上升沿并保持高电平，待到第二次触发时，可以输出一个脉宽可编程的低电平脉冲；下计数时，配置为 PWM1 模式时，由于通道初始化输出为低电平，所以第一次触发会产生一个上升沿并保持高电平，待到第二次触发时，可以输出一个脉宽可编程的低电平脉冲。

13.3.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0/1/2; y=A/B) 将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 M23CR.CIS 控制：

- M23CR.CIS=0b00 时，比较匹配无输出
- M23CR.CIS=0b01 时，上计数时比较匹配
- M23CR.CIS=0b10 时，下计数时比较匹配
- M23CR.CIS=0b11 时，上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制：

- CRCHx.CISB=0b00 时，通道 x 比较匹配无输出
- CRCHx.CISB=0b01 时，通道 x 上计数时比较匹配
- CRCHx.CISB=0b10 时，通道 x 下计数时比较匹配
- CRCHx.CISB=0b11 时，通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。更多信息请参见 [Timer 触发 ADC](#) 章节相关描述。

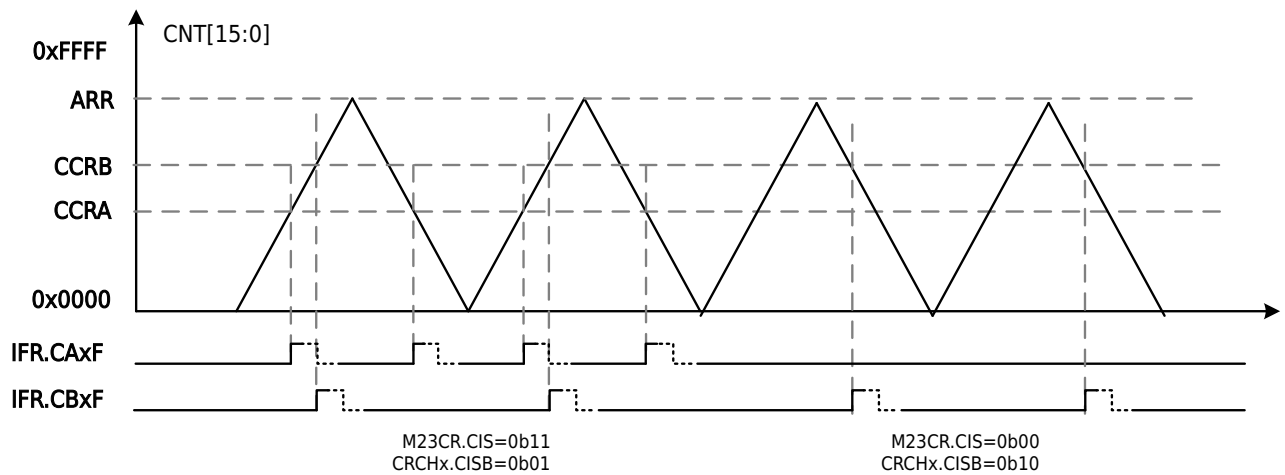


图 13-48 比较中断示意图

13.3.6.11 捕获输入

在三角波计数或者锯齿波计数模式下都可以设置捕获功能（M23CR.MODE=2/3），可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。每个通道的捕获边沿可以单独设置，通过寄存器 CRCHx.CFy/CRy（x=0/1/2；y=A/B）选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

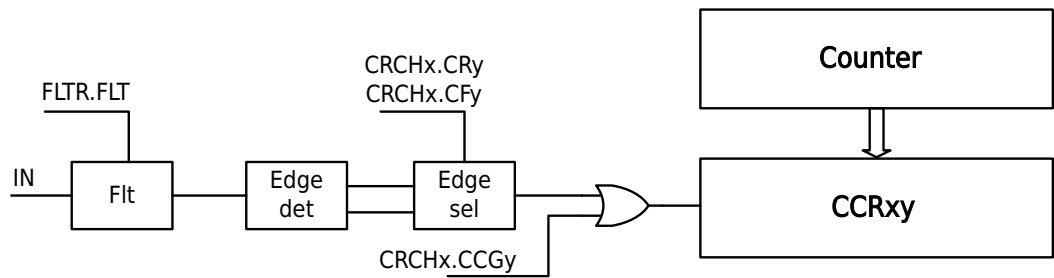


图 13-49 捕获功能框图

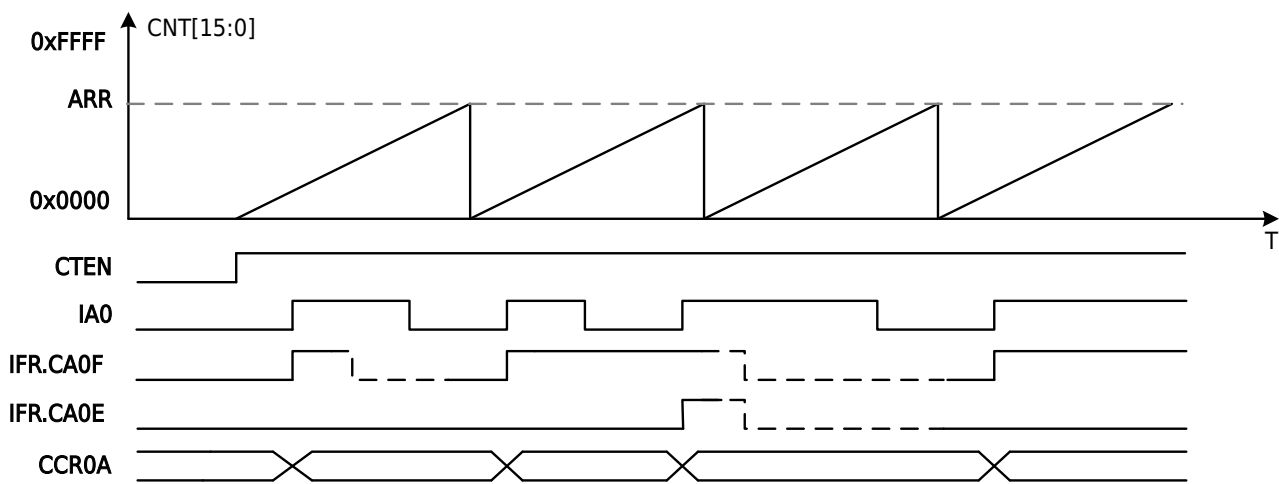


图 13-50 捕获时序图

CH0A 的捕获输入

CH0A 的捕获输入信号如下表所示。

表 13-8 CH0A 捕获输入信号

MSCR.IA0S	TIM0/1/2	TIM3
0b0	CH0A	CH0A
0b1	CH0A ETR GATE 异或输入	CH0A CH1A CH2A 异或输入

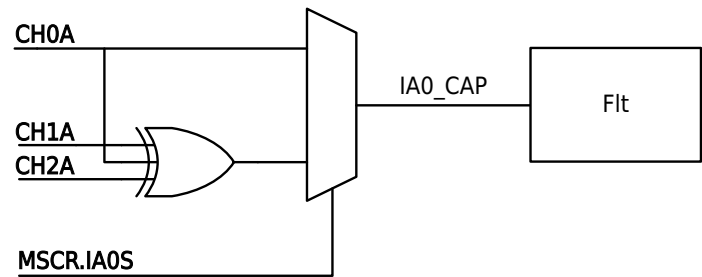


图 13-51 TIM3 CH0A 捕获输入端口选择

CH0B 的捕获输入

CH0B 的捕获输入信号如下表所示。

MSCR.IB0S	TIM0/1/2/3	
0b0	CH0B	
0b1	内部触发 MSCR.TS 选择信号	
	MSCR.TS 取值	通道 B 捕获输入
	0b000	ETR 滤波相位选择输出信号，滤波相位选择可配
	0b001	内部互联 ITR0
	0b010	内部互联 ITR1
	0b011	内部互联 ITR2
	0b100	内部互联 ITR3
	0b101	CH0A 边沿
	0b110	CH0A 滤波输出信号，滤波功能可配
	0b111	Reserved

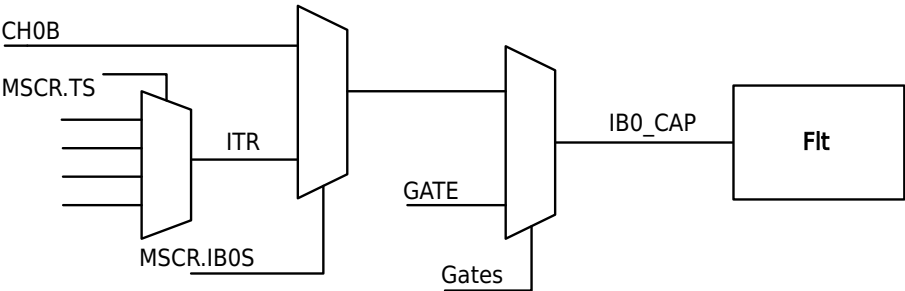


图 13-52 CH0B 捕获输入端口选择

Gates 信号为选择 PWM 互补输出功能生成的信号，当 PWM 为单点比较的互补输出时，比较捕获寄存器 CCR0B 没有使用，自动切换到使用 GATE 作为捕获/比较输出。

 **注意**

当 GATE 配置为比较输出时，GATE 的输出模式仅支持配置为强制输出为 0 或 1 和 PWM 模式有效（即 OCMB0 配置为 0/1/6/7）。

13.3.7 模式 2/3 从模式

定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

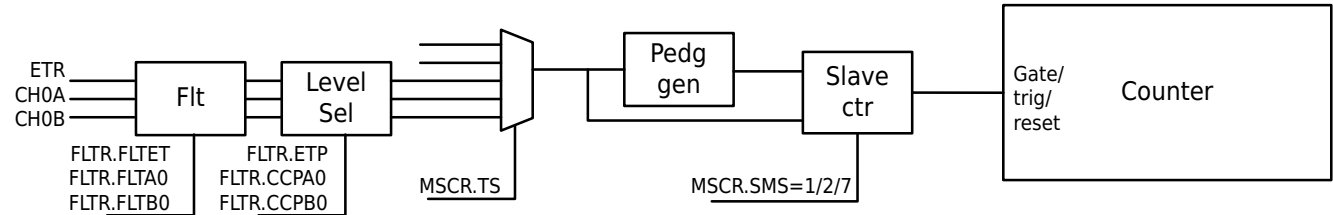


图 13-53 模式 2/3 从模式示意图

从模式功能选择通过 MSCR.SMS 选择：

	取值	说明
MSCR.SMS	0b001	从模式功能选择复位模式
	0b010	从模式功能选择触发模式
	0b111	从模式功能选择门控模式
MSCR.TS	0b000	端口 ETR 的滤波相位选择后的信号 ETFP，滤波功能可配置
	0b001	内部互联信号 ITR0
	0b010	内部互联信号 ITR1
	0b011	内部互联信号 ITR2
	0b100	内部互联信号 ITR3
	0b101	端口 CH0A 的边沿信号
	0b110	端口 CH0A 的滤波相位选择后的信号 IAFP，滤波功能可配置
	0b111	端口 CH0B 的滤波相位选择后的信号 IBFP，滤波功能可配置

门控计数

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 CH0A 为低电平时向上计数：

- 配置通道 CH0A 以检测 CH0A 上的低电平。配置输入滤波器带宽（本例中不需要滤波，所以保持 FLTR.FLTAE=0b000）。设置 FLTR.CCPAE=1 以确定极性（只检测低电平）。
- 设置 MSCR.SMS=0b111，配置定时器为门控模式；设置 MSCR.TS=0b110，选择 CH0A 作为输入源。
- 设置 M23CR.CTEN=1，启动计数器。在门控模式下，如果 CTEN=0，则计数器不能启动，不论触发输入电平如何。只要 CH0A 为低，计数器开始依据内部时钟计数，一旦 CH0A 变高则停止计数。



说明

门控输入不能是脉冲信号。当门控计数时，TS 不能配置为 0b101；当 TS 配置为 0b001/0b010/0b011/0b100 时，需要注意 ITR 是否为脉冲信号。

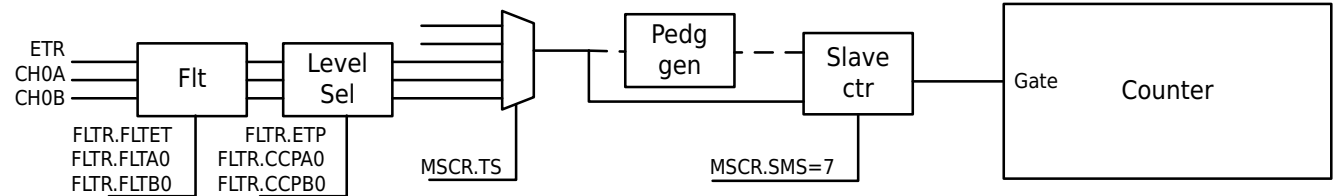


图 13-54 模式 2/3 从模式门控功能

触发功能

使用外部触发（CH0A、CH0B、ETR）可以使定时器同步启动，使用定时器内部互联信号结合 MSCR.MSM 也可以配置定时器同步启动，相关操作可参见[定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）示例](#)和[定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM1 的 UG 触发 TIM0/2）示例](#)。触发信号为输入信号的上升沿。也可以使用软件写 M23CR.TG 启动软件触发功能。

输入端上选中的事件使能计数器。在下面的例子中，计数器在 CH0B 输入的上升沿开始向上计数：

- 配置通道 CH0B 检测 CH0B 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 FLTR.FLTB0=0b000）。设置 FLTR.CCPB0=0 以确定极性（不反向）。
- 设置 MSCR.SMS=0b010，配置定时器为触发模式；设置 MSCR.TS=0b111，选择 CH0B 作为输入源。当 CH0B 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。CH0B 上升沿和计数器启动计数之间的延时，取决于 CH0B 输入端的同步电路。



说明

如果使用下降沿触发，先选择触发极性，然后再选择模式，否则会产生误触发。

复位计数

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 M23CR.URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器（ARR，CCRxy）都会被更新。

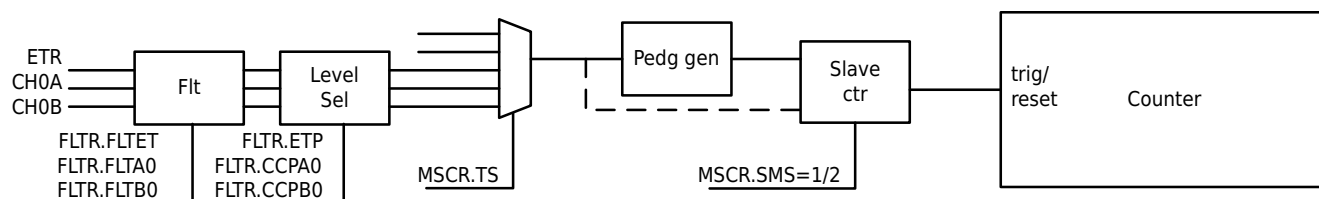


图 13-55 模式 2/3 从模式触发和复位功能

13.3.8 正交编码计数功能

MSCR.SMS=4/5/6, 对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP 和 IBFP 的相位关系进行编码计数。IAFP、IBFP 为端口输入 CH0A、CH0B 的滤波相位选择的信号。

模式 1 使用 CH0A 的边沿计数。模式 2 使用 CH0B 的边沿计数。模式 3 使用 CH0A 和 CH0B 的边沿都计数。为了保证计数相位的正确性，需要保证 A/B 输入的相位大于一个脉冲宽度的相位差，A/B 输入脉冲宽度需要大于两个脉冲宽度。

如下表所示，IAFP 边沿条件对应 IBFP 电平信号；IBFP 边沿条件对应 IAFP 电平信号。例如 MOD1，当 IBFP 为高电平时候，IAFP 被检测到上升沿，则计数器向下计数；当 IBFP 为高电平的时候，IAFP 被检测到下降沿，则计数器向上计数。

模式	正交信号电平条件		正交信号 IAFP 边沿条件		正交信号 IBFP 边沿条件	
	IBFP	IAFP	上升沿	下降沿	上升沿	下降沿
MOD1	高电平	-	向下计数	向上计数	-	-
	低电平	-	向上计数	向下计数	-	-
MOD2	-	高电平	-	-	向上计数	向下计数
	-	低电平	-	-	向下计数	向上计数
MOD3	高电平	-	向下计数	向上计数	-	-
	低电平	-	向上计数	向下计数	-	-
	-	高电平	-	-	向上计数	向下计数
	-	低电平	-	-	向下计数	向上计数

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号，IAFP/IBFP 为端口滤波相位选择后的信号。

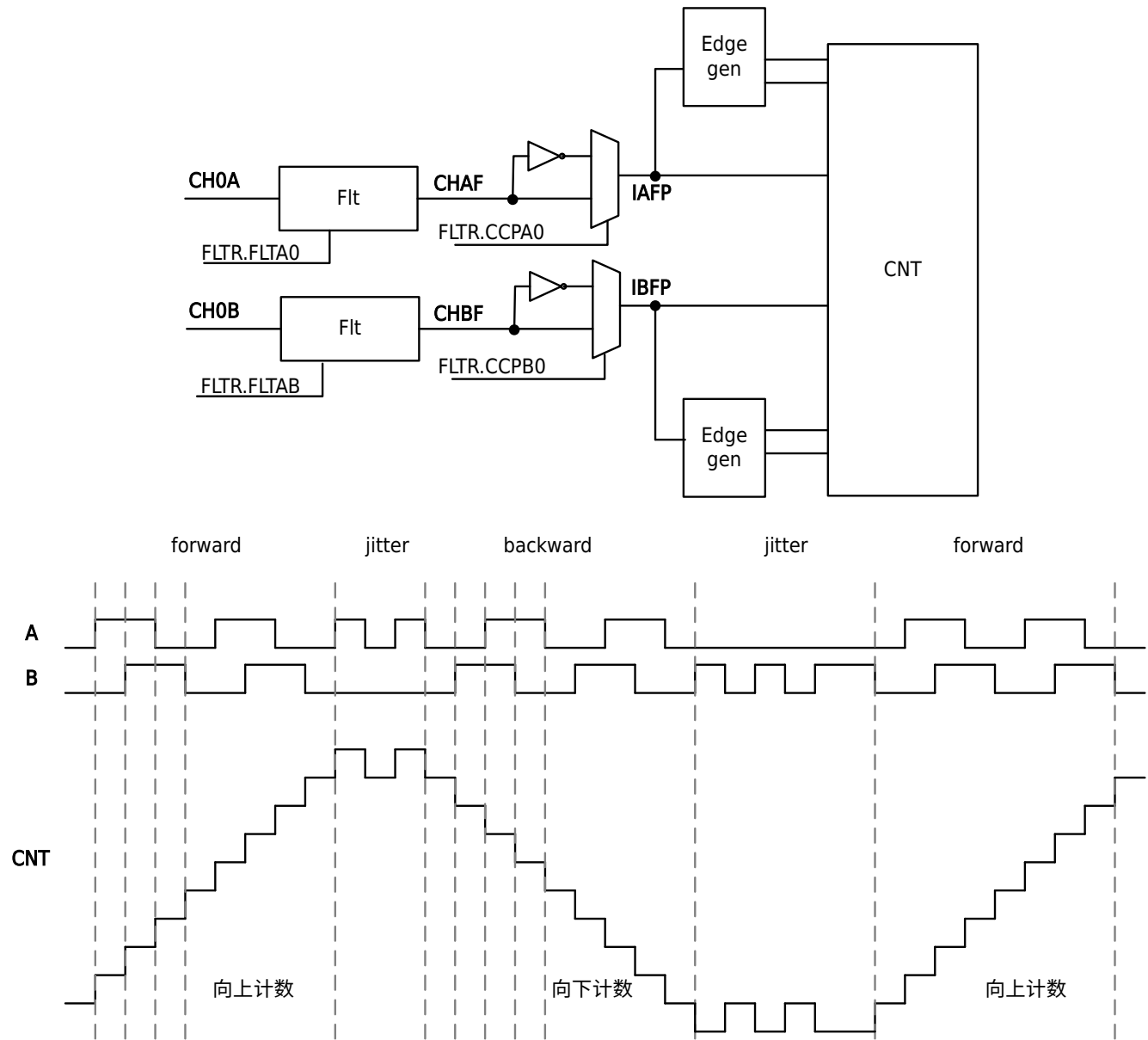


图 13-57 正交编码计数模式 3 计数示意图

13.3.9 Timer 触发 ADC

UEV 触发 ADC

当产生事件更新 UEV，且 ADTR.UEVE 和 ADTR.ADTE 使能，可以向 ADC 输出触发请求。

比较匹配触发 ADC

CCR0A、CCR1A、CCR2A 比较匹配可以配置触发 ADC：

- 模式 3：通过控制寄存器 M23CR.CIS 统一控制 CHxA 通道的比较匹配条件：可以选择上计数比较匹配、下计数比较匹配或上下计数都比较匹配触发。
- 模式 2：不需要设置 M23CR.CIS。

CCR0B、CCR1B、CCR2B 比较匹配可以配置触发 ADC：

- 模式 3：通过寄存器 CRCHx.CISB 分别控制 CHxB 通道的比较匹配条件：可以选择上计数比较匹配、下计数比较匹配或上下计数都比较匹配触发。
- 模式 2：不需要设置 CRCHx.CISB。

CCR3A、CCR3B 比较匹配也可以配置触发 ADC：

- 模式 3: CRCH3A.CIS 和 CRCH3B.CIS 分别控制通道 3A 和通道 3B 的比较匹配条件: 可以选择上计数比较匹配、下计数比较匹配或上下计数都比较匹配触发。
- 模式 2: 不需要设置 CRCH3A.CIS 和 CRCH3B.CIS。

Timer 的 TRGO 触发信号可以触发 ADC, 关于 TRGO 输出请参见寄存器 [TIM_MSCR.MMS](#) 位功能描述。

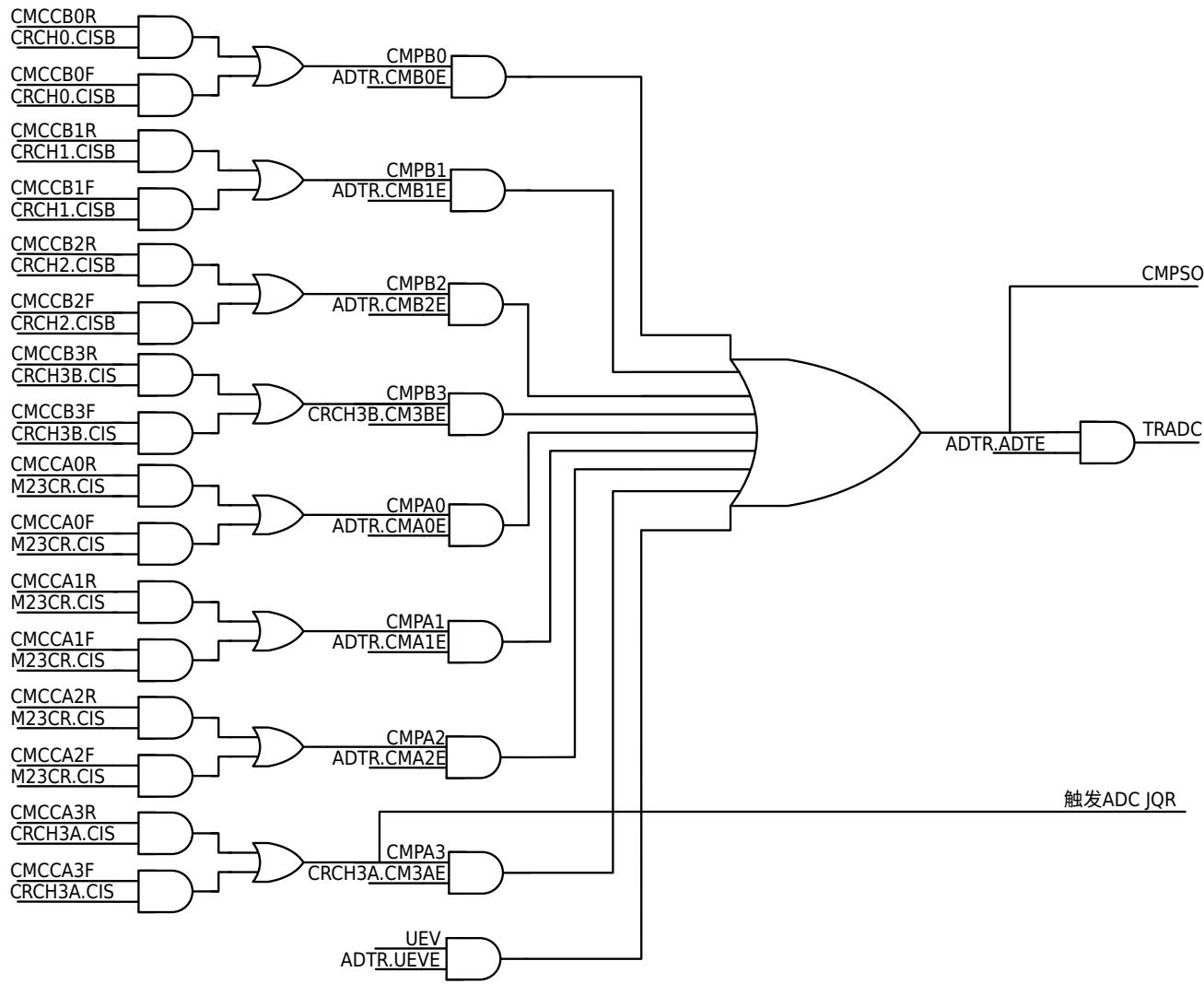


图 13-58 Timer 触发 ADC

13.3.10 逐波限流和刹车控制

逐波限流

M23CR.OCCE 可以选择使能或禁止逐波限流功能。M23CR.OCCS 可以选择 OCREF_CLR 信号来自 VC 输出的信号 REFCLR 或者来自 ETR 端口滤波相位选择后的信号 ETFP。

VC0/1 的 CR2.TIMxCLR (x=0/1/2/3) 设置是否使能 REFCLR 信号到 TIM0/1/2/3。CR2.POL 决定 REFCLR 信号高电平有效 (CR2.POLR=0) 还是低电平有效 (CR2.POL=1)。FLTR.ETP 设置输入 ETR 端口输入的电平是高电平有效 (FLTR.ETP=0) 还是低电平有效 (FLTR.ETP=1)，FLTR.FLTET 设置 ETR 的滤波时间。

当 DTR.AOE=0, DTR.MOE=1 时, M23CR.OCCE 使能后, 有效的 OCREF_CLR 信号可以将当前的 OCREFxy (x=0/1/2, y=A/B) 信号清零 (该清零动作在 FLTR.OCMyx>1 (x=0/1/2, y=A/B) 时有效), 而 DTR.MOE 不会被清零。当 OCREF_CLR 信号失效后, 下一个事件更新 UEV 后 PWM 继续输出。

刹车控制

刹车控制可以通过以下方式实现:

- 1. VC 比较器控制：VC0/1 的 CR2.POL 决定 VC 输出刹车电平信号是高电平有效（CR2.POL 为 0）还是低电平有效（CR2.POL 为 1）。将 VCx_CR2.TIMBK 使能，并且将 DTR.VCE 使能，当 VC 输出有效电平的时候，产生有效的刹车条件，可以控制刹车功能。
- 2. 指定的外部端口控制：将这个指定端口配置为 TIMx_BK（x=0/1/2/3）复用功能，在 FLTR.BKP 设置刹车有效电平（FLTR.BKP=0 为高电平刹车，FLTR.BKP=1 为低电平刹车），在 FLTR.FLTBK 设置输入滤波时间，使能 DTR.BKE 端口刹车。当有效电平从该端口输入，则刹车条件有效，控制刹车功能。
- 3. 系统 Safety 刹车（XTH fail, lockup）控制。
- 4. M23CR.BG 软件刹车控制。

当 DTR.AOE=0，DTR.MOE=1 时，刹车条件生效之后，DTR.MOE 位会被清零，PWM 引脚按照 CRCHx.BKSA/CRCHx.BKSB（x=0/1/2）的预设配置指定动作。如果要恢复 PWM 输出，需要待刹车条件失效后，将 DTR.MOE 重新置位。

当 DTR.AOE=1，DTR.MOE=1 时，刹车条件生效之后，DTR.MOE 位会被清零，PWM 引脚按照 CRCHx.BKSA/CRCHx.BKSB 的预设配置指定动作。待刹车条件失效后，DTR.MOE 自动置位，PWM 恢复输出。

两通道的输出的 TIM0/1/2 可以选择使用 TIM0 的刹车端口控制 TIM1/2 的刹车功能，也可以使用各自的刹车输入控制各自的刹车功能。当 DTR.BKSEL 设置为 1 时，TIM1/2 共用 TIM0 的刹车输入。

13.3.11 中断

表 13-13 TIM0/1/2/3 中断列表

中断控制寄存器	中断名
MxCR.UIE(x=0/1/2/3)	溢出中断/事件更新中断
CRCHx.CIEA(x=0/1/2)	通道（0/1/2）A 比较捕获中断
CRCHx.CIEB(x=0/1/2)	通道（0/1/2）B 比较捕获中断
CRCH3y.CIE(y=A/B)	通道 3A/B 比较中断
M23CR.BIE	刹车中断
M23CR.TIE	触发中断
M23CR.OVE	上溢中断
M23CR.UNDE	下溢中断

13.3.12 Timer 互联

TRGO 输出信号可以连接到其他定时器的 ITR 信号。连接关系如下：

表 13-14 定时器互联信号列表

	ITR0	ITR1	ITR2	ITR3
Timer0	-	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO
Timer1	TIM0_TRGO	-	TIM2_TRGO	TIM3_TRGO
Timer2	TIM0_TRGO	TIM1_TRGO	-	TIM3_TRGO
Timer3	TIM0_TRGO	TIM1_TRGO	TIM2_TRGO	-

13.3.13 输入互联

13.3.13.1 GATE 输入互联

当 GPIO_TIMGS.TIMx_G=0b000 时，门控 GATE 输入为 Pxy_SEL 选择的端口输入；当 GPIO_TIMGS.TIMx_G=0b001~0b111 时，连接其他模块的输入或输出。详细信息请参见《端口控制器 (GPIO)》章节寄存器“端口辅助功能定时器门控选择 (GPIO_TIMGS)”相关描述。

13.3.13.2 ETR 输入互联

当 GPIO_TIMES.TIMx_E=0b000 时，ETR 输入为 Pxy_SEL 选择的端口输入；当 GPIO_TIMES.TIMx_E=0b001~0b111 时，连接其他模块的输入或输出。详细信息请参见《端口控制器 (GPIO)》章节寄存器“端口辅助功能定时器 ETR 选择 (GPIO_TIMES)”相关描述。

13.3.13.3 CHx 捕获输入互联

当 GPIO_TIMCPS0.TIMx_Cy=0b000 时，捕获输入 (Timer0/1/2 的 CHA，Timer3 的 CH0A/CH0B) 为 Pxy_SEL 选择的端口输入，当 GPIO_TIMCPS0.TIMx_Cy=0b001~0b111 时，连接其他模块的输入或输出。详细信息请参见《端口控制器 (GPIO)》章节寄存器“端口辅助功能定时器捕获输入选择 0 (GPIO_TIMCPS0)”相关描述。

13.3.14 DMA

Timer 支持软件和硬件触发 DMA 进行数据传输。支持数据从其他位置写入定时器，或从定时器读出写入其他位置。可应用于数据捕获后数据的自动搬运和更改周期值或占空比的脉冲宽度的自动调整。每个定时器有两个 DMA 请求，A 请求触发源可选择比较捕获 A、外部触发 ETR；B 请求触发源可选择比较捕获 B、事件更新 UEV。

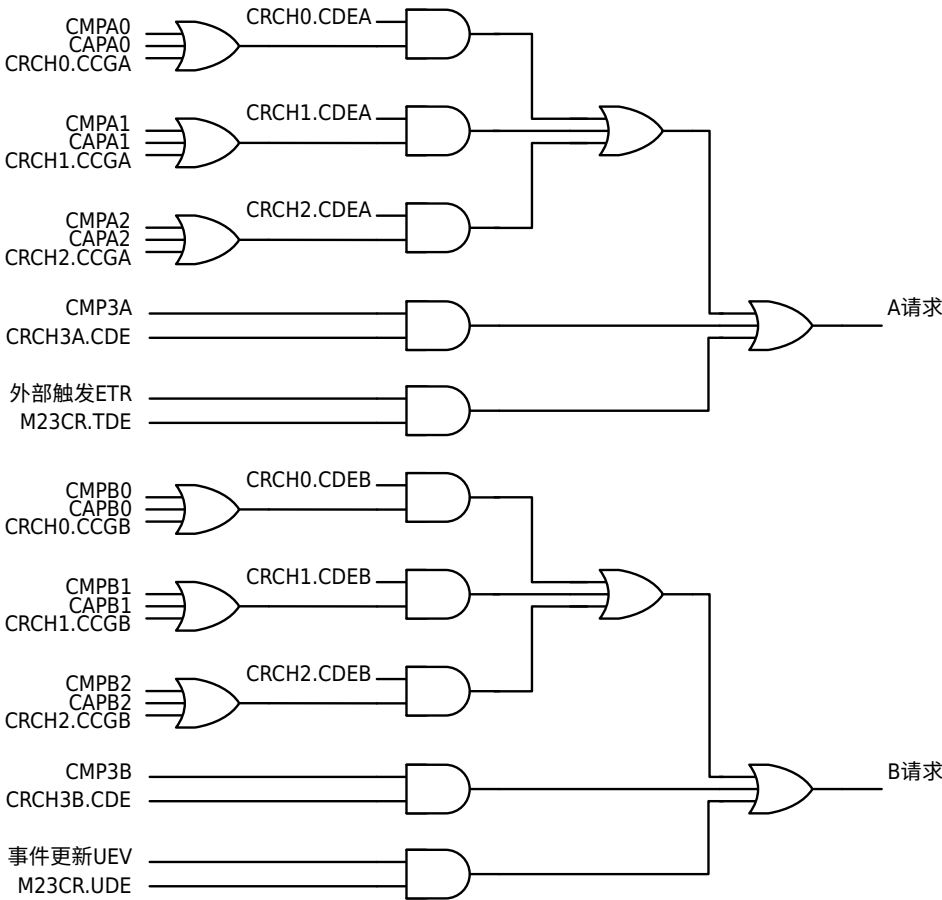


图 13-59 DMA 请求触发源

13.4 操作示例

13.4.1 模式 0 计数定时器功能设置示例

操作步骤

- **重载定时器设置**
 - 步骤 1. 设置定时器模式 M0CR.MODE=0;
 - 步骤 2. 设置装载值 ARR;
 - 步骤 3. 设置计数器初值 CNT;
 - 步骤 4. 清除中断标志;
 - 步骤 5. 使能中断 M0CR.UIE;
 - 步骤 6. 使能重载模式 M0CR.MD;
 - 步骤 7. 开启定时器 M0CR.CTEN。
- **门控外部时钟自由计数设置**
 - 步骤 1. 设置定时器模式 M0CR.MODE=0;
 - 步骤 2. 设置计数器初值 CNT;
 - 步骤 3. 使能门控功能 M0CR.GATE;
 - 步骤 4. 选择门控有效电平 M0CR.GATEP;
 - 步骤 5. 清除中断标志;
 - 步骤 6. 使能中断 M0CR.UIE;
 - 步骤 7. 使能外部时钟模式 M0CR.CT;
 - 步骤 8. 开启定时器 M0CR.CTEN。
- **BUZZER 输出控制**
 - 步骤 1. 根据输出频率设置合适的 ARR 值;
 - 步骤 2. 设置定时器为重载模式, 参考[重载定时器设置](#);
 - 步骤 3. 使能输出使能 DTR.MOE;
 - 步骤 4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

13.4.2 模式 1 脉宽测量 PWC 设置示例

操作步骤

- **脉冲低电平测量设置**
 - 步骤 1. 设置为脉冲测量模式 M1CR.MODE=1;
 - 步骤 2. 设置 MSCR.TS 选择测量的信号, 再依据 MSCR.TS 设置, 配置 MSCR.IA0S/IB0S (IB0S 不可配置为 1), 选择需要测量的通道;
 - 步骤 3. 设置 M1CR.EDG2ND=0, M1CR.EDG1ST=1 选择测量低电平;
 - 步骤 4. 清除中断标志;
 - 步骤 5. 使能溢出中断 M1CR.UIE;
 - 步骤 6. 使能测量结束中断 CR0.CIEA;
 - 步骤 7. 使能定时器 M1CR.CTEN;
 - 步骤 8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志;
 - 步骤 9. 等待下次测量。
- **脉冲高电平单次测量设置**
 - 步骤 1. 设置为脉冲测量模式 M1CR.MODE=1;
 - 步骤 2. 设置 MSCR.TS 选择测量的信号, 再依据 MSCR.TS 设置, 配置 MSCR.IA0S/IB0S (IB0S 不可配置为 1), 选择需要测量的通道;

- 步骤 3. 设置脉冲单次测量模式 M1CR.ONESHOT=1;
- 步骤 4. 设置 M1CR.EDG2ND=1, M1CR.EDG1ST=0 选择测量高电平;
- 步骤 5. 清除中断标志;
- 步骤 6. 使能溢出中断 M1CR.UIE;
- 步骤 7. 使能测量结束中断 CR0.CIEA;
- 步骤 8. 使能定时器 M1CR.CTEN;
- 步骤 9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志;
- 步骤 10. 测量结束。

13.4.3 模式 2/3 比较捕获模式设置示例

操作步骤

- 边沿对齐独立 PWM 输出设置

- 步骤 1. 设置模式 M23CR.MODE=2;
- 步骤 2. 设置锯齿波计数方向 M23CR.DIR;
- 步骤 3. 设置 PWM 周期值 ARR;
- 步骤 4. 设置计数器初值 (初值必须小于周期值);
- 步骤 5. 设置 PWM 比较值 CCRxA、CCRxB;
- 步骤 6. 设置 PWM 输出比较模式 FLTR.OCMAx、FLTR.OCMBx 为 6 或 7;
- 步骤 7. 清除相关中断标志;
- 步骤 8. 使能相应的中断;
- 步骤 9. 设置输出极性 FLTR.CCPAx、FLTR.CCPBx;
- 步骤 10. 使能输出 DTR.MOE;
- 步骤 11. 使能定时器 M23CR.CTEN。

- 中心对齐互补 PWM 输出设置

- 步骤 1. 设置模式 M23CR.MODE=3;
- 步骤 2. 设置互补输出 M23CR.COMP;
- 步骤 3. 设置 PWM 周期值 ARR;
- 步骤 4. 设置计数器初值 (初值必须小于周期值);
- 步骤 5. 设置 PWM 比较值 CCRxA (CCRxB 不需要设置, PWM 输出与这个寄存器无关);
- 步骤 6. 设置 PWM 输出比较模式 FLTR.OCMAx、FLTR.OCMBx 为 6 或 7;
- 步骤 7. 清除相关中断标志;
- 步骤 8. 使能相应的中断;
- 步骤 9. 设置输出极性 FLTR.CCPAx、FLTR.CCPBx;
- 步骤 10. 使能输出 DTR.MOE;
- 步骤 11. 使能定时器 M23CR.CTEN。

- 非对称中心对齐带死区互补 PWM 输出设置

- 步骤 1. 设置模式 M23CR.MODE=3;
- 步骤 2. 设置互补输出 M23CR.COMP;
- 步骤 3. 设置双点比较使能 M23CR.PWM2S;
- 步骤 4. 设置 PWM 周期值 ARR;
- 步骤 5. 设置计数器初值 (初值必须小于周期值);
- 步骤 6. 设置 PWM 比较值 CCRxA、CCRxB (上计数比较点为 CCRxA, 下计数比较点为 CCRxB);
- 步骤 7. 设置 PWM 输出比较模式 FLTR.OCMAx、FLTR.OCMBx 为 6 或 7;
- 步骤 8. 清除相关中断标志;

- 步骤 9. 使能相应的中断；
- 步骤 10. 设置输出极性 FLTR.CCPAx、FLTR.CCPBx；
- 步骤 11. 设置死区使能 DTR.DTEN；
- 步骤 12. 设置死区时间 DTR.DTR；
- 步骤 13. 使能输出 DTR.MOE；
- 步骤 14. 使能定时器 M23CR.CTEN。
- **捕获功能设置**，设置 CH0B 为上升沿捕获功能
 - 步骤 1. 选择计数器计数方式，设置 M23CR.MODE=2；
 - 步骤 2. 选择 CH0B 为捕获模式 CRCH0.CSB=1；
 - 步骤 3. 根据需求选择输入滤波 FLTR.FLTB0；
 - 步骤 4. 选择捕获的边沿 CRCH0.CRB=1；
 - 步骤 5. 设置 ARR 改变周期值；
 - 步骤 6. 清除相关中断标志；
 - 步骤 7. 清除捕获标志，使能相应中断；
 - 步骤 8. 启动定时器 M23CR.CTEN；
 - 步骤 9. 查询到捕获标志后，读取 CCR0B 获得捕获值。

**说明**

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

- **互补 PWM 输出时使用 GATE 作为 PWM 输出功能**
 - 步骤 1. 设置为 PWM 互补输出模式，相关步骤参考[中心对齐互补 PWM 输出设置](#)；
 - 步骤 2. 设置 CCR0B，设置 GATE PWM 比较值；
 - 步骤 3. 设置 FLTR.OCMB0=6/7，设置 PWM 输出模式；
 - 步骤 4. 使能 PWM 输出 DTR.MOE；
 - 步骤 5. 启动定时器 M23CR.CTEN。
- **互补 PWM 输出时使用 GATE 下降沿作为捕获输入功能**
 - 步骤 1. 设置为 PWM 互补输出模式，相关步骤参考[中心对齐互补 PWM 输出设置](#)；
 - 步骤 2. 设置 GATE 作为捕获输入 M23CR.CSG；
 - 步骤 3. 选择 GATE 捕获输入边沿设置 M23CR.CFG（GATE 作为捕获输入 B 通道滤波设置无效）；
 - 步骤 4. 清除相关中断标志；
 - 步骤 5. 清除捕获标志，使能相应中断；
 - 步骤 6. 启动定时器 M23CR.CTEN；
 - 步骤 7. 查询到捕获标志后，读取 CCR0B 获得捕获值。

13.4.4 互补 PWM 输出时 B 比较作为 ADC 触发功能

操作步骤

- 步骤 1. 设置为 PWM 互补输出模式，相关步骤参考[中心对齐互补 PWM 输出设置](#)
- 步骤 2. 设置 CCRxB 设置 ADC 触发比较点
- 步骤 3. 设置 CRCHx.CISB 选择上计数下计数比较匹配（锯齿波不需要选择）
- 步骤 4. 选择 ADC 触发比较的源 ADTR
- 步骤 5. 使能 ADC 触发 ADTR.ADTE

13.4.5 定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM0 的 CTEN 触发 TIM1/2）示例

操作步骤

- 步骤 1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
- 步骤 2. 设置 TIM0 为主模式 MSCR.MSM=1
- 步骤 3. 设置 TIM0 的 CTEN 触发另外两个定时器 MSCR.MMS=1
- 步骤 4. 保持 TIM1/2 的 MSCR.MSM=0
- 步骤 5. 设置 TIM1/2 为触发模式 MSCR.SMS=2
- 步骤 6. 选择 TIM1/2 的触发源为 TIM0 的 TRGO，MSCR.TS=1
- 步骤 7. 使能定时器 TIM0，启动定时器 M23CR.CTEN=1

13.4.6 定时器 TIM0/1/2 同时启动 PWM 输出设置（使用 TIM1 的 UG 触发 TIM0/2）示例

操作步骤

- 步骤 1. 参考 PWM 输出设置，设置定时器 0/1/2 的脉冲调整输出
- 步骤 2. 设置 TIM1 为触发主模式 MSCR.MSM=1
- 步骤 3. 设置 TIM1 的 UG 触发另外两个定时器 MSCR.MMS=0
- 步骤 4. 保持 TIM0/2 的 MSCR.MSM=0
- 步骤 5. 设置 TIM0/2 为触发模式 MSCR.SMS=2
- 步骤 6. 选择 TIM0/2 的触发源为 TIM0 的 TRGO，MSCR.TS=2
- 步骤 7. 使能定时器 TIM1，启动定时器 M23CR.UG=1

13.4.7 DMA 设置示例

操作步骤

● 捕获数据 DMA 数据传输

- 步骤 1. 使能 DMA
- 步骤 2. 使能 DMA 通道使能
- 步骤 3. 选择定时器 DMA 的通道
- 步骤 4. 设置传输类型、传输长度、传输方式
- 步骤 5. 设置源起始地址，目标起始地址
- 步骤 6. 设置源地址、目标地址的递增方式
- 步骤 7. 根据需要使能 DMA 中断
- 步骤 8. 参考数据捕获流程设置数据捕获
- 步骤 9. 使能 CRCHx.CDEy 使能捕获数据触发 DMA 传输



说明

源地址设置为通道比较捕获寄存器，源地址固定，目标地址变化。

● 脉冲调整 DMA 数据传输

- 步骤 1. 使能 DMA
- 步骤 2. 使能 DMA 通道使能
- 步骤 3. 选择定时器 DMA 的通道
- 步骤 4. 设置传输类型、传输长度、传输方式
- 步骤 5. 设置源起始地址，目标起始地址
- 步骤 6. 设置源地址、目标地址的递增方式
- 步骤 7. 根据需要使能 DMA 中断
- 步骤 8. 参考脉冲调整输出设置脉冲调制输出

步骤 9. 根据需要进行硬件触发 DMA 的条件 (UEV 触发或者比较匹配触发)**说明**

目标地址设置为通道比较捕获寄存器，源地址变化，目标地址固定。

13.5 寄存器

13.5.1 寄存器总表

- Timer0 基地址: 0x40003000
- Timer1 基地址: 0x40003400
- Timer2 基地址: 0x40003800
- Timer3 基地址: 0x40009800

表 13-15 模式 0 定时器寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	TIM_ARR	16 位模式重载寄存器/周期	32
0x004	TIM_CNT	16 位模式计数寄存器	32
0x008	TIM_CNT32	32 位模式计数寄存器	32
		 说明 TIM_CNT32 是 TIM_ARR 与 TIM_CNT 的重映射寄存器。为了方便 32 位定时器写操作。	
0x00C	TIM_MOCR	模式 0 控制寄存器	32
0x010	TIM_IFR	中断标志寄存器	32
0x014	TIM_ICLR	中断标志清除寄存器	32
0x030	TIM_DTR	死区时间寄存器	32

表 13-16 模式 1 脉宽测量 PWC 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x004	TIM_CNT	16 位模式计数寄存器	32
0x00C	TIM_M1CR	模式 1 控制寄存器	32
0x010	TIM_IFR	中断标志寄存器	32
0x014	TIM_ICLR	中断标志清除寄存器	32
0x018	TIM_MSCR	主从模式控制寄存器	32
0x01C	TIM_FLTR	输入滤波寄存器	32
0x024	TIM_CR0	控制寄存器	32
0x03C	TIM_CCR0A	比较捕获寄存器	32

表 13-17 模式 2/3 比较捕获模式寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	TIM_ARR	16 位模式重载寄存器/周期	32

偏移地址	寄存器	描述	位宽
0x004	TIM_CNT	16 位模式计数寄存器	32
0x00C	TIM_M23CR	模式 2/3 控制寄存器	32
0x010	TIM_IFR	中断标志寄存器	32
0x014	TIM_ICLR	中断标志清除寄存器	32
0x018	TIM_MSCR	主从模式控制寄存器	32
0x01C	TIM_FLTR	输出控制/输入滤波寄存器	32
0x020	TIM_ADTR	ADC 触发控制寄存器	32
0x024	TIM_CRCH0	通道 0 控制寄存器	32
0x028	TIM_CRCH1	通道 1 控制寄存器 (仅 TIM3 存在)	32
0x02C	TIM_CRCH2	通道 2 控制寄存器 (仅 TIM3 存在)	32
0x030	TIM_DTR	死区时间寄存器	32
0x034	TIM_RCR	重复周期设置寄存器	32
0x038	TIM_ARRDM	重载寄存器/周期映射地址	32
0x03C	TIM_CCR0A	通道 0A 比较捕获寄存器	32
0x040	TIM_CCR0B	通道 0B 比较捕获寄存器	32
0x044	TIM_CCR1A	通道 1A 比较捕获寄存器 (仅 TIM3 存在)	32
0x048	TIM_CCR1B	通道 1B 比较捕获寄存器 (仅 TIM3 存在)	32
0x04C	TIM_CCR2A	通道 2A 比较捕获寄存器 (仅 TIM3 存在)	32
0x050	TIM_CCR2B	通道 2B 比较捕获寄存器 (仅 TIM3 存在)	32
0x054	TIM_CCR3A	通道 3A 比较寄存器 (仅 TIM3 存在)	32
0x058	TIM_CRCH3A	通道 3A 控制寄存器 (仅 TIM3 存在)	32
0x05C	TIM_CCR3B	通道 3B 比较寄存器 (仅 TIM3 存在)	32
0x060	TIM_CRCH3B	通道 3B 控制寄存器 (仅 TIM3 存在)	32

13.5.2 模式 0 计数定时器寄存器

13.5.2.1 16 位模式重载寄存器 (TIM_ARR)

[illegible]

位/位域	标记	功能描述	读写
31:16	<i>Reserved</i>	-	-
15:0	ARR	16 位重载定时器重载值寄存器	RW

13.5.2.2 16 位模式计数寄存器 (TIM_CNT)

[illegible]

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CNT	16 位重载定时器计数值寄存器	RW

13.5.2.3 32 位模式计数寄存器 (TIM CNT32)

[illegible]

位/位域	标记	功能描述	读写
31:0	CNT32	32 位定时器计数值寄存器 仅在模式 0 32 位定时器自由计数模式下有效，其他模式禁止写该寄存器	RW

13.5.2.4 模式 0 控制寄存器 (TIM_M0CR)

Offset	Bit Position																																							
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00600008																																							
Name	Reserved																MODE		Reserved	UIE		GATEP		GATE		Reserved	PRS		TOGEN		CT		MD		CTEN					
Access																	RW			RW	RW	RW			RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:14	Reserved	-	-
13:12	MODE	工作模式 0b00: 定时器模式 0 0b01: PWC 模式 0b10: 锯齿波模式 0b11: 三角波模式	RW
11	Reserved	-	-
10	UIE	中断使能控制 0b0: 禁止 0b1: 使能	RW
9	GATEP	端口 GATE 极性控制 0b0: 端口 GATE 高电平有效 0b1: 端口 GATE 低电平有效	RW
8	GATE	定时器门控使能 0b0: 无门控, CTEN=1 时定时器工作 0b1: 门控使能, 端口 GATE 有效并且 CTEN=1 时定时器才工作	RW
7	Reserved	-	-
6:4	PRS	内部时钟分频选择 0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: 64 0b111: 256	RW
3	TOGEN	模式 0 下翻转输出使能 0b0: 翻转输出关闭, CHA、CHB 输出为低电平 0b1: 翻转输出使能	RW
2	CT	计数时钟选择 0b0: 内部计数时钟 TCLK 0b1: 外部计数时钟 ETR	RW

位/位域	标记	功能描述	读写
1	MD	模式选择 32 定时/16 定时模式选择 ● 0b0: 32 位自由计数 ● 0b1: 16 位重载计数	RW
0	CTEN	定时器使能 ● 0b0: 禁止 ● 0b1: 使能	RW

13.5.2.5 中断标志寄存器 (TIM_IFR)

Offset	Bit Position																																
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																																
Access																																RO	UIF

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	UIF	溢出中断 ● 0b0: 计数器未溢出 ● 0b1: 计数器已溢出	RO

13.5.2.6 中断标志清除寄存器 (TIM_ICLR)

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x003FFFFFF(TIM0/1/2);0x000FFFFFF(TIM3)																															
Name	Reserved																															UIF
Access																																WO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	UIF	溢出中断清除 写 0 清除标志，写 1 无效；读出恒为 1	WO

13.5.2.7 死区时间寄存器 (TIM_DTR)

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																			MOE	Reserved											
Access																					RW											

位/位域	标记	功能描述	读写
31:13	Reserved	-	-
12	MOE	翻转输出使能 ● 0b0：翻转输出为输入状态 ● 0b1：翻转端口为输出状态	RW
11:0	Reserved	-	-

13.5.3 模式 1 脉宽测量 PWC 寄存器

13.5.3.1 16 位模式计数寄存器 (TIM_CNT)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNT															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CNT	计数值寄存器	RW

13.5.3.2 模式 1 控制寄存器 (TIM_M1CR)

Offset	Bit Position																																	
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00600008																																	
Name	Reserved																		ONESHOT	MODE		Reserved	UIE		EDG2ND	EDG1ST	Reserved	PRS		Reserved	CT	Reserved	CTEN	
Access																			RW	RW		RW	RW	RW		RW		RW		RW				

位/位域	标记	功能描述	读写															
31:15	Reserved	-	-															
14	ONESHOT	单次触发模式选择 0b0: 循环测量 0b1: 完成一次脉冲测量自动结束，再次测量需要重新使能 CTEN	RW															
13:12	MODE	工作模式 0b00: 定时器模式 0 0b01: PWC 模式 0b10: 锯齿波模式 0b11: 三角波模式	RW															
11	Reserved	-	-															
10	UIE	中断使能控制 计数到 0xFFFF 才会溢出并产生溢出标志 0b0: 禁止 0b1: 使能	RW															
9	EDG2ND	脉冲宽度测量结束边沿选择 详见 EDG1ST 功能描述	RW															
8	EDG1ST	脉冲宽度测量开始边沿选择 <table><tr><th>EDG2ND</th><th>EDG1ST</th><th>测量</th></tr><tr><td>0b0</td><td>0b0</td><td>上沿-上沿周期</td></tr><tr><td>0b0</td><td>0b1</td><td>低电平宽度</td></tr><tr><td>0b1</td><td>0b0</td><td>高电平宽度</td></tr><tr><td>0b1</td><td>0b1</td><td>下沿-下沿周期</td></tr></table>	EDG2ND	EDG1ST	测量	0b0	0b0	上沿-上沿周期	0b0	0b1	低电平宽度	0b1	0b0	高电平宽度	0b1	0b1	下沿-下沿周期	RW
EDG2ND	EDG1ST	测量																
0b0	0b0	上沿-上沿周期																
0b0	0b1	低电平宽度																
0b1	0b0	高电平宽度																
0b1	0b1	下沿-下沿周期																
7	Reserved	-	-															
6:4	PRS	内部时钟分频选择 0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: 64 0b111: 256	RW															
3	Reserved	-	-															

位/位域	标记	功能描述	读写
2	CT	计数时钟选择 0b0: 内部计数时钟 TCLK 0b1: 外部计数时钟 ETR	RW
1	Reserved	-	-
0	CTEN	脉冲宽度测量使能 0b0: 禁止 0b1: 使能	RW

13.5.3.3 中断标志寄存器 (TIM_IFR)

Offset	Bit Position																																	
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																														CA0F		Reserved	
Access																															RO		RO	UIF

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2	CA0F	脉冲宽度测量中断标志 0b0: 脉冲宽度测量未完成 0b1: 脉冲宽度测量已完成	RO
1	Reserved	-	-
0	UIF	溢出中断标志 0b0: 计数器未溢出 0b1: 计数器已溢出	RO

13.5.3.4 中断标志清除寄存器 (TIM_ICLR)

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x003FFFFFF(TIM0/1/2);0x000FFFFFF(TIM3)																															
Name	Reserved																												CA0F		Reserved	UIF
Access																													WO		WO	

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2	CA0F	脉冲宽度测量中断标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
1	Reserved	-	-
0	UIF	溢出中断清除 写 0 清除标志，写 1 无效；读出恒为 1	WO

13.5.3.5 主从模式控制寄存器 (TIM_MSCR)

Offset	Bit Position																																		
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	Reserved																						Reserved						Reserved						
Access																					RW	IB0S	RW	IA0S					RW		TS				

位/位域	标记	功能描述	读写
31:13	Reserved	-	-
12	IB0S	CH0B 输入选择 0b0: CH0B 0b1: 内部触发 TS 选择信号	RW
11	IA0S	IA0 输入选择 0b0: CH0A 0b1: - CH0A CH1A CH2A XOR (TIM3) - CH0A ETR GATE XOR (TIM0/1/2)  说明 设置为 1 后端口的任意一个端口变化将导致输入变化。	RW
10:8	Reserved	-	-
7:5	TS	触发选择 0b000: 端口 ETR 的滤波相位选择后的信号 ETFP 0b001: 内部互联信号 ITR0 0b010: 内部互联信号 ITR1 0b011: 内部互联信号 ITR2 0b100: 内部互联信号 ITR3 0b101: 禁止设置 0b110: 端口 CH0A 的滤波相位选择后的信号 IAFP 0b111: 端口 CH0B 的滤波相位选择后的信号 IBFP	RW
4:0	Reserved	-	-

13.5.3.6 输入滤波寄存器 (TIM_FLTR)

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name				Reserved																						Reserved						
	ETP	FLTET																						FLTBO				FLTA0				
Access	RW	RW																						RW				RW				

位/位域	标记	功能描述	读写
31	ETP	ETR 输入相位选择 0b0: 同相位 0b1: 反向输入	RW
30:28	FLTET	ETR 滤波控制 采样时钟为 PCLK 或 PCLK 分频, 采样点个数为 N 0b000: 无滤波 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
27:7	Reserved	-	-
6:4	FLTBO	CHB 输入滤波控制 采样时钟为 PCLK 或 PCLK 分频, 采样点个数为 N 0b000: 无滤波 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$  说明 需要配置 CR0.CSB 为 1。	RW
3	Reserved	-	-
2:0	FLTA0	CHA 输入滤波控制 采样时钟为 PCLK 或 PCLK 分频, 采样点个数为 N 0b000: 无滤波 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$	RW

位/位域	标记	功能描述	读写
2:0	FLTA0	● 0b011: $F_{sample}=PCLK, N=6$ ● 0b100: $F_{sample}=PCLK/4, N=4$ ● 0b101: $F_{sample}=PCLK/4, N=6$ ● 0b110: $F_{sample}=PCLK/8, N=4$ ● 0b111: $F_{sample}=PCLK/8, N=6$  说明 需要配置 CR0.CSA 为 1。	RW

13.5.3.7 控制寄存器 (TIM_CR0)

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00003000																															
Name	Reserved																								CIEA	Reserved	CSB	CSA	Reserved			
Access																																

位/位域	标记	功能描述	读写
31:9	Reserved	-	-
8	CIEA	脉冲宽度测量完成中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
7:6	Reserved	-	-
5	CSB	B 通道捕获/比较功能选择 当脉冲宽度测量使用通道 B 滤波时, 需要设置 CSB 为 1 ● 0b0: 比较模式 ● 0b1: 捕获模式	RW
4	CSA	A 通道捕获/比较功能选择 当脉冲宽度测量使用通道 A 滤波时, 需要设置 CSA 为 1 ● 0b0: 比较模式 ● 0b1: 捕获模式	RW
3:0	Reserved	-	-

13.5.3.8 比较捕获寄存器 (TIM_CCR0A)

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CCR0A															
Access																	RO CCR0A															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CCR0A	脉冲宽度测量结果	RO

13.5.4 模式 2/3 比较捕获模式寄存器

13.5.4.1 16 位模式重载寄存器 (TIM_ARR)

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ARR															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	ARR	重载寄存器/周期寄存器 具有缓存功能，在计数器未使能或者缓存没有使能时，缓存寄存器可以立刻更新	RW

13.5.4.2 16 位模式计数寄存器 (TIM_CNT)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CNT															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CNT	重载定时器计数值寄存器 使用 PWM 比较输出，三角波模式或者锯齿波模式上计数时，初始化 CNT 值需要小于 ARR 的值	RW

13.5.4.3 模式 2/3 控制寄存器 (TIM_M23CR)

Offset	Bit Position																																		
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00600008																																		
Name	Reserved	UNDE	OVE	DIR	BG	UG	TG	OCCE	CIS	BIE	TIE	TDE	URS	OCCS	CSG	ONESHOT	MODE	UDE	UIE	CFG	CRG	BUFEN	PRS	PWM2S	CT	COMP	CTEN								
Access		RW	RW	RW/RO	WO	WO	WO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

位/位域	标记	功能描述	读写
31:30	Reserved	-	-
29	UNDE	下溢出中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
28	OVE	上溢出中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
27	DIR	计数方向 只有在锯齿波模式下可以写。其他模式下只读，写无效 ● 0b0: 上计数 ● 0b1: 下计数  说明 从其他模式切换到中心对齐模式 DIR 自动清零。中心对齐模式时，软件事件更新和从模式外部触发复位模式 DIR 自动清零。	RW/RO
26	BG	软件刹车 自动清零，写 1 产生软件刹车，写 0 无效	WO
25	UG	软件更新 自动清零，写 1 产生软件更新，写 0 无效 初始化计数器并更新缓存寄存器到相应寄存器（缓存使能），PRS 位也会被清零。	WO
24	TG	软件触发 自动清零，写 1 产生软件触发，写 0 无效 在触发模式 SMS=2 且 MODE=2/3 下都可以触发	WO
23	OCCE	OCREF 清除使能 ● 0b0: OCREF 输出不受 OCREF_CLR 影响 ● 0b1: OCREF_CLR 信号可以清除 OCREF 输出	RW

位/位域	标记	功能描述	读写
22:21	CIS	通道 A 比较匹配模式 ● 0b00: 无匹配 ● 0b01: 上计数时比较匹配 ● 0b10: 下计数时比较匹配 ● 0b11: 上/下计数时比较都匹配  说明 通道 B 比较匹配模式由 CRCHx 寄存器 CISB 位控制。 该位不影响 OCREF 的输出。	RW
20	BIE	刹车中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
19	TIE	触发中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
18	TDE	触发启动 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
17	URS	更新源 ● 0b0: 上溢出/下溢出/软件更新 UG/从模式复位 ● 0b1: 上溢出/下溢出	RW
16	OCCS	OCREF_CLR 信号源选择 ● 0b0: 电压比较器 VC 输出, VC 选择在 VCx_CR2 寄存器设置 ● 0b1: ETR 端口滤波相位选择后的信号 当 OCCE 有效时, OCREF_CLR 可以将 OCREF 的比较输出信号清零 (当 OCMyx>1 时有效), 下一个 uev 事件后继续比较输出	RW
15	CSG	GATE 在 PWM 互补模式下捕获/比较选择 只有在 PWM 互补输出时有效, 使用 CCR0B 作为 GATE 的比较或捕获通道 ● 0b0: 比较 ● 0b1: 捕获	RW
14	ONESHOT	单次触发模式选择 ● 0b0: 循环计数 ● 0b1: 发生事件更新后定时器停止	RW
13:12	MODE	工作模式 ● 0b00: 定时器模式 0 ● 0b01: PWC 模式 ● 0b10: 锯齿波模式 ● 0b11: 三角波模式	RW
11	UDE	更新触发 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
10	UIE	更新中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW

位/位域	标记	功能描述	读写
9	CFG	GATE 作为捕获输入时, 下沿捕获有效控制 只有在 PWM 互补输出时有效 ● 0b0: 下沿捕获无效 ● 0b1: 下沿捕获有效	RW
8	CRG	GATE 作为捕获输入时, 上沿捕获有效控制 只有在 PWM 互补输出时有效 ● 0b0: 上沿捕获无效 ● 0b1: 上沿捕获有效	RW
7	BUFPEN	自动重载 ARR 缓存使能 ● 0b0: 周期缓存无效, 写入后立刻影响周期值 ● 0b1: 周期缓存使能, 写入后下个事件更新后才影响到周期值	RW
6:4	PRS	内部时钟分频选择 ● 0b000: 1 ● 0b001: 2 ● 0b010: 4 ● 0b011: 8 ● 0b100: 16 ● 0b101: 32 ● 0b110: 64 ● 0b111: 256	RW
3	PWM2S	OCREFA 双点比较选择 ● 0b0: 双点比较使能, 使用 CCRA、CCRB 比较控制 OCREFA 输出 ● 0b1: 单点比较使能, 只使用 CCRA 比较控制 OCREFA 输出 (缺省值)  说明 OCREFB 不受影响, 仍然使用 CCRB 控制 OCREFB 输出。	RW
2	CT	计数时钟选择 ● 0b0: 内部计数时钟 TCLK ● 0b1: 外部计数时钟 ETR	RW
1	COMP	PWM 互补输出模式选择 ● 0b0: 独立 PWM 输出 ● 0b1: 互补 PWM 输出	RW
0	CTEN	定时器使能 ● 0b0: 禁止 ● 0b1: 使能 可以外部触发使能, 在 ONESHOT 模式下结束时该位自动清零	RW

13.5.4.4 中断标志寄存器 (TIM_IFR)

Offset	Bit Position																																									
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Reset	0x00000000																																									
Name	Reserved												CB3F	CA3F	UNDF	OVF	TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Reserved											
Access													RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO					

位/位域	标记	功能描述	读写
31:20	Reserved	-	-
19	CB3F	通道 CH3B 比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
18	CA3F	通道 CH3A 比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
17	UNDF	下溢出中断标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
16	OVF	上溢出中断标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
15	TIF	触发中断标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
14	BIF	刹车中断标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
13	CB2E	通道 CH2B 捕获数据丢失标志 仅 TIM3 存在 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO
12	CB1E	通道 CH1B 捕获数据丢失标志 仅 TIM3 存在 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO
11	CB0E	通道 CH0B 捕获数据丢失标志 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO

位/位域	标记	功能描述	读写
10	CA2E	通道 CH2A 捕获数据丢失标志 仅 TIM3 存在 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO
9	CA1E	通道 CH1A 捕获数据丢失标志 仅 TIM3 存在 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO
8	CA0E	通道 CH0A 捕获数据丢失标志 ● 0b0: 无数据丢失 ● 0b1: 数据丢失	RO
7	CB2F	通道 CH2B 发生捕获/比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
6	CB1F	通道 CH1B 发生捕获/比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
5	CB0F	通道 CH0B 发生捕获/比较匹配标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
4	CA2F	通道 CH2A 发生捕获/比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
3	CA1F	通道 CH1A 发生捕获/比较匹配标志 仅 TIM3 存在 ● 0b0: 没有发生 ● 0b1: 发生	RO
2	CA0F	通道 CH0A 发生捕获/比较匹配标志 ● 0b0: 没有发生 ● 0b1: 发生	RO
1	<i>Reserved</i>	-	-
0	UIF	事件更新中断标志 ● 0b0: 没有发生 ● 0b1: 发生	RO

13.5.4.5 中断标志清除寄存器 (TIM_ICLR)

Offset	Bit Position																																						
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Reset	0x003FFFFFF(TIM0/1/2);0x000FFFFFF(TIM3)																																						
Name	Reserved												CB3F	CA3F	UNDF	OVF	TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Reserved	UIF							
Access													WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO

位/位域	标记	功能描述	读写
31:20	Reserved	-	-
19	CB3F	通道 CH3B 比较匹配标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
18	CA3F	通道 CH3A 比较匹配标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
17	UNDF	下溢出中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
16	OVF	上溢出中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
15	TIF	触发中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
14	BIF	刹车中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
13	CB2E	通道 CH2B 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
12	CB1E	通道 CH1B 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
11	CB0E	通道 CH0B 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
10	CA2E	通道 CH2A 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
9	CA1E	通道 CH1A 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO
8	CA0E	通道 CH0A 捕获数据丢失标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
7	CB2F	通道 CH2B 捕获/比较匹配标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1 仅 TIM3 存在	WO

位/位域	标记	功能描述	读写
6	CB1F	通道 CH1B 捕获/比较匹配标志清除 写 0 清除标志，写 1 无效；读出恒为 1 仅 TIM3 存在	WO
5	CB0F	通道 CH0B 捕获/比较匹配标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
4	CA2F	通道 CH2A 捕获/比较匹配标志清除 写 0 清除标志，写 1 无效；读出恒为 1 仅 TIM3 存在	WO
3	CA1F	通道 CH1A 捕获/比较匹配标志清除 写 0 清除标志，写 1 无效；读出恒为 1 仅 TIM3 存在	WO
2	CA0F	通道 CH0A 捕获/比较匹配标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
1	<i>Reserved</i>	-	-
0	UIF	事件更新中断标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO

13.5.4.6 主从模式控制寄存器 (TIM_MSCR)

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				SMS		TS						MMS			
																	IB0S		IA0S						MSM		CCDS					
Access																	RW		RW		RW		RW		RW		RW					

位/位域	标记	功能描述	读写
31:13	Reserved	-	-
12	IB0S	CH0B 输入选择 0b0: CH0B 0b1: 内部触发 TS 选择信号	RW
11	IA0S	IA0 输入选择 0b0: CH0A 0b1: - CH0A CH1A CH2A XOR (TIM3) - CH0A ETR GATE XOR (TIM0/1/2)  说明 设置为 1 后端口的任意一个端口变化将导致输入变化。	RW
10:8	SMS	从模式功能选择 0b000: 使用内部时钟 0b001: 复位功能 0b010: 触发模式 0b011: 外部时钟模式 0b100: 正交编码计数模式 1 0b101: 正交编码计数模式 2 0b110: 正交编码计数模式 3 0b111: 门控功能	RW
7:5	TS	触发选择 0b000: 端口 ETR 的滤波相位选择后的信号 ETFP 0b001: 内部互联信号 ITR0 0b010: 内部互联信号 ITR1 0b011: 内部互联信号 ITR2 0b100: 内部互联信号 ITR3 0b101: 端口 CH0A 的边沿信号 0b110: 端口 CH0A 的滤波相位选择后的信号 IAFP 0b111: 端口 CH0B 的滤波相位选择后的信号 IBFP	RW

位/位域	标记	功能描述	读写																											
4	MSM	主从选择 ● 0b0: 无延时 ● 0b1: 延时使能，使主从计数器同时启动 说明 使用触发模式时，从模式设置为 0，主模式设置为 1，可以使主从计数同时启动。	RW																											
3	CCDS	比较模式下 DMA 比较触发选择 ● 0b0: 比较匹配触发 DMA ● 0b1: 比较匹配不触发 DMA，事件更新代替比较匹配触发 DMA	RW																											
2:0	MMS	主模式输出（TIMx_TRGO）选择 用于内部互联，连接到其他定时器的 ITRx <table><tr><th></th><th>TIM0/1/2</th><th>TIM3</th></tr><tr><td>0b000</td><td>软件更新 UG，写 M23CR.UG</td><td>软件更新 UG，写 M23CR.UG 软件更新 TG，写 M23CR.TG</td></tr><tr><td>0b001</td><td>定时器使能 CTEN</td><td>定时器使能 CTEN</td></tr><tr><td>0b010</td><td>定时器事件更新 UEV</td><td>定时器事件更新 UEV</td></tr><tr><td>0b011</td><td>比较匹配选择输出 CMP5O</td><td>比较匹配选择输出 CMP5O</td></tr><tr><td>0b100</td><td>定时器比较参数输出 OCREFOA</td><td>定时器比较参数输出 OCREFOA</td></tr><tr><td>0b101</td><td>定时器比较参数输出 OCREFOB</td><td>定时器比较参数输出 OCREF1A</td></tr><tr><td>0b110</td><td>定时器比较参数输出 OCREFOB</td><td>定时器比较参数输出 OCREF2A</td></tr><tr><td>0b111</td><td>定时器比较参数输出 OCREFOB</td><td>定时器比较参数输出 OCREFOB</td></tr></table> 说明 配置 MMS 为 0b000/0b010/0b011 时，输出为脉冲信号。		TIM0/1/2	TIM3	0b000	软件更新 UG，写 M23CR.UG	软件更新 UG，写 M23CR.UG 软件更新 TG，写 M23CR.TG	0b001	定时器使能 CTEN	定时器使能 CTEN	0b010	定时器事件更新 UEV	定时器事件更新 UEV	0b011	比较匹配选择输出 CMP5O	比较匹配选择输出 CMP5O	0b100	定时器比较参数输出 OCREFOA	定时器比较参数输出 OCREFOA	0b101	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREF1A	0b110	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREF2A	0b111	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREFOB	RW
	TIM0/1/2	TIM3																												
0b000	软件更新 UG，写 M23CR.UG	软件更新 UG，写 M23CR.UG 软件更新 TG，写 M23CR.TG																												
0b001	定时器使能 CTEN	定时器使能 CTEN																												
0b010	定时器事件更新 UEV	定时器事件更新 UEV																												
0b011	比较匹配选择输出 CMP5O	比较匹配选择输出 CMP5O																												
0b100	定时器比较参数输出 OCREFOA	定时器比较参数输出 OCREFOA																												
0b101	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREF1A																												
0b110	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREF2A																												
0b111	定时器比较参数输出 OCREFOB	定时器比较参数输出 OCREFOB																												

13.5.4.7 输出控制/输入滤波寄存器 (TIM_FLTR)

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	ETP	FLTET		BKP		FLTBK		CCPB2		OCMB2/FLTB2		CCPA2		OCMA2/FLTA2		CCPB1		OCMB1/FLTB1		CCPA1		OCMA1/FLTA1		CCPB0/CCPB0		OCMB0/FLTB0		CCPA0/CCPA0		OCMA0/FLTA0		
Access	RW	RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		

位/位域	标记	功能描述	读写
31	ETP	ETR 输入相位选择 0b0: 同相位 0b1: 反向输入	RW
30:28	FLTET	ETR 滤波控制 采样时钟为 PCLK 或 PCLK 分频, 采样点个数为 N 0b000: 无滤波 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
27	BKP	刹车 BK 输入相位选择 0b0: 同相位 0b1: 反向输入	RW
26:24	FLTBK	刹车输入滤波控制 采样时钟为 PCLK 或 PCLK 分频, 采样点个数为 N 0b000: 无滤波 0b001: $F_{\text{sample}} = \text{PCLK}$, $N=2$ 0b010: $F_{\text{sample}} = \text{PCLK}$, $N=4$ 0b011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 0b100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 0b101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 0b110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 0b111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
23	CCPB2	比较功能: CH2B 通道比较输出相位控制 0b0: 正常输出 0b1: 反向输出	RW

位/位域	标记	功能描述	读写
22:20	OCMB2	比较功能：CH2B 通道比较控制 参考 OCMB0 设置	RW
	FLTB2	捕获功能：CH2B 输入通道滤波设置 参考 FLTBK 设置	RW
19	CCPA2	比较功能：CH2A 通道比较输出相位控制 ● 0b0：正常输出 ● 0b1：反向输出	RW
18:16	OCMA2	比较功能：CH2A 通道比较控制 参考 OCMB0 设置	RW
	FLTA2	捕获功能：CH2A 输入通道滤波设置 参考 FLTBK 设置	RW
15	CCPB1	比较功能：CH1B 通道比较输出相位控制 ● 0b0：正常输出 ● 0b1：反向输出	RW
14:12	OCMB1	比较功能：CH1B 通道比较控制 参考 OCMB0 设置	RW
	FLTB1	捕获功能：CH1B 输入通道滤波设置 参考 FLTBK 设置	RW
11	CCPA1	比较功能：CH1A 通道比较输出相位控制 ● 0b0：正常输出 ● 0b1：反向输出	RW
10:8	OCMA1	比较功能：CH1A 通道比较控制 参考 OCMB0 设置	RW
	FLTA1	捕获功能：CH1A 输入通道滤波设置 参考 FLTBK 设置	RW
7	CCPB0	比较功能：输出比较模式 CCPBx 比较输出 CHBx 端口极性控制 ● 0b0：正常输出 ● 0b1：反向输出	RW
	CCPB0	编码计数与从模式门控功能：输入相位控制 CCPB0 从模式门控，复位，外部触发，外部时钟使用 CH0B 端口输入极性控制 ● 0b0：正常输入 ● 0b1：反向输入	RW
6:4	OCMB0	比较功能：CH0B 通道比较控制 ● 0b000：强制为 0 ● 0b001：强制为 1 ● 0b010：比较匹配时强制为 0 ● 0b011：比较匹配时强制为 1 ● 0b100：比较匹配时翻转 ● 0b101：比较匹配时输出一个计数周期的高电平 ● 0b110：PWM 模式 1 ▶ 单点比较 上计数时 CNT<CCRxy 输出为高电平，下计数时 CNT>CCRxy 输出为低电平	RW

位/位域	标记	功能描述	读写
6:4	OCMB0	▶ 双点比较 ◆ 锯齿波上计数 $CCRxA < CNT \leq CCRxB$ 输出为低电平 ◆ 锯齿波下计数 $CCRxA < CNT \leq CCRxB$ 输出为高电平 ◆ 三角波上计数 $CNT < CCRxA$ 输出为高电平，下计数 $CNT > CCRxB$ 输出为低电平 ● 0b111: PWM 模式 2 ▶ 单点比较 - 上计数时 $CNT < CCRxy$ 输出为低电平，下计数时 $CNT > CCRxy$ 输出为高电平 ▶ 双点比较 ◆ 锯齿波上计数 $CCRxA \leq CNT < CCRxB$ 输出为高电平 ◆ 锯齿波下计数 $CCRxA \leq CNT < CCRxB$ 输出为低电平 ◆ 三角波上计数 $CNT < CCRxA$ 输出为低电平，下计数 $CNT > CCRxB$ 输出为高电平	
	FLTBO	捕获功能: CH0B 输入通道滤波设置 参考 FLTBK 设置  说明 为了保证 OCMB0 的 PWM 输出设置, GATE 在 PWM 互补模式下作为捕获输入, 滤波设置无效。	RW
3	CCPA0	比较功能: 输出比较模式 CCPAx 比较输出 CHAx 端口极性控制 ● 0b0: 正常输出 ● 0b1: 反向输出	RW
	CCPA0	编码计数与从模式门控功能: 输入相位控制 CCPA0 从模式门控, 复位, 外部触发, 外部时钟使用 CH0A 端口输入极性控制 ● 0b0: 正常输入 ● 0b1: 反向输入	RW
2:0	OCMA0	比较功能: A 通道比较控制 参考 OCMB0 设置	RW
	FLTA0	捕获功能: CH0A 输入通道滤波设置 参考 FLTBK 设置	RW

13.5.4.8 ADC 触发控制寄存器 (TIM_ADTR)

Offset	Bit Position																																									
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Reset	0x00000000																																									
Name	Reserved																											ADTE		CMB2E		CMB1E		CMB0E		CMA2E		CMA1E		CMA0E		UEVE
Access																												RW		RW		RW		RW		RW		RW		RW		RW

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7	ADTE	使能 ADC 触发全局控制 ● 0b0: 禁止 ● 0b1: 使能	RW
6	CMB2E	通道 2B 比较匹配触发 ADC 使能 仅 TIM3 存在 ● 0b0: 禁止 ● 0b1: 使能	RW
5	CMB1E	通道 1B 比较匹配触发 ADC 使能 仅 TIM3 存在 ● 0b0: 禁止 ● 0b1: 使能	RW
4	CMB0E	通道 0B 比较匹配触发 ADC 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
3	CMA2E	通道 2A 比较匹配触发 ADC 使能 仅 TIM3 存在 ● 0b0: 禁止 ● 0b1: 使能	RW
2	CMA1E	通道 1A 比较匹配触发 ADC 使能 仅 TIM3 存在 ● 0b0: 禁止 ● 0b1: 使能	RW
1	CMA0E	通道 0A 比较匹配触发 ADC 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
0	UEVE	事件更新触发 ADC 使能 ● 0b0: 禁止 ● 0b1: 使能	RW

13.5.4.9 通道 x 控制寄存器 (TIM_CRCHx) (x=0/1/2)

偏移地址 (Offset) : 0x024 (CRCH0) , 0x028 (CRCH1) , 0x02C (CRCH2)



说明

CRCH1/CRCH2 仅 TIM3 存在。

Offset	Bit Position																																																
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
Reset	0x00003000																																																
Name	Reserved																																																
Access																	WO	WO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW									

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	CCGB	捕获比较 B 软件触发 硬件自动清零, 写 1 产生下述动作, 写 0 无效 - 在比较模式下只产生中断 - 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中	WO
14	CCGA	捕获比较 A 软件触发 硬件自动清零, 写 1 产生下述动作, 写 0 无效 - 在比较模式下只产生中断 - 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中	WO
13:12	CISB	B 通道比较匹配设置 0b00: 无匹配 0b01: 上计数时比较匹配 0b10: 下计数时比较匹配 0b11: 上/下计数时比较都匹配	RW
11	CDEB	B 捕获比较触发 DMA 使能 0b0: 禁止 0b1: 使能	RW
10	CDEA	A 捕获比较触发 DMA 使能 0b0: 禁止 0b1: 使能	RW
9	CIEB	B 捕获比较触发中断使能 0b0: 禁止 0b1: 使能	RW
8	CIEA	A 捕获比较触发中断使能 0b0: 禁止 0b1: 使能	RW
7	BUFEB	比较功能: B 比较缓存使能控制 0b0: 禁止 0b1: 使能	RW

位/位域	标记	功能描述	读写
6	BUFEA	比较功能：A 比较缓存使能控制 ● 0b0：禁止 ● 0b1：使能	RW
5	CSB	B 通道捕获/比较功能选择 ● 0b0：比较模式 ● 0b1：捕获模式	RW
4	CSA	A 通道捕获/比较功能选择 ● 0b0：比较模式 ● 0b1：捕获模式	RW
3:2	CFB_CRB	捕获功能：B 通道上升下降沿捕获使能 ● 0b00：禁止 ● 0b01：上升沿捕获 ● 0b10：下降沿捕获 ● 0b11：上升沿下降沿均捕获	RW
	BKSB	比较功能：B 通道输出刹车电平控制 ● 0b00：高阻输出 ● 0b01：对输出无影响 ● 0b10：强制输出低电平 ● 0b11：强制输出高电平  说明 无 DEBUG_STOP 模式。	RW
1:0	CFA_CRA	捕获功能：A 通道上升下降沿捕获使能 ● 0b00：禁止 ● 0b01：上升沿捕获 ● 0b10：下降沿捕获 ● 0b11：上升沿下降沿均捕获	RW
	BKSA	比较功能：A 通道输出刹车电平控制 ● 0b00：高阻输出 ● 0b01：对输出无影响 ● 0b10：强制输出低电平 ● 0b11：强制输出高电平  说明 DEBUG_STOP 模式下，A 和 B 通道的刹车电平控制都由 BKSA 来控制。	RW

13.5.4.10 死区时间寄存器 (TIM_DTR)

Offset	Bit Position																																	
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																			VCE	SAFEEN	MOE	AOE	BKE	DTEN	BKSEL	DTR							
Access																			RW	RW	RW	RW	RW	RW	RW	RW								

位/位域	标记	功能描述	读写																				
31:15	Reserved	-	-																				
14	VCE	VC 刹车使能 ● 0b0: 禁止 ● 0b1: 使能	RW																				
13	SAFEEN	Safety 刹车使能 (XTH fail, lockup) ● 0b0: 禁止 ● 0b1: 使能	RW																				
12	MOE	PWM 输出使能 ● 0b0: 禁止 ● 0b1: 使能	RW																				
11	AOE	PWM 输出自动使能 ● 0b0: 禁止 ● 0b1: 使能	RW																				
10	BKE	端口刹车使能 ● 0b0: 禁止 ● 0b1: 使能	RW																				
9	DTEN	死区控制使能 ● 0b0: 禁止 ● 0b1: 使能	RW																				
8	BKSEL	刹车选择 ● 0b0: 使用自己的刹车控制 ● 0b1: TIM1/2 使用 TIM0 的刹车控制  说明 TIM0 和 TIM3 该位选择无效。	RW																				
7:0	DTR	死区时间 <table border="1"> <thead> <tr> <th>DTR</th><th>T</th><th>dt</th><th>step</th></tr> </thead> <tbody> <tr> <td>DTR[7]=0b0</td><td>DTR[6:0]+2</td><td>2~129</td><td>1</td></tr> <tr> <td>DTR[7:6]=0b10</td><td>{DTR[5:0]+64}*2+2</td><td>130~256</td><td>2</td></tr> <tr> <td>DTR[7:5]=0b110</td><td>{DTR[4:0]+32}*8+2</td><td>258~506</td><td>8</td></tr> <tr> <td>DTR[7:5]=0b111</td><td>{DTR[4:0]+32}*16+2</td><td>514~1010</td><td>16</td></tr> </tbody> </table>	DTR	T	dt	step	DTR[7]=0b0	DTR[6:0]+2	2~129	1	DTR[7:6]=0b10	{DTR[5:0]+64}*2+2	130~256	2	DTR[7:5]=0b110	{DTR[4:0]+32}*8+2	258~506	8	DTR[7:5]=0b111	{DTR[4:0]+32}*16+2	514~1010	16	RW
DTR	T	dt	step																				
DTR[7]=0b0	DTR[6:0]+2	2~129	1																				
DTR[7:6]=0b10	{DTR[5:0]+64}*2+2	130~256	2																				
DTR[7:5]=0b110	{DTR[4:0]+32}*8+2	258~506	8																				
DTR[7:5]=0b111	{DTR[4:0]+32}*16+2	514~1010	16																				

13.5.4.11 重复周期设置值寄存器 (TIM_RCR)

Offset	Bit Position																																	
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																						UND		OV		RCR							
Access																							RW	RW										

位/位域	标记	功能描述	读写
31:10	Reserved	-	-
9	UND	重复计数控制的下溢出屏蔽 ● 0b0: 下计数溢出使能 ● 0b1: 下计数溢出屏蔽	RW
8	OV	重复计数控制的上溢出屏蔽 ● 0b0: 上计数溢出使能 ● 0b1: 上计数溢出屏蔽	RW
7:0	RCR	重复周期计数值 设置 RCR+1 个周期, 由[9:8]控制的溢出产生事件更新, 计数器上溢出或下溢出时内部 RCR_CNT 减 1, 当计数到零后 RCR_CNT 重载 RCR 的值, 并且产生事件更新 UEV 信号	RW

13.5.4.12 重载寄存器/周期映射地址 (TIM_ARRDM)

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ARRDM															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	ARRDM	重载寄存器/周期寄存器 具有缓存功能, 在计数器未使能或者缓存没有使能时, 缓存寄存器可以立刻更新	RW

13.5.4.13 通道 x 比较捕获寄存器 (TIM_CCRxy) (x=0/1/2, y=A/B)

偏移地址 (Offset) : 0x03C (CCR0A) , 0x040 (CCR0B) , 0x044 (CCR1A) , 0x048 (CCR1B) , 0x04C (CCR2A) , 0x050 (CCR2B)

说明

CCR1A/CCR1B/CCR2A/CCR2B 仅 TIM3 存在。

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CCRxy															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CCRxy	比较捕获寄存器 比较具有缓存功能	RW

13.5.4.14 通道 3A 比较寄存器 (TIM_CCR3A) (仅 TIM3 存在)

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CCR3A															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CCR3A	比较寄存器 比较具有缓存功能	RW

13.5.4.15 通道 3A 控制寄存器 (TIM_CRCH3A) (仅 TIM3 存在)

