



内置 12 位 ADC / 16 位 PWM / 触摸按键 / 空气传感器 / 彩屏 32 位专用 SOC

彩屏电子烟专用 CA1020 系列 中文用户手册 REV1.0

深圳市锦锐科技股份有限公司

电话：0755-83949938

传真：0755-83949977

<http://www.cachip.com.cn>

地址：中国广东省深圳市南山区沙河西路深圳湾科技生态园一区 2 栋 B 座 5 层

重要声明：本公司保留对以下所有产品在可靠性、功能和设计方面作进一步说明的权利，同时保留在未通知的情况下，对本产品所有文档做更改的权利。客户在使用此产品时，请向我公司销售人员索取最新文档。特此声明！

目 录

1 概述.....	9
2 基本特性.....	9
3 芯片型号功能介绍.....	13
4 引脚及其描述.....	14
4.1 封装定义.....	14
4.2 引脚描述.....	16
5 存储器和总线架构.....	20
5.1 系统架构.....	20
5.1.1 总线架构.....	20
5.1.2 存储器组织.....	21
5.1.3 存储器映像.....	21
6 通用输入输出 (GPIO) 及复用定义.....	23
6.1 概述.....	23
6.2 功能描述.....	24
6.2.1 输入模式.....	24
6.2.2 输出模式.....	25
6.2.3 复用功能模式.....	26
6.2.4 模拟功能模式.....	26
6.3 复位后状态.....	27
6.4 外部中断/唤醒线.....	27
6.5 复用功能.....	27
6.6 软件 I/O 复用表.....	27
6.7 引脚寄存器描述.....	29
6.7.1 GPIO 配置寄存器.....	31
6.7.2 GPIOA 输出数据寄存器 (GPIOAO).....	32
6.7.3 GPIOB 输出数据寄存器 (GPIOBO).....	32
6.7.4 GPIOA 输入数据寄存器 (GPIOAI).....	32
6.7.5 GPIOB 输入数据寄存器 (GPIOBI).....	33
7 外部 FLASH 控制器.....	34
7.1 FLS 简介.....	34
7.2 FLS 寄存器概览.....	34
7.2.1 FLSCMD.....	34
7.2.1 FLSCR.....	35
8 中断和事件.....	36
8.1 嵌套向量中断寄存器.....	36
8.1.1 SysTick 校准值寄存器.....	36
8.1.2 中断和异常向量.....	36
8.2 外部中断/事件控制器 (EXTI).....	38
8.2.1 简介.....	38
8.2.2 主要特性.....	38
8.2.3 功能描述.....	39
8.2.4 EXTI 线路映射.....	41
8.3 中断控制寄存器.....	41
8.3.1 中断屏蔽寄存器 (EXTI_IMR).....	42
8.3.2 事件屏蔽寄存器 (EXTI_EMR).....	42
8.3.4 下降沿触发选择寄存器 (EXTI_FTISR).....	43
8.3.5 软件中断事件寄存器 (EXTI_SWIER).....	44
8.3.6 挂起寄存器 (EXTI_PR).....	44
8.3.7 外部中断选择寄存器 (EXTI_CR).....	45

9 时钟和电源系统	46
9.1 时钟系统介绍.....	46
9.2 时钟控制寄存器.....	47
9.2.1 时钟控制寄存器 (RCC_CR).....	47
9.2.2 APB1 外设复位寄存器 (RCC_APB1RSTR).....	50
9.2.3 APB1 外设时钟使能寄存器 (RCC_APB1ENR).....	52
9.2.4 控制/状态寄存器 (RCC_CSR).....	54
10 供电和复位系统	56
10.1 供电系统.....	56
10.2 复位系统.....	57
10.2.1 掉电复位 (BOR).....	57
10.2.2 低电压复位.....	57
10.2.3 看门狗复位.....	57
10.2.4 软复位.....	57
11 DMA 控制器	58
11.1 简介.....	58
11.2 主要特性.....	58
11.3 功能框图.....	59
11.4 功能描述.....	59
11.4.1 DMA 操作.....	59
11.4.2 通道优先级和仲裁器.....	60
11.4.3 DMA 通道和传输数量.....	60
11.4.4 可编程的数据位宽.....	60
11.4.5 外设/内存地址递增.....	62
11.4.6 通道配置流程.....	62
11.4.7 循环模式.....	62
11.4.8 错误管理.....	63
11.4.9 中断.....	63
11.4.10 DMA 请求映射.....	63
11.5 DMA 寄存器.....	65
11.5.1 DMA 寄存器概览.....	65
11.5.2 DMA 中断状态寄存器 (DMA_ISR).....	65
11.5.3 DMA 中断标志清除寄存器 (DMA_IFCR).....	66
11.5.4 DMA 通道 x 配置寄存器 (DMA_CCRx)(x = 1, 2).....	66
11.5.5 DMA 通道 x 传输数量寄存器 (DMA_CNDTRx)(x = 1, 2).....	68
11.5.6 DMA 通道 x 外设地址寄存器 (DMA_CPARx)(x = 1, 2).....	69
12 高级定时器 1 (TIM1)	70
12.1 TIM1 简介.....	70
12.2 TIM1 主要特性.....	70
12.3 TIM1 功能描述.....	71
12.3.1 时基单元.....	71
12.3.2 计数器模式.....	73
12.3.3 重复计数器.....	76
12.3.4 时钟源.....	78
12.3.5 捕获/比较通道.....	78
12.3.6 输入捕获模式.....	80
12.3.7 输出比较模式.....	80
12.3.8 PWM 模式.....	81
12.3.13 互补输出和死区插入.....	84
12.4 定时器 1 寄存器概览.....	86
12.4.1 TIM1 控制寄存器 1(TIM1_CR1).....	86
12.4.2 TIM1 控制寄存器 2(TIM1_CR2).....	89

12.4.3	TIM1 DMA/中断使能寄存器(TIM1_DIER).....	90
12.4.4	TIM1 状态寄存器(TIM1_SR).....	91
12.4.5	TIM1 事件产生寄存器(TIM1_EGR).....	92
12.4.6	TIM1 捕获/比较模式寄存器 1(TIM1_CCMR1).....	93
12.4.7	TIM1 捕获/比较模式寄存器 2(TIM1_CCMR2).....	96
12.4.8	TIM1 捕获/比较使能寄存器(TIM1_CCER).....	98
12.4.9	TIM1 计数器(TIM1_CNT).....	101
12.4.10	TIM1 预分频器(TIMx_PSC).....	101
12.4.11	TIM1 自动重装载寄存器(TIMx_ARR).....	101
12.4.12	TIM1 重复计数寄存器(TIMx_RCR).....	101
12.4.13	TIM1 捕获/比较寄存器 1(TIMx_CCR1).....	102
12.4.14	TIM1 捕获/比较寄存器 2(TIM1_CCR2).....	102
12.4.15	TIM1 捕获/比较寄存器 3(TIM1_CCR3).....	103
12.4.16	TIM1 捕获/比较寄存器 4(TIM1_CCR4).....	103
12.4.17	TIM1 死区寄存器(TIM1_BDTR).....	103
12.4.18	TIM1 DMA 控制寄存器(TIM1_DCR).....	105
13	通用定时器 14 (TIM14)	107
13.1	TIM14 简介	107
13.2	TIM14 主要特性	107
13.3	TIM14 功能描述	107
13.3.1	时基单元	107
13.3.2	计数器模式	108
13.3.3	重复计数器	109
13.3.4	时钟源	110
13.3.5	比较通道	111
13.3.6	输出比较模式	112
13.3.7	PWM 模式	113
13.4	定时器 14 寄存器概览	114
13.4.1	TIM14 控制寄存器 1(TIM14_CR1).....	114
13.4.2	TIM14 控制寄存器 2(TIM14_CR2).....	116
13.4.3	TIM14 DMA/中断使能寄存器(TIM14_DIER).....	117
13.4.4	TIM14 状态寄存器(TIM14_SR).....	118
13.4.5	TIM14 事件产生寄存器(TIM14_EGR).....	119
13.4.6	TIM14 比较模式寄存器 1(TIM14_CCMR1).....	120
13.4.7	TIM14 比较使能寄存器(TIM14_CCER).....	122
13.4.8	TIM14 计数器(TIM14_CNT).....	123
12.4.9	TIM14 预分频器(TIM14_PSC).....	123
13.4.10	TIM14 自动重装载寄存器(TIM4_ARR).....	123
12.4.11	TIM14 重复计数寄存器(TIM14_RCR).....	123
13.4.12	TIM14 比较寄存器 1(TIM14_CCR1).....	124
13.4.13	TIM14 比较寄存器 2(TIM14_CCR2).....	124
13.4.14	TIM14 DMA 控制寄存器(TIM14_DCR).....	125
13.4.15	TIM14 连续模式的 DMA 地址(TIM14_DMAR).....	126
14	基本定时器 (TIM6 和 TIM7)	127
14.1	基本定时器与通用定时器简介	127
14.2	基本定时器主要特性	127
14.3	基础定时器描述	128
14.3.1	时基单元	128
14.3.2	计数模式	128
14.3.3	时钟选择	129
14.4	TIMx 寄存器描述	130
14.4.1	寄存器概览	130

14.4.2	控制寄存器 1(TIMx_CR1)	131
14.4.3	DMA 中断使能寄存器(TIMx_DIER)	132
14.4.4	状态寄存器(TIMx_SR)	132
14.4.5	事件产生寄存器(TIMx_EGR)	132
14.4.6	计数器(TIMx_CNT)	133
14.4.7	预分频器(TIMx_PSC)	133
14.4.8	自动重装载寄存器(TIMx_ARR)	133
15	独立看门狗定时器 (IWDG)	134
15.1	简介	134
15.2	IWDG 主要特性	134
15.3	功能描述	135
15.3.1	寄存器访问保护	135
15.4	用户界面	136
15.4.1	操作流程	136
15.5	IWDG 寄存器概览	137
15.5.1	IWDG 密钥寄存器 (IWDG_KR)	137
15.5.2	IWDG 预分频寄存器 (IWDG_PR)	137
15.5.3	IWDG 重装载寄存器 (IWDG_RLR)	138
15.5.4	IWDG 状态寄存器 (IWDG_SR)	139
16	窗口看门狗 (WWDG)	140
16.1	简介	140
16.2	主要特征	140
16.3	功能描述	140
16.4	刷新看门狗和中断产生的时序	141
16.5	用户配置	142
16.5.1	WWDG 配置流程	142
16.6	WWDG 寄存器	143
16.6.1	WWDG 寄存器总览	143
16.6.2	控制寄存器 (WWDG_CR)	143
16.6.3	配置寄存器 (WWDG_CFR)	144
16.6.4	状态寄存器 (WWDG_SR)	144
17	通用串行接口 (UART0/UART1/UART2)	145
17.1	UART 寄存器概览	145
17.2	状态寄存器 (UART_CR)	145
17.3	波特率配置寄存器 (UART_S1REL)	146
17.4	发送接收缓冲器 (UART_BUF)	146
18	I²C 接口	147
18.1	简介	147
18.2	I ² C 寄存器描述	147
18.2.1	I ² C 寄存器概览	147
18.2.2	控制寄存器 1(I ² C_CCR)	147
18.2.3	控制寄存器 2(I ² C_DAT)	148
18.2.4	I ² C 地址寄存器 (I ² C_ADR)	148
18.2.5	I ² C 状态寄存器 (I ² C_FLG)	149
19	LED 级联控制	150
19.1	LED 级联简介	150
19.1.1	PWMLED 输出原理图	150
19.1.2	PWM 输出波形	150
19.1.3	单线级联 LED 驱动	150
19.2	PWMLED 寄存器概览	151
19.2.1	PWMCON	152
19.2.2	LEDAT	152

19.2.3 PWMDIV	153
19.2.4 PWMDUT	153
19.2.5 LEDWTM	153
19.2.6 LEDUT	153
20 SPI 接口	154
20.1 SPI 简介	154
20.2 SPI 主要特性	154
20.2.1 SPI 特性	154
20.3 SPI 功能描述	155
20.3.1 概述	155
20.3.2 从选择 (SSB) 脚管理	156
20.3.2 SPI 从模式	158
20.3.3 SPI 主模式	159
20.3.4 单工通信	160
20.3.5 状态标志	160
20.3.6 利用 DMA 的 SPI 通信	161
20.3.7 错误标志	162
20.3.8 关闭 SPI	162
20.3.9 SPI 中断	164
20.4 SPI 寄存器描述	165
20.4.1 SPI 寄存器概览	165
20.4.2 SPI 控制寄存器 1(SPI_CR1)	165
20.4.3 SPI 控制寄存器 2(SPI_CR2)	167
20.5.4 SPI 状态寄存器 (SPI_SR)	168
20.5.5 SPI 数据寄存器 (SPI_DR)	169
21 实时时钟 (RTC)	170
21.1 简介	170
21.2 主要特性	170
21.2.1 功能概述	171
21.2.2 模块复位	171
21.2.4 寄存器配置	172
21.2.5 RTC 标志的设置	172
21.3 RTC 寄存器	173
21.3.1 RTC 寄存器概览	173
21.3.2 RTC 控制寄存器高位 (RTC_CRH)	174
21.3.3 RTC 控制寄存器低位 (RTC_CRL)	174
21.4.4 RTC 预分频装载寄存器 (RTC_PRLH/RTC_PRL)	176
21.3.5 RTC 预分频器余数寄存器 (RTC_DIVH / RTC_DIVL)	177
21.3.6 RTC 计数器寄存器 (RTC_CNTH / RTC_CNTL)	177
21.3.7 RTC 闹钟寄存器 (RTC_ALRH/RTC_ALRL)	178
22 模/数字转换器 (ADC)	179
22.1 功能简介	179
22.2 主要特性	179
22.3 结构框图	179
22.4 功能描述	180
22.5 ADC 寄存器概览	180
22.5.1 ADC 控制信号寄存器 (ADCFG)	181
22.5.2 ADC 校准寄存器 (ADCCAL)	182
22.5.3 ADC 比较模式阈值下限值设定寄存器 (ADCDPL)	183
22.5.4 ADC 比较模式阈值上限值设定寄存器 (ADCDPH)	183
22.5.5 ADC 转换值寄存器 (ADCDA)	183
22.5.6 ADC 控制寄存器 (ADCTRL)	184

22.5.7 ADCSEQ.....	185
22.5.8 ADCCHSEL.....	185
22.5.9 ADCDALAT0.....	186
22.5.10 ADCDALAT1.....	186
22.5.11 ADCDALAT2.....	186
22.5.12 ADCDALAT3.....	186
22.5.13 ADCDALAT4.....	187
23 模拟比较器 (COMP)	188
23.1 模拟比较器简介.....	188
23.2 比较器寄存器.....	188
23.2.1 COMP0 配置寄存器 (CMP0)	188
23.2.2 COMP1 配置寄存器 (CMP1)	189
24 LD0 线性低电压输出.....	190
24.1 LD0 简介.....	190
24.2 LD0 寄存器.....	190
24.2.1 LD0 配置寄存器 (LD0)	190
25 LVD 可编程电压监测器.....	191
25.1 LVD 简介.....	191
25.2 LVD 寄存器.....	191
25.2.1 LVD 控制寄存器 CR.....	191
26 触摸按键 (Touch Key)	192
26.1 功能简介.....	192
26.2 主要特性.....	192
26.4 功能描述.....	193
26.4.1 触摸时钟预分频.....	193
26.4.2 低功耗模式.....	193
26.5 寄存器描述.....	193
26.5.1 寄存器概览.....	193
26.5.2 TK 控制寄存器 (TKCON)	194
26.5.3 TK 定时器启动时间寄存器 (TKMTS)	195
26.5.4 TK 通道配置寄存器 (TKCH01)	196
26.5.5 TK 通道配置寄存器 (TKCH23)	197
26.5.6 TK 通道配置寄存器 (TKCH45)	198
26.5.7 TK 中断标识寄存器 (TKIF)	199
26.5.8 TK 触摸采集数据寄存器 x(TKMSx)(x=0,1,2,3,4,5).....	199
27 空气传感器.....	200
27.1 空气传感器介绍.....	200
27.2 寄存器描述.....	200
27.2.1 传感器检测周期寄存器 (MICCYC)	200
27.2.2 传感器计数结果寄存器 (MICDAT)	200
27.2.3 传感器时钟沿阈值 (MICTHR)	201
27.2.4 传感器时钟沿最大阈值 (MICSEL)	201
28 程序下载和仿真介绍.....	202
28.1 程序下载.....	202
28.2 在线仿真.....	202
28.3 调试模式.....	202
29 电气特性.....	203
29.1 极限参数.....	203
29.2 直流电气特性.....	203
29.3 ESD/EFT 特性.....	206
29.4 内部 RC 时钟温度特性.....	206
29.5 ADC 电气特性.....	207

29.6 LDO 电压输出管脚 (3V_OUT) 特性	207
30 封装类型	208
31.1 封装形式:	208

上海锦钛科技

1 概述

CA1020 系列是一款高性能的彩屏电子烟专用芯片，采用高性能的 ARM Cortex®-M0 的 32 位内核，工作主频 64MHz 频率（最高可支持 72MHz），SRAM 18K 字节，兼容外挂 1M-256M Bit 的 SPI - NOR Flash，支持 1/2/4 线 QSPI 通讯模式，支持升级外部 flash 程序。支持 320 X 240 像素彩屏扫描，速率最高可达 25 帧/秒。内置 4 路空气传感器支持外部 MIC 空气开关检测；内置 4 路专用电子烟防干烧测阻专用管脚；2 路模拟比较器用于过流过压保护功能；还内置了 28 路 12 Bit ADC、12 路触摸按键、4 组互补型 PWM、I2C、3 组 UART、3 组 4 线 SPI、低电压检测（LVD），内置 LDO 电压输出可配置 3.3V - 2.6V 输出，可节省外部 LDO。

工作温度范围-40 至+105℃，宽电压工作范围 1.8V 至 5.5V，支持 Sleep、Stop 和低速运行三种省电模式以适应不同功耗要求的应用。强大的功能及优越的抗干扰性能使其可广泛应用于 TFT 彩屏电子烟、TFT 智能家居控制面板、TFT 家电控制面板。

2 基本特性

◆ 内核

- CPU: ARM Cortex® - M0 的 32 位内核
- 主频 64MHz 频率（最高可支持 72MHz）

◆ 存储器

- 支持外置 1 线、2 线、4 线 1M - 256MB Bit SPI - Nor Flash
- SRAM: 18KB

◆ 工作电压

- 工作电压: 1.8 - 5.5V

◆ 工作温度

- 工作温度: -40℃ - +105℃

◆ 时钟系统

- 外部低速振荡器: 32.768KHz
- 内置低速 RC 振荡器: 96KHz
- 内置高速 RC 振荡器: 16MHz，精度为±1%@5V/25℃（出厂校正）
- 内置 PLL 锁相环

◆ 内置 RTC

- 支持闹钟事件，周期性唤醒，可实现年、月、日及时钟功能
- 支持毫秒中断，秒中断，闹钟中断。

◆ 定时器 (TIMER)

- 1 个 16 位 8 通道高级定时器用于 8 通道 PWM 输出，带死区时间发生器
- 支持 4 组互补型 PWM 应用
- 1 个 16 位通用定时器，每个定时器多达 2 路输入捕获或输出比较通道。
- 2 个 16 位基本定时器和 SysTick 定时器
- 独立、窗口看门狗定时器

◆ 中断系统

- 支持嵌套中断向量，可软件配置 4 级优先级
- 支持 GPIO、IWDG、ADC、UART0、UART1、UART2 定时器、WWDG、I2C、SPI、PWM 和 SYSTICK、LVD 作为中断源
- 所有 GPIO 都支持外部中断唤醒功能。支持上升沿、下降沿、双沿中断

◆ DMA

- 2 个两通道的 DMA
- 支持 TIMER、UART、SPI、ADC 等硬件触发 DMA 请求
- 通道优先级支持软件配置
- 独立的数据源地址和目的数据地址；数据传输宽度可以字节、半字、字；
- 支持循环的缓冲器管理
- DMA 支持：存储器到存储器、外设到存储器、存储器到外设

◆ 输入输出管脚 (GPIO)

- 最多支持 45 个 GPIO 口，支持推挽、开漏、上拉、下拉、高阻模式
- 内置上拉电阻 10K，下拉电阻 15K。弱上拉电阻 45K，弱下拉电阻 45K
- 推电流支持：4mA、8mA、12mA、16mA 四级可设置
灌电流支持：4mA、10mA、18mA、24mA 四级可设置
- 4 个通道 (PA22~PA25) 管脚，可用于发热丝测阻应用 (和比较器管脚共用)
- 支持 13 个 3.3V 供电管脚，用于外部 flash 和 TFT 彩屏通讯及控制
- 内置一路 3.3V 的 LDO 电源输出管脚，可设置关断、3.3V/3.0V/2.8V/2.6V 电压输出给外设供电，节省外部 LDO 器件

◆ 模/数转换器 (ADC)

- 1 个 28 通道的 12 位 ADC
- 支持 3 种基准电压源：VDD、内部基准、外部基准
- 支持单次模式和连续模式采样
- 选择内部电压为基准电压时可测量 VDD 电压
- 内置比较功能，支持硬件触发启动转换功能

◆ 空气传感器

- 支持咪头空气开关检测功能，
- 支持硅咪、正装（驻极体 MIC）和反装（背极体 MIC）
- 支持咪头金属膜差异校正功能
- 支持 PFM 模式，增强客户体验
- 支持中断唤醒 MCU 功能

◆ 触摸按键（Touch Key）

- 内置触摸感应控制器
- 支持 12 触摸通道，支持按键、滑条、转圈触摸功能
- 支持触摸唤醒，省电模式

◆ 模拟比较器（COMP）

- 支持 2 组独立的模拟比较器
- 支持 64 级可编程输入 1V-VDD 阈值电压
- 支持短路保护功能
- 支持中断和延时功能
- 模拟比较器内接 ADC，可应用发热丝测阻功能

◆ SPI 接口

- 内置 3 个 SPI 接口，支持主从模式，最高支持 32M 速率
- *SPI-1 支持 1、2、4 线通讯，专用用于与外部 flash 通讯，内接 DMA，管脚 3V 供电
- *SPI-2 支持 1、2 线通讯，用于与 TFT 彩屏通讯，内接 DMA，管脚 3V 供电
- *SPI-3 支持 1、2 线通讯，可用于蓝牙 BLE 射频通信。内接 DMA，管脚 VDD 供电

◆ RGB 级联

- 支持 1 路级联 LED 驱动，扫描频率大于 400Hz/S，数据发送速度 800Kbps，直接控制 WS2812 或类似的驱动芯片，符合单色或七彩 LED 灯带产品的需求。

◆ 通用串行接口（UART）

- 支持 3 个 UART 接口
- 支持 1 字节接收缓存

◆ I2C 接口

- 内置 1 路 I2C 接口，支持主从模式，支持标准/快速模式。

◆ 低电压检测（LVD）

- 可配置电压检测范围 1.8 - 4.8V（每级 0.1V）
- 可设置低电压中断

◆ 低功耗模式

- 支持 STOP 模式
- MCU STOP 模式, LDO 关闭, 外部中断唤醒模式, 电流低至 10uA
- MCU STOP 模式, LDO 开启, 外部 Nor SPI FLASH 进入待机模式, 外部中断唤醒和 MIC 唤醒模式, 电流低至 16uA
- MCU 进入 STOP 模式, LDO 开启, 外部 Nor SPI FLASH 进入待机模式, 外部中断唤醒和 MIC 唤醒和触摸按键唤醒模式, 电流低至 20uA

◆ 串行两线仿真调试下载

- 支持两线下载 (PA0 - SWD, PA1 - SWC)
- 支持仿真功能
- 支持两线下载升级外部 SPI - FLASH, 方便产品开发和升级

◆ 封装形式: QFN48/QFN32

3 芯片型号功能介绍

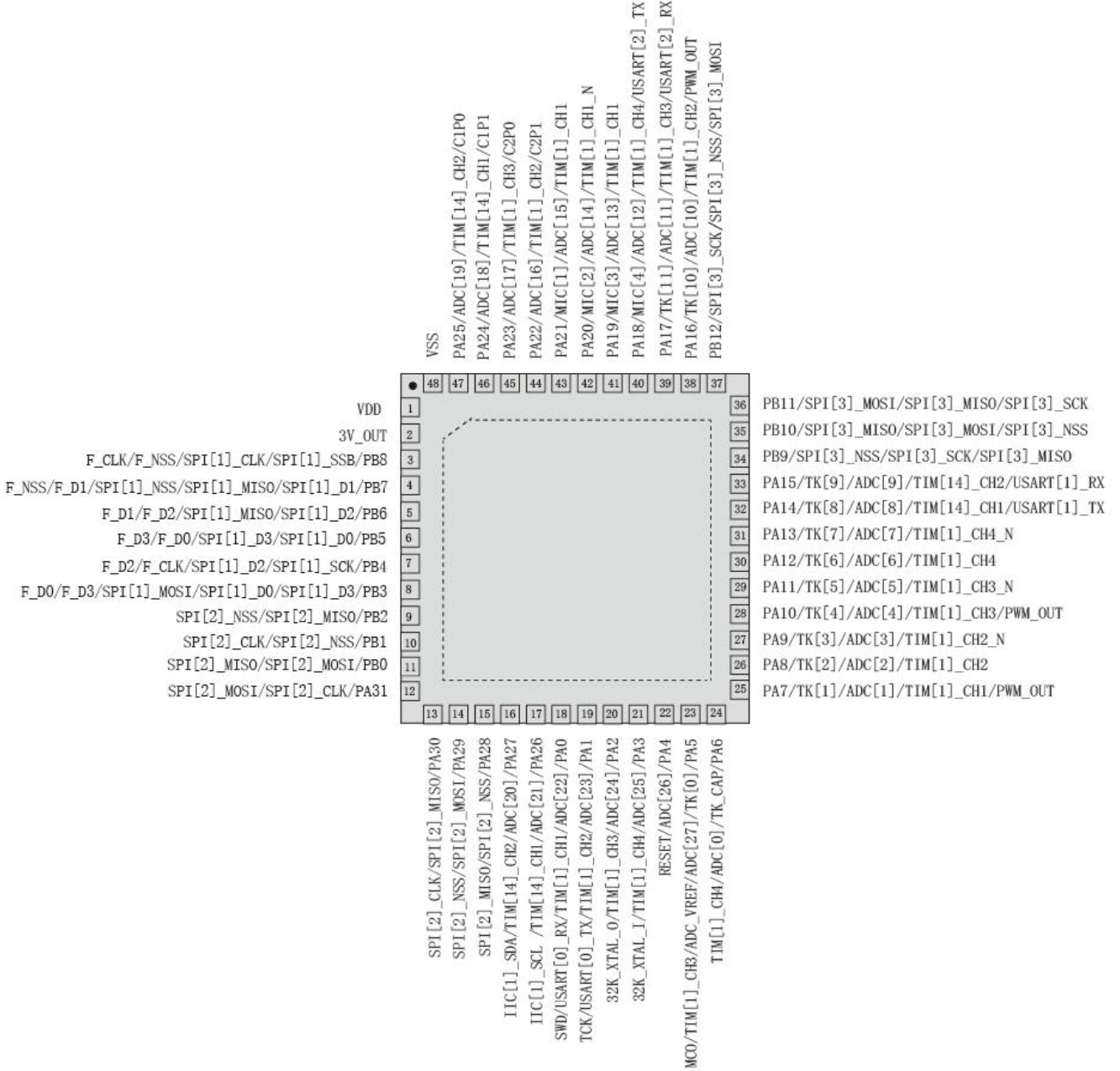
表 3-1 CA1020 系列具体型号功能特点

芯片型号	外置 Flash	SRam [BYTE]	内部高速 RC 振荡器	内部低速 RC 振荡器	外部振荡器 [32.768KHz]	GPIO 数量	通用 16 位定时器数量	UART 数量	I ² C	SPI	定时器 [PWM] 通道 (TIMER)	电容式触摸按键	12 位 ADC 通道数量	空气传感器通道	模拟比较器	片上仿真下载功能	工作电压 [V]	封装形式
CA1020N5	√	18K	√	√	√	29	3	2	-	2	15	8	16	2	√	√	1.8-5.5	QFN32 (4x4mm)
CA1020N6	√	18K	√	√	√	45	3	3	1	3	21	12	28	4	√	√	1.8-5.5	QFN48 (5x5mm)

4 引脚及其描述

4.1 封装定义

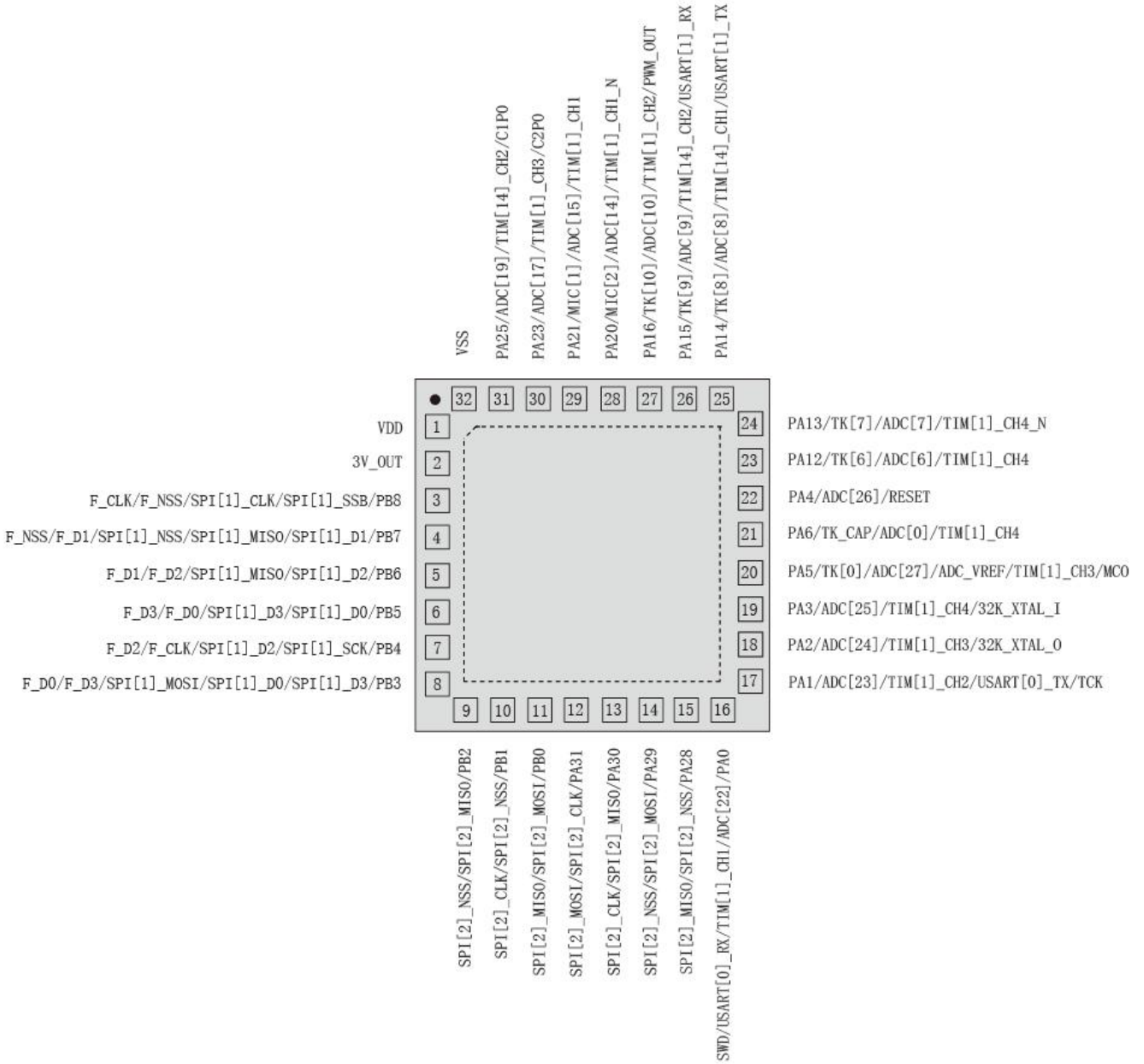
型号: CA1020N6



QFN48 封装引脚图

备注: 下载仿真管脚为: Pin18(SWD) / Pin19(TCK);

型号: CA1020N5



QFN32 封装引脚图

备注: 下载仿真管脚为: Pin16(SWD) / Pin17(TCK);

4.2 引脚描述

引脚描述

管脚名称			管脚功能		默认功能
QFP32	QFN48	电源域			
1	1	VDD	VDD	芯片供电管脚	VDD
2	2	3V	3V_OUT	LDO 电压输出 2.6V—3.3V 输出可调	3V_OUT
3	3	3V	PB8/SPI1_SSB/SPI1_CLK/F_NSS/F_CLK	通用双向 I/O 口 SPI1 SSB 端口 SPI1 CLK 端口 FLASH NSS 端口 FLASH CLK 端口	FLASH NSS
4	4	3V	PB7/SPI1_D1/SPI1_MISO/SPI1_NSS/F_D1/F_NSS	通用双向 I/O 口 SPI1 D1 端口 SPI1 MISO 端口 SPI1 NSS 端口 FLASH D1 端口 FLASH NSS 端口	FLASH D1
5	5	3V	PB6/SPI1_D2/SPI1_MISO/F_D2/F_D1	通用双向 I/O 口 SPI1 D2 端口 SPI1 MISO 端口 FLASH D2 端口 FLASH D1 端口	FLASH D2
6	6	3V	PB5/SPI1_D0/SPI1_D3/F_D0/D_D3	通用双向 I/O 口 SPI1 D0 端口 SPI1 D3 端口 FLASH D0 端口 FLASH D3 端口	FLASH D0
7	7	3V	PB4/SPI1_SCK/SPI1_D2/F_CLK/F_D2	通用双向 I/O 口 SPI1 SCK 端口 SPI1 D2 端口 FLASH CLK 端口 FLASH D2 端口	FLASH CLK
8	8	3V	PB3/SPI1_D3/SPI1_D0/SPI1_MOSI/F_D3/F_D0	通用双向 I/O 口 SPI1 D3 端口 SPI1 D0 端口 SPI1 MOSI FLASH D3 端口 FLASH D0 端口	FLASH D3
9	9	3V	PB2/SPI2_MISO/SPI2_NSS	通用双向 I/O 口 SPI2 MISO 端口 SPI2 NSS 端口	高阻
10	10	3V	PB1/SPI2_NSS/SPI2_CLK	通用双向 I/O 口 SPI2 NSS 端口 SPI2 CLK 端口	高阻
11	11	3V	PB0/SPI2_MOSI/SPI2_MISO	通用双向 I/O 口 SPI2 MOSI 端口 SPI2 MISO 端口	高阻

12	12	3V	PA31/SPI2_CLK/SPI2_MOSI	通用双向 I/O 口 SPI2 CLK 端口 SPI2 MOSI 端口	高阻
13	13	3V	PA30/SPI2_MISO/SPI2_CLK	通用双向 I/O 口 SPI2 MISO 端口 SPI2 CLK 端口	高阻
14	14	3V	PA29/SPI2_MOSI/SPI2_NSS	通用双向 I/O 口 SPI2 MOSI 端口 SPI2 NSS 端口	高阻
15	15	3V	PA28/SPI2_NSS/SPI2_MISO	通用双向 I/O 口 SPI2 NSS 端口 SPI2 MISO 端口	高阻
	16	VDD	PA27/ADC_20/TIM14_CH2/IIC_SDA	通用双向 I/O 口 ADC 20 通道 TIM14 CH2 通道 IIC SDA 端口	高阻
	17	VDD	PA26/ADC_21/TIM14_CH1/IIC_SCL	通用双向 I/O 口 ADC 21 通道 TIM14 CH1 通道 IIC SCL 端口	高阻
16	18	VDD	PA0/ADC_22/TIM1_CH1/UART_RX/SWD	通用双向 I/O 口 ADC 22 通道 TIM1 CH1 通道 UART RX 端口 SWD (SWDIO) 下载端口	SWD (SWDIO)
17	19	VDD	PA1/ADC_23/TIM1_CH2/UART_TX/TCK	通用双向 I/O 口 ADC 23 通道 TIM1 CH2 通道 UART TX 端口 TCK (SWCLK) 下载端口	TCK (SWCLK)
18	20	VDD	PA2/ADC_24/TIM1_CH3/32K_XTAL_O	通用双向 I/O 口 ADC 24 通道 TIM1 CH3 通道 HSE 外部低速晶振输出 端口	高阻
19	21	VDD	PA3/ADC_25/TIM1_CH4/32K_XTAL_O	通用双向 I/O 口 ADC 25 通道 TIM1 CH4 通道 HSE 外部低速晶振 (时钟) 输入端口	高阻
22	22	VDD	PA4/ADC_26/RESET	通用双向 I/O 口 ADC 26 通道 RESET 端口 (低电平复位)	RESET (复位) (内部上拉)
20	23	VDD	PA5/TK_0/ADC_27/ADC_VREF/TIM1_C H3/MCO	通用双向 I/O 口 TK 0 通道 ADC 27 通道 ADC VREF 管脚 TIM1 CH3 通道 MCO 管脚	高阻
21	24	VDD	PA6/TK_CAP/ADC_0/TIM1_CH4	通用双向 I/O 口 TK CAP 管脚 ADC 0 通道 TIM1 CH4 通道	高阻

	25	VDD	PA7/TK_1/ADC_1/TIM1_CH1/RGB_OUT	通用双向 I/O 口 TK 1 通道 ADC 1 通道 TIM1 CH1 通道 级联 RGB 输出管脚	高阻	
	26	VDD	PA8/TK_2/ADC_2/TIM1_CH2	通用双向 I/O 口 TK 2 通道 ADC 2 通道 TIM1 CH2 通道	高阻	
	27	VDD	PA9/TK_3/ADC_3/TIM1_CH2N	通用双向 I/O 口 TK 3 通道 ADC 3 通道 TIM1 CH2 互补通道	高阻	
	28	VDD	PA10/TK_4/ADC_4/TIM1_CH3/RGB_OUT	通用双向 I/O 口 TK 4 通道 ADC 4 通道 TIM1 CH3 通道 级联 RGB 输出管脚	高阻	
	29	VDD	PA11/TK_5/ADC_5/TIM1_CH3N	通用双向 I/O 口 TK 5 通道 ADC 5 通道 TIM1 CH5 互补通道	高阻	
23	30	VDD	PA12/TK_6/ADC_6/TIM1_CH4	通用双向 I/O 口 TK 6 通道 ADC 6 通道 TIM1 CH4 通道	高阻	
	24	31	VDD	PA13/TK_7/ADC_7/TIM1_CH4N	通用双向 I/O 口 TK 7 通道 ADC 7 通道 TIM1 CH4 互补通道	高阻
	25	32	VDD	PA14/TK_8/ADC_8/TIM14_CH1/UART1_TX	通用双向 I/O 口 TK 8 通道 ADC 8 通道 TIM14 CH1 通道 UART1 TX 管脚	高阻
	26	33	VDD	PA15/TK_9/ADC_9/TIM14_CH2/UART1_RX	通用双向 I/O 口 TK 9 通道 ADC 9 通道 TIM14 CH2 通道 UART1 RX 管脚	高阻
	34	VDD	PB9/SPI3_NSS/SPI3_SCK/SPI3_MISO	通用双向 I/O 口 SPI3 NSS 管脚 SPI3 SCK 管脚 SPI3 MISO 管脚	高阻	
	35	VDD	PB10/SPI3_MISO/SPI3_MOSI/SPI3_NSS	通用双向 I/O 口 SPI3 MISO 管脚 SPI3 MOSI 管脚 SPI3 NSS 管脚	高阻	
	36	VDD	PB11/SPI3_MOSI/SPI3_MISO/SPI3_SCK	通用双向 I/O 口 SPI3 MOSI 管脚 SPI3 MISO 管脚 SPI3 SCK 管脚	高阻	
	37	VDD	PB12/SPI3_SCK/SPI3_NSS/SPI3_MOSI	通用双向 I/O 口 SPI3 SCK 管脚 SPI3 NSS 管脚 SPI3 MOSI 管脚	高阻	
27	38	VDD	PA16/TK_10/ADC_10/TIM1_CH2/RGB_OUT	通用双向 I/O 口 TK 10 通道 ADC 10 通道	高阻	

				TIM1 CH2 通道 级联 RGB OUT 管脚	
	39	VDD	PA17/TK_11/ADC_11/TIM1_CH3/UART2_RX	通用双向 I/O 口 TK 11 通道 ADC 11 通道 TIM1 CH3 通道 UART2 RX 管脚	高阻
	40	VDD	PA18/MIC_4/ADC_12/TIM1_CH4/UART2_TX	通用双向 I/O 口 MIC 4 通道 ADC 12 通道 TIM1 CH4 通道 UART2 TX 管脚	高阻
	41	VDD	PA19/MIC_3/ADC_13/TIM1_CH1	通用双向 I/O 口 MIC 3 通道 ADC 13 通道 TIM1 CH3 通道	高阻
28	42	VDD	PA20/MIC_2/ADC_14/TIM1_CH1N	通用双向 I/O 口 MIC 2 通道 ADC 14 通道 TIM1 CH1 互补通道	高阻
29	43	VDD	PA21/MIC_1/ADC_15/TIM1_CH1	通用双向 I/O 口 MIC 1 通道 ADC 15 通道 TIM1 CH1 通道	高阻
	44	VDD	PA22/ADC_16/TIM1_CH2/C2P1	通用双向 I/O 口 ADC 16 通道 TIM1 CH2 通道 C2P1 比较器 2 输入管脚 1	高阻
30	45	VDD	PA23/ADC_17/TIM1_CH3/C2P0	通用双向 I/O 口 ADC 17 通道 TIM1 CH3 通道 C2P0 比较器 2 输入管脚 0	高阻
	46	VDD	PA24/ADC_18/TIM1_CH1/C1P1	通用双向 I/O 口 ADC 18 通道 TIM1 CH1 通道 C1P1 比较器 1 输入管脚 1	高阻
31	47	VDD	PA25/ADC_19/TIM1_CH2/C1P0	通用双向 I/O 口 ADC 19 通道 TIM1 CH2 通道 C1P0 比较器 1 输入管脚 0	高阻
32	48	VSS	VSS	芯片供电管脚	VSS

5 存储器和总线架构

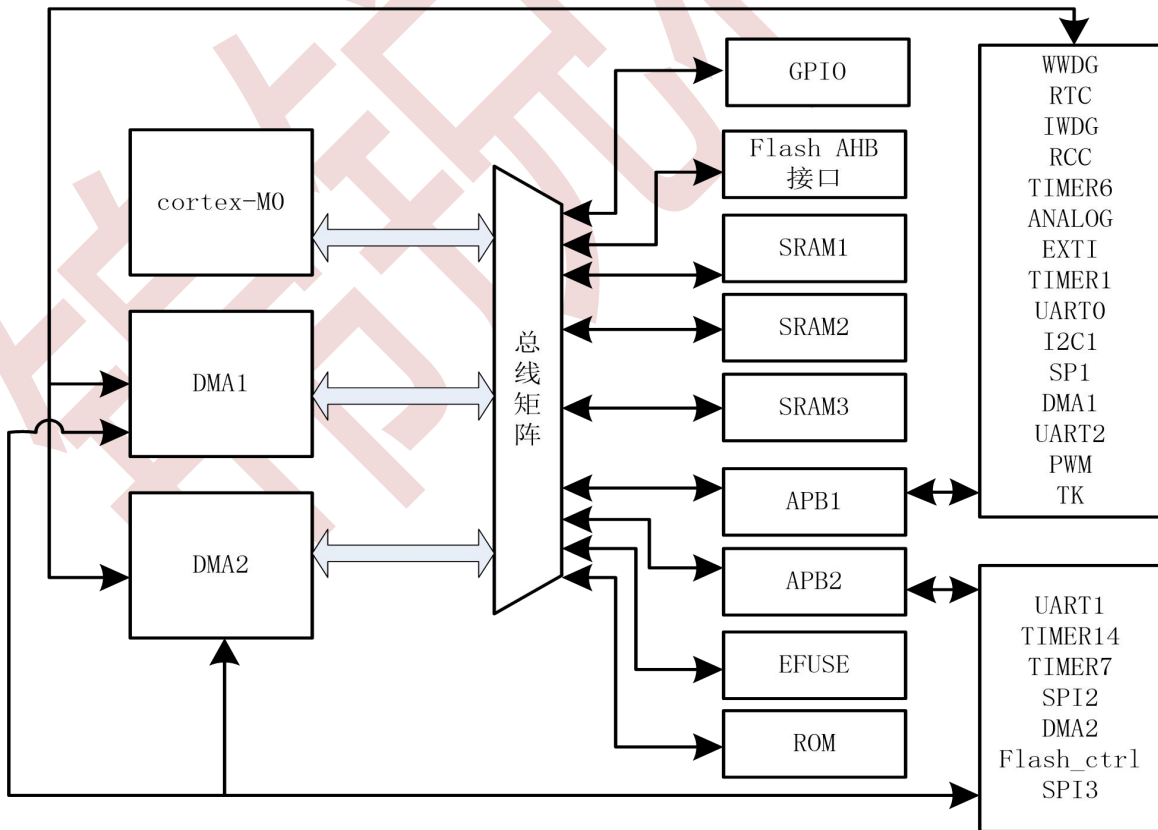
5.1 系统架构

5.1.1 总线架构

主系统由以下部分构成：

- 三个主驱动单元：
 - ◆ Cortex®-M0 内核系统总线
 - ◆ 通用 DMA
- 四个被动单元
 - ◆ 内部3个SRAM
 - ◆ 内部启动程序存储器 ROM
 - ◆ AHB 到 AHB 的桥，它连接一些 AHB 设备
 - ◆ AHB 到 APB 的桥(AHB2APBx, x=1,2),它连接所有的 APB 设备

备这些都是通过一个多级的 AHB 总线构架相互连接。



- CPU 系统总线：连接 Cortex®-M0 内核的总线到总线矩阵，用来指令预取，数据加载（常量加载和调试访问）及 AHB/APB 外设访问。
- DMA 总线：DMA 的 AHB 主控接口连接到总线矩阵，总线矩阵协调着内核和 DMA 到 SRAM、闪存和外设的访问。
- 总线矩阵协调内核系统总线和 DMA 主控总线之间的访问仲裁，仲裁利用轮算法。总线矩阵包含 3 个驱动部件（CPU 的系统总线、2 个 DMA 总线）和 6 个从部件（闪存存储器接口、SRAM、ADC 和 AHB 系统总线 1/2）。AHB 一些外设通过总线矩阵与系统总线 1 相连，系统总线 2 连接 2 个 AHB2APB 桥。
- 系统包含 2 个 AHB2APB 桥，即 AHB2APB1 和 AHB2APB2。其中 APB1 包含 15 个 APB 外设，PCLK 的最高速度为 72MHz；APB2 包含 7 个 APB 外设，PCLK 最高速度为 72MHz。

5.1.2 存储器组织

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个线性地址空间内。

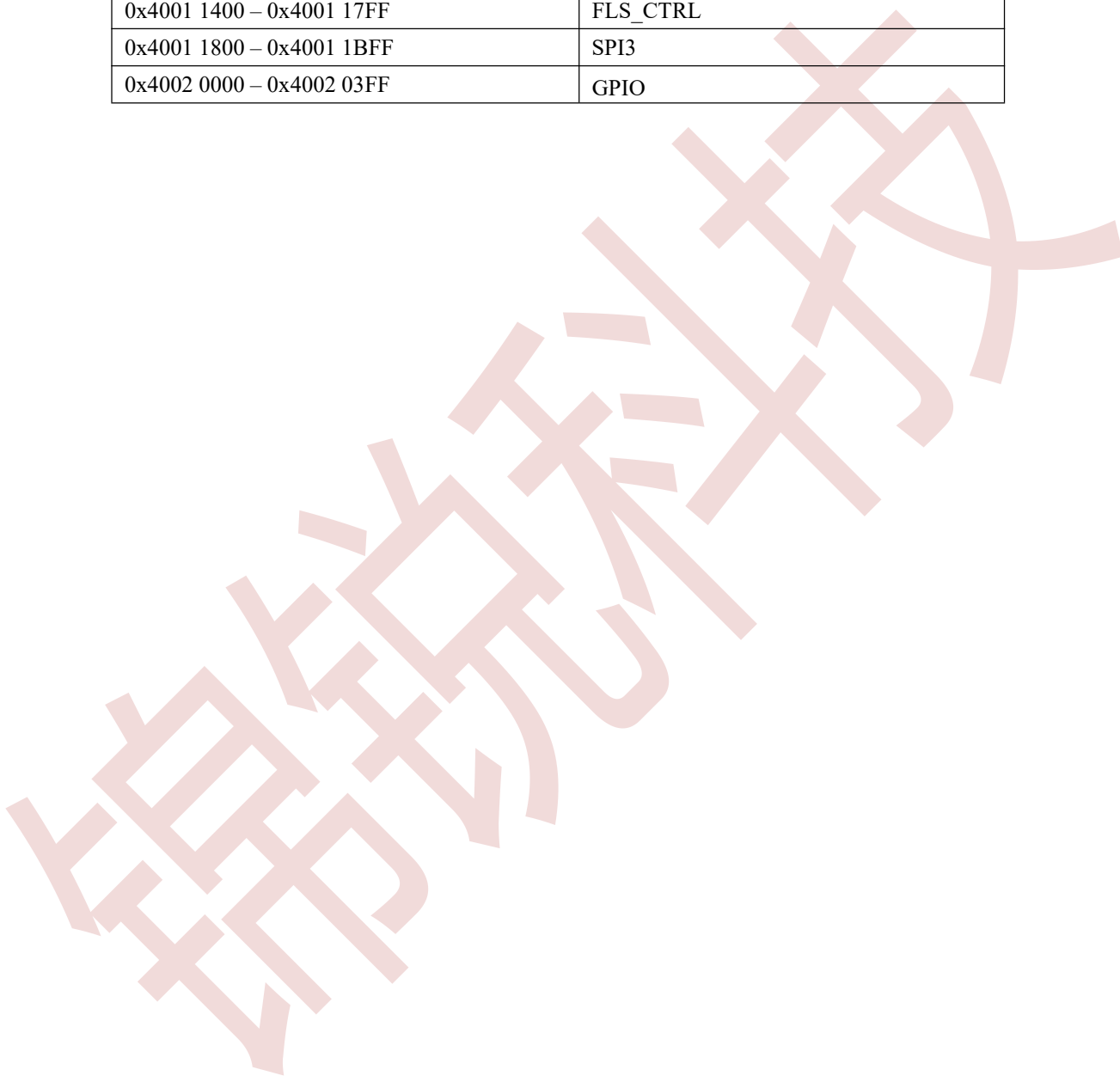
数据字节以小端格式存放在存储器中。一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。

5.1.3 存储器映像

请参考相应器件的数据手册中的存储器映像图。CA1020 中内置外设的起始地址。

起始地址	外设
0x0800 0000 – 0x09FF FFFF	外挂 FLASH
0x1FFF 8000 – 0x1FFF 800C	EFUSE
0x0000 0000 – 0x0000 07FF	ROM
0x2000 0000 – 0x2000 3FFF	SRAM1
0x2000 8000 – 0x2000 83FF	SRAM2
0x2000 9000 – 0x2000 93FF	SRAM3
0x4000 0000 – 0x4000 03FF	WWDG
0x4000 0400 – 0x4000 07FF	RTC
0x4000 0800 – 0x4000 0BFF	IWDG
0x4000 0C00 – 0x4000 0FFF	保留
0x4000 1000 – 0x4000 13FF	RCC
0x4000 1400 – 0x4000 17FF	TIM6
0x4000 1800 – 0x4000 1BFF	ANALOG
0x4000 1C00 – 0x4000 1FFF	EXTI
0x4000 2000 – 0x4000 23FF	TIM1
0x4000 2400 – 0x4000 27FF	UART0
0x4000 2800 – 0x4000 2BFF	I2C1
0x4000 2C00 – 0x4000 2FFF	SPI1
0x4000 3000 – 0x4000 33FF	DMA1
0x4000 3400 – 0x4000 37FF	UART2

0x4000 3800 – 0x4000 3BFF	PWM
0x4000 3C00 – 0x4000 3FFF	TK
0x4001 0000 – 0x4001 03FF	UART1
0x4001 0400 – 0x4001 07FF	TIM14
0x4001 0800 – 0x4001 0BFF	TIM7
0x4001 0C00 – 0x4001 0FFF	SPI2
0x4001 1000 – 0x4001 13FF	DMA2
0x4001 1400 – 0x4001 17FF	FLS_CTRL
0x4001 1800 – 0x4001 1BFF	SPI3
0x4002 0000 – 0x4002 03FF	GPIO



6 通用输入输出口（GPIO）及复用定义

6.1 概述

GPIO（General purpose input/output）即通用型 I/O，AFIO（Alternate-function input/output）即复用功能 I/O。芯片 GPIO 共被分为 2 组（GPIOA/GPIOB）。GPIO 端口和其他的复用外设共用引脚，用户可以根据需求灵活配置。每个 GPIO 引脚都可以独立配置成输出、输入或复用的外设功能端口。PA22~PA25 引脚是大推电流管脚。

GPIO 端口具有以下特性：

■GPIO 端口可由软件分别配置成以下模式：

- ◆ 输入浮空
- ◆ 输入上拉
- ◆ 输入下拉
- ◆ 模拟功能
- ◆ 推挽复用功能及上/下拉可配
- ◆ 开漏复用功能及上/下拉可配

■单独的位设置或位清除功能

■所有 I/O 支持外部中断功能

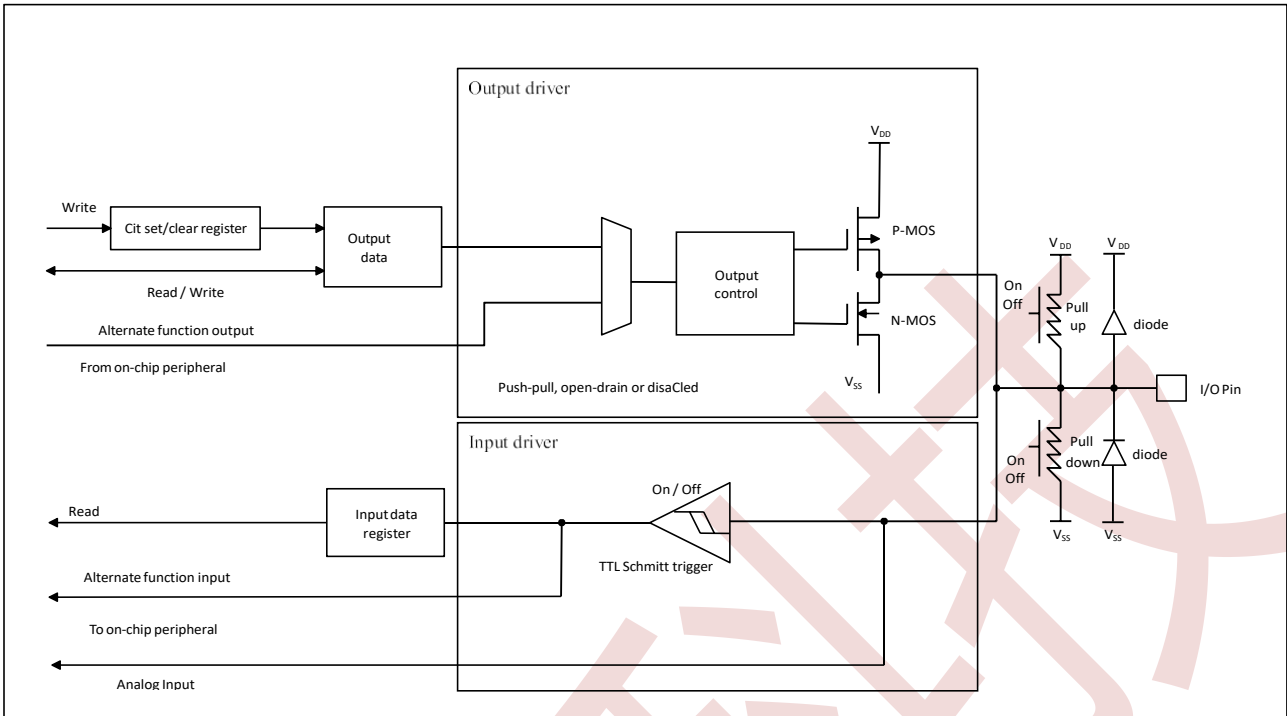
■所有 I/O 支持低功耗模式唤醒，上升或下降沿可配置

- ◆ 16 个 EXTI 可用于 SLEEP 或 STOP 模式唤醒，所有 I/O 可复用为 EXTI

■支持软件重新映射 I/O 复用功能

每个 I/O 端口位可以任意编程，但必须按照 32 位字访问 I/O 端口寄存器（不允许 16 位半字或 8 位字节访问）。下图给出了一个 I/O 端口的基本结构。

图5-1 I/O 端口的基本结构



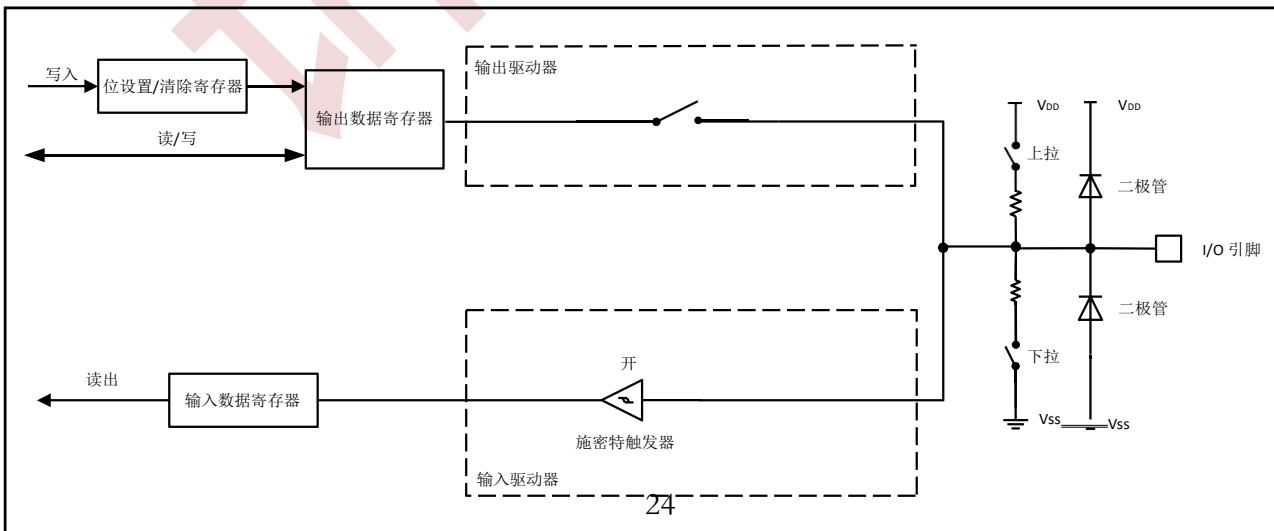
6.2 功能描述

6.2.1 输入模式

当I/O 端口配置为输入模式时：

- 输出缓冲器被禁止
- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PU_EN/GPIOx_PD_EN 寄存器的配置
- 对输入数据寄存器的读访问得到 I/O 状态

