

HC32F420 系列

32 位 ARM® Cortex®-M4 微控制器

参考手册

Rev1.30 2026 年 08 月

声 明

- ★ 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/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。
- ★ 任何带有“®”或“™”标识的图形或字样是 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。
- ★ 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。

©2026 小华半导体有限公司 保留所有权利

目 录

声 明.....	2
目 录.....	3
表索引.....	18
图索引.....	21
简介 (Overview)	26
1 系统结构	27
1.1 概述	27
1.2 系统构架图.....	27
1.3 存储器及外设地址分配	28
2 工作模式	29
2.1 工作模式概述	29
2.2 工作模式切换	29
2.3 运行模式 (Active Mode)	31
2.4 休眠模式 (Sleep Mode)	31
2.5 深度休眠模式 (Deep Sleep Mode)	31
2.6 功能模块工作状态	32
2.7 寄存器描述.....	33
2.7.1 寄存器列表	33
2.7.2 系统控制寄存器 (SCB_SCR)	33
3 系统控制器 (SYSCTRL)	34
3.1 系统时钟介绍	34
3.1.1 时钟架构图	35
3.1.2 内部高速 RC 时钟 RCH	35
3.1.3 内部低速 RC 时钟 RCL.....	35
3.1.4 外部晶振时钟 XTH	36
3.1.5 锁相环时钟 PLL.....	36
3.1.6 时钟启动过程	37
3.2 系统时钟切换	38
3.2.1 标准的时钟切换流程	38
3.2.2 RCHCLKD 不同输出频率间切换流程	40
3.2.3 从其它时钟切换到 RCL 示例.....	40
3.2.4 从其它时钟切换到 RCHCLKD 示例.....	40
3.2.5 PLL (参考时钟为 RCHCLKD) 与 RCHCLKD 相互切换示例	41

3.3	片内外设时钟控制	42
3.4	中断唤醒控制	42
3.4.1	从深度休眠模式唤醒后执行中断服务程序的方法	42
3.4.2	从深度休眠模式唤醒后不执行中断服务程序的方法	42
3.4.3	使用退出休眠特性	43
3.5	寄存器描述	44
3.5.1	寄存器列表	44
3.5.2	系统控制寄存器 0 (SYSCTRL_CR0)	45
3.5.3	系统控制寄存器 1 (SYSCTRL_CR1)	46
3.5.4	时钟使能控制寄存器 2 (SYSCTRL_CR2)	47
3.5.5	序列控制寄存器 3 (SYSCTRL_CR3)	48
3.5.6	RCH 控制寄存器 (SYSCTRL_RCHCR)	48
3.5.7	RCL 控制寄存器 (SYSCTRL_RCLCR)	49
3.5.8	XTH 控制寄存器 (SYSCTRL_XTHCR)	50
3.5.9	PLL 控制寄存器 (SYSCTRL_PLLCR)	51
3.5.10	RCH 校准值寄存器 (SYSCTRL_RCHTRIM)	52
3.5.11	RCL 校准值寄存器 (SYSCTRL_RCLTRIM)	52
3.5.12	时钟状态寄存器 (SYSCTRL_STATE)	53
3.5.13	APB0 外设时钟控制寄存器 (SYSCTRL_APBEN0)	54
3.5.14	APB1 外设时钟控制寄存器 (SYSCTRL_APBEN1)	55
3.5.15	AHB0 外设时钟控制寄存器 (SYSCTRL_AHBEN0)	56
3.5.16	AHB1 外设时钟控制寄存器 (SYSCTRL_AHBEN1)	56
3.5.17	定时器调试暂停控制寄存器 (SYSCTRL_DEBUG)	57
3.5.18	GPIO 辅助控制寄存器 0 (SYSCTRL_IOAUX0)	58
3.5.19	GPIO 辅助控制寄存器 1 (SYSCTRL_IOAUX1)	59
3.5.20	GPIO 辅助控制寄存器 2 (SYSCTRL_IOAUX2)	59
4	复位控制器 (RESET)	60
4.1	复位控制器介绍	60
4.1.1	上电下电复位 POR PDR	61
4.1.2	外部复位管脚复位	61
4.1.3	WDT 复位	61
4.1.4	LVD 低电压复位	61
4.1.5	Cortex-M4 SYSRESETREQ 复位	61
4.1.6	Cortex-M4 LOCKUP 复位	61
4.2	寄存器描述	62

4.2.1	寄存器列表	62
4.2.2	APB0 外设复位控制寄存器 (SYSCTRL_APBIRST0)	63
4.2.3	APB1 外设复位控制寄存器 (SYSCTRL_APBIRST1)	64
4.2.4	AHB0 外设复位控制寄存器 (SYSCTRL_AHBIRST0)	65
4.2.5	AHB1 外设复位控制寄存器 (SYSCTRL_AHBIRST1)	65
4.2.6	复位标志寄存器 (SYSCTRL_RSTFLAG)	66
5	中断控制器 (NVIC)	67
5.1	概述	67
5.2	中断优先级	68
5.3	中断向量表	69
5.4	中断输入和挂起行为	70
5.5	中断等待	71
5.6	中断源	72
5.7	中断结构	74
5.8	软件基本操作	74
5.8.1	外部中断使能	74
5.8.2	NVIC 中断使能和清除使能	74
5.8.3	NVIC 中断挂起和清除挂起	74
5.8.4	NVIC 中断优先级	75
5.8.5	NVIC 中断屏蔽	75
5.9	寄存器描述	76
5.9.1	寄存器列表	76
5.9.2	中断使能设置寄存器 0 (NVIC_ISER0)	77
5.9.3	中断使能设置寄存器 1 (NVIC_ISER1)	77
5.9.4	中断使能清除寄存器 0 (NVIC_ICER0)	78
5.9.5	中断使能清除寄存器 1 (NVIC_ICER1)	78
5.9.6	中断挂起设置寄存器 0 (NVIC_ISPR0)	79
5.9.7	中断挂起设置寄存器 1 (NVIC_ISPR1)	79
5.9.8	中断挂起清除寄存器 0 (NVIC_ICPR0)	80
5.9.9	中断挂起清除寄存器 1 (NVIC_ICPR1)	80
5.9.10	中断优先级寄存器 (NVIC_IPRx) (x=0~15)	81
5.9.11	中断屏蔽寄存器 (PRIMASK)	81
6	通用输入输出端口控制器 (GPIO)	82
6.1	简介	82
6.2	主要特性	82

6.3	功能描述	83
6.3.1	端口电路框图	83
6.3.2	端口模式配置	83
6.3.3	端口复位状态	84
6.3.4	端口模拟功能	84
6.3.5	端口通用输入输出功能	85
6.3.6	端口复用功能	88
6.3.7	端口时钟输出	91
6.3.8	端口外部中断	92
6.4	寄存器描述	93
6.4.1	寄存器列表	93
6.4.2	端口数模配置寄存器 (GPIO_ADS)	94
6.4.3	端口输入输出方向配置寄存器 (GPIO_DIR)	94
6.4.4	端口输出类型寄存器 (GPIO_OD)	95
6.4.5	端口输出驱动能力寄存器 (GPIO_DR)	95
6.4.6	端口上拉下拉寄存器 (GPIO_PUPD)	96
6.4.7	端口数据输入寄存器 (GPIO_IN)	96
6.4.8	端口数据输出寄存器 (GPIO_OUT)	96
6.4.9	端口复位寄存器 (GPIO_BRR)	97
6.4.10	端口置位复位寄存器 (GPIO_BSRR)	97
6.4.11	端口复用功能低位寄存器 (GPIO_AFRL)	98
6.4.12	端口复用功能高位寄存器 (GPIO_AFRH)	98
6.4.13	端口高电平中断使能配置寄存器 (GPIO_HIE)	99
6.4.14	端口低电平中断使能配置寄存器 (GPIO_LIE)	99
6.4.15	端口上升沿中断使能配置寄存器 (GPIO_RIE)	99
6.4.16	端口下降沿中断使能配置寄存器 (GPIO_FIE)	100
6.4.17	端口中断状态寄存器 (GPIO_IFR)	100
6.4.18	端口中断清除寄存器 (GPIO_ICR)	100
7	I2C 总线 (I2C)	101
7.1	概述	101
7.2	主要特性	101
7.3	协议描述	102
7.3.1	I2C 总线上数据传输	102
7.3.2	I2C 总线上的应答	103
7.3.3	I2C 总线上的仲裁	104

7.4	功能描述	105
7.4.1	功能框图	105
7.4.2	串行时钟发生器	106
7.4.3	输入滤波器	106
7.4.4	地址比较器	106
7.4.5	应答标志位	106
7.4.6	中断产生器	107
7.4.7	工作模式	107
7.4.8	状态码表述	113
7.5	编程示例	115
7.5.1	主机发送示例	115
7.5.2	主机接收示例	115
7.5.3	从机接收示例	116
7.5.4	从机发送示例	117
7.6	寄存器描述	118
7.6.1	寄存器列表	118
7.6.2	波特率发生器使能寄存器 (I2C_BRREN)	119
7.6.3	波特率计数器配置寄存器 (I2C_BRR)	119
7.6.4	配置寄存器 (I2C_CR)	120
7.6.5	数据寄存器 (I2C_DR)	120
7.6.6	状态寄存器 (I2C_STAT)	121
7.6.7	从机地址 0 寄存器 (I2C_ADDR0)	121
7.6.8	从机地址 1 寄存器 (I2C_ADDR1)	121
7.6.9	从机地址 2 寄存器 (I2C_ADDR2)	122
7.6.10	从机地址匹配寄存器 (I2C_MATCH)	122
8	串行外设接口 (SPI)	123
8.1	概述	123
8.2	主要特性	123
8.3	功能描述	123
8.3.1	功能框图	123
8.3.2	通信格式和时序	124
8.3.3	从机选择管脚配置	125
8.3.4	全双工通信	126
8.3.5	单线半双工通信	126
8.3.6	单工通信	127

8.3.7	多机通信	128
8.3.8	主机模式延后采样	129
8.3.9	状态标志和错误标志	130
8.3.10	中断的产生和清除	133
8.4	编程示例	134
8.4.1	主机全双工收发数据示例	134
8.4.2	从机全双工收发数据示例	134
8.4.3	单线半双工发送接收数据示例	135
8.5	寄存器描述	137
8.5.1	寄存器列表	137
8.5.2	SPI 控制寄存器 0 (SPI_CR0)	138
8.5.3	SPI 控制寄存器 1 (SPI_CR1)	139
8.5.4	SPI 单线半双工输出使能寄存器 (SPI_HDOE)	140
8.5.5	SPI 内部从机选择寄存器 (SPI_SSI)	140
8.5.6	SPI 状态寄存器 (SPI_SR)	141
8.5.7	SPI 中断清除寄存器 (SPI_ICR)	142
8.5.8	SPI 数据寄存器 (SPI_DR)	142
9	时钟校准模块 (CTRIM)	143
9.1	概述	143
9.2	主要特性	143
9.3	功能描述	143
9.3.1	RCH 时钟校准模式	143
9.3.2	RCL 时钟校准模式	144
9.3.3	低功耗定时器模式	145
9.4	软件操作流程	146
9.4.1	RCH 时钟自动校准示例	146
9.4.2	RCL 时钟自动校准示例	147
9.5	寄存器描述	148
9.5.1	寄存器列表	148
9.5.2	自动装载寄存器 (CTRIM_ARR)	149
9.5.3	误差计数器寄存器 (CTRIM_CNT)	149
9.5.4	控制寄存器 0 (CTRIM_CR0)	150
9.5.5	控制寄存器 1 (CTRIM_CR1)	151
9.5.6	中断使能寄存器 (CTRIM_IER)	152
9.5.7	中断和状态寄存器 (CTRIM_ISR)	153

9.5.8	中断状态清除寄存器 (CTRIM_ICR)	153
9.5.9	误差结果寄存器 (CTRIM_FCAP)	154
9.5.10	TRIM 结果寄存器 (CTRIM_TVAL)	154
9.5.11	误差上限寄存器 (CTRIM_FLIM)	154
10	FLASH 控制器 (FLASH)	155
10.1	概述	155
10.2	容量划分	155
10.3	读等待周期	155
10.4	FLASH 操作 (读、写、擦)	156
10.4.1	页擦除 (Sector Erase)	156
10.4.2	全片擦除 (Chip Erase)	156
10.4.3	写操作 (Program)	157
10.4.4	读操作 (Read)	158
10.4.5	Cache 功能	158
10.5	FLASH 安全保护	159
10.5.1	页面擦写保护	159
10.5.2	PC 地址擦写保护	159
10.5.3	寄存器写保护	159
10.5.4	数据读出保护	160
10.6	寄存器描述	161
10.6.1	寄存器列表	161
10.6.2	控制寄存器 (FLASH_CR)	162
10.6.3	等待周期设置 (FLASH_WAIT)	162
10.6.4	LOCK 状态寄存器 (FLASH_LOCKSTATE)	162
10.6.5	CACHE 控制寄存器 (FLASH_CACHE)	163
10.6.6	中断使能寄存器 (FLASH_IER)	163
10.6.7	中断标志寄存器 (FLASH_IFR)	163
10.6.8	中断标志清除寄存器 (FLASH_ICR)	164
10.6.9	Bypass 序列寄存器 (FLASH_BYPASS)	164
10.6.10	Sector 擦写保护寄存器 (FLASH_SLOCK0)	164
10.6.11	Sector 擦写保护寄存器 (FLASH_SLOCK1)	165
11	RAM 控制器 (RAM)	166
11.1	概述	166
11.2	功能描述	166
11.2.1	RAM 地址范围	166

12 DMA 控制器 (DMAC)	167
12.1 DMAC 简介	167
12.2 DMAC 主要特性	167
12.3 DMA 功能框图	168
12.4 功能描述	169
12.4.1 DMAC 传输模式概述	169
12.4.2 DMA 软件 Block 传输模式	169
12.4.3 DMA 软件 Burst 传输模式	170
12.4.4 DMA 硬件 Block 传输模式	171
12.4.5 DMA 硬件 Burst 传输模式	172
12.4.6 DMA 传输暂停	173
12.4.7 DMA 其它配置	173
12.4.8 DMA 中断	174
12.5 寄存器描述	175
12.5.1 寄存器列表	175
12.5.2 所有通道配置寄存器 (DMAC_CONF)	176
12.5.3 通道 x 配置 A 寄存器 (DMAC_CONFAx)	177
12.5.4 通道 x 配置 B 寄存器 (DMAC_CONFBx)	179
12.5.5 通道 x 传输源地址寄存器 (DMAC_SRCADR _x)	181
12.5.6 通道 x 目的地址寄存器 (DMAC_DSTADR _x)	181
13 复合定时器 (CTIM)	182
13.1 概述	182
13.1.1 BTIM 主要特性	183
13.1.2 GTIM 主要特性	183
13.2 BTIM 功能描述	184
13.2.1 功能框图	184
13.2.2 滤波单元	184
13.2.3 计数单元	185
13.2.4 定时器模式	186
13.2.5 计数器模式	186
13.2.6 触发启动模式	187
13.2.7 门控模式	188
13.2.8 定时器级联	189
13.2.9 中断	189
13.3 GTIM 功能描述	190

13.3.1	功能框图	190
13.3.2	滤波单元	190
13.3.3	计数单元	191
13.3.4	定时器模式	192
13.3.5	计数器模式	192
13.3.6	触发启动模式	193
13.3.7	门控模式	194
13.3.8	比较捕获功能	195
13.3.9	DMA 触发	196
13.3.10	中断	197
13.3.11	片内外设互联	197
13.4	BTIM 寄存器描述	198
13.4.1	寄存器列表	198
13.4.2	重载寄存器 (BTIMx_ARR) (x=0~8)	199
13.4.3	计数寄存器 (BTIMx_CNT) (x=0~8)	199
13.4.4	控制寄存器 (BTIMx_CR) (x=0~8)	200
13.4.5	中断使能 (BTIMx_IER) (x=0~8)	201
13.4.6	中断标志寄存器 (BTIMx_IFR) (x=0~8)	201
13.4.7	中断标志清除寄存器 (BTIMx_ICR) (x=0~8)	202
13.4.8	复合中断标志寄存器 (BTIM_AIFR)	202
13.4.9	复合中断标志清除寄存器 (BTIM_AICR)	203
13.5	GTIM 寄存器描述	204
13.5.1	寄存器列表	204
13.5.2	重载寄存器 (GTIM_ARR)	205
13.5.3	计数寄存器 (GTIM_CNT)	205
13.5.4	控制寄存器 1 (GTIM_CR1)	206
13.5.5	控制寄存器 0 (GTIM_CR0)	207
13.5.6	中断使能控制寄存器 (GTIM_IER)	208
13.5.7	中断标志寄存器 (GTIM_IFR)	209
13.5.8	中断标志清除寄存器 (GTIM_ICR)	210
13.5.9	比较捕获控制寄存器 (GTIM_CMMR)	211
13.5.10	比较捕获寄存器 (GTIM_CCRy) (y=0,1,2,3)	211
14	高级定时器 (ATIM)	212
14.1	概述	212
14.2	主要特性	212

14.2.1	ATIM3 主要特性	212
14.2.2	ATIM0 主要特性	214
14.3	功能描述	215
14.3.1	定时器时钟	215
14.3.2	定时计数器	215
14.3.3	定时器预分频	215
14.3.4	模式 0 计数定时器功能	216
14.3.5	模式 1 脉宽测量 PWC.....	219
14.3.6	模式 2/3 比较捕获模式.....	222
14.3.7	模式 2/3 从模式	243
14.3.8	正交编码计数功能	245
14.3.9	Timer 触发 ADC.....	247
14.3.10	刹车控制	247
14.3.11	中断	248
14.3.12	定时器互联	248
14.3.13	输入互联	248
14.4	寄存器描述.....	249
14.4.1	寄存器列表	249
14.4.2	模式 0 寄存器描述	250
14.4.3	模式 1 寄存器描述	253
14.4.4	模式 2,3 寄存器描述	259
15	独立看门狗 (IWDG)	278
15.1	概述	278
15.2	主要特性	278
15.3	功能描述	278
15.3.1	功能框图	278
15.3.2	窗口看门狗模式.....	278
15.3.3	独立看门狗模式.....	279
15.3.4	看门狗溢出处理.....	279
15.3.5	看门狗暂停	279
15.3.6	看门狗停止及再次运行	279
15.3.7	寄存器写保护	279
15.4	编程示例	280
15.4.1	配置 IWDG 为窗口看门狗	280
15.4.2	配置 IWDG 为独立看门狗	280

15.4.3	看门狗超时时间 (10kHz 的独立时钟)	281
15.5	寄存器描述	282
15.5.1	寄存器列表	282
15.5.2	键值寄存器 (IWDT_KR)	283
15.5.3	控制寄存器 (IWDT_CR)	284
15.5.4	重载寄存器 (IWDT_ARR)	285
15.5.5	状态寄存器 (IWDT_SR)	285
15.5.6	窗口寄存器 (IWDT_WINR)	286
16	窗口看门狗 (WWDT)	287
16.1	概述	287
16.2	主要特性	287
16.3	功能描述	287
16.3.1	功能框图	287
16.3.2	使能看门狗	288
16.3.3	配置看门狗超时	288
16.3.4	更新递减计数器	288
16.3.5	预溢出中断	288
16.4	编程示例	288
16.5	寄存器描述	289
16.5.1	寄存器列表	289
16.5.2	控制寄存器 0 (WWDT_CR0)	290
16.5.3	控制寄存器 1 (WWDT_CR1)	290
16.5.4	状态寄存器 (WWDT_SR)	291
17	通用同步异步收发器 (USART)	292
17.1	概述	292
17.2	主要特性	292
17.2.1	异步串口特性	292
17.2.2	同步串口特性	292
17.2.3	智能卡接口特性	292
17.2.4	红外通信特性	293
17.2.5	LIN 特性	293
17.3	功能说明	294
17.3.1	通信波特率编程	294
17.3.2	自动波特率检测	300
17.3.3	DMA 连续通信	300

17.3.4	硬件流控	301
17.3.5	多功能定时计数器	303
17.3.6	中断	305
17.3.7	异步串口模式	306
17.3.8	同步串口模式	311
17.3.9	智能卡(7816)模式	317
17.3.10	红外模式	322
17.3.11	LIN 模式	325
17.4	寄存器描述.....	330
17.4.1	寄存器列表	330
17.4.2	控制寄存器 1 (USART_CR1)	331
17.4.3	控制寄存器 2 (USART_CR2)	333
17.4.4	控制寄存器 3 (USART_CR3)	335
17.4.5	波特率寄存器 (USART_BRR)	337
17.4.6	定时计数寄存器 (USART_TOR)	337
17.4.7	状态寄存器 (USART_SR)	338
17.4.8	发送数据寄存器 (USART_TDR)	340
17.4.9	中断使能寄存器 (USART_IER)	341
17.4.10	中断标志清除寄存器 (USART_ICR)	342
17.4.11	接收数据寄存器 (USART_RDR)	343
17.4.12	地址寄存器 (USART_ADDR)	343
17.4.13	地址掩码寄存器 (USART_ADDMR)	343
17.4.14	波特率检出值寄存器 (USART_BDR)	344
17.4.15	红外调制寄存器 (USART_IR)	344
18	循环冗余校验 (CRC)	345
18.1	概述	345
18.2	主要特性	345
18.3	功能描述	346
18.3.1	工作模式	346
18.3.2	编码方式	346
18.3.3	写入位宽	346
18.4	编程示例	347
18.4.1	CRC-16 编码模式	347
18.4.2	CRC-16 检验模式	347
18.4.3	CRC-32 编码模式	347

18.4.4	CRC-32 检验模式	347
18.5	寄存器描述	349
18.5.1	寄存器列表	349
18.5.2	控制和状态寄存器 (CRC_CSR)	350
18.5.3	结果寄存器 (CRC_RESULT)	350
18.5.4	数据寄存器 (CRC_DR)	351
19	模数转换器 (ADC)	352
19.1	模块简介	352
19.2	ADC 框图	352
19.3	转换时序及转换速度	353
19.4	单次转换模式	353
19.5	扫描转换模式	355
19.5.1	顺序扫描单次转换模式	355
19.5.2	顺序扫描连续转换模式	357
19.5.3	顺序扫描断续转换模式	357
19.5.4	扫描转换触发 DMA 读取	357
19.6	连续转换累加模式	357
19.7	ADC 转换外部触发源	360
19.8	ADC 转换结果比较	360
19.9	ADC 中断	361
19.10	使用温度传感器测量环境温度	361
19.11	寄存器描述	363
19.11.1	寄存器列表	363
19.11.2	ADC 模拟器件选择寄存器 (ADC_BGR)	364
19.11.3	ADC 基本配置寄存器 0 (ADC_CR0)	365
19.11.4	ADC 基本配置寄存器 1 (ADC_CR1)	367
19.11.5	ADC 顺序扫描转换通道配置寄存器 0 (ADC_SQR0)	368
19.11.6	ADC 顺序扫描转换通道配置寄存器 1 (ADC_SQR1)	368
19.11.7	ADC 顺序扫描转换通道配置寄存器 2 (ADC_SQR2)	369
19.11.8	ADC 转换结果 (ADC_RESULT)	369
19.11.9	ADC 转换结果累加值 (ADC_RESULTACC)	369
19.11.10	ADC 比较上阈值 (ADC_HT)	370
19.11.11	ADC 比较下阈值 (ADC_LT)	370
19.11.12	ADC 中断使能 (ADC_IER)	371
19.11.13	ADC 中断标志寄存器 (ADC_IFR)	372

19.11.14	ADC 中断清除寄存器 (ADC_ICR)	373
19.11.15	ADC 外部中断触发源配置寄存器 0 (ADC_EXTTRIGGER0)	374
19.11.16	ADC 单次转换启动控制寄存器 (ADC_SGLSTART)	376
19.11.17	ADC 顺序扫描转换启动控制寄存器 (ADC_SQRSTART)	376
20	模拟电压比较器 (VC)	377
20.1	简介	377
20.2	主要特性	377
20.3	功能描述	378
20.3.1	功能框图	378
20.3.2	建立/响应时间	378
20.3.3	迟滞功能	378
20.3.4	比较结果极性配置	379
20.3.5	窗口比较功能	379
20.3.6	滤波描述	379
20.4	寄存器描述	380
20.4.1	寄存器列表	380
20.4.2	控制寄存器 0 (VC_CR0)	381
20.4.3	控制寄存器 1 (VC_CR1)	382
20.4.4	控制寄存器 2 (VC_CR2)	384
20.4.5	状态寄存器 (VC_SR)	384
21	低电压检测器 (LVD)	385
21.1	LVD 简介	385
21.2	主要特性	385
21.3	功能描述	385
21.3.1	功能框图	385
21.3.2	迟滞功能	386
21.3.3	数字滤波	386
21.4	编程示例	387
21.4.1	LVD 配置为低电压复位	387
21.4.2	LVD 配置为电压变化中断	387
21.5	寄存器描述	388
21.5.1	寄存器列表	388
21.5.2	LVD 配置寄存器 (LVD_CR)	389
21.5.3	LVD 中断寄存器 (LVD_SR)	390
22	调试控制器 (DBGMC)	391

22.1	简介	391
22.2	DBGC 系统框图	391
22.3	SW-DP 调试端口 (SWD)	392
22.3.1	SWD 端口引脚	392
22.3.2	SW-DP 调试端口引脚	392
22.3.3	SWD 引脚上的内部上拉	392
22.4	寄存器描述	393
22.4.1	寄存器列表	393
22.4.2	定时器调试暂停控制寄存器 (SYSCtrl_DEBUG)	393
23	器件电子签名	394
23.1	产品唯一身份标识 (UID) 寄存器 (48 位)	394
23.2	产品型号寄存器	394
23.3	FLASH 容量寄存器	395
23.4	RAM 容量寄存器	395
23.5	管脚数量寄存器	395
24	附录 A SysTick 定时器	396
24.1	SysTick 定时器简介	396
24.2	设置 SysTick	396
24.3	SysTick 寄存器	397
24.3.1	SysTick 控制和状态寄存器 (CTRL)	397
24.3.1	SysTick 重载寄存器 (LOAD)	397
24.3.2	SysTick 当前值寄存器 (VAL)	397
24.3.3	SysTick 校准值寄存器 (CALIB)	398
	版本修订记录	399

表索引

表 1-1	地址划分表	28
表 1-2	外设地址划分表	28
表 2-1	各工作模式下复位信号	30
表 2-2	各功能模块在不同工作模式下工作状态	32
表 2-3	系统控制寄存器列表	33
表 3-1	休眠状态下的处理器收到中断请求时的行为	42
表 3-2	SYSCTRL 寄存器列表	44
表 4-1	复位寄存器列表	62
表 5-1	Cortex-M4 处理器中断一览	67
表 5-2	外部中断与 NVIC 中断输入对应关系	72
表 5-3	中断寄存器列表	76
表 6-1	端口功能配置表	83
表 6-2	PINMUX 复用功能	89
表 6-3	IOAUX1/2 复用功能	90
表 6-4	GPIO 基地址列表	93
表 6-5	GPIO 寄存器列表	93
表 7-1	I2C 时钟信号波特率	106
表 7-2	I2C 状态码表述	113
表 7-3	I2C 基地址列表	118
表 7-4	I2C 寄存器列表	118
表 8-1	SPI 中断	133
表 8-2	SPI 基地址列表	137
表 8-3	SPI 寄存器列表	137
表 9-1	CTRIM 寄存器列表	148
表 10-1	FLASH 容量划分	155
表 10-2	芯片保护等级表	160
表 10-3	FLASH 寄存器列表	161
表 11-1	RAM 地址映射	166
表 12-1	DMA 传输模式表	169
表 12-2	DMA 寄存器列表	175
表 13-1	管脚功能与定时器功能对照表	182
表 13-2	BTIM 定时器中断	189
表 13-3	GTIM 中断源列表	197

表 13-4	GTIM 触发互联列表	197
表 13-5	GTIM ETR 源列表	197
表 13-6	GTIM CH3 信号来源列表	197
表 13-7	BTIM 基地址列表	198
表 13-8	BTIM 寄存器列表	198
表 13-9	GTIM 基地址列表	204
表 13-10	GTIM 寄存器列表	204
表 14-1	ATIM PRS 分频比列表	215
表 14-2	4MHz 时钟下 Buzzer 输出频率与重载模式配置值对应列表	218
表 14-3	PWC 模式测量起止边沿配置表	219
表 14-4	PWC 模式测量信号源选择	219
表 14-5	捕获输入 CH0A 信号列表	239
表 14-6	捕获输入 CH0B 信号列表	240
表 14-7	MSCR 寄存器从模式和触发源列表	243
表 14-8	ATIM 中断源与中断请求列表	248
表 14-9	定时器互联信号列表	248
表 14-10	ATIM0 ETR 信号来源列表	248
表 14-11	ATIM0 CHA 输入信号来源列表	248
表 14-12	ATIM 基地址列表	249
表 14-13	ATIM 寄存器列表	249
表 15-1	IWDT 超时时间 (10kHz 的独立时钟)	281
表 15-2	IWDT 寄存器列表	282
表 16-1	WWDT 寄存器列表	289
表 17-1	USART 各模式采样计数和最高波特率	294
表 17-2	PCLK 为 4MHz 的波特率编程示例	295
表 17-3	PCLK 为 8MHz 的波特率编程示例	296
表 17-4	PCLK 为 16MHz 的波特率编程示例	296
表 17-5	PCLK 为 24MHz 的波特率编程示例	297
表 17-6	PCLK 为 32MHz 的波特率编程示例	298
表 17-7	PCLK 为 48MHz 的波特率编程示例	299
表 17-8	多功能定时计数器功能	303
表 17-9	USART 各模式下中断有效性	305
表 17-10	USART 基地址列表	330
表 17-11	USART 寄存器列表	330
表 18-1	CRC 寄存器列表	349

表 19-1	ADC 最高转换速度	353
表 19-2	ADC 中断	361
表 19-3	ADC 基地址列表	363
表 19-4	ADC 寄存器列表	363
表 20-1	VC 基地址列表	380
表 20-2	VC 寄存器列表	380
表 21-1	LVD 寄存器列表	388
表 22-1	SWD 端口引脚	392
表 22-2	SW-DP 调试端口引脚	392
表 22-3	DBGC 寄存器列表	393
表 23-1	UID 定义	394
表 24-1	SysTick 寄存器	397

图索引

图 1-1 系统构架图	27
图 2-1 工作模式状态转换图	29
图 3-1 时钟控制模块框图	35
图 3-2 PLL 系统框图	36
图 3-3 晶振时钟启动示意图	37
图 3-4 时钟切换示意图	39
图 4-1 复位来源示意图	60
图 5-1 只使用了高 3 位的优先级寄存器	68
图 5-2 中断向量表	69
图 5-3 中断激活和挂起状态	70
图 5-4 中断挂起状态被清除然后被重新确认	70
图 5-5 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	71
图 5-6 中断处理中产生的中断挂起也可以被确认	71
图 6-1 通用端口电路示意图	83
图 6-2 高阻-模拟配置示意图	84
图 6-3 输入悬空/上拉/下拉配置示意图	86
图 6-4 读取端口管脚数据同步图	86
图 6-5 输出配置示意图	87
图 6-6 复用功能配置示意图	88
图 6-7 端口外部中断示意图	92
图 7-1 I2C 传输协议	102
图 7-2 START 和 STOP 条件	102
图 7-3 I2C 总线上位传输	103
图 7-4 I2C 总线上应答信号	103
图 7-5 I2C 总线上的仲裁	104
图 7-6 I2C 功能模块图	105
图 7-7 主发送模式数据同步图	107
图 7-8 I2C 主机发送状态图	108
图 7-9 主接收模式数据同步图	109
图 7-10 I2C 主机接收状态图	109
图 7-11 从接收模式数据同步图	110
图 7-12 从机接收状态图	110
图 7-13 从发送模式数据同步图	111

图 7-14 I2C 从机发送状态图	111
图 7-15 I2C 广播呼叫状态图	112
图 8-1 SPI 框图	123
图 8-2 SPI 通信格式时序图	124
图 8-3 从机选择 NSS 管脚配置	125
图 8-4 SPI 全双工通信	126
图 8-5 SPI 单线半双工通信	126
图 8-6 SPI 单工发送通信示意	127
图 8-7 SPI 单工接收通信示意	127
图 8-8 SPI 多机系统	128
图 8-9 SPI 主机模式一般标志位说明图	132
图 8-10 SPI 从机模式一般标志位说明图	132
图 9-1 RCH 时钟校准工作框图	143
图 9-2 RCL 时钟校准工作框图	144
图 9-3 低功耗定时器工作框图	145
图 12-1 DMA 功能框图	168
图 12-2 DMA 软件 Block 传输	170
图 12-3 DMA 软件 Burst 传输	170
图 12-4 DMA 硬件 Block 传输	171
图 12-5 DMA 硬件 Burst 传输	172
图 13-1 BTIM 框图	184
图 13-2 BTIM 滤波单元示意图	184
图 13-3 BTIM 计数器波形	185
图 13-4 BTIM 定时模式框图	186
图 13-5 BTIM 计数器模式框图	186
图 13-6 BTIM 触发启动器模式框图	187
图 13-7 BTIM 触发启动模式计数示意图	187
图 13-8 BTIM 门控模式框图	188
图 13-9 BTIM 门控模式计数示意图	188
图 13-10 BTIM 定时器级联示意图	189
图 13-11 GTIM 框图	190
图 13-12 GTIM 滤波单元示意图	190
图 13-13 GTIM 计数器波形	191
图 13-14 GTIM 定时模式框图	192
图 13-15 GTIM 计数器模式框图	192

图 13-16	GTIM 触发启动器模式框图	193
图 13-17	GTIM 触发启动模式计数示意图	193
图 13-18	GTIM 门控模式框图	194
图 13-19	GTIM 门控模式计数示意图	194
图 13-20	GTIM 捕获功能示意图	195
图 13-21	GTIM 比较功能示意图	196
图 14-1	ATIM3 框图	213
图 14-2	ATIM0 框图	214
图 14-3	自由计数框图	216
图 14-4	重载计数波形	216
图 14-5	16 位重载计数波形	217
图 14-6	32 位自由计数波形	217
图 14-7	自由计数时序图	217
图 14-8	重载计数时序图 (预分频设置为 2)	218
图 14-9	PWC 测量框图	219
图 14-10	上升沿到下降沿高电平测量	220
图 14-11	下降沿到下降沿周期测量	220
图 14-12	上升沿到上升沿周期测量	220
图 14-13	上升沿到上升沿周期测量单次模式	221
图 14-14	计数器框图	222
图 14-15	无预分频的向上计数	223
图 14-16	带预分频的上计数	224
图 14-17	不带预分频的下计数	224
图 14-18	带预分频的下计数	224
图 14-19	带预分频的上下计数	225
图 14-20	边沿对齐计数器波形 (DIR = 1)	225
图 14-21	边沿对齐计数器波形 (DIR = 0)	225
图 14-22	中心对齐计数器波形	226
图 14-23	重复计数器产生更新时序 (RCR.UD=0 RCR.OV=0)	226
图 14-24	上溢屏蔽时候重复计数器产生更新时序 (RCR.OV=1, RCR.UD=0)	227
图 14-25	下溢屏蔽时候重复计数器产生更新时序 (RCR.OV=0, RCR.UD=1)	227
图 14-26	三角波模式缓存使能	228
图 14-27	三角波模式缓存无效	228
图 14-28	锯齿波模式上计数缓存使能	229
图 14-29	锯齿波模式上计数缓存无效	229

图 14-30	锯齿波模式下计数缓存使能.....	229
图 14-31	锯齿波模式下计数缓存无效.....	230
图 14-32	锯齿波模式下计数比较缓存使能.....	230
图 14-33	OCREF 输出框图	231
图 14-34	锯齿波计数单点比较 OCREF 输出波形 (OCMx=111)	232
图 14-35	三角波计数单点比较 OCREF 输出波形 (OCMx=111)	232
图 14-36	锯齿波计数双点比较 OCREF 输出 (OCMx=111)	233
图 14-37	三角波计数双点比较 OCREF 输出 (OCMx=111)	233
图 14-38	独立 PWM 输出框图	233
图 14-39	CCPx=0 时 PWM 输出波形	234
图 14-40	CCPx=1 时 PWM 输出波形	234
图 14-41	互补 PWM 输出框图	234
图 14-42	互补 PWM 输出波形图.....	234
图 14-43	互补 PWM 输出波形图.....	235
图 14-44	死区时间.....	235
图 14-45	三角波模式单脉冲计数.....	236
图 14-46	锯齿波上计数单脉冲模式	237
图 14-47	锯齿波下计数单脉冲模式	237
图 14-48	中断示意图	238
图 14-49	捕获功能框图.....	239
图 14-50	捕获时序图	239
图 14-51	ATIM3 CH0A 端口选择.....	240
图 14-52	CHB 端口选择	240
图 14-53	从模式示意图.....	243
图 14-54	门控功能.....	243
图 14-55	触发和复位功能	244
图 14-56	编码计数功能框图.....	245
图 14-57	编码计数.....	246
图 14-58	ADC 触发	247
图 15-1	IWDT 功能框图	278
图 16-1	WWDT 功能框图	287
图 17-1	异步串口模式硬件流控示例.....	301
图 17-2	CTS 硬件流控信号	302
图 17-3	RTS 硬件流控信号	302
图 17-4	异步串口功能框图.....	306

图 17-5	异步串口模式数据格式.....	307
图 17-6	发送数据示例.....	307
图 17-7	接收数据示例.....	308
图 17-8	多处理器通信数据格式.....	309
图 17-9	同步串口模式字符格式图例.....	312
图 17-10	同步串口模式发送数据图例 1.....	313
图 17-11	同步串口模式发送数据图例 2.....	314
图 17-12	同步串口模式接收数据图例 1.....	315
图 17-13	同步串口模式接收数据图例 2.....	315
图 17-14	同步串口模式注意事项.....	316
图 17-15	智能卡连接示意图.....	317
图 17-16	智能卡模式同步时序和采样时序图	318
图 17-17	智能卡模式数据格式	318
图 17-18	智能卡模式数据格式示例	319
图 17-19	智能卡模式发送数据图例	320
图 17-20	智能卡模式接收数据图例	321
图 17-21	红外数据调制.....	322
图 17-22	38K 红外调制.....	324
图 17-23	LIN 总线数据和帧结构.....	325
图 19-1	ADC 示意框图	352
图 19-2	ADC 转换时序图	353
图 19-3	ADC 顺序扫描转换过程示例	355
图 19-4	ADC 连续转换累加过程示例	358
图 19-5	ADC 单次转换或顺序扫描转换外部触发源示意图	360
图 19-6	ADC 转换结果比较	361
图 20-1	VC 功能框图	378
图 20-2	VC 迟滞功能	378
图 20-3	VC 滤波响应时间	379
图 21-1	LVD 框图.....	385
图 21-2	LVD 迟滞响应.....	386
图 21-3	LVD 滤波输出.....	386
图 22-1	调试控制系统.....	391

简介 (Overview)

HC32F420 系列是一款采用高性能 ARM® Cortex®-M4 内核的 32 位微控制器，工作电压范围为 2.0~5.5V、工作温度范围为-40~105°C，具备丰富的外设配置和多种工作模式以适用不同的应用场景。

本产品提供多种封装形式，各封装形式下外设配置不同。

典型应用

- 物联网应用、智能家居
- 智能仪表、智能交通、医疗设备、手持设备
- 电机应用、变频控制
- 报警系统、打印机、扫描仪等工业应用

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”。

1 系统结构

1.1 概述

本产品系统由以下部分组成：

- 4 个 AHB Master：
 - Cortex-M4 IBUS
 - Cortex-M4 DBUS
 - Cortex-M4 SBUS
 - DMA 控制器
- 多个 AHB Slave：
 - FLASH 存储器
 - SRAM 存储器
 - 1 个 AHB to APB Bridge 包含所有 APB 接口外设
 - 2 个 AHB to AHB Bridge 包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。

1.2 系统构架图

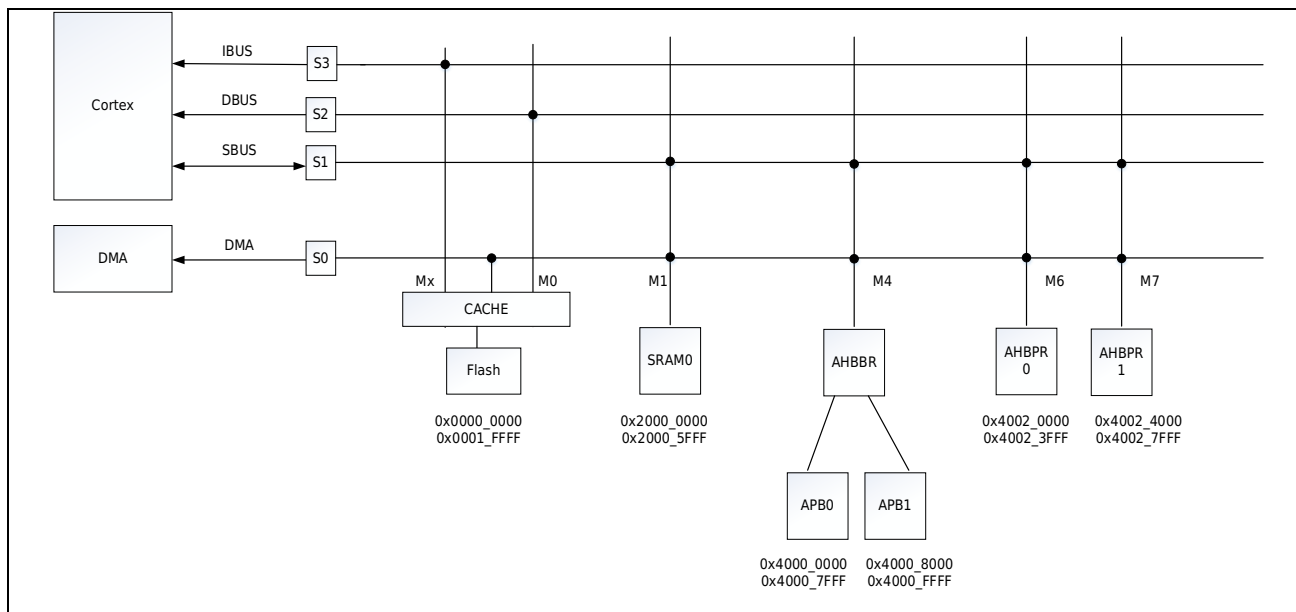


图 1-1 系统构架图

1.3 存储器及外设地址分配

表 1-1 地址划分表

类别	Boundary Address	Size	Memory Area
存储器	0x0000 0000 – 0x0001 FFFF	128K Byte	FLASH
	0x2000 0000 – 0x2000 5FFF	24K Byte	SRAM
APB 外设 AHBRR	0x4000 0000 – 0x4000 7FFF	32K Byte	APB0
	0x4000 8000 – 0x4000 FFFF	32K Byte	APB1
AHB 外设	0x4002 0000 – 0x4002 3FFF	16K Byte	AHBPR0
	0x4002 4000 – 0x4002 8FFF	16K Byte	AHBPR1

表 1-2 外设地址划分表

类别	基地址	Size	外设
APB0	0x4000 0000	0x400	USART0
	0x4000 0400	0x400	USART1
	0x4000 0800	0x400	USART2
	0x4000 0C00	0x400	USART3
	0x4000 2800	0x400	I2C0
	0x4000 2C00	0x400	I2C1
	0x4000 3800	0x400	SPI0
	0x4000 3C00	0x400	SPI1
	0x4000 5000	0x400	CTIM0
	0x4000 5400	0x400	CTIM1
	0x4000 5800	0x400	CTIM2
	0x4000 6000	0x400	ATIM0
	0x4000 6C00	0x400	ATIM3
APB1	0x4000 8000	0x400	CTRIM
	0x4000 8800	0x400	WWDT
	0x4000 8C00	0x400	IWDT
	0x4000 A000	0x400	ADC0
	0x4000 A400	0x400	ADC1
	0x4000 A800	0x400	VC
	0x4000 AC00	0x400	LVD
	0x4000 B800	0x400	SYSCTRL
	0x4000 BC00	0x400	保留
AHBPR0	0x4002 0000	0x400	DMAC
	0x4002 0800	0x400	CRC
	0x4002 1400	0x400	FLASH
AHBPR1	0x4002 4000	0x400	GPIOA
	0x4002 4400	0x400	GPIOB
	0x4002 4800	0x400	GPIOC
	0x4002 4C00	0x400	GPIOD

2 工作模式

2.1 工作模式概述

本产品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。

本产品的工作模式如下所示：

- 运行模式（Active Mode）：CPU 运行，片内外设运行
- 休眠模式（Sleep Mode）：CPU 停止，片内外设运行
- 深度休眠模式（Deep Sleep Mode）：CPU 停止，低功耗片内外设运行

2.2 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 WFI 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。

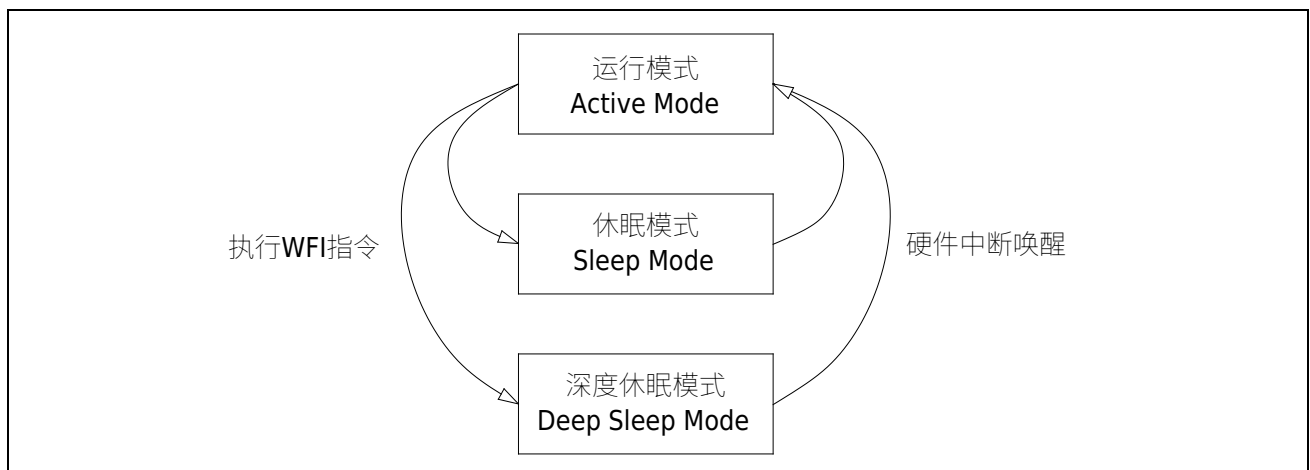


图 2-1 工作模式状态转换图

在各种工作模式下，CPU 可响应的中断见【中断源】章节。

在各种工作模式下，可响应的复位信号见下表。

表 2-1 各工作模式下复位信号

复位源	运行模式	休眠模式	深度休眠模式
上电掉电复位 POR/ PDR	✓	✓	✓
硬件复位 NRESET 管脚	✓	✓	✓
硬件复位 LVD	✓	✓	✓
硬件复位 IWDT	✓	✓	✓
硬件复位 WWDT	✓	✓	—
硬件复位 Cortex-M4 LOCKUP	✓	—	—
软件复位 Cortex-M4 SYSRESETREQ	✓	—	—

2.3 运行模式 (Active Mode)

在该工作模式 MCU 所有功能模块均正常工作，功耗最大。

系统复位后或从中断唤醒后，MCU 均工作于运行状态。

在运行模式下可通过一定的措施降低运行功耗：

- 为系统时钟 (SystemClk) 选择更低的频率，修改 `SYSCTRL_CR0.CLKSRC`
- 降低系统的内核时钟 (HCLK) 的频率，修改 `SYSCTRL_CR0.HCLKPRS`
- 降低系统的片内外设时钟 (PCLK) 的频率，修改 `SYSCTRL_CR0.PLCK0PRS/ PLCK1PRS`
- 关闭未使用的片内外设的时钟，修改 `SYSCTRL_APBEN0/APBEN1` 和 `SYSCTRL_AHBEN0/ AHBEN1`

2.4 休眠模式 (Sleep Mode)

在该工作模式下 CPU 停止运行，其它片内外设均正常工作，功耗略低于运行模式。

在运行模式下，设置 `SCB_SCR.SLEEPDEEP` 为 0，执行 WFI 指令即可使 MCU 工作于休眠模式。当 MCU 响应硬件中断时，即可退出休眠模式回到运行模式。

在休眠模式下可通过一定的措施降低运行功耗：

- 为系统时钟 (SystemClk) 选择更低的频率，修改 `SYSCTRL_CR0.CLKSRC`
- 降低系统的内核时钟 (HCLK) 的频率，修改 `SYSCTRL_CR0.HCLKPRS`
- 降低系统的片内外设时钟 (PCLKx) 的频率，修改 `SYSCTRL_CR0.PLCK0PRS/ PLCK1PRS`
- 关闭未使用的片内外设的时钟，修改 `SYSCTRL_APBEN0/APBEN1` 和 `SYSCTRL_AHBEN0/ AHBEN1`
- 设置 `SCB_SCR.SLEEPONEXIT` 为 1，使 MCU 执行完成中断服务程序后自动进入休眠模式

2.5 深度休眠模式 (Deep Sleep Mode)

在该工作模式下 CPU 停止运行，高速时钟 (RCH) 自动关闭。如果 HCLK 来源为高速时钟，则片内高速外设停止运行，仅片内低功耗外设正常工作；如果 HCLK 来源为低速时钟 (RCL)，则片内外设均正常工作。该模式下功耗远低于休眠模式下的功耗。本系统不支持系统时钟为 PLL 时进入该模式，建议在进入该模式前将系统时钟切换为 `RCHCLKD`。`RCHCLKD` 是指 24M 的 RCH 时钟经过 `SYSCTRL_RCHCR.DIV` 寄存器配置后所产生的时钟，例如 `SYSCTRL_RCHCR.DIV=1` 则 `RCHCLKD` 频率为 6MHz。

若设置 `SYSCTRL_CR1.WAKEUPCLK` 为 1，则 MCU 退出深度休眠模式时的系统时钟的来源为 RCH，以实现快速唤醒。

在运行模式下，设置 `SCB_SCR.SLEEPDEEP` 为 1，执行 WFI 指令即可使 MCU 工作于深度休眠模式。当 MCU 响应硬件中断时，即可退出深度休眠模式回到运行模式。

在休眠模式下可通过一定的措施降低运行功耗：

- 选择 HCLK 来源为高速时钟，修改 SYSCTRL_CR0.CLKSRC
- 关闭未使用的片内低功耗外设的时钟，修改 SYSCTRL_APBEN0/ APBEN1 和 SYSCTRL_AHBEN0/ AHBEN1
- 设置 SCB_SCR.SLEEPONEXIT 为 1，使 MCU 执行完成中断服务程序后自动进入休眠模式

2.6 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 2-2 各功能模块在不同工作模式下工作状态

功能模块	运行模式	休眠模式	深度休眠模式
CM4	✓	—	—
DMAC	✓	✓	—
GPIO	✓	✓	✓
VC	✓	✓	✓
LVD	✓	✓	✓
ADC	✓	✓	—
ATIM	✓	✓	—
CTIM	✓	✓	—
SPI	✓	✓	—
I2C	✓	✓	—
USART	✓	✓	—
CTRIM	✓	✓	✓
WWDT	✓	✓	—
IWDT	✓	✓	✓
CRC	✓	✓	

2.7 寄存器描述

2.7.1 寄存器列表

基地址：0xE000 ED10

表 2-3 系统控制寄存器列表

寄存器名	偏移地址	复位值
系统控制寄存器 (SCB_SCR)	0x00	0x0000 0000

2.7.2 系统控制寄存器 (SCB_SCR)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											SEV ONPE ND	Res	SLEE PDEE P	SLEE PONE XIT	Res

位	标记	位名	功能	读写
b31~b5	Res	保留位	保留，必须保持复位值	
b4	SEVONPEND		1：每次新的中断挂起都会产生一个事件，如果使用了WFE 休眠，它可用于唤醒处理器	RW
b3	Res	保留位	保留，必须保持复位值	
b2	SLEEPDEEP		1：执行 WFI 进入深度休眠，本产品进入 Deep Sleep 模式 0：执行 WFI 进入休眠，本产品进入 Sleep/Idle 模式	RW
b1	SLEEPONEXIT		1：当退出异常处理并返回程序线程时，处理器自动进入休眠模式(WFI) 0：该特性就会被自动禁止	RW
b0	Res	保留位	保留，必须保持复位值	

3 系统控制器 (SYSCTRL)

3.1 系统时钟介绍

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟。为了确保振荡器精度，内部时钟均具有校准功能。

本产品支持以下四个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 RCH（输出频率为 22~24MHz）
- 内部低速 RC 时钟 RCL（38.4K 与 32.768K 可配置）
- 外部高速晶振时钟 XTH
- 锁相环时钟 PLL

本产品还支持以下两个辅助时钟源：

- 内部 RC10K 时钟，仅供 IWDT 使用
- 内部 RC150K 时钟，仅供 LVD 和 VC 模块使用

注：

- 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，详见【系统时钟切换】章节。
- 2：XTH 可以不接晶振，直接从 PD00 管脚输入时钟信号。

通过 PCLKx_OUT 管脚可以输出 PCLK 时钟信号；通过 MCO 管脚可以输出 RCH/ RCL/ XTH/ PLL/ RC150K/ RC10K 时钟信号。

3.1.1 时钟架构图

本产品的时钟架构如下方所示。

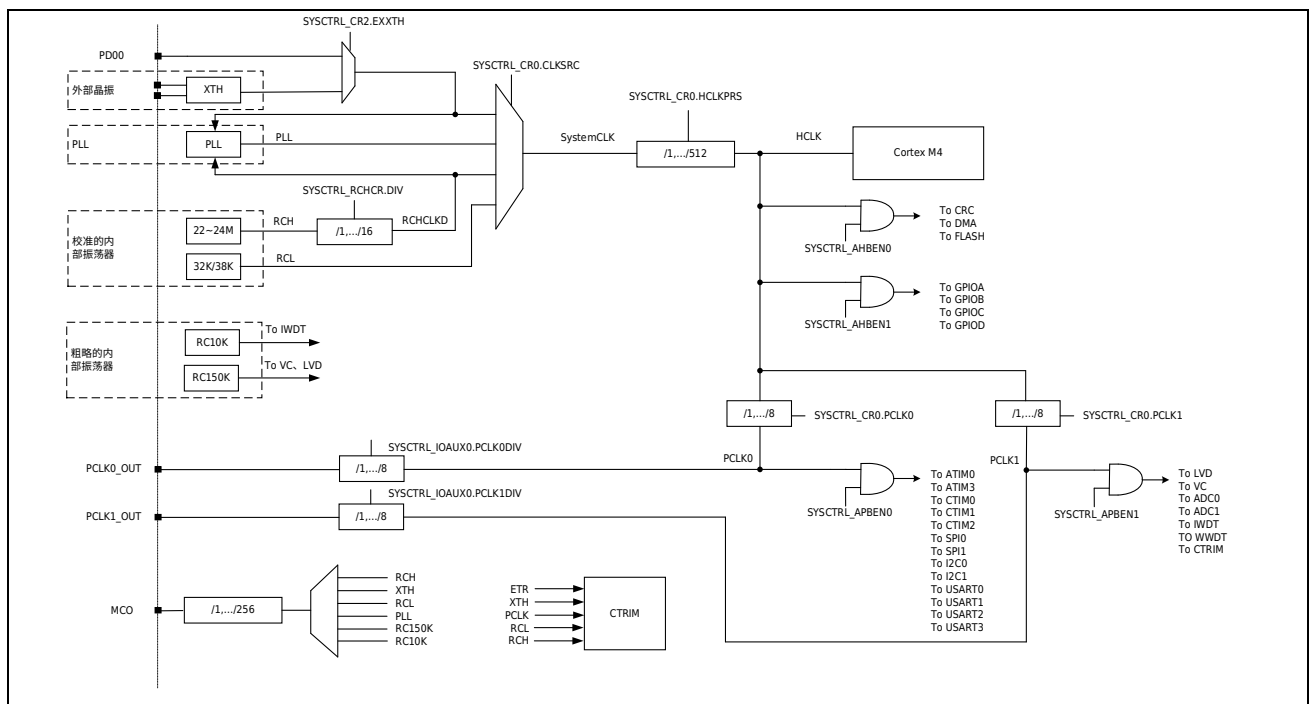


图 3-1 时钟控制模块框图

3.1.2 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为 RCHCLKD (其频率为 4MHz)；当系统进入 DeepSleep，此高速时钟会自动关闭。内部高速时钟从唤醒到稳定仅需 5us，在进入深度休眠模式前将系统时钟切换为 RCHCLKD 则系统可快速响应中断。

更改寄存器 SYSCTRL_RCHTRIM.TRIM 的数值即可微调 RCH 的振荡频率。寄存器数值每增加 1 则 RCH 的振荡频率增加约 0.2%。RCH 振荡频率的工作范围为 22~24MHz。芯片在出厂时已预调好的 2 个频率点 22.12MHz 和 24MHz；如需其它频率请手动调整该寄存器的数值。若在 CTRIM 模块中使能 RCH 实时校准功能，则 RCH 在全工作范围内的精度均可达 0.5%，详见【时钟校准模块 (CTRIM)】章节。

更改寄存器 SYSCTRL_RCHCR.DIV 的数值即可调整 RCH 振荡频率与 RCHCLKD 的分频比，可调整分频比为 1/2/4/6/8/10/12/14/16。

3.1.3 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 SYSCTRL_RCLTRIM.TRIM 来选择其输出频率，可供选择的频率为 38.4KHz、32.768KHz。当系统进入 DeepSleep，此低速时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

3.1.4 外部晶振时钟 XTH

外部晶振时钟需外接一个 8M~32M 的晶体

XTH 也可以不接晶振，直接从 PD00 管脚输入时钟信号。从 PD00 输入时钟信号的方法为：配置 PD00 引脚为 GPIO 数字输入；设置 SYSCTRL_CR2.EXXTH 为 1。

3.1.5 锁相环时钟 PLL

内建 PLL 支持 15~84MHz 的时钟输出，PLL 的参考时钟源可选择为 RCHCLKD 与 XTH。本系统不支持系统时钟为 PLL 时进入 DeepSleep，建议在进入 DeepSleep 前将系统时钟切换为其它时钟。

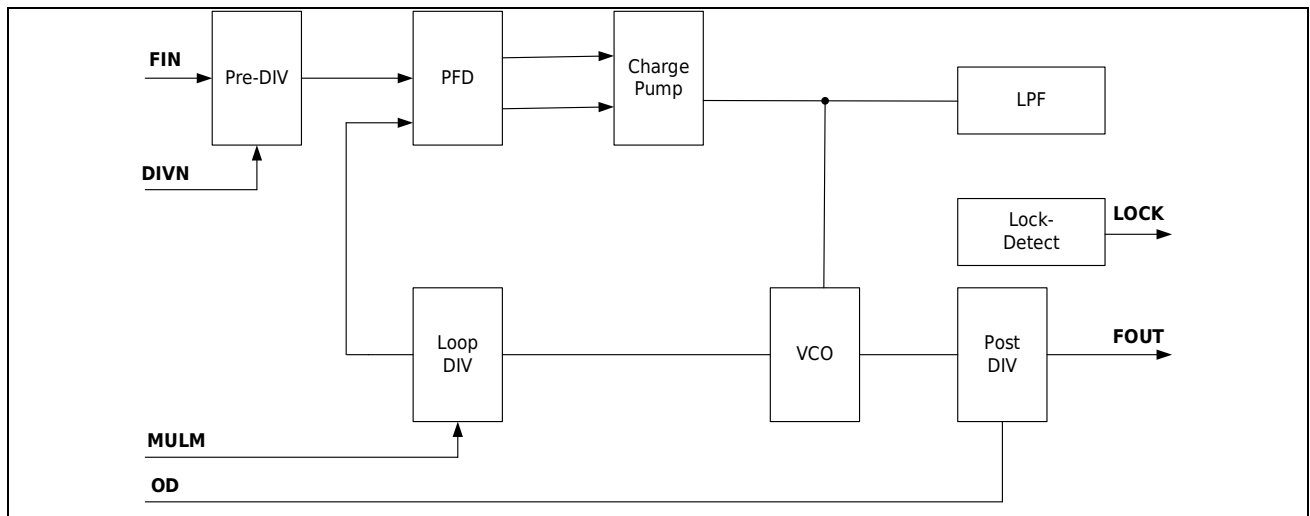


图 3-2 PLL 系统框图

FIN 输入频率范围 4MHz~32MHz

FOUT 输出频率范围 15MHz~84MHz

$$FOUT = FIN * \frac{MULM}{DIVN} * \frac{1}{OD}$$

FIN/DIVN 频率范围 4MHz~16MHz

FIN*MULM/DIVN 频率范围 120MHz~200MHz

MULM 取值范围 2~63

DIVN 取值范围 1~63

OD 取值范围为 OD 位所设置的除频值

注：MULM、DIVN 和 OD 为 SYSCTRL_PLLCR 寄存器的控制位

3.1.6 时钟启动过程

上述 4 种时钟源都需要启动稳定时间，下图以外部 XTH 为例说明时钟的启动稳定过程。

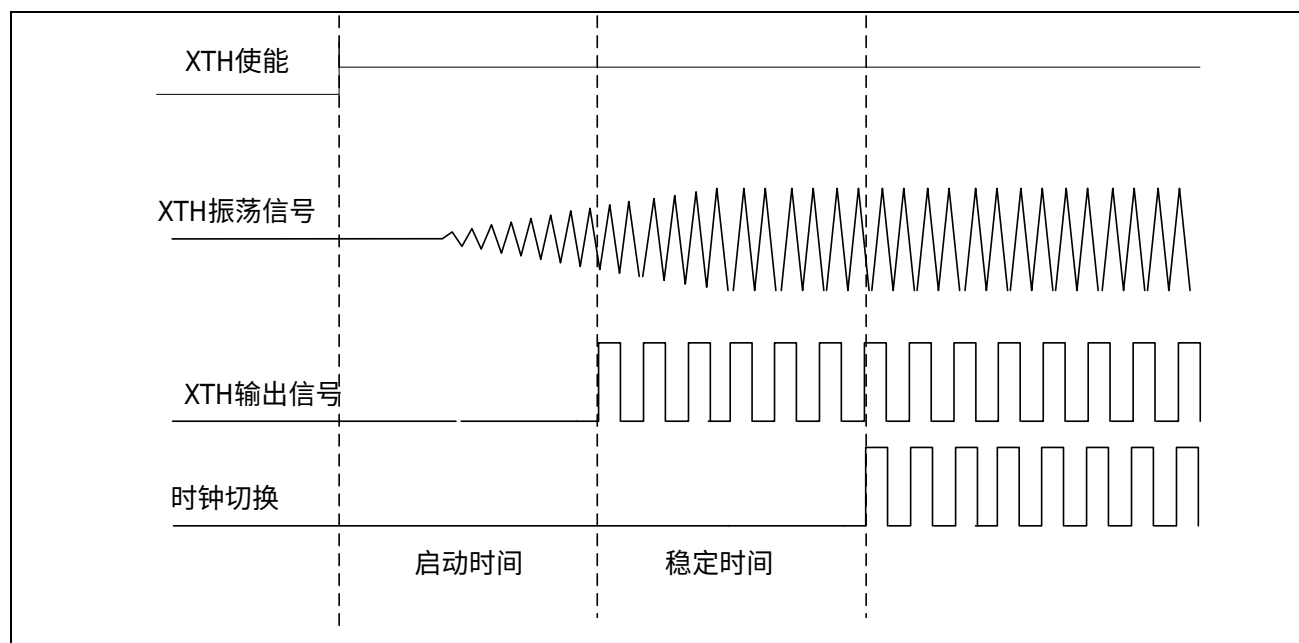


图 3-3 晶振时钟启动示意图

3.2 系统时钟切换

系统时钟可选 4 种时钟源：RCHCLKD、RCL、XTH、PLL，由寄存器 SYSCTRL_CR0.CLKSRC 进行控制。RCHCLKD、RCL、PLL、XTH 这 4 个时钟源中的任意两者可以相互切换，切换过程中需确保 HCLK 不大于 24MHz。

建议按照下文所描述的时钟切换流程进行时钟切换，以减少时钟切换出错的机率。

时钟切换时需要同步配置 FLASH_WAIT 寄存器：

- 时钟频率不大于 24MHz，设置 FLASH_WAIT 为 0（1 个周期）
- 时钟频率在 24~48MHz 间，设置 FLASH_WAIT 为 1（2 个周期）
- 时钟频率在 48~72MHz 间，设置 FLASH_WAIT 为 2（3 个周期）
- 时钟频率在 72~84MHz 间，设置 FLASH_WAIT 为 3（4 个周期）

注：

设置 FLASH_WAIT 的值，需要先向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5 然后再对 FLASH_WAIT 进行赋值，详见【FLASH 控制器（FLASH）】章节。

3.2.1 标准的时钟切换流程

操作流程如下：

Step1. 如新时钟源需要外部引脚，则将该引脚设置为适当的模式；

注：接外部晶振时需要模拟引脚；接外部时钟输入时需要 GPIO 数字输入并输入外部时钟信号；

Step2. 配置新时钟源的振荡参数；

Step3. 使能新时钟源的振荡器；

Step4. 根据当前时钟源和新时钟源两者中较高的频率，按 Flash 控制器章节流程配置 FLASH_WAIT 寄存器；

Step5. 等待新时钟源输出稳定的频率；

Step6. 配置 SYSCTRL_CR0.CLKSRC，选择系统时钟的来源为新时钟源；

Step7. 根据新时钟源的频率，按 Flash 控制器章节流程配置 FLASH_WAIT 寄存器；

Step8. 关闭不再使用的时钟源。

下图为时钟切换时序图：

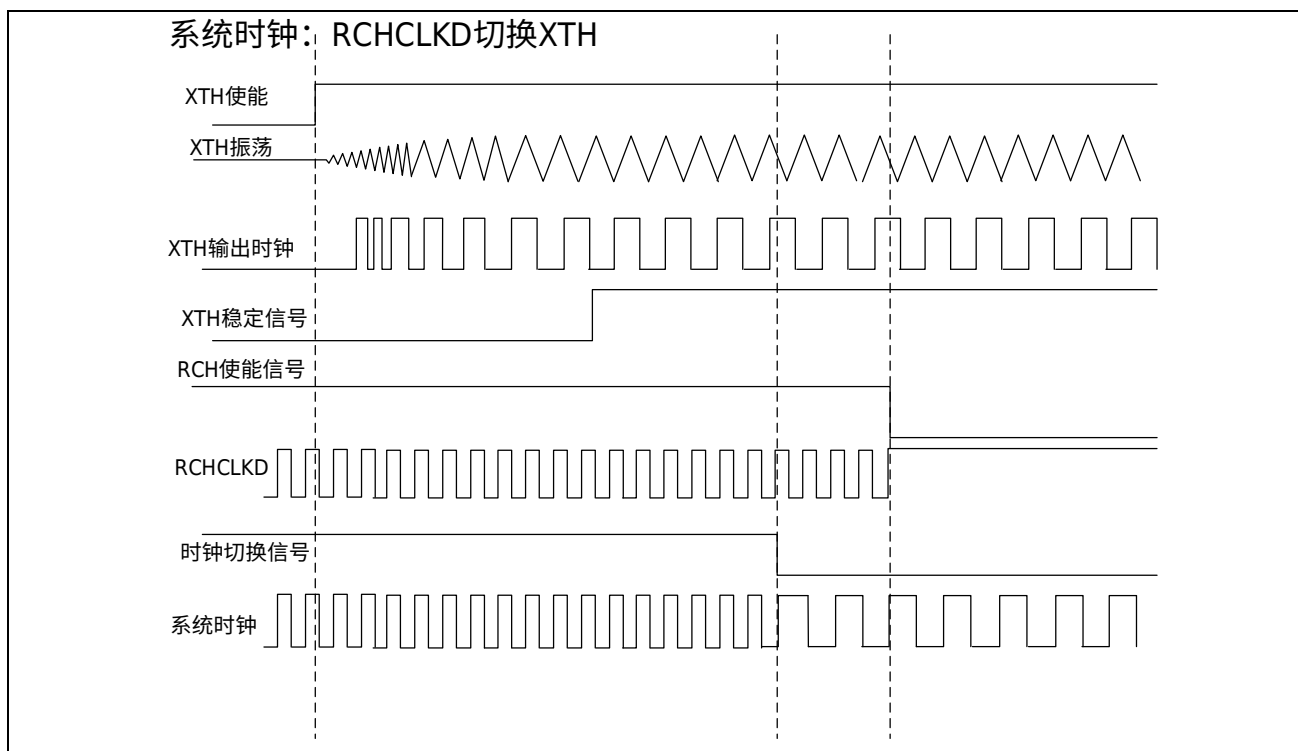


图 3-4 时钟切换示意图

3.2.2 RCHCLKD 不同输出频率间切换流程

配置 SYSCTRL_RCHCR.DIV 即可使 RCH 输出不同的时钟频率。

操作流程如下方所示：

- Step1. 根据当前 RCHCLKD 与目标 RCHCLKD 两者中较高的频率，按 Flash 章节流程配置 FLASH_WAIT 寄存器；
- Step2. 配置 SYSCTRL_RCHCR.DIV，选择 RCH 振荡频率与 RCHCLKD 频率的分频比；
- Step3. 根据当前 RCHCLKD 的频率，按 Flash 章节流程配置 FLASH_WAIT 寄存器。

3.2.3 从其它时钟切换到 RCL 示例

操作流程如下：

- Step1. 配置 SYSCTRL_RCLTRIM.TRIM 及 SYSCTRL_RCLCR.WAITCYCLE；
- Step2. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step3. 设置 SYSCTRL_CR2.RCL 为 1，使能 RCL 振荡电路；
- Step4. 查询等待 SYSCTRL_STATE.RCL 标志变为 1，RCL 输出稳定时钟；
- Step5. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step6. 设置 SYSCTRL_CR0.CLKSRC 为 2，将系统时钟切换为 RCL；
- Step7. 按 Flash 章节流程设置 FLASH_WAIT 为 0；
- Step8. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step9. 设置 SYSCTRL_CR2.xxx 为 0，关闭不再使用的时钟源。

3.2.4 从其它时钟切换到 RCHCLKD 示例

操作流程如下：

- Step1. 配置 SYSCTRL_RCHTRIM.TRIM 及 SYSCTRL_RCHCR.DIV；
- Step2. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step3. 设置 SYSCTRL_CR2.RCH 为 1，使能 RCH 振荡电路；
- Step4. 查询等待 SYSCTRL_STATE.RCH 标志变为 1，RCH 输出稳定时钟；
- Step5. 根据当前时钟与目标 RCHCLKD 两者中较高的频率，按 Flash 章节流程配置 FLASH_WAIT 寄存器；
- Step6. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；

- Step7. 设置 SYSCTRL_CR0.CLKSRC 为 0，将系统时钟切换为 RCHCLKD；
- Step8. 根据 RCHCLKD 的频率，按 Flash 章节流程配置 FLASH_WAIT；
- Step9. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step10. 设置 SYSCTRL_CR2.xxx 为 0，关闭不再使用的时钟源。

3.2.5 PLL（参考时钟为 RCHCLKD）与 RCHCLKD 相互切换示例

从 RCHCLKD 切换到 PLL 操作流程如下：

- Step1. 设置 SYSCTRL_PLLCR.SRC 为 0，选择 PLL 时钟源为 RCHCLKD；
- Step2. 根据 PLL 输入频率范围配置 SYSCTRL_RCHCR.DIV 值使 RCHCLKD 在 4MHz~12MHz 之间；
- Step3. 根据 PLL 输出频率，配置 SYSCTRL_PLLCR 中的 MULM、DIVN、OD、BYPASS、OEN，根据电源噪声配置 LKEN。要求 FIN/DIVN 范围：4M-16MHz，FIN*MULM/DIVN 范围：120M-200MHz；
- Step4. 配置 SYSCTRL_PLLCR.WAITCYCLE，选择适当的 PLL 稳定时间；
- Step5. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step6. 设置 SYSCTRL_CR2.PLL 为 1，使能 PLL 振荡电路；
- Step7. 根据 PLL 输出频率，按 Flash 章节流程配置 FLASH_WAIT；
- Step8. 查询等待 SYSCTRL_STATE.PLL 标志变为 1，PLL 输出稳定时钟；
- Step9. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step10. 设置 SYSCTRL_CR0.CLKSRC 为 4，将系统时钟切换为 PLL；

警告：切换过程中不可以更改 SYSCTRL_RCHCR。

注：当 PLL 的输入时钟选择为 RCH 时，该输入频率不可以为 24MHz。

从 PLL 切换到 RCHCLKD 操作流程如下：

- Step1. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step2. 设置 SYSCTRL_CR0.CLKSRC 为 0，将系统时钟切换为 RCHCLKD；
- Step3. 根据 RCHCLKD 的频率，按 Flash 章节流程配置 FLASH_WAIT；
- Step4. 向 SYSCTRL_CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step5. 设置 SYSCTRL_CR2.xxx 为 0，关闭不再使用的时钟源；

警告：切换过程中不可以更改 SYSCTRL_RCHCR。

3.3 片内外设时钟控制

大多数片内外设都需要有时钟才能正常工作。当使能片内外设的配置时钟（PCLKx）时，用户程序可以正常读写片内外设的寄存器；当使能片内设的工作时钟时，片内外设可以正常工作。部分片内外设的工作时钟固定为 PCLKx，部分片内外设的工作时钟可以由用户程序自行配置。

通过设置 SYSCTRL_APBEN0/ APBEN1 和 SYSCTRL_AHBEN0/ AHBEN1 的控制位为 1，即可使能相应片内外设的配置时钟。

3.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在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。

休眠状态下的处理器收到中断请求时的行为如下表所示：

表 3-1 休眠状态下的处理器收到中断请求时的行为

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

3.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

- Step1. 使能需要唤醒处理器的模块所对应的 NVIC
- Step2. 使能需要唤醒处理器的模块所对应的中断
- Step3. 设置 SCB_SCR.SLEEPDEEP 为 1
- Step4. 执行 WFI 指令以进入深度休眠模式
- Step5. 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

例程：

```
SCB->SCR |= 0x04;
while(1)
{
    __asm("WFI");
}
```

3.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

- Step1. 使能需要唤醒处理器的模块所对应的 NVIC
- Step2. 使能需要唤醒处理器的模块所对应的中断

- Step3. 设置 PRIMASK 为 1
- Step4. 设置 SCB_SCR.SLEEPDEEP 为 1
- Step5. 执行 WFI 指令以进入深度休眠模式
- Step6. 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
- Step7. 清除中断标志，清除中断挂起状态
- Step8. 执行用户定义的操作

例程：

```
__asm("CPSID I"); //Set PRIMASK
SCB_SCR |= 0x04;
while(1)
{
    __asm("WFI");
    M4_TIMx->ICLR = 0x00; //Clear Int Flag
    NVIC_ClearPendingIRQ(TIMx_IRQn); //Clear Pending Flag
    //执行用户定义的操作
}
```

3.4.3 使用退出休眠特性

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模式。利用退出休眠特性，处理器可以尽可能多的处于休眠模式。

Cortex-M4 利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，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。

- Step1. 使能需要唤醒处理器的模块所对应的 NVIC
- Step2. 使能需要唤醒处理器的模块所对应的中断
- Step3. 设置 SCB_SCR.SLEEPDEEP 为 1
- Step4. 设置 SCB_SCR.SLEEPONEXIT 为 1
- Step5. 执行 WFI 指令以进入深度休眠模式
- Step6. 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
- Step7. 退出中断服务时自动进入休眠模式

例程：

```
SCB_SCR |= 0x04;
SCB_SCR |= 0x02;
while(1)
{
    __asm("WFI");
}
```

3.5 寄存器描述

3.5.1 寄存器列表

基地址：0x4000 B800

表 3-2 SYSCTRL 寄存器列表

寄存器名	偏移地址	复位值
SYSCTRL_CR0 (系统控制寄存器 0)	0x00	0x0000 0000
SYSCTRL_CR1 (系统控制寄存器 1)	0x04	0x0000 0000
SYSCTRL_CR2 (时钟使能控制寄存器 2)	0x08	0x0000 0001
SYSCTRL_CR3 (序列控制寄存器)	0x0C	0x0000 0000
SYSCTRL_RCHCR (RCH 控制寄存器)	0x10	0x0000 0002
SYSCTRL_RCLCR (RCL 控制寄存器)	0x18	0x0000 0000
SYSCTRL_XTHCR (XTH 控制寄存器)	0x14	0x0000 0002
SYSCTRL_PLLCR (PLL 控制寄存器)	0x20	0x000C 02C0
SYSCTRL_RCHTRIM (RCH 校准值寄存器)	0x24	24M 校准值地址所在的校准值
SYSCTRL_RCLTRIM (RCL 校准值寄存器)	0x28	32.768k 校准值地址所在的校准值
SYSCTRL_STATE (时钟状态寄存器)	0x2C	0x0000 0001
SYSCTRL_APBEN0 (APB0 外设时钟控制寄存器)	0x30	0x0000 0000
SYSCTRL_APBEN1 (APB1 外设时钟控制寄存器)	0x34	0x0000 8000
SYSCTRL_AHBEN0 (AHB0 外设时钟控制寄存器)	0x38	0x0000 0020
SYSCTRL_AHBEN1 (AHB1 外设时钟控制寄存器)	0x3C	0x0000 0000
SYSCTRL_DEBUG (定时器调试暂停控制寄存器)	0x64	0x0000 0000
SYSCTRL_IOAUX0 (GPIO 辅助控制寄存器 0)	0x70	0x0000 0000
SYSCTRL_IOAUX1 (GPIO 辅助控制寄存器 1)	0x74	0x0000 0000
SYSCTRL_IOAUX2 (GPIO 辅助控制寄存器 2)	0x78	0x0000 0000

3.5.2 系统控制寄存器 0 (SYSCTRL_CR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				PCLK1PRS		PCLK0PRS		HCLKPRS				Res	CLKSRC		

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留，必须保持复位值	
b11~b10	PCLK1PRS	PCLK1 时钟来源选择	00： HCLK	RW
			01： HCLK/2	
			10： HCLK/4	
			11： HCLK/8	
b9~b8	PCLK0PRS	PCLK0 时钟来源选择	00： HCLK	RW
			01： HCLK/2	
			10： HCLK/4	
			11： HCLK/8	
b7~b4	HCLKPRS	HCLK 时钟来源选择	0000： SystemClk	RW
			0001： SystemClk/2	
			0010： SystemClk/4	
			0011： SystemClk/8	
			0100： SystemClk/16	
			0101： SystemClk/32	
			0110： SystemClk/64	
			0111： SystemClk/128	
			1000： SystemClk/256	
			1001： SystemClk/512	
其它值： SystemClk				
b3	Res	保留位	保留，必须保持复位值	
b2~b0	CLKSRC	SystemClk 时钟来源选择	000： 内部高速时钟 RCHCLKD (RCHCLKD 为 RCH 经 SYSCTRL_RCHCR.DIV 分频后值)	RW
			001： 外部高速时钟 XTH	
			010： 内部低速时钟 RCL	
			011： 保留	
			100： 内部高速时钟 PLL	
注：				
1) 设置 SYSCTRL_CR1.WAKEUPCLK=1，唤醒后时钟自动切换到 RCHCLKD，否则保持原时钟				
2) 写入非有效值时，值保持原值不变				

注:

每次改写 SYSCTRL_CR0/ CR1/ CR2 的值, 均需要先对 SYSCTRL_CR3 依次写入 0x5A5A、0x5A5A。这样的步骤可有效防止对 SYSCTRL_CR0/ CR1/ CR2 寄存器的误操作。

3.5.3 系统控制寄存器 1 (SYSCTRL_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	GTIM2CFG		GTIM1CFG		GTIM0CFG		WAKE UPCLK	LOCK UPEN	FAULT SW	XTHF LT	SPI_P CTRL	Res		SW DIO	

位	标记	位名	功能	读写
b31~b14	Res	保留位	保留, 必须保持复位值	
b13~b12	GTIM2CFG	GTIM2 功能选择	00: GTIM 做为通用定时器 GTIM2 01: GTIM 做为基本定时器 BTIM6/7/8 1x: 保留, 禁止设置	RW
b11~b10	GTIM1CFG	GTIM1 功能选择	00: GTIM 做为通用定时器 GTIM1 01: GTIM 做为基本定时器 BTIM3/4/5 1x: 保留, 禁止设置	RW
b9~b8	GTIM0CFG	GTIM0 功能选择	00: GTIM做为通用定时器 GTIM0 01: GTIM 做为基本定时器 BTIM0/1/2 1x: 保留, 禁止设置	RW
b7	WAKEUPCLK	配置 DeepSleep 唤醒时, 系统时钟的来源	1: 切换系统时钟来源为 RCHCLKD, 原系统时钟来源保持使能 0: 保持原系统时钟来源。	RW
b6	LOCKUPEN	Cortex-M Lockup 功能配置	0: 关闭 1: 使能 注: 使能后, CPU 读到无效指令时会复位 MCU, 可增强系统可靠性。	RW
b5	FAULTSW	系统时钟来源为晶体且晶体失效时, 自动切换系统时钟来源配置	0: 保持系统时钟来源不变 1: 切换系统时钟来源为 RCHCLKD 注: 使用该功能时, 硬件开启 XTH 时钟滤波功能, XTH 时钟振荡频率不可以超过 10M, XTH 作为 PLL 的参考时钟时, 禁止开启该功能。	RW
b4	XTHFLT	XTH 时钟滤波使能。 (仅 XTH 频率小于 12M 可以使用)	0: 禁止 1: 使能 注: 1) 使用 XTH 做系统时钟时, 禁止更改该功能 2) XTH 作为 PLL 的参考时钟时, 禁止开启该功能 3) 需要软件开启 RCH 时钟	RW
b3	SPI_PCTRL	SPI端口方向控制	0: SPI模块控制 1: GPIO 控制	RW
b2~b1	Res	保留位	保留, 必须保持复位值	
b0	SWDIO	配置 SWD 端口功能	0: PA13、PA14 作为 SWD 端口, GPIO 功能不可用。 1: PA13、PA14 作为 GPIO 端口, SWD 功能不可用。	RW

3.5.4 时钟使能控制寄存器 2 (SYSCTRL_CR2)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										EXXTH	PLL	Res	RCL	XTH	RCH

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
			0: 使用内部时钟, 1: 使用 PD00 输入外部时钟	
b5	EXXTH	XTH外部输入时钟使能	注: 使用 PD00 输入时钟时, 需设置 PD00 为数字输入端口, 且 CR2.XTH 为 1; 当使用 PD00 外部输入时钟时 PD01 管脚禁止作为 GPIO 使用, PD01 端口需要配置为模拟状态。 该控制位使能前确保 PD00 有时钟输入。	RW
			0: 关闭 1: 使能	
b4	PLL	时钟 PLL 使能控制	注: 1) 当系统进入 DeepSleep, 此时钟会自动关闭 2) 系统时钟选择为当前时钟时无法关闭当前时钟	RW
b3	Res	保留位	保留, 必须保持复位值	
			0: 关闭 1: 使能	
b2	RCL	内部低速时钟 RCL使能控制	注: 1) 当系统进入DeepSleep, 此时钟保持之前状态 2) 系统时钟选择为当前时钟时无法关闭当前时钟	RW
			0: 关闭 1: 使能	
b1	XTH	外部高速时钟 XTH 使能控制	注: 1) 当系统进入 DeepSleep, 此时钟会自动关闭 2) 系统时钟选择为当前时钟时无法关闭当前时钟	RW
			0: 关闭 1: 使能	
b0	RCH	内部高速时钟 RCH 使能控制	注: 1) 系统进入 DeepSleep, 此时钟会自动关闭 2) 系统时钟选择为当前时钟时无法关闭当前时钟	RW

3.5.5 序列控制寄存器 3 (SYSCTRL_CR3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CR3															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CR3	序列控制	寄存器 SYSCTRL_CR0/1/2 保护系列控制寄存器, 对 SYSCTRL_CR3 先写 0x5A5A, 再写 0xA5A5, 启动对于寄存器 SYSCTRL_CR0/1/2 的写操作, 只要对寄存器 SYSCTRL_CR0/1/2 写操作了, 这个保护位自动回复保护状态, 需要重新写入序列打开保护。	WO

3.5.6 RCH 控制寄存器 (SYSCTRL_RCHCR)

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												DIV			

位	标记	位名	功能	读写
b31~b4	Res	保留位	保留, 必须保持复位值	
b3~b0	DIV	RCH 输出频率配置	0000: 12.00 MHz	RW
			0001: 6.00 MHz	
			0010: 4.00 MHz	
			0011: 3.00 MHz	
			0100: 2.40 MHz	
			0101: 2.00 MHz	
			0110: 1.71 MHz	
			0111: 1.50 MHz	
			1000: 24.00 MHz	
			其它值: 禁止	

3.5.7 RCL 控制寄存器 (SYSCTRL_RCLCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														WAITCYCLE	
位		标记			位名			功能					读写		
b31~b2		Res			保留位			保留，必须保持复位值							
b1~b0		WAITCYCLE			内部低速时钟 RCL 稳定时间选择			11: 256 个周期 10: 64 个周期 01: 16 个周期 00: 4 个周期					RW		

3.5.8 XTH 控制寄存器 (SYSCTRL_XTHCR)

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							DETIE	DETTIME	DETEN	WAITCYCLE	RANGE		DRIVER		

位	标记	位名	功能	读写
b31~b9	Res	保留位	保留, 必须保持复位值	
b8	DETIE	中断使能控制	1: 使能 Fault 与 Fail 中断 0: 禁止 Fault 与 Fail 中断	RW
b7	DETTIME	晶体失效检测时间配置	0: 2ms 检测间隔 1: 8ms 检测间隔	RW
b6	DETEN	起振失效检测与工作失效检测使能控制	1: 使能 0: 禁止 注: 开启该功能需要软件开启 RCL 时钟使能	RW
b5~b4	WAITCYCLE	外部高速时钟 XTH 稳定时间选择	00: 8192 个周期 01: 32768 个周期 10: 131072 个周期 11: 262144 个周期 注: 强烈建议将 XTH 的稳定时间设置为 11。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
b3~b2	RANGE	外部晶振工作频率选择	11: 20M~32M 10: 16M~24M 01: 8M~16M 00: 8M~12M 注: 上述为推荐配置, 用户应根据实际应用做适当调整。	RW
b1~b0	DRIVER	外部晶振驱动能力选择	11: 最强驱动能力 10: 默认驱动能力 (推荐值) 01: 弱驱动能力 00: 最弱驱动能力 注: 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

3.5.9 PLL 控制寄存器 (SYSCTRL_PLLCR)

复位值: 0x000C 02C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										MULM					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LKEN	BYPASS	DIVN						OEN	WAITCYCLE			OD		SRC	

位	标记	位名	功能	读写
b31~b22	Res	保留位	保留, 必须保持复位值	
b21~b16	MULM	Loop divider	2 ≤ MULM ≤ 63	RW
b15	LKEN	Lock enable	1: Force LOCK=1	RW
b14	BYPASS	Bypass mode	1: FOUT = FIN/DIVN	RW
b13~b8	DIVN	Pre divider	1 ≤ DIVN ≤ 63	RW
b7	OEN	Output enable	输出使能	RW
b6~b4	WAITCYCLE	PLL 稳定时间选择	000: 128 个 PLL 周期 001: 256 个 PLL 周期 010: 512 个 PLL 周期 011: 1024 个 PLL 周期 100: 2048 个 PLL 周期 101: 4096 个 PLL 周期 110: 8192 个 PLL 周期 111: 16384 个 PLL 周期	RW
b3~b2	OD	输出控制	除频值为 00: 1 01: 2 10: 4 11: 8	RW
b1~b0	SRC	输入时钟选择	00: RCH 时钟 01: XTH 时钟 (由 SYSCTRL_CR2.EXXTH 决定从管脚 PD00 输入或晶振内部产生) 1X: XTH 内部时钟	RW
注: 当输入时钟选择为 RCH 时钟时, 该输入频率不可以为 24MHz。				