Offset	Bit Position																																					
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																										CM3AE		C3AEN		CIS		CDE		CIE		BUF	
Access																											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	CM3AE	ADC 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
5	C3AEN	通道比较使能 ● 0b0: 禁止 ● 0b1: 使能	RW
4:3	CIS	通道比较匹配设置 ● 0b00: 无匹配 ● 0b01: 上计数时比较匹配 ● 0b10: 下计数时比较匹配 ● 0b11: 上/下计数时比较都匹配	RW
2	CDE	比较触发 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
1	CIE	比较触发中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
0	BUF	比较缓存使能 ● 0b0: 禁止 ● 0b1: 使能	RW

13.5.4.16 通道 3B 比较寄存器 (TIM_CCR3B) (仅 TIM3 存在)

Offset	Bit Position																															
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																CCR3B															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	CCR3B	比较寄存器 比较具有缓存功能	RW

13.5.4.17 通道 3B 控制寄存器 (TIM_CRCH3B) (仅 TIM3 存在)

Offset	Bit Position																																					
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																										CM3BE		C3BEN		CIS		CDE		CIE		BUF	
Access																											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	CM3BE	ADC 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
5	C3BEN	通道比较使能 ● 0b0: 禁止 ● 0b1: 使能	RW
4:3	CIS	通道比较匹配设置 ● 0b00: 无匹配 ● 0b01: 上计数时比较匹配 ● 0b10: 下计数时比较匹配 ● 0b11: 上/下计数时比较都匹配	RW
2	CDE	比较触发 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
1	CIE	比较触发中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
0	BUF	比较缓存使能 ● 0b0: 禁止 ● 0b1: 使能	RW

14 实时时钟（RTC）

14.1 简介

实时时钟/日历提供秒、分、时、日、周、月、年的信息，每月的天数和闰年的天数可自动调整。时钟操作可通过 RTC_CR0.AMPM 寄存器位，决定采用 24 或 12 小时格式。具有最小周期为 0.5~32s 的周期中断，60μs~145Hour 唤醒定时器中断。

14.2 主要特性

表 14-1 RTC 的基本特性

基本特性	说明
时钟源	片外低速晶振 XTL（32.768kHz） 片内低速振荡器 RCL（32kHz） 片外高速晶振 XTH
基本功能	可计算 00~99 年之间的秒、分、时、日、周、月、年
	可自动进行闰年调整
	可配置为 24 或 12 小时格式
	可程序控制启动或停止
	具有两组闹钟功能
	具有高精度 1Hz 方波信号输出
	可配置的唤醒定时器
	时间戳功能
中断	具有 0.5~32s 可配置的周期中断
	具有精确到秒的闹钟中断 A/B
	唤醒定时器中断
	时间戳中断

14.3 功能说明

实时时钟的时钟源可配置为外部低速晶振，外部高速晶振，内部低速时钟 RCL；默认使用外部低速晶振。RTC_SEC、RTC_MIN、RTC_HOUR、RTC_DAY、RTC_WEEK、RTC_MON、RTC_YEAR、RTC_ALMA、RTC_ALMB 寄存器只受上电复位影响，上电后状态不定，需要初始化。其他寄存器只受上电复位与模块复位控制，其他复位源不能复位这些寄存器。

所有软件写入和读取的日期时间值都为 BCD 码，无须十六进制转换为十进制。

任何无效的日期时间将无法写入，比如 32 日、25 时、70 秒、B 月等。

14.3.1 功能框图

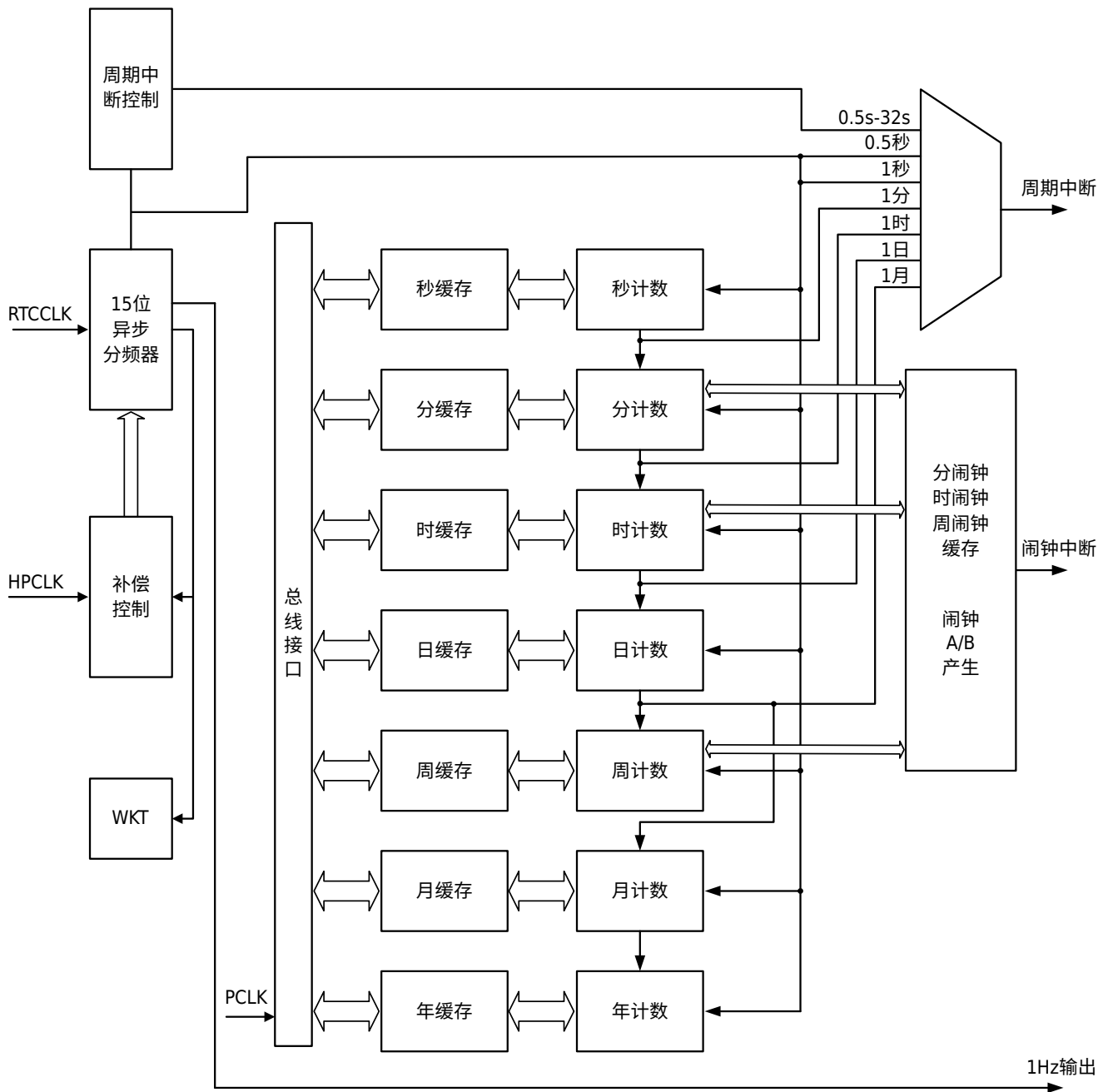


图 14-1 RTC 框图

14.3.2 上电设定

RTC 在上电之后复位一次，在系统不掉电的情况下，外部各种复位请求都不能复位 RTC，RTC 会一直处于计数状态。在上电之后，设定日历初始值、闹钟设置、误差补偿、中断等之后，启动 RTC。

14.3.3 时钟误差补偿

由于外部晶振存在误差，在需要得到高精度的计数结果时，需要对该误差进行补偿，可配置两种补偿精度（0.96ppm 或 0.06ppm）。补偿方法分为两种：第一种，基于自身时钟 RTCCLK 的误差补偿；第二种，基于高速时钟 HPCLK 的误差补偿。

基于自身时钟的误差补偿原理与计算（0.96ppm）

由于计数器采用 32.768kHz 的时钟计数，如果需要对每秒精度进行补偿时，只能按照 32.768kHz 的整数周期补偿，则每秒补偿的最小单位为 $(1/32768) \times 10^6 = 30.5\text{ppm}$ ，无法满足高精度的要求。

那么要在 32.768kHz 的计数时钟下实现精度较高的时钟补偿时，需要在算法上做调整，将最大补偿周期扩大 32 倍。即在只能补偿的最小单位为 30.5ppm 的情况下，平均到每秒的补偿单位变为 $30.5\text{ppm}/32 = 0.96\text{ppm}$ 。为了满足精度较高的时钟补偿要求，且补偿发生在每 32 秒内比较均匀的范围内，该寄存器中引入了 5 位小数的设定。

举个例子：当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 1.0000203Hz，则：

实际发振频率 = $32768 \times 1.0000203 \approx 32768.6651904$ （实际发振频率需要至少精确到小数点后三位）

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32768.6651904 - 32768) / 32768 \times 10^6 \\ &= 20.3\text{ppm}\end{aligned}$$

根据公式：

$$\text{CR}[8:0] = \left(\frac{\text{补偿目标值} [\text{ppm}] \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000\text{B}$$

如果补偿目标值为 +20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{CR}[8:0] &= (20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= (0.6651904)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= 0000.10101\text{B} + 0001.00000\text{B} \\ &= 0001.10101\text{B}\end{aligned}$$

如果补偿目标值为 -20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}\text{CR}[8:0] &= (-20.3 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= (-0.6651904)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B}\end{aligned}$$

基于自身时钟的误差补偿原理与计算（0.06ppm）

实现 0.06ppm 补偿精度需要将最大补偿周期扩大 512 倍。则在只能补偿的最小单位为 30.5ppm 的情况下，平均到每秒的补偿单位变为 $30.5\text{ppm}/512 = 0.06\text{ppm}$ 。满足了精度较高的时钟补偿要求。而且补偿发生在每 512 秒内比较均匀的范围内。所以，该寄存器中引入了 9 位小数的设定。

举个例子：当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 0.9999888Hz，则：

实际发振频率 = $32768 \times 0.9999888 \approx 32767.6329984$ （实际发振频率需要至少精确到小数点后五位）

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32767.6329984 - 32768) / 32768 \times 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

根据公式：

$$CR[12:0] = \left(\frac{\text{补偿目标值 [ppm]} \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.000000000B$$

如果补偿目标值为-11.29ppm，计算相应的寄存器值如下：

$$\begin{aligned} CR[12:0] &= (-11.29 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.000000000B \\ &= (-0.37)_{\text{取2的补码}} + 0001.000000000B \\ &= 1111.101000011B + 0001.000000000B \\ &= 0000.101000011B \end{aligned}$$

基于高速时钟的误差补偿原理与计算

该方式的计算方法与基于自身时钟的误差补偿相同。由于引入了高速时钟 HPCLK (XTH 外部高速时钟)，本来需要在最多 32 秒内累计的 1/32768 秒误差可分散到每 1 秒，针对每 1 秒进行的补偿，实现平均的每秒高精度 1Hz 时钟输出。

需要根据频率设定正确的高速补偿参数值 RTC_HCOMP.PARA。

$$RTC_HCOMP.PARA = f/32768/32$$

f 为补偿频率，建议配置为 32.768kHz 的整数倍：

- 0.06ppm 补偿精度：f 取值范围是 16~48MHz。
- 0.96ppm 补偿精度：f 取值范围是 4~48MHz。

14.3.4 时间戳功能

将 RTC_CR2.TSE 位置 1 可使能时间戳。

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器 (RTC_TSTR 和 RTC_TSDR) 中。

发生时间戳事件时，RTC_IFR.TS (时间戳标志位) 将置 1。

通过将 RTC_IER.TS 位置 1，可在发生时间戳事件时生成中断。

如果在时间戳标志 (TS) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOV) 将置 1，而时间戳寄存器 (RTC_TSTR 和 RTC_TSDR) 将保持上一事件的结果。

14.3.5 唤醒定时器

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。可通过 RTC_CR2.WUTE 位来使能此唤醒功能。

唤醒定时器的时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。
- 1、2、4 或 8 分频的秒脉冲。

完成初始化后，定时器开始递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_IFR.WU 标志会置 1，并且唤醒寄存器会使用 RTC_WUTR 寄存器的值自动重载。

之后必须用软件清零 WU 标志。

暂停唤醒定时器不会产生重载动作，需要重新定时需要对 RTC_WUTR 寄存器产生一个写动作。

通过将 RTC_IER.WU 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

系统复位以及低功耗模式 (休眠、深度休眠) 对唤醒定时器没有任何影响。

在 RTC 计数开始后，如果系统要进入低功耗模式，需要确保已经将寄存器的信息同步到 RTC，请执行下列任意一种确认后再进行模式切换。

- 在配置完 RTC 寄存器后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
- 在配置完 RTC 寄存器后，设定 RTC_CR1.WAIT=1，查询 RTC_CR1.WAITF=1。再设定 RTC_CR1.WAIT=0，查询 RTC_CR1.WAITF=0 后再进行模式切换。

14.3.6 RTC 中断

RTC 支持 4 种中断类型。闹钟 A/B 中断、周期中断、时间戳中断与唤醒定时器中断。

闹钟 A/B 中断

当 RTC_IER.ALMx=1 时，若当前日历时间与闹钟寄存器相等时，触发闹钟中断。通过设置 MSK 相关位可以屏蔽相应闹钟时间位的比较。

周期中断

当 RTC_IER.PRD=1 时，选择的周期发生后，触发周期中断，由于共用中断，通过标志寄存器位来区分。

唤醒定时器中断

当 RTC_IER.WU=1 时，唤醒定时器计数器计数到零，触发中断，由于共用中断，通过标志寄存器位来区分。

时间戳中断

当 RTC_IER.TS=1 时，发生时间戳触发时，产生中断，由于共用中断，通过标志寄存器位来区分。

14.4 操作示例

14.4.1 RTC 计数开始设置

操作步骤

- 步骤 1. 设置 RTC_CR0.START=0，计数停止。
- 步骤 2. 设置 RTC_CR0.AMPM 和 RTC_CR0.PRDS，RTC_CR0.PRDX 配置时制和中断周期。
- 步骤 3. 设置 RTC_CR1.CKSEL 选择 RTC 的计时时钟。
- 步骤 4. 设置秒、分、时、周、日、月、年的日历计数寄存器。
- 步骤 5. 需要进行时钟误差补偿时，设置计数时钟误差补偿寄存器 RTC_COMPEN。
- 步骤 6. 清除中断标志位寄存器 RTC_IFR，并使能中断。
- 步骤 7. 设置 RTC_CR0.START=1，计数开始。

14.4.2 读出计数寄存器

背景信息

有三种读取计数寄存器的方法。

操作步骤

● 任意时刻读取方式 1

- 步骤 1. 设置 RTC_CR1.WAIT=1，停止日历寄存器计数，进入读写模式。
- 步骤 2. 查询直到 RTC_CR1.WAITF=1。
- 步骤 3. 读出秒、分、时、周、日、月、年计数寄存器值。
- 步骤 4. 设置 RTC_CR1.WAIT=0，计数器计数。
- 步骤 5. 查询直到 RTC_CR1.WAITF=0。

● 任意时刻读取方式 2

- 步骤 1. 读出分、时、周、日、月、年计数寄存器值。
- 步骤 2. 读出秒计数寄存器值。
- 步骤 3. 再次读出秒计数寄存器值。
- 步骤 4. 判断两次秒的读出值是否相同，不同重新从第一步开始，相同读取结束。

● 中断读取方式

在 RTC 周期中断服务中读取秒、分、时、周、日、月、年计数寄存器值。因为中断发生后到下次数据改变至少 0.5s 的时间。

14.4.3 写入计数寄存器

背景信息

在计数模式下更改秒寄存器会复位秒计数，写分、时、周、日、月、年计数寄存器值不会影响 RTC 计数。



说明

RTC 寄存器带有写保护，参考 RTC_WPR 寄存器解除写保护后才可以写入。

在 RTC 未启动模式下写秒、分、时、周、日、月、年计数寄存器不需要等 WAIT。

操作步骤

- 步骤 1. 设定 RTC_CR1.WAIT=1，停止日历寄存器计数，进入读写模式。
- 步骤 2. 查询直到 RTC_CR1.WAITF=1。
- 步骤 3. 写入秒、分、时、周、日、月、年计数寄存器值。
- 步骤 4. 设定 RTC_CR1.WAIT=0，计数器重新开始计数。



注意

必须在 1 秒内完成所有写操作。

- 步骤 5. 查询直到 RTC_CR1.WAITF=0。

14.4.4 闹钟设定

操作步骤

- 步骤 1. 设定 RTC_CR2.ALME=0，闹钟禁止。
- 步骤 2. 设定 RTC_IER.ALME=1，闹钟中断许可。
- 步骤 3. 闹钟 RTC_ALMx 设定，根据需要配置闹钟时间。
- 步骤 4. 设定 RTC_CR2.ALME=1，闹钟许可。
- 步骤 5. 等待发生中断。
- 步骤 6. 由于闹钟中断和定周期中断共用中断请求信号，则当 RTC_IFR.ALME=1 时，进入闹钟中断处理；否则进入定周期中断处理。

14.4.5 1Hz 输出

背景信息

RTC 可选择输出一般精度，较高精度和高精度 3 种 1Hz 时钟。当时钟误差补偿功能有效时输出较高精度的 1Hz 时钟；当使用不同频率的 HPCLK 时输出高精度的 1Hz 时钟。需要根据 HPCLK 频率配置系统控制寄存器，三种 1Hz 输出方法如下：

操作步骤

- 一般精度的 1Hz 输出（无时钟补偿）
 - 步骤 1. 设定 RTC_CR0.START=0，计数停止。
 - 步骤 2. RTC 输出引脚设定。
 - 步骤 3. 设定 RTC_CR0.HZ1OE=1，时钟输出许可。
 - 步骤 4. 设定 RTC_CR0.START=1，计数开始。
 - 步骤 5. 等待 2 个计数周期以上。
 - 步骤 6. 1Hz 输出开始。
- 较高精度的 1Hz 输出（低速补偿）
 - 步骤 1. 设定 RTC_CR0.START=0，计数停止。
 - 步骤 2. RTC 输出引脚设定。
 - 步骤 3. 设定 RTC_CR0.HZ1OE=1，时钟输出许可。

- 步骤 4. 配置 RTC_COMPEN.CR, 设定补偿数。
- 步骤 5. 设定 RTC_COMPEN.EN=1, 误差补偿有效。
- 步骤 6. 设定 RTC_CR0.START=1, 计数开始。
- 步骤 7. 等待 2 个计数周期以上。
- 步骤 8. 1Hz 输出开始。
- 高精度的 1Hz 输出 (在较高精度输出的基础上, 为 RTC 提供高速补偿时钟)
 - 步骤 1. 设定 RTC_CR0.START=0, 计数停止。
 - 步骤 2. RTC 输出引脚设定。
 - 步骤 3. 设定 RTC_CR0.HZ1OE=1, 时钟输出许可。
 - 步骤 4. 设定 RTC_CR0.HZ1SEL=1, 选择输出高精度 1Hz 时钟。
 - 步骤 5. 根据配置的频率设定高速补偿参数 RTC_HCOMP.PARA。
 - 步骤 6. 配置 RTC_COMPEN.CR[12:0], 设定补偿数。
 - 步骤 7. 设定 RTC_COMPEN.EN=1, 误差补偿有效。
 - 步骤 8. 设定 RTC_COMPEN.CSTEPS=0, 0.96ppm 精度补偿有效; 或设定 RTC_COMPEN.CSTEPS=1, 0.06ppm 精度补偿有效。
 - 步骤 9. 设定 RTC_CR0.START=1, 计数开始。
 - 步骤 10. 等待 2 个计数周期以上。
 - 步骤 11. 1Hz 输出开始。

14.5 寄存器

14.5.1 寄存器总表

基地址：0x40001400

表 14-2 实时时钟 (RTC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	RTC_CR0	控制寄存器 0	32
0x004	RTC_CR1	控制寄存器 1	32
0x008	RTC_SEC	秒计数寄存器	32
0x00C	RTC_MIN	分计数寄存器	32
0x010	RTC_HOUR	时计数寄存器	32
0x014	RTC_WEEK	周计数寄存器	32
0x018	RTC_DAY	日计数寄存器	32
0x01C	RTC_MON	月计数寄存器	32
0x020	RTC_YEAR	年计数寄存器	32
0x030	RTC_COMPEN	时钟误差补偿寄存器	32
0x034	RTC_HCOMP	高速补偿精度参数寄存器	32
0x03C	RTC_CR2	控制寄存器 2	32
0x040	RTC_ALMA	闹钟 A 寄存器	32
0x044	RTC_ALMB	闹钟 B 寄存器	32
0x048	RTC_TSTR	时间戳时间寄存器	32
0x04C	RTC_TSDR	时间戳日期寄存器	32
0x050	RTC_WUTR	唤醒定时器寄存器	32
0x054	RTC_IER	中断使能寄存器	32
0x058	RTC_IFR	中断标志寄存器	32
0x05C	RTC_ICR	中断清除寄存器	32
0x060	RTC_WPR	保护寄存器	32

14.5.2 控制寄存器 0 (RTC_CR0)

Offset	Bit Position																																
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																																
Name	Reserved																	PRDSEL	PRDX					START	HZ1SEL	HZ1OE	Reserved	AMPM	PRDS				
Access																		RW	RW					RW	RW	RW		RW	RW				

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14	PRDSEL	周期中断源选择 0b0: 使用 PRDS 所设定的周期中断时间间隔 0b1: 使用 PRDX 所设定的周期中断时间间隔	RW
13:8	PRDX	设置产生周期中断的时间间隔 可设定的范围为 0.5 秒到 32 秒, 步进为 0.5 秒。 0b000000: 0.5 秒 0b000001: 1 秒 0b111110: 31.5 秒 0b111111: 32 秒	RW
7	START	RTC 计数使能 0b0: 停止 RTC 计数器 0b1: 使能 RTC 计数器	RW
6	HZ1SEL	1Hz 输出选择 0b0: 普通精度 1Hz 输出 0b1: 高精度 1Hz 输出	RW
5	HZ1OE	1Hz 输出使能 0b0: 禁止 1Hz 输出 0b1: 使能 1Hz 输出	RW
4	Reserved	-	-
3	AMPM	时制选择 0b0: 12 小时制 0b1: 24 小时制	RW

位/位域	标记	功能描述	读写
2:0	PRDS	设置产生中断的时间间隔 ● 0b000: 不产生周期中断 ● 0b001: 0.5 秒 ● 0b010: 1 秒 ● 0b011: 1 分钟 (每分 00 秒) ● 0b100: 1 小时 (每时 00 分 00 秒) ● 0b101: 1 天 (每日 00 时 00 分 00 秒) ● 0b110/0b111: 1 月 (每月 1 日 00 时 00 分 00 秒)  注意 如需要在 START=1 时写入更改周期中断的时间间隔，操作步骤如下： 1. 在 NVIC 中关闭 RTC 中断； 2. 更改周期中断的时间间隔； 3. 清除 RTC 中断标志； 4. 使能 RTC 中断。	RW

14.5.3 控制寄存器 1 (RTC_CR1)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																															
Name	Reserved																				CKSEL		Reserved						WAITF		WAIT	
Access																					RW								RO		RW	

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10:8	CKSEL	RTC 时钟选择 0b000/0b001: XTL 32.768k 0b010/0b011: RCL 32k 0b100: XTH/128（晶振为 4M 时选择此项） 0b101: XTH/256（晶振为 8M 时选择此项） 0b110: XTH/512（晶振为 16M 时选择此项） 0b111: XTH/1024（晶振为 32M 时选择此项）	RW
7:2	Reserved	-	-
1	WAITF	读写等待有效标志 0b0: 非写入/读出状态 0b1: 写入/读出状态  注意 WAIT 位设定是否有效标志。在写入/读出前请确认该位是否为“1”。计数过程中，在 WAIT 位清“0”后等待写入完成后该位才清“0”。	RO
0	WAIT	读写等待设定 0b0: 正常计数模式 0b1: 写入/读出模式  注意 在写入/读出时请将该位置“1”，由于计数器在连续计数，请在 1 秒的时间内完成写入/读出操作并将该位清“0”。	RW

14.5.4 秒计数寄存器 (RTC_SEC)

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								SECH				SECL			
Access																									RW				RW			

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6:4	SECH	秒计数十位值 秒十位计数值 BCD 码	RW
3:0	SECL	秒计数个位值 秒个位计数值 BCD 码	RW

表示 0~59 秒，采用十进制计数。请写入十进制 0~59 的 BCD 码，写入超出范围的值将被忽略。

14.5.5 分计数寄存器 (RTC_MIN)

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								MINH				MINL			
Access																									RW				RW			

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6:4	MINH	分计数十位值 分十位计数值 BCD 码	RW
3:0	MINL	分计数个位值 分个位计数值 BCD 码	RW

表示 0~59 分，采用十进制计数。请写入十进制 0~59 的 BCD 码，写入超出范围的值将被忽略。

14.5.6 时计数寄存器（RTC_HOUR）

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								HOURH				HOURL			
Access																									RW				RW			

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5:4	HOURH	时计数十位值 时十位计数值 BCD 码	RW
3:0	HOURL	时计数个位值 时个位计数值 BCD 码	RW

24 小时时制时，表示 0~23 小时。

12 小时时制时，表示 0~12 小时。Bit5=0 表示 AM，则 01~12 表示上午；Bit5=1 表示 PM，则 01~12 表示下午。

请根据控制位 RTC_CR0.AMPM 的值，设定正确的十进制的 BCD 码，写入超出范围的值将被忽略。

具体时间取值可参考下表：

24 小时时制（AMPM=1）		12 小时时制（AMPM=0）	
时间	寄存器值	时间	寄存器值
00 时	0x00	AM 12 时	0x12
01 时	0x01	AM 01 时	0x01
02 时	0x02	AM 02 时	0x02
03 时	0x03	AM 03 时	0x03
04 时	0x04	AM 04 时	0x04
05 时	0x05	AM 05 时	0x05
06 时	0x06	AM 06 时	0x06
07 时	0x07	AM 07 时	0x07
08 时	0x08	AM 08 时	0x08
09 时	0x09	AM 09 时	0x09
10 时	0x10	AM 10 时	0x10
11 时	0x11	AM 11 时	0x11
12 时	0x12	PM 12 时	0x32
13 时	0x13	PM 01 时	0x21
14 时	0x14	PM 02 时	0x22
15 时	0x15	PM 03 时	0x23
16 时	0x16	PM 04 时	0x24
17 时	0x17	PM 05 时	0x25

24 小时时制 (AMPM=1)		12 小时时制 (AMPM=0)	
时间	寄存器值	时间	寄存器值
18 时	0x18	PM 06 时	0x26
19 时	0x19	PM 07 时	0x27
20 时	0x20	PM 08 时	0x28
21 时	0x21	PM 09 时	0x29
22 时	0x22	PM 10 时	0x30
23 时	0x23	PM 11 时	0x31

14.5.7 日计数寄存器 (RTC_DAY)

Offset	Bit Position															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0xFFFFFFFF															
Name	Reserved														DAYH	DAYL
Access															RW	RW

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5:4	DAYH	日计数十位值 日十位计数值 BCD 码	RW
3:0	DAYL	日计数个位值 日个位计数值 BCD 码	RW

表示 1~31 日，采用十进制计数，自动计算闰年和月份。请写入十进制 1~31 的 BCD 码，写入超出范围的值将被忽略。

具体日计数取值可参考下表：

月份	日计数值
2 月（普通年）	01~28
2 月（闰年）	01~29
4、6、9、11 月	01~30
1、3、5、7、8、10、12 月	01~31

14.5.8 周计数寄存器 (RTC_WEEK)

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																															WEEK
Access																																RW

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2:0	WEEK	周计数值 周计数值 BCD 码	RW

十进制 0~6 表示周日~周六。请写入十进制 0~6 的 BCD 码，写入超出范围的值将被忽略。
具体周计数取值可参考下表：

周	周计数值
周日	0x00
周一	0x01
周二	0x02
周三	0x03
周四	0x04
周五	0x05
周六	0x06

14.5.9 月计数寄存器 (RTC_MON)

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																												MON			
Access																													RW			

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4:0	MON	月计数值 月计数值 BCD 码	RW

表示 1~12 月，采用十进制计数。请写入十进制 1~12 的 BCD 码，写入超出范围的值将被忽略。

14.5.10 年计数寄存器 (RTC_YEAR)

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																								YEARH				YEARL			
Access																									RW				RW			

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:4	YEARH	年计数十位值 年十位计数值 BCD 码	RW
3:0	YEARL	年计数个位值 年个位计数值 BCD 码	RW

表示 0~99 年，采用十进制计数，根据月进位计数。自动计算闰年如：00、04、08、...、92、96 等。请写入十进制 0~99 的 BCD 码，写入超出范围的值将被忽略。

14.5.11 时钟误差补偿寄存器 (RTC_COMPEN)

Offset	Bit Position																																		
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000020（只有上电与模块复位对该寄存器复位有效）																																		
Name	Reserved																EN	CSTEPS	Reserved	CR															
Access																	RW	RW																	

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	EN	补偿使能 ● 0b0: 禁止时钟误差补偿 ● 0b1: 使能时钟误差补偿	RW
14	CSTEPS	补偿精度 ● 0b0: 补偿精度 0.96ppm ● 0b1: 补偿精度 0.06ppm	RW
13	Reserved	-	-
12:0	CR	补偿值 CSTEPS=0b0: 通过补偿值设定, 可针对每秒进行±0.96ppm 的精度补偿。补偿值为 9 位带小数点的 2 的补码, 后 5 位为小数部分。可补偿范围 274.6ppm~212.6ppm, 最小微分误差±0.48ppm, 最小分辨率 0.96ppm。 CSTEPS=0b1: 通过补偿值设定, 可针对每秒进行±0.05960ppm 的精度补偿。补偿值为 13 位带小数点的 2 的补码, 后 9 位为小数部分。可补偿范围 274.65820ppm~213.56344ppm, 最小微分误差±0.0298ppm, 最小分辨率 0.05960ppm。	RW

0.96ppm 具体补偿精度请参考下表:

补偿值设定										补偿值
EN	CR[8:0]									
1	1	0	0	0	0	0	0	0	0	-274.6ppm
	1	0	0	0	0	0	0	0	1	-273.7ppm
	...									...
	0	0	0	0	1	1	1	1	1	-0.96ppm
	0	0	0	1	0	0	0	0	0	0ppm
	...									...
	0	1	1	1	1	1	1	1	0	+211.7ppm
	0	1	1	1	1	1	1	1	1	+212.6ppm
0	X	X	X	X	X	X	X	X	X	无补偿

0.06ppm 具体补偿精度请参考下表：

补偿值设定														补偿值
EN&CSTEPS	CR[12:0]													
1	1	0	0	0	0	0	0	0	0	0	0	0	0	-274.65820ppm
	1	0	0	0	0	0	0	0	0	0	0	0	1	-274.59859ppm
	...													...
	0	0	0	0	1	1	1	1	1	1	1	1	1	-0.059605ppm
	0	0	0	1	0	0	0	0	0	0	0	0	0	0ppm
	...													...
	0	1	1	1	1	1	1	1	1	1	1	1	0	+213.50383ppm
	0	1	1	1	1	1	1	1	1	1	1	1	1	+213.56344ppm
0	X	X	X	X	X	X	X	X	X	X	X	X	X	无补偿

14.5.12 高速补偿精度参数寄存器（RTC_HCOMP）

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																															
Name	Reserved																								PARA							
Access																									RW							

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5:0	PARA	补偿参数 根据频率设定对应的高速补偿参数 $RTC_HCOMP.PARA = f/32768/32$ f 为补偿频率，建议配置为 32.768kHz 的整数倍： ● 0.06ppm 补偿精度：f 取值范围是 16~48MHz。 ● 0.96ppm 补偿精度：f 取值范围是 4~48MHz。	RW

具体高速补偿参数请参考下表：

补偿频率 f（MHz）	补偿参数 PARA
4	0b000100
6	0b000110
8	0b001000
12	0b001100
16	0b001111
20	0b010011
24	0b010111
32	0b011111
48	0b101110

14.5.13 控制寄存器 2 (RTC_CR2)

Offset	Bit Position																																				
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																																				
Name	Reserved																							ALMBE	ALMAE	Reserved	WUTE	TSE	Reserved	TSEDG	WUCKSEL						
Access																								RW	RW		RW	RW		RW	TSEDG	RW					

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10	ALMBE	闹钟 B 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
9	ALMAE	闹钟 A 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
8	Reserved	-	-
7	WUTE	唤醒定时器使能 ● 0b0: 禁止 ● 0b1: 使能	RW
6	TSE	时间戳使能 ● 0b0: 禁止 ● 0b1: 使能	RW
5:4	Reserved	-	-
3	TSEDG	时间戳边沿选择 ● 0b0: 上升沿 ● 0b1: 下降沿	RW
2:0	WUCKSEL	唤醒定时器时钟源选择 ● 0b000: RTC/2 ● 0b001: RTC/4 ● 0b010: RTC/8 ● 0b011: RTC/16 ● 0b100: RTC_Sec/1 ● 0b101: RTC_Sec/2 ● 0b110: RTC_Sec/4 ● 0b111: RTC_Sec/8	RW

14.5.14 闹钟 A 寄存器 (RTC_ALMA)

Offset	Bit Position																																			
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0xFFFFFFFF																																			
Name	Reserved	WEEK							MSKH	Reserved	HOURH				HOURL				MSKM	MINH				MINL				MSKS	SECH				SECL			
Access		RW							RW		RW				RW				RW	RW				RW				RW	RW				RW			

位/位域	标记	功能描述	读写
31	Reserved	-	-
30:24	WEEK	闹钟星期位 Bit24~Bit30 分别对应周日~周六： 对应位值为 1 时，代表每周该日闹钟有效。 例如： ● 设置为 0b0000001 代表周日闹钟设定有效。 ● 设置为 0b0000011 代表周日、周一闹钟设定有效。 ● 设置为 0b1000000 代表周六闹钟设定有效。	RW
23	MSKH	闹钟时位屏蔽位 ● 0b0：时位匹配才会产生闹钟 A ● 0b1：闹钟 A 中时位无关	RW
22	Reserved	-	-
21:20	HOURH	闹钟时十位值 闹钟时十位值 BCD 码	RW
19:16	HOURL	闹钟时个位值 闹钟时个位值 BCD 码	RW
15	MSKM	闹钟分位屏蔽位 ● 0b0：分位匹配才会产生闹钟 A ● 0b1：闹钟 A 中分位无关	RW
14:12	MINH	闹钟分十位值 闹钟分十位值 BCD 码	RW
11:8	MINL	闹钟分个位值 闹钟分个位值 BCD 码	RW
7	MSKS	闹钟秒位屏蔽位 ● 0b0：秒位匹配才会产生闹钟 A ● 0b1：闹钟 A 中秒位无关	RW
6:4	SECH	闹钟秒十位值 闹钟秒十位值 BCD 码	RW
3:0	SECL	闹钟秒个位值 闹钟秒个位值 BCD 码	RW



说明

位域操作时需注意，当配置 12 小时制时，HOURL 寄存器不能写入 0。

14.5.15 闹钟 B 寄存器 (RTC_ALMB)

Offset	Bit Position																																			
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0xFFFFFFFF																																			
Name	Reserved	WEEK							MSKH	Reserved	HOURH				HOURL				MSKM	MINH				MINL				MSKS	SECH				SECL			
Access		RW							RW		RW				RW				RW	RW				RW				RW	RW				RW			

位/位域	标记	功能描述	读写
31	Reserved	-	-
30:24	WEEK	闹钟星期位 Bit24~Bit30 分别对应周日~周六： 对应位值为 1 时，代表每周该日闹钟有效。 例如： ● 设置为 0b0000001 代表周日闹钟设定有效。 ● 设置为 0b0000011 代表周日、周一闹钟设定有效。 ● 设置为 0b1000000 代表周六闹钟设定有效。	RW
23	MSKH	闹钟时位屏蔽位 ● 0b0：时位匹配才会产生闹钟 B ● 0b1：闹钟 B 中时位无关	RW
22	Reserved	-	-
21:20	HOURH	闹钟时十位值 闹钟时十位值 BCD 码	RW
19:16	HOURL	闹钟时个位值 闹钟时个位值 BCD 码	RW
15	MSKM	闹钟分位屏蔽位 ● 0b0：分位匹配才会产生闹钟 B ● 0b1：闹钟 B 中分位无关	RW
14:12	MINH	闹钟分十位值 闹钟分十位值 BCD 码	RW
11:8	MINL	闹钟分个位值 闹钟分个位值 BCD 码	RW
7	MSKS	闹钟秒位屏蔽位 ● 0b0：秒位匹配才会产生闹钟 B ● 0b1：闹钟 B 中秒位无关	RW
6:4	SECH	闹钟秒十位值 闹钟秒十位值 BCD 码	RW
3:0	SECL	闹钟秒个位值 闹钟秒个位值 BCD 码	RW



说明

位域操作时需注意，当配置 12 小时制时，HOURL 寄存器不能写入 0。

14.5.16 时间戳时间寄存器 (RTC_TSTR)

Offset	Bit Position																																	
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																																	
Name	Reserved										HOURH		HOURL				Reserved	MINH				MINL				Reserved	SECH				SECL			
Access											RO		RO					RO				RO					RO				RO			

位/位域	标记	功能描述	读写
31:22	Reserved	-	-
21:20	HOURH	时计数十位值 时计数十位值 BCD 码  说明 该位为 24 小时制，不受 RTC_CR0.AMPM 影响。	RO
19:16	HOURL	时计数个位值 时计数个位值 BCD 码  说明 该位为 24 小时制，不受 RTC_CR0.AMPM 影响。	RO
15	Reserved	-	-
14:12	MINH	分计数十位值 分计数十位值 BCD 码	RO
11:8	MINL	分计数个位值 分计数个位值 BCD 码	RO
7	Reserved	-	-
6:4	SECH	秒计数十位值 秒计数十位值 BCD 码	RO
3:0	SECL	秒计数个位值 秒计数个位值 BCD 码	RO

14.5.17 时间戳日期寄存器 (RTC_TSDR)

Offset	Bit Position																															
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000 (只有上电与模块复位对该寄存器复位有效)																															
Name	Reserved																WEEK				MON				Reserved		DAYH		DAYL			
Access																	RO				RO						RO		RO			

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:13	WEEK	周计数值 周计数值，格式同 RTC_WEEK	RO
12:8	MON	月计数值 月计数值，格式同 RTC_MON	RO
7:6	Reserved	-	-
5:4	DAYH	日计数十位值 日计数十位值 BCD 码	RO
3:0	DAYL	日计数个位值 日计数个位值 BCD 码	RO

14.5.18 唤醒定时器寄存器 (RTC_WUTR)

Offset	Bit Position																															
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FFFF（只有上电与模块复位对该寄存器复位有效）																															
Name	Reserved																ARR															
Access																	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	ARR	唤醒定时器周期值	RW

14.5.19 中断使能寄存器 (RTC_IER)

Offset	Bit Position																																			
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																																			
Name	Reserved																								PRD		Reserved		TS		WU		ALMB		ALMA	
Access																									RW				RW		RW		RW		RW	

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	PRD	周期中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
5:4	Reserved	-	-
3	TS	时间戳中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
2	WU	定时中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
1	ALMB	闹钟 B 中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
0	ALMA	闹钟 A 中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW

14.5.20 中断标志寄存器 (RTC_IFR)

Offset	Bit Position																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Reset	0x00000000（只有上电与模块复位对该寄存器复位有效）																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
Name	Reserved																										RO	PRD	Reserved	RO	TSOV	RO	TS	RO	WU	RO	ALMB	RO	ALMA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
Access																											RO																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	PRD	周期中断标志 ● 0b0: 未发生中断 ● 0b1: 已发生中断	RO
5	Reserved	-	-
4	TSOV	时间戳溢出标志 ● 0b0: 未发生溢出 ● 0b1: 已发生溢出	RO
3	TS	时间戳中断标志 ● 0b0: 未发生中断 ● 0b1: 已发生中断	RO
2	WU	定时中断标志 ● 0b0: 未发生中断 ● 0b1: 已发生中断	RO
1	ALMB	闹钟 B 中断标志 ● 0b0: 未发生中断 ● 0b1: 已发生中断	RO
0	ALMA	闹钟 A 中断标志 ● 0b0: 未发生中断 ● 0b1: 已发生中断	RO

14.5.21 中断清除寄存器 (RTC_ICR)

Offset	Bit Position																																	
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x000003FF																																	
Name	Reserved																											PRD	Reserved	TSOV	TS	WU	ALMB	ALMA
Access																											WO		WO	WO	WO	WO	WO	WO

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	PRD	周期中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
5	Reserved	-	-
4	TSOV	时间戳溢出标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
3	TS	时间戳中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
2	WU	定时中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
1	ALMB	闹钟 B 中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
0	ALMA	闹钟 A 中断标志清除 写 0 清除标志, 写 1 无效; 读出恒为 1	WO

14.5.22 保护寄存器 (RTC_WPR)

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								WPR							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	WPR	保护序列 对 WPR 先写 0xCA, 再写 0x53, 解除 RTC 寄存器写保护; 对 WPR 先写 0xCA, 再写 0xFF (不等于 0x53), 开启 RTC 寄存器写保护。 除了 WPR 与 ICR, RTC 寄存器所有寄存器处于写保护状态, 上电默认不可以写。	WO

15 独立看门狗 (IWDG)

15.1 简介

本产品具有一个嵌入式独立看门狗外设，具有安全性高和使用灵活的优点。此独立看门狗可检测并解决由软件错误导致的故障，在其计数值达到设定的超时值时触发中断或复位。该看门狗由 IWDGCLK 时钟（时钟源为 RC10K）驱动，即使主时钟发生故障时仍可正常工作。

15.2 主要特性

- 独立的 RC 振荡器
- 溢出时触发中断或复位
- 通过特定写序列，可以停止正在运行的看门狗
- DeepSleep 模式时可配置为自动暂停计数

15.3 功能说明

15.3.1 功能框图

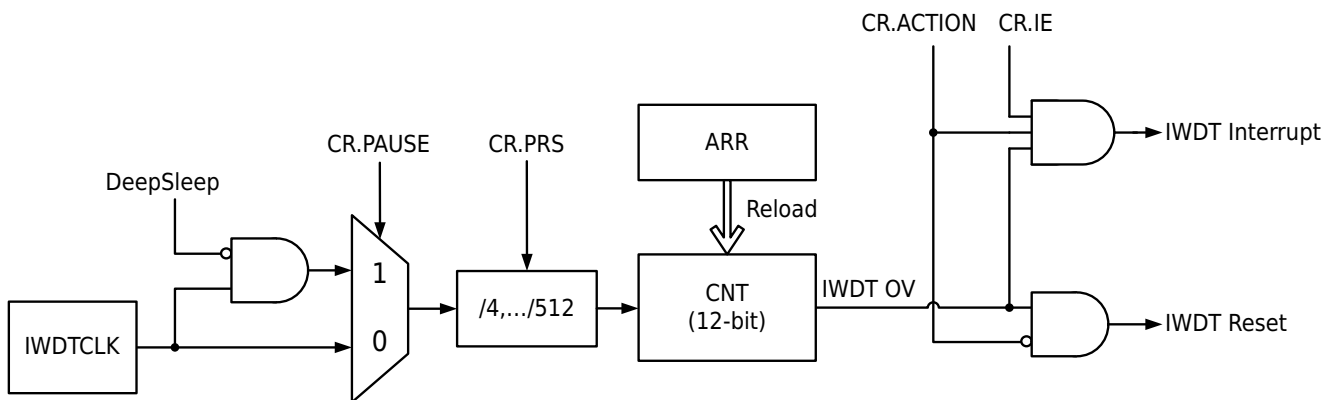


图 15-1 IWDG 功能框图

启动看门狗后，计数器从 0xFFF 开始递减计数。每当向 IWDG_KR 寄存器写入 0xAAAA 时，IWDG_ARR 的值就会被重载到计数器。当计数值递减到 0x000 时，看门狗会产生溢出信号；当对计数器进行重载时计数值大于 IWDG_WINR 寄存器的值也会产生溢出信号。配置 IWDG_CR 相应控制位可使看门狗溢出后产生复位或中断。通过配置 IWDG_CR 相应控制位可使看门狗在 DeepSleep 时自动暂停计数。

15.3.2 窗口看门狗模式

配置 IWDG_WINR 寄存器的值小于 IWDG_ARR 时，IWDG 工作于窗口看门狗模式。在该模式下，有两种条件触发看门狗溢出：计数值递减到 0x000、在计数值大于窗口值时对看门狗进行重载。更改 IWDG_WINR 寄存器值时，硬件会自动对看门狗进行重载操作，将递减计数器值复位为 IWDG_ARR。用户程序应当在计数值小于等于 IWDG_WINR 寄存器数值且计数值递减到 0x000 前对看门狗进行重载操作。

15.3.3 独立看门狗模式

配置 IWDG_WINR 寄存器的值大于等于 IWDG_ARR 时，IWDG 工作于独立看门狗模式。在该模式下，仅有一种条件会触发看门狗溢出：计数值递减到 0x000。用户程序应当在计数值递减到 0x000 前对看门狗进行重载操作。

15.3.4 看门狗溢出处理

看门狗的溢出信号，可以触发中断也可以触发复位。当设置 IWDG_CR.ACTION 为 0 时，看门狗的溢出信号会复位 MCU；当设置 IWDG_CR.ACTION 为 1 且 IWDG_CR.IE 为 1 时，看门狗的溢出信号会触发中断，通过向 IWDG_SR 寄存器写 0，可以将 IWDG_SR.OV 标志位清 0。

15.3.5 看门狗暂停

若设置 IWDG_CR.PAUSE 为 1，则 MCU 进入 DeepSleep 时看门狗自动暂停运行；MCU 退出 DeepSleep 时，看门狗继续运行。使用该特性，则 MCU 在低功耗状态时不用频繁唤醒以喂狗，可实现更低的待机功耗。

15.3.6 看门狗停止及再次运行

向 IWDG_KR 寄存器依次写入 0x1234、0x5678 可以停止正在运行的看门狗。

向 IWDG_KR 寄存器写入 0xCCCC 即可使看门狗重新运行，其配置保持不变。

15.3.7 寄存器写保护

IWDG_CR、IWDG_ARR 和 IWDG_WINR 寄存器具有写保护。若需修改这三个寄存器，应在看门狗启动后向 IWDG_KR 寄存器写入 0x5555 以解除这些寄存器的写保护；向 IWDG_KR 寄存器写入其它值则会使能这些寄存器的写保护。

15.4 操作示例

15.4.1 配置 IWDG 为窗口看门狗

- 步骤 1. 向 IWDG_KR 寄存器写入 0xCCCC，使能 IWDG。
- 步骤 2. 向 IWDG_KR 寄存器写入 0x5555，解除寄存器写保护。
- 步骤 3. 配置 IWDG_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- 步骤 4. 配置 IWDG_ARR，选择看门狗的溢出周期。
- 步骤 5. 等待 IWDG_SR.ARRF 为 0。
- 步骤 6. 向 IWDG_WINR 写入需要的数值，配置窗口大小。
- 步骤 7. 等待 IWDG_SR.WINRF 变为 0，自动刷新计数器值为 IWDG_ARR。
- 步骤 8. 在计数值小于等于 IWDG_WINR 寄存器数值且计数值递减到 0 前，向 IWDG_KR 寄存器写入 0xAAAA，对 IWDG 进行重载。

15.4.2 配置 IWDG 为独立看门狗

- 步骤 1. 向 IWDG_KR 寄存器写入 0xCCCC，使能 IWDG。
- 步骤 2. 向 IWDG_KR 寄存器写入 0x5555，解除寄存器写保护。
- 步骤 3. 配置 IWDG_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- 步骤 4. 配置 IWDG_ARR，选择看门狗的溢出周期。
- 步骤 5. 等待 IWDG_SR.ARRF 为 0。
- 步骤 6. 刷新计数器值为 IWDG_ARR 的值 (IWDG_KR=0x0000AAAA)。
- 步骤 7. 在计数值递减到 0 前，向 IWDG_KR 寄存器写入 0xAAAA，对 IWDG 进行重载。

15.4.3 看门狗超时时间 (10kHz 的独立时钟)

表 15-1 IWDG 超时时间 (10kHz 的独立时钟)

预分频系数	PRS[2:0]位	最短时间 (ms) ARR[11:0]=0x000	最长时间 (ms) ARR[11:0]=0xFFFF
4 分频	0b000	$0.1 \times 4 = 0.4$	$0.1 \times 4 \times 4096 = 1638.4$
8 分频	0b001	$0.1 \times 8 = 0.8$	$0.1 \times 8 \times 4096 = 3276.8$
16 分频	0b010	$0.1 \times 16 = 1.6$	$0.1 \times 16 \times 4096 = 6553.6$
32 分频	0b011	$0.1 \times 32 = 3.2$	$0.1 \times 32 \times 4096 = 13107.2$
64 分频	0b100	$0.1 \times 64 = 6.4$	$0.1 \times 64 \times 4096 = 26214.4$
128 分频	0b101	$0.1 \times 128 = 12.8$	$0.1 \times 128 \times 4096 = 52428.8$
256 分频	0b110	$0.1 \times 256 = 25.6$	$0.1 \times 256 \times 4096 = 104857.6$
512 分频	0b111	$0.1 \times 512 = 51.2$	$0.1 \times 512 \times 4096 = 209715.2$

15.5 寄存器

15.5.1 寄存器总表

基地址：0x40001000

表 15-2 独立看门狗 (IWDT) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x80	IWDT_KR	键值寄存器	32
0x84	IWDT_CR	控制寄存器	32
0x88	IWDT_ARR	重载寄存器	32
0x8C	IWDT_SR	状态寄存器	32
0x90	IWDT_WINR	窗口寄存器	32

15.5.2 键值寄存器 (IWDT_KR)

Offset	Bit Position																															
0x80	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																KR															
Access																																

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	KR	IWDT 键值 ● 写入 0xCCCC：启动 IWDT 计数器 ● 写入 0xAAAA：重载 IWDT 计数值 ● 依次写入 0x1234、0x5678：停止看门狗 ● 写入 0x5555：解除 IWDT_ARR、IWDT_WINR、IWDT_CR 的写保护 ● 写入非 0x5555：使能 IWDT_ARR、IWDT_WINR、IWDT_CR 的写保护	WO

15.5.3 控制寄存器 (IWDT_CR)



说明

- 需要向 IWDT_KR 寄存器写入 0x5555 后，才可修改本寄存器。
- IWDT_SR.CRF 为 0，才可修改本寄存器。

Offset	Bit Position																																			
0x84	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000000																																			
Name	Reserved																										PAUSE		ACTION		IE		Reserved		PRS	
Access																											RW		RW		RW				RW	

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	PAUSE	DeepSleep 模式下 WDT 暂停控制 ● 0b0: DeepSleep 模式下 IWDT 继续运行 ● 0b1: DeepSleep 模式下 IWDT 自动暂停	RW
5	ACTION	IWDT 溢出后动作配置 ● 0b0: 看门狗溢出后产生复位 ● 0b1: 看门狗溢出后产生中断	RW
4	IE	IWDT 中断使能控制 ● 0b0: 禁止看门狗中断 ● 0b1: 使能看门狗中断	RW
3	Reserved	-	-
2:0	PRS	配置看门狗计数时钟与 RC10K 振荡器的分频比 ● 0b000: 4 分频 ● 0b001: 8 分频 ● 0b010: 16 分频 ● 0b011: 32 分频 ● 0b100: 64 分频 ● 0b101: 128 分频 ● 0b110: 256 分频 ● 0b111: 512 分频	RW

15.5.4 重载寄存器 (IWDT_ARR)



说明

- 向 IWDT_KR 寄存器写入 0x5555 后，才可修改本寄存器。
- IWDT_SR.ARRF 为 0，才可修改本寄存器。

Offset	Bit Position																															
0x88	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000FFF																															
Name	Reserved																				RW ARR											
Access																																

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	ARR	IWDT 重载值	RW

15.5.5 状态寄存器 (IWDT SR)



说明

向该寄存器写 0，可以将 IWDG SR.OV 标志位清 0。

[illegible]

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4	RUN	WDT 运行标志 0b0: WDT 未运行 0b1: WDT 正在运行	RO
3	OV	WDT 溢出标志 写 0 清除溢出标志，写 1 对该位的值无影响 读取： 0b0: WDT 未发生溢出 0b1: WDT 发生溢出	RW
2	WINRF	WINR 寄存器更新标志 0b0: WINR 寄存器更新完成 0b1: WINR 寄存器正在更新  注意 只有 WINRF 为 0 时才可以更新 WINR 寄存器。	RO
1	ARRF	ARR 寄存器更新标志 0b0: ARR 寄存器更新完成 0b1: ARR 寄存器正在更新  注意 只有 ARRF 为 0 时才可以更新 ARR 寄存器。	RO
0	CRF	CR 寄存器更新标志 0b0: CR 寄存器更新完成 0b1: CR 寄存器正在更新  注意 只有 CRF 为 0 时才可以更新 CR 寄存器。	RO

15.5.6 窗口寄存器（IWDT_WINR）



说明

- 向 IWDT_KR 寄存器写入 0x5555 后，才可修改本寄存器。
- IWDT_SR.WINRF 为 0，才可修改本寄存器。

Offset	Bit Position																															
0x90	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000FFF																															
Name	Reserved																WINR															
Access																	RW															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	WINR	窗口比较器比较值	RW

16 窗口看门狗 (WWDT)

16.1 简介

窗口看门狗 (WWDT) 通常用于监测由非预期错误所触发的软件故障。当计数器数值递减到 0x40 时产生中断，当计数器数值递减到 0x3F 时产生复位；若用户程序在计数器数值大于窗口值时更新计数器值则产生复位。故用户程序只能在限定的窗口内（计数器值小于等于窗口值且大于 0x3F）更新 WWDT 计数器的数值。

16.2 主要特性

- 7-bit 递减计数器
- PCLK 分频时钟作为计数时钟
- 当 PCLK 为 48MHz 时，其超时时间为 85μs ~ 699ms
- 计数值递减到 0x40 时产生中断
- 计数值递减到 0x3F 时产生复位
- 计数值大于窗口值时更新计数器产生复位

16.3 功能说明

16.3.1 功能框图

启动看门狗后，计数器 WCNT 的数值在每个 WWDTCLK 的上升沿递减 1。当计数器递减到 0x40 时看门狗会产生预溢出信号 POV，使能 WWDT_CR1.IE 可使看门狗产生预溢出中断；当计数器递减到 0x3F 时看门狗会产生复位信号。用户程序可通过写 WWDT_CR0.WCNT 以更新 WCNT 从而阻止看门狗产生复位信号。若用户更新 WCNT 时，WCNT 的数值大于 WINR 的数值，则看门狗也会产生复位信号。

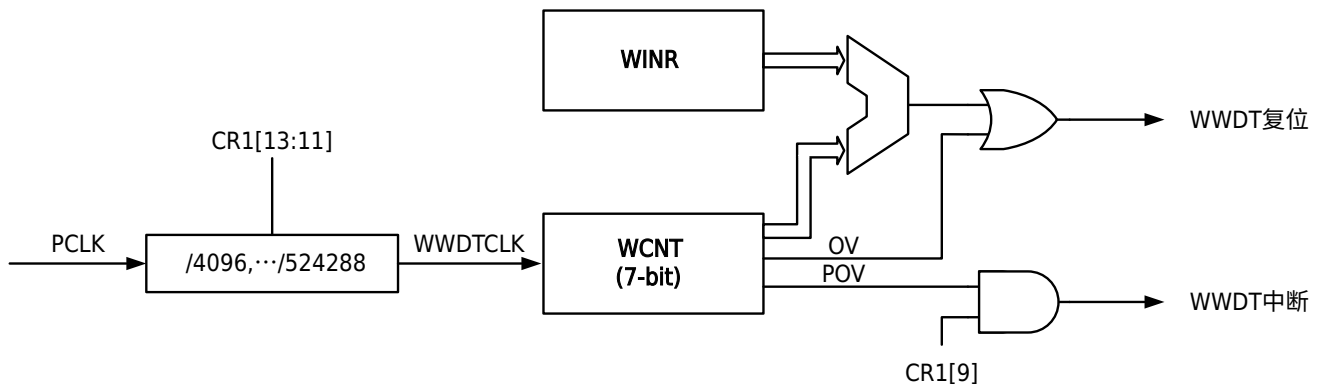


图 16-1 WWDT 功能框图

16.3.2 使能看门狗

系统复位后，看门狗处于关闭状态，设置 WWDT_CR0.EN 为 1 即可使能看门狗。用户程序不能通过向 WWDT_CR0.EN 写入 0 来关闭已经使能的看门狗。建议采用向 WWDT_CR0 写入 0xFF 的方式使能看门狗，以确保使能看门狗时不会触发复位。

16.3.3 配置看门狗超时

通过合理配置 WWDT_CR0.WCNT 和 WWDT_CR1.PRS 即可获得所需要的看门狗超时周期。当 PCLK 为 48MHz 时，其超时周期的范围为 85μs ~ 699ms。

超时时间计算公式如下：

$$T_{WWDT} = T_{PCLK} \times 4096 \times 2^{CR1.PRS} \times (CR0.WCNT - 0x3F)$$

16.3.4 更新递减计数器

向 WWDT_CR0.WCNT 写入数据，即可更新 WWDT 递减计数器。若写入的数值小于等于 0x3F 则直接发生复位；若更新计数器时当前计数值大于窗口值则直接发生复位。故用户程序只能在递减计数器的数值小于等于窗口值且大于 0x3F 时，对递减计数器进行更新。

16.3.5 预溢出中断

若递减计数器递减到 0x40 且 WWDT_CR1.IE 为 1 则 WWDT 会产生预溢出中断，用户程序可以在中断服务程序中更新递减计数器以阻止 WWDT 产生复位；退出中断服务前用户程序应向 WWDT_SR 寄存器写入 0x00 以清除 WWDT 预溢出标志。

16.4 操作示例

16.4.1 配置看门狗窗口溢出周期示例

背景信息

以下示例配置看门狗的溢出周期为 786432 个 PCLK，窗口为溢出周期的 1/3。

操作步骤

- 步骤 1. 配置 WWDT_CR1.PRS 为 2，WWDT 的计数时钟为 PCLK/16384。
- 步骤 2. 配置 WWDT_CR1.WINR 为 0x4F，WWDT 的更新窗口为 0x4F~0x40。
- 步骤 3. 配置 WWDT_CR0.WCNT 为 0x6F，WWDT 的溢出周期为 786432 个 PCLK。
- 步骤 4. 设置 WWDT_CR0.EN 为 1，启动看门狗。
- 步骤 5. 当递减计数器数值小于等于窗口值且大于 0x3F 时，向 WWDT_CR0.WCNT 写入 0x6F，重载看门狗计数器。

16.5 寄存器

16.5.1 寄存器总表

基地址：0x4000B800

 注意

WWDT 产生复位后影响寄存器值。

表 16-1 窗口看门狗 (WWDT) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	WWDT_CR0	控制寄存器 0	32
0x04	WWDT_CR1	控制寄存器 1	32
0x08	WWDT_SR	状态寄存器	32

16.5.2 控制寄存器 0 (WWDT_CR0)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000007F																															
Name	Reserved																								RW EN		RW WCNT					
Access																									RW EN		RW WCNT					

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7	EN	WWDT 使能控制 该位置 1 后不能软件清零 ● 0b0：禁止 WWDT ● 0b1：使能 WWDT	RW
6:0	WCNT	读：WWDT 计数器当前计数值 写：更新 WWDT 的计数器当前计数值	RW

16.5.3 控制寄存器 1 (WWDT_CR1)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000007F																															
Name	Reserved																PRS		Reserved		IE		Reserved		WINR							
Access																	RW				RW				RW							

位/位域	标记	功能描述	读写
31:14	Reserved	-	-
13:11	PRS	WWDT 计数时钟配置 0b000: PCLK / 4096 0b001: PCLK / 8192 0b010: PCLK / 16384 0b011: PCLK / 32768 0b100: PCLK / 65536 0b101: PCLK / 131072 0b110: PCLK / 262144 0b111: PCLK / 524288	RW
10	Reserved	-	-
9	IE	WWDT 预溢出中断使能控制 0b0: 禁止预溢出中断 0b1: 使能预溢出中断	RW
8:7	Reserved	-	-
6:0	WINR	看门狗窗口值 用户程序只能在计数器值小于等于窗口值且大于 0x3F 更新 WWDT 计数器的数值	RW

16.5.4 状态寄存器 (WWDT_SR)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															POV
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	POV	WWDT 预溢出标志 写 0 清除预溢出标志，写 1 对该位的值无影响 读取： ● 0b0：WWDT 未发生预溢出 ● 0b1：WWDT 已发生预溢出	RW

17 低功耗同步异步收发器 (LPUART)

17.1 简介

低功耗同步异步接收器 (LPUART) 是一种允许在微控制器低功耗模式下进行全双工通信的 UART。当微控制器处于 DeepSleep 模式时, LPUART 也可以正常接收 UART 数据帧, 通过接收完成中断唤醒微控制器。

17.2 主要特性

- 配置时钟 PCLK
- 传输时钟 SCLK (SCLK 可选择 XTL、RCL 以及 PCLK)
- 支持同步半双工、异步全双工、单线半双工传输
- 可编程串行通信功能
 - ▶ 两种字符长度: 8 比特、9 比特
 - ▶ 三种校验方式: 无检验、奇校验、偶校验
 - ▶ 三种停止长度: 1 比特、1.5 比特、2 比特
- 支持低功耗模式下收发数据
- 16-bit 波特率计数器
- 支持硬件流控 (RTS、CTS)
- 支持多机通讯、自动地址识别
- 支持通过 DMA 数据传输

17.3 功能说明

17.3.1 功能框图

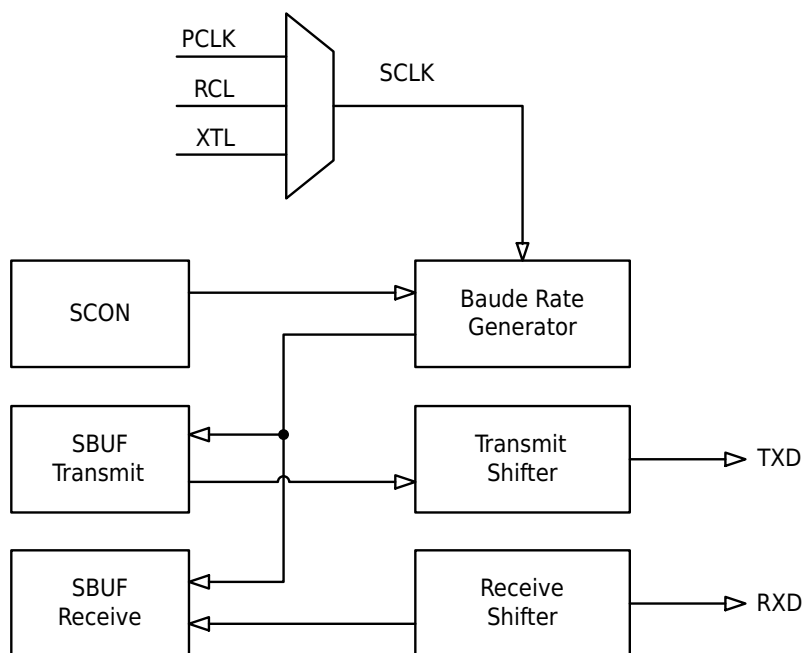


图 17-1 LPUART 功能框图

17.3.2 时钟说明

LPUART 模块有两个时钟：

- 配置时钟 PCLK

配置时钟用于系统 APB 总线对 LPUART 模块进行寄存器配置，配置时钟固定为 APB 总线时钟 PCLK。当系统进入 DeepSleep 模式，PCLK 时钟将会停止。

- 传输时钟 SCLK

传输时钟用于 LPUART 数据收发逻辑工作，其时钟来源可选择外部低速晶振时钟（XTL）、内部低速时钟（RCL）以及 PCLK 时钟。

当系统进入 DeepSleep 模式，如果 SCLK 选择外部低速晶振时钟（XTL）或者内部低速时钟（RCL），LPUART 仍旧可以进行正常的收发数据，而不受系统 DeepSleep 模式的影响。

17.3.3 工作模式

LPUART 支持多种工作模式：同步半双工模式、异步全双工模式、单线半双工模式。

通过 LPUART_SCON.SM 和 LPUART_SCON.HDSEL 搭配，即可配置出所需要的各种工作模式。

17.3.3.1 Mode0~Mode3 功能对比

配置 LPUART_SCON.SM 可选择不同的传输模式：Mode0~Mode3。这四种工作模式的主要功能对比如下表所示：

表 17-1 LPUART Mode0/1/2/3 数据结构

工作模式		传输位宽	数据组成	波特率
Mode0	同步半双工模式	8-bit	Data(8bits)	$BaudRate = \frac{f_{SCLK}}{12}$
Mode1	异步全双工模式	10~11-bit	Start(1bit) + Data(8bits) + Stop(1~2bits)	$BaudRate = \frac{f_{SCLK}}{OVER \times SCNT}$
Mode2	异步全双工模式	11~12-bit	Start(1bit) + Data(8bits) + B8(1bit) + Stop(1~2bits)	$BaudRate = \frac{f_{SCLK}}{OVER}$
Mode3	异步全双工模式	11~12-bit	Start(1bit) + Data(8bits) + B8(1bit) + Stop(1~2bits)	$BaudRate = \frac{f_{SCLK}}{OVER \times SCNT}$

 **说明**

- 当 SCLK 选择 XTL（频率 32.768kHz）时，Mode1/2/3 可以支持 9600/4800 波特率，波特率的产生详见寄存器 LPUART_BSEL/LPUART_MODU。
- Mode0 只能作为主机发送 LPUART 同步移位时钟，不可以作为从机接收外部输入的 LPUART 同步移位时钟。
- f_{SCLK} 为 SCLK 的时钟频率。
- OVER 取值：
 - ▶ LPUART_SCON.OVER=0b00 时，Mode1/3 模式 OVER 取值为 16；Mode2 模式 OVER 取值为 32。
 - ▶ LPUART_SCON.OVER=0b01 时，Mode1/3 模式 OVER 取值为 8；Mode2 模式 OVER 取值为 16。
 - ▶ LPUART_SCON.OVER=0b10 时，Mode1/3 模式 OVER 取值为 4；Mode2 模式 OVER 取值为 8。
- SCNT 的定义详见寄存器 LPUART_SCNT 介绍。
- B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

表 17-2 B8 数据含义

应用场景	LPUART_SCON.ADRDET	LPUART_SCON.B8CONT[1:0]	B8 数据含义
奇偶校验	-	0b01/0b10	接收时，B8 是所收到的 8bits 数据的奇偶校验位。 发送时，B8 是所发送的 8bits 数据的奇偶校验位。
多机通讯	0b1	-	B8=1，代表当前是地址帧。 B8=0，代表当前是数据帧。
其他	0b0	0b00/0b11	接收/发送数据 Bit8。

当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制。

17.3.3.2 Mode0 数据收发

● Mode0 发送数据

当设置 LPUART_SCON.REN 为 0 时 LPUART 工作于发送状态，将待发送的数据写入 LPUART_SBUF 寄存器中，则数据将从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

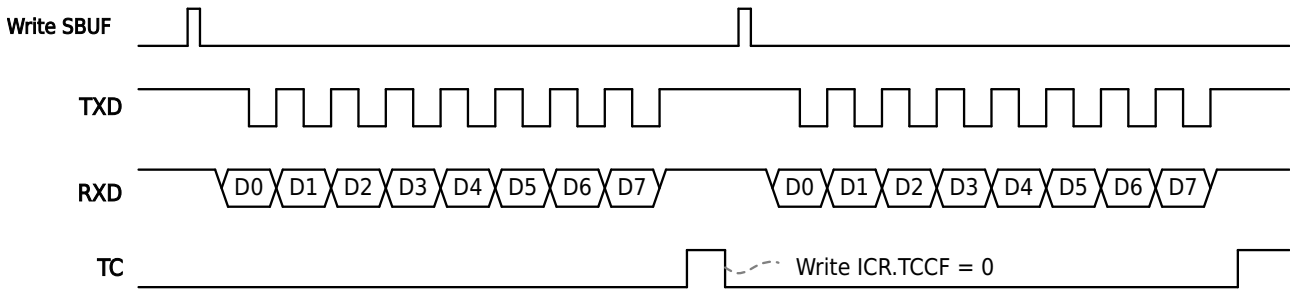


图 17-2 Mode0 发送数据

● Mode0 接收数据

当设置 LPUART_SCON.REN 为 1 时 LPUART 工作于接收状态。每当 LPUART_ISR.RC 标志为 0 时，TXD 管脚会输出 8 个同步移位时钟，数据将从 RXD 输入（低位在先，高位在后）。每次将 LPUART_ISR.RC 标志清零即可接收下一个字节。

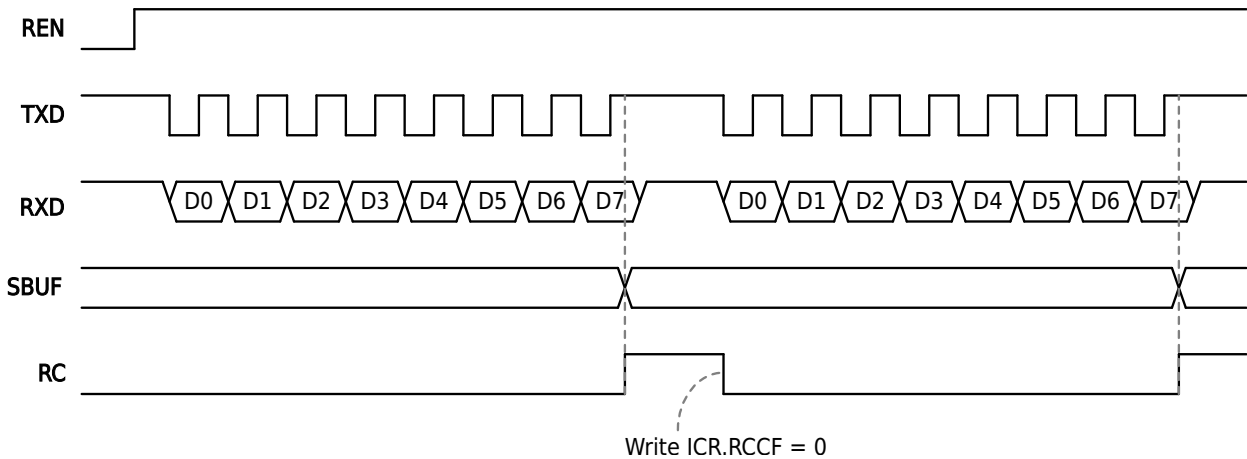


图 17-3 Mode0 接收数据

17.3.3.3 Mode1 数据收发

● Mode1 发送数据

无论 LPUART_SCON.REN 为 0 或 1，都可以正常发送数据。
将数据写入 LPUART_SBUF，硬件自动将数据加载到发送移位寄存器并从 TXD 管脚移位输出（低位在先，高位在后），LPUART_ISR.TXE 标志硬件置 1。
用户查询到 LPUART_ISR.TXE 标志为 1 时，即可向 LPUART_SBUF 寄存器写入下一个待发送的数据。
每当 TXD 管脚发送完成一帧数据时，LPUART_ISR.TC 标志会被硬件置 1 以指示当前的发送进展。

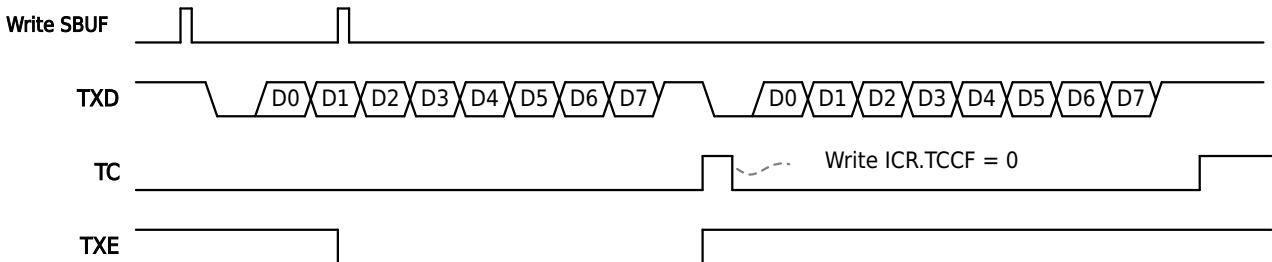


图 17-4 Mode1 发送数据

● Mode1 接收数据

接收数据时，将 LPUART_SCON.REN 设置为 1，并将 LPUART_ISR.RC 位清 0。
当一帧数据接收完成，LPUART_ISR.RC 标志会被硬件置 1，接收到的数据从 LPUART_SBUF 寄存器中读出。

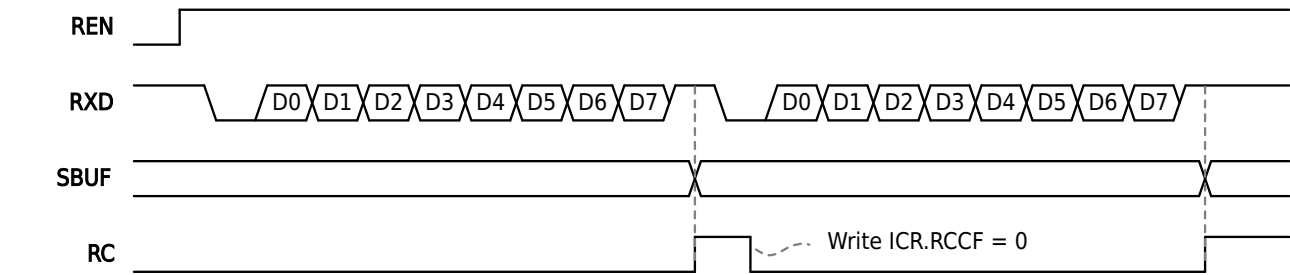


图 17-5 Mode1 接收数据

17.3.3.4 Mode2/Mode3 数据收发

● Mode2/3 发送数据

无论 LPUART_SCON.REN 为 0 或 1，都可以正常发送数据。
将数据写入 LPUART_SBUF，硬件自动将数据加载到发送移位寄存器并从 TXD 管脚移位输出（低位在先，高位在后），LPUART_ISR.TXE 标志硬件置 1。
用户查询到 LPUART_ISR.TXE 标志为 1 时，即可向 LPUART_SBUF 寄存器写入下一个待发送的数据。
每当 TXD 管脚发送完成一帧数据时，LPUART_ISR.TC 标志会被硬件置 1 以指示当前的发送进展。

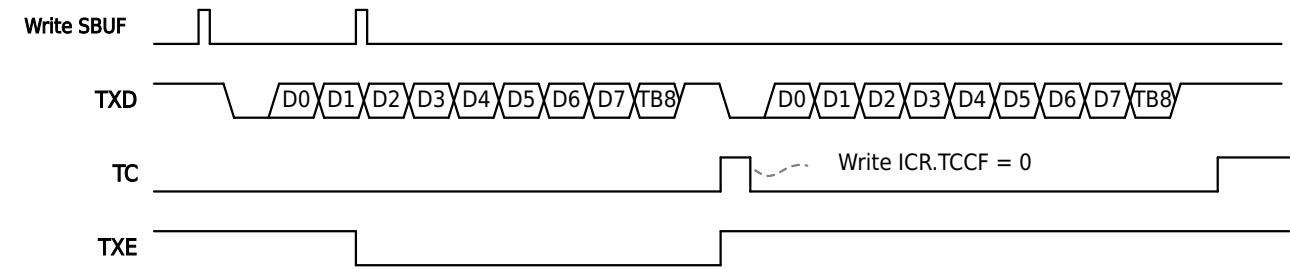


图 17-6 Mode2/Mode3 发送数据

● Mode2/3 接收数据

接收数据时，需要将 LPUART_SCON.REN 设置为 1，并将 LPUART_ISR.RC 位清 0。
每当一帧数据接收完成，LPUART_ISR.RC 标志会被硬件置 1，接收到的数据可以从 LPUART_SBUF 寄存器中读出。

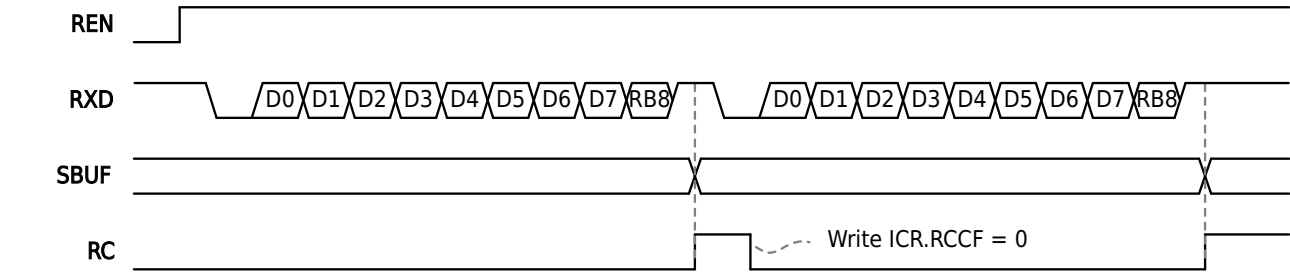


图 17-7 Mode2/Mode3 接收数据

17.3.3.5 单线半双工数据收发

当 LPUART_SCON.HDSEL=0b1，可进入单线半双工传输模式：

- TX 与 RX 信号线在模块内部相连。

- RX 信号不再使用，接收发送的数据都经由 TX 信号完成。TX 信号的方向控制由硬件逻辑完成，无需软件控制。
- 配置 LPUART_SCON.HDSEL=0b1 后，要等待一位数据的发送时长（即 1/BaudRate 秒后），才能正常接收。
- 当发送缓存为空时，TX 信号始终为输入（接收状态）。一旦向发送缓存填入一个数据，TX 信号变为输出（发送状态）。当发送完成，发送缓存变为空，TX 信号又回到输入（接收状态）。
- 无数据传输时，TXD 引脚始终处于输入状态。用户程序需配置该引脚为开漏输出并外接上拉电阻。

**注意**

1. 即使模块处于接收过程中，只要发送缓存被填入数据，TX 信号立即变为输出（发送状态）。接收过程将被打断，并得到一个错误的接收数据。
2. 单线半双工模式与 Mode0 模式的异同：
 - 同：两者都是用一根信号分时复用地完成了数据收发。
 - 异：Mode0 为同步传输，有同步时钟。而单线半双工模式为异步传输，所有的波特率计算还是遵照 Mode1~Mode3 的波特率公式，仅仅是收发功能复用到了 TX 这一根信号上面。

单线半双工传输支持 9600/4800 波特率，但是需要注意，在 9600/4800 波特率下使用单线半双工模式接收数据时，需要遵循以下步骤：

- 步骤 1.** 参考《端口控制器 (GPIO)》章节管脚数字复用功能的相关描述，将 TXD 映射到需要的管脚；并配置 TXD 为开漏输出并使能上拉电阻。
- 步骤 2.** 配置 SYSCTRL 模块寄存器 PERI_CLKEN0.LPUARTx 为 1，使能 LPUART 配置时钟及工作时钟。
- 步骤 3.** 配置 LPUART_SCON.SCLKSEL，选择 LPUART 的传输时钟来源。
- 步骤 4.** 配置 LPUART_SCON.HDSEL=0b1，使能单线模式。
- 步骤 5.** 加入至少 500μs 的延时，初始化 TXD 为接收状态。
- 步骤 6.** 配置 LPUART_BSEL 和 LPUART_MODU 寄存器，使能 9600/4800 波特率调制。
- 步骤 7.** 后续步骤参考“[接收数据示例](#)”章节。

其他情形，例如在 9600/4800 波特率下使用单线半双工模式发送数据，或采用配置 LPUART_SCNT 的方式进行波特率调制时，均参考“[发送数据示例](#)”或“[接收数据示例](#)”章节中的步骤正常配置即可。

17.3.4 波特率生成

Mode0~Mode3 生成波特率的公式如下所示：

- Mode0 波特率生成公式：
$$\text{BaudRate} = \frac{f_{\text{SCLK}}}{12}$$
- Mode1 波特率生成公式：
$$\text{BaudRate} = \frac{f_{\text{SCLK}}}{\text{OVER} \times \text{SCNT}}$$
- Mode2 波特率生成公式：
$$\text{BaudRate} = \frac{f_{\text{SCLK}}}{\text{OVER}}$$
- Mode3 波特率生成公式：
$$\text{BaudRate} = \frac{f_{\text{SCLK}}}{\text{OVER} \times \text{SCNT}}$$

**说明**

- f_{SCLK} 为 SCLK 的时钟频率。
- OVER 取值：
 - ▶ LPUART_SCON.OVER=0b00 时，Mode1/3 模式 OVER 取值为 16；Mode2 模式 OVER 取值为 32。
 - ▶ LPUART_SCON.OVER=0b01 时，Mode1/3 模式 OVER 取值为 8；Mode2 模式 OVER 取值为 16。
 - ▶ LPUART_SCON.OVER=0b10 时，Mode1/3 模式 OVER 取值为 4；Mode2 模式 OVER 取值为 8。
- SCNT 的定义详见寄存器 LPUART_SCNT 介绍。

Mode1/Mode3 波特率设置示例

表 17-3 SCLK=4MHz 波特率计算表

波特率	SCLK=4MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%	104	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%	52	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%	26	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%	13	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%	7	35714.29	-6.99%
57600	17	58823.53	2.12%	9	55555.56	-3.55%	4	62500.00	8.51%
76800	13	76923.08	0.16%	7	71428.57	-6.99%	3	83333.33	8.51%
115200	9	111111.11	-3.55%	4	125000.00	8.51%	2	125000.00	8.51%
128000	8	125000.00	-2.34%	4	125000.00	-2.34%	2	125000.00	-2.34%
256000	4	250000.00	-2.34%	2	250000.00	-2.34%	1	250000.00	-2.34%
1000000	1	1000000.00	0.00%	1	/	/	0	/	/

表 17-4 SCLK=8MHz 波特率计算表

波特率	SCLK=8MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%	208	2403.85	0.16%
4800	417	4796.16	-0.08%	208	4807.69	0.16%	104	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%	52	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%	26	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%	13	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%	9	55555.56	-3.55%
76800	26	76923.08	0.16%	13	76923.08	0.16%	7	71428.57	-6.99%
115200	17	117647.06	2.12%	9	111111.11	-3.55%	4	125000.00	8.51%
128000	16	125000.00	-2.34%	8	125000.00	-2.34%	4	125000.00	-2.34%
256000	8	250000.00	-2.34%	4	250000.00	-2.34%	2	250000.00	-2.34%
1000000	2	1000000.00	0.00%	1	1000000.00	0.00%	1	/	/
2000000	1	2000000.00	0.00%	1	/	/	0	/	/

表 17-5 SCLK=16MHz 波特率计算表

波特率	SCLK=16MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%	417	2398.08	-0.08%
4800	833	4801.92	0.04%	417	4796.16	-0.08%	208	4807.69	0.16%
9600	417	9592.33	-0.08%	208	9615.38	0.16%	104	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%	52	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%	26	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%	17	58823.53	2.12%
76800	52	76923.08	0.16%	26	76923.08	0.16%	13	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%	9	111111.11	-3.55%
128000	31	129032.26	0.81%	16	125000.00	-2.34%	8	125000.00	-2.34%

波特率	SCLK=16MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%	4	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%	1	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%	1	/	/
4000000	1	4000000.00	0.00%	1	/	/	0	/	/

表 17-6 SCLK=24MHz 波特率计算表

波特率	SCLK=24MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%	625	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%	313	4792.33	-0.16%
9600	625	9600.00	0.00%	313	9584.66	-0.16%	156	9615.38	0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%	78	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%	39	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%	26	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%	20	75000.00	-2.34%
115200	52	115384.62	0.16%	26	115384.62	0.16%	13	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%	12	125000.00	-2.34%
256000	23	260869.57	1.90%	12	250000.00	-2.34%	6	250000.00	-2.34%
1000000	6	1000000.00	0.00%	3	1000000.00	0.00%	2	/	/
2000000	3	2000000.00	0.00%	2	/	/	1	/	/
6000000	1	6000000.00	0.00%	1	/	/	0	/	/

表 17-7 SCLK=32MHz 波特率计算表

波特率	SCLK=32MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	3333	2400.24	0.01%	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	1667	4799.04	-0.02%	833	4801.92	0.04%	417	4796.16	-0.08%
9600	833	9603.84	0.04%	417	9592.33	-0.08%	208	9615.38	0.16%
19200	417	19184.65	-0.08%	208	19230.77	0.16%	104	19230.77	0.16%
38400	208	38461.54	0.16%	104	38461.54	0.16%	52	38461.54	0.16%
57600	139	57553.96	-0.08%	69	57971.01	0.64%	35	57142.86	-0.79%
76800	104	76923.08	0.16%	52	76923.08	0.16%	26	76923.08	0.16%
115200	69	115942.03	0.64%	35	114285.71	-0.79%	17	117647.06	2.12%
128000	63	126984.13	-0.79%	31	129032.26	0.81%	16	125000.00	-2.34%
256000	31	258064.52	0.81%	16	250000.00	-2.34%	8	250000.00	-2.34%
1000000	8	1000000.00	0.00%	4	1000000.00	0.00%	2	1000000.00	0.00%
2000000	4	2000000.00	0.00%	2	2000000.00	0.00%	1	2000000.00	0.00%
4000000	2	4000000.00	0.00%	1	4000000.00	0.00%	1	/	/
8000000	1	8000000.00	0.00%	1	/	/	0	/	/

表 17-8 SCLK=48MHz 波特率计算表

波特率	SCLK=48MHz								
	OVER=4			OVER=8			OVER=16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	5000	2400.00	0.00%	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	2500	4800.00	0.00%	1250	4800.00	0.00%	625	4800.00	0.00%
9600	1250	9600.00	0.00%	625	9600.00	0.00%	313	9584.66	-0.16%
19200	625	19200.00	0.00%	313	19169.33	-0.16%	156	19230.77	0.16%
38400	313	38338.66	-0.16%	156	38461.54	0.16%	78	38461.54	0.16%
57600	208	57692.31	0.16%	104	57692.31	0.16%	52	57692.31	0.16%
76800	156	76923.08	0.16%	78	76923.08	0.16%	39	76923.08	0.16%
115200	104	115384.62	0.16%	52	115384.62	0.16%	26	115384.62	0.16%
128000	94	127659.57	-0.27%	47	127659.57	-0.27%	23	130434.78	1.90%
256000	47	255319.15	-0.27%	23	260869.57	1.90%	12	250000.00	-2.34%
1000000	12	1000000.00	0.00%	6	1000000.00	0.00%	3	1000000.00	0.00%
2000000	6	2000000.00	0.00%	3	2000000.00	0.00%	2	1500000.00	-25.00%
4000000	4	3000000.00	0.00%	2	3000000.00	0.00%	1	3000000.00	0.00%
6000000	3	4000000.00	0.00%	2	/	/	1	/	/
12000000	1	12000000.00	0.00%	1	/	/	0	/	/