图 5-2 输入 浮空/上拉/下拉模式

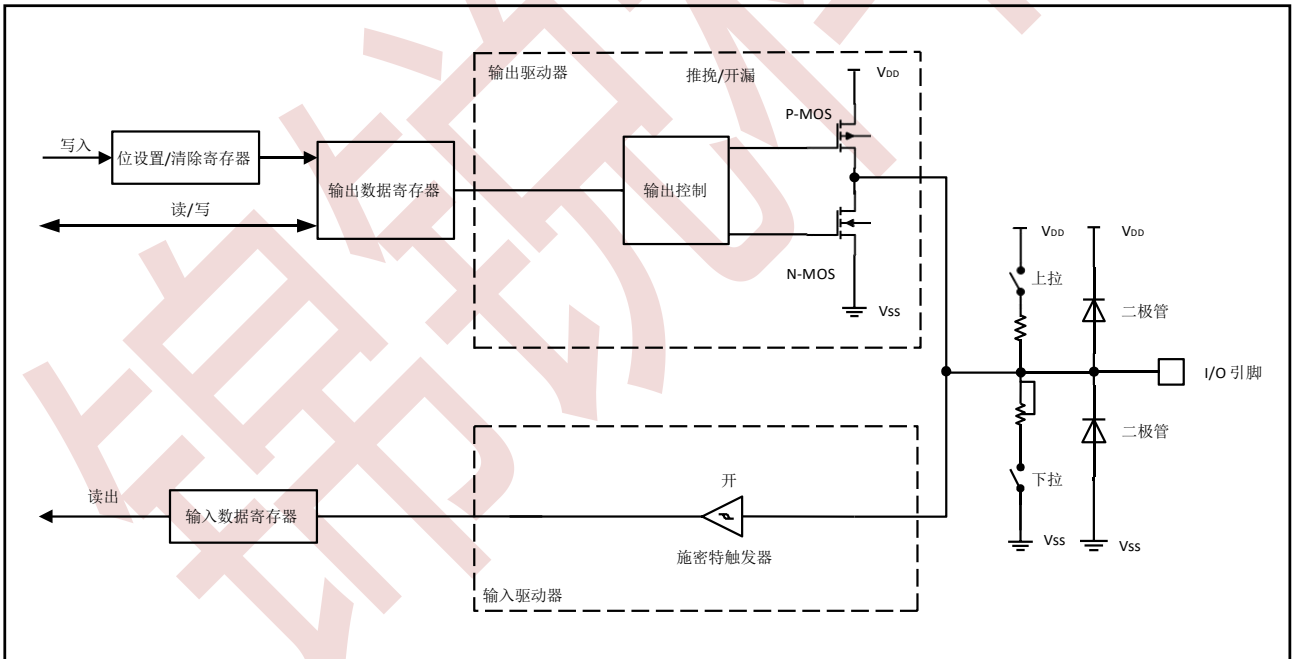


6.2.2 输出模式

当I/O 端口配置为输出模式时：

- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PU_EN/GPIOx_PD_EN 寄存器的配置
- 输出缓冲器被激活
 - ◆ 开漏模式： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平
输出数据寄存器上的'1'使端口置于高阻状态（P-MOS 从不被激活）
 - ◆ 推挽模式： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平
输出数据寄存器上的'1'激活 P-MOS，引脚输出高电平
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后写入的值

图 5-3 输出模式



6.2.3 复用功能模式

当I/O 端口配置为复用功能模式时：

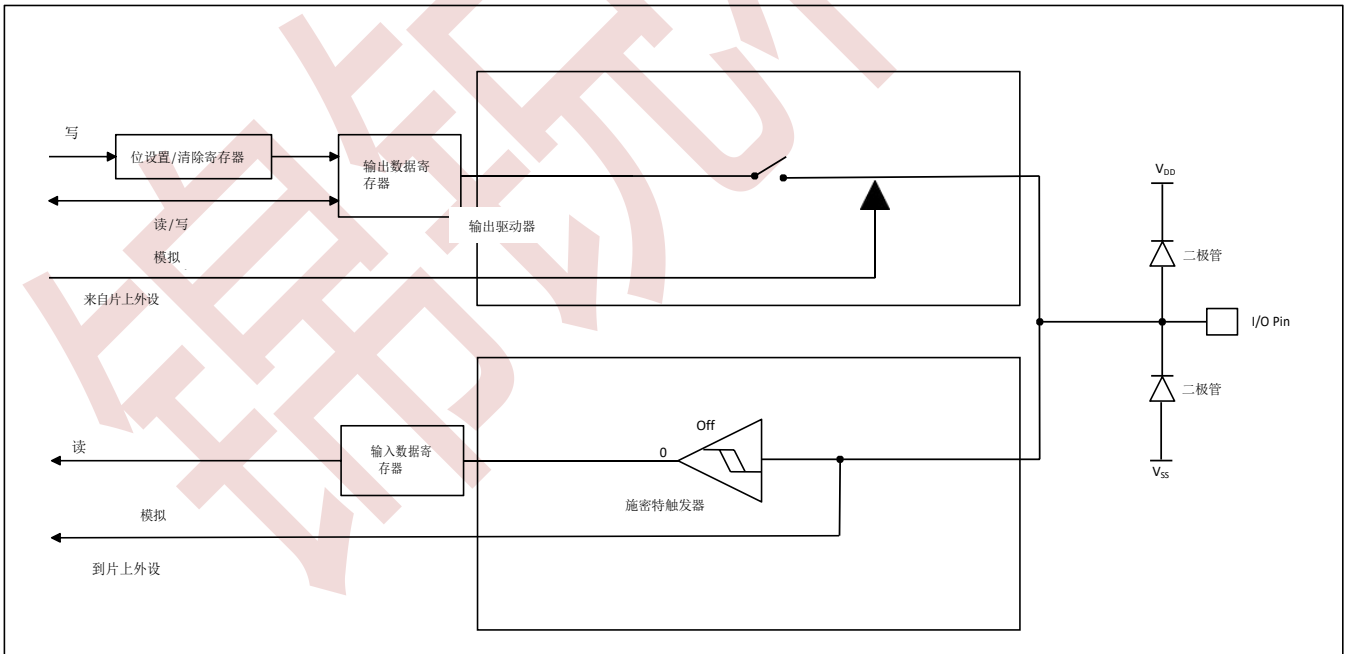
- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx_PU_EN/PD_EN 寄存器的配置
- 在开漏或推挽式配置中，输出缓冲器由外设控制
- 内置外设的信号驱动输出缓冲器
- 在每个时钟周期，出现在 输入驱动上的数据被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后写入的值

6.2.4 模拟功能模式

当I/O 端口被配置为模拟功能模式时：

- 上拉和下拉电阻被禁止
- 读取输入数据寄存器时数值为'0'
- 输出缓存器被禁止施密特触发输入被禁止，输出值被强置为'0'（实现了每个模拟 I/O 引脚上的零消耗）

图 5-4 高阻抗的模拟功能模式



6.3 复位后状态

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成高阻。

但有以下几个例外的信号：

- NRST 默认无 GPIO 功能：
 - ◆ PA4:NRST 复用模式
- 复位后，调试系统相关的引脚默认配置为 SWD 接口 I/O 配置：
 - ◆ PA1: SWCLK 置于复用模式
 - ◆ PA0: SWDIO 置于复用模式

6.4 外部中断/唤醒线

所有端口都有外部中断能力，可以在 EXTI 模块中配置：

- 端口必须配置成输入模式
- 所有端口可配置用于 SLEEP/STOP 模式唤醒，支持上升或下降沿可配
- 通用 I/O 端口连接到 16 个外部中断/事件线上，由寄存器 SYSCFG->EXTICR[4] 配置

6.5 复用功能

当 I/O 端口被配置为复用功能模式时，使用前必须对端口位配置寄存器 GPIOx_OD_EN(GPIOx[15])，GPIOx_PU_EN/PD_EN(GPIOx[7:6])，GPIOx_PU_SEL/PD_SEL(GPIOx[13:12]) 和 GPIOx_AF(GPIOx[2:0]) (x=A, B) 设置，复用输入或输出由外设确定。

6.6 软件 I/O 复用表

为拓展不同器件封装下的复用外设功能灵活性，同一个引脚可以复用为不同的功能。每个 IO 有多达 8 个可复用的功能 (AF0~ AF7)。复位后，可以通过软件配置相应的寄存器 (GPIOx_AF) 来重新映射 IO 复用功能。

引脚复用功能映射表

取值 名称	0	1	2	3	4	5	6	7
PA0	高阻	数字输入	数字输出	SWD	TIM1_CH1	UART0_RX	ADC_22	
PA1	高阻	数字输入	数字输出	TCK	TIM1_CH2	UART0_TX	ADC_23	
PA2	高阻	数字输入	数字输出		TIM1_CH3	XTAL_O	ADC_24	
PA3	高阻	数字输入	数字输出		TIM1_CH4	XTAL_I	ADC_25	
PA4	高阻	数字输入	数字输出			RESET	ADC_26	
PA5	高阻	数字输入	数字输出	TK0	TIM1_CH3	MCO	ADC_27	ADC_VREF
PA6	高阻	数字输入	数字输出	TK_CAP	TIM1_CH4		ADC_0	
PA7	高阻	数字输入	数字输出	TK1	TIM1_CH1	RGB_OUT	ADC_1	
PA8	高阻	数字输入	数字输出	TK2	TIM1_CH2		ADC_2	
PA9	高阻	数字输入	数字输出	TK3	TIM1_CH2N		ADC_3	
PA10	高阻	数字输入	数字输出	TK4	TIM1_CH3	RGB_OUT	ADC_4	
PA11	高阻	数字输入	数字输出	TK5	TIM1_CH3N		ADC_5	
PA12	高阻	数字输入	数字输出	TK6	TIM1_CH4		ADC_6	
PA13	高阻	数字输入	数字输出	TK7	TIM1_CH4N		ADC_7	
PA14	高阻	数字输入	数字输出	TK8	TIM14_CH1	UART1_TX	ADC_8	
PA15	高阻	数字输入	数字输出	TK9	TIM14_CH2	UART1_RX	ADC_9	
PA16	高阻	数字输入	数字输出	TK10	TIM1_CH2	RGB_OUT	ADC_10	
PA17	高阻	数字输入	数字输出	TK11	TIM1_CH3	UART2_RX	ADC_11	
PA18	高阻	数字输入	数字输出	MIC_4	TIM1_CH3	UART2_TX	ADC_12	
PA19	高阻	数字输入	数字输出	MIC_3	TIM1_CH1		ADC_13	
PA20	高阻	数字输入	数字输出	MIC_2	TIM1_CH1N		ADC_14	
PA21	高阻	数字输入	数字输出	MIC_1	TIM1_CH1		ADC_15	
PA22	高阻	数字输入	数字输出		TIM1_CH2	C2PI	ADC_16	
PA23	高阻	数字输入	数字输出		TIM1_CH3	C2PO	ADC_17	
PA24	高阻	数字输入	数字输出		TIM14_CH1	C1PI	ADC_18	
PA25	高阻	数字输入	数字输出		TIM14_CH2	C1PO	ADC_19	
PA26	高阻	数字输入	数字输出		TIM14_CH1	I2C1_SCL	ADC_21	
PA27	高阻	数字输入	数字输出		TIM14_CH2	I2C1_SDA	ADC_20	
PA28	高阻	数字输入	数字输出	SPI2_NSS	SPI2MISO			
PA29	高阻	数字输入	数字输出	SPI2_MOSI	SPI2_NSS			
PA30	高阻	数字输入	数字输出	SPI2_MISO	SPI2_CLK			
PA31	高阻	数字输入	数字输出	SPI2_CLK	SPI2_MOSI			
PB0	高阻	数字输入	数字输出	SPI2_MOSI	SPI2_MISO			
PB1	高阻	数字输入	数字输出	SPI2_NSS	SPI2_CLK			
PB2	高阻	数字输入	数字输出	SPI2_MISO	SPI2_NSS			
PB3	高阻	数字输入	数字输出	SPI1_D3	SPI1_MOSI	FLASH_D3	FLASH_D0	
PB4	高阻	数字输入	数字输出	SPI1_SCK	SPI1_D2	FLASH_CLK	FLASH_D2	
PB5	高阻	数字输入	数字输出	SPI1_D0	SPI1_D3	FLASH_D0	FLASH_D3	
PB6	高阻	数字输入	数字输出	SPI1_D2	SPI1_MISO	FLASH_D2	FLASH_D1	
PB7	高阻	数字输入	数字输出	SPI1_MISO	SPI1_NSS	FLASH_D1	FLASH_NSS	
PB8	高阻	数字输入	数字输出	SPI1_SSB	SPI1_CLK	FLASH_NSS	FLASH_CLK	
PB9	高阻	数字输入	数字输出	SPI3_NSS	SPI3_SCK	SPI3_MISO		
PB10	高阻	数字输入	数字输出	SPI3_MISO	SPI3_MOSI	SPI3_NSS		
PB11	高阻	数字输入	数字输出	SPI3_MOSI	SPI3_MISO	SPI3_SCK		
PB12	高阻	数字输入	数字输出	SPI3_SCK	SPI3_NSS	SPI3_MOSI		

6.7 引脚寄存器描述

地址偏移	寄存器缩写	寄存器名	复位值
0x00	GPIOA0	GPIOA0 配置寄存器	0x7F1B
0x04	GPIOA1	GPIOA1 配置寄存器	0x7F1B
0x08	GPIOA2	GPIOA2 配置寄存器	0x7F18
0x0C	GPIOA3	GPIOA3 配置寄存器	0x7F18
0x10	GPIOA4	GPIOA4 配置寄存器	0x7F1D
0x14	GPIOA5	GPIOA5 配置寄存器	0x7F18
0x18	GPIOA6	GPIOA6 配置寄存器	0x7F18
0x1C	GPIOA7	GPIOA7 配置寄存器	0x7F18
0x20	GPIOA8	GPIOA8 配置寄存器	0x7F18
0x24	GPIOA9	GPIOA9 配置寄存器	0x7F18
0x28	GPIOA10	GPIOA10 配置寄存器	0x7F18
0x2C	GPIOA11	GPIOA11 配置寄存器	0x7F18
0x30	GPIOA12	GPIOA12 配置寄存器	0x7F18
0x34	GPIOA13	GPIOA13 配置寄存器	0x7F18
0x38	GPIOA14	GPIOA14 配置寄存器	0x7F18
0x3C	GPIOA15	GPIOA15 配置寄存器	0x7F18
0x40	GPIOA16	GPIOA16 配置寄存器	0x7F18
0x44	GPIOA17	GPIOA17 配置寄存器	0x7F18
0x48	GPIOA18	GPIOA18 配置寄存器	0x7F18
0x4C	GPIOA19	GPIOA19 配置寄存器	0x7F18
0x50	GPIOA20	GPIOA20 配置寄存器	0x7F18
0x54	GPIOA21	GPIOA21 配置寄存器	0x7F18
0x58	GPIOA22	GPIOA22 配置寄存器	0x7F38
0x5C	GPIOA23	GPIOA23 配置寄存器	0x7F38
0x60	GPIOA24	GPIOA24 配置寄存器	0x7F38
0x64	GPIOA25	GPIOA25 配置寄存器	0x7F38
0x68	GPIOA26	GPIOA26 配置寄存器	0x7F18
0x6C	GPIOA27	GPIOA27 配置寄存器	0x7F18
0x70	GPIOA28	GPIOA28 配置寄存器	0x7F18
0x74	GPIOA29	GPIOA29 配置寄存器	0x7F18
0x78	GPIOA30	GPIOA30 配置寄存器	0x7F18
0x7C	GPIOA31	GPIOA31 配置寄存器	0x7F18
0x80	GPIOB0	GPIOB0 配置寄存器	0x7F18
0x84	GPIOB1	GPIOB1 配置寄存器	0x7F18
0x88	GPIOB2	GPIOB2 配置寄存器	0x7F18
0x8C	GPIOB3	GPIOB3 配置寄存器	0x7F1A
0x90	GPIOB4	GPIOB4 配置寄存器	0x7F1D
0x94	GPIOB5	GPIOB5 配置寄存器	0x7F1D
0x98	GPIOB6	GPIOB6 配置寄存器	0x7F1A
0x9C	GPIOB7	GPIOB7 配置寄存器	0x7F1D

0xA0	GPIOB8	GPIOB8 配置寄存器	0x7F1D
0xA4	GPIOB9	GPIOB9 配置寄存器	0x7F18
0xA8	GPIOB10	GPIOB10 配置寄存器	0x7F18
0xAC	GPIOB11	GPIOB11 配置寄存器	0x7F18
0xB0	GPIOB12	GPIOB12 配置寄存器	0x7F18
0xB4	保留	保留	保留
0xB8	保留	保留	保留
0xBC	保留	保留	保留
0xC0	GPIOA0	GPIOA 输出寄存器	0x0
0xC4	GPIOB0	GPIOB 输出寄存器	0x0
0xC8	GPIOA1	GPIOA 输入寄存器	0x0
0xCC	GPIOB1	GPIOB 输入寄存器	0x0

6.7.1 GPIO 配置寄存器

地址偏移: 0x00~0x0B0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD_EN	SMIT_EN	PU_SEL	PD_SEL	DRVN[1:0]		SR[1:0]		PU_EN	PD_EN	DRV_P2	DRV_P1[1:0]		AF[2:0]		
rw	rw	rw	rw	rw		rw		rw	rw		rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD_EN	SMIT_EN	PU_SEL	PD_SEL	DRVN[1:0]		SR[1:0]		PU_EN	PD_EN	DRV_P2	DRV_P1[1:0]		AF[2:0]		
rw	rw	rw	rw	rw		rw		rw	rw		rw		rw		

位 15	Open Drain 使能, 为 1 选择 OD, 为 0 选择推挽输出
位 14	为 1 输入的 SMIT 使能, 为 0 输入的反相器使能
位 13	为 1 时上拉电阻 10K, 为 0 时上拉 45K 电阻
位 12	为 1 时下拉电阻 15K, 为 0 时下拉 45K 电阻
位 11:10	DRVN: 灌电流输出强度选择 00:4mA 01:10mA 10:18mA 11:24mA
位 9:8	输出斜率控制: 00: 最慢斜率控制 01: 10: 11: 最快斜率控制
位 7	为 1 时上拉电阻使能, 0 不使能
位 6	为 1 时下拉电阻使能, 0 不使能
位 5	1: 强 0: 弱 注: 只有 GPIOA22, GPIOA23, GPIOA24, GPIOA25 此位有效
位 4:3	DRV_P: 推电流输出强度选择: 00:4mA 01:8mA 10:12mA 11:16mA
位 2:0	引脚复用选择, 见引脚复用功能映射表

6.7.2 GPIOA 输出数据寄存器 (GPIOAO)

地址偏移: 0x0C0

复位值: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	12	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPIOAO[31:0]																																
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0	GPIOAO0~31: 输出配置寄存器 0: 输出低电平 1: 输出高电平
--------	---

6.7.3 GPIOB 输出数据寄存器 (GPIOBO)

地址偏移: 0x0C4

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留			GPIOBO[12:0]													
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 12:0	GPIOBO0~12: 输出配置寄存器 0: 输出低电平 1: 输出高电平
--------	---

6.7.4 GPIOA 输入数据寄存器 (GPIOAI)

地址偏移: 0x0C8

复位值: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	12	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPIOAOI[31:0]																																
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0	GPIOAOI0~31: 输入配置寄存器 0: 输入低电平 1: 输入高电平
--------	--

6.7.5 GPIOB 输入数据寄存器 (GPIOBI)

地址偏移: 0x0Cc

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			GPIOBI[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 12:0			GPIOBO0~12: 输入配置寄存器 0: 输入低电平 1: 输入高电平												

7 外部 FLASH 控制器

7.1 FLS 简介

内核通过 FLS 控制器来访问外部的 SPI NOR FLASH，并向外部 SPI NOR FLASH 读取程序运行过程中所需的指令。FLS 支持 1M-256M bit 的外部 SPI NOR FLASH 访问读取。支持外部 SPI NOR FLASH 的三地址以及 4 地址模式，支持外部 SPI NOR FLASH 的单线、双线、四线读取。

注意：设置 FLS 控制器的代码必须全部放在 SRAM 中执行。

7.2 FLS 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	FLSCMD	FLASH 命令寄存器	0xFFFFFFFF
0x04	FLSCR	FLASH 控制寄存器	0x00000001

7.2.1 FLSCMD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CSHICYE		ADR4SEL	DUMMY_CYCLE				RDCMD							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	保留
位 14:13	CSHICYE 表示 CS 信号高电平时间超过一个时钟周期时配置，需要几个就配置几个周期，默认值为 1，CS 高电平 2 个周期，该寄存器可以不需要关闭 SPIEN 修改
位 12	ADR4SEL 0: 设置 FLASH 读工作于三地址模式 1: 是指 FLASH 读工作于四地址模式
位 11:8	DUMMY_CYCLE 快速读取指令需要 DUMMY CYCLE 周期配置 0x0 表示 1 个周期 ... 0xF 表示 16 个周期
位 7:0	RDCMD: 代表 SPI FLASH 支持的命令 03H: 三字节地址模式单线读 13H: 三字节地址模式双线读 BBH: 三字节地址模式四线读 BCH: 四字节地址模式单线读 EBH: 四字节地址模式双线读 ECH: 四字节地址模式四线读

7.2.1 FLSCR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															SPIEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:1	保留
位 0	<p>SPIEN</p> <p>0: FLS 接口失能</p> <p>1: FLASH 接口使能，结合 FLSCMD 寄存器的设置，内核可以直接读取外部 SPI 接口的 FLASH 中的数据。</p>

8 中断和事件

8.1 嵌套向量中断寄存器

特性

- 22 个可屏蔽中断通道（不包含 16 个 Cortex®-M0 的中断线）。
- 4 个可编程的优先等级（使用了 2 位中断优先级）；
- 低延迟的异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；

嵌套向量中断控制器（NVIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。嵌套向量中断控制器管理着包括内核异常等中断。

8.1.1 SysTick 校准值寄存器

系统嘀嗒校准值固定为系统时钟，当系统时钟设定为 PCLK/1000，产生 1ms 时间基准。

8.1.2 中断和异常向量

表 8-3-1 中断向量表

位置	优先级	优先级类型	名称	说明	地址
-	-	-	-	保留 (Reserved)	0x0000 0000
-	-3	固定	Reset	复位 (Reset)	0x0000 0004
-	-2	固定	NMI	不可屏蔽中断。RCC 时钟安全系统 (CSS) 连接到 NMI 向量。	0x0000 0008
-	-1	固定	HardFault	所有类型的错误 (fault)	0x0000 000C
-	3	可设置	SVCALL	通过 SWI 指令调用的系统服务	0x0000 002C
-	5	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000 0040
1	8	可设置	-	-	0x0000 0044
2	9	可设置	RTC	RTC 中断 (联接 EXTI 线 17)	0x0000 0048
3	10	可设置	-	-	0x0000 004C
4	11	可设置	MICINT	-	0x0000 0050
5	12	可设置	EXTI0_1	EXTI 线[1:0] 中断	0x0000 0054
6	13	可设置	EXTI2_3	EXTI 线[3:2] 中断	0x0000 0058
7	14	可设置	EXTI4_15	EXTI 线[15:4] 中断	0x0000 005C

8	15	可设置	TK	触摸中断（连接 EXTI 线 16）	0x0000 0060
9	16	可设置	DMA1_CH0	DMA1 通道 0 中断	0x0000 0064
10	17	可设置	DMA1_CH1	DMA1 通道 1 中断	0x0000 0068
11	18	可设置	DMA2_CH0	DMA2 通道 0 中断	0x0000 006C
12	19	可设置	DMA2_CH1	DMA2 通道 1 中断	0x0000 0070
13	20	可设置	TIM1_UP_TRG	TIM1 更新、触发中断	0x0000 0074
14	21	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000 0078
15	22	可设置	PWM	PWM 中断	0x0000 007C
16	23	可设置	SPI2	SPI2 中断	0x0000 0080
17	24	可设置	TIM6	TIM6 中断	0x0000 0084
18	25	可设置	LVD	LVD 中断	0x0000 0088
19	26	可设置	TIM14_UP_TRG	TIM14 更新、触发中断	0x0000 008C
20	27	可设置	ADC0	ADC0 全局中断	0x0000 0090
21	28	可设置	SPI3	SPI3 中断	0x0000 0094
22	29	可设置	TIM7	TIM7 中断	0x0000 0098
23	30	可设置	I2C1	I2C1 全局中断	0x0000 009C
24	31	可设置	CMP	CMP 全局中断	0x0000 00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000 00A4
26	33	可设置	UART0	UART0 全局中断	0x0000 00A8
27	34	可设置	UART1	UART1 全局中断	0x0000 00AC
28	35	可设置	UART2	UART2 全局中断	0x0000 00B0
29	36	可设置	0x5975616E	固定保留数据	0x0000 00B4
30	37	可设置	0x486F6E67	固定保留数据	0x0000 00B8
31	38	可设置	-	-	0x0000 00BC

8.2 外部中断/事件控制器（EXTI）

8.2.1 简介

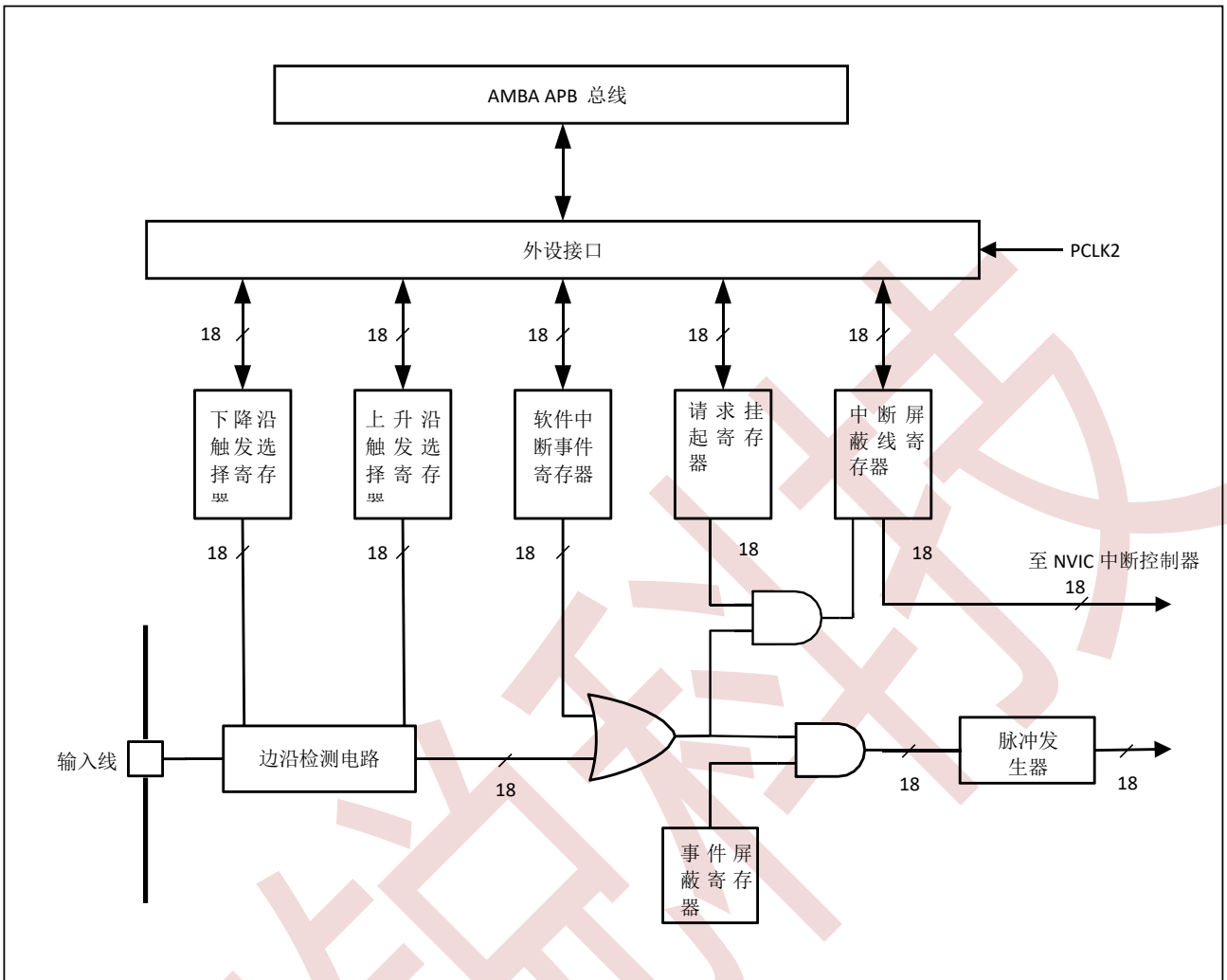
外部中断/事件控制器包含 18 个产生中断/事件触发的边沿检测电路，每条输入线可以独立地配置脉冲或挂起输入类型，以及上升沿、下降沿或者双边沿 3 种触发事件类型，也可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求，可通过在挂起寄存器的对应位写'1'操作，清除中断请求。

8.2.2 主要特性

EXTI 控制器的主要特性如下：

- 支持 18 个软件中断/事件请求
- 每条输入线对应的中断/事件都能独立配置触发或屏蔽
- 每条中断线都有独立的状态位
- 支持脉冲或挂起输入类型
- 支持上升沿、下降沿或双边沿 3 种触发事件类型
- 可唤醒退出低功耗模式

图6-1 外部中断/事件控制器框图



8.2.3 功能描述

要产生中断，必须配置外部中断控制器的 NVIC 中断通道使能相应的中断线。通过沿触发配置寄存器 EXTI_RTISR 和 EXTI_FTISR 选择上升沿、下降沿或双边沿触发事件类型，并将中断屏蔽寄存器 EXTI_IMR 的相应位写‘1’开放允许中断请求。当外部中断线上检测到预设的边沿触发极性，将产生一个中断请求，对应的挂起位也随之被置‘1’。在挂起寄存器的对应位写‘1’，将清除该中断请求。

要产生事件，必须配置并使能对应的事件线。根据需要的边沿检测极性，设置上升/下降沿触发配置寄存器，同时在事件屏蔽寄存器的相应位写‘1’允许中断请求。当事件线上发生预设的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置‘1’。

另外，通过在软件中断/事件寄存器写‘1’，也可以通过软件产生中断/事件请求。

- 硬件中断配置，根据需要选择配置 18 条线路作为中断源：
 - ◆ 配置 18 条中断线的屏蔽位 (EXTI_IMR)；
 - ◆ 配置所选中断线的触发配置位 (EXTI_RTISR 和 EXTI_FTISR)；
 - ◆ 配置对应到外部中断控制器的 NVIC 中断通道的使能和屏蔽位，使 18 条中断线中的请求可以被正确地响应。

- 硬件事件配置，根据需要选择配置 18 条线路作为事件源：
 - ◆ 配置 17 条事件线的屏蔽位（EXTI_EMR）；
 - ◆ 配置所选事件线的触发配置位（EXTI_RTZR 和EXTI_FTZR）。
- 软件中断/事件配置，根据需要选择配置 18 条线路作为软件中断/事件线：
 - ◆ 配置 17 条中断/事件线屏蔽位（EXTI_IMR,EXTI_EMR）；
 - ◆ 配置软件中断事件寄存器的请求位（EXTI_SWIER）。

8.2.4 EXTI 线路映射

表 6-2 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PA0;PA16;PB0	EXTI_CR 寄存器中的 Exti0_sel
EXTI1	PA1;PA17;PB1	EXTI_CR 寄存器中的 Exti1_sel
EXTI2	PA2;PA18;PB2	EXTI_CR 寄存器中的 Exti2_sel
EXTI3	PA3;PA19;PB3	EXTI_CR 寄存器中的 Exti3_sel
EXTI4	PA4;PA20;PB4	EXTI_CR 寄存器中的 Exti4_sel
EXTI5	PA5;PA21;PB5	EXTI_CR 寄存器中的 Exti5_sel
EXTI6	PA6;PA22;PB6	EXTI_CR 寄存器中的 Exti6_sel
EXTI7	PA7;PA23;PB7	EXTI_CR 寄存器中的 Exti7_sel
EXTI8	PA8;PA24;PB8	EXTI_CR 寄存器中的 Exti8_sel
EXTI9	PA9;PA25;PB9	EXTI_CR 寄存器中的 Exti9_sel
EXTI10	PA10;PA26;PB10	EXTI_CR 寄存器中的 Exti10_sel
EXTI11	PA11;PA27;PB11	EXTI_CR 寄存器中的 Exti11_sel
EXTI12	PA12;PA28	EXTI_CR 寄存器中的 Exti12_sel
EXTI13	PA13;PA29	EXTI_CR 寄存器中的 Exti13_sel
EXTI14	PA14;PA30	EXTI_CR 寄存器中的 Exti14_sel
EXTI15	PA15;PA31	EXTI_CR 寄存器中的 Exti15_sel

所有的 GPIO 均可用作 EXTI 的触发源用于产生中断或事件请求，通过配置 SYSCFG 章节的 SYSCFG_EXTICRx 寄存器，同时支持内部模块（包括 TK、RTC）触发。具体存在的连接关系如下表所示。

- EXTI 线16 连接到 TK 唤醒事件
- EXTI 线17 连接到 RTC 全局中断事件

8.3 中断控制寄存器

表 8-4-2 寄存器 IP

地址偏移	寄存器缩写	寄存器名	复位值
0x00	EXTI_IMR	中断屏蔽寄存器	0x00000000
0x04	EXTI_EMR	事件屏蔽寄存器	0x00000000
0x08	EXTI_RTSTR	上升沿触发选择寄存器	0x00000000
0x0C	EXTI_FTSTR	下降沿触发选择寄存器	0x00000000
0x10	EXTI_SWIER	软件中断事件寄存器	0x00000000
0x14	EXTI_PR	挂起寄存器	0x00000000
0x18	EXTI_CR	外部中断选择寄存器	0x00000000

8.3.1 中断屏蔽寄存器 (EXTI_IMR)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														IMR17	IMR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR4	IMR4	IMR3	IMR2	IMR1	IMR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 31:18		保留, 必须始终保持为复位状态 (0)。													
位 17:0		IMRx: 线 x 上的中断屏蔽 (Interrupt Mask on line x) 0: 屏蔽来自线 x 上的中断请求; 1: 开放来自线 x 上的中断请求。													

8.3.2 事件屏蔽寄存器 (EXTI_EMR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														EMR17	EMR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMR15	EMR14	EMR13	EMR12	EMR11	EMR10	EMR9	EMR8	EMR7	EMR6	EMR4	EMR4	EMR3	EMR2	EMR1	EMR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 31:18		保留, 必须始终保持为复位状态 (0)。													
位 17:0		EMRx: 线 x 上的事件屏蔽 (Event Mask on line x) 0: 屏蔽来自线 x 上的事件请求; 1: 开放来自线 x 上的事件请求。													

8.3.3 上升沿触发选择寄存器 (EXTI_RTISR)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														RTSR17	RTSR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTSR15	RTSR14	RTSR13	RTSR12	RTSR11	RTSR10	RTSR9	RTSR8	RTSR7	RTSR6	RTSR5	RTSR4	RTSR3	RTSR2	RTSR1	RTSR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18	保留, 必须始终保持为复位状态 (0)。
位 17:0	RTSRx: 线 x 上的上升沿触发事件配置位 (Rising trigger event configuration bit of line x) 0: 禁止输入线 x 上的上升沿触发 (中断和事件) 1: 允许输入线 x 上的上升沿触发 (中断和事件)

注意: 外部唤醒线是边沿触发的, 这些线上不能出现毛刺信号。
在写 EXTI_RTISR 寄存器时, 在外部中断线上的上升沿信号不能被识别, 挂起位也不会被置位。在同一中断线上, 可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

8.3.4 下降沿触发选择寄存器 (EXTI_FTISR)

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														FTSR17	FTSR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTSR15	FTSR14	FTSR13	FTSR12	FTSR11	FTSR10	FTSR9	FTSR8	FTSR7	FTSR6	FTSR5	FTSR4	FTSR3	FTSR2	FTSR1	FTSR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18	保留, 必须始终保持为复位状态 (0)。
位 17:0	FTSRx: 线 x 上的下降沿触发事件配置位 (Falling trigger event configuration bit of line x) 0: 禁止输入线 x 上的下降沿触发 (中断和事件) 1: 允许输入线 x 上的下降沿触发 (中断和事件)

注意: 外部唤醒线是边沿触发的, 这些线上不能出现毛刺信号。
在写 EXTI_FTISR 寄存器时, 在外部中断线上的下降沿信号不能被识别, 挂起位不会被置位。在同一中断线上, 可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

8.3.5 软件中断事件寄存器 (EXTI_SWIER)

偏移地址: 0x10

复位值: 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留														SWIER 17	SWIER 16
															rw	rw
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER9	SWIER8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER1	SWIER0
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 31:18	保留, 必须始终保持为复位状态 (0)。															
位 17:0	SWIERx: 线 x 上的软件中断 (Software interrupt on line x) 当该位为'0'时, 写'1'将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断, 则此时将产生一个中断。 注: 通过清除 EXTI_PR 的对应位 (写入'1'), 可以清除该位为'0'。															

8.3.6 挂起寄存器 (EXTI_PR)

偏移地址: 0x14

复位值: 0xXXXX XXXX

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留														PR17	PR16
															rcw1	rcw1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1
位 31:18	保留, 必须始终保持为复位状态 (0)。															
位 17:0	PRx: 挂起位 0: 没有发生触发请求 1: 发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件, 该位被置'1'。在该位中写入'1'可以清除它, 也可以通过改变边沿检测的极性清除。															

8.3.7 外部中断选择寄存器 (EXTI_CR)

偏移地址: 0x18

复位值: 0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exti15_sel	Exti14_sel	Exti13_sel	Exti12_sel	Exti11_sel	Exti10_sel	Exti9_sel	Exti8_sel								
														rcw1	rcw1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Exti7_sel	Exti6_sel	Exti5_sel	Exti4_sel	Exti3_sel	Exti2_sel	Exti1_sel	Exti0_sel								
rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1	rc w1

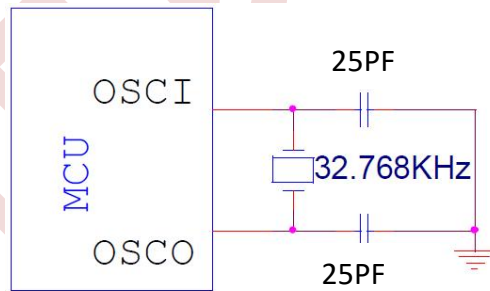
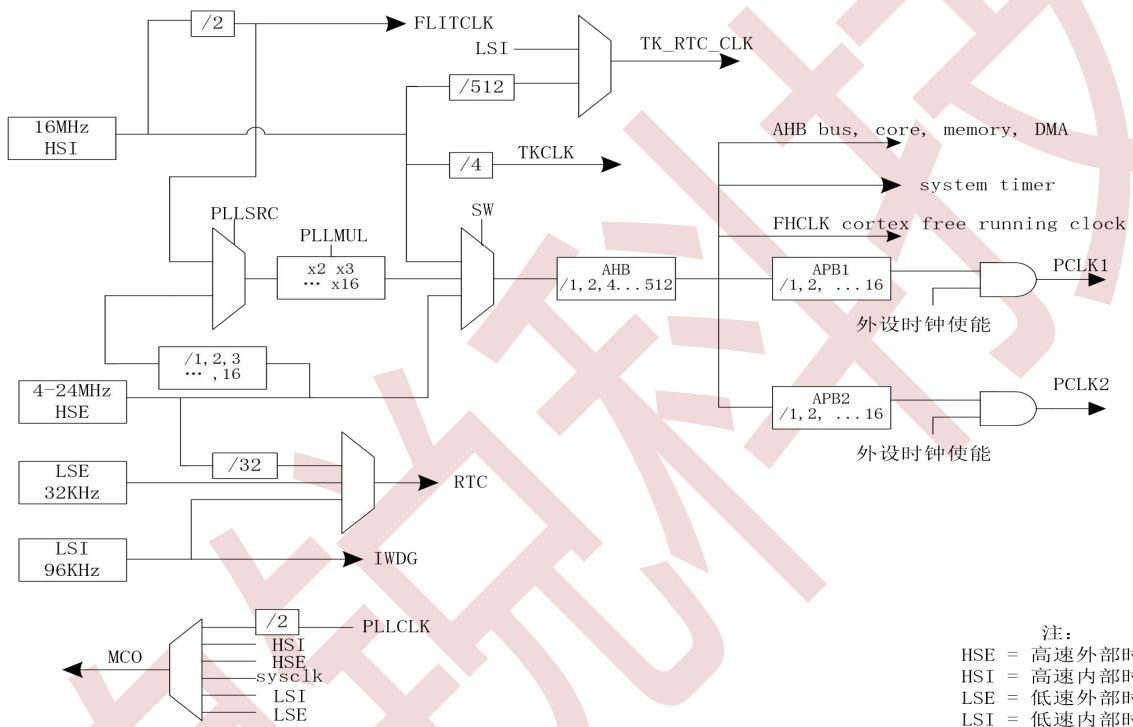
Extix_sel	外部中断选择寄存器 x: 对应的外部中断号 2'b00: 表示选择 ioa0~15 对应管脚外部中断; 2'b01: 表示选择 ioa16~31 对应管脚外部中断; 2'b10: 表示选择 iob0~4 对应管脚外部中断; 其他: 表示不选择任何 io 管脚作为外部中断;
-----------	---

9 时钟和电源系统

9.1 时钟系统介绍

CA1020 系列芯片共支持以下时钟源：

- 内置 16MHz RC 振荡器
- 内置 96KHz RC 振荡器
- 内置 PLL 锁相环
- 支持外部 32.768KHz 晶体振荡器



32.768KHz 参考典型电路图

- 重要提醒:**
1. 硬件设计时晶振负载电容必须和芯片地连接，晶振补偿电容尽量靠近芯片 GND 引脚。
32.768KHz 石英晶振必须使用直径 3mmx8mm 的晶振规格。
 2. 以上电路及元件参数仅供参考，使用不同厂家晶振在电路使用中参数可能需要修改。

9.2 时钟控制寄存器

地址偏移	寄存器缩写	寄存器名	复位值
0x00	RCC_CR	时钟控制寄存器	0x00008083
0x10	RCC_APB1RSTR	APB1 外设复位寄存器	0x00000000
0x1C	RCC_AHB1ENR	AHB1 外设时钟使能寄存器	0x00000014
0x24	RCC_CSR	控制/状态寄存器	0x0C000000

9.2.1 时钟控制寄存器 (RCC_CR)

偏移地址: 0x00

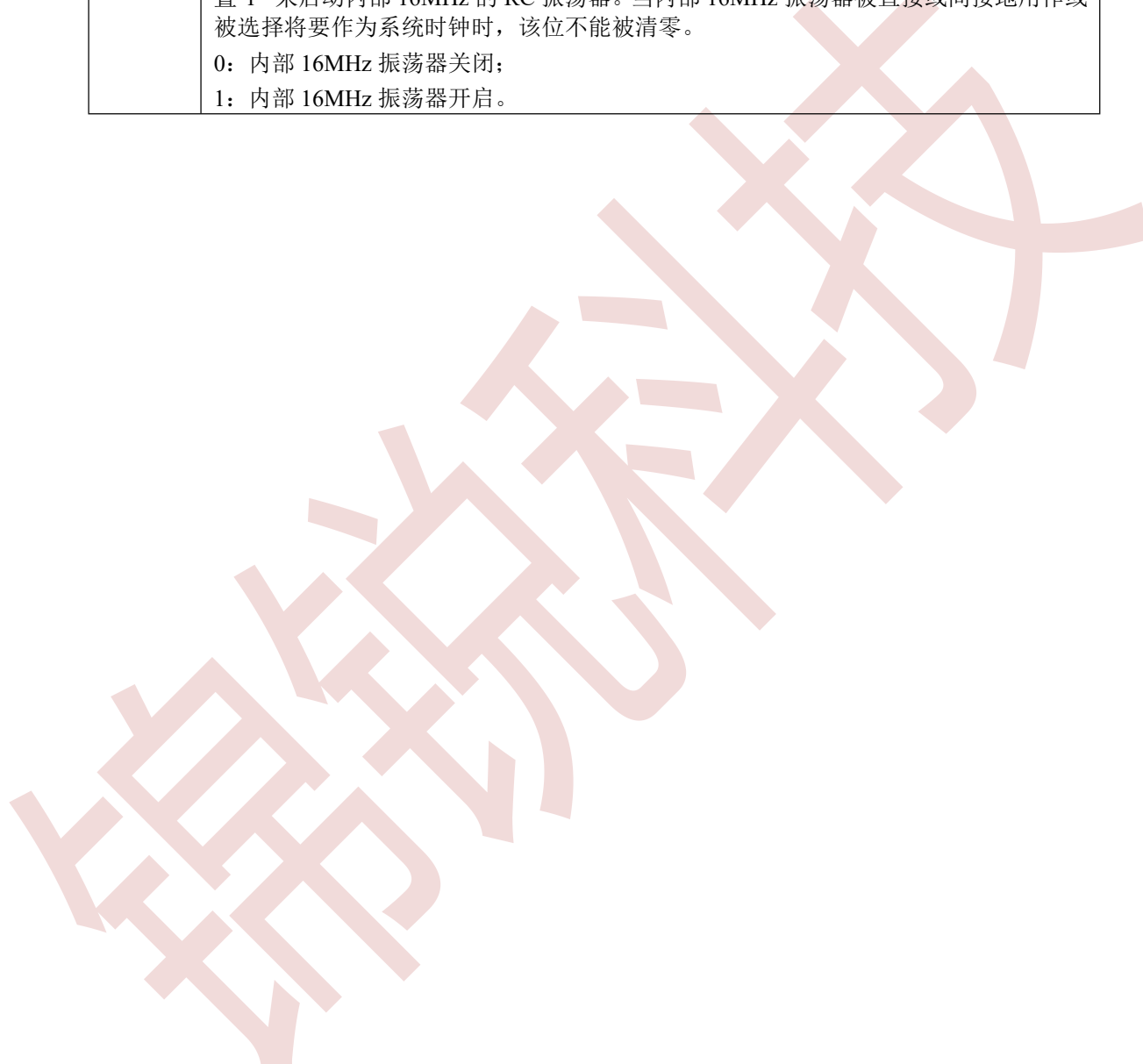
复位值: 0x00008083

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLLRDY	PLLOFN	PLLRDYF	HSIRDYF	LSERDYF	LSIRDYF	MCO		PLLMUL	
						r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLMUL		PPRE			HPRE			SWS		SW		HSIRDY	HSION		
												r	rw		

位 31:26	保留, 始终读为 0。
位 25	PLLRDY: PLL 时钟就绪标志 (PLL clock ready flag)PLL 锁定后由硬件置'1'。 0: PLL 未锁定; 1: PLL 锁定。
位 24	PLL 使能由软件置'1'或清零 当进入待机和停止模式时, 该位由硬件清零。当 PLL 时钟被用作或被选择将要作为系统时钟时, 该位不能被清零。 0: PLL 关闭 1: PLL 使能
位 23	PLL 就绪中断标志, 在 PLL 就绪且 PLLRDYIE 位被置'1'时, 由硬件置'1'。 由软件通过置'1'PLLRDYC 位来清除。 0: 无 PLL 上锁产生的时钟就绪中断; 1: PLL 上锁导致时钟就绪中断
位 22	HSI 就绪中断标志, 在内部高速时钟就绪且 HSIRDYIE 位被置'1'时, 由硬件置'1'。 由软件通过置'1'HSIRDYC 位来清除。 0: 无内部 8MHz RC 振荡器产生的时钟就绪中断; 1: 内部 8MHz RC 振荡器导致时钟就绪中断
位 21	LSE 就绪中断标志, 在外部低速时钟就绪且 LSERDYIE 位被置'1'时, 由硬件置'1'。 由软件通过置'1'LSERDYC 位来清除。 0: 无外部 32kHz 振荡器产生的时钟就绪中断; 1: 外部 32kHz 振荡器导致时钟就绪中断
位 20	LSI 就绪中断标志, 在内部低速时钟就绪且 LSIRDYIE 位被置'1'时, 由硬件置'1'。

	<p>由软件通过置'1' LSIRDYC 位来清除。</p> <p>0: 无内部 40kHz RC 振荡器产生的时钟就绪中断;</p> <p>1: 内部 40kHz RC 振荡器导致时钟就绪中断</p>
位 19~17	<p>MCO: 微控制器时钟输出由软件置'1'或清零</p> <p>0001: 没有时钟输出</p> <p>0010: LSI 时钟</p> <p>0011: LSE 时钟</p> <p>0100: 系统时钟 sysclk</p> <p>0101: HSI 时钟</p> <p>0111: PLL 时钟 2 分频后输出</p> <p>注意: 该时钟输出在启动和切换 MCO 时钟源时可能会被截断</p>
位 16~13	<p>PLL 倍频系数, 由软件设置来确定 PLL 倍频系数。只有在 PLL 关闭的情况下才可被写入。</p> <p>注意: PLL 的输出频率不能超过 64MHz</p> <p>0000: PLL 2 倍频输出 1000: PLL 10 倍频输出</p> <p>0001: PLL 3 倍频输出 1001: PLL 11 倍频输出</p> <p>0010: PLL 4 倍频输出 1010: PLL 12 倍频输出</p> <p>0011: PLL 5 倍频输出 1011: PLL 13 倍频输出</p> <p>0100: PLL 6 倍频输出 1100: PLL 14 倍频输出</p> <p>0101: PLL 7 倍频输出 1101: PLL 15 倍频输出</p> <p>0110: PLL 8 倍频输出 1110: PLL 16 倍频输出</p> <p>0111: PLL 9 倍频输出 1111: PLL 16 倍频输出</p>
位 12~10	<p>低速 APB 预分频 (APB1) 由软件置'1'或清'0'来控制低速 APB1 时钟 (PCLK1) 的预分频系数。注意: 软件必须保证 APB1 时钟频率不超过 32MHz。</p> <p>0xx: HCLK 不分频</p> <p>100: HCLK 2 分频</p> <p>101: HCLK 4 分频</p> <p>110: HCLK 8 分频</p> <p>111: HCLK 16 分频</p>
位 9~6	<p>AHB 预分频</p> <p>由软件置'1'或清'0'来控制 AHB 时钟的预分频系数。</p> <p>0xxx: SYSCLK 不分频</p> <p>1000: SYSCLK 2 分频 1100: SYSCLK 64 分频</p> <p>1001: SYSCLK 4 分频 1101: SYSCLK 128 分频</p> <p>1010: SYSCLK 8 分频 1110: SYSCLK 256 分频</p> <p>1011: SYSCLK 16 分频 1111: SYSCLK 512 分频</p>
位 5~4	<p>系统时钟切换状态, 由硬件置'1'或清'0'来指示哪一个时钟源被作为系统时钟。</p> <p>00: HSI 作为系统时钟;</p> <p>01: LSI 作为系统时钟;</p> <p>10: PLL 输出作为系统时钟;</p> <p>11: 不可用</p>
位 3~2	<p>系统时钟切换, 由软件置'1'或清'0'来选择系统时钟源。</p> <p>在从停止或待机模式中返回时或直接或间接作为系统时钟的 HSE 出现故障时, 由硬件强制选择</p> <p>HSI 作为系统时钟 (如果时钟安全系统已经启动)</p> <p>00: HSI 作为系统时钟;</p> <p>01: LSI 作为系统时钟;</p> <p>10: PLL 输出作为系统时钟;</p> <p>11: 不可用</p>

位 1	<p>HSIRDY: 内部高速时钟就绪标志 (Internal high-speed clock ready flag) 由硬件置'1'来指示内部 16MHz 振荡器已经稳定。在 HSION 位清零后, 该位需要 6 个内部 16MHz 振荡器周期清零。 0: 内部 16MHz 振荡器没有就绪; 1: 内部 16MHz 振荡器就绪。</p>
位 0	<p>HSION: 内部高速时钟使能 (Internal high-speed clock enable) 由软件置'1'或清零。 当从停止模式返回或用作系统时钟的外部 4-24MHz 振荡器发生故障时, 该位由硬件置'1' 来启动内部 16MHz 的 RC 振荡器。当内部 16MHz 振荡器被直接或间接地用作或被选择将要作为系统时钟时, 该位不能被清零。 0: 内部 16MHz 振荡器关闭; 1: 内部 16MHz 振荡器开启。</p>



9.2.2 APB1 外设复位寄存器 (RCC_APB1RSTR)

偏移地址: 0x10

复位值: 0x0000 0000

保留											SPI3RS T	SPI2RS T	TIM7R ST	TIM14 RST	UAR TIR ST
											rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TK RST	PWMR ST	UA RT2 RS T	保留	SPI1R ST	I2C 1R ST	UAR TORS T	TIM1 RST	保留	ANA RST	TIM6 RST	保留	PWRRS T	保留	保留	WW DGR ST
rw	rw	rw		rw	rw	rw	rw		rw	rw		rw			rw

位 21~31	保留，始终读为 0。
位 20	SPI3 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 19	SPI2 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 18	TIM7 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 17	TIM14 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 16	UART1 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 15	TK 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 14	PWM 复位 由软件置 '1' 或清 '0' 0: 无作用; 1: 复位
位 13	UART2 复位

	由软件置'1'或清'0'0: 无作用; 1: 复位
位 12	保留, 始终读为 0。
位 11	SPI1RST: SPI1 时钟复位 (SPI1 clock reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位 SPI1。
位 10	I2C1RST: I2C 1 复位 (I2C 1 reset) 由软件置'1'或清'0'0: 无作用; 1: 复位 I2C 1。
位 9	UART0RST: UART0 复位 由软件置'1'或清'0' 0: 无作用; 1: 复位 UART0。
位 8	TIM1 复位 由软件置'1'或清'0'0: 无作用; 1: 复位
位 7	保留, 始终读为 0。
位 6	ANA 复位 由软件置'1'或清'0'0: 无作用; 1: 复位
位 5	TIM6 复位 由软件置'1'或清'0'0: 无作用; 1: 复位
位 4	保留, 始终读为 0。
位 3	PWR 复位 由软件置'1'或清'0'0: 无作用; 1: 复位
位 1~2	保留, 始终读为 0。
位 0	WWDGRST: 窗口看门狗复位 由软件置'1'或清'0' 0: 无作用; 1: 复位窗口看门狗。

9.2.3 APB1 外设时钟使能寄存器 (RCC_APB1ENR)

偏移地址: 0x1C

复位值: 0x0000 0000

注: 当外设时钟没有启用时, 软件不能读出外设寄存器的数值, 返回的数值始终是 0x0。

3130	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留								SPI3 EN	FLS EN	SRAMEN	DMA2 EN	SPI2EN	TIM7EN	TIM14 EN	UART1EN
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKEN	PWMEN	UART2EN	DMEN	SPI1EN	I2C1EN	UART0EN	TIM1EN		ADC0EN	TIM6EN					WWDGEN
rw		rw			rw		rw				rw				

位 24~31	保留, 始终读为 0。
位 23	SPI3EN:SPI3 时钟使能 (SPI3 clock enable) 由软件置'1'或清'0' 0: SPI3 时钟关闭; 1: SPI3 时钟开启。
位 22	FLSEN:FLS 时钟使能 (FLS clock enable) 由软件置'1'或清'0' 0: FLS 时钟关闭; 1: FLS 时钟开启。
位 21	SRAMEN:SRAM 时钟使能 (SRAM clock enable) 由软件置'1'或清'0' 0: SRAM 时钟关闭; 1: SRAM 时钟开启。
位 20	DMA2EN:DMA2 时钟使能 (DMA2 clock enable) 由软件置'1'或清'0' 0: DMA2 时钟关闭; 1: DMA2 时钟开启。
位 19	SPI2EN:SPI2 时钟使能 (SPI2 clock enable) 由软件置'1'或清'0' 0: SPI2 时钟关闭; 1: SPI2 时钟开启。
位 18	TIM7EN: 定时器 7 时钟使能 (Timer 7 clock enable) 由软件置'1'或清'0' 0: 定时器 7 时钟关闭; 1: 定时器 7 时钟开启。
位 17	TIM14EN: 定时器 14 时钟使能 (Timer 14 clock enable) 由软件置'1'或清'0' 0: 定时器 14 时钟关闭; 1: 定时器 14 时钟开启。

位 16	<p>UART1EN: UART1 时钟使能 (UART 1 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: UART1 时钟关闭;</p> <p>1: UART1 时钟开启。</p>
位 15	<p>TKEN: TK 时钟使能 (TKclock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: TK 时钟关闭;</p> <p>1: TK 时钟开启。</p>
位 14	<p>PWMEN: PWM 时钟使能 (PWM clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: PWM 时钟关闭;</p> <p>1: PWM 时钟开启。</p>
位 13	<p>UART2EN: UART2 时钟使能 (UART 2 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: UART2 时钟关闭;</p> <p>1: UART2 时钟开启。</p>
位 12	<p>DMA1EN: DMA1 时钟使能 (DMA1clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: DMA1 时钟关闭;</p> <p>1: DMA1 时钟开启。</p>
位 11	<p>SPI1EN: SPI1 时钟使能 (SPI1 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: SPI1 时钟关闭;</p> <p>1: SPI1 时钟开启。</p>
位 10	<p>I2C1EN: I2C 1 时钟使能 (I2C 1 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: I2C 1 时钟关闭;</p> <p>1: I2C 1 时钟开启。</p>
位 9	<p>UART0EN: UART0 时钟使能 (UART0 1 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: UART0 时钟关闭;</p> <p>1: UART0 时钟开启。</p>
位 8	<p>TIM1EN: 定时器 1 时钟使能 (Timer 1 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: 定时器 1 时钟关闭;</p> <p>1: 定时器 1 时钟开启。</p>
位 7	保留, 始终读为 0。
位 6	<p>ADC0EN: ADC0 时钟使能 (ADC0 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: ADC0 时钟关闭;</p> <p>1: ADC0 时钟开启。</p>
位 5	<p>TIM6EN: 定时器 6 时钟使能 (Timer 6 clock enable)</p> <p>由软件置'1'或清'0'</p> <p>0: 定时器 6 时钟关闭;</p>