3.5.10 RCH 校准值寄存器 (SYSCTRL_RCHTRIM)

复位值：24M 校准值地址所在的校准值（上电下电复位与外部复位管脚复位会重新装载该值）

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRIM								
位	标记	位名	功能										读写		
b31~b9	Res	保留位	保留，必须保持复位值												
b8~b0	TRIM	内部高速时钟频率调整,更改该寄存器的数值即可调整 RCH 的基础频率。	寄存器数值每增加1则RCH的输出频率增加约0.2%，总调整范围为 22 - 24 MHz。 FLASH中已保存了24MHz的校准值，将FLASH内的校准值读出并写入RCHTRIM即可获得精准的频率。 24M 校准值地址：0x010011E4 - 0x010011E5 22M 校准值地址：0x010011E0 - 0x010011E1										RW		

3.5.11 RCL 校准值寄存器 (SYSCTRL_RCLTRIM)

复位值：32.768k 校准值地址所在的校准值（上电下电复位与外部复位管脚复位会重新装载该值）

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRIM								
位	标记	位名	功能										读写		
b31~b10	Res	保留位	保留，必须保持复位值												
b9~b0	TRIM	内部低速时钟频率调整,更改该寄存器的数值即可调整 RCL 的基础频率。	将 FLASH 内的校准值读出并写入 RCLTRIM 即可获得精准的频率。 38.4k 校准值地址： 0x010011EC - 0x010011ED 32.768k 校准值地址：0x010011E8 - 0x010011E9										RW		

3.5.12 时钟状态寄存器 (SYSCTRL_STATE)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						XTHF AULT	XTHF AIL	Res		PLLL OCK	PLL	Res	RCL	XTH	RC H

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	XTHFAULT	XTH 工作中失效标志	0: 工作正常 1: 工作中失效, 写0清除 注: 如果 XTH 作为系统时钟并使用失效时钟自动切换, 失效后使用软件将时钟切换到 RCH 后再清除该标志。	RO
b8	XTHFAIL	XTH 起振标志	0: 起振正常 1: 起振失败, 写 0 清除	RO
b7~b6	Res	保留位	保留, 必须保持复位值	
b5	PLLLOCK	PLL 锁定状态	0: 未锁定 1: 已经锁定	RO
b4	PLL	PLL 稳定状态	0: 未稳定 1: 已经稳定	RO
b3	Res	保留位	保留, 必须保持复位值	
b2	RCL	RCL 时钟稳定状态	0: 未稳定 1: 已经稳定	RO
b1	XTH	XTH 时钟稳定状态	0: 未稳定 1: 已经稳定	RO
b0	RCH	RCH 时钟稳定状态	0: 未稳定 1: 已经稳定	RO

3.5.13 APB0 外设时钟控制寄存器 (SYSCTRL_APBEN0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				ATIM3	Res		ATIM0	Res	CTIM2	CTIM1	CTIM0	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SPI1	SPI0	Res		I2C1	I2C0	Res						USAR T3	USAR T2	USAR T1	USAR T0

位	标记	位名	功能	读写
b31~b28	Res	保留位	保留, 必须保持复位值	
b27	ATIM3	ATIM3 模块时钟使能	1: 使能 0: 禁止	RW
b26~b25	Res	保留位	保留, 必须保持复位值	
b24	ATIM0	ATIM0 模块时钟使能	1: 使能 0: 禁止	RW
b23	Res	保留位	保留, 必须保持复位值	
b22	CTIM2	CTIM2 模块时钟使能	1: 使能 0: 禁止	RW
b21	CTIM1	CTIM1 模块时钟使能	1: 使能 0: 禁止	RW
b20	CTIM0	CTIM0 模块时钟使能	1: 使能 0: 禁止	RW
b19~b16	Res	保留位	保留, 必须保持复位值	
b15	SPI1	SPI1 模块时钟使能	1: 使能 0: 禁止	RW
b14	SPI0	SPI0 模块时钟使能	1: 使能 0: 禁止	RW
b13~b12	Res	保留位	保留, 必须保持复位值	
b11	I2C1	I2C1 模块时钟使能	1: 使能 0: 禁止	RW
b10	I2C0	I2C0 模块时钟使能	1: 使能 0: 禁止	RW
b9~b4	Res	保留位	保留, 必须保持复位值	
b3	USART3	USART3 模块时钟使能	1: 使能 0: 禁止	RW
b2	USART2	USART2 模块时钟使能	1: 使能 0: 禁止	RW
b1	USART1	USART1 模块时钟使能	1: 使能 0: 禁止	RW
b0	USART0	USART0 模块时钟使能	1: 使能 0: 禁止	RW

3.5.14 APB1 外设时钟控制寄存器 (SYSCTRL_APBEN1)

复位值: 0x0000 8000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				LVD	VC	ADC1	ADC0	Res				IWDT	WWDT	Res	CTRIM

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11	LVD	LVD 模块时钟使能	1: 使能 0: 禁止	RW
b10	VC	VC 模块时钟使能	1: 使能 0: 禁止	RW
b9	ADC1	ADC1 模块时钟使能	1: 使能 0: 禁止	RW
b8	ADC0	ADC0 模块时钟使能	1: 使能 0: 禁止	RW
b7~b4	Res	保留位	保留, 必须保持复位值	
b3	IWDT	IWDT 模块时钟使能	1: 使能 0: 禁止	RW
b2	WWDT	WWDT 模块时钟使能	1: 使能 0: 禁止	RW
b1	Res	保留位	保留, 必须保持复位值	
b0	CTRIM	CTRIM 模块时钟使能	1: 使能 0: 禁止	RW

3.5.15 AHB0 外设时钟控制寄存器 (SYSCTRL_AHBEN0)

复位值: 0x0000 0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										FLASH	Res	CRC	Res	DMA	C
位		标记		位名				功能				读写			
b31~b6		Res		保留位				保留，必须保持复位值							
b5		FLASH		FLASH 控制模块时钟使能				1：使能 0：禁止				RW			
b4~b3		Res		保留位				保留，必须保持复位值							
b2		CRC		CRC 模块时钟使能				1：使能 0：禁止				RW			
b1		Res		保留位				保留，必须保持复位值							
b0		DMAC		DMAC 模块时钟使能				1：使能 0：禁止				RW			

3.5.16 AHB1 外设时钟控制寄存器 (SYSCTRL_AHBEN1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												GPIO	GPIO	GPIO	GPIO
												D	C	B	A
位	标记		位名		功能		读写								
b31~b4	Res		保留位		保留, 必须保持复位值										
b3	GPIOD		GPIOD 模块时钟使能		1: 使能 0: 禁止		RW								
b2	GPIOC		GPIOC 模块时钟使能		1: 使能 0: 禁止		RW								
b1	GPIOB		GPIOB 模块时钟使能		1: 使能 0: 禁止		RW								
b0	GPIOA		GPIOA 模块时钟使能		1: 使能 0: 禁止		RW								

3.5.17 定时器调试暂停控制寄存器 (SYSCTRL_DEBUG)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			WWDT T	IWDT	CTRI M	Res			CTIM 2	CTIM 1	CTIM 0	ATIM 3	Res		ATIM 0

位	标记	位名	功能	读写
b31~b13	Res	保留位	保留, 必须保持复位值	
b12	WWDT	调试时, WWDT 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b11	IWDT	调试时, IWDT 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b10	CTRIM	调试时, CTRIM 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b9~b7	Res	保留位	保留, 必须保持复位值	
b6	CTIM2	调试时, CTIM2 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b5	CTIM1	调试时, CTIM1 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b4	CTIM0	调试时, CTIM0 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b3	ATIM3	调试时, ATIM3 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW
b2~b1	Res	保留位	保留, 必须保持复位值	
b0	ATIM0	调试时, ATIM0 计数功能配置	1: 在 SWD 调试界面下, 暂停计数功能 0: 在 SWD 调试界面下, 正常计数功能	RW

3.5.18 GPIO 辅助控制寄存器 0 (SYSCTRL_IOAUX0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	MCOSEL			Res	MCODIV			Res	PCLK1DIV			Res	PCLK0DIV		

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14~b12	MCOSEL	MCO 管脚时钟选择	000: RCH 模拟输出 001: XTH 模拟输出 010: RCL 模拟输出 011: 保留 100: PLL 模拟输出 101: RC150K 模拟输出 110: RC10K 模拟输出 111: 保留	RW
b11	Res	保留位	保留, 必须保持复位值	
b10~b8	MCODIV	MCO 分频控制	000: 关闭输出 001: 直接输出 010: 8 分频输出 011: 64 分频输出 其它: 256 分频输出	RW
b7	Res	保留位	保留, 必须保持复位值	
b6~b4	PCLK1DIV	PCLK1 分频控制	000: 关闭输出 001: 直接输出 010: 2 分频输出 011: 4 分频输出 其它: 8 分频输出	RW
b3	Res	保留位	保留, 必须保持复位值	
b2~b0	PCLK0DIV	PCLK0 分频控制	000: 关闭输出 001: 直接输出 010: 2 分频输出 011: 4 分频输出 其它: 8 分频输出	RW

3.5.19 GPIO 辅助控制寄存器 1 (SYSCTRL_IOAUX1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	CTIM2ETR			Res	CTIM1ETR			Res	CTIM0ETR			Res	ATIM0ETR		

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14~b12	CTIM2ETR	CTIM2 ETR 设置	参见 GPIO 章节或定时器章节描述	RW
b11	Res	保留位	保留, 必须保持复位值	
b10~b8	CTIM1ETR	CTIM1 ETR 设置	参见 GPIO 章节或定时器章节描述	RW
b7	Res	保留位	保留, 必须保持复位值	
b6~b4	CTIM0ETR	CTIM0 ETR 设置	参见 GPIO 章节或定时器章节描述	RW
b3	Res	保留位	保留, 必须保持复位值	
b2~b0	ATIM0ETR	ATIM0 ETR 设置	参见 GPIO 章节或定时器章节描述	RW

3.5.20 GPIO 辅助控制寄存器 2 (SYSCTRL_IOAUX2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	CTIM2CAP3			Res	CTIM1CAP3			Res	CTIM0CAP3			Res	ATIM0CAP0		

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14~b12	CTIM2CAP3	CTIM2 CAP 设置	参见 GPIO 章节或定时器章节描述	RW
b11	Res	保留位	保留, 必须保持复位值	
b10~b8	CTIM1CAP3	CTIM1 CAP 设置	参见 GPIO 章节或定时器章节描述	RW
b7	Res	保留位	保留, 必须保持复位值	
b6~b4	CTIM0CAP3	CTIM0 CAP 设置	参见 GPIO 章节或定时器章节描述	RW
b3	Res	保留位	保留, 必须保持复位值	
b2~b0	ATIM0CAP0	ATIM0 CAP 设置	参见 GPIO 章节或定时器章节描述	RW

4 复位控制器 (RESET)

4.1 复位控制器介绍

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。

- 上电掉电复位 POR PDR
- 外部 Reset 管脚，低电平为复位信号
- IWDT 复位
- WWDT 复位
- LVD 低电压复位
- Cortex-M4 SYSRESETREQ 软件复位
- Cortex-M4 LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示；复位完成时查询 SYSCTRL_RSTFLAG 寄存器即获取上一次的复位来源。每当完成对 SYSCTRL_RSTFLAG 寄存器的查询后，应将该寄存器清零，以便下一次复位时该寄存器能正常指示其复位源。

下图描述各区域的复位来源。

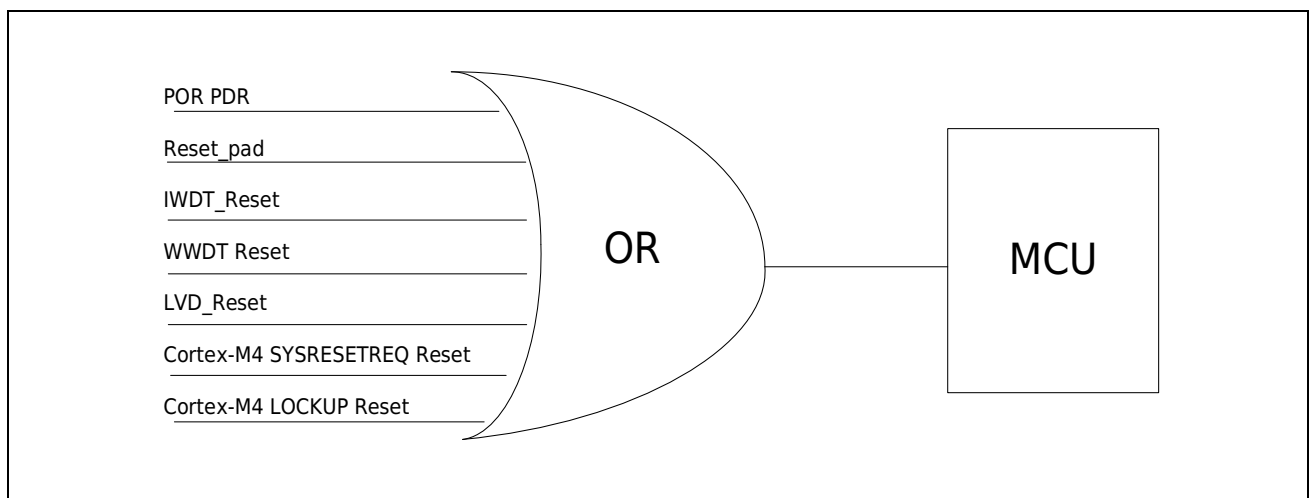


图 4-1 复位来源示意图

4.1.1 上电下电复位 POR PDR

本产品有两个供电区域：VCC 区域、VCORE 区域。

所有的模拟模块及 IO 工作于 VCC 区域；其它模块工作于 VCORE 区域。

VCC 区域上电时，当 VCC 电压低于 POR 阈值电压时，会产生 VCC 电源域复位信号；VCC 区域下电时，当 VCC 电压低于 PDR 阈值电压时，会产生 VCC 电源域复位信号。

VCORE 区域上电时，当 VCORE 电压低于 POR 阈值电压时，会产生 VCORE 域复位信号；VCORE 区域下电时，当 VCORE 电压低于 PDR 阈值电压时，会产生 VCORE 域复位信号。

VCC 电源域复位信号和 VCORE 域复位信号均会将芯片的寄存器复位到初始化状态。

4.1.2 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 20 μ s（典型值）的毛刺信号，因此，加到复位引脚上的低电平信号必须大于 20 μ s，才能确保芯片可靠复位。

4.1.3 WDT 复位

看门狗复位，请参考 IWDT/WWDT 一章说明。

4.1.4 LVD 低电压复位

LVD 复位，请参考 LVD 一章说明。

4.1.5 Cortex-M4 SYSRESETREQ 复位

Cortex-M4 软件复位。

4.1.6 Cortex-M4 LOCKUP 复位

当 Cortex-M4 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个 CORE 区域。

4.2 寄存器描述

4.2.1 寄存器列表

基地址：0x4000 B800

表 4-1 复位寄存器列表

寄存器名	偏移地址	复位值
SYSCTRL_APBRS0 (APB0 外设复位控制寄存器)	0x40	0x0970 CC0F
SYSCTRL_APBRS1 (APB1 外设复位控制寄存器)	0x44	0x0000 0F0D
SYSCTRL_AHRS0 (AHB0 外设复位控制寄存器)	0x48	0x0000 0025
SYSCTRL_AHRS1 (AHB1 外设复位控制寄存器)	0x4C	0x0000 000F
SYSCTRL_RSTFLAG (复位标志寄存器)	0x60	0x0000 0003

4.2.2 APB0 外设复位控制寄存器 (SYSCTRL_APB0RST0)

复位值: 0x0970 CC0F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				ATIM3	Res		ATIM0	Res	CTIM2	CTIM1	CTIM0	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SPI1	SPI0	Res		I2C1	I2C0	Res						USAR T3	USAR T2	USAR T1	USAR T0

位	标记	位名	功能	读写
b31~b28	Res	保留位	保留, 必须保持复位值	
b27	ATIM3	ATIM3 模块复位	1: 无动作 0: 复位	RW
b26~b25	Res	保留位	保留, 必须保持复位值	
b24	ATIM0	ATIM0 模块复位	1: 无动作 0: 复位	RW
b23	Res	保留位	保留, 必须保持复位值	
b22	CTIM2	CTIM2 模块复位	1: 无动作 0: 复位	RW
b21	CTIM1	CTIM1 模块复位	1: 无动作 0: 复位	RW
b20	CTIM0	CTIM0 模块复位	1: 无动作 0: 复位	RW
b19~b16	Res	保留位	保留, 必须保持复位值	
b15	SPI1	SPI1 模块复位	1: 无动作 0: 复位	RW
b14	SPI0	SPI0 模块复位	1: 无动作 0: 复位	RW
b13~b12	Res	保留位	保留, 必须保持复位值	
b11	I2C1	I2C1 模块复位	1: 无动作 0: 复位	RW
b10	I2C0	I2C0 模块复位	1: 无动作 0: 复位	RW
b9~b4	Res	保留位	保留, 必须保持复位值	
b3	USART3	USART3 模块复位	1: 无动作 0: 复位	RW
b2	USART2	USART2 模块复位	1: 无动作 0: 复位	RW
b1	USART1	USART1 模块复位	1: 无动作 0: 复位	RW
b0	USART0	USART0 模块复位	1: 无动作 0: 复位	RW

4.2.3 APB1 外设复位控制寄存器 (SYSCTRL_APBRS1)

复位值: 0x0000 0F0D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				LVD	VC	ADC1	ADC0	Res				IWDT	WWD T	Res	CTR M

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11	LVD	LVD 模块复位	1: 无动作 0: 复位	RW
b10	VC	VC 模块复位	1: 无动作 0: 复位	RW
b9	ADC1	ADC1 模块复位	1: 无动作 0: 复位	RW
b8	ADC0	ADC0 模块复位	1: 无动作 0: 复位	RW
b7~b4	Res	保留位	保留, 必须保持复位值	
b3	IWDT	IWDT 模块复位	1: 无动作 0: 复位	RW
b2	WWD T	WWD T 模块复位	1: 无动作 0: 复位	RW
b1	Res	保留位	保留, 必须保持复位值	
b0	CTR M	CTR M 模块复位	1: 无动作 0: 复位	RW

4.2.4 AHB0 外设复位控制寄存器 (SYSCTRL_AHBRST0)

复位值: 0x0000 0025

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										FLASH	Res	CRC	Res	DMA	C
位		标记		位名				功能				读写			
b31~b6		Res		保留位				保留，必须保持复位值							
b5		FLASH		FLASH 控制模块复位				1：无动作 0：复位				RW			
b4~b3		Res		保留位				保留，必须保持复位值							
b2		CRC		CRC 模块复位				1：无动作 0：复位				RW			
b1		Res		保留位				保留，必须保持复位值							
b0		DMAC		DMAC 模块复位				1：无动作 0：复位				RW			

4.2.5 AHB1 外设复位控制寄存器 (SYSCTRL_AHBRST1)

复位值: 0x0000 000F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												GPIO D	GPIO C	GPIO B	GPIO A
位		标记		位名		功能							读写		
b31~b4		Res		保留位		保留，必须保持复位值									
b3		GPIOD		GPIOD 模块复位		1：无动作 0：复位							RW		
b2		GPIOC		GPIOC 模块复位		1：无动作 0：复位							RW		
b1		GPIOB		GPIOB 模块复位		1：无动作 0：复位							RW		
b0		GPIOA		GPIOA 模块复位		1：无动作 0：复位							RW		

4.2.6 复位标志寄存器 (SYSCTRL_RSTFLAG)

复位值: 0x0000 0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								NRST	SYSREQ	LOCKUP	WWDT	IWDT	LVD	CORE	VCC

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7	NRST	NRESET 端口复位标志	1: 发生端口复位 0: 无端口复位发生 写 0 清除, 写 1 无效 注: 上电时, 该值与外接复位电路相关	RW
b6	SYSREQ	Cortex-M4 CPU 软件复位标志	1: 发生 Cortex-M4 CPU 软件复位 0: 无 Cortex-M4 CPU 软件复位发生 写 0 清除, 写 1 无效	RW
b5	LOCKUP	Cortex-M4 CPU Lockup 复位标志	1: 发生 Cortex-M4 CPU Lockup 复位 0: 无 Cortex-M4 CPU Lockup 复位发生 写 0 清除, 写 1 无效	RW
b4	WWDT	WWDT 复位标志	1: 发生 WWDT 复位 0: 无 WWDT 复位发生 写 0 清除, 写 1 无效	RW
b3	IWDT	IWDT 复位标志	1: 发生 IWDT 复位 0: 无 IWDT 复位发生 写 0 清除, 写 1 无效	RW
b2	LVD	LVD 复位标志	1: 发生 LVD 复位 0: 无 LVD 复位发生 写 0 清除, 写 1 无效	RW
b1	CORE	VCORE 域复位标志	1: VCORE 域发生复位 0: VCORE 域无复位发生 写 0 清除, 写 1 无效	RW
b0	VCC	VCC 电源域复位标志	1: VCC 电源域发生复位 0: VCC 电源域无复位发生 写 0 清除, 写 1 无效	RW

注意: 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。

5 中断控制器 (NVIC)

5.1 概述

Cortex-M4 处理器内置了嵌套向量中断控制器 (NVIC)，本产品中最多支持 64 个中断请求 (IRQ) 输入，以及 1 个不可屏蔽中断 (NMI) 输入（在本产品系统中并未使用）。另外，处理器还支持多个内部异常。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。具体如下表所示：

表 5-1 Cortex-M4 处理器中断一览

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（在本系统中没有使用）
3	硬件错误	-1	错误处理异常
4-10	保留	NA	...
11	SVC	可编程	通过 SVC 指令调用管理程序
12-13	保留	NA	...
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
...	...	...	...
79	中断#63	可编程	外部中断#63

本章节只对处理器的 64 个外部中断请求（中断#0 到中断#63）做详细介绍，处理器内部异常的具体情况可参考其它相关文档。同时，本章节只讨论处理器内核中 NVIC 的中断处理机制，外设模块自身的中断产生机制不在此处展开讨论。

5.2 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 3 位宽，并且使用中断优先级寄存器的最高 3 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）、0x20、0x40、0x60、0x80、0xA0、0xC0、和 0xE0（最低）。

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

图 5-1 只使用了高 3 位的优先级寄存器

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。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另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断 #0 和中断 #1 使能且具有相同的优先级，在它们同时被触发时，中断 #0 会首先执行。

5.3 中断向量表

当 Cortex-M4 处理器要处理中断服务请求时，它需要首先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
0x0000 004C	中断#3向量	19
0x0000 0048	中断#2向量	18
0x0000 0044	中断#1向量	17
0x0000 0040	中断#0向量	16
0x0000 003C	SysTick向量	15
0x0000 0038	PendSV向量	14
0x0000 0034	未使用	13
0x0000 0030	未使用	12
0x0000 002C	SVC向量	11
0x0000 0028	未使用	10
0x0000 0024	未使用	9
0x0000 0020	未使用	8
0x0000 001C	未使用	7
0x0000 0018	未使用	6
0x0000 0014	未使用	5
0x0000 0010	未使用	4
0x0000 000C	硬件错误异常	3
0x0000 0008	NMI向量	2
0x0000 0004	复位向量	1
0x0000 0000	MSP初始值	0

图 5-2 中断向量表

其中，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。

5.4 中断输入和挂起行为

Cortex-M4 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。该过程如下图所示：

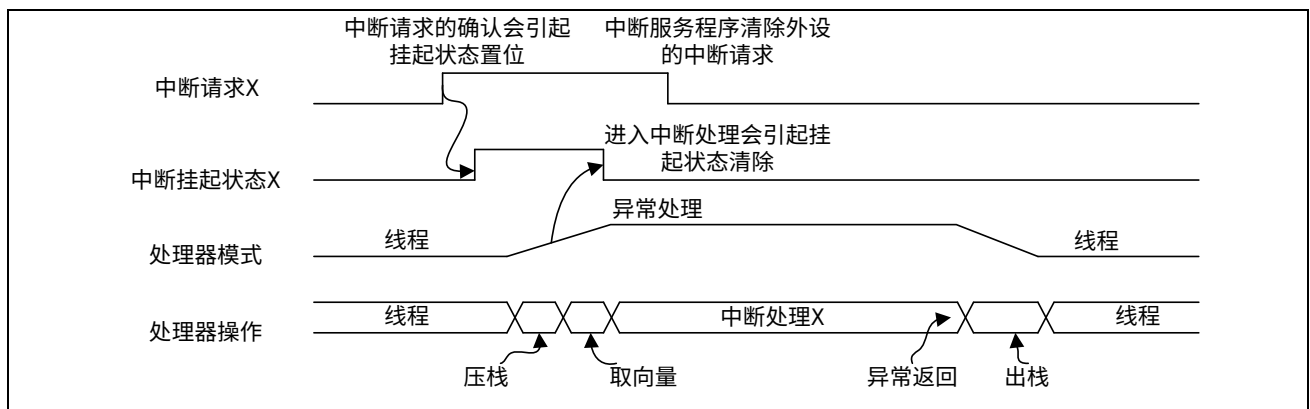


图 5-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 NVIC_ICPR 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

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。该过程如下图所示：

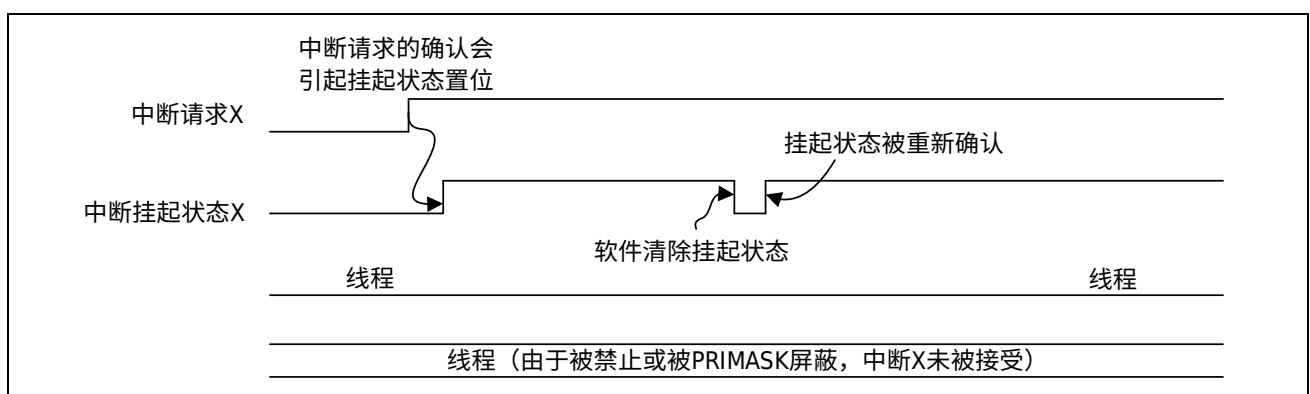


图 5-4 中断挂起状态被清除然后被重新确认

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。该过程如下图所示：

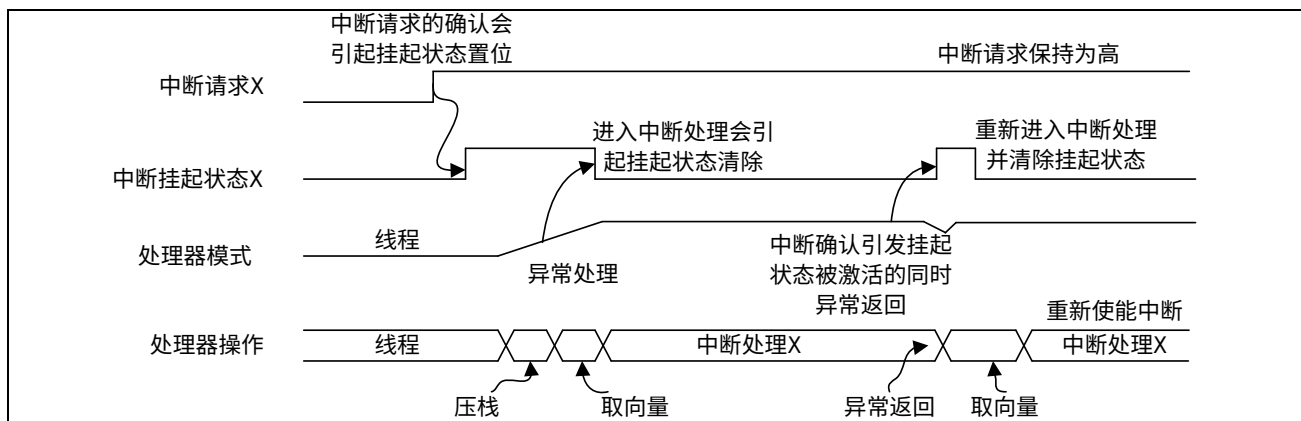


图 5-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

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。该过程如下图所示：

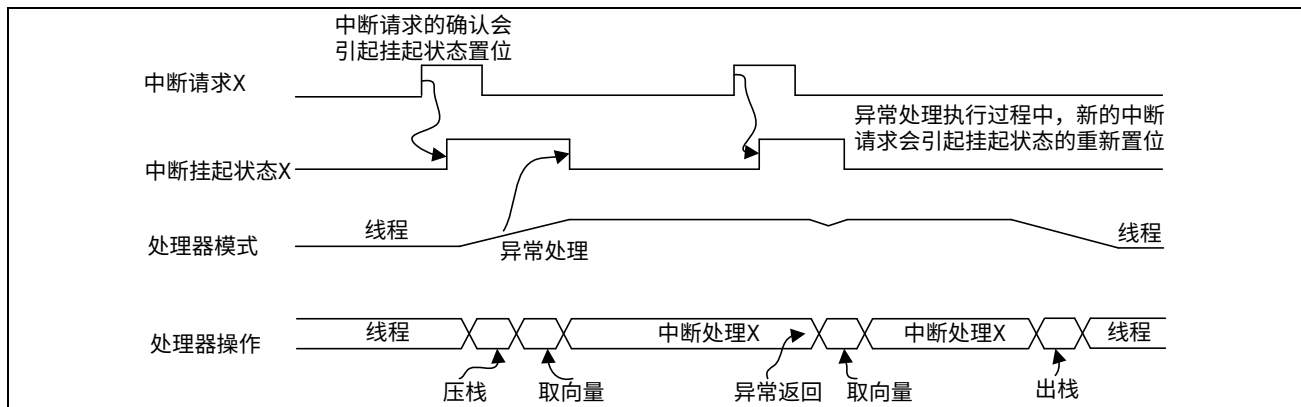


图 5-6 中断处理中产生的中断挂起也可以被确认

5.5 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 或是其它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。

5.6 中断源

因为 Cortex-M4 处理器的 NVIC 支持最多 128 个外部中断，而在本系统中，外部中断源使用了 64 个，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：

表 5-2 外部中断与 NVIC 中断输入对应关系

NVIC 中断输入	外部中断源	运行模式	休眠模式	深度休眠模式
中断#0	CTRIM	✓	✓	✓
中断#1	VC	✓	✓	✓
中断#2	LVD	✓	✓	✓
中断#3	GPIOA	✓	✓	✓
中断#4	GPIOB	✓	✓	✓
中断#5	GPIOC	✓	✓	✓
中断#6	PIOD	✓	✓	✓
中断#7	IWD	✓	✓	✓
中断#8	WWD	✓	✓	-
中断#9	CLK_FAULT	✓	✓	-
中断#10	ADC	✓	✓	-
中断#11	FLASHCFG	✓	✓	-
中断#12	USART0_RX	✓	✓	-
中断#13	USART0_TX	✓	✓	-
中断#14	USART0_FLAG	✓	✓	-
中断#15	USART1_RX	✓	✓	-
中断#16	USART1_TX	✓	✓	-
中断#17	USART1_FLAG	✓	✓	-
中断#18	USART2_RX	✓	✓	-
中断#19	USART2_TX	✓	✓	-
中断#20	USART2_FLAG	✓	✓	-
中断#21	USART3_RX	✓	✓	-
中断#22	USART3_TX	✓	✓	-
中断#23	USART3_FLAG	✓	✓	-
中断#24	I2C0	✓	✓	-
中断#25	I2C1	✓	✓	-
中断#26	SPI0_TX	✓	✓	-
中断#27	SPI0_RX	✓	✓	-
中断#28	SPI0_SSR	✓	✓	-
中断#29	SPI0_ERR	✓	✓	-
中断#30	SPI1_TX	✓	✓	-

NVIC 中断输入	外部中断源	运行模式	休眠模式	深度休眠模式
中断#31	SPI1_RX	✓	✓	—
中断#32	SPI1_SSR	✓	✓	—
中断#33	SPI1_ERR	✓	✓	—
中断#34	GTIM0_UEV_ETR_BTIMA	✓	✓	—
中断#35	GTIM0_CH01_BTIMB	✓	✓	—
中断#36	GTIM0_CH23_BTIMC	✓	✓	—
中断#37	GTIM1_UEV_ETR_BTIMA	✓	✓	—
中断#38	GTIM1_CH01_BTIMB	✓	✓	—
中断#39	GTIM1_CH23_BTIMC	✓	✓	—
中断#40	GTIM2_UEV_ETR_BTIMA	✓	✓	—
中断#41	GTIM2_CH01_BTIMB	✓	✓	—
中断#42	GTIM2_CH23_BTIMC	✓	✓	—
中断#43	ATIM0_UEV	✓	✓	—
中断#44	ATIM0_CHA	✓	✓	—
中断#45	ATIM0_CHB	✓	✓	—
中断#46	ATIM0_BKETR	✓	✓	—
中断#47	ATIM3_UEV	✓	✓	—
中断#48	ATIM3_CHYA	✓	✓	—
中断#49	ATIM3_CHYB	✓	✓	—
中断#50	ATIM3_BK_ETR_OV_UD_CMP3	✓	✓	—
中断#51	DMA_CH0	✓	✓	—
中断#52	DMA_CH1	✓	✓	—
中断#53	DMA_CH2	✓	✓	—
中断#54	DMA_CH3	✓	✓	—
中断#55	DMA_CH4	✓	✓	—
中断#56	DMA_CH5	✓	✓	—
中断#57~63	NC	—	—	—

注：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。

5.7 中断结构

- 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 中断使能设置寄存器 NVIC_ISER.SETENA 置位与否，中断挂起设置寄存器 NVIC_ISPR.SETPEND 都会被置位，表示相应的外设中断源有中断产生。
- 只有中断使能设置寄存器 NVIC_ISER.SETENA 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，中断挂起设置寄存器 NVIC_ISPR.SETPEND 由硬件自动清除。
- 中断优先级寄存器 NVIC_IPR0-NVIC_IPR15 设置了 64 个中断源的优先级，0x00 优先级最高，0xE0 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。

5.8 软件基本操作

5.8.1 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，请参考外设模块各自的章节描述。

5.8.2 NVIC 中断使能和清除使能

Cortex-M4 处理器支持最多 128 个中断源，每个中断源都对应有一个中断使能位和清除使能位。

可以调用 CMSIS-Core 函数或配置寄存器使能中断和禁止中断

void NVIC_EnableIRQ(IRQn_Type IRQn) 使能外部中断

void NVIC_DisableIRQ(IRQn_Type IRQn) 禁止外部中断

5.8.3 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，硬件会自动清除挂起状态。

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

可以调用 CMSIS-Core 函数清除中断挂起

void NVIC_ClearPendingIRQ(IRQn_Type IRQn);

5.8.4 NVIC 中断优先级

设置 NVIC_IPRx 寄存器决定 IRQx 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-M4 处理器支持。

可以调用 CMSIS-Core 函数设置中断优先级

```
void NVIC_SetPriority (IRQn_Type IRQn, uint32_t priority);
```

5.8.5 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 NVIC_PRIMASK 实现。特殊寄存器 NVIC_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 NVIC_PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0（可设置值的最高优先级）。