17.3.5 帧错误检测

当 LPUART 工作在 Mode1/2/3 时，具有帧错误检测功能：

- 在接收数据时，若未在预期时间内识别出停止位，即为帧错误。
- 检测到帧错误时，LPUART_ISR.FE 会被硬件置 1，用户程序应及时清除该标志。

17.3.6 多机通讯

当工作在 Mode2/3 时，将 LPUART_SCON.ADRDET 设置为 1，可开启多机通讯地址自动识别功能。主机可以发送地址帧及数据帧，从机可以在收到匹配的地址帧后才接收数据帧。

主机发送地址帧及数据帧

当 LPUART_SBUF.DATA8 为 1 则发送地址帧，LPUART_SBUF.DATA 为目标从机的地址。

当 LPUART_SBUF.DATA8 为 0 则发送数据帧，LPUART_SBUF.DATA 为待发送的数据。

从机接收地址帧及数据帧

当 LPUART_SCON.ADRDET 为 1 时，从机只接收与自己地址相匹配的地址帧，忽略数据帧及不匹配的地址帧。当从机从 RXD 接收到地址帧时，硬件自动对比地址帧内的地址与本机地址，只有两者匹配时才会将收到的地址存入 LPUART_SBUF 寄存器并置位 LPUART_ISR.RC。

当 LPUART_SCON.ADRDET 为 0 时，从机只接收数据帧并忽略地址帧。用户程序应当在地址匹配后才设置 LPUART_SCON.ADRDET 为 0 以接收主机后继发送的数据帧。

从机地址配置

从机地址由广播地址和用户配置地址组成：

- 广播地址：LPUART_SADDR | LPUART_SADEN
- 用户配置地址：LPUART_SADDR & LPUART_SADEN

收到地址帧时，检测到从总线收到的地址与 LPUART_SADEN 的值相与后与用户配置地址相同，或收到的地址与广播地址相与后与广播地址相同，则认为收到了与从机地址相匹配的地址帧。

例：LPUART_SADDR=0x55，LPUART_SADEN=0xF0，则从机可以匹配的地址为 0xF5、0xF7、0xFD、0xFF（广播地址）和 0x50~0x5F（用户配置地址）。

17.3.7 硬件流控

通过增加 CTS 和 RTS 信号可以实现硬件流控的功能，LPUART 模块会根据 CTS 和 RTS 的高低电平自动控制数据的收发，而无需通过软件来判断。

两个 LPUART 模块之间的硬件流控示意图如下所示：

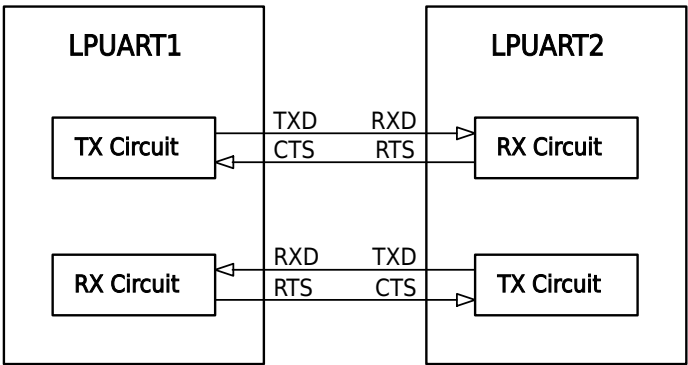


图 17-8 LPUART 硬件流控

RTS 流控

当 RTS 流控使能时（LPUART_SCON.RTSEN 为 1），RTS 管脚指示本模块的接收状态：

- 接收缓存空时，RTS 管脚变为低电平，通知发送端发送下一帧数据。
- 接收缓存满时，RTS 管脚变为高电平，通知发送端暂停发送下一帧数据。

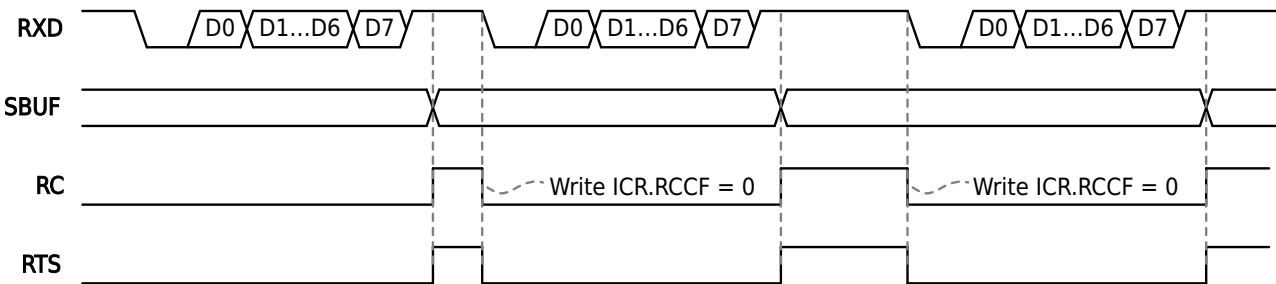


图 17-9 RTS 硬件流控信号

CTS 流控

当 CTS 流控使能时（LPUART_SCON.CTSEN 设置为 1），CTS 管脚指示本模块下一帧数据的发送：

- 若 CTS 低电平，LPUART 发送下一帧数据。
- 若 CTS 高电平，LPUART 发送完当前帧后，等待 CTS 变成低电平再进行下一帧的发送。

当 CTS 电平发生变化，LPUART_ISR.CTSIF 自动置 1，读 LPUART_ISR.CTS 获取当前 CTS 管脚的电平值。

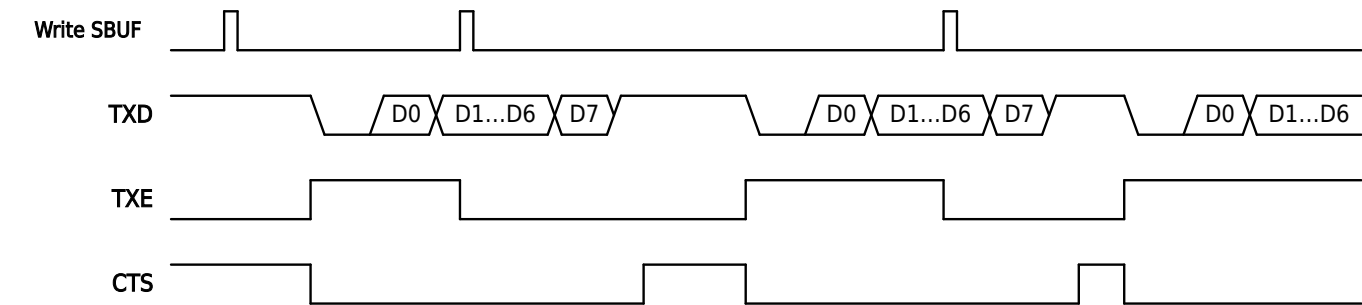


图 17-10 CTS 硬件流控信号

17.3.8 DMA 传输

- LPUART 模块支持 DMA 的握手协议，可以通过 DMA 进行数据传输。
- 当设置 LPUART_SCON.DMATXEN 为 1 时，可以用 DMA 控制 LPUART 对外发送数据。
每当 LPUART_ISR.TXE 为 1 时，DMA 会向 LPUART_SBUF 写入一帧待发送的数据，直到 DMA 中所配置的数据全部搬运完毕。当开始使用此功能时，请先配置 DMA 的各个寄存器及使能 DMA，然后再配置 LPUART 的各个寄存器。当要结束此功能时，请先关闭 LPUART_SCON.DMATXEN，再关闭 DMA（如 DMA 的关闭使用的是 DMA 硬件自清，那么一旦查询到 DMA 的使能已经硬件清 0，就立刻关闭 LPUART_SCON.DMATXEN）。
 - 当设置 LPUART_SCON.DMARXEN 为 1 时，可以用 DMA 读取并保存 LPUART 收到的数据。
每当 LPUART_ISR.RC 为 1 时，DMA 会从 LPUART_SBUF 读出一帧接收到的数据，直到 DMA 中所配置的数据全部搬运完毕。

17.3.9 收发端缓存

接收缓存

LPUART 模块接收端有一个帧（8bits/9bits）的接收缓存，即保持接受的数据帧直至下一帧数据的 Stop 位被接收才更新该数据帧。

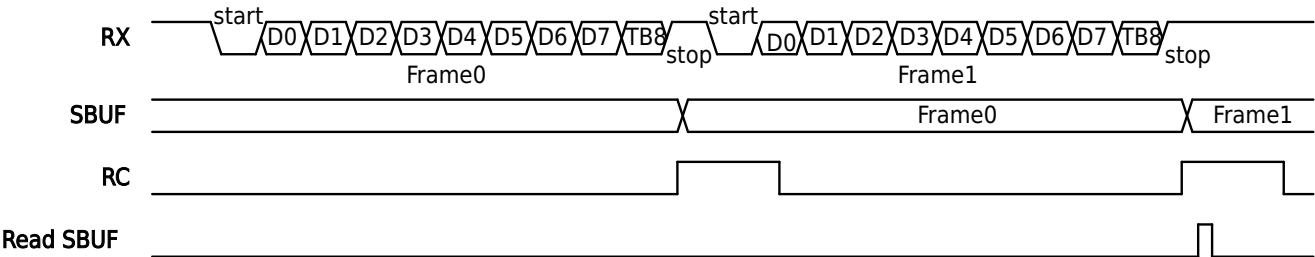


图 17-11 接收缓存

发送缓存

- LPUART 模块发送端有一个帧（8bits/9bits）的发送缓存，当 LPUART 在发送当前帧时，下一个发送数据软件写入 LPUART_SBUF。
- 当 LPUART_ISR.TXE=0 时，表明当前发送缓存满，LPUART_SBUF 不能写入下一个发送数据。
 - 当 LPUART_ISR.TXE=1 时，表明当前发送缓存空，LPUART_SBUF 可以写入下一个发送数据，在完成当前数据传输后，硬件自动把发送缓存中的数据装载入移位寄存器中发送出去。

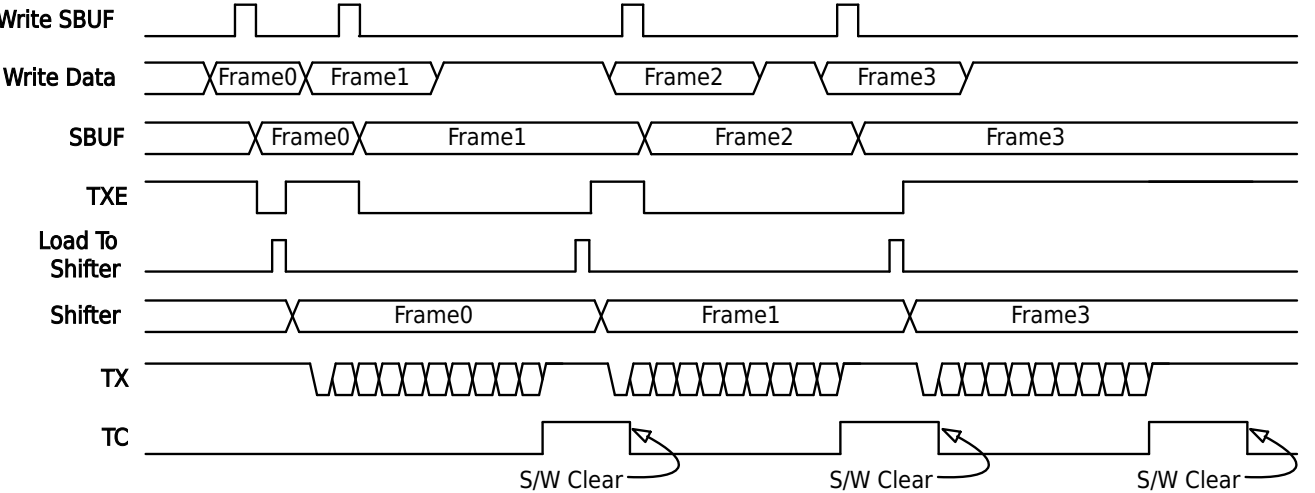


图 17-12 发送缓存

17.4 操作示例

17.4.1 发送数据示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，将 TXD 映射到需要的管脚；并配置 TXD 为 CMOS 输出。
- 步骤 2. 配置 SYSCTRL 模块寄存器 PERI_CLKEN0.LPUARTx 为 1，使能 LPUART 配置时钟及工作时钟。
- 步骤 3. 配置 LPUART_SCON.SCLKSEL，选择 LPUART 的传输时钟来源。
- 步骤 4. 配置 LPUART_SCON.SM，选择 LPUART 的工作模式。
- 步骤 5. 配置 LPUART_SCON.OVER 及 LPUART_SCNT 寄存器，配置通信波特率。
- 步骤 6. 配置 LPUART_SCON.STOPBIT、LPUART_SCON.B8CONT，配置 LPUART 的 STOP 长度，奇偶校验方式。
- 步骤 7. 向 ICR 寄存器写入 0x00，清除所有状态标志。
- 步骤 8. 当查询到 ISR.TXE 为 1，向 LPUART_SBUF 寄存器写入一帧待发送的数据。
- 步骤 9. 查询等待 ISR.TC 变为 1，本次写入的数据帧发送完成。
- 步骤 10. 向 ICR.TCCF 写入 0x00，清除发送完成标志。
- 步骤 11. 如待发送的数据未完成，则跳转到步骤 8。

17.4.2 接收数据示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，将 RXD 映射到需要的管脚；并配置 RXD 为输入并使能上拉电阻。
- 步骤 2. 配置 SYSCTRL 模块寄存器 PERI_CLKEN0.LPUARTx 为 1，使能 LPUART 配置时钟及工作时钟。
- 步骤 3. 配置 LPUART_SCON.SCLKSEL，选择 LPUART 的传输时钟来源。
- 步骤 4. 配置 LPUART_SCON.SM，选择 LPUART 的工作模式。
- 步骤 5. 配置 LPUART_SCON.OVER 及 LPUART_SCNT 寄存器，配置通信波特率。
- 步骤 6. 配置 LPUART_SCON.STOPBIT、LPUART_SCON.B8CONT，配置 LPUART 的 STOP 长度，奇偶校验方式。
- 步骤 7. 向 ICR 寄存器写入 0x00，清除所有状态标志。
- 步骤 8. 设置 LPUART_SCON.REN 为 1，使能 RXD 数据接收。
- 步骤 9. 查询等待 ISR.RC 变为 1，接收完成一帧数据。
- 步骤 10. 查询 ISR.PE 和 ISR.FE，以确认接收到的数据帧是否有效。若无效则进行出错处理；若有效则从 LPUART_SBUF 寄存器读出数据并保存。
- 步骤 11. 向 ICR.RCCF 写入 0x00，清除接收完成标志。
- 步骤 12. 如待接收的数据未完成，则跳转到步骤 9。

17.5 寄存器

17.5.1 寄存器总表

- LPUART0 基地址：0x40000000
- LPUART1 基地址：0x40008000
- LPUART2 基地址：0x4000B400

表 17-9 低功耗同步异步收发器（LPUART）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	LPUART_SBUF	数据寄存器	32
0x04	LPUART_SCON	控制寄存器	32
0x08	LPUART_SADDR	地址寄存器	32
0x0C	LPUART_SADEN	地址掩码寄存器	32
0x10	LPUART_ISR	中断标志位寄存器	32
0x14	LPUART_ICR	中断标志位清除寄存器	32
0x18	LPUART_SCNT	波特率寄存器	32
0x1C	LPUART_BSEL	波特率产生寄存器	32
0x20	LPUART_MODU	9600/4800 波特率调制器寄存器	32

17.5.2 数据寄存器 (LPUART_SBUF)



说明

SBUF 寄存器内有数据时，禁止修改发送数据功能相关控制位。

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																							DATA8		DATA						
Access																								RW								

位/位域	标记	功能描述	读写
31:9	Reserved	-	-
8	DATA8	Bit8 数据位 - 在 Mode0/1 下，读取该位为 0，写入该位无效 - 在 Mode2/3 下，分以下情况： - 当硬件奇偶校验位开启时，接收时该位为接收数据奇偶校验位，校验由硬件进行，如校验出错，校验错误标志位 PE 置 1；发送时该位无效，发送数据奇偶校验位由硬件计算并发送。 - 当硬件奇偶校验位关闭时，接收时该位为接收数据 Bit8；发送时该位为发送数据 Bit8。 说明 当开启多机通讯模式，接收数据奇偶校验自动关闭；发送数据奇偶校验仍受 B8CONT 控制。	RW
7:0	DATA	数据 发送数据时，将待发送数据写入该寄存器；接收数据时，从该寄存器读出已收到的数据	RW

17.5.3 控制寄存器 (LPUART_SCON)

Offset	Bit Position																																								
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
Reset	0x00000000																																								
Name	Reserved										HDSEL	FEIE	CTSIE	CTSEN	RTSEN	DMATXEN	DMARXEN	STOPBIT	PEIE	SCLKSEL	OVER	TXEIE	SM	ADDRDET	REN	B8CONT	TCIE	RCIE													
Access											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

位/位域	标记	功能描述	读写
31:23	Reserved	-	-
22	HDSEL	单线半双工功能使能 ● 0b0: 禁止 ● 0b1: 使能	RW
21	FEIE	帧错误中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
20	CTSIE	CTS 信号翻转中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
19	CTSEN	CTS 信号使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
18	RTSEN	RTS 信号使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
17	DMATXEN	DMA 发送数据使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
16	DMARXEN	DMA 接收数据使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
15:14	STOPBIT	STOPBIT 位持续时间配置 ● 0b00: 1-bit ● 0b01: 1.5-bit ● 0b10: 2-bit ● 0b11: 禁止设置  注意 Mode0 时必须将该位设置为 0b00。	RW
13	PEIE	奇偶校验错误中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW

位/位域	标记	功能描述	读写
12:11	SCLKSEL	传输时钟 SCLK 来源选择 ● 0b00: PCLK ● 0b01: PCLK ● 0b10: XTL ● 0b11: RCL	RW
10:9	OVER	过采样控制 ● Mode0: 无效 ● Mode1/3: ▶ 0b00: 16 采样分频 ▶ 0b01: 8 采样分频 ▶ 0b10: 4 采样分频 ▶ 0b11: 禁止设置 ● Mode2: ▶ 0b00: 32 采样分频 ▶ 0b01: 16 采样分频 ▶ 0b10: 8 采样分频 ▶ 0b11: 禁止设置	RW
8	TXEIE	TX Buf 空中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
7:6	SM	工作模式 ● 0b00: Mode0, 同步半双工, Data(8bits) ● 0b01: Mode1, 异步全双工, Start(1bit) + Data(8bits) + Stop(1~2bits) ● 0b10: Mode2, 异步全双工, Start(1bit) + Data(8bits) + B8(1bit) + Stop(1~2bits) ● 0b11: Mode3, 异步全双工, Start(1bit) + Data(8bits) + B8(1bit) + Stop(1~2bits)	RW
5	ADRDET	多机通讯地址自动识别使能 ● 0b0: 禁止 ● 0b1: 使能	RW
4	REN	接收使能 ● Mode0: ▶ 0b0: 发送 ▶ 0b1: 接收 ● Mode1/2/3: ▶ 0b0: 发送 ▶ 0b1: 接收/发送  说明 该位为 1 时, 禁止修改接收数据相关控制位。	RW
3:2	B8CONT	DATA8 数据位功能配置 ● 0b00: 由软件读写 LPUART_SBUF.DATA8 来决定 ● 0b01: 硬件偶校验 ● 0b10: 硬件奇校验 ● 0b11: 禁止设置	RW

位/位域	标记	功能描述	读写
1	TCIE	发送完成中断使能 ● 0b0：禁止 ● 0b1：使能	RW
0	RCIE	接收完成中断使能 ● 0b0：禁止 ● 0b1：使能	RW

17.5.4 地址寄存器 (LPUART_SADDR)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								SADDR							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	SADDR	从机设备地址寄存器	RW

17.5.5 地址掩码寄存器 (LPUART_SADEN)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								SADEN							
Access																									RW							

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	SADEN	从机设备地址掩码寄存器 每个 Bit 对应一个地址位： ● 0b0：不参与地址匹配 ● 0b1：参与地址匹配	RW

17.5.6 中断标志位寄存器 (LPUART_ISR)

Offset	Bit Position																																									
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Reset	0x00000008																																									
Name	Reserved																																									
Access																											RO	CTS	RO	CTSIF	RO	PE	RO	TXE	RO	FE	RO	TC	RO	RC		

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	CTS	CTS 信号标志 只能通过硬件清零 ● 0b0: CTS 信号为低电平 ● 0b1: CTS 信号为高电平	RO
5	CTSIF	CTS 信号翻转标志 ● 0b0: CTS 信号没有发生翻转 ● 0b1: CTS 信号发生翻转	RO
4	PE	奇偶校验错误标志 ● 0b0: 无奇偶校验错误 ● 0b1: 奇偶校验错误	RO
3	TXE	Tx Buffer 空标志 只能通过硬件清零 ● 0b0: Tx Buffer 非空 ● 0b1: Tx Buffer 空	RO
2	FE	帧错误标志 ● 0b0: 未发生帧错误 ● 0b1: 已发生帧错误	RO
1	TC	发送完成标志 ● 0b0: 发送未完成 ● 0b1: 发送完成	RO
0	RC	接收完成标志 ● 0b0: 接收未完成 ● 0b1: 接收完成	RO

17.5.7 中断标志位清除寄存器 (LPUART_ICR)

[illegible]

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5	CTSIFCF	ISR.CTSIF 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
4	PECF	ISR.PE 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
3	Reserved	-	-
2	FECF	ISR.FE 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
1	TCCF	ISR.TC 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
0	RCCF	ISR.RC 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO

17.5.8 波特率寄存器 (LPUART SCNT)

[illegible]

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	SCNT	波特率计数器	RW

17.5.9 波特率产生寄存器 (LPUART_BSEL)

Offset	Bit Position																															
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															BSEL
Access																																RW

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1:0	BSEL	波特率产生方式 0b00-0b01：用 OVER 和 SCNT 产生波特率（即和之前一样） 0b10：当 SCLK 选择 XTL 时（LPUART_SCON.SCLKSEL=0b10），用调制方式产生 9600 波特率 0b11：当 SCLK 选择 XTL 时（LPUART_SCON.SCLKSEL=0b10），用调制方式产生 4800 波特率	RW

17.5.10 9600/4800 波特率调制器寄存器 (LPUART_MODU)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000054A																															
Name	Reserved																MODU															
Access																	RW															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	MODU	9600/4800 波特率调制器 当 SCLK 选择 XTL 时 (LPUART_SCON.SCLKSEL=0b10)，并且 BSEL.BSEL=0b10/0b11 时，该寄存器有效： ● BSEL.BSEL=0b10 时，产生 9600 波特率所需的调制器设置，需配此寄存器的值为 0x54A。 ● BSEL.BSEL=0b11 时，产生 4800 波特率所需的调制器设置，需配此寄存器的值为 0xEFB。	RW

产生 9600 波特率原理

当 SCLK 选择 XTL 时 (LPUART_SCON.SCLKSEL=0b10)，SCLK 频率为 32768，LPUART 波特率为 9600 (BSEL.BSEL=0b10)，每一位需要 $32768/9600=3.413$ 个 SCLK，如果每一位 3 个 SCLK 则每位的误差为 -0.413 个 SCLK，如果每一位 4 个 SCLK 则每位的误差为 +0.587 个 SCLK。

Bit	Name	Number of SCLK	LPUART_MODU	Final inaccuracy (number of SCLK)
0	Start	3	0	-0.413
1	DATA[0]	4	1	+0.174(=-0.413+0.587)
2	DATA[1]	3	0	-0.239(=+0.174-0.413)
3	DATA[2]	4	1	+0.348(=-0.239+0.587)
4	DATA[3]	3	0	-0.065(=+0.348-0.413)
5	DATA[4]	3	0	-0.478(=-0.065-0.413)
6	DATA[5]	4	1	+0.109(=-0.478+0.587)
7	DATA[6]	3	0	-0.304(=+0.109-0.413)
8	DATA[7]	4	1	+0.283(=-0.304+0.587)
9	Parity/stop1	3	0	-0.130(=+0.283-0.413)
10	Stop1/stop2	4	1	+0.457(=-0.13+0.587)
11	Stop2	3	0	+0.044(=+0.457-0.413)

产生 4800 波特率原理

当 SCLK 选择 XTL 时 (LPUART_SCON.SCLKSEL=0b10)，SCLK 频率为 32768，LPUART 波特率为 4800 (BSEL.BSEL=0b11)，每一位需要 $32768/4800=6.827$ 个 SCLK，如果每一位 6 个 SCLK 则每位的误差为 -0.827 个 SCLK，如果每一位 7 个 SCLK 则每位的误差为 +0.173 个 SCLK。

Bit	Name	Number of SCLK	LPUART_MODU	Final inaccuracy (number of SCLK)
0	Start	7	1	+0.173

Bit	Name	Number of SCLK	LPUART_MODU	Final inaccuracy (number of SCLK)
1	DATA[0]	7	1	+0.346(=+0.173+0.173)
2	DATA[1]	6	0	-0.481(=+0.346-0.827)
3	DATA[2]	7	1	-0.308(=-0.481+0.173)
4	DATA[3]	7	1	-0.135(=-0.308+0.173)
5	DATA[4]	7	1	+0.038(=-0.135+0.173)
6	DATA[5]	7	1	+0.211(=+0.038+0.173)
7	DATA[6]	7	1	+0.384(=+0.211+0.173)
8	DATA[7]	6	0	-0.443(=+0.384-0.827)
9	Parity/stop1	7	1	-0.270(=-0.443+0.173)
10	Stop1/stop2	7	1	-0.097(=-0.270+0.173)
11	Stop2	7	1	+0.076(=-0.097+0.173)

18 通用同步异步收发器 (USART)

18.1 简介

本产品搭载 2 个通用串行同步异步收发器模块 (USART)，能够灵活地与外部设备进行数据交换。该模块支持通用异步串口通信、通用同步串口通信、智能卡协议、红外通信协议和 LIN 协议。

18.2 主要特性

- 两种波特率生成方式：整数分频、小数分频
- 两种数据序列顺序：MSB 先传、LSB 先传
- 内建多功能定时计数器，支持接收超时等功能
- 支持 TXD、RXD 引脚交换
- 支持 TXD、RXD 电平反向
- 支持 DMA 连续通讯

18.2.1 异步串口特性

- 异步传输数据
- 两种帧长度：8 比特、9 比特
- 帧校验：奇校验、偶校验、无校验
- 两种停止位：1 比特、2 比特
- 支持硬件流控 (CTS/RTS)
- 支持单线半双工通信
- 支持波特率自动检测
- 支持多个处理器间通信

18.2.2 同步串口特性

- 同步传输数据
- 数据长度固定为 8 位
- 支持硬件流控 (CTS/RTS)
- 两种通信时钟源：内部时钟源、外部时钟源

18.2.3 智能卡接口特性

- 通信协议：ISO/IEC7816-3
- 8 种 ETU 长度：16~512
- 支持硬件出错检测与重传
- 支持主机接口

18.2.4 红外通信特性

- 支持 3/16 无载波模式
- 支持 38K 载波调制模式，占空比可调，调制电平可配置

18.2.5 LIN 特性

- 支持同步间隔段发送和检测
- 支持同步段发送和波特率自动检测

18.3 功能说明

本章将对异步串口模式、同步串口模式、智能卡模式、红外模式、LIN 模式进行详细说明。
异步串口、同步串口、智能卡、红外和 LIN 功能一次只能启动其中一个功能。

 **说明**

本模块支持输出信号反相控制 (USART_CR2.TXINV) 和输入信号反相控制 (USART_CR2.RXINV) , 如无特殊说明, 本章中涉及的图、表和文字均为输出反相控制和输入反相控制关闭的情况。

18.3.1 USART 通用功能

18.3.1.1 通信波特率编程

 **说明**

通信波特率编程适用于异步串口模式、同步串口模式、智能卡模式、红外模式和 LIN 模式。

本模块支持两种波特率生成方式：整数波特率生成方式、小数波特率生成方式。
两种波特率生成方式的公式如下所示：

$$\text{Baud}_{\text{整数}} = \frac{\text{PCLK}}{4^{\text{PSC}} \times \text{采样计数} \times (\text{DIVI} + 1)}$$
$$\text{Baud}_{\text{小数}} = \frac{\text{PCLK} \times (128 + \text{DIVF})}{4^{\text{PSC}} \times \text{采样计数} \times (\text{DIVI} + 1) \times 256}$$

表 18-1 各模式采样计数和最高波特率

功能模式	采样计数	最高波特率 (单位: bps)
异步串口	采样计数=8×(2-OVER8)	PCLK/8
同步串口	采样计数=4	内部时钟源时: PCLK/8 外部时钟源时: PCLK/6 (外部输入时钟的最大频率要求为 PCLK/6)
智能卡接口	采样计数=2×BCN	PCLK/32
红外通信	采样计数=8×2	由红外调制模式决定
LIN	采样计数=8×(2-OVER8)	20k

其中：

- Baud：波特率
- PCLK：时钟源
- PSC：USART_BRR.PSC 设定值
- OVER8：USART_CR1.OVER8 设定值
- BCN：USART_CR3.BCN 设定值
- DIVI：USART_BRR.DIVI 设定值
- DIVF：USART_BRR.DIVF 设定值

已知通信波特率，计算 USART_BRR 寄存器值的步骤如下所示：

步骤 1. 配置 USART_BRR.PSC，使下述表达式的计算结果小于等于 256。

$$\frac{\text{PCLK}}{4^{\text{PSC}} \times \text{Baud} \times \text{采样计数}}$$

步骤 2. 根据下方公式计算 USART_BRR.DIVI 的值，其中的 INT 函数代表向下取整。

$$DIVI = INT\left(\frac{PCLK}{4^{PSC} \times Baud \times \text{采样计数}} - 1\right)$$

步骤 3. 根据下方公式计算 USART_BRR.DIVF 的值，其中的 INT 函数代表向下取整。

$$DIVF = INT\left(\frac{Baud \times \text{采样计数} \times (DIVI + 1) \times 256 \times 4^{PSC} \times 1.0}{PCLK} - 128 + 0.5\right)$$

步骤 4. 当 DIVF 小于 127.5 时，设置 USART_BRR.FBME 为 1，使能小数波特率发生器；
当 DIVF 大于 127.5 时，设置 USART_BRR.FBME 为 0，禁止小数波特率发生器。

异步串口通信波特率编程样例

表 18-2 PCLK 为 4MHz 的波特率编程示例

目标波特率	PCLK=4MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	0	207	NA	2404	0.16%	0	103	NA	2404	0.16%
4800	0	103	NA	4808	0.16%	0	51	NA	4808	0.16%
9600	0	51	NA	9615	0.16%	0	25	NA	9615	0.16%
19200	0	25	NA	19231	0.16%	0	12	NA	19231	0.16%
76800	0	5	108	76823	0.03%	0	2	108	76823	0.03%
115200	0	3	108	115234	0.03%	0	1	108	115234	0.03%
230400	0	1	108	230469	0.03%	0	0	108	230469	0.03%
250000	0	1	NA	250000	0.00%	0	0	NA	250000	0.00%
500000	0	0	NA	500000	0.00%					

表 18-3 PCLK 为 8MHz 的波特率编程示例

目标波特率	PCLK=8MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	1	103	NA	2404	0.16%	0	207	NA	2404	0.16%
4800	0	207	NA	4808	0.16%	0	103	NA	4808	0.16%
9600	0	103	NA	9615	0.16%	0	51	NA	9615	0.16%
19200	0	51	NA	19231	0.16%	0	25	NA	19231	0.16%
76800	0	12	NA	76923	0.16%	0	5	108	76823	0.03%
115200	0	7	108	115234	0.03%	0	3	108	115234	0.03%
230400	0	3	108	230469	0.03%	0	1	108	230469	0.03%
250000	0	3	NA	250000	0.00%	0	1	NA	250000	0.00%
500000	0	1	NA	500000	0.00%	0	0	NA	500000	0.00%
750000	0	0	64	750000	0.00%					
1000000	0	0	NA	1000000	0.00%					

表 18-4 PCLK 为 16MHz 的波特率编程示例

目标波特率	PCLK=16MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	2	51	NA	2404	0.16%	1	103	NA	2404	0.16%
4800	1	103	NA	4808	0.16%	0	207	NA	4808	0.16%
9600	0	207	NA	9615	0.16%	0	103	NA	9615	0.16%
19200	0	103	NA	19231	0.16%	0	51	NA	19231	0.16%
76800	0	25	NA	76923	0.16%	0	12	NA	76923	0.16%

目标波特率	PCLK=16MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
115200	0	16	123	115349	0.13%	0	7	108	115234	0.03%
230400	0	7	108	230469	0.03%	0	3	108	230469	0.03%
250000	0	7	NA	250000	0.00%	0	3	NA	250000	0.00%
500000	0	3	NA	500000	0.00%	0	1	NA	500000	0.00%
750000	0	1	64	750000	0.00%	0	0	64	750000	0.00%
1000000	0	1	NA	1000000	0.00%	0	0	NA	1000000	0.00%
1500000	0	0	64	1500000	0.00%					
2000000	0	0	NA	2000000	0.00%					

表 18-5 PCLK 为 24MHz 的波特率编程示例

目标波特率	PCLK=24MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	2	77	NA	2404	0.16%	2	38	NA	2404	0.16%
4800	2	38	NA	4808	0.16%	1	77	NA	4808	0.16%
9600	1	77	NA	9615	0.16%	0	155	NA	9615	0.16%
19200	0	155	NA	19231	0.16%	0	77	NA	19231	0.16%
76800	0	38	NA	76923	0.16%	0	18	121	76789	-0.01%
115200	0	25	NA	115385	0.16%	0	12	NA	115385	0.16%
230400	0	12	NA	230769	0.16%	0	5	108	230469	0.03%
250000	0	11	NA	250000	0.00%	0	5	NA	250000	0.00%
500000	0	5	NA	500000	0.00%	0	2	NA	500000	0.00%
750000	0	3	NA	750000	0.00%	0	1	NA	750000	0.00%
1000000	0	2	NA	1000000	0.00%	0	0	43	1001953	0.20%
1500000	0	1	NA	1500000	0.00%	0	0	NA	1500000	0.00%
2000000	0	0	43	2003906	0.20%					
2500000	0	0	85	2496094	-0.16%					
3000000	0	0	NA	3000000	0.00%					

表 18-6 PCLK 为 32MHz 的波特率编程示例

目标波特率	PCLK=32MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	2	103	NA	2404	0.16%	2	51	NA	2404	0.16%
4800	2	51	NA	4808	0.16%	1	103	NA	4808	0.16%
9600	1	103	NA	9615	0.16%	0	207	NA	9615	0.16%
19200	0	207	NA	19231	0.16%	0	103	NA	19231	0.16%
76800	0	51	NA	76923	0.16%	0	25	NA	76923	0.16%
115200	0	33	123	115349	0.13%	0	16	123	115349	0.13%
230400	0	16	123	230699	0.13%	0	7	108	230469	0.03%
250000	0	15	NA	250000	0.00%	0	7	NA	250000	0.00%
500000	0	7	NA	500000	0.00%	0	3	NA	500000	0.00%
750000	0	4	112	750000	0.00%	0	1	64	750000	0.00%
1000000	0	3	NA	1000000	0.00%	0	1	NA	1000000	0.00%

目标波特率	PCLK=32MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
1500000	0	1	64	1500000	0.00%	0	0	64	1500000	0.00%
2000000	0	1	NA	2000000	0.00%	0	0	NA	2000000	0.00%
2500000	0	0	32	2500000	0.00%					
3000000	0	0	64	3000000	0.00%					
4000000	0	0	NA	4000000	0.00%					

表 18-7 PCLK 为 48MHz 的波特率编程示例

目标波特率	PCLK=48MHz									
	OVER8=1					OVER8=0				
	PSC	DIVI	DIVF	实际波特率	误差%	PSC	DIVI	DIVF	实际波特率	误差%
2400	2	155	NA	2404	0.16%	2	77	NA	2404	0.16%
4800	2	77	NA	4808	0.16%	2	38	NA	4808	0.16%
9600	2	38	NA	9615	0.16%	1	77	NA	9615	0.16%
19200	1	77	NA	19231	0.16%	0	155	NA	19231	0.16%
76800	0	77	NA	76923	0.16%	0	38	NA	76923	0.16%
115200	0	51	NA	115385	0.16%	0	25	NA	115385	0.16%
230400	0	25	NA	230769	0.16%	0	12	NA	230769	0.16%
250000	0	23	NA	250000	0.00%	0	11	NA	250000	0.00%
500000	0	11	NA	500000	0.00%	0	5	NA	500000	0.00%
750000	0	7	NA	750000	0.00%	0	3	NA	750000	0.00%
1000000	0	5	NA	1000000	0.00%	0	2	NA	1000000	0.00%
1500000	0	3	NA	1500000	0.00%	0	1	NA	1500000	0.00%
2000000	0	2	NA	2000000	0.00%	0	0	43	2003906	0.20%
2500000	0	1	85	2496094	-0.16%	0	0	85	2496094	-0.16%
3000000	0	1	NA	3000000	0.00%	0	0	NA	3000000	0.00%
4000000	0	0	43	4007812.5	0.20%					
6000000	0	0	NA	6000000	0.00%					

18.3.1.2 自动波特率检测

 **说明**

通信波特率检测适用于异步串口模式和 LIN 模式。

本功能需结合多功能定时计数器实现自动波特率检测。

通过设置 USART_CR1.TOE=5 和 USART_CR3.ABRE=1，USART 模块进入自动波特率检测模式。硬件检测到 RXD 上有输入数据边沿后，计数器从 0 开始计数，当收到数据为 0x55 的数据帧则可以检出通信波特率。当自动波特率检测完成后，则停止计数，硬件自动设置 USART_SR.ABR 为 1 并将波特率检出值存储于 USART_BDR 寄存器，若设置 USART_IER.ABR 为 1 则可以产生自动波特率检出中断。

通过如下公式即可计算出通信波特率：

$$\text{Baud} = \frac{f_{\text{PCLK}}}{4^{\text{PSC}} \times (\text{BDR} + 1)}$$

其中：

- f_{PCLK} ：PCLK 的频率
- PSC：USART_BRR.PSC 的值
- BDR：USART_BDR 寄存器的值

当不再需要检测波特率时，应设置 USART_CR3.ABRE 为 0 以退出自动波特率检测模式。

 说明

- 当被检测的波特率和自动波特率检测范围不匹配时，USART_BDR 的值有发生溢出的可能。
- 若自动波特率检测过程中，计数器发生溢出，硬件自动设置 USART_SR.TO 为 1，若使能相应中断则可以产生多功能定时计数器中断。

18.3.1.3 DMA 连续通信

 说明

- DMA 连续通信适用于异步串口模式、同步串口模式、智能卡模式、红外模式和 LIN 模式。
- 单线半双工工作时不支持 DMA 模式。

USART 模块支持 DMAC 的硬件握手逻辑。

- 写 USART_CR2.TXDMAEN=1，启动 USART 发送的 DMAC 硬件握手逻辑。当发送缓存为空时，USART 会向 DMAC 发出数据搬运请求。DMAC 收到该发送请求信号，则从 DMAC 源地址搬运一个帧的发送数据到 USART_TDR 中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。
- 写 USART_CR2.RXDMAEN=1，启动 USART 接收的 DMAC 硬件握手逻辑。当一帧接收完成，USART 会向 DMAC 发出数据搬运请求。DMAC 收到该接收请求信号，则从 USART_RDR 中把接收数据搬运至 DMAC 的目标地址中。上述步骤重复发生，直到 DMAC 中所配置的数据长度全部搬运完毕。

18.3.1.4 硬件流控

 说明

硬件流控适用于异步串口模式和同步串口模式。

硬件流控是指通过 CTS 和 RTS 信号控制 2 个 USART 设备之间的串行数据流，其连接方式如下图所示。硬件流控制功能分为两种类型，RTS 流控功能和 CTS 流控功能。

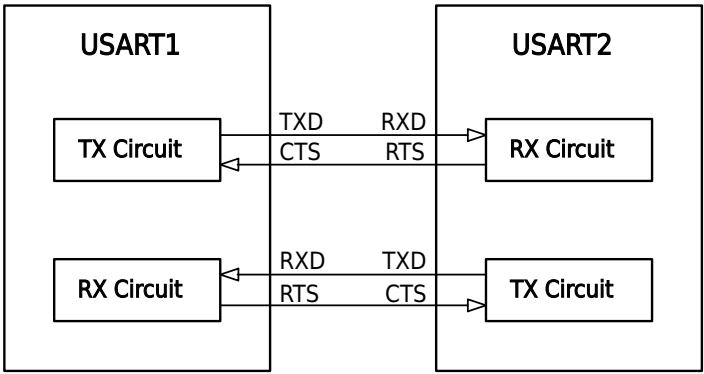


图 18-1 异步串口模式硬件流控示例

CTS 流控功能

当设置 USART_CR2.CTSEN 为 1 时，本模块具有 CTS 流控功能：

- 当 CTS 管脚输入为高电平时，TXD 管脚不会向外发送数据；
- 当 CTS 管脚输入为低电平时，TXD 管脚向外发送数据。

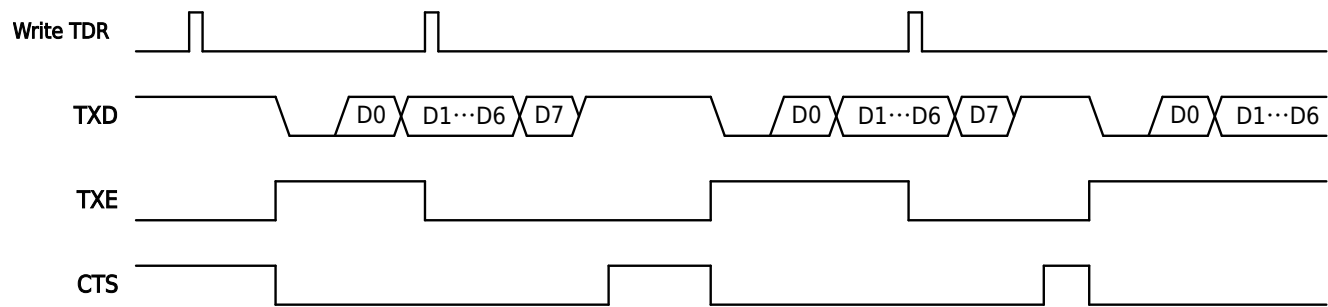


图 18-2 CTS 硬件流控信号



说明

若正在发送数据时，CTS 管脚输入信号变成高电平，则正在发送的数据帧不受影响，当前数据帧发送完成后，暂停发送新的数据。

RTS 流控功能

当设置 USART_CR2.RTSEN 为 1 时，本模块具有 RTS 流控功能：

- 当接收缓存空时，RTS 管脚输出低电平；
- 当接收缓存满时，RTS 管脚输出高电平。

异步串口模式下，只有满足如下所有条件时，RTS 管脚才会输出低电平：

1. USART_CR1.RE 为 1，已使能接收电路。
2. USART_SR.RC 为 0，USART_RDR 寄存器内数据已被读取。
3. USART_SR.PE、USART_SR.FE 和 USART_SR.OR 为 0，没有接收数据校验错误、接收帧错误和上溢错误。

如果不能同时满足以上全部条件，RTS 管脚则输出高电平。

同步串口模式下，只有满足如下所有条件时，RTS 管脚才会输出低电平：

1. 接收使能（USART_CR1.RE=1），且不正在接收数据。
2. USART_RDR 寄存器中没有未读取的数据（USART_CR1.RE=1 时）。
3. USART_TDR 更新完成（如果 USART_CR1.TE=1 时）。
4. USART_SR.OR 为 0，没有接收数据上溢错误。

如果不能同时满足以上全部条件，RTS 管脚则输出高电平。

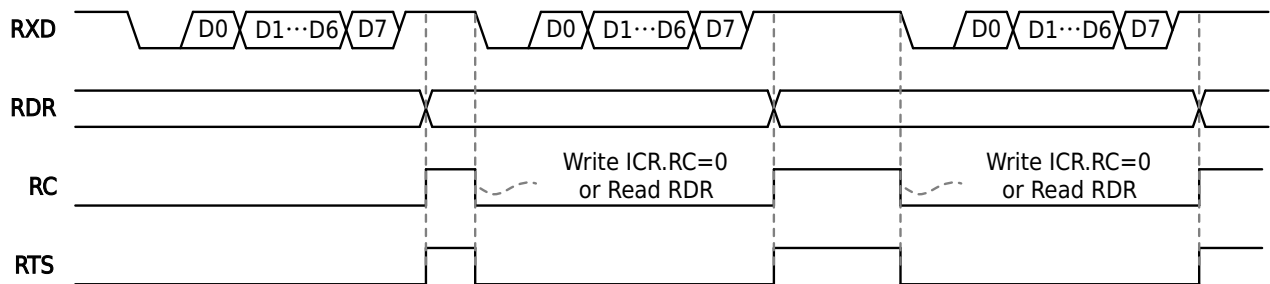


图 18-3 RTS 硬件流控信号

18.3.1.5 多功能定时计数器

本模块内嵌多功能定时计数器，可实现五种定时计数功能。其定时计数周期公式如下：

$$t_{PCLK} \times 4^{PSC} \times (TOR + 1)$$

其中：

- t_{PCLK} ：PCLK 的周期
- PSC：USART_BRR.PSC 的值
- TOR：USART_TOR 寄存器的值

当 PCLK 频率为 48MHz 时，其定时计数范围最小值为 0.02μs，最大值约为 1.4s。



说明

- 只有当 USART_CR1.TOE 为 0 时才能写入其它值以启动定时计数器。
- 当 USART_SR.TO 为 1 或完成自动波特率检测后，硬件自动设置 USART_CR1.TOE 为 0。

几种功能对比如下所示：

表 18-8 多功能定时计数器功能

功能	计数器启动条件	应用场景
通用定时	向 USART_CR1.TOE 写入 1 后立即启动	通用定时器
接收超时计数	向 USART_CR1.TOE 写入 2 且接收使能后启动	在限定的时间等待另一主机回复信息
接收帧间超时计数	向 USART_CR1.TOE 写入 3 且接收使能，收到第一帧数据后启动	等待不定长的数据包传输完成
同步间隔段测量	向 USART_CR1.TOE 写入 4，USART_CR3.LBDE=1 后，硬件检测到 RXD 上有下降沿	LIN 模式
自动波特率检测	向 USART_CR1.TOE 写入 5，USART_CR3.ABRE=1 后，硬件检测到 RXD 上有输入数据边沿	波特率检测（异步串口模式/LIN 模式）

通用定时功能

USART_CR1.TOE 为 1 时工作于通用定时功能。当向 USART_CR1.TOE 写入 1 后，计数器立即启动，从 0 开始计数，如果计数值等于 USART_TOR 时，硬件自动设置 USART_SR.TO 为 1，若使能相应中断则可以产生多功能定时计数器中断。

接收超时计数功能

USART_CR1.TOE 为 2 时工作于接收超时计数功能。当向 USART_CR1.TOE 写入 2 且接收使能后，计数器立即启动，从 0 开始计数，如果计数值等于 USART_TOR 时，RXD 还没有收到新的数据，则停止计数，硬件自动设置 USART_SR.TO 为 1，若使能相应中断则可以产生多功能定时计数器中断。

如果 RXD 接收到新的数据（USART_SR.RC=1）时，计数值小于 USART_TOR，则计数器被清零并再次启动计数功能，从 0 开始计数。

接收帧间超时计数功能

USART_CR1.TOE 为 3 时工作于接收帧间超时计数功能。当向 USART_CR1.TOE 写入 3 且接收使能后，计数器在收到第一个字节的停止位后从 0 开始计数，如果计数值等于 USART_TOR 时，RXD 还没有收到新的数据，则停止计数，硬件自动设置 USART_SR.TO 为 1，若使能相应中断则可以产生多功能定时计数器中断。

如果 RXD 接收到新的数据（USART_SR.RC=1）时，计数值小于 USART_TOR，则计数器被清零并再次启动计数功能，从 0 开始计数。

同步间隔段测量功能

USART_CR1.TOE 为 4 时工作于同步间隔段测量功能。向 USART_CR1.TOE 写入 4 并向 USART_CR2.LBDE 写入 1，硬件自动识别输入信号 RXD 的下降沿后对低电平从 0 开始计数，如果计数值等于 USART_TOR 时，则停止计数，硬件自动设置 USART_SR.LBD 为 1，若使能相应中断则可以产生同步间隔段检出中断。

如果对低电平长度计数的计数值小于 USART_TOR 时，则等待下个 RXD 的下降沿后进入新一轮同步间隔段测量。

自动波特率检测功能

USART_CR1.TOE 为 5 时工作于自动波特率检测功能。

详细信息请参见“[自动波特率检测](#)”章节。

18.3.1.6 中断

参考《中断控制器 (NVIC) 》“功能说明-中断源”章节的中断源和 NVIC 中断输入的对应关系，本章节的 2 个 USART 各提供 1 类中断：USARTx 中断。

表 18-9 各模式中断有效性

功能名称	中断使能位 (USART_IER)	异步串口	同步串口	智能卡	红外	LIN
发送数据空中断	TXE	✓	✓	✓	✓	✓
发送完成中断	TC	✓	✓ ⁽²⁾	✓	✓	✓
接收完成中断	RC	✓	✓	✓	✓	✓
接收数据校验错误中断	PE	✓	×	✓	✓	×
接收帧错误中断	FE	✓	×	✓	✓	✓
接收数据上溢错误中断	OR	✓	✓	✓	✓	✓
多功能定时计数中断	TO	✓	✓	✓	✓	✓
多机通信模式下地址匹配中断	AM	✓	×	×	×	×
CTS 中断	CTS	✓	✓	×	×	×
自动波特率检出中断	ABR	✓	×	×	×	✓
同步间隔段检出中断	LBD	×	×	×	×	✓
输入数据唤醒中断	DWK	✓	×	×	✓	✓



说明

1. “✓”表示有效；“×”表示无效。
2. 同步串口模式下，使用外部时钟时不支持发送完成中断。

18.3.2 异步串口模式

18.3.2.1 模式概述

异步串行接口，提供了一个灵活的异步传输的全双工数据交换接口，通常也被用作 RS232 标准通信。

接口通过 1~5 个引脚与其他设备连接在一起，应用中通常仅使用两个管脚：RXD 和 TXD。

- RXD：接收数据输入，通过过采样技术来区别数据和噪音。
- TXD：发送数据输出，当不发送数据时，TXD 管脚保持高电平。

通过设置 USART_CR2.SYNC=0 来选择异步串口模式。

在异步串口模式下，须将以下位清零：

- USART_CR3.SCEN、USART_CR3.LINEN 和 USART_CR3.IREN

18.3.2.2 功能框图

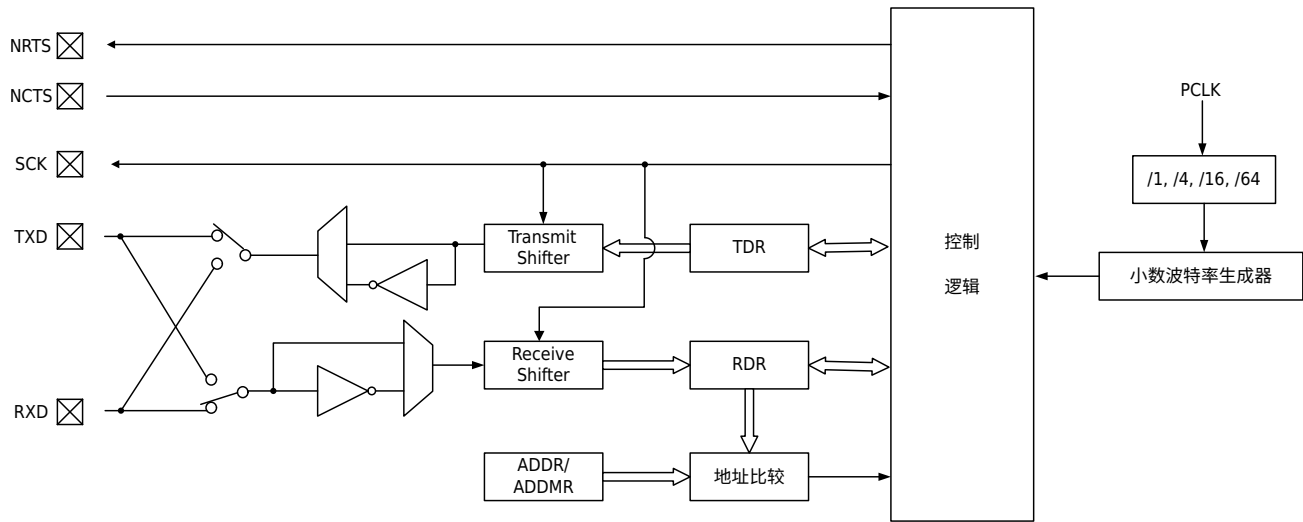


图 18-4 异步串口功能框图

各管脚功能说明如下方所示：

- SCK：USART 工作时钟输出
- TXD：USART 发送数据管脚
- RXD：USART 接收数据管脚
- NRTS：USART 硬件流控 RTS 输出管脚
- NCTS：USART 硬件流控 CTS 输入管脚

18.3.2.3 数据格式

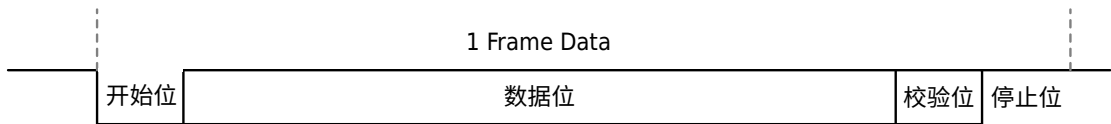


图 18-5 异步串口模式数据格式

USART 串行数据帧的格式为：起始位、8~9 个数据位、奇偶校验位（可选）、1~2 个停止位。

起始位的极性与数据线空闲状态相反，停止位与数据线空闲状态相同。

开始位和停止位都用于异步数据传输过程中的数据同步。

本模块支持两种数据位顺序：先传输 LSB 或先传输 MSB。

发送时，USART 模块读出发送缓存中的数据，然后从 TXD 管脚按上述数据帧格式输出。

接收时，USART 模块查找 RXD 管脚上的起始位，找到起始位后按上述数据帧格式接收数据位、奇偶校验位、停止位。如果没有找到停止位，则 USART 模块会认为整个数据帧传输失败。

本模块支持两种查找起始位的方法：RXD 管脚上的下降沿或 RXD 管脚上的低电平（USART_CR2.RXINV=0 时）。

18.3.2.4 发送器

当设置 USART_CR1.TE 为 1，TXD 管脚可以对外发送数据。当 USART_TDR 寄存器内有数据，则硬件自动将数据加载到发送移位寄存器，并按照用户配置的数据帧格式从 TXD 管脚移位输出。每当 USART_TDR 寄存器内的数据加载到移位寄存器时，USART_SR.TXE 标志由硬件置 1。用户查询到 USART_SR.TXE 标志为 1 时，即可向 USART_TDR 寄存器写入下一个待发送的数据。每当 TXD 管脚发送完成一帧数据时，USART_SR.TC 标志会被硬件置 1 以指示当前的发送进展。

下图为发送两个字节的示例；其中数据长度为 8 比特，无奇偶校验，一个 STOP 位。

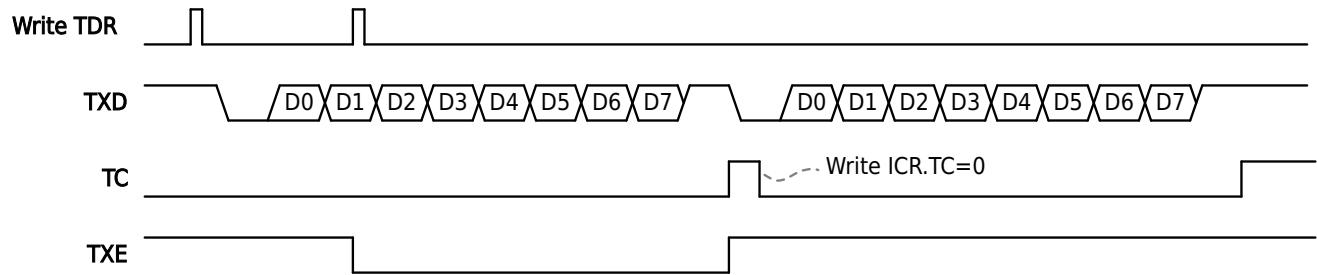


图 18-6 发送数据示例



说明

- 向 USART_TDR 中写入待发送数据前，USART_CR1.TE 位必须先置 1。
- 数据发送期间不应写 USART_CR1.TE=0，否则会冻结波特率计数器，造成 TXD 管脚输出非预期的波形。

18.3.2.5 接收器

当设置 USART_CR1.RE 为 1，本模块可以接收从 RXD 管脚收到的数据。每当一帧数据接收完成，USART_SR.RC 标志会被硬件置 1，接收到的数据可以从 USART_RDR 寄存器中读出。

本模块支持两种起始位检测方式：检测到 RXD 管脚上的电平作为起始位、检测到 RXD 管脚上的边沿作为起始位，详见 USART_CR1.SBS 说明。

本模块支持两种采样方式：8 倍采样、16 倍采样。8 倍采样是指每个波特率周期对 RXD 管脚上的信号进行 8 次采样；16 倍采样是指每个波特率周期对 RXD 管脚上的信号进行 16 次采样。

本模块在接收数据时，可检测三种错误：上溢错误、奇偶校验错误、帧错误。发生任何一种错误后，本模块继续接收数据，用户程序应尽快进行出错处理，并清除错误标志以重启数据接收。

- 上溢错误：当 USART_SR.RC 为 1（USART_RDR 中的数据未读取）且接收到一帧新的数据时，发生上溢错误。发生该错误时，USART_SR.OR 会被硬件置 1，新收到的数据会覆盖前帧数据。向 USART_ICR.OR 写入 0，可以清除该标志。如果 USART_SR.RC 为 1 后没有读走 USART_RDR 中的数据，但通过写 USART_ICR.RC=0 把 USART_SR.RC 清为 0，那么 USART_SR.OR 不会被置 1。
- 奇偶校验错误：当检测到数据帧的奇偶校验不正确时，发生奇偶校验错误。发生该错误时，USART_SR.PE 会被硬件置 1，接收到的数据被传送到 USART_RDR。向 USART_ICR.PE 写入 0，可以清除该标志。
- 帧错误：当接收数据帧未在预期时间检测到停止位时，发生帧错误。发生该帧错误时，USART_SR.FE 会被硬件置 1，接收到的数据被传送到 USART_RDR。向 USART_ICR.FE 写入 0，可以清除该标志。

下图为接收两个字节的示例：其中数据长度为 8 比特，无奇偶校验，一个 STOP 位。

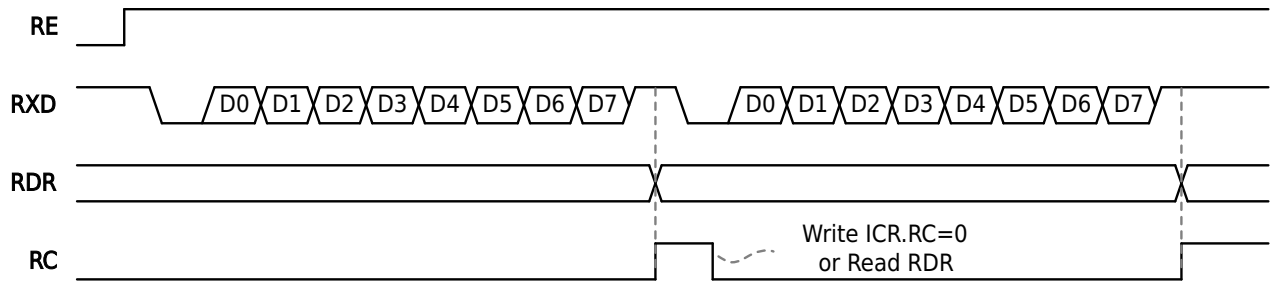


图 18-7 接收数据示例

18.3.2.6 单线半双工

当设置 USART_CR1.COMB 和 USART_CR1.RE 为 1、USART_CR1.TE 为 0 时，本模块工作于单线半双工模式，仅使用 TXD 管脚进行数据的发送与接收。

当没有发送数据时，TXD 管脚处于输入态，等待接收总线上其它主机发出数据。每当用户向 USART_TDR 寄存器写入数据后，TXD 管脚自动变为输出态以便向总线上发送用户写到 USART_TDR 内寄存器的数据。

为防止单总线出现冲突，用户应采取适当的应用层保护措施，以确保不会出现多个主机同时向总线发送数据。



说明

单线半双工工作时不支持 DMA 模式。

18.3.2.7 多处理器通信



说明

- 多处理器通信只适用于异步串口模式。
- 多处理器通信时，作为从机的 USART 模块需要将 TXD 配置为开漏输出；从机在完成发送数据后，需要及时设置 USART_CR1.TE 为 0，再进行主机和其他从机之间的通信。

当设置 USART_CR1.MPE 为 1 时，USART 工作于多处理器通讯模式。主机可以发送地址帧及数据帧，从机可以在收到匹配的地址帧后才接收数据帧。

数据格式

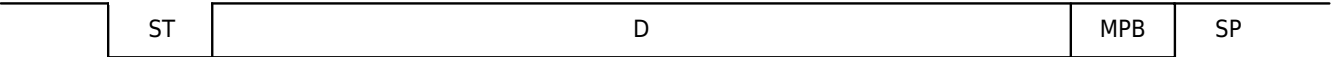


图 18-8 多处理器通信数据格式

- 开始位（ST）：固定由一位的低电平构成。
- 数据位（D）：可以配置成 8 位或者 9 位。
- 多处理器位（MPB）：可以配置成发送通信数据或者发送 ID。
- 停止位（SP）：固定为高电平，可以配置成 1 位或者 2 位。

主机发送地址帧及数据帧

- 如果字长为 8，即 USART_CR1.CHLEN=0：
 - ▶ 当 USART_TDR.MPID 为 1 则发送地址帧，USART_TDR[7:0]为目标从机的地址。
 - ▶ 当 USART_TDR.MPID 为 0 则发送数据帧，USART_TDR[7:0]为待发送的数据。
- 如果字长为 9，即 USART_CR1.CHLEN=1：
 - ▶ 当 USART_TDR.MPID 为 1 则发送地址帧，USART_TDR[7:0]为目标从机的地址。
 - ▶ 当 USART_TDR.MPID 为 0 则发送数据帧，USART_TDR[8:0]为待发送的数据。

从机接收地址帧及数据帧

设置 USART_CR1.SLME 为 1 时，从机只接收与自己地址相匹配的地址帧，忽略不匹配的地址帧及数据帧。当从机从 RXD 接收到地址帧时，硬件自动对比地址帧内的地址与本机地址；只有两者匹配时才会将收到的地址存入 USART_RDR 寄存器并置位 USART_SR.RC 及 USART_SR.AM。

从机地址配置

从机地址由用户配置地址（USART_ADDR）和地址掩码（USART_ADDMR）相与组成。
从机地址：USART_ADDR & USART_ADDMR
收到地址帧时，检测到从总线收到的地址与 USART_ADDMR 的值相与后值与用户配置地址相同，则认为收到了与从机地址相匹配的地址帧。
例：USART_ADDR=0x55，USART_ADDMR=0xF0，则从机可以匹配的地址为 0x50~0x5F（用户配置地址）。

18.3.3 同步串口模式

18.3.3.1 模式概述

同步串口模式下数据的发送和接收都与 SCK 时钟管脚同步。

通过设置 USART_CR2.SYNC=1 来选择同步串口模式。

在同步串口模式下，须将以下位清零：

- USART_CR1.COMB、USART_CR1.SLME、USART_CR1.MPE、USART_CR1.OVER8、USART_CR1.STOP、USART_CR1.CHLEN、USART_CR1.PCE
- USART_CR3.SCEN、USART_CR3.LINEN、USART_CR3.IREN

18.3.3.2 时钟

同步串口模式的时钟源有两种选择，可通过 USART_CR2.CLKC 设置：

- 内部时钟源：内部波特率生成器生成的时钟
同步时钟从 CK 管脚输出，一帧数据输出 8 个时钟脉冲，既不发送数据也不接收数据时，时钟输出固定为高电平。
- 外部时钟源：CK 管脚输入的时钟
外部时钟源即从 CK 管脚输入的时钟作为通信时钟。

18.3.3.3 数据格式

同步串口模式下一帧数据固定由 8 位组成，一帧数据的收发需要 8 个同步时钟脉冲。

发送数据时数据在同步时钟的下降沿送出，接收数据时数据在同步时钟的上升沿被采样。

除连续传输模式外，同步时钟在没有数据传输时固定为高电平；在有数据传输时，发送数据最后一位发送完后，数据线保持最后一位的值。

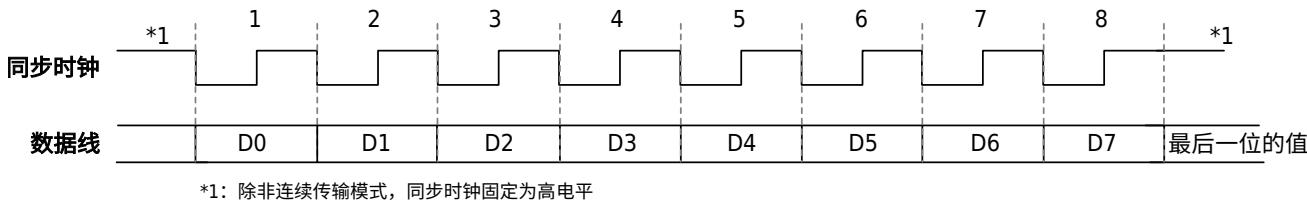


图 18-9 同步串口模式数据格式

18.3.3.4 发送器

发送使能位 USART_CR1.TE=1 时，发送移位寄存器中的数据在 TXD 管脚输出，相应的时钟在 CK 管脚输出或输入。

发送数据寄存器 USART_TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

发送数据寄存器 USART_TDR 的写入可通过中断程序完成（CPU 或者 DMA），一次中断只能写入一次数据。

发送数据设定步骤（使用内部时钟源）

- 步骤 1. 将 USART_CR1，USART_SR 寄存器设定为复位值；
- 步骤 2. 初始化端口，配置 TXD 为数字输入带上拉；将 TXD 映射到需要的管脚；将 SCK 映射到需要的管脚作为输出时钟；
- 步骤 3. 通过 USART_CR2.CLKC 位选择内部时钟源；
- 步骤 4. 设定 USART_CR1（除了 TE），USART_CR2，USART_CR3 寄存器；
- 步骤 5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- 步骤 6. 设定 USART_CR1.TE=1，USART_IER.TXE=1；
- 步骤 7. 通过中断程序写数据到 USART_TDR，数据传输到发送移位寄存器，发送开始；



说明

CTS 功能有效时，CTS 管脚输入低电平时数据传输到发送移位寄存器，发送开始。

- 步骤 8. 如果需要连续发送数据，则重复步骤 7；
- 步骤 9. 通过中断程序写入最后一个发送数据，写 USART_IER.TXE=0，USART_IER.TC=1，最后一个数据发送结束后，发生发送完成中断。

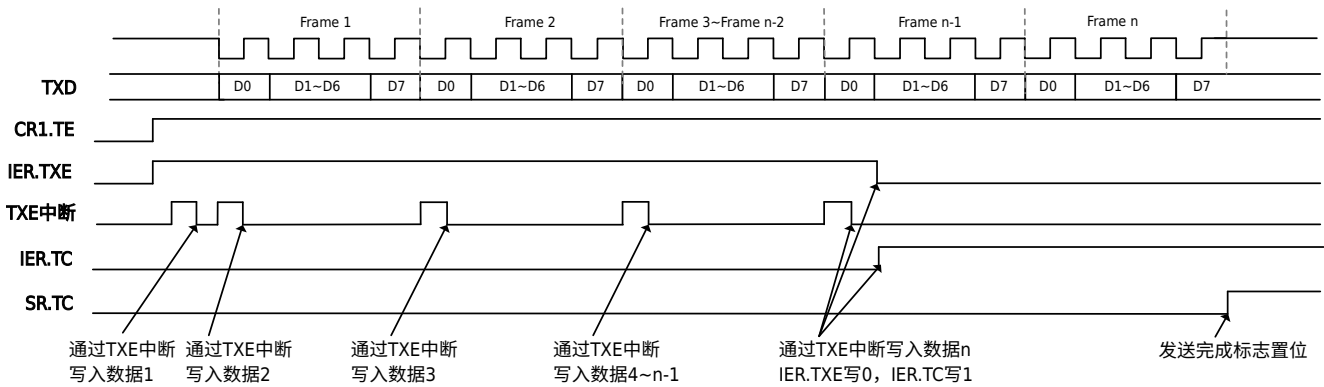


图 18-10 同步串口模式发送数据图例 1 (LSB/无 CTS 功能) (使用内部时钟源)

发送数据设定步骤 (使用外部时钟源)

- 步骤 1. 将 USART_CR1，USART_SR 寄存器设定为复位值；
- 步骤 2. 初始化端口，配置 TXD 为数字输入带上拉；将 TXD 映射到需要的管脚；将 SCK 映射到需要的管脚作为输入时钟；
- 步骤 3. 通过 USART_CR2.CLKC 位选择外部时钟源；
- 步骤 4. 设定 USART_CR1 (除了 TE)，USART_CR2，USART_CR3 寄存器；
- 步骤 5. 设定 USART_CR1.TE=1，USART_IER.TXE=1；
- 步骤 6. 通过中断程序写数据到 USART_TDR，数据传输到发送移位寄存器，发送开始；



说明

CTS 功能有效时，CTS 管脚输入低电平时数据传输到发送移位寄存器，发送开始。

- 步骤 7. 如果需要连续发送数据，则重复步骤 6；
- 步骤 8. 通过中断程序写入最后一个发送数据，继续等待 TXE 中断，在下一个中断程序中写入一个任意的无效数据 (如果最后一个数据最高位为 1，建议为 0xFF；如果最后一个数据最高位为 0，建议为 0x0)，最后一个数据发送结束后，发生 TXE 中断表示完成发送。

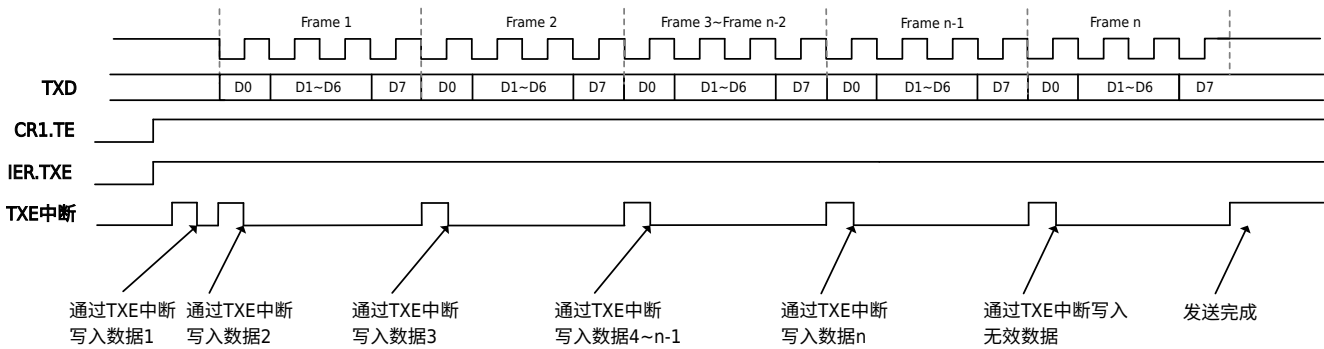


图 18-11 同步串口模式发送数据图例 1 (LSB/无 CTS 功能) (使用外部时钟源)

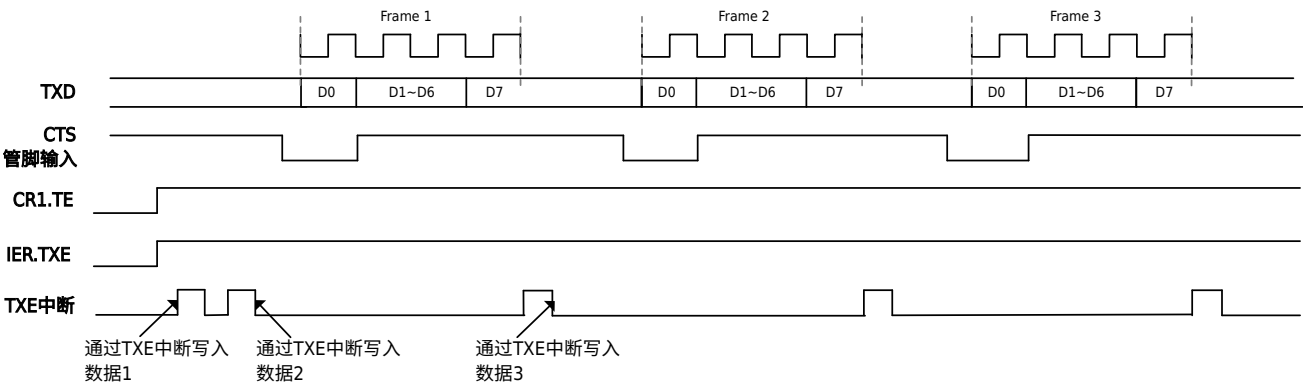


图 18-12 同步串口模式发送数据图例 2 (LSB/CTS 功能)

发送器中断

同步串口模式使用内部时钟源时，发送器支持两种中断，发送数据寄存器空中断和发送完成中断；使用外部时钟源时，发送器仅支持发送数据寄存器空中断。

USART_IER.TXE=1，USART_TDR 寄存器的值传送到发送移位寄存器时中断发生。

使用内部时钟源时，USART_IER.TC=1，发送数据的最后一位时 USART_TDR 寄存器没有更新则中断发生。

18.3.3.5 接收器

接收数据设定步骤

- 步骤 1. 将 USART_CR1，USART_SR 寄存器设定为复位值；
- 步骤 2. 初始化端口，配置 RXD 为数字输入带上拉；将 RXD 映射到需要的管脚；将 SCK 映射到需要的管脚；
- 步骤 3. 通过 USART_CR2.CLKC 位选择时钟源；
- 步骤 4. 设定 USART_CR1（除了 RE），USART_CR2，USART_CR3 寄存器；
- 步骤 5. 如果选择内部时钟源，需要设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- 步骤 6. 设定 USART_IER.RC=1，USART_IER.OR=1，USART_CR1.RE=1；



说明

使用 RTS 功能时，USART_CR1.RE=1 后 RTS 管脚输出低电平。

- 步骤 7. 输入同步时钟或者内部生成的同步时钟工作时开始接收数据，数据接收到移位寄存器。
 - 发生上溢错误时，数据被覆盖并发生上溢错误中断，进行错误处理；
 - 无上溢错误发生时，接收到的数据传送到 USART_RDR 寄存器中，发生接收完成中断，通过接收完成中断程序将接收到的数据在接收到下一帧数据最后一位前读取，之后重复步骤 7 实现连续接收数据功能。



说明

使用 RTS 功能时，数据读取后 RTS 管脚输出低电平。

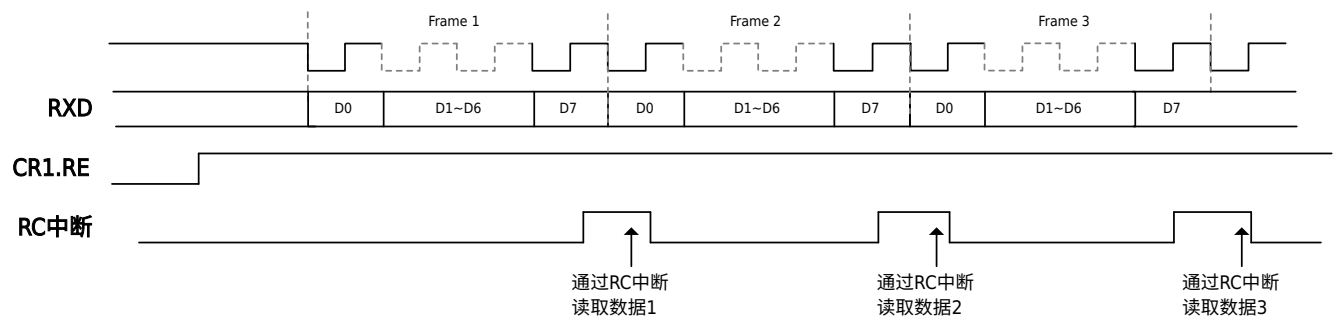


图 18-13 同步串口模式接收数据图例 1 (LSB/无 RTS 功能)

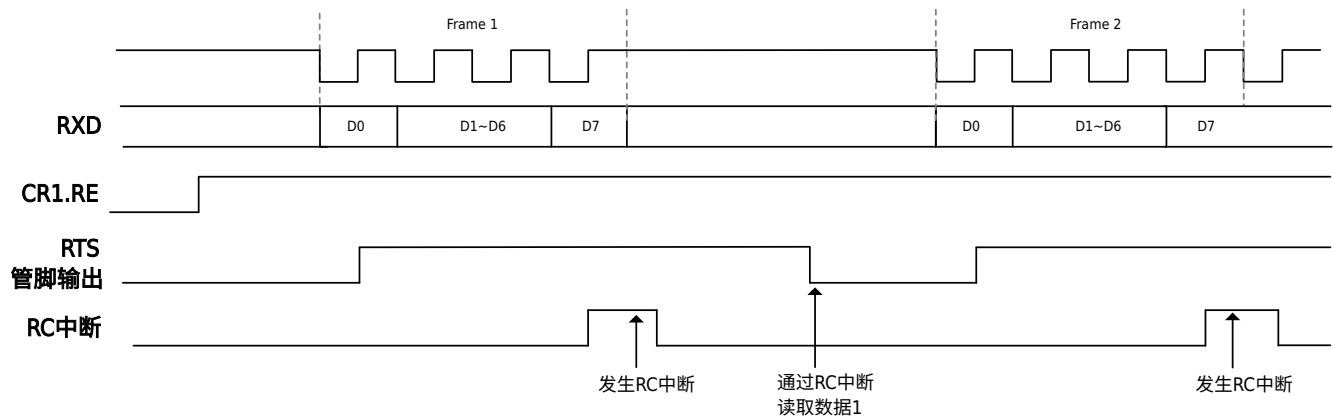


图 18-14 同步串口模式接收数据图例 2 (LSB/RTS 功能)

错误处理

同步串口模式下接收数据时接收错误为上溢错误 (USART_SR.OR)，上溢错误发生的条件是 USART_RDR 值未被读取的情况下又收到新的数据。发生上溢错误后未读取的数据被覆盖。可以通过写对应的清零寄存器位将错误标志清零来重启数据传输。

为了避免上溢错误发生，应该在收到当前帧最后一位之前将前一帧收到的数据读取。

接收器中断

同步串口模式下接收器支持两种中断，接收完成中断和上溢错误中断。

USART_IER.RC=1，数据从移位寄存器传送到 USART_RDR 时中断发生。

USART_IER.OR=1，接收数据发生上溢错误中断发生。

以上中断均在第 8 个 CK 管脚上升沿后产生。

18.3.3.6 同时发送接收数据

USART 同步串口模式支持全双工动作，同时发送和接收数据。

同时发送和接收数据时需要一个命令把 USART_CR1.RE 和 USART_CR1.TE 同时写 1，设好中断使能位，其它设定流程与发送器和接收器相同。

18.3.3.7 同步串口模式注意事项

1. 使用外部输入时钟发送数据时，在 USART_TDR 更新后等待 6 个 PCLK 以上再输入时钟。输入时钟总周期要大于等于 6 个 PCLK，高电平和低电平期间必须大于等于 2 个 PCLK。
2. 连续发送数据时，下一帧数据需要在当前帧最后一位发送前完成更新。

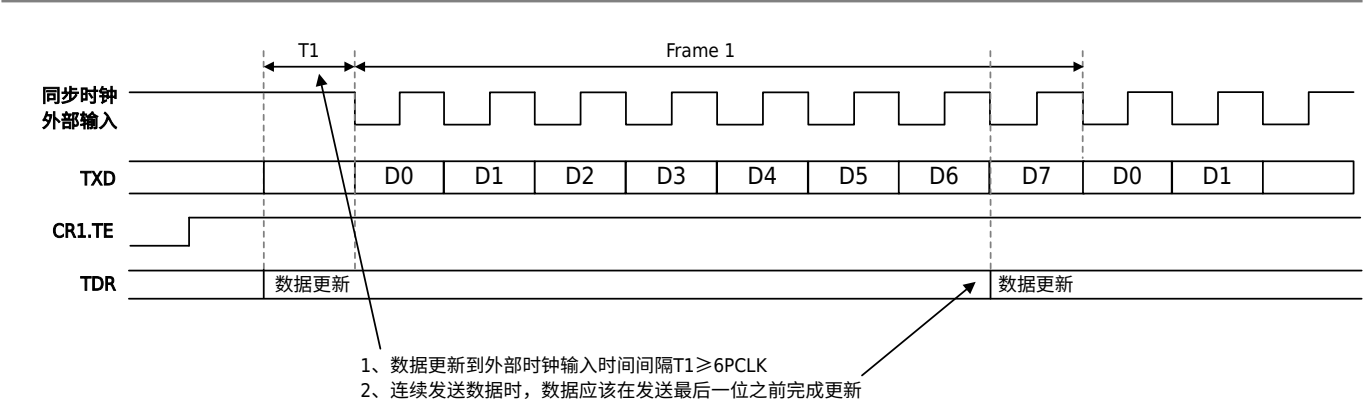


图 18-15 同步串口模式注意事项

18.3.4 智能卡 (7816) 模式

18.3.4.1 模式概述

ISO7816 是一种接触式的读卡协议，本模块只作为智能卡主机接口，通过写 USART_CR3.SCEN=1 选择智能卡模式。智能卡模式下，须将以下位清零：

- USART_CR1.COMB、USART_CR1.SLME、USART_CR1.MPE、USART_CR1.SBS、USART_CR1.OVER8
- USART_CR2.SYNC
- USART_CR3.LINEN、USART_CR3.IREN

此外，需要设置 USART_CR2.CLKC 位，为智能卡提供时钟。

18.3.4.2 连接示意图

智能卡接口支持 ISO/IEC 7816-3 规定的智能卡通信协议，同时支持 T=0（字符模式）和 T=1（块模式）。

下图为智能卡模式的连接示意图：

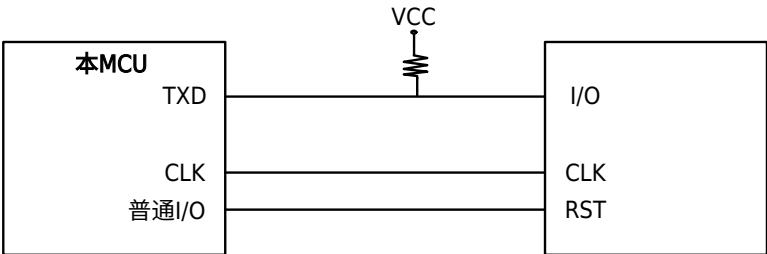


图 18-16 智能卡连接示意图

18.3.4.3 时钟

内部时钟源

智能卡模式时只能使用内部波特率生成器生成的时钟作为时钟源。

一位数据传输的基本时钟数为 USART_CR3.BCN 设定值。

寄存器 USART_CR2.CLKC 位和 USART_BRR 用来设置智能卡模式的时钟输出。

采样和接收容差

检测到接收管脚（TXD）下降沿后，USART 会基于内部基本时钟对接收数据进行时钟同步，从而开始数据接收。

每位接收数据被采样三次作多数表决后送入移位寄存器。

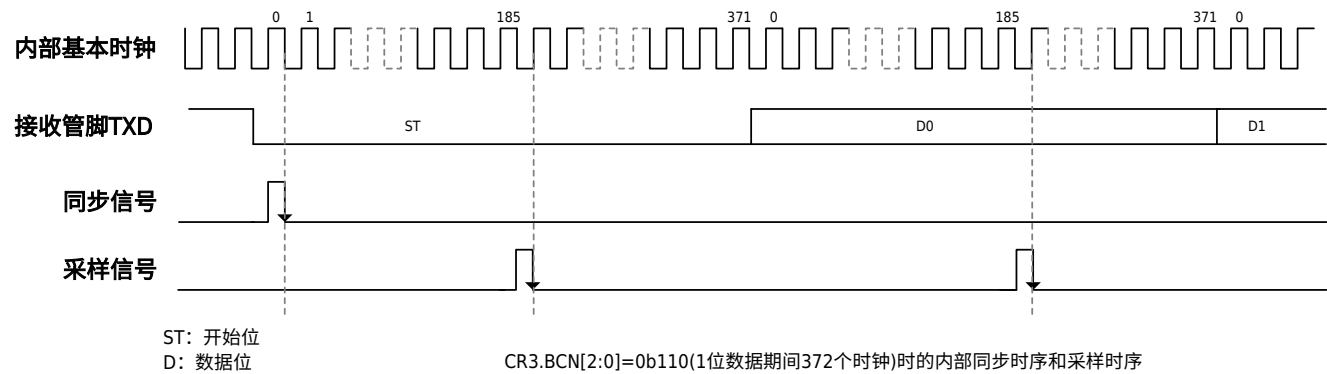


图 18-17 智能卡模式内部同步时序和采样时序图

接收容差计算公式：

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

其中：

- M：接收容差
- N：一位数据传输所需要的时钟数（BCN[2:0]设定值）
- L：帧的长度（10）
- D：时钟占空比
- F：时钟偏差

18.3.4.4 数据格式

智能卡模式时一帧数据由起始位，数据位和校验位组成。

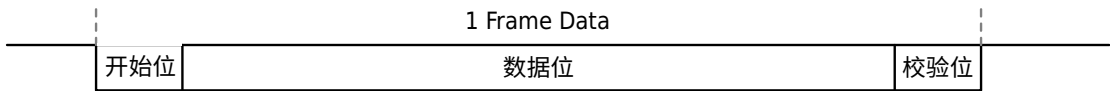


图 18-18 智能卡模式数据格式

- 起始位 (ST)：固定由一位的低电平构成。
- 数据位 (D)：固定为 8 位数据。
- 校验位 (P)：需配置成 1 位偶校验。

下图给出数据 0x3B 数据格式的图例。

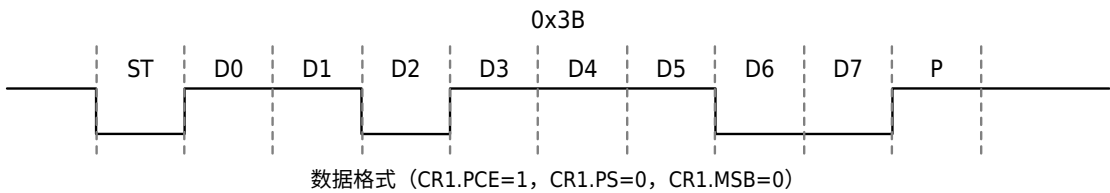


图 18-19 能卡模式数据格式示例

18.3.4.5 智能卡的初始化设定步骤

- 步骤 1. 将 USART_CR1 寄存器设定为复位值；
- 步骤 2. 初始化端口，均设置为数字输入带上拉；设定需要使用的管脚；
- 步骤 3. 状态寄存器确认，USART_SR 寄存器设定为复位值；
- 步骤 4. 设定 USART_CR1（除了 TE、RE）、USART_CR2、USART_CR3 寄存器；
- 步骤 5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- 步骤 6. 设定 USART_CR2.CLKC 位的时钟控制；

