



	Total page	
	Doc No	RM-TM32G07x
	Rev.	1.05
文件名稱	TM32G07x 參考手冊	





TM32G07x

参考手册

Rev 1.05

hitenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. hitenx does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. hitenx products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses hitenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold hitenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that hitenx was negligent regarding the design or manufacture of the part.

修订历史

版本	日期	描述
V1.00	2023/03/01	新颁
V1.01	2023/7/28	<ol style="list-style-type: none">1. 增加 section18.3.11 对 DAC 参考源电压使用限制的描述。2. 去除 section18.5.24-27 对 COMP 窗口模式的支持。3. 修改 section5.2.3 和 section5.4.2 对 PVD 检测引脚由 PD1 修正为 PC4。
V1.02	2024/1/16	<ol style="list-style-type: none">1. 修改 section16.3.3 表 16-1 触发与开始之间转换的延迟。2. 修改 section16.10.2 ADC_CFG1 的 vbgsel 寄存器说明。3. 修改 section16.10.6 ADC_SAMPT 最小采样时间为 2.5adc clk。4. 修改 section18.5 CMPx_CSR(x:1-4)的 bit14-13 描述。5. 修改寄存器与位名与固件库一致。
V1.03	2024/1/30	<ol style="list-style-type: none">1. 合并《API 应用笔记》的部分条目。2. 修改文档名为 TM32G07x 参考手册。
V1.04	2024/2/24	<ol style="list-style-type: none">1. 修改 section4 文本格式2. 修改图 18-13. 修改 HSISYS 为 18MHz
V1.05	2024/5/16	<ol style="list-style-type: none">1. 修改部分格式错误2. 修改部分内容错误3. 修改 section17.3.2 修改“至少 30us”为“至少 1 个 rck 延时”4. 修改 section17.3.2 增加对 TKx 采样数据地址的描述。

目录

1	文档约定 (Documentation conventions)	33
1.1	寄存器列表符号定义	33
1.2	术语及定义	33
2	产品简介 (Product Introduction)	35
3	存储器与总线架构 (Memory and bus architecture)	36
3.1	系统架构	36
3.1.1	System Bus	37
3.1.2	DMA Bus	38
3.1.3	Bus Matrix	38
3.1.4	AHB to APB 总线桥	38
3.2	存储器	38
3.2.1	SRAM	38
3.2.2	Flash	38
3.3	存储器映射	39
3.4	启动配置	42
3.4.1	存储区重映射	43
3.4.2	Bootloader	44
4	Flash 存储器 (Flash)	45
4.1	简介	45
4.2	Flash 主要特性	45
4.3	Flash 功能描述	45
4.3.1	Flash 存储器组成	45
4.3.2	Flash 读取访问等待周期	47
4.3.3	Flash 解锁	47
4.3.4	flash 擦除操作	49
4.3.5	User flash 区编程操作	50
4.4	Option bytes 区操作说明	52
4.4.1	选项字节组成	52
4.4.2	选项字节更新	53
4.4.3	选项字节加载	54
4.5	Flash 安全保护机制	55
4.5.1	Flash 读出保护	55
4.5.2	代码读出保护	59
4.5.3	Flash 写入保护	60
4.5.4	Flash 用户安全区域	61
4.5.5	User flash 区强制启动	62
4.6	Flash 中断	62
4.7	Flash 寄存器	63
4.7.1	Flash 访问控制寄存器 (FLASH_ACR)	63

4.7.2	Flash 控制解锁密钥寄存器 (FLASH_KEYR)	63
4.7.3	Flash 选项字节解锁密钥寄存器 (FLASH_OPTKEYR)	64
4.7.4	Flash 状态寄存器 (FLASH_SR)	64
4.7.5	Flash 控制寄存器 (FLASH_CR)	67
4.7.6	选项字节寄存器 1 (FLASH_OPTR1)	69
4.7.7	选项字节寄存器 2 (FLASH_OPTR2)	70
4.7.8	代码读出保护区 1A 起始地址寄存器 (FLASH_PCROP1ASR)	71
4.7.9	代码读出保护区 1A 结束地址寄存器 (FLASH_PCROP1AER)	71
4.7.10	代码读出保护区 1B 起始地址寄存器 (FLASH_PCROP1BSR)	72
4.7.11	代码读出保护区 1B 结束地址寄存器 (FLASH_PCROP1BER)	72
4.7.12	写保护区 1A 起始地址寄存器 (FLASH_WRP1ASR)	73
4.7.13	写保护区 1A 结束地址寄存器 (FLASH_WRP1AER)	73
4.7.14	写保护区 1B 起始地址寄存器 (FLASH_WRP1BSR)	74
4.7.15	写保护区 1B 结束地址寄存器 (FLASH_WRP1BER)	74
4.7.16	用户安全配置寄存器 (FLASH_SECR)	75
4.7.17	芯片封装参数寄存器 (FLASH_PACKAGER)	75
4.7.18	芯片产品型号参数寄存器 (FLASH_DEVTYPR)	76
	Flash 空间参数寄存器 (FLASH_UFSIZER)	76
4.7.19	SRAM 空间参数寄存器 (FLASH_SRAMSIZER)	77
5	电源管理单元 (PWR)	78
5.1	电源	78
5.1.1	内部调压器	78
5.2	电源电压监测	79
5.2.1	上电复位和掉电复位	79
5.2.2	欠压复位	79
5.2.3	可编程电压检测	79
5.3	低功耗模式	81
5.3.1	Run	83
5.3.2	低功耗模式进入和退出	84
5.3.3	Sleep	84
5.3.4	Stop	85
5.4	PWR 寄存器	87
5.4.1	电源管理控制寄存器 1 (PWR_CR1)	87
5.4.2	电源管理控制寄存器 2 (PWR_CR2)	88
5.4.3	电源管理状态寄存器 2 (PWR_SR2)	90
5.4.4	电源管理状态清除寄存器 (PWR_SCR)	90
6	复位和时钟控制器 (RCC)	92
6.1	复位	92
6.1.1	POR/PDR 复位	92
6.1.2	系统复位	92
6.2	时钟	92

6.2.1	HSE 时钟	94
6.2.2	HSI 时钟	96
6.2.3	MSI 时钟	96
6.2.4	PLL 时钟	96
6.2.5	LSE 时钟	97
6.2.6	LSI 时钟	98
6.2.7	系统时钟	99
6.2.8	LSE CSS	99
6.2.9	PLL CSS	100
6.2.10	外设异步时钟选择	100
6.2.11	外设时钟使能寄存器	100
6.2.12	TIMx 时钟	100
6.2.13	IWDG 时钟	101
6.2.14	ADC 时钟	101
6.2.15	LCD 时钟	101
6.2.16	RTC 时钟	101
6.2.17	时钟输出功能	101
6.2.18	基于 TIM4 的内/外部时钟测量	102
6.3	低功耗模式	103
6.4	RCC 寄存器	104
6.4.1	时钟控制/状态寄存器 (RCC_CR)	104
6.4.2	时钟稳定配置寄存器 (RCC_STABCR)	107
6.4.3	时钟配置寄存器 (RCC_CFGR)	108
6.4.4	PLL 配置寄存器 (RCC_PLLCFGR)	110
6.4.5	时钟中断使能寄存器 (RCC_CIER)	112
6.4.6	时钟中断标志寄存器 (RCC_CIFR)	113
6.4.7	时钟中断清除寄存器 (RCC_CICR)	114
6.4.8	GPIO 端口复位寄存器 (RCC_IOPRSTR)	116
6.4.9	AHB 外设复位寄存器 (RCC_AHBSTR)	116
6.4.10	APB1 外设复位寄存器 (RCC_APBSTR1)	117
6.4.11	APB2 外设复位寄存器 (RCC_APBSTR2)	119
6.4.12	I/O 端口时钟使能寄存器 (RCC_IOPENR)	120
6.4.13	AHB 外设时钟使能寄存器 (RCC_AHBENR)	121
6.4.14	APB1 外设时钟使能寄存器 (RCC_APBENR1)	122
6.4.15	APB2 外设时钟使能寄存器 (RCC_APBENR2)	124
6.4.16	外设异步时钟配置寄存器 (RCC_CCIPR)	125
6.4.17	控制寄存器 (RCC_BDCR)	126
6.4.18	时钟控制/状态寄存器 (RCC_CSR)	128
6.4.19	HSI Trim 寄存器 (RCC_HSITRIMR)	130
6.4.20	MSI Trim 寄存器 (RCC_MSITRIMR)	131
6.4.21	BGR Trim 寄存器 (RCC_BGRTRIMR)	131

6.4.22	DAC Trim 寄存器 (RCC_DACTRIMR)	132
6.4.23	OPAMP Trim 寄存器 (RCC_OPATRIMR)	133
7	通用输入/输出接口 (GPIO)	134
7.1	简介	134
7.2	GPIO 主要特性	134
7.3	GPIO 功能描述	134
7.3.1	通用 I/O (GPIO)	136
7.3.2	I/O 引脚功能复用器和映射	136
7.3.3	I/O 端口配置寄存器	137
7.3.4	I/O 端口数据寄存器	137
7.3.5	I/O 数据位操作	138
7.3.6	GPIO 锁定机制	138
7.3.7	外部中断线/唤醒线	138
7.3.8	输入配置	139
7.3.9	输出配置	140
7.3.10	复用功能配置	141
7.3.11	复用功能选择	141
7.3.12	模拟模式配置	142
7.4	GPIO 寄存器	143
7.4.1	GPIO 端口模式寄存器 (GPIOx_MODER) (x = A~D)	143
7.4.2	GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A~D)	143
7.4.3	GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A~D)	144
7.4.4	GPIO 端口输入寄存器 (GPIOx_IDR) (x = A~D)	144
7.4.5	GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A~D)	145
7.4.6	GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A~D)	145
7.4.7	GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A~D)	146
7.4.8	GPIO 复用功能低位寄存器 (GPIOx_AFR1) (x = A~D)	147
7.4.9	GPIO 复用功能高位寄存器 (GPIOx_AFR2) (x = A~D)	147
7.4.10	GPIO 端口位复位寄存器 (GPIOx_BRR) (x = A~D)	148
8	系统配置控制器 (SYSCFG)	149
8.1	简介	149
8.2	系统配置寄存器	149
8.2.1	系统配置寄存器 1 (SYSCFG_CFGR1)	149
8.2.2	系统配置寄存器 2 (SYSCFG_CFGR2)	150
9	外设互联 (Interconnect Matrix)	151
9.1	简介	151
9.2	外设互联概述	151
9.3	外设互联功能描述	152
9.3.1	从 TIM1/2/3/4/5 到 TIM1/2/3/4/5	152
9.3.2	从 ADC、SAC、CSS、PVD、SYSERR 到 TIM1	152
9.3.3	从 SAC、LSE 到 TIM2/3	153

9.3.4	从 SAC 到 TIM4	153
9.3.5	从 HSE、LSE、LSI、MSI、MCO 到 TIM4	153
9.3.6	从 RTC、SAC 到 LPTIM	153
9.3.7	从 HSE、LSE、HSI、MSI、LSI、PLL、SYSCLK 到 MCO	153
9.3.8	从 LSE、LSI 到 LSCO	153
9.3.9	从 TIM3/4、USART2/3 到 IRTIM	154
9.3.10	从 TIM1/2/4/6、PLA、EXTI、TS、BGR 到 ADC	154
9.3.11	从 TIM1/2/3/4/5/6/7、PLA、LPTIM、EXTI 到 DAC	154
9.3.12	从 DMAMUX 到 DMA	154
9.3.13	从 TIM4、LPTIM、EXTI、LCD 到 DMAMUX	154
9.3.14	从 BGR 到 SAC	154
9.3.15	从 TIM1/2/3/4/5/6/7、SAC、LPTIM、USART1/2/3、SPI、RTC 到 PLA	154
10	直接存储器访问控制器 (DMA)	155
10.1	简介	155
10.2	DMA 主要特性	155
10.3	DMA 功能说明	156
10.3.1	DMA 框图	156
10.3.2	DMA 信号	156
10.3.3	DMA 请求源	156
10.3.4	DMA 传输	156
10.3.5	优先级	157
10.3.6	传输模式	157
10.3.7	数据位宽	158
10.3.8	地址递增	158
10.3.9	循环模式	158
10.3.10	配置流程	159
10.3.11	暂停和停止通道	159
10.3.12	错误标志	160
10.4	DMA 中断	160
10.5	DMA 寄存器	161
10.5.1	DMA 中断状态寄存器 (DMA_ISR)	161
10.5.2	DMA 中断标志清除寄存器 (DMA_IFCR)	163
10.5.3	DMA 通道 x 控制寄存器 (DMA_CCRx)	165
10.5.4	DMA 通道 x 待传输次数寄存器 (DMA_CNDTRx)	166
10.5.5	DMA 通道 x 源地址寄存器 (DMA_CSARx)	167
10.5.6	DMA 通道 x 目的地址寄存器 (DMA_CDARx)	167
11	DMA 请求多路器 (DMAMUX)	169
11.1	简介	169
11.2	DMAMUX 主要特性	169
11.3	DMAMUX 功能说明	170
11.3.1	DMAMUX 功能框图	170

11.3.2	DMAMUX 信号.....	170
11.3.3	DMAMUX 信号映射.....	170
11.3.4	DMAMUX 请求复用器.....	172
11.3.5	DMAMUX 请求生成器.....	172
11.3.6	配置流程.....	173
11.4	DMAMUX 中断.....	174
11.5	DMAMUX 寄存器.....	174
11.5.1	DMAMUX 请求复用器通道 x 控制寄存器 (DMAMUX_CxCR)	174
11.5.2	DMAMUX 请求生成器 x 控制寄存器 (DMAMUX_RGxCR)	175
11.5.3	DMAMUX 请求生成器中断状态寄存器 (DMAMUX_RGSR)	176
11.5.4	DMAMUX 请求生成器中断标志清除寄存器 (DMAMUX_RGCFR)	177
12	嵌套向量中断控制器 (NVIC)	178
12.1	简介.....	178
12.2	主要特征.....	178
12.3	中断向量表.....	178
13	扩展中断和事件控制器 (EXTI)	180
13.1	简介.....	180
13.2	EXTI 主要功能.....	180
13.3	EXTI 模块示意图.....	180
13.4	EXTI 的主要信号连接及功能.....	181
13.5	EXTI 通道及信号对应表.....	182
13.6	EXTI 功能描述.....	183
13.6.1	EXTI 可配置通道.....	184
13.6.2	EXTI 直接通道.....	185
13.6.3	I/O 多路选择器.....	186
13.7	EXTI 寄存器.....	187
13.7.1	EXTI 上升沿触发选择寄存器 (EXTI_RTSTR)	187
13.7.2	EXTI 下降沿触发选择寄存器 (EXTI_FTSTR)	187
13.7.3	EXTI 上升沿触发事件挂起寄存器 (EXTI_RPR)	188
13.7.4	EXTI 下降沿触发事件挂起寄存器 (EXTI_FPR)	188
13.7.5	EXTI I/O 选择寄存器 1 (EXTI_EXTICR1)	189
13.7.6	EXTI I/O 选择寄存器 2 (EXTI_EXTICR2)	190
13.7.7	EXTI 中断唤醒屏蔽寄存器 (EXTI_IMR)	192
13.7.8	EXTI 事件唤醒屏蔽寄存器 (EXTI_EMR)	192
14	硬件除法器单元 (HDIV)	194
14.1	简介.....	194
14.2	HDIV 主要特性.....	194
14.3	HDIV 功能说明.....	194
14.3.1	HDIV 功能描述.....	194
14.4	HDIV 寄存器.....	195
14.4.1	HDIV 被除数寄存器 (HDIV_DIVEDR)	195

14.4.2	HDIV 除数寄存器 (HDIV_DIVR)	195
14.4.3	HDIV 商寄存器 (HDIV_QUOTR)	195
14.4.4	HDIV 余数寄存器 (HDIV_RMDR)	196
14.4.5	HDIV 符号寄存器 (HDIV_SIGNR)	196
14.4.6	HDIV 状态寄存器 (HDIV_SR)	196
15	循环冗余校验 (CRC)	198
15.1	简介	198
15.2	CRC 主要特性	198
15.3	CRC 功能描述	198
15.3.1	CRC 框图	198
15.3.2	CRC 内部信号	198
15.3.3	CRC 操作说明	199
15.4	CRC 寄存器	200
15.4.1	CRC 控制状态寄存器 (CRC_CR)	200
15.4.2	CRC 结果寄存器 (CRC_RR)	201
15.4.3	CRC 数据寄存器 (CRC_DR)	201
16	模数转换器 (ADC)	202
16.1	简介	202
16.2	ADC 主要特性	202
16.3	ADC 功能说明	203
16.3.1	校准	204
16.3.2	开关控制	205
16.3.3	时钟源	206
16.3.4	输入通道	207
16.3.5	通道选择	208
16.3.6	采样时间	209
16.3.7	转换模式	209
16.3.8	启动转换	212
16.3.9	转换时间	212
16.3.10	停止转换	213
16.3.11	转换触发模式及触发沿	213
16.3.12	转换结束、采用阶段结束	214
16.3.13	转换序列结束	215
16.3.14	转换时序示例	215
16.3.15	ADC 参考源选择	216
16.4	ADC 数据管理	217
16.4.1	数据寄存器及数据对齐	217
16.4.2	数据溢出	218
16.4.3	在不使用 DMA 情况下管理转换的数据序列	218
16.4.4	在不使用 DMA 且不发生溢出的情况下管理转换的数据	219
16.4.5	使用 DMA 管理转换的数据	219

16.5	低功耗模式.....	219
16.5.1	自动等待模式.....	219
16.6	模拟看门狗.....	220
16.6.1	模拟看门狗简介.....	220
16.6.2	ADC_AWDGx_OUT 信号输出生成.....	220
16.7	温度传感器和内部参考电压.....	222
16.8	电源电压监测.....	225
16.9	ADC 中断.....	225
16.10	ADC 寄存器.....	226
16.10.1	ADC 控制寄存器 (ADC_CR).....	226
16.10.2	ADC 配置寄存器 1 (ADC_CFGR1).....	227
16.10.3	ADC 配置寄存器 2 (ADC_CFGR2).....	230
16.10.4	ADC 中断和状态寄存器 (ADC_ISR).....	232
16.10.5	ADC 中断使能寄存器 (ADC_IER).....	233
16.10.6	ADC 采样时间寄存器 (ADC_SMPR).....	234
16.10.7	ADC 通道选择寄存器 1 (ADC_CHSELR1).....	235
16.10.8	ADC 通道选择寄存器 2 (ADC_CHSELR2).....	236
16.10.9	ADC 看门狗 1 监控电压阈值寄存器 (ADC_AWD1TR).....	237
16.10.10	ADC 校准系数 (ADC_CALFACT).....	238
16.10.11	ADC 数据寄存器 (ADC_DR).....	239
17	触摸按键 (ATK).....	240
17.1	简介.....	240
17.2	主要特性.....	240
17.3	功能说明.....	240
17.3.1	结构框图.....	240
17.3.2	功能描述.....	241
17.4	ATK 寄存器.....	243
17.4.1	TKA 控制寄存器 (CRA).....	243
17.4.2	TKB 控制寄存器 (CRB).....	245
17.4.3	TK 状态寄存器 (ISR).....	247
17.4.4	TK 启动通道选择寄存器 (CHSR).....	247
17.4.5	TKA 通道使能寄存器 (CHENRA).....	249
17.4.6	TKB 通道使能寄存器 (CHENRB).....	249
17.4.7	TKA 通道调试参数 1 寄存器 (ITRIMRA1).....	250
17.4.8	TKA 通道调试参数 2 寄存器 (ITRIMRA2).....	250
17.4.9	TKA 通道调试参数 3 寄存器 (ITRIMRA3).....	251
17.4.10	TKA 通道调试参数 4 寄存器 (ITRIMRA4).....	252
17.4.11	TKA 通道调试参数 5 寄存器 (ITRIMRA5).....	252
17.4.12	TKA 通道调试参数六寄存器 (ITRIMRA6).....	253
17.4.13	TKB 通道调试参数 1 寄存器 (ITRIMRB1).....	254
17.4.14	TKB 通道调试参数 2 寄存器 (ITRIMRB2).....	254

17.4.15	TKB 通道调试参数 3 寄存器 (ITRIMRB3)	255
17.4.16	TKB 通道调试参数 4 寄存器 (ITRIMRB4)	256
17.4.17	TKB 通道调试参数 5 寄存器 (ITRIMRB5)	256
17.4.18	TKB 通道调试参数 6 寄存器 (ITRIMRB6)	257
17.4.19	TK 引脚互换寄存器 (PINABSR)	257
17.4.20	TKA RAM (TKA_RAM)	258
17.4.21	TKB RAM (TKB_RAM)	258
18	模拟智能组合 (SAC)	259
18.1	简介	259
18.2	主要特性	259
18.3	SAC 功能描述	260
18.3.1	结构框图	260
18.3.2	引脚和内部信号	260
18.3.3	复位和时钟源	261
18.3.4	比较器锁存机制	261
18.3.5	比较器迟滞	262
18.3.6	比较器输出滤波功能	262
18.3.7	比较器中断	263
18.3.8	DAC 通道使能	263
18.3.9	DAC 数据格式	263
18.3.10	DAC 转换	264
18.3.11	DAC 输出电压	265
18.3.12	DAC 触发选择	265
18.3.13	DMA 请求	266
18.3.14	噪声生成器	267
18.3.15	三角波生成器	267
18.3.16	DAC 通道模式	268
18.3.17	DAC 输出缓冲器校准	269
18.3.18	DAC 双通道转换	269
18.4	DAC 中断	274
18.5	SAC 寄存器	274
18.5.1	DAC 控制寄存器 (DAC_CR)	274
18.5.2	DAC 软件触发寄存器 (DAC_SWTRIGR)	277
18.5.3	DAC 1 通道 12 位右对齐数据保持寄存器(DAC_DHR12R1)	277
18.5.4	DAC 1 通道 12 位左对齐数据保持寄存器(DAC_DHR12L1)	278
18.5.5	DAC 1 通道 8 位右对齐数据保持寄存器(DAC_DHR8R1)	278
18.5.6	DAC 2 通道 12 位右对齐数据保持寄存器(DAC_DHR12R2)	279
18.5.7	DAC 2 通道 12 位左对齐数据保持寄存器(DAC_DHR12L2)	279
18.5.8	DAC 2 通道 8 位右对齐数据保持寄存器(DAC_DHR8R2)	279
18.5.9	DAC 双通道 12 位右对齐数据保持寄存器(DAC_DHR12RD)	280
18.5.10	DAC 双通道 12 位左对齐数据保持寄存器(DAC_DHR12LD)	280

18.5.11	DAC 双通道 8 位右对齐数据保持寄存器(DAC_DHR8RD).....	281
18.5.12	DAC 1 通道数据输出寄存器(DAC_DOR1)	281
18.5.13	DAC 2 通道数据输出寄存器(DAC_DOR2)	282
18.5.14	DAC 状态寄存器(DAC_SR).....	282
18.5.15	DAC 模式控制寄存器(DAC_MCR).....	283
18.5.16	DAC 控制寄存器 2 (DAC_CR2)	283
18.5.17	DAC 软件触发寄存器 (DAC_SWTRIGR2)	286
18.5.18	DAC 3 通道 12 位右对齐数据保持寄存器(DAC_DHR12R3)	287
18.5.19	DAC 4 通道 12 位右对齐数据保持寄存器(DAC_DHR12R4)	287
18.5.20	DAC 3 通道数据输出寄存器(DAC_DOR3)	288
18.5.21	DAC 4 通道数据输出寄存器(DAC_DOR4)	288
18.5.22	DAC 状态寄存器(DAC_SR2).....	288
18.5.23	DAC 模式控制寄存器 2(DAC_MCR2).....	289
18.5.24	COMP1 控制和状态寄存器 (COMP1_CSR)	290
18.5.25	COMP2 控制和状态寄存器 (COMP2_CSR)	292
18.5.26	COMP3 控制和状态寄存器 (COMP3_CSR)	293
18.5.27	COMP4 控制和状态寄存器 (COMP4_CSR)	295
18.5.28	OPAMP1 控制和状态寄存器 (OPAMP1_CSR)	297
18.5.29	OPAMP2 控制和状态寄存器 (OPAMP2_CSR)	299
18.5.30	OPAMP3 控制和状态寄存器 (OPAMP3_CSR)	300
18.5.31	OPAMP4 控制和状态寄存器 (OPAMP4_CSR)	302
19	液晶显示/发光二极管控制器 (LCD/LED)	304
19.1	简介	304
19.2	LCD 主要特性	304
19.3	LED 主要特性	305
19.4	功能描述.....	305
19.4.1	概述.....	305
19.4.2	LED 矩阵模式.....	306
19.4.3	LED 点阵模式.....	308
19.4.4	LCD 模式.....	312
19.5	低功耗模式.....	322
19.6	LCD 寄存器	322
19.6.1	LCD 控制寄存器 (LCD_CR)	322
19.6.2	LCD 显示存储器 (LCD_RAM)	324
20	高级控制定时器 (TIM1)	325
20.1	TIM1 简介.....	325
20.2	TIM1 主要特性.....	325
20.3	TIM1 功能说明.....	328
20.3.1	时基单元.....	328
20.3.2	计数器模式.....	329
20.3.3	重复计数器.....	339

20.3.4	外部触发输入	341
20.3.5	时钟选择	342
20.3.6	捕获/比较通道	345
20.3.7	输入捕获模式	347
20.3.8	PWM 输入模式	349
20.3.9	强制输出模式	350
20.3.10	输出比较模式	350
20.3.11	PWM 模式	351
20.3.12	不对称 PWM 模式	355
20.3.13	组合 PWM 模式	355
20.3.14	组合 3 相 PWM 模式	356
20.3.15	互补输出和死区插入	357
20.3.16	使用断路功能	359
20.3.17	双向断路输入	364
20.3.18	发生外部事件时清除 OCxREF 信号	366
20.3.19	生成 6 步 PWM	367
20.3.20	单脉冲模式	368
20.3.21	可重触发单脉冲模式	370
20.3.22	编码接口模式	371
20.3.23	UIF 位重映射	373
20.3.24	定时器输入异或功能	373
20.3.25	霍尔传感器接口	374
20.3.26	定时器同步	376
20.3.27	ADC 同步	379
20.3.28	DMA 连续传输模式	380
20.3.29	调试模式	381
20.4	TIM1 寄存器	381
20.4.1	TIM1 控制寄存器 (TIM1_CR1)	381
20.4.2	TIM1 控制寄存器 2 (TIM1_CR2)	383
20.4.3	TIM1 从模式控制寄存器 (TIM1_SMCR)	386
20.4.4	TIM1 DMA/中断使能寄存器 (TIM1_DIER)	389
20.4.5	TIM1 状态寄存器 (TIM1_SR)	391
20.4.6	TIM1 事件产生寄存器 (TIM1_EGR)	393
20.4.7	TIM1 捕获/比较模式寄存器 1_输出比较模式 (TIM1_CCMR1)	395
20.4.8	TIM1 捕获/比较模式寄存器 1_输入捕获模式 (TIM1_CCMR1)	398
20.4.9	TIM1 捕获/比较模式寄存器 2_输出比较模式 (TIM1_CCMR2)	399
20.4.10	TIM1 捕获/比较模式寄存器 2_输入捕获模式 (TIM1_CCMR2)	401
20.4.11	TIM1 捕获/比较使能寄存器 1 (TIM1_CCER)	402
20.4.12	TIM1 计数值寄存器 (TIM1_CNT)	405
20.4.13	TIM1 预分频器寄存器 (TIM1_PSC)	406
20.4.14	TIM1 自动重载值寄存器 (TIM1_ARR)	406

20.4.15	TIM1 重复计数器寄存器 (TIM1_RCR)	407
20.4.16	TIM1 捕获/比较寄存器 1 (TIM1_CCR1)	407
20.4.17	TIM1 捕获/比较寄存器 2 (TIM1_CCR2)	408
20.4.18	TIM1 捕获/比较寄存器 3 (TIM1_CCR3)	408
20.4.19	TIM1 捕获/比较寄存器 4 (TIM1_CCR4)	409
20.4.20	TIM1 断路和死区寄存器 (TIM1_BDTR)	409
20.4.21	TIM1 DMA 控制寄存器 (TIM1_DCR)	413
20.4.22	TIM1 DMA 全传输地址寄存器 (TIM1_DMAR)	414
20.4.23	TIM1 配置寄存器 (TIM1_OR1)	415
20.4.24	TIM1 捕获/比较模式寄存器 3 (TIM1_CCMR3)	415
20.4.25	TIM1 捕获/比较寄存器 5 (TIM1_CCR5)	416
20.4.26	TIM1 捕获/比较寄存器 6 (TIM1_CCR6)	417
20.4.27	TIM1 轮换功能寄存器 1 (TIM1_AF1)	418
20.4.28	TIM1 轮换功能寄存器 2 (TIM1_AF2)	419
20.4.29	TIM1 定时器输入选择寄存器 (TIM1_TISEL)	420
21	通用定时器 (TIM2/3/4)	422
21.1	TIM2/3/4 简介	422
21.2	TIM2/3/4 主要特性	422
21.3	TIM2/3/4 功能说明	423
21.3.1	时基单元	423
21.3.2	计数器模式	425
21.3.3	重复计数器	436
21.3.4	时钟选择	437
21.3.5	捕获/比较通道	440
21.3.6	输入捕获模式	442
21.3.7	PWM 输入模式	443
21.3.8	输出比较模式	444
21.3.9	强制输出模式	446
21.3.10	PWM 模式	446
21.3.11	不对称 PWM 模式	449
21.3.12	组合 PWM 模式	450
21.3.13	单脉冲模式	451
21.3.14	可重触发单脉冲模式	453
21.3.15	发生外部事件时清除 OCxREF 信号	454
21.3.16	UIF 位重映射	455
21.3.17	定时器输入异或功能	455
21.3.18	定时器与外部触发同步	456
21.3.19	定时器同步	460
21.3.20	DMA 连续传输模式	464
21.3.21	调试模式	466
21.4	TIM2/3/4 寄存器	466

21.4.1	TIM2/3/4 控制寄存器 1 (TIM2/3/4_CR1)	466
21.4.2	TIM2/3/4 控制寄存器 2 (TIM2/3/4_CR2)	468
21.4.3	TIM2/3/4 从模式控制寄存器 (TIM2/3/4_SMCR)	469
21.4.4	TIM2/3/4 DMA/中断使能寄存器 (TIM2/3/4_DIER)	472
21.4.5	TIM2/3/4 状态寄存器 (TIM2/3/4_SR)	473
21.4.6	TIM2/3/4 事件产生寄存器 (TIM2/3/4_EGR)	475
21.4.7	TIM2/3/4 捕获/比较模式寄存器 1_输出比较模式 (TIM2/3/4_CCMR1)	476
21.4.8	TIM2/3/4 捕获/比较模式寄存器 1_输入捕获模式 (TIM2/3/4_CCMR1)	479
21.4.9	TIM2/3/4 捕获/比较模式寄存器 2_输出比较模式 (TIM2/3/4_CCMR2)	481
21.4.10	TIM2/3/4 捕获/比较模式寄存器 2_输入捕获模式 (TIM2/3/4_CCMR2)	482
21.4.11	TIM2/3/4 捕获/比较使能寄存器 1 (TIM2/3/4_CCER)	483
21.4.12	TIM2/3/4 计数值寄存器 (TIM2/3/4_CNT)	485
21.4.13	TIM2/3/4 预分频值寄存器 (TIM2/3/4_PSC)	485
21.4.14	TIM2/3/4 自动重载值寄存器 (TIM2/3/4_ARR)	486
21.4.15	TIM2/3/4 重复计数器寄存器 (TIM2/3/4_RCR)	486
21.4.16	TIM2/3/4 捕获/比较寄存器 1 (TIM2/3/4_CCR1)	487
21.4.17	TIM2/3/4 捕获/比较寄存器 2 (TIM2/3/4_CCR2)	487
21.4.18	TIM2/3/4 捕获/比较寄存器 3 (TIM2/3/4_CCR3)	488
21.4.19	TIM2/3/4 捕获/比较寄存器 4 (TIM2/3/4_CCR4)	488
21.4.20	TIM2/3/4 DMA 控制寄存器 (TIM2/3/4_DCR)	489
21.4.21	TIM2/3/4 DMA 全传输地址寄存器 (TIM2/3/4_DMAR)	490
21.4.22	TIM2/3/4 配置寄存器 1 (TIM2/3/4_OR1)	490
21.4.23	TIM2/3/4 轮换功能寄存器 1 (TIM2/3/4_AF1)	491
21.4.24	TIM2/3 定时器输入选择寄存器 (TIM2/3_TISEL)	491
21.4.25	TIM4 定时器输入选择寄存器 (TIM4_TISEL)	492
22	基本定时器 (TIM5/6/7)	494
22.1	简介	494
22.2	TIM5/6/7 主要特性	494
22.3	TIM5/6/7 功能说明	495
22.3.1	时基单元	495
22.3.2	计数模式	496
22.3.3	UIF 位重映射	498
22.3.4	时钟源	499
22.3.5	调试模式	499
22.4	TIM5/6/7 寄存器	499
22.4.1	TIM5/6/7 控制寄存器 1 (TIM_CR1)	499
22.4.2	TIM5/6/7 控制寄存器 2 (TIM_CR2)	501
22.4.3	TIM5/6/7 DMA/中断使能寄存器 (TIM_DIER)	501
22.4.4	TIM5/6/7 状态寄存器 (TIM_SR)	502
22.4.5	TIM5/6/7 事件产生寄存器 (TIM_EGR)	502
22.4.6	TIM5/6/7 计数值寄存器 (TIM_CNT)	503

22.4.7	TIM5/6/7 预分频值寄存器 (TIM_PSC)	503
22.4.8	TIM5/6/7 自动重载值寄存器 (TIM_ARR)	504
23	低功耗定时器 (LPTIM)	505
23.1	简介	505
23.2	LPTIM 主要特性	505
23.3	LPTIM 功能说明	506
23.3.1	LPTIM 框图	506
23.3.2	LPTIM 引脚和内部信号	506
23.3.3	LPTIM 触发映射	507
23.3.4	LPTIM 复位和时钟	507
23.3.5	干扰滤波器	508
23.3.6	预分频器	509
23.3.7	触发多路复用器	509
23.3.8	工作模式	509
23.3.9	超时功能	511
23.3.10	生成波形	511
23.3.11	寄存器更新	512
23.3.12	计数器模式	513
23.3.13	定时器使能	513
23.3.14	定时器计数器复位	514
23.3.15	编码器模式	514
23.4	调试模式	520
23.5	LPTIM 低功耗模式	520
23.6	中断 LPTIM	521
23.7	LPTIM 寄存器	521
23.7.1	LPTIM 中断和状态寄存器 (LPTIM_ISR)	521
23.7.2	LPTIM 中断清零寄存器 (LPTIM_ICR)	522
23.7.3	LPTIM 中断使能寄存器(LPTIM_IER)	523
23.7.4	LPTIM 配置寄存器(LPTIM_CFGR)	525
23.7.5	LPTIM 控制寄存器 (LPTIM_CR)	528
23.7.6	LPTIM 比较寄存器 (LPTIM_CMP)	529
23.7.7	LPTIM 自动重载寄存器 (LPTIM_ARR)	529
23.7.8	LPTIM 计数器寄存器 (LPTIM_CNT)	530
23.7.9	LPTIM 配置寄存器 2 (LPTIM_CFGR2)	530
24	SysTick 定时器 (SysTick)	532
24.1	简介	532
24.2	主要特性	532
24.3	功能描述	532
24.4	SysTick 寄存器	533
25	红外控制模块 (IRTIM)	534
26	独立看门狗 (IWDG)	535

26.1	简介	535
26.2	功能特性.....	535
26.3	功能说明.....	535
26.3.1	结构框图	535
26.3.2	窗口功能.....	536
26.3.3	硬件方式使能看门狗.....	537
26.3.4	寄存器访问保护	537
26.3.5	独立看门狗超时设置.....	537
26.3.6	调试模式.....	538
26.4	IWDG 寄存器	538
26.4.1	IWDG 控制寄存器 (IWDG_KR)	538
26.4.2	IWDG 预分频寄存器 (IWDG_PR)	538
26.4.3	IWDG 重载寄存器 (IWDG_RLR)	539
26.4.4	IWDG 状态寄存器 (IWDG_SR)	540
26.4.5	IWDG 窗口寄存器 (IWDG_WINR)	540
27	窗口看门狗 (WWDG)	542
27.1	简介	542
27.2	功能特性.....	542
27.3	功能说明.....	542
27.3.1	结构框图	542
27.3.2	递减计数器.....	543
27.3.3	看门狗中断.....	543
27.3.4	窗口看门狗超时设置.....	544
27.3.5	调试模式.....	545
27.4	WWDG 寄存器.....	545
27.4.1	WWDG 控制寄存器 (WWDG_CR)	545
27.4.2	WWDG 配置寄存器 (WWDG_CFR)	545
27.4.3	WWDG 状态寄存器 (WWDG_SR)	546
28	I2C 接口 (I2C)	547
28.1	简介	547
28.2	主要特性.....	547
28.3	I2C 功能描述.....	548
28.3.1	I2C 框图	548
28.3.2	I2C 引脚和内部信号.....	548
28.3.3	I2C 时钟要求.....	548
28.3.4	模式选择	549
28.3.5	I2C 初始化	550
28.3.6	软件复位.....	555
28.3.7	数据传输.....	556
28.3.8	从模式.....	557
28.3.9	主模式.....	560

28.3.10	低功耗模式唤醒.....	564
28.3.11	DMA 传输.....	566
28.3.12	错误条件.....	567
28.4	低功耗模式说明.....	569
28.5	中断.....	569
28.6	I2C 寄存器.....	570
28.6.1	控制寄存器 1 (I2C_CR1).....	570
28.6.2	控制寄存器 2 (I2C_CR2).....	572
28.6.3	地址寄存器 1 (I2C_OAR1).....	574
28.6.4	地址寄存器 2 (I2C_OAR2).....	575
28.6.5	时钟配置寄存器 (I2C_TIMINGR).....	576
28.6.6	中断和状态寄存器 (I2C_ISR).....	577
28.6.7	中断清除寄存器 (I2C_ICR).....	579
28.6.8	接收数据寄存器 (I2C_RXDR).....	580
28.6.9	发送数据寄存器 (I2C_TXDR).....	581
29	通用异步收发器 (USART).....	582
29.1	简介.....	582
29.2	主要特性.....	582
29.3	USART 扩展功能.....	583
29.4	USART 功能列表.....	583
29.5	USART 功能概述.....	584
29.5.1	USART 框图.....	584
29.5.2	USART 帧格式.....	587
29.5.3	USART 校验控制.....	589
29.5.4	USART FIFO 和阈值设置.....	589
29.5.5	USART 发送器.....	590
29.5.6	USART 接收器.....	594
29.5.7	USART 波特率生成.....	600
29.5.8	USART 接收容差.....	601
29.5.9	USART 自动波特率检测.....	603
29.5.10	USART 多机通信.....	606
29.5.11	USART Modbus 通信.....	608
29.5.12	USART 同步模式.....	609
29.5.13	USART 单线半双工通信.....	613
29.5.14	USART 接收超时.....	613
29.5.15	USART IrDA SIR 编解码模块.....	614
29.5.16	USART 使用 DMA 通信.....	616
29.5.17	RS232 硬件流控和 RS485 控制.....	618
29.5.18	USART 低功耗管理.....	620
29.5.19	USART 中断请求表.....	623
29.6	USART 寄存器.....	624

29.6.1	USART 控制寄存器 1 (USART_CR1) -- 【FIFO 使能时】	624
29.6.2	USART 控制寄存器 1 (USART_CR1) -- 【FIFO 禁用】	628
29.6.3	USART 控制寄存器 2 (USART_CR2)	631
29.6.4	USART 控制寄存器 3 (USART_CR3)	635
29.6.5	USART 波特率分频寄存器 (USART_BRR)	638
29.6.6	USART 保护时间和预分频器寄存器 (USART_GTPR)	639
29.6.7	USART 超时及块传输长度寄存器 (USARTx_RTOR)	640
29.6.8	USART 请求寄存器 (USART_RQR)	640
29.6.9	USART 中断和状态寄存器 (USART_ISR) -- 【FIFO 使能】	641
29.6.10	USART 中断和状态寄存器 (USART_ISR) -- 【FIFO 未使能】	648
29.6.11	USART 中断标志清零寄存器(USART_ICR)	653
29.6.12	USART 接收数据寄存器(USARTx_RDR)	654
29.6.13	USART 发送数据寄存器(USARTx_TDR)	655
29.6.14	USART 预分频器寄存器(USARTx_PRESC)	655
30	低功耗通用异步收发器 (LPUART)	656
30.1	简介	656
30.2	主要特性	657
30.3	LPUART 功能一览	658
30.4	LPUART 功能详解	659
30.4.1	LPUART 框图	659
30.4.2	LPUART 信号详解	660
30.4.3	LPUART 字符详解	660
30.4.4	LPUART FIFO 和阈值设置	661
30.4.5	LPUART 发送器	662
30.4.6	LPUART 接收器	666
30.4.7	LPUART 波特率产生	669
30.4.8	LPUART 接收容差	670
30.4.9	LPUART 多机通信模式	671
30.4.10	LPUART 校验控制	673
30.4.11	LPUART 单线半双工通信	674
30.4.12	LPUART 使用 DMA 通信	675
30.4.13	LPUART RS232 硬件流控和 RS485 驱动器	678
30.5	LPUART 低功耗模式	680
30.5.1	LPUART 对低功耗模式的支持	680
30.5.2	低功耗模式和静默模式	681
30.5.3	LPUART 内核时钟 (LPUART_KCLK) 停止时从 Stop 模式唤醒	682
30.5.4	确定从低功耗模式唤醒芯片的最高波特率	683
30.5.5	LPUART 中断请求表	683
30.6	LPUART 寄存器	685
30.6.1	LPUART 控制寄存器 1 (LPUART_CR1) -- (FIFO 使能)	685
30.6.2	LPUART 控制寄存器 1 (LPUART_CR1) -- (FIFO 禁止)	688

30.6.3	LPUART 控制寄存器 2 (LPUART_CR2)	691
30.6.4	LPUART 控制寄存器 3 (LPUART_CR3)	693
30.6.5	LPUART 波特率分频寄存器 (LPUART_BRR)	696
30.6.6	LPUART 请求寄存器 (LPUART_RQR)	697
30.6.7	LPUART 中断和状态寄存器 (LPUART_ISR) -- (FIFO 使能)	698
30.6.8	LPUART 中断和状态寄存器 (LPUART_ISR) -- (FIFO 禁止)	702
30.6.9	LPUART 中断标志清零寄存器(LPUART_ICR)	706
30.6.10	LPUART 接收数据寄存器(LPUART_RDR)	707
30.6.11	LPUART 发送数据寄存器(LPUARTx_TDR)	708
30.6.12	LPUART 预分频器寄存器(LPUART_PRESC)	709
31	串行外设接口 (SPI)	710
31.1	简介	710
31.2	SPI 主要特性	710
31.3	SPI 功能说明	711
31.3.1	SPI 框图	711
31.3.2	SPI 引脚和信号	711
31.3.3	单主机通信	712
31.3.4	多主机通信	713
31.3.5	从机通信	713
31.3.6	NSS 管理	714
31.3.7	通信模式	714
31.3.8	状态标志	715
31.3.9	错误或状态标志	717
31.3.10	初始化 SPI	718
31.3.11	数据发送和接收	718
31.3.12	禁止 SPI	721
31.4	SPI 中断	721
31.5	SPI 寄存器	722
31.5.1	SPI 控制寄存器 1 (SPIx_CR1)	722
31.5.2	SPI 控制寄存器 2 (SPIx_CR2)	725
31.5.3	SPI FIFO 清空寄存器 (SPIx_FIFOCLR)	725
31.5.4	SPI 中断和状态寄存器 (SPIx_ISR)	726
31.5.5	SPI 中断标志清零寄存器 (SPIx_ICR)	728
31.5.6	SPI 数据寄存器 (SPIx_DR)	728
32	实时时钟计数器 (RTC)	730
32.1	简介	730
32.2	RTC 主要特性	730
32.3	RTC 功能说明	730
32.3.1	RTC 框图	731
32.3.2	RTC 初始化和配置	732
32.3.3	RTC 低功耗模式切换	732

32.3.4	RTC 读取日历.....	733
32.3.5	RTC 写入日历寄存器.....	733
32.3.6	RTC 闹钟设定.....	734
32.3.7	RTC 输出时钟校准.....	734
32.3.8	RTC 时钟输出.....	734
32.3.9	RTC 中断.....	735
32.4	RTC 寄存器	735
32.4.1	RTC 控制寄存器 0 (RTC_CR0)	735
32.4.2	RTC 控制寄存器 1 (RTC_CR1)	737
32.4.3	RTC 秒计数寄存器 (RTC_SECR)	738
32.4.4	RTC 分计数寄存器 (RTC_MINR)	738
32.4.5	RTC 时计数寄存器 (RTC_HOUR)	739
32.4.6	RTC 日计数寄存器 (RTC_DAYR)	740
32.4.7	RTC 周计数寄存器 (RTC_WEEKR)	740
32.4.8	RTC 月计数寄存器 (RTC_MONR)	741
32.4.9	RTC 年计数寄存器 (RTC_YEARR)	741
32.4.10	RTC 分闹钟寄存器 (RTC_ALMMINR)	742
32.4.11	RTC 时闹钟寄存器 (RTC_ALMHOURR)	742
32.4.12	RTC 周闹钟寄存器 (RTC_ALMWEEKR)	743
32.4.13	RTC 时钟校准寄存器 (RTC_COMPENR)	743
33	可编程逻辑阵列 (PLA)	745
33.1	简介	745
33.2	功能特性.....	745
33.3	功能说明.....	746
33.3.1	结构框图.....	746
33.3.2	PLA 配置顺序.....	747
33.3.3	输入多路复用器选择.....	747
33.3.3.1	PLU 多路复用器输入选择.....	747
33.3.4	LUT 配置.....	749
33.4	PLA 寄存器	750
33.4.1	PLA 控制寄存器 (PLA_CR)	750
33.4.2	PLA 标志寄存器 (PLA_SR)	751
33.4.3	PLU0 配置寄存器 (PLA_CFGR0)	753
33.4.4	PLU1 配置寄存器 (PLA_CFGR1)	754
33.4.5	PLU2 配置寄存器 (PLA_CFGR2)	755
33.4.6	PLU3 配置寄存器 (PLA_CFGR3)	756
34	调试接口 (Debug Support)	758
34.1	概述	758
34.2	参考文档.....	758
34.3	SWD 接口	759
34.3.1	SWD 接口引脚.....	759

34.3.2	SWD 引脚分配	759
34.4	SWD 协议	759
34.4.1	协议简介	759
34.4.2	SWD 协议序列	759
34.4.3	SW-DP 状态机（复位、空闲状态、ID 代码）	760
34.4.4	DP 和 AP 的读写访问	760
34.4.5	SW-DP 寄存器	761
34.4.6	SW-AP 寄存器	762
34.4.7	内核调试	762
34.5	BPU（断点单元）	763
34.5.1	BPU 功能	763
34.6	DWT（数据观察点）	763
34.6.1	DWT 功能	763
34.6.2	DWT 程序计数器采样寄存器	763
34.7	MCU 调试组件	763
34.7.1	对低功耗模式的调试支持	764
34.7.2	对定时器、看门狗和 I2C 的调试支持	764
34.8	DBG 寄存器	764
34.8.1	数据寄存器（DBG_DAT）	764
34.8.2	控制寄存器（DBG_CR）	765
34.8.3	APB1 冻结寄存器（DBG_APB_FZ1）	765
34.8.4	APB2 冻结寄存器（DBG_APB_FZ2）	766
35	Cortex®-M0+内核寄存器	768
35.1	系统控制寄存器	768
35.1.1	应用中断和复位控制寄存器	768
35.1.2	系统控制寄存器（SCR）	768
35.2	SysTick 寄存器	769
35.2.1	SysTick 控制和状态寄存器（SYST_CSR）	769
35.2.2	SysTick 重载值寄存器（SYST_RVR）	770
35.2.3	SysTick 当前计数值寄存器（SYST_CVR）	770
35.2.4	SysTick 校准值寄存器（SYST_CALIB）	771
35.3	NVIC 寄存器	771
35.3.1	NVIC 中断使能寄存器（NVIC_ISER）	771
35.3.2	NVIC 中断清使能寄存器（NVIC_ICER）	772
35.3.3	NVIC 中断挂起使能寄存器（NVIC_ISPR）	772
35.3.4	NVIC 中断挂起清寄存器（NVIC_ICPR）	772
35.3.5	NVIC 优先级寄存器（NVIC_IPRn）	773
36	器件电子签名	774

表目录

表 3-1	存储器映射.....	40
表 3-2	存储器地址范围.....	40
表 3-3	外设寄存器地址范围.....	41
表 3-4	启动配置.....	43
表 4-1	Flash 存储器组成.....	46
表 4-2	HCLK 时钟频率与 Flash 读取访问等待周期的对应关系.....	47
表 4-3	安全保护机制对 Flash 擦除操作的影响.....	49
表 4-4	安全保护机制对 Flash 编程操作的影响.....	50
表 4-5	选项字节组成.....	52
表 4-6	RDP 保护等级配置.....	58
表 4-7	PCROP 保护区域配置.....	59
表 4-8	WRP 保护区域配置.....	60
表 4-9	Flash 中断.....	63
表 5-1	低功耗模式概述.....	82
表 5-2	低功耗模式下可用外设.....	82
表 5-3	低功耗模式进入和退出方式.....	84
表 5-4	Sleep 说明.....	85
表 5-5	Stop 说明.....	86
表 6-1	时钟关系表.....	93
表 6-2	HSE 时钟源.....	94
表 6-3	HSE 时钟配置.....	95
表 6-4	LSE 时钟源.....	97
表 6-5	LSE 时钟配置.....	98
表 6-6	外设异步时钟来源.....	100
表 7-1	GPIO 端口配置表 ⁽¹⁾	135
表 9-1	互联外设.....	151
表 10-1	DMA 内部信号.....	156
表 10-2	DMA 地址对齐.....	158
表 10-3	DMA 地址递增.....	158
表 10-4	DMA 中断.....	160
表 11-1	DMAMUX 信号.....	170
表 11-2	DMAMUX 输出请求信号.....	171
表 11-3	DMAMUX 请求生成触发信号.....	171
表 11-4	DMAMUX 中断.....	174
表 12-1	中断向量表.....	178
表 13-1	EXTI 信号说明.....	181
表 13-2	EVG 信号说明.....	181
表 13-3	EXTI 通道及信号对应表.....	182
表 13-4	EXTI 屏蔽功能.....	183
表 13-5	EXTI 输入信号类型与寄存器的关系表.....	184

表 15-1	CRC 内部信号.....	198
表 16-1	触发与转换开始之间的延迟.....	207
表 16-2	触发极性.....	214
表 16-3	外部触发源.....	214
表 16-4	数据对齐方式和分辨率.....	217
表 16-5	模拟看门狗监测数据格式.....	220
表 16-6	ADC 中断.....	225
表 17-1	存储在 TKA RAM 中的 TK 数据的排列(TKM=0).....	241
表 17-2	存储在 TKA RAM 中的 TK 数据的排列(TKM=1).....	242
表 17-3	存储在 TKA RAM 中的 TK 数据的排列(TKM=2).....	242
表 17-4	存储在 TKA RAM 中的 TK 数据的排列(TKM=3).....	243
表 18-1	DAC 输入/输出引脚.....	261
表 18-2	COMP 中断控制.....	263
表 18-3	DAC 触发选择.....	265
表 18-4	通道输出模式汇总.....	268
表 18-5	DAC 中断.....	274
表 19-1	LED 矩阵驱动对应显示配置表.....	306
表 20-1	定时器输出行为与 BRK/BRK2 输入.....	364
表 20-2	断路保护解除条件.....	365
表 20-3	计数方向与编码器信号的关系.....	371
表 20-4	TIMx 内部触发连接.....	389
表 20-5	具有断路功能的互补通道 OCx 和 OCxN 的输出控制位.....	405
表 23-1	LPTIM 输入输出引脚.....	506
表 23-2	LPTIM 内部信号.....	507
表 23-3	LPTIM 外部触发连接.....	507
表 23-4	预分频器分频比.....	509
表 23-5	计数器方向与编码器信号的关系.....	515
表 23-6	低功耗模式对 LPTIM 的影响.....	520
表 23-7	中断事件.....	521
表 28-1	I2C 引脚.....	548
表 28-2	I2C 内部信号.....	548
表 28-3	I2C 规范时序要求.....	551
表 28-4	I2C_KCLK=18MHz 配置示例.....	553
表 28-5	低功耗模式说明.....	569
表 28-6	中断请求和状态清除.....	569
表 29-1	USART 功能列表.....	583
表 29-2	USART1 框图信号说明表.....	584
表 29-3	USART2\3 框图信号说明表.....	585
表 29-4	USART 校验控制.....	589
表 29-5	采样数据和噪声检测 (ONEBIT=0).....	599
表 29-6	BRR [3:0] = 0000 时的 USART 接收容差.....	602

表 29-7	BRR [3:0] ≠ 0000 时的 USART 接收容差	603
表 29-8	USART 中断请求表	623
表 30-1	LPUART 支持功能一览	658
表 30-2	LPUART_KCLK_PRES=32.768KHz 时的波特率误差	670
表 30-3	LPUART 的接收容差	671
表 30-4	LPUART 校验控制	673
表 30-5	低功耗模式说明	682
表 30-6	LPUART 中断请求表	683
表 31-1	SPI 引脚	711
表 31-2	SPI 内部信号	712
表 31-3	NSS 配置 ⁽¹⁾	714
表 31-4	通信模式配置	714
表 31-5	SPI 中断	721
表 32-1	RTC 输入/输出引脚	732
表 32-2	RTC 内部输入/输出信号	732
表 33-1	PLA_CFGRx.MXA	747
表 33-2	PLA_CFGRx.MXB	748
表 33-3	LUT 真值表	749
表 34-1	SWD 接口引脚分配	759
表 34-2	数据包请求	759
表 34-3	ACK 响应	760
表 34-4	数据传输	760
表 34-5	SW-DP 寄存器	761
表 34-6	SW-AP 寄存器	762
表 34-7	内核调试寄存器	762

图目录

图 3-1	系统架构图.....	37
图 3-2	Flash 物理区域划分.....	39
图 4-1	RDP 保护等级变更.....	58
图 5-1	电源结构框图.....	78
图 5-2	POR/PDR 和 BOR 阈值.....	79
图 5-3	PVD 阈值.....	80
图 5-4	PVD 滤波.....	81
图 5-5	低功耗模式转换框图.....	81
图 6-1	时钟树.....	94
图 6-2	TIM4 在捕获模式下的频率测量.....	103
图 7-1	I/O 基本结构.....	135
图 7-2	I/O 引脚输入配置（浮空/上拉/下拉）.....	139
图 7-3	I/O 引脚输出配置.....	140
图 7-4	I/O 引脚复用功能配置.....	141
图 7-5	I/O 引脚高阻态模拟配置.....	142
图 10-1	DMA 框图.....	156
图 11-1	DMAMUX 框图.....	170
图 11-2	DMAMUX 通道事件生成.....	172
图 11-3	DMAMUX 通道请求生成.....	173
图 13-1	EXTI 框图.....	181
图 13-2	可配置事件&直接事件的事件输出框图.....	184
图 13-3	可配置事件的中断输出框图.....	185
图 13-4	可配置事件的唤醒输出框图.....	185
图 13-5	直接通道的唤醒信号输出框图.....	186
图 13-6	EXTI 多路复用器.....	186
图 15-1	CRC 框图.....	198
图 15-2	CRC 计算单元框图.....	199
图 15-3	输入输出数据反转框图.....	199
图 16-1	ADC 结构框图.....	203
图 16-2	ADC 校准时序.....	205
图 16-3	校准系数加载时序.....	205
图 16-4	使能/禁止 ADC.....	206
图 16-5	ADC 时钟结构.....	206
图 16-6	ADC 输入通道.....	208
图 16-7	ADC 转换模式.....	210
图 16-8	ADC 转换时序.....	213
图 16-9	停止正在进行的转换.....	213
图 16-10	单次转换模式，软件触发.....	215
图 16-11	连续转换模式，软件触发.....	215

图 16-12	单次转换模式，硬件触发.....	216
图 16-13	连续转换，硬件触发.....	216
图 16-14	ADC 参考源选择.....	217
图 16-15	数据溢出处理示例.....	218
图 16-16	等待模式转换（单次、软件触发）.....	220
图 16-17	ADC_WDx_OUT 信号生成.....	221
图 16-18	ADC_AWDGx_OUT 信号生成 (软件不清零 AWDGx_EVT 标志).....	221
图 16-19	模拟看门狗阈值更新.....	222
图 16-20	温度传感器和 VBGR 通道结构图.....	223
图 16-21	VCC 通道采集结构图.....	225
图 17-1	ATK 结构框图.....	240
图 18-1	SAC 结构框图.....	260
图 18-2	COMP 迟滞.....	262
图 18-3	COMP 输出滤波.....	262
图 18-4	DAC 单通道模式下的数据寄存器.....	264
图 18-5	DAC 双通道模式下的数据寄存器.....	264
图 18-6	关闭触发（TEN=0）时的转换时序.....	265
图 18-7	LFSR 寄存器计算算法.....	267
图 18-8	LFSR 产生波形的 DAC 转换（使能软件触发）.....	267
图 18-9	生成 DAC 三角波.....	268
图 18-10	生成三角波波形的 DAC 转换（使能软件触发）.....	268
图 19-1	LCD/LED 控制器结构框图.....	305
图 19-2	LED 矩阵图.....	306
图 19-3	LED7*8 点阵图.....	309
图 19-4	LED7*7 点阵图.....	310
图 19-5	LED6*7 点阵图.....	311
图 19-6	LED6*6 点阵图.....	311
图 19-7	LED5*5 点阵图.....	312
图 19-8	LED4*4 点阵图.....	312
图 19-9	LCD 模式控制器结构框图.....	315
图 19-10	LCD 驱动波形框架.....	316
图 19-11	LCD 电压发生器.....	317
图 19-12	1/4Duty, 1/3Bias 驱动波形示例.....	318
图 19-13	LED 和 LCD 驱动应用流程示例.....	321
图 20-1	高级控制定时器框图.....	327
图 20-2	预分频器分频由 1 变为 2 时的计数器时序图.....	329
图 20-3	预分频器分频由 1 变为 4 时的计数器时序图.....	329
图 20-4	计数器时序图，1 分频内部时钟.....	330
图 20-5	计数器时序图，2 分频内部时钟.....	331
图 20-6	计数器时序图，4 分频内部时钟.....	331
图 20-7	计数器时序图，N 分频内部时钟.....	332

图 20-8 计数器时序图, ARPE=0 时更新事件 (TIM1_ARR 未预装载)	332
图 20-9 计数器时序图, ARPE=1 时更新事件 (TIM1_ARR 已预装载)	333
图 20-10 计数器时序图, 1 分频内部时钟	334
图 20-11 计数器时序图, 2 分频内部时钟	334
图 20-12 计数器时序图, 4 分频内部时钟	335
图 20-13 计数器时序图, N 分频内部时钟	335
图 20-14 计数器时序图, 未使用重复计数器时更新事件	336
图 20-15 计数器时序图, 1 分频内部时钟, TIM1_ARR=0x06	337
图 20-16 计数器时序图, 2 分频内部时钟	337
图 20-17 计数器时序图, 4 分频内部时钟, TIM1_ARR=0x36	338
图 20-18 计数器时序图, N 分频内部时钟	338
图 20-19 计数器时序图, ARPE=1 时的更新事件 (计数器下溢)	339
图 20-20 计数器时序图, ARPE=1 时的更新事件 (计数器上溢)	339
图 20-21 不同模式和 TIM1_RCR 寄存器设置下的更新频率示例	341
图 20-22 外部触发输入模块	341
图 20-23 TIM1 ETR 输入电路	342
图 20-24 正常模式下的控制电路, 1 分频内部时钟	342
图 20-25 TI2 外部时钟连接示例	343
图 20-26 外部时钟模式 1 下的控制电路	344
图 20-27 外部触发输入模块	344
图 20-28 外部时钟模式 2 下的控制电路	345
图 20-29 捕获/比较通道 (示例: 通道 1 输入阶段)	345
图 20-30 捕获/比较通道 1 主电路	346
图 20-31 捕获/比较通道的输出阶段 (通道 1、通道 2 和通道 3)	346
图 20-32 捕获/比较通道的输出阶段 (通道 4)	347
图 20-33 捕获/比较通道的输出阶段 (通道 5 和通道 6)	347
图 20-34 PWM 输入模式时序	350
图 20-35 输出比较模式, 翻转 OC1	351
图 20-36 边沿对齐模式的 PWM 波形 (ARR=8)	353
图 20-37 中心对齐模式 PWM 波形 (ARR=8)	354
图 20-38 50%占空比时产生的 2 个相移 PWM 信号	355
图 20-39 通道 1 和通道 2 上的组合 PWM 模式	356
图 20-40 三相组合 PWM 信号 (每个周期多个触发脉冲)	357
图 20-41 带死区插入的互补输出	358
图 20-42 延迟时间大于负脉冲宽度的死区波形	358
图 20-43 延迟时间大于正脉冲宽度的死区波形	358
图 20-44 断路和断路 2 电路概述	361
图 20-45 响应 BRK 上的断路事件的不同输出行为 (OSS1=1)	363
图 20-46 BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSS1=1)	364
图 20-47 BRK 使能后的 PWM 输出状态 (OSS1=0)	364
图 20-48 输出重定向 (图中未显示 BRK2 请求)	366

图 20-49 清除 TIM1 的 OCxREF	367
图 20-50 COM 事件生成 6 步 PWM 的示例 (OSSR=1)	368
图 20-51 单脉冲模式示例.....	369
图 20-52 可再触发单脉冲模式.....	370
图 20-53 编码器接口模式下的计数器工作示例.....	372
图 20-54 TI1FP1 极性反相时的编码器接口模式示例.....	372
图 20-55 测量 3 个信号上边沿之间的时间间隔.....	374
图 20-56 霍尔传感器接口的示例.....	375
图 20-57 复位模式下的控制电路.....	376
图 20-58 门控模式下的控制电路.....	377
图 20-59 触发模式下的控制电路.....	378
图 20-60 外部时钟模式 2+触发模式下的控制电路.....	379
图 21-1 TIM2/3/4 通用控制定时器框图	423
图 21-2 预分频器分频由 1 变为 2 计数器时序图.....	425
图 21-3 预分频器分频由 1 变为 4 计数器时序图.....	425
图 21-4 计数器时序图, 1 分频内部时钟.....	426
图 21-5 计数器时序图, 2 分频内部时钟.....	427
图 21-6 计数器时序图, 4 分频内部时钟.....	427
图 21-7 计数器时序图, N 分频内部时钟.....	428
图 21-8 计数器时序图, ARPE=0 时更新事件 (未预装载)	428
图 21-9 计数器时序图, ARPE=1 时更新事件 (已预装载)	429
图 21-10 计数器时序图, 1 分频内部时钟.....	430
图 21-11 计数器时序图, 2 分频内部时钟.....	430
图 21-12 计数器时序图, 4 分频内部时钟.....	431
图 21-13 计数器时序图, N 分频内部时钟.....	431
图 21-14 计数器时序图, 未使用重复计数器时更新事件.....	432
图 21-15 计数器时序图, 1 分频内部时钟, TIM2/3/4_ARR=0x06.....	433
图 21-16 计数器时序图, 2 分频内部时钟.....	434
图 21-17 计数器时序图, 4 分频内部时钟, TIM2/3/4_ARR=0x36.....	434
图 21-18 计数器时序图, N 分频内部时钟.....	435
图 21-19 计数器时序图, ARPE=1 时的更新事件 (计数器下溢)	435
图 21-20 计数器时序图, ARPE=1 时的更新事件 (计数器上溢)	436
图 21-21 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例.....	437
图 21-22 正常模式下的控制电路, 1 分频内部时钟.....	438
图 21-23 TI2 外部时钟连接示例.....	438
图 21-24 外部时钟模式 1 下的控制电路.....	439
图 21-25 外部触发输入模块.....	439
图 21-26 外部时钟模式 2 下的控制电路.....	440
图 21-27 捕获比较通道 1 输入阶段电路.....	441
图 21-28 捕获比较通道 1 主电路.....	441
图 21-29 捕获比较通道 1 输出阶段电路.....	442

图 21-30 输入捕获模式时序.....	444
图 21-31 输出比较模式，翻转 OC1	445
图 21-32 边沿对齐的 PWM 模式 (ARR=8)	447
图 21-33 中心对齐的 PWM 模式 (ARR=8)	448
图 21-34 50%占空比时产生的 2 个相移 PWM 信号.....	449
图 21-35 通道 1 和通道 2 上的组合 PWM 模式.....	451
图 21-36 单脉冲模式.....	452
图 21-37 可重触发单脉冲模式.....	454
图 21-38 清除 TIM2/3/4 的 OCxREF.....	455
图 21-39 测量 3 个信号上边沿之间的时间间隔.....	456
图 21-40 复位模式下的控制电路.....	457
图 21-41 门控模式下的控制电路.....	458
图 21-42 触发模式下的控制电路.....	458
图 21-43 外部时钟模式 2+触发模式下的控制电路.....	460
图 21-44 主从定时器示例.....	460
图 21-45 使用定时器 x 的 OC1REF 信号对定时器 y 实施门控控制.....	461
图 21-46 使用定时器 x 的使能信号对定时器 y 实施门控控制.....	462
图 21-47 使用定时器 x 的更新信号对定时器 y 实施触发控制.....	463
图 21-48 使用定时器 x 的使能信号对定时器 y 实施触发控制.....	463
图 21-49 使用定时器 x 的 TI1 输入触发定时器 x 和定时器 y.....	464
图 22-1 基本定时器框图.....	494
图 22-2 预分频器分频由 1 变为 2 时的计数器时序图.....	496
图 22-3 预分频器分频由 1 变为 4 时的计数器时序图.....	496
图 22-4 计数器时序图，1 分频内部时钟.....	497
图 22-5 计数器时序图，N 分频内部时钟.....	497
图 22-6 计数器时序图，ARPE=0 时更新事件（未预装载）.....	498
图 22-7 计数器时序图，ARPE=1 时更新事件（已预装载）.....	498
图 22-8 正常模式下的控制电路，1 分频内部时钟.....	499
图 23-1 低功耗定时器框图.....	506
图 23-2 干扰滤波器时序图.....	508
图 23-3 LPTIM 输出波形，单次计数模式配置.....	510
图 23-4 LPTIM 输出波形，单次计数模式配置且激活置一单触发模式（WAVE 位置 1）.....	510
图 23-5 LPTIM 输出波形、连续计数模式配置.....	511
图 23-6 生成波形.....	512
图 23-7 正交编码器模式下的计数序列.....	516
图 23-8 非交编码器模式下的计数序列.....	517
图 23-9 非交编码器背靠背波形无法解码.....	517
图 23-10 非交编码器通道 1 缺失.....	518
图 23-11 非交编码器通 2 道缺失.....	518
图 23-12 非交编码器无法计数波形.....	519
图 23-13 非交编码模式错误中断标志置 1.....	519

图 25-1	IRTIM 内部硬件连接.....	534
图 26-1	独立看门狗结构框图.....	535
图 26-2	独立看门狗超时时间图.....	537
图 27-1	窗口看门狗结构框图.....	542
图 27-2	窗口看门狗超时时间图.....	544
图 28-1	I2C 框图.....	548
图 28-2	I2C 总线协议.....	550
图 28-3	I2C 初始化流程.....	550
图 28-4	I2C_TIMINGR 寄存器配置生成 SCL 时序	552
图 28-5	7 位地址读写过程.....	554
图 28-6	10 位地址读写过程.....	554
图 28-7	数据接收.....	556
图 28-8	数据发送.....	556
图 28-9	从模式使能时钟低电平延长发送数据.....	558
图 28-10	从模式禁止时钟低电平延长发送数据.....	559
图 28-11	从模式使能时钟低电平延长接收数据.....	559
图 28-12	从模式禁止时钟低电平延长接收数据.....	560
图 28-13	HEAD10R 置 1 或清 0 在 10 位地址模式先发送再接收数据过程中的作用	561
图 28-14	主模式自动结束模式发送数据时序.....	562
图 28-15	主模式软件结束模式发送数据时序.....	563
图 28-16	主模式自动结束模式接收数据时序.....	564
图 28-17	主模式软件结束模式接收数据时序.....	564
图 28-18	低功耗模式唤醒时序.....	565
图 28-19	低功耗模式未唤醒时序.....	566
图 28-20	I2C 总线仲裁.....	568
图 29-1	USART 框图.....	584
图 29-2	USART2\3 框图.....	585
图 29-3	字符组成示意图.....	588
图 29-4	可编程停止位示例.....	591
图 29-5	发送过程 TC 和 TXE 的动作 (FIFO 禁用)	593
图 29-6	8 倍/16 倍过采样时的起始位检测.....	594
图 29-7	USART_KCLK 时钟分频框图(USART1)	598
图 29-8	数据采样-8 倍过采样 (ONEBIT=0)	599
图 29-9	数据采样-16 倍过采样 (ONEBIT=0)	599
图 29-10	波特率自动检测的四种模式.....	605
图 29-11	使用空闲帧检测的静默模式.....	607
图 29-12	使用地址匹配检测时的静默模式.....	608
图 29-13	同步通信主模式下的连接例.....	610
图 29-14	同步主模式数据-时钟示意图 1 (M[1:0]="00")	610
图 29-15	同步主模式数据-时钟示意图 2 (M[1:0]="01")	611
图 29-16	同步从模式数据-时钟示意图 (M[1:0]="00")	612

图 29-17	IrDA 功能使能时 USART 内部结构框图.....	614
图 29-18	标准模式下 IrDA SIR 数据调制	615
图 29-19	USART 通过 DMA 发送示意图.....	617
图 29-20	USART 通过 DMA 接收数据示例.....	618
图 29-21	两个 USART 间串行通信连接例.....	619
图 29-22	RTS 流控	619
图 29-23	CTS 流控示例	619
图 29-24	检测到有效唤醒事件.....	622
图 29-25	未检测到有效唤醒事件.....	622
图 30-1	LPUART 框图	659
图 30-2	LPUART 字长详解	661
图 30-3	配置停止位.....	663
图 30-4	发送过程 TC 和 TXE 的动作（FIFO 禁止时）	665
图 30-5	LPUART_KCLK 时钟分频框图.....	669
图 30-6	多机通信连接示意图.....	671
图 30-7	使用空闲帧检测的方式从静默模式唤醒.....	672
图 30-8	使用地址匹配检测时的静默模式.....	673
图 30-9	LPUART 通过 DMA 发送示意图	676
图 30-10	LPUART 通过 DMA 接收示意图	677
图 30-11	两个 LPUART 间串行通信连接例.....	678
图 30-12	RTS 动作示例	678
图 30-13	CTS 动作示例	679
图 30-14	RS485 DE 信号发送/接收控制时序	680
图 30-15	检测到有效唤醒事件.....	682
图 30-16	未检测到有效唤醒事件.....	682
图 31-1	SPI 框图.....	711
图 31-2	单主单从应用.....	712
图 31-3	一主多从应用.....	712
图 31-4	多主多从应用.....	713
图 31-5	从机应用.....	714
图 31-6	通信模式时序.....	715
图 31-7	主从模式 BSY 状态	717
图 31-8	主从模式通信时序.....	720
图 32-1	RTC 结构框图	731
图 33-1	PLA 顶层框图	746
图 33-2	PLA 内部框图	746
图 34-1	MCU 调试支持框图.....	758

1 文档约定 (Documentation conventions)

1.1 寄存器列表符号定义

读/写 (rw)	软件可以读写这些位。
只读 (ro)	软件只能读取这些位。
只写 (wo)	软件只能写入该位。读取该位时将返回复位值。
读取/写 1 清零 (rc_w1)	软件可以读取该位，也可以通过写入“1”将该位清零。写入“0”对该位的值无影响。
读取/写 0 清零 (rc_w0)	软件可以读取该位，也可以通过写入“0”将该位清零。写入“1”对该位的值无影响。
读取/写清零 (rc_w)	软件可以读取该位,写该位时，将自动清零。写入具体值不重要。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将自动清零。写入任何值对该位的值无影响。
读取/读取置位 (rs_r)	软件可以读取该位。读取该位时，将自动置“1”。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置“1”。写入“0”对该位的值无影响。
读/仅可写入一次 (rwo)	软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。
只读，写触发 (rt_w1)	软件可以读取该位。写入“1”时，将触发事件，但不会影响该位的值。
保留 (Res.)	保留位，必须保持复位值。

1.2 术语及定义

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

字	32 位数据。
半字	16 位数据。
字节	8 位数据。
SWD	串行线调试协议。

IAP（在应用中编程）	IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
ICP（在线编程）	ICP 是指可以在器件安装于用户应用电路板上时使用 SWD 协议或自举程序对微控制器的 Flash 进行编程。
AHB	高级高性能总线。
APB	高级外设总线。
OBL	选项字节加载器
Run	运行模式。
Sleep	睡眠模式。
Stop	停止模式。
HSE	外部快速晶振/谐振器。
LSE	外部慢速晶振/谐振器。
HSI	内部快速振荡器。
LSI	内部慢速振荡器。
MSI	内部低速振荡器。
PLL	锁相环电路。
PVD	可编程电压检测。
BOR	欠压复位。
RTC	实时时钟计数器。
USART	通用异步同步收发器。
PLA	可编程逻辑阵列。

2 产品简介 (Product Introduction)

TM32G0 系列产品搭载高性能 ARM[®] Cortex[®] M0+内核的 32 位微控制器, 最高工作频率可达 48MHz, 具有高抗干扰, 可 5V 工作, 并内置大容量存储器和丰富的外设功能模块。

TM32G0 系列产品集成多种功能在芯片上, 包括最大 250K 字节的闪存(Flash) 程序存储器, 最大 32K 字节 SRAM, 8 阶欠压复位(BOR), 32 阶低电压检测(PVD), 智能组合模拟模块 (SAC), 多路标准 USART 和低功耗通用异步收发器 (LPUART), 多路增强型定时器和低功耗定时器, 实时计数器(RTC), 多通道的 12 位模数转换器(ADC), 高灵敏度的触控电路(ATK), LCD/LED 驱动器电路, 2 路 SPI 接口, 2 路 I2C 接口, 硬件除法器, 硬件循环冗余校验(CRC)电路, 多通道直接存储器访问控制器(DMA), 看门狗定时器(WDG)和可编程逻辑阵列 (PLA)。它的高可靠性和低功耗的特性, 可广泛适用于消费电子及家用电器产品。

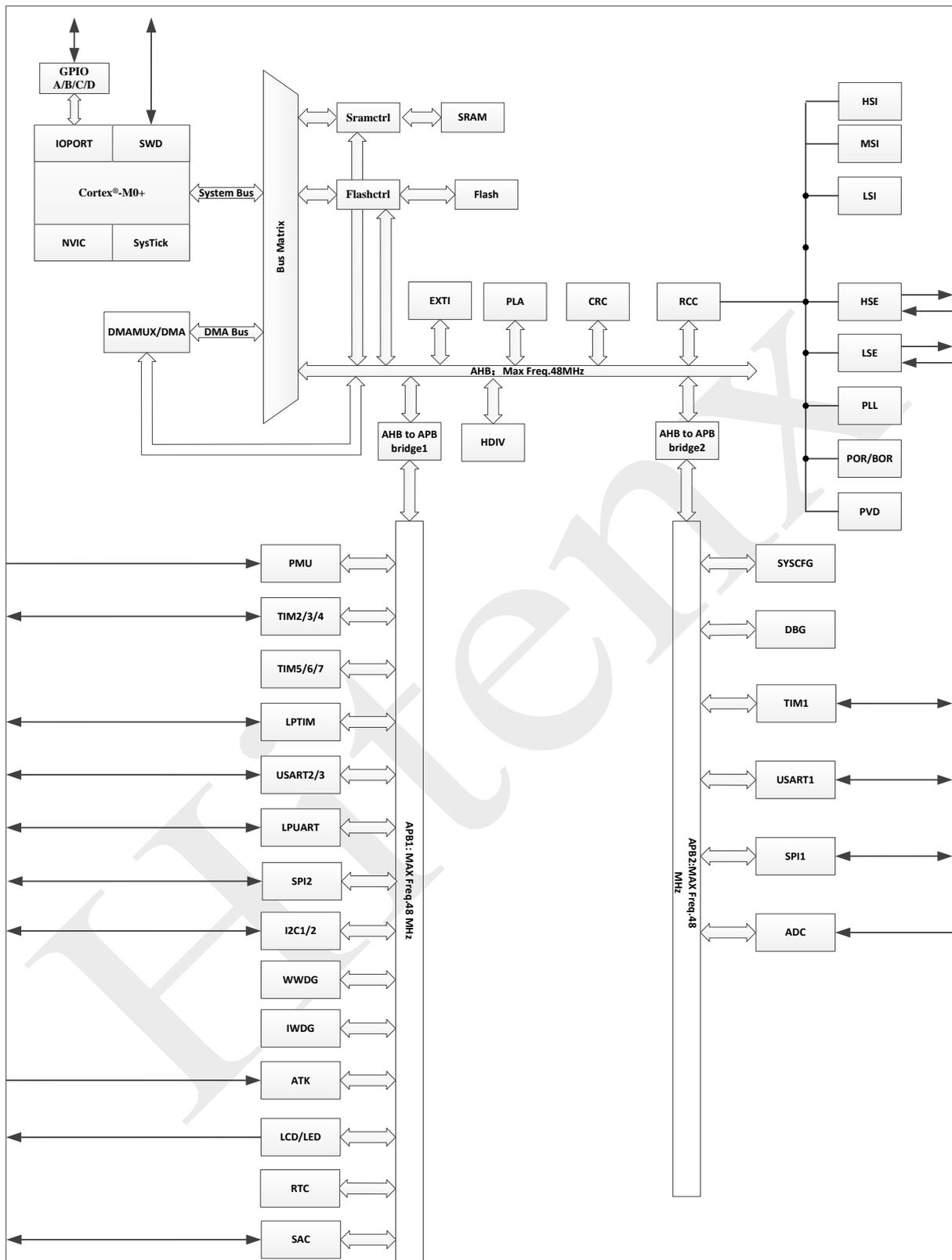
3 存储器与总线架构 (Memory and bus architecture)

3.1 系统架构

主系统包括:

- 两个主机:
 - Cortex® -M0+ core
 - DMA
- 三个从机:
 - SRAM
 - Flash
 - AHB: 带有 AHB-to-APB 的总线桥, 连接 AHB 和 APB 所有外设

所有外设通过多层 AHB 总线架构连接, 具体如下图:

图 3-1 系统架构图


3.1.1 System Bus

该总线用于将 Cortex®-M0+内核连接到 Bus Matrix, 内核通过此总线进行取指、

数据操作以及访问 AHB/APB 外设。

3.1.2 DMA Bus

该总线用于将 DMA 的 AHB 接口连接到 Bus Matrix，DMA 通过此总线访问 Flash、SRAM 以及 AHB/APB 外设。

3.1.3 Bus Matrix

总线矩阵包含两个主机（CPU、DMA）和三个从机（Flash、SRAM、带有 AHB-to-APB 总线桥的 AHB）；使用 Round Robin 算法管理 Cortex®-M0+内核的 System Bus 和 DMA Bus 的访问仲裁。

3.1.4 AHB to APB 总线桥

AHB to APB 总线桥用于连接 AHB 和 2 条 APB 总线，可以实现 AHB 和 APB 总线的同步；APB1 和 APB2 的最高工作频率可达 48MHz。

3.2 存储器

3.2.1 SRAM

SRAM 总容量最大支持 32KB，支持按字节（8 位）、半字（16 位）或全字（32 位）三种读写操作。字节操作的地址必须按字节对齐，半字操作的目标地址必须按半字对齐（地址最低位为 1'b0），字操作的地址必须按字对齐（地址最低两位为 2'b00）。如果读写操作的目标地址没有按照位宽规定对齐，该操作无效，并且系统会产生 HardFault 错误中断。本控制器支持以最高系统时钟频率进行读写操作，无需插入等待周期。

3.2.2 Flash

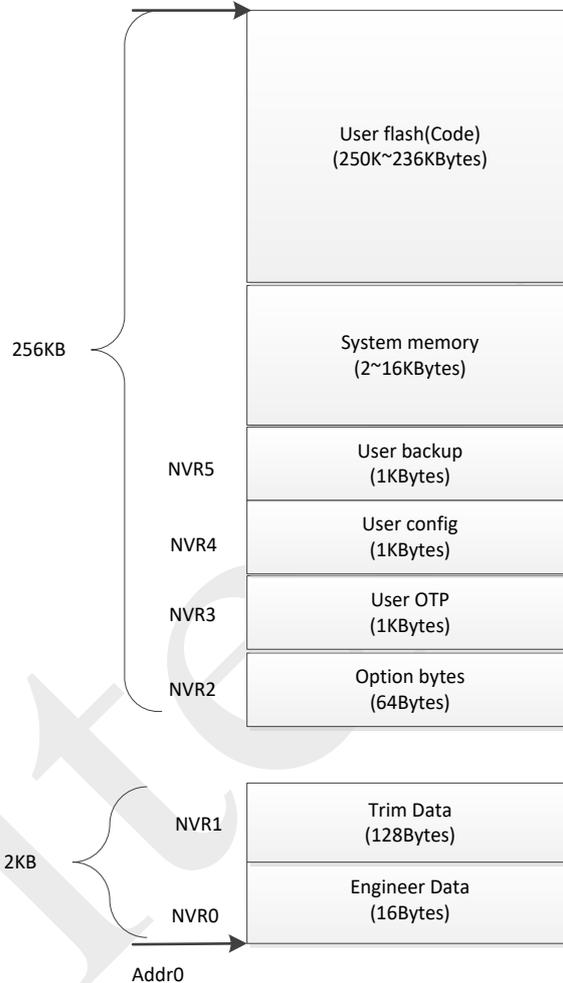
Flash 按照功能分为如下区域：

- User flash 代码区：包含应用程序和用户数据，由用户管理；
- System memory 区：包含 Bootloader 和部分重要 API；
- Engineer 数据区：包含 UID 等工厂数据；
- Trim 数据区：包含 BGR 等出厂校准数据
- optionbytes 区：包含外设和存储器保护配置的选项字节；
- 用户 OTP 区：一次可编程区域；
- 用户 Config 区：用户用于配置一些特殊数据；
- 用户 backup 区：用户用于备份特殊数据。

Flash 接口根据 AHB 协议实现指令访问和数据访问。它并支持通过 Flash 寄存器控制 Flash 操作（编程/擦除）。

Flash 物理逻辑地址划分如下：

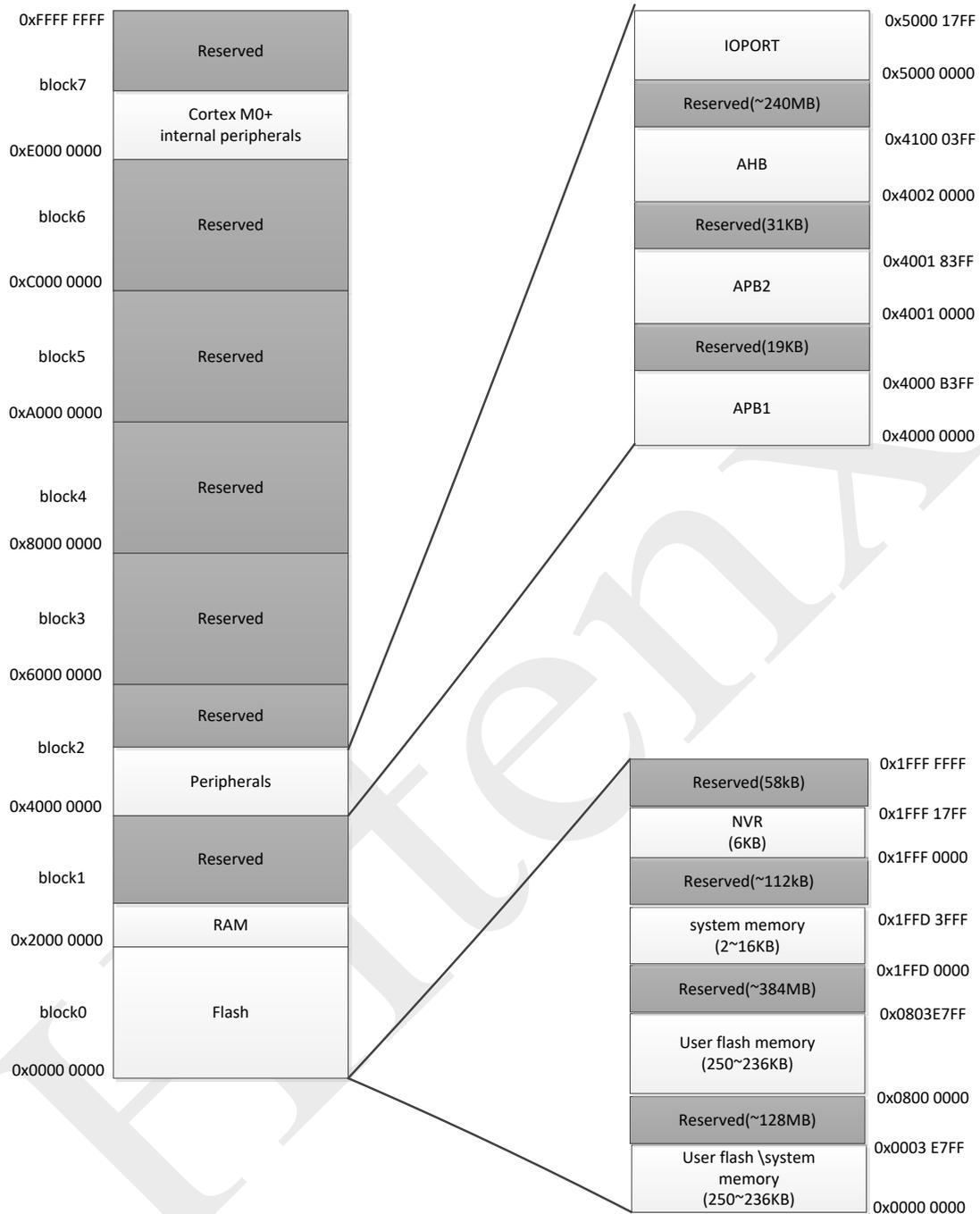
图 3-2 Flash 物理区域划分



3.3 存储器映射

程序存储器、数据存储器、寄存器和 I/O 端口排列在一个地址连续的 4 GB 地址空间内。

数据按小端格式保存在存储器中，地址最小的存储单元保存该字的最低位数据，可寻址的存储空间分为 8 个块，每个块为 512 MB。

表 3-1 存储器映射


未分配给片上存储器和外设的所有存储映射区域均视为“保留区”，具体存储映射参见下表。

表 3-2 存储器地址范围

存储类型	地址范围	大小	存储内容
SRAM	0x2000 8000 - 0x3FFF FFFF	~512MB	Reserved
	0x2000 0000 - 0x2000 7FFF	32KB	SRAM
Code	0x1FFF 1800 - 0x1FFF FFFF	58KB	Reserved
	0x1FFF 1400 - 0x1FFF 17FF	1KB	User Backup

	0x1FFF 1000 - 0x1FFF 13FF	1KB	User Config
	0x1FFF 0C00 - 0x1FFF 0FFF	1KB	User OTP
	0x1FFF 0800 - 0x1FFF 083F	64B	Option Bytes
	0x1FFF 0400 - 0x1FFF 047F	128B	Trim data
	0x1FFF 0100 - 0x1FFF 010F	16B	Engineer Data
	0x1FFD 4000 - 0x1FFE FFFF	~112KB	Reserved
	0x1FFD 0000 - 0x1FFD3FFF	2~16KB	Bootloader+API(System memory), depending on BOOT_SIZE configuration
	0x0802 C000 - 0x1FFC FFFF	~384MB	Reserved
	0x0800 0000 - 0x0803 E7FF	250~236KB	User program , depending on BOOT_SIZE configuration
	0x0002 C000 - 0x07FF FFFF	~128M	Reserved
	0x0000 0000 - 0x0003 E7FF	250~236KB	User Flash, System memory depending on nBOOT_SEL configuration

表 3-3 外设寄存器地址范围

总线	地址范围	大小	外设
CPU	0xE000 0000 - 0xE00F FFFF	1MB	Cortex®-M0+ internal peripherals
IOPORT	0x5000 1000 - 0x5FFF FFFF	~256MB	Reserved
	0x5000 0C00 - 0x5000 0FFF	1KB	GPIOD
	0x5000 0800 - 0x5000 0BFF	1KB	GPIOC
	0x5000 0400 - 0x5000 07FF	1KB	GPIOB
	0x5000 0000 - 0x5000 03FF	1KB	GPIOA
AHB	0x4003 0000 - 0x4FFF FFFF	~256MB	Reserved
	0x4002 8C00 - 0x4002 FFFF	29KB	Reserved
	0x4002 8800 - 0x4002 8BFF	1KB	HDIV
	0x4002 3C00 - 0x4002 87FF	19KB	Reserved
	0x4002 3800 - 0x4002 3BFF	1KB	PLA
	0x4002 3400 - 0x4002 37FF	1KB	Reserved
	0x4002 3000 - 0x4002 33FF	1KB	CRC
	0x4002 2400 - 0x4002 2FFF	3KB	Reserved
	0x4002 2000 - 0x4002 23FF	1KB	FLASHCTRL
	0x4002 1C00 - 0x4002 1FFF	1KB	Reserved
	0x4002 1800 - 0x4002 1BFF	1KB	EXTI
	0x4002 1400 - 0x4002 17FF	1KB	Reserved
	0x4002 1000 - 0x4002 13FF	1KB	RCC
	0x4002 0C00 - 0x4002 0FFF	1KB	Reserved
	0x4002 0800 - 0x4002 0BFF	1KB	DMAMUX
	0x4002 0400 - 0x4002 07FF	1KB	Reserved
	0x4002 0000 - 0x4002 03FF	1KB	DMA
APB2	0x4001 5C00 - 0x4001 FFFF	41KB	Reserved
	0x4001 5800 - 0x4001 5BFF	1KB	DBG

总线	地址范围	大小	外设
	0x4001 3C00 - 0x4001 57FF	7KB	Reserved
	0x4001 3800 - 0x4001 3BFF	1KB	USART1
	0x4001 3400 - 0x4001 37FF	1KB	Reserved
	0x4001 3000 - 0x4001 33FF	1KB	SPI1
	0x4001 2C00 - 0x4001 2FFF	1KB	TIM1
	0x4001 2800 - 0x4001 2BFF	1KB	Reserved
	0x4001 2400 - 0x4001 27FF	1KB	ADC
	0x4001 0400 - 0x4001 23FF	8KB	Reserved
	0x4001 01B0 - 0x4001 03FF	1KB	Reserved
	0x4001 0000 - 0x4001 01AF		SYSCFG
APB1	0x4000 8400 - 0x4000 FFFF	29KB	Reserved
	0x4000 8000 - 0x4000 83FF	1KB	LPUART
	0x4000 7C00 - 0x4000 7FFF	1KB	LPTIM
	0x4000 7800 - 0x4000 7BFF	1KB	Reserved
	0x4000 7400 - 0x4000 77FF	1KB	SAC
	0x4000 7000 - 0x4000 73FF	1KB	PWR
	0x4000 5C00 - 0x4000 6FFF	5KB	Reserved
	0x4000 5800 - 0x4000 5BFF	1KB	I2C2
	0x4000 5400 - 0x4000 57FF	1KB	I2C1
	0x4000 4C00 - 0x4000 53FF	2KB	Reserved
	0x4000 4800 - 0x4000 4BFF	1KB	USART3
	0x4000 4400 - 0x4000 47FF	1KB	USART2
	0x4000 3C00 - 0x4000 43FF	2KB	Reserved
	0x4000 3800 - 0x4000 3BFF	1KB	SPI2
	0x4000 3400 - 0x4000 37FF	1KB	Reserved
	0x4000 3000 - 0x4000 33FF	1KB	IWDG
	0x4000 2C00 - 0x4000 2FFF	1KB	WWDG
	0x4000 2800 - 0x4000 2BFF	1KB	RTC
	0x4000 2400 - 0x4000 27FF	1KB	LCD/LED
	0x4000 2000 - 0x4000 23FF	1KB	ATK
	0x4000 1800 - 0x4000 1FFF	2KB	Reserved
	0x4000 1400 - 0x4000 17FF	1KB	TIM7
	0x4000 1000 - 0x4000 13FF	1KB	TIM6
	0x4000 0C00 - 0x4000 0FFF	1KB	TIM5
	0x4000 0800 - 0x4000 0BFF	1KB	TIM4
	0x4000 0400 - 0x4000 07FF	1KB	TIM3
0x4000 0000 - 0x4000 03FF	1KB	TIM2	

3.4 启动配置

TM32G07x 微控制器提供了两种启动模式，通过 BOOT0 引脚和选项字节

(BOOT_LOCK、RDP、nBOOT_SEL、nBOOT0) 进行选择；具体启动模式选择如下表：

表 3-4 启动配置

BOOT_LOCK bit	RDP	nBOOT_SEL	BOOT0 pin	nBOOT0 bit	启动模式	
0	0	0	0	x	User flash	
			1	x	Bootloader	
		1	x	1	User flash	
			x	0	Bootloader	
	1	0	0	0	x	User flash
				1	x	Bootloader
		1	1	x	1	User flash
				x	0	Bootloader
2	x	x	x	User flash		
1	x	x	x	x	User flash	

复位后，根据选择的启动模式，将对应的存储空间重映射到 0x0000 0000 地址；CPU 先从地址 0x0000 0000 获取栈顶值，然后从地址 0x0000 0004 获取代码的启动地址，开始执行程序。

使用选项字节进行启动模式更改，需要通过 POR/PDR 复位、BOR 复位或选项字节加载复位来生效选项字节，以达到更改启动模式的目的。

根据启动模式的选择，User flash 或 Bootloader 可以通过如下地址进行访问：

- 从 User flash 启动：User flash（地址开始于 0x0800 0000）的存储空间会同时被映射到地址 0x0000 0000，但原始地址 0x0800 0000 的存储空间也可以进行访问；也就是说，User Flash 的存储空间通过 0x0000 0000 或 0x0800 0000 均可以访问；
- 从 Bootloader 启动：Bootloader（地址开始于 0x1FFD 0000）的存储空间会同时被映射到地址 0x0000 0000，但原始地址 0x1FFD 0000 的存储空间也可以进行访问；

当 RDP 为等级 2 时，无论 nBOOT_SEL、nBOOT0 以及 BOOT0 引脚配置如何，均会从 User Flash 启动。

当 BOOT_LOCK（位于选项字节的 FLASH_SEC 寄存器）置位时，无论 RDP、nBOOT_SEL、nBOOT0 以及 BOOT0 引脚配置如何，均会强制从 User Flash 启动；BOOT_LOCK 配置详细描述参见 [User Flash 区强制启动](#)。

3.4.1 存储区重映射

当启动模式确定后，就可以通过 0x0000 0000 地址对重映射的存储器进行访问；软件也可以通过修改 MEM_MODE[1:0] 位（位于系统控制寄存器(SYSCFG_CR)）

对 0x0000 0000 地址的存储器映射进行更改。

3.4.2 **Bootloader**

Bootloader 位于 System memory，用于对 User flash 进行编程。

更详细的 Bootloader 信息参见 《TM32G07x Bootloader 用户手册》。

HITENX

4 Flash 存储器 (Flash)

4.1 简介

Flash 存储器连接在 AHB 总线上，由 Flash 控制器统一管理，可对存储器执行取指、读取、编程和擦除操作，并具有安全访问机制和读写保护等功能。

4.2 Flash 主要特性

- 程序和数据存储空间,其中:
 - 页大小: 1K 字节
 - NVR 区大小: 6K 字节
 - System memory 区大小: 16K 字节(默认值, 2~16KB 可配置)
 - 用户程序区大小: 236K 字节(默认值, 250~236KB 可配置)
- 32-bits 位宽读取,16/32-bits 位宽写入
- 支持页擦除
- 可配置 3 种读出保护等级 (RDP0/RDP1/RDP2)
- 2 块可配置的代码读出保护区域 (PCROP)
- 2 块可配置的写入保护区域 (WRP)
- 可配置大小的用户安全存储区域

4.3 Flash 功能描述

4.3.1 Flash 存储器组成

Flash 存储器按 32 位读操作执行读写访问，可存储指令和数据。

Flash 存储器的组成如下：

- User flash 区：该存储区用于存放用户的指令和数据，存储空间为 250~236KB，每页 1KB。
- System memory 区：该存储区的存储空间为 2~16KB，通过选项字节 boot_size 进行配置，用于存放 Bootloader 和部分重要 API。
- NVR：该存储区的存储空间为 6KB，其中它包括：
 - Engineer Data 区：该存储区用于存放芯片工厂加工、UID 等信息。
 - Option bytes 区：该存储区用于存放选项字节。
 - Trim Data 区：该存储区用于存放芯片的出厂校准参数。

- 用户 OTP 区： 该存储区用于存放用户需配置的重要信息如 SN 号，用户只能编程一次。
- 用户配置数据区： 该存储区用于存放用户需配置的重要数据，由用户写入。
- 用户备份数据区： 该存储区可用做用户数据备份，由用户写入。

在 Flash 存储器组成表中，详细列出了各区域的地址和大小。

表 4-1 Flash 存储器组成

区域	地址（默认 boot_cfg 配置）	空间大小（字节）	说明
User flash	0x0800 0000 – 0x0800 03FF	1024	Page 0

	0x0800 7C00 – 0x0800 7FFF	1024	Page 31
	0x0800 8000 – 0x0800 83FF	1024	Page 32

	0x0800 FC00 – 0x0800 FFFF	1024	Page 63
	0x0801 0000 – 0x0801 03FF	1024	Page 64

	0x0801 7C00 – 0x0801 7FFF	1024	Page 95
	0x0801 8000 – 0x0801 83FF	1024	Page 96

	0x0801 FC00 – 0x0801 FFFF	1024	Page 127
	0x0802 0000 – 0x0802 03FF	1024	Page 128

	0x0802 7C00 – 0x0802 7FFF	1024	Page 159
	0x0802 8000 – 0x0802 83FF	1024	Page 160

	0x0802 FC00 – 0x0802 FFFF	1024	Page 191
	0x0803 0000 – 0x0803 03FF	1024	Page 192
...	
0x0803 AC00 – 0x0803 AFFF	1024	Page 235	
System flash	0x1FFD 0000 – 0x1FFD 03FF	1024	Page0
	0x1FFD 0400 – 0x1FFD 07FF	1024	Page1

	0x1FFD3800~ 0x1FFD3BFF	1024	Page14
	0x1FFD3C00~ 0x1FFD3FFF	1024	Page15
NVR	0x1FFF 0000 – 0x1FFF 03FF	1024	Engineer data
	0x1FFF 0400 – 0x1FFF 07FF	1024	Trim data
	0x1FFF 0800 – 0x1FFF 0BFF	1024	Option bytes
	0x1FFF 0C00 – 0x1FFF 0FFF	64	User OTP
	0x1FFF 1000 – 0x1FFF 13FF	128	User config
	0x1FFF 1400 – 0x1FFF 17FF	16	User backup

4.3.2 Flash 读取访问等待周期

Flash 存储器连接在 AHB 总线上，读取 Flash 时使用 HCLK 时钟。当 HCLK 的时钟频率超出 Flash 存储器的工作频率时，就会造成数据读取错误，此时需要插入等待周期。

Flash 访问控制寄存器 (FLASH_ACR) 中的 LATENCY[1:0] 位域，用于配置 Flash 读取访问的等待周期，HCLK 时钟频率与 Flash 读取访问等待周期的对应关系请见下表。

表 4-2 HCLK 时钟频率与 Flash 读取访问等待周期的对应关系

频率范围	等待周期	LATENCY[1:0]配置
$HCLK \leq 20\text{MHz}$	0 HCLK	LATENCY[1:0] = 0x
$20\text{MHz} < HCLK \leq 32\text{MHz}$	1 HCLK	LATENCY[1:0] = 10
$32\text{MHz} < HCLK \leq 48\text{MHz}$	2 HCLK	LATENCY[1:0] = 11

为保证 Flash 读取访问不出现异常或错误，当要改变 HCLK 的时钟频率时，必须按照特定步骤进行配置。

- 提高 HCLK 频率的配置步骤：
 - 1) 通过配置 FLASH_ACR 寄存器中的 LATENCY[1:0] 位域，增大 Flash 读取访问的等待周期；
 - 2) 读取 LATENCY[1:0] 位域，检查等待周期已配置成功；
 - 3) 提高 HCLK 频率，可通过配置 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 SW[2:0] 位域，切换更高频率的时钟源，或通过配置 HPRE[2:0] 位域，减小系统时钟的分频值；
 - 4) 配置 SW[2:0] 位域后，必须对 RCC_CFG 寄存器中的 SWS[2:0] 位域进行检查，确认系统时钟已切换完成。
- 降低 HCLK 频率的配置步骤：
 - 1) 降低 HCLK 频率，可通过配置 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 SW[2:0] 位域，切换更低频率的时钟源，或通过配置 HPRE[2:0] 位域，增大系统时钟的分频值；
 - 2) 配置 SW[2:0] 位域后，必须对 RCC_CFGR 寄存器中的 SWS[2:0] 位域进行检查，确认系统时钟已切换完成；
 - 3) 通过 FLASH_ACR 寄存器中的 LATENCY[1:0] 位域，减小 Flash 读取访问的等待周期；
 - 4) 读取 LATENCY[1:0] 位域，检查等待周期已配置成功。

4.3.3 Flash 解锁

为防止 Flash 被意外修改，增加了保护措施，必须向特定寄存器写入密钥，才能解锁相关功能的配置权限。

4.3.3.1 Flash 控制寄存器解锁

复位后，Flash 控制寄存器（FLASH_CR）将处于写保护锁定状态。要配置 FLASH_CR 寄存器，就要先进行解锁操作。

FLASH_CR 寄存器的解锁操作，必须严格按照以下步骤顺序执行：

- 1) 向 FLASH_KEYR 寄存器写入密钥 1：0x4567 0123；
- 2) 向 FLASH_KEYR 寄存器写入密钥 2：0xCDEF 89AB；
- 3) 检查 FLASH_CR 寄存器中的 LOCK 位，当该位清 0 时，表明 FLASH_CR 寄存器已解锁。

解锁完成后，才能对 FLASH_CR 寄存器进行配置。

密钥必须严格按照顺序写入，如果出现以下情况，将产生总线错误同时触发 HardFault 中断，直到再次复位后，才能重新对 FLASH_CR 寄存器进行解锁：

- 向 FLASH_KEYR 寄存器写入错误的密钥值；
- 解锁顺序错误，先向 FLASH_KEYR 寄存器写入密钥 2：0xCDEF 89AB；
- 解锁后继续向 FLASH_KEYR 寄存器写入任意值（包括密钥）。

将 FLASH_CR 寄存器中的 LOCK 位重新置 1，能恢复 FLASH_CR 寄存器的写保护锁定状态。通过复位，也能使 FLASH_CR 寄存器恢复成写保护锁定状态。

4.3.3.2 Flash 选项字节解锁

复位后，Flash 选项字节处于写保护锁定状态，所有的选项字节加载寄存器、FLASH_CR 寄存器中的 OBL_LAUNCH 位和 OPTSTRT 位，都会被写保护。要对选项字节进行更新，就先要进行解锁操作。

Flash 选项字节的解锁操作，必须严格按照以下步骤顺序执行：

- 1) 先解锁 Flash 控制寄存器 FLASH_CR（详见：[Flash 控制寄存器解锁](#)）；
- 2) 向 FLASH_OPTKEYR 寄存器写入密钥 1：0x0819 2A3B；
- 3) 向 FLASH_OPTKEYR 寄存器写入密钥 2：0x4C5D 6E7F；
- 4) 检查 FLASH_CR 寄存器中的 OPTLOCK 位，当该位清 0 时，表明 Flash 选项字节已解锁。

解锁完成后，才能对选项字节加载寄存器及其控制位（OBL_LAUNCH 和 OPTSTRT）进行配置。

密钥必须严格按照顺序写入，如果出现以下情况，将产生总线错误同时触发 HardFault 中断，直到再次复位后，才能重新对 Flash 选项字节进行解锁：

- 向 FLASH_OPTKEYR 寄存器写入错误的密钥值；
- 解锁顺序错误，先向 FLASH_OPTKEYR 寄存器写入密钥 2：0x4C5D 6E7F；

- 解锁后继续向 FLASH_OPTKEYR 寄存器写入任意值（包括密钥）；
- 在对 FLASH_CR 寄存器解锁前，向 FLASH_OPTKEYR 寄存器写入任意值（包括密钥）。

将 FLASH_CR 寄存器中的 OPTLOCK 位重新置 1，能恢复 Flash 选项字节的写保护锁定状态。通过复位，也能使 Flash 选项字节恢复成写保护锁定状态。

当 FLASH_CR 寄存器恢复成写保护锁定状态时(LOCK 位置 1)，Flash 选项字节也会被恢复成写保护锁定状态，OPTLOCK 位将自动置 1。

4.3.4 flash 擦除操作

flash 区支持以下擦除方式：

- 页擦除（1K 字节）

Flash 存储器在执行擦除操作时，不能同时进行读取操作，需要等待存储器完成擦除操作后，读取操作才能正常进行，擦除完成后的 Flash 数据为全 1。

无论采用哪种擦除模式，都会受到 FLASH 安全保护机制的影响。

表 4-3 安全保护机制对 Flash 擦除操作的影响

SEC_PROT	写保护 WRP	代码读出保护 PCROP	擦除操作	错误标志	总线错误
0	否	否	正常擦除	无	无
	否	是	终止擦除	WRPERR 标志置 1	
	是	否			
	是	是			
1	x			无	触发 HardFault

注意：在 Flash 存储器执行擦除操作时，如果芯片发生复位，则不能保证 Flash 存储器中内容的正确性。

4.3.4.1 User flash 区页擦除步骤

对 User flash 区进行页擦除操作（1K 字节），可遵循以下步骤：

- 1) 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 标志，以确认当前没有正在执行的 Flash 操作；
- 2) 检查 FLASH_SR 寄存器，确认错误标志均已清除；
- 3) 解锁 *Flash 控制寄存器 (FLASH_CR)*，使 LOCK 位清 0；（详见：*Flash 控制寄存器解锁*）
- 4) 配置 FLASH_CR 寄存器中的 ERM[1:0]位域为 01，进入页擦除模式；
- 5) 配置 FLASH_CR 寄存器中的 PNB[7:0]位域，选择待擦除区域的页号；
- 6) 配置 FLASH_CR 寄存器中的 STRT 位置 1，启动 Flash 擦除，同时 BSY 标志将自动置 1；
- 7) 查询并等待 BSY 标志清 0，表明擦除操作已完成，此时 STRT 位也将自动

清 0;

- 8) 如果要对多个页执行擦除操作, 可重复执行步骤 5 到 7;
- 9) 配置 FLASH_CR 寄存器中的 ERM[1:0]位域为 00, 退出擦除模式;
- 10) 配置 FLASH_CR 寄存器中的 LOCK 位置 1, 恢复 FLASH_CR 寄存器的写保护锁定状态。

注意: 如果在进入擦除模式前, 未使能 HSI 时钟, 当 FLASH_CR 寄存器中的 ERM[1:0]位域配置为非 00 时, HSI 时钟源将自动使能, 该时钟源用作 Flash 的擦除时钟, 当 ERM[1:0]位域配置为 00 时, HSI 时钟源将被自动关闭。

4.3.4.2 User flash 区擦除错误

在对 User flash 区执行擦除操作的过程中, 可能会出现以下错误标志:

- 擦除序列错误标志 PGSERR:
 - 当 FLASH_CR 寄存器中的 STRT 位被置 1 时, 如果 PG 位也为 1, 则将产生擦除序列错误, FLASH_SR 寄存器中的 PGSERR 标志将置 1。
 - 当 FLASH_CR 寄存器中的 STRT 位被置 1 时, 如果 ERM[1:0] 位域为 00, 则将产生擦除序列错误, PGSERR 标志将置 1。
 - 当有错误标志 (PGSERR、SIZERR、PGAERR、WRPERR) 未被清除时, 配置 FLASH_CR 寄存器中的 STRT 位置 1, 则将产生擦除序列错误, PGSERR 标志将置 1。
- 写保护错误标志 WRPERR:

如果要擦除的区域受到安全保护机制的影响, 处于受保护的状态, 擦除操作将被终止, FLASH_SR 寄存器中的 WRPERR 标志将置 1。

4.3.5 User flash 区编程操作

对 Flash 存储器执行编程操作, 每次能写入的数据长度固定为 16 位(半字)或 32bits (字), 不支持其它长度的数据写入。

FLASH 存储器在执行编程操作时, 不能同时进行读取操作, 需要等待存储器完成编程操作后, 读取操作才能正常进行。

与 Flash 擦除操作类似, 编程操作也会受到安全保护机制的影响:

表 4-4 安全保护机制对 Flash 编程操作的影响

SEC_PROT	写保护 WRP	代码读出保护 PCROP	编程操作	错误标志	总线错误
0	否	否	正常编程	无	无
	否	是	终止编程	WRPEER 标志置 1	
	是	否			

SEC_PROT	写保护 WRP	代码读出保护 PCROP	编程操作	错误标志	总线错误
	是	是			
1		x		无	触发 HardFault

注意： 在对 Flash 存储器执行编程操作时，如果芯片发生复位，则不能保证 Flash 存储器中内容的正确性。

4.3.5.1 User flash 区编程操作步骤

对 User flash 区进行编程操作，可遵循以下步骤：

- 1) 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 标志，以确认当前没有正在执行的 Flash 操作；
- 2) 检查 FLASH_SR 寄存器，确认错误标志均已清除；
- 3) 解锁 *Flash 控制寄存器 (FLASH_CR)*，使 LOCK 位清 0（详见：*Flash 控制寄存器解锁*）；
- 4) 配置 FLASH_CR 寄存器中的 PG 为 1，进入 Flash 编程模式；
- 5) 向 Flash 目标地址写入 16 或 32bits 数据，写入后 BSY 标志将自动置 1；
- 6) 查询并等待 BSY 标志清 0，表明编程操作已完成；
- 7) 如果要对多个地址进行编程，可重复步骤 5 和 6；
- 8) 配置 FLASH_CR 寄存器中的 PG 位为 0，退出 Flash 编程模式；
- 9) 配置 FLASH_CR 寄存器中的 LOCK 位置 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

注意： 如果在进入编程模式前，未使能 HSI 时钟，当 FLASH_CR 寄存器中的 PG 位域配置为 1 时，HSI 时钟源将自动使能，该时钟源用作 Flash 的编程时钟，当 PG 位域配置为 0 时，HSI 时钟源将被自动关闭。

4.3.5.2 User flash 区编程错误

在对 User flash 区进行编程操作的过程中，可能会出现以下错误标志：

- 编程序列错误标志 PGSERR：
 - 在 FLASH_CR 寄存器中的 PG 位被置 1 前，向 Flash 地址写入数据，则将产生编程序列错误，FLASH_SR 寄存器中的 PGSERR 标志将置 1。
 - 当 FLASH_CR 寄存器中的 PG 位为 1 时，ERM [1:0]位域不为 00，则将产生编程序列错误，PGSERR 标志将置 1。
 - 当有错误标志（PGSERR、SIZERR、PGAERR、WRPERR）未被清除时，向 Flash 地址写入数据，则将产生编程序列错误，PGSERR 标志将置 1。
- 编程位宽错误标志 SIZERR：

如果向 Flash 地址写入字节，则将产生编程位宽错误，FLASH_SR 寄存器中的 SIZERR 标志将置 1。

- 编程地址未对齐错误标志 PGAERR:

如果编程操作的目标地址没有按字或半字对齐（地址最低 1 位不为 b0），则将产生编程地址未对齐错误，FLASH_SR 寄存器中的 PGAERR 标志将置 1。

- 编程错误标志 PROGERR:

当编程操作的目标地址已经被写入过值，该地址存储的数据不是全 1（32 位编程：0xFFFFFFFF，16 位编程：0xFFFF）时，如果继续向该地址写入数据，则将产生编程错误，FLASH_SR 寄存器中的 PROGERR 标志将置 1。

4.4 Option bytes 区操作说明

4.4.1 选项字节组成

选项字节存放于 Flash 存储器的 Option bytes 区，用于保存与芯片硬件功能相关的配置项，用户可根据应用对选项字节进行配置，以实现特定的硬件功能。

为了校验选项字节的正确性，在 Option bytes 区的每个字（32bits）被拆分成两部分，低 16bits 存放选项字节，高 16bits 存放选项字节的反码。

选项字节的详细组成如下表所示：

表 4-5 选项字节组成

存储地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FFF0800	Reserved		BOR_HYS		BOR_LEV			BOR_EN	RDP							
0x1FFF0804	NRST_MODE	nBOOT_SEL	BOOT0	BOOTSIZE			BOOTCFG	Reserved	Reserved	IWDG_STOP		IWDG_SW	Reserved			
0x1FFF0808	Reserved							PCROP1A_STRT								
0x1FFF080C	Reserved							PCROP1A_END								

存储地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x1FFF0810	Reserved								PCROP1B_STRT								
0x1FFF0814	Reserved								PCROP1B_END								
0x1FFF0818	Reserved								WRP1A_STRT								
0x1FFF081C	Reserved								WRP1A_END								
0x1FFF0820	Reserved								WRP1B_STRT								
0x1FFF0824	Reserved								WRP1B_END								
0x1FFF0828	BOOT_LOCK	Reserved								SEC_SIZE							

复位后，硬件会自动将 Option bytes 区中的内容，加载到寄存器里，这些寄存器被称为选项字节加载寄存器，选项字节中各控制位的作用，可查看以下寄存器的详细描述：

- *FLASH_OPTR1*：选项字节寄存器 1
- *FLASH_OPTR2*：选项字节寄存器 2
- *FLASH_PCROP1ASR*：代码读出保护区 1A 起始地址寄存器
- *FLASH_PCROP1AER*：代码读出保护区 1A 结束地址寄存器
- *FLASH_PCROP1BSR*：代码读出保护区 1B 起始地址寄存器
- *FLASH_PCROP1BER*：代码读出保护区 1B 结束地址寄存器
- *FLASH_WRP1ASR*：写保护区 1A 起始地址寄存器
- *FLASH_WRP1AER*：写保护区 1A 结束地址寄存器
- *FLASH_WRP1BSR*：写保护区 1B 起始地址寄存器
- *FLASH_WRP1BER*：写保护区 1B 结束地址寄存器
- *FLASH_SECR*：用户安全配置寄存器

4.4.2 选项字节更新

Option bytes 区与 User flash 区不同，用户不能直接对 Option bytes 区执行擦除或编程操作，而是要通过对应的选项字节加载寄存器进行配置。

对选项字节的更新遵循以下步骤：

- 1) 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 标志位，以确认当前没有正在执行的 Flash 操作；
- 2) 检查 FLASH_SR 寄存器，确认错误标志均已清除；
- 3) 解锁 Flash 选项字节的写保护，使 *Flash 控制寄存器 (FLASH_CR)* 中的 OPTLOCK 位清 0（详见：*Flash 选项字节解锁*）；
- 4) 配置选项字节加载寄存器；
- 5) 将 FLASH_CR 寄存器中的 OPTSTRT 位置 1，启动选项字节的更新，同时 BSY 标志将自动置 1；
- 6) 查询并等待 BSY 标志清 0，表明选项字节已更新完成，此时 OPTSTRT 位也将被自动清 0；
- 7) 配置 FLASH_CR 寄存器中的 OPTLOCK 位置 1，恢复选项字节的写保护锁定状态；
- 8) 此时 FLASH_CR 寄存器处于解锁状态，可根据需要配置 FLASH_CR 寄存器中的 LOCK 位置 1，将其恢复成写保护锁定状态。

在启动选项字节的更新后，将自动生成选项字节的反码，并对 Option bytes 区进行擦除，随后会将选项字节及其反码更新到 Option bytes 区中。

在完成对选项字节的更新后，直接读取 Option bytes 区，可获取更新后的选项字节值。但是更新后的选项字节此时并未生效，如果读取选项字节加载寄存器，获得的仍将是最近一次已加载生效的选项字节值。

当 FLASH_CR 寄存器中的 OPTSTRT 位置 1 时，会启动选项字节的更新，此时 Flash 控制器将自动更新 Option bytes 区。

4.4.3 选项字节加载

要使 Option bytes 区中的选项字节生效，可通过以下方式对选项字节进行加载：

- 以下复位将触发加载：
 - POR/PDR 复位
 - BOR 复位
- 配置 FLASH_CR 寄存器中的 OBL_LAUNCH 位置 1 执行加载，完成后将触发系统复位。

在选项字节加载过程中，将自动对选项字节及其反码进行校验，如果出现错误，将维持复位状态不再执行任何操作，直到发生 POR/PDR 复位。

当选项字节加载生效后，Option bytes 区中的选项字节将被自动复制到对应的加载寄存器中，读取选项字节加载寄存器，获得的总是加载生效后的选项字节值。

4.5 Flash 安全保护机制

Flash 存储器支持以下安全保护机制：

- Flash 读出保护（RDP）：对存储器的操作权限进行限制
- Flash 代码读出保护（PCROP）：禁止对受保护的区域执行读取、编程或擦除操作，仅允许取指操作。
- Flash 写入保护（WRP）：禁止对受保护的区域执行编程或擦除操作，但允许取指和读取操作。
- Flash 用户安全区域：在复位后，用户安全区域处于未保护状态，可执行取指、读取、编程和擦除操作。当安全区域使能后，该区域不可见，禁止对该区域执行任何操作，直到重新复位。

4.5.1 Flash 读出保护

4.5.1.1 RDP 保护权限

Flash 读出保护（RDP）：

RDP0：

- 从 User flash 启动
 - User flash 区：允许取指、读取、编程、擦除
 - System memory 区：允许取指(BOOTCFG=0)，允许取指、读取、编程、擦除(BOOTCFG=1)
 - NVR 区： Engineer Data 区允许读取
 - Trim Data 区允许读取
 - Option bytes 区允许读取、更新（通过选项字节加载寄存器）
 - User OTP 区允许读取、可编程一次
 - User Config 区允许读取、编程、擦除
 - User backup 区允许读取、编程、擦除
- 从 Bootloader 启动
 - User flash 区：允许取指、读取、编程、擦除
 - System memory 区：允许取指、读取
 - NVR 区： Engineer Data 区允许读取
 - Trim Data 区允许读取

Option bytes 区允许读取、更新（通过选项字节加载寄存器）

User OTP 区允许读取

User Config 区允许读取、编程、擦除

User backup 区允许读取、编程、擦除

- 调试接口

- User flash 区：允许取指、读取、编程、擦除

- System memory 区：允许取指、读取

- NVR 区：Engineer Data 区允许读取

Trim Data 区允许读取

Option bytes 区允许读取、更新（通过选项字节加载寄存器）

User OTP 区允许读取

User Config 区允许读取、编程、擦除

User backup 区允许读取、编程、擦除

注意： 从 Bootloader 启动时，禁用调试接口。

对于 Flash 存储器，当 Flash 控制器检测到无权限的非法操作时，FLASH_SR 寄存器中的 ACERR 错误标志将立刻置 1，非法操作会被立刻终止，并产生总线访问错误，触发 HardFault 中断。

RDP 1:

- 从 User flash 启动

- User flash 区：允许取指、读取、编程、擦除

- System memory 区：仅允许取指

- NVR 区：Engineer Data 区允许读取

Trim Data 区允许读取

Option bytes 区允许读取、更新（通过选项字节加载寄存器）

User OTP 区允许读取、可编程一次

User Config 区允许读取、编程、擦除

User backup 区允许读取、编程、擦除

- 从 Bootloader 启动

- User flash 区：禁止执行任何操作
- System memory 区：允许读取、取指
- NVR 区：Engineer Data 区允许读取
 - Trim Data 区允许读取
 - Option bytes 区允许读取、更新（通过选项字节加载寄存器）
 - User OTP 区允许读取
 - User Config 区允许读取
 - User backup 区允许读取