可以通过多种方法编程 NVIC_PRIMASK 寄存器，若使用汇编语言，可以参考【中断屏蔽寄存器 (PRIMASK)】使用汇编指令设置和清除 NVIC_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
```

5.9 寄存器描述

5.9.1 寄存器列表

基地址：0xE000 E000

表 5-3 中断寄存器列表

寄存器名	偏移地址	复位值
NVIC_ISER0 (中断使能设置寄存器 0)	0x100	0x0000 0000
NVIC_ISER1 (中断使能设置寄存器 1)	0x104	0x0000 0000
NVIC_ICER0 (中断使能清除寄存器 0)	0x180	0x0000 0000
NVIC_ICER1 (中断使能清除寄存器 1)	0x184	0x0000 0000
NVIC_ISPR0 (中断挂起设置寄存器 0)	0x200	0x0000 0000
NVIC_ISPR1 (中断挂起设置寄存器 1)	0x204	0x0000 0000
NVIC_ICPR0 (中断挂起清除寄存器 0)	0x280	0x0000 0000
NVIC_ICPR1 (中断挂起清除寄存器 1)	0x284	0x0000 0000
NVIC_IPRx (中断优先级寄存器) (x=0~15)	0x400~0x43C	0x0000 0000
PRIMASK (中断屏蔽寄存器)	-	0x0000 0000

5.9.2 中断使能设置寄存器 0 (NVIC_ISER0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SETENA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SETENA[15:0]															

位	标记	位名	功能	读写
			写“1”置位, 写“0”无效	
			[0]: IRQ0	
b31~b0	SETENA	中断#0 到中断#31 使能设置	[1]: IRQ1	RW
			[2]: IRQ2	
				
			[31]: IRQ31	

5.9.3 中断使能设置寄存器 1 (NVIC_ISER1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SETENA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SETENA[15:0]															

位	标记	位名	功能	读写
			写“1”置位, 写“0”无效	
			[0]: IRQ32	
b31~b0	SETENA	中断#32 到中断#63 使能设置	[1]: IRQ33	RW
			[2]: IRQ34	
				
			[31]: IRQ63	

5.9.4 中断使能清除寄存器 0 (NVIC_ICER0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRENA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRENA[15:0]															

位	标记	位名	功能	读写
			写“1”清零, 写“0”无效	
			[0]: IRQ0	
b31~b0	CLRENA	中断#0 到中断#31 使能清除	[1]: IRQ1	RW
			[2]: IRQ2	
				
			[31]: IRQ31	

5.9.5 中断使能清除寄存器 1 (NVIC_ICER1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRENA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRENA[15:0]															

位	标记	位名	功能	读写
			写“1”清零, 写“0”无效	
			[0]: IRQ32	
b31~b0	CLRENA	中断#32 到中断#63 使能清除	[1]: IRQ33	RW
			[2]: IRQ34	
				
			[31]: IRQ63	

5.9.6 中断挂起设置寄存器 0 (NVIC_ISPR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SETPEND[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SETPEND[15:0]															

位	标记	位名	功能	读写
			写“1”置位, 写“0”无效	
			[0]: IRQ0	
b31~b0	SETPEND	中断#0 到中断#31 挂起设置	[1]: IRQ1	RW
			[2]: IRQ2	
				
			[31]: IRQ31	

5.9.7 中断挂起设置寄存器 1 (NVIC_ISPR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SETPEND[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SETPEND[15:0]															

位	标记	位名	功能	读写
			写“1”置位, 写“0”无效	
			[0]: IRQ32	
b31~b0	SETPEND	中断#32 到中断#63 挂起设置	[1]: IRQ33	RW
			[2]: IRQ34	
				
			[31]: IRQ63	

5.9.8 中断挂起清除寄存器 0 (NVIC_ICPR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRPEND[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRPEND[15:0]															

位	标记	位名	功能	读写
			写“1”清零，写“0”无效	
			[0]: IRQ0	
b31~b0	CLRPEND	中断#0 到中断#31 挂起清除	[1]: IRQ1	RW
			[2]: IRQ2	
				
			[31]: IRQ31	

5.9.9 中断挂起清除寄存器 1 (NVIC_ICPR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRPEND[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRPEND[15:0]															

位	标记	位名	功能	读写
			写“1”清零，写“0”无效	
			[0]: IRQ32	
b31~b0	CLRPEND	中断#32 到中断#63 挂起清除	[1]: IRQ33	RW
			[2]: IRQ34	
				
			[31]: IRQ63	

5.9.10 中断优先级寄存器 (NVIC_IPRx) (x=0~15)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IPRx[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IPRx[15:0]															

位	标记	位名	功能	读写
中断#0 到中断#63 的优先级				
[31:29]: 中断# x*4+3 的优先级				
[23:21]: 中断# x*4+2 的优先级				
[15:13]: 中断# x*4+1 的优先级				
[7:5]: 中断# x*4+0 的优先级				
其中, 000 优先级最高, 111 优先级最低				
b31~b0	IPRx	中断优先级		RW

5.9.11 中断屏蔽寄存器 (PRIMASK)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															PRIMASK

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
置位后, 除了 NMI 和硬件错误异常外的其它中断都会被屏蔽掉;				
清零后, 所有异常和中断都不会被屏蔽掉;				
b0	PRIMASK	中断屏蔽	该特殊寄存器需要通过 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。在处理对时间敏感的应用时, 需要操作 PRIMASK 寄存器。	RW

6 通用输入输出端口控制器 (GPIO)

6.1 简介

GPIO 是 MCU 最基本的输入输出通道。本产品包含四组输入输出端口，其中 PA、PB、PC 为 16 位，PD 为 4 位。实际可用的 GPIO 端口数量请参考相应的规格书以获得更多信息。

每个管脚都有相关的控制和配置寄存器，以满足特定应用的需求。通过配置相应的寄存器，管脚可以被配置为以下三类工作模式：模拟外设的输入输出、数字外设的输入输出、GPIO 的输入输出。

6.2 主要特性

- 功能复用：每只管脚可配置为多种数字或模拟功能
- 输入模式：模拟输入、高阻输入、上拉输入、下拉输入
- 输出模式：模拟输出、推挽输出、开漏输出（可使能上拉）
- 信号输入：输入信号到达数据输入寄存器（GPIO_IN）及片内外设
- 信号输出：输出信号来源为数据输出寄存器（GPIO_OUT）或片内外设
- 原子操作：位操作寄存器（GPIO_BSRR、GPIO_BRR）实现 IO 操作原子化
- 中断类型：上升沿中断、下降沿中断、高电平中断、低电平中断
- 辅助功能：输出各种内部时钟、片内外设输入信号重定向

6.3 功能描述

6.3.1 端口电路框图

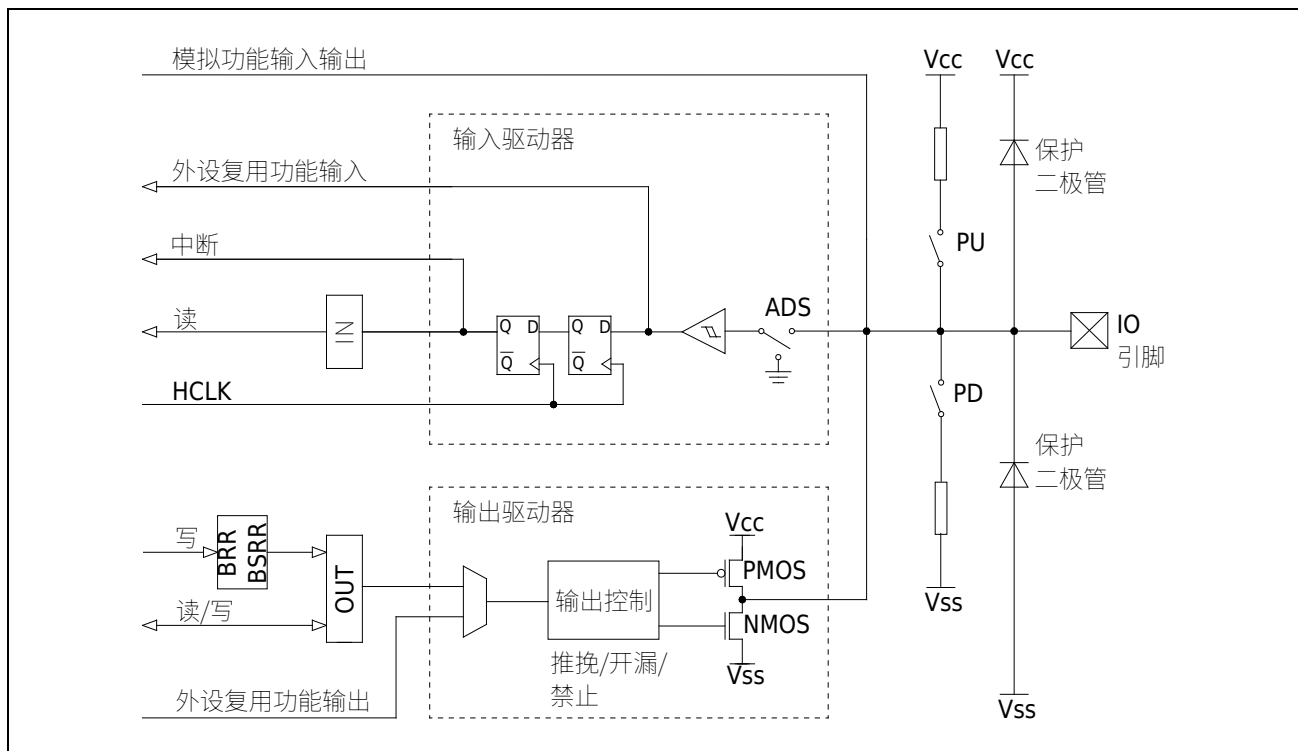


图 6-1 通用端口电路示意图

6.3.2 端口模式配置

通过控制寄存器，可以将端口配置为多种工作模式，其真值表如下方所示：

表 6-1 端口功能配置表

管脚模式	IN	ADS	AFRL/AFRH	DIR	OD	OUT	PU	PD
模拟	0	1	-	-	-	-	-	-
高阻输入	X	0	0	1	-	-	0	0
上拉输入	1	0	0	1	-	-	1	-
下拉输入	0	0	0	1	-	-	0	1
开漏输出	X	0	0	0	1	1	0	0
输出 0	0	0	0	0	0	0	-	-
输出 1	0	0	0	0	0	1	-	-
外设输出	X	0	>0	0	0	-	-	-
外设上拉输入	X	0	>0	1	0	-	1	-
外设下拉输入	X	0	>0	1	0	-	0	1

注：-代表任意值；X 代表不确定

6.3.3 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，大多数 I/O 端口被配置为模拟输入无上下拉状态；其它管脚的状态如下：

- SWCLK(PA14): 数字输入+上拉
- SWDIO(PA13): 数字输入+上拉
- PB11: 数字输入+下拉

6.3.4 端口模拟功能

当需要使用模拟外设（ADC、LVD.....）的相应功能时，需要配置端口为模拟功能。实现方法为设置 GPIO_ADS 寄存器相应的位为 1，使片内模拟外设连接到该端口。

在该配置下，端口具有如下特性：

- 模拟外设连接到端口
- 端口数字功能关闭：不能输出电平；读取管脚电平恒为 0；无内置上拉及下拉

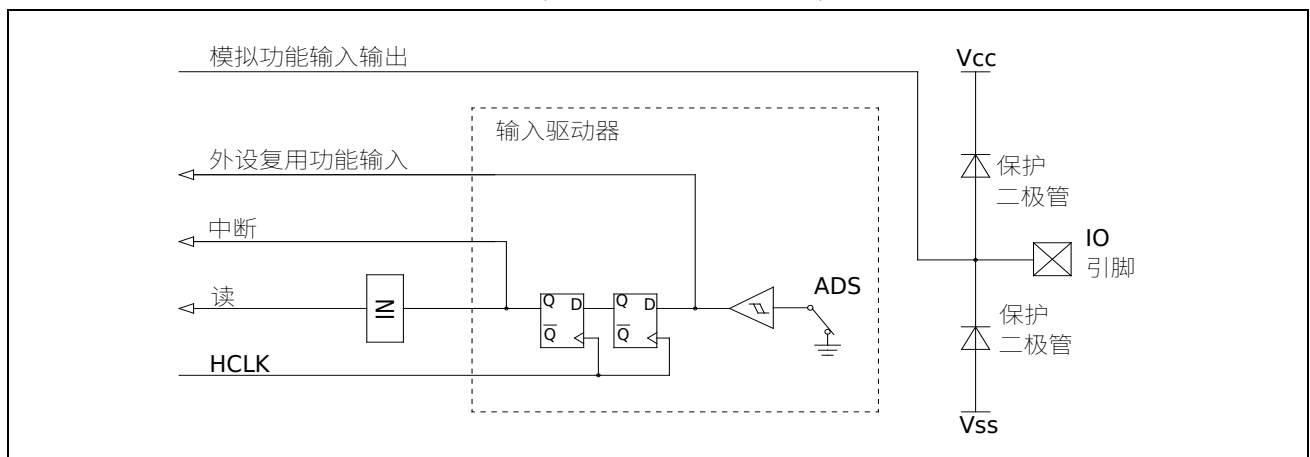


图 6-2 高阻-模拟配置示意图

6.3.5 端口通用输入输出功能

当需要使用端口的通用输入输出功能时，需要配置端口为数字功能并设置数字复用功能寄存器的值为 0。实现方法为设置 GPIO_ADS 寄存器相应的位为 0，设置 GPIO_AFRL 和 GPIO_AFRH 寄存器相应的位为 0。在该配置下，端口具有如下特性：

- 输入模式：可使能上拉或下拉
- 推挽输出：支持位操作
- 开漏输出：可使能上拉；支持位操作

每个 GPIO 具有 6 个控制寄存器，其功能描述如下：

- GPIO_ADS 寄存器可配置端口为数字端口
- GPIO_DIR 寄存器可配置端口为输入或输出
- GPIO_PUPD.PU 寄存器可使能端口的上拉
- GPIO_PUPD.PD 寄存器可使能端口的下拉
- GPIO_OD 寄存器可配置端口的输出类型为推挽或开漏
- GPIO_DR 寄存器输出端口驱动能力配置

每个 GPIO 具有 2 个数据寄存器，其功能描述如下：

- GPIO_IN 寄存器存储通过 I/O 输入的数据，只可进行读操作
- GPIO_OUT 寄存器存储待输出数据，可对其进行读/写访问

注意：

对 GPIO_OUT 寄存器进行【读-修改-写】的过程不是原子操作，不建议采用该方式修改端口输出电平。

每个 GPIO 有 2 个位操作寄存器，其功能描述如下：

- GPIO_BSRR 寄存器的 BS 位可将 GPIO_OUT 相应的比特置 1
 寄存器的 BR 位可将 GPIO_OUT 相应的比特清 0
- GPIO_BRR 寄存器位可将 GPIO_OUT 相应的比特清 0

注意：

通过位操作寄存器对 GPIO_OUT 的值进行修改的过程为原子操作，建议采用位操作的方式修改端口输出电平。

6.3.5.2 输出模式

当 GPIO 被配置为输出模式下，其简化框图如下所示：

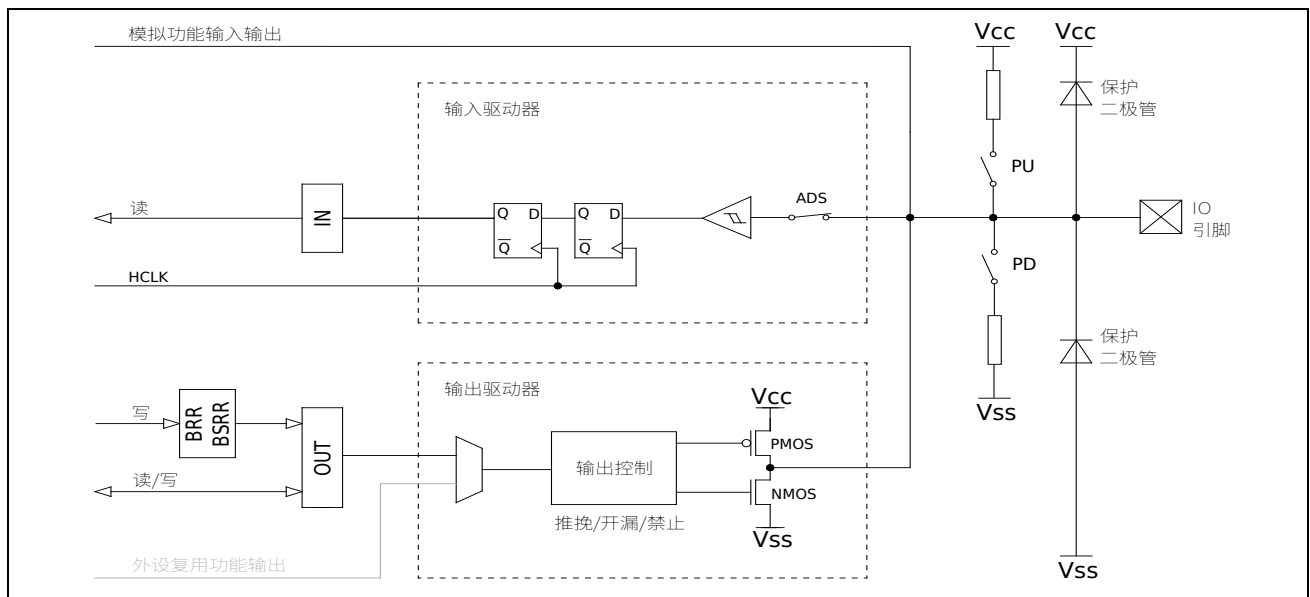


图 6-5 输出配置示意图

每一个端口可以通过向 GPIO_BRR、GPIO_BSRR 或 GPIO_OUT 写入数据以控制 IO 口的输出电平。

GPIO_BRR、GPIO_BSRR 为位操作寄存器，每次可修改同一组端口中 IO 的数量为 1~16 个，且同一组端口中未被修改的 IO 保持其状态不变，使用上述两个寄存器对 IO 进行修改的过程为原子操作。对寄存器 GPIO_OUT 的写入值即为每个位所对应的输出 PIN 脚的输出值。

6.3.6 端口复用功能

当需要使用数字外设（GPIO、USART、SPI.....）的相应功能时，需要配置端口为数字功能。操作方法为设置 GPIO_ADS 相应的位为 0 并配置 GPIO_AFRL 或 GPIO_AFRH，以实现将特定的片内外设连接到端口。

在该配置下，端口具有如下特性：

- 输入模式：可使能上拉或下拉；外部信号输入到片内外设；GPIO_IN 可读取外部信号可以产生端口中断
- 输出模式：可选择开漏（可使能上拉）或推挽；片内外设输出信号到端口；不可以产生端口中断

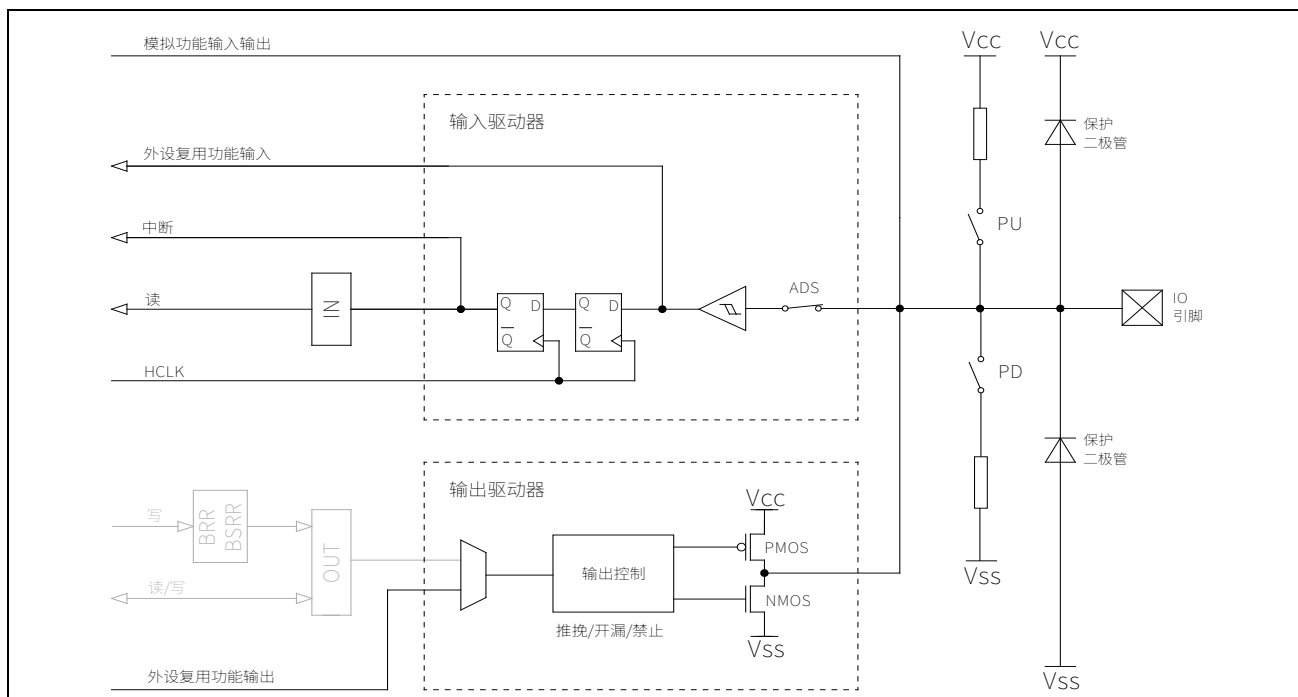


图 6-6 复用功能配置示意图

本芯片每一个端口的数字复用功能如下表所示。

表 6-2 PINMUX 复用功能

管脚 名称	数字功能					
	AF0	AF1	AF2	AF3	AF4	AF5
PD03	GPIO	ATIM0_BK	CTIM0_CH0	CTIM1_TOGP	ATIM3_BK	USART2_TXD
PC13	GPIO	SPI0_SCK	CTIM2_CH0	I2C1_SCL	ATIM3_CH0A	USART3_NCTS
PC14	GPIO	SPI0_MISO	CTIM2_CH1	I2C1_SDA	ATIM3_CH1A	USART3_NRTS
PC15	GPIO	SPI0_MOSI	CTIM2_ETR		ATIM3_CH2A	USART3_SCK
PD00	GPIO	SPI0_CS	CTIM0_CH2	I2C0_SCL	ATIM3_CH0B	I2C1_SCL
PD01	GPIO	LVD_OUT	CTIM0_CH3	I2C0_SDA	ATIM3_CH2B	I2C1_SDA
PC00	GPIO	ATIM0_BK	CTIM1_TOGP	CTIM1_ETRTOG	SPI0_MISO	
PC01	GPIO	ATIM0_CH0A	CTIM1_TOGN		SPI0_MOSI	
PC02	GPIO	ATIM0_CH0B		SPI0_CS	SPI1_MISO	USART0_NCTS
PC03	GPIO	ATIM0_GATE		SPI0_SCK	SPI1_MOSI	USART0_NRTS
PA00	GPIO	ATIM0_ETR	CTIM0_CH0	CTIM0_ETR	VC0_OUT	USART1_NCTS
PA01	GPIO	MCO	CTIM0_CH1	SPI1_CS	USART2_NCTS	USART1_NRTS
PA02	GPIO	CTIM1_TOGN	CTIM0_CH2	VC1_OUT	USART3_NRTS	USART1_TXD
PA03	GPIO	LVD_OUT	CTIM0_CH3	SPI1_SCK	USART3_TXD	USART1_RXD
PA04	GPIO	ATIM3_CH0A	PCLK0_OUT	SPI0_CS	USART3_RXD	USART1_SCK
PA05	GPIO	ATIM3_CH1A	CTIM0_CH0	CTIM0_ETR	USART3_NCTS	SPI0_SCK
PA06	GPIO	ATIM3_CH2A	USART2_RXD	CTIM1_CH0	ATIM3_BK	SPI0_MISO
PA07	GPIO	ATIM3_CH0B	USART3_TXD	CTIM1_CH1	USART2_NRTS	SPI0_MOSI
PC04	GPIO	ATIM3_CH1B		I2C0_SCL		USART0_SCK
PC05	GPIO	ATIM3_CH2B	CTIM0_TOGP	I2C0_SDA	ATIM0_ETR	
PB00	GPIO	ATIM3_CH1B	CTIM2_ETR	CTIM1_CH2	ATIM0_CH0A	USART3_SCK
PB01	GPIO	ATIM3_CH2B	CTIM2_TOGP	CTIM1_CH3	ATIM0_CH0B	PCLK1_OUT
PB02	GPIO	CTIM0_CH3	CTIM2_TOGN	ATIM3_GATE	ATIM0_GATE	USART2_RXD
PB10	GPIO	CTIM0_CH2			I2C1_SCL	USART2_TXD
PB12	GPIO	I2C0_SCL	ATIM3_BK	CTIM1_ETRTOG	SPI1_CS	USART2_SCK
PB13	GPIO	I2C0_SDA	ATIM3_CH0B		SPI1_SCK	USART2_NCTS
PB14	GPIO		ATIM3_CH1B	CTIM1_TOGP	SPI1_MISO	USART2_NRTS
PB15	GPIO	MCO	ATIM3_CH2B	CTIM1_TOGN	SPI1_MOSI	
PC06	GPIO	ATIM3_ETR	CTIM2_CH0	CTIM1_CH0		USART2_NCTS
PC07	GPIO		CTIM2_CH1	CTIM1_CH1	SPI1_SCK	USART2_NRTS
PC08	GPIO	I2C1_SCL	CTIM2_CH2	CTIM1_CH2		USART2_SCK
PC09	GPIO	I2C1_SDA	CTIM2_CH3	CTIM1_CH3		
PA08	GPIO	CTIM0_TOGP	ATIM3_CH0A	CTIM1_ETRTOG	USART3_RXD	USART0_SCK
PA09	GPIO	CTIM0_TOGN	ATIM3_CH1A	ATIM0_BK	CTIM1_CH2	USART0_TXD

管脚 名称	数字功能					
	AF0	AF1	AF2	AF3	AF4	AF5
PA10	GPIO	I2C1_SCL	ATIM3_CH2A	ATIM0_CH0A	CTIM1_CH3	USART0_RXD
PA11	GPIO	VC0_OUT	ATIM3_GATE	ATIM0_CH0B	USART1_NCTS	USART0_NCTS
PA12	GPIO	VC1_OUT	ATIM3_ETR	ATIM0_GATE	USART1_NRTS	USART0_NRTS
PA13	GPIO	ATIM3_BK	CTIM2_CH2	ATIM0_ETR	USART1_RXD	I2C0_SCL
PA14	GPIO	CTIM0_CH1	CTIM2_CH3	USART2_SCK	USART1_TXD	I2C0_SDA
PA15	GPIO	SPI0_CS	CTIM1_CH0	CTIM1_ETR	USART1_SCK	USART0_TXD
PC10	GPIO		USART3_TXD		ATIM3_CH0A	ATIM0_CH0A
PC11	GPIO	CTIM0_TOGN	USART3_RXD		ATIM3_CH1A	ATIM0_CH0B
PC12	GPIO		USART3_NCTS	CTIM2_ETR	ATIM3_CH2A	ATIM0_GATE
PD02	GPIO		USART3_NRTS	CTIM1_ETR	ATIM3_CH0B	ATIM0_ETR
PB03	GPIO	CTIM0_CH1	USART0_RXD	I2C1_SDA	ATIM3_CH1B	SPI0_SCK
PB04	GPIO	CTIM1_CH0	CTIM2_TOGP	PCLK0_OUT	ATIM3_CH2B	SPI0_MISO
PB05	GPIO	CTIM1_CH1	CTIM2_TOGN	USART3_SCK	USART2_TXD	SPI0_MOSI
PB06	GPIO	USART0_TXD	CTIM2_CH0	SPI1_MISO	I2C0_SCL	USART2_RXD
PB07	GPIO	USART0_RXD	CTIM2_CH1	SPI1_MOSI	I2C0_SDA	PCLK1_OUT
PB11	GPIO					
PB08	GPIO	CTIM1_ETRTOG	CTIM2_CH2		I2C0_SCL	ATIM0_BK
PB09	GPIO	SPI1_CS	CTIM2_CH3	I2C0_SDA	I2C1_SDA	

表 6-3 IOAUX1/2 复用功能

ATIM0CAP0[2:0]	ATIM0_CHA	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA00	PA08	PB00
CTIM0CAP3[6:4]	CTIM0_CH3	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA01	PA09	PB01
CTIM1CAP3[10:8]	CTIM1_CH3	USART2_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA02	PA10	PB02
CTIM2CAP3[14:12]	CTIM2_CH3	USART3_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA03	PA11	PB03
ATIM0ETR[2:0]	ATIM0_ETR	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA04	PA12	PB04
CTIM0ETR[6:4]	CTIM0_ETR	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA05	PA13	PB05
CTIM1ETR[10:8]	CTIM1_ETR	USART2_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA06	PA14	PB06
CTIM2ETR[14:12]	CTIM2_ETR	USART3_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA07	PA15	PB07

6.3.7 端口时钟输出

6.3.7.1 输出 PCLK0

配置 GPIO 的寄存器，可以输出 PCLK0 的 2 分频输出，下方示例从 PA04 输出 PCLK0：

- Step1. 设置系统控制模块的 SYSCTRL_IOAUX0.PCLK0DIV 为 2，选择 2 分频；
- Step2. 设置 GPIOA_ADS.PIN[4]为 0，配置 PA04 为数字端口；
- Step3. 设置 GPIOA_DIR.PIN[4]为 0，配置 PA04 为输出模式；
- Step4. 设置 GPIOA_AFRL.AFSEL4 为 2，配置 PA04 的复用功能为 PCLK0_OUT。

6.3.7.2 输出 MCO

配置 GPIO 的寄存器，可以输出 RCH/RCL/XTH/PLL/RC10/RC150K，下方示例从 PA01 输出 RCH/8：

- Step1. 设置系统控制模块的 SYSCTRL_IOAUX0.MCOSEL 为 0，选择 RCH；
- Step2. 设置系统控制模块的 SYSCTRL_IOAUX0.MCODIV 为 2，选择 8 分频；
- Step3. 设置 GPIOA_ADS.PIN[1]为 0，配置 PA01 为数字端口；
- Step4. 设置 GPIOA_DIR.PIN[1]为 0，配置 PA01 为输出模式；
- Step5. 设置 GPIOA_AFRL.AFSEL1 为 1，配置 PA01 的复用功能为 MCO。

6.3.8 端口外部中断

当端口被配置为数字功能（GPIO_ADS 为 0）且工作于输入模式（GPIO_DIR 为 1）时，每一个端口均具有中断功能。每个端口支持 4 种类型的中断：上升沿中断、下降沿中断、高电平中断、低电平中断；对应的中断使能寄存器分别为：GPIO_RIE、GPIO_FIE、GPIO_HIE、GPIO_LIE。当中断触发时，通过查询中断状态寄存器（GPIO_IFR）即可判断是哪一个端口触发了中断，通过中断清除寄存器（GPIO_ICR）就可以清除相应的中断状态标志位。

GPIO 外部中断相关的简化框图如下所示：

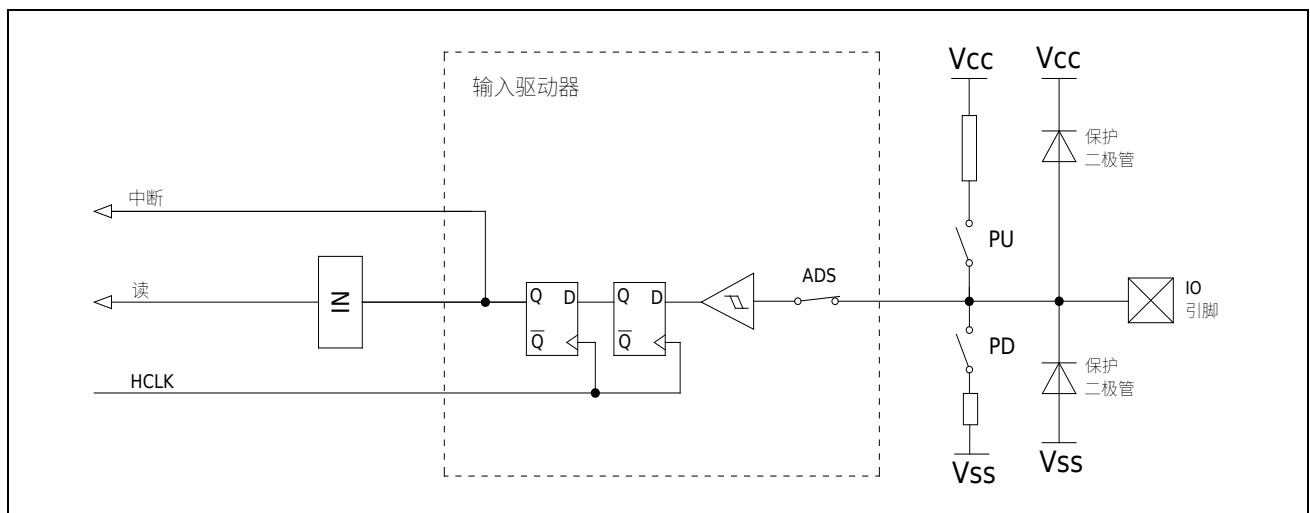


图 6-7 端口外部中断示意图

6.4 寄存器描述

6.4.1 寄存器列表

表 6-4 GPIO 基地址列表

名称	基地址	描述
GPIOA	0x4002 4000	GPIOA 基地址
GPIOB	0x4002 4400	GPIOB 基地址
GPIOC	0x4002 4800	GPIOC 基地址
GPIOD	0x4002 4C00	GPIOD 基地址

注：

GPIOA、GPIOB、GPIOC 有 16 比特有效位，GPIOD 有 4 比特有效位（PD00-PD03）。对不存在的位，写操作无效，读出值始终为 0。

表 6-5 GPIO 寄存器列表

寄存器名	偏移地址	复位值
端口数模配置寄存器（GPIO_ADS）	0x00	(PA)0x0000 9FFF (PB)0x0000 F7FF (PC)0x0000 FFFF (PD)0x0000 000F
端口输入输出方向配置寄存器（GPIO_DIR）	0x04	(PA)0x0000 FFFF (PB)0x0000 FFFF (PC)0x0000 FFFF (PD)0x0000 000F
端口输出类型寄存器（GPIO_OD）	0x08	0x0000 0000
端口输出驱动能力寄存器（GPIO_DR）	0x10	0x0000 0000
端口上拉下拉寄存器（GPIO_PUPD）	0x18	(PA)0x0000 6000 (PB)0x0800 0000 (PC)0x0000 0000 (PD)0x0000 0000
端口数据输入寄存器（GPIO_IN）	0x1C	0x0000 xxxx
端口数据输出寄存器（GPIO_OUT）	0x20	0x0000 xxxx
端口复位寄存器（GPIO_BRR）	0x24	0x0000 0000
端口置位复位寄存器（GPIO_BSRR）	0x28	0x0000 0000
端口复用功能低位寄存器（GPIO_AFR_L）	0x30	0x0000 0000
端口复用功能高位寄存器（GPIO_AFR_H）	0x34	0x0000 0000
端口高电平中断使能配置寄存器（GPIO_HIE）	0x38	0x0000 0000
端口低电平中断使能配置寄存器（GPIO_LIE）	0x3C	0x0000 0000
端口上升沿中断使能配置寄存器（GPIO_RIE）	0x40	0x0000 0000
端口下降沿中断使能配置寄存器（GPIO_FIE）	0x44	0x0000 0000
端口中断状态寄存器（GPIO_IFR）	0x48	0x0000 0000
端口中断清除寄存器（GPIO_ICR）	0x4C	(PA)0x0000 FFFF (PB)0x0000 FFFF (PC)0x0000 FFFF (PD)0x0000 000F

6.4.2 端口数模配置寄存器 (GPIO_ADS)

复位值: (PA) 0x0000 9FFF

(PB) 0x0000 F7FF

(PC) 0x0000 FFFF

(PD) 0x0000 000F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口数字功能或模拟功能控制	0: 将端口配置为数字功能 1: 将端口配置为模拟功能										RW		

6.4.3 端口输入输出方向配置寄存器 (GPIO_DIR)

复位值: (PA) 0x0000 FFFF

(PB) 0x0000 FFFF

(PC) 0x0000 FFFF

(PD) 0x0000 000F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口输入输出方向控制	0: 将端口配置成输出 1: 将端口配置成输入										RW		

6.4.4 端口输出类型寄存器 (GPIO_OD)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口输出方式控制位	0: 推挽输出 (复位值) 1: 开漏输出										RW		

6.4.5 端口输出驱动能力寄存器 (GPIO_DR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口输出驱动能力控制位	0: 高驱动能力 (复位值) 1: 低驱动能力										RW		

6.4.6 端口上拉下拉寄存器 (GPIO_PUPD)

复位值: (PA)0x0400 6000

(PB)0x0000 0000

(PC)0x0000 0000

(PD)0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PD															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PU															

6.4.7 端口数据输入寄存器 (GPIO_IN)

复位值: 0x0000 XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记		位名			功能								读写	
b31~b16	Res		保留位			保留，必须保持复位值									
b15~b0	PIN		端口输入数据			-								RO	

6.4.8 端口数据输出寄存器 (GPIO_OUT)

复位值: 0x0000 XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位		标记		位名		功能								读写	
b31~b16		Res		保留位		保留，必须保持复位值									
b15~b0		PIN		端口输出数据		-								RW	

6.4.9 端口复位寄存器 (GPIO_BRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口位清零控制	0: 不影响 GPIO_OUT 寄存器相应的比特 1: 设置 GPIO_OUT 寄存器相应的比特为 0										WO		

6.4.10 端口置位复位寄存器 (GPIO_BSRR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BR															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BS															
位	标记	位名	功能										读写		
b31~b16	BR	端口位清零控制	0: 不影响 GPIO_OUT 寄存器相应的比特 1: 设置 GPIO_OUT 寄存器相应的比特为 0										WO		
b15~b0	BS	端口位置位控制	0: 不影响 GPIO_OUT 寄存器相应的比特 1: 设置 GPIO_OUT 寄存器相应的比特为 1 注: BR 和 BS 同时发生时以 BS 优先。										WO		

6.4.11 端口复用功能低位寄存器 (GPIO_AFRL)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AFSEL7				AFSEL6				AFSEL5				AFSEL4			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AFSEL3				AFSEL2				AFSEL1				AFSEL0			

位	标记	位名	功能	读写
b31~b0	AFSELy y = 0-7	端口复用功能控制	0000: GPIO	RW
			0001: AF1	
			0010: AF2	
			0011: AF3	
			0100: AF4	
			0101: AF5	
			0110-1111: 保留位	
			注 1: AF1 - AF5 的具体功能详见【端口复用功能】章节	

6.4.12 端口复用功能高位寄存器 (GPIO_AFRH)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AFSEL15				AFSEL14				AFSEL13				AFSEL12			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AFSEL11				AFSEL10				AFSEL9				AFSEL8			

位	标记	位名	功能	读写
b31~b0	AFSELy y = 8-15	端口复用功能控制	0000: GPIO	RW
			0001: AF1	
			0010: AF2	
			0011: AF3	
			0100: AF4	
			0101: AF5	
			0110-1111: 保留位	
			注 1: AF1 - AF5 的具体功能详见【端口复用功能】章节	

6.4.13 端口高电平中断使能配置寄存器 (GPIO_HIE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口高电平中断使能控制	0: 禁止相应端口的高电平中断 1: 使能相应端口的高电平中断										RW		

6.4.14 端口低电平中断使能配置寄存器 (GPIO_LIE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口低电平中断使能控制	0: 禁止相应端口的低电平中断 1: 使能相应端口的低电平中断										RW		

6.4.15 端口上升沿中断使能配置寄存器 (GPIO_RIE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口上升沿中断使能控制	0: 禁止相应端口的上升沿中断 1: 使能相应端口的上升沿中断										RW		

6.4.16 端口下降沿中断使能配置寄存器 (GPIO_FIE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口下降沿中断使能控制	0: 禁止相应端口的下降沿中断 1: 使能相应端口的下降沿中断										RW		

6.4.17 端口中断状态寄存器 (GPIO_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口中断状态标志	0: 未发生符合条件的中断 1: 已发生符合条件的中断										RO		

6.4.18 端口中断清除寄存器 (GPIO_ICR)

复位值: (PA)0x0000 FFFF

(PB)0x0000 FFFF

(PC)0x0000 FFFF

(PD)0x0000 000F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	PIN	端口中断标志清除	0: 清除相应的中断标志位 1: 无功能										R1W0		

7 I2C 总线 (I2C)

7.1 概述

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。

7.2 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) /高速 (1Mbps) 三种工作速率
- 从机支持 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

7.3 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。

7.3.1 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)。

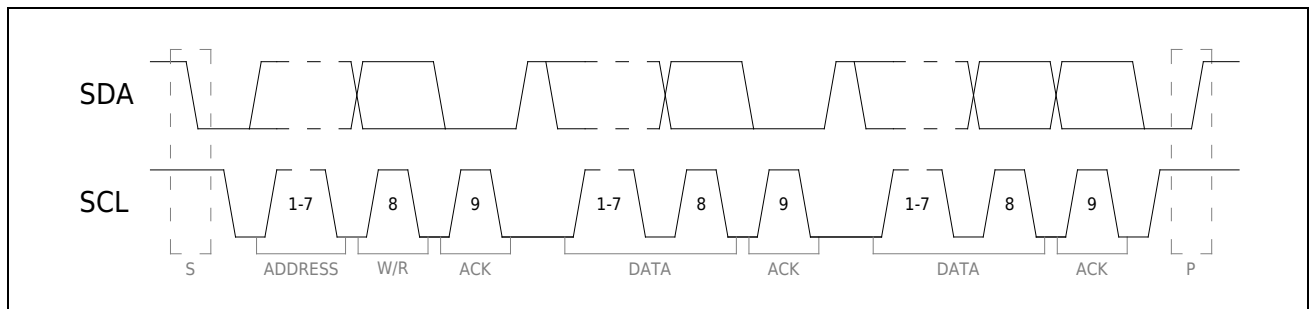


图 7-1 I2C 传输协议

■ 起始信号、重复起始信号、停止信号

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，表明总线上产生了起始信号。

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。

当 SCL 线为高时，SDA 线上出现由低到高的信号，被定义为停止信号。主机向总线发出停止信号结束数据传送。

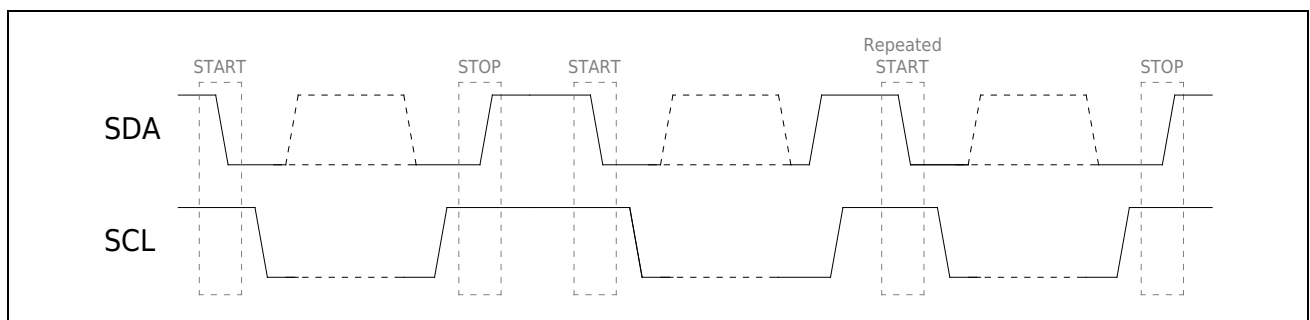


图 7-2 START 和 STOP 条件

■ 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通过在第 9 个 SCL 时钟周期将 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。

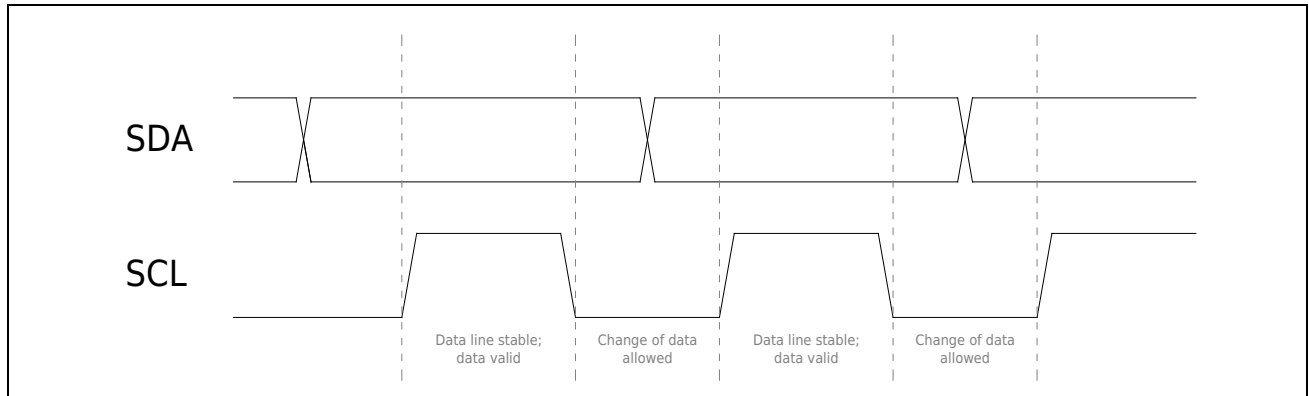


图 7-3 I2C 总线上位传输

7.3.2 I2C 总线上的应答

每传输一个字节，后面跟随一个应答位。每个字节发送完毕后，发送端将释放 SDA，而接收端需要通过拉低 SDA 线来回应发送端。ACK 为一个低电平信号，当时钟信号为高时，SDA 保持低电平则表明接收端已成功接收到发送端的数据。

当主机作为发送器件时，如果从机上产生无响应信号(NACK)，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，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

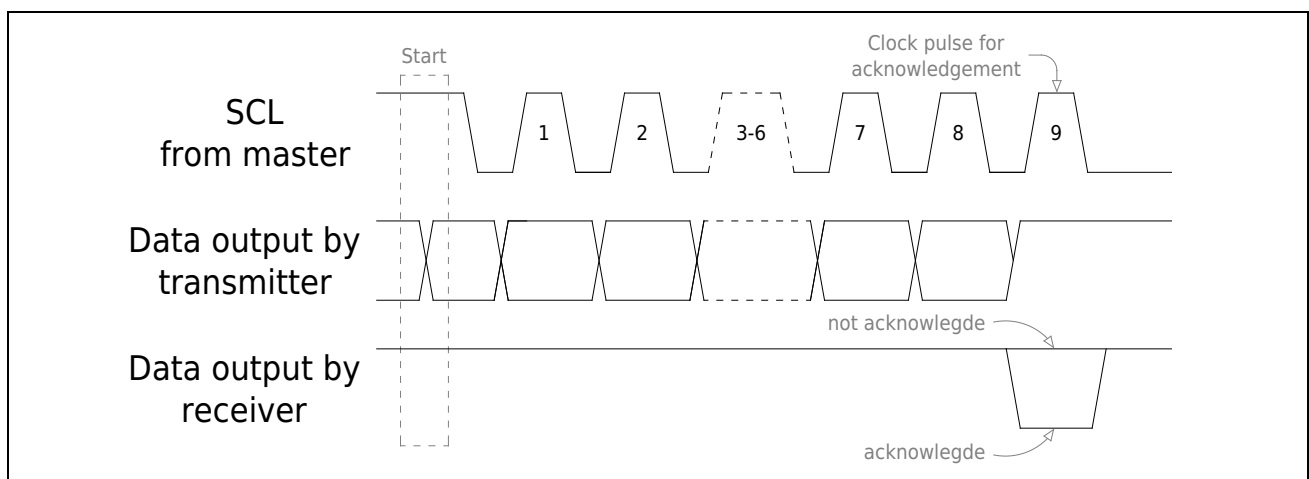


图 7-4 I2C 总线上应答信号

7.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：SCL 线上的同步和 SDA 线上的仲裁。

■ 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。如果从机希望主机降低传送速度可以通过将 SCL 主动拉低延长其低电平时间来通知主机，当主机在准备下一次传送时发现 SCL 的电平被拉低时则进行等待，直到从机完成操作并释放 SCL 线的控制权。

■ 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。

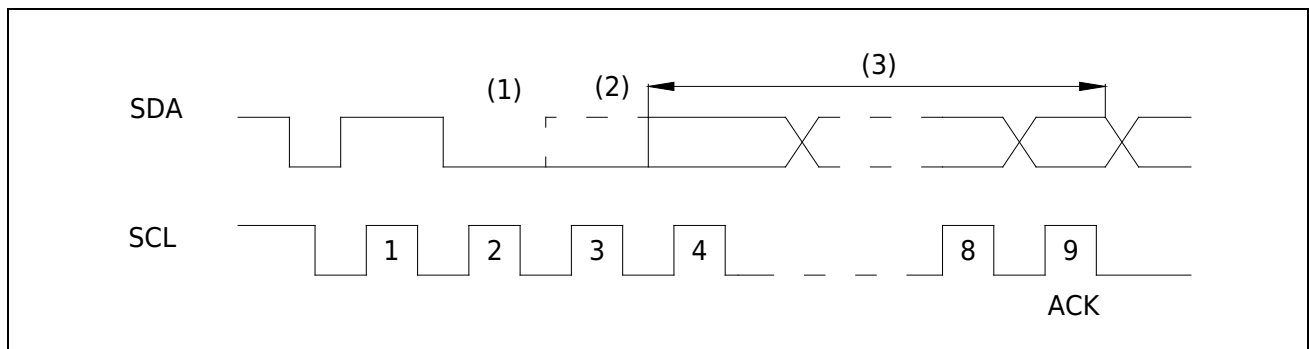


图 7-5 I2C 总线上的仲裁

- 1) 器件 1 和器件 2 同时发送串行数据；
- 2) 第三个 SCL 周期中，器件 1 发送逻辑 1，器件 2 发送逻辑 0，器件 2 的逻辑 0 将拉低 SDA，使器件 1 的逻辑 1 被撤销，进而引起器件 1 仲裁丢失，进入从接收模式；
- 3) 器件 1 仲裁失败进入从机接收模式后，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。器件 2 赢得仲裁后，SDA 上的数据传输由器件 2 来启动。

7.4 功能描述

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 组件需要使用到管脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。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。

7.4.1 功能框图

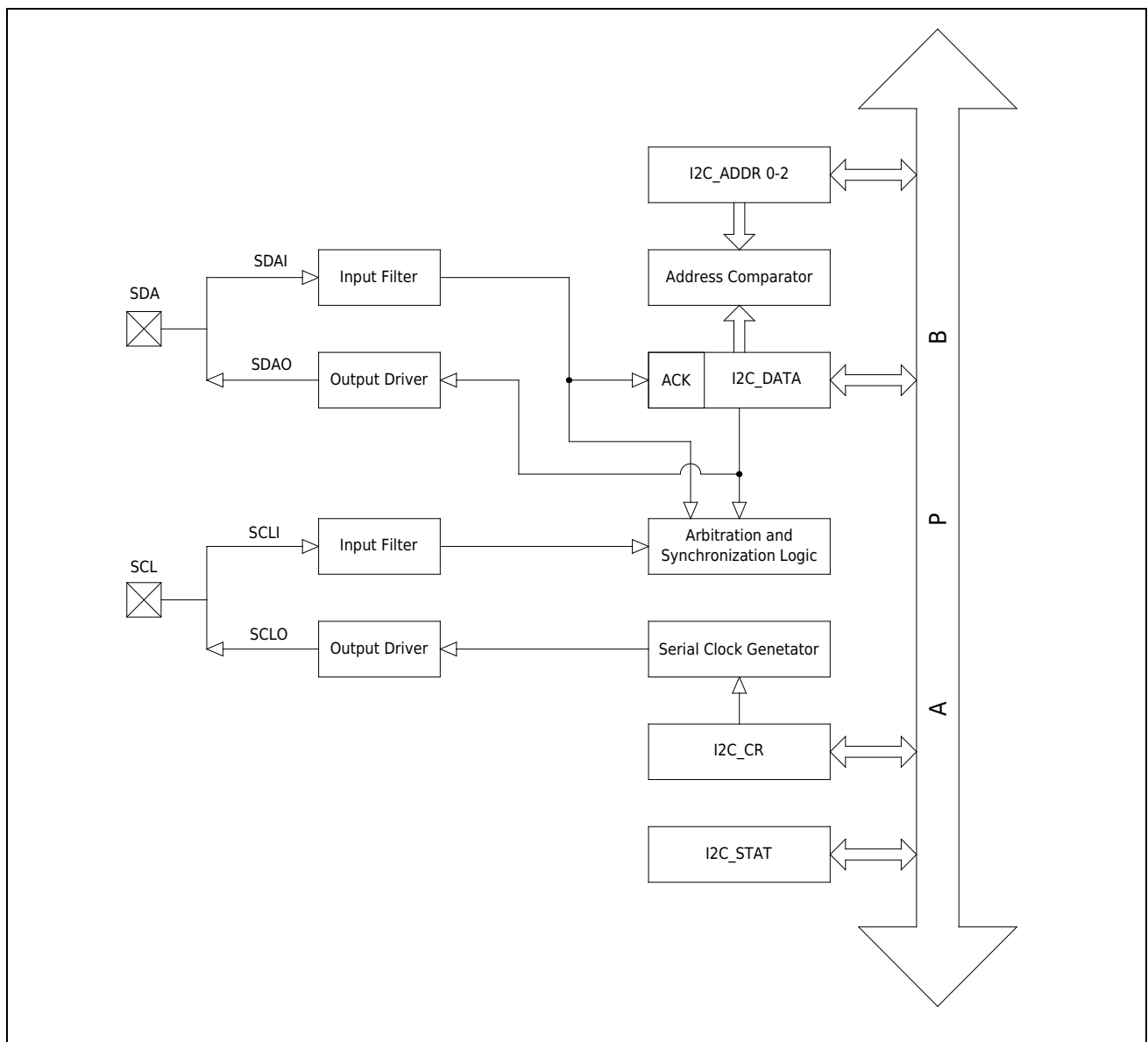


图 7-6 I2C 功能模块图

7.4.2 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 $f_{SCL} = f_{PCLK} / 8 / (I2C_BRR + 1)$ ，其中 I2C_BRR 应大于 0。

下表列出了 PCLK 频率与 I2C_BRR 组合时，SCL 信号的输出频率值。

表 7-1 I2C 时钟信号波特率

PCLK (KHz)	I2C_BRR						
	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

7.4.3 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.FLT 的值可变滤波的强度。

当本模块作为主机时，如果 I2C_BRR 的值小于等于 9，应设置 I2C_CR.FLT 为 1；如果 I2C_BRR 的值大于 9，应设置 I2C_CR.FLT 为 0。

当本模块作为从机时，如果 PCLK 与 SCL 频率比值小于等于 40 时，应设置 I2C_CR.FLT 为 1；如果 PCLK 与 SCL 频率比值大于 40 时，应设置 I2C_CR.FLT 为 0。

7.4.4 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与进行比较。根据 I2C_ADDR0 寄存器的 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。

7.4.5 应答标志位

I2C_CR.AA 标志位为应答标志位。当 AA 位为 1 时，I2C 模块收到数据后回应答位，当 AA 位为 0 时，I2C 模块收到数据后回非应答位。

7.4.6 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。

7.4.7 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 而在高速模式下可达 400Kbps，在超高速模式下可达 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_BRR 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 收到从机应答位 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。

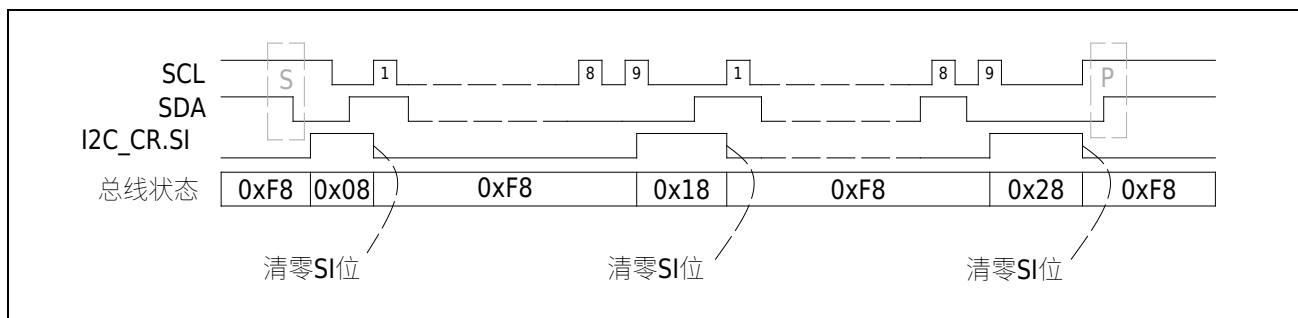


图 7-7 主发送模式数据同步图

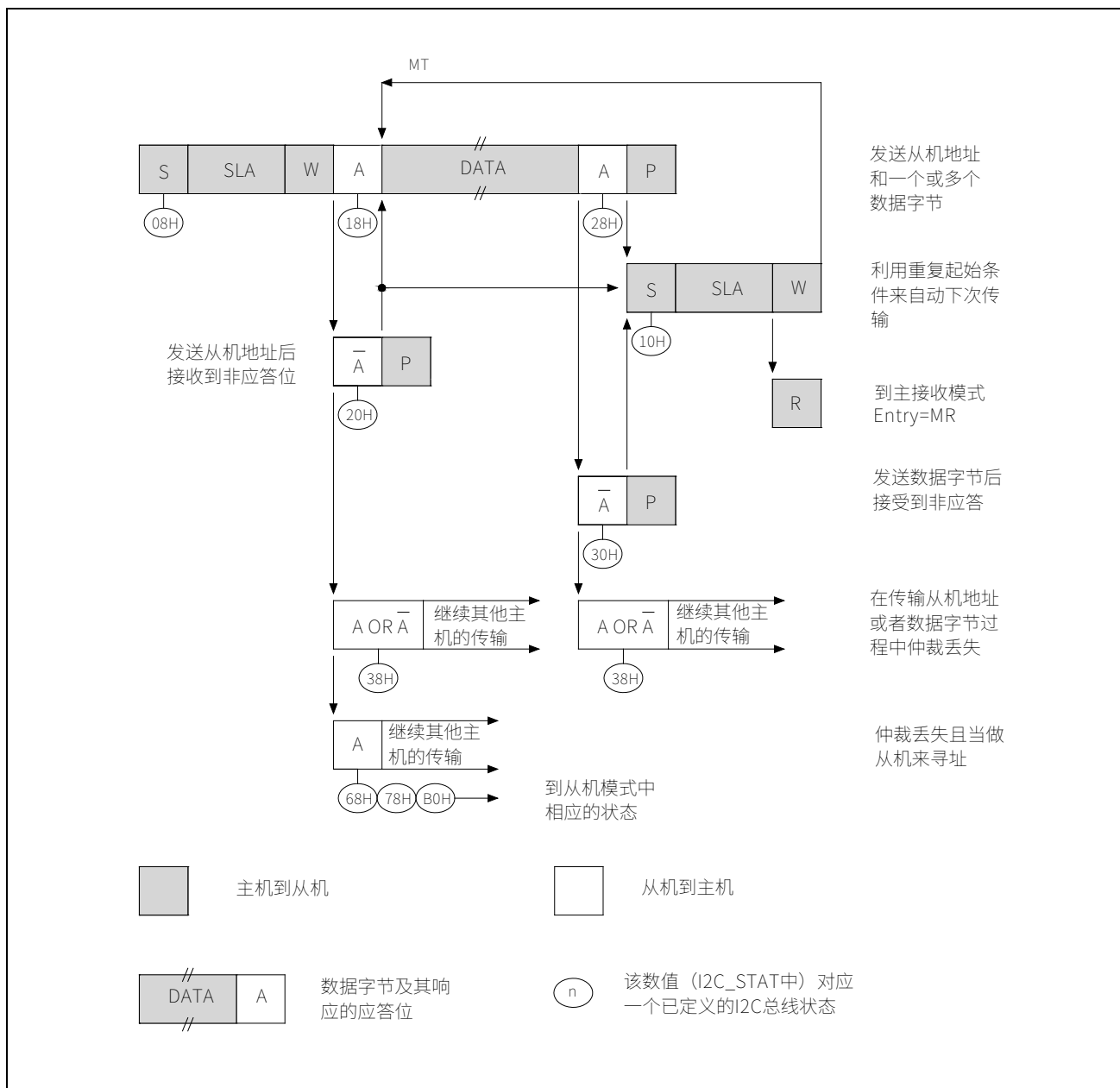


图 7-8 I2C 主机发送状态图

■ 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，从机地址为 0。当 I2C_ADDR0.GC 及 I2C_CR.AA 都为置 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。

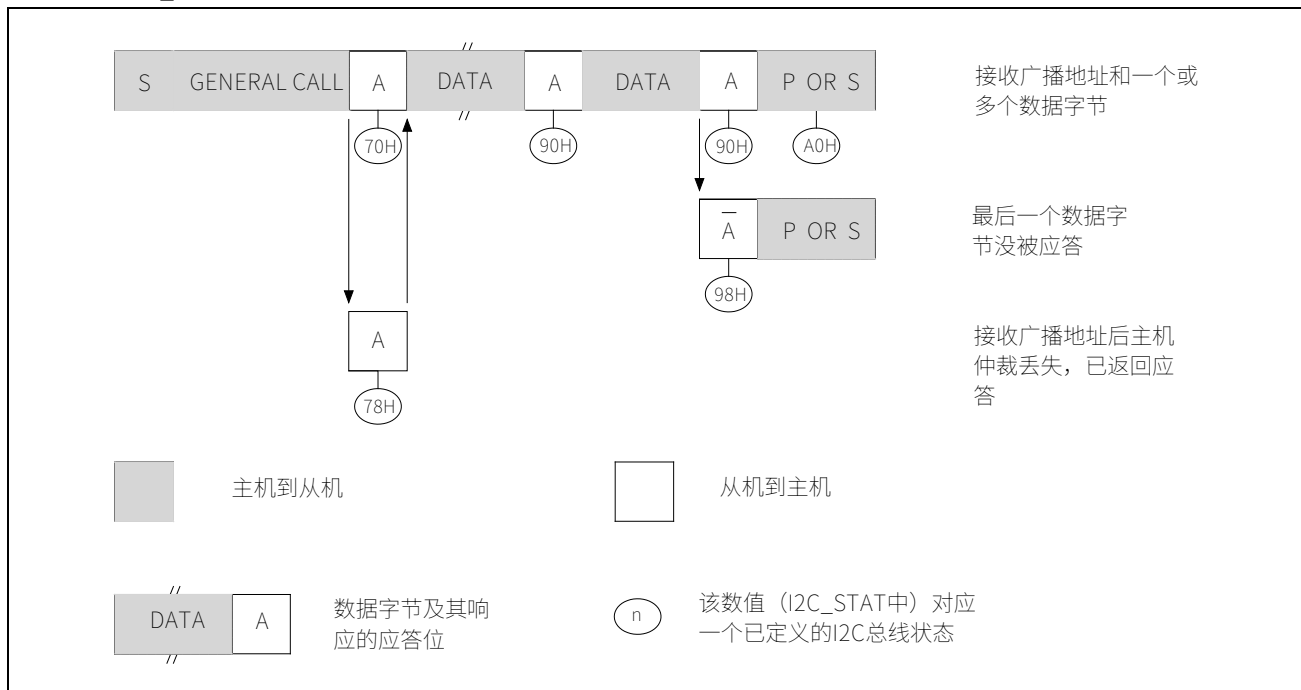


图 7-15 I2C 广播呼叫状态图

7.4.8 状态码表述

I2C 状态寄存器中有两种特殊的状态：F8H 和 00H。

F8H：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。

00H：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 7-2 I2C 状态码表述

状态代码	描述
主发送模式	
08H	已发送起始条件
10H	已发送重复起始条件
18H	已发送 SLA+W，已接收 ACK
20H	已发送 SLA+W，已接收非 ACK
28H	已发送 I2C_DATA 中的数据，已接收 ACK
30H	已发送 I2C_DATA 中的数据，已接收非 ACK
38H	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	
08H	已发送起始条件
10H	已发送重复起始条件
38H	在非 ACK 中丢失仲裁（即主机在回复 ACK 时发现 SDA 已经被其它器件拉低）
40H	已发送 SLA+R，已接收 ACK
48H	已发送 SLA+R，已接收非 ACK
50H	已接收数据字节，ACK 已返回
58H	已接收数据字节，非 ACK 已返回
从接收模式	
60H	已接收自身的 SLA+W，已返回 ACK
68H	主控时在 SLA+读写丢失仲裁，已接收自身的 SLA+W，已返回 ACK
80H	前一次寻址使用自身从地址，已接收数据字节，已返回 ACK
88H	前一次寻址使用自身从地址，已接收数据字节，已返回非 ACK
A0H	接收到停止条件或重复起始条件后，进入非寻址的从机状态，仍可被寻址
从发送模式	
A8H	已接收自身的 SLA+R，已返回 ACK
B0H	当主机时丢失仲裁，已接收自身 SLA+R，已返回 ACK
B8H	已发送第一个数据字节，已接收 ACK
C0H	已发送第一个数据字节，已接收非 ACK

状态代码	描述
C8H	装入的数据字节已被发送，已接收 ACK
广播呼叫模式	
70H	已接收广播地址（0x00），已返回 ACK
78H	主控时在 SLA+读写丢失仲裁，已接收广播地址，已返回 ACK
90H	前一次寻址使用广播地址，已接收数据字节，已返回 ACK
98H	前一次寻址使用广播地址，已接收数据字节，已返回非 ACK
A0H	接收到停止条件或重复起始条件后，进入非寻址的从机状态，仍可被寻址
其余杂项状态	
F8H	无可用的相关状态信息，SI=0
00H	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

7.5 编程示例

7.5.1 主机发送示例

Step1. 按 GPIO 章节管脚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；

注意：SCL、SDA 需按照 I2C 协议的要求外接上拉电阻。

Step2. 设置 SysCtrl_PeriClkEn0.I2C 为 1，使能 I2C 配置时钟和工作时钟；

Step3. 向 PeriReset0.I2C 依次写入 0、1，复位 I2C 模块；

Step4. 配置 I2C_BRR，使 SCL 的时钟速率符合应用需求；

Step5. 设置 I2C_BRREN 为 1，使能 SCL 时钟发生器；

Step6. 设置 I2C_CR.EN 为 1，使能 I2C 模块；

Step7. 设置 I2C_CR.STA 为 1，总线尝试发送 Start 信号；

Step8. 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；

Step9. 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；

Step10. 向 I2C_DATA 中写入 SLA+W，设置 I2C_CR.SI 为 0，发送 SLA+W；

Step11. 等待 I2C_CR.SI 变为 1，SLA+W 已发送到总线上；

Step12. 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；

Step13. 向 I2C_DATA 写入待发送的数据，设置 I2C_CR.SI 为 0，发送数据；

Step14. 等待 I2C_CR.SI 变为 1，数据已发送到总线上；

Step15. 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；

Step16. 如待发送的数据未完成，则跳转到 Step13 继续执行；

Step17. 设置 I2C_CR.STO 为 1，设置 I2C_CR.SI 为 0，总线尝试发送 Stop 信号。

7.5.2 主机接收示例

Step1. 按 GPIO 章节管脚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；

注意：SCL、SDA 需按照 I2C 协议的要求外接上拉电阻。

Step2. 设置 SysCtrl_PeriClkEn0.I2C 为 1，使能 I2C 配置时钟和工作时钟；

- Step3. 向 PeriReset0.I2C 依次写入 0、1，复位 I2C 模块；
- Step4. 配置 I2C_BRR，使 SCL 的时钟速率符合应用需求；
- Step5. 设置 I2C_BRREN 为 1，使能 SCL 时钟发生器；
- Step6. 设置 I2C_CR.EN 为 1，使能 I2C 模块；
- Step7. 设置 I2C_CR.STA 为 1，总线尝试发送 Start 信号；
- Step8. 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；
- Step9. 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；
- Step10. 向 I2C_DATA 写入 SLA+R，设置 I2C_CR.SI 为 0，发送 SLA+R；
- Step11. 等待 I2C_CR.SI 变为 1，SLA+R 已发送到总线上；
- Step12. 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；
- Step13. 设置 I2C_CR.AA 为 1，使能应答标志；
- Step14. 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；
- Step15. 等待 I2C_CR.SI 变为 1，从 I2C_DATA 读取已接收到的数据；
- Step16. 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；
- Step17. 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；
- Step18. 如待接收的数据未完成，则跳转到 Step14 继续执行；
- Step19. 设置 I2C_CR.STO 为 1，设置 I2C_CR.SI 为 0，总线尝试发送 Stop 信号。

7.5.3 从机接收示例

- Step1. 按 GPIO 章节管脚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；
- Step2. 设置 SysCtrl_PeriClkEn0.I2C 为 1，使能 I2C 配置时钟和工作时钟；
- Step3. 向 PeriReset0.I2C 依次写入 0、1，复位 I2C 模块；
- Step4. 设置 I2C_CR.EN 为 1，使能 I2C 模块；
- Step5. 配置 I2C_ADDRx 为从机地址；
- Step6. 设置 I2C_CR.AA 为 1，使能应答标志；
- Step7. 等待 I2C_CR.SI 变为 1，被 SLA+W 寻址；

- Step8. 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；
- Step9. 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；
- Step10. 等待 I2C_CR.SI 变为 1，从 I2C_DATA 中读取已接收到的数据；
- Step11. 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；
- Step12. 如待接收的数据未完成，则跳转到 Step9 继续执行；
- Step13. 设置 I2C_CR.AA 为 0，设置 I2C_CR.SI 为 0。

7.5.4 从机发送示例

- Step1. 按 GPIO 章节管脚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；
- Step2. 设置 SysCtrl_PeriClkEn0.I2C 为 1，使能 I2C 配置时钟和工作时钟；
- Step3. 向 PeriReset0.I2C 依次写入 0、1，复位 I2C 模块；
- Step4. 设置 I2C_CR.EN 为 1，使能 I2C 模块；
- Step5. 配置 I2C_ADDRx 为从机地址；
- Step6. 设置 I2C_CR.AA 为 1，使能应答标志；
- Step7. 等待 I2C_CR.SI 变为 1，被 SLA+R 寻址；
- Step8. 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；
- Step9. 向 I2C_DATA 写入待发送的数据，设置 I2C_CR.SI 为 0，发送数据；
- Step10. 等待 I2C_CR.SI 变为 1，数据已发送到总线上；
- Step11. 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；
- Step12. 如待发送的数据未完成，则跳转到 Step9 继续执行；
- Step13. 设置 I2C_CR.AA 为 0，设置 I2C_CR.SI 为 0。

7.6 寄存器描述

7.6.1 寄存器列表

表 7-3 I2C 基地址列表

名称	基地址	描述
I2C0	0x4000 2800	I2C0 基地址
I2C1	0x4000 2c00	I2C1 基地址

表 7-4 I2C 寄存器列表

寄存器名	偏移地址	复位值
I2C波特率发生器使能寄存器 (I2C_BRREN)	0x00	0x0000 0000
I2C波特率计数器配置寄存器 (I2C_BRR)	0x04	0x0000 0000
I2C配置寄存器 (I2C_CR)	0x08	0x0000 0000
I2C数据寄存器 (I2C_DR)	0x0c	0x0000 0000
I2C状态寄存器 (I2C_STAT)	0x14	0x0000 0000
I2C从机地址0寄存器 (I2C_ADDR0)	0x10	0x0000 0000
I2C从机地址1寄存器 (I2C_ADDR1)	0x20	0x0000 0000
I2C从机地址2寄存器 (I2C_ADDR2)	0x24	0x0000 0000
I2C从机地址匹配寄存器 (I2C_MATCH)	0x28	0x0000 0000

7.6.2 波特率发生器使能寄存器 (I2C_BRREN)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														EN	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	EN	波特率发生器使能控制	1: 使能 0: 禁止	RW

7.6.3 波特率计数器配置寄存器 (I2C_BRR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								BRR							

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7~b0	BRR	I2C 总线 SCL 波特率配置	$f_{SCL} = f_{PCLK} / 8 / (BRR + 1)$, 其中 $BRR > 0$	RW

7.6.4 配置寄存器 (I2C_CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EN	STA	STO	SI	AA	Res	FLT

位	标记	位名	功能	读写
b31~b7	Res	保留位	保留, 必须保持复位值	
b6	EN	模块使能控制	1: 使能 0: 禁止	RW
b5	STA	总线状态控制	1: 向总线发送 START 0: 无功能 注意: START 发送完成后, 用户程序应设置该比特为 0。	RW
b4	STO	总线状态控制	1: 向总线发送 STOP 0: 无功能 注意: STOP 发送完成后硬件自动设置该比特为 0。	RW
b3	SI	I2C 中断标志	读出 1: 已发生 I2C 中断 读出 0: 未发生 I2C 中断 写入 1: 无功能 写入 0: 清除 I2C 中断标志并使状态机执行下一个动作	RW
b2	AA	应答控制	1: 在应答阶段发送 ACK 0: 在应答阶段发送 NAK	RW
b1	Res	保留位	保留, 必须保持复位值	
b0	FLT	I2C 滤波参数配置	1: 简单滤波, 更快的通信速率 0: 高级滤波, 更高的抗干扰性能 注: 详见输入滤波器章节。	RW

7.6.5 数据寄存器 (I2C_DR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									DR						

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7~b0	DR	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

7.6.6 状态寄存器 (I2C_STAT)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								STAT							
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7~b0	STAT	I2C 状态寄存器	状态值的具体定义详见【状态码表述】章节										RO		

7.6.7 从机地址 0 寄存器 (I2C_ADDR0)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								Addr							GC
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7~b1	Addr	I2C 从机模式地址 0											RW		
b0	GC	广播地址应答使能	1: 使能 0: 禁止										RW		

7.6.8 从机地址 1 寄存器 (I2C_ADDR1)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								Addr							Res.
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7~b1	Addr	I2C 从机模式地址 1											RW		
b0	Res	保留位	保留, 必须保持复位值												

7.6.9 从机地址 2 寄存器 (I2C_ADDR2)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								Addr						Res.	

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7~b1	Addr	I2C 从机模式地址 2		RW
b0	Res	保留位	保留, 必须保持复位值	

7.6.10 从机地址匹配寄存器 (I2C_MATCH)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													AD2F	AD1F	AD0F

位	标记	位名	功能	读写
b31~b3	Res	保留位	保留, 必须保持复位值	
b2	AD2F	I2C 从机模式地址 2 匹配标志位	1: 从总线收到的设备地址与 ADDR2 相同 0: 从总线收到的设备地址与 ADDR2 不相同	RO
b1	AD1F	I2C 从机模式地址 1 匹配标志位	1: 从总线收到的设备地址与 ADDR1 相同 0: 从总线收到的设备地址与 ADDR1 不相同	RO
b0	AD0F	I2C 从机模式地址 0 匹配标志位	1: 从总线收到的设备地址与 ADDR0 相同 0: 从总线收到的设备地址与 ADDR0 不相同	RO

8 串行外设接口 (SPI)

8.1 概述

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。

8.2 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

8.3 功能描述

8.3.1 功能框图

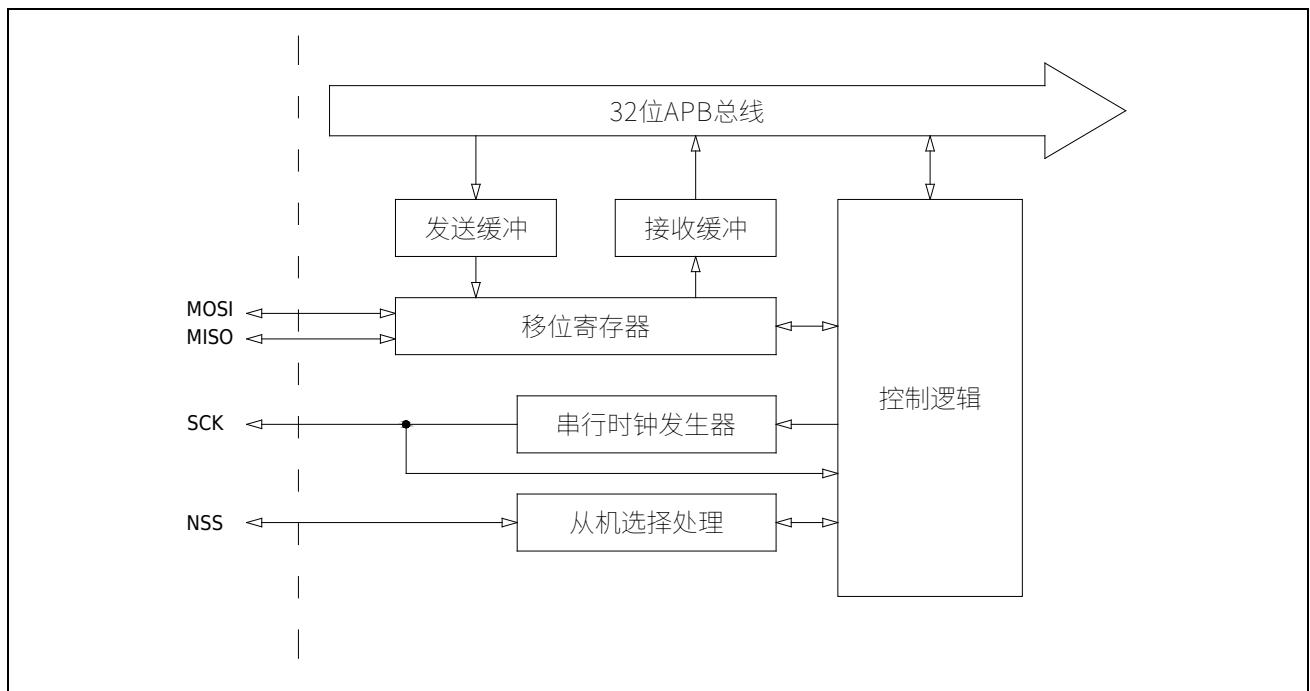


图 8-1 SPI 框图

8.3.2 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。

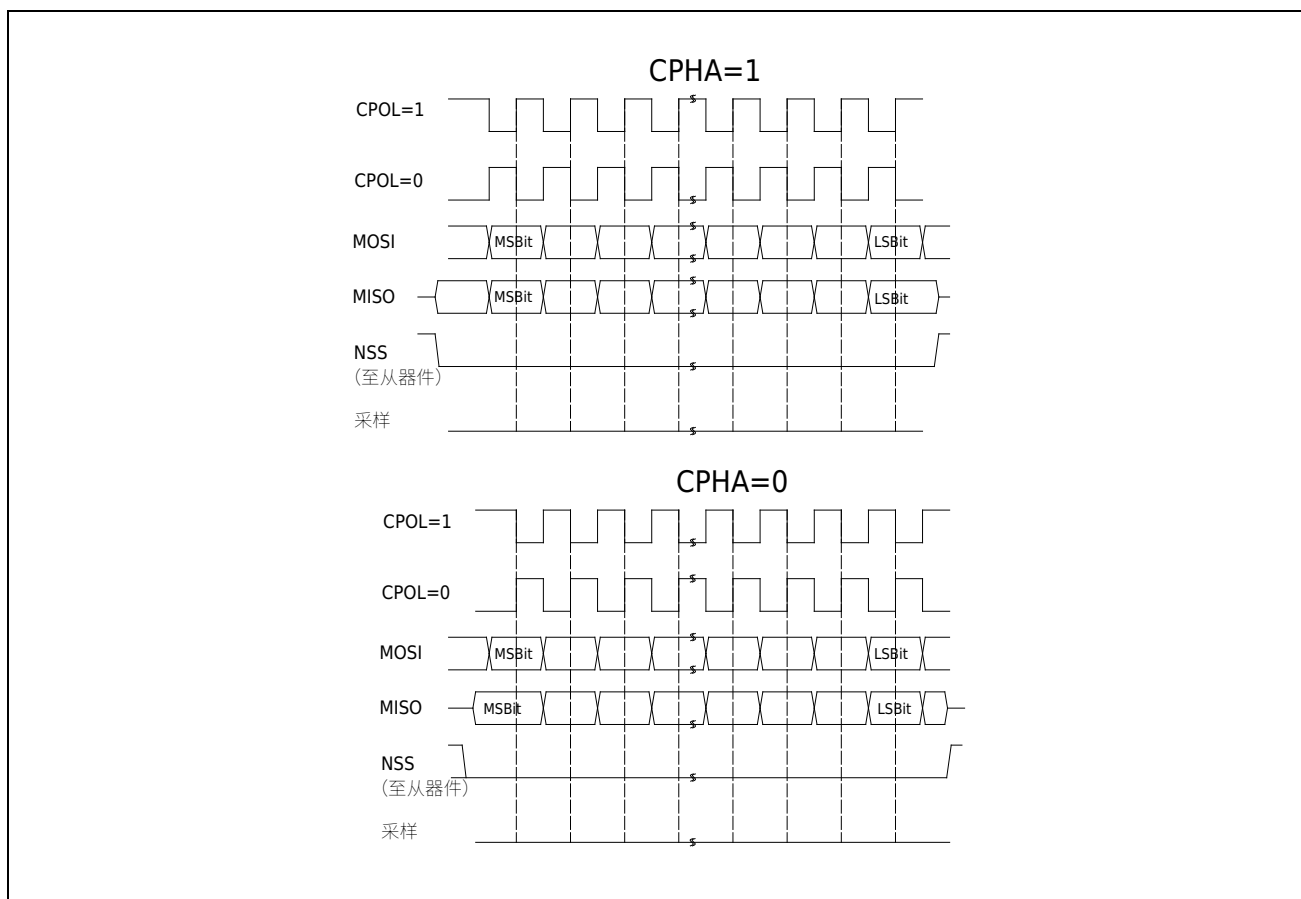


图 8-2 SPI 通信格式时序图

8.3.3 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。

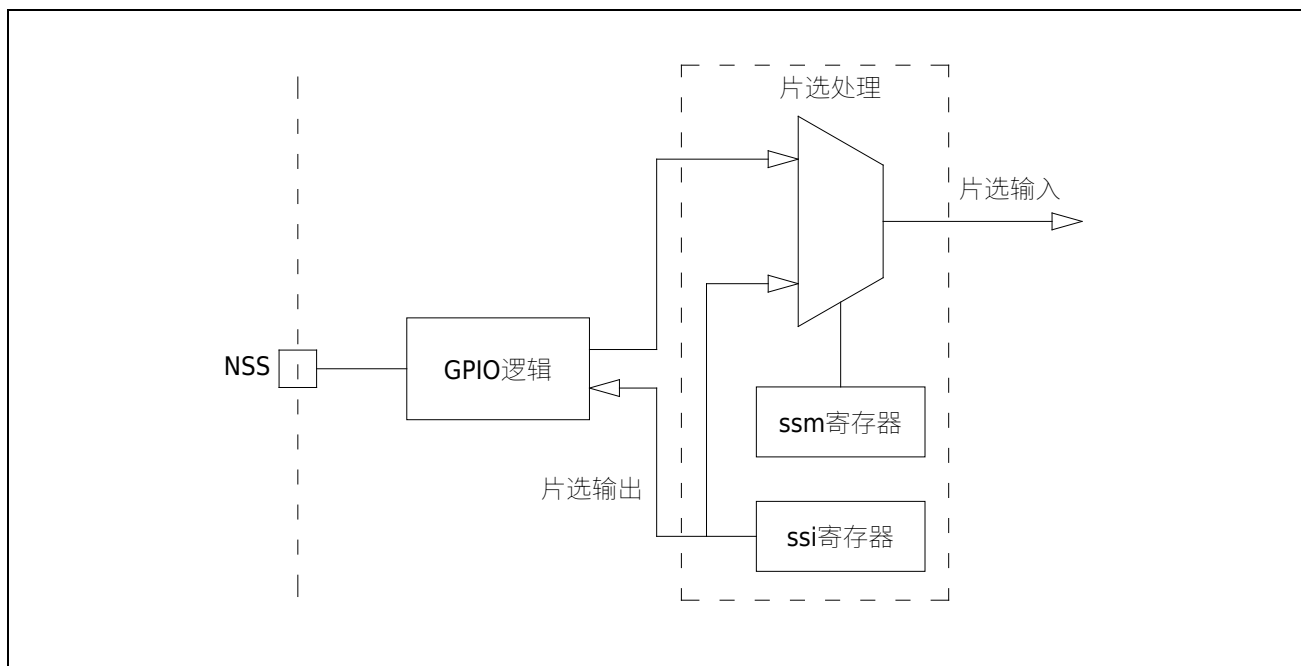


图 8-3 从机选择 NSS 管脚配置

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，只有将本模块的寄存器 SPI_SSI.SSI 配置为 0，才能选中本模块。

8.3.4 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的双向的通信。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 主机将通过 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。

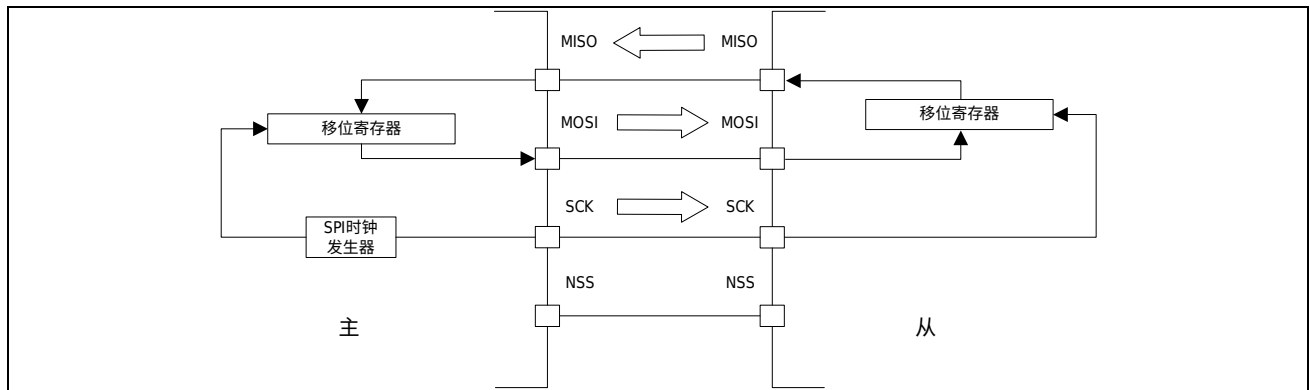


图 8-4 SPI 全双工通信

8.3.5 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的双向的通信。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 HDOE.HDOE 决定，当 HDOE.HDOE 为 1 时，数据线进行发送。未用到的管脚不参与 SPI 通信，可供其它功能使用。当主机工作于接收状态时，需要向 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。

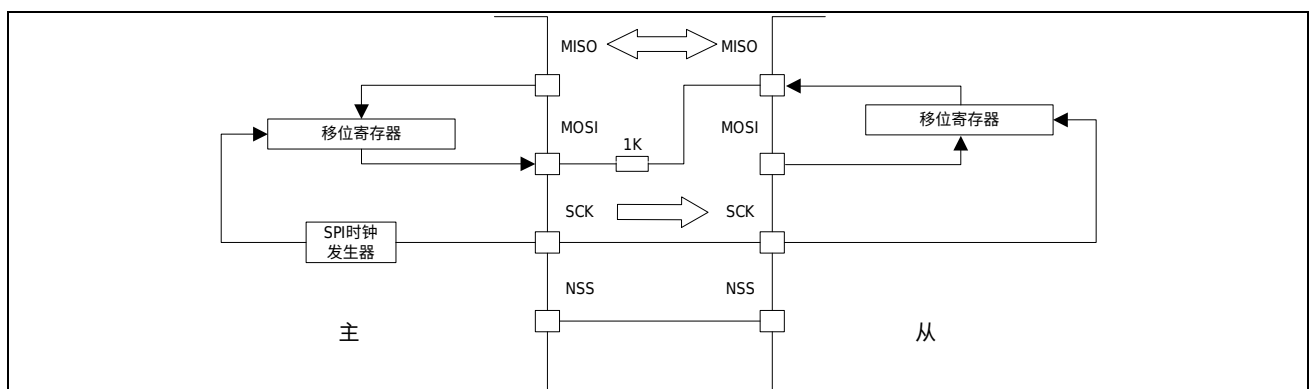


图 8-5 SPI 单线半双工通信

8.3.6 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，可供其它功能使用。

当主机工作于发送状态，从机工作于接收状态时，主机向 DR 寄存器写入数据以触发 SPI 主机发送 SCK 进行数据传输。

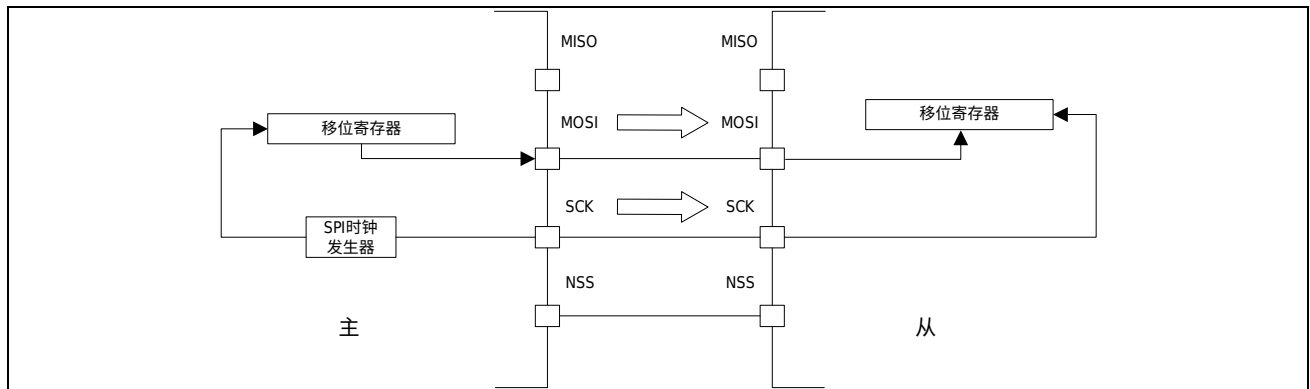


图 8-6 SPI 单工发送通信示意

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 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。

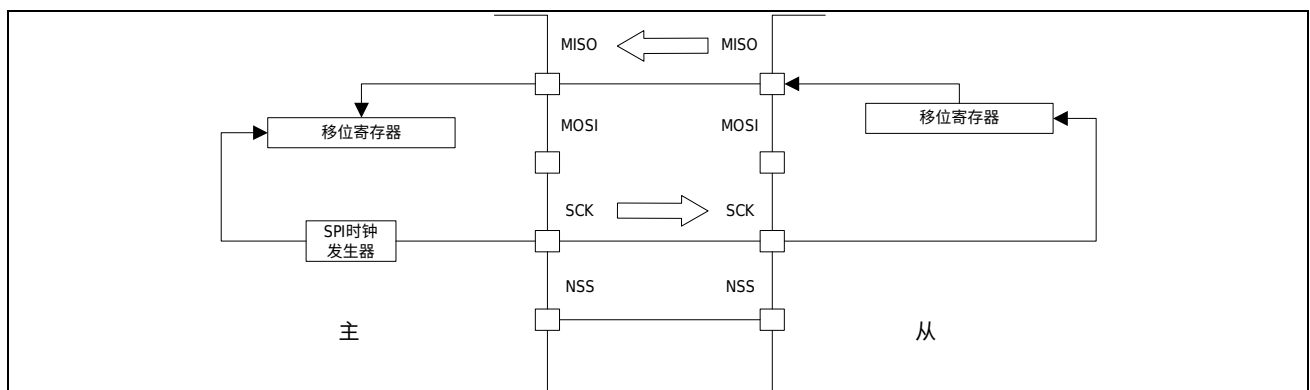


图 8-7 SPI 单工接收通信示意

注意：

1. 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。
2. 该配置下，两个闲置的管脚（MOSI 或者 MISO）均可用作 GPIO。

8.3.7 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 管脚一旦被其它主机的 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。

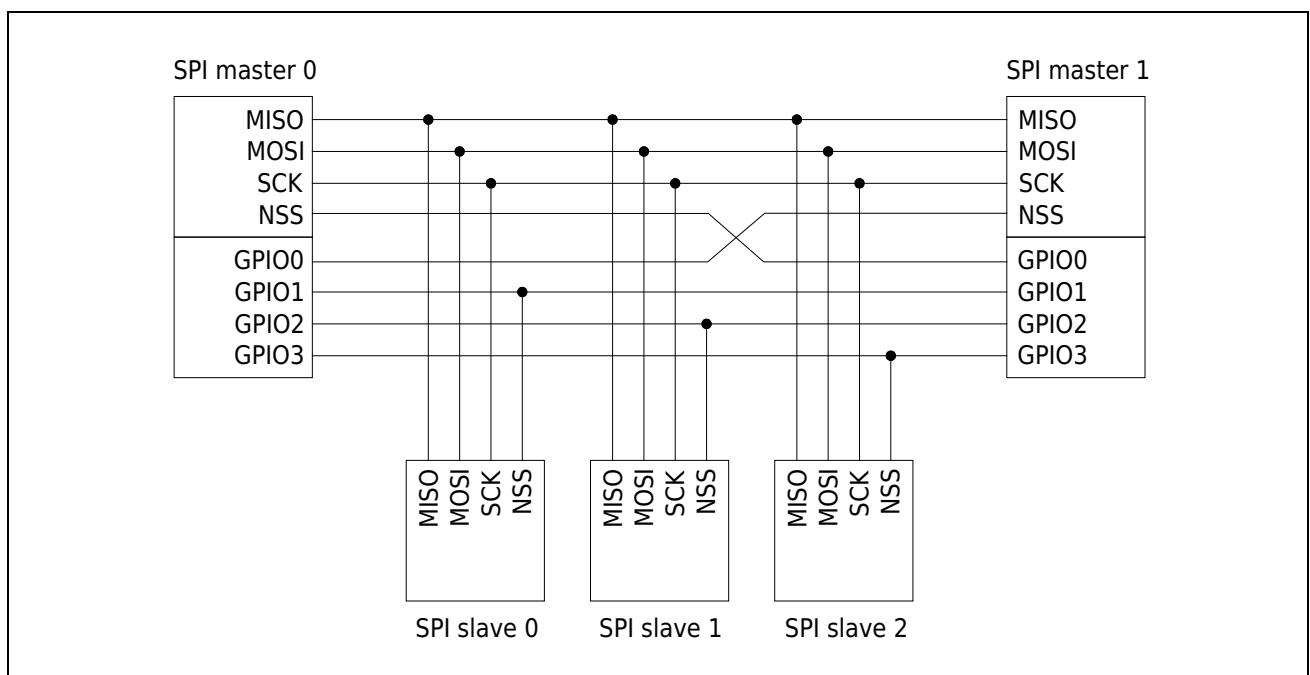


图 8-8 SPI 多机系统

8.3.8 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。

注意：

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。

8.3.9 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，其它情况不会触发该错误。

CPOL=1,CPHA=1时的主模式示例

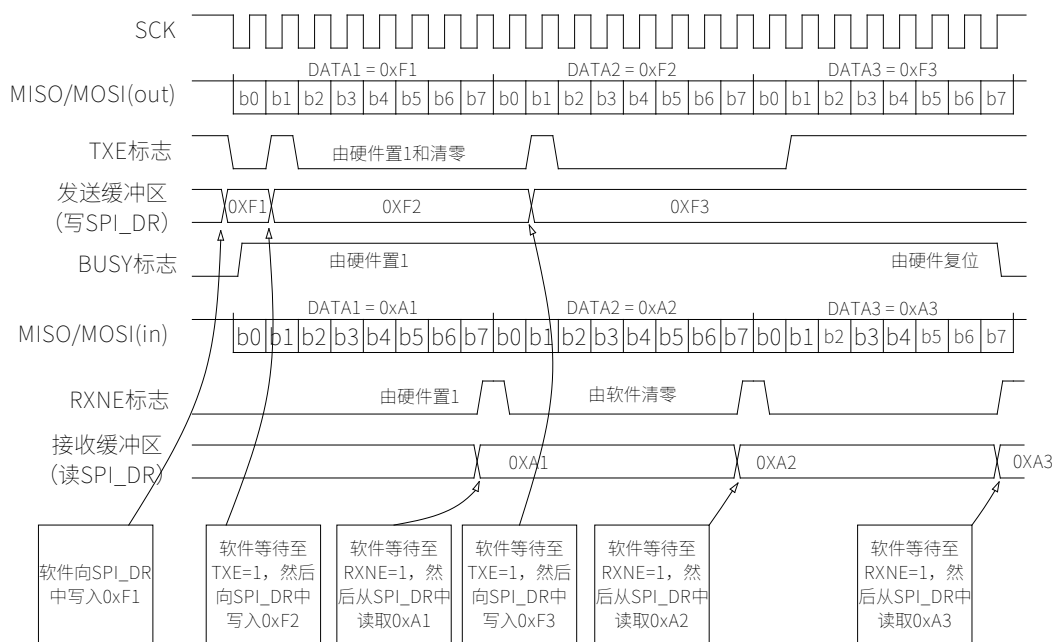


图 8-9 SPI 主机模式一般标志位说明图

CPOL=1,CPHA=1时的从模式示例

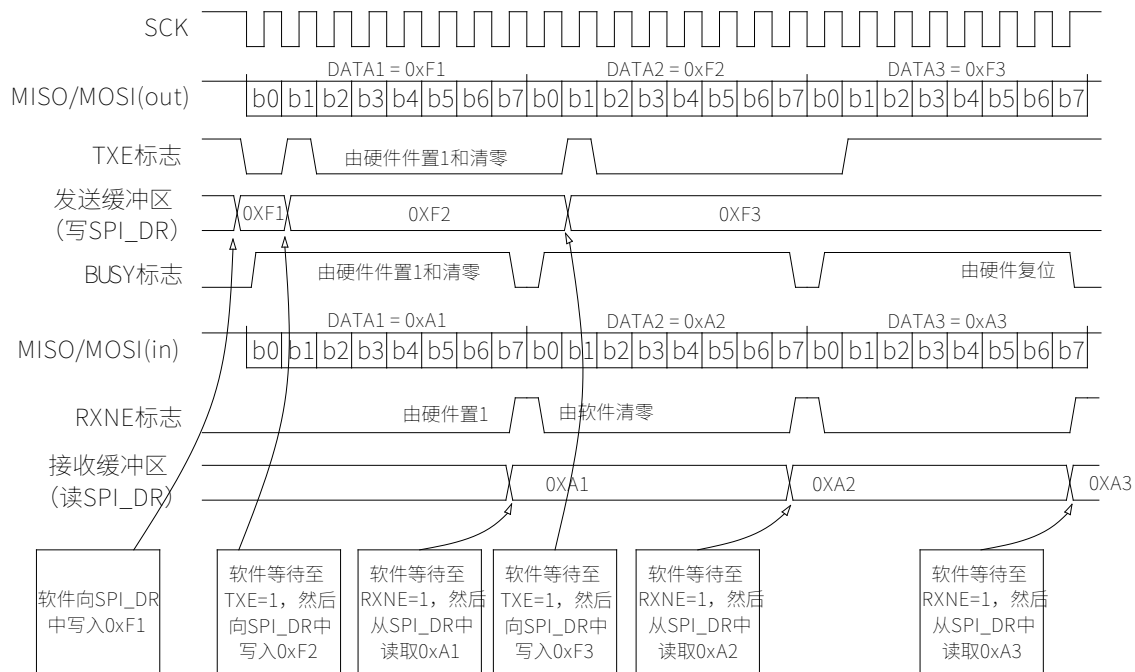


图 8-10 SPI 从机模式一般标志位说明图

8.3.10 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 8-1 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.SSRC 写 0
从机选择输入下降沿	SSF	SSFIE	向 SPI_ICR.SSFC 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.RXNEC 写 0
从机模式下从机选择错误	SSERR	SSERRIE	向 SPI_ICR.SSERRC 写 0
接收缓冲上溢错误	OVF	OVFIE	向 SPI_ICR.OVFC 写 0
从机模式发送缓冲下溢错误	UDF	UDFIE	向 SPI_ICR.UDFC 写 0
主机模式下模式错误	MODF	MODFIE	向 SPI_ICR.MODFC 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。

8.4 编程示例

8.4.1 主机全双工收发数据示例

- Step1. 按 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；
- Step2. 设置 SYSCTRL_APB0EN.SPI 为 1，使能 SPI 配置时钟及工作时钟；
- Step3. 设置 CR0.MSTR 为 1，选择主机模式；
- Step4. 设置 CR0.BRR，配置通信波特率；
- Step5. 设置 CR0.CM 为 0，选择全双工通信方式；
- Step6. 设置 CR0.SSM 为 1，配置从机选择模式；
- Step7. 设置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位；
- Step8. 设置 CR0.WIDTH，配置每帧的数据宽度；
- Step9. 设置 CR0.LSBF，配置收发数据时的数据序列顺序；
- Step10. 设置 CR0.EN 为 1，使能 SPI 模块；
- Step11. 设置 SSI.SSI 为 0，选中对应的从机；
- Step12. 将待发送数据写入 DR 寄存器，SPI 自动开始数据传输；
- Step13. 查询等待 SR.RXNE 变为 1，完成一帧数据的发送与接收；
- Step14. 从 DR 寄存器中读出接收到的数据并保存；
- Step15. 若待发送与接收的数据未完成，则跳转到 Step12 继续执行。

8.4.2 从机全双工收发数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述，将 NSS、SCK、MOSI 及 MISO 映射到需要的管脚；
- Step2. 设置 SYSCTRL_APB0EN.SPI 为 1，使能 SPI 配置时钟及工作时钟；
- Step3. 设置 CR0.MSTR 为 0，选择从机模式；
- Step4. 设置 CR0.CM 为 0，选择全双工通信方式；
- Step5. 设置 CR0.SSM，配置从机选择模式；
- Step6. 设置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位；
- Step7. 设置 CR0.WIDTH，配置每帧的数据宽度；

- Step8. 设置 CR0.LSBF，配置收发数据时的数据序列顺序；
- Step9. 设置 CR0.EN 为 1，使能 SPI 模块；
- Step10. 设置 ICR.TXES 为 0，清空发送缓冲区；
- Step11. 查询等待 SR.SSLVL 变为 0，SPI 主机选中本模块；
- Step12. 将待发送的数据写入 DR 寄存器，准备好待发送的数据；
- Step13. 查询等待 SR.RXNE 变为 1，完成一帧数据的发送与接收；
- Step14. 从 DR 寄存器中读出接收到的数据并保存；
- Step15. 若待发送与接收的数据未完成，则跳转到 Step12 继续执行；
- Step16. 等待 SR.SSLVL 变为 1，完成本次通信。

8.4.3 单线半双工发送接收数据示例

- Step1. 按 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；
- Step2. 设置 SYSCTRL_APB0EN.SPI 为 1，使能 SPI 配置时钟及工作时钟；
- Step3. 主机设置 CR0.MSTR 为 1，从机设置 CR0.MSTR 为 0；
- Step4. 主机设置 CR0.BRR，配置通信波特率；
- Step5. 设置 CR0.CM 为 11，选择单线半双工通信方式；
- Step6. 配置 CR0.SSM，配置从机选择模式；
- Step7. 配置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位；
- Step8. 配置 CR0.WIDTH，配置每帧的数据宽度；
- Step9. 配置 CR0.LSBF，配置发送数据时的数据序列顺序；
- Step10. 设置 CR0.EN 为 1，使能 SPI 模块；
- Step11. 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；
- Step12. 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；
- Step13. 主机将待发送数据写入 DR 寄存器，SPI 自动开始数据传输；
- Step14. 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；

- Step15. 从机从 DR 寄存器中读出接收到的数据并保存；
- Step16. 若主机待发送的数据未完成，则跳转到 Step13 继续执行；
- Step17. 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；
- Step18. 从机查询等待 SR.TXE 变为 1，发送缓存空闲；
- Step19. 若从机待发送的数据未完成，则向 DR 寄存器写入一帧数据；
- Step20. 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；