位 1:4	1: 定时器 6 时钟开启。 保留, 始终读为 0。
位 0	WWDGEN: 窗口看门狗时钟使能 (Window watchdog clock enable) 由软件置 '1' 或清 '0' 0: 窗口看门狗时钟关闭; 1: 窗口看门狗时钟开启。

9.2.4 控制/状态寄存器 (RCC_CSR)

偏移地址: 0x24

复位值: 0x0C00 0000, 除复位标志外由系统复位清除, 复位标志只能由电源复位清除。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR	WWDG	IWDG	SFT	POR	PIN	保留	RMVF	V18	保留						
RSTF	RSTF	RSTF	RSTF	RSTF	RSTF			pwrrstf							
rw	rw	rw	rw	rw	rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						INTMAP	LPDS	保留	DBP	RTCS	RTCE	LS	LSI	LSIO	
										EL	N	EO	RDY	N	
													r		rw

位 31	低功耗复位标志, 在低功耗管理复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0: 无低功耗管理复位发生; 1: 发生低功耗管理复位
位 30	窗口看门狗复位标志, 在窗口看门狗复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0: 无窗口看门狗复位发生; 1: 发生窗口看门狗复位
位 29	独立看门狗复位标志, 在独立看门狗复位发生在 VDD 区域时由硬件置'1'; 由软件通过写 RMVF 位清除。 0: 无独立看门狗复位发生; 1: 发生独立看门狗复位
位 28	软件复位标志, 在软件复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0: 无软件复位发生; 1: 发生软件复位
位 27	上电/掉电复位标志, 在上电/掉电复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0: 无上电/掉电复位发生; 1: 发生上电/掉电复位

位 26	NRST 管脚复位标志，在 NRST 管脚复位发生时由硬件置'1'；由软件通过写 RMVF 位清除。 0：无 NRST 管脚复位发生； 1：发生 NRST 管脚复位。
位 25	保留
位 24	清除复位标志，由软件置'1'来清除复位标志。 0：无作用； 1：清除复位标志
位 23	由 PORRSTF 驱动，由软件通过写 RMVF 位清除 PORRSTF 来清除
位 10~9	该位用于设置服务程序是在 RAM 或 FLASH、ROM 中执行，用于 ROM 中断服务程序地址跳转使用
位 8	LPDS：深睡眠下的低功耗 PDDS=0 时，与 PDDS 位协同操作 0：在停机模式下电压调压器开启 1：在停机模式下电压调压器处于低功耗模式
位 6~7	保留。
位 5	DBP：取消后备区域的写保护 在复位后，RTC 和后备寄存器处于被保护状态以防意外写入。设置这位允许写入这些寄存器。 0：禁止写入 RTC 和后备寄存器
位 4	RTCSEL：RTC 时钟源选择 由软件设置来选择 RTC 时钟源。可通过设置 BDRST 位来清除。 1：LSE 振荡器作为 RTC 时钟； 0：LSI 振荡器作为 RTC 时钟；
位 3	RTCEN：RTC 时钟使能由软件置'1'或清'0' 0：RTC 时钟关闭； 1：RTC 时钟开启。
位 2	LSEON：外部低速振荡器使能由软件置'1'或清'0' 0：外部 32768Hz 振荡器关闭； 1：外部 32768Hz 振荡器开启。
位 1	LSIRDY：内部低速振荡器就绪 (Internal low-speed oscillator ready) 由硬件置'1'或清'0'来指示内部 96kHz RC 振荡器是否就绪。 0：内部 96kHz RC 振荡器时钟未就绪； 1：内部 96kHz RC 振荡器时钟就绪。
位 0	LSION：内部低速振荡器使能 (Internal low-speed oscillator enable) 由软件置'1'或清'0'。 0：内部 96kHz RC 振荡器关闭； 1：内部 96kHz RC 振荡器开启。

10 供电和复位系统

10.1 供电系统

在 CA1020 系列芯片 VDD 和 VSS 引脚间接入 1.8V - 5.5V 的电源，此电源可直接给芯片内部数字及模拟系统供电。需要注意的是，不同的供电电压条件下，芯片支持运行的最高频率和功耗并不相同，具体请查看电气特性章节。

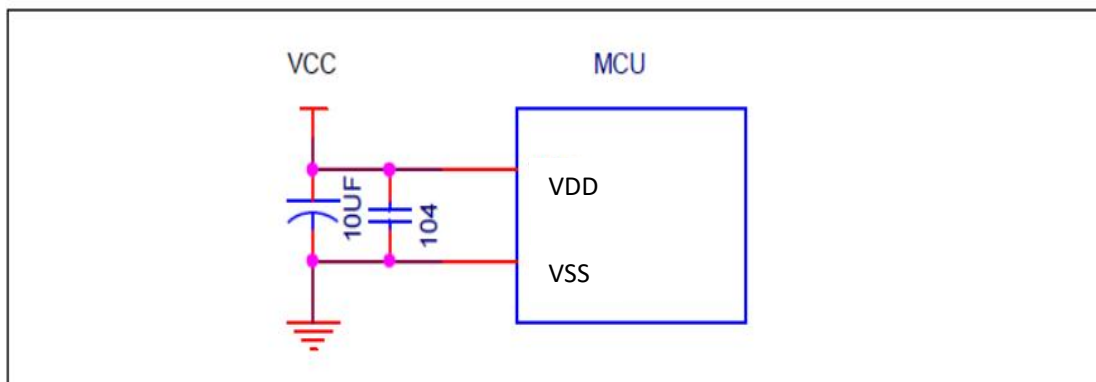


图 10-1-1 芯片供电典型电路图

- 重要提醒：**
1. 以上电路中，滤波电容 10uF 和 104 为芯片供电电路标配，不可省略，此电容须靠近芯片电源引脚摆放，否则有可能会导导致芯片工作异常。
 2. 以上电路及元件参数仅供参考，根据外围工作环境及不同电压供电参数可能需要修改。

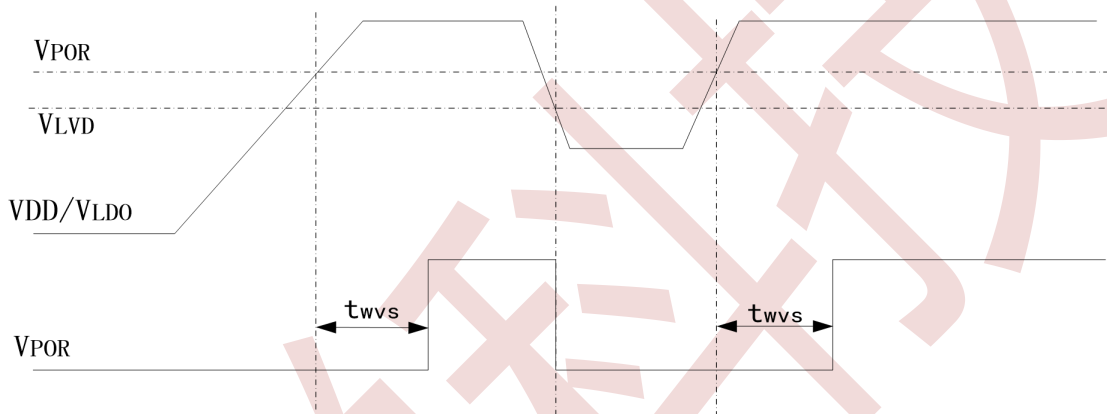
10.2 复位系统

CA1020 系列芯片有多个内部和外部复位源。

- 上电复位 (POR)

系统上电呈现逐渐上升的曲线形式，需要一定时间才能达到正常的工作电压。上电复位是基于电源电压 VDD，当电压低于检测阈值时，上电复位信号有效。

上电复位电路能够保证芯片在上电过程中处于复位状态，芯片上电后能够从一个已知的稳定的状态开始运行。上电复位信号也会被芯片内部的计数器展宽，以保证上电后内部的各种模拟模块能够进入稳定的工作状态。



twvs: 等待电压稳定时间

图 10-2-2 上电复位电路示例及上电过程

10.2.1 掉电复位 (BOR)

利用掉电复位，可以为芯片提供电源跌落（例如受到干扰或者负载变化）的预警信号。一旦发现电源电压 VDD 下降到某一个阈值时，就使芯片及时复位以免系统工作状态不正常或者程序执行错误。

10.2.2 低电压复位

低电压检测 (LVD) 可以在多种工作模式下持续监控电源电压 VDD。当 VDD 低于 LVD 设定的域值电压超过 20us 就可以产生复位信号（前提是 LVD 设置为复位模式）。

10.2.3 看门狗复位

看门狗定时器负责监控处理器执行指令的情况，通过合适的配置，如果看门狗定时器在特定时间段内未被刷新，则可以产生复位信号。上电复位后，看门狗定时器是关闭的，用户需要时，再配置开启。

10.2.4 软复位

芯片可以在程序控制下执行软复位。

11 DMA 控制器

11.1 简介

直接存储器存取（DMA）用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据可以通过 DMA 快速地移动，这就节省了 CPU 的资源来做其他操作。

DMA 控制器有 5 个逻辑通道。每个逻辑通道用于服务来自单个或多个外设的内存访问请求。内部仲裁器控制不同 DMA 通道的优先级。

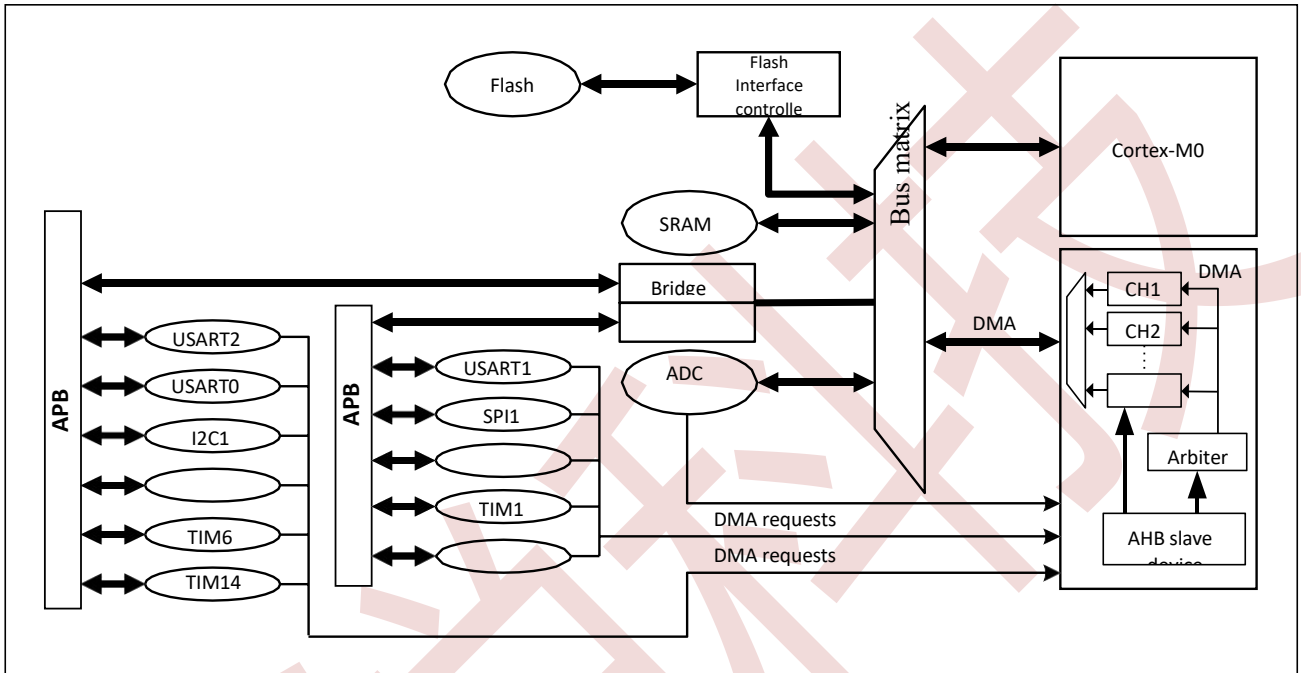
11.2 主要特性

DMA 主要特性：

- 两个 DMA 控制器，每个 DMA 控制有两个通道。
- 每个 DMA 通道支持硬件请求和软件触发来启动传输，并由软件配置。
- 每个 DMA 通道都有专用的软件优先级（共有四级：很高、高、中和底），可以单独配置。具有相同软件优先级的通道将进一步比较硬件索引（通道号）以确定最终优先级（索引号越低的通道优先级越高）。
- 独立数据源和目标数据区的传输宽度（字节、半字、全字），模拟打包和拆包的过程。源和目标地址必须按数据传输宽度对齐。
- 每个通道可配置循环传输模式。
- 每个通道有 3 个独立的事件标志和中断（传输完成、半传输、传输错误）和 1 个全局中断标志（由 3 个事件的逻辑或设置）。
- 支持内存到内存、内存到外设和外设到内存三种传输类型。
- Flash、SRAM、AHB、APB1 和 APB2 外设均可作为访问的源和目标。
- 可配置数据传输数（0~65535）。

11.3 功能框图

图7-1 DMA 框图



11.4 功能描述

DMA 控制器和 Cortex™-M0 内核共享相同的系统数据总线。当 CPU 和 DMA 同时访问同一个目标 (RAM 或外设) 时，DMA 请求会暂停 CPU 访问系统总线几个周期，由总线仲裁器进行循环调度。这允许 CPU 获得至少一半的系统总线（内存或外围设备）带宽。

11.4.1 DMA 操作

DMA 请求可以由硬件外设或软件触发，DMA 控制器根据通道的优先级处理请求。根据配置的传输地址和位宽从源地址读取数据，然后将读取的数据存储在目的地址空间中。一次操作后，控制器计算剩余传输次数，并更新下一次传输的源地址和目的地址。

每个 DMA 数据传输包括三个操作：

- 数据访问：根据传输方向确定源地址 (DMA_CPARx 或 DMA_CMARx)，从源地址读取数据。
- 数据存储：根据传输方向确定目的地址 (DMA_CPARx 或 DMA_CMARx)，将读取的数据存储到目的地址空间。
- 计算未完成操作的数量，对 DMA_CNDTRx 寄存器进行减量操作，更新下一个操作的源地址和目的地址。

11.4.2 通道优先级和仲裁器

仲裁器根据通道请求的优先级来启动外设/存储器的访问。

优先权管理分 2 个阶段：

- 软件：每个通道的优先权可以在 DMA_CCRx 寄存器中设置，有 4 个等级：
 - ◆ 最高优先级
 - ◆ 高优先级
 - ◆ 中等优先级
 - ◆ 低优先级
- 硬件：如果 2 个请求有相同的软件优先级，则较低编号的通道比较高编号的通道有较高的优先权。举个例子，通道 2 优先于通道 4。

11.4.3 DMA 通道和传输数量

每个通道都可以在指定地址的外设寄存器和内存地址之间进行 DMA 传输。DMA 传输的数据数量是可编程的，最大支持值为 65535。DMA_CNDTRx 寄存器在每次传输后递减。

11.4.4 可编程的数据位宽

外设和内存传输数据位宽支持字节、半字和字，可以通过 DMA_CCRx.PSIZE 和 DMA_CCRx.MSIZE 进行编程。

当 DMA_CCRx.PSIZE 和 DMA_CCRx.MSIZE 不同时，DMA 模块根据表 7-1 对齐数据。

表 7-1 可编程的数据宽度和大小端操作（当 PINC = MINC = 1）

Source width (bit)	Destination width (bit)	Number of transfer (bit)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
8	8	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W B0 [7:0] @0x0 2: R B1 [7:0] @0x1, W B1 [7:0] @0x1 3: R B2 [7:0] @0x2, W B2 [7:0] @0x2 4: R B3 [7:0] @0x3, W B3 [7:0] @0x3	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3
8	16	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 00B0 [15:0] @0x0 2: R B1 [7:0] @0x1, W 00B1 [15:0] @0x2 3: R B2 [7:0] @0x2, W 00B2 [15:0] @0x4 4: R B3 [7:0] @0x3, W 00B3 [15:0] @0x6	0x0 / 00B0 0x2 / 00B1 0x4 / 00B2 0x6 / 00B3
8	32	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 000000B0 [31:0] @0x0 2: R B1 [7:0] @0x1, W 000000B1 [31:0] @0x4 3: R B2 [7:0] @0x2, W 000000B2 [31:0] @0x8 4: R B3 [7:0] @0x3, W 000000B3 [31:0] @0xC	0x0 / 000000B0 0x4 / 000000B1 0x8 / 000000B2 0xC / 000000B3
16	8	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B0 [7:0] @0x0 2: R B3B2 [15:0] @0x2, W B2 [7:0] @0x1 3: R B5B4 [15:0] @0x4, W B4 [7:0] @0x2 4: R B7B6 [15:0] @0x6, W B6 [7:0] @0x3	0x0 / B0 0x1 / B2 0x2 / B4 0x3 / B6

Source width (bit)	Destination width (bit)	Number of transfer (bit)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
16	16	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B1B0 [15:0] @0x0 2: R B3B2 [15:0] @0x2, W B3B2 [15:0] @0x2 3: R B5B4 [15:0] @0x4, W B5B4 [15:0] @0x4 4: R B7B6 [15:0] @0x6, W B7B6 [15:0] @0x6	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6
16	32	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W 0000B1B0 [31:0] @0x0 2: R B3B2 [15:0] @0x2, W 0000B3B2 [31:0] @0x4 3: R B5B4 [15:0] @0x4, W 0000B5B4 [31:0] @0x8 4: R B7B6 [15:0] @0x6, W 0000B7B6 [31:0] @0xC	0x0 / 0000B1B0 0x4 / 0000B3B2 0x8 / 0000B5B4 0xC / 0000B7B6
32	8	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B0 [7:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B4 [7:0] @0x1 3: R BBBAB9B8 [31:0] @0x8, W B8 [7:0] @0x2 4: R BFBEBDBC [31:0] @0xC, W BC [7:0] @0x3	0x0 / B0 0x1 / B4 0x2 / B8 0x3 / BC
32	16	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B1B0 [15:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B5B4 [15:0] @0x2 3: R BBBAB9B8 [31:0] @0x8, W B9B8 [15:0] @0x4 4: R BFBEBDBC [31:0] @0xC, W BDBC [15:0] @0x6	0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC
32	32	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B3B2B1B0 [31:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B7B6B5B4 [31:0] @0x4 3: R BBBAB9B8 [31:0] @0x8, W BBBAB9B8 [31:0] @0x8 4: R BFBEBDBC [31:0] @0xC, W BFBEBDBC [31:0] @0xC	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC

注意:

DMA 总是向 HWDATA[31:0]提供完整的 32 位数据, 无论它是什么目标大小 (HSIZE 仍然遵循设备支持字节/半字操作的目标大小设置)。它提供的 HWDATA[31:0]遵循以下规则:

- 当源大小小于目标大小时, DMA 用0 填充 MSB, 直到它们的大小匹配并将其复制为 32 位。例如, 源是8 位数据 0x55, 目标大小是 16 位。DMA 用0 填充源数据使其成为 16 位0x0055, 然后将其复制为 32 位数据 0x0055_0055 并提供给 HWDATA[31:0]; (如果目标大小为 32 位, 则 DMA 只会用 0 填充源数据)。
- 当源大小大于或等于目标大小且小于 32 位时, DMA 将源数据复制到 32 位数据。例如, 源数据为 8 位数据 0x1F, HWDATA[31:0] = 0x1F1F_1F1F。如果源数据是 16 位数据 0x2345, 则 HWDATA[31:0] = 0x2345_2345。

这保证了仅支持字操作的外设不会产生总线错误, 并且所需的数据仍然可以通过额外的位 (即 0 填充) 移动到我们想要的位置。如果用户想要配置一个 8 位寄存器但与 32 位地址边界对齐, 则源大小应设置为 8 位, 目标大小应设置为 32 位, 因此额外的位将用 0 填充。

11.4.5 外设/内存地址递增

DMA_CCRx.PINC 和 DMA_CCRx.MINC 分别控制外设地址和内存地址是否使能自动递增模式。软件在传输过程中不能写（可以读）地址寄存器。

- 在自动递增模式下，下一个要传输的地址在每次传输后根据数据位宽（1、2 或 4）自动增加。第一次传输的地址存储在 DMA_CPARx 或 DMA_CMARx 寄存器中。
- 在固定模式下，地址始终固定为初始地址。

在传输结束时（即传输计数变为 0），将根据当前是否工作于循环模式进行不同的处理。

- 在非循环模式下，DMA 在传输完成后停止。要开始新的 DMA 传输，需要在禁用 DMA 通道的情况下重写 DMA_CNDTRx 寄存器中的传输数量。
- 在循环模式下，在传输结束时，DMA_CNDTRx 寄存器的内容会自动重新加载其初始值，并且当前内部外设或内存地址也会重新加载 DMA_CPARx 或 DMA_CMARx 寄存器设置的初始基地址。

11.4.6 通道配置流程

详细配置流程如下：

1. 配置中断屏蔽位，1：启用中断，0：禁用中断。
2. 配置通道外设地址和内存地址以及传输方向。
3. 配置通道优先级，0：最低，3：最高。
4. 配置外设和内存地址增量。
5. 配置通道传输块大小。
6. 如有必要，配置循环模式。
7. 如果是存储器到存储器，配置 MEM2MEM 模式（注：要配置 DMA 工作在 M2M 模式，用户需要将相应的通道选择值设置为保留值）。
8. 在通道 1~8 上重复第 1~8 步。
9. 最后使能相应通道。

如果使用软件提供中断服务，则软件必须查询中断状态寄存器以检查发生了哪个中断（软件需要向中断标志清除位写 1 来清除相应的中断）。在使用通道之前，应清除该通道对应的所有中断。

如果中断是传输完成中断，软件可以配置下一次传输，或者向用户报告该通道传输完成。

11.4.7 循环模式

循环模式用于处理循环缓冲区和连续的数据传输（如 ADC 的扫描模式）。在 DMA_CCRx 寄存器中的 CIRC 位用于开启这一功能。当启动了循环模式，数据传输的数目变为 0 时，将会自动地被恢复成配置通道时设置的初值，DMA 操作将会继续进行。

11.4.8 错误管理

读写一个保留的地址区域，将会产生 DMA 传输错误。当在 DMA 读写操作时发生 DMA 传输错误时，硬件会自动地清除发生错误的通道所对应的通道配置寄存器（DMA_CCRx）的 EN 位，该通道操作被停止。此时，在 DMA_IFCR 寄存器中对应该通道的传输错误中断标志位（TEIF）将被置位，如果在 DMA_CCRx 寄存器中设置了传输错误中断允许位，则将产生中断。

11.4.9 中断

- 传输完成中断：

通道数据传输完成时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

- 半传输中断：

当传输了一半的通道数据时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

- 传输错误中断：

总线返回错误时产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

表7-2 DMA 中断请求

中断事件	事件标志位	使能控制位
传输过半	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

11.4.10 DMA 请求映射

从外设（TIMx[x=1、3、6、14]、ADC0、ADC1、SPI1、I2C1 和 USARTx[x=0、1、2]）产生的 5 个请求，通过逻辑或输入到 DMA1 控制器，这意味着同时只能有一个请求有效。参见下图的 DMA1 请求映象。

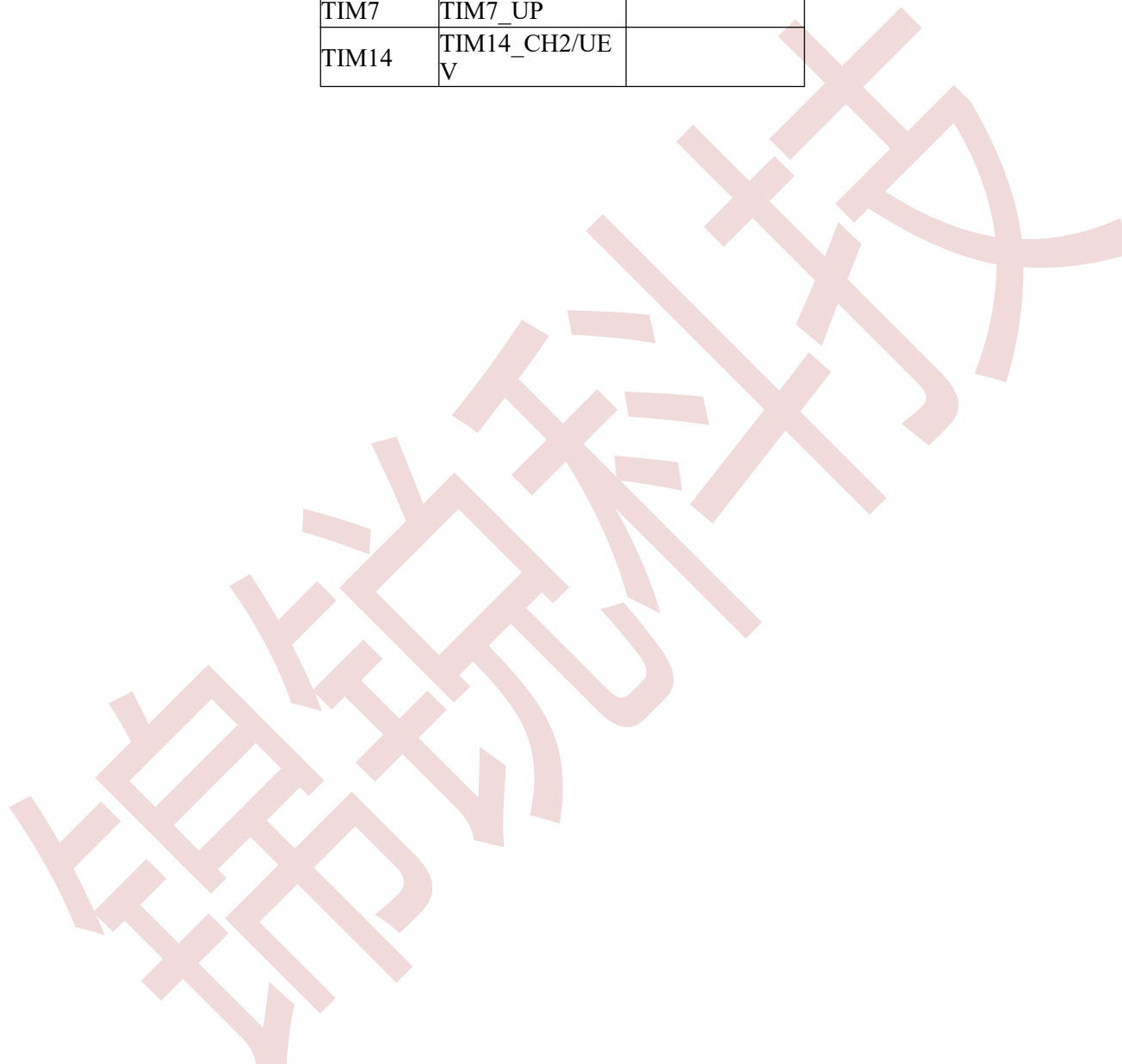
外设的 DMA 请求，可以通过设置相应外设寄存器中的控制位，被独立地开启或关闭。

表7-3 DMA1 请求映射

外设	通道 1	通道 2
ADC	ADC0	
SPI	SPI1_TX	SPI1_RX
SPI3	SPI3_TX	SPI3_RX
USART1	USART1_TX	USART1_RX
TIM1		TIM1_CH1/CH2
TIM6	TIM6_UP	
TIM14	TIM14_CH1	

表7-4 DMA2 请求映射

外设	通道 1	通道 2
SPI2	SPI2_TX	SPI2_RX
USART0	USART0_TX	USART0_RX
USART2	USART2_TX	USART2_RX
TIM1	TIM1_UEV	TIM1_CH3/CH4
TIM7	TIM7_UP	
TIM14	TIM14_CH2/UEV	



11.5 DMA 寄存器

11.5.1 DMA 寄存器概览

表 7-4 DMA 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	DMA_ISR	DMA 中断状态寄存器	0x00000000
0x04	DMA_IFCR	DMA 中断标志清除寄存器	0x00000000
0x08+20×(n-1)	DMA_CCRx	DMA 通道 x 配置寄存器	0x00000000
0x0C+20×(n-1)	DMA_CNDTRx	DMA 通道 x 传输数量寄存器	0x00000000
0x10+20×(n-1)	DMA_CPARx	DMA 通道 x 外设地址寄存器	0x00000000
0x14+20×(n-1)	DMA_CMARx	DMA 通道 x 存储器地址寄存器	0x00000000

11.5.2 DMA 中断状态寄存器 (DMA_ISR)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
r r r r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r r r r r r r r								r	r	r	r	r	r	r	r

位 31:8	保留, 始终读为 0。
位 7, 3	TEIFx: 通道 x 的传输错误标志 (x = 1, 2) (Channel x transfer error flag) 硬件设置这些位。在 DMA_IFCR 寄存器的相应位写入 '1' 可以清除这里对应的标志位。 0: 在通道 x 没有传输错误 (TE); 1: 在通道 x 发生了传输错误 (TE)。
位 6, 2	HTIFx: 通道 x 的半传输标志 (x = 1, 2) (Channel x half transfer flag) 硬件设置这些位。在 DMA_IFCR 寄存器的相应位写入 '1' 可以清除这里对应的标志位。 0: 在通道 x 没有半传输事件 (HT); 1: 在通道 x 产生了半传输事件 (HT)。
位 5, 1	TCIFx: 通道 x 的传输完成标志 (x = 1, 2) (Channel x transfer complete flag) 硬件设置这些位。在 DMA_IFCR 寄存器的相应位写入 '1' 可以清除这里对应的标志位。 0: 在通道 x 没有传输完成事件 (TC); 1: 在通道 x 产生了传输完成事件 (TC)。
位 4, 0	GIFx: 通道 x 的全局中断标志 (x = 1, 2) (Channel x global interrupt flag) 硬件设置这些位。在 DMA_IFCR 寄存器的相应位写入 '1' 可以清除这里对应的标志位。 0: 在通道 x 没有 TE、HT 或 TC 事件; 1: 在通道 x 产生了 TE、HT 或 TC 事件。

11.5.3 DMA 中断标志清除寄存器 (DMA_IFCR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CTEIF ₂	CHTIF ₂	CTCIF ₂	CGIF ₂	CTEIF ₁	CHTIF ₁	CTCIF ₁	CGIF ₁
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:8	保留, 始终读为 0。
位 7, 3	CTEIF _x : 清除通道 x 的传输错误标志 (x = 1, 2) (Channel x transfer error clear) 这些位由软件设置和清除。 0: 不起作用 1: 清除 DMA_ISR 寄存器中的对应 TEIF 标志。
位 6, 2	CHTIF _x : 清除通道 x 的半传输标志 (x = 1, 2) (Channel x half transfer clear) 这些位由软件设置和清除。 0: 不起作用 1: 清除 DMA_ISR 寄存器中的对应 HTIF 标志。
位 5, 1	CTCIF _x : 清除通道 x 的传输完成标志 (x = 1, 2) (Channel x transfer complete clear) 这些位由软件设置和清除。 0: 不起作用 1: 清除 DMA_ISR 寄存器中的对应 TCIF 标志。
位 4, 0	CGIF _x : 清除通道 x 的全局中断标志 (x = 1, 2) (Channel x global interrupt clear) 这些位由软件设置和清除。 0: 不起作用 1: 清除 DMA_ISR 寄存器中的对应的 GIF、TEIF、HTIF 和 TCIF 标志。

11.5.4 DMA 通道 x 配置寄存器 (DMA_CCRx)(x = 1, 2)

偏移地址: 0x08 + 20 x (通道编号 - 1)

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MEM2MEM	PL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

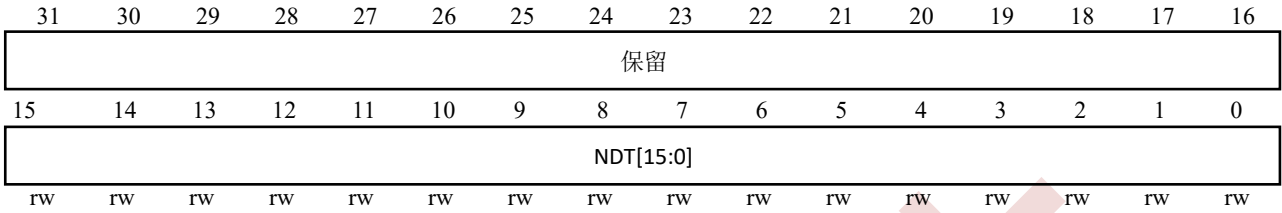
位 31:15	保留, 始终读为 0。
位 14	MEM2MEM: 存储器到存储器模式 (Memory to memory mode) 该位由软件设置和清除。 0: 非存储器到存储器模式; 1: 启动存储器到存储器模式。

位 13:12	<p>PL[1:0]: 通道优先级 (Channel priority level) 这些位由软件设置和清除。 00: 低 01: 中 10: 高 11: 最高</p>
位 11:10	<p>MSIZE[1:0]: 存储器数据宽度 (Memory size) 这些位由软件设置和清除。 00:8 位 01:16 位 10:32 位 11: 保留</p>
位 9:8	<p>PSIZE[1:0]: 外设数据宽度 (Peripheral size) 这些位由软件设置和清除。 00:8 位 01:16 位 10:32 位 11: 保留</p>
位 7	<p>MINC: 存储器地址增量模式 (Memory increment mode)该 位由软件设置和清除。 0: 不执行存储器地址增量操作 1: 执行存储器地址增量操作</p>
位 6	<p>PINC: 外设地址增量模式 (Peripheral increment mode) 该位由软件设置和清除。 0: 不执行外设地址增量操作 1: 执行外设地址增量操作</p>
位 5	<p>CIRC: 循环模式 (Circular mode)该 位由软件设置和清除。 0: 不执行循环操作 1: 执行循环操作</p>
位 4	<p>DIR: 数据传输方向 (Data transfer direction) 该位由软件设置和清除。 0: 从外设读 1: 从存储器读</p>
位 3	<p>TEIE: 允许传输错误中断 (Transfer error interrupt enable) 该位由软件设置和清除。 0: 禁止 TE 中断 1: 允许 TE 中断</p>
位 2	<p>HTIE: 允许半传输中断 (Half transfer interrupt enable) 该位由软件设置和清除。 0: 禁止 HT 中断 1: 允许 HT 中断</p>
位 1	<p>TCIE: 允许传输完成中断 (Transfer complete interrupt enable) 该位由软件设置和清除。 0: 禁止 TC 中断 1: 允许 TC 中断</p>
位 0	<p>EN: 通道开启 (Channel enable)该 位由软件设置和清除。 0: 通道不工作 1: 通道开启</p>

11.5.5 DMA 通道 x 传输数量寄存器 (DMA_CNDTRx)(x = 1, 2)

偏移地址: 0x0C + 20 x (通道编号 - 1)

复位值: 0x0000 0000



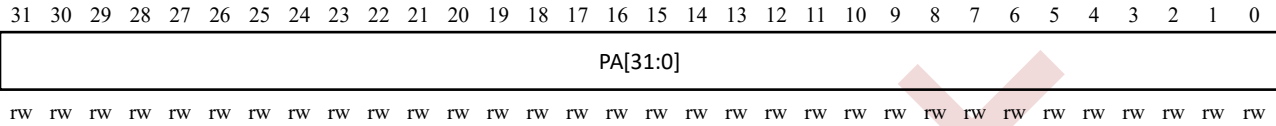
位 31:16	保留, 始终读为 0。
位 15:0	<p>NDT[15:0]: 数据传输数量 (Number of data to transfer)</p> <p>数据传输数量为 0 至 65535。这个寄存器只能在通道不工作 (DMA_CCRx 的 EN=0) 时写入。通道开启后该寄存器变为只读, 指示剩余的待传输字节数目。寄存器内容在每次 DMA 传输后递减。</p> <p>数据传输结束后, 寄存器的内容或者变为 0; 或者当该通道配置为自动重新加载模式时, 寄存器的内容将被自动重新加载为之前配置时的数值。</p> <p>当寄存器的内容为 0 时, 无论通道是否开启, 都不会发生任何数据传输。</p>

11.5.6 DMA 通道 x 外设地址寄存器 (DMA_CPARx)(x = 1, 2)

偏移地址: 0x10 + 20 x (通道编号 - 1)

复位值: 0x0000 0000

当开启通道 (DMA_CCRx 的 EN=1) 时不能写该寄存器。



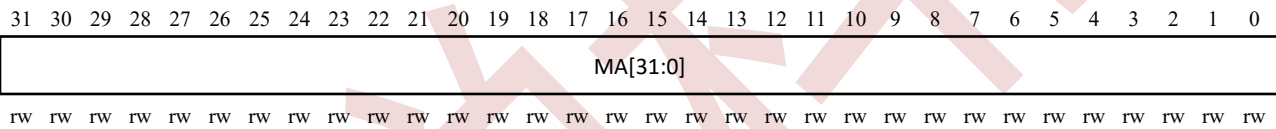
位 31:0	<p>PA[31:0]: 外设地址 (Peripheral address)</p> <p>外设数据寄存器的基地址, 作为数据传输的源或目标。</p> <p>当 PSIZE='01'(16 位), 不使用 PA[0]位。操作自动地与半字地址对齐。</p> <p>当 PSIZE='10'(32 位), 不使用 PA[1:0]位。操作自动地与字地址对齐。</p>
--------	--

11.5.7 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)(x = 1, 2)

偏移地址: 0x14 + 20 x (通道编号 - 1)

复位值: 0x0000 0000

当开启通道 (DMA_CCRx 的 EN=1) 时不能写该寄存器。



位 31:0	<p>MA[31:0]: 存储器地址</p> <p>存储器地址作为数据传输的源或目标。</p> <p>当 MSIZE='01'(16 位), 不使用 MA[0]位。操作自动地与半字地址对齐。</p> <p>当 MSIZE='10'(32 位), 不使用 MA[1:0]位。操作自动地与字地址对齐。</p>
--------	--

12 高级定时器 1 (TIM1)

12.1 TIM1 简介

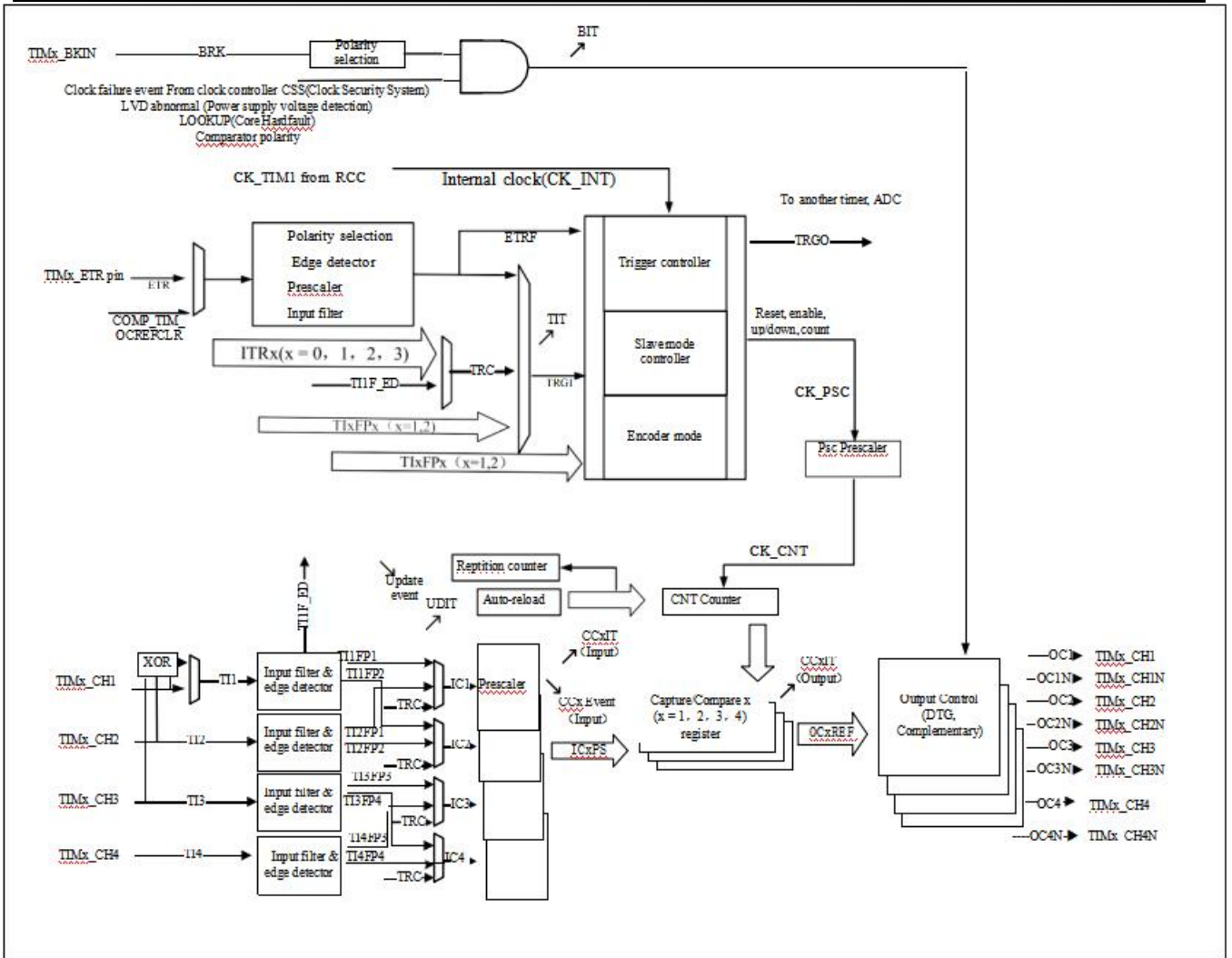
高级控制定时器(TIM1)由一个 16 位的自动装载计数器组成,它由一个可编程的预分频器驱动。它适合多种用途,包含测量输入信号的脉冲宽度(输入捕获),或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。使用定时器预分频器和 RCC 时钟控制预分频器,可以实现脉冲宽度和波形周期的调节。

高级控制定时器(TIM1)和通用定时器(TIMx)是完全独立的,它们不共享任何资源。它们可以同步操作。

12.2 TIM1 主要特性

- 16 位自动装载计数器。(可实现向上计数、向下计数、向上/下计数)
- 16 位可编程预分频器。(分频系数可配置为 1 到65536 之间的任意值)
- 可编程重复计数器
- 最多 4 个独立通道:
 - ◆ 输入捕获
 - ◆ 输出比较
 - ◆ PWM 生成(边缘或中央对齐模式)
 - 如下事件发生时产生中断/DMA:
 - ◆ 更新事件
 - ◆ 触发事件
 - ◆ 输入捕获
 - ◆ 输出比较
 - 死区时间可编程的互补输出

TIM1 框图



事件

中断和DMA 输出

捕获通道 1 输入可以来自 IOM 或比较器输出

12.3 TIM1 功能描述

12.3.1 时基单元

TIM1 的时基单元主要包括：计数器寄存器 (TIM1_CNT)、预分频器寄存器 (TIM1_PSC)、自动预装载寄存器 (TIM1_ARR) 和重复计数器寄存器 (TIM1_RCR)。

计数单元由一个 16 位的计数器和对应的自动预装载寄存器组成，可以实现递增计数，递减计数，递增和递减计数的功能。

计数器的时钟由预分频器提供，预分频器由预分频计数器和对应的寄存器组成，分频系数为 1-65536，可以随时写入，在下次更新事件时生效。

自动预装载寄存器有预装载功能的 16 位影子寄存器，通过设置 TIM1_CR1 寄存器的 ARPE 位选择写入 ARR 寄存器的值立即生效或发生更新事件时载入影子寄存器。

图 9-2 自动预装载

TIMx_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

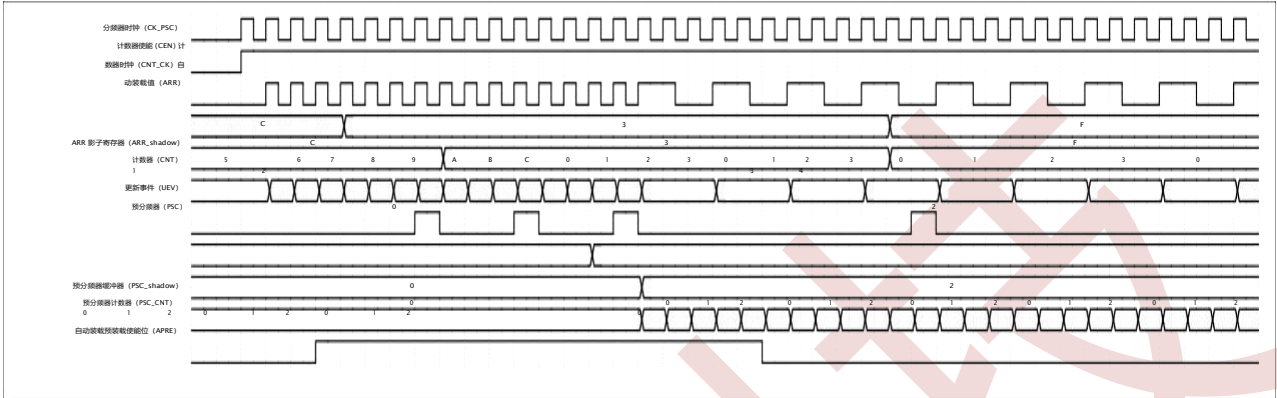
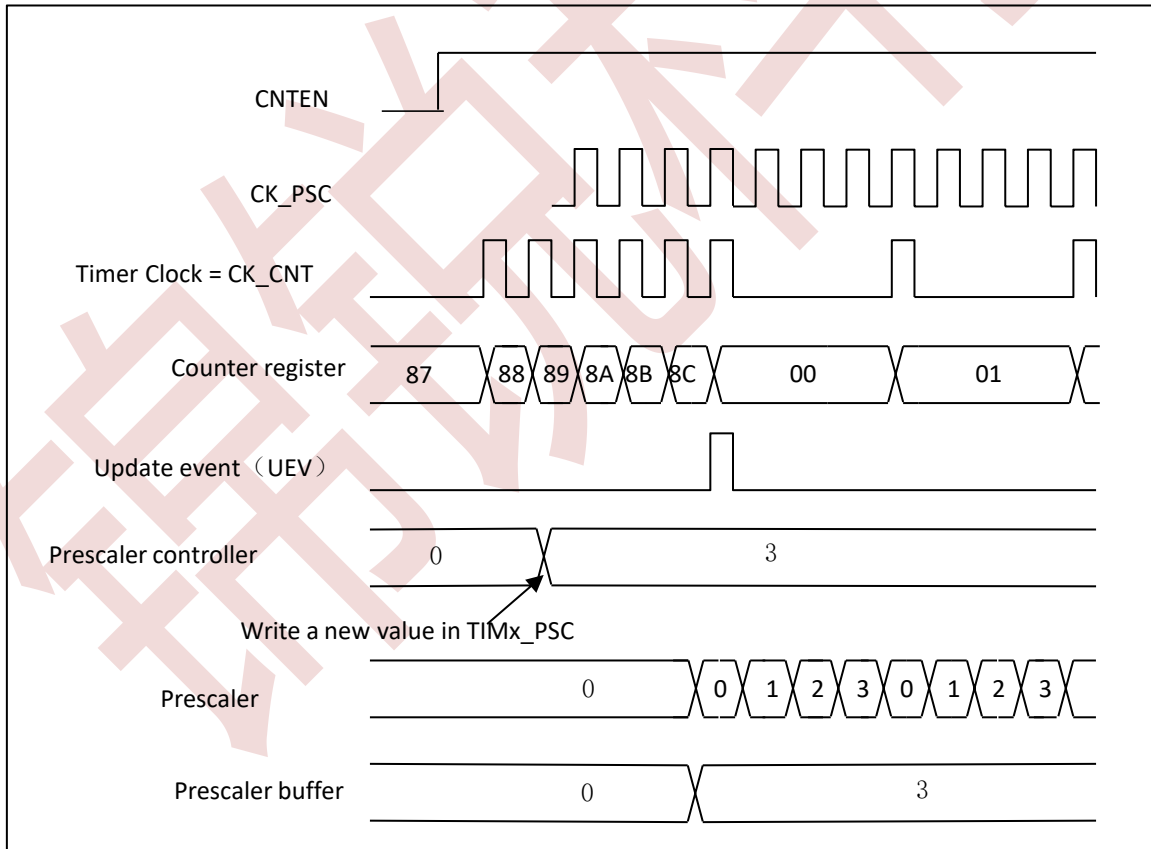


图9-3 当预分频的参数从 1 到4，计数器的时序图



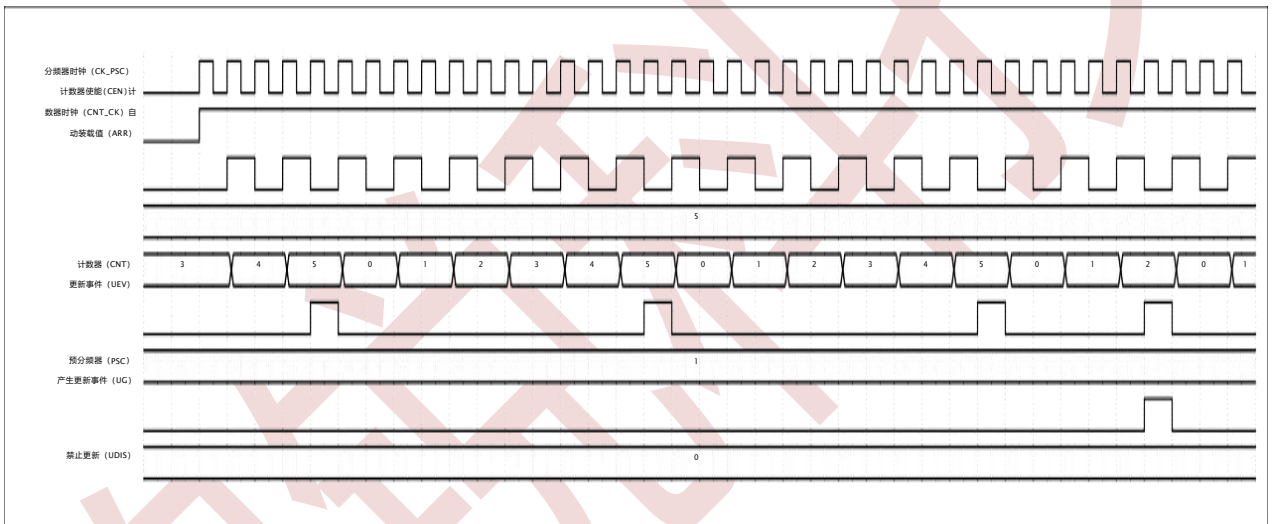
12.3.2 计数器模式

通过配置 TIM1_CR1 寄存器的 DIR 位和 CMS 位可以选择计数器的计数模式，可以分为三种计数模式，递增计数模式、递减计数模式和中央对齐计数模式（递增/递减计数模式），下面对每种计数模式做详细介绍。

向上计数模式

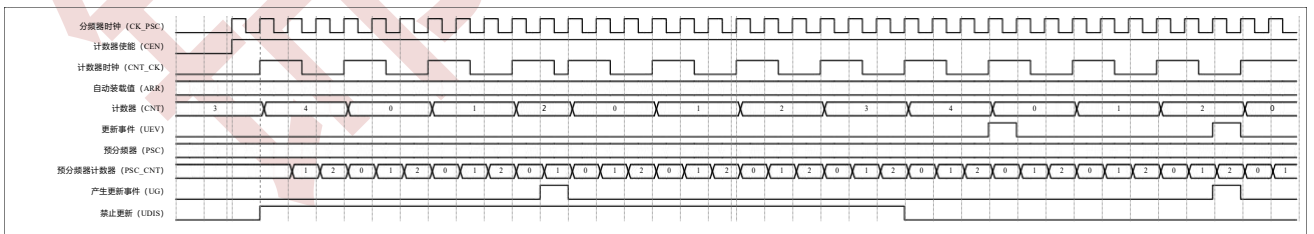
配置 TIM1_CR1 寄存器 CMS=0, DIR=0, 选择向上计数模式。

向上计数模式下，在使能 TIM1_CR1 寄存器的 CEN 后计数器由 0 开始向上计数，直至 TIM1_ARR 的值，产生一个计数器上溢事件（更新事件），并从 0 开始重新递增计数。当用户启用了重复计数功能，重复计数器在每次上溢事件时递减计数，只有当重复计数器从设定值递减到 0 时，才会产生更新事件。设置 IM1_EGR 寄存器的 UG=1, 同样可以产生一个更新事件。



向上计数模式 (UDIS=0)

通过配置 TIM1_CR1 寄存器的 UDIS=1, 可禁止产生更新事件，当计数器发生上溢事件时，不产生更新事件。此时若配置 UG=1, 不产生更新事件，但是计数器和预分频器计数器会被初始化，从零开始递增计数。



递增计数模式 (UDIS=1 禁止产生更新事件)

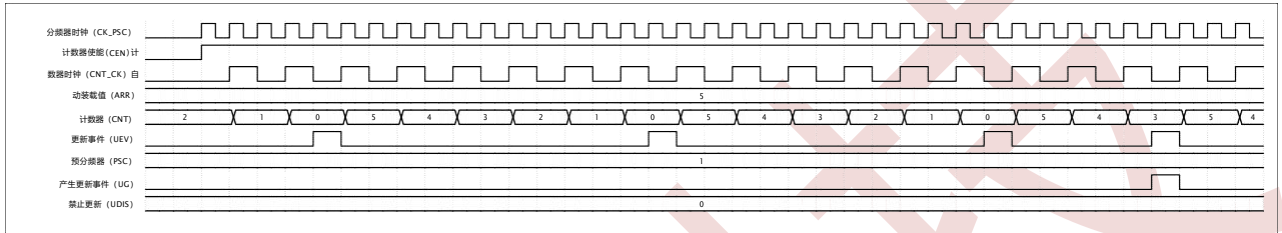
注：发生更新事件时

- 重复计数器被载入 RCR 寄存器中的值，并重新开始递减计数。
- ARR 寄存器中的值被载入 ARR 影子寄存器中。
- 预分频器的预装载值生效。

向下计数模式

配置 TIM1_CR1 寄存器的 CMS=0, DIR=1, 选择向下计数模式。

向下计数模式下, 计数器从自动预装载值 TIM1_ARR 开始向下计数, 计数到 0 时, 产生一个下溢事件 (更新事件)。当用户启用了重复计数功能后, 重复计数器在每次下溢事件时递减计数, 只有当重复计数器从设定值递减到 0 时, 才会产生更新事件; 设置 TIM1_EGR 寄存器的 UG=1, 同样可以产生一个更新事件, 更新事件后计数器从自动预装载值 TIM1_ARR 开始重新递减计数 (TIM1_CR1 寄存器 UDIS=0)。



向下计数模式 (UDIS=0)

通过配置 TIM1_CR1 寄存器的 UDIS=1, 可禁止产生更新事件, 当计数器发生下溢事件时, 不产生更新事件。此时若配置 UG=1, 同样不产生更新事件, 但是计数器和预分频器计数器会被初始化, 从 TIM1_ARR 开始计数。

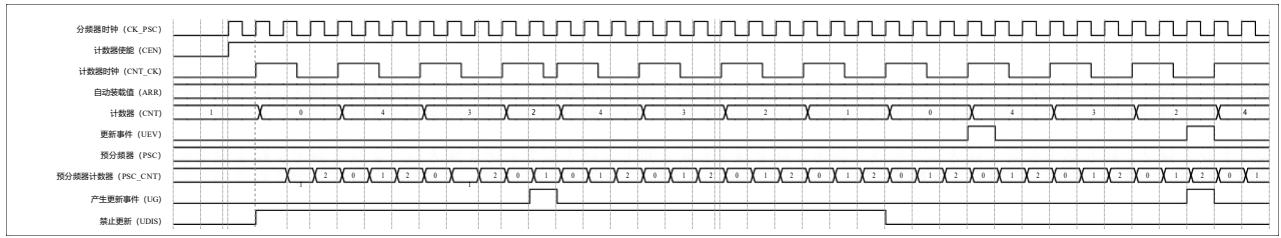


图 9-7 向下计数模式 (UDIS=1 禁止产生更新事件)

中央对齐模式

配置 TIM1_CR1 寄存器的 CMS ≠ 0 (此时写入 DIR 无效), 选择中央对齐计数模式。

中央对齐计数模式, 向上计数和向下计数交替进行。向上计数到 ARR-1 时, 产生一个上溢事件, 然后从 ARR 开始向下计数到 1, 产生一个下溢事件, 再从 0 开始向上计数。

当用户启用了重复计数功能后, 重复计数器在每次上溢事件或下溢事件时递减重复计数器值, 只有当重复计数器从设定值递减到 0 时, 才会产生更新事件; 设置 TIM1_EGR 寄存器的 UG=1, 同样可以产生一个更新事件, 更新事件后计数器从 0 开始重新递增计数 (TIM1_CR1 寄存器 UDIS=0)。

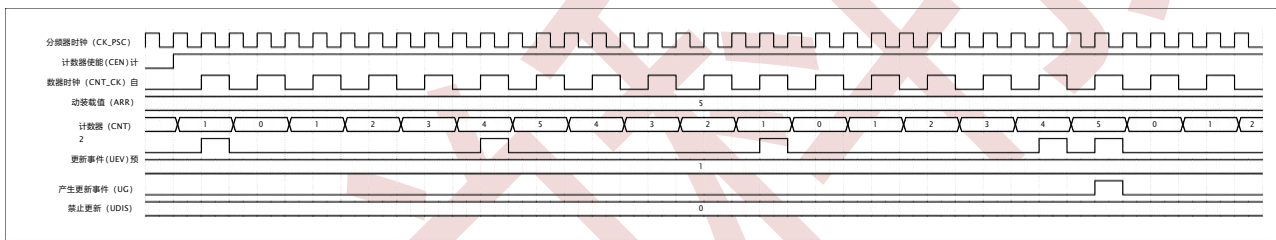
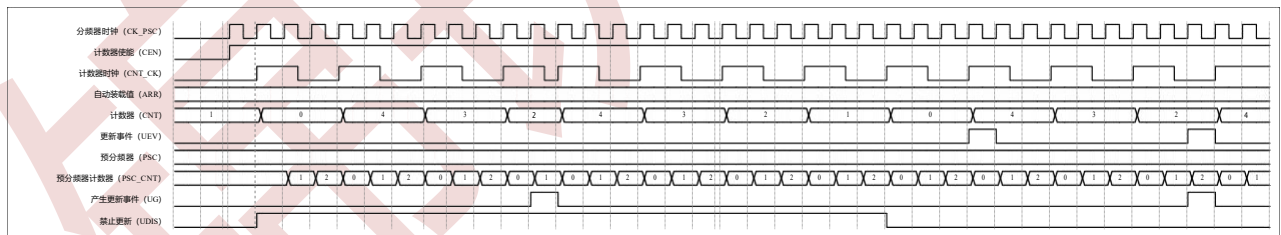