- 禁用调试接口

RDP 2:

- 从 User flash 启动
 - User flash 区：允许取指、读取、编程、擦除
 - System memory 区：仅允许取指
 - NVR 区：Engineer Data 区允许读取
 - Trim Data 区允许读取
 - Option bytes 区允许读取、更新（通过选项字节加载寄存器）
 - User OTP 区允许读取、可编程一次
 - User Config 区允许读取、编程、擦除
 - User backup 区允许读取、编程、擦除
- 禁止从 Bootloader 启动
- 禁用调试接口

对于 Flash 存储器，当 Flash 控制器检测到无权限的非法操作时，FLASH_SR 寄存器中的 ACERR 错误标志将立刻置 1，非法操作会被立刻终止，并产生总线访问错误，触发 HardFault 中断。

对 Option bytes 区的更新，必须通过相应的选项字节加载寄存器来完成。当保护等级为 RDP2 时，不允许更新该区域，此时如果 FLASH_CR 寄存器中的 OPTSTRT 位被置 1，则 FLASH_SR 寄存器中的 ACERR 错误标志将置 1，对 Option bytes 区的操作将被终止，并产生总线访问错误，触发 HardFault 中断。

4.5.1.2 RDP 保护等级配置

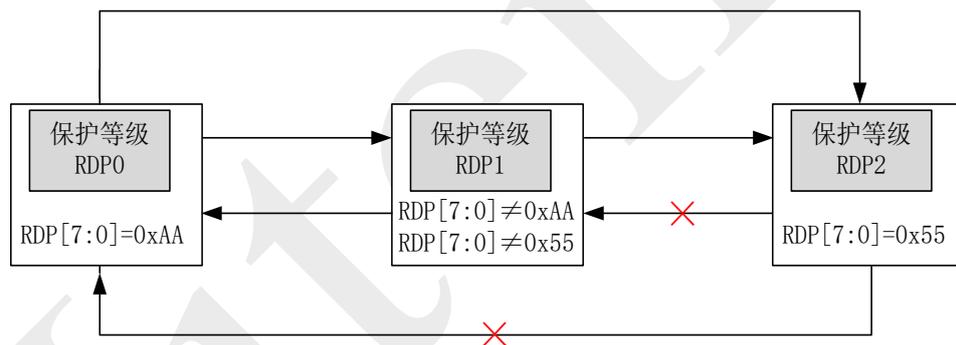
选项字节中的 RDP[7:0] 位域，用于配置 RDP 的保护等级，必须通过 FLASH_OPTR1 寄存器中的 RDP[7:0] 位域进行配置，当选项字节加载生效后，RDP 的保护等级也将生效，选项字节的相关使用方法详见：[Option bytes 区操作说明](#)。

表 4-6 RDP 保护等级配置

RDP 保护等级	RDP[7:0]配置值
RDP 0	0xAA（默认值）
RDP 1	除 0xAA 和 0x55 以外的值
RDP 2	0x55

RDP 的保护等级可以直接从 RDP0 升级到 RDP1，也可以直接从 RDP0 或 RDP1 升级到 RDP2，保护等级的提升不会影响存储器中的内容，RDP 保护等级的变更规则如下图所示：

图 4-1 RDP 保护等级变更



为了防止通过降低 RDP 等级，读取或修改存储器中的内容，当保护等级从 RDP1 降低到 RDP0 时，硬件将自动执行以下操作：

- 擦除整个 User flash 区中的内容
- 修改 Option bytes 区中部分选项字节的配置：
 - 关闭 PCROP 保护区域：
 - PCROP1y_STRT[7:0]被更新为 0xFF (y=A、B)
 - PCROP1y_END[7:0]被更新为 0x00
 - 关闭 WRP 保护区域：
 - WRP1y_STRT[7:0]被更新为 0xFF (y=A、B)
 - WRP1y_END[7:0]被更新为 0x00
 - 关闭用户安全区域：SEC_SIZE[8:0]被更新为 0x000

- BOOT_LOCK 位被更新为 0

4.5.2 代码读出保护

代码读出保护（PCROP）只作用于 User flash 区，仅允许对受保护区域执行取指操作，禁止对受保护区域执行读取、擦除或编程操作。

对受 PCROP 保护的区域执行非法操作，将产生相应的错误标志，但不会造成总线访问错误，也不会触发 HardFault 中断：

- PCROP 读取错误标志 RDERR：
对受 PCROP 保护的区域执行读取操作时，FLASH_SR 寄存器中的 RDERR 标志将置 1，读取操作将被终止。
- 写保护错误标志 WRPERR：
对受 PCROP 保护的区域执行编程或擦除操作时，FLASH_SR 寄存器中的 WRPERR 标志将置 1，编程或擦除操作将被终止。

在 User flash 区，最多能指定两块 PCROP 保护区域，可配置的最小保护区域为 2KB（2 页），保护区域必须按页（1K 字节）进行配置。

注意：两块 PCROP 保护区域 1A 和 1B，可以是独立的，也可以是重叠的。

选项字节中的 PCROP1y_STRT[7:0]和 PCROP1y_END[7:0]位域（y=A、B），用于配置 PCROP 保护区域的范围。

通过 FLASH_PCROP1ySR 寄存器（y=A、B）中的 PCROP1y_STRT[7:0]位域，配置保护区域的起始地址。通过 FLASH_PCROP1yER 寄存器中的 PCROP1y_END[7:0]位域，配置保护区域的结束地址。当选项字节加载生效后，保护区域将同时生效，选项字节的相关使用方法详见：[Option bytes 区操作说明](#)。

表 4-7 PCROP 保护区域配置

寄存器配置 (y=A 或 B)	PCROP 保护区域
PCROP1y_STRT > PCROP1y_END	保护区域关闭（无效）
PCROP1y_STRT = PCROP1y_END	整个 User flash 存储区
PCROP1y_STRT < PCROP1y_END	从 PCROP1y_STRT 到 PCROP1y_END 的区域

当 PCROP1y_STRT < PCROP1y_END 时，受 PCROP 保护的区域范围由 PCROP1y_STRT 和 PCROP1y_END 决定。

PCROP 保护区域的起始地址为：

User flash 区基址（0x0800 0000） + PCROP1y_STRT × 0x400

PCROP 保护区域的结束地址为：

User flash 区基址（0x0800 0000） + (PCROP1y_END+1) × 0x400 - 1

举例，要对 0x0800 8000 到 0x0800 8FFF 之间的存储区域进行保护：

- 1) 地址 0x0800 8000 位于 Page 32，则保护区域的起始页号 PCROP1A_STRT 应设置为 32；
- 2) 地址 0x0800 8FFF 位于 Page 35，则保护区域的结束页号 PCROP1A_END 应设置为 34。

以上 PCROP 保护区域生效后，实际受 PCROP 保护的地址范围为 0x0800 8000 ~ 0x0800 8FFF。当配置了有效的 PCROP 保护区域后，可以通过重新配置选项字节，扩大受保护的区域，扩大后的保护区域将在下次选项字节成功加载后生效。

要关闭已生效的 PCROP 保护区域，只能通过使 RDP 保护等级从 RDP1 降低到 RDP0 来实现，此时 PCROP 的保护会失效，保护区域中的内容将被直接擦除，保护区域也将被关闭。

如果试图通过重新配置选项字节，缩小或关闭受保护的区域，则在选项字节更新时，不会将修改后的内容写入到 Option bytes 区，故不会生效。

4.5.3 Flash 写入保护

写入保护 (WRP) 只作用于 User flash 区，只允许对受保护区域执行取指或读取操作，禁止对受保护区域执行擦除或编程操作。

对受 WRP 保护的区域执行非法操作，将产生相应的错误标志，但不会造成总线访问错误，也不会触发 HardFault 中断：

- 写保护错误标志 WRPEERR：

对受 WRP 保护的区域执行编程或擦除操作时，FLASH_SR 寄存器中的 WRPEERR 标志将置 1，编程或擦除操作将被终止。

在 User flash 区，最多能指定两块 WRP 保护区域，可配置的最小保护区域为 1 页，保护区域必须按页 (1K 字节) 进行配置。

注意：两块 WRP 保护区域 1A 和 1B，可以是独立的，也可以是重叠的。

选项字节中的 WRP1y_STRT[7:0]和 WRP1y_END[7:0]位域 (y=A、B)，用于配置 WRP 保护区域的范围。

通过 FLASH_WRP1ySR 寄存器 (y=A、B) 中的 WRP1y_STRT[7:0]位域，配置保护区域的起始地址。通过 FLASH_WRP1yER 寄存器中的 WRP1y_END[7:0]位域，配置保护区域的结束地址。当选项字节加载生效后，保护区域将同时生效，选项字节的相关使用方法详见：[Option bytes 区操作说明](#)。

表 4-8 WRP 保护区域配置

寄存器配置 (y=A 或 B)	WRP 保护区域
--------------------	----------

WRP1y_STRT > WRP1y_END	保护区域关闭（无效）
WRP1y_STRT = WRP1y_END	保护当前页
WRP1y_STRT < WRP1y_END	从 WRP1y_STRT 到 WRP1y_END 的区域

当 $WRP1y_STRT < WRP1y_END$ 时，受 WRP 保护的区域范围由 WRP1y_STRT 和 WRP1y_END 决定。

WRP 保护区域的起始地址为：

User flash 区基址（0x0800 0000）+ $WRP1y_STRT \times 0x400$

WRP 保护区域的结束地址为：

User flash 区基址（0x0800 0000）+ $(WRP1y_END + 1) \times 0x400 - 1$

举例，要对 0x0800 8000 到 0x08008FFF 之间的存储区域进行保护：

- 1) 地址 0x0800 8000 位于 Page 32，则保护区域的起始页号 WRP1A_STRT 应设置为 32；
- 2) 地址 0x0800 8FFF 位于 Page 35，则保护区域的结束页号 WRP1A_END 应设置为 34。

当以上 WRP 保护区域生效后，实际受 WRP 保护的地址范围为 0x0800 8000 到 0x0800 8FFF。

在配置了有效的 WRP 保护区域后，可以通过重新配置选项字节，扩大、缩小或关闭受保护的区域，修改后的保护区域范围，将在下次选项字节成功加载后生效。

当保护等级从 RDP1 降低到 RDP0 时，WRP 的保护会失效，保护区域中的内容将被直接擦除，保护区域也将被关闭。

4.5.4 Flash 用户安全区域

在复位后，用户安全区域处于未保护状态，可以对该区域执行取指、读取、编程或擦除操作。如果将 FLASH_CR 寄存器中的 SEC_PROT 位置 1，则会使能安全区域的保护机制，此时该区域会变为不可见状态，任何对该区域的取指、读取、编程或擦除操作，都将产生总线访问错误，并触发 HardFalut 中断。

注意： 在使能用户安全区域的保护机制前，必须先将中断向量表移出安全区域。

用户安全区域的保护机制一旦使能，直到下次复位后才会失效，即 SEC_PROT 位置 1 后，不能被软件清 0，直到下次复位才会将其清 0。

选项字节中的 SEC_SIZE[8:0]位域，用于配置用户安全区域的大小，可配置的最小单位为页（1024 字节），通过 FLASH_SECR 寄存器中的 SEC_SIZE[8:0]位域进行配置，当选项字节加载生效后，用户安全区域的大小将同时生效，选项字节的相关使用方法详见：[Option bytes 区操作说明](#)。

用户安全区域的大小生效后：

- 用户安全区域的起始地址固定为：0x0800 0000
- 用户安全区域的结束地址为：0x0800 0000 + 0x400 × SEC_SIZE - 1

如果 SEC_SIZE[8:0]位域为 0，则表示不存在用户安全区域。

只有在保护等级为 RDP0 时，才能对选项字节的 SEC_SIZE[8:0]位域进行更新。当保护等级从 RDP1 降低到 RDP0 时，用户安全区域会失效，安全区域中的内容将被直接擦除，安全区域也将被关闭。

4.5.5 User flash 区强制启动

选项字节中的 BOOT_LOCK 位，用于控制芯片强制从 User flash 区启动，通过 FLASH_SECR 寄存器中的 BOOT_LOCK 位进行配置，当选项字节加载生效后，BOOT_LOCK 将生效，选项字节的相关使用方法详见：[Option bytes 区操作说明](#)。

BOOT_LOCK 位在 RDP0 或 RDP1 下均可置 1，但仅在 RDP0 下能被清 0，或当保护等级从 RDP1 降低到 RDP0 时，该位会被自动清 0。

4.6 Flash 中断

Flash 包含 3 种中断：

- 编程/擦除操作完成中断

配置 FLASH_CR 寄存器中的 EOPIE 位置 1，使能该中断，当编程或擦除操作完成时，FLASH_SR 寄存器中的 EOP 标志将置 1 并触发中断。

注意： EOP 标志仅在 EOPIE 位置 1 后才有效，否则无论编程或擦除操作是否完成，该标志都将保持为 0

- 编程/擦除操作异常中断

配置 FLASH_CR 寄存器中的 ERRIE 位置 1，使能该中断，当发生以下任意一种错误时，FLASH_SR 寄存器中的 OPERR 标志将置 1 并触发中断：

- PGSERR：编程/擦除序列错误；
- SIZERR：编程位宽错误；
- PGAERR：编程地址未对齐错误；
- WRPERR：写保护错误；
- PROGERR：编程错误；

注意： OPERR 标志，仅在 ERRIE 位置 1 后才有效，否则无论编程或擦除操作是否发生异常，该标志都将保持为 0

- PCROP 读取错误中断

配置 FLASH_CR 寄存器中的 RDERRIE 位置 1，使能该中断，当读取受 PCROP 保护的区域时，FLASH_SR 寄存器中的 RDERR 标志将置 1 并触发中断。

表 4-9 Flash 中断

中断源	中断事件	中断标志	中断使能控制位	中断清除方式
Flash	编程/擦除操作完成	EOP	EOPIE	向 EOP 标志位写 1
	编程/擦除操作异常	OPERR	ERRIE	向 OPERR 标志位写 1
	读取 PCROP 保护区	RDERR	RDERRIE	向 RDERR 标志位写 1

4.7 Flash 寄存器

4.7.1 Flash 访问控制寄存器 (FLASH_ACR)

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

位/位域	名称	描述
31:9	保留	必须保持复位值
8	保留	必须保持复位值
7:2	保留	必须保持复位值
1:0	LATENCY[1:0]	Flash 读取访问等待周期 0x: 0 等待周期 10: 1 等待周期 11: 2 等待周期

4.7.2 Flash 控制解锁密钥寄存器 (FLASH_KEYR)

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
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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

位/位域	名称	描述
31:0	KEY[31:0]	用于解锁Flash控制寄存器（FLASH_CR） 解锁FLASH_CR寄存器。 解锁时按顺序写入以下密钥： 密钥1：0x4567 0123 密钥2：0xCDEF 89AB 解锁步骤及注意事项详见： Flash控制寄存器解锁 。

4.7.3 Flash 选项字节解锁密钥寄存器（FLASH_OPTKEYR）

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
OPTKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:0	OPTKEY[31:0]	用于解锁选项字节及其相关控制位 解锁范围包括，所有的选项字节加载寄存器、FLASH_CR寄存器中的OBL_LAUNCH位和OPTSTRT位。 解锁时按顺序写入以下密钥： 密钥1：0x0819 2A3B 密钥2：0x4C5D 6E7F 解锁步骤及注意事项详见： Flash选项字节解锁 。

4.7.4 Flash 状态寄存器（FLASH_SR）

偏移地址：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				ACERR	RDERR	OPERR	EOP	Res.							BSY
				rc_wl	rc_wl	rc_wl	rc_wl								r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res.		
r								rc_wl	rc_wl	rc_wl	rc_wl	rc_wl			

位/位域	名称	描述
------	----	----

31:28	保留	必须保持复位值
27	ACERR	访问权限错误标志 该标志写1清0，写0无效。 当检测到非法访问Flash存储器时，此位将置1，并触发HardFault中断。 0: 正常 1: 发生非法访问错误
26	RDERR	PCROP读取错误标志 该标志写1清0，写0无效。 当读取受PCROP保护的区域时，此位将置1。如果FLASH_CR寄存器中的RDERRIE位已置1，则此位置1时将触发中断。 0: 正常 1: 发生PCROP读取错误
25	OPERR	编程/擦除操作异常中断标志 该标志写1清0，写0无效。 当Flash执行编程/擦除操作时发生异常，PGSERR、SIZERR、PGAERR、WRPERR或PROGERR中的任意一个标志置1，则此位也将置1并触发中断。 0: 正常 1: 发生编程/擦除操作异常 <i>注意：OPERR标志，仅在FLASH_CR寄存器中的ERRIE位置1后才有效，否则无论编程或擦除操作是否发生异常，该标志都将保持为0。</i>
24	EOP	编程/擦除操作完成中断标志 该标志写1清0，写0无效。 当Flash编程/擦除操作完成时，此位将置1并触发中断。 0: 编程/擦除操作未完成 1: 编程/擦除操作已完成 <i>注意：EOP标志，仅在FLASH_CR寄存器中的EOPIE位置1后才有效，否则无论编程或擦除操作是否完成，该标志都将保持为0。</i>
23:17	保留	必须保持复位值
16	BSY	Flash编程/擦除操作状态标志 当Flash执行编程或擦除操作时，此位置1。在操作完成或发生错误时，此位将自动清0。 0: Flash处于空闲状态 1: Flash处于忙状态，正在执行编程/擦除操作 <i>注意：BSY位为1时，不能修改FLASH_CR寄存器。</i>

15:8	保留	必须保持复位值
7	PGSERR	<p>Flash编程/擦除序列错误标志 该标志写1清0，写0无效。</p> <p>0: 正确 1: 发生编程/擦除序列错误</p> <p>Flash执行编程/擦除操作时，未遵循指定的步骤顺序执行，当出现以下情况时，此位将置1： 编程操作时： - PG位置1前，向Flash地址写入数据。 - PG位为1时，ERM [1:0]位域不为00。 擦除操作时： - STRT位置1时，PG位未清0。 - STRT位置1时，ERM [1:0]位域仍为0。</p> <p>存在错误标志时： - 当有错误标志（PROGERR、SIZERR、PGAERR、WRPERR）未被清除时，配置STRT位置1或向Flash地址写入数据。</p>
6	SIZERR	<p>编程位宽错误标志 该标志写1清0，写0无效。 当Flash的编程位宽不是32bits或16bits时，此位将置1</p> <p>0: 正确 1: 发生编程位宽错误</p>
5	PGAERR	<p>编程地址未对齐错误标志 该标志写1清0，写0无效。 当编程地址未按字或半字对齐时（目标地址的最低位不为b0），此位将置1</p> <p>0: 正常 1: 发生编程地址未对齐错误</p>
4	WRPERR	<p>写保护错误标志 该标志写1清0，写0无效。 当执行编程/擦除操作的区域，属于受保护的区域（受WRP或PCROP保护），此位将置1。</p> <p>0: 正常 1: 发生写保护错误</p>
3	PROGERR	<p>编程错误标志 该标志写1清0，写0无效。 当目标地址已执行过编程操作（非0xFFFF FFFF值），再次向该地址写入数据时，此位将置1。</p> <p>0: 正常 1: 发生编程错误</p>

2:0 保留 必须保持复位值

4.7.5 Flash 控制寄存器 (FLASH_CR)

偏移地址：0x14

复位值：0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPTLOCK	Res.	SEC_PROT	OBL_LAUNCH	RDERRIE	ERRIE	EOPIE	Res.							
rs	rs		rw	rc_wl	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PNB[7:0]								Res.		OPTSTRT	STRT	ERM	PG		
rw	rw	rw	rw	rw	rw	rw	rw			rs	rs	rw	rw	rw	

位/位域	名称	描述
31	LOCK	FLASH_CR寄存器锁定控制 配置此位置1后，FLASH_CR寄存器将被写保护锁定。 当检测到正确的解锁序列时，此位自动清0。 解锁步骤及注意事项详见： Flash控制寄存器解锁 。
30	OPTLOCK	选项字节及相关控制位锁定控制 配置此位置1后，所有的选项字节加载寄存器、FLASH_CR寄存器中的OBL_LAUNCH位和OPTSTRT位将被写保护锁定。 当检测到正确的解锁序列时，此位自动清0。 解锁步骤及注意事项详见： Flash选项字节解锁 。
29	保留	必须保持复位值
28	SEC_PROT	用户安全区域保护机制使能控制 此位置1后，不能被软件清0，直到下次复位才会将其清0： 0：用户安全区域处于未保护状态 1：使能用户安全区域保护机制
27	OBL_LAUNCH	选项字节加载控制 0：选项字节加载完成 1：加载选项字节，并触发系统复位 当FLASH_CR寄存器中的OPTLOCK位置1时，此位处于写保护锁定状态，不能被修改。 解锁步骤及注意事项详见： Flash选项字节解锁
26	RDERRIE	PCROP读取错误中断使能控制 此位置1后，当FLASH_SR寄存器中的RDERR标志置1时，将触发中断。 0：禁止PCROP读取错误中断

		1: 使能PCROP读取错误中断
25	ERRIE	编程/擦除操作异常中断使能控制 此位置1后, FLACH_SR寄存器中的OPERR标志将生效, 当OPERR标志置1时, 将触发中断。 0: 禁止编程/擦除操作异常中断 1: 使能编程/擦除操作异常中断
24	EOPIE	编程/擦除操作完成中断使能控制 此位置1后, FLACH_SR寄存器中的EOP标志将生效, 当EOP标志置1时, 将触发中断。 0: 禁止编程/擦除操作完成中断 1: 使能编程/擦除操作完成中断
23:16	保留	必须保持复位值
15:8	PNB[7:0]	Flash擦除页选择 根据FLASH_CR寄存器中ERM [1:0]位域的配置, 选择要擦除的页。 - 页擦除模式, PNB[7:0]用于指定要擦除的Flash页号(取决Boot_size和Boot_cfg): 0x00: Page 0 0x01: Page 1 ⋮ 0xEB: Page 235(缺省情况)
7:5	保留	必须保持复位值
4	OPTSTRT	启动选项字节更新 此位置1时, 将启动选项字节的更新。 更新完成后, 此位将自动清0。 当FLASH_CR寄存器中的OPTLOCK位置1时, 此位处于写保护锁定状态, 不能被修改。 解锁步骤及注意事项详见: Flash选项字节解锁 。
3	STRT	启动Flash擦除 此位置1时, 将启动Flash擦除。 擦除完成后, 此位自动清0。
2:1	ERM[1:0]	Flash擦除模式选择 00: 退出擦除模式 01: 页擦除模式 1x: 预留
0	PG	Flash编程模式控制

- 0: 退出Flash编程模式
- 1: 进入Flash编程模式

4.7.6 选项字节寄存器 1 (FLASH_OPTR1)

偏移地址: 0x20

复位值: bit: 11111111 11111111 1111XXXX XXXXXXXX

芯片上电值 (从 POR 复位到 NVR 加载完成前): 0xFFFF E0AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		BOR_HYS[1:0]		BOR_LEV[2:0]			BOR_EN	RDP[7:0]							
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13:12	BOR_HYS[1:0]	BOR迟滞选择 00: 低档 01: 中低档 10: 中档 11: 高档
11:9	BOR_LEV[2:0]	BOR阈值等级 000: level0上升/下降(default) 001: level2上升/下降 010: level4上升/下降 011: level6上升/下降 100: level8上升/下降 101: level10上升/下降 110: level12上升/下降 111: level14上升/下降
8	BOR_EN	BOR使能控制 0: 禁止BOR(default) 1: 使能BOR
7:0	RDP[7:0]	RDP保护等级选择 0xAA: 等级0 (RDP0) (default) 0x55: 等级2 (RDP2) 其他值: 等级1 (RDP1)

4.7.7 选项字节寄存器 2 (FLASH_OPTR2)

偏移地址：0x24

复位值：bit:11111111 11111111 XXXXXXXX1 11XX1111

芯片上电值（从POR复位到NVR加载完成前）：0xFFFF 7DDF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRST_MODE	nBOOT_SEL	nBOOT0	BOOTSIZ[2:0]			BOOTCFG	Res.			IWDG_STOP	IWDG_SW	Res.			
rw	rw	rw	rw	rw	rw	rw				rw	rw				

位/位域	名称	描述
31:16	保留	必须保持复位值
15	NRST_MODE	外部复位管脚使能 0: PD13固定做NRST管脚 1: PD13做GPIO管脚
14	nBOOT_SEL	启动配置选择 0: 启动条件使用BOOT0引脚 1: 启动条件使用nBOOT0 相关内容详见： 启动配置 。
13	nBOOT0	启动条件，相关内容详见： 启动配置 。
12:10	BOOTSIZ	System memory区大小 000: 2KB 001: 4KB 010: 6KB 011: 8KB 100: 10KB 101: 12KB 110: 14KB 111: 16KB
9	BOOTCFG	System memory使用配置 0: 厂商bootloader 1: system memory被用作用户程序或数据区即：system memory大小为0KB
8:6	保留	必须保持复位值
5	IWDG_STOP	独立看门狗在Stop模式下计数器停止控制：

- 0: 独立看门狗在Stop下计数器停止计数（缺省）
- 1: 独立看门狗在Stop下计数器正常运行

4	IWDG_SW	独立看门狗的使能方式： 0: 由硬件控制使能独立看门狗（芯片上电自动使能） 1: 由软件控制使能独立看门狗（缺省）
3:0	保留	必须保持复位值

4.7.8 代码读出保护区 1A 起始地址寄存器（FLASH_PCROP1ASR）

偏移地址：0x28

复位值：bit:11111111 11111111 11111111 XXXXXXXX

芯片上电值（从 POR 复位到 NVR 加载完成前）：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PCROP1A_STRT[7:0]							
								rw rw rw rw rw rw rw rw							

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	PCROP1A_STRT[7:0]	PCROP1A保护区域的起始地址页号 保护区域基于User flash的起始地址（0x0800 0000），如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到PCROP保护。 PCROP1A保护区域的起始地址： User flash区基址 + PCROP1A_STRT[7:0] × 0x400

4.7.9 代码读出保护区 1A 结束地址寄存器（FLASH_PCROP1AER）

偏移地址：0x2C

复位值：bit:11111111 11111111 11111111 XXXXXXXX

芯片上电值（从 POR 复位到 NVR 加载完成前）：0xFFFF FF00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PCROP1A_END[7:0]							
								rw rw rw rw rw rw rw rw							

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	PCROP1A_END[7:0]	PCROP1A保护区域的结束地址页号 保护区域基于User flash的起始地址（0x0800 0000），如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到PCROP保护。 PCROP1A保护区域的结束地址： User flash区基址 + (PCROP1A_END[7:0]+1) × 0x400 - 1

4.7.10 代码读出保护区 1B 起始地址寄存器（FLASH_PCROP1BSR）

偏移地址：0x30

复位值：bit:11111111 11111111 11111111 XXXXXXXX

芯片上电值（从 POR 复位到 NVR 加载完成前）：0xFFFFFFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PCROP1B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	PCROP1B_STRT[7:0]	PCROP1B保护区域的起始地址页号 保护区域基于User flash的起始地址（0x0800 0000），如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到PCROP保护。 PCROP1B保护区域的起始地址： User flash区基址 + PCROP1B_STRT[7:0] × 0x400

4.7.11 代码读出保护区 1B 结束地址寄存器（FLASH_PCROP1BER）

偏移地址：0x34

复位值：bit:11111111 11111111 1111 XXXXXXXX

芯片上电值（从 POR 复位到 NVR 加载完成前）：0xFFFF FF00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PCROP1B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

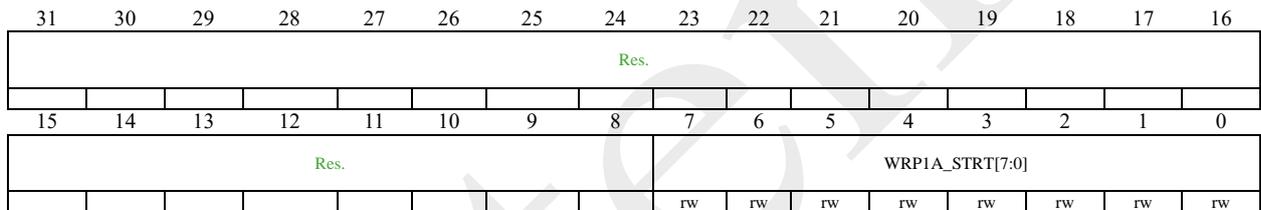
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	PCROP1B_END[7:0]	PCROP1B保护区域的结束地址页号 保护区域基于User flash的起始地址 (0x0800 0000)，如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到PCROP保护。 PCROP1B保护区域的结束地址： User flash区基址 + (PCROP1B_END[7:0]+1) × 0x400 - 1

4.7.12 写保护区 1A 起始地址寄存器 (FLASH_WRP1ASR)

偏移地址: 0x38

复位值: bit:11111111 11111111 11111111 XXXXXXXX

芯片上电值 (从 POR 复位到 NVR 加载完成前): 0xFFFF FFFF



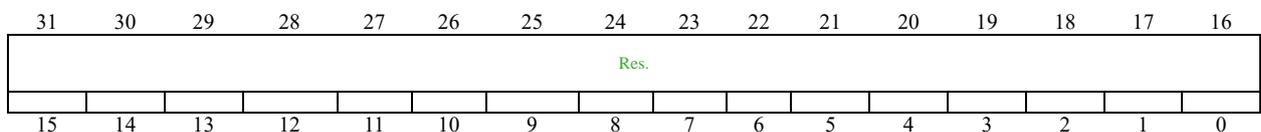
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WRP1A_STRT[7:0]	WRP1A保护区域的起始地址页号 保护区域基于User flash的起始地址 (0x0800 0000)，如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到WRP保护。 WRP1A保护区域的起始地址： User flash区基址 + WRP1A_STRT[7:0] × 0x400

4.7.13 写保护区 1A 结束地址寄存器 (FLASH_WRP1AER)

偏移地址: 0x3C

复位值: bit: 11111111 11111111 11111111 XXXXXXXX

芯片上电值 (从 POR 复位到 NVR 加载完成前): 0xFFFF FF00



Res.							WRP1A_END[7:0]							
							rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WRP1A_END[7:0]	<p>WRP1A保护区域的结束地址页号</p> <p>保护区基于User flash的起始地址 (0x0800 0000)，如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到WRP保护。</p> <p>WRP1A保护区域的结束地址： User flash区基址 + (WRP1A_END[7:0]+1) × 0x400 - 1</p>

4.7.14 写保护区 1B 起始地址寄存器 (FLASH_WRP1BSR)

偏移地址: 0x40

复位值: bit: 11111111 11111111 11111111 XXXXXXXX

芯片上电值 (从 POR 复位到 NVR 加载完成前): 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							WRP1B_STRT[7:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WRP1B_STRT[7:0]	<p>WRP1B保护区域的起始地址页号</p> <p>保护区基于User flash的起始地址 (0x0800 0000)，如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到WRP保护。</p> <p>WRP1B保护区域的起始地址： User flash区基址 + WRP1B_STRT[7:0] × 0x400</p>

4.7.15 写保护区 1B 结束地址寄存器 (FLASH_WRP1BER)

偏移地址: 0x44

复位值: bit: 11111111 11111111 11111111 XXXXXXXX

芯片上电值 (从 POR 复位到 NVR 加载完成前): 0xFFFF FF00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								WRP1B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WRP1B_END[7:0]	WRP1B保护区域的结束地址页号 保护区域基于User flash的起始地址（0x0800 0000），如果User flash被重映射到0x0000 0000，则重映射后的地址和原地址均会受到WRP保护。 WRP1B保护区域的结束地址： User flash区基址 + (WRP1B_END[7:0]+1) × 0x400 - 1

4.7.16 用户安全配置寄存器（FLASH_SECUR）

偏移地址：0x48

复位值：bit:00000000 00000000 X000000X XXXXXXXX

芯片上电值（从POR复位到NVR加载完成前）：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_LOCK	Res.						SEC_SIZE[8:0]								
rw							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	BOOT_LOCK	用于强制从User flash启动。 0: 根据nBOOT_SEL、nBOOT0和BOOT0引脚的配置启动 1: 强制从User flash区启动
14:9	保留	必须保持复位值
8:0	SEC_SIZE[8:0]	用户安全区域占用User flash页的数量 用户安全区域的大小为：SEC_SIZE[8:0] × 0x400。 注意：当配置SEC_SIZE[8:0]位域大于256时，用户安全区域的大小为整个User flash区（256KB）。

4.7.17 芯片封装参数寄存器（FLASH_PACKAGER）

偏移地址：0x50

复位值：上电复位值 0xFFFF FFF0 ，从 Flash 中加载后 0xFFFF FFF8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.													PACKAGE[3:0]			
													r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值
3:0	PACKAGE[3:0]	封装类型（具体封装形式见datasheet） 0000：48 pin封装 0001：44 pin封装 0010：32 pin封装 0011：28 pin封装 0100：24 pin封装 1000：64 pin封装

4.7.18 芯片产品型号参数寄存器（FLASH_DEVTYPR）

偏移地址：0x54

复位值：0xFFFF3602 从 Flash 中加载

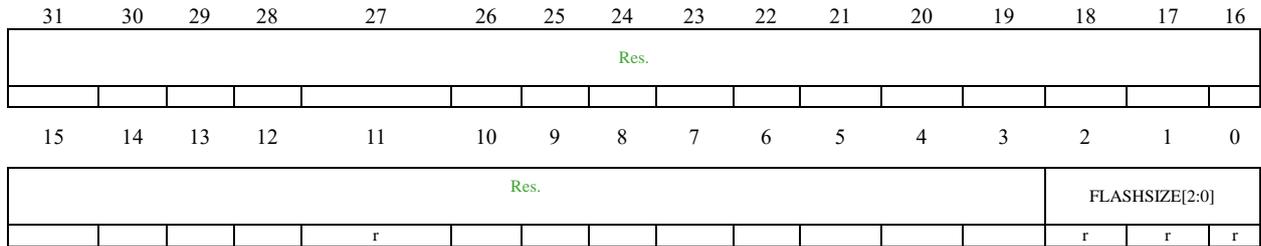
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVTYP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DEVTYP[15:0]	产品类型定义 默认值：3602

Flash 空间参数寄存器（FLASH_UFSIZER）

偏移地址：0x58

复位值：0xFFFF FFFC 从 Flash 中加载

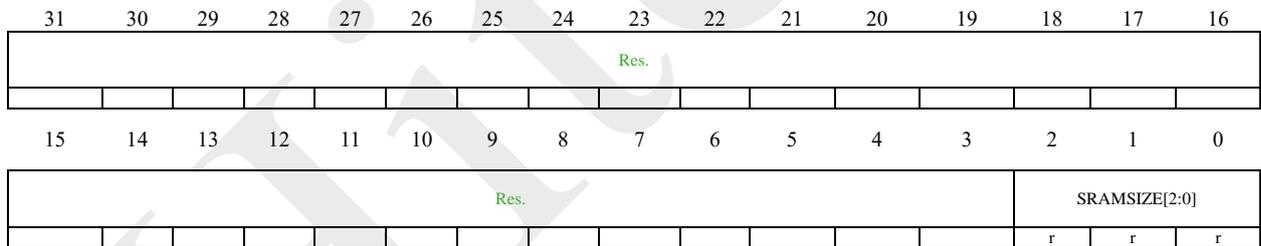


位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	FLASHSIZE[2:0]	User flash区存储空间大小 000: 16KB 001: 32KB 010: 64KB 011: 128KB 1xx: 250~236KB (缺省)

4.7.19 SRAM 空间参数寄存器 (FLASH_SRAMSIZER)

偏移地址: 0x5C

复位值: 0xFFFF FFFC 从 Flash 中加载

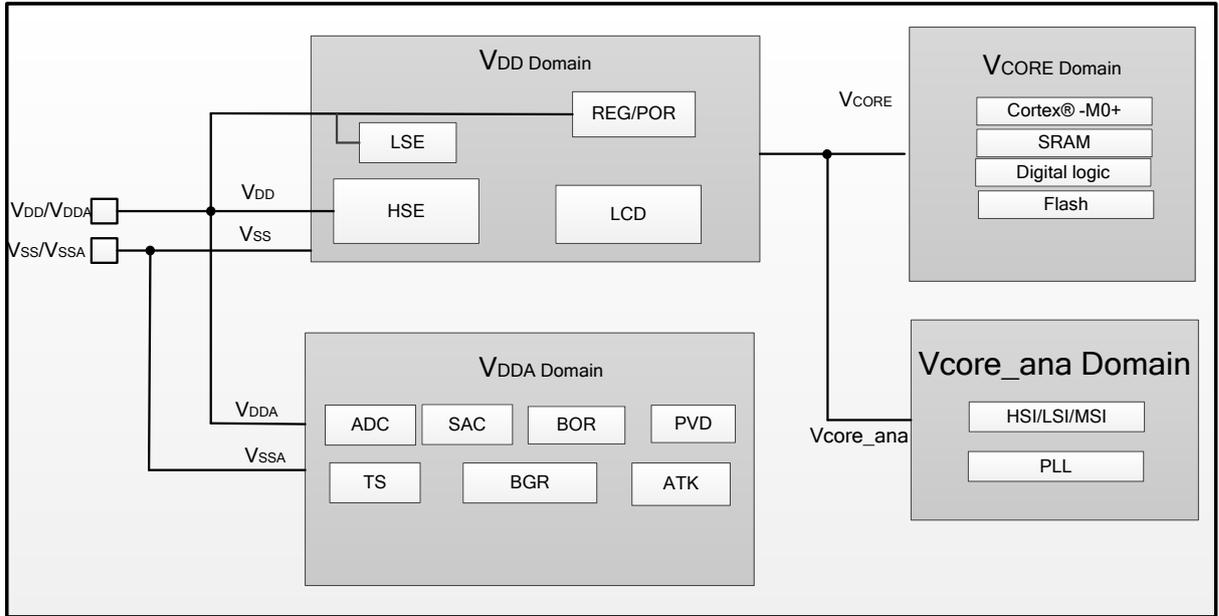


位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	SRAMSIZE[2:0]	SRAM存储空间大小 000: 4KB 001: 8KB 010: 16KB 011: 24KB 1xx: 32KB (缺省)

5 电源管理单元 (PWR)

5.1 电源

图 5-1 电源结构框图



芯片电源分为单电源域，每个电源域电压以及包含外设如下：

- V_{DD} 引脚输入电压为 2.0V~5.5V，该电源域包含 LCD、HSE、LSE、调压器 (REG/POR)；调压器给 V_{CORE} 域和 V_{core_ana} 域供电。
- V_{DDA} 输入电压为 2.0V~5.5V，为模拟外设供电，包括 ADC、SAC、BGR、ATK、TS (温度传感器)、BOR 和 PVD。
- V_{CORE} 域由内部调压器供电，电压为 1.5V；包含 Cortex® -M0+、SRAM、Flash、数字外设和 MSI、HSI、LSI、PLL。

5.1.1 内部调压器

内部调压器提供电源给 V_{CORE} 域和 Flash，与低功耗模式结合，进而降低系统功耗。

调压器在 Run 模式下，根据不同时钟源的选择，有三种 Run 模式，详见 [Run](#)。

5.2 电源电压监测

5.2.1 上电复位和掉电复位

芯片内置上电复位（POR）和掉电复位模块（PDR），该模块可工作在所有功耗模式。

当 V_{DD} 电压达到 2.2V 时，等待 $t_{RSTTEMPO}$ (典型 1ms) 后释放复位，当 V_{DD} 电压低于 2.1V 时，触发复位。

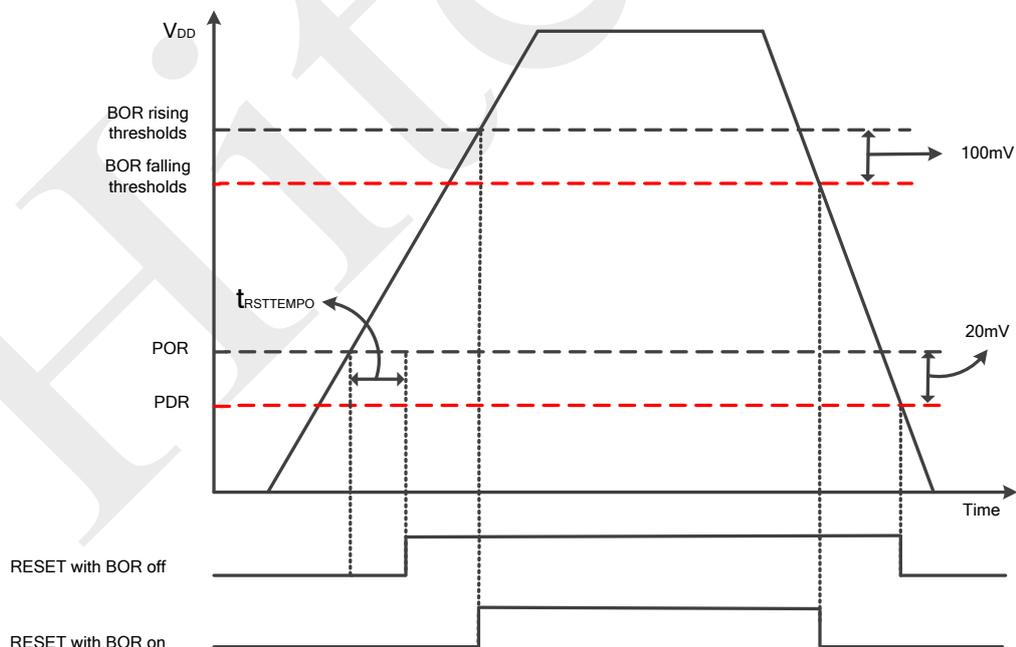
5.2.2 欠压复位

可通过配置选项字节寄存器(**FLASH_OPTR1**)配置 BOR 使能和 BOR 的阈值，BOR 阈值有 8 档可选。

BOR 每个阈值档位同时设定了上升阈值和下降阈值：

- 使能 BOR 时，当 V_{DD} 电压高于配置 BOR 上升阈值时，复位释放；当 V_{DD} 电压低于 BOR 下降阈值时，触发复位；
- 禁止 BOR 时， V_{DD} 高于 2.2V 时释放复位，参见[上电复位和掉电复位 \(POR/PDR\)](#)；

图 5-2 POR/PDR 和 BOR 阈值



5.2.3 可编程电压检测

PVD 可监测 V_{DD} 和 IO 引脚上的电压，当监测电压与 PVD 阈值比较结果满足设定条件时，会触发 PVD 报警。

PVD 监测信号源有四种： V_{DD} 、PC4、PD4、PD6，通过[电源管理控制寄存器 2](#)

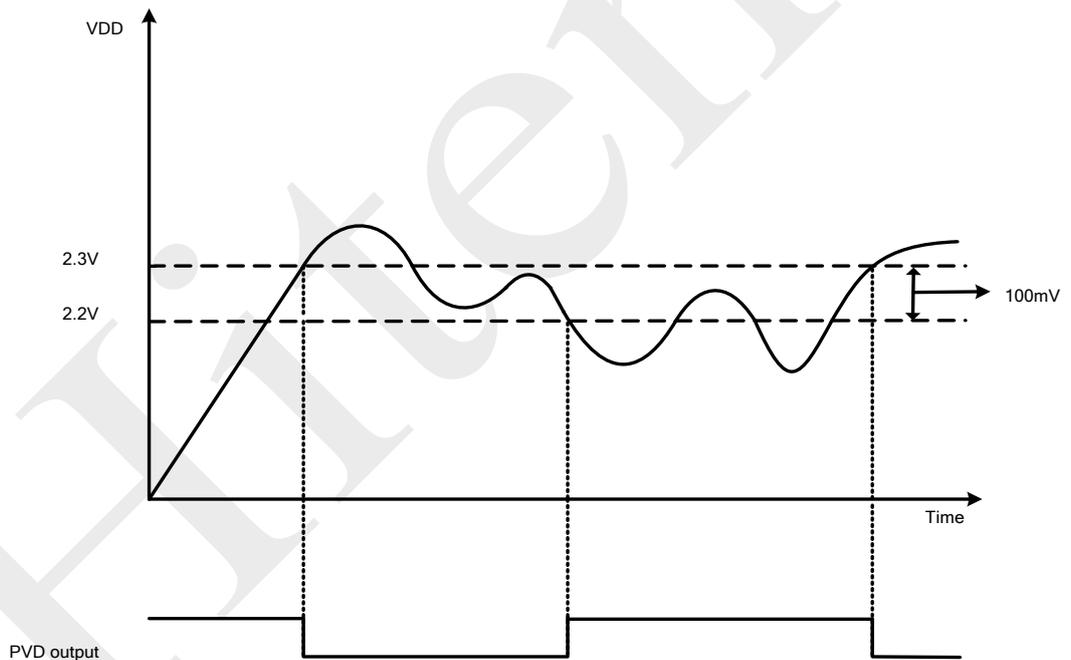
(*PWR_CR2*) 中 PVDS[1:0]选择；当选择监测 IO 引脚时，引脚输入电压不能大于 V_{DD} 。

触发 PVD 报警时，如果 PVD 报警中断使能，则会生成 PVD 报警中断；触发条件有以下几种：

- 监测电压高于阈值时触发报警；
- 监测电压低于阈值时触发报警；
- 当监测电压从高于阈值到低于阈值变化时触发报警；
- 当监测电压从低于阈值到高于阈值变化时触发报警；

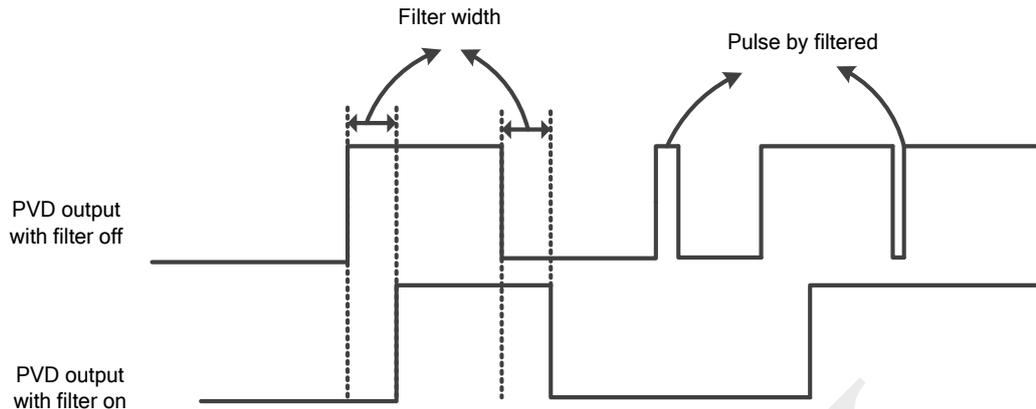
PVD 阈值有多档可选，由 *电源管理控制寄存器 2 (PWR_CR2)* 中 PVDT[4:0]决定；PVD 具有迟滞功能，当 PVD 阈值等级选定后，阈值下限电压会比阈值上限电压低 100mV，参见下图 (PVDT[4:0]为 00010)。

图 5-3 PVD 阈值



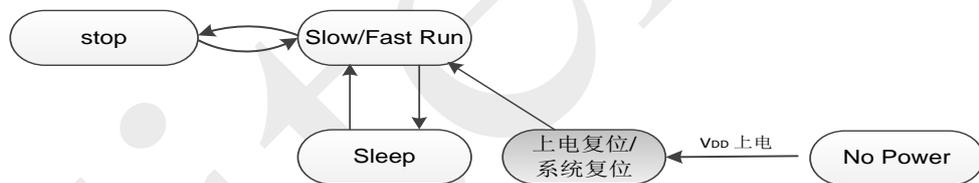
PVD 支持数字滤波，滤波时间由 *电源管理控制寄存器 2 (PWR_CR2)* 中 FTIME[2:0]决定，通过将 FEN 置 1 (*电源管理控制寄存器 2 (PWR_CR2)*)，来使能数字滤波；当滤波使能时，PVDS 为滤波后 PVD 状态。

由于滤波时间基于 PCLK 时钟，在 Stop 模式下工作时，由于 PCLK 已经停止，数字滤波功能不再起作用。

图 5-4 PVD 滤波


5.3 低功耗模式

复位后，MCU 处于 Run 模式，系统时钟源为 HSI SYS（18MHz 的 2 分频）。MCU 提供多种低功耗模式，以降低系统的功耗；另外，还可以通过降低系统时钟频率、关闭未使用外设的时钟来降低 Run 模式下的系统功耗。

图 5-5 低功耗模式转换框图


MCU 提供 2 种低功耗模式：

- **Sleep**：在 Run 基础上，仅关闭 CPU 时钟，所有外设（包括 Cortex®-M0+ 内核的外设，例如：NVIC、SysTick 等）均保持运行状态，当有中断或者事件发生时，可以唤醒 CPU；详细内容参见 [Sleep](#)。
- **Stop**：V_{CORE} 电源域（包括 CPU、数字外设）的时钟关闭，PLL、HSI、MSI、HSE 等处于 PowerDown 状态，LSI 和 LSE 根据需要可保持运行；SRAM 和寄存器的数据保持。

一些具有唤醒能力的外设在此模式下可以通过请求开启 HSI，使用 HSI 时钟来检测唤醒条件，当满足唤醒条件时，唤醒系统。

退出 Stop 后，系统时钟源为 HSI SYS（HSI 的 2 分频）。

详细内容参见 [Stop](#)。

表 5-1 低功耗模式概述

功耗模式	模式描述	进入方式	唤醒源	唤醒后时钟
Sleep	仅关闭 CPU 的时钟。	SLEEPDEEP 清 0 WFI 或 从中断服务函数中返回	所有 NVIC 中断	与进入前的时钟保持一致
		SLEEPDEEP 清 0 WFE	唤醒事件	
Stop	VCORE 电源域（包括 CPU、数字外设）的时钟关闭； HSI、MSI、HSE、PLL 等处 PowerDown 状态； LSI 和 LSE 可正常工作。	SLEEPDEEP 置 1 LP_MODE 为 0 WFI 或 从中断服务函数中返回	与 EXTI 信号线连接的外设生成的中断	HSISYS（HSI 的 2 分频）
		SLEEPDEEP 置 1 LP_MODE 为 0 WFE	与 EXTI 信号线连接的外设生成的唤醒事件	

表 5-2 低功耗模式下可用外设

外设	Run	Sleep	Stop	
			功能	唤醒源
CPU	√	×	×	×
Flash	√	√	√	×
SRAM	√	√	√	×
BOR	○	○	○	○
POR/PDR	√	√	√	√
PVD	○	○	○	○
HSI	○	○	○ ⁽²⁾	×
MSI	○	○	×	×
LSI	○	○	○	×
PLL	○	○	×	×
HSE	○	○	×	×
LSE	○	○	○	×
LSE_CSS	○	○	×	○
PLL_CSS	○	○	×	×
DMA	○	○	×	×
USART1	○	○	○ ⁽³⁾	○ ⁽³⁾
USART2/3	○	○	×	×
LPUART	○	○	○ ⁽³⁾	○ ⁽³⁾
I2C1/2	○	○	○ ⁽⁴⁾	○ ⁽⁴⁾
SPI1/2	○	○	×	×
TIM1/2/3/4/5/6/7	○	○	×	×

LPTIM	○	○	○	○
IWDG	○	○	○	○
WWDG	○	○	×	×
SysTick	○	○	×	×
ADC	○	○	×	×
SAC	○	○	○	○
TS	○	○	×	×
GPIO	○	○	○	○
LCD	○	○	○	×
ATK	○	○	×	×
CRC	○	○	×	×
DIV	○	○	×	×
PLA	○	○	×	×
RTC	○	○	○	○

1. √-表示常开，×-表示不可用，○-可配置；
2. 在 Stop 下，可通过置位 [时钟控制/控制寄存器 \(RCC_CR\)](#) 中 HSIKERN 保持 HSI 常开；
3. 在 Stop 下，USART1 和 LPUART 可使能唤醒功能，当检测到匹配的唤醒事件（起始位、地址匹配、接收非空）时，生成唤醒事件；
4. 在 Stop 下，I2C 可使能唤醒功能，当检测到地址匹配后，生成唤醒事件；

5.3.1 Run

芯片复位后进入 Run，默认系统时钟源为 HSI，时钟频率为 HSI 的 2 分频，最高支持 48MHz；该模式下，所有外设均可使用。

Run 模式根据选择的时钟源不同，有三种功耗模式，具体如下：

- 慢速 Run 模式：当选择 LSI 或 LSE（32.768K）时，MCU 处于慢速 Run 模式，该 Run 模式运行功耗最小；当需要长期慢速运行时，可使用该模式；
- 低速 Run 模式：当选择 MSI 时，MCU 处于低速 Run 模式，该模式支持最高系统时钟为 3.686MHz；
- 高速 Run 模式：当使能 HSE、HSI 或 PLL 时，MCU 处于高速 Run 模式，该模式最高系统时钟可达 48MHz。

另外，Run 模式下还可以通过如下措施降低功耗：

- 通过配置相关的预分频寄存器，降低 SYSCLK、HCLK、PCLK1 和 PCLK2 的时钟频率；
- 禁止不使用的的外设时钟；
- 为了进一步降低功耗，可以选择进入 Sleep 或 Stop；

5.3.2 低功耗模式进入和退出

结合 Cortex® -M0+对低功耗的支持，低功耗模式有三种进入方式：WFI (wait for interrupt)、WFE (wait for event)和 Return from ISR (退出中断服务函数)；三种方式的具体如下：

- **WFI (wait for interrupt):** 等待中断，NVIC 使能的中断信号可唤醒该模式（配置并使能外设的中断信号，并且使能对应的 NVIC 中断，可以进入中断服务函数；包括 EXTI 和各个外设的中断信号）。
- **WFE (wait for event):** 等待事件，唤醒事件分以下三种情况：
 - **NVIC 使能的中断信号：**配置并使能外设的中断信号，并且使能对应的 NVIC 中断，可以进入中断服务函数；包括 EXTI 和各自外设的中断信号。
 - **NVIC 中断挂起信号：**当 SEVONPEND 为 1（SCR 寄存器）时，配置并使能外设的中断信号，但 NVIC 的中断未使能；此时中断挂起信号也可以唤醒通过 WFE 进入的低功耗模式，不会进入中断服务函数。
 - **EXTI 生成的事件：**配置 EXTI 相关信号线生成事件，此事件可以唤醒通过 WFE 进入的低功耗模式，与中断信号无关。
- **Return from ISR (退出中断服务函数)：**退出最低优先级中断服务函数后，立即进入低功耗模式；该方式通过置位 **系统控制寄存器 (SCR)** 中 SLEEPONEXIT 实现；只有 NVIC 使能的中断信号可唤醒该模式。

低功耗模式的进入和退出方式如下表：

表 5-3 低功耗模式进入和退出方式

进入方式	SLEEPONEXIT ⁽¹⁾	退出低功耗模式的唤醒信号	
		Sleep	Stop
WFI	0	NVIC 使能的中断信号	唤醒引脚、IWDG、BOR、POR
WFE	0	唤醒事件	
Return from ISR	1	NVIC 使能的中断信号	

1. Cortex® -M0+中 SCR 寄存器的 bit1，参见 **系统控制寄存器 (SCR)**；

5.3.3 Sleep

进入 Sleep 模式仅关闭 CPU 的时钟，所有外设均可正常工作。

Sleep 模式下 IO 状态

在 Sleep 模式下，所有 IO 状态与 Run 模式保持一致。

进入 Sleep 模式

清零内核 [系统控制寄存器 \(SCR\)](#) 中 SLEEPDEEP 后，通过如下几种方式进入 Sleep 模式：

- 当内核 SCR 寄存器中 SLEEPONEXIT 为 0 时，执行 WFI 或 WFE，MCU 立即进入 Sleep 模式；
- 当内核 SCR 寄存器中 SLEEPONEXIT 为 1 时，退出最低优先级中断服务函数时，MCU 立即进入 Sleep 模式；

退出 Sleep 模式

根据进入 Sleep 模式的方式不同，唤醒源有一定差异，参见 [低功耗模式进入和退出](#)；退出 Sleep 模式后，MCU 进入 Run 模式。

表 5-4 Sleep 说明

行为类型	描述说明
进入 Sleep	1. SLEEPDEEP 清 0； 2. 执行 WFI；
	1. SLEEPDEEP 清 0； 2. 无唤醒事件处于挂起状态（无待处理的唤醒事件） 3. 执行 WFE；
	1. SLEEPDEEP 清 0； 2. SLEEPONEXIT 置 1； 3. 退出最低优先级中断服务函数；
退出 Sleep	通过 WFI 或退出中断服务函数方式进入低功耗模式：所有的中断服务请求（中断使能）
	通过 WFE（SEVONPEND 为 0）方式进入低功耗模式： - 所有的中断服务请求（NVIC 中断使能） - EXTI 生成的事件
	通过 WFE（SEVONPEND 为 1）方式进入低功耗模式： - 所有的中断服务请求，不论是否 NVIC 中断使能 - EXTI 生成的事件
退出延迟	无

5.3.4 Stop

Stop 基于 Cortex®-M0+ 的深度休眠功能实现，V_{CORE} 电源域（包括 CPU、数字外设）的时钟关闭，PLL、HSI、MSI、HSE 等处于 PowerDown 状态；SRAM 和寄存器的数据保持。

LSI 和 LSE 根据配置可保持正常运行；一些具有唤醒功能的外设（USART1、LPUART、I2C）在 Stop 下，可通过开启 HSI，使用 HSI 时钟来检测唤醒事件，当唤醒事件匹配时唤醒系统；如果不匹配，则关闭 HSI。

Stop 下可工作的外设参见 [表：低功耗模式下可用外设](#)。

ADC 和 TS（温度传感器）在 Stop 下虽然无法工作，但仍会产生功耗；应该进入 Stop 前，关闭这些外设。

Stop 模式下 IO 状态

在 Stop 模式下，除下面 IO 外，均与进入 Stop 模式前的状态保持一致：

- NRST 引脚（请参见数据手册中的复用功能映射）
- LSE 相关引脚（请参见数据手册中的复用功能映射）
- LSCO 相关引脚（请参见数据手册中的复用功能映射）

进入 Stop 模式

在 Run 模式下，置位 [系统控制寄存器 \(SCR\)](#) 中 SLEEPDEEP，配置 LP_MODE 为 0 时，选择低功耗模式为 Stop；通过如下几种方式进入 Stop：

- 当内核 SCR 寄存器中 SLEEPONEXIT 为 0 时，执行 WFI 或 WFE，MCU 立即进入 Stop；
- 当内核 SCR 寄存器中 SLEEPONEXIT 为 1 时，退出最低优先级中断服务函数时，MCU 立即进入 Stop；

退出 Stop 模式

根据进入 Stop 模式的方式不同，唤醒源有一定差异，参见 [低功耗模式进入和退出](#)。

退出 Stop 模式后，MCU 进入 Run 模式，此时系统时钟源为 HSI SYS，频率为 HSI 的 2 分频。

表 5-5 Stop 说明

行为类型	描述说明
进入 Stop	1. SLEEPDEEP 置 1； 2. LP_MODE 配置为 0； 3. 执行 WFI；
	1. SLEEPDEEP 置 1； 2. LP_MODE 配置为 0； 3. 无唤醒事件处于挂起状态（无待处理的唤醒事件）； 4. 执行 WFE；
	1. SLEEPDEEP 置 1； 2. SLEEPONEXIT 置 1； 3. LP_MODE 配置为 0； 4. 退出最低优先级中断服务函数；
退出 Stop	通过 WFI 或退出中断服务函数方式进入低功耗模式：与 EXTI 信号线连接的外设的中断服务请求（中断使能），参见 EXTI 信号线连接表
	通过 WFE（SEVONPEND 为 0）方式进入低功耗模式：

	<ul style="list-style-type: none"> - 与 EXTI 信号线连接的外设中断服务请求 (NVIC 中断使能), 参见 EXTI 信号线连接表 - EXTI 生成的事件
	通过 WFE (SEVONPEND 为 1) 方式进入低功耗模式: <ul style="list-style-type: none"> - 与 EXTI 信号线连接的外设中断服务请求, 不论是否 NVIC 中断使能, 参见 EXTI 信号线连接表 - EXTI 生成的事件
	唤醒引脚、NRST 引脚、RTC 中断请求、IWDG、BOR、POR
退出延迟	下面两个唤醒时间最长者为最终唤醒时间: <ul style="list-style-type: none"> - HSI 唤醒时间+Flash 唤醒时间 - 调压器唤醒时间

5.4 PWR 寄存器

PWR 寄存器仅支持 32bit 访问。

5.4.1 电源管理控制寄存器 1 (PWR_CR1)

偏移地址: 0x00

复位值: 0x0000 9001;

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BGREN	LPR	BORPEN	VBGOS	PDLDO	Res.										LPMS
rw	rw	rw	rw	rw	rw										rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	BGREN	BGR使能 0: 禁止BGR 1: 使能BGR
14	LPR	VPULL使能 0: 禁止VPULL 1: 使能VPULL 主Regulator被关闭时, 打开VPULL可提供较小电流给系统使用。
13	BORPEN	BOR和POR在低功耗模式下间断性检测 0: 不使能 1: 使能
12:11	VBGOS	VBGR选择 当BGR使能时, 选择使用VBGR做为OPAMP/CMP/DAC模块的参考

源，使用这个位进行控制，ADC如果使用VBGR做为参考源，不受这位选择控制，它在ADC_CFGR1的VBGSEL进行选择，当BGR使能后，用户需要等待100us时间让让VBGR信号稳定。

- 00:1.25V
- 01:2.5V;
- 10:3V (默认)
- 11:4V

- 10 PDLDO 主Regulator关闭
0: 不关闭
1: 关闭主Regulator, 1.5V来自VPULL
- 9:1 保留 必须保持复位值
- 0 LPMS 低功耗模式选择:
0: Stop模式
1: 预留, 暂未定义

5.4.2 电源管理控制寄存器 2 (PWR_CR2)

偏移地址: 0x04

复位值: 0x0000 02CC

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																
Res.												PVDIE	HTEN	LTEN	FTEN	RTEN
												rw	rw	rw	rw	rw
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																
FEN	FTIME[2:0]			Res.		PVD_HYS		PVDS		PVDT					PVDE	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:21	保留	必须保持复位值
20	PVDIE	PVD报警中断使能 0: 禁止 1: 使能
19	HTEN	监测电压高于阈值报警使能 0: 禁止 1: 使能
18	LTEN	监测电压低于阈值报警使能 0: 禁止 1: 使能

17	FTEN	<p>监测电压从高于阈值到低于阈值变化时报警使能</p> <p>0: 禁止</p> <p>1: 使能</p>
16	RTEN	<p>监测电压从低于阈值到高于阈值变化时报警使能</p> <p>0: 禁止</p> <p>1: 使能</p>
15	FEN	<p>PVD监测信号数字滤波使能</p> <p>当进入Stop时，PVD除滤波功能无效外其他功能均正常。</p> <p>0: 禁止</p> <p>1: 使能</p>
14:12	FTIME[2:0]	<p>PVD监测信号数字滤波时间</p> <p>000: 滤波时间为2个PCLK</p> <p>001: 滤波时间为4个PCLK</p> <p>010: 滤波时间为8个PCLK</p> <p>011: 滤波时间为16个PCLK</p> <p>100: 滤波时间为32个PCLK</p> <p>101: 滤波时间为64个PCLK</p> <p>110: 滤波时间为128个PCLK</p> <p>111: 滤波时间为256个PCLK</p>
9:8	PVD_HYS	<p>PVD迟滞</p> <p>00: 低档</p> <p>01: 中低档</p> <p>10: 中档</p> <p>11: 高档</p>
7:6	PVDS[1:0]	<p>PVD监测信号源选择:</p> <p>00: PC4引脚输入电压</p> <p>01: PD4引脚输入电压</p> <p>10: PD6引脚输入电压</p> <p>11: V_{DD}</p> <p><i>注意: 当PVD监测信号选择PC4/PD4/PD6时, 这些引脚输入电压不能高于V_{DD}。</i></p>
5:1	PVDT[4:0]	<p>PVD阈值等级</p> <p>00000: level0上升/下降</p> <p>00001: level1上升/下降</p> <p>00010: level2上升/下降</p> <p>...</p> <p>00110: level6上升/下降</p> <p>00111: level7上升/下降</p> <p>01000: level8上升/下降</p>

...
 10000: level16上升/下降
 10001: level17上升/下降
 10010: level18上升/下降
 ...
 11100: level28上升/下降
 11101: level29上升/下降
 11110: level30上升/下降
 11111: level31上升/下降

0 PVDE PVD监测使能
 0: 禁止
 1: 使能

5.4.3 电源管理状态寄存器 2 (PWR_SR2)

偏移地址: 0x14

复位值: 0x0000 0000;

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

位/位域	名称	描述
31:13	保留	必须保持复位值
12	PVDF	PVD报警中断状态标志 当发生PVD报警事件时，该位置位；通过置位CPVDF (PWR_SCR寄存器)来清零该位。
11	PVDS	PVD当前的监测状态（如果滤波使能，该状态表示过滤后的状态） 0: 被监测的信号电压高于PVD的阈值 1: 被监测的信号电压低于PVD的阈值
10:0	保留	必须保持复位值

5.4.4 电源管理状态清除寄存器 (PWR_SCR)

偏移地址: 0x18

复位值: 0x0000 0000

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

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

位/位域	名称	描述
31:13	保留	必须保持复位值
12	CPVDIF	清除PVD报警中断状态标志 写1将PWR_SR2寄存器中PVDIF标志清0。
11:0	保留	必须保持复位值

6 复位和时钟控制器（RCC）

6.1 复位

芯片复位分别为 POR/PDR 复位、BOR 复位和系统复位。

6.1.1 POR/PDR 复位

POR/PDR 复位范围如下：

- V_{CORE} 域下的所有寄存器（包括 Trim 相关的寄存器）；

6.1.2 系统复位

系统复位将复位 V_{CORE} 域的寄存器（不包括 Trim、系统配置选项字节寄存器）。

只要发生以下事件之一，就会产生系统复位：

- NRST 引脚低电平（请参见 [NRST 引脚低电平复位](#)）
- 窗口看门狗事件（WWDG 复位）
- 独立看门狗事件（IWDG 复位）
- 软件复位（请参见 [软件复位](#)）
- 选项字节加载复位
- LOCKUP 复位
- BOR 复位

可通过查询寄存器 RCC_CSR 中的复位标志确定复位源。

NRST 引脚低电平复位

对于外部复位，当 NRST 引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20us（典型值）的毛刺信号，因此，加到该引脚上的低电平信号必须大于 20us，才能保证芯片可靠复位。

软件复位

要对器件进行软件复位，必须将 Cortex®-M0+寄存器（[应用中断和复位控制寄存器](#)）中的 SYSRESETREQ 位置 1。

6.2 时钟

时钟控制电源提供了一系列频率的时钟源，包括：

- HSI（内部高速）18MHz RC 振荡器时钟

- MSI（内部低速）3.686MHz RC 振荡器时钟
- HSE 外部高速时钟(4~24MHz)
- LSI（内部慢速）32kHz RC 振荡器时钟
- LSE 外部低速时钟(32.768kHz)

对于每个时钟源，都可单独使能或禁止，以降低功耗。

复位后，HSISYS 作为系统时钟。

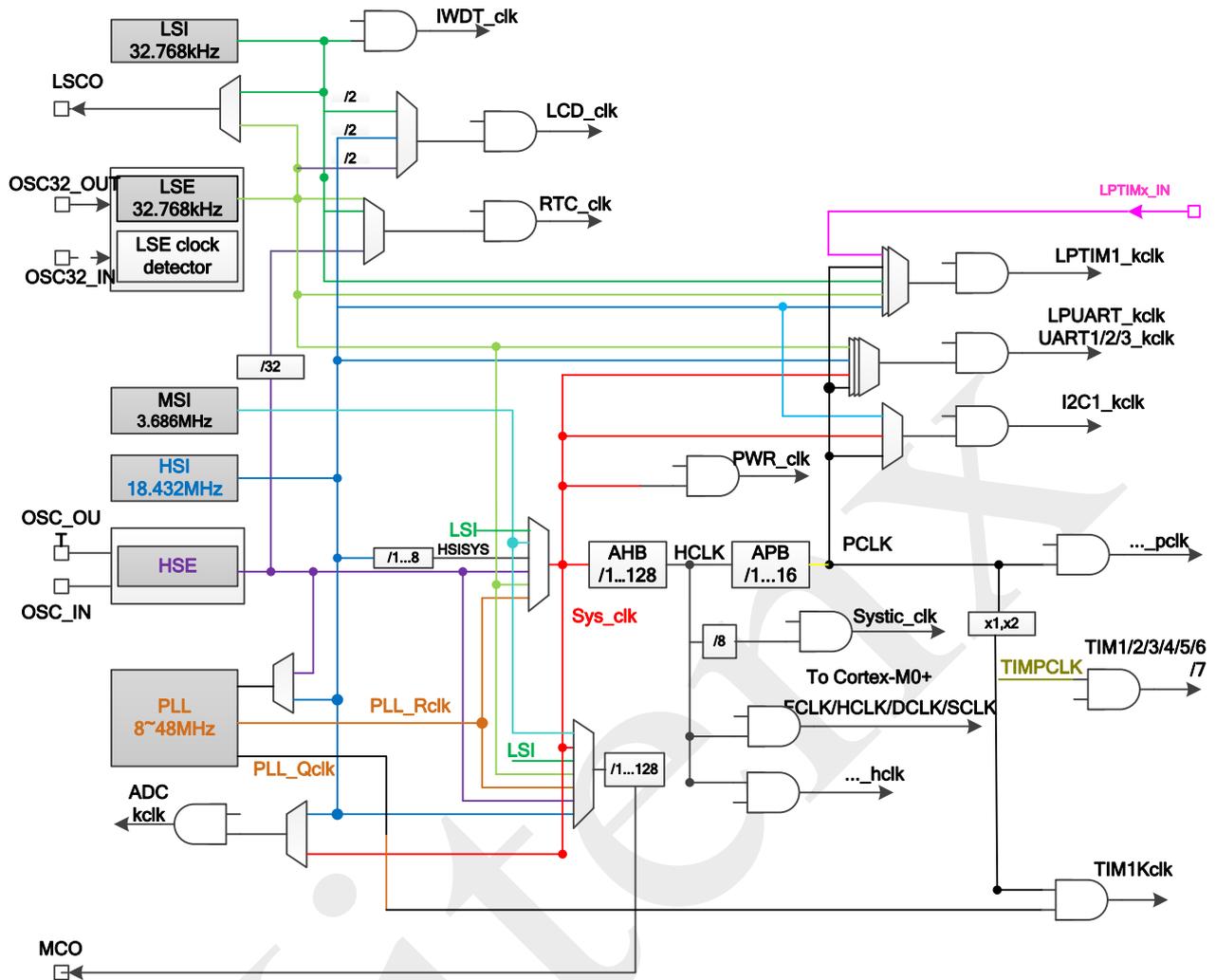
可通过多个预分频器配置 AHB、APB1 和 APB2 的时钟。AHB、APB1 和 APB2 的最大频率为 48MHz。

下表为输入/输出时钟关系表：

表 6-1 时钟关系表

输出时钟	输入时钟	说明
HSISYS	HSI	HSI 的 1 到 8 分频。 系统复位后，HSI 时钟作为系统时钟源
PLLCLK	HSI、HSE 和 HSE 的 2 分频	PLL 的输出时钟
SYSCLK	LSE、LSI、HSE、MSI、PLLCLK、HSISYS	系统时钟，通过寄存器选择初始时钟源作为 SYSCLK 输入
HCLK	SYSCLK	AHB 时钟，来自 SYSCLK 的 1 到 128 分频
PCLK1	HCLK	APB1 时钟，来自 HCLK 的 1 到 16 分频
PCLK2	HCLK	APB2 时钟，来自 HCLK 的 1 到 16 分频
TIMPCLK1/2	PCLK1/2	PCLK1/2 的 1 或倍频

图 6-1 时钟树



6.2.1 HSE 时钟

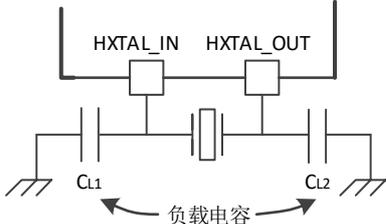
外部高速时钟信号（HSE）有 2 种时钟源：

- 外部晶体谐振器/外部陶瓷谐振器（外部晶体模式）
- 外部时钟源（旁路模式）

晶体谐振器和负载电容必须尽可能地靠近芯片引脚，以尽量减少输出失真和起振稳定时间。负载电容值必须根据所选晶体谐振器的不同做适当调整。

表 6-2 HSE 时钟源

时钟源	硬件配置
外部时钟 (旁路模式)	

时钟源	硬件配置
外部晶体谐振器/ 外部陶瓷谐振器 (外部晶体模 式)	

外部晶体谐振器（外部晶体模式）

4~24MHz 外部晶体可提供更高精度的时钟。

相关硬件配置，可参见表：[HSE 时钟源](#)。有关参数信息，请参见 Datasheet 相关章节。

等待 HSE 时钟的稳定时间可通过[时钟稳定配置寄存器 \(RCC_STABCR\)](#) 中的 HSESS[2:0]位域来配置。

HSE 可通过 RCC_CR 寄存器中的 HSEON 位来使能或禁止。

[时钟控制寄存器 \(RCC_CR\)](#) 中的 HSERDY 标志指示 HSE 时钟是否稳定。如在[时钟中断使能寄存器 \(RCC_CIER\)](#) 中使能中断，则 HSE 稳定时便会产生中断。

使用 RCC_CR 寄存器中的 HSEGAIN 位域，可更改起振电路的驱动能力，以实现启动时间和低功耗之间的最佳平衡，增强对不同晶体的兼容性。

注意： HSE 时钟使能后，修改该位域，可能引起 HSE 时钟异常。

HSE 时钟配置：

- 使能 HSE 外部晶体模式：配置 RCC_CR 寄存器中的 HSEON 位为 1，之后 HSEBYP 位清 0；
- 使能 HSE 旁路模式：配置 RCC_CR 寄存器中的 HSEON 位为 1，之后 HSEBYP 位置 1；
- 禁止 HSE 时钟：配置 RCC_CR 寄存器中的 HSEON 位为 0。

表 6-3 HSE 时钟配置

HSEON	HSEBYP	状态
0	x	禁止 HSE
1	0	使能 HSE 外部晶体模式
1	1	使能 HSE 旁路模式

外部时钟源（HSE 旁路模式）

在此模式下，必须提供外部时钟源。最高频率不超过 48MHz。

RCC_CR 寄存器的 HSEON 和 HSEBYP 位置 1 可选择此模式。必须使用占空比为 40~60% 的外部时钟信号（方波、正弦波或三角波）来驱动 HSE_IN 引脚，HSE_OUT 引脚可用作 GPIO。

6.2.2 HSI 时钟

HSI 时钟是由内部的 18MHz RC 振荡器生成的。可用作系统时钟或 PLL 输入。HSI 从启动到稳定时间仅需 2us(典型值)，全电压全温度范围内的精度为 ±1.5%，可免外接晶体谐振器。

校准

由于生产制造导致不同芯片 RC 振荡器频率的差异，因此会对 HSI 进行出厂校准，达到全温度下 ±1.5% 的精度。

可通过 *HSITRIM 寄存器 (RCC_HSITRIMR)* 的 HSIT[9:0]位域对所选 HSI 频率进行微调。

如何测量 HSI 频率，可参见：[基于 TIM4 的内部/外部时钟测量](#)。

HSI 可通过 RCC_CR 寄存器中的 HSION 位来使能或禁止。

时钟控制寄存器 (RCC_CR) 中的 HSIRDY 标志指示 HSI 是否稳定。如在 *时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则 HSI 稳定时便会产生中断。

进入 Stop 模式，被唤醒后系统时钟自动切换为 HSI SYS（HSI 的 2 分频）。

6.2.3 MSI 时钟

MSI 时钟是由内部 RC 振荡器生成的。频率为 3.686MHz。

MSI 可通过 RCC_CR 寄存器中的 MSION 位来使能或禁止。

时钟控制寄存器 (RCC_CR) 中的 MSIRDY 标志指示 MSI 是否稳定。如在 *RCC_CIER* 寄存器中使能中断，则 MSI 稳定时便会产生中断。

校准

由于生产制造导致不同芯片 RC 振荡器频率的差异，因此会对 MSI 进行出厂校准，达到全温度下 ±1.5% 的精度。

可通过 *MSITRIM 寄存器 (RCC_MSITRIMR)* 的 MSIT[9:0]位域对所选 MSI 频率进行微调。

如何测量 MSI 频率，可参见：[基于 TIM4 的内部/外部时钟测量](#)。

6.2.4 PLL 时钟

内部 PLL 的输入时钟源由 HSI 或 HSE 时钟提供，其时钟输入范围为 4MHz~24MHz；时钟输出频率范围为 8MHz~48MHz。可参见 [图：时钟树](#) 和 [时](#)

钟数和PLL 配置寄存器 (RCC_PLLCFGR)。

注意: PLL 作为系统时钟，其输出频率最高不超过 48MHz。

PLL 配置 (输入时钟、倍频因子和分频因子的选择) 必须在使能 PLL 之前完成，一旦使能 PLL 后，这些参数不能更改。

要修改 PLL 配置，请按照以下步骤操作：

- 1) 将**时钟控制寄存器 (RCC_CR)** 中的 PLLON 位设置为 0 禁止 PLL；
- 2) 等待 PLLRDY 为 0，表示 PLL 已停止；
- 3) 根据需要更改参数；
- 4) 通过将 PLLON 位置 1，再次使能 PLL；
- 5) 等待 PLLRDY 为 1，表示 PLL 已稳定
- 6) 通过配置 **PLL 配置寄存器 (RCC_PLLCFGR)** 中的 PLLREN，使能所需要的 PLLRCLK 输出；

时钟控制寄存器 (RCC_CR) 中的 PLLRDY 标志指示 PLL 时钟是否稳定。如果在 RCC_CIER 寄存器中使能中断，则 PLL 稳定时便会产生中断。

如果 PLLRCLK 用作系统时钟，则 PLLREN 软件清 0 无效。

当 PLL 时钟发生异常时，系统时钟不会发生变化，但会产生相应的中断。请参阅 **PLL CSS**。

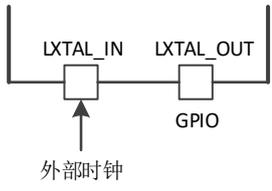
6.2.5 LSE 时钟

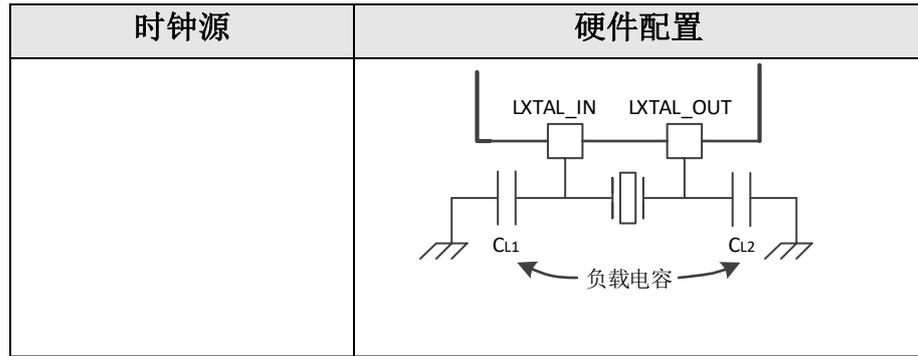
外部低速时钟信号 (LSE) 有 2 种时钟源：

- 外部晶体谐振器/外部陶瓷谐振器 (外部晶体模式)
- 外部时钟源 (旁路模式)

晶体谐振器和负载电容必须尽可能地靠近芯片引脚，以尽量减少输出失真和起振稳定时间。负载电容值必须根据所选晶体谐振器的不同做适当调整。

表 6-4 LSE 时钟源

时钟源	硬件配置
外部时钟 (旁路模式)	
外部晶体谐振器/ 外部陶瓷谐振器 (外部晶体模式)	



外部晶体（LSE 外部晶体模式）

32.768 kHz 低速外部晶体，可作为实时时钟（RTC）的时钟源来提供实时计数或其它定时功能，具有功耗低的优点。

相关硬件配置，可参见表：[LSE 时钟源](#)。有关参数信息，请参见 Datasheet 相关章节。

LSE 通过[控制寄存器（RCC_BDCR）](#)中的1位来使能或禁止。

等待 LSE 时钟的稳定时间可通过[时钟稳定配置寄存器（RCC_STABCR）](#)中的 LSESS[2:0]来配置。

RCC_BDCR 寄存器的 LSE RDY 标志指示 LSE 时钟是否稳定。如在 RCC_CIER 寄存器中使能中断，则 LSE 稳定时便会产生中断。

使用 RCC_BDCR 寄存器中的 LSE DRV[1:0]位域，可更改起振电路的驱动能力，以实现启动时间和低功耗之间的最佳平衡，增强对不同晶体的兼容性。

注意： LSE 时钟使能后，修改该位域，可能引起 LSE 时钟异常。

LSE 时钟配置：

- 使能 LSE 外部晶体模式：配置 RCC_BDCR 寄存器中的 LSEON 位为 1，之后 LSEBYP 位清 0；
- 禁止 LSE 时钟：配置 RCC_BDCR 寄存器中的 LSEON 位为 0。

表 6-5 LSE 时钟配置

LSEON	LSEBYP	状态
0	x	禁止 LSE
1	0	使能 LSE 外部晶体模式
1	1	使能 STX 旁路模式

6.2.6 LSI 时钟

LSI 时钟频率为 32 kHz，可在 Stop 模式下保持运行，为 IWDG、RTC、LCD 提供时钟。

LSI 可通过[时钟控制/状态寄存器（RCC_CSR）](#)中的 LSION 位来使能或禁止。

RCC_CSR 寄存器中的 LSIRDY 标志指示 LSI 是否稳定。如在 [时钟中断使能寄存器 \(RCC_CIER\)](#) 中使能中断，则 LSI 稳定时便会产生中断。

6.2.7 系统时钟

可以使用下列不同的时钟源来驱动系统时钟 (SYSCLK)：

- MSI
- HSI SYS
- HSE
- PLLRCLK
- LSI
- LSE

系统时钟最高频率为 48 MHz。系统复位后，选择 HSI SYS (HSI 的 2 分频) 作为系统时钟。

当时钟切换时，如果选择尚未稳定的时钟源，则切换在该时钟源稳定后才会进行。[时钟配置寄存器 \(RCC_CFGR\)](#) 中的 SWS 指示当前哪个时钟正作为系统时钟。

6.2.8 LSE CSS

将 [控制寄存器 \(RCC_BDCR\)](#) 中的 LSECSSON 位置 1，来使能 LSE CSS。仅可通过或检测到 LSE 故障后禁止该功能。

LSE CSS 使能后，其检测功能在 LSE RDY 置 1 后生效，并且当 LSI 时钟稳定 (LSE RDY 为 1，且 LSIRDY 为 1) 后，该功能才能正常检测。

LSE CSS 可工作在所有工作模式；系统复位，LSE CSS 功能不受影响。

当检测到 LSE 故障时，LSE 不会再向 RTC 提供 LSE 时钟，但其寄存器不受影响。

当 LSE 时钟发生故障时，RCC_CIFR 寄存器中的 LSECSSF 被置位，并生成 LSE CSS 中断。该中断与 Cortex®-M0+ NMI (不可屏蔽中断) 异常相连接。因此，在 NMI_ISR 中，必须将 [时钟中断清零寄存器 \(RCC_CICR\)](#) 中的 LSECSSC 位置 1，以清除 LSE CSS 中断。之后软件需禁止 LSE CSS (LSECSSON 为 0)，禁止 LSE 时钟 (LSEON 为 0)，更改 RTC 时钟源，或者采取必要措施来确保应用的安全。

如果 LSE 作为系统时钟，当检测到故障时，则系统时钟切换为 LSI。

注意： 当 LSE 时钟发生故障时，RCC_BDCR 寄存器中的 LSECSSD 也会被置位，软件需禁止 LSE CSS 功能 (此时 LSECSSD 位将自动清 0)。

当使能 LSE CSS 时，LSE 的频率必须高于 20kHz，以免发生 CSS 误报。

6.2.9 PLL CSS

可由软件通过写入**时钟控制寄存器 (RCC_CR)**中的 PLLCSON 位, 来使能 PLL CSS 功能。

当 PLL CSS 使能后, 其检测功能在 PLL 时钟稳定后生效。如果 PLL 时钟失锁, PLLRDY 被自动清 0, 且 RCC_CIFR 寄存器中的 PLLCSSF 被置位, 并生成 PLL CSS 中断。该中断与 Cortex®-M0+ NMI (不可屏蔽中断) 异常相连接。因此, 在 NMIISR 中用户必须将**时钟中断清零寄存器 (RCC_CICR)**中的 PLLCSSC 位置 1, 以清除 PLL CSS 中断。

6.2.10 外设异步时钟选择

所有外设时钟均由其总线时钟 (HCLK、PCLK1 和 PCLK2) 提供, 但有些外设的内核时钟可单独配置。

表 6-6 外设异步时钟来源

模块	异步时钟来源
IWDG	LSI
ADC	SYSCLK 的 1/2/4 分频或 HSI
USART1	PCLK2、LSE、HSI、SYSCLK
LPUART	PCLK1、LSE、HSI、SYSCLK
LPTIM	LSI、LSE、HSI、PCLK1
TIM1	PLLQCLK、PCLK2 的 1 或 2 倍频
I2C	PCLK1、HSI、SYSCLK
RTC	LSI、LSE、HSE 的分频时钟
LCD	LSI 和 LSE 的 2 分频(LCD)或 HSI 的 2 分频 (LED)

6.2.11 外设时钟使能寄存器

每个外设时钟均有使能控制位, 参见 RCC_AHBxENR、RCC_APBxENR 寄存器。

当外设时钟未使能时, 不支持外设寄存器进行读写访问。

注意: 使能位的同步机制可为外设产生无干扰时钟。使能位置 1 后, 在时钟使能前存在时长为 2 个时钟周期的延迟。使能外设时钟后, 软件必须等待一段延迟才能访问外设寄存器。

6.2.12 TIMx 时钟

TIMx 时钟根据 APB 预分频值自动设定: 如果 APB 预分频值为 1, 则 TIM 时钟频率等于 APB 频率; 否则, 等于 APB 频率的 2 倍。

TIM1 计数时钟除了支持 TIMx 时钟外还可以使用 PLLQ 时钟作为计数时钟。

6.2.13 IWDG 时钟

IWDG 时钟通过选项字节或软件访问使能，此时 LSI 将被强制使能，且不能被禁止。在 LSI 使能并稳定后，为 IWDG 提供时钟。

6.2.14 ADC 时钟

ADC 异步时钟来自 SYSCLK 的分频时钟或 HSI，它可以达到 48MHz。当选择 SYSCLK 时，可有如下预分频值：1、2、4。具体时钟配置可参见[数模转换器\(ADC\): 时钟源](#)。

6.2.15 LCD 时钟

LCD 时钟源，可以是 LSE、LSI 分频时钟。由 [LCD 控制寄存器 \(LCD_CR\)](#) 中的 LCDCKS 位域进行编程选择。

- LSE 外部低速时钟
- LSI (内部低速) RC 振荡器时钟

LED 时钟源固定采用为 HSI 的 2 分频时钟，当 LED 使能时，如果 HSI 关闭，硬件将自动使能 HSI。

6.2.16 RTC 时钟

RTC 时钟源，可以是 LSE、LSI 或 HSE 的分频时钟。由[时钟控制/状态寄存器 \(RCC_BDCR\)](#) 中的 RTCSEL[2:0]位域进行编程选择。

- LSE 外部低速时钟
- LSI (内部低速) RC 振荡器时钟
- HSE 的 128 分频、256 或 512 或 1024 分频时钟

选择 RTC 时钟源后，仅在以下情况下可更改时钟源：检测到 LSE CSS 故障。

只有 PCLK 频率大于或等于 RTC_KCLK 频率，才可以保证 RTC 正常运行。

6.2.17 时钟输出功能

- MCO (microcontroller clock output)

允许将时钟通过 MCO 引脚输出到外部。可选择以下时钟信号之一作为 MCO 输出时钟。

- HSI 二分频
- MSI
- LSI
- HSE

- LSE
- SYSCLK
- PLLRCLK

由*时钟配置寄存器(RCC_CFGR)*中的 MCOSEL[3:0]位域选择输出的时钟。所选时钟可以通过 RCC_CFGR 寄存器中的 MCOPRE[2:0]位域进行分频。

- LSCO (low speed clock output)

允许将低速时钟通过 LSCO 引脚输出到外部：

- LSI
- LSE

在 Stop 模式下，可正常输出。时钟源选择由*时钟控制/状态寄存器(RCC_BDCR)*中的 LSCOSEL 控制，并通过 LSCOEN 使能时钟输出。

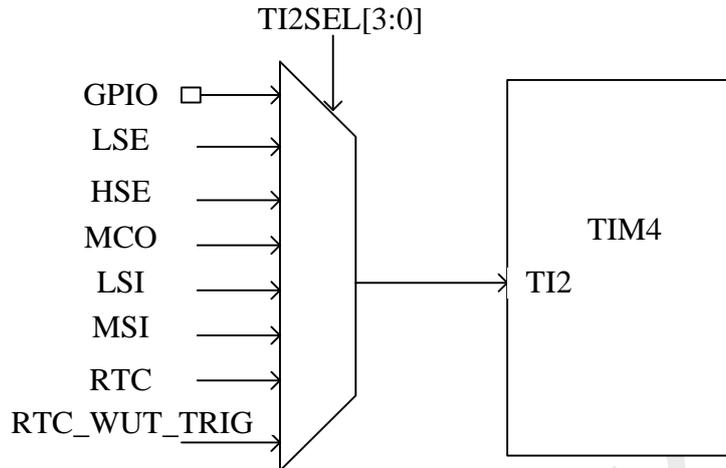
MCO 时钟输出需要配置 MCO 引脚为复用功能；LSCO 引脚配置为 GPIO 复用功能。

6.2.18 基于 TIM4 的内/外部时钟测量

所有板上时钟源的频率都可通过对 TIM4 通道 1 的输入捕获进行间接测量，如图：所示。

通过配置 *TIM4_TISEL 寄存器的 TI2SEL[3:0]* 来选择不同的输入捕获源：

- GPIO (请参见数据手册中的复用功能映射)
- LSE
- HSE
- MCO(此选择由时钟配置寄存器(RCC_CFGR)的 MCOSEL[3:0] 位控制)
- LSI
- MSI
- RTC
- RTC_WUT_TRIG (唤醒中断信号。在这种情况下，应使能 RTC 中断)

图 6-2 TIM4 在捕获模式下的频率测量


MSI 校准

将 LSE 连接到 TIM4 的通道 2，利用 LSE 的高精度（通常为几十 ppm），和定时器的输入捕获功能能够精确测量 MSI 的时钟（MSI 作为系统时钟）。借助 LSE 信号连续边沿之间的 MSI 时钟计数值，即可对 MSI 时钟周期进行测量，通过不断调整校准值，直到 MSI 频率达到精度要求。

其基本原理是基于相对的测量（例如，MSI/LSE 比），因此，精度与两个时钟源之比紧密相关。比率越大，测量效果越好。

MSI Trim 寄存器（RCC_MSITRIMR）为 MSI 的校准寄存器。

如果 LSE 不可用，为了尽可能达到最精确的校准，HSE_RTC 将是更好的选择。

但是，如果 MSI 时钟频率较低（通常在 3.6 MHz 左右），测量精度会不够好。在这种情况下，建议：

- 累加连续多次捕获的结果。
- 使用定时器的输入捕获预分频值（最多每 8 个周期进行 1 次捕获）。
- 使用 RTC 唤醒中断信号（如果 RTC 的时钟由 LSE 提供）作为通道 2 输入捕获的输入。这样可提高测量精度，为此必须使能 RTC 唤醒中断。

6.3 低功耗模式

- 可通过软件禁止 AHB 和 APB 未使用的外设时钟，包括 DMA 时钟。
- 在 Sleep 模式下，与 Run 模式一致，仅关闭 CPU 时钟。
- 在 Stop 模式下，将停止 V_{CORE} 域中的所有快速时钟，并禁止 PLL、HSISYS、MSI 和 HSE。当 HSI_AWON 配置为 1 时，即使 MCU 处于 Stop 模式（如果选择 HSI 作为该外设的时钟源），但所有 USART1、LPUART 和 I2C1 也均可用 HSI 时钟。

- 在 Stop 模式（如果选择 LSE 作为该外设的时钟源）并使能 LSE 时，LSE 可为 USART1 和 LPUART 提供时钟。在这种情况下，LSE 在 Stop 模式下始终保持正常输出。

通过将 *DBG_CR 寄存器* 中的 DBG_STOP 位置 1，可使时钟在 Stop 模式下保持使能状态，便于调试。

当退出 Stop 模式时，系统时钟为 HSI SYS。

如果正在执行 Flash 编程操作，则将延迟到 Flash 操作结束后再进入 Stop 模式。如果正在访问 APB 总线，则将延迟到 APB 访问结束后再进入 Stop 模式。

6.4 RCC 寄存器

6.4.1 时钟控制/状态寄存器 (RCC_CR)

偏移地址：0x00

复位值：0x0000 0D00

访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	HSE DIV			Res	PLL CSSON	PLL RDY	PLL ON	HSE X2S	HSE X1S	HSE GAIN	Res	HSE RDY	HSE BYP	HSEON	
15	rw	rw	rw	rs	r	rw	rw	rw	rw	rw	rs	r	rw	rw	
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			HSIDIV	HSI RDY	HSIKERON	HSI ON	Res.			MSIRDY	MSION	Res.			
			rw	rw	r	rw	rw				r	rw			

位/位域	名称	描述
31	保留	必须保持复位值
30:28	HSE DIV[2:0]	HSE 时钟分频系数 00x: 保留 010: 1分频 011: 2分频 100: 4分频 101: 8分频 110: 16分频 111: 32分频
27	保留	必须保持复位值
26	PLLCSSON	PLL CSS 使能 当该位置 1，硬件将在 PLL 时钟失锁后，产生 PLL CSS 报警，同时清除 PLLRDY 位。 0: 禁止（当 PLL 失锁后，该功能被禁止） 1: 使能（当 PLL 稳定后，该检测功能才会生效；PLL 禁止

		时，该检测功能自动关闭，同时该位清0)
25	PLLRDY	PLL时钟稳定标志 用于指示PLL时钟稳定状态。 0: PLL时钟未稳定 1: PLL时钟已稳定
24	PLLON	PLL时钟使能 当进入Stop模式时由硬件清0。 当PLL时钟用作系统时钟时，则此位不可清0。 0: 禁止 1: 使能
23	HSEX2S	HSE XO输出管脚选择 0: PC1 1: PC3
22:21	HSEX1S	HSE XI输入管脚选择 00: PA4 01: PC0 10: PC2 11: 保留
20	HSEGAIN	HSE驱动能力选择 0: 正常驱动 1: 大驱动
19	保留	必须保持复位值
18	HSERDY	HSE时钟稳定标志 当配置HSEON为0时，HSERDY将在3个HSE时钟周期后自动清0。 0: HSE未稳定 1: HSE已稳定
17	HSEBYP	HSE旁路模式 0: 外部晶体模式 1: 旁路模式 注意，只有HSEON为1，该位配置才有效
16	HSEON	HSE时钟使能 在进入Stop模式时自动停止HSE。如果HSE直接或间接用作系统时钟，则该位不可清0。 0: 禁止 1: 使能

15:13	保留	必须保持复位值
12:11	HSIDIV[1:0]	<p>HSI时钟分频因子</p> <p>00: HSI时钟1分频作为HSISYS时钟</p> <p>01: HSI时钟2分频作为HSISYS时钟（缺省）</p> <p>10: HSI时钟4分频作为HSISYS时钟</p> <p>11: HSI时钟8分频作为HSISYS时钟</p>
10	HSIRDY	<p>HSI时钟稳定标志</p> <p>当配置HSION为0时，HSI_RDY将在3个HSI时钟周期后自动清0。</p> <p>0: HSI时钟未稳定</p> <p>1: HSI时钟已稳定</p>
9	HSIKERON	<p>始终为外设使能HSI</p> <p>用于强制开启HSI（即使在Stop模式下）。HSI为USART1、LPUART和I2C1外设提供时钟。HSI在Stop模式下保持输出，可避免由于HSI启动时间而减慢通信速度。该位不受HSION位影响。</p> <p>0: 对HSI时钟无影响</p> <p>1: 即使在Stop模式下也保持HSI使能</p>
8	HSION	<p>HSI时钟使能</p> <p>在进入Stop模式时硬件会停止HSI时钟。</p> <p>0: 禁止</p> <p>1: 使能</p>
7:5	保留	必须保持复位值
4	MSIRDY	<p>MSI时钟稳定标志</p> <p>当配置MSION为0时，MSIRDY将在3个MSI时钟周期后自动清0。</p> <p>0: MSI时钟未稳定</p> <p>1: MSI时钟已稳定</p>
3	MSION	<p>MSI时钟使能</p> <p>当MSI用作系统时钟时，该位不能被清0。</p> <p>0: 禁止</p> <p>1: 使能</p>
2:0	保留	必须保持复位值

6.4.2 时钟稳定配置寄存器 (RCC_STABCR)

偏移地址：0x04

复位值：0x0000 xxxx(从 NVR 加载，加载前上电默认值：0x0B23)

访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Res.																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.		MSISS[2:0]			LSESS[2:0]			Res.		HSISS[2:0]			Res.		HSESS[2:0]		
		rw	rw	rw	rw	rw	rw			rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13:11	MSISS[2:0]	等待 MSI 时钟稳定的周期数 00: 4 个时钟周期 01: 8 个时钟周期(默认) 10: 16 个时钟周期 11: 32 个时钟周期 Bit2 预留，只使用低 2 位
10:8	LSESS[2:0]	等待 LSE 时钟稳定的周期数 000: 2 个时钟周期 001: 1024 个时钟周期 010: 4096 个时钟周期 011: 8192 个时钟周期 (默认) 100: 16384 个时钟周期 101: 32768 个时钟周期 110: 65536 个时钟周期 111: 131072 个时钟周期
7	保留	必须保持复位值
6:4	HSISS[2:0]	等待 HSI 时钟稳定的周期数 000: 2 个时钟周期 001: 4 个时钟周期 010: 8 个时钟周期 (默认) 011: 16 个时钟周期 1xx: 32 个时钟周期
3	保留	必须保持复位值

2:0 HSESS[2:0] 等待 HSE 时钟稳定的周期数
 000: 2 个时钟周期
 001: 1024 个时钟周期
 010: 4096 个时钟周期
 011: 8192 个时钟周期 (默认)
 100: 16384 个时钟周期
 101: 32768 个时钟周期
 110: 65536 个时钟周期
 111: 131072 个时钟周期

6.4.3 时钟配置寄存器 (RCC_CFGR)

偏移地址: 0x08

复位值: 0x0000 0000

访问: 0~2 等待周期, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCOPRE[2:0]			Res.	MCOSEL[2:0]			Res.			P2PRE[2:0]				
	rw	rw	rw		rw	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
Res.	PIPRE[2:0]			Res.	HPRE[2:0]			Res.	SWS[2:0]			SW[2:0]			
	rw	rw	rw		rw	rw	rw			r	r	r	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值
30:28	MCOPRE[2:0]	时钟输出预分频 需在MCO输出使能之前更改此预分频值。 000: 1分频 001: 2分频 010: 4分频 011: 8分频 100: 16分频 101: 32分频 110: 64分频 111: 128分频
27	保留	必须保持复位值
26:24	MCOSEL[2:0]	时钟输出源选择 000: 禁止MCO输出 001: 选择SYSCLK 010: 选择MSI 011: 选择HSI的二分频 100: 选择HSE

		101: 选择PLLCLK 110: 选择LSI 111: 选择LSE
23:19	保留	必须保持复位值
18:16	P2PRE[2:0]	APB2预分频器 用于选择APB2时钟 (PCLK2) 0xx: HCLK 1分频 100: HCLK 2分频 101: HCLK 4分频 110: HCLK 8分频 111: HCLK 16分频 <i>注意: 当更改APB2预分频器值时, 其过程会插入0~15个等待周期。</i>
15	保留	必须保持复位值
14:12	P1PRE[2:0]	APB1预分频器 用于选择APB1时钟 (PCLK1) 0xx: HCLK 1分频 100: HCLK 2分频 101: HCLK 4分频 110: HCLK 8分频 111: HCLK 16分频 <i>注意: 当更改APB1预分频器值时, 其过程会插入0~15个等待周期。</i>
11	保留	必须保持复位值
10:8	HPRE[2:0]	AHB 预分频器 用于选择AHB时钟 000: SYSCLK 1分频 001: SYSCLK 2分频 010: SYSCLK 4分频 011: SYSCLK 8分频 100: SYSCLK 16分频 101: SYSCLK 32分频 110: SYSCLK 64分频 111: SYSCLK 128分频 <i>注意: 当更改AHB预分频器值时, 其过程会插入0~15个等待周期。</i>
7:6	保留	必须保持复位值

- 5:3 SWS[2:0] 系统时钟源指示
 000: HSYSYS
 001: HSE
 010: PLLRCLK
 011: LSI
 100: LSE
 101: MSI
 其它: 保留
- 2:0 SW[2:0] 系统时钟源选择
 000: HSYSYS用作系统时钟
 001: HSE用作系统时钟
 010: PLLRCLK用作系统时钟
 011: LSI用作系统时钟
 100: LSE用作系统时钟
 101: MSI用作系统时钟
 其它: 保留
*注意: 当系统时钟源切换时, 切换过程会插入1~2个等待周期。
 配置系统时钟源之前需先打开对应时钟源开关。*

6.4.4 PLL 配置寄存器 (RCC_PLLCFGR)

偏移地址: 0x0C

复位值: 0x0000 0118

访问: 无等待, 字、半字、字节访问。

此寄存器用于根据公式配置 PLL 时钟输出:

$$f_{VCO} = f_{PLLIN} \times (N/M)$$

$$f_{PLLRCLK} = f_{VCO}/PLLRDIV$$

$$f_{PLLQCLK} = f_{VCO}/PLLQDIV$$

其中 f_{PLLIN} 是 PLL 输入时钟; f_{VCO} 是 PLL VCO 时钟; $f_{PLLRCLK}$ 是 PLLRCLK, $f_{PLLQCLK}$ 是 PLLQCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				PLLQ[2:0]			PLLQEN	Res.				PLLR[2:0]			PLLREN
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				PLLM[2:0]			PLLN[4:0]				Res.	PLLSRC[1:0]			
				rw	rw	rw	rw	rw	rw	rw	rw			rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27:25	PLLQ[2:0]	PLL VCO的Q分频系数

		000: 1分频
		001: 2分频
		010: 3分频
		011: 4分频
		100: 5分频
		101: 6分频
		110: 7分频
		111: 8分频
24	PLLQEN	PLLCLKQ时钟输出使能 0: 禁止 1: 使能
23:20	保留	必须保持复位值
19:17	PLLR[2:0]	PLL VCO的R分频系数 000: 1分频 001: 2分频 010: 3分频 011: 4分频 100: 5分频 101: 6分频 110: 7分频 111: 8分频
16	PLLREN	PLLCLKR时钟输出使能 0: 禁止 1: 使能
15:11	保留	必须保持复位值
10:8	PLLM[2:0]	PLL输入时钟的分频因子M 000: 1分频 001: 2分频(默认) 010: 3分频 011: 4分频 100: 5分频 101: 6分频 110: 7分频 111: 8分频 <i>注意: 此位域仅在PLL不使能的情况下才可被改写; 经PLLM设置后的时钟频率范围为: 4MHz~24MHz</i>
7:3	PLLN[4:0]	PLL输入时钟的倍频因子N 00000: 1倍频

- 00001: 2 倍频
- 00010: 3 倍频
- 00011: 4 倍频(默认)
- ⋮
- 11110: 31 倍频
- 11111: 32 倍频

*注意：此位域仅在PLL不使能的情况下才可被改写；
经PLL设置后的时钟频率范围为：8MHz~48MHz*

2 保留 必须保持复位值

1:0 PLLSRC[1:0] PLL输入时钟源选择
该位域只能在禁止PLL时钟输入时写入。
00: 无效值
01: HSE时钟2分频
10: HSI
11: HSE
*注意：不使用PLL时钟时，PLLSRC的值应配置为“00”，以降低
功耗。PLL输入时钟频率范围为：4MHz~24MHz。*

6.4.5 时钟中断使能寄存器 (RCC_CIER)

偏移地址：0x18

复位值：0x0000 0000

访问：无等待，字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										PLLRDYIE	HSERDYIE	HSIRDYIE	MSIRDYIE	LSERDYIE	LSIRDYIE
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值
5	PLLRDYIE	PLL稳定中断使能 0: 禁止 1: 使能
4	HSERDYIE	HSE稳定中断使能 0: 禁止 1: 使能
3	HSIRDYIE	HSI稳定中断使能

		0: 禁止 1: 使能
2	MSIRDYIE	MSI稳定中断使能 0: 禁止 1: 使能
1	LSERDYIE	LSE稳定中断使能 0: 禁止 1: 使能
0	LSIRDYIE	LSI稳定中断使能 0: 禁止 1: 使能

6.4.6 时钟中断标志寄存器 (RCC_CIFR)

偏移地址: 0x1C

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PLLCSSF	LSECSSF	Res.	Res.	PLLRDYF	HSERDYF	HSIRDYF	MSIRDYF	LSERDYF	LSIRDYF	
					r	r			r	r	r	r	r	r	

位/位域	名称	描述
31:11	保留	必须保持复位值
10	PLLCSSF	PLL CSS中断标志 当PLL CSS功能生效后, 若发生PLL失锁, 则该位被置1。将PLLCSSC位置1, 可清除该位。 0: 未触发PLL CSS中断 1: 触发PLL CSS中断 <i>注意: 当HSE作为PLL的时钟输入时, HSE的CSS检测异常, 该位同样被置位。</i>
9	LSECSSF	LSE CSS中断标志 当LSE CSS功能生效后, 若检测到故障时, 该位被置1。将LSECSSC位置1, 可清除该位。 0: 未触发LSE CSS中断 1: 触发LSE CSS中断

8	保留	必须保持复位值
7:6	保留	必须保持复位值
5	PLLRDYF	<p>PLL时钟稳定中断标志</p> <p>当PLL时钟稳定，且PLLRDYDIE置1时，该位被置1。</p> <p>将PLLRDYC位置1，可清除该位。</p> <p>0: 未触发PLL稳定中断</p> <p>1: 触发PLL稳定中断</p>
4	HSERDYF	<p>HSE时钟稳定中断标志</p> <p>当HSE时钟稳定，且HSERDYDIE置1时，该位被置1。</p> <p>将HSERDYC位置1，可清除该位</p> <p>0: 未触发HSE稳定中断</p> <p>1: 触发HSE稳定中断</p>
3	HSIRDYF	<p>HSI时钟稳定中断标志</p> <p>当HSI时钟稳定，且HSIRDYDIE置1时，该位被置1。</p> <p>将HSIRDYC位置1，可清除该位。</p> <p>0: 未触发HSI稳定中断</p> <p>1: 触发HSI稳定中断</p> <p><i>注意: 当 HSION 未置1, 但外设选择HSI二分频作为外设时钟时, 该位不会置1, 也不会产生中断。</i></p>
2	MSIRDYF	<p>MSI时钟稳定中断标志</p> <p>当MSI时钟稳定，且MSIRDYDIE置1时，该位被置1。</p> <p>将MSIRDYC位置1，可清除该位。</p> <p>0: 未触发MSI稳定中断</p> <p>1: 触发MSI稳定中断</p>
1	LSERDYF	<p>LSE时钟稳定中断标志</p> <p>当LSE时钟稳定，且LSERDYDIE置1时，该位被置1。</p> <p>将LSERDYC位置1，可清除该位。</p> <p>0: 未触发LSE稳定中断</p> <p>1: 触发LSE稳定中断</p>
0	LSIRDYF	<p>LSI时钟稳定中断标志</p> <p>当LSI时钟稳定，且LSIRDYDIE置1时，该位被置1。</p> <p>将LSIRDYC位置1，可清除该位。</p> <p>0: 未触发LSI稳定中断</p> <p>1: 触发LSI稳定中断</p>

6.4.7 时钟中断清除寄存器 (RCC_CICR)

偏移地址: 0x20

复位值：0x0000 0000

访问：无等待，字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Res.																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.				PLL CSSC		LSE CSSC		Res.		Res.		PLL RDYC	HSE RDYC	HSI RDYC	MSI RDYC	LSE RDYC	LSI RDYC
						w	w	w			w	w	w	w	w	w	w

位/位域	名称	描述
31:11	保留	必须保持复位值
10	PLLCSSC	PLL CSS中断清除 0: 无影响 1: 将PLLCSSF标志清0
9	LSECSSC	LSE CSS中断清除 0: 无影响 1: 将LSECSSF标志清0
8	保留	必须保持复位值
7:6	保留	必须保持复位值
5	PLLRDYC	PLL时钟稳定中断清零 0: 无影响 1: 将PLLRDYF标志清0
4	HSERDYC	HSE时钟稳定中断清零 0: 无影响 1: 将HSERDYF标志清0
3	HSIRDYC	HSI时钟稳定中断清零 0: 无影响 1: 将HSIRDYF标志清0
2	MSIRDYC	MSI时钟就绪中断清零 0: 无影响 1: 将MSIRDYF标志清0
1	LSERDYC	LSE时钟稳定中断清零 0: 无影响 1: 将LSERDYF标志清0

- 0 LSIRDYC LSI时钟稳定中断清零
 0: 无影响
 1: 将LSIRDYF标志清0

6.4.8 GPIO 端口复位寄存器 (RCC_IOPRSTR)

偏移地址: 0x24

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

注意: GPIO 端口复位软件流程为: 相应 bit 位置 1, 复位 GPIO 端口; 清除该位, 使其 GPIO 端口处于正常工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值
3	GPIODRST	GPIOD复位 0: 正常工作模式 1: 复位GPIOD
2	GPIOCRST	GPIOC复位 0: 正常工作模式 1: 复位GPIOC
1	GPIOBRST	GPIOB复位 0: 正常工作模式 1: 复位GPIOB
0	GPIOARST	GPIOA复位 0: 正常工作模式 1: 复位GPIOA

6.4.9 AHB 外设复位寄存器 (RCC_AHBRSTR)

偏移地址: 0x28

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

注意： 所有外设端口复位软件流程为：相应bit 位置1，复位外设端口；清除该位，使其外设端口处于正常工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDIV RST.	PLA RST	Res.	CRC RST	Res.											DMA1 RST
rw	rw		rw							r	r	r	r	r	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	HDIVRST	HDIV复位 0: 正常工作模式 1: 复位HDIV
14	PLARST	PLA复位 0: 正常工作模式 1: 复位PLA模块
13	保留	必须保持复位值
12	CRCRST	CRC复位 0: 正常工作模式 1: 复位CRC
11:1	保留	必须保持复位值
0	DMA1RST	DMA1复位 0: 正常工作模式 1: 复位DMA1

6.4.10 APB1 外设复位寄存器 (RCC_APB1RSTR1)

偏移地址：0x2C

复位值：0x0000 0000

访问：无等待，字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LPTIM1 RST	Res.	SAC RST	PWR RST	Res.						I2C2 RST	I2C1 RST	LPUART 1RST	Res.	USART3 RST	USART2 RST	Res.
rw		rw	rw						rw	rw	rw		rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ATK RST	SPI2 RST							LCD RST	TIM7 RST	TIM6 RST	TIM5 RST	Res.	Res.	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw						rw	rw	rw	r	rw		rw	rw	rw	

位/位域	名称	描述
31	LPTIM1RST	LPTIM1复位 0: 正常工作模式 1: 复位LPTIM1
30	保留	必须保持复位值
29	SACRST	SAC寄存器复位 0: 正常工作模式 1: 复位SAC模块
28	PWRRST	PWR寄存器复位 0: 正常工作模式 1: 复位PWR模块
27:23	保留	必须保持复位值
22	I2C2RST	I2C2复位 0: 正常工作模式 1: 复位I2C2
21	I2C1RST	I2C1复位 0: 正常工作模式 1: 复位I2C1
20	LPUART1RST	LPUART1复位 0: 正常工作模式 1: 复位LPUART1
19	保留	必须保持复位值
18	USART3RST	USART3复位 0: 正常工作模式 1: 复位USART3
17	USART2RST	USART2复位 0: 正常工作模式 1: 复位USART2
16	保留	必须保持复位值
15	ATKRST	ATK复位 0: 正常工作模式 1: 复位ATK
14	SPI2RST	SPI2复位

		0: 正常工作模式 1: 复位SPI2
13:9	保留	必须保持复位值
8	LCDRST	LCD复位 0: 正常工作模式 1: 复位LCD
7	TIM7RST	TIM7复位 0: 正常工作模式 1: 复位TIM7
6	TIM6RST	TIM6复位 0: 正常工作模式 1: 复位TIM6
5	TIM5RST	TIM5复位 0: 正常工作模式 1: 复位TIM5
4:3	保留	必须保持复位值
2	TIM4RST	TIM4复位 0: 正常工作模式 1: 复位TIM4
1	TIM3RST	TIM3复位 0: 正常工作模式 1: 复位TIM3
0	TIM2RST	TIM2复位 0: 正常工作模式 1: 复位TIM2

6.4.11 APB2 外设复位寄存器 (RCC_APBSTR2)

偏移地址: 0x30

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.				DBG RST	Res.							ADC1 RST	Res.			
				rw								rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

Res.	USART1 RST	Res.	SPI1 RST	TIM1 RST	Res.				Res.	Res.			SYSCFG RST
	rw		rw	rw									rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	DBGRST	调试接口复位 0: 正常工作模式 1: 复位调试接口
26:21	保留	必须保持复位值
20	ADC1RST	ADC1复位 0: 正常工作模式 1: 复位ADC1
19:15	保留	必须保持复位值
14	USART1RST	USART1复位 0: 正常工作模式 1: 复位USART1
13	保留	必须保持复位值
12	SPI1RST	SPI复位 0: 正常工作模式 1: 复位SPI
11	TIM1RST	TIM1复位 0: 正常工作模式 1: 复位TIM1
10:1	保留	必须保持复位值
0	SYSCFGRST	SYSCFG复位 0: 正常工作模式 1: 复位SYSCFG

6.4.12 I/O 端口时钟使能寄存器 (RCC_IOPENR)

偏移地址: 0x34

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。



Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值
3	GPIODEN	GPIOD时钟使能 0: 禁止 1: 使能
2	GPIOCEN	GPIOC时钟使能 0: 禁止 1: 使能
1	GPIOBEN	GPIOB时钟使能 0: 禁止 1: 使能
0	GPIOAEN	GPIOA时钟使能 0: 禁止 1: 使能

6.4.13 AHB 外设时钟使能寄存器 (RCC_AHBENR)

偏移地址: 0x38

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDIV EN.	PLA EN	Res.	CRC EN	Res.											DMA1 EN
rw	rw		rw												rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	HDIVEN	HDIV时钟使能 0: 禁止 1: 使能
14	PLAEN	PLA时钟使能

		0: 禁止 1: 使能
13	保留	必须保持复位值
12	CRCEN	CRC时钟使能 0: 禁止 1: 使能
11:1	保留	必须保持复位值
0	DMA1EN	DMA1时钟使能 0: 禁止 1: 使能

6.4.14 APB1 外设时钟使能寄存器 (RCC_APBENR1)

偏移地址: 0x3C

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 EN	Res.	SAC EN	PWR EN	Res.				I2C2 EN	I2C1 EN	LPUART1 EN	Res	USART3 EN	USART2 EN	Res.	
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
ATK EN	SPI2 EN	Res.		WWDG EN	Res	Res.	LCD EN	TIM7 EN	TIM6 EN	TIM5 EN	Res.		TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31	LPTIM1EN	LPTIM时钟使能 0: 禁止 1: 使能
30	保留	必须保持复位值
29	SACEN	SAC时钟使能 0: 禁止 1: 使能
28	PWREN	PWR时钟使能 0: 禁止 1: 使能
27:23	保留	必须保持复位值

22	I2C2EN	I2C2时钟使能 0: 禁止 1: 使能
21	I2C1EN	I2C1时钟使能 0: 禁止 1: 使能
20	LPUART1EN	LPUART1时钟使能 0: 禁止 1: 使能
19	保留	必须保持复位值
18	USART3EN	USART3时钟使能 0: 禁止 1: 使能
17	USART2EN	USART2时钟使能 0: 禁止 1: 使能
16	保留	必须保持复位值
15	ATKEN	ATK时钟使能 0: 禁止 1: 使能
14	SPI2EN	SPI2时钟使能 0: 禁止 1: 使能
13: 12	保留	必须保持复位值
11	WWDGEN	WWDG时钟使能 0: 禁止 1: 使能
10:9	保留	必须保持复位值
8	LCDEN	LCD时钟使能 0: 禁止 1: 使能
7	TIM7EN	TIM7时钟使能

		0: 禁止 1: 使能
6	TIM6EN	TIM6时钟使能 0: 禁止 1: 使能
5	TIM5EN	TIM5时钟使能 0: 禁止 1: 使能
4:3	保留	必须保持复位值
2	TIM4EN	TIM4时钟使能 0: 禁止 1: 使能
1	TIM3EN	TIM3时钟使能 0: 禁止 1: 使能
0	TIM2EN	TIM2时钟使能 0: 禁止 1: 使能

6.4.15 APB2 外设时钟使能寄存器 (RCC_APBENR2)

偏移地址: 0x40

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.				DBG EN	Res.							ADC1 EN	Res.			
				rw								rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	USART1 EN	Res.	SPI1 EN	TIM1 EN	Res.						Res.	Res.		SYSCFG EN		
	rw		rw	rw										rw		

位/位域	名称	描述
31:28	保留	必须保持复位值
27	DBGEN	调试接口时钟使能 0: 禁止 1: 使能

26:21	保留	必须保持复位值
20	ADC1EN	ADC1时钟使能 0: 禁止 1: 使能
19:15	保留	必须保持复位值
14	USART1EN	USART1时钟使能 0: 禁止 1: 使能
13	保留	必须保持复位值
12	SPI1EN	SPI1时钟使能 0: 禁止 1: 使能
11	TIM1EN	TIM1时钟使能 0: 禁止 1: 使能
10:1	保留	必须保持复位值
0	SYSCFGEN	SYSCFG时钟使能 0: 禁止 1: 使能

6.4.16 外设异步时钟配置寄存器 (RCC_CCIPR)

偏移地址: 0x54

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC1SEL[1:0]		Res.						TIM1SEL	Res.	LPTIM1SEL[2:0]			Res.		
rw	rw							rw		rw	rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		I2C1SEL[1:0]		LPUART1SEL[1:0]		Res.						USART1SEL [1:0]			
		rw	rw	rw	rw									rw	rw

位/位域	名称	描述
31:30	ADC1SEL[1:0]	ADC1时钟源选择 00: SYSCLK 01: SYSCLK 2分频

		10: SYSCLK 4分频
		11: HSI
29:23	保留	必须保持复位值
22	TIM1SEL[1:0]	TIM1时钟源选择 0: PCLK2 1: PLLQCLK
21	保留	必须保持复位值
20:18	LPTIM1SEL[2:0]	LPTIM1时钟源选择 000: PCLK1 001: LSI 010: HSI 011: LSE 1xx: 保留
17:14	保留	必须保持复位值
13:12	I2C1SEL[1:0]	I2C1时钟源选择 00: PCLK1 01: SYSCLK 10: HSI 11: 保留
11:10	LPUART1SEL[1:0]	LPUART1时钟源选择 00: PCLK1 01: SYSCLK 10: HSI 11: LSE
9:2	保留	必须保持复位值
1:0	USART1SEL[1:0]	USART1时钟源选择 00: PCLK2 01: SYSCLK 10: HSI 11: LSE

6.4.17 控制寄存器 (RCC_BDCR)