- Step21. 主机查询等待 SR.RXNE 变为 1，完成一帧数据的接收；
- Step22. 主机从 DR 寄存器中读出接收到的数据并保存；
- Step23. 若从机待发送的数据未完成，则跳转到 Step18 继续执行；
- Step24. 从机等待 SR.SSLVL 变为 1，设置 HDOE.HDOE 为 0，完成本次通信。

8.5 寄存器描述

8.5.1 寄存器列表

表 8-2 SPI 基地址列表

名称	基地址	描述
SPI0	0x4000 3800	SPI0 基地址
SPI1	0x4000 3C00	SPI1 基地址

表 8-3 SPI 寄存器列表

寄存器名	偏移量	复位值
SPI 控制寄存器 0 (SPI_CR0)	0x00	0x0000 0700
SPI 控制寄存器 1 (SPI_CR1)	0x04	0x0000 0000
SPI 单线半双工输出使能寄存器 (SPI_HDOE)	0x08	0x0000 0000
SPI 内部从机选择寄存器 (SPI_SSI)	0x0C	0x0000 0001
SPI 状态寄存器 (SPI_SR)	0x10	0x0000 0001
SPI 中断清除寄存器 (SPI_ICR)	0x14	0x0000 00FF
SPI 数据寄存器 (SPI_DR)	0x18	0x0000 0000

8.5.2 SPI 控制寄存器 0 (SPI_CR0)

复位值: 0x0000 0700

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SMP	CM		SSM	WIDTH				LSBF	MSTR	CPOL	CPHA	BRR			EN
位		标记		位名				功能						读写	
b31~b16		Res		保留位				保留，必须保持复位值							
b15		SMP		主机模式延后采样位				0：SPI 传统方式，主机接收数据按照 CPOL 和 CPHA 的设定进行采样 1：延后采样方式，采样时刻相比传统方式延后半个 SCK 周期						RW	
b14~b13		CM		通信方式配置				00：全双工双向 01：单工只发 10：单工只收 11：单线半双工						RW	
b12		SSM		从机选择配置				主机模式 1：SSI 寄存器决定从机选择输出值 0：NSS 管脚决定从机选择输入值 从机模式 1：SSI 寄存器决定从机选择输入值 0：NSS 管脚决定从机选择输入值						RW	
b11~b8		WIDTH		帧数据宽度配置				0011：4bit 0100：5bit ... 1110：15bit 1111：16bit						RW	
b7		LSBF		数据帧高低位顺序选择				1：最低有效位 LSB 收发在前 0：最高有效位 MSB 收发在前						RW	
b6		MSTR		工作模式配置				1：主机模式 0：从机模式						RW	
b5		CPOL		串行时钟极性配置				1：待机时高电平 0：待机时低电平						RW	
b4		CPHA		串行时钟相位配置				1：前边沿移位/后边沿采样 0：前边沿采样/后边沿移位						RW	
b3~b1		BRR		主机模式波特率配置				000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 ... 110：PCLK/128 111：保留						RW	
b0		EN		使能控制				1：使能 SPI 模块 0：禁止 SPI 模块						RW	

8.5.3 SPI 控制寄存器 1 (SPI_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						DMATXEN	DMA RXEN	MODFIE	SSERRIE	OVFIE	UDFIE	SSRIE	SSFIE	RXNEIE	TXEIE

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	DMATXEN	发送 DMA 使能	1: 使能 0: 禁用	RW
b8	DMARXEN	接收 DMA 使能	1: 使能 0: 禁用	RW
b7	MODFIE	模式错误中断使能控制	1: 使能 0: 禁用	RW
b6	SSERRIE	从机模式下的从机选择错误中断使能控制	1: 使能 0: 禁用	RW
b5	OVFIE	接收缓冲上溢错误中断使能控制	1: 使能 0: 禁用	RW
b4	UDFIE	从机模式下发送缓冲下溢错误中断使能控制	1: 使能 0: 禁用	RW
b3	SSRIE	从机选择输入上升沿中断使能控制	1: 使能 0: 禁用	RW
b2	SSFIE	从机选择输入下降沿中断使能控制	1: 使能 0: 禁用	RW
b1	RXNEIE	接收缓冲非空中断使能控制	1: 使能 0: 禁用	RW
b0	TXEIE	发送缓冲空中断使能控制	1: 使能 0: 禁用	RW

8.5.4 SPI 单线半双工输出使能寄存器 (SPI_HDOE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														HDOE	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	HDOE	MOSI/MISO 收发切换控制	0: 只收 1: 只发 注: 仅在单线半双工通信时有效	RW

8.5.5 SPI 内部从机选择寄存器 (SPI_SSI)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														SSI	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	SSI	内部从机选择	当 SPI_CR0.SSM 寄存器为 1 时有效 (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)	RW

8.5.6 SPI 状态寄存器 (SPI_SR)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SSLVL	BUSY	MOD F	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	SSLVL	从机选择输入的状态	1: 从机选择输入是高电平 0: 从机选择输入是低电平 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。	RO
b8	BUSY	总线忙标志位	1: 总线忙 0: 总线空闲	RO
b7	MODF	模式错误标志位 (主机模式下从机选择输入为低则触发 MODF)	1: 出错 0: 正常	RO
b6	SSERR	从机模式下的从机选择错误标志位	1: 出错 0: 正常	RO
b5	OVF	接收缓冲上溢错误标志位	1: 出错 0: 正常	RO
b4	UDF	从机模式下的发送缓冲下溢错误标志位	1: 出错 0: 正常	RO
b3	SSR	从机选择输入上升沿标志位	1: 出现了上升沿 0: 未出现上升沿 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。	RO
b2	SSF	从机选择输入下降沿标志位	1: 出现了下降沿 0: 未出现下降沿 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。	RO
b1	RXNE	接收缓冲非空标志位	1: 接收缓冲非空 0: 接收缓冲空	RO
b0	TXE	发送缓冲空标志位	1: 发送缓冲空 0: 发送缓冲非空	RO

8.5.7 SPI 中断清除寄存器 (SPI_ICR)

复位值: 0x0000 00FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MODF	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7	MODF	ISR.MODF 标志清除位	写 0 清除 ISR.MODF 标志, 读出为 1。	R1W0
b6	SSERR	ISR.SSERR 标志清除位	写 0 清除 ISR.SSERR 标志, 读出为 1。	R1W0
b5	OVF	ISR.OVF 标志清除位	写 0 清除 ISR.OVF 标志, 读出为 1。	R1W0
b4	UDF	ISR.UDF 标志清除位	写 0 清除 ISR.UDF 标志, 读出为 1。	R1W0
b3	SSR	ISR.SSR 标志清除位	写 0 清除 ISR.SSR 标志, 读出为 1。	R1W0
b2	SSF	ISR.SSF 标志清除位	写 0 清除 ISR.SSF 标志, 读出为 1。	R1W0
b1	RXNE	ISR.RXNE 标志清除位	写 0 清除 ISR.RXNE 标志, 读出为 1。	R1W0
b0	TXE	ISR.TXE 清除位	写 0 设置 ISR.TXE 标志为 1, 读出为 1。	R1W0

8.5.8 SPI 数据寄存器 (SPI_DR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	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

9 时钟校准模块（CTRIM）

9.1 概述

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 下仍可正常工作。

9.2 主要特性

- 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，周期约为 60us ~ 65536s（32.768KHz 频率下）
- 定时模式：5 种时钟源可供选择
- 定时模式：翻转输出

9.3 功能描述

9.3.1 RCH 时钟校准模式

当配置 CTRIM.CR1.MD 为 0x00 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。

RCH 时钟校准时的工作框图如下方所示。

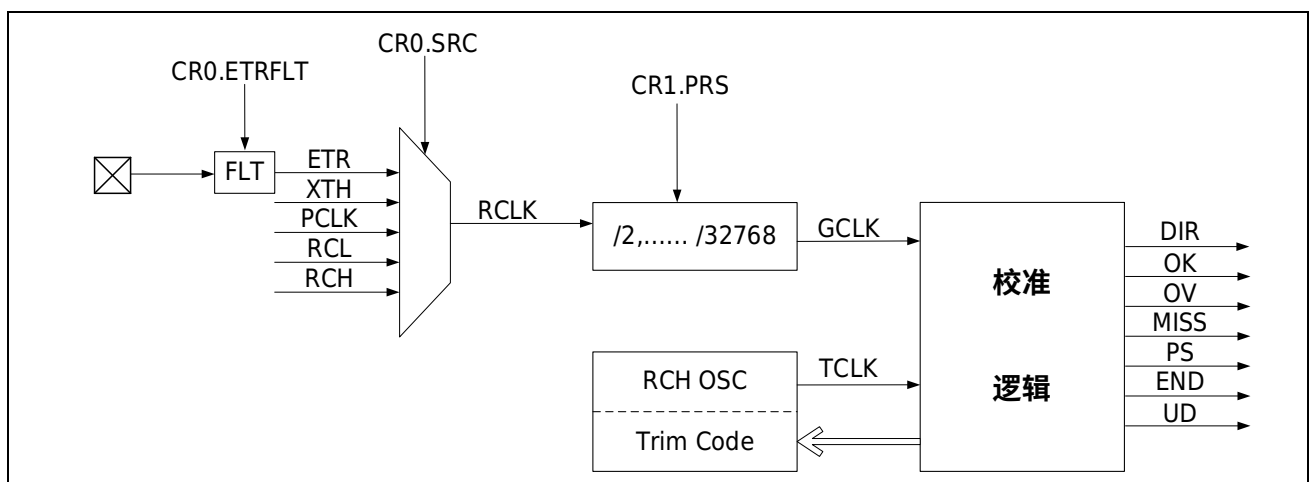


图 9-1 RCH 时钟校准工作框图

校准 RCH 时，需要向该模块提供一个精准的低频时钟信号，可通过 CTRIM_CR0.SRC 选择时钟源。注意，该低频时钟信号的频率应小于 RCH 频率的 1/2。该低频时钟信号经过 CTRIM_CR1.PRS 设置的分频输出后生成 GCLK，GCLK 作为校准逻辑单元的输入。

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，以 CTRIM_CR0.STEP 设置的值作为初始步进量进行递减计数，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等于 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，并置位 CTRIM_ISR.END；

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 Trim Code 并进行下一轮计数。自动调整规则为：在每个 GCLK 周期结束时，若计数方向为减计数则对 RCH 的 Trim Code 增加步进量；若计数方式为加计数则对 RCH 的 Trim Code 减少步进量。

校准逻辑会自动调整步进量的数值以实现 RCH 输出的频率逐次逼近目标值。当步进量被调整为 1 时，则置位 CTRIM_ISR.END 以标识自动校准结束。

当误差计数器计数值递减为零时，生成 UD（下溢）事件并置位 CTRIM_ISR.UD；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，置位 CTRIM_ISR.MISS。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

单次校准完成后用户需将 CTRIM_TVAL[8:0] 写入 SYSCtrl_RCHTRIM 寄存器并根据需要关闭校准模块。

9.3.2 RCL 时钟校准模式

当配置 CTRIM.CR1.MD 为 0x01 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。

RCL 时钟校准时的工作框图如下方所示。

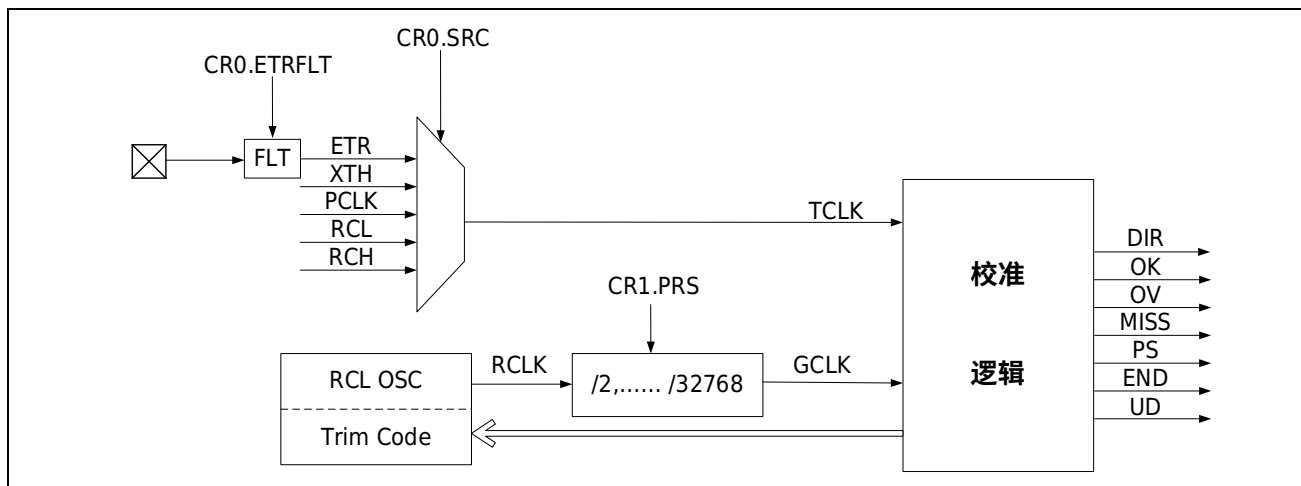


图 9-2 RCL 时钟校准工作框图

校准 RCL 时，需要向该模块提供一个精准的高频时钟信号，可通过 CTRIM_CR0.SRC 选择时钟源。该时钟源作为 TCLK 输入到校准逻辑单元。

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，以 CTRIM_CR0.STEP 设置的值作为初始步进量进行递减计数，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等于 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，并置位 CTRIM_ISR.END；

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 Trim Code 并进行下一轮计数。自动调整规则为：在每个 GCLK 周期结束时，若计数方向为减计数则对 RCL 的 Trim Code 减少步进量；若计数方式为加计数则对 RCL 的 Trim Code 增加步进量。

校准逻辑会自动调整步进量的数值以实现 RCL 输出的频率逐次逼近目标值。当步进量被调整为 1 时，则置位 CTRIM_ISR.END 以标识自动校准结束。

当误差计数器计数值递减为零时，生成 UD（下溢）事件并置位 CTRIM_ISR.UD；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，置位 CTRIM_ISR.MISS。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

单次校准完成后将 CTRIM_TVAL[8:0]写入 SYSCTRL_RCLTRIM[8:0]寄存器并根据需要关闭校准模块。

注：SYSCTRL_RCLTRIM[9]为频率选择位，需保持不变。

9.3.3 低功耗定时器模式

当配置 CTRIM.CR1.MD 为 0x02 或 0x03 时，该模块工作于通用低功耗定时器模式。该模式下 CTIRM 具有通用定时器的定时及中断功能，若定时器的时钟选择为 RCL 则其在 DeepSleep 下可正常工作，支持自动唤醒，周期 60us ~ 65536s。通用低功耗定时器时的工作框图如下方所示。

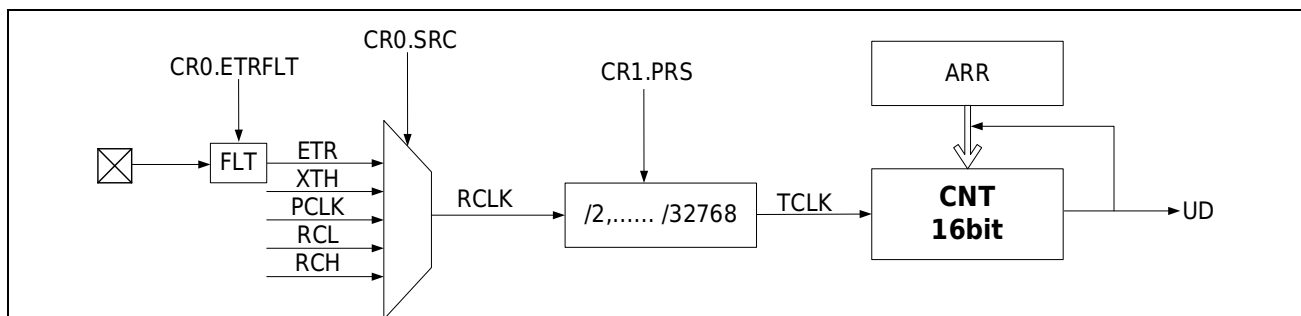


图 9-3 低功耗定时器工作框图

定时器计数值从 ARR 减计数，当计数到 0 后生成 UD（下溢）事件并置位 CTRIM_ISR.UD，如果使能中断（CTRIM_IER.UD）则会产生 UD（下溢）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。注：使用外部 ETR 输入时，TOG 输出不可用。

9.4 软件操作流程

9.4.1 RCH 时钟自动校准示例

Step1. 按照端口复用功能，配置相应端口的功能为 CTRIM_ETRTOG，并向其提供 1KHz 的方波；

Step2. 设置 CTRIM_CR1.MD 为 0，选择 RCH 时钟校准模式；

Step3. 设置 CTRIM_CR0.SRC 为 0，外部 ETR 管脚输入信号作 RCLK；

Step4. 设置 CTRIM_CR1.PRS 为 1，对 RCLK 进行 2 分频，GCLK 频率为 500Hz；

Step5. 配置 CTRIM_ARR 为 47999，一个 GCLK 的计数周期内目标计数值为 48000；

注：RCH 目标校准值为 24MHz，GCLK 的频率为 500Hz，

0.5s（500Hz）内误差计数器对 RCH 输出频率的理论计数值=24M/ 500=48000，

即 CTRIM_ARR=48000-1=47999；

Step6. 设置 CTRIM_FLIM 为 240，选择 RCH 校正精度为 0.5%；

注：48000 * 0.005=240。

Step7. 设置 CTRIM_CR0.STEP 为 4，选择初始校准步长为 16；

Step8. 设置 CTRIM_CR1.AUTO 为 1，使能自动校准；

Step9. 设置 CTRIM_CR1.EN 为 1，启动时钟校准；

Step10. 等待 CTRIM_ISR.END 为 1，自动校准完成。

注：查询 CTRIM_ISR.OK 可判断 Trim 完成时精度是否满足设置要求。

9.4.2 RCL 时钟自动校准示例

- Step1. 按照端口复用功能，配置相应端口的功能为 CTRIM_ETRTOG，并向其提供 1MHz 的方波；
- Step2. 设置 CTRIM_CR1.MD 为 1，选择 RCL 时钟校准模式；
- Step3. 设置 CTRIM_CR0.SRC 为 0，外部 ETR 管脚输入信号作 TCLK；
- Step4. 设置 CTRIM_CR1.PRS 为 0x0A，对 RCLK(RCL 输出频率 32.768KHz)进行 1024 分频，GCLK 目标频率为 32Hz；
- Step5. 配置 CTRIM_ARR 为 31249，一个 GCLK 的计数周期内目标计数值为 31250；
- 注：RCL 目标校准值为 32.768KHz，GCLK 的频率为 32Hz，
31.25ms (32Hz) 内误差计数器对 TCLK(1MHz)的理论计数值=1M/ 32=31250，
即 CTRIM_ARR=31250-1=31249；
- Step6. 设置 CTRIM_FLIM 为 156，选择 RCL 校正精度为 0.5%；
- 注：31250 * 0.005 = 156。
- Step7. 设置 CTRIM_CR0.STEP 为 4，选择初始校准步长为 16；
- Step8. 设置 CTRIM_CR1.AUTO 为 1，使能自动校准；
- Step9. 设置 CTRIM_CR1.EN 为 1，启动时钟校准；
- Step10. 等待 CTRIM_ISR.END 为 1，自动校准完成。

注：查询 CTRIM_ISR.OK 可判断 Trim 完成时精度是否满足设置要求。

9.5 寄存器描述

9.5.1 寄存器列表

基地址：0x4000 8000

表 9-1 CTRIM 寄存器列表

寄存器名	偏移地址	复位值
自动装载寄存器 (CTRIM_ARR)	0x00	0x0000 FFFF
误差计数器寄存器 (CTRIM_CNT)	0x04	0x0000 0000
控制寄存器 0 (CTRIM_CR0)	0x0C	0x0000 0000
控制寄存器 1 (CTRIM_CR1)	0x10	0x0000 0000
中断使能寄存器 (CTRIM_IER)	0x14	0x0000 0000
中断和状态寄存器 (CTRIM_ISR)	0x18	0x0000 0000
中断标志清除寄存器 (CTRIM_ICR)	0x1C	0x0000 003F
误差结果寄存器 (CTRIM_FCAP)	0x20	0x0000 FFFF
TRIM 结果寄存器 (CTRIM_TVAL)	0x24	0x0000 0100
误差上限寄存器 (CTRIM_FLIM)	0x28	0x0000 0010

9.5.2 自动装载寄存器 (CTRIM_ARR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARR															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	ARR	误差计数器重载值	误差计数器重载值										RW		

9.5.3 误差计数器寄存器 (CTRIM_CNT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	CNT	误差计数器计数值	误差计数器计数值										RO		

9.5.4 控制寄存器 0 (CTRIM_CR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					SRC			Res	ETRFLT			Res	STEP		
位		标记		位名			功能							读写	
b31~b11		Res		保留位			保留，必须保持复位值								
b10~b8		SRC		时钟来源选择			000: ETR							RW	
							001: XTH								
							010: 保留								
							011: PCLK								
							100: RCL								
							101: RCH								
							110: 保留								
							111: 保留								
b7		Res		保留位			保留，必须保持复位值								
b6~b4		ETRFLT		外部输入管脚(ETR)滤波配置			000: 无滤波							RW	
							001: $F_{\text{sample}} = \text{PCLK}$, N=2								
							010: $F_{\text{sample}} = \text{PCLK}$, N=4								
							011: $F_{\text{sample}} = \text{PCLK}$, N=6								
							100: $F_{\text{sample}} = \text{PCLK}/4$, N=4								
							101: $F_{\text{sample}} = \text{PCLK}/4$, N=6								
							110: $F_{\text{sample}} = \text{PCLK}/8$, N=4								
							111: $F_{\text{sample}} = \text{PCLK}/8$, N=6								
b3		Res		保留位			保留，必须保持复位值								
b2~b0		STEP		TRIM 初始步进量配置			000: 0x10							RW	
							001: 0x02								
							010: 0x04								
							011: 0x08								
							100: 0x10								
101: 0x20															
110: 0x40															
111: 0x80															

注意: 该寄存器只有在 EN 为 0 时才可以更改。

9.5.5 控制寄存器 1 (CTRIM_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							OST	PRS				AUT O	MD		EN

位	标记	位名	功能	读写
b31~b9	Res	保留位	保留, 必须保持复位值	
b8	OST	校准模式配置	0: 实时校准模式, TRIM 过程持续进行 1: 单次校准模式, STEP变为1时结束	RW
b7~b4	PRS	参考时钟预分频配置	0000: 保留, 不可配置 1000: 256 分频 0001: 2 分频 1001: 512 分频 0010: 4 分频 1010: 1024 分频 0011: 8 分频 1011: 2048 分频 0100: 16 分频 1100: 4096 分频 0101: 32 分频 1101: 8192 分频 0110: 64 分频 1110: 16384 分频 0111: 128分频 1111: 32768分频	RW
b3	AUTO	RCH / RCL 自动 TRIM 使能配置	1: 使能 0: 禁止	RW
b2~b1	MD	工作模式配置	00: RCH 校准模式, CR0.SRC 需选择低速精准时钟 01: RCL 校准模式, CR0.SRC 需选择高速精准时钟 1x: 定时器模式	RW
b0	EN	定时器使能控制	1: 使能 0: 禁止	RW

9.5.6 中断使能寄存器 (CTRIM_IER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										OK	OV	MISS	PS	END	UD

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
b5	OK	自动 Trim 结果在范围内中断使能配置	1: 使能 0: 禁止	RW
b4	OV	TrimCode 溢出中断使能配置	1: 使能 0: 禁止	RW
b3	MISS	计数失败中断使能配置	1: 使能 0: 禁止	RW
b2	PS	GCLK 参考时钟上升沿中断使能配置	1: 使能 0: 禁止	RW
b1	END	自动 Trim 结束中断使能配置	1: 使能 0: 禁止	RW
b0	UD	误差计数器下溢中断使能配置	1: 使能 0: 禁止	RW

9.5.7 中断和状态寄存器 (CTRIM_ISR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									DIR	OK	OV	MISS	PS	END	UD

位	标记	位名	功能	读写
b31~b7	Res	保留位	保留, 必须保持复位值	
b6	DIR	误差计数器计数方向	0: 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 1: 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
b5	OK	Trim 精度标志	1: Trim 完成时精度满足 FLIM 要求 0: Trim完成时精度不满足FLIM要求	RO
b4	OV	TrimCode 溢出标志	1: RCH/RCL 的 TrimCode 已调整为 0x000 或 0x1FF 0: RCH/RCL的TrimCode不为0x000或0x1FF	RO
b3	MISS	计数失败标志	1: 在 GCLK 的一个周期内, 误差计数器已计数到 ARR 0: 在GCLK的一个周期内, 误差计数器未计数到ARR	RO
b2	PS	GCLK 参考时钟上升沿标志	1: 检测到 GCLK 参考时钟上升沿 0: 未检测到GCLK参考时钟上升沿	RO
b1	END	自动 Trim 结束标志	1: 自动 Trim 已结束 0: 自动Trim进行中	RO
b0	UD	误差计数器下溢标志	1: 误差计数器已计数到 0 0: 误差计数器未计数到0	RO

9.5.8 中断状态清除寄存器 (CTRIM_ICR)

复位值: 0x0000 003F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									OK	OV	MISS	PS	END	UD	

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
b5	OK	清除 OK 标志位	写0清除校准成功OK标志	R1W0
b4	OV	清除 OV 标志位	写0清除TrimCode溢出标志	R1W0
b3	MISS	清除 MISS 标志位	写0清除计数失败标志	R1W0
b2	PS	清除 PS 标志位	写0清除GCLK参考时钟上升沿标志	R1W0
b1	END	清除 END 标志位	写0清除自动Trim结束标志	R1W0
b0	UD	清除 UD 标志位	写0清除误差计数器下溢标志	R1W0

9.5.9 误差结果寄存器 (CTRIM_FCAP)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FCAP															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留，必须保持复位值												
频率误差捕获值															
b15~b0	FCAP	频率误差捕获值	注： 1) 在每个GCLK周期结束时，自动捕捉误差计数器的计数值存入FCAP寄存器										RO		

9.5.10 TRIM 结果寄存器 (CTRIM_TVAL)

复位值: 0x0000 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TVAL															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留，必须保持复位值												
自动 TRIM 结果															
b15~b0	TVAL	自动TRIM结果	注： 1) 自动 TRIM 结束后，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		

9.5.11 误差上限寄存器 (CTRIM_FLIM)

复位值: 0x0000 0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								FLIM							
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7~b0	FLIM	误差允许值	误差允许值										RW		

10 FLASH 控制器 (FLASH)

10.1 概述

本系统包含一块 128K 字节(Byte)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。本控制器支持的读写位宽为 half-word (16 位) 和 word (32 位)。**注意:** half-word 操作的目标地址必须按半字对齐(地址最低比特为 0), word 操作的地址必须按字对齐(地址最低两比特均为 0)。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。

10.2 容量划分

本控制器支持的 FLASH 容量为 128KB, 每个页 (Sector) 的容量为 512 字节(Byte), 其组成方式如下方所示。

表 10-1 FLASH 容量划分

地址	序号	地址	序号
0x00000 - 0x001FF	Sector0		
0x00200 - 0x003FF	Sector1	0x1F000 - 0x1F1FF	Sector248
0x00400 - 0x005FF	Sector2	0x1F200 - 0x1F3FF	Sector249
0x00600 - 0x007FF	Sector3	0x1F400 - 0x1F5FF	Sector250
0x00800 - 0x009FF	Sector4	0x1F600 - 0x1F7FF	Sector251
0x00A00 - 0x00BFF	Sector5	0x1F800 - 0x1F9FF	Sector252
0x00C00 - 0x00DFF	Sector6	0x1FA00 - 0x1FBFF	Sector253
0x00E00 - 0x00FFF	Sector7	0x1FC00 - 0x1DFFF	Sector254
.....		0x1FE00 - 0x1FFFF	Sector255

10.3 读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz, 故当 CPU 频率 (HCLK) 大于 24MHz 时, CPU 从 FLASH 取指必需要插入等待周期(设置 FLASH_WAIT 大于 0)。当 FLASH_WAIT 等于 1 时, 每两个 HCLK 可以完成一次 FLASH 取指操作。

10.4 FLASH 操作（读、写、擦）

10.4.1 页擦除（Sector Erase）

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_IFR.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_IFR.BUSY 变为 0）。

Sector 擦除操作步骤如下：

- Step1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step2. 配置 FLASH_CR.OP 为 2，设置 Flash 操作模式为 Sector 擦除；
- Step3. 检查 FLASH_CR.OP 是否为 2，如不为 2 则跳转到 Step1；
- Step4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step5. 设置 FLASH_SLOCKx 相应的比特为 1，去除该 Sector 的擦写保护；
- Step6. 检查 FLASH_SLOCKx 相应的比特是否为 1，如不为 1 则跳转到 Step4；
- Step7. 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；
例： $* ((\text{unsigned int} *) 0x00000200) = 0x00000000$ 。
- Step8. 等待 FLASH_IFR.BUSY 变为 0，Sector 擦除操作完成；
- Step9. 如需擦除其它 Sector，重复 Step4 - Step8。如完成擦除，则跳转到 Step10；
- Step10. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step11. 配置 FLASH_CR.OP 为 0，设置 Flash 操作模式为读；
- Step12. 检查 FLASH_CR.OP 是否为 0，如不为 0 则跳转到 Step10；如为 0，结束操作。

10.4.2 全片擦除（Chip Erase）

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，并置起相应的出错标志位；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_IFR.BUSY 变为 0）。

Chip 擦除操作步骤如下：

- Step1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step2. 配置 FLASH_CR.OP 为 3，设置 Flash 操作模式为 Chip 擦除；

- Step3. 检查 FLASH_CR.OP 是否为 3，如不为 3 则跳转到 Step1；
- Step4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step5. 设置 FLASH_SLOCKx 为 0xFFFFFFFF，去除所有 Sector 的擦写保护；
- Step6. 检查 FLASH_SLOCKx 是否为 0xFFFFFFFF，如不为 0xFFFFFFFF 则跳转到 Step4；
- Step7. 对待擦除的 Chip 内的任意地址进行写操作，触发 Chip 擦除；
- 例： $* ((\text{unsigned int } *) 0x00000000) = 0x00$ 。
- Step8. 等待 FLASH_IFR.BUSY 变为 0，Chip 擦除操作完成；
- Step9. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step10. 配置 FLASH_CR.OP 为 0，设置 Flash 操作模式为读；
- Step11. 检查 FLASH_CR.OP 是否为 0，如不为 0 则跳转到 Step10；如为 0，结束操作。

10.4.3 写操作 (Program)

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。支持写入 2 种数据长度：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_IFR.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_IFR.BUSY 变为 0)。

Half-word 写操作步骤如下：

- Step1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step2. 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入；
- Step3. 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step1；
- Step4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step5. 设置 FLASH_SLOCKx 相应的比特为 1，去除擦写保护；
- Step6. 检查 FLASH_SLOCKx 相应的比特是否为 1，如不为 1 则跳转到 Step4；
- Step7. 对待写入的目标地址进行 Half-word 写操作，触发写入操作；
- 例： $* ((\text{unsigned short int } *) 0x00001232) = 0xABCD$ 。
- Step8. 等待 FLASH_IFR.BUSY 变为 0，写入操作完成；
- Step9. 如需写 Half-word 到其它地址，重复 Step7 - Step8。如完成写，则跳转到 Step10；

- Step10. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step11. 配置 FLASH_CR.OP 为 0，设置 Flash 操作模式为读；
- Step12. 检查 FLASH_CR.OP 是否为 0，如不为 0 则跳转到 Step10；如为 0，结束操作。

Word 写操作步骤如下：

- Step1. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step2. 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入；
- Step3. 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step1；
- Step4. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step5. 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护；
- Step6. 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step4；
- Step7. 对待写入的目标地址进行 Word 写操作，触发写入操作；
- 例： $* ((\text{unsigned int } *) 0x00001234) = 0x55667788。$
- Step8. 等待 FLASH_IFR.BUSY 变为 0，写入操作完成；
- Step9. 如需写 Word 到其它地址，重复 Step7 - Step8。如完成写，则跳转到 Step10；
- Step10. 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写；
- Step11. 配置 FLASH_CR.OP 为 0，设置 Flash 操作模式为读；
- Step12. 检查 FLASH_CR.OP 是否为 0，如不为 0 则跳转到 Step10；如为 0，结束操作。

10.4.4 读操作 (Read)

支持读出 3 种数据长度：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。

例：

Byte 读操作： $\text{temp} = * ((\text{unsigned char } *) 0x0000 1231)$

Half-word 读操作 $\text{temp} = * ((\text{unsigned short int } *) 0x0000 1232)$

Word 读操作 $\text{temp} = * ((\text{unsigned int } *) 0x0000 1234)$

10.4.5 Cache 功能

支持数据与指令的 cache 缓存功能，如果系统工作频率大于 24M 开启 wait 等待周期 FLASH_WAIT 大于 0 时，开启 cache 指令与数据缓存功能将可以提高系统性能。

注意：对于 FLASH 进行擦和写操作，必须关闭 cache。

10.5 FLASH 安全保护

10.5.1 页面擦写保护

FLASH 存储器共 128K 字节，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。

10.5.2 PC 地址擦写保护

若 CPU 在 FLASH 中执行程序且当前 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。基于该特性，只有从 RAM 中执行代码时，才可以对 FLASH 进行片擦操作。

10.5.3 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如下所示：

FLASH_CR、FLASH_SLOCKx、FLASH_WAIT、FLASH_CACHE。

不需要通过写序列方式即可更改的寄存器如下所示：

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：

- Step1. 向 FLASH_BYPASS 寄存器写入 0x5A5A；
- Step2. 向 FLASH_BYPASS 寄存器写入 0xA5A5；
- Step3. 对待修改的寄存器写入目标值；
- Step4. 验证待修改的寄存器当前值是否与目标相同，如不则跳转到 Step1；
- Step5. 执行其它操作。

注意：

- 写 0x5A5A 和写 0xA5A5 这两步操作之间不可插入任何写操作，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。

10.5.4 数据读出保护

本芯片内置 FLASH 支持 4 级读保护，当前保护等级可通过 FLASH_LOCKSTATE 寄存器读出。使用编程器对芯片进行编程时，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。

表 10-2 芯片保护等级表

保护等级	芯片功能
Level0	可以通过ISP及SWD对FLASH进行读写
Level1	数据不可读出 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 可以通过ISP降级到Level0，降级后FLASH内容全为FF SWD接口被禁止，数据不可读出
Level3	数据不可读出 ISP及SWD接口均被禁止，即芯片只能进行一次编程

10.6 寄存器描述

10.6.1 寄存器列表

基地址：0x4002 1400

表 10-3 FLASH 寄存器列表

寄存器名	偏移地址	复位值
控制寄存器 (FLASH_CR)	0x00	0x0080 (高16位保留)
等待周期设置 (FLASH_WAIT)	0x04	0x0000 (高16位保留)
LOCK状态寄存器 (FLASH_LOCKSTATE)	0x08	0x0000 (高16位保留)
CACHE控制寄存器 (FLASH_CACHE)	0x0C	0x0000 (高16位保留)
中断使能寄存器 (FLASH_IER)	0x10	0x0000 (高16位保留)
中断标志寄存器 (FLASH_IFR)	0x14	0x000F (高16位保留)
中断标志清除寄存器 (FLASH_ICR)	0x18	0x0000 (高16位保留)
Bypass序列寄存器 (FLASH_BYPASS)	0x1C	0x0000 (高16位保留)
Sector擦写保护寄存器 (FLASH_SLOCK0)	0x30	0x0000 0000
Sector擦写保护寄存器 (FLASH_SLOCK1)	0x34	0x0000 0000

10.6.2 控制寄存器 (FLASH_CR)

复位值: 0x0080

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res								DPSTB_EN	Res						OP	
位	标记		位名		功能						读写					
b15~b8		Res	保留位		保留，必须保持复位值											
b7		DPSTB_EN	FLASH dpstb使能Mask位		0：当系统进入deepsleep模式，FLASH不进入低功耗模式； 1：当系统进入deepsleep模式，FLASH进入低功耗模式；						RW					
b6~b2		Res	保留位		保留，必须保持复位值											
b1~b0		OP	FLASH 操作		00：读（read）；01：写（program）； 10：页擦除（sector erase）；11：全片擦除（chip erase） 注：软件对 Flash 执行擦除或写操作之后，需要重新将 FLASH_CR.OP 控制位设置为 0。						RW					

10.6.3 等待周期设置 (FLASH_WAIT)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												WAIT			
位	标记	位名	功能											读写	
b15~b4	Res	保留位	保留, 必须保持复位值												
b3~b0	WAIT	读FLASH 周期数	0~24MHz: 0000: 1个周期; 24~48MHz: 0001: 2个周期; 48~72MHz: 0010: 3个周期; 72~96MHz: 0011: 4个周期; 禁止设置其它值 (本系列产品最高时钟为84MHz)											RW	

10.6.4 LOCK 状态寄存器 (FLASH_LOCKSTATE)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														STATE	
位	标记	位名	功能											读写	
b15~b2	Res	保留位	保留, 必须保持复位值												
b1~b0	STATE	FLASH 保护状态	00: Level0, ISP可读写, SWD可读写 01: Level1, ISP可降级, SWD可降级; 数据不可读出 10: Level2, ISP可降级, SWD无功能; 数据不可读出 11: Level3, ISP无功能, SWD无功能; 数据不可读出											RO	

10.6.5 CACHE 控制寄存器 (FLASH_CACHE)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													INVA LID	DEN	IEN
位	标记	位名	功能		读写										
b15~b3	Res	保留位	保留, 必须保持复位值												
b2	INVALID	cache 复位	先写1, 再写0, 实现cache复位		RW										
b1	DEN	data cache enable	1: 使能数据cache 0: 关闭数据 cache		RW										
b0	IEN	instruction cache enable	1: 使能指令cache 0: 关闭指令cache		RW										

10.6.6 中断使能寄存器 (FLASH_IER)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														PROT	PC
位	标记	位名	功能		读写										
b15~b2	Res	保留位	保留, 必须保持复位值												
b1	PROT	FLASH擦写保护报警中断使能	1: 使能FLASH擦写保护报警中断 0: 禁止FLASH擦写保护报警中断		RW										
b0	PC	FLASH擦写 PC 地址报警中断使能	1: 使能FLASH擦写 PC 地址报警中断 0: 禁止FLASH擦写 PC 地址报警中断		RW										

10.6.7 中断标志寄存器 (FLASH_IFR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											BUSY	Res		PROT	PC
位	标记	位名	功能										读写		
b15~b5	Res	保留位	保留，必须保持复位值												
b4	BUSY	FLASH总线忙标志位	1：FLASH总线当前处于忙状态 0：FLASH 总线当前处于空闲状态										RO		
b3~b2	Res	保留位	保留，必须保持复位值												
b1	PROT	FLASH擦写保护报警中断标志位	1：发生FLASH擦写保护报警中断 0：未发生FLASH擦写保护报警中断										RO		
b0	PC	FLASH擦写 PC 地址报警中断标志位	1：发生FLASH擦写 PC 地址报警中断 0：未发生FLASH擦写 PC 地址报警中断										RO		

10.6.8 中断标志清除寄存器 (FLASH_ICR)

复位值：0x000F

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														PROT	PC
位	标记	位名	功能		读写										
b15~b2	Res	保留位	保留，必须保持复位值												
b1	PROT	清除保护报警中断标志位	1：无效 0：清除		R1W0										
b0	PC	清除 PC 地址报警中断标志位	1：无效 0：清除		R1W0										

10.6.9 Bypass 序列寄存器 (FLASH_BYPASS)

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BYSEQ															
位	标记	位名	功能		读写										
b15~b0	BYSEQ	Bypass序列位	在修改本模块寄存器之前，必须对 BYSEQ[15:0] 寄存器写入 0x5a5a-0xa5a5 序列。每次写入该 Bypass 序列后，只可以修改一次寄存器。如需再次修改寄存器，必须再次输入 0x5a5a-0xa5a5 序列。		RW										

10.6.10 Sector 擦写保护寄存器 (FLASH_SLOCK0)

复位值：0x0000 0000（外设复位无效）

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SLOCK[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLOCK[15:0]															
位	标记	位名	功能		读写										
			0：不允许擦写；1：允许擦写												
			Bit[0] 对应：Sector0~3												
			Bit [1] 对应：Sector4~7												
			Bit [2] 对应：Sector8~11												
			Bit [3] 对应：Sector12~15												
															
			Bit [30] 对应：Sector120~123												
			Bit [31] 对应：Sector124~127												

10.6.11 Sector 擦写保护寄存器 (FLASH_SLOCK1)

复位值：0x00000000（外设复位无效）

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SLOCK[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLOCK[15:0]															

位	标记	位名	功能	读写
b31~b0	SLOCK	Sector擦写保护位	0：不允许擦写；1：允许擦写	RW
			Bit[0]对应：Sector128~131	
			Bit[1]对应：Sector132~135	
			Bit[2]对应：Sector136~139	
				
			Bit[30]对应：Sector248~251	
			Bit[31]对应：Sector252~255	

11 RAM 控制器（RAM）

11.1 概述

本产品中包含一块起始地址为 0x0x2000_0000，容量高达 24K 字节（Byte）的 SRAM，支持字节（8 位）、半字（16 位）、字（32 位）三种读写操作，零等待周期。

11.2 功能描述

11.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示：

表 11-1 RAM 地址映射

地址范围	大小	Memory类型
0x2000 0000 - 0x2000 1FFF	8KByte	SRAM
0x2000 0000 - 0x2000 3FFF	16KByte	
0x2000 0000 - 0x2000 5FFF	24KByte	

12 DMA 控制器 (DMAC)

12.1 DMAC 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 6 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, 且有一个仲裁器来协调各个 DMA 请求的优先权。

12.2 DMAC 主要特性

- 6 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: 0x0000 0000 ~ 0xFFFF FFFF
- 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

12.3 DMA 功能框图

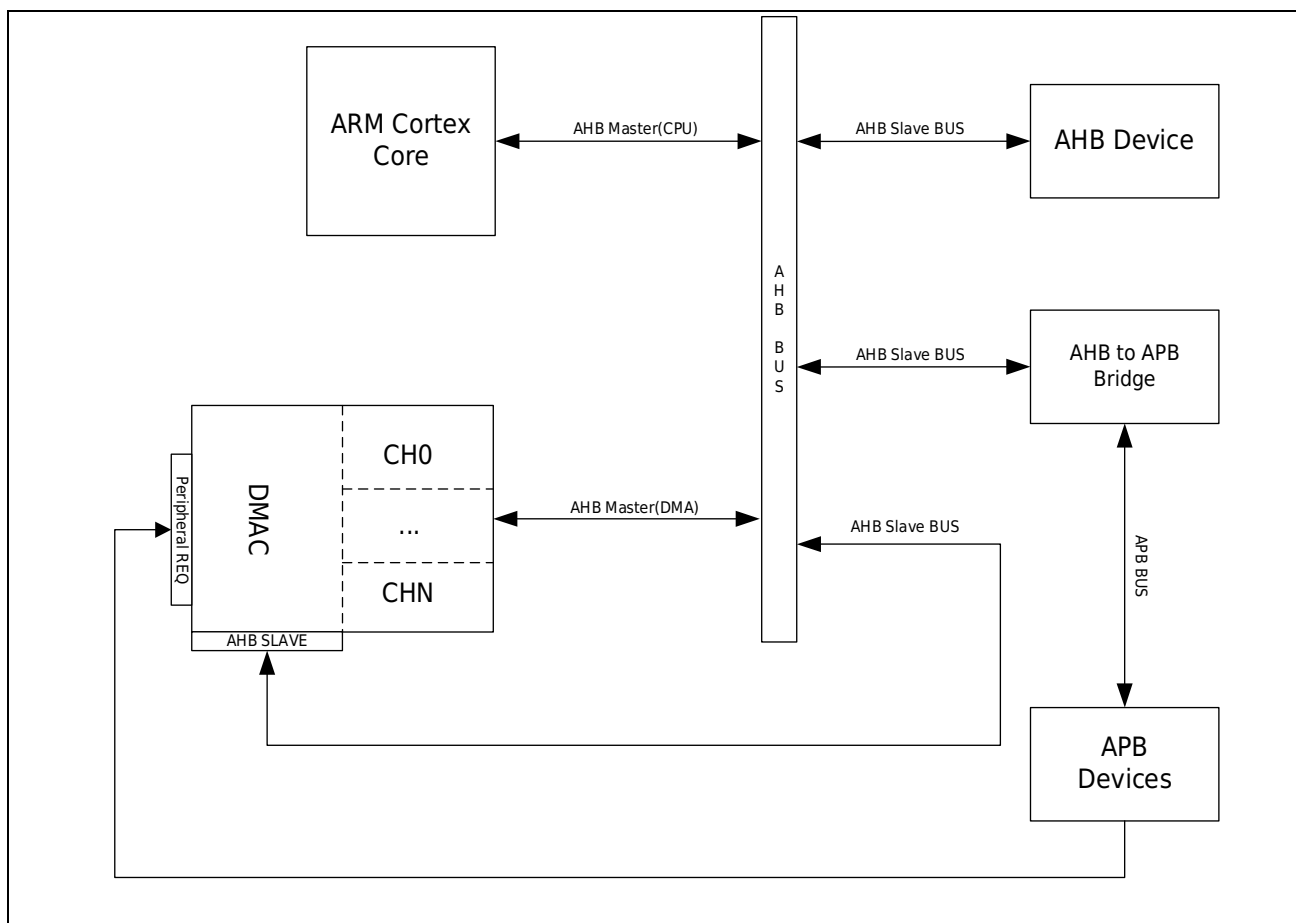


图 12-1 DMA 功能框图

- DMAC

DMAC 具有 6 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。

- 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。

12.4 功能描述

12.4.1 DMAC 传输模式概述

本 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。

4 种传输模式对比如下方表格所示：

表 12-1 DMA 传输模式表

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 - 存储器到有 DMA 请求的外设

12.4.2 DMA 软件 Block 传输模式

当配置 DMAC_CONFAX.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_CONFAX.ST 写入 1, 则触发 DMAC 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 会插入一个传输间隙, 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 16bit, 地址自增。

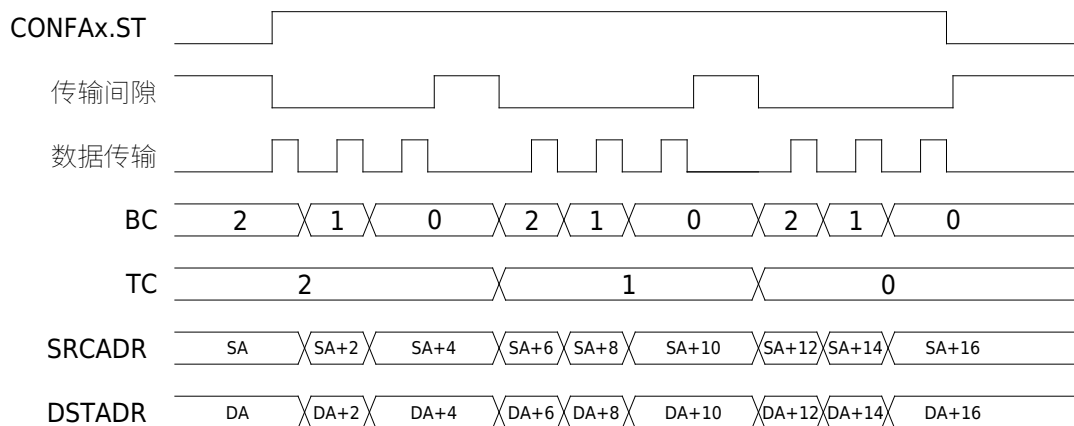


图 12-2 DMA 软件 Block 传输

12.4.3 DMA 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 16bit, 地址自增。

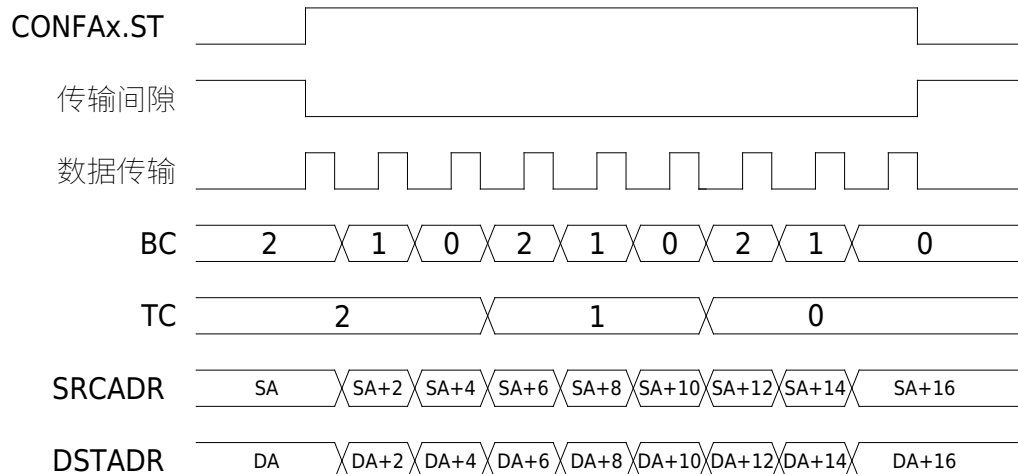


图 12-3 DMA 软件 Burst 传输

12.4.4 DMA 硬件 Block 传输模式

当配置 DMAC_CONFAX.TRI_SEL 为相应触发源, 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 会插入一个传输间隙, 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。

注意: 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 信号如下所示:

- UARTx 接收 Buf 非空、发送 Buf 空
- SPIx 接收 Buf 非空、发送 Buf 空
- GTIMx 捕获完成、比较匹配
- ATIMx 捕获完成、比较匹配
- ADCx 顺序单次转换完成

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 8bit, 目标地址自增, 源地址固定, 传输完成自动禁止。

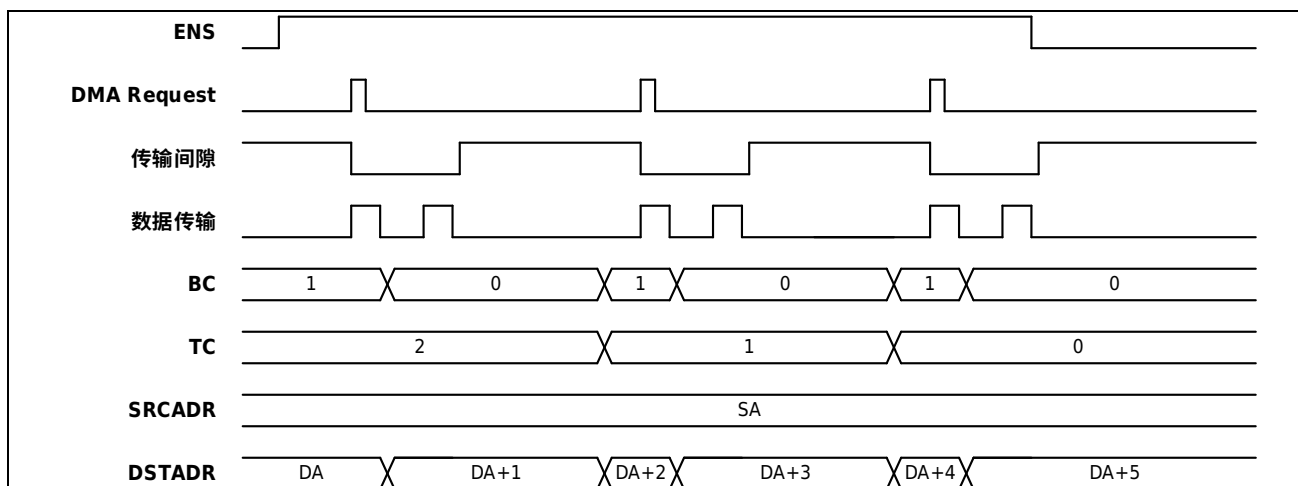


图 12-4 DMA 硬件 Block 传输

12.4.5 DMA 硬件 Burst 传输模式

当配置 DMAC_CONF x .TRI_SEL=非零，DMAC_CONF x .MODE=1 时，DMAC 工作于硬件 Burst 传输模式。

该模式下一旦外设 DMA request 信号出现，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。

DMAC 支持的硬件 request 信号如下所示：

- UART x 接收 Buf 非空、发送 Buf 空
- SPI x 接收 Buf 非空、发送 Buf 空
- GTIM x 捕获完成、比较匹配
- ATIM x 捕获完成、比较匹配
- ADC 顺序单次转换完成

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 8bit，源地址自增，目标地址固定，传输完成自动禁止。

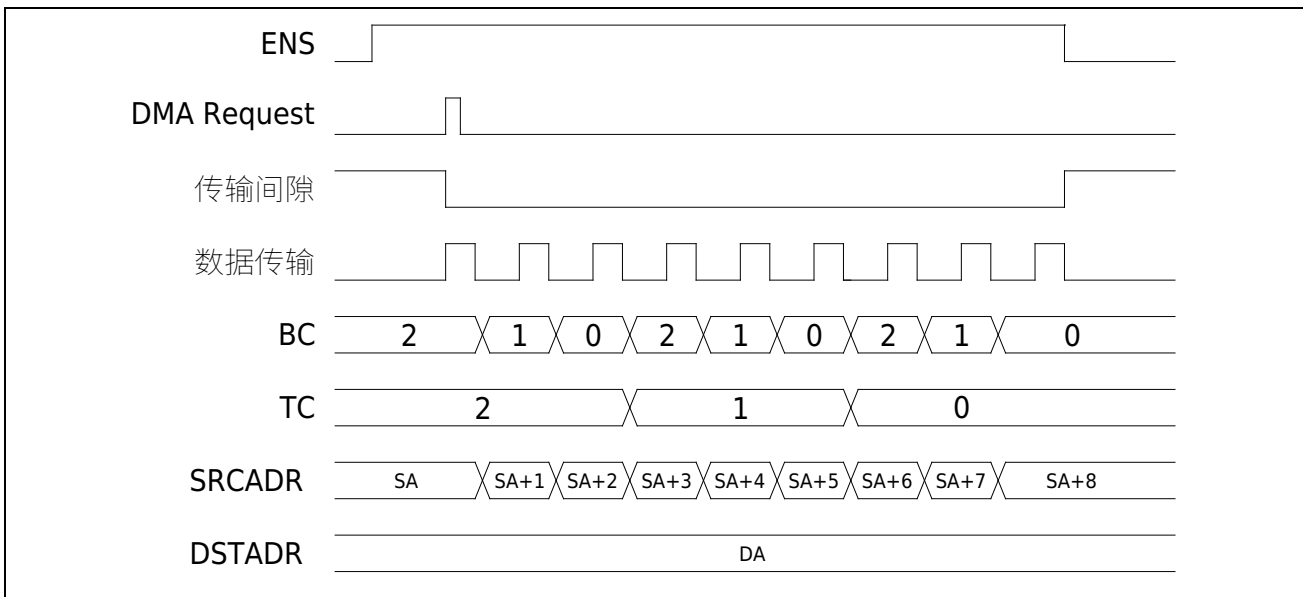


图 12-5 DMA 硬件 Burst 传输

12.4.6 DMA 传输暂停

当 DMAC 工作于 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 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，DMAC 需要收到下一次触发信号才会继续刚才未完成的传输。

当 DMAC 工作于 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。当 DMAC 工作于 Burst 传输模式且没有进行数据传输时，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 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，且当前正处于数据传输的 DMA 通道在数据传输完成后也不会进入暂停态。

12.4.7 DMA 其它配置

12.4.7.1 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 8bit、16bit、32bit，注意该位宽应与 DMA 的源地址与目标地址的位宽完全一致。

12.4.7.2 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 UART / SPI 每次只能向 DMA 提供一个数据，UART / 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。

12.4.7.3 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。

12.4.7.4 通道优先级

当 DMAC_CONF.PRIO=0 时，DMAC 工作在固定优先模式下。此时优先级顺序为：CH0>CH1>CH2>CH3>CH4>CH5，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=1 时，DMAC 工作在循环优先模式下。此时 CH0、CH1、CH2、CH3、CH4 和 CH5 轮流占有总线以进行数据传输。

12.4.7.5 自动重装载

如果使能自动重装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。

12.4.8 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。

5 种触发 DMA 中断的条件如下：

- 传输地址超出寻址范围
- 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

12.5 寄存器描述

12.5.1 寄存器列表

基地址：0x4002 0000

表 12-2 DMA 寄存器列表

寄存器名	偏移地址	复位值
所有通道配置寄存器 (DMAC_CONF)	0x00	0x0000 0000
通道 0 配置 A 寄存器 (DMAC_CONFA0)	0x10	0x0000 0000
通道 0 配置 B 寄存器 (DMAC_CONFB0)	0x14	0x0000 0000
通道 0 传输源地址寄存器 (DMAC_SRCADR0)	0x18	0x0000 0000
通道 0 传输目的地址寄存器 (DMAC_DSTADR0)	0x1C	0x0000 0000
通道 1 配置 A 寄存器 (DMAC_CONFA1)	0x20	0x0000 0000
通道 1 配置 B 寄存器 (DMAC_CONFB1)	0x24	0x0000 0000
通道 1 传输源地址寄存器 (DMAC_SRCADR1)	0x28	0x0000 0000
通道 1 传输目的地址寄存器 (DMAC_DSTADR1)	0x2C	0x0000 0000
通道 2 配置 A 寄存器 (DMAC_CONFA2)	0x30	0x0000 0000
通道 2 配置 B 寄存器 (DMAC_CONFB2)	0x34	0x0000 0000
通道 2 传输源地址寄存器 (DMAC_SRCADR2)	0x38	0x0000 0000
通道 2 传输目的地址寄存器 (DMAC_DSTADR2)	0x3C	0x0000 0000
通道 3 配置 A 寄存器 (DMAC_CONFA3)	0x40	0x0000 0000
通道 3 配置 B 寄存器 (DMAC_CONFB3)	0x44	0x0000 0000
通道 3 传输源地址寄存器 (DMAC_SRCADR3)	0x48	0x0000 0000
通道 3 传输目的地址寄存器 (DMAC_DSTADR3)	0x4C	0x0000 0000
通道 4 配置 A 寄存器 (DMAC_CONFA4)	0x50	0x0000 0000
通道 4 配置 B 寄存器 (DMAC_CONFB4)	0x54	0x0000 0000
通道 4 传输源地址寄存器 (DMAC_SRCADR4)	0x58	0x0000 0000
通道 4 传输目的地址寄存器 (DMAC_DSTADR4)	0x5C	0x0000 0000
通道 5 配置 A 寄存器 (DMAC_CONFA5)	0x60	0x0000 0000
通道 5 配置 B 寄存器 (DMAC_CONFB5)	0x64	0x0000 0000
通道 5 传输源地址寄存器 (DMAC_SRCADR5)	0x68	0x0000 0000
通道 5 传输目的地址寄存器 (DMAC_DSTADR5)	0x6C	0x0000 0000

12.5.2 所有通道配置寄存器 (DMAC_CONF)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN	Res		PRI0	HALT				Res							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															

位	标记	位名	功能	读写
b31	EN	DMA控制器使能	1: 使能DMA控制器 0: 禁止DMA控制器 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
b30~b29	Res	保留位	保留, 必须保持复位值	
b28	PRI0	DMA通道优先级配置	DMA通道优先级配置 1: 循环优先级模式, CH0/CH1/CH2/CH3/CH4/CH5轮流获取总线访问权 0: 固定优先级模式, CH0>CH1>CH2>CH3>CH4>CH5优先级	RW
b27~b24	HALT	DMA全部通道传输暂停控制	DMA全部通道传输暂停控制 0000: 恢复所有通道数据传输 xxxx: 暂停所有通道数据传输	RW
b23~b0	Res	保留位	保留, 必须保持复位值	

12.5.3 通道 x 配置 A 寄存器 (DMAC_CONFAX)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ENS	PAS	ST	TRI_SEL							Res		BC			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TC															

位	标记	位名	功能	读写
b31	ENS	DMA 通道使能控制	1: 使能当前DMA通道; 若CONFBx.MSK为0, 则传输完成时该位自动清零 0: 禁止当前DMA通道 注意: 当禁止DMA通道时, 将强制停止正在进行的传输, 造成不可预知的结果。	RW
b30	PAS	DMA 通道数据传输暂停控制	1: 暂停DMA通道数据传输 0: 恢复DMA通道数据传输	RW
b29	ST	DMA 通道软件触发控制	1: 触发 DMA 通道启动软件 Block / Burst 传输, 传输完成时该位自动清零 0: 停止DMA通道软件Block / Burst传输	RW
b28~b22	TRI_SEL	DMA 通道触发源配置	0x00: 软件触发	RW
			0x40: USART0 发送 Buf 空	
			0x41: USART0 接收 Buf 非空	
			0x42: USART1 发送 Buf 空	
			0x43: USART1 接收 Buf 非空	
			0x44: USART2 发送 Buf 空	
			0x45: USART2 接收 Buf 非空	
			0x46: USART3 发送 Buf 空	
			0x47: USART3 接收 Buf 非空	
			0x48: GTIM0 发生捕获事件或比较事件	
			0x49: GTIM1 发生捕获事件或比较事件	
			0x4A: GTIM2 发生捕获事件或比较事件	
			0x4B: Reserved	
			0x4C: Reserved	
			0x4D: ADC0 顺序单次转换完成	
			0x4E: Reserved	
			0x4F: ADC1 顺序单次转换完成	
			0x50: SPI0 发送 Buf 空	
			0x51: SPI0 接收 Buf 非空	
			0x52: SPI1 发送 Buf 空	
			0x53: SPI1 接收 Buf 非空	
			0x54: Reserved	
			0x55: Reserved	
			0x56: Reserved	
			0x57: Reserved	
			0x58: ATIM0 通道 A, 发生捕获事件或比较事件或外部触发	
			0x59: ATIM0 通道 B, 发生捕获事件或比较事件或事件更新	

			0x5A: Reserved	
			0x5B: Reserved	
			0x5C: Reserved	
			0x5D: Reserved	
			0x5E: ATIM3 通道 A, 发生捕获事件或比较事件或外部触发	
			0x5F: ATIM3 通道 B, 发生捕获事件或比较事件或事件更新	
			0x60~0x7F: Reserved	
b21~b20	Res	保留位	保留, 必须保持复位值	
b19~b16	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 注意: 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
b15~b0	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 注意: DMA 传输的总数据量为 (TC + 1) * (BC + 1)	RW

12.5.4 通道 x 配置 B 寄存器 (DMAC_CONFBx)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MODE	WIDTH	FS	FD	RC	RS	RD	ERR_I E	FIS_I E	STAT					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														MSK	

位	标记	位名	功能	读写
b31~b30	Res	保留位	保留, 必须保持复位值	
b29~b28	MODE	DMA通道传输模式配置	00: Block 传输 01: Burst 传输 10: Reserved 11: Reserved	RW
b27~b26	WIDTH	DMA传输数据位宽	00: 8bit 01: 16bit 10: 32bit 11: Reserved	RW
b25	FS	DMA通道源地址自增使能	1: 源地址在传输过程中保持不变 0: 每传输完成一个数据, 源地址进行自增; 数据宽度为 8bit / 16bit / 32bit 所对应的自增量分别为 1 / 2 / 4	RW
b24	FD	DMA通道目标地址自增使能	1: 目标地址在传输过程中保持不变 0: 每传输完成一个数据, 目标地址进行自增; 数据宽度为8bit / 16bit / 32bit所对应的自增量分别为1 / 2 / 4	RW
b23	RC	DMA通道BC和TC重载使能	1: 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 0: 禁止 BC/TC 重载, 传输完成后 BC/TC 的值均为 0	RW
b22	RS	DMA通道源地址重载使能	1: 使能源地址重载, 传输完成后源地址的值为写入的初值 0: 禁止源地址重载, 传输完成后源地址的值不确定	RW
b21	RD	DMA通道目标地址重载使能	1: 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 0: 禁止目标地址重载, 传输完成后目标地址的值不确定	RW
b20	ERR_IE	DMA通道传输错误中断使能	1: 传输错误时, 产生中断 0: 传输错误时, 禁止中断 注意: 在中断服务程序中, 需要对 STAT 写入 0x00 以清除中断状态	RW
b19	FIS_IE	DMA 通道传输完成中断使能	1: 传输完成时, 产生中断 0: 传输完成时, 禁止中断 注意: 在中断服务程序中, 需要对 STAT 写入 0x00 以清除中断状态	RW
b18~b16	STAT	DMA通道当前传输状态	000: 初始值 001: 传输错误, 传输地址超出寻址范围 010: 传输错误, 外设请求停止 DMA 011: 传输错误, 访问传输来源地址出错 100: 传输错误, 访问传输目的地址出错 101: 传输完成 110: 保留 111: 传输暂停	RW
b15~b1	Res	保留位	保留, 必须保持复位值	