图 9-8 中央计数模式 (UDIS=0)

通过配置 TIM1_CR1 寄存器的 UDIS=1, 可禁止产生更新事件, 当计数器发生上溢或下溢事件时, 不产生更新事件。此时若配置 UG=1, 同样不产生更新事件, 但是计数器和预分频器计数器会被初始化, 从零开始重新计数。

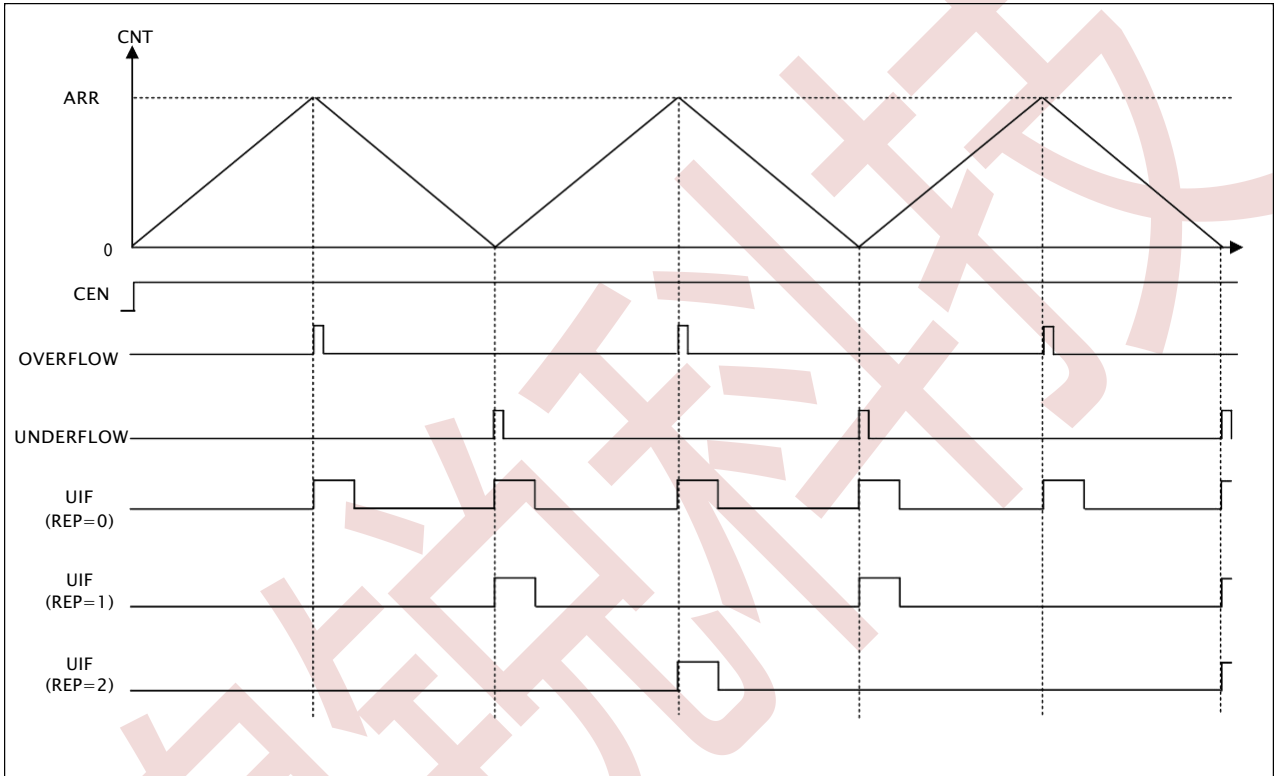


中央计数模式 (UDIS=1 禁止产生更新事件)

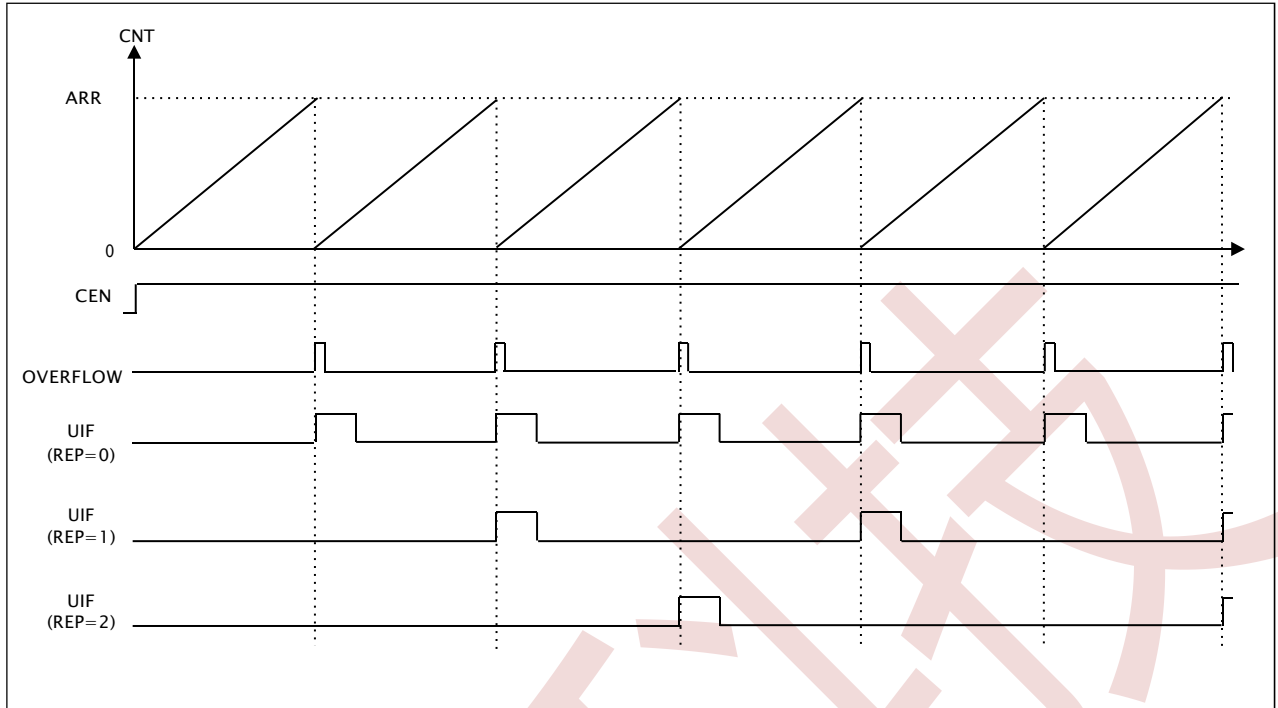
12.3.3 重复计数器

重复计数器可以用来调整更新事件产生的频率。边沿对齐模式下，向上计数时，重复计数器在计数器每次上溢时递减；向下计数时，重复计数器在计数器每次下溢时递减。中央对齐模式下，重复计数器在计数器上溢和下溢时皆递减。通过配置 TIM1_RCR 寄存器的 REP 来调整更新事件产生的频率，重复计数器在 REP+1 个计数周期后产生更新事件。在中央对齐模式下，更新事件在上溢还是在下溢时产生，由写入 REP 的值来决定。

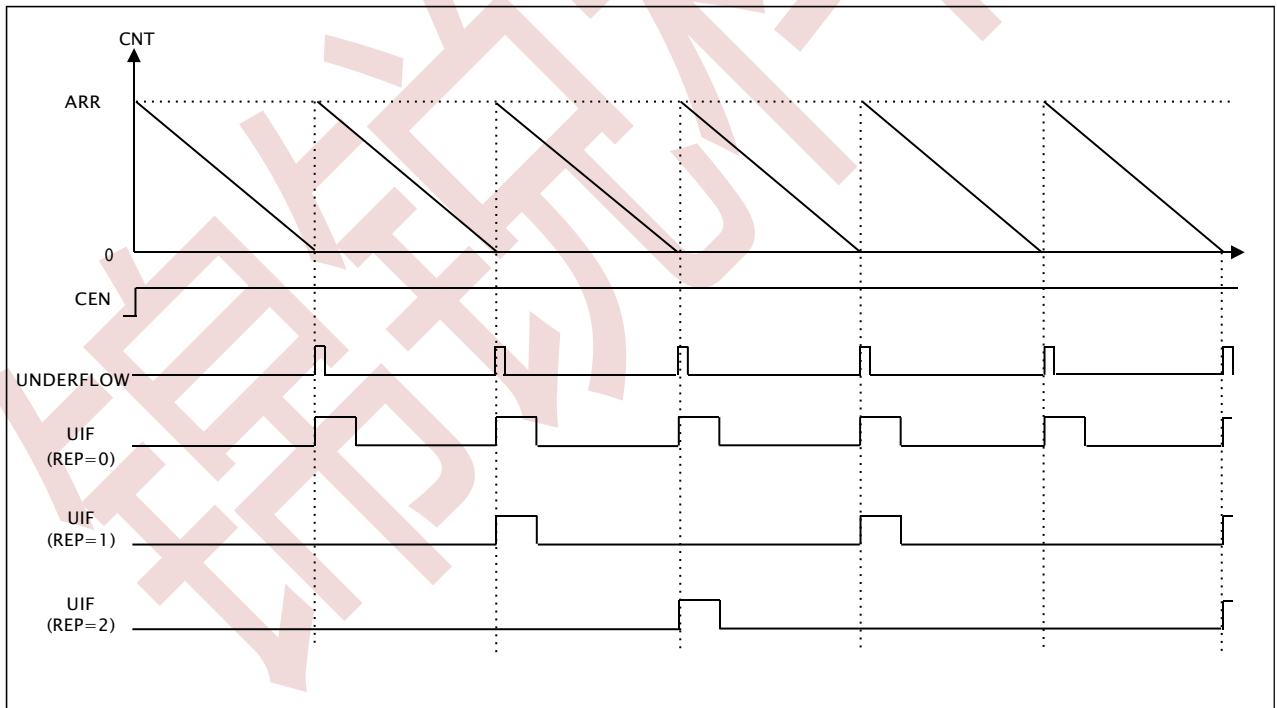
发生更新事件，REP 的值会更新至实时重复计数器 REP_CNT 中。允许对 REP_CNT 实时写入以实现更新事件发生时间点的灵活调整。



中央对齐模式重复计数时序图



边沿对齐模式向上计数时序图



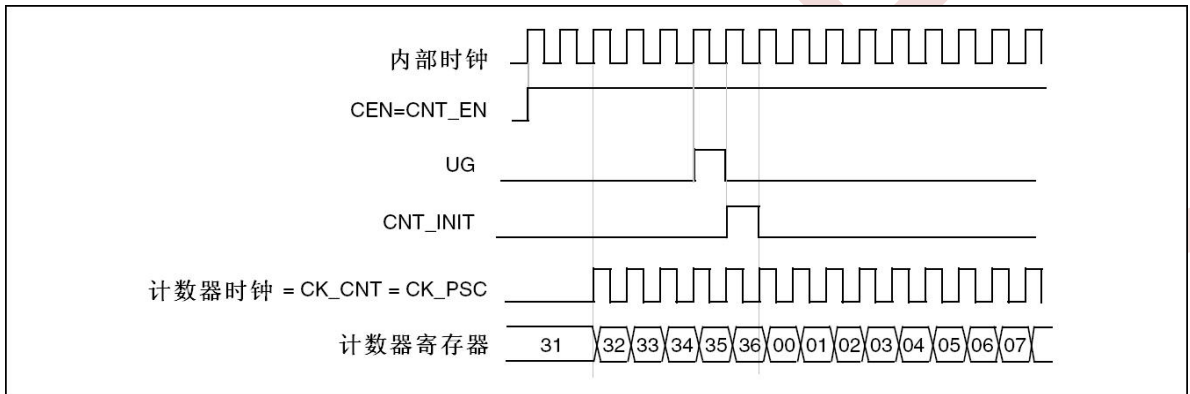
边沿对齐模式向下计数时序图

12.3.4 时钟源

■ 定时器内部时钟：CK_INT

只要 CEN 位被写成 '1'，预分频器的时钟就由内部时钟 CK_INT 提供。控制位为 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器)，并且只能被软件修改 (UG 位仍被自动清除)

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。一般模式下的控制电路，内部时钟分频因子为 1



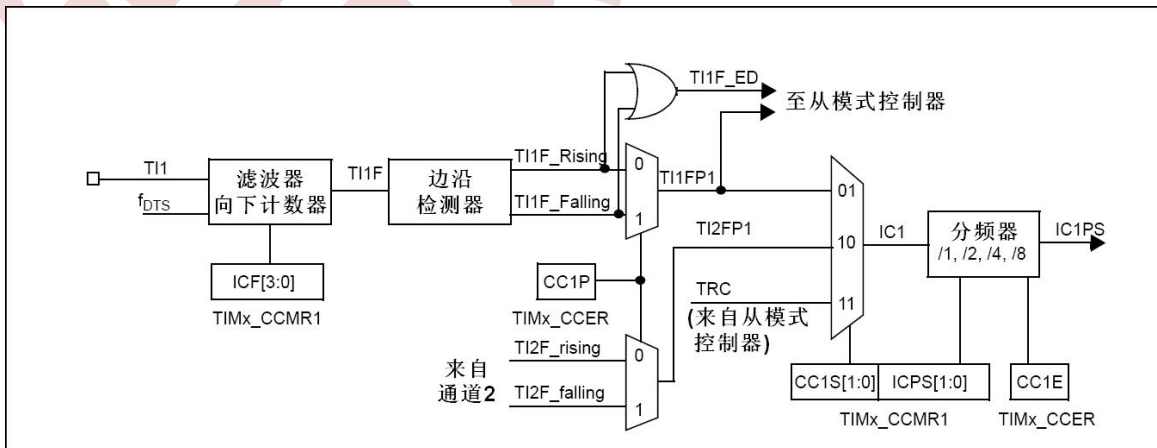
12.3.5 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器(包含影子寄存器)，包括捕获的输入部分(数字滤波、多路复用和预分频器)，和输出部分(比较器和输出控制)。

下图是一个捕获/比较通道概览。

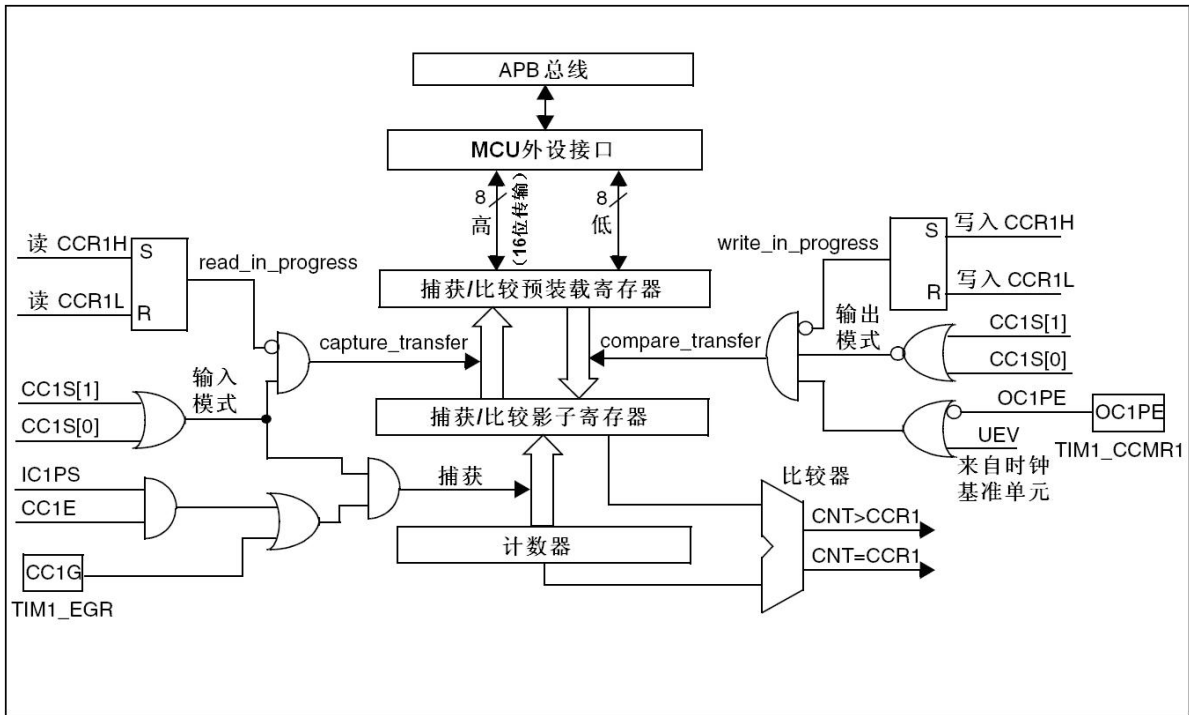
输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号(TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器(ICxPS)。

捕获/比较通道(如：通道 1 输入部分)

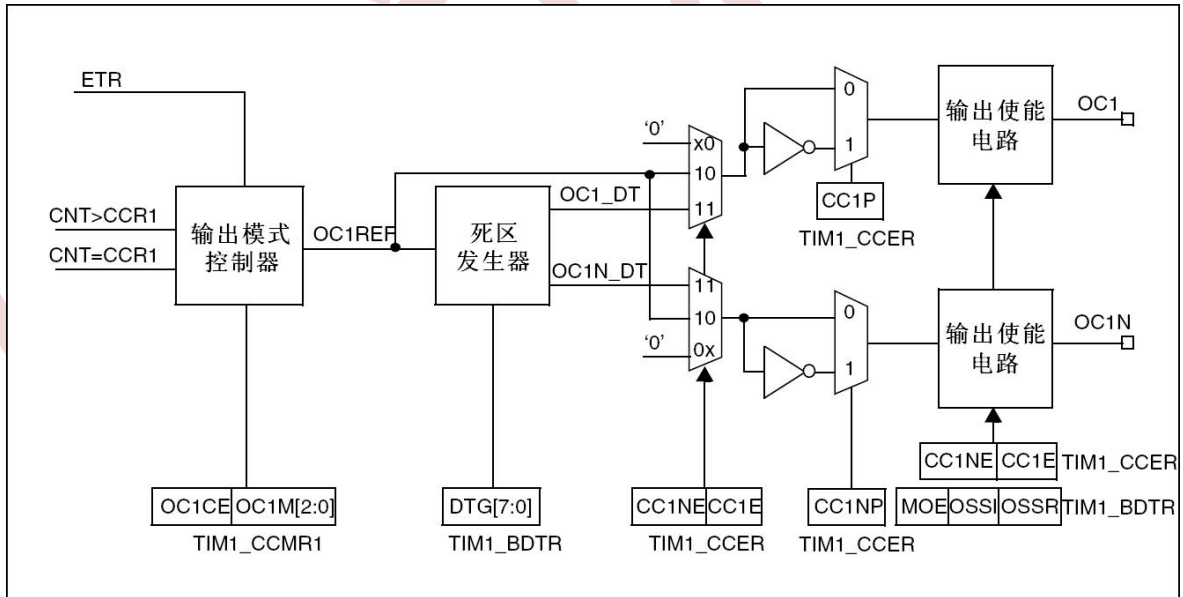


输出部分产生一个中间波形 OCxRef (高有效) 作为基准，链的末端决定最终输出信号的极性。

捕获/比较通道 1 的主电路



捕获/比较通道的输出部分(通道 1 至 4)



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

12.3.6 输入捕获模式

输入捕获模式下，当检测到信号 ICx 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断使能，发生捕获事件时，将产生相应的中断请求。发生捕获事件时，会将状态寄存器 (TIM1_SR) 中的捕获标志位 CCxIF 置 1，通过配置 CCxIF=0 或读取 TIM1_CCRx 中的数据，清除 CCxIF 标志位。当 CCxIF 未被清零时，发生输入捕获事件，重复捕获标志位 CCxOF 将会被置 1，通过配置 CCxOF=0，可以清除 CCxOF 标志位。

例如，通过采样 TI1 输入信号的有效沿，在 TI1 的上升沿来到时捕获当前计数器的值，锁存到 TIM1_CCR1 寄存器中，步骤如下：

1. 配置 TIM1_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上。
2. 配置 TIM1_CCMR1 寄存器的 IC1F[3:0]，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 TIM1_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIM1_CCMR1 寄存器的 IC1PSC[1:0]，选择预分频系数。
5. 配置 TIM1_DIER 寄存器的 CC1IE=1，使能通道 1 的捕获/比较通道 1 中断请求。

注：

- 当通道配置为输入模式时，TIM1_CCRx 寄存器属性变为只读。
- 如果发生了两次以上连续捕获，但 CCxIF 标志未被清零，则重复捕获标志 CCxOF 被置 1。为了避免丢失重复捕获标志 CCxOF 置 1 之前可能产生的捕获信息，建议在读出重复捕获标志之前读取数据。
- 设置 TIM1_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断请求。

12.3.7 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位 (TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

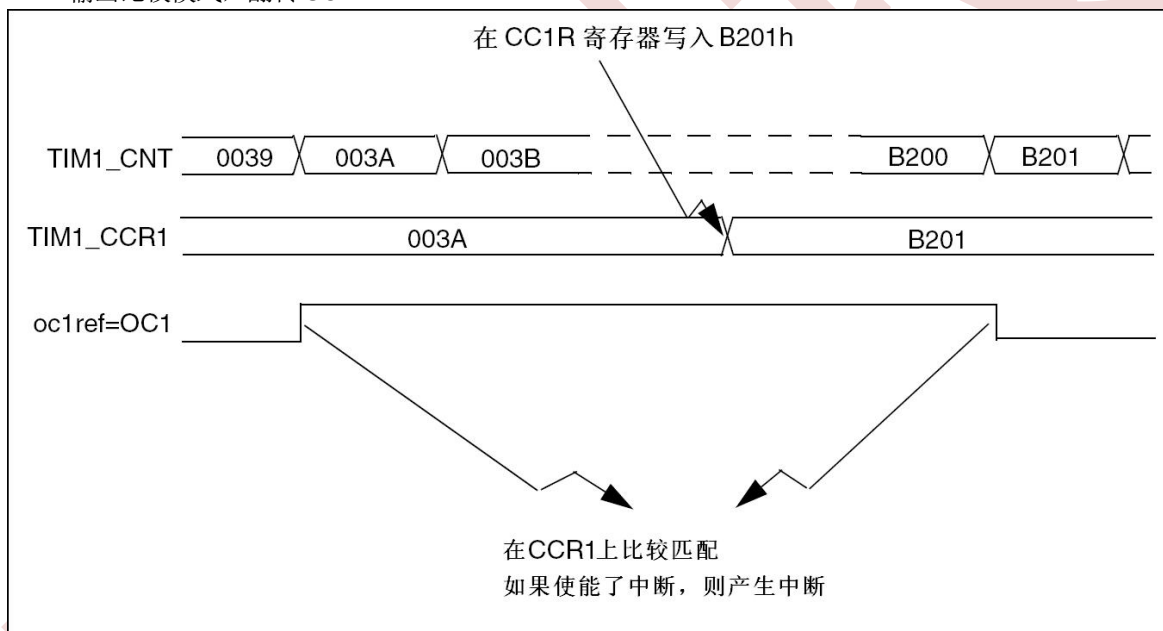
同步的精度可以达到计数器的一个计数周期。

输出比较模式的配置步骤:

1. 配置计数器时钟(内部, 预分频器)。
- 2.将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
- 3.如果要产生一个中断请求, 设置 CCxIE 位。
- 4.选择输出模式, 例如:
 - ◆ 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚, 设置 OCxM=011
 - ◆ 置 OCxPE = 0 禁用预装载寄存器
 - ◆ 置 CCxP = 0 选择极性为高电平有效
 - ◆ 置 CCxE = 1 使能输出
- 5.设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形, 条件是未使用预装载寄存器(OCxPE='0', 否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 9-23 输出比较模式, 翻转 OC1



12.3.8 PWM 模式

在 PWM 模式下, 根据 TIM1_ARR 寄存器和 TIM1_CCRx 寄存器的值, 产生一个频率、占空比可控的 PWM 波形。

配置与通道 x 对应的 TIM1_CCMRx 寄存器的 OCxM=110 或 OCxM=111, 选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下, 计数器和 CCRx 会一直进行比较, 根据配置和比较结果, 通道 x 输出不同的信号, 因此 TIM1 可以产生 4 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 TIM1_CCRx 的预装载功能和 TIM1_ARR 寄存器的预装载功能。写入 TIM1_CCRx 预装载寄存器和 TIM1_ARR 预装载寄存器的值在发生下个更新事件时, 才会生效, 载入相应的影子寄存器。PWM 模式下, 使能计数器前设置 TIM1_EGR 的 UG=1, 产生更新事件用于初始化所有的寄存器。

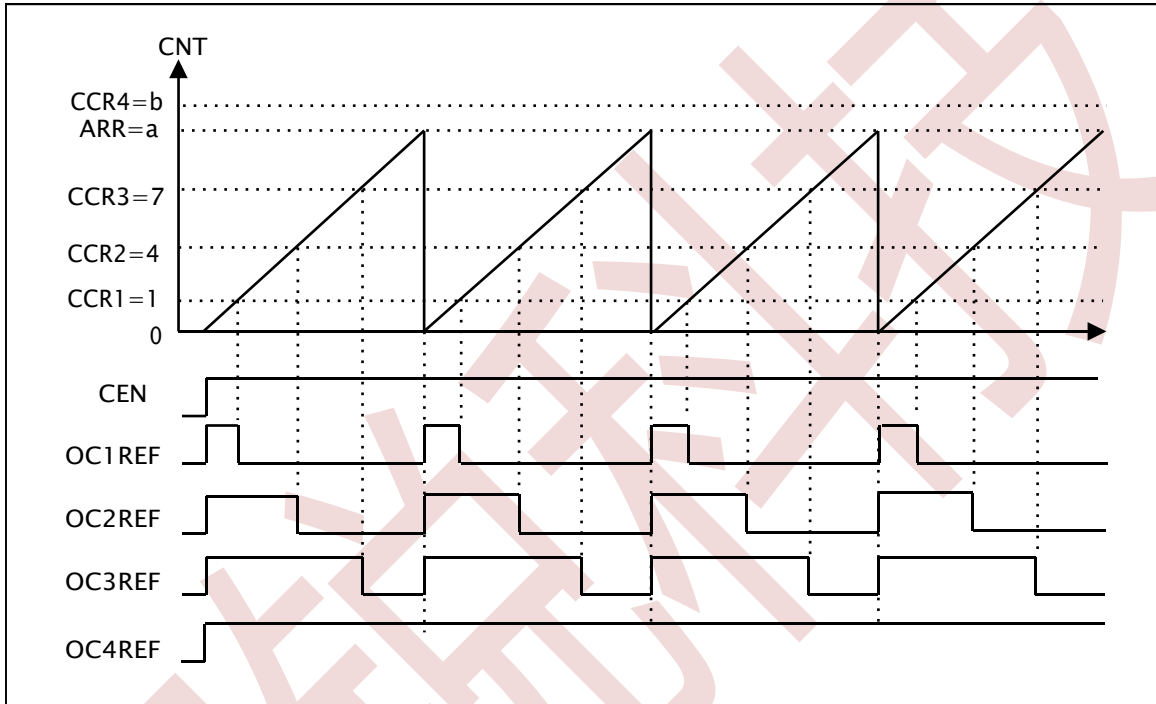
配置 TIM1_CCER 寄存器的 CCxP 选择 OCx 的有效极性。配置 TIM1_CCER 寄存器的 CCxE、CCxNE 位和 TIM1_BDTR 寄存器的 MOE、OSSI、OSSR 位控制 OCx 的输出使能。配置 TIM1_CR1 寄存器的 CMS 位, 可以选择产生边沿对齐或中央对齐的 PWM 信号。

- 1.CMS=00, 边沿对齐模式, 再进一步配置 DIR, 选择递增或递减计数模式。

- 2.CMS=01, 中央对齐模式 1。
- 3.CMS=10, 中央对齐模式 2。
- 4.CMS=11, 中央对齐模式 3。

PWM 边沿对齐模式--向上计数模式

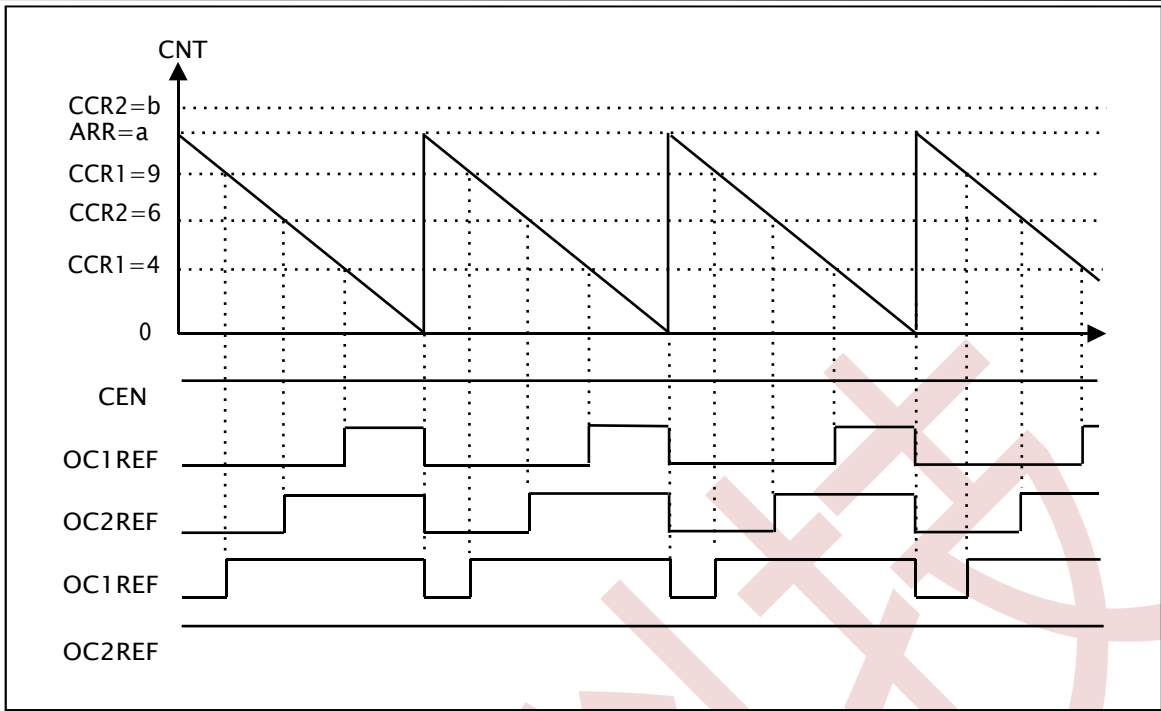
在向上计数模式配置的基础上, 配置 TIM1_CCMRx 寄存器的 CCxS=00, 选择输出模式, OCxM=110, 选择 PWM 模式 1, 当 $TIM1_CNT < TIM1_CCR_x$ 时通道 x (OCxREF) 为有效电平, 否则为无效电平。如果 TIM1_CCRx 中的比较值大于自动重载值(TIM1_ARR), 则 OCxREF 保持为有效电平。如果比较值为 0, 则 OCxREF 保持为无效电平。下图为 CCR1=1, CCR2=4, CCR3=7, CCR4=b, ARR=a 时边沿对齐向上计数时 PWM 模式 1 的波形实例。



边沿对齐向上计数时 PWM 模式 1 的波形

PWM 边沿对齐模式--向下计数模式

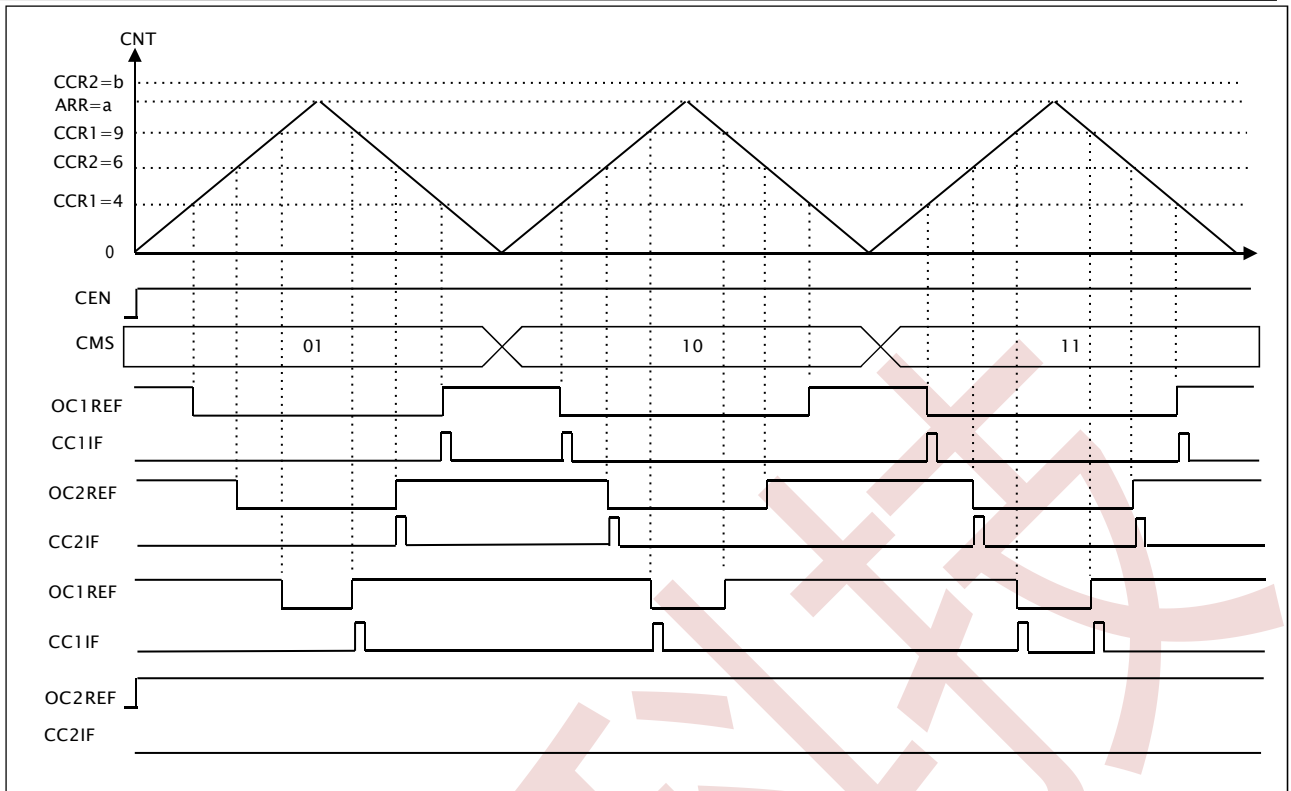
在向下计数模式配置的基础上, 配置 TIM1_CCMRx 寄存器的 CCxS=00, 选择输出模式, OCxM=110, 选择 PWM 模式 1, 当 $TIM1_CNT > TIM1_CCR_x$ 时通道 x (OCxREF) 为无效电平, 否则有效电平。下图为 CCR1=4, CCR2=6, CCR1=9, CCR2=b, ARR=a 时边沿对齐向下计数时 PWM 模式 1 的波形实例。



边沿对齐向下计数时 PWM 模式 1 的波形

PWM 中央对齐模式

首先配置 TIM1 计数器为中央对齐计数模式，配置 TIM1_CCMRx 寄存器的 CCxS=00，选择输出模式，根据配置不同的 CMS，比较输出中断标志位在计数器向下计数时被设置 (CMS=01)、在计数器向上计数时被设置 (CMS=10)、或在计数器向上或向下计数时被设置 (CMS=11)。下图为 CCR1=4, CCR2=6, CCR1=9, CCR2=b, ARR=a 时中央对齐 PWM 模式 1 的波形实例。



中央对齐 PWM 模式 1 的波形

12.3.13 互补输出和死区插入

高级控制定时器 (TIM1) 能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性 (电平转换的延时、电源开关的延时等) 来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性 (主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制: TIMx_CCER 寄存器的 CCxE 和 CCxNE 位, TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位, 详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是, 在转换到 IDLE 状态时 (MOE 下降到 0) 死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区, 如果存在刹车电路, 则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效:

- OCx 输出信号与参考信号相同, 只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反, 只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度 (OCx 或者 OCxN), 则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

带死区插入的互补输出

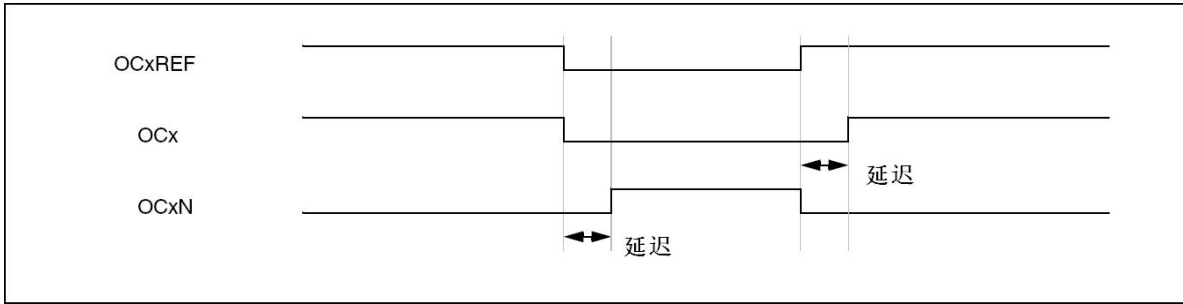


图 9-30 死区波形延迟大于负脉冲

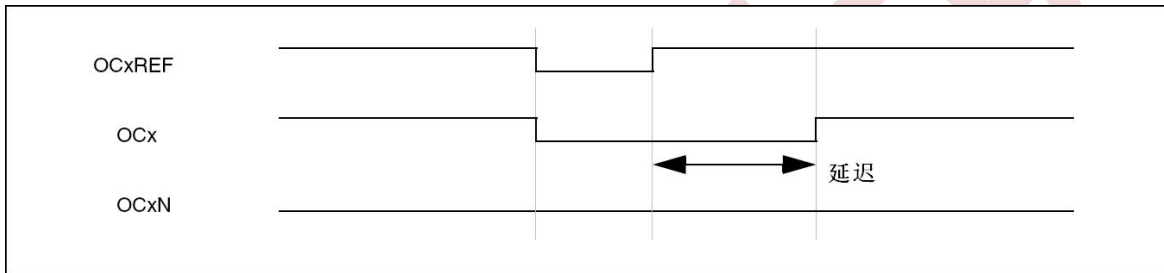
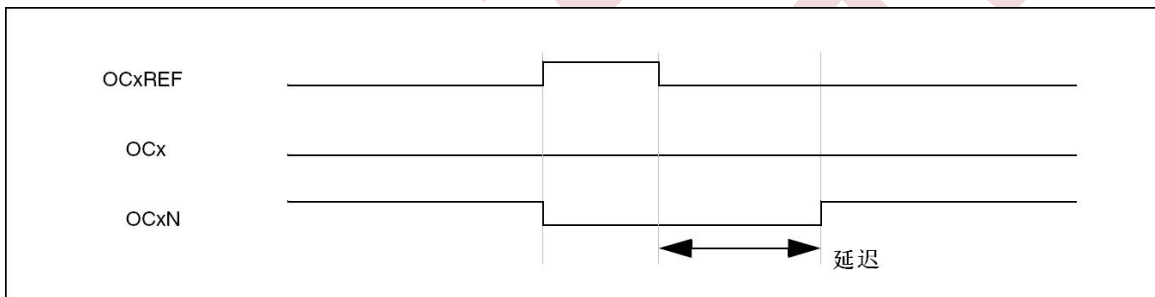


图 9-31 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的，是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。详见死区寄存器 (TIMx_BDTR) 中的延时计算。

12.4 定时器 1 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	TIM1_CR1	控制寄存器 1	0x0000
0x04	TIM1_CR2	控制寄存器 2	0x0000
0x0C	TIM1_DIER	中断使能寄存器	0x0000
0x10	TIM1_SR	状态寄存器	0x0000
0x14	TIM1_EGR	事件产生寄存器	0x0000
0x18	TIM1_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIM1_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIM1_CCER	捕获/比较使能寄存器	0x0000
0x24	TIM1_CNT	计数器	0x0000
0x28	TIM1_PSC	预分频率器	0x0000
0x2C	TIM1_ARR	自动装载寄存器	0x0000
0x30	TIM1_RCR	重复计数寄存器	0x0000
0x34	TIM1_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIM1_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIM1_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIM1_CCR4	捕获/比较寄存器 4	0x0000
0x44	TIM1_DTR	死区寄存器	0x0000
0x48	TIM1_DCR	DMA 控制寄存器	0x0000
0x4C	TIM1_DMAR	连续模式的 DMA 地址	0x0000

12.4.1 TIM1 控制寄存器 1(TIM1_CR1)

偏移地址: 0x00

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]	ARPE	CMS[1:0]	DIR	保留	URS	UDIS	CEN		
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:10	保留, 始终读为 0。
位 9:8	CKD[1:0]: 时钟分频因子 (Clock division) 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(TIx)所用的采样时钟之间的分频比例。 00: tDTS = tCK_INT 01: tDTS = 2 x tCK_INT 10: tDTS = 4 x tCK_INT 11: 保留, 不要使用这个配置
位 7	ARPE: 自动重载预装载允许位 (Auto-reload preload enable) 0: TIMx_ARR 寄存器没有缓冲; 1: TIMx_ARR 寄存器被装入缓冲器。

位 6:5	<p>CMS[1:0]: 选择中央对齐模式 (Center-aligned mode selection)00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。</p> <p>01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下计数时被设置。</p> <p>10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。</p> <p>11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。</p> <p>注: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。</p>
位 4	<p>DIR:方向</p> <p>0: 计数器向上计数;</p> <p>1: 计数器向下计数。</p> <p>注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。</p>
位 3	保留
位 2	<p>URS: 更新请求源 (Update request source)</p> <p>软件通过该位选择 UEV 事件的源</p> <p>0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求:</p> <ol style="list-style-type: none"> 1. 计数器溢出/下溢 2. 设置 UG 位 3. 从模式控制器产生的更新 <p>1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。</p>

<p>位 1</p>	<p>UDIS: 禁止更新 (Update disable) 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生： 1. 计数器溢出/下溢 2. 设置 UG 位 3. 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注：更新影子寄存器) 1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCR_x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。</p>
<p>位 0</p>	<p>CEN: 使能计数器 (Counter enable) 0: 禁止计数器； 1: 使能计数器。 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。</p>

12.4.2 TIM1 控制寄存器 2(TIM1_CR2)

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OIS4 N	OIS4	OIS3 N	OIS3	OIS2 N	OIS2	OIS1 N	OIS1	TIIS	保留						
rw	rw	rw	rw	rw	rw	rw	rw	rw							

位 15	OIS4N: 输出空闲状态 4(OC4N 输出)。参见 OIS1N 位。
位 14	OIS4: 输出空闲状态 4(OC4 输出)。参见 OIS1 位。
位 13	OIS3N: 输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。
位 12	OIS3: 输出空闲状态 3(OC3 输出)。参见 OIS1 位。
位 11	OIS2N: 输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。
位 10	OIS2: 输出空闲状态 2(OC2 输出)。参见 OIS1 位。
位 9	OIS1N: 输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0: 当 MOE=0 时, 死区后 OC1N=0; 1: 当 MOE=0 时, 死区后 OC1N=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
位 8	OIS1: 输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=0; 1: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=1。 注: 已经设置了 LOCK(TIMx_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。
位 7	TIIS: TI1 选择 (TI1 selection) 0: TIMx_CH1 引脚连到 TI1 输入; 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。
位 6: 0	保留

12.4.3 TIM1 DMA/中断使能寄存器(TIM1_DIER)

偏移地址: 0x0C

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CC4D	CC3D	CC2D	CC1D	UDE	保留			CC4I	CC3I	CC2I	CC1I	UIE	
		E	E	E	E					E	E	E	E		
		rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	

位 15:13	保留，始终读为 0。
位 12	CC4DE: 允许捕获/比较 4 的 DMA 请求 (Capture/Compare 4 DMA request enable)0: 禁止捕获/比较 4 的 DMA 请求; 1: 允许捕获/比较 4 的 DMA 请求。
位 11	CC3DE: 允许捕获/比较 3 的 DMA 请求 (Capture/Compare 3 DMA request enable)0: 禁止捕获/比较 3 的 DMA 请求; 1: 允许捕获/比较 3 的 DMA 请求。
位 10	CC2DE: 允许捕获/比较 2 的 DMA 请求 (Capture/Compare 2 DMA request enable)0: 禁止捕获/比较 2 的 DMA 请求; 1: 允许捕获/比较 2 的 DMA 请求。
位 9	CC1DE: 允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable)0: 禁止捕获/比较 1 的 DMA 请求; 1: 允许捕获/比较 1 的 DMA 请求。
位 8	UDE: 允许更新的 DMA 请求 (Update DMA request enable)0: 禁止更新的 DMA 请求; 1: 允许更新的 DMA 请求。
位 7:5	保留。
位 4	CC4IE: 允许捕获/比较 4 中断 (Capture/Compare 4 interrupt enable)0: 禁止捕获/比较 4 中断; 1: 允许捕获/比较 4 中断。
位 3	CC3IE: 允许捕获/比较 3 中断 (Capture/Compare 3 interrupt enable)0: 禁止捕获/比较 3 中断; 1: 允许捕获/比较 3 中断。
位 2	CC2IE: 允许捕获/比较 2 中断 (Capture/Compare 2 interrupt enable)0: 禁止捕获/比较 2 中断; 1: 允许捕获/比较 2 中断。
位 1	CC1IE: 允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable)0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。
位 0	UIE: 允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

12.4.4 TIM1 状态寄存器(TIM1_SR)

偏移地址: 0x10

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CC4 OF	CC3 OF	CC2 OF	CC1 OF	保留					CC4I F	CC3I F	CC2I F	CC1I F	UIF
		rc w0	rc w0	rc w0	rc w0						rc w0	rc w0	rc w0	rc w0	rc w0

位 15:13	保留, 始终读为 0。
位 12	CC4OF: 捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
位 11	CC3OF: 捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
位 10	CC2OF: 捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
位 9	CC1OF: 捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到 TIMx_CCR1 寄存器时, CC1IF 的状态已经为'1'。
位 8: 5	保留, 始终读为 0。
位 4	CC4IF: 捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
位 3	CC3IF: 捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
位 2	CC2IF: 捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
位 1	CC1IF: 捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置 1,但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生; 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIMx_CCR1 清'0'。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

位 0	<p>UIF: 更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。</p> <p>0: 无更新事件产生;</p> <p>1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1':</p> <ul style="list-style-type: none"> - 若 TIMx_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。
-----	---

12.4.5 TIM1 事件产生寄存器(TIM1_EGR)

偏移址:0x14

复值:0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											CC4G	CC3G	CC2G	CC1G	UG
											w	w	w	w	w

位 15:5	保留, 始终读为 0。
位 4	CC4G: 产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
位 3	CC3G: 产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
位 2	CC2G: 产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
位 1	<p>CC1G: 产生捕获/比较 1 事件 (Capture/Compare 1 generation) 该位由软件置'1', 用于产生一个捕获/比较事件, 由硬件自动清'0'。</p> <p>0: 无动作;</p> <p>1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出: 设置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若通道 CC1 配置为输入: 当前的计数器值被捕获至 TIMx_CCR1 寄存器; 设置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若 CC1IF 已经为 1, 则设置 CC1OF=1。</p>
位 0	<p>UG: 产生更新事件 (Update generation)该位由软件置'1', 由硬件自动清'0'。 0: 无动作;</p> <p>1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0'; 若 DIR=1(向下计数)则计数器取 TIMx_ARR 的值。</p>

12.4.6 TIM1 捕获/比较模式寄存器 1(TIM1_CCMR1)

偏移地址: 0x18

复位值 0x0000

通道可用于输入(捕获模式)或输出(比较模式), 通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输入模式下的功能。因此必须注意, 同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OC2M[2:0]			OC2P E	OC2F E	CC2S[1:0]		保留	OC1M[2:0]			OC1P E	OC1F E	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式:

位 15	保留
位 14:12	OC2M[2:0]: 输出比较 2 模式 (Output Compare 2 mode)
位 11	OC2PE: 输出比较 2 预装载使能 (Output Compare 2 preload enable)
位 10	OC2FE: 输出比较 2 快速使能 (Output Compare 2 fast enable)
位 9:8	CC2S[1:0]: 捕获/比较 2 选择。(Capture/Compare 2 selection)该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 通道被配置为输出; 01: CC2 通道被配置为输入, IC2 映射在 TI2 上; 10: CC2 通道被配置为输入, IC2 映射在 TI1 上; 11: 保留 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
位 7	保留

<p>位 6:4</p>	<p>OC1M[2:0]: 输出比较 1 模式 (Output Compare 1 mode)</p> <p>该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用;</p> <p>001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1(TIMx_CCR1)相同时, 强制 OC1REF 为高。</p> <p>010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1(TIMx_CCR1)相同时, 强制 OC1REF 为低。</p> <p>011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平, 否则为无效电 平。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输 出)则该位不能被修改。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
<p>位 3</p>	<p>OC1PE: 输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输 出)则该位不能被修改。</p> <p>注 2: 仅在单脉冲模式下(TIMx_CR1 寄存器的 OPM=1), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。</p>
<p>位 2</p>	<p>OC1FE: 输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。</p> <p>OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
<p>位 1:0</p>	<p>CC1S[1:0]: 捕获 / 比较 1 选择。(Capture/Compare 1 selection)这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: 保留</p> <p>注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>

输入捕获模式:

位 15:12	IC2F[3:0]: 输入捕获 2 滤波器 (Input capture 2 filter)																
位 11:10	IC2PSC[1:0]: 输入/捕获 2 预分频器 (Input capture 2 prescaler)																
位 9:8	<p>CC2S[1:0]: 捕获 / 比较 2 选择 (Capture/Compare 2 selection)这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC2 通道被配置为输出;</p> <p>01: CC2 通道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 通道被配置为输入, IC2 映射在 TI1 上;</p> <p>11: 保留</p> <p>注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。</p>																
位 7:4	<p>IC1F[3:0]: 输入捕获 1 滤波器 (Input capture 1 filter)</p> <p>这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变:</p> <table border="0"> <tr> <td>0000: 无滤波器, 以 f_{DTS} 采样</td> <td>1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=6$</td> </tr> <tr> <td>0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=2$</td> <td>1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=8$</td> </tr> <tr> <td>0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=4$</td> <td>1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=5$</td> </tr> <tr> <td>0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=8$</td> <td>1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=6$</td> </tr> <tr> <td>0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=6$</td> <td>1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=8$</td> </tr> <tr> <td>0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=8$</td> <td>1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=5$</td> </tr> <tr> <td>0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=6$</td> <td>1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=6$</td> </tr> <tr> <td>0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=8$</td> <td>1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=8$</td> </tr> </table>	0000: 无滤波器, 以 f_{DTS} 采样	1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=6$	0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=2$	1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=8$	0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=4$	1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=5$	0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=8$	1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=6$	0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=6$	1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=8$	0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=8$	1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=5$	0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=6$	1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=6$	0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=8$	1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=8$
0000: 无滤波器, 以 f_{DTS} 采样	1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=6$																
0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=2$	1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8, N=8$																
0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=4$	1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=5$																
0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}, N=8$	1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=6$																
0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=6$	1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16, N=8$																
0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2, N=8$	1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=5$																
0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=6$	1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=6$																
0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4, N=8$	1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32, N=8$																
位 3:2	<p>IC1PSC[1:0]: 输入/捕获 1 预分频器 (Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。</p> <p>一旦 CC1E=0(TIMx_CCER 寄存器中), 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p> <p>11: 每 8 个事件触发一次捕获。</p>																
位 1:0	<p>CC1S[1:0]: 捕获 / 比较 1 选择 (Capture/Compare 1 Selection)这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>11: 保留</p> <p>注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>																

12.4.7 TIM1 捕获/比较模式寄存器 2(TIM1_CCMR2)

偏移地址: 0x1C

复位值 0x0000

参看以上 CCMR1 寄存器的描述

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OC4M[2:0]			OC4P E	OC4F E	CC4S[1:0]		保留	OC3M[2:0]			OC3P E	OC3F E	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

输出比较模式:

位 15	保留
位 14:12	OC4M[2:0]: 输出比较 4 模式 (Output compare 4 mode)
位 11	OC4PE: 输出比较 4 预装载使能 (Output compare 4 preload enable)
位 10	OC4FE: 输出比较 4 快速使能 (Output compare 4 fast enable)
位 9:8	CC4S[1:0]: 捕获 / 比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上; 11: 保留 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
位 7	保留
位 6:4	OC3M[2:0]: 输出比较 3 模式 (Output compare 3 mode)
位 3	OC3PE: 输出比较 3 预装载使能 (Output compare 3 preload enable)
位 2	OC3FE: 输出比较 3 快速使能 (Output compare 3 fast enable)
位 1:0	CC3S[1:0]: 捕获 / 比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3 通道被配置为输出; 01: CC3 通道被配置为输入, IC3 映射在 TI3 上; 10: CC3 通道被配置为输入, IC3 映射在 TI4 上; 11: 保留 注: CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式:

位 15:12	IC4F[3:0]: 输入捕获 4 滤波器 (Input capture 4 filter)
位 11:10	IC4PSC[1:0]: 输入/捕获 4 预分频器 (Input capture 4 prescaler)
位 9:8	CC4S[1:0]: 捕获 / 比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上; 11: 保留 注: CC4S 仅在通道关闭时(TIMx_CCER 寄存器的 CC4E=0)才是可写的。
位 7:4	IC3F[3:0]: 输入捕获 3 滤波器 (Input capture 3 filter)
位 3:2	IC3PSC[1:0]: 输入/捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0	<p>CC3S[1:0]: 捕获/比较 3 选择 (Capture/compare 3 selection)这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC3 通道被配置为输出;</p> <p>01: CC3 通道被配置为输入, IC3 映射在 TI3 上;</p> <p>10: CC3 通道被配置为输入, IC3 映射在 TI4 上;</p> <p>11: 保留</p> <p>注: CC3S 仅在通道关闭时(TIMx_CCER 寄存器的 CC3E=0)才是可写的。</p>
-------	--

12.4.8 TIM1 捕获/比较使能寄存器(TIM1_CCER)

偏移地址: 0x20

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	CC4NP: 输入/捕获 4 输出极性。参考 CC1NP 的描述。
位 14	CC4NE: 输入/捕获 4 输出使能。参考 CC1NE 的描述。
位 13	CC4P: 输入/捕获 4 输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
位 12	CC4E: 输入/捕获 4 输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
位 11	CC3NP: 输入/捕获 3 互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
位 10	CC3NE: 输入/捕获 3 互补输出使能 (Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
位 9	CC3P: 输入/捕获 3 输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
位 8	CC3E: 输入/捕获 3 输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
位 7	CC2NP: 输入/捕获 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
位 6	CC2NE: 输入/捕获 2 互补输出使能 (Capture/Compare 2 complementary output enable) 参考 CC1NE 的描述。
位 5	CC2P: 输入/捕获 2 输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
位 4	CC2E: 输入/捕获 2 输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
位 3	<p>CC1NP: 输入/捕获 1 互补输出极性 (Capture/Compare 1 complementary output polarity)</p> <p>0: OC1N 高电平有效;</p> <p>1: OC1N 低电平有效。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出) 则该位不能被修改。</p>

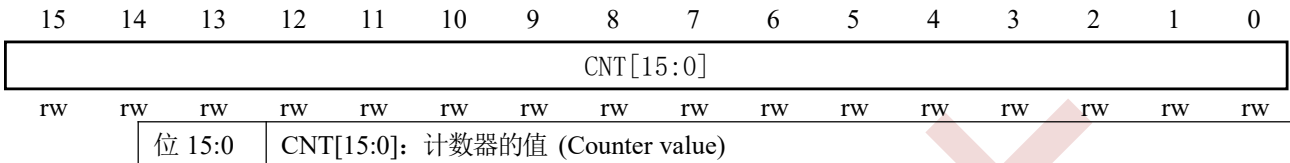
位 2	<p>CC1NE: 输入/捕获 1 互补输出使能 (Capture/Compare 1 complementary output enable)</p> <p>0: 关闭— OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。</p> <p>1: 开启— OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。</p>
位 1	<p>CC1P: 输入/捕获 1 输出极性 (Capture/Compare 1 output polarity)</p> <p>CC1 通道配置为输出:</p> <p>0: OC1 高电平有效;</p> <p>1: OC1 低电平有效。</p> <p>CC1 通道配置为输入:</p> <p>该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。</p> <p>0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相。</p> <p>1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 3 或 2, 则该位不能被修改。</p>

位 0	<p>CC1E: 输入/捕获 1 输出使能 (Capture/Compare 1 output enable)</p> <p>CC1 通道配置为输出:</p> <p>0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。</p> <p>CC1 通道配置为输入:</p> <p>该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。</p> <p>0: 捕获禁止;</p> <p>1: 捕获使能。</p>
-----	---