偏移地址: 0x5C

复位值: 0x0000 0000

访问：无等待，字、半字、字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						LSCO SEL	LSCO EN	Res.						BD RST	
rw						rw	rw	rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	RTC LPW	RTCFAD			RTCSEL[2:0]				LSE CSSD	LSE_ CSSON	LSE DRV[1:0]		LSE RDY	LSE BYP	LSE ON
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	r	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25	LSCOSEL	低速时钟输出选择 0: 选择LSI时钟 1: 选择LSE时钟
24	LSCEN	低速时钟输出使能 0: 禁止 1: 使能
23:17	保留	必须保持复位值
16	BDRST	RTC模块复位使能 0: 禁止 1: 使能
15	RTCEN	RTC时钟使能 0: 禁止 1: 使能
14	RTCLPW	RTC模块低功耗控制 0: 低功耗模式禁止 1: 低功耗模式使能 使能后，RTC模块进低功耗状态后，其寄存器不可读写
13:11	RTCFAD	RTC高速时钟校准频率选择 000: 4M; 001: 6M; 010: 8M; 011: 12M; 100: 16M; 101: 20M; 110: 24M; 111: 32M
10:8	RTCSEL[2:0]	RTC时钟源选择 000: 无时钟 001: 保留 010: LSE时钟 011: LSI时钟 100: HSE的128分频时钟 101: HSE的256分频时钟

		110: HSE的512分频时钟
		111: HSE的1024分频时钟
		<i>注意: 选择RTC时钟源后, 仅在以下情况下可更改时钟源:</i>
		- <i>检测到LSE CSS故障</i>
		<i>当RTC使用LSI或LSE时钟源时, 需要在开启RTC模块时钟后延时至少40us, 才能配置RTC模块内的寄存器</i>
7	保留	必须保持复位值
6	LSECSSD	LSE CSS故障检测状态指示 0: 未检测到故障 1: 检测到故障 <i>注意: 当LSECSSON位清0时, 该位也被清0。</i>
5	LSECSSON	LSE CSS使能 0: 禁止 1: 使能 <i>注意: 该位一旦使能便不能更改。</i>
4:3	LSEDRV[1:0]	LSE驱动能力 00: 160nA 01: 280nA 10: 415nA 11: 530nA
2	LSERDY	LSE时钟稳定标志 0: LSE未稳定 1: LSE已稳定
1	LSEBYP	LSE旁路模式 0: 外部晶体模式 1: 旁路模式 <i>注意, 只有LSEON为1, 该位配置才有效</i>
0	LSEON	LSE使能 0: 禁止 1: 使能

6.4.18 时钟控制/状态寄存器 (RCC_CSR)

偏移地址: 0x60

复位值: 0xXXX0 0000

访问: 0~3 个等待周期, 字、半字、字节访问。

此寄存器可通过POR/PDR、BOR复位；系统复位仅可复位bit0到bit16。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	WWDG RSTF	IWWG RSTF	SFT RSTF	POR RSTF	PIN RSTF	OBL RSTF	LOCKUP RSTF	Res.	BOR RSTF	Res					RMVF
15	r	r	r	r	r	r	r	r	r	5	4	3	2	1	0
Res.							LOCKUP RSTEN	Res					LSI RDY	LSI ON	
							rw						r	rw	

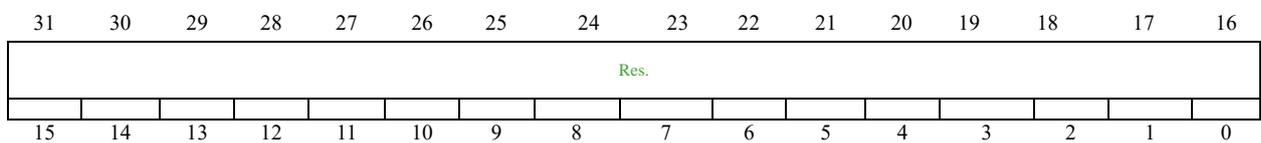
位/位域	名称	描述
31	保留	必须保持复位值
30	WWDGRSTF	窗口看门狗复位标志 窗口看门狗复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生窗口看门狗复位 1：产生窗口看门狗复位
29	IWDGRSTF	独立窗口看门狗复位标志 独立看门狗复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生独立看门狗复位 1：产生独立看门狗复位
28	SFTRSTF	软件复位标志 软件复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生软件复位 1：产生软件复位
27	PORRSTF	POR/PDR复位标志 POR/PDR复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生POR/PDR复位 1：产生POR/PDR复位
26	PINRSTF	NRST引脚复位标志 NRST引脚复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生来自NRST引脚的复位 1：发产生自NRST引脚的复位
25	OBLRSTF	选项字节加载复位标志 选项字节加载复位时，该位被置1。将RMVF位置1可清除该位。 0：未产生来自选项字节加载的复位 1：产生来自选项字节加载的复位
24	LOCKUPRSTF	LOCKUP复位标志 当LOCKUP功能使能（LOCKUPRSTEN置1），且产生LOCKUP故障时，该位被置1。将RMVF位置1可清除该位。

		0: 未产生LOCKUP的复位 1: 产生LOCKUP的复位
23	保留	必须保持复位值
22	BORRSTF	BOR复位标志 BOR复位时, 该位被置1。将RMVF位置1可清除该位。 0: 未产生BOR复位 1: 产生BOR复位
21:17	保留	必须保持复位值
16	RMVF	清除复位标志 由软件置1, 用于将复位标志清0, 同时该位也被清0, 读为0。 0: 无影响 1: 清除复位标志
15:9	保留	必须保持复位值
8	LOCKUPRSTEN	LOCKUP复位使能 0: 禁止 1: 使能
7:2	保留	必须保持复位值
1	LSIRDY	LSI稳定标志 当LSION位被清0后, LSIRDY将在3个LSI时钟周期后自动清0。 0: LSI未稳定 1: LSI已稳定
0	LSION	LSI使能 0: 禁止LSI 1: 使能LSI <i>注意: 如果是LSI作为独立看门狗或RTC时钟源时, 即使LSION未使能 (LSION为0), LSIRDY也会被自动置1 (此时LSI默认使能, 但对应的LSION仍为0)。</i>

6.4.19 HSI Trim 寄存器 (RCC_HSI TRIMR)

偏移地址: 0x100

复位值: 0x0000 0XXX, 初始值从 NVR 中加载(加载前为 0x140)



Res.						HSIT[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9:0	HSIT	HSI 时钟调校准值，默认值为 0x140。

6.4.20 MSI Trim 寄存器 (RCC_MSITRIMR)

偏移地址：0x104

复位值：0x00XX XXXX，初始值从 Flash 加载(加载前为 0x140)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						MSIT [9:0]									
						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
Res.											LSIT [5:0]				
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25:16	MSIT[9:0]	MSI时钟调校准值，默认值为0x140。
15:6	保留	必须保持复位值
5:0	LSIT[5:0]	LSI 时钟调校准值，默认值为 0x20。

6.4.21 BGR Trim 寄存器 (RCC_BGRTRIMR)

偏移地址：0x108

复位值：0x0000 00XX，初始值从 Flash 加载

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											BGR [4:0]				
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值
4:0	BGR[5:0]	BGR Trim 值

加载前上电默认值为：0x10。

6.4.22 DAC Trim 寄存器 (RCC_DACTRIMR)

偏移地址：0x10C

复位值：0x XX XX XX XX，初始值从 Flash 加载

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			DAC4 [4:0]					Res.			DAC3 [4:0]				
			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
Res.			DAC2 [4:0]					PLL_PED		PLL_LT	DAC1 [4:0]				
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28:24	DAC4	DAC4BUF Trim 值 加载前上电默认值为：0x10。
23:21	保留	必须保持复位值
20:16	DAC3	DAC3BUF Trim 值 加载前上电默认值为：0x10。
15:13	保留	必须保持复位值
12:8	DAC2	DAC2BUF Trim 值 加载前上电默认值为：0x10。
7:6	PLL_PED	PLL相位错误检测选择 00: 1ns 01: 4ns(默认) 10: 7ns 11: 10ns
5	PLL_LT	PLL LOCK时钟周期选择 0: 16个时钟 1: 32个时钟
4:0	DAC1	DAC1BUF Trim 值 加载前上电默认值为：0x10。

6.4.23 OPAMP Trim 寄存器 (RCC_OPATRIMR)

偏移地址: 0x110

复位值: 0x XX XX XX XX, 初始值从 Flash 加载

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			OPA4 [4:0]					Res.			OPA3 [4:0]				
			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
Res.			OPA2 [4:0]					Res.			OPA1 [4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值
28:24	OPA4	OPAMP4 Trim 值 加载前上电默认值为: 0x10。
23:21	保留	必须保持复位值
20:16	OPA3	OPAMP3 Trim 值 加载前上电默认值为: 0x10。
15:13	保留	必须保持复位值
12:8	OPA2	OPAMP2 Trim 值 加载前上电默认值为: 0x10。
7:5	保留	必须保持复位值
4:0	OPA1	OPAMP1 Trim 值 加载前上电默认值为: 0x10。

7 通用输入/输出接口 (GPIO)

7.1 简介

每个通用 I/O 端口包括 3 个 32 位配置寄存器 (GPIOx_MODER、GPIOx_OTYPER、GPIOx_PUPDR)，2 个 32 位数据寄存器 (GPIOx_IDR、GPIOx_ODR)，1 个 32 位位置位/复位寄存器 (GPIOx_BSRR)、1 个 32 位复位寄存器 (GPIOx_BRR)；所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器 (GPIOx_AFRH、GPIOx_AFRL)。

7.2 GPIO 主要特性

- GPIO 模式
 - 输入
 - 输出
 - 复用功能 (每个 I/O 最多支持 8 路复用功能)
 - 模拟模式
- 上拉/下拉单独可配
- 推挽/开漏单独可配
- 引脚复用灵活，允许将 I/O 引脚用作 GPIO 或外设功能中的一种
- 置位和复位寄存器 (GPIOx_BSRR)，对 GPIOx_ODR 具有按位写权限
- 锁定机制 (GPIOx_LCKR)，可锁定 I/O 端口配置

7.3 GPIO 功能描述

软件可在配置寄存器中对每个 GPIO 的模式进行独立配置，多种可选的模式组合如下：

- 输入浮空
- 输入上拉
- 输入下拉
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出

- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏
- 模拟功能

每个 I/O 端口对应的寄存器位均可自由编程,但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。

图 7-1 描述了 I/O 电路的基本结构,表 7-1 给出了可能的 I/O 配置方案。

图 7-1 I/O 基本结构

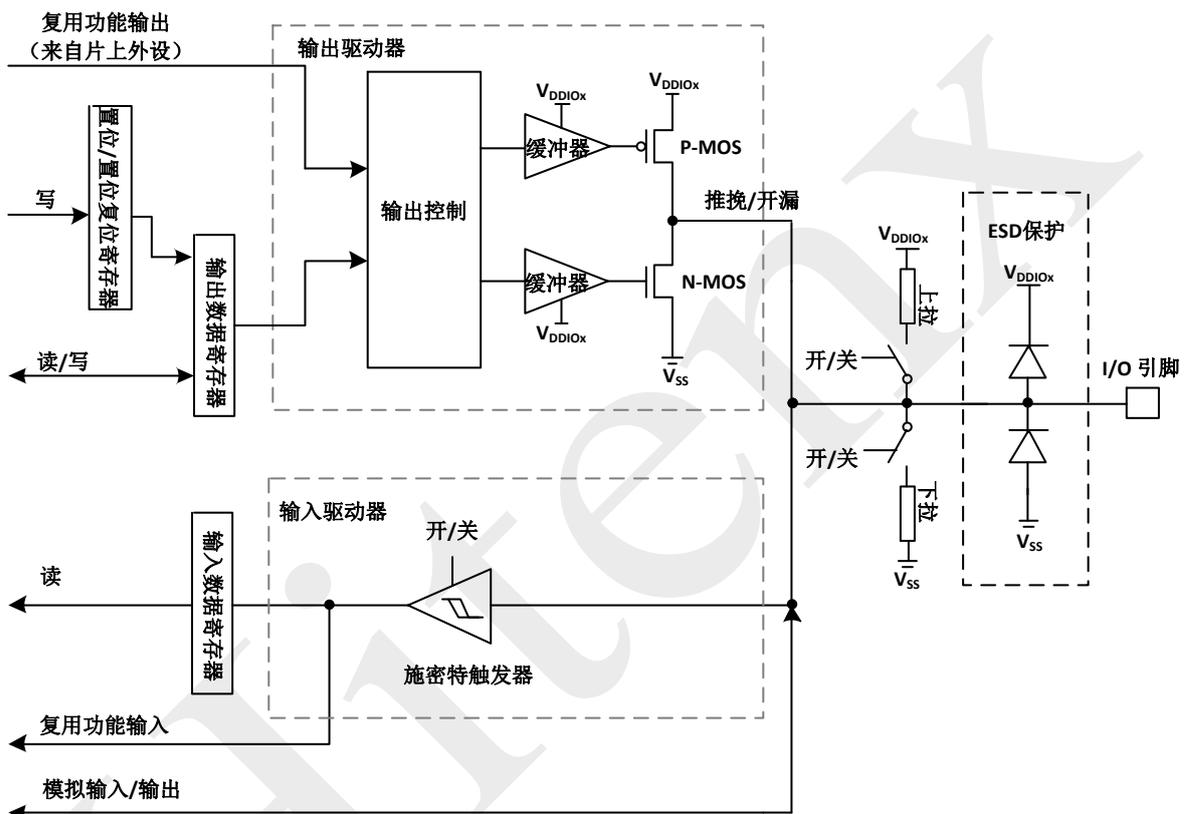


表 7-1 GPIO 端口配置表⁽¹⁾

I/O 端模式组合		GPIO 模式 MODE[1:0]	输出类型 OT	上拉/下拉 PUPD[1:0]	
输入	浮空	00	-	0	0
	上拉		-	0	1
	下拉		-	1	0
输出	推挽	01	0	0	0
	推挽+上拉		0	0	1
	推挽+下拉		0	1	0
	开漏		1	0	0
	开漏+上拉		1	0	1
	开漏+下拉		1	1	0
	推挽	10	0	0	0

I/O 端模式组合		GPIO 模式 MODE[1:0]	输出类型 OT	上拉/下拉 PUPD[1:0]	
复用	推挽+上拉		0	0	1
	推挽+下拉		0	1	0
	开漏		1	0	0
	开漏+上拉		1	0	1
	开漏+下拉		1	1	0
模拟	输入/输出	11	-	上拉/下拉禁用	
			-		
			-		

1. “-”配置无效。

7.3.1 通用 I/O (GPIO)

复位完成前，除 PA13 作为 BOOT0 引脚外，其它 I/O 端口均处于模拟模式。

复位完成后调试端口引脚处于复用功能上拉/下拉状态，PD13（NRST）数字输入模式上拉状态，其它引脚仍然处于模拟模式。

- PA13：复用功能 SWCLK 处于下拉状态
- PA12：复用功能 SWDIO 处于上拉状态

7.3.2 I/O 引脚功能复用器和映射

I/O 引脚功能包括 GPIO、外设复用功能和附加功能。

调试功能(Debug):

- 芯片复位完成后，默认调试引脚硬件配置为复用功能 0
 - 复用功能 0，即复用调试功能 SWDIO/ SWCLK（PA12/PA13）。

GPIO:

应用程序可在 I/O 模式寄存器 GPIOx_MODER 中，将 I/O 配置为输入或输出。

复用功能 (Alternate functions):

每个 I/O 电路结构中包含一个引脚复用器，可将各外设模块的复用功能映射到不同的 I/O 引脚上，实现芯片与板载外设或模块通信。这种 I/O 复用架构下，每个 I/O 引脚一次只允许映射一种复用功能，确保了不同外设的复用功能在同一个 I/O 上不会发生应用冲突。

- 芯片复位后，复用功能寄存器 GPIOx_AFRH/GPIOx_AFRL 的值默认选择复用功能 0(Alternate Function 0)
- 复用功能配置
 - 在 GPIOx_AFRH/GPIOx_AFRL 寄存器中，将 I/O 映射到所需的复

用功能；每个引脚最多支持 8 种特定的复用功能，参见表：datasheet。通过 GPIOx_OTYPER 和 GPIOx_PUPDR 寄存器，分别选择输出类型、上拉/下拉。

- 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能。

附加功能 (Additional functions):

- 模拟功能
 - 对于 ADC、SAC、LCD/LED、ATK，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC、SAC、LCD/LED 和 ATK 寄存器中使能。
- 其它功能
 - 如振荡器，在相关的 RCC 寄存器中配置所需功能。配置这些外设附加功能后，GPIO 寄存器中的对应配置无效。

振荡器(HSE/LSE): HSE_IN 和 HSE_OUT 通过 RCC 的 RCC_CR 寄存器 HSEBYP 和 HSEON 位控制 IO 实现输入和输出(当时钟源是外部晶体时,通过模拟输入、输出;当时钟源选择用户外部时钟时,通过数字输入,此时 HSE_OUT 引脚可用作 GPIO);

7.3.3 I/O 端口配置寄存器

每个通用 I/O 端口包括 3 个 32 位配置寄存器(GPIOx_MODER、GPIOx_OTYPER、GPIOx_PUPDR) 用于配置端口的每个 GPIO。

- GPIO 模式寄存器 GPIOx_MODER，用于配置 I/O 模式。
 - 输入、输出、复用、模拟模式。
- 输出类型寄存器 GPIOx_OTYPER，用于配置输出类推挽/开漏。
- 上拉/下拉寄存器 GPIOx_PUPDR 用于配置上拉/下拉。

7.3.4 I/O 端口数据寄存器

每个 I/O 端口都具有 2 个 16 位数据寄存器：输入和输出数据寄存器 (GPIOx_IDR 和 GPIOx_ODR)。

- GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。
- GPIOx_IDR 是一个只读寄存器，通过 I/O 引脚输入的数据存储到 GPIOx_IDR 中。

7.3.5 I/O 数据位操作

置位/复位寄存器 GPIOx_BSRR 允许应用程序对输出数据寄存器 GPIOx_ODR 按位操作，即对输出数据寄存器中的每个位单独执行置位和复位。

- 置位/复位寄存器中的 BS[15:0]和 BR[15:0]的 bit(i)与 GPIOx_ODR 寄存器中的 bit(i)对应，分别执行置位操作和复位操作
- 置位/复位寄存器中的任何位写 0 无效，不会改变输出数据寄存器中对应位的状态
- 同时尝试对某个位执行置位和复位操作，则置位操作优先。

应用程序可在单个 AHB 时钟周期下，通过写访问 GPIOx_BSRR 寄存器修改 1 个或多个 GPIOx_ODR 寄存器中的对应位，因此不会被中断打断，无需软件关闭中断。软件直接操作 GPIOx_ODR 寄存器，需要先读出寄存器值再写入，至少需要 2 个 AHB 时钟周期，因此修改寄存器值的过程，会被读取和写入之间产生的中断打断。

由于上述特性，通过 GPIOx_BSRR 寄存器修改 GPIOx_ODR 寄存器对应位的值是一个“单次”操作，执行过程中不会锁定 GPIOx_ODR 寄存器位，确保了软件可随时直接访问访问 GPIOx_ODR 寄存器位而不受影响。

7.3.6 GPIO 锁定机制

GPIO 锁定机制用于保护重要的 I/O 引脚配置，避免因软件误操作而更改。锁定设置完成后，GPIOx_MODER、GPIOx_OTYPER、GPIOx_PUPDR、GPIOx_AFRH 和 GPIOx_AFRL 寄存器中被锁定的 I/O 引脚对应的位锁定，软件无法更改直至芯片复位或被锁定的 I/O 端口自身复位。（参见：[I/O 端口复位寄存器 RCC_IOPRST](#)）

向锁定寄存器 GPIOx_LCKR 中写入锁定序列，可实现 I/O 端口的锁定，锁定序列 LCKx 如下：

- 1) LCK bit(s): LCKK='1'(16bit) + LCK[15:0]
- 2) LCK bit(s): LCKK='0'(16bit) + LCK[15:0]
- 3) LCK bit(s): LCKK='1'(16bit) + LCK[15:0]

其中，锁定序列第 16 位 LCKK 为锁定 Key，LCK[15:0] 用于指定被锁定的 I/O 引脚；锁定序列写入流程参见 [GPIO 端口配置锁定寄存器 GPIOx_LCKR](#)，锁定过程中以字访问方式写/读 GPIOx_LCKR 寄存器，不能更改 LCK[15:0]的值。

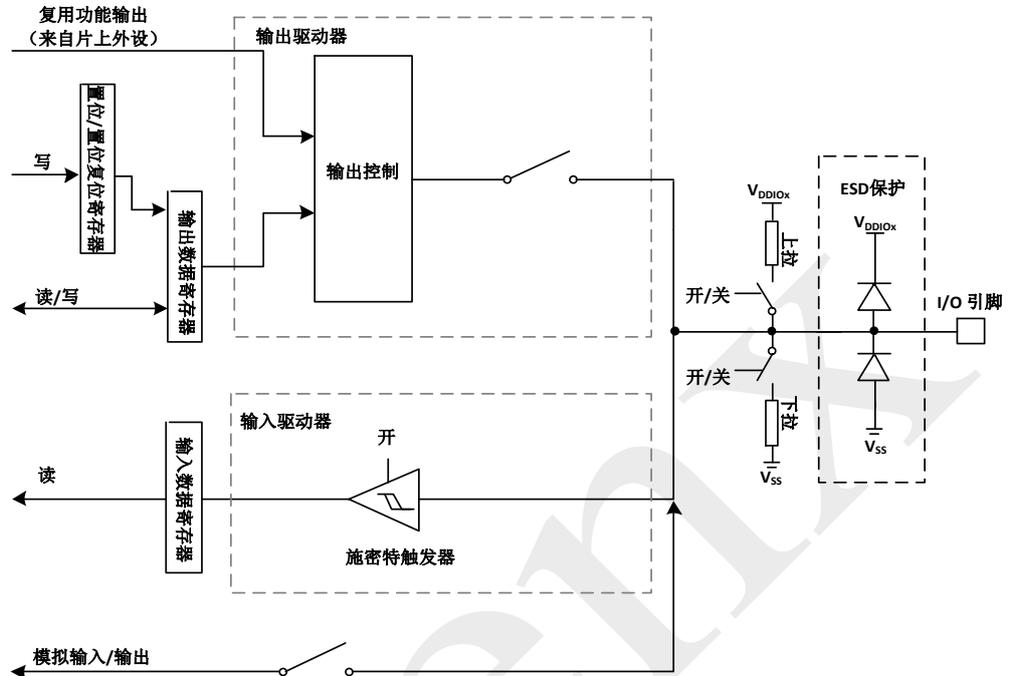
7.3.7 外部中断线/唤醒线

全部 GPIO 端口均具有外部中断功能，且由 EXTI 进行配置和管理，参见 [EXTI 控制器](#)。使用外部中断线时，为保持输入中断触发有效，应禁止将与之对应的

GPIO 用作振荡器引脚或配置为模拟模式。

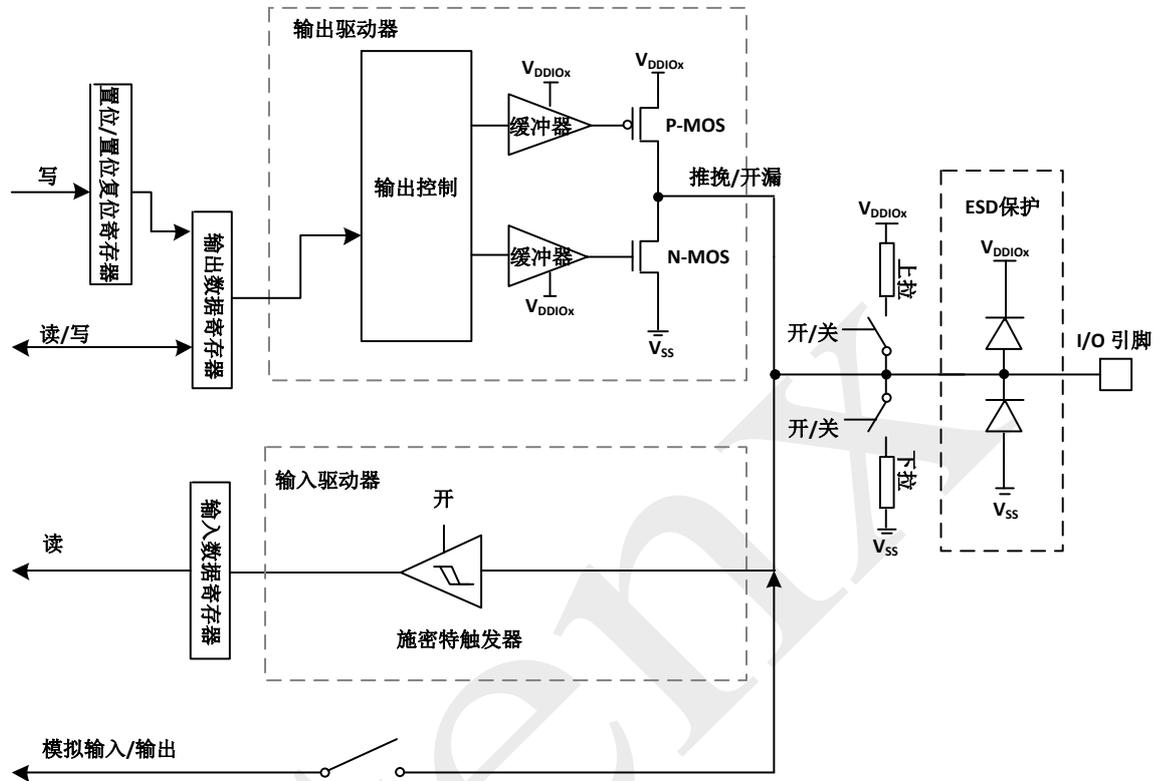
7.3.8 输入配置

图 7-2 I/O 引脚输入配置（浮空/上拉/下拉）



配置 I/O 为输入时，I/O 电路结构示意图如上图所示：

- 施密特触发器输入使能
- 上拉、下拉电阻的选择由 GPIOx_PUPDR 寄存器中的值决定
- 输出缓冲器禁用
- 每个 AHB 时钟周期采样一次 I/O 引脚上的数据，保存到输入数据寄存器
- 读访问输入数据寄存器获取 I/O 状态

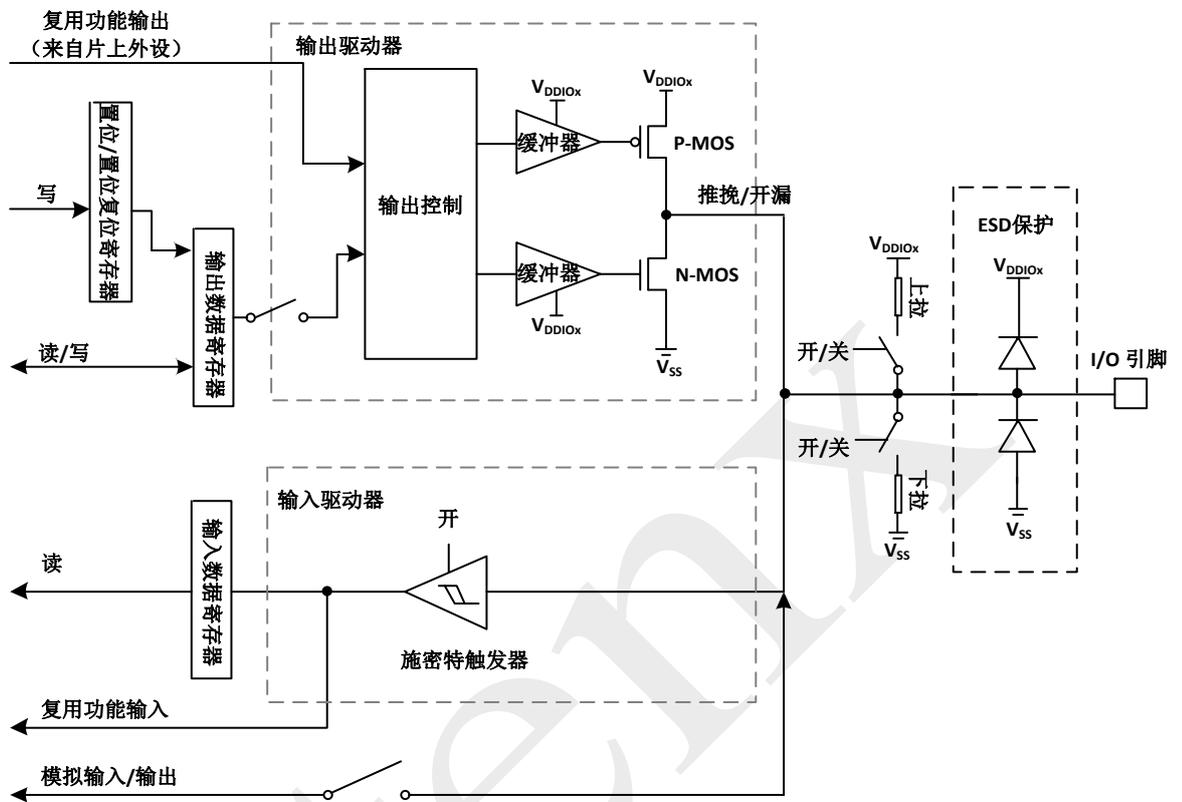
7.3.9 输出配置
图 7-3 I/O 引脚输出配置


配置 I/O 为输出时，I/O 电路结构示意图如上图所示：

- 输出缓冲器使能：
 - 开漏模式：输出数据寄存器写 0，激活 N-MOS 输出低；输出数据寄存器写 1，输出高阻态。
 - 推挽模式：输出数据寄存器写 0，激活 N-MOS 输出低；输出数据寄存器写 1，激活 P-MOS 输出高。
- 软件读访问输出数据寄存器获取最后写入值
- 施密特触发器输入使能
- 上拉、下拉电阻的选择由 GPIOx_PUPDR 寄存器的值决定
- 每个 AHB 时钟周期采样一次 I/O 引脚上的数据，保存到输入数据寄存器
- 读访问输入数据寄存器获取 I/O 状态

7.3.10 复用功能配置

图 7-4 I/O 引脚复用功能配置



配置 I/O 为复用功能时，I/O 电路结构如上图所示：

- 施密特触发器输入使能
- 上拉、下拉电阻的选择由 GPIOx_PUPDR 寄存器中的值决定
- 每个 AHB 时钟周期采样一次 I/O 引脚上的数据，保存到输入数据寄存器
- 读访问输入数据寄存器获取 I/O 状态
- 输出缓冲器使能
 - 可配置为开漏或推挽模式；
 - 输出缓冲器由外设信号驱动。（发送器使能信号和输出数据）

7.3.11 复用功能选择

用户可通过复用功能寄存器 GPIOx_AFRH 和 GPIOx_AFRL 为每个 GPIO 分配可用的复用输入/输出功能。每个 GPIO 引脚包含多个外设复用功能，即应用程序可根据外设需求，将外设复用功能连接到包含该复用功能的引脚上。外设复用功能参见 datasheet。

注意： LCD 控制器输出端子 COM 和 SEG 复用功能选择：

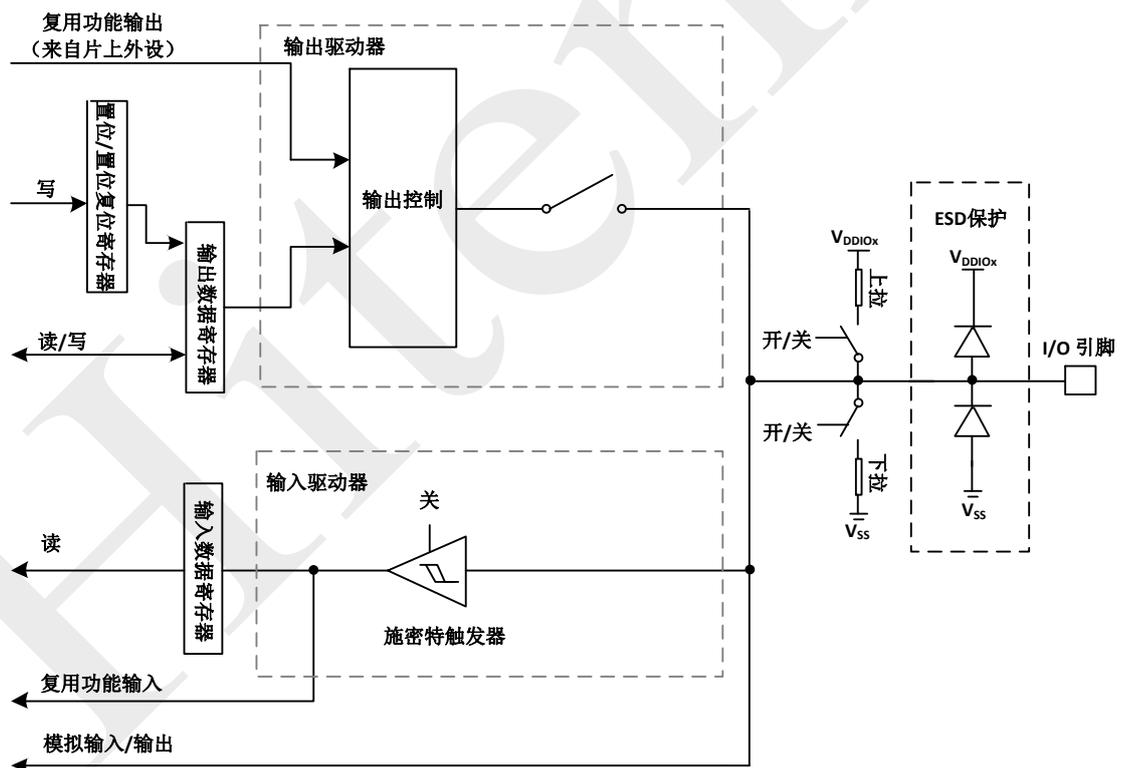
LCD 输出端子复用功能在复用功能寄存器中进行选择, 但 COM 和 SEG 为模拟端子, 对应的 I/O 电路结构参见模拟功能配置。

COM 和 SEG 端子为模拟端子作为复用功能放在复用功能寄存器中的原因如下:

- COM 和 SEG 端子数量较多, 在 LCD 控制器进行配置需要增加额外的配置寄存器
- 不同封装规模的芯片 COM 和 SEG 端子数量不同, 在 LCD 控制器中增加寄存器进行配置不利于 I/P 设计的系列化
- 在复用功能寄存器中进行配置无须增加额外的 COM 和 SEG 端子配置寄存器

7.3.12 模拟模式配置

图 7-5 I/O 引脚高阻态模拟配置



配置 I/O 为模拟模式时, I/O 电路结构如上图所示:

- 输出缓冲器禁用
- 施密特触发器输入禁用, 强制施密特触发器输出为恒定值(0)。
 - I/O 引脚的模拟输入功耗为 0。

- 上拉、下拉电阻禁用
- 读访问输入数据寄存器的值为 0

7.4 GPIO 寄存器

7.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x = A~D)

偏移地址: 0x00

复位值: 端口 A 0xFAFF FFFF

 端口 D 0xF3FF FFFF

 其它端口 0xFFFF FFFF

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw										

位/位域	名称	描述
------	----	----

31:0	MODEy [1:0]:	Portx端口I/O引脚y的I/O模式配置。(x=A~D; y = 0 ~15)
------	--------------	--

00: 输入模式

01: 通用输出模式

10: 复用功能模式

11: 模拟模式 (复位状态)

注意: 复位状态下, PA12和PA13为复用功能模式。

7.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A~D)

偏移地址: 0x04

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值
-------	----	---------

15:0	OTy[15:0]	Portx端口I/O引脚y的输出类型配置。(x=A~D; y = 0 ~15)
------	-----------	---

- 0: 推挽输出（复位状态）
- 1: 开漏输出

7.4.3 GPIO 端口上拉/下拉寄存器（GPIOx_PUPDR）（x = A~D）

偏移地址：0x0C

复位值：端口 A 0x0900 0000
 端口 D 0x0400 0000
 其它端口 0x0000 0000

访问：无等待，字、半字、字节访问

31		30		29		28		27		26		25		24		23		22		21		20		19		18		17		16	
PUPD15[1:0]				PUPD 14[1:0]				PUPD13[1:0]				PUPD12[1:0]				PUPD11[1:0]				PUPD10[1:0]				PUPD9[1:0]				PUPD8[1:0]			
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	
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
PUPD7[1:0]				PUPD6[1:0]				PUPD5[1:0]				PUPD4[1:0]				PUPD3[1:0]				PUPD2[1:0]				PUPD1[1:0]				PUPD0[1:0]			
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	

位/位域	名称	描述
31:0	PUPDy [1:0]	Portx端口I/O引脚y的上/下拉配置。（x=A~D； y = 0~15） 00: 无上拉或下拉 01: 上拉 10: 下拉 11: 保留 <i>注意：复位状态下，PA12为上拉；PA13为下拉</i>

7.4.4 GPIO 端口输入寄存器（GPIOx_IDR）（x = A~D）

偏移地址：0x10

复位值： 0x0000 XXXX

访问：无等待，字、半字、字节访问（可选）

31		30		29		28		27		26		25		24		23		22		21		20		19		18		17		16	
Res.																															
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0																
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r																

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	IDy[15:0]	Portx端口I/O引脚y的输入数据。（x=A~D； y = 0~15） 这些位为只读。它们包含相应 I/O 端口的输入值。

7.4.5 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A~D)

偏移地址: 0x14

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ODy	Portx端口I/O引脚y的输出数据。(x=A~D; y=0~15) 这些位可通过软件读取和写入。 <i>注意: 对于原子置位/复位, 通过写入GPIOx_BSRR 或 GPIOx_BRR 寄存器, 可分别置位和/或复位ODx位。</i>

7.4.6 GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A~D)

偏移地址: 0x18

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	BRy	Portx端口I/O引脚y的复位配置。(x=A~D; y=0~15) 这些位为只写。若读取这些位则返回 0x0000。 0: 不会对相应的ODx位执行任何操作 1: 复位相应的ODx位 <i>注意: 如果同时对BSx和BRx置位, 则BSx的优先级更高。</i>
15:0	BSy	Portx端口I/O引脚y的置位配置。(x=A~D; y=0~15) 这些位为只写。若读取这些位则返回 0x0000。 0: 不会对相应的 ODx 位执行任何操作 1: 将相应的 ODx 位置 1

7.4.7 GPIO 端口配置锁定寄存器 (GPIO_x_LCKR) (x = A~D)

偏移地址: 0x1C

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

注意: 在对 GPIO_x_LOCK 寄存器执行锁定序列写/读期间, 只允许使用字访问 (32 位长)。

每个锁定位冻结一个特定的配置寄存器 (控制寄存器和复用功能寄存器)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值
16	LCKK	<p>锁定Key</p> <p>可随时读取此位。可使用锁定Key写序列对其进行修改。</p> <p>0: 端口配置锁定Key未激活</p> <p>1: 端口配置锁定Key已激活。直至下一次MCU 复位或GPIO复位前, GPIO_x_LCKR寄存器处于锁定状态。</p> <p>锁定Key写/读序列:</p> <p>写 LCK[16] = '1' + LCK[15:0]</p> <p>写 LCK[16] = '0' + LCK[15:0]</p> <p>写 LCK[16] = '1' + LCK[15:0]</p> <p>读 LCK</p> <p>读 LCK[16] = '1' (此读操作作为可选操作, 但它可确认锁定已激活)</p> <p><i>注意: 在锁定Key写序列期间, 不能更改LCK[15:0]的值。</i></p> <p><i>锁定序列中的任何错误都将中止锁定操作。</i></p> <p><i>在任一端口的第一个锁定序列之后, 对 LCKK 位的任何读访问都将返回“1”, 直到下一次 MCU 复位或GPIO复位为止。</i></p>
15:0	LCKy	<p>Portx端口I/O引脚 (x=A~D; y = 0~15)</p> <p>这些位都是读/写位, 但只能在 LCKK 位等于“0”时执行写操作。</p> <p>0: 端口配置未锁定</p> <p>1: 端口配置已锁定</p> <p>当执行正确的写/读序列时, 此位域的值用于指定被锁定的端口, 且在写序列期间不应更改; 被锁定的端口配置 (寄存</p>

器), 在芯片下次复位前, 将无法修改。

7.4.8 GPIO 复用功能低位寄存器 (GPIO_x_AFRL) (x = A~D)

偏移地址: 0x20

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:0	AFSELy [3:0]	Portx端口I/O引脚y的复用功能选择。(x=A~D; y = 0~7) AFSELy 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7

7.4.9 GPIO 复用功能高位寄存器 (GPIO_x_AFRH) (x = A~D)

偏移地址: 0x24

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:0	AFSELY [3:0]	Portx端口I/O引脚y的复用功能选择。(x=A~D; y=8~15) AFSELY 选择: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7

7.4.10 GPIO 端口位复位寄存器 (GPIOx_BRR) (x = A~D)

偏移地址: 0x28

复位值: 0x0000 0000

访问: 无等待, 字、半字、字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BRy	Portx端口I/O引脚位复位。(x=A~D; y=0~15) 这些位为只写。 读取这些位可返回值0x0000 0: 不会对相应的ODx位执行任何操作 1: 复位相应的ODx位

8 系统配置控制器 (SYSCFG)

8.1 简介

系统配置控制器，主要特性及应用功能如下：

- 配置红外 IRTIM 调制信号和输出极性
- TIM1 内部断路输入源使能控制
- 重映射存储器

8.2 系统配置寄存器

8.2.1 系统配置寄存器 1 (SYSCFG_CFGR1)

偏移地址：0x00

复位值：0x0000 00xx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.											IR_MOD[1:0]		IR_POL		MEM_MODE	
											rw	rw	rw	rw	rw	

位/位域	名称	描述
31:5	保留	必须保持复位值。
4:3	IR_MOD[1:0]	IRTIM调制信号选择：（参见： 红外控制模块 (IRTIM) ） 00: TIM4 01: USART2 10: USART3 11: 保留
2	IR_POL	IR_OUT输出信号极性选择 0: IRTIM (IR_OUT) 输出信号未反相 1: IRTIM (IR_OUT) 输出信号反相
1:0	MEM_MODE[1:0]	存储器映射选择位 此位域用于选择物理重映射，可旁路硬件启动选择，使相应的存储器映射到地址0x0000 0000。复位后，该位域根据启动配置进行地址映射，参见表： 启动配置 。 0x: 将User Flash映射到地址0x0000 0000 1x: 将System memory映射到地址0x0000 0000

8.2.2 系统配置寄存器 2 (SYSCFG_CFGR2)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													PVD_LOCK	Res.	LOCKUP_LOCK
													rw		rw

位/位域	名称	描述
31:3	保留	必须保持复位值
2	PVD_LOCK	PVD 锁定使能位 软件置1, 仅通过系统复位清0。 用于使能, 并作为TIM1断路输入源; 也可用于PWR_CR2寄存器写保护使能控制。 0: PVD中断与TIM1断路输入断开; PWR_CR2为可读可写 1: PVD 中断与 TIM1 断路输入连接; PWR_CR2 为只读
1	保留	必须保持复位值
0	LOCKUP_LOCK	Cortex®-M0+ LOCKUP输出使能位 软件置1, 仅通过系统复位清0。 用于使能, 并与TIM1断路输入的Cortex®-M0+ LOCKUP (HardFault) 连接。 0: Cortex®-M0+ LOCKUP (HardFault) 输出与TIM1断路输入断开 1: Cortex®-M0+ LOCKUP (HardFault) 输出与TIM1断路输入连接

9 外设互联 (Interconnect Matrix)

9.1 简介

为了减少软件操作的延迟和节省 CPU 资源，从而降低功耗，多个外设的内部直接连接。

互联功能可在多种功耗模式下工作，由使用到的外设决定，参见各个外设章节。

9.2 外设互联概述

表 9-1 互联外设

	TIM1	TIM2	TIM3	TIM4	TIM5	LPTIM	MCO	LSCO	IRTIM	ADC	DMA	DMAMUX	SAC	PLA
TIM1	×	9.3.1	9.3.1	9.3.1	9.3.1	×	×	×	×	9.3.10	×	×	9.3.11	5
TIM2	9.3.1	×	9.3.1	9.3.1	9.3.1	×	×	×	×	9.3.10	×	×	9.3.11	5
TIM3	9.3.1	9.3.1	×	9.3.1	9.3.1	×	×	×	9.3.9	×	×	×	9.3.11	5
TIM4	9.3.1	9.3.1	9.3.1	×	9.3.1	×	×	×	9.3.9	9.3.10	×	9.3.13	9.3.4	5
TIM5	9.3.1	9.3.1	9.3.1	9.3.1	×	×	×	×	×	×	×	×	9.3.11	×
TIM6	×	×	×	9.3.10	×	×	×	×	×	9.3.10	×	×	9.3.11	5
TIM7	×	×	×	×	×	×	×	×	×	×	×	×	9.3.11	5
LPTIM	×	×	×	×	×	×	×	×	×	×	×	9.3.13	9.3.11	5
USART1	×	×	×	×	×	×	×	×	×	×	×	×	×	5
USART2	×	×	×	×	×	×	×	×	9.3.9	×	×	×	×	5
USART3	×	×	×	×	×	×	×	×	9.3.9	×	×	×	×	5
SPI1	×	×	×	×	×	×	×	×	×	×	×	×	×	5
SPI2	×	×	×	×	×	×	×	×	×	×	×	×	×	5
ADC	9.3.2	×	×	×	×	×	×	×	×	×	×	×	×	5
TS	×	×	×	×	×	×	×	×	×	9.3.10	×	×	×	×
BGR	×	×	×	×	×	×	×	×	×	9.3.10	×	×	9.3.14	×
HSE	×	×	×	×	×	×	9.3.7	×	×	×	×	×	×	×
LSE	×	×	9.3.3	9.3.5	×	×	9.3.7	9.3.8	×	×	×	×	×	×
HSI	×	×	×	9.3.5	×	×	9.3.7	×	×	×	×	×	×	×
LSI	×	×	×	9.3.5	×	×	9.3.7	9.3.8	×	×	×	×	×	×
MSI	×	×	×	9.3.5	×	×	9.3.7	×	×	×	×	×	×	×
PLL	×	×	×	×	×	×	9.3.7	×	×	×	×	×	×	×
MCO	×	×	×	9.3.5	×	×	×	×	×	×	×	×	×	×

	TIM1	TIM2	TIM3	TIM4	TIM5	LPTIM	MCO	LSCO	IRTIM	ADC	DMA	DMAMUX	SAC	PLA
SYSCCLK	×	×	×	×	×	×	9.3.7	×	×	×	×	×	×	5
EXTI	×	×	×	×	×	×	×	×	×	9.3.10	×	9.3.13	9.3.11	×
RTC	×	×	×	9.3.5	×	9.3.6	×	×	×	×	×	×	×	5
SAC	9.3.2	×	9.3.3	9.3.4	×	9.3.6	×	×	×	×	×	×	×	5
SYSEERR ⁽²⁾	9.3.2	×	×	×	×	×	×	×	×	×	×	×	×	×
DMAMUX	×	×	×	×	×	×	×	×	×	×	9.3.12	×	×	×
PVD	9.3.2	×	×	×	×	×	×	×	×	×	×	×	×	×
CSS	9.3.2	×	×	×	×	×	×	×	×	×	×	×	×	×
LCD	×	×	×	×	×	×	×	×	×	×	×	9.3.13	×	×
PLA	×	×	×	9.3.10	×	×	×	×	×	9.3.10	×	×	9.3.11	×

1. ×-表示无互联

2. SYSEERR: Cortex® -M0+ LOCKUP

9.3 外设互联功能描述

9.3.1 从 TIM1/2/3/4/5 到 TIM1/2/3/4/5

TIM1/2/3/4/5 为了实现定时器的级联和同步功能，将 TIMx_TRGO 信号输出给其他定时器，用作同步或者触发信号；定时器支持多了同步或者触发信号，通过 ITRx 来选择使用，ITRx 与其他定时器的 TIMx_TRGO 连接；参见各个定时器的 [定时器同步](#) (TIM1) 章节。

9.3.2 从 ADC、SAC、CSS、PVD、SYSEERR 到 TIM1

输入到 TIM1 的信号连接有如下三种：

- 外部触发输入信号 (ETR) 参见 [外部触发输入](#)：
 - ADC 的看门狗输出，参见 [ADC_AWDGx_OUT 信号输出生成](#)；
 - SAC 的输出
- 通过通道输入信号 (TIx) 输入的为 SAC 的输出，参见 [输入捕获模式](#)；
- 断路输入信号参见 [使用断路功能](#)：
 - SYS-ERR: Cortex®-M0+ LOCKUP 输出
 - PVD 输出
 - CSS 故障触发信号
 - SAC 的输出

9.3.3 从 SAC、LSE 到 TIM2/3

SAC 和 LSE 输入到 TIM2/3 信号连接如下：

- 外部触发输入信号（ETR）参见[时钟选择](#)：
 - LSE 时钟
 - SAC 的输出
- 通过通道输入信号（TIx）输入的为 SAC 的输出，参见[输入捕获模式](#)；

9.3.4 从 SAC 到 TIM4

SAC 的输出通过通道输入信号（TIx）连接到 TIM4，参见[输入捕获模式](#)；

9.3.5 从 HSE、LSE、LSI、MSI、MCO 到 TIM4

TIM4 连接多个时钟源，可以实现时钟校准和测量的功能，连接到 TIM4 的信号如下：

- SAC 的输出通过外部触发输入信号（ETR）输入到 TIM4，参见
- 通道输入信号（TIx）参见[输入捕获模式](#)和 [TIM4 定时器输入选择寄存器 \(TIM4_TISEL\)](#)：
 - 时钟源信号：HSE(分频值可设置)、LSE、MSI、LSI、MCO、RTC_KCLK
 - RTC 的唤醒中断

9.3.6 从 RTC、SAC 到 LPTIM

输入到 LPTIM 的信号连接有如下两种：

- 通道输入信号（TIx）参见[LPTIM 引脚和内部信号](#)：
 - SAC 输出
- 触发输入信号参见[LPTIM 触发映射](#)：
 - RTC 闹钟
 - SAC 输出

9.3.7 从 HSE、LSE、HSI、MSI、LSI、PLL、SYSCLK 到 MCO

时钟输出功能可以将时钟通过 MCO 引脚输出，参见[时钟输出功能](#)。

9.3.8 从 LSE、LSI 到 LSCO

低速时钟输出可以将 LSE、LSI 通过 LSCO 引脚输出，参见[时钟输出功能](#)。

9.3.9 从 TIM3/4、USART2/3 到 IRTIM

TIM3/TIM4 的 TIM_x_OC1 和 USART2/3 的 TX 内部信号与 IRTIM 相连接，实现红外控制，详见[红外控制模块 \(IRTIM\)](#)。

9.3.10 从 TIM1/2/4/6、PLA、EXTI、TS、BGR 到 ADC

输入到 ADC 的信号有如下两种：

- 通道输入信号：TS、BGR，参见[通道选择](#)；
- 触发信号：TIM1 的 TRGO2 和 CC4、EXTI line11、TIM2/4/6 的 TRGO，PLAOUT1 和 PLAOUT2 参见[转换触发模式及触发沿](#)；

9.3.11 从 TIM1/2/3/4/5/6/7、PLA、LPTIM、EXTI 到 DAC

输入到 DAC 的信号连接如下：

- 触发信号：TIM1/2/3/4/5/6/7 的 TRGO、PLAOUT0~3、LPTIM 输出、EXTI line9，参见[DAC 触发选择](#)；

9.3.12 从 DMAMUX 到 DMA

DMA 的通道请求信号来源于 DMAMUX 的通道输出，参见[DMAMUX 信号映射](#)。

9.3.13 从 TIM4、LPTIM、EXTI、LCD 到 DMAMUX

本节列出的是 DMAMUX 的触发信号，详细的触发信号和请求信号参见[DMAMUX 信号映射](#)。

9.3.14 从 BGR 到 SAC

BGR 通过 COMP 的负端输入到 COMP，参见[引脚和内部信号](#)。

9.3.15 从 TIM1/2/3/4/5/6/7、SAC、LPTIM、USART1/2/3、SPI、RTC 到 PLA

参见 PLA 模块 [PLU 多路复用器输入选择](#)。

10 直接存储器访问控制器（DMA）

10.1 简介

直接存储器访问控制器（DMA）是总线矩阵的一个主机。DMA 用于存储器和存储器、存储器和外设、外设和外设之间数据传输，传输过程中无需 CPU 参与。

DMA 有 4 个通道，每个通道的请求源可配置，内置仲裁器处理不同优先级的通道请求。

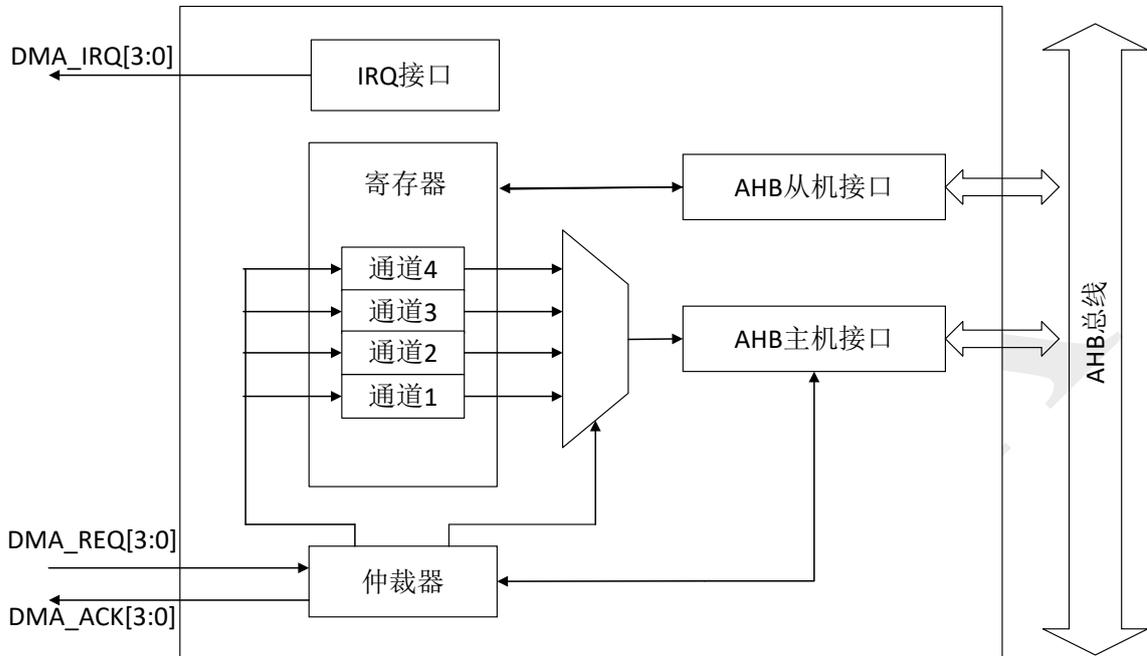
10.2 DMA 主要特性

- 存储器到存储器、存储器到外设、外设到存储器以及外设到外设的数据传输
- 每个通道可配置为由外设请求信号触发或由软件触发
- 每个通道可配置四个优先级（超高、高、中或低），优先级相同的情况下，通道号较小的优先级高
- 单次传输数据的位宽可配置为字节、半字或字
- 可配置的传输次数：0 到 65535
- 支持循环模式
- 可触发中断的事件：传输完成、传输一半和传输错误

10.3 DMA 功能说明

10.3.1 DMA 框图

图 10-1 DMA 框图



10.3.2 DMA 信号

表 10-1 DMA 内部信号

信号名称	信号类型	说明
$DMA_REQ[3:0]$	输入	DMA 请求信号
$DMA_ACK[3:0]$	输出	DMA 应答信号
$DMA_IRQ[3:0]$	输出	DMA 中断信号

10.3.3 DMA 请求源

DMA 请求源来自 DMAMUX，由 DMAMUX 管理 AHB/APB 外设的 DMA 请求，与 DMA 通道的连接，详见 [DMAMUX 信号映射](#)。

10.3.4 DMA 传输

DMA 传输可由外设请求触发，也可由软件触发；当存储器到存储器传输时使用软件触发。

外设请求触发进行一次数据传输的执行顺序如下：

- 1) 外设向 DMA 控制器发送单个 DMA 请求信号。
- 2) DMA 控制器根据对应通道的配置来执行此次数据传输，参见下文单次传输过程。

- 3) 只要收到外设发送的请求信号，DMA 控制器就会向外设发送应答信号。
- 4) 外设获得 DMA 控制器的应答信号后，便会立即释放其请求。
- 5) 一旦外设使请求失效，DMA 控制器就会释放应答信号。

外设可重复发送请求，每请求一次，执行一次上述操作。

软件触发（DMA_CCRx 寄存器 SRE 置 1）在通道使能后立即启动传输，无外设请求和应答时序。

DMA 单次传输过程：

- 1) 首先 DMA 从源地址读取单个数据（字节、半字或字）。再向目的地址写入单个数据（字节、半字或字）。
- 2) DMA_CNDTRx 寄存器在传输后递减。该寄存器包含待传输数据项的剩余数目。

10.3.5 优先级

DMA_CCRx 寄存器 POL[1:0]中配置通道优先级，共超高、高、中和低四个级别。如果两个通道配置的优先级相同，则通道号低的优先。例如，通道 1 的优先级高于通道 3。

当多个通道同时接收到请求时，最高优先级的通道优先启动传输。完成本次传输后仲裁器会在其他挂起通道（不包含当前通道）中再选择优先级最高的通道启动传输。

由于每次仲裁不包括当前通道，所以下次启动传输通道的优先级可能低于当前通道。例如：具有三个不同优先级的通道同时请求传输且传输请求连续，那么会在两个高优先级的通道中进行交替传输；当其中一个通道传输完成后，剩下两个通道再次进行交替传输。

10.3.6 传输模式

- 存储器到存储器

源和目的地址均为存储器时，由软件触发（DMA_CCRx 寄存器的 SRE 位置 1）启动传输，无需外设请求，在通道使能后会启动传输，直到 DMA_CNDTRx 寄存器达到零后，传输停止。

注意：通道为存储器到存储器传输模式时，需禁止循环模式（CIRC 清 0），否则会重复执行 DMA 传输。

- 存储器到外设或外设到存储器

存储器到外设或外设到存储器传输模式下，通道使能后，由外设的 DMA 请求触发启动数据传输。

- 外设到外设

源和目的地址都为外设时，分为两种情况：

- 有外设请求触发 DMA 通道。每次传输只能由当前通道设置的外设请求发起，另一外设作为此通道的被动访问外设。
- 无外设请求触发 DMA 通道。此时传输由软件触发（SRE 位置 1），通道使能后会启动传输。除源和目的地址不是存储器地址外，与存储器到存储器传输模式相同。

10.3.7 数据位宽

单次传数据位宽（字节、半字或字）由 DMA_CCRx 寄存器的 SIZE[1:0]位决定。

注意：传输地址与数据位宽保持对齐，否则寄存器会自动对齐地址，导致传输的数据错误。

表 10-2 DMA 地址对齐

SIZE[1:0]	数据位宽	源和目的地址
00	字节	无限制
01	半字	地址的第 0 位为 0
10	字	地址的第 0 位和第 1 位均为 0

10.3.8 地址递增

DMA_CCRx 寄存器的 SINC 或 DINC 决定源或目的地址指针在每次传输完成后是否递增，详情见下表。

表 10-3 DMA 地址递增

SINC	DINC	源地址	目的地址
0	0	地址固定	地址固定
0	1	地址固定	地址递增
1	0	地址递增	地址固定
1	1	地址递增	地址递增

SINC 或 DINC 置 1 时，则源地址指针或者目的地址指针在本次传输的地址加上 1、2 或 4，增加多少取决于 SIZE[1:0]配置的数据位宽。

首次传输的地址写入 DMA_CSARx 和 DMA_CDARx 寄存器中，在传输过程中，这些寄存器将保持初始写入的值。递增后的地址保存在内部的地址寄存器。

10.3.9 循环模式

当通道配置为非循环模式，在最后一次数据传输完成后（即待传输的数据数量 DMA_CNDTRx 减小到零后），通道继续保持使能状态，但是不处理任何 DMA 请求。由于通道使能状态下，无法修改 DMA_CNDTRx 寄存器的值，因此重新

使用通道进行相同配置传输时只需先禁止 DMA 通道,然后修改 DMA_CNDTRx 的值,并再次使能通道即可。

注意: 禁止通道 (DMA_CCRx 寄存器 EN 位清 0), DMA_CCRx、DMA_CSARx 和 DMA_CDARx 寄存器仍为上次配置的值。

当通道配置为循环模式,最后一次数据传输完成后,DMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 DMA_CSARx 和 DMA_CDARx 寄存器中的起始地址。循环模式可用于处理循环缓冲区,例如 ADC 10 个通道循环扫描的数据保存在 RAM 一段固定地址的缓冲区内;ADC 10 个通道的数据采集完成后再次回到第一个通道时,DMA 的起始地址也自动切回到初始地址,ADC 再次请求传输时,相同通道的数据将覆盖上一次数据。

10.3.10 配置流程

配置 DMA 通道 x 时需按照以下步骤操作:

- 1) 在 DMA_CCRx 寄存器中配置下列参数:
 - 传输数据位宽 SIZE[1:0]
 - 源地址递增模式 SINC
 - 目的地址递增模式 DINC
 - 通道优先级 PL[1:0]
 - 软件请求触发 SRE
 - 循环模式 CIRC
 - 传输完成中断 TC、传输一半中断 HT 和传输错误中断 TE
- 2) 配置 DMAMUX 输出请求。参见: [DMAMUX 配置流程](#)。
- 3) 在 DMA_CNDTRx 寄存器中设置传输的数据次数。
- 4) 在 DMA_CSARx 寄存器中设置读取数据的源地址。
- 5) 在 DMA_CDARx 寄存器中设置写入数据的目的地址。
- 6) 将 DMA_CCRx 寄存器中的 EN 位置 1 以使能通道。
- 7) 如果软件触发请求禁止,使能外设的 DMA 请求,等待外设发出请求以启动传输。如果软件触发请求使能,立即启动传输。

10.3.11 暂停和停止通道

● 暂停和恢复通道

如果通道 x 为无外设请求的存储器到存储器传输模式,仅将 DMA_CCRx 寄存器中的 EN 位清 0,通道其他配置不更改,可暂停此通道。再次 DMA_CCRx 寄存器中的 EN 位置 1 恢复通道传输。

如果通道 x 为有外设请求的传输模式，将外设请求禁止以暂停此通道传输。再次将外设请求使能恢复传输。

- 停止和重启通道

如果通道 x 为无外设请求的模式，仅将 DMA_CCRx 寄存器中的 EN 位清 0，即停止通道传输。如果重启此通道进行其他传输，重新配置新的传输参数并启动传输即可。

如果通道 x 为有外设请求的传输模式，先将外设请求禁止，然后禁止 DMA 通道（DMA_CCRx 寄存器中的 EN 位清 0），确保 DMA 没有挂起的请求。如果重启此通道进行其他传输，重新配置新的传输参数并启动传输即可。

10.3.12 错误标志

当通道 x 对保留的地址空间执行读写操作时，将生成 DMA 传输错误 TEIFx。发生了 DMA 传输错误时，则硬件会将相应 DMA_CCRx 寄存器的 EN 位清零，从而自动禁止出错的通道 x。在 DMA_ISR 寄存器的 TEIFx 位为 1 时，通道 x 不能被再次使能（DMA_CCRx 寄存器的 EN 置 1）。将 DMA_IFCR 寄存器中 CTEIFx 位置 1，TEIFx 标志将同步清 0。

如果通道 x 为外设请求参与传输，首先要停止外设的 DMA 模式，以禁止任何挂起或后续新生成的 DMA 请求。清除 TEIFx 后再正常地将 DMA 和外设重新配置，以便进行新的传输。

10.4 DMA 中断

DMA 中断可由每个通道的传输完成事件、传输一半事件或传输错误事件生成。每个事件可配置使能或者禁止中断。DMA 中断详情参见下表。

表 10-4 DMA 中断

中断事件	事件标志	使能控制位	清除方法
通道 x 传输完成	TCIFx	TCIE	DMA_IFCR 寄存器 CTCIFx 置 1 或者 DMA_IFCR 寄存器 CGIFx 置 1
通道 x 传输一半	HTIFx	HTIE	DMA_ICR 寄存器 CHTIFx 置 1 或者 DMA_ICR 寄存器 CGIFx 置 1
通道 x 传输错误	TEIFx	TEIE	DMA_ICR 寄存器 CTEIFx 置 1 或者 DMA_ICR 寄存器 CGIFx 置 1
通道 x 全局	GIFx	无使能位	DMA_ICR 寄存器 GCFx 置 1 或者 DMA_ICR 寄存器 CTCIFx、CHTIFx 和 CTEIFx 均置 1

10.5 DMA 寄存器

DMA 寄存器必须按字（32 位）进行访问。

10.5.1 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	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/位域	名称	描述
31:16	保留	必须保持复位值。
15	TEIF4	通道 4 的传输错误标志 传输错误时置 1。将 DMA_IFCR 寄存器中的 CTEIF4 位置 1，此位将清 0。 0: 无 TE 事件 1: 发生 TE 事件
14	HTIF4	通道 4 的传输一半标志 传输一半时置 1。将 DMA_IFCR 寄存器中的 CHTIF4 位置 1，此位将清 0。 0: 无 HT 事件 1: 发生 HT 事件
13	TCIF4	通道 4 的传输完成标志 传输完成时置 1。将 DMA_IFCR 寄存器中的 CTCIF4 位置 1，此位将清 0。 0: 无 TC 事件 1: 发生 TC 事件
12	GIF4	通道 4 的全局标志 传输错误、传输一半或传输完成时置 1。将 DMA_IFCR 寄存器中的 CGIF4 位置 1，TEIF4、HTIF4、TCIF4 和 GIF4 位都将清 0。 0: 无 TE、HT 或 TC 事件 1: 发生 TE、HT 或 TC 事件

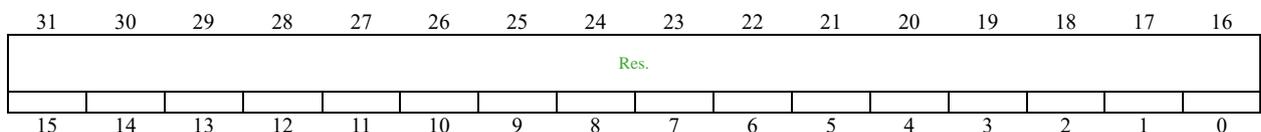
11	TEIF3	通道 3 的传输错误标志 传输错误时置 1。将 DMA_IFCR 寄存器中的 CTEIF3 位置 1，此位将清 0。 0: 无 TE 事件 1: 发生 TE 事件
10	HTIF3	通道 3 的传输一半标志 传输一半时置 1。将 DMA_IFCR 寄存器中的 CHTIF3 位置 1，此位将清 0。 0: 无 HT 事件 1: 发生 HT 事件
9	TCIF3	通道 3 的传输完成标志 传输完成时置 1。将 DMA_IFCR 寄存器中的 CTCIF3 位置 1，此位将清 0。 0: 无 TC 事件 1: 发生 TC 事件
8	GIF3	通道 3 的全局标志 传输错误、传输一半或传输完成时置 1。将 DMA_IFCR 寄存器中的 CGIF3 位置 1，TEIF3、HTIF3、TCIF3 和 GIF3 位都将清 0。 0: 无 TE、HT 或 TC 事件 1: 发生 TE、HT 或 TC 事件
7	TEIF2	通道 2 的传输错误标志 传输错误时置 1。将 DMA_IFCR 寄存器中的 CTEIF2 位置 1，此位将清 0。 0: 无 TE 事件 1: 发生 TE 事件
6	HTIF2	通道 2 的传输一半标志 传输一半时置 1。将 DMA_IFCR 寄存器中的 CHTIF2 位置 1，此位将清 0。 0: 无 HT 事件 1: 发生 HT 事件
5	TCIF2	通道 2 的传输完成标志 传输完成时置 1。将 DMA_IFCR 寄存器中的 CTCIF2 位置 1，此位将清 0。 0: 无 TC 事件

		1: 发生 TC 事件
4	GIF2	<p>通道 2 的全局标志</p> <p>传输错误、传输一半或传输完成时置 1。将 DMA_IFCR 寄存器中的 CGIF2 位置 1，TEIF2、HTIF2、TCIF2 和 GIF2 位都将清 0。</p> <p>0: 无 TE、HT 或 TC 事件</p> <p>1: 发生 TE、HT 或 TC 事件</p>
3	TEIF1	<p>通道 1 的传输错误标志</p> <p>传输错误时置 1。将 DMA_IFCR 寄存器中的 CTEIF1 位置 1，此位将清 0。</p> <p>0: 无 TE 事件</p> <p>1: 发生 TE 事件</p>
2	HTIF1	<p>通道 1 的传输一半标志</p> <p>传输一半时置 1。将 DMA_IFCR 寄存器中的 HTCF1 位置 1，此位将清 0。</p> <p>0: 无 TH 事件</p> <p>1: 发生 TH 事件</p>
1	TCIF1	<p>通道 1 的传输完成标志</p> <p>传输完成时置 1。将 DMA_IFCR 寄存器中的 CTCIF1 位置 1，此位将清 0。</p> <p>0: 无 TC 事件</p> <p>1: 发生 TC 事件</p>
0	GIF1	<p>通道 1 的全局标志</p> <p>传输错误、传输一半或传输完成时置 1。将 DMA_IFCR 寄存器中的 CGIF1 位置 1，TEIF1、HTIF1、TCIF1 和 GIF1 位都将清 0。</p> <p>0: 无 TE、HT 或 TC 事件</p> <p>1: 发生 TE、HT 或 TC 事件</p>

10.5.2 DMA 中断标志清除寄存器 (DMA_IFCR)

偏移地址: 0x04

复位值: 0x0000 0000



CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CTEIF4	通道 4 的传输错误清除 写 1 清除 DMA_ISR 寄存器中 TEIF4 标志。
14	CHTIF4	通道 4 的传输一半标志清除 写 1 清除 DMA_ISR 寄存器中 HTIF4 标志。
13	CTCIF4	通道 4 的传输完成标志 写 1 清除 DMA_ISR 寄存器中 TCIF4 标志。
12	CGIF4	通道 4 的全局标志 写 1 清除 DMA_ISR 寄存器中 TEIF4、HTIF4、TCIF4 和 GIF4 标志。
11	CTEIF3	通道 3 的传输错误清除 写 1 清除 DMA_ISR 寄存器中 TEIF3 标志。
10	CHTIF3	通道 3 的传输一半标志清除 写 1 清除 DMA_ISR 寄存器中 HTIF3 标志。
9	CTCIF3	通道 3 的传输完成标志 写 1 清除 DMA_ISR 寄存器中 TCIF3 标志。
8	CGIF3	通道 3 的全局标志 写 1 清除 DMA_ISR 寄存器中 TEIF3、HTIF3、TCIF3 和 GIF3 标志。
7	CTEIF2	通道 2 的传输错误清除 写 1 清除 DMA_ISR 寄存器中 TEIF2 标志。
6	CHTIF2	通道 2 的传输一半标志清除 写 1 清除 DMA_ISR 寄存器中 HTIF2 标志。
5	CTCIF2	通道 2 的传输完成标志 写 1 清除 DMA_ISR 寄存器中 TCIF2 标志。
4	CGIF2	通道 2 的全局标志

- 写 1 清除 DMA_ISR 寄存器中 TEIF2、HTIF2、TCIF2 和 GIF2 标志。
- 3 CTEIF1 通道 1 的传输错误清除
写 1 清除 DMA_ISR 寄存器中 TEIF1 标志。
- 2 CHTIF1 通道 1 的传输一半标志清除
写 1 清除 DMA_ISR 寄存器中 HTIF1 标志。
- 1 CTCIF1 通道 1 的传输完成标志
写 1 清除 DMA_ISR 寄存器中 TCIF1 标志。
- 0 CGIF1 通道 1 的全局标志
写 1 清除 DMA_ISR 寄存器中 TEIF1、HTIF1、TCIF1 和 GIF1 标志。

10.5.3 DMA 通道 x 控制寄存器 (DMA_CCRx)

偏移地址：0x08+0x14×(x-1)，(x=1 到 4)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CIRC	SRE	PL[1:0]		SIZE[1:0]		SINC	DINC	TEIE	HTIE	TCIE	EN
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
11	CIRC	循环模式 0: 禁止 1: 使能 <i>注意：使能通道后为只读，禁止写入。</i>
10	SRE	软件请求触发使能 0: 禁止，通道使能后必须有硬件请求输入才启动传输。 1: 使能，通道使能后即启动传输，硬件请求输入无效。 <i>注意：使能通道后为只读，禁止写入。</i>
9:8	PL[1:0]	优先级： 00: 低 01: 中

		10: 高
		11: 超高
		<i>注意：使能通道后为只读，禁止写入。</i>
7:6	SIZE[1:0]	传输数据
		00: 源和目的数据均为 8 位
		01: 源和目的数据均为 16 位
		10: 源和目的数据均为 32 位
		11: 保留
		<i>注意：使能通道后为只读，禁止写入。</i>
5	SINC	源地址递增
		0: 禁止
		1: 使能
		<i>注意：使能通道后为只读，禁止写入。</i>
4	DINC	目的地址递增
		0: 禁止
		1: 使能
		<i>注意：使能通道后为只读，禁止写入。</i>
3	TEIE	传输错误中断使能
		0: 禁止
		1: 使能
2	HTIE	传输一半中断使能
		0: 禁止
		1: 使能
1	TCIE	传输完成中断使能
		0: 禁止
		1: 使能
0	EN	通道使能
		发生通道传输错误后，此位清零。在 DMA_ISR 寄存器的 TEIFx 位清 0 后，此位才能再次置 1。
		0: 禁止
		1: 使能

10.5.4 DMA 通道 x 待传输次数寄存器 (DMA_CNDTRx)

偏移地址：0x0C+0x14×(x-1)，(x=1 到 4)

复位值：0x0000 0000

注意： 使能通道后为只读，禁止写入。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	NDT[15:0]	待传输次数 使能通道后，每次 DMA 传输后，该位域都会减 1，指示剩余的待传输次数。如果该位域为零，则不会处理任何传输请求。

10.5.5 DMA 通道 x 源地址寄存器 (DMA_CSARx)

偏移地址：0x10+0x14×(x-1)，(x=1 到 4)

复位值：0x0000 0000

注意： 使能通道后为只读，禁止写入。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA[31:16]															
rw	rw	rw	rw	rw	rw	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
SA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	SA[31:0]	源地址 SIZE[1:0]=01 (16 位) 时，忽略 SA[31:0]的位 0，访问将自动对齐到半字地址。 SIZE[1:0]=10 (32 位) 时，忽略 SA[31:0]的位 1 和位 0，访问将自动对齐到字地址。

10.5.6 DMA 通道 x 目的地址寄存器 (DMA_CDARx)

偏移地址：0x14+0x14×(x-1)，(x=1 到 4)

复位值：0x0000 0000

注意： 使能通道后为只读，禁止写入。

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

DA[31:16]															
rW	rW	rW	rW	rW	rW	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
DA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:0	DA[31:0]	<p>存储器地址</p> <p>此寄存器由软件置 1 和清零。</p> <p>SIZE[1:0]=01（16 位）时，忽略 DA[31:0]的位 0，访问将自动对齐到半字地址。</p> <p>SIZE[1:0]=10（32 位）时，忽略 DA[31:0]的位 1 和位 0，访问将自动对齐到字地址。</p>



11 DMA 请求多路器 (DMAMUX)

11.1 简介

DMA 请求多路器 (DMAMUX) 用于管理外设的 DMA 请求信号并传递给 DMA 控制器。

DMAMUX 包括请求复用器和请求生成器两个子模块。

请求复用器包含 4 个通道，每个通道通过可配置的映射关系连接外设和 DMA。每个 DMA 通道可从多个外设的 DMA 请求中选择其中一个。

请求生成器包含 2 个通道，每个通道通过触发信号生成 DMA 请求。

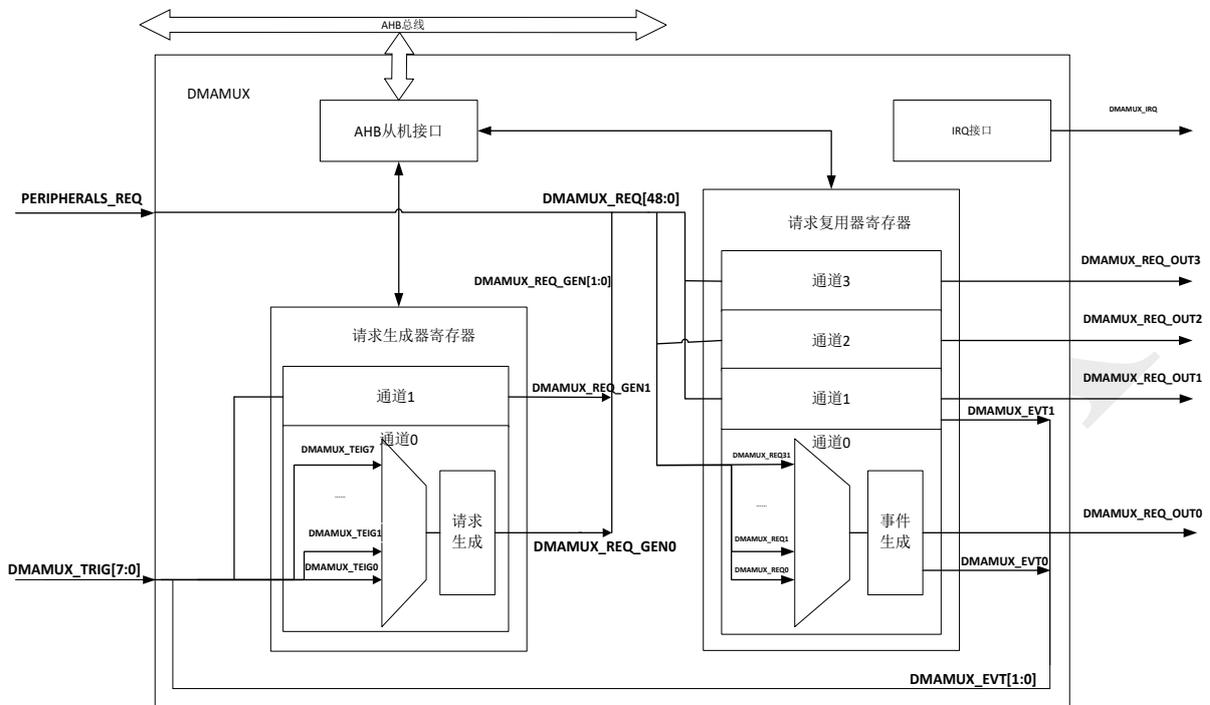
11.2 DMAMUX 主要特性

- 4 通道的可编程 DMA 请求复用器
- 2 通道 DMA 请求生成器
- 每个 DMA 请求生成器的通道均具有：
 - 8 个触发信号源
 - 请求生成计数器
 - 触发事件的溢出标志
- 每个 DMA 请求复用器的通道均具有：
 - 多个 DMA 请求源
 - 一个 DMA 请求输出
 - 事件输出 (仅限于通道 0 和 1)
 - DMAMUX 事件生成计数器 (仅限于通道 0 和 1)

11.3 DMAMUX 功能说明

11.3.1 DMAMUX 功能框图

图 11-1 DMAMUX 框图



11.3.2 DMAMUX 信号

表 11-1 DMAMUX 信号

信号名称	信号类型	说明
PERIPHERALS_REQ	输入	外设 DMA 请求信号。
DMAMUX_REQ_GEN	输出	DMAMUX 请求生成器输出的 DMA 请求信号。
DMAMUX_REQ_OUT	输出	DMAMUX 输出请求信号。DMAMUX 请求复用器的每个通道输出请求信号与 DMA 的对应通道输入请求信号连接。
DMAMUX_EVT	输出	DMAMUX 事件信号。DMAMUX 请求复用器的 0 和 1 通道可配置输出事件信号。
DMAMUX_TRIG	输入	DMAMUX 触发信号。

11.3.3 DMAMUX 信号映射

DMAMUX 请求复用器通道 0-3 的输出与相对应的 DMA 通道 1-4 相连。输入的外设请求信号或者自身生成的请求信号输出至 DMA 控制器。DMAMUX 输出请求信号和请求生成的触发信号分别见下表。

表 11-2 DMAMUX 输出请求信号

请求序号	请求外设	请求信号	
		DMA 通道 1, 3	DMA 通道 2, 4
1	DMAMUX	DMAMUX_REQ_GEN0	
2		DMAMUX_REQ_GEN1	
3	ADC	ADC	
4	I2C1	I2C1_RX	I2C1_TX
5	I2C2	I2C2_RX	I2C2_TX
6	LPUART	LPUART_RX	LPUART_TX
7	SPI1	SPI1_RX	SPI1_TX
8	SPI2	SPI2_RX	SPI2_TX
9	TIM1	TIM1_CH1	TIM1_CH2
10		TIM1_CH3	TIM1_CH4
11		TIM1_TRIG_COM	TIM1_UP
12	TIM2	TIM2_CH1	TIM2_CH2
13		TIM2_CH3	TIM2_CH4
14		TIM2_TRIG	TIM2_UP
15	TIM3	TIM3_CH1	TIM3_CH2
16		TIM3_CH3	TIM3_CH4
17		TIM3_TRIG	TIM3_UP
18	TIM4	TIM4_CH1	TIM4_CH2
19		TIM4_CH3	TIM4_CH4
20		TIM4_TRIG	TIM4_UP
21	TIM5/TIM6/TIM7	TIM5_UP	TIM6_UP
22		TIM7_UP	保留
23	USART1	USART1_RX	USART1_TX
24	USART2	USART2_RX	USART2_TX
25	USART3	USART3_RX	USART3_TX
26	DAC1&DAC2	DAC_CH1	DAC_CH2
27	DAC3&DAC4	DAC_CH3	DAC_CH4
28	TKA&TKB	TKA	TKB

表 11-3 DMAMUX 请求生成触发信号

请求生成触发序号	请求生成触发信号
0	EXTI LINE1
1	EXTI LINE2
2	EXTI LINE3
3	DMAMUX_EVT0
4	DMAMUX_EVT1
5	LPTIM_OUT
6	TIM4_OC2
7	保留

11.3.4 DMAMUX 请求复用器

DMAMUX 请求复用器每个通道可实现 DMA 请求的实际复用选择。DMA 请求来自外设或 DMAMUX 内部请求生成器。

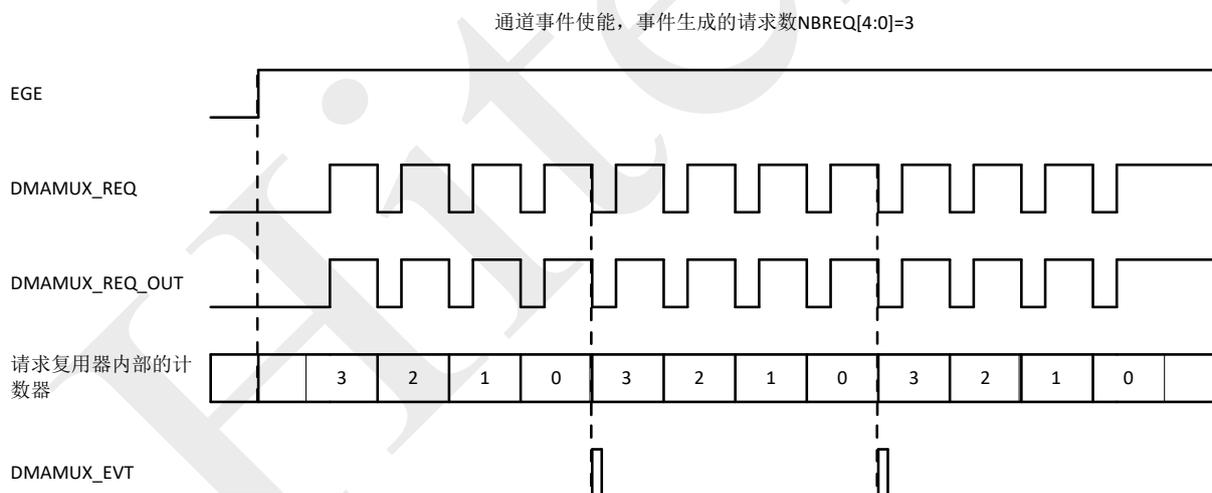
请求复用器通道 x 输出给 DMA 控制器的请求信号由 DMAMUX_CxCR 寄存器中的 DMAREQ_ID[4:0] 的值配置，参见表：DMAMUX 输出请求信号。

注意： 禁止将 DMAMUX 请求复器的两个不同通道同时配置为同一个外设 DMA 请求。当 DMA 不同通道同时收到接收同一外设请求时，会产生未知的硬件错误。

请求复用器的通道 0 和 1 除了配置 DMA 请求复用选择外，还可配置事件生成，实现通道互联。

请求复用器内部有一个可编程请求计数器，用于实现通道事件生成功能，即生成输出事件前的 DMA 请求的数量。通道事件生成功能由 DMAMUX_CxCR 寄存器的 EGE 位控制。EGE 使能时，通道 x 请求复用器内部计数器自动重新装入 DMAMUX_CxCR 寄存器的 NBREQ[4:0] 的值，每执行完成一次 DMA 传输该计数器减一。当内部计数器发生下溢时，会生成一个通道事件，如下图所示。因此通道 x 每传输 NBREQ[4:0]+1 个请求即生成一个事件。

图 11-2 DMAMUX 通道事件生成



注意： 如果 EGE 使能且 NBREQ[4:0]=0，则在每个的 DMA 请求结束后会生成一个事件。

11.3.5 DMAMUX 请求生成器

DMAMUX 请求生成器在接收到有效的触发信号后将产生 DMA 请求。DMAMUX 请求生成器有两个通道。请求生成器通道的输出连接到请求复用器的输入。

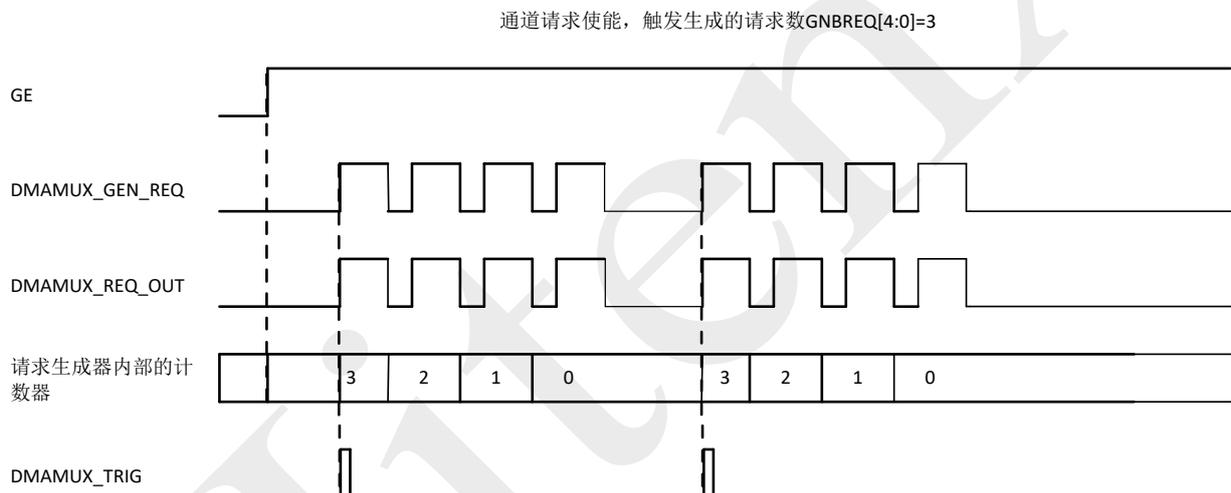
每个请求生成器通道 x 在 DMAMUX_RGxCR 寄存器中都有使能位 GE。

请求生成器通道 x 的触发输入通过 DMAMUX_RGxCR 寄存器中的 SIG_ID[2:0] 的值来选择。触发信号源参见表：[DMAMUX 请求生成触发信号](#)。

触发信号的有效边沿可通过 DMAMUX_RGxCR 寄存器中的 GPOL[1:0]位配置为上升沿、下降沿或任一边沿有效。

DMAMUX 请求生成器内部有一个可编程请求计数器，用于指示触发事件后 DMAMUX 请求生成器所生成的 DMA 请求数量。有效触发信号生成的 DMA 请求数量由 DMAMUX_RGxCR 寄存器中的 GNBREQ[4:0]的值配置。触发事件后，在相应通道开始输出 DMA 请求。DMA 控制器每次处理完成 DMAMUX 生成的请求时，该计数器都会递减。计数器下溢时，请求生成器会停止生成 DMA 请求，因此触发事件后生成的 DMA 请求的数量为 GNBREQ[4:0]+1。DMA 请求计数器将在出现下一个触发事件时自动装入设定值，如下图所示。

图 11-3 DMAMUX 通道请求生成



注意： 触发有效边沿后的状态需保持稳定的时间持续 2 个 AHB 时钟周期以上，确保触发事件有效边沿被检测到。

注意： 写入 DMAMUX_RGxCR 寄存器后，触发信号在 3 个 AHB 时钟周期后才可被检测到。

当上一触发信号生成请求还未全完成（内部计数器未发生下溢）时，又发生新的触发信号，则发生触发溢出事件，DMAMUX_RGSR 寄存器中 OFx 标志位置 1。将 DMAMUX_RGCFR 寄存器中的清除溢出标志位 COFx 置 1，溢出标志 OFx 将清 0。

11.3.6 配置流程

DMAMUX 请求复用器通道请求输出与 DMA 通道请求输入相连接，使用 DMA 功能除了配置 DMA 外还需配置 DMAMUX。

- 1) 配置通道 x 的 DMA 传输参数。参见：[DMA 配置流程](#)。

- 2) 配置通道 x 的 DMAMUX_CxCR 寄存器 DMAREQ_ID[4:0]请求 ID。
- 3) 如果 DMAREQ_ID[4:0]设置为 DMAMUX 请求生成器，则配置请求生成器通道 y 的 DMAMUX_RGxCR 寄存器，参数如下：
 - 请求生成触发信号 SIG_ID[2:0],如果 SIG_ID[2:0]设置为 DMAMUX_EVT0 或 DMAMUX_EVT1，则配置 DMAMUX_CxCR 寄存器的请求数量 NBREQ[4:0]位和事件生成使能 EGE 位
 - 要生成的 DMA 请求数量 GNBREQ[4:0]
 - 请求生成触发信号极性 GPOL[1:0]
 - 触发溢出中断 OIE

11.4 DMAMUX 中断

DMAMUX 请求生成器触发溢出事件可配置使能或禁止生成中断,DMAMUX 中断详情参见下表。

表 11-4 DMAMUX 中断

中断事件	事件标志	使能控制位	清除方法
请求生成器 触发溢出	O _{Fx}	OIE	DMAMUX_RGCFR 寄存器 COF 置 1

11.5 DMAMUX 寄存器

DMAMUX 寄存器可按字节（8 位）、半字（16 位）或字（32 位）访问。

11.5.1 DMAMUX 请求复用器通道 x 控制寄存器（DMAMUX_CxCR）

偏移地址：0x00+0x04×x，（x=0~3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		NBREQ[4:0]					EGE	Res.			DMAREQ_ID[4:0]				
		rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:9	NBREQ[4:0]	生成事件的 DMA 请求数 每传输 NBREQ+1 个请求时将生成通道事件。 <i>注意：使能事件生成后为只读，禁止写入。</i>

*DMAMUX_C2CR*和*DMAMUX_C3CR*此位保留，必须保持复位值。

- 8 EGE 事件生成使能
0: 禁止事件生成
1: 使能事件生成
注意: DMAMUX_C2CR 和 DMAMUX_C3CR 此位保留，必须保持复位值。

- 7:5 保留 必须保持复位值。

- 4:0 DMAREQ_ID[4:0] 请求信号源标识
1-27: 有效输出请求信号，参见表: *DMAMUX 输出请求信号*。
其他: 无输出请求信号。

11.5.2 DMAMUX 请求生成器 x 控制寄存器 (DMAMUX_RGxCR)

偏移地址: $0x100+0x04 \times x$, ($x=0\sim 1$)

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								GNBREQ[4:0]				GPOL[1:0]		GE	
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OIE	Res.				SIG_ID[2:0]		
								rw					rw	rw	rw

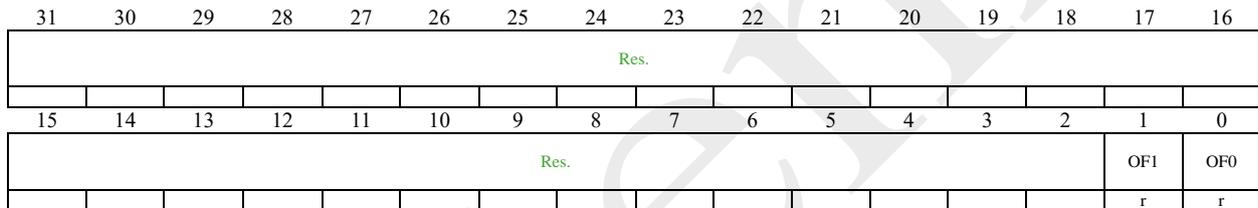
位/位域	名称	描述
31:14	保留	必须保持复位值。
23:19	GNBREQ[4:0]	生成 DMA 请求数 每个有效触发信号将生成 GNBREQ+1 个 DMA 请求。 <i>注意: 使能请求生成后为只读，禁止写入。</i>
18:17	GPOL[1:0]	触发极性 请求生成触发信号的的有效边沿极性: 00: 无有效边沿，不会触发请求生成 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
16	GE	请求生成器使能 0: 禁止请求生成器 1: 使能请求生成器

15:9	保留	必须保持复位值。
8	OIE	触发溢出中断使能 0: 禁止中断 1: 使能中断
7:3	保留	必须保持复位值。
2:0	SIG_ID[2:0]	触发信号源标识 选择请求生成触发信号源，参见表: DMAMUX 请求生成触发信号 。

11.5.3 DMAMUX 请求生成器中断状态寄存器 (DMAMUX_RGSR)

偏移地址: 0x140

复位值: 0x0000 0000



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OF1	请求生成器通道 1 触发溢出标志 当上一触发信号生成请求还未全完成（内部计数器未发生下溢）时，又发生新的触发信号，产生触发溢出事件此位置 1。 将 DMAMUX_RGCFR 寄存器中的 COF1 位置 1，此位将清 0。 0: 无 OF 事件 1: 发生 OF 事件
0	OF0	请求生成器通道 0 触发溢出标志 当上一触发信号生成请求还未全完成（内部计数器未发生下溢）时，又发生新的触发信号，产生触发溢出事件此位置 1。 将 DMAMUX_RGCFR 寄存器中的 COF0 位置 1，此位将清 0。 0: 无 OF 事件 1: 发生 OF 事件

11.5.4 DMAMUX 请求生成器中断标志清除寄存器 (DMAMUX_RGCFR)

偏移地址: 0x144

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														COF1	COF0
														w	w

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	COF1	请求生成器通道 1 触发溢出标志清除 写 1 清除 DMAMUX_RGSR 寄存器中 OF1 标志。
0	COF0	请求生成器通道 0 触发溢出标志清除 写 1 清除 DMAMUX_RGSR 寄存器中 OF0 标志。

12 嵌套向量中断控制器 (NVIC)

12.1 简介

嵌套向量中断控制器(NVIC)实现高效的异常处理及中断处理。所有的中断均由NVIC 管理。

12.2 主要特征

- 支持 32 个可屏蔽中断，以及 Cortex®-M0+ 的 16 个中断
- 低延时的异常及中断处理
- 提供系统控制寄存器
- 4 个可编程优先级，“0”代表最高优先级
- 中断向量表中的编号代表硬件优先级，当两个软件优先级相同的中断同时发生时，编号小的中断优先

NVIC和处理器内核紧密配合，实现了低延迟的中断处理，以及对晚到高优先级中断的高效处理。当异常发生时，NVIC会自动将处理状态压栈保存，中断子程序执行完毕会自动将状态出栈。详细信息请参见 *Cortex-M0+ Technical Reference Manual*。

12.3 中断向量表

表 12-1 中断向量表

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000_0000
-	-3	固定	复位	复位	0x0000_0004
-	-2	固定	NMI_Handler	不可屏蔽中断。HSE 的 CSS 中断、LSE 的 CSS 中断、PLL 的 CSS 中断均连接到 NMI 向量	0x0000_0008
-	-1	固定	HardFault_Handler	所有类型的错误	0x0000_000C
-	-	-	保留	保留	0x0000_0010 0x0000_0014 ~ 0x0000_0028
-	3	可设置	SVC_Handler	通过 SWI 指令调用的系统服务	0x0000_002C
-	-	-	保留	保留	0x0000_0030 0x0000_0034
-	5	可设置	PendSV_Handler	可挂起的系统服务请求	0x0000_0038
-	6	可设置	SysTick_Handler	系统节拍定时器	0x0000_003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000_0040

位置	优先级	优先级类型	缩略语	说明	地址
1	8	可设置	PVD	电源电压检出中断 (EXTI18)	0x0000_0044
2	9	可设置	RTC	RTC全局中断 (EXTI19)	0x0000_0048
3	10	可设置	FLASH	Flash 全局中断	0x0000_004C
4	11	可设置	RCC	RCC 全局中断	0x0000_0050
5	12	可设置	EXTI[1:0]	EXTI 通道 0 和 1 中断	0x0000_0054
6	13	可设置	EXTI[3:2]	EXTI 通道 2 和 3 中断	0x0000_0058
7	14	可设置	EXTI[15:4]	EXTI 通道 4 到 15 中断	0x0000_005C
8	15	可设置	PLA	PLA 全局中断 (EXTI20)	0x0000_0060
9	16	可设置	DMA1_Channel1	DMA1 通道 1 中断	0x0000_0064
10	17	可设置	DMA1_Channel2/3	DMA1 通道 2 和通道 3 中断	0x0000_0068
11	18	可设置	DMA1_Channel4/ DMAMUX	DMA1 通道 4 和 DMAMUX 中断	0x0000_006C
12	19	可设置	SAC(COMP)	比较器 1~4 中断 (EXTI16-17)	0x0000_0070
13	20	可设置	TIM1_BRK_UP_TRG_COM	TIM1 断路、更新、触发和换向事件中断	0x0000_0074
14	21	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000_0078
15	22	可设置	TIM2	TIM2 全局中断	0x0000_007C
16	23	可设置	TIM3	TIM3 全局中断	0x0000_0080
17	24	可设置	LPTIM	LPTIM 全局中断 (EXTI24)	0x0000_0084
18	25	可设置	TIM4	TIM4 全局中断	0x0000_0088
19	26	可设置	TIM5	TIM5 全局中断	0x0000_008C
20	27	可设置	TIM6/DAC	TIM6 全局中断和 DAC	0x0000_0090
21	28	可设置	TIM7	TIM7 全局中断	0x0000_0094
22	29	可设置	ATK	ATK 全局中断	0x0000_0098
23	30	可设置	I2C1	I2C1 全局中断 (EXTI21)	0x0000_009C
24	31	可设置	I2C2	I2C2 全局中断	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000_00A4
26	33	可设置	SPI2	SPI2 全局中断	0x0000_00A8
27	34	可设置	USART1	USART1全局中断 (EXTI22)	0x0000_00AC
28	35	可设置	USART2	USART2 全局中断	0x0000_00B0
29	36	可设置	LPUART	LPUART 全局中断 (EXTI23)	0x0000_00B4
30	37	可设置	USART3	USART3 全局中断	0x0000_00B8
31	38	可设置	ADC	ADC 全局中断	0x0000_00BC

注意：灰色部分对应 Cortex®-M0+ 中断。

13 扩展中断和事件控制器 (EXTI)

13.1 简介

扩展中断和事件控制器 EXTI 用于管理 CPU 唤醒和系统的唤醒。EXTI 的输入有可配置通道和直接通道两种。输出信号有三种，EXTI 可以输出唤醒事件到 PWR，可以产生中断信号输入到 NVIC 模块，还可以集中管理事件源，产生事件输出，经过 EVG 输出到 CPU 事件处理模块，参见 [EXTI 框图](#)。

EXTI 管理 26 个事件通道，包括 18 个可配置通道，和 8 个直接通道。

EXTI 内部集成了 I/O 端口多路选择器，可将任意 I/O 配置为唤醒源。

13.2 EXTI 主要功能

- 由输入的任意事件唤醒系统。
- 可配置通道，来自不含中断功能的外设及 GPIO，功能如下：
 - 触发沿可选。
 - 分别为上升沿和下降沿触发的中断提供独立的挂起状态位。
 - 为事件和中断分别提供独立的屏蔽功能，灵活管理 CPU 的唤醒、中断处理以及事件响应。
- 直接通道，来自含中断功能的外设，功能如下：
 - 上升沿触发。
 - 由于外设中已包含中断标志位，EXTI 不再重复提供挂起状态位。
 - 为事件和中断分别提供独立的屏蔽功能，灵活管理 CPU 的唤醒、中断处理以及事件响应。
- 集成 I/O 端口多路选择器，可将任意 I/O 配置为唤醒源。

13.3 EXTI 模块示意图

EXTI 的输入分为可配置通道和直接通道两种。输出分为事件输出，中断输出，和唤醒输出三种。

EXTI 信号处理模块内部实现的功能参见 [EXTI 功能描述](#)。

图 13-1 EXTI 框图

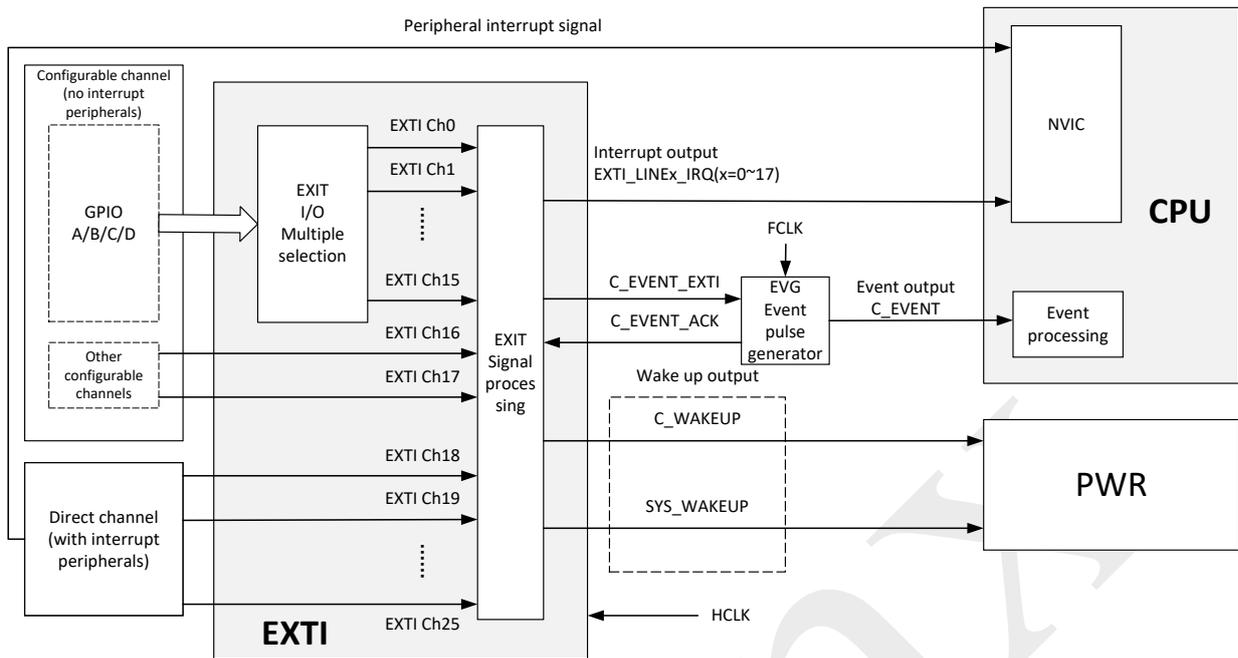


表 13-1 EXTI 信号说明

信号名称	类型	功能
EXTI_LINEx_IRQ	输出	由配置事件产生的中断信号 (x=0~17)
C_EVENT_EXTI	输出	EXTI 事件输出信号, 高电平有效, 与 HCLK 同步
C_EVENT_ACK	输入	异步反馈信号, 用于清零已受理的 C_EVENT_EXTI 信号
C_WAKEUP	输出	输出到 PWR 的 CPU 唤醒请求信号, 与 HCLK 同步
SYS_WAKEUP	输出	输出到 PWR 的异步系统唤醒请求, 用于唤醒系统时钟 SYSCLK 和 HCLK
HCLK	输入	EXTI 时钟, 同 AHB 总线时钟

表 13-2 EVG 信号说明

信号名称	类型	功能
C_EVENT_EXTI	输入	EXTI 事件输入, 高电平有效, 与 HCLK 同步
C_EVENT_FB	输出	异步反馈信号, 用于清零已受理的 C_EVENT_EXTI 信号
C_EVENT	输出	输出到 CPU 的事件输出信号, 与 CPU 时钟同步
FCLK	输入	Cortex®-M0+时钟 (CPU 时钟)

13.4 EXTI 的主要信号连接及功能

对于没有实现中断的外设, 其唤醒信号连接到 EXTI 的可配置通道。EXTI 为可配置通道提供了事件挂起标志, 挂起标志置位代表有中断待处理, 需要软件写“1”将标志位清零。EXTI 的事件挂起标志与中断标志位类似, 都可以触发 CPU 中断。

对于本身已实现唤醒事件及中断的外设，唤醒事件连接到 EXTI 的直接通道。直接通道的中断信号不通过 EXTI，直接输入到 NVIC 模块中，可以触发 CPU 的中断。直接通道的外设中已有中断标志位，所以 EXTI 中不再重复提供挂起标志位。直接通道的中断或唤醒事件处理过程中，CPU 清空外设的中断标志位。

事件输出由 EXTI 信号处理模块授权，输出事件经由 EVG 输入到 CPU 的事件处理模块。

EXTI 的唤醒输出连接到 PWR 模块，用于唤醒 CPU 和 CPU 子系统。

EXTI 中主要实现了两个部分，I/O 多路选择和 EXTI 信号处理。所有的 GPIO 端口都接入到 EXTI I/O 多路选择器，可选择将任意 GPIO 配置为 EXTI 的可配置通道，选中的 I/O 可将 CPU 从低功耗模式唤醒。EXTI 信号处理模块的功能请参考 [EXTI 功能描述](#)。

EXTI 的唤醒输出信号可以将 CPU 从低功耗模式下唤醒，唤醒方式主要有以下两种：

- 在使用 WFE 进入低功耗模式时，输入到 CPU 事件处理模块的事件信号可以将 CPU 从低功耗模式唤醒，中断信号也可以将 CPU 从低功耗模式唤醒。
- 在使用 WFI 进入低功耗模式时，中断信号可以将 CPU 从低功耗模式唤醒。

详情参见 [低功耗模式进入和退出](#)。

13.5 EXTI 通道及信号对应表

表 13-3 EXTI 通道及信号对应表

EXTI 通道	事件源	类型	外设关联的信号
0~15	GPIO	可配置	GPIO 输入信号
16	COMP1	可配置	SAC 比较器 1 输出结果
17	COMP2	可配置	SAC 比较器 2 输出结果
18~23			保留
24	PVD	直接	PVD 中断请求
25	RTC	直接	RTC 中断请求，包括闹钟中断、唤醒定时器中断
26	PLA	直接	PLA 中断请求
27	I2C1	直接	唤醒使能（WUPEN=1）时，地址匹配事件可用于唤醒，无需开中断。
28	USART1	直接	低功耗使能（UEWK=1）时，以下中断在未屏蔽时支持唤醒： 唤醒匹配事件中断(地址匹配)、接收非空中

EXTI 通道	事件源	类型	外设关联的信号
29	LPUART1	直接	断, 接收 FIFO 非空中断、接收 FIFO 满中断、达到接收 FIFO 阈值中断、发送 FIFO 空中断、达到发送 FIFO 阈值中断
30	LPTIM1	直接	以下中断请求在未屏蔽时支持唤醒: 比较匹配、自动重载匹配、外部触发事件、自动重载寄存器更新成功、比较寄存器更新成功
31	LSE CSS	直接	监测到 LSE 时钟故障时输出的中断信号

13.6 EXTI 功能描述

可配置通道的使能由 EXTI 控制, 通过配置触发沿寄存器使能相应通道。直接通道的使能在外设中。已使能的通道信号是否可以唤醒系统, 在中断唤醒屏蔽寄存器 EXTI_IMR 或事件唤醒屏蔽寄存器 EXTI_EMR 中设置。详见下表:

表 13-4 EXTI 屏蔽功能

唤醒屏蔽寄存器的配置		唤醒屏蔽寄存器的配置效果			
中断唤醒屏蔽寄存器 IMR.IMn	事件唤醒屏蔽寄存器 EMR.EMn	可配置通道挂起寄存器 RPR.RPIFn FPR.FPIFn	中断输出	事件输出	唤醒输出
0	0	无效	屏蔽	屏蔽	屏蔽
	1	无效	屏蔽	有效	有效
1	0	有效	有效	屏蔽	有效
	1	有效	有效	有效	有效

对于可配置通道, 事件的触发沿在 EXTI_RTSR/FTSR 寄存器中可配置, 检测到触发沿时即产生一个事件信号。产生事件信号时, 如果中断未屏蔽 (EXTI_IMR.IMn=1), 则事件挂起寄存器 EXTI_RPR/FPR 中的相应 RPIFn/FPIFn 位被置位, 产生唤醒信号唤醒系统时钟, 产生中断请求信号输入到 NVIC, 唤醒 CPU 并触发 CPU 中断。后续需将 RPIFn/FPIFn 位软件写“1”以清除中断请求。

直接通道的中断和事件的产生在外设模块中管理, EXTI 对直接通道仅支持上升沿检测。中断标志以及中断使能相关寄存器都在外设模块中, EXTI 不为直接事件提供挂起寄存器。中断未屏蔽时, 直接事件的中断请求可将 CPU 子系统唤醒。

未屏蔽的事件 (EXTI_EMR.EMn=1) 可以产生事件请求, 也可以作为唤醒源唤醒 CPU。EXTI 中的挂起寄存器 EXTI_RPR/FPR 仅响应需中断请求的可配置通道。

可配置通道与直接通道和 EXTI 寄存器的关系见下表:

表 13-5 EXTI 输入信号类型与寄存器的关系表

寄存器	触发沿选择		沿触发事件挂起		外部中断选择 (I/O 多路选择)		中断唤醒屏蔽	事件唤醒屏蔽
	<i>RTSR</i>	<i>FTSR</i>	<i>RPR</i>	<i>FPR</i>	<i>EXTICR1</i>	<i>EXTICR2</i>	<i>IMR</i>	<i>EMR</i>
可配置	√	√	√	√	√	√	√	√
直接	×	×	×	×	×	×	√	√

注意：“√”代表该寄存器适用于此通道，“×”代表该寄存器不适用于此通道。

以下章节对可配置通道和直接通道的三种输出（事件输出、中断输出和唤醒输出）分别进行说明。

13.6.1 EXTI 可配置通道

可配置事件的触发沿通过寄存器 EXTI_RTSR/FTSR 可选为上升沿、下降沿或双沿。

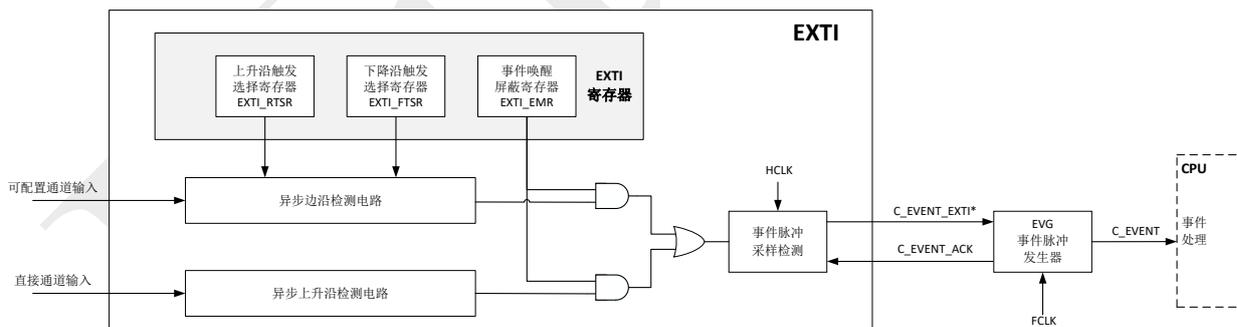
可配置通道的事件输出

可配置通道的事件输出由 EXTI_EMR 寄存器管理屏蔽，其结果通过 EVG 输出到 CPU 事件处理模块，可触发 CPU 的事件响应。

事件挂起寄存器(EXTI_RPR / EXTI_FPR)仅用于支持可配置通道的中断，任何事件输出无论是否屏蔽，都不影响事件挂起寄存器的状态。

参见下图：

图 13-2 可配置事件&直接事件的事件输出框图



注意：C_EVENT_EXTI 信号同时也是唤醒输出逻辑中的一路输入，参见图 13-4。

可配置通道的中断输出

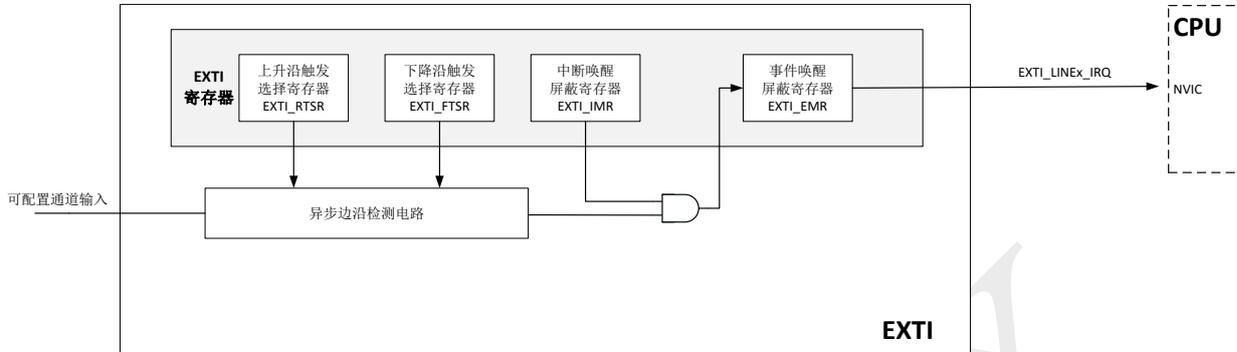
EXTI 为可配置通道实现中断管理，提供事件挂起寄存器（EXTI_RPR/FPR）。产生的中断信号 EXTI_LINE_x_IRQ 输入到 NVIC 管理，可触发 CPU 中断。

如果可配置通道的中断未屏蔽（EXTI_IMR.IM_n=1），在检测到有效触发沿时，事件挂起寄存器 EXTI_RPR/FPR 中的相应 RPIFn/FPIFn 位被置位。事件挂起标志位 RPIFn/FPIFn 的功能与中断标志位类似，被置位的挂起标志位需由软件清零。

事件挂起寄存器仅响应来自可配置通道且未被屏蔽的（IMn=1）中断请求。

参见下图：

图 13-3 可配置事件的中断输出框图



可配置事件的唤醒请求输出

CPU 的唤醒请求由 EXTI 集中管理，可配置通道的事件和中断都可以配置为唤醒源。唤醒请求输出到 PWR，用于唤醒系统时钟及 CPU。

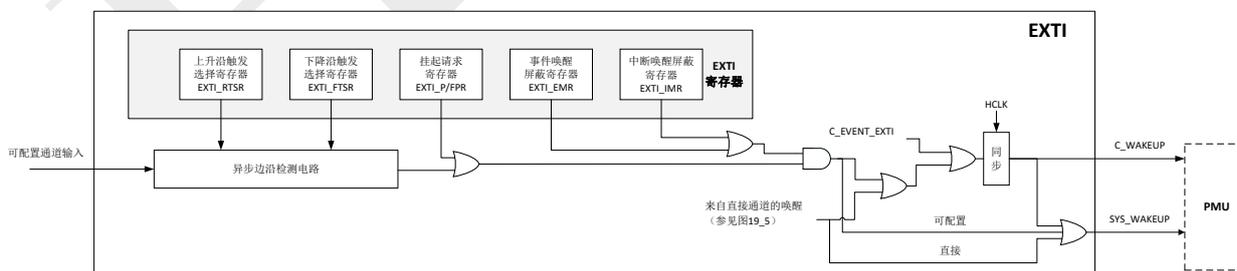
对于可配置事件，唤醒信号的触发沿在触发沿选择寄存器（EXTI_RTZR/FTZR）中可配置。

可配置通道通过事件或者中断唤醒 CPU。使用中断唤醒 CPU 时，挂起请求标志被置位，在唤醒后需通过软件将标志位清零，此标志位还可以用于定位触发唤醒的中断源。

EXTI_EMR 寄存器和 EXTI_IMR 寄存器中可设置事件和中断是否能将系统唤醒。

详见下图：

图 13-4 可配置事件的唤醒输出框图



注意：C_EVENT_EXTI 信号是事件输出的汇总，参见图 13-2。

13.6.2 EXTI 直接通道

直接通道的外设模块中已经包含了中断以及事件相关寄存器。EXTI 中不再为直接通道提供挂起寄存器。

EXTI 对直接通道仅支持上升沿检测。

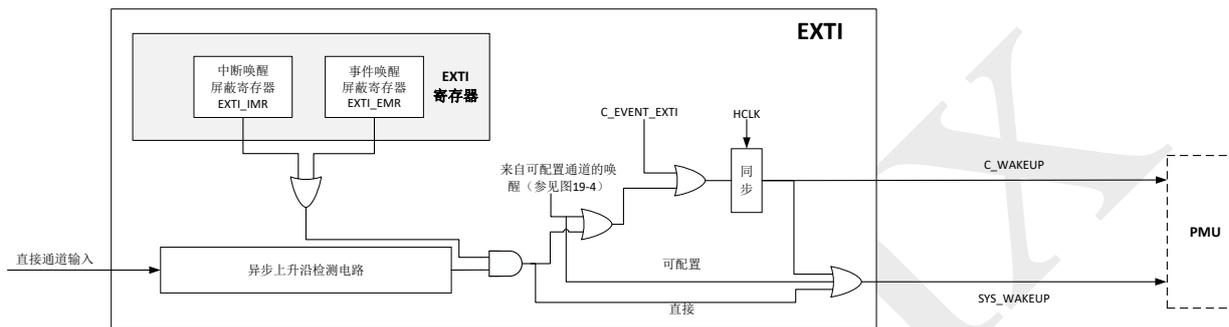
来自直接通道的事件或中断都可以触发唤醒请求。EXTI_EMR 寄存器 EXTI_IMR 寄存器用来设置该事件或中断是否能将系统唤醒。

直接通道的中断与 EXTI 无关，EXTI 对于直接通道仅提供事件输出及唤醒输出的相关支持。

直接通道的事件输出逻辑参见 [可配置事件&直接事件的事件输出框图](#)。

直接通道的唤醒输出示意图参见下图：

图 13-5 直接通道的唤醒信号输出框图

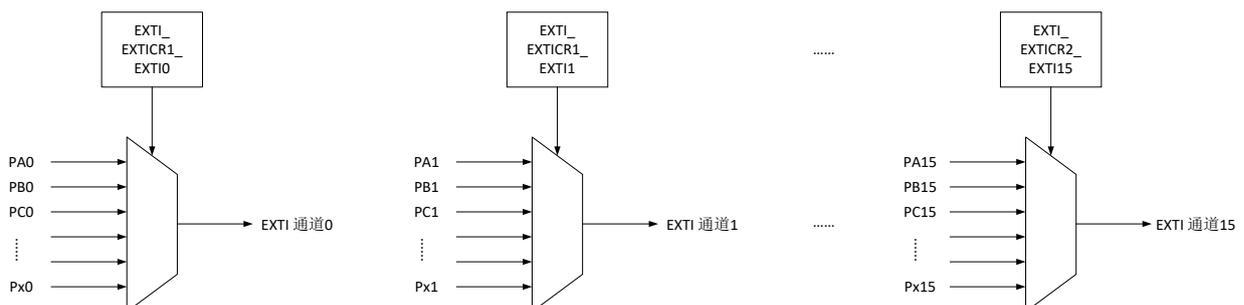


注意： 唤醒过程有以下几个步骤：首先 SYS_WAKEUP 信号用于唤醒 CPU 及子系统的时钟，时钟唤醒后与 HCLK 同步的 C_WAKEUP 信号产生，用于唤醒 PWR。同时时钟唤醒后产生的事件输出或中断输出可将 CPU 唤醒。在使用中断唤醒时（EXTI_IMR.IMn=1）中断信号输出到 NVIC 中唤醒 CPU。如果仅在 EXTI 中允许中断唤醒，但 NVIC 中屏蔽了作为唤醒源的中断，或 NVIC 模块未开启时，CPU 及系统时钟被 EXTI 唤醒，但是由于 NVIC 无响应所以 CPU 不会被唤醒，中断子程序也不会执行，这种状态可能会造成额外的功耗损失。因此 EXTI 中配置中断作为唤醒源时需注意 NVIC 的配置。