b0	MSK	DMA 通道传输完成时自动禁能配置	1: DMA 通道传输完成时, CONFAX.ENS 保持不变 0: DMA 通道传输完成时, CONFAX.ENS 自动清零	RW
----	-----	-------------------	--	----

12.5.5 通道 x 传输源地址寄存器 (DMAC_SRCADR_x)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SRCADR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SRCADR[15:0]															

位	标记	位名	功能	读写
b31~b0	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

12.5.6 通道 x 目的地址寄存器 (DMAC_DSTADR_x)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DSTADR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DSTADR[15:0]															

位	标记	位名	功能	读写
b31~b0	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

13 复合定时器 (CTIM)

13.1 概述

CTIM 是一个复合定时器，可以配置为一个带有 4 组比较捕获功能的通用定时器 GTIM，也可以配置为一组（三个）只有定时功能的基本定时器 BTIM。通过 SysCtrl_CR1.GTIMxCFG 控制，该位为 1 时 BTIM 有效，该位为 0 时 GTIM 有效。本产品每个 CTIM 可以独立灵活配置为 GTIM 或 BTIM。

BTIM 是一组具有基本定时功能的定时计数器，每组包含 3 个 16 位的具有重载功能的定时器。本产品最多可配置 9 个基本定时器：BTIM0~8。

GTIM 是一个可具有 4 路比较捕获功能的 16 位定时器。本产品最多可配置 3 个通用定时器：GTIM0~2。

表 13-1 管脚功能与定时器功能对照表

CTIM(x=0,1,2)	GTIM(x=0,1,2)	BTIM(y=0,3,6)
CTIMx_ETR	GTIMx_ETR	BTIMy_ETR
CTIMx_TOGP	GTIMx_TOGP	BTIMy+0_TOGP
CTIMx_TOGN	GTIMx_TOGN	BTIMy+0_TOGN
CTIMx_CH0	GTIMx_CH0	BTIMy+1_TOGP
CTIMx_CH1	GTIMx_CH1	BTIMy+1_TOGN
CTIMx_CH2	GTIMx_CH2	BTIMy+2_TOGP
CTIMx_CH3	GTIMx_CH3	BTIMy+2_TOGN

注：

1. BTIM0/1/2 共用 CTIM0_ETR；BTIM3/4/5 共用 CTIM1_ETR；BTIM6/7/8 共用 CTIM2_ETR。
2. 当 x=0 时，y=0；当 x=1 时，y=3；当 x=2 时，y=6。

13.1.1 BTIM 主要特性

- 3 个 16bit 定时器
- 每个定时器 2 个翻转输出
- 单脉冲模式
- 外部输入计数功能
- 门控功能
- 外部触发功能
- 级联功能

13.1.2 GTIM 主要特性

- 16bit 定时器
- 2 个翻转输出
- 单脉冲模式
- 外部输入计数功能
- 门控功能
- 外部触发功能
- 4 路独立 PWM 输出，调整范围 0% ~ 100%
- 4 路独立捕获
- 级联功能

13.2 BTIM 功能描述

13.2.1 功能框图

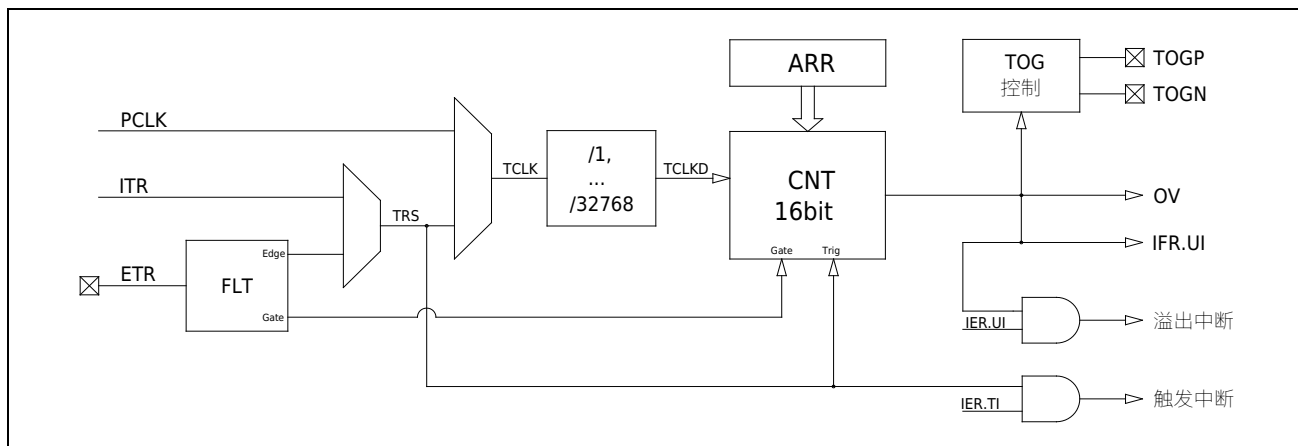


图 13-1 BTIM 框图

注：

1. 一组 3 个 BTIM，共用一个 ETR。
2. ITR 信号来源详见【定时器级联】功能章节。

13.2.2 滤波单元

外部管脚输入的 ETR 信号经滤波、极性选择、边沿检测后，输出门控信号及边沿信号供计数单元及触发单元使用。由于 ETR 信号采用 PCLK 进行同步滤波，故输入的 ETR 信号的频率应低于 PCLK 的频率的二分之一。

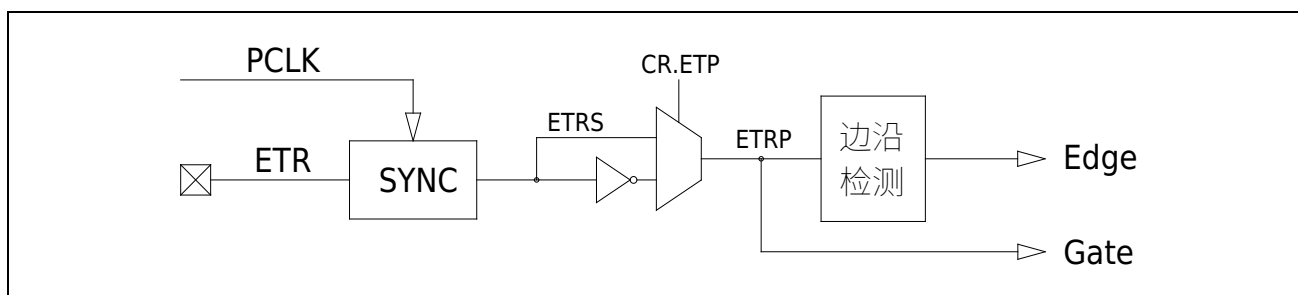


图 13-2 BTIM 滤波单元示意图

13.2.3 计数单元

在每个 TCLKD 信号的上升沿，内部计数器 CNT 自动加 1。计数到 ARR 后产生溢出信号 OV（宽度为一个 PCLK），并复位内部计数值为 0 并再次开始加计数。计数器的计数范围为 0 至 ARR，计数单元的计数周期为 ARR+1。

定时器支持 2 种计数模式，One-Shot 模式与连续计数模式。当设置 BTIMx_CR.OST 为 1 时，定时器工作于 One-Shot 模式；发生溢出后定时器立即停止计数，硬件自动清除 CEN。当设置 BTIMx_CR.OST 为 0 时，定时器工作于连续计数模式；只要 BTIMx_CR.CEN 为 1，定时器会一直进行计数。

每次溢出时，BTIMx_IFR.UI 会被硬件置 1，若使能相应中断则可以产生溢出中断，用户应在中断服务程序中清除该溢出标志。

若 BTIMx_CR.TOGEN 为 1，则每溢出一次 TOGP/TOGN 输出电平翻转一次；用户可通过 GPIO 的复用功能将 TOGP/TOGN 信号输出到管脚用于驱动外部电路。

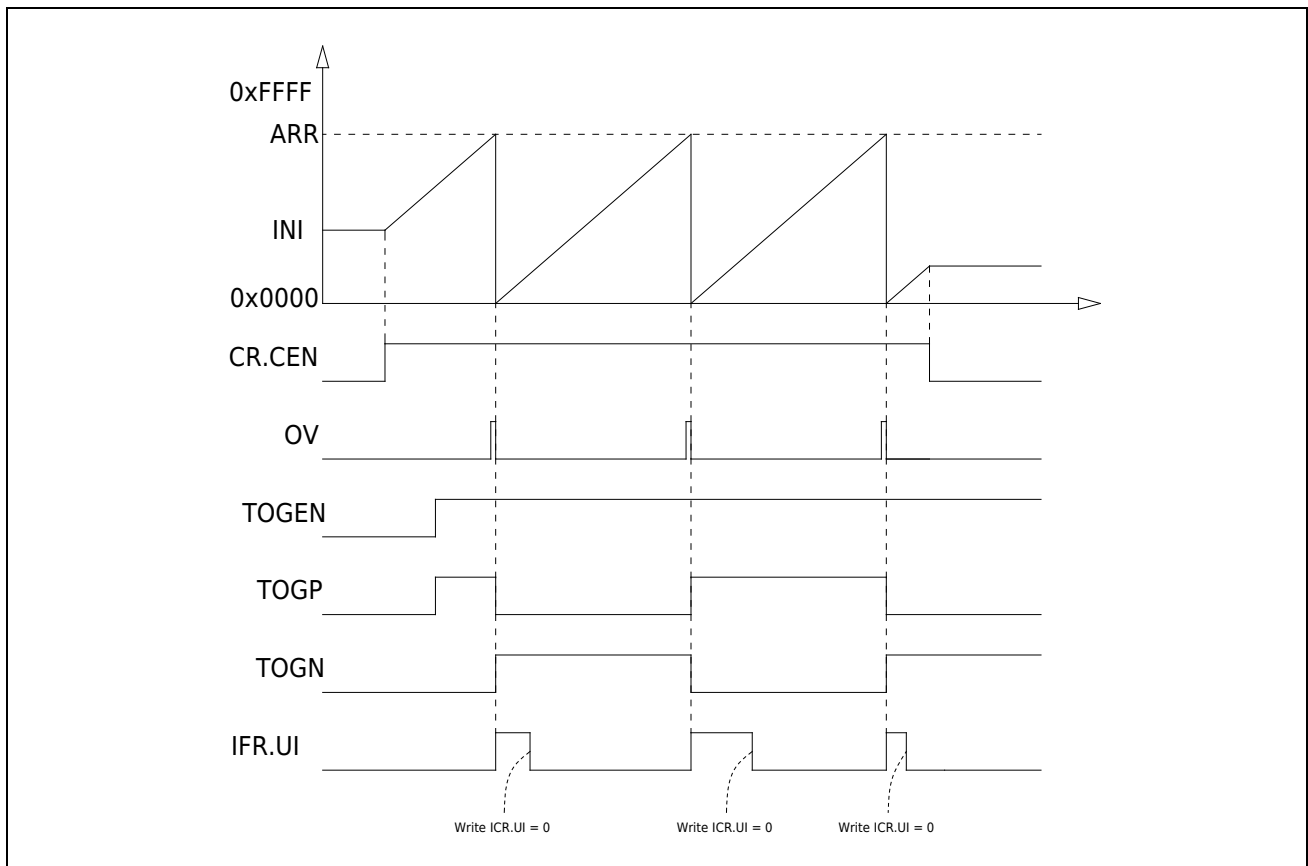


图 13-3 BTIM 计数器波形

13.2.4 定时器模式

当设置 BTIMx_CR.MD 为 0x00 时, BTIMx 工作于定时器模式。在该模式下, 计数时钟为 PCLK 分频后的时钟 TCLKD, 当设置 BTIMx_CR.CEN 为 1 后, 定时器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

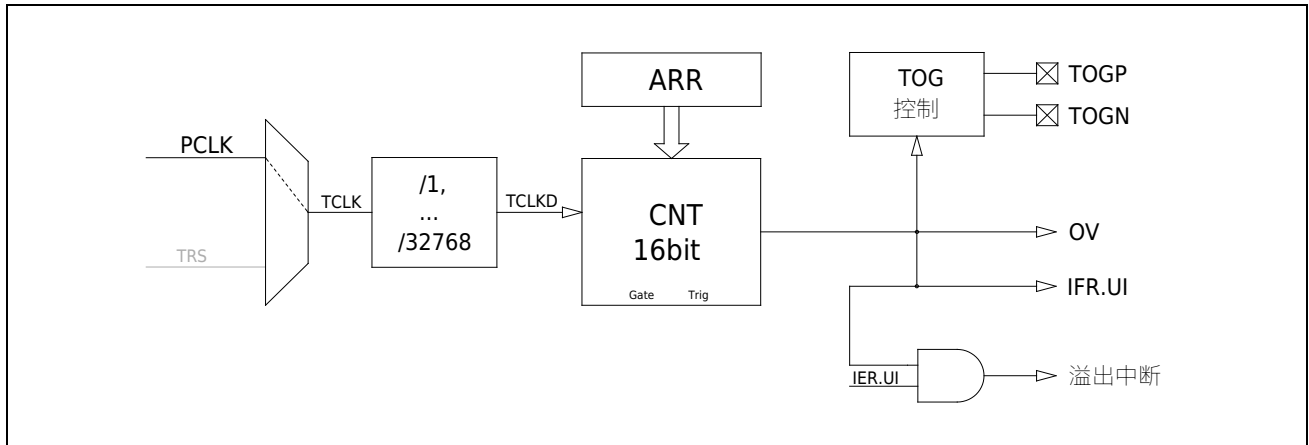


图 13-4 BTIM 定时模式框图

13.2.5 计数器模式

当设置 BTIMx_CR.MD 为 0x01 时, BTIMx 工作于计数器模式。在该模式下, 计数时钟为分频后的 ITR 或 ETR 管脚输入的信号。当设置 BTIMx_CR.CEN 为 1 后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx_CR.TRS 可以选择计数时钟源为 ITR 或 ETR; 通过 BTIMx_CR.ETP 可以选择对 ETR 管脚输入信号的上升沿或下降沿进行计数。

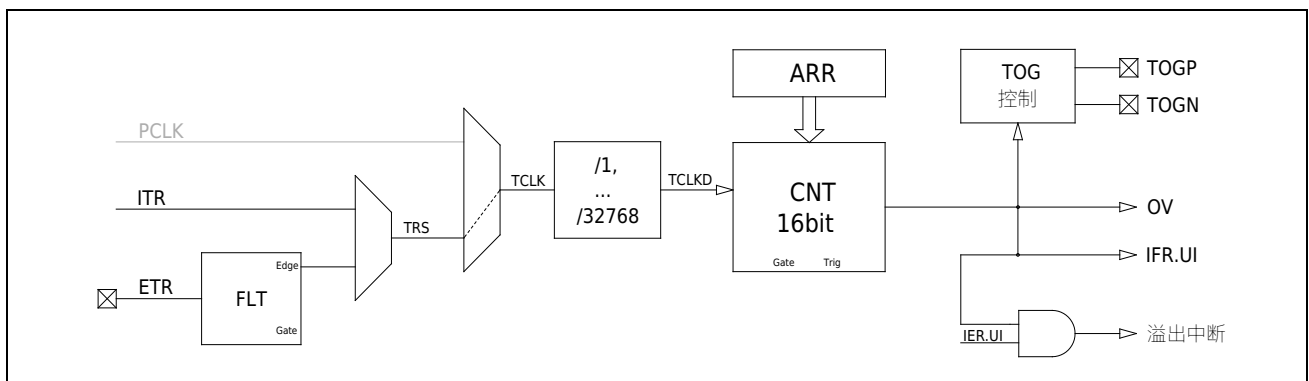


图 13-5 BTIM 计数器模式框图

13.2.6 触发启动模式

当设置 BTIMx_CR.MD 为 0x02 时, BTIMx 工作于触发启动模式, 计数时钟为 PCLK 分频后的时钟 TCLKD。该模式下定时器具有两种启动方法: 设置 BTIMx_CR.CEN 为 1 或 TRS 信号上出现符合预期的电平变化。

启动后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数; 在任意时刻设置 BTIMx_CR.CEN 为 0, 则可以停止定时器。

通过 BTIMx_CR.TRS 选择 TRS 信号来源为 ETR 或 ITR。

通过 BTIMx_CR.ETP 可以选择 ETR 管脚输入信号的上升沿或下降沿启动定时器。

每次检测到符合预期的触发信号时, BTIMx_IFR.TI 会被硬件置 1, 若使能相应中断则可以产生触发中断, 用户应在中断服务程序中清除该触发标志。

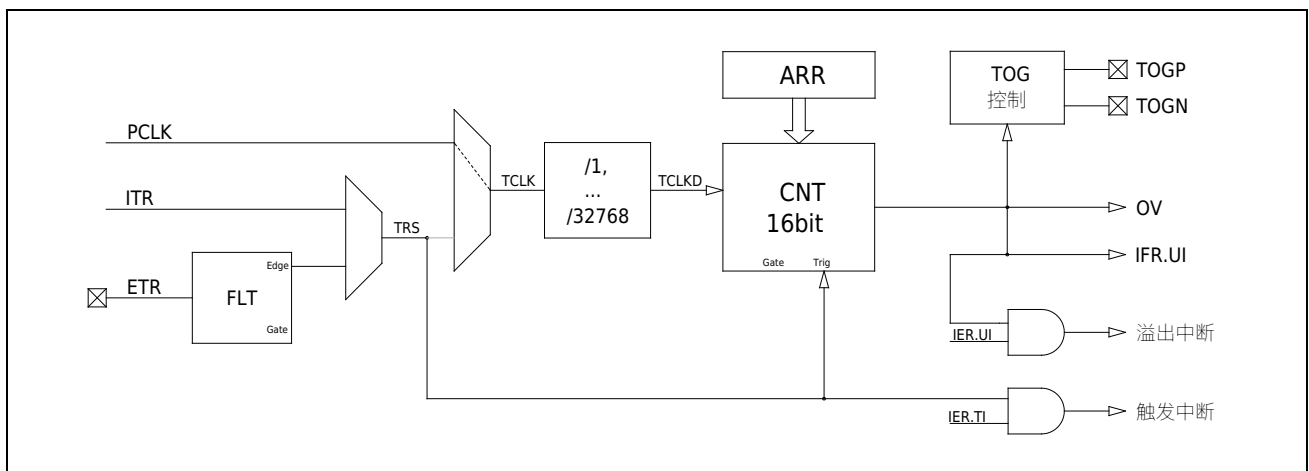


图 13-6 BTIM 触发启动器模式框图

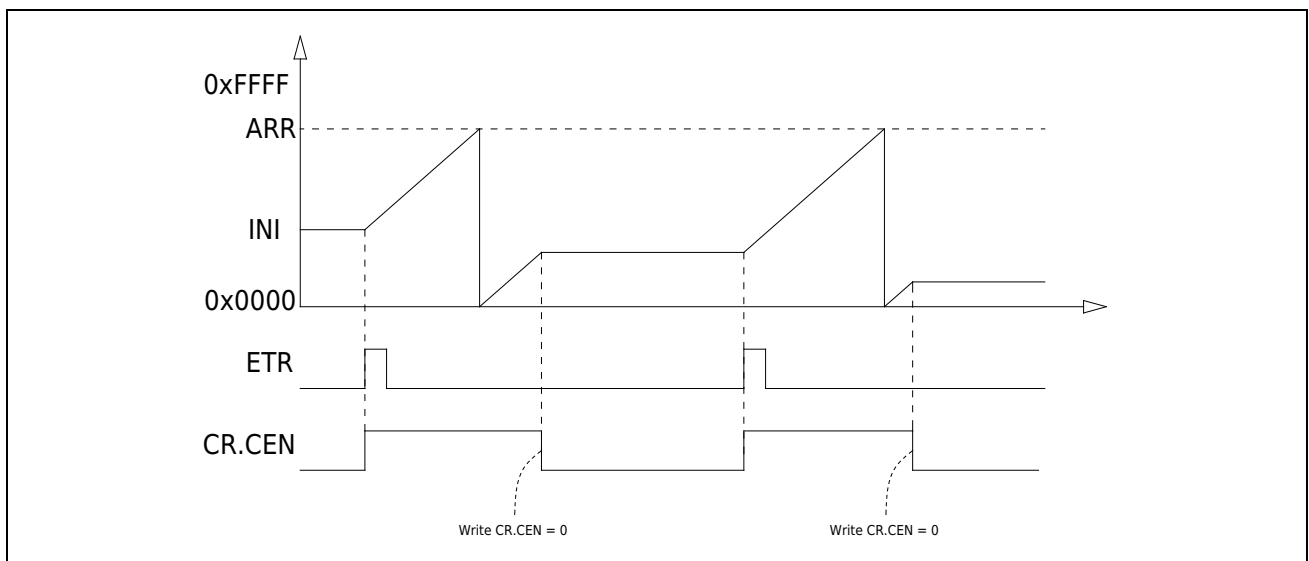


图 13-7 BTIM 触发启动模式计数示意图

13.2.7 门控模式

当设置 BTIMx_CR.MD 为 0x03 时，BTIMx 工作于门控模式，计数时钟为 PCLK 分频后的时钟 TCLKD。该模式下定时器需要满足两个条件才会进行计数：BTIMx_CR.CEN 为 1、ETR 管脚上的电平符合预期。定时器运行时，计数器从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx_CR.ETP 可以选择 ETR 管脚输入信号的高电平或低电平为计数电平。

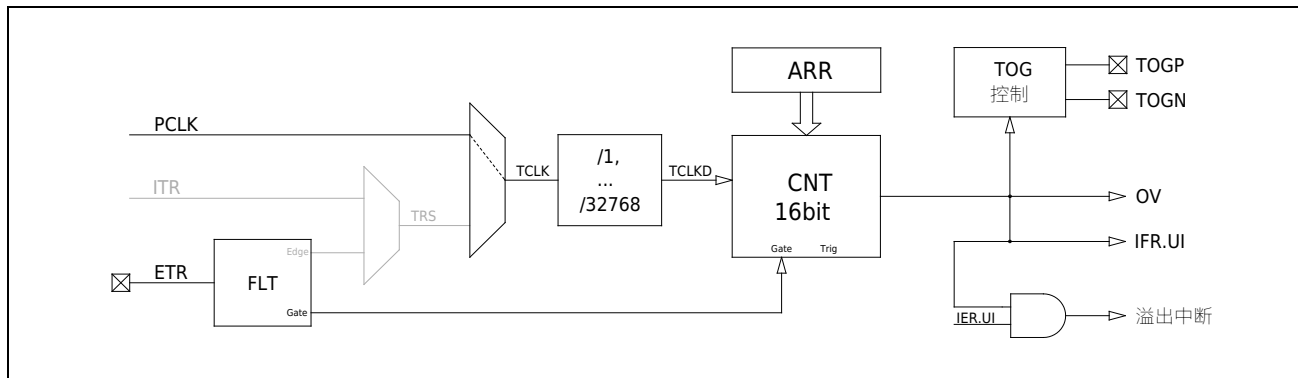


图 13-8 BTIM 门控模式框图

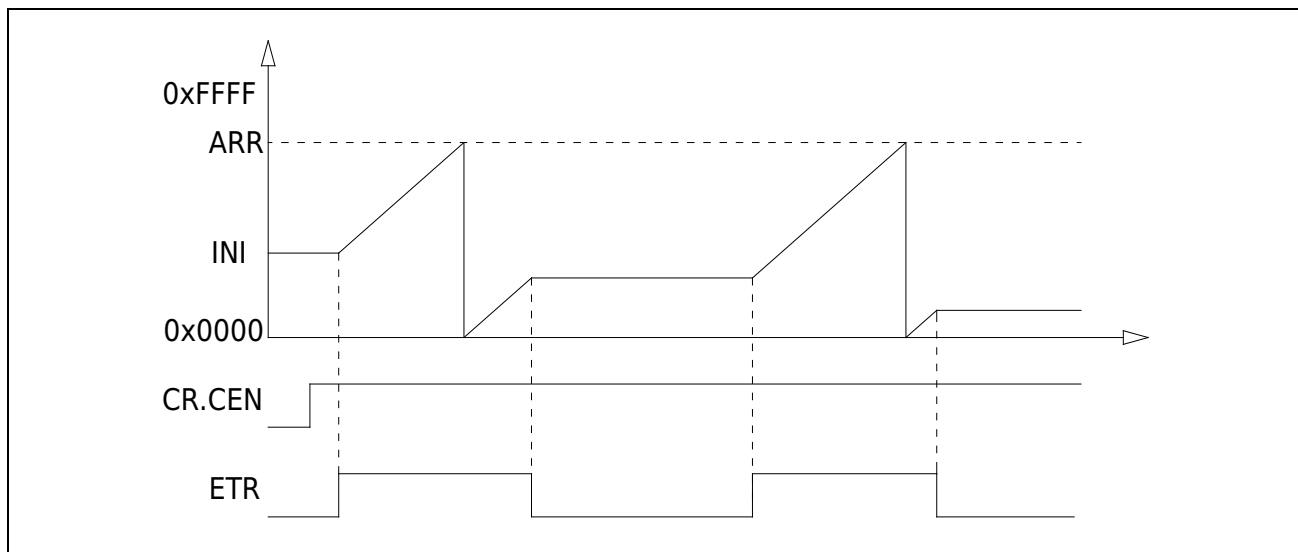


图 13-9 BTIM 门控模式计数示意图

13.2.8 定时器级联

通过适当的配置 BTIM 可以级联使用，如将两个 16 位的定时器组合成一个 32 位的定时器。级联使用时，第一级定时器的计数时钟可以是 PCLK 或来自 ETR 管脚的输入信号，后级定时器的计数时钟为前级定时器的溢出信号。信号级联已在芯片内固化，其级联方式如下所示。

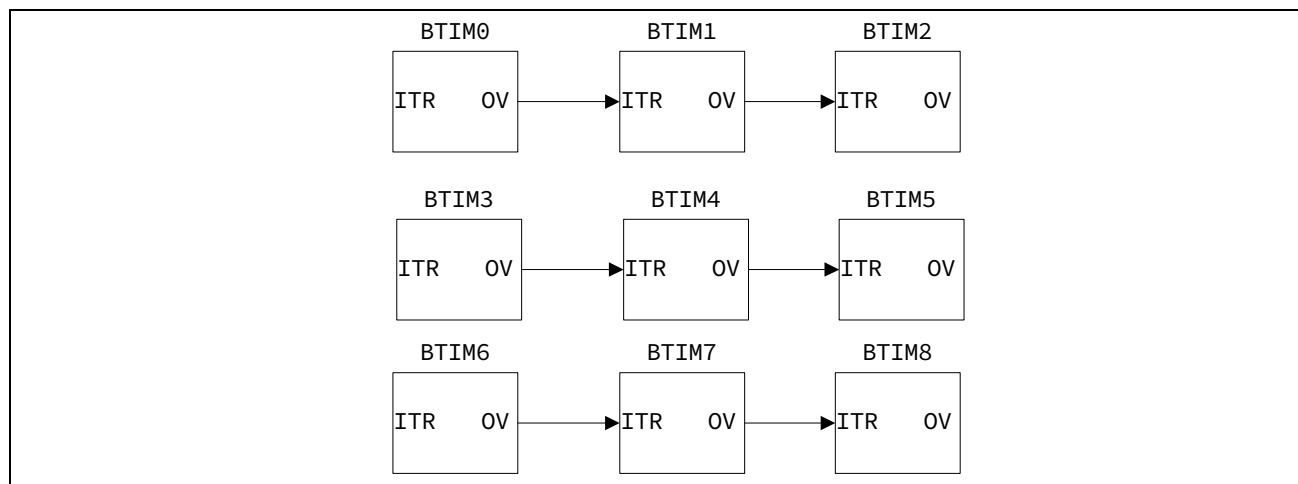


图 13-10 BTIM 定时器级联示意图

13.2.9 中断

表 13-2 BTIM 定时器中断

中断	说明
GTIM0_UEV_ETR_BTIMA	BTIM0 溢出/触发中断
GTIM0_CH01_BTIMB	BTIM1 溢出/触发中断
GTIM0_CH23_BTIMC	BTIM2 溢出/触发中断
GTIM1_UEV_ETR_BTIMA	BTIM3 溢出/触发中断
GTIM1_CH01_BTIMB	BTIM4 溢出/触发中断
GTIM1_CH23_BTIMC	BTIM5 溢出/触发中断
GTIM2_UEV_ETR_BTIMA	BTIM6 溢出/触发中断
GTIM2_CH01_BTIMB	BTIM7 溢出/触发中断
GTIM2_CH23_BTIMC	BTIM8 溢出/触发中断

13.3 GTIM 功能描述

13.3.1 功能框图

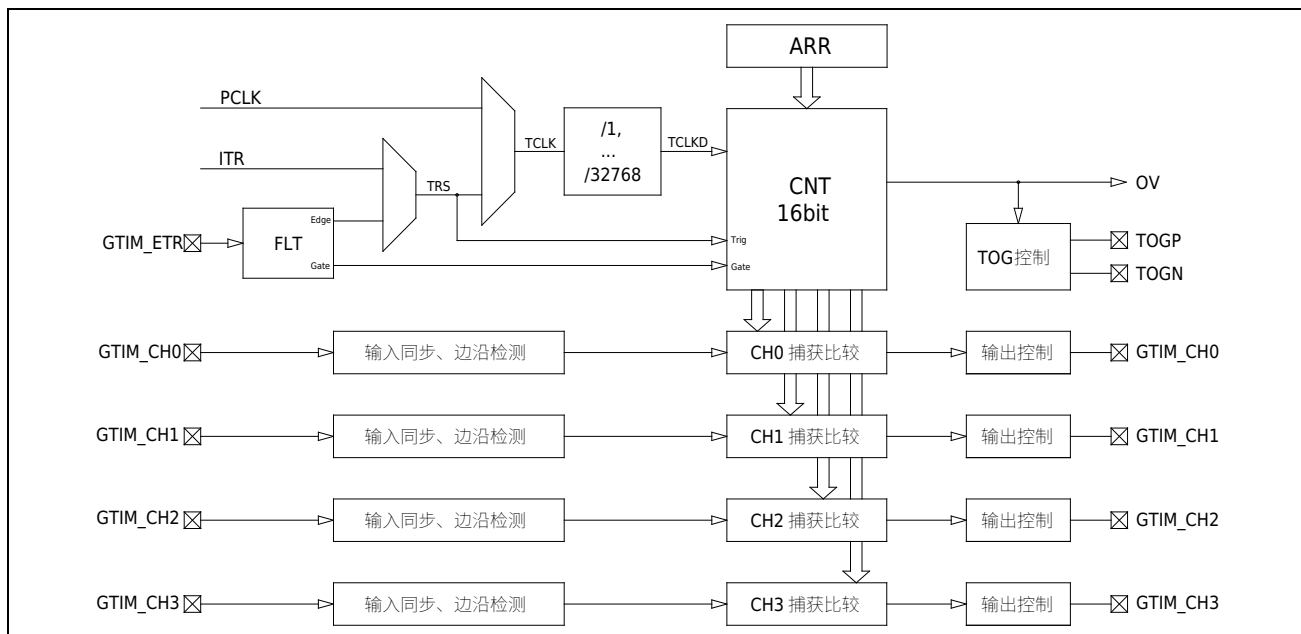


图 13-11 GTIM 框图

注：ITR 信号来源详见级联功能章节。

13.3.2 滤波单元

外部管脚输入的 ETR 信号经同步滤波、极性选择、边沿检测后，输出门控信号及边沿信号供计数单元及触发单元使用。由于 ETR 信号采用滤波时钟进行同步滤波，故输入的 ETR 信号的频率应低于滤波时钟的频率的二分之一。

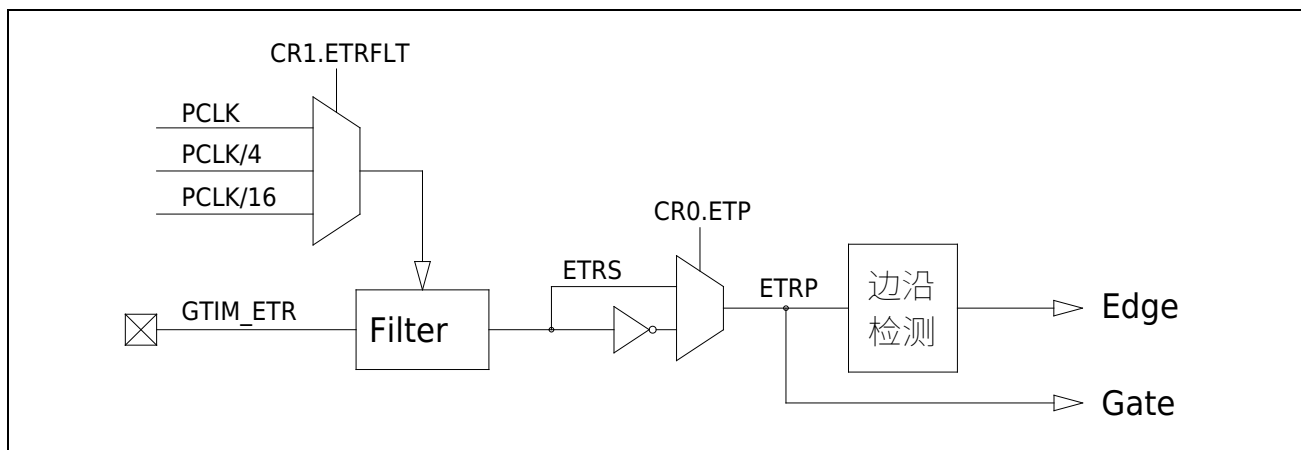


图 13-12 GTIM 滤波单元示意图

13.3.3 计数单元

在每个 TCLKD 信号的上升沿，内部计数器 CNT 自动加 1。计数到 ARR 后产生溢出信号 OV（宽度为一个 PCLK），并复位内部计数值为 0 并再次开始加计数。计数器的计数范围为 0 至 ARR，计数单元的计数周期为 ARR+1。

定时器支持 2 种计数模式，One-Shot 模式与连续计数模式。当设置 GTIM_CR0.OST 为 1 时，定时器工作于 One-Shot 模式；发生溢出后定时器立即停止计数，硬件自动清除 CEN。当设置 GTIM_CR0.OST 为 0 时，定时器工作于连续计数模式；只要 GTIM_CR0.CEN 为 1，定时器会一直进行计数。

每次溢出时，GTIM_IFR.UI 会被硬件置 1，若使能相应中断则可以产生溢出中断，用户应在中断服务程序中清除该溢出标志。

若 GTIM_CR0.TOGEN 为 1，则每溢出一次 TOGP/TOGN 输出电平翻转一次；用户可通过 GPIO 的复用功能将 TOGP/TOGN 信号输出到管脚用于驱动外部电路。

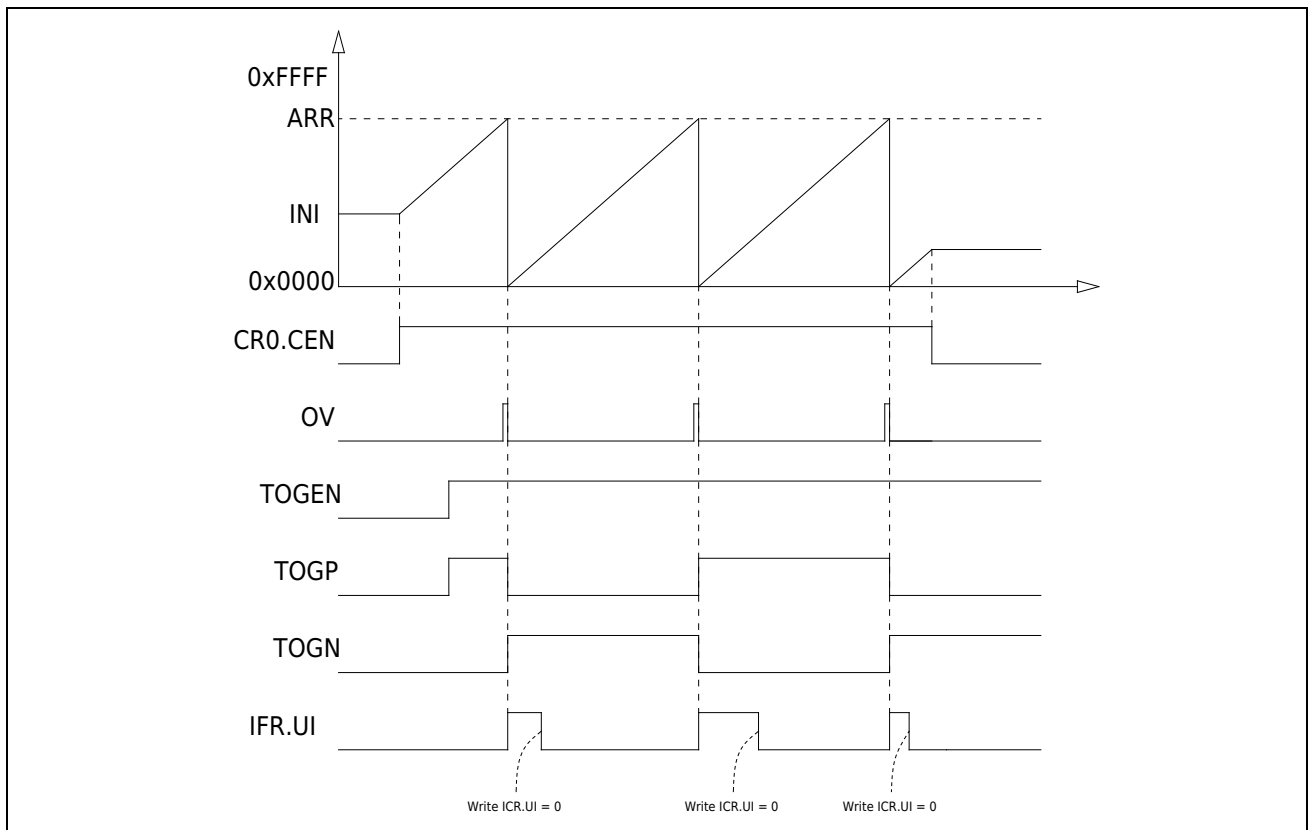


图 13-13 GTIM 计数器波形

13.3.4 定时器模式

当设置 GTIM_CR0.MD 为 0x00 时, GTIM 工作于定时器模式。在该模式下, 计数时钟为 PCLK 分频后的 TCLKD, 当设置 GTIM_CR0.CEN 为 1 后, 定时器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

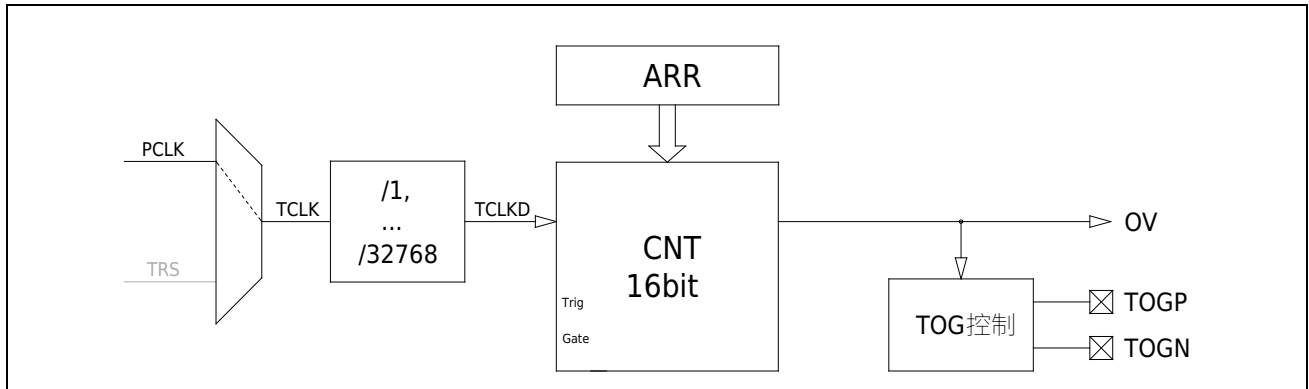


图 13-14 GTIM 定时模式框图

13.3.5 计数器模式

当设置 GTIM_CR0.MD 为 0x01 时, GTIM 工作于计数器模式。在该模式下, 计数时钟为分频后的 ITR 或 ETR 管脚输入的信号。当设置 GTIM_CR0.CEN 为 1 后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 GTIM_CR0.TRS 可以选择计数时钟源为 ITR 或 ETR; 通过 GTIM_CR0.ETP 可以选择对 ETR 管脚输入信号的上升沿或下降沿进行计数。

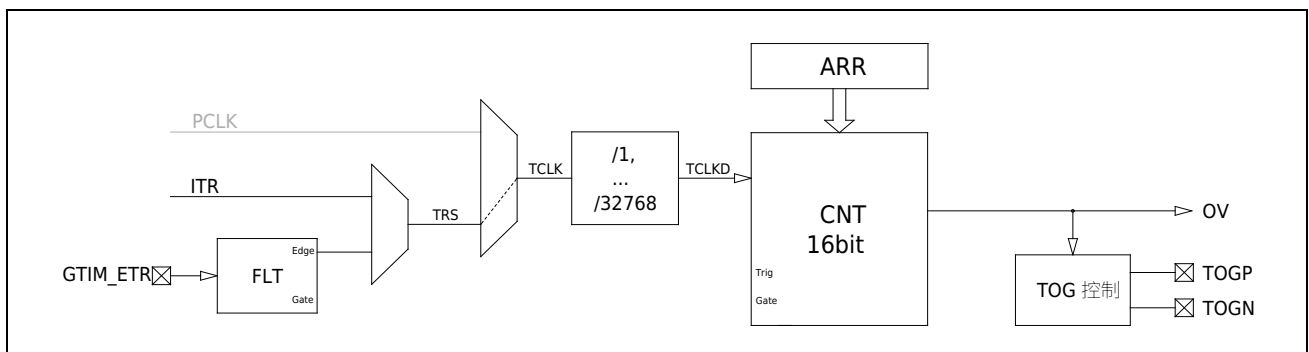


图 13-15 GTIM 计数器模式框图

13.3.6 触发启动模式

当设置 GTIM_CR0.MD 为 0x02 时，GTIM 工作于触发启动模式，计数时钟为 PCLK 分频后的 TCLKD。该模式下定时器具有两种启动方法：设置 GTIM_CR0.CEN 为 1 或 TRS 信号上符合预期的电平变化。

启动后，计数器即从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数；在任意时刻设置 GTIM_CR0.CEN 为 0，则可以停止定时器。

通过 GTIM_CR0.TRS 选择 TRS 信号来源为 ETR 或 ITR。

通过 GTIM_CR0.ETP 可以选择 ETR 管脚输入信号的上升沿或下降沿启动定时器。

每次检测到符合预期的触发信号时，GTIM_IFR.TI 会被硬件置 1，若使能相应中断则可以产生触发中断，用户应在中断服务程序中清除该触发标志。

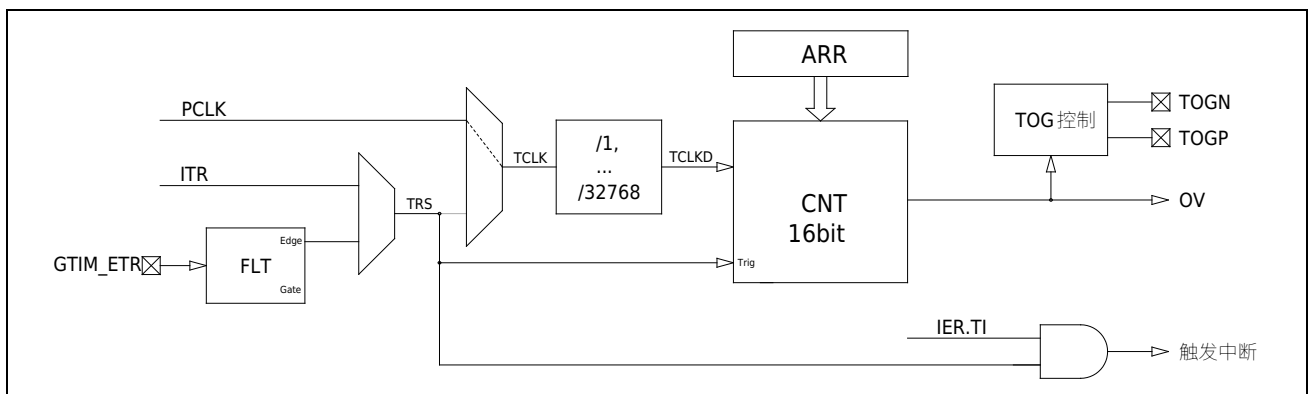


图 13-16 GTIM 触发启动器模式框图

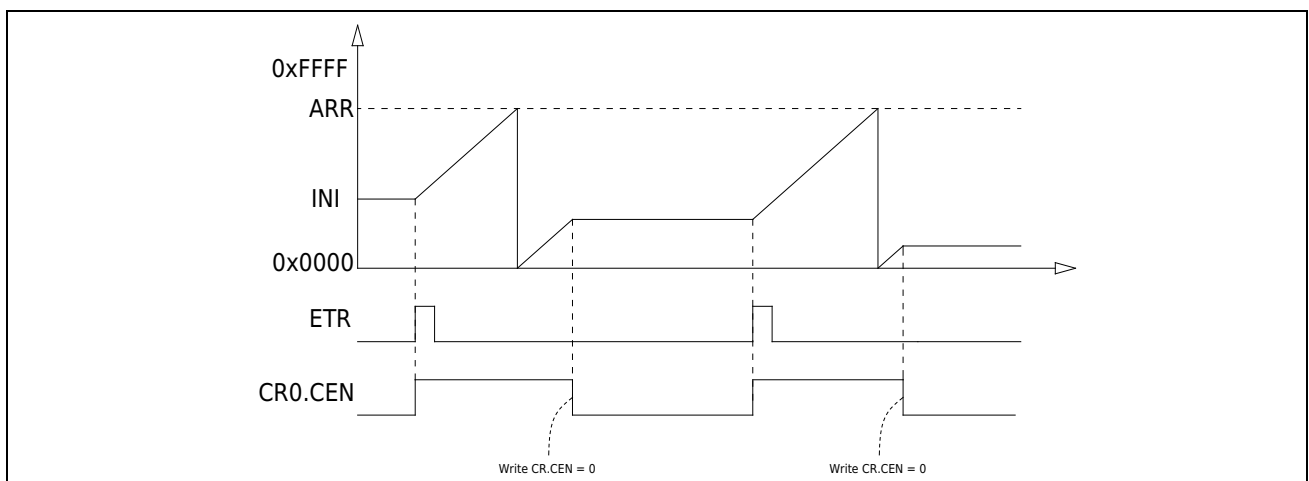


图 13-17 GTIM 触发启动模式计数示意图

13.3.7 门控模式

当设置 GTIM_CR0.MD 为 0x03 时, GTIM 工作于门控模式, 计数时钟为 PCLK 分频后的 TCLKD。该模式下定时器需要满足两个条件才会进行计数: GTIM_CR0.CEN 为 1、ETR 管脚上的电平符合预期。定时器运行时, 计数器从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 GTIM_CR0.ETP 可以选择 ETR 管脚输入信号的高电平或低电平为计数电平。

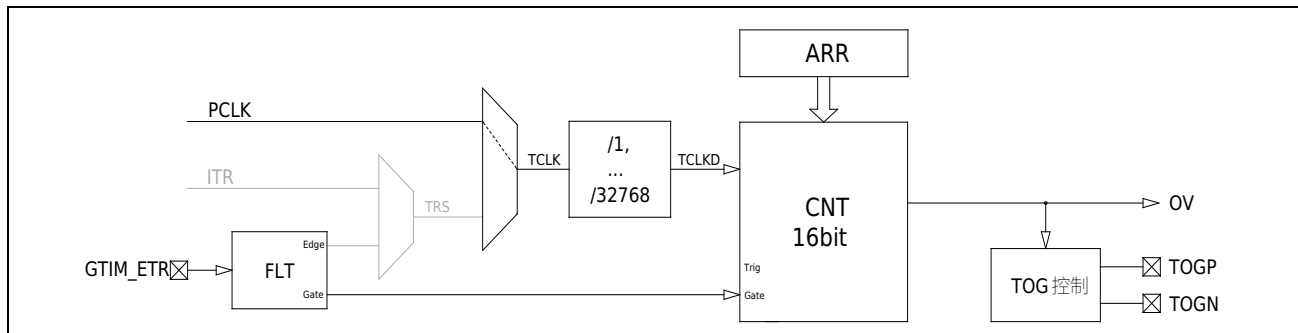


图 13-18 GTIM 门控模式框图

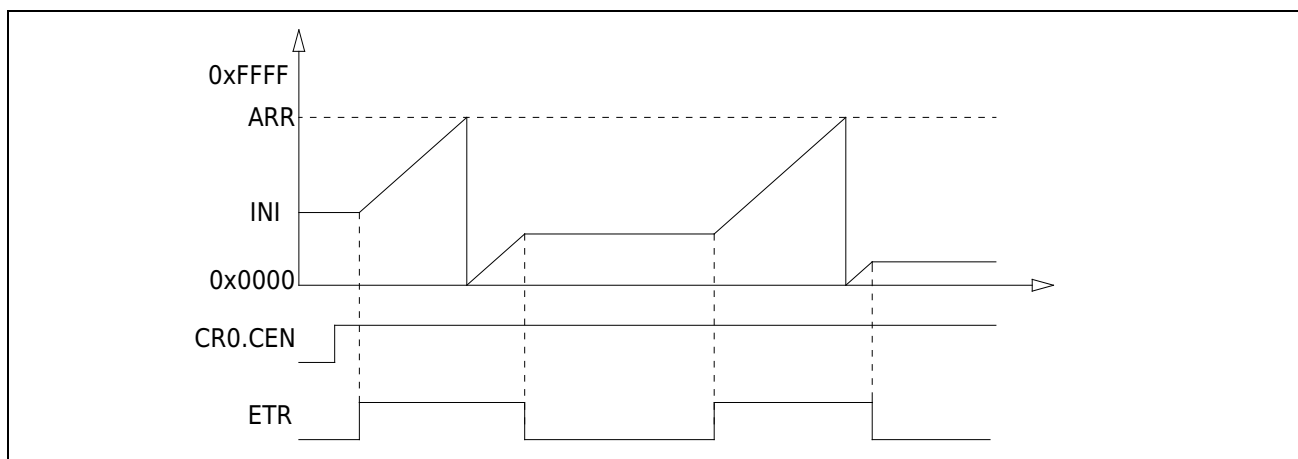


图 13-19 GTIM 门控模式计数示意图

13.3.8.2 比较功能

当配置 CMMR.CCyM 为 4 ~ 7 时，CHy 工作于比较模式，可以较出四种波形：

- 100：强制输出低电平
- 101：强制输出高电平
- 110：PWM 正向输出（CNT $\geq$ CCRy 输出高电平）
- 111：PWM 反向输出（CNT $<$ CCRy 输出高电平）

注：当 ARR 值不为 0xFFFF 时，可以实现 PWM 调整范围为 0% ~ 100%。

例：在正向输出模式 ARR 为 99 时，CCR 为 0~100 对应的占空比为 100% ~ 0%。

下图给出了两个通道进行比较输出时的示意。其中 CH0 采用 PWM 正向输出；CH1 采用 PWM 反向输出。

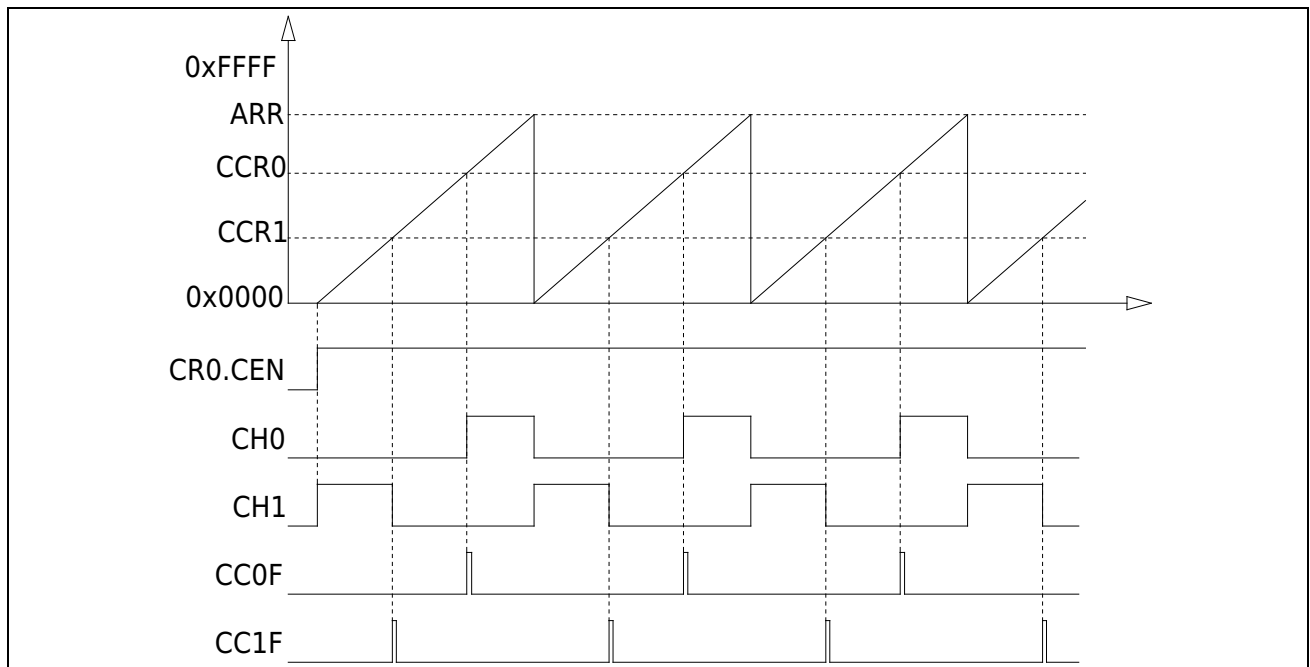


图 13-21 GTIM 比较功能示意图

13.3.9 DMA 触发

比较捕获可以产生触发信号触发 DMA。使能使用 GTIM_IER.DCCx(x=0~3)控制。

13.3.10 中断

表 13-3 GTIM 中断源列表

中断	说明
GTIM0_UEV_ETR_BTIMA	GTIM0 溢出/触发中断
GTIM0_CH01_BTIMB	GTIM0 通道 01 比较捕获中断
GTIM0_CH23_BTIMC	GTIM0 通道 23 比较捕获中断
GTIM1_UEV_ETR_BTIMA	GTIM1 溢出/触发中断
GTIM1_CH01_BTIMB	GTIM1 通道 01 比较捕获中断
GTIM1_CH23_BTIMC	GTIM1 通道 23 比较捕获中断
GTIM2_UEV_ETR_BTIMA	GTIM2 溢出/触发中断
GTIM2_CH01_BTIMB	GTIM2 通道 01 比较捕获中断
GTIM2_CH23_BTIMC	GTIM2 通道 23 比较捕获中断

13.3.11 片内外设互联

触发互连

表 13-4 GTIM 触发互联列表

CR0.TRS	00	01	10	11
GTIM0	ETR	ATIM0_TRGO	ADC0_ExtTrigger0.TRIG.	ADC0_ExtTrigger1.TRIG.
GTIM1	ETR	ATIM3_TRGO	ADC1_ExtTrigger0.TRIG.	ADC1_ExtTrigger1.TRIG.
GTIM2	ETR	ATIM0_TRGO	GTIM1 溢出.	ATIM3_TRGO.

当 SYSCTRL.IOAUX1.ETRx 为 0 时, GTIM_ETR 信号来源由 GPIO_AFRH/ GPIO_AFRL 决定; 当 SYSCTRL.IOAUX1.ETRx 为 1~7 时, GTIM_ETR 信号的来源由下表决定。

表 13-5 GTIM ETR 源列表

SYSCTRL.IOAUX1	0	1	2	3	4	5	6	7
CTIM0ETR[6:4]	GTIM0_ETR	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA05	PA13	PB05
CTIM1ETR[10:8]	GTIM1_ETR	USART2_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA06	PA14	PB06
CTIM2ETR[14:12]	GTIM2_ETR	USART3_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA07	PA15	PB07

当 SYSCTRL.IOAUX2.CAPx 为 0 时, GTIM_CH3 的捕获信号来源由 GPIO_AFRH/ GPIO_AFRL 决定; 当 SYSCTRL.IOAUX2.CAPx 为 1~7 时, GTIM_CH3 信号的来源由下表决定。

表 13-6 GTIM CH3 信号来源列表

SYSCTRL.IOAUX2	0	1	2	3	4	5	6	7
CTIM0CAP[6:4]	GTIM0_CH3	USART1_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA01	PA09	PB01
CTIM1CAP[10:8]	GTIM1_CH3	USART2_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA02	PA10	PB02
CTIM2CAP[14:12]	GTIM2_CH3	USART3_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA03	PA11	PB03

13.4 BTIM 寄存器描述

13.4.1 寄存器列表

表 13-7 BTIM 基地址列表

名称	基地址	描述
BTIM0	0x4000 5000	BTIM0 基地址
BTIM1	0x4000 5100	BTIM1 基地址
BTIM2	0x4000 5200	BTIM2 基地址
BTIM3	0x4000 5400	BTIM3 基地址
BTIM4	0x4000 5500	BTIM4 基地址
BTIM5	0x4000 5600	BTIM5 基地址
BTIM6	0x4000 5800	BTIM6 基地址
BTIM7	0x4000 5900	BTIM7 基地址
BTIM8	0x4000 5A00	BTIM8 基地址

注意：BTIM 在 SysCtrl_CR1.GTIMxCFG 为 1 时才有效。

表 13-8 BTIM 寄存器列表

(x=0~8)

寄存器名	偏移地址	复位值
重载寄存器 (BTIMx_ARR)	0x00	0x0000 FFFF
计数器寄存器 (BTIMx_CNT)	0x04	0x0000 0000
控制寄存器 (BTIMx_CR)	0x10	0x0000 0000
中断使能寄存器 (BTIMx_IER)	0x14	0x0000 0000
中断标志寄存器 (BTIMx_IFR)	0x18	0x0000 0000
中断清除寄存器 (BTIMx_ICR)	0x1C	0x0000 0003
复合中断标志寄存器 (BTIM_AIFR)	0x48	0x0000 0000
复合中断清除寄存器 (BTIM_AICR)	0x4C	0x0000 003F

13.4.2 重载寄存器 (BTIMx_ARR) (x=0~8)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	ARR	定时器重载值	-	RW

13.4.3 计数寄存器 (BTIMx_CNT) (x=0~8)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CNT	定时器计数值	-	RW

13.4.4 控制寄存器 (BTIMx_CR) (x=0~8)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ETP	TRS			OST	PRS			TOGEN	MD		CEN

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11	ETP	外部管脚输入的 ETR 信号极性选择	0: ETR 正向(触发模式上升沿有效, 门控模式高电平有效) 1: ETR 反向(触发模式下降沿有效, 门控模式低电平有效)	RW
b10~b9	TRS	触发源选择	00: ETR 管脚输入的信号 01: ITR, 详见定时器级联功能章节 注意: BTIM0~2 共用一个 ETR 管脚; BTIM3~5 共用另一个 ETR 管脚, BTIM6~8 共用另一个 ETR 管脚	RW
b8	OST	单次/连续计数模式控制	0: 连续计数模式 1: 单次计数模式	RW
b7~b4	PRS	预分频器分频系数配置	0: DIV1 4: DIV16 8: DIV256 12: DIV4096 1: DIV2 5: DIV32 9: DIV512 13: DIV8192 2: DIV4 6: DIV64 10: DIV1024 14: DIV16384 3: DIV8 7: DIV128 11: DIV2048 15: DIV32768	RW
b3	TOGEN	TOG 管脚输出使能控制	0: TOGP、TOGN 输出电平均为 0 1: TOGP、TOGN 输出电平相反的信号	RW
b2~b1	MD	定时器工作模式配置	00: 定时器模式, 计数时钟为 PCLK 01: 计数器模式, 计数时钟为 TRS 信号 10: 触发启动模式, 计数时钟为 PCLK, TRS 信号触发计数器启动 11: 门控模式, 计数时钟为 PCLK, ETR 管脚输入信号作为门控	RW
b0	CEN	定时器使能控制	0: 定时器停止 1: 定时器使能 注意: 触发启动模式触发后, CEN 由硬件置 1	RW

13.4.5 中断使能 (BTIMx_IER) (x=0~8)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														TI	UI

位	标记	位名	功能	读写
b31~b2	Res	保留位	保留, 必须保持复位值	
b1	TI	触发中断使能控制	0: 禁止触发中断 1: 使能触发中断	RW
b0	UI	溢出中断使能控制	0: 禁止溢出中断 1: 使能溢出中断	RW

13.4.6 中断标志寄存器 (BTIMx_IFR) (x=0~8)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														TI	UI

位	标记	位名	功能	读写
b31~b2	Res	保留位	保留, 必须保持复位值	
b1	TI	触发标志	0: 未发生触发事件 1: 已发生触发事件	RO
b0	UI	计数器溢出标志	0: 计数器未溢出 1: 计数器已溢出	RO

13.4.7 中断标志清除寄存器 (BTIMx_ICR) (x=0~8)

复位值: 0x0000 0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														TI	UI

位	标记	位名	功能	读写
b31~b2	Res	保留位	保留, 必须保持复位值	
1	TI	触发标志清除	0: 清除触发标志 1: 无功能	R1W0
0	UI	计数器溢出标志清除	0: 清除计数器溢出标志 1: 无功能	R1W0

13.4.8 复合中断标志寄存器 (BTIM_AIFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										TI2	UI2	TI1	UI1	TI0	UI0

位	标记	位名	功能	读写
b31~b2	Res	保留位	保留, 必须保持复位值	
b5	TI2	BTIM2/5/8 触发标志	0: 未发生触发事件 1: 已发生触发事件	R0
b4	UI2	BTIM2/5/8 溢出标志	0: 计数器未溢出 1: 计数器已溢出	R0
b3	TI1	BTIM1/4/7 触发标志	0: 未发生触发事件 1: 已发生触发事件	R0
b2	UI1	BTIM1/4/7 溢出标志	0: 计数器未溢出 1: 计数器已溢出	R0
b1	TI0	BTIM0/3/6 触发标志	0: 未发生触发事件 1: 已发生触发事件	R0
b0	UI0	BTIM0/3/6 溢出标志	0: 计数器未溢出 1: 计数器已溢出	R0

13.4.9 复合中断标志清除寄存器 (BTIM_AICR)

复位值: 0x0000 003F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										TI2	UI2	TI1	UI1	TI0	UI0

位	标记	位名	功能	读写
b31~b2	Res	保留位	保留, 必须保持复位值	
b5	TI2	BTIM2/5/8 触发标志清除	0: 清除触发标志 1: 无功能	R1W0
b4	UI2	BTIM2/5/8 溢出标志清除	0: 清除溢出标志 1: 无功能	R1W0
b3	TI1	BTIM1/4/7 触发标志清除	0: 清除触发标志 1: 无功能	R1W0
b2	UI1	BTIM1/4/7 溢出标志清除	0: 清除溢出标志 1: 无功能	R1W0
b1	TI0	BTIM0/3/6 触发标志清除	0: 清除触发标志 1: 无功能	R1W0
b0	UI0	BTIM0/3/6 溢出标志清除	0: 清除溢出标志 1: 无功能	R1W0

13.5 GTIM 寄存器描述

13.5.1 寄存器列表

表 13-9 GTIM 基地址列表

Timer	基地址	描述
GTIM0	0x4000 5000	GTIM0基地址
GTIM1	0x4000 5400	GTIM1基地址
GTIM2	0x4000 5800	GTIM2基地址

注：GTIM 在 SysCtrl_CR1.GTIMxCFG 为 0 时有效，为 1 时无效。

表 13-10 GTIM 寄存器列表

寄存器名	偏移地址	复位值
重载寄存器 (GTIM_ARR)	0x300	0x0000 FFFF
计数寄存器 (GTIM_CNT)	0x304	0x0000 0000
比较捕获控制寄存器 (GTIM_CMMR)	0x308	0x0000 0000
控制寄存器1 (GTIM_CR1)	0x30C	0x0000 0000
控制寄存器0 (GTIM_CR0)	0x310	0x0000 0000
中断使能控制寄存器 (GTIM_IER)	0x314	0x0000 0000
中断标志寄存器 (GTIM_IFR)	0x318	0x0000 0000
中断标志清除寄存器 (GTIMx_ICR)	0x31C	0x0000 003F
CH0比较/捕获寄存器 (GTIM_CCR0)	0x320	0x0000 FFFF
CH1比较/捕获寄存器 (GTIM_CCR1)	0x324	0x0000 0000
CH2比较/捕获寄存器 (GTIM_CCR2)	0x328	0x0000 FFFF
CH3比较/捕获寄存器 (GTIM_CCR3)	0x32C	0x0000 0000

13.5.2 重载寄存器 (GTIM_ARR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARR															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	ARR	定时器重载值											RW		

13.5.3 计数寄存器 (GTIM_CNT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	CNT	定时器计数值	-										RW		

13.5.4 控制寄存器 1 (GTIM_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ETRFLT		Res					

位	标记	位名	功能	读写
b31~b7	Res	保留位	保留, 必须保持复位值	
采样时钟为PCLK或PCLK分频, 采样点数为N				
000: 无滤波				
001: $F_{\text{sample}} = \text{PCLK}$, $N=2$				
010: $F_{\text{sample}} = \text{PCLK}$, $N=4$				
b6~b4	ETRFLT	ETR 管脚输入信号滤波配置	011: $F_{\text{sample}} = \text{PCLK}$, $N=6$ 100: $F_{\text{sample}} = \text{PCLK}/4$, $N=4$ 101: $F_{\text{sample}} = \text{PCLK}/4$, $N=6$ 110: $F_{\text{sample}} = \text{PCLK}/8$, $N=4$ 111: $F_{\text{sample}} = \text{PCLK}/8$, $N=6$	RW
b3~b0	Res	保留位	保留, 必须保持复位值	

13.5.5 控制寄存器 0 (GTIM_CR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ETP	TRS		OST	PRS				TOGEN	MD		CEN

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11	ETP	外部输入ETR极性选择	0: 外部输入ETR正向(触发模式上升沿有效, 门控模式高电平有效) 1: 外部输入 ETR 反向(触发模式下降沿有效, 门控模式低电平有效)	RW
b10~b9	TRS	触发源选择	00: ETR管脚输入的信号 01: 内部互联1 10: 内部互联2 11: 内部互联3 Note: 内部互联详见片内外设互联功能描述	RW
b8	OST	单次/连续计数模式控制	0: 连续计数模式 1: 单次计数模式	RW
b7~b4	PRS	预分频器分频系数配置	0: DIV1 4: DIV16 8: DIV256 12: DIV4096 1: DIV2 5: DIV32 9: DIV512 13: DIV8192 2: DIV4 6: DIV64 10: DIV1024 14: DIV16384 3: DIV8 7: DIV128 11: DIV2048 15: DIV32768 Note: 捕获模式预分频系数只能设置 0,1,2,3	RW
b3	TOGEN	TOG 管脚输出使能控制	0: TOGP、TOGN 输出电平均为 0 1: TOGP、TOGN 输出电平相反的信号	RW
b2~b1	MD	定时器工作模式配置	00: 定时器模式, 计数时钟为 PCLK 01: 计数器模式, 计数时钟为 TRS 信号 10: 触发启动模式, 计数时钟为 PCLK, TRS 信号触发计数器启动 11: 门控模式, 计数时钟为 PCLK, ETR 管脚输入信号作为门控	RW
b0	CEN	定时器使能控制	0: 定时器停止 1: 定时器使能 注意: 触发启动模式触发后, CEN 由硬件置 1	RW

13.5.6 中断使能控制寄存器 (GTIM_IER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				C0TRAD	OVTRAD	DCC3	DCC2	DCC1	DCC0	CC3	CC2	CC1	CC0	TI	UI

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11	C0TRAD	比较捕获0触发ADC使能	0: 禁止 1: 使能	RW
b10	OVTRAD	溢出触发ADC使能控制	0: 禁止 1: 使能	RW
b9	DCC3	CH3触发DMA使能控制	0: 禁止 1: 使能	RW
b8	DCC2	CH2触发DMA使能控制	0: 禁止 1: 使能	RW
b7	DCC1	CH1触发DMA使能控制	0: 禁止 1: 使能	RW
b6	DCC0	CH0触发DMA使能控制	0: 禁止 1: 使能	RW
b5	CC3	CH3捕获比较中断使能控制	0: 禁止 1: 使能	RW
b4	CC2	CH2捕获比较中断使能控制	0: 禁止 1: 使能	RW
b3	CC1	CH1捕获比较中断使能控制	0: 禁止 1: 使能	RW
b2	CC0	CH0捕获比较中断使能控制	0: 禁止 1: 使能	RW
b1	TI	触发中断使能控制	0: 禁止 1: 使能	RW
b0	UI	溢出中断使能控制	0: 禁止 1: 使能	RW

13.5.7 中断标志寄存器 (GTIM_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										CC3	CC2	CC1	CC0	TI	UI

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
b5	CC3	CH3比较捕获中断标志	0: 未发生比较捕获事件 1: 已发生比较捕获事件	RW
b4	CC2	CH2比较捕获中断标志	0: 未发生比较捕获事件 1: 已发生比较捕获事件	RW
b3	CC1	CH1比较捕获中断标志	0: 未发生比较捕获事件 1: 已发生比较捕获事件	RW
b2	CC0	CH0比较捕获中断标志	0: 未发生比较捕获事件 1: 已发生比较捕获事件	RW
b1	TI	触发中断标志	0: 未发生触发事件 1: 已发生触发事件	RW
b0	UI	溢出中断标志	0: 计数器未溢出 1: 计数器已溢出	RW

13.5.8 中断标志清除寄存器 (GTIM_ICR)

复位值: 0x0000 003F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										CC3	CC2	CC1	CC0	TI	UI

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
b5	CC3	CH3 比较捕获中断标志清除	0: 清除比较捕获中断标志 1: 无功能	R1W0
b4	CC2	CH2 比较捕获中断标志清除	0: 清除比较捕获中断标志 1: 无功能	R1W0
b3	CC1	CH1 比较捕获中断标志清除	0: 清除比较捕获中断标志 1: 无功能	R1W0
b2	CC0	CH0 比较捕获中断标志清除	0: 清除比较捕获中断标志 1: 无功能	R1W0
b1	TI	触发中断标志清除	0: 清除比较捕获中断标志 1: 无功能	R1W0
b0	UI	溢出中断标志清除	0: 清除计数器溢出标志 1: 无功能	R1W0

13.5.9 比较捕获控制寄存器 (GTIM_CMMR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	CC3M			Res	CC2M			Res.	CC1M			Res	CC0M		

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14~b12	CC3M	CH3捕获比较模式配置	功能描述详见 CC0M	RW
b11	Res	保留位	保留, 必须保持复位值	
b10~b8	CC2M	CH2捕获比较模式配置	功能描述详见 CC0M	RW
b7	Res	保留位	保留, 必须保持复位值	
b6~b4	CC1M	CH1捕获比较模式配置	功能描述详见 CC0M	RW
b3	Res	保留位	保留, 必须保持复位值	
b2~b0	CC0M	CH0捕获比较模式配置	000: 无功能	RW
			001: 上升沿捕获	
			010: 下降沿捕获	
			011: 上升下降沿同时捕获	
			100: 强制输出低电平	
			101: 强制输出高电平	
			110: PWM正向输出 (CNT ≥ CCR0输出高电平)	
			111: PWM反向输出 (CNT < CCR0输出高电平)	

13.5.10 比较捕获寄存器 (GTIM_CCRy) (y=0,1,2,3)

复位值: 0x0000 FFFF/ 0x0000 0000/ 0x0000 FFFF/ 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CCR	16 位比较捕获寄存器		RW

14 高级定时器 (ATIM)

14.1 概述

ATIM3 是由 1 个计数单元和 7 个比较单元组成的定时器，支持 6 个独立 PWM 输出或 3 对互补 PWM 输出。支持 6 个捕获输入。

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。

ATIM0 是由 1 个计数单元和 2 个比较单元组成的定时器，支持 2 个独立 PWM 输出或 1 对互补 PWM 输出。支持 2 个捕获输入。

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。

14.2 主要特性

14.2.1 ATIM3 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)
- 最多 6 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部输入计数功能

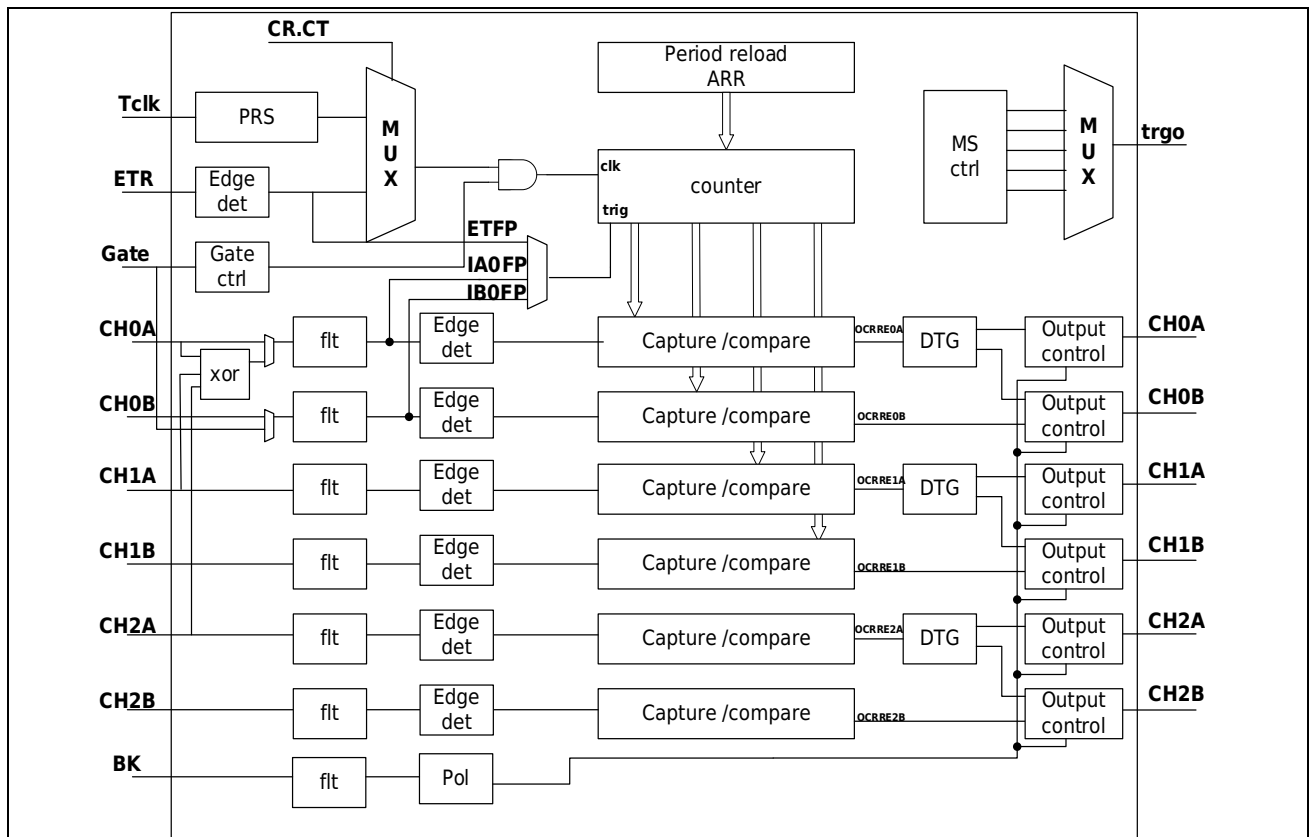


图 14-1 ATIM3 框图

14.2.2 ATIM0 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
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能

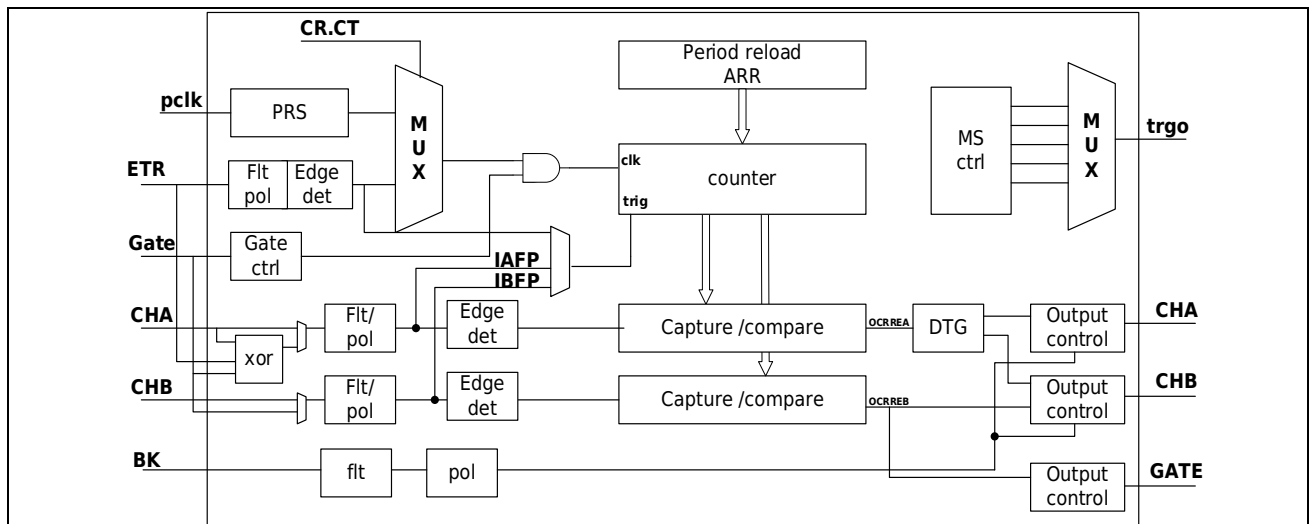


图 14-2 ATIM0 框图

14.3 功能描述

14.3.1 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。

14.3.2 定时计数器

高级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。

14.3.3 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：

表 14-1 ATIM PRS 分频比列表

PRS	0b000	0b 001	0b 010	0b 011	0b 100	0b 101	0b 110	0b 111
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。

14.3.4 模式 0 计数定时器功能

这种模式下，计数器向上计数。计数器支持两种计数模式，自由计数模式与重载模式，可以选择外部时钟 ETR 计数或系统时钟计数。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB，这种模式下 CHA,CHB 反相。

自由计数模式从设置计数值计数到 0xFFFF FFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

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。

14.3.4.1 功能框图

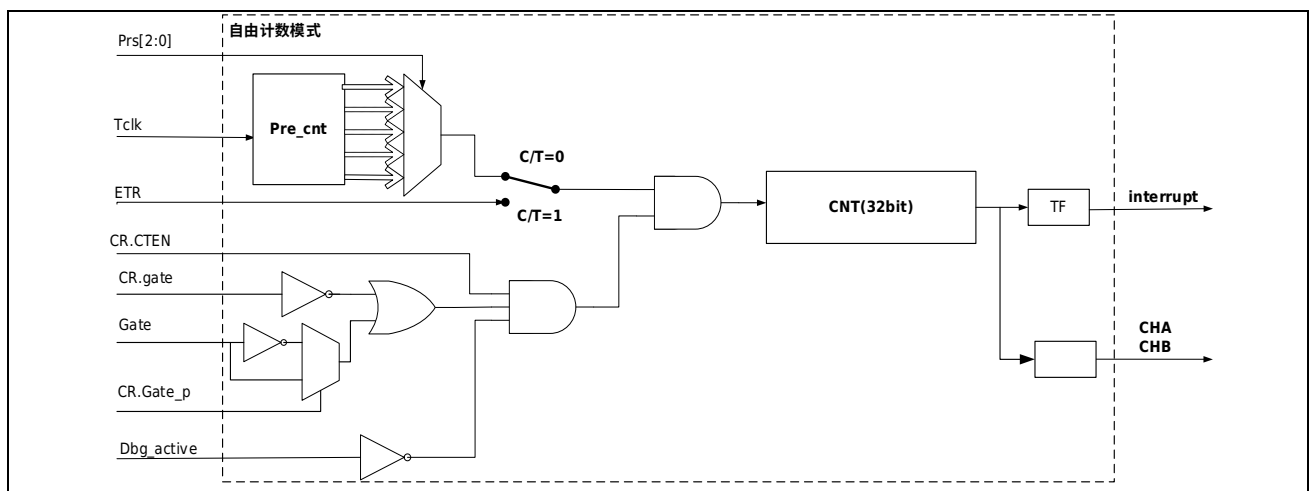


图 14-3 自由计数框图

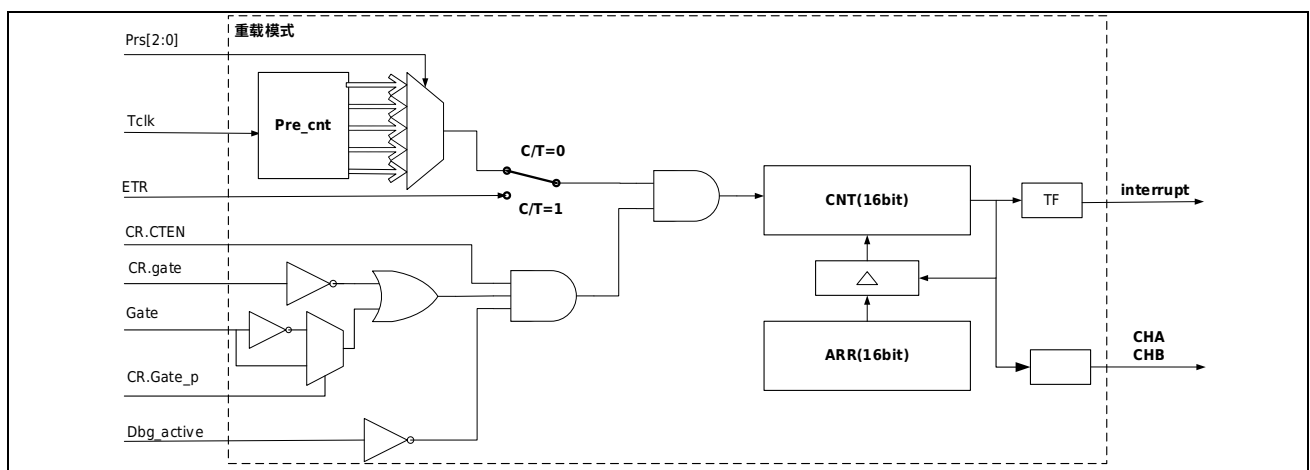


图 14-4 重载计数框图

14.3.4.2 计数波形

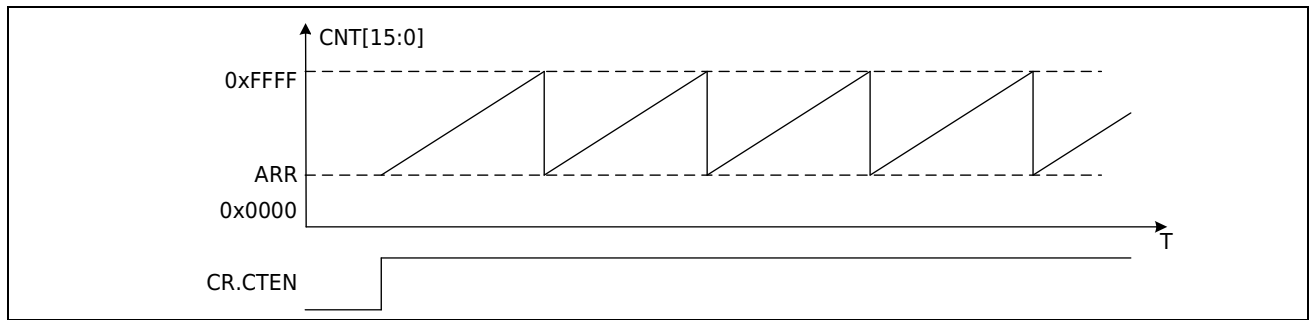


图 14-5 16 位重载计数波形

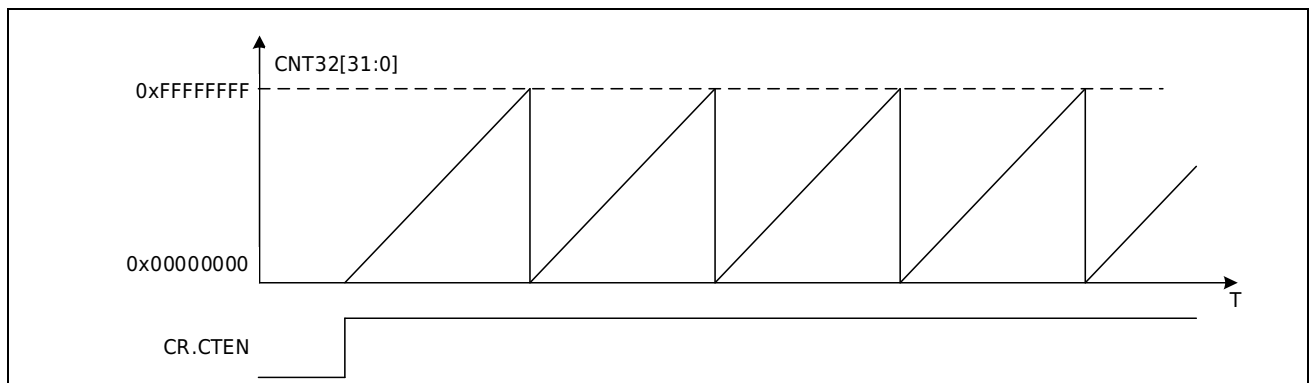


图 14-6 32 位自由计数波形

14.3.4.3 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 TCLK 采样，因此外部输入时钟频率不能超过系统的 TCLK 时钟。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。

14.3.4.4 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。

14.3.4.5 时序图

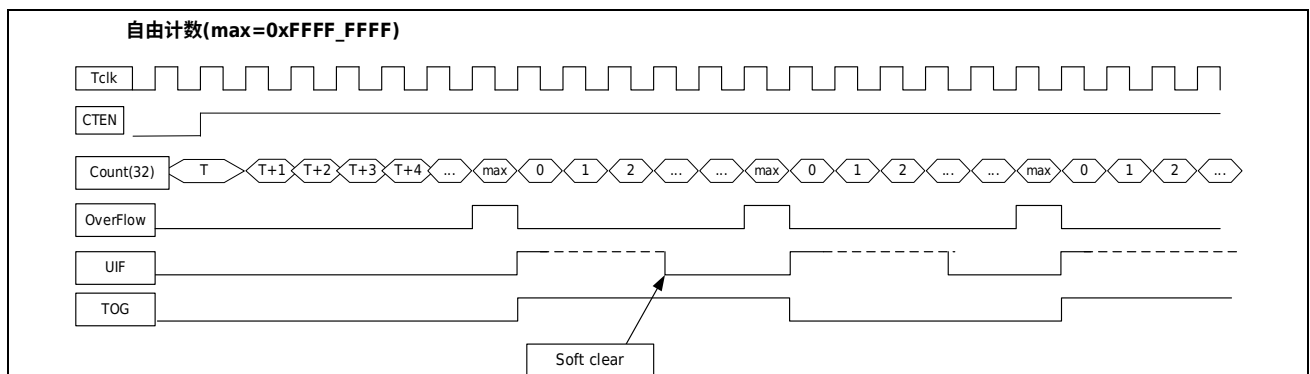


图 14-7 自由计数时序图

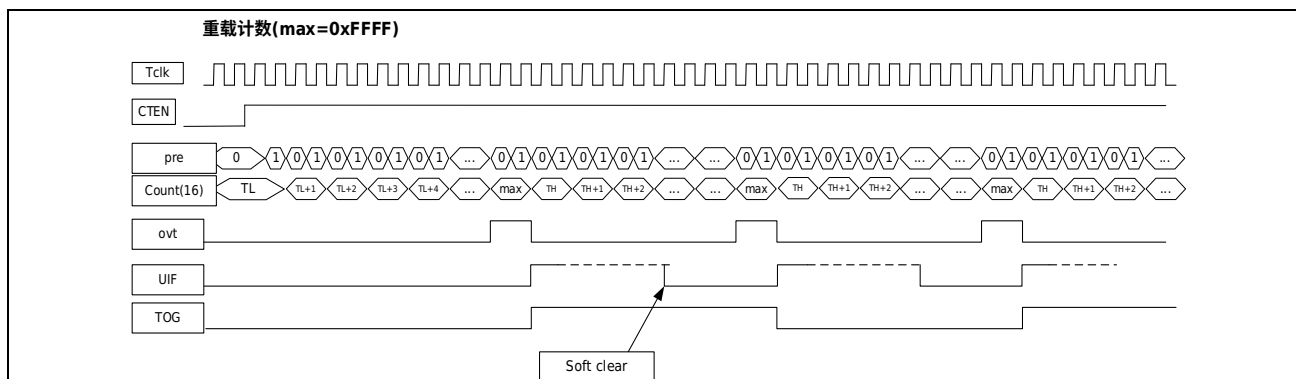


图 14-8 重载计数时序图（预分频设置为 2）

14.3.4.6 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 CR.TOGEN 为 0 可以同时设置端口 CHA,CHB 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 timer 重载模式配置如下：

表 14-2 4MHz 时钟下 Buzzer 输出频率与重载模式配置值对应列表

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

14.3.4.7 设置示例

重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出

14.3.5 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，并且产生脉冲宽度测量中断标志 PWC，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。

表 14-3 PWC 模式测量起止边沿配置表

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。

14.3.5.1 PWC 功能框图

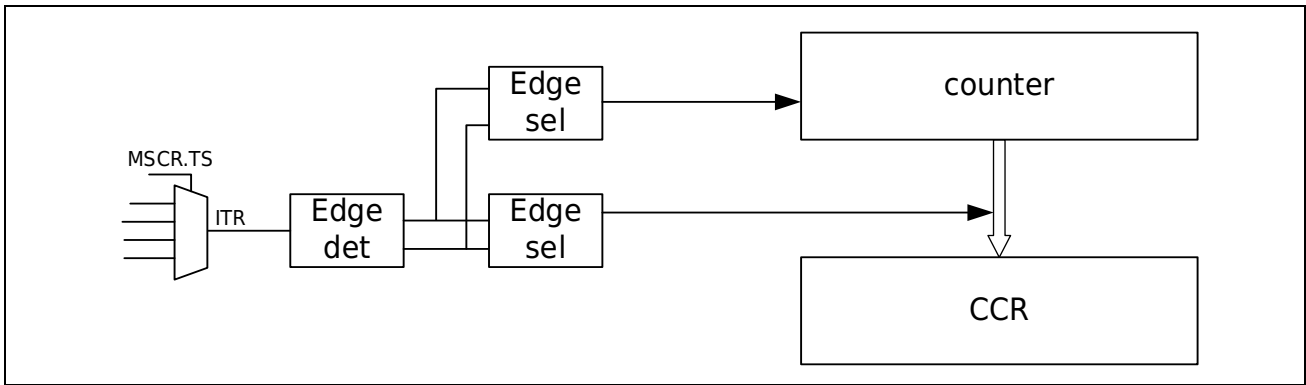


图 14-9 PWC 测量框图

表 14-4 PWC 模式测量信号源选择

MSCR.TS	触发选择
	000：端口ETR的滤波相位选择后的信号ETFP ； 001：内部互联信号 ITR0 010：内部互联信号 ITR1; 011：内部互联信号 ITR2; 100：内部互联信号 ITR3; 101：无效 110：端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效） 111：端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）

14.3.5.2 PWC 波形测量时序图

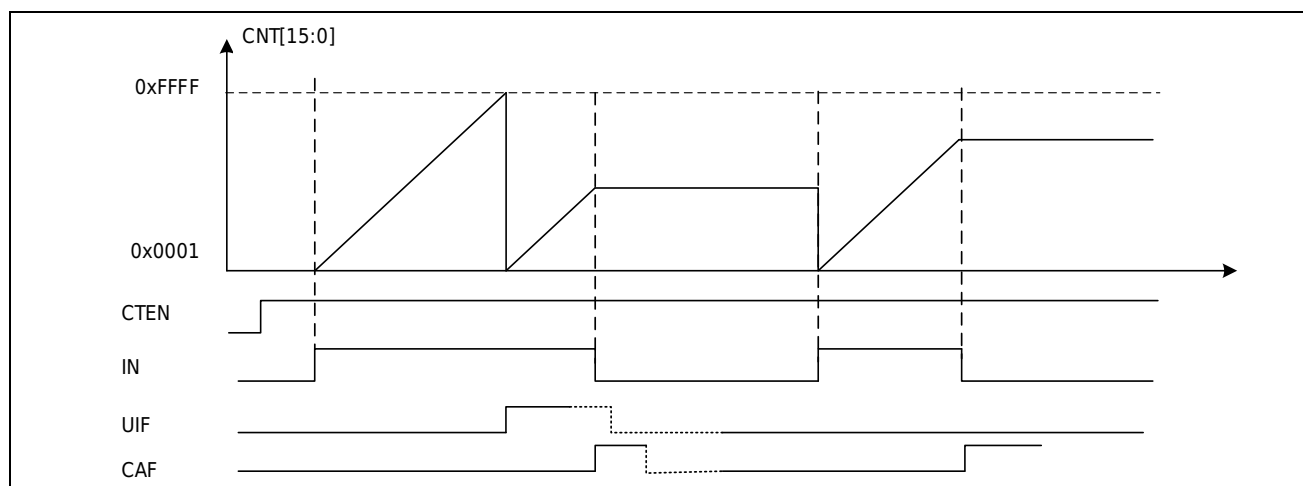


图 14-10 上升沿到下降沿高电平测量

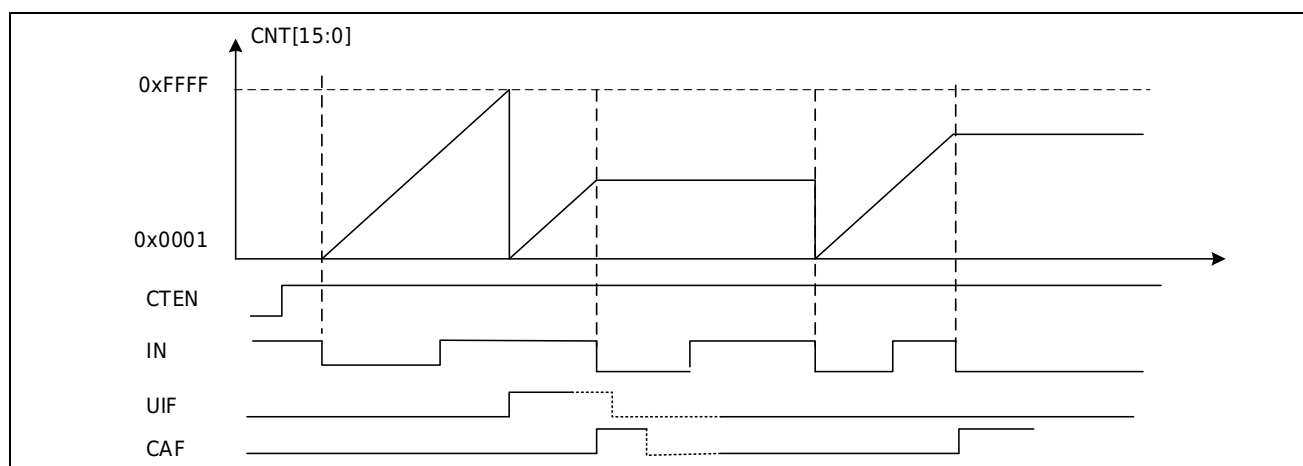


图 14-11 下降沿到下降沿周期测量

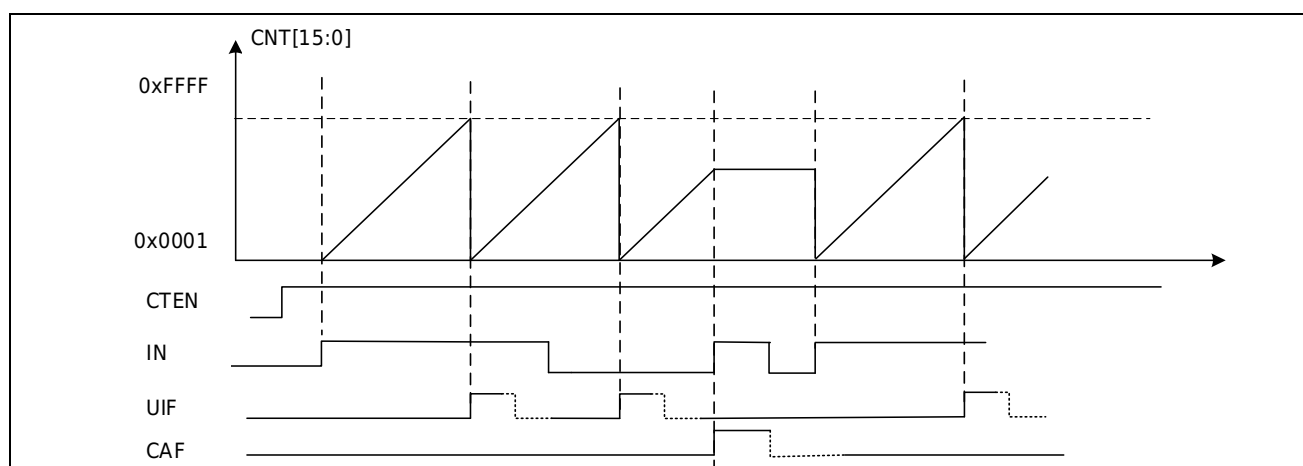


图 14-12 上升沿到上升沿周期测量

通过寄存 MSCR.TS 选择测量信号源。

000 ETFP: ETR 外部输输入滤波后的相位选择信号, 可选择外部滤波与输入反向

- 001 ITR0: Timer 内部互联信号 0, 其它 timer 的 TRGO 输出
- 010 ITR1: Timer 内部互联信号 1, 其它 timer 的 TRGO 输出
- 011 ITR2: Timer 内部互联信号 2, 其它 timer 的 TRGO 输出
- 100 ITR3: Timer 内部互联信号 3, 其它 timer 的 TRGO 输出
- 101 IA0ED: 无效
- 110 IAFP: CH0A 外部输输入滤波后的相位选择信号, 可选择外部滤波
- 111 IBFP: CH0B 外部输输入滤波后的相位选择信号, 可选择外部滤波

14.3.5.3 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。

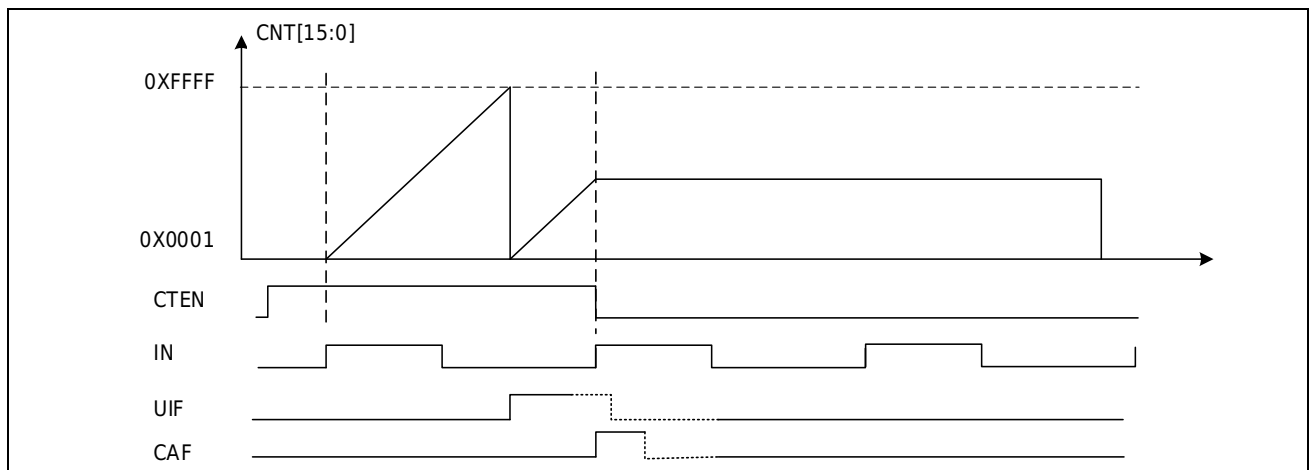


图 14-13 上升沿到上升沿周期测量单次模式

14.3.5.4 设置示例

脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE
6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN
8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志

9. 等待下次测量

脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量高电平
5. 清除中断标志
6. 使能溢出中断 M1CR.UIE
7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

14.3.6 模式 2/3 比较捕获模式

14.3.6.1 计数器

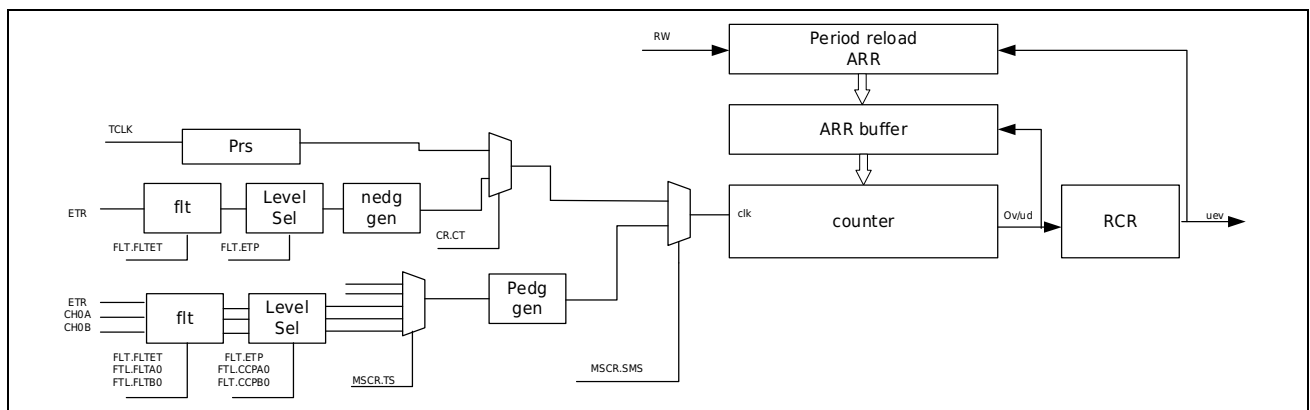


图 14-14 计数器框图

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。

计数器基本单元包括：

- 计数器寄存器 CNT
- 预分频寄存器 CR.PRS
- 自动重载寄存器 ARR

- 重复计数寄存器 RCR
- 时钟选择控制寄存器 FLT, CR0, MSCR, CR

自动重载寄存器 ARR 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, 自动重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且缓存功能有效时, 写入到自动重载寄存器的值不会立刻更新到缓存寄存器, 当事件更新后才从自动重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, 复位功能参考【模式 2/3 从模式】章节。

14.3.6.2 计数器波形

模式 2 为锯齿波计数波形, 通过设置 CR.DIR 可以更改计数方向。

设置 CR.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 (ATIMx_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寄存器中编程的次数加一次 (ATIMx_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 ATIMx_CR 寄存器的 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 (ATIMx_IFR 寄存器中的 UIF 位) 置 1 (取决于 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

以下图示显示 ARR=0x2C 时不同计数方向的计数器波形:

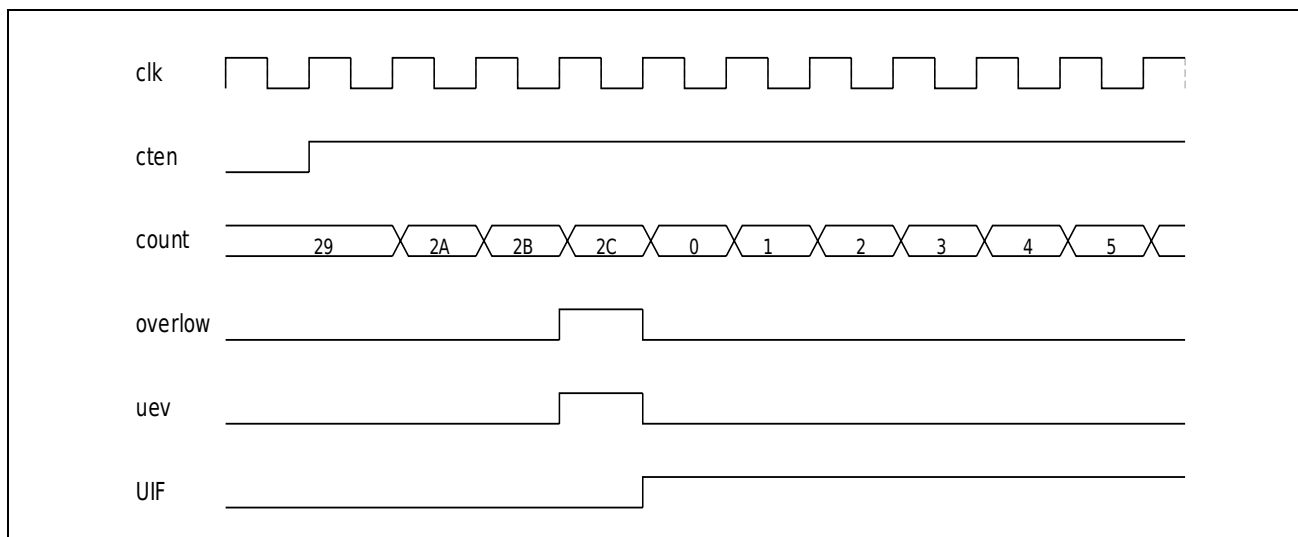


图 14-15 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1,

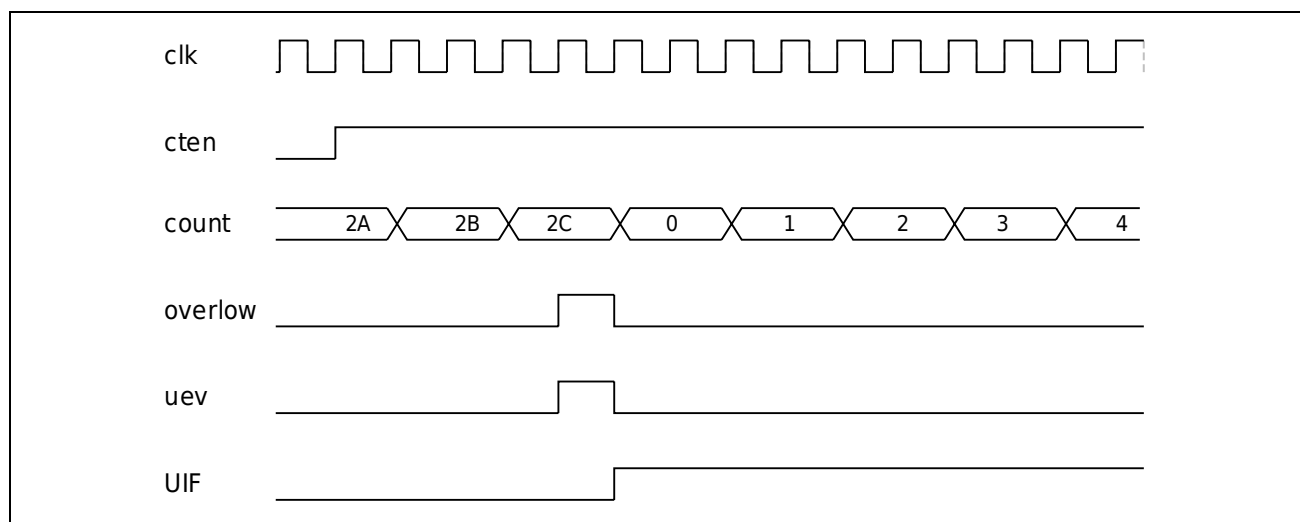


图 14-16 带预分频的上计数

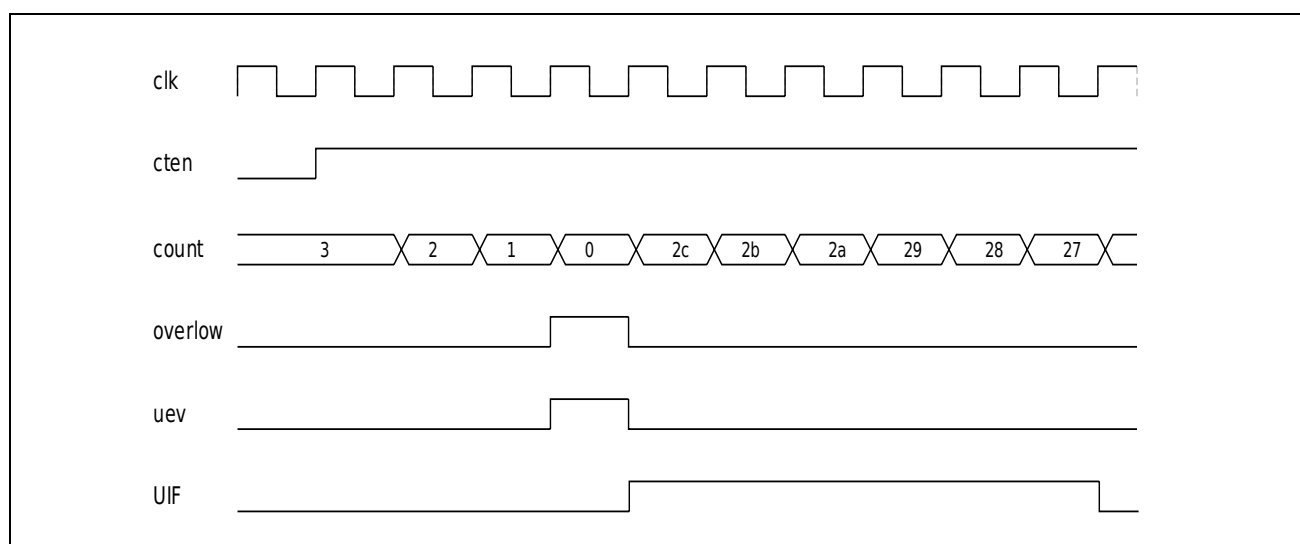


图 14-17 不带预分频的下计数

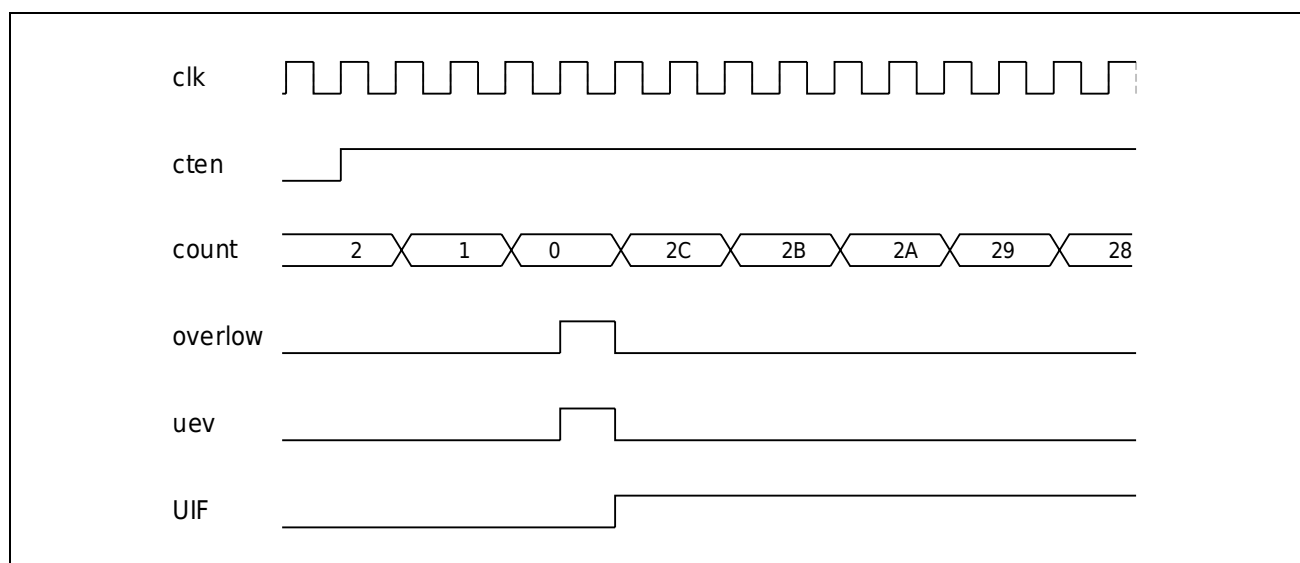


图 14-18 带预分频的下计数

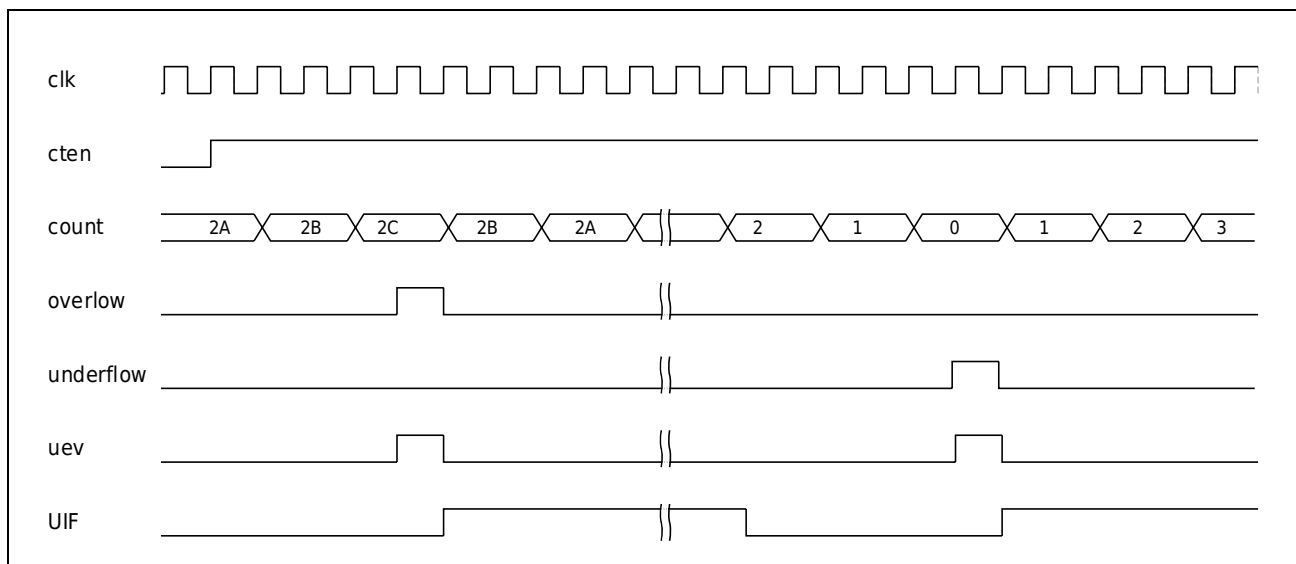


图 14-19 带预分频的上下计数

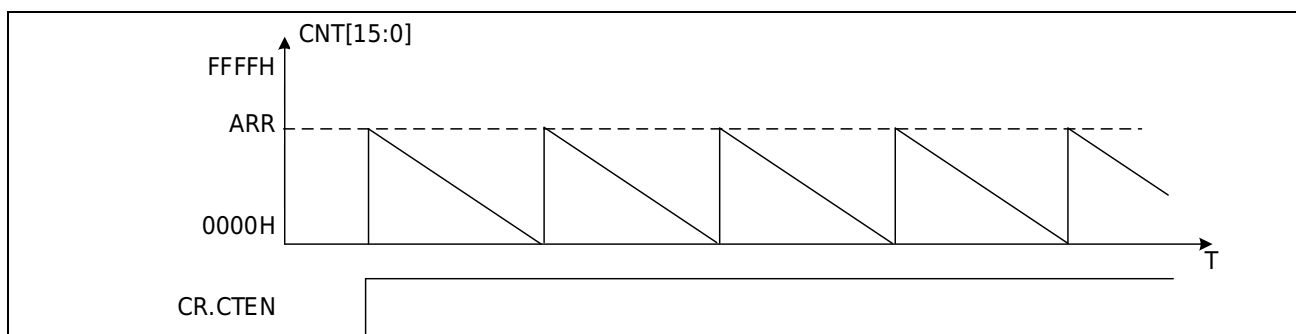


图 14-20 边沿对齐计时器波形(DIR =1)

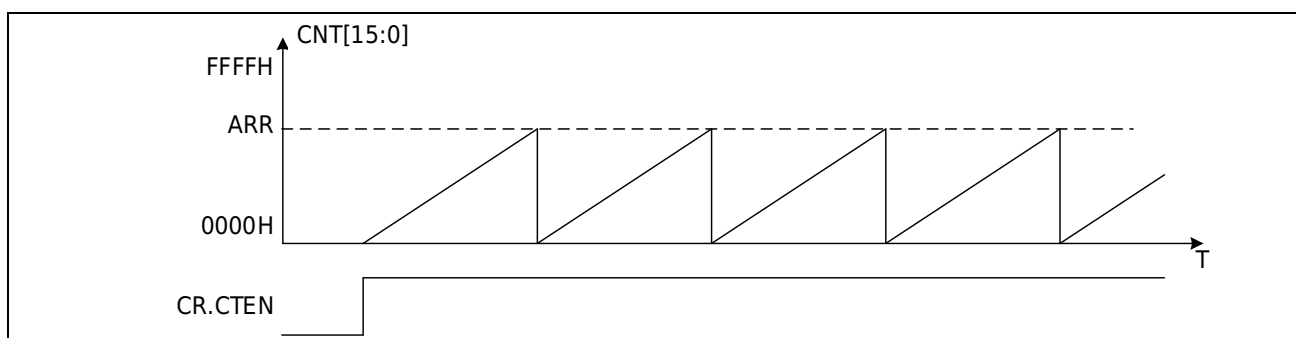


图 14-21 边沿对齐计时器波形(DIR =0)

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 CR.DIR 方向位是只读的，写值无效。从其它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。

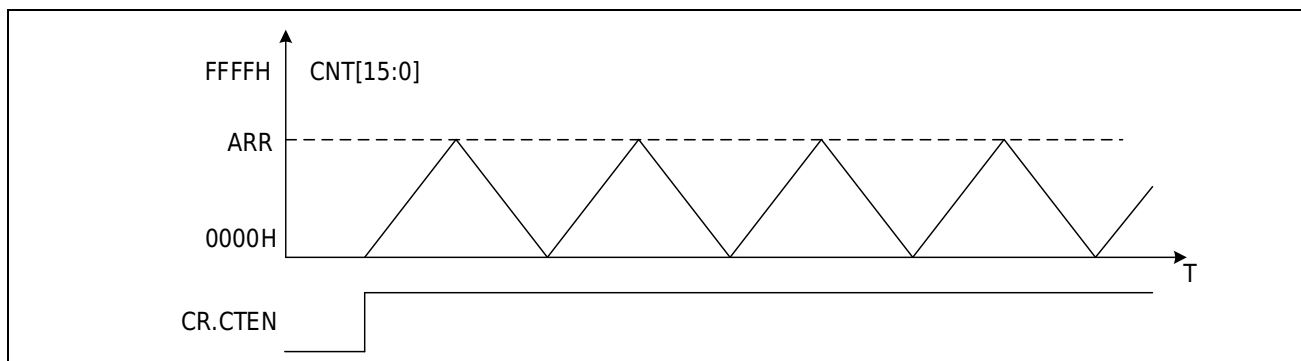


图 14-22 中心对齐计数器波形

14.3.6.3 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，自动重载寄存器更新到自动重载缓存寄存器。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 $RCR.UD = 0$ 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

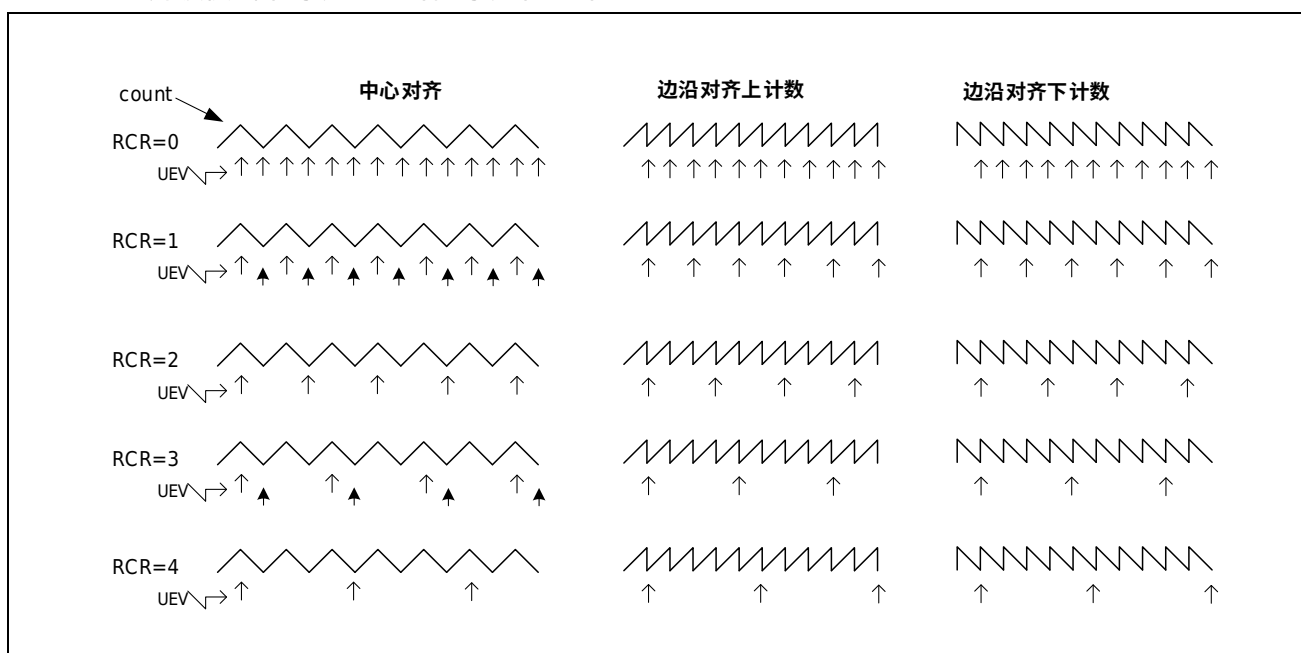


图 14-23 重复计数器产生更新时序 ($RCR.UD=0$ $RCR.OV=0$)

$RCR.OV$ 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；

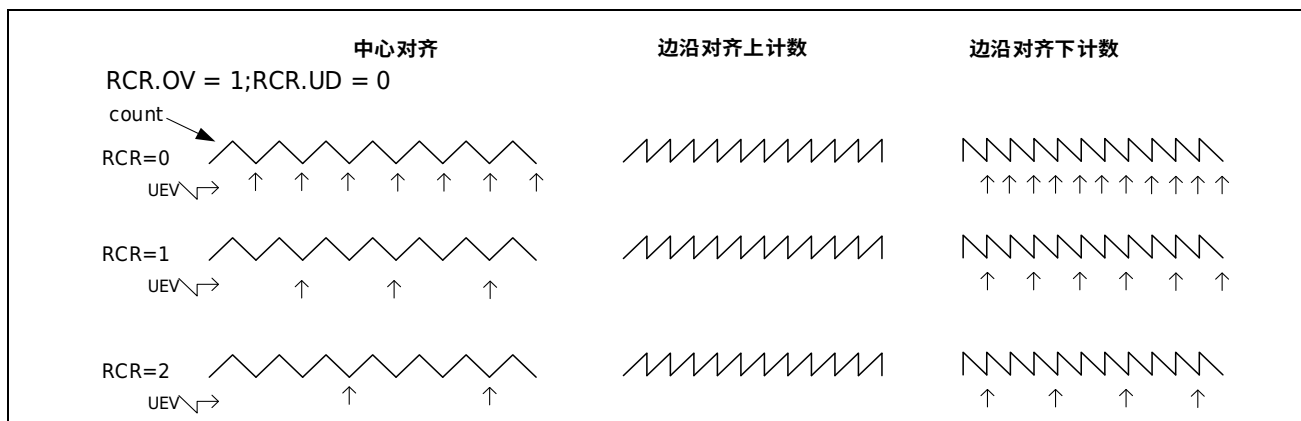


图 14-24 上溢屏蔽时候重复计数器产生更新时序 (RCR.OV=1, RCR.UD=0)

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

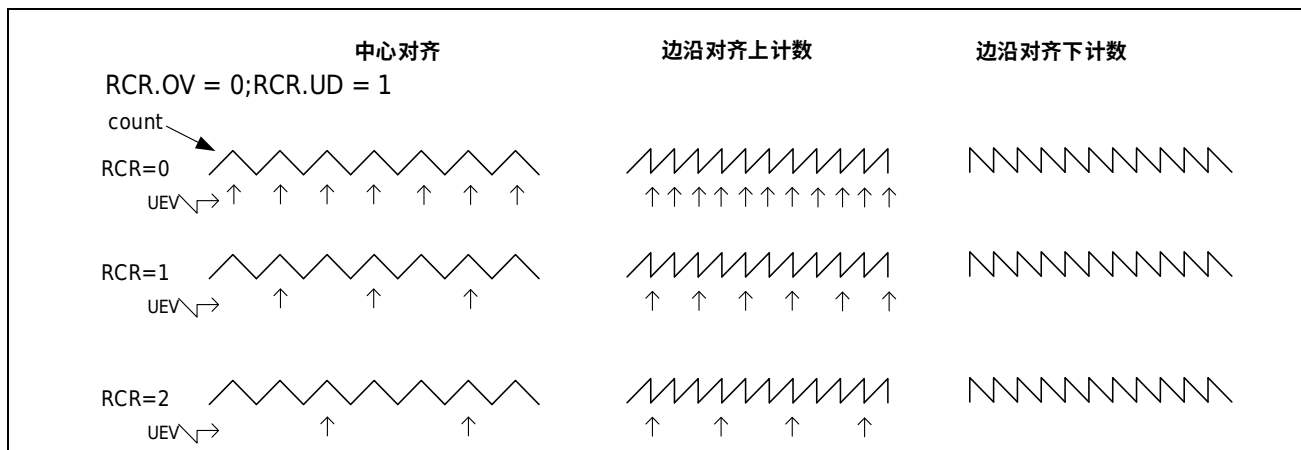


图 14-25 下溢屏蔽时候重复计数器产生更新时序 (RCR.OV=0, RCR.UD=1)

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 CR.UG 产生软件及从模式复位事件更新 UEV; 这时需要配置 CR.URS。

14.3.6.4 数据缓存

自动重载寄存器 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：

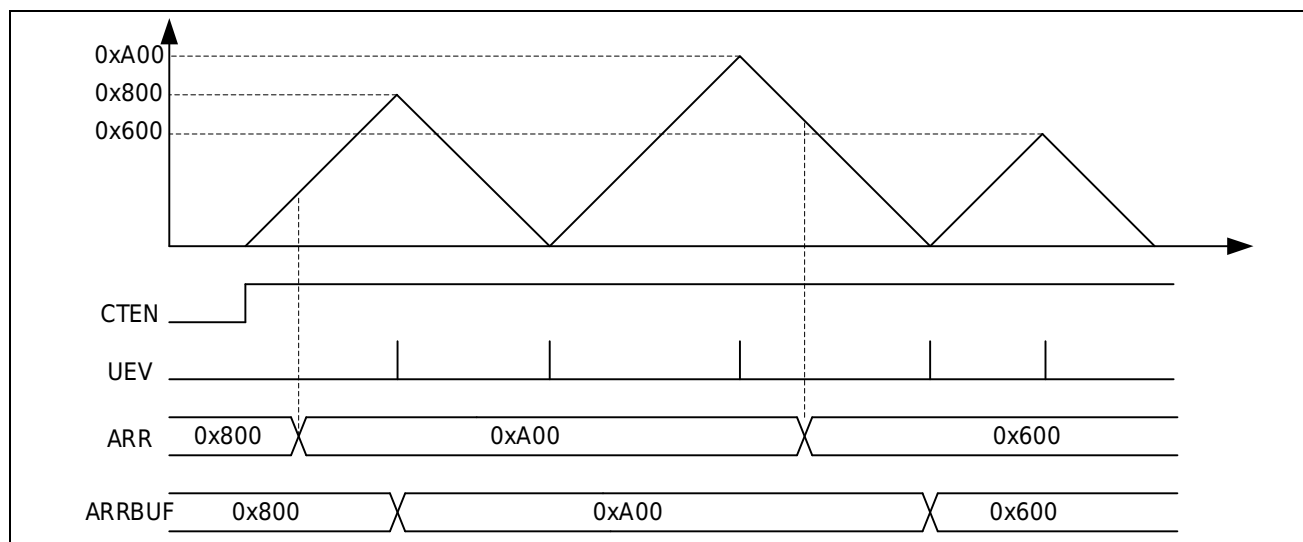


图 14-26 三角波模式缓存使能

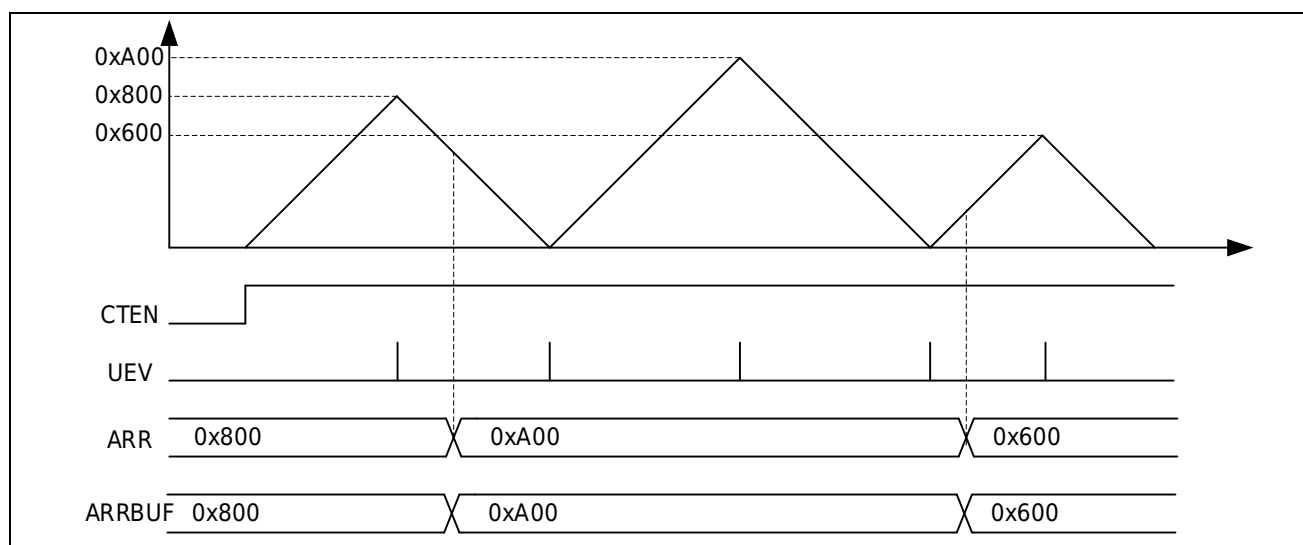


图 14-27 三角波模式缓存无效

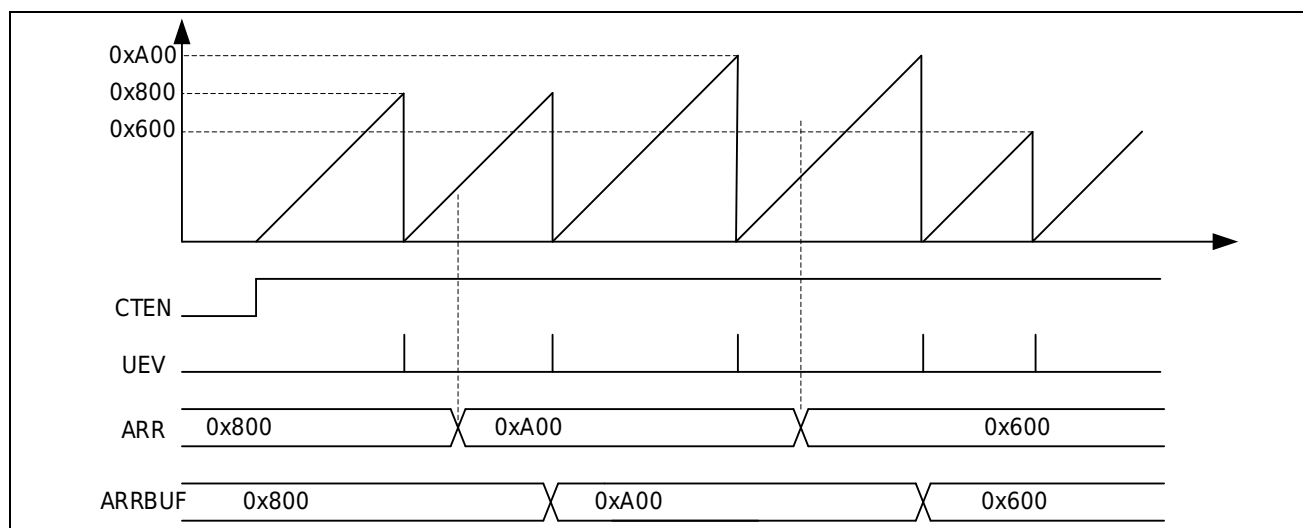


图 14-28 锯齿波模式上计数缓存使能

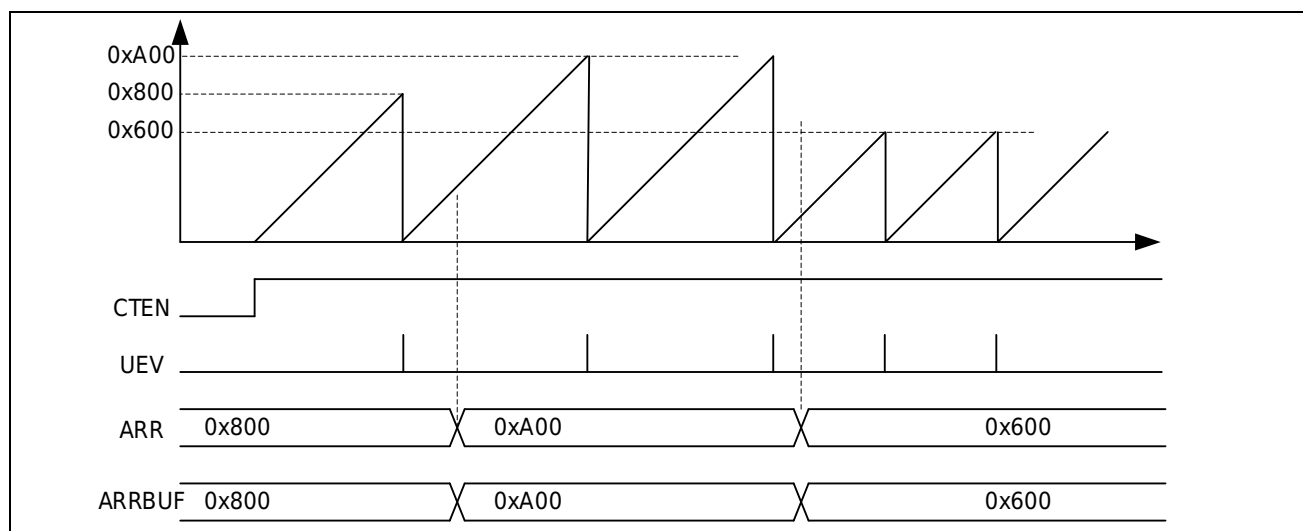


图 14-29 锯齿波模式上计数缓存无效

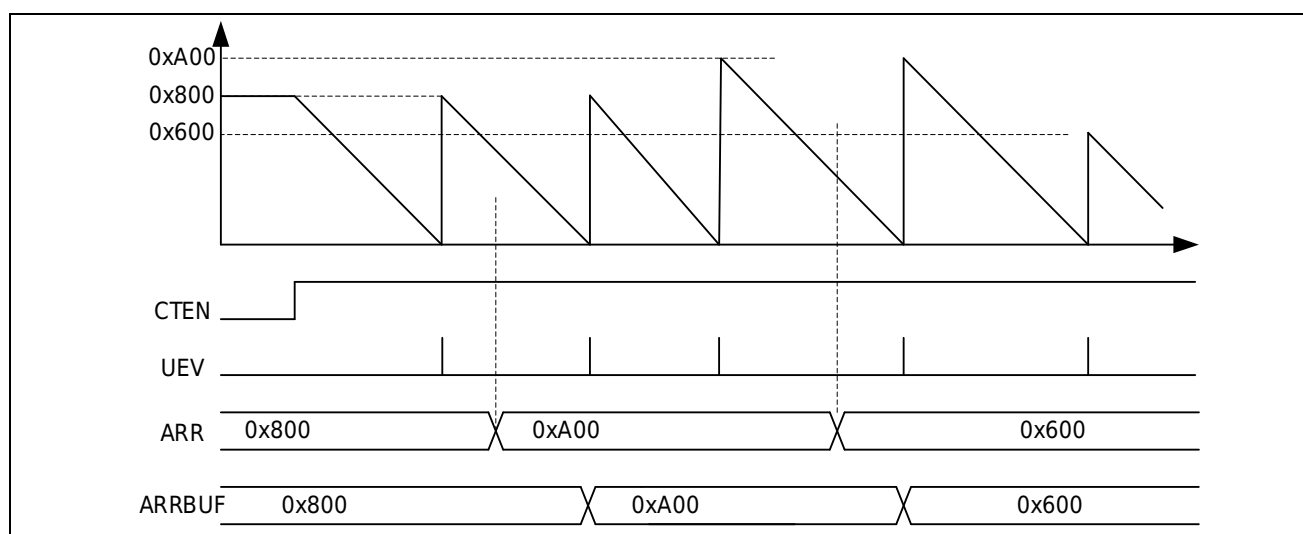


图 14-30 锯齿波模式下计数缓存使能

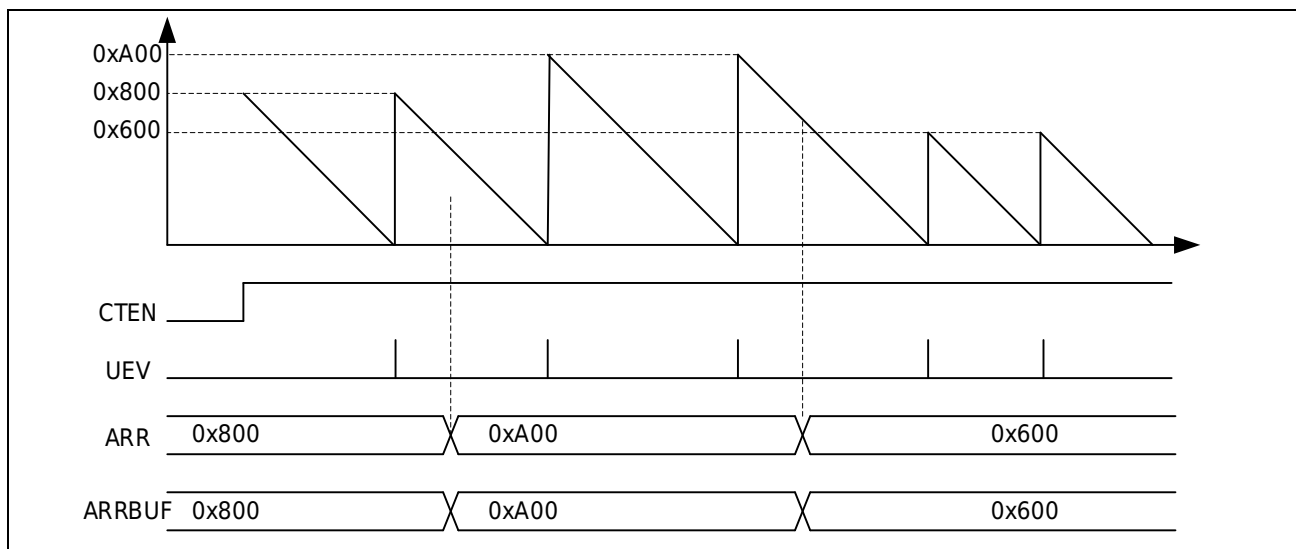


图 14-31 锯齿波模式下计数缓存无效

注意：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。

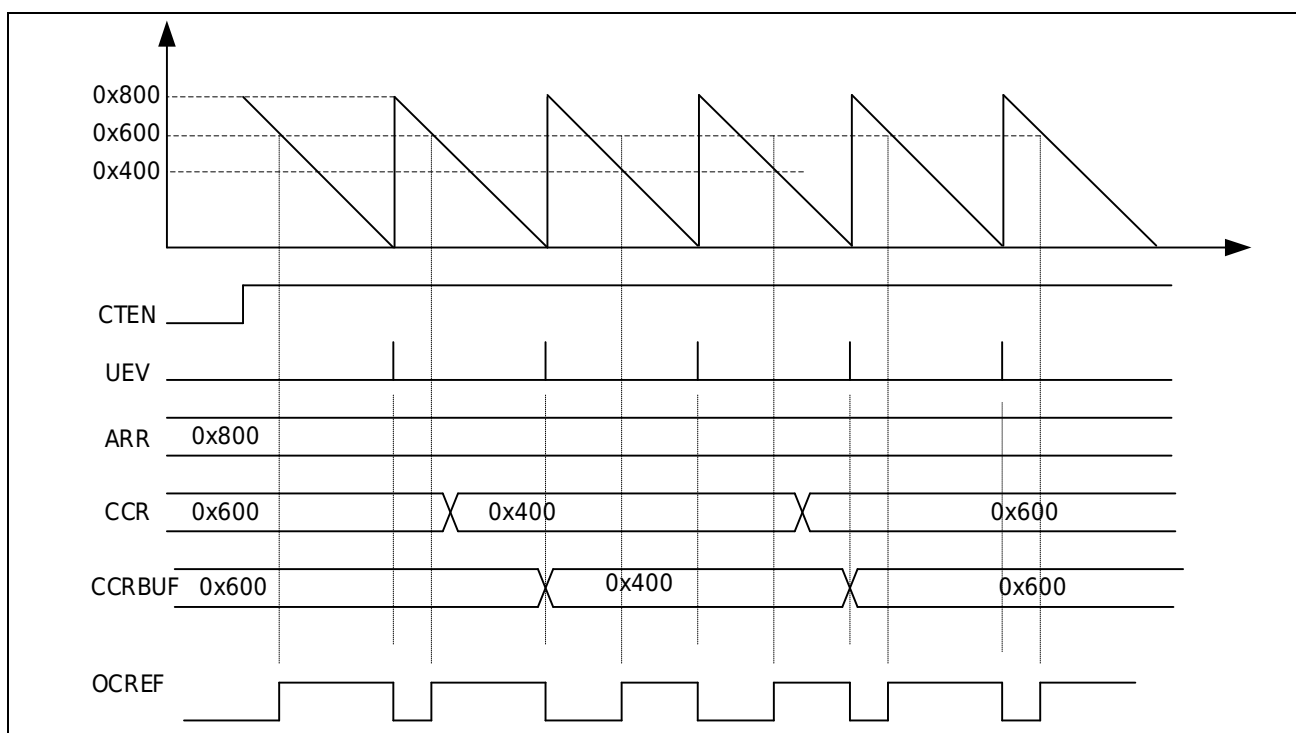


图 14-32 锯齿波模式下计数比较缓存使能

14.3.6.5 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。

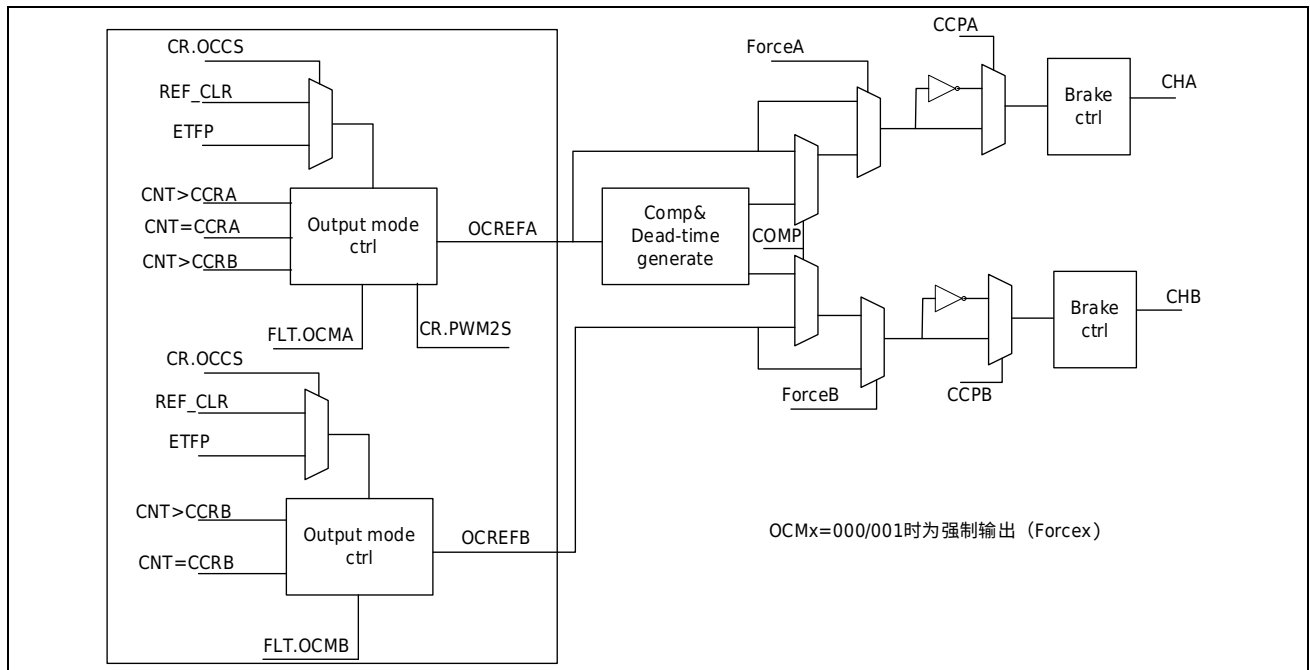


图 14-33 OCREF 输出框图

OCREF 输出使用 OCMx 选择

000: 强制为 0

001: 强制为 1

010: 比较匹配时强制为 0

011: 比较匹配时强制为 1

100: 比较匹配时翻转

101: 比较匹配时输出一个计数周期的高电平

110: PWM 模式 1

单点比较:

上计数时 $CNT < CCR_{xy}$ 输出高, 下计数时 $CNT > CCR_{xy}$ 输出为低电平

双点比较:

- 1) 锯齿波上计数 $CCR_A < CNT \leq CCR_B$ 输出为低电平
- 2) 锯齿波下计数 $CCR_A < CNT \leq CCR_B$ 输出为高电平

3) 三角波上计数 $CNT < CCRxA$ 输出高, 下计数 $CNT > CCRxB$ 为低电平

111: PWM 模式 2

单点比较:

上计数时 $CNT < CCRxy$ 输出低, 下计数时 $CNT > CCRxy$ 输出为高电平

双点比较:

1) 锯齿波上计数 $CCRxA \leq CNT < CCRxB$ 输出为高电平

2) 锯齿波下计数 $CCRxA \leq CNT < CCRxB$ 输出为低电平

3) 三角波上计数 $CNT < CCRxA$ 输出低, 下计数 $CNT > CCRxB$ 为高电平

注: 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。

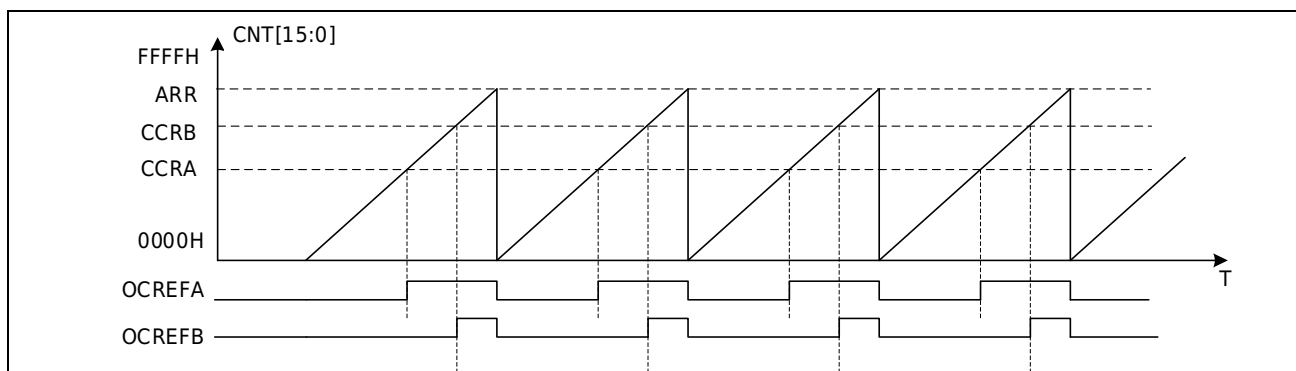


图 14-34 锯齿波计数单点比较 OREF 输出波形 (OCMx=111)

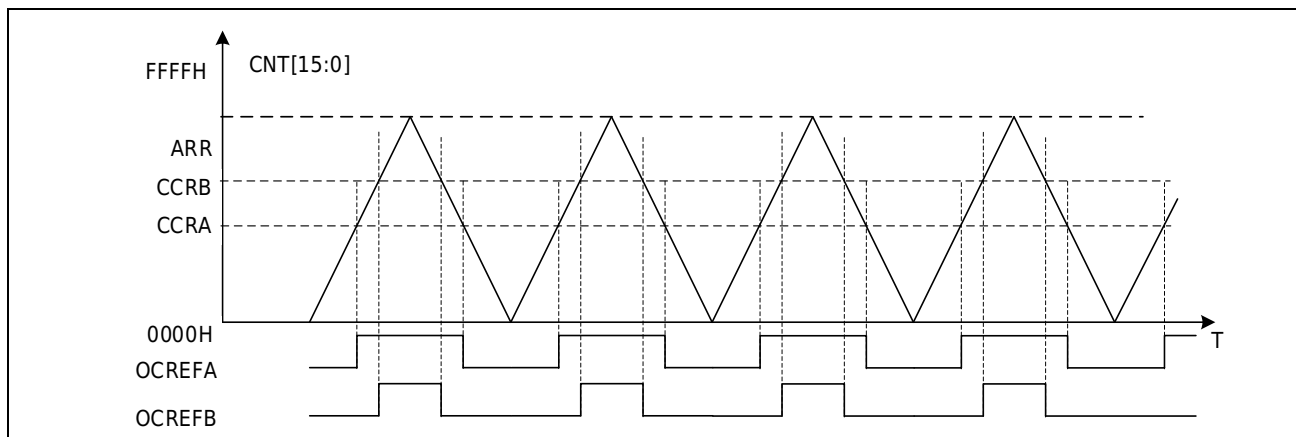


图 14-35 三角波计数单点比较 OREF 输出波形 (OCMx=111)

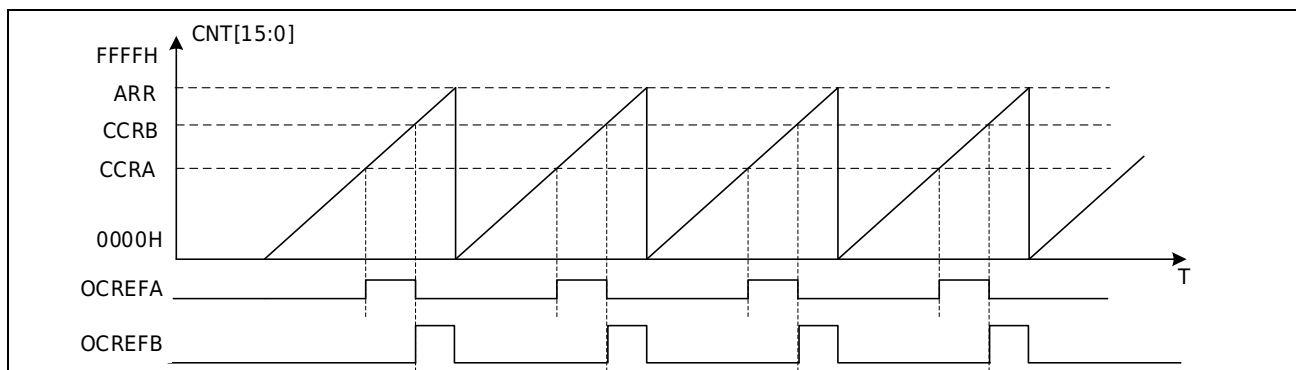


图 14-36 锯齿波计数双点比较 OCREF 输出 (OCMx=111)

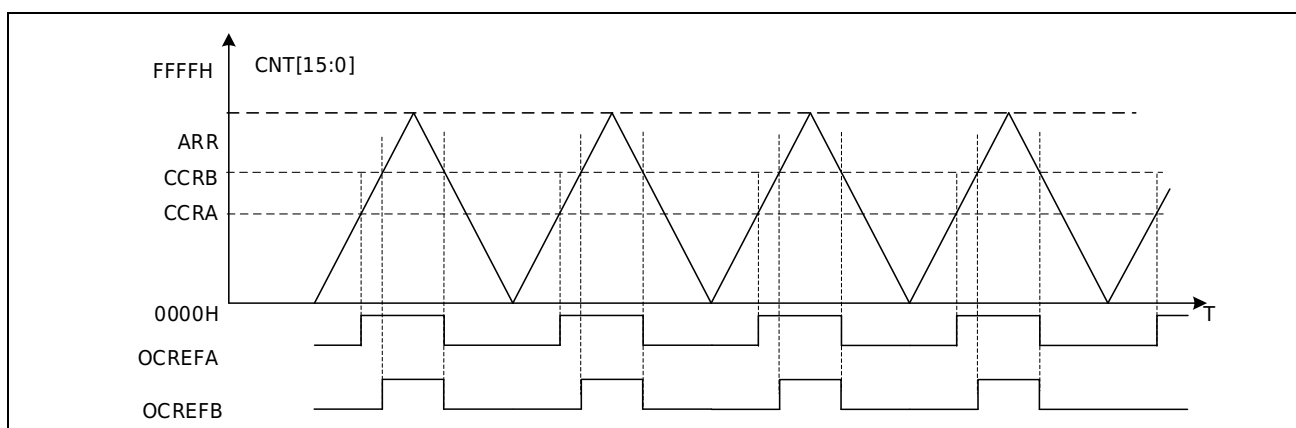


图 14-37 三角波计数双点比较 OCREF 输出 (OCMx=111)

14.3.6.6 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。

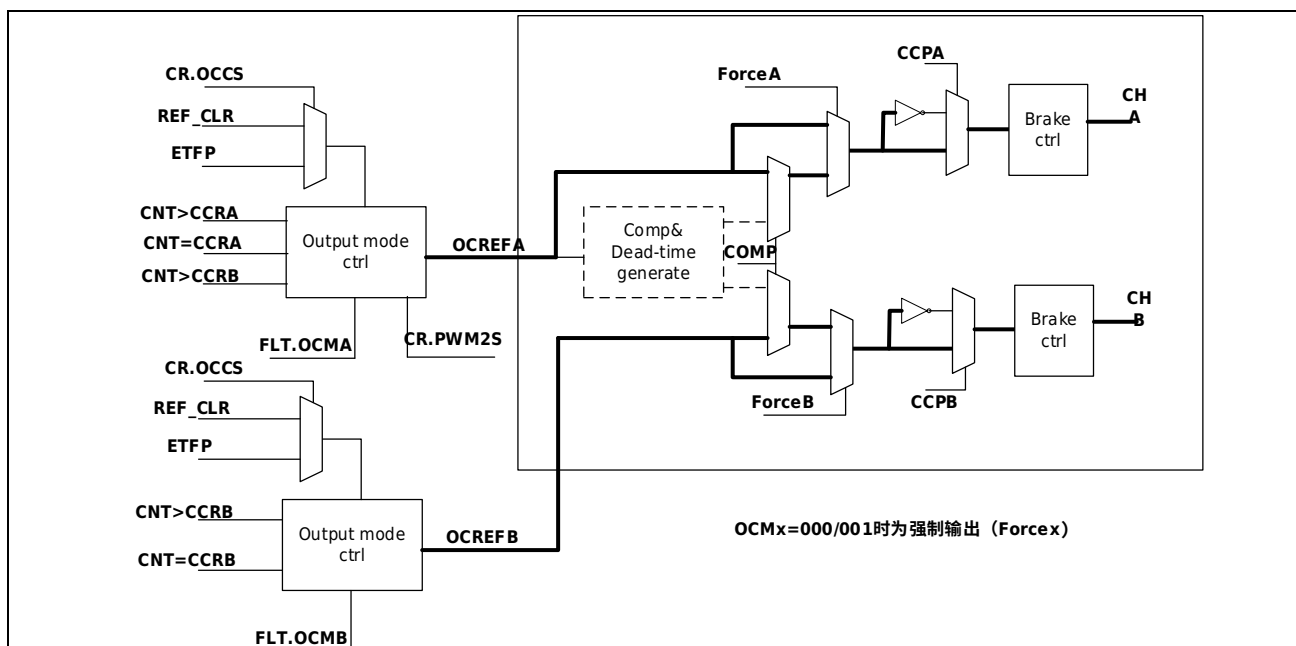


图 14-38 独立 PWM 输出框图

PWM 输出与 OCREF 关系

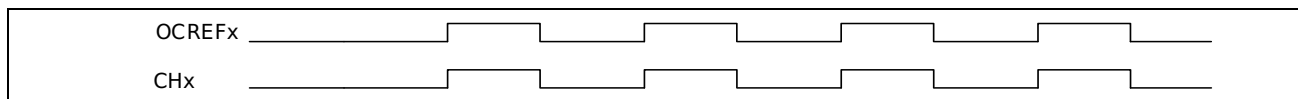


图 14-39 CCPx=0 时 PWM 输出波形

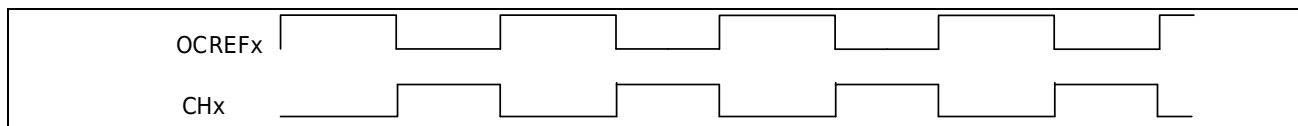


图 14-40 CCPx=1 时 PWM 输出波形

14.3.6.7 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。

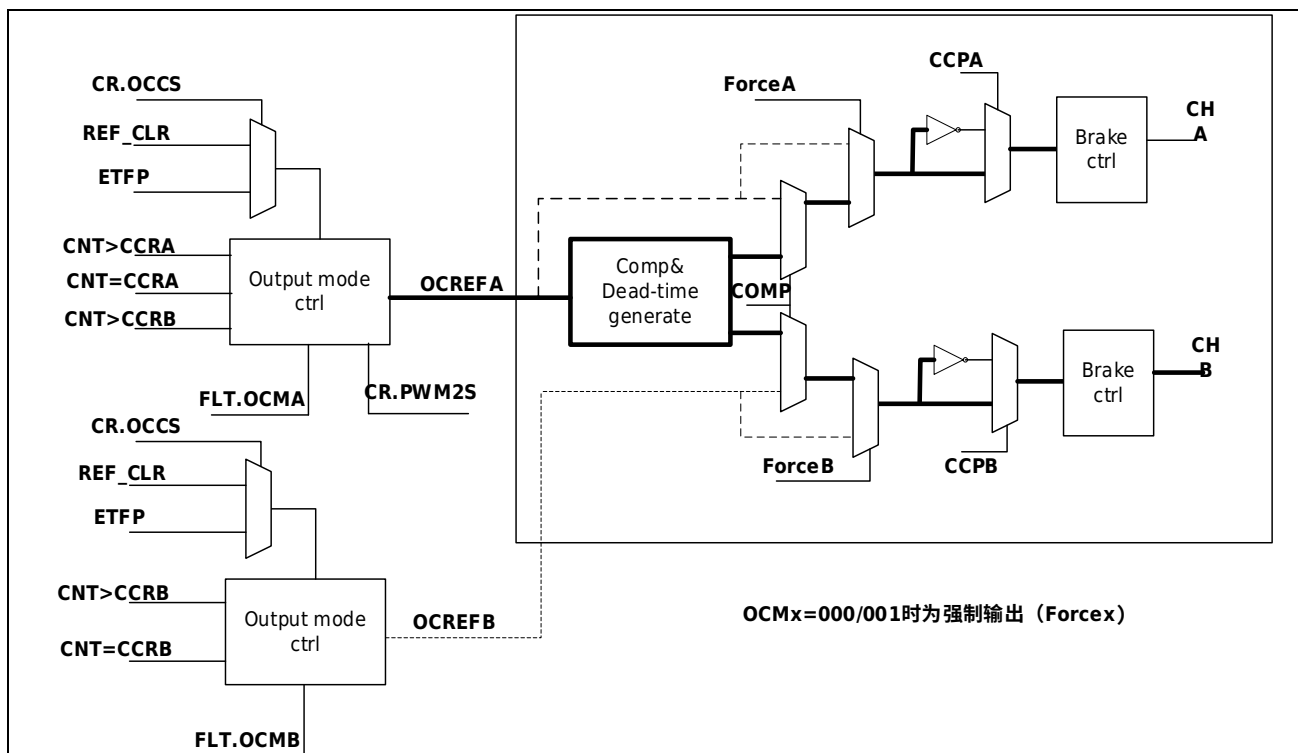


图 14-41 互补 PWM 输出框图

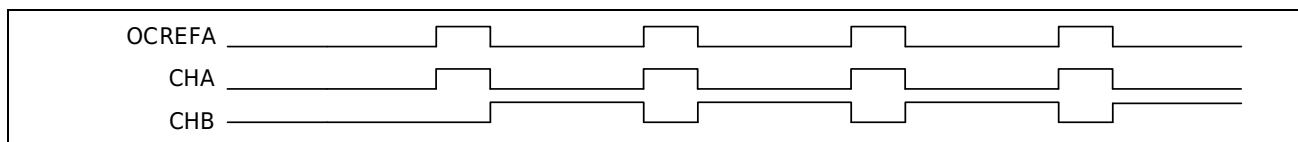


图 14-42 互补 PWM 输出波形图

14.3.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

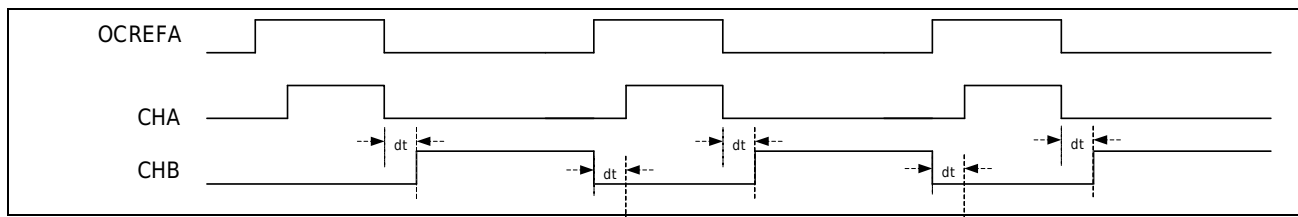


图 14-43 互补 PWM 输出波形图

死区时间使用 8 位 DTR 控制，死区时间 dt 与 DTR 的关系如下

DTR[7] = 0	$T = DTR[6:0] + 2$	2-129	step=1
DTR[7:6] = 10	$T = \{DTR[5:0] + 64\} * 2 + 2$	130-256	step=2
DTR[7:5] = 110	$T = \{DTR[4:0] + 32\} * 8 + 2$	258-506	step=8
DTR[7:5] = 111	$T = \{DTR[4:0] + 32\} * 16 + 2$	514-1010	step=16

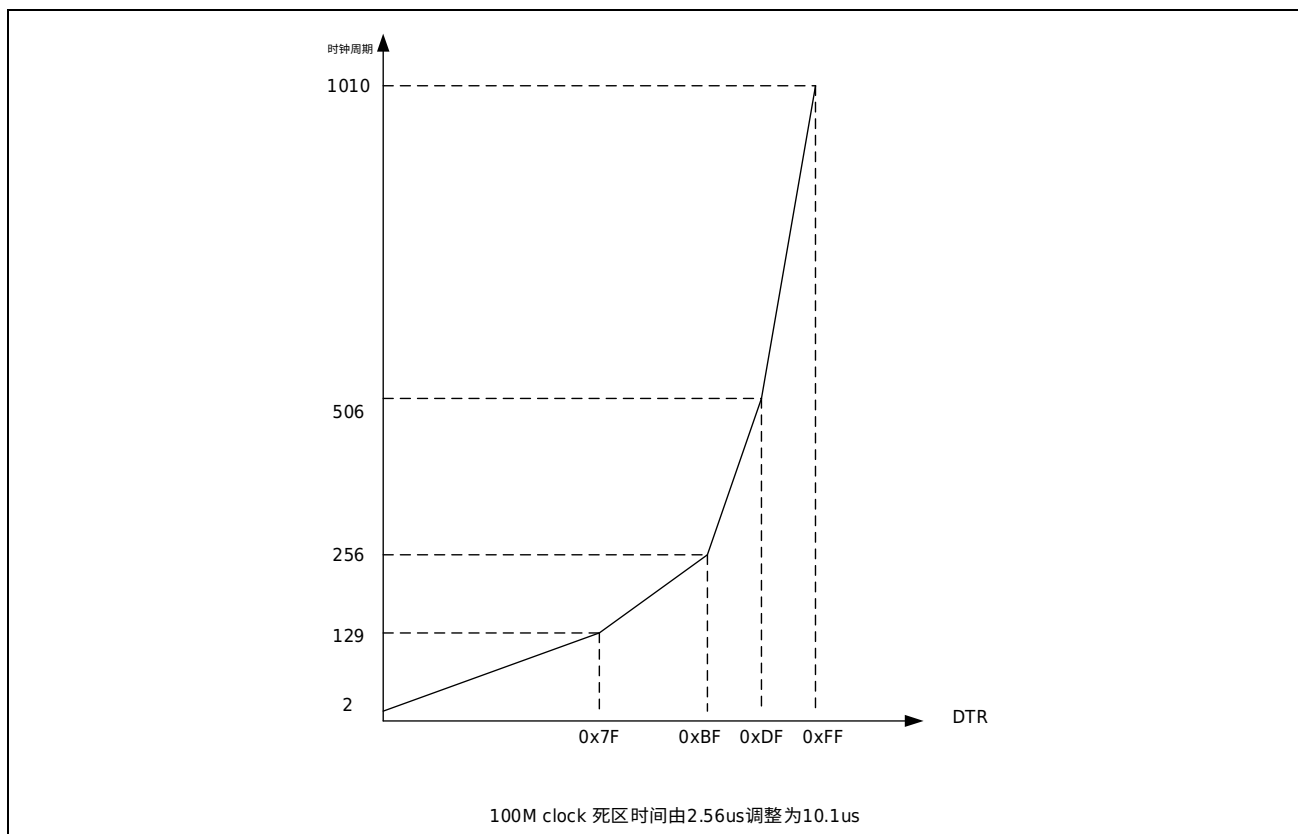


图 14-44 死区时间

14.3.6.9 单脉冲输出

单脉冲模式(ONE SHOT)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 ATIMx_M23CR 寄存器中的 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。

因为发生下一个更新事件 UEV 时，计数器才会停止，因此当 RCR 设置初始值不为 0 的时候，定时器会经过 RCR 值+1 次溢出事件后才会停止。如果 RCR 寄存器的上溢或者下溢屏蔽功能使能，则需 RCR 计数的时候需要忽略对应的屏蔽溢出事件。

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， $0 < CCRxy$ ）
- 递减计数时： $CNT > CCRxy$

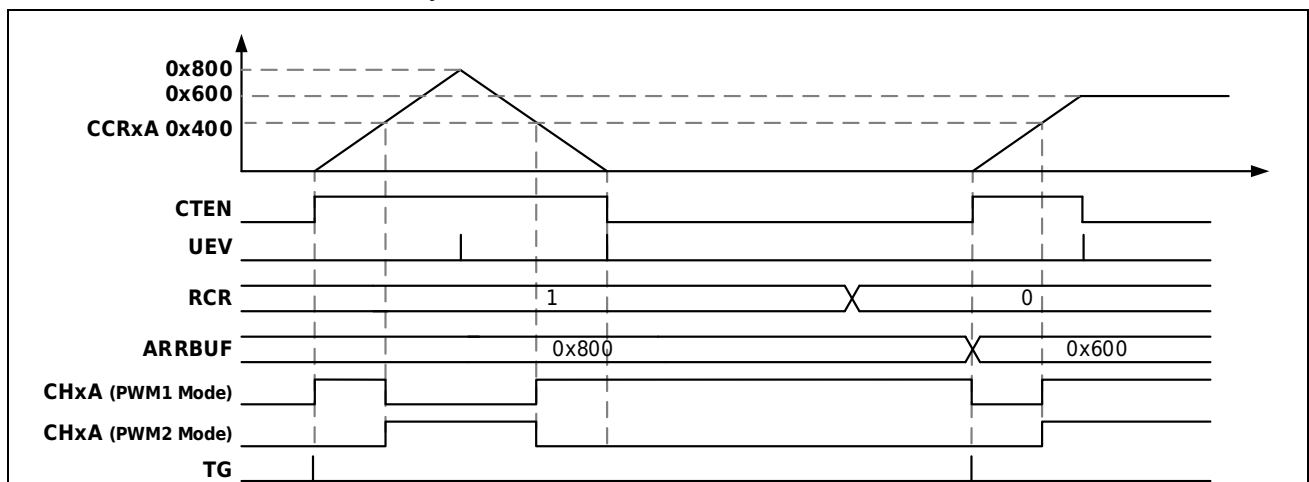


图 14-45 三角波模式单脉冲计数

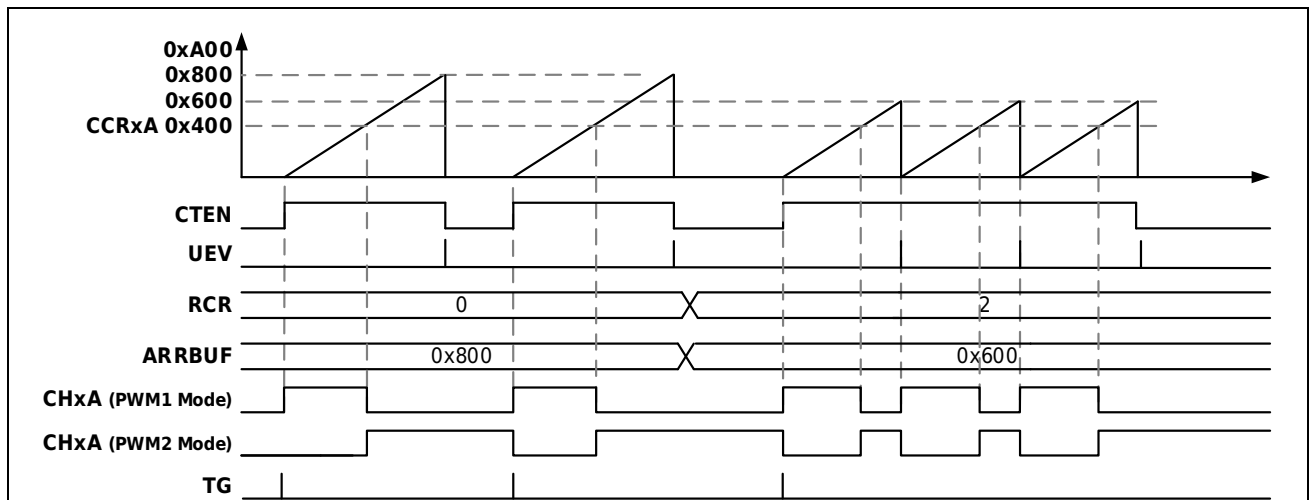


图 14-46 锯齿波上计数单脉冲模式

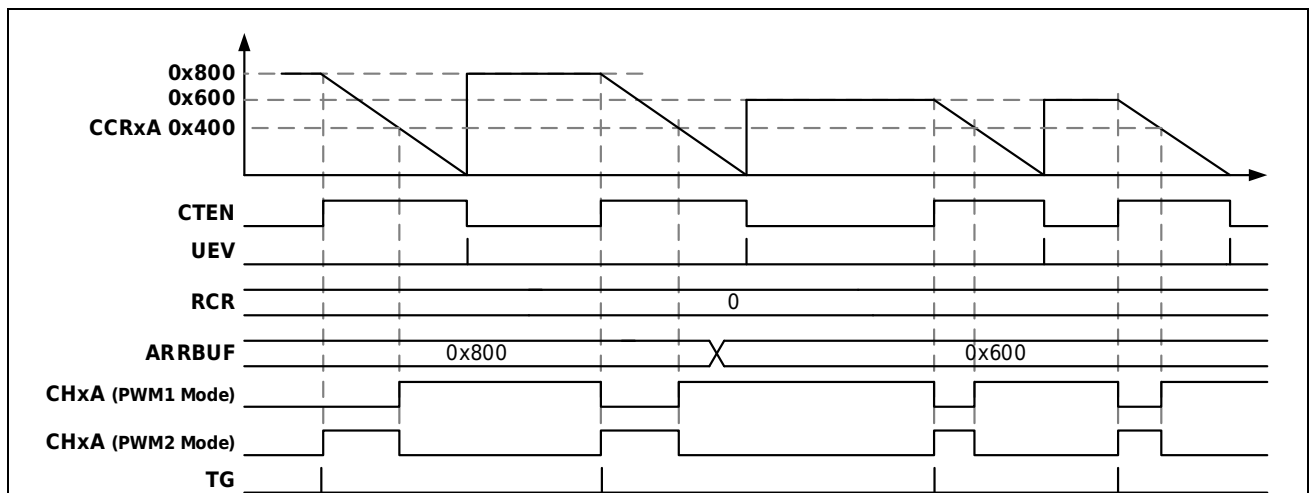


图 14-47 锯齿波下计数单脉冲模式

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。

14.3.6.10 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=2'b00 时 比较匹配无输出

M23CR.CIS=2'b01 时 上计数时比较匹配

M23CR.CIS=2'b10 时 下计数时比较匹配

M23CR.CIS=2'b11 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=2'b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2'b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2'b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2'b11 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。详见【Timer 触发 ADC】章节。

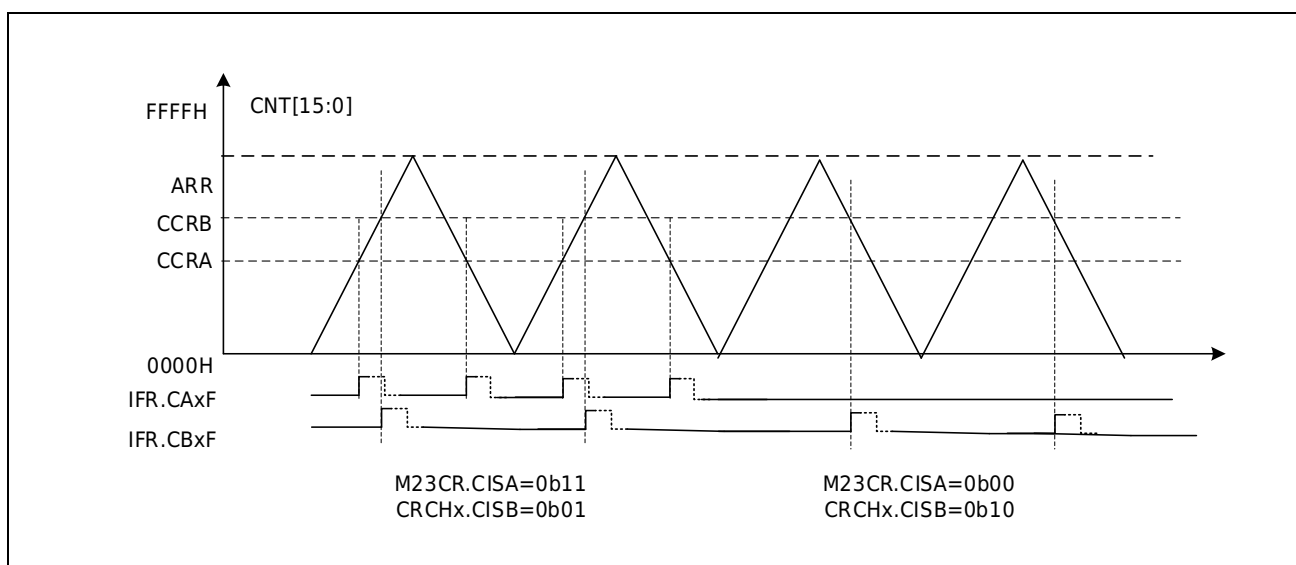


图 14-48 中断示意图

14.3.6.11 捕获输入

M23CR.MODE=2/3

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，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。

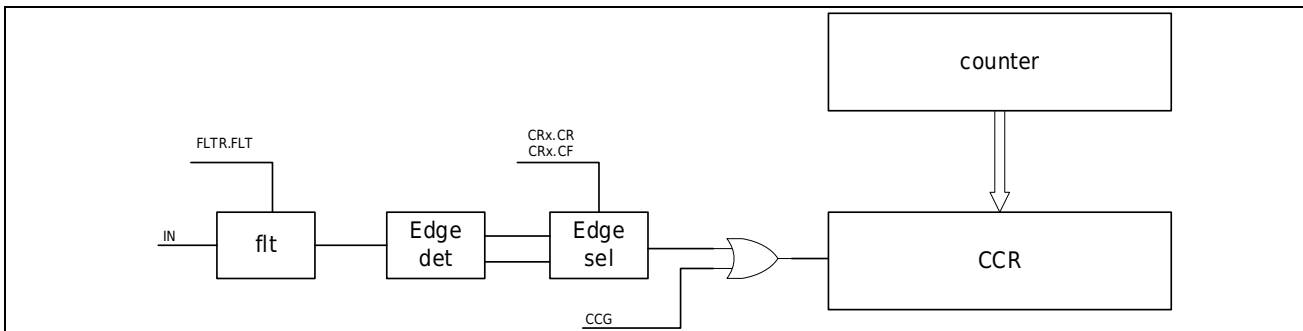


图 14-49 捕获功能框图

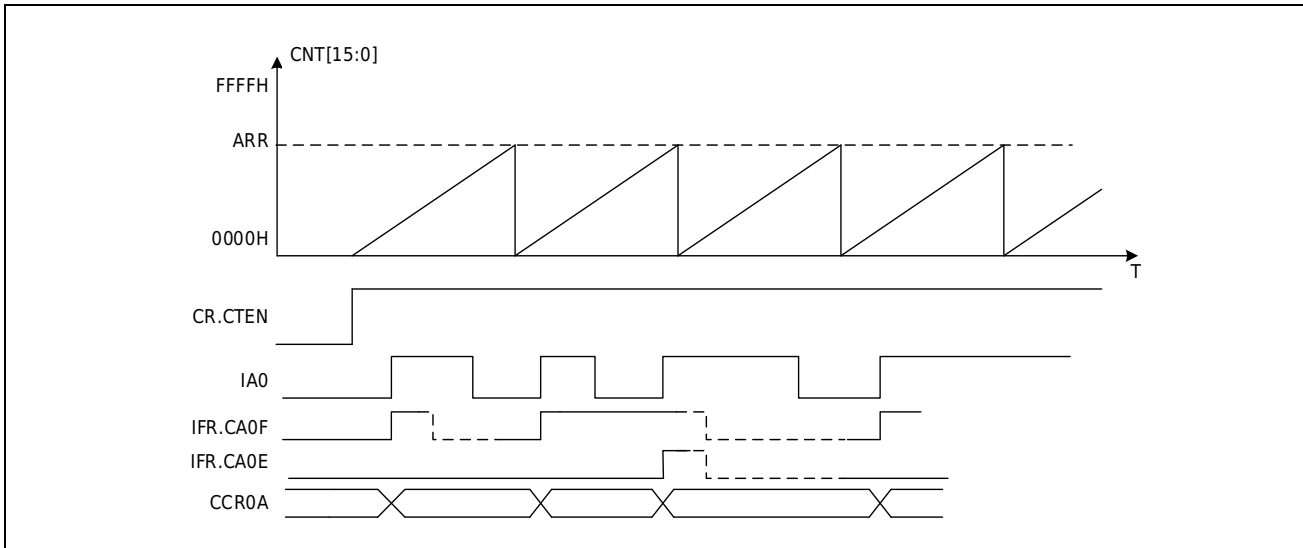


图 14-50 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A，CH1A，CH2A 的异或输入。

表 14-5 捕获输入 CH0A 信号列表

IA0S	0	1
ATIM3	CH0A	CH0A CH1A CH2A异或输入
ATIM0	CHA	CHA ETR GATE异或输入

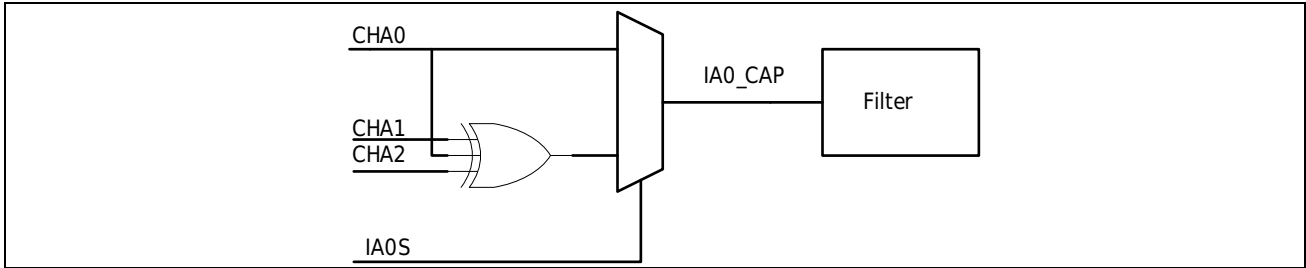


图 14-51 ATIM3 CH0A 端口选择

CH0B 的捕获输入通过 MSCR.IB0S 选择 CH0B 输入还是内部触发 MSCR.TS 选择的信号。

表 14-6 捕获输入 CH0B 信号列表

IB0S	0	1	
ATIM3	CHB0	内部触发MSCR.TS选择信号	
		MCSCR.TS	通道B捕获输入
		000	ETR滤波相位选择输出信号，滤波相位选择可配
		001	内部互联ITR0
		010	内部互联ITR1
		011	内部互联ITR2
		100	内部互联ITR3
		101	CH0A边沿
		110	CH0A滤波输出信号，滤波功能可配
		111	Reserved

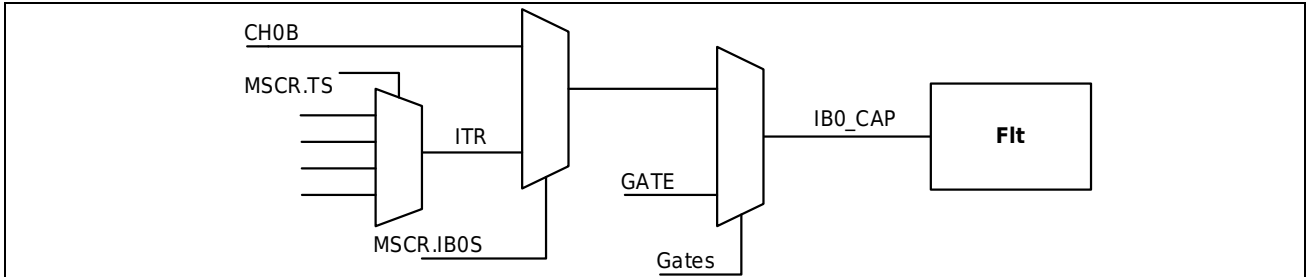


图 14-52 CHB 端口选择

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 时，GATE 输出有效）。

14.3.6.12 设置示例

边沿对齐独立 PWM 输出设置

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

中心对齐互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
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)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 M23CR.CTEN

三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置两个比较使能 M23CR.PWM2S
4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
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

7. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断
10. 设置输出极性 FLT.CCPAx FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DT
13. 使能输出 DTR.MOE
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

1. 选择计数器计数方式，设置 mode=2
2. 选择 CH0B 为捕获模式 CRCHx.CSB=1
3. 根据需要选择输入滤波 FLT.FLTB0
4. 选择捕获的边沿 CRCHx.CRB=1
5. 设置 ARR 改变周期值
6. 启动定时器 M23CR.CTEN
7. 清除相关中断标志
8. 清除捕获标志，使能相应中断
9. 查询到捕获标志后，读取 CCR0B 获得捕获值

注：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。

14.3.7 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。

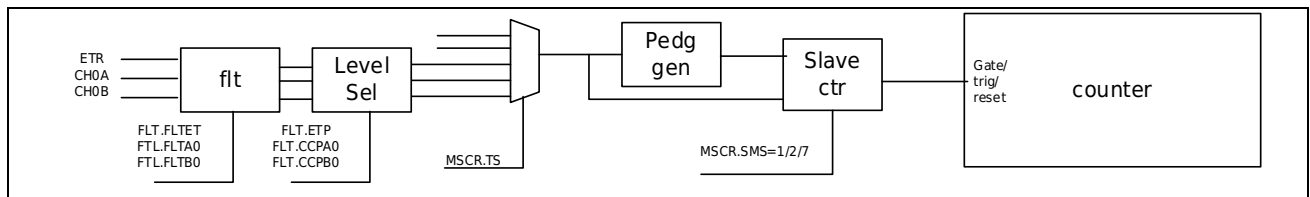


图 14-53 从模式示意图

从模式功能选择通过 MSCR.SMS 选择；

表 14-7 MSCR 寄存器从模式和触发源列表

MSCR.SMS	从模式功能选择 001：复位功能 010：触发模式 111：门控功能
MSCR.TS	触发选择 000：端口 ETR 的滤波相位选择后的信号 ETFP，滤波功能可配置 001：内部互联信号 ITR0 010：内部互联信号 ITR1 011：内部互联信号 ITR2 100：内部互联信号 ITR3 101：端口 CH0A 的边沿信号； 110：端口 CH0A 的滤波相位选择后的信号 IAFP，滤波功能可配置 111：端口 CH0B 的滤波相位选择后的信号 IBFP，滤波功能可配置

14.3.7.1 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 FLT.FLTA0=000）。置 FLTR 寄存器中 CCPA0=1 以确定极性（只检测低电平）。
- 置 MSCR 寄存器中 SMS=111，配置定时器为门控模式；置 MSCR 寄存器中 TS=110，选择 CH0A 作为输入源。
- 置 CR 寄存器中 CTEN=1，启动计数器，当 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。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。

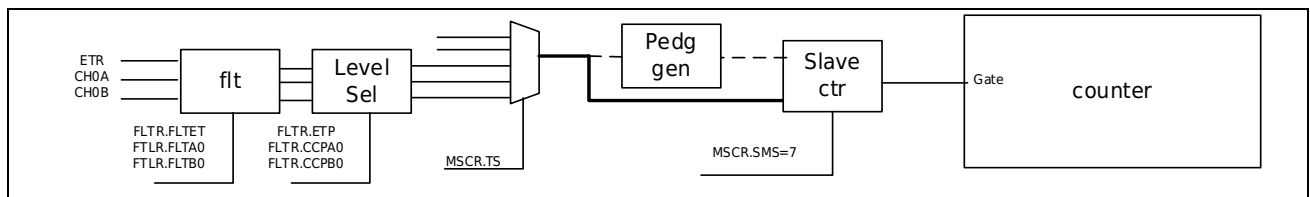


图 14-54 门控功能

14.3.7.2 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.MMS 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 CR.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 FLT.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性 (不反向)。
- 置 MSCR 寄存器中 SMS=010, 配置定时器为触发模式; 置 MSCR 寄存器中 TS=111, 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。

注意: 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。

14.3.7.3 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 CR 寄存器的 URS 位为低, 还产生一个更新事件 UEV; 然后所有的预装载寄存器 (ARR, CCRx) 都被更新了。

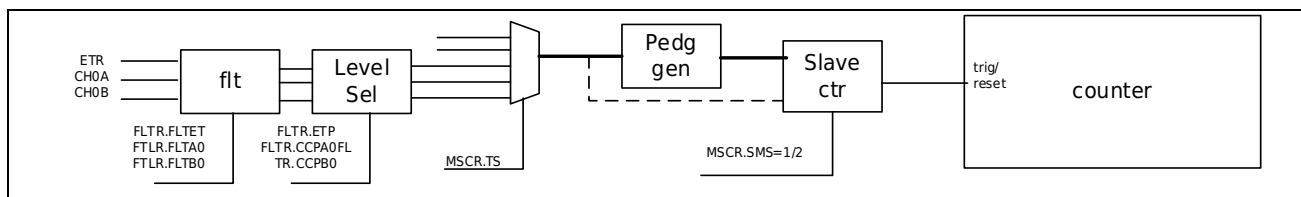


图 14-55 触发和复位功能

14.3.8 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, 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,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。

	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	Rising	falling	Rising	falling
MOD1	High		Down	Up	-	-
	Low		up	Down	-	-
MOD2		High	-	-	up	Down
		Low	-	-	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

注：上表格中白底部分为 CNT 计数动作

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。

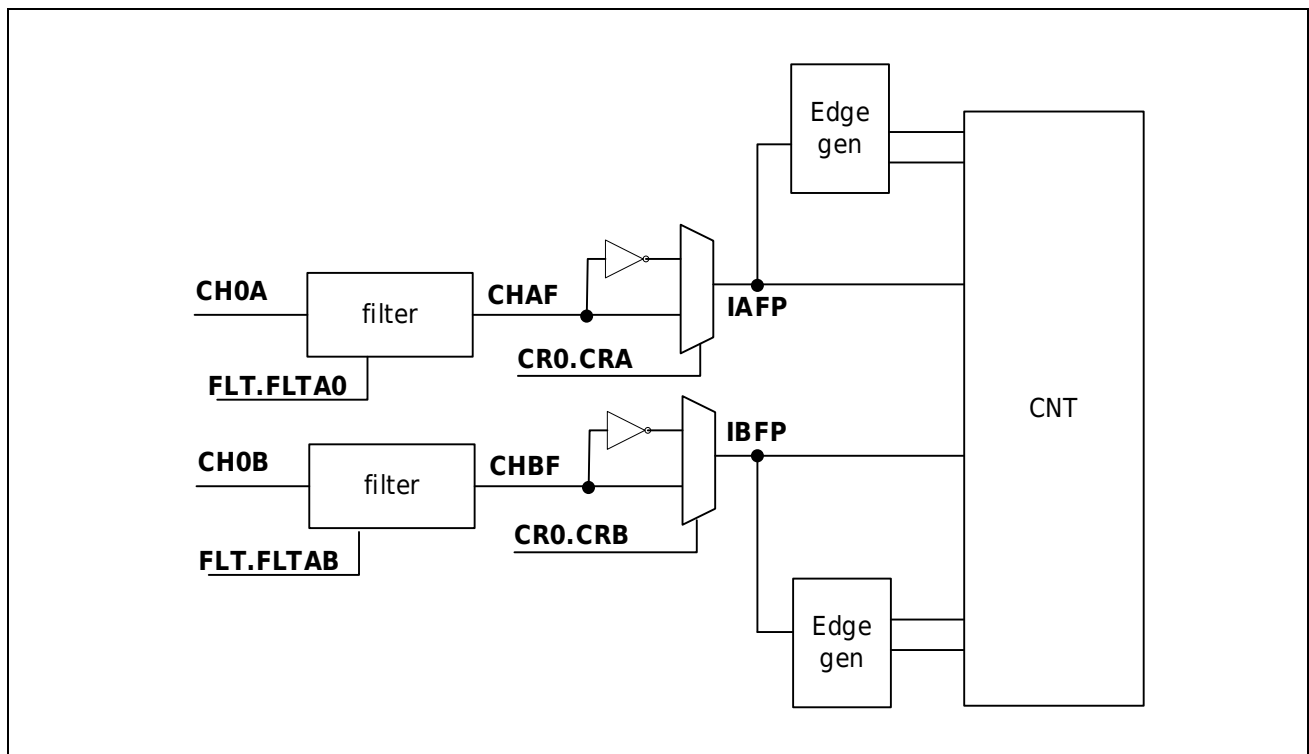


图 14-56 编码计数功能框图

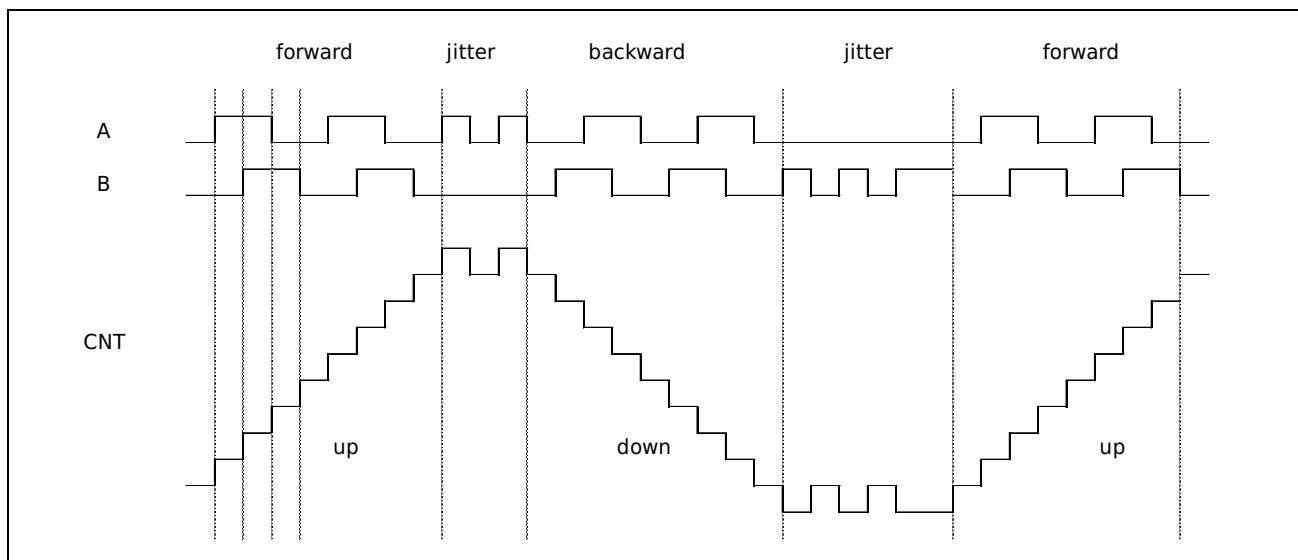


图 14-57 编码计数

14.3.9 Timer 触发 ADC

CCR0A,CCR1A,CCR2A 比较匹配可以配置触发 ADC,中心对齐 PWM 时只能通过控制寄存器 M23CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCR0B,CCR1B,CCR2B,CCR3 比较匹配可以配置触发 ADC,中心对齐 PWM 时,可以通过寄存器 CRCHx.CISB 分别控制三个匹配触发点(上升,下降,上升下降)。

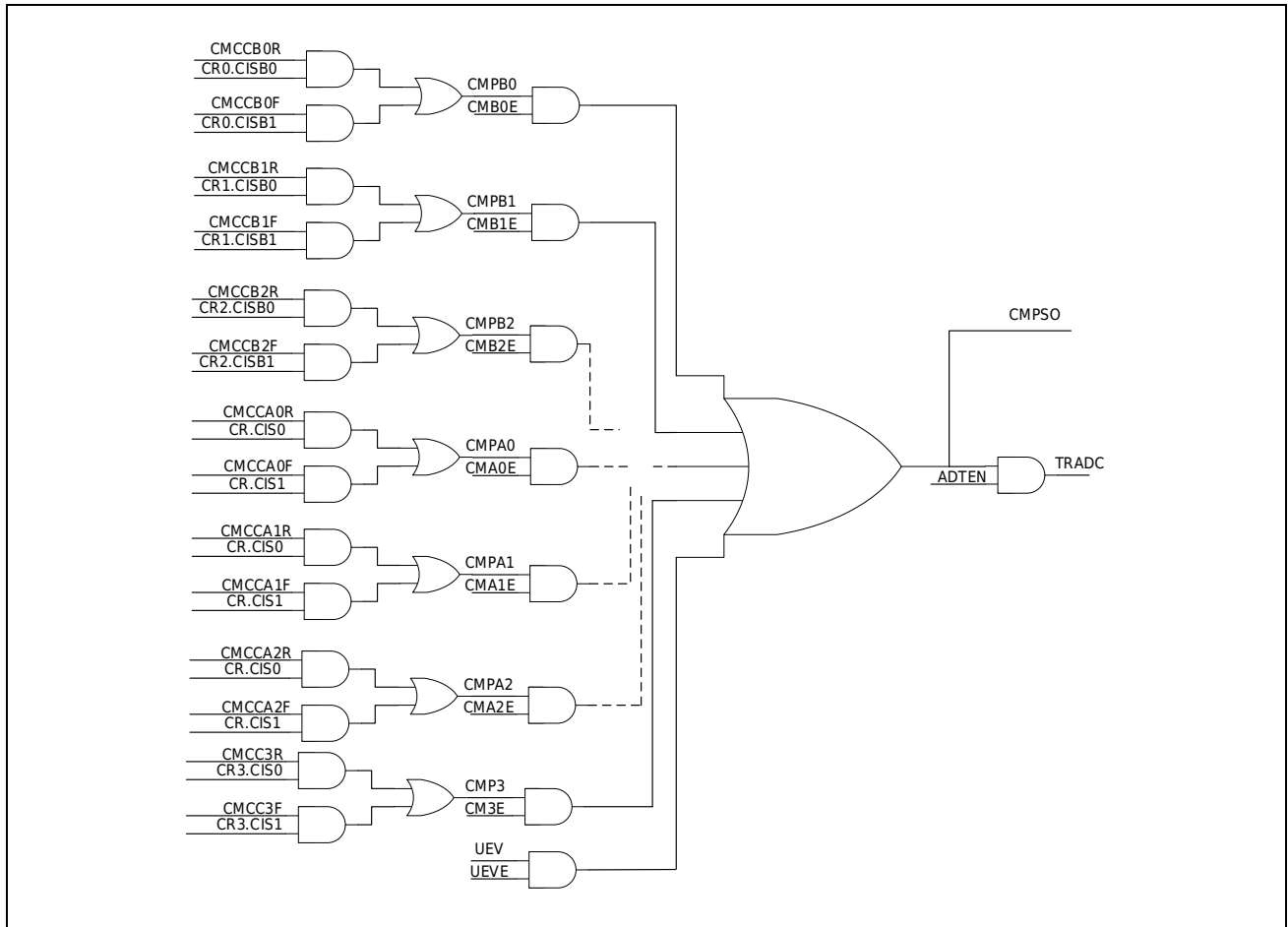


图 14-58 ADC 触发

互补 PWM 输出时 B 比较作为 ADC 触发功能

1. 设置为 PWM 互补输出模式,参考“中心对齐互补 PWM 输出设置”
2. 设置 CCRxB 设置 ADC 触发比较点
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)
4. 选择 ADC 触发比较的源 ADTR
5. 使能 ADC 触发 ADTR.ADTE

14.3.10 刹车控制

外部 BK 端口输入可控制刹车,系统 fail 可以控制刹车功能。通过 CR.BG 可以实现软件刹车功能,控制输出端口到设定的状态。

14.3.11 中断

表 14-8 ATIM 中断源与中断请求列表

中断	说明
ATIM0_UEV	ATIM0 事件更新
ATIM0_CHA	ATIM0 通道 A 比较捕获中断
ATIM0_CHB	ATIM0 通道 B 比较捕获中断
ATIM0_BKETR	ATIM0 刹车/触发中断
ATIM3_UEV	ATIM3 事件更新
ATIM3_CHYA	ATIM3 通道 A 比较捕获中断
ATIM3_CHYB	ATIM3 通道 B 比较捕获中断
ATIM3_BK_ETR_OV_UD_CMP3	ATIM3 刹车/触发中断/通道 3 比较/上溢/下溢

14.3.12 定时器互联

TRGO 输出信号可以连接到其它定时器的 ITR 信号。连接关系如下：

表 14-9 定时器互联信号列表

	ITR0	ITR1	ITR2	ITR3
ATIM0	GTIM0 溢出	GTIM1 溢出	GTIM2 溢出	ATIM3 TRGO
ATIM3	GTIM0 溢出	GTIM1 溢出	GTIM2 溢出	ATIM0 TRGO

14.3.13 输入互联

ATIM0 的 ETR 输入可以从端口直接输入，也可通过端口功能寄存器 SYSCTRL->IOAUX1 选择可以连通到其它模块或端口。

当 ATIM0ETR=0x0 时，捕获输入是 AFRH/AFRL 选择的端口输入，当 ATIM0ETR =0x1~0x7 时，也可通过端口功能寄存器 SYSCTRL->IOAUX1 选择，连接其它模块的输入或输出。