12.4.9 TIM1 计数器(TIM1_CNT)

偏移地址: 0x24

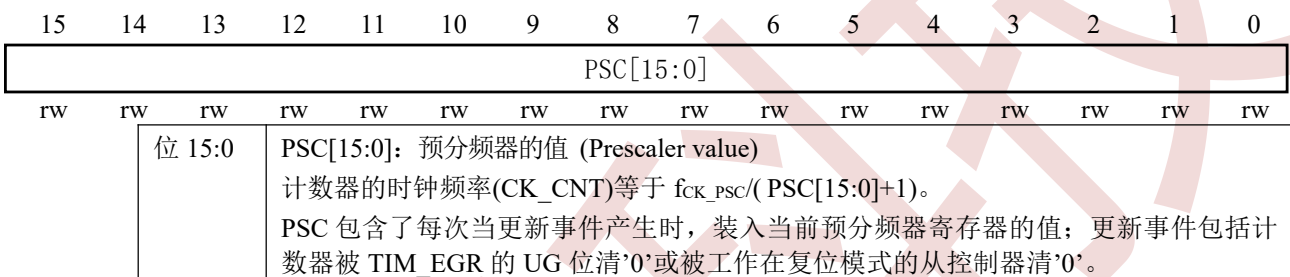
复位值: 0x0000



12.4.10 TIM1 预分频器(TIMx_PSC)

偏移地址: 0x28

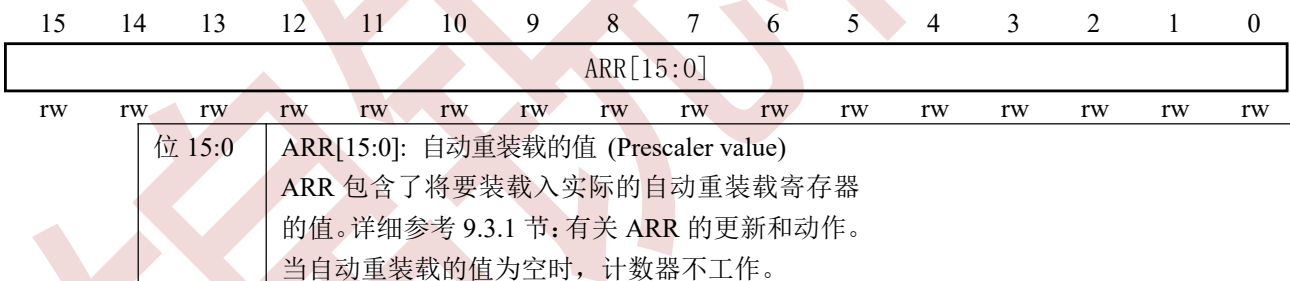
复位值: 0x0000



12.4.11 TIM1 自动重载寄存器(TIMx_ARR)

偏移地址:0x2C

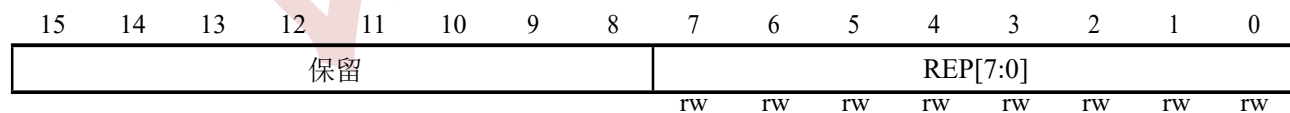
复位值:0x0000



12.4.12 TIM1 重复计数寄存器(TIMx_RCR)

偏移地址: 0x30

复位值: 0x0000



位 15:8	保留，始终读为 0。
位 7:0	<p>REP[7:0]: 重复计数器的值 (Repetition counter value)</p> <p>开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。</p> <p>每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。</p> <p>这意味着在 PWM 模式中，(REP+1)对应着：</p> <ol style="list-style-type: none"> 1. 在边沿对齐模式下，PWM 周期的数目； 2. 在中心对称模式下，PWM 半周期的数目；

12.4.13 TIM1 捕获/比较寄存器 1(TIMx_CCR1)

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>CCR1[15:0]: 捕获/比较通道 1 的值 (Capture/Compare 1 value)</p> <p>若 CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。</p> <p>当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入： CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。</p>														

12.4.14 TIM1 捕获/比较寄存器 2(TIM1_CCR2)

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>CCR2[15:0]: 捕获/比较通道 2 的值 (Capture/Compare 2 value)</p> <p>若 CC2 通道配置为输出： CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。</p> <p>当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入： CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。</p>														

12.4.15 TIM1 捕获/比较寄存器 3(TIM1_CCR3)

偏移地址：0x3C

复位值 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCR3[15:0]															
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
	位 15:0	<p>CCR3[15:0]: 捕获/比较通道 3 的值 (Capture/Compare 3 value)</p> <p>若 CC3 通道配置为输出： CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。</p> <p>如果在 TIMx_CCMR3 寄存器(OC3PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。</p> <p>当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入： CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。</p>														

12.4.16 TIM1 捕获/比较寄存器 4(TIM1_CCR4)

偏移地址：0x40

复位值 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCR4[15:0]															
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
	位 15:0	<p>CCR4[15:0]: 捕获/比较通道 4 的值 (Capture/Compare 4 value)</p> <p>若 CC4 通道配置为输出： CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。</p> <p>如果在 TIMx_CCMR4 寄存器(OC4PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。</p> <p>当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入： CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。</p>														

12.4.17 TIM1 死区寄存器(TIM1_BDTR)

偏移地址：0x44

复位值 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOE	保留			OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
	r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	<p>MOE: 主输出使能 (Main output enable)</p> <p>0: 禁止 OC 和 OCN 输出或强制为空闲状态;</p> <p>1: 如果设置了相应的使能位(TIMx_CCER 寄存器的 CCxE、CCxNE 位), 则开启 OC 和 OCN 输出。</p>
位 14: 12	保留
位 11	<p>OSSR: 运行模式下“关闭状态”选择 (Off-state selection for Run mode)</p> <p>该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明(TIM1 捕获/比较使能寄存器(TIMx_CCER))。</p> <p>0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, 首先开启 OC/OCN 并输出无效电平, 然后置 OC/OCN 使能输出信号=1。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。</p>
位 10	<p>OSSI: 空闲模式下“关闭状态”选择 (Off-state selection for Idle mode)</p> <p>该位用于当 MOE=0 且通道设为输出时。</p> <p>参考 OC/OCN 使能的详细说明(TIM1 捕获/比较使能寄存器(TIMx_CCER))。</p> <p>0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, OC/OCN 首先输出其空闲电平, 然后 OC/OCN 使能输出信号=1。</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。</p>
位 9:8	<p>LOOK[1:0]: 锁定设置 (Lock configuration)该位为防止软件错误而提供写保护。</p> <p>00: 锁定关闭, 寄存器无写保护;</p> <p>01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位;</p> <p>10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位; 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位);</p> <p>注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容冻结直至复位。</p>

位 7:0	<p>UTG[7:0]: 死区发生器设置 (Dead-time generator setup)</p> <p>这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间:</p> <p>DTG[7:5]=0xx => $DT=DTG[7:0] \times T_{dtg}$, $T_{dtg} = T_{DTS}$;</p> <p>DTG[7:5]=10x => $DT=(64+DTG[5:0]) \times T_{dtg}$, $T_{dtg} = 2 \times T_{DTS}$;</p> <p>DTG[7:5]=110 => $DT=(32+DTG[4:0]) \times T_{dtg}$, $T_{dtg} = 8 \times T_{DTS}$;</p> <p>DTG[7:5]=111 => $DT=(32+DTG[4:0]) \times T_{dtg}$, $T_{dtg} = 16 \times T_{DTS}$;</p> <p>例: 若 TDS = 125ns(8MHZ), 可能的死区时间为:</p> <p>0 到 15875ns, 若步长时间为 125ns;</p> <p>16us 到 31750ns, 若步长时间为 250ns;</p> <p>32us 到 63us, 若步长时间为 1us;</p> <p>64us 到 126us, 若步长时间为 2us;</p> <p>注: 一旦 LOCK 级别(TIMx_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3, 则不能修改这些位。</p>
-------	--

12.4.18 TIM1 DMA 控制寄存器(TIM1_DCR)

偏移地址: 0x48

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DBL[4:0]					保留			DBA[4:0]					
		rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	

位 15:13	保留, 始终读为 0。
位 12:8	<p>DBL[4:0]: DMA 连续传送长度 (DMA burst length)</p> <p>这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字(双字节)或字节:</p> <p>00000: 1 次传输 00001: 2 次传输</p> <p>00010: 3 次传输 </p> <p>..... 10001: 18 次传输</p> <p>例: 我们考虑这样的传输: DBL=7, DBA=TIM2_CR1</p> <p>如果 DBL=7, DBA=TIM2_CR1 表示待传输数据的地址, 那么传输的地址由下式给出: (TIMx_CR1 的地址) + DBA + (DMA 索引), 其中 DMA 索引 = DBL</p> <p>其中(TIMx_CR1 的地址) + DBA 再加上 7, 给出了将要写入或者读出数据的地址, 这样数据的传输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。</p> <p>根据 DMA 数据长度的设置, 可能发生以下情况:</p> <p>如果设置数据为半字(16 位), 那么数据就会传输给全部 7 个寄存器。</p> <p>如果设置数据为字节, 数据仍然会传输给全部 7 个寄存器: 第一个寄存器包含第一个 MSB 字节, 第二个寄存器包含第一个 LSB 字节, 以此类推。因此对于定时器, 用户必须指定由 DMA 传输的数据宽度。</p>
位 7:5	保留, 始终读为 0。
位 4:0	<p>DBA[4:0]: DMA 基地址 (DMA base address)</p> <p>这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时), DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量:</p> <p>00000: TIMx_CR1,</p> <p>00001: TIMx_CR2,</p> <p>00010: TIMx_SMCR,</p>

--	-------

12.4.19 TIM1 连续模式的 DMA 地址(TIM1_DMAR)

偏移地址: 0x4C

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0	DMAB[15:0]: DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + DBA + DMA 索引, 其中: “TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1)所在的地 址; “DBA”是 TIMx_DCR 寄存器中定义的基地址; “DMA 索引”是由 DMA 自动控制的偏移量,它取决于 TIMx_DCR 寄存器中定义的 DBL。
--------	---

13 通用定时器 14 (TIM14)

13.1 TIM14 简介

通用控制定时器(TIM14)由一个 16 位的自动装载计数器组成,它由一个可编程的预分频器驱动。

它适合多种用途,产生输出波形(输出比较、PWM 等)。

使用定时器预分频器和 RCC 时钟控制预分频器,可以实现脉冲宽度和波形周期的调节。

13.2 TIM14 主要特性

- 16 位自动装载计数器。(可实现向上计数、向下计数、向上/下计数)
- 16 位可编程预分频器。(分频系数可配置为 1 到65536 之间的任意值)
- 可编程重复计数器
- 最多2 个独立通道:
- ◆ 输出比较
- ◆ PWM 生成
 - 如下事件发生时产生中断/DMA:
 - ◆ 更新事件
 - ◆ 输出比较

13.3 TIM14 功能描述

13.3.1 时基单元

TIM14 的时基单元主要包括:计数器寄存器 (TIM1_CNT)、预分频器寄存器 (TIM1_PSC)、自动预装载寄存器 (TIM1_ARR) 和重复计数器寄存器 (TIM1_RCR)。

计数单元由一个 16 位的计数器和对应的自动预装载寄存器组成,可以实现递增计数功能。

计数器的时钟由预分频器提供,预分频器由预分频计数器和对应的寄存器组成,分频系数为 1-65536,可以随时写入,在下次更新事件时生效。

自动预装载寄存器有预装载功能的 16 位影子寄存器,通过设置 TIM1_CR1 寄存器的 ARPE 位选择写入 ARR 寄存器的值立即生效或发生更新事件时载入影子寄存器。

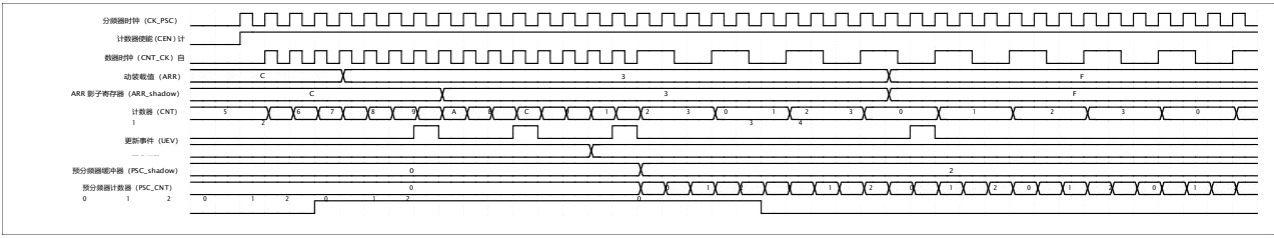
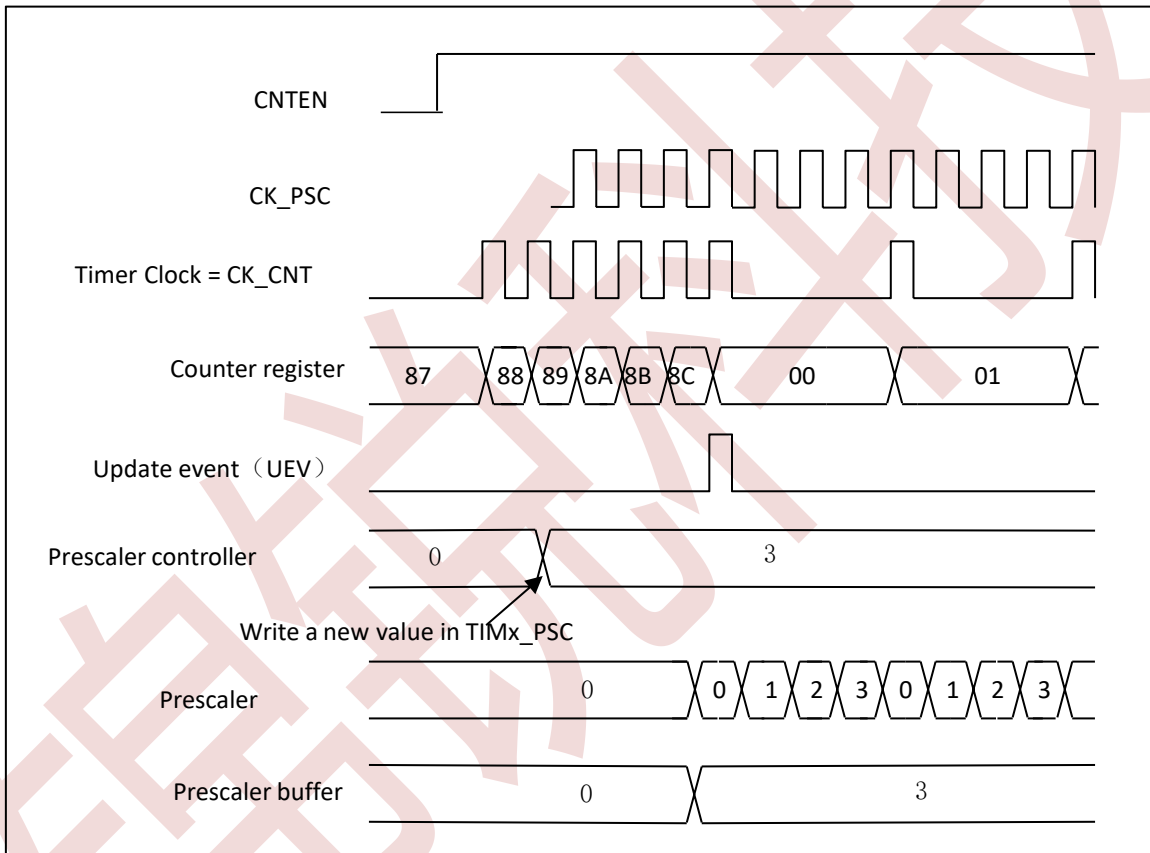


图 9-2 自动预装载

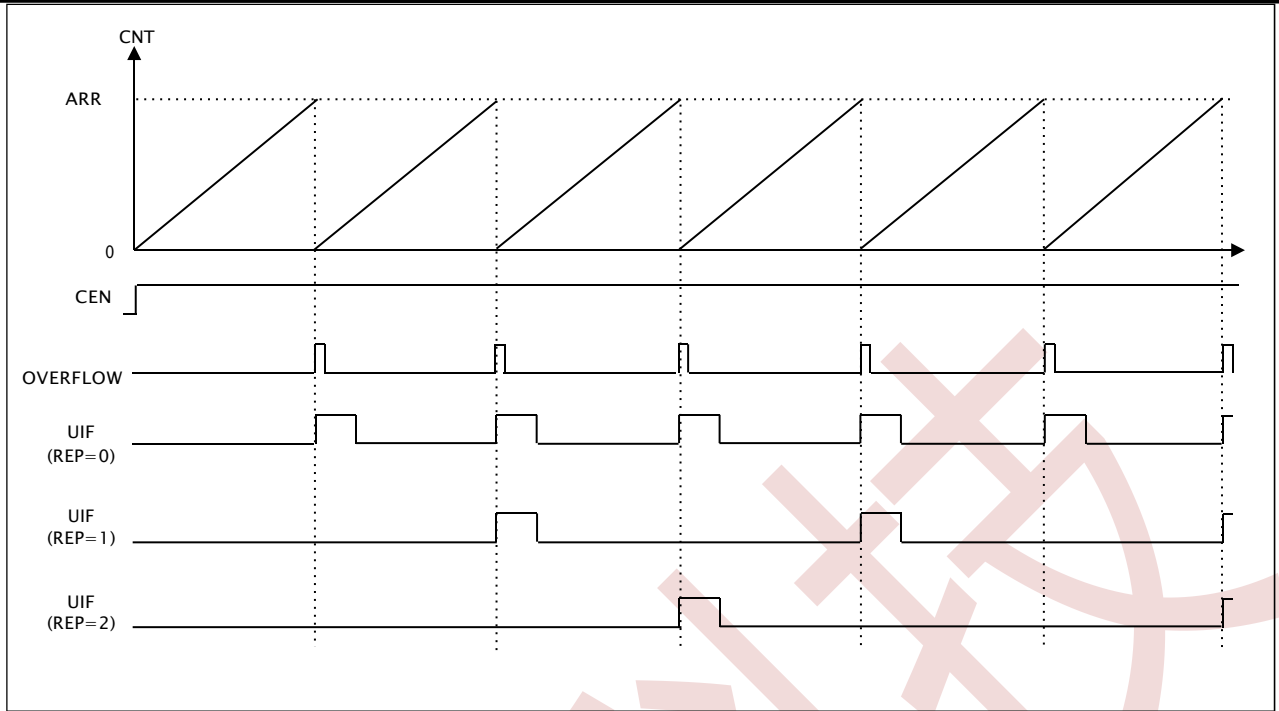
TIMx_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图9-3 当预分频的参数从 1 到4，计数器的时序图



13.3.2 计数器模式

向上计数模式下，在使能 TIM1_CR1 寄存器的 CEN 后计数器由 0 开始向上计数，直至 TIM1_ARR 的值，产生一个计数器上溢事件（更新事件），并从 0 开始重新递增计数。当用户启用了重复计数功能，重复计数器在每次上溢事件时递减计数，只有当重复计数器从设定值递减到 0 时，才会产生更新事件。设置 IM1_EGR 寄存器的 UG=1，同样可以产生一个更新事件。



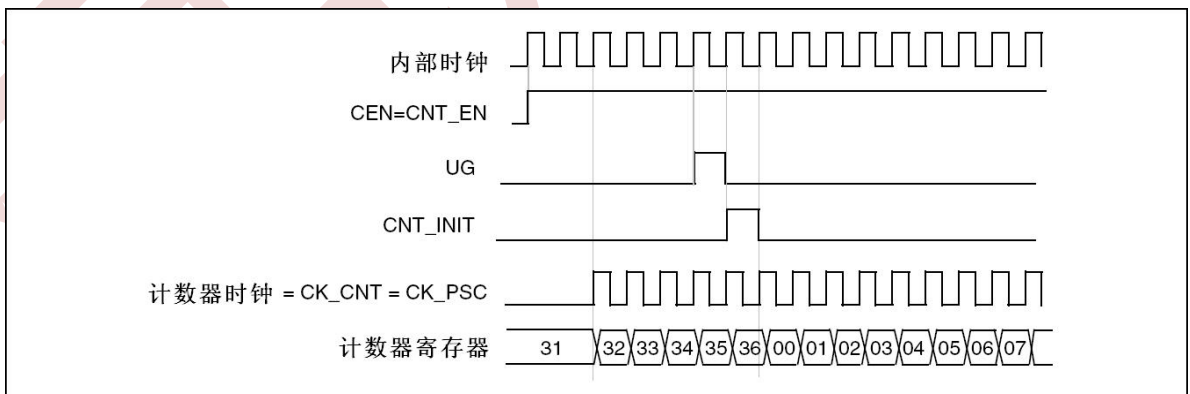
边上计数时序图

13.3.4 时钟源

内部时钟源(CK_INT)

只要 CEN 位被写成'1'，预分频器的时钟就由内部时钟 CK_INT 提供。控制位为 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器)，并且只能被软件修改 (UG 位仍被自动清除)

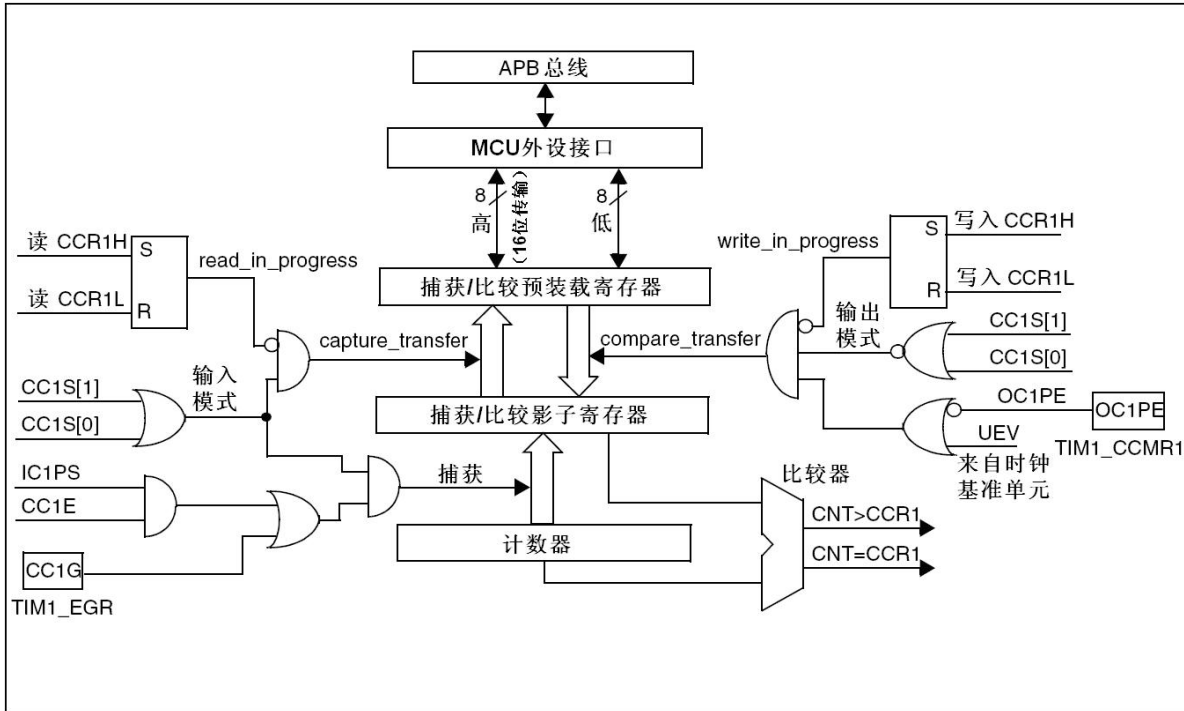
下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。一般模式下的控制电路，内部时钟分频因子为1



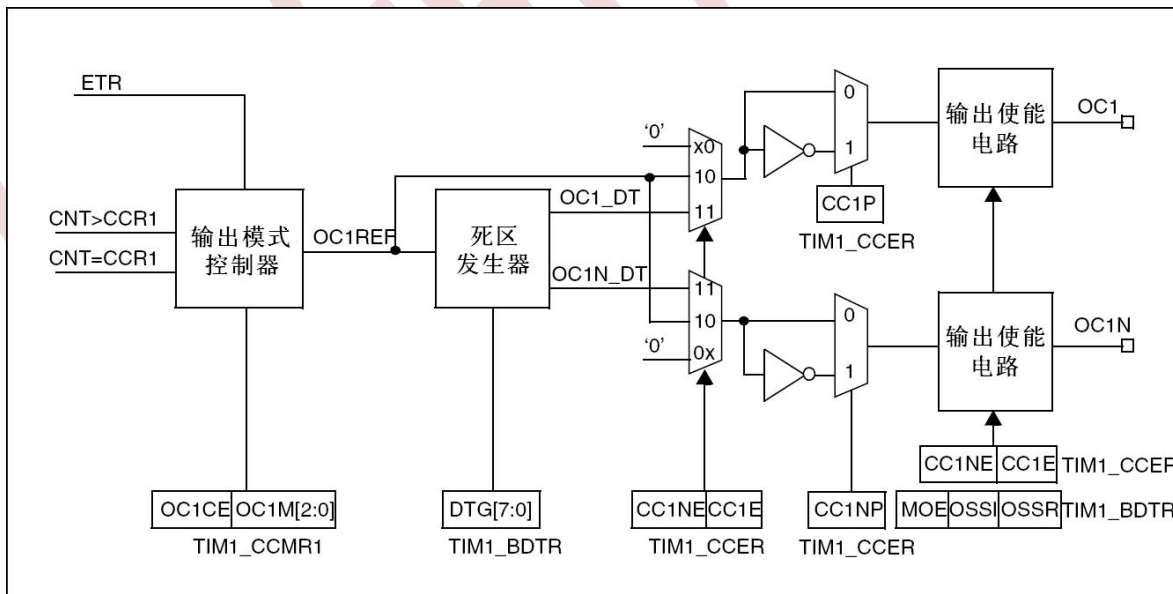
13.3.5 比较通道

每一个比较通道都是围绕着一个比较寄存器(包含影子寄存器).

比较通道主电路



比较通道的输出部分(通道 1 至 2)



比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

13.3.6 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。

当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIM14_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

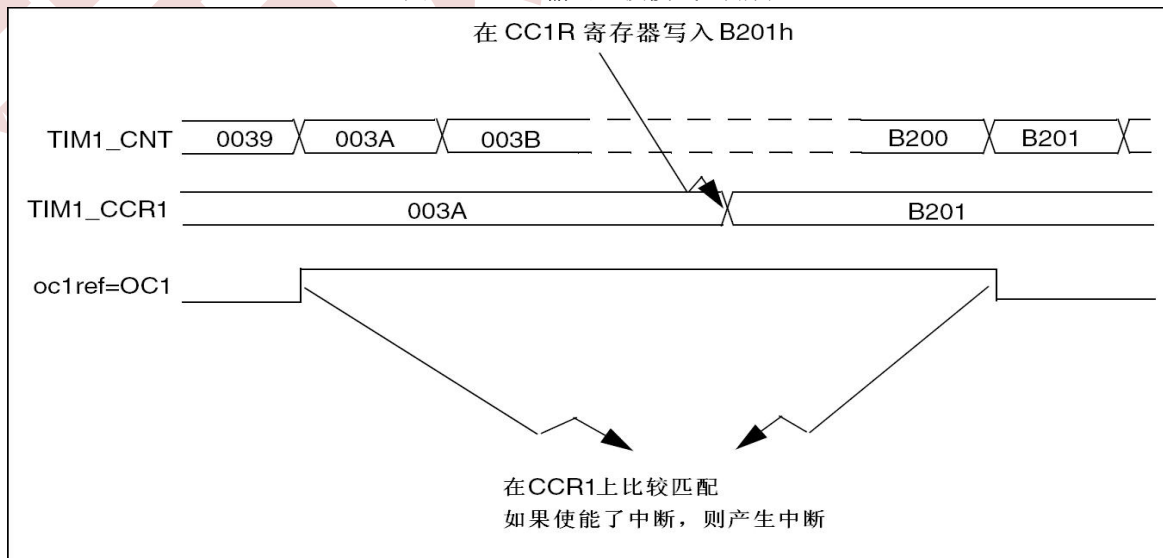
同步的精度可以达到计数器的一个计数周期。

输出比较模式的配置步骤：

- 1.配置计数器时钟(内部，预分频器)。
- 2.将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
- 3.如果要产生一个中断请求，设置 CCxIE 位。
- 4.选择输出模式，例如：
 - ◆ 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
 - ◆ 置 OCxPE = 0 禁用预装载寄存器
 - ◆ 置 CCxP = 0 选择极性为高电平有效
 - ◆ 置 CCxE = 1 使能输出
- 5.设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE='0')，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 9-23 输出比较模式，翻转 OC1



13.3.7 PWM 模式

在 PWM 模式下，根据 TIM14_ARR 寄存器和 TIM14_CCRx 寄存器的值，产生一个频率、占空比可控的 PWM 波形。

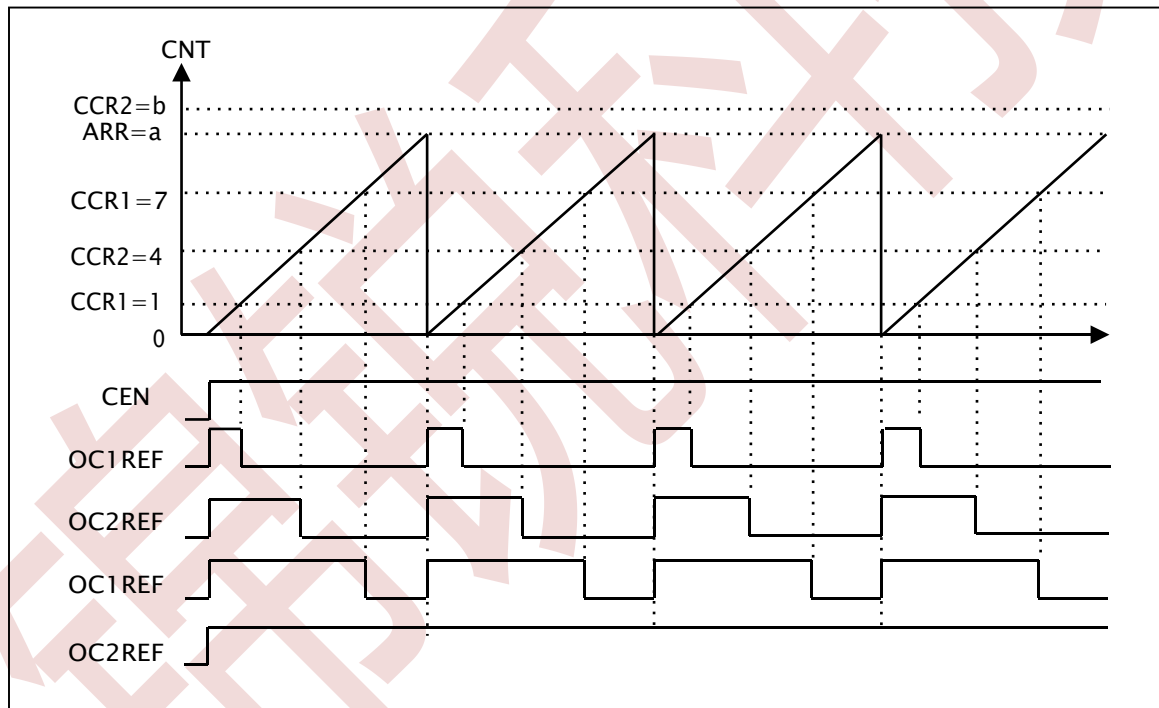
配置与通道 x 对应的 TIM1_CCMRx 寄存器的 OCxM=110 或 OCxM=111，选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下，计数器和 CCRx 会一直进行比较，根据配置和比较结果，通道 x 输出不同的信号，因此 TIM1 可以产生 4 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 TIM1_CCRx 的预装载功能和 TIM1_ARR 寄存器的预装载功能。写入 TIM1_CCRx 预装载寄存器和 TIM1_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 TIM1_EGR 的 UG=1，产生更新事件用于初始化所有的寄存器。

配置 TIM1_CCER 寄存器的 CCxP 选择 OCx 的有效极性。配置 TIM1_CCER 寄存器的 CCxE 位，配置 TIM1_CR1 寄存器的 CMS 位，可以选择产生边沿对齐的 PWM 信号。

1. CMS=00，边沿对齐模式，再进一步配置 DIR，选择递增或递减计数模式。

PWM 边沿对齐模式--向上计数模式

在向上计数模式配置的基础上，配置 TIM14_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，选择 PWM 模式 1，当 TIM14_CNT < TIM14_CCRx 时通道 x (OCxREF) 为有效电平，否则为无效电平。如果 TIM14_CCRx 中的比较值大于自动重装载值 (TIM14_ARR)，则 OCxREF 保持为有效电平。如果比较值为 0，则 OCxREF 保持为无效电平。下图为 CCR1=1, CCR2=4, CCR1=7, CCR2=b, ARR=a 时边沿对齐向上计数时 PWM 模式 1 的波形实例。



边沿对齐向上计数时 PWM 模式 1 的波形

13.4 定时器 14 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	TIM14_CR1	控制寄存器 1	0x0000
0x04	TIM14_CR2	控制寄存器 2	0x0000
0x0C	TIM14_DIER	中断使能寄存器	0x0000
0x10	TIM14_SR	状态寄存器	0x0000
0x14	TIM14_EGR	事件产生寄存器	0x0000
0x18	TIM14_CCMR1	比较模式寄存器 1	0x0000
0x1C	保留		0x0000
0x20	TIM14_CCER	比较使能寄存器	0x0000
0x24	TIM14_CNT	计数器	0x0000
0x28	TIM14_PSC	预分频率器	0x0000
0x2C	TIM14_ARR	自动装载寄存器	0x0000
0x30	TIM14_RCR	重复计数寄存器	0x0000
0x34	TIM14_CCR1	比较寄存器 1	0x0000
0x38	TIM14_CCR2	比较寄存器 2	0x0000
0x3C	保留		0x0000
0x40	保留		0x0000
0x44	保留		0x0000
0x48	TIM14_DCR	DMA 控制寄存器	0x0000
0x4C	TIM14_DMAR	连续模式的 DMA 地址	0x0000

13.4.1 TIM14 控制寄存器 1(TIM14_CR1)

偏移地址: 0x00

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留				保留		ARP	CMS[1:0]		DIR	保留	URS	UDIS	CEN			
				rw		rw	rw		rw	rw	rw	rw	rw	rw		

位 15:10	保留, 始终读为 0。
位 9:8	保留
位 7	ARPE: 自动重装载预装载允许位 (Auto-reload preload enable) 0: TIMx_ARR 寄存器没有缓冲;

	1: TIMx_ARR 寄存器被装入缓冲器。
位 6:3	保留
位 2	<p>URS: 更新请求源 (Update request source) 软件通过该位选择 UEV 事件的源</p> <p>0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求:</p> <ul style="list-style-type: none"> 4. 计数器溢出 5. 设置 UG 位 <p>1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出才产生更新中断或 DMA 请求。</p>
位 1	<p>UDIS: 禁止更新 (Update disable) 软件通过该位允许/禁止 UEV 事件的产生</p> <p>0: 允许 UEV。更新(UEV)事件由下述任一事件产生:</p> <ul style="list-style-type: none"> 4. 计数器溢出 5. 设置 UG 位 6. 从模式控制器产生的更新 <p>具有缓存的寄存器被装入它们的预装载值。(译注: 更新影子寄存器)</p> <p>1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCR_x)保持它们的值。如果设置了 UG 位, 则计数器和预分频器被重新初始化。</p>
位 0	<p>CEN: 使能计数器 (Counter enable)</p> <p>0: 禁止计数器;</p> <p>1: 使能计数器。</p>

13.4.2 TIM14 控制寄存器 2(TIM14_CR2)

偏移地址: 0x04

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				OIS2	保留	OIS1	保留								
				rw			rw								

位 15: 12	保留
位 11	OIS2: 输出空闲状态 2(OC2 输出)。参见 OIS1 位。
位 10	保留
位 9	保留
位 8	OIS1: 输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=0; 1: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=1。
位 7	保留
位 6: 4	保留
位 3	CCDS: 比较的 DMA 选择 0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求; 1: 当发生更新事件时, 送出 CCx 的 DMA 请求。
位 2: 0	保留

13.4.3 TIM14 DMA/中断使能寄存器(TIM14_DIER)

偏移地址: 0x0C

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CC2D E	CC1D E	UDE	保留					CC2I E	CC1I E	UIE
					rw	rw	rw						rw	rw	rw

位 15:11	保留, 始终读为 0。
位 10	CC2DE: 允许比较 2 的 DMA 请求 (Compare 2 DMA request enable)0: 禁止比较 2 的 DMA 请求; 1: 允许比较 2 的 DMA 请求。
位 9	CC1DE: 允许比较 1 的 DMA 请求 (Compare 1 DMA request enable)0: 禁止比较 1 的 DMA 请求; 1: 允许比较 1 的 DMA 请求。
位 8	UDE: 允许更新的 DMA 请求 (Update DMA request enable)0: 禁止更新的 DMA 请求; 1: 允许更新的 DMA 请求。
位 7:3	保留。
位 2	CC2IE: 允许比较 2 中断 (Compare 2 interrupt enable)0: 禁止比较 2 中断; 1: 允许比较 2 中断。
位 1	CC1IE: 允许比较 1 中断 (Compare 1 interrupt enable)0: 禁止比较 1 中断; 1: 允许比较 1 中断。
位 0	UIE: 允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

13.4.4 TIM14 状态寄存器(TIM14_SR)

偏移地址: 0x10

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CC2 OF	CC1 OF	保留						CC2I F	CC1I F	UIF	
				rc w0	rc w0							rc w0	rc w0	rc w0	

位 15:11	保留, 始终读为 0。
位 10	保留
位 9	保留
位 8: 3	保留, 始终读为 0。
位 2	CC2IF: 比较 2 中断标记 (Compare 2 interrupt flag) 参考 CC1IF 描述。
位 1	CC1IF: 比较 1 中断标记 (Compare 1 interrupt flag) 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置 1,但在中心对称模式下除外(参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生; 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。
位 0	UIF: 更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1': - 若 TIMx_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。

13.4.5 TIM14 事件产生寄存器(TIM14_EGR)

偏移址:0x14

复值:0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CC2G	CC1G	UG
													w	w	w

位 15:3	保留，始终读为 0。
位 2	CC2G：产生比较 2 事件 (Compare 2 generation) 参考 CC1G 描述。
位 1	CC1G：产生比较 1 事件 (Compare 1 generation) 该位由软件置'1'，用于产生一个比较事件，由硬件自动清'0'。 0：无动作； 1：在通道 CC1 上产生一个比较事件；若开启对应的中断和 DMA，则产生相应的中断和 DMA。
位 0	UG：产生更新事件 (Update generation)该位由软件置'1'，由硬件自动清'0'。0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。

13.4.6 TIM14 比较模式寄存器 1(TIM14_CCMR1)

偏移地址: 0x18

复位值 0x0000

通道可用输出(比较模式), 通道的方向由相应的 CCxS 位定义。OCxx 描述了通道在输出模式下的功能,

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		保留	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w

输出比较模式:

位 15	保留
位 14:12	OC2M[2:0]: 输出比较 2 模式 (Output Compare 2 mode)
位 11	OC2PE: 输出比较 2 预装载使能 (Output Compare 2 preload enable)
位 10	OC2FE: 输出比较 2 快速使能 (Output Compare 2 fast enable)
位 9:8	CC2S[1:0]: 比较 2 选择。(Compare 2 selection)该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 通道被配置为输出; 01: 保留; 10: 保留; 11: 保留 注: CC2S 仅在通道关闭时(TIMx_CCER 寄存器的 CC2E=0)才是可写的。
位 7	保留
位 6:4	OC1M[2:0]: 输出比较 1 模式 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与比较寄存器 1(TIMx_CCR1)相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与比较寄存器 1(TIMx_CCR1)相同时, 强制 OC1REF 为低。 011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM 模式 2— 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平, 否则为无效电 平。

	<p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
位 3	<p>OC1PE: 输出比较 1 预装载使能 (Output Compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p>
位 2	<p>OC1FE: 输出比较 1 快速使能 (Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>
位 1:0	<p>CC1S[1:0]: 比较 1 选择。(Compare 1 selection) 这 2 位定义通道的方向(输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: 保留; 10: 保留; 11: 保留 注: CC1S 仅在通道关闭时(TIMx_CCER 寄存器的 CC1E=0)才是可写的。</p>

13.4.7 TIM14 比较使能寄存器(TIM14_CCER)

偏移地址：0x20

复位值 0x0000

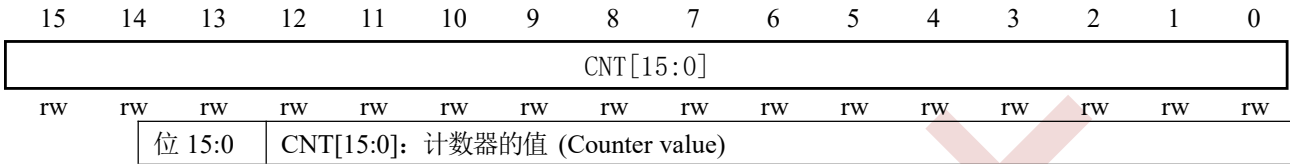
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										CC2P	CC2E	保留		CC1P	CC1E
										rw	rw			rw	rw

位 15: 6	保留
位 5	CC2P: 捕获 2 输出极性 (Compare 2 output polarity) 参考 CC1P 的描述。
位 4	CC2E: 捕获 2 输出使能 (Compare 2 output enable) 参考 CC1E 的描述。
位 3:2	保留
位 1	CC1P: 捕获 1 输出极性 (Compare 1 output polarity) 0: OC1 高电平有效; 1: OC1 低电平有效。
位 0	CC1E: 捕获 1 输出使能 (Compare 1 output enable) 0: 关闭 — OC1 禁止输出, 1: 开启 — OC1 信号输出到对应的输出引脚。

13.4.8 TIM14 计数器(TIM14_CNT)

偏移地址: 0x24

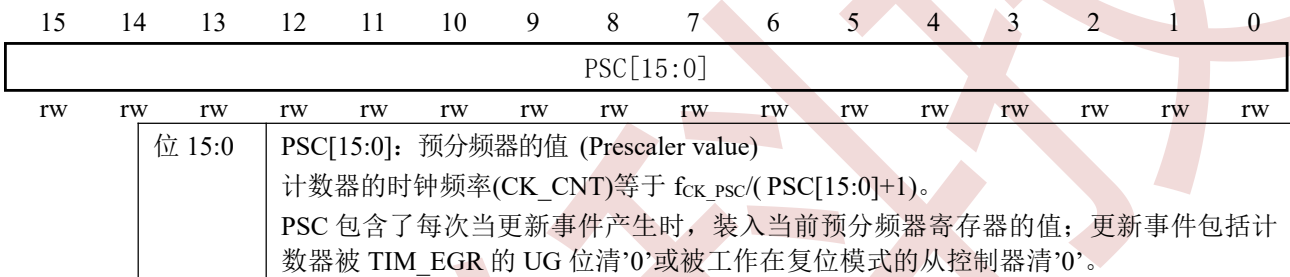
复位值: 0x0000



12.4.9 TIM14 预分频器(TIM14_PSC)

偏移地址: 0x28

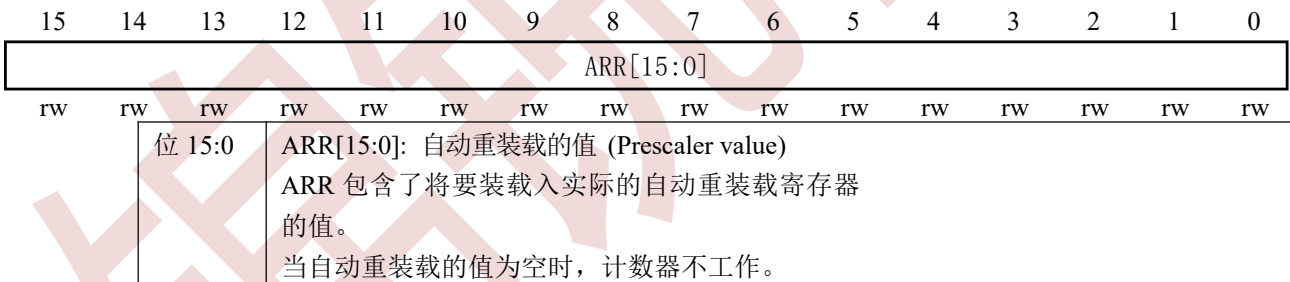
复位值: 0x0000



13.4.10 TIM14 自动重载寄存器(TIM4_ARR)

偏移地址:0x2C

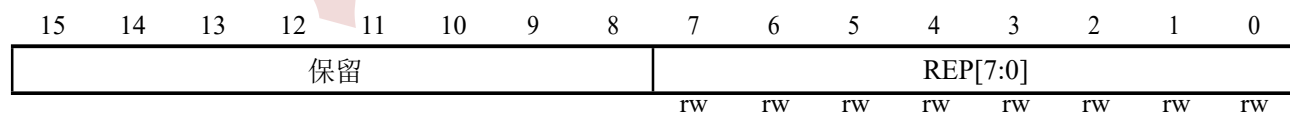
复位值:0x0000



12.4.11 TIM14 重复计数寄存器(TIM14_RCR)

偏移地址: 0x30

复位值: 0x0000



位 15:8	保留，始终读为 0。
位 7:0	<p>REP[7:0]: 重复计数器的值 (Repetition counter value)</p> <p>开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。</p> <p>每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。</p> <p>这意味着在 PWM 模式中，(REP+1)对应着：</p> <ol style="list-style-type: none"> 3. 在边沿对齐模式下，PWM 周期的数目； 4. 在中心对称模式下，PWM 半周期的数目；

13.4.12 TIM14 比较寄存器 1(TIM14_CCR1)

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>CCR1[15:0]: 比较通道 1 的值 (Compare 1 value)</p> <p>CCR1 包含了装入当前比较 1 寄存器的值(预装载值)。</p> <p>如果在 TIMx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前比较 1 寄存器中。</p> <p>当前比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。</p>														

13.4.13 TIM14 比较寄存器 2(TIM14_CCR2)

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>CCR2[15:0]: 比较通道 2 的值 (Compare 2 value)</p> <p>CCR2 包含了装入当前比较 2 寄存器的值(预装载值)。</p> <p>如果在 TIMx_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前比较 2 寄存器中。</p> <p>当前比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。</p>														

13.4.14 TIM14 DMA 控制寄存器(TIM14_DCR)

偏移地址: 0x48

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			DBL[4:0]					保留			DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13	保留, 始终读为 0。
位 12:8	<p>DBL[4:0]: DMA 连续传送长度 (DMA burst length)</p> <p>这些位定义了 DMA 在连续模式下的传送长度(当对 TIMx_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字(双字节)或字节:</p> <p>00000: 1 次传输 00001: 2 次传输</p> <p>00010: 3 次传输 </p> <p>..... 10001: 18 次传输</p> <p>例: 我们考虑这样的传输: DBL=7, DBA=TIM2_CR1</p> <p>如果 DBL=7, DBA=TIM2_CR1 表示待传输数据的地址, 那么传输的地址由下式给出: (TIMx_CR1 的地址) + DBA + (DMA 索引), 其中 DMA 索引 = DBL</p> <p>其中(TIMx_CR1 的地址) + DBA 再加上 7, 给出了将要写入或者读出数据的地址, 这样数据的传输将发生在从地址(TIMx_CR1 的地址) + DBA 开始的 7 个寄存器。</p> <p>根据 DMA 数据长度的设置, 可能发生以下情况:</p> <p>如果设置数据为半字(16 位), 那么数据就会传输给全部 7 个寄存器。</p> <p>如果设置数据为字节, 数据仍然会传输给全部 7 个寄存器: 第一个寄存器包含第一个 MSB 字节, 第二个寄存器包含第一个 LSB 字节, 以此类推。因此对于定时器, 用户必须指定由 DMA 传输的数据宽度。</p>
位 7:5	保留, 始终读为 0。
位 4:0	<p>DBA[4:0]: DMA 基地址 (DMA base address)</p> <p>这些位定义了 DMA 在连续模式下的基地址(当对 TIMx_DMAR 寄存器进行读或写时), DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量:</p> <p>00000: TIMx_CR1,</p> <p>00001: TIMx_CR2,</p> <p>00010: TIMx_SMCR,</p> <p>.....</p>

13.4.15 TIM14 连续模式的 DMA 地址(TIM14_DMAR)

偏移地址: 0x4C

复位值 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>DMAB[15:0]: DMA 连续传送寄存器 (DMA register for burst accesses)</p> <p>对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + DBA + DMA 索引, 其中:</p> <p>“TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1)所在的地 址; “DBA”是 TIMx_DCR 寄存器中定义的基地址; “DMA 索引”是由 DMA 自动控制的偏移量,它取决于 TIMx_DCR 寄存器中定义的 DBL。</p>														

14 基本定时器 (TIM6 和 TIM7)

14.1 基本定时器与通用定时器简介

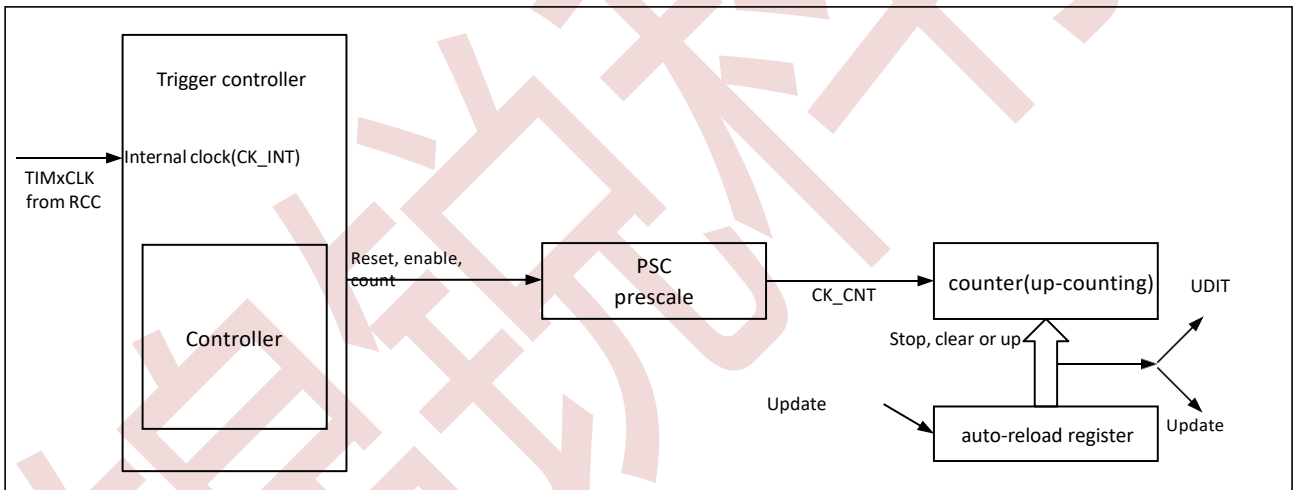
基本定时器 TIM6 和 TIM7 包含一个 16 位自动装载计数器。

14.2 基本定时器主要特性

- 16 位自动重载向上计数计数器。
- 16 位可编程预分频器。（分频系数可配置 1 到 65536 之间的任意值）
- 产生中断/DMA 的事件

◆ 更新事件

图11-1 TIM6 的框图



14.3 基础定时器描述

14.3.1 时基单元

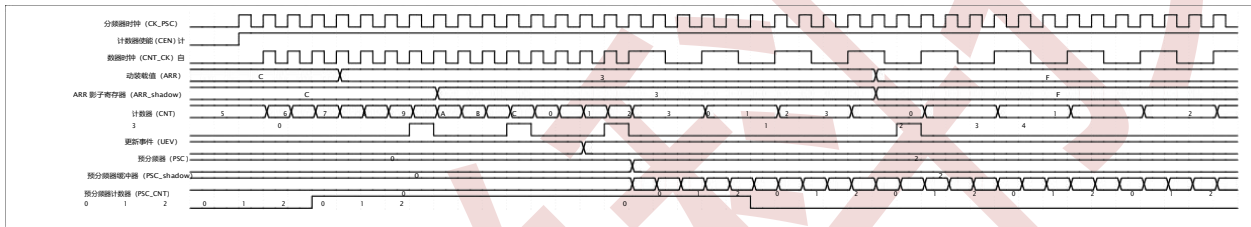
时基单元主要包括：预分频器、计数器、自动重载和重复计数器。计数器寄存器（TIMx_CNT）、预分频器寄存器（TIMx_PSC）和自动预装载寄存器（TIMx_ARR）。

计数单元由一个 16 位的计数器和对应的自动预装载寄存器组成，可以实现递增计数功能。

计数器的时钟由预分频器提供，预分频器由预分频计数器和对应的寄存器组成，分频系数为 1-65536，可以随时写入，在下次更新事件时生效。

自动预装载寄存器有预装载功能的 16 位影子寄存器，通过设置 TIMx_CR1 寄存器的 ARPE 位选择写入 ARR 寄存器的值立即生效或发生更新事件时载入影子寄存器。

图 11-2 自动预装载



14.3.2 计数模式

在 TIMx 仅支持递增计数模式。在使能 TIMx_CR1 寄存器的 CEN 后计数器由 0 开始递增计数，直至 TIMx_ARR 的值，产生一个计数器上溢事件（更新事件）并从 0 开始重新递增计数。设置 TIMx_EGR 寄存器的 UG=1，同样可以产生一个更新事件。

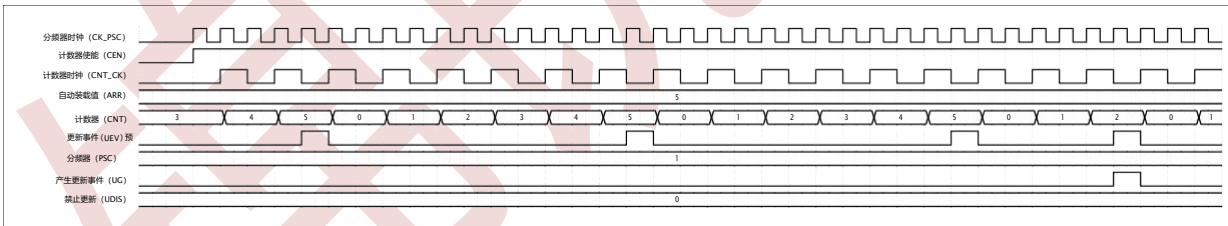


图 11-3 递增计数模式 (UDIS=0)

通过配置 TIMx_CR1 寄存器的 UDIS=1，可禁止产生更新事件，当计数器发生上溢事件时，不产生更新事件。此时若配置 UG=1，不产生更新事件，但是计数器和预分频器计数器会被初始化，从零开始递增计数。

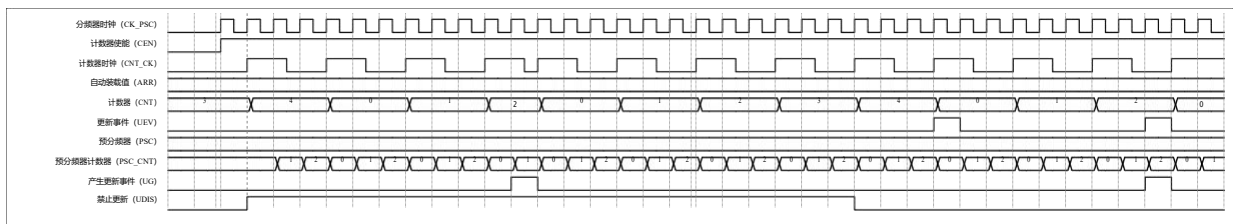


图 11-4 递增计数模式 (UDIS=1 禁止产生更新事件)

注：发生更新事件时

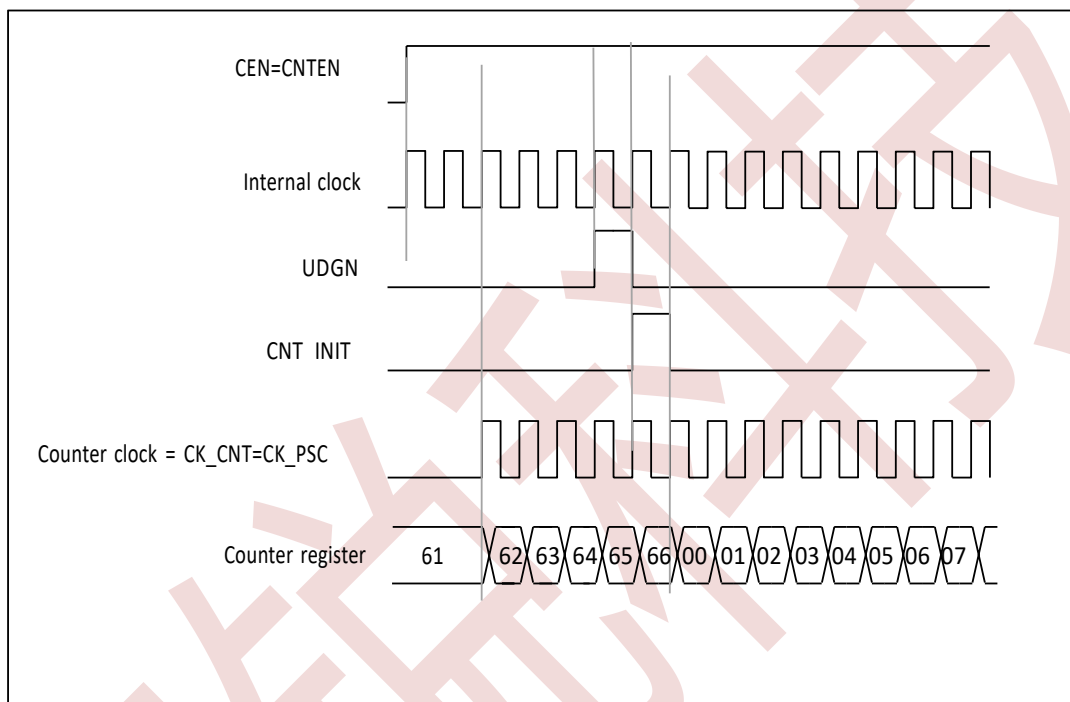
- ARR 寄存器中的值被载入 ARR 影子寄存器中。

14.3.3 时钟选择

- 定时器内部时钟：CK_INT

前提是 TIMx_CTRL1.CNTEN 位由软件写为'1'，预分频器的时钟源由内部时钟 CK_INT 提供。

正常模式下的控制电路，内部时钟分频系数为 1



14.4 TIMx 寄存器描述

有关寄存器中使用的缩写，请参阅第 1.1 节

这些外设寄存器可以作为半字（16 位）或一个字（32 位）操作。

14.4.1 寄存器概览

表 11-1 TIM6 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x0C	TIMx_DIER	中断使能寄存器	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000

14.4.2 控制寄存器 1(TIMx_CR1)

偏移地址: 0x00

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								ARPE	保留			OPM	URS	UDIS	CEN
res								rw	res			rw	rw	rw	rw