13.6.3 I/O 多路选择器

EXTI 的 I/O 多路选择器可选任意 GPIO 作为 EXTI 的事件信号源。所有的 GPIO 作为可配置事件分组接到 16 个多路选择器上，对应 EXTI 通道 0~15。在 EXTI_EXTICR1 和 EXTI_EXTICR2 寄存器中选择作为信号源的 GPIO。

图 13-6 EXTI 多路复用器



13.7 EXTI 寄存器

每个寄存器为 32 位，支持以字（32bits）、半字（16bits）或字节（Byte）为单位访问。

13.7.1 EXTI 上升沿触发选择寄存器（EXTI_RTSR）

偏移地址：0x00

复位值：0x0000 0000

注意： 配置为上升沿触发时，需要保证输入信号没有毛刺。
如果在写此寄存器的同时在相应通道上出现上升沿，该通道对应的挂起位不会被置位。

下降沿触发和上升沿触发可以同时启用，同时启用时任意沿都会触发中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														RT17	RT16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17:0	RTx	设置EXTI可配置通道 x(x=0~17)为上升沿触发。 每个bit为对应一个通道，为该通道的事件或中断使能上升沿触发 0：禁用上升沿触发。 1：使能上升沿触发。

13.7.2 EXTI 下降沿触发选择寄存器（EXTI_FTSR）

偏移地址：0x004

复位值：0x0000 0000

注意： 配置为下降沿触发时，需要保证输入信号上没有毛刺。
如果在写此寄存器的同时在配置的相应通道上出现下降沿，相应通道的挂起位不会被置位。

下降沿触发和上升沿触发可以同时启用，同时启用时任意沿都会触发中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														FT17	FT16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														rw	rw

FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17:0	FTx	设置EXTI可配置通道x(x=0~17)为下降沿触发。 每个bit为对应一个通道，为该通道的事件或中断使能下降沿触发。 0: 禁用下降沿触发。 1: 使能下降沿触发。

13.7.3 EXTI 上升沿触发事件挂起寄存器 (EXTI_RPR)

偏移地址: 0x008

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														RPIF17	RPIF16
														rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
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/位域	名称	描述
31:18	保留	必须保持复位值。
17:0	RPIF	在EXTI可配置通道 x(x=0~17)检测到上升沿触发事件并挂起，表示事件待处理。 各个bit在相应通道检测到上升沿事件时被置位。 软件写1清零。 0: 未检测到上升沿事件挂起请求。 1: 上升沿事件请求已挂起。

13.7.4 EXTI 下降沿触发事件挂起寄存器 (EXTI_FPR)

偏移地址: 0x00C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														FPIF17	FPIF16
														rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
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/位域	名称	描述
31:18	保留	必须保持复位值。
17:0	FPIF _x	EXTI可配置通道 $x(x=0\sim 17)$ 检测到下降沿触发事件并挂起，表示事件待处理。 各个bit在相应通道检测到下降沿事件时被置位。 软件写1清零。 0: 未检测到下降沿事件挂起请求。 1: 下降沿事件请求已挂起。

13.7.5 EXTI I/O 选择寄存器 1 (EXTI_EXTICR1)

偏移地址: 0x050

复位值: 0x0000 0000

EXTI_x 位域对应 GPIO 端口号

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI7[31:28]				EXTI6[27:24]				EXTI5[23:20]				EXTI4[19:16]			
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[15:12]				EXTI2[11:8]				EXTI1[7:4]				EXTI0[3:0]			
rw	rw	rw	rw												

位/位域	名称	描述
31:28	EXTI7[31:28]	EXTI7 GPIO端口选择 此位域用于配置EXTI7的中断源 0x0: PA7 引脚 0x1: PB7 引脚 0x2: PC7 引脚 0x3: PD7 引脚 其他配置均保留
27:24	EXTI6[27:24]	EXTI6 GPIO端口选择 此位域用于配置EXTI6的中断源 0x0: PA6 引脚 0x1: PB6 引脚 0x2: PC6 引脚 0x3: PD6 引脚 其他配置均保留
23:20	EXTI5[23:20]	EXTI5 GPIO端口选择 此位域用于配置EXTI5的中断源 0x0: PA5 引脚 0x1: PB5 引脚 0x2: PC5 引脚 0x3: PD5 引脚 其他配置均保留

19:16	EXTI4[19:16]	<p>EXTI4 GPIO端口选择</p> <p>此位域用于配置EXTI4的中断源</p> <p>0x0: PA4 引脚</p> <p>0x1: PB4 引脚</p> <p>0x2: PC4 引脚</p> <p>0x3: PD4 引脚</p> <p>其他配置均保留</p>
15:12	EXTI3[15:12]	<p>EXTI3 GPIO端口选择</p> <p>此位域用于配置EXTI3的中断源</p> <p>0x0: PA3 引脚</p> <p>0x1: PB3 引脚</p> <p>0x2: PC3 引脚</p> <p>0x3: PD3 引脚</p> <p>其他配置均保留</p>
11:8	EXTI2[11:8]	<p>EXTI2 GPIO端口选择</p> <p>此位域用于配置EXTI2的中断源</p> <p>0x0: PA2 引脚</p> <p>0x1: PB2 引脚</p> <p>0x2: PC2 引脚</p> <p>0x3: PD2 引脚</p> <p>其他配置均保留</p>
7:4	EXTI_1[7:4]	<p>EXTI_1 GPIO端口选择</p> <p>此位域用于配置EXTI1的中断源</p> <p>0x0: PA1 引脚</p> <p>0x1: PB1 引脚</p> <p>0x2: PC1 引脚</p> <p>0x3: PD1 引脚</p> <p>其他配置均保留</p>
3:0	EXTI0[3:0]	<p>EXTI0 GPIO端口选择</p> <p>此位域用于配置EXTI0的中断源</p> <p>0x0: PA0 引脚</p> <p>0x1: PB0 引脚</p> <p>0x2: PC0 引脚</p> <p>0x3: PD0 引脚</p> <p>其他配置均保留</p>

13.7.6 EXTI I/O 选择寄存器 2 (EXTI_EXTICR2)

偏移地址: 0x054

复位值: 0x0000 0000

EXTIx 位域对应 GPIO 端口号

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15[31:28]				EXTI14[27:24]				EXTI13[23:20]				EXTI12[19:16]			
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EXTI1[15:12]				EXTI0[11:8]				EXTI9[7:4]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	EXTI15[31:24]	EXTI15 GPIO端口选择 此位域用于配置EXTI15的中断源 0x0: PA15 引脚 0x1: PB15 引脚 0x2: PC15 引脚 其他配置均保留
27:24	EXTI14[27:24]	EXTI14 GPIO端口选择 此位域用于配置EXTI14的中断源 0x0: PA14 引脚 0x1: PB14 引脚 0x2: PC14 引脚 其他配置均保留
23:20	EXTI13[23:20]	EXTI13 GPIO端口选择 此位域用于配置EXTI13的中断源 0x0: PA13 引脚 0x1: PB13 引脚 0x2: PC13 引脚 0x3: PD13 引脚 其他配置均保留
19:16	EXTI12[19:16]	EXTI12 GPIO端口选择 此位域用于配置EXTI12的中断源 0x0: PA12 引脚 0x1: PB12 引脚 0x2: PC12 引脚 0x3: PD12 引脚 其他配置均保留
15:12	EXTI11[15:12]	EXTI11 GPIO端口选择 此位域用于配置EXTI11的中断源 0x0: PA11 引脚 0x1: PB11 引脚 0x2: PC11 引脚 0x3: PD11 引脚 其他配置均保留
11:8	EXTI10[11:8]	EXTI10 GPIO端口选择 此位域用于配置EXTI10的中断源 0x0: PA10 引脚 0x1: PB10 引脚 0x2: PC10 引脚 0x3: PD10 引脚 其他配置均保留
7:4	EXTI9[7:4]	EXTI9 GPIO端口选择 此位域用于配置EXTI9的中断源 0x0: PA9 引脚 0x1: PB9 引脚 0x2: PC9 引脚

0x3: PD9 引脚
其他配置均保留

3:0 EXTI8[3:0] EXTI8 GPIO端口选择
此位域用于配置EXTI8的中断源
0x0: PA8 引脚
0x1: PB8 引脚
0x2: PC8 引脚
0x3: PD8 引脚
其他配置均保留

13.7.7 EXTI 中断唤醒屏蔽寄存器 (EXTI_IMR)

偏移地址: 0x070

复位值: 0xFFFC 0000

默认值为屏蔽来自可配置通道的中断唤醒, 使能来自直接通道的中断唤醒。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	Res						IM17	IM16
rw							rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw							

位/位域	名称	描述
31:24	IMx	允许直接通道x的中断唤醒CPU (x=18~25) 置位表示允许来自此通道的事件中断将CPU唤醒。清零表示该通道的事件中断不会唤醒CPU。 此位域对应直接事件 0: 禁用中断唤醒 1: 使能中断唤醒
23:18	保留	必须保持复位值。
17:0	IMx	允许可配置通道x的中断唤醒CPU (x=0~17) 置位表示允许来自此通道的事件中断将CPU唤醒。清零表示该通道的事件中断不会唤醒CPU。 0: 禁用中断唤醒 1: 使能中断唤醒

13.7.8 EXTI 事件唤醒屏蔽寄存器 (EXTI_EMR)

偏移地址: 0x074

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EM25	EM24	EM23	EM22	EM21	EM20	EM19	EM18	Res						EM17	EM16

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
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	EMx	<p>允许可配置通道x的事件唤醒CPU (x=18~25)</p> <p>置位表示允许来自此通道的事件将系统唤醒。清零表示来自该通道的事件不会唤醒CPU。</p> <p>0: 禁用事件唤醒。</p> <p>1: 使能事件唤醒。</p>
23:18	保留	必须保持复位值。
17:0	EMx	<p>允许可配置通道x的事件唤醒CPU (x=0~17)</p> <p>置位表示允许来自此通道的事件将系统唤醒。清零表示来自该通道的事件不会唤醒CPU。</p> <p>0: 禁用事件唤醒。</p> <p>1: 使能事件唤醒。</p>

14 硬件除法器单元 (HDIV)

14.1 简介

HDIV (Hardware Divider) 是一个能自动执行 32 位有符号/无符号整数的硬件除法器。

14.2 HDIV 主要特性

HDIV 硬件除法器主要支持以下功能：

- 有符号/无符号整数除法计算可配置
- 32 位被除数和除数
- 输出 32 位商和 32 位余数
- 除数为零警告标志位，除法运算结束标志位
- 10 个时钟周期完成一次除法运算
- 写除数寄存器触发除法运算开始
- 读商寄存器/余数寄存器时自动等待计算结束

14.3 HDIV 功能说明

14.3.1 HDIV 功能描述

- **HDIV 使用说明：**
 - 1) 在 AHB 外设时钟使能寄存器 (**RCC_AHBENR**) 中打开硬件除法器的时钟使能位。
 - 2) 配置寄存器 HDIV_SIGNR，设置有符号/无符号除法运行。
 - 3) 配置寄存器 HDIV_DIVEDR，设置被除数。
 - 4) 配置寄存器 HDIV_DIVR，设置除数。
 - 5) 除法运算开始，查询寄存器 HDIV_SR 运算结束标志位 END,END 位 1 标志运算结束。读寄存器 HDIV_QUOTR 得到商，读寄存器 HDIV_RMDR 得到余数。
 - 6) 当除数为零时，除法运算立即结束，运算结果保持上一次运算的结果，同时除数为零警告标志位 ZERO 被置起。
 - 7) 在除法运算结束之前，读寄存器 HDIV_QUOTR/HDIV_RMDR 时，CPU 将被保持直到运算结束。

举例：计算一个无符号除法，被除数为 0x7250A3FB,除数为 0x12345678

- 1) 配置寄存器 HDIV_SIGNR 为 0，即无符号除法运算。

- 2) 配置寄存器 HDIV_DIVEDR 为 0x7250A3FB，即设置被除数。
- 3) 配置寄存器 HDIV_DIVR 为 0x12345678，即设置除数,计算开始。
- 4) 查询寄存器 HDIV_SR 运算结束标志位 END,END 为 1 标志，运算结束。
- 5) 读寄存器 HDIV_QUOTR 得到商 0x6。
- 6) 读寄存器 HDIV_RMDR 得到余数 (0x5169D2B)。

14.4 HDIV 寄存器

14.4.1 HDIV 被除数寄存器 (HDIV_DIVEDR)

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
DIVED[31:16]															
rw	rw	rw	rw	rw	rw	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
DIVED[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	DIVED	被除数寄存器位。

14.4.2 HDIV 除数寄存器 (HDIV_DIVR)

偏移地址：0x04

复位值：0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV[31:16]															
rw	rw	rw	rw	rw	rw	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
DIV[15:0]															
rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	DIV	除数 (写该寄存器自动触发除法运算)。

14.4.3 HDIV 商寄存器 (HDIV_QUOTR)

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
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

QUOT[31:16]															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUOT[15:0]															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro
位/位域	名称		描述												

31:0 QUOT 商结果寄存器。

14.4.4 HDIV 余数寄存器 (HDIV_RMDR)

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
RMD[31:16]															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMD[15:0]															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

位/位域	名称		描述												
31:0	RMD		余数结果寄存器位。												

14.4.5 HDIV 符号寄存器 (HDIV_SIGNR)

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

位/位域	名称		描述												
31:1	保留		必须保持复位值。												

0 SIGN 符号选择寄存器:
 0: 无符号除法运算;
 1: 有符号除法运算。

14.4.6 HDIV 状态寄存器 (HDIV_SR)

偏移地址: 0x14

复位值：0x0000 0001

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	ZERO	除数为零警告标志位： 0：除数不为零； 1：除数为零。
0	END	除法运算结束标志位： 0：运算进行中； 1：运算结束。

15 循环冗余校验 (CRC)

15.1 简介

CRC 根据初始值和多项式计算输入数据(8 位、16 位、32 位)的 CRC 值。CRC 运算通常用于验证数据在传输及存储时的完整性。

15.2 CRC 主要特性

- 支持 CRC-16 和 CRC-32 多项式,其中
 - CRC-16 多项式: 0x1021

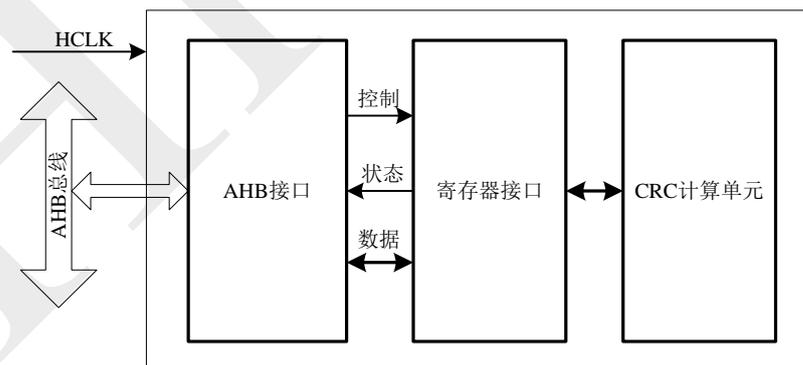
$$X^{16}+X^{12}+X^5+1$$
 - CRC-32 多项式: 0x4C11DB7

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$
- 支持计算模式和校验模式
- 寄存器访问支持 3 种位宽: 8 位、16 位和 32 位
- 1 个 AHB 时钟周期完成 CRC 计算

15.3 CRC 功能描述

15.3.1 CRC 框图

图 15-1 CRC 框图



15.3.2 CRC 内部信号

表 15-1 CRC 内部信号

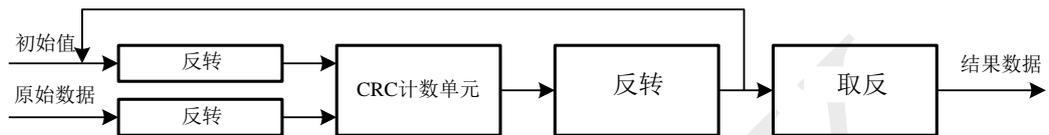
信号名称	信号类型	说明
HCLK	输入	AHB 时钟

15.3.3 CRC 操作说明

CRC 的 32 位数据寄存器 CRC_DR，用于输入原始数据。CRC 的 32 位结果寄存器 CRC_RR 用于输出计算生成的 CRC 值和输入初始值。

CRC 的计算单元将输入到 CRC_DR 的原始数据、CRC_RR 的初始值（或者前一次计算生成的数据反转后的值）反转后输入到 CRC 计算模块，计算生成的数据反转后按位取反，取反生成的结果数据输出到 CRC_RR。见下图。

图 15-2 CRC 计算单元框图



CRC 寄存器可按 8 位、16 位、32 位位宽访问。可以随时调整数据位宽，从而减少输入数据的写操作次数。例如，对 7 个字节数据进行 CRC 计算，可先按字写入，再按半字写入，最后按字节写入。不同位宽访问示例如下：

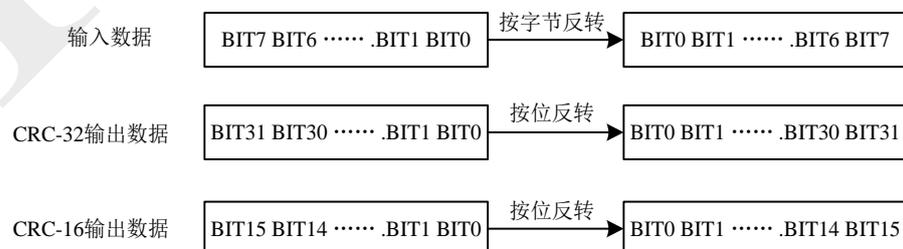
- 8 位位宽访问示例：0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16 位位宽访问示例：0x1100, 0x3322, 0x5544, 0x7766
- 32 位位宽访问示例：0x33221100, 0x77665544

CRC 1 个 AHB 时钟周期即可完成 8 位、16 位、32 位数据计算，因此可立即向数据寄存器写入下一个数据，不需要等待时间。

CRC 输入数据按字节反转（见 [图：输入输出数据反转框图](#)）。例如：输入数据 0x11223344，执行反转后数据为 0x8844CC22。

CRC 输出数据按位反转（见 [图：输入输出数据反转框图](#)）并且与 0xFFFFFFFF（或 0xFFFF）异或。例如：CRC-32 输出数据 0x11223344，执行反转后数据为 0x22CC4488，再执行取反后数据为 0xDD33BB77。

图 15-3 输入输出数据反转框图



CRC 支持计算模式和校验模式。CRC 计算模式是指向 CRC 输入原始数据，然后获取 CRC 计算结果。CRC 校验模式是向 CRC 输入待校验的原始数据和待检验的原始数据的 CRC 值，验证原始数据和 CRC 值是否匹配。两种模式操作步骤如下：

- 计算模式
 - 1) 配置多项式，将多项式配置值写入控制状态寄存器 CRC_CR 中 POLYSIZE 位。CRC-16 多项式配置值为 0，CRC-32 多项式配置值为 1。
 - 2) 将初始值写入结果寄存器 CRC_RR。
 - 3) 将待计算的原始数据按 8 位、16 位、32 位位宽方式，依次写入数据寄存器 CRC_DR。
 - 4) 读取结果寄存器 CRC_RDR 中 CRC 值。
- 校验模式
 - 1) 配置多项式，将多项式配置值写入控制状态寄存器 CRC_CR 中 POLYSIZE 位。CRC-16 多项式配置值为 0，CRC-32 多项式配置值为 1。
 - 2) 将初始值写入结果寄存器 CRC_RR。
 - 3) 将待校验的原始数据按 8 位、16 位、32 位位宽方式，依次写入数据寄存器 CRC_DR。
 - 4) 将待校验的原始数据的 CRC 值按 8 位、16 位、32 位位宽方式写入数据寄存器 CRC_DR。
 - 5) 读取控制状态寄存器 CRC_CR 中 FLAG 标志位，判定 CRC 校验是否正确。FLAG 位为 1 时，表示当前 CRC 校验正确；反之，当前 CRC 校验错误。

15.4 CRC 寄存器

CRC 寄存器可按字节（8 位）、半字（16 位）或字（32 位）访问。

15.4.1 CRC 控制状态寄存器（CRC_CR）

偏移地址：0x00

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														FLAG	POLY SIZE
														r	rw

位/位域	名称	描述
31:2	保留	写入无效。
1	FLAG	CRC 校验结果标志位 0: 当前 CRC 校验错误

1: 当前 CRC 校验正确

0 POLYSIZE CRC 多项式选择位
 0: CRC-16
 1: CRC-32

15.4.2 CRC 结果寄存器 (CRC_RR)

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
RESULT[31:16]															
rw	rw	rw	rw	rw	rw	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
RESULT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	RESULT[31:0]	CRC 计算结果 读取 RESULT[15:0]以获取 CRC-16 的计算结果; 读取 RESULT[31:0]以获取 CRC-32 的计算结果; 向 RESULT[15:0]写入初始值以初始化 CRC-16 计算; 向 RESULT[31:0]写入初始值以初始化 CRC-32 计算。

15.4.3 CRC 数据寄存器 (CRC_DR)

偏移地址: 0x80

复位值: 0x0000 0000

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

位/位域	名称	描述
31:0	DR[31:0]	输入数据

16 模数转换器（ADC）

16.1 简介

MCU 内置 12 位逐次趋近型（SAR）模数转换器（ADC），可以对 56 路外部通道和 3 路芯片内部通道的电压信号进行模数转换，并可以配置为单次、连续或间断转换模式。ADC 转换结果存储在一个 12 位数据寄存器，可以配置为左对齐或右对齐数据存储模式。

ADC 具有模拟看门狗功能，用于监控 ADC 通道输入电压是否超出用户设定阈值的上限或下限。

16.2 ADC 主要特性

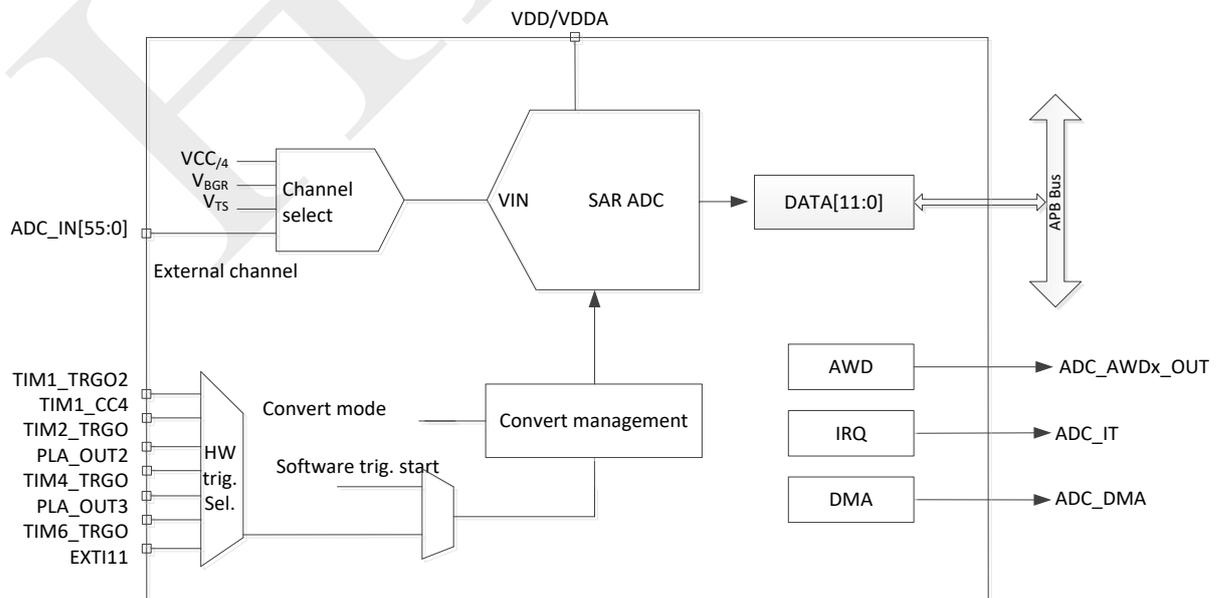
- 高性能
 - 可编程采样时间；
 - 支持 1Msps ADC 转换速率；
 - 自校准；
 - 可配置数据对齐方式：左对齐、右对齐；
 - 支持 DMA。
- 低功耗
 - ADC 可以选择同步时钟 PCLK、异步时钟两种时钟源。通过选择异步时钟源，可以降低 PCLK 频率从而使 MCU 以低功耗模式运行，同时仍可保持最优的 ADC 性能。例如，无论 PCLK 的频率如何，都可以保持 1 μ s 的转换时间；
 - 等待模式：ADC 转换完成后停止运行，直到 ADC 数据被读取。此模式可以防止 ADC 在低频应用中转换数据溢出；
- 模拟输入通道
 - 56 路外部输入通道（通过模拟 PIN 输入）；
 - 3 路芯片内部通道：
 - 1 路用于采集内部温度传感器 (V_{TS}) 电压；
 - 1 路用于采集参考电压 (V_{BGR}) ；
 - 1 路用于监控 1/4 外部 VCC 电源输入。
- 启动转换方式
 - 软件触发；

- 硬件触发：触发极性（上升沿、下降沿、双沿）可配置（触发源为内部定时器事件或 GPIO 输入事件）。
- 转换模式
 - 单通道转换，或者扫描一系列通道；
 - 单次模式，每次触发时转换一次选定的输入通道；
 - 连续模式，连续转换选定的输入通道；
 - 间断模式。
- 中断触发事件
 - ADC 就绪事件；
 - 转换结束事件；
 - 序列转换结束事件；
 - 模拟看门狗事件；
 - 溢出事件。
- 模拟看门狗
- ADC 电源电压：2.4 V~5.5V
- ADC 输入范围：VSSA ≤ VIN ≤ VDDA

16.3 ADC 功能说明

ADC 结构框图如下图。

图 16-1 ADC 结构框图



16.3.1 校准

由于制造所产生的差异，各芯片的偏移误差有所不同。ADC 具有校准功能，用于消除偏移误差。在校准过程中，ADC 会计算校准系数，此校准系数在 ADC 下电之前一直有效。另外在校准过程中，应用程序禁止使用 ADC，需等待校准完成。

校准前需使能 ADC，即将 ADC_CR 寄存器的 ADEN 位置 1。为了使校准更加准确，校准次数可配置为 8、16、32、64 次，结果为校准多次的平均值；默认为 8 次，需在校准启动之前配置 ADC_CALFACT 寄存器的 CALTIMES 位。应用程序通过将 ADC_CR 寄存器的 ADCAL 位置 1 启动校准。校准过程中 ADCAL 位一致保持为 1；当校准完成后，此位清零。之后，可从 ADC_CALFACT 寄存器读取校准系数（位 5~位 0）。

当 ADC 外设电源 VDDA 移除或运行条件发生改变（VDDA、温度等），建议重新进行校准。

在下列情况下，校准系数会清除：

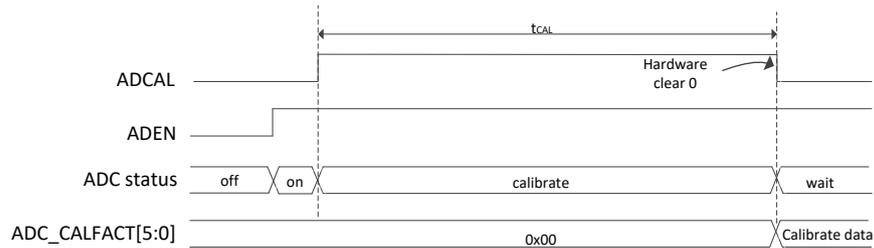
- ADC 外设复位。

由于 ADC 外设电源 VDDA 移除时，校准系数丢失。因此可以将校准参数保存到 FLASH，当重新使能 ADC 时，采用此系数加载到校准寄存器。这样只要电压、温度等条件不改变，可以采用之前保存的校准系数，无需重新执行校准流程，从而节省校准时间。

如果 ADC 已使能但未进行转换（ADEN=1 且 ADSTART=0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。校准系数载入过程不会对转换的启动造成延迟。

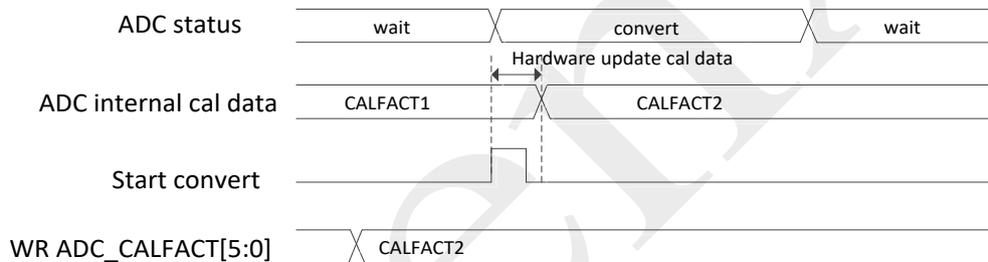
ADC 校准步骤：

- 1) 确保 ADEN 置 1；
- 2) 将 ADCAL 置 1，使能校准；
- 3) 等待校准完成，直至 ADCAL=0（或直至 EOCAL=1）。如果已将 ADC_IER 寄存器中的 EOCALIE 位置 1 来使能中断，可会产生校准中断，在中断服务函数中处理校准完成事件；
- 4) 校准系数可从 ADC_CALFACT 读取。

图 16-2 ADC 校准时序


校准系数软件加载步骤:

- 1) 确保 ADEN=1 且 ADSTART=0 (ADC 启动时没有进行任何转换);
- 2) 将保存的校准系数写入 ADC_CALFACT 寄存器;
- 3) 启动新的转换后, 将立即使用校准系数。

图 16-3 校准系数加载时序


16.3.2 开关控制

ADC 使能、禁止由以下两个 ADC_CR 寄存器位进行控制:

- 通过将 ADC_CR 寄存器的 ADEN 位置 1, 使能 ADC。使能 ADC 后需要等待一段稳定时间 t_{STAB} , 当 ADC 准备就绪, ADRDY 标志会立即置 1。
- 通过将 ADC_CR 寄存器的 ADDIS 置 1, 禁止 ADC, 并使 ADC 进入掉电模式。当 ADC 被完全禁止后, ADEN 位和 ADDIS 位由硬件自动清零。

使能 ADC 后, 如果配置为软件触发模式, 则通过软件将 ADC_CR 寄存器的 ADSTART 位置 1 启动 ADC 转换; 如果配置为外部硬件触发模式, 则等待外部触发事件产生时, 启动 ADC 转换。

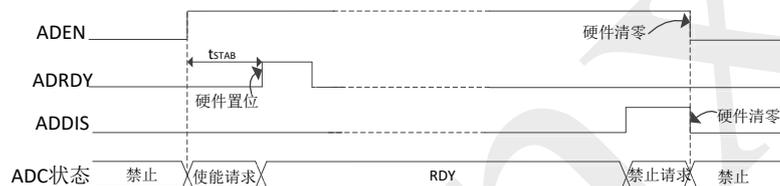
使能 ADC 步骤:

- 1) 将 ADC_ISR 寄存器中的 ADRDY 位编程为 1, 清除此位;
- 2) 将 ADC_CR 寄存器中的 ADEN 位置 1;
- 3) 等待 ADC 启动完成状态位 ADRDY 位置 1 (经过 ADC 启动时间 t_{STAB} 后, 此位会置 1)。如果 ADRDYIE 中断使能, 则会产生 ADC 使能完成中断, 此时可在中断服务函数中进行处理。

禁止 ADC 步骤:

- 1) 检查 ADC_CR 寄存器中的 ADSTART 位是否为 0，以确保当前未执行任何转换。如有需要，通过向 ADC_CR 寄存器中的 ADSTP 位写入 1，并等待此位清零，以停止正在进行的转换；
- 2) 将 ADC_CR 寄存器中的 ADDIS 位置 1；
- 3) 如有需要，可等待 ADC_CR 寄存器中的 ADEN=0，这表明 ADC 已完全禁止（ADEN=0 后，ADDIS 会自动复位）；
- 4) 将 ADC_ISR 寄存器中的 ADRDY 位编程为 1，将此位清零（可选）。

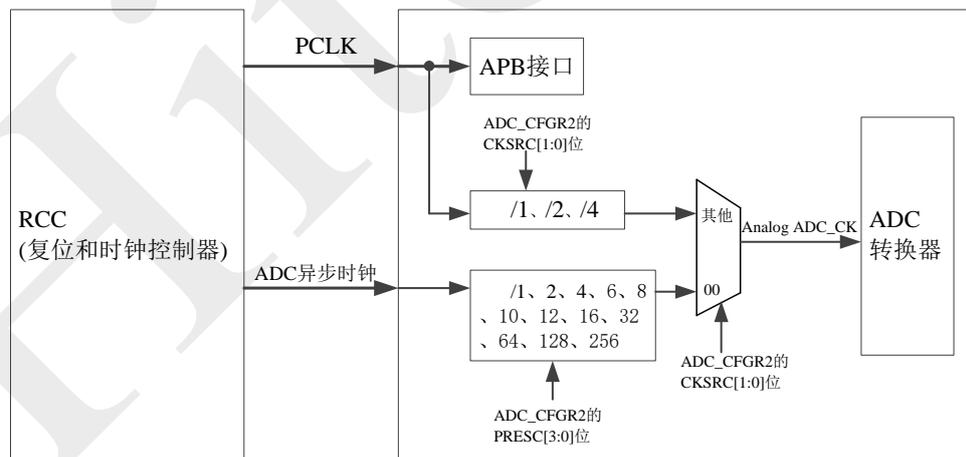
图 16-4 使能/禁止 ADC



16.3.3 时钟源

ADC 采用双时钟域架构，除了 APB 时钟（PCLK）外，ADC 还可由专属 ADC 时钟驱动。此 ADC 时钟与 APB 时钟异步，并独立于 APB 时钟。

图 16-5 ADC 时钟结构



1. 请参见复位和时钟控制(RCC)，了解 PCLK 和 ADC 异步时钟的使能方式。

模拟 ADC 的输入时钟可从以下两个不同的时钟源之间进行选择：

- ADC 时钟可以选择名为“ADC 异步时钟”的特定时钟源，该时钟源独立于 APB 时钟，并与 APB 时钟异步。

更多有关此时钟源的信息，请参见 RCC 部分。

要选择此时钟源，ADC_CFGR2 寄存器的 CKMODE[1:0]位需配置为“11”。

可以通过对 ADC_CFGR2 寄存器中的 PRESC[3:0]位进行配置，从而对 ADC 异步时钟源进行分频（1、2、4、6、8、10、12、16、32、64、128、256 分频）。

- ADC 时钟可以选择 ADC 总线接口的 APB 时钟，且可以对 APB 时钟进行分频（根据 CKMODE[1:0]寄存器位可以配置为 1、2、4 分频）。

要选择此时钟源，ADC_CFGR2 寄存器的 CKMODE[1:0] 位不得配置为“11”。

选择 ADC 异步时钟源的优势在于无论 APB 时钟源频率如何设置，都可以达到最大 ADC 时钟频率。

选择 APB 时钟源的优势在于无需进行时钟域重新同步。如果 ADC 由定时器触发，而且要求 ADC 触发延时确定，可选择此时钟源（否则，重新同步 APB 时钟与 ADC 异步时钟这两个时钟域会为触发时刻带来不确定性）。

表 16-1 触发与转换开始之间的延迟

ADC 时钟源	CKMODE [1:0]	触发事件与转换开始之间的延迟
PCLK 1 分频	00	延迟是确定的（无抖动），等于 4 个 ADC 时钟周期
PCLK 2 分频	01	延迟是确定的（无抖动），等于 3.5 个 ADC 时钟周期
PCLK 4 分频	10	延迟是确定的（无抖动），等于 3.25 个 ADC 时钟周期
HSI 二分频， SYSCLK ⁽¹⁾	11	延迟是不确定的（存在抖动）

1. 选择 RCC_CCIPR 寄存器的 ADCSEL 位。

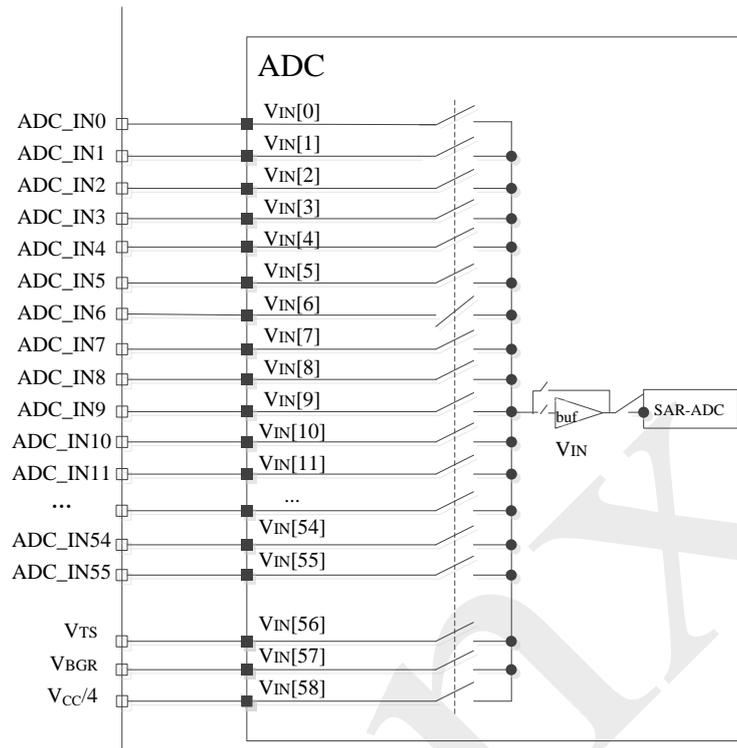
注意： 选择 CKMODE[1:0]=00 (PCLK 1 分频) 时，用户必须确保 PCLK 时钟占空比 50%。为此，在 RCC 参数配置中，用户必须选择占空比为 50% 的系统时钟，并且必须以旁路模式在 RCC 内配置 APB 预分频系数（请参见 RCC 部分）。对于内部源时钟，这仅意味着 AHB 和 APB 预分频器不会对时钟进行分频。

注意： ADC_CLK 的最大频率参见 MCU 数据手册。

16.3.4 输入通道

ADC 输入可以选择连接到外部通道或内部通道，如下图所示。

其中包括 56 路外部输入通道，以及 3 路内部输入通道（VBGR、VTs、VCC/4）。

图 16-6 ADC 输入通道


16.3.5 通道选择

ADC 模块具有 59 路输入通道：

- 56 路外部模拟输入通道，通过 GPIO 引脚输入 (ADC_IN_x)；
- 3 路内部模拟输入通道(温度传感器、内部参考电压、外部电源电压通道)。

ADC 可以转换单个通道，也可以自动扫描通道序列。

待转换通道序列须在 ADC_CHSELR 通道选择寄存器中进行配置：每个模拟输入通道都有对应选择位 (SQ_x)。

- 全通道配置序列
 - 扫描序列长度最多可达 8 通道；
 - 通道扫描顺序不由通道号确定，通过 ADC_CHSELR1 和 ADC_CHSELR2 寄存器的 SQ1[5:0] to SQ8 [5:0] 位来配置；
 - 通道扫描序列只可以选择通道 0 到通道 58；
 - 如果扫描序列检测到 SQ_x[5:0]=0b111111，将会忽略后续 SQ_x[5:0] 寄存器设定；
 - 如果扫描序列检测没有检测到 SQ_x[5:0]=0b111111，则会扫描所有的 8 个通道。

仅当 $ADSTART=0$ 时（即当前未进行任何转换），软件才可以对 $CHSELR$ 寄存器位执行写操作。

ADC 内部通道：温度传感器、 V_{BGR} 和 V_{CC} 。

- 温度传感器连接至通道 $ADC_VIN[56]$ ；
- 内部参考电压 V_{BGR} 连接至通道 $ADC_VIN[57]$ ；
- V_{CC} 通道连接至通道 $ADC_VIN[58]$ 。

16.3.6 采样时间

在开始进行模数转换之前，ADC 需要在待测量电压源与 ADC 内置采样电容之间建立直接连接。用户需配置采样时间，该采样时间必须足以使输入电压源为采样电容充电，并将电容电压保持在输入电压水平。

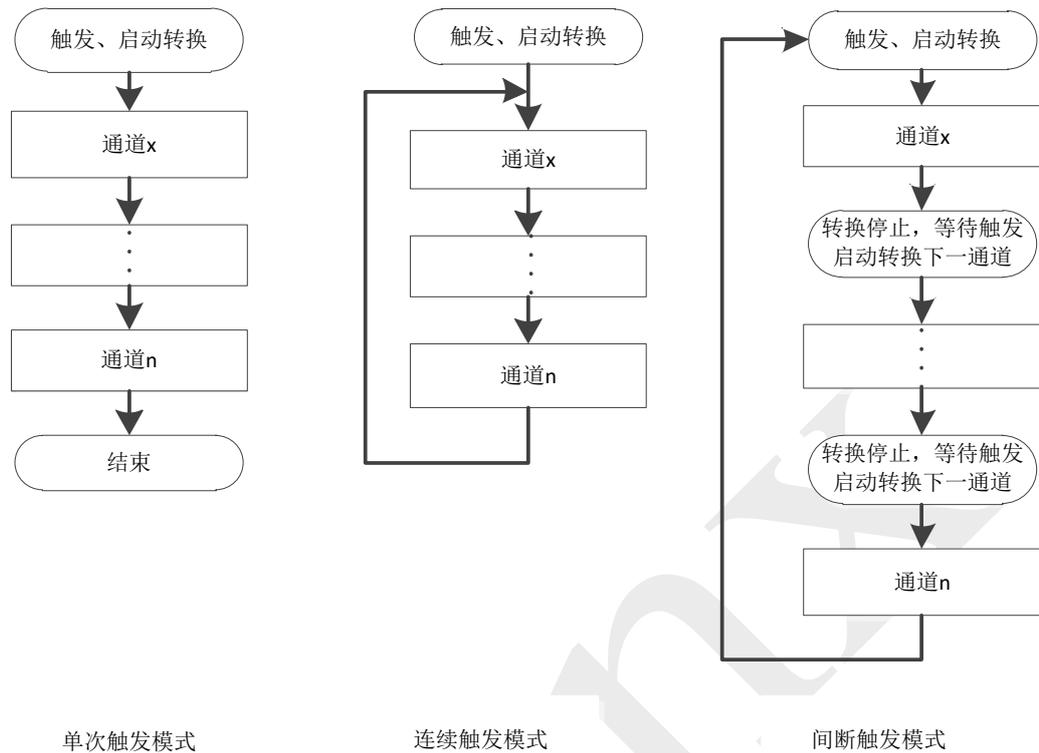
ADC 模块可根据应用情况调整采样时间，这样可根据输入电压源的输入阻抗调整转换速度。

ADC 会在一定 ADC 时钟周期内对输入电压进行采样，该采样时钟周期数由 ADC_SMPR 寄存器的 $SMP[3:0]$ 位进行设置。

每个 ADC 输入通道通过配置 ADC_SMPR 寄存器的 SMP 位来配置采样时间。

16.3.7 转换模式

ADC 控制器具有单次、连续、间断三种转换模式。

图 16-7 ADC 转换模式


单次转换模式

单次转换模式下，ADC 执行单次转换序列。当软件触发或外部硬件触发事件产生，ADC 对所选通道只进行一次转换。用户通过设置 ADC_CFGR1 寄存器中的 CONT[1:0]位为“00”，选择此模式。

启动转换方式：

- 软件触发，将 ADC_CR 寄存器中 ADSTART 位置 1，则立即启动转换；
- 硬件触发，由外部硬件触发源启动转换。

在单次转换序列中，每个通道转换完成后，ADC 执行以下操作：

- 转换数据存储于 16 位 ADC_DR 寄存器；
- 转换结束标志 (EOC) 置 1；
- 如果转换结束中断使能 (EOCIE 置 1)，则产生中断。

在所有通道转换序列完成后，ADC 执行以下操作：

- 转换序列结束 (EOS) 标志置 1；
- 如果转换序列结束中断使能 (EOSIE 置 1)，则产生中断。

在所有通道转换完成后，ADC 停止，等待下一次外部触发事件产生，或软件触发位 ADSTART 位置 1。

注意：当转换序列长度为1时，ADC只转换所选定的单个通道。

连续转换模式

连续转换模式下，ADC会对所选择的通道序列连续进行转换。当软件触发或外部硬件触发事件产生，ADC对所选通道序列进行一次转换。但与单次转换模式不同的是，在通道序列转换完成后，ADC会重新启动，自动执行相同的转换序列。用户通过设置ADC_CFGR1寄存器中的CONT[1:0]位为“01”，选择此模式。

启动转换方式：

- 软件触发，将ADC_CR寄存器中ADSTART位置1，则立即启动转换；
- 硬件触发，由外部硬件触发源启动转换。

在转换序列中，每个通道转换完成后，ADC执行以下操作：

- 转换数据存储于12位ADC_DR寄存器；
- 转换结束标志（EOC）置1；
- 如果转换结束中断使能（EOCIE置1），则产生中断。

在所有通道转换序列完成后，ADC执行以下操作：

- 转换序列结束（EOS）标志置1；
- 如果转换序列结束中断使能（EOSIE置1），则产生中断。

在转换序列所有的通道转换完成后，ADC会重新启动，自动执行相同的转换序列。

注意：当转换序列长度为1时，ADC只连续转换所选定的单个通道。

间断转换模式

将ADC_CFGR1寄存器中的CONT[1:0]位置为“10”，可启用间断转换模式。

在间断转换模式下，通道序列中的每个通道都需要通过硬件或软件事件触发才可以启动转换。相反，如果不启用间断转换模式，单个硬件或软件触发事件可以启动通道序列中的所有通道的转换。

示例：

- 如果CONT[1:0]位设置为“10”，待转换通道为0、3、7、10
 - 第一次触发：转换通道0，当转换完成时产生EOC事件；
 - 第二次触发：转换通道3，当转换完成时产生EOC事件；
 - 第三次触发：转换通道7，当转换完成时产生EOC事件；
 - 第四次触发：转换通道10，当转换完成时，同时产生EOC和EOS事件；

- 第五次触发：转换通道 0，当转换完成时产生 EOC 事件
- 第六次触发：转换通道 3，当转换完成时产生 EOC 事件
- ...
- 如果 CONT[1:0]不设置为“10”，待转换通道为 0、3、7、10
 - 第一次触发：转换整个通道序列：即依次转换通道 0、3、7、10。每个通道转换完成时都会生成 EOC 事件，最后一个通道转换完成还会生成 EOS 事件。
 - 任何后续触发事件都将重新启动整个通道序列转换。

16.3.8 启动转换

通过将 ADC_CR 寄存器的 ADSTART 位置 1，软件启动 ADC 转换。

ADSTART 位置 1 后，ADC 进行以下处理：

- 如果设置为软件触发模式 (EXTEN = 00)，则立即启动电压转换；
- 如果设置为外部硬件触发模式 (EXTEN ≠ 00)，则根据所设置的触发沿，在下一个触发事件产生后，启动电压转换。

ADSTART 位也用于指示 ADC 转换是否处于正在执行状态。当 ADSTART 位清零时，指示 ADC 处于空闲状态，此时可以重新配置 ADC。

在以下情况下，ADSTART 位由硬件清零：

- 在单次转换模式下，软件触发方式 (CONT[1:0]=00)：
转换序列结束时 (EOS=1)，此位清零。
- 在间隔转换模式下，软件触发方式 (CONT[1:0]=10、EXTEN=00)：
在每个通道转换结束时 (EOC=1)，此位清零。
- 在所有模式配置下 (CONT[1:0]=xx、EXTEN= xx)：
软件执行 ADSTP 流程时，此位清零。

注意： 在连续转换模式下 (CONT[1:0]=01)，由于转换序列会重新启动，因此当转换序列结束后 (EOS 位置 1)，ADSTART 位不会被硬件清零。

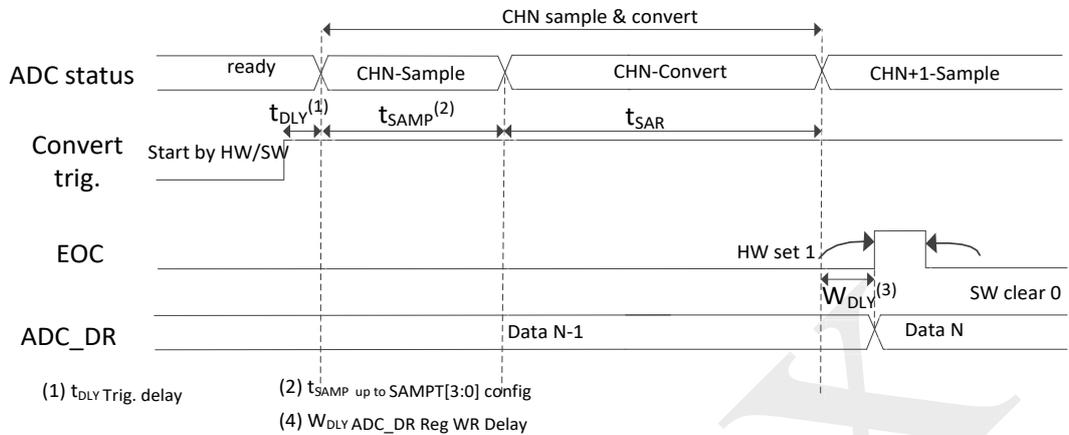
注意： 在单次转换模式下，当选择外部硬件触发方式时 (CONT[1:0]=00，且 EXTEN=00)，当转换序列完成后 (EOS 位置 1)，ADSTART 位不会被硬件清零。这样，软件无需重新将 ADSTART 位置 1，以确保不会错过下一次触发事件。

16.3.9 转换时间

ADC 进行模数转换包含电压采样及逐次逼近计算电压值 (SAR) 两个阶段，因

此模数转换时间（即从转换开始到转换结束的时间）是所配置的采样时间与逐次趋近（SAR）时间的总和。其中，逐次逼近时间（SAR）由 ADC 分辨率确定。

图 16-8 ADC 转换时序



16.3.10 停止转换

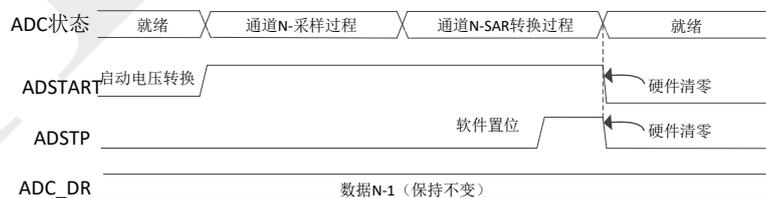
软件可以将 ADC_CR 寄存器中的 ADSTP 位置 1，以停止正在进行的转换。这样可以复位 ADC 操作，使 ADC 处于空闲状态，从而准备好进行新的操作。

当软件置位 ADSTP，任何正在进行的转换会被终止，且转换结果会被丢弃（ADC_DR 寄存器不会更新为当前转换结果）。

通道扫描序列也会终止，且复位，这意味着重启 ADC 将会重新开始新的转换序列。

一旦停止转换操作完成，ADSTART 位和 ADSTART 位均会被硬件清零。软件必须等待 ADSTART=0，然后才能开始进行新的转换。

图 16-9 停止正在进行的转换



16.3.11 转换触发模式及触发沿

可以通过软件或外部事件（定时器捕获事件）触发 ADC 进行通道转换。

- 如果 EXTEN[1:0] 控制位不等于“0b00”，则选择外部事件触发模式。此时如果所配置极性的触发事件产生，则会触发 ADC 转换。

当 ADSTART 置 1 时，触发选择立即生效。

- 如果 EXTEN[1:0] 控制位等于“0b00”，则选择软件触发模式。此时如果软件将 ADSTART 位置 1，则立即触发 ADC 转换。

当 ADC 正在处于转换过程中，外部硬件触发会被忽略。如果寄存器位 ADSTART = 0，则会忽略任何硬件触发事件。

下表展示了寄存器 EXTEN[1:0]与触发极性之间的对应关系。

表 16-2 触发极性

触发源	TRIGEN[1:0]
禁止触发检测（软件触发）	00
在上升沿检测	01
在下降沿检测	10
在上升沿和下降沿均检测	11

注意： 仅当 ADC 未进行转换 (ADSTART= 0) 时，才可以更改外部触发事件的极性。

EXTSEL[2:0]寄存器控制位用于选择具体哪一路外部硬件触发源（总共可选择 8 路外部触发源）。

下表展示了外部触发源列表。

如果设置为软件触发模式，则将 ADC_CR 寄存器中 ADSTART 位置 1 时，立即生成软件触发事件，启动 ADC 转换。

表 16-3 外部触发源

名称	源	EXTSEL[2:0]
TRG0	TIM1_TRGO2	000
TRG1	TIM1_CC4	001
TRG2	TIM2_TRGO	010
TRG3	PLAOUT1*	011
TRG4	TIM4_TRGO	100
TRG5	PLAOUT2*	101
TRG6	TIM6_TRGO	110
TRG7	EXTI line11	111

注意： 仅当 ADC 未进行转换 (ADSTART = 0) 时，才可以更改触发源选择。

*注：当选择 PLA 输出作为的外部触发源时，为避免毛刺干扰，PLA 内部自动将 DFF 作为 PLA 的输出，但需软件配置 PLA 选择 SYSCLK 作为 DFF 输入时钟。

16.3.12 转换结束、采用阶段结束

针对每次转换完成事件，ADC 都会给出转换结束标志（EOC）作为指示。

当转换完成，转换数据结果输出到 ADC_DR 寄存器后，ADC 立即将 ADC_ISR 寄存器中 EOC 标志置 1。此时如果 ADC_IER 寄存器的转换完成中断使能 EOCIE 位置 1，可产生转换完成中断。

EOC 标志可通过以下方式清零：

- 向 EOC 标志位写 1；
- 读取 ADC_DR 寄存器数据。

此中断用于使处理与 ADC 转换阶段同步进行。通常情况下，可在 ADC 转换阶段的隐藏时间中访问模拟复用器，这样在下次采样开始时便将模拟复用器配置完毕。

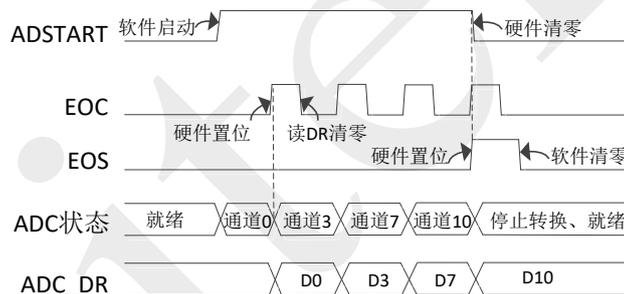
16.3.13 转换序列结束

当转换序列结束时，ADC 将 ADC_ISR 寄存器的 EOS 标志置 1 作为指示。

当转换序列的最后一个通道转换完成，从而在 ADC_DR 寄存器可得到相应转换结果时，ADC 立即将 ADC_ISR 寄存器中 EOS 标志置 1。此时如果 ADC_IER 寄存器的转换序列完成中断使能 EOSIE 位置 1，可产生转换序列完成中断。软件通过向 EOS 标志位写 1 清除此标志。

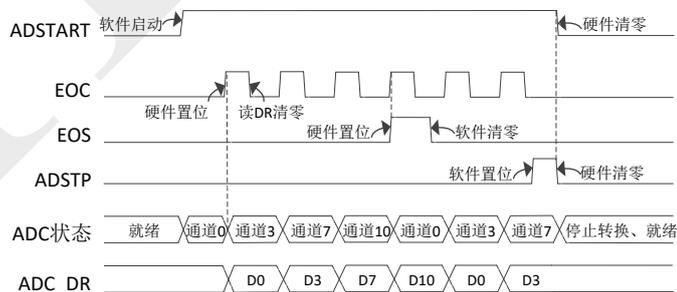
16.3.14 转换时序示例

图 16-10 单次转换模式，软件触发

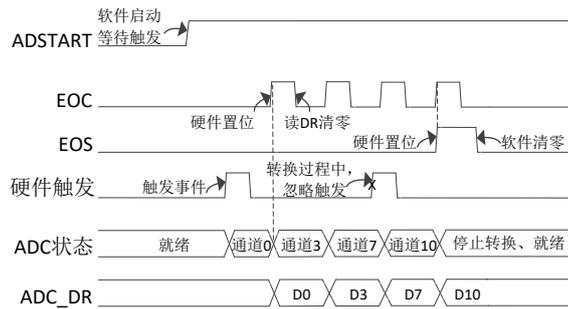


1. 转换通道选择 0、3、7、10，正向扫描。

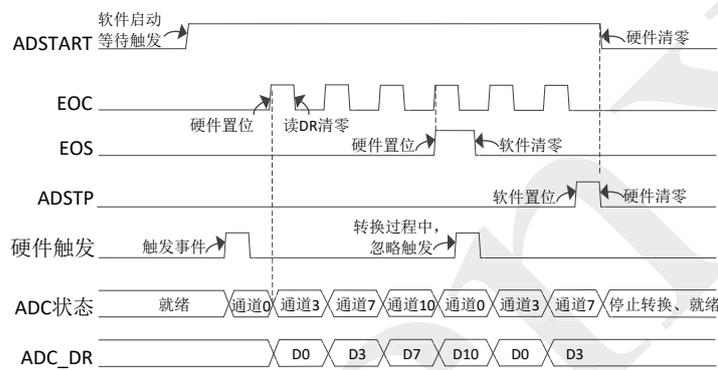
图 16-11 连续转换模式，软件触发



1. 转换通道选择 0、3、7、10，正向扫描，软件停止转换。

图 16-12 单次转换模式，硬件触发


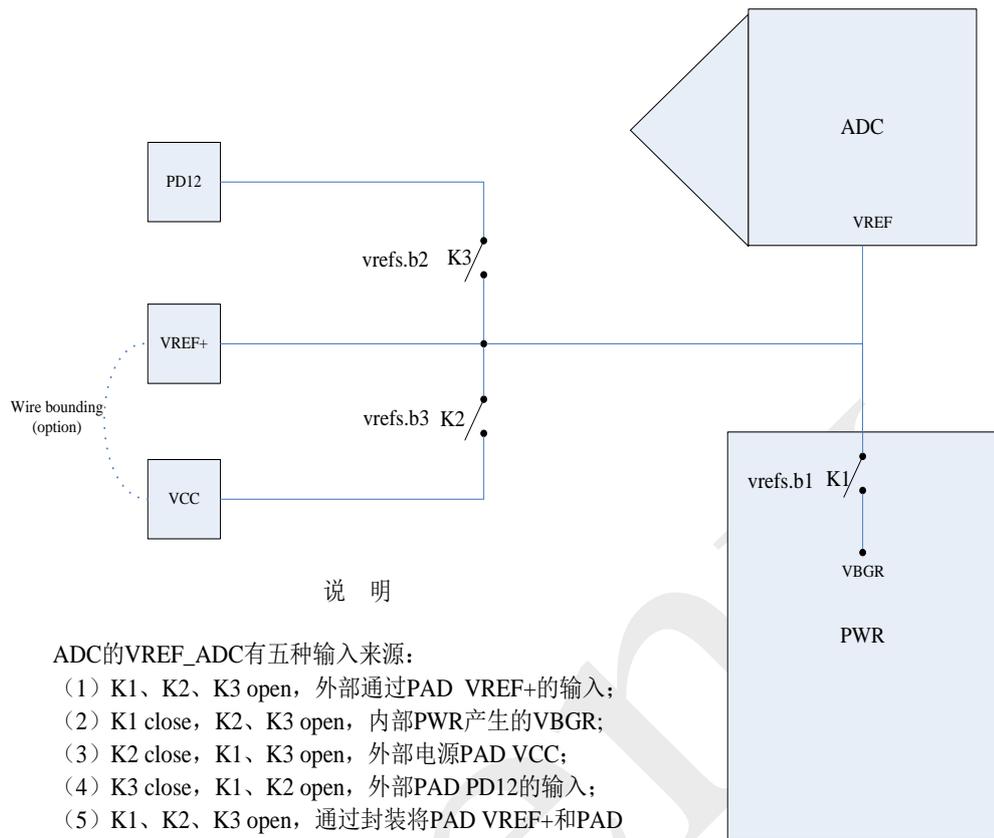
1. 转换通道选择 0、3、7、10，正向扫描。

图 16-13 连续转换，硬件触发


1. 转换通道选择 0、3、7、10，正向扫描。

16.3.15 ADC 参考源选择

本芯片为提到提供多种参考源选择,包括 VDD 电源、外部模拟输入参考和内部 V_{BGR} 。具体参考 ADC_CFGR1 寄存器的 VREFS 位进行选择,当 VREFS 位的 Bit1 为 1 即关闭 K1, V_{BGR} 被选择为 ADC 的参考源;当 VREFS 位的 Bit2 为 1 即关闭 K2, PAD12 外部输入被选择为 ADC 的参考源;当 VREFS 位的 Bit3 为 1 即关闭 K3, VCC 电源被选择为 ADC 的参考源。默认上电为外部 VREF+管脚作为 ADC 的参考源。应用需要保证 VREFS 位其中只能有 1 位使能,并保证使能时 VREF+管脚不做输入使用。

图 16-14 ADC 参考源选择


说 明

ADC的VREF_ADC有五种输入来源:

- (1) K1、K2、K3 open, 外部通过PAD VREF+的输入;
- (2) K1 close, K2、K3 open, 内部PWR产生的VBGR;
- (3) K2 close, K1、K3 open, 外部电源PAD VCC;
- (4) K3 close, K1、K2 open, 外部PAD PD12的输入;
- (5) K1、K2、K3 open, 通过封装将PAD VREF+和PAD VCC绑定在一起, 此种情况下, VREF只能有一种输入即VCC, 前面四种输入都失效;

另外, PWR产生的VBGR作为ADC的VREF输入时:

- (1) 其可同时输出到PAD VREF+, 给外部器件使用, 此时K1 close, K2、K3 open;
- (2) 其可同时输出到PAD PD12, 给外部器件使用, 此时K1、K3 close, K2 open;

16.4 ADC 数据管理

16.4.1 数据寄存器及数据对齐

每次转换结束时(此时 EOC 事件产生), 转换结果都会存储在位宽为 16 位的 ADC_DR 数据寄存器中。

寄存器 ADC_DR 内的数据格式取决于所配置的数据对齐方式以及 ADC 分辨率。

ADC_CFGR1 寄存器中的 ALIGN 位用于配置转换后存储在 ADC_DR 寄存器中数据的对齐方式。

转换数据可以配置为右对齐(ALIGN=0)或左对齐(ALIGN=1)方式, 如下图所示。

表 16-4 数据对齐方式和分辨率

ALIGN	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0				DR[11:0]											
1	DR[11:0]												0x0			

16.4.2 数据溢出

由于 ADC 所有通道的转换结果都会输出到同一个 ADC_DR 数据寄存器，因此如果之前转换的结果未被 CPU 或 DMA 及时读取，当新的转换完成而向 ADC_DR 数据寄存器输出转换结果时，会产生数据溢出事件。此时数据溢出标志(OVR)会置位。

当新的转换完成，如果此时 EOC 标志仍为“1”，ADC_ISR 寄存器的数据溢出 OVR 标志会置 1。此时如果 ADC_IER 寄存器的数据溢出中断使能 OVRIE 位置 1，可产生数据溢出中断。

当产生数据溢出事件，ADC 会继续保持工作状态并可以继续转换，除非通过软件将 ADC_CR 寄存器的 ADSTP 位置 1，从而停止转换并复位转换序列。

软件通过向 OVR 标志位写 1 清除此标志。

可以对 ADC_CFGR1 寄存器的 OVRMOD 位进行编程，从而配置当发生数据溢出事件时，继续保留 ADC_DR 数据寄存器中之前的转换结果，还是被新的转换数据覆盖。

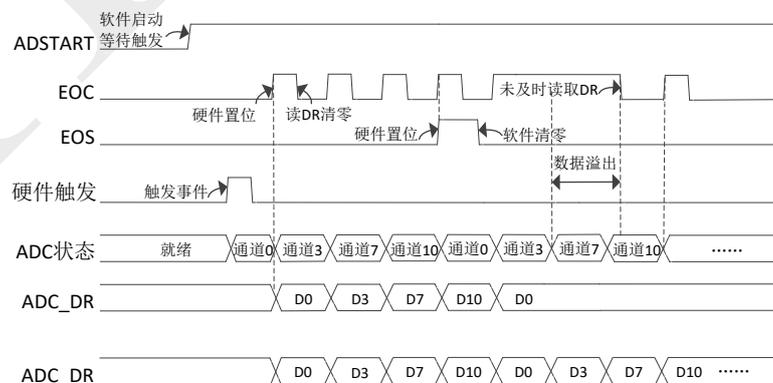
- OVRMOD 位清 0

当产生溢出事件，ADC_DR 数据寄存器中之前的转换结果会被保留，防止其被覆盖：即会保留原转换数据，并丢弃新的转换结果。如果 OVR 保持为 1，ADC 可以继续转换，但会丢弃后续转换数据。

- OVRMOD 位置 1

ADC_DR 数据寄存器会被新的转换结果所覆盖，从而丢弃之前未读取的数据。如果 OVR 保持为 1，ADC 可以继续转换，ADC_DR 寄存器始终存储最新转换得出的数据。

图 16-15 数据溢出处理示例



16.4.3 在不使用 DMA 情况下管理转换的数据序列

如果转换过程足够慢，则可以使用软件来处理转换序列。在这种情况下，软件必须使用 EOC 标志及其相关中断来处理各个通道的转换结果。每次转换完成

时，ADC_ISR 寄存器的 EOC 位会置 1，此时可以读取 ADC_DR 寄存器从而得到此次转换结果。

在这种情况下，ADC_CFGR1 寄存器中的 OVRMOD 位应配置为 0，从而将数据溢出事件作为错误进行管理。

16.4.4 在不使用 DMA 且不发生溢出的情况下管理转换的数据

当 ADC 转换单一或多通道，而且无需在每次转换后都读取转换结果时，适宜采用以下处理方式：OVRMOD 寄存器位必须配置为 1，数据溢出 OVR 标志应被软件忽略。在这种配置下，数据溢出事件不会阻止 ADC 继续进行转换，ADC_DR 寄存器始终存储最新的转换数据。

16.4.5 使用 DMA 管理转换的数据

由于 ADC 所有通道的转换结果都会输出到同一个 ADC_DR 数据寄存器，因此在多通道转换序列情况下，采用 DMA 处理转换数据可以提高效率。这样可以避免存储在 ADC_DR 寄存器的转换结果丢失。

当 DMA 模式已使能（ADC_CFGR1 寄存器中的 DMAEN 位置 1），在每个通道转换完成后，ADC 都会生成 DMA 请求。此时，DMA 控制器会将 ADC_DR 寄存器的转换结果传输到软件配置的目标地址。

注意： ADC_CFG1 寄存器中的 DMAEN 位须在 ADC 校准阶段完成后进行设置。

尽管如此，如果因 DMA 控制器无法及时处理 DMA 传输请求而导致数据溢出事件产生 (OVR=1)，ADC 会停止生成 DMA 请求，新转换得到的数据不会通过 DMA 进行传输。这意味着可将传输到目标地址（如 RAM）的所有数据都视为有效数据。

根据 OVRMOD 寄存器位的配置，ADC_DR 寄存器的转换结果可以保留或被新转换数据覆盖。

DMA 传输请求会禁止，直至软件将数据溢出标志位 OVR 清零。

16.5 低功耗模式

16.5.1 自动等待模式

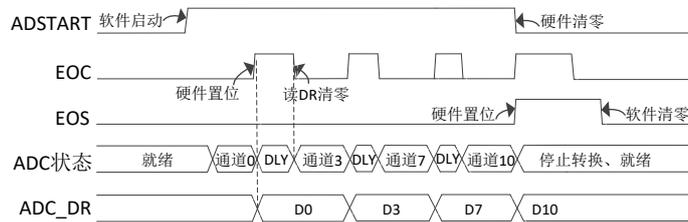
自动等待模式可用于简化软件设计，并可以优化采用低频时钟应用的性能（此类应用可能存在 ADC 数据溢出的风险）。

当 ADC_CFGR1 寄存器中的 WAIT 位置 1，使能等待模式转换，仅当之前的转换数据进行处理，即 ADC_DR 寄存器已读取或者 EOC 位已清零后，ADC 才可以开始新的转换。

自动等待模式可以自动调整 ADC 的速度，使其适应系统读取转换数据的速度。

注意： 在 ADC 转换进行时或在读访问之前的等待时间内发生的硬件触发事件会被忽略。

图 16-16 等待模式转换（单次、软件触发）



1. 转换通道选择 0、3、7、10，正向扫描。

16.6 模拟看门狗

ADC 具有一个模拟看门狗 AWDG, 用于监控当前转换通道电压是否在配置的电压范围之内（窗口）。

16.6.1 模拟看门狗简介

拟看门狗通过设置 ADC_CFGR1 寄存器的 AWD1EN 位来监控所当前转换的通道电压。

模拟看门狗监控通道转换电压的方式是对完整的 12 位原始转换数据与所设置阈值进行比较（左对齐）。因此如果 ADC 分辨率小于 12 位（由 RES[1:0]位来配置），则模拟看门狗监控电压阈值的 LSB 必须保持清零。

下表对所有可能的分辨率，模拟看门狗进行监控电压比较方式进行说明。

表 16-5 模拟看门狗监测数据格式

模拟看门狗比较对象		注释
原始转换数据，左对齐(1)	阈值	
DATA[11:0]	LT1[11:0] 和 HT1[11:0]	-

1. 进行任何对齐计算之前，会对原始转换数据进行看门狗比较。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 ADC 模拟看门狗电压监控状态标志位 (AWD1) 置 1。此时如果 ADC_IER 寄存器的 AWD1xIE 中断使能位置 1，则会产生模拟看门狗监控电压中断。模拟看门狗监控电压阈值在 ADC_AWDxTR 寄存器的 HT1[11:0] 和 LT1[11:0] 位进行设置。

软件向 AWDG 状态标志位写 1，可清零此标志。

16.6.2 ADC_AWDGx_OUT 信号输出生成

每个模拟看门狗关联一个内部硬件信号 ADC_AWDGx_OUT (x 为看门狗序号)，此信号可以直接连接到一些片上定时器 Timer(参看定时器 Timer 章节如何选择 ADC_AWDGx_OUT 信号作为 ETR)的 ETR 输入(外部触发)。

当关联的模拟看门狗被使能后，ADC_AWDGx_OUT 被激活。

- 当被监控通道的转换电压超出所设置的阈值范围时，ADC_AWDGx_OUT 会被置高。
- 当下一个被监控通道转换电压未超出所设置的阈值范围时，ADC_AWDGx_OUT 会被复位。如果下一次监控通道的转换电压仍超出所设置的阈值范围，ADC_AWDGx_OUT 会保持高的状态。
- 当禁止 ADC (设置 ADDIS 为 1)，ADC_AWDGx_OUT 也会被复位。注意停止转换 (ADSTP 设定为 1) 时，可能会清除 ADC_AWDGx_OUT 的状态。

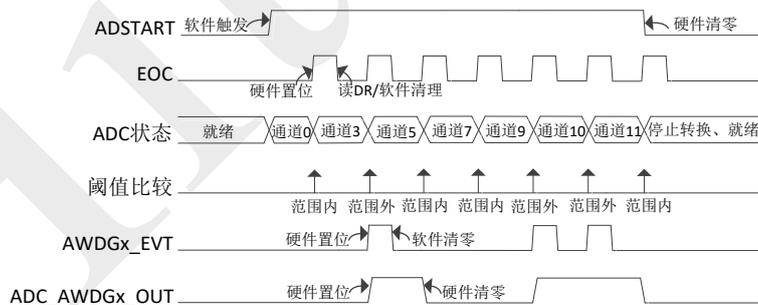
WDx_EVT 标志由硬件置位，软件清零。此标志不会影响 ADC_AWDGx_OUT 信号的产生 (例如，当 AWDx 标志未被软件清零，仍保持为 1 时，ADC_AWDGx_OUT 也可以翻转)。

ADC_AWDGx_OUT 信号由 ADC_CLK 域产生，即使 APB 时钟停止也可以产生此信号。

每次 ADC 转换完成后进行 ADC 模拟看门狗监控电压比较。ADC_AWDGx_OUT 信号上升沿和下降沿变化在监控电压比较后的两个 ADC_CLK 时钟周期产生。

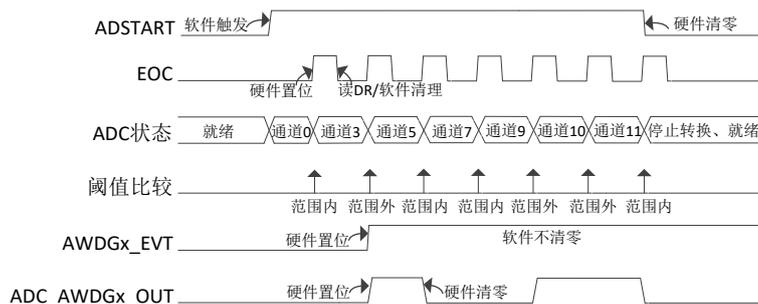
由于 ADC_AWDGx_OUT 信号由 ADC_CLK 时钟域产生，而 AWDGx_EVT 标志由 APB 时钟域产生，这些信号的上升沿不同步。

图 16-17 ADC_WDx_OUT 信号生成



1. 转换通道单次转换，软件触发。

图 16-18 ADC_AWDGx_OUT 信号生成 (软件不清零 AWDGx_EVT 标志)



1. 转换通道选择 0、3、5、7、9、10、11，单次转换，软件触发。

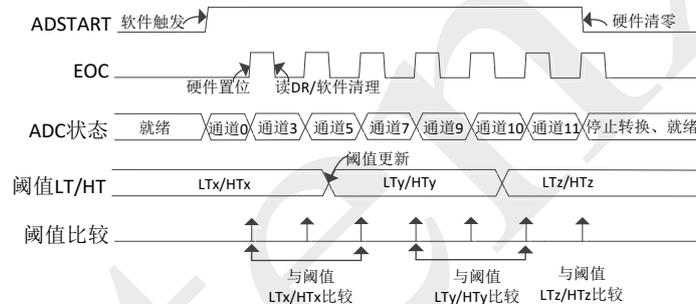
模拟看门狗电压监控阈值 LT1[11:0]、HT1[11:0]可以在 ADC 模数转换过程中(在 ADC 转换开始和转换结束之间的状态)进行改变。

如果在 ADC 模数转换过程中改变了监测通道的电压阈值 LT1、HT1，则模拟看门狗会屏蔽本次电压监控比较。当开始新的转换时，将会取消以上屏蔽，新的 ADC 看门狗阈值将会应用于下一次的 ADC 转换。

ADC 会在每一次转换结束时进行看门狗电压监控比较。在阈值更新过程中，如果当前 ADC 转换数据超出了新的阈值区间，将不会产生中断或者 ADC_AWDGx_OUT 信号。中断或者 ADC_AWDGx_OUT 信号仅在阈值更新之后的 ADC 转换结束时发生。

如果 ADC_AWDGx_OUT 信号已经产生，则所设置的新阈值不会影响 ADC_AWDGx_OUT 信号。

图 16-19 模拟看门狗阈值更新



1. 转换通道选择 0、3、5、7、9、10、11，单次转换，软件触发。

16.7 温度传感器和内部参考电压

温度传感器用于测量器件的结温 (TJ)。温度传感器在内部连接到 ADC VIN[56] 输入通道，该通道用于将传感器输出电压转换为数字值。温度传感器输入通道的采样时间必须大于数据手册中指定的最小 TS_temp 值。不使用时可将温度传感器置于掉电模式。

内部参考电压 (V_{BGR}) 为 ADC 和比较器提供了一个稳定的 (带隙基准) 电压输出。

V_{BGR} 内部连接到 ADC_VIN[57] 输入通道。由于制造工艺的差异，在生产测试期间对每个器件 V_{BGR} 的精确电压单独进行了测量，并存储于系统存储区。

温度传感器、内部参考电压与 ADC 之间连接结构如图 18-25 所示。

ADC 中使能 ADC_VIN[56] (温度传感器) 的转换须将 VTSEN 位置 1，使能 ADC_VIN[57] (V_{BGR}) 的转换须将 VBGREN 位置 1 并将 CHBEN 位置 1。

温度传感器的输出电压随温度线性变化。由于制造工艺的差异，该线性的偏移量取决于各个芯片 (芯片之间的温度变化可达 45°C)。

未校准的内部温度传感器更适用于对温度变化而非绝对温度进行测量。为提高温度传感器测量的准确性，在生产过程中对每个器件的温度传感器进行了校准，其校准值存储于系统存储器。

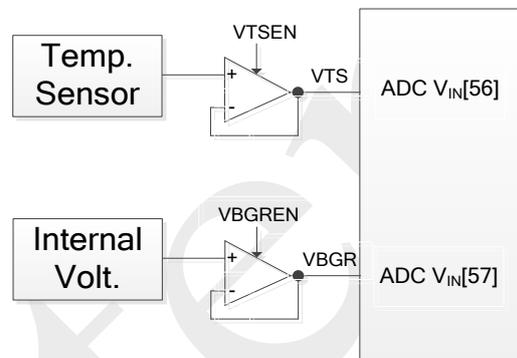
在制造过程中，会将温度传感器的校准数据和内部参考电压存储在系统存储区。在应用中，可以使用这些数据以提高温度传感器或内部参考电压的精度。其它相关信息，请参见数据手册。

温度传感器主要特性：

- 支持的温度范围：-40°C 到 105°C。
- 线性度：最高 ±2°C，精度取决于校准情况。

注意：当使用温度传感器或 VBGR 时需要使能 BGR。

图 16-20 温度传感器和 VBGR 通道结构图



利用温度传感器读取温度步骤：

- 1) 选择 ADC_VIN[56] 输入通道；
- 2) 选择合适的采样时间 (TS_temp)；
- 3) 将 ADC_CFGR2 寄存器的 VTSEN 位置 1，将温度传感器唤醒，并等待其稳定（稳定时间 t_{START}）；
- 4) 将 ADC_CR 寄存器的 ADSTART 位置 1（或通过外部硬件触发），启动 ADC 转换；
- 5) 读取 ADC_DR 寄存器中生成的 VTS 转换数据；
- 6) 使用以下公式计算温度。

$$\text{Temperature(in } ^\circ\text{C)} = \frac{TS_CAL2_TEMP - TS_CAL1_TEMP}{TS_CAL2 - TS_CAL1} \times (TS_DATA - TS_CAL1) + TS_CAL1_TEMP$$

其中：

- TS_CAL2 是通过 TS_CAL2_TEMP 获得的温度传感器校准值(TS_CAL2 值参看数据手册)
- TS_CAL1 是通过 TS_CAL1_TEMP 获得的温度传感器校准值(TS_CAL1

值参看数据手册)

- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值

更多关于 TS_CAL1 和 TS_CAL2 校准值的信息，请参见数据手册。

注意： 温度传感器唤醒需要一个启动时间，之后其才能输出正确的 VTS。ADC 在上电后同样需要一个启动时间，为尽可能缩短延迟时间，应同时将 ADEN 位和 VTSEN 位置 1。

使用内部参考电压 V_{BGR} 计算实际的 VREF+ 电压

ADC 的参考电压 VREF+ 可能会有变化，或无法获得准确值。在制造过程中在 VREF+=3V 的条件下测试得到的 ADC 内部参考电压 (V_{BGR}) 及其校准数据可用于评估实际的 VREF+ 电压水平。

由以下公式可得出为器件供电的实际参考电压 VREF+：

$$VREF+ = 3V \times \frac{V_{BGR_CAL}}{V_{BGR_DATA}}$$

其中：

- V_{BGR_CAL} 是 V_{BGR} 的校准值；
- V_{BGR_DATA} 是由 ADC 转换得到的实际 V_{BGR} 输出值。

将电源相关的 ADC 测量值转换为绝对电压值

根据采集到的通道电压与电源电压的比例关系，ADC 将通道电压转换为数字值。对于大部分应用，需要将该比例关系转换成与 VREF+ 无关的电压。

对于 VREF+ 已知、ADC 转换值进行了右对齐的应用，可使用以下公式计算得到该通道电压的绝对值：

$$V_{CHANNELX} = \frac{VREF+}{FULL_SCALE} \times ADC_DATAx$$

对于 VREF+ 值未知的应用，必须使用内部参考电压 V_{BGR}，因此 VREF+ 可替换为使用内部参考电压 V_{BGR} 计算实际的 VREF+ 电压的表达式，从而得出以下公式：

$$V_{CHANNELX} = \frac{3V \times V_{BGR_CAL} \times ADC_DATAx}{V_{BGR_DATA} \times FULL_SCALE}$$

其中：

- V_{BGR_CAL} 是 V_{BGR} 校准值；
- ADC_DATAx 是由 ADC 在通道 x 上测得的值（右对齐）；
- V_{BGR_DATA} 是由 ADC 转换得到的实际 V_{BGR} 输出值；

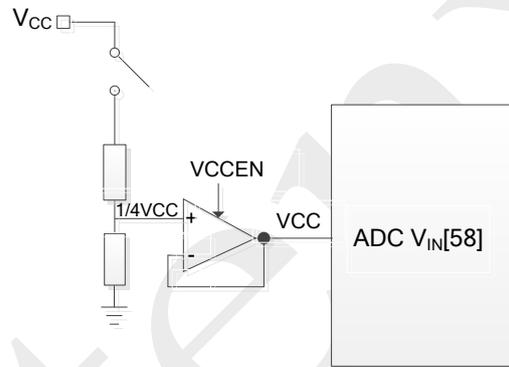
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 12 位，该值为 $2^{12} - 1 = 4095$ ；如果分辨率为 8 位，该值为 $2^8 - 1 = 255$ 。

注意： 如果进行 ADC 测量时使用的数据输出格式并非 12 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。

16.8 电源电压监测

ADC_CFGR2 寄存器的 VCC 位可用于测量 VCC 引脚上的电源电压。VCCEN 位置 1 时，会自动使能此桥接分压器，以将 VCC 连接到 ADC_VIN[58] 输入通道。因此，ADC 采集转换得出的数字值是 VCC 电压的四分之一。为防止出现额外的功耗，建议仅在执行 ADC 转换时才使能桥接分压器。

图 16-21 VCC 通道采集结构图



16.9 ADC 中断

发生下列任一事件均可生成中断：

- 校准结束（EOCAL 标志）
- ADC 上电，ADC 就绪后（ADRDY 标志）
- 任何通道转换结束（EOC 标志）
- 转换序列结束（EOS 标志）
- 模拟看门狗电压监控超出所设置电压阈值（AWD1 标志）
- 发生数据溢出时（OVR 标志）

表 16-6 ADC 中断

中断事件	事件标志	使能控制位
校准结束	EOCAL	EOCALIE
ADC 就绪	ADRDY	ADRDYIE
转换结束	EOC	EOCIE
转换序列结束	EOS	EOSIE
看门狗状态位置	AWD1	AWD1IE

数据溢出	OVR	OVRIE
------	-----	-------

16.10 ADC 寄存器

16.10.1 ADC 控制寄存器 (ADC_CR)

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	
ADCAL	Res.															
rs																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.												ADSTP	Res.	ADSTAR T	ADDIS	ADEN
												rs		rs	rs	rs

位/位域	名称	描述
31	ADCAL	<p>ADC校准</p> <p>该位由软件置 1, 用于启动ADC校准。校准完成后, 由硬件自动清零。</p> <p>0: 校准已完成或未处于校准态;</p> <p>1: 写入1启动ADC校准。读取值为1表示正在校准过程中。</p> <p><i>注意: 仅当ADEN置1, ADC就绪后 (ADRDY 标志置1), 且 ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0, 才允许通过软件将ADCAL位置 1。</i></p>
30:5	保留	必须保持复位值。
4	ADSTP	<p>ADC停止转换。</p> <p>该位由软件置 1, 用于停止和丢弃正在进行的转换 (ADSTP 命令)。</p> <p>当转换已有效丢弃、并且 ADC 已准备好接收新的开始转换命令时, 会通过硬件将该位清零。</p> <p>0: 当前未执行 ADC 停止转换命令;</p> <p>1: 写入1可停止 ADC。读取值为 1 表示正在执行 ADSTP 命令。</p> <p><i>注意: 仅当 ADSTART=1 且 ADDIS=0 时 (ADC 已使能、可能正在进行转换、并且没有任何待处理的禁止 ADC 的请求), 才允许通过软件将 ADSTP 置 1。</i></p>
3	保留	必须保持复位值。
2	ADSTART	<p>ADC 开始转换。</p> <p>此位由软件置 1, 用于开始 ADC 转换。根据EXTEN [1:0] 配</p>

置位的值，可以立即开始转换（软件触发配置），也可以在发生硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

- 在单次转换模式下，如果选择了软件触发 (EXTEN=00)：出现转换序列结束 (EOS) 标志时清零。
- 在不连续转换模式下，如果选择了软件触发 (EXTEN=00)：出现转换结束(EOC) 标志时清零。
- 在所有其他情况下：执行完 ADSTP 命令后，件将 ADSTP 位清零的同时清零。由硬

0：当前未进行 ADC 转换；

1：写入1可开始 ADC。读取值为1表示 ADC 正在工作，可能正在进行转换。

注意：仅当 ADEN=1 且 ADDIS=0 时 (ADC 已使能，并且没有任何待处理的禁止 ADC 的请求)，才允许通过软件将 ADSTART 置 1。

1 ADDIS

禁止ADC

该位通过软件置 1，用于禁止 ADC 并使其进入掉电状态 (OFF 状态)。

ADC 已有效禁止后，会立即通过硬件将该位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令；

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注意：仅当 ADEN=1 且 ADSTART=0 时 (这可确保当前未进行任何转换)，才允许通过软件将 ADDIS 置 1。

0 ADEN

使能ADC

该位通过软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对该位清零。

0：禁止ADC (OFF 状态)；

1：写入1使能ADC。

注意：仅当 ADC_CR 寄存器的所有位均为 0 时 (ADCAL=0、ADSTP=0、ADSTART=0、ADDIS=0 且 ADEN=0)，才允许通过软件将 ADEN 位置 1。

16.10.2 ADC 配置寄存器 1 (ADC_CFGR1)

偏移地址：0x04

复位值：0x0090 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								VREFS			VBGS		EXTSEL[2:0]		
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WAIT	CONT[1:0]		OVRMOD	EXTEN[1:0]		Res.			ALIGN	Res.			AWD1EN	DMAEN
	rw	rw	rw	rw	rw	rw				rw				rw	rw

位/位域	名称	描述
30:24	保留	必须保持复位值。
23:21	VREFS	ADC参考电压选择，详见18.3.15章节 100: VCC 010: 输入管脚PD12（PD12需要配置为模拟模式） 001: VBGO 000: 都不选 <i>注意：当选择使用内部VBGO作为ADC参考源时，使能ADC后需要100us的时间用于VBGR的稳定，然后才能启动ADC采样。 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>
20:19	VBGS	ADC的VBGO参考电压选择 00: 预留 01: 2.5V 10: 3V 11: 4V <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>
18:16	EXTSEL[2:0]	外部触发选择 这些位可选择用于触发转换开始的外部事件。 000: TRG0 001: TRG1 010: TRG2 011: TRG3 100: TRG4 101: TRG5 110: TRG6 111: TRG7 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i>
15	保留	必须保持复位值。
14	WAIT	自动等待模式 此位由软件置1和清零，用于使能/禁止等待转换模式。 0: 禁止等待转换模式； 1: 使能等待转换模式。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>

13:12	CONT[1:0]	转换模式选择 这些位由软件置1和清零，用于选择转换模式。 00：单次转换模式； 01：连续转换模式； 10：不连续转换模式； 其它：保留。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i>
11	OVRMOD	数据溢出管理模式 该位通过软件进行置 1 和清零，并用于配置数据溢出的管理方式。 0：如果检测到溢出，ADC_DR 寄存器会保留原有数据； 1：如果检测到溢出，ADC_DR 寄存器会被上一转换结果覆盖。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>
10:9	EXTEN[1:0]	外部触发使能和极性选择 这些位由软件置1和清零，用于选择外部触发极性并使能触发。 00：禁止硬件触发检测（可通过软件开始转换）； 01：在上升沿执行硬件触发检测； 10：在下降沿执行硬件触发检测； 11：在上升沿和下降沿都执行硬件触发检测。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i>
8:6	保留	必须保持复位值。
5	ALIGN	数据对齐模式 此位由软件置 1 和清零，用于选择右对齐或左对齐。 0：右对齐； 1：左对齐。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>
4:3	保留	必须保持复位值。
1	AWD1EN	当前转换通道模拟看门狗功能使能 0：禁用 1：使能 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i>
0	DMAEN	直接存储器访问使能

此位由软件置1和清零，用于使能 DMA 请求的生成。这样便可使用 DMA 控制器自动管理转换的数据。

0: 禁止 DMA;

1: 使能 DMA。

注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

16.10.3 ADC 配置寄存器 2 (ADC_CFGR2)

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
CKMODE[1:0]		Res.		PRESC[3:0]				Res.				VCCEN	VTSEN	VBGREN	
rw	rw			rw	rw	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
Res.														CHBEN	
															rw

位/位域	名称	描述
31:30	CKMODE[1:0]	<p>ADC时钟源</p> <p>此位由软件置 1 和清零，选择为模拟 ADC 提供的时钟。</p> <p>00: PCLK（同步时钟模式）。仅当 PCLK 的时钟占空比为 50% 时，才必须使能此配置（必须绕过RCC 中配置的 APB 预分频系数，并且系统时钟占空比必须在 50% 以内）。</p> <p>01: PCLK/2（同步时钟模式）</p> <p>10: PCLK/4（同步时钟模式）</p> <p>11: ADCCLK（异步时钟模式），在产品级生成（请参见 RCC 部分）在所有异步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。</p> <p><i>注意：仅当 ADC 已禁止时（ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。</i></p>
29:28	保留	必须保持复位值。
27:24	PRESC[3:0]	<p>ADC 预分频系数</p> <p>此位由软件置1和清零，用于选择 ADC 的时钟频率。该时钟为所有ADC所共用。</p> <p>0000: 输入 ADC 时钟未分频</p> <p>0001: 输入 ADC 时钟 2 分频</p> <p>0010: 输入 ADC 时钟 4 分频</p> <p>0011: 输入 ADC 时钟 6 分频</p> <p>0100: 输入 ADC 时钟 8 分频</p> <p>0101: 输入 ADC 时钟 10 分频</p> <p>0110: 输入 ADC 时钟 12 分频</p>

		0111: 输入 ADC 时钟 16 分频 1000: 输入 ADC 时钟 32 分频 1001: 输入 ADC 时钟 64 分频 1010: 输入 ADC 时钟 128 分频 1011: 输入 ADC 时钟 256 分频 其它: 保留 <i>注意: 仅当 ADC 已禁止时(ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0), 才允许通过软件对这些位执行写操作。</i>
23:19	保留	必须保持复位值。
18	VCCEN	VCC通道使能 此位由软件置1和清零, 用于使能/禁止VCC通道。 0: 禁止VCC通道; 1: 使能VCC通道。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对此位执行写操作。</i>
17	VTSEN	温度传感器使能 此位由软件置1和清零, 用于使能/禁止温度传感器。 0: 禁止温度传感器; 1: 使能温度传感器。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对此位执行写操作。</i>
16	VBGREN	VBGR通道输入使能 此位由软件置1和清零, 用于使能/禁止 VBGR 输入。 0: 禁止 VBGR输入; 1: 使能 VBGR输入。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对此位执行写操作。</i> 当ADC START前, 先配置VREFS=VBGO, 并选择合适的VBGS。
15:1	保留	必须保持复位值。
0	CHBEN	通道输入BUFFER使能 0: 禁止通道输入BUFFER。 1: 使能通道输入BUFFER。 <i>注意: 仅当 START=0 时 (这可确保当前未进行任何转换), 才允许通过软件对此位执行写操作。</i> 当使能通道输入BUFFER, 外部输入信号通过放大器后与ADC相连, 用于高阻信号。使用BUFFER功能, 最大速率200kbps。 以下几种情况需要打开BUFFER功能: 1) 外部驱动很弱信号

2) 内部TS、BGR、AVCC信号的测量。

16.10.4 ADC 中断和状态寄存器 (ADC_ISR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				EOCAL	Res.		Res.	AWD1	Res.		OVR	EOS	EOC	Res.	ADRDY
				rc_wl				rc_wl			rc_wl	rc_wl	rc_wl		rc_wl

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	EOCAL	校准结束标志 校准完成时，该位由硬件置 1。通过软件写入 1 可将该位清零。 0：校准未完成； 1：校准已完成。
10:8	保留	必须保持复位值。
7	AWD1	ADC看门狗事件标志 当转换电压超出在 ADC_AWD1TR 寄存器中编程的阈值范围时，硬件会将该位置 1。通过软件写入 1 可将该位清零。 0：未发生模拟看门狗事件（或标志事件已通过软件确认并清零）； 1：发生模拟看门狗事件。
6:5	保留	必须保持复位值。
4	OVR	ADC溢出 该位在发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将该位清零。 0：未发生溢出事件（或标志事件已通过软件确认并清零）； 1：发生溢出。
3	EOS	序列结束标志 在由 ADC_CHSELR 寄存器选择的一系列通道转换结束时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。 0：转换序列未完成（或标志事件已通过软件确认并清零）； 1：转换序列已完成。
2	EOC	转换结束标志 当通道的每次转换结束，新数据结果出现在 ADC_DR 寄存器

时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取 ADC_DR 寄存器都可将该位清零。

0: 通道转换未完成（或标志事件已通过软件确认并清零）；

1: 通道转换已完成。

1	保留	必须保持复位值。
0	ADRDY	<p>ADC就绪</p> <p>ADC 使能后(位 ADEN=1)以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。</p> <p>0: ADC 未准备好开始转换（或标志事件已通过软件确认并清零）；</p> <p>1: ADC 已准备好开始转换。</p>

16.10.5 ADC 中断使能寄存器 (ADC_IER)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Res.																	
Res.																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.				EOCALIE	Res.			AWD1IE	Res.				OVR1E	EOS1E	EOC1E	Res.	ADRDYIE
				rw				rw					rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	EOCALIE	<p>校准结束中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止校准结束中断。</p> <p>0: 禁止校准结束中断；</p> <p>1: 使能校准结束中断。</p> <p><i>注意： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p>
10:8	保留	必须保持复位值。
7	AWD1IE	<p>模拟看门狗 1 中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止模拟看门狗1中断。</p> <p>0: 禁止模拟看门狗1中断；</p> <p>1: 使能模拟看门狗1中断。</p> <p><i>注意： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p>
6:5	保留	必须保持复位值。

- | | | |
|---|---------|--|
| 4 | OVRIE | <p>溢出中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止溢出中断。</p> <p>0: 禁止溢出中断；</p> <p>1: 使能溢出中断，OVR 位置 1 时产生中断。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p> |
| 3 | EOSIE | <p>转换序列结束中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止转换序列结束中断。</p> <p>0: 禁止 EOS 中断；</p> <p>1: 使能 EOS 中断，EOS 位置 1 时产生中断。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p> |
| 2 | EOCIE | <p>转换结束中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止转换结束中断。</p> <p>0: 禁止 EOC 中断；</p> <p>1: 使能 EOC 中断，EOC 位置 1 时产生中断。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p> |
| 1 | 保留 | <p>必须保持复位值。</p> |
| 0 | ADRDYIE | <p>ADC 就绪中断使能</p> <p>此位由软件置 1 和清零，用于使能/禁止 ADC 就绪中断。</p> <p>0: 禁止 ADRDY 中断；</p> <p>1: 使能 ADRDY 中断。ADRDY 位置 1 时产生中断。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。</i></p> |

16.10.6 ADC 采样时间寄存器 (ADC_SMPR)

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													SMP[3:0]		
rw													rw rw rw rw		

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	SMP[3:0]	<p>采样时间选择</p> <p>这些位由软件写入，用于选择应用于所有通道的采样时间。</p>

- 0000: 2.5 个 ADC 时钟周期
- 0001: 3.5 个 ADC 时钟周期
- 0010: 7.5 个 ADC 时钟周期
- 0011: 12.5 个 ADC 时钟周期
- 0100: 19.5 个 ADC 时钟周期
- 0101: 39.5 个 ADC 时钟周期
- 0110: 79.5 个 ADC 时钟周期
- 0111: 119.5 个 ADC 时钟周期
- 1000: 159.5 个 ADC 时钟周期
- 1001: 199.5 个 ADC 时钟周期
- 1010: 239.5 个 ADC 时钟周期
- 其它: 3.5个 ADC 时钟周期

注意: 仅当 $ADSTART=0$ 时 (这可确保当前未进行任何转换), 才允许通过软件对这些位执行写操作。

16.10.7 ADC 通道选择寄存器 1 (ADC_CHSELR1)

偏移地址: 0x1C

复位值: 0x3F3F3F3F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		SQ4[5:0]						Res.		SQ3[5:0]					
		rw	rw	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
Res.		SQ2[5:0]						Res.		SQ1[5:0]					
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:24	SQ4[5:0]	序列的第 4 次转换 这些位可通过软件定义哪个通道 (CH0...CH58) 纳入转换序列的第 4 次转换。0b111111 表明序列结束。 当低的转换序列通道被设定为 0b111111 (序列结束) 时, 这些位将会被忽略。 000000: CH0 000001: CH1 ... 101001: CH58 111111: 没有通道被选择 (序列结束)
23:22	保留	必须保持复位值。
21:16	SQ3[5:0]	序列的第 3 次转换

注意: 仅当 $ADSTART=0$ 时 (这可确保当前未进行任何转换), 才允许通过软件对这些位执行写操作。

这些位可通过软件定义哪个通道（CH0…CH58）纳入转换序列的第3次转换。0b111111 表明序列结束。

当低的转换序列通道被设定为0b111111(序列结束)时，这些位将会被忽略。

通道选择请参考 SQ4[5:0] 的说明。

注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

15:14	保留	必须保持复位值。
13:8	SQ2[5:0]	<p>序列的第 2 次转换</p> <p>这些位可通过软件定义哪个通道（CH0…CH58）纳入转换序列的第2次转换。0b111111表明序列结束。</p> <p>当低的转换序列通道被设定为0b111111(序列结束)时，这些位将会被忽略。</p> <p>通道选择请参考 SQ4[5:0] 的说明。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i></p>
7:6	保留	必须保持复位值。
5:0	SQ1[5:0]	<p>序列的第 1次转换</p> <p>这些位可通过软件定义哪个通道（CH0…CH58）纳入转换序列的第1次转换。0b111111表明序列结束。</p> <p>当低的转换序列通道被设定为0b111111(序列结束)时，这些位将会被忽略。</p> <p>通道选择请参考 SQ4[5:0] 的说明。</p> <p><i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i></p>

16.10.8 ADC 通道选择寄存器 2 (ADC_CHSELR2)

偏移地址：0x20

复位值：0x3F3F3F3F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		SQ8[5:0]					Res.		SQ7[5:0]						
		rw	rw	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
Res.		SQ6[5:0]					Res.		SQ5[5:0]						
		rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:24	SQ8[5:0]	<p>序列的第 8次转换</p> <p>这些位可通过软件定义哪个通道（CH0…CH58）纳入转换序列</p>

		<p>的第8次转换。0b111111表明序列结束。 当低的转换序列通道被设定为0b111111(序列结束)时, 这些位将会被忽略。 通道选择请参考 SQ4[5:0] 的说明。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 并且才允许通过软件对这些位执行写操作。</i></p>
23:22	保留	必须保持复位值。
21:16	SQ7[5:0]	<p>序列的第 7 次转换 这些位可通过软件定义哪个通道 (CH0...CH58) 纳入转换序列的第3次转换。0b111111 表明序列结束。 当低的转换序列通道被设定为0b111111(序列结束)时, 这些位将会被忽略。 通道选择请参考 SQ4[5:0] 的说明。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对这些位执行写操作。</i> <i>*本寄存器的操作参考ADC_CHSELRI</i></p>
15:14	保留	必须保持复位值。
13:8	SQ6[5:0]	<p>序列的第 6 次转换 这些位可通过软件定义哪个通道 (CH0...CH58) 纳入转换序列的第2次转换。0b111111表明序列结束。 当低的转换序列通道被设定为0b111111(序列结束)时, 这些位将会被忽略。 通道选择请参考 SQ6[5:0] 的说明。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对这些位执行写操作。</i></p>
7:6	保留	必须保持复位值。
5:0	SQ5[5:0]	<p>序列的第5次转换 这些位可通过软件定义哪个通道 (CH0...CH58) 纳入转换序列的第1次转换。0b111111表明序列结束。 当低的转换序列通道被设定为0b111111(序列结束)时, 这些位将会被忽略。 通道选择请参考 SQ4[5:0] 的说明。 <i>注意: 仅当 ADSTART=0 时 (这可确保当前未进行任何转换), 才允许通过软件对这些位执行写操作。</i></p>

16.10.9 ADC 看门狗 1 监控电压阈值寄存器 (ADC_AWD1TR)

偏移地址: 0x28

复位值: 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				HT1[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				LT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	HT1[11:0]	ADC看门狗 1 电压监控电压高阈值 用于定义ADC看门狗监控电压阈值上限。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i>
15:12	保留	必须保持复位值。
11:0	LT1[11:0]	ADC看门狗 1 电压监控电压低阈值 用于定义ADC看门狗监控电压阈值下限。 <i>注意：仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。</i>

16.10.10 ADC 校准系数 (ADC_CALFACT)

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAINK								GAINB							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CALTIMES		Res.		CALFACT[5:0]							
				rw	rw					rw	rw	rw	rw	rw	rw

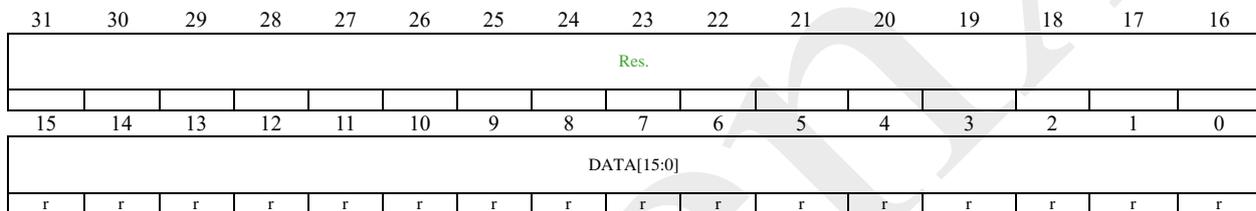
位/位域	名称	描述
31:24	GAINK	GAIN校准参数K，校准公式： $y=kx+b$ 由软件根据ADC输出数据计算出参数K和B后写入到该寄存器
23:16	GAINB	GAIN校准参数B。
15:11	保留	必须保持复位值。
10:9	CALTIMES	校准次数，每次启动校准，硬件将自动按下面配置的次数进行多次采样转换，并将转换后的数据进行平均后输出到数据寄存器。 00：4次 01：8次 10：16次 11：64次

8:6	保留	必须保持复位值。
5:0	CALFACT[5:0]	<p>OFFSET校准系数OFFSET</p> <p>这些位可由硬件或软件写入。</p> <ul style="list-style-type: none"> - 校准完成后，会立即由硬件更新为校准系数。 - 软件可向这些位写入新的校准系数。新的转换，会立即应用新校准系数。 <p><i>注意：仅当 ADEN=1 且 ADSTART=0 时 (ADC 已使能、当前未执行任何校准和转换)，才允许通过软件对这些位执行写操作。</i></p>

16.10.11 ADC 数据寄存器 (ADC_DR)

偏移地址：0x40

复位值：0x0000 0000



位/位域	名称	描述
31:15	保留	必须保持复位值。
15:0	DATA[15:0]	<p>转换后的数据 (Converted data)</p> <p>这些位为只读。其中包含上一转换通道的转换结果。实际采样有效数据是12位，数据可以采用左对齐和右对齐。</p>

17 触摸按键 (ATK)

17.1 简介

触摸键提供了一种简单、可靠的方法来实现手指触摸检测。在触摸扫描操作期间，设备支持 2 组 21 个通道的触摸键检测。

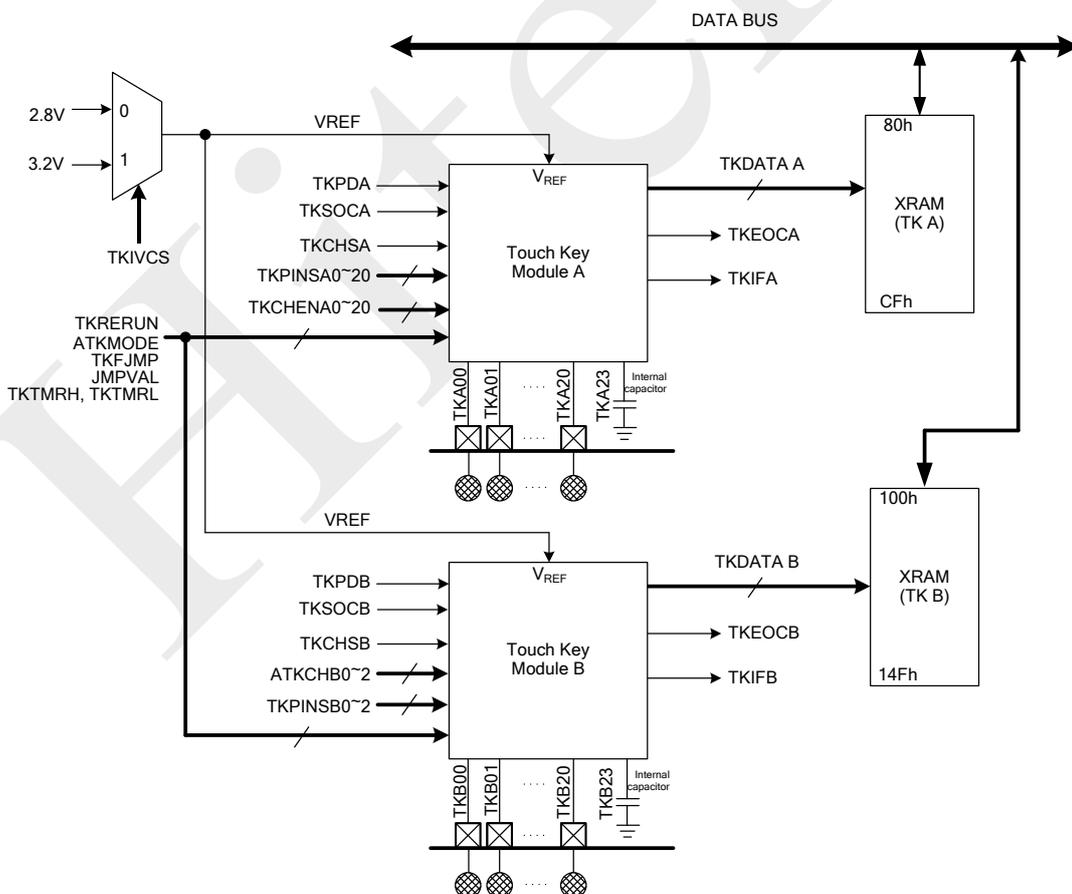
17.2 主要特性

- 支持对灵敏度要求较高的触控应用
- 具有很强的抗干扰性
- 最大支持 42 路通道的触摸键检测
- 支持低功耗模式

17.3 功能说明

17.3.1 结构框图

图 17-1 ATK 结构框图



17.3.2 功能描述

要使用触摸键，用户应正确设置引脚，通过 *GPIO 端口模式寄存器*将引脚设置为模拟通道。

在 TK 模式下，用户通过 *APB1 外设时钟使能寄存器*(RCC_APBENR1)使能 ATK 模块时钟，然后将 TKSOC 位置 1 以开始触控按键转换，转换结束后 TKSOC 位自动清零。如果输入给 ATK 的时钟频率太慢，硬件可能无法清除 TKSOC。TKxEOC=0 表示正在进行转换，TKxEOC=1 表示转换完成，并且触摸键计数结果存在模块 RAM 中(TKA/TKB)。在 TKxEOC=1 之后，用户必须等待至少 1 个 rck 延时才能进行下一次转换。但如果将 TKxRERUN 设置为 1，则将始终转换 TK，无需为每次转换设置 TKSOC。减少/增加 TKTMR 可以减少/增加 TKDATA 以便适应系统条件。

ATK 具有内部内置的参考电容器来模拟按键行为。设置 TKxCHS=17h 并开始扫描可以获取内部参考电容器(TKCAP)的 TK 计数数据。由于内部电容器不受水或手机辐射的影响，因此对于比较背景噪声很有用。设置 TKFJMP，可以通过硬件控制自动更改触摸键时钟的频率，这有助于提高抗噪能力。

寄存器 ATKCHA/B 用于指定扫描 TK 通道，并且每个位都映射到 TK 引脚。TK 扫描将从低位扫描到高位。如果 TKM=0，则 TK 最多可以扫描 22 个通道，TK00~TK20 和 TKCAP(TK23),每个通道扫描一次。如果 TKM=1，则 TK 最多可以扫描 20 个通道，每个通道扫描两次。如果 TKM=2，则 TK 最多可以扫描 10 个通道，每个通道被扫描 4 次。如果 TKM=3，则 TK 最多可以扫描 5 个通道，每个通道被扫描 8 次。ATKCHA 和 ATKCHB 用来指定 TK 开始扫描的第一个通道。

TKA 和 TKB RAM 最大各支持 40 个 16 位数据。

若 TKx 的采样通道索引 $n_1, n_2, n_3, \dots, n_k$ ，TKx 的起始地址为 BaseAddr_x (见章节 17.4.20-17.4.21)。当条件 TKM=0 时，对应存储于 TKx RAM 的 TK 数据地址为 BaseAddr_x, BaseAddr_x+(n_2-n_1)*2, BaseAddr_x+(n_3-n_1)*2, ..., BaseAddr_x+(n_k-n_1)*2; 当条件 TKM≠0 时，对应存储于 TKx RAM 的 TK 数据地址为 BaseAddr_x, BaseAddr_x+2, BaseAddr_x+4, ..., BaseAddr_x+($k-1$)*2。

例如：条件 TKM=0，扫描 TKA15/TKA14/TKA08/TKA07/TKA06/TKA02，TKPINS_A=1100_0001_1100_0100, ATKCHA=1100_0001_1100_0100, THCHSA=0x02(指定第一个扫描通道)。

表 17-1 存储在 TKA RAM 中的 TK 数据的排列(TKM=0)

ATK 偏移地址	数据
0x80	TKA00[15:0]
0x82	TKA01[15:0]
0x84	TKA02[15:0]

0xA8	TKA20[15:0]
0xAE	TKA23[15:0]

其中 TKA 有效数据为[13:0],第 15 位和第 14 位硬件固定补 0。

例如：条件 TKM=1，扫描 TKA15/TKA14/TKA08/TKA07/TKA06/TKA02，TKPinsa=1100_0001_1100_0100, ATKCHA=1100_0001_1100_0100, THCHSA=0x02(指定第一个扫描通道)

表 17-2 存储在 TKA RAM 中的 TK 数据的排列(TKM=1)

地址	数据
0x80	TKA02 [15:0] 第一次
0x82	TKA02[15:0] 第二次
0x84	TKA06[15:0] 第一次
0x86	TKA06[15:0] 第二次

0x94	TKA15[15:0] 第一次
0x96	TKA15[15:0] 第二次

其中 TKA 有效数据为[13:0],第 15 位和第 14 位固定补 0。

例如：条件 TKM=2，扫描 TKA15/TKA14/TKA08/TKA07/TKA06/TKA02，TKPinsa=1100_0001_1100_0100, ATKCHA=1100_0001_1100_0100, THCHSA=0x02(指定第一个扫描通道)

表 17-3 存储在 TKA RAM 中的 TK 数据的排列(TKM=2)

地址	数据
0x80	TKA02 [15:0] 第一次
0x82	TKA02[15:0] 第二次
0x84	TKA02[15:0] 第三次
0x86	TKA02[15:0] 第四次
0x88	TKA06 [15:0] 第一次
0x8A	TKA06[15:0] 第二次
0x8C	TKA06[15:0] 第三次
0x8E	TKA06[15:0] 第四次

0xA8	TKA15[15:0] 第一次
0xAA	TKA15[15:0] 第二次
0xAC	TKA15[15:0] 第三次
0xAE	TKA15[15:0] 第四次

其中 TKA 有效数据为[13:0],第 15 位和第 14 位固定补 0。

例如：条件 TKM=3，扫描 TKA15/TKA14/TKA08/TKA07/TKA06/TKA02，TKPinsa=1100_0001_1100_0100, ATKCHA=1100_0001_1100_0100, THCHSA=0x02(指定第一个扫描通道)

表 17-4 存储在 TKARAM 中的 TK 数据的排列(TKM=3)

地址	数据
0x80	TKA02 [15:0] 第一次
0x82	TKA02[15:0] 第二次
0x84	TKA02[15:0] 第三次
0x86	TKA02[15:0] 第四次
0x88	TKA02 [15:0] 第五次
0x8A	TKA02[15:0] 第六次
0x8C	TKA02[15:0] 第七次
0x8E	TKA02[15:0] 第八次

0xC0	TKA14[15:0] 第一次
0xC2	TKA14[15:0] 第二次
0xC4	TKA14[15:0] 第三次
0xC6	TKA14[15:0] 第四次
0xC8	TKA14[15:0] 第五次
0xCA	TKA14[15:0] 第六次
0xCC	TKA14[15:0] 第七次
0xCE	TKA14[15:0] 第八次

其中 TKA 有效数据为[13:0],第 15 位和第 14 位固定补 0。

TKBRAM 内容排布和 TKA 相同,TKBRAM 的偏移地址范围为 0x100~14F。对于 TKA 和 TKBRAM 可通过 CPU 或者 DMA 进行读写访问。

17.4 ATK 寄存器

17.4.1 TKA 控制寄存器 (CRA)

偏移地址：0x00

复位值：0x00FF 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				TKTMR[11:0]											
				rw	rw	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

TKPD	DMAEN	Res.	TKSPEAD	TKJMPVAL			TKFJMP	Res.	TKIFEN	TKXC AP	TKM		TKS HILD	TKS OC
rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	TKTMR	触摸键A扫描长度位11~0调节 00: 最短, FFF: 最长
15	TKPD	触摸键A使能 0: 启用 1: 禁用
14	DMAEN	直接存储器访问使能 此位由软件置1和清零, 用于使能DMA请求的生成。这样便可使用DMA控制器自动管理TKA和TKB的转换数据。 0: 禁止DMA 1: 使能DMA 注意: 仅当TKASOC和TKBSOC=0时(这可确保DMA当前未进行任何转换), 才允许通过软件对此位执行写操作。 当TKM=0(单次通道扫描), 因数据存放地址不连续(取决于通道使能和内部电容采样通道使能), 不建议使能DMA, 建议CPU读取方式。 当TKM设置为多次通道扫描, 被采样的数据被存放在RAM的连续地址中, 可以采用DMA进行读取。
13	保留	必须保持复位值。
12	TKSPEAD	触摸键A时钟频率扩展 0: 禁用 1: 启用
11:9	TKJMPVAL	触摸键A时钟频率微调, 仅在TKAJMP=0时可用 000: 频率最低, 111:频率最高。
8	TKFJMP	触摸键A时钟频率自动调整选项 0: 禁用, 1: 启用。
7:6	保留	必须保持复位值。
5	TKIFEN	触摸键A转换结束中断使能 0: 不使能, 1: 使能。

- 4 TKXCAP 触摸键A外部电容器选择
0: 禁止触摸键A外部电容器,
1: 启用触摸键A外部电容器。
- 3:2 TKM TKA和TKB触摸键扫描模式
00:每个通道扫描1次, 最多22个TK通道
01:每个通道扫描2次, 最多20个TK通道
10:每个通道扫描4次, 最多10个TK通道
11:每个通道扫描8次, 最多5个TK通道
- 1 TKSHILD 触摸键A屏蔽选项
0: 禁用,
1: 启用。
- 0 TKSOC 触摸键A转换开始
设置为1即开始启动触摸键A转换, 如果模块输入时钟足够快,
则在转换结束时, 该位硬件自动清零。软件也可写0清该标志

17.4.2 TKB 控制寄存器 (CRB)

偏移地址: 0x04

复位值: 0x00FF 8000

31 30 29 28				27 26 25 24 23 22 21 20 19 18 17 16											
Res.				TKTMR[11:0]											
				rw	rw	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											
TKPD	DMAEN	TKSPEAD	TKJMPVAL			TKFJMP	Res.	TKIFEN	TKXC AP	TKM		TKS HILD	TKS OC		
rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	TKTMR	触摸键B扫描长度位11~0调节 00: 最短, FFF: 最长
15	TKPD	触摸键B使能 0: 启用 1: 禁用
14	DMAEN	直接存储器访问使能 此位由软件置1和清零, 用于使能DMA请求的生成。这样便可使用DMA控制器自动管理TKA和TKB的转换数据。

		0: 禁止DMA
		1: 使能DMA
		注意: 仅当TKASOC和TKBSOC=0时(这可确保DMA当前未进行任何转换), 才允许通过软件对此位执行写操作。
		当TKM=0(单次通道扫描), 因数据存放地址不连续(取决于通道使能和内部电容采样通道使能), 不建议使能DMA, 建议CPU读取方式。
		当TKM设置为多次通道扫描, 被采样的数据被存放在RAM的连续地址中, 可以采用DMA进行读取。
13	保留	必须保持复位值。
12	TKSPEAD	触摸键B时钟频率扩展 0: 禁用 1: 启用
11:9	TKJMPVAL	触摸键B时钟频率微调, 仅在TKBFJMP=0时可用 000: 频率最低, 111:频率最高。
8	TKFJMP	触摸键B时钟频率自动调整选项 0: 禁用, 1: 启用。
7:6	保留	必须保持复位值。
5	TKIFEN	触摸键B转换结束中断使能 0: 不使能, 1: 使能。
4	TKXCAP	触摸键B外部电容器选择 0: 禁止触摸键B外部电容器, 1: 启用触摸键B外部电容器。
3:2	TKM	TKB触摸键扫描模式 00:每个通道扫描1次, 最多22个TK通道 01:每个通道扫描2次, 最多20个TK通道 10:每个通道扫描4次, 最多10个TK通道 11:每个通道扫描8次, 最多5个TK通道
1	TKSHILD	触摸键B屏蔽选项 0: 禁用, 1: 启用。
0	TKSOC	触摸键B转换开始 设置为1即开始启动触摸键B转换, 如果模块输入时钟足够快,

则在转换结束时，该位硬件自动清零。软件也可写0清该标志

17.4.3 TK 状态寄存器 (ISR)

偏移地址：0x08

复位值：0x0000 0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							TKBIF	TKBEOC	Res.					TKAIF	TKAEOC
							rc_wl	ro						rc_wl	ro

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TKBIF	触摸键B中断标志 在触摸键B转换结束后硬件自动置1，需要软件写0或者将TKBSOC位写1清除该标志。
8	TKBEOC	触摸键B转换结束标志 在TKBOC=1之后TKBEOC可能有3us的延迟，所以软件必须等待足够时间才能轮询该标志。 0：表示正在进行转换 1：表示转换完成
7:2	保留	必须保持复位值。
1	TKAIF	触摸键A中断标志 在触摸键A转换结束后硬件自动置1，需要软件写0或者将TKASOC位写1清除该标志。
0	TKAEOC	触摸键A转换结束标志 在TKASOC=1之后TKAEOC可能有3us的延迟，所以软件必须等待足够时间才能轮询该标志。 0：表示正在进行转换 1：表示转换完成