步骤 7. 设定 USART_CR1 寄存器（TE、RE 位），根据需要设置 TE 和 RE，TE 和 RE 不要同时设定为 1。发送模式切换到接收模式，或者接收模式切换到发送模式时，需要重新设定步骤 7。

使用智能卡功能时，字长 USART_CR1.CHLEN、校验控制位 USART_CR1.PCE、停止位长度 USART_CR1.STOP 可以是任意值，奇偶校验位 USART_CR1.PS 根据需要须合理设置为奇或偶，反相控制位 USART_CR2.TXINV=0。数据接收和发送都通过 TXD 进行传输，与 USART_CR1.COMB 值无关。

18.3.4.6 智能卡模式动作说明

智能卡模式时，中断（发送数据空中断）的标志位为 USART_SR.TXE。USART_SR.TXE=1 而且 USART_IER.TXE=1 时产生中断。

功能概述

发送数据时两帧数据间（从校验位结束到下一帧的起始位开始）有 2etu（Elementary Time Unit）以上的保护时间。

发送数据时如果检测到接收方发送的错误信号，则经过 2etu 后自动重发数据。

接收数据发生校验错误时，发送 1etu 的低电平即错误信号，错误信号送出的时序为接收开始经过 10.5etu。1etu 长度为 USART_CR3.BCN 值所决定的 SCK 管脚的时钟个数。

发送说明

1. 一帧数据发送完成后，如果检测到接收方发送的错误信号，则置 USART_SR.FE=1 发生帧错误中断（当 USART_IER.FE=1 时），USART_SR.TC 标志不置 1，数据自动重发。USART_SR.FE 位必须在接收到下一帧校验位之前清零。
2. 一帧数据无错误发送完成，置 USART_SR.TC=1，发生中断（USART_IER.TC=1）。通过中断程序写入数据，则可连续发送数据。

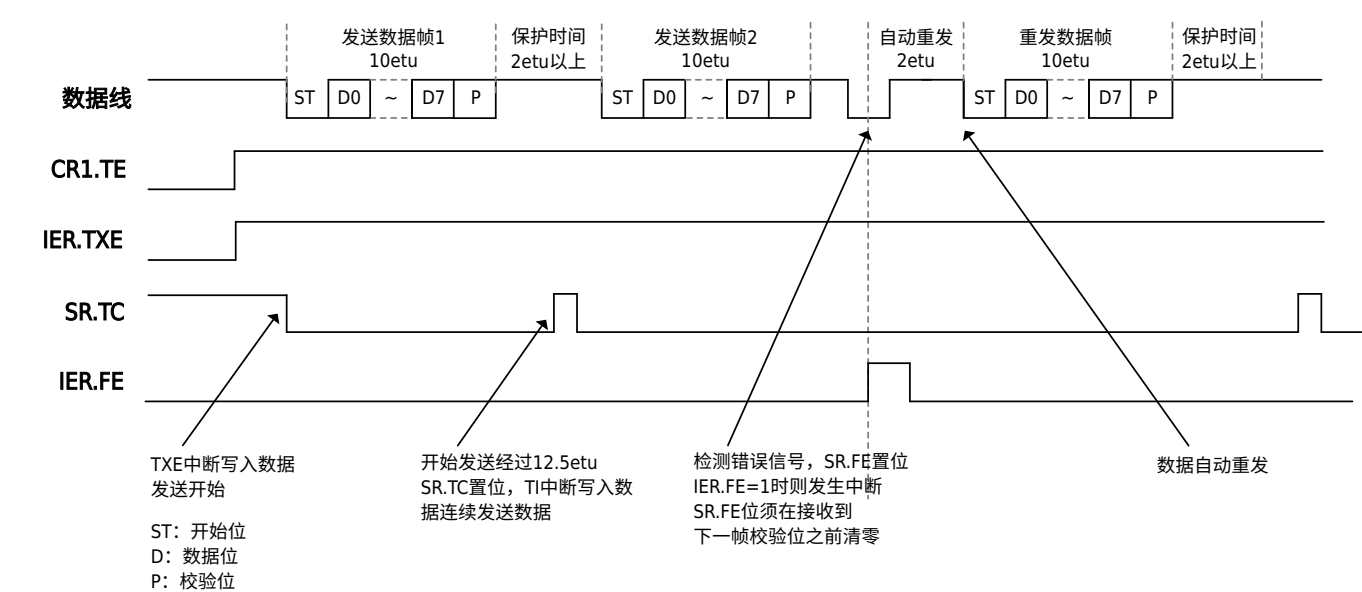


图 18-20 智能卡模式发送数据图例

接收说明

1. 接收数据时，如果检测到校验错误，则 USART_SR.PE=1，如果 USART_IER.PE=1 则发生校验错误中断。USART_SR.PE 需要在接收到下一帧校验位之前清零。
2. 发生校验错误时，会发送 1etu 的低电平，即错误信号，要求发送方重新发送数据。
3. 正常接收数据，可以通过中断程序读取接收到的数据，连续接收。
4. 接收数据时，检测上溢错误。

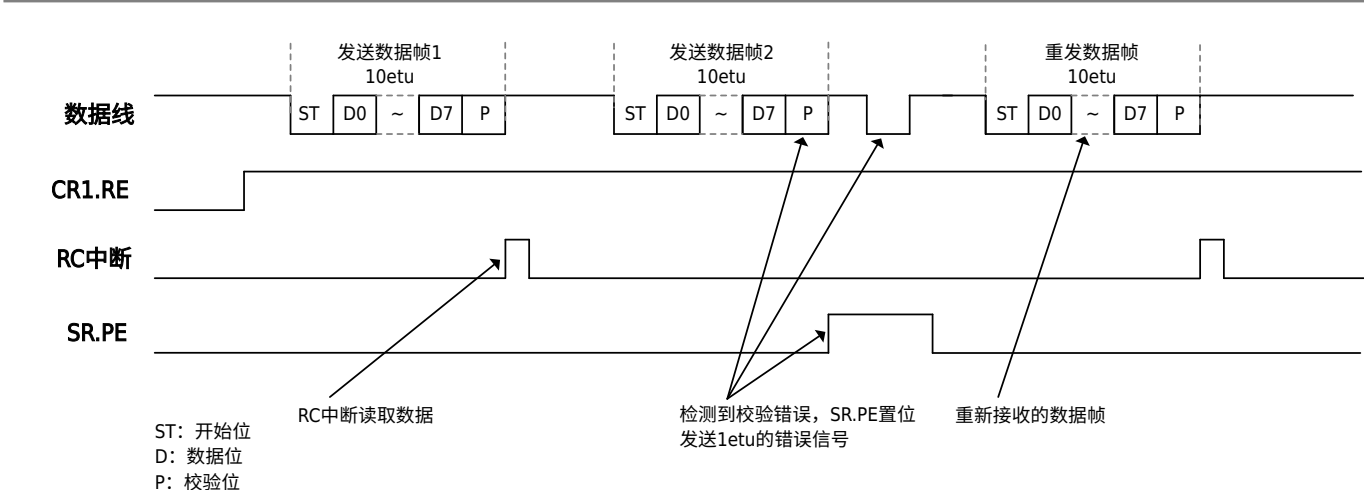


图 18-21 智能卡模式接收数据图例

18.3.5 红外模式

18.3.5.1 模式概述

通过写 USART_CR3.IREN=1 来选择红外模式。红外模式下，须将以下位清零：

- USART_CR1.COMB、USART_CR1.SLME、USART_CR1.MPE
- USART_CR3.SCEN、USART_CR3.LINEN

本模块包含 2 种红外模式：3/16 波特率周期的红外脉冲和 38K 调制输出。同一模块的 2 种模式不能同时工作。

使用时合理配置字长 USART_CR1.CHLEN、校验控制位 USART_CR1.PCE、USART_CR1.PS、停止位长度 USART_CR1.STOP、反相控制位 USART_CR2.TXINV、USART_CR2.RXINV 等寄存器位，以便满足红外应用需求。

18.3.5.2 红外模式 1：红外脉冲宽度为波特率周期的 3/16

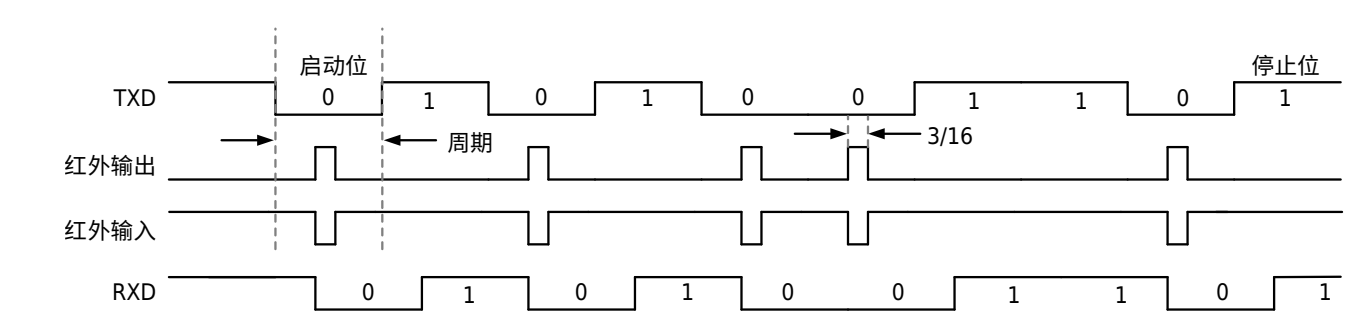


图 18-22 红外数据调制

通过写 USART_CR3.IREN=1 和 USART_CR3.IRCW=0 来选择 3/16 波特率周期模式。红外模式下，USART_CR3.LINEN 和 USART_CR3.SCEN 位必须清零。

红外协议物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0 (如上图)。

USART 的红外输出脉冲流会发送到芯片外部的红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

芯片外部的红外接收解码器将接收到的不归零 (NRZ) 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平 (标记状态)。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

1. 红外通信是一个半双工通信协议。为防止破坏数据，应避免同时进行发送和接收。
2. “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16 (参见上图)。

3. 规范要求脉冲宽度值要大于 $1.41\mu\text{s}$ 。
4. 红外模式下，USART_CR1 和 USART_CR2 寄存器中某些位被屏蔽为无效位，用户写 1 或写 0 都不影响红外正常工作，详见寄存器的位功能描述。
5. 红外模式下输入输出信号可以通过 USART_CR2.RXINV 和 USART_CR2.TXINV 取反后再输入到 USART 模块或从 USART 模块输出。

18.3.5.3 红外模式 2：占空比可调的 38K 调制

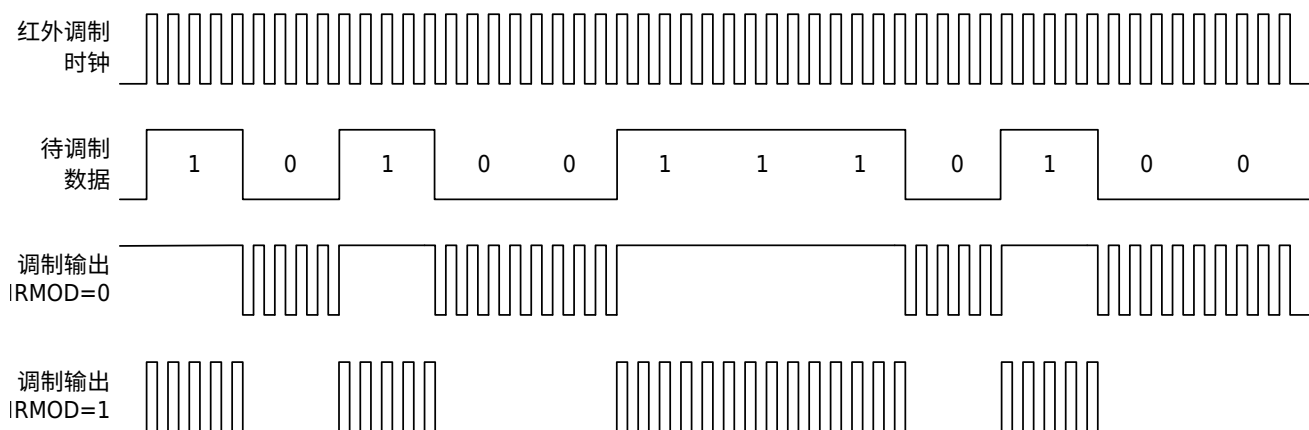


图 18-23 38K 红外调制

红外调制的方式为：发送数据 0 时输出红外调制，发送数据 1 时不调制。38K 红外调制频率由寄存器 USART_IR 的值和模块当前工作频率决定，其计算公式：

$$F = \text{PCLK} / (\text{USART_IR.DCNTH} + \text{USART_IR.DCNTL} + 2)$$

其中：

- PCLK：模块工作频率
- USART_IR.DCNTH：38K 调制频率的高电平持续时间
- USART_IR.DCNTL：38K 调制频率的低电平持续时间

应用上考虑到 PNP 和 NPN 两种红外管的需求，可以设置 USART_CR2.RXINV 和 USART_CR2.TXINV 将接收和发送的数据反相。USART_IR.DCNTH 和 USART_IR.DCNTL 是不反相时的占空比设置，如果发送数据反相，则这两个数据需要互换。

18.3.6 LIN 模式

18.3.6.1 模式概述

通过写 USART_CR3.LINEN=1 来选择 LIN 模式。LIN 模式下，须将以下位清零：

- USART_CR1.COMB、USART_CR1.STOP、USART_CR1.SLME、USART_CR1.MPE、USART_CR1.PCE
- USART_CR2.SYNC
- USART_CR3.SCEN、USART_CR3.IREN

LIN 是 Local Interconnect Network（局域互联网络）的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。

LIN 通信是单主控通信，一个主控设备最多可连接 15 个设备。

总线上各设备间是“线与”的关系，显性电平起主导作用：

- 总线上任一节点发送显性电平（低电平）时，总线呈显性电平；
- 只有所有的节点都发送隐性电平（高电平）或不发送信息（不发送任何信息时总线默认呈隐性电平）时，总线才呈现隐性电平。

LIN 总线支持 ISO9141；LIN 从机设备通过 LIN 连接到主控设备，主要用于开关，传动装置和传感器等控制。

本产品中的 USART 中集成了多功能定时器，方便实现 LIN 总线功能，包含数据传输、波特率设定、同步间隔段检测、同步段的测量以及波特率计算等，RXD 管脚边沿中断 (USART_IER.DWK) 和 I/O 端口外部中断输入功能可用于唤醒信号的检测。

根据应用，USART 作为 LIN 使用时是 RXD/TXD 双线模式，外部需配置相应芯片用于升压和总线转换。

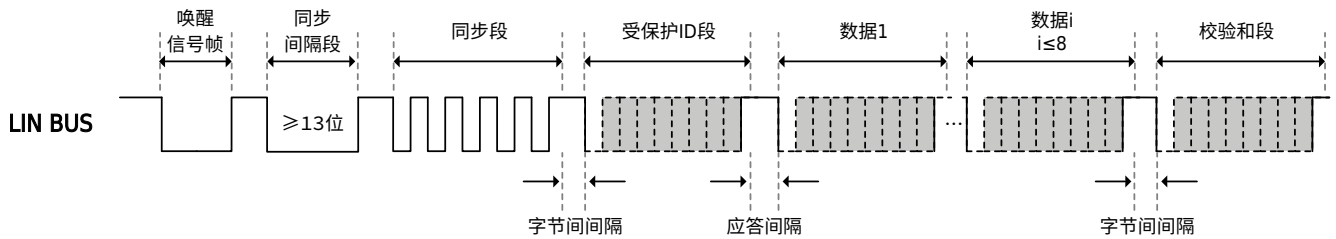


图 18-24 LIN 总线数据和帧结构



说明

唤醒信号帧仅当设备需要唤醒时才需要。

18.3.6.2 LIN 数据格式

LIN 的一帧数据由 1 个起始位、8 个数据位和 1 个停止位组成，以 LSB 方式发送和接收数据。

18.3.6.3 LIN 发送

唤醒信号帧的发送

设置合适的波特率，通过发送数据 0x80 实现唤醒信号帧的发送。

同步间隔段发送

同步间隔段表示一帧数据的开始，至少持续 13 位（以主机节点速率为准）的显性电平（低电压）。本产品中采用 USART_CR3.BREAK 和 USART_TDR 作为发送显性电平的同步间隔时间，USART_TDR 寄存器中设置发送持续位的“0”的个数。

同步段发送

通过发送数据 0x55 实现同步段数据的发送。

LIN 主控设备软件操作步骤

- 步骤 1. 将 USART_CR1 寄存器设定为复位值。
- 步骤 2. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，配置 TXD 和 RXD 为数字输入带上拉，将 TXD 和 RXD 映射到需要的管脚。
- 步骤 3. 设定 USART_CR1（除 TE），USART_CR2，USART_CR3（除 LINEN）寄存器。
- 步骤 4. 设定 USART_BRR.PSC 选择预分频值。
- 步骤 5. 设定 USART_BRR 寄存器，设定信号帧的波特率。
- 步骤 6. （可选）唤醒信号帧的发送：
 - a. 设定 USART_CR1.TE=1。
 - b. 写通信数据 0x80 到 USART_TDR，数据传输到移位寄存器，发送开始。
 - c. 等待发送完成。
- 步骤 7. 同步间隔段的发送：
 - a. 设定 USART_CR3.BREAK=1，USART_CR3.LINEN=1。
 - b. 如果未执行步骤 6 则设定 USART_CR1.TE=1，否则跳过此步操作。
 - c. 写同步间隔段长度到 USART_TDR，发送同步间隔段低电平到总线。
 - d. 等待发送完成。
- 步骤 8. 同步段的发送，其过程与异步串口发送过程相同：

a. 写数据 0x55 到 USART_TDR。

b. 等待发送完成。

步骤 9. 设置字节间间隔：

a. 写字节间等待时间长度到 USART_TOR 中。

b. 写 USART_CR1.TOE=1 启动软件定时。

c. 等待 USART_SR.TO=1。

步骤 10. 受保护 ID 段：写受保护 ID 段数据 (ID) 到 USART_TDR 中。

步骤 11. 等待发送完成。

步骤 12. 设置应答间隔：

a. 写应答等待时间长度到 USART_TOR 中。

b. 写 USART_CR1.TOE=1 启动软件定时。

c. 等待 USART_SR.TO=1。

步骤 13. 数据段 1：数据发送者可能是主控也可能是从机，具体由应用及受保护 ID 段参数决定，如果是主控发送数据，则写数据到 USART_TDR 中，等待发送完成 (USART_SR.TC=1)。

步骤 14. 数据段 1 的字节间间隔，参考步骤 9 执行。

步骤 15. 数据段 2：参考步骤 13 的数据段 1。

步骤 16. 数据段 2 的字节间间隔，参考步骤 14。

步骤 17.

步骤 18. 数据段 n：参考步骤 13 的数据段 1。

步骤 19. 数据段 n 的字节间间隔：参考步骤 14。

步骤 20. 校验和段（校验和由软件实现）：如果数据段是主控发送数据，则参考步骤 13。

步骤 21. 等待发送完成，LIN 操作结束，控制寄存器清零。



说明

必须设置校验位 USART_CR1.PCE=0，停止位 USART_CR1.STOP=0，USART_CR1.COMB=0。

18.3.6.4 LIN 接收

唤醒信号帧的检测

通过 I/O 端口的外部中断输入功能检测下降沿或设 USART_IER.DWK=1 产生中断来实现唤醒信号帧的检测；如果 LIN 处于激活状态则跳过此步。

同步间隔段的检测

通过写同步间隔段检测使能位 USART_CR3.LBDE=1 和写 USART_CR1.TOE=4 设置同步间隔段检测功能来实现同步间隔段的检测。

在 USART_TOR 中写入期望的低电平宽度（例如 11 个波特率位宽度，PCLK=48M，LIN 波特率 20k，USART_BRR.PSC=1，则 TOR=48M/20k/4*11=6600=0x19C8）。如果在 USART_CR3.LBDE=1 之后检测到 RXD 的下降沿，则内部定时器开始计数：

- 如果 RXD=0 期间计数器值等于 USART_TOR 值则硬件置 USART_SR.LBD=1，产生中断（如果 USART_IER.LBD=1）；
- 如果计数器值小于 USART_TOR 值时检测到 RXD 上升沿，则停止计数并回到空闲状态继续监测下一个同步间隔段的到来。

同步段的波特率测量

通过写波特率自动检测使能位 USART_CR3.ABRE=1 来启动 USART 内部定时器（定时器的计数时钟是 USART 的 PCLK）测量波特率：

硬件连续测量 RXD 管脚的 5 个下降沿间的周期，将测量到的 4 个周期值累加然后除以 8 即为通信波特率值，这一过程由硬件完成无需软件参与。

$$\text{波特率} = 1 / (t_{\text{PCLK}} * 4^{\text{PSC}} * (\text{BDR} + 1))$$

其中：

- t_{PCLK} ：模块工作时钟周期
- PSC：USART_BRR.PSC 值
- BDR：USART_BDR 值

测量完成后硬件将波特率值写入到 USART_BDR 中，同时硬件置位自动波特率检测标志位 USART_SR.ABR；在 USART_SR.ABR=0 时，不建议读取 USART_BDR 的值，软件可以置 USART_IER.ABR=1，当 USART_SR.ABR=1 后产生硬件中断。软件可以根据 USART_BDR 值调整当前 USART 的波特率以便更精确同步。波特率的设置请参见“[通信波特率编程](#)”。

LIN 从机设备软件操作步骤

- 步骤 1.** 将 USART_CR1 寄存器设定为复位值。
- 步骤 2.** 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，配置 TXD 和 RXD 为数字输入带上拉，将 TXD 和 RXD 映射到需要的管脚。
- 步骤 3.** 设定 USART_CR1 (除 RE)、USART_CR2、USART_CR3、USART_IER 寄存器及 USART_CR3.LINEN=1 (如果有步骤 5 则 USART_CR3.LINEN=0)。
- 步骤 4.** 设置 USART_BRR 的预分频和波特率初始值。
- 步骤 5.** (可选) 等待唤醒信号到来：设定 I/O 端口外部中断为下降沿检测或设 USART_IER.DWK=1，等待 CPU 被唤醒。
- 步骤 6.** 设置同步间隔段低电平宽度，例如期望 13 个波特率，则按 USART_TOR 寄存器中公式算出 TOR 值后写入 TOR 寄存器 (如果有步骤 5 则设 USART_CR3.LINEN=1)。
- 步骤 7.** 设定 USART_CR3.LBDE=1、USART_IER.LBD、USART_CR1.TOE=4 (同步间隔段测量) 和 USART_CR1.RE=1。如果 USART_IER.LBD=0 则至步骤 8，否则等中断发生后至步骤 8。
- 步骤 8.** 软件查询 USART_SR.LBD 值，如果是 1 则表示检测到同步间隔段，然后进入同步段测量。否则继续查询 USART_SR.LBD 值。
- 步骤 9.** 同步段波特率测量：设 USART_CR3.LBDE=0，设定 USART_IER.ABR 值和 USART_CR3.ABRE=1，USART_CR1.TOE=5。如果 USART_IER.ABR=0 则循环查询 USART_SR.ABR 值，如果 USART_IER.ABR=1 则等待中断发生后至步骤 10。
- 步骤 10.** 设 USART_CR3.ABRE=0，根据 USART_BDR 值，如果需要调整波特率则设 USART_CR1.RE=0，然后重设 USART_BRR，波特率的设置参考“[通信波特率编程](#)”；最后设 USART_CR1.RE=1；如果不需要调整波特率则跳至步骤 11。
- 步骤 11.** 接收受保护 ID 段，数据以及校验和，过程与异步串口接收数据过程相同：
 - a. 接收受保护 ID 段：查询 USART_SR.RC 值，如果为 1 则读取 USART_RDR 值并根据协议约定执行这个值所规定的操作；如果 USART_SR.RC=0 则继续查询。这步也可用中断完成。



说明

如果从机解析受保护 ID 段后被要求发送数据，则忽略下面两个步骤，设 USART_CR1.TE=1 并参考“[LIN 发送](#)”的步骤 13~21 操作。

- b. 接收数据：查询 USART_SR.RC 值，如果为 1 则读取 USART_RDR，否则继续查询。
- c. 接收校验和：查询 USART_SR.RC 值，如果为 1 则读取 USART_RDR，否则继续查询。收到校验和后，对之前收到的 N 个数据作校验，然后与收到的校验和比较，根据比较结果执行相应操作。

步骤 12. 接收结束，进入休眠状态，等待下次被唤醒。

18.4 操作示例

18.4.1 异步串口模式发送数据编程示例

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述, USART_CR2.TXINV=0 时配置 TXD 为数字输入带上拉, USART_CR2.TXINV=1 时配置 TXD 为数字输入带下拉, 将 TXD 映射到需要的管脚。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.USARTn (n=0/1) 为 1, 使能 USART 配置时钟及工作时钟。
- 步骤 3. 向 USART_CR1 写入 0x00, 使 USART 模块回复到初始态。
- 步骤 4. 配置 USART_CR2.SYNC 为 0, 使 USART 模块工作于异步串口模式。
- 步骤 5. 配置 USART_CR1.OVER8 及 USART_BRR 寄存器, 配置通信波特率。
- 步骤 6. 配置 USART_CR1.CHLEN, 配置帧内数据位的个数。
- 步骤 7. 配置 USART_CR1.STOP、USART_CR1.PCE 及 USART_CR1.PS, 配置停止位长度及奇偶校验方式。
- 步骤 8. 配置 USART_CR1.MSB, 配置发送数据时的数据序列顺序。
- 步骤 9. 若 USART 需要工作在单线半双工模式, 则设置 USART_CR1.COMB 和 USART_CR1.RE 为 1、USART_CR1.TE 为 0, 并跳转至步骤 11 继续执行; 若 USART 作为多处理器通信的从机, 则配置 TXD 为开漏输出。
- 步骤 10. 设置 USART_CR1.TE 为 1, 使能 TXD 发送功能。
- 步骤 11. 向 USART_ICR.TC 写入 0x00, 清除发送完成标志。
- 步骤 12. 向 USART_TDR 寄存器写入一帧待发送的数据。
- 步骤 13. 查询等待 USART_SR.TC 变为 1, 本次写入的数据帧发送完成。
- 步骤 14. 如待发送的数据未完成, 则跳转到步骤 11 继续执行; 如数据已发送完成且 USART 作为多处理器通信的从机, 设置 USART_CR1.TE 为 0。

18.4.2 异步串口模式接收数据编程示例

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述, 将 RXD 映射到需要的管脚; USART_CR2.RXINV=0 时配置 RXD 为数字输入带上拉, USART_CR2.RXINV=1 时配置 RXD 为数字输入带下拉, 以防止发送端未使能而接收端先使能的情况下 RXD 线悬空导致的误触发。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.USARTn (n=0/1) 为 1, 使能 USART 配置时钟及工作时钟。
- 步骤 3. 向 USART_CR1 写入 0x00, 使 USART 模块回复到初始态。
- 步骤 4. 配置 USART_CR2.SYNC 为 0, 使 USART 模块工作于异步串口模式。
- 步骤 5. 配置 USART_CR1.OVER8 及 USART_BRR 寄存器, 配置通信波特率。
- 步骤 6. 配置 USART_CR1.CHLEN, 配置帧内数据位的个数。
- 步骤 7. 配置 USART_CR1.STOP、USART_CR1.PCE 及 USART_CR1.PS, 配置停止位长度及奇偶校验方式。
- 步骤 8. 配置 USART_CR1.MSB, 配置接收数据时的数据序列顺序。
- 步骤 9. 配置 USART_CR1.SBS, 选择起始位的判定方式。
- 步骤 10. 向 USART_ICR 寄存器写入 0x00, 清除所有状态标志。
- 步骤 11. 设置 USART_CR1.RE 为 1。
- 步骤 12. 查询等待 USART_SR.RC 变为 1, 接收完成一帧数据。
- 步骤 13. 查询 USART_SR.PE、USART_SR.FE、USART_SR.OR, 以确认接收到的数据帧是否有效。若无效则进行出错处理; 若有效则从 USART_RDR 寄存器读出数据并保存。
- 步骤 14. 向 USART_ICR.RC 写入 0x00, 清除接收完成标志。
- 步骤 15. 如待接收的数据未完成, 则跳转到步骤 12 继续执行。

18.4.3 红外模式 1：发送数据设定

- 步骤 1. 将 USART_CR1 寄存器设定为复位值。
- 步骤 2. 初始化端口，均设置为数字输入带上拉；设定红外需要使用的管脚。
- 步骤 3. 设定 USART_CR1 中除 TE 以外的值，设定 USART_CR2，USART_CR3 寄存器。
- 步骤 4. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率。
- 步骤 5. 设定 USART_CR3.IREN=1 (USART_CR2.TXINV 根据需要配置)，USART_CR1.TE=1，设定 USART_IER.TXE=1。
- 步骤 6. 通过发送中断程序写通信数据到 USART_TDR，发送开始。
- 步骤 7. 如果需要连续发送数据时，重复步骤 6。
- 步骤 8. 通过发送中断程序写入最后一个发送数据，并写 USART_IER.TXE=0，USART_IER.TC=1，最后一个数据发送结束后，发生发送完成中断。
- 步骤 9. 写 USART_CR1.TE=0。

18.4.4 红外模式 1：接收数据设定

- 步骤 1. 将 USART_CR1 寄存器设定为复位值。
- 步骤 2. 初始化端口，均设置为输入带上拉；设定红外需要使用的管脚。
- 步骤 3. 设定 USART_IER.RC=1。
- 步骤 4. 设定 USART_CR1 中除 RE 以外的值，设定 USART_CR2，USART_CR3 寄存器。
- 步骤 5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率。
- 步骤 6. USART_CR3.IREN=1 (USART_CR2.RXINV 根据需要配置)，USART_CR1.RE=1。
- 步骤 7. 当检测到开始位时，内部同步后将数据接收到移位寄存器，检查校验位和停止位：
 - a. 停止位是低电平时，发生帧错误，接收到的数据传送到 USART_RDR 寄存器中，并发生接收帧错误中断，执行错误中断处理程序。
 - b. 发生数据上溢错误时，前帧数据被覆盖并发生数据上溢错误中断，执行错误中断处理程序。
 - c. 发生错误时，发生奇偶校验错误，接收到的数据被传送到 USART_RDR，并发生奇偶校验错误中断，执行错误中断处理程序。
 - d. 无错误发生时，接收到的数据传送到 USART_RDR 寄存器中，发生接收完成中断，通过中断程序将接收到的数据读取后重复步骤 7 连续接收数据。
- 步骤 8. 写 USART_CR1.RE=0。

18.5 寄存器

寄存器位访问权限说明：

- 1. USART_CR1.RE=0 并且 USART_CR1.TE=0 时，软件可以写所有的寄存器。
- 2. USART_CR1.RE 和 USART_CR1.TE 只要有一个为 1，软件可写的寄存器只有 USART_IER、USART_ICR、USART_TDR、USART_TOR、USART_BRR 和 USART_CR1.TOE、USART_CR1.SLME、USART_CR2.TXDMAEN、USART_CR2.RXDMAEN。
- 3. USART_CR1.RE 和 USART_CR1.TE 只要有一个为 1，USART_CR1 能被软件写入新值的条件是：新写入的值中 USART_CR1.TE=0 并且 USART_CR1.RE=0。
- 4. LIN 模式下 USART_BRR 寄存器值可以在 USART_CR1.TE 值为 1 时更新。
- 5. 对 USART_CR3 寄存器的读写没有限制。

18.5.1 寄存器总表

- USART0 基地址：0x4000A000
- USART1 基地址：0x4000A400

表 18-10 通用同步异步收发器 (USART) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	USART_CR1	控制寄存器 1	32
0x04	USART_CR2	控制寄存器 2	32
0x08	USART_CR3	控制寄存器 3	32
0x0C	USART_BRR	波特率寄存器	32
0x10	USART_TOR	定时计数寄存器	32
0x14	USART_SR	状态寄存器	32
0x18	USART_TDR	发送数据寄存器	32
0x1C	USART_IER	中断使能寄存器	32
0x20	USART_ICR	中断标志清除寄存器	32
0x24	USART_RDR	接收数据寄存器	32
0x28	USART_ADDR	地址寄存器	32
0x2C	USART_ADDMR	地址掩码寄存器	32
0x30	USART_BDR	波特率检出值寄存器	32
0xF0	USART_IR	红外调制寄存器	32

18.5.2 控制寄存器 1 (USART_CR1)

Offset	Bit Position																																																						
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																							
Reset	0x00000000																																																						
Name	Reserved																		SLME			MPE			COMB			SBS			TOE			OVER8			MSB			STOP			CHLEN			PCE			PS			TE			RE
Access																			RW			RW			RW			RW					RW							RW					RW			RW			RW			RW	

位/位域	标记	功能描述	读写
31:15	Reserved	所有位读出的 0，写入时必须写 0	-
14	SLME	多处理器通信时，静默模式使能位 0b0：正常模式 0b1：静默模式  说明 - SLME=1 时，MPB 位为 0 的数据不会从接收移位寄存器读取到接收数据寄存器，同时错误标志 PE 和 FE 位也不置位。接收到 MPB 为 1 的 ID 数据并且地址匹配时，SLME 自动清零，开始正常的接收数据动作。 - SLME 位只在异步串口模式多处理器通信时有效，其它模式时此位须设置为 0。	RW
13	MPE	多处理器通信功能使能位（仅适用于异步串口模式） 0b0：无效 0b1：有效  说明 - 非异步串口模式须设置为 0。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
12	COMB	异步串口模式时，收发端口复合使能位 0b0：TXD 和 RXD 分开使用 0b1：异步串口单总线模式，TXD 和 RXD 合用 TXD 端口  说明 - 非异步串口模式须设置为 0。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
11	SBS	异步串口模式接收数据时，起始位检测方式设定位 0b0：接收数据 RXD 管脚边沿（USART_CR2.RXINV=0 为下降沿，USART_CR2.RXINV=1 为上升沿）作为起始位 0b1：接收数据 RXD 管脚电平（USART_CR2.RXINV=0 为低电平，USART_CR2.RXINV=1 为高电平）作为起始位  说明 - 非异步串口模式须设置为 0。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW

位/位域	标记	功能描述	读写
10:8	TOE	多功能定时器工作模式配置位 ● 0b000: 停止定时计数器 ● 0b001: 通用定时功能 ● 0b010: 接收定时功能 ● 0b011: 数据帧间定时功能 ● 0b100: LIN 模式下, 同步间隔段计数使能 ● 0b101: LIN 模式或异步串口模式下, 波特率自动检测使能 ● 0b110~0b111: 保留, 同 0b000, 关闭计数功能  说明 ● 所有功能的 TOE 都由硬件清零。 ● 如果从一种功能 (如 TOE=0b010) 切换到另一种功能 (如 TOE=0b001), 则需要软件先写 TOE=0b000 后再写 TOE=0b001。	RW
7	OVER8	异步串口模式/LIN 模式时, 采样模式设定位 ● 0b0: 16 倍采样 ● 0b1: 8 倍采样  说明 ● 同步串口模式和智能卡模式须设置为 0, 红外模式设置无效。 ● 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
6	MSB	MSB/LSB 方式选择位 ● 0b0: LSB 方式 ● 0b1: MSB 方式  说明 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
5	STOP	停止位长度设定位 ● 0b0: 1 个停止位 ● 0b1: 2 个停止位  说明 ● 同步串口模式和 LIN 模式须设置为 0。 ● 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
4	CHLEN	发送/接收数据长度设定位 ● 0b0: 8 位 ● 0b1: 9 位  说明 ● 同步串口模式和 LIN 模式须设置为 0。 ● 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
3	PCE	奇偶校验有效位 ● 0b0: 无校验 ● 0b1: 校验  说明 ● 同步串口模式和 LIN 模式须设置为 0。 ● 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW

位/位域	标记	功能描述	读写
2	PS	奇偶校验选择位 ● 0b0: 偶校验 ● 0b1: 奇校验  说明 ● PS 位只在 PCE=1 时有效。 ● 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
1	TE	发送使能位 ● 0b0: 发送禁止 ● 0b1: 发送使能  说明 ● 同步串口模式下 (USART_CR2.SYNC=1)，TE 位只能在 TE=0 且 RE=0 (发送/接收禁止) 时写 1，如果是同步模式全双工，则 TE 和 RE 必须同时写 1；TE=1 或 RE=1 时，此位只能与 RE 位同时写 0；TE=1 和 RE=1 时，此位只能与 RE 位同时写 0。USART_CR2.SYNC=0 时不受限制。 ● 单线半双工工作时 TE 设置无效。	RW
0	RE	接收使能位 ● 0b0: 接收禁止 ● 0b1: 接收使能  说明 同步串口模式下 (USART_CR2.SYNC=1)，RE 位只能在 TE=0 且 RE=0 (发送/接收禁止) 时写 1，如果是同步模式全双工，则 RE 和 TE 必须同时写 1；TE=1 或 RE=1 时此位只能与 TE 位同时写 0；TE=1 和 RE=1 时此位只能与 TE 位同时写 0。USART_CR2.SYNC=0 时不受限制。	RW

18.5.3 控制寄存器 2 (USART_CR2)

Offset	Bit Position																																							
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																						CLKC		SWAP		SYNC		TXINV		RXINV		CTSEN		RTSEN		TXDMAEN		RXDMAEN	
Access																							RW		RW		RW		RW		RW		RW		RW		RW		RW	

位/位域	标记	功能描述	读写
31:10	Reserved	所有位读出为 0，写入时必须写 0	-
9:8	CLKC	异步串口模式/同步串口模式/智能卡模式时，时钟控制位 异步串口模式 0b00：时钟源为内部波特率生成器生成的时钟，时钟不输出到 CK 管脚，CK 管脚可以当作普通 IO 使用 0b01：时钟源为内部波特率生成器生成的时钟，时钟输出到 CK 管脚，输出时钟频率和波特率相同 0b10/0b11：禁止设置 同步串口模式 0b00/0b01：时钟源为内部波特率生成器生成的时钟，输出到 CK 管脚 0b10/0b11：时钟源为外部输入时钟，输入时钟的频率和波特率相同 智能卡模式 0b00：时钟源为内部波特率生成器生成的时钟，时钟不输出到 CK 管脚，CK 管脚可以当作普通 IO 使用 0b01：时钟源为内部波特率生成器生成的时钟，时钟输出到 CK 管脚 0b10/0b11：禁止设置  说明 - 红外模式和 LIN 模式设置无效，CK 都不能输出到 CK 管脚。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
7	SWAP	TXD/RXD 引脚互换控制位 0b0：不互换 0b1：互换  说明 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW

位/位域	标记	功能描述	读写
6	SYNC	同步模式选择位 0b0: 异步串口模式 0b1: 同步串口模式  说明 - 异步串口模式、智能卡模式和 LIN 模式须写入 0。 - 同步串口模式须写入 1。 - 红外模式设置无效。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
5	TXINV	输出信号反相控制位 0b0: 不反相 0b1: TXD 输出管脚信号反相  说明 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
4	RXINV	输入信号反相控制位 0b0: 不反相 0b1: RXD 输入管脚信号反相  说明 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
3	CTSEN	异步串口模式/同步串口模式时, CTS 功能使能位 0b0: 禁止 0b1: 使能  说明 - 非异步串口模式和非同步串口模式须设置为 0。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
2	RTSEN	异步串口模式/同步串口模式时, RTS 功能使能位 0b0: 禁止 0b1: 使能  说明 - 非异步串口模式和非同步串口模式须设置为 0。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW
1	TXDMAEN	TX DMAC 硬件握手信号使能位 0b0: 关闭硬件握手信号 0b1: 打开硬件握手信号  说明 单线半双工工作时不支持 DMA 模式。	RW
0	RXDMAEN	RX DMAC 硬件握手信号使能位 0b0: 关闭硬件握手信号 0b1: 打开硬件握手信号  说明 单线半双工工作时不支持 DMA 模式。	RW

18.5.4 控制寄存器 3 (USART_CR3)

Offset	Bit Position																																
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																IRMOD	IRCW	IREN	Reserved				BCN				SCEN	BREAK	ABRE	LBDE	LINEN	
Access																	RW	RW	RW					RW	RW				RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:16	Reserved	所有位读出于 0，写入时必须写 0	-
15	IRMOD	红外模式时，38K 调制方式 0b0：数据 0 时输出 38K 载波 0b1：数据 1 时输出 38K 载波  说明 非红外模式设置无效。	RW
14	IRCW	红外模式时，红外载波选择 0b0：3/16 模式 0b1：38K 载波模式  说明 非红外模式设置无效。	RW
13	IREN	红外模式时，红外模式使能位 0b0：红外功能模块关闭 0b1：红外功能模块开启  说明 非红外模式须设置为 0。	RW
12:10	Reserved	所有位读出于 0，写入时必须写 0	-
9:8	Reserved	-	-
7:5	BCN	智能卡模式时，设定一位数据传输期间的基本时钟数 0b000：16 0b001：32 0b010：64 0b011：128 0b100：186 0b101：256 0b110：372 0b111：512  说明 非智能卡模式设置无效。	RW

位/位域	标记	功能描述	读写
4	SCEN	智能卡模式时，模式使能位 ● 0b0：禁止智能卡模式 ● 0b1：使能智能卡模式  说明 非智能卡模式须设置为 0。	RW
3	BREAK	LIN 模式时，发送同步间隔段使能 ● 0b0：写入 USART_TDR 寄存器的值代表正常发送的数据 ● 0b1：写入 USART_TDR 寄存器的值代表需要发送同步间隔显性电平 的个数，该值应不小于 13，发送完成后自动硬件清零  说明 非 LIN 模式须设置为 0。	RW
2	ABRE	异步串口模式/LIN 模式时，自动波特率检测使能 ● 0b0：停止检测波特率 ● 0b1：启动检测波特率  说明 非异步串口模式和非 LIN 模式须设置为 0。	RW
1	LBDE	LIN 模式时，同步间隔段检测使能位 ● 0b0：禁止同步间隔段检测 ● 0b1：允许同步间隔段检测  说明 非 LIN 模式须设置为 0。	RW
0	LINEN	LIN 模式时，模式使能位 ● 0b0：禁止 LIN 模式 ● 0b1：使能 LIN 模式  说明 非 LIN 模式须设置为 0。	RW
1:0	Reserved	所有位读出为 0，写入时必须写 0	-

18.5.5 波特率寄存器 (USART_BRR)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000FF7F																															
Name	Reserved														PSC		DIVI						FBME		DIVF							
Access															RW		RW						RW		RW							

位/位域	标记	功能描述	读写
31:18	Reserved	-	-
17:16	PSC	内部时钟源时，预分频器选择位 0b00: PCLK 0b01: PCLK/4 0b10: PCLK/16 0b11: PCLK/64	RW
15:8	DIVI	整数分频寄存器	RW
7	FBME	内部波特率生成器选择时，小数波特率生成使能位 0b0: 禁止 0b1: 使能	RW
6:0	DIVF	小数分频寄存器 DIVF 仅在 FBME=1 时设定值有效；模块在 8M 及 8M 以下工作频率时建议开启小数分频。	RW

18.5.6 定时计数寄存器 (USART_TOR)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												TOR																			
Access																																

位/位域	标记	功能描述	读写
31:20	Reserved	所有位读出于 0，写入时必须写 0	-
19:0	TOR	定时计数时间 定时计数时间 = $t_{PCLK} * 4^{PSC} * (TOR + 1)$	RW

18.5.7 状态寄存器 (USART_SR)

Offset	Bit Position																																	
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000001																																	
Name	Reserved																		DWK	LBD	ABR	CTS	CTSV	AM	MPB	Reserved	TO	OR	FE	PE	RC	TC	TXE	
Access																			RO	RO	RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO	RO

位/位域	标记	功能描述	读写
31:15	Reserved	所有位读出为 0，写入时必须写 0	-
14	DWK	异步串口模式/红外模式/LIN 模式时，RXD 线唤醒标志 ● 0b0：未发生唤醒 ● 0b1：已发生唤醒 置位条件：硬件置 1。 清零条件：USART_ICR.DWK 位写入 0。	RO
13	LBD	LIN 模式时，同步间隔段检测标志 ● 0b0：未检测到同步间隔段 ● 0b1：已检测到同步间隔段 置位条件：硬件置 1。 清零条件：USART_ICR.LBD 位写入 0。	RO
12	ABR	异步串口模式/LIN 模式时，自动波特率检测标志 ● 0b0：检测未完成 ● 0b1：检测已完成 置位条件：硬件置 1。 清零条件：USART_ICR.ABR 位写入 0。	RO
11	CTS	异步串口模式/同步串口模式时，CTS 中断标志位 ● 0b0：CTS 信号没有发生翻转 ● 0b1：CTS 信号发生翻转 置位条件：硬件置 1。 清零条件：USART_ICR.CTS 位写入 0。	RO
10	CTSV	异步串口模式/同步串口模式时，CTS 管脚当前电平值 ● 0b0：CTS 管脚为低电平 ● 0b1：CTS 管脚为高电平	RO
9	AM	多处理器通信时，地址匹配标志 ● 0b0：地址不匹配 ● 0b1：地址匹配	RO
8	MPB	多处理器通信时，多处理器标志位 ● 0b0：接收数据为数据 ● 0b1：接收数据为 ID  说明 MPB 位只在多处理器通信时有效。	RO
7	Reserved	读出为 0，写入时必须写 0	-

位/位域	标记	功能描述	读写
6	TO	定时器匹配或溢出标志 ● 0b0: 定时器没有匹配或溢出 ● 0b1: 定时器已经匹配或溢出 置位条件: 硬件置 1。 清零条件: USART_ICR.TO 位写入 0。	RO
5	OR	上溢错误标志 ● 0b0: 没有上溢 ● 0b1: 发生上溢 置位条件: 接收数据寄存器有值的情况下, 又接收到新的数据。 清零条件: USART_ICR.OR 位写入 0。  说明 ● RE=0 并不能复位 OR 位。 ● OR=1 后接收到的数据会覆盖前面的数据。	RO
4	FE	异步串口模式/智能卡模式/红外模式/LIN 模式时, 接收帧错误标志 ● 0b0: 无接收帧错误 ● 0b1: 发生接收帧错误 置位条件: ● 异步串口模式/红外模式/LIN 模式: 接收的数据帧的停止位为 0。 ● 智能卡模式: 采样到低电平错误信号标志。 清零条件: USART_ICR.FE 位写入 0。  说明 ● RE=0 并不能复位 FE 位。 ● FE=1 后接收到的数据会覆盖前面的数据。	RO
3	PE	异步串口模式/智能卡模式/红外模式时, 接收数据校验错误标志 ● 0b0: 无校验错误 ● 0b1: 发生校验错误 置位条件: 接收数据发生奇偶校验错误。 清零条件: USART_ICR.PE 位写入 0。  说明 ● RE=0 并不能复位 PE 位。 ● PE=1 后接收到的数据会覆盖前面的数据。	RO
2	RC	数据接收完成标志 ● 0b0: 非接收完成状态 ● 0b1: 接收已完成 置位条件: 接收到停止位。 清零条件: ● USART_RDR 寄存器数据被软件读取后。 ● USART_ICR.PE 位写入 0。	RO

位/位域	标记	功能描述	读写
1	TC	发送完成标志位 ● 0b0：发送数据中 ● 0b1：发送数据完成 置位条件： ● 异步模式和时钟同步模式：一帧数据发送完成时（包括停止位），发送数据寄存器为空（TXE=1）。 ● 智能卡模式：最后 1 字节的数据送出后经过特定时间后，FE=0 而且发送数据寄存器为空（TXE=1）。 TC 置位的具体时序为：校验位送出后经过 2.5 位时间。 清零条件：USART_ICR.TC 位写入 0。  说明 同步模式下使用外部时钟进行通信时，不支持该标志位。	RO
0	TXE	发送数据寄存器空标志 ● 0b0：数据未传输到移位寄存器 ● 0b1：数据传输到移位寄存器 置位条件：数据传输到移位寄存器后硬件置 1。 清零条件：数据写入发送数据寄存器后硬件清 0。	RO

18.5.8 发送数据寄存器 (USART_TDR)

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000001FF																															
Name	Reserved																					MPID		TDR								
Access																						RW	RW									

位/位域	标记	功能描述	读写
31:10	Reserved	所有位读出于 0，写入时必须写 0	-
9	MPID	多处理器 ID 位 ● 0b0：发送数据 ● 0b1：发送 ID  说明 MPID 位只在多处理器通信时有效，其它模式须设定为 0。	RW
8:0	TDR	发送数据  说明 最高位 TDR[8]仅在异步串口模式而且数据长度设定为 9 位时有效。	RW

18.5.9 中断使能寄存器 (USART_IER)

Offset	Bit Position																																
0x1C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																	DWK	LBD	ABR	CTS	Reserved	AM	Reserved	TO	OR	FE	PE	RC	TC	TXE		
Access																		RW	RW	RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:15	Reserved	所有位读出于 0，写入时必须写 0	-
14	DWK	异步串口模式/红外模式/LIN 模式时，RXD 管脚边沿中断使能 0b0：禁止 0b1：使能  说明 USART_CR2.RXINV=0 为下降沿，USART_CR2.RXINV=1 为上升沿。	RW
13	LBD	LIN 模式时，同步间隔段中断使能 0b0：禁止 0b1：使能	RW
12	ABR	异步串口模式/LIN 模式时，自动波特率检测的中断使能 0b0：禁止 0b1：使能	RW
11	CTS	异步串口模式/同步串口模式时，CTS 信号翻转中断使能 0b0：禁止 0b1：使能	RW
10	Reserved	读出于 0，写入时必须写 0	-
9	AM	多机通信模式下，地址匹配后的中断使能 0b0：禁止 0b1：使能	RW
8:7	Reserved	所有位读出于 0，写入时必须写 0	-
6	TO	定时/计数器溢出中断使能 0b0：禁止 0b1：使能	RW
5	OR	接收数据寄存器未被读取又收到新数据而产生的错误中断使能 0b0：禁止 0b1：使能	RW
4	FE	异步串口模式/智能卡模式/红外模式/LIN 模式时，接收帧错误的中断使能 0b0：禁止 0b1：使能	RW

位/位域	标记	功能描述	读写
3	PE	异步串口模式/智能卡模式/红外模式时，一帧数据接收完后校验错误的中断使能 ● 0b0：禁止 ● 0b1：使能	RW
2	RC	一帧数据接收完成并写入接收数据寄存器后的中断请求使能位 ● 0b0：禁止 ● 0b1：使能	RW
1	TC	一帧数据发送完成的中断请求使能位 ● 0b0：禁止 ● 0b1：使能	RW
0	TXE	发送缓存为空状态的中断使能 ● 0b0：禁止 ● 0b1：使能	RW

18.5.10 中断标志清除寄存器 (USART_ICR)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00007A7E																															
Name	Reserved																	DWK	LBD	ABR	CTS	Reserved	AM	Reserved	TO	OR	FE	PE	RC	TC	Reserved	
Access																		WO	WO	WO	WO		WO		WO	WO	WO	WO	WO	WO	WO	

位/位域	标记	功能描述	读写
31:15	Reserved	所有位读出于 0，写入时必须写 0	-
14	DWK	DWK 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
13	LBD	LBD 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
12	ABR	ABR 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
11	CTS	CTS 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
10	Reserved	读出于 0，写入时必须写 0	-
9	AM	AM 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
8:7	Reserved	所有位读出于 0，写入时必须写 0	-
6	TO	TO 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
5	OR	OR 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
4	FE	FE 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
3	PE	PE 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
2	RC	RC 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
1	TC	TC 标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
0	Reserved	读出于 0，写入时必须写 0	-

18.5.11 接收数据寄存器 (USART_RDR)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								RDR							
Access																																

位/位域	标记	功能描述	读写
31:9	Reserved	所有位读出为 0，写入时必须写 0	-
8:0	RDR	接收数据  说明 最高位 RDR[8]仅在异步串口模式而且数据长度设定为 9 位时有效。	RO

18.5.12 地址寄存器 (USART_ADDR)

Offset	Bit Position																															
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								RW ADDR							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	所有位读出为 0，写入时必须写 0	-
7:0	ADDR	从机配置地址 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位	RW

18.5.13 地址掩码寄存器 (USART_ADDMR)

Offset	Bit Position																															
0x2C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																ADDM															
Access																	RW															

位/位域	标记	功能描述	读写
31:8	Reserved	所有位读出为 0，写入时必须写 0	-
7:0	ADDM	从机地址掩码 0b0：对应位不参与匹配 0b1：对应位参与匹配  说明 - 从机地址=USART_ADDR.ADDR & USART_ADDMR.ADDM。 - 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位。	RW

18.5.14 波特率检出值寄存器 (USART_BDR)

Offset	Bit Position																															
0x30	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																BDR															
Access																																

位/位域	标记	功能描述	读写
31:12	Reserved	所有位读出为 0，写入时必须写 0	-
11:0	BDR	异步串口模式/LIN 模式时，波特率检出值通过如下公式即可计算出通信波特率： $\text{Baud} = \frac{f_{\text{PCLK}}}{4^{\text{PSC}} \times (\text{BDR} + 1)}$  说明 仅适用于异步串口模式/LIN 模式。	RO

18.5.15 红外调制寄存器 (USART_IR)

Offset	Bit Position																															
0xF0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				DCNTH												Reserved				DCNTL											
Access					RW																RW											

位/位域	标记	功能描述	读写
31:27	Reserved	-	-
26:16	DCNTH	红外模式时，高电平计数值 38K 高电平维持时间 $t=t_{pCLK}*(DCNTH+1)$	RW
15:11	Reserved	-	-
10:0	DCNTL	红外模式时，低电平计数值 38K 低电平维持时间 $t=t_{pCLK}*(DCNTL+1)$	RW

19 I2C 总线 (I2C)

19.1 简介

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。

I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器 (I2C_STAT) 能反映 I2C 总线控制器和 I2C 总线的状态。

19.2 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准 (100kbps) /快速 (400kbps) /快速增强 (1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持三个从机地址
- 支持广播地址
- 支持中断状态查询功能

19.3 协议描述

I2C 总线使用连接设备的 SCL (串行时钟总线) 和 SDA (串行数据总线) 来传送信息。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节 (最高位 MSB 开始传输)，后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

19.3.1 I2C 总线上数据传输

通常标准 I2C 传输协议包含四个部分：起始 (S) 或重复起始信号 (Sr)，从机地址及读写位，传输数据，停止信号 (P)。

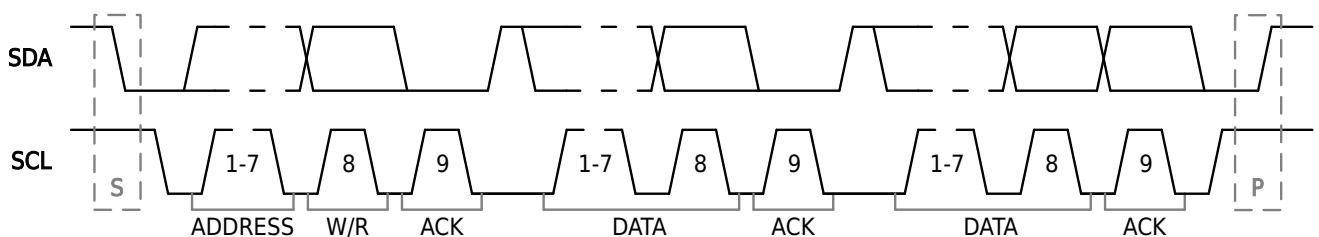


图 19-1 I2C 传输协议

- 起始信号、重复起始信号、停止信号

起始信号：当总线处于空闲状态下 (SCL 和 SDA 线同时为高)，SDA 线上出现由高到低的信号，即产生了起始信号。

重复起始信号：当两个起始信号之间没有停止信号时，即产生了重复起始信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信 (例如：从写入设备到从设备读出) 而不释放总线。

停止信号：当 SCL 线为高时，SDA 线上出现由低到高的信号，即产生了停止信号。主机向总线发出停止信号结束数据传送。

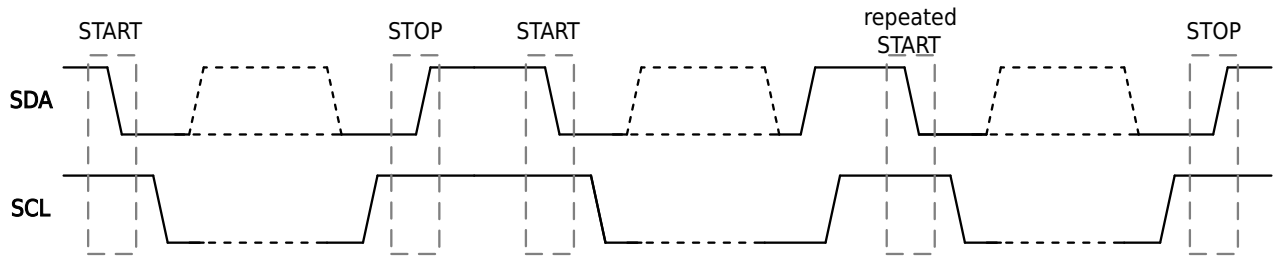


图 19-2 START、repeated START 和 STOP

- 从机地址及读写位
当起始信号产生后，主机立即传输数据的第一字节：7 位从机地址+读写位，读写位控制从机的数据传输方向（0：写；1：读）。被主机寻址的从机会在第 9 个 SCL 时钟周期的上升沿到来之前将 SDA 置为低电平作为应答。
- 传输数据
数据传输过程中，一个 SCL 时钟脉冲传输一个数据位，且 SDA 线只有在 SCL 为低时才可以改变。

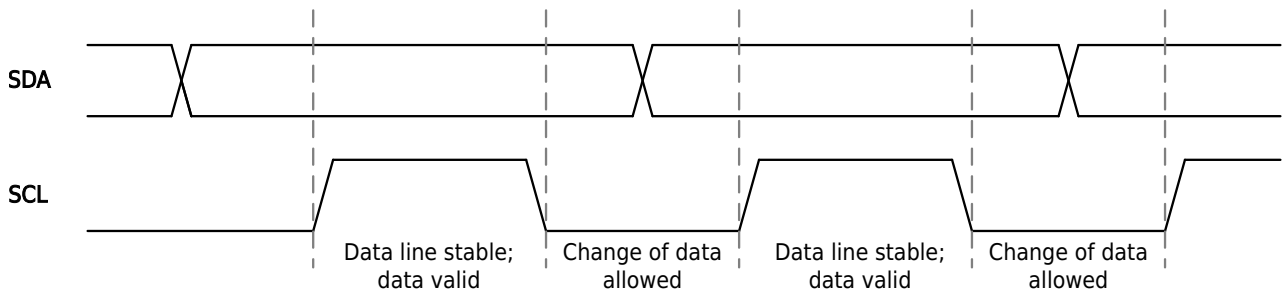


图 19-3 I2C 总线上位传输

19.3.2 I2C 总线上的应答信号

发送器件每传输一个字节之后，会在第九个 SCL 周期开始后释放 SDA，而接收器件则会在该 SCL 周期的上升沿之前产生一个应答信号，其中应答位有两种：

- ACK：为一个低电平信号，即接收器件在第九个 SCL 周期上升沿之前将 SDA 拉低，且在第九个 SCL 周期的高电平期间一直保持低电平。该信号表明接收端已成功接收到发送端的数据。
- NACK：为一个高电平信号，即接收器件在第九个 SCL 周期期间不对 SDA 做任何操作，SDA 一直保持高电平。当主机作为发送器件时，如果从机产生非应答信号（NACK），则主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。当主机作为接收器件时，如果主机产生非应答信号（NACK），从机将释放 SDA 线，之后由主机产生停止信号或重复起始信号。

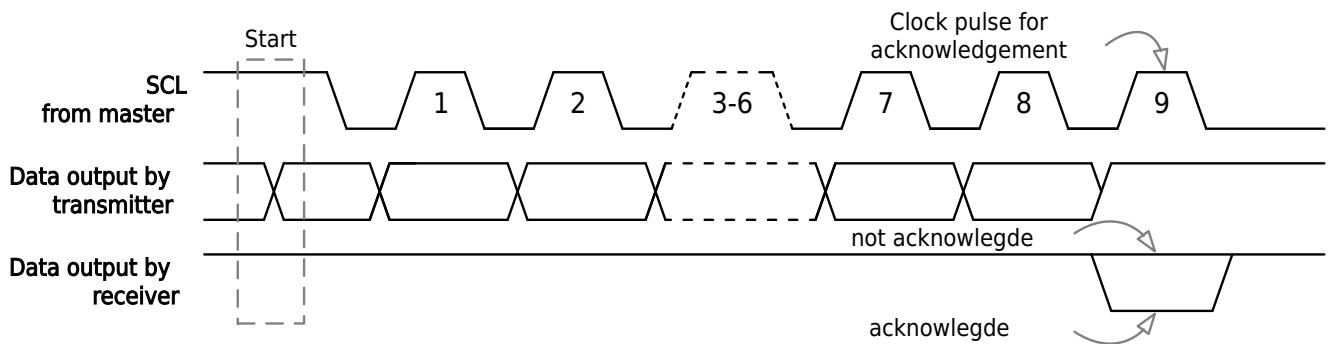


图 19-4 I2C 总线上应答信号

19.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：

- SCL 线上的同步 (时钟同步)

由于 I2C 总线具有线“与”的逻辑功能，SCL 线上只要有一个节点发送低电平，总线上就表现低电平。当所有的节点都发送高电平时，总线才能表现为高电平。所以，时钟低电平的时间由时钟低电平期最长的器件决定，而时钟的高电平时间由时钟高电平期最短的器件决定。由于 I2C 这种特性，当多个主机同时发送时钟信号时，在总线上表示的是统一的时钟信号。

当从机可以以较快的速率接收到一字节数据，但需要花费更多的时间来处理该数据时，从机可以在接收完数据并返回 ACK 后通过拉低 SCL 来暂停总线数据传输，主机将进入等待状态，直到从机处理完且释放 SCL 之后，主机才进行下一个字节的传输。该从机时钟延展功能是通过控制 I2C_CR.SI 来实现的，当接收到数据后，I2C_CR.SI 将会被硬件置 1，此时 SCL 被从机拉低，从机在处理完数据之后再将 I2C_CR.SI 清零以释放 SCL。

- SDA 线上的仲裁

SDA 线上的仲裁也是由于 I2C 总线具有线“与”的逻辑功能。主机在发送数据后，通过比较总线上的数据来决定是否退出竞争。丢失仲裁的主机立即切换到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。仲裁失败的主机继续输出时钟脉冲（在 SCL 上），直到发送完当前的串行字节。通过这种原理可以保证 I2C 总线在多个主机企图控制总线时保证数据的不丢失。

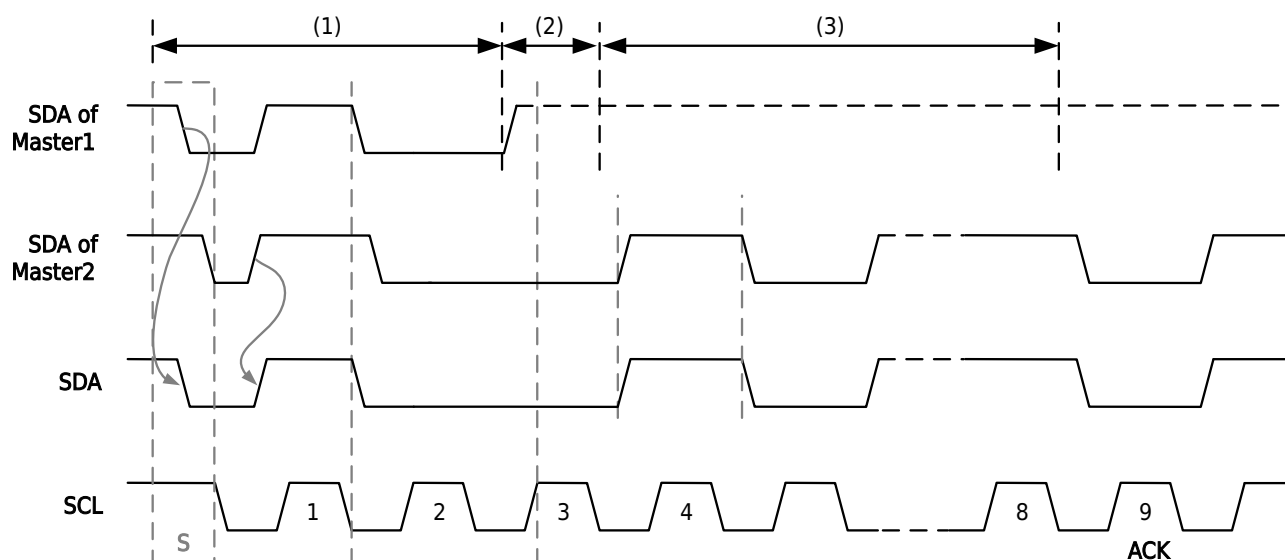


图 19-5 I2C 总线上的仲裁

上图中，触发仲裁的流程处理如下：

- ▶ 阶段（1）：主机 1 和主机 2 在发送完起始信号后，同时发送串行数据。
- ▶ 阶段（2）：第 3 个 SCL 周期低电平期间，主机 1 发送逻辑 1，主机 2 发送逻辑 0，主机 2 的逻辑 0 将拉低 SDA，使主机 1 的逻辑 1 被撤销，进而引起主机 1 仲裁丢失，进入未被寻址的从机状态。
- ▶ 阶段（3）：主机 1 仲裁失败进入未被寻址的从机状态后，仍产生时钟脉冲，直至发送完当前字节，且不再为下个字节的传输产生时钟脉冲。主机 2 赢得仲裁后，SDA 上的数据传输由主机 2 来启动。

19.4 功能说明

I2C 总线使用双线在连接到总线 SCL（串行时钟线）和 SDA（串行数据线）的设备间传送信息。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。由于只有无方向端口，I2C 需要将使用到的引脚配置为开漏输出。每个连接到总线的设备都能通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。I2C 总线状态能在状态寄存器中进行查询。

19.4.1 功能框图

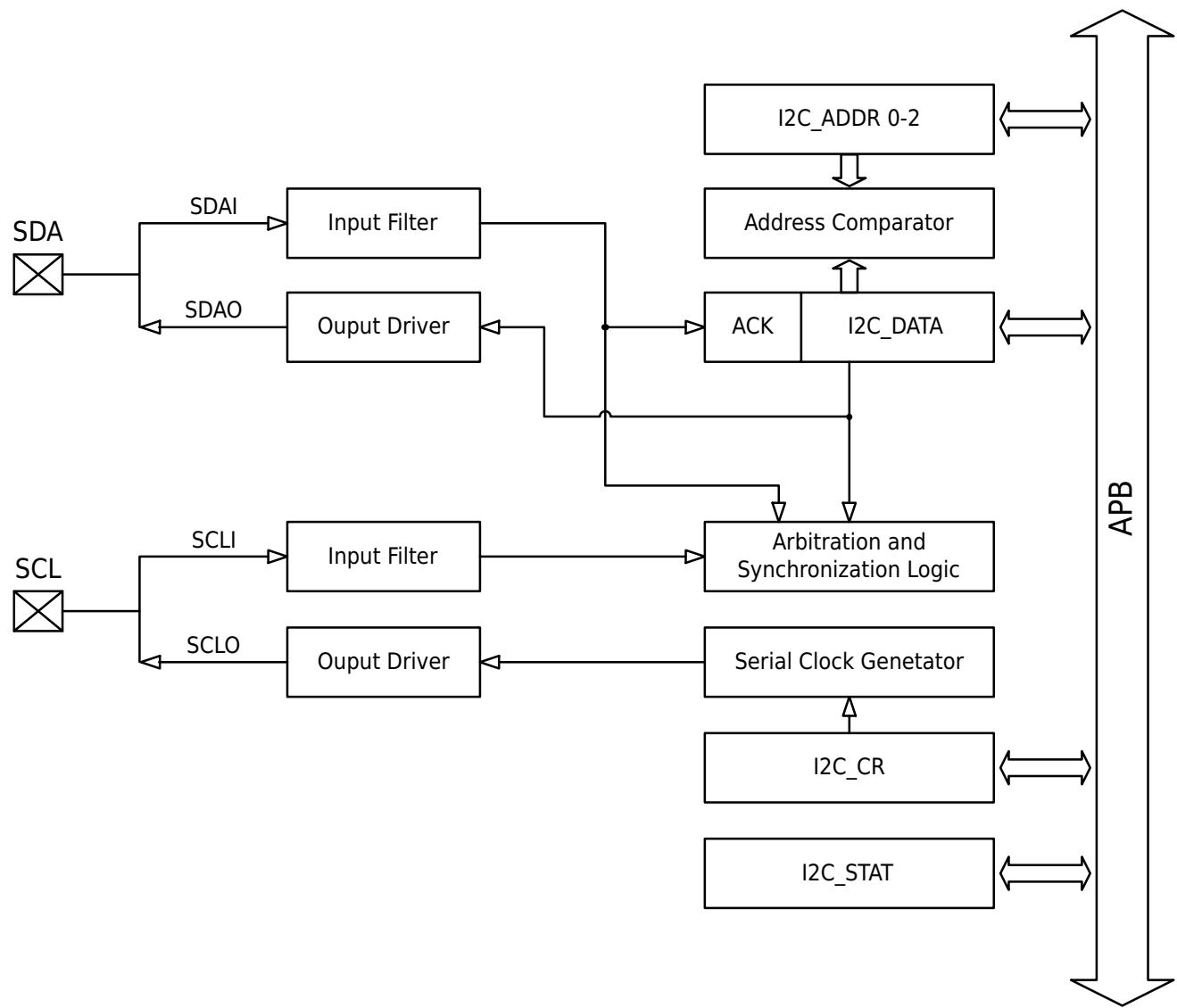


图 19-6 I2C 功能模块图

19.4.2 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 PCLK 信号的频率关系为 $f_{SCL} = f_{PCLK} / 8 / (I2C_TM + 1)$ ，其中 I2C_TM 应大于 0。
下表列出了 PCLK 频率与 I2C_TM 组合时，SCL 信号的输出频率值。

表 19-1 I2C 时钟信号波特率

PCLK (kHz)	I2C_TM						
	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

19.4.3 输入滤波器

输入信号与 PCLK 同步，低于 PCLK 周期的尖峰脉冲信号会被滤除。通过调整 I2C_CR.H1M 的值可改变滤波的强度。

- 作为主机时，如果 I2C_TM 的值小于等于 9，应设置 I2C_CR.H1M 为 1，进行简单滤波；如果 I2C_TM 的值大于 9，应设置 I2C_CR.H1M 为 0，进行高级滤波。
- 作为从机时，如果 PCLK 与 SCL 频率比值小于等于 30 时，应设置 I2C_CR.H1M 为 1，进行简单滤波；如果 PCLK 与 SCL 频率比值大于 30 时，应设置 I2C_CR.H1M 为 0，进行高级滤波。

19.4.4 地址比较器

I2C 地址比较器将接收到的 7 位地址与三个从机地址进行比较。根据 I2C_ADDR0.GC 位的配置，当 GC 配置为 0 时，首次接收到的 8 位字节高 7 位和 I2C_ADDRx 寄存器的 7 位 Addr 值比较；当 GC 配置为 1 时，首次接收到的 8 位字节高 7 位和广播地址 (0x00) 比较。如果任何一者相同，I2C_CR.SI 位会被置 1 并产生一个中断请求。查询 I2C_MATCH 寄存器，以获取匹配成功的地址为序号。

19.4.5 应答标志位

I2C_CR.AA 为应答控制位。当 AA 位为 1 时，I2C 模块收到数据后回应该应答信号 (ACK)；当 AA 位为 0 时，I2C 模块收到数据后回应非应答信号 (NACK)。

19.4.6 中断产生器

I2C_CR.SI 为中断标志，每当状态寄存器 (I2C_STAT) 的值产生变化时 (除变为 0xF8 外)，SI 标志位都会被置 1。当产生中断时，通过查询状态寄存器 (I2C_STAT) 可获知 I2C 总线的状态，以确定中断的实际来源。为了进行下一步的操作，SI 标志位必须通过软件清零。

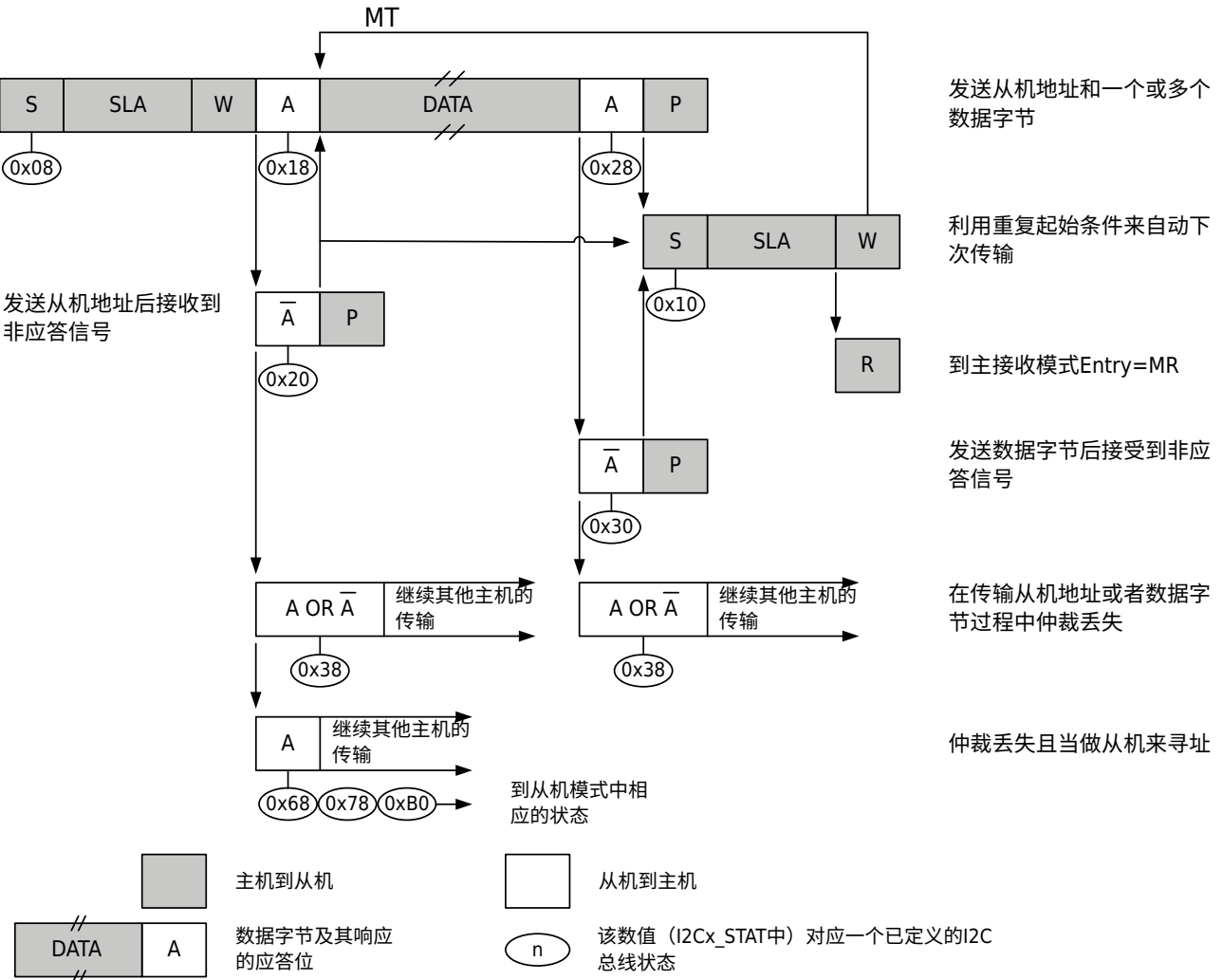
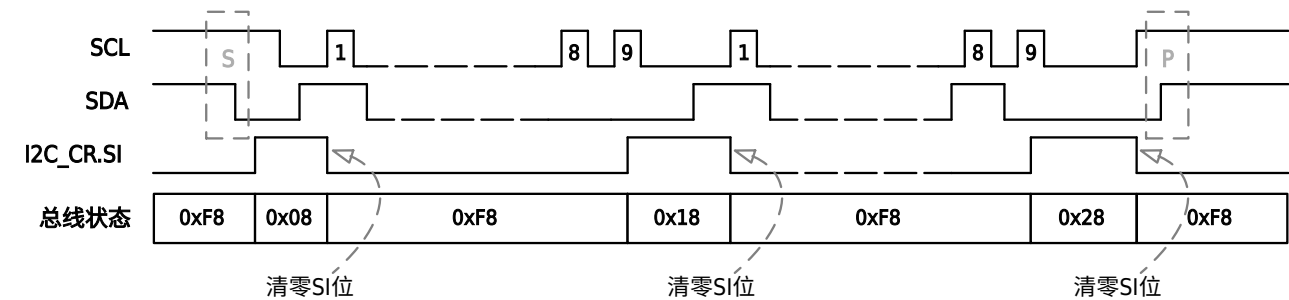
19.4.7 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100kbps 而在快速模式下可达 400kbps，在快速增强模式下可达 1Mbps，并且可以在四种模式下工作：主机发送模式、主机接收模式、从机接收模式、从机发送模式。还有一种特殊模式广播呼叫模式，其操作方式与从机接收模式类似。

● 主机发送模式

主机发送多个字节到从机，主机产生时钟，所以需要在 I2C_TM 中填入设定值。主机发送模式中需要将 I2C_CR.STA 置 1。当总线空闲时，主机发起一个起始位 START。若成功 I2C_CR.SI 被置 1。接下来把从机地址和写位 (SLA+W) 写入 I2C_DATA 中，清零 SI 位后，总线上发出 SLA+W。

主机发出 SLA+W 收到从机应答信号 (ACK) 后，SI 被置 1。接下来根据用户定义格式发送数据。所有数据发送完后，将 I2C_CR.STO 置 1，清零 SI 位后发出 STOP 信号，也可以发送重复起始信号进行新一轮数据传输。



● 主机接收模式

主机接收模式，由从机传输数据。初始化设置与主机发送模式相同，主机发送起始位以后，I2C_DATA 应写入从机地址和“读位”(SLA+R)。收到从机应答信号 (ACK) 后 I2C_CR.SI 被置 1。SI 清 0 后开始接收从机数据，若 I2C_CR.AA 为 1，主机收到数据后回应答信号 (ACK)；若为 0 主机收到数据后回非应答信号 (NACK)，之后主机可以发停止信号或重复起始信号开始下一轮的数据传输。

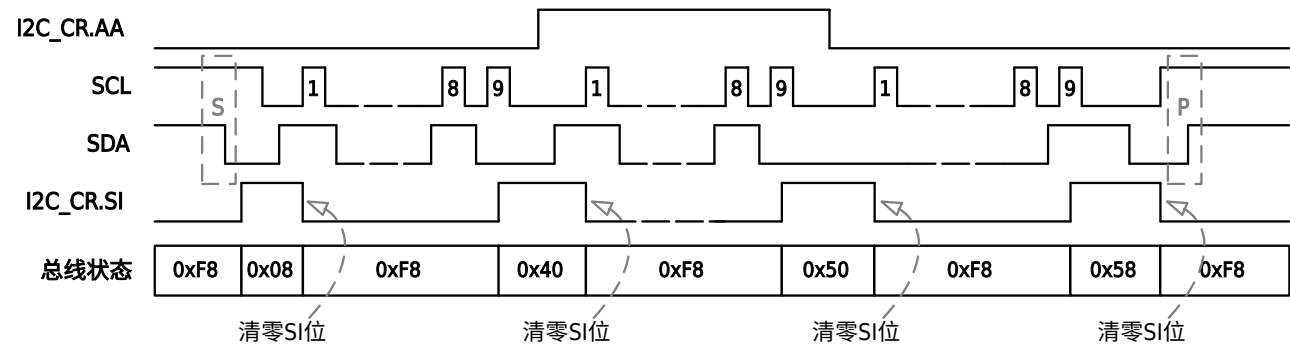


图 19-9 主机接收模式数据同步图

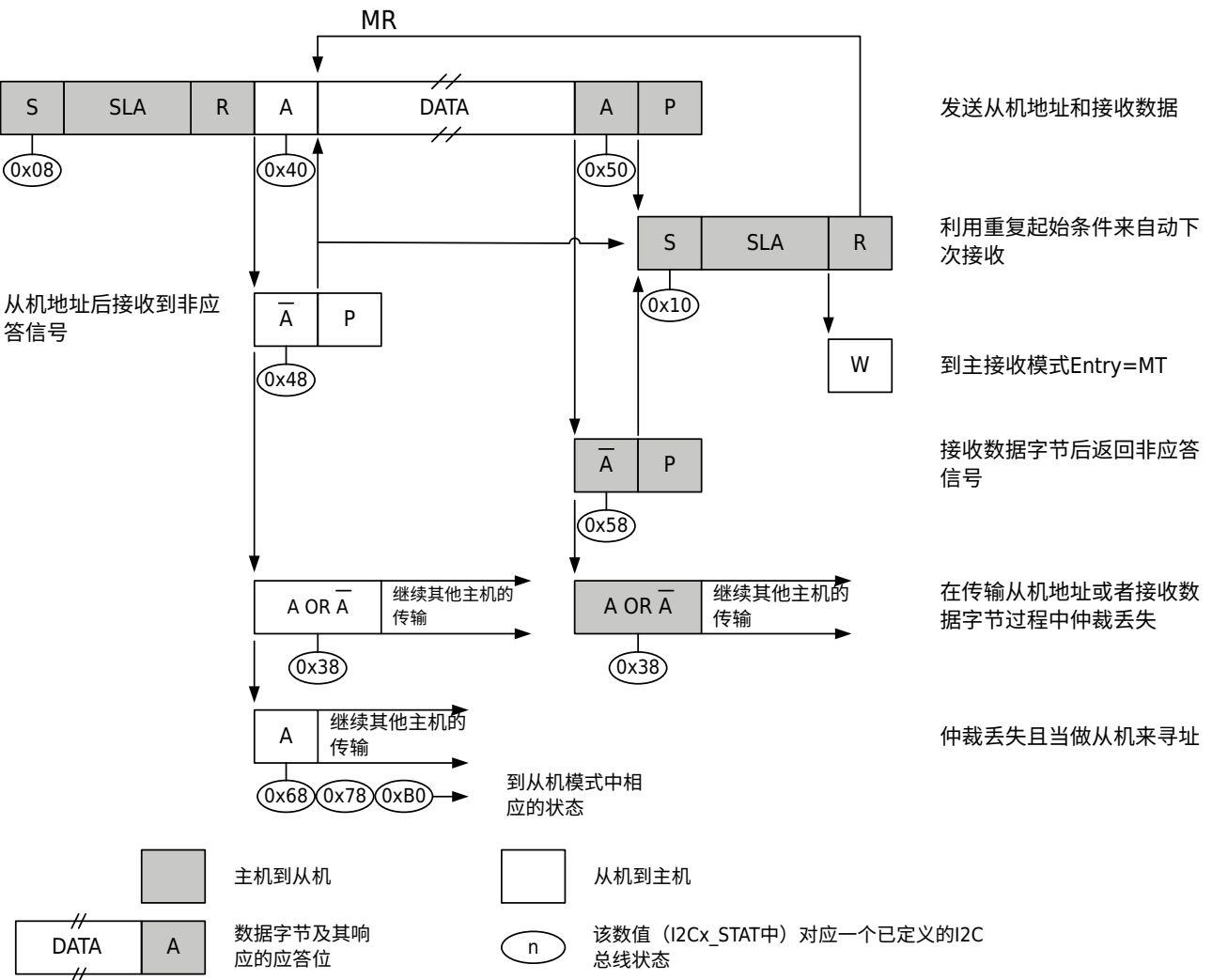


图 19-10 I2C 主机接收状态图

● 从机接收模式

在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2C_ADDRx 应写入从机地址，I2C_CR.AA 置 1 用以响应主机的寻址。上述初始化后，从机进入空闲模式，等待“写”信号（SLA+W）。若主机仲裁失败，也会直接进入未被寻址的从机接收模式。

当从机被“写”信号 SLA+W 寻址到后，需要清零 SI 位，以便从主机接收数据。如果在传输过程中 I2C_CR.AA=0，从机将在接收完下一字节后返回非应答信号（NACK），从机也将转为未寻址从机，与主机联系终止，不再接收数据，且 I2C_DATA 保持之前接收到的数据。从机地址识别可通过置位 AA 来恢复，这意味着 AA 位可临时将 I2C 模块从 I2C 总线上分离出来。