表 14-10 ATIM0 ETR 信号来源列表

SYSCTRL->IOAUX1	0	1	2	3	4	5	6	7
ATIM0ETR [2:0]	ATIM0_ETR	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA04	PA12	PB04

ATIM0 的 CHA 的输入信号可以从 AFRH/AFRL 选择的端口输入，也可通过端口功能寄存器 SYSCTRL->IOAUX2 选择可以连通到其它模块或端口。

当 CAP0=0x0 时，捕获输入是 AFRH/AFRL 选择的端口输入，当 CAP0=0x1~0x7 时，连接其它模块的输入或输出。

表 14-11 ATIM0 CHA 输入信号来源列表

SYSCTRL->IOAUX2	0	1	2	3	4	5	6	7
ATIM0CAP0[2:0]	ATIM0_CHA	USART0_RXD	VC0_OUT	LVD_OUT	VC1_OUT	PA00	PA08	PB00

14.4 寄存器描述

14.4.1 寄存器列表

表 14-12 ATIM 基地址列表

名称	基地址	描述
ATIM0	0x4000 6000	ATIM0基地址
ATIM3	0x4000 6C00	ATIM3基地址

表 14-13 ATIM 寄存器列表

寄存器名	偏移地址	复位值
自动重载寄存器 (ATIMx_ARR)	0x000	0x0000 0000
16位模式计数寄存器 (ATIMx_CNT)	0x004	0x0000 0000
32位模式计数寄存器 (ATIMx_CNT32)	0x008	0x0000 0000
模式0控制寄存器 (按不同模式进行描述) (ATIMx_M0CR)	0x00C	0x0060 0008
模式1控制寄存器 (按不同模式进行描述) (ATIMx_M1CR)	0x00C	0x0060 0008
模式23控制寄存器 (按不同模式进行描述) (ATIMx_M23CR)	0x00C	0x0060 0008
中断标志寄存器 (ATIMx_IFR)	0x010	0x0000 0000
中断清除寄存器 (ATIMx_ICLR)	0x014	0x0000 FFFF
主从模式控制寄存器 (ATIMx_MSCR)	0x018	0x0000 0000
滤波控制寄存器 (ATIMx_FLTR)	0x01C	0x0000 0000
ADC触发控制寄存器 (ATIMx_ADTR)	0x020	0x0000 0000
比较单元0控制寄存器 (ATIMx_CR0/ATIMx_CRCH0)	0x024	0x0000 3000
比较单元1控制寄存器 (ATIM3_CRCH1)	0x028	0x0000 3000
比较单元2控制寄存器 (ATIM3_CRCH2)	0x02C	0x0000 3000
死区寄存器 (ATIMx_DTR)	0x030	0x0000 0000
重复计数寄存器 (ATIMx_RCR)	0x034	0x0000 0000
Timer重载寄存器/周期映射地址 (ATIMx_ARRDM)	0x038	0x0000 0000
比较0A寄存器 (ATIMx_CCR0A)	0x03C	0x0000 0000
比较0B寄存器 (ATIMx_CCR0B)	0x040	0x0000 0000
比较1A寄存器 (ATIM3_CCR1A)	0x044	0x0000 0000
比较1B寄存器 (ATIM3_CCR1B)	0x048	0x0000 0000
比较2A寄存器 (ATIM3_CCR2A)	0x04C	0x0000 0000
比较2B寄存器 (ATIM3_CCR2B)	0x050	0x0000 0000
比较3A寄存器 (ATIM3_CCR3)	0x054	0x0000 0000
比较单元3A控制寄存器 (ATIM3_CRCH3)	0x058	0x0000 0000

(x=0, 3)

14.4.2 模式 0 寄存器描述

14.4.2.1 自动重载寄存器 (ATIMx_ARR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	ARR	16 位重载定时器 重载值寄存器	-	RW

14.4.2.2 16 位模式计数寄存器 (ATIMx_CNT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CNT	16 位重载定时器 计数值寄存器	-	RW

14.4.2.3 32 位模式计数寄存器 (ATIMx_CNT32)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT32[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT32[15:0]															

位	标记	位名	功能	读写
b31~b0	CNT32	32 位定时器 计数值寄存器	注: 仅在模式 0 32 位定时器自由计数模式下有效, 其它模式禁止写该寄存器	RW

14.4.2.4 控制寄存器 (ATIMx_MOCR)

复位值: 0x0060 0008

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	MODE		Res	UI	GATE P	GATE	Res	PRS				TOGEN	CT	MD	CEN

位	标记	位名	功能	读写
b31~b14	Res	保留位	保留, 必须保持复位值	
b13~b12	MODE	工作模式	00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11 三角波模式	RW
b11	Res	保留位	保留, 必须保持复位值	
b10	UI	溢出中断使能控制	0: 禁止溢出中断 1: 使能溢出中断	RW
b9	GATEP	端口 GATE 极性控制	0: 端口 GATE 高电平有效 1: 端口 GATE 低电平有效	RW
b8	GATE	定时器门控使能	0: 无门控, CTEN=1 时定时器工作 1: 门控使能, 端口 GATE 有效并且CTEN=1时定时器才工作	RW
b7	Res	保留位	保留, 必须保持复位值	
b6~b4	PRS	内部时钟分频选择	000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;	RW
b3	TOGEN	模式0下翻转输出使能	0: 翻转输出关闭, CHA,CHB 输出为低电平 1: 翻转输出使能	RW
b2	CT	计数时钟选择	0: 内部计数时钟 TCLK 1: 外部计数时钟 ETR	RW
b1	MD	模式选择32定时/16定时模式选择	0: 32位自由计数 1: 16位重载计数	RW
b0	CEN	定时器使能	0: 定时器停止; 1: 定时器使能	RW

14.4.2.5 中断标志寄存器 (ATIMx_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														UI	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	UI	溢出中断标志	0: 计数器未溢出 1: 计数器已溢出	RO

14.4.2.6 中断标志清除寄存器 (ATIMx_ICLR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														UI	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	UI	溢出中断清除	0: 清除溢出标志 1: 无功能	R1W0

14.4.2.7 死区时间寄存器 (ATIMx_DTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			MO	Res											

位	标记	位名	功能	读写
b31~b13	Res	保留位	保留, 必须保持复位值	
b12	MO	翻转输出使能	0: 翻转端口为输入状态 1: 翻转端口输出使能	RW
b11~b0	Res	保留位	保留, 必须保持复位值	

14.4.3 模式 1 寄存器描述

14.4.3.1 16 位模式计数寄存器（ATIMx_CNT）

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
Res																	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
CNT																	
位		标记		位名		功能						读写					
b31~b16		Res		保留位		保留，必须保持复位值											
b15~b0		CNT		计数值寄存器		-						RW					

14.4.3.2 控制寄存器 (ATIMx_M1CR)

复位值: 0x0060 0008

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	Ones hot	Mode		Res	UI	Edg2nd	Edg1st	Res	PRS			Res	CT	Res	CEN

位	标记	位名	功能	读写																				
b31~b15	Res	保留位	保留, 必须保持复位值																					
b14	Oneshot	单次触发模式选择	0: 循环测量 1: 完成一次脉冲测量自动结束, 再次测量需要重新使能CTEN	RW																				
b13~b12	MODE	工作模式	00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11 三角波模式	RW																				
b11	Res	保留位	保留, 必须保持复位值																					
b10	UI	溢出中断使能控制	0: 禁止溢出中断 1: 使能溢出中断 注: 计数到 0xFFFF 才会溢出并产生溢出标志	RW																				
b9	Edg2nd	脉冲宽度测量结束边沿选择																						
b8	Edg1st	脉冲宽度测量开始边沿选择	<table border="1"> <tr> <td>edg2nd</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>Edg1st</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr> <td>测量</td><td>上沿-上沿</td><td>低电平宽度</td><td>高电平宽度</td><td>下沿-下沿</td></tr> <tr> <td></td><td>周期</td><td></td><td></td><td>周期</td></tr> </table>	edg2nd	0	0	1	1	Edg1st	0	1	0	1	测量	上沿-上沿	低电平宽度	高电平宽度	下沿-下沿		周期			周期	RW
edg2nd	0	0	1	1																				
Edg1st	0	1	0	1																				
测量	上沿-上沿	低电平宽度	高电平宽度	下沿-下沿																				
	周期			周期																				
b7	Res	保留位	保留, 必须保持复位值																					
b6~b4	PRS	内部时钟分频选择	000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;	RW																				
b3	Res	保留位	保留, 必须保持复位值																					
b2	CT	计数时钟选择	0: 内部计数时钟TCLK 1: 外部计数时钟 ETR;	RW																				
b1	Res	保留位	保留, 必须保持复位值																					
b0	CEN	脉冲宽度测量使能	0: 禁止; 1: 使能	RW																				

14.4.3.3 中断标志寄存器 (ATIMx_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													PWC	Res.	UI
位	标记	位名	功能										读写		
b31~b3	Res	保留位	保留, 必须保持复位值												
b2	PWC	脉冲宽度测量中断标志	0: 脉冲宽度测量未完成 1: 脉冲宽度测量已完成										RO		
b1	Res	保留位	保留, 必须保持复位值												
b0	UI	溢出中断标志	0: 计数器未溢出 1: 计数器已溢出										RO		

14.4.3.4 中断标志清除寄存器 (ATIMx_ICLR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													PWC	Res.	UI
位	标记	位名	功能										读写		
b31~b3	Res	保留位	保留, 必须保持复位值												
b2	PWC	脉冲宽度测量中断标志清除	0: 清除脉冲宽度测量中断标志 1: 无功能										R1W0		
b1	Res	保留位	保留, 必须保持复位值												
b0	UI	溢出中断清除	0: 清除溢出标志 1: 无功能										R1W0		

14.4.3.5 主从模式控制寄存器 (ATIMx_MSCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		IB0S	IA0S	Res			TS			Res					

位	标记	位名	功能	读写
b31~b13	Res	保留位	保留, 必须保持复位值	
b12	IB0S	CH0B 输入选择	0: CH0B; 1: 内部触发 TS 选择信号;	RW
b11	IA0S	IA0 输入选择	0: CH0A; 1: CH0A CH1A CH2A XOR(TIM3) 注: 设置为 1 后端口的任意一个端口变化将导致输入变化	RW
b10~b8	Res	保留位	保留, 必须保持复位值	
b7~b5	TS	触发选择	000: 端口ETR的滤波相位选择后的信号ETFP ; 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 无效; 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口 CH0B 的滤波相位选择后的信号 IBFP;	RW
b4~b0	Res	保留位	保留, 必须保持复位值	

14.4.3.6 输出控制滤波 (ATIMx_FLTR)

复位值: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ETP	FLTET			Res											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									FLTBO			Res	FLTA0		

位	标记	位名	功能	读写
b31	ETP	ETR 输入相位选择	0: 同相位; 1: 反向输入;	RW
b30~b28	FLTET	ETR滤波控制	0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效	RW
b27~b7	Res	保留位	保留, 必须保持复位值	
b6~b4	FLTBO	CHB 输入滤波控制	0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效 注: 需要配置 CRCH0.CSB 为 1	RW
b3	Res	保留位	保留, 必须保持复位值	
b2~b0	FLTA0	CHA 输入滤波控制	0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效 注: 需要配置 CR0.CSA 为 1	RW

14.4.3.7 控制寄存器 (ATIMx_CR0)

复位值: 0x0000 3000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							CIEA	Res		CSB	CSA	Res			

位	标记	位名	功能	读写
b31~b9	Res	保留位	保留, 必须保持复位值	
b8	CIEA	脉冲宽度测量完成中断使能	0: 禁止 1: 使能	RW
b7~b6	Res	保留位	保留, 必须保持复位值	
b5	CSB	B通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b4	CSA	A通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b3~b0	Res	保留位	保留, 必须保持复位值	

14.4.3.8 比较捕获寄存器 (ATIMx_CCR0A)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCR0A															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CCR0A	脉冲宽度测量结果		RO

14.4.4 模式 2,3 寄存器描述

14.4.4.1 自动重载寄存器 (ATIMx_ARR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	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

14.4.4.2 16 位模式计数寄存器 (ATIMx_CNT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	CNT	重载定时器 计数值寄存器	-	RW

注: 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。

14.4.4.3 控制寄存器 (ATIMx_M23CR)

复位值: 0x0060 0008

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	UND	OV	DIR	BG	UG	TG	OCC	CIS	BI	TI	TD	URS	OCCS		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CSG	Ones hot	Mode	UD	UI	CFG	CRG	BUFP	PRS	Pwm 2s	CT	COMP	CTEN			

位	标记	位名	功能	读写
b31~b30	Res	保留位	保留, 必须保持复位值	
b29	UND	下溢出中断使能	0: 中断禁止 1: 中断使能	RW
b28	OV	上溢出中断使能	0: 中断禁止 1: 中断使能	RW
b27	DIR	计数方向	0: 上计数 1: 下计数 注: 只有在锯齿波模式下可以写。其它模式下只读, 写禁止。 从其它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。	RW/RO
b26	BG	软件刹车, 自动清零	0: 写0无效 1: 写1产生软件刹车	W1
b25	UG	软件更新, 自动清零	0: 写0无效 1: 写1产生软件更新 注: 初始化计数器并更新缓存寄存器到相应寄存器 (缓存使能), 预除频计数器也会被清零。	W1
b24	TG	软件触发, 自动清零	0: 写 0 无效 1: 写1产生软件触发 注: 需要在触发模式SMS=2且mode=2/3下都可以触发	W1
b23	OCC	OCREF 清除使能	0: OCREF 输出不受 OCREF_CLR 影响 1: OCREF_CLR信号可以清除OCREF输出	RW
b22~b21	CIS	通道A比较中断模式	00: 无中断, 01: 上沿中断, 10: 下沿中断, 11: 上下沿都中断 注: 通道 B 比较中断单独控制在 CRCHx 寄存器中 CISB	RW
b20	BI	刹车中断使能	0: 中断禁止 1: 中断使能	RW
b19	TI	触发中断使能	0: 中断禁止 1: 中断使能	RW
b18	TD	触发DMA使能	0: 触发DMA禁止 1: 触发DMA使能	RW
b17	URS	更新源	0: 上溢出/下溢出/软件更新UG/从模式复位; 1: 上溢出/下溢出	RW
b16	OCCS	OCREF 清除源选择	0: 电压比较器 VC 输出, VC 选择在 VCx_OUTCFG 寄存器设置 1: ETR 端口滤波相位选择后的信号 当 OCCE 有效时, OC_clr 可以清除 OCREF 的比较输出信号为零,	RW

(当 OCMx>1 时有效)，下一个 uev 事件后继续比较输出				
b15	CSG	GATE 在 PWM 互补模式下捕获 /比较选择	0: 比较 1: 捕获 注：使用 CCR0B 作为 GATE 的比较或捕获通道，只有在 PWM 互补输出时有效。	RW
b14	Oneshot	单次触发模式选择	0: 循环计数 1: 发生事件更新后定时器停止。	RW
b13~b12	MODE	工作模式	00 定时器模式0; 01 PWC模式 10 锯齿波模式; 11 三角波模式	RW
b11	UD	更新触发 DMA 使能	0: 禁止更新触发 DMA 1: 使能更新触发 DMA	RW
b10	UI	更新中断使能	0: 禁止更新中断 1: 使能更新中断 注：计数到 0xFFFF 才会溢出并产生溢出标志	RW
b9	CFG	GATE 作为捕获输入时，下沿捕获有效控制	0: 下沿捕获无效 1: 下沿捕获有效 注：只有在 PWM 互补输出时有效	RW
b8	CRG	GATE 作为捕获输入时，上沿捕获有效控制	0: 上沿捕获无效 1: 上沿捕获有效 注：只有在 PWM 互补输出时有效	RW
b7	BUFP	自动重载ARR缓存使能	0: 周期缓存无效，写入后立刻影响周期值 1: 周期缓存使能，写入后下个事件更新后才影响到周期值。	RW
b6~b4	PRS	内部时钟分频选择	000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;	RW
b3	PWM2S	OCREFA双点比较选择(缺省值为1)	0: 双点比较使能，使用CCRA,CCRB比较控制OCREFA输出 1: 单点比较使能，只使用CCRA比较控制OCREFA输出 注：OCREFB 不受影响，仍然使用 CCRB 控制 OCREFB 输出	RW
b2	CT	计数时钟选择	0: 内部计数时钟TCLK 1: 外部计数时钟 ETR;	RW
b1	COMP	PWM互补输出模式选择	0: 独立PWM输出 1: 互补PWM输出	RW
b0	CTEN	定时器使能	0: 禁止； 1: 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

14.4.4.4 中断标志寄存器 (ATIMx_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													CA3	Und	OV
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TI	BI	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res	UIF

位	标记	位名	功能	读写
b31~b19	Res	保留位	保留, 必须保持复位值	RO
b18	CA3	通道CH3比较匹配标志	0: 没有发生 1: 发生	RO
b17	UND	下溢出中断标志	0: 没有发生 1: 发生	RO
b16	OV	上溢出中断标志	0: 没有发生 1: 发生	RO
b15	TI	触发中断标志	0: 没有发生 1: 发生	RO
b14	BI	刹车中断标志	0: 没有发生 1: 发生	RO
b13	CB2E	通道CH2B捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b12	CB1E	通道CH1B捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b11	CB0E	通道CH0B捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b10	CA2E	通道CH2A捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b9	CA1E	通道CH1A捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b8	CA0E	通道CH0A捕获数据丢失标志	0: 无数据丢失 1: 数据丢失	RO
b7	CB2F	通道CH2B发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b6	CB1F	通道CH1B发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b5	CB0F	通道CH0B发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b4	CA2F	通道CH2A发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b3	CA1F	通道CH1A发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b2	CA0F	通道CH0A发生捕获/比较匹配标志	0: 没有发生 1: 发生	RO
b1	Res	保留位	保留, 必须保持复位值	

b0	UIF	事件更新中断标志	0: 没有发生 1: 发生	RO
----	-----	----------	------------------	----

14.4.4.5 中断标志清除寄存器 (ATIMx_ICLR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													CA3	Und	OV
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TI	BI	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF

位	标记	位名	功能	读写
b31~b19	Res	保留位	保留, 必须保持复位值	R1W0
b18	CA3	通道CH3比较匹配标志清除	0: 清除通道CH3A比较匹配标志 1: 无功能	R1W0
b17	UND	下溢出中断标志清除	0: 清除下溢出中断标志 1: 无功能	R1W0
b16	OV	上溢出中断标志清除	0: 清除上溢出中断标志 1: 无功能	R1W0
b15	TI	触发中断标志清除	0: 清除触发中断标志 1: 无功能	R1W0
b14	BI	刹车中断标志清除	0: 清除刹车中断标志 1: 无功能	R1W0
b13	CB2E	通道CH2B捕获数据丢失标志清除	0: 清除通道CH2B捕获数据丢失标志 1: 无功能	R1W0
b12	CB1E	通道CH1B捕获数据丢失标志清除	0: 清除通道CH1B捕获数据丢失标志 1: 无功能	R1W0
b11	CB0E	通道CH0B捕获数据丢失标志清除	0: 清除通道CH0B捕获数据丢失标志 1: 无功能	R1W0
b10	CA2E	通道CH2A捕获数据丢失标志清除	0: 清除通道CH2A捕获数据丢失标志 1: 无功能	R1W0
b9	CA1E	通道CH1A捕获数据丢失标志清除	0: 清除通道CH1A捕获数据丢失标志 1: 无功能	R1W0
b8	CA0E	通道CH0A捕获数据丢失标志清除	0: 清除通道CH0A捕获数据丢失标志 1: 无功能	R1W0
b7	CB2F	通道CH2B捕获/比较匹配标志清除	0: 清除通道CH2B捕获/比较匹配标志 1: 无功能	R1W0
b6	CB1F	通道CH1B捕获/比较匹配标志清除	0: 清除通道CH1B捕获/比较匹配标志 1: 无功能	R1W0
b5	CB0F	通道CH0B捕获/比较匹配标志清除	0: 清除通道CH0B捕获/比较匹配标志 1: 无功能	R1W0
b4	CA2F	通道CH2A捕获/比较匹配标志清除	0: 清除通道CH2A捕获/比较匹配标志 1: 无功能	R1W0
b3	CA1F	通道CH1A捕获/比较匹配标志清除	0: 清除通道CH1A捕获/比较匹配标志 1: 无功能	R1W0
b2	CA0F	通道CH0A捕获/比较匹配标志清除	0: 清除通道CH0A捕获/比较匹配标志 1: 无功能	R1W0
b1	Res	保留位	保留, 必须保持复位值	

b0	UIF	事件更新中断清除	0: 清除事件更新中断标志 1: 无功能	R1W0
----	-----	----------	-------------------------	------

14.4.4.6 主从模式控制寄存器 (ATIMx_MSCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			IB0S	IA0S	SMS			TS			MSM	CCDS	MMS		

位	标记	位名	功能	读写
b31~b13	Res	保留位	保留, 必须保持复位值	RW
b12	IB0S	CH0B 输入选择	0: CH0B; 1: 内部触发 TS 选择信号;	RW
b11	IA0S	IA0 输入选择	0: CH0A; 1: CH0A CH1A CH2A XOR(ATIM3) 0: CHA; 1: CHA ETR GATE XOR(ATIM0) 注: 设置为 1 后端口的任意一个端口变化将导致输入变化	RW
b10~b8	SMS	从模式功能选择	000: 使用内部时钟; 001: 复位功能; 010: 触发模式; 011: 外部时钟模式 100: 正交编码计数模式1; 101: 正交编码计数模式2; 110: 正交编码计数模式3; 111: 门控功能	RW
b7~b5	TS	触发选择	000: 端口ETR的滤波相位选择后的信号ETFP; 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口 CH0B 的滤波相位选择后的信号 IBFP;	RW
b4	MSM	主从选择	0: 无延时 1: 延时使能, 使主送计数器同时启动. 注: 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
b3	CCDS	比较模式下 DMA 比较触发选择	0: 比较匹配触发DMA; 1: 比较匹配不触发DMA, 事件更新代替比较匹配触发DMA	RW
b2~b0	MMS	主模式输出选择	用于内部互联, 连接到其它定时器的ITRx 000: 软件更新UG, 写CR.UG; 001: 定时器使能CTEN; 010: 定时器事件更新UEV; 011: 比较匹配选择输出CMPSO; 100: 定时器比较参数输出OCREF0A 101: ATIM0: 定时器比较参数输出OCREF0B ATIM3: 定时器比较参数输出OCREF1A	RW

110:

ATIM0: 定时器比较参数输出 OCREF0B

ATIM3: 定时器比较参数输出 OCREF2A

111: 定时器比较参数输出 OCREF0B

14.4.4.7 输出控制/输入滤波 (ATIMx_FLTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ETP	FLTET			BKP	FLTBK			CCPB2	OCMB2			CCPA2	OCMA2		
-	-			-	-			-	FLT2			-	FLTA2		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCPB1	OCMB1			CCPA1	OCMA1			CCPB0	OCMB0			CCPA0	OCMA0		
-	FLT1			-	FLTA1			CCPB0	FLT0			CCPA0	FLTA0		

位	标记	位名	功能	读写
b31	ETP	ETR 输入相位选择	0: 同相位; 1: 反向输入;	RW
b30~b28	FLTET	ETR滤波控制	0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效	RW
b27	BKP	刹车BK输入相位选择	0: 同相位输入 1: 反向输入;	RW
b26~b24	FLTBK	刹车输入滤波控制	滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效	RW
b23	CCPB2	CH2B通道比较输出极性控制	0: 正常输出; 1: 反向输出	RW
b22~b20	OCMB2 FLT2	CH2B通道比较控制 / CH2B通道输入滤波设置	比较功能: CH2B通道比较控制;参考OCMB0 捕获功能: CH2B 通道输入滤波设置, 参考 FLT2BK	RW
b19	CCPA2	CH2A通道比较输出极性控制	0: 正常输出; 1: 反向输出	RW
b18~b16	OCMA2 FLTA2	CH2A通道比较控制捕获功能 / CH2A通道输入 滤波设置	比较功能: CH2A通道比较控制;参考OCMB0 捕获功能: CH2A 通道输入滤波设置, 参考 FLT2BK	RW
b15	CCPB1	CH1B通道比较输出极性控制	0: 正常输出; 1: 反向输出	RW
b14~b12	OCMB1 FLT1	CH1B通道比较控制 / CH1B通道输入滤波设置	比较功能: CH1B通道比较控制;参考OCMB0 捕获功能: CH1B 通道输入滤波设置, 参考 FLT1BK	RW
b11	CCPA1	CH1A通道比较输出极性控制	0: 正常输出; 1: 反向输出	RW
b10~b8	OCMA1 FLTA1	CH1A通道比较控制捕获功能 / CH1A通道输入 滤波设置	比较功能: CH1A通道比较控制;参考OCMB0 捕获功能: CH1A 通道输入滤波设置, 参考 FLT1BK	RW
b7	CCPB0	CH0B端口比较输出极性控制 / CH0B端口输入相位与极性控制	比较功能: 输出比较模式 CH0B端口比较输出极性控制 0: 正常输出; 1: 反向输出	RW

			编码计数与从模式门控功能：输入相位控制 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 0：正常输入； 1：反向输入	
			比较功能：CHxB通道比较控制： 000：强制为0 001：强制为1 010：比较匹配时强制为0 011：比较匹配时强制为1 100：比较匹配时翻转 101：比较匹配时输出一个计数周期的高电平 110：PWM 模式1 单点比较： 上计数时CNT<CCRxy输出高，下计数时CNT>CCRxy输出为低电平 双点比较： 1) 锯齿波上计数 CCRxA<CNT≤CCRxB输出为低电平 2) 锯齿波下计数 CCRxA<CNT≤CCRxB输出为高电平 3) 三角波上计数CNT<CCRxA输出高，下计数CNT>CCRxB为低电平 111 PWM 模式2 单点比较： 上计数时CNT<CCRxy输出低，下计数时CNT>CCRxy输出为高电平 双点比较： 1) 锯齿波上计数 CCRxA≤CNT<CCRxB输出为高电平 2) 锯齿波下计数 CCRxA≤CNT<CCRxB输出为低电平 3) 三角波上计数CNT<CCRxA输出低，下计数CNT>CCRxB为高电平 捕获功能：CH0B输入通道滤波设置，参考FLTBK	
b6~b4	OCMB0 FLTB0	CH0B通道比较控制 / CH0B通道输入滤波设置	1) 锯齿波上计数 CCRxA<CNT≤CCRxB输出为低电平 2) 锯齿波下计数 CCRxA<CNT≤CCRxB输出为高电平 3) 三角波上计数CNT<CCRxA输出高，下计数CNT>CCRxB为低电平 111 PWM 模式2 单点比较： 上计数时CNT<CCRxy输出低，下计数时CNT>CCRxy输出为高电平 双点比较： 1) 锯齿波上计数 CCRxA≤CNT<CCRxB输出为高电平 2) 锯齿波下计数 CCRxA≤CNT<CCRxB输出为低电平 3) 三角波上计数CNT<CCRxA输出低，下计数CNT>CCRxB为高电平 捕获功能：CH0B输入通道滤波设置，参考FLTBK	RW
b3	CCPA0	CH0A端口比较输出极性控制/ CH0B输入相位与极性控制	比较功能：CH0A端口比较输出极性控制： 0：正常输出； 1：反向输出 编码计数与从模式门控功能：输入相位控制 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： 0：正常输入； 1：反向输入	RW
b2~b0	OCMA0/ FLTA0	CH0A 通道比较控制/ CH0A 通道输入滤波设置	比较功能：A 通道比较控制；参考 OCMB0 捕获功能：CH0A 输入通道滤波设置，参考 FLTBK	RW

14.4.4.8 ADC 触发控制寄存器 (ATIMx_ADTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ADT	CMB2	CMB1	CMB0	CMA2	CMA1	CMA0	UEV

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留, 必须保持复位值	
b7	ADT	使能ADC触发全局控制	0: 禁止 1: 使能	RW
b6	CMB2	通道CH2B比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b5	CMB1	通道CH1B比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b4	CMB0	通道CH0B比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b3	CMA2	通道CH2A比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b2	CMA1	通道CH1A比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b1	CMA0	通道CH0A比较匹配触发ADC使能	0: 禁止 1: 使能	RW
b0	UEV	事件更新触发ADC使能	0: 禁止 1: 使能	RW

14.4.4.9 通道 0 控制寄存器 (ATIMx_CRCH0)

复位值: 0x0000 3000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa		

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15	CCGB	捕获比较B软件触发, 硬件自动清零	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。	RW
b14	CCGA	捕获比较A软件触发, 硬件自动清零	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。	RW
b13~b12	CISB	B通道比较匹配设置	00 无匹配 01 上升匹配 10 下降匹配 11 双匹配	RW
b11	CDEB	B捕获比较触发DMA使能	0: 禁止 1: 使能	RW
b10	CDEA	A捕获比较触发DMA使能	0: 禁止 1: 使能	RW
b9	CIEB	B捕获比较触发中断使能	0: 禁止 1: 使能	RW
b8	CIEA	A捕获比较触发中断使能	0: 禁止 1: 使能	RW
b7	BUFEB	比较功能: B比较缓存使能控制	0: 禁止 1: 使能	RW
b6	BUFEA	比较功能: A比较缓存使能控制	0: 禁止 1: 使能	RW
b5	CSB	B通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b4	CSA	A通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b3	CFB	B通道下降沿捕获使能	0: 禁止 1: 使能	RW
b2	CRB	B通道上升沿捕获使能	0: 禁止 1: 使能	RW
b3~b2	BKSB	B通道比较功能输出刹车电平控制	00: 高阻输出 01: 保持之前输出 10: 强制输出低电平 11: 强制输出高电平; 注: 无debug_stop模式	RW
b1	CFA	A通道下降沿捕获使能	0: 禁止	RW

			1: 使能	
b0	CRA	A通道上升沿捕获使能	0: 禁止 1: 使能	RW
b1~b0	BKSA	A通道比较功能输出刹车电平控制	00: 高阻输出 01: 保持之前输出 10: 强制输出低电平 11: 强制输出高电平 注: (debug stop 模式下, A 和 B 通道的刹车电平控制都由 BKSA 来控制)	RW

14.4.4.10 通道 1/2 控制寄存器 (ATIM3_CRCH1/2)

复位值: 0x0000 3000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa		

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15	CCGB	捕获比较B软件触发, 硬件自动清零	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。	RW
b14	CCGA	捕获比较A软件触发, 硬件自动清零	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。	RW
b13~b12	CISB	B通道比较匹配设置	00 无匹配 01 上升匹配 10 下降匹配 11 双匹配	RW
b11	CDEB	保留位	保留, 必须保持复位值	
b10	CDEA	保留位	保留, 必须保持复位值	
b9	CIEB	B捕获比较触发中断使能	0: 禁止 1: 使能	RW
b8	CIEA	A捕获比较触发中断使能	0: 禁止 1: 使能	RW
b7	BUFEB	比较功能: B比较缓存使能控制	0: 禁止 1: 使能	RW
b6	BUFEA	比较功能: A比较缓存使能控制	0: 禁止 1: 使能	RW
b5	CSB	B通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b4	CSA	A通道捕获/比较功能选择	0: 比较模式 1: 捕获模式	RW
b3	CFB	B通道下降沿捕获使能	0: 禁止 1: 使能	RW
b2	CRB	B通道上升沿捕获使能	0: 禁止 1: 使能	RW
b3~b2	BKSB	B通道比较功能输出刹车电平控制	00: 高阻输出 01: 保持之前输出 10: 强制输出低电平 11: 强制输出高电平; 注: 无debug_stop模式	RW
b1	CFA	A通道下降沿捕获使能	0: 禁止 1: 使能	RW
b0	CRA	A通道上升沿捕获使能	0: 禁止	RW

			1: 使能	
			00: 高阻输出	
			01: 保持之前输出	
b1~b0	BKSA	A通道比较功能输出刹车电平控制	10: 强制输出低电平	RW
			11: 强制输出高电平	
			注: (debug stop 模式下, A 和 B 通道的刹车电平控制都由 BKSA 来控制)	

14.4.4.11 死区时间寄存器 (ATIMx_DTR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res.	VCE	Safe	MO	AO	BK	DT	Bksel	DTR							

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14	VCE	VC 刹车使能	0: 禁止 1: 使能	RW
b13	Safeen	Safety 刹车使能 (osc fail,brown down,lockup)	0: 禁止 1: 使能	RW
b12	MO	PWM输出使能	0: 禁止 1: 使能	RW
b11	AO	PWM输出自动使能	0: 禁止 1: 使能	RW
b10	BK	端口刹车使能	0: 禁止 1: 使能	RW
b9	DT	死区控制使能	0: 禁止 1: 使能	RW
b8	bksel	保留位	保留, 必须保持复位值	RW
b7~b0	DTR	死区时间寄存器	DTR[7] =0 T=DTR[6:0]+2 DTR[7:6] =10 T={DTR[5:0]+64}*2 +2 DTR[7:5] =110 T={DTR[4:0]+32}*8 +2 DTR[7:5] =111 T={DTR[4:0]+32}*16 +2	2-129 step=1 130-256 step=2 258-506 step=8 514-1010 step=16

14.4.4.12 重复计数寄存器 (ATIMx_RCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						UD	OV	RCR							
位	标记	位名	功能										读写		
b31~b10	Res	保留位	保留, 必须保持复位值												
b9	UD	重复计数控制下溢出屏蔽	0: 下计数溢出使能 1: 下计数溢出屏蔽										RW		
b8	OV	重复计数控制上溢出屏蔽	0: 上计数溢出使能 1: 上计数溢出屏蔽										RW		
b7~b0	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		

14.4.4.13 通道 0 比较捕获寄存器 (ATIMx_CCR0A/B)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCR0y															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	CCR0y	比较捕获寄存器	作为比较时具有缓存功能(y=A,B)										RW		

14.4.4.14 通道 1/2 比较捕获寄存器 (ATIM3_CCR1/2 A/B)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCRxy															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	CCR0y	比较捕获寄存器	作为比较时具有缓存功能(x=1,2; y=A,B)										RW		

14.4.4.15 通道 3 比较捕获寄存器 (ATIM3_CCR3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CCR3															
位	标记	位名	功能										读写		
b31~b16	Res	保留位	保留, 必须保持复位值												
b15~b0	CCR3	比较捕获寄存器	作为比较时具有缓存功能										RW		

14.4.4.16 通道 3 控制寄存器 (ATIM3_CRCH3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										C3	CIS	CD	CI	BUF	
位	标记	位名	功能										读写		
b31~b6	Res	保留位	保留, 必须保持复位值												
b5	C3	通道比较使能	0: 禁止 1: 使能										RW		
b4~b3	CIS	通道比较匹配设置	00 无匹配 01 上升匹配 10 下降匹配 11 双匹配										RW		
b2	CD	比较触发DMA使能	0: 禁止 1: 使能										RW		
b1	CI	比较触发中断使能	0: 禁止 1: 使能										RW		
b0	BUF	比较缓存使能控制	0: 禁止 1: 使能										RW		

15 独立看门狗 (IWDG)

15.1 概述

此器件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。该看门狗由内部低速时钟 (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
- 溢出周期 400us - 209s
- 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 模式时可自动暂停计数

15.3 功能描述

15.3.1 功能框图

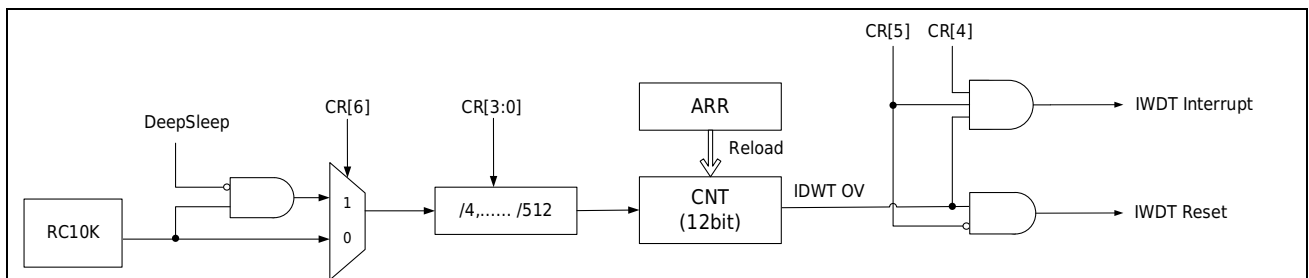


图 15-1 IWDG 功能框图

启动看门狗后，计数器从 0xFFFF 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 IWDT_WINR 寄存器的值大于等于 IWDT_ARR 时，IWDT 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 IWDT_CR.ACTION 为 0 时，看门狗的溢出信号会复位 MCU；当设置 IWDT_CR.ACTION 为 1 且 IWDT_CR.IE 为 1 时，看门狗的溢出信号会触发中断。

15.3.5 看门狗暂停

若设置 IWDT_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 IWDT_KR 寄存器依次写入 0x1234，0x5678 可以停止正在运行的看门狗。

向 IWDT_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

IWDT_CR、IWDT_ARR 和 IWDT_WINR 寄存器具有写保护。若需修改这三个寄存器，应向 IWDT_KR 寄存器写入 0x5555 以解除这些寄存器的写保护；向 IWDT_KR 寄存器写入其它值则会使能这些寄存器的写保护。

15.4 编程示例

15.4.1 配置 IWDG 为窗口看门狗

- Step1. 向 IWDG_KR 寄存器写入 0xCCCC，使能 IWDG。
- Step2. 延时 500μs。
- Step3. 向 IWDG_KR 寄存器写入 0x5555，解除寄存器写保护。
- Step4. 配置 IWDG_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- Step5. 等待 IWDG_SR.PRSF 为 0。
- Step6. 配置 IWDG_ARR，选择看门狗的溢出周期。
- Step7. 等待 IWDG_SR.ARRF 为 0。
- Step8. 向 IWDG_WINR 写入需要的数值，配置窗口大小。
- Step9. 等待 IWDG_SR.WINRF 变为 0，自动刷新计数器值为 IWDG_ARR。
- Step10. 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

- Step1. 向 IWDG_KR 寄存器写入 0xCCCC，使能 IWDG。
- Step2. 延时 500μs。
- Step3. 向 IWDG_KR 寄存器写入 0x5555，解除寄存器写保护。
- Step4. 配置 IWDG_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- Step5. 等待 IWDG_SR.PRSF 为 0。
- Step6. 配置 IWDG_ARR，选择看门狗的溢出周期。
- Step7. 等待 IWDG_SR.ARRF 为 0。
- Step8. 刷新计数器值为 IWDG_ARR 的值 (IWDG_KR = 0x0000 AAAA)。
- Step9. 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 IWDI 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]=0xFFF
4 分频	000	$0.1 \times 4 = 0.4$	$0.1 \times 4 \times 4096$
8 分频	001	0.1×8	$0.1 \times 8 \times 4096$
16 分频	010	0.1×16	$0.1 \times 16 \times 4096$
32 分频	011	0.1×32	$0.1 \times 32 \times 4096$
64 分频	100	0.1×64	$0.1 \times 64 \times 4096$
128 分频	101	0.1×128	$0.1 \times 128 \times 4096$
256 分频	110	0.1×256	$0.1 \times 256 \times 4096$
512 分频	111	0.1×512	$0.1 \times 512 \times 4096 = 209715.2$

15.5 寄存器描述

15.5.1 寄存器列表

IWDT 基地址：0x4000 8C80

表 15-2 IWDT 寄存器列表

寄存器名	偏移地址	复位值
键值寄存器 (IWDT_KR)	0x00	0x0000 0000
控制寄存器 (IWDT_CR)	0x04	0x0000 0000
重载寄存器 (IWDT_ARR)	0x08	0x0000 0FFF
状态寄存器 (IWDT_SR)	0x0C	0x0000 0000
窗口寄存器 (IWDT_WINR)	0x10	0x0000 0FFF

15.5.2 键值寄存器 (IWDT_KR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KR															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15~b0	KR	IWDT键值寄存器	写入0xCCCC: 启动IWDT计数器	WO
			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	

注:

- 在启动 IWDT 计数器后, 即向 IWDT_KR 寄存器写入 0xCCCC 使能 IWDT 后, 需至少添加 500us 延时以等待时钟稳定, 而后再进行 IWDT 的其它寄存器的读写。

15.5.3 控制寄存器 (IWDT_CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Res																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Res									PAUSE	ACTION	IE	Res	PRS			
位	标记		位名		功能									读写		
b31~b7		Res	保留位		保留，必须保持复位值											
b6		PAUSE	Deepsleep模式下WDT暂停控制		0：DeepSleep模式下IWDT继续运行 1：DeepSleep模式下IWDT自动暂停									RW		
b5		ACTION	IWDT溢出后动作配置		0：看门狗溢出后产生复位 1：看门狗溢出后产生中断									RW		
b4		IE	IWDT中断使能控制		0：禁止看门狗中断 1：使能看门狗中断									RW		
b3		Res	保留位		保留，必须保持复位值											
b2~b0		PRS	配置看门狗计数时钟与RC10K振荡器的分频比		111：512分频					011：32分频					RW	
					110：256分频					010：16分频						
					101：128分频					001：8分频						
					100：64分频					000：4分频						

注:

1. 需要向 IWDT_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
2. IWDT_SR.PRSF 为 0, 才可修改本寄存器。

15.5.4 重载寄存器 (IWDT_ARR)

复位值: 0x0000 0FFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ARR											

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11~b0	ARR	IWDT重载值	IWDT重载值	RW

注:

1. 向 IWDT_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
2. IWDT_SR.ARRF 为 0, 才可修改本寄存器。

15.5.5 状态寄存器 (IWDT_SR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											RUN	OV	WINRF	ARRF	PRSF

位	标记	位名	功能	读写
b31~b5	Res	保留位	保留, 必须保持复位值	
b4	RUN	WDT运行标志	1: WDT正在运行 0: WDT没有运行	RO
b3	OV	WDT溢出标志	1: WDT溢出, 写0清零 0: WDT没有溢出	RW0
b2	WINRF	WINR寄存器更新标志	1: WINR寄存器正在更新 0: WINR寄存器更新完成 注意: 只有 WINRF 位为 0 时才可以更新WINR寄存器。	RO
b1	ARRF	ARR寄存器更新标志	1: ARR寄存器正在更新 0: ARR寄存器更新完成 注意: 只有 ARRF 位为 0 时才可以更新ARR寄存器。	RO
b0	PRSF	CR寄存器更新标志	1: CR寄存器正在更新 0: CR寄存器更新完成 注意: 只有 PRSF 为 0 时才可以更新CR寄存器。	RO

15.5.6 窗口寄存器 (IWDT_WINR)

复位值: 0x00000FFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WINR											
位	标记	位名	功能										读写		
b31~b12	Res	保留位	保留, 必须保持复位值												
b11~b0	WINR	窗口比较器比较值	窗口比较器比较值										RW		

注:

- 1. 向 IWDT_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
- 2. IWDT_SR.WINRF 为 0, 才可修改本寄存器。

16 窗口看门狗 (WWDT)

16.1 概述

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

- 6 比特递减计数器
- PCLK 分频时钟作为计数时钟
- 当 PCLK 为 48MHz 时，其超时时间为 85 μ s ~ 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 功能描述

16.3.1 功能框图

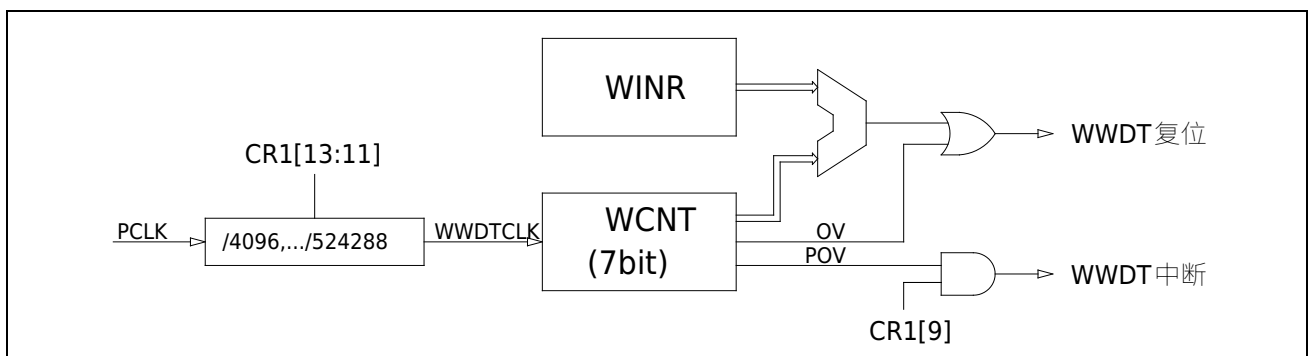


图 16-1 WWDT 功能框图

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 WWDT_CR0.WCNT 时，WWDT_CR0.WCNT 的数值大于 WWDT_CR1.WINR 的数值，则看门狗也会产生复位信号。

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。

超时时间的计算公式如下方所示：

$$T_{\text{WWDT}} = T_{\text{PCLK}} \times 4096 \times 2^{\text{CR1.PRS}} \times (\text{CR0.WCNT} - 0x3F)$$

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 编程示例

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。

- Step1. 配置 WWDT_CR1.PRS 为 2，WWDT 的计数时钟为 PCLK/ 16384；
- Step2. 配置 WWDT_CR1.WINR 为 0x4F，WWDT 的更新窗口为 0x4F~0x40；
- Step3. 配置 WWDT_CR0.WCNT 为 0x6F，WWDT 的溢出周期为 786432 个 PCLK；
- Step4. 设置 WWDT_CR0.EN 为 1，启动看门狗；
- Step5. 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，以重载看门狗计数器。

16.5 寄存器描述

16.5.1 寄存器列表

WWDT 基地址：0x4000 8800

表 16-1 WWDT 寄存器列表

寄存器名	偏移地址	复位值
控制寄存器0 (WWDT_CR0)	0x00	0x0000 007F
控制寄存器1 (WWDT_CR1)	0x04	0x0000 007F
状态寄存器 (WWDT_SR)	0x08	0x0000 0000

注：WWDT 产生复位后影响寄存器值。

16.5.2 控制寄存器 0 (WWDT_CR0)

复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								EN	WCNT						
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7	EN	WWDT 使能控制	该位使能后不能软件清零 1: 使能WWDT 0: 禁止WWDT (状态)										RW1		
b6~b0	WCNT	WWDT 计数器	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)

复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		PRS			Res	IE	Res		WINR						
位		标记		位名			功能							读写	
b31~b14		Res		保留位			保留，必须保持复位值								
b13~b11		PRS		WWDT计数时钟配置			000：PCLK / 4096							RW	
							001：PCLK / 8192								
							010：PCLK / 16384								
							011：PCLK / 32768								
							100：PCLK / 65536								
							101：PCLK / 131072								
							110：PCLK / 262144								
							111：PCLK / 524288								
b10		Res		保留位			保留，必须保持复位值								
b9		IE		WWDT预溢出中断使能控制			1：使能预溢出中断 0：禁止预溢出中断							RW	
b8~b7		Res		保留位			保留，必须保持复位值								
b6~b0		WINR		看门狗窗口值			看门狗窗口值							RW	

16.5.4 状态寄存器 (WWDT_SR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															POV
位		标记		位名		功能								读写	
b31~b1		Res		保留位		保留，必须保持复位值									
b0		POV		WWDT预溢出标志		1：WWDT已发生预溢出，写0以清除该标志 0：WWDT未发生预溢出								RW0	

17 通用同步异步收发器 (USART)

17.1 概述

本产品搭载 4 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。

17.2 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 TXD、RXD 引脚交换
- 支持 TXD、RXD 电平反向
- 支持 DMA 连续通讯

17.2.1 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
- 支持休眠模式唤醒功能

17.2.2 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

17.2.3 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 IO 信号边沿整型
- 支持主机接口

17.2.4 红外通信特性

- 支持 $\frac{3}{16}$ 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
- 支持休眠模式唤醒功能

17.2.5 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
- 支持休眠模式唤醒功能

17.3 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。

注：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。

17.3.1 通信波特率编程

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 17-1 USART 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 = $8 \times (2 - \text{OVER8})$	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 = $2 \times \text{BCN}$	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 = $8 \times (2 - \text{OVER8})$	20K

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 所设定的基本时钟数

DIVI： USART_BRR.DIVI 设定值

DIVF： USART_BRR.DIVF 设定值

注：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

已知通信波特率，计算 USART_BRR 寄存器值的步骤如下所示：

Step1. 配置 USART_BRR.PSC，使下述表达式的计算结果小于等于 256；

$$\frac{\text{PCLK}}{4^{\text{PSC}} \times \text{Baud} \times \text{采样计数}}$$

Step2. 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；

$$\text{DIVI} = \text{INT} \left(\frac{\text{PCLK}}{4^{\text{PSC}} \times \text{Baud} \times \text{采样计数}} - 1 \right)$$

Step3. 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；

$$\text{DIVF} = \text{INT} \left(\frac{\text{Baud} \times \text{采样计数} \times (\text{DIVI} + 1) \times 256 \times 4^{\text{PSC}} \times 1.0}{\text{PCLK}} - 128 + 0.5 \right)$$

Step4. 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。

17.3.1.1 异步串口通信波特率编程样例

表 17-2 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 17-3 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 17-4 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%
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 17-5 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 17-6 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%
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 17-7 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%					

17.3.2 自动波特率检测

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:

$$\text{Baud} = \frac{f_{\text{PCLK}}}{4^{\text{PSC}} \times (\text{BDR} + 1)}$$

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。

注:

1. 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。
2. 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。
3. 通信波特率检测适用于异步串口模式和 LIN 模式。

17.3.3 DMA 连续通信

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。

注:

1. 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。
2. 单线半双工工作时不支持 DMA 模式。

17.3.4 硬件流控

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。

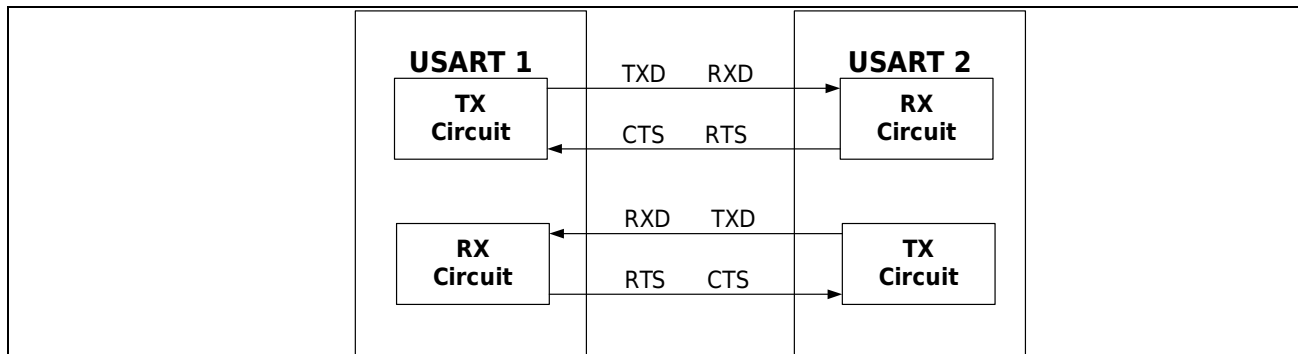


图 17-1 异步串口模式硬件流控示例

CTS 流控功能

当设置 USART_CR2.CTSSEN 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;

注:

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.RTSSEN 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。

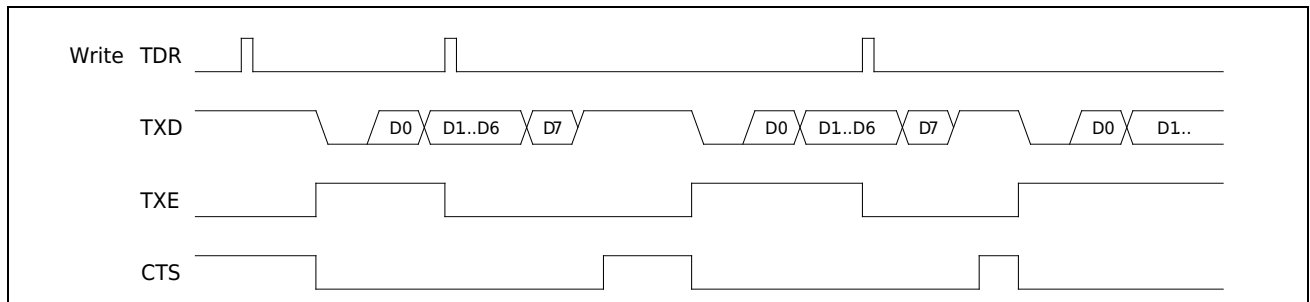


图 17-2 CTS 硬件流控信号

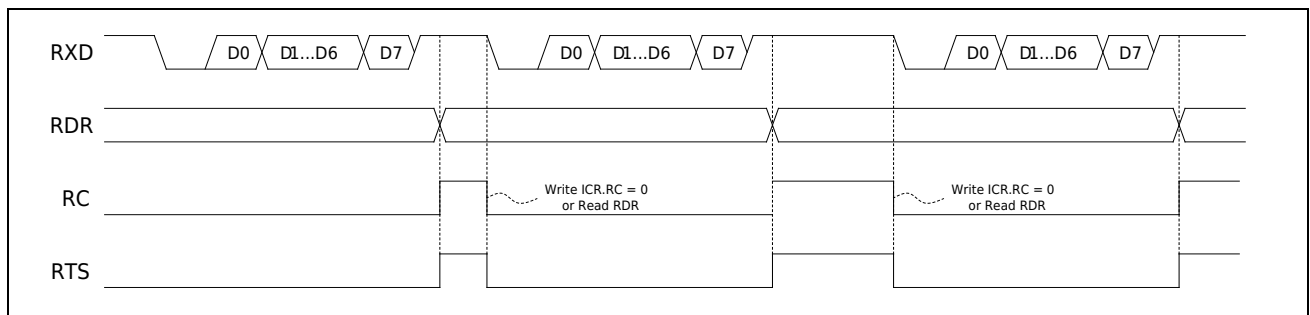


图 17-3 RTS 硬件流控信号

注：硬件流控适用于异步串口模式和同步串口模式。

17.3.5 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为 $t_{PCLK} \times 4^{PSC} \times (TOR+1)$ 。

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 0.02 μ s，最大值约为 1.4s。

本定时计数器可实现五种功能，其功能对比如下所示：

表 17-8 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 后立即启动	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，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）