17.4.4 TK 启动通道选择寄存器 (CHSR)

偏移地址：0x0C

复位值：0x0000 1F1F

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBRERUN	Res.	TKBIVCS	TKBCHS					TKARERUN	Res.	TKAIVCS	TKACHS				
rw		rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	TKBRERUN	TKB自动重新启动，无需重新设置TKBSOC即可重新启动TKB转换器。 0: 自动重启禁用，每次TKB转换都需执行一次TKBSOC 1: 自动重启启用，TKBSOC执行一次后，TKB将连续转换而需重新设置TKBSOC。
14	保留	必须保持复位值。
13	TKBIVCS	触摸键B操作电压选择 0: 2.8V 1: 3.6V
12:8	TKBCHS	指定触摸按键B第一个扫描通道 00000: TKB0 00001: TKB1 00010: TKB2 00011: TKB3 10011: TKB19 10100: TKB20 10111: TKBCAP内部参考电容通道
7	TKARERUN	TKA自动重新启动，无需重新设置TKSOCA即可重新启动TKA转换器。 0: 自动重启禁用，每次TKA转换都需执行一次TKSOCA 1: 自动重启启用，TKSOCA执行一次后，TKA将连续转换而需重新设置TKSOCA。
6:5	TKAIVCS	触摸键A操作电压选择 0: 2.8V 1: 3.6V
4:0	TKACHS	指定触摸按键A第一个扫描通道 00000: TKA0

- 00001: TKA1
- 00010: TKA2
- 00011: TKA3
-
- 10011: TKA19
- 10100: TKA20
- 10111: TKACAP内部参考电容通道

17.4.5 TKA 通道使能寄存器 (CHENRA)

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	
Res.								TKA23E N	Res.		TKACHEN [20:16]					
								rw			rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TKACHEN [15:0]																
rw		rw		rw		rw		rw		rw		rw		rw		

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	TKA23EN	TKACAP内部参考电容通道扫描使能 0: 禁用, 1: 启用
22:21	保留	必须保持复位值。
20:0	TKACHEN	TKA20~TKA0通道扫描使能 0: 禁用, 1: 启用

17.4.6 TKB 通道使能寄存器 (CHENRB)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.								TKB23E N	Res.		TKBCHEN[20:16]					
								rw			rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TKBCHEN[15:0]																
rw		rw		rw		rw		rw		rw		rw		rw		

位/位域	名称	描述
31:24	保留	必须保持复位值。

23	TKB23EN	TKBCAP内部参考电容通道扫描使能 0: 禁用, 1: 启用
22:21	保留	必须保持复位值。
20:0	TKBCHEN	TKB20~TKB0通道扫描使能 0: 禁用, 1: 启用

17.4.7 TKA 通道调试参数 1 寄存器 (ITRIMRA1)

偏移地址: 0x20

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH3TRIM							Res.	TKCH2TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH1TRIM							Res.	TKCH0TRIM						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH3TRIM	触摸按键A第四个通道CH3的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH2TRIM	触摸按键A第三个通道CH2的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH1TRIM	触摸按键A第二个通道CH1的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH0TRIM	触摸按键A第一个通道CH0的ITRIM值, 每次上电使用TK模块前, 软件通过尝试写入该通道不同的ITRIM值, 让该通道转换后的数据符合预期, 从而确定该ITRIM值。

17.4.8 TKA 通道调试参数 2 寄存器 (ITRIMRA2)

偏移地址: 0x24

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

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

Res.	TKCH7TRIM							Res.	TKCH6TRIM							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH5TRIM							Res.	TKCH4TRIM							
	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	

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH7TRIM	触摸按键A第八个通道CH7的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH6TRIM	触摸按键A第七个通道CH6的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH5TRIM	触摸按键A第六个通道CH5的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH4TRIM	触摸按键A第五个通道CH4的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.9 TKA 通道调试参数 3 寄存器 (ITRIMRA3)

偏移地址：0x28

复位值：0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	TKCH11TRIM							Res.	TKCH10TRIM							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH9TRIM							Res.	TKCH8TRIM							
	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	

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH11TRIM	触摸按键A第十二个通道CH11的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH10TRIM	触摸按键A第十一个通道CH10的ITRIM值

15	保留	必须保持复位值。
14:8	TKCH9TRIM	触摸按键A第十个通道CH9的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH8TRIM	触摸按键A第九个通道CH8的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.10 TKA 通道调试参数 4 寄存器 (ITRIMRA4)

偏移地址：0x2C

复位值：0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH15TRIM							Res.	TKCH14TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH13TRIM							Res.	TKCH12TRIM						
	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

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH15TRIM	触摸按键A第十六个通道CH15的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH14TRIM	触摸按键A第十五个通道CH14的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH13TRIM	触摸按键A第十四个通道CH13的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH12TRIM	触摸按键A第十三个通道CH12的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.11 TKA 通道调试参数 5 寄存器 (ITRIMRA5)

偏移地址：0x30

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	TKCH19TRIM							Res.	TKCH18TRIM							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	TKCH17TRIM							Res.	TKCH16TRIM							
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH19TRIM	触摸按键A第二十个通道CH19的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH18TRIM	触摸按键A第十九个通道CH18的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH17TRIM	触摸按键A第十八个通道CH17的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH16TRIM	触摸按键A第十七个通道CH16的ITRIM值，每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.12 TKA 通道调试参数六寄存器 (ITRIMRA6)

偏移地址: 0x34

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH23TRIM							Res.							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TKCH20TRIM							
	rw	rw	rw	rw	rw	rw	rw		rw						

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH23TRIM	触摸按键A内部电容参考通道CH23的ITRIM值
23:7	保留	必须保持复位值。
6:0	TKCH20TRIM	触摸按键A第二十个通道CH20的ITRIM值，

每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.13 TKB 通道调试参数 1 寄存器 (ITRIMRB1)

偏移地址: 0x40

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH3TRIM							Res.	TKCH2TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH1TRIM							Res.	TKCH0TRIM						
	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

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH3TRIM	触摸按键B第四个通道CH3的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH2TRIM	触摸按键B第三个通道CH2的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH1TRIM	触摸按键B第二个通道CH1的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH0TRIM	触摸按键B第一个通道CH0的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.14 TKB 通道调试参数 2 寄存器 (ITRIMRB2)

偏移地址: 0x44

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH7TRIM							Res.	TKCH6TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH5TRIM							Res.	TKCH4TRIM						
	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

位/位域	名称	描述
31	保留	必须保持复位值。

30:24	TKCH7TRIM	触摸按键B第八个通道CH7的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH6TRIM	触摸按键B第七个通道CH6的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH5TRIM	触摸按键B第六个通道CH5的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH4TRIM	触摸按键B第五个通道CH4的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.15 TKB 通道调试参数 3 寄存器 (ITRIMRB3)

偏移地址：0x48

复位值：0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	TKCH11TRIM							Res.	TKCH10TRIM							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	TKCH9TRIM							Res.	TKCH8TRIM							
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH11TRIM	触摸按键B第十二个通道CH11的ITRIM值
23	保留	读为0，写入无效。
22:16	TKCH10TRIM	触摸按键B第十一个通道CH10的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH9TRIM	触摸按键B第十个通道CH9的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH8TRIM	触摸按键第B九个通道CH8的ITRIM值， 每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而

确定该ITRIM值。

17.4.16 TKB 通道调试参数 4 寄存器 (ITRIMRB4)

偏移地址: 0x4C

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH15TRIM							Res.	TKCH14TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH13TRIM							Res.	TKCH12TRIM						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH15TRIM	触摸按键B第十六个通道CH15的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH14TRIM	触摸按键B第十五个通道CH14的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH13TRIM	触摸按键B第十四个通道CH13的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH12TRIM	触摸按键B第十三个通道CH12的ITRIM值，每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.17 TKB 通道调试参数 5 寄存器 (ITRIMRB5)

偏移地址: 0x50

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH19TRIM							Res.	TKCH18TRIM						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TKCH17TRIM							Res.	TKCH16TRIM						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

31	保留	必须保持复位值。
30:24	TKCH19TRIM	触摸按键B第二十个通道CH19的ITRIM值
23	保留	必须保持复位值。
22:16	TKCH18TRIM	触摸按键B第十九个通道CH18的ITRIM值
15	保留	必须保持复位值。
14:8	TKCH17TRIM	触摸按键B第十八个通道CH17的ITRIM值
7	保留	必须保持复位值。
6:0	TKCH16TRIM	触摸按键B第十七个通道CH16的ITRIM值，每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.18 TKB 通道调试参数 6 寄存器 (ITRIMRB6)

偏移地址: 0x54

复位值: 0x00000000(datasheet 不体现,除上电复位外其它复位无法复位该寄存器)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TKCH23TRIM							Res.							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TKCH20TRIM							
	rw	rw	rw	rw	rw	rw	rw		rw						

位/位域	名称	描述
31	保留	必须保持复位值。
30:24	TKCH23TRIM	触摸按键B内部电容参考通道CH23的ITRIM值
23:7	保留	必须保持复位值。
6:0	TKCH20TRIM	触摸按键B第二十一个通道CH20的ITRIM值，每次上电使用TK模块前，软件通过尝试写入该通道不同的ITRIM值，让该通道转换后的数据符合预期，从而确定该ITRIM值。

17.4.19 TK 引脚互换寄存器 (PINABSR)

偏移地址: 0x58

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.											TKPINABS[20:16]				
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKPINABS [15:0]															
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:0	TKPINABS	TKA[20:0]引脚和TKB[20:0] 引脚被互调 0: 禁用, 1: 启用 如PAD00默认固定为TKA[00]的通道, PAD20固定TKB[00]的通道, 当TKPINABS[0]为1即PAD00调整为TKB[00]的通道, PAD20固定为TKA[00]的通道。

17.4.20 TKA RAM (TKA_RAM)

偏移地址: 0x80~0xCF

复位值: 0x00000000

17.4.21 TKB RAM (TKB_RAM)

偏移地址: 0x100~0x14F

复位值: 0x00000000

18 模拟智能组合 (SAC)

18.1 简介

MCU 模拟智能组合模块内置四个模拟比较器 (COMP)、四个运算放大器 (OPAMP)和四个 12 位的数模转换器(DAC)。这三个子模块可以单独开关使用也可以组合使用。其中比较器用于比较两个输入模拟电压的大小, 并根据比较结果输出高/低电平。当比较器“+”端输入电压高于“-”端输入电压时, 比较器输出为高电平。当比较器“+”端输入电压低于“-”端输入电压时, 比较器输出为低电平。支持模拟信号触发从低功耗模式唤醒和模拟信号整形应用。

运算放大器可以配置为同向、反向、不同增益值组合, 也可以使用外部电阻进行级联。OPAMP 的输入范围是 $0\sim AVCC$, 输出范围是 $0.1V$ 到 $AVCC-0.1V$ 。

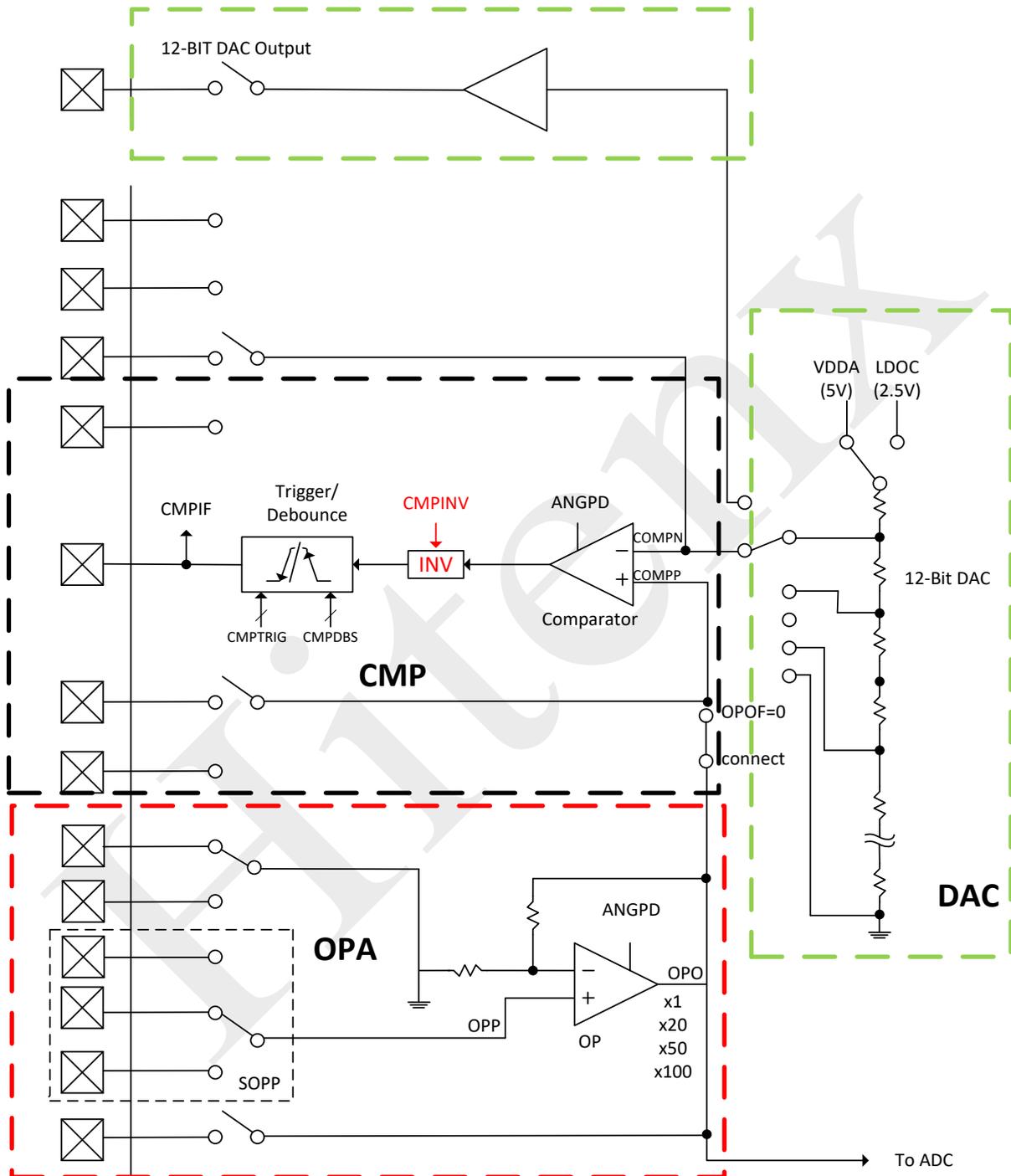
18.2 主要特性

- 比较器可以配置正相输入和负相输入信号, 灵活选择输入电压:
 - 多路复用 I/O;
 - DAC 输出;
 - 内部参考电压 V_{BGR} 及其三个因数分压值 (1/4、1/2、3/4, 通过分压调节器进行分压);
 - 内部参考电压 V_{core} 。
- 比较器可编程迟滞电压
- 比较器输出可以片内连接到 I/O 或 TIMER 输入, 可以触发以下事件:
 - 捕捉事件;
 - 为实现快速 PWM 关断的刹车事件。
- 比较器输出可以进行数字滤波, 且滤波时间可配置
- 比较器都可以产生中断, 并可以从睡眠和停止模式唤醒芯片(通过 EXTI 控制器)
- OPAMP 可编程增益
- OPAMP 可输出到比较器
- OPAMP 输入可灵活配置

18.3 SAC 功能描述

18.3.1 结构框图

图 18-1 SAC 结构框图



18.3.2 引脚和内部信号

用作 SAC 输入的 GPIO 必须配置为模拟模式 ([GPIO 端口模式寄存器](#))。

SAC 输出也可以连接到 GPIO。

比较器输出也可以在内部重定向到 TIMER 输入,可用于时序测量的输入捕获等应用。

DAC 输出特征:

- 四路输出通道
- DAC_OUTx 可与输出引脚断开连接并用作普通 GPIO
- DAC_OUTx 可通过片内信号与其它外设连接 (如比较器和 ADC)
- DAC 输出通道 (缓冲/非缓冲)

DAC 包含四个独立的输出通道。每个输出通道都可以通过片内信号连接到其它外设,如 COMP、ADC。在这种配置下,DAC 输出通道可与 DAC_OUTx 输出引脚断开连接,相应的 GPIO 可用于其他用途。

DAC 输出可以连接到输出缓冲器,从而对 DAC 输出信号实现缓冲。

表 18-1 DAC 输入/输出引脚

引脚名称	信号类型	说明
VREF+	输入,正模拟参考电压	DAC 正参考电压, $VREF+ \leq VDDA_{max}$
VDDA	输入,模拟电源	模拟电源
VSSA	输入,模拟电源地	模拟电源地
DAC_OUTx	模拟输出信号	DAC 通道 x 模拟输出

注意: 当选择 VBGR 及其分压 (1/2VBGR) 作为比较器反相输入信号时需要使能 BGR。

注意: 当选择 VBGR 及其分压 (1/2VBGR) 作为比较器反相输入信号时需延时 200us 等待内部 scaler 稳定。

注意: 当选择 BGR 分压 (1/2BGR) 作为比较器反相输入信号时,会使能内部分压电阻桥,因此会产生电阻桥功耗。而选择 BGR 作为输入信号时,不会使能电阻桥。

18.3.3 复位和时钟源

SAC 时钟与 PCLK (APB2 时钟) 同步。

RCC 控制器中单独提供 SAC 时钟使能控制位。

注意: 比较器极性选择逻辑和到端口的输出重定向独立于 PCLK 时钟。因此,即使在停止低功耗模式下比较器仍能正常工作。

18.3.4 比较器锁存机制

比较器可用于过流或热保护等安全用途。对于具有特定功能安全要求的应用,必须保证在发生意外寄存器访问或程序计数器损坏时,不能更改比较器编程参数设置。为此,需要对比较器控制和状态寄存器进行写保护 (只读)。

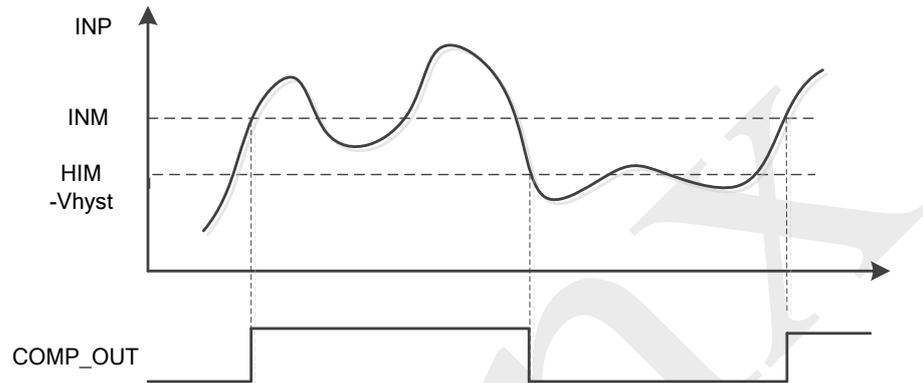
比较器锁存机制通过寄存器 LOCK 位置 1 来使能。这将导致整个 COMP_CSR

寄存器变成只读，包括 LOCK 位。只能通过 MCU 复位来禁止比较器控制和状态寄存器写保护功能。

18.3.5 比较器迟滞

比较器具有迟滞特性，可在有噪声信号时避免输出产生意外变化。

图 18-2 COMP 迟滞

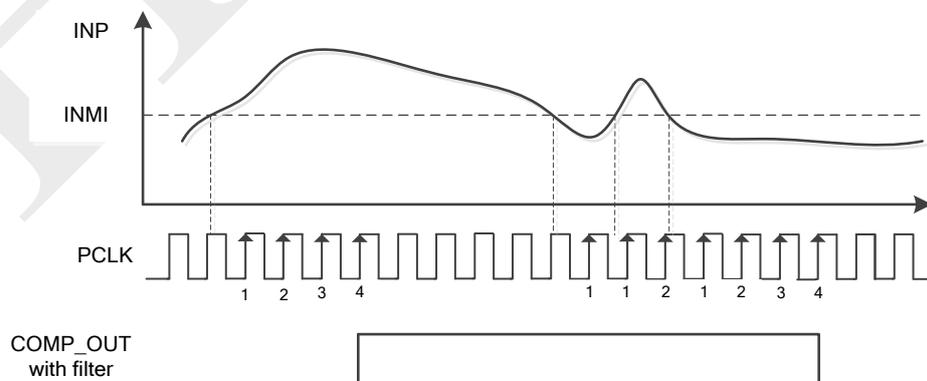


18.3.6 比较器输出滤波功能

比较器具有输出滤波功能，来滤除系统噪声。可以通过设置寄存器 EN 位使能滤波功能，此时采用系统时钟 PCLK 进行输出滤波，滤波时钟周期通过 FT 来设置。

注意： 由于采用 PCLK 进行输出滤波，因此在停止低功耗模式下比较器输出滤波功能不可用。

图 18-3 COMP 输出滤波



18.3.7 比较器中断

比较器输出片内连接到中断和事件 EXTI 控制器。每个比较器都有其各自的 EXTI 线，能够产生中断或事件。该机制还可用于退出低功耗模式。

更多详细信息，请参见中断和事件 EXTI 部分。

使能 COMP 中断步骤：

- 1) 将对应于 COMP 输出事件的 EXTI 线配置为中断模式并使能，然后选择上升沿触发、下降沿触发或双沿触发模式；
- 2) 配置并使能映射到相应 EXTI 线的 NVIC IRQ 通道；
- 3) 使能比较器 COMP。

表 18-2 COMP 中断控制

中断事件	中断标志	中断使能	从 Sleep 模式下唤醒	从 Stop 模式下唤醒
COMP 输出	COMP_OUT	通过 EXTI	支持	支持

18.3.8 DAC 通道使能

通过将 DAC_CR 寄存器中的相应 ENx 位置 1，可使能对应 DAC 转换通道，从而使其上电。将 ENx 位置 1，经过一段启动时间 t_{WAKEUP} 后，DAC 通道被使能。

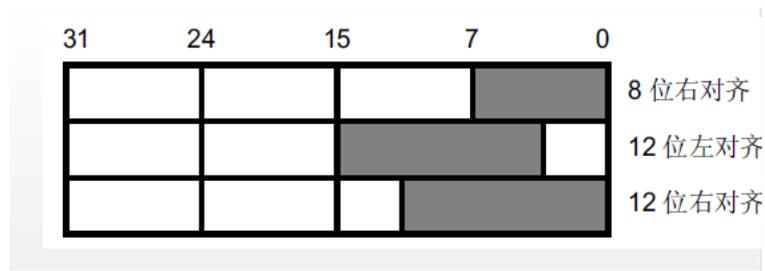
注意： ENx 位只会使能 DAC 转换通道的模拟电路。因此即使 ENx 位复位，DAC 转换通道的数字接口仍可处于使能状态。

18.3.9 DAC 数据格式

根据所选的配置模式，输入数据必须按如下方式写入指定寄存器：

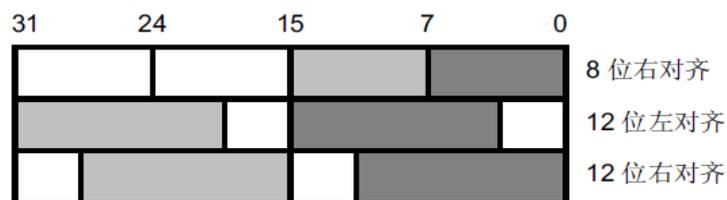
- 对于 DAC 单通道模式，有三种可能的方式：
 - 8 位右对齐：软件必须将输入数据加载到 DAC_DHR8Rx[7:0]位（存储到 DHRx[11:4]位）；
 - 12 位左对齐：软件必须将输入数据加载到 DAC_DHR12Lx[15:4]位（存储到 DHRx[11:0]位）；
 - 12 位右对齐：软件必须将输入数据加载到 DAC_DHR12Rx[11:0]位（存储到 DHRx[11:0]位）。

根据 DAC_DHRyyyx 寄存器的数据加载，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，此寄存器在片内没有对应的地址映射）。之后，通过软件或外部事件触发，DHRx 寄存器数据将被自动加载到 DORx 寄存器（数据输出寄存器 x）。

图 18-4 DAC 单通道模式下的数据寄存器


- 对于 DAC 双通道模式，有三种可能的方式：
 - 8 位右对齐：将 DAC 1 通道的输入数据加载到 DAC_DHR8RD[7:0]位（存储到 DHR1[11:4]位），将 DAC 2 通道的输入数据加载到 DAC_DHR8RD[15:8]位（存储到 DHR2[11:4]位）；
 - 12 位左对齐：将 DAC 1 通道的输入数据加载到 DAC_DHR12LD[15:4]位（存储到 DHR1[11:0]位），将 DAC 2 通道的输入数据加载到 DAC_DHR12LD[31:20]位（存储到 DHR2[11:0]位）；
 - 12 位右对齐：将 DAC 1 通道的输入数据加载到 DAC_DHR12RD[11:0]位（存储到 DHR1[11:0]位），将 DAC 2 通道的输入数据加载到 DAC_DHR12RD[27:16]位（存储到 DHR2[11:0]位）。

根据 DAC_DHRyyyD 寄存器的数据加载，用户写入的数据将移位并存储到 DHR1 和 DHR2（数据保持寄存器，此寄存器在片内没有对应的地址映射）。之后，通过软件或外部事件触发，DHR1 和 DHR2 寄存器将被自动加载到 DOR1 和 DOR2 寄存器（数据输出寄存器）。

图 18-5 DAC 双通道模式下的数据寄存器


18.3.10 DAC 转换

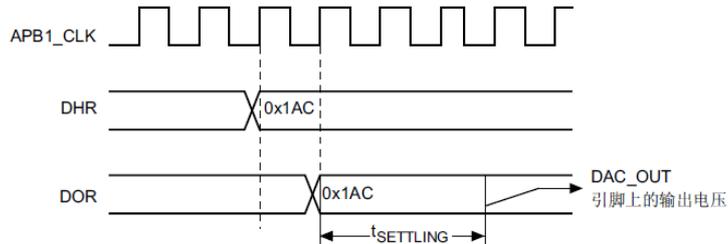
用户无法直接向数据输出寄存器 DAC_DORx 写入数据，任何数据都必须通过 DAC_DHRx 寄存器加载（写入 DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12RD 或 DAC_DHR12LD）才能传输到 DAC 转换通道。

如果未选择硬件触发方式（DAC_CR 寄存器中的 TENx 位复位），那么经过一个 APB（dac_pclk）时钟周期后，DAC_DHRx 寄存器中存储的数据将自动传输到 DAC_DORx 寄存器。但是，如果选择硬件触发方式（DAC_CR 寄存器中的 TENx 位置 1）且触发事件产生时，将在触发信号后经过三个 APB（dac_pclk）时钟周

期才进行数据传输。如果选择软件触发方式，一旦 DAC_SWTR 寄存器的 SWTRIG 位置 1，只需一个 APB 时钟周期即可进行数据传输。

当 DAC_DHRx 数据内容加载到 DAC_DORx 时，经过一段时间 $t_{SETTLING}$ 后，DAC 才会转换输出模拟电压，具体转换时间取决于电源电压和模拟输出负载。

图 18-6 关闭触发 (TEN=0) 时的转换时序



18.3.11 DAC 输出电压

当选择 VREF 作为参考源时，DAC 不支持 VREF=1.25V 时的功能转换。DAC 经过线性转换后，数字输入会转换为 0 到 VREF+ 之间的输出电压。各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DAC \text{ 输出} = V_{REF} \times \frac{DOR}{4096}$$

18.3.12 DAC 触发选择

如果 DAC_CR 寄存器的 TENx 控制位置 1，DAC 可选择外部事件（定时计数器 Timer、外部中断 EXTI）触发或软件触发开启转换。DAC_CR 寄存器的 TSELx[3:0] 控制位用于选择触发转换的事件。触发事件如下表。

如果选择外部事件触发，每当 DAC 接口在所选触发源上检测到上升沿时，DAC_DHRx 寄存器中存储的最后一个数据即会传输到 DAC_DORx 寄存器中。发生外部触发事件后需经过三个 APB 时钟周期，DAC_DORx 寄存器才会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，DAC_DHRx 寄存器的内容只需一个 APB 时钟周期即可传输到 DAC_DORx 寄存器，之后 DAC 转换即会开始。DAC_DHRx 寄存器内容加载到 DAC_DORx 寄存器后，SWTRIG 即由硬件复位。

注意： DAC_CR 寄存器的 ENx 位置 1 时，无法对 TSELx[3:0] 位进行更改。

表 18-3 DAC 触发选择

触发源	类型	TSELx[3:0]
SWTRIG	软件触发	0000
TRG1	TIM1 TRGO	0001
TRG2	TIM2 TRGO	0010
TRG3	TIM3 TRGO	0011

TRG4	TIM4 TRGO	0100
TRG5	TIM5 TRGO	0101
TRG6	TIM6 TRGO	0110
TRG7	LPTIM_OUT	0111
TRG8	TIM7 TRGO	1000
TRG9	PLAOUT0*	1001
TRG10	PLAOUT1*	1010
TRG11	PLAOUT2*	1011
TRG12	PLAOUT3*	1100
TRG13	EXTI_LINE9	1101
TRG14	保留	1110
TRG15	保留	1111

*注：当选择 PLA 输出作为的外部触发源时，为避免毛刺干扰，PLA 内部自动将 DFF 作为 PLA 的输出，但需软件配置 PLA 选择 SYSCLK 作为 DFF 输入时钟。

18.3.13 DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAEN_x 位置 1 时，如果发生外部触发事件（而不是软件触发），则 DAC_DHR_x 寄存器的值会传输到 DAC_DOR_x 寄存器，当传输完成时会产生 DMA 请求。

在双通道模式下，如果两个 DMAEN_x 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，应仅将相应 DMAEN_x 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

由于 DAC_DHR_x 向 DAC_DOR_x 传输数据后 DAC 才会产生 DMA 请求，因此必须在发生第一个触发事件之前通过软件将第一个数据写入 DAC_DHR_x。

DMA 溢出

DAC 的 DMA 请求没有缓冲队列。因此，如果第二个外部触发到达时尚未收到 DMA 控制器的第一个外部触发的应答，则 DAC 将不会发出新的 DMA 请求，并且 DAC_SR 寄存器中的 DMA 通道下溢标志 DMAUDR_x 将置 1，以作为错误状态指示。DAC 通道仍将继续对之前数据进行转换。

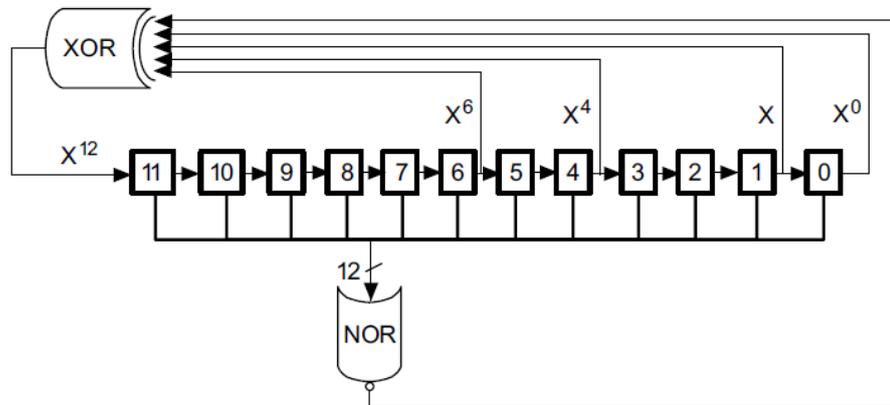
软件应通过写入 1 来将 DMAUDR_x 标志清零，并将所用 DMA 数据流的 DMAEN 位清零，重新初始化 DMA 和 DAC 通道 x，以便重新启动 DMA 数据传输。同时，软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢错误。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道,如果使能 DAC_CR 寄存器中相应的中断控制位 DMAUDRIEx,发生 DMA 下溢错误时还会产生 DMA 溢出中断。

18.3.14 噪声生成器

DAC 具有噪声生成器功能,为了生成可变振幅的伪噪声,可使用 LFSR (线性反馈移位寄存器)。将 WAVEx[1:0]置为“01”即可使能 DAC 噪声生成器。LFSR 中的预加载值为 0xAAA。在每次发生硬件触发事件后,经过三个 APB 时钟周期,该寄存器会依照特定的计算算法完成更新。

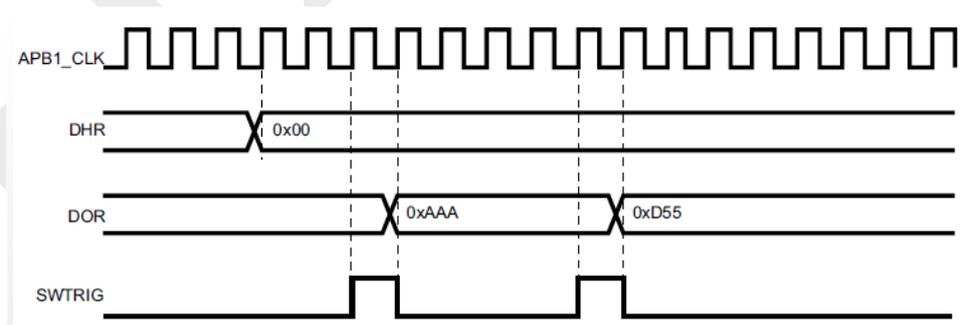
图 18-7 LFSR 寄存器计算算法



LFSR 值可以通过 DAC_CR 寄存器中的 MAMPx[3:0]位来部分或完全屏蔽,在不发生溢出的情况下,该值将与 DAC_DHRx 的内容相加,然后传输到 DAC_DORx 寄存器中。

如果 LFSR 为 0x0000,将向其注入“1”(防锁定机制)。可以通过复位 WAVEx[1:0]位来将 LFSR 波形产生器功能关闭。

图 18-8 LFSR 产生波形的 DAC 转换 (使能软件触发)



注意: 要生成噪声,必须通过将 DAC_CR 寄存器的 TENx 位置 1 来使能 DAC 触发转换模式。

18.3.15 三角波生成器

可以在直流或渐变信号上叠加一个小幅三角波。将 WAVEx[1:0]置为“10”,即可选择 DAC 三角波生成器。振幅通过 DAC_CR 寄存器中的 MAMPx[3:0]位进行

配置。每次发生硬件触发事件后，经过三个 APB 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DAC_DHRx 寄存器内容相加，所得总和将传输到 DAC_DORx 寄存器。只要小于 MAMPx[3:0] 位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVEx[1:0] 位来将三角波产生器功能关闭。

图 18-9 生成 DAC 三角波

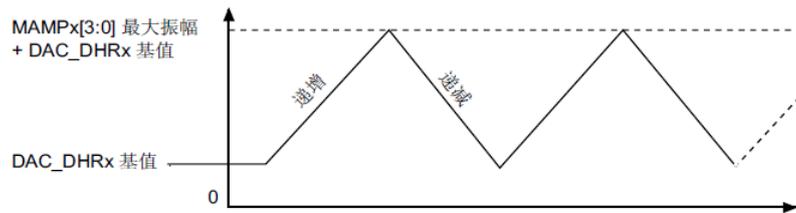
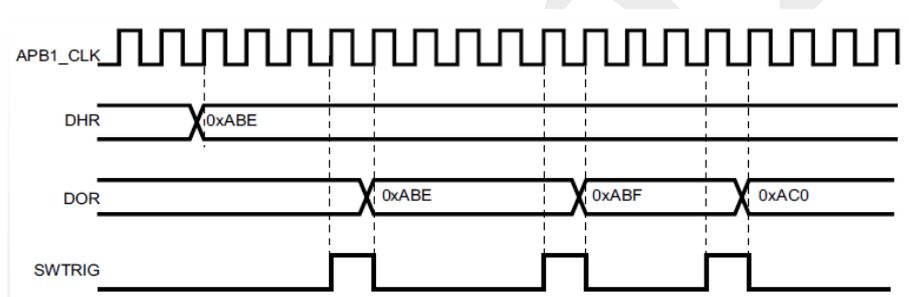


图 18-10 生成三角波波形的 DAC 转换（使能软件触发）



注意： 要生成三角波，必须通过将 DAC_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发模式。MAMPx[3:0] 位必须在使能 DAC 之前进行配置，否则将无法更改。

18.3.16 DAC 通道模式

每个 DAC 通道均可配置为标准模式或采样和保持模式。使用 DAC 时，可以使能输出缓冲器，以实现高驱动能力。使能输出缓冲器之前，需要进行偏移电压校准。此校准是在生产期间执行的（复位后加载），可在应用运行期间通过软件进行调整。

标准模式

在标准模式下，可通过更改输出缓冲器状态以及更改 DAC_OUTx 引脚片内互连方式实现多种输出模式组合。

表 18-4 通道输出模式汇总

MODE[8:0]	通道输出描述
BIT0	DAC BUF 使能，0：不使能，1：使能 当使能 BUF 时，需要软件等待 10us 让 BUF 启动并稳定工作
BIT1	DAC 结果输出到 PAD，0：不输出，1：输出
BIT2	DAC 输出是否旁路 BUF，0：DAC 直接输出，1：DAC+BUF 输出

BIT4:3	DAC BUF 模式选择: 00: 正常模式, 其它: 校准模式
BIT5	DAC BUF 校正源选择: 0: VSSA, 1: VBG
BIT6	DAC BUF 增益选择: 0: 1 倍, 1: 100 倍
BIT8:7	DAC 输出到 PAD 的管脚选择: 00:PAD1,01:PAD2, 10 和 11:PAD3 (DAC 输出 PAD1、PAD2、PAD3 具体看表 4-8PAD 的附加功能)

18.3.17 DAC 输出缓冲器校准

N 位分辨率的数模转换器(DAC)的传递函数为:

$$V_{out} = ((D/2^{N-1}) \times G \times V_{ref}) + V_{os}$$

其中, VOUT 为模拟输出, D 为数字输入, G 为增益, V_{ref} 为标称满量程电压, V_{os} 为偏移电压。对于理想的 DAC, G=1 且 Vos=0。

由于输出缓冲器所具有的特性, 不同芯片的电压偏移可能有所不同, 并会在模拟输出上引入绝对偏移误差。为补偿 V_{os}, 需要通过调整技术进行校准。

提供以下校准技术:

- 出厂调整 (始终使能)

DAC 输出缓冲器偏移是在出厂时进行调整的。RCC_DACTRIMR 寄存器中 DACxBUF_TRIM [4:0]位的默认值是出厂调整值, 会在 DAC 数字接口复位时载入。

注意: 更多关于标称出厂调整条件的详细信息, 请参见数据手册。

18.3.18 DAC 双通道转换

为了在同时需要两个 DAC 通道的应用中有效利用总线带宽, DAC 模块提供三个双通道寄存器可供操作: DHR8RD、DHR12RD 和 DHR12LD。这样, 只需一个寄存器访问即可同时驱动两个 DAC 通道。要生成相应波形, 无需访问 DHRxxxD 寄存器。DAC 的两个输出通道可以分别单独进行转换, 也可以同时进行转换。

通过两个 DAC 通道和这三个双通道寄存器可以实现 11 种转换模式。但如果需要, 所有这些转换模式也都可以通过单独的 DHRx 寄存器来实现。

以下内容将介绍所有这些模式。

18.3.18.1 独立触发 (不产生波形)

要将 DAC 配置为此转换模式, 需要遵循以下步骤:

- 1) 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为不同的值, 以配置不同的触发源。
- 3) 将 DAC 双通道数据加载到所需 DHR 寄存器 (DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

当产生 DAC 1 通道触发信号，DHR1 寄存器的内容传输到 DAC_DOR1（延时三个 APB 时钟周期）。

当产生 DAC 2 通道触发信号，DHR2 寄存器的内容传输到 DAC_DOR2（延时三个 APB 时钟周期）。

18.3.18.2 独立触发（生成单个 LFSR 波形）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为不同的值，以配置不同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 01，并在 MAMP_x[3:0]位中配置相同的 LFSR 掩码值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

当产生 DAC 1 通道触发信号，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内相加，所得总和传输到 DAC_DOR1 中（延时三个 APB 时钟周期）。LFSR1 计数器随即更新。

当产生 DAC 2 通道触发信号，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和传输到 DAC_DOR2 中（延时三个 APB 时钟周期）。LFSR2 计数器随即更新。

18.3.18.3 独立触发（生成不同 LFSR 波形）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为不同的值，以配置不同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 01，并在 MAMP1[3:0]和 MAMP2[3:0]位中设置不同的 LFSR 掩码值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

当产生 DAC 1 通道触发信号，LFSR1 计数器内容（使用 MAMP1[3:0]配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（延时三个 APB 时钟周期）。LFSR1 计数器随即更新。

当产生 DAC 2 通道触发信号，LFSR2 计数器内容（使用 MAMP2[3:0]配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（延时三个 APB 时钟周期）。LFSR2 计数器随即更新。

18.3.18.4 独立触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为不同的值，以配置不同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 1x，并在 MAMP_x[3:0]位中配置相同的最大振幅值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

当产生 DAC 1 通道触发信号，DAC 1 通道三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（延时三个 APB 时钟周期）。DAC 1 通道三角波计数器随即更新。

当产生 DAC 2 通道触发信号，DAC 2 通道三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（延时三个 APB 时钟周期）。DAC 2 通道三角波计数器随即更新。

18.3.18.5 独立触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为不同的值，以配置不同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 1x，并在 MAMP1[3:0]和 MAMP2[3:0]位中设置不同的最大振幅值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

当产生 DAC 1 通道触发信号，DAC 1 通道三角波计数器内容（使用 MAMP1[3:0]配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（延时三个 APB 时钟周期）。DAC 1 通道三角波计数器随即更新。

当产生 DAC 2 通道触发信号，DAC 2 通道三角波计数器内容（使用 MAMP2[3:0]配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（延时三个 APB 时钟周期）。DAC 2 通道三角波计数器随即更新。

18.3.18.6 同步软件启动

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

在此配置中，DHR1 和 DHR2 寄存器内容会在一个 APB 时钟周期后分别传输到 DAC_DOR1 和 DAC_DOR2 中。

18.3.18.7 同步触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为相同的值，为两个 DAC 通道配置相同的触发源。
- 3) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

当产生触发信号，DHR1 和 DHR2 寄存器内容将分别传输到 DAC_DOR1 和 DAC_DOR2 中（延时三个 APB 时钟周期）。

18.3.18.8 同步触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为相同的值，为两个 DAC 通道配置相同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 01，并在 MAMP_x[3:0]位中配置相同的 LFSR 掩码值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

当产生触发信号，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和传输到 DAC_DOR1 中（延时三个 APB 时钟周期）。LFSR1 计数器随即更新。同时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和传输到 DAC_DOR2 中（延时三个 APB 时钟周期）。LFSR2 计数器随即更新。

18.3.18.9 同步触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下步骤：

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为相同的值，为两个 DAC 通道配置相同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 01，并在 MAMP1[3:0]和 MAMP2[3:0]位中设置不同的 LFSR 掩码值。

- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器 (DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

当产生触发信号, LFSR1 计数器内容 (使用 MAMP1[3:0]配置的掩码) 与 DHR1 寄存器内容相加, 所得总和传输到 DAC_DOR1 中 (延时三个 APB 时钟周期)。LFSR1 计数器随即更新。

同时, LFSR2 计数器内容 (使用 MAMP2[3:0]配置的掩码) 与 DHR2 寄存器内容相加, 所得总和传输到 DAC_DOR2 中 (延时三个 APB 时钟周期)。LFSR2 计数器随即更新。

18.3.18.10 同步触发 (生成单个三角波)

要将 DAC 配置为此转换模式, 需要遵循以下步骤:

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为相同的值, 为两个 DAC 通道配置相同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 1_x, 并在 MAMP_x[3:0]位中配置相同的最大振幅值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器 (DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

当产生触发信号, DAC 1 通道三角波计数器内容 (使用相同的三角波振幅) 与 DHR1 寄存器内容相加, 所得总和传输到 DAC_DOR1 中 (延时三个 APB 时钟周期)。DAC 1 通道三角波计数器随即更新。

同时, DAC 2 通道三角波计数器内容 (使用相同的三角波振幅) 与 DHR2 寄存器内容相加, 所得总和传输到 DAC_DOR2 中 (延时三个 APB 时钟周期)。DAC 2 通道三角波计数器随即更新。

18.3.18.11 同步触发 (生成不同三角波)

要将 DAC 配置为此转换模式, 需要遵循以下步骤:

- 1) 将两个 DAC 通道触发使能位 TRIGEN1 和 TRIGEN2 置 1。
- 2) 将 TSEL1[2:0]和 TSEL2[2:0]设置为相同的值, 为两个 DAC 通道配置相同的触发源。
- 3) 将两个 DAC 通道的 WAVE_x[1:0]设置为 1_x, 并在 MAMP1[3:0]和 MAMP2[3:0]位中设置不同的最大振幅值。
- 4) 将 DAC 双通道数据加载到所需 DHR 寄存器 (DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD)。

当产生触发信号到, DAC 1 通道三角波计数器内容 (使用 MAMP1[3:0]配置的

三角波振幅)与 DHR1 寄存器内容相加,所得总和传输到 DAC_DOR1 中(延时三个 APB 周期)。DAC 1 通道三角波计数器随即更新。

同时, DAC 2 通道三角波计数器内容(使用 MAMP2[3:0]配置的三角波振幅)与 DHR2 寄存器内容相加,所得总和传输到 DAC_DOR2 中(延时三个 APB 时钟周期)。DAC 2 通道三角波计数器随即更新。

18.4 DAC 中断

表 18-5 DAC 中断

中断事件	事件标志	使能控制位
DMA 溢出	DMAUDRx	DMAUDRIEx

18.5 SAC 寄存器

18.5.1 DAC 控制寄存器 (DAC_CR)

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
SVREF2	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[3:0]				TEN2	EN2
rw		rw	rw	rw	rw	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
SVREF1	Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[3:0]				TEN1	EN1
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	SVREF2	DAC2 参考源选择。 0: AVDD, 1: VREF
30	保留	必须保持复位值
29	DMAUDRIE2	DAC2通道DMA溢出中断使能 此位由软件置1和清零。 0: 禁止DAC2通道DMA溢出中断。 1: 使能DAC2通道DMA溢出中断。
28	DMAEN2	DAC2通道DMA模式使能 此位由软件置1和清零。 0: 禁止DAC2通道DMA模式。 1: 使能DAC2通道DMA模式。
27:24	MAMP2[3:0]	DAC2通道掩码/振幅选择器 这些位由软件写入,用于在生成噪声波模式下选择掩码,或者在生成三角波模式下选择振幅。

		0000: 不屏蔽LFSR的位0/三角波振幅等于1
		0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3
		0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7
		0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15
		0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31
		0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63
		0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127
		0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255
		1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511
		1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023
		1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047
		≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095
23:22	WAVE2[1:0]	<p>DAC2通道噪声波/三角波生成器使能</p> <p>此位由软件置1和清零。</p> <p>这些位由软件置1或清零。</p> <p>00: 禁止生成波</p> <p>01: 使能生成噪声波</p> <p>1x: 使能生成三角波</p> <p><i>注意: 只在位TEN2=1 (使能DAC 2通道触发) 时使用。</i></p>
21:18	TSEL2[3:0]	<p>DAC2通道触发器选择</p> <p>这些位用于选择DAC 2通道的外部触发事件</p> <p>有关触发配置和映射的详细信息, 请参见触发模式表。</p> <p><i>注意: 只在位TEN2=1 (使能DAC 2通道触发) 时使用。</i></p>
17	TEN2	<p>DAC2通道触发使能</p> <p>此位由软件置1和清零, 以使能/禁止DAC 2通道触发</p> <p>0: 禁止DAC 2通道触发, 写入DAC_DHR2寄存器的数据在延时一个APB时钟周期之后传输到DAC_DOR2寄存器。</p> <p>1: 使能DAC 2通道触发, DAC_DHR2寄存器的数据在延时三个APB时钟周期之后传输到DAC_DOR2寄存器。</p> <p><i>注意: 如果选择软件触发, DAC_DHR2寄存器的内容只需延时一个APB时钟周期即可传输到DAC_DOR2寄存器。</i></p>
16	EN2	<p>DAC2通道使能</p> <p>此位由软件置1和清零, 以使能/禁止DAC2通道。</p> <p>0: 禁止DAC2通道。</p> <p>1: 使能DAC2通道。</p>
15	SVREF1	<p>DAC1 参考源选择。</p> <p>0: AVDD, 1: VREF</p>
14	保留	<p>必须保持复位值</p>

13	DMAUDRIE1	<p>DAC1通道DMA溢出中断使能</p> <p>此位由软件置1和清零。</p> <p>0: 禁止DAC1通道DMA溢出中断。</p> <p>1: 使能DAC1通道DMA溢出中断。</p>
12	DMAEN1	<p>DAC1通道DMA模式使能</p> <p>此位由软件置1和清零。</p> <p>0: 禁止DAC1通道DMA模式。</p> <p>1: 使能DAC1通道DMA模式。</p>
11:9	MAMP1[3:0]	<p>DAC1通道掩码/振幅选择器</p> <p>这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。</p> <p>0000: 不屏蔽LFSR的位0/三角波振幅等于1</p> <p>0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3</p> <p>0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7</p> <p>0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15</p> <p>0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31</p> <p>0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63</p> <p>0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127</p> <p>0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255</p> <p>1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511</p> <p>1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023</p> <p>1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047</p> <p>≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095</p>
7:6	WAVE1[1:0]	<p>DAC1通道噪声波/三角波生成器使能</p> <p>此位由软件置1和清零。</p> <p>这些位由软件置1或清零。</p> <p>00: 禁止生成波</p> <p>01: 使能生成噪声波</p> <p>1x: 使能生成三角波</p> <p><i>注意: 只在位TEN1=1 (使能DAC1通道触发) 时使用。</i></p>
5:2	TSEL1[3:0]	<p>DAC1通道触发器选择</p> <p>这些位用于选择DAC1通道的外部触发事件</p> <p>有关触发配置和映射的详细信息，请参见触发模式表。</p> <p><i>注意: 只在位TEN1=1 (使能DAC1通道触发) 时使用。</i></p>
1	TEN1	<p>DAC1通道触发使能</p> <p>此位由软件置1和清零，以使能/禁止DAC1通道触发</p> <p>0: 禁止DAC1通道触发，写入DAC_DHR1寄存器的数据在延时一个APB时钟周期之后传输到DAC_DOR1寄存器。</p> <p>1: 使能DAC1通道触发，DAC_DHR1寄存器的数据在延时三个APB时钟周期之后传输到DAC_DOR1寄存器。</p>

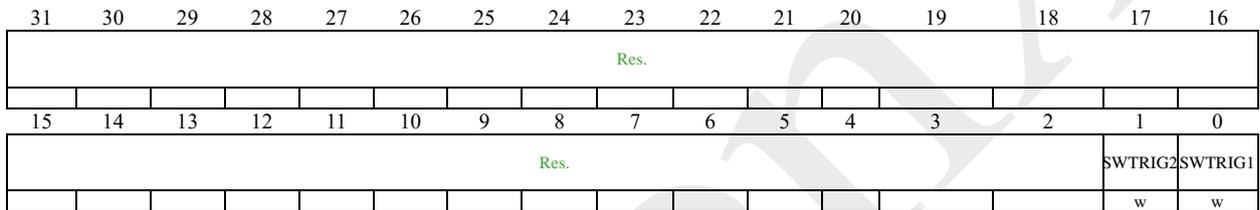
注意: 如果选择软件触发, DAC_DHR1 寄存器的内容只需延时一个APB时钟周期即可传输到DAC_DOR1 寄存器。

- 0 EN1 DAC1通道使能
 此位由软件置1和清零, 以使能/禁止DAC1通道。
 0: 禁止DAC1通道。
 1: 使能DAC1通道。

18.5.2 DAC 软件触发寄存器 (DAC_SWTRIGR)

偏移地址: 0x04

复位值: 0x0000 0000

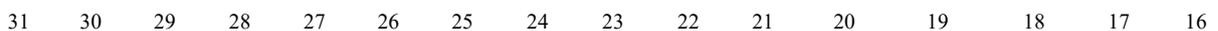


位/位域	名称	描述
31:4	保留	必须保持复位值。
1	SWTRIG2	DAC2通道软件触发 该位由软件置1, 用于在软件触发模式下触发DAC。 0: 不触发 1: 软件触发 注意: 一旦DAC_DHR2 寄存器值加载到DAC_DOR2 寄存器中, 该位即会由硬件清零 (延时一个APB时钟周期之后)。
0	SWTRIG1	DAC1通道软件触发 该位由软件置1, 用于在软件触发模式下触发DAC。 0: 不触发 1: 软件触发 注意: 一旦DAC_DHR1 寄存器值加载到DAC_DOR1 寄存器中, 该位即会由硬件清零 (延时一个APB时钟周期之后)。

18.5.3 DAC 1 通道 12 位右对齐数据保持寄存器(DAC_DHR12R1)

偏移地址: 0x08

复位值: 0x0000 0000



Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DACC1DHR[11:0]	DAC 1通道12位右对齐数据 这些位通过软件写入，用于指定DAC 1通道的12位数据。

18.5.4 DAC 1 通道 12 位左对齐数据保持寄存器(DAC_DHR12L1)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	DACC1DHR[11:0]	DAC 1通道12位左对齐数据。 这些位通过软件写入，用于指定DAC 1通道的12位数据。
3:0	保留	必须保持复位值。

18.5.5 DAC 1 通道 8 位右对齐数据保持寄存器(DAC_DHR8R1)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。

7:0 DACC1DHR[7:0] DAC 1通道8位右对齐数据
 这些位通过软件写入，用于指定DAC 1通道的8位数据。

18.5.6 DAC 2 通道 12 位右对齐数据保持寄存器(DAC_DHR12R2)

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DACC2DHR[11:0]	DAC 2通道12位右对齐数据 这些位通过软件写入，用于指定DAC 2通道的12位数据。

18.5.7 DAC 2 通道 12 位左对齐数据保持寄存器(DAC_DHR12L2)

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	DACC2DHR[11:0]	DAC 2通道12位左对齐数据。 这些位通过软件写入，用于指定DAC 2通道的12位数据。
3:0	保留	必须保持复位值。

18.5.8 DAC 2 通道 8 位右对齐数据保持寄存器(DAC_DHR8R2)

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DACC2DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	DACC2DHR[7:0]	DAC 2通道8位右对齐数据 这些位通过软件写入，用于指定DAC 2通道的8位数据。

18.5.9 DAC 双通道 12 位右对齐数据保持寄存器(DAC_DHR12RD)

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DACC2DHR[11:0]											
				rw	rw	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
Res.				DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	DACC2DHR[11:0]	DAC2通道12位右对齐数据 这些位由软件写入，用于为DAC2通道指定12位数据。
15:12	保留	必须保持复位值。
11:0	DACC1DHR[11:0]	DAC1通道12位右对齐数据 这些位由软件写入，用于为DAC1通道指定12位数据。

18.5.10 DAC 双通道 12 位左对齐数据保持寄存器(DAC_DHR12LD)

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.			
rw	rw	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
DACC1DHR[11:0]												Res.			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:20	DACC2DHR[11:0]	DAC 2通道12位左对齐数据 这些位由软件写入，用于为DAC 2通道指定12位数据。
19:16	保留	必须保持复位值。
15:4	DACC1DHR[11:0]	DAC 1通道12位左对齐数据 这些位通过软件写入，用于指定DAC 1通道的12位数据。
3:0	保留	必须保持复位值。

18.5.11 DAC 双通道 8 位右对齐数据保持寄存器(DAC_DHR8RD)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	DACC2DHR[7:0]	DAC 2通道8位右对齐数据 这些位由软件写入，用于为DAC2通道指定8位数据。
7:0	DACC1DHR[7:0]	DAC 1通道8位右对齐数据 这些位由软件写入，用于为DAC1通道指定8位数据。

18.5.12 DAC 1 通道数据输出寄存器(DAC_DOR1)

偏移地址：0x2C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DACC1DOR[11:0]	DAC 1通道数据输出 这些位为只读，其中包含DAC 1通道的数据输出。

18.5.13 DAC 2 通道数据输出寄存器(DAC_DOR2)

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DACC2DOR[11:0]	DAC 2通道数据输出 这些位为只读，其中包含DAC 2通道的数据输出。

18.5.14 DAC 状态寄存器(DAC_SR)

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDR2	Res.												
		rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDR1	Res.												
		rc_w1													

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DMAUDR2	DAC 2通道DMA溢出标志 此位由硬件置1，由软件写1清零。 0：DAC 2通道未发生DMA溢出错误状况。

1: DAC 2通道发生DMA溢出错误状况（当前所选触发源以高于DMA服务能力的频率驱动DAC 2通道转换）。

28:14 保留 必须保持复位值。

13 DMAUDR1 DAC 1通道DMA溢出标志
 此位由硬件置1，由软件清零（写入1）。
 0: DAC 1通道未发生DMA溢出错误状况。
 1: DAC 1通道发生DMA溢出错误状况（当前所选触发源以高于DMA服务能力的频率驱动DAC 1通道转换）。

12:0 保留 必须保持复位值。

18.5.15 DAC 模式控制寄存器(DAC_MCR)

偏移地址：0x3C

复位值：0x0000 0000

31								30								29								28								27								26								25								24								23								22								21								20								19								18								17								16							
Res.																MODE2[8:0]																																																																																																															
rw																rw																																																																																																															
15								14								13								12								11								10								9								8								7								6								5								4								3								2								1								0							
Res.																MODE1[8:0]																																																																																																															
rw																rw																																																																																																															

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:16	MODE2[8:0]	DAC2通道模式 只有在 DAC2 通道不使能的情况下才能写该位，详见表 18-4
15:9	保留	必须保持复位值。
8:0	MODE1[8:0]	DAC 1通道模式 只有在 DAC1 通道不使能的情况下才能写该位，详见表 18-4

18.5.16 DAC 控制寄存器 2 (DAC_CR2)

偏移地址：0x40

复位值：0x0000 0000

31				30				29				28				27				26				25				24				23				22				21				20				19				18				17				16			
SVR EF4		Res.		DMAU RIE4		DMAEN4		MAMP4[3:0]				WAVE4[1:]				TSEL4[3:0]				TEN4				EN4																																							
rw				rw		rw		rw				rw				rw				rw				rw																																							
15				14				13				12				11				10				9				8				7				6				5				4				3				2				1				0			
SVR EF3		Res.		DMAU DRIE3		DMAEN3		MAMP3[3:0]				WAVE3[1:0]				TSEL3[3:0]				TEN3				EN3																																							
rw				rw		rw		rw				rw				rw				rw				rw																																							

位/位域	名称	描述
31	SVREF4	DAC4 参考源选择。 0: AVDD, 1: VREF
28:22	保留	必须保持复位值。
29	DMAUDRIE4	DAC4通道DMA溢出中断使能 此位由软件置1和清零。 0: 禁止DAC4通道DMA溢出中断。 1: 使能DAC4通道DMA溢出中断。
28	DMAEN4	DAC4通道DMA模式使能 此位由软件置1和清零。 0: 禁止DAC4通道DMA模式。 1: 使能DAC4通道DMA模式。
27:24	MAMP4[3:0]	DAC4通道掩码/振幅选择器 这些位由软件写入, 用于在生成噪声波模式下选择掩码, 或者在生成三角波模式下选择振幅。 0000: 不屏蔽LFSR的位0/三角波振幅等于1 0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3 0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7 0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15 0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31 0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63 0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127 0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255 1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511 1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023 1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047 ≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095
23:22	WAVE4[1:0]	DAC4通道噪声波/三角波生成器使能 此位由软件置1和清零。 这些位由软件置1或清零。 00: 禁止生成波 01: 使能生成噪声波 1x: 使能生成三角波 <i>注意: 只在位TEN4=1 (使能DAC 4通道触发) 时使用。</i>
21:18	TSEL4[3:0]	DAC4通道触发器选择 这些位用于选择DAC 4通道的外部触发事件 有关触发配置和映射的详细信息, 请参见触发模式表。 <i>注意: 只在位TEN4=1 (使能DAC 4通道触发) 时使用。</i>

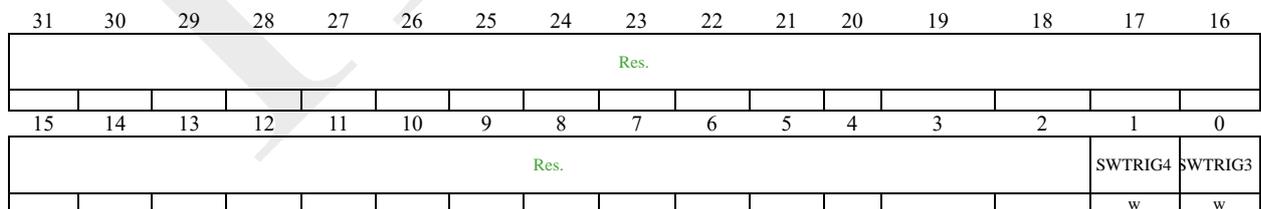
17	TEN4	<p>DAC4通道触发使能</p> <p>此位由软件置1和清零，以使能/禁止DAC 4通道触发</p> <p>0: 禁止DAC 4通道触发，写入DAC_DHR4寄存器的数据在延时一个APB时钟周期之后传输到DAC_DOR4寄存器。</p> <p>1: 使能DAC 4通道触发，DAC_DHR4寄存器的数据在延时三个APB时钟周期之后传输到DAC_DOR4寄存器。</p> <p><i>注意：如果选择软件触发，DAC_DHR4寄存器的内容只需延时一个APB时钟周期即可传输到DAC_DOR4寄存器。</i></p>
16	EN4	<p>DAC4通道使能</p> <p>此位由软件置1和清零，以使能/禁止DAC4通道。</p> <p>0: 禁止DAC4通道。</p> <p>1: 使能DAC4通道。</p>
15	SVREF3	<p>DAC3 参考源选择。</p> <p>0: AVDD, 1: VREF</p>
14	保留	<p>必须保持复位值。</p>
13	DMAUDRIE3	<p>DAC3通道DMA溢出中断使能</p> <p>此位由软件置1和清零。</p> <p>0: 禁止DAC3通道DMA溢出中断。</p> <p>1: 使能DAC3通道DMA溢出中断。</p>
12	DMAEN3	<p>DAC3通道DMA模式使能</p> <p>此位由软件置1和清零。</p> <p>0: 禁止DAC3通道DMA模式。</p> <p>1: 使能DAC3通道DMA模式。</p>
11:8	MAMP3[3:0]	<p>DAC3通道掩码/振幅选择器</p> <p>这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。</p> <p>0000: 不屏蔽LFSR的位0/三角波振幅等于1</p> <p>0001: 不屏蔽LFSR的位[1:0]/三角波振幅等于3</p> <p>0010: 不屏蔽LFSR的位[2:0]/三角波振幅等于7</p> <p>0011: 不屏蔽LFSR的位[3:0]/三角波振幅等于15</p> <p>0100: 不屏蔽LFSR的位[4:0]/三角波振幅等于31</p> <p>0101: 不屏蔽LFSR的位[5:0]/三角波振幅等于63</p> <p>0110: 不屏蔽LFSR的位[6:0]/三角波振幅等于127</p> <p>0111: 不屏蔽LFSR的位[7:0]/三角波振幅等于255</p> <p>1000: 不屏蔽LFSR的位[8:0]/三角波振幅等于511</p> <p>1001: 不屏蔽LFSR的位[9:0]/三角波振幅等于1023</p> <p>1010: 不屏蔽LFSR的位[10:0]/三角波振幅等于2047</p> <p>≥1011: 不屏蔽LFSR的位[11:0]/三角波振幅等于4095</p>

- 7:6 WAVE3[1:0] DAC3通道噪声波/三角波生成器使能
此位由软件置1和清零。
这些位由软件置1或清零。
00: 禁止生成波
01: 使能生成噪声波
1x: 使能生成三角波
注意: 只在位TEN3=1 (使能DAC 3通道触发) 时使用。
- 5:2 TSEL3[3:0] DAC3通道触发器选择
这些位用于选择DAC 3通道的外部触发事件
有关触发配置和映射的详细信息, 请参见触发模式表。
注意: 只在位TEN3=1 (使能DAC 3通道触发) 时使用。
- 1 TEN3 DAC4通道触发使能
此位由软件置1和清零, 以使能/禁止DAC3通道触发
0: 禁止DAC3通道触发, 写入DAC_DHR3寄存器的数据在延时一个APB时钟周期之后传输到DAC_DOR3寄存器。
1: 使能DAC 3通道触发, DAC_DHR3寄存器的数据在延时三个APB时钟周期之后传输到DAC_DOR3寄存器。
注意: 如果选择软件触发, DAC_DHR3寄存器的内容只需延时一个APB时钟周期即可传输到DAC_DOR3寄存器。
- 0 EN3 DAC3通道使能
此位由软件置1和清零, 以使能/禁止DAC3通道。
0: 禁止DAC3通道。
1: 使能DAC3通道。

18.5.17 DAC 软件触发寄存器 (DAC_SWTRIGR2)

偏移地址: 0x44

复位值: 0x0000 0000



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SWTRIG4	DAC4通道软件触发 该位由软件置1, 用于在软件触发模式下触发DAC。 0: 不触发

1: 软件触发

注意: 一旦DAC_DHR4寄存器值加载到DAC_DOR4寄存器中, 该位即会由硬件清零(延时一个APB时钟周期之后)。

0 SWTRIG3

DAC3通道软件触发

该位由软件置1, 用于在软件触发模式下触发DAC。

0: 不触发

1: 软件触发

注意: 一旦DAC_DHR3寄存器值加载到DAC_DOR3寄存器中, 该位即会由硬件清零(延时一个APB时钟周期之后)。

18.5.18 DAC 3 通道 12 位右对齐数据保持寄存器(DAC_DHR12R3)

偏移地址: 0x48

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DAC3DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DACC3DHR[11:0]	DAC 3通道12位右对齐数据 这些位通过软件写入, 用于指定DAC 3通道的12位数据。

18.5.19 DAC 4 通道 12 位右对齐数据保持寄存器(DAC_DHR12R4)

偏移地址: 0x54

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DAC4DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值。

11:0 DACC4DHR[11:0] DAC 4通道12位右对齐数据
 这些位通过软件写入，用于指定DAC 4通道的12位数据。

18.5.20 DAC 3 通道数据输出寄存器(DAC_DOR3)

偏移地址：0x6C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC3DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
------	----	----

31:12	保留	必须保持复位值。
-------	----	----------

11:0 DACC3DOR[11:0] DAC3通道数据输出
 这些位为只读，其中包含DAC3通道的数据输出。

18.5.21 DAC 4 通道数据输出寄存器(DAC_DOR4)

偏移地址：0x70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC4DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
------	----	----

31:12	保留	必须保持复位值。
-------	----	----------

11:0 DACC4DOR[11:0] DAC 4通道数据输出
 这些位为只读，其中包含DAC4通道的数据输出。

18.5.22 DAC 状态寄存器(DAC_SR2)

偏移地址：0x74

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DMAUDR 4	Res.												
		rc_wl													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		DMAUDR3	Res.												
		rc_w1													

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DMAUDR4	DAC 4通道DMA溢出标志 此位由硬件置1，由软件写1清零。 0: DAC 4通道未发生DMA溢出错误状况。 1: DAC 4通道发生DMA溢出错误状况（当前所选触发源以高于DMA服务能力的频率驱动DAC4通道转换）。
28:14	保留	必须保持复位值。
13	DMAUDR3	DAC 3通道DMA溢出标志 此位由硬件置1，由软件清零（写入1）。 0: DAC 3通道未发生DMA溢出错误状况。 1: DAC 3通道发生DMA溢出错误状况（当前所选触发源以高于DMA服务能力的频率驱动DAC 3通道转换）。
13:0	保留	必须保持复位值。

18.5.23 DAC 模式控制寄存器 2(DAC_MCR2)

偏移地址：0x7C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							MODE4[8:0]								
							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
Res.							MODE3[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:16	MODE4[8:0]	DAC4通道模式 只有在 DAC4 通道不使能的情况下才能写该位，详见表 18-4
15:9	保留	必须保持复位值。
8:0	MODE3[8:0]	DAC 3通道模式 只有在 DAC3 通道不使能的情况下才能写该位，详见表 18-4

18.5.24 COMP1 控制和状态寄存器 (COMP1_CSR)

偏移地址: 0x80

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	FE	FT[2:0]			Res.								
rw	r		rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	Res.	Res.	Res.	INPSEL[3:0]			INMSEL[3:0]			Res.			EN		
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw				rw

位/位域	名称	描述
31	LOCK	COMP1_CSR寄存器锁定位 此位由软件置1, 系统复位后清零。用于锁定COMP1控制寄存器 (COMP1_CSR [31:0])的全部内容。 0: COMP1_CSR [31:0]寄存器位可读/写 1: COMP1_CSR [31:0]寄存器位只读
30	VALUE	比较器1输出状态 此位只读。表示比较器的输出状态 (滤波及极性控制后的输出结果)。
29	保留	必须保持复位值。
28	FE	比较器1输出滤波使能 0: 禁止输出滤波功能 1: 使能输出滤波功能
27:25	FT[2:0]	比较器1输出滤波时钟周期选择 (PCLK时钟) 000: 2个时钟周期 001: 4个时钟周期 010: 8个时钟周期 011: 16个时钟周期 100: 32个时钟周期 101: 64个时钟周期 110: 128个时钟周期 111: 256个时钟周期 <i>注意: 滤波时间的配置只有在比较器1EN=1时才有效。</i>
25:16	保留	必须保持复位值。
15	POLARITY	比较器1输出极性选择

		0: 比较器1输出状态不反相 1: 比较器1输出状态反相
14	保留	必须保持复位值
13	保留	必须保持复位值
12	保留	必须保持复位值
11:8	INPSEL[3:0]	比较器1正相输入信号选择(具体定义看datasheet) 0000: INP1 0001: INP2 0010: INP3 0011: INP4 0100: INP5 0101: INP6 0110: INP7 0111: INP8 1000: INP9 1001: INP10 1111: OPAMP1 其它: 保留
7:4	INMSEL[3:0]	比较器1反相输入信号选择(具体定义看datasheet) 0000: INM1 0001: INM2 0010: INM3 0011: INM4 0100: INP5 0101: INM6 0110: INM7 0111: INM8 1000: INM9 1001: INM10 1111: DAC channel 1 其它: 保留
3:1	保留	必须保持复位值
0	EN	比较器1使能 0: 禁止比较器 1: 使能比较器

18.5.25 COMP2 控制和状态寄存器 (COMP2_CSR)

偏移地址: 0x84

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	FE	FT[2:0]			Res.								
rw	r		rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	Res.	Res.	Res.	INPSEL[3:0]			INMSEL[3:0]			Res.			EN		
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw				rw

位/位域	名称	描述
31	LOCK	COMP2_CSR寄存器锁定位 此位由软件置1, 系统复位后清零。用于锁定COMP2控制寄存器 (COMP2_CSR[31:0])的全部内容。 0: COMP2_CSR [31:0]寄存器位可读/写 1: COMP2_CSR [31:0]寄存器位只读
30	VALUE	比较器2输出状态 此位只读。表示比较器的输出状态 (滤波及极性控制后的输出结果)。
29	保留	必须保持复位值。
28	FE	比较器2输出滤波使能 0: 禁止输出滤波功能 1: 使能输出滤波功能
27:25	FT[2:0]	比较器2输出滤波时钟周期选择 (PCLK时钟) 000: 2个时钟周期 001: 4个时钟周期 010: 8个时钟周期 011: 16个时钟周期 100: 32个时钟周期 101: 64个时钟周期 110: 128个时钟周期 111: 256个时钟周期 <i>注意: 滤波时间的配置只有在比较器2EN=1时才有效。</i>
24:16	保留	必须保持复位值。
15	POLARITY	比较器2输出极性选择 0: 比较器输出状态不反相 1: 比较器输出状态反相
14	保留	必须保持复位值

13	保留	必须保持复位值
12	保留	必须保持复位值
11:8	INPSEL[3:0]	比较器1正相输入信号选择(具体定义看datasheet) 0000: INP1 0001: INP2 0010: INP3 0011: INP4 0100: INP5 0101: INP6 0110: INP7 0111: INP8 1000: INP9 1001: INP10 1111: OPAMP2 其它: 保留
7:4	INMSEL[3:0]	比较器1反相输入信号选择(具体定义看datasheet) 0000: INM1 0001: INM2 0010: INM3 0011: INM4 0100: INP5 0101: INM6 0110: INM7 0111: INM8 1000: INM9 1001: INM10 1111: DAC channel 2 其它: 保留
3:1	保留	必须保持复位值。
0	EN	比较器2使能 0: 禁止比较器 1: 使能比较器

18.5.26 COMP3 控制和状态寄存器 (COMP3_CSR)

偏移地址: 0x88

复位值: 0x0000 0000

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

LOCK	VALUE	Res.	FE	FT[2:0]				Res.									
rw	r		rw	rw	rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
POLARITY	Res.	Res.	Res.	INPSEL[3:0]				INMSEL[3:0]				Res.			EN		
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw						rw

位/位域	名称	描述
31	LOCK	COMP3_CSR寄存器锁定位 此位由软件置1，系统复位后清零。用于锁定COMP3控制寄存器 (COMP3_CSR[31:0])的全部内容。 0: COMP3_CSR [31:0]寄存器位可读/写 1: COMP3_CSR [31:0]寄存器位只读
30	VALUE	比较器3输出状态 此位只读。表示比较器的输出状态（滤波及极性控制后的输出结果）。
29	保留	必须保持复位值。
28	FE	比较器3输出滤波使能 0: 禁止输出滤波功能 1: 使能输出滤波功能
27:25	FT[2:0]	比较器3输出滤波时钟周期选择（PCLK时钟） 000: 2个时钟周期 001: 4个时钟周期 010: 8个时钟周期 011: 16个时钟周期 100: 32个时钟周期 101: 64个时钟周期 110: 128个时钟周期 111: 256个时钟周期 <i>注意：滤波时间的配置只有在比较器3EN=1时才有效。</i>
24:16	保留	必须保持复位值。
15	POLARITY	比较器3输出极性选择 0: 比较器输出状态不反相 1: 比较器输出状态反相
14	保留	必须保持复位值
13	保留	必须保持复位值
12	保留	必须保持复位值

- 11:8 INPSEL[3:0] 比较器1正相输入信号选择(具体定义看datasheet)
 0000: INP1
 0001: INP2
 0010: INP3
 0011: INP4
 0100: INP5
 0101: INP6
 0110: INP7
 0111: INP8
 1000: INP9
 1001: INP10
 1111: OPAMP3
 其它: 保留
- 7:4 INMSEL[3:0] 比较器1反相输入信号选择(具体定义看datasheet)
 0000: INM1
 0001: INM2
 0010: INM3
 0011: INM4
 0100: INP5
 0101: INM6
 0110: INM7
 0111: INM8
 1000: INM9
 1001: INM10
 1111: DAC channel 3
 其它: 保留
- 3:1 保留 必须保持复位值。
- 0 EN 比较器3使能
 0: 禁止比较器
 1: 使能比较器

18.5.27 COMP4 控制和状态寄存器 (COMP4_CSR)

偏移地址: 0x8C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	FE	FT[2:0]				Res.							
rw	r		rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY	Res.	Res.	Res.	INPSEL[2:0]				INMSEL[2:0]						EN	
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw				rw

位/位域	名称	描述
31	LOCK	COMP4_CSR 寄存器锁定位 此位由软件置1，系统复位后清零。用于锁定COMP4控制寄存器 (COMP4_CSR[31:0])的全部内容。 0: COMP4_CSR [31:0]寄存器位可读/写 1: COMP4_CSR [31:0]寄存器位只读
30	VALUE	比较器4输出状态 此位只读。表示比较器的输出状态（滤波及极性控制后的输出结果）。
29	保留	必须保持复位值。
28	FE	比较器4输出滤波使能 0: 禁止输出滤波功能 1: 使能输出滤波功能
27:25	FT[2:0]	比较器4输出滤波时钟周期选择（PCLK时钟） 000: 2个时钟周期 001: 4个时钟周期 010: 8个时钟周期 011: 16个时钟周期 100: 32个时钟周期 101: 64个时钟周期 110: 128个时钟周期 111: 256个时钟周期 <i>注意：滤波时间的配置只有在比较器2EN=1时才有效。</i>
24:16	保留	必须保持复位值。
15	POLARITY	比较器4输出极性选择 0: 比较器输出状态不反相 1: 比较器输出状态反相
14	保留	必须保持复位值
13	保留	必须保持复位值
12	保留	必须保持复位值
11:8	INPSEL[3:0]	比较器1正相输入信号选择(具体定义看datasheet) 0000: INP1 0001: INP2 0010: INP3 0011: INP4

- 0100: INP5
 - 0101: INP6
 - 0110: INP7
 - 0111: INP8
 - 1000: INP9
 - 1001: INP10
 - 1111: OPAMP4
 - 其它: 保留
- 7:4 INMSEL[3:0] 比较器1反相输入信号选择(具体定义看datasheet)
- 0000: INM1
 - 0001: INM2
 - 0010: INM3
 - 0011: INM4
 - 0100: INP5
 - 0101: INM6
 - 0110: INM7
 - 0111: INM8
 - 1000: INM9
 - 1001: INM10
 - 1111: DAC channel 4
 - 其它: 保留
- 3:1 保留 必须保持复位值。
- 0 EN 比较器4使能
- 0: 禁止比较器
 - 1: 使能比较器

18.5.28 OPAMP1 控制和状态寄存器 (OPAMP1_CSR)

偏移地址: 0xC0

复位值: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK		Res.													
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPAEN	OPAINTCEN	SOPPLL	SOTVL	Res.			VP_SEL		VM_SEL	SOPG			SOPM		
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	LOCK	OPAMP1_CSR寄存器锁定位 此位由软件置1，系统复位后清零。用于OPAMP1控制寄存器(OPAMP1_CSR [31:0])的全部内容。

		0: OPAMP1_CSR [31:0]寄存器位可读/写 1: OPAMP1_CSR [31:0]寄存器位只读
30:16	保留	必须保持复位值
15	OPAEN	OPAMP1使能 0: 不使能, 1: 使能 当使能OPAMP1, 软件需等待5us让OPAMP启动
14	OPAINTOEN	OPAMP1输出到PAD使能 0: 不使能, 1: 使能
13	SOPPLL	OPAMP1正相输入拉低使能。 0: 不使能。 1: 使能。
12	SOTVL	OPAMP1校正源选择 0: VSSA 1: VBGO
11:9	保留	必须保持复位值
8:7	VP_SEL	OPAMP1正相输入信号选择 00: PC9 01: PA13 10: PA12 11: PA12
6	VM_SEL	OPAMP1负相输入信号选择 0: PD10 1: PC10
5:3	SOPG	OPAMP1输入增益选择 000: 20倍 001: 25倍 010: 30倍 011: 35倍 100: 100倍 101: 105倍 110: 110倍 111: 115倍
2:0	SOPM	OPAMP1操作模式 000: 校正模式1 001: 校正模式2

- 010: 正常模式, 负相输入
- 011: 正常模式, 正相输入
- 100: 正常模式, 负相输入, 正相带1K欧姆电阻
- 110: 反馈断开, 开环放大
- 其它: 保留

18.5.29 OPAMP2 控制和状态寄存器 (OPAMP2_CSR)

偏移地址: 0xC4

复位值: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LOCK		Res.														
rw																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OPAEN	OPAINTC EN	SOPPLL	SOTVL	Res.			VP_SEL	VM_SEL	SOPG				SOPM			
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	LOCK	OPAMP2_CSR 寄存器锁定位 此位由软件置1, 系统复位后清零。用于OPAMP2控制寄存器 (OPAMP2_CSR [31:0])的全部内容。 0: OPAMP2_CSR [31:0]寄存器位可读/写 1: OPAMP2_CSR [31:0]寄存器位只读
30:16	保留	必须保持复位值
15	OPAEN	OPAMP2使能 0: 不使能, 1: 使能 当使能OPAMP2, 软件需等待5us让OPAMP启动
14	OPAINTCEN	OPAMP2输出到PAD使能 0: 不使能, 1: 使能
13	SOPPLL	OPAMP2正相输入拉低使能。 0: 不使能。 1: 使能。
12	SOTVL	OPAMP2校正源选择 0: VSSA 1: VBGO
11:9	保留	必须保持复位值
8:7	VP_SEL	OPAMP2正相输入信号选择

		00: PC12
		01: PD8
		10: PD7
		11: PD7
6	VM_SEL	OPAMP2负相输入信号选择 0: PB7 1: PD9
5:3	SOPG	OPAMP2输入增益选择 000: 20倍 001: 25倍 010: 30倍 011: 35倍 100: 100倍 101: 105倍 110: 110倍 111: 115倍
2:0	SOPM	OPAMP2操作模式 000: 校正模式1 001: 校正模式2 010: 正常模式, 负相输入 011: 正常模式, 正相输入 100: 正常模式, 负相输入, 正相带1K欧姆电阻 110: 反馈断开, 开环放大 其它: 保留

18.5.30 OPAMP3 控制和状态寄存器 (OPAMP3_CSR)

偏移地址: 0xC8

复位值: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.														
r/w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPAEN	OPAINTO EN	SOPPLL	SOTVL	Res.			VP_SEL	VM_SEL	SOPG				SOPM		
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

位/位域	名称	描述
31	LOCK	OPAMP3_CSR寄存器锁定位 此位由软件置1, 系统复位后清零。用于OPAMP3控制寄存器(OPAMP3_CSR [31:0])的全部内容。 0: OPAMP3_CSR [31:0]寄存器位可读/写

		1: OPAMP3_CSR [31:0]寄存器位只读
30:16	保留	必须保持复位值
15	OPAEN	OPAMP3使能 0: 不使能, 1: 使能 当使能OPAMP3, 软件需等待5us让OPAMP启动
14	OPAINTOEN	OPAMP3输出到PAD使能 0: 不使能, 1: 使能
13	SOPPLL	OPAMP3正相输入拉低使能。 0: 不使能。 1: 使能。
12	SOTVL	OPAMP3校正源选择 0: VSSA 1: VBGO
11:9	保留	必须保持复位值
8:7	VP_SEL	OPAMP3正相输入信号选择 00: PB11 01: PB10 10: PB8 11: PB8
6	VM_SEL	OPAMP3负相输入信号选择 0: PB12 1: PB9
5:3	SOPG	OPAMP3输入增益选择 000: 20倍 001: 25倍 010: 30倍 011: 35倍 100: 100倍 101: 105倍 110: 110倍 111: 115倍
2:0	SOPM	OPAMP3操作模式 000: 校正模式1 001: 校正模式2 010: 正常模式, 负相输入

011: 正常模式, 正相输入
 100: 正常模式, 负相输入, 正相带1K欧姆电阻
 110: 反馈断开, 开环放大
 其它: 保留

18.5.31 OPAMP4 控制和状态寄存器 (OPAMP4_CSR)

偏移地址: 0xCC

复位值: 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPAEN	OPAINTOEN	SOPPLL	SOTVL	Res.			VP_SEL		VM_SEL	SOPG			SOPM		
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	LOCK	OPAMP4_CSR 寄存器锁定位 此位由软件置1, 系统复位后清零。用于OPAMP4控制寄存器 (OPAMP4_CSR [31:0])的全部内容。 0: OPAMP4_CSR [31:0]寄存器位可读/写 1: OPAMP4_CSR [31:0]寄存器位只读
30:16	保留	必须保持复位值
15	OPAEN	OPAMP4使能 0: 不使能, 1: 使能 当使能OPAMP4, 软件需等待5us让OPAMP启动
14	OPAINTOEN	OPAMP4输出到PAD使能 0: 不使能, 1: 使能
13	SOPPLL	OPAMP4正相输入拉低使能。 0: 不使能。 1: 使能。
12	SOTVL	OPAMP4校正源选择 0: VSSA 1: VBGO
11:9	保留	必须保持复位值
8:7	VP_SEL	OPAMP4正相输入信号选择 00: PA15

		01: PD3 10: PD2 11: PD2
6	VM_SEL	OPAMP4负相输入信号选择 0: PB13 1: PD4
5:3	SOPG	OPAMP4输入增益选择 000: 20倍 001: 25倍 010: 30倍 011: 35倍 100: 100倍 101: 105倍 110: 110倍 111: 115倍
2:0	SOPM	OPAMP4操作模式 000: 校正模式1 001: 校正模式2 010: 正常模式, 负相输入 011: 正常模式, 正相输入 100: 正常模式, 负相输入, 正相带1K欧姆电阻 110: 反馈断开, 开环放大 其它: 保留

19 液晶显示/发光二极管控制器 (LCD/LED)

19.1 简介

该模块可以配置三种驱动模式：LCD 驱动模式，LED 矩阵模式，LED 点阵模式。通过寄存器配置，它仅同时支持一种操作模式。LCD 控制器是一款适用于单色无源液晶显示器（段码 LCD）的数字控制器/驱动器，最多具有 8 个公用端（COM）和 28 个区段端（SEG），用以驱动 112 (4x28) 或 192 (8x24) 个 LCD 图像元素（像素或完整符号）。公用和区段端的确切数量取决芯片封装及引脚规模。

LCD 由若干图像元素组成，这些图像元素均可点亮或熄灭。每个图像元素都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，相应的图像元素可见。

图像元素两电极间施加到液晶的电压，称为显示驱动电压。显示驱动电压必须为交流，以避免液晶中出现电泳效应（这将影响显示效果），因此必须在两电极间以特定的驱动波形施加电压，以避免出现直流（DC）。

词汇表	
LCD	单色无源显示器，带有直接引向图像元素电极的端子。
LED	发光二极管，一种常用的发光器件。
COM	连接到多个图像元素（24 个）的公用电气连接端子。
SEG	连接到图像元素的区段电气连接端子。
Bias	驱动 LCD 时使用的电压等级。定义为 $1/($ 用于驱动 LCD 显示器的电压等级数 $- 1)$ 。
Duty	定义为 $1/($ 给定 LCD 显示器上的公用端子数 $)$ 。
Frame	显示驱动波形的一个周期。
f_{frame}	每秒帧数，即，每秒激励 LCD 图像元素的次数。

19.2 LCD 主要特性

- 4x28~8x24 LCD 驱动器
- 支持 1/4、1/5、1/6 和 1/8 占空比
- 支持 1/3 偏置电压
- 多达 8 个 32 位 LCD RAM 数据显示存储器
- 可设置的 0~7 级显示对比度

- 未使用的 COM 和 SEG 引脚可配置为 I/O 或其它外设复用功能
- 支持低功耗模式
 - LCD 控制器可在 Sleep 和 Stop 下进行显示，也可以禁止以降低功耗。
- 内置反相功能以降低功耗和 EMI（电磁干扰）

19.3 LED 主要特性

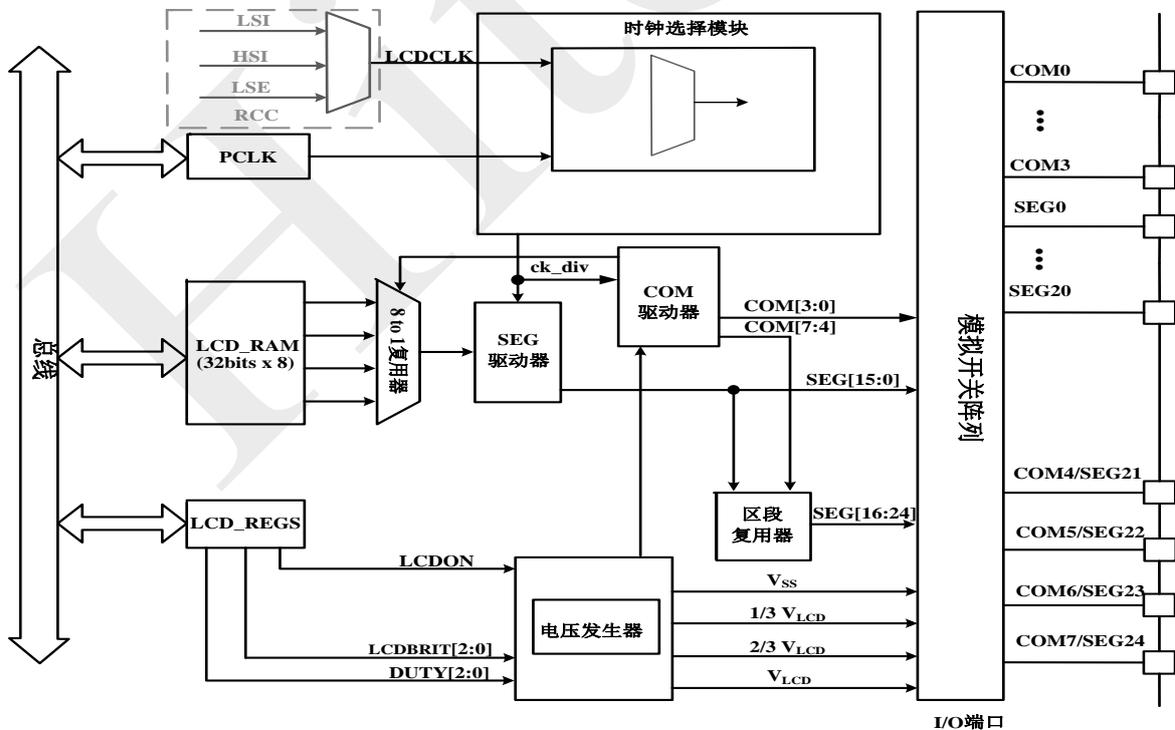
- 矩阵模式：8*8，16 引脚，最多 64 点
- 点矩阵模式：7*8、6*7、5*6、4*5，8 引脚，最高 56 点

19.4 功能描述

19.4.1 概述

LCD/LED 控制器主要包括 LCD 和 LED 两种模式的电路，注意由时钟选择、电压发生器、公共和区段控制器（COM/SEG）、LCD RAM 模块构成。（请参见下图）

图 19-1 LCD/LED 控制器结构框图



19.4.2 LED 矩阵模式

该芯片在矩阵模式下支持 LED 控制器和驱动器。如果 LEDMODE=00b, LCDON 和 SELLED=1, LED 矩阵模式将启用。它提供 8 个 segment 引脚和 8 个 Common 引脚来驱动 64 像素的 LED 模块，其中 COM 引脚具有较高的灌电流。可以通过 LCDBRIT 设置 LED 的亮度，当该位设置为 1111b 时，它是最高亮度。另外 LEDBRITM 用于设置亮度和均匀度，当 LEDBRITM=0 时，可以获得更好的显示均匀性。当 LEDBRITM=1 时，可以获得更好的显示亮度。

图 19-2 LED 矩阵图

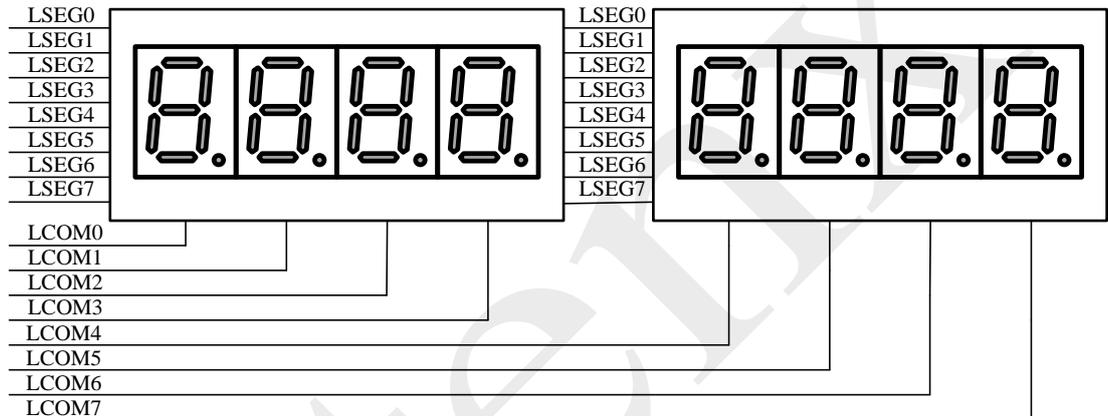


表 19-1 LED 矩阵驱动对应显示配置表

LCD_RAM	COM	Bit[7:0]
0x10	LCOM0	LSEG7~LSEG0
0x18	LCOM1	LSEG7~LSEG0
0x20	LCOM2	LSEG7~LSEG0
0x28	LCOM3	LSEG7~LSEG0
0x30	LCOM4	LSEG7~LSEG0
0x38	LCOM5	LSEG7~LSEG0
0x40	LCOM6	LSEG7~LSEG0
0x4C	LCOM7	LSEG7~LSEG0

LCD_RAM 中相应地址的每个数据位对应于每个 LED 点亮状态(1:点亮, 0:不点亮)

当使能 LED 开启 LCD_CR.LCDON=1、且 LED 矩阵模式 LCD_CR.LEDMODE=00 时, LED 矩阵模式将占用所使用管脚的管脚附加功能, 其他模块不可使用这些管脚的附加功能。包括 8 个 LSEG 管脚和多个 LCOM 管脚。

表 19-2 LED 矩阵模式时占用的附加功能管脚列表

序号	管脚名	LED 矩阵模式附加功能	CR.DU TY=000	CR.DU TY=001	CR.DU TY=010	CR.DU TY=011	CR.DU TY=100	CR.DU TY=101	CR.DUTY =110/111
1	PA6	LSE G7	√	√	√	√	√	√	√
2	PA7	LSE G6	√	√	√	√	√	√	√
3	PA8	LSE G5	√	√	√	√	√	√	√
4	PA9	LSE G4	√	√	√	√	√	√	√
5	PC4	LSE G3	√	√	√	√	√	√	√
6	PC5	LSE G2	√	√	√	√	√	√	√
7	PC6	LSE G1	√	√	√	√	√	√	√
8	PC7	LSE G0	√	√	√	√	√	√	√
9	PA5	LC OM 0	√	√	√	√	√	√	√
10	PA4	LC OM 1	√	√	√	√	√	√	√

1	P A 3	LC OM 2		√	√	√	√	√	√
1	P A 2	LC OM 3			√	√	√	√	√
1	P A 1	LC OM 4				√	√	√	√
1	P A 0	LC OM 5					√	√	√
1	P A 14	LC OM 6						√	√
1	P A 15	LC OM 7							√

19.4.3 LED 点阵模式

如果 LEDMODE=10b, LCDON 和 SELLED=1,则 LED 点阵模式启动。LED 点阵是通过 7*8 点矩阵。对应 LED0~LED7 端口, 最多可配置 7x8=56 个 LED 点进行驱动, LED 的对应位置在下图的 7*8 点矩阵中标记地址, LCD_RAM 中的显示配置对应于 LED 相应地址的状态(1 表示点亮, 0 表示不点亮)。支持多达 56 灯的 LED 驱动器。使用 DUTY 选择点矩阵 4*4、5*5、6*6、7*7 和 7*8,相应的 LED 地址保持不变。可以通过 LCDBRIT 设置 LED 的亮度, 设置为 1111b,它是最高亮度。另外 LEDBRITM 用于设置亮度和均匀度, 当 LEDBRITM=0 时, 可以获得更好的显示均匀性。当 LEDBRITM=1 时, 可以获得更好的显示亮度。

表 19-3 LED 点阵驱动对应显示配置表

LCD_RAM	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x10	7	6	5	4	3	2	1	0
0x18	15	14	13	12	11	10	9	8
0x20	23	22	21	20	19	18	17	16
0x28	31	30	29	28	27	26	25	24
0x30	39	38	37	36	35	34	33	32

LCD_RAM	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x38	47	46	45	44	43	42	41	40
0x40	55	54	53	52	51	50	49	48

LCD_RAM 中相应地址的每个数据位对应于每个 LED 点亮状态(1:点亮, 0:不点亮)

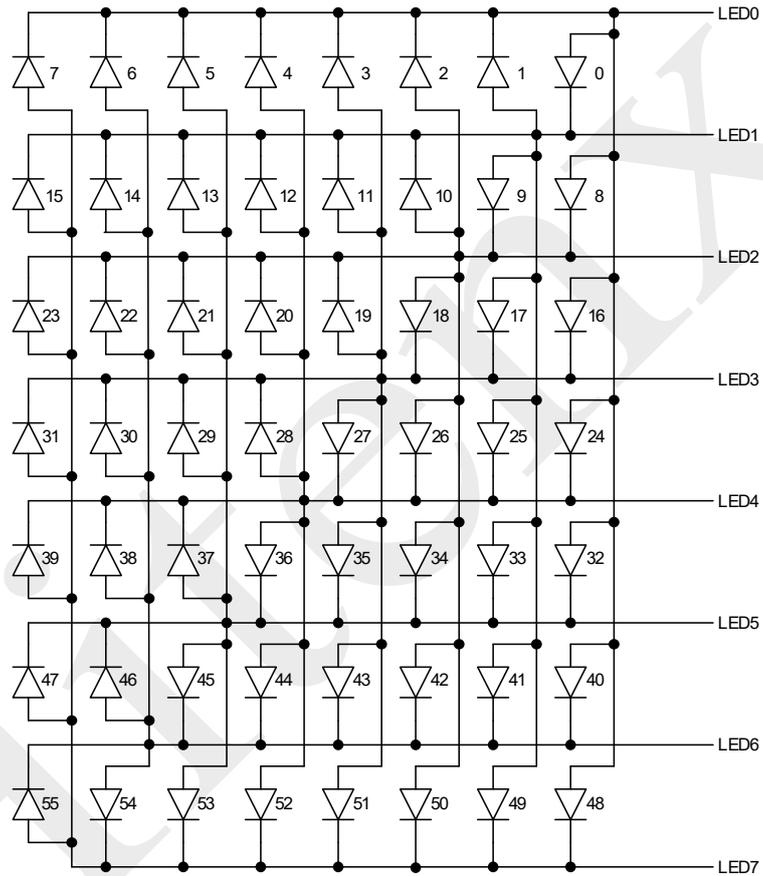
图 19-3 LED7*8 点阵图


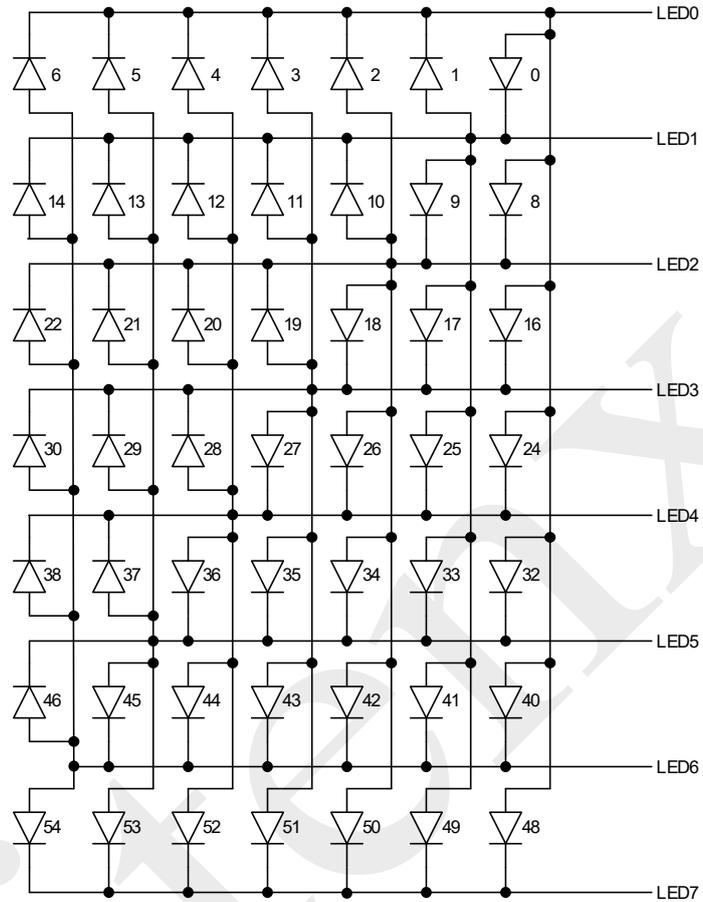
图 19-4 LED7*7 点阵图


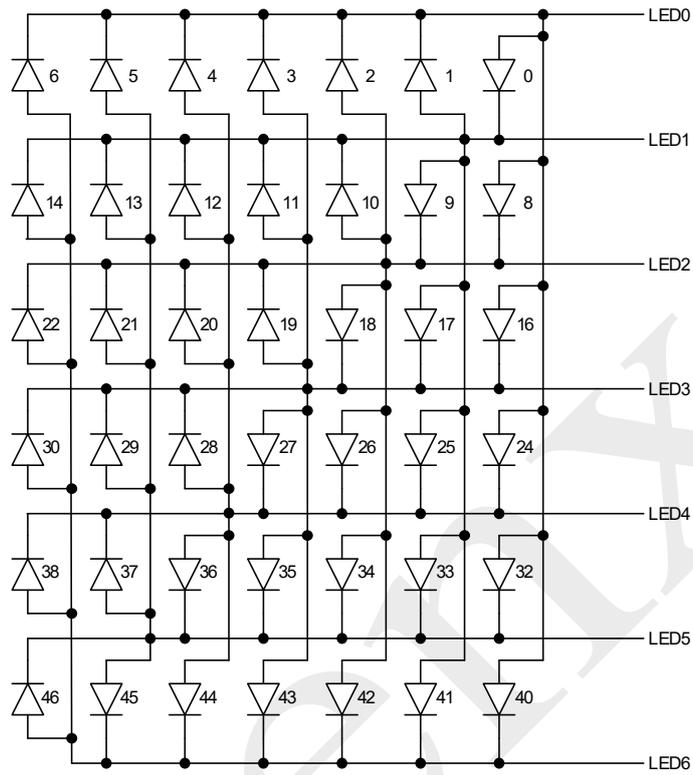
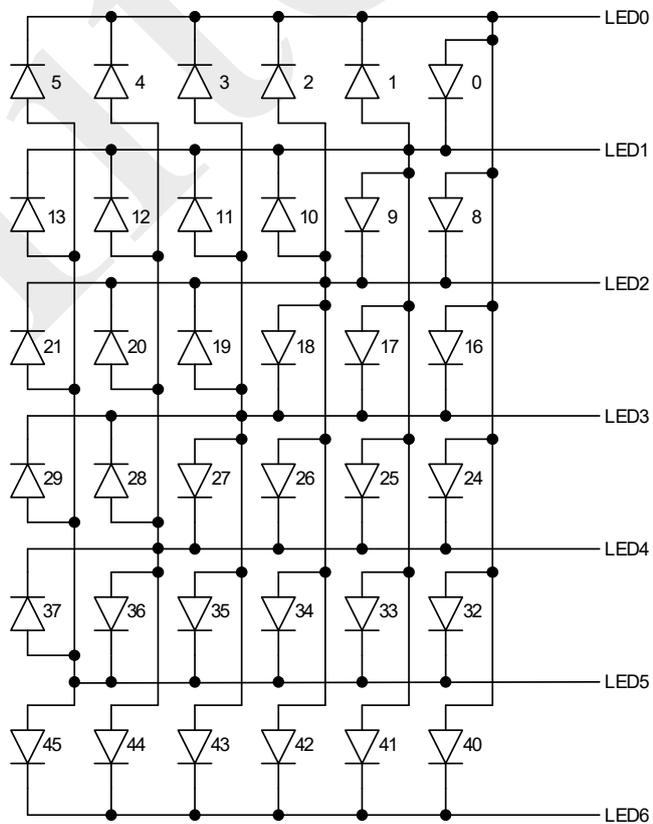
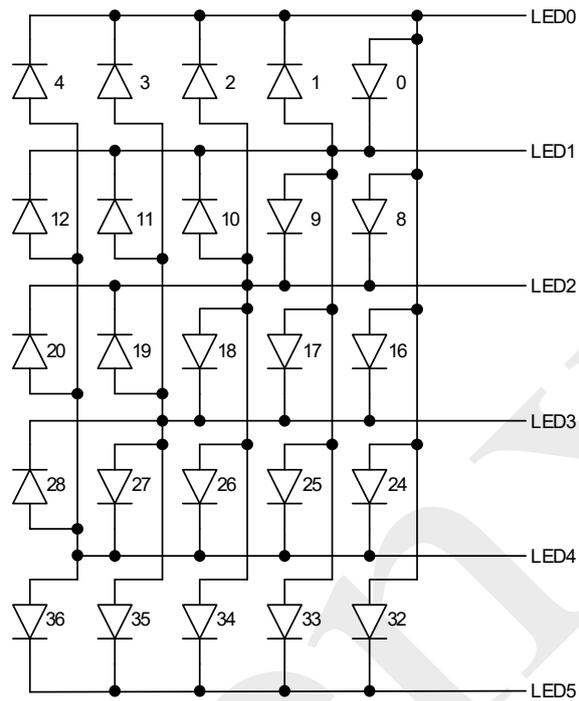
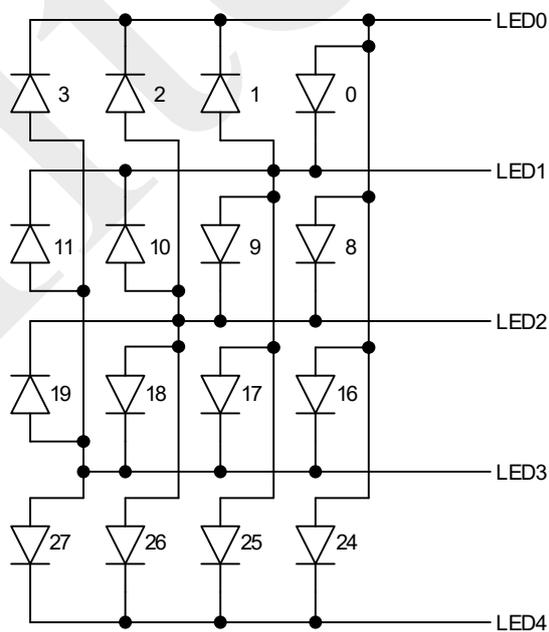
图 19-5 LED6*7 点阵图

图 19-6 LED6*6 点阵图


图 19-7 LED5*5 点阵图

图 19-8 LED4*4 点阵图


19.4.4 LCD 模式

该芯片支持 LCD 控制器和驱动器。LCD 驱动器能够选择 112 点(4COM/28SEG)

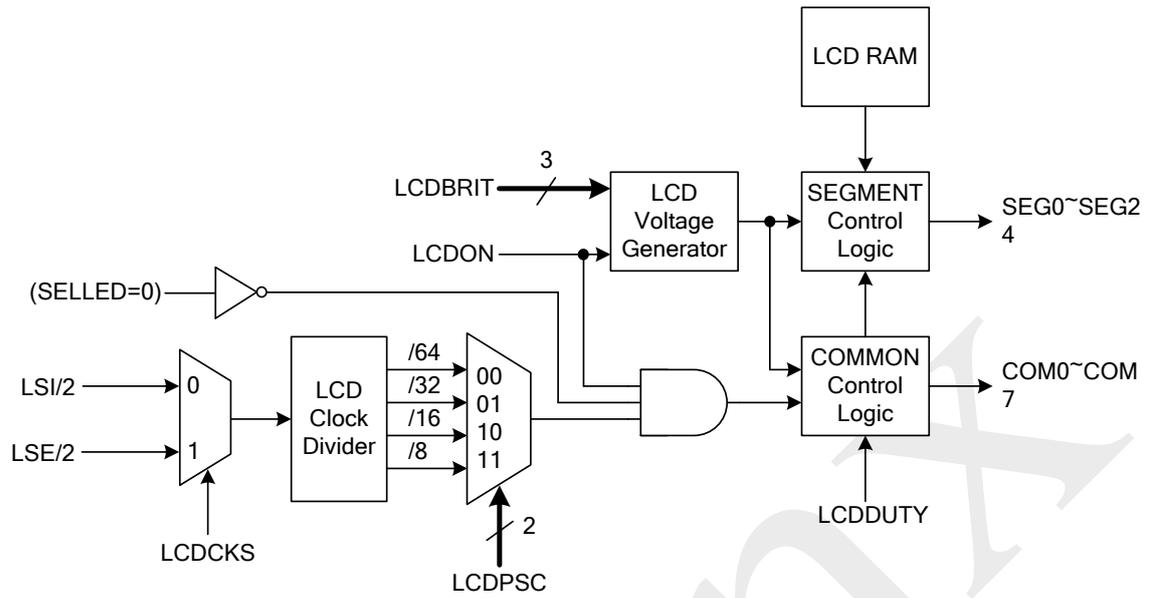
或 192 点(8COM/24SEG)来驱动 LCD 面板。它能够驱动 1/3 偏压。LCD 时钟源由 LSI/2 产生，取决于寄存器位 LCDCKS。时钟速率可以通过 LCDPSC 寄存器位选择除以 8、16、32 和 64。如果 LSI/2 是 LCD 时钟源，则 VCC 电压电平会影响 LSI 频率和 LCD 帧速率。

当使能 LCD 开启 LCD_CR.LCDON=1 时，LCD 模式将占用所使用管脚的管脚附加功能，其他模块不可使用这些管脚的附加功能。包括 32 个 SEG/COM 管脚。

表 19-4 LED 矩阵模式时占用的附加功能管脚列表

序号	管脚名	LCD 模式附加功能
1	PA0	COM5,SEG25
2	PA1	COM4,SEG24
3	PA2	COM3
4	PA3	COM2
5	PA4	COM1
6	PA5	COM0
7	PC6	SEG23
8	PC7	SEG22
9	PB14	SEG21
10	PB0	SEG20
11	PB1	SEG19
12	PB2	SEG18
13	PB5	SEG17
14	PB6	SEG16
15	PA10	SEG15
16	PA11	SEG14
17	PA12	SEG13

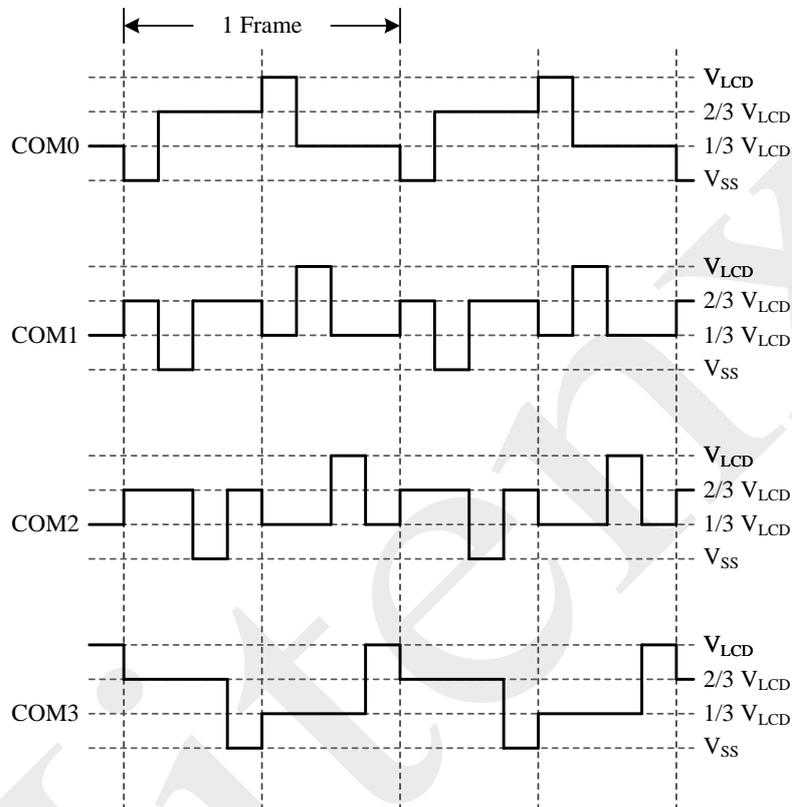
18	PA13	SEG12
19	PC9	SEG11
20	PC10	SEG10
21	PC11	SEG9
22	PC12	SEG8
23	PB7	SEG7
24	PB8	SEG6
25	PB9	SEG5
26	PB10	SEG4
27	PC13	SEG3
28	PB11	SEG2
29	PB12	SEG1
30	PC14	SEG0
31	PA14	COM7,SEG27
32	PA15	COM6,SEG26

图 19-9 LCD 模式控制器结构框图


19.4.4.1 LCD 驱动波形

LCD 控制器在 COM 引脚上输出的任何重复波形的频率都可以用来表示 LCD 帧速率，下图显示了一个 LCD 波形框架。

图 19-10 LCD 驱动波形框架



LCD 帧速率如下所示：

表 19-5 LCD 帧速率

LCDCKS	LCDPSC	LCD 帧率(Hz)			
		DUTY			
		1/4 占空比	1/5 占空比	1/6 占空比	1/8 占空比
LSI/2 (16384Hz)	/8	256	204	170	128
	/16	128	102	85	64
	/32	64	51	42	32
	/64	32	25	21	16

下图 LCD 电压发生器显示了由电阻组成的内部电压发生器。LCDON 和 SELLED 控制从 VCC 到地的电流。如果 LCDON=0 或 SELLED=1，则 PMOS 将关闭路径，以便所有 LCD 电压均为 0V。如果 LCDON=1 和 SELLED=0，则电阻分压器将产生多个电压，以提供 LCD 控制模块用于生成所需的波形。LCDBRIT 控制位将打开/短路开关以确定 V_{LCD} 。下表显示

了与 LCDBRIT 对应的 V_{LCD} 。由于 LCDON=1 和 SELLED=0 时 DC 路径始终接通，因此分压器电路将消耗电流。

图 19-11 LCD 电压发生器

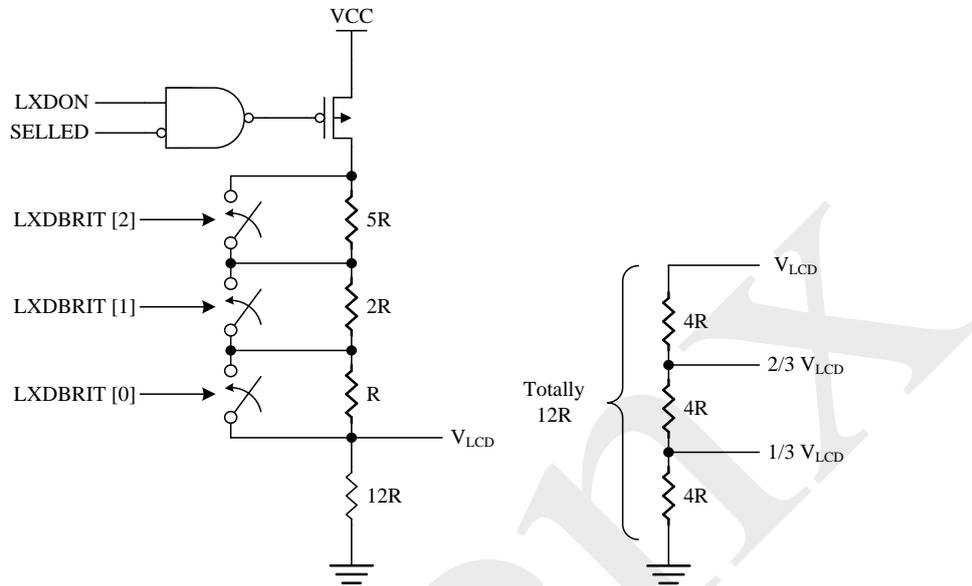
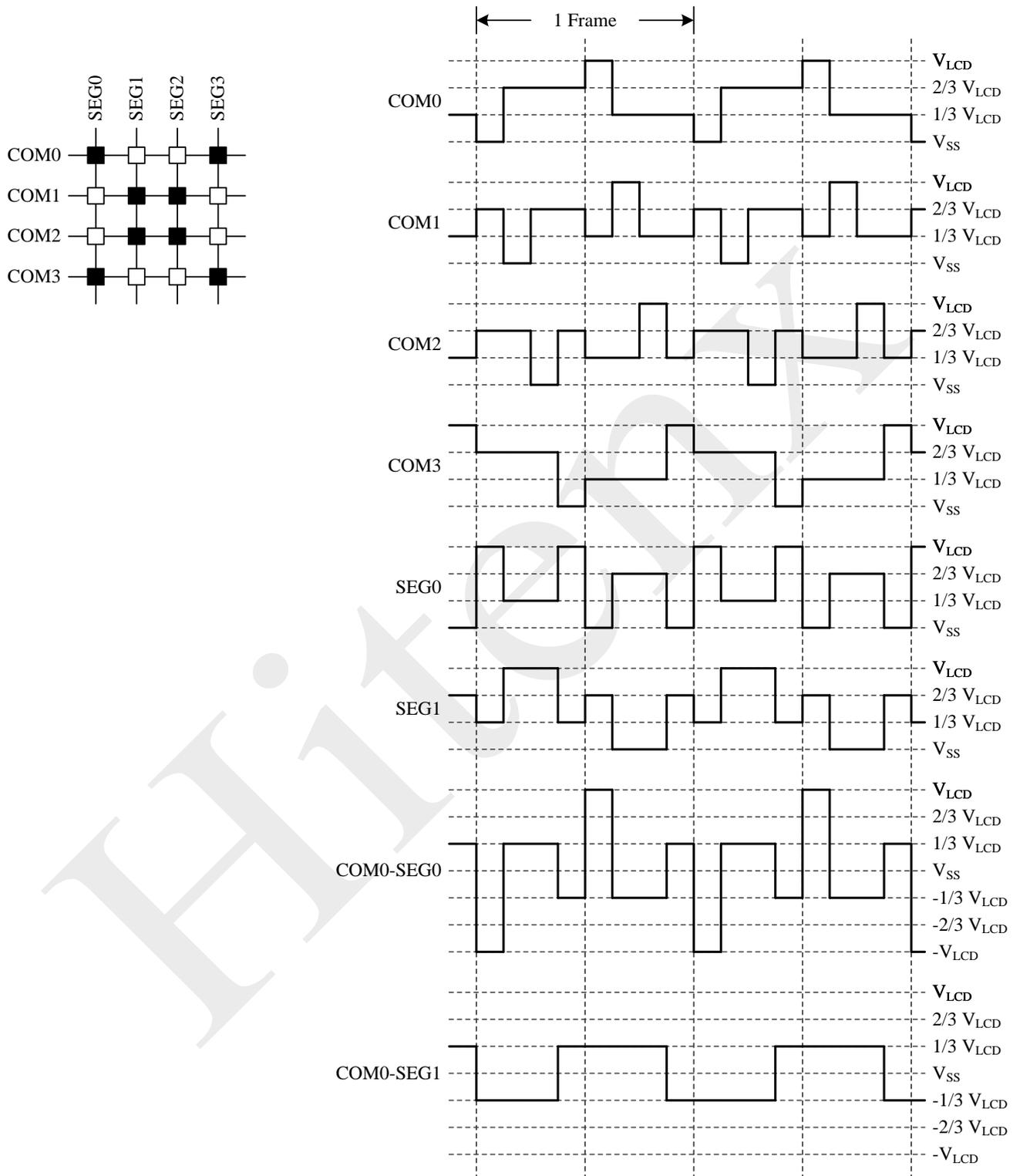