位 15:8	保留, 始终读为 0。
位 7	ARPE: 自动重载预装载使能 (Auto-reload preload enable)0: TIMx_ARR 寄存器没有缓冲 1: TIMx_ARR 寄存器具有缓冲
位 6:4	保留, 始终读为 0。
位 3	保留
位 2	URS: 更新请求源 (Update request source) 该位由软件设置和清除, 以选择 UEV 事件的请求源。 0: 如果使能了中断或 DMA, 以下任一事件可以产生一个更新中断或 DMA 请求: <ul style="list-style-type: none"> - 计数器上溢或下溢 - 设置 UG 位 - 通过从模式控制器产生的更新 1: 如果使能了中断或 DMA, 只有计数器上溢或下溢可以产生更新中断或 DMA 请求。
位 1	UDIS: 禁止更新 (Update disable) 该位由软件设置和清除, 以使能或禁止 UEV 事件的产生。 0: UEV 使能。更新事件(UEV)可以由下列事件产生: <ul style="list-style-type: none"> - 计数器上溢或下溢 - 设置 UG 位 - 通过从模式控制器产生的更新 产生更新事件后, 带缓冲的寄存器被加载为预加载数值。 1: 禁止 UEV。不产生更新事件(UEV), 影子寄存器保持它的内容(ARR、PSC)。但是如果设置了 UG 位或从模式控制器产生了一个硬件复位, 则计数器和预分频器将被重新初始化。
位 0	CEN: 计数器使能 (Counter enable)0: 关闭计数器 1: 使能计数器

14.4.3 DMA 中断使能寄存器(TIMx_DIER)

偏移地址: 0x0C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								UDE	保留						UIE
res								rw	res						rw
位 15:9		保留, 始终读为 0。													
位 8		UDE: 更新 DMA 请求使能 (Update DMA request enable)0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求													
位 7:1		保留, 始终读为 0。													
位 0		UIE: 更新中断使能 (Update interrupt enable)0: 禁止更新中断 1: 使能更新中断													

14.4.4 状态寄存器(TIMx_SR)

偏移地址: 0x10

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															UIF
res															rc w0
位 15:1		保留, 始终读为 0。													
位 0		UIF: 更新中断标志 (Update interrupt flag) 硬件在更新中断时设置该位, 它由软件清除。 0: 没有产生更新。 1: 产生了更新中断。下述情况下由硬件设置该位: - 计数器产生上溢或下溢并且 TIMx_CR1 中的 UDIS=0; - 如果 TIMx_CR1 中的 URS=0 并且 UDIS=0, 当使用 TIMx_EGR 寄存器的 UG 位重新初始化计数器 CNT 时。													

14.4.5 事件产生寄存器(TIMx_EGR)

偏移地址: 0x14

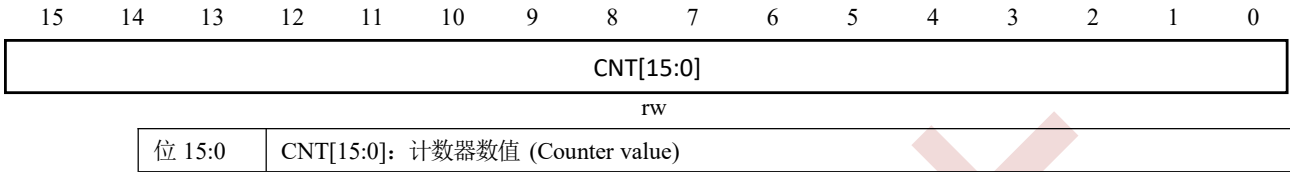
复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															UG
res															w
位 15:1		保留, 始终读为 0。													
位 0		UG: 产生更新事件 (Update generation)该 位由软件设置, 由硬件自动清除。 0: 无 作用 1: 重新初始化定时器的计数器并产生对寄存器的更新。注意: 预分频器也被清除(但预分频系数 不变)。													

14.4.6 计数器(TIMx_CNT)

偏移地址: 0x24

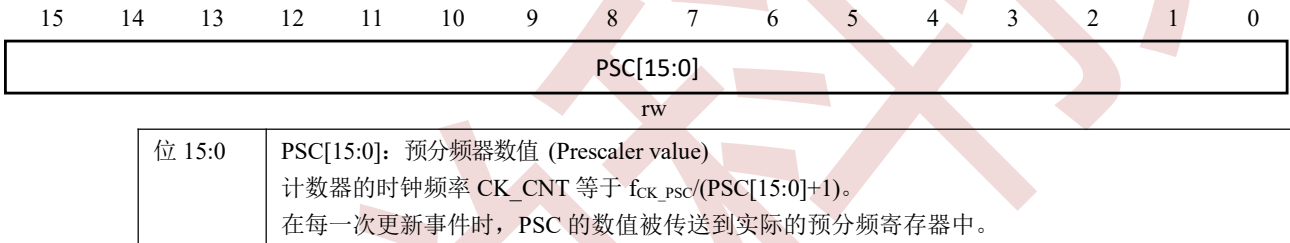
复位值: 0x0000



14.4.7 预分频器(TIMx_PSC)

偏移地址: 0x28

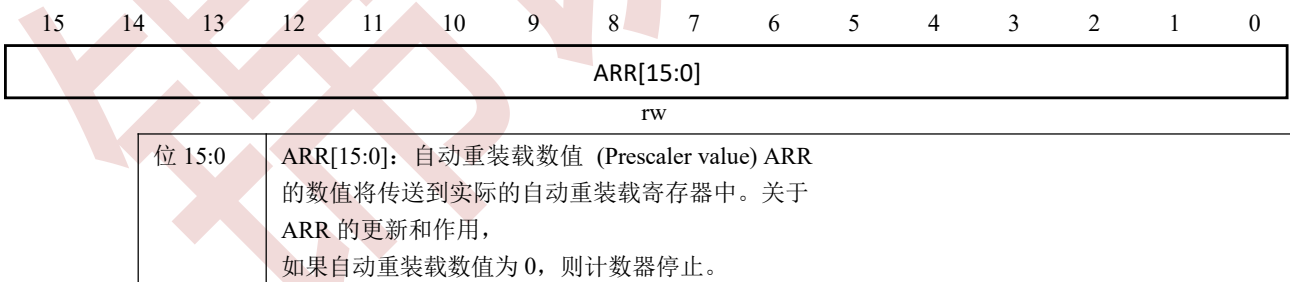
复位值: 0x0000



14.4.8 自动重装载寄存器(TIMx_ARR)

偏移地址: 0x2C

复位值: 0x0000



15 独立看门狗定时器（IWDG）

15.1 简介

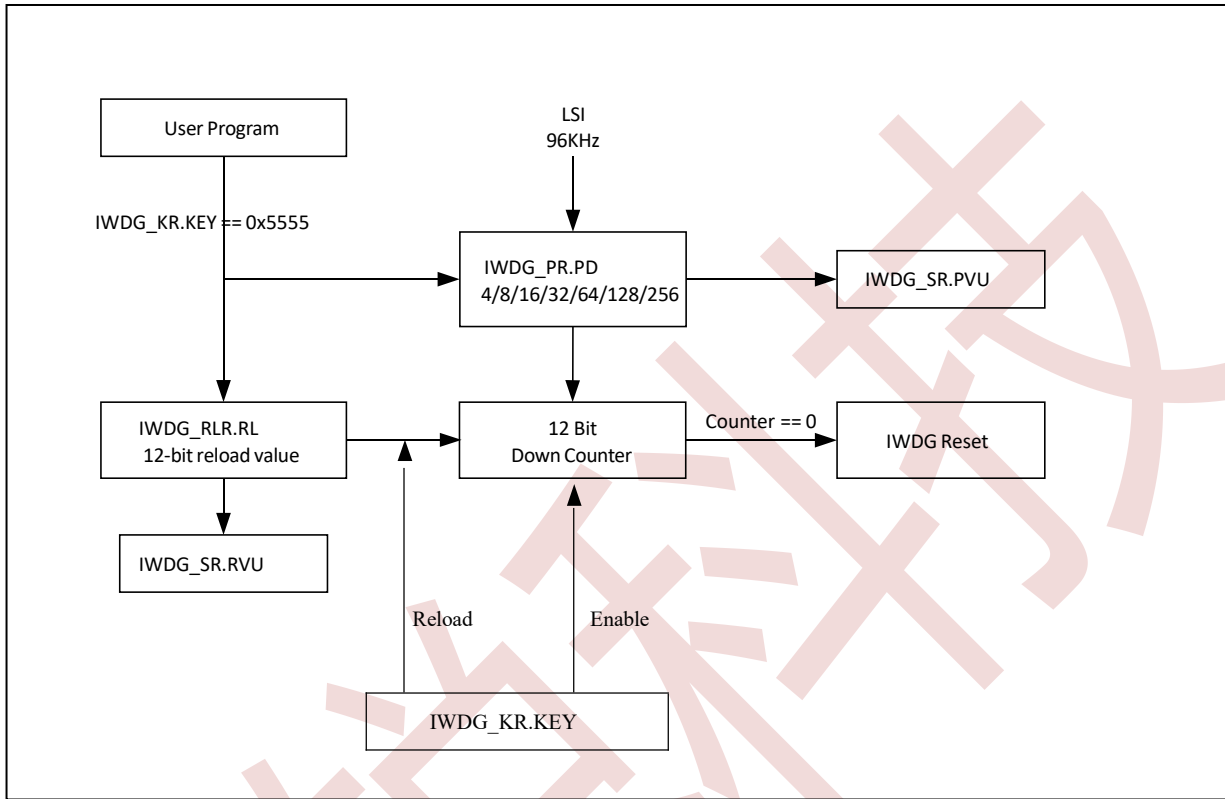
CA1020 内置独立看门狗（IWDG）和窗口看门狗（WWDG）定时器，解决软件错误导致的问题。看门狗定时器使用非常灵活，提高了系统的安全性和定时控制的准确性。

独立看门狗（IWDG）由运行在 96KHz 的低速内部时钟（LSI 时钟）驱动，在死循环事件或 MCU 卡死发生时，它仍然可以运行。这可以提供更高的安全级别、定时精度和看门狗的灵活性。它可以通过重置来解决由于软件故障引起的系统故障。IWDG 最适合需要看门狗在主应用程序之外作为完全独立进程运行但时序精度限制较低的应用程序。

15.2 IWDG 主要特性

- 独立运行的 12 位递减计数器
- 时钟由 RC 振荡器提供，在 STOP 模式下仍能正常工作
- 可以匹配复位和低功耗唤醒
- 看门狗被激活后，当递减计数器达到 0x000 时产生复位

15.3 功能描述



密钥寄存器 (IWDG_KR) 中写入 0xCCCC, 开始启用独立看门狗; 此时计数器开始从其复位值 0xFFFF 递减计数。当计数器计数到末尾 0x000 时, 会产生一个复位信号 (IWDG_RESET)。

无论何时, 只要密钥寄存器 IWDG_KR 中写入 0xAAAA, IWDG_RLR 中的值就会被重新加载到计数器, 从而避免产生看门狗复位。

如果通过选项字节使能“硬件看门狗定时器”功能, 则看门狗将在系统上电后自动开始运行并产生系统复位, 除非软件在计数器到达‘0’之前重新加载计数器。

15.3.1 寄存器访问保护

IWDG_PR 和 IWDG_RLR 寄存器具有写保护功能。在修改这两个寄存器数据之前, 必须先配置 IWDG_KR 寄存器为 0x5555。配置成其他任何数据, 都将再次启动寄存器写保护。IWDG_SR.PVU 指示预分频器值更新是否正在进行。IWDG_SR.RVU 指示 IWDG 是否正在更新重载值。当预分频器值和/或重载值更新时, 硬件设置 IWDG_SR.PVU 位和/或 IWDG_SR.RVU 位。预分频器值和/或重载值更新完成后, 硬件清除 IWDG_SR.PVU 位和/或 IWDG_SR.RVU 位。

重装载操作 (IWDG_KR 配置 0xAAAA) 也会启动写保护功能。

15.4 用户界面

IWDG 模块用户界面包含 4 个寄存器：密钥寄存器（IWDG_KR）、预分频寄存器（IWDG_PR）、重装载寄存器（IWDG_RLR）和状态寄存器（IWDG_SR）。

15.4.1 操作流程

当IWDG 从软件（将0xAAAA 写入IWDG_KR.KEYV[15:0]位）或硬件（清零FLASH_OB.WDG_SW 位）复位启用时。它从 0xFFF 开始递减计数。向下计数间隙由预分频 LSI 时钟确定。重新加载计数器后，新一轮递减计数器的值将从 IWDG_RLR.REL[11:0]中的值开始，而不是 0xFFF。

程序正常运行时，软件需要在计数器到达 0 前喂狗，开始新一轮的递减计数。当计数器达到 0 时，表示程序故障。IWDG 在这种情况下产生复位信号。

如果用户想要配置 IWDG 预分频和重装载值寄存器，需要先将 0x5555 写入 IWDG_KR.KEY[15:0]。然后确认 IWDG_SR.RVU 位和 IWDG_SR.PVU 位。IWDG_SR.RVU 位指示重装载值更新正在进行，IWDG_SR.PVU 表示预分频值更新正在进行。只有当这两位为 0 时，用户才能更新相应的值。当更新正在进行时，硬件将相应位设置为 1。此时，读取 IWDG_PR.PD[2:0]或IWDG_RLR.RL[11:0]无效，因为数据需要同步到 LSI 时钟域。从 IWDG_PR.PD[2:0]或 IWDG_RLR.RL[11:0]读取的值将在硬件清除 IWDG_SR.PVU 位或 IWDG_SR.RVU 位后才有效。

如果应用程序使用多个重装载值或预分频值，则必须等到 IWDG_SR.RVU 位复位后才能更改重装载值，IWDG_SR.PVU 位复位后才能更改预分频值。但是，在更新预分频值和重装载值后，或只更新预分频值后，或只更新重装载值后，无需等到 IWDG_SR.RVU 位或 IWDG_SR.PVU 位复位后才能继续执行代码（即使在进入低功耗模式的情况下，写入操作也会被考虑并完成）。

看门狗超时时间（96kHz 的输入时钟（LSI），如表13-1。

表13-1 IWDG 计数最大和最小复位时间

预分频系数	PD [2:0]	最短时间 (ms) RL [11:0]=0x001	最长时间 (ms) RL [11:0]=0xFFFF
/4	000	0.04166	170.625
/8	001	0.0833	341.25
/16	010	0.1666	682.5
/32	011	0.333	1365
/64	100	0.666	2730
/128	101	1.333	5460
/256	11x	2.666	10920

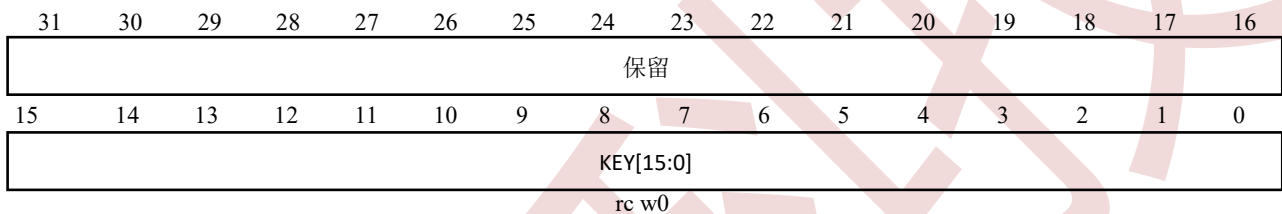
15.5 IWDG 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	IWDG_KR	密钥寄存器	0x00000000
0x04	IWDG_PR	预分频寄存器	0x00000000
0x08	IWDG_RLR	重装载寄存器	0x00000FFF
0x0C	IWDG_SR	状态寄存器	0x00000000

15.5.1 IWDG 密钥寄存器 (IWDG_KR)

偏移地址: 0x00

复位值: 0x00000000



位域	名称	描述
31:16	保留	保留, 必须保持复位值。
15:0	KEY[15:0]	密钥寄存器: 只有特定的值才能发挥特定的作用 0xCCCC: 启动看门狗计数器, 如果硬件看门狗使能则无效, (如果选择了硬件看门狗, 则不受该命令字限制) 0xAAAA: 用 IWDG_RLR 寄存器中的 RL 值重新加载计数器以防止复位 0x5555: 禁用 IWDG_PR 和 IWDG_RLR 寄存器的写保护

15.5.2 IWDG 预分频寄存器 (IWDG_PR)

偏移地址: 0x04

复位值: 0x00000000

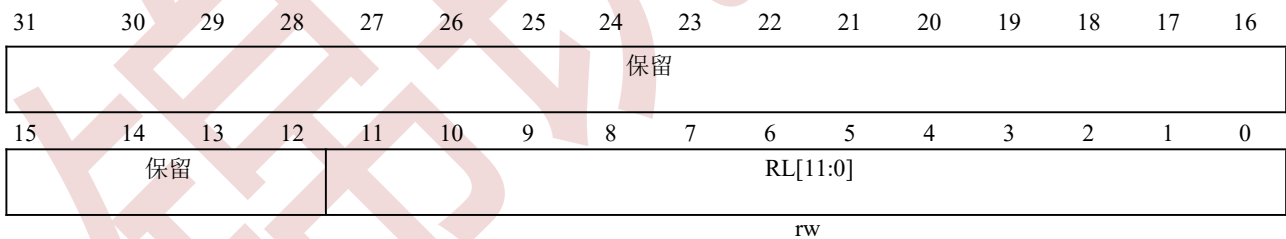


位域	名称	描述
31:3	保留	保留，必须保持复位值。
2:0	PD[2:0]	预分频因子 当IWDG_KR.KEY[15:0]不是0x5555 时具有写访问保护。IWDG_SR.PVU 位必须为0，否则PD[2:0]值无法更改。分频系数如下： 000：预分频因子=4 001：预分频因子=8 010：预分频因子=16 011：预分频因子=32 100：预分频因子=64 101：预分频因子=128 其他：预分频因子=256 注意：读取该寄存器将返回来自 VDD 电压域的预分频值。如果正在进行写操作，则回读值可能无效。因此，读取值仅在 IWDG_SR.PVU 位为0 时有效。

15.5.3 IWDG 重装载寄存器 (IWDG_RLR)

偏移地址：0x08

复位值：0x0000FFF



位域	名称	描述
31:12	保留	保留，必须保持复位值。
11:0	RL[11:0]	看门狗计数器重装载值。 带写保护。定义看门狗计数器的重装载值，每次将 0xAAAA 写入 IWDG_KR.KEY[15:0]位时将其加载到计数器。然后计数器从该值开始倒计时。看门狗超时周期可以根据这个重装载值和时钟预分频值计算，参考表13-1。 该寄存器只能在 IWDG_SR.RVU 位为0 时修改。 注意：读取该寄存器将返回来自 VDD 电压域的重装载值。如果正在进行写操作，则回读值可能无效。因此，读取值仅在 IWDG_SR.RVU 位为0 时有效。

15.5.4 IWDG 状态寄存器 (IWDG_SR)

偏移地址: 0x0C

复位值: 0x00000000



位域	名称	描述
31:2	保留	保留，必须保持复位值。
1	RVU	看门狗重装载值更新 重装载值更新：该位表示正在更新重装载值。硬件置位，硬件清零。软件只能在 IWDG_KR.KEY[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_RLR.RL[11:0]的值。
0	PVU	看门狗预分频值更新 预分频值更新：该位表示正在更新预分频值。硬件置位，硬件清零。软件只能在 IWDG_KR.KEY[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_PR.PD[2:0]的值。

16 窗口看门狗 (WWDG)

16.1 简介

窗口看门狗 (WWDG) 的时钟是由 APB1 时钟频率除以 1024 得到的, 通过时间窗口的配置来检测程序运行是否异常。因此, WWDG 适用于精确定时, 常用于监控因外部干扰或无法预见的逻辑条件导致应用程序偏离其正常操作顺序的软件故障。当 WWDG 递减计数器在达到窗口寄存器值之前或 WWDG_CR.T6 位变为 0 之后刷新时, 系统复位发生。

16.2 主要特征

- 7 位独立递减计数器可编程
- WWDG 启用后, 在以下情况下会发生复位
 - ◆ 递减计数器的值小于 0x40

16.3 功能描述

如果看门狗被启动 (WWDG_CR 寄存器中的 WDGA 位被置'1'), 并且当 7 位 (T[6:0]) 递减计数器从 0x40 翻转到 0x3F(T6 位清零) 时, 则产生一个复位。

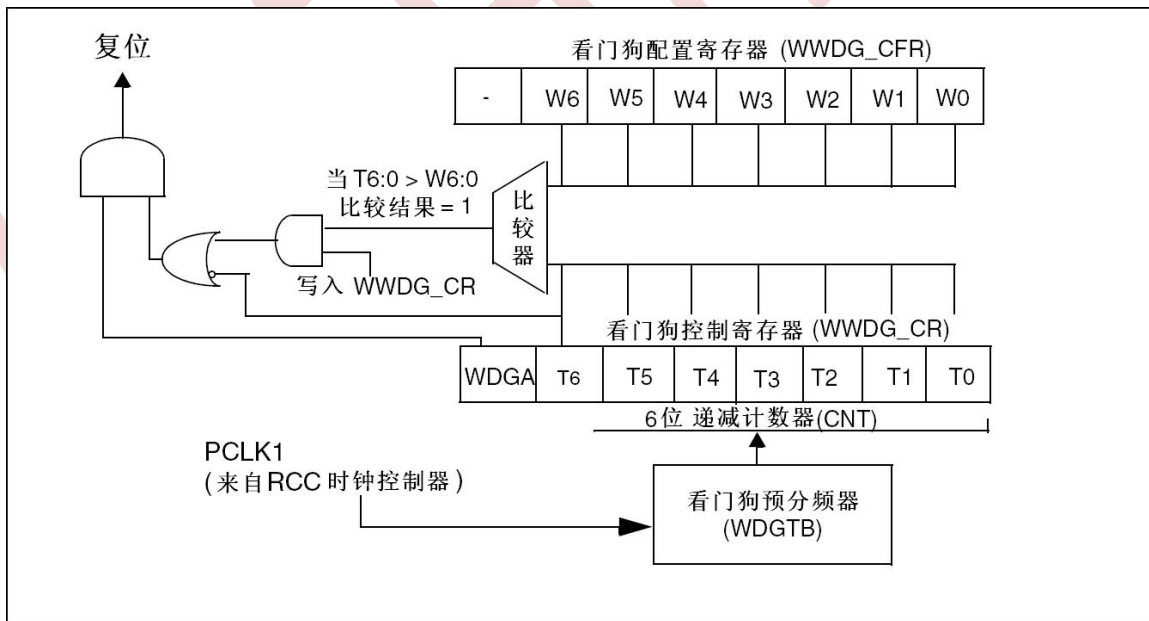


图14-1 窗口看门狗功能框图

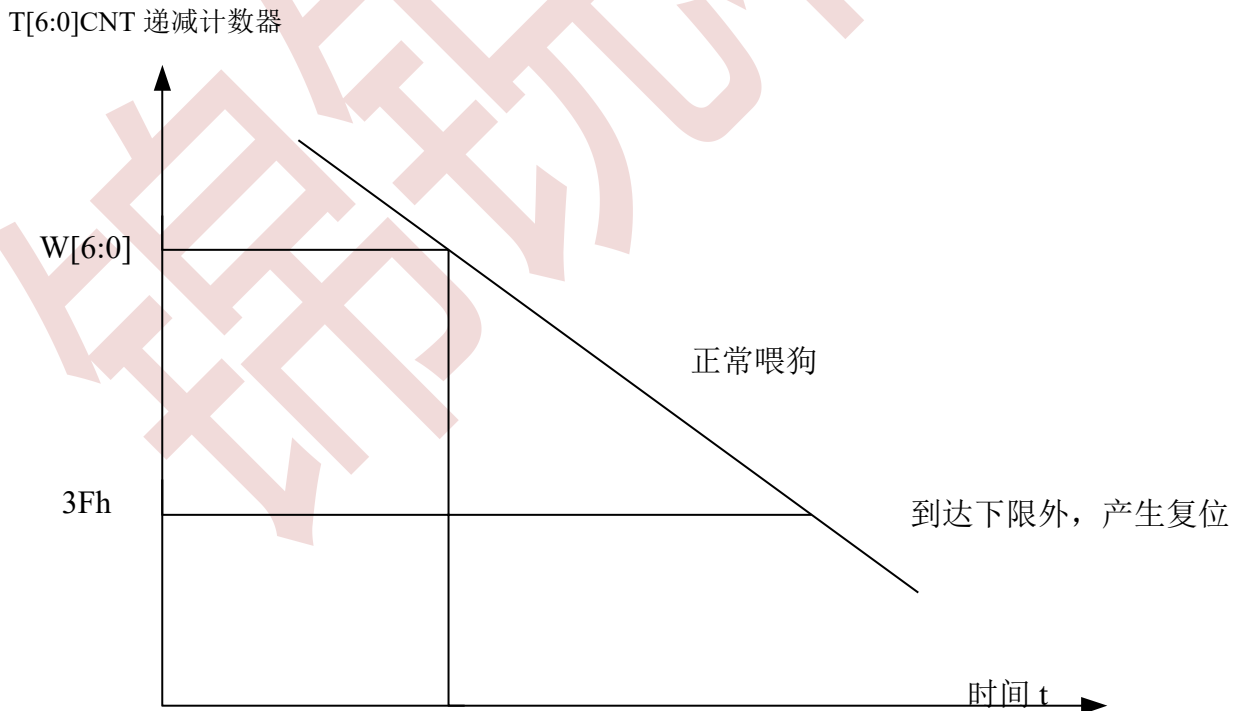
应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值小于窗口寄存器的值时, 才能进行写操作。储存在 WWDG_CR 寄存器中的数值必须在 0xFF 和 0xC0 之间:

- 启动看门狗
 - ◆ 在系统复位后，看门狗总是处于关闭状态，设置 WWDG_CR 寄存器的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。
- 控制递减计数器
 - ◆ 递减计数器处于自由运行状态，即使看门狗被禁止，递减计数器仍继续递减计数。当看门狗被启用时，T6 位必须被设置，以防止立即产生一个复位。
 - ◆ T[5:0]位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的。
 - ◆ 配置寄存器（WWDG_CFR）中包含窗口的上限值：要避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重新装载，下图描述了窗口寄存器的工作过程。
 - ◆ 另一个重装计数器的方法是利用早期唤醒中断（EWI）。设置 WWDG_CFR 寄存器中的 WEI 位开启该中断。当递减计数器到达 0x40 时，则产生此中断，相应的中断服务程序（ISR）可以用来加载计数器以防止 WWDG 复位。在 WWDG_SR 寄存器中写'0'可以清除该中断。

注： 可以用 T6 位产生一个软件复位（设置 WDGA 位为'1'，T6 位为'0'）。

16.4 刷新看门狗和中断产生的时序

图14-2 WWDG 超时时间坐标图



看门狗刷新窗口在 WWDG_CFR.W[6:0]值（最大值 0x7F）和 0x3F 之间。计数器使用分频后的 APB1 时钟从 0x7F 向下计数到 0x3F，最大计数时间和最小计数时间如表 14-1 所示

（假设 APB 时钟为 64MHz），计算公式为：

$$T_{WWDG} = T_{PCLK1} \times 1024 \times 2^{WWDGTB} \times (T[5:0] + 1)$$

其中：

T_{WWDG} :WWDG 超时

T_{PCLK1} :APB1 时钟间隔，单位为：ms

PCLK1=64MHz 时的最小-最大超时时长

表14-1 WWDG 的最大和最小计数时间

WWDGTB	最小超时 (μs) T[5:0] = 0x00	最大超时值 (ms) T[5:0] = 0x3F
0	16	1.024
1	32	2.048
2	64	4.096
3	128	8.192

16.5 用户配置

16.5.1 WWDG 配置流程

- 1.使能 WWDG 模块的时钟
- 2.软件设置 WWDG_CFR.WDGTB[8:7]位来配置 WWDG 的预分频因子
- 3.软件配置 WWDG_CR.T[6:0]位，设置计数器的起始值。需要将 WWDG_CR.T[6]位设置为 1，以防止在启用后立即复位
- 4.配置 WWDG_CFR.W[6:0]位配置上边界窗口值
- 5.设置 WWDG_CR.WDGA[7]位使能 WWDG
- 6.软件操作 WWDG_SR.EWIF[0]位清除唤醒中断标志
- 7.配置 WWDG_CFR.EWI[9]位使能提前唤醒中断

16.6 WWDG 寄存器

16.6.1 WWDG 寄存器总览

表 14-2 WWDG 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	WWDG_CR	控制寄存器	0x0000007F
0x04	WWDG_CFG	配置寄存器	0x0000007F
0x08	WWDG_SR	状态寄存器	0x00000000

16.6.2 控制寄存器 (WWDG_CR)

偏移地址: 0x0C

复位值: 0x0000007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								WDGA	T6	T5	T4	T3	T2	T1	T0
								rs	rw	rw	rw	rw	rw	rw	rw

位 31:8	保留。
位 7	WDGA: 激活位 (Activation bit) 此位由软件置'1', 但仅能由硬件在复位后清'0'。当 WDGA=1 时, 看门狗可以产生复位。 0: 禁止看门狗 1: 启用看门狗
位 6:0	T[6:0]: 7 位计数器 (MSB 至 LSB) (7-bit counter) 这些位用来存储看门狗的计数器值。每 (1024x2 ^{WDG TB}) 个 PCLK1 周期减 1。当计数器值从 40h 变为 3Fh 时 (T6 变成 0), 产生看门狗复位。

16.6.3 配置寄存器 (WWDG_CFR)

地址偏移量: 0x04

复位值: 0x7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						EWI	WDGTB	W6	W5	W4	W3	W2	W1	W0	
						rs	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10	保留。
位 9	EWI: 提前唤醒中断 (Early wakeup interrupt) 此位若置'1', 则当计数器值达到 40h, 即产生中断。 此中断只能由硬件在复位后清除。
位 8:7	WDGTB[1:0]: 时基 (Timer base) 预分频器的时基可以设置如下: 00: CK 计时器时钟 (PCLK1 除以 1024) 除以 1 01: CK 计时器时钟 (PCLK1 除以 1024) 除以 2 10: CK 计时器时钟 (PCLK1 除以 1024) 除以 4 11: CK 计时器时钟 (PCLK1 除以 1024) 除以 8
位 6:0	W[6:0]: 7 位窗口值 (7-bit window value) 这些位包含了用来与递减计数器进行比较用的窗口值。

16.6.4 状态寄存器 (WWDG_SR)

地址偏移量: 0x08

复位值: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															EWIF

rc w0	
位 31:1	保留。
位 0	EWIF: 提前唤醒中断标志 (Early wakeup interrupt flag) 当计数器值达到 40h 时, 此位由硬件置'1'。它必须通过软件写'0'来清除。对此位写'1'无效。若中断未被使能, 此位也会被置'1'。

17 通用串行接口 (UART0/UART1/UART2)

芯片内置 3 组通用串口，UART0、UART1 和 UART2 是设计完全相同的两个全双工异步串行数据收发器，UARTx (x=0、1、2,代指 UART0、UART1、UART2) 支持一个字节的接收缓存。

17.1 UART 寄存器概览

表 21-7 USART 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	UART_CR	USART 控制寄存器	0x000000C0
0x04	UART_S1REL	USART 波特比率寄存器	0x00000000
0x08	UART_BUF	USART 数据寄存器	0x00000000

17.2 状态寄存器 (UART_CR)

地址偏移: 0x00

复位值: 0x00C0

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

保留															
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

DMA TXEN	DMAR XEN	DAMP LS	DNUM					SM1	UIIE	SM21	REN1	TB81	RB81	TI1	RI1
			rc w0					r	rc w0	rc w0	r	r	r	r	r

位 15	DMA 发送 TX 数据使能
位 14	DMA 接收 RX 数据使能
位 13	第一个发送数据 DMA 请求脉冲，写 1 产生 dma 发送请求脉冲，硬件自动清零，不通过 dma 传输数据，该位不需要设置
位 12:8	波特率配置寄存器， 发送时，须满足 $DNUMx \geq 0$ ；接收时， $DNUMx \geq 6$ $BRx = F_{sys} * (1 / (DNUMx + 1 * 1024 - SxREL))$
位 7	UART1 模式选择位
位 6	UART1 中断使能位，1 有效
位 5	多机通信使能位，1 有效
位 4	串行接收使能位，1 有效
位 3	发送数据的第 9 位 在模式 A，这个位用于 UART1 传送数据，对应传送数据的第 9 位 (例如奇偶校验或多主机通信)，由软件控制
位 2	接收数据的第 9 位 在模式 A，这个位用于 UART1 接收数据，对应接收数据的第 9 位； 在模式 B，这个位是接收到的停止位
位 1	传送中断标志位，1 有效，写 1 清 0
位 0	接收中断标志位，1 有效，写 1 清 0

17.3 波特率配置寄存器 (UART_S1REL)

地址偏移: 0x04

复位值: 不确定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								S1REL							
		rw		rw		rw		rw		rw		rw		rw	
位 31:9		保留位, 硬件强制为 0													
位 9:0		波特率配置寄存器													

17.4 发送接收缓冲器 (UART_BUF)

地址偏移: 0x08

复位值: 0x0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								S0BUF							
rw		rw		rw		rw		rw		rw		rw		rw	
位 31:8		保留位, 硬件强制为 0													
位 7:0		发送接收缓存													

18 I²C 接口

18.1 简介

I²C（芯片间）总线接口连接微控制器和串行 I²C 总线。它提供多主机功能，控制所有 I²C 总线特定的时序、协议、仲裁和定时。支持标准（高达 100kHz）和快速（高达 400kHz）两种模式。

18.2 I²C 寄存器描述

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

18.2.1 I²C 寄存器概览

表 20-3 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	I2C_CCR	控制寄存器	0x0000
0x04	I2C_DAT	数据寄存器	0x0000
0x08	I2C_ADR	地址寄存器	0x0000
0x0C	I2C_FLG	状态寄存器	0x0000

18.2.2 控制寄存器 1(I2C_CCR)

地址偏移：0x00复

位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR								CE	IE	STA	STP	SHD	AAK	CBSE	STFE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

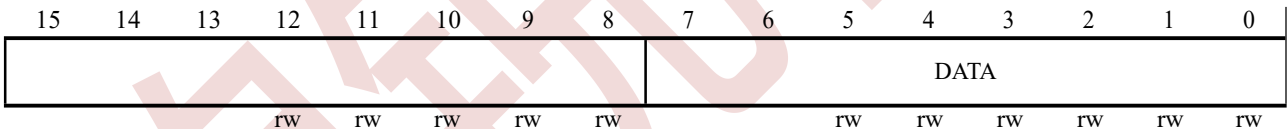
位 15:8	<p>I2C 时钟配置寄存器</p> <p>采样频率为 I2C 工作时钟的 2I2CCR[15:13]分频，当 I2CCR[15:13]等于</p> <p>000:Fsample=Fi2cclk</p> <p>001:Fsample=Fi2cclk/2</p> <p>010:Fsample=Fi2cclk/4</p> <p>...</p> <p>111:Fsample=Fi2cclk/128</p> <p>输出频率为采样频率的 (I2CCR[12:8]+1) 分频，</p> <p>Fscl=Fi2cclk/(2I2CCR[15:13]*I2CCR[12:8]+1)</p> <p>例如 I2CCR[12:8]=9 时，当 I2CCR[15:13]等于</p> <p>000:Fscl=Fi2cclk/(1*10)</p> <p>001:Fscl=Fi2cclk/(2*10)</p> <p>010:Fscl=Fi2cclk/(4*10)</p> <p>...</p> <p>111:Fscl=Fi2cclk/(128*10)</p>
--------	---

	备注： 1. 当 I2CCCR[15:13] = 0 时，如果对 I2CCCR[12:8] 写小于 9 的值，将自动按 9 的值计算。 2. 当 I2CCCR[15:13] > 0 时，如果对 I2CCCR[12:8] 写小于 7 的值，将自动按 7 的值计算
位 7	I2C 模块使能位，1 有效
位 6	I2C 中断使能位，1 有效
位 5	I2C 发送 START 信号控制位，1 有效， 检测到 START 信号后将自动清 0
位 4	I2C 发送 STOP 信号控制位，1 有效， 检测到 STOP 信号后将自动清 0
位 3	为 1 时，如果 I2CF 为 1，那么当 SCL 变低之后，I2CF 将会使 SCL 保持在低的状态
位 2	I2C 发送 ACK 信号控制位，1 有效 备注： 当 I2C 接口配置为从机模式时，这一位须预先置 1，否则即使地址匹配也不会回复 ACK，从而无法被寻址。
位 1	CBUS 兼容使能位 当这一位设置为 1 时，将会使传输忽略 ACK 位的状态判断，以兼容 CBUS 总线。
位 0	为 1 时，I2C 模块检测到 START 信号时将置位 I2CF

18.2.3 控制寄存器 2(I2C_DAT)

地址偏移：0x04 复

位值：0x0000

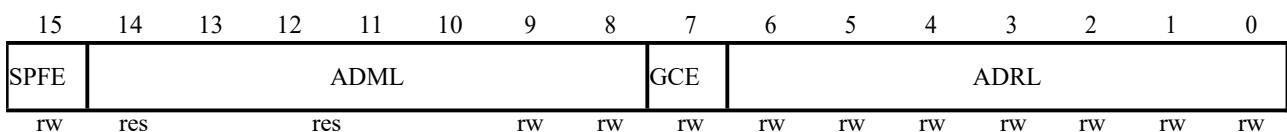


位 7:0	发送和接收数据缓存 备注： 当 I2CF 为 1 时，建议改写/读取 I2CDAT 时，让 I2CF 保持在 1，等处理完成之后再清除 I2CF，以继续传输，这样可以避免总线发生不必要的错误。
-------	--

18.2.4 I2C 地址寄存器 (I2C_ADR)

地址偏移：0x08

复位值：0x0000



位 15	为 1 时，I2C 模块检测到 STOP 信号时将置位 I2CF
------	----------------------------------

位 14:8	I2C 从地址按位屏蔽寄存器，为从机时有效 当 I2CADM[n](n=0~6)=1 时，对应的地址位 I2CADR[n] 将不比对（即认为无论收到 1 还是 0 都算匹配）
位 7	识别广播地址（00H）使能位，1 有效
位 6:0	I2C 从机地址，作为从机时有效 备注： （在 AAK 为 1 的前提下）7 位地址模式时，接收的第一个地址字节高 7 位和 I2CADR 匹配，则回复 ACK，进入从机模式。

18.2.5 I2C 状态寄存器（I2C_FLG）

地址偏移：0x0C

复位值：0x0000

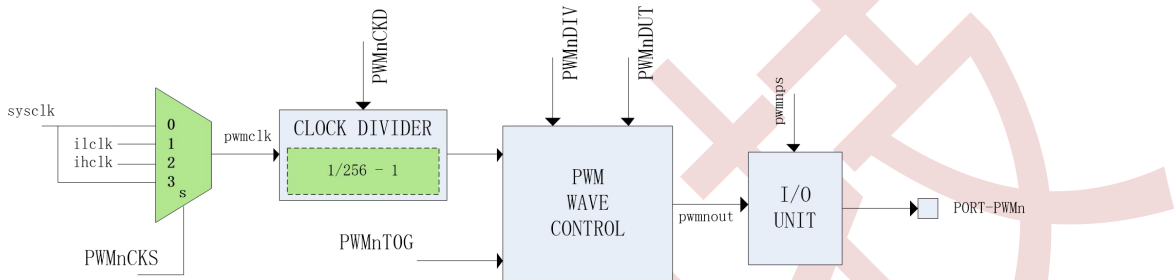
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							FLG	STA							
rw	res	res	res	res	res	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位[15:9]		保留													
位 8		I2C 中断标志，1 有效，写 1 清 0 备注： 1. 每字节地址或数据传输完成后（收到/发送完 ACK/NAK），将置位 I2CF。 2. 总线出错时，将置位 I2CF。 3. 当 STFE=0 时，检测到 START 信号，I2CF 不会置 1。 4. 当 SPFE=0 时，检测到 STOP 信号，I2CF 不会置 1。													
位 7:0		I2C 状态寄存器 00H:（主/从）总线错误 08H:（主/从）检测到 START 信号（只在 STFE=1 时才有效） 18H:（主）已发送地址+写位，已接收到应答信号 20H:（主）已发送地址+写位，无接收到应答信号 28H:（主）已发送/接收一字节数据，已检测到应答信号 30H:（主）已发送/接收一字节数据，无检测到应答信号 38H:（主）失去仲裁（主机失去仲裁后会变为从机） 40H:（主）已发送地址+读位，已接收到应答信号 48H:（主）已发送地址+读位，无接收到应答信号 60H:（从）已接收地址+写位，已发送出应答信号 70H:（主/从）已接收广播地址，已发送出应答信号（主机或从机都会变为从机） 80H:（从）已发送/接收一字节数据，已检测到应答信号 88H:（从）已发送/接收一字节数据，无检测到应答信号 A0H:（主/从）检测到 STOP 信号（只在 SPFE=1 时才有效） A8H:（从）已接收地址+读位，已发送出应答信号 F8H:（主/从）总线空闲													

19 LED 级联控制

19.1 LED 级联简介

CA1020 系列芯片最多有 1 通道 PWM 输出, PWM 周期和占空比可在 16 位范围内任意配置。扫描频率大于 400Hz/S, 数据发送速度 800Kbps, 直接控制 WS2812 或类似的驱动芯片, 符合单色或七彩 LED 灯带产品的需求。

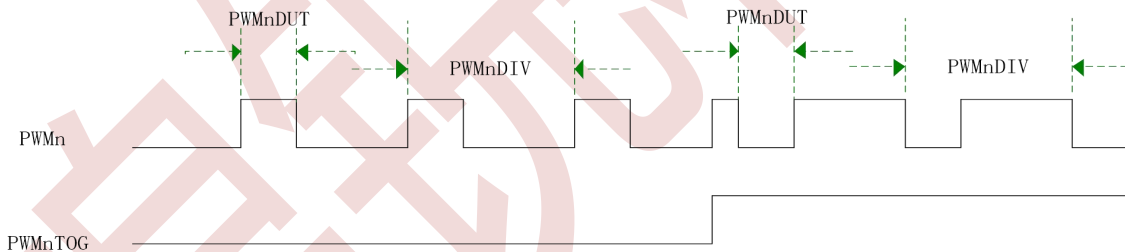
19.1.1 PWMLLED 输出原理图



19.1.2 PWM 输出波形

PWM 使能后, PWM 计数器开始累加计数, 当计数值不大于 PWMnDUT 时, PWM 引脚输出高电平 (PWMnTOG=0), 当计数值大于 PWMnDUT 时, PWM 引脚输出低电平 (PWMnTOG=1)。当计数值与 PWMnDIV 相等时, 一个 PWM 周期完成, PWM 计数器重置并开始下一周期计数, 此时将产生 PWM 中断。

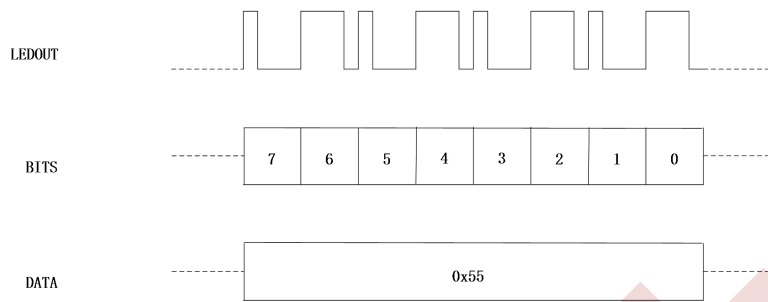
当 PWM 波形满足条件 $PWMnDIV > PWMnDUT > 0$ 时, PWM 波形如图所示。



PWM 中断通过寄存器 PWMCON 的 PWMIE 位使能, PWMIE 位对应的是 PWM 计数器计数到顶点 (即等于 PWMnDIV) 产生的中断。

18.1.3 单线级联 LED 驱动

通道支持单线级联 LED 驱动, 级联 LED 的典型驱动时序图。



级联 LED 时序图

位码示意图如图 18-2-4 所示。



位码示意图

在级联 LED 时序图中, 位码 0 的高电平时间宽度由 PWMnDUT (n=1/2) 配置, 位码 1 的高电平时间宽度由 LEDUT 配置, 而位周期时间由 PWMnDIV 配置。当 PWMnMOD 不为 0 时, 级联 LED 驱动使能, LEDAT0/LEDAT1 分别为 LEDn (n=0/1) 的数据寄存器, 当 LEFn (n=0/1) 为 0 时, 可以向 LEDAT0/LEDAT1 写入 LED 数据。写入 LEDAT0/LEDAT1 即启动 LED 驱动数据发送, 当 LEDn (n=0/1) 发送器正处于发送状态时, LEBSYn (n=0/1) 置 1, 当发送器处于空闲状态时, LEBSYn 变为 0。LED 发送器有一字节的发送缓存, 当数据寄存器和缓存寄存器都有数据时, LEFn (n=0) 位置 1, 当缓存寄存器的数据发送完后, 会自动从数据寄存器中加载, 同时 LEFn (n=0) 位置 0, LEFn (n=0)=0 表示可以重新向 LEDAT0/LEDAT1 装载数据。当 PWMnMOD 不为 0 时, PWMnMOD 也同时表示发送完 PWMnMOD 个字节后插入等待时间, 等待时间由 LEWTM 来配置。

当 PWMnPOL=1 (n=1/2) 时, LEDAT0/LEDAT1 的数据反相, 即: 例如写入 01010101B, 实际发送出来的是 10101010B。

19.2 PWMLLED 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	PWMCON	GPIOA0 配置寄存器	0x7F1B
0x04	LEDAT	GPIOA1 配置寄存器	0x7F1B
0x08	PWMDIV	GPIOA2 配置寄存器	0x7F18
0x0C	PWMDUT	GPIOA3 配置寄存器	0x7F18
0x10	LEDWTM	GPIOA4 配置寄存器	0x7F1D
0x14	LEDUT	GPIOA5 配置寄存器	0x7F18

19.2.1 PWMCON

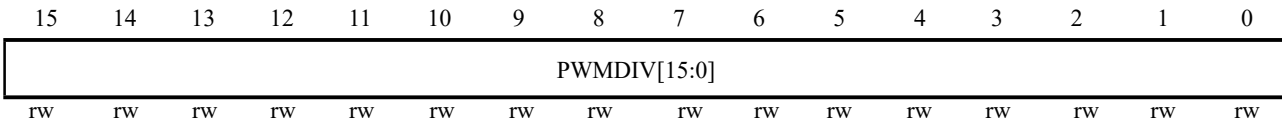
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWMCKD								PWM IE	PWM TOG	PWMMOD			PWM POL	PWM IF	PWM EN
rw								rw	rw	rw			rw	rw	rw
位 15:8		PWMCKD:PWM 工作时钟预分频配置寄存器 00H: 不分频 01H: 2分频 02H: 3分频 FEH: 255分频 FFH: 256分频													
位 7		PWMIE:PWM 中断使能, 1 有效													
位 6		PWMTOG:PWM 输出取反使能控制位, 1 有效													
位 5:3		PWMMOD:PWMm 作为 LED 驱动时, 连续发送字节数配置寄存器, 0 表示 PWM 不作为 LED 驱动使用, 1~7 表示 PWM 每发送 1~7 字节数据就暂停 1 次 备注: 1. 详细使用参考 LEWTM。													
位 2		PWMPOL: PWM 作为 LED 驱动时, 发送数据取反使能控制位, 1 有效 备注: 1. 当 PWMMOD! =0 时, 对应的 PWMPOL 的值才有意义; 2. 当 PWMPOL=1 时, 如果对应的 LEDAT=01010101B, 那么实际上发送的将会是 10101010B													
位 1		PWM 中断标志位, 写 1 清 0													
位 0		PWM 使能控制位, 1 有效													

19.2.2 LEDAT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						LEF0	LEBS Y0	LEDAT							
rw						rw	rw	rw							
位 15:10		保留													
位 9		LEF0: LEDAT0 数据缓存满标志, 1 表示 LEDAT0 的数据缓存处于满状态(此时如果对 LEDAT0 写数据是无效的), 0 表示可以向 LEDAT0 写入新的数据 备注: LEDAT0 有 1 字节的数据缓存, 所以在 LEDAT0 数据存储空间没有写数据之前, 可以连续写入 2 字节数据, 之后再想向 LEDAT0 内写数据, 都必须在检测到 LEF0 为 0 才可以, 因为 LEF0 为 0 表示 LEDAT0 数据缓存中至少有 1 字节已经发送完成。													
位 8		LEBSY0: LEDAT0 数据发送忙标志, 1 表示此时 LEDAT0 的数据缓存中的数据还没有全部发送完成, 0 表示全部发送完成													

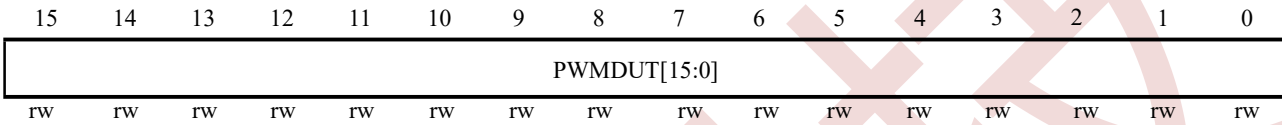
位 7:0	LEDAT: LED 驱动数据 备注: LEDAT 的数据按照从 MSB 到 LSB 的顺序发送
-------	---

19.2.3 PWMDIV



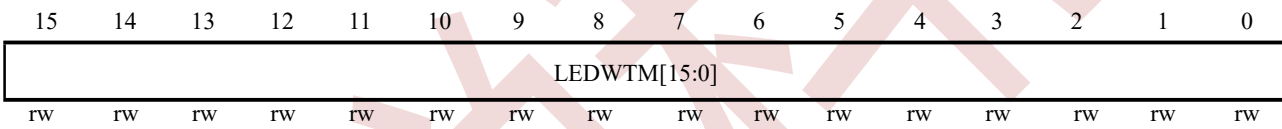
位 15:0	PWMDIV: PWM 周期配置寄存器
--------	---------------------

19.2.4 PWMDUT



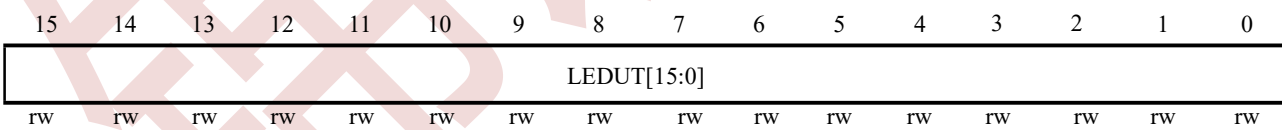
位 15:0	PWMDUT: LED 发送数据“0”占空比配置寄存器
--------	-----------------------------

19.2.5 LEDWTM



位 15:0	LED 暂停时间配置寄存器 备注: 1. 每发送 PWMMOD 字节数据之后, 暂停 (LEWTM+1) 个 PWM 的工作时钟后进入下一次传输。 2. LED1 的暂停时间都由同一个 LEWTM 决定。
--------	---

19.2.6 LEDUT



位 15:0	LED 发送数据“1”占空比配置寄存器 备注: 1. 级联 LED 的驱动波形中, 每 1 位数据的周期都由对应的 PWM1DIV 决定, 而数据“1”的占空比由 LEDUT 决定, 数据“0”的占空比由 PWMmDUT 决定; 2. 如果 LEDAT=01010101B, 同时对应的 PWMPOL=1, 那么实际发送的数据按照 BIT7-BIT6-BIT5-BIT4-BIT3-BIT2-BIT1-BIT0 顺序就是 1-0-1-0-1-0-1-0, 而且 BIT7/BIT5/BIT3/BIT1 的占空比由 LEDUT 决定, BIT6/BIT4/BIT2/BIT0 的占空比由对应的 PWMDUT 决定, 即 LEDUT 的起效在 PWMPOL 之后; 3. LED1 的数据“1”的占空比都由同一个 LEDUT 决定。
--------	--

20 SPI接口

20.1 SPI 简介

串行外设接口（SPI）允许芯片与外部设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为外部从设备提供通信时钟（SCK）。接口还能以多主机配置方式工作。

它可用于多种用途，包括使用一条双向数据线的双线单工同步传输，还可使用 CRC 校验的可靠通信。

20.2 SPI 主要特性

20.2.1 SPI 特性

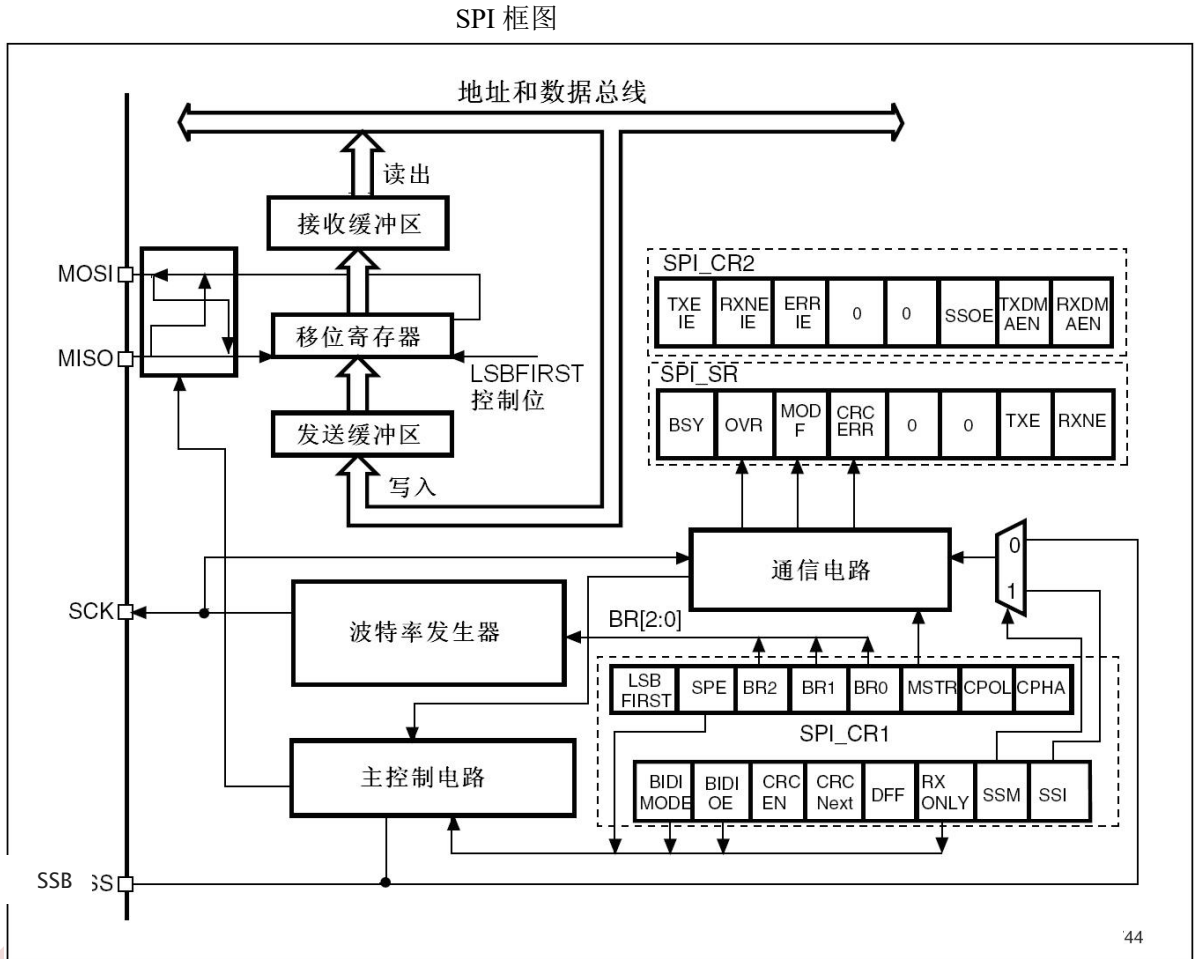
- 3 线全双工同步传输
- 带或不带第三根双向数据线的双线单工同步传输
- 8 或 16 位传输帧格式选择
- 主或从操作
- 支持多主模式
- 8 个主模式波特率预分频系数（最大为 $f_{\text{CLK}}/2$ ）
- 从模式频率（最大为 $f_{\text{CLK}}/2$ ）
- 主模式和从模式的快速通信：最大 SPI 速度达到 16MHz
- 主模式和从模式下均可以由软件或硬件进行 SSB 管理：主/从操作模式的动态改变
- 可编程的时钟极性和相位
- 可编程的数据顺序，MSB 在前或 LSB 在前
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持可靠通信的硬件 CRC
 - 在发送模式下，CRC 值可以被作为最后一个字节发送
 - 对接收到的最后一个字节自动进行 CRC 校验
- 可触发中断的主模式故障、过载以及 CRC 错误标志

支持 DMA 功能的 1 字节发送和接收缓冲器：产生发送和接收请求

20.3 SPI 功能描述

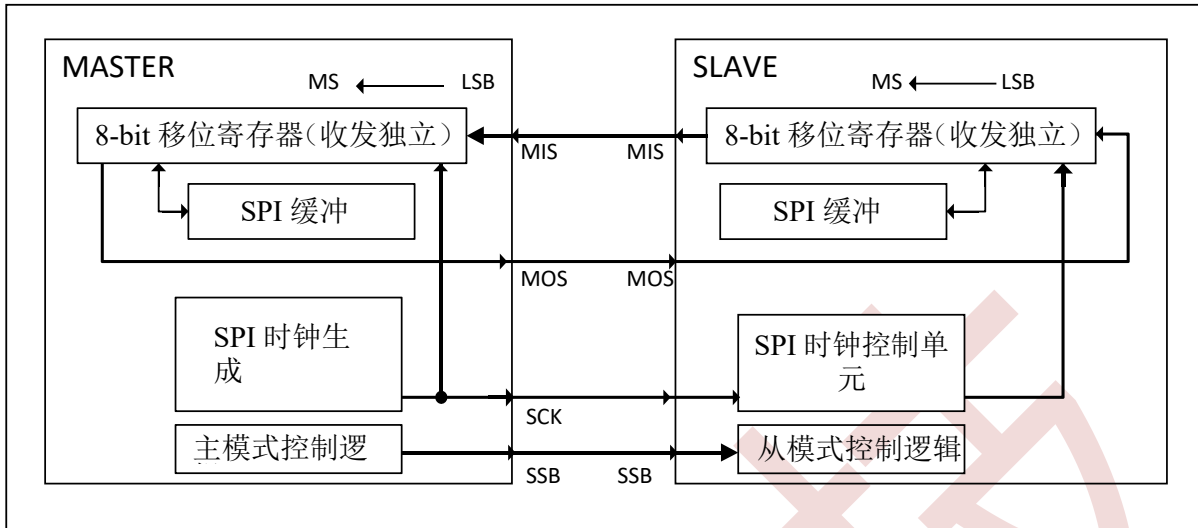
20.3.1 概述

SPI 功能框图



通常 SPI 通过 4 个管脚与外部器件相连：

- **MISO**: 主设备输入/从设备输出管脚。该管脚在从模式下发送数据，在主模式下接收数据。
- **MOSI**: 主设备输出/从设备输入管脚。该管脚在主模式下发送数据，在从模式下接收数据。
- **SCK**: 串口时钟，作为主设备的输出，从设备的输入。
- **SSB**: 从设备选择。这是一个可选的管脚，用来选择主/从设备。它的功能是用来作为“片选管脚”，让主设备可以单独地与特定从设备通信，避免数据线上的冲突。从设备的 SSB 管脚可以由主设备当作一个标准的 IO 来驱动。一旦被使能（SSOE 位），SSB 管脚也可以作为输出管脚，并在 SPI 设置为主模式时拉低；此时，所有 SSB 管脚连接到主设备 SSB 管脚的 SPI 设备，会检测到低电平，如果它们被设置为 SSB 硬件模式，就会自动进入从设备状态。当 SPI 被配置成主模式，并且 SSB 被配置成输出（MSTR=1,SSOE=0），如果 SSB 被拉低，SPI 进入主模式错误状态：MSTR 位被自动清除，并且器件被配置成从模式。