注：

1. 只有当 USART_CR1.TOE 为 0 时才能写入其它值以启动定时计数器。
2. 当 USART_SR.TO 为 1 后，硬件自动设置 USART_CR1.TOE 为 0。

17.3.5.1 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。

17.3.5.2 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 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。

17.3.5.3 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 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。

17.3.5.4 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。

17.3.5.5 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。

参考自动波特率检测章节了解更多关于自动波特率检测功能。

17.3.6 中断

参考【中断控制器（NVIC）】章节的“外部中断与 NVIC 中断输入对应关系表格”，本章节的 4 个 USART 各提供 3 类中断：TX 中断、RX 中断和 FLAG 中断。

- 1) TX 中断对应 USART_IER.TXE 或 USART_IER.TC 中断使能
- 2) RX 中断对应 USART_IER.RC 中断使能
- 3) FLAG 中断对应 USART_IER[14:3]中的任意一个或多个中断使能

表 17-9 USART 各模式下中断有效性

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	●	●	●	●	●
接收完成中断	RC	●	●	●	●	●
接收数据校验错误中断	PE	●	X	●	●	X
接收帧错误中断	FE	●	X	●	●	●
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	●	X	X	X	X
CTS中断	CTS	●	●	X	X	X
自动波特率检出中断	ABR	●	X	X	X	●
同步间隔段检出中断	LBD	X	X	X	X	●
输入数据唤醒中断	DWK	●	X	X	●	●

注：

●：有效

X：无效

17.3.7 异步串口模式

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 ~ 4 个引脚与其它设备连接在一起，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

17.3.7.1 功能框图

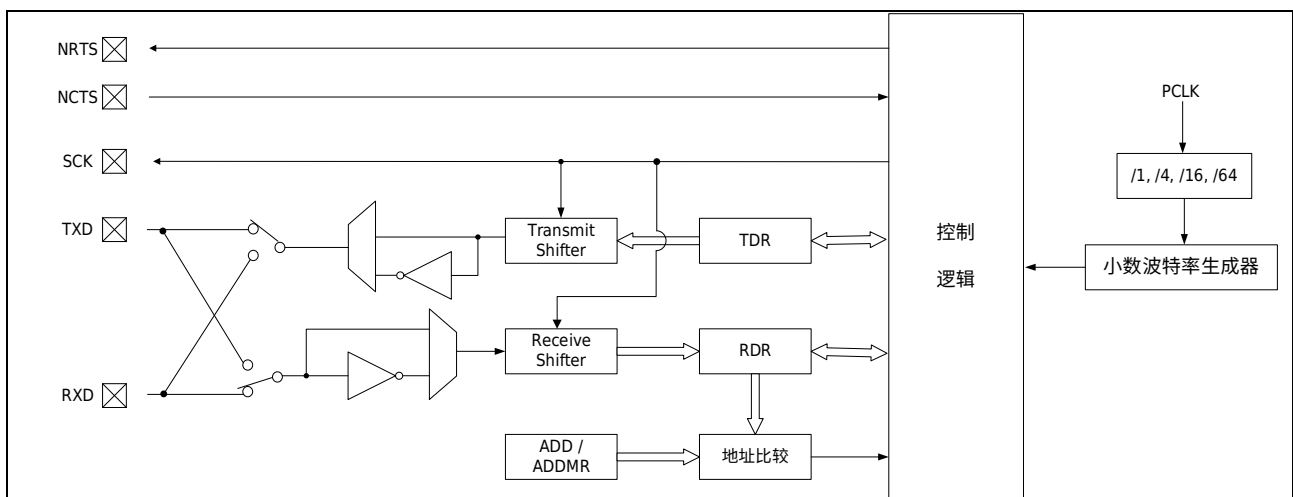


图 17-4 异步串口功能框图

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

17.3.7.2 数据格式

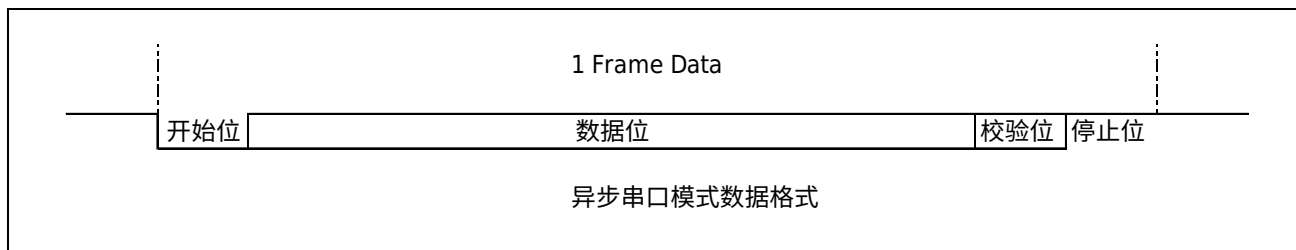


图 17-5 异步串口模式数据格式

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。

起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）。

17.3.7.3 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。

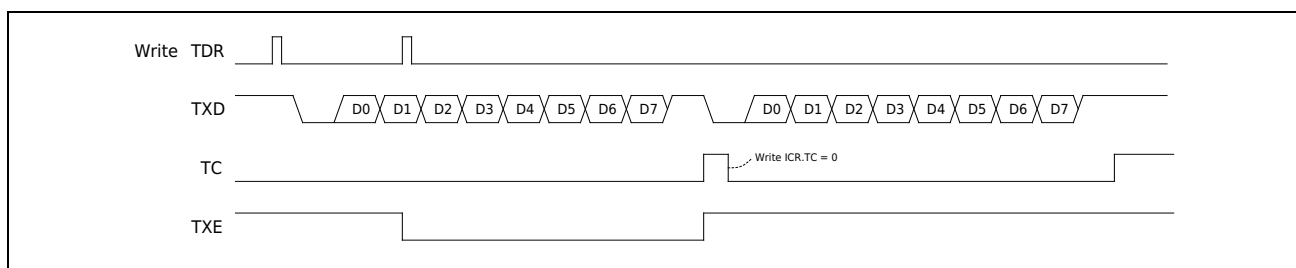


图 17-6 发送数据示例

注：

1. 向 USART_TDR 中写入待发送数据前，USART_CR1.TE 位必须先置 1。

2. 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。

17.3.7.4 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外理，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。

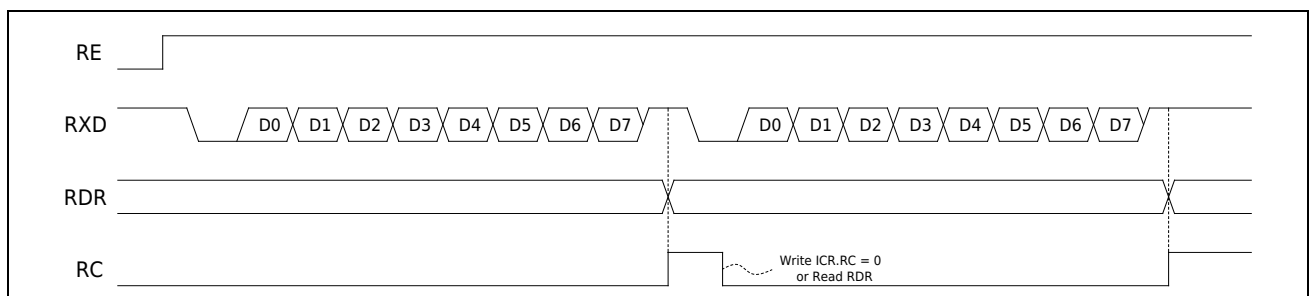


图 17-7 接收数据示例

17.3.7.5 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。

注：单线半双工工作时不支持 DMA 模式。

17.3.7.6 多处理器通信

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

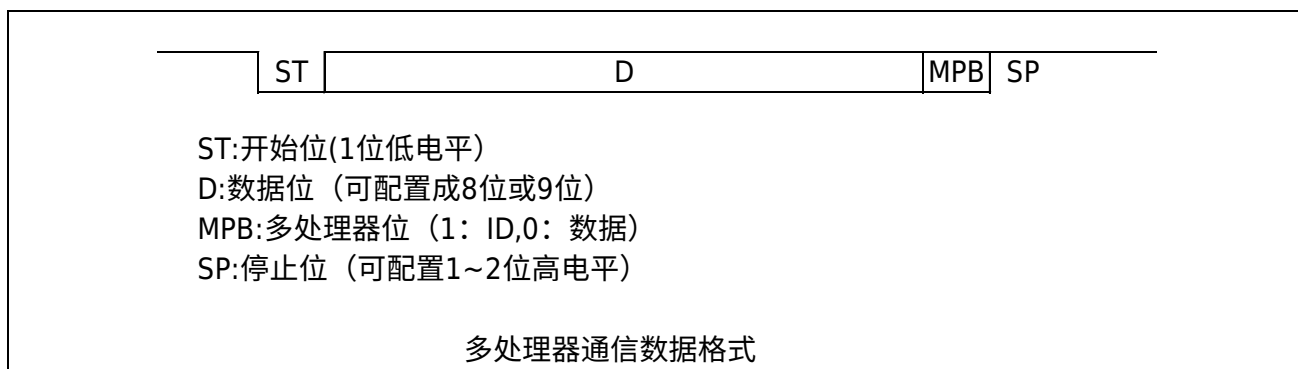


图 17-8 多处理器通信数据格式

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)。

注：多处理器通信只适用于异步串口模式。

17.3.7.7 发送数据编程示例

- Step1. 按 GPIO 章节管脚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，设置 GPIOx_AFRH/AFRL 将 TXD 映射到需要的管脚；
- Step2. 设置 APBEN0.USARTn(n=0/1/2/3)为 1，使能 USART 配置时钟及工作时钟；
- Step3. 向 USART_CR1 写入 0x00，使 USART 模块回复到初始态；
- Step4. 配置 USART_CR2.SYNC 为 0，使 USART 模块工作于异步串口模式；
- Step5. 配置 USART_CR1.OVER8 及 USART_BRR 寄存器，配置通信波特率；
- Step6. 配置 USART_CR1.CHLEN，配置帧内数据位的个数；
- Step7. 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；
- Step8. 配置 USART_CR1.MSB，配置发送数据时的数据序列顺序；
- Step9. 设置 USART_CR1.TE 为 1，使能 TXD 发送功能；
- Step10. 向 USART_ICR.TC 写入 0x00，清除发送完成标志；
- Step11. 向 USART_TDR 寄存器写入一帧待发送的数据；
- Step12. 查询等待 USART_SR.TC 变为 1，本次写入的数据帧发送完成；

Step13. 如待发送的数据未完成，则跳转到 Step10 继续执行。

17.3.7.8 接收数据编程示例

- Step1. 按 GPIO 章节管脚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；
- Step2. 设置 APBEN0.USARTn(n=0/1/2/3) 为 1，使能 USART 配置时钟及工作时钟；
- Step3. 向 USART_CR1 写入 0x00，使 USART 模块回复到初始态；
- Step4. 配置 USART_CR2.SYNC 为 0，使 USART 模块工作于异步串口模式；
- Step5. 配置 USART_CR1.OVER8 及 USART_BRR 寄存器，配置通信波特率；
- Step6. 配置 USART_CR1.CHLEN，配置帧内数据位的个数；
- Step7. 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；
- Step8. 配置 USART_CR1.MSB，配置接收数据时的数据序列顺序；
- Step9. 配置 USART_CR1.SBS，选择起始位的判定方式；
- Step10. 向 USART_ICR 寄存器写入 0x00，清除所有状态标志；
- Step11. 设置 USART_CR1.RE 为 1；
- Step12. 查询等待 USART_SR.RC 变为 1，接收完成一帧数据；
- Step13. 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；
- Step14. 向 USART_ICR.RC 写入 0x00，清除接收完成标志；
- Step15. 如待接收的数据未完成，则跳转到 Step12 继续执行。

17.3.8 同步串口模式

同步串口模式是指数据的发送和接收都是与 SCK 时钟管脚同步的。

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 和 USART_CR3.IREN

17.3.8.1 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。

17.3.8.2 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。

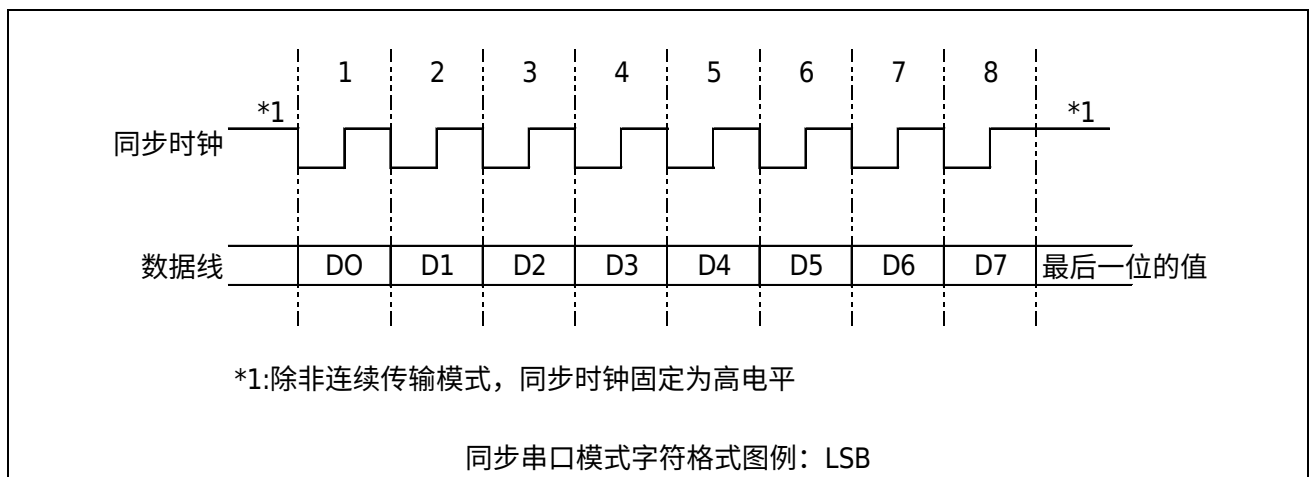


图 17-9 同步串口模式字符格式图例

17.3.8.3 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。

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。

发送数据操作步骤

- Step1. 将 USART_CR1, USART_SR 寄存器设定为复位值;
- Step2. 初始化端口, 配置 TXD 为数字输入带上拉; 设置 GPIOx_AFRH/AFRL 将 TXD 映射到需要的管脚;
- Step3. 通过 USART_CR2.CLKC 位选择时钟源;
- Step4. 设定 USART_CR1 (除了 TE)、USART_CR2、USART_CR3 寄存器;
- Step5. 设定 USART_BRR.PSC 选择预分频值, 设定 USART_BRR 通信波特率;
- Step6. 设定 USART_CR1.TE=1, USART_IER.TXE=1;
- Step7. 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 (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);
- Step8. 如果需要连续发送数据, 则重复步骤 7;
- Step9. 通过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。

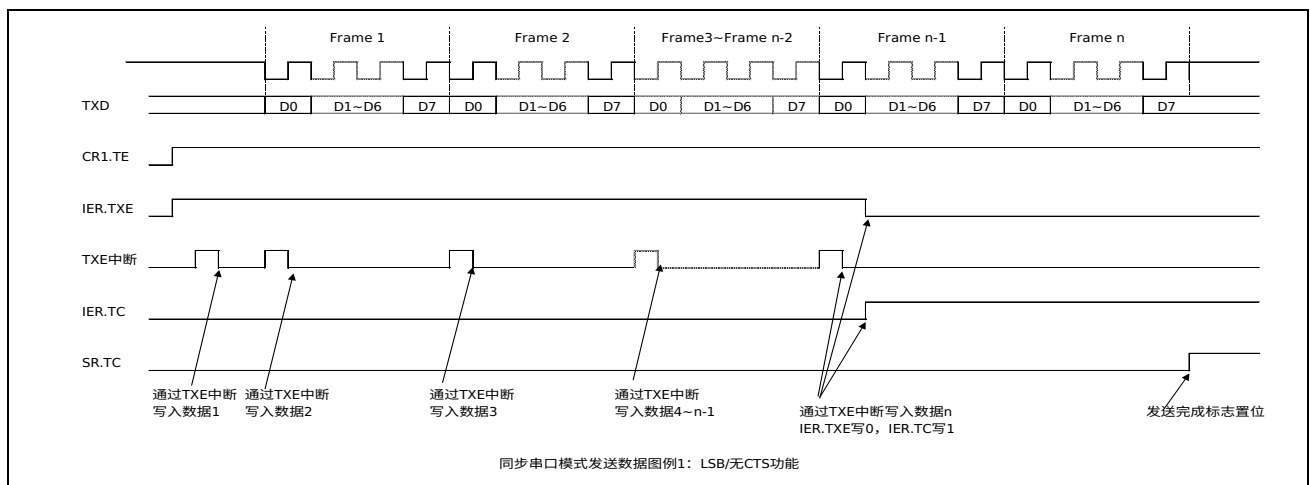


图 17-10 同步串口模式发送数据图例 1

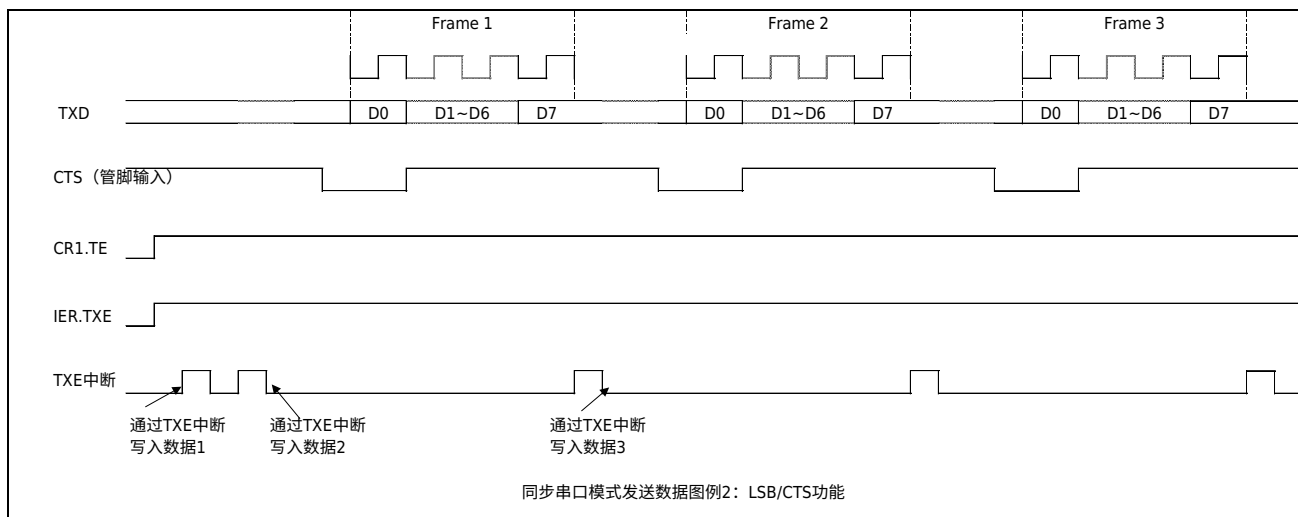


图 17-11 同步串口模式发送数据图例 2

发送器中断

同步串口模式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。

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。

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。

17.3.8.4 接收器

接收数据操作步骤

- Step1. 将 USART_CR1，USART_SR 寄存器设定为复位值；
- Step2. 初始化端口，配置 RXD 为数字输入带上拉；设置 GPIOx_AFRH/AFRL 将 RXD 映射到需要的管脚；
- Step3. 通过 USART_CR2.CLKC 位选择时钟源；
- Step4. 设定 USART_CR1（除了 RE）、USART_CR2、USART_CR3 寄存器；
- Step5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- Step6. 设定 USART_IER.RC=1，USART_IER.OR=1，USART_CR1.RE=1（使用 RTS 功能时，USART_CR1.RE=1 后 RTS 管脚输出低电平）；
- Step7. 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。
 - 1) 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
 - 2) 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

(使用 RTS 功能时, 数据读取后 RTS 管脚输出低电平)

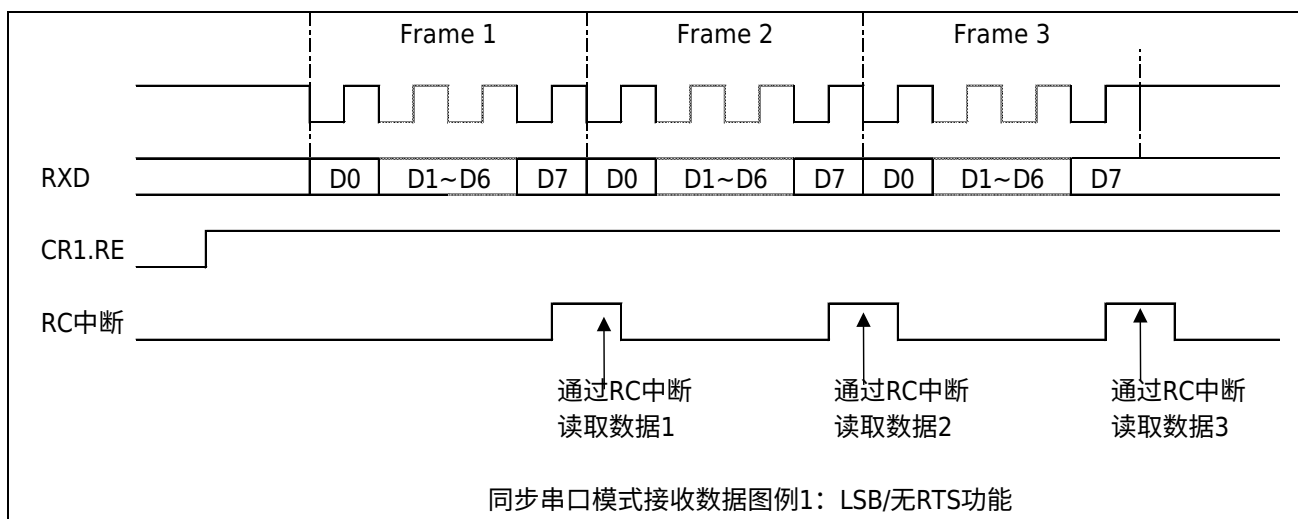


图 17-12 同步串口模式接收数据图例 1

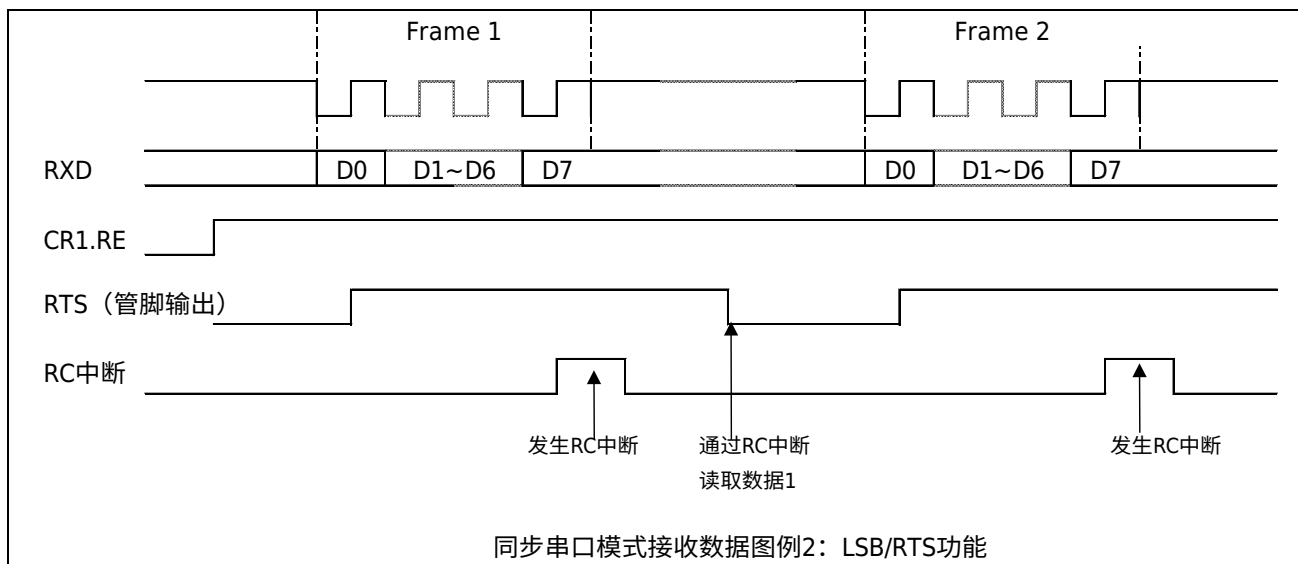


图 17-13 同步串口模式接收数据图例 2

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。

17.3.8.5 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。

17.3.8.6 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。

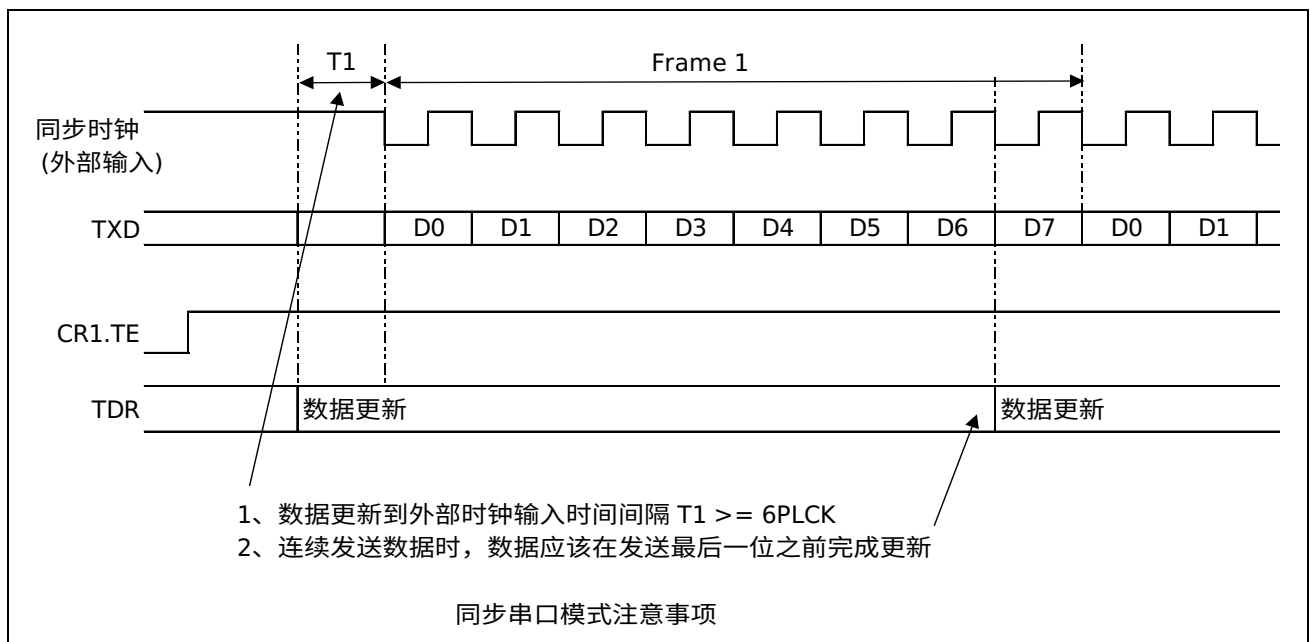


图 17-14 同步串口模式注意事项

17.3.9 智能卡(7816)模式

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 和 USART_CR3.IREN

此外，需要设置 USART_CR2.CLKC 位，为智能卡提供时钟。

17.3.9.1 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：

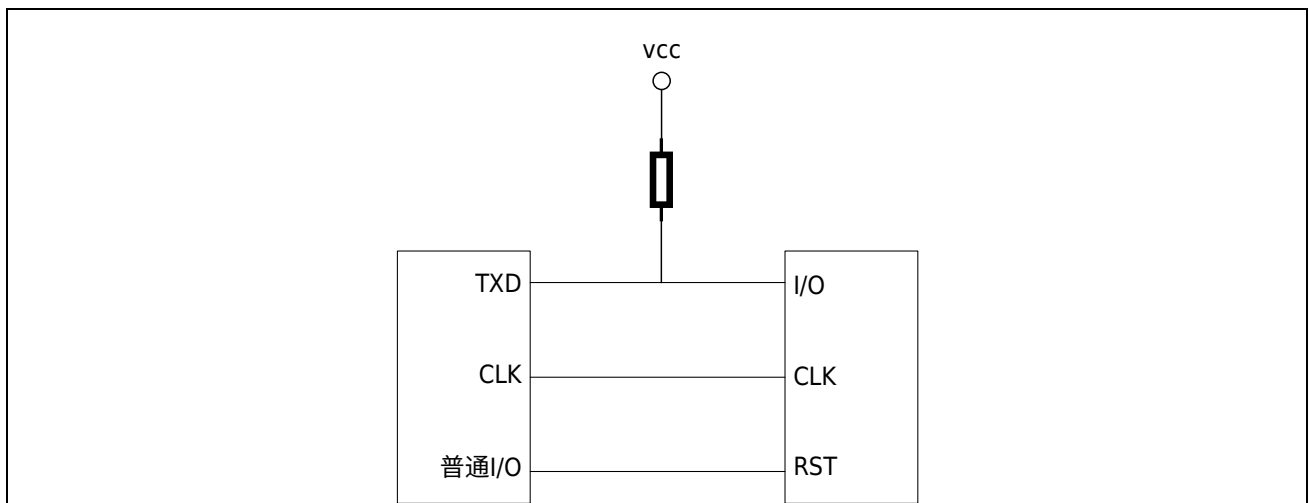


图 17-15 智能卡连接示意图

17.3.9.2 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。

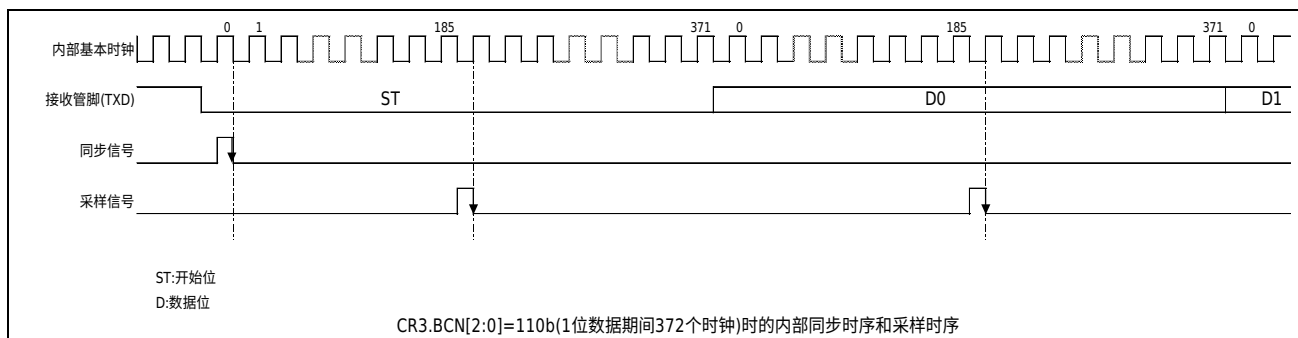


图 17-16 智能卡模式同步时序和采样时序图

接收容差的计算公式如下：

$$M[\%] = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

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

17.3.9.3 数据格式

智能卡模式下一帧数据由起始位，数据位和校验位组成。

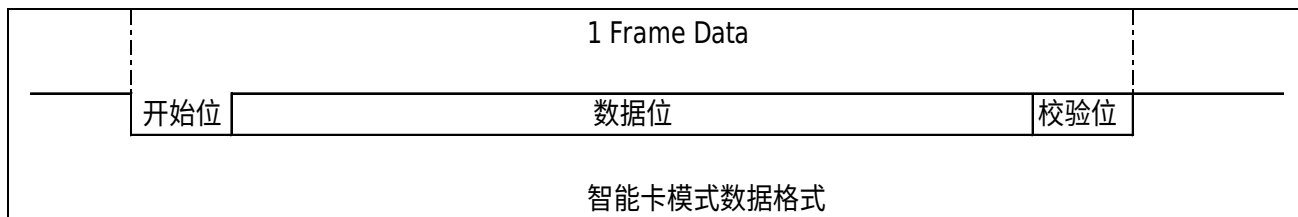


图 17-17 智能卡模式数据格式

起始位

起始位固定为一位低电平。

数据位

数据位固定为 8 位数据。

校验位

校验位需配置成 1 位偶校验。

下图给出数据 0x3B 数据格式的图例。

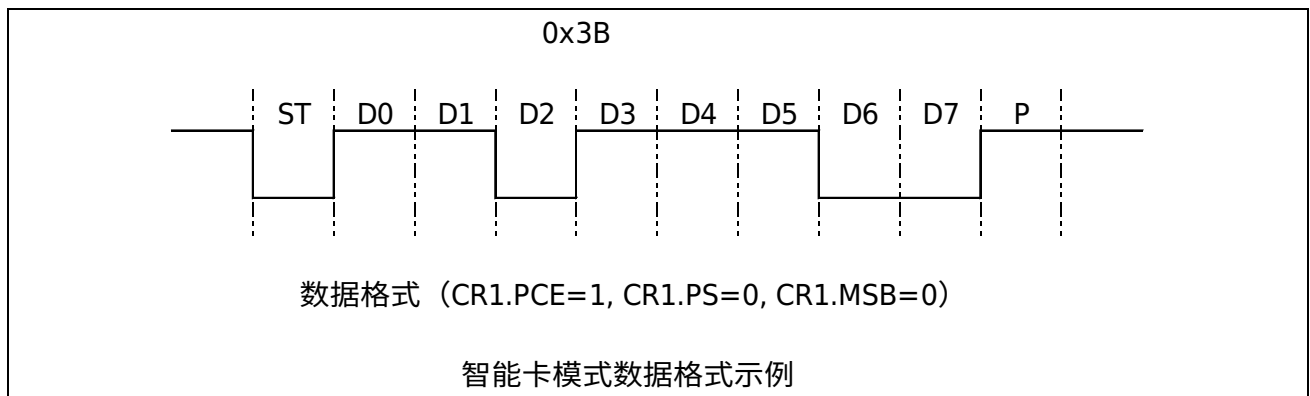


图 17-18 智能卡模式数据格式示例

17.3.9.4 智能卡的初始化设定步骤

- Step1. 将 USART_CR1 寄存器设定为复位值；
- Step2. 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；
- Step3. 状态寄存器确认，USART_SR 寄存器设定为复位值；
- Step4. 设定 USART_CR1 (除了 TE,RE)、USART_CR2, USART_CR3 寄存器；
- Step5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- Step6. 设定 USART_CR2.CLKC 位的时钟控制；
- Step7. 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。

17.3.9.5 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。

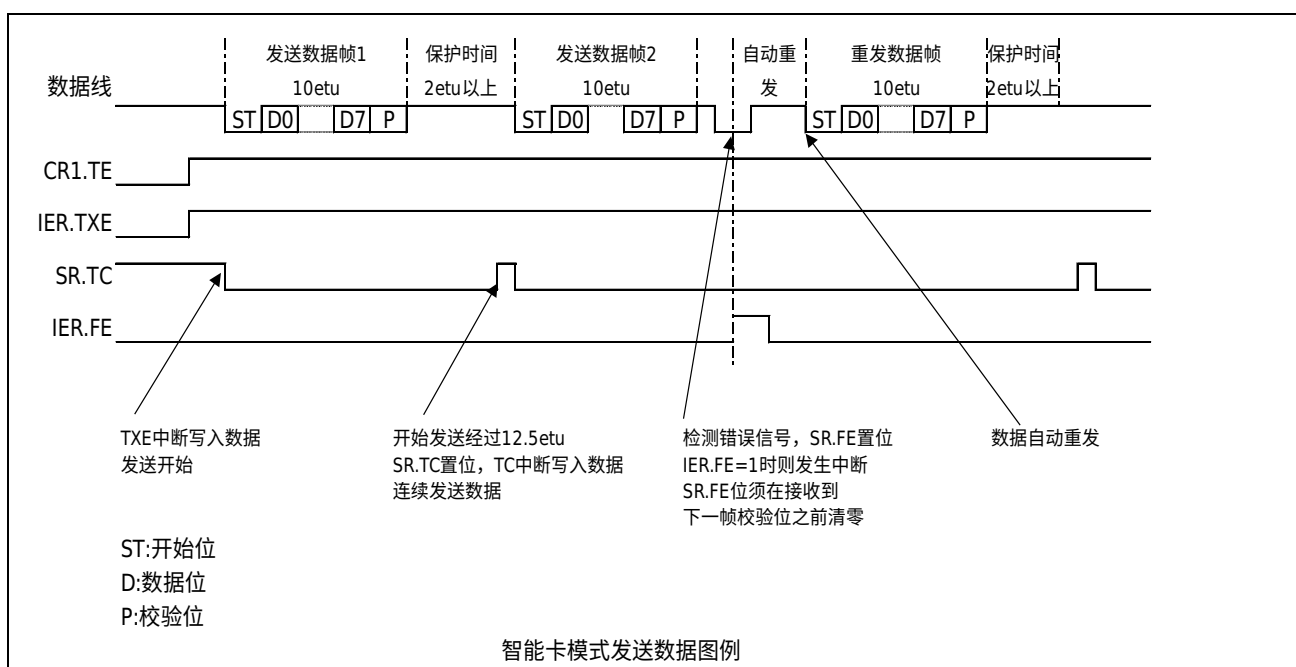


图 17-19 智能卡模式发送数据图例

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。

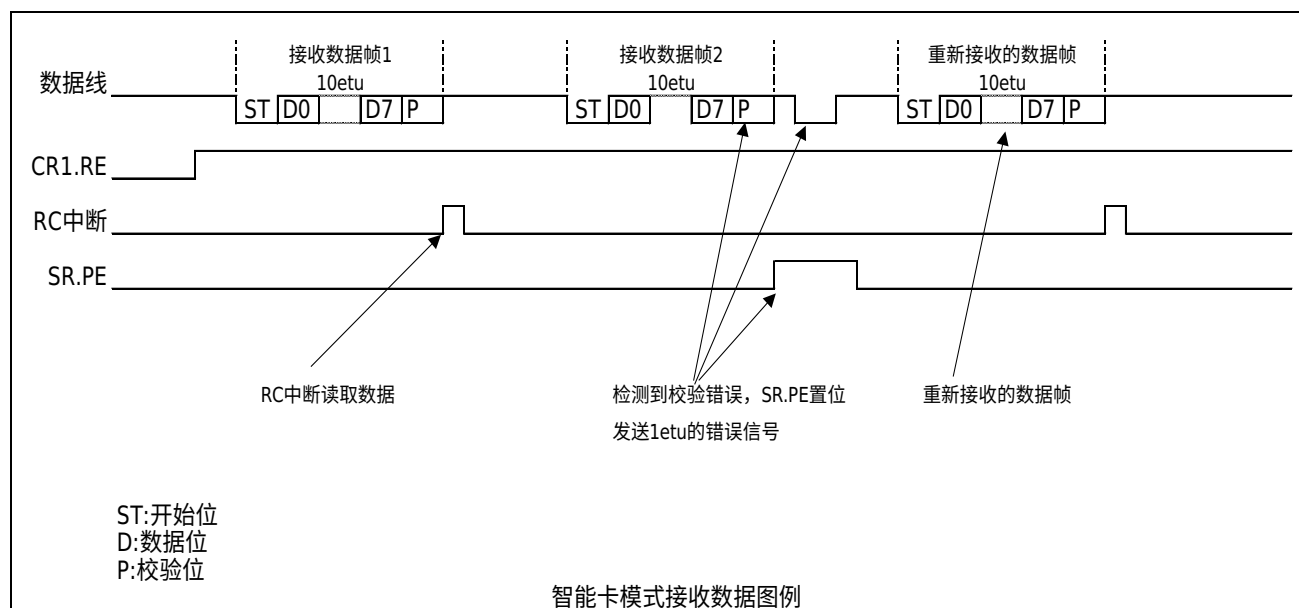


图 17-20 智能卡模式接收数据图例

17.3.10 红外模式

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 和 USART_CR3.LINEN

本模块包含 2 种红外模式： $\frac{3}{16}$ 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。

17.3.10.1 红外模式 1：红外脉冲宽度为波特率周期的 $\frac{3}{16}$

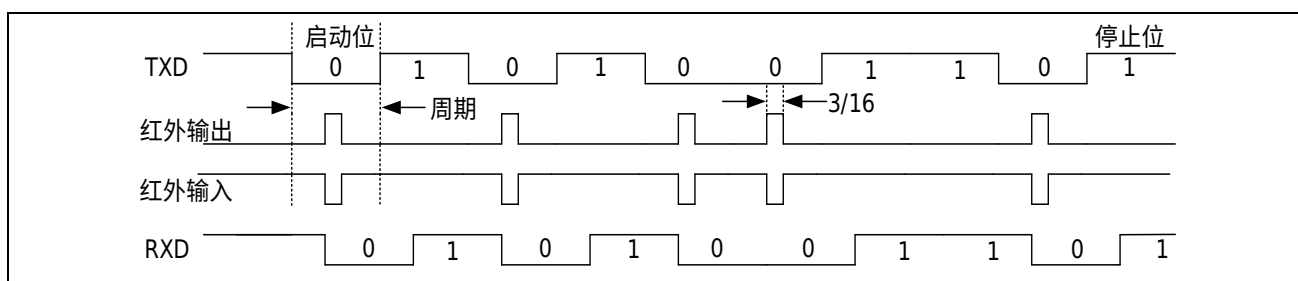


图 17-21 红外数据调制

通过写 USART_CR3.IREN=1 和 USART_CR3.IRCW=0 来选择 $\frac{3}{16}$ 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 $\frac{3}{16}$ 。

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 $\frac{3}{16}$ （参见上图）。
- 3) 规范要求脉冲宽度值要大于 1.41us。
- 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，详见寄存器部分的注释。

- 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。

发送数据设定步骤

- Step1. 将 USART_CR1 寄存器设定为复位值；
- Step2. 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；
- Step3. 设定 USART_CR1 中除 TE 以外的值，设定 USART_CR2，USART_CR3 寄存器；
- Step4. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- Step5. 设 USART_CR3.IREN=1 (USART_CR2.TXINV 根据需要配置)，USART_CR1.TE=1，设定 USART_IER.TXE=1；
- Step6. 通过发送中断程序写通信数据到 USART_TDR，发送开始；
- Step7. 如果需要连续发送数据时，重复步骤 6；
- Step8. 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；
- Step9. 写 USART_CR1.TE=0。

接收数据设定步骤

- Step1. 将 USART_CR1 寄存器设定为复位值；
- Step2. 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；
- Step3. 设定 USART_IER.RC=1；
- Step4. 设定 USART_CR1 中除 RE 以外的值，设定 USART_CR2，USART_CR3 寄存器；
- Step5. 设定 USART_BRR.PSC 选择预分频值，设定 USART_BRR 通信波特率；
- Step6. USART_CR3.IREN=1 (USART_CR2.RXINV 根据需要配置)，USART_CR1.RE=1；
- Step7. 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；
- 1) 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
 - 2) 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
 - 3) 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

- 4) 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

Step8. 写 USART_CR1.RE=0。

17.3.10.2 红外模式 2：占空比可调的 38K 调制

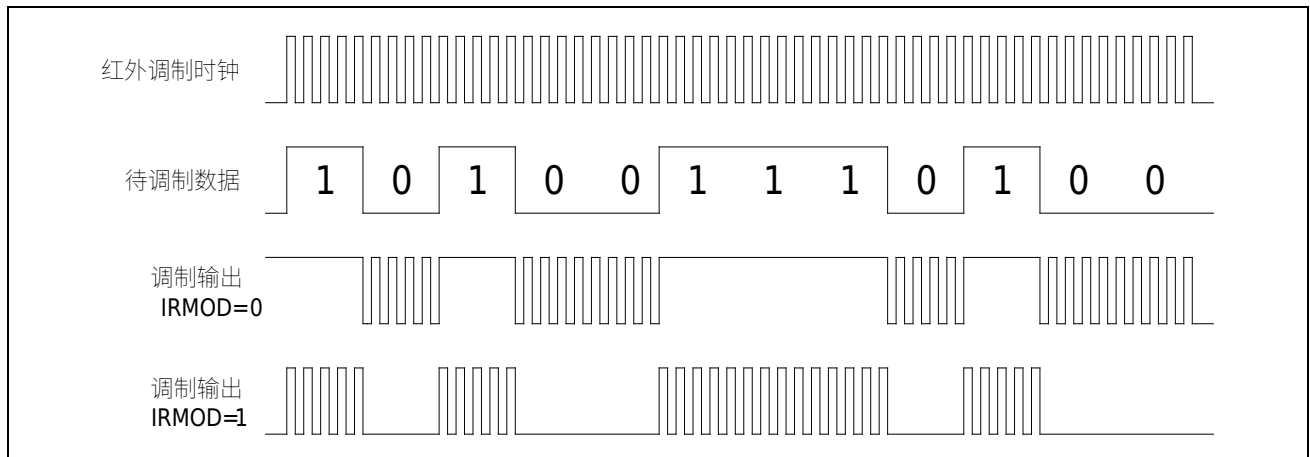


图 17-22 38K 红外调制

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：

$$F = PCLK / (USART_IR.DCNTH + USART_IR.DCNTL + 2)$$

其中：PCLK 为模块工作频率，

USART_IR.DCNTH 决定 38K 调制频率的高电平持续时间，

USART_IR.DCNTL 决定 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。

17.3.11 LIN 模式

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 和 USART_CR1.STOP、USART_CR1.SLME、USART_CR1.MPE、USART_CR1.PCE
- USART_CR2.SYNC
- USART_CR3.SCEN 和 USART_CR3.IREN

17.3.11.1 功能简介

LIN 是 Local Interconnect Network (局域互连网络) 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。

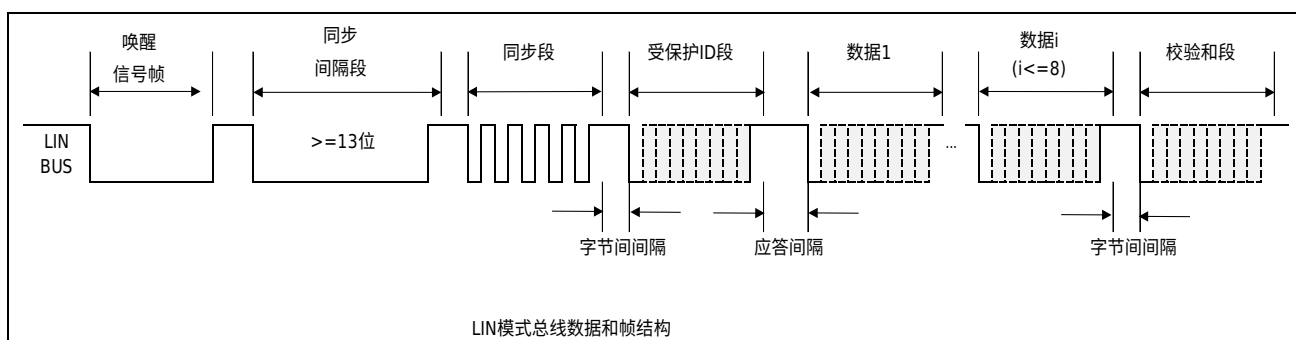


图 17-23 LIN 总线数据和帧结构

注：唤醒信号帧仅当设备需要唤醒时才需要

17.3.11.2 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。

17.3.11.3 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

- Step1. 将 USART_CR1 寄存器设定为复位值；
- Step2. 按 GPIO 章节管脚数字复用功能的相关描述，配置 TXD 和 RXD 为数字输入带上拉，设置 GPIOx_AFRH/AFRL 将 TXD 和 RXD 映射到需要的管脚；
- Step3. 设定 USART_CR1（除 TE），USART_CR2，USART_CR3（除 LINEN）寄存器；
- Step4. 设定 USART_BRR.PSC 选择预分频值；
- Step5. 设定 USART_BRR 寄存器，设定信号帧的波特率；

（以下步骤 6~8 为唤醒信号帧的发送，如无必要则跳过）

- Step6. 设定 USART_CR1.TE=1；
- Step7. 写通信数据 0x80 到 USART_TDR，数据传输到移位寄存器，发送开始；
- Step8. 等待发送完成；

（以下步骤 9~12 为同步间隔段的发送）

- Step9. 设定 USART_CR3.BREAK=1，USART_CR3.LINEN=1；
- Step10. 如果跳过上面 6~8 步骤则设定 USART_CR1.TE=1，否则跳过此步操作；
- Step11. 写同步间隔段长度到 USART_TDR，发送同步间隔段低电平到总线；
- Step12. 等待发送完成；

（以下步骤 13~14 为同步段的发送，其过程与异步串口发送过程相同）

- Step13. 写数据 0x55 到 USART_TDR；

Step14. 等待发送完成；

(以下步骤 15~17 为字节间间隔)

Step15. 写字节间等待时间长度到 USART_TOR 中；

Step16. 写 USART_CR1.TOE=1 启动软件定时；

Step17. 等待 USART_SR.TO=1；

Step18. 受保护 ID 段：写受保护 ID 段数据 (ID) 到 USART_TDR 中；

Step19. 等待发送完成；

(以下 20~22 步骤为应答间隔)

Step20. 写应答等待时间长度到 USART_TOR 中；

Step21. 写 USART_CR1.TOE=1 启动软件定时；

Step22. 等待 USART_SR.TO=1；

Step23. 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)；

Step24. 数据段 1 的字节间间隔，参考步骤 15~17 执行；

Step25. 数据段 2：参考步骤 23 的数据段 1；

Step26. 数据段 2 的字节间间隔，参考步骤 24；

Step27.

Step28. 数据段 n：参考步骤 23 的数据段 1；

Step29. 数据段 n 的字节间间隔：参考步骤 24；

Step30. 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 23；

Step31. 等待发送完成，LIN 操作结束，控制寄存器清零。

注：必须设置校验位 USART_CR1.PCE=0，停止位 USART_CR1.STOP=0，USART_CR1.COMB=0。

17.3.11.4 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。

$$\text{波特率} = 1 / (tPCLK * 4PSC * (BDR + 1))$$

tPCLK：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。波特率的设置参考【通信波特率编程】。

LIN 从机设备软件操作步骤

- Step1. 将 USART_CR1 寄存器设定为复位值；
- Step2. 按 GPIO 章节管脚数字复用功能的相关描述，配置 TXD 和 RXD 为数字输入带上拉，设置 GPIOx_AFRH/AFRL 将 TXD 和 RXD 映射到需要的管脚；
- Step3. 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）；
- Step4. 设置 USART_BRR 的预分频和波特率初始值；

（以下步骤 5 是等待唤醒信号到来，如无必要则跳过）

- Step5. 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；
- Step6. 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）；
- Step7. 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；
- Step8. 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；

（以下步骤 9 为同步段波特率测量）

- Step9. 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；
- Step10. 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“通信时钟源及波特率编程”；最后设 USART_CR1.RE=1；如果不需要调整波特率则跳至步骤 11；

（以下步骤 11~13 为接收受保护 ID 段，数据以及校验和，过程与异步串口接收数据过程相同；如果从机解析受保护 ID 段后被要求发送数据，则忽略下面步骤 12 和 13，设 USART_CR1.TE=1 并参考【LIN 发送】的步骤 23~31 操作）

- Step11. 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；
- Step12. 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；
- Step13. 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；
- Step14. 接收结束，进入休眠状态，等待下次被唤醒。

17.4 寄存器描述

17.4.1 寄存器列表

表 17-10 USART 基地址列表

名称	基地址	描述
USART0	0x4000 0000	USART0基地址
USART1	0x4000 0400	USART1基地址
USART2	0x4000 0800	USART2基地址
USART3	0x4000 0C00	USART3基地址

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。

表 17-11 USART 寄存器列表

寄存器名	偏移地址	复位值
控制寄存器1 (USART_CR1)	0x00	0x0000 0000
控制寄存器2 (USART_CR2)	0x04	0x0000 0000
控制寄存器3 (USART_CR3)	0x08	0x0000 0000
波特率寄存器 (USART_BRR)	0x0C	0x0000 FF7F
定时计数寄存器 (USART_TOR)	0x10	0x0000 0000
状态寄存器 (USART_SR)	0x14	0x0000 0001
发送数据寄存器 (USART_TDR)	0x18	0x0000 01FF
中断使能寄存器 (USART_IER)	0x1C	0x0000 0000
中断标志清除寄存器 (USART_ICR)	0x20	0x0000 7A7E
接收数据寄存器 (USART_RDR)	0x24	0x0000 0000
地址寄存器 (USART_ADDR)	0x28	0x0000 0000
地址掩码寄存器 (USART_ADDMR)	0x2C	0x0000 0000
波特率检出值寄存器 (USART_BDR)	0x30	0x0000 0000
红外调制寄存器 (USART_IR)	0xF0	0x0000 0000

17.4.2 控制寄存器 1 (USART_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	SLME	MPE	COMB	SBS	TOE		OVER8	MSB	STOP	CHLEN	PCE	PS	TE	RE	

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14	SLME	多处理器通信时, 静默模式使能位	0: 正常模式 1: 静默模式 注: 1) 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。 2) 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
b13	MPE	多处理器通信功能使能位 (仅适用于异步串口模式)	0: 无效 1: 有效 注: 1) 非异步串口模式须设置为0。 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b12	COMB	异步串口模式时, 收发端口复合使能位	0: TXD和RXD分开使用 1: 异步串口单总线模式, TXD和RXD合用TXD端口 注: 1) 非异步串口模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b11	SBS	异步串口模式接收数据时, 起始位检测方式设定位	0: 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 1: 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 注: 1) 非异步串口模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b10~b8	TOE	多功能定时器工作模式配置位	000: 停止定时计数器 001: 通用定时功能 010: 接收定时功能 011: 数据帧间定时功能 100: LIN模式下, 同步间隔段计数使能 101: LIN模式或异步串口模式下, 波特率自动检测使能 110~111: 保留, 同000, 关闭计数功能 注: 1) 所有功能的TOE都由硬件清零 2) 如果从一种功能 (如TOE=010) 切换到另一种功能 (如TOE=001), 则需要软件先写TOE=000后再写TOE=001	RW

b7	OVER8	异步串口模式/LIN模式时，采样模式设定位	0: 16倍采样 1: 8倍采样 注： 1) 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 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b6	MSB	MSB/LSB方式选择位	0: LSB方式 1: MSB方式 注： 1) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b5	STOP	停止位长度设定位	0: 1个停止位 1: 2个停止位 注： 1) 同步串口模式和LIN模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b4	CHLEN	发送/接收数据长度设定位	0: 8位 1: 9位 注： 1) 同步串口模式和LIN模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b3	PCE	奇偶校验有效位	0: 无校验 1: 校验 注： 1) 同步串口模式和LIN模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b2	PS	奇偶校验选择位	0: 偶校验 1: 奇校验 注： 1) PS位只在PCE=1时有效 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位。	RW
b1	TE	发送使能位	0: 发送禁止 1: 发送使能 注： 1) 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 2) 单线半双工工作时TE设置无效	RW
b0	RE	接收使能位	0: 接收禁止 1: 接收使能 注： 1) 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

17.4.3 控制寄存器 2 (USART_CR2)

复位值: 0x0000 0000

注: [11:10]为保留位, 写入数据必须为复位值

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						CLKC	SWAP	SYNC	TXINV	RXINV	CTSEN	RTSEN	TXDMAEN	RXDMAEN	

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9~b8	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 00: 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 01: 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 同步串口模式 00, 01: 时钟源为内部波特率生成器生成的时钟, 输出到CK管脚 10, 11: 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 00: 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 01: 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 10, 11: 设定禁止 注: 1) 红外模式和LIN模式设置无效, CK都不能输出到CK管脚 2) 只有在 USART_CR1.TE 和 USART_CR1.RE 同时为 0 时才能写此位	RW
			0: 不互换 1: 互换 注: 1) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
			0: 异步串口模式 1: 同步串口模式 注: 1) 异步串口模式、智能卡模式和LIN模式须写入0 2) 同步串口模式须写入1 3) 红外模式设置无效 4) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
			0: 不反相 1: TXD输出管脚信号反相 注: 1) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b4	RXINV	输入信号反相控制位	0: 不反相 1: RXD输入管脚信号反相 注:	RW

			1) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	
b3	CTSEN	异步串口模式/同步串口模式时，CTS功能使能位	0: 禁止 1: 使能 注： 1) 非异步串口模式和非同步串口模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b2	RTSEN	异步串口模式/同步串口模式时，RTS功能使能位	0: 禁止 1: 使能 注： 1) 非异步串口模式和非同步串口模式须设置为0 2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位	RW
b1	TXDMAEN	TX DMAC硬件握手信号使能位	0: 关闭硬件握手信号 1: 打开硬件握手信号 注： 1) 单线半双工工作时不支持DMA模式	RW
b0	RXDMAEN	RX DMAC硬件握手信号使能位	0: 关闭硬件握手信号 1: 打开硬件握手信号 注： 1) 单线半双工工作时不支持DMA模式	RW

17.4.4 控制寄存器 3 (USART_CR3)

复位值: 0x0000 0000

注: [12:10]为保留位, 写入数据必须为复位值

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IRMOD	IRCW	IREN	Res			SMCPU		BCN			SCE	BRE	ABR	LBDE	LINE

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15	IRMOD	红外模式时, 38K调制方式	0: 数据0时输出38K载波 1: 数据1时输出38K载波 注: 1) 非红外模式设置无效	RW
b14	IRCW	红外模式时, 红外载波选择	0: $\frac{3}{16}$ 模式 1: 38K载波模式 注: 1) 非红外模式设置无效	RW
b13	IREN	红外模式时, 红外模式使能位	0: 红外功能模块关闭 1: 红外功能模块开启 注: 1) 非红外模式须设置为0	RW
b12~10	Res	保留位	保留, 必须保持复位值	
b9~8	SMCPU	智能卡模式时, 输出边沿整形	00: 开漏时输出无边沿整形 01: 开漏时强输出1个PCLK周期CMOS高电平 10: 开漏时强输出2个PCLK周期CMOS高电平 11: 开漏时强输出4个PCLK周期CMOS高电平 注: 1) 非智能卡模式设置无效	RW
b7~5	BCN	智能卡模式时, 设定一位数据传输期间的基本时钟数	000: 16 001: 32 010: 64 011: 128 100: 186 101: 256 110: 372 111: 512 注: 1) 非智能卡模式设置无效	RW
b4	SCEN	智能卡模式时, 模式使能位	0: 禁止智能卡模式 1: 使能智能卡模式 注: 1) 非智能卡模式须设置为0	RW
b3	BREAK	LIN模式时, 发送同步间隔段使能	1: 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 0: 写入USART_TDR寄存器的值代表正常发送的数据 注:	RW

			1) 非LIN模式须设置为0	
b2	ABRE	异步串口模式/LIN模式时，自动波特率检测使能	0: 停止检测波特率 1: 启动检测波特率 注： 1) 非异步串口模式和非LIN模式须设置为0	RW
b1	LBDE	LIN模式时，同步间隔段检测使能位	0: 禁止同步间隔段检测 1: 允许同步间隔段检测 注： 1) 非LIN模式须设置为0	RW
b0	LINEN	LIN模式时，模式使能位	0: 禁止LIN模式 1: 使能LIN模式 注： 1) 非LIN模式须设置为0	RW

17.4.5 波特率寄存器 (USART_BRR)

复位值: 0x0000 FF7F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														PSC	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIVI								FBME	DIVF						

位	标记	位名	功能	读写
b31~b18	Res	保留位	保留, 必须保持复位值	
b17~b16	PSC	内部时钟源时, 预分频器选择位	00: PCLK	RW
			01: PCLK/4	
			10: PCLK/16	
			11: PCLK/64	
b15~b8	DIVI	整数分频	整数分频寄存器	RW
b7	FBME	内部波特率生成器选择时, 小数波特率生成使能位	0: 禁止 1: 使能	RW
b6~b0	DIVF	小数分频	小数分频寄存器	RW
			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。	

17.4.6 定时计数寄存器 (USART_TOR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												TOR[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TOR[15:0]															

位	标记	位名	功能	读写
b31~b20	Res	保留位	保留, 必须保持复位值	
b19~b0	TOR	定时计数时间	定时计数时间= $t_{PCLK} * 4^{PSC} * (TOR+1)$	RW

17.4.7 状态寄存器 (USART_SR)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DWK	LBD	ABR	CTS	CTSV	AM	MPB	Res	TO	OR	FE	PE	RC	TC	TXE

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14	DWK	异步串口模式/红外模式/LIN模式时, RXD线唤醒标志	硬件置1, 软件清零 0: 未发生唤醒 1: 已发生唤醒	RO
b13	LBD	LIN模式时, 同步间隔段检测标志	硬件置1, 软件清零 0: 未检测到同步间隔段 1: 已检测到同步间隔段	RO
b12	ABR	异步串口模式/LIN模式时, 自动波特率检测标志	硬件置1, 软件清零 0: 检测未完成 1: 检测已完成	RO
b11	CTS	异步串口模式/同步串口模式时, CTS中断标志位	硬件置1, 软件清零 0: CTS信号没有发生翻转 1: CTS信号发生翻转	RO
b10	CTSV	异步串口模式/同步串口模式时, CTS管脚当前电平值	0: CTS管脚为低电平 1: CTS管脚为高电平	RO
b9	AM	多处理器通信时, 地址匹配标志	0: 地址不匹配 1: 地址匹配	RO
b8	MPB	多处理器通信时, 多处理器标志位	0: 接收数据为数据 1: 接收数据为ID 注: 1) MPB位只在多处理器通信时有效	RO
b7	Res	保留位	保留, 必须保持复位值	
b6	TO	定时器匹配或溢出标志	0: 定时器没有匹配或溢出 1: 定时器已经匹配或溢出	RO
b5	OR	上溢错误标志	硬件置1, 软件清零 0: 没有上溢 1: 发生上溢 注: 1) OR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。 2) RE=0并不能复位OR位。 3) OR=1后接收到的数据会覆盖前面的数据。	RO
b4	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	硬件置1, 软件清零 0: 无接收帧错误 1: 发生接收帧错误 注: 1) 异步串口模式/红外模式/LIN模式下置位条件: 接收的数据帧的停	RO

			止位为0。 2) 智能卡模式下置位条件：采样到低电平错误信号标志。 3) RE=0并不能复位FE位。 4) FE=1后接收到的数据会覆盖前面的数据。	
b3	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	硬件置1，软件清零 0：无校验错误 1：发生校验错误 注： 1) PE置位条件：接收数据发生奇偶校验错误。 2) RE=0并不能复位PE位。 3) PE=1后接收到的数据会覆盖前面的数据。	RO
b2	RC	数据接收完成标志	0：非接收完成状态 1：接收已完成 注： 1) 接收到停止位后被硬件置1，数据被软件读取后自动清零，软件也可以写USART_ICR.RC=0来清零。	RO
b1	TC	发送完成标志位	硬件置1，软件清零 0：发送数据中 1：发送数据完成 注： 1) 异步模式和时钟同步模式下置位条件：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）。 TC清零条件：向USART_ICR.TC写0。 2) 智能卡模式下置位条件：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。 TC清零条件：向USART_ICR.TC写0。	RO
b0	TXE	发送数据寄存器空标志	0：数据未传输到移位寄存器 1：数据传输到移位寄存器 注： 1) TXE由硬件置1和清0：数据传输到移位寄存器后硬件置1，数据写入发送数据寄存器后硬件清0。	RO

17.4.8 发送数据寄存器 (USART_TDR)

复位值: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						MPID	TDR								

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	MPID	多处理器ID位	0: 发送数据 1: 发送ID 注: 1) MPID位只在多处理器通信时有效,其它模式须设定为0。	RW
b8~b0	TDR	发送数据	发送数据 注: 1) 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

17.4.9 中断使能寄存器 (USART_IER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DWK	LBD	ABR	CTS	Res	AM	Res	TO	OR	FE	PE	RC	TC	TXE	

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14	DWK	异步串口模式/红外模式/LIN模式时, RXD管脚边沿中断使能	RXD管脚边沿中断使能 (USART_CR2.RXINV=0为下降沿, USART_CR2.RXINV=1为上升沿) 0: 中断禁止 1: 中断允许	RW
b13	LBD	LIN模式时, 同步间隔段中断使能	0: 中断禁止 1: 中断允许	RW
b12	ABR	异步串口模式/LIN模式时, 自动波特率检测的中断使能	0: 中断禁止 1: 中断允许	RW
b11	CTS	异步串口模式/同步串口模式时, CTS信号翻转中断使能	0: 中断禁止 1: 中断允许	RW
b10	Res	保留位	保留, 必须保持复位值	
b9	AM	多机通信模式下, 地址匹配后的中断使能	0: 中断禁止 1: 中断允许	RW
b8~b7	Res	保留位	保留, 必须保持复位值	
b6	TO	定时/计数器溢出中断使能	0: 中断禁止 1: 中断允许	RW
b5	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	0: 中断禁止 1: 中断允许	RW
b4	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	0: 中断禁止 1: 中断允许	RW
b3	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	0: 中断禁止 1: 中断允许	RW
b2	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	0: 中断禁止 1: 中断允许	RW
b1	TC	一帧数据发送完成的中断请求使能位	0: 中断禁止 1: 中断允许	RW
b0	TXE	发送缓存为空状态的中断使能	0: 中断禁止 1: 中断允许	RW

17.4.10 中断标志清除寄存器 (USART_ICR)

复位值: 0x0000 7A7E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DWK	LBD	ABR	CTS	Res	AM	Res	TO	OR	FE	PE	RC	TC	Res	

位	标记	位名	功能	读写
b31~b15	Res	保留位	保留, 必须保持复位值	
b14	DWK	DWK清零位	0: 清零DWK 1: 不影响DWK值	WO
b13	LBD	LBD清零位	0: 清零LBD 1: 不影响LBD值	WO
b12	ABR	ABR清零位	0: 清零ABR 1: 不影响ABR值	WO
b11	CTS	CTS清零位	0: 清零CTS 1: 不影响CTS值	WO
b10	Res	保留位	保留, 必须保持复位值	
b9	AM	AM清零位	0: 清零AM 1: 不影响AM值	WO
b8~b7	Res	保留位	保留, 必须保持复位值	
b6	TO	TO清零位	0: 清零TO 1: 不影响TO值	WO
b5	OR	OR清零位	0: 清零OR 1: 不影响OR值	WO
b4	FE	FE清零位	0: 清零FE 1: 不影响FE值	WO
b3	PE	PE清零位	0: 清零PE 1: 不影响PE值	WO
b2	RC	RC清零位	0: 清零RC 1: 不影响RC值	WO
b1	TC	TC清零位	0: 清零TC 1: 不影响TC值	WO
b0	Res	保留位	保留, 必须保持复位值	

17.4.11 接收数据寄存器 (USART_RDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								RDR							
位	标记	位名	功能										读写		
b31~b9	Res	保留位	保留, 必须保持复位值												
b8~b0	RDR	接收数据	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		

17.4.12 地址寄存器 (USART_ADDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ADDR							
位	标记	位名	功能										读写		
b31~b8	Res	保留位	保留, 必须保持复位值												
b7~b0	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		

17.4.13 地址掩码寄存器 (USART_ADDMR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ADDM							

位	标记	位名	功能	读写
b31~b8	Res	保留位	保留，必须保持复位值	
b7~b0	ADDM	从机地址掩码	0: 对应位不参与匹配	RW
			1: 对应位参与匹配	
			注:	
			1) 从机地址=USART_ADDR.ADDR & USART_ADDMR.ADDM。	
			2) 只有在USART_CR1.TE和USART_CR1.RE同时为0时才能写此位。	

17.4.14 波特率检出值寄存器 (USART_BDR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				BDR											

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11~b0	BDR	异步串口模式/LIN模式时, 波特率检出值	通过如下公式可计算出通信波特率 $\text{Baud} = \frac{f_{\text{PCLK}}}{4^{\text{PSC}} \times (\text{BDR} + 1)}$ 注: 1) 仅适用于LIN模式或异步串口模式。	RO

17.4.15 红外调制寄存器 (USART_IR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					DCNTH										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					DCNTL										

位	标记	位名	功能	读写
b31~b27	Res	保留位	保留, 必须保持复位值	
b26~b16	DCNTH	红外模式时, 高电平计数值	38K高电平维持时间 $t = t_{\text{PCLK}} * (\text{DCNTH} + 1)$	RW
b15~b11	Res	保留位	保留, 必须保持复位值	
b10~b0	DCNTL	红外模式时, 低电平计数值	38K低电平维持时间 $t = t_{\text{PCLK}} * (\text{DCNTL} + 1)$	RW

18 循环冗余校验（CRC）

18.1 概述

循环冗余校验（CRC）计算单元将数据流或数据块作为输入，在生成多项式的控制下生成一个输出数。该输出数常用于验证数据传输或存储的正确性和完整性。本模块支持计算 CRC 值和检验 CRC 值。

18.2 主要特性

- 一种执行标准：ISO/IEC13239
- 两种编码方式：CRC-16、CRC-32
- 三种写入位宽：8bit、16bit、32bit
- 两种工作模式：CRC 编码模式、CRC 校验模式
- CRC-16/ X25 多项式： $x^{16}+x^{12}+x^5+1$
- CRC-32 多项式： $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

18.3 功能描述

18.3.1 工作模式

本模块支持两种工作模式：CRC 编码模式、CRC 校验模式。

CRC 编码模式是指向 CRC 模块输入一定数量的原始数据,获取 CRC 模块生成的输出值(CRC_RESULT)。

CRC 校验模式是指向 CRC 模块输入一定数量的已编码的数据序列(原始数据+CRC 校验值),验证原始数据与 CRC 校验值是否匹配(CRC_CSR.FLAG)。

18.3.2 编码方式

本模块支持两种编码方式 CRC-16、CRC-32,其计算结果分别为 16 比特和 32 比特。通过 CRC_CSR.CODE 配置需要使用的编码方式。

CRC-16 多项式: $x^{16}+x^{12}+x^5+1$ 。

CRC-32 多项式: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ 。

18.3.3 写入位宽

本模块支持三种写入位宽: 8bit、16bit、32bit。不同位宽的写入需要符合“位宽一致、先低后高”的原则,即“每次写入数据,都必须写入到与本次有效数据位宽相等的寄存器中,并且较低位数据的写入先于较高位数据”。

下方展示了同一序列数据采用三种位宽进行写入的方法,其输出结果相同。

- 8bit 位宽写入: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16bit 位宽写入: 0x1100, 0x3322, 0x5544, 0x7766
- 32bit 位宽写入: 0x33221100, 0x77665544

18.4 编程示例

18.4.1 CRC-16 编码模式

- Step1. 向 CRC_CSR.CODE 写入 0x00，选择 CRC-16；
- Step2. 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算；
- Step3. 将待编码的原始数据依次写入 CRC_DR 寄存器，写入位宽可选择 8bit、16bit、32bit；
- Step4. 查询 CRC_CSR.BUSY，直到 CRC_CSR.BUSY 变为零；
- Step5. 读取 CRC_RESULT[15:0]以获取 CRC-16 的计算结果。

18.4.2 CRC-16 检验模式

- Step1. 向 CRC_CSR.CODE 写入 0x00，选择 CRC-16；
- Step2. 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算；
- Step3. 将已编码的数据序列（原始数据+CRC 校验值）依次写入 CRC_DR 寄存器，写入位宽可选择 8bit、16bit、32bit；
- Step4. 查询 CRC_CSR.BUSY，直到 CRC_CSR.BUSY 变为零；
- Step5. 根据 CRC_CSR.FLAG 的值判定已编码的数据序列是否被篡改。

18.4.3 CRC-32 编码模式

- Step1. 向 CRC_CSR.CODE 写入 0x01，选择 CRC-32；
- Step2. 向 CRC_RESULT[31:0]写入 0xFFFFFFFF，初始化 CRC 计算；
- Step3. 将待编码的原始数据依次写入 CRC_DR 寄存器，写入位宽可选择 8bit、16bit、32bit；
- Step4. 查询 CRC_CSR.BUSY，直到 CRC_CSR.BUSY 变为零；
- Step5. 读取 CRC_RESULT[31:0]以获取 CRC-32 的计算结果。

18.4.4 CRC-32 检验模式

- Step1. 向 CRC_CSR.CODE 写入 0x01，选择 CRC-32；
- Step2. 向 CRC_RESULT[31:0]写入 0xFFFFFFFF，初始化 CRC 计算；
- Step3. 将已编码的数据序列（原始数据+CRC 校验值）依次写入 CRC_DR 寄存器，写入位宽可选择 8bit、16bit、32bit；
- Step4. 查询 CRC_CSR.BUSY，直到 CRC_CSR.BUSY 变为零；

Step5. 根据 CRC_CSR.FLAG 的值判定已编码的数据序列是否被篡改。

18.5 寄存器描述

18.5.1 寄存器列表

基地址：0x4002 0800

表 18-1 CRC 寄存器列表

寄存器名	偏移地址	复位值
控制和状态寄存器 (CRC_CSR)	0x00	0x0000 0001
结果寄存器 (CRC_RESULT)	0x04	0x0000 0000
数据寄存器 (CRC_DR)	0x80	0x0000 0000

18.5.2 控制和状态寄存器 (CRC_CSR)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													BUSY	FLAG	CODE

位	标记	位名	功能	读写
b31~b3	Res	保留位	保留, 必须保持复位值	
b2	BUSY	CRC 忙标志	0: 当前 CRC 空闲或计算完成 1: 当前CRC忙碌或正在计算	RO
b1	FLAG	CRC 校验结果	0: 当前 CRC 校验错误 1: 当前CRC校验正确	RO
b0	CODE	CRC 编码方式选择	0: CRC-16 编码 1: CRC-32编码	RW

18.5.3 结果寄存器 (CRC_RESULT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RESULT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULT[15:0]															

位	标记	位名	功能	读写
b31~b0	RESULT	CRC 计算结果	读取 RESULT[15:0]以获取 CRC-16 的计算结果 读取 RESULT[31:0]以获取 CRC-32 的计算结果 向 RESULT[15:0]写入 0xFFFF 以初始化 CRC-16 计算 向RESULT[31:0]写入0xFFFFFFFF以初始化CRC-32计算	RW

18.5.4 数据寄存器（CRC_DR）

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DATA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DATA[15:0]															
位	标记	位名	功能										读写		
b31~b0	DATA	CRC 写入数据	本寄存器用于写入需要运算的数据，支持 3 种写入位宽 8bit 写入方式： * ((uint8_t *)0x40020880) = 0xXX 16bit 写入方式： * ((uint16_t *)0x40020880) = 0xFFFF 32bit 写入方式： * ((uint32_t *)0x40020880) = 0xFFFFFFFF										WO		

19 模数转换器 (ADC)

19.1 模块简介

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。内部集成了 2 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。具有以下特性：

- 12 位转换精度
- 1M SPS 转换速度
- 18 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
- 4 种参考源：AVCC 电压、ExRef 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
- 多种转换模式：单次转换、顺序扫描单次转换、顺序扫描连续转换、顺序扫描断续转换、顺序扫描单次转换累加
- 输入通道电压阈值监测
- 软件可配置 ADC 的转换速率
- 内置信号跟随器，可转换高阻信号
- 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

19.2 ADC 框图

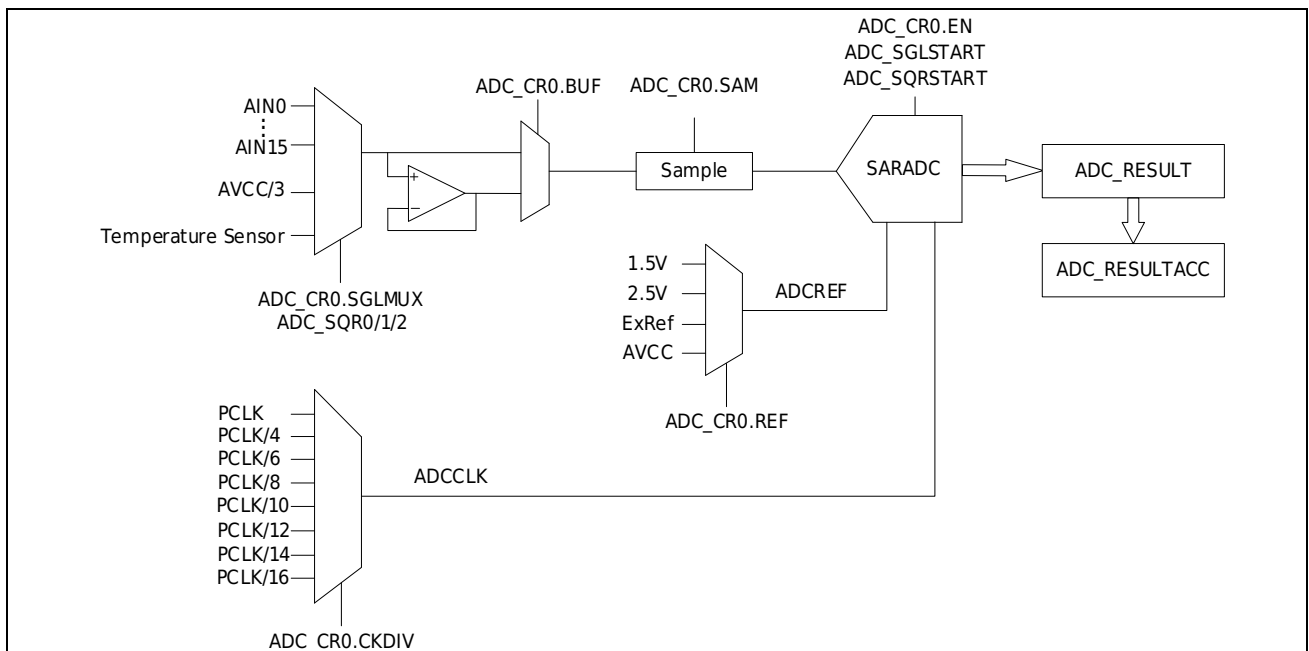


图 19-1 ADC 示意图

19.3 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 转换由转换过程及逐次比较过程组成。其中转换过程需要 3~10 个 ADCCLK，由 ADC_CR0.SAM 配置；逐次比较过程需要 15 个 ADCCLK。所以，一次 ADC 转换共需要 18~25 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。

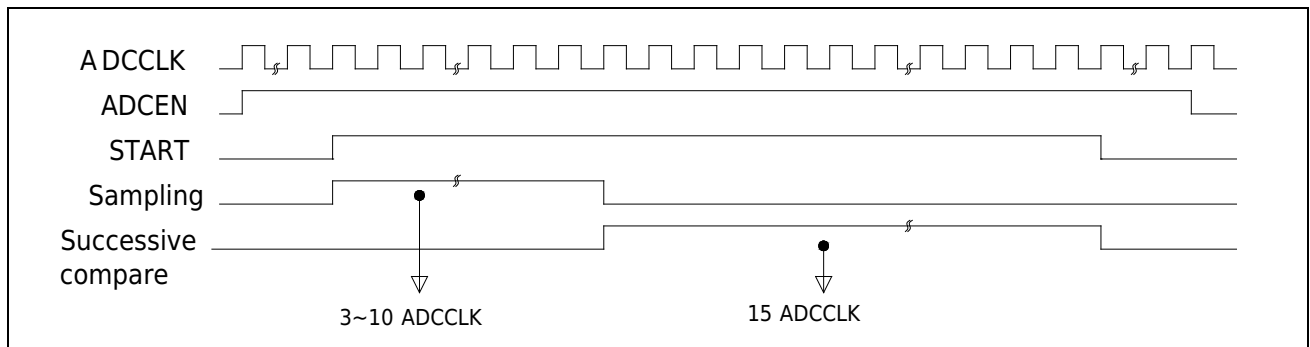


图 19-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关，最高转换速度如下表所示：

表 19-1 ADC 最高转换速度

ADC 参考电压	AVCC 电压	最高转换速度	最大 ADCCLK 频率
内部 1.5V	1.8~5.5V	200K SPS	5MHz
内部 2.5V	2.8~5.5V	200K SPS	5MHz
AVCC / ExRef	1.8~2.4V	200K SPS	5MHz
AVCC / ExRef	2.4~2.7V	500K SPS	12.5MHz
AVCC / ExRef	2.7~5.5V	1M SPS	25MHz

19.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 19 路 ADC 通道进行转换。该模式既可通过设置 ADC_SGLSTART.START 位启动也可通过设置 ADC_EXTTRIG0 的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_IFR.ESG 位会自动置 1，转换结果保存在 ADC_RESULT 寄存器中。

通过 ADC_SGLSTART.START 位启动 ADC 单次转换操作流程：

Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；

Step2. 设置 GPIOB_ADS.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；

注：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。

Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；

- Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；
- Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；
- Step6. 设置 ADC_CR1.Mode 为 0，选择单次转换模式；
- Step7. 配置 ADC_CR0.REF，选择 ADC 的参考电压；
- Step8. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；
- Step9. 配置 ADC_CR0.SGLMUX，选择待转换的通道；
- Step10. 设置 ADC_ICR.ESG 为 0，清除 ADC_IFR.ESG 标志；
- Step11. 设置 ADC_SGLSTART.START 为 1，启动 ADC 单次转换；
- Step12. 等待 ADC_IFR. ESG 变为 1，读取 ADC_RESULT 寄存器以获取 ADC 转换结果；
- Step13. 如需对其它通道进行转换，重复执行 Step9~Step12；
- Step14. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 单次转换操作流程：

- Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；
 - Step2. 设置 GPIOB_ADS.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；
- 注：**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。
- Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；
 - Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；
 - Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；
 - Step6. 设置 ADC_CR1.MODE 为 0，选择单次转换模式；
 - Step7. 设置 ADC_IER.ESG 为 1，使能 ADC 中断；
 - Step8. 使能 NVIC 中断向量表中的 ADC 中断；
 - Step9. 配置 ADC_CR0.REF，选择 ADC 的参考电压；
 - Step10. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；
 - Step11. 配置 ADC_CR0.SGLMUX，选择待转换的通道；
 - Step12. 设置 ADC_ICR 为 0x0，清除 ADC 中断标志；
 - Step13. 配置 ADC_EXTTRIGGER0，选择外部触发条件；

Step14. 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；

Step15. 如需对其它通道进行转换，重复执行 Step11~Step14；

Step16. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

19.5 扫描转换模式

扫描转换模式分为顺序扫描单次转换，顺序扫描连续转换和顺序扫描断续转换三种模式。三种模式各自单独工作时，均可连续对多个通道进行多次转换。

19.5.1 顺序扫描单次转换模式

顺序扫描单次转换模式最多可进行 16 次连续转换，转换的总次数由 ADC_SQR2.CNT 进行配置；可配置所有 20 个通道进行转换，待转换通道由 ADC_SQRx.CHxMUX 进行配置。该模式既可通过设置 ADC_SQRSTART.START 位启动也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。启动转换后，ADC 模块依次转换 CH0MUX~CHxMUX 中配置的通道直到总转换次数完成。每次转换后转换结果保存在 ADC_RESULT 寄存器中，ADC_IFR.EOC 位会自动置 1。转换结果需要及时读出。ADC 模块完成总转换次数后，ADC_IFR.EOS 位会自动置 1。下图演示了对 AIN0、AIN1、AIN5 进行 8 次转换的顺序扫描转换过程。其中顺序扫描转换通道 7，4，1 配置为 AIN0，转换通道 6，3，0 配置为 AIN1，转换通道 5，2 配置为 AIN5。ADC_SQRSTART.START 置 1 后，ADC 模块会依次对顺序扫描转换通道 0~7 进行转换。

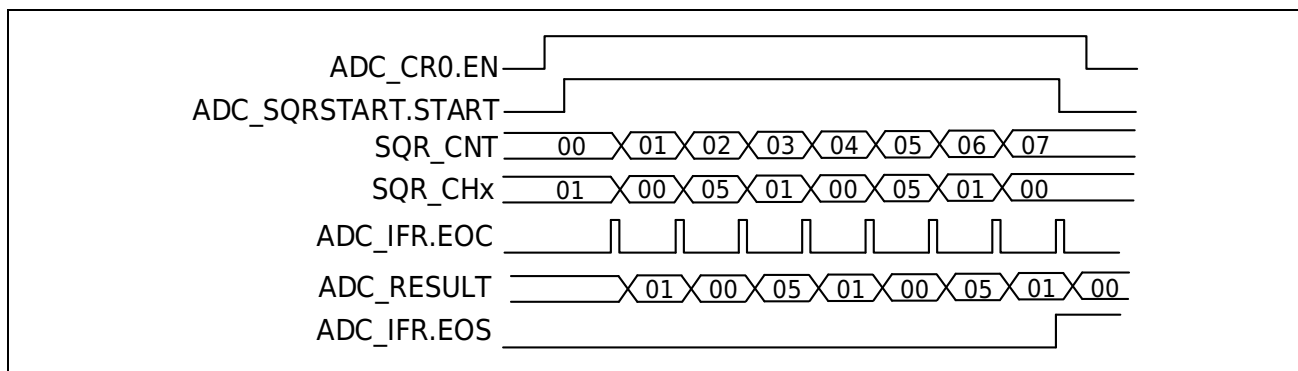


图 19-3 ADC 顺序扫描转换过程示例

通过 ADC_SQRSTART.START 位启动 ADC 顺序扫描转换操作流程：

Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；

Step2. 设置 GPIOB_ADS.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；

注：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。

Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；

Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；

- Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；
- Step6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式；
- Step7. 配置 ADC_CR0.REF，选择 ADC 的参考电压；
- Step8. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；