图 19-12 1/4Duty, 1/3Bias 驱动波形示例


LCD 显示模式

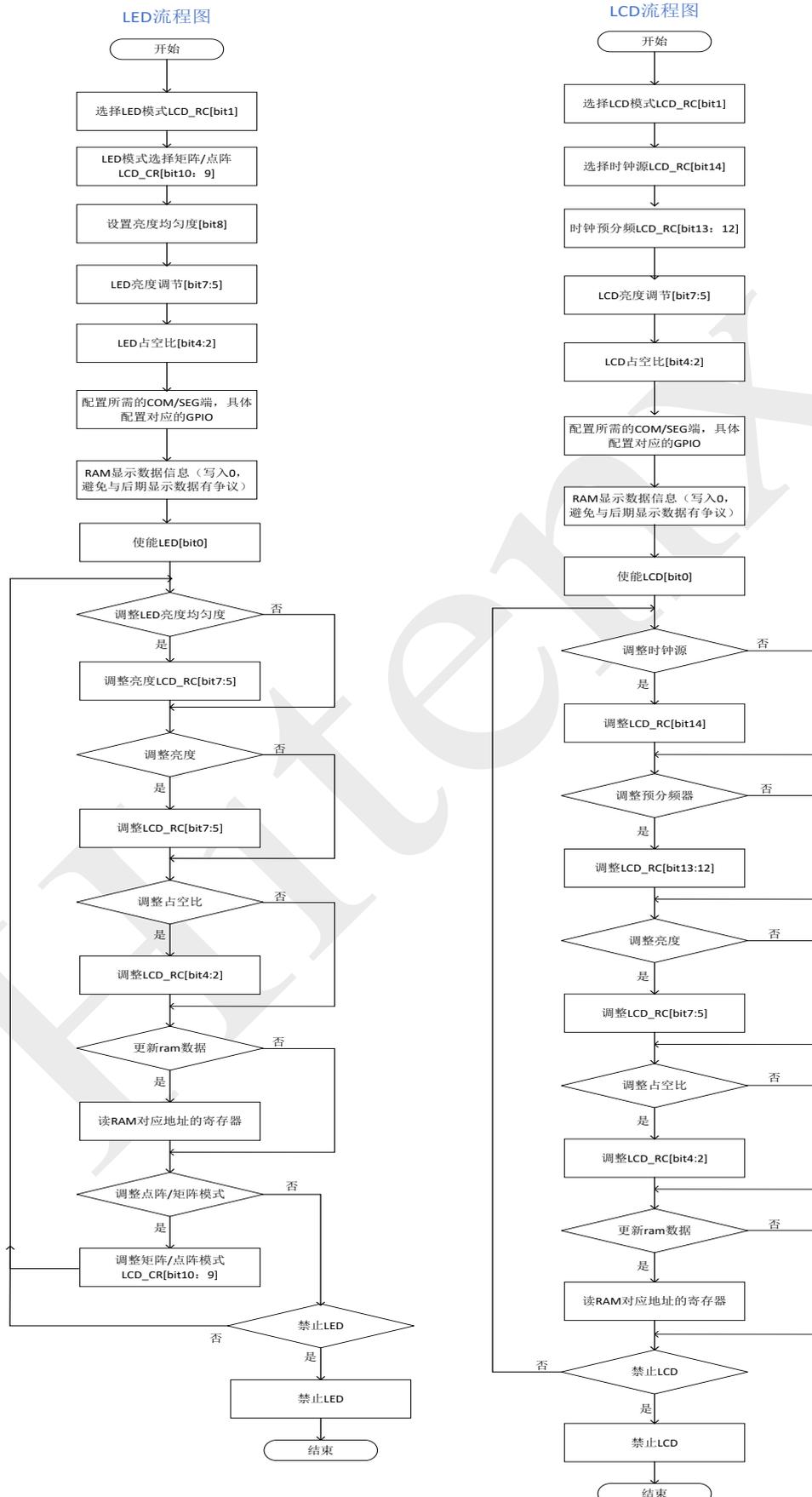
LCD 显示器上的图像元素由 COM 和 SEG 组成的信号矩阵控制，每个 COM 和 SEG 的交叉点对应一个图像元素，以 COM 为公共端，SEG 为区段端。LCD 控制器输出波形，驱动 LCD 显示时，以 SEG 端子为显示单元，即同一个 COM 的不同 SEG 在同一个字节中；表 19-6 给出了 COM、SEG 以及 LCD_RAM 对应关系。

表 19-6 LCD 控制器显示模式

COM	LCD_RAM	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM0	0x13~10	Res.				SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM1	0x1b~18	Res.				SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM2	0x23~20	Res.				SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM3	0x2b~28	Res.				SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM4	0x33~30	Res.				SEG27	SEG26	SEG25	Res.	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM5	0x3b~38	Res.				SEG27	SEG26	Res.	Res.	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COM6	0x43~40	Res.				SEG27	Res.	Res.	Res.	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

COM	LCD_RAM	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
COM7	0x4b~48	Res.									SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

HITENX

图 19-13 LED 和 LCD 驱动应用流程示例


19.5 低功耗模式

当时钟源选择 LSI 时，在低功耗模式下，LCD 控制器仍可驱动 LCD 液晶屏进行显示。不需要显示时可禁用以降低功耗。

表 19-7 低功耗模式下 LCD 控制器行为

低功耗模式	说明
Stop	LCD 控制器已激活，则保持激活状态
Sleep	

19.6 LCD 寄存器

19.6.1 LCD 控制寄存器 (LCD_CR)

偏移地址：0x00

复位值：0x000000e0

注意： 禁止 LCD 控制器操作，在驱动波形的下一扫描周期开始（当前扫描周期结束时生效，保证当前扫描周期的完整性，即保证加载到 LCD 上的驱动波形平均电压为 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LCDCKS	LCDPSC		LEDHOLD	LEDMODE		LEDBRITM	LCDBRIT			DUTY[2:0]			SELLED	LCDON
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14	LCDCKS	LCD 时钟源选择(注意：LED 时钟源固定为 HSI) 0：LSI/2 1：保留
13:12	LCDPSC	LCD/LED 时钟预分频器选择 00：LCDCKS 除以 64 01：LCDCKS 除以 32 10：LCDCKS 除以 16 11：LCDCKS 除以 8
11	LEDHOLD	LED 时钟保持 0：LED 扫描 1：LED 时钟保持

10:9	LEDMODE	LED模式选择 00: 矩阵模式 01: 保留 10: 点阵模式 11: 保留
8	LEDBRITM	LED亮度模式: 0: 均匀亮度模式(恒流驱动) 1: 亮度增强模式(电压驱动)
7:5	LCDBRIT	LCD/LED亮度控制 000: 0级 (最暗) 111: 7级 (最亮)
4:2	DUTY[2:0]	LCD/LED占空比选择: LCD选择(SELLED=0) 000: 1/4占空比(COM 0~3) 001: 1/4占空比(COM 0~3) 010: 1/5占空比(COM 0~4) 011: 1/6占空比(COM 0~5) 100: 1/6占空比(COM 0~5) 101: 1/8占空比(COM 0~7) 110: 1/8占空比(COM 0~7) 111: 1/8占空比(COM 0~7) LED选择:矩阵模式(SELLED=1, LEDMODE=00b) 000: 1/2占空比(LCOM 0~1) 001: 1/3占空比(LCOM 0~2) 010: 1/4占空比(LCOM 0~3) 011: 1/5占空比(LCOM 0~4) 100: 1/6占空比(LCOM 0~5) 101: 1/7占空比(LCOM 0~6) 110: 1/8占空比(LCOM 0~7) 111: 1/8占空比(LCOM 0~7) LED选择:点阵模式(SELLED=1, LEDMODE=10b) 000: 4x4, (LED 0~4) 001: 5x5, (LED 0~5) 010: 6x6, (LED 0~6) 011: 6x7, (LED 0~6) 100: 7x7, (LED 0~7) 101: 7x8, (LED 0~7) 11x: 保留
1	SELLED	LCD/LED功能选择:

0: LCD
1: LED。

0 LCDON LCD控制器使能:
0: 禁止LCD/LED控制器
1: 使能LCD/LED控制器

19.6.2 LCD 显示存储器 (LCD_RAM)

偏移地址: 0x10~0x4f

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SD[31:16]															
rW	rW	rW	rW	rW	rW	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
SD[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:0	SD[31:0]	LCD显示寄存器: 每个bit位对应一个LCD屏的像素点 0: 像素未激活 (不显示) 1: 像素激活 (显示) 注意: LCD_RAM与COM/SEG端子对应关系参见表21-3

20 高级控制定时器 (TIM1)

20.1 TIM1 简介

高级控制定时器 TIM1 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

高级控制定时器 TIM1 和通用定时器 TIM2/3/4 彼此完全独立，不共享任何资源。

20.2 TIM1 主要特性

本模块主要功能特性如下：

- 16 位递增、递减、递增/递减自动重载计数器；
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间；
- 多达 6 个独立通道，可用于：
 - 输入捕获（但通道 5 和通道 6 除外）；
 - 输出比较；
 - PWM 生成（边沿和中心对齐模式）；
 - 单脉冲模式输出
- 单脉冲模式输出；
- 可再触发单脉冲模式；
- 带可编程死区的互补输出；
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路；
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器；
- 2 个断路输入，用于将定时器的输出信号置于用户可选的安全配置中；
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - 输入捕获

- 输出比较

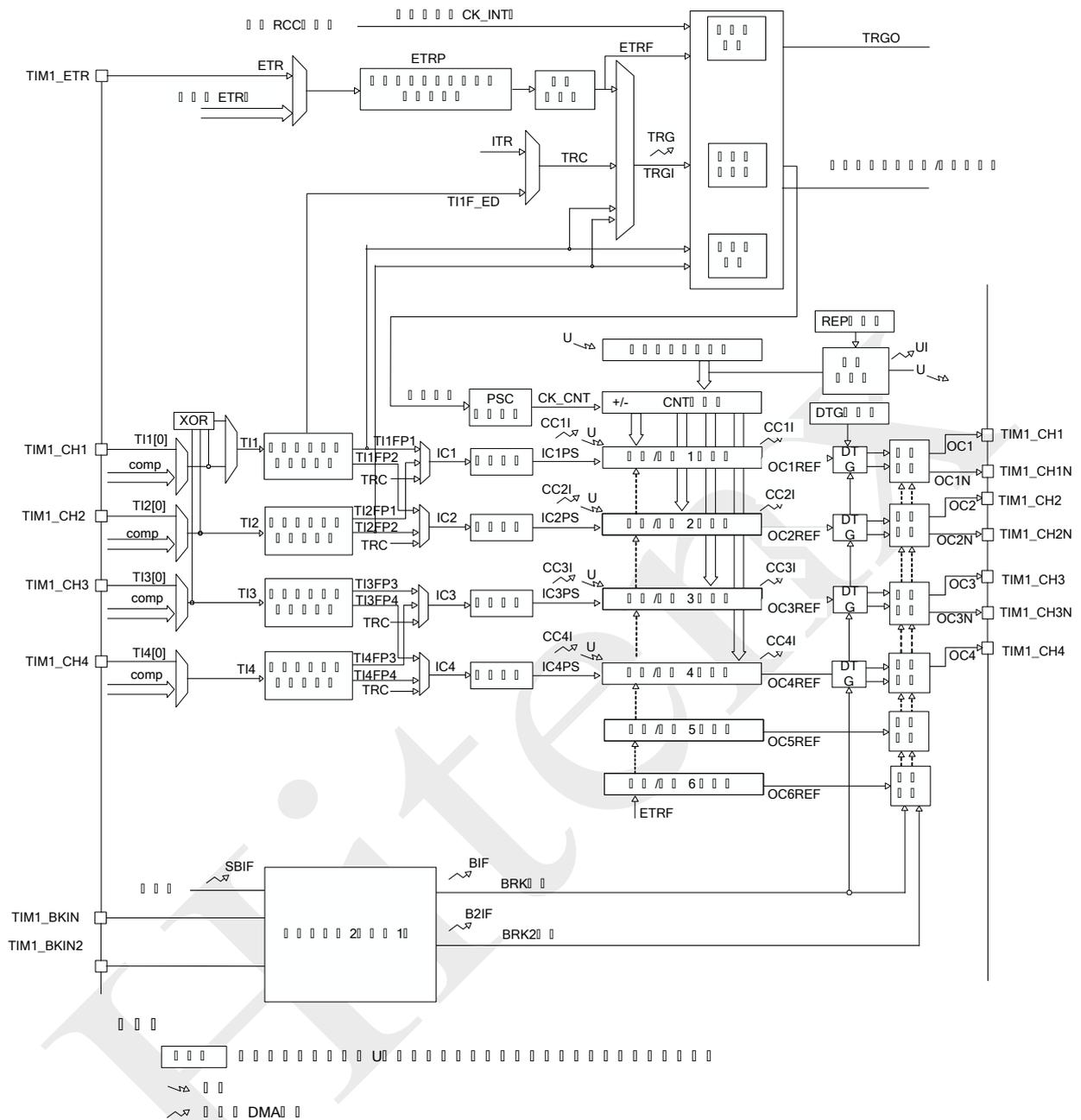
- 支持增量（正交）编码器和霍尔传感器电路；
- 触发输入用作外部时钟或逐周期电流管理；
- DMA burst 传输功能。

与 CPU 接口：

- 挂接在 APB2 总线上，只支持 32 位访问
- 访问模块内部未定义寄存器空间时，读数据为 0，写数据无效

时钟与复位：

- 双时钟域
- 异步复位，低电平有效

图 20-1 高级控制定时器框图


2. 断路和断路 2 的详细描述, 可参见图: 断路和断路 2 电路概述。

20.3 TIM1 功能说明

20.3.1 时基单元

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

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

时基单元包括：

- 计数器寄存器 (TIM1_CNT)
- 预分频器寄存器 (TIM1_PSC)
- 自动重载寄存器 (TIM1_ARR)
- 重复计数器寄存器 (TIM1_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 *TIM1 控制寄存器 (TIM1_CR1)* 中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIM1_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

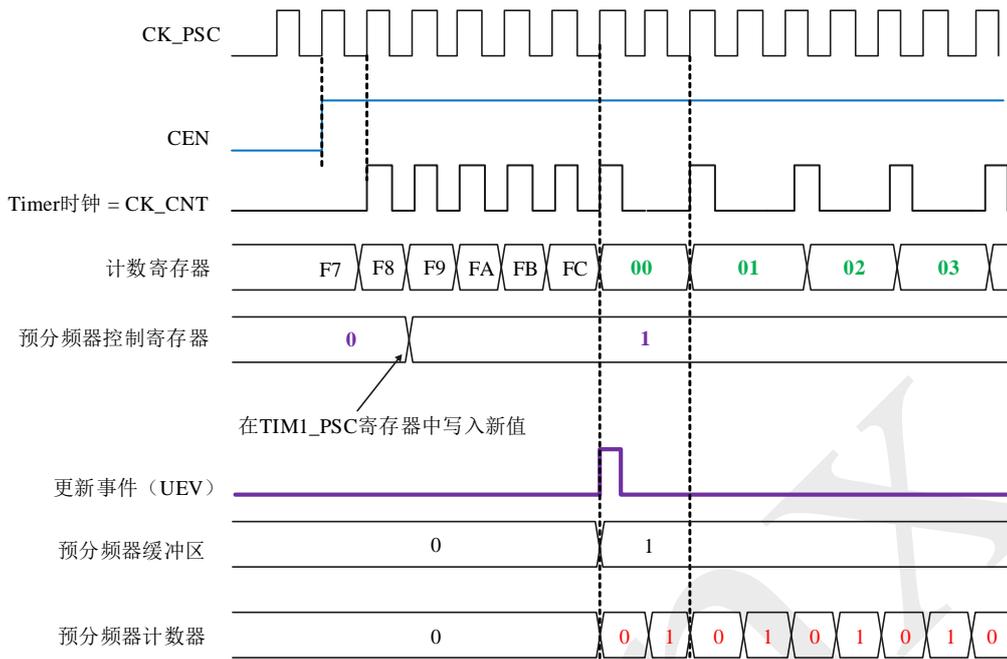
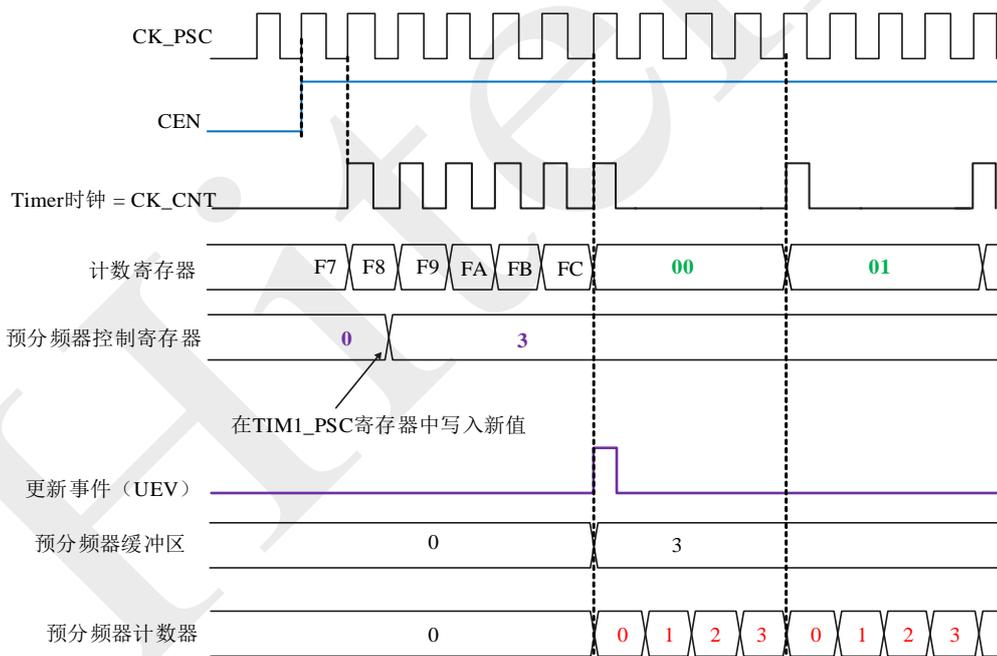
计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM1_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意： 计数器将在 TIM1_CR1 寄存器的 CEN 位置 1 时刻的 1 个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位 *TIM1 预分频器寄存器 (TIM1_PSC)* 所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

以下各图以一些示例说明在预分频比实时变化时计数器的行为：

图 20-2 预分频器分频由 1 变为 2 时的计数器时序图

图 20-3 预分频器分频由 1 变为 4 时的计数器时序图


20.3.2 计数器模式

递增计数模式

在递增计数模式下,计数器从 0 计数到自动重载值(TIM1_ARR 寄存器的内容),然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器,则当递增计数的重复次数达到重复计数器寄存器中设定的次数加一次(TIM1_RCR+1)后,将生成更新事件(UEV)。否则,将在每次计数器上溢时产生更新事件。

将 *TIM1 时间发生寄存器 (TIM1_EGR)* 的 UG 位置 1 (通过软件或使用从模式控制器) 时, 也将产生更新事件。

通过软件将 *TIM1 控制寄存器 (TIM1_CR1)* 中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过, 计数器和预分频器计数器都会重新从 0 开始计数 (而预分频比保持不变)。

此外, 如果 TIM1_CR1 寄存器中的 URS 位 (更新请求选择) 已置 1, 则将 UG 位置 1 会生成更新事件 UEV, 但不会将 UIF 标志置 1 (因此, 不会发送任何中断或 DMA 请求)。这样一来, 如果在发生捕获事件时将计数器清 0, 将不会同时产生更新中断和捕获中断。

发生更新事件时, 将更新所有寄存器且将更新标志 (*TIM1 状态寄存器 (TIM1_SR)* 中的 UIF 位) 置 1 (取决于 URS 位):

- 重复计数器中将重新装载 TIM1_RCR 寄存器的内容。
- 使用预装载值 (TIM1_ARR) 更新自动重载影子寄存器。
- 预分频器的缓冲区中将重新装载预装载值 (TIM1_PSC 寄存器的内容)。

以下各图以一些示例说明当 TIM1_ARR=0x36 时不同时钟频率下计数器的行为。

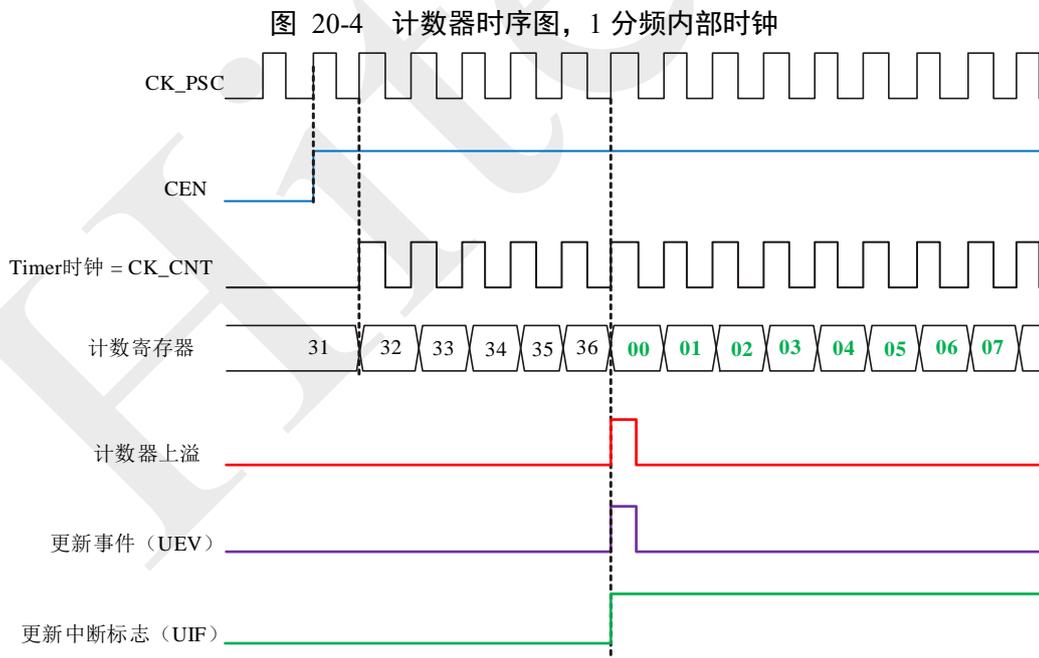


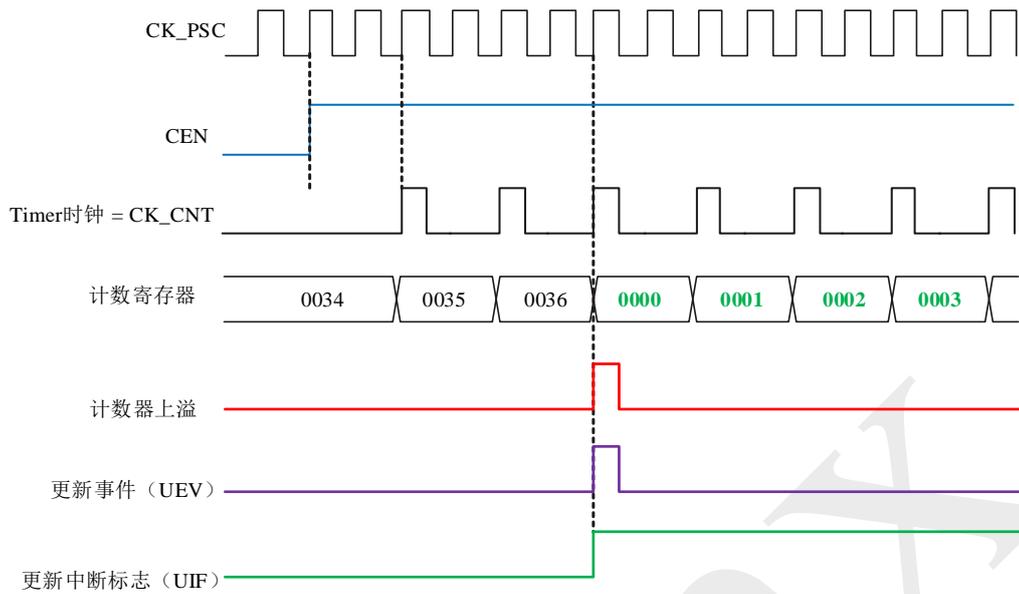
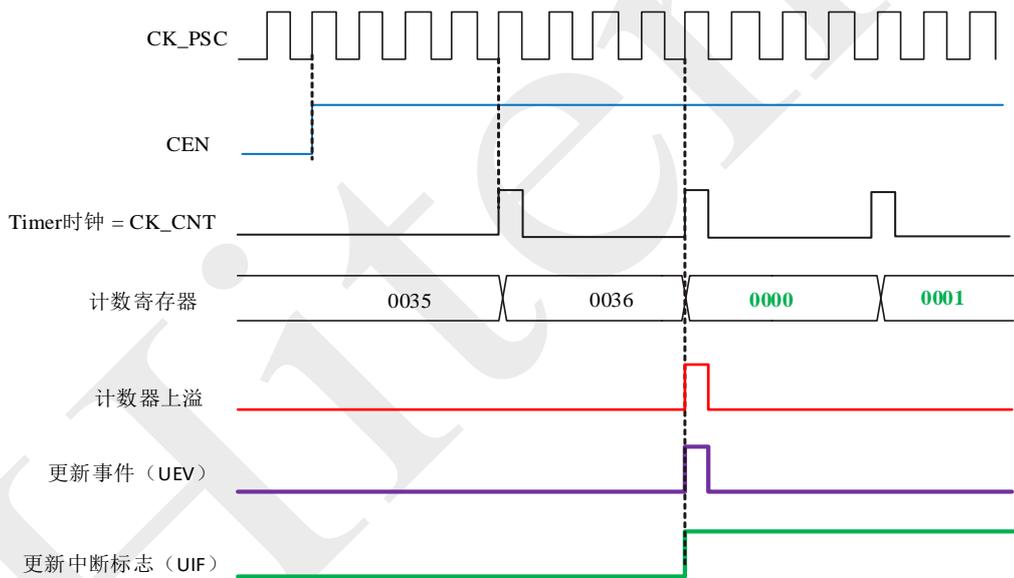
图 20-5 计数器时序图, 2 分频内部时钟

图 20-6 计数器时序图, 4 分频内部时钟


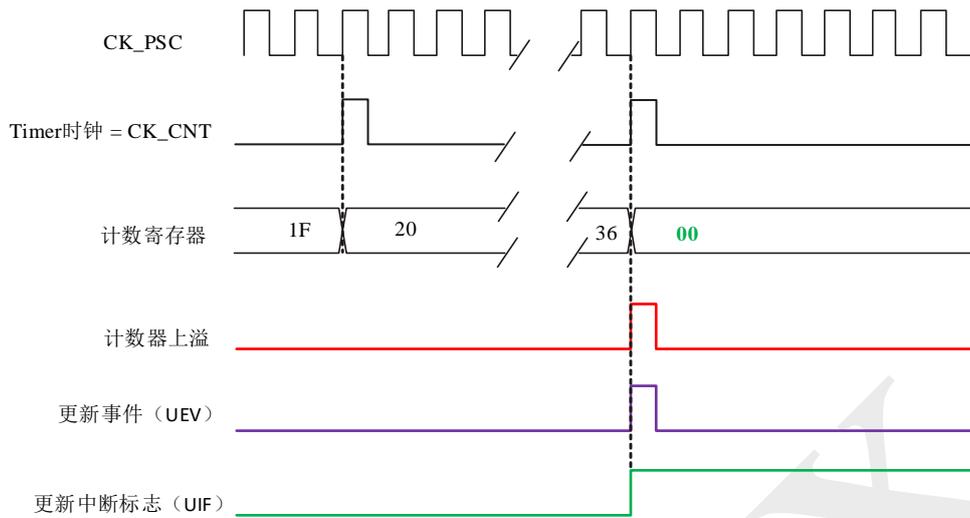
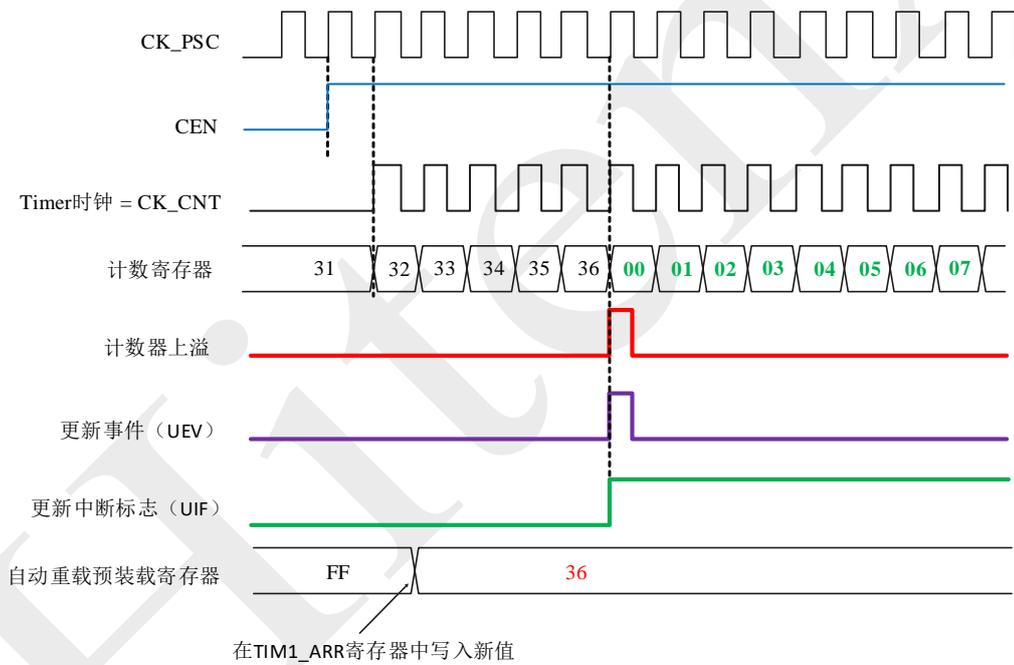
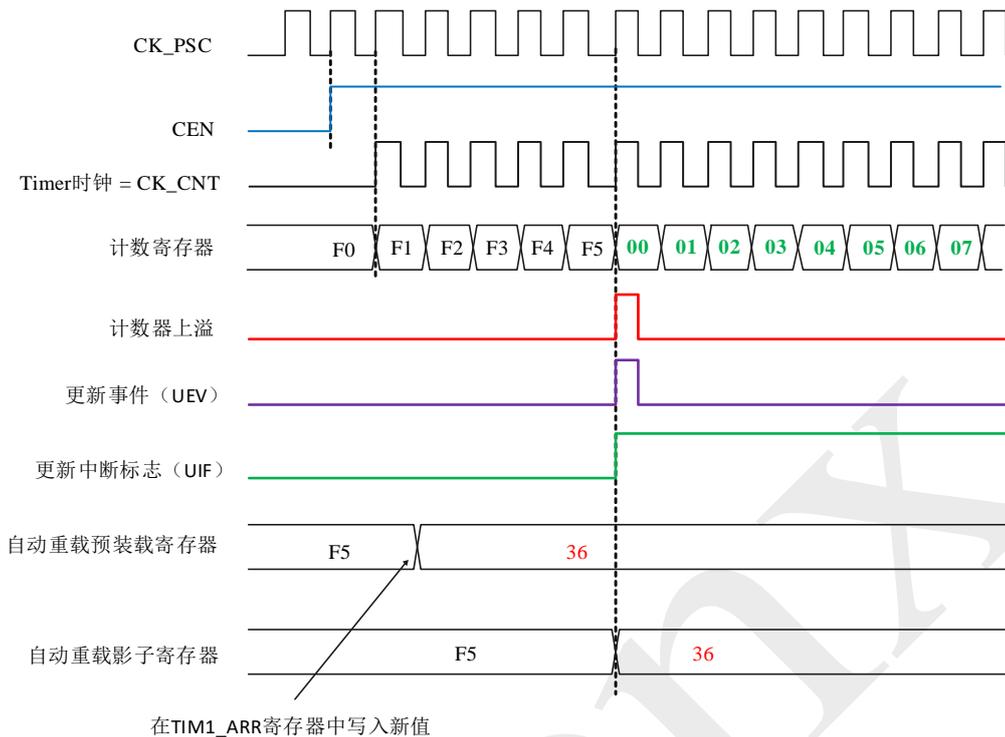
图 20-7 计数器时序图, N 分频内部时钟

图 20-8 计数器时序图, ARPE=0 时更新事件 (TIM1_ARR 未预装载)


图 20-9 计数器时序图, ARPE=1 时更新事件 (TIM1_ARR 已预装载)


递减计数模式

在递减计数模式下, 计数器从自动重载值 (TIM1_ARR 寄存器的内容) 开始递减计数到 0, 然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器, 则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIM1_RCR+1) 后, 将产生更新事件 (UEV)。否则, 将在每次计数器下溢时产生更新事件。

将 *TIM1 时间发生寄存器 (TIM1_EGR)* 的 UG 位置 1 (通过软件或使用从模式控制器) 时, 也将产生更新事件。

通过软件将 *TIM1 控制寄存器 (TIM1_CR1)* 中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载

寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过, 计数器会重新从当前自动重载值开始计数, 而预分频器计数器则重新从 0 开始计数 (但预分频比保持不变)。

此外, 如果 TIM1_CR1 寄存器中的 URS 位 (更新请求选择) 已置 1, 则将 UG 位置 1 会生成更新事件 UEV, 但不会将 UIF 标志置 1 (因此, 不会发送任何中断或 DMA 请求)。这样一来, 如果在发生捕获事件时将计数器清 0, 将不会同时产生更新中断和捕获中断。

发生更新事件时, 将更新所有寄存器且将更新标志 (*TIM1 状态寄存器 (TIM1_SR)* 中的 UIF 位) 置 1 (取决于 URS 位):

- 重复计数器中将重新装载 *TIM1 重复计数器寄存器 (TIM1_RCR)* 的内容；
- 预分频器的缓冲区中将重新装载预装载值 (TIM1_PSC 寄存器的内容)；
- 自动重载有效寄存器将以预装载值 (TIM1_ARR 寄存器的内容) 进行更新。

注意: *ARR 寄存器更新在计数器重载之前被更新, 因此下一个周期就是预期的值。*

以下各图以一些示例说明当 TIM1_ARR=0x36 时不同时钟频率下计数器的行为。

图 20-10 计数器时序图, 1 分频内部时钟

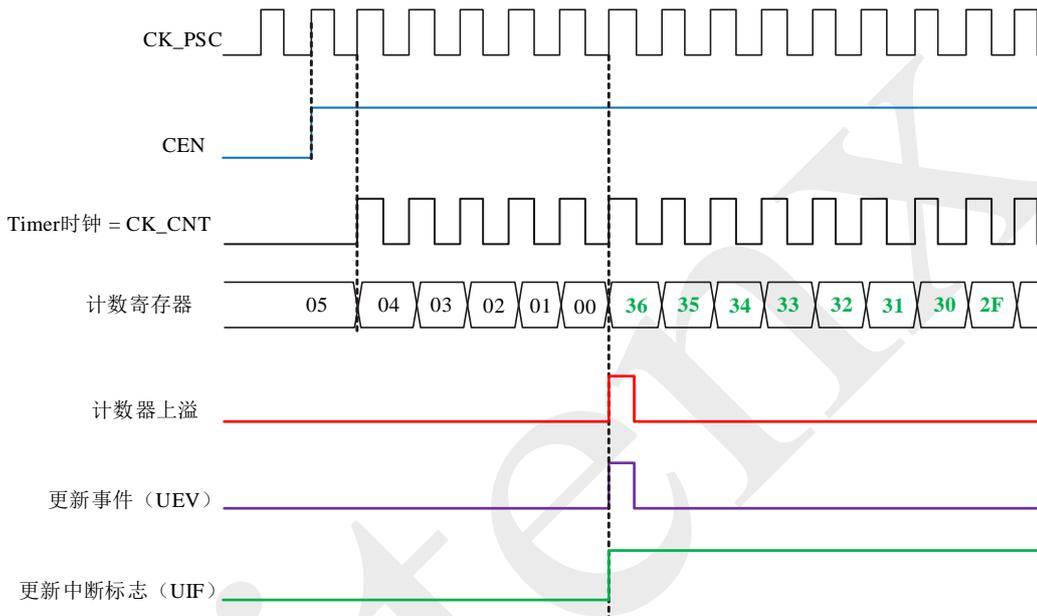


图 20-11 计数器时序图, 2 分频内部时钟

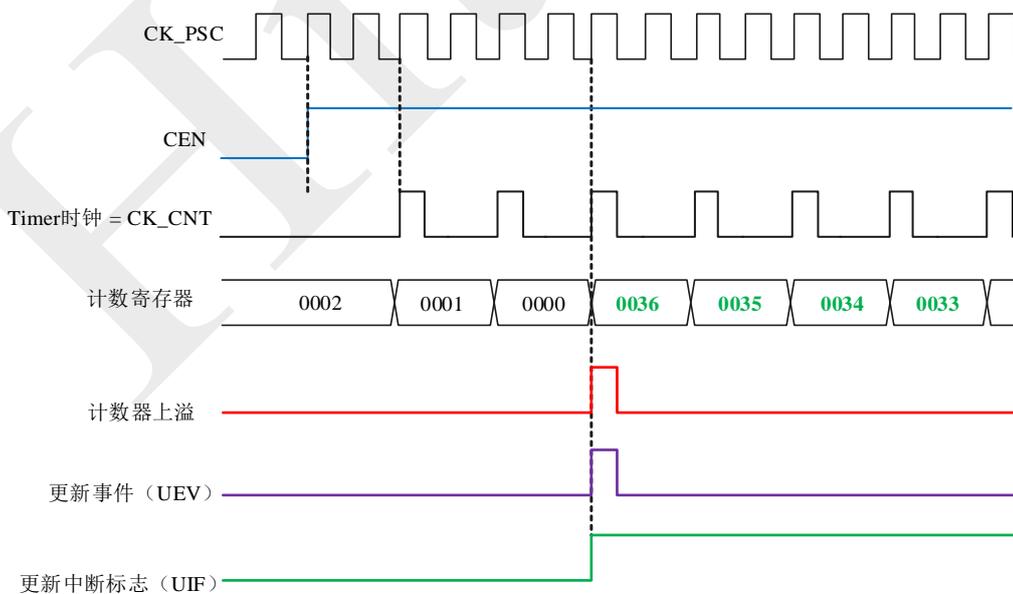


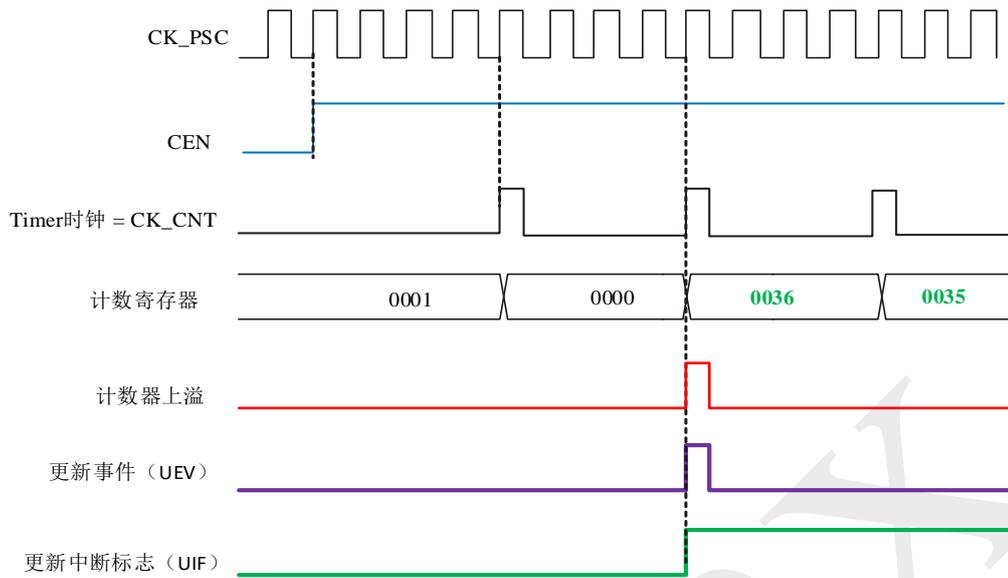
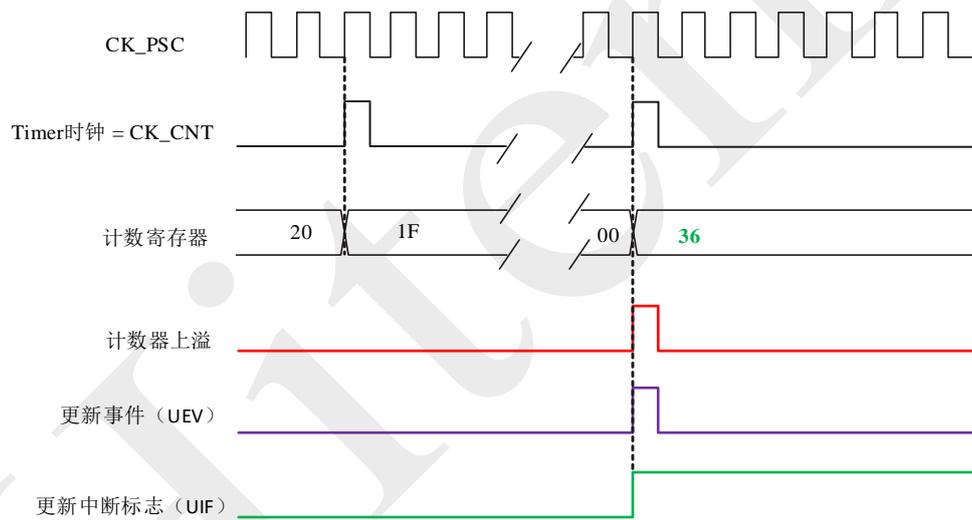
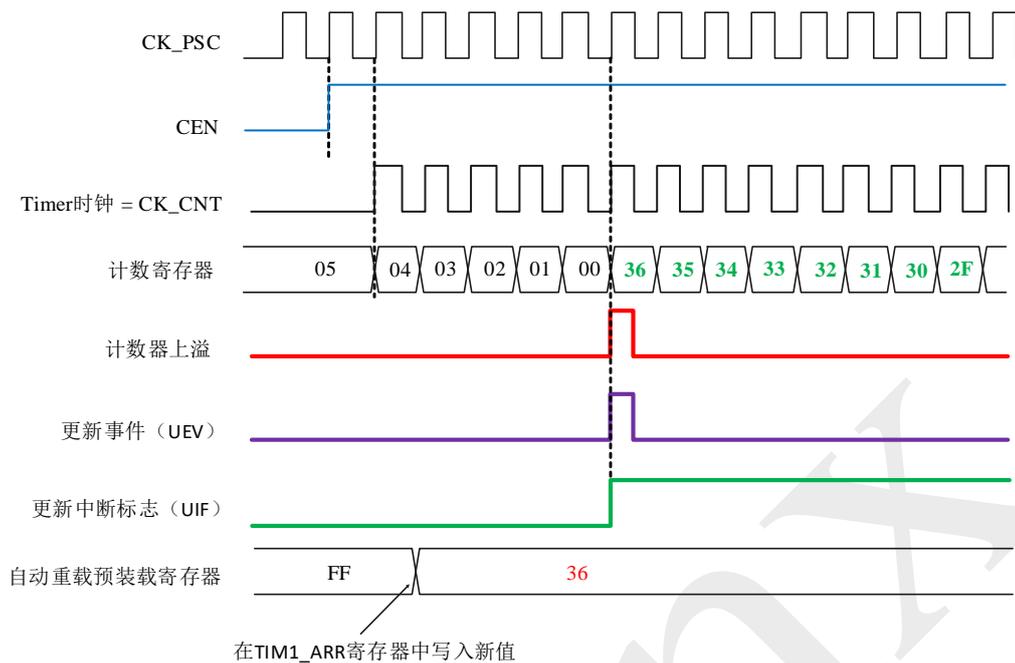
图 20-12 计数器时序图, 4 分频内部时钟

图 20-13 计数器时序图, N 分频内部时钟


图 20-14 计数器时序图，未使用重复计数器时更新事件


中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIM1_ARR 寄存器的内容）-1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 *TIM1 控制寄存器 (TIM1_CR1)* 中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式置 1，即：计数器递减计数（中心对齐模式 1，CMS=“01”）、计数器递增计数（中心对齐模式 2，CMS=“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS=“11”）。

此模式下，无法写入方向位（TIM1_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 *TIM1 事件产生寄存器 (TIM1_EGR)* 中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。通过软件将 TIM1_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM1_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清 0，将不会同时产生更新中断和捕获中断。

发生更新事件时,将更新所有寄存器且将更新标志(*TIM1 状态寄存器(TIM1_SR)*中的 UIF 位)置 1 (取决于 URS 位):

- 重复计算器中将重新装载 *TIM1 重复计算器寄存器 (TIM1_RCR)*;
- 预分频器的缓冲区中将重新装载预装载值 (*TIM1 预分频器寄存器 (TIM1_PSC)*)。
- 自动重载活动寄存器将以预装载值 (*TIM1 自动重载值寄存器 (TIM1_ARR)*) 进行更新。注意, 如果更新操作是由计数器上溢触发的, 则 ARR 寄存器在计数器重载之前更新, 因此, 下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 20-15 计数器时序图, 1 分频内部时钟, TIM1_ARR=0x06

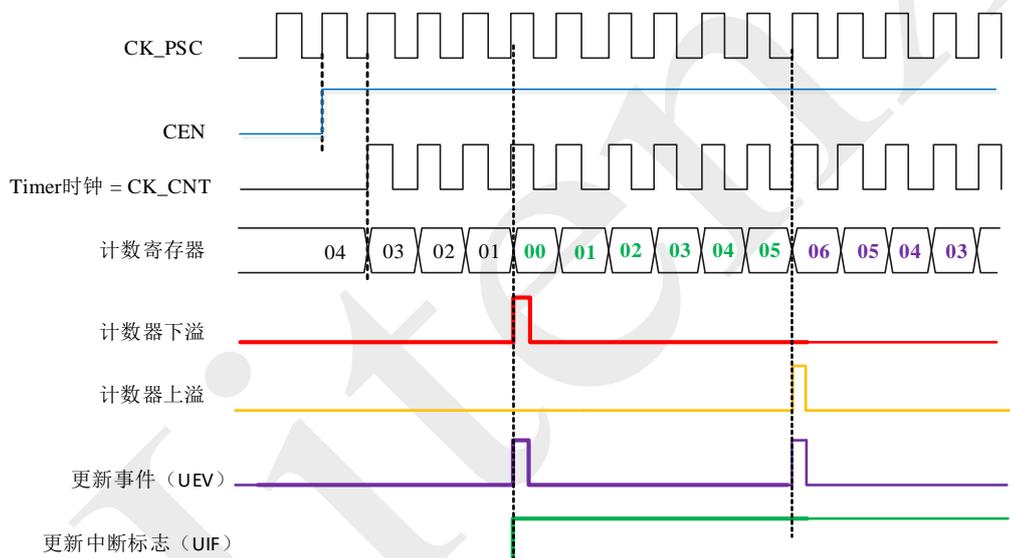


图 20-16 计数器时序图, 2 分频内部时钟

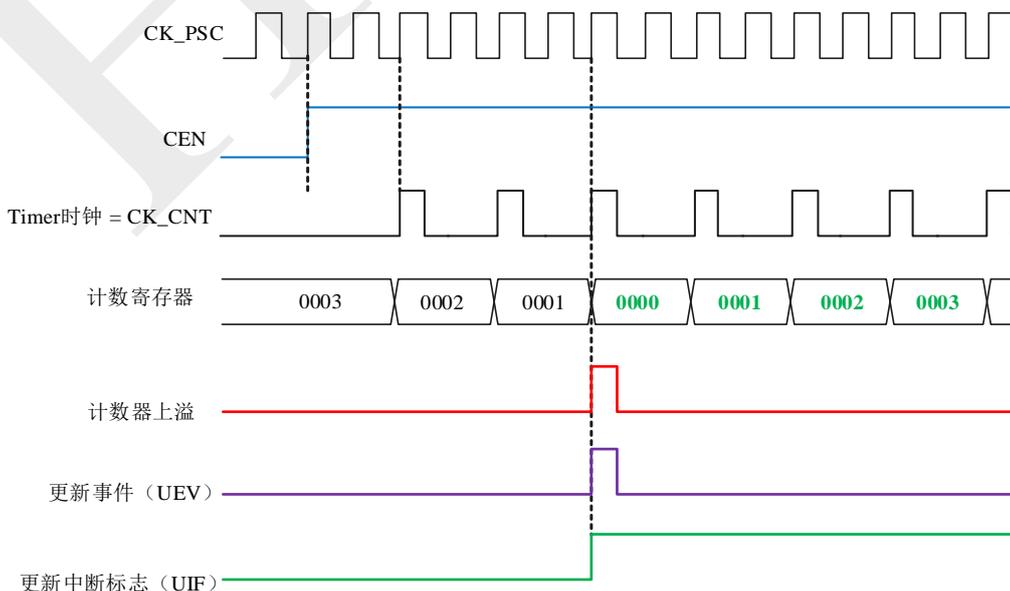
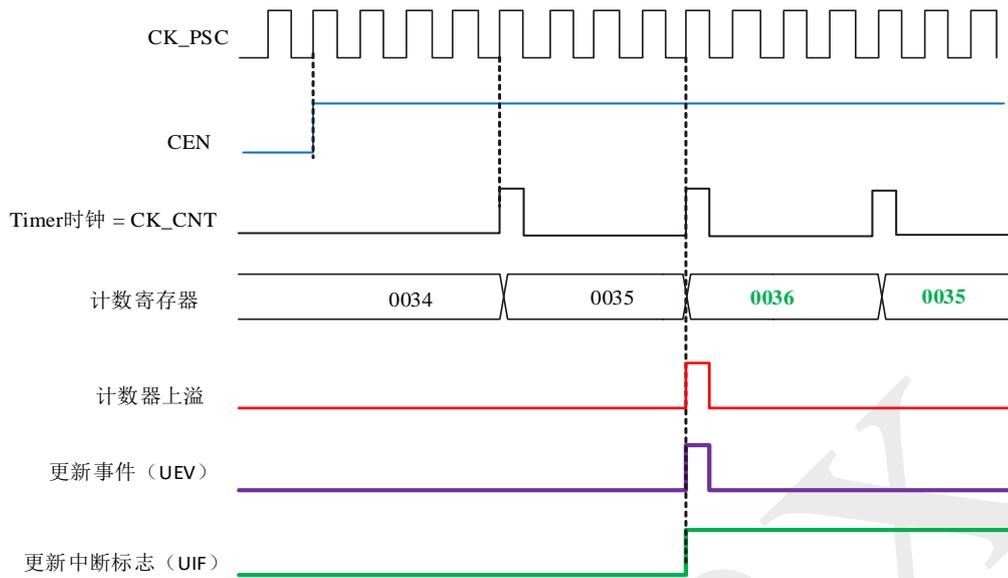


图 20-17 计数器时序图, 4 分频内部时钟, TIM1_ARR=0x36


3. 此处使用的中心对其模式 2 或模式 3 与上溢 UIF 一起更新。

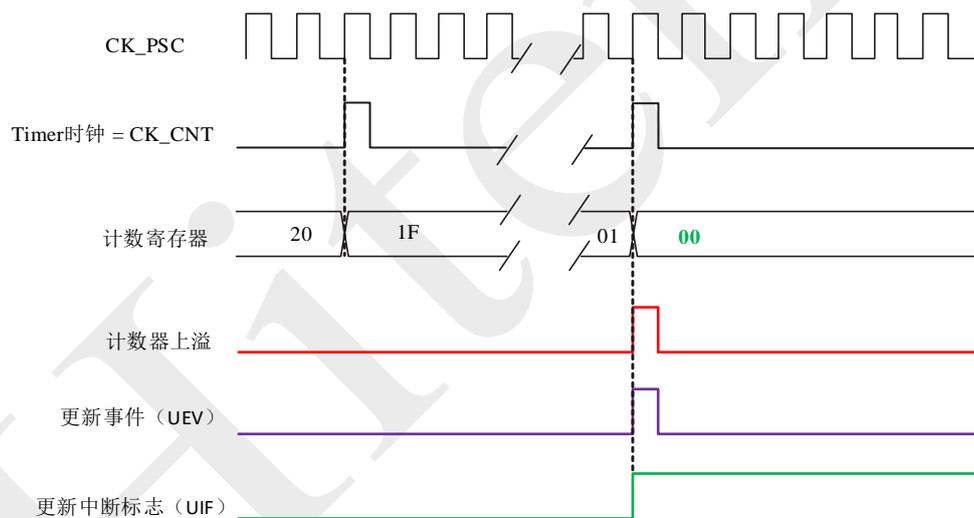
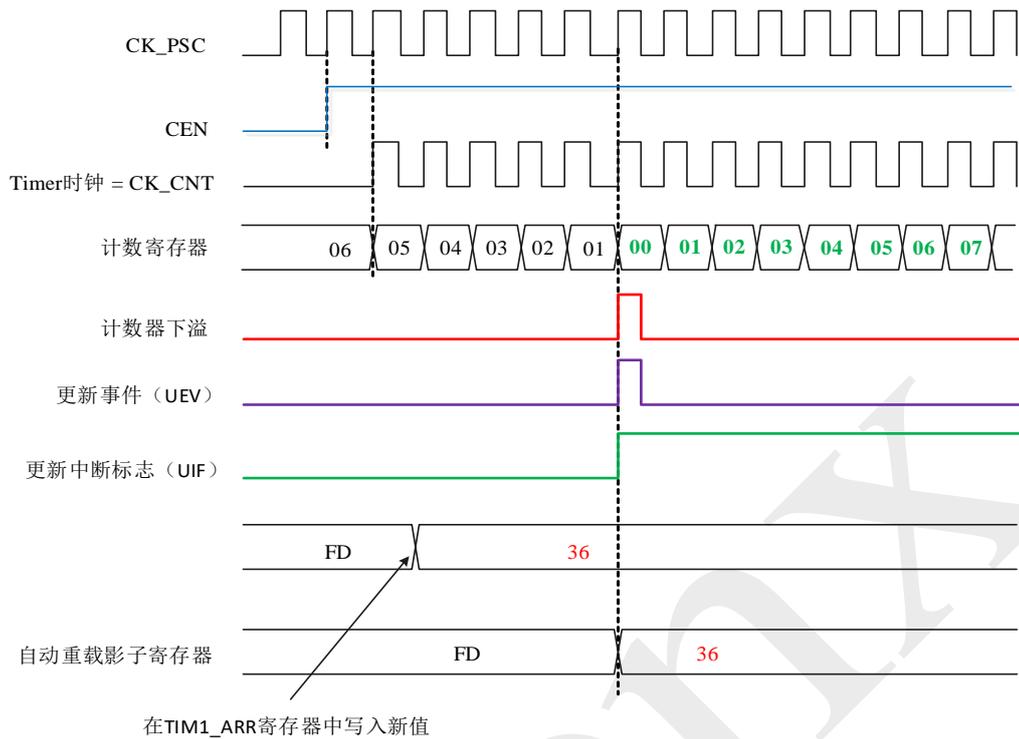
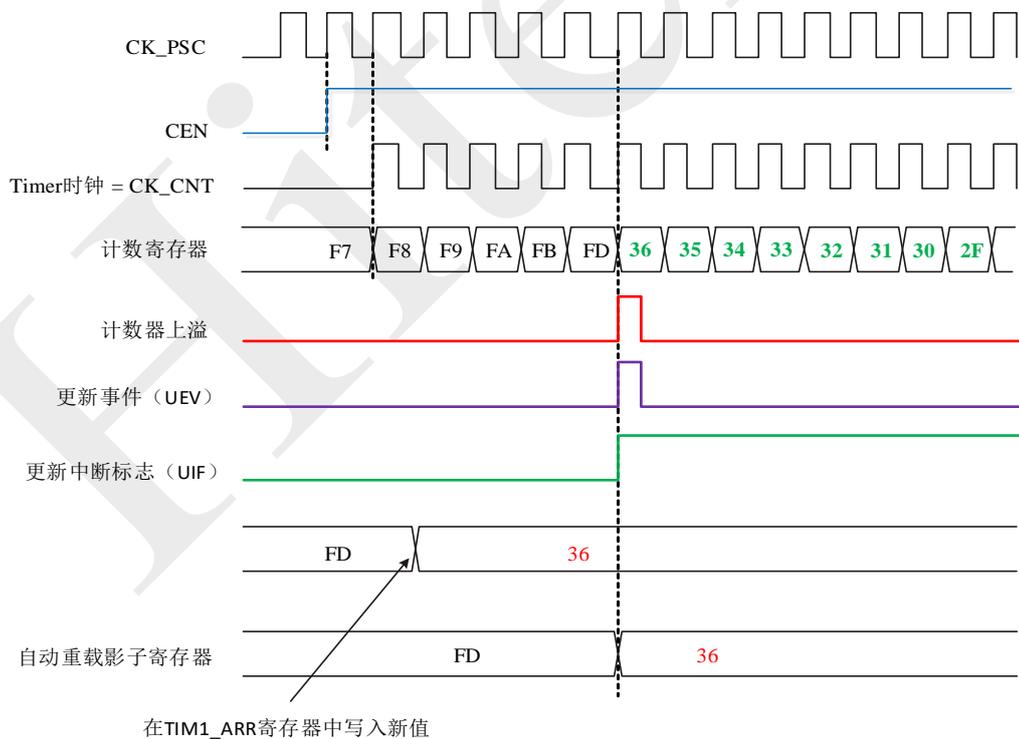
图 20-18 计数器时序图, N 分频内部时钟


图 20-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

图 20-20 计数器时序图，ARPE=1 时的更新事件（计数器上溢）


20.3.3 重复计数器

时基单元介绍如何因计数器上溢/下溢而生成更新事件（UEV）。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 $N+1$ 个计数器上溢或下溢（其中， N 是 *TIM1 重复计数器寄存器 (TIM1_RCR)* 的值），数据就将从预装载寄存器转移到影子寄存器（*TIM1 自动重载寄存器 (TIM1_ARR)*、*TIM1 预分频器寄存器 (TIM1_PSC)* 以及比较模式下的 *TIM1_CCRx* 捕获/比较寄存器）。

重复计数器在下列情况下递减：

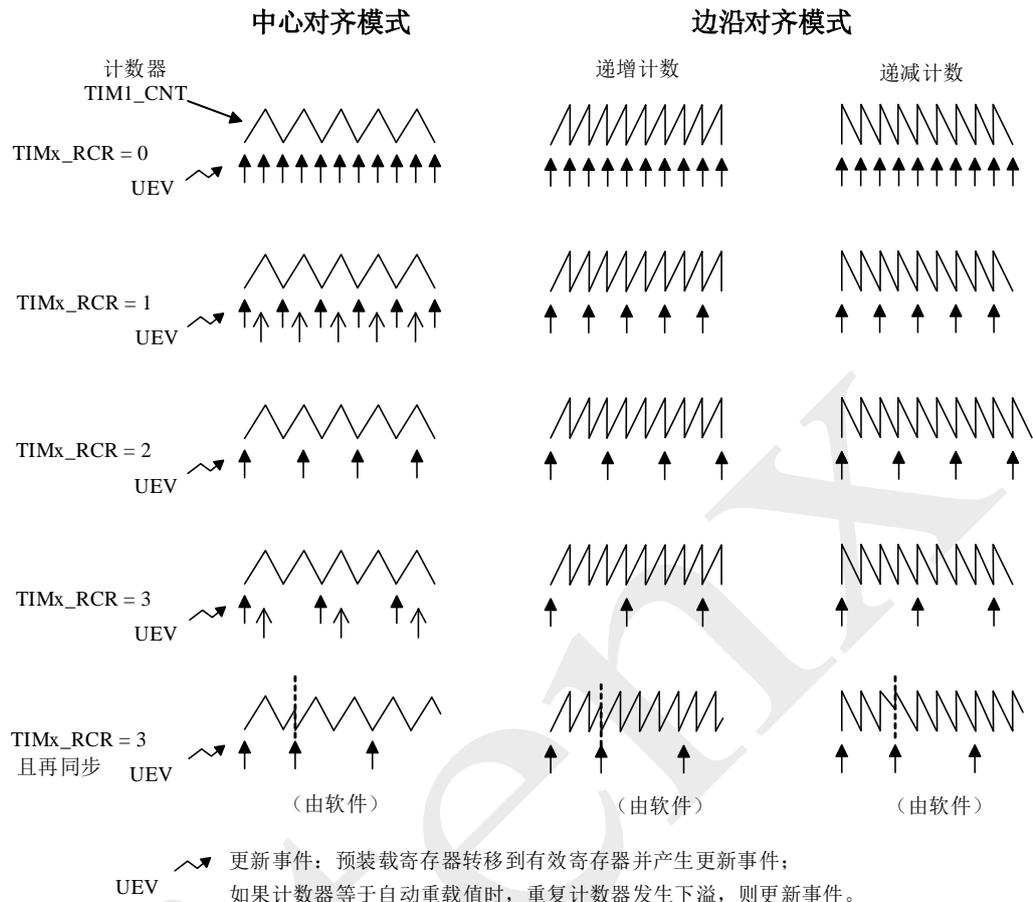
- 递增计数模式下的每个计数器上溢；
- 递减计数模式下的每个计数器下溢；
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2 \times T_{ck}$ 。

重复计数器是自动重载类型；其重复率为 *TIM1_RCR* 寄存器所定义的值。当更新事件由软件（通过将 *TIM1 时间发生寄存器 (TIM1_EGR)* 的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 *TIM1_RCR* 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 $RCR=3$ ，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 20-21 不同模式和 TIM1_RCR 寄存器设置下的更新频率示例



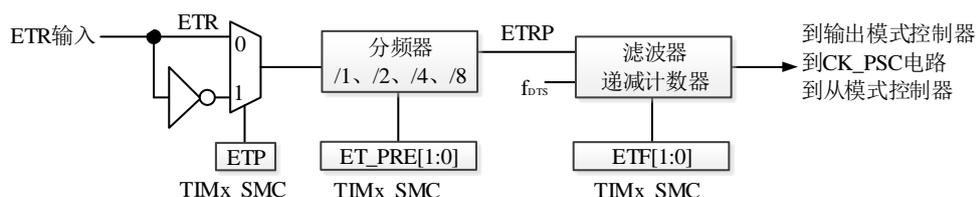
20.3.4 外部触发输入

定时器具有一个外部触发输入 ETR，它可用作：

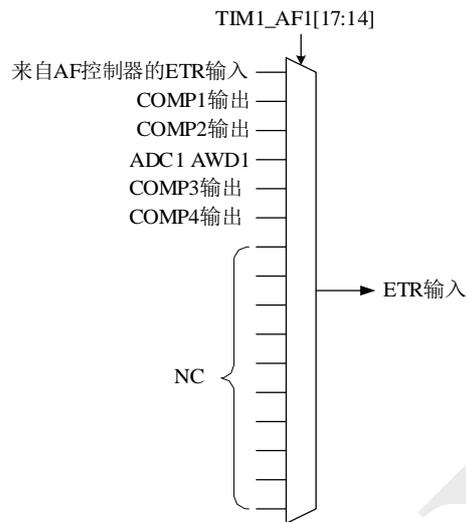
- 外部时钟（外部时钟模式 2，请参见[时钟选择](#)）
- 用于从模式的触发信号（请参见[定时器同步](#)）
- 用于逐周期电流调节的 PWM 复位输入

下面[图：外部触发输入模块](#)介绍了 ETR 输入的调节过程。输入极性通过 [TIM1 从模式控制寄存器 \(TIM1_SMCR\)](#) 中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位域编程的分频比进行预分频，然后通过 ETF[3:0] 位域进行数字滤波。

图 20-22 外部触发输入模块



ETR 输入来自多个源：输入引脚（默认配置）、比较器输出和模拟看门狗。可通过 [TIM1 轮换功能寄存器 1 \(TIM1_AF1\)](#) 中的 ETRSEL[3:0] 位域进行选择。

图 20-23 TIM1 ETR 输入电路


20.3.5 时钟选择

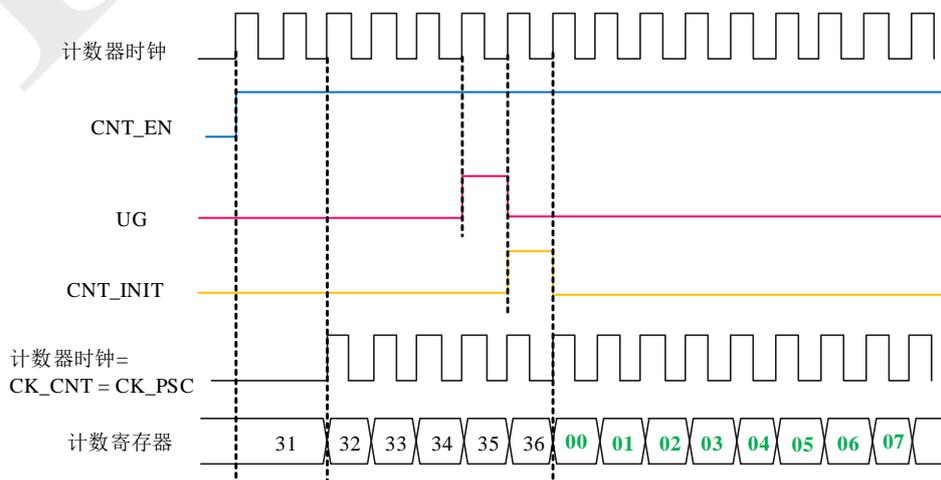
计数器时钟可由下列时钟源提供：

- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=“000”), 则 CEN 位、DIR 位 (*TIM1 控制寄存器 (TIM1_CR1)*) 和 UG 位 (*TIM1 事件产生寄存器 (TIM1_EGR)*) 为实际控制位, 并且只能通过软件进行更改 (UG 除外, 仍保持自动清 0)。当对 CEN 位写入 1 时, 预分频器的时钟就由内部时钟 CK_INT 提供。

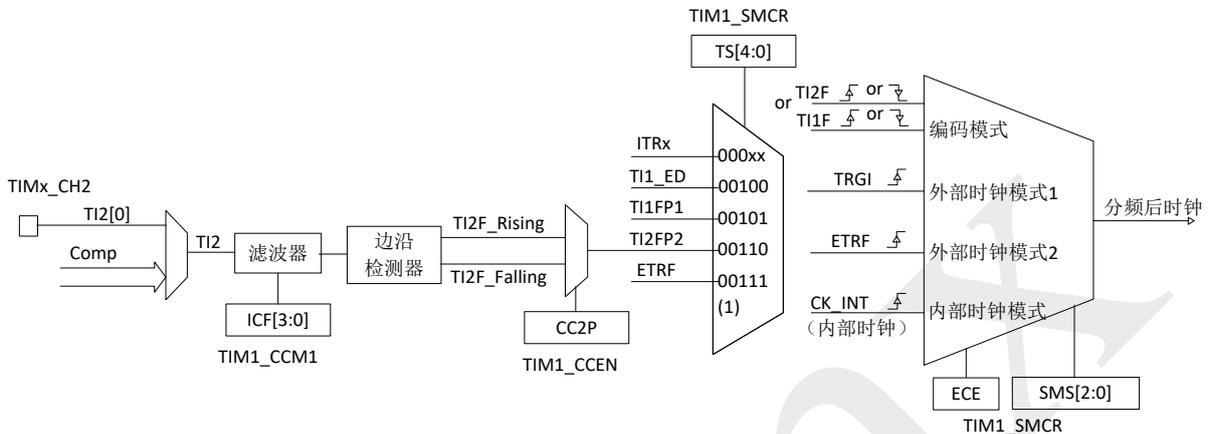
下图显示了正常模式下控制电路与递增计数器的行为 (没有预分频的情况下)。

图 20-24 正常模式下的控制电路, 1 分频内部时钟


外部时钟源模式 1

当 *TIM1* 从模式控制寄存器 (*TIM1_SMCR*) 中的 SMS=“111”时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 20-25 TI2 外部时钟连接示例



- 保留“01000”到“11111”的代码。

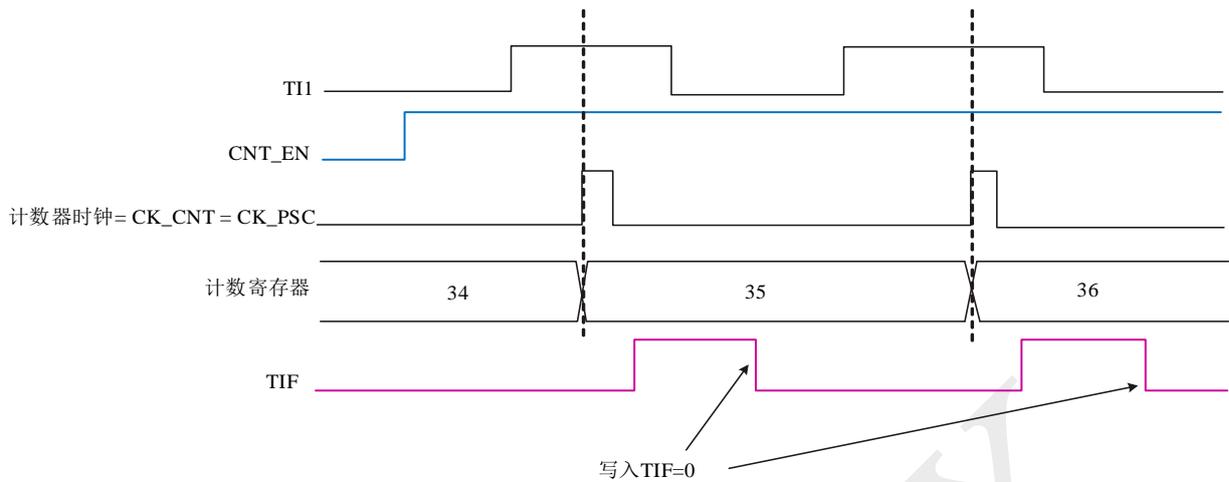
例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

- 通过 *TIM1* 定时器输入选择寄存器 (*TIM1_TISEL*) 中的 TI2SEL[3:0] 位域选择适当的 TI2x 源（内部或外部）；
- 通过在 *TIM1* 捕获/比较寄存器 (*TIM1_CCMR1*) 中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
- 通过在 *TIM1_CCMR1* 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=“0000”）。
- 通过在 *TIM1* 捕获/比较使能寄存器 (*TIM1_CCER*) 中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
- 通过在 *TIM1* 从模式控制寄存器 (*TIM1_SMCR*) 中写入 SMS=“111”，使定时器在外部时钟模式 1 下工作。
- 通过在 *TIM1_SMCR* 寄存器中写入 TS=“00110”来选择 TI2 作为触发输入源。
- 通过在 *TIM1* 控制寄存器 (*TIM1_CR1*) 中写入 CEN=1 来使能计数器。

注意： 由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

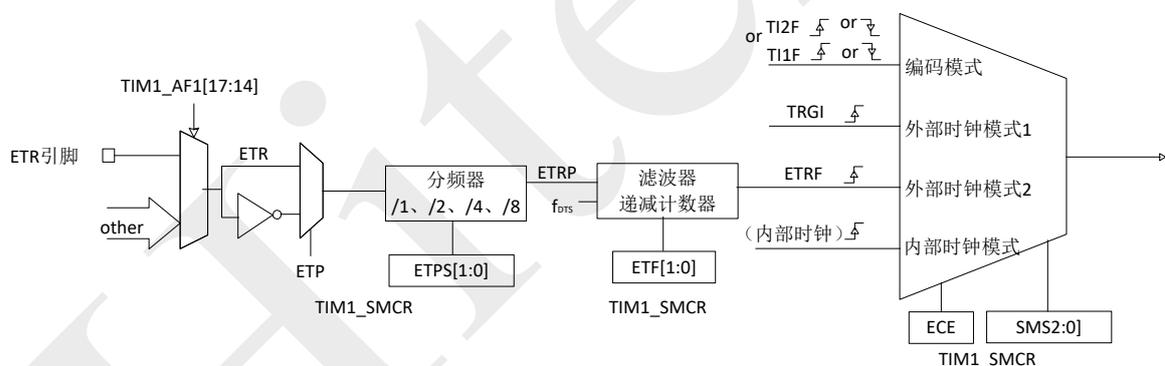
TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 20-26 外部时钟模式 1 下的控制电路


外部时钟源模式 2

通过在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 ECE=1 可选择此模式。计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

下图简要介绍了外部触发输入模块。

图 20-27 外部触发输入模块


例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

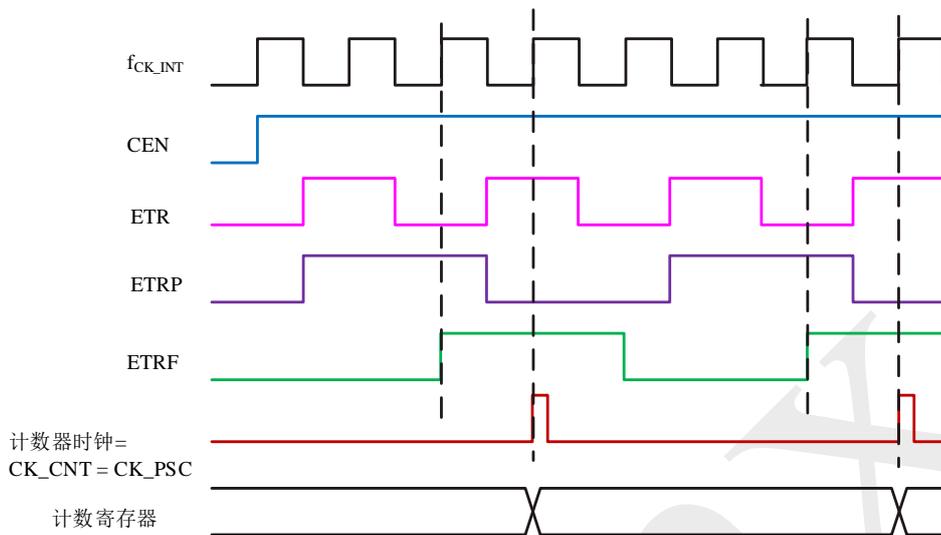
- 1) 由于此例中不需滤波器，因此在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 ETF[3:0]=“0000”。
- 2) 通过在 TIM1_SMCR 寄存器中写入 ETPS[1:0]=“01”来设置预分频器。
- 3) 通过在 TIM1_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
- 4) 通过在 TIM1_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
- 5) 通过在 *TIM1 控制寄存器 (TIM1_CR1)* 中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器可正确捕获的最大频率最多为 TIMxCLK 频率的 1/4。当 ETRP 信号更快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

设置。

图 20-28 外部时钟模式 2 下的控制电路



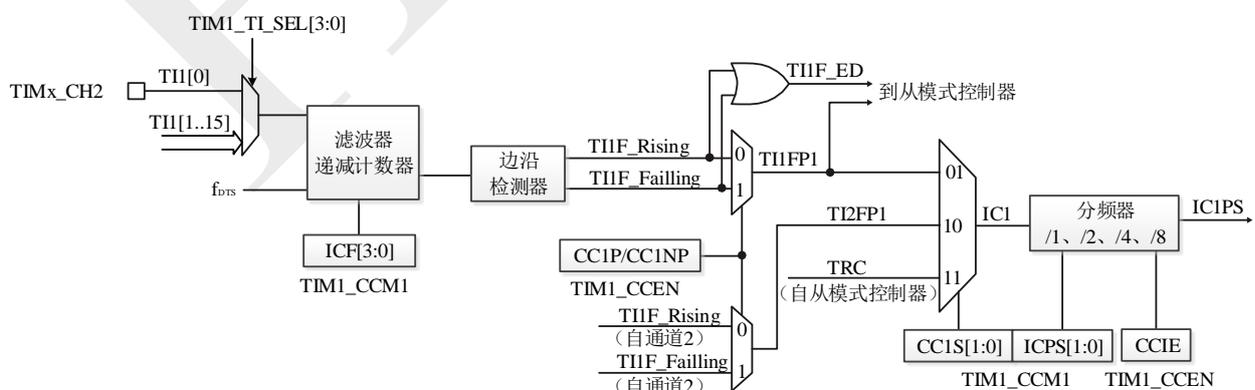
20.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出阶段（比较器和输出控制）构建而成。

图：捕获/比较通道（示例：通道 1 输入阶段）到图：捕获/比较通道的输出阶段（通道 5 和通道 6） 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 $TIxF$ 。然后，带有极性选择功能的边沿检测器生成一个信号 ($TIxFPx$)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 ($ICxPS$)，而后再进入捕获寄存器。

图 20-29 捕获/比较通道（示例：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准： $OCxRef$ （高电平有效）。链的末端决定最终输出信号的极性。

图 20-30 捕获/比较通道 1 主电路

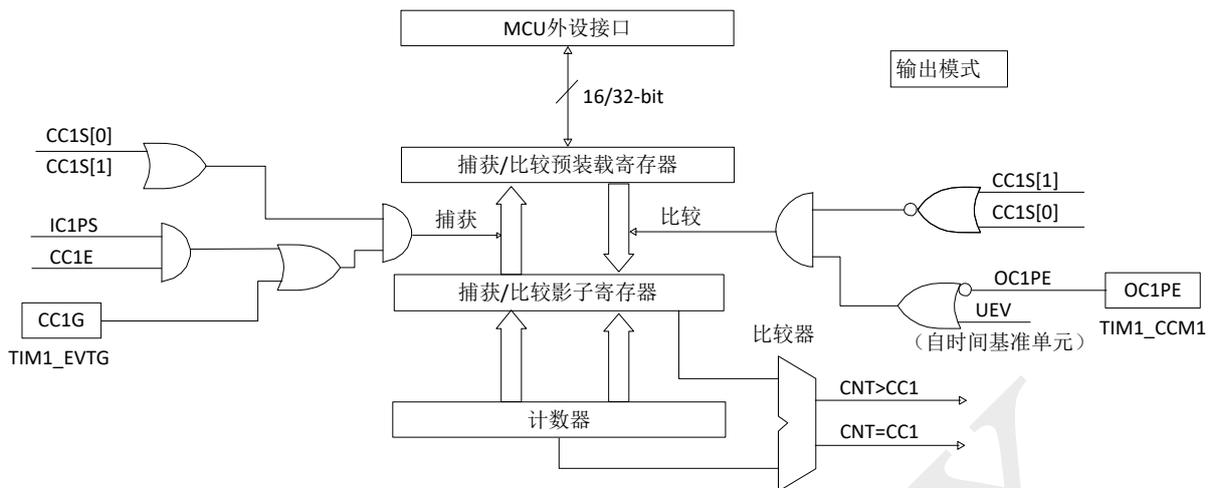
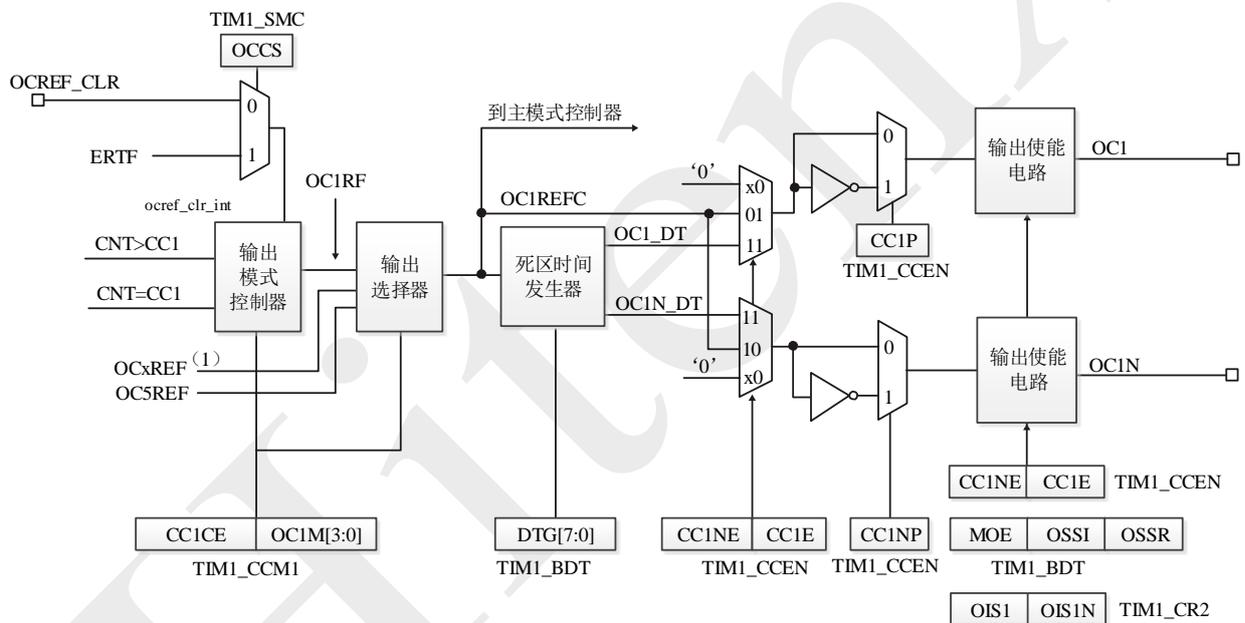
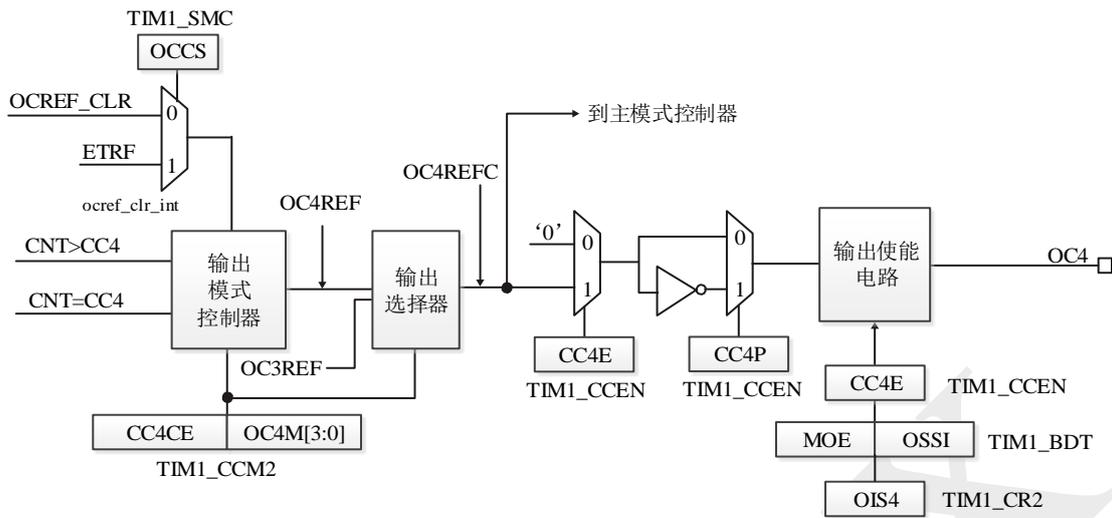
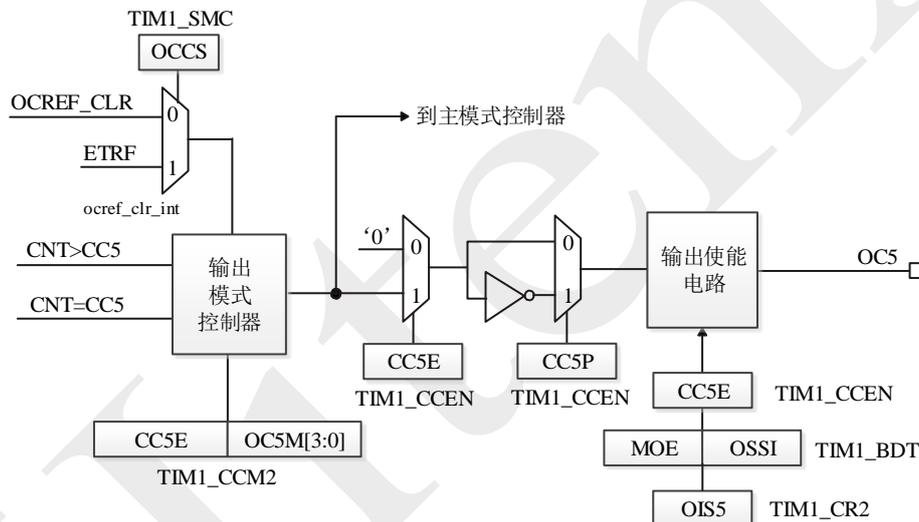


图 20-31 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3）



2. OCxREF，其中 x 为互补通道的序号

图 20-32 捕获/比较通道的输出阶段（通道 4）

图 20-33 捕获/比较通道的输出阶段（通道 5 和通道 6）


3. 不适用于外部。

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。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

20.3.7 输入捕获模式

在输入捕获模式下，当相应的 IC_x 信号检测到跳变沿后，将使用 TIM1 捕获/比较寄存器 (TIM1_CCR_x) 来锁存计数器的值。发生捕获事件时，会将相应的 CC_xIF 标志 (*TIM1 状态寄存器 (TIM1_SR)*) 置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CC_xIF 标志已处于高位，则会将重复捕获标志

CCxOF (TIM1_SR 寄存器) 置 1。可通过软件将 CCxIF 清 0，方法是：向 CCxIF 写入“0”，或读取存储在 TIM1_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清 0。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 *TIM1 捕获/比较寄存器 (TIM1_CCR1)* 中。具体操作步骤如下：

- 1) 通过定时器输入选择 *TIM1 定时器输入选择寄存器 (TIM1_TISEL)* 中的 TI1SEL[3:0] 位域选择适当的 TI1x 源（内部或外部）；
- 2) 选择有效输入：TIM1_CCR1 必须连接到 TI1 输入，因此向 *TIM1 捕获/比较寄存器 (TIM1_CCMR1)* 中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIM1_CCR1 寄存器将处于只读状态；
- 3) 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIM1_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIM1_CCMR1 寄存器中的 IC1F 位写入“0011”。
- 4) 通过在 *TIM1 捕获/比较使能寄存器 (TIM1_CCER)* 中将 CC1P 位和 CC1NP 位写入“0”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 5) 对输入预分频器进行编程。在本例中，希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIM1_CCMR1 寄存器中的 IC1PS 位写入“00”）。
- 6) 通过将 TIM1_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 7) 如果需要，可通过将 *TIM1 DMA/中断使能寄存器 (TIM1_DIER)* 中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIM1_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清 0，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注意： 通过软件将 *TIM1 事件产生寄存器 (TIM1_EGR)* 中的相应 CCxG 位置 1 可生成

IC 中断和/或 DMA 请求。

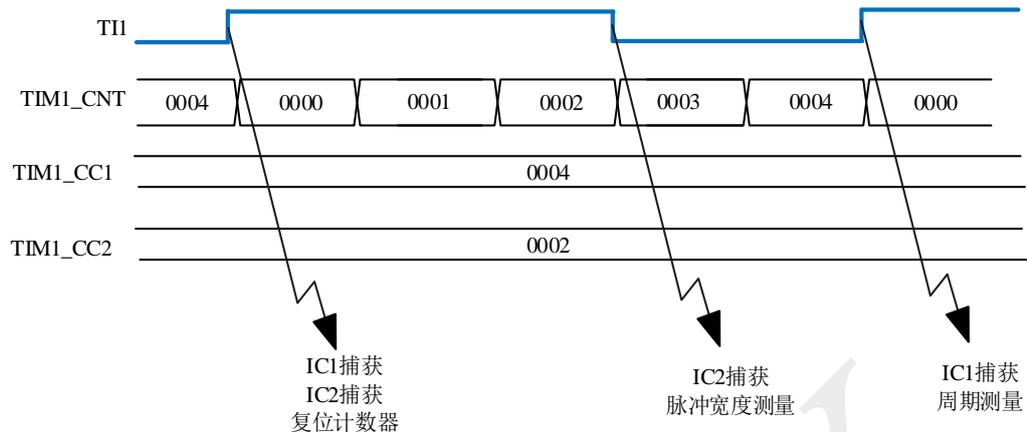
20.3.8 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 *TIM1 捕获/比较寄存器 1 (TIM1_CCR1)* 中）和占空比（位于 *TIM1 捕获/比较寄存器 2 (TIM1_CCR2)* 中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 1) 通过定时器输入选择中的 TI1SEL[3:0]位域选择适当的 TI1x 源（内部或外部）；
- 2) 选择 TIM1_CCR1 寄存器的有效输入：向 TIM1_CCMR1 寄存器中的 CC1S 位写入“01”（选择 TI1）。
- 3) 选择 TI1FP1 的有效极性（用于在 TIM1_CCR1 中捕获和计数器清 0）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 4) 选择 TIM1_CCR1 寄存器的有效输入：向 TIM1_CCMR1 寄存器中的 CC2S 写入“10”（选择 TI1）。
- 5) 选择 TI1FP2 的有效极性（用于在 TIM1_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
- 6) 选择有效触发输入：向 TIM1 从模式控制寄存器（TIM1_SMCR）中的 TS 位写入“101”（选择 TI1FP1）。
- 7) 将从模式控制器配置为复位模式：向 TIM1_SMCR 寄存器中的 SMS 位写入“0100”。
- 8) 使能捕获：向 TIM1 捕获/比较使能寄存器（TIM1_CCER）中的 CC1E 位和 CC2E 位写入“1”。

图 20-34 PWM 输入模式时序


20.3.9 强制输出模式

在输出模式（TIM1_CCMRx 寄存器中的 CCxS 位=“00”）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号（OCxREF/OCx）强制设置为有效电平，用户只需向相应 TIM1_CCMRx 寄存器中的 OCxM 位写入“0101”。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=>将 OCx 强制设置为高电平。

通过向 TIM1_CCMRx 寄存器中的 OCxM 位写入“0100”，可将 OCxREF 信号强制设置为低电平。

无论如何，TIM1_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

20.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一段时间。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在单片机内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIM1_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM1_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCxM=“0000”），也可设置为有效电平（OCxM=“0001”）、无效电平（OCxM=“0010”）或进行翻转（OCxM=“0011”）。
- 将中断状态寄存器中的标志置 1（TIM1_SR 寄存器中的 CCxIF 位）。

- 如果相应中断使能位（TIM1_DIER 寄存器中的 CCxIE 位）置 1，将生成中断。
- 如果相应使能位（TIM1_DIER 寄存器的 CCxDE 位，TIM1_CR2 寄存器的 CC_DMASEL 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

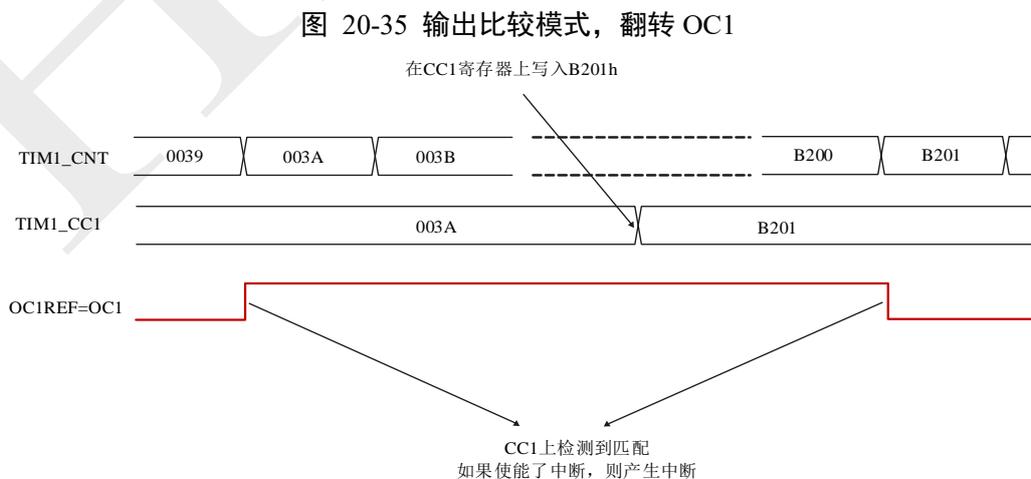
使用 TIM1_CCMRx 寄存器中的 OCxPE 位，可将 TIM1_CCRx 寄存器配置为带或不带预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

其操作步骤如下：

- 1) 选择计数器时钟（内部、外部、预分频器）。
- 2) 在 *TIM1 自动重载值寄存器 (TIM1_ARR)* 和 TIM1_CCRx 寄存器中写入所需数据。
- 3) 如果要生成中断请求，则需将 CCxIE 位置 1。
- 4) 选择输出模式。例如：
 - 当 CNT 与 CCx 匹配时，写入 OCxM=“0011”以翻转 OCx 输出引脚；
 - 写入 OCxPE=0 以禁止预装载寄存器；
 - 写入 CCxP=0 以选择高电平有效极性；
 - 写入 CCxE=1 以使能输出。
- 5) 通过将 *TIM1 控制寄存器 (TIM1_CR1)* 中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIM1_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=“0”，否则 TIM1_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。

下图给出了一个示例。



20.3.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 *TIM1 自动重载值寄存器*

(*TIM1_ARR*) 的值决定，其占空比则由 *TIM1_CCRx* 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 *OCx* 输出对应一个 PWM），只需向 *TIM1_CCMRx* 寄存器的 *OCxM* 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 *TIM1_CCMRx* 寄存器中的 *OCxPE* 位置 1 使能相应预装载寄存器，最后通过将 *TIM1 控制寄存器 (TIM1_CR1)* 中的 *ARPE* 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 *TIM1 事件产生寄存器 (TIM1_EGR)* 中的 *UG* 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 *TIM1 捕获/比较使能寄存器 (TIM1_CCER)* 的 *CCxP* 位）。可将其编程为高电平有效或低电平有效。通过 *CCxE*、*CCxNE*、*MOE*、*OSSI* 和 *OSSR* 位（寄存器 *TIM1_CCER* 和寄存器 *TIM1_BDTR*）的组合使能 *OCx* 输出。

在 PWM 模式（1 或 2）下，*TIM1 计数值寄存器 (TIM1_CNT)* 总是与 *TIM1_CCRx* 进行比较，以确定是 $TIM1_CCRx \leq TIM1_CNT$ 还是 $TIM1_CNT \leq TIM1_CCRx$ （取决于计数器计数方向）。

根据 *TIM1_CR1* 寄存器中的 *CMS* 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

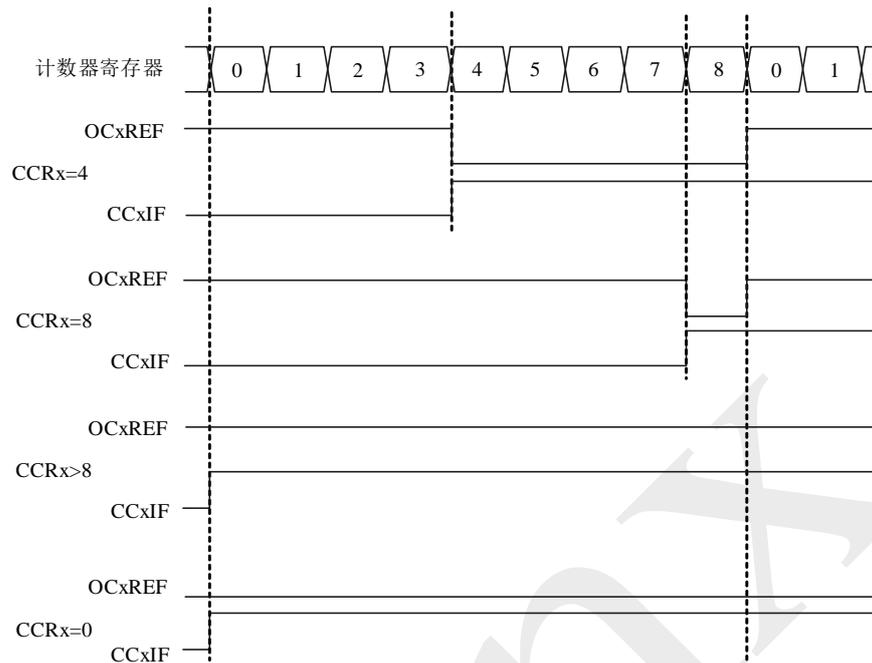
PWM 边沿对齐模式

- 递增计数配置

当 *TIM1 控制寄存器 (TIM1_CR1)* 中的 *DIR* 位为低时执行递增计数。请参见 [递增计数模式](#)。

以下以 PWM 模式 1 为例。只要 $TIM1_CNT < TIM1_CCRx$ ，PWM 参考信号 *OCxREF* 便为高电平，否则为低电平。如果 *TIM1_CCRx* 中的比较值大于自动重载值（*TIM1_ARR* 中），则 *OCxREF* 保持为“1”。如果比较值为 0，则 *OCxRef* 保持为“0”。

举例介绍边沿对齐模式的一些 PWM 波形（*TIM1_ARR=8*）。

图 20-36 边沿对齐模式的 PWM 波形 (ARR=8)


- **递减计数配置**

当 TIM1_CR1 寄存器中的 DIR 位为高时执行递减计数。请参考 [递减计数模式](#)。

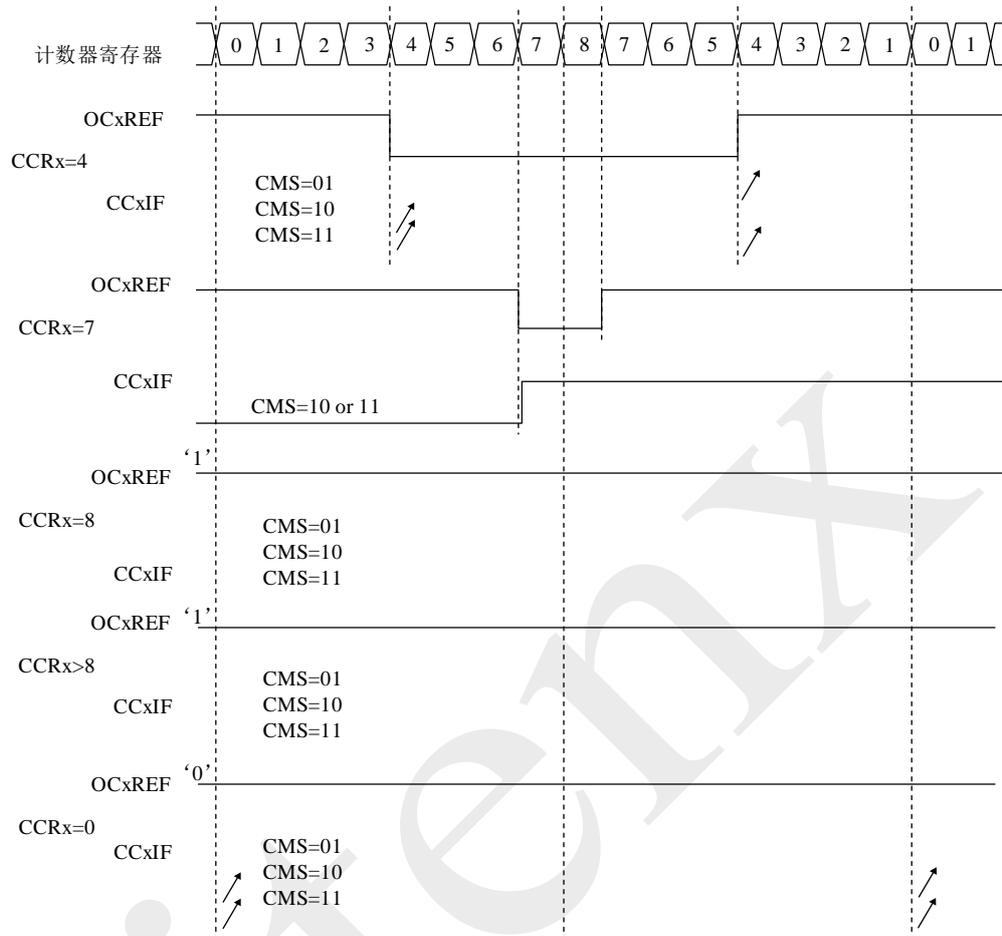
在 PWM 模式 1 下，只要 $TIM1_CNT > TIM1_CCR_x$ ，参考信号 OCxRef 即为低电平，否则其为高电平。如果 TIM1_CCRx 中的比较值大于 TIM1_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

PWM 中心对齐模式

当 *TIM1 控制寄存器 (TIM1_CR1)* 中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIM1_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见中心对齐模式（递增/递减计数）。

下图显示了中心对齐模式的 PWM 波形，在此例中：

- TIM1_ARR=8。
- PWM 模式为 PWM 模式 1。
- 在根据 TIM1_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 20-37 中心对齐模式 PWM 波形 (ARR=8)


中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIM1_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
 - 如果写入计数器中的值大于自动重载值 (TIM1_CNT > TIM1_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
 - 如果向计数器写入“0”或 TIM1_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIM1_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

20.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIM1_ARR 寄存器的值确定，而占空比和相移则由一对 TIM1_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIM1_CCR1 和 TIM1_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIM1_CCR3 和 TIM1_CCR4 控制

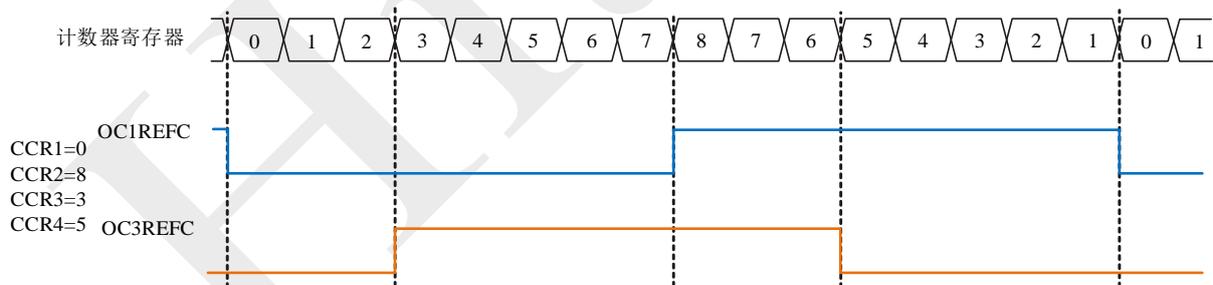
两个通道可以独立选择不对称 PWM 模式（每对 CC 寄存器一个 OCx 输出），只需向 TIM1_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注意：出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

下图显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 20-38 50%占空比时产生的 2 个相移 PWM 信号



20.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIM1_ARR 寄存器的值确定，而占空比和延时则由两个 TIM1_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIM1_CCR1 和 TIM1_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIM1_CCR3 和 TIM1_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIM1_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

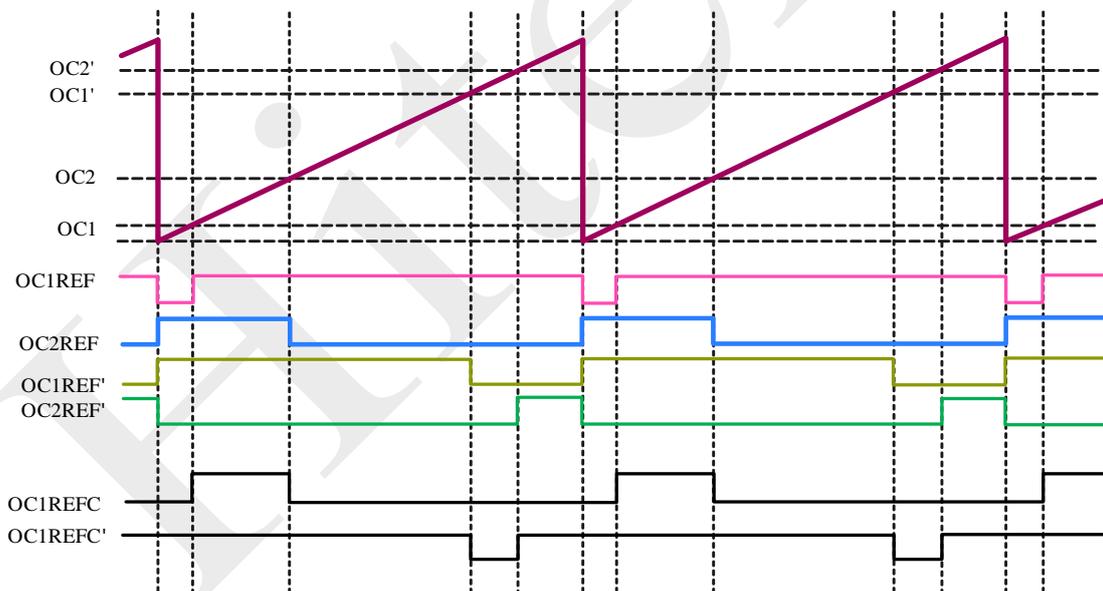
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注意： 出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

下图显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 1 在组合 PWM 模式 1 下配置。
- 通道 2 在 PWM 模式 2 下配置。

图 20-39 通道 1 和通道 2 上的组合 PWM 模式



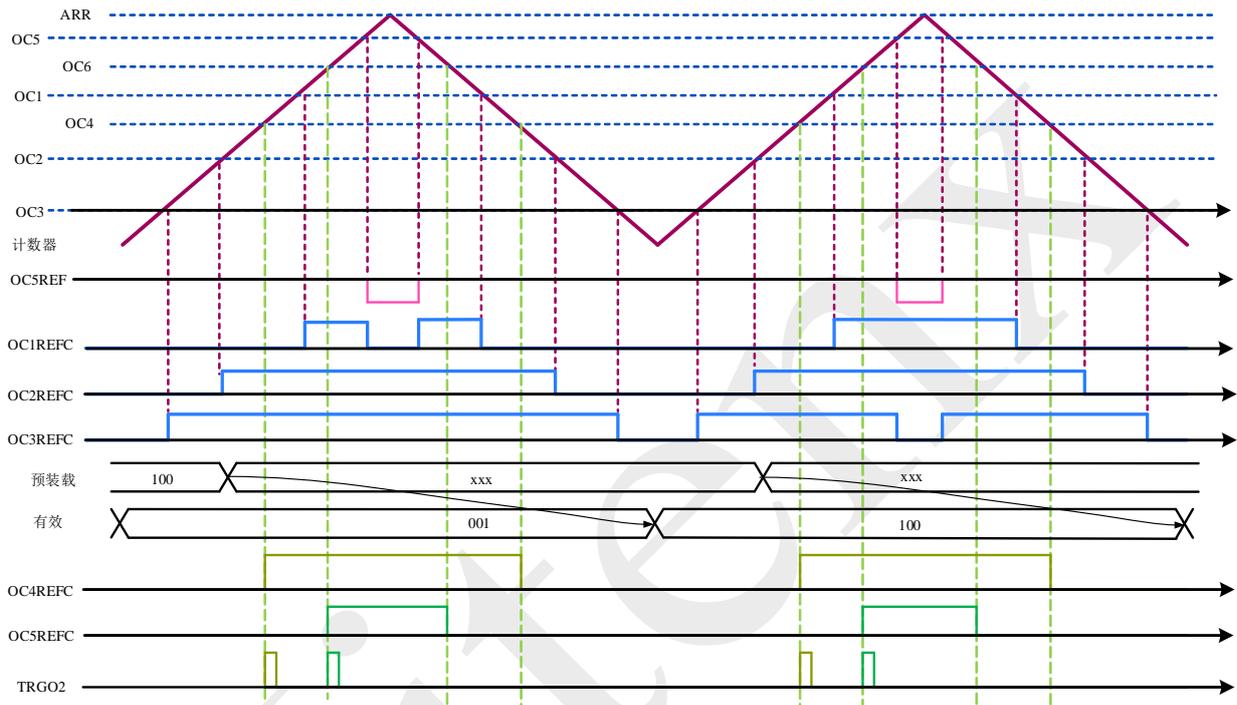
OC1REFC = OC1REF 与 OC2REF 的与计算
 OC1REFC' = OC1REF' 与 OC2REF' 的或计算

20.3.14 组合 3 相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。凭借 TIM1_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成。

- 如果 GC5C1 置 1，则 OC1REFC 由 TIM1_CCR1 和 TIM1_CCR5 控制
 - 如果 GC5C2 置 1，则 OC2REFC 由 TIM1_CCR2 和 TIM1_CCR5 控制
 - 如果 GC5C3 置 1，则 OC3REFC 由 TIM1_CCR3 和 TIM1_CCR5 控制
- 通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1]中的至少一位置 1。

图 20-40 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见 [ADC 同步](#)。

20.3.15 互补输出和死区插入

高级控制定时器（TIM1）可以输出两路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟……）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OC_x 或互补输出 OC_{xN}）。可通过对 [TIM 捕获/比较使能寄存器 \(TIM1_CCER\)](#) 中的 CC_{xP} 和 CC_{xNP} 位执行写操作来完成极性选择。

互补信号 OC_x 和 OC_{xN} 通过以下多个控制位的组合进行激活：TIM1_CCER 寄存器中的 CC_{xE} 和 CC_{xNE} 位以及 [TIM1 断路和死区寄存器 \(TIM1_BDTR\)](#) 和 [TIM1 控制寄存器 2 \(TIM1_CR2\)](#) 中的 MOE、OIS_x、OIS_{xN}、OSSI 和 OSSR 位。更多详细信息，请参见 [表：具有断路功能的互补通道 OC_x 和 OC_{xN} 的输出控制](#)

位。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在断路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）。

图 20-41 带死区插入的互补输出

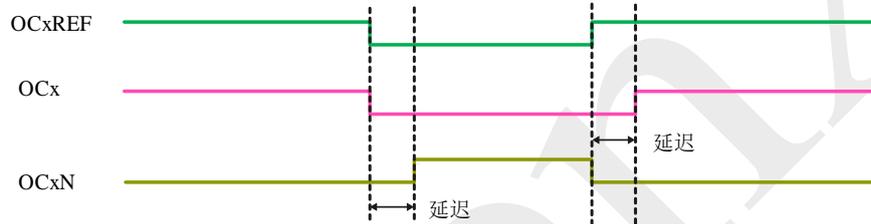


图 20-42 延迟时间大于负脉冲宽度的死区波形

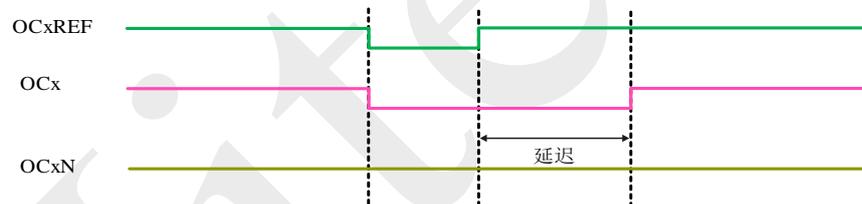
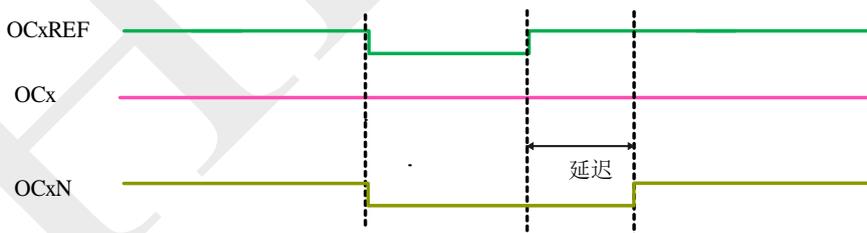


图 20-43 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 [TIM1 断路和死区寄存器 \(TIM1_BDTR\)](#) 寄存器中的 DTG 位进行编程。

OCxREF 重定向到 OCx 或 OCxN

将 OCxREF 重定向到 OCx 或 OCxN 在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIM1_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者

两个输出同时处于有效电平，两者互补并且带死区。

注意： 如果仅使能 $OCxN$ ($CCxE=0, CCxNE=1$)，两者不互补，一旦 $OCxREF$ 为高电平， $OCxN$ 即变为有效。例如，如果 $CCxNP=0$ ，则 $OCxN=OCxRef$ 。另一方面，如果同时使能 OCx 和 $OCxN$ ($CCxE=CCxNE=1$)， OCx 在 $OCxREF$ 为高电平时变为有效，而 $OCxN$ 则与之互补，在 $OCxREF$ 为低电平时变为有效。

20.3.16 使用断路功能

断路功能的目的是保护由 TIM1 定时器产生的 PWM 信号所驱动功率开关。两个断路输入通常连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个断路通道。一个断路通道收集系统级故障（时钟失效或内核的 LOOKUP 等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。断路 2 通道只包括应用故障，能够将输出强制为无效状态。

断路期间的输出使能信号和输出电平取决于多个控制位：

- **TIM1 断路和死区寄存器 (TIM1_BDTR)** 中的 MOE 位，允许通过软件使能/禁止输出，在发生断路和断路 2 事件时复位。
- TIM1_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放控制权给 GPIO 控制器（通常使其处于高阻态模式）
- **TIM1 控制寄存器 (TIM1_CR2)** 中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 $OCxN$ 输出同时设置为有效电平。更多详细信息，请参见表：**具有断路功能的互补通道 OCx 和 $OCxN$ 的输出控制位**。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIM1_BDTR 寄存器中的 BKE 位和 BK2E 位置 1，可使能断路功能。可通过配置同一寄存器中的 BKP 位和 BK2P 位来选择断路输入的极性。BKxE 和 BKxP 位可同时修改。对 BKxE 和 BKxP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIM1_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

可以使用 **TIM1 轮换功能寄存器 1 (TIM1_AF1)** 和 **TIM1 轮换功能寄存器 2**

(*TIM1_AF2*) 从多个源产生断路，这些源可以单独使能并使用可编程边沿有效性。

断路 (BRK) 通道的源:

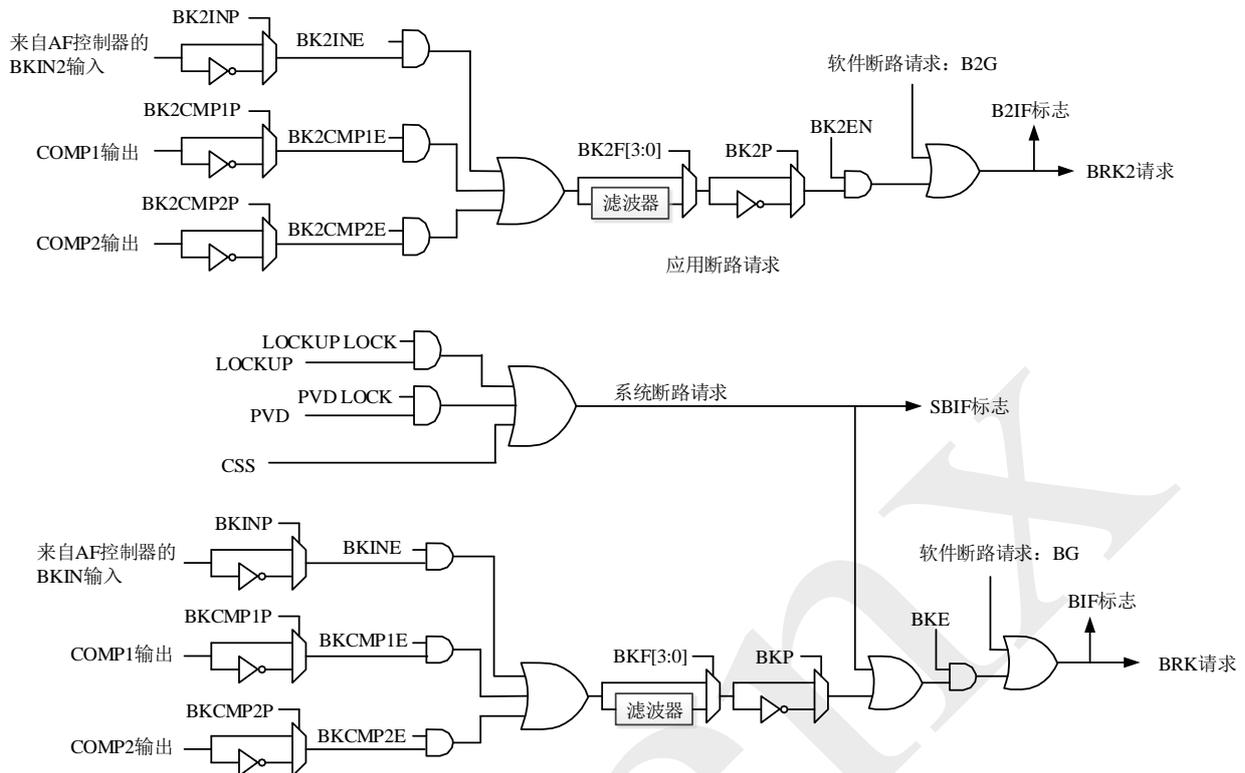
- 连接到 BKIN 引脚的外部源 (由 AFIO 控制器设定), 具有极性选择和可选的数字滤波
- 内部源:
 - Cortex®-M0+ LOCKUP 输出
 - PVD 输出
 - CSS 检测器产生时钟故障事件
 - 比较器的输出, 具有极性选择和可选的数字滤波

断路 2 (BRK2) 的源:

- 连接到 BK2IN 引脚的外部源 (由 AFIO 控制器设定), 具有极性选择和可选的数字滤波
- 来自比较器输出的内部源

也可由软件通过 *TIM1 事件产生寄存器 (TIM1_EGR)* 中的 BG 和 B2G 位产生断路事件。无论 BKE 和 BK2E 使能位的值如何, 都可以使用 BG 和 B2G 通过软件生成断路。

在所有源进入定时器 BRK 或 BRK2 输入之前, 对其进行 OR 运算, 如下图所示。

图 20-44 断路和断路 2 电路概述


注意： 只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 CSS）来保证能够处理断路事件。

发生断路之一（其中一个断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 *TIM1 控制寄存器 2 (TIM1_CR2)* 的 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
 - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
 - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况稍长一些（约 2 个 ck_tim 时钟周期）。
 - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器

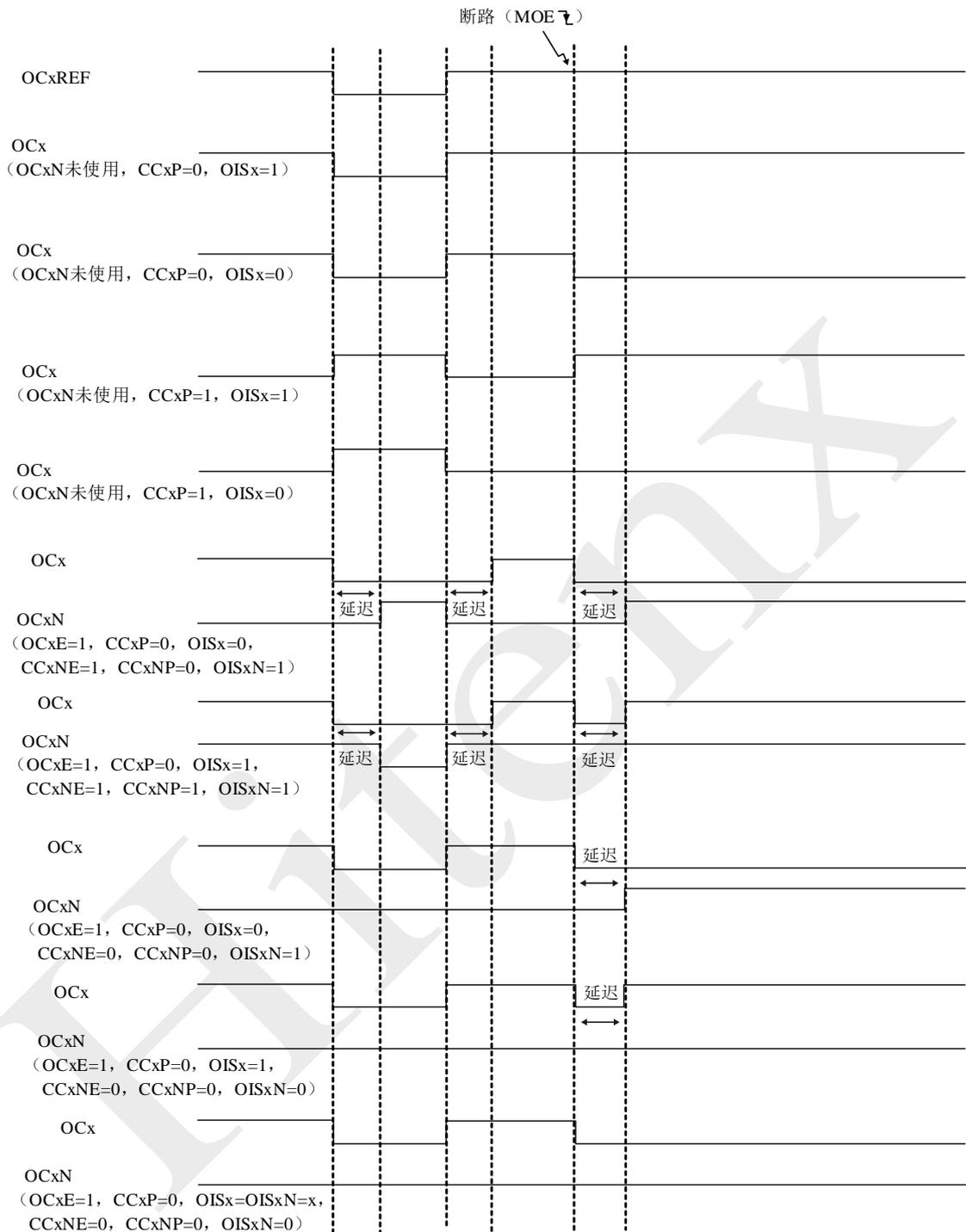
接管), 否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。