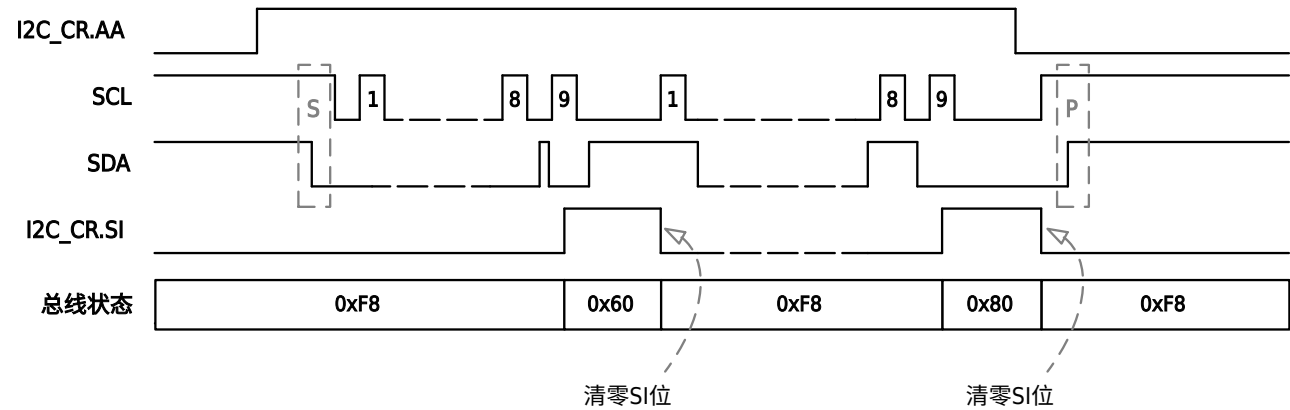


图 19-11 从接收模式数据同步图

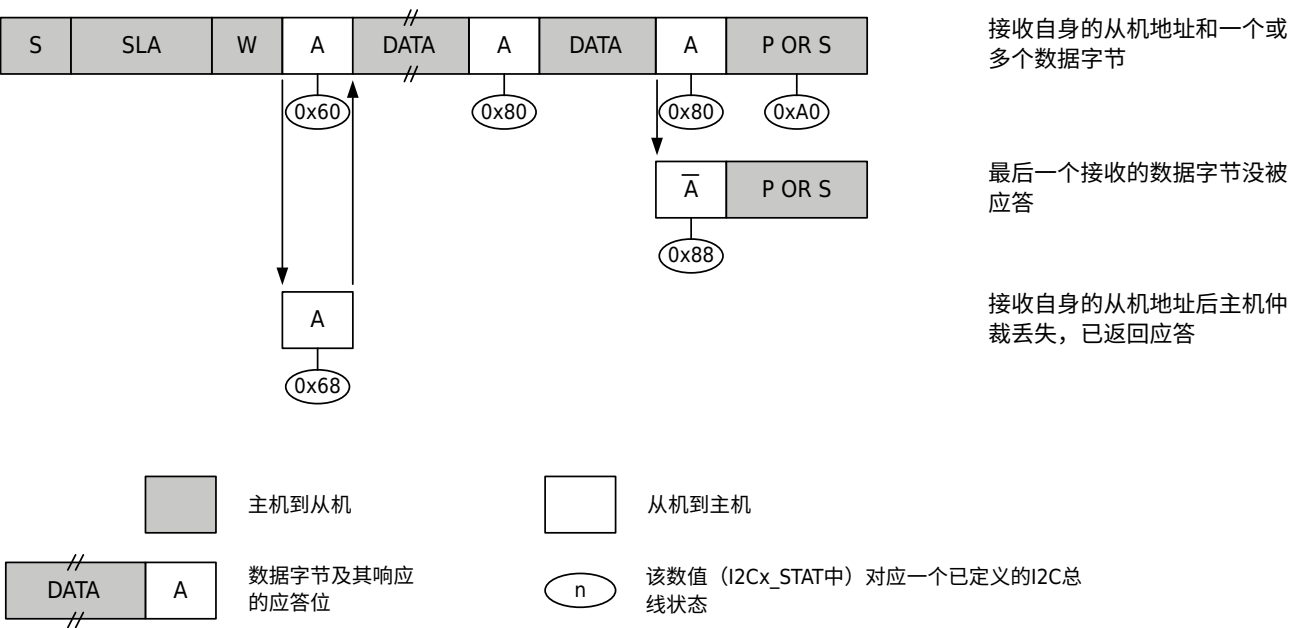


图 19-12 从机接收状态图

● 从机发送模式

从机发送模式，数据由从机发送给主机。当初始化 I2C_ADDRx 及 I2C_CR.AA 值后，器件等待直到自身地址被“读”信号 (SLA+R) 寻址。若主机仲裁失败，也可进入未被寻址的从机发送模式。

当从机被“读”信号 SLA+R 寻址，需要将 SI 清零用以向主机发送数据。通常主机接收每字节数据后会返回应答信号 (ACK)。

如果传输过程中主机的 I2C_CR.AA 清零，从机将不再发送数据，并释放 SDA，且将自身变为未寻址从机。

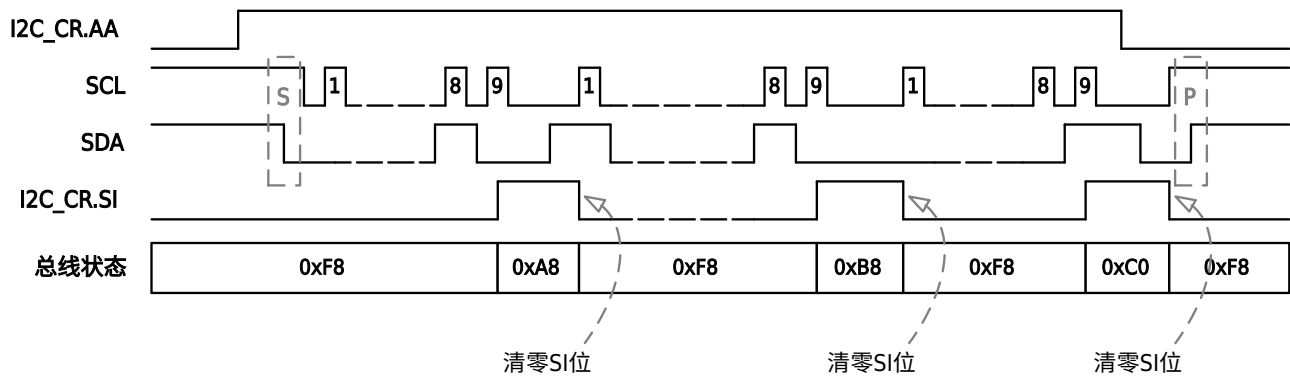


图 19-13 从机发送模式数据同步图

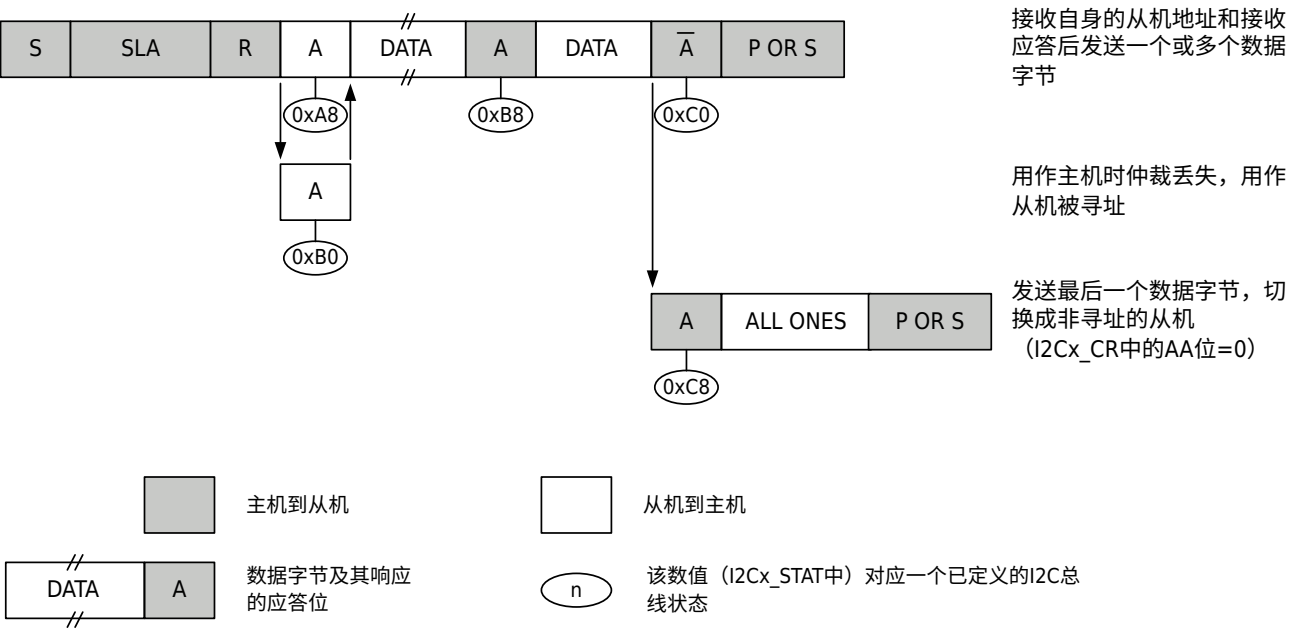


图 19-14 I2C 从机发送状态图

● 广播呼叫模式

广播呼叫模式是一种特殊的从机接收模式，寻址方式为 0x00，从机地址和读写都为 0。当 I2C_ADDR0.GC 及 I2C_CR.AA 都置为 1，使能接收广播呼叫模式。在该模式下 I2C_STAT 值与普通从机接收模式 I2C_STAT 值不同。仲裁失败也可能进入广播呼叫模式。

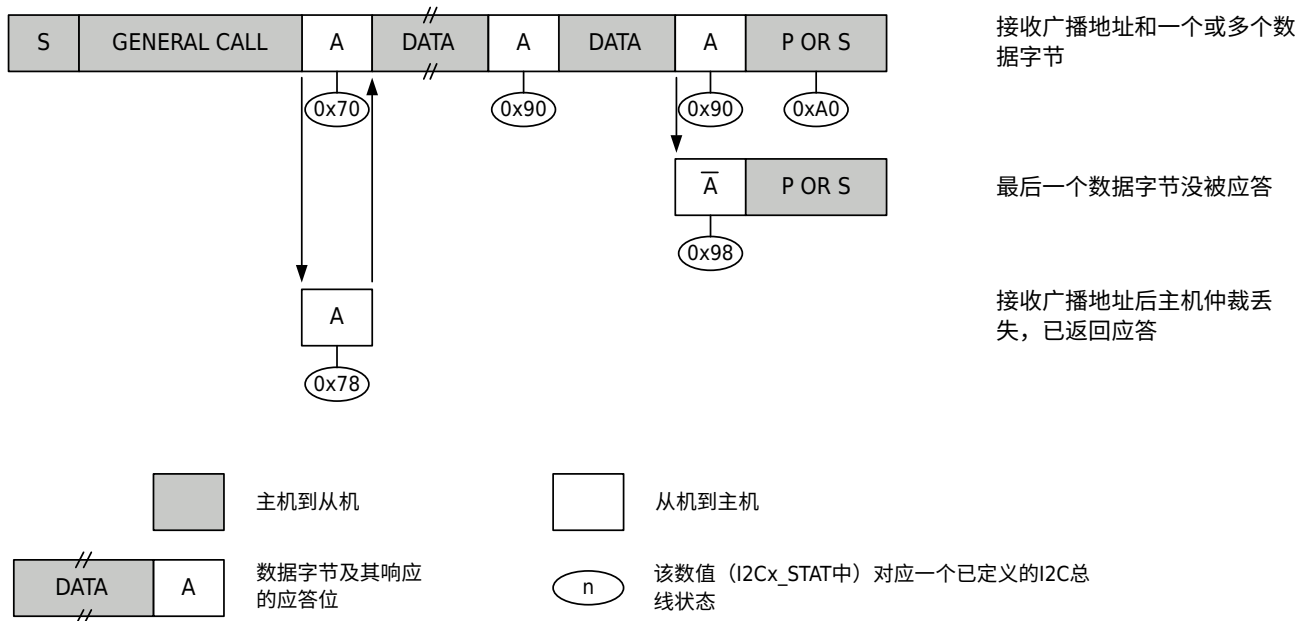


图 19-15 I2C 广播呼叫状态图

19.4.8 状态码表述

I2C 状态寄存器中有两种特殊的状态:

- 0xF8: 这个状态码表示没有任何可用的相关信息，因为串行中断标志“SI”还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。
- 0x00: 该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时 SI 置位。

表 19-2 I2C 状态码表述

状态代码	描述
主发送模式	
0x08	已发送起始条件
0x10	已发送重复起始条件
0x18	已发送 SLA+W，已接收 ACK
0x20	已发送 SLA+W，已接收 NACK
0x28	已发送 I2C_DATA 中的数据，已接收 ACK
0x30	已发送 I2C_DATA 中的数据，已接收 NACK
0x38	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	

状态代码	描述
0x08	已发送起始条件
0x10	已发送重复起始条件
0x38	在 NACK 中丢失仲裁 (即主机在回复 ACK 时发现 SDA 已经被其他器件拉低)
0x40	已发送 SLA+R, 已接收 ACK
0x48	已发送 SLA+R, 已接收 NACK
0x50	已接收数据字节, ACK 已返回
0x58	已接收数据字节, NACK 已返回
从接收模式	
0x60	已接收自身的 SLA+W, 已返回 ACK
0x68	主控时在 SLA+读写丢失仲裁, 已接收自身的 SLA+W, 已返回 ACK
0x80	前一次寻址使用自身从地址, 已接收数据字节, 已返回 ACK
0x88	前一次寻址使用自身从地址, 已接收数据字节, 已返回 NACK
0xA0	静态寻址时, 接收到停止条件或重复起始条件
从发送模式	
0xA8	已接收自身的 SLA+R, 已返回 ACK
0xB0	当主机时丢失仲裁, 已接收自身 SLA+R, 已返回 ACK
0xB8	已发送数据, 已接收 ACK
0xC0	已发送数据字节, 已接收 NACK
0xC8	装入的数据字节已被发送, 已接收 ACK
广播呼叫模式	
0x70	已接收广播地址 (0x00), 已返回 ACK
0x78	主控时在 SLA+读写丢失仲裁, 已接收广播地址, 已返回 ACK
0x90	前一次寻址使用广播地址, 已接收数据字节, 已返回 ACK
0x98	前一次寻址使用广播地址, 已接收数据字节, 已返回 NACK
0xA0	静态寻址时, 接收到停止条件或重复起始条件
其余杂项状态	
0xF8	无可用的相关状态信息, SI=0
0x00	传输过程出现总线错误, 或外部干扰使 I2C 进入未定义的状态

19.5 操作示例

19.5.1 主机发送示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述, 将 SCL、SDA 映射到需要的管脚; 并配置 SCL、SDA 管脚为开漏输出模式。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.I2Cx 为 1, 使能 I2C 工作时钟。
- 步骤 3. 向 SYSCTRL 模块寄存器 PERI_RESET0.I2Cx 依次写入 0、1, 复位 I2C 模块。
- 步骤 4. 配置 I2C_TM, 使 SCL 的时钟速率符合应用需求。
- 步骤 5. 设置 I2C_TMRUN 为 1, 使能 SCL 时钟发生器。
- 步骤 6. 设置 I2C_CR.ENS 为 1, 使能 I2C 模块。
- 步骤 7. 设置 I2C_CR.STA 为 1, 总线尝试发送 Start 信号。
- 步骤 8. 等待 I2C_CR.SI 变为 1, Start 信号已发送到总线上, 设置 I2C_CR.STA 为 0。
- 步骤 9. 查询 I2C_STAT, 如果该寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理。
- 步骤 10. 向 I2C_DATA 中写入 SLA+W, 设置 I2C_CR.SI 为 0, 发送 SLA+W。
- 步骤 11. 等待 I2C_CR.SI 变为 1, SLA+W 已发送到总线上。
- 步骤 12. 查询 I2C_STAT, 如果该寄存器值为 0x18, 继续执行下一步骤。否则进行出错处理。
- 步骤 13. 向 I2C_DATA 写入待发送的数据, 设置 I2C_CR.SI 为 0, 发送数据。
- 步骤 14. 等待 I2C_CR.SI 变为 1, 数据已发送到总线上。
- 步骤 15. 查询 I2C_STAT, 如果该寄存器值为 0x28, 继续执行下一步骤。否则进行出错处理。
- 步骤 16. 如待发送的数据未完成, 则跳转到步骤 13 继续执行。
- 步骤 17. 设置 I2C_CR.STO 为 1, 设置 I2C_CR.SI 为 0, 总线尝试发送 Stop 信号。

19.5.2 主机接收示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述, 将 SCL、SDA 映射到需要的管脚; 并配置 SCL、SDA 管脚为开漏输出模式。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.I2Cx 为 1, 使能 I2C 工作时钟。
- 步骤 3. 向 SYSCTRL 模块寄存器 PERI_RESET0.I2Cx 依次写入 0、1, 复位 I2C 模块。
- 步骤 4. 配置 I2C_TM, 使 SCL 的时钟速率符合应用需求。
- 步骤 5. 设置 I2C_TMRUN 为 1, 使能 SCL 时钟发生器。
- 步骤 6. 设置 I2C_CR.ENS 为 1, 使能 I2C 模块。
- 步骤 7. 设置 I2C_CR.STA 为 1, 总线尝试发送 Start 信号。
- 步骤 8. 等待 I2C_CR.SI 变为 1, Start 信号已发送到总线上, 设置 I2C_CR.STA 为 0。
- 步骤 9. 查询 I2C_STAT, 如果寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理。
- 步骤 10. 向 I2C_DATA 写入 SLA+R, 设置 I2C_CR.SI 为 0, 发送 SLA+R。
- 步骤 11. 等待 I2C_CR.SI 变为 1, SLA+R 已发送到总线上。
- 步骤 12. 查询 I2C_STAT, 如果寄存器值为 0x40, 继续执行下一步骤, 否则进行出错处理。
- 步骤 13. 设置 I2C_CR.AA 为 1, 使能应答标志。
- 步骤 14. 设置 I2C_CR.SI 为 0, 从机发送数据, 主机根据 I2C_CR.AA 发送 ACK 或 NACK。
- 步骤 15. 等待 I2C_CR.SI 变为 1, 从 I2C_DATA 读取已接收到的数据。
- 步骤 16. 查询到 I2C_STAT, 如果该寄存器值为 0x50 或 0x58, 继续执行下一步骤, 否则进行出错处理。
- 步骤 17. 如果待接收的数据只差最后一个字节, 设置 I2C_CR.AA 为 0, 使能非应答标志。
- 步骤 18. 如待接收的数据未完成, 则跳转到步骤 14 继续执行。

步骤 19. 设置 I2C_CR.STO 为 1，设置 I2C_CR.SI 为 0，总线尝试发送 Stop 信号。

19.5.3 从机发送示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.I2Cx 为 1，使能 I2C 工作时钟。
- 步骤 3. 向 SYSCTRL 模块寄存器 PERI_RESET0.I2Cx 依次写入 0、1，复位 I2C 模块。
- 步骤 4. 设置 I2C_CR.ENS 为 1，使能 I2C 模块。
- 步骤 5. 配置 I2C_ADDRx 为从机地址。
- 步骤 6. 设置 I2C_CR.AA 为 1，使能应答标志。
- 步骤 7. 等待 I2C_CR.SI 变为 1，被 SLA+R 寻址。
- 步骤 8. 查询 I2C_STAT，如果该寄存器的值为 0xA8，继续执行下一步骤，否则进行出错处理。
- 步骤 9. 向 I2C_DATA 写入待发送的数据，设置 I2C_CR.SI 为 0，发送数据。
- 步骤 10. 等待 I2C_CR.SI 变为 1，数据已发送到总线上。
- 步骤 11. 查询 I2C_STAT，如果该寄存器的值为 0xB8 或 0xC0 时，继续执行下一步骤，否则进行出错处理。
- 步骤 12. 如待发送的数据未完成，则跳转到步骤 9 继续执行。
- 步骤 13. 设置 I2C_CR.AA 为 0，设置 I2C_CR.SI 为 0。

19.5.4 从机接收示例

操作步骤

- 步骤 1. 参考《端口控制器 (GPIO)》章节端口数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.I2Cx 为 1，使能 I2C 工作时钟。
- 步骤 3. 向 SYSCTRL 模块寄存器 PERI_RESET0.I2Cx 依次写入 0、1，复位 I2C 模块。
- 步骤 4. 设置 I2C_CR.ENS 为 1，使能 I2C 模块。
- 步骤 5. 配置 I2C_ADDRx 为从机地址。
- 步骤 6. 设置 I2C_CR.AA 为 1，使能应答标志。
- 步骤 7. 等待 I2C_CR.SI 变为 1，被 SLA+W 寻址。
- 步骤 8. 查询 I2C_STAT，如果该寄存器值为 0x60，继续执行下一步骤，否则进行出错处理。
- 步骤 9. 设置 I2C_CR.SI 为 0，主机发送数据，从机根据 I2C_CR.AA 返回 ACK 或 NACK。
- 步骤 10. 等待 I2C_CR.SI 变为 1，从 I2C_DATA 中读取已接收到的数据。
- 步骤 11. 查询 I2C_STAT，如果该寄存器值为 0x80，继续执行下一步骤，否则进行出错处理。
- 步骤 12. 如待接收的数据未完成，则跳转到步骤 9 继续执行。
- 步骤 13. 设置 I2C_CR.AA 为 0，设置 I2C_CR.SI 为 0。

19.6 寄存器

19.6.1 寄存器总表

- I2C0 基地址：0x40000400
- I2C1 基地址：0x40008400

表 19-3 I2C 总线 (I2C) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	I2C_TMRUN	I2C 波特率发生器使能寄存器	32
0x04	I2C_TM	I2C 波特率计数器配置寄存器	32
0x08	I2C_CR	I2C 配置寄存器	32
0x0C	I2C_DATA	I2C 数据寄存器	32
0x14	I2C_STAT	I2C 状态寄存器	32
0x10	I2C_ADDR0	I2C 从机地址 0 寄存器	32
0x20	I2C_ADDR1	I2C 从机地址 1 寄存器	32
0x24	I2C_ADDR2	I2C 从机地址 2 寄存器	32
0x28	I2C_MATCH	I2C 从机地址匹配寄存器	32

19.6.2 波特率发生器使能寄存器 (I2C_TMRUN)

Offset	Bit Position																															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	TME	波特率发生器使能控制 ● 0b0：禁止 ● 0b1：使能	RW

19.6.3 波特率计数器配置寄存器 (I2C_TM)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	TM	I2C 总线 SCL 波特率配置 $f_{SCL} = f_{PCLK}/8/(TM+1)$ ，其中 $TM>0$	RW

19.6.4 配置寄存器 (I2C_CR)

Offset	Bit Position																																							
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																										ENS		STA		STO		SI		AA		Reserved		H1M	
Access																											RW		RW		RW		RW		RW				RW	

位/位域	标记	功能描述	读写
31:7	Reserved	-	-
6	ENS	模块使能控制 0b0: 禁止 0b1: 使能	RW
5	STA	总线状态控制 0b0: 无功能 0b1: 向总线发送 START  说明 START 发送完成后, 用户程序应设置该位为 0。	RW
4	STO	总线状态控制 0b0: 无功能 0b1: 向总线发送 STOP  说明 STOP 发送完成后硬件自动设置该位为 0。	RW
3	SI	I2C 中断标志 写 0 清除 I2C 中断标志并使状态机执行下一个动作, 写 1 对该位的值无影响 读取: 0b0: 未发生 I2C 中断 0b1: 已发生 I2C 中断	RW
2	AA	应答控制 0b0: 在应答阶段发送 NAK 0b1: 在应答阶段发送 ACK	RW
1	Reserved	-	-
0	H1M	I2C 滤波参数配置 0b0: 高级滤波, 更高的抗干扰性能 0b1: 简单滤波, 更快的通信速率  说明 相关功能请参见输入滤波器。	RW

19.6.5 数据寄存器 (I2C_DATA)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								DAT							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	DAT	数据寄存器 - 在发送模式下，写入待发送的数据 - 在接收模式下，读出接收到的数据	RW

19.6.6 状态寄存器 (I2C_STAT)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000000F8																															
Name	Reserved																								STAT							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	STAT	I2C 状态寄存器 状态码的具体定义请参见 状态码表述	RO

19.6.7 从机地址 0 寄存器 (I2C_ADDR0)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								RW ADDR0				GC			
Access																									RW							

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:1	ADDR0	I2C 从机模式地址 0	RW
0	GC	广播地址应答使能 ● 0b0: 禁止 ● 0b1: 使能	RW

19.6.8 从机地址 1 寄存器 (I2C_ADDR1)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								ADDR1						Reserved	
Access																									RW							

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:1	ADDR1	I2C 从机模式地址 1	RW
0	Reserved	-	-

19.6.9 从机地址 2 寄存器 (I2C_ADDR2)

Offset	Bit Position																															
0x24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								ADDR2						Reserved	
Access																									RW							

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:1	ADDR2	I2C 从机模式地址 2	RW
0	Reserved	-	-

19.6.10 从机地址匹配寄存器 (I2C_MATCH)

Offset	Bit Position																																					
0x28	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																																					
Access																																	RO	AD2F	RO	AD1F	RO	AD0F

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2	AD2F	I2C 从机模式地址 2 匹配标志位 ● 0b0: 从总线收到的设备地址与 ADDR2 不相同 ● 0b1: 从总线收到的设备地址与 ADDR2 相同	RO
1	AD1F	I2C 从机模式地址 1 匹配标志位 ● 0b0: 从总线收到的设备地址与 ADDR1 不相同 ● 0b1: 从总线收到的设备地址与 ADDR1 相同	RO
0	AD0F	I2C 从机模式地址 0 匹配标志位 ● 0b0: 从总线收到的设备地址与 ADDR0 不相同 ● 0b1: 从总线收到的设备地址与 ADDR0 相同	RO

说明

地址匹配标志位在以下三种情况下会清零：

- 模块复位时
- START/STOP 发送时
- 从机内部 STOP 时

20 高速集成电路总线 (HSI2C)

20.1 简介

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。

本 I2C 模块符合 SMBus 总线规范 (Version 3)。SMBus 是一种单端简易双线总线，通常用于低带宽通信。

20.2 主要特性

HSI2C 支持：

- 支持标准、快速、快速增强、高速模式。
- 从机模式下的高速模式 (Hs-mode)。
- 支持多主机，包括同步和仲裁。多主机意味着可以存在任意数量的主机。此外，主机模式和从机模式可以在传输间隙切换（在发送 STOP 后）。
- 支持 7 位和 10 位寻址。
- 支持软件复位、START 字节和设备 ID（也需要软件支持）。
- 支持 DMA 发送、接收。

HSI2C 主机支持：

- 一个字的命令/发送缓冲区（8 位发送数据+3 位命令），一个字的接收缓冲区（8 位接收数据）。
- 命令缓冲区将在启动传输之前等待空闲的 I2C 总线。
- 命令缓冲区可以初始化（重复）START 和 STOP 条件以及一个或多个主机接收传输。
- STOP 条件可以从命令缓冲区生成，也可以在发送缓冲区为空时自动生成。
- 灵活的接收数据匹配功能，可以在数据匹配时产生中断或丢弃不需要的数据。
- 标志和可选中断，用于表示重复 START 条件、STOP 条件、仲裁失败、NACK 和命令字错误。
- 支持可配置的总线空闲超时和引脚拉低超时。

HSI2C 从机支持：

- 独立的 I2C 从机寄存器，以最大限度地减少主/从机之间切换带来的软件额外开销。
- 支持 SMBus 报警地址和广播地址。
- 支持时钟延展。
- 支持发送数据寄存器中断，支持接收数据寄存器中断。
- 软件可控的 ACK 或 NACK，可配置的 ACK/NACK 位时钟延展。以避免发送缓冲区欠载和接收缓冲区溢出错误。
- 支持数据包末尾、STOP 条件或误码检测的标志和中断。
- 支持超低功耗唤醒模式。

20.3 功能说明

20.3.1 系统框图

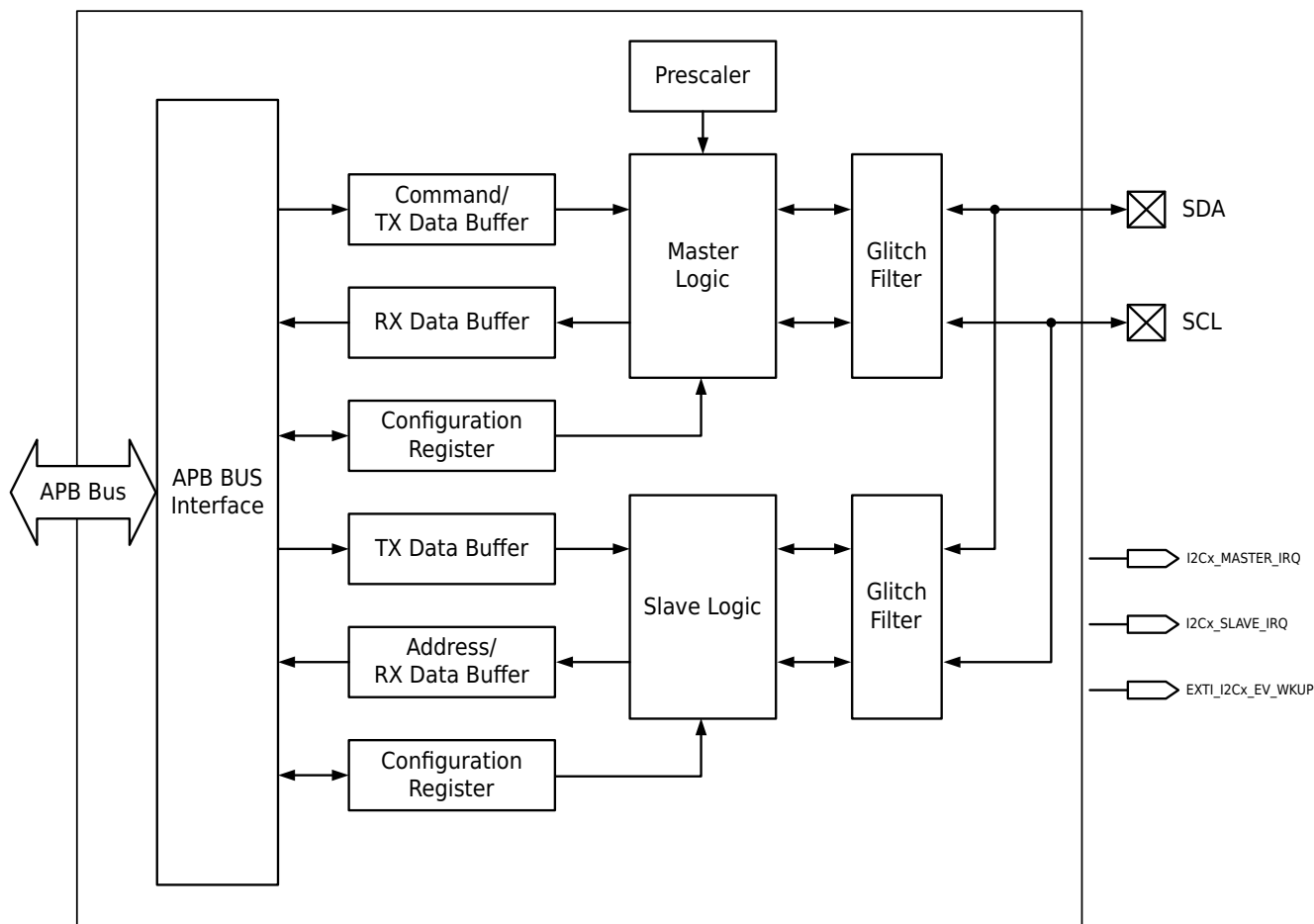


图 20-1 I2C 系统框图

20.3.2 协议描述

I2C 总线由一根时钟线（SCL），一根数据线（SDA）构成。所有的连接器件必须是漏极开路输出。SCL 和 SDA 线外接上拉电阻，电阻阻值取决于系统应用。

通常情况下，一个完整的通信过程包括下列 4 部分：

1. START 条件
2. 地址传输
3. 数据传输
4. STOP 条件

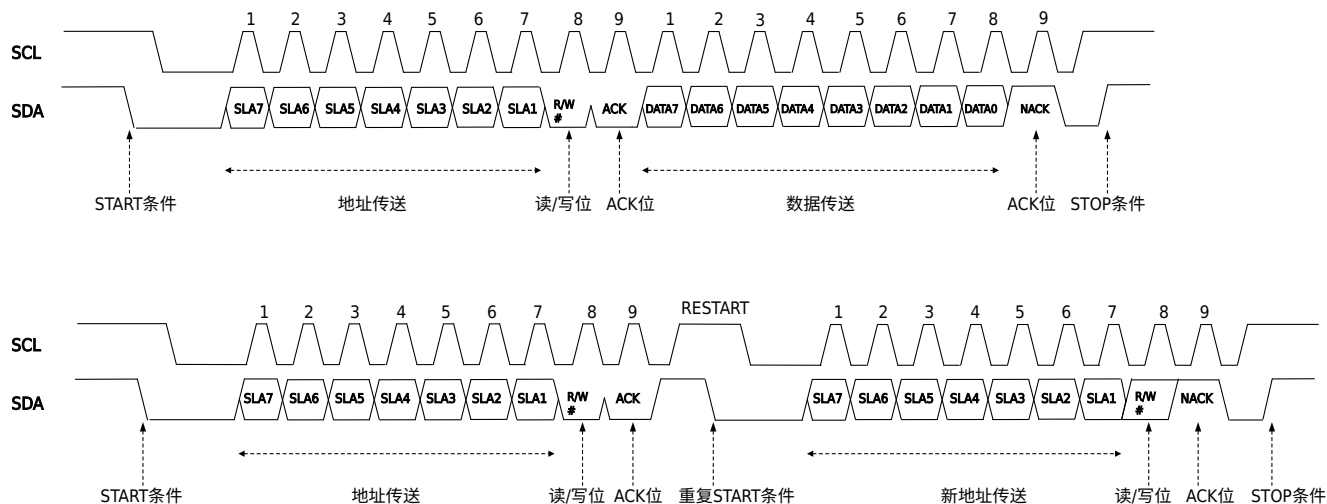


图 20-2 I2C 总线的时序图

20.3.3 主机模式

20.3.3.1 发送和命令缓冲区命令

发送缓存存储命令数据以启动各种 I2C 操作。只要启用 I2C（设置 MCR.MEN 位），就可以通过发送缓存区中的命令启动以下操作：

- 地址字节和期望 ACK 或 NACK 的 START 或 STOP 条件。
- 发送数据（这是对发送缓冲区进行零扩展字节写入的默认行为）。
- 接收一个字节的的数据（也可以配置为丢弃接收数据，不存储在接收缓冲区中）。
- STOP 条件（也可以配置为在发送缓冲区为空时发送 STOP 条件）。

在启动条件和停止条件之间可以插入多个发送和接收命令；发送和接收命令不得交叉。接收数据命令、接收数据和丢弃命令可以交叉，以确保只有所需的接收数据存储在接收缓冲器中。

I2C 主机在接收数据命令的最后一个字节上自动发送 NACK，除非缓冲区中的下一个命令也是接收数据命令。如果在接收数据命令完成时发送缓冲区为空，也会自动发送 NACK。

所有发送完成后，应关闭 I2C 主机使能（MCR.MEN）。

20.3.3.2 主机操作

无论何时启动 I2C，它都会监视 I2C 总线，以检测 I2C 总线何时空闲（MSR.BBF）。如果 SCL 或 SDA 为低电平，则 I2C 总线不再被视为空闲；如果检测到 STOP 条件或总线空闲超时（MCFGR2.BUSIDLE 配置），I2C 总线将变为空闲。在 I2C 总线空闲后，发送缓冲区不为空，然后 I2C 主机在 I2C 总线上发起传输。参考以下步骤：

- 总线空闲等待时间等于（MCCR0.CLKLO+1）乘以预分频器（MCFGR1.PRESCALE）。
- 使用主机时钟配置 0（MCCR0）中的时序配置发送 START 条件和地址字节；如果配置为高速模式，则使用主机时钟配置 1（MCCR1）的时序配置。

根据发送缓冲区的配置，执行主机发送传输或主机接收传输。

- 在主机接收传输的最后一个字节上发送 NACK，除非发送缓冲区中的下一个命令也是接收数据命令，并且发送缓冲区不为空。
- 重复 START 或 STOP 条件的发送由发送缓冲器和/或 MCFGR1.AUTOSTOP 进行配置。重复 START 条件可以更改所使用的时序配置寄存器。

当 I2C 主机被禁用时（MCR.MEN 被清除或模式输入），I2C 会持续清空发送缓冲区，直到 STOP 条件被发送。然而，I2C 将不再暂停 I2C 总线等待发送或接收缓冲区，并且在发送缓冲区为空后，I2C 会自动生成 STOP 条件。

I2C 主机可以在下面的特定条件下停止 I2C 总线；这导致 SCL 在字节的第一位持续拉低，直到条件解除：

- I2C 主设备已启用且正忙，发送缓冲区为空，MCFGR1.AUTOSTOP 被清除。
- I2C 主设备已启用并正在接收数据，接收数据未被丢弃（由于命令或接收数据匹配），接收缓冲区已满。

20.3.3.3 接收缓冲区和数据匹配

接收缓冲器用于在主机接收传输期间存储接收数据。接收数据还可以被配置为丢弃接收数据，而不是存储在接收缓冲区中；这是由发送缓冲器中的命令字配置的。

接收数据支持接收数据匹配功能，该功能可以将接收到的数据与两个字节之一或掩码数据字节进行匹配。数据匹配功能还可以被配置为仅比较上次（重复）START 条件以来的前一个或两个接收到的数据字。由于命令字而丢弃的接收数据不会造成数据匹配，并将第一个接收到的数据字上的匹配延迟到接收到丢弃的数据之后。接收器匹配功能还可以配置为丢弃所有接收数据，直到检测到数据匹配为止，使用 MCFGR1.RDMO 控制位。在数据匹配后清除 MCFGR1.RDMO 控制位，在清除 MSR.DMF 之前清除 MCFGR1.RDMO，以允许接收所有后续数据。

20.3.3.4 时序参数

I2C 主机可以配置以下时序参数。高速模式（主机时钟配置 1（MCCR1））和其他模式（主机时钟配置 0（MCCR0））的参数可以分别配置。这允许使用常规时序参数发送高速模式主机代码，然后切换到高速模式时序（在重复 START 后），直到下一个 STOP 条件产生。

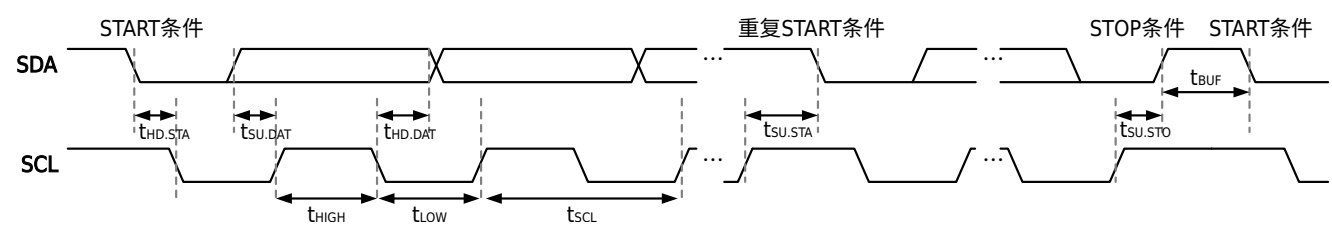


图 20-3 I2C 总线时序参数图

I2C 主机（Master）的时序参数通常与 I2C 功能时钟周期相关，参数配置需要符合对应模式要求的规格。

表 20-1 主机时序参数

参数	符号	时序参数（I2C 功能时钟周期）
SCL 时钟周期	t_{SCL}	$(CLKHI+CLKLO+2+SCL_LATENCY) \times 2^{PRESCALE}$
SCL 时钟低时间	t_{LOW}	$(CLKLO+1) \times 2^{PRESCALE}$
SCL 时钟高时间	t_{HIGH}	$(CLKHI+1+SCL_LATENCY) \times 2^{PRESCALE}$
（重复）START 条件保持时间	$t_{HD:STA}$	$(SETHOLD+1) \times 2^{PRESCALE}$
数据建立时间	$t_{SU:DAT}$	$(SDA_LATENCY+1) \times 2^{PRESCALE}$
数据保持时间	$t_{HD:DAT}$	$(DATAVD+1) \times 2^{PRESCALE}$
重复 START 条件建立时间	$t_{SU:STA}$	$(SETHOLD+1+SCL_LATENCY) \times 2^{PRESCALE}$
STOP 条件建立时间	$t_{SU:STO}$	$(SETHOLD+1+SCL_LATENCY) \times 2^{PRESCALE}$
总线空闲（STOP 条件至 START 条件）	t_{BUF}	$(CLKLO+1+SDA_LATENCY) \times 2^{PRESCALE}$
数据有效时间	$t_{VD:DAT}$	$(DATAVD+1) \times 2^{PRESCALE}$
数据应答有效时间	$t_{VD:ACK}$	$(DATAVD+1) \times 2^{PRESCALE}$

下表定义了延迟参数，这些参数假设上升时间小于一个 I2C 功能时钟周期。上升时间取决于多个因素，包括 I/O 传播延迟、I2C 总线负载以及外部上拉电阻的阻值。较大的上升时间会增加信号通过同步器（和毛刺滤波器）所需的时钟周期数，从而增加延迟。

表 20-2 同步延迟

参数	定义
SCL_LATENCY	$\text{ROUNDDOWN}((2 + \text{FILTSCl} + \text{SCL_RISETIME}) / 2^{\text{PRESCALE}})$
SDA_LATENCY	$\text{ROUNDDOWN}((2 + \text{FILTSdA} + \text{SDA_RISETIME}) / 2^{\text{PRESCALE}})$

I2C 配置需要遵循以下时序限制，从而避免 I2C 总线上出现意外 START 或 STOP 条件，同时也可避免 I2C 主机检测到意外的 START 或 STOP 条件。当 SCL 在发送（重复）START 或 STOP 条件之外为高电平时，SDA 不能改变。

表 20-3 I2C 时序参数限制

参数	最小值	最大值	说明
CLKLO	0x03	-	同时满足 $\text{CLKLO} > \text{SCL_LATENCY}$
CLKHI	0x01	-	配置 CLKHI 以满足 I2C 规范中的占空比要求
SETHOLD	0x02	-	同时满足 $\text{SETHOLD} > \text{SDA_LATENCY}$
DATAVD	0x01	$\text{CLKLO} - \text{SDA_LATENCY} - 1$	配置 DATAVD 以满足 I2C 规范中的数据保持要求
FILTSCl	0x00	$[\text{CLKLO} \times 2^{\text{PRESCALE}}] - 3$	FILTCSL 和 FILTSdA 是唯一不乘以 2^{PRESCALE} 的参数
FILTSdA	FILTSCl	$[\text{CLKLO} \times 2^{\text{PRESCALE}}] - 3$	配置大于 FILTSL 的 FILTSdA 可以延迟 SDA 输入，以补偿板级偏移
BUSIDLE	$(\text{CLKLO} + \text{SETHOLD} + 2) \times 2$	-	同时满足 $\text{BUSIDLE} > \text{CLKHI} + 1$

时序参数必须配置为满足 I2C 规范的要求；这取决于所支持的模式和 I2C 功能时钟频率。当使用不同的时钟配置寄存器在两种模式之间切换时（例如，快速和高速模式），PRESCALE 系数必须在模式之间保持恒定。下面提供了一些示例时序配置。

表 20-4 I2C 示例时序配置

I2C 模式	时钟频率	波特率	PRESCALE	FILTSCl / FILTSdA	SETHOLD	CLKO	CLKHI	DATAVD
Standard	8MHz	100kbps	0x0	0xD/0xF	0x37	0x29	0x16	0x0D
Fast	8MHz	400kbps	0x0	0x0/0x0	0x04	0x0B	0x05	0x02
Fast+	8MHz	1Mbps	0x0	0x0/0x0	0x02	0x03	0x01	0x01
Standard	48MHz	100kbps	0x2	0x2/0xE	0x3A	0x3B	0x39	0x0C
Fast	48MHz	400kbps	0x0	0x1/0x1	0x1D	0x3E	0x35	0x0F
Fast	48MHz	400kbps	0x2	0x1/0x1	0x07	0x11	0x0B	0x03

I2C 模式	时钟频率	波特率	PRESCAL E	FILTSL/ FILTSDA	SETHOLD	CLKO	CLKHI	DATAVD
Fast+	48MHz	1Mbps	0x2	0x1/0x1	0x03	0x06	0x04	0x04
HS 模式	48MHz	3.2Mbps	0x0	0x0/0x0	0x07	0x08	0x03	0x01

20.3.3.5 错误条件说明

I2C 主机在活动时监控错误，以下情况会生成错误标志并阻止发送新的 START 条件，直到该标志被软件清除：

- 检测到 START 或 STOP 条件，且该条件不是由 I2C 主机生成的（MSR.ALF 被置位）。
软件必须响应 MSR.ALF 标志，用于终止现有命令（通过设置 MCR.RST）。
- 在 SDA 上发送数据，但接收到不同的数据（MSR.ALF 被置位）。
- 在发送数据时检测到 NACK，且 MCFGR1.IGNACK 为清除状态（MSR.NDF 被置位）。
- 检测到 NACK，但地址字节的期望响应为 ACK，且 MCFGR1.IGNACK 为清除状态（MSR.NDF 被置位）。
- 检测到 ACK，但地址字节的期望响应为 NACK，且 MCFGR1.IGNACK 为清除状态（MSR.NDF 被置位）。
- 在没有 START 条件的情况下，发送缓冲区正在请求发送或接收数据（MSR.FEF 被置位）。
- SCL（或 SDA，如果设置了 MCFGR1.TIMECFG）在（MCFGR3.PINLOW*256）预分频器周期内为低电平且没有发生引脚状态变化（MSR.PLTF 被置位）。

软件必须响应 MSR.PLTF 标志，用于终止现有命令（通过设置 MCR.RST）。

当 SCL 和 SDA 在（MCFGR2.BUSIDLE+1）预分频器周期内保持高电平时，MCFGR2.BUSIDLE 字段可用于强制将 I2C 总线视为空闲。通常，当 I2C 主设备首次启用时，I2C 总线被认为是空闲的。但是，当 MCFGR2.BUSIDLE 配置为大于零时，SCL 和/或 SDA 必须保持高电平持续（MCFGR2.BUSIDLE+1）个预分频周期后，I2C 总线才会首次被视为空闲。

20.3.3.6 中断和 DMA 请求

下表列出了可以产生 I2C 主机中断和主机发送/接收 DMA 请求的主机模式标志。

标志	说明	中断	唤醒	DMA 请求
TDF	数据可写入发送缓冲区	✓	-	✓
RDF	可以从接收缓冲区读取数据	✓	-	✓
EPF	主机已发送重复 START 或 STOP 条件	✓	-	-
SDF	主机已发送 STOP 条件	✓	-	-
NDF	● 在地址字节期间，主机预期收到 ACK，但检测到 NACK ● 在地址字节期间，主机预期收到 NACK，但检测到 ACK ● 在主机发送数据字节期间，主机检测到 NACK	✓	-	-
ALF	● 在错误的时间检测到 START/STOP 条件，主机失去了仲裁 ● 主机正在发送数据，但收到的数据与发送的数据不同	✓	-	-
FEF	主机期望命令缓冲区中出现 START 条件，但命令缓冲区的下一个条目不是 START 条件	✓	-	-

标志	说明	中断	唤醒	DMA 请求
PLTF	引脚低超时已启用，SCL（或者已配置 SDA）保持低电平时间长于配置的超时时间	✓	-	-
DMF	接收到的数据与配置的数据匹配，由 MTDR.CMD 配置的接收和丢弃的数据不会置位 DMF	✓	-	-
MBF	I2C 主机正忙于发送/接收数据	-	-	-
BBF	I2C 主机已启用，在 I2C 总线上检测到活动，但未检测到 STOP 条件，也未发生总线空闲超时（如果已启用）	-	-	-

20.3.4 从机模式

为了在 I2C 总线上执行所有从机模式传输，I2C 从机逻辑独立于 I2C 主机逻辑运行。

20.3.4.1 地址匹配

I2C 从机可以配置：

- 在 7 位或 10 位寻址模式下匹配两个地址之一。
- 在 7 位或 10 位寻址模式下匹配一个地址范围。
- 匹配广播地址，并生成相应的标志。
- 匹配 SMBus 报警地址，并生成相应的标志。
- 检测高速模式主机代码，禁用数字滤波器和输出有效延迟时间，直到检测到下一个 STOP 条件。

在匹配到有效地址后，I2C 从机会自动执行从机发送传输或从机接收传输，直到：

- 检测到 NACK（除非设置了 SCR1.IGNACK）
- 检测到比特错误（I2C 从机正在驱动 SDA，但采样到不同的值）
- 检测到（重复）START 或 STOP 条件

20.3.4.2 发送和接收数据

- 发送和接收数据寄存器是双缓冲的，仅在从机发送传输和从机接收传输期间分别更新。
- 接收到的从机地址可以配置为从机接收数据寄存器（SRDR）或从机地址状态寄存器（SASR）读取。
- 发送数据寄存器可以配置为仅在检测到从机发送传输后请求数据，或者在发送数据寄存器为空时请求新数据。
- 仅当发送数据标志（SSR.TDF）置高时，才能写从机发送数据寄存器（STDR）。
- 仅当接收数据标志（SSR.RDF）置高（或地址有效标志置高且 SCR1.RXCFG=1）时，才能读取从机接收数据寄存器（SRDR）。
- 仅当地址有效标志（SSR.AVF）置高时，才能读取从机地址状态寄存器（SASR）。

20.3.4.3 时钟延展

I2C 从机支持时钟延展。可以配置以下条件来执行时钟延展：

- 在地址字节的第 9 个时钟脉冲期间，地址有效标志（SSR.AVF）置高。
- 在从机发送传输的第 9 个时钟脉冲周期，发送数据标志（SSR.TDF）置高。
- 在从机接收传输的第 9 个时钟脉冲期间，接收数据标志（SSR.RDF）置高。
- 在地址字节或从机接收传输的第 8 个时钟脉冲期间，发送 ACK 标志（SSR.TAF）置高。在高速模式下，此配置被禁用。
- 时钟延展也可以扩展到 SCR2.CLKHOLD 周期，以允许额外的建立时间，从而支持外部对 SDA 引脚进行采样。在高速模式下，此配置被禁用。

除非通过 SCR2.CLKHOLD 配置进行扩展，否则在启用时钟延展时，SDA 更新后，时钟延展将延展一个外设总线时钟周期。

20.3.4.4 时序参数

I2C 从机可以配置以下时序参数。当 SCR2.FILTEN 清除时或当 I2C 从机检测到高速模式时，这些参数将被禁用。禁用时，I2C 从机直接从 I2C 总线计时，可能无法满足 I2C 规范的所有时序要求（例如标准/快速模式下的 SDA 最小保持时间）。

- 从 SCL 拉低到 SDA 更新的 SDA 数据有效时间
- 启用时钟延展时的 SCL 保持时间，以增加采样外部 SDA 时的建立时间
- SCL 毛刺过滤时间
- SDA 毛刺过滤时间

I2C 从机对时序参数有以下限制：

- SCR2.FILTSDA 必须配置为大于或等于 SCR2.FILTSCS（除非补偿 SDA 和 SCL 之间的板级偏移）。
- SCR2.DATAVD 的配置必须小于最小 SCL 低电平周期。

20.3.4.5 错误条件说明

I2C 从机可以检测以下错误情况：

- 当 I2C 从机驱动 SDA 时，采样值与预期值不同，位错误标志（SSR.BEF）会置高。
- 当发送数据欠载或接收数据溢出时，缓冲区错误标志（SSR.FEF）会置高。为避免发生欠载和溢出，可打开时钟延展功能。
- 当 SCR1.RXCFCG 置高时，如地址溢出，缓冲区错误标志（SSR.FEF）会置高，否则不会标记地址溢出。为避免发生溢出的情况，可打开时钟延展功能。

I2C 从机不会因 SCL 和/或 SDA 被拉低而实现超时检测。如果需要检测这种情况，则应使用 I2C 主机逻辑，以便在检测到此类条件时，软件可以复位 I2C 从机。

20.3.4.6 中断和 DMA 请求

下表列出了可以产生 I2C 从机中断和从机发送/接收 DMA 请求的从机模式标志。

标志	说明	中断	唤醒	DMA 请求
TDF	数据可写入从机发送数据寄存器（STDR）	✓	-	✓
RDF	可以从从机接收数据寄存器（SRDR）读取数据	✓	-	✓
AVF	可以从从机地址状态寄存器（SASR）读取地址	✓	✓	-
TAF	ACK/NACK 可以写入从机发送 ACK 寄存器（STAR）	✓	-	-
RSF	从机检测到地址匹配，随后出现重复 START 条件	✓	-	-
SDF	从机检测到地址匹配，随后出现 STOP 条件	✓	-	-
BEF	从机正在发送数据，但收到的数据与发送的数据不同	✓	-	-
FEF	● 发送数据欠载 ● 接收数据溢出 ● 地址状态溢出（当接收数据配置 SCR1.RXCFCG=1 时） FEF 标志只能在禁用时钟延展时被置位。	✓	-	-
AM0F	从机检测到与 SADDR.ADDR0 的地址匹配	✓	-	-
AM1F	从机检测到与 SADDR.ADDR1 的地址匹配或使用地址范围	✓	-	-

标志	说明	中断	唤醒	DMA 请求
GCF	从机检测到与广播地址匹配的地址	✓	-	-
SARF	从机检测到与 SMBus 报警地址匹配的地址	✓	-	-
SBF	I2C 从机正忙于接收地址字节或正在发送/接收数据	-	-	-
BBF	I2C 从机已启用，在 I2C 总线上检测到 START 条件，但未检测到 STOP 条件	-	-	-

20.3.5 通信电平选择功能

HSI2C 接口支持不同电压下电平信号通信。功能详情参见《端口控制器（GPIO）》章节“[HSI2C 通信电平选择功能](#)”相关描述。

20.4 操作示例

20.4.1 主机发送模式

主机发送模式下的操作流程如下：

- 步骤 1. 使能 I2C 模块时钟，复位 I2C 模块。
- 步骤 2. 配置 I2C 模块 SCL、SDA 管脚。
- 步骤 3. 根据传输波特率配置 I2C 寄存器，使发送的 SCL 的频率和传输波特率一致：
 - MCFGR1.PRESCALE
 - MCCR0.CLKLO
 - MCCR0.CLKHI
 - MCCR0.DATAVD
 - MCCR0.ETHOLD
- 步骤 4. 设置 MCR.MEN 为 1，使能 I2C 模块。
- 步骤 5. 查询 MSR.BBF 是否为 0，为 0 则继续进行下一步操作。
- 步骤 6. 向 MTDR 寄存器写入 START 指令和地址+写操作，例如 0x4AA。
- 步骤 7. 等待 MSR.TDF 为 1 后将待发送数据写入 MTDR，继续查询 MSR.TDF，等待 MSR.TDF 为 1 后将下一个待发送数据写入 MTDR。循环本步骤操作直到把所有待发送数据写入 MTDR。
- 步骤 8. 继续查询 MSR.TDF，等待 MSR.TDF 为 1 后向 MTDR.CMD 写入停止（STOP）命令，例如 0x2。查询 I2C 主机 MSR.SDF 是否为 1，为 1 表示 STOP 条件已发出，发送操作结束。

20.4.2 主机接收模式

主机接收模式下的操作流程如下（此处以 7-bit 地址匹配为例，其他方式参考协议内容）：

- 步骤 1. 使能 I2C 模块时钟，复位 I2C 模块。
- 步骤 2. 配置 I2C 模块 SCL、SDA 管脚。
- 步骤 3. 根据传输波特率配置 I2C 寄存器，使发送的 SCL 的频率和传输波特率一致：
 - MCFGR1.PRESCALE
 - MCCR0.CLKLO
 - MCCR0.CLKHI
 - MCCR0.DATAVD
 - MCCR0.ETHOLD
- 步骤 4. 设置 MCR.MEN 为 1，使能 I2C 模块。

- 步骤 5.** 查询 MSR.BBF 是否为 0，为 0 则继续进行下一步操作。
- 步骤 6.** 向 MTDR 寄存器写入 START 指令和地址+读操作，例如 0x4AB。
- 步骤 7.** 等待 MSR.TDF 为 1 后将接收数据指令+数据数量（数据数量为实际接收数据数量减 1）写入 MTDR，例如 0x103，表示接收 4 个数据。
- 步骤 8.** 等待 MSR.RDF 为 1 后从 MRDR 读取接收到的数据，继续查询 MSR.RDF，等待 MSR.RDF 为 1 后从 MRDR 读取下一个接收到的数据。循环本步骤操作直到把所有接收到的数据从 MRDR 读出。
- 步骤 9.** 继续查询 MSR.TDF，等待 MSR.TDF 为 1 后向 MTDR.CMD 写入停止 (STOP) 命令，例如 0x2。查询 I2C 主机 MSR.SDF 是否为 1，为 1 表示 STOP 条件已发出，接收操作结束。

20.4.3 从机发送模式

从机发送模式下的操作流程如下：

- 步骤 1.** 使能 I2C 模块时钟，复位 I2C 模块。
- 步骤 2.** 配置 I2C 模块 SCL、SDA 管脚。
- 步骤 3.** 配置 I2C 从机控制 1 寄存器和从机地址寄存器：
- SCR1.ADDRMODE
 - SCR1.TXCFG
 - SADDR.ADDR0
- 步骤 4.** 设置 SCR0.SEN 为 1，使能 I2C 模块。
- 步骤 5.** 查询 SSR.AVF 是否为 1，为 1 则继续进行下一步操作。
- 步骤 6.** 读取 SASR.RADDR[0]：
- 为 0 表示主机写操作，从机处于接收模式；
 - 为 1 表示主机读操作，从机处于发送模式。
- 步骤 7.** 等待 SSR.TDF 为 1 后把待发送数据写入 STDR，继续查询 SSR.TDF，等待 SSR.TDF 为 1 后将下一个待发送数据写入 STDR。循环本步骤操作直到把所有待发送数据写入 STDR。
- 步骤 8.** 查询 SSR.AVF 是否为 0，为 0 表示已读取 I2C 地址状态寄存器 SASR。查询 I2C 从机 SSR.SDF 是否为 1，为 1 表示已检测到 STOP 条件，发送操作结束。

20.4.4 从机接收模式

从机接收模式下的操作流程如下：

- 步骤 1.** 使能 I2C 模块时钟，复位 I2C 模块。
- 步骤 2.** 配置 I2C 模块 SCL、SDA 管脚。
- 步骤 3.** 配置 I2C 从机控制 1 寄存器和从机地址寄存器：
- SCR1.ADDRMODE
 - SADDR.ADDR0
- 步骤 4.** 设置 SCR0.SEN 为 1，使能 I2C 模块。
- 步骤 5.** 查询 SSR.AVF 是否为 1，为 1 则继续进行下一步操作。
- 步骤 6.** 读取 SASR.RADDR[0]：
- 为 0 表示主机写操作，从机处于接收模式；
 - 为 1 表示主机读操作，从机处于发送模式。
- 步骤 7.** 等待 SSR.RDF 为 1 后从 SRDR 读取接收到的数据，继续查询 SSR.RDF，等待 SSR.RDF 为 1 后从 SRDR 读取下一个接收到的数据。循环本步骤操作直到把所有接收到的数据从 SRDR 读出。
- 步骤 8.** 查询 SSR.AVF 是否为 0，为 0 表示已读取 I2C 地址状态寄存器 SASR，查询 I2C 从机 SSR.SDF 是否为 1，为 1 表示已检测到 STOP 条件，接收操作结束。

20.4.5 主机高速模式切换

主机在复位后工作在快速模式，如果主机需要切换到高速模式，必须完成以下操作流程：

步骤 1. 使能 I2C 模块时钟，复位 I2C 模块。

步骤 2. 配置 I2C 模块 SCL、SDA 管脚。

步骤 3. 根据传输波特率配置 I2C 寄存器，使发送的 SCL 的频率和传输波特率一致：

- MCFGR1.PRESCALE
- MCCR0.CLKLO
- MCCR0.CLKHI
- MCCR0.DATAVD
- MCCR0.ETHOLD

步骤 4. 根据高速模式的波特率配置 I2C 寄存器，使切换后发送的 SCL 的频率和高速模式波特率一致：

- MCCR1.CLKLO
- MCCR1.CLKHI
- MCCR1.DATAVD
- MCCR1.ETHOLD

步骤 5. 设置 MCR.MEN 为 1，使能 I2C 模块。

步骤 6. 查询 MSR.BBF 是否为 0，为 0 则继续进行下一步操作。

步骤 7. 向 MTDR 寄存器写入 START 指令和控制码 (0x00001xxx)，例如 0x509。主机期望收到 NACK 应答后切换到高速模式。

步骤 8. 等待 MSR.TDF 为 1 后向 MTDR 寄存器写入 START 指令和地址+读/写操作，例如 0x6AB/0x6AA。

步骤 9. 后续的操作：

- 读操作请参考[主机接收模式](#)的步骤 7~9。
- 写操作请参考[主机发送模式](#)的步骤 7~8。

20.5 寄存器

20.5.1 寄存器总表

HSI2C 基地址: 0x4000BC00

表 20-7 高速集成电路总线 (HSI2C) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x010	I2C_MCR	主机控制寄存器	32
0x014	I2C_MSR	主机状态寄存器	32
0x018	I2C_MIER	主机中断使能寄存器	32
0x01C	I2C_MSCR	主机状态清除寄存器	32
0x024	I2C_MCFGR1	主机配置 1 寄存器	32
0x028	I2C_MCFGR2	主机配置 2 寄存器	32
0x02C	I2C_MCFGR3	主机配置 3 寄存器	32
0x030	I2C_MCCR0	主机时钟配置 0 寄存器	32
0x034	I2C_MCCR1	主机时钟配置 1 寄存器	32
0x038	I2C_MTDR	主机发送数据寄存器	32
0x03C	I2C_MRDR	主机接收数据寄存器	32
0x040	I2C_MDMR	主机数据配置寄存器	32
0x04C	I2C_MDER	主机 DMA 使能寄存器	32
0x110	I2C_SCR0	从机控制 0 寄存器	32
0x114	I2C_SSR	从机状态寄存器	32
0x118	I2C_SIER	从机中断使能寄存器	32
0x11C	I2C_SSCR	从机状态清除寄存器	32
0x120	I2C_SCR1	从机控制 1 寄存器	32
0x124	I2C_SCR2	从机控制 2 寄存器	32
0x128	I2C_STAR	从机发送 ACK 寄存器	32
0x12C	I2C_STDR	从机发送数据寄存器	32
0x130	I2C_SRDR	从机接收数据寄存器	32
0x13C	I2C_SDER	从机 DMA 使能寄存器	32
0x140	I2C_SADDR	从机地址寄存器	32
0x150	I2C_SASR	从机地址状态寄存器	32

20.5.2 主机控制寄存器 (I2C MCR)

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																											DBGEN	Reserved			
Access																											RW		RW	RST	RW	MEN

位/位域	标记	功能描述	读写
31:4	<i>Reserved</i>	-	-
3	DBGEN	调试启用 在调试模式下启用主机。 ● 0b0：在调试模式中禁用主机 ● 0b1：在调试模式中启用主机	RW
2	<i>Reserved</i>	-	-
1	RST	软件重置 ● 0b0：主机逻辑不重置 ● 0b1：主机逻辑重置 重置所有内部主机逻辑和寄存器，主控制寄存器（MCR）除外。RST 保持置位，直到被软件清除。重置立即生效，并保持置位，直到被软件清除。清除软件重置之前不需要最小延迟。	RW
0	MEN	主机启用 ● 0b0：禁用主机模式 ● 0b1：启用主机模式	RW

20.5.3 主机状态寄存器 (I2C_MSR)

Offset	Bit Position																																							
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000001																																							
Name	Reserved								Reserved							DMF	PLTF	FEF	ALF	NDF	SDF	EPF		Reserved							RDF	TDF								
Access							RO	BBF	RO	MBF							RO	DMF	RO	PLTF	RO	FEF	RO	ALF	RO	NDF	RO	SDF	RO	EPF							RO	RDF	RO	TDF

位/位域	标记	功能描述	读写
31:26	Reserved	-	-
25	BBF	总线忙标志 0b0: I2C 总线空闲 0b1: I2C 总线忙	RO
24	MBF	主机忙标志 0b0: I2C 主机空闲 0b1: I2C 主机忙	RO
23:15	Reserved	-	-
14	DMF	数据匹配标志 0b0: 尚未收到匹配数据 0b1: 已收到匹配数据 表示接收的数据已与 MDMR.MATCH0 和/或 MDMR.MATCH1 字段匹配（由 MCFGR1.MATCFG 配置）。由于 MTDR.CMD 字段而丢弃的接收数据不会导致 DMF 设置。	RO
13	PLTF	引脚低超时标志 0b0: 引脚低电平超时未发生或已禁用 0b1: 引脚低电平超时已发生 在 SCL 和/或 SDA 输入低超过引脚低周期（引脚低超时，MCFGR3.PINLOW）时设置，即使在 I2C 主机空闲时。 1. 软件负责解决引脚低条件 2. 只要引脚低超时持续，就不能清除 PLTF 3. 在 I2C 可以开始 START 条件之前，必须清除 PLTF。	RO
12	FEF	缓冲区错误标志 0b0: 无错误 0b1: 在没有 START 条件的情况下发送或接收数据的主机 检测在未首先生成（重复）START 条件的情况下发送或接收数据的尝试。当 MCFGR1.AUTOSTOP=1 时，如果发送缓冲器下溢，则可能发生这种情况。当 FEF 置位时，I2C 主机发送 STOP 条件（如果忙碌），并且在 FEF 被清除之前不会启动新的 START 条件。	RO

位/位域	标记	功能描述	读写
11	ALF	仲裁丢失标志 ● 0b0: 主机未丢失仲裁 ● 0b1: 主机丢失仲裁 如果存在以下任一条件会置位: 1. I2C 主机发送逻辑 1, 并在 I2C 总线上检测逻辑 0 2. I2C 主机在发送数据时检测到 START 或 STOP 条件 当 ALF 置位时, I2C 主机释放 I2C 总线 (变为空闲), 并且在 ALF 被清除之前, I2C 主机不会启动新的 START 条件。	RO
10	NDF	NACK 检测标志 ● 0b0: 未检测到不期望 NACK ● 0b1: 检测到不期望 NACK 如果 I2C 主机在发送地址或数据时检测到它不期望的 NACK 会置位。置位后, 在 NDF 被清除之前, 主机不会开始新的 START 条件。如果给定地址需要 NACK (由命令字配置), 如果未生成 NACK, 则 NDF 置位。NDF 置位时, 如果 MCFGR1.AUTOSTOP=1, 或发送缓冲区不为空, I2C 主机自动发送 STOP 条件。	RO
9	SDF	停止检测标志 ● 0b0: 主机未生成 STOP 情况 ● 0b1: 主机已生成 STOP 情况 当 I2C 主机生成 STOP 条件时置位。	RO
8	EPF	结束数据包标志 ● 0b0: 主机未生成 STOP 或重复 START 条件 ● 0b1: 主机已生成 STOP 和重复 START 条件 当 I2C 主机生成重复的 START 条件或 STOP 条件时置位。主机首次生成 START 条件时不会置位。	RO
7:2	Reserved	-	-
1	RDF	接收数据标志 ● 0b0: 接收数据未就绪 ● 0b1: 接收数据就绪 每当接收缓冲区已满时, 都会置位接收数据标志 RDF。	RO
0	TDF	发送数据标志 ● 0b0: 发送数据请求未产生 ● 0b1: 发送数据请求已产生 当发送缓冲区为空时 TDF 置位。	RO

20.5.4 主机中断使能寄存器 (I2C_MIER)

Offset	Bit Position																																												
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
Reset	0x00000000																																												
Name	Reserved																			DMIE		PLTIE		FEIE		ALIE		NDIE		SDIE		EPIE	Reserved								RDIE		TDIE		
Access																				RW		RW		RW		RW		RW		RW		RW		RW									RW		RW

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14	DMIE	数据匹配中断使能 ● 0b0：禁用 ● 0b1：使能	RW
13	PLTIE	引脚低超时中断使能 ● 0b0：禁用 ● 0b1：使能	RW
12	FEIE	缓冲区错误中断使能 ● 0b0：禁用 ● 0b1：使能	RW
11	ALIE	仲裁丢失中断使能 ● 0b0：禁用 ● 0b1：使能	RW
10	NDIE	NACK 检测中断使能 ● 0b0：禁用 ● 0b1：使能	RW
9	SDIE	STOP 检测中断使能 ● 0b0：禁用 ● 0b1：使能	RW
8	EPIE	结束数据包中断使能 ● 0b0：禁用 ● 0b1：使能	RW
7:2	Reserved	-	-
1	RDIE	接收数据中断使能 ● 0b0：禁用 ● 0b1：使能	RW
0	TDIE	发送数据中断使能 ● 0b0：禁用 ● 0b1：使能	RW

20.5.5 主机状态清除寄存器 (I2C_MSCR)

Offset	Bit Position																																							
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved																										Reserved													
Access																			WO	DMC	WO	PLTC	WO	FEC	WO	ALC	WO	NDC	WO	SDC	WO	EPC								

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14	DMC	数据匹配标志清除 写 1 清除标志位，写 0 无效	WO
13	PLTC	引脚低超时标志清除 写 1 清除标志位，写 0 无效	WO
12	FEC	缓冲区错误标志清除 写 1 清除标志位，写 0 无效	WO
11	ALC	仲裁丢失标志清除 写 1 清除标志位，写 0 无效	WO
10	NDC	NACK 检测标志清除 写 1 清除标志位，写 0 无效	WO
9	SDC	STOP 检测标志清除 写 1 清除标志位，写 0 无效	WO
8	EPC	结束数据包标志清除 写 1 清除标志位，写 0 无效	WO
7:0	Reserved	-	-

20.5.6 主机配置 1 寄存器 (I2C_MCFGR1)



说明

仅当 I2C 主机被禁用时，才可以修改本寄存器。

Offset	Bit Position																																										
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Reset	0x00000000																																										
Name	Reserved																MATCFG		RDMO	Reserved	TIMECFG		IGNACK	AUTOSTOP		Reserved						PRESCALE											
Access																	RW		RW		RW		RW																			RW	

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:13	MATCFG	匹配配置 配置以下数值会导致 MSR.DMF 被置位。请参阅主机数据配置寄存器 (MDMR)。 0b000: 禁用匹配 0b001: 禁止设置 0b010: 启用匹配 (第一个数据字等于 MDMR.MATCH0 或 MDMR.MATCH1) 0b011: 启用匹配 (任何数据字等于 MDMR.MATCH0 或 MDMR.MATCH1) 0b100: 启用匹配 (第一个数据字等于 MDMR.MATCH0 并且第二个数据字等于 MDMR.MATCH1) 0b101: 启用匹配 (任何数据字等于 MDMR.MATCH0 并且下一个数据字等于 MDMR.MATCH1) 0b110: 启用匹配 (第一个数据字逻辑与 MDMR.MATCH1 等于 MDMR.MATCH0 逻辑与 MDMR.MATCH1) 0b111: 启用匹配 (任何数据字逻辑与 MDMR.MATCH1 等于 MDMR.MATCH0 逻辑与 MDMR.MATCH1)	RW
12	RDMO	仅接收数据匹配 0b0: 接收的数据存储在接收缓冲器中 0b1: 除非设置了数据匹配标志 (MSR.DMF)，否则丢弃接收的数据 启用时，将丢弃不会导致数据匹配标志 (MSR.DMF) 置位的所有接收数据。MSR.DMF 置位后，RDMO 配置被忽略。禁用 RDMO 时，请在清除 MSR.DMF 之前清除 RDMO，以确保没有接收数据丢失。	RW
11	Reserved	-	-
10	TIMECFG	超时配置 0b0: MSR.PLTF 置位如果 SCL 低电平持续时间超过配置的超时 0b1: MSR.PLTF 置位如果 SCL 或 SDA 低电平持续时间超过配置的超时 TIMECFG 配置引脚低电平超时标志字段 (MSR.PLTF) 的超时置位。	RW

位/位域	标记	功能描述	读写
9	IGNACK	IGNACK 配置 0b0: I2C 主机正常接收 ACK/NACK 0b1: I2C 主机将接收的 NACK 视为 ACK 设置时，接收的 NACK 字段被忽略，并假设为 ACK。IGNACK 位需要在超快速模式下设置。	RW
8	AUTOSTOP	自动 STOP 生成 0b0: 无效果 0b1: 每当发送缓冲器为空且 I2C 主机忙时，自动生成 STOP 状态启用时，每当 I2C 主机忙碌且发送缓冲器为空时，就会生成 STOP 条件；也可以使用发送缓冲器命令生成 STOP 条件。	RW
7:3	Reserved	-	-
2:0	PRESCALE	预分频器配置 配置用于所有 I2C 主机的时钟预分频器，数字毛刺滤波器除外。 0b000: PCLK 时钟 0b001: PCLK 时钟 2 分频 0b010: PCLK 时钟 4 分频 0b011: PCLK 时钟 8 分频 0b100: PCLK 时钟 16 分频 0b101: PCLK 时钟 32 分频 0b110: PCLK 时钟 64 分频 0b111: PCLK 时钟 128 分频	RW

20.5.7 主机配置 2 寄存器 (I2C_MCFGR2)



说明

仅当 I2C 主机被禁用时，才可以修改本寄存器。

Offset	Bit Position																																
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x80800000																																
Name	FILTBPSDA	Reserved			FILTSDA			FILTBPSCL	Reserved			FILTSCCL			Reserved			BUSIDLE															
Access	RW				RW			RW				RW						RW															

位/位域	标记	功能描述	读写
31	FILTBPSDA	模拟滤波器 SDA 旁路 0b0: SDA 通过模拟滤波器 0b1: SDA 旁路模拟滤波器	RW
30:28	Reserved	-	-
27:24	FILTSDA	毛刺滤波器 SDA 为 SDA 输入配置 I2C 主机数字毛刺滤波器。 1. 配置为 0 将禁用毛刺滤波器 2. 等于或小于 FILTSDA 周期长的毛刺被过滤掉并忽略 3. 通过毛刺滤波器的延迟等于 FILTSDA 周期，并且必须配置为小于最小 SCL 低电平或高电平周期 4. 毛刺滤波器周期计数不受 PRESCALE 配置的影响，并且在高速模式下自动绕过毛刺滤波器周期数	RW
23	FILTBPSCL	模拟滤波器 SCL 旁路 0b0: SCL 通过模拟滤波器 0b1: SCL 旁路模拟滤波器	RW
22:20	Reserved	-	-
19:16	FILTSCCL	毛刺滤波器 SCL 为 SCL 输入配置 I2C 主数字毛刺滤波器。 1. 配置为 0 会禁用毛刺滤波器 2. 等于或小于 FILTSCCL 周期的毛刺被过滤掉并忽略。FILTSCCL 周期基于功能时钟 3. 通过毛刺滤波器的延迟等于 FILTSCCL 周期，并且必须配置为小于最小 SCL 低周期或高周期 4. 毛刺滤波器周期计数不受 PRESCALE 配置的影响，并且在高速模式下自动绕过毛刺滤波器周期计数	RW
15:12	Reserved	-	-
11:0	BUSIDLE	总线空闲超时 以时钟周期为单位配置总线空闲超时周期。 1. 如果 SCL 和 SDA 的高值持续超过 BUSIDLE 周期，则假设 I2C 总线空闲，并且主机可以生成 START 条件 2. 当总线空闲超时设置为零时，总线空闲超时被禁用	RW

20.5.8 主机配置 3 寄存器（I2C_MCFGR3）

说明

仅当 I2C 主机被禁用时，才可以修改本寄存器。

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												PINLOW												Reserved							
Access													RW																			

位/位域	标记	功能描述	读写
31:20	Reserved	-	-
19:8	PINLOW	引脚低超时 以时钟周期为单位配置引脚低超时标志。 1. 如果 SCL 或 SCL/SDA 二者之一的低电平持续时间超过（PINLOW*256）个周期，则 MSR.PLTF 被置位。 2. 当引脚低超时设置为零时，引脚低超时功能被禁用。	RW
7:0	Reserved	-	-

20.5.9 主机时钟配置 0 寄存器 (I2C_MCCR0)



说明

当 I2C 主机启用时，无法更改本寄存器，并用于标准、快速、快速模式加和超快速传输。

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	DATAVD							Reserved	SETHOLD							Reserved	CLKHI							Reserved	CLKLO						
Access		RW								RW								RW								RW						

位/位域	标记	功能描述	读写
31:30	Reserved	-	-
29:24	DATAVD	数据有效延迟 用作 SDA 数据保持时间的最小周期数（减 1）。配置时间必须小于最小 SCL 低周期。	RW
23:22	Reserved	-	-
21:16	SETHOLD	建立保持延迟 主机在这些条件下使用的最小周期数（减 1）： 1. START 条件的保持时间 2. 重复 START 条件的建立和保持时间 3. STOP 条件的建立时间 建立时间会延长检测到外部 SCL 引脚上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，这段时间等于 $(2 + \text{FILT SCL}) / 2^{\text{PRESCALE}}$ 个周期。	RW
15:14	Reserved	-	-
13:8	CLKHI	时钟高周期 SCL 时钟被主机驱动为高电平的最小周期数（减 1）。 SCL 高时间延长为检测外部 SCL 引脚上的上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，该时间等于 $(2 + \text{FILT SCL}) / 2^{\text{PRESCALE}}$ 周期。	RW
7:6	Reserved	-	-
5:0	CLKLO	时钟低周期 SCL 时钟被主机驱动为低电平的最小周期数（减 1）。 时钟低周期值也用于 STOP 和 START 条件之间的最小总线空闲时间；这被扩展为检测外部 SCL 引脚上的上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，该时间等于 $(2 + \text{FILT SCL}) / 2^{\text{PRESCALE}}$ 周期。	RW

20.5.10 主机时钟配置 1 寄存器 (I2C_MCCR1)



说明

当 I2C 主设备启用并用于高速模式传输时，无法更改本寄存器。高速模式的单独时钟配置允许在切换到高速模式（由 MCCR1 配置时序）之前，在快速模式（由主时钟配置 0 (MCCR0) 配置时序）下进行仲裁。

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	DATAVD							Reserved	SETHOLD							Reserved	CLKHI							Reserved	CLKLO						
Access		RW								RW								RW								RW						

位/位域	标记	功能描述	读写
31:30	Reserved	-	-
29:24	DATAVD	数据有效延迟 用作 SDA 数据保持时间的最小周期数（减 1）。配置时间必须小于最小 SCL 低周期。	RW
23:22	Reserved	-	-
21:16	SETHOLD	建立保持延迟 主机在这些条件下使用的最小周期数（减 1）： 1. START 条件的保持时间 2. 重复 START 条件的建立和保持时间 3. STOP 条件的建立时间 建立时间会延长检测到外部 SCL 引脚上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，这段时间等于 $(2 + \text{FILTSCl}) / 2^{\text{PRESCALE}}$ 个周期。	RW
15:14	Reserved	-	-
13:8	CLKHI	时钟高周期 SCL 时钟被主机驱动为高电平的最小周期数（减 1）。 SCL 高时间延长为检测外部 SCL 引脚上的上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，该时间等于 $(2 + \text{FILTSCl}) / 2^{\text{PRESCALE}}$ 周期。	RW
7:6	Reserved	-	-
5:0	CLKLO	时钟低周期 SCL 时钟被主机驱动为低电平的最小周期数（减 1）。 时钟低周期值也用于 STOP 和 START 条件之间的最小总线空闲时间；这被扩展为检测外部 SCL 引脚上的上升沿所需的时间。忽略由于外部负载导致的任何额外板延迟，该时间等于 $(2 + \text{FILTSCl}) / 2^{\text{PRESCALE}}$ 周期。	RW

20.5.11 主机发送数据寄存器（I2C_MTDR）

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																				WO CMD		WO DATA									
Access																					WO		WO									

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10:8	CMD	命令数据 0b000：发送 DATA[7:0] 0b001：接收（DATA[7:0]+1）字节 0b010：生成 STOP 条件 0b011：接收和丢弃（DATA[7:0]+1）字节 0b100：生成（重复）START 和发送地址在 DATA[7:0] 0b101：生成（重复）START 和发送地址在 DATA[7:0]。此传输期望返回 NACK 0b110：在高速模式生成（重复）START 和发送地址在 DATA[7:0] 0b111：在高速模式生成（重复）START 和发送地址在 DATA[7:0] 中。此传输期望返回 NACK	WO
7:0	DATA	发送数据 对数据执行 8 位写入且零扩展 CMD 字段	WO

20.5.12 主机接收数据寄存器 (I2C_MRDR)



说明

读取接收数据寄存器会返回 I2C 主机接收的尚未丢弃的数据。

Offset	Bit Position																																
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00004000																																
Name	Reserved																		RXEMPTY	Reserved						DATA							
Access																			RO							RO							

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14	RXEMPTY	RX 空状态 RXEMPTY 给出主接收数据缓冲区的空状态 0b0: 接收缓冲区不为空 0b1: 接收缓冲区为空	RO
13:8	Reserved	-	-
7:0	DATA	接收数据 接收数据可以由于 CMD 字段而丢弃，或者可以将主机配置为丢弃不匹配的数据	RO

20.5.13 主机数据配置寄存器 (I2C_MDMR)



说明

仅当 I2C 主机被禁用或空闲时，才可以修改本寄存器。

Offset	Bit Position																																					
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved								MATCH1								Reserved								MATCH0													
Access									RW																RW													

位/位域	标记	功能描述	读写
31:24	Reserved	-	-
23:16	MATCH1	匹配 1 值 启用接收数据匹配时与接收数据进行比较	RW
15:8	Reserved	-	-
7:0	MATCH0	匹配 0 值 启用接收数据匹配时与接收数据进行比较	RW

20.5.14 主机 DMA 使能寄存器 (I2C_MDER)



说明

仅当 I2C 主机被禁用或空闲时，才可以修改本寄存器。

Offset	Bit Position																															
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												RDDE		TDDE	
Access																													RW		RW	

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	RDDE	接收 DMA 使能 ● 0b0：禁止 ● 0b1：使能	RW
0	TDDE	发送 DMA 使能 ● 0b0：禁止 ● 0b1：使能	RW

20.5.15 从机控制 0 寄存器 (I2C_SCR0)

Offset	Bit Position																																
0x110	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																																
Access																															RW	RST	SEN

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	RST	软件重置 ● 0b0：从机模式逻辑不重置 ● 0b1：从机模式逻辑重置 重置立即生效，并保持置位，直到被软件清除。在清除软件重置之前，不需要最小延迟。	RW
0	SEN	启用 I2C 从机模式 ● 0b0：禁用从机模式 ● 0b1：启用从机模式	RW

20.5.16 从机状态寄存器 (I2C_SSR)

Offset	Bit Position																															
0x114	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved						Reserved						RO SARF	RO GCF	RO AM1F	RO AM0F	RO FEF	RO BEF	RO SDF	RO RSF	Reserved						RO TAF	RO AVF	RO RDF	RO TDF		
Access													RO	RO	RO	RO	RO	RO	RO	RO							RO	RO	RO	RO		

位/位域	标记	功能描述	读写
31:26	Reserved	-	-
25	BBF	总线忙标志 ● 0b0: I2C 总线空闲 ● 0b1: I2C 总线忙	RO
24	SBF	从机忙标志 ● 0b0: I2C 从机空闲 ● 0b1: I2C 从机忙	RO
23:16	Reserved	-	-
15	SARF	SMBus 报警响应标志 ● 0b0: SMBus 报警响应被禁用或未检测到 ● 0b1: SMBus 报警响应被启用并检测到  说明 1. 通过读取从机地址状态寄存器 (SASR) 来清除 SARF。 2. SARF 无法生成异步唤醒。	RO
14	GCF	广播标志 指示从机是否检测到广播地址。 ● 0b0: 从机未检测到广播地址或广播地址被禁用 ● 0b1: 从机检测到广播地址  说明 1. 通过读取从机地址状态寄存器 (SASR) 来清除广播标志。 2. 广播标志无法生成异步唤醒。	RO
13	AM1F	地址匹配 1 标志 表示所接收的地址已与从机地址寄存器 (SADDR) 配置的 ADDR1 字段或 ADDR0 到 ADDR1 范围匹配。 ● 0b0: 尚未收到 ADDR1 或 ADDR0/ADDR1 范围匹配地址 ● 0b1: 已收到 ADDR1.ADDR0/ADDR1 区域匹配地址  说明 1. 读取从地址状态寄存器 (SASR) 会清除地址匹配 1 标志。 2. 地址匹配 1 标志无法生成异步唤醒。 当将 SCR1.ADDRMODE 配置为 0x6 或 0x7 时, 请使用地址 ADDR0/ADDR1 区域匹配, 如果 ADDR0>ADDR1, AM1F 将为 0。	RO

位/位域	标记	功能描述	读写
12	AM0F	地址匹配 0 标志 表示接收的地址已与 SCR1.ADDRM0DE 配置的 ADDR0 字段匹配。 0b0: 尚未接收到 ADDR0 匹配地址 0b1: 已接收到 ADDR0 匹配地址  说明 - 通过读取从机地址状态寄存器 (SASR) 来清除 AM0F。 - AM0F 无法生成异步唤醒。	RO
11	FEF	缓冲区错误标志 FEF 只能在禁用时钟延展时置位。 0b0: 未检测到缓冲区下溢或上溢 0b1: 检测到缓冲区下溢或上溢	RO
10	BEF	位错误标志 0b0: 从机未检测到位错误 0b1: 从机检测到位错误  说明 如果 I2C 从机发送逻辑 1 并在 I2C 总线上检测到逻辑 0, 则位错误标志 BEF 置位。从机忽略传输的其余部分, 直到下一个 (重复) START 条件。	RO
9	SDF	STOP 检测标志 0b0: 从机未检测到 STOP 0b1: 从机检测到 STOP SDF 在 I2C 从机检测到 STOP 条件且 I2C 从机与最后一个地址字节匹配时置位。	RO
8	RSF	重复 START 标志 0b0: 从机未检测到重复 START 条件 0b1: 从机检测到重复 START 条件 当 I2C 从机检测到重复 START 条件, 并且如果 I2C 从机匹配最后一个地址字节, 则置位重复启动标志 RSF。当从机首次检测到 START 条件时, RSF 不置位。	RO
7:4	Reserved	-	-
3	TAF	发送 ACK 标志 0b0: 不需要发送 ACK/NACK 0b1: 需要发送 ACK/NACK TAF 通过写入从机发送 ACK 寄存器 (STAR) 来清除。	RO
2	AVF	地址有效标志 0b0: 地址状态寄存器无效 0b1: 地址状态寄存器有效 通过读取从机地址状态寄存器 (SASR) 来清除地址有效标志 AVF。当 SCR1.RXCFG=1 时, SSR.AVF 也通过读取从机接收数据寄存器 (SRDR) 来清除。	RO
1	RDF	接收数据标志 0b0: 接收数据未就绪 0b1: 接收数据就绪 通过读取从机接收数据寄存器 (SRDR) 来清除接收数据标志 RDF。当 SCR1.RXCFG=1 时, 如果 SSR.AVF 被置位, 读取从机接收数据寄存器 (SRDR) 时, RDF 不会被清除。	RO

位/位域	标记	功能描述	读写
0	TDF	发送数据标志 ● 0b0：未请求发送数据 ● 0b1：请求发送数据 通过将写入从机发送数据寄存器（STDR）来清除发送数据标志 TDF。 当 SCR1.TXCFG=0 时，如果检测到 NACK 或重复 START 或 STOP 条件，则 TDF 也被清除。	RO

20.5.17 从机中断使能寄存器 (I2C_SIER)

Offset	Bit Position															
0x118	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0x00000000															
Name	Reserved															
Access	RW															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	SARIE	SMBus 报警响应中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
14	GCIE	广播中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
13	AM1IE	地址匹配 1 中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
12	AM0IE	地址匹配 0 中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
11	FEIE	缓冲错误中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
10	BEIE	位错误中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
9	SDIE	STOP 检测中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
8	RSIE	重复 START 中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
7:4	Reserved	-	-
3	TAIE	发送 ACK 中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW
2	AVIE	地址有效中断使能 ● 0b0: 禁用 ● 0b1: 使能	RW

位/位域	标记	功能描述	读写
1	RDIE	接收数据中断使能 ● 0b0：禁用 ● 0b1：使能	RW
0	TDIE	发送数据中断使能 ● 0b0：禁用 ● 0b1：使能	RW

20.5.18 从机状态清除寄存器 (I2C_SSCR)

Offset	Bit Position																																					
0x11C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000000																																					
Name	Reserved																				FEC		BEC		SDC		RSC		Reserved									
Access																					WO	WO	WO	WO														

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11	FEC	缓冲区错误标志清除 写 1 清除标志位，写 0 无效	WO
10	BEC	位错误标志清除 写 1 清除标志位，写 0 无效	WO
9	SDC	STOP 检测标志清除 写 1 清除标志位，写 0 无效	WO
8	RSC	重复 START 标志清除 写 1 清除标志位，写 0 无效	WO
7:0	Reserved	-	-

20.5.19 从机控制 1 寄存器 (I2C_SCR1)



说明

仅当 I2C 从机被禁用时，才可以修改本寄存器。

Offset	Bit Position																																				
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000000																																				
Name	Reserved														ADDRMODE		Reserved		HSMEN	IGNACK		RXCFG		TXCFG		SAEN		GCEN		Reserved		ACKSTALL	TXDSTALL	RXSTALL	ADRSTALL		
Access															RW				RW	RW		RW		RW		RW		RW		RW				RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:19	Reserved	-	-
18:16	ADDRMODE	地址模式 选择导致地址匹配的条件 0b000: 地址匹配 0 (7-bit) 0b001: 地址匹配 0 (10-bit) 0b010: 地址匹配 0 (7-bit) 或地址匹配 1 (7-bit) 0b011: 地址匹配 0 (10-bit) 或地址匹配 1 (10-bit) 0b100: 地址匹配 0 (7-bit) 或地址匹配 1 (10-bit) 0b101: 地址匹配 0 (10-bit) 或地址匹配 1 (7-bit) 0b110: 从地址匹配 0 (7-bit) 到地址匹配 1 (7-bit) 0b111: 从地址匹配 0 (10-bit) 到地址匹配 1 (10-bit)	RW
15:14	Reserved	-	-
13	HSMEN	高速模式启用 0b0: 禁用 HS 模式主机代码的检测 0b1: 启用 HS 模式主机代码的检测 从机地址 0000_1XX 的高速模式主机代码的检测，但不会导致该代码上的地址匹配。当置位并检测到任何 HS 模式主机代码时，SCR2.FILTEN 和 SCR1.ACKSTALL 位被忽略，直到检测到下一个 STOP 条件。	RW
12	IGNACK	忽略 NACK 0b0: 检测到 NACK 时从机结束传输 0b1: 检测到 NACK 时从机不结束传输 当置位忽略 NACK 时，I2C 从机在检测到 NACK 后继续传输。在超快速模式下需要置位忽略 NACK 位。	RW
11	RXCFG	接收数据配置 0b0: 读取从机接收数据寄存器返回接收的数据并清除接收数据标志 (SSR.RDF) 0b1: 在设置地址有效标志 (SSR.AVF) 时读取从机接收数据寄存器，返回从机地址状态寄存器并清除地址有效标志。当地址有效标志清除时读取从机接收数据寄存器，返回接收的数据并清除接收数据标志 (SSR.RDF)	RW
10	TXCFG	发送标志配置 0b0: 发送数据标志仅在从机发送过程中当从机发送数据寄存器为空时才被置位	RW

位/位域	标记	功能描述	读写
10	TXCFG	0b1: 每当从机发送数据寄存器为空时, 发送数据标志就会被置位。发送数据标志总是在检测到从机传输结束时的 NACK 之前被置位。这可能导致额外的一个字被写入发送数据缓冲区。 1. 当 TXCFG=0 时, 当检测到从机发送时, 从机发送数据寄存器自动清空。这使得每当检测到从机发送时, 发送数据标志置位, 并导致发送数据标志在从机发送结束时翻转。 2. 当 TXCFG=1 时, 每当从机发送数据寄存器为空, 发送数据标志置位, 当发送数据寄存器满时, 发送数据标志翻转。这允许在检测到从机发送传输之前填充发送数据寄存器, 但可以导致在从机发送传输的最后一个字节上检测到 NACK 之前写入发送数据寄存器。	RW
9	SAEN	SMBus 报警启用 0b0: 禁用 SMBus 报警匹配 0b1: 启用 SMBus 报警匹配	RW
8	GCEN	广播启用 0b0: 广播地址被禁用 0b1: 广播地址被启用	RW
7:4	Reserved	-	-
3	ACKSTALL	ACK SCL 暂停 0b0: 时钟延展禁用 0b1: 时钟延展启用 在从机发送地址字节和从机接收地址及数据字节期间, 启用 SCL 时钟延展, 以便软件在发送 ACK 或 NACK 之前写入从机发送 ACK 寄存器 (STAR)。时钟延展发生在发送第 9 位时, 因此与高速模式不兼容。当启用 ACKSTALL 时, 无需设置 SCR1.RXSTALL 或 SCR1.ADRSTALL。	RW
2	TXDSTALL	TX Data SCL 暂停 0b0: 时钟延展禁用 0b1: 时钟延展启用 当 SSR.TDF=1 时, 在从机发送传输期间启用 SCL 时钟延展。时钟延展发生在第 9 位之后, 因此与高速模式兼容。	RW
1	RXSTALL	RX SCL 暂停 0b0: 时钟延展禁用 0b1: 时钟延展启用 当 SSR.RDF=1 时, 在从机接收传输期间启用 SCL 时钟延展。时钟延展发生在第 9 位之后, 因此与高速模式兼容。	RW
0	ADRSTALL	Address SCL 暂停 0b0: 时钟延展禁用 0b1: 时钟延展启用 当 SSR.AVF=1 时启用 SCL 时钟延展。时钟延展仅在第 9 位之后发生, 因此与高速模式兼容。	RW

20.5.20 从机控制 2 寄存器 (I2C_SCR2)



说明

仅当 I2C 从机被禁用时，才可以修改本寄存器。

Offset	Bit Position																																	
0x124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved								FILTSDA				FILTSCS				Reserved	FILTEN	DATAVD								Reserved				CLKHOLD			
Access									RW				RW					RW	RW												RW			

位/位域	标记	功能描述	读写
31:24	Reserved	-	-
23:20	FILTSDA	毛刺滤波器 SDA 为 SDA 输入配置 I2C 从机数字毛刺滤波器。 1. 配置为 0 禁用毛刺滤波器 2. 等于或小于 FILTSDA 周期的毛刺被过滤掉并忽略 3. 通过毛刺滤波器的延迟等于 FILTSDA+2 个周期，并且必须配置为小于最小的 SCL 低或高周期 4. 毛刺滤波器周期计数不受 PRESCALE 配置的影响，并且在高速模式下禁用毛刺过滤器周期计数	RW
19:16	FILTSCS	毛刺滤波器 SCL 为 SCL 输入配置 I2C 从机数字毛刺滤波器。 1. 配置为 0 禁用毛刺滤波器 2. 等于或小于 FILTSCS 周期的毛刺被过滤掉并忽略 3. 通过毛刺滤波器的延迟等于 FILTSCS+2 个周期，并且必须配置为小于最小的 SCL 低或高周期 4. 毛刺过滤器周期计数不受 PRESCALE 配置的影响，并且在高速模式下禁用了毛刺过滤周期计数	RW
15	Reserved	-	-
14	FILTEN	过滤器启用 ● 0b0：禁用从机模式的数字滤波器和输出延时计数器 ● 0b1：启用从机模式的数字滤波器和输出延时计数器 FILTEN 应该只在 I2C 从机被禁用时更新。	RW
13:8	DATAVD	数据有效延迟 为 I2C 从机配置 SDA 数据有效延迟时间，并等于 FILTSCS+DATAVD+2 个周期。 1. 数据有效延迟必须配置为小于最小 SCL 低周期 2. I2C 从机数据有效延迟时间不受 PRESCALE 配置的影响，并且 I2C 从机数据有效延迟时间在高速模式下被禁用	RW
7:4	Reserved	-	-
3:0	CLKHOLD	时钟保持时间 配置 I2C 从机在启用时钟延展功能时的最小时钟保持时间。 1. 最小保持时间等于 CLKHOLD+3 个周期 2. I2C 从机时钟保持时间不受 PRESCALE 配置的影响，并且在高速模式下禁用 I2C 从机时钟保持时间	RW

20.5.21 从机发送 ACK 寄存器（I2C_STAR）

Offset	Bit Position																															
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															TXNACK
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	TXNACK	发送 NACK 0b0：为每个接收字写入发送 ACK 0b1：为每个接收的字写入发送 NACK 在接收到每个字后，软件可以发送 ACK（逻辑 0）或 NACK（逻辑 1）；发送 NACK 选择使用哪个：ACK 或 NACK - 当设置 SCR1.ACKSTALL 时，必须为每个匹配的地址字节和每个接收的字写入一次发送 NACK。必须设置 ACKSTALL，因为这会暂停数据传输，直到软件读取接收到的字（并决定是否用 ACK 或 NACK 响应） - 要配置默认 ACK/NACK，还可以在 I2C 从机被禁用或空闲时写入发送 NACK	RW

 **说明**

STAR 只能在 ACK SCL 暂停位置位（SCR1.ACKSTALL）时写入。SCR1.ACKSTALL 位在 ACK/NACK 位期间启用时钟延展，在此期间，可以通过软件写入 STAR 寄存器。该逻辑确保在 STAR 寄存器更新后，时钟延展至少持续 1 个总线时钟周期。使用时钟保持时间字段（SCR2.CLKHOLD）可以进一步延长该时钟延展时间。

20.5.22 从机发送数据寄存器（I2C_STDR）

Offset	Bit Position																															
0x12C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																								DATA							
Access																																

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7:0	DATA	发送数据 写入从机发送数据寄存器（STDR）将 I2C 从机发送数据存储在从机发送数据寄存器中	WO

 **说明**

时钟延展（启用或禁用）会影响传输数据的时间。SCR1.TXDSTALL 位在从机发送传输的第一个数据位期间启用时钟延展。

1.