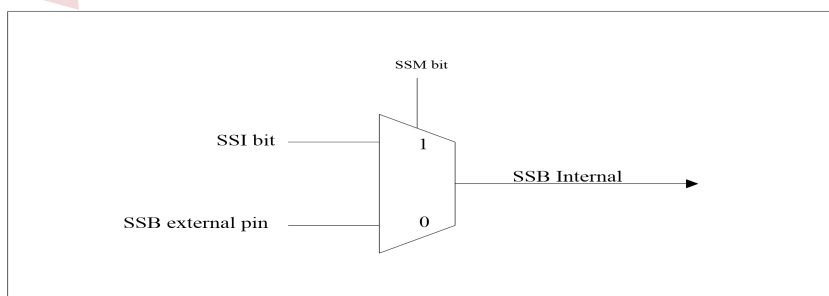
MOSI 脚相互连接，MISO 脚相互连接。这样，数据在主和从之间串行地传输（MSB 位在前）。通信总是由主设备发起。主设备通过 MOSI 脚把数据发送给从设备，从设备通过 MISO 引脚回传数据。这意味着全双工通信的数据输出和数据输入是用同一个时钟信号同步的；时钟信号由主设备通过 SCK 脚提供。

20.3.2 从选择（SSB）脚管理

有 2 种 SSB 模式，通过 SPI_CR1 寄存器的 SSM 位来选择硬件或软件从设备选择管理：

- 软件 SSB 模式（SSM=1）：可以通过设置 SPI_CR1 寄存器的 SSM 位来使能这种模式（见图 22_2）。在这种模式下外部 SSB 管脚可以用作它用，而 SLAVE 选择信号可以通过写 SPI_CR1 的 SSI 位来进行内部驱动。
- 硬件 SSB 模式（SSM=0），取决于 SSB 输出的配置（SPI_CR2 的 SSOE 位）分两种情况：
 - SSB 输出被使能（SSM=0,SSOE=1）：当 MCU 工作为 SPI 主模式，当主机开始通信时 SSB 管脚被拉低，并且一直保持到 SPI 被关闭。
 - SSB 输出被关闭（SSM=0,SSOE=0）：此配置允许工作在主模式环境下的器件具备多主机功能。如果器件被设置成从机，SSB 引脚作为一个传统的 SSB 输入：当 SSB 为低时，从机被选中，为高时则没有选中。

硬件/软件的从选择管理



时钟信号的相位和极性

SPI_CR 寄存器的 CPOL 和 CPHA 位，能够组合成四种可能的时序关系。

CPOL（时钟极性）位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CPOL 被清'0'，SCK 引脚在空闲状态保持低电平；如果 CPOL 被置'1'，SCK 引脚在空闲状态保持高电平。

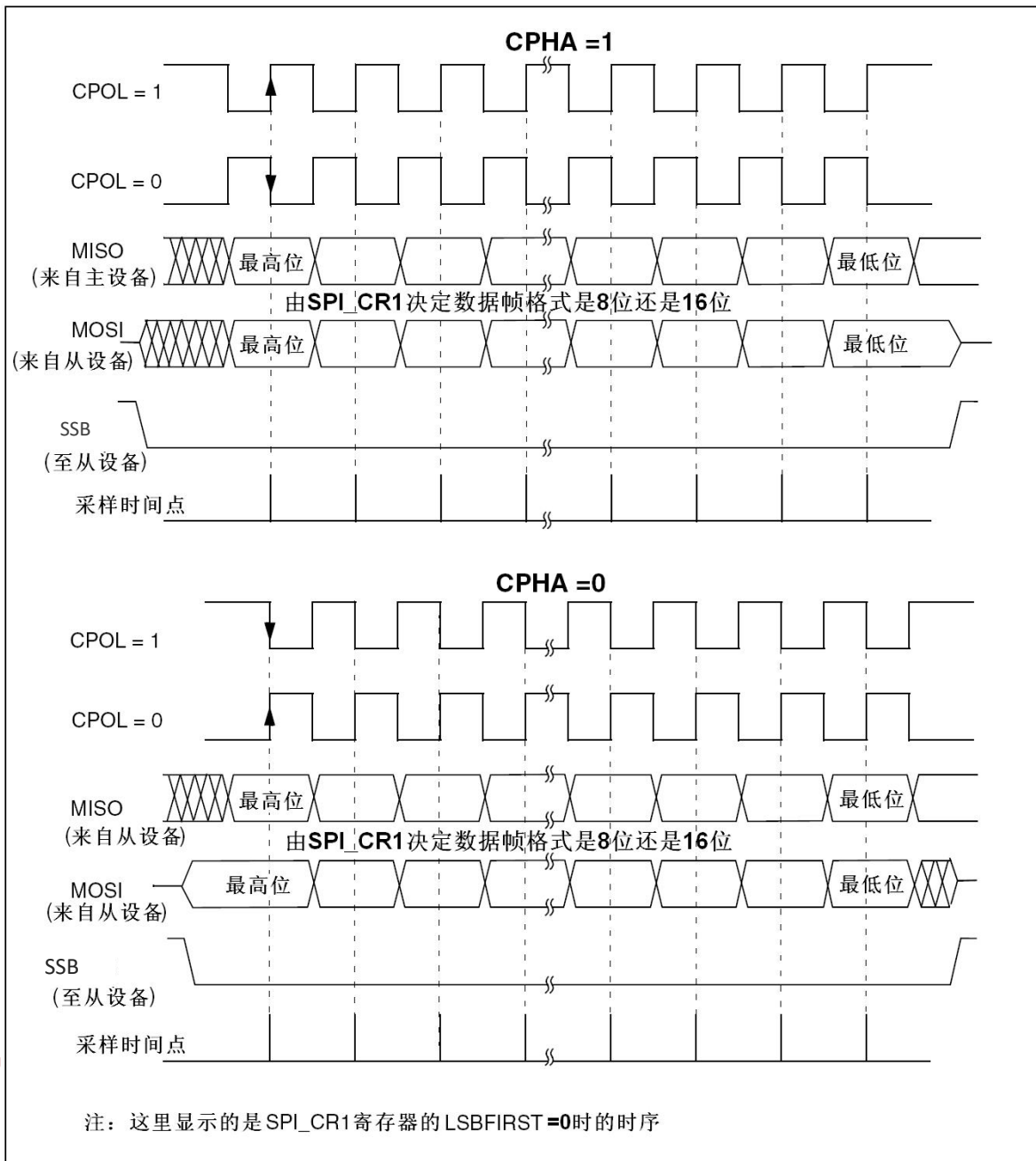
如果 CPHA（时钟相位）位被置'1'，SCK 时钟的第二个边沿（CPOL 位为 0 时就是下降沿，CPOL 位为 1 时就是上升沿）进行数据位的采样，数据在第二个时钟边沿被锁存。如果 CPHA 位被清'0'，SCK 时钟的第一边沿（CPOL 位为 0 时就是上升沿，CPOL 位为 1 时就是下降沿）进行数据位采样，数据在第一个时钟边沿被锁存。

CPOL 时钟极性和 CPHA 时钟相位的组合选择数据采样的时钟边沿。

图 22_4 显示了 SPI 传输的 4 种 CPHA 和 CPOL 位组合。此图可以解释作为主设备和从设备的 SCK 脚、MISO 脚、MOSI 引脚直接连接的主或从时序图。

- 注意:
1. 在改变 CPOL/CPHA 位之前，必须清除 SPE 位将 SPI 禁止。
 2. 主机和从机必须配置成相同的时序模式。
 3. SCK 的空闲状态必须和 SPI_CR1 寄存器指定的极性一致（CPOL 为 1 时，空闲时应上拉 SCK 为高电平；CPOL 为 0 时，空闲时应下拉 SCK 为低电平）。
 4. 数据帧格式（8 位或 16 位）由 SPI_CR1 寄存器的 DFF 位选择，并且决定发送/接收的数据长度。

数据时钟时序图



1. 这些时序体现的都是 SPI_CR1 寄存器的 LSBFIRST 被重置（置 0）时的情况

数据帧格式

根据 SPI_CR1 寄存器中的 LSBFIRST 位，输出数据位时可以 MSB 在先也可以 LSB 在先。

根据 SPI_CR1 寄存器的 DFF 位，每个数据帧可以是 8 位或是 16 位。所选择的数据帧格式对发送和/或接收都有效。

20.3.2 SPI 从模式

在从配置里，SCK 引脚用于接收到从主设备来的串行时钟。SPI_CR1 寄存器中 BR[2:0]的设置不影响数据传输速率。

注意：推荐在主机发送时钟之前使能 SPI 的 slave。如果没有，可能会发送不期望的数据发送。从机的数据寄存器需要在第一个通信时钟沿或者正在进行的通信结束前被准备好。在主机和从机使能前，必须为通信时钟的极性设置一个稳定的状态值。

配置步骤

1. 设置 DFF 位以定义数据帧格式为 8 位或 16 位。
2. 选择 CPOL 和 CPHA 位来定义数据传输和串行时钟之间的相位关系（见图 22_4）。为保证正确的数据传输，从设备和主设备的 CPOL 和 CPHA 位必须配置成相同的方式。
3. 帧格式（MSB 在前还是 LSB 在前取决于 SPI_CR1 寄存器中的 LSBFIRST 位）必须和主设备相同。
4. 硬件模式下（参考从选择（SSB）脚管理部分），在完整的数据帧（8 位或 16 位）发送过程中，SSB 引脚必须为低电平。软件模式下，设置 SPI_CR1 寄存器中的 SSM 位并清除 SSI 位。
5. 清除 MSTR 位，设置 SPE 位，使相应引脚工作于 SPI 模式下。

在这个配置里，MOSI 引脚是数据输入，MISO 引脚是数据输出。

数据发送过程

数据字在一个写入周期被并行地写入发送缓冲器。

当从设备收到时钟信号，并且在 MOSI 引脚上出现第一个数据位时，发送过程开始，第一个位被发送出去。余下的位（对于 8 位数据帧格式，还有 7 位；对于 16 位数据帧格式，还有 15 位）被装进移位寄存器。当发送缓冲器中的数据传输到移位寄存器时，SPI_SP 寄存器里的 TXE 标志被设置。如果设置了 SPI_CR2 寄存器上的 TXEIE 位，将会产生中断。

数据接收过程

对于接收方，当数据接收完成时：

- 移位寄存器中的数据传送到接收缓冲器，SPI_SR 寄存器中的 RXNE 标志被设置。
- 如果设置了 SPI_CR2 寄存器中的 RXEIE 位，则产生中断。

在最后一个采样时钟边沿后延迟 3 个 PCLK 周期，RXNE 位被置'1'，移位寄存器中接收到的数据字节被传送到接收缓冲器。当读 SPI_DR 寄存器时，SPI 设备返回这个值。

读 SPI_DR 寄存器时，RXNE 位被清除。

20.3.3 SPI 主模式

在主配置时，串行时钟在 SCK 脚产生。

配置步骤

1. 通过 SPI_CR1 寄存器的 BR[2:0]位定义串行时钟波特率。
2. 选择 CPOL 和 CPHA 位，定义数据传输和串行时钟间的相位关系（见图 22_4）。
3. 设置 DFF 位来定义 8 或 16 位数据帧格式。
4. 配置 SPI_CR1 寄存器的 LSBFIRST 位定义帧格式。
5. 如果 SSB 引脚需要工作在输入模式，硬件模式中在整个数据帧传输期间应把 SSB 脚连接到高电平；在软件模式中，需设置 SPI_CR1 寄存器的 SSM 和 SSI 位。如果 SSB 引脚工作在输出模式，则只需设置 SSOE 位。
6. 必须设置 MSTR 和 SPE 位（只当 SSB 脚被连到高电平，这些位才能保持置位）。

在这个配置中，MOSI 脚是数据输出，而 MISO 脚是数据输入。

数据发送过程

当一字节写进发送缓冲器时，发送过程开始。

在发送第一个数据位时，数据自动被并行地(通过内部总线)传入移位寄存器，而后串行地移出到 MOSI 脚上；MSB 在先还是 LSB 在先，取决于 SPI_CR1 寄存器中的 LSBFIRST 位。数据从发送缓冲器传输到移位寄存器时 TXE 标志将被置位，如果设置 SPI_CR1 寄存器中的 TXEIE 位，将产生中断。

数据接收过程

对于接收器来说，当数据传输完成时：

- 移位寄存器里的数据传送到接收缓冲器，并且 RXNE 标志被置位。
- 如果 SPI_CR2 寄存器中的 RXEIE 位被设置，则产生中断。

在最后一个采样时钟沿后延迟 0.5 个 SCK 周期，RXNE 位被设置，在移位寄存器中接收到的数据自动被传送到接收缓冲器。读 SPI_DR 寄存器时，SPI 设备返回接收到的数据字。

读 SPI_DR 寄存器将清除 RXNE 位。

一旦传输开始，如果下一个将发送的数据被放进了发送缓冲器，就可以维持一个连续的传输流。在试图写发送缓冲器之前，需确认 TXE 标志应该是 1。

注意：在发送之间，当一个主机和需要取消选中的 SPI 从机通信时，SSB 引脚必须被配置成一个 GPIO 或者其他的 GPIO 必须被软件使用并翻转。

20.3.4 单工通信

SPI 能够以两种配置工作于单工方式：

- 1 条时钟线和 1 条双向数据线
- 1 条时钟线和 1 条数据线（只接收或只发送方式）

1 条时钟线和 1 条双向数据线（BIDIMODE=1）

设置 SPI_CR1 寄存器中的 BIDIMODE 位而启用此模式。在这个模式中，SCK 用作时钟，主模式中的 MOSI 或从模式中的 MISO 用作数据通信。传输的方向由 SPI_CR2 寄存器里的 BIDIOE 控制，当这个位是 1 的时候，数据线是输出，否则是输入。

1 条时钟线和 1 条单向数据线（BIDIMODE=0）

在此模式下，可以将 SPI 应用在只发送模式或者只接收模式。

- 只发送模式类似于全双工模式（BIDIMODE=0, RXONLY=0）：数据在发送引脚被发送（主模式下是 MOSI 引脚，从模式下是 MISO 引脚），接收引脚（主模式下为 MISO，从模式下为 MOSI）可以被用来作为一个通用型的 IO。在此状况下，应用只需要忽略 RX 缓冲器（数据寄存器被读时，不包含接收值）。
- 在只接收模式下，可以通过设置 SPI_CR1 寄存器中的 RXONLY 位来禁止 SPI 输出功能。这样的话，发送的 IO 端口（主模式下为 MOSI，从模式下为 MISO）就不需要使用，可以用作其他用途。

为启动只接收模式通信，配置并使能 SPI：

- 在主模式中，一旦使能 SPI，通信立即启动，当 SPE 位复位时通信即停止，并且当前的接收停止。在此模式下不需要读 BSY 标识位。当 SPI 通信进行时一直被置 1。
- 在从模式中，只要 SSB 被拉低（或者在 SSB 软件模式下 SSI 位为 0）以及 SCK 持续送到从设备，SPI 就一直接收。

注意：当 SPI_CR1 寄存器中的 RXONLY 位为'0'时，SPI 可以工作于只发送模式，接收脚（主设备的 MISO，或者从设备的 MOSI）可以当作通用 IO 口使用。因此读数据寄存器时，读不到接收的值。

20.3.5 状态标志

应用程序通过 3 个状态标志可以完全监控 SPI 总线的状态。

忙（Busy）标志

BSY 标识由硬件设置和清除（写此标识位无影响）。BSY 标识表示 SPI 通信层的状态。

当 BSY 位被置 1，表明 SPI 正在忙于通信。在主模式的双向接收模式（MSTR=1，BIDIMODE=1 和 BIDIOE=0）下是例外，BSY 标识位在接收期间保持低状态。

如果软件想要关闭 SPI 或者进入停止状态（或者关闭外设时钟），此标识可以有效检测传输的末尾。可以避免破坏最后一次传输。为此，下面描述的步骤必须被重视。

此标识也可以用来避免在多主机系统中的写入冲突。

当传输开始时，BSY 标识被设置，在主模式的双向接收模式（MSTR=1，BIDIMODE=1 和 BIDIOE=0）

下是例外。

在此情况下被清除：

- 传输完成时（主模式下如果通信是连续不断时则例外）
- SPI 被关闭
- 发生主模式错误（MODF=1）

当通信不是连续不断时，在每次通信间隔，BSY 标识会变为低。

当通信是连续不断的时：

- 主模式下，BSY 标识位在所有的传输中保持为高。
- 从模式下，BSY 标识位在每次传输间隔有一个 SPI 时钟周期变为低

注意：不要使用 BSY 标识去处理每个数据的发送或是接收。最好是使用 TXE 和 RXNE 来代替。

当它被设置时，表明 SPI 正忙于通信，并且/或者在发送缓冲器里有一个有效的数据字正在等待被发送。此标志的目的是说明在 SPI 总线上是否有正在进行的通信。以下情况时此标志将被置位：

1. 数据被写进主设备的 SPI_DR 寄存器上。
2. SCK 时钟出现在从设备的时钟引脚上。

发送/接收一个字（字节）完成后，BUSY 标志立即清除；此标志由硬件设置和清除。监视此标志可以避免写冲突错误。写此标志无效。仅当 SPE 位被设置时此标志才有意义。

注：在主接收模式下（单线双向），不要查询忙标志位（BUSY_FLAG）

发送缓冲器空闲标志（TXE）

此标志被置位时表明发送缓冲器为空，因此下一个待发送的数据可以写进缓冲器里。当发送缓冲器被写入待发送的数据时，TXE 标志被清除。

接收缓冲器非空（RXNE）

此标志为‘1’时表明在接收缓冲器中包含有效的接收数据。读 SPI 数据寄存器可以清除此标志。

20.3.6 利用 DMA 的 SPI 通信

为了达到最大通信速度，需要及时往 SPI 填充发送数据，同样接收缓冲器中的接收数据也必须及时读出以防止溢出。为了方便传输，SPI 实现了一种采用简单的请求/应答的 DMA 机制。

当 SPI_CR2 寄存器上的对应使能位被设置时，发出 DMA 传输请求。发送缓冲器和接收缓冲器亦有各自的 DMA 请求。

- 在发送中，每次 TXE 被置 1，产生 DMA 请求。然后 DMA 对 SPI_DR 寄存器写（清除 TXE 标识）。
- 在接收中，每次 RXNE 被置 1，产生 DMA 请求。然后 DMA 读 SPI_DR 寄存器（清除 RXNE 标识）。

当 SPI 只被用来发送数据，可能只需要使能 SPI Tx 的 DMA 通道。在此情况下，由于接收数据没有被读取，OVR 标识被设置。

当 SPI 只被用来接收数据，可能只需要使能 SPI Rx 的 DMA 通道。

在发送模式，当 DMA 已经写入所有要发送的数据（DMA_ISR 寄存器的 TCIF 标识位被设置），可以监控 BSY 标识位确保 SPI 通信完成。避免在禁用 SPI 或者进入停止模式前破坏最后一次发送。软件必须先等待 TXE=1 然后直到 BSY=0。

注意：在不连续的通信中，在对 SPI_DR 的写操作和 BSY 被置 1 之间有两个 APB 时钟周期的延迟。因此，在写入最后一个数据后，强制先等待 TXE=1，然后直到 BSY=0。

20.3.7 错误标志

主模式错误 (MODF)

主模式故障仅发生在：在片选引脚硬件模式管理下，主设备的 SSB 脚被拉低时；或者在片选引脚软件模式管理下，SSI 位为低时；MODF 位被自动置位。主模式故障对 SPI 设备有以下影响：

- MODF 位被置位，如果设置了 ERRIE 位，则产生 SPI 中断。
- SPE 位被复位。这将停止一切输出，并且关闭 SPI 接口。
- MSTR 位被复位，因此强迫此设备进入从模式。

下面的步骤用于清除 MODF 位：

1. 当 MODF 位被置位时，执行一次对 SPI_SR 寄存器的读或写操作
2. 然后写 SPI_CR1 寄存器

在有多个 MCU 的系统中，为了避免出现多个从设备的冲突，在 MODF 位的清除序列中，SSB 引脚必须被拉高，再对 MODF 位进行清零。在清零序列之后，SPE 和 MSTR 位可以恢复到它们的原始状态。出于安全的考虑，当 MODF 位被置位的情况下，硬件不允许设置 SPE 和 MSTR 位。通常配置下，从设备的 MODF 位不能被置位。然而，在多主配置里，一个设备可以在从模式下被设置 MODF 位。此时，MODF 位指示可能出现了系统控制的多主机冲突。中断程序可以执行一个复位或返回到默认状态来从错误状态中恢复。

溢出错误

当主设备已经发送了数据字节，以及从设备还没有清除前一个数据字节产生的 RXNE 时，发生溢出错误。当产生溢出错误时：

- OVR 位被设置；当设置了 ERRIE 位时，则产生中断。

此时，接收器缓冲器的内容不会更新为主设备发送的新数据，读 SPI_DR 寄存器返回的是之前未读的字节，所有随后传送的字节都被丢弃。

依次读 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

20.3.8 关闭 SPI

当通讯结束，可以通过关闭 SPI 外设来终止通讯。清除 SPE 位即可关闭 SPI。

对某些配置而言，在传输正在进行时关闭 SPI 或者进入停止模式会导致传输被破坏以及/或者 BSY 标识位可能会变得不可靠。

只要设备不处于主传送模式下，在最后一个字节的传输未完成时关闭 SPI 并不会影响通讯的可靠性。

为了避免这些影响，在关闭 SPI 时建议遵照以下的步骤：

在主或从的全双工模式下 (BIDIMODE=0, RXONLY=0)

1. 等到 RXNE=1，接收最后一个数据
2. 等到 TXE=1
3. 等到 BSY=0
4. 关闭 SPI(SPE=0)，以及最后进入停止模式（或者关闭外设时钟）

在主或从的单向只发送模式 (BIDIMODE=0, RXONLY=0) 或者双向发送模式 (BIDIMODE=1, BIDIOE=1) 下

在最后一个数据被写入 SPI_DR 寄存器后：

1. 等到 TXE=1
2. 等到 BSY=0
3. 关闭 SPI(SPE=0)，以及最后进入停止模式（或者关闭外设时钟）

在主机单向只接收模式 (MSTR=1, BIDIMODE=0, RXONLY=1) 或者主机双向接收模式 (MSTR=1, BIDIMODE=1, BIDIOE=0) 下

这种情况下必须用特有的方式管理，以确保 SPI 不会启动一个新的传输：

1. 等待倒数第二个出现的 $RXNE=1$
2. 然后在关闭 SPI($SPE=0$) 之前等待一个 SPI 时钟周期（使用软件循环）
3. 然后在进入停止模式（或者关闭外设时钟）之前等待最后一个 $RXNE=1$

注意：在主双向接收模式（ $MSTR=1, BIDIMODE=1, BIDIOE=0$ ）下，传输期间 BSY 标识保持为低。

在从的只接收模式（ $MSTR=0, BIDIMODE=0, RXONLY=1$ ）或者双向接收模式（ $MSTR=0, BIDIMODE=1, BIDIOE=0$ ）下

1. 如果你想进入停止模式，在进入停止模式（或者关闭外设时钟）之前必须首先等待 $BSY=0$

20.3.9 SPI 中断

表 22-1 SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲器空标志	TXE	TXEIE
接收缓冲器非空标志	RXNE	RXNEIE
主模式错误事件	MODF	ERRIE
溢出错误	OVR	

20.4 SPI 寄存器描述

20.4.1 SPI 寄存器概览

表 22-2 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	SPI_CR1	SPI 控制寄存器 1	0x0000
0x04	SPI_CR2	SPI 控制寄存器 2	0x0000
0x08	SPI_SR	SPI 状态寄存器	0x0002
0x0C	SPI_DR	SPI 数据寄存器	0x0000

20.4.2 SPI 控制寄存器 1(SPI_CR1)

地址偏移: 0x00

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MOD E	BIDI OE	X4SEL	DFE	RX ONL Y	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MST R	CPOL	CPHA	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	BIDIMODE: 双向数据模式使能 (Bidirectional data mode enable)0: 选择“双线双向”模式; 1: 选择“单线双向”模式。
位 14	BIDIOE: 双向模式下的输出使能 (Output enable in bidirectional mode) 和 BIDIMODE 位一起决定在“单线双向”模式下数据的输出方向 0: 输出禁止 (只收模式); 1: 输出使能 (只发模式)。 这个“单线”数据线在主设备端为 MOSI 引脚, 在从设备端为 MISO 引脚。
位 12~13	2 线/4 线模式选择: 2'b01 表示 spi 选择 2 线模式输入输出; 2'b10 表示 spi 选择 4 线模式输入输出; 其他: 表示选择单线模式输入输出

位 11	<p>DFE: 数据帧格式 (Data frame format) 0: 使用 8 位数据帧格式进行发送/接收; 1: 使用 16 位数据帧格式进行发送/接收。 注: 只有 SPI 禁止 (SPE=0) 时, 才能写该位, 否则出错。</p>								
位 10	<p>RXONLY: 只接收 (Receive only) 该位和 BIDIMODE 位一起决定在“双线双向”模式下的传输方向。在多个从设备的配置中, 在未被访问的从设备上该位被置 1, 使得只有被访问的从设备有输出, 从而不会造成数据线上数据冲突。 0: 全双工 (发送和接收); 1: 禁止输出 (只接收模式)。</p>								
位 9	<p>SSM: 软件从设备管理 (Software slave management) 当 SSM 被置位时, SSB 引脚上的电平由 SSI 位的值决定。0: 禁止软件从设备管理; 1: 启用软件从设备管理。</p>								
位 8	<p>SSI: 内部从设备选择 (Internal slave select) 该位只在 SSM 位为 '1' 时有意义。它决定了 SSB 上的电平, 在 SSB 引脚上的 I/O 操作无效。</p>								
位 7	<p>LSBFIRST: 帧格式 (Frame format) 0: 先发送 MSB; 1: 先发送 LSB。 注: 当通信进行时不能改变该位的值。</p>								
位 6	<p>SPE: SPI 使能 (SPI enable) 0: 禁止 SPI 设备; 1: 开启 SPI 设备。</p>								
位 5:3	<p>BR[2:0]: 波特率控制 (Baud rate control)</p> <table style="width: 100%; border: none;"> <tr> <td style="padding: 0 10px;">000: $f_{PCLK}/2$</td> <td style="padding: 0 10px;">001: $f_{PCLK}/4$</td> <td style="padding: 0 10px;">010: $f_{PCLK}/8$</td> <td style="padding: 0 10px;">011: $f_{PCLK}/16$</td> </tr> <tr> <td style="padding: 0 10px;">100: $f_{PCLK}/32$</td> <td style="padding: 0 10px;">101: $f_{PCLK}/64$</td> <td style="padding: 0 10px;">110: $f_{PCLK}/128$</td> <td style="padding: 0 10px;">111: $f_{PCLK}/256$</td> </tr> </table> <p>当通信正在进行的时候, 不能修改这些位。</p>	000: $f_{PCLK}/2$	001: $f_{PCLK}/4$	010: $f_{PCLK}/8$	011: $f_{PCLK}/16$	100: $f_{PCLK}/32$	101: $f_{PCLK}/64$	110: $f_{PCLK}/128$	111: $f_{PCLK}/256$
000: $f_{PCLK}/2$	001: $f_{PCLK}/4$	010: $f_{PCLK}/8$	011: $f_{PCLK}/16$						
100: $f_{PCLK}/32$	101: $f_{PCLK}/64$	110: $f_{PCLK}/128$	111: $f_{PCLK}/256$						
位 2	<p>MSTR: 主设备选择 (Master selection) 0: 配置为从设备; 1: 配置为主设备。 注: 当通信正在进行的时候, 不能修改该位。</p>								
位 1	<p>CPOL: 时钟极性 (Clock polarity) 0: 空闲状态时, SCK 保持低电平; 1: 空闲状态时, SCK 保持高电平。 注: 当通信正在进行的时候, 不能修改该位。</p>								
位 0	<p>CPHA: 时钟相位 (Clock phase) 0: 数据采样从第一个时钟边沿开始; 1: 数据采样从第二个时钟边沿开始。 注: 当通信正在进行的时候, 不能修改该位。</p>								

20.4.3 SPI 控制寄存器 2(SPI_CR2)

地址偏移: 0x04

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						X4OEN	X2OEN	TXEIE	RXNEIE	ERRIE	保留		SSOE	TXDMAEN	RXDMAEN
res								rw	rw	rw	res		rw	rw	rw

位 15:10	保留
位 9	1 表示 4 线模式信号线输出使能打开; 0 表示关闭 4 线模式的输出端口
位 8	1 表示 2 线模式信号线输出使能打开; 0 表示关闭 2 线模式的输出端口
位 7	TXEIE: 发送缓冲区空中断使能 (Tx buffer empty interrupt enable) 0: 禁止 TXE 中断; 1: 允许 TXE 中断, 当 TXE 标志置位为'1'时产生中断请求。
位 6	RXNEIE: 接收缓冲区非空中断使能 (RX buffer not empty interrupt enable) 0: 禁止 RXNE 中断; 1: 允许 RXNE 中断, 当 RXNE 标志置位时产生中断请求。
位 5	ERRIR: 错误中断使能 (Error interrupt enable) 当错误 (CRCERR、OVR、MODF) 产生时, 该位控制是否产生中断 0: 禁止错误中断; 1: 允许错误中断。
位 4:3	保留位, 硬件强制为 0。
位 2	SSOE: SS 输出使能 (SS output enable) 0: 禁止在主模式下 SS 输出, 该设备可以工作在多主设备模式; 1: 设备开启时, 开启主模式下 SS 输出, 该设备不能工作在多主设备模式。
位 1	TXDMAEN: 发送缓冲区 DMA 使能 (Tx buffer DMA enable)当该位被设置时, TXE 标志一旦被置位就发出 DMA 请求 0: 禁止发送缓冲区 DMA; 1: 启动发送缓冲区 DMA。
位 0	RXDMAEN: 接收缓冲区 DMA 使能 (Rx buffer DMA enable)当该位被设置时, RXNE 标志一旦被置位就发出 DMA 请求 0: 禁止接收缓冲区 DMA; 1: 启动接收缓冲区 DMA。

20.5.4 SPI 状态寄存器 (SPI_SR)

地址偏移: 0x08

复位值: 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BSY	OVR	MOD	保留			TXE	RXNE
res								r	r	r	rc w0			r	r

位 15:8	保留位, 硬件强制为 0
位 7	BSY: 忙标志 (Busy flag)0: SPI 不忙; 1: SPI 正忙于通信, 或者发送缓冲非空。该位由硬件置位或者复位。
位 6	OVR: 溢出标志 (Overrun flag)0: 没有出现溢出错误; 1: 出现溢出错误。
位 5	MODF: 模式错误 (Mode fault)0: 没有出现模式错误; 1: 出现模式错误。
位 4:2	保留
位 1	TXE: 发送缓冲为空 (Transmit buffer empty)0: 发送缓冲非空; 1: 发送缓冲为空。
位 0	RXNE: 接收缓冲非空 (Receive buffer not empty) 0: 接收缓冲为空; 1: 接收缓冲非空。

20.5.5 SPI 数据寄存器 (SPI_DR)

地址偏移: 0x0C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
位 15:0	<p>DR[15:0]: 数据寄存器 (Data register)</p> <p>待发送或者已经收到的数据</p> <p>数据寄存器对应两个缓冲区: 一个用于写(发送缓冲); 另外一个用于读(接收缓冲)。写操作将数据写到发送缓冲区; 读操作将返回接收缓冲区里的数据。</p> <p>对 SPI 模式的注释: 根据 SPI_CR1 的 DFF 位对数据帧格式的选择, 数据的发送和接收可以是 8 位或者 16 位的。为保证正确的操作, 需要在启用 SPI 之前就确定好数据帧格式。</p> <p>对于 8 位的数据, 缓冲器是 8 位的, 发送和接收时只会用到 SPI_DR[7:0]。在接收时, SPI_DR[15:8]被强制为 0。</p> <p>对于 16 位的数据, 缓冲器是 16 位的, 发送和接收时会用到整个数据寄存器, 即 SPI_DR[15:0]。</p>														

21 实时时钟（RTC）

21.1 简介

RTC 模块内部包含一组连续计数的计数器。它作为一个独立的定时器，在相应软件配置下，可提供日历功能。修改计数器的值可以重新设置系统当前的时间。

RTC 模块和时钟配置系统（RCC_BDCR 寄存器）处于后备区域，即在系统复位或从停机模式唤醒后，RTC 的设置和时间维持不变。

系统复位后，对后备寄存器和 RTC 的访问被禁止，这是为了防止对后备区域（BKP）的意外写操作。执行以下操作将使能对后备寄存器和 RTC 的访问：

- 设置寄存器 RCC_APB1ENR 的 PWREN 和 BKPEN 位，使能电源和后备接口时钟
- 设置寄存器 RCC_CR 的 DBP 位，使能对后备寄存器和 RTC 的访问。

21.2 主要特性

- 可编程的预分频系数：分频系数最高为 2^{20} 。
- 32 位的可编程计数器，可用于较长时间段的测量。
- 包含 2 个独立的时钟：RTC 时钟以及用于 APB1 接口的 PCLK1，其中 RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一以上。
- 可以选择以下 2 种 RTC 的时钟源：
 - ◆ LSE 振荡器时钟；
 - ◆ LSI 振荡器时钟。
- 4 个专门的可屏蔽中断：
 - ◆ 闹钟中断，用来产生一个软件可编程的闹钟中断。
 - ◆ 毫秒中断，用来产生一个软件可编程的毫秒中断。
 - ◆ 秒中断，用来产生一个可编程的周期性中断信号（最长可达 1 秒）。
 - ◆ 溢出中断，指示内部可编程计数器溢出并回转为 0 的状态。
- 2 个独立的复位类型：
 - ◆ APB 接口由系统复位；
 - ◆ RTC 核心（预分频器、闹钟、计数器和分频器）只能由后备域复位。

21.2.1 功能概述

RTC 主要包含两个部分，APB1 接口部分与可编程计数部分。

APB1 接口部分用于连接 APB1 总线，该部分包含一组 16 位寄存器。

可编程预分频器，最大可编程时钟周期 SC_CLK 为 1 秒。如果在 RTC_CR 控制寄存器中设置相应的中断使能位，则在每个 SC_CLK 时钟周期中会产生一个中断（秒中断）。

当 32 位可编程计数器计数到 64 分频时，此位由硬件置'1'。如果 RTC_CRH 寄存器中 MSIE=1，则产生中断。（毫秒秒中断）。

内部 32 位的可编程计数器可用于配置系统时间。系统时间按 SC_CLK 时钟周期累加，当设置 RTC_CR 控制寄存器中相应的中断使能位，并且系统时间的值与 RTC_ALR 寄存器值匹配时，会产生一个闹钟中断。

21.2.2 模块复位

RTC_PRL、RTC_ALR、RTC_CNT、RTC_DIV 寄存器只能通过上电复位或备份域软件复位。RTC 的其他寄存器可以通过系统复位或电源复位进行复位。

21.2.3 寄存器读取

RTC 核完全独立于 RTC APB1 接口。

软件可以通过 APB1 接口访问 RTC 的寄存器组的值，包括分频计数以及闹钟。但是可读寄存器的值和 RTC 标志位只有在 RTC 时钟与 RTC APB1 时钟重新同步时被更新。

当 APB1 接口处于关闭状态，重新打开后就可以进行读操作，但由于跨时钟域数据没有同步完成，通过 APB1 接口读出的 RTC 寄存器数据可能是错误数据或者 0。

可能造成这种情况的几种情形如下：

- 发生系统复位或电源复位
- 系统刚从停机模式唤醒。

所有以上情况中，APB1 接口被禁止时（复位、无时钟或断电）RTC 核仍保持运行状态。因此，如果 RTC 的 APB1 接口处于被禁止的状态，读取 RTC 寄存器时，则软件必须等待硬件置位 RTC_CRL 寄存器中的同步标志位（RSF），读操作才能继续。

注： RTC 的 APB1 接口不受 WFI 和 WFE 等低功耗模式的影响。

21.2.4 寄存器配置

配置 RTC_CRL 寄存器中的 CNF 位为 1 后 RTC 进入配置模式，才能开启对 RTC_PRL、RTC_CNT、RTC_ALR、RTC_MSR 寄存器的写操作。只有当前一次的写操作结束后才能开始下一次对寄存器的写操作。查询 RTC_CR 寄存器中的 RTOFF 状态位，来判断 RTC 寄存器是否处于更新中，只有当 RTOFF 状态位是'1'时，才能写 RTC 寄存器。

具体配置过程如下：

- 查询 RTOFF 位，等待 RTOFF 的值变为'1'；
- 配置 CNF 位为'1'，RTC 进入配置模式；
- 对 RTC 寄存器进行写操作；
- 配置 CNF 位为 0，RTC 退出配置模式；
- 查询 RTOFF，当 RTOFF 位等于 1 时，写操作完成；

注：只有当 CNF 标志位被清除时，才能进行下一次写操作，该过程至少需要 3 个时钟周期；

21.2.5 RTC 标志的设置

RTC 核心的计数过程中，改变 RTC 计数器之前设置 RTC 秒标志（SECF）。

当计数器到达 0x00000000 之前的最后一个 RTC 时钟周期时，RTC 溢出标志（OWF）置 1。RTC_CNT 开始计数后，在计数到闹钟寄存器值加 1（RTC_ALR+1）之前的时钟周期中，配置 RTC_ALARM 与闹钟标志 ALRF。

对 RTC 闹钟的写操作必须使用下述过程之一与 RTC 秒标志同步：

- 使用 RTC 闹钟中断，并在中断处理程序中修改 RTC 闹钟和/或 RTC 计数器。
- 等待 RTC 控制寄存器中的 SECF 位被设置，再更改 RTC 闹钟和/或 RTC 计数器。

图 23-2 RTC 秒和闹钟波形图示例，PR=0003，ALARM=00004

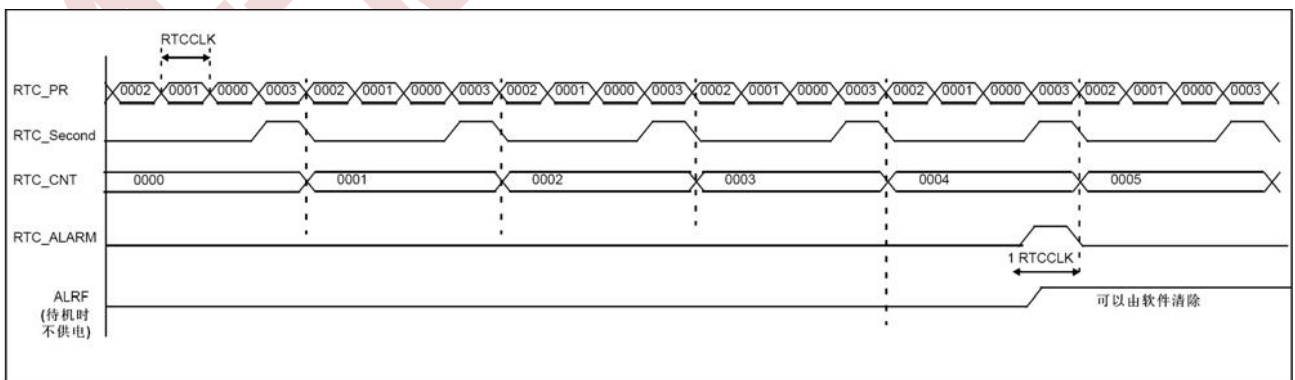
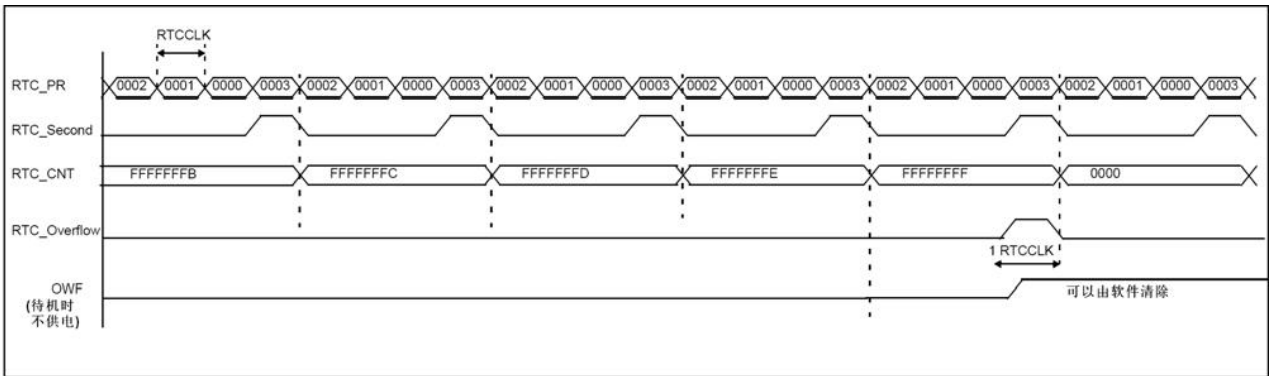


图 23-3 RTC 溢出波形图示例，PR=0003



21.3 RTC 寄存器

关于寄存器描述中的缩略词，请参考 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

21.3.1 RTC 寄存器概览

RTC 寄存器是 16 位可寻址寄存器，具体描述如下：

表 23-1 RTC 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	RTC_CRH	RTC 控制寄存器高位	0x0000
0x04	RTC_CRL	RTC 控制寄存器低位	0x0020
0x08	RTC_PRLH	RTC 预分频装载寄存器高位	0x0000
0x0C	RTC_PRLH	RTC 预分频装载寄存器低位	0x8000
0x10	RTC_DIVH	RTC 预分频器余数寄存器高位	0x0000
0x14	RTC_DIVL	RTC 预分频器余数寄存器低位	0x8000
0x18	RTC_CNTH	RTC 计数器寄存器高位	0x0000
0x1C	RTC_CNTL	RTC 计数器寄存器低位	0x0000
0x20	RTC_ALRH	RTC 闹钟寄存器高位	0xFFFF
0x24	RTC_ALRL	RTC 闹钟寄存器低位	0xFFFF

	1:32 位可编程计数器溢出。
位 5	<p>RTOFF: RTC 操作关闭 (RTC operation OFF)</p> <p>RTC 模块利用这位来指示对其寄存器进行的最后一次操作的状态, 指示操作是否完成。若此位为'0', 则表示无法对任何的 RTC 寄存器进行写操作。此位为只读位。</p> <p>0: 上一次对 RTC 寄存器的写操作仍在进行;</p> <p>1: 上一次对 RTC 寄存器的写操作已经完成。</p>
位 4	<p>CNF: 配置标志 (Configuration flag)</p> <p>此位必须由软件置'1'以进入配置模式, 从而允许向 RTC_CNT、RTC_ALR 或 RTC_PRL 寄存器 写入数据。只有当此位在被置'1'并重新由软件清'0'后, 才会执行写操作。</p> <p>0: 退出配置模式 (开始更新 RTC 寄存器);</p> <p>1: 进入配置模式。</p>
位 3	<p>RSF: 寄存器同步标志 (Registers synchronized flag)</p> <p>每当 RTC_CNT 寄存器和 RTC_DIV 寄存器由软件更新或清'0'时, 此位由硬件置'1'。在 APB1 复位后, 或 APB1 时钟停止后, 此位必须由软件清'0'。要进行任何的读操作之前, 用户程序必须等待这位被硬件置'1', 以确保 RTC_CNT、RTC_ALR 或 RTC_PRL 已经被同步。</p> <p>0: 寄存器尚未被同步;</p> <p>1: 寄存器已经被同步。</p>
位 2	<p>OWF: 溢出标志 (Overflow flag)</p> <p>当 32 位可编程计数器溢出时, 此位由硬件置'1'。如果 RTC_CRH 寄存器中 OWIE=1, 则产生中 断。此位只能由软件清'0'。对此位写'1'是无效的。</p> <p>0: 无溢出;</p> <p>1:32 位可编程计数器溢出。</p>
位 1	<p>ALRF: 闹钟标志 (Alarm flag)</p> <p>当 32 位可编程计数器达到 RTC_ALR 寄存器所设置的预定值, 此位由硬件置'1'。如果 RTC_CRH 寄存器中 ALRIE=1, 则产生中断。此位只能由软件清'0'。对此位写'1'是无效的。</p> <p>0: 无闹钟;</p> <p>1: 有闹钟。</p>
位 0	<p>SECF: 秒标志 (Second flag)</p> <p>当 32 位可编程预分频器溢出时, 此位由硬件置'1'同时 RTC 计数器加 1。因此, 此标志为分辨率可 编程的 RTC 计数器提供一个周期性的信号 (通常为 1 秒)。如果 RTC_CRH 寄存器中 SECIE=1, 则产生中断。此位只能由软件清除。对此位写'1'是无效的。</p> <p>0: 秒标志条件不成立;</p> <p>1: 秒标志条件成立。</p>

RTC 的功能由这个控制寄存器控制。当前一个写操作还未完成时 (RTOFF=0 时, 详见 23.3.5 节), 不能写 RTC_CR 寄存器。

注: 1 任何标志位都将保持挂起状态, 直到适当的 RTC_CR 请求位被软件复位, 表示所请求的中断已经被接受。

2 在复位时禁止所有中断, 无挂起的中断请求, 可以对 RTC 寄存器进行写操作。

3 当 APB1 时钟不运行时, OWF、ALRF、SECF 和 RSF 位不被更新。

4 OWF、ALRF、SECF 和 RSF 位只能由硬件置位, 由软件来清零。

5 若 ALRF=1 且 ALRIE=1, 则允许产生 RTC 全局中断。如果在 EXTI 控制器中允许产生 EXTI 线 17

中断，则允许产生 RTC 全局中断和 RTC 闹钟中断。

若 $ALRF=1$ ，如果在 EXTI 控制器中设置了 EXTI 线 17 的中断模式，则允许产生 RTC 闹钟中断；如果在 EXTI 控制器中设置了 EXTI 线 17 的事件模式，则这条线上会产生一个脉冲（不会产生 RTC 闹钟中断）。

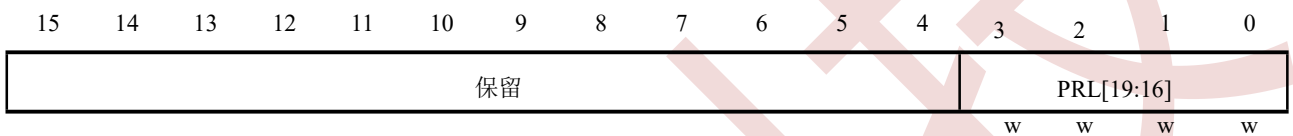
21.4.4 RTC 预分频装载寄存器（RTC_PRLH/RTC_PRL）

预分频装载寄存器用来保存 RTC 预分频器的周期计数值。它们受 RTC_CR 寄存器的 RTOFF 位保护，仅当 RTOFF 值为 '1' 时允许进行写操作。

RTC 预分频装载寄存器高位（RTC_PRLH）

偏移地址：0x08

只写复位值：0x0000

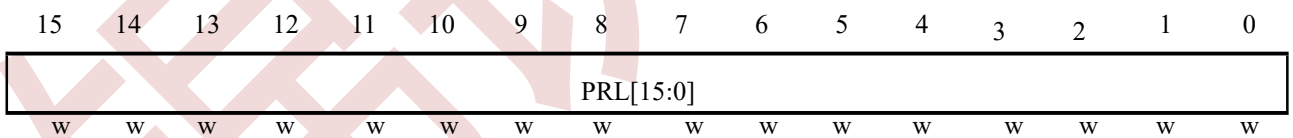


位 15:6	保留，被硬件强制为 0。
位 3:0	<p>PRL[19:16]: RTC 预分频装载值高位 (RTC prescaler reload value high)</p> <p>根据以下公式，这些位用来定义计数器的时钟频率：</p> $f_{TR_CLK} = f_{RTCCLK} / (PRL[19:0] + 1)$ <p>注：不推荐使用 0 值，否则无法正确的产生 RTC 中断和标志位。</p>

RTC 预分频装载寄存器低位（RTC_PRL）

偏移地址：0x0C

复位值：0x8000



位 15:0	<p>PRL[15:0]: RTC 预分频装载值低位</p> <p>根据以下公式，这些位用来定义计数器的时钟频率：</p> $f_{TR_CLK} = f_{RTCCLK} / (PRL[19:0] + 1)$
--------	---

注：如果输入时钟频率是 32.768kHz (f_{RTCCLK})，这个寄存器中写入 7FFFh 可获得周期为 1 秒钟的信号。

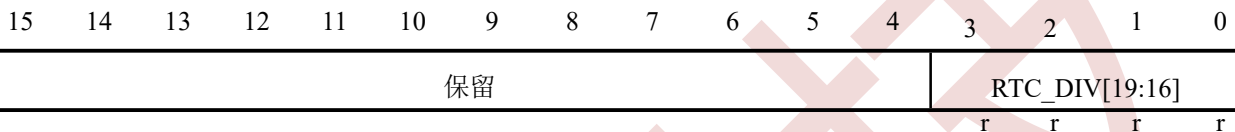
21.3.5 RTC 预分频器余数寄存器 (RTC_DIVH / RTC_DIVL)

在 TR_CLK 的每个周期里,RTC 预分频器中计数器的值都会被重新设置为 RTC_PRL 寄存器的值。用户可通过读取 RTC_DIV 寄存器,以获得预分频计数器的当前值,而不停止分频计数器的工作,从而获得精确的时间测量。此寄存器是只读寄存器,其值在 RTC_PRL 或 RTC_CNT 寄存器中的值发生改变后,由硬件重新装载。

RTC 预分频器余数寄存器高位 (RTC_DIVH)

偏移地址: 0x10

复位值: 0x0000

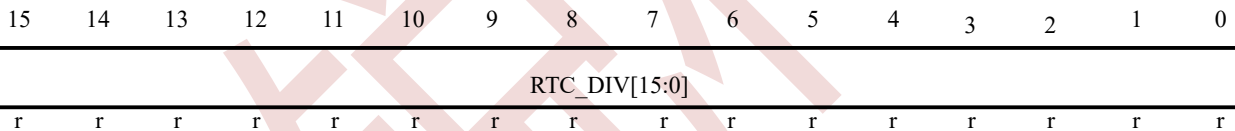


位 15:4	保留
位 3:0	RTC_DIV[19:16]: RTC 时钟分频器余数高位 (RTC clock divider high)

RTC 预分频器余数寄存器低位 (RTC_DIVL)

偏移地址: 0x14

复位值: 0x8000



位 15:0	RTC_DIV[15:0]: RTC 时钟器余数低位 (RTC clock divider low)
--------	--

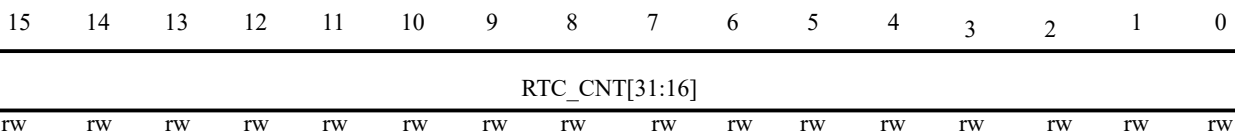
21.3.6 RTC 计数器寄存器 (RTC_CNTH / RTC_CNTL)

RTC 核有一个 32 位可编程的计数器,可通过两个 16 位的寄存器访问。计数器以预分频器产生的 TR_CLK 时间基准为参考进行计数。RTC_CNT 寄存器用来存放计数器的计数值。他们受 RTC_CR 的位 RTOFF 写保护,仅当 RTOFF 值为 '1' 时,允许写操作。在高或低寄存器 (RTC_CNTH 或 RTC_CNTL) 上的写操作,能够直接装载到相应的可编程计数器,并且重新装载 RTC 预分频器。当进行读操作时,直接返回计数器内的计数值 (系统时间)。

RTC 计数器寄存器高位 (RTC_CNTH)

偏移地址: 0x18

复位值: 0x0000

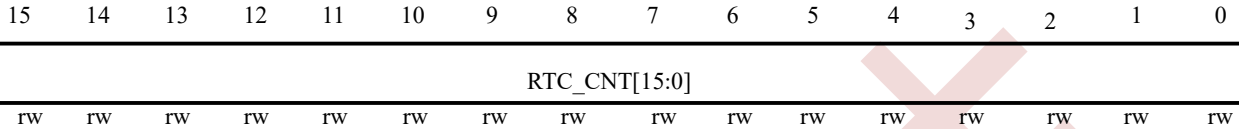


位 15:0	<p>RTC_CNT[31:16]: RTC 计数器高位 (RTC counter high)</p> <p>可通过读 RTC_CNTH 寄存器来获得 RTC 计数器当前值的高位部分。要对此寄存器进行写操作前，必须先进入配置模式。</p>
--------	--

RTC 计数器寄存器低位 (RTC_CNTL)

偏移地址: 0x1C

复位值: 0x0000



位 15:0	<p>RTC_CNTL[15:0]: RTC 计数器低位。</p> <p>可通过读 RTC_CNTL 寄存器来获得 RTC 计数器当前值的低位部分。要对此寄存器进行写操作，必须先进入配置模式。</p>
--------	---

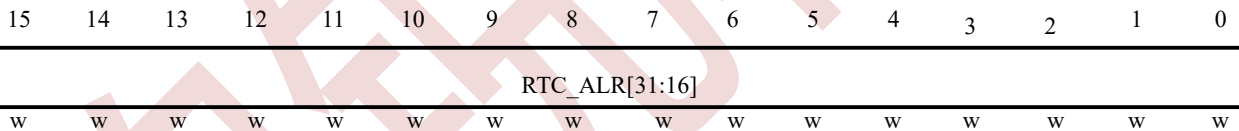
21.3.7 RTC 闹钟寄存器 (RTC_ALRH/RTC_ALRL)

当可编程计数器的值与 RTC_ALR 中的 32 位值相等时，即触发一个闹钟事件，并且产生 RTC 闹钟中断。此寄存器受 RTC_CR 寄存器里的 RTOFF 位写保护，仅当 RTOFF 值为'1'时，允许写操作。

RTC 闹钟寄存器高位 (RTC_ALRH)

偏移地址: 0x20

复位值: 0xFFFF

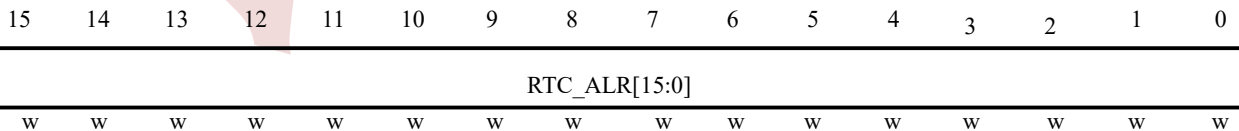


位 15:0	<p>RTC_ALRH[31:16]: RTC 闹钟值高位 (RTC alarm high)</p> <p>此寄存器用来保存由软件写入的闹钟时间的高位部分。要对此寄存器进行写操作，必须先进入配置模式 (参见 23.3.5 节)。</p>
--------	---

RTC 闹钟寄存器低位 (RTC_ALRL)

偏移地址: 0x24

复位值: 0xFFFF



位 15:0	<p>RTC_ALRL[15:0]: RTC 闹钟值低位 (RTC alarm low)</p> <p>此寄存器用来保存由软件写入的闹钟时间的低位部分。要对此寄存器进行写操作，必须先进入配置模式 (参见 23.3.5 节)。</p>
--------	---

22 模/数字转换器（ADC）

22.1 功能简介

模拟/数字转换器是 12 位逐次逼近寄存器型（SAR）ADC，最多提供多达 28 个输入通道。ADC 时钟源是系统时钟，可设置时钟预分频。ADC 有多种参考电压源可选，其中选择内部电压为参考电压时可用于检测芯片供电电压。ADC 选择内部电压为参考电压时有自动校正功能，避免芯片一致性问题。

22.2 主要特性

- 12 位的分辨率
- 最多提供多达 28+1 个输入通道
- 支持 ADC 中断
- 可设置 ADC 时钟预分频
- 多种参考电压可选：内部参考电压、外部参考、VDD。
- 支持 VDD 和参考地电压的测量
- 选择内部参考电压时，支持自动数据校正功能
- 选择内部 ADC 通道采用内部参考电压时，可测量 VDD 电压
- 输入电压范围：VSS ≤ VIN ≤ VDD。