- 将断路状态标志 (*TIMI 状态寄存器 (TIMI_SR)* 中的 SBIF、BIF 和 B2IF 位) 置 1。如果 *TIMI DMA/中断使能寄存器 (TIMI_DIER)* 中的 BIE 位置 1, 则会产生中断。
- 如果 *TIMI 断路和死区寄存器 (TIMI_BDTR)* 中的 AOE 位置 1, 则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。例如, 这可用于执行调节。否则, MOE 将始终保持低电平, 直到应用将其再次置 1。这种情况下, 这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注意: 断路输入为电平有效。因此, 当断路输入有效电平时, 不能将 MOE 位置 1 (自动或通过软件)。同时, 不能将状态标志 BIF 和 B2IF 清 0。

除断路输入和输出管理外, 断路电路内部还实施了写保护, 用以保护应用的安全。通过该功能, 用户可冻结多个参数配置 (死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性)。应用可以通过 *TIMI 断路和死区寄存器 (TIMI_BDTR)* 中的 LOCK 位, 从 3 种保护级别中进行选择。MCU 复位后只能对 LOCK 位执行一次写操作。

下图所示为输出对断路响应行为的示例。

图 20-45 响应 BRK 上的断路事件的不同输出行为 (OSSSI=1)


两个断路输入针对定时器输出具有不同的行为:

- BRK 输入可禁止 (无效状态) PWM 输出, 也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止 (无效状态) PWM 输出。

BRK 输入的优先级高于 BRK2 输入, 如下表所示。

注意: BRK2 必须只在 OSSR=OSSSI=1 时使用。

表 20-1 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN 输出 (下桥臂开关)	OCx 输出 (上桥臂开关)
有效	×	- 无效，之后强制为输出状态（死区后） - 如果 OSSI=0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	有效	无效	关闭	关闭

下图给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIM1_CCER 寄存器中的 CCxP=CCxNP=0）。

图 20-46 BRK 和 BRK2 引脚使能后的 PWM 输出状态（OSSI=1）

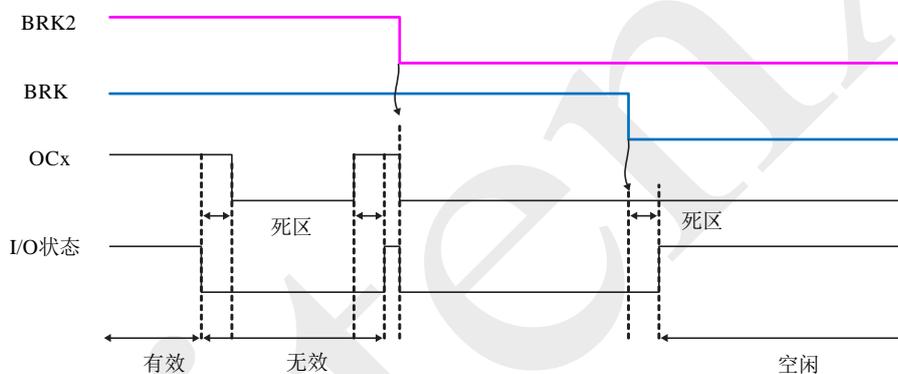
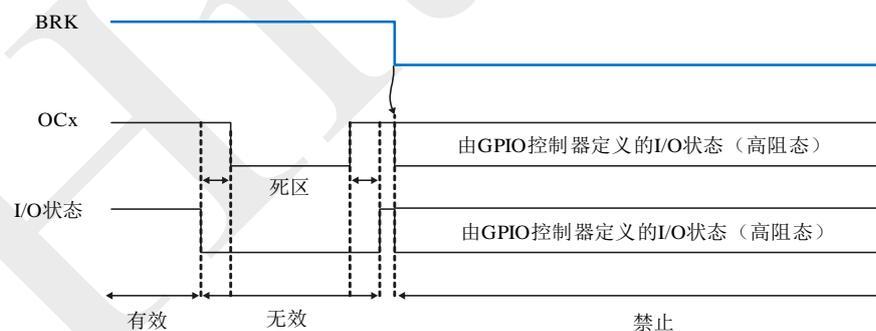


图 20-47 BRK 使能后的 PWM 输出状态（OSSI=0）



20.3.17 双向断路输入

TIM1 具有双向断路 I/O，如 [图：输出重定向](#)（图中未显示 BRK2 请求）所示。它们可以：

- 板级全局断路信号，用于向外部 MCU 或栅极驱动器发送故障信号，唯一的引脚作为输入和输出状态引脚。
- 在必须将多个内部和外部断路输入合并时，将内部断路源和多个外部开漏比较器输出”或”连接在一起，触发唯一断路事件。

使用 *TIM1 断路和死区寄存器 (TIM1_BDTR)* 的 BKBID 和 BK2BID 位，将断路和断路 2 输入配置为双向模式。可以使用 TIM1_BDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（锁定级别 1 或更高）。

双向模式可以用于断路和断路 2 输入，需要将 I/O 配置为低电平有效极性的开漏模式（使用 BKINP、BKP、BK2INP 和 BK2P 位进行配置）。任何来自系统（例如 CSS）、片上外设或断路输入的断路请求都会强制将断路输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件断路事件（BG 和 B2G）也会导致断路 I/O 被强制置为“0”，从而向外部组件指示定时器已进入断路状态。但是仅在使能断路（BKxEN=1）时有效。当生成软件断路事件，并且 BKxEN=0 时，输出会置于安全状态，并将断路标志置 1，但对断路（2）I/O 无影响。

安全解除机制可防止系统被明确锁定（断路输入上的低电平会触发断路，进而将相同输入强制置为低电平）。

当 BKDSRM（BK2DSRM）位置 1 时，会释放断路输出以清除故障信号，从而使系统能够重新获得保护。

在任何情况下都不能禁止断路保护电路：

- 断路输入路径始终有效：即使 BKDSRM（BK2DSRM）位置 1 且释放开漏控制，断路事件也仍然有效。这样可以在发生断路期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BKxDSRM 位不能解除断路保护（请参见下表）。

表 20-2 断路保护解除条件

MOE	BKDIR (BK2DIR)	BKDSRM (BK2DSRM)	断路保护状态
0	0	×	保护
0	1	0	保护
0	1	1	解除
1	×	×	保护

启动和重新启动断路电路

默认情况下（外设复位配置）会启动断路电路（在输入或双向模式下）。

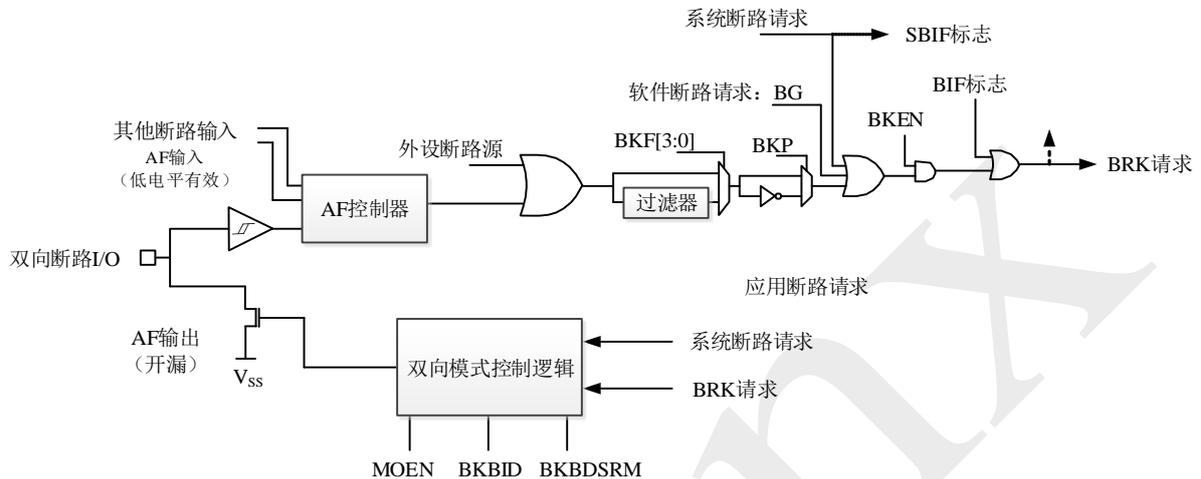
发生断路（断路 2）事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM（BK2DSRM）位置 1，以释放输出控制
- 软件必须等待系统断路条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）

- 软件必须轮询 BKDSRM (BK2DSRM) 位，直到该位由硬件清零 (当应用程序断路条件消失时)

此后，断路电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 20-48 输出重定向 (图中未显示 BRK2 请求)



20.3.18 发生外部事件时清除 OCxREF 信号

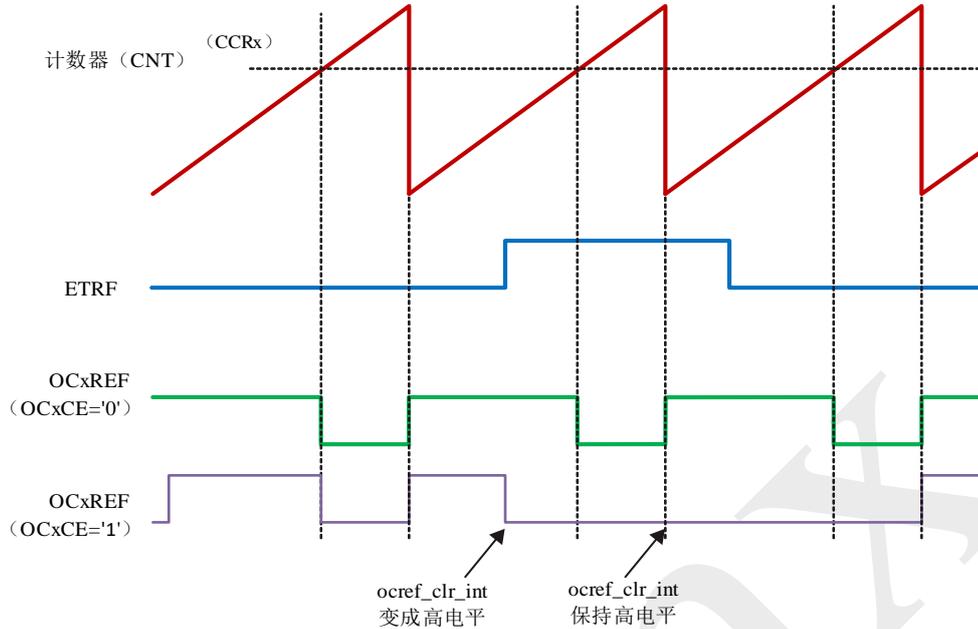
对于给定通道，在 ETRF 输入施加高电平 (相应 TIM1_CCMRx 寄存器中的 OCxCE 使能位置 1)，可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。

选择 ETRF 时，ETR 必须配置如下：

- 1) 必须关闭外部触发预分频器：*TIM1 从模式控制寄存器 (TIM1_SMCR)* 中的 ETPS[1:0] 位置 “00”。
- 2) 必须禁止外部时钟模式 2：*TIM1_SMCR* 寄存器中的 ECE 位置 “0”。
- 3) 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据用户需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIM1 编程为 PWM 模式。

图 20-49 清除 TIM1 的 OCxREF


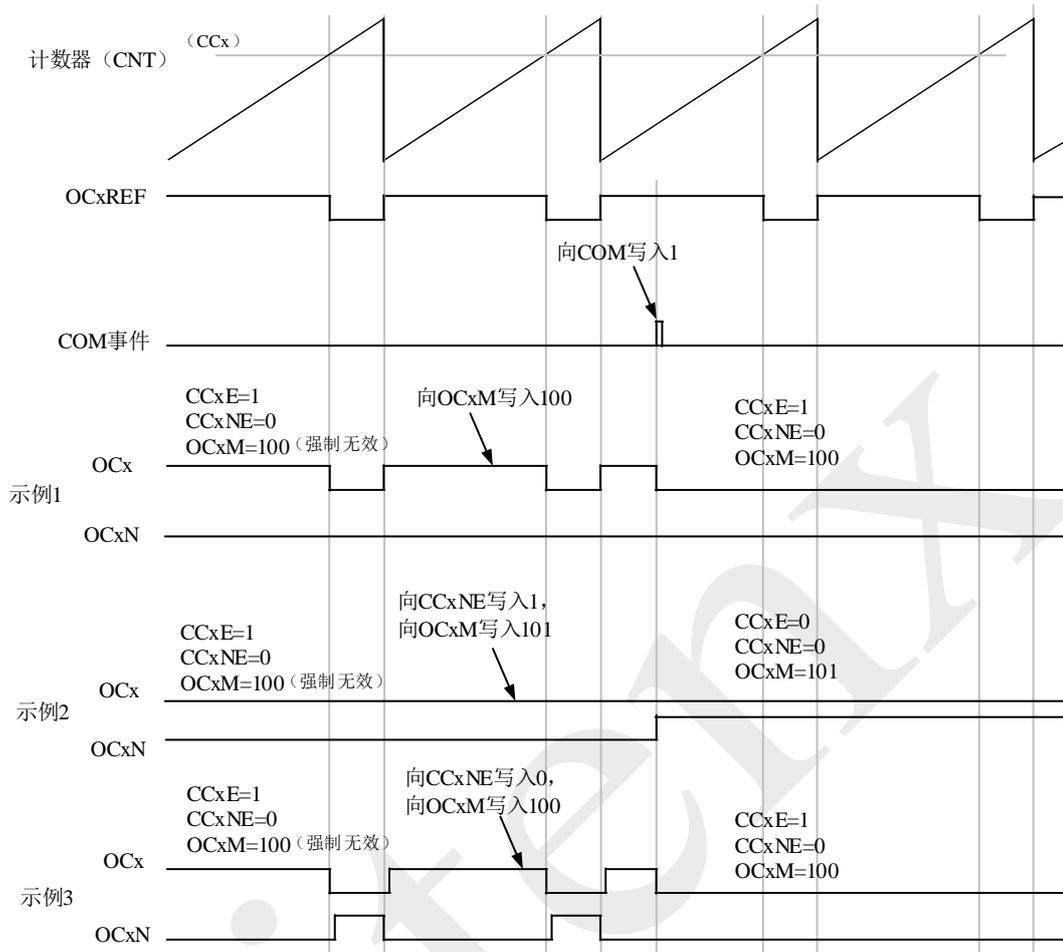
注意： 如果 PWM 的占空比为 100% ($CCx > ARR$)，则下次计数器溢出时会再次使能 $OCxREF$ 。

20.3.19 生成 6 步 PWM

当通道使用互补输出时， $OCxM$ 、 $CCxE$ 和 $CCxNE$ 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 $TIM1_EGR$ 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位 ($TIM1$ 状态寄存器 ($TIM1_SR$) 中的 COMIF 位) 将会置 1。这时，如果 $TIM1$ DMA/中断使能寄存器 ($TIM1_DIER$) 中的 COMIE 位置 1，将产生中断；如果 $TIM1_DIER$ 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

下图以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 $OCxN$ 输出的行为。

图 20-50 COM 事件生成 6 步 PWM 的示例 (OSSR=1)


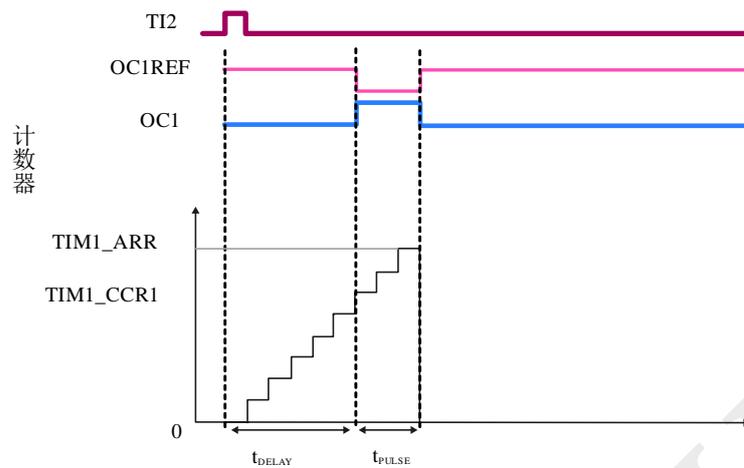
20.3.20 单脉冲模式

单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下, 计数器可以在一个激励信号的触发下启动, 并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 *TIM1 控制寄存器 (TIM1_CR1)* 中的 OPM 位置 1, 即可选择单脉冲模式。这样, 发生下一更新事件 UEV 时, 计数器将自动停止。

只有当比较值与计数器初始值不同时, 才能正确产生一个脉冲。启动前 (定时器等等待触发时), 必须进行如下配置:

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

图 20-51 单脉冲模式示例


例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

- 1) 通过定时器输入选择 *TIM1 定时器输入选择寄存器 (TIM1_TISEL)* 中的 TI1SEL[3:0] 位域选择适当的 TI1x 源（内部或外部）；
- 2) 在 TIM1_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
- 3) 在 *TIM1 捕获/比较使能寄存器 (TIM1_CCER)* 中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
- 4) 在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 TS=“110”，将 TI2FP2 配置为从模式控制器的触发（TRGI）。
- 5) 在 TIM1_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 *TIM1 控制寄存器 (TIM1_CR1)* 的值定义。
- t_{PULSE} 由自动重载值与比较值之差（TIM1_ARR-TIM1_CCR1）来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM1_CCMR1 寄存器中写入 OC1M=“111”，以使能 PWM 模式 2。如果需要，可选择在 TIM1_CCMR1 寄存器的 OC1PE 和 TIM1_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIM1_CC1 寄存器中写入比较值并在 TIM1_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIM1_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM1_CR1 寄存器的 OPM 位写

入“1”，以便在发生下一更新事件（计数器从自动重载值返回到0）时使计数器停止计数。TIM1_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（ t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIM1_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

20.3.21 可重触发单脉冲模式

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如 [单脉冲模式](#) 所述：

- -发生触发时，脉冲立即产生（无可编程延时）
- -如果在上一个触发完成前发生新的触发，脉冲将延长

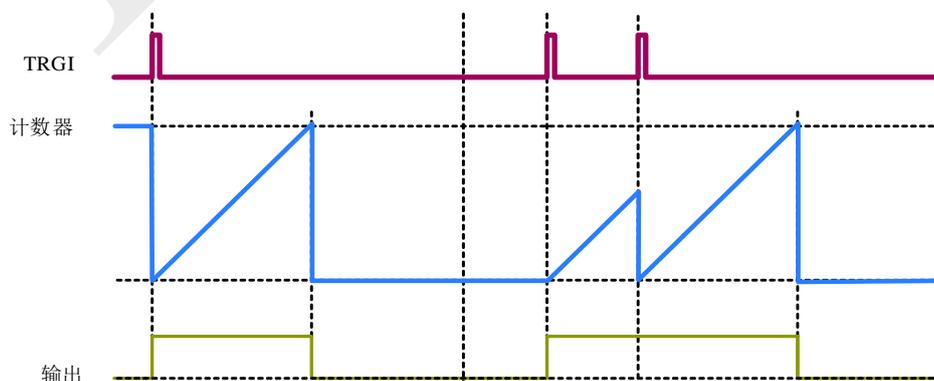
定时器必须处于从模式，TIM1_SMCR 寄存器中的位 SMS[3:0]=“1000”（组合复位+触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0]位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCx 必须高于或等于 ARR。

注意： 出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

注意： 此模式不能与中心对齐 PWM 模式一起使用。在 [TIM1 控制寄存器 1 \(TIM1_CR1\)](#) 中必须设置 CMS[1:0]=“00”。

图 20-52 可再触发单脉冲模式



20.3.22 编码接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIM1_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。TI1 和 TI2 两个输入用于连接正交编码器。请参见 *表：计数方向与编码器信号的关系*。如果使能计数器（在 *TIM1 控制寄存器 (TIM1_CR1)* 的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIM1_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIM1_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIM1_ARR。同样，捕获、比较、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注意： 使能编码器模式时，预分频器必须设置为零。

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 20-3 计数方向与编码器信号的关系

有效边沿	相反信号的电平 (TI1FP1 对应 TI2, TI2FP2 对应 TI1)	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在TI1处 计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在TI2处 计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
仅在TI1和TI2处 均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

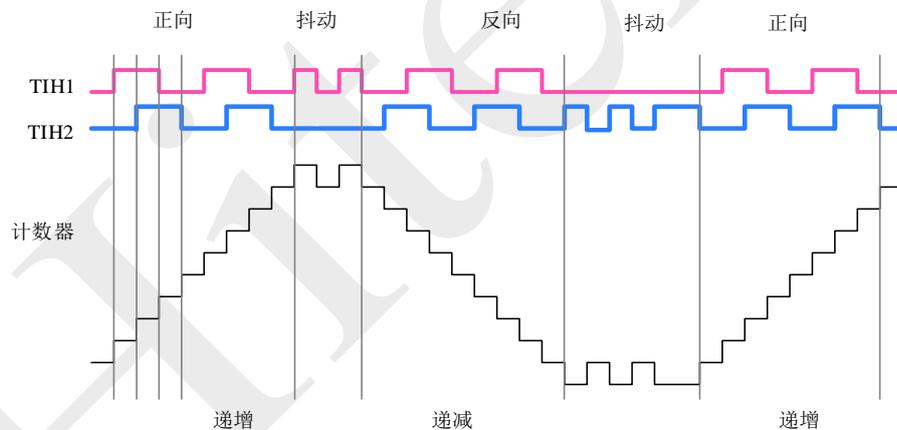
正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将

编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

下图以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

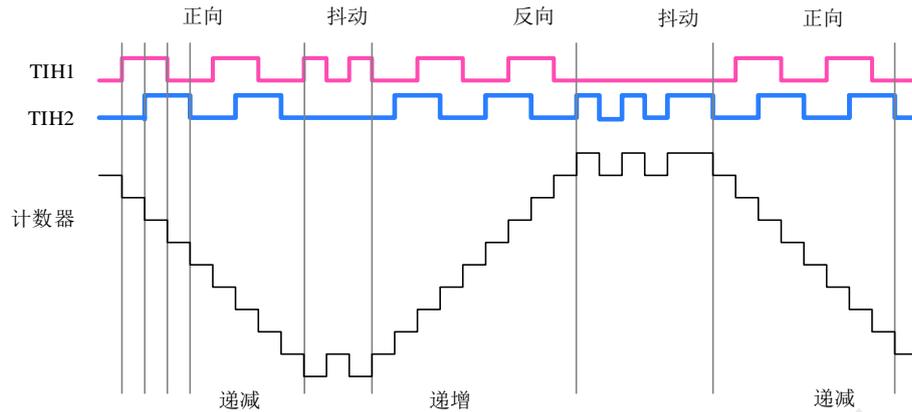
- CC1S= “01” (TIM1_CCMR1 寄存器, TI1FP1 映射到 TI1 上)。
- CC2S= “01” (TIM1_CCMR2 寄存器, TI1FP2 映射到 TI2 上)。
- CC1P= “0”, CC1NP= “0” (*TIM1 捕获/比较使能寄存器 (TIM1_CCER)*, TI1FP1 未反相, TI1FP1=TI1)。
- CC2P= “0”, CC2NP= “0” (TIM1_CCER 寄存器, TI1FP2 未反相, TI1FP2=TI2)。
- SMS= “011” (*TIM1 从模式控制寄存器 (TIM1_SMCR)*, 两个输入在上升沿和下降沿均有效)。
- CEN= “1” (*TIM1 控制寄存器 1 (TIM1_CR1)*, 使能计数器)。

图 20-53 编码器接口模式下的计数器工作示例



下图举例说明 TI1FP1 极性反相时计数器的行为 (除 CC1P= “1” 外, 其他配置与上例相同)。

图 20-54 TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过 DMA 请求读取计数器值。

在 32 位定时器实现中，当 *TIM1 控制寄存器 (TIM1_CR1)* 的 UIFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

20.3.23 UIF 位重映射

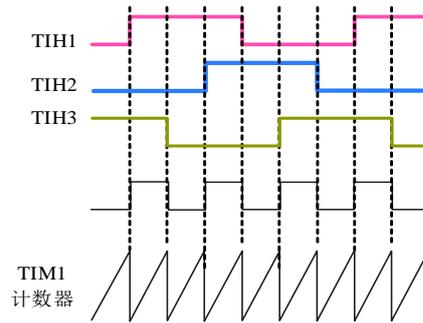
TIM1_CR1 寄存器中的 UIFREMAP 位强制将更新中断标志 UIF 连续复制到 *TIM1 计数值寄存器 (TIM1_CNT)* 的位 31 (TIM1_CNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

20.3.24 定时器输入异或功能

通过 *TIM1 控制寄存器 2 (TIM1_CR2)* 中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIM1_CH1 到 TIM1_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔（如下图所示）。

图 20-55 测量 3 个信号上边沿之间的时间间隔


20.3.25 霍尔传感器接口

可通过用于产生电机驱动 PWM 信号的高级控制定时器 TIM1 以及 [图：霍尔传感器接口的示例](#)中称为“接口定时器”的另一个定时器 TIM1，实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIM1_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（[图：捕获/比较通道（示例：通道 1 输入阶段）](#)）。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器（TIM1）各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器（TIM1）。

示例：霍尔输入与一个 TIM1 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

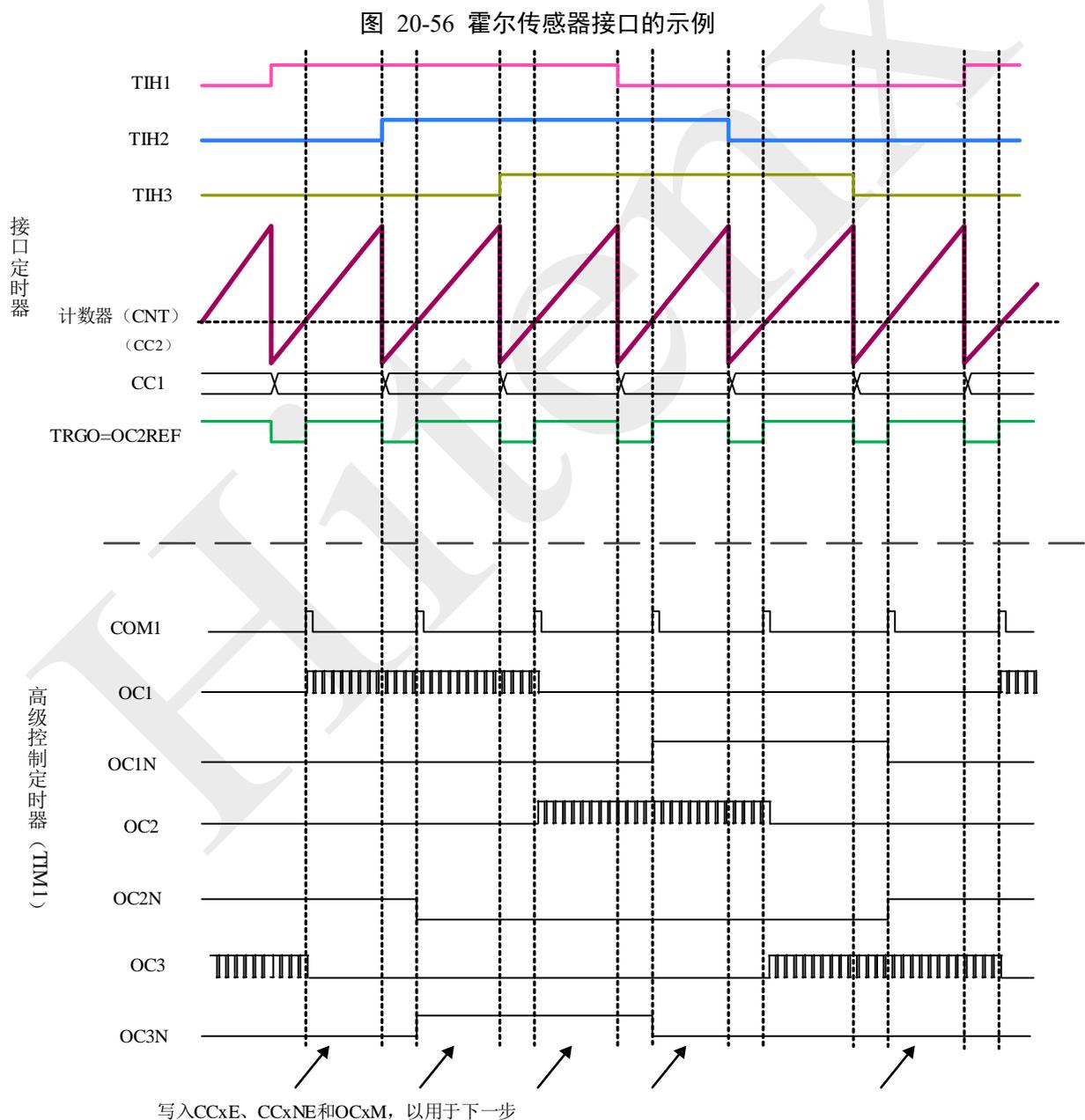
- 向 TIM1_CR2 寄存器的 TI1SEL 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIM1_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIM1_CCMR1 寄存器的 CC1S 位写入“11”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM2 模式，并具有所需延迟：向 TIM1_CCMR1 寄存器

的 OC2M 位写入“111”，CC2S 位写入“00”。

- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIM1_CR2 寄存器的 MM_SEL 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIM1_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIM1_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

下图为本示例的示意图。



20.3.26 定时器同步

TIM1 定时器从内部连接在一起，以实现定时器同步或链接。详细信息可参考定时器同步模式。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 *TIM1 控制寄存器 1 (TIM1_CR1)* 中的 URS 位为 0，且 UDIS 位也为 0，则会生成更新事件 UEV。然后，所有预装载寄存器 (TIM1_ARR 和 TIM1_CCRx) 都将更新。

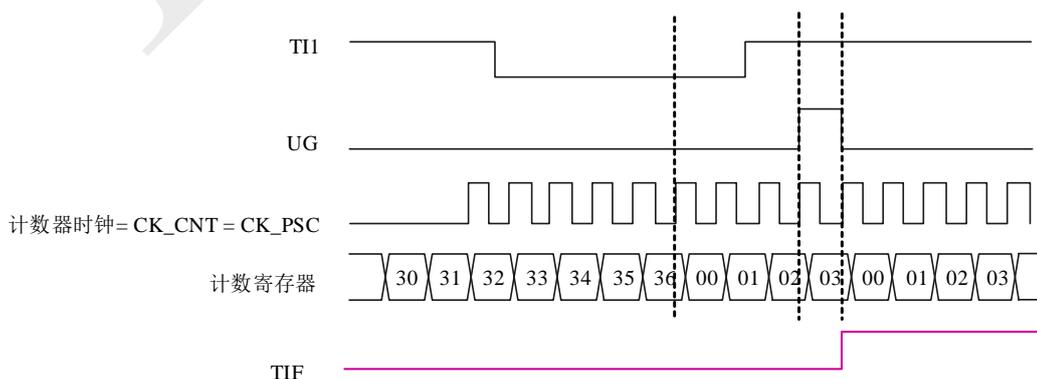
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM1_CCMR1 寄存器中的 CC1S=01。在 *TIM1 捕获/比较使能寄存器 (TIM1_CCER)* 中写入 CC1P=“0”和 CC1NP=“0”，验证极性（仅检测上升沿）。
- 在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 SMS=“100”，将定时器配置为复位模式。在 TIM1_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
- 在 TIM1_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志 (TIM1_SR 寄存器中的 TIF 位) 置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIM1_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIM1_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 20-57 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

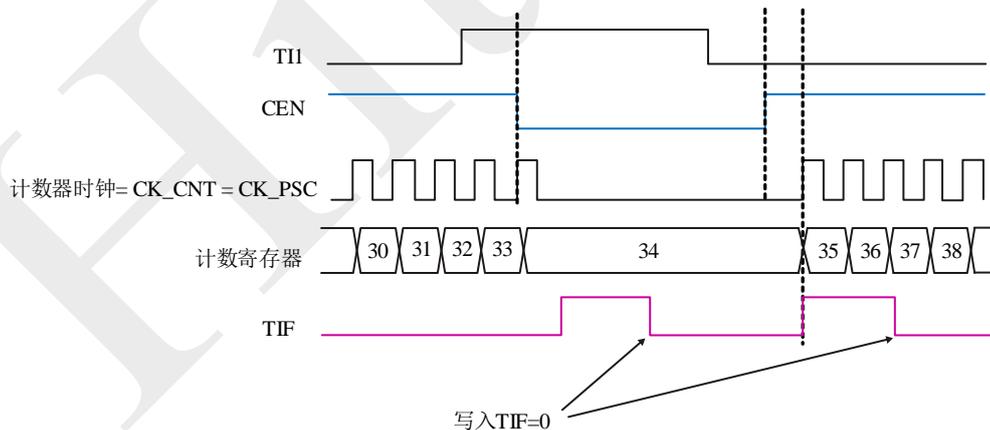
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM1_CCMR1 寄存器中的 CC1S=01。在 *TIM1 捕获/比较使能寄存器 (TIM1_CCER)* 中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 中写入 SMS=“101”，将定时器配置为门控模式。在 TIM1_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
- 在 *TIM1 控制寄存器 1 (TIM1_CR1)* 中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，*TIM1 状态寄存器 (TIM1_SR)* 中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 20-58 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIM1_CCMR1

寄存器中的 CC2S=“01”。在 TIM1_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。

- 在 TIM1_SMCR 寄存器中写入 SMS=“110”，将定时器配置为触发模式。在 TIM1_SMCR 寄存器中写入 TS=“110”，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 20-59 触发模式下的控制电路



从模式：组合复位+触发模式

在这种情况下，在出现所选触发输入（TRGI）上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

从模式：外部时钟模式 2+触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIM1_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

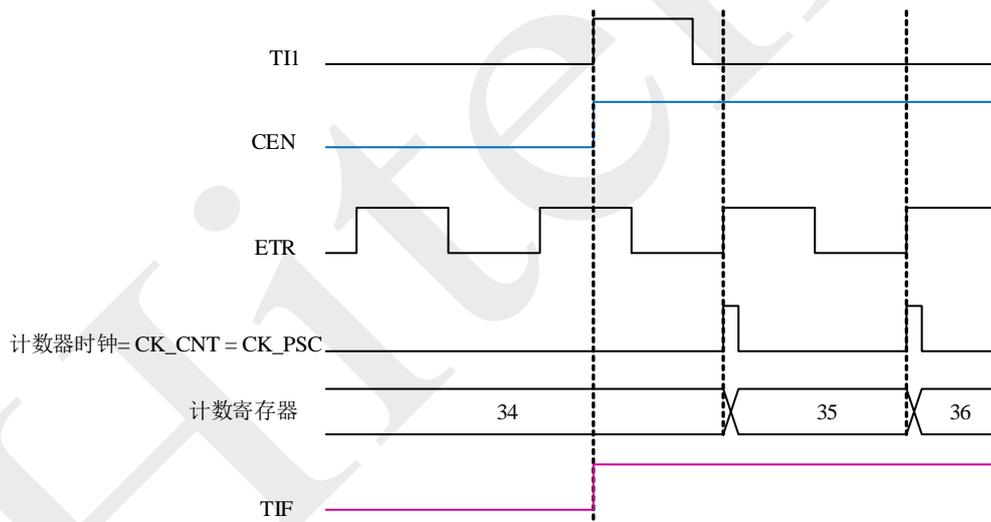
- 1) 通过对 *TIM1 从模式控制寄存器 (TIM1_SMCR)* 进行如下编程，配置外部触发输入电路：
 - ETF=“0000”：无滤波器。
 - ETPS=“00”：禁止预分频器。
 - ETP=“0”：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。

- 2) 如下配置通道 1，以检测 TI 的上升沿：
 - IC1F= “0000”：无滤波器。
 - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
 - TIM1_CCMR1 寄存器中 CC1S= “01”，只选择输入捕获源。
 - TIM1_CCER 寄存器中 CC1P= “0” 且 CC1NP= “0”，以确定极性（仅检测上升沿）。
- 3) 在 TIM1_SMCR 寄存器中写入 SMS= “110”，将定时器配置为触发模式。在 TIM1_SMCR 寄存器中写入 TS= “101”，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 20-60 外部时钟模式 2+触发模式下的控制电路



注意： 必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

20.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，

它们可通过 *TIM1 控制寄存器 2 (TIM1_CR2)* 中的 MM_SEL2[3:0]位选择。

图：三相组合 PWM 信号 (每个周期多个触发脉冲) 给出了三相电机驱动的应用示例。

注意： 必须先使能接收 TRGO 或 TRGO2 信号的从外设 (定时器、ADC 等) 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率 (预分频器)。

注意： 必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。

20.3.28 DMA 连续传输模式

TIM1 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟 *TIM1 全传输地址寄存器 (TIM1_DMAR)*。发生给定的定时器事件时，定时器会启动 DMA 请求序列 (突发)。每次写入 TIM1_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIM1_DCR 寄存器中的 DBL[4:0]位设置 DMA 连续传送长度。当对 TIM1_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数 (按半字或字节)。

TIM1_DCR 寄存器中的 DBA[4:0]位定义 DMA 传送的 DMA 基址 (通过 TIM1_DMAR 地址执行读/写访问时)。DBA 定义为从 TIM1_CR1 寄存器地址开始计算的偏移量：

示例：

00000: TIM1_CR1

00001: TIM1_CR2

00010: TIM1_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCx 寄存器 (x=2、3、4) 的内容更新为通过 DMA 传输到 CCx 寄存器中的多个半字。

具体操作步骤如下：

- 1) 将相应的 DMA 通道配置如下：
 - DMA 通道外设地址为 DMA 寄存器地址。
 - DMA 通道存储器地址为包含要通过 DMA 传输到 CCx 寄存器的数据的 RAM 缓冲区地址。

- 要传输的数据量=3（参见下文注释）。
 - 禁止循环模式。
- 2) 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：DBL=3 次传输，DBA=0xE。
 - 3) 使能 TIM1 更新 DMA 请求（*TIM1 DMA/中断使能寄存器 (TIM1_DIER)* 中的 UDE 位置 1）。
 - 4) 使能 TIM1。
 - 5) 使能 DMA 通道。

本例适用于每个 CCx 寄存器只更新一次的情况。如果每个 CCx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CC2，data2 传输到 CC3，data3 传输到 CC4；在第二个更新 DMA 请求期间，data4 传输到 CC2，data5 传输到 CC3，data6 传输到 CC4。

注意： 可以将空值写入保留的寄存器中。

20.3.29 调试模式

当微控制器进入调试模式（Cortex®-M0+内核停止）时，TIM1 计数器会根据 *DBG_APB_FZ2 寄存器* 的 TIM1_HOLD 配置位选择继续正常工作或者停止工作。为了安全起见，当计数器停止（DBG_TIM1_STOP=1）时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位=1），或者通过 GPIO 控制器（OSSI 位=0）来控制输出，通常将其强制为高阻态。

20.4 TIM1 寄存器

20.4.1 TIM1 控制寄存器 (TIM1_CR1)

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

位/位域	名称	描述
31:12	保留	必须保持复位值

11	UIFREMAP	<p>UIF状态位重映射</p> <p>0: UIF状态位不进行重映射</p> <p>1: UIF状态位重映射到TIM1_CNT寄存器的bit31</p>
10	保留	必须保持复位值
9:8	CKD[1:0]	<p>时钟分频</p> <p>此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (ETR、TIx) 之间的分频比</p> <p>00: $t_{DTS}=t_{CK_INT}$</p> <p>01: $t_{DTS}=2\times t_{CK_INT}$</p> <p>10: $t_{DTS}=4\times t_{CK_INT}$</p> <p>11: 保留</p>
7	ARPE	<p>自动重载预装载使能</p> <p>0: TIM1_ARR寄存器不进行缓冲</p> <p>1: TIM1_ARR寄存器进行缓冲</p>
6:5	CMS[1:0]	<p>中心对齐模式选择</p> <p>00: 边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。</p> <p>01: 中心对齐模式1。计数器交替进行递增计数和递减计数。仅当计数器递减计时，配置为输出的通道 (TIM1_CCMRx寄存器中的CCxS=00) 的输出比较中断标志才置1。</p> <p>10: 中心对齐模式2。计数器交替进行递增计数和递减计数。仅当计数器递增计时，配置为输出的通道 (TIM1_CCMRx寄存器中的CCxS=00) 的输出比较中断标志才置1。</p> <p>11: 中心对齐模式3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道 (TIM1_CCMRx寄存器中的CCxS=00) 的输出比较中断标志都会置1。</p> <p><i>注意: 只要计数器处于使能状态 (CEN=1), 就不得从边沿对齐模式切换为中心对齐模式。</i></p>
4	DIR	<p>方向</p> <p>0: 计数器递增计数</p> <p>1: 计数器递减计数</p> <p><i>注意: 当定时器配置为中心对齐模式或编码器模式时, 该位为只读状态。</i></p>
3	OPM	<p>单脉冲模式</p> <p>0: 计数器在发生更新事件时不会停止计数</p> <p>1: 计数器在发生下一更新事件时停止计数 (将CEN位清零)</p>
2	URS	更新请求源

此位由软件置1和清零，用以选择UEV事件源。

0: 使能时，所有以下事件都会产生更新中断或DMA请求。此类事件包括：

- 计数器上溢/下溢
- 将UG位置1
- 通过从模式控制器生成的更新事件

1: 使能时，只有计数器上溢/下溢会生成更新中断或DMA请求。

1 UDIS

更新禁止

此位由软件置1和清零，用以使能/禁止UEV事件生成。

0: 使能UEV。更新（UEV）事件可通过以下事件之一产生：

- 计数器上溢/下溢
- 将UG位置1
- 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1: 禁止UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC和CCx）保持不变。但如果将UG位置1，或者从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

0 CEN

计数器使能

0: 禁止计数器

1: 使能计数器

注意：只有事先通过软件将CEN位置1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将CEN位置1。在单脉冲模式下，当发生更新事件时会自动将CEN位清零。

20.4.2 TIM1 控制寄存器 2 (TIM1_CR2)

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
Res.								MMS2[3:0]				Res	OIS6	Res	OIS5
								rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TIIS	MMS[2:0]			CCDS	CCUS [2:0]		CC PC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位/位域

名称

描述

31:24 保留 必须保持复位值

23:20 MMS2[3:0] 主模式选择2
这些位可选择将发送到ADC以实现同步的信息（TRGO2）。这些

位的组合如下：

- 0000: 复位——TIM1_EGR寄存器中的UG位用作触发输出（TRGO2）。如果复位由触发输入生成（从模式控制器配置为复位模式），则TRGO2上的信号相比实际复位会有延迟。
 - 0001: 使能——计数器使能信号CEN用作触发输出（TRGO2）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由CEN控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO2上会存在延迟，选择主/从模式时除外（请参见TIM1从模式控制寄存器（TIM1_SMCR）中MSM位的说明）。
 - 0010: 更新——选择更新事件作为触发输出（TRGO2）。例如，主定时器可用作从定时器的预分频器。
 - 0011: 比较脉冲——CC1IF标志置1时（即使已为高），只要发生捕获或比较匹配，触发输出（TRGO2）都会发送一个正脉冲。
 - 0100: 比较——OC1REF信号用作触发输出（TRGO2）
 - 0101: 比较——OC2REF信号用作触发输出（TRGO2）
 - 0110: 比较——OC3REF信号用作触发输出（TRGO2）
 - 0111: 比较——OC4REF信号用作触发输出（TRGO2）
 - 1000: 比较——OC5REF信号用作触发输出（TRGO2）
 - 1001: 比较——OC6REF信号用作触发输出（TRGO2）
 - 1010: 比较脉冲——OC4REF上升沿或下降沿时，TRGO2上生成脉冲
 - 1011: 比较脉冲——OC6REF上升沿或下降沿时，TRGO2上生成脉冲
 - 1100: 比较脉冲——OC4REF或OC6REF上升沿时，TRGO2上生成脉冲
 - 1101: 比较脉冲——OC4REF上升沿或OC6REF下降沿时，TRGO2上生成脉冲
 - 1110: 比较脉冲——OC5REF或OC6REF上升沿时，TRGO2上生成脉冲
 - 1111: 比较脉冲——OC5REF上升沿或OC6REF下降沿时，TRGO2上生成脉冲
- 注意：必须先使能从定时器或ADC的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或ADC的时钟。*

19	保留	必须保持复位值
18	OIS6	输出空闲状态6（OC6输出），请参见OIS1位
17	保留	必须保持复位值
16	OIS5	输出空闲状态5（OC5输出），请参见OIS1位
15	保留	必须保持复位值

14	OIS4	输出空闲状态4 (OC4输出), 请参见OIS1位
13	OIS3N	输出空闲状态3 (OC3N输出), 请参见OIS1N位
12	OIS3	输出空闲状态3 (OC3输出), 请参见OIS1位
11	OIS2N	输出空闲状态2 (OC2N输出), 请参见OIS1N位
10	OIS2	输出空闲状态2 (OC2输出), 请参见OIS1位
9	OIS1N	输出空闲状态1 (OC1N输出) 0: 当MOE=0时, 经过死区时间后OC1N=0 1: 当MOE=0时, 经过死区时间后OC1N=1 <i>注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1、2或3, 此位即无法修改。</i>
8	OIS1	输出空闲状态1 (OC1输出) 0: 当MOE=0时, (如果OC1N有效, 则经过死区时间之后)OC1=0 1: 当MOE=0时, (如果OC1N有效, 则经过死区时间之后)OC1=1 <i>注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1、2或3, 此位即无法修改。</i>
7	TIIS	TI1选择 0: TIM1_CH1引脚连接到TI1输入 1: TIM1_CH1、CH2和CH3引脚连接到TI1输入 (异或组合)
6:4	MMS[2:0]	主模式选择 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下: 000: 复位——TIM1_EGR寄存器中的UG位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则TRGO上的信号相比实际复位会有延迟。 001: 使能——计数器使能信号CNT_EN用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号由CEN控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时, TRGO上会存在延迟, 选择主/从模式时除外 (请参见TIM1_SMCR寄存器中MSM位的说明)。 010: 更新——选择更新事件作为触发输出 (TRGO)。例如, 主定时器可用作从定时器的预分频器 011: 比较脉冲——一旦发生输入捕获或比较匹配事件, 当CC1IF标志被置1时 (即使已为高), 触发输出都会发送一个正脉冲。(TRGO)

100: 比较——OC1REF信号用作触发输出 (TRGO)
 101: 比较——OC2REF信号用作触发输出 (TRGO)
 110: 比较——OC3REF信号用作触发输出 (TRGO)
 111: 比较——OC4REF信号用作触发输出 (TRGO)
注意: 必须先使能从定时器或ADC的时钟, 才能从主定时器接收事件; 并且从主定时器接收触发信号时, 不得实时更改从定时器或ADC的时钟。

3	CCDS	捕获/比较DMA选择 0: 发生CCx事件时发送CCxDMA请求 1: 发生更新事件时发送CCxDMA请求
2	CCUS	捕获/比较控制更新选择 0: 如果捕获/比较控制位进行预装载 (CCPC=1), 仅通过将COMG位置1来对这些位进行更新 1: 如果捕获/比较控制位进行预装载 (CCPC=1), 可通过将COMG位置1或TRGI的上升沿对这些位进行更新。 <i>注意: 此位仅对具有互补输出的通道有效。</i>
1	保留	必须保持复位值
0	CCPC	捕获/比较预装载控制 0: CCxE、CCxNE和OCxM位未进行预装载 1: CCxE、CCxNE和OCxM位进行了预装载, 写入这些位后, 仅当发生换向事件 (COM) (COMG位置1或在TRGI上检测到上升沿, 取决于CCUS位) 时才会对这些位进行更新。 <i>注意: 此位仅对具有互补输出的通道有效。</i>

20.4.3 TIM1 从模式控制寄存器 (TIM1_SMCR)

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	
Res.														TS[4:3]		Res.				SMS[3]											
rw		rw		rw		rw		rw		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	
ETP		ECE		ETPS[1:0]		ETF[3:0]				MSM		TS[2:0]		OCCS		SMS[2:0]															
rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31:22	保留	必须保持复位值
21:20	TS[4:3]	触发选择——位 4:3 请参考 TS 说明——位 2:0

19:17	保留	必须保持复位值
16	SMS[3]	从模式选择——位3 请参见SMS说明——位2:0
15	ETP	外部触发极性 此位可选择将ETR还是ETR用于触发操作 0: ETR未反相, 高电平或上升沿有效 1: ETR反相, 低电平或下降沿有效
14	ECE	外部时钟使能 此位可使能外部时钟模式2。 0: 禁止外部时钟模式2 1: 使能外部时钟模式2。计数器时钟由ETRF信号的任意有效边沿提供。 <i>注意: 将ECE位置1与选择外部时钟模式1并将TRGI连接到ETRF (SMS= “111” 且TS= “111”) 具有相同效果。 外部时钟模式2可以和以下从模式同时使用: 复位模式、门控模式和触发模式。不过此类情况下TRGI不得连接ETRF (TS位不得为“111”)。 如果同时使能外部时钟模式1和外部时钟模式2, 则外部时钟输入为ETRF。</i>
13:12	ETPS	外部触发预分频器 外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00: 预分频器关闭 01: 2 分频 ETRP 频率 10: 4 分频 ETRP 频率 11: 8 分频 ETRP 频率
11:8	ETF[3:0]	外部触发滤波器 此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成, 每 N 个连续事件才视为一个有效输出边沿: 0000: 无滤波器, 按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, $N=2$ 0010: $f_{SAMPLING}=f_{CK_INT}$, $N=4$ 0011: $f_{SAMPLING}=f_{CK_INT}$, $N=8$ 0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$

		1001: $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=8
		1010: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5
		1011: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6
		1100: $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8
		1101: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5
		1110: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6
		1111: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8
7	MSM	主/从模式 0: 不执行任何操作 1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟, 以使当前定时器与其从定时器实现完美同步 (通过 TRGO)。此设置适用于由单个外部事件对多个定时器进行同步的情况。
21,20,6,5,4	TS[4:0]	触发选择 此位域可选择将要用于同步计数器的触发输入。 00000: 内部触发 0 (ITR0) 00001: 内部触发 1 (ITR1) 00010: 内部触发 2 (ITR2) 00011: 内部触发 3 (ITR3) 00100: TI1 边沿检测器 (TI1F_ED) 00101: 滤波后的定时器输入 1 (TI1FP1) 00110: 滤波后的定时器输入 2 (TI1FP2) 00111: 外部触发输入 (ETRF) 其他: 保留 <i>注意: 有关各定时器 ITRx 含义的详细信息, 请参见表: TIM1 内部触发连接;</i> <i>这些位只能在未使用的情况下 (例如, SMS= “000” 时) 进行更改, 以避免转换时出现错误的边沿检测; 其他位位于同一寄存器的位置 16。</i>
3	OCCS	OCREF 清除信号选择 0: 清除信号连接到 SAC COMP1 或 COMP2 输出, 由 TIM1_OR1.OCREF_CLR 位决定。 1: 清除信号连接到 ETRF。
16,2,1,0	SMS[3:0]	从模式选择 选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关 (请参见输入控制寄存器和控制寄存器说明)。 0000: 禁止从模式——如果 CEN= “1”, 预分频器时钟直接由内部时钟提供。 0001: 编码器模式 1——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。 0010: 编码器模式 2——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

- 0011: 编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。
 - 0100: 复位模式——在出现所选触发输入（TRGI）上升沿时，重新初始化计数器并生成一个寄存器更新事件。
 - 0101: 门控模式——触发输入（TRGI）为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
 - 0110: 触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
 - 0111: 外部时钟模式 1——由所选触发信号（TRGI）的上升沿提供计数器时钟。
 - 1000: 组合复位+触发模式——在出现所选触发输入（TRGI）上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
- 注意：如果将 TI1F_ED 选作触发输入（TS=“100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平；必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

表 20-4 TIMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM2	TIM4	TIM3	保留
TIM3	TIM2	TIM1	TIM4	保留
TIM4	TIM1	TIM3	TIM2	保留
TIM2	TIM1	TIM3	TIM4	保留

20.4.4 TIM1 DMA/中断使能寄存器（TIM1_DIER）

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	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

位/位域	名称	描述
31:15	保留	必须保持复位值

14	TDE	触发DMA请求使能 0: 禁止触发DMA请求。 1: 使能触发DMA请求。
13	COMDE	COMDMA请求使能 0: 禁止COMDMA请求 1: 使能COMDMA请求
12	CC4DE	捕获/比较4DMA请求使能 0: 禁止CC4DMA请求。 1: 使能CC4DMA请求。
11	CC3DE	捕获/比较3DMA请求使能 0: 禁止CC3DMA请求。 1: 使能CC3DMA请求。
10	CC2DE	捕获/比较2DMA请求使能 0: 禁止CC2DMA请求。 1: 使能CC2DMA请求。
9	CC1DE	捕获/比较1DMA请求使能 0: 禁止CC1DMA请求。 1: 使能CC1DMA请求。
8	UDE	更新DMA请求使能 0: 禁止更新DMA请求。 1: 使能更新DMA请求。
7	BIE	断路中断使能 0: 禁止断路中断 1: 使能断路中断
6	TIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	COMIE	COM中断使能 0: 禁止COM中断 1: 使能COM中断
4	CC4IE	捕获/比较4中断使能 0: 禁止CC4中断。 1: 使能CC4中断。
3	CC3IE	捕获/比较3中断使能 0: 禁止CC3中断。

- 1: 使能CC3中断。
- 2 CC2IE 捕获/比较2中断使能
0: 禁止CC2中断。
1: 使能CC2中断。
- 1 CC1IE 捕获/比较1中断使能
0: 禁止CC1中断。
1: 使能CC1中断。
- 0 UIE 更新中断使能
0: 禁止更新中断。
1: 使能更新中断。

20.4.5 TIM1 状态寄存器 (TIM1_SR)

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
Res.														CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1IF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0													

位/位域	名称	描述
31:18	保留	必须保持复位值
17	CC6IF	比较6中断标志 请参见CC1IF说明 <i>注意: 通道6只能配置为输出</i>
16	CC5IF	比较5中断标志 请参见CC1IF说明 <i>注意: 通道5只能配置为输出</i>
15:14	保留	必须保持复位值
13	SBIF	系统断路中断标志 只要系统断路输入变为有效状态, 此标志便由硬件置1。系统断路输入无效后可通过软件对其清零。 此标志必须复位以使PWM重新开始工作。 0: 未发生断路事件。 1: 在系统断路输入上检测到有效电平。如果TIM1_DIER寄存器

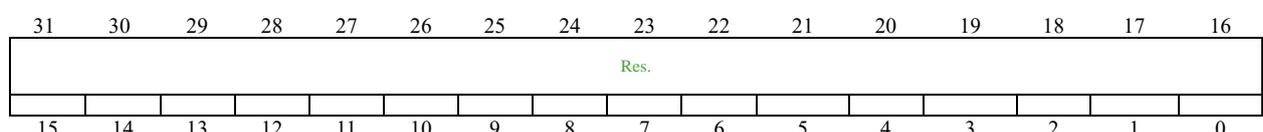
		中BIE=1，则会生成中断。
12	CC4OF	捕获/比较4重复捕获标志 请参见CC1OF说明
11	CC3OF	捕获/比较3重复捕获标志 请参见CC1OF说明
10	CC2OF	捕获/比较2重复捕获标志 请参见CC1OF说明
9	CC1OF	捕获/比较2重复捕获标志 仅当CCER.CCx E=1，且将相应通道配置为输入捕获模式时，此标志位才会由硬件置1。通过软件写入”0”可将该位清零。 0: 未检测到重复捕获 1: TIM1_CCR1寄存器中已捕获到计数器值且CC1IF标志已置1
8	B2IF	断路2中断标志 只要断路2输入变为有效状态，此标志便由硬件置1。断路2输入无效后可通过软件对其清零。 0: 未发生断路事件。 1: 在断路2输入上检测到有效电平。如果TIM1_DIER寄存器中BIE=1，则会生成中断。
7	BIF	断路中断标志 只要断路输入变为有效状态，此标志便由硬件置1。断路输入无效后可通过软件对其清零。 0: 未发生断路事件。 1: 在断路输入上检测到有效电平。如果TIM1_DIER寄存器中BIE=1，则会生成中断。
6	TIF	触发中断标志 在除门控模式以外的所有模式下，当使能从模式控制器后在TRGI输入上检测到有效边沿时，该标志将由硬件置1。选择门控模式时，该标志将在计数器启动或停止时置1。但需要通过软件清零。 0: 未发生触发事件 1: 触发中断挂起
5	COMIF	换向事件中断标志 此标志在发生换向事件时（捕获/比较控制位CCx E、CCx NE和OCx M已更新时）由硬件置1。但需要通过软件清零。 0: 未发生换向事件。 1: 换向中断挂起。

4	CC4IF	捕获/比较4中断标志 请参见CC1IF说明
3	CC3IF	捕获/比较3中断标志 请参见CC1IF说明
2	CC2IF	捕获/比较2中断标志 请参见CC1IF说明
1	CC1IF	捕获/比较1中断标志 如果通道CC1配置为输出： 当计数器与比较值匹配时，此标志由硬件置1，中心对齐模式下除外（请参见TIM1_CR1寄存器中的CMS位说明）。但需要通过软件清零。 0：不匹配 1：TIM1_CNT计数器的值与TIM1_CCR1寄存器的值匹配。当TIM1_CCR1的值大于TIM1_ARR的值时，CC1IF位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。 如果通道CC1配置为输入： 此位将在发生捕获事件时由硬件置1。通过软件或读取TIM1_CCR1寄存器将该位清零。 0：未发生输入捕获事件 1：TIM1_CCR1寄存器中已捕获到计数器值（IC1上已检测到与所选极性匹配的边沿）
0	UIF	更新中断标志 该位在发生更新事件时通过硬件置1。但需要通过软件清零。 0：未发生更新。 1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置1： <ul style="list-style-type: none"> - 上溢或下溢并且当TIM1_CR1寄存器中UDIS=0时； - TIM1_CR1寄存器中的URS=0且UDIS=0，并且由软件使用TIM1_EGR寄存器中的UG位重新初始化CNT时； - TIM1_CR1寄存器中的URS=0且UDIS=0，并且CNT由触发事件重新初始化时（参见<i>TIM1从模式控制寄存器 (TIM1_SMCR)</i>）。

20.4.6 TIM1 事件产生寄存器 (TIM1_EGR)

偏移地址：0x14

复位值：0x0000 0000



Res.							B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG	
							w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:9	保留	必须保持复位值
8	B2G	断路2生成 此位由软件置1以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 生成断路2事件。MOE位清零且B2IF标志置1。使能后可发生相关中断
7	BG	断路生成 此位由软件置1以生成事件，并由硬件自动清零。 0: 不执行任何操作。 1: 生成断路事件。MOE位清零且BIF标志置1。使能后可发生相关中断或DMA传输事件。
6	TG	触发生成 此位由软件置1以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: TIM1_SR寄存器中的TIF标志置1。使能后可发生相关中断或DMA传输事件
5	COMG	捕获/比较控制位的更新生成 该位可通过软件置1，并由硬件自动清零 0: 不执行任何操作 1: CCPC位置1时，可更新CCxE、CCxNE和OCxM位 <i>注意：此位仅对具有互补输出的通道有效</i>
4	CC4G	捕获/比较4生成 请参见CC1G说明
3	CC3G	捕获/比较3生成 请参见CC1G说明
2	CC2G	捕获/比较2生成 请参见CC1IF说明
1	CC1G	捕获/比较1生成 此位由软件置1以生成事件，并由硬件自动清零。 0: 不执行任何操作 1: 通道1上生成捕获/比较事件： 如果通道CC1配置为输出： 使能时，CC1IF标志置1并发送相应的中断或DMA请求。 如果通道CC1配置为输入：

TIM1_CCR1寄存器中将捕获到计数器当前值。使能时，CC1IF标志置1并发送相应的中断或DMA请求。如果CC1IF标志已为高电平，CC1OF标志将置1。

- 0 UG 更新生成
该位可通过软件置1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或DIR=0（递增计数），计数器将清零；如果DIR=1（递减计数），计数器将使用自动重载值（TIM1_ARR）

20.4.7 TIM1 捕获/比较模式寄存器 1_输出比较模式 (TIM1_CCMR1)

偏移地址：0x18

复位值：0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							OC2M[3]	Res.							OC1M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值
24	OC2M[3]	输出比较2模式——位3 请参见OC2M说明——位14:12
23:17	保留	必须保持复位值
16	OC1M[3]	输出比较1模式——位3 请参见OC1M说明——位6:4
15	OC2CE	输出比较2清零使能
14:12	OC2M[3:0]	输出比较2模式
11	OC2PE	输出比较2预装载使能
10	OC2FE	输出比较2快速使能
9:8	CC2S[1:0]	捕获/比较2选择 此位域定义通道方向（输入/输出）以及所使用的输入。

		00: CC2通道配置为输出
		01: CC2通道配置为输入, IC2映射到TI2上
		10: CC2通道配置为输入, IC2映射到TI1上
		11: CC2通道配置为输入, IC2映射到TRC上。此模式仅在通过 TS位 (TIM1_SMCR寄存器) 选择内部触发输入时有效
		<i>注意: 仅当通道关闭时(TIM1_CCER中的CC2E=0), 才可向CC2S 位写入数据。</i>
7	OC1CE	输出比较1清零使能
		0: OC1Ref不受ETRF输入影响
		1: ETRF输入上检测到高电平时, OC1Ref立即清零
16,6,5,4	OC1M[3] OC1M[2:0]	输出比较1模式
		这些位定义提供OC1和OC1N的输出参考信号OC1REF的行为。 OC1REF为高电平有效, 而OC1和OC1N的有效电平则取决于 CC1P位和CC1NP位。
		0000: 冻结——输出比较寄存器TIM_CCR1与计数器TIM_CNT 进行比较不会对输出造成任何影响。(该模式用于生成时 基)
		0001: 将通道1设置为匹配时输出有效电平。当计数器TIM_CNT 与捕获/比较寄存器1 (TIM_CCR1) 匹配时, OC1REF信号 强制变为有效电平
		0010: 将通道1设置为匹配时输出无效电平。当计数器TIM_CNT 与捕获/比较寄存器1 (TIM_CCR1) 匹配时, OC1REF信号 强制变为无效电平
		0011: 翻转——TIM_CNT=TIM_CCR1时, OC1REF发生翻转
		0100: 强制变为无效电平——OC1REF强制变为无效电平
		0101: 强制变为有效电平——OC1REF强制变为有效电平
		0110: PWM模式1——在递增计数模式下, 只要TIM_CNT <TIM_CCR1, 通道1便为有效状态, 否则为无效状态。在 递减计数模式下, 只要TIM_CNT>TIM_CCR1, 通道1便 为无效状态(OC1REF=0), 否则为有效状态(OC1REF=1)
		0111: PWM模式2——在递增计数模式下, 只要TIM_CNT <TIM_CCR1, 通道1便为无效状态, 否则为有效状态。在 递减计数模式下, 只要TIM_CNT>TIM_CCR1, 通道1便 为有效状态, 否则为无效状态。
		1000: 可再触发OPM模式1——在递增计数模式下, 通道为有效 状态, 直至(在TRGI信号上)检测到触发事件。然后, 在 PWM模式1下进行比较, 通道会在下一次更新时再次变为 有效状态。在递减计数模式下, 通道为无效状态, 直至(在 TRGI信号上)检测到触发事件。然后, 在PWM模式1下进 行比较, 通道会在下一次更时再次变为无效状态。
		1001: 可再触发OPM模式2——在递增计数模式下, 通道为无效 状态, 直至(在TRGI信号上)检测到触发事件。然后, 在

PWM模式2下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在TRGI信号上）检测到触发事件。然后，在PWM模式2下进行比较，通道会在下一次更新时再次变为有效状态

1010: 保留

1011: 保留

1100: 组合PWM模式1——OC1REF与在PWM模式1下的行为相同。OC1REFC是OC1REF和OC2REF的逻辑或运算结果

1101: 组合PWM模式2——OC1REF与在PWM模式2下的行为相同。OC1REFC是OC1REF和OC2REF的逻辑与运算结果。

1110: 不对称PWM模式1——OC1REF与在PWM模式1下的行为相同。计数器递增计数时，OC1REFC输出OC1REF；计数器递减计数时，OC1REFC输出OC2REF。

1111: 不对称PWM模式2——OC1REF与在PWM模式2下的行为相同。计数器递增计数时，OC1REFC输出OC1REF；计数器递减计数时，OC1REFC输出OC2REF

注意：在PWM模式1或PWM模式2下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF电平才会发生更改。

3 OC1PE

输出比较1预装载使能

0: 禁止与TIM1_CCR1相关的预装载寄存器。可随时向TIM1_CCR1写入数据，写入后将立即使用新值

1: 使能与TIM1_CCR1相关的预装载寄存器。可读/写访问预装载寄存器。TIM1_CCR1预装载值在每次生成更新事件时都会装载到活动寄存器中

注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用PWM模式（TIM1_CR1寄存器中的OPM位置1）。其它情况下则无法保证该行为。

2 OC1FE

输出比较1快速使能

此位用于加快触发输入事件对CC输出的影响

0: 即使触发开启，CC1也将根据计数器和CC1值正常工作。触发输入出现边沿时，激活CC1输出的最短延迟时间为5个时钟周期

1: 触发输入上出现有效边沿相当于CC1输出上的比较匹配。随后，无论比较结果如何，OC都设置为比较电平。采样触发输入和激活CC1输出的延迟时间缩短为3个时钟周期。仅当通道配置为PWM1或PWM2模式时，OCFE才会起作用

1:0 CC1S[1:0]

捕获/比较1选择

此位域定义通道方向（输入/输出）以及所使用的输入。

00: CC1通道配置为输出

01: CC1通道配置为输入，IC1映射到TI1上

- 10: CC1通道配置为输入, IC1映射到TI2上
 11: CC1通道配置为输入, IC1映射到TRC上。此模式仅在通过TS位 (TIM1_SMCR寄存器) 选择内部触发输入时有效
注意: 仅当通道关闭时(TIM1_CCER中的CC1E=0), 才可向CC1S位写入数据。

20.4.8 TIM1 捕获/比较模式寄存器 1_输入捕获模式 (TIM1_CCMR1)

偏移地址: 0x18

复位值: 0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC2F	输入捕获2滤波器
11:10	IC2PSC[1:0]	输入捕获2预分频器
9:8	CC2S[1:0]	捕获/比较2选择 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC2通道配置为输出。 01: CC2通道配置为输入, IC2映射到TI2上。 10: CC2通道配置为输入, IC2映射到TI1上。 11: CC2通道配置为输入, IC2映射到TRC上。此模式仅在通过TS位 (TIM1_SMCR寄存器) 选择内部触发输入时有效 <i>注意: 仅当通道关闭时(TIM1_CCER中的CC2E=0), 才可向CC2S位写入数据。</i>
7:4	IC1F[2:0]	输入捕获1滤波器 此位域可定义TI1输入的采样频率和适用于TI1的数字滤波器带宽。数字滤波器由事件计数器组成, 每N个连续事件才视为一个有效输出边沿: 0000: 无滤波器, 按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8

- 0110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$
- 0111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$
- 1000: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=6$
- 1001: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=8$
- 1010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=5$
- 1011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$
- 1100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$
- 1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$
- 1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$
- 1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$

3:2 IC1PSC[1:0]

输入捕获1预分频器

此位域定义CC1输入（IC1）的预分频比。

只要CC1E=0（TIM1_CCER寄存器），预分频器便立即复位。

- 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获
- 01: 每发生2个事件便执行一次捕获
- 10: 每发生4个事件便执行一次捕获
- 11: 每发生8个事件便执行一次捕获

1:0 CC1S [1:0]

捕获/比较1选择

此位域定义通道方向（输入/输出）以及所使用的输入。

- 00: CC1通道配置为输出
 - 01: CC1通道配置为输入，IC1映射到TI1上
 - 10: CC1通道配置为输入，IC1映射到TI2上
 - 11: CC1通道配置为输入，IC1映射到TRC上。此模式仅在通过TS位（TIM1_SMCR寄存器）选择内部触发输入时有效
- 注意: 仅当通道关闭时(TIM1_CCER中的CC1E=0), 才可向CC1S位写入数据。*

20.4.9 TIM1 捕获/比较模式寄存器 2_输出比较模式 (TIM1_CCMR2)

偏移地址: 0x1C

复位值: 0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							OC4M[3]	Res.							OC3M[3]
rw							rw							rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域

名称

描述

31:25 保留 必须保持复位值

24	OC4M[3]	输出比较4模式——位3 请参见OC4M说明——位14:12
23:17	保留	必须保持复位值
16	OC3M[3]	输出比较3模式——位3
15	OC4CE	输出比较4清零使能
14:12	OC4M[3:0]	输出比较4模式
11	OC4PE	输出比较4预装载使能
10	OC4FE	输出比较4快速使能
9:8	CC4S[1:0]	捕获/比较4选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC4通道配置为输出 01: CC4通道配置为输入，IC4映射到TI4上 10: CC4通道配置为输入，IC4映射到TI3上 11: CC4通道配置为输入，IC4映射到TRC上。此模式仅在通过 TS位（TIM1_SMCR寄存器）选择内部触发输入时有效 <i>注意: 仅当通道关闭时(TIM1_CCER中的CC4E=0), 才可向CC4S 位写入数据。</i>
7	OC3CE	输出比较3清零使能 0: OC3Ref不受ETRF输入影响 1: ETRF输入上检测到高电平时，OC3Ref立即清零
6:4	OC3M[2:0]	输出比较3模式
3	OC3PE	输出比较3预装载使能
2	OC3FE	输出比较3快速使能
1:0	CC3S[1:0]	捕获/比较3选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3通道配置为输出 01: CC3通道配置为输入，IC3映射到TI3上 10: CC3通道配置为输入，IC3映射到TI4上 11: CC3通道配置为输入，IC3映射到TRC上。此模式仅在通过 TS位（TIM1_SMCR寄存器）选择内部触发输入时有效 <i>注意: 仅当通道关闭时(TIM1_CCER中的CC3E=0), 才可向CC3S 位写入数据。</i>

20.4.10 TIM1 捕获/比较模式寄存器 2_输入捕获模式 (TIM1_CCMR2)

偏移地址: 0x1C

复位值: 0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	IC4F	输入捕获4滤波器
11:10	IC4PSC[1:0]	输入捕获4预分频器
9:8	CC4S[1:0]	捕获/比较4选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC4通道配置为输出。 01: CC4通道配置为输入, IC4映射到TI4上。 10: CC4通道配置为输入, IC4映射到TI3上。 11: CC4通道配置为输入, IC4映射到TRC上。此模式仅在通过 TS位 (TIM1_SMCR寄存器) 选择内部触发输入时有效 <i>注意: 仅当通道关闭时(TIM1_CCER中的CC4E=0), 才可向CC4S 位写入数据。</i>
7:4	IC3F[3:0]	输入捕获3滤波器 此位域可定义TI3输入的采样频率和适用于TI3的数字滤波器带宽。数字滤波器由事件计数器组成, 每N个连续事件才视为一个有效输出边沿: 0000: 无滤波器, 按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6

1100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$

1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$

1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$

1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$

3:2 IC3PSC[1:0]

输入捕获1预分频器

此位域定义CC3输入（IC3）的预分频比。

只要CC3E=0（TIM1_CCER寄存器），预分频器便立即复位。

00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获

01: 每发生2个事件便执行一次捕获

10: 每发生4个事件便执行一次捕获

11: 每发生8个事件便执行一次捕获

1:0 CC3S [1:0]

捕获/比较3选择

此位域定义通道方向（输入/输出）以及所使用的输入。

00: CC3通道配置为输出

01: CC3通道配置为输入，IC3映射到TI3上

10: CC3通道配置为输入，IC3映射到TI4上

11: CC3通道配置为输入，IC3映射到TRC上。此模式仅在通过TS位（TIM1_SMCR寄存器）选择内部触发输入时有效

注意：仅当通道关闭时(TIM1_CCER中的CC3E=0)，才可向CC3S位写入数据。

20.4.11 TIM1 捕获/比较使能寄存器 1 (TIM1_CCER)

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										CC6P	CC6E	Res.		CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	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	rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	CC6P	捕获/比较6输出极性 请参见CC1P说明
20	CC6E	捕获/比较6输出使能 请参见CC1E说明
19:18	保留	必须保持复位值

17	CC5P	捕获/比较5输出极性 请参见CC1P说明
16	CC5E	捕获/比较5输出使能 请参见CC1E说明
15	CC4NP	捕获/比较4互补输出极性 请参见CC1NP说明
14	保留	必须保持复位值
13	CC4P	捕获/比较4输出极性 请参见CC1P说明
12	CC4E	捕获/比较4输出使能 请参见CC1E说明
11	CC3NP	捕获/比较3互补输出极性 请参见CC1NP说明
10	CC3NE	捕获/比较3互补输出使能 请参见CC1NE说明
9	CC3P	捕获/比较3输出极性 请参见CC1P说明
8	CC3E	捕获/比较3输出使能 请参见CC1E说明
7	CC2NP	捕获/比较2互补输出极性 请参见CC1NP说明
6	CC2NE	捕获/比较2互补输出使能 请参见CC1NE说明
5	CC2P	捕获/比较2输出极性 请参见CC1P说明
4	CC2E	捕获/比较2输出使能 请参见CC1E说明
3	CC1NP	捕获/比较1互补输出极性 CC1通道配置为输出: 0: OC1N高电平有效 1: OC1N低电平有效 CC1通道配置为输入:

此位与CC1P配合使用，用以定义TI1FP1/TI2FP1的极性。请参见CC1P说明。

2	CC1NE	<p>捕获/比较1互补输出使能</p> <p>0: 关闭——OC1N未激活。OC1N电平是MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的函数。</p> <p>1: 开启——在相应输出引脚上输出OC1N信号，具体取决于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位。</p> <p><i>注意：此位将在具有互补输出的通道上进行预装载。如果TIM1_CR2寄存器中的CCPC位置1，则仅当生成换向事件时，CC1NE有效位才会从预装载位获取新值。</i></p>
1	CC1P	<p>捕获/比较1输出极性</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>CC1NP/CC1P位可针对触发或捕获操作选择TI1FP1和TI2FP1的极性。</p> <p>00: 未反相/上升沿触发 电路对TIxFP1上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1未反相（在门控模式或编码器模式下执行触发操作）。</p> <p>01: 反相/下降沿触发 电路对TIxFP1下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1反相（在门控模式或编码器模式下执行触发操作）。</p> <p>10: 保留，不使用此配置。</p> <p>11: 未反相/上升沿和下降沿均触发 电路对TIxFP1上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。</p>
0	CC1E	<p>捕获/比较1输出使能</p> <p>CC1通道配置为输出：</p> <p>0: 关闭——OC1未激活</p> <p>1: 开启——在相应输出引脚上输出OC1信号</p> <p>CC1通道配置为输入：</p> <p>此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器1（TIM1_CCR1）中。</p> <p>0: 禁止捕获</p> <p>1: 使能捕获</p>

表 20-5 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态 ⁽¹⁾		
MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态	
1	×	×	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0		
			0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
			0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
			×	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
			1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
			1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态（输出使能为无效状态） OCxN=CCxNP
0	0 1	×	×	×	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。		
			0	0	关闭状态（输出使能为无效状态）		
			0	1	异步：OCx=CCxP、OCxN=CCxNP（如果触发 BRK 或 BRK2）。		
			1	0	随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。		
			1	1	注意：BRK2 只能在 OSSI = OSSR = 1 时使用。		

20.4.12 TIM1 计数值寄存器 (TIM1_CNT)

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIFCPY		Res.													
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UIFCPY	UIF副本 该位是TIM1_SR寄存器中UIF位的只读副本。如果TIM1_CR1中

的UIFREMAP位复位，则位31保留，读为0。

30:16	保留	必须保持复位值
15:0	CNT[15:0]	计数器值

20.4.13 TIM1 预分频器寄存器 (TIM1_PSC)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK_PSC} / (PSC[15:0] + 1)$ 。 PSC 包含每次发生更新事件 (包括计数器通过 TIM1_EGR 寄存器中的 UG 位清零时, 或在配置为“复位模式”时通过触发控制器清零时) 时要装载到有效预分频器寄存器的值。

20.4.14 TIM1 自动重载值寄存器 (TIM1_ARR)

偏移地址: 0x2C

复位值: 0x0000 FFFF

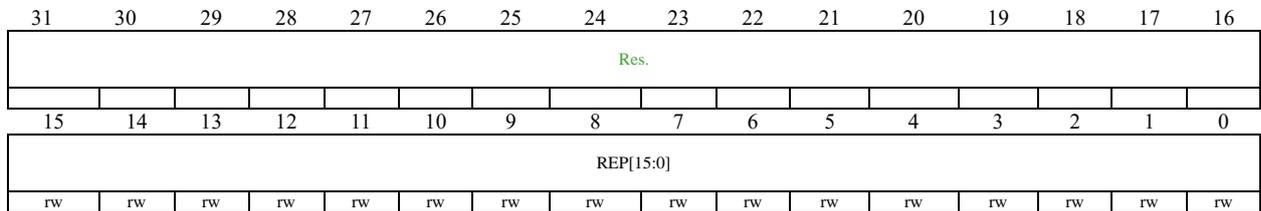
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ARR[15:0]	自动重载值 ARR为要装载到实际自动重载寄存器的值。 有关ARR更新和行为的更多详细信息, 请参见: 时基单元 。 当自动重载值为空时, 计数器不工作。

20.4.15 TIM1 重复计数器寄存器 (TIM1_RCR)

偏移地址: 0x30

复位值: 0x0000 0000

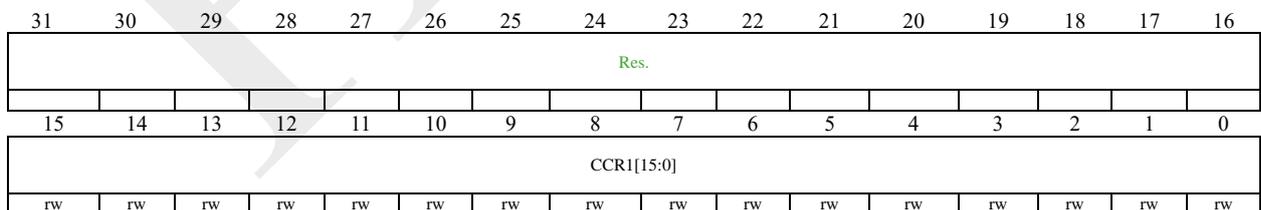


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	REP[15:0]	重复计数器值 使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向有效寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 与REP_CNT相关的减计数器每次计数到0时，都将生成一个更新事件并且计数器从REP值重新开始计数。由于只有生成重复更新事件U_RC时，REP_CNT才会重载REP值，因此在生成下一重复更新事件之前，无论向TIM1_RCR寄存器写入何值都无影响。 这意味着PWM模式下（REP+1）相当于： 边沿对齐模式下的PWM周期数。 中心对齐模式下的PWM半周期数。

20.4.16 TIM1 捕获/比较寄存器 1 (TIM1_CCR1)

偏移地址: 0x34

复位值: 0x0000 0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR1[15:0]	捕获/比较1值 如果通道CC1配置为输出： CC1为要装载到有效捕获/比较1寄存器的值（预装载值）。如果没有通过TIM1_CCMR1寄存器中的OC1PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有

效的捕获/比较寄存器1)。有效捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC1输出上发出信号的值。

如果通道CC1配置为输入：

CR1为上一个输入捕获1事件（IC1）发生时的计数器值。只能读取TIM1_CCR1寄存器，无法对其进行编程。

当配置为霍尔传感器接口时，该位数值不可读取。

20.4.17 TIM1 捕获/比较寄存器 2 (TIM1_CCR2)

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR2[15:0]	捕获/比较2值的低16位 如果通道CC2配置为输出： CC2是捕获/比较寄存器2的预装载值。 如果没有通过TIM1_CCMR2寄存器中的OC2PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器2）。 实际捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC2输出上发出信号的值。 如果通道CC2配置为输入： CC2为上一个输入捕获2事件（IC2）发生时的计数器值。只能读取TIM1_CCR2寄存器，无法对其进行编程。 当配置为霍尔传感器接口时，该位数值不可读取。

20.4.18 TIM1 捕获/比较寄存器 3 (TIM1_CCR3)

偏移地址：0x3C

复位值：0x0000 0000

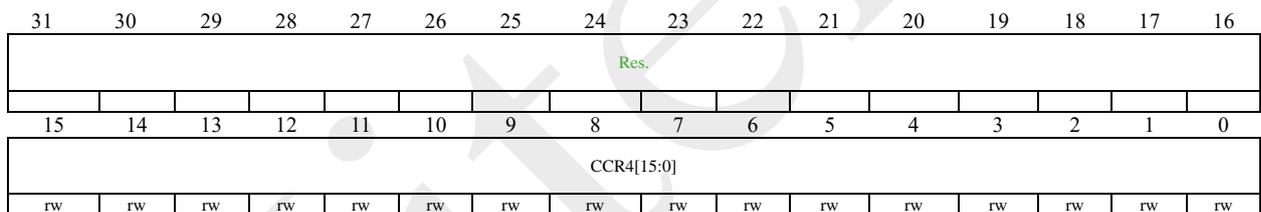
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR3[15:0]	<p>捕获/比较3值的低16位</p> <p>如果通道CC3配置为输出:</p> <p>CC3是捕获/比较寄存器3的预装载值。如果没有通过TIM1_CCMR3寄存器中的OC3PE位来使能预装载功能,则该值立刻生效;否则只在发生更新事件时生效(拷贝到实际起作用的捕获/比较寄存器3)。</p> <p>实际捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC3输出上发出信号的值。</p> <p>如果通道CC3配置为输入:</p> <p>CC3为上一个输入捕获3事件(IC3)发生时的计数器值。只能读取TIM1_CCR3寄存器,无法对其进行编程。</p> <p>当配置为霍尔传感器接口时,该位数值不可读取。</p>

20.4.19 TIM1 捕获/比较寄存器 4 (TIM1_CCR4)

偏移地址: 0x40

复位值: 0x0000 0000



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR4[15:0]	<p>捕获/比较4值的低16位</p> <p>如果通道CC4配置为输出:</p> <p>CC4是捕获/比较寄存器4的预装载值。如果没有通过TIM1_CCMR4寄存器中的OC4PE位来使能预装载功能,则该值立刻生效;否则只在发生更新事件时生效(拷贝到实际起作用的捕获/比较寄存器4)。</p> <p>实际捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC4输出上发出信号的值。</p> <p>如果通道CC3配置为输入:</p> <p>CC4为上一个输入捕获4事件(IC4)发生时的计数器值。只能读取TIM1_CCR4寄存器,无法对其进行编程。</p>

20.4.20 TIM1 断路和死区寄存器 (TIM1_BDTR)

偏移地址: 0x44

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值
29	BK2BID	断路2双向 请参见BKBID说明
28	BKBID	断路双向 0: 断路输入BRK为输入模式 1: 断路输入BRK为双向模式 在双向模式下（BKBID位置1），断路输入配置为输入模式和开漏输出模式。任何激活的断路事件都将使断路输入上呈逻辑低电平，以向外部器件指示发生了内部断路事件。 <i>注意：只要编程了LOCK（TIM1_BDTR寄存器中的LOCK位）级别1，此位即无法修改。</i> <i>注意：对该位执行任何写操作后，都需要经过1个APB时钟周期的延迟才生效。</i>
27	BK2DSRM	断路2解除 请参见BKDSRM说明。
26	BKDSRM	断路解除 0: 启动断路输入BRK 1: 解除断路输入BRK 当没有断路源被激活时，此位由硬件清零。必须通过软件将BKDSRM位置1以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。 <i>注意：对该位执行任何写操作后，都需要经过1个APB时钟周期的延迟才生效。</i>
25	BK2P	断路2极性 0: 断路输入BRK2为低电平有效 1: 断路输入BRK2为高电平有效 <i>注意：只要编程了LOCK（TIM1_BDTR寄存器中的LOCK位）级别1，此位即无法修改。</i> <i>对该位执行任何写操作后，都需要经过1个APB时钟周期的延迟才生效。</i>
24	BK2E	断路2使能 该位可使能完整的断路2保护（如 图：断路和断路2电路概述 部分所示）。 0: 禁止断路2功能

1: 使能断路2功能

注意: BRKIN2 必须只在 OSSR=OSSI=1 时使用。

编程了 LOCK (TIM1_BDTR 寄存器中的 LOCK 位) 级别后, 此位即无法修改。

对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。

23:20
BK2F[3:0]
断路2滤波器

此位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件

计数器组成, 每 N 个连续事件才视为一个有效输出边沿:

0000: 无滤波器, BRK2 异步工作

0001: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=2

0010: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=4

0011: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=8

0100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, N=6

0101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, N=8

0110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, N=6

0111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, N=8

1000: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, N=6

1001: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, N=8

1010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=5

1011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=6

1100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=8

1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=5

1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=6

1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=8

注意: 编程了 LOCK (TIM1_BDTR 寄存器中的 LOCK 位) 级别后, 此位即无法修改。

19:16
BKF[3:0]
断路滤波器 (Breakfilter)

此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成, 每 N 个连续事件才视为一个有效输出边沿:

0000: 无滤波器, BRK 异步工作

0001: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=2

0010: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=4

0011: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, N=8

0100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, N=6

0101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, N=8

0110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, N=6

0111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, N=8

1000: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, N=6

1001: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, N=8

1010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=5

1011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=6

1100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, N=8

1101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=5

1110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=6

1111: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, N=8

注意: 编程了LOCK (TIM1_BDTR 寄存器中的LOCK位) 级别1后, 此位即无法修改。

15	MOE	<p>主输出使能</p> <p>只要断路输入 (BRK或BRK2) 为有效状态, 此位便由硬件异步清零。此位由软件置1, 也可根据AOE位状态自动置1。此位仅对配置为输出的通道有效。</p> <p>0: 响应断路事件 (2个)。禁止OC和OCN输出响应断路事件或向MOE写入0时: OC和OCN输出被禁止或被强制为空闲状态, 具体取决于OSSI位。</p> <p>1: 如果OC和OCN输出的相应使能位 (TIM1_CCER寄存器中的CCxE和CCxNE位) 均置1, 则使能OC和OCN输出。</p> <p>有关详细信息, 请参见OC/OCN使能说明 (TIM1捕获/比较使能寄存器 (TIM1_CCER))。</p>
14	AOE	<p>自动输出使能</p> <p>0: 控制MOE位只能由软件置1</p> <p>1: MOE位可由软件置1, 也可在发生下一更新事件时自动置1 (如果断路输入BRK和BRK2均无效)</p> <p><i>注意: 只要编程了LOCK (TIM1_BDTR 寄存器中的LOCK位) 级别1, 此位即无法修改。</i></p>
13	BKP	<p>断路极性</p> <p>0: 断路输入BRK为低电平有效</p> <p>1: 断路输入BRK为高电平有效</p> <p><i>注意: 只要编程了LOCK (TIM1_BDTR 寄存器中的LOCK位) 级别1, 此位即无法修改。</i></p> <p><i>对该位执行任何写操作后, 都需要经过1个APB时钟周期的延迟才生效。</i></p>
12	BKE	<p>断路使能</p> <p>该位可使能完整的断路保护 (如图: 断路和断路2 电路概述部分所示)。</p> <p>0: 禁止断路功能</p> <p>1: 使能断路功能</p> <p><i>注意: 编程了LOCK (TIM1_BDTR 寄存器中的LOCK位) 级别1后, 此位即无法修改。</i></p> <p><i>对该位执行任何写操作后, 都需要经过1个APB时钟周期的延迟才生效。</i></p>
11	OSSR	<p>运行模式下的关闭状态选择</p> <p>此位在MOE=1时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出, 则不存在OSSR。</p> <p>0: 处于无效状态时, 禁止OC/OCN输出 (定时器释放输出控制, 由强制高阻态的GPIO逻辑接管)。</p> <p>1: 处于无效状态时, 一旦CCxE=1或CCxNE=1, 便使能OC/OCN输出并将其设为无效电平 (输出仍由定时器控制)。</p> <p><i>注意: 编程了LOCK (TIM1_BDTR 寄存器中的LOCK位) 级别2后, 此位即无法修改。</i></p>

- 10 OSSI 空闲模式下的关闭状态选择
当由于断路事件或软件写操作而使MOE=0时，此位作用于配置为输出的通道。
0: 处于无效状态时，禁止OC/OCN输出（定时器释放输出控制，由强制高阻态的GPIO逻辑接管）。
1: 处于无效状态时，首先将OC/OCN输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。
注意：编程了LOCK (TIM1_BDTR寄存器中的LOCK位)级别2后，此位即无法修改。
- 9:8 LOCK[1:0] 锁定配置
这些位用于针对软件错误提供写保护。
00: 关闭锁定——不对任何位提供写保护。
01: 锁定级别1，此时无法对TIM1_BDTR寄存器中的DTG位、TIM1_CR2寄存器中的OISx和OISxN位以及TIM1_BDTR寄存器中的BK2BID、BKBID、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR和DTG[7:0]位执行写操作。
10: 锁定级别2，此时无法对锁定级别1中适用的各位、CC极性位（TIM1_CCER寄存器中的CCxP/CCxNP位，只要通过CCxS位将相关通道配置为输出）以及OSSR和OSSI位执行写操作。
11: 锁定级别3，此时无法对锁定级别2中适用的各位、CC控制位（TIM1_CCMRx寄存器中的OCxM和OCxPE位，只要通过CCxS位将相关通道配置为输出）执行写操作。
注意：复位后只能对LOCK位执行一次写操作。对TIM1_BDTR寄存器执行写操作后其中的内容将冻结，直到下一次复位。
- 7:0 DTG[7:0] 配置死区发生器
此位域定义插入到互补输出之间的死区持续时间。DT与该持续时间相对应。
DTG[7:5]=0xx=>DT=DTG[7:0] × t_{DTG}，其中t_{DTG}=t_{DTs}。
DTG[7:5]=10x=>DT= (64+DTG[5:0]) × t_{DTG}，其中t_{DTG}=2 × t_{DTs}。
DTG[7:5]=110=>DT= (32+DTG[4:0]) × t_{DTG}，其中t_{DTG}=8 × t_{DTs}。
DTG[7:5]=111=>DT= (32+DTG[4:0]) × t_{DTG}，其中t_{DTG}=16 × t_{DTs}。
示例：如果t_{DTG}=125ns（8MHz），则可能的死区值为：
0到15875ns（步长为125ns），
16μs到31750ns（步长为250ns），
32μs到63μs（步长为1μs），
64μs到126μs（步长为2μs）
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位)级别1、2或3，此位域即无法修改。

20.4.21 TIM1 DMA 控制寄存器 (TIM1_DCR)

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DBL[4:0]				Res.				DBA[4:0]			
				rw								rw			

位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	DBL[4:0]	DMA连续传送长度 该5位向量定义了DMA的传送次数（当对TIM1_DMAR寄存器进行读或写时，定时器进行一次连续传送）。 00000：1次传送， 00001：2次传送， 00010：3次传送， ... 10001：18次传送。
7:5	保留	必须保持复位值
4:0	DBA [4:0]	DMA基址 该5位向量定义DMA传输的基址（通过TIM1_DMAR地址进行读/写访问时）。DBA定义为从TIM1_CR1寄存器地址开始计算的偏移量。 示例： 00000：TIM1_CR1， 00001：TIM1_CR2， 00010：TIM1_SMCR， ... 示例：以下面的传送为例：DBL=7次传送且DBA=TIM1_CR1。这种情况下将向/从自TIM1_CR1地址开始的7个寄存器传输数据。

20.4.22 TIM1 DMA 全传输地址寄存器（TIM1_DMAR）

偏移地址：0x4C

复位值：0x0000 0000

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

位/位域	名称	描述
31:0	DMAB [31:0]	DMA 连续传送寄存器 对 DMA 寄存器执行读或写操作将访问位于如下地址的寄存器： (TIM1_CR1 地址) + (DBA+DMA 索引) × 4 其中 TIM1_CR1 地址为控制寄存器 1 的地址, DBA 为 TIM1_DCR 寄存器中配置的 DMA 基址, DMA 索引由 DMA 传输自动控制, 其范围介于 0 到 DBL (TIM1_DCR 寄存器中配置的 DBL) 之间。

20.4.23 TIM1 配置寄存器 (TIM1_OR1)

偏移地址: 0x50

复位值: 0x0000 0000

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

位/位域	名称	描述
31:1	保留	必须保持复位值
31:0	OCREF_CLR	Ocref_clr 源选择 0: 比较器 1 输出作为 Ocref_clr 的输入 0: 比较器 2 输出作为 Ocref_clr 的输入

20.4.24 TIM1 捕获/比较模式寄存器 3 (TIM1_CCMR3)

偏移地址: 0x54

复位值: 0x0000 0000

输出比较模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								OC6M[3]	Res.						OC5M[3]
								rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6CE	OC6M[2:0]			OC6PE	OC6FE	Res.		OC5CE	OC5M[2:0]			OC5PE	OC5FE	Res.	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:25	保留	必须保持复位值
23:17	保留	必须保持复位值

15	OC6CE	输出比较6清零使能
24, 14:12	OC6M[3:0]	输出比较6模式
11	OC6PE	输出比较2预装载使能
10	OC6FE	输出比较6快速使能
9:8	保留	必须保持复位值
7	OC5CE	输出比较5清零使能
16,6,5,4	OC5M[3:0]	输出比较5模式
3	OC5PE	输出比较 1 预装载使能
2	OC5FE	输出比较1快速使能
1:0	保留	必须保持复位值

20.4.25 TIM1 捕获/比较寄存器 5 (TIM1_CCR5)

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.												
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	GC5C3	通道5和通道3组 通道3输出上失真： 0：OC5REF对OC3REFC无影响 1：OC3REFC是OC3REFC和OC5REF的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在TIM1CCMR2中选择了预装载功能）。 <i>注意：也可在组合PWM信号上应用此失真。</i>
30	GC5C2	通道5和通道2组 通道2输出上失真： 0：OC5REF对OC2REFC无影响 1：OC2REFC是OC2REFC和OC5REF的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在TIM1CCMR1中选择了预装载功能）。 <i>注意：也可在组合PWM信号上应用此失真。</i>

29	GC5C1	<p>通道5和通道1组 通道1输出上失真： 0：OC5REF对OC1REFC无影响 1：OC1REFC是OC1REFC和OC5REF的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在TIM1CCMR1中选择了预装载功能）。 <i>注意：也可在组合PWM信号上应用此失真。</i></p>
28:16	保留	必须保持复位值
15:0	CCR5[15:0]	<p>捕获/比较5值 CCR5是捕获/比较寄存器5的预装载值。 如果没有通过TIM1_CCMR3寄存器中的OC5PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器5）。 有效捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC5输出上发出信号的值。</p>

20.4.26 TIM1 捕获/比较寄存器 6 (TIM1_CCR6)

偏移地址：0x5C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CCR6[15:0]	<p>捕获/比较6值 CCR6是捕获/比较寄存器6的预装载值。 如果没有通过TIM1_CCMR3寄存器中的OC6PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器6）。 有效捕获/比较寄存器中包含要与计数器TIM1_CNT进行比较并在OC6输出上发出信号的值。</p>

20.4.27 TIM1 轮换功能寄存器 1 (TIM1_AF1)

偏移地址: 0x60

复位值: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.		BKCOMP2P	BKCOMP1P	BKINP	Res.						BKCOMP2E	BKCOMP1E	BKINE
rw	rw			rw	rw	rw							rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值
17:14	ETRSEL [3]	ETR源选择信号 0000: GPIO连接到ETR输入 0001: COMP1输出连接到ETR输入 0010: COMP2输出连接到ETR输入 0011: ADCAWD1 0110: COMP3输出连接到ETR输入 0111: COMP4输出连接到ETR输入 其他值: 保留 注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1, 这些位即无法修改。
13:12	保留	必须保持复位值
11	BKCOMP2P	BRKCOMP2输入极性 此位选择COMP2输入灵敏度, 必须与BKP极性位一起编程。 0: COMP2 输入为高电平有效 1: COMP2输入为低电平有效 注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1, 此位即无法修改。
10	BKCOMP1P	BRKCOMP1输入极性 此位选择COMP1输入灵敏度, 必须与BKP极性位一起编程。 0: COMP1输入为高电平有效 1: COMP1输入为低电平有效 注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1, 此位即无法修改。
9	BKINP	BRKBKIN输入极性 此位选择BKIN复用功能输入灵敏度, 必须与BKP极性位一起编程。 0: BKIN输入为高电平有效 1: BKIN输入为低电平有效 注意: 只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1, 此位即无法修改。

别1，此位即无法修改。

8:3	保留	必须保持复位值
2	BKCMP2E	BRKCOMP2使能 此位使能定时器BRK输入的COMP2。COMP2输出与其他BRK源进行“或”运算。 0: 禁止COMP2输入 1: 使能COMP2输入 <i>注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。</i>
1	BKCMP1E	BRKCOMP1使能 此位使能定时器BRK输入的COMP1。COMP1输出与其他BRK源进行“或”运算。 0: 禁止COMP1输入 1: 使能COMP1输入 <i>注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。</i>
0	BKINE	BRKBKIN输入使能 此位使能定时器BRK输入的BKIN复用功能。BKIN输入与其他BRK源进行“或”运算。 0: 禁止BKIN输入 1: 使能BKIN输入 <i>注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。</i>

20.4.28 TIM1 轮换功能寄存器 2 (TIM1_AF2)

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				BK2CMP2P	BK2CMP1P	BK2INP	Res.						BK2CMP2E	BK2CMP1E	BK2INE
				rw	rw	rw							rw	rw	rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11	BK2CMP2P	BRK2COMP2输入极性 此位选择COMP2输入灵敏度，必须与BK2P极性位一起编程。 0: COMP2 输入为高电平有效 1: COMP2输入为低电平有效 <i>注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。</i>

- 10 BK2CMP1P BRK2COMP1输入极性
此位选择COMP1输入灵敏度，必须与BK2P极性位一起编程。
0: COMP1输入为高电平有效
1: COMP1输入为低电平有效
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。
- 9 BK2INP BRK2BKIN输入极性
此位选择BK2IN复用功能输入灵敏度，必须与BK2P极性位一起编程。
0: BKIN输入为高电平有效
1: BKIN输入为低电平有效
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。
- 8:3 保留 必须保持复位值
- 2 BK2CMP2E BRK2COMP2使能
此位使能定时器BRK2输入的COMP2。COMP2输出与其他BRK2源进行“或”运算。
0: 禁止COMP2输入
1: 使能COMP2输入
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。
- 1 BK2CMP1E BRK2COMP1使能
此位使能定时器BRK2输入的COMP1。COMP1输出与其他BRK2源进行“或”运算。
0: 禁止COMP1输入
1: 使能COMP1输入
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。
- 0 BK2INE BRK2BKIN输入使能
此位使能定时器BRK2输入的BK2IN复用功能。BK2IN输入与其他BRK源进行“或”运算。
0: 禁止BK2IN输入
1: 使能BK2IN输入
注意：只要编程了LOCK (TIM1_BDTR寄存器中的LOCK位) 级别1，此位即无法修改。

20.4.29 TIM1 定时器输入选择寄存器 (TIM1_TISEL)

偏移地址: 0x68

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				TI4SEL[3:0]				Res.				TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				TI2SEL[3:0]				Res.				TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27:24	TI4SEL[3:0]	选择 TI4[0]到 TI4[15]作为输入 0000: TIM1_CH4 作为输入 0001: COMP2 输出 其他: 保留
23:20	保留	必须保持复位值
19:16	TI3SEL[3:0]	选择 TI3[0]到 TI3[15]输入 0000: TIM1_CH3 输入 0001: COMP1 输出 其他: 保留
15:12	保留	必须保持复位值
11:8	TI2SEL[3:0]	选择 TI2[0]到 TI2[15]输入 0000: TIM1_CH2 输入 0001: COMP2 输出 0010: COMP4 输出 其他: 保留
7:4	保留	必须保持复位值
3:0	TI1SEL[3:0]	选择 TI1[0]到 TI1[15]输入 0000: TIM1_CH1 输入 0001: COMP1 输出 0010: COMP3 输出 其他: 保留

21 通用定时器 (TIM2/3/4)

21.1 TIM2/3/4 简介

通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获）或生成输出波形（输出比较和 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

这些定时器彼此完全独立，不共享任何资源。

21.2 TIM2/3/4 主要特性

本模块主要功能特性如下：

- 16 位递增、递减、递增/递减自动重载计数器；
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间；
- 4 个独立通道（文中进行通道功能描述时会使用 x 表示 4 个通道中的任何一个）可用于：
 - 输入捕获；
 - 输出比较；
 - PWM 生成（边沿和中心对齐模式）；
 - 单脉冲模式输出；
- 触发输入用作外部时钟或逐周期电流管理；
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路；
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）；
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）；
 - 输入捕获；
 - 输出比较；

与 CPU 接口：

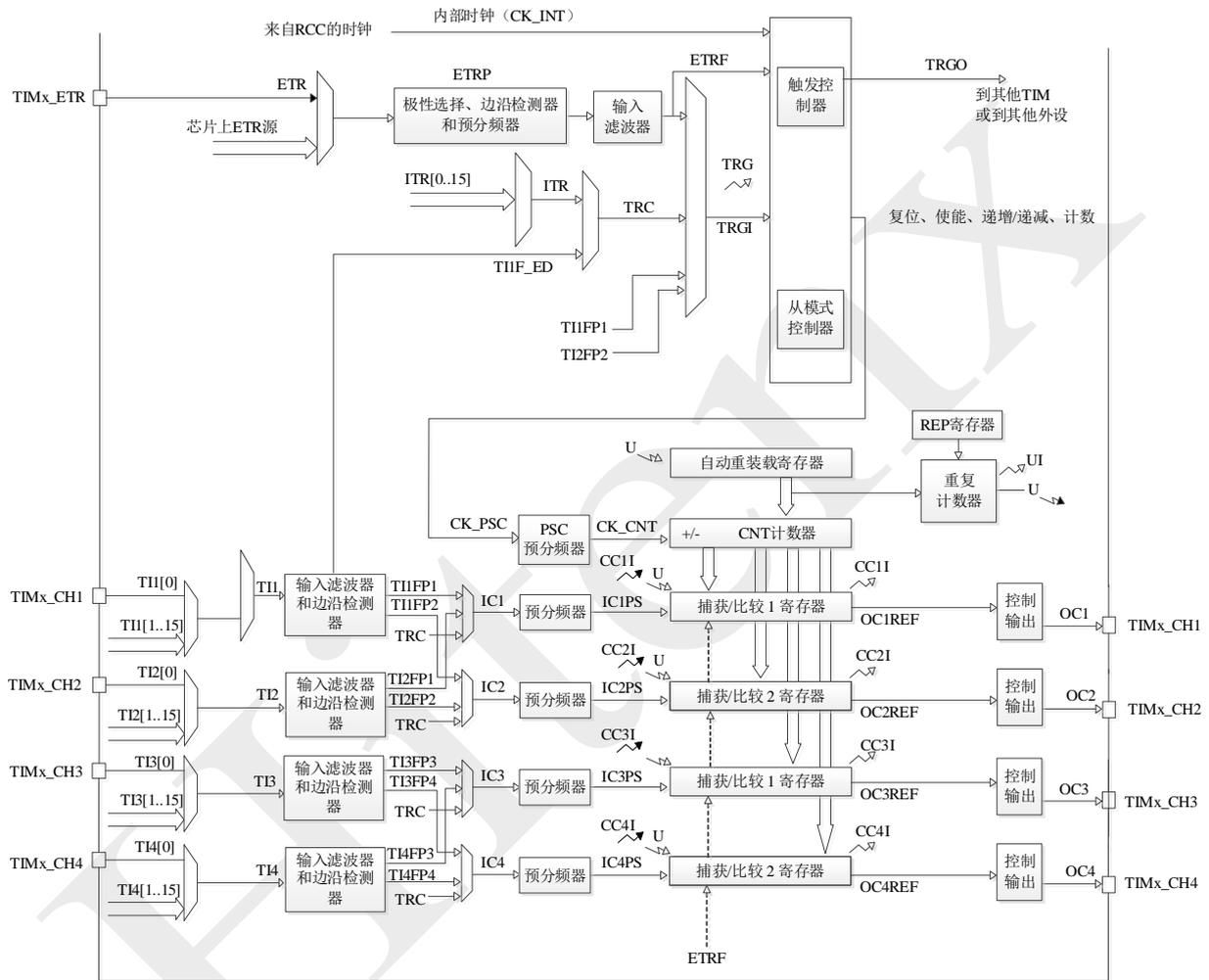
- 挂接在 APB 总线上，只支持 32 位访问

- 访问模块内部未定义寄存器空间时，读数据为 0，写数据无效

时钟与复位：

- 双时钟域
- 异步复位，低电平有效

图 21-1 TIM2/3/4 通用控制定时器框图



注意：

寄存器 根据控制位，在发生U事件后，预装载寄存器内的数据转移到有效寄存器中

→ 事件

↔

21.3 TIM2/3/4 功能说明

21.3.1 时基单元

可编程定时器的主要模块由一个 16 位及其相关的自动重载寄存器组成。计数器可递增计数、递减计数或同时递增和递减计数。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIM2/3/4_CNT)
- 预分频器寄存器 (TIM2/3/4_PSC)
- 自动重载寄存器 (TIM2/3/4_ARR)
- 重复寄存器 (TIM2/3/4_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。

预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIM2/3/4_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值 (或者在递减计数时达到下溢值) 并且 TIM2/3/4_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

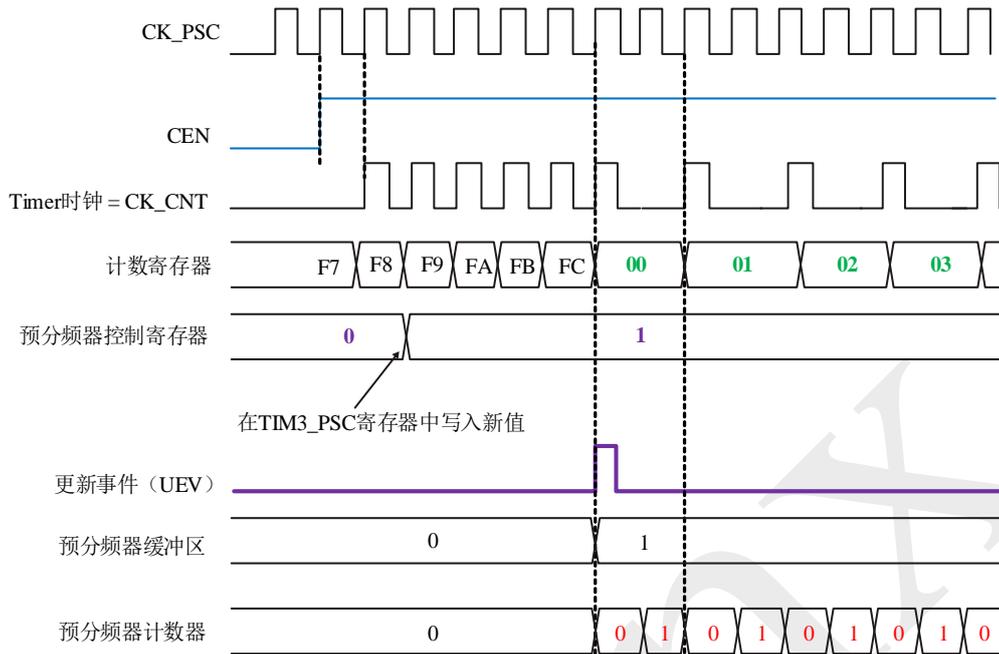
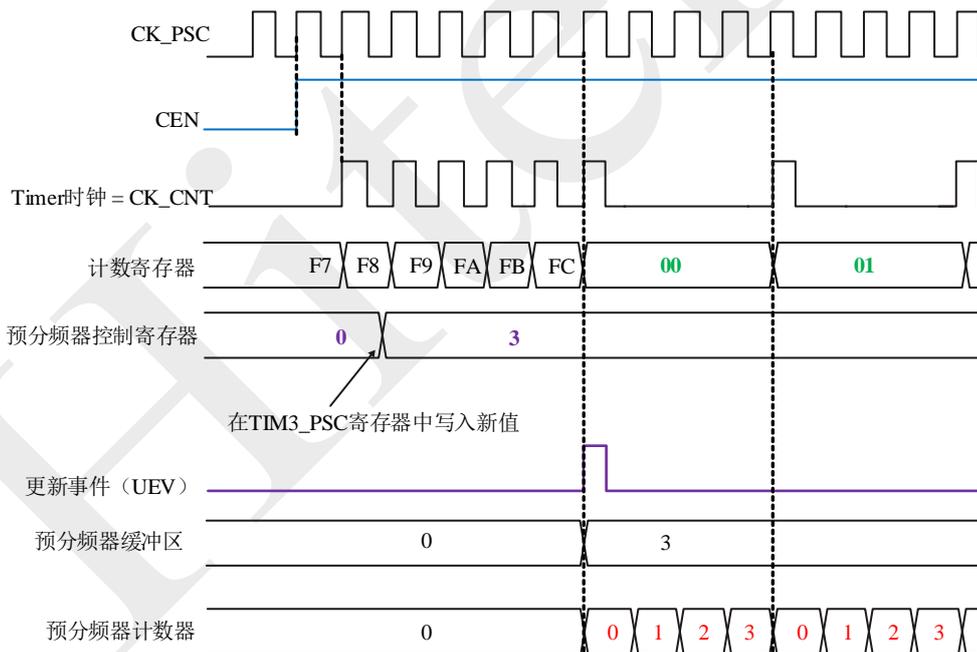
计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM2/3/4_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器 (有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明)。

注意： 计数器将在 TIM2/3/4_CR1 寄存器的 CEN 位置 1 时刻的 1 个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位寄存器 (TIM2/3/4_PSC 寄存器) 所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

下图以一些示例说明在预分频比实时变化时计数器的行为：

图 21-2 预分频器分频由 1 变为 2 计数器时序图

图 21-3 预分频器分频由 1 变为 4 计数器时序图


21.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIM2/3/4_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中设定的次数加一次（(TIM2/3/4_RCR) + 1）后，将生成更新事件（UEV）。否则，将

在每次计数器上溢时产生更新事件。

通过软件将 TIM2/3/4_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。

此外，如果 TIM2/3/4_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM2/3/4_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIM2/3/4_RCR 寄存器的内容。
- 使用预装载值（TIM2/3/4_ARR）更新自动重载影子寄存器。
- 预分频器的缓冲区中将重新装载预装载值（TIM2/3/4_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIM2/3/4_ARR=0x36 时不同时钟频率下计数器的行为。

图 21-4 计数器时序图，1 分频内部时钟

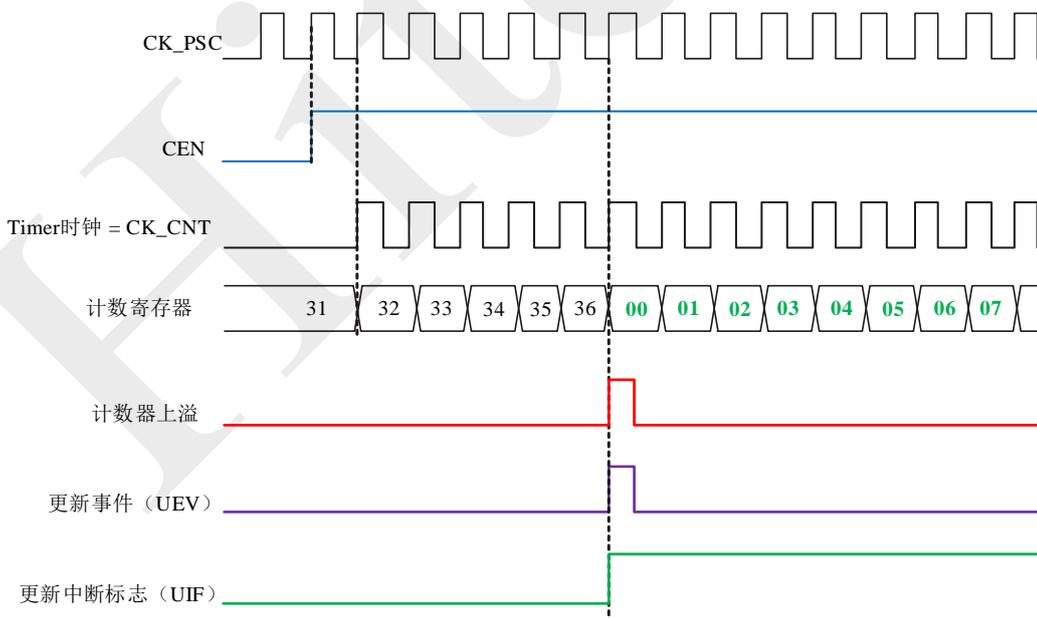


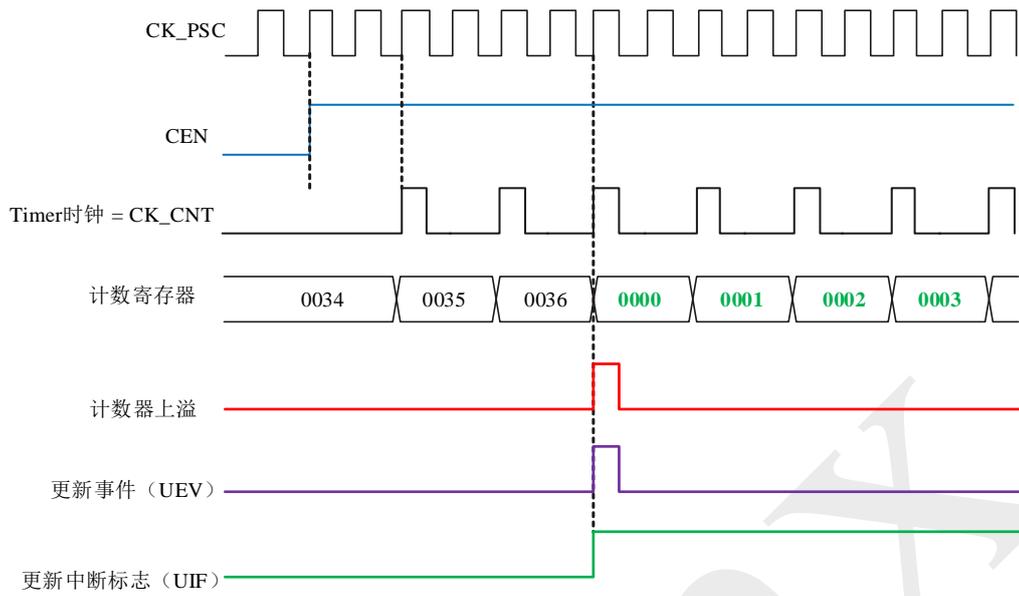
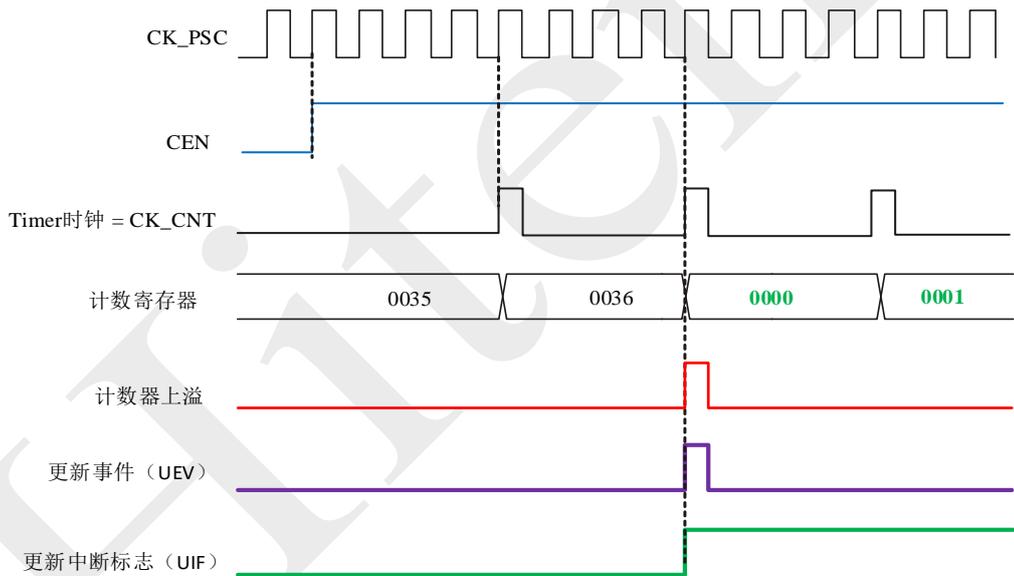
图 21-5 计数器时序图, 2 分频内部时钟

图 21-6 计数器时序图, 4 分频内部时钟


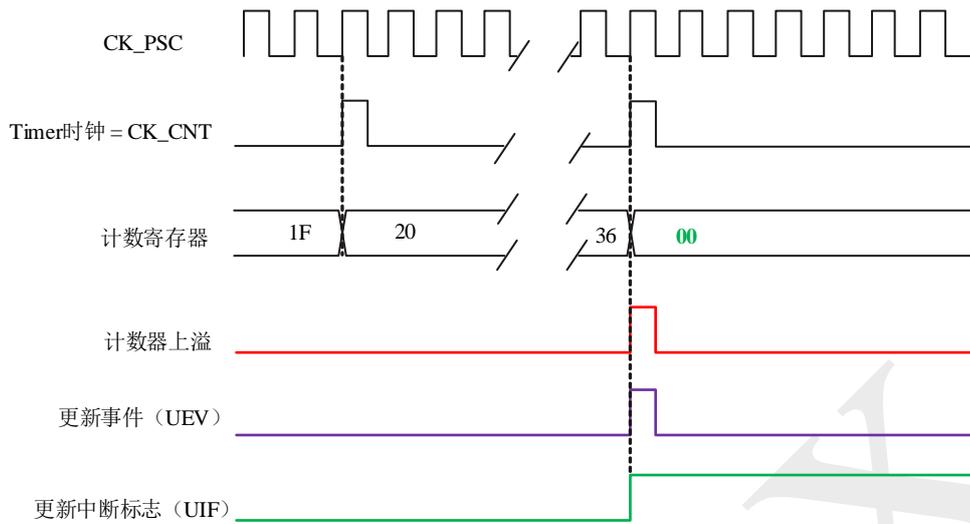
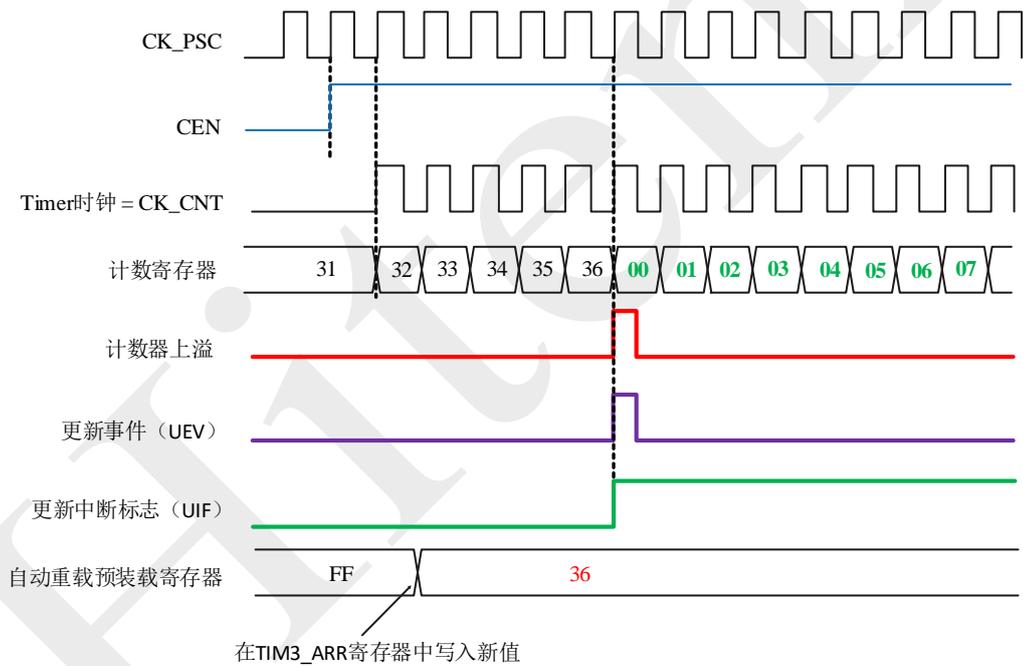
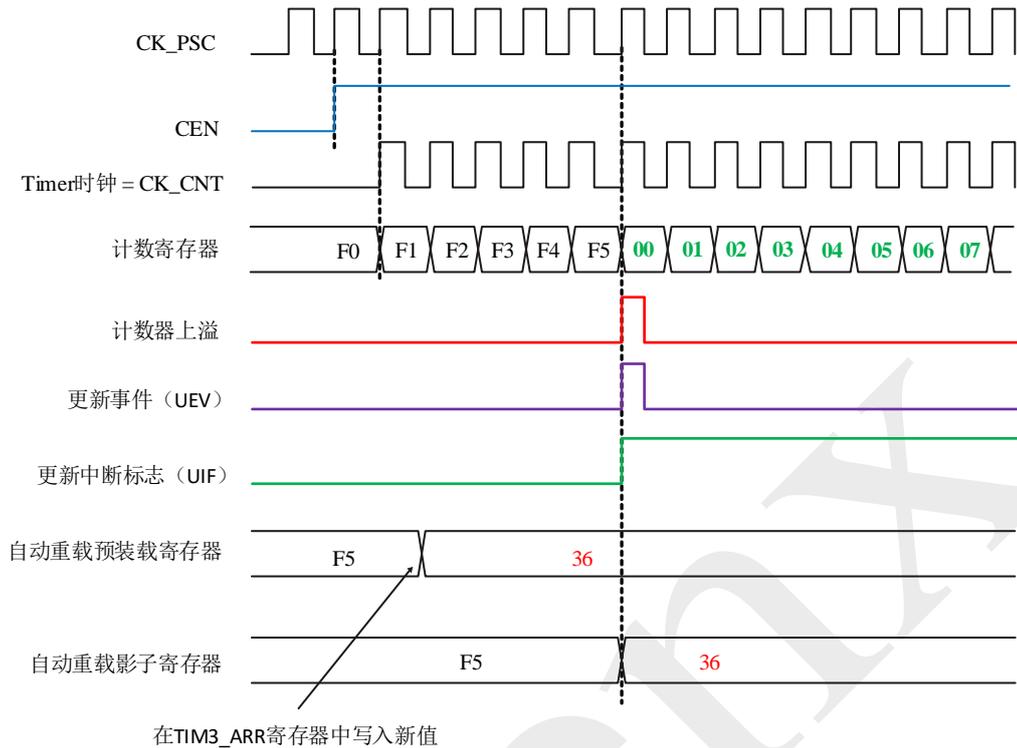
图 21-7 计数器时序图, N 分频内部时钟