如果启用时钟延展（SCR1.TXDSTALL=1），则发送数据传输暂停，直到 STDR 更新。在 STDR 更新后，时钟延展被扩展至少 1 个总线时钟周期，并且使用时钟保持时间字段（SCR2.CLKHOLD）延长时间。

2.

如果时钟延展被禁用（SCR1.TXDSTALL=0），则应在从机发送传输开始之前写入发送数据；否则（即，如果在从机发送传输开始之前未写入发送数据）缓冲区错误标志（SSR.FEF）置位。

20.5.23 从机接收数据寄存器（I2C_SRDR）

Offset	Bit Position																																		
0x130	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00004000																																		
Name	Reserved																			Reserved								DATA							
																	RO	SOF	RO	REMP															
Access																	RO											RO							

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	SOF	帧开始 表示这是否是第一个数据字 ● 0b0：表示这不是（重复）START 或 STOP 条件以来的第一个数据字 ● 0b1：表示这是自（重复）START 或 STOP 条件之后的第一个数据字	RO
14	REMP	RX 空 从机接收数据寄存器的空状态 ● 0b0：接收数据寄存器不为空 ● 0b1：接收数据寄存器为空	RO
13:8	Reserved	-	-
7:0	DATA	接收数据 I2C 从机接收的数据	RO

20.5.24 从机 DMA 使能寄存器 (I2C_SDER)



说明

仅当 I2C 从机被禁用时，才可以修改本寄存器。

Offset	Bit Position																																	
0x13C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																												RW		RW		RW	
Access																													RW		RW		RW	

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2	AVDE	地址 DMA 使能 地址 DMA 请求和接收 DMA 请求共用。如果同时使能地址 DMA 请求和接收 DMA 请求，配置 SCR1.RXCFG 为 1 后允许 DMA 从 SRDR 寄存器读取地址。	RW
1	RDDE	接收 DMA 使能 ● 0b0：禁止 ● 0b1：使能	RW
0	TDDE	发送 DMA 使能 ● 0b0：禁止 ● 0b1：使能	RW

20.5.25 从机地址寄存器（I2C_SADDR）



说明

仅当 I2C 从机被禁用时，才可以修改本寄存器。

Offset	Bit Position																															
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved				ADDR1								Reserved				ADDR0								Reserved							
Access					RW												RW															

位/位域	标记	功能描述	读写
31:27	Reserved	-	-
26:17	ADDR1	地址 1 值 与接收到的地址进行比较以检测从机地址。 - 在 10 位模式下，将第一地址字节与{11110，ADDR1[26:25]}进行比较，并将第二地址字节与 ADDR1[24:17]进行比较 - 在 7 位模式中，将地址与 ADDR1[23:17]进行比较	RW
16:11	Reserved	-	-
10:1	ADDR0	地址 0 值 与接收到的地址进行比较以检测从机地址。 - 在 10 位模式下，第一个地址字节与{11110，ADDR0[10:9]}进行比较，第二个地址字节则与 ADDR0[8:1]进行比较 - 在 7 位模式中，地址与 ADDR0[7:1]进行比较	RW
0	Reserved	-	-

20.5.26 从机地址状态寄存器（I2C_SASR）

Offset	Bit Position																															
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00004000																															
Name	Reserved																	ANV	Reserved	RADDR												
Access																		RO		RO												

位/位域	标记	功能描述	读写
31:15	Reserved	-	-
14	ANV	地址无效 ● 0b0：接收地址（RADDR）有效 ● 0b1：接收地址（RADDR）无效	RO
13:11	Reserved	-	-
10:0	RADDR	接收地址 每当设置 AMF 时，接收地址就会更新；通过读取从机地址状态寄存器来清除 AMF。 ● 在 7 位模式下，地址字节存储在 RADDR[7:0] ● 在 10 位模式中，第一地址字节为{11110，RADDR[10:9]，RADDR[0]}，第二地址字节为 RADDR[8:1]。因此，R/W 位始终存储在 RADDR[0]中	RO

21 串行外设接口 (SPI)

21.1 简介

串行外设接口 (SPI) 在主从模式下均提供了符合 SPI 通信协议的数据传输功能，可用于 MCU 和外部器件之间进行同步串行通信，本模块支持全双工、单线半双工及单工通信。

21.2 主要特性

- 可配置为主机或者从机，支持多机模式
- 主机模式最大分频系数为 PCLK/2
- 从机模式最大分频系数为 PCLK/4
- 多种通信模式：全双工、单线半双工、单工
- 两种传输顺序：先收发 MSB 或先收发 LSB
- 多种数据帧长度：4bits~16bits
- 两种 NSS 方式：硬件控制、软件控制
- 可配置的串行时钟极性和相位
- 支持 DMA
- 支持主机模式延后采样

21.3 功能说明

21.3.1 功能框图

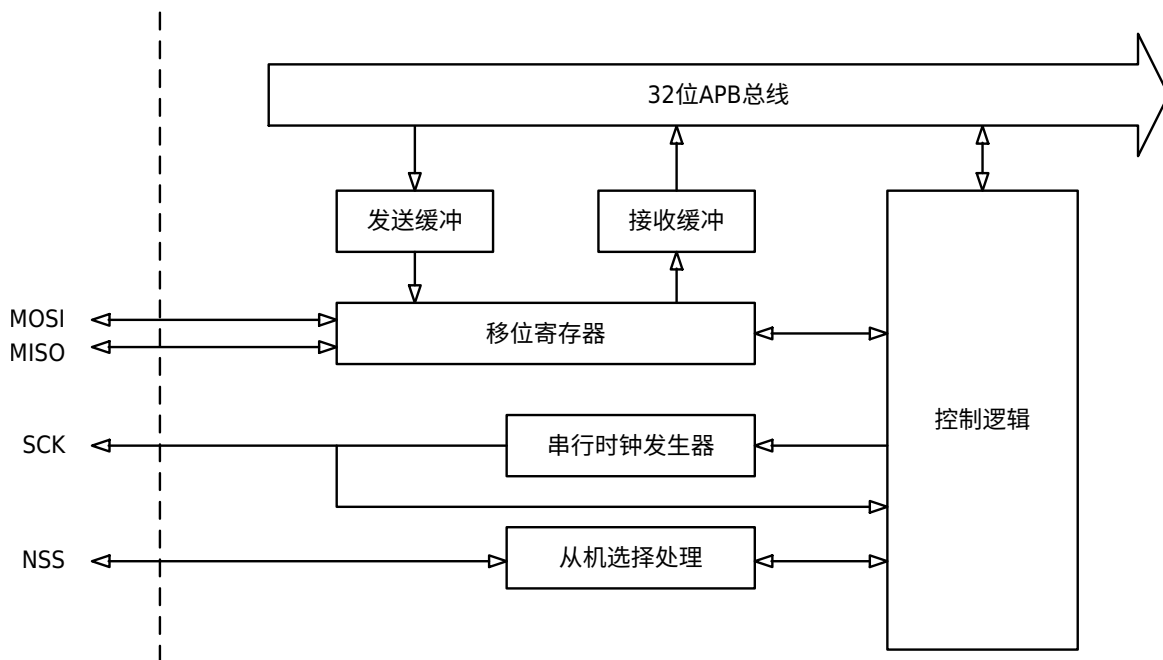


图 21-1 SPI 框图



说明

本章节描述涉及的 NSS 引脚即为《端口控制器 (GPIO)》章节“[端口数字复用功能](#)”中的 SPIx_CS。

21.3.2 通信格式和时序

SPI 串行帧格式取决于串行时钟极性 CPOL、串行时钟相位 CPHA 和数据帧格式。SPI 接口正常通信要求所有的主机和从机必须配置为相同的通信格式。

时钟极性 CPOL 是指 SPI 不传输数据的空闲状态下串行时钟的电平状态，当 SPI_CR0.CPOL 为 0 时，串行时钟 SCK 在空闲时为低电平；当 SPI_CR0.CPOL 为 1 时，串行时钟 SCK 在空闲时为高电平。

时钟相位 CPHA 是指 SPI 的主从机根据串行时钟 SCK 进行采样和移位的先后顺序，当 SPI_CR0.CPHA 为 0 时，主从机在前边沿采样数据，后边沿移位数据；当 SPI_CR0.CPHA 为 1 时，主从机在前边沿移位数据，后边沿采样数据。

通过设置 SPI_CR0.WIDTH 可配置每帧数据的宽度为 4bits 到 16bits。

通过设置 SPI_CR0.LSBF 可配置每帧数据的传输顺序为先传输 LSB 或先传 MSB。

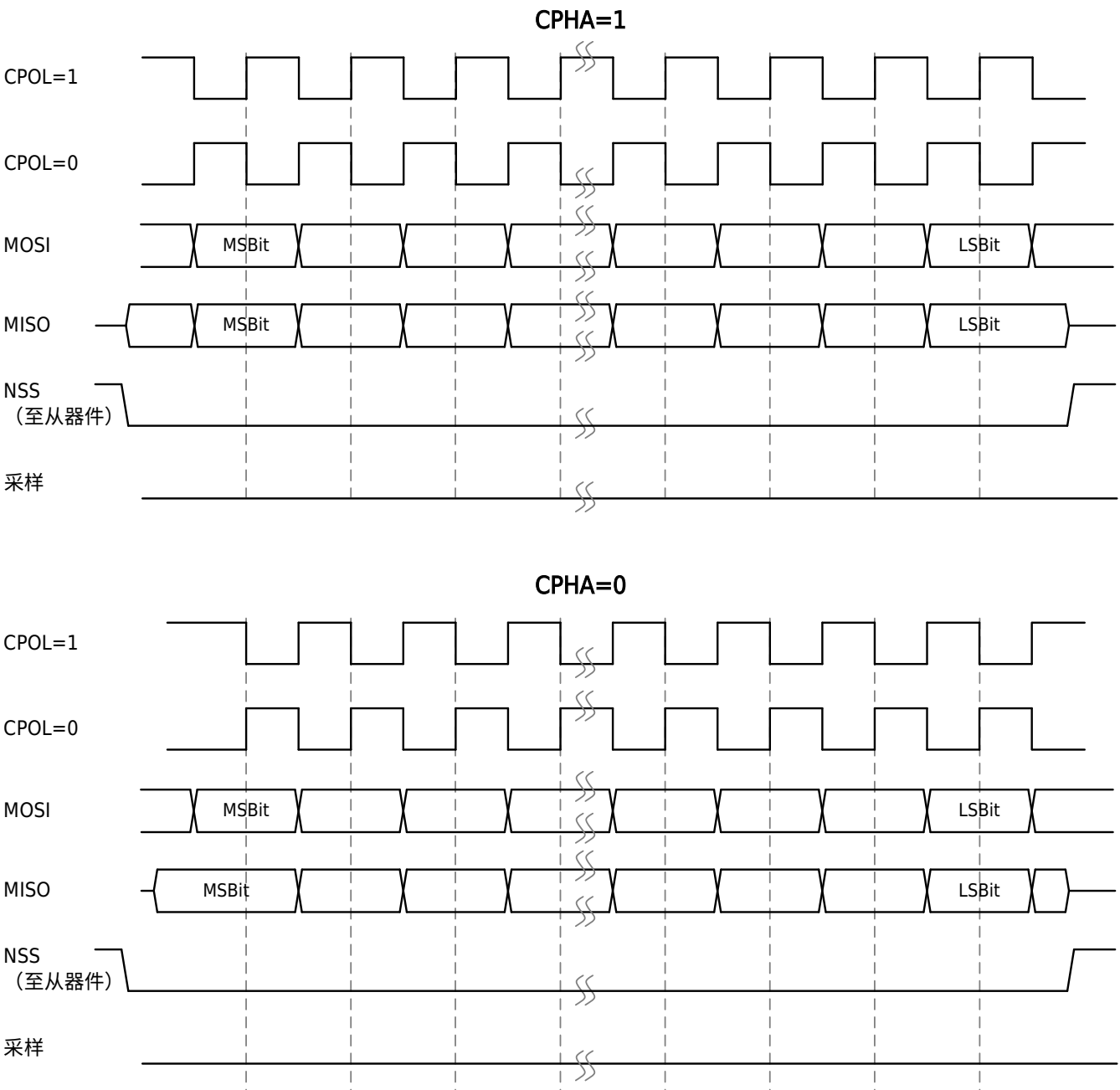


图 21-2 SPI 通信格式时序图

21.3.3 从机选择管脚配置

本模块提供了多种从机选择方式以适配不同的应用场景。

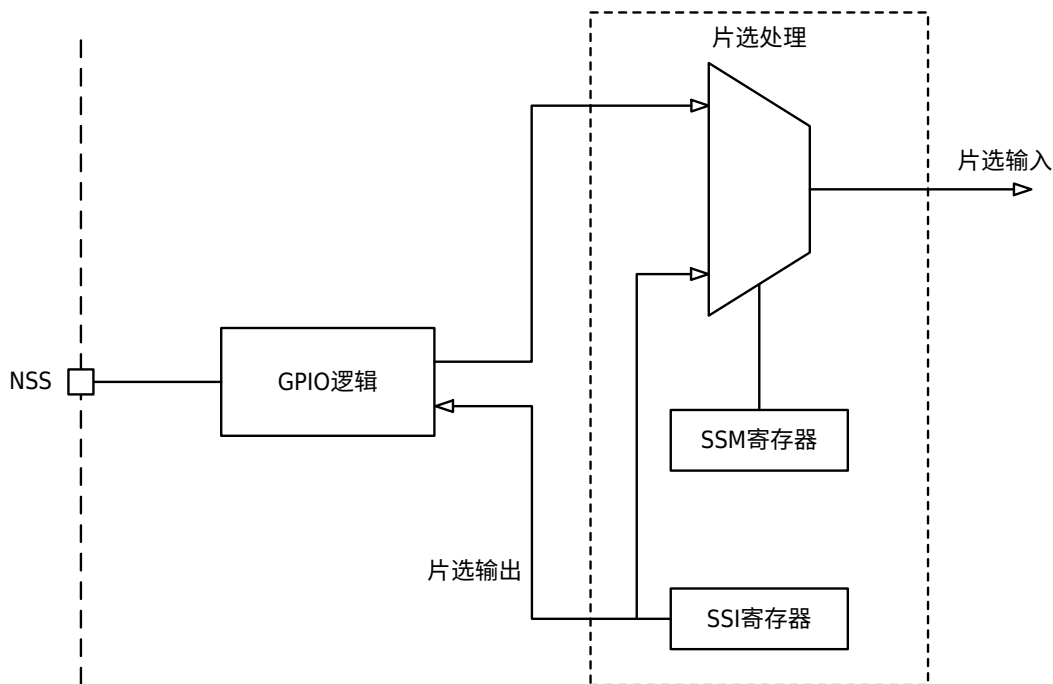


图 21-3 从机选择 NSS 管脚配置

- 当 SPI 工作于主机模式且 SPI_CR0.SSM 为 0 时，从机选择 NSS 是输入，其状态由 NSS 管脚的电平决定。
- 当 SPI 工作于主机模式且 SPI_CR0.SSM 为 1 时，从机选择 NSS 是输出，NSS 管脚的输出电平由内部寄存器 SPI_SSI.SSI 决定。当 SPI_SSI.SSI 为 0 时，NSS 管脚输出低电平；当 SPI_SSI.SSI 为 1 时，NSS 管脚输出高电平。
- 当 SPI 工作于从机模式且 SPI_CR0.SSM 为 0 时，NSS 管脚的低电平可选中本模块。
- 当 SPI 工作于从机模式且 SPI_CR0.SSM 为 1 时，NSS 管脚不再具有片选功能，寄存器 SPI_SSI.SSI 为 0 时可选中本模块。

21.3.4 全双工通信

全双工通信的方式，允许主从机依托两根单向的数据线进行同时收发双向的通信。该模式下，主机通过 SCK 脚提供时钟信号来同步数据的输入与输出。主从机 MOSI 脚相互连接，MISO 脚相互连接。主机通过 MOSI 脚把数据发送给从机，从机通过 MISO 管脚回传数据。每当用户向 SPI_DR 寄存器写入数据时，SPI 主机会通过 SCK、MOSI 将数据发送到 SPI 从机；并通过 SCK、MISO 接收来自从机的数据。

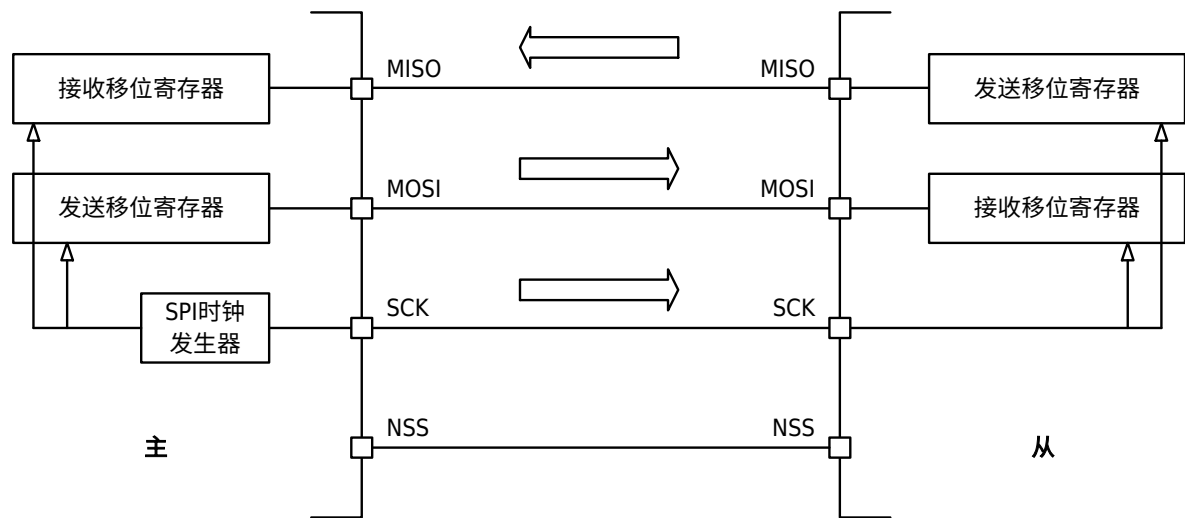


图 21-4 SPI 全双工通信

21.3.5 单线半双工通信

单线半双工通信的方式，允许主从机依托一根双向的数据线进行分时收发双向的通信。作为主机时使用的管脚是 MOSI，作为从机时使用的管脚是 MISO。数据传输方向由 SPI_HDOE.HDOE 决定，当 SPI_HDOE.HDOE 为 1 时，数据线进行发送。未用到的管脚不参与 SPI 通信，可供其他功能使用。当主机工作于接收状态时，需要向 SPI_DR 寄存器写入任意数据以触发 SPI 主机发送 SCK 进行数据传输。当主从机之间进行单线半双工通信时，由于收发的切换可能不同步，它们对于双向的共用数据线可能出现驱动冲突。为避免此类情况对器件可能产生的损害，建议在数据线上串入限流电阻以保护芯片管脚。

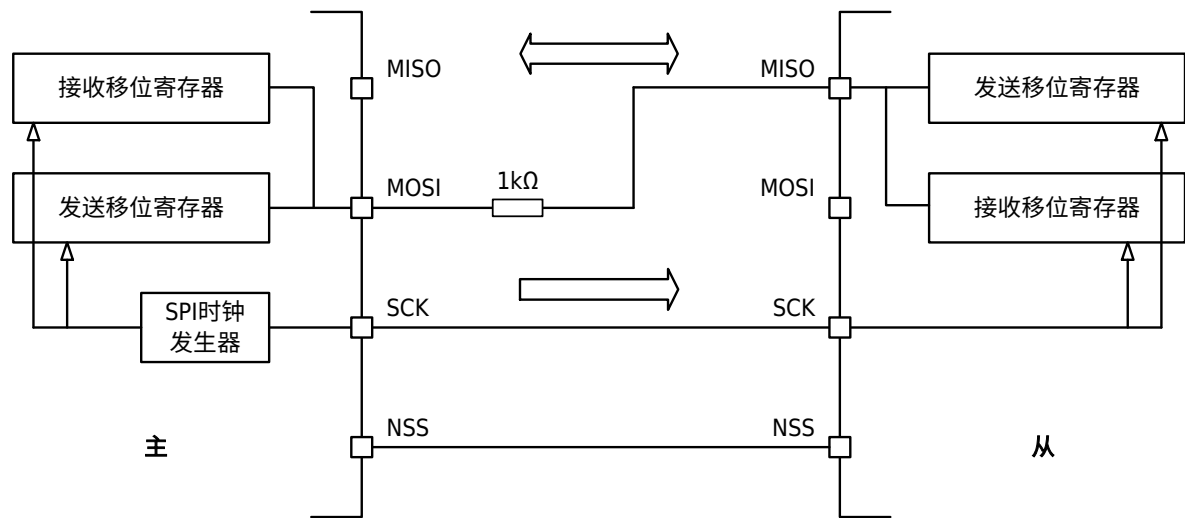


图 21-5 SPI 单线半双工通信

21.3.6 单工通信

单工通信的方式，允许主从机依托一根单向的数据线进行固定接收或固定发送的单向的通信，通过设置 SPI_CR0.CM 选择单工只发或单工只收模式。未用到的管脚不参与 SPI 通信，可供其他功能使用。当主机工作于发送状态，从机工作于接收状态时，主机向 SPI_DR 寄存器写入数据以触发 SPI 主机发送 SCK 进行数据传输。

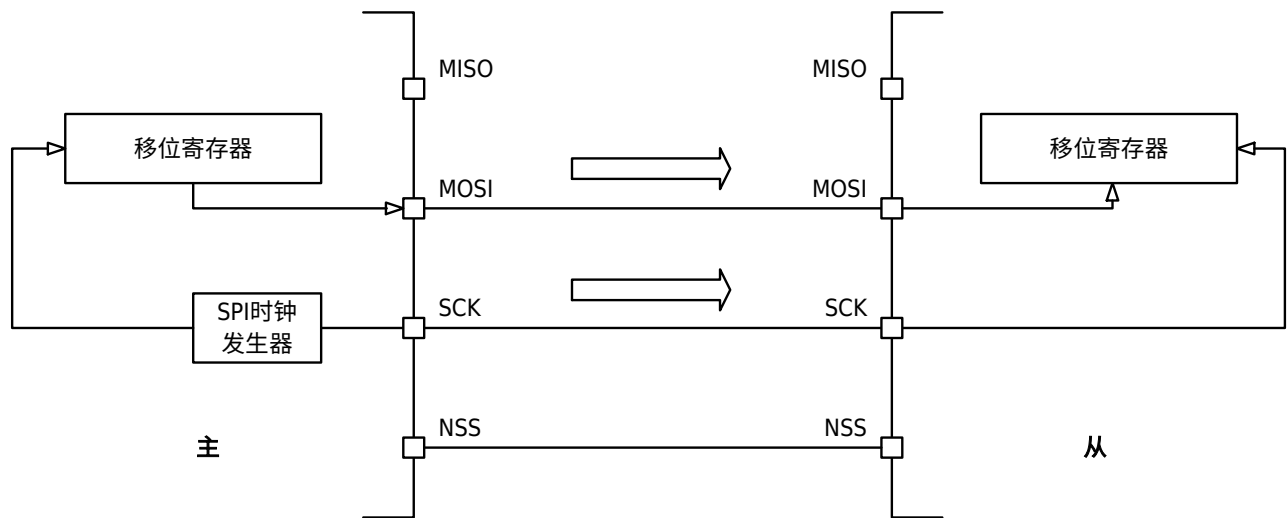


图 21-6 SPI 单工发送通信示意

当主机工作于接收状态，从机工作于发送状态时，也需要主机向 SPI_DR 寄存器写入任意数据以触发 SPI 主机发送 SCK 进行数据传输。

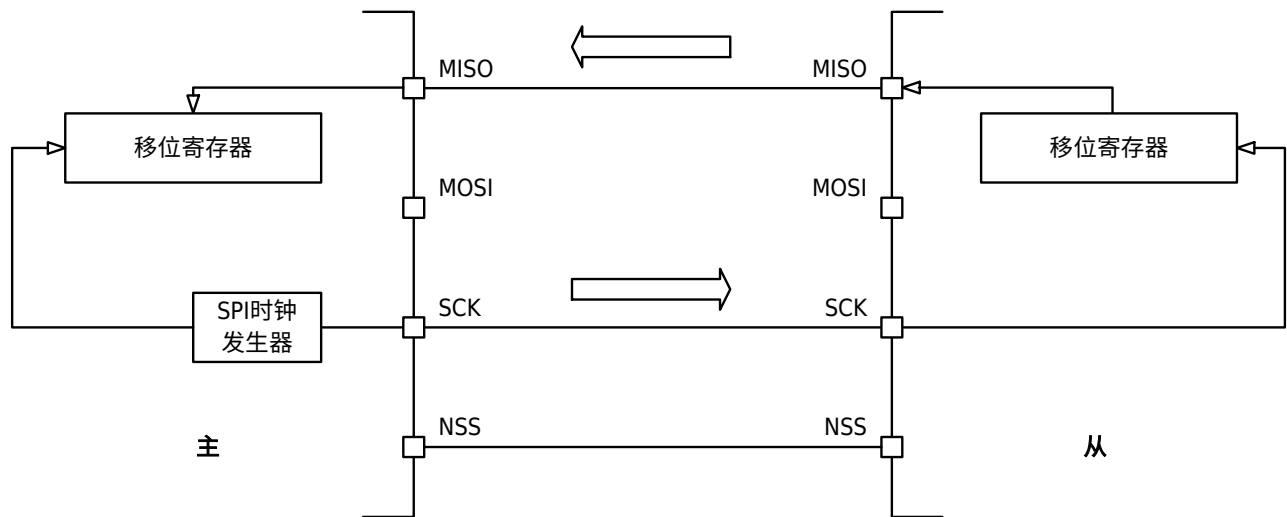


图 21-7 SPI 单工接收通信示意

 **说明**

- NSS 管脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些管脚。之后，必须在内部为主器件和从器件处理硬件控制流。
- 该配置下，两个闲置的管脚（MOSI 或者 MISO）均可用作 GPIO。

21.3.7 多机通信

多机通信的系统中，主机的从机选择 NSS 管脚配置为硬件输入（SPI_CR0.SSM 为 0），由 GPIO 负责输出从机选择信号，且主机的 NSS 输入都由另一台主机的 GPIO 驱动。主机的从机选择 NSS 管脚一旦被其他主机的 GPIO 拉低选中，就会产生模式错误 MODF，会造成本模块 SPI 强制返回未使能状态（即 SPI_CR0.EN 被清零），且主从模式选择寄存器（SPI_CR0.MSTR）也会被清零（即切换到从机模式），当系统需要再次使用 SPI 通信的时候，需要重新复位并初始化该模块。因此，当主机再次尝试获取 SPI 总线控制权时，需要检测 NSS 输入是否为高，若为高且没有 MODF 的情况下，可以成功获取总线控制权。多个从机的系统中，每台从机共享串行时钟和数据线，但是需要各自独立的从机选择信号，这就要求主机通过不同的 GPIO 向每个从机输出对应的从机选择信号。

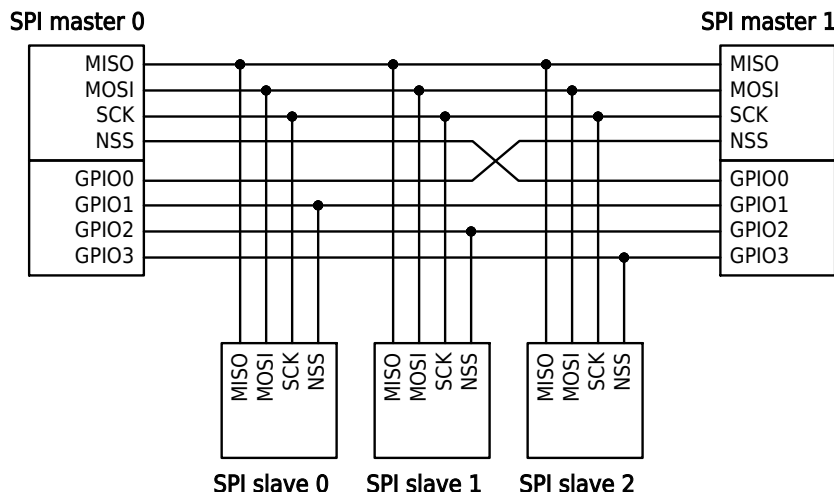


图 21-8 SPI 多机系统

21.3.8 主机模式延后采样

在 SPI 通信中，由于存在以下因素：

- 从机从收到串行时钟 SCK 的移位沿到 MISO 发出数据所需的时间；
- 数据从从机数据输出 MISO 到主机数据输入 MISO 所经历的传播时间。

主机收到有效数据的时刻，相比于串行时钟的移位沿，有一定的延时。在传统的 SPI 协议中，主机要在移位沿之后半个 SCK 周期的采样沿对 MISO 数据进行采样，这就要求上述的延时必须小于半个 SCK 周期，且半个 SCK 周期与该延时的差值要满足 MISO 建立时间的需求。因此，当上述的延时较为明显时，SPI 通信的速率会被制约，速率较高会出现数据错误。

本模块提供了主机模式延后采样的功能，能有效缓解上述问题带来的通信速率瓶颈问题。通过将寄存器 SPI_CR0.SMP 置 1，开启主机模式延后采样。此时，本模块作为 SPI 主机，会以传统 SPI 标准的采样沿为基准，延后半个 SCK 周期进行采样。这样既为整个系统提供了更高的对延迟的容忍度，又不会影响下一个比特的数据传输，从而达到更高的通信速率。



说明

1. 此功能仅在主机模式下有效，从机模式下无此功能。
2. SPI_CR0.SMP 的值并不影响主机和从机的通信格式配置——无论是否开启本功能，主机和从机都要按照常规 SPI 的要求，配置成相同的 CPOL 和 CPHA。

21.3.9 状态标志和错误标志

SPI 提供多种状态标志位来指示当前的工作状态，包括一般状态标志和错误标志。

从机选择输入 (NSS) 状态

从机选择输入状态指示当前输入的从机选择信号的电平，对应寄存器 SPI_SR.SSLVL。当 SPI_SR.SSLVL 为 0 时，从机选择输入是低电平；当 SPI_SR.SSLVL 为 1 时，从机选择输入是高电平。

总线忙标志位

总线忙标志位指示当前 SPI 接口处于忙碌还是空闲状态，对应寄存器 SPI_SR.BUSY。当 SPI_SR.BUSY 为 0 时，SPI 接口处于空闲状态，当前没有数据传输；当 SPI_SR.BUSY 为 1 时，SPI 接口处于忙碌状态，当前正在准备进行或正在进行数据传输。

从机选择输入上升沿标志位

从机选择输入上升沿标志位指示当前输入的从机选择信号是否出现了上升沿，对应寄存器 SPI_SR.SSR。当 SPI_SR.SSR 为 0 时，从机选择输入未出现上升沿；当 SPI_SR.SSR 为 1 时，从机选择输入出现了上升沿。

从机选择输入下降沿标志位

从机选择输入下降沿标志位指示当前输入的从机选择信号是否出现了下降沿，对应寄存器 SPI_SR.SSF。当 SPI_SR.SSF 为 0 时，从机选择输入未出现下降沿；当 SPI_SR.SSF 为 1 时，从机选择输入出现了下降沿。

发送缓冲空标志位

发送缓冲空标志位指示当前 SPI 的发送缓冲是否已装载待发送数据，对应寄存器 SPI_SR.TXE。当 SPI_SR.TXE 为 0 时，SPI 的发送缓冲已装载待发送数据；当 SPI_SR.TXE 为 1 时，SPI 的发送缓冲处于空状态，可以将下一个待发送的数据写入。

接收缓冲非空标志位

接收缓冲非空标志位指示当前 SPI 的接收缓冲是否存在已接收的数据，对应寄存器 SPI_SR.RXNE。当 SPI_SR.RXNE 为 0 时，SPI 的接收缓冲尚未装载有效的接收数据；当 SPI_SR.RXNE 为 1 时，SPI 的接收缓冲装载了已接收的数据，可供读出。

从机模式下的从机选择错误标志位

当 SPI 工作于从机模式且正在进行有效的数据传输时，输入的从机选择信号要保持选中状态（低电平）来保证正常的通信，如果从机选择在此期间被拉高，就会发生从机选择错误，对应的寄存器 SPI_SR.SSERR 会置位。

发生从机选择错误，会造成对应的数据传输出错，SPI 接口也会因从机选择的拉高而进入未选中状态。从机选择错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是，从机选择错误 SSERR 是只出现于从机模式下的错误，且仅当有效数据尚在传输时从机选择被拉高才会触发，帧与帧之间没有有效数据传输时正常的从机选择拉高并不触发该错误。

接收缓冲上溢错误标志位

SPI 的接收缓冲器的大小为一帧，当接收缓冲中的数据没有被读出且新一帧的数据已经接收完成时，发生的就是接收缓冲上溢错误，对应的寄存器 SPI_SR.OVF 会置位。

发生接收缓冲上溢错误，缓冲器中原有数据会被新接收的数据覆盖，造成数据的丢失。接收缓冲上溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

从机模式下的发送缓冲下溢错误标志位

SPI 的发送缓冲器的大小为一帧，当从机模式下发送缓冲已空且新一帧的传输开始时，发生的就是发送缓冲下溢错误，对应的寄存器 SPI_SR.UDF 会置位。

发生从机模式下发送缓冲下溢错误，意味着发送缓冲器中没有及时填入数据，错过了给主机发送数据。从机模式下缓冲下溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

主机模式下的模式错误标志位

当 SPI 工作于主机模式且从机选择管脚被配置为输入时，输入的从机选择信号要一直保持非选中状态（高电平）来保证正常的通信，如果从机选择被拉低，就会发生模式错误，对应的寄存器 SPI_SR.MODF 会置位。

发生模式错误，会造成 SPI 强制返回未使能状态（即 SPI_CR0.EN 被清零），主从模式选择寄存器（SPI_CR0.MSTR）也会被清零（即切换到从机模式）。模式错误发生后，对应的错误标志位置位并不阻止用户程序重新使能 SPI 并进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是，模式错误 MODF 是只出现于主机模式特定配置下的错误，仅当从机选择被配置为输入且被拉低才会触发，其他情况不会触发该错误。

CPOL=1.CPHA=1时的主模式示例

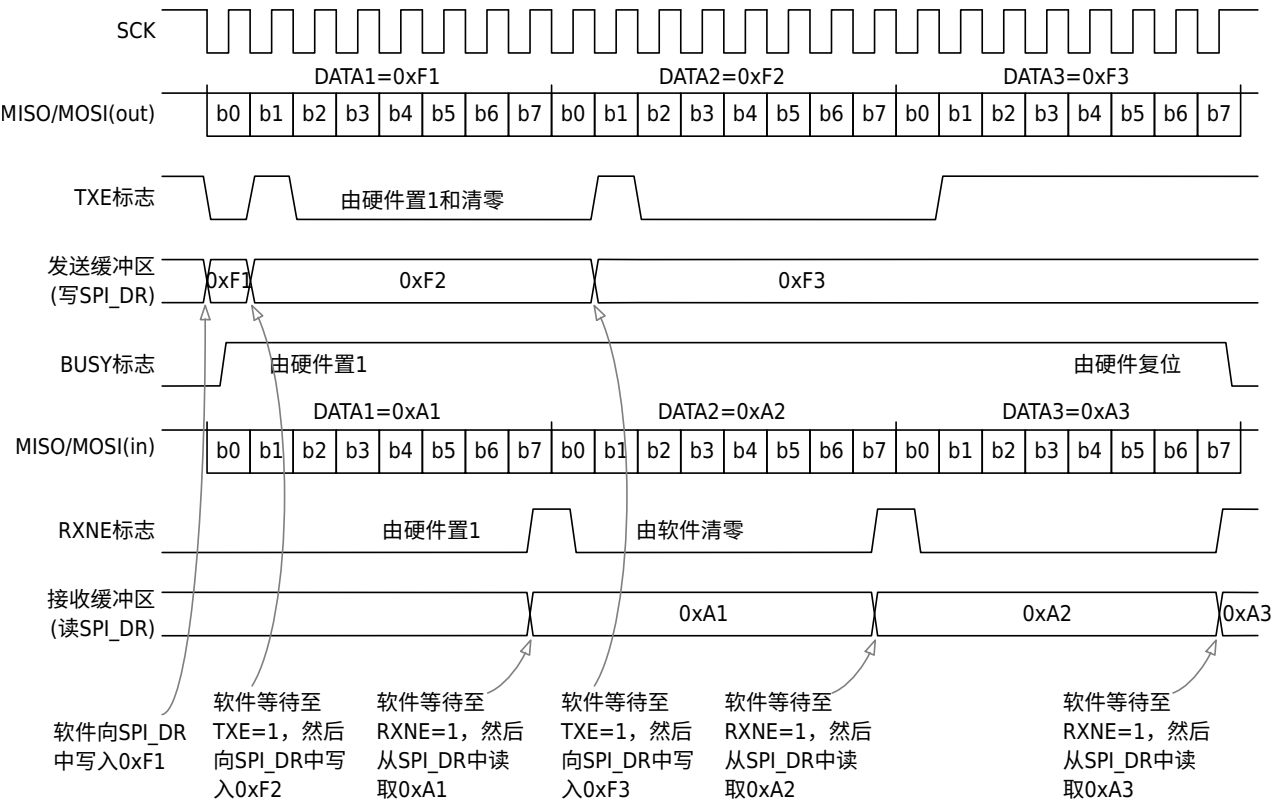


图 21-9 SPI 主机模式一般标志位说明图

CPOL=1.CPHA=1时的从模式示例

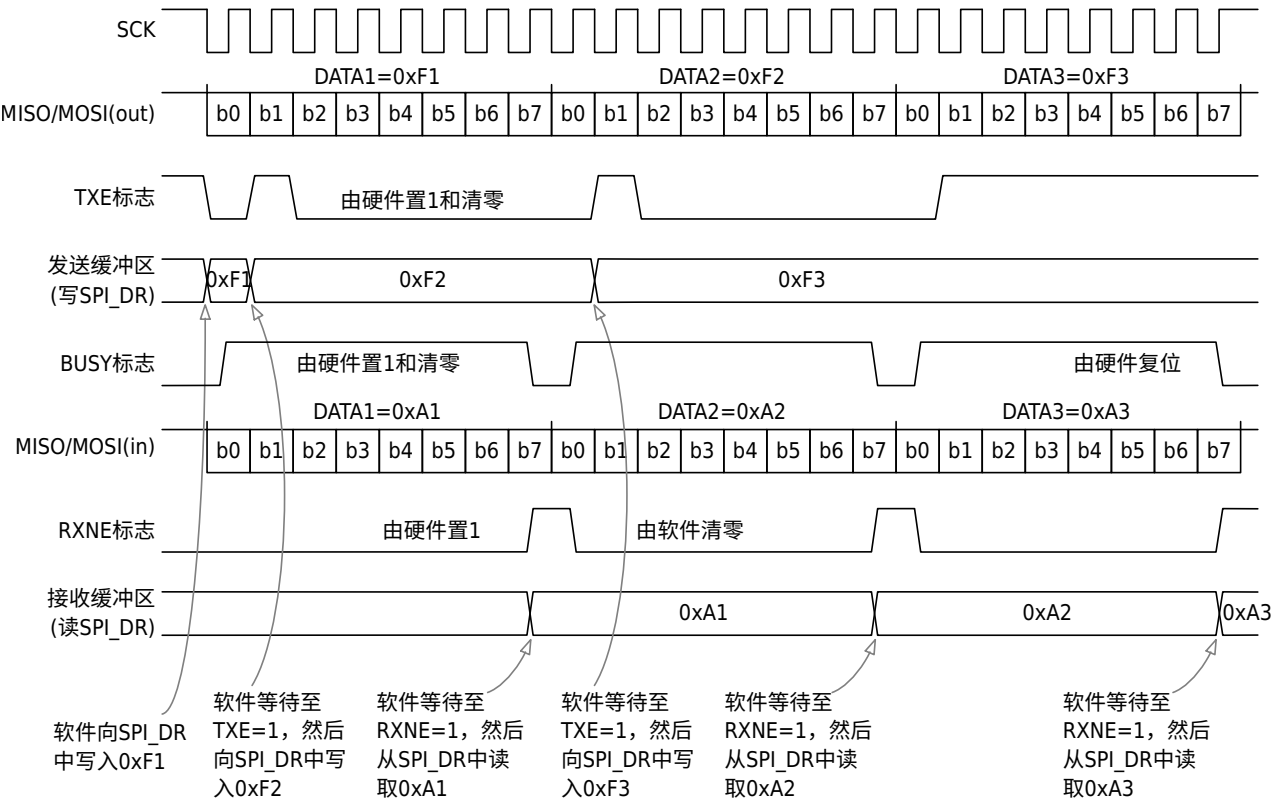


图 21-10 SPI 从机模式一般标志位说明图

21.3.10 中断的产生和清除

SPI 支持从不同的事件产生中断请求，具体信息如下：

表 21-1 SPI 中断

中断事件	对应标志位 (SPI_SR)	对应中断使能 (SPI_CR1)	清除中断标志的方法
从机选择输入上升沿	SSR	SSRIE	向 SPI_ICR.SSR 写 0
从机选择输入下降沿	SSF	SSFIE	向 SPI_ICR.SSF 写 0
发送缓冲空	TXE	TXEIE	向数据寄存器 DR 中写数据
接收缓冲非空	RXNE	RXNEIE	从数据寄存器 DR 中读数据或向 SPI_ICR.RXNE 写 0
从机模式下从机选择错误	SSERR	SSERRIE	向 SPI_ICR.SSERR 写 0
接收缓冲上溢错误	OVF	OVFIE	向 SPI_ICR.OVF 写 0
从机模式发送缓冲下溢错误	UDF	UDFIE	向 SPI_ICR.UDF 写 0
主机模式下模式错误	MODF	MODFIE	向 SPI_ICR.MODF 写 0

如上表，每种中断事件都有独立的中断使能控制，其中由错误事件产生的中断也都有独立的中断清除寄存器。对于上述的中断事件（或者请求），无论中断是否使能，都可以使用表中所述的方法来清除中断标志。

21.4 操作示例

21.4.1 主机全双工收发数据示例

- 步骤 1. 按《端口控制器（GPIO）》章节管脚数字复用功能的相关描述，将 NSS、SCK、MOSI、MISO 映射到需要的管脚。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.SPIx 为 1，使能 SPIx 配置时钟及工作时钟。
- 步骤 3. 设置 CR0.MSTR 为 1，选择主机模式。
- 步骤 4. 设置 CR0.BRR，配置通信波特率。
- 步骤 5. 设置 CR0.CM 为 0，选择全双工通信方式。
- 步骤 6. 设置 CR0.SSM 为 1，配置从机选择模式。
- 步骤 7. 设置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位。
- 步骤 8. 设置 CR0.WIDTH，配置每帧的数据宽度。
- 步骤 9. 设置 CR0.LSBF，配置收发数据时的数据序列顺序。
- 步骤 10. 设置 CR0.EN 为 1，使能 SPI 模块。
- 步骤 11. 设置 SSI.SSI 为 0，选中对应的从机。
- 步骤 12. 将待发送数据写入 DR 寄存器，SPI 自动开始数据传输。
- 步骤 13. 查询等待 SR.RXNE 变为 1，完成一帧数据的发送与接收。
- 步骤 14. 从 DR 寄存器中读出接收到的数据并保存。
- 步骤 15. 若待发送与接收的数据未完成，则跳转到步骤 12 继续执行。

21.4.2 从机全双工收发数据示例

- 步骤 1. 按《端口控制器（GPIO）》章节管脚数字复用功能的相关描述，将 NSS、SCK、MOSI、MISO 映射到需要的管脚。

- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.SPIx 为 1，使能 SPIx 配置时钟及工作时钟。
- 步骤 3. 设置 CR0.MSTR 为 0，选择从机模式。
- 步骤 4. 设置 CR0.CM 为 0，选择全双工通信方式。
- 步骤 5. 设置 CR0.SSM，配置从机选择模式。
- 步骤 6. 设置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位。
- 步骤 7. 设置 CR0.WIDTH，配置每帧的数据宽度。
- 步骤 8. 设置 CR0.LSBF，配置收发数据时的数据序列顺序。
- 步骤 9. 设置 CR0.EN 为 1，使能 SPI 模块。
- 步骤 10. 设置 ICR.TXE 为 0，清空发送缓冲区。
- 步骤 11. 查询等待 SR.SSLVL 变为 0，SPI 主机选中本模块。
- 步骤 12. 将待发送的数据写入 DR 寄存器，准备好待发送的数据。
- 步骤 13. 查询等待 SR.RXNE 变为 1，完成一帧数据的发送与接收。
- 步骤 14. 从 DR 寄存器中读出接收到的数据并保存。
- 步骤 15. 若待发送与接收的数据未完成，则跳转到步骤 12 继续执行。
- 步骤 16. 等待 SR.SSLVL 变为 1，完成本次通信。

21.4.3 单线半双工发送接收数据示例

- 步骤 1. 按《端口控制器 (GPIO)》章节管脚数字复用功能的相关描述，将主机的 NSS、SCK、MOSI 和从机的 NSS、SCK、MISO 映射到需要的管脚。
- 步骤 2. 设置 SYSCTRL 模块寄存器 PERI_CLKEN0.SPIx 为 1，使能 SPIx 配置时钟及工作时钟。
- 步骤 3. 主机设置 CR0.MSTR 为 1，从机设置 CR0.MSTR 为 0。
- 步骤 4. 主机设置 CR0.BRR，配置通信波特率。
- 步骤 5. 设置 CR0.CM 为 0b11，选择单线半双工通信方式。
- 步骤 6. 配置 CR0.SSM，配置从机选择模式。
- 步骤 7. 配置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位。
- 步骤 8. 配置 CR0.WIDTH，配置每帧的数据宽度。
- 步骤 9. 配置 CR0.LSBF，配置发送数据时的数据序列顺序。
- 步骤 10. 设置 CR0.EN 为 1，使能 SPI 模块。
- 步骤 11. 主机设置 SSI.SSI 为 0，选中对应的从机；从机等待 SR.SSLVL 变为 0 选中本模块。
- 步骤 12. 主机设置 HDOE.HDOE 为 1，切换至发送模式；从机设置 HDOE.HDOE 为 0，切换至接收模式。
- 步骤 13. 主机将待发送数据写入 DR 寄存器，SPI 自动开始数据传输。
- 步骤 14. 主机查询等待 SR.BUSY 变为 0，完成一帧数据的发送；从机查询等待 SR.RXNE 变为 1，完成一帧数据的接收。
- 步骤 15. 从机从 DR 寄存器中读出接收到的数据并保存。
- 步骤 16. 若主机待发送的数据未完成，则跳转到步骤 13 继续执行。
- 步骤 17. 主机设置 HDOE.HDOE 为 0，切换至接收模式；从机设置 HDOE.HDOE 为 1，切换至发送模式。
- 步骤 18. 从机查询等待 SR.TXE 变为 1，发送缓存空闲。
- 步骤 19. 若从机待发送的数据未完成，则向 DR 寄存器写入一帧数据。
- 步骤 20. 主机向 DR 寄存器写入任意数据，SPI 主机发送 SCK 以进行数据传输。
- 步骤 21. 主机查询等待 SR.RXNE 变为 1，完成一帧数据的接收。
- 步骤 22. 主机从 DR 寄存器中读出接收到的数据并保存。
- 步骤 23. 若从机待发送的数据未完成，则跳转到步骤 18 继续执行。
- 步骤 24. 从机等待 SR.SSLVL 变为 1，设置 HDOE.HDOE 为 0，完成本次通信。

21.5 寄存器

21.5.1 寄存器总表

- SPI0 基地址：0x40000800
- SPI1 基地址：0x40008800

表 21-2 串行外设接口（SPI）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	SPI_CR0	SPI 控制寄存器 0	32
0x04	SPI_CR1	SPI 控制寄存器 1	32
0x08	SPI_HDOE	SPI 单线半双工输出使能寄存器	32
0x0C	SPI_SSI	SPI 内部从机选择寄存器	32
0x10	SPI_SR	SPI 状态寄存器	32
0x14	SPI_ICR	SPI 中断清除寄存器	32
0x18	SPI_DR	SPI 数据寄存器	32

21.5.2 SPI 控制寄存器 0 (SPI_CR0)

Offset	Bit Position																																					
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x00000700																																					
Name	Reserved																				WIDTH								BRR									
																	SMP	CM	SSM	LSBF					MSTR	CPOL	CPHA	EN										
Access																	RW		RW		RW			RW		RW		RW		RW			RW			RW		

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15	SMP	主机模式延后采样位 0b0: SPI 传统方式, 主机接收数据按照 CPOL 和 CPHA 的设定进行采样 0b1: 延后采样方式, 采样时刻相比传统方式延后半个 SCK 周期	RW
14:13	CM	通信方式配置 0b00: 全双工双向 0b01: 单工只发 0b10: 单工只收 0b11: 单线半双工	RW
12	SSM	从机选择配置 主机模式: 0b0: NSS 管脚决定从机选择输入值 0b1: SSI 寄存器决定从机选择输出值 从机模式: 0b0: NSS 管脚决定从机选择输入值 0b1: SSI 寄存器决定从机选择输入值	RW
11:8	WIDTH	帧数据宽度配置 0b0011: 4-bit 0b0100: 5-bit 0b1110: 15-bit 0b1111: 16-bit - 其他: 禁止设置	RW
7	LSBF	数据帧高低位顺序选择 0b0: 最高有效位 MSB 收发在前 0b1: 最低有效位 LSB 收发在前	RW
6	MSTR	工作模式配置 0b0: 从机模式 0b1: 主机模式	RW
5	CPOL	串行时钟极性配置 0b0: 待机时低电平 0b1: 待机时高电平	RW

位/位域	标记	功能描述	读写
4	CPHA	串行时钟相位配置 ● 0b0: 前边沿采样/后边沿移位 ● 0b1: 前边沿移位/后边沿采样	RW
3:1	BRR	主机模式波特率配置 ● 0b000: PCLK/2 ● 0b001: PCLK/4 ● 0b010: PCLK/8 ● 0b011: PCLK/16 ● 0b100: PCLK/32 ● 0b101: PCLK/64 ● 0b110: PCLK/128 ● 0b111: 禁止设置	RW
0	EN	使能控制 ● 0b0: 禁止 SPI 模块 ● 0b1: 使能 SPI 模块	RW

21.5.3 SPI 控制寄存器 1 (SPI_CR1)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																						DMATXEN	DMARXEN	MODFIE	SSERRIE	OVFIE	UDFIE	SSRIE	SSFIE	RXNEIE	TXEIE
Access																							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:10	Reserved	-	-
9	DMATXEN	发送 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
8	DMARXEN	接收 DMA 使能 ● 0b0: 禁止 ● 0b1: 使能	RW
7	MODFIE	模式错误中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
6	SSERRIE	从机模式下的从机选择错误中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
5	OVFIE	接收缓冲上溢错误中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
4	UDFIE	从机模式下发送缓冲下溢错误中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
3	SSRIE	从机选择输入上升沿中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
2	SSFIE	从机选择输入下降沿中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
1	RXNEIE	接收缓冲非空中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
0	TXEIE	发送缓冲空中断使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW

21.5.4 SPI 单线半双工输出使能寄存器 (SPI_HDOE)

Offset	Bit Position																															
0x08	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															HDOE
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	HDOE	MOSI/MISO 收发切换控制 0b0: 只收 0b1: 只发  说明 仅在单线半双工通信时有效。	RW

21.5.5 SPI 内部从机选择寄存器 (SPI_SSI)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000001																															
Name	Reserved																															SSI
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	SSI	内部从机选择 主机模式下寄存器决定从机选择输出值 从机模式下寄存器决定从机选择输入值  说明 当 SPI_CR0.SSM 寄存器为 1 时有效。	RW

21.5.6 SPI 状态寄存器 (SPI SR)

Offset	Bit Position																																											
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Reset	0x00000201																																											
Name	Reserved																								RO	SSLVL	RO	BUSY	RO	MODF	RO	SSERR	RO	OVF	RO	UDF	RO	SSR	RO	SSF	RO	RXNE	RO	TXE
Access																									RO	SSLVL	RO	BUSY	RO	MODF	RO	SSERR	RO	OVF	RO	UDF	RO	SSR	RO	SSF	RO	RXNE	RO	TXE

位/位域	标记	功能描述	读写
31:10	Reserved	-	-
9	SSLVL	从机选择输入的状态 ● 0b0: 从机选择输入是低电平 ● 0b1: 从机选择输入是高电平  说明 当主机模式且 SPI_CR0.SSM=1 时，该位设置无效。	RO
8	BUSY	总线忙标志位 ● 0b0: 总线空闲 ● 0b1: 总线忙	RO
7	MODF	模式错误标志位 ● 0b0: 正常 ● 0b1: 出错  说明 主机模式下从机选择输入为低则触发 MODF。	RO
6	SSERR	从机模式下的从机选择错误标志位 ● 0b0: 正常 ● 0b1: 出错	RO
5	OVF	接收缓冲上溢错误标志位 ● 0b0: 正常 ● 0b1: 出错	RO
4	UDF	从机模式下的发送缓冲下溢错误标志位 ● 0b0: 正常 ● 0b1: 出错	RO
3	SSR	从机选择输入上升沿标志位 ● 0b0: 未出现上升沿 ● 0b1: 出现了上升沿  说明 当主机模式且 SPI_CR0.SSM=1 时，该位设置无效。	RO

位/位域	标记	功能描述	读写
2	SSF	从机选择输入下降沿标志位 ● 0b0: 未出现下降沿 ● 0b1: 出现了下降沿  说明 当主机模式且 SPI_CR0.SSM=1 时，该位设置无效。	RO
1	RXNE	接收缓冲非空标志位 ● 0b0: 接收缓冲空 ● 0b1: 接收缓冲非空	RO
0	TXE	发送缓冲空标志位 ● 0b0: 发送缓冲非空 ● 0b1: 发送缓冲空	RO

21.5.7 SPI 中断清除寄存器 (SPI_ICR)

Offset	Bit Position																															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000000FF																															
Name	Reserved																								MODF	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE
Access																									WO	WO	WO	WO	WO	WO	WO	WO

位/位域	标记	功能描述	读写
31:8	Reserved	-	-
7	MODF	SR.MODF 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
6	SSERR	SR.SSERR 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
5	OVF	SR.OVF 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
4	UDF	SR.UDF 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
3	SSR	SR.SSR 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
2	SSF	SR.SSF 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
1	RXNE	SR.RXNE 标志清除位 写 0 清除标志, 写 1 无效; 读出恒为 1	WO
0	TXE	SR.TXE 清除位 写 0 设置 SR.TXE 标志为 1, 写 1 无效; 读出恒为 1	WO

21.5.8 SPI 数据寄存器 (SPI_DR)

Offset	Bit Position																															
0x18	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																DR <															

位/位域	标记	功能描述	读写
31:16	Reserved	-	-
15:0	DR	SPI 数据位 发送数据时, 将待发送数据写入该寄存器 接收数据时, 从该寄存器读出已收到的数据	RW

22 模数转换器 (ADC)

22.1 简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。本产品内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器 (SAR ADC) 模块。

22.2 主要特性

- 12 位转换精度
- 最大 2Msps 转换速度
- 最多 19 路输入通道，包括 16 路外部管脚输入、1 路内部温度传感器电压、1 路 1/3 AVCC 电压、1 路内置 1.2V 基准电压 VREF1P2
- 4 种参考源：AVCC 电压、EXVREF 引脚、内置 1.5V 参考电压、内置 2.5V 参考电压
- ADC 的电压输入范围：0~Vref
- 内置信号跟随器，可转换高阻信号
- 多种转换模式：单次转换、SQR 扫描转换、JQR 扫描转换
- 软件可配置 ADC 的转换速率
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性
- ADC 转换结果连续累加
- 输入通道电压阈值监测

22.3 功能说明

22.3.1 ADC 框图

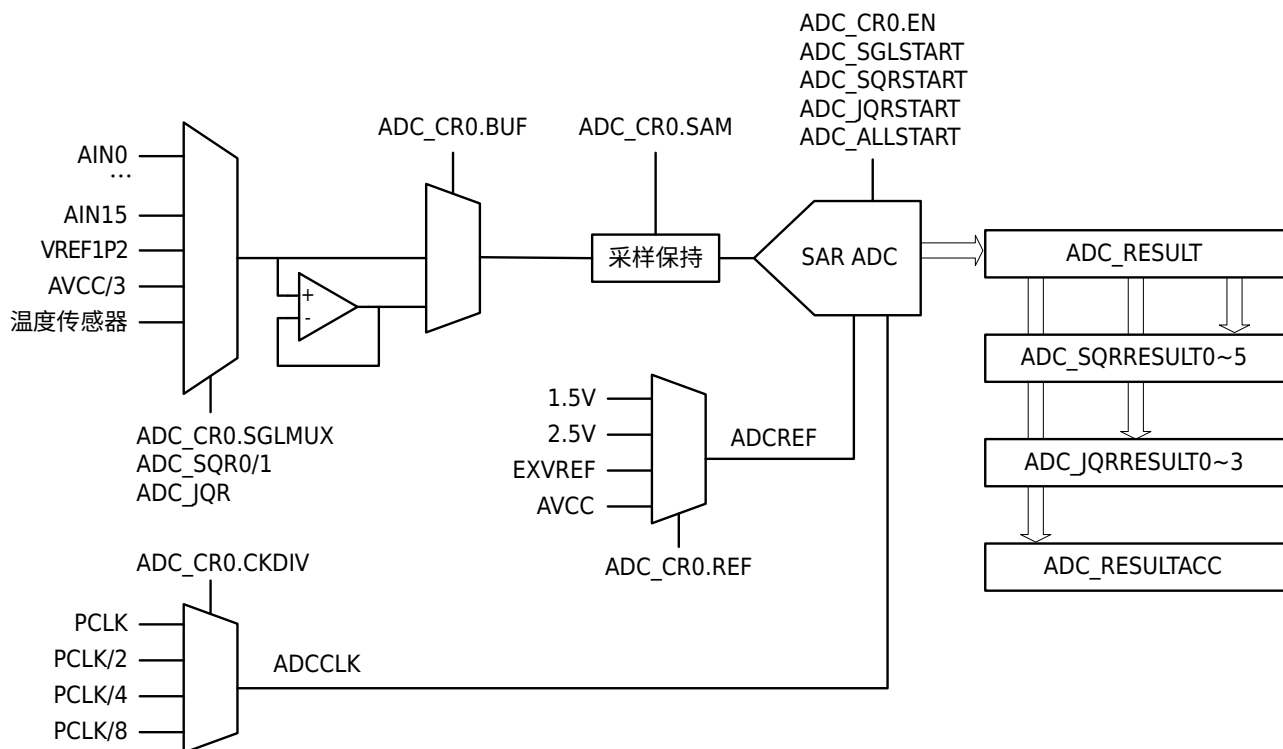


图 22-1 ADC 示意框图

22.3.2 转换时序及转换速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由采样过程及逐次比较过程组成。其中采样过程需要 2~127 个 ADCCLK，由 ADC_CR0.SAM 配置；逐次比较过程需要 12 个 ADCCLK。所以，一次 ADC 转换共需要 14~139 个 ADCCLK。

ADC 转换速度的单位为 sps，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADCCLK 的频率/一次 ADC 转换所需要的 ADCCLK 的个数。

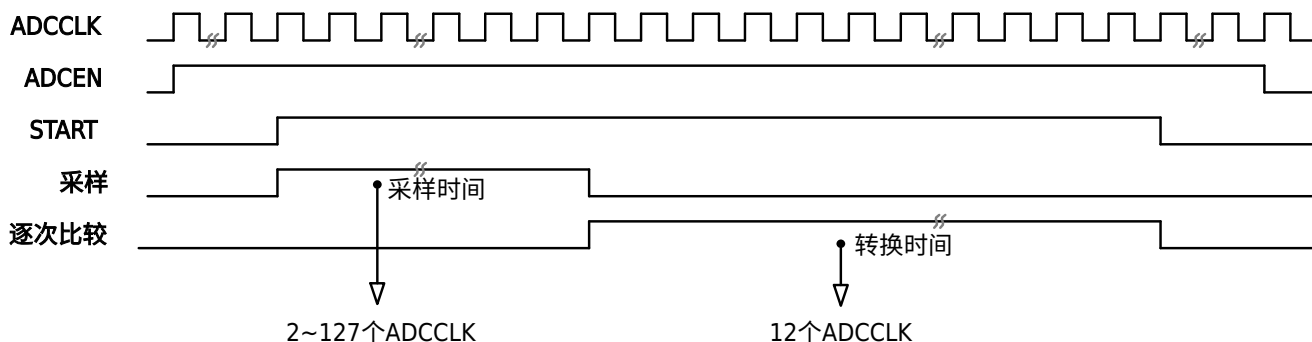


图 22-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关，详细信息请参见《数据手册》的“电气特性”章节相关描述。

22.3.3 单次转换模式

在单次转换模式下，ADC 启动后只执行一次采样转换，可对所有 ADC 通道进行转换。该模式既可通过设置 ADC_SGLSTART.START 位软启动，也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_IFR.SGLIF 位会自动置 1，转换结果保存在 ADC_RESULT 寄存器中。

如果需要一直转换某个通道，可以将 ADC_ALLSTART.START 位设置为 1。如果需要停止可以将 ADC_ALLSTART.START 位设置为 0，或者直接关闭 ADC (ADC_CR0.EN=0)。

22.3.4 扫描转换模式

扫描转换模式分为 SQR 扫描转换和 JQR 扫描转换两种模式。两种模式各自单独工作时，均可连续对多个通道进行转换；JQR 序列可以在 SQR 序列工作时进行插队转换，具体行为参见表 22-1 描述。

SQR 扫描转换模式

SQR 扫描转换模式最多可进行 6 次连续转换，转换的总次数由 ADC_SQR1.CNT 进行配置；可配置所有通道进行转换，待转换通道由 ADC_SQR0.CHxMUX 进行配置。该模式既可通过设置 ADC_SQRSTART.START 位软启动也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。启动转换后，ADC 模块依次转换 ADC_SQR0.CHxMUX~ADC_SQR0.CH0MUX 中配置的通道直到完成总转换次数。ADC 模块完成总转换次数后，ADC_IFR.SQRIF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_SQRRESULTx~ADC_SQRRESULT0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 6 次转换的 SQR 扫描转换过程。其中 SQR 扫描转换通道 4/1 配置为 AIN1，转换通道 3/0 配置为 AIN0，转换通道 5/2 配置为 AIN5。ADC_SQRSTART.START 置 1 后，ADC 模块会依次对 SQR 扫描转换通道 5~0 进行转换。

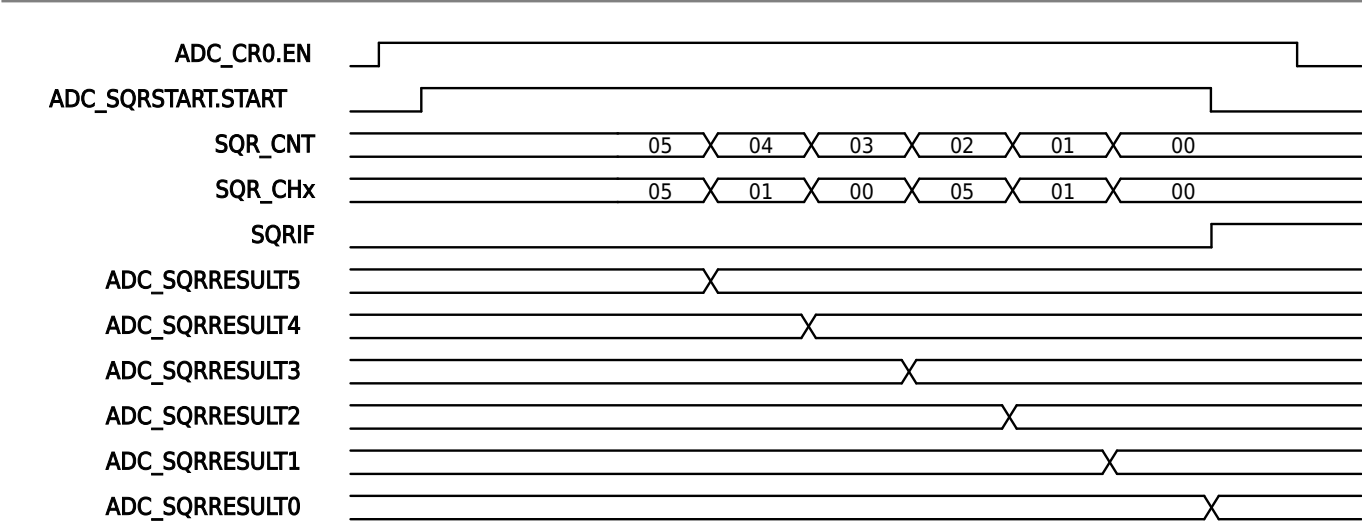


图 22-3 ADC SQR 扫描转换过程示例

JQR 扫描转换模式

JQR 扫描转换模式最多可进行 4 次连续转换，转换的总次数由 ADC_JQR.CNT 进行配置；可配置所有通道进行转换，待转换通道由 ADC_JQR.CHxMUX 进行配置。该模式既可通过设置 ADC_JQRSTART.START 位启动也可通过设置 ADC_EXTTRIGGER1 的外部触发启动。该模式可以在 SQR 扫描转换队列中插入，也可以独立开启。启动转换后，ADC 模块依次转换 ADC_JQR.CHxMUX~ADC_JQR.CH0MUX 中配置的通道直到完成总转换次数。ADC 模块完成总转换次数后，ADC_IFR.位会自动置 1，转换结果保存在转换通道所对应的 ADC_JQRRESULTx~ADC_JQRRESULT0 寄存器中。下图演示了对 AIN0、AIN1、AIN5 进行 4 次转换的 JQR 扫描转换的过程。其中 JQR 扫描转换通道 3、2、1、0 分别设置为 AIN5、AIN0、AIN1、AIN5。ADC_JQRSTART.START 置 1 后，ADC 模块会依次对 JQR 扫描转换通道 3~0 进行转换。

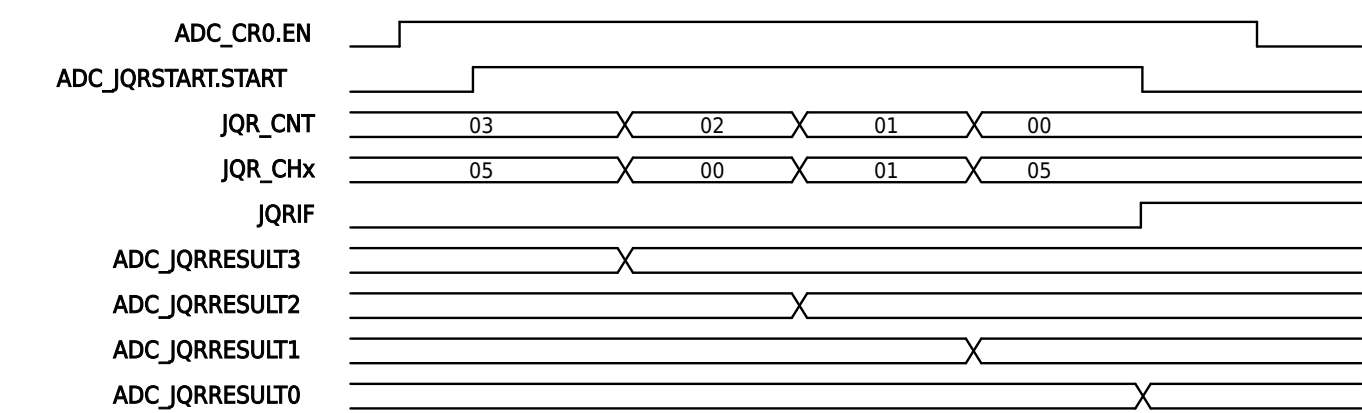


图 22-4 ADC JQR 扫描转换过程示例

JQR 扫描转换的执行优先级高于 SQR 扫描转换，当 SQR 扫描转换正在进行时，如果 JQR 扫描转换启动，则 SQR 扫描在完成当前通道的转换后暂停，等到 JQR 扫描转换完成后，再继续执行剩下的通道转换。

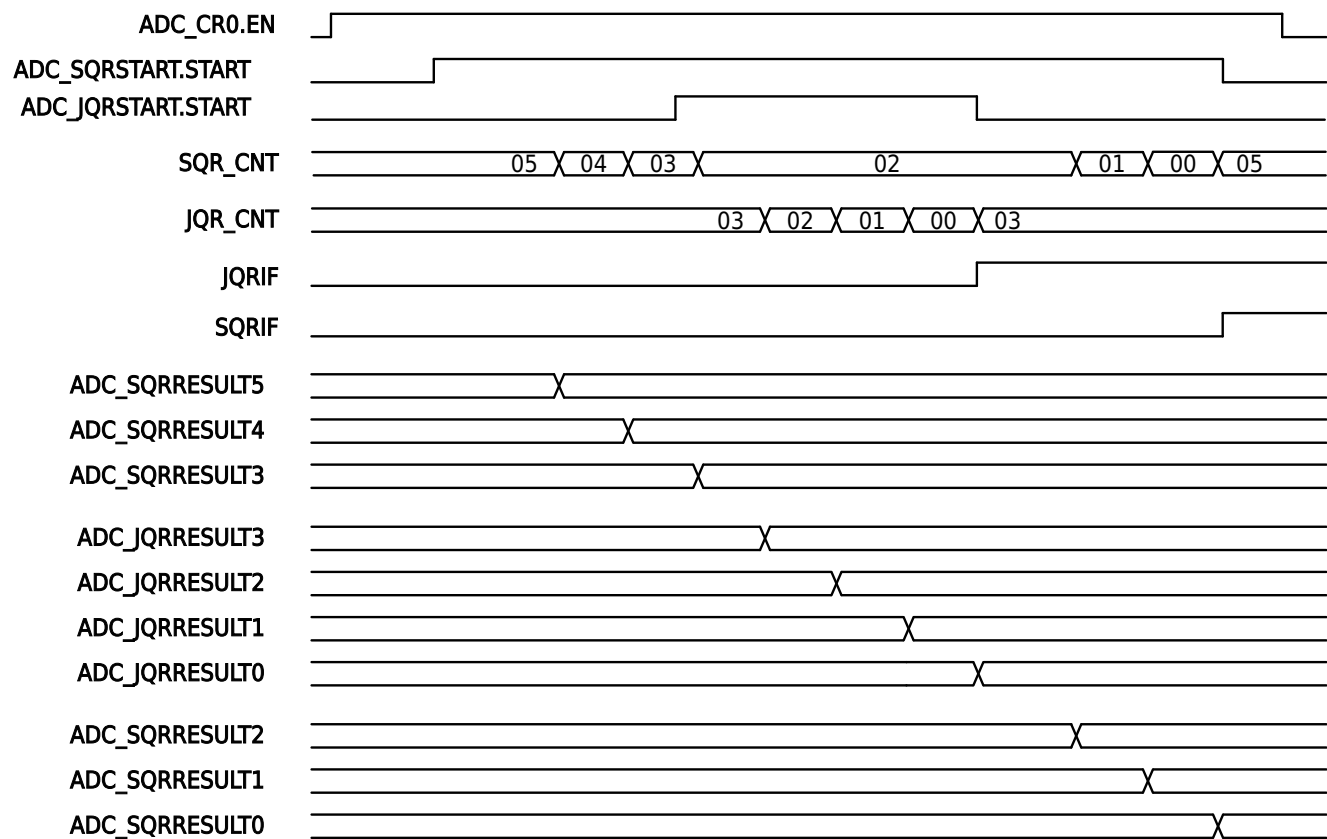


图 22-5 ADC SQR 扫描过程中进行 JQR 扫描示例

表 22-1 插队行为状态表

当前队列状态	插队行为	
	启动 SQR 扫描	启动 JQR 扫描
JQR 扫描中	无响应，继续当前 JQR 扫描	无响应，继续当前 JQR 扫描
SQR 扫描中	无响应，继续当前 SQR 扫描	完成当前通道转换后切换到 JQR 扫描，JQR 扫描转换完成后继续之前的 SQR 队列



说明

若同时触发 JQR&SQR 扫描，JQR 扫描优先，JQR 扫描完成后再执行 SQR 扫描。

扫描转换触发 DMA 读取

SQR 扫描和 JQR 扫描转换完成后，可自动触发 DMA 读取转换结果。SQR 扫描模式下，通过配置 ADC_CR1.DMASQR 为 1 使能该功能；JQR 扫描模式下，通过配置 ADC_CR1.DMAJQR 为 1 使能该功能。

22.3.5 ADC 转换结果累加

通过设置 ADC_CR1.RACCEN 位使能转换累加功能，在此模式下可对 ADC 每次转换的结果进行累加，转换结果的累加值保存在 ADC_RESULTACC 寄存器中。下图以 SQR 模式下的连续累加过程举例，其中，对 AIN0、AIN1、AIN5 进行总计 6 次连续转换。SQR 扫描转换通道 3/0 配置为 AIN0，转换通道 5/2 配置为 AIN1，转换通道 4/1 配置为 AIN5。ADC_SQRSTART.START 置 1 后，ADC 模块会按照 SQR 扫描转换通道配置依次进行转换，直到 SQR_CNT 的计数值变为 0。每次转换完成时，ADC_RESULTACC 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

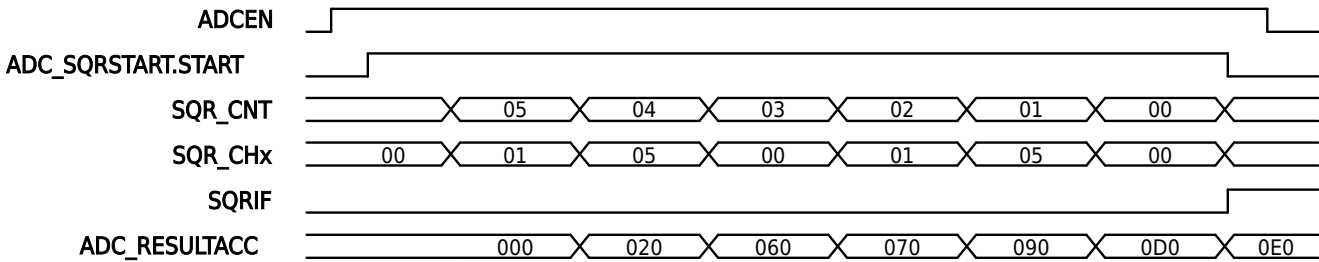


图 22-6 ADC 连续转换累加过程示例

22.3.6 ADC 转换外部触发源

ADC 转换既可通过软件配置启动，也可通过外部触发启动。
配置 ADC_EXTTRIGGER0 寄存器可设置 ADC 单次转换或 SQR 扫描转换的外部触发源。
配置 ADC_EXTTRIGGER1 寄存器可设置 ADC JQR 扫描转换的外部触发源。
详细的触发源请参见 [ADC_EXTTRIGGER0](#) 和 [ADC_EXTTRIGGER1](#) 寄存器相关描述。

22.3.7 ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 ADC_CR1.HTCMP、ADC_CR1.LTCMP、ADC_CR1.REGCMP 置 1。该功能可实现对模拟量的自动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序介入。监测通道选择通过 ADC_CR1.THCH 进行配置。

- 上阈值比较：当 ADC 转换结果位于[ADC_HT, 4095]区间内则 ADC_IFR.HTIF 置 1；向 ADC_ICR.HTIC 写入 0 则清零 ADC_IFR.HTIF。
- 下阈值比较：当 ADC 转换结果位于[0, ADC_LT)区间内则 ADC_IFR.LTIF 置 1；向 ADC_ICR.LTIC 写入 0 则清零 ADC_IFR.LTIF。
- 区间值比较：当 ADC 转换结果位于[ADC_LT, ADC_HT)区间内则 ADC_IFR.REGIF 置 1；向 ADC_ICR.REGIC 写入 0 则清零 ADC_IFR.REGIF。