- Step9. 配置 ADC_SQRx.CHxMUX，选择顺序扫描转换通道；
- Step10. 配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数；
- 注：**总转换次数需与 Step9 中配置的转换通道数保持一致。
- Step11. 设置 ADC_ICR.EOS/EOC 为 0，清除 ADC_IFR.EOC/EOS 标志；
- Step12. 设置 ADC_SQRSTART.START 为 1，启动 ADC 顺序扫描转换；
- Step13. 等待 ADC_IFR.EOC 变为 1，读取 ADC_RESULT 寄存器以获取相应通道的转换结果。等待 ADC_IFR.EOS 变为 1 结束顺序扫描转换；
- Step14. 如需对其它通道进行转换，重复执行 Step9~Step13；
- Step15. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 顺序扫描转换操作流程：

- Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；
- Step2. 设置 GPIOB_ADS.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；
- 注：**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。
- Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；
- Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；
- Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；
- Step6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式；
- Step7. 设置 ADC_CR0.IE 为 1，使能 ADC 中断；
- Step8. 使能 NVIC 中断向量表中的 ADC 中断；
- Step9. 配置 ADC_CR0.REF，选择 ADC 的参考电压；
- Step10. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；
- Step11. 配置 ADC_SQRx.CHxMUX，选择顺序扫描转换通道；
- Step12. 配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数；

注：总转换次数需与 Step11 中配置的转换通道数保持一致。

Step13. 设置 ADC_IFR 为 0x0，清除 ADC 中断标志；

Step14. 配置 ADC_EXTTRIGGER0，选择外部触发条件；

Step15. 当外部触发条件触发 ADC 完成转换时，ADC 模块会产生 EOC 中断。用户可在 ADC 中断服务程序中读取 ADC_RESULT 寄存器以获取相应通道的转换结果。当 EOS 中断后当前序列转换结束；

Step16. 如需对其它通道进行转换，重复执行 Step12~Step15；

Step17. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

19.5.2 顺序扫描连续转换模式

顺序扫描连续转换模式可以对 ADC 连续转换，单次转换次数由 ADC_SQR2.CNT 进行配置；可配置选择任意通道进行转换，待转换通道由 ADC_SQRx.CHxMUX 进行配置。该模式既可通过设置 ADC_SQRSTART.START 位启动也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。启动转换后，ADC 模块依次转换 CH0MUX~CHxMUX 中配置的通道转换，一个通道转换完成后 ADC_IFR.EOC 位会自动置 1，转换结果保存在 ADC_RESULT 寄存器中，然后切换到下一通道，ADC 模块完成 ADC_SQR2.CNT 配置的转换次数后，ADC_IFR.EOS 位会自动置 1。并自动启动下一次转换，直到通过软件写 0 到 ADC_SQRSTART.START，转换完当前正在转换的通道后停止。

19.5.3 顺序扫描断续转换模式

顺序扫描断续转换模式，可以对 ADC_SQR2.CNT 和 ADC_SQRx.CHxMUX 进行配置的通道依次转换。该模式既可通过设置 ADC_SQRSTART.START 位启动也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。每次启动进行一个通道转换，然后等待下次启动转换下个通道，ADC_IFR.EOC 位会自动置 1，转换结果保存在 ADC_RESULT 寄存器中。

19.5.4 扫描转换触发 DMA 读取

顺序扫描转换完成后，可自动触发 DMA 读取转换结果。顺序扫描模式下，通过配置 ADC_IER.DEOC 为 1 使能该功能。

19.6 连续转换累加模式

在连续单次转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可配置所有 19 个通道进行转换。ADC 转换的总次数由 ADC_SQR2.CNT 进行配置；待转换通道由 ADC_SQRx.CHxMUX 进行配置。该模式既可通过设置 ADC_SQRSTART.START 位启动也可通过设置 ADC_EXTTRIGGER0 的外部触发启动。启动连续转换后，ADC 模块依次转换 CHxMUX~CH0MUX 中配

置的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.EOS 位会自动置 1，转换结果的累加值保存在 ADC_RESULTACC 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。顺序扫描转换通道 9, 6, 3, 0 配置为 AIN0，转换通道 8, 5, 2 配置为 AIN5，转换通道 7, 4, 1 配置为 AIN1。ADC_SQRSTART.START 置 1 后，ADC 模块会按照顺序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。

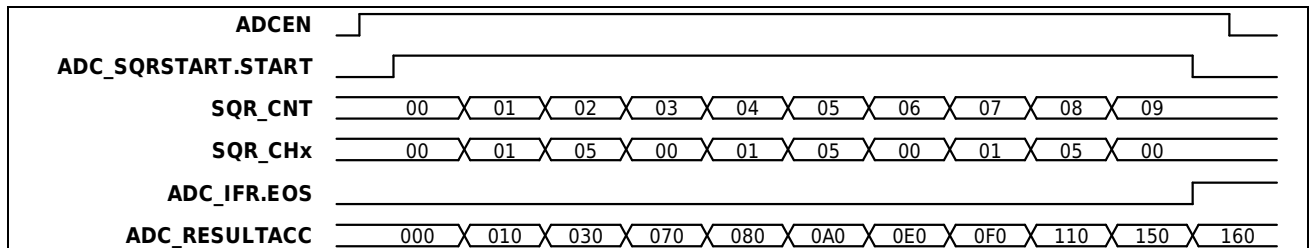


图 19-4 ADC 连续转换累加过程示例

通过 ADC_SQRSTART.START 位启动 ADC 连续转换累加操作流程：

- Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；
- Step2. 设置 GPIOB_ADS.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；
- 注：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。
- Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；
- Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；
- Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；
- Step6. 设置 ADC_CR1.MODE 为 1，选择扫描转换模式；
- Step7. 设置 ADC_CR1.RACCEN 为 1，使能 ADC 转换自动累加功能；
- Step8. 配置 ADC_CR0.REF，选择 ADC 的参考电压；
- Step9. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；
- Step10. 配置 ADC_SQRx.CHxMUX，选择顺序扫描转换通道；
- Step11. 配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数；
- 注：总转换次数需与 Step10 中配置的转换通道数保持一致。
- Step12. 设置 ADC_ICR.EOS 为 0，清除 ADC_IFR.EOS 标志；
- Step13. 设置 ADC_CR1.RACCCLR 为 0，清零 ADC_RESULTACC 寄存器；
- Step14. 设置 ADC_SQRSTART.START 为 1，启动 ADC 顺序扫描转换；

Step15. 等待 ADC_IFR.EOS 变为 1，读取 ADC_RESULTACC 寄存器以获取转换结果累加值；

Step16. 如需对其它通道进行转换，重复执行 Step11~Step15；

Step17. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 连续转换累加操作流程：

Step1. 配置 GPIOx_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口；

Step2. 设置 GPIOBADs.bit1 为 1，将 ADC 外部参考电压引脚（PB01）配置为模拟端口；

注：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。

Step3. 设置 ADC_BGR.BGR_EN 为 1，使能 BGR 模块；

Step4. 设置 ADC_CR0.EN 为 1，使能 ADC 模块；

Step5. 延时 20us，等待 ADC 及 BGR 模块启动完成；

Step6. 设置 ADC_CR1.MODE 为 1，选择扫描单次转换模式；

Step7. 设置 ADC_IER.EOS 为 1，使能 ADC 中断；

Step8. 使能 NVIC 中断向量表中的 ADC 中断；

Step9. 设置 ADC_CR1.RACCEN 为 1，使能 ADC 转换自动累加功能；

Step10. 配置 ADC_CR0.REF，选择 ADC 的参考电压；

Step11. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV，设置 ADC 的转换速度；

Step12. 配置 ADC_SQRx.CHxMUX，选择顺序扫描转换通道；

Step13. 配置 ADC_SQR2.CNT，选择顺序扫描转换的总转换次数；

注：总转换次数需与 Step12 中配置的转换通道数保持一致。

Step14. 设置 ADC_IFR 为 0x0，清除 ADC 中断标志；

Step15. 设置 ADC_CR1.RACCCLR 为 0，清零 ADC_RESULTACC 寄存器；

Step16. 配置 ADC_EXTTRIGGER0，选择外部触发条件；

Step17. 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；

Step18. 如需对其它通道进行转换，重复执行 Step12~Step17；

Step19. 设置 ADC_CR0.EN 及 ADC_BGR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

19.7 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 单次转换或顺序扫描转换的外部触发源。

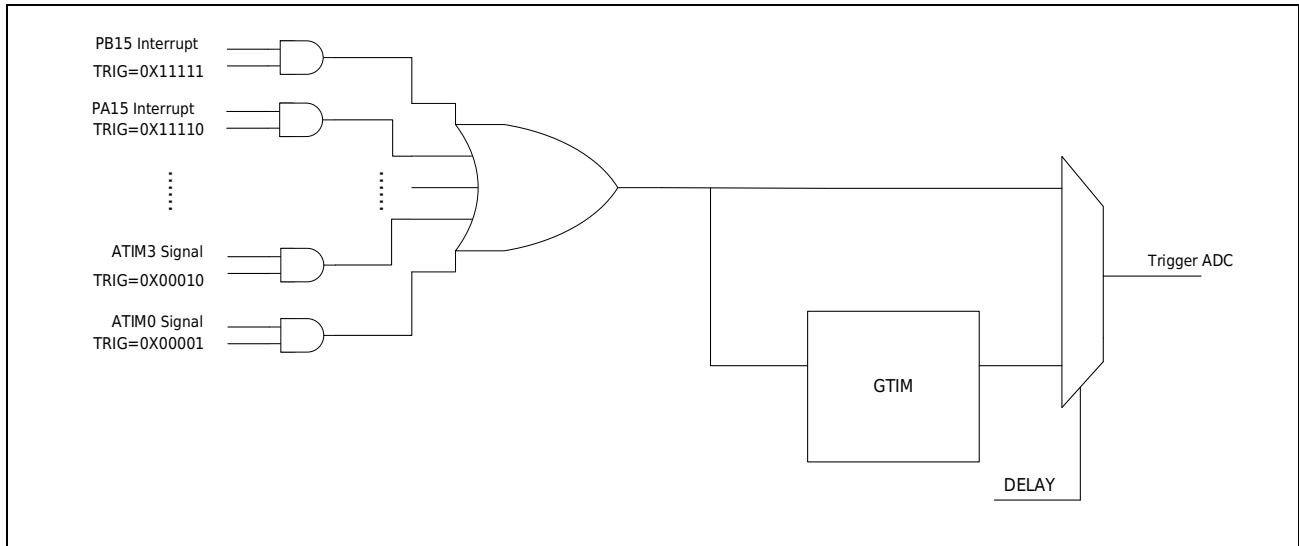


图 19-5 ADC 单次转换或顺序扫描转换外部触发源示意图

19.8 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 HTCMP、LTCMP、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.HTI 置 1；向 ADC_ICR.HTI 写入 0 则清零 ADC_IFR.HTI。

下阈值比较：当 ADC 转换结果位于[0, ADC_LT)区间内则 ADC_IFR.LTI 置 1；向 ADC_ICR.LTI 写入 0 则清零 ADC_IFR.LTI。

区间值比较：当 ADC 转换结果位于[ADC_LT, ADC_HT)区间内则 ADC_IFR.RGI 置 1；向 ADC_ICR.RGI 写入 0 则清零 ADC_IFR.RGI。

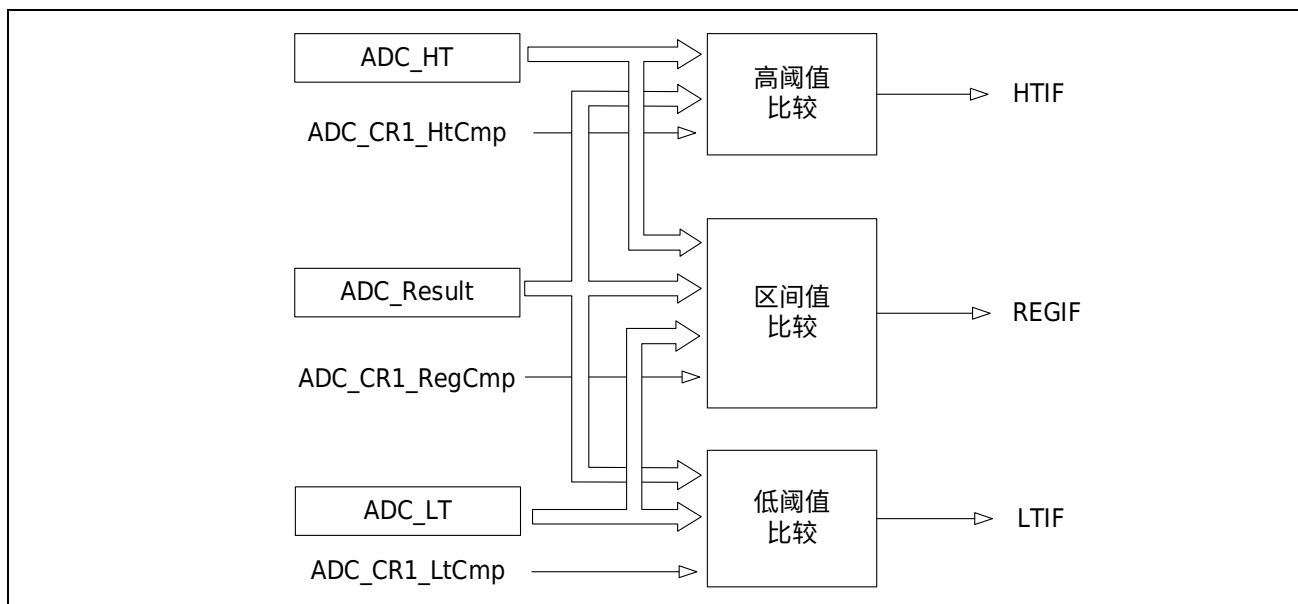


图 19-6 ADC 转换结果比较

19.9 ADC 中断

ADC 中断请求如下表所示：

表 19-2 ADC 中断

中断源	中断标志	中断使能
ADC 转换结果溢出	ADC_IFR.OV	ADC_IER.OV
ADC 扫描单通道转换完成	ADC_IFR.EOC	ADC_IER.EOC
ADC 顺序扫描转换完成	ADC_IFR.EOS	ADC_IER.EOS
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.RGI	ADC_IER. RGI
ADC转换结果位于上阈值区域 (大于等于上阈值)	ADC_IFR.HTI	ADC_IER.HTI
ADC转换结果比较下阈值区域 (小于下阈值)	ADC_IFR.LTI	ADC_IER.LTI
ADC 单次转换完成	ADC_IFR.ESG	ADC_IER.ESG

19.10 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的环境温度。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。

计算公式如下：

$$\text{环境温度} = \text{TS_ENV} / 2 + 0.09354 \times \text{Vref} \times (\text{AdcValue} - \text{Trim})$$

其中：

TS_ENV: 校准值对应的环境温度*2, 计算时需要从 Flash 存储器中读出, 其存放地址为 0x0100 1234 ~ 0x0100 1237。

Vref: 当前 ADC 模块的参考电压, 取值为 1.5 或 2.5。

AdcValue: ADC 模块测量温度传感器输出电压的结果, 取值为 0~4095。

Trim: 32Bits 的校准值, 计算时需要从 Flash 存储器中读出, 其存放地址详见下表。

ADC 参考电压	校准值存放地址
内部 1.5V	0x0100 122C ~ 0x0100 122F
内部 2.5V	0x0100 1230 ~ 0x0100 1233

计算示例如下:

条件 1: Vref=2.5、AdcValue=0x7E5、Trim=0x76C、TS_ENV=0x32:

温度 1: $0x32/2 + 0.09354 \times 2.5 \times (0x7E5 - 0x76C) = 53.30^{\circ}\text{C}$ 。

条件 2: Vref=1.5、AdcValue=0x72D、Trim=0x76C、TS_ENV=0x32:

温度 2: $0x32/2 + 0.09354 \times 1.5 \times (0x72D - 0x76C) = 16.16^{\circ}\text{C}$ 。

通过 ADC 测量环境温度操作流程:

- Step1. 设置 ADC_BGR 为 3, 使能 BGR 模块和温度传感器模块;
- Step2. 设置 ADC_CR0.EN 为 1, 使能 ADC 模块;
- Step3. 延时 20μs, 等待 ADC、BGR 模块和温度传感器模块启动完成;
- Step4. 设置 ADC_CR1.MODE 为 0, 选择单次转换模式;
- Step5. 配置 ADC_CR0.REF, 选择 ADC 的参考电压为内部 1.5V 或内部 2.5V;
- Step6. 配置 ADC_CR0.SAM 及 ADC_CR0.CKDIV, 设置 ADC 的转换速度;
- Step7. 设置 ADC_CR0.SGLMUX 为 0x11, 选择待转换的通道为温度传感器的输出;
- Step8. 设置 ADC_CR0.BUF 为 1, 使能输入信号放大器;
- Step9. 设置 ADC_ICR. ESG 为 0, 清除 ADC_IFR. ESG 标志;
- Step10. 设置 ADC_SGLSTART.START 为 1, 启动 ADC 单次转换;
- Step11. 等待 ADC_IFR. ESG 变为 1, 读取 ADC_RESULT 寄存器以获取 ADC 转换结果;
- Step12. 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;
- Step13. 读取温度传感器校准值, 根据公式计算当前的环境温度。

19.11 寄存器描述

19.11.1 寄存器列表

表 19-3 ADC 基地址列表

名称	基地址	描述
ADC0	0x4000 A000	ADC0 基地址
ADC1	0x4000 A400	ADC1 基地址

表 19-4 ADC 寄存器列表

寄存器名	偏移地址	复位值
ADC 模拟器件选择寄存器 (ADC_BGR)	0x000	0x0000 0000
ADC 基本配置寄存器 0 (ADC_CR0)	0x004	0x0000 27F0
ADC 基本配置寄存器 1 (ADC_CR1)	0x008	0x0000 8000
ADC 中断与 DMA 触发使能 (ADC_IER)	0x030	0x0000 0000
ADC 顺序扫描转换通道配置寄存器 0 (ADC_SQR0)	0x040	0x0000 0000
ADC 顺序扫描转换通道配置寄存器 1 (ADC_SQR1)	0x044	0x0000 0000
ADC 顺序扫描转换通道配置寄存器 2 (ADC_SQR2)	0x048	0x0000 0000
ADC 转换结果 (ADC_RESULT)	0x0A0	0x0000 0000
ADC 转换结果累加值 (ADC_RESULTACC)	0x0A4	0x0000 0000
ADC 比较上阈值 (ADC_HT)	0x0A8	0x0000 0FFF
ADC 比较下阈值 (ADC_LT)	0x0AC	0x0000 0000
ADC 中断标志寄存器 (ADC_IFR)	0x0B0	0x0000 0000
ADC 中断清除寄存器 (ADC_ICR)	0x0B4	0x0000 03FF
ADC 外部中断触发源配置寄存器 0 (ADC_EXTTRIG0)	0x0B8	0x0000 0000
ADC 单次转换启动控制寄存器 (ADC_SGLSTART)	0x0C0	0x0000 0000
ADC 顺序扫描转换启动控制寄存器 (ADC_SQRSTART)	0x0C4	0x0000 0000

19.11.2 ADC 模拟器件选择寄存器 (ADC_BGR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														TS_EN	BGR_EN
位	标记	位名	功能		读写										
b31~b2	Res	保留位	保留，必须保持复位值												
b1	TS_EN	TS 模块使能	1: TS 模块使能 0: TS 模块关闭		RW										
b0	BGR_EN	BGR 模块使能	1: BGR 模块使能 0: BGR 模块关闭		RW										

注：该寄存器只有 ADC0 中存在，ADC1 中不存在。

19.11.3 ADC 基本配置寄存器 0 (ADC_CR0)

复位值: 0x0000 27F0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		SAM		BUF	REF		SGLMUX					CKDIV		EN	

位	标记	位名	功能	读写
b31~b14	Res	保留位	保留, 必须保持复位值	
b13~b12	SAM	ADC 采样周期选择	00: 3 个采样周期 01: 6 个采样周期 10: 9 个采样周期 11: 10 个采样周期, 注: ADC 转换周期固定 15 个	RW
b11	BUF	内置跟随器使能控制	0: 关闭跟随器, 外部输入信号与 ADC 直接相连。 1: 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。 注 1: 使能该功能时, 最大速率为 200Ksps 注 2: 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、DAC 输出	RW
b10~b9	REF	ADC 参考电压选择 (仅 ADC0 中寄存器有效, ADC0/1 共用)	00: 内部 1.5V 01: 内部 2.5V 10: 外部参考电压 ExRef (PB01) 11: AVCC 电压	RW
b8~b4	SGLMUX	单次转换模式转换通道选择	0000: AIN0 (PA00) 00001: AIN1 (PA01) 00010: AIN2 (PA02) 00011: AIN3 (PA03) 00100: AIN4 (PA04) 00101: AIN5 (PA05) 00110: AIN6 (PA06) 00111: AIN7 (PA07) 01000: AIN8 (PB00) 01001: AIN9 (PB01) 01010: AIN10 (PC00) 01011: AIN11 (PC01) 01100: AIN12 (PC02) 01101: AIN13 (PC03) 01110: AIN14 (PC04) 01111: AIN15 (PC05) 10000: AVCC/3 注: ADC_CR0.Buf 必须为 1 10001: 内置温度传感器输出电压 注: ADC_CR0.Buf 必须为 1 10010: Reserved	RW

b3~b1	CKDIV	ADC 时钟选择	000: PCLK 时钟	RW
			001: PCLK 时钟 4 分频	
			010: PCLK 时钟 6 分频	
			011: PCLK 时钟 8 分频	
			100: PCLK 时钟 10 分频	
			101: PCLK 时钟 12 分频	
			110: PCLK 时钟 14 分频	
			111: PCLK 时钟 16 分频	
b0	EN	ADC 使能控制	1: 使能 ADC	RW
			0: 禁止 ADC	

19.11.4 ADC 基本配置寄存器 1 (ADC_CR1)

复位值: 0x0000 8000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RACCCLR	REGCMP	HTCMP	LTCMP	RACCEN	OVMD	Res		THCH				ALIGN		MODE	

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15	RACCCLR	ADC 转换结果累加寄存器清零	1: 无作用; 0: ADC 转换结果累加寄存器 (ADC_ResultAcc) 清零。	R1W0
b14	REGCMP	ADC 区间比较控制	1: 使能区间比较 0: 禁止区间比较	RW
b13	HTCMP	ADC 高阈值比较控制	1: 使能高阈值比较 0: 禁止高阈值比较	RW
b12	LTCMP	ADC 低阈值比较控制	1: 使能低阈值比较 0: 禁止低阈值比较	RW
b11	RACCEN	ADC 转换结果自动累加控制	1: 使能 ADC 转换结果自动累加功能 0: 禁止 ADC 转换结果自动累加功能	RW
b10	OVMD	ADC 过转换时, Adc_Result 存储方式	0: 保留上次数据, 丢弃新的数据 1: 覆盖上次数据, 保留新的数据	RW
b9~b8	Res	保留位	保留, 必须保持复位值	
b7~b3	THCH	阈值比较通道选择	0000: 选择通道 0 进行阈值比较 00001: 选择通道 1 进行阈值比较 00010: 选择通道 2 进行阈值比较 10010: 选择通道 18 进行阈值比较	RW
b2	ALIGN	转换结果对齐控制	1: 转换结果 16Bits 左对齐存储 0: 转换结果 16Bits 右对齐存储	RW
b1~b0	MODE	转换模式	00 单次 01 顺序扫描单次序列 10 顺序扫描连续序列 11 顺序扫描断续序列	RW

19.11.5 ADC 顺序扫描转换通道配置寄存器 0 (ADC_SQR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		CH5MUX					CH4MUX				CH3MUX[4:1]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH3MUX[0]		CH2MUX				CH1MUX				CH0MUX					

位	标记	位名	功能	读写
b31~b30	Res	保留位	保留, 必须保持复位值	
b29~b25	CH5MUX	顺序扫描转换通道 5 选择	设置参见 ADC_CR0.SGLMUX	RW
b24~b20	CH4MUX	顺序扫描转换通道 4 选择	设置参见 ADC_CR0.SGLMUX	RW
b19~b15	CH3MUX	顺序扫描转换通道 3 选择	设置参见 ADC_CR0.SGLMUX	RW
b14~b10	CH2MUX	顺序扫描转换通道 2 选择	设置参见 ADC_CR0.SGLMUX	RW
b9~b5	CH1MUX	顺序扫描转换通道 1 选择	设置参见 ADC_CR0.SGLMUX	RW
b4~b0	CH0MUX	顺序扫描转换通道 0 选择	设置参见 ADC_CR0.SGLMUX	RW

19.11.6 ADC 顺序扫描转换通道配置寄存器 1 (ADC_SQR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		CH11MUX					CH10MUX				CH9MUX[4:1]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH9MUX[0]		CH8MUX				CH7MUX				CH6MUX					

位	标记	位名	功能	读写
b31~b30	Res	保留位	保留, 必须保持复位值	
b29~b25	CH11MUX	顺序扫描转换通道 11 选择	设置参见 ADC_CR0.SGLMUX	RW
b24~b20	CH10MUX	顺序扫描转换通道 10 选择	设置参见 ADC_CR0.SGLMUX	RW
b19~b15	CH9MUX	顺序扫描转换通道 9 选择	设置参见 ADC_CR0.SGLMUX	RW
b14~b10	CH8MUX	顺序扫描转换通道 8 选择	设置参见 ADC_CR0.SGLMUX	RW
b9~b5	CH7MUX	顺序扫描转换通道 7 选择	设置参见 ADC_CR0.SGLMUX	RW
b4~b0	CH6MUX	顺序扫描转换通道 6 选择	设置参见 ADC_CR0.SGLMUX	RW

19.11.7 ADC 顺序扫描转换通道配置寄存器 2 (ADC_SQR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CNT				CH15MUX[4:1]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH15MUX[0]	CH14MUX					CH13MUX					CH12MUX				

位	标记	位名	功能	读写
b31~b24	Res	保留位	保留, 必须保持复位值	
b23~b20	CNT	顺序扫描转换次数		RW
b19~b15	CH15MUX	顺序扫描转换通道 15 选择	设置参见 ADC_CR0.SGLMUX	RW
b14~b10	CH14MUX	顺序扫描转换通道 14 选择	设置参见 ADC_CR0.SGLMUX	RW
b9~b5	CH13MUX	顺序扫描转换通道 13 选择	设置参见 ADC_CR0.SGLMUX	RW
b4~b0	CH12MUX	顺序扫描转换通道 12 选择	设置参见 ADC_CR0.SGLMUX	RW

19.11.8 ADC 转换结果 (ADC_RESULT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULT															

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	RO
b15~b0	RESULT	ADC 转换结果		RO

19.11.9 ADC 转换结果累加值 (ADC_RESULTACC)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												RESULTACC[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULTACC[15:0]															

位	标记	位名	功能	读写
b31~b20	Res	保留位	保留, 必须保持复位值	
b19~b0	RESULTACC	ADC 转换累加值		RO

19.11.10 ADC 比较上阈值 (ADC_HT)

复位值: 0x0000 0FFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HT											

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11~b0	HT	ADC 转换结果比较上阈值		RW

19.11.11 ADC 比较下阈值 (ADC_LT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				LT											

位	标记	位名	功能	读写
b31~b12	Res	保留位	保留, 必须保持复位值	
b11~b0	LT	ADC 转换结果比较下阈值		RW

19.11.12 ADC 中断使能 (ADC_IER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						OV	Res	DEOC	EOC	Res	EOS	RGI	HTI	LTI	ESG

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	OV	ADC 转换结果溢出中断使能	0: 禁止 1: 使能	RW
b8	Res	保留位	保留, 必须保持复位值	
b7	DEOC	顺序单通道转换完成触发 DMA	0: 禁止 1: 使能	RW
b6	EOC	顺序单通道转换完成中断使能	0: 禁止 1: 使能	RW
b5	Res	保留位	保留, 必须保持复位值	
b4	EOS	顺序序列转换完成中断使能	0: 禁止 1: 使能	RW
b3	RGI	区间比较中断使能	0: 禁止 1: 使能	RW
b2	HTI	高阈值比较中断使能	0: 禁止 1: 使能	RW
b1	LTI	低阈值比较中断使能	0: 禁止 1: 使能	RW
b0	ESG	单次转换中断使能	0: 禁止 1: 使能	RW

19.11.13 ADC 中断标志寄存器 (ADC_IFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						OV	Res		EOC	Res	EOS	RGI	HTI	LTI	ESG

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	OV	ADC 转换结果溢出标志		RO
b8~b7	Res	保留位	保留, 必须保持复位值	
b6	EOC	ADC 顺序扫描转换单次完成标志	1: ADC 顺序扫描转换完成 0: ADC 顺序扫描转换未完成	RO
b5	Res	保留位	保留, 必须保持复位值	
b4	EOS	ADC 顺序扫描转换完成标志	1: ADC 顺序扫描转换完成 0: ADC 顺序扫描转换未完成	RO
b3	RGI	ADC 转换结果比较区间标志	1: ADC 转换结果位于[ADC_LT , ADC_HT)区间内 0: ADC 转换结果位于[ADC_LT , ADC_HT)区间外	RO
b2	HTI	ADC 转换结果比较上阈值标志	1: ADC 转换结果位于[ADC_HT, 4095]区间内 0: ADC 转换结果位于[ADC_HT, 4095]区间外	RO
b1	LTI	ADC 转换结果比较下阈值标志	1: ADC 转换结果位于[0, ADC_LT)区间内 0: ADC 转换结果位于[0, ADC_LT)区间外	RO
b0	ESG	ADC 单次转换完成标志	1: ADC 单次转换完成 0: ADC 单次转换未完成	RO

19.11.14 ADC 中断清除寄存器 (ADC_ICR)

复位值: 0x0000 03FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						OV	Res		EOC	Res	EOS	RGI	HTI	LTI	ESG

位	标记	位名	功能	读写
b31~b10	Res	保留位	保留, 必须保持复位值	
b9	OV	ADC 转换结果溢出标志清除位	写 0 清除数据覆盖标志清除 写 1 无作用	R1W0
b8~b7	Res	保留位	保留, 必须保持复位值	
b6	EOC	ADC 顺序扫描转换单次完成标志清除位	写 0 清除 ADC 顺序扫描转换单次完成标志 写 1 无作用	R1W0
b5	Res	保留位	保留, 必须保持复位值	
b4	EOS	ADC 顺序扫描转换完成标志清除位	写 0 清除 ADC 顺序扫描转换完成标志 写 1 无作用	R1W0
b3	RGI	ADC 转换结果比较区间标志清除位	写 0 清除 ADC 转换结果比较区间标志 写 1 无作用	R1W0
b2	HTI	ADC 转换结果比较上阈值标志清除位	写 0 清除 ADC 转换结果比较上阈值 写 1 无作用	R1W0
b1	LTI	ADC 转换结果比较下阈值标志清除位	写 0 清除 ADC 转换结果比较下阈值标志 写 1 无作用	R1W0
b0	ESG	ADC 单次转换完成标志清除位	写 0 清除 ADC 单次转换完成标志 写 1 无作用	R1W0

19.11.15 ADC 外部中断触发源配置寄存器 0 (ADC_EXTTRIGGER0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										DELAY	TRIG				

位	标记	位名	功能	读写
b31~b6	Res	保留位	保留, 必须保持复位值	
b5	DELAY	ADC 单次转换或顺序扫描转换外部中断触发延时控制	1: Trig 选择的触发信号触发 GTIM, GTIM 信号触发 ADC (ADC0 触发 GTIM0. ADC1 触发 GTIM1) 0: 直接使用 TRIG 选择信号触发 ADC	RW
b4~b0	TRIG	ADC 单次转换或顺序扫描转换外部中断触发信号选择	11111 PB15 中断触发 ADC 转换	RW
			11110 PA15 中断触发 ADC 转换	
			11101 PB13 中断触发 ADC 转换	
			11100 PA13 中断触发 ADC 转换(note: PA13 默认是 swd, 需要复用为 GPIO)	
			11011 PB11 中断触发 ADC 转换	
			11010 PA11 中断触发 ADC 转换	
			11001 PB09 中断触发 ADC 转换	
			11000 PA09 中断触发 ADC 转换	
			10111 PB07 中断触发 ADC 转换	
			10110 PA07 中断触发 ADC 转换	
			10101 PB05 中断触发 ADC 转换	
			10100 PA05 中断触发 ADC 转换	
			10011 PB03 中断触发 ADC 转换	
			10010 PA03 中断触发 ADC 转换	
			10001 PB01 中断触发 ADC 转换	
			10000 PA01 中断触发 ADC 转换	
			01111 LVD 中断触发 ADC 转换	
			01110 VC1 中断触发 ADC 转换	
			01101 VC0 中断触发 ADC 转换	
			01100 DMA 中断触发 ADC 转换	
			01011 SPI1 中断触发 ADC 转换	
			01010 SPI0 中断触发 ADC 转换	
			01001 USART3 中断触发 ADC 转换	
			01000 USART2 中断触发 ADC 转换	
			00111 USART1 中断触发 ADC 转换	
			00110 USART0 中断触发 ADC 转换	
			00101 GTIM2 信号触发 ADC 转换	
			00100 GTIM1 信号触发 ADC 转换	
			00011 GTIM0 信号触发 ADC 转换	
			00010 ATIM3 信号触发 ADC 转换	
			00001 ATIM0 信号触发 ADC 转换	
			00000 无外设触发	

注:

1. 定时器触发 ADC 有专用的控制信号，在定时器内部控制。
2. 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。

19.11.16 ADC 单次转换启动控制寄存器 (ADC_SGLSTART)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														START	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	START	ADC 单次转换启动控制	CR0.EN=1 时, 才可正确回读 1: 启动 ADC 单次转换 0: 停止 ADC 单次转换	RW

19.11.17 ADC 顺序扫描转换启动控制寄存器 (ADC_SQRSTART)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														START	

位	标记	位名	功能	读写
b31~b1	Res	保留位	保留, 必须保持复位值	
b0	START	ADC 顺序扫描转换启动控制	CR0.EN=1 时, 才可正确回读 1: 启动 ADC 顺序扫描转换 0: 停止 ADC 顺序扫描转换	RW

20 模拟电压比较器 (VC)

20.1 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

20.2 主要特性

- 支持 64 阶分压基准
- 支持 11 个外部输入端口和 4 个片内输入信号
- 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

20.3 功能描述

20.3.1 功能框图

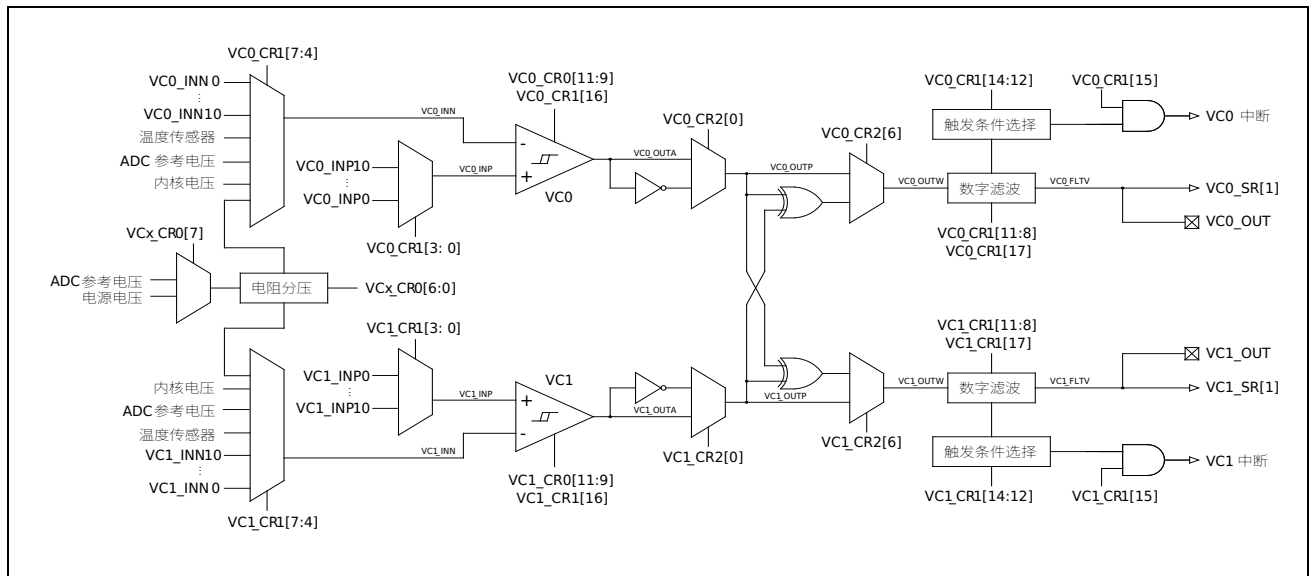


图 20-1 VC 功能框图

20.3.2 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的结果的时间由 VCx_CR0.BIAS 控制位决定，电流越大 VC 响应越快，典型值从 200ns 到 20us 四档可调。

如果选择 ADC 参考电压作为比较基准电压或 CR0.BIAS 为 2 或 3 时，则需要使能 BGR 模块。BGR 模块的启动时间为 Tstable(BGR)，电压比较器需要等待 BGR 模块稳定后才能正常输出。

20.3.3 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 VC 工作图示如下：

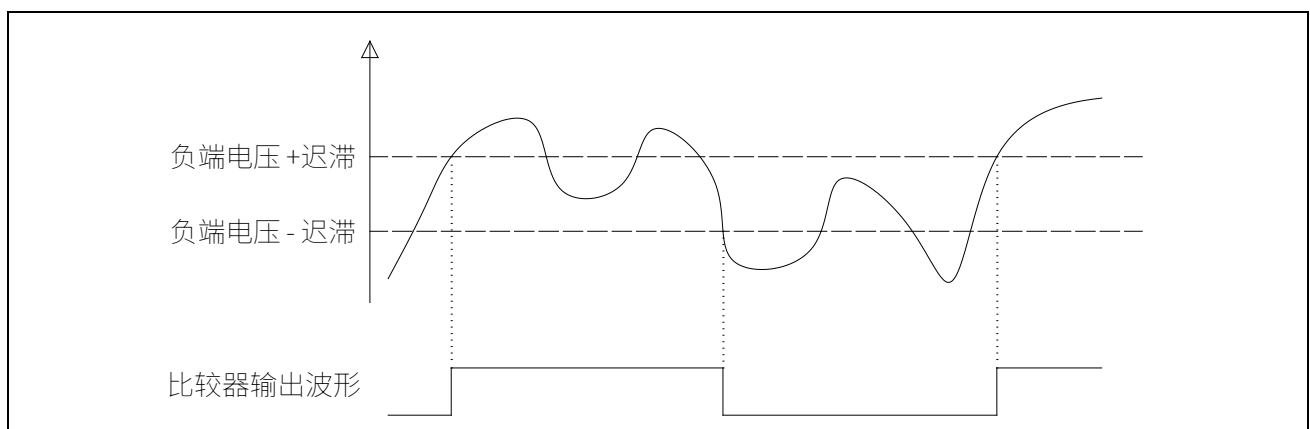


图 20-2 VC 迟滞功能

20.3.4 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。

20.3.5 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。

20.3.6 滤波描述

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。

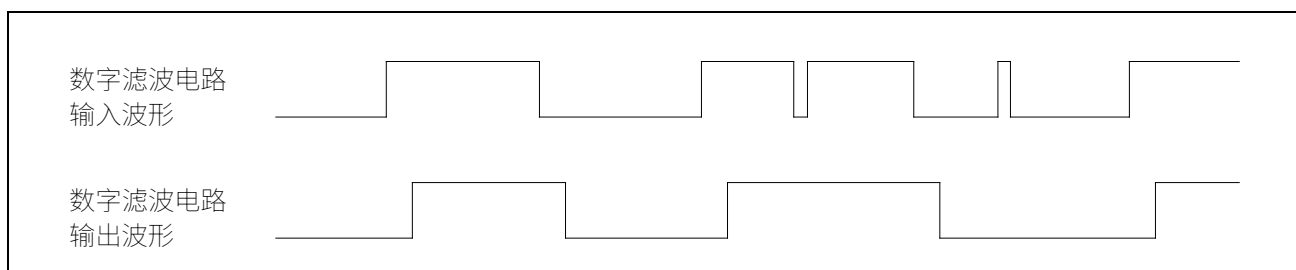


图 20-3 VC 滤波响应时间

20.4 寄存器描述

20.4.1 寄存器列表

表 20-1 VC 基地址列表

名称	基地址	描述
VC0	0x4000 A800	VC0 基地址
VC1	0x4000 A810	VC1 基地址

表 20-2 VC 寄存器列表

寄存器名	偏移地址	复位值
VC 控制寄存器 0 (VC_CR0)	0x00	0x0000 0020
VC 控制寄存器 1 (VC_CR1)	0x04	0x0000 0000
VC 控制寄存器 2 (VC_CR2)	0x08	0x0000 0000
VC 状态寄存器 (VC_SR)	0x0C	0x0000 0000

20.4.2 控制寄存器 0 (VC_CR0)

复位值: 0x0000 0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HYS		BIAS		REF	Div En	DIV					

位	标记	位名	功能	读写
b31~b12	Reserved	保留位	保留, 必须保持默认值	
b11~b10	HYS	迟滞功能配置	11: 迟滞电压约30mV 10: 迟滞电压约20mV 01: 迟滞电压约10mV 00: 无迟滞	RW
b9~b8	BIAS	功耗选择	11: 高功耗模式; 需要开启BGR, BGR启动时间见数据手册 10: 中功耗模式; 需要开启BGR, BGR启动时间见数据手册 01: 中低功耗模式 00: 低功耗模式 注: 功耗越大, 响应速度越快	RW
b7	REF	电阻分压电路参考电压选择	1: ADC的参考电压, 需使能ADC、BGR 0: AVCC 注意: VC0/VC1共用	RW
b6	DivEn	电阻分压电路使能控制	1: 使能 0: 禁止 注意: VC0/VC1共用	RW
b5~b0	DIV	电阻分压电路输出电压配置	111111: 64/64Vref 111110: 63/64 Vref ... 000001: 2/64 Vref 000000: 1/64 Vref 注意: VC0/VC1共用, Vref来源由REF控制位决定	RW

20.4.3 控制寄存器 1 (VC_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														FltClk	EN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IE	Level	Rising	Falling	FltTime			FLTEN	NSEL				PSEL			

位	标记	位名	功能	读写
b31~b18	Res	保留	保留, 必须保持复位值	
b17	FltClk	滤波时钟选择	0: 内置 RC 振荡器, 频率约 150KHz 1: PCLK 注意: 当滤波时钟为 PCLK 时, VC 不能唤醒 DeepSleep 模式的 MCU	RW
b16	EN	使能控制	0: 禁止 1: 使能	RW
b15	IE	中断使能控制	0: 禁止 1: 使能	RW
b14	Level	滤波信号高电平触发中断使能控制	0: 禁止 1: 使能	RW
b13	Rising	滤波信号上升沿触发中断使能控制	0: 禁止 1: 使能	RW
b12	Falling	滤波信号下降沿触发中断使能控制	0: 禁止 1: 使能	RW
b11~b9	FltTime	滤波时间配置	000: 滤除宽度小于 1 个滤波时钟的信号 001: 滤除宽度小于 3 个滤波时钟的信号 010: 滤除宽度小于 7 个滤波时钟的信号 011: 滤除宽度小于 15 个滤波时钟的信号 100: 滤除宽度小于 63 个滤波时钟的信号 101: 滤除宽度小于 255 个滤波时钟的信号 110: 滤除宽度小于 1023 个滤波时钟的信号 111: 滤除宽度小于 4095 个滤波时钟的信号	RW
b8	FLTEN	滤波功能使能配置	1: 使能, 滤波宽度由 FltTime 控制 0: 禁止, 滤波器输出波形与输入波形相同	RW
b7~b4	NSEL	电压比较器负端输入信号选择	0000: VCx_INN0 (PA00) 0001: VCx_INN1 (PA01) 0010: VCx_INN2 (PA02) 0011: VCx_INN3 (PA03) 0100: VCx_INN4 (PA04) 0101: VCx_INN5 (PA05) 0110: VCx_INN6 (PB00) 0111: VCx_INN7 (PB01) 1000: VCx_INN8 (PB12) 1001: VCx_INN9 (PB04) 1010: VCx_INN10 (PB05) 1011: 内置电阻分压输出电压 1100: 内置温度传感器输出电压 1101: Reserved	RW

			1110: ADC 模块所配置的参考电压 (需使能 ADC)	
			1111: 内核电压	
			0000: VCx_INP0 (PA00)	
			0001: VCx_INP1 (PA01)	
			0010: VCx_INP2 (PA02)	
			0011: VCx_INP3 (PA03)	
			0100: VCx_INP4 (PA04)	
b3~b0	PSEL	电压比较器正端输入信号选择	0101: VCx_INP5 (PA05)	RW
			0110: VCx_INP6 (PB00)	
			0111: VCx_INP7 (PB01)	
			1000: VCx_INP8 (PB12)	
			1001: VCx_INP9 (PB04)	
			1010: VCx_INP10 (PB05)	
			其它: 保留	

20.4.4 控制寄存器 2 (VC_CR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									Window	ATIMBK	ATIM3CLR	Res		ATIM0CLR	POL
位	标记	位名	功能										读写		
b31~b7	Res	保留位	保留, 必须保持复位值												
b6	Window	窗口比较功能配置	1: 使能 0: 禁止										RW		
b5	ATIMBK	VCx 滤波后信号输出到 ATIM0/3 刹车使能控制	1: 使能 0: 禁止										RW		
b4	ATIM3CLR	VCx 滤波后信号输出到 ATIM3 REFCLR 使能控制	1: 使能 0: 禁止										RW		
b3~b2	Res	保留位	保留, 必须保持复位值												
b1	ATIM0CLR	VCx 滤波后信号输出到 ATIM0 REFCLR 使能控制	1: 使能 0: 禁止										RW		
b0	POL	VCx_OUTP 信号极性配置	1: 正端大于负端时 VCx_OUTP 低 0: 正端大于负端时 VCx_OUTP 高										RW		

20.4.5 状态寄存器 (VC_SR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													FLT V	INT F	
位	标记	位名	功能										读写		
b31~b2	Res	保留	保留, 必须保持复位值												
b1	FLT V	滤波后的信号电平	1: 高电平 0: 低电平										RO		
b0	INT F	中断标志	1: 已发生中断; 写 0 清除该标志 0: 未发生中断										RW		

21 低电压检测器（LVD）

21.1 LVD 简介

LVD 可用于监测电源电压及芯片管脚的电压。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。

21.2 主要特性

- 4 路监测源，AVCC、PC13、PB08、PB07
- 16 阶阈值电压，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
- 8 阶滤波配置，防止误触发
- 具备迟滞功能，强力抗干扰
- 支持在 DeepSleep 模式下工作

21.3 功能描述

21.3.1 功能框图

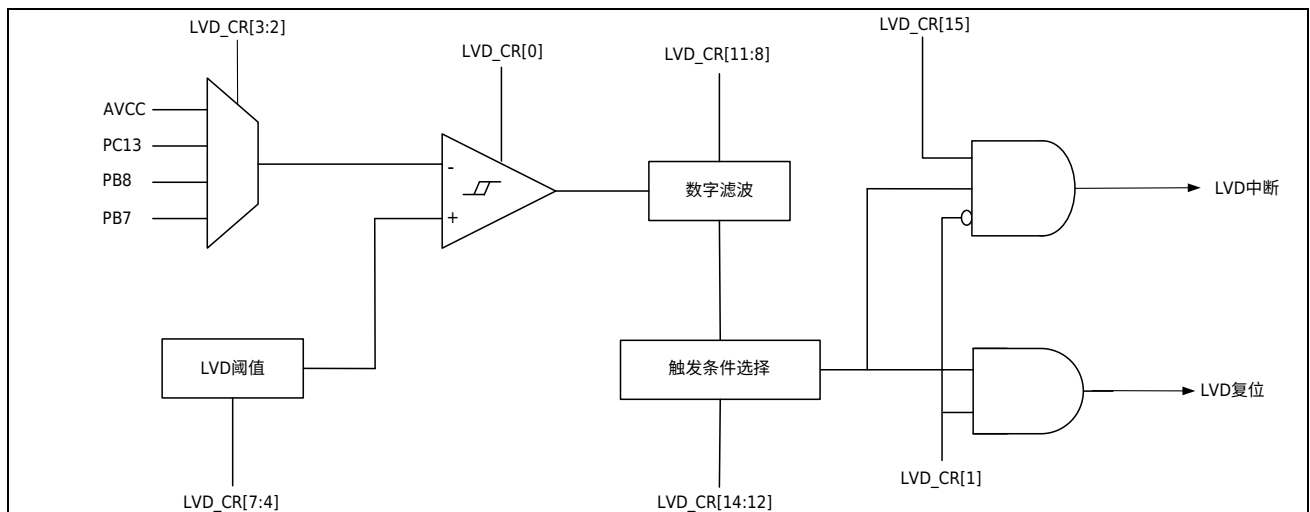


图 21-1 LVD 框图

21.3.2 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高于或低于阈值电压 20mV 后才发生翻转，其输入输出信号示意如下：

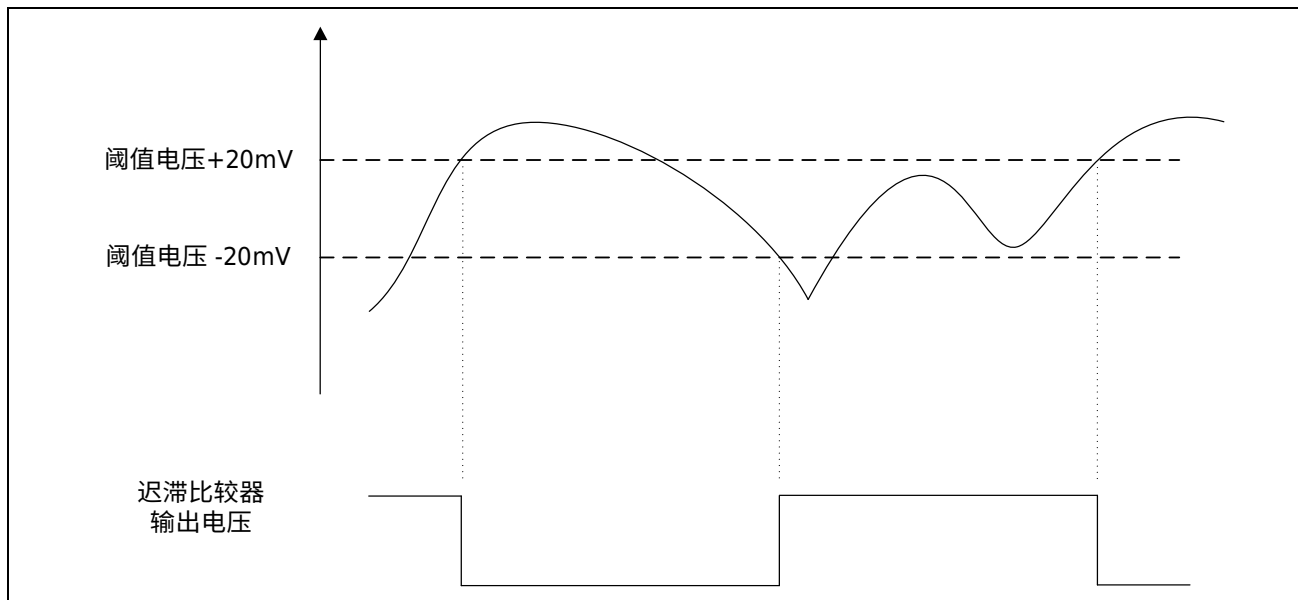


图 21-2 LVD 迟滞响应

21.3.3 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.FltTime 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 150KHz 的内置 RC 振荡器。

使能数字滤波模块时滤波器输入输出波形示意如下所示：

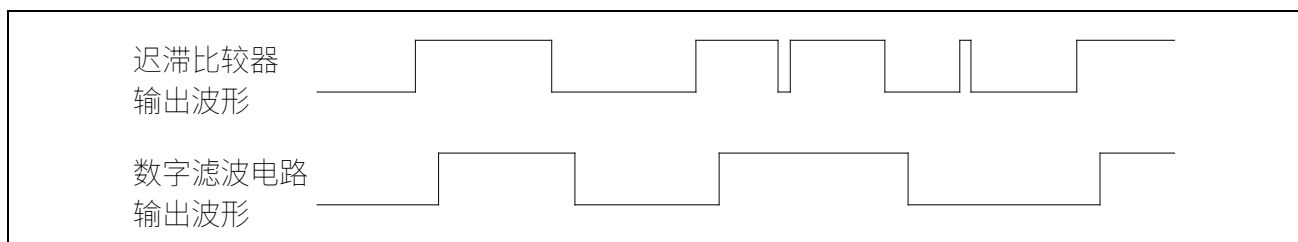


图 21-3 LVD 滤波输出

21.4 编程示例

21.4.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。

配置方法如下所示：

- Step1. 配置 LVD_CR.Source，选择待监测的电压来源；
- Step2. 配置 LVD_CR.VTDS，选择 LVD 的阈值电压；
- Step3. 配置 LVD_CR.FltTime，选择 LVD 滤波时间；
- Step4. 设置 LVD_CR.FltEn 为 1，使能 LVD 滤波；
- Step5. 设置 LVD_CR.HTEN 为 1，选择高电平触发 LVD 动作；
- Step6. 设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位；
- Step7. 设置 LVD_CR.EN 为 1，使能 LVD。

21.4.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。

配置方法如下所示：

- Step1. 配置 LVD_CR.Source，选择待监测的电压来源；
- Step2. 配置 LVD_CR.VTDS，选择 LVD 的阈值电压；
- Step3. 配置 LVD_CR.FltTime，选择 LVD 滤波时间；
- Step4. 设置 LVD_CR.FltEn 为 1，使能 LVD 滤波；
- Step5. 设置 LVD_CR.RTEN 和 LVD_CR.FTEN 为 1，选择电平变化触发 LVD 动作；
- Step6. 设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断；
- Step7. 设置 LVD_CR.IE 为 1，使能 LVD 中断；
- Step8. 使能 NVIC 中断向量表中的 LVD 中断；
- Step9. 设置 LVD_CR.EN 为 1，使能 LVD；
- Step10. 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_SR 写入 0x00 以清除中断标志。

21.5 寄存器描述

21.5.1 寄存器列表

基地址：0x4000 AA00

表 21-1 LVD 寄存器列表

寄存器名	偏移地址	复位值
LVD配置寄存器 (LVD_CR)	0x00	0x0000 0000
LVD中断标志寄存器 (LVD_SR)	0x04	0x0000 0000

21.5.2 LVD 配置寄存器 (LVD_CR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IE	HTEN	RTEN	FTEN	FltTime			FltEn	VTDS				Source		ACT	EN

位	标记	位名	功能	读写
b31~b16	Res	保留位	保留, 必须保持复位值	
b15	IE	LVD 中断使能	1: 使能; 0: 禁止。	RW
b14	HTEN	高电平触发使能 (被监测电压低于阈值电压)	1: 使能; 0: 禁止。	RW
b13	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)	1: 使能; 0: 禁止。	RW
b12	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)	1: 使能; 0: 禁止。	RW
b11~b9	FltTime	数字滤波时间配置 (滤波时钟为150K时钟)	滤波时间数据参考数据手册 注: 滤波时间仅在FLTEN为1时有效。	RW
b8	FltEn	数字滤波使能配置	1: 使能数字滤波 0: 禁止数字滤波 注: 使能数字滤波时, 系统将自动使能RCL时钟。	RW
b7~b4	VTDS	LVD 阈值电压选择	阈值电压数据参考数据手册	RW
b3~b2	Source	LVD 监测来源选择	00: AVCC 电压 01: PC13端口输入信号的电压 10: PB08端口输入信号的电压 11: PB07端口输入信号的电压	RW
b1	ACT	LVD 触发动作选择	1: 系统复位 0: NVIC中断	RW
b0	EN	LVD 使能控制	1: 使能LVD 0: 禁止LVD	RW

21.5.3 LVD 中断寄存器 (LVD_SR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														FLTV	INTF
位	标记	位名	功能										读写		
b31~b2	Res	保留位	保留, 必须保持复位值												
b1	FLTV	数字滤波器输出的电平值	1: 数字滤波器输出高电平 0: 数字滤波器输出低电平										RO		
b0	INTF	LVD 中断标志	1: 已产生LVD中断标志, 写0清零 0: 未产生LVD中断标志										RW0		

22 调试控制器 (DBGC)

本产品参考如下 ARM 技术文档：

- Cortex™-M4F r0p1 技术参考手册 (TRM)
- ARM 调试接口 V5
- ARM CoreSight 设计套件版本 r0p1 技术参考手册

22.1 简介

本 MCU 的内核是 ARM® Cortex®-M4，该内核包含用于高级调试功能的硬件。利用这些调试功能，可以在取指（指令断点）或访问数据（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。

调试接口：

- 串行调试接口 SWD

22.2 DBGC 系统框图

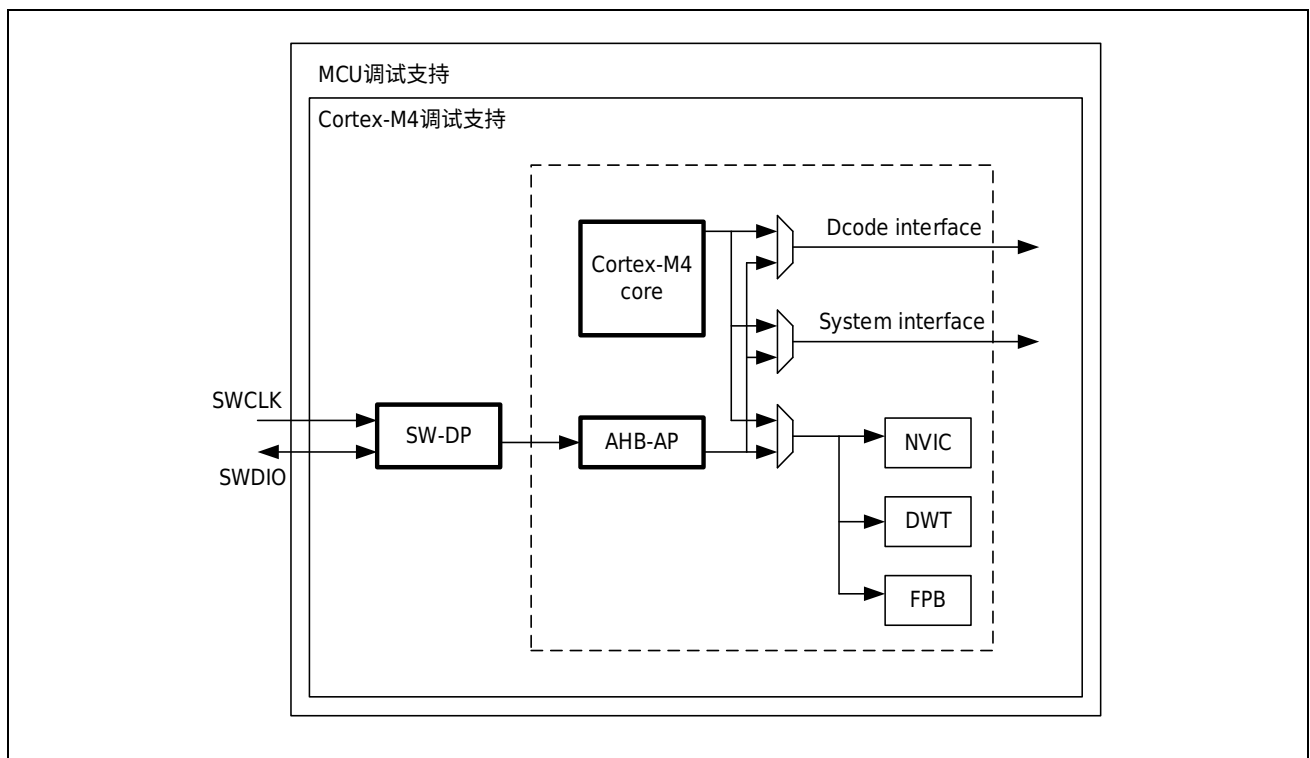


图 22-1 调试控制系统

ARM Cortex™-M4F 内核提供集成片上调试支持。它包括：

- SW-DP: SWD 调试端口
- AHP-AP: AHB 访问端口

- FPB: Flash 指令断点
- DWT: 数据观察点触发

注意: 有关 ARM Cortex™-M4F 内核支持的调试功能的更多信息, 请参见《Cortex™-M4F-r0p1 技术参考手册》和《CoreSight 设计套件 r0p1 技术参考手册》。

22.3 SW-DP 调试端口 (SWD)

该 MCU 内核集成了 SWD 调试端口。该端口是 ARM 标准 CoreSight 调试端口。

22.3.1 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:

表 22-1 SWD 端口引脚

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

22.3.2 SW-DP 调试端口引脚

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 SYSCtrl_CR1.SWDIO 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:

表 22-2 SW-DP 调试端口引脚

【加密芯片】选项	SYSCtrl_CR1.SWDIO 配置	PA13/PA14 功能
加密	0	NA
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

22.3.3 SWD 引脚上的内部上拉

用户软件禁止 SWD 引脚的调试功能后, 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 置于如下等效的状态:

- SWDIO: 输入上拉
- SWCLK: 输入上拉

由于内置上拉电阻, 因此无需添加外部电阻。

22.4 寄存器描述

22.4.1 寄存器列表

基地址：0x4000 B800

表 22-3 DBGC 寄存器列表

寄存器名	偏移地址	复位值
SYSCTRL_DEBUG (定时器调试暂停控制寄存器)	0x64	0x0000 0000

22.4.2 定时器调试暂停控制寄存器 (SYSCTRL_DEBUG)

具体寄存器内容请参考【定时器调试暂停控制寄存器 (SYSCTRL_DEBUG)】章节。

23 器件电子签名

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 HC32Fxxx / HC32Lxxx 微控制器。

23.1 产品唯一身份标识 (UID) 寄存器 (48 位)

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

48 位的唯一设备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。48 位的唯一设备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。

基址：0x0100 14E0

表 23-1 UID 定义

偏移地址	描述	UID Bits (48 bits)							
		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]							

23.2 产品型号寄存器

0x0100 1140 ~ 0x0100 114F 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：48433332463432304B41544200000000 所代表的产品型号为 HC32F420KATB。

23.3 FLASH 容量寄存器

基地址：0x0100 1160

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FLASHSIZE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FLASHSIZE [15:0]															

位	标记	位名	功能	读写
			注：以 Byte 为单位	
			0x00008000 代表 32KB	
			0x00010000 代表 64KB	
			0x00020000 代表 128KB	
				
b31~b0	FLASHSIZE	产品内置 Flash 的容量		RO

23.4 RAM 容量寄存器

基地址：0x0100 1164

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RAMSIZE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RAMSIZE [15:0]															

位	标记	位名	功能	读写
			注：以 Byte 为单位	
			0x00000800 代表 2KB	
			0x00001000 代表 4KB	
			0x00002000 代表 8KB	
				
b31~b0	RAMSIZE	产品内置 RAM 的容量		RO

23.5 管脚数量寄存器

基地址：0x0100 116A

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PINCOUNT [15:0]															

位	标记	位名	功能	读写
			注：以只为单位	
			0x0020 代表 32 只	
			0x0030 代表 48 只	
				
b15~b0	PINCOUNT	产品管脚的数量		RO

24 附录 A SysTick 定时器

24.1 SysTick 定时器简介

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-M4 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。

24.2 设置 SysTick

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:

- Step1. 配置 SysTick->CTRL. ENABLE 为 0, 禁止 SysTick。
- Step2. 配置 SysTick->CTRL. CLKSOURCE, 选择 SysTick 的时钟源。
- Step3. 配置 SysTick->LOAD, 选择 SysTick 的溢出周期。
- Step4. 向 SysTick->VAL 写入任意值, 清零 SysTick->VAL 及 SysTick->CTRL. COUNTFLAG。
- Step5. 配置 SysTick->CTRL. TICKINT 为 1, 使能 SysTick 中断。
- Step6. 配置 SysTick->CTRL. ENABLE 为 1, 使能 SysTick。
- Step7. 在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

注意: 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: 例如 4MHz	3999	1ms
XTL: 例如 32.768KHz	327	10.01ms

24.3 SysTick 寄存器

表 24-1 SysTick 寄存器

寄存器名	地址	CMSIS 符号
SysTick 控制和状态寄存器 (SYS_CSR)	0XE000 E010	SysTick->CTRL
SysTick 重载寄存器 (SYS_RVR)	0XE000 E014	SysTick->LOAD
SysTick 当前值寄存器 (SYS_CVR)	0XE000 E018	SysTick->VAL
SysTick 校准值寄存器 (SYS_CALIR)	0XE000 E01C	SysTick->CALIB

24.3.1 SysTick 控制和状态寄存器 (CTRL)

位	符号	功能描述	类型	复位值
31:17	Reserved	-	-	-
16	COUNTFLAG	SysTick 定时器溢出标志 1: SysTick 定时器发生下溢出 0: SysTick 定时器未发生溢出 读该寄存器, 可清除 COUNTFLAG 标志	RO	0
15:3	Reserved	-	-	-
2	CLKSOURCE	SysTick 时钟源选择 1: 使用内核时钟(HCLK) 0: 使用参考时钟(XTL) 注: 当选择参考时钟作为 SysTick 的时钟源时, 需要同步使能 XTL 时钟源及 SysCtrl.PERI_CLKEN0.TICK	RW	0
1	TICKINT	SysTick 中断使能 1: 使能中断 0: 禁止中断	RW	0
0	ENABLE	SysTick 定时器使能 1: 使能 SysTick 0: 禁止 SysTick	RW	0

24.3.1 SysTick 重载寄存器 (LOAD)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

24.3.2 SysTick 当前值寄存器 (VAL)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	CURRENT	读取该寄存器, 获取 SysTick 定时器的当前计数值 写任意值到该寄存器, 清零该寄存器及 COUNTFLAG	RW	未定义

24.3.3 SysTick 校准值寄存器（CALIB）

位	符号	功能描述	类型	复位值
31	NOREF	SysTick 当前计数时钟标志 1: 当前计数时钟为内核时钟 0: 当前计数时钟为外部参考时钟（XTL）	RO	-
30	SKEW	TENMS 精度指示 1: TENMS 值代表粗略的 10ms 0: TENMS 值代表精确的 10ms	RO	
29:24	Reserved	-	-	
23:0	TENMS	10 毫秒校准值	RO	-

版本修订记录

版本号	修订日期	修订内容
Rev1.00	2023/03/31	初版发布。
Rev1.10	2024/01/17	1) 复位控制器 (RESET): 修改 SYSCTRL_APBRST1、SYSCTRL_AHBRST0 和 SYSCTRL_RSTFLAG 的复位值; 删除 SYSCTRL_RSTFLAG 的 b2~b7 中“需要软件初始化及清除”, b7 添加注意事项。 2) 中断控制器 (NVIC): 修改“图 5-6 中断处理中产生的中断挂起也可以被确认”上文字为“中断服务程序”。 3) 通用输入输出端口控制器 (GPIO): 修改 6.3.5 端口通用输入输出功能中 GPIO 的控制寄存器数目描述为 6。 4) 时钟校准模块 (CTRIM): 添加 CTRIM_CR1 的 PRS 位为 0b1000 的分频值。 5) FLASH 控制器 (FLASH): FLASH 操作的擦除和写操作后添加将 OP 写 0 步骤; 添加 FLASH_CR 的 OP 位注意事项。 6) DMA 控制器 (DMAC): 添加 DMAC_CONFAx 的 TRI_SEL 位为 ATIM0 和 ATIM3 的外部触发和事件更新的描述。 7) 复合定时器 (CTIM): 修改 13.1 概述中 CTIM 的数目描述为“每个 CTIM 可以独立灵活配置为 GTIM 或 BTIM”; 添加 GTIM 的捕获功能的预分频值限制, 以及使用外部 ETR 计数时, 不能使用捕获功能; GTIM_CR0 寄存器的 PRS 位添加 Note: 捕获模式预分频系数只能设置 0, 1, 2, 3; 修改 GTIM_CR1 的 ETRFLT 位为 0b110 和 0b111 的分频值。 8) 高级定时器 (ATIM): 修改表 14-5 捕获输入 CH0A 信号列表内容; 修改图 14-51 名为“ATIM3 CH0A 端口选择”; 修改 ATIMx_M23CR 的 DIR 位描述为写禁止; ATIMx_MSCR 的 IA0S 位添加 ATIM0 的信号描述, 修改 TIM3 为 ATIM3, MMS 位添加对于 ATIM3 的描述。 9) 独立看门狗 (IWDG): 15.4 编程示例中, 在向 IWDG_KR 寄存器写入 0xCCCC 后添加 500us 延时; IWDG_KR 寄存器下添加延时 500us 注意事项。 10) 通用同步异步收发器 (USART): 17.3.9.2 时钟的采样和接收容差描述中, 智能卡 (7816) 模式中的接收管脚修改为 TXD; 修改“图 17-16 智能卡模式同步时序和采样时序图”中的接收管脚名为“接收管脚 (TXD)”; 修改“图 17-19 智能卡模式发送数据图例”中保护时间为 2_{etu} 以上; 删除 17.3.11.2 LIN 数据格式的“LIN 总线上的数据行为如下图所示”描述。

版本号	修订日期	修订内容
		11) 模数转换器 (ADC): 修改 19.6 转换累加模式中文字描述为“转换通道 8, 5, 2 配置为 AIN5, 转换通道 7, 4, 1 配置为 AIN1”。 12) 低电压检测器 (LVD): 添加 LVD 支持在 DeepSleep 模式下工作的描述。 13) 新增器件电子签名、SysTick 定时器章节。
Rev1.20	2024/05/06	1) 系统控制器 (SYSCTRL): 3.1.4 外部晶振时钟 XTH 中删除“并使能 XTH”描述; 3.1.5 锁相环时钟 PLL: 更新参数描述内容; 3.2.5 PLL (参考时钟为 RCHCLKD) 与 RCHCLKD 相互切换示例: 修改 Step2 中的 4MHz~24MHz 为 4MHz~12MHz, 并添加注意事项; 3.5.4 时钟使能控制寄存器 2 (SYSCTRL_CR2): 修改 b5 功能描述中的注意事项; 3.5.9 PLL 控制寄存器 (SYSCTRL_PLLCR): b1~b0 功能描述中添加注意事项。 2) 模数转换器 (ADC): 19.1 模块介绍: 19 路输入通道修改为 18 路, 删除“一路内建 BGR 1.2V 电压”; 19.2 ADC 框图: 删除图中 1.2V Ref 描述; 19.10 使用温度传感器测量环境温度: 修改公式和对应地址值, 修改计算示例中数值, 修改操作流程中寄存器名; 19.11.3 ADC 基本配置寄存器 0 (ADC_CR0): 删除 b11 功能描述中“一路内建 BGR 1.2V 电压”; 删除 b8~b4 功能描述中“10010 内部基准 1.2V 输出电压”。 3) 模拟电压比较器 (VC): 20.2 主要特性: 5 个片内输入信号修改为 4 个; 20.3.1 功能框图: 删除图中 1.2V 基准电压描述; 20.4.3 控制寄存器 1 (VC_CR1): 删除 b7~b4 功能描述中“1101 内置基准电压 1.2V”。
Rev1.21	2025/01/14	1) 系统控制器 (SYSCTRL): 3.5.4 时钟使能控制寄存器 2 (SYSCTRL_CR2): 修改 EXXTH 位的功能描述中的注意事项。
Rev1.22	2025/05/07	1) 串行外设接口 (SPI): 8.2 主要特性: 删除关于主从机最高通信速率的描述。
Rev1.23	2026/02/11	1) 复位控制器 (RESET): 4.2.6 复位标志寄存器 (SYSCTRL_RSTFLAG): 新增说明“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”。 2) 高级定时器 (ATIM): 14.3.6 模式 2/3 比较捕获模式: 单脉冲输出章节脉冲计数时序图细节优化, 新增相关说明; 捕获输入章节新增注意“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 时, GATE 输出有效)”。 3) 独立看门狗 (IWDG): 15.5.3 控制寄存器 (IWDG_CR): 注意第 2 点说明, 描述变更为“IWDG_SR.PRSF 为 0, 才可修改本寄存器”; 15.5.5 状

版本号	修订日期	修订内容
		态寄存器 (IWDT_SR): Bit0 位名由 “IWDT_CR.PRS 寄存器更新标志” 修改为 “CR 寄存器更新标志”, 同步修改功能描述对应取值描述, 说明修改为 “只有 PRSF 为 0 时才可以更新 CR 寄存器”。 4) 模数转换器 (ADC): 19.6 连续转换累加模式: ADC_IFR.SQRIF 改为 ADC_IFR.EOS, ADC_ICR.SQRIC 改为 ADC_ICR.EOS, 图 19-4 中 SQRIF 修改为 ADC_IFR.EOS; 19.9 ADC 中断: 新增 ADC 转换结果溢出中断 OV 相关描述; ADC_IER.OV、ADC_IFR.OV、ADC_ICR.OV 位名描述形式统一为 “ADC 转换结果溢出 XXX”。
Rev1.30	2026/08/19	1) DMA 控制器 (DMAC): “12.4.4 DMA 硬件 Block 传输模式” 章节, 修改硬件 Block 传输示意图。 2) 高级定时器 (ATIM): “14.3.6 模式 2/3 比较捕获模式-捕获输入” 章节, 修改 CH0B 捕获输入信号选择使用约束。