22.3 结构框图

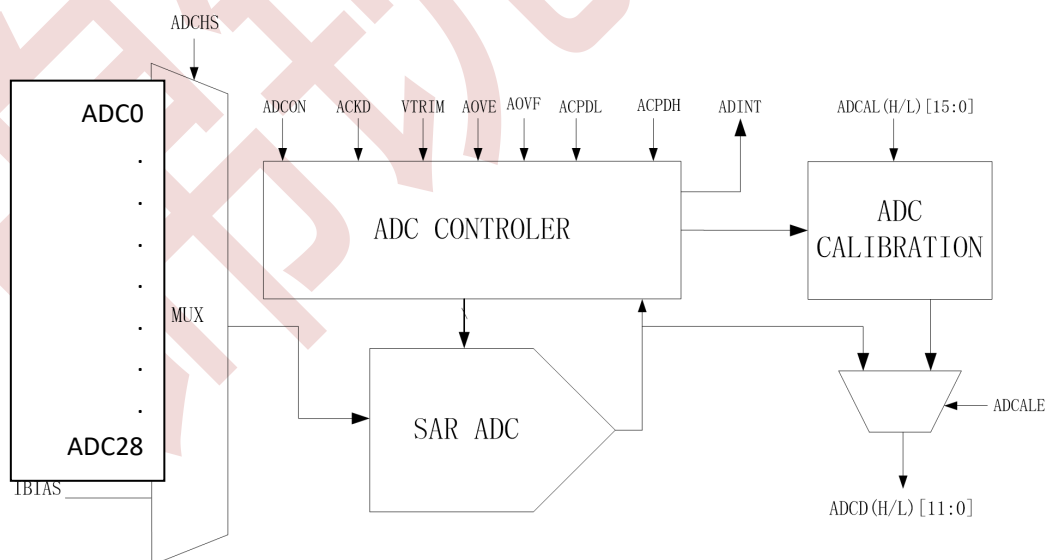


图 20-3-1 ADC 结构示意图

22.4 功能描述

- 支持单通道转换。
- 支持多通道转换，一次可最多转换 6 个通道。

ADC 的启动通过 AST 位使能，设置 AST=1 后，ADC 模块对 ADCHS 选择的输入电压源进行模/数转换。ADC 可通过 ACKD 设置时钟预分频，由系统时钟预分频后的时钟作为 ADC 转换时钟。在 ADC 时钟不变的条件下，ADC 的单次转换时间是由 HTME 设置的，转换时间为 $(13+2^{HTME})$ 个 ADC 时钟周期。当转换结束后，12 位的 A/D 值会被加载到寄存器 ADCDH、ADCDL，转换完后的 2.5 个时钟周期，AST 位自动清 0，同时中断标志 ADIF 位会置 1，如果 ADC 中断使能，会产生 ADC 中断。图 20-4-1 为 ADC 的转换时序图。

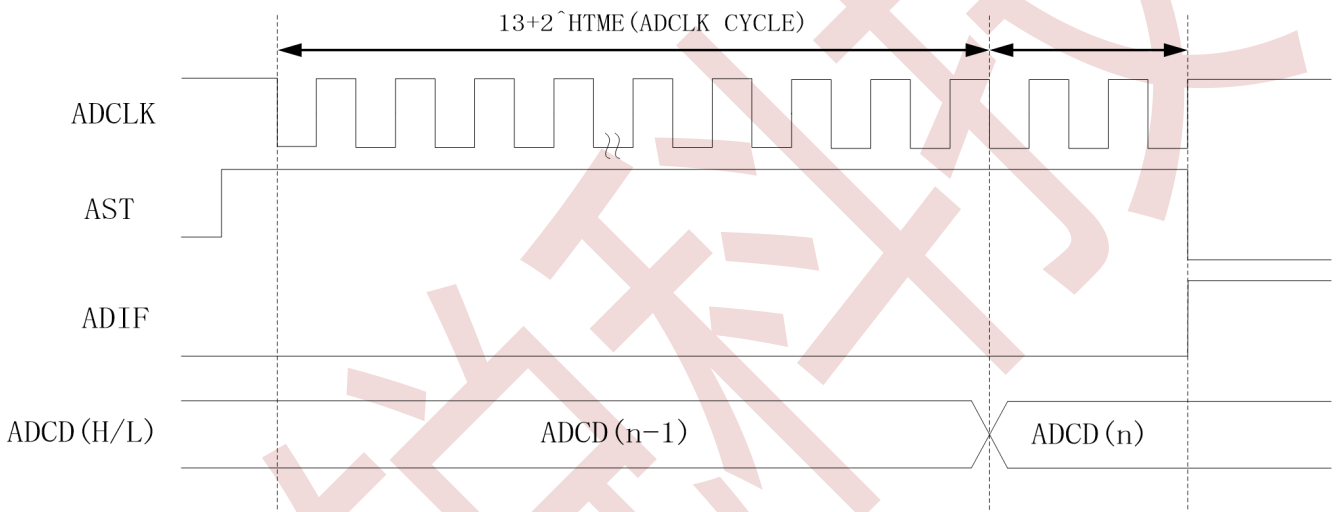


图 20-4-1 ADC 时序示意图

● ADC 数据校正

当选择内部 1.5V 作为参考电压时，由于芯片的离散性，每个芯片的内部电压不一定完全相同，导致每个芯片的 ADC 转换结果也有一定的偏差，所以在 ADC 转换完成后，有必要对 AD 值进行校正。芯片在出厂时，会对每个芯片的内部电压进行测试，得出与内部电压成反比例的校正系数，在芯片上电启动时，自动将此校正系数加载到寄存器 ADCALL、ADCALH，当 ADC 转换完成后自动将 AD 值根据校正寄存器 ADCALL、ADCALH 的值进行等比例换算，得出准确的 AD 值，最终的 AD 值也是存放在寄存器 ADCD 中。此功能通过 ADCALE 使能，对于用户来说，在应用时只需要设置 ADCALE=1 即可，校正过程是自动完成的。

22.5 ADC 寄存器概览

表 15-5 ADC 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x18	ADCFG	配置寄存器	0x0000
0x1C	ADCCAL	校准寄存器	0xxxx

0x20	ADCPDL	比较模式阈值下限值设定寄存器	0x0000
0x24	ADCPDH	比较模式阈值上限值设定寄存器	0x0000
0x28	ADCDALAT0	数据寄存器	0x0000
0x2C	ADCTRL	控制寄存器	0x0000
0x34	ADCSEQ		0x0000
0x38	ADCCHSEL		0x8000
0x3C	ADCDALAT0	扫描模式第 0 通道数据	0x0000
0x40	ADCDALAT1	扫描模式第 1 通道数据	0x0000
0x44	ADCDALAT2	扫描模式第 2 通道数据	0x0000
0x48	ADCDALAT3	扫描模式第 3 通道数据	0x0000
0x4c	ADCDALAT4	扫描模式第 4 通道数据	0x0000

22.5.1 ADC 控制信号寄存器 (ADCFG)

偏移地址: 0x18

复位值: 0x00000000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	AOVF	AOVE	ADCH[4:0]				ADCIF	ADIE	AST/ADC_	HTIME			VSEL		
rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

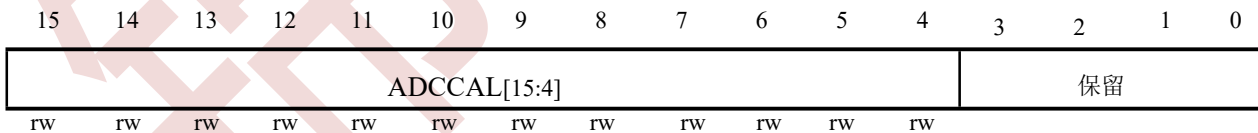
位 15	DMAEN:DMA 使能位, 1 有效
位 14	AOVF: 比较模式溢出标志位
位 13	AOVE: 比较模式使能位, 1 有效
位 12:8	ADCH: 通道选择信号: 为 1 时选择该通道, 为 0 时通道不选择, 一次只能选择一个通道 (由数字控制)。 ADC_CH<26:0>选择来自 PAD 的信号 ADC_CH<27>VDD/4 连续采样时, 必须第 1 个通道的值由 MCU 程序写入
位 7	ADCIF:ADC 中断采样完成标志位, 写 1 清 0 当 ADC 发生了采样完成事件, 该位被置'1'。在该位中写入'1'可以清除它,
位 6	ADIE:ADC 中断使能位, 1 有效

位 5	AST/ADC_EN:ADC 转换开始控制位，写 1 启动转换，转换后硬件自动清 0 连续扫描通道时，必须先配置好连续扫描通道个数及选择通道位置
位 4:2	HTIME: 采样时间控制位： 111: 采样时间为 128 个 ADC_CLK 110: 采样时间为 64 个 ADC_CLK 101: 采样时间为 32 个 ADC_CLK 100: 采样时间为 16 个 ADC_CLK 011: 采样时间为 8 个 ADC_CLK 010: 采样时间为 4 个 ADC_CLK 001: 采样时间为 2 个 ADC_CLK 000: 采样时间为 1 个 ADC_CLK
位 1:0	VSEL: 参考电压选择位： 00: 内部 1.5V 参考电压 01: 外部电源 VDD 作为参考电压 10: 外部参考电压 11: 内部 1.5V 参考电压

22.5.2 ADC 校准寄存器 (ADCCAL)

地址偏移: 0x1C

复位值: 0x0000 8000

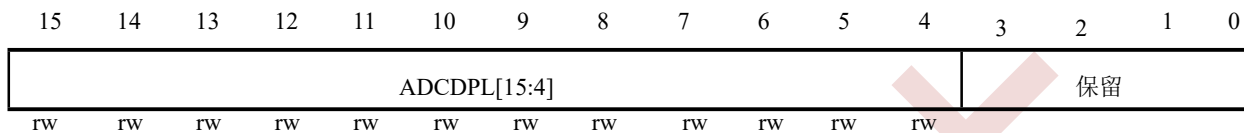


位 15:4	ADCCAL: ADC 校准寄存器，只有 ADCALE=1 并且选择参考电压为内部 1.5V 才有效。有效时，ADC 的输出按照如下公式： $ADCDL = (ADC \text{ 转换结果} * ADCCAL) / 32768$
位 3:0	保留

22.5.3 ADC 比较模式阈值下限值设定寄存器 (ADC DPL)

地址偏移: 0x20

复位值: 0x0000

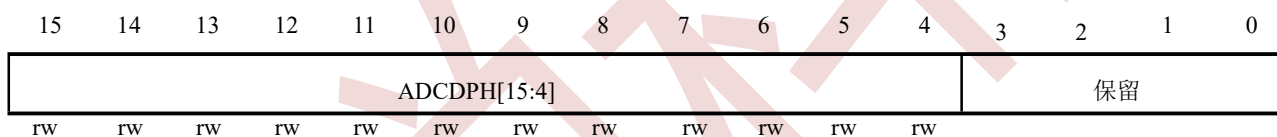


位 15:4	ADC DPL: 比较模式阈值下限值设定寄存器
位 3:0	保留

22.5.4 ADC 比较模式阈值上限值设定寄存器 (ADC DPH)

地址偏移: 0x24

复位值: 0x0000

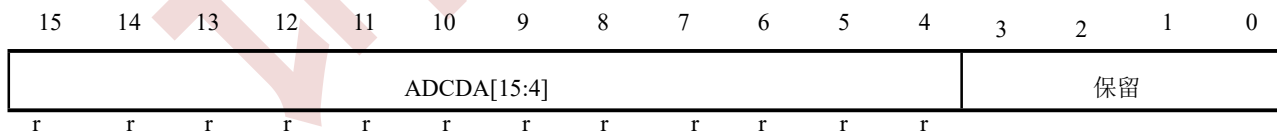


位 15:4	ADC DPH: 比较模式阈值上限值设定寄存器
位 3:0	保留

22.5.5 ADC 转换值寄存器 (ADC DA)

地址偏移: 0x28

复位值: 0x0000



位 15:4	ADC DA: ADC 转换值 连续采样时, 该寄存器保存为最后一个通道的数据
位 3:0	保留

22.5.6 ADC 控制寄存器 (ADCTRL)

地址偏移: 0x2C

复位值: 0x0000

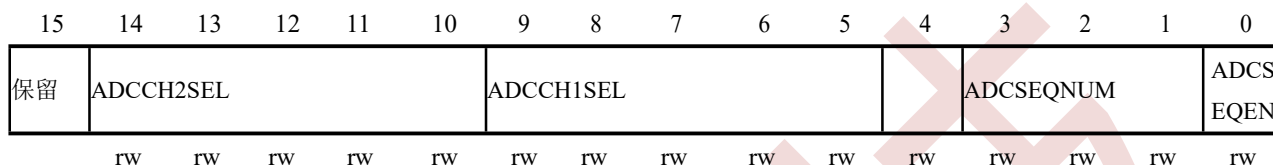
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ADC ALE	GAIN		AMP_ EN	ADC_DIV[9:4]						ADC 1ST	ADC_TRGSEL[2:0]			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	保留
位 14	ADC 校准使能位, 1 有效; 此位只有选择参考电压为内部 1.5V 时才有效, 当 ADCALE=1, ADC 的转换结果将根据 ADCAL 寄存器的数值进行校准。具体参考寄存器 ADCAL 说明
位 13:11	AMP 增益控制位: 000:2 倍 001:4 倍 010:8 倍 011:16 倍 其他: 32 倍
位 10	运放使能信号, 1 使能, 0 不使能
位 9:4	ADC DIV: ADC 时钟分频 0: PCLK 时钟 2 分频 1: PCLK 时钟 4 分频 2: PCLK 时钟 6 分频 3: PCLK 时钟 8 分频 ... 63: PCLK 时钟 128 分频
位 3	ADC 模拟部分开启, 1 表示开启, 0 表示关闭
位 2:0	ADC_TRGSEL: ADC 触发启动选择寄存器: 0: 关闭触发选项 1: timer1 ch4 高电平触发 2: timer14 ch1 高电平触发 3: timer1 ch3 高电平触发 4: timer1 ch2 高电平触发 5: timer1 ch1 高电平触发 6: exti4 高电平触发 7: exti15 高电平触发

22.5.7 ADCSEQ

偏移地址：0x34

复位值：0x00000000

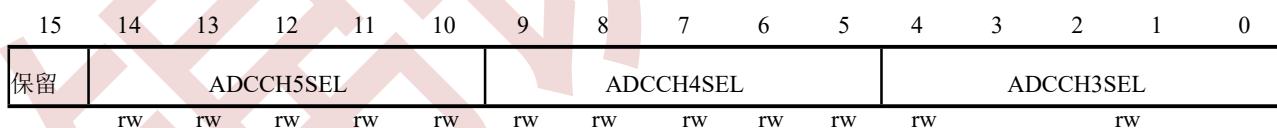


位 15	保留
位 14:10	ADC 连续扫描通道 3 位置选择
位 9:5	ADC 连续扫描通道 2 位置选择
位 4	
位 3:1	ADC 连续扫描通道个数，最多支持 6 个通道连续扫描，0 表示 1 个通道，5 表示 6 个通道，
位 0	ADC 连续扫描模式开启

22.5.8 ADCCHSEL

地址偏移：0x38

复位值：0x0000 0000

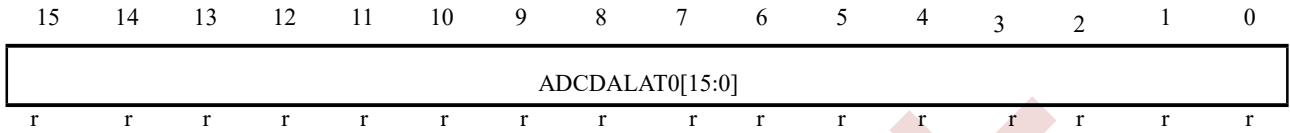


位 15	保留
位 14:10	ADC 连续扫描通道 5 位置选择
位 9:5	ADC 连续扫描通道 4 位置选择
位 4:0	ADC 连续扫描通道 3 位置选择

22.5.9 ADCDALAT0

地址偏移: 0x3C

复位值: 0x0000

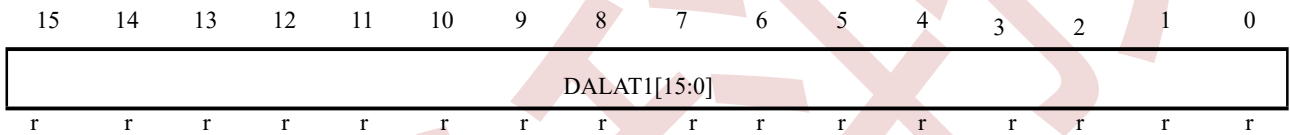


位 15:0	ADC 连续扫描通道 0 数据锁存
--------	-------------------

22.5.10 ADCDALAT1

地址偏移: 0x40

复位值: 0x0000

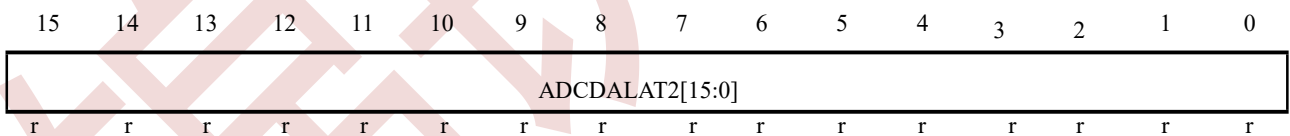


位 15:0	ADC 连续扫描通道 1 数据锁存
--------	-------------------

22.5.11 ADCDALAT2

地址偏移: 0x44

复位值: 0x0000

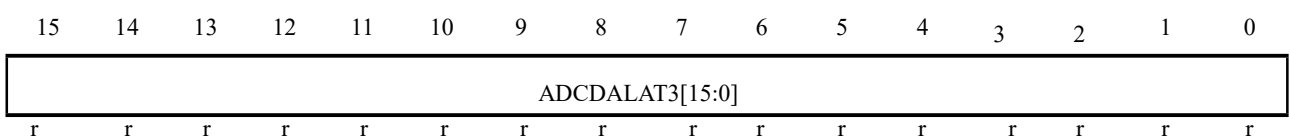


位 15:0	ADC 连续扫描通道 2 数据锁存
--------	-------------------

22.5.12 ADCDALAT3

地址偏移: 0x48

复位值: 0x0000

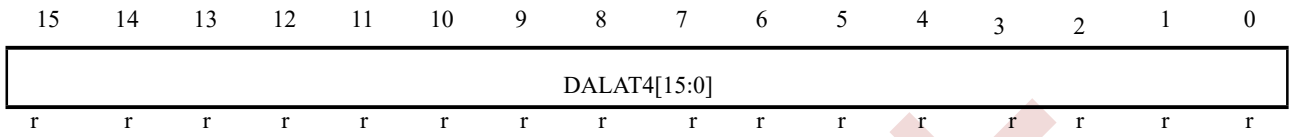


位 15:0	ADC 连续扫描通道 3 数据锁存
--------	-------------------

22.5.13 ADCDALAT4

地址偏移: 0x4C

复位值: 0x0000



位 15:0	ADC 连续扫描通道 4 数据锁存
--------	-------------------

23 模拟比较器 (COMP)

23.1 模拟比较器简介

CA1020 内置 2 个模拟比较器，每个比较器完全独立，它们不共享任何资源。

- ◆ 比较器可用于多种功能：
 - 支持比较器中断功能；
 - 支持 1V~VDD, 64 级电压阈值设置；
 - 支持比较器模块内接 ADC 测量功能

23.2 比较器寄存器

地址偏移	寄存器缩写	寄存器名	复位值
0x60	CMP0	COMP0 配置寄存器	0x0000
0x64	CMP1	COMP1 配置寄存器	0x0000

23.2.1 COMP0 配置寄存器 (CMP0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COO UT	保留			CMP0 FLG	CMP 0IE	TRIM0						SELP 0	COHY EN	CMP 0EN	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15	比较器输出结果
位 14:11	保留
位 10	cmp0 中断标志位，写 1 清 0
位 9	cmp0 中断使能
位 8:3	INN 参考电压调整位 当 TRIM<5:0>=000000 时，INN0=0.03125V。 在 000000~111111 范围内，IINN0 以 0.03125V 步距递增，共 64 档。 当 TRIM0<5:0>=111111 时， INN=2V.
位 2	VP 参考电压调整位 0:INP0 1:INP1
位 1	为 0 时，CMP 下降沿表示产生 CMP 中断，为 1 时，表示 CMP 上升沿产生 CMP 中断
位 0	比较器使能信号，1 使能，0 不使能

23.2.2 COMP1 配置寄存器 (CMP1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1OUT	保留			CMP1FLG	CMP1IE	TRIM1						SELP1	C1HYEN	CMP1EN	
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 15	比较器输出结果
位 14:11	保留
位 10	Cmp1 中断标志位, 写 1 清 0
位 9	Cmp1 中断使能
位 8:3	INN 参考电压调整位 当 TRIM<5:0>=000000 时, INN0=0.03125V。 在 000000~111111 范围内, IINN0 以 0.03125V 步距递增, 共 64 档。 当 TRIM0<5:0>=111111 时, INN=2V.
位 2	VP 参考电压调整位 0:INP0 1:INP1
位 1	为 0 时, CMP 下降沿表示产生 CMP 中断, 为 1 时, 表示 CMP 上升沿产生 CMP 中断
位 0	比较器使能信号, 1 使能, 0 不使能

24 LDO 线性低电压输出

24.1 LDO 简介

CA1020 内置 1 个 LDO 电压输出，可以给外部的 NOR SPI FLASH 以及 TFT 彩屏已经其他需要 3V 器件供电。LDO 在封装的第 2 引脚，可通过寄存器设置四个挡位的电压输出分别是：3.3V/3V/2.8V/2.6V。支持软件控制关闭电压输出功能。

24.2 LDO 寄存器

地址偏移	寄存器缩写	寄存器名	复位值
0x04	LDO	LDO 配置寄存器	

24.2.1 LDO 配置寄存器（LDO）

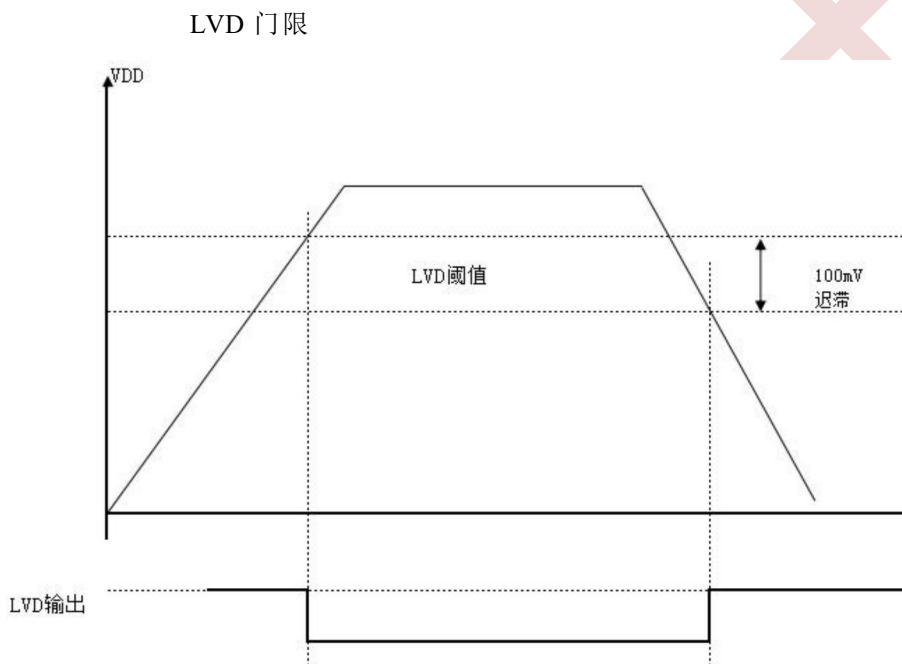
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MOD	LEVEL	EN	保留，这些位的值用户不可随意更改，必须保持为默认值											
rw	rw	rw	rw	rw											

位 15:14	保留
位 13	INN 参考电压调整位 当 TRIM<5:0>=000000 时，INN0=0.03125V。 在 000000~111111 范围内，IINN0 以 0.03125V 步距递增，共 64 档。 当 TRIM0<5:0>=111111 时， INN=2V.
位 12:11	LDO 输出电压档位 0: 2.6V 1: 2.8V 2: 3V 3: 3.3V（上电默认）
位 10	EN 1: LDO 对外输出开启 0: LDO 对外输出关闭
位 9:0	保留，这些位的值用户不可随意更改，必须保持为默认值

25 LVD 可编程电压监测器

25.1 LVD 简介

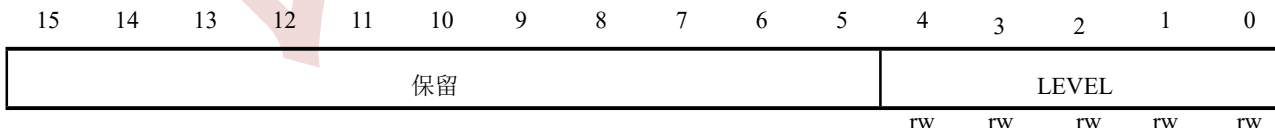
芯片内置 LVD 模块，1.8-4.8V 可按照每级 0.1V 设置，用户可利用 LVD 对 VDD 电压与 LVD 控制寄存器（CR）中的 LEVEL[4:0]位进行比较来监控电源，这几位选择监控电压的阈值。当 VDD 下降到 LVD 阈值以下阈值时，根据设置，就会产生 LVD 中断。例如，这一特性可用于执行紧急关闭任务。



25.2 LVD 寄存器

地址偏移	寄存器缩写	寄存器名	复位值
0x08	CR	LVD 控制配置寄存器	

25.2.1 LVD 控制寄存器 CR



位 15:5	保留
位 4:0	LVD 检测电平选择： $LVD_LEVEL_{<4:0}> * 0.1V + 1.7V$

26 触摸按键 (Touch Key)

26.1 功能简介

CA1020 系列芯片的触摸功能模块具有优越的性能，触摸模块最大可支持多达 16 个通道。针对有低功耗需求的应用，还设计了芯片在 STOP 模式时仍能正常工作的机制。

26.2 主要特性

- 最大支持 12 个通道
- 支持低功耗模式
- 支持触摸中断
- 支持充放电时钟预分频
- 支持手动和自动启动模式
- 比较器阈值有多级可选
- STOP 模式下可设自动唤醒阈值

● 26.3 结构图

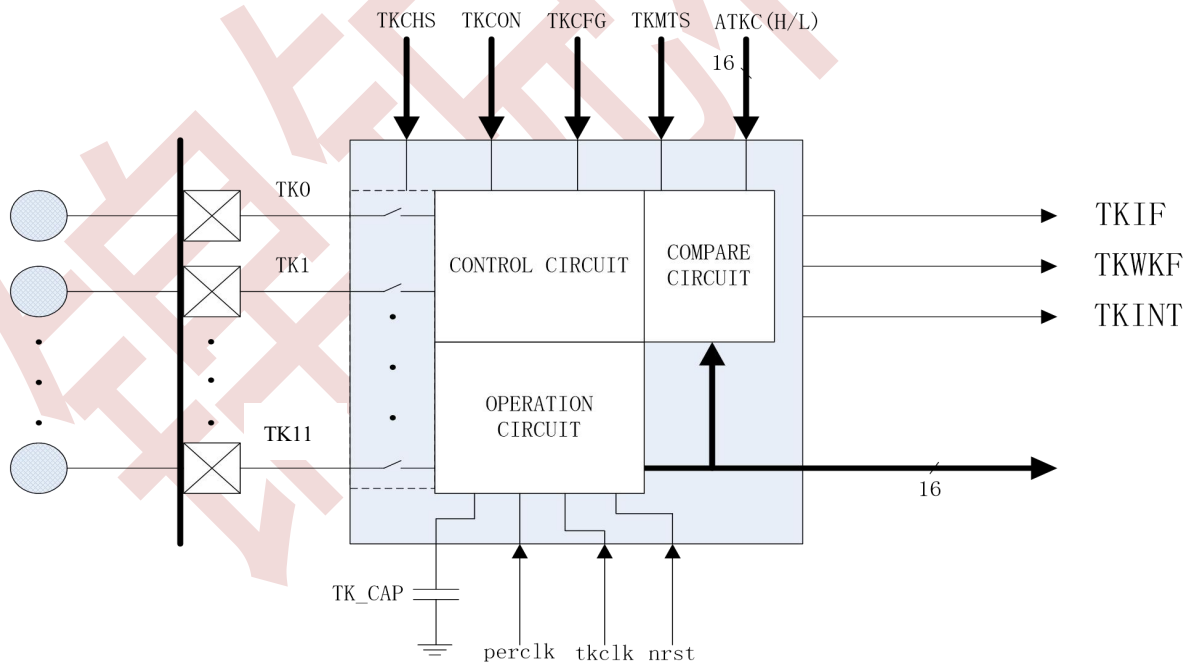


图 24-1 触摸模块结构图

26.4 功能描述

26.4.1 触摸时钟预分频

触摸控制器充放电的时钟频率对触摸的性能至关重要，当充放电频率太高时，有可能造成对触摸电极的充电不充分从而导致手指触摸时触摸数据变化量变小。触摸时钟预分频通过 TKDIV 进行设置，通过设置合理的值可以使触摸的性能更优。

26.4.2 低功耗模式

为了实现触摸功能的低功耗应用，触摸模块设计了相应的省电机理。在 STOP 模式下，只要触摸的充放电时钟源 LSI 处于开启状态，触摸模块就可以保持正常的充放电和计数。当触摸采集完成后，如果 TWKE=0，触摸采集完成中断会唤醒 CPU，软件在 CPU 唤醒之后可以读取触摸数据，然后再次进入 STOP 模式。另外，触摸模块还设计了触摸阈值自动比较功能，用户可通过阈值设置寄存器设置一组通道的触发阈值，在 STOP 模式，触摸控制器仍然可以将采集的触摸数据和阈值进行比较，当触摸数据超过阈值时，如果 TWKE=1，会产生阈值触发中断并唤醒 CPU，CPU 被唤醒后就可以进行正常的触摸采集和判断。

26.5 寄存器描述

26.5.1 寄存器概览

地址偏移	寄存器缩写	寄存器名	复位值
0x00	TKCON	TK 控制寄存器	0x00001F00
0x04	TKMTS	TK 定时器启动时间寄存器	0x00000000
0x08	TKCH01	TK 通道配置寄存器 01	0x00000000
0x0C	TKCH23	TK 通道配置寄存器 23	0x00000000
0x10	TKCH45	TK 通道配置寄存器 45	0x00000000
0x14	TKIF	TK 中断标识寄存器	0x00000000
0x18	TKMS0	TK 触摸采集数据寄存器 0	0x00000000
0x1c	TKMS1	TK 触摸采集数据寄存器 1	0x00000000
0x20	TKMS2	TK 触摸采集数据寄存器 2	0x00000000
0x24	TKMS3	TK 触摸采集数据寄存器 3	0x00000000
0x28	TKMS4	TK 触摸采集数据寄存器 4	0x00000000
0x2c	TKMS5	TK 触摸采集数据寄存器 5	0x00000000
0x30	MICCYC	咪头检测周期寄存器	0x00000000
0x34	MICDAT	咪头技术结果寄存器	0x00000000
0x38	MICTHR	咪头时钟沿阈值	0x00000000
0x3c	MICSEL	咪头时钟沿最大阈值	0x00000000

26.5.2 TK 控制寄存器 (TKCON)

地址偏移: 0x00

复位值: 0x0000 1F00



位 31:16	保留位，硬件强制为 0
位 15:13	<p>TKDIV: 触摸时钟分频</p> <p>000: 不分频</p> <p>001: 2 分频</p> <p>010: 3 分频</p> <p>011: 4 分频</p> <p>100: 5 分频</p> <p>101: 6 分频</p> <p>110: 7 分频</p> <p>111: 8 分频</p>
位 12:8	<p>TKTMS: 外挂调制电容放电时间</p> <p>放电时间 = TKTMS x 32 x 时钟周期</p> <p>在 TKDIV=0 (即 TK 时钟频率为 4MHz) 的条件下, 放电时间范围是: 32us - 992us</p> <p>备注: TKTMS 不能设置为 0。</p>
位 7	<p>TKST: 数据采集启动使能</p> <p>0: 禁用</p> <p>1: 启动</p> <p>在软件启动 (TMEN=0) 的情况下, 由硬件清 0;</p> <p>定时器启动 (TMEN=1) 的情况下需由软件清 0。</p>
位 6	<p>TKIE: TK 中断使能</p> <p>0: 禁用中断</p> <p>1: 使能中断</p>
位 5	<p>TMEN: TK 启动方式</p> <p>0: 通过 TKST 位置 1 启动</p> <p>1: 通过定时器启动</p> <p>注: 定时器启动需将 TKST 置 1 启动定时器</p>
位 4	咪头检测使能

	0: 表示不检测咪头信号 1: 表示检测咪头信号
位 3	触摸低速时钟选择, 1 表示选择
位 2:0	VRS: 比较器阈值电压基准 000: 阈值电压最高 111: 阈值电压最低

26.5.3 TK 定时器启动时间寄存器 (TKMTS)

地址偏移: 0x04

复位值: 0x0000 0000

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



位 15:14	TKPWC: 按键补偿 0: 不补偿 1: 补偿
位 13:12	VDS: 内部运放输出电压选择 00:2.0V 01:2.5V 10:3.0V 11:4.0V
位 11:10	VIRS: 内部基准电压选择 00:1.0V 01:1.5V 10:2.0V 11:2.5V
位 9	TKPWS: 充电电源选择 0: 外部电源当作充电电源 1: 内部运放当作充电电源
位 8	TKCVS: 充电基准电压选择 0: 外部基准电压 1: 内部基准电压

位 7:0	TKMTS: 定时器启动时间 启动时间=(TKMTS+1)x 32 x 低速时钟周期, 如果低速时钟频率为 96K, 时间范围是 0.333ms - 85.33ms。
-------	--

26.5.4 TK 通道配置寄存器 (TKCH01)

地址偏移: 0x08

复位值: 0x0000 0000



位 4~0	TK0PS: 通道 0 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能
位 12~8	TK1PS: 通道 1 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能

26.5.5 TK 通道配置寄存器 (TKCH23)

地址偏移: 0x0C

复位值: 0x0000 0000

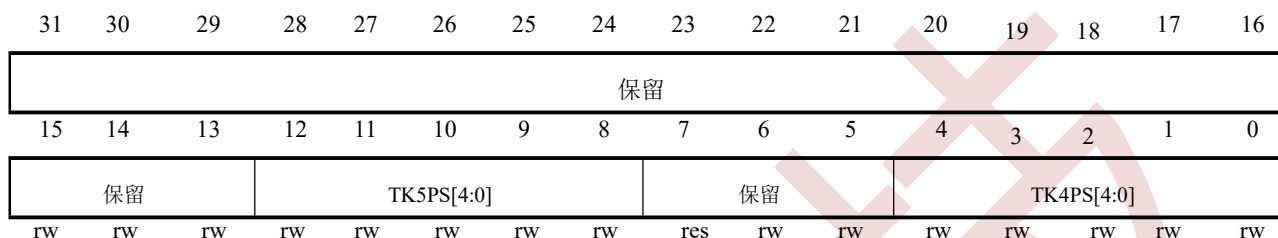


位 4~0	TK2PS: 通道 2 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能
位 12~8	TK3PS: 通道 3 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能

26.5.6 TK 通道配置寄存器 (TKCH45)

地址偏移: 0x10

复位值: 0x0000 0000

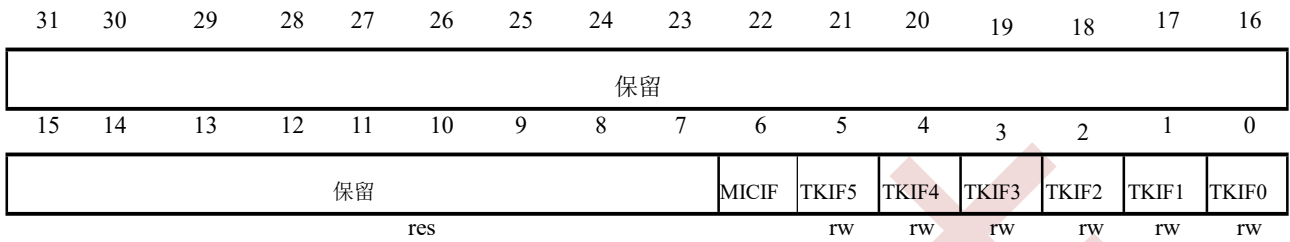


位 4~0	TK4PS: 通道 4 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能
位 12~8	TK5PS: 通道 5 选择 00000: TK0~TK11 关闭 00001: 选择 TK0 00010: 选择 TK1 00011: 选择 TK2 01011: 选择 TK11 10001: 选择内部参考电容 其余: 不能选择做 TK 功能

26.5.7 TK 中断标识寄存器 (TKIF)

地址偏移: 0x14

复位值: 0x0000 0000

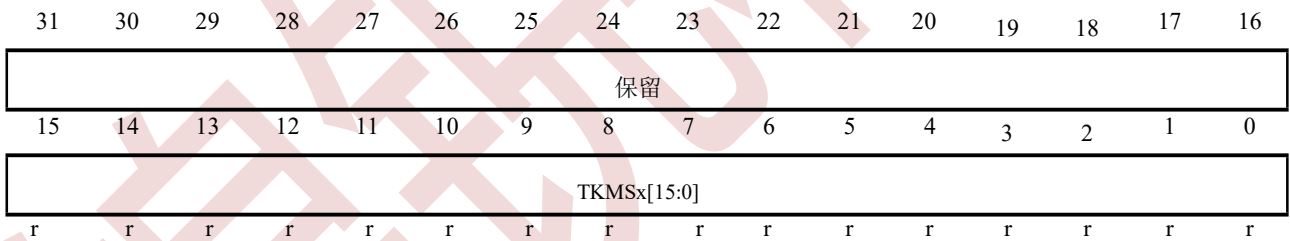


位 15:7	保留位, 硬件强制为 0
位 6	咪头唤醒中断标志位
位 5:0	TKIFx: 触摸采集中断标志位 按顺序分别对应 6 个选定通道, TKIFx 表示此通道触摸数据采样完成。 可由软件写 1 清零。

26.5.8 TK 触摸采集数据寄存器 x(TKMSx)(x=0,1,2,3,4,5)

地址偏移: 0x18+4x 通道编号

复位值: 0x0000 0000



位 31:16	保留位, 硬件强制为 0
位 15:0	TKMSx: 触摸采样数据

27 空气传感器

27.1 空气传感器介绍

芯片内置 1 个空气传感器模块，支持 4 路咪头空气开关检测功能，支持硅咪、正装（驻极体 MIC）和反装（背极体 MIC），支持咪头金属膜差异校正功能，支持感应电容可调功能，支持 PFM 模式，增强客户体验；支持中断唤醒 MCU 功能。

27.2 寄存器描述

27.2.1 传感器检测周期寄存器（MICCYC）

地址偏移：0x30

复位值：0x0000 0000



位 31:16	保留位，硬件强制为 0
位 15:0	设置检测咪头的时间周期：低速时钟周期 X (rtcCnt+1) ns

27.2.2 传感器计数结果寄存器（MICDAT）

地址偏移：0x34

复位值：0x0000 0000



位 31:16	保留位，硬件强制为 0
位 15:0	在规定低速时钟时间内检测咪头时钟沿个数

27.2.3 传感器时钟沿阈值 (MICTHR)

地址偏移: 0x38

复位值: 0x0000 0000



位 31:16	保留位, 硬件强制为 0
位 15:0	咪头时钟沿阈值

27.2.4 传感器时钟沿最大阈值 (MICSEL)

地址偏移: 0x3c

复位值: 0x0000 0000



位 15	咪头阈值比较方向: 0: 表示大于设置阈值产生中断或事件触发 1: 表示小于设置阈值产生中断或事件触发
位 14:4	保留
位 3:0	咪头通道选择, 0 表示不选择任何通道

28 程序下载和仿真介绍

28.1 程序下载

CA1020 系列芯片主要采用 2 线下载方式下载程序，芯片通过 SWD/TCK 接口与下载工具相连接，默认的升级接口为 PA0(SWD),PA1(TCK)。

更多关于程序下载步骤的细节请参考“CACHIP 开发下载工具使用说明”。

28.2 在线仿真

CA1020 系列芯片支持在线仿真，芯片与仿真器之间通过 SWD/TCK 接口进行通信，出厂默认的接口是 PA0 和 PA1。要注意的是，由于芯片与仿真器间通过 SWD/TCK 的管脚通信，所以与仿真器连接的引脚不能设置为其他功能，并且应用程序里不能使用 UART0 功能，否则将无法进入仿真模式。

更多关于仿真功能的细节可参考仿真器的相关文档介绍。

28.3 调试模式

在调试模式下（Cortex-M0 内核停止），IWDG 计数器将继续正常工作或停止，具体取决于调试模块中的 DBG_CTRL.IWDG_STOP 位。如果该位设置为“1”，则计数器停止。该位为“0”时，计数器正常工作。详细见3.3.2 外设调试支持章节。

29 电气特性

29.1 极限参数

参数	最小值	最大值	单位
直流供电电压	-0.3	5.5	V
I/O 引脚输入电压	-0.3	VDD+0.3	V
工作环境温度	-40	105	°C
储存温度	-55	125	°C
CPU 工作频率	-	72	MHz

备注：超过“极限参数”范围有可能对芯片造成损坏，无法预期芯片在上述范围外的工作状态，若长期在标示范围外工作，可能会影响芯片的可靠性。

29.2 直流电气特性

直流电气特性（TA=25°C，除非其它说明）

芯片参数	符号	工作电压	最小值	典型值	最大值	单位	测试条件
正常工作电流	Inor	VDD=3.3V	-	1.272	-	mA	系统时钟 16MHz flash=P25Q80H
		VDD=5V	-	1.281	-		系统时钟 16MHz flash=P25Q80H
		VDD=3.3V	-	3.730	-		系统时钟 64MHz flash=P25Q80H
		VDD=5V	-	3.792	-		系统时钟 64MHz flash=P25Q80H
Stop 模式电流	Istp	VDD=3.3V	-	18	-	uA	系统时钟 16MHz flash=P25Q80H flash 处于 Standby 模式
		VDD=5V	-	19	-		系统时钟 16MHz flash=P25Q80H flash 处于 Standby 模式
		VDD=3.3V	-	18	-		系统时钟 64MHz flash=P25Q80H flash 处于 Standby 模式
		VDD=5V	-	19	-		系统时钟 64MHz flash=P25Q80H flash 处于 Standby 模式
		VDD=3.3V	-	9	-		系统时钟 16MHz flash=P25Q80H flash 处于 Deep Power-down(DP) 模式
		VDD=5V	-	10	-		系统时钟 16MHz flash=P25Q80H flash 处于 Deep Power-down(DP) 模式
		VDD=3.3V	-	9	-		系统时钟 64MHz flash=P25Q80H flash 处于 Deep Power-down(DP) 模式
		VDD=5V	-	10	-		系统时钟 64MHz flash=P25Q80H

							flash 处于 Deep Power-down(DP)模式	
IO 端口输入高电压 (斯密特模式开 启)	V _{hi1}	VDD=3.3V	1.61		3.3	V	PA28 至 PB02	
			1.61		3.3		PA0 至 PA27	
		VDD=5V	1.61		3.3		PA28 至 PB02	
			2.35		5		PA0 至 PA27	
IO 端口输入高电压 (斯密特模式关 闭)	V _{hi2}	VDD=3.3V	-	0.5*VDD	3.3	V	PA28 至 PB02	
			-		3.3		PA0 至 PA27	
		VDD=5V	-	1.65	3.3		PA28 至 PB02	
			-	0.5*VDD	5		PA0 至 PA27	
IO 端口输入低电压 (斯密特模式开 启)	V _{lo1}	VDD=3.3V	0	-	1.28	V	PA28 至 PB02	
			0	-	1.28		PA0 至 PA27	
		VDD=5V	0	-	1.28		PA28 至 PB02	
			0	-	1.8		PA0 至 PA27	
IO 端口输入低电压 (斯密特模式关 闭)	V _{lo2}	VDD=3.3V	0	0.5*VDD	-	V	PA28 至 PB02	
			0		-		PA0 至 PA27	
		VDD=5V	0	1.65	-		PA28 至 PB02	
			0	0.5*VDD	-		PA0 至 PA27	
IO 端口灌电流	I _{pu}	VDD=3.3V	PA28 至 PB02 : 2mA@ DRVN=0 5mA@ DRVN=1 9mA@ DRVN=2 12mA@ DRVN=3 PA0 至 PA27: 2mA@ DRVN=0 5mA@ DRVN=1 9mA@ DRVN=2 12mA@ DRVN=3			-	mA	IO 设为推挽输出模式, 4 级驱 动能力可选 (DRVN), Vol=0.1*VDD
		VDD=5V	PA28 至 PB02 : 3mA@ DRVN=0 8mA@ DRVN=1 13mA@ DRVN=2 18mA@ DRVN=3 PA0 至 PA27: 4mA@ DRVN=0 10mA@ DRVN=1 18mA@ DRVN=2 24mA@ DRVN=3			-		
IO 端口推电流	I _{ol}	VDD=3.3V	PA28 至 PB02 : 1.5mA@ DRVP=0 2.4mA@ DRVP=1 3.2mA@ DRVP=2 3.6mA@ DRVP=3			-	mA	IO 设为推挽输出模式, 4 级驱 动能力 可选 (DRVP), Voh=0.9*VDD

				PA0 至 PA27: 1.8mA@ DRVP=0 3.6mA@ DRVP=1 6.0mA@ DRVP=2 8.0mA@ DRVP=3			
		VDD=5V		PA28 至 PB02 : 2.8mA@ DRVP=0 5.3mA@ DRVP=1 9.0mA@ DRVP=2 11mA@ DRVP=3 PA0 至 PA27: 4mA@ DRVP=0 8mA@ DRVP=1 12mA@ DRVP=2 16mA@ DRVP=3	-		
PA22~PA25 强推电流	I _{ol}	VDD=3.3V	-	PA22 至 PA25: 1.8mA@ DRVP=0 3.6mA@ DRVP=1 6.0mA@ DRVP=2 8.0mA@ DRVP=3 10mA@ DRVP=4 11mA@ DRVP=5 14mA@ DRVP=6 15mA@ DRVP=7	-	mA	IO 设为推挽输出模式， 8 级驱动能力 可选 (DRVP), V _{oh} =0.9*VDD
		VDD=5V	-	PA22 至 PA25: 4mA@ DRVP=0 8mA@ DRVP=1 12mA@ DRVP=2 16mA@ DRVP=3 20mA@ DRVP=4 23mA@ DRVP=5 28mA@ DRVP=6 30mA@ DRVP=7	-		
IO 端口下拉电阻	R _{d1}	强下拉	-	15	-	KΩ	
		弱下拉	-	45	-		
IO 端口上拉电阻	R _{u1}	强上拉	-	10	-		
		弱上拉	-	45	-		

说明：以上参数是随机抽取的典型芯片测试结果，仅供参考。不同型号的FLASH工作电流和低功耗模式的电流有差异。不同的芯片以及同一个芯片不同的IO口测量的上下拉电阻也有差异。

29.3 ESD/EFT 特性

符号	模式	条件	封装	最大值	单位
V _{ESD}	静电放电 (人体放电模式 HBM)	TA=+25°C	QFN48	±8000	V
	静电放电 (组件放电模式 CDM)			±1000	V
	静电放电 (机器放电模式 MM)			±400	V

符号	条件	封装	通过值	单位
EFT	F _{sys} = 64MHz / UART 通讯	QFN48	±4000	V

符号	条件	封装	通过值	单位
Latch UP	STEP:25mA	QFN48	±250	mA

29.4 内部 RC 时钟温度特性

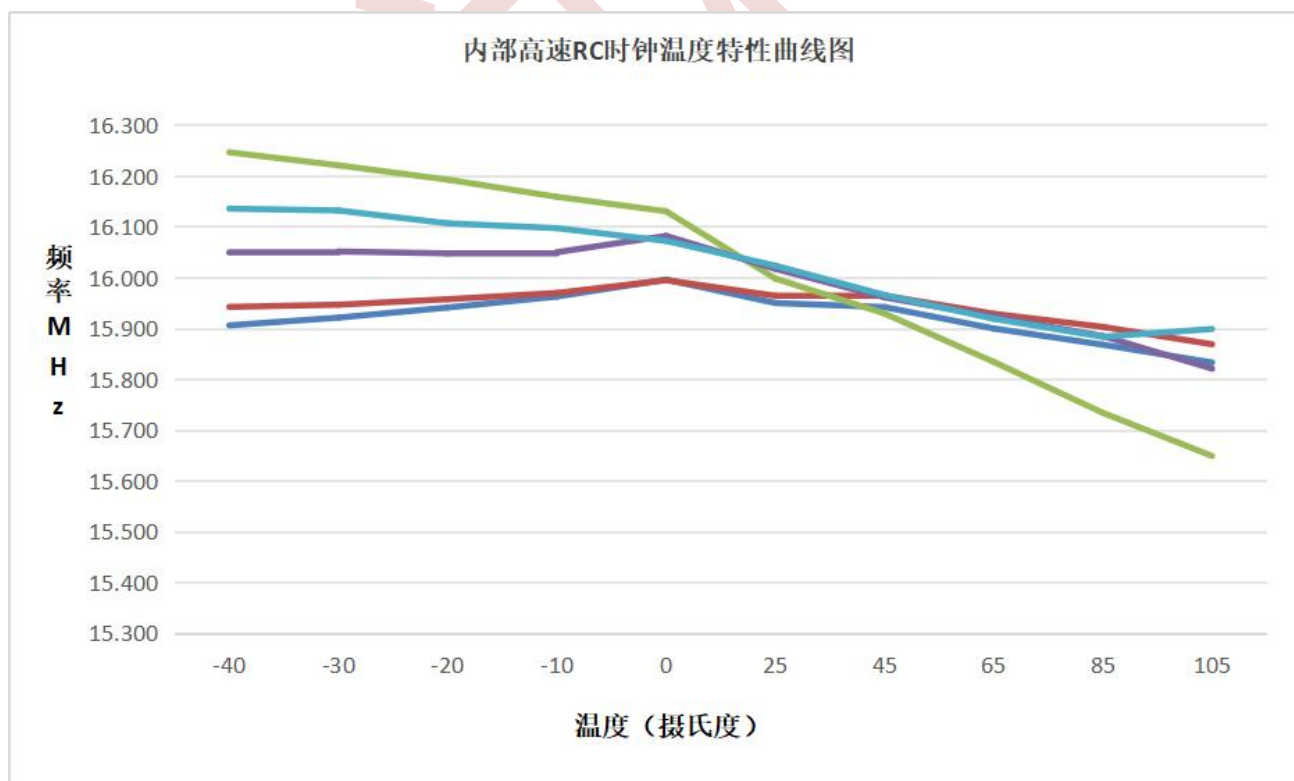


图 27-4-1 内部 RC 时钟温度特性曲线图

备注：以上图形数据为随机抽取部分芯片实测数据，数据仅供参考。

29.5 ADC 电气特性

模/数转换器 (ADC) 电气特性 (Ta=25°C, 参考电压为 VDD)

芯片参数	符号	最小值	典型值	最大值	单位	条件
工作电压	V _{AD}	1.8		5.5	V	
ADC 精度	NR		11		Bit	GND<=V _{in} <=V _{ref} , ADC 参考电压 VDD 或外部基准
			10		Bit	GND<=V _{in} <=V _{ref} , ADC 参考电压为内部基准
ADC 外部基准电压	V _{ex}	1.1	-	VDD	V	
ADC 输入电压	V _{in}	0	-	VDD	V	
ADC 输入电阻	R _{in}	2	-	-	MΩ	VDD=5V
ADC 转换电流	I _{ADC}	-	180	-	uA	VDD=5V
微分非线性误差	DNL	-	-	±3	LSB	VDD=5V
积分非线性误差	INL	-	-	±3	LSB	VDD=5V
满刻度误差	EF	-	±3	±4	LSB	VDD=5V
偏移量误差	E _z	-	±0.5	±1	LSB	VDD=5V
转换时间	T _{CON}	-	16	-	时钟周期	

备注: (1) ADC 输入电阻是直流条件下 ADC 自身的输入电阻;
 (2) 测试 ADC 时连接通路的信号源内阻需要小于 10K Ω

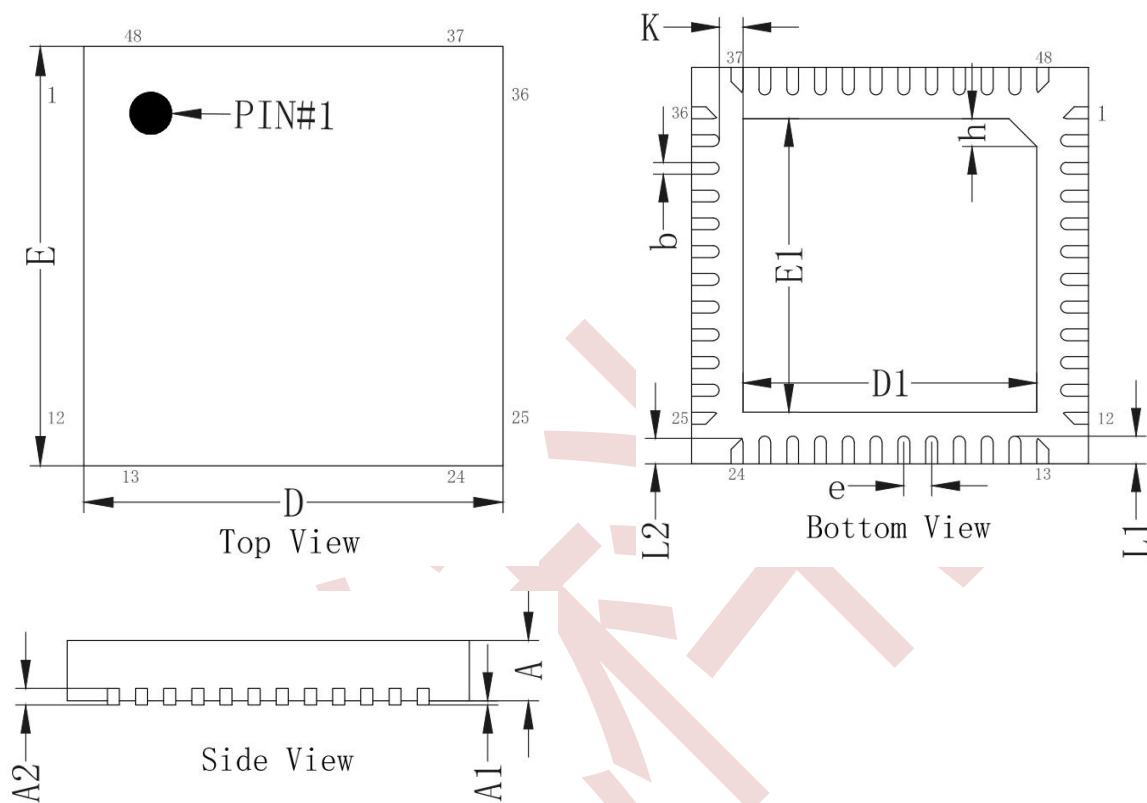
29.6 LDO 电压输出管脚 (3V_OUT) 特性

LDO 电压输出管脚 (3V_OUT) 负载能力测试 (滤波电容 10uF)				
空载电压 (V)	负载 (17 Ω)		负载 (7 Ω)	
	工作电压 (V)	工作电流 (mA)	工作电压 (V)	工作电流 (mA)
3.251	3.16	114	3.09	206
2.758	2.687	94	2.633	167
2.562	2.497	86	2.44	151

30 封装类型

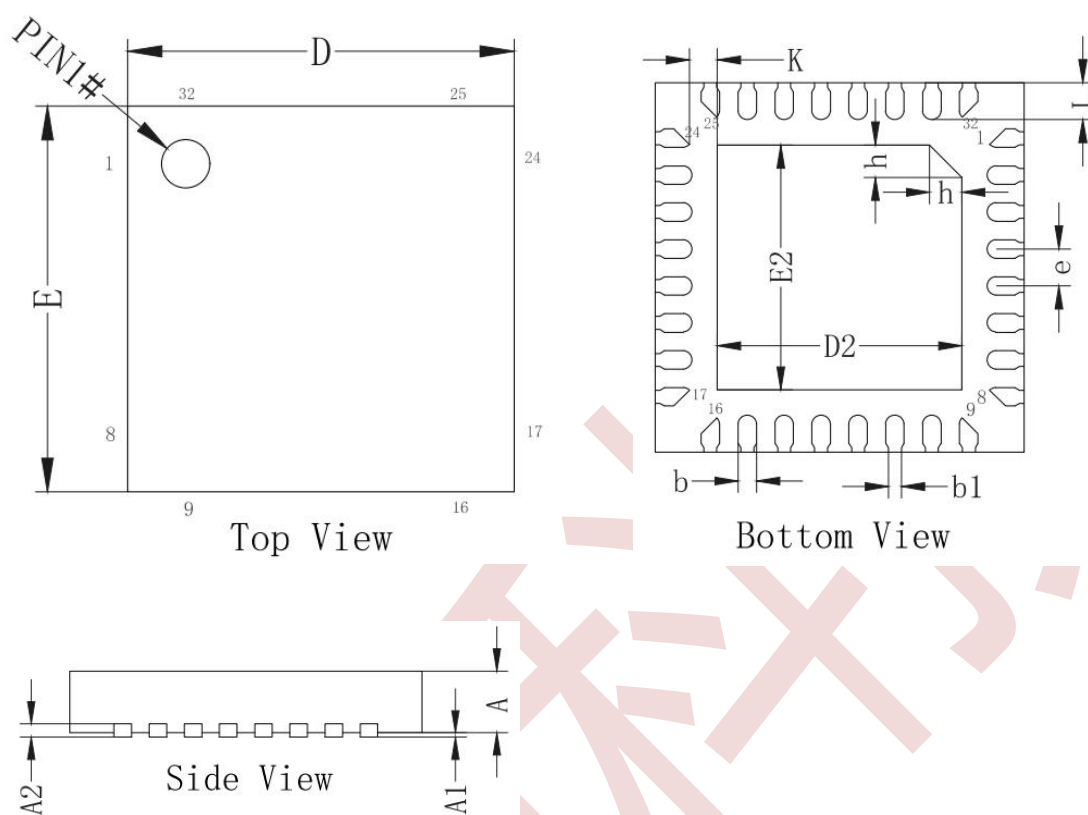
31.1 封装形式:

QFN48 (5×5 mm) 封装



序号	最小值 (mm)	标准值 (mm)	最大值 (mm)
A	0.7	0.75	0.8
A1	0.00	0.02	0.05
A2	0.203 REF		
D	4.90	5.0	5.10
E	4.90	5.0	5.10
D1	3.65	3.70	3.75
E1	3.65	3.70	3.75
e	0.35 BSC		
L1	0.30	0.35	0.40
L2	0.27	0.32	0.37
b	0.10	0.15	0.20
K	0.25	0.30	0.35
h	0.30	0.35	0.40

QFN32 (4×4 mm) 封装



序号	最小值 (mm)	标准值 (mm)	最大值 (mm)
A	0.65	0.70	0.75
A1	0.00	0.02	0.05
A2	0.152 REF		
D	3.90	4.0	4.10
E	3.90	4.0	4.10
D2	2.55	2.65	2.75
E2	2.55	2.65	2.75
e	0.40 BSC		
h	0.30	0.35	0.40
L	0.30	0.40	0.45
K	0.25	0.30	0.35
b	0.15	0.20	0.25
b1	0.14 REF		