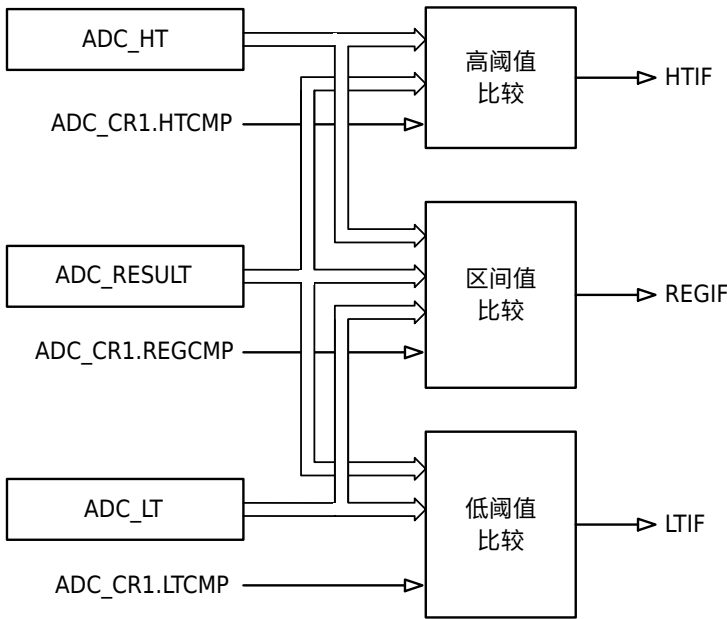


图 22-7 ADC 转换结果比较示意图

22.3.8 ADC 中断

ADC 中断请求如下表所示：

表 22-2 ADC 中断

中断源	中断标志	中断使能
ADC JQR 扫描转换完成	ADC_IFR.JQRIF	ADC_CR0.IE
ADC SQR 扫描转换完成	ADC_IFR.SQRIF	
ADC 转换结果位于区间值区域（大于等于下阈值小于上阈值）	ADC_IFR.REGIF	
ADC 转换结果位于上阈值区域(大于等于上阈值)	ADC_IFR.HTIF	
ADC 转换结果比较下阈值区域（小于下阈值）	ADC_IFR.LTIF	
ADC 单次转换完成	ADC_IFR.SGLIF	

22.3.9 使用温度传感器测量环境温度

温度传感器的输出电压会随环境温度的改变而变化，故根据温度传感器的输出电压即可计算出相应环境温度。当 ADC 模块的测量通道选择温度传感器的输出电压时，即可测量环境温度。

环境温度 T_A 的计算公式如下：

$$T_A(^{\circ}\text{C}) = \begin{cases} \frac{1}{16} \cdot T_0 + K \cdot V_{\text{ref}} \cdot (C_{1_1\text{P5ref}} - C_{0_1\text{P5ref}}), \dots\dots\dots (V_{\text{ref}} = 1.5\text{V}) \\ \frac{1}{16} \cdot T_0 + K \cdot V_{\text{ref}} \cdot (C_{1_2\text{P5ref}} - C_{0_2\text{P5ref}}), \dots\dots\dots (V_{\text{ref}} = 2.5\text{V}) \end{cases}$$

其中：

- V_{ref} 为当前 ADC 模块的参考电压，典型值为 1.5V 或 2.5V。
- $C_{1_1\text{P5ref}}$ 和 $C_{1_2\text{P5ref}}$ 为 ADC 模块测量温度传感器输出电压的结果，取值为 0~4095。
- T_0 是在温度测试时记录的环境温度的表征值，其值为实际温度数值的绝对值的 16 倍。
- K 是温度传感器的输出曲线斜率，由于生产工艺的影响，不同芯片的温度曲线的斜率各不相同，典型值请参见《数据手册》。
- $C_{0_1\text{P5ref}}$ 和 $C_{0_2\text{P5ref}}$ 都是温度测试获得的校准值，在电源电压为 3.3V、环境温度为 T_0 的测试条件下 ADC 模块测量温度传感器输出电压的结果。

表 22-3 温度传感器校准值信息表

温度	校准值名称	校准值存放地址	位宽
常温	$C_{0_1\text{P5ref}}$	0x00101BE8~0x00101BE9	12 bits
	$C_{0_2\text{P5ref}}$	0x00101BEA~0x00101BEB	12 bits
	T_0	0x00101BEC~0x00101BED	16 bits

计算示例如下：

条件 1： $V_{\text{ref}}=2.5$ 、 $C_{0_2\text{P5ref}}=1389$ 、 $C_{1_2\text{P5ref}}=1180$ 、 $K=0.0798$ 、 $T_0=400$ ：

温度 1：

$$T_A(^{\circ}\text{C}) = \frac{1}{16} \cdot T_0 + K \cdot V_{\text{ref}} \cdot (C_{1_2\text{P5ref}} - C_{0_2\text{P5ref}}) = 25 + 0.0798 \times 2.5 \times (1180 - 1389) = -16.7^{\circ}\text{C}$$

条件 2： $V_{\text{ref}}=1.5$ 、 $C_{0_1\text{P5ref}}=2317$ 、 $C_{1_1\text{P5ref}}=2858$ 、 $K=0.0798$ 、 $T_0=400$ ：

温度 2：

$$T_A(^{\circ}\text{C}) = \frac{1}{16} \cdot T_0 + K \cdot V_{\text{ref}} \cdot (C_{1_1P5\text{ref}} - C_{0_1P5\text{ref}}) = 25 + 0.0798 \times 1.5 \times (2858 - 2317) = 89.76^{\circ}\text{C}$$

22.4 操作示例

22.4.1 通过 ADC_SGLSTART.START 位启动 ADC 单次转换操作流程

操作步骤

步骤 1. 配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

步骤 2. 设置 PBADS[1]为 1，将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

步骤 3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块。

步骤 4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块。

步骤 5. 延迟 T_{ADCSTART} 和 $T_{\text{stable(BGR)}}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。

步骤 6. 设置 ADC_CR1.MODE 为 0，选择单次转换模式。

步骤 7. 配置 ADC_CR0.REF，选择 ADC 的参考电压。

步骤 8. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。

步骤 9. 配置 ADC_CR0.SGLMUX，选择待转换的通道。

步骤 10. 设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

步骤 11. 设置 ADC_SGLSTART.START 为 1，启动 ADC 单次转换。如果还需要一直转换，可以将 ADC_ALLSTART.START 设置为 1。

步骤 12. 等待 ADC_IFR.SGLIF 变为 1，读取 ADC_RESULT 寄存器以获取 ADC 转换结果。

步骤 13. 如需对其它通道进行转换，重复执行步骤 9~12。如果已经设置了一直转换，在对其他通道进行转换前需要将 ADC_ALLSTART.START 设置为 0，停止一直转换后再执行步骤 9~12。

步骤 14. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

22.4.2 通过外部触发启动 ADC 单次转换操作流程

操作步骤

步骤 1. 配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

步骤 2. 设置 PBADS[1]为 1，将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

步骤 3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块。

步骤 4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块。

步骤 5. 延迟 T_{ADCSTART} 和 $T_{\text{stable(BGR)}}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。

步骤 6. 设置 ADC_CR1.MODE 为 0，选择单次转换模式。

步骤 7. 设置 ADC_CR0.IE 为 1，使能 ADC 中断。

步骤 8. 使能 NVIC 中断向量表中的 ADC 中断。

步骤 9. 配置 ADC_CR0.REF，选择 ADC 的参考电压。

步骤 10. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。

步骤 11. 配置 ADC_CR0.SGLMUX，选择待转换的通道。

步骤 12. 设置 ADC_ICR 为 0x0，清除 ADC 中断标志。

步骤 13. 配置 ADC_EXTTRIGGER0, 选择外部触发条件。

步骤 14. 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_RESULT 寄存器以获取 ADC 转换结果。

步骤 15. 如需对其它通道进行转换, 重复执行步骤 11~14。

步骤 16. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

22.4.3 通过 ADC_SQRSTART.START 位启动 ADC SQR 扫描转换操作流程

操作步骤

步骤 1. 配置 PAADS~PCADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

步骤 2. 设置 PBADS[1]为 1, 将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

步骤 3. 设置 ADC_BGR.BGR_EN 为 1, 使能 BGR 模块。

步骤 4. 设置 ADC_CR0.EN 为 1, 使能 ADC 模块。

步骤 5. 延迟 $T_{ADCSTART}$ 和 $T_{stable(BGR)}$ 两者中最大的时间 (具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述), 等待 ADC 及 BGR 模块启动完成。

步骤 6. 设置 ADC_CR1.MODE 为 1, 选择扫描转换模式。

步骤 7. 配置 ADC_CR0.REF, 选择 ADC 的参考电压。

步骤 8. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV, 设置 ADC 的转换速度。

步骤 9. 配置 ADC_SQR0.CHxMUX, 选择 SQR 扫描转换通道。

步骤 10. 配置 ADC_SQR1.CNT, 选择 SQR 扫描转换的总转换次数。

总转换次数需与步骤 9 中配置的转换通道数保持一致。

步骤 11. 设置 ADC_ICR.SQRIC 为 0, 清除 ADC_IFR.SQRIF 标志。

步骤 12. 设置 ADC_SQRSTART.START 为 1, 启动 ADC SQR 扫描转换。

步骤 13. 等待 ADC_IFR.SQRIF 变为 1, 读取 ADC_SQRRESULTx~ADC_SQRRESULT0 寄存器以获取相应通道的转换结果。

步骤 14. 如需对其它通道进行转换, 重复执行步骤 9~13。

步骤 15. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

22.4.4 通过外部触发启动 ADC SQR 扫描转换操作流程

操作步骤

步骤 1. 配置 PAADS~PCADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。

步骤 2. 设置 PBADS[1]为 1, 将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

步骤 3. 设置 ADC_BGR.BGR_EN 为 1, 使能 BGR 模块。

步骤 4. 设置 ADC_CR0.EN 为 1, 使能 ADC 模块。

步骤 5. 延迟 $T_{ADCSTART}$ 和 $T_{stable(BGR)}$ 两者中最大的时间 (具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述), 等待 ADC 及 BGR 模块启动完成。

步骤 6. 设置 ADC_CR1.MODE 为 1, 选择扫描转换模式。

步骤 7. 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。

步骤 8. 使能 NVIC 中断向量表中的 ADC 中断。

步骤 9. 配置 ADC_CR0.REF, 选择 ADC 的参考电压。

- 步骤 10. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV, 设置 ADC 的转换速度。
- 步骤 11. 配置 ADC_SQR0.CHxMUX, 选择 SQR 扫描转换通道。
- 步骤 12. 配置 ADC_SQR1.CNT, 选择 SQR 扫描转换的总转换次数。
总转换次数需与步骤 11 中配置的转换通道数保持一致。
- 步骤 13. 设置 ADC_ICR 为 0x0, 清除 ADC 中断标志。
- 步骤 14. 配置 ADC_EXTTRIGGER0, 选择外部触发条件。
- 步骤 15. 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_SQRRESULTx~ADC_SQRRESULT0 寄存器以获取相应通道的转换结果。
- 步骤 16. 如需对其它通道进行转换, 重复执行步骤 11~15。
- 步骤 17. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

22.4.5 通过 ADC_JQRSTART.START 位启动 ADC JQR 扫描转换操作流程

操作步骤

- 步骤 1. 配置 PAADS~PCADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。
- 步骤 2. 设置 PBADS[1]为 1, 将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

- 步骤 3. 设置 ADC_BGR.BGR_EN 为 1, 使能 BGR 模块。
- 步骤 4. 设置 ADC_CR0.EN 为 1, 使能 ADC 模块。
- 步骤 5. 延迟 $T_{ADCSTART}$ 和 $T_{stable(BGR)}$ 两者中最大的时间 (具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述), 等待 ADC 及 BGR 模块启动完成。
- 步骤 6. 设置 ADC_CR1.MODE 为 1, 选择扫描转换模式。
- 步骤 7. 配置 ADC_CR0.REF, 选择 ADC 的参考电压。
- 步骤 8. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV, 设置 ADC 的转换速度。
- 步骤 9. 配置 ADC_JQR.CHxMUX, 选择 JQR 扫描转换通道。
- 步骤 10. 配置 ADC_JQR.CNT, 选择 JQR 扫描转换的总转换次数。
总转换次数需与步骤 9 中配置的转换通道数保持一致。
- 步骤 11. 设置 ADC_ICR.JQRIC 为 0, 清除 ADC_IFR.JQRIF 标志。
- 步骤 12. 设置 ADC_JQRSTART.START 为 1, 启动 ADC JQR 扫描转换。
- 步骤 13. 等待 ADC_IFR.JQRIF 变为 1, 读取 ADC_JQRRESULTx~ADC_JQRRESULT0 寄存器以获取相应通道的转换结果。
- 步骤 14. 如需对其它通道进行转换, 重复执行步骤 9~13。
- 步骤 15. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

22.4.6 通过外部触发启动 ADC JQR 扫描转换操作流程

操作步骤

- 步骤 1. 配置 PAADS~PCADS 相应的位, 将待转换的 ADC 通道配置为模拟端口。
- 步骤 2. 设置 PBADS[1]为 1, 将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚, 则可以略过本步骤。

- 步骤 3. 设置 ADC_BGR.BGR_EN 为 1, 使能 BGR 模块。
- 步骤 4. 设置 ADC_CR0.EN 为 1, 使能 ADC 模块。

- 步骤 5. 延迟 T_{ADCSTART} 和 $T_{\text{stable(BGR)}}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。
- 步骤 6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式。
- 步骤 7. 设置 ADC_CR0.IE 为 1，使能 ADC 中断。
- 步骤 8. 使能 NVIC 中断向量表中的 ADC 中断。
- 步骤 9. 配置 ADC_CR0.REF，选择 ADC 的参考电压。
- 步骤 10. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。
- 步骤 11. 配置 ADC_JQR.CHxMUX，选择 JQR 扫描转换通道。
- 步骤 12. 配置 ADC_JQR.CNT，选择 JQR 扫描转换的总转换次数。
总转换次数需与步骤 11 中配置的转换通道数保持一致。
- 步骤 13. 设置 ADC_ICR 为 0x0，清除 ADC 中断标志。
- 步骤 14. 配置 ADC_EXTTRIGGER1，选择外部触发条件。
- 步骤 15. 当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_JQRRESULTx~ADC_JQRRESULT0 寄存器以获取相应通道的转换结果。
- 步骤 16. 如需对其它通道进行转换，重复执行步骤 11~15。
- 步骤 17. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

22.4.7 通过 ADC_SQRSTART.START 位启动 ADC 连续转换累加操作流程

操作步骤

- 步骤 1. 配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。
- 步骤 2. 设置 PBADS[1] 为 1，将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

- 步骤 3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块。
- 步骤 4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块。
- 步骤 5. 延迟 T_{ADCSTART} 和 $T_{\text{stable(BGR)}}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。
- 步骤 6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式。
- 步骤 7. 设置 ADC_CR1.RACCEN 为 1，使能 ADC 转换自动累加功能。
- 步骤 8. 配置 ADC_CR0.REF，选择 ADC 的参考电压。
- 步骤 9. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。
- 步骤 10. 配置 ADC_SQR0.CHxMUX，选择 SQR 扫描转换通道。
- 步骤 11. 配置 ADC_SQR1.CNT，选择 SQR 扫描转换的总转换次数。
总转换次数需与步骤 10 中配置的转换通道数保持一致。
- 步骤 12. 设置 ADC_ICR.SQRIC 为 0，清除 ADC_IFR.SQRIF 标志。
- 步骤 13. 设置 ADC_CR1.RACCCLR 为 0，清零 ADC_RESULTACC 寄存器。
- 步骤 14. 设置 ADC_SQRSTART.START 为 1，启动 ADC SQR 扫描转换。
- 步骤 15. 等待 ADC_IFR.SQRIF 变为 1，读取 ADC_RESULTACC 寄存器以获取转换结果累加值。
- 步骤 16. 如需对其它通道进行转换，重复执行步骤 10~15。
- 步骤 17. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

22.4.8 通过外部触发启动 ADC 连续转换累加操作流程

操作步骤

- 步骤 1. 配置 PAADS~PCADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

步骤 2. 设置 PBADS[1] 为 1，将 ADC 外部参考电压引脚配置为模拟端口。



说明

如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

步骤 3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块。

步骤 4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块。

步骤 5. 延迟 $T_{ADCSTART}$ 和 $T_{stable(BGR)}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。

步骤 6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式。

步骤 7. 设置 ADC_CR0.IE 为 1，使能 ADC 中断。

步骤 8. 使能 NVIC 中断向量表中的 ADC 中断。

步骤 9. 设置 ADC_CR1.RACCEN 为 1，使能 ADC 转换自动累加功能。

步骤 10. 配置 ADC_CR0.REF，选择 ADC 的参考电压。

步骤 11. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。

步骤 12. 配置 ADC_SQR0.CHxMUX，选择 SQR 扫描转换通道。

步骤 13. 配置 ADC_SQR1.CNT，选择 SQR 扫描转换的总转换次数。
总转换次数需与步骤 12 中配置的转换通道数保持一致。

步骤 14. 设置 ADC_ICR 为 0x0，清除 ADC 中断标志。

步骤 15. 设置 ADC_CR1.RACCCLR 为 0，清零 ADC_RESULTACC 寄存器。

步骤 16. 配置 ADC_EXTTRIGGER0，选择外部触发条件。

步骤 17. 当外部触发条件触发 ADC 完成转换时，ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_RESULTACC 寄存器以获取转换结果累加值。

步骤 18. 如需对其它通道进行转换，重复执行步骤 12~17。

步骤 19. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

22.4.9 通过 ADC 测量环境温度操作流程

操作步骤

步骤 1. 设置 ADC_BGR 为 3，使能 BGR 模块和温度传感器模块。

步骤 2. 设置 ADC_CR0.EN 为 1，使能 ADC 模块。

步骤 3. 延迟 $T_{ADCSTART}$ 和 $T_{stable(BGR)}$ 两者中最大的时间（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压/ADC 特性”章节描述），等待 ADC 及 BGR 模块启动完成。

步骤 4. 设置 ADC_CR1.MODE 为 0，选择单次转换模式。

步骤 5. 配置 ADC_CR0.REF，选择 ADC 的参考电压为内部 1.5V 或内部 2.5V。

步骤 6. 配置 ADC_CR0.SAM 及 ADC_CR0.CLKDIV，设置 ADC 的转换速度。

步骤 7. 设置 ADC_CR0.SGLMUX 为 0x10，选择待转换的通道为温度传感器的输出。

步骤 8. 设置 ADC_CR0.BUF 为 1，使能输入信号放大器。

步骤 9. 设置 ADC_ICR.SGLIC 为 0，清除 ADC_IFR.SGLIF 标志。

步骤 10. 设置 ADC_SGLSTART.START 为 1，启动 ADC 单次转换。

步骤 11. 等待 ADC_IFR.SGLIF 变为 1，读取 ADC_RESULT 寄存器以获取 ADC 转换结果。

步骤 12. 设置 ADC_CR0.EN 及 ADC_BGR 为 0，关闭 ADC 模块、BGR 模块、温度传感器模块。

步骤 13. 读取温度传感器校准值，根据公式计算当前的环境温度。

22.5 寄存器

22.5.1 寄存器总表

基地址：0x40002400

表 22-4 模数转换器 (ADC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	ADC_BGR	BGR 控制寄存器	32
0x004	ADC_CR0	ADC 基本配置寄存器 0	32
0x008	ADC_CR1	ADC 基本配置寄存器 1	32
0x040	ADC_SQR0	ADC SQR 扫描转换通道配置寄存器 0	32
0x048	ADC_SQR1	ADC SQR 扫描转换通道配置寄存器 1	32
0x04C	ADC_JQR	ADC JQR 扫描转换通道配置寄存器	32
0x050	ADC_SQRRESULT0	ADC SQR 扫描转换通道 0 转换结果寄存器	32
0x054	ADC_SQRRESULT1	ADC SQR 扫描转换通道 1 转换结果寄存器	32
0x058	ADC_SQRRESULT2	ADC SQR 扫描转换通道 2 转换结果寄存器	32
0x05C	ADC_SQRRESULT3	ADC SQR 扫描转换通道 3 转换结果寄存器	32
0x060	ADC_SQRRESULT4	ADC SQR 扫描转换通道 4 转换结果寄存器	32
0x064	ADC_SQRRESULT5	ADC SQR 扫描转换通道 5 转换结果寄存器	32
0x090	ADC_JQRRESULT0	ADC JQR 扫描转换通道 0 转换结果寄存器	32
0x094	ADC_JQRRESULT1	ADC JQR 扫描转换通道 1 转换结果寄存器	32
0x098	ADC_JQRRESULT2	ADC JQR 扫描转换通道 2 转换结果寄存器	32
0x09C	ADC_JQRRESULT3	ADC JQR 扫描转换通道 3 转换结果寄存器	32
0x0A0	ADC_RESULT	ADC 转换结果寄存器	32
0x0A4	ADC_RESULTACC	ADC 转换结果累加值寄存器	32
0x0A8	ADC_HT	ADC 比较上阈值寄存器	32
0x0AC	ADC_LT	ADC 比较下阈值寄存器	32
0x0B0	ADC_IFR	ADC 中断标志寄存器	32
0x0B4	ADC_ICR	ADC 中断清除寄存器	32
0x0B8	ADC_EXTTRIGGER0	ADC 单次转换或 SQR 扫描转换外部中断触发源配置寄存器	32
0x0BC	ADC_EXTTRIGGER1	ADC JQR 扫描转换外部中断触发源配置寄存器	32
0x0C0	ADC_SGLSTART	ADC 单次转换启动控制寄存器	32

偏移地址	寄存器	描述	位宽
0x0C4	ADC_SQRSTART	ADC SQR 扫描转换启动控制寄存器	32
0x0C8	ADC_JQRSTART	ADC JQR 扫描转换启动控制寄存器	32
0x0CC	ADC_ALLSTART	ADC 一直转换启动停止控制寄存器	32

22.5.2 BGR 控制寄存器 (ADC_BGR)

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														TS_EN	BGR_EN
Access																															RW	RW

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	TS_EN	内置温度传感器使能控制 ● 0b0：禁止内部温度传感器 ● 0b1：使能内部温度传感器， 必须使能 BGR  说明 温度传感器使能 $T_{stable(TS)}$ 时间后（具体数值请参见《数据手册》“电气特性-工作条件-温度传感器特性”章节描述），才能输出稳定信号。	RW
0	BGR_EN	BGR 使能控制 ● 0b0：禁止 BGR ● 0b1：使能 BGR  说明 ● SYSCTRL 模块寄存器 PERI_CLKEN0.ADC 为 1 时，才可以操作此寄存器。 ● BGR 使能 $T_{stable(BGR)}$ 时间后（具体数值请参见《数据手册》“电气特性-工作条件-内置的参考电压”章节描述），才能输出稳定的高精度参考电压。BGR 稳定后才可以被其它模块使用，故用户操作中应加入等待 BGR 稳定的步骤。 ● 当使用 ADC 时，必须使能 BGR。 ● 当使用 VC 时，需根据 VC 寄存器的配置决定是否使能 BGR。 ● 芯片进入低功耗模式之前建议关闭 BGR。	RW

22.5.3 ADC 基本配置寄存器 0 (ADC_CR0)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000427F0																															
Name	Reserved												SAM		Reserved		Reserved		BUF		REF	SGLMUX						CLKDIV		Reserved		
Access													RW			RW		BUF		REF	RW						RW			RW		

位/位域	标记	功能描述	读写
31:21	Reserved	-	-
20:17	SAM	ADC 采样周期选择 ● 0b0000: 2 个 ADCCLK ● 0b0001: 3 个 ADCCLK ● 0b0010: 4 个 ADCCLK ● 0b0011: 6 个 ADCCLK ● 0b0100: 8 个 ADCCLK ● 0b0101: 10 个 ADCCLK ● 0b0110: 12 个 ADCCLK ● 0b0111: 16 个 ADCCLK ● 0b1000: 20 个 ADCCLK ● 0b1001: 28 个 ADCCLK ● 0b1010: 36 个 ADCCLK ● 0b1011: 48 个 ADCCLK ● 0b1100: 64 个 ADCCLK ● 0b1101: 72 个 ADCCLK ● 0b1110: 96 个 ADCCLK ● 0b1111: 127 个 ADCCLK  说明 仅 ADC 空闲状态下可修改。	RW
16	Reserved	-	-
15	IE	ADC 中断控制 ● 0b0: 禁止 ● 0b1: 使能	RW
14:12	Reserved	-	-
11	BUF	内置跟随器使能控制 ● 0b0: 关闭跟随器, 外部输入信号与 ADC 直接相连 ● 0b1: 打开跟随器, 外部输入信号通过跟随器后与 ADC 相连, 用于测量高阻信号  说明 使能该功能时, 最大速率为 200ksps。以下条件必须使能该功能: ● 待转换的信号输出阻抗较高 ● 待转换的信号为: AVCC/3、内置温度传感器输出电压、VREF1P2 该位仅在 CR0.EN 为 1 时生效。	RW

位/位域	标记	功能描述	读写
10:9	REF	ADC 参考电压选择 ● 0b00: 内部 1.5V ● 0b01: 内部 2.5V (电源电压需超过 2.8V) ● 0b10: 外部参考电压 EXVREF (PB01) ● 0b11: AVCC 电压	RW
8:4	SGLMUX	单次转换模式转换通道选择 ● 0b00000: AIN0 (PA00) ● 0b00001: AIN1 (PA01) ● 0b00010: AIN2 (PA02) ● 0b00011: AIN3 (PA03) ● 0b00100: AIN4 (PA04) ● 0b00101: AIN5 (PA05) ● 0b00110: AIN6 (PA06) ● 0b00111: AIN7 (PA07) ● 0b01000: AIN8 (PB00) ● 0b01001: AIN9 (PB01) ● 0b01010: AIN10 (PC00) ● 0b01011: AIN11 (PC01) ● 0b01100: AIN12 (PC02) ● 0b01101: AIN13 (PC03) ● 0b01110: AIN14 (PC04) ● 0b01111: AIN15 (PC05) ● 0b10000: 内置温度传感器输出电压 ● 0b10001: AVCC/3 ● 0b10010: VREF1P2 ● 其他: 禁止设置  说明 ● 设置为 0b10000~0b10010 时, ADCCLK 必须小于等于 2MHz, 采样周期必须大于等于 36 ADCCLK。 ● 仅 ADC 空闲状态下可修改。	RW
3:2	CLKDIV	ADC 时钟选择 ● 0b00: PCLK 时钟 ● 0b01: PCLK 时钟 2 分频 ● 0b10: PCLK 时钟 4 分频 ● 0b11: PCLK 时钟 8 分频  注意 ● 当 HCLK/PCLK 所选择的系统时钟分频大于 1 分频时, 必须将 CLKDIV 的分频设置为大于 1 分频。 ● 仅 ADC 空闲状态下可修改。	RW
1	Reserved	-	-

位/位域	标记	功能描述	读写
0	EN	ADC 使能控制 写 1 使能 ADC，写 0 关闭 ADC 读取： ● 0b0：ADC 处于关闭状态 ● 0b1：ADC 处于开启状态  说明 ● 建议在 ADC 处于空闲状态下关闭以避免非预期行为。 ● 当该位设置为 0 后，ADC 关闭过程会持续一段时间。如果在此期间重新将该位设置为 1，会使 ADC 继续工作，可能产生非预期行为。建议使能 ADC 之前先读取该位，确认为 0 后再置 1。	RW

22.5.4 ADC 基本配置寄存器 1 (ADC_CR1)

[illegible]

位/位域	标记	功能描述	读写
31:16	<i>Reserved</i>	-	-
15	RACCCLR	ADC 转换结果累加寄存器清零 写 0 将 ADC_RESULTACC 寄存器清零，写 1 无效；读出恒为 1	WO
14	REGCMP	ADC 区间比较控制 ● 0b0：禁止 ● 0b1：使能	RW
13	HTCMP	ADC 高阈值比较控制 ● 0b0：禁止 ● 0b1：使能	RW
12	LTCMP	ADC 低阈值比较控制 ● 0b0：禁止 ● 0b1：使能	RW
11	RACCEN	ADC 转换结果自动累加控制 ● 0b0：禁止 ● 0b1：使能	RW
10	MODE	ADC 转换模式选择 ● 0b0：单次转换模式 ● 0b1：扫描转换模式	RW
9	DMAJQR	JQR 扫描触发 DMA 读取控制 ● 0b0：禁止 ● 0b1：使能	RW
8	DMASQR	SQR 扫描触发 DMA 读取控制 ● 0b0：禁止 ● 0b1：使能	RW
7:3	THCH	阈值比较通道选择 设置请参见 ADC_CR0.SGLMUX	RW
2:0	<i>Reserved</i>	-	-

22.5.5 ADC SQR 扫描转换通道配置寄存器 0 (ADC_SQR0)

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	CH5MUX				CH4MUX				CH3MUX				CH2MUX				CH1MUX				CH0MUX										
Access		RW				RW				RW				RW				RW				RW										

位/位域	标记	功能描述	读写
31:30	Reserved	-	-
29:25	CH5MUX	SQR 扫描转换通道 5 选择 设置请参见 ADC_CR0.SGLMUX	RW
24:20	CH4MUX	SQR 扫描转换通道 4 选择 设置请参见 ADC_CR0.SGLMUX	RW
19:15	CH3MUX	SQR 扫描转换通道 3 选择 设置请参见 ADC_CR0.SGLMUX	RW
14:10	CH2MUX	SQR 扫描转换通道 2 选择 设置请参见 ADC_CR0.SGLMUX	RW
9:5	CH1MUX	SQR 扫描转换通道 1 选择 设置请参见 ADC_CR0.SGLMUX	RW
4:0	CH0MUX	SQR 扫描转换通道 0 选择 设置请参见 ADC_CR0.SGLMUX	RW

22.5.6 ADC SQR 扫描转换通道配置寄存器 1 (ADC_SQR1)

Offset	Bit Position																															
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														CNT	
Access																																

位/位域	标记	功能描述	读写
31:3	Reserved	-	-
2:0	CNT	SQR 扫描转换次数 实际转换次数为 CNT+1，最大有效配置为 0x5，即 6 次。设置大于 0x5 的值，实际进行 6 次转换。 依次转换 SQR 扫描的 CHxMUX~CH0MUX 选择的通道（x=CNT）。  说明 仅 ADC 空闲状态下可修改。	RW

22.5.7 ADC JQR 扫描转换通道配置寄存器 (ADC_JQR)

Offset	Bit Position																															
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved										CNT	CH3MUX				CH2MUX				CH1MUX				CH0MUX								
Access											RW	RW				RW				RW				RW								

位/位域	标记	功能描述	读写
31:22	Reserved	-	-
21:20	CNT	JQR 扫描转换次数 实际转换次数为 CNT+1 依次转换 JQR 扫描的 CHxMUX~CH0MUX 选择的通道 (x=CNT)  说明 仅 ADC 空闲状态下可修改。	RW
19:15	CH3MUX	JQR 扫描转换通道 3 选择 设置请参见 ADC_CR0.SGLMUX	RW
14:10	CH2MUX	JQR 扫描转换通道 2 选择 设置请参见 ADC_CR0.SGLMUX	RW
9:5	CH1MUX	JQR 扫描转换通道 1 选择 设置请参见 ADC_CR0.SGLMUX	RW
4:0	CH0MUX	JQR 扫描转换通道 0 选择 设置请参见 ADC_CR0.SGLMUX	RW

22.5.8 ADC SQR 扫描转换通道 x 转换结果寄存器 (ADC_SQRRESULTx) (x = 0~5)

偏移地址 (Offset) : 0x050 (SQRRESULT0) , 0x054 (SQRRESULT1) , 0x058 (SQRRESULT2) , 0x05C (SQRRESULT3) , 0x060 (SQRRESULT4) , 0x064 (SQRRESULT5)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RO RESULT															
Access																	RO															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	RESULT	ADC SQR 扫描转换通道 x 转换结果	RO

22.5.9 ADC JQR 扫描转换通道 x 转换结果寄存器 (ADC_JQRRESULTx) (x = 0~3)

偏移地址 (Offset) : 0x090 (JQRRESULT0) , 0x094 (JQRRESULT1) , 0x098 (JQRRESULT2) , 0x09C (JQRRESULT3)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RO RESULT															
Access																	RO															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	RESULT	ADC JQR 扫描转换通道 x 转换结果	RO

22.5.10 ADC 转换结果寄存器 (ADC_RESULT)

Offset	Bit Position																															
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																RO RESULT															
Access																	RO RESULT															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	RESULT	ADC 转换结果	RO

22.5.11 ADC 转换结果累加值寄存器 (ADC_RESULTACC)

Offset	Bit Position																															
0x0A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												RO RESULTACC																			
Access													RO RESULTACC																			

位/位域	标记	功能描述	读写
31:20	Reserved	-	-
19:0	RESULTACC	ADC 转换累加值	RO

22.5.12 ADC 比较上阈值寄存器 (ADC_HT)

Offset	Bit Position																															
0x0A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000FFF																															
Name	Reserved																															
Access																	RW HT															

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	HT	ADC 转换结果比较上阈值	RW

22.5.13 ADC 比较下阈值寄存器 (ADC_LT)

Offset	Bit Position																															
0x0AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																LT															
Access																																

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:0	LT	ADC 转换结果比较下阈值	RW

22.5.14 ADC 中断标志寄存器 (ADC_IFR)

Offset	Bit Position																																																													
0x0B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																														
Reset	0x00000000																																																													
Name	Reserved																										JQIF						SQIF						REGIF						HTIF						LTIF						SGLIF					
Access																																																					RO						RO			

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5	JQIF	ADC JQR 扫描转换完成标志 ● 0b0: 转换未完成 ● 0b1: 转换完成	RO
4	SQIF	ADC SQR 扫描转换完成标志 ● 0b0: 转换未完成 ● 0b1: 转换完成	RO
3	REGIF	ADC 转换结果比较区间标志 ● 0b0: ADC 转换结果位于[ADC_LT, ADC_HT)区间外 ● 0b1: ADC 转换结果位于[ADC_LT, ADC_HT)区间内	RO
2	HTIF	ADC 转换结果比较上阈值标志 ● 0b0: ADC 转换结果位于[ADC_HT, 4095]区间外 ● 0b1: ADC 转换结果位于[ADC_HT, 4095]区间内	RO
1	LTIF	ADC 转换结果比较下阈值标志 ● 0b0: ADC 转换结果位于[0, ADC_LT)区间外 ● 0b1: ADC 转换结果位于[0, ADC_LT)区间内	RO
0	SGLIF	ADC 单次转换完成标志 ● 0b0: 转换未完成 ● 0b1: 转换完成	RO

22.5.15 ADC 中断清除寄存器 (ADC_ICR)

Offset	Bit Position																																					
0x0B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x0000003F																																					
Name	Reserved																										WO	JQRIC	WO	SQRIC	WO	REGIC	WO	HTIC	WO	LTIC	WO	SGLIC
Access																											WO	JQRIC	WO	SQRIC	WO	REGIC	WO	HTIC	WO	LTIC	WO	SGLIC

位/位域	标记	功能描述	读写
31:6	Reserved	-	-
5	JQRIC	ADC JQR 扫描转换完成标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
4	SQRIC	ADC SQR 扫描转换完成标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
3	REGIC	ADC 转换结果比较区间标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
2	HTIC	ADC 转换结果比较上阈值标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
1	LTIC	ADC 转换结果比较下阈值标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO
0	SGLIC	ADC 单次转换完成标志清除 写 0 清除标志，写 1 无效；读出恒为 1	WO

22.5.16 ADC 单次转换或 SQR 扫描转换外部中断触发源配置寄存器 (ADC_EXTTRIGGER0)

Offset	Bit Position																															
0x0B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												EXTTRIGEN			
Access																													RW			

位/位域	标记	功能描述	读写
31:4	Reserved	-	-
3:0	EXTTRIGEN	ADC 触发信号选择 0b0000: 无外设触发 0b0001: TIM0_TRADC 上升沿 0b0010: TIM1_TRADC 上升沿 0b0011: TIM2_TRADC 上升沿 0b0100: TIM3_TRADC 上升沿 0b0101: LPTIM0_OV 上升沿 0b0110: USART0 中断请求 0b0111: LPUART0 中断请求 0b1000: VC0 中断请求 0b1001: RTC 中断请求 0b1010: SPI0 中断请求 0b1011: PA03 中断请求 0b1100: PB03 中断请求 0b1101: PC03 中断请求 0b1110: PD03 中断请求 0b1111: PA11 中断请求	RW

说明

- TIM0/1/2/3 触发 ADC 自动转换不需要使能相应中断，配置 TIM 的相应的触发使能寄存器即可，具体操作请参见《通用定时器 (TIM0/1/2/3) 》的“[功能说明-Timer 触发 ADC](#)”章节。
- 除 TIM0/1/2/3、LPTIM0 外，其他外设触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。
- 为保证 LPTIM0_OV 可以正常触发 ADC，LPTIM 计数时钟频率需低于 PCLK。

22.5.17 ADC JQR 扫描转换外部中断触发源配置寄存器 (ADC_EXTTRIGGER1)

Offset	Bit Position																															
0x0BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																												EXTTRIGEN			
Access																													RW			

位/位域	标记	功能描述	读写
31:4	Reserved	-	-
3:0	EXTTRIGEN	ADC 触发信号选择 0b0000: 无外设触发 0b0001: TIM0_TRGO 上升沿 0b0010: TIM1_TRGO 上升沿 0b0011: TIM2_TRGO 上升沿 0b0100: TIM3_CMP3A 上升沿 0b0101: LPTIM1_OV 上升沿 0b0110: USART1 中断请求 0b0111: LPUART1 中断请求 0b1000: VC1 中断请求 0b1001: DMA 中断请求 0b1010: SPI1 中断请求 0b1011: PA07 中断请求 0b1100: PB07 中断请求 0b1101: PC07 中断请求 0b1110: PD07 中断请求 0b1111: PA15 中断请求	RW

 **说明**

- TIM0/1/2/3 触发 ADC 自动转换不需要使能相应中断，配置 TIM 的相应的触发使能寄存器即可，具体操作请参见《通用定时器 (TIM0/1/2/3) 》的“[功能说明-Timer 触发 ADC](#)”章节。
- 除 TIM0/1/2/3、LPTIM1 外，其他外设触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发，需要清除中断标志。如果不需要进入中断服务程序，请不要使能 NVIC 的中断使能。
- 为保证 LPTIM1_OV 可以正常触发 ADC，LPTIM 计数时钟频率需低于 PCLK。

22.5.18 ADC 单次转换启动控制寄存器 (ADC_SGLSTART)

Offset	Bit Position																															
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															START
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	START	ADC 单次转换启动控制 写 1 启动 ADC 单次转换，禁止写 0；单次转换完成后硬件自动清 0 读取： 0b0：ADC 处于空闲状态 0b1：ADC 单次转换中  说明 - START 为 1 时，写 1 无效。 - 该位仅在 CR1.MODE 为 0 时有效。	RW

22.5.19 ADC SQR 扫描转换启动控制寄存器 (ADC_SQRSTART)

Offset	Bit Position																															
0x0C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															START
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	START	ADC SQR 扫描转换启动控制 写 1 启动 ADC SQR 扫描转换，禁止写 0；SQR 扫描完成后硬件自动清 0 读取： 0b0：ADC 处于空闲状态 0b1：SQR 转换中  说明 - START 为 1 时，写 1 无效。 - ADC 空闲状态下，写 1 有效。 - 该位仅在 CR1.MODE 为 1 时有效。	RW

22.5.20 ADC JQR 扫描转换启动控制寄存器 (ADC_JQRSTART)

Offset	Bit Position																															
0x0C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															START
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	START	ADC JQR 扫描转换启动控制 写 1 启动 ADC JQR 扫描转换，禁止写 0；JQR 扫描完成后硬件自动清 0 读取： 0b0：ADC 处于空闲状态 0b1：JQR 转换中  说明 - START 为 1 时，写 1 无效。 - ADC 空闲状态下，写 1 有效。 - 该位仅在 CR1.MODE 为 1 时有效。	RW

22.5.21 ADC 一直转换启动停止控制寄存器 (ADC_ALLSTART)

Offset	Bit Position																															
0x0CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															START
Access																																RW

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	START	ADC 单次转换一直转换启动控制 写 1 启动 ADC 一直转换，写 0 停止 ADC 转换 读取： ● 0b0：ADC 处于空闲状态 ● 0b1：ADC 连续转换中  说明 该位仅在 CR1.MODE 为 0 时有效。	RW

23 液晶控制器 (LCD)

23.1 简介

LCD 控制器是一款适用于单色无源液晶显示器 (LCD) 的数字控制器/驱动器，最多具有 8 个公用端子 (COM)，用以驱动 160 (4x40)、228 (6x38) 或 288 (8x36) 个 LCD 图像元素。端子的确切数量取决于数据手册中所述的器件引脚。

LCD 由若干区段 (像素或完整符号) 组成，这些区段均可点亮或熄灭。每个区段都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，相应的区段可见。区段电压必须为交流，以避免液晶中出现电泳效应 (这将影响显示效果)。之后，必须在区段两端生成波形以避免出现直流。

词汇表

- 液晶 (LCD)：无源显示面板，带有直接引向区段的端子。
- 公用 (COM)：连接到多个区段的电气连接端子。
- 偏置 (BIAS)：驱动 LCD 时使用的电压等级，定义为 $1/(\text{驱动 LCD 显示的电压等级数}-1)$ 。
- 区段 (SEG)：最小可视单元 (LCD 显示器上的最小组成元素，线条或点)。
- 占空比 (DUTY)：定义为 $1/(\text{LCD 显示器上的公用端子数})$ 的数字。
- 帧：写入区段的波形的一个周期。
- 帧速率：每秒帧数，即每秒激励 LCD 区段的次数。

23.2 主要特性

- 高度灵活的帧速率控制
- 支持静态、1/2、1/3、1/4、1/6 和 1/8 占空比
- 支持 1/2、1/3、1/4 偏置
- 多达 16 个寄存器的 LCD 数据 RAM
- 可通过软件配置 LCD 的对比度
- 3 种驱动波形生成方式
 - ▶ 内部电阻分压、外部电阻分压，外部电容分压方式
 - ▶ 可通过软件配置内部电阻分压方式的功耗，从而匹配 LCD 面板所需的电容电荷
- 支持低功耗模式：LCD 控制器可在 Active、Sleep、DeepSleep 模式下进行显示
- 可配置帧中断
- 支持 LCD 闪烁功能且可配置多种闪烁频率
- 未使用的 LCD 区段和公共引脚可配置为数字或模拟功能
- 支持死区插入

23.3 功能说明

23.3.1 功能框图

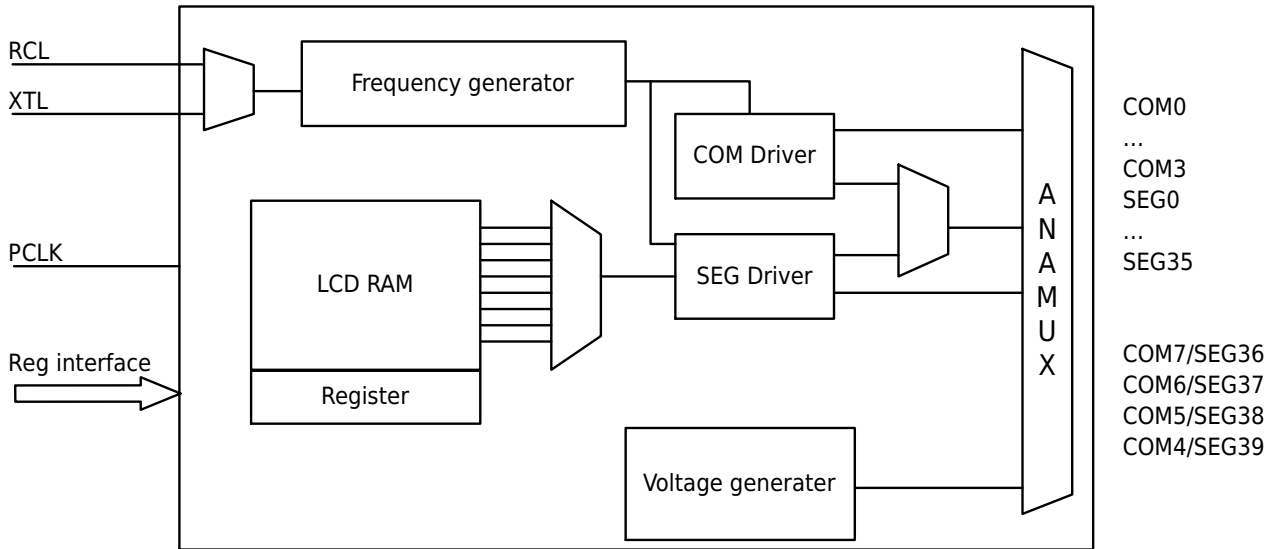


图 23-1 LCD 功能框图

- LCD RAM 和 Register 是 LCD 的数据和控制寄存器，用以存放 SEG 数据和控制 LCD，可通过 APB 总线配置，并使 CPU 产生中断。
- Frequency generator 是 LCD 内部频率生成模块，用于生成时钟、驱动 COM 和 SEG 输出，可以从 RCL 和 XTL 两个时钟源中选取源时钟。
- Voltage generator 是模拟电压产生模块，用于驱动 COM 和 SEG 上的电压，并通过模拟矩阵 ANAMUX，实现 LCD 显示。

23.3.2 帧频率

LCD 控制器可使用从集成时钟分频器输出的 LCD 时钟信号来产生 SEG 和 COM 线的时序。LCD 时钟频率可由 LCD_CR1.PSC 和 LCD_CR1.DIV 位配置，计算公式如下：

$$f_{LCD} = f_{in_clk} / (2^{(PSC+1)} \times (DIV+16))$$

其中 f_{LCD} 为 LCD 控制器的时钟基准，相当于相时钟。 f_{in_clk} 为时钟生成器输入时钟，可通过 LCD_CR1.CLKSRC 设置从 RCL 和 XTL 中选择。

LCD 的时钟频率相当于相频率。一个 TYPE B 帧是一个奇数帧或一个偶数帧，它们拥有有效的 COM 一样多的相。占空比定义为 $1 / (\text{SLCD 屏显示需要的 COM 数})$ ， f_{frame} 帧频率计算公式如下：

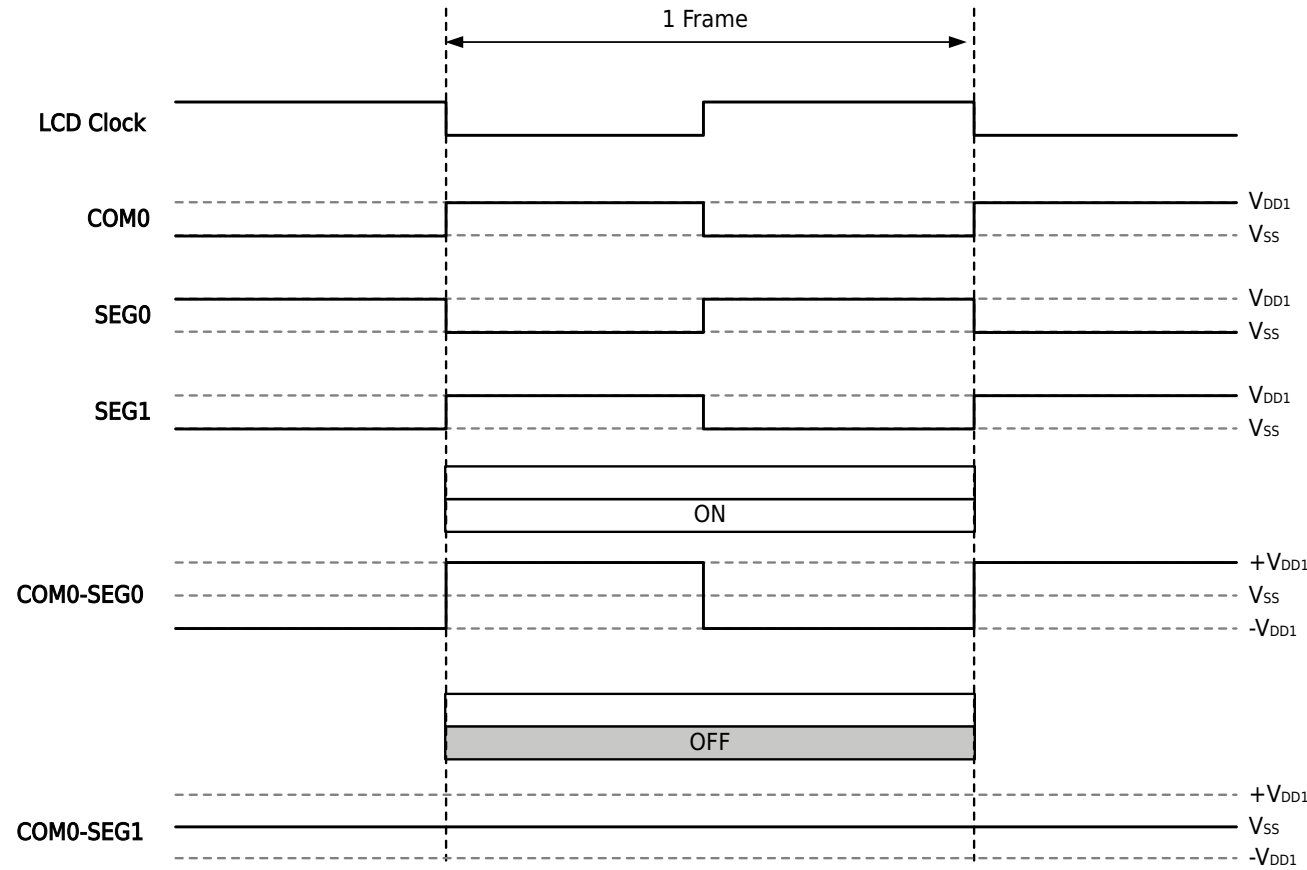
$$f_{frame} = f_{LCD} \times \text{Duty}$$

23.3.3 LCD 驱动波形

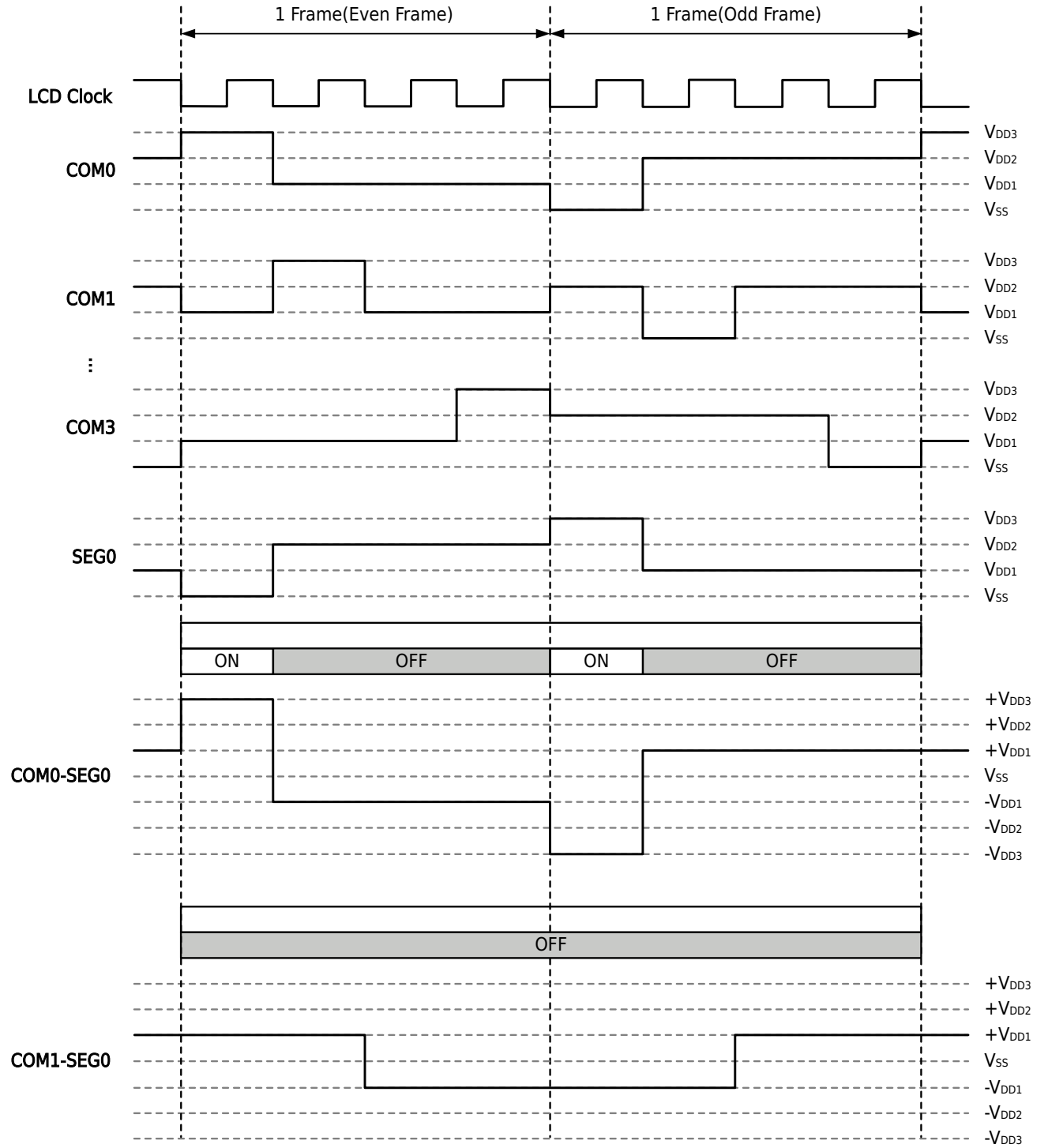
LCD 支持 6 种占空比 (Duty) 的驱动波形：静态、1/2、1/3、1/4、1/6 和 1/8，由 LCD_CR0.DUTY 进行设置。LCD 支持 3 种偏置 (Bias) 的驱动波形：1/2、1/3 和 1/4，由 LCD_CR0.BIAS 进行设置。

各模式下的驱动波形如下方所示，当 COM 数大于等于 4 的时候，会省略部分 COM。COM 按顺序依次有效，一个 COM 有效时，SEG 输出合适的电平，与 COM 电平共同施加在 LCD 面板上，当电压差大于 LCD 导通电压时，LCD 面板对应像素点被点亮（如图中 ON 状态）；反之，LCD 面板对应像素点不会被点亮（如图中 OFF 状态）。

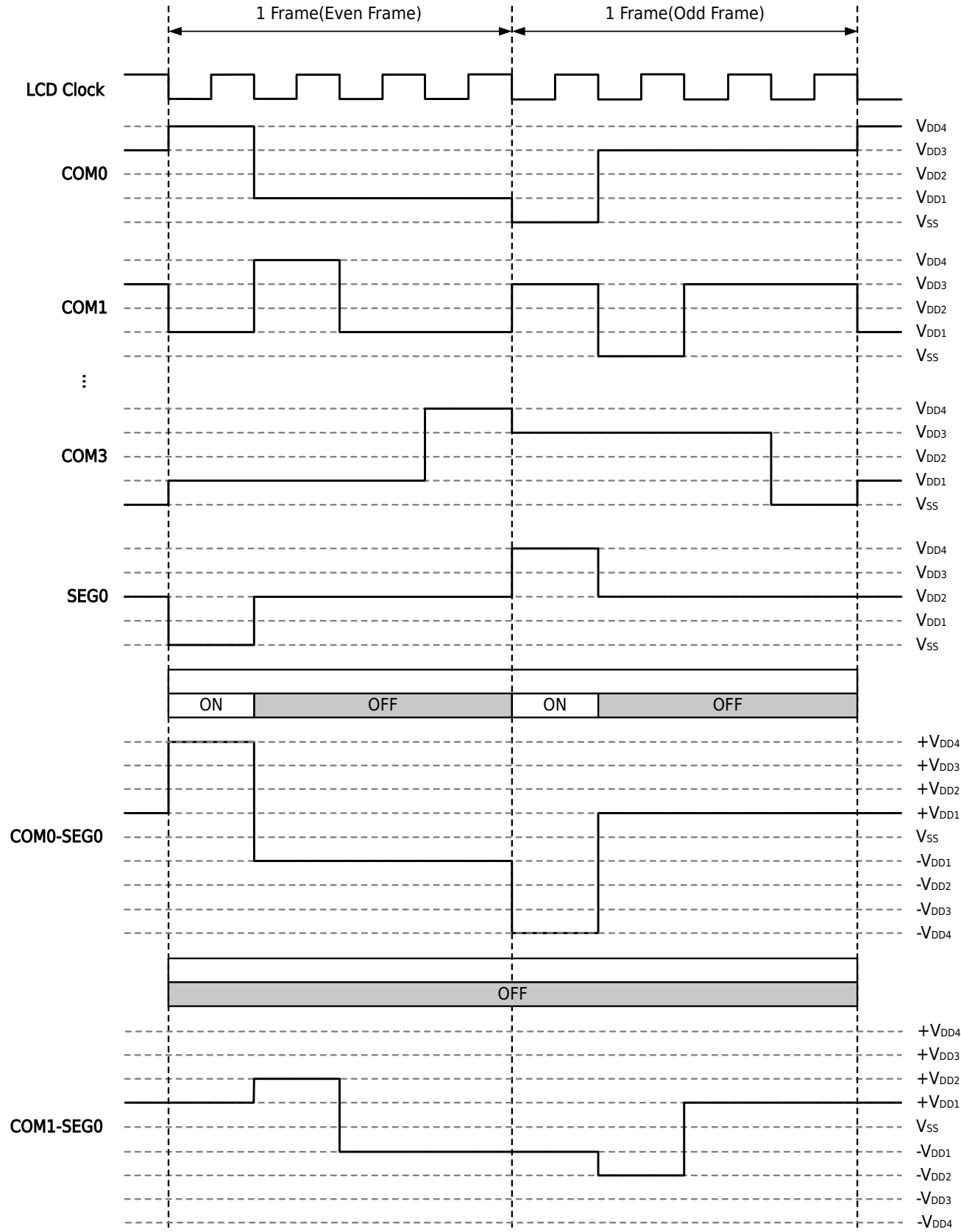
静态驱动波形



1/4Duty 1/3Bias Type B 驱动波形

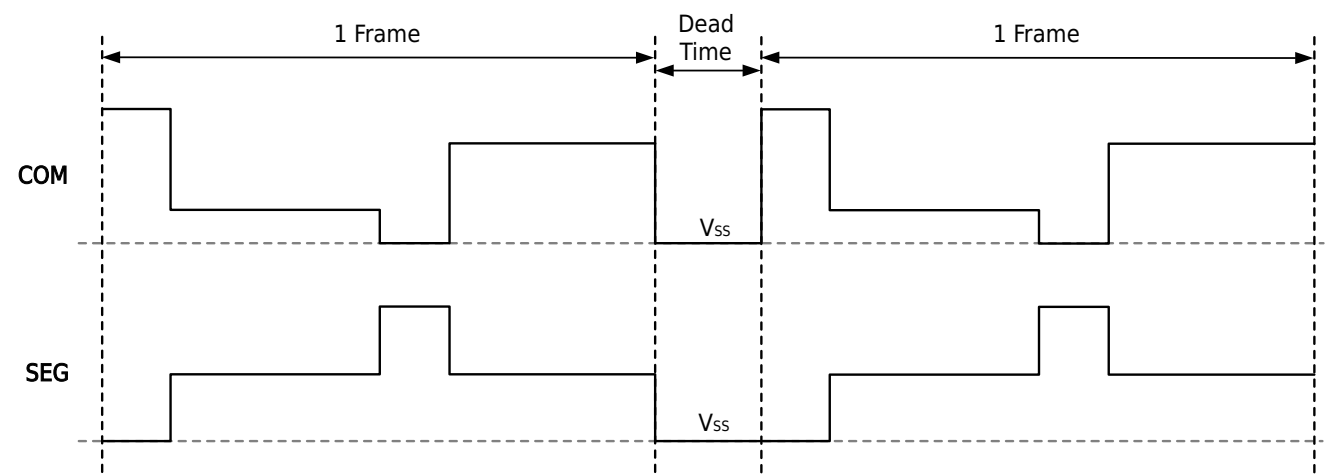


1/4Duty 1/4Bias Type B 驱动波形



23.3.4 死区插入

在帧与帧之间，可以插入死区时间来调节对比度。死区时间内，所有的 COM 和 SEG 都接到 VSS，死区时间长度由 LCD_CR1.DEAD 调节。如果是 TYPE B，则只有在偶数帧后，才能插入死区时间。死区时间插入波形如下：



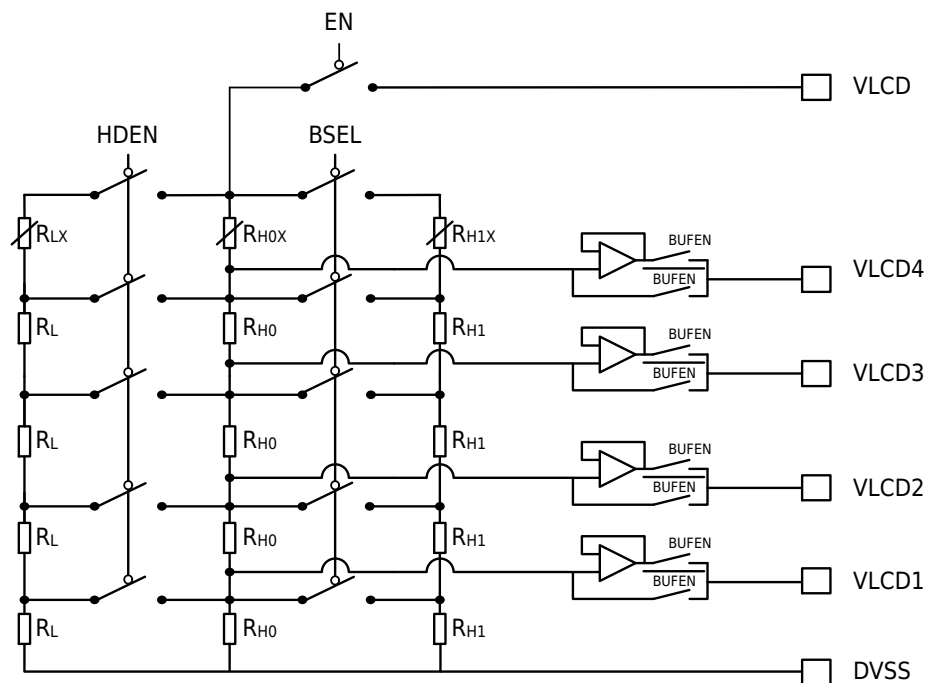
23.3.5 LCD Bias 产生电压

LCD 的 Bias 电压具有 3 种来源：内部电阻分压、外部电阻分压、外部电容分压。

1. 当选择内部电阻分压时，芯片会自动切换到内部的电路，这种模式驱动能力较弱。如果需要驱动的 LCD 屏幕较大，建议选择外部电阻分压或外部电容分压模式。也可以通过配置寄存器 LCD_CR1.DRVRES 来增强驱动能力。大功耗、中功耗、小功耗的电阻值逐渐减少；相同功耗配置下，1/2bias、1/3bias、1/4bias 的电阻值逐渐增大。
2. 当选择外部电阻分压时，根据系统功耗和 LCD 驱动能力来调整阻值，推荐使用 500kΩ 电阻。当选择外部电容分压时，驱动能力较强，推荐使用 100nF 电容。但这两种电路需要用户在外围引脚搭建相关电路，并占用 5 个 IO 口。

内部电阻模式

下图为 1/4 Bias 内部电阻示意图：



内部电阻模式 VLCDH, VLCD1~VLCD4 可以作为 LCD SEG 输出或者 IO 端口使用。

在转换期间, 为了快速到达静态状态, 低值电阻 (R_L) 被使用以增加电流。之后低值电阻被关闭, 高值电阻 (R_H) 被使用以减小功耗。 R_L 被使用的时间长度依据 LCD_CR1.DRVRES。通过配置 LCD_CR1.DRVRES $\geq 0xE$, R_L 可以一直被使用。

两组 R_H (R_{H0} 和 R_{H1}) 用于选择不同的功耗模式, 通过配置 LCD_CR0.BSEL 来实现大、中和小功耗的选择。

增强模式:

LCD 模块集成了可配置的电压输出驱动器, 可通过配置 LCD_CR0.VODEN 进入增强模式。启用电压输出驱动器, 可以减少由电桥上的 LCD 电容负载引起的电压干扰, 从而获得稳定的电压, 达到增强 LCD 驱动能力的目的。

在增强模式下, 高值电阻器桥 (RHN) 将产生一个中间电压, LCD_CR1.DRVRES 配置将被忽略, 低值电阻器桥 (RLN) 将被自动禁用, 从而降低了功耗。



说明

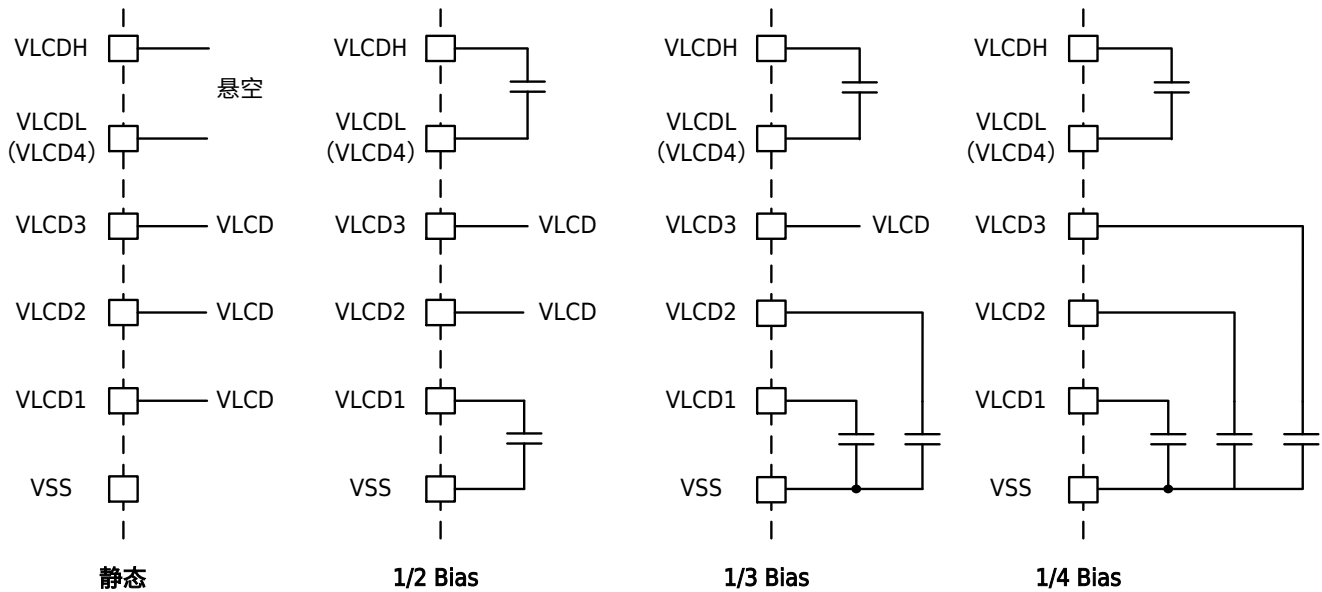
仅当未激活 LCD 控制器时才能配置电压输出驱动器。

建议在保证 LCD 的最高电压不低于 1.8V 时才打开 buffer, 否则可能会影响 LCD 的显示效果。

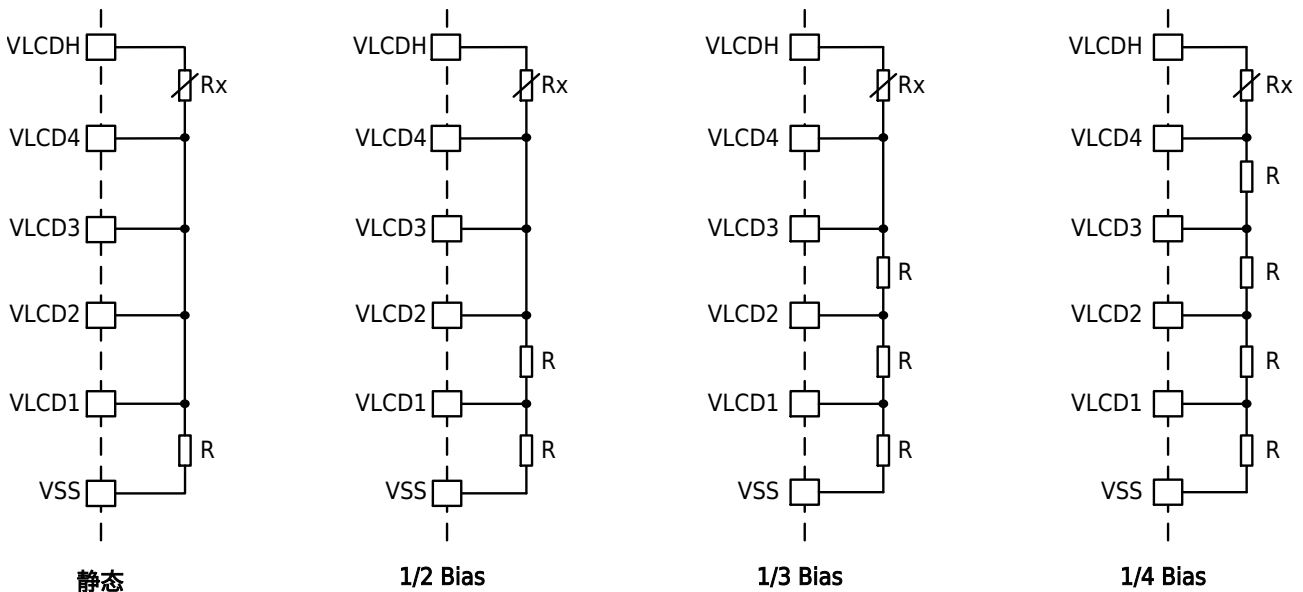
内部电阻模式, LCD 的驱动电压由 CR0.CONTRAST 控制, 如下表所示:

CR0.CONTRAST	VLCD4/VLCD
0	1.000
1	0.947
2	0.900
3	0.857
4	0.818
5	0.782
6	0.750
7	0.720
8	0.692
9	0.666
10	0.643
11	0.621
12	0.600
13	0.581
14	0.562
15	0.545

外部电容模式



外部电阻模式



说明

- Rx 为可调电阻，用于调节 LCD 显示对比度。
- 根据使用 LCD 屏幕选择合适的电阻 R。

23.3.6 DMA

LCD 支持软件和硬件触发 DMA 数据传输，可以将需要显示的内容从 RAM 或 ROM 中自动搬到 LCD 显示 RAM 中。硬件触发使用的是帧中断信号。

23.3.7 中断

当 LCD 设置有效时，LCD 中断可以配置为帧数产生中断。



说明

LCD 配置为静态模式 (LCD_CR0.DUTY=0b000) 时，不支持中断功能。

23.3.8 LCD 显示模式

LCD 支持两种显示模式。一种以 COM 为显示单元，同一个 SEG 的所有 COM 段在同一字节中（模式 0）。另外一种为同一个 COM 的不同 SEG 在同一个字节中（模式 1）。根据 LCD 面板选择合适的显示方式可以简化程序操作。



说明

其中 SEG0~SEG31 支持两种显示模式，SEG32~SEG39 只支持显示模式 1。如果使用 SEG32~SEG39，请按照显示模式 1 去配置 SEG32~SEG39 的 RAM 数据。

LCD 显示模式 1 (MODE=1)

1/8 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																LCDRAM0	
COM1																																LCDRAM1	
COM2																																LCDRAM2	
COM3																																LCDRAM3	
COM4																																LCDRAM4	
COM5																																LCDRAM5	
COM6																																LCDRAM6	
COM7																																LCDRAM7	
	SEG32	SEG33	SEG34	SEG35																													
COM0																																LCDRAM8	
COM1																																LCDRAM9	
COM2																																LCDRAMA	
COM3																																LCDRAMB	
COM4																																LCDRAMC	
COM5																																LCDRAMD	
COM6																																LCDRAME	
COM7																																LCDRAMF	

1/6 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
COM4																																	LCDRAM4
COM5																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37																											
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
COM3																																	LCDRAMB
COM4																																	LCDRAMC
COM5																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

1/4 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
COM3																																	LCDRAM3
																																	LCDRAM4
																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39																									
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
COM3																																	LCDRAMB
																																	LCDRAMC
																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

1/3 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
COM2																																	LCDRAM2
																																	LCDRAM3
																																	LCDRAM4
																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39																									
COM0																																	LCDRAM8
COM1																																	LCDRAM9
COM2																																	LCDRAMA
																																	LCDRAMB
																																	LCDRAMC
																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

1/2 Duty

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	
COM0																																	LCDRAM0
COM1																																	LCDRAM1
																																	LCDRAM2
																																	LCDRAM3
																																	LCDRAM4
																																	LCDRAM5
																																	LCDRAM6
																																	LCDRAM7
	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39																									
COM0																																	LCDRAM8
COM1																																	LCDRAM9
																																	LCDRAMA
																																	LCDRAMB
																																	LCDRAMC
																																	LCDRAMD
																																	LCDRAME
																																	LCDRAMF

LCD 显示模式 0 (MODE=0)

1/8 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG27								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

1/6 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5			COM0	COM1	COM2	COM3	COM4	COM5			
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG27								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

1/4 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					COM0	COM1	COM2	COM3					
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG27								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

1/3 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2						COM0	COM1	COM2						
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG27								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

1/2 Duty

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25	Bit26	Bit27	Bit28	Bit29	Bit30	Bit31	
COM0	COM1							COM0	COM1							COM0	COM1							COM0	COM1							
SEG0								SEG1								SEG2								SEG3								LCDRAM0
SEG4								SEG5								SEG6								SEG7								LCDRAM1
SEG8								SEG9								SEG10								SEG11								LCDRAM2
SEG12								SEG13								SEG14								SEG15								LCDRAM3
SEG16								SEG17								SEG18								SEG19								LCDRAM4
SEG20								SEG21								SEG22								SEG23								LCDRAM5
SEG24								SEG25								SEG26								SEG27								LCDRAM6
SEG28								SEG29								SEG30								SEG31								LCDRAM7

23.4 操作示例

23.4.1 内部电阻模式操作流程

- 步骤 1. 开启 LCD 时钟和 GPIO 时钟。
- 步骤 2. 将 LCD 使用到的管脚配置为模拟功能。
- 步骤 3. 配置 LCD_CR1.CLKSRC 选择工作时钟源。
- 步骤 4. 配置 LCD_CR0.BIAS 选择偏置。
- 步骤 5. 配置 LCD_CR0.DUTY 选择占空比。
- 步骤 6. 配置 LCD_CR1.PSC 和 LCD_CR1.DIV 选择帧频率。
- 步骤 7. 配置 LCD_CR1.MODE 选择显示模式。
- 步骤 8. 配置 LCD_CR0.VODEN 选择是否使能电压输出驱动，如果使能，则不需要配置步骤 9。
- 步骤 9. 配置 LCD_CR1.DRVRES 选择驱动脉冲时间。
- 步骤 10. 配置 LCD_POEN0 和 LCD_POEN1，使能相应管脚的 SEG 和 COM 功能。
- 步骤 11. 配置 LCD_CR0.EN 为 1，使能 LCD。
- 步骤 12. 向 LCD RAM 写入需要显示的内容。



注意

当管脚不作为 LCD 的 SEG 或 COM 使用时，禁止使能管脚的 SEG 或 COM 功能。

23.4.2 外部电容模式操作流程

- 步骤 1. 根据不同预期配置在芯片的外围引脚搭建相关电路。
- 步骤 2. 开启 LCD 时钟和 GPIO 时钟。
- 步骤 3. 将 LCD 使用到的管脚配置为模拟功能。
- 步骤 4. 配置 LCD_CR1.CLKSRC 选择工作时钟源。
- 步骤 5. 配置 LCD_CR0.BIAS 选择偏置。
- 步骤 6. 配置 LCD_CR0.DUTY 选择占空比。
- 步骤 7. 配置 LCD_CR1.PSC 和 LCD_CR1.DIV 选择帧频率。
- 步骤 8. 配置 LCD_CR0.CPCLK 选择电容分压模式驱动频率（此模式需要一定时间才能完成电容分压，显示开始时，可能会有短暂显示效果不佳的情况）。
- 步骤 9. 配置 LCD_POEN0 和 LCD_POEN1，使能相应管脚的 SEG 和 COM 功能。
- 步骤 10. 配置 LCD_CR0.EN 为 1，使能 LCD。
- 步骤 11. 向 LCD RAM 写入需要显示的内容。



注意

当管脚不作为 LCD 的 SEG 或 COM 使用时，禁止使能管脚的 SEG 或 COM 功能。

23.4.3 外部电阻模式操作流程

- 步骤 1. 根据不同预期配置在芯片的外围引脚搭建相关电路。
- 步骤 2. 开启 LCD 时钟和 GPIO 时钟。
- 步骤 3. 将 LCD 使用到的管脚配置为模拟功能。
- 步骤 4. 配置 LCD_CR1.CLKSRC 选择工作时钟源。
- 步骤 5. 配置 LCD_CR0.BIAS 选择偏置。
- 步骤 6. 配置 LCD_CR0.DUTY 选择占空比。
- 步骤 7. 配置 LCD_CR1.PSC 和 LCD_CR1.DIV 选择帧频率。

步骤 8. 配置 LCD_POEN0 和 LCD_POEN1，使能相应管脚的 SEG 和 COM 功能。

步骤 9. 配置 LCD_CR0.EN 为 1，使能 LCD。

步骤 10. 向 LCD RAM 写入需要显示的内容。

**注意**

当管脚不作为 LCD 的 SEG 或 COM 使用时，禁止使能管脚的 SEG 或 COM 功能。

23.5 寄存器

23.5.1 寄存器总表

基地址：0x40009C00

表 23-12 液晶控制器 (LCD) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	LCD_CR0	LCD 配置寄存器 0	32
0x004	LCD_CR1	LCD 配置寄存器 1	32
0x008	LCD_SR	LCD 状态寄存器	32
0x00C	LCD_INTCLR	LCD 中断清除寄存器	32
0x010	LCD_POEN0	LCD 输出配置寄存器 0	32
0x014	LCD_POEN1	LCD 输出配置寄存器 1	32
0x040	LCD_RAM0	LCD RAM0	32
0x044	LCD_RAM1	LCD RAM1	32
0x048	LCD_RAM2	LCD RAM2	32
0x04C	LCD_RAM3	LCD RAM3	32
0x050	LCD_RAM4	LCD RAM4	32
0x054	LCD_RAM5	LCD RAM5	32
0x058	LCD_RAM6	LCD RAM6	32
0x05C	LCD_RAM7	LCD RAM7	32
0x060	LCD_RAM8	LCD RAM8	32
0x064	LCD_RAM9	LCD RAM9	32
0x068	LCD_RAMA	LCD RAM10	32
0x06C	LCD_RAMB	LCD RAM11	32
0x070	LCD_RAMC	LCD RAM12	32
0x074	LCD_RAMD	LCD RAM13	32
0x078	LCD_RAME	LCD RAM14	32
0x07C	LCD_RAMF	LCD RAM15	32

23.5.2 配置寄存器 0 (LCD_CR0)

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000000DC																															
Name	Reserved																VODEN	CONTRAST				BSEL		DUTY		TYPE		CPCLK		BIAS		EN
Access																	RW	RW				RW		RW		RW		RW		RW		RW

位/位域	标记	功能描述	读写
31:17	Reserved	-	-
16	VODEN	电压输出驱动使能 0b0: 禁止 0b1: 使能  说明 该位只能在 LCD_CR0.EN 为 0 时设置。 仅当 BIAS 电压来源选择内部电阻分压时有效。	RW
15:12	CONTRAST	LCD 对比度调整 CONTRAST 值越大, LCD 波形的幅度越小 0b0000: LCD 波形幅度最大, 对比度最大 0b1111: LCD 波形幅度最小, 对比度最小  说明 仅当 BIAS 电压来源选择内部电阻分压时有效。	RW
11:9	BSEL	BIAS 电压来源选择 0b000: 外部电阻模式, 需要外部电路配合 0b001: 电容分压模式, 需要外部电路配合 0b010: 内部电阻分压, 中功耗模式 0b011: 禁止设置 0b100: 内部电阻分压, 低功耗模式 0b101: 禁止设置 0b110: 内部电阻分压, 大功耗模式 0b111: 禁止设置	RW
8:6	DUTY	LCD duty 配置 0b000: 静态 0b001: 1/2 duty 0b010: 1/3 duty 0b011: 1/4 duty 0b100: 禁止设置 0b101: 1/6 duty 0b110: 禁止设置 0b111: 1/8 duty	RW

位/位域	标记	功能描述	读写																				
8:6	DUTY	表 23-15 COMy/SEGx PAD 在不同 duty 下功能配置 <table border="1"> <thead> <tr> <th>PAD</th><th>1/4 duty</th><th>1/6 duty</th><th>1/8 duty</th></tr> </thead> <tbody> <tr> <td>COM4/SEG39</td><td>SEG39</td><td>COM4</td><td>COM4</td></tr> <tr> <td>COM5/SEG38</td><td>SEG38</td><td>COM5</td><td>COM5</td></tr> <tr> <td>COM6/SEG37</td><td>SEG37</td><td>SEG37</td><td>COM6</td></tr> <tr> <td>COM7/SEG36</td><td>SEG36</td><td>SEG36</td><td>COM7</td></tr> </tbody> </table>	PAD	1/4 duty	1/6 duty	1/8 duty	COM4/SEG39	SEG39	COM4	COM4	COM5/SEG38	SEG38	COM5	COM5	COM6/SEG37	SEG37	SEG37	COM6	COM7/SEG36	SEG36	SEG36	COM7	RW
PAD	1/4 duty	1/6 duty	1/8 duty																				
COM4/SEG39	SEG39	COM4	COM4																				
COM5/SEG38	SEG38	COM5	COM5																				
COM6/SEG37	SEG37	SEG37	COM6																				
COM7/SEG36	SEG36	SEG36	COM7																				
5	TYPE	LCD TYPE 配置 ● 0b0: 波形 B ● 0b1: 禁止设置	RW																				
4:3	CPCLK	电压泵时钟频率选择 ● 0b00: $f_{in_clk}/16$ ● 0b01: $f_{in_clk}/8$ ● 0b10: $f_{in_clk}/4$ ● 0b11: $f_{in_clk}/2$	RW																				
2:1	BIAS	LCD BIAS 配置 ● 0b00: 1/2 bias ● 0b01: 1/3 bias ● 0b10: 1/4 bias ● 0b11: 禁止设置	RW																				
0	EN	● 0b0: 禁止 ● 0b1: 使能	RW																				

23.5.3 配置寄存器 1 (LCD_CR1)

Offset	Bit Position																																				
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reset	0x00000000																																				
Name	Reserved								DEAD				PSC				DIV				DRVRES		Reserved	DMAEN	IE	MODE	CLKSRC	BLINKEN	BLINKCNT								
Access									RW				RW				RW				RW			RW	RW	RW	RW	RW		RW							

位/位域	标记	功能描述	读写
31:25	Reserved	-	-
24:22	DEAD	死区时间配置 0b000: 无死区时间 0b001: 1 相周期死区时间 0b010: 2 相周期死区时间 0b011: 3 相周期死区时间 0b100: 4 相周期死区时间 0b101: 5 相周期死区时间 0b110: 6 相周期死区时间 0b111: 7 相周期死区时间	RW
21:19	PSC	LCD 时钟预分频器 0b000: $f_{PSC} = f_{in_clk}/2$ 0b001: $f_{PSC} = f_{in_clk}/4$ 0b010: $f_{PSC} = f_{in_clk}/8$ 0b011: $f_{PSC} = f_{in_clk}/16$ 0b100: $f_{PSC} = f_{in_clk}/32$ - 其他: 禁止设置	RW
18:15	DIV	LCD 时钟分频器 0b0000: $f_{LCD} = f_{PSC}/16$ 0b0001: $f_{LCD} = f_{PSC}/17$ 0b1110: $f_{LCD} = f_{PSC}/30$ 0b1111: $f_{LCD} = f_{PSC}/31$	RW
14:12	DRVRES	内部电阻模式增强驱脉冲时间控制 0b000: OFF, 关闭增强驱动脉冲 0b001: $1/f_{in_clk}$ 0b010: $2/f_{in_clk}$ 0b011: $4/f_{in_clk}$ 0b100: $8/f_{in_clk}$ 0b101: $16/f_{in_clk}$ 0b110: $32/f_{in_clk}$ 0b111: ALL ON	RW

位/位域	标记	功能描述	读写
11	<i>Reserved</i>	-	-
10	DMAEN	DMA 硬件触发使能 ● 0b0: 禁止 ● 0b1: 使能	RW
9	IE	中断使能 ● 0b0: 禁止 ● 0b1: 使能	RW
8	MODE	LCD RAM 显示模式选择 ● 0b0: 模式 0 ● 0b1: 模式 1	RW
7	CLKSRC	LCD 时钟源选择 ● 0b0: RCL ● 0b1: XTL	RW
6	BLINKEN	LCD 闪屏配置 ● 0b0: 禁止 ● 0b1: 使能	RW
5:0	BLINKCNT	闪屏频率与 LCD 中断间隔设置 LCD 闪烁频率 = LCD 帧频率 / (BlinkCnt+1) LCD 中断间隔 = (1/LCD 帧频率) × (BlinkCnt+1)	RW

23.5.4 状态寄存器 (LCD_SR)

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															RO INTF
Access																																

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	INTF	中断标志 ● 0b0：未产生中断 ● 0b1：产生中断	RO

23.5.5 中断清除寄存器 (LCD_INTCLR)

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																															INTF
Access																																WO

位/位域	标记	功能描述	读写
31:1	Reserved	-	-
0	INTF	中断标志清除 写 1 清除中断标志，写 0 无效；读出恒为 0	WO

23.5.6 输出配置寄存器 0 (LCD_POEN0)

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:0	Sx (x=31~0)	SEGx 输出控制位 0b0: SEG 输出使能 0b1: SEG 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW

23.5.7 输出配置寄存器 1 (LCD_POEN1)

Offset	Bit Position																																					
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reset	0x01F000FF																																					
Name	Reserved							C3	C2	C1	C0	MUX	Reserved											S39C4	S38C5	S37C6	S36C7	S35	S34	S33	S32							
Access								RW	RW	RW	RW	RW												RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:25	Reserved	-	-
24:21	Cx (x=3~0)	COMx 输出控制位 0b0: COM 输出使能 0b1: COM 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW
20	MUX	SEG32~SEG35 端口功能选择 参数选择详见下表	RW
19:8	Reserved	-	-
7	S39C4	SEG39/COM4 输出控制位 SEG/COM 管脚功能选择由 CR0.DUTY 决定 0b0: SEG/COM 输出使能 0b1: SEG/COM 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW
6	S38C5	SEG38/COM5 输出控制位 SEG/COM 管脚功能选择由 CR0.DUTY 决定 0b0: SEG/COM 输出使能 0b1: SEG/COM 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW
5	S37C6	SEG37/COM6 输出控制位 SEG/COM 管脚功能选择由 CR0.DUTY 决定 0b0: SEG/COM 输出使能 0b1: SEG/COM 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW
4	S36C7	SEG36/COM7 输出控制位 SEG/COM 管脚功能选择由 CR0.DUTY 决定 0b0: SEG/COM 输出使能 0b1: SEG/COM 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW
3:0	Sx (x=35~32)	SEGx 输出控制位 0b0: SEG 输出使能 0b1: SEG 输出关闭, 可以使用其他功能, 如 IO、模拟输入输出	RW

VLCDy/SEGx PAD 状态描述							寄存器配置				
状态		VLCDH/ SEG35	VLCD4/ SEG34	VLCD3/ SEG33	VLCD2/ SEG32	VLCD1/ SEG31	MUX	S<35:31>	BSEL		
VLCDy/SEGx PAD 选择 GPIO, LCD disable(LCD_ON=0)时		IO	IO	IO	IO	IO	-	-	-	-	-
VLCDy/SEGx PAD 选择 GPIO, LCD enable(LCD_ON=1)时, 这时只能选择内部电阻工作模式	小电阻（大电流）模式	IO	IO	IO	IO	IO	0b1	0b11111	1	1	0
	中电阻（中电流）模式	IO	IO	IO	IO	IO	0b1	0b11111	0	1	0
	大电阻（小电流）模式	IO	IO	IO	IO	IO	0b1	0b11111	1	0	0
VLCDy/SEGx PAD 选择 GPIO 或 LCD 特殊管脚, LCD enable(LCD_ON=1)时, 选择外部电阻工作模式, 内部电阻断路		VLCDH	VLCD4	VLCD3	VLCD2	VLCD1	0b0	0bX0000	0	0	0
VLCDy/SEGx PAD 选择 GPIO 或 LCD 特殊管脚, LCD enable(LCD_ON=1)时, 选择内部电阻工作模式	小电阻（大电流）模式	SEG35/IO	SEG34/IO	SEG33/IO	SEG32/IO	SEG31/IO	0b0	-	1	1	0
	中电阻（中电流）模式	SEG35/IO	SEG34/IO	SEG33/IO	SEG32/IO	SEG31/IO	0b0	-	0	1	0
	大电阻（小电流）模式	SEG35/IO	SEG34/IO	SEG33/IO	SEG32/IO	SEG31/IO	0b0	-	1	0	0
VLCDy/SEGx PAD 选择 LCD 特殊管脚, LCD enable(LCD_ON=1)时, 选择外部电容工作模式, 内部电阻断路		VLCDH	VLCDL	VLCD3	VLCD2	VLCD1	0b0	0b00000	0	0	1

23.5.8 LCD_RAMx (x = 0~7)

偏移地址 (Offset) : 0x040 (LCDRAM0) , 0x044 (LCDRAM1) , 0x048 (LCDRAM2) , 0x04C (LCDRAM3) , 0x050 (LCDRAM4) , 0x054 (LCDRAM5) , 0x058 (LCDRAM6) , 0x05C (LCDRAM7)

Offset	Bit Position																																
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0xFFFFFFFF																																
Name	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	标记	功能描述	读写
31:0	Dn (n=31~0)	LCD 点输出 具体显示请参见 LCD 显示模式 相关描述。 0b0: 对应的 SEG COM 交叉点不亮 0b1: 对应的 SEG COM 交叉点亮	RW

23.5.9 LCD_RAMx (x = 8~F)

偏移地址 (Offset) : 0x060 (LCDRAM8) , 0x064 (LCDRAM9) , 0x068 (LCDRAMA) , 0x06C (LCDRAMB) , 0x070 (LCDRAMC) , 0x074 (LCDRAMD) , 0x078 (LCDRAME) , 0x07C (LCDRAMF)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	Reserved																															
Access																																

位/位域	标记	功能描述	读写
31:24	Reserved	-	-
23:0	Dn (n=23~0)	LCD 点输出 具体显示请参见 LCD 显示模式 相关描述。 0b0: 对应的 SEG COM 交叉点不亮 0b1: 对应的 SEG COM 交叉点亮	RW

24 模拟电压比较器 (VC)

24.1 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。

- 当“+”输入端电压高于“-”输入端电压时，电压比较器输出为高电平；
- 当“+”输入端电压低于“-”输入端电压时，电压比较器输出为低电平。

24.2 主要特性

- 支持 64 阶分压基准
- 支持 8 个内外部输入信号
- 支持 2 种输出极性
- 支持窗口比较功能
- 多种滤波时间配置，有效增强芯片的抗干扰能力
- 支持 3 种中断触发方式：高电平/上升沿/下降沿
- 电压比较器的输出可以作为定时器的输入
- 支持在 DeepSleep 模式下工作



说明

使用 VC 需要使能 BGR，具体请参见《模拟其他寄存器》章节 BGR 寄存器相关描述。

24.3 功能说明

24.3.1 功能框图

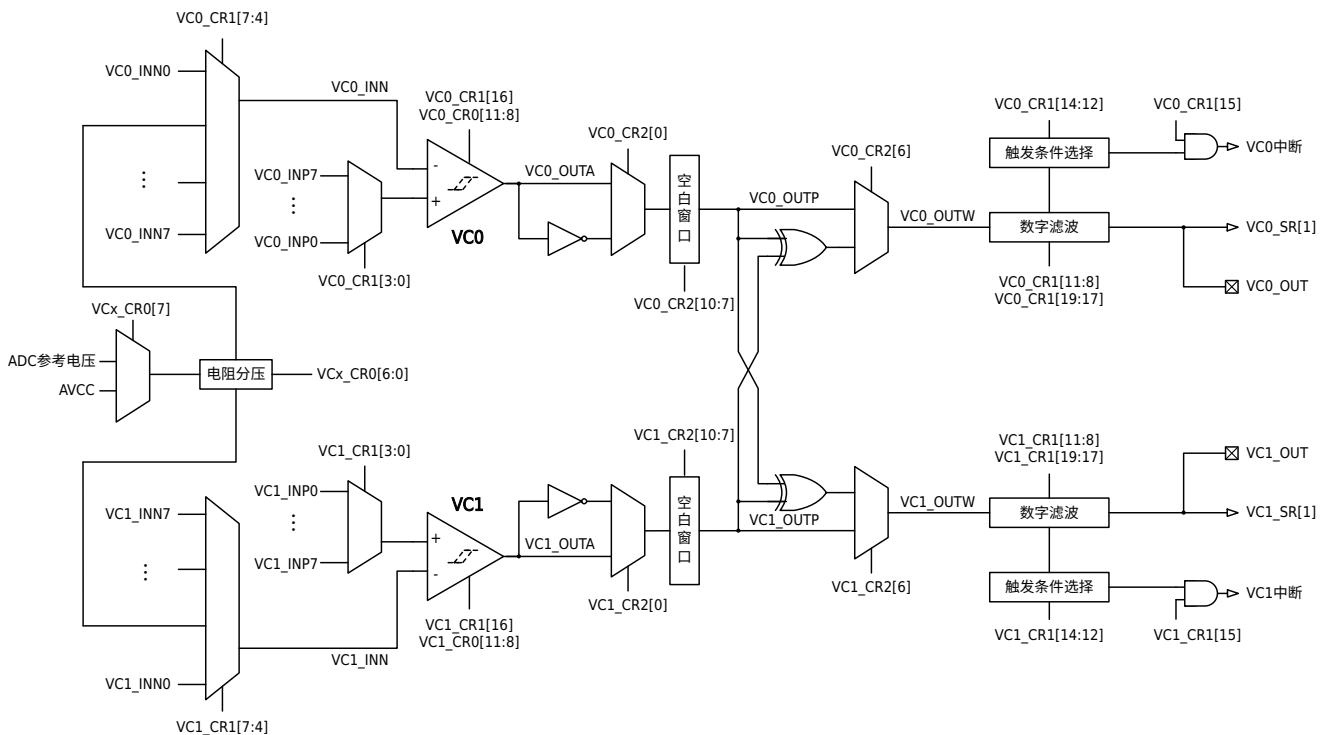


图 24-1 VC 功能框图

24.3.2 建立/响应时间

当使用电压比较器时，从 VC 使能或者 VC 的两端输入电压变化到输出正确结果的时间由 VCx_CR0.BIAS 位决定，电流越大 VC 响应越快，具体时间请参见《数据手册》“电气特性-工作条件-VC 特性”章节描述。

当 VCx_CR0.BIAS=0b10/0b11 时，需要启动 BGR (BGR 启动后的稳定时间 $T_{stable(BGR)}$ 请参见《数据手册》“电气特性-工作条件-内置的参考电压”章节描述)，电压比较器需要等待内部 BGR 稳定后才能正常输出。

当电阻分压电路使能时 (VCx_CR0.DIVEN=0b1)，电压比较器需要等待电阻分压电路稳定 (T_{VDAC_setup}) 后才能正常输出。如果使用了 ADC 参考电压作为比较基准电压 (VCx_CR0.REF=0b1)，电压比较器则需要等待 T_{VDAC_setup} 和 $T_{stable(REFBUF)}$ 中最大值时间才能正常输出。



说明

REFBUF 启动后的稳定时间 $T_{stable(REFBUF)}$ 请参见《数据手册》“电气特性-工作条件-内置的参考电压”章节描述；电压比较器中内置 DAC 在开启后输出分压电压的稳定时间 T_{VDAC_setup} 请参见《数据手册》“电气特性-工作条件-VC 特性”章节描述。

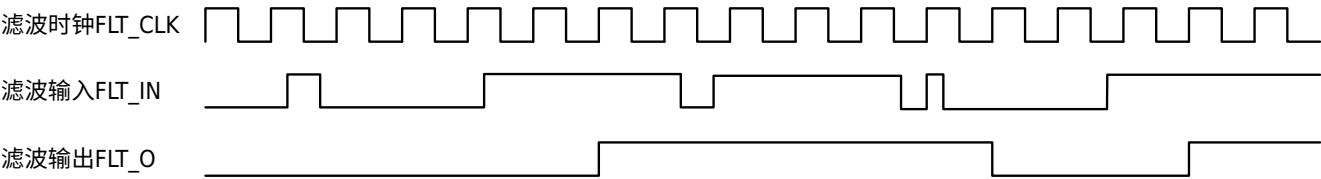
24.3.3 数字滤波

在电压比较器固有的建立/响应时间之外，用户可以设置更长的滤波时间来过滤掉系统噪声，比如马达停止时的大电流噪声。

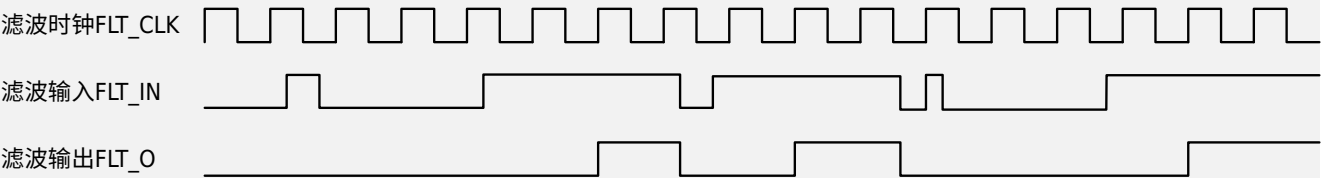
从寄存器 VCx_SR[1] 可以读出数字滤波后的信号电平；当 GPIO 功能配置为 VCx_OUT 时，数字滤波后的信号可以从 GPIO 输出以方便测量。

使能数字滤波模块，选择滤波时间为 2 个滤波时钟周期，滤波示意如下所示：

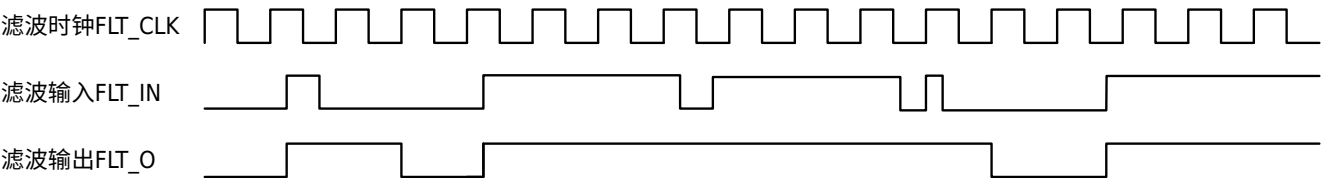
普通滤波模式



滤峰模式



滤谷模式



滤刺模式

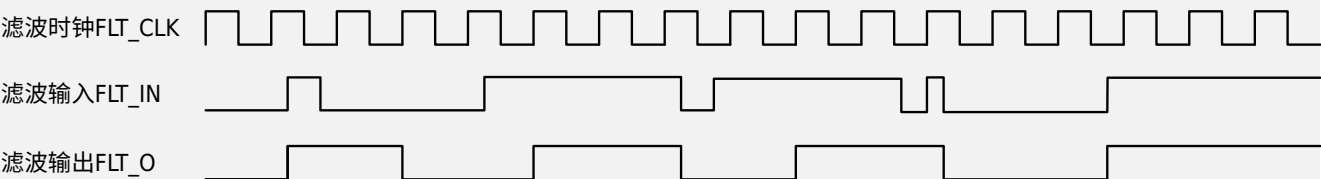


图 24-2 VC 滤波响应时间

24.3.4 迟滞功能

电压比较器可以选择迟滞功能，迟滞功能使能后的图示如下：

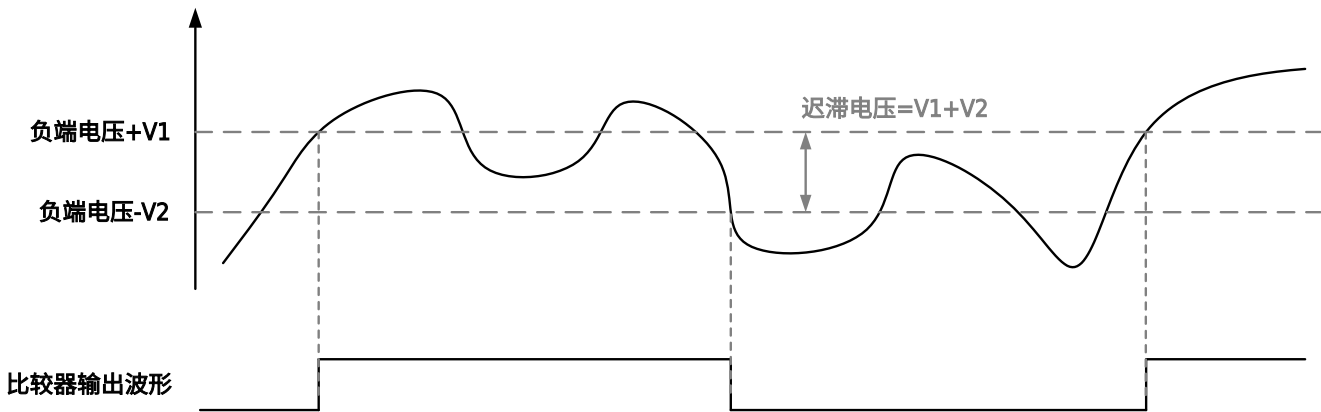


图 24-3 VC 迟滞功能

24.3.5 比较结果极性配置

当配置 VCx_CR2.POL 为 0 时，VCx_OUTA 与 VCx_OUTP 电平相同。

当配置 VCx_CR2.POL 为 1 时，VCx_OUTA 与 VCx_OUTP 电平相反。

24.3.6 窗口比较功能

当配置 VCx_CR2.WINDOW 为 0 时，VCx_OUTW 信号与 VCx_OUTP 信号相同。

当配置 VCx_CR2.WINDOW 为 1 时，VCx_OUTW 信号为 VC0_OUTP 与 VC1_OUTP 的异或值。

24.3.7 空白窗口

定时器 TIM0/3 输出的 PWM 信号可以作为空白窗口信号对比较器的输出进行控制，配置 VCx_CR2.BLANK 开启空白窗口功能或选择相应 PWM 空白窗口信号。

空白窗口的目的是防止 PWM 周期开始时，出现的电流毛刺导致 VC 比较结果异常。空白窗口功能示意图如下：

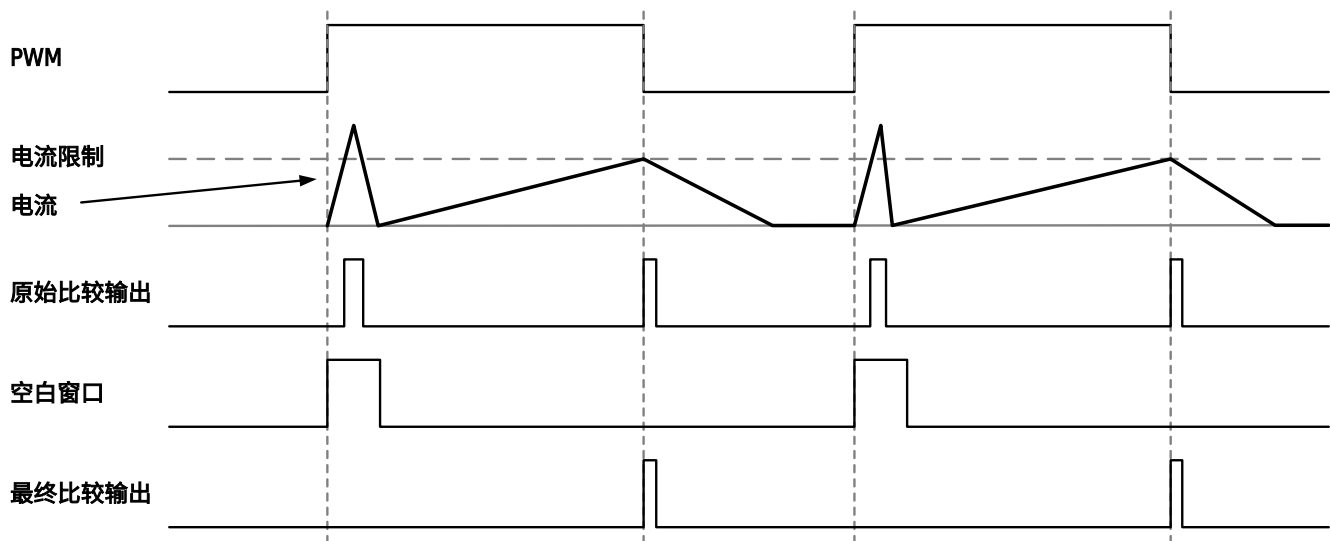


图 24-4 VC 空白窗口

24.4 寄存器

24.4.1 寄存器总表

基地址：0x40002800

表 24-1 模拟电压比较器 (VC) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	VC0_CR0	VC0 控制寄存器 0	32
0x04	VC0_CR1	VC0 控制寄存器 1	32
0x08	VC0_CR2	VC0 控制寄存器 2	32
0x0C	VC0_SR	VC0 状态寄存器	32
0x10	VC1_CR0	VC1 控制寄存器 0	32
0x14	VC1_CR1	VC1 控制寄存器 1	32
0x18	VC1_CR2	VC1 控制寄存器 2	32
0x1C	VC1_SR	VC1 状态寄存器	32
0x20	VC_TRIM	VC 偏移微调寄存器	32

24.4.2 控制寄存器 0 (VCx_CR0) (x=0~1)

偏移地址 (Offset) : 0x00 (VC0_CR0) , 0x10 (VC1_CR0)

Offset	Bit Position																																			
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reset	0x00000020																																			
Name	Reserved																				HYS		BIAS		REF		DIVEN		DIV							
Access																					RW		RW		RW		RW		RW							

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:10	HYS	迟滞功能配置 0b00: 无迟滞 0b01: 迟滞电压约 10mV 0b10: 迟滞电压约 20mV 0b11: 迟滞电压约 30mV	RW
9:8	BIAS	功耗选择 0b00: 低功耗档 0b01: 中低功耗档 0b10: 中功耗档 0b11: 高功耗档  说明 功耗档位越大，响应速度越快。具体档位的电流规格请参见《数据手册》“电气特性-VC 特性”章节。 选择 0b10 和 0b11 时，需要开启 BGR，等待时间详见《数据手册》。	RW
7	REF	电阻分压电路参考电压选择 0b0: AVCC 0b1: ADC 的参考电压  说明 VC0/VC1 共用该位设置。	RW
6	DIVEN	电阻分压电路使能控制 0b0: 禁止 0b1: 使能  说明 VC0/VC1 共用该位设置。	RW

位/位域	标记	功能描述	读写
5:0	DIV	电阻分压电路输出电压配置 ● 0b000000: 0/64 Vref ● 0b000001: 1/64 Vref ● ... ● 0b111110: 62/64 Vref ● 0b111111: 63/64 Vref  说明 VC0/VC1 共用该位设置，Vref 来源由 REF 控制位决定。	RW

 说明

在使能 VC 的情况下，如果对上表中的 VC 的控制寄存器作修改，需要等待 T_{setup} 时间 VC 才能正常工作， T_{setup} 的数值详见《数据手册》。

24.4.3 控制寄存器 1 (VCx_CR1) (x=0~1)

偏移地址 (Offset) : 0x04 (VC0_CR1) , 0x14 (VC1_CR1)

Offset	Bit Position																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x000000FF																															
Name	Reserved													FLT_MODE	FLTCLK	EN	IE	LEVEL	RISING	FALLING	FLTTIME			NSEL			PSEL					
Access														RW	RW	RW	RW	RW	RW	RW	RW			RW			RW					

位/位域	标记	功能描述	读写
31:20	Reserved	-	-
19:18	FLT_MODE	滤波器模式选择 0b00: 普通滤波模式, 滤波器输入翻转后, 需要保持一定时间, 滤波器输出才会翻转 0b01: 滤峰模式, 滤波器输入置逻辑 1 后, 需要保持一定时间, 滤波输出才会置逻辑 1; 而滤波器输入置 0, 滤波器输出会立即置 0, 这个模式的目的是滤除不够长的输入波峰 0b10: 滤谷模式, 滤波器输入置逻辑 0 后, 需要保持一定时间, 滤波输出才会置逻辑 0; 而滤波器输入置 1, 滤波器输出会立即置 1, 这个模式的目的是滤除不够长的输入波谷 0b11: 滤刺模式, 滤波器输入翻转后, 输出也会立即翻转, 之后会在一定时间内无视滤波器的输入。在这个模式下, 滤波器输出会紧随输入, 同时会避免输出信号出现毛刺	RW
17	FLTCLK	滤波时钟选择 0b0: RC256K 0b1: PCLK	RW
16	EN	VC 使能控制 0b0: 禁止 0b1: 使能	RW
15	IE	VC 中断使能控制 0b0: 禁止 0b1: 使能	RW
14	LEVEL	滤波信号高电平触发中断使能控制 0b0: 禁止 0b1: 使能	RW
13	RISING	滤波信号上升沿触发中断使能控制 0b0: 禁止 0b1: 使能	RW
12	FALLING	滤波信号下降沿触发中断使能控制 0b0: 禁止 0b1: 使能	RW
11:8	FLTTIME	滤波时间配置 N 表示滤波时钟周期数 0x0: 数字滤波无效	RW

位/位域	标记	功能描述	读写
11:8	FLTTIME	● 0x1: N = 2 ● 0x2: N = 4 ● 0x3: N = 8 ● 0x4: N = 16 ● 0x5: N = 32 ● 0x6: N = 64 ● 0x7: N = 128 ● 0x8: N = 256 ● 0x9: N = 512 ● 0xA: N = 1024 ● 0xB: N = 2048 ● 其他: N = 4096	RW
7:4	NSEL	电压比较器负端输入信号选择 ● 0b0000: VCx_INN0 (x=0, PA00; x=1, PA02) ● 0b0001: VCx_INN1 (PA04) ● 0b0010: VCx_INN2 (PA05) ● 0b0011: VCx_INN3 (x=0, PB01; x=1, PB03) ● 0b0100: VCx_INN4 (VCAP 电压) ● 0b0101: VCx_INN5 (内置电阻分压输出电压) ● 0b0110: VCx_INN6 (内置温度传感器输出电压) ● 0b0111: VCx_INN7 (PA04) ● 其他: 禁止设置	RW
3:0	PSEL	电压比较器正端输入信号选择 ● 0b0000: VCx_INP0 (PA01) ● 0b0001: VCx_INP1 (PA03) ● 0b0010: VCx_INP2 (PA06) ● 0b0011: VCx_INP3 (PA07) ● 0b0100: VCx_INP4 (PB04) ● 0b0101: VCx_INP5 (PB05) ● 0b0110: VCx_INP6 (PB06) ● 0b0111: VCx_INP7 (PB07) ● 其他: 禁止设置	RW



说明

在使能 VC 的情况下, 如果对 NSEL/PSEL 作修改, 需要等待 T_{setup} 时间 VC 才能正常工作, T_{setup} 的数值详见《数据手册》。

24.4.4 控制寄存器 2 (VCx_CR2) (x=0~1)

偏移地址 (Offset) : 0x08 (VC0_CR2) , 0x18 (VC1_CR2)

Offset	Bit Position																															
'	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																					RW BLANK		RW WINDOW	RW TIMBK	RW TIM3CLR	RW TIM2CLR	RW TIM1CLR	RW TIM0CLR	RW POL		
Access																						RW		RW	RW	RW	RW	RW	RW	RW		

位/位域	标记	功能描述	读写
31:11	Reserved	-	-
10:7	BLANK	VCx 空白窗口源选择 ● 0b0000: 无空白窗口 ● 0bxxx1: TIM0 OCREFB ● 0bxx1x: TIM3 OCREFOB ● 0bx1xx: TIM3 OCREF1B ● 0b1xxx: TIM3 OCREF2B	RW
6	WINDOW	窗口比较功能配置 ● 0b0: 禁止 ● 0b1: 使能	RW
5	TIMBK	VCx 滤波后信号输出到 TIM0/1/2/3 刹车使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
4	TIM3CLR	VCx 滤波后信号输出到 TIM3 REFCLR 使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
3	TIM2CLR	VCx 滤波后信号输出到 TIM2 REFCLR 使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
2	TIM1CLR	VCx 滤波后信号输出到 TIM1 REFCLR 使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
1	TIM0CLR	VCx 滤波后信号输出到 TIM0 REFCLR 使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW
0	POL	VCx_OUTP 信号极性配置 ● 0b0: 正端大于负端时 VCx_OUTP 高 ● 0b1: 正端大于负端时 VCx_OUTP 低	RW

24.4.5 状态寄存器 (VCx_SR) (x=0~1)

偏移地址 (Offset) : 0x0C (VC0_SR) , 0x1C (VC1_SR)

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														RO	FLTV
Access																															RW	INTF

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	FLTV	滤波后的信号电平 ● 0b0: 低电平 ● 0b1: 高电平	RO
0	INTF	中断标志 写 0 清除标志, 写 1 无效 读取: ● 0b0: 未发生中断 ● 0b1: 已发生中断	RW

24.4.6 偏移微调寄存器 (VC_TRIM)

Offset	Bit Position																															
0x20	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000 （外设复位无效）																															
Name	Reserved																TRIM1				Reserved				TRIM0							
Access																	RW								RW							

位/位域	标记	功能描述	读写
31:12	Reserved	-	-
11:8	TRIM1	VC1 校准值 Flash 中保存了 2 组校准值。请根据实际情况选择相应 Flash 地址，将该地址内的校准值读出并写入 VC_TRIM.TRIM1。 - 比较器正端或负端的参考电压小于等于 1/2 AVCC 时：0x00101BC2 - 比较器正端或负端的参考电压大于 1/2 AVCC 时：0x00101BC3	RW
8:4	Reserved	-	-
3:0	TRIM0	VC0 校准值 Flash 中保存了 2 组校准值。请根据实际情况选择相应 Flash 地址，将该地址内的校准值读出并写入 VC_TRIM.TRIM0。 - 比较器正端或负端的参考电压小于等于 1/2 AVCC 时：0x00101BC0 - 比较器正端或负端的参考电压大于 1/2 AVCC 时：0x00101BC1	RW

25 低电压检测器 (LVD)

25.1 简介

LVD 可用于监测 DVCC 及芯片管脚的电压。当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号，用户可根据该信号执行一些紧急任务。

25.2 主要特性

- 4 路监测源
- 16 阶阈值电压，1.8~4.8V 可选
- 8 种触发条件，高电平、上升沿、下降沿组合
- 2 种触发结果：复位、中断
- 12 阶滤波配置，防止误触发
- 具备迟滞功能，强力抗干扰

25.3 功能说明

25.3.1 LVD 框图

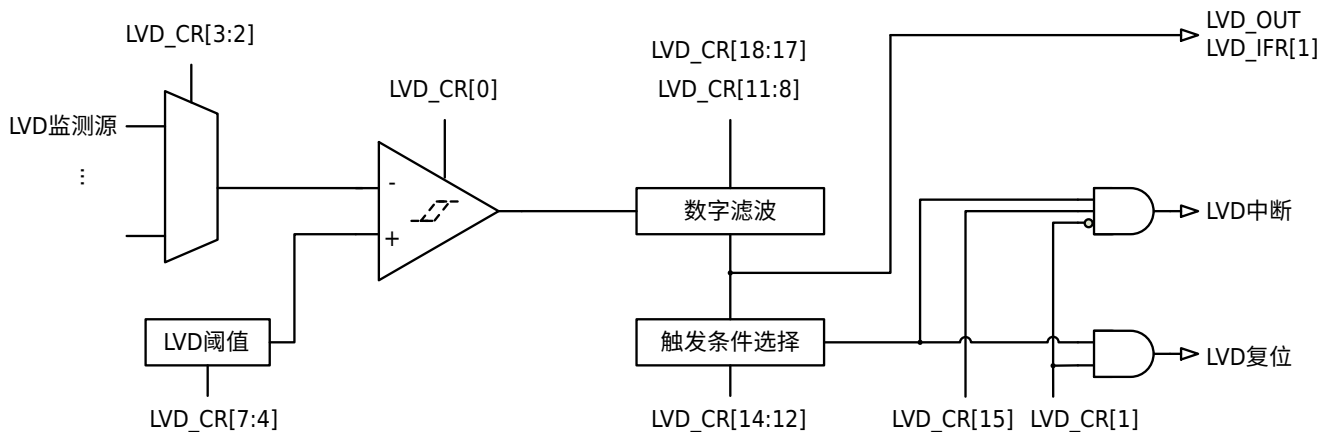


图 25-1 LVD 框图



说明

LVD 监测源的选择请参见“[LVD 配置寄存器 \(LVD_CR\)](#)”的 SOURCE 位功能描述。

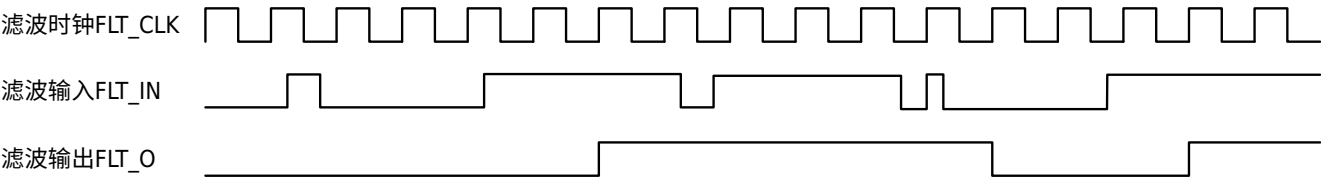
25.3.2 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.DEBOUNCE_TIME 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。数字滤波模块所使用的滤波时钟为振荡频率约 256kHz 的内置 RC 振荡器。

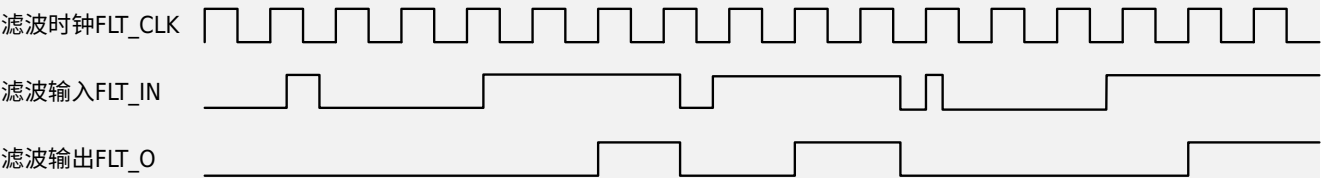
从寄存器 LVD_IFR[1] 可以读出数字滤波后的信号电平；当 GPIO 功能配置为 LVD_OUT 时，数字滤波后的信号可以从 GPIO 输出以方便测量。

使能数字滤波模块，选择滤波时间为 2 个滤波时钟周期，滤波示意如下所示：

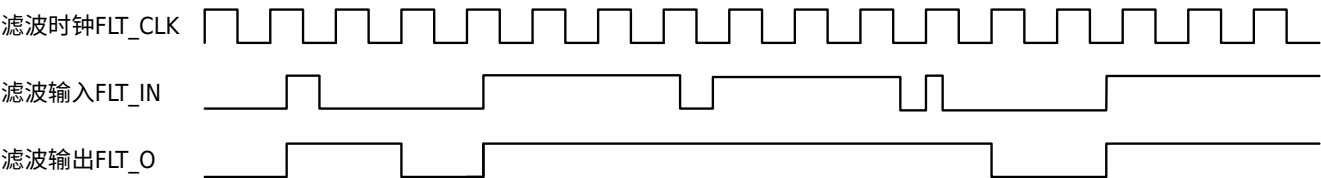
普通滤波模式



滤峰模式



滤谷模式



滤刺模式

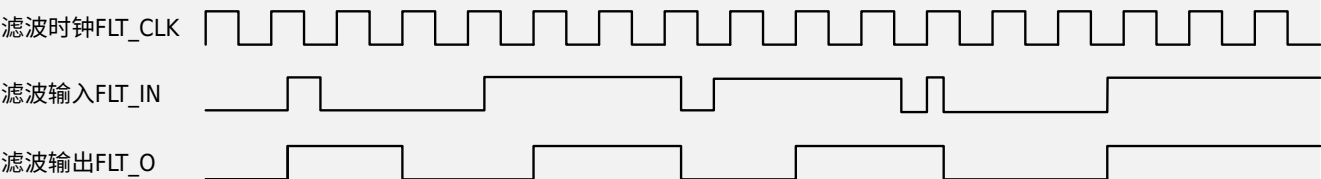


图 25-2 LVD 滤波输出

25.3.3 迟滞功能

LVD 内置的电压比较器具有迟滞功能，可以增强芯片的抗干扰能力。LVD 的输出信号会等到输入信号相较阈值电压高于或低于迟滞电压/2 后才发生翻转，其输入输出信号示意如下：

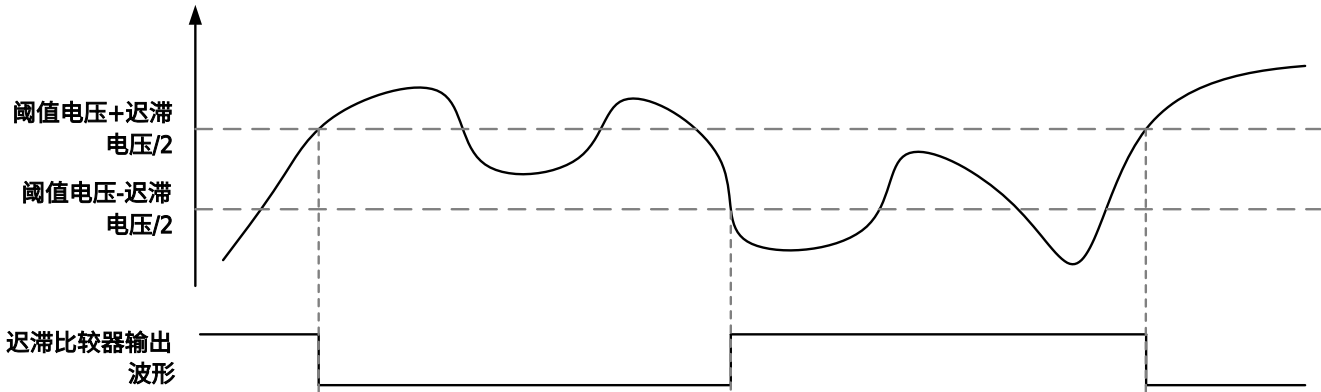


图 25-3 LVD 迟滞响应

25.4 操作示例

25.4.1 LVD 配置为低电压复位

背景信息

在本模式下，监测电压低于阈值电压时复位 MCU。

操作步骤

- 步骤 1. 配置 LVD_CR.SOURCE_SEL，选择待监测的电压来源。
- 步骤 2. 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。
- 步骤 3. 配置 LVD_CR.DEBOUNCE_TIME，选择 LVD 滤波时间。
- 步骤 4. 配置 LVD_CR.FLT_MODE，选择滤波方式。
- 步骤 5. 设置 LVD_CR.HTEN 为 1，选择高电平触发 LVD 动作。
- 步骤 6. 设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。
- 步骤 7. 设置 LVD_CR.LVDEN 为 1，使能 LVD。

25.4.2 LVD 配置为电压变化中断

背景信息

在本模式下，监测电压高于或低于阈值电压时产生中断。

操作步骤

- 步骤 1. 配置 LVD_CR.SOURCE_SEL，选择待监测的电压来源。
- 步骤 2. 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。
- 步骤 3. 配置 LVD_CR.DEBOUNCE_TIME，选择 LVD 滤波时间。
- 步骤 4. 配置 LVD_CR.FLT_MODE，选择滤波方式。
- 步骤 5. 设置 LVD_CR.RTEN 和 LVD_CR.FTEN 为 1，选择电平变化触发 LVD 动作。
- 步骤 6. 设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。
- 步骤 7. 设置 LVD_CR.LVDEN 为 1，使能 LVD。
- 步骤 8. 等待建立时间 T_{setup} 的最大值加上设置的滤波时间。
- 步骤 9. 设置 LVD_CR.IE 为 1，使能 LVD 中断。
- 步骤 10. 使能 NVIC 中断向量表中的 LVD 中断。
- 步骤 11. 在 LVD 的中断服务程序中执行用户需要进行的操作；退出中断服务程序前向 LVD_IFR 写入 0x00 以清除中断标志。

25.5 寄存器

25.5.1 寄存器总表

基地址：0x40002C00

表 25-1 低电压检测器 (LVD) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x000	LVD_CR	LVD 配置寄存器	32
0x004	LVD_IFR	LVD 中断标志寄存器	32

25.5.2 LVD 配置寄存器 (LVD_CR)

Offset	Bit Position																																							
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Reset	0x00000000																																							
Name	Reserved														FLT_MODE		Reserved		IE		HTEN		RTEN		FTEN		DEBOUNCE_TIME				VTDS				SOURCE_SEL		ACT		LV DEN	
Access															RW				RW		RW		RW		RW		RW				RW				RW		RW		RW	

位/位域	标记	功能描述	读写
31:19	Reserved	-	-
18:17	FLT_MODE	滤波器模式选择 0b00: 普通滤波模式, 滤波器输入翻转后, 需要保持一定时间, 滤波器输出才会翻转 0b01: 滤峰模式, 滤波器输入置逻辑 1 后, 需要保持一定时间, 滤波输出才会置逻辑 1; 而滤波器输入置 0, 滤波器输出会立即置 0, 这个模式的目的是滤除不够长的输入波峰 0b10: 滤谷模式, 滤波器输入置逻辑 0 后, 需要保持一定时间, 滤波输出才会置逻辑 0; 而滤波器输入置 1, 滤波器输出会立即置 1, 这个模式的目的是滤除不够长的输入波谷 0b11: 滤刺模式, 滤波器输入翻转后, 输出也会立即翻转, 之后会在一定时间内无视滤波器的输入。在这个模式下, 滤波器输出会紧随输入, 同时会避免输出信号出现毛刺	RW
16	Reserved	-	-
15	IE	LVD 中断使能 0b0: 禁止 0b1: 使能	RW
14	HTEN	高电平触发使能 被监测电压低于阈值电压 0b0: 禁止 0b1: 使能	RW
13	RTEN	上升沿触发使能 被监测电压从高于阈值电压变为低于阈值电压 0b0: 禁止 0b1: 使能	RW
12	FTEN	下降沿触发使能 被监测电压从低于阈值电压变为高于阈值电压 0b0: 禁止 0b1: 使能	RW
11:8	DEBOUNCE_TIME	数字滤波时间配置 N 表示滤波时钟周期数 0x0: 数字滤波无效 0x1: N = 2 0x2: N = 4	RW

位/位域	标记	功能描述	读写
11:8	DEBOUNCE_TIME	● 0x3: N = 8 ● 0x4: N = 16 ● 0x5: N = 32 ● 0x6: N = 64 ● 0x7: N = 128 ● 0x8: N = 256 ● 0x9: N = 512 ● 0xA: N = 1024 ● 0xB: N = 2048 ● 其他: N = 4096	RW
7:4	VTDS	LVD 阈值电压选择 ● 0b0000: 1.8V ● 0b0001: 2.0V ● 0b0010: 2.2V ● 0b0011: 2.4V ● 0b0100: 2.6V ● 0b0101: 2.8V ● 0b0110: 3.0V ● 0b0111: 3.2V ● 0b1000: 3.4V ● 0b1001: 3.6V ● 0b1010: 3.8V ● 0b1011: 4.0V ● 0b1100: 4.2V ● 0b1101: 4.4V ● 0b1110: 4.6V ● 0b1111: 4.8V	RW
3:2	SOURCE_SEL	LVD 监测来源选择 ● 0b00: LVD_IN0 (DVCC 电压) ● 0b01: LVD_IN1 (PC13 端口输入电压) ● 0b10: LVD_IN2 (PB08 端口输入电压) ● 0b11: LVD_IN3 (PB07 端口输入电压)	RW
1	ACT	LVD 触发动作选择 ● 0b0: NVIC 中断 ● 0b1: 系统复位	RW
0	LVDEN	LVD 使能控制 ● 0b0: 禁止 ● 0b1: 使能	RW

25.5.3 LVD 中断标志寄存器 (LVD_IFR)

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																														FILTER	INTF
Access																															RO	RW

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	FILTER	数字滤波器输出的电平值 ● 0b0：数字滤波器输出低电平 ● 0b1：数字滤波器输出高电平	RO
0	INTF	LVD 中断标志 写 0 清除中断标志，写 1 对该位的值无影响 读取： ● 0b0：未发生中断 ● 0b1：发生中断	RW

26 CRC 运算（CRC）

26.1 简介

在许多应用中，都需要 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛应用。本模块可采用 CRC16/CCITT，CRC16/CCITT-FALSE，CRC16/XMODE，CRC16/X25，CRC32，CRC32/MPEG-2 共计 6 种算法对数据进行运算和校验。

26.2 功能说明

26.2.1 功能框图

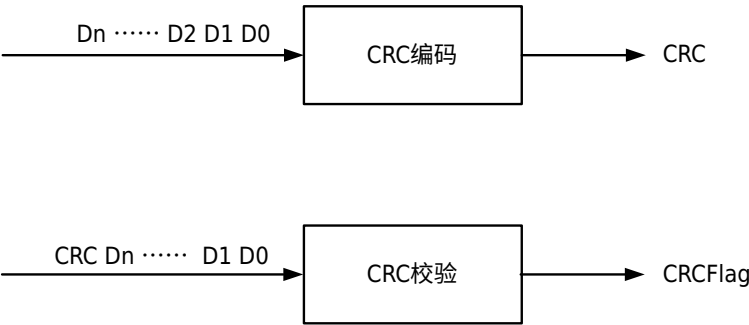


图 26-1 CRC 应用示意图

26.2.2 CRC 参数模型

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 16 位、32 位的 CRC。CRC 各算法对应参数模型如下表。

表 26-1 CRC 参数模型

CRC 算法名称	多项式公式	POLY	INIT	XOROUT	REFIN	REFOUT	校验值
CRC16/X25	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	0xFFFF	TRUE	TRUE	0xF0B8
CRC16/CCITT	$X^{16}+X^{12}+X^5+1$	0x1021	0x0000	0x0000	TRUE	TRUE	0x0000
CRC16/CCITT-FALSE	$X^{16}+X^{12}+X^5+1$	0x1021	0xFFFF	0x0000	FALSE	FALSE	0x0000
CRC16/XMODEM	$X^{16}+X^{12}+X^5+1$	0x1021	0x0000	0x0000	FALSE	FALSE	0x0000
CRC32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	0xFFFFFFFF	TRUE	TRUE	0xDEBB20E3
CRC32/MPEG-2	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	0x04C11DB7	0xFFFFFFFF	0x00000000	FALSE	FALSE	0x00000000

- POLY：十六进制多项式。
- INIT：CRC 初始值，和位宽一致。
- XOROUT：计算结果与此参数进行异或运算后得到最终的 CRC 编码，和位宽一致。
- REFIN：原始数据是否进行高低位翻转，TRUE 为翻转，FALSE 为不翻转。
- REFOUT：编码结束后，得到的 CRC 编码是否进行高低位翻转，TRUE 为翻转，FALSE 为不翻转。

在使用本模块时，需要根据编码方式需求配置 CR.SEL 位并写入初值 INIT，算法内其他参数已做预配置。本模块功能包括：

- CRC 编码和 CRC 校验;
- 3 种位宽访问方式 8 位, 16 位, 32 位:
 - ▶ 8 位位宽下输入数据示例为 0x00、0x11、0x22、0x33、0x44、0x55、0x66、0x77;
 - ▶ 16 位位宽下输入数据示例为 0x1100、0x3322、0x5544、0x7766;
 - ▶ 32 位位宽下输入数据示例为 0x33221100、0x77665544。

26.2.3 CRC16 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值, 操作流程如下:

1. 向 CRC_CR.CR 写入 0b0 和 CRC_CR.SEL 写入对应数据, 选择对应 CRC16 规格算法。
2. 根据 [CRC 参数模型](#)向 CRC_RESULT[15:0]写入 0x0000 或者 0xFFFF, 初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式, 依次写入 CRC_DATA 寄存器。



说明

按 8 位组织方式写 CRC 值, 应先写低位, 后写高位。

4. 读取 CRC_RESULT[15:0], 即可获得 16 位 CRC 编码值。

26.2.4 CRC16 校验模式

校验模式可以校验已编码的数据是否被篡改, 操作流程如下所示:

1. 向 CRC_CR.CR 写入 0b0 和 CRC_CR.SEL 写入对应数据, 选择对应 CRC16 规格算法。
2. 根据 [CRC 参数模型](#)向 CRC_RESULT[15:0]写入 0x0000 或者 0xFFFF, 初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式, 依次写入 CRC_DATA 寄存器。



说明

按 8 位组织方式写 CRC 值, 应先写低位, 后写高位。

4. 校验码以 16 位位宽将数据写入 CRC_DATA 寄存器。



说明

CRC16/CCITT-FALSE 和 CRC16/XMODEM 在校验时, 需要对校验码进行高 8 位和低 8 位对调操作。
例如: 0x1234->0x3412

5. 读取 CRC_CR.FLAG 的值, 为 1 则表示校验成功, 为 0 则表示校验失败。

26.2.5 CRC32 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值, 操作流程如下:

1. 向 CRC_CR.CR 写入 0b1 和 CRC_CR.SEL 写入对应数据, 选择对应 CRC32 规格算法。
2. 向 CRC_RESULT[31:0]写入 0xFFFFFFFF, 初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式, 依次写入 CRC_DATA 寄存器。



说明

按 8 位组织方式写 CRC 值, 应先写低位, 后写高位。

4. 读取 CRC_RESULT[31:0], 即可获得 32 位 CRC 编码值。

26.2.6 CRC32 校验模式

校验模式可以校验已编码的数据是否被篡改, 操作流程如下所示:

1. 向 CRC_CR.CR 写入 0b1 和 CRC_CR.SEL 写入对应数据, 选择对应 CRC32 规格算法。
2. 向 CRC_RESULT[31:0]写入 0xFFFFFFFF, 初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式, 依次写入 CRC_DATA 寄存器。

**说明**

按 8 位组织方式写 CRC 值，应先写低位，后写高位。

4. 校验码按 32 位组织方式，写入 CRC_DATA 寄存器。

**说明**

CRC32/MPEG-2 在校验时，需要对其高 16 位和低 16 位进行对调操作，且分别对其高、低 16 位中的 16 位数据进行高 8 位和低 8 位进行对调操作。

例如：0x12345678->0x78563412

5. 读取 CRC_CR.FLAG 的值，为 1 则表示校验成功，为 0 则表示校验失败。

26.3 寄存器

26.3.1 寄存器总表

基地址：0x40020800

表 26-2 CRC 运算（CRC）寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	CRC_CR	CRC 控制寄存器	32
0x04	CRC_RESULT	CRC 结果寄存器	32
0x80~0xFC	CRC_DATA	CRC 数据寄存器	32

26.3.2 控制寄存器 (CRC_CR)

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000001																																	
Name	Reserved																												RW SEL		RO FLAG		RW CR	
Access																															RW		RW	

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4:2	SEL	CRC 选择位 CR=0b0 时，Bit4~2 对应 CRC 算法选择： 0b000：CRC16/X25 0b001：CRC16/CCITT 0b010：CRC16/CCITT-FALSE 0b011：CRC16/XMODEM - 其他：禁止设置 CR=0b1 时，Bit4~2 对应 CRC 算法选择： 0b000：CRC32 0b001：CRC32/MPEG-2 - 其他：禁止设置	RW
1	FLAG	校验结果位 0b0：当前校验错误 0b1：当前校验正确	RO
0	CR	运算控制位 0b0：CRC16 0b1：CRC32	RW

26.3.3 结果寄存器 (CRC_RESULT)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	RW RESULT																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	RESULT	结果位 ● 选择 CRC16 时，取 RESULT[15:0] ● 选择 CRC32 时，取 RESULT[31:0]	RW

26.3.4 数据寄存器（CRC_DATA）

Offset	Bit Position																															
0x80~0xF C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DATA																															
Access	WO																															

位/位域	标记	功能描述	读写
31:0	DATA	需要运算的数据 写入需要运算的数据	WO

 **说明**

CRC_DATA 寄存器位于 CRC 基址的 0x80 偏移地址处，基址的 0x84~0xFC 偏移地址处为 CRC_DATA 寄存器的映射地址，其访问和操作权限与 CRC_DATA 寄存器相同。

27 真随机数发生器 (TRNG)

27.1 简介

真随机数模块产生 64 位真随机数。

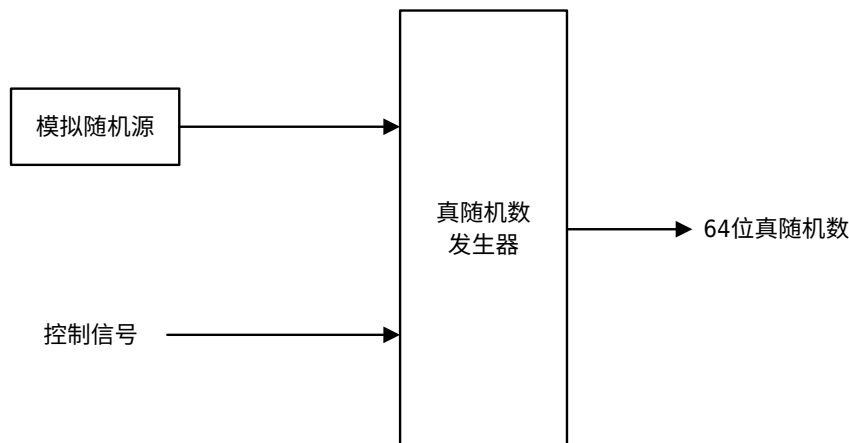


图 27-1 真随机数发生器功能框图

27.2 功能说明

27.2.1 真随机数产生原理

本模块采用内部的模拟随机源，每次启动都可以产生 64bits 真随机数。此外，还可以对真随机数生成的方式进行软件配置，详细内容可查看寄存器描述章节。生成的 64 位真随机数分别存放在 DATA0 和 DATA1 寄存器中。

27.3 操作示例

27.3.1 生成 64bits 真随机数的操作流程（上电第一次）

操作步骤

- 步骤 1.** 打开随机源电路：将 TRNG_CR.TRNGCIR_EN 设置为 0b1，启动随机源电路，开始输出串行真随机数。
- 步骤 2.** 选择重新装载初始值：将 TRNG_MODE.LOAD 设置为 0b1，使新产生的真随机数的初始值从随机源获得。
- 步骤 3.** 选择 PRNG64 的直接反馈的方式：将 TRNG_MODE.FDBK 设置为 0b1，将反馈信号与随机源异或后输入 PRNG 中。
- 步骤 4.** 选择 PRNG64 的移位次数：将 TRNG_MODE.CNT 设置为 0b110，选择移位 256 次。
- 步骤 5.** 生成真随机数：将 TRNG_CR.TRNG_RUN 设置为 0b1，硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 TRNG_CR.TRNG_RUN 清零。
- 步骤 6.** 选择不重新装载初始值：将 TRNG_MODE.LOAD 设置为 0b0。
- 步骤 7.** 选择 PRNG64 的直接反馈的方式：将 TRNG_MODE.FDBK 设置为 0b0，将反馈信号直接输入 PRNG 中。
- 步骤 8.** 选择 PRNG64 的移位次数：将 TRNG_MODE.CNT 设置为 0b100，选择移位 64 次。
- 步骤 9.** 生成真随机数：将 TRNG_CR.TRNG_RUN 设置为 0b1，硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 TRNG_CR.TRNG_RUN 清零。

步骤 10. 读取真随机数：软件在查询到 TRNG_CR.TRNG_RUN 变为“0”后，通过读取 TRNG_DATA0 和 TRNG_DATA1，得到 64-bit 真随机数。

步骤 11. 完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：将 TRNG_CR.TRNGCIR_EN 设置为 0b0，关闭随机源电路。

27.3.2 生成 64bits 真随机数的操作流程（非上电第一次生成）

操作步骤

步骤 1. 打开随机源电路：将 TRNG_CR.TRNGCIR_EN 设置为 0b1，启动随机源电路，开始输出串行真随机数。

步骤 2. 选择不重新装载初始值：将 TRNG_MODE.LOAD 设置为 0b0。

步骤 3. 选择 PRNG64 的直接反馈的方式：将 TRNG_MODE.FDBK 设置为 0b1，将反馈信号与随机源异或后输入 PRNG 中。

步骤 4. 选择 PRNG64 的移位次数：将 TRNG_MODE.CNT 设置为 0b110，选择移位 256 次。

步骤 5. 生成真随机数：将 TRNG_CR.TRNG_RUN 设置为 0b1，硬件根据真随机数生成配置进行操作，在操作完成后，硬件自动将 TRNG_CR.TRNG_RUN 清零。

步骤 6. 选择 PRNG64 的直接反馈的方式：将 TRNG_MODE.FDBK 设置为 0b0，将反馈信号直接输入 PRNG 中。

步骤 7. 选择 PRNG64 的移位次数：将 TRNG_MODE.CNT 设置为 0b100，选择移位 64 次。

步骤 8. 读取真随机数：软件在查询到 TRNG_CR.TRNG_RUN 变为“0”后，通过读取 TRNG_DATA0 和 TRNG_DATA1，得到 64-bit 真随机数。

如果需要继续生成新的真随机数，请跳转到步骤 2，直到满足要求。

步骤 9. 完成真随机数的生成后，推荐选择关闭随机源电路，节省功耗：将 TRNG_CR.TRNGCIR_EN 设置为 0b0，关闭随机源电路。

27.4 寄存器

27.4.1 寄存器总表

基地址：0x40008C00

表 27-1 真随机数发生器 (TRNG) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	TRNG_CR	控制寄存器	32
0x04	TRNG_MODE	模式寄存器	32
0x0C	TRNG_DATA0	数据寄存器 0	32
0x10	TRNG_DATA1	数据寄存器 1	32

27.4.2 控制寄存器 (TRNG_CR)

Offset	Bit Position																																
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																														RW		TRNG_RUN
Access																															RW		TRNGCIR_EN

位/位域	标记	功能描述	读写
31:2	Reserved	-	-
1	TRNG_RUN	真随机数生成控制 软件写入“1”，开始产生新的 64bits 随机数；运行完毕后，硬件清零 ● 0b0：随机数产生完成 ● 0b1：写 1 启动随机数产生，读 1 表示随机数正在产生	RW
0	TRNGCIR_EN	随机源电路使能 ● 0b0：关闭随机源 ● 0b1：打开随机源	RW

27.4.3 模式寄存器 (TRNG_MODE)

Offset	Bit Position																															
0x04	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000010																															
Name	Reserved																								CNT		FDBK	LOAD				
Access																											RW	RW	RW			

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4:2	CNT	64-bit TRNG 的反馈移位次数 ● 0b000: 移位 0 次 (即输出随机源的采样值) ● 0b001: 移位 8 次 ● 0b010: 移位 16 次 ● 0b011: 移位 32 次 ● 0b100: 移位 64 次 ● 0b101: 移位 128 次 ● 0b110: 移位 256 次 ● 0b111: 禁止设置	RW
1	FDBK	在移位操作时, 64-bit TRNG 的反馈信号是否与随机源进行异或操作 ● 0b0: 不进行异或操作 ● 0b1: 进行异或操作	RW
0	LOAD	在产生新的随机数时, 64-bit TRNG 是否从随机源获得新的初始值 ● 0b0: 不装载新的初始值 (产生伪随机数) ● 0b1: 装载新的初始值 (产生真随机数)	RW

27.4.4 数据寄存器 0 (TRNG_DATA0)

Offset	Bit Position																															
0x0C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	DATA0																															
Access	RO																															

位/位域	标记	功能描述	读写
31:0	DATA0	低 32 位的随机数	RO

27.4.5 数据寄存器 1 (TRNG_DATA1)

Offset	Bit Position																															
0x10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Name	DATA1																															
Access	RO																															

位/位域	标记	功能描述	读写
31:0	DATA1	高 32 位的随机数	RO

28 高级加密标准模块 (AES)

28.1 简介

AES (The Advanced Encryption Standard) 是美国国家标准技术研究所 (NIST) 在 2000 年 10 月 2 日正式宣布的新的数据加密标准。

AES 的分组长度固定为 128 位，而密钥长度支持 128、192 和 256 位。对于加密来说，其输入是一个明文分组和一个密钥，输出是一个密文分组；对解密而言，输入是一个密文分组和一个密钥，而输出是一个明文分组。此过程如下所示：

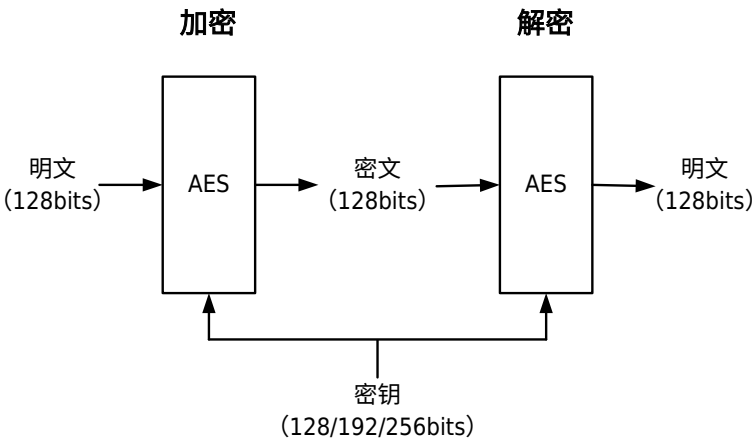


图 28-1 AES 的加解密示意图

AES 算法处理的基本单位是字节，128 位信息被分成 16 个字节，按顺序复制到一个 4×4 的矩阵中，称为状态 (state)，AES 的所有变换都是基于状态矩阵的变换，该矩阵上保存着计算的中间结果。

AES 是一个密钥迭代分组密码，包含了轮变换对状态的重复作用。AES 的轮变换由四个操作组成：SubBytes、ShiftRows、MixColumns、AddRoundKey。其中，SubBytes 包括求每个字节在 GF (2⁸) 中的模逆元和一个仿射变换；ShiftRows 是一个字节换位，它将状态中的行按照不同的偏移量进行循环移位；MixColumns 对状态各列进行线性变换；AddRoundKey，状态中的各字节与轮密钥进行逐位异或操作。AES 的加密流程如下所示：

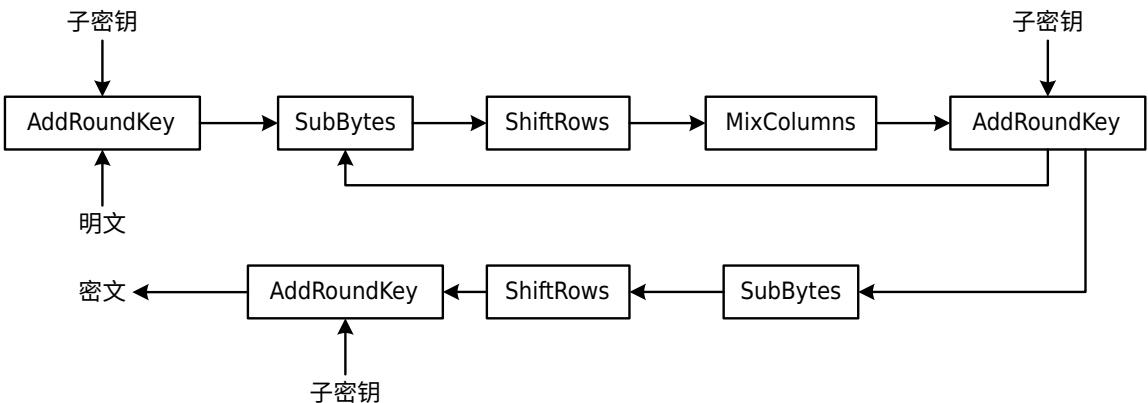


图 28-2 AES 的加密流程图

图中所用子密钥需要由初始密钥扩展而来，且密钥的扩展过程和加密过程是同步进行的。

由于明文固定为 128 位，加密过程运行的轮数就取决于密钥的长度。比如，密钥为 128 位时，运行轮数为 10 轮；密钥为 192 位时，运行轮数为 12 轮；密钥为 256 位时，运行轮数为 14 轮。除了最后一轮缺少 MixColumns 变换，其余各轮均进行完整的轮变换操作。

解密流程与加密流程有所区别，首先必须完成所有密钥的扩展，解密过程从扩展的最后一轮子密钥往回使用；然后是轮变换的四个操作变成了相应的逆运算：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey。InvSubBytes 中的模逆运算仍然保持，但仿射变换改为逆变换；InvShiftRows 和 InvMixColumns 变成相应的逆变换；AddRoundKey 保持不变。

直接解密流程的轮变换对四个操作的调用顺序为：InvShiftRows、InvSubBytes、AddRoundKey、InvMixColumns，与加密流程的调用顺序不一致，但使用的密钥与加密流程一致；等价解密流程的轮变换对四个操作的调用顺序为：InvSubBytes、InvShiftRows、InvMixColumns、AddRoundKey，与加密流程的调用顺序完全一致，只是每一轮的子密钥需要进行 InvMixColumns 运算。

关于详细的算法表述，可以参见标准《FIPS PUB 197》。

28.2 主要特性

- 执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合《FIPS PUB 197》对算法原理的描述
- 支持 128、192 和 256 位密钥

28.3 功能说明

28.3.1 异常机制

- 只支持 32 位访问，其它位宽的访问会导致系统异常，进入硬件异常中断。
- 访问 AES 模块的偏移地址大于等于 0x80 的地址，会导致系统异常，进入硬件异常中断。

28.3.2 运行时间说明

本模块从启动一次运算（AES_CR.START 写入 1）到该次运算结束（AES_CR.START 恢复到 0）所需时间如下表所示：

表 28-1 AES 加解密运行时间

操作	128 位密钥	192 位密钥	256 位密钥
加密	220 cycles	260 cycles	300 cycles
解密	290 cycles	332 cycles	398 cycles

28.4 操作示例

本模块共有两个功能：加密、解密。对两个功能的操作有一些共同的特点，下面先介绍其共同点，再分别介绍每个功能的标准操作流程。

28.4.1 加解密操作共同点

1. 在 AES 加解密过程中，数据寄存器会改变，如果下次运算的被操作数据就是本次运算的结果，就无需重新写入数据。
2. 支持 128、192 和 256 位密钥，128 位密钥写入偏移地址 0x20~0x2C，192 位密钥写入偏移地址 0x20~0x34，256 位密钥写入偏移地址 0x20~0x3C。
3. 判断模块运算结束的方法：不断读取 AES_CR.START，如果其值变为 0，则表示运算结束。

28.4.2 加密操作流程

操作步骤

- 步骤 1. 将待加密的 128 位数据写入数据寄存器（AES_DATA）中。
- 步骤 2. 将加密密钥写入密钥寄存器（AES_KEY）中。
- 步骤 3. 根据密钥长度设置 AES_CR.KEYSIZE。
- 步骤 4. 将 AES_CR.MODE 设置为 0，启动加密模式。

- 步骤 5. 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。
步骤 3~5 可以同时进行。
- 步骤 6. 查询 AES_CR.START，等待 AES_CR.START 的值恢复为 0，模块运算结束。
- 步骤 7. 读取数据寄存器 (AES_DATA)，获得 128 位密文。

28.4.3 解密操作流程

操作步骤

- 步骤 1. 将待解密的 128 位数据写入数据寄存器 (AES_DATA) 中。
- 步骤 2. 将解密密钥写入密钥寄存器 (AES_KEY) 中。
- 步骤 3. 根据密钥长度设置 AES_CR.KEYSIZE。
- 步骤 4. 将 AES_CR.MODE 设置为 1，启动解密模式。
- 步骤 5. 向控制寄存器中的 AES_CR.START 写入 1，启动模块进行运算。
步骤 3~5 可以同时进行。
- 步骤 6. 查询 AES_CR.START，等待 AES_CR.START 的值恢复为 0，模块运算结束。
- 步骤 7. 读取数据寄存器 (AES_DATA)，获得 128 位明文。

28.4.4 数据示例

128 位密钥

128 位明文: 0xFFEEDDCCBBAA99887766554433221100
128 位密钥: 0x0F0E0D0C0B0A09080706050403020100
128 位密文: 0x5AC5B47080B7CDD830047B6AD8E0C469

表 28-2 128 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
AES_KEY0	0x03020100	AES_DATA0	0x33221100
AES_KEY1	0x07060504	AES_DATA1	0x77665544
AES_KEY2	0x0B0A0908	AES_DATA2	0xBBAA9988
AES_KEY3	0x0F0E0D0C	AES_DATA3	0xFFEEDDCC
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
AES_KEY0	0x03020100	AES_DATA0	0xD8E0C469
AES_KEY1	0x07060504	AES_DATA1	0x30047B6A
AES_KEY2	0x0B0A0908	AES_DATA2	0x80B7CDD8
AES_KEY3	0x0F0E0D0C	AES_DATA3	0x5AC5B470

192 位密钥

128 位明文: 0xFFEEDDCCBBAA99887766554433221100
192 位密钥: 0x17161514131211100F0E0D0C0B0A09080706050403020100
128 位密文: 0x91710DECA070AF6EE0DF4C86A47CA9DD

表 28-3 192 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
AES_KEY0	0x03020100	AES_DATA0	0x33221100
AES_KEY1	0x07060504	AES_DATA1	0x77665544
AES_KEY2	0x0B0A0908	AES_DATA2	0xBBAA9988
AES_KEY3	0x0F0E0D0C	AES_DATA3	0xFFEEDDCC
AES_KEY4	0x13121110		
AES_KEY5	0x17161514		
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
AES_KEY0	0x03020100	AES_DATA0	0xA47CA9DD
AES_KEY1	0x07060504	AES_DATA1	0xE0DF4C86
AES_KEY2	0x0B0A0908	AES_DATA2	0xA070AF6E
AES_KEY3	0x0F0E0D0C	AES_DATA3	0x91710DEC
AES_KEY4	0x13121110		
AES_KEY5	0x17161514		

256 位密钥

128 位明文：0xFFEEDDCCBBAA99887766554433221100

256 位密钥：0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A09080706050403020100

128 位密文：0x8960494B9049FCAEBF456751CAB7A28E

表 28-4 256 位操作寄存器示例

加密前			
寄存器	值 (密钥)	寄存器	值 (明文)
AES_KEY0	0x03020100	AES_DATA0	0x33221100
AES_KEY1	0x07060504	AES_DATA1	0x77665544
AES_KEY2	0x0B0A0908	AES_DATA2	0xBBAA9988
AES_KEY3	0x0F0E0D0C	AES_DATA3	0xFFEEDDCC
AES_KEY4	0x13121110		
AES_KEY5	0x17161514		
AES_KEY6	0x1B1A1918		
AES_KEY7	0x1F1E1D1C		
加密后			
寄存器	值 (密钥)	寄存器	值 (密文)
AES_KEY0	0x03020100	AES_DATA0	0xCAB7A28E
AES_KEY1	0x07060504	AES_DATA1	0xBF456751
AES_KEY2	0x0B0A0908	AES_DATA2	0x9049FCAE
AES_KEY3	0x0F0E0D0C	AES_DATA3	0x8960494B
AES_KEY4	0x13121110		
AES_KEY5	0x17161514		
AES_KEY6	0x1B1A1918		
AES_KEY7	0x1F1E1D1C		

28.5 寄存器

28.5.1 寄存器总表

基地址：0x40021400

表 28-5 高级加密标准模块 (AES) 寄存器偏移地址

偏移地址	寄存器	描述	位宽
0x00	AES_CR	控制寄存器	32
0x10+0x4*n	AES_DATAn	数据寄存器 n (n=0~3)	32
0x20+0x4*n	AES_KEYn	密钥寄存器 n (n=0~7)	32
0x40	AES_CRDM	控制寄存器映射地址	32

28.5.2 控制寄存器 (AES_CR)

Offset	Bit Position																																	
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reset	0x00000000																																	
Name	Reserved																										KEYSIZE		Reserved		MODE		START	
Access																													RW		RW		RW	

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4:3	KEYSIZE	密钥长度 0b00: 密钥长度为 128 位 0b01: 密钥长度为 192 位 0b10: 密钥长度为 256 位 0b11: 密钥长度为 128 位	RW
2	Reserved	-	-
1	MODE	运算模式 0b0: 加密运算 0b1: 解密运算	RW
0	START	运算启动控制 0b0: 运算结束或未启动 0b1: 启动运算	RW

说明

- AES_CR.START 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
- 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。
- AES_CR 寄存器位于 AES 基地址 0 偏移地址处，基地址的 0x40 偏移地址处为 AES_CR 寄存器的映射地址，其访问和操作权限与 AES_CR 寄存器相同。

28.5.3 数据寄存器 (AES_DATAn) (n=0~3)

偏移地址 (Offset) : 0x10+0x4*n

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	DATA																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	DATA	明文/密文数据 存放 AES 算法的 128 比特明文/密文	RW

数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128 位明文	128 位密文	128 位密文	128 位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应填入数据如下（以数据 0xFFEEDDCCBBAA99887766554433221100 为例）：

表 28-11 寄存器填入数据

AES_DATA0	AES_DATA1	AES_DATA2	AES_DATA3
0x33221100	0x77665544	0xBBAA9988	0xFFEEDDCC

 **说明**

- 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则对本寄存器的读取将得到全 0。

28.5.4 密钥寄存器 (AES_KEYn) (n=0~7)

偏移地址 (Offset) : 0x20+0x4*n

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	KEY																															
Access	RW																															

位/位域	标记	功能描述	读写
31:0	KEY	AES 密钥 存放 AES 算法的 128 (KEY0~KEY3)、192 (KEY0~KEY5)、256 (KEY0~KEY7) 位密钥	RW

密钥寄存器由 8 个 32 位的寄存器组成，存放输入的初始密钥。写操作时需要分别对 8 个 32 位的寄存器进行操作。寄存器对应填入数据分别参考下述举例。

128 位密钥举例：0x0F0E0D0C_0B0A0908_07060504_03020100

表 28-14 寄存器填入数据-128 位密钥

AES_KEY0	AES_KEY1	AES_KEY2	AES_KEY3
0x03020100	0x07060504	0x0B0A0908	0x0F0E0D0C

192 位密钥举例：0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

表 28-15 寄存器填入数据-192 位密钥

AES_KEY0	AES_KEY1	AES_KEY2	AES_KEY3	AES_KEY4	AES_KEY5
0x03020100	0x07060504	0x0B0A0908	0x0F0E0D0C	0x13121110	0x17161514

256 位密钥举例：

0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

表 28-16 寄存器填入数据-256 位密钥

AES_KEY0	AES_KEY1	AES_KEY2	AES_KEY3	AES_KEY4	AES_KEY5	AES_KEY6	AES_KEY7
0x03020100	0x07060504	0x0B0A0908	0x0F0E0D0C	0x13121110	0x17161514	0x1B1A1918	0x1F1E1D1C

说明

- 对于本寄存器的写入只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略对本寄存器的写操作。
- 对于本寄存器的读取只能在本模块没有处于运算状态时（即 AES_CR.START=0 时）才能进行，否则对本寄存器的读取将得到全 0。

28.5.5 控制寄存器映射地址 (AES_CRDM)

Offset	Bit Position																															
0x40	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved																										KEYSIZE		Reserved	MODE	START	
Access																											RW		RW	RW		

位/位域	标记	功能描述	读写
31:5	Reserved	-	-
4:3	KEYSIZE	密钥长度 0b00: 密钥长度为 128 位 0b01: 密钥长度为 192 位 0b10: 密钥长度为 256 位 0b11: 密钥长度为 128 位	RW
2	Reserved	-	-
1	MODE	运算模式 0b0: 加密运算 0b1: 解密运算	RW
0	START	运算启动控制 0b0: 运算结束或未启动 0b1: 启动运算	RW

 **说明**

- AES_CR.START 位的操作方法是：软件对本位写入 1 后，本模块将启动运行，本次运行结束后本模块硬件会自动将本位清 0，软件查询到本位为 0 即表示本次运行完成。
- 对本寄存器的写入操作只能在本模块不处于运算状态时（即 AES_CR.START=0 时）才能进行，否则硬件将自动忽略写操作。读操作则不受此限制。

29 SWD 调试接口

29.1 简介

本系列产品使用 ARM Cortex-M0+内核，该处理器内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指（指令断点）或访问数据（数据断点）时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当本系列微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。



说明

SWD 能把 DeepSleep 唤醒，请注意不要以 SWD 接口断开作为 DeepSleep 的状态。

29.2 功能说明

29.2.1 调试系统框图

本产品使用了 ARM Cortex-M0+内核，包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。当调试主机与 MCU 相连并进行调试时，将使用调试功能。

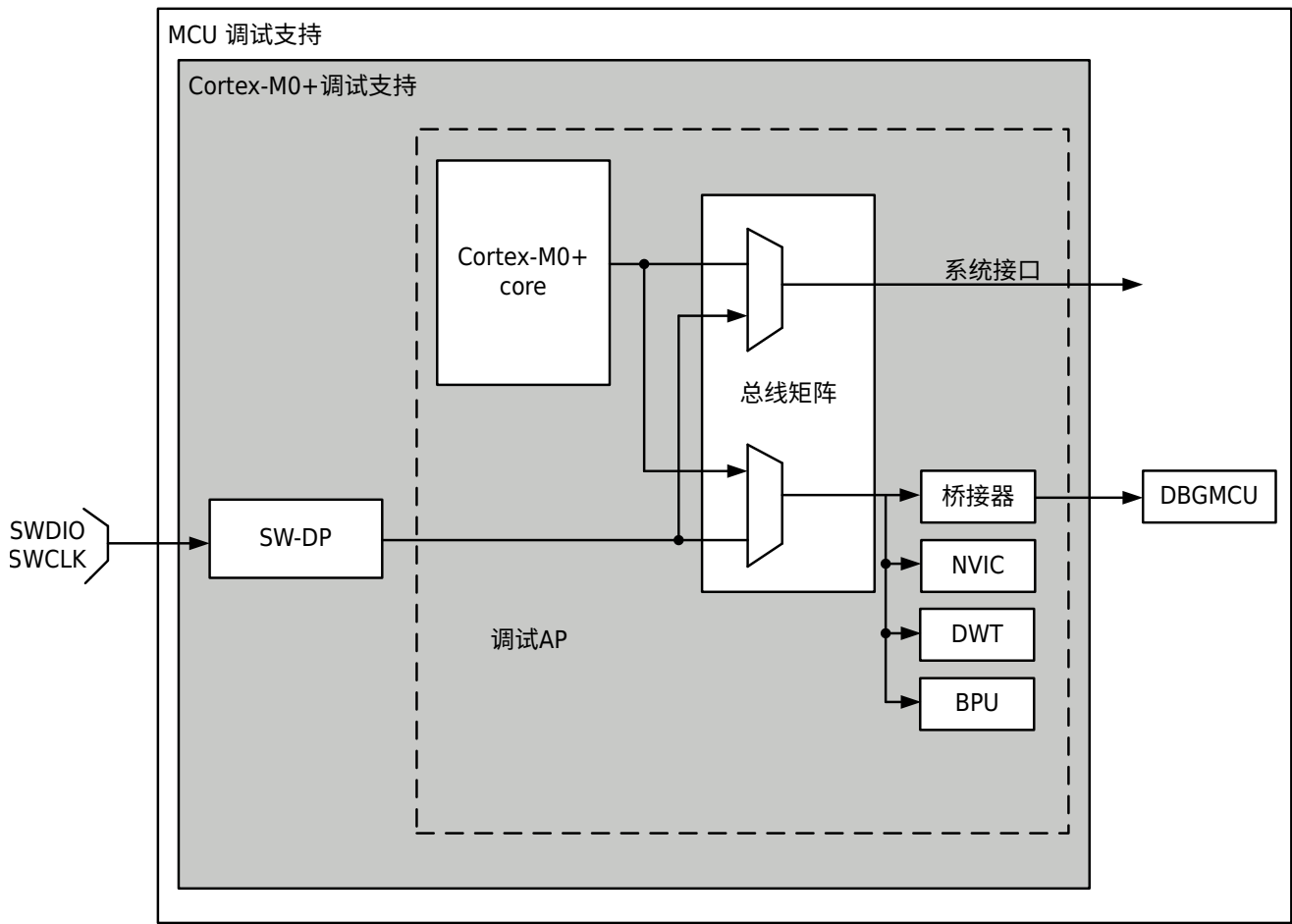


图 29-1 调试支持框图

Cortex-M0+内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex-M0+内核提供集成片上调试支持，包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发



说明

有关 ARM® Cortex-M0+内核支持的调试功能的详细信息，请参见《Cortex-M0+技术参考手册》。

29.2.2 ARM®参考文档

- Cortex-M0+技术参考手册（TRM）
可从 developer.arm.com 获取。
- ARM®调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册
- ARMv6-M Architecture Reference Manual

29.2.3 调试端口引脚

SWD 端口引脚

本系列的 SWD 接口，需要用到 2 个引脚，如下表所示。

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入/输出	PA13

SW-DP 引脚分配

如果烧录程序时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 PA13/PA14 引脚均被初始化为可被调试器使用的专用引脚。用户可设置 SYSCTRL1.SWD_USE_IO 寄存器来禁止 SWD 引脚的调试功能，SWD 引脚将被释放以用作普通 GPIO。SWD 引脚的配置与功能小结如下表所示：

【加密芯片】选项	SWD_USE_IO 配置	PA13/PA14 功能
加密	0	-
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

SWD 引脚上的内部上拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

29.2.4 SWD 端口

29.2.4.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM®建议采用 100k）。这些上拉电阻可在内部配置，无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

29.2.4.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

表 29-3 主机发送的数据包请求

位	名称	说明
0	启动	必须为 0b1
1	APnDP	● 0b0：DP 访问 ● 0b1：AP 访问
2	RnW	● 0b0：写请求 ● 0b1：读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0b0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 0b1

有关 DPACC 和 APACC 寄存器的详细说明，请参见《Cortex-M0+技术参考手册》。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动。

表 29-4 目标发送的确认响应

位	名称	说明
2:0	ACK	● 0b001：FAULT ● 0b010：WAIT ● 0b100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

表 29-5 主机或目标发送的数据传输

位	名称	说明
31:0	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

29.2.4.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码，该代码符合 JEP-106 标准，此 ID 代码是默认的 ARM®代码，设置为 **0x0BC11477**（相当于 Cortex®-M0+）。

 **说明**

在目标读取此 ID 代码前，SW-DP 状态机是不工作的。

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见《Cortex-M0+技术参考手册》和《ARM® CoreSight 设计套件版本 r1p1 技术参考手册》。

29.2.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

29.2.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	RW	SELECT 寄存器的 CTRLSEL 位	寄存器	说明
0b00	读取	-	IDCODE	制造商代码设置为 Cortex®-M0+的默认 ARM®代码。 0x0BC11477 （标识 SW-DP）
0b00	写	-	ABORT	-

A[3:2]	RW	SELECT 寄存器的 CTRLSEL 位	寄存器	说明
0b01	读/写	0	DP-CTRL/STAT	目的： ● 请求系统或调试上电 ● 配置 AP 访问的传输操作 ● 控制比较和验证操作 ● 读取一些状态标志（上溢和上电确认）
0b01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续时间）
0b10	读取	-	READ RESEND	允许从已损坏的调试软件传输中恢复读取数据，无需重复执行原始 AP 传输。
0b10	写	-	SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口
0b11	读/写	-	READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 事务时提供读取 AP 请求的结果）。 此读取缓冲区捕获 AP 中的数据，显示为前一次读取的结果，无需启动新操作。

29.2.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	A[3:2]值	说明
0x0	0b00	Reserved，必须保持复位值。
0x4	0b01	DP CTRL/STAT 寄存器。用于： ● 请求系统或调试上电 ● 配置 AP 访问的传输操作 ● 控制比较和验证操作 ● 读取一些状态标志（上溢和上电确认）
0x8	0b10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 ● Bit[31:24]：APSEL，选择当前 AP(select the current AP) ● Bit[23:8]：Reserved ● Bit[7:4]：APBANKSEL，在当前 AP 上选择活动的 4 字寄存器窗口 ● Bit[3:0]：Reserved

地址	A[3:2]值	说明
0xC	0b11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）

29.2.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见《Cortex-M0+技术参考手册》。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0（VC_CORRESET）
- 使能调试停止控制和状态寄存器的位 0（C_DEBUGEN）

29.2.6 BPU（断点单元）

Cortex-M0+ BPU 实现提供 4 个断点寄存器。

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见《ARMv6-M Architecture Reference Manual》和《ARM® CoreSight 组件技术参考手册》。

29.2.7 DWT（数据观察点）

Cortex-M0+ DWT 实现提供了两个观察点寄存器组。

DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，详细描述请参见《ARMv6-M Architecture Reference Manual》。

DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器（DWT_PCSR）。此寄存器允许调试程序定期采样 PC，无需停止处理器，可提供粗略分析。详细描述请参见《ARMv6-M Architecture Reference Manual》。

Cortex-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

29.2.8 MCU 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。用于 PWM 控制电机时需要使用这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要使用这种方式。

30 器件电子签名

30.1 简介

电子签名存放在闪存存储器模块的系统存储区域，可以通过 SWD 或者 CPU 读取。它包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的微控制器。

30.2 功能说明

30.2.1 产品唯一身份标识 (UID)

唯一身份标识符典型应用场景：

- 用作序列号
- 在对内部 Flash 进行编程前将 UID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

唯一身份标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。唯一身份标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

30.3 寄存器

30.3.1 产品唯一身份标识（UID）寄存器（80 位）

基地址：0x00101E8C

Offset	Description	UID Bits (80bits)							
		7	6	5	4	3	2	1	0
0	Y Coordinate on the wafer	UID[7:0]							
1	X Coordinate on the wafer	UID[15:8]							
2	Wafer Number	UID[23:16]							
3	Lot Number	UID[31:24]							
4		UID[39:32]							
5		UID[47:40]							
6		UID[55:48]							
7		UID[63:56]							
8		UID[71:64]							
9		UID[79:72]							

30.3.2 产品型号寄存器

0x00101B40~0x00101B4F 存储了产品型号的 ASCII 码。如产品型号不足 16 字节，则以 0x00 进行填充。
例：484333324C3132364B41544200000000 所代表的产品型号为 HC32L126KATB。

30.3.3 FLASH 容量寄存器

基地址：0x00101B60

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	-																															
Name	FLASHSIZE																															
Access	RO																															

位/位域	标记	功能描述	读写
31:0	FLASHSIZE	产品内置 Flash 的容量 以 Byte 为单位 0x00020000 代表 Flash 容量为 128K Byte	RO

30.3.4 RAM 容量寄存器

基地址：0x00101B64

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	-																															
Name	RAMSIZE																															
Access	RO																															

位/位域	标记	功能描述	读写
31:0	RAMSIZE	产品内置 RAM 的容量 以 Byte 为单位 0x00004000 代表 RAM 容量为 16K Byte	RO

30.3.5 管脚数量/Flash 扇区 Size 配置寄存器

基地址：0x00101B68

Offset	Bit Position																															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	-																															
Name	PINCOUNT																FLASHSECTORSIZE															
Access																																

位/位域	标记	功能描述	读写
31:16	PINCOUNT	产品管脚的数量 以只为单位 0x0040 代表产品管脚数量为 64	RO
15:0	FLASHSECTOR RSIZE	Flash 扇区 Size 以 Byte 为单位 0x0200 代表 512Byte	RO

A 术语表

术语	全称	说明
Byte	-	字节，8 位数据
Half Word	-	半字，16 位数据
Word	-	字，32 位数据
Double Word	-	双字，64 位数据
IAP	-	指可以在用户程序运行期间对微控制器的 Flash 进行重新编程（在应用中编程）
ICP	-	指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程（在线编程）
AHB	Advanced High-performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	低速外设总线
DMA	Direct Memory Access	直接存储器访问
TIM/TMR	-	定时器

修订记录

文档版本日期	修改说明
Rev1.20 2026/08/28	1. 实时时钟（RTC）：“时间戳时间寄存器（RTC_TSTR）”章节 HOURH/HOURL 位新增说明“该位为 24 小时制，不受 RTC_CR0.AMPM 影响”； 2. 低功耗同步异步收发器（LPUART）：“多机通讯”章节优化从机地址配置描述； 3. 通用定时器（TIM0/1/2/3）：“模式 2/3 比较捕获模式-捕获输入”章节 CH0B 的捕获输入描述优化，当 MSCR.IB0S 配置为 1 时，不支持 MSCR.TS 选择 CH0B 滤波信号； 4. 模数转换器（ADC）：新增内置信号跟随器功能； 5. SWD 调试接口：SW-DP 的内部 ID 代码修改为“0x0BC11477”；
Rev1.10 2026/02/12	1. 系统控制器（SYSCTRL）：“从其它时钟切换到 XTH 示例”章节步骤 7 删除软件延迟等待时间；“XTH 控制寄存器（XTH_CR）”章节 STABLE 位删除软件延迟等待时间；
Rev1.00 2025/09/12	第一次正式发布。