图 21-8 计数器时序图, ARPE=0 时更新事件 (未预装载)


图 21-9 计数器时序图，ARPE=1 时更新事件（已预装载）


递减计数模式

在递减计数模式下，计数器从自动重载值（TIM2/3/4_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIM2/3/4_RCR) + 1）后，将产生更新事件（UEV）。否则，将在每次计数器下溢时产生更新事件。

将 TIM2/3/4 时间发生寄存器（TIM2/3/4_EGR）的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIM2/3/4 控制寄存器（TIM2/3/4_CR1）寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIM2/3/4_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM2/3/4_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIM2/3/4 重复计数器寄存器 (TIM2/3/4_RCR) 的内容;
- 预分频器的缓冲区中将重新装载预装载值(TIM2/3/4_PSC 寄存器的内容);
- 自动重载有效寄存器将以预装载值 (TIM2/3/4_ARR 寄存器的内容) 进行更新。

注意: ARR 寄存器更新在计数器重载之前被更新, 因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIM2/3/4_ARR=0x36 时不同时钟频率下计数器的行为。

图 21-10 计数器时序图, 1 分频内部时钟

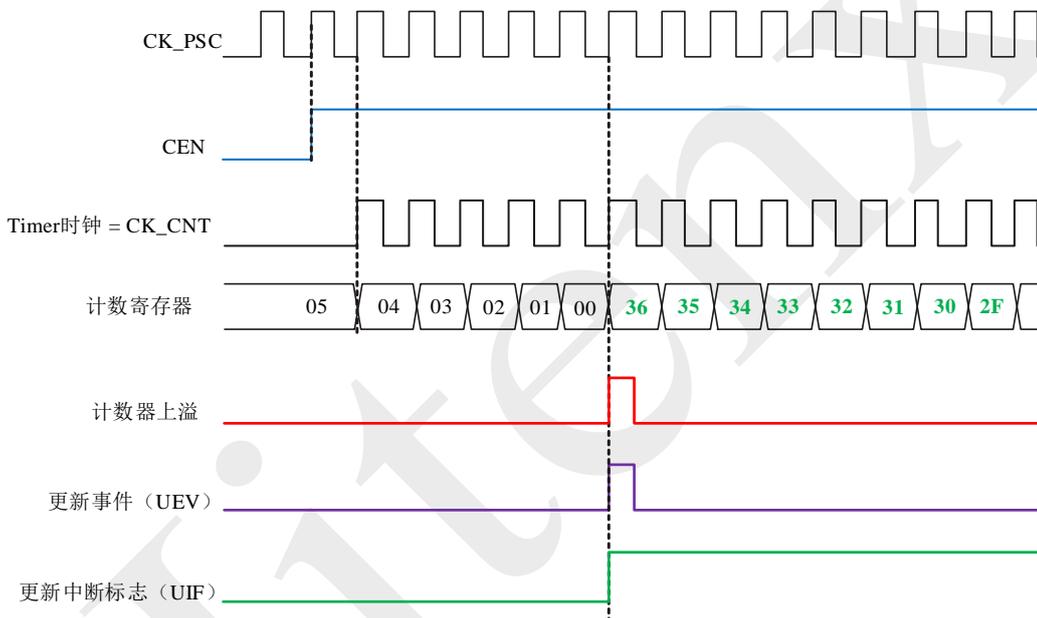


图 21-11 计数器时序图, 2 分频内部时钟

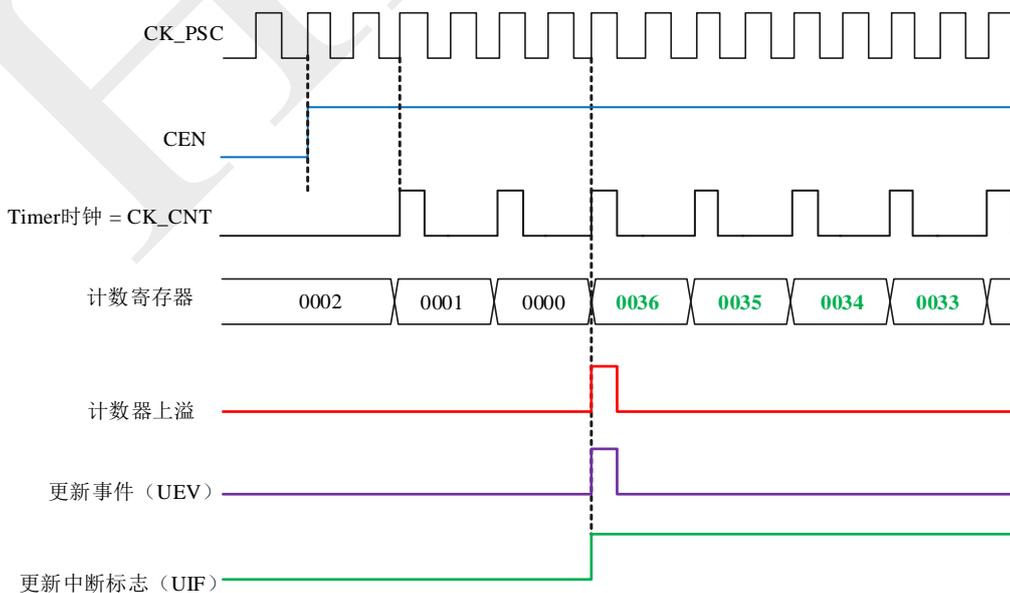


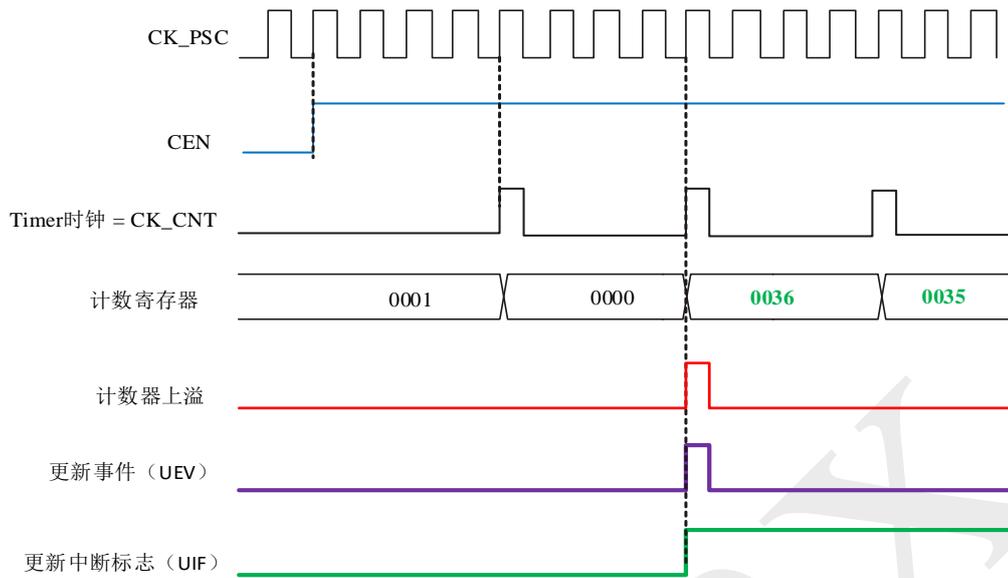
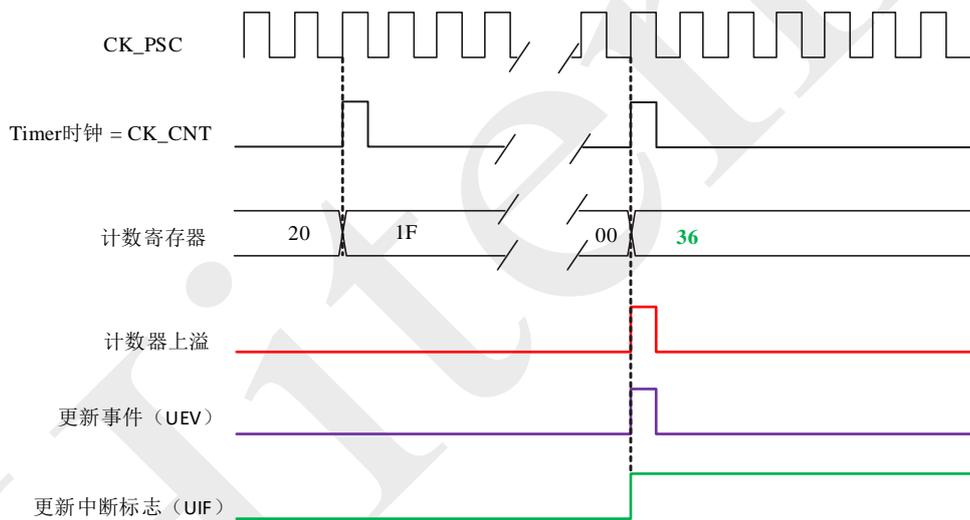
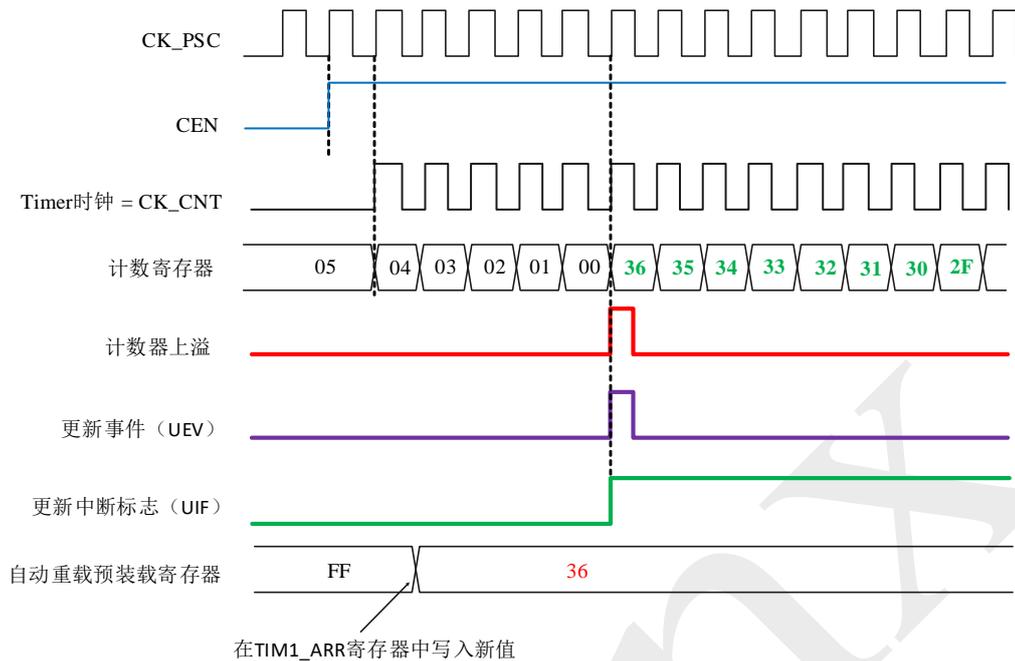
图 21-12 计数器时序图, 4 分频内部时钟

图 21-13 计数器时序图, N 分频内部时钟


图 21-14 计数器时序图，未使用重复计数器时更新事件


中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIM2/3/4_ARR 寄存器的内容）- 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM2/3/4 控制寄存器（TIM2/3/4_CR1）寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式置 1，即：计数器递减计数（中心对齐模式 1，CMS=“01”）、计数器递增计数（中心对齐模式 2，CMS=“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS=“11”）。

此模式下无法写入方向位（TIM2/3/4_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM2/3/4 事件产生寄存器（TIM2/3/4_EGR）中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。通过软件将 TIM2/3/4_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM2/3/4_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不

会同时产生更新中断和捕获中断。

- 重复计算器中将重新装载 TIM2/3/4 重复计算器寄存器 (TIM2/3/4_RCR);
- 预分频器的缓冲区中将重新装载预装载值 (TIM2/3/4 预分频器寄存器 (TIM2/3/4_PSC))。
- 自动重载活动寄存器将以预装载值 (TIM2/3/4 自动重载值寄存器 (TIM2/3/4_ARR)) 进行更新。注意, 如果更新操作是由计数器上溢触发的, 则 ARR 寄存器在计数器重载之前更新, 因此, 下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

注意, 如果更新操作是由计数器上溢触发的, 则 ARR 寄存器在计数器重载之前更新, 因此, 下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 21-15 计数器时序图, 1 分频内部时钟, TIM2/3/4_ARR=0x06

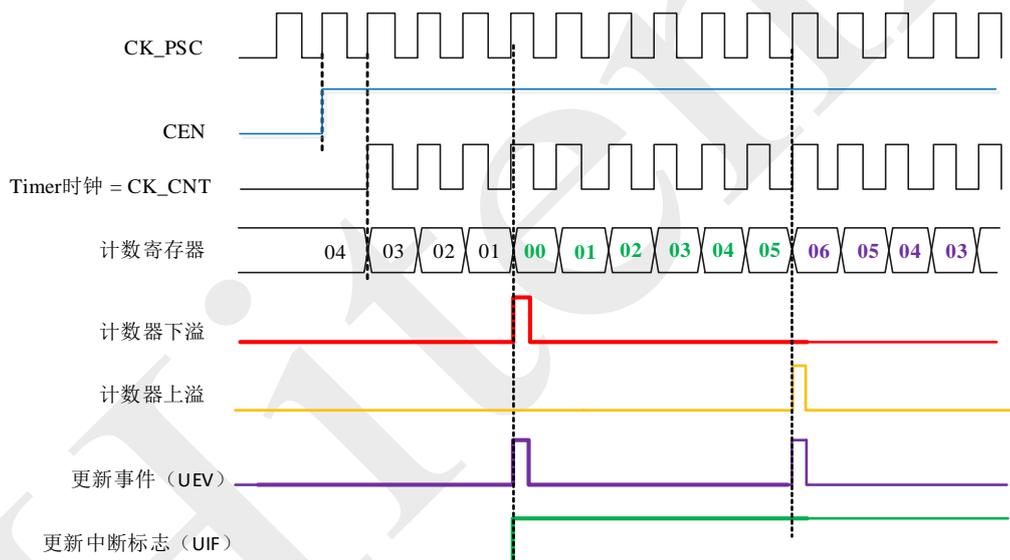
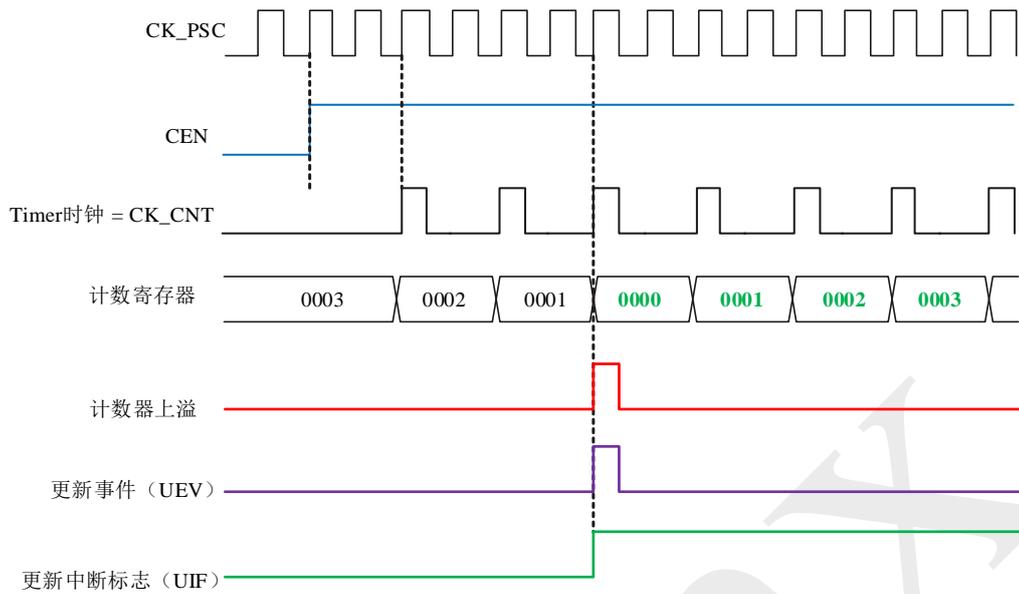
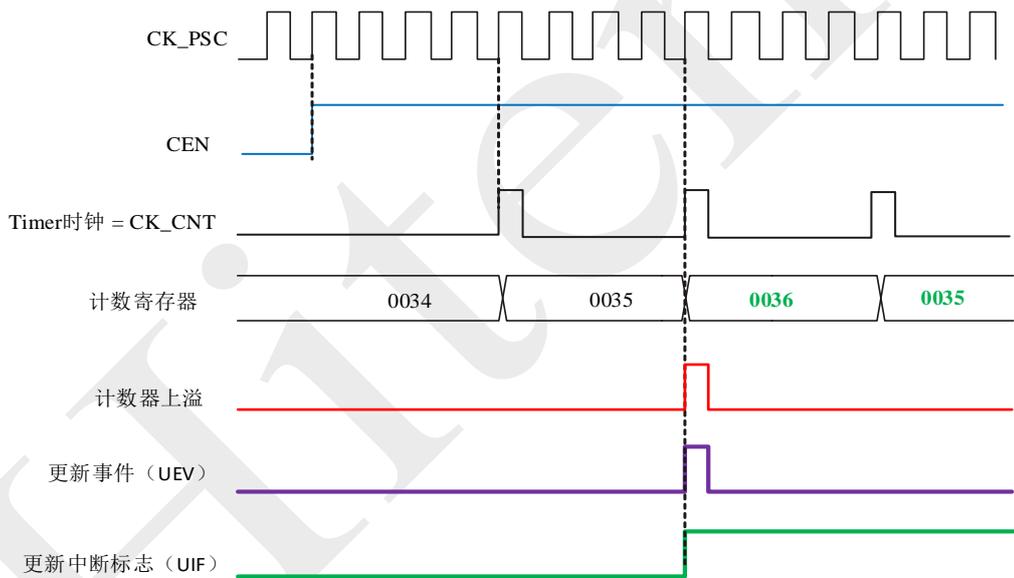


图 21-16 计数器时序图，2 分频内部时钟

图 21-17 计数器时序图，4 分频内部时钟，TIM2/3/4_ARR=0x36


1 此处使用的中心对齐模式 2 或模式 3 与上溢 UIF 一起更新

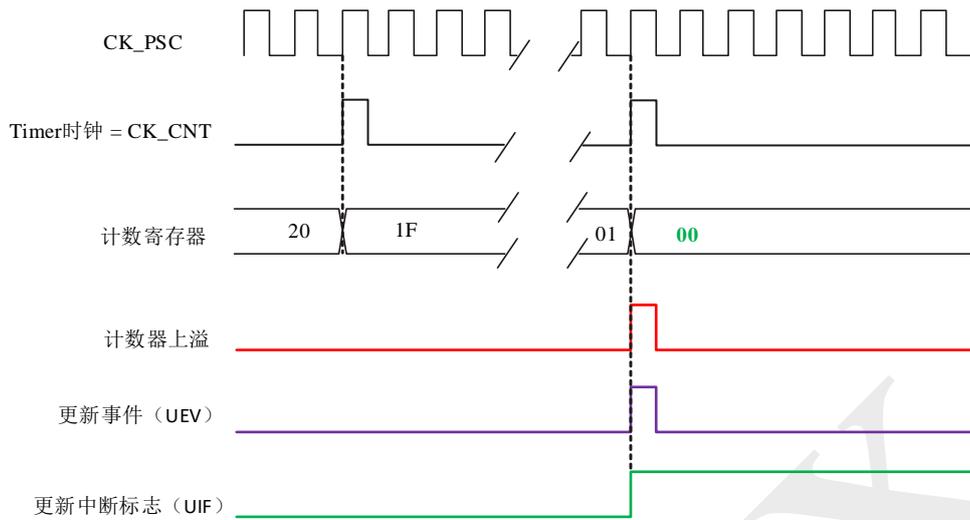
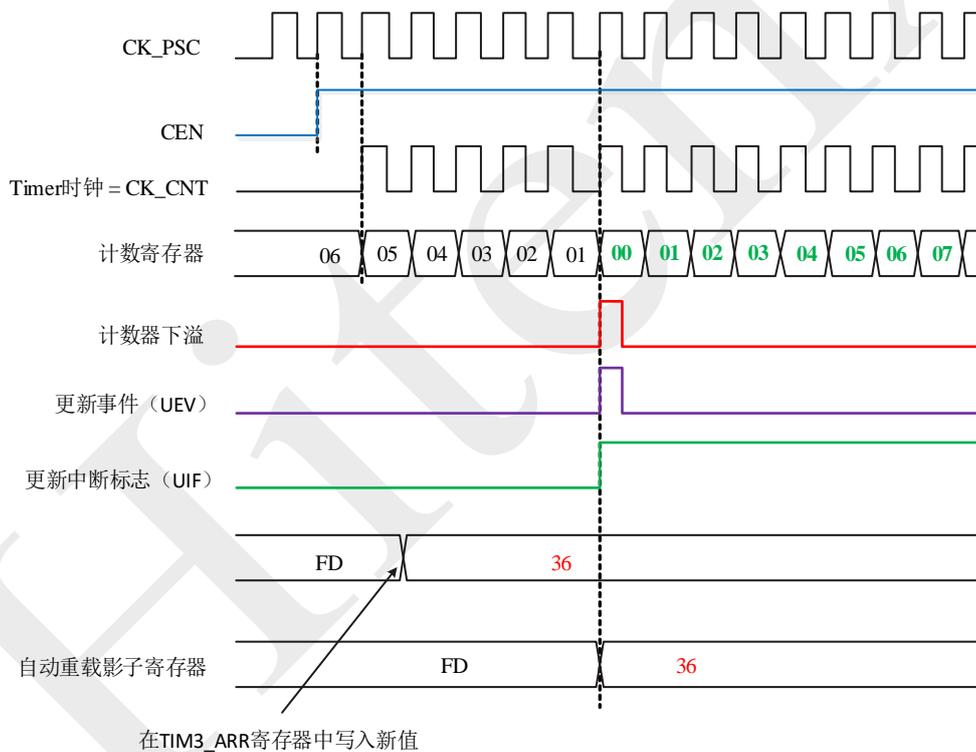
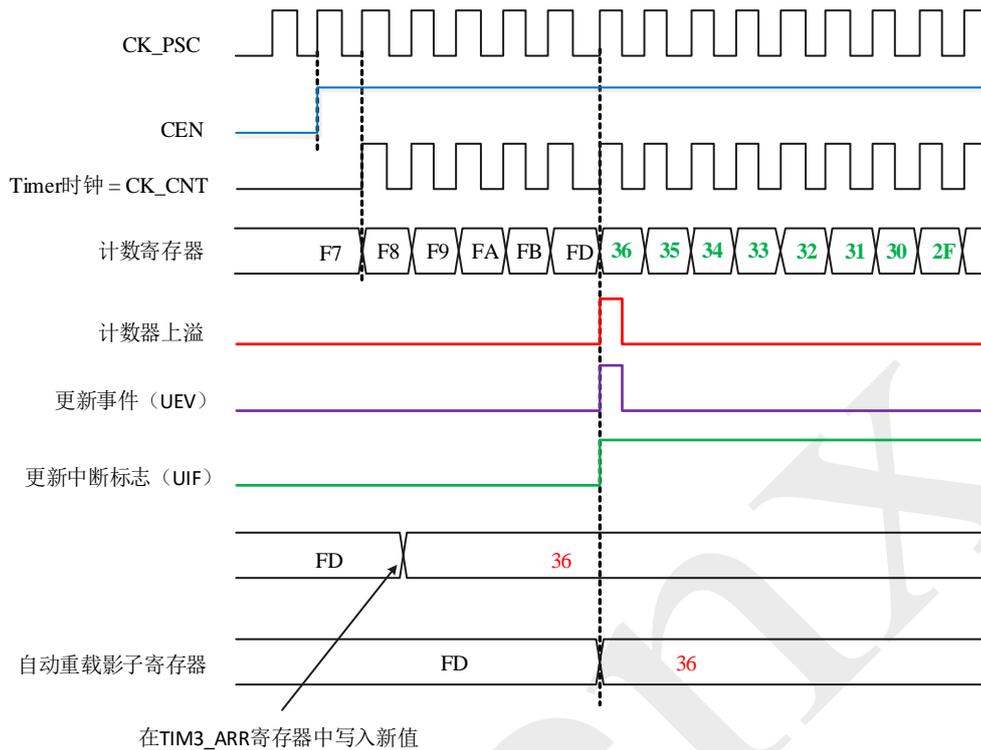
图 21-18 计数器时序图, N 分频内部时钟

图 21-19 计数器时序图, ARPE=1 时的更新事件 (计数器下溢)


图 21-20 计数器时序图，ARPE=1 时的更新事件（计数器上溢）


21.3.3 重复计数器

时基单元介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 $N+1$ 个计数器上溢或下溢（其中， N 是 TIM2/3/4 重复计数器寄存器 (TIM2/3/4_RCR) 的值），数据就将从预装载寄存器转移到影子寄存器 (TIM2/3/4 自动重载寄存器 (TIM2/3/4_ARR)、TIM1 预分频器寄存器 (TIM2/3/4_PSC) 以及比较模式下的 TIM2/3/4_CCRx 捕获/比较寄存器)。

重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢；
- 递减计数模式下的每个计数器下溢；
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2 \times T_{ck}$ 。

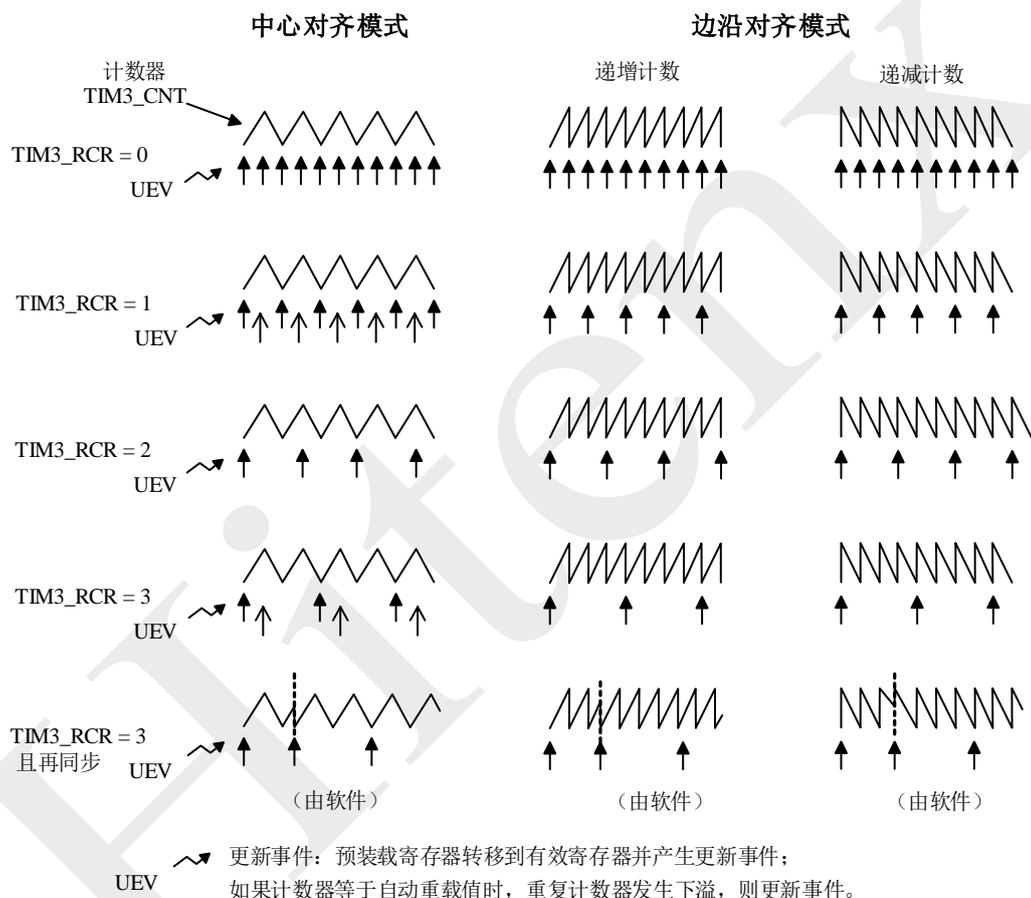
重复计数器是自动重载类型；其重复率为 TIM2/3/4_RCR 寄存器所定义的值。当更新事件由软件（通过将 TIM2/3/4 时间发生寄存器 (TIM2/3/4_EGR) 的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIM2/3/4_RCR 寄存器的

内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 RCR=3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 21-21 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



21.3.4 时钟选择

计数器时钟可由下列时钟源提供：

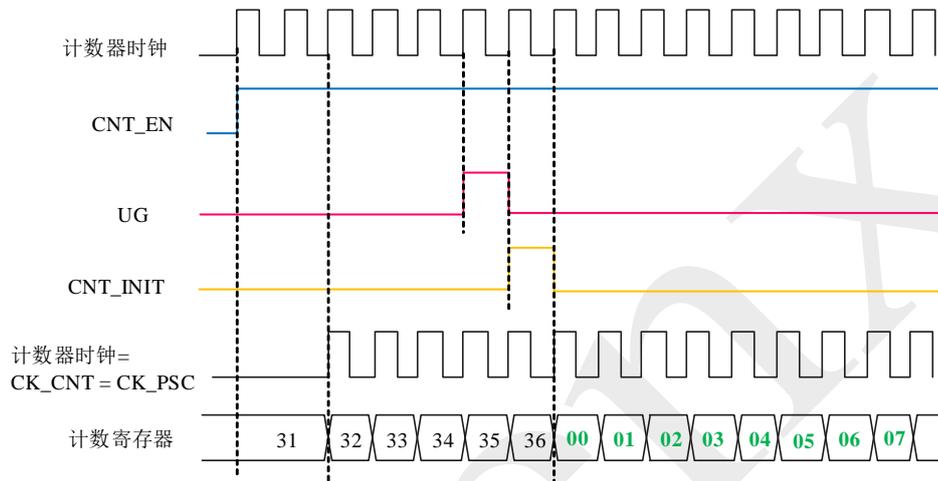
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入 (ITRx)：将一个定时器用作另一个定时器的预分频器

内部时钟源 (CK_INT)

如果禁止从模式控制器 (TIM2/3/4_SMCR 寄存器中 SMS=000), 则 CEN 位、DIR 位 (TIM2/3/4_CR 寄存器中) 和 UG 位 (TIM2/3/4_EGR 寄存器中) 为实际控制位, 并且只能通过软件进行更改 (UG 除外, 仍自动清零)。当对 CEN 位写入 1 时, 预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了正常模式下控制电路与递增计数器的行为 (没有预分频的情况下)。

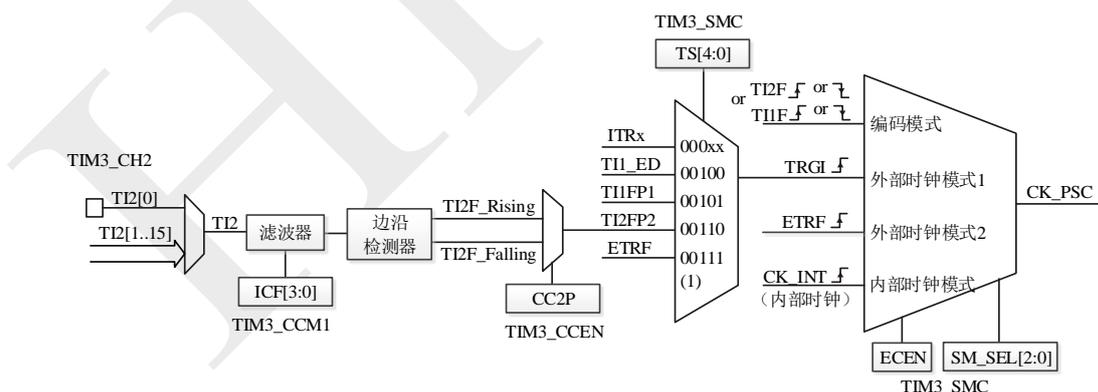
图 21-22 正常模式下的控制电路, 1 分频内部时钟



外部时钟源模式 1

当 TIM2/3/4_SMCR 寄存器中的 SMS=111 时, 可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 21-23 TI2 外部时钟连接示例



1 保留 “01000” 到 “11111” 的代码。

例如, 要使递增计数器在 TI2 输入出现上升沿时计数, 请执行以下步骤:

- 1) 通过 TIM2/3/4 定时器输入选择寄存器 (TIM2/3/4_TISEL) 中的 TI2SEL[3:0] 位域选择适当的 TI2x 源 (内部或外部);
- 2) 通过在 TIM2/3/4_CCMR1 寄存器中写入 CC2S=“01” 来配置通道 2, 使其能够检测 TI2 输入的上升沿。

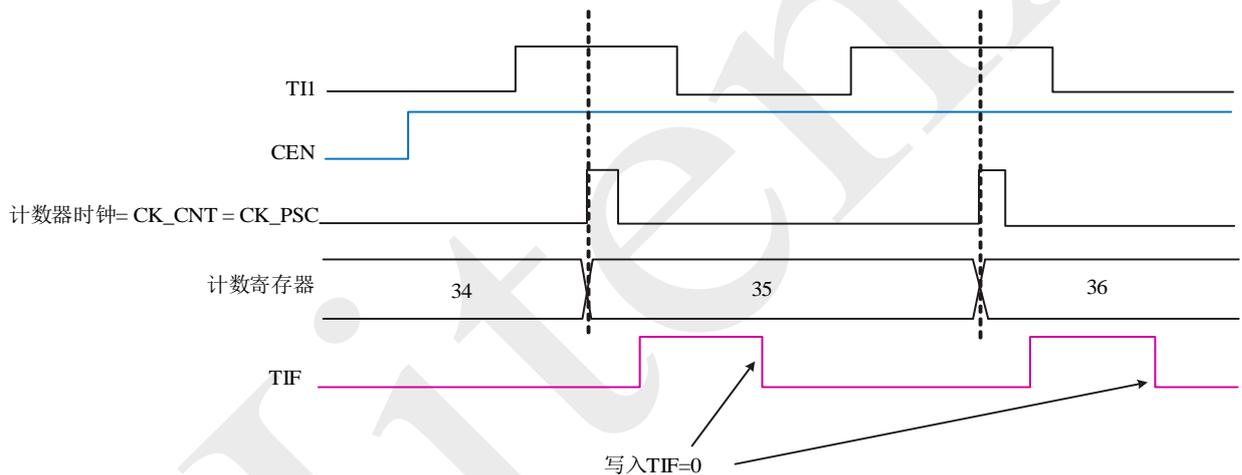
- 3) 通过在 TIM2/3/4_CCMR1 寄存器中写入 IC2F[3:0]位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=“0000”）。
- 4) 通过在 TIM2/3/4_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
- 5) 通过在 TIM2/3/4_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
- 6) 通过在 TIM2/3/4_SMCR 寄存器中写入 TS=110 来选择 TI2 作为输入源。
- 7) 通过在 TIM2/3/4_CR1 寄存器中写入 CEN=1 来使能计数器。

注意： 由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的同步电路引起的。

图 21-24 外部时钟模式 1 下的控制电路



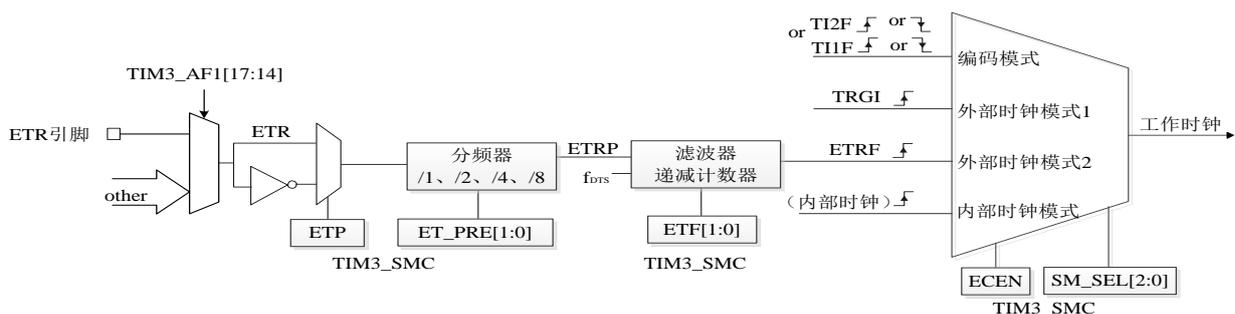
外部时钟源模式 2

通过在 TIM2/3/4_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

下图简要介绍了外部触发输入模块。

图 21-25 外部触发输入模块



例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

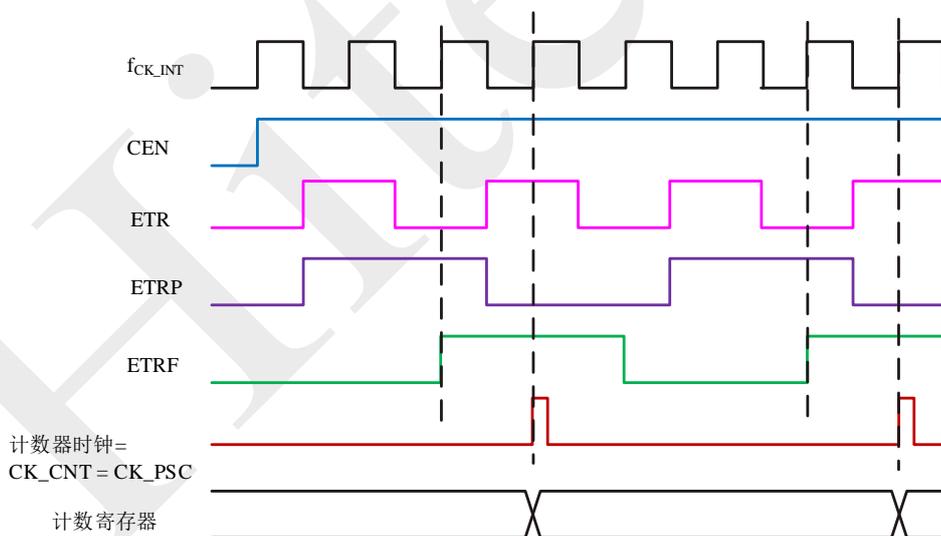
由于此例中不需滤波器，因此在 TIM2/3/4_SMCR 寄存器中写入 ETF[3:0]=0000

- 1) 通过在 TIM2/3/4_AF1 寄存器中写入 ETRSEL[3:0]来选择合适的 ETR 来源（内部或外部）。
- 2) 由于此例中不需滤波器，因此在 TIM2/3/4_SMCR 寄存器中写入 ETF[3:0]=0000
- 3) 通过在 TIM2/3/4_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
- 4) 通过在 TIM2/3/4_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
- 5) 通过在 TIM2/3/4_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
- 6) 通过在 TIM2/3/4_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器可正确捕获的最大频率最多为 TIM2/3/4CLK 频率的 1/4。当 ETRP 信号更快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频设置。

图 21-26 外部时钟模式 2 下的控制电路



21.3.5 捕获/比较通道

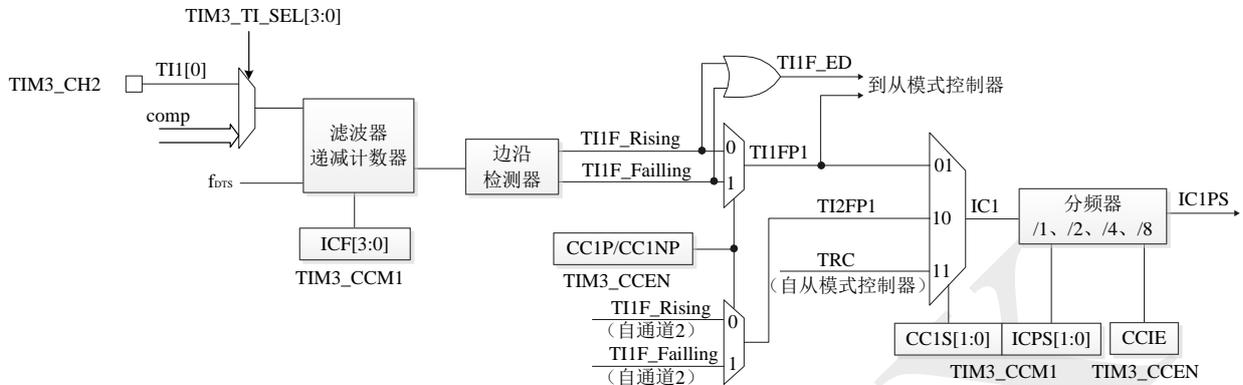
每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下面 3 张图简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控

制器的触发输入，也可用作捕获命令。该信号先进行预分频（ICxPS），而后再进入捕获寄存器。

图 21-27 捕获比较通道 1 输入阶段电路



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 21-28 捕获比较通道 1 主电路

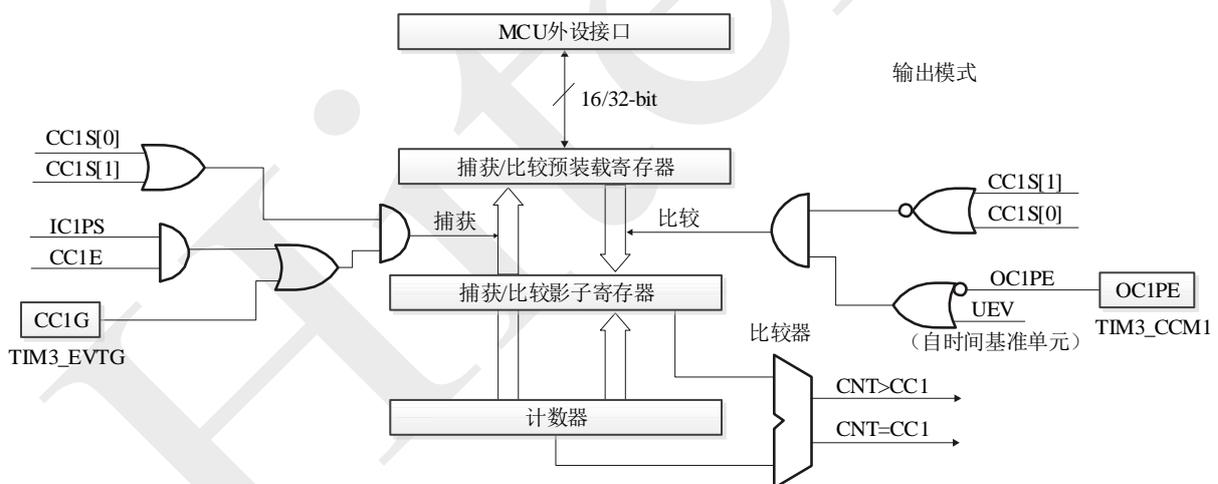
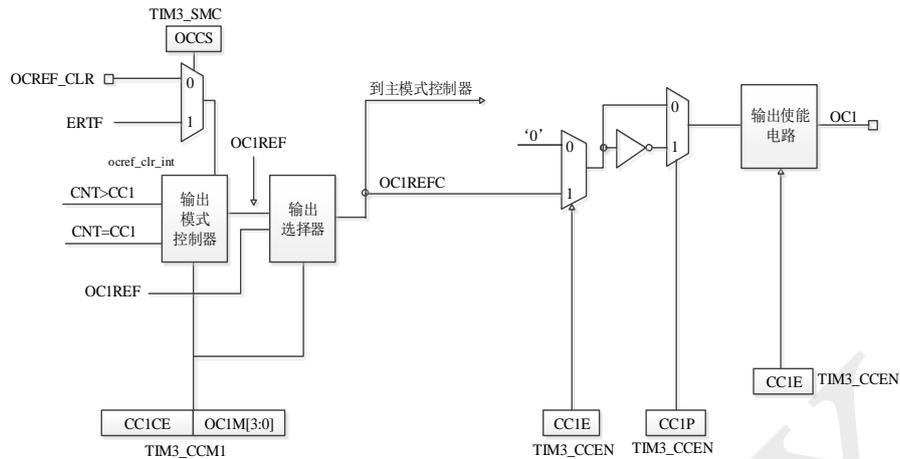


图 21-29 捕获比较通道 1 输出阶段电路


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。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

21.3.6 输入捕获模式

在输入捕获模式下，当相应的 IC_x 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIM2/3/4_CCR_x) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志 (TIM2/3/4_SR 寄存器) 置 1，并可发送中断或 DMA 请求 (如果已使能)。如果发生捕获事件时 CC_xIF 标志已处于高位，则会将重复捕获标志 CC_xOF (TIM2/3/4_SR 寄存器) 置 1。可通过软件方法向 CC_xIF 写入 0 来给 CC_xIF 清零，或读取存储在 TIM2/3/4_CCR_x 寄存器中的已捕获数据。向 CC_xOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM2/3/4_CCR1 中。具体操作步骤如下：

- 1) 通过定时器输入选择 TIM2/3/4 定时器输入选择寄存器 (TIM2/3/4_TISEL) 中的 TI1SEL[3:0] 位域选择适当的 TI1_x 源 (内部或外部)。
- 2) 选择有效输入：TIM2/3/4_CCR1 必须连接到 TI1 输入，因此向 TIM2/3/4_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM2/3/4_CCR1 寄存器将处于只读状态。
- 3) 根据连接到定时器的信号，对所需的输入滤波带宽进行编程 (如果输入为 TI_x 之一，则对 TIM2/3/4_CCMR_x 寄存器中的 IC_xF 位进行编程)。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电

平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIM2/3/4_CCMR1 寄存器中的 IC1F 位写入“0011”。

- 4) 通过向 TIM2/3/4_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿（本例中为上升沿）。
- 5) 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIM2/3/4_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 6) 通过将 TIM2/3/4_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 7) 如果需要，可通过将 TIM2/3/4_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，或者通过将该寄存器中的 CC1_DMAEN 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIM2/3/4_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1_DMAEN 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注意： 通过软件将 TIM2/3/4_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。

21.3.7 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

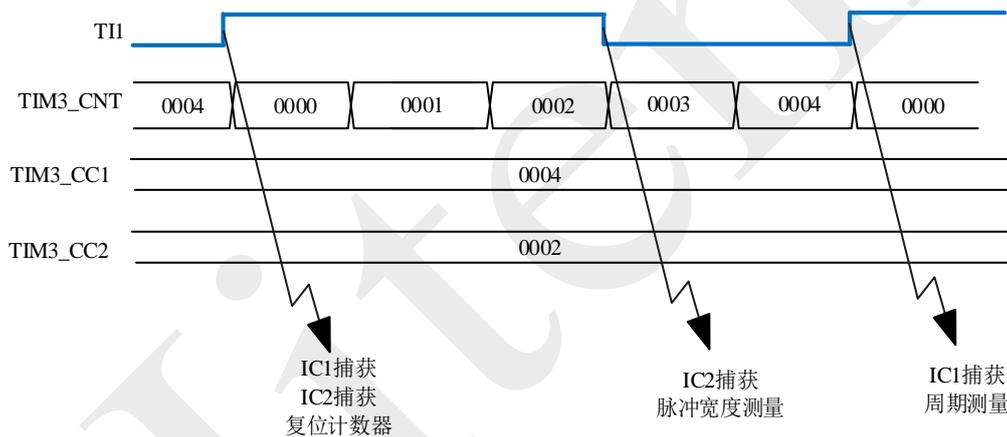
- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIM2/3/4_CCR1 寄存器中）和占空比（位于 TIM2/3/4_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 1) 通过定时器输入选择中的 TI1SEL[3:0]位域选择适当的 TI1x 源（内部或外部）。

- 2) 选择 TIM2/3/4_CCR1 的有效输入：向 TIM2/3/4_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 3) 选择 TI1FP1 的有效极性（同时用于 TIM2/3/4_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 4) 选择 TIM2/3/4_CCR2 的有效输入：向 TIM2/3/4_CCMR1 寄存器中的 CC2S 位写入 10（选择 TI1）。
- 5) 选择 TI1FP2 的有效极性（用于 TIM2/3/4_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
- 6) 选择有效触发输入：向 TIM2/3/4_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。
- 7) 将从模式控制器配置为复位模式：向 TIM2/3/4_SMCR 寄存器中的 SMS 位写入 100。
- 8) 使能捕获：向 TIM2/3/4_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 21-30 输入捕获模式时序



21.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一段时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIM2/3/4_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM2/3/4_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCxM=“0000”），也可设置为有效电平（OCxM=“0001”）、无效电平（OCxM=“0010”）或进行翻转（OCxM=“0011”）。
- 将中断状态寄存器中的标志置 1（TIM2/3/4_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIM2/3/4_DIER 寄存器中的 CCxIE 位）置 1，将生成中断。
- 如果相应使能位（TIM2/3/4_DIER 寄存器的 CCx_DMAEN 位，TIM3_CR2

寄存器的 CC_DMASEL 位，用来选择 DMA 请求) 置 1，将发送 DMA 请求。

使用 TIM2/3/4_CCMRx 寄存器中的 OCxPE 位，可将 TIM2/3/4_CCRx 寄存器配置为带或不带预装载寄存器。

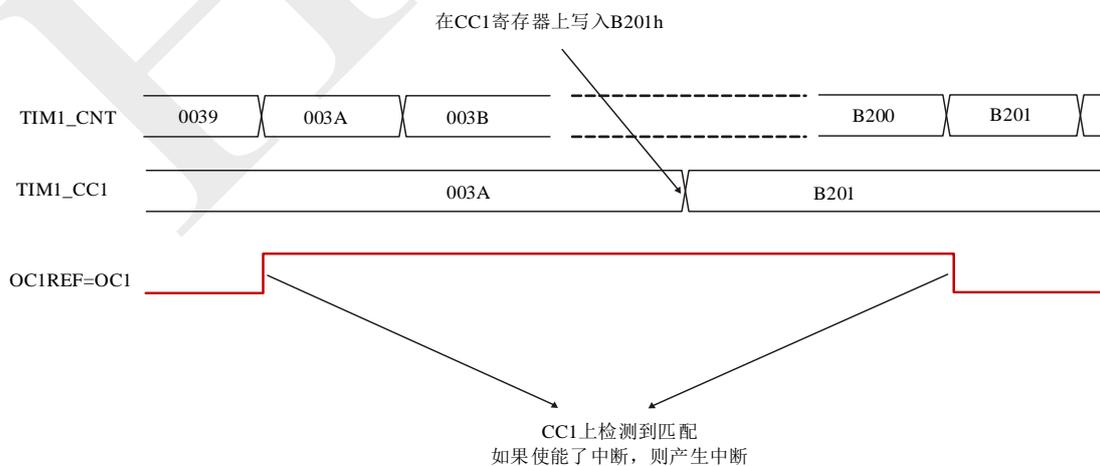
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

- 1) 选择计数器时钟（内部、外部、预分频器）。
- 2) 在 TIM2/3/4 自动重载值寄存器（TIM2/3/4_ARR）和 TIM2/3/4_CCRx 寄存器中写入所需数据。
- 3) 如果要生成中断请求，则需将 CCxIE 位置 1。
- 4) 选择输出模式。例如：
 - 当 CNT 与 CCx 匹配时，写入 OCxM=“0011”以翻转 OCx 输出引脚；
 - 写入 OCxPE=0 以禁止预装载寄存器；
 - 写入 CCxP=0 以选择高电平有效极性；
 - 写入 CCxE=1 以使能输出。
- 5) 通过将 TIM2/3/4 控制寄存器（TIM2/3/4_CR1）中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM2/3/4_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM2/3/4_CCRx 影子寄存器）。下图给出了一个示例。

图 21-31 输出比较模式，翻转 OC1



21.3.9 强制输出模式

在输出模式（TIM2/3/4_CCMRx 寄存器中的 CCxS 位=00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号（OCxREF/OCx）强制设置为有效电平，只需向相应 TIM2/3/4_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=>OCx 强制设置为高电平。

通过向 TIM2/3/4_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIM2/3/4_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。

21.3.10 PWM 模式

PWM 模式可以生成一个信号，该信号频率由 TIM2/3/4_ARR 寄存器值决定，其占空比则由 TIM2/3/4_CCRx 寄存器值决定。

通过向 TIM2/3/4_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIM2/3/4_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIM2/3/4_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIM2/3/4_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIM2/3/4_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIM3_CCER 寄存器中的 CCxE 位置 1 来使能。

在 PWM 模式（1 或 2）下，TIM2/3/4_CNT 总是与 TIM2/3/4_CCRx 进行比较，以确定是 $TIM2/3/4_CCRx \leq TIM2/3/4_CNT$ 还是 $TIM2/3/4_CNT \leq TIM2/3/4_CCRx$ （取决于计数器计数方向）。不过，为符合 OCREF_CLR 功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCREF），OCREF 信号仅在以下情况下变为有效状态：

- 1) 比较结果发生改变，或
- 2) 输出比较模式（TIM2/3/4_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”）

或“111”)

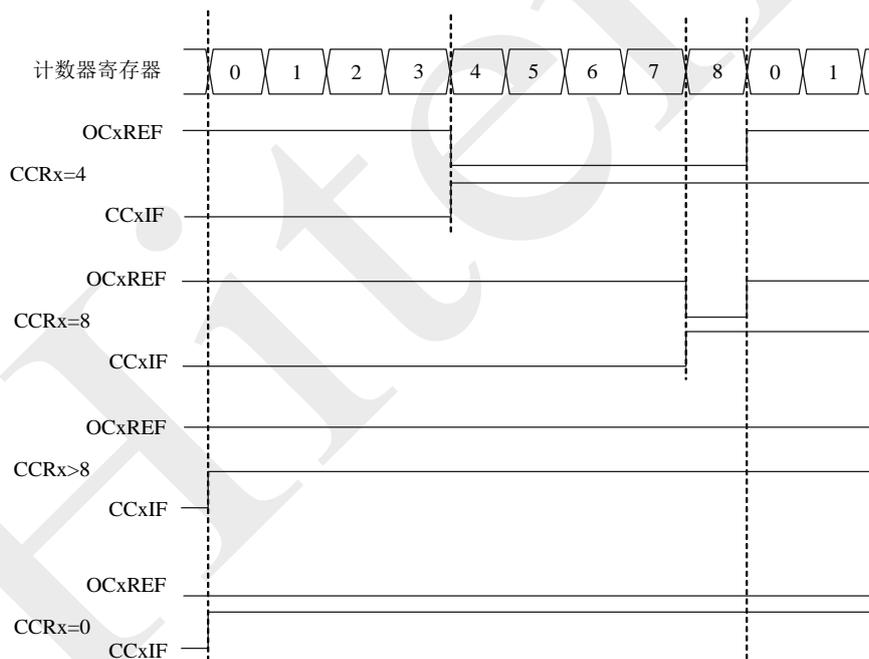
定时器运行期间，可以通过软件强制 PWM 输出

根据 TIM2/3/4_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

PWM 边沿对齐模式

- 递增计数配置
- 当 TIM2/3/4_CR1 寄存器中的 DIR 位为低时执行递增计数。
- 以下以 PWM 模式 1 为例。只要 $TIM2/3/4_CNT < TIM2/3/4_CCR_x$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIM2/3/4_CCRx 中的比较值大于自动重载值 (TIM2/3/4_ARR 中)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。图 21-32 举例介绍边沿对齐模式的一些 PWM 波形 (TIM2/3/4_ARR=8)。

图 21-32 边沿对齐的 PWM 模式 (ARR=8)



- 递减计数配置
- 当 TIM2/3/4_CR1 寄存器中的 DIR 位为高时执行递减计数。
- 在 PWM 模式 1 下，只要 $TIM2/3/4_CNT > TIM2/3/4_CCR_x$ ，参考信号 OCxREF 便为低电平，否则为高电平。如果 TIM2/3/4_CCRx 中的比较值大于 TIM2/3/4_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

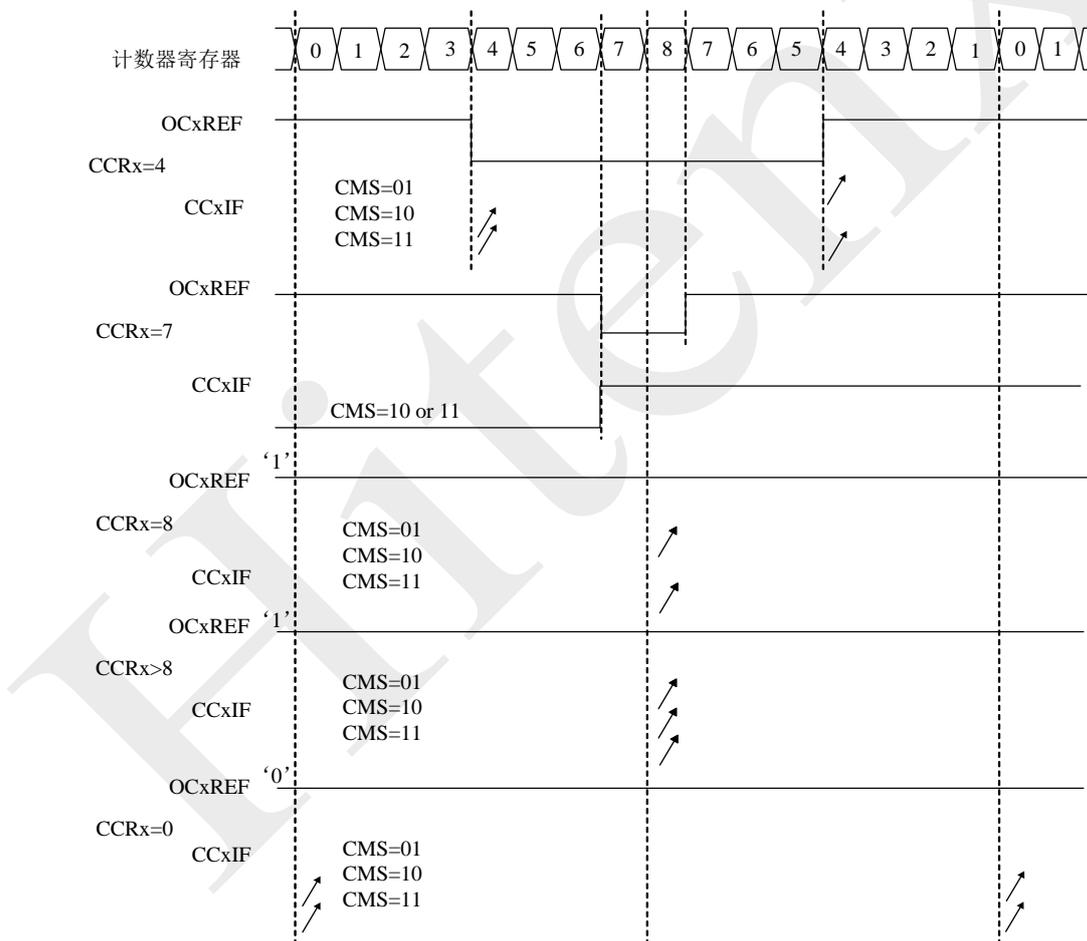
PWM 中心对齐模式

当 TIM2/3/4_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 OCxREF/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIM2/3/4_CR1 寄存器中的方向位（DIR）由硬件更新，不得通过软件更改。

下图显示了中心对齐模式的 PWM 波形，在此例中：

- TIM2/3/4_ARR=8。
- PWM 模式为 PWM 模式 1。
- 在根据 TIM2/3/4_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 21-33 中心对齐的 PWM 模式 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIM2/3/4_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：

- 如果写入计数器中的值大于自动重载值 ($TIM2/3/4_CNT > TIM2/3/4_ARR$), 计数方向不会更新。例如, 如果计数器之前递增计数, 则继续递增计数。
- 如果向计数器写入“0”或 $TIM2/3/4_ARR$ 的值, 计数方向会更新, 但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是: 在启动计数器前通过软件生成更新 (将 $TIM2/3/4_EGR$ 寄存器中的 UG 位置 1), 并且不要在计数器运行过程中对其执行写操作。

21.3.11 不对称 PWM 模式

在不对称模式下, 生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 $TIM2/3/4_ARR$ 寄存器的值确定, 而占空比和相移则由一对 $TIM2/3/4_CCR_x$ 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM, 这样每半个 PWM 周期便会调节一次 PWM:

- OC1REFC (或 OC2REFC) 由 $TIM2/3/4_CCR1$ 和 $TIM2/3/4_CCR2$ 控制
- OC3REFC (或 OC4REFC) 由 $TIM2/3/4_CCR3$ 和 $TIM2/3/4_CCR4$ 控制

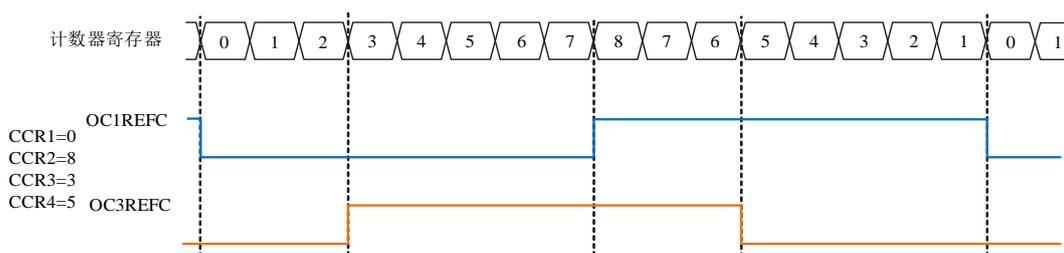
两个通道可以独立选择不对称 PWM 模式 (每对 CCR 寄存器一个 OC_x 输出), 只需向 $TIM2/3/4_CCMR_x$ 寄存器的 OC_xM 位写入“1110” (不对称 PWM 模式 1) 或“1111” (不对称 PWM 模式 2)。

注意: 出于兼容性原因, $OC_xM[3:0]$ 位域分为两部分, 最高有效位与最低有效的 3 位不相邻。

给定通道用作不对称 PWM 通道时, 也可使用其互补通道。例如, 如果通道 1 上产生 OC1REFC 信号 (不对称 PWM 模式 1), 则由于不对称 PWM 模式 1 的原因, 通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

下图显示了不对称 PWM 模式下可以产生的信号示例 (通道 1 到通道 4 在不对称 PWM 模式 1 下配置)。与死区发生器配合使用时, 这可控制相移全桥直流到直流转换器。

图 21-34 50%占空比时产生的 2 个相移 PWM 信号



21.3.12 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的两个脉冲间允许存在可编程延时和相移。频率由 TIM2/3/4_ARR 寄存器的值确定，而占空比和延时则由两个 TIM2/3/4_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIM2/3/4_CCR1 和 TIM2/3/4_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIM2/3/4_CCR3 和 TIM2/3/4_CCR4 控制

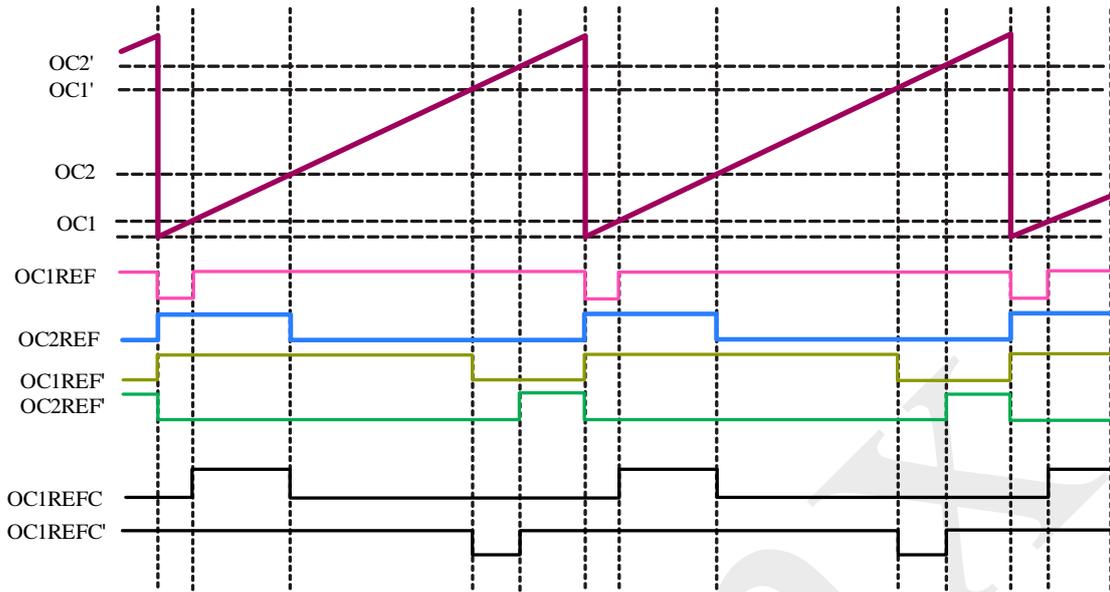
两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIM2/3/4_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注：出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

下图显示了可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 1 在组合 PWM 模式 1 下配置。
- 通道 2 在 PWM 模式 2 下配置。

图 21-35 通道 1 和通道 2 上的组合 PWM 模式


OC1REFC = OC1REF 与 OC2REF 的与计算
 OC1REFC' = OC1REF' 与 OC2REF' 的或计算

21.3.13 单脉冲模式

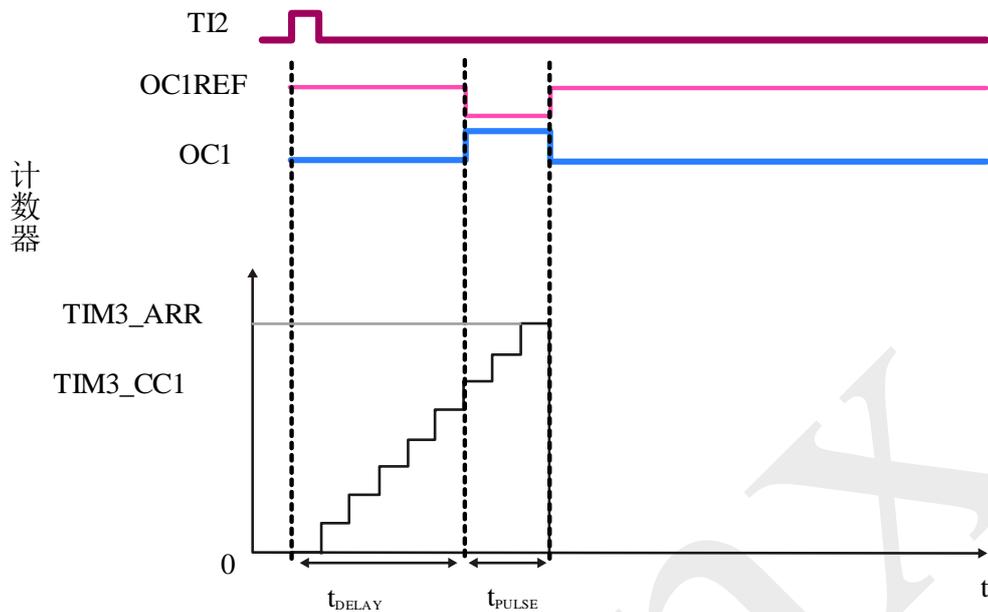
单脉冲模式（OPM）是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIM2/3/4_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等等待触发时），必须进行如下配置：

- 递增计数时：CNT < CCx ≤ ARR（特别注意，0 < CCx）
- 递减计数时：CNT > CCx

图 21-36 单脉冲模式



例如上图，希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

- 1) 在 TIM2/3/4_CCMR1 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
- 2) 在 TIM2/3/4_CCER 寄存器中写入 CC2P=0 和 CC2NP="0"，使 TI2FP2 能够检测上升沿。
- 3) 在 TIM2/3/4_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 4) 在 TIM2/3/4_SMCR 寄存器中写入 SMS="110" (触发模式)，使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM2/3/4 控制寄存器 (TIM2/3/4_CR1) 的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIM2/3/4_ARR-TIM2/3/4_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM2/3/4_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIM2/3/4_CCMR1 寄存器的 OC1PE 和 TIM2/3/4_CR1 寄存器的 ARPE 中写入 1，使能预装载寄存器。这种情况下，必须在 TIM2/3/4_CCR1 寄存器中写入比较值并在 TIM2/3/4_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的

值为“0”。

在本例中，TIM2/3/4_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM2/3/4_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIM2/3/4_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（ t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIM2/3/4_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

21.3.14 可重触发单脉冲模式

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如单脉冲模式所述：

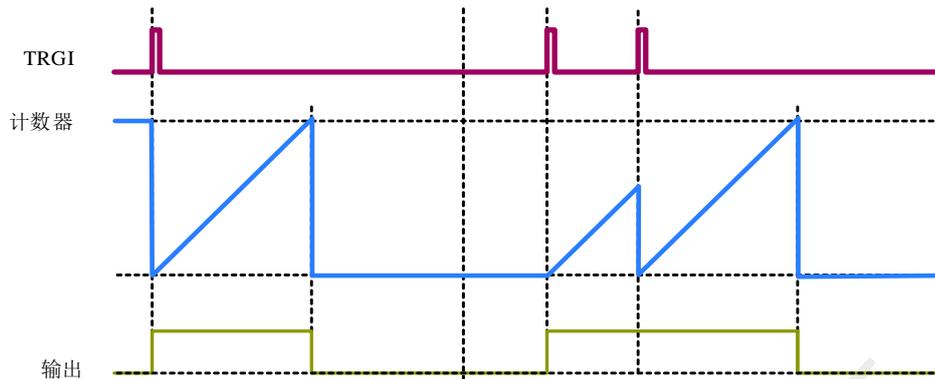
- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIM2/3/4_SMCR 寄存器中的位 SMS[3:0]=“1000”（组合复位+触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0]位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCx 必须高于或等于 ARR。

注意： 出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

注意： 此模式不能与中心对齐 PWM 模式一起使用。在 TIM2/3/4 控制寄存器 1（TIM2/3/4_CR1）中必须设置 CMS[1:0]=“00”。

图 21-37 可重触发单脉冲模式


21.3.15 发生外部事件时清除 OCxREF 信号

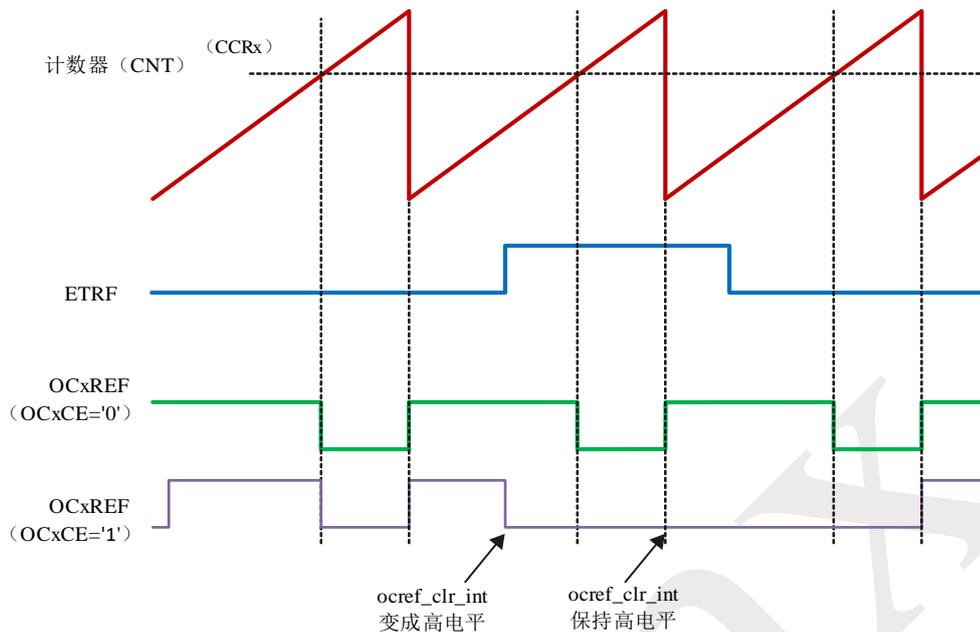
对于给定通道，在 ETRF 输入施加高电平（相应 TIM2/3/4_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件（UEV）。

该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。

选择 ETRF 时，ETR 必须配置如下：

- 1) 必须关闭外部触发预分频器：TIM2/3/4 从模式控制寄存器（TIM2/3/4_SMCR）中的 ETPS[1:0]位置“00”。
- 2) 必须禁止外部时钟模式 2：TIM2/3/4_SMCR 寄存器中的 ECE 位置“0”。
- 3) 外部触发极性(ETP)和外部触发滤波器(ETF)可根据用户需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIM2/3/4 编程为 PWM 模式。

图 21-38 清除 TIM2/3/4 的 OCxREF


注意： 如果 PWM 的占空比为 100% ($CCx > ARR$)，则下次计数器溢出时会再次使能 $OCxREF$ 。

21.3.16 UIF 位重映射

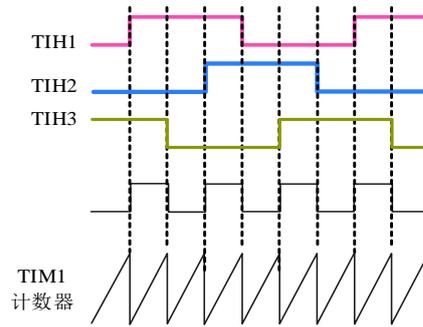
TIM2/3/4_CR1 寄存器中的 UIFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIM2/3/4CNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

21.3.17 定时器输入异或功能

借助 TIM2/3/4_CR2 寄存器中的 TI1_XOR_SEL 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIM2/3/4_CH1，TIM2/3/4_CH2 和 TIM2/3/4_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔（如下图所示）。

图 21-39 测量 3 个信号上边沿之间的时间间隔


21.3.18 定时器与外部触发同步

TIM2/3/4 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

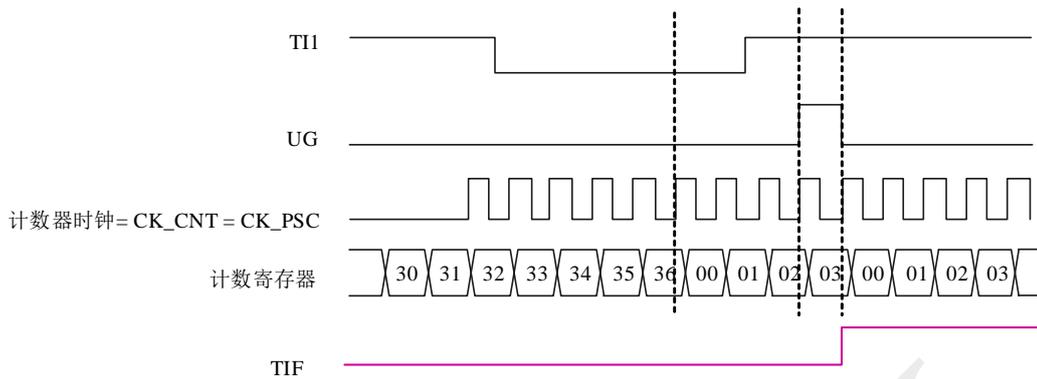
当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIM2/3/4_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIM2/3/4_ARR 和 TIM2/3/4_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM2/3/4_CCMR1 寄存器中的 CC1S=01。在 TIM2/3/4_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
- 在 TIM2/3/4_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM2/3/4_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIM2/3/4_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIM2/3/4_SR 寄存器中的 TI 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIM2/3/4_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIM2/3/4_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 21-40 复位模式下的控制电路


从模式：门控模式

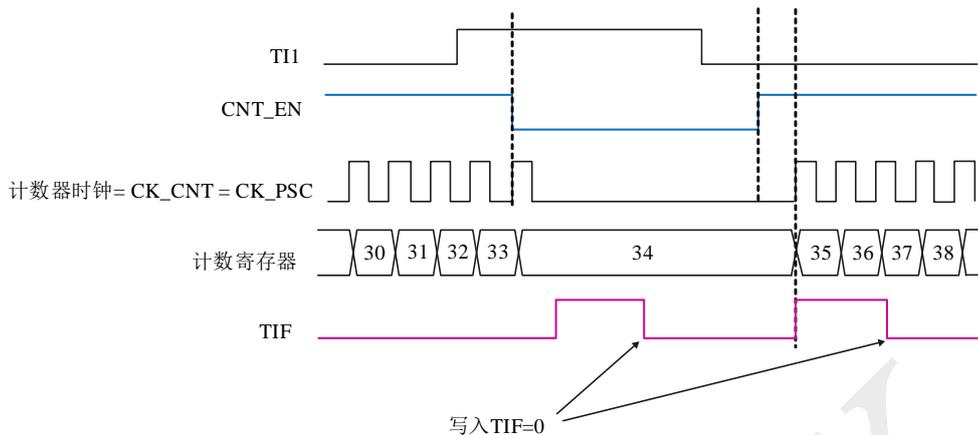
输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM2/3/4_CCMR1 寄存器中的 CC1S=01。在 TIM2/3/4 捕获/比较使能寄存器（TIM2/3/4_CCER）中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIM2/3/4 从模式控制寄存器（TIM2/3/4_SMCR）中写入 SMS=“101”，将定时器配置为门控模式。在 TIM2/3/4_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
- 在 TIM2/3/4 控制寄存器 1（TIM2/3/4_CR1）中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIM2/3/4_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 21-41 门控模式下的控制电路


注： 由于门控模式作用于电平而非边沿，因此在门控模式下，“ $CCxP=CCxNP=1$ ”（同时检测上升沿和下降沿）不发挥任何作用。

从模式：触发模式

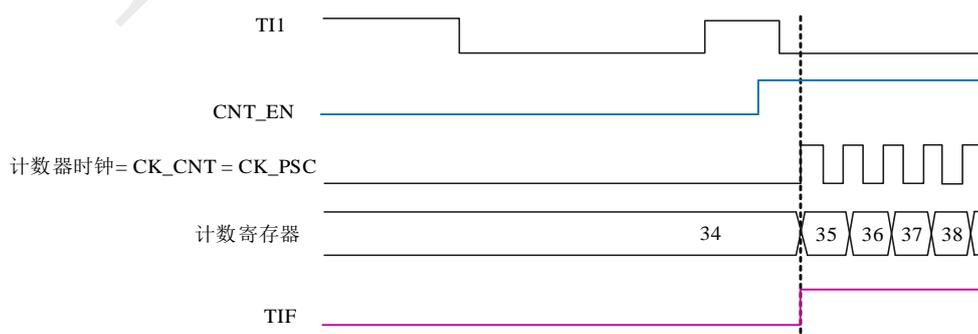
所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIM2/3/4_CCMR1 寄存器中的 CC2S=01。在 TIM2/3/4_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIM2/3/4_SMCR 寄存器中写入 SMS=“110”，将定时器配置为触发模式。在 TIM2/3/4_SMCR 寄存器中写入 TS=“110”，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 21-42 触发模式下的控制电路


从模式：外部时钟模式 2+触发模式

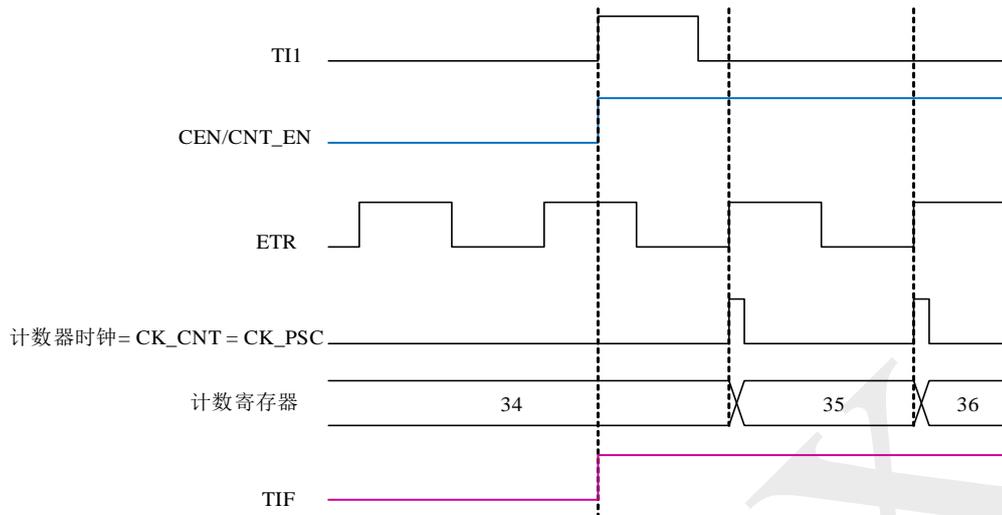
外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIM2/3/4_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

- 1) 通过对 TIM2/3/4 从模式控制寄存器 (TIM2/3/4_SMCR) 进行如下编程，配置外部触发输入电路：
 - ETF= “0000”：无滤波器。
 - ETPS= “00”：禁止预分频器。
 - ETP= “0”：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
- 2) 如下配置通道 1，以检测 TI 的上升沿：
 - IC1F= “0000”：无滤波器。
 - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
 - TIM2/3/4_CCMR1 寄存器中 CC1S= “01”，只选择输入捕获源。
 - TIM2/3/4_CCER 寄存器中 CC1P= “0” 且 CC1NP= “0”，以确定极性（仅检测上升沿）。
- 3) 在 TIM2/3/4_SMCR 寄存器中写入 SMS= “110”，将定时器配置为触发模式。在 TIM2/3/4_SMCR 寄存器中写入 TS= “101”，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

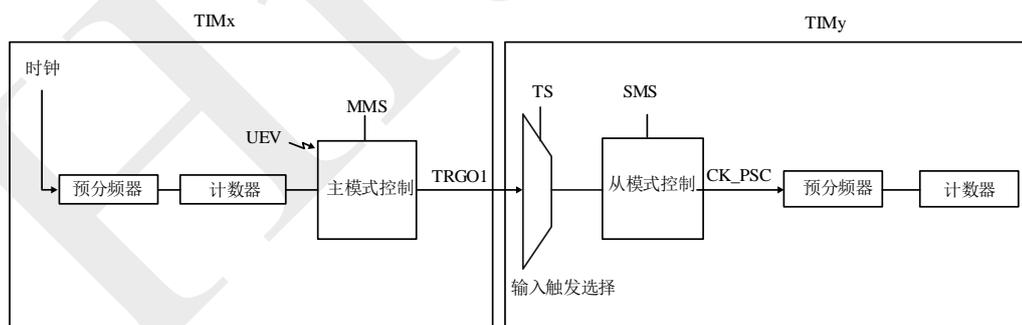
图 21-43 外部时钟模式 2+触发模式下的控制电路


21.3.19 定时器同步

TIM2/3/4 定时器从内部连接在一起，以实现定时器同步或链接。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

下图的主/从定时器示例简要介绍了触发选择和主模式选择框图。

注意：必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

图 21-44 主从定时器示例


将一个定时器用作另一个定时器的预分频器

例如，可以将定时器 x 配置为定时器 y 的预分频器。请参见上图。为此，需遵循以下步骤：

- 1) 将定时器 x 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如在 TIM2/3/4_CR2 寄存器中写入 MM_SEL=010，则每次生成更新事件时，TRGO1 都会输出一个上升沿。
- 2) 要将定时器 x 的 TRGO1 输出连接到定时器 y，必须将定时器 y 配置为从模式，使用 ITR1 作为内部触发。通过 TIMy_SMCR 寄存器中的 TS 位（写入

TS=000) 可对此进行选择。

- 3) 然后将从模式控制器设为外部时钟模式 1 (在 TIMy_SMCR 寄存器中写入 SMS=111)。这样一来, 定时器 y 的时钟将由定时器 x 周期性触发信号的上升沿 (与定时器 x 的计数器上溢对应) 提供。
- 4) 最后必须通过将这两个定时器的相应 CEN 位 (TIM3_CR1 寄存器) 置 1 同时使能二者。

注: 如选择定时器 x 的 OCx 信号作为触发输出 (MM_SEL=1xx), 该信号的上升沿将用于驱动定时器 y 的计数器。

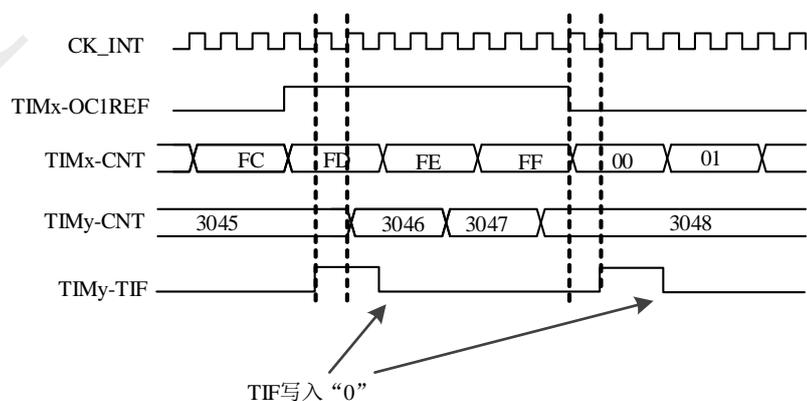
使用一个定时器使能另一个定时器

本例中通过定时器 x 的输出比较 1 来使能定时器 y。相关连接图, 请参见图 31-46。仅当定时器 x 的 OC1REF 为高电平时, 定时器 y 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT}=f_{CK_INT}/3$)。

- 1) 将定时器 x 配置为主模式, 发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM2/3/4_CR2 寄存器中的 MM_SEL=100)。
- 2) 配置定时器 x 的 OC1REF 波形 (TIM2/3/4_CCMR1 寄存器)。
- 3) 配置定时器 y 以接收来自定时器 x 的输入触发 (TIMy_SMCR 寄存器中的 TS=000)。
- 4) 将定时器 y 配置为门控模式 (TIMy_SMCR 寄存器中的 SMS=101)。
- 5) 通过向 CEN 位 (TIMy_CR1 寄存器) 写入 “1” 使能定时器 y。
- 6) 通过向 CEN 位 (TIM2/3/4_CR1 寄存器) 写入 “1” 启动定时器 x。

注: 计数器 2 的时钟与计数器 1 不同步, 此模式仅影响定时器 y 的计数器使能信号。

图 21-45 使用定时器 x 的 OC1REF 信号对定时器 y 实施门控控制



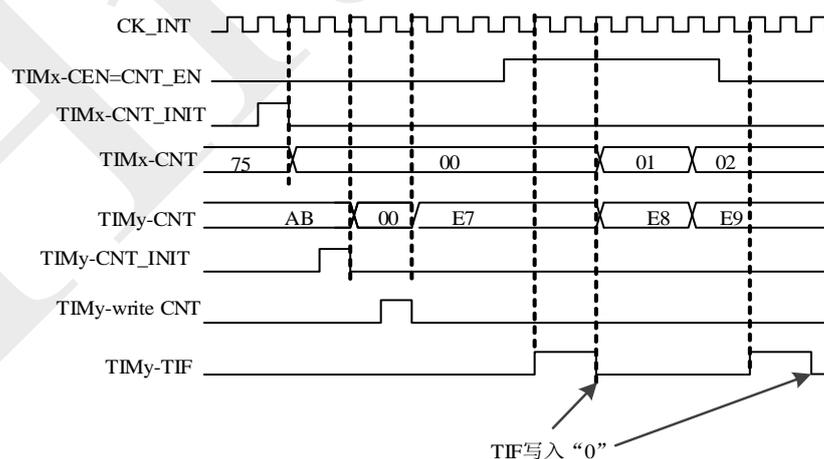
在上图的示例中, 定时器 y 的计数器和预分频器在启动前未进行初始化。因此

从各自的当前值开始计数。启动定时器 x 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIM2/3/4_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，定时器 x 与定时器 y 同步。定时器 x 为主模式，从 0 开始计数。定时器 y 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 TIM2/3/4_CR1 寄存器中通过向 CEN 位写入“0”来禁止定时器 x 时，定时器 y 将停止：

- 1) 将定时器 x 配置为主模式，发送其输出比较 1 参考信号（OC1REF）作为触发输出（TIM2/3/4_CR2 寄存器中的 MM_SEL=100）。
- 2) 配置定时器 x 的 OC1REF 波形（TIM2/3/4_CCMR1 寄存器）。
- 3) 配置定时器 y 以接收来自定时器 x 的输入触发（TIMy_SMCR 寄存器中的 TS=000）。
- 4) 将定时器 y 配置为门控模式（TIMy_SMCR 寄存器中的 SMS=101）。
- 5) 通过向 UG 位（TIM2/3/4_EGR 寄存器）写入“1”复位定时器 x。
- 6) 通过向 UG 位（TIMy_EGR 寄存器）写入“1”复位定时器 y。
- 7) 通过在定时器 y 的计数器（TIMy_CNTL）中写入“0xE7”使定时器 y 初始化为 0xE7。
- 8) 通过向 CEN 位（TIMy_CR1 寄存器）写入“1”使能定时器 y。
- 9) 通过向 CEN 位（TIM2/3/4_CR1 寄存器）写入“1”启动定时器 x。
- 10) 通过向 CEN 位（TIM2/3/4_CR1 寄存器）写入“0”停止定时器 x。

图 21-46 使用定时器 x 的使能信号对定时器 y 实施门控控制



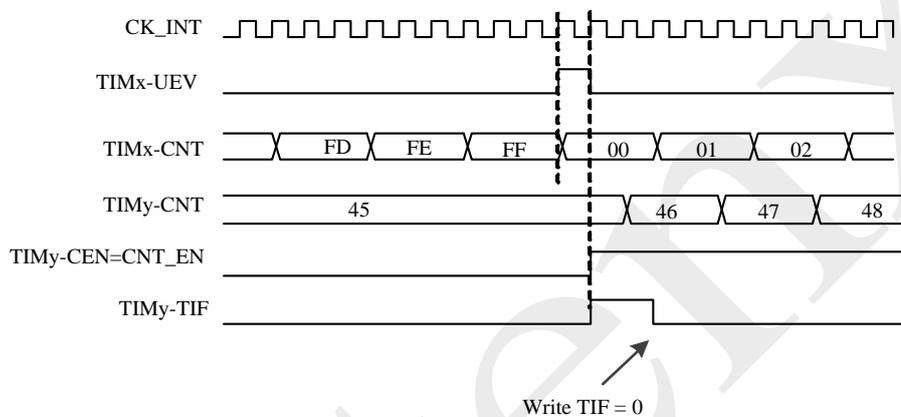
使用一个定时器启动另一个定时器

本例中使用定时器 x 的更新事件使能定时器 y。相关连接图，请参见图 31-46。只要定时器 x 生成更新事件，定时器 y 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 y 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM_CR1 寄存器的 CEN 位写入“0”后停止计数。两个

计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT}=f_{CK_INT}/3$)。

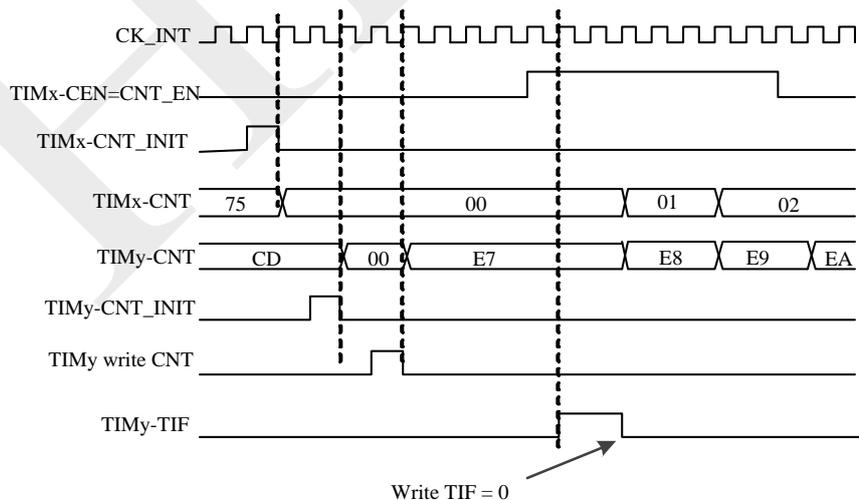
- 1) 将定时器 x 配置为主模式，发送其更新事件 (UEV) 作为触发输出 (TIM2/3/4_CR2 寄存器中的 MM_SEL=010)。
- 2) 配置定时器 x 的周期 (TIM2/3/4_ARR 寄存器)。
- 3) 配置定时器 y 以接收来自定时器 x 的输入触发 (TIM_SMCR 寄存器中的 TS=000)。
- 4) 将定时器 y 配置为触发模式 (TIM_SMCR 寄存器中的 SMS=110)。
- 5) 通过向 CEN 位 (TIM2/3/4_CR1 寄存器) 写入 “1” 启动定时器 x。

图 21-47 使用定时器 x 的更新信号对定时器 y 实施触发控制



如上述示例所示，用户可以在开始计数之前初始化两个计数器。图 21-47 显示了与图 21-48 具有相同配置，只不过处于触发模式 (TIMy_SMCR 寄存器中的 SMS=110) 而非门控模式的计数行为。

图 21-48 使用定时器 x 的使能信号对定时器 y 实施触发控制



使用一个外部触发同步的启动 2 个定时器

本例中，定时器 x 的 TI1 输入出现上升沿时使能定时器 x，使能定时器 x 的同时使能定时器 y。相关连接图，请参见图 23-44。要确保两个计数器对齐，定时器

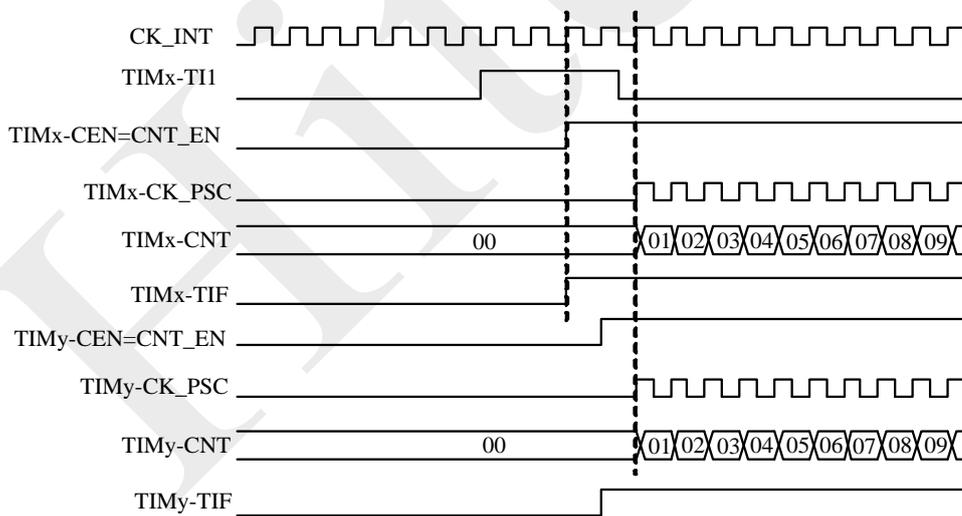
x 必须配置为主/从模式（对应的 TI1 为从，对应的定时器 y 为主）：

- 1) 将定时器 x 配置为主模式，发送其使能信号作为触发输出（TIM2/3/4_CR2 寄存器中的 MM_SEL=001）。
- 2) 将定时器 x 配置为从模式以接收来自 TI1 的输入触发（TIM2/3/4_SMCR 寄存器中的 TS=100）。
- 3) 将定时器 x 配置为触发模式（TIM2/3/4_SMCR 寄存器中的 SMS=110）。
- 4) 通过写入 MSM=1（TIM2/3/4_SMCR 寄存器）将定时器 x 配置为主/从模式。
- 5) 配置定时器 y 以接收来自定时器 x 的输入触发（TIM_SMCR 寄存器中的 TS=000）。
- 6) 将定时器 y 配置为触发模式（TIM_SMCR 寄存器中的 SMS=110）。

当 TI1（定时器 3）出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TI 标志都置 1。

注：本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器（TIM2/3/4_CNT）进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在定时器 x 的 CNT_EN 与 CK_PSC 之间产生了延迟。

图 21-49 使用定时器 x 的 TI1 输入触发定时器 x 和定时器 y



21.3.20 DMA 连续传输模式

TIM2/3/4 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIM2/3/4_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIM2/3/4_DMAR

寄存器都会重定向到其中一个定时器寄存器。

TIM2/3/4_DCR 寄存器中的 DBL[4:0]位设置 DMA 连续传送长度。当对 TIM2/3/4_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIM2/3/4_DCR 寄存器中的 DBA[4:0]位定义 DMA 传送的 DMA 基址（通过 TIM2/3/4_DMAR 地址执行读/写访问时）。DBA 定义为从 TIM2/3/4_CR1 寄存器地址开始计算的偏移量：

示例：

00000: TIM2/3/4_CR1

00001: TIM2/3/4_CR2

00010: TIM2/3/4_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCx 寄存器（x=2）的内容更新为通过 DMA 传输到 CCx 寄存器中的多个半字。

具体操作步骤如下：

- 1) 将相应的 DMA 通道配置如下：
 - DMA 通道外设地址为 DMA 寄存器地址。
 - DMA 通道存储器地址为包含要通过 DMA 传输到 CCx 寄存器的数据的 RAM 缓冲区地址。
 - 要传输的数据量=3（参见下文注释）。
 - 禁止循环模式。
- 2) 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：DBL=3 次传输，DBA=0xE。
- 3) 使能 TIM2/3/4 更新 DMA 请求（TIM2/3/4 DMA/中断使能寄存器（TIM2/3/4_DIER）中的 UDE 位置 1）。
- 4) 使能 TIM2/3/4。
- 5) 使能 DMA 通道。

本例适用于每个 CCx 寄存器只更新一次的情况。如果每个 CCx 寄存器要更新两次，则要传输的数据量应为 4。下面以包含 data1、data2、data3、data4 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CC1，data2 传输到 CC2，；在第二个更新 DMA 请求期间，data3 传输到 CC1，data4 传输到 CC2。

注： 可以将空值写入保留的寄存器中。

21.3.21 调试模式

当微控制器进入调试模式时（Cortex®-M0+内核停止），TIM2/3/4 计数器会根据 *DBG_APB_FZI 寄存器* 的 TIM2/3/4_HOLD 配置位选择继续正常工作或者停止工作。

21.4 TIM2/3/4 寄存器

21.4.1 TIM2/3/4 控制寄存器 1 (TIM2/3/4_CR1)

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

位/位域	名称	描述
31:12	保留	读入 0，写入无效
11	UIFREMAP	UIF状态位重映射 0: UIF状态位不进行重映射 1: UIF 状态位重映射到 TIM2/3/4_CNT 寄存器的 bit31
10	保留	读入 0，写入无效
9:8	CKD[1:0]	时钟分频 此位域指示定时器时钟（CK_INT）频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比 00: $t_{DTS}=t_{CK_INT}$ 01: $t_{DTS}=2 \times t_{CK_INT}$ 10: $t_{DTS}=4 \times t_{CK_INT}$ 11: 保留
7	ARPE	自动重载预装载使能 0: TIM2/3/4_ARR 寄存器不进行缓冲 1: TIM2/3/4_ARR 寄存器进行缓冲
6:5	CMS	中心对齐模式选择 00: 边沿对齐模式。计数器根据方向位（DIR）递增计数或递减计数 01: 中心对齐模式1。计数器交替进行递增计数和递减计数。仅当计数器递减计时，配置为输出的通道（TIM2/3/4_CCMRx 寄存器中的 CCxS=00）的输出比较中

		断标志才置1
		10: 中心对齐模式2。计数器交替进行递增计数和递减计数。 仅当计数器递增计时，配置为输出的通道 (TIM2/3/4_CCMRx寄存器中的CCxS=00)的输出比较中 断标志才置1
		11: 中心对齐模式3。计数器交替进行递增计数和递减计数。 当计数器递增计数或递减计数时，配置为输出的通道 (TIM2/3/4_CCMRx寄存器中的CCxS=00)的输出比较中 断标志都会置1
		<i>注意：只要计数器处于使能状态 (CEN=1)，就不得从边沿对 齐模式切换为中心对齐模式。</i>
4	DIR	计数方向 0: 计数器递增计数 1: 计数器递减计数 <i>注意：当定时器配置为中心对齐模式或编码器模式时，该位为 只读状态。</i>
3	OPM	单脉冲模式 0: 计数器在发生更新事件时不会停止计数 1: 计数器在发生下一更新事件时停止计数 (将CEN位清零)
2	URS	更新请求源 此位由软件置1和清零，用以选择UEV事件源。 0: 使能时，所有以下事件都会产生更新中断或DMA请求。此 类事件包括： - 计数器上溢/下溢 - 将UG位置1 - 通过从模式控制器生成的更新事件 1: 使能时，只有计数器上溢/下溢会生成更新中断或DMA请 求。
1	UDIS	更新禁止 此位由软件置1和清零，用以使能/禁止UEV事件生成。 0: 使能UEV。更新 (UEV) 事件可通过以下事件之一产生： - 计数器上溢/下溢 - 将UG位置1 - 通过从模式控制器生成的更新事件 然后更新影子寄存器的值。 1: 禁止UEV。不会生成更新事件，各影子寄存器的值 (ARR、PSC和CCx) 保持不变。但如果将UG位置1，或者 从模式控制器接收到硬件复位，则会重新初始化计数器和 预分频器。

0

CEN

计数器使能

0: 禁止计数器

1: 使能计数器

注意: 只有事先通过软件将CEN位置1, 才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将CEN位置1。

注意: 在单脉冲模式下, 当发生更新事件时会自动将CEN位清零。

21.4.2 TIM2/3/4 控制寄存器 2 (TIM2/3/4_CR2)

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	
Res																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res								TI1S	MMS[2:0]			CCDS	Res			
								rw	rw			rw				

位/位域	名称	描述
31:8	保留	读入 0, 写入无效
7	TI1S	TI1选择 0: TIM2/3/4_CH1引脚连接到TI1输入 1: TIM2/3/4_CH1、CH2和CH3引脚连接到TI1输入（异或组合）
6:4	MMS[2:0]	主模式选择 这些位可选择主模式下将要发送到从定时器以实现同步的信息（TRGO）。这些位的组合如下： 000: 复位——TIM2/3/4_EGR 寄存器中的 UG 位用作触发输出（TRGO）。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。 001: 使能——计数器使能信号 CEN 用作触发输出（TRGO）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIM3 从模式控制寄存器 (TIM3_SMCR) 中 MSM 位的说明）。 010: 更新——选择更新事件作为触发输出（TRGO）。例如，主定时器可用作从定时器的预分频器。 011: 比较脉冲——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高电平），触发输出都会发送一个

正脉冲。

100: 比较——OC1REF 信号用作触发输出 (TRGO)

101: 比较——OC2REF 信号用作触发输出 (TRGO)

110: 比较——OC3REF 信号用作触发输出 (TRGO)

111: 比较——OC4REF 信号用作触发输出 (TRGO)

注意: 必须先使能从定时器或 ADC 的时钟, 才能从主定时器接收事件; 并且从主定时器接收触发信号时, 不得实时更改从定时器或 ADC 的时钟。

- 3 CCDS 捕获/比较 DMA 选择
0: 发生 CCx 事件时发送 CCxDMA 请求
1: 发生更新事件时发送 CCxDMA 请求
- 2:0 保留 读入 0, 写入无效

21.4.3 TIM2/3/4 从模式控制寄存器 (TIM2/3/4_SMCR)

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
Res										TS[4:3]		Res			SMS[3]
										rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			MSM	TS[2:0]		OCCS	SMS[2:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:22	保留	读入 0, 写入无效
21:20	TS[4:3]	触发选择——位 4:3 请参考 TS 说明——位 2:0
19:17	保留	读入 0, 写入无效
16	SMS[3]	从模式选择——位3 请参见 SMS 说明——位 2:0
15	ETP	外部触发极性 此位可选择将ETR还是ETR用于触发操作 0: ETR未反相, 高电平或上升沿有效 1: ETR反相, 低电平或下降沿有效
14	ECE	外部时钟使能 此位可使能外部时钟模式2。

0: 禁止外部时钟模式2

1: 使能外部时钟模式2。计数器时钟由ETRF信号的任意有效边沿提供。

注意：将ECE位置1与选择外部时钟模式1并将TRGI连接到ETRF (SMS=111且TS=111) 具有相同效果。

外部时钟模式2可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下TRGI不得连接ETRF (TS位不得为111)。

如果同时使能外部时钟模式1和外部时钟模式2，则外部时钟输入为ETRF。

13:12 ETPS

外部触发预分频器

外部触发信号ETRP频率不得超过CK_INT频率的1/4。可通过使能预分频器来降低ETRP频率。这种方法在输入快速外部时钟时非常有用。

00: 预分频器关闭

01: 2分频ETRP频率

10: 4分频ETRP频率

11: 8分频ETRP频率

11:8 ETF[3:0]

外部触发滤波器

此位域可定义ETRP信号的采样频率和适用于ETRP的数字滤波器带宽。数字滤波器由事件计数器组成，每N个连续事件才视为一个有效输出边沿：

0000: 无滤波器，按 f_{DTS} 频率进行采样

0001: $f_{SAMPLING}=f_{CK_INT}$, N=2

0010: $f_{SAMPLING}=f_{CK_INT}$, N=4

0011: $f_{SAMPLING}=f_{CK_INT}$, N=8

0100: $f_{SAMPLING}=f_{DTS}/2$, N=6

0101: $f_{SAMPLING}=f_{DTS}/2$, N=8

0110: $f_{SAMPLING}=f_{DTS}/4$, N=6

0111: $f_{SAMPLING}=f_{DTS}/4$, N=8

1000: $f_{SAMPLING}=f_{DTS}/8$, N=6

1001: $f_{SAMPLING}=f_{DTS}/8$, N=8

1010: $f_{SAMPLING}=f_{DTS}/16$, N=5

1011: $f_{SAMPLING}=f_{DTS}/16$, N=6

1100: $f_{SAMPLING}=f_{DTS}/16$, N=8

1101: $f_{SAMPLING}=f_{DTS}/32$, N=5

1110: $f_{SAMPLING}=f_{DTS}/32$, N=6

1111: $f_{SAMPLING}=f_{DTS}/32$, N=8

7 MSM

主/从模式

0: 不执行任何操作

1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使

当前定时器与其从定时器实现完美同步（通过TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

21,20,6,5,4 TS

TS: 触发选择

此位域可选择将要用于同步计数器的触发输入。

00000: 内部触发0 (ITR0)

00001: 内部触发1 (ITR1)

00010: 内部触发2 (ITR2)

00011: 内部触发3 (ITR3)

00100: TI1边沿检测器 (TI1F_ED)

00101: 滤波后的定时器输入1 (TI1FP1)

00110: 滤波后的定时器输入2 (TI2FP2)

00111: 外部触发输入 (ETRF)

其他: 保留

3 OCCS

OCCS清除信号选择

0: 清除信号连接到COMP1或者COMP2输出，由TIM2/3/4_OR1寄存器的OCCS_CLR位决定。

1: 清除信号连接到 ETRF。

16,2,1,0 SMS[2:0]

从模式选择

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000: 禁止从模式——如果CEN=“1”，预分频器时钟直接由内部时钟提供。

0100: 复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101: 门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110: 触发模式——触发信号TRGI出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111: 外部时钟模式1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000: 组合复位+触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

注意: 如果将TI1F_ED选作触发输入 (TS=“100”), 则不得使用门控模式。实际上, TI1F每次转换时, TI1F_ED都输出1个脉冲, 而门控模式检查的则是触发信号的电平。

注意: 必须先使能接收TRGO或TRGO2信号的从外设 (定时器、ADC等) 的时钟, 才能从主定时器接收事件; 并且

从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

21.4.4 TIM2/3/4 DMA/中断使能寄存器（TIM2/3/4_DIER）

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
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	TDE	Res	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	读入 0，写入无效
14	TDE	触发DMA请求使能 0：禁止触发DMA请求 1：使能触发DMA请求
13	保留	读入 0，写入无效
12	CC4DE	捕获/比较4DMA请求使能 0：禁止CC4 DMA请求。 1：使能 CC4 DMA 请求。
11	CC3DE	捕获/比较3DMA请求使能 0：禁止CC3 DMA请求。 1：使能 CC3DMA 请求。
10	CC2DE	捕获/比较2 DMA请求使能 0：禁止CC2 DMA请求。 1：使能CC2 DMA请求。
9	CC1DE	捕获/比较1 DMA请求使能 0：禁止CC1 DMA请求。 1：使能CC1 DMA请求。
8	UDE	更新DMA请求使能 0：禁止更新DMA请求。 1：使能更新DMA请求。
7	保留	读入0，写入无效
6	TIE	触发中断使能

		0: 禁止触发中断。 1: 使能触发中断。
5	保留	读入0, 写入无效
4	CC4IE	捕获/比较4中断使能 0: 禁止CC4中断 1: 使能 CC4 中断
3	CC3IE	捕获/比较3中断使能 0: 禁止CC3中断 1: 使能CC3中断
2	CC2IE	捕获/比较2中断使能 0: 禁止CC2中断 1: 使能CC2中断
1	CC1IE	捕获/比较1中断使能 0: 禁止CC1中断 1: 使能CC1中断
0	UIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

21.4.5 TIM2/3/4 状态寄存器 (TIM2/3/4_SR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	读入 0, 写入无效
12	CC4OF	捕获/比较4重复捕获标志 请参见CC1OF说明
11	CC3OF	捕获/比较3重复捕获标志 请参见CC1OF说明

10	CC2OF	捕获/比较2重复捕获标志 请参见CC1OF说明
9	CC1OF	捕获/比较1重复捕获标志 仅当CCER.CCx E=1，且将相应通道配置为输入捕获模式时，此标志位才会由硬件置1。通过软件写入“0”可将该位清零。 0: 未检测到重复捕获 1: TIM2/3/4_CCR1寄存器中已捕获到计数器值且CC1IF标志已置1。
8:7	保留	读入0，写入无效
6	TIF	触发中断标志 在除门控模式以外的所有模式下，当使能从模式控制器后在TRGI输入上检测到有效边沿时，该标志将由硬件置1。选择门控模式时，该标志将在计数器启动或停止时置1。但需要通过软件清零。 0: 未发生触发事件 1: 触发中断挂起
5	保留	读入0，写入无效
4	CC4IF	捕获/比较4中断标志 请参见CC1IF说明
3	CC3IF	捕获/比较3中断标志 请参见CC1IF说明
2	CC2IF	捕获/比较2中断标志 请参见CC1IF说明
1	CC1IF	捕获/比较1中断标志 如果通道CC1配置为输出: 当计数器与比较值匹配时，此标志由硬件置1，中心对齐模式下除外（请参见TIM2/3/4控制寄存器1（TIM2/3/4_CR1）中的CMS位说明）。但需要通过软件清零。 0: 不匹配 1: TIM2/3/4_CNT计数器的值与TIM2/3/4_CCR1寄存器的值匹配。当TIM2/3/4_CCR1的值大于TIM2/3/4_ARR的值时，CC1IF位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。 如果通道CC1配置为输入: 此位将在发生捕获事件时由硬件置1。通过软件或读取TIM2/3/4_CCR1寄存器将该位清零。

- 0: 未发生输入捕获事件
- 1: TIM2/3/4_CCR1寄存器中已捕获到计数器值（IC1上已检测到与所选极性匹配的边沿）

0	UIF	<p>更新中断标志</p> <p>该位在发生更新事件时通过硬件置1。但需要通过软件清零。</p> <p>0: 未发生更新。</p> <p>1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置1:</p> <ul style="list-style-type: none"> - 上溢或下溢并且当TIM2/3/4_CR1寄存器中UDIS=0时。 - TIM2/3/4_CR1寄存器中的URS=0且UDIS=0, 并且由软件使用TIM2/3/4_EGR寄存器中的UG位重新初始化CNT时。 - TIM2/3/4_CR1寄存器中的URS=0且UDIS=0, 并且CNT由触发事件重新初始化时（参见同步控制寄存器说明）。
---	-----	---

21.4.6 TIM2/3/4 事件产生寄存器 (TIM2/3/4_EGR)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w				w	w	w

位/位域	名称	描述
31:7	保留	读入 0, 写入无效
6	TG	<p>触发生成</p> <p>此位由软件置1以生成事件, 并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: TIM2/3/4_SR寄存器中的TIF标志置1。使能后可发生相关中断或DMA传输事件。</p>
5	保留	读入 0, 写入无效
4	CC4G	<p>捕获/比较 4 生成</p> <p>请参见 CC1G 说明</p>
3	CC3G	<p>捕获/比较 3 生成</p> <p>请参见 CC1G 说明</p>

2	CC2G	捕获/比较 2 生成 请参见 CC1G 说明
1	CC1G	<p>CC1G: 捕获/比较1生成 此位由软件置1以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作 1: 通道1上生成捕获/比较事件:</p> <p>如果通道CC1配置为输出: 使能时，CC1IF标志置1并发送相应的中断或DMA请求。</p> <p>如果通道CC1配置为输入: TIM_CCR1寄存器中将捕获到计数器当前值。使能时，CC1IF标志置1并发送相应的中断或DMA请求。如果CC1IF标志已为高电平，CC1OF标志将置1。</p>
0	UG	<p>UG: 更新生成 该位可通过软件置1，并由硬件自动清零。</p> <p>0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或DIR=0（递增计数），计数器将清零；如果DIR=1（递减计数），计数器将使用自动重载值（TIM2/3/4_ARR）。</p>

21.4.7 TIM2/3/4 捕获/比较模式寄存器 1_输出比较模式 (TIM2/3/4_CCMR1)

偏移地址: 0x18

复位值: 0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31				30				29				28				27				26				25				24				23				22				21				20				19				18				17				16			
Res												OC2M[3]				Res												OC1M[3]																																			
												rw																rw																																			
15				14				13				12				11				10				9				8				7				6				5				4				3				2				1				0			
OC2CE				OC2M[2:0]				OC2PE				OC2FE				CC2S[1:0]				OC1CE				OC1M[2:0]				OC1PE				OC1FE				CC1S[1:0]																											
rw				rw				rw				rw				rw				rw				rw				rw				rw				rw																											

位/位域	名称	描述
31:25	保留	读入 0，写入无效
24	OC2M[3]	输出比较 2 模式——位 3 请参考 OC2M 说明——位 2:0
23:17	保留	读入 0，写入无效

16	OC1M[3]	输出比较 1 模式——位 3 请参考 OC1M 说明——位 2:0
15	OC2CE	输出比较 2 清零使能
24,14,13,12	OC2M[3:0]	输出比较2模式 请参考OC1M说明
11	OC2PE	输出比较2预装载使能
10	OC2FE	输出比较2快速使能
8	保留	读入 0，写入无效
9:8	CC2S[1:0]	捕获/比较2选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC2通道配置为输出 01: CC2通道配置为输入，IC2映射到TI2上 10: CC2通道配置为输入，IC2映射到TI1上 11: CC2通道配置为输入，IC2映射到TRC上。此模式仅在通过TS位（TIM2/3/4_SMCR寄存器）选择内部触发输入时有效 <i>注意：仅当通道关闭时（TIM2/3/4_CCER中的CC2E=0），才可向CC2S位写入数据。</i>
7	OC1CE	输出比较1清零使能 0: OC1Ref不受ETRF输入影响 1: ETRF输入上检测到高电平时，OC1Ref立即清零
16,6,5,4	OC1M[3:0]	输出比较1模式 这些位定义提供OC1和OC1N的输出参考信号OC1REF的行为。OC1REF为高电平有效，而OC1和OC1N的有效电平则取决于CC1P位和CC1NP位。 0000: 冻结——输出比较寄存器TIM2/3/4_CCR1与计数器TIM2/3/4_CNT进行比较不会对输出造成任何影响。（该模式用于生成时基）。 0001: 将通道1设置为匹配时输出有效电平。当计数器TIM2/3/4_CNT与捕获/比较寄存器1(TIM2/3/4_CCR1)匹配时，OC1REF信号强制变为有效电平。 0010: 将通道1设置为匹配时输出无效电平。当计数器TIM2/3/4_CNT与捕获/比较寄存器1(TIM2/3/4_CCR1)匹配时，OC1REF信号强制变为无效电平。 0011: 翻转——TIM2/3/4_CNT=TIM2/3/4_CCR1时，OC1REF发生翻转。

- 0100: 强制变为无效电平——OC1REF强制变为无效电平。
- 0101: 强制变为有效电平——OC1REF强制变为有效电平。
- 0110: PWM模式1——在递增计数模式下，只要 $TIM2/3/4_CNT < TIM2/3/4_CCR1$ ，通道1便为有效状态，否则为无效状态。在递减计数模式下，只要 $TIM_CNT > TIM2/3/4_CCR1$ ，通道1便为无效状态 (OC1REF=0)，否则为有效状态 (OC1REF=1)。
- 0111: PWM模式2——在递增计数模式下，只要 $TIM2/3/4_CNT < TIM2/3/4_CCR1$ ，通道1便为无效状态，否则为有效状态。在递减计数模式下，只要 $TIM2/3/4_CNT > TIM2/3/4_CCR1$ ，通道1便为有效状态，否则为无效状态。
- 1000: 可再触发OPM模式1——在递增计数模式下，通道为有效状态，直至（在TRGI信号上）检测到触发事件。然后，在PWM模式1下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在TRGI信号上）检测到触发事件。然后，在PWM模式1下进行比较，通道会在下一次更新时再次变为无效状态。
- 1001: 可再触发OPM模式2——在递增计数模式下，通道为无效状态，直至（在TRGI信号上）检测到触发事件。然后，在PWM模式2下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在TRGI信号上）检测到触发事件。然后，在PWM模式2下进行比较，通道会在下一次更新时再次变为有效状态。
- 1010: 保留。
- 1011: 保留。
- 1100: 组合PWM模式1——OC1REF与在PWM模式1下的行为相同。OC1REFC是OC1REF和OC2REF的逻辑或运算结果。
- 1101: 组合PWM模式2——OC1REF与在PWM模式2下的行为相同。OC1REFC是OC1REF和OC2REF的逻辑与运算结果。
- 1110: 不对称PWM模式1——OC1REF与在PWM模式1下的行为相同。计数器递增计数时，OC1REFC输出OC1REF；计数器递减计数时，OC1REFC输出OC2REF。
- 1111: 不对称PWM模式2——OC1REF与在PWM模式2下的行为相同。计数器递增计数时，OC1REFC输出OC1REF；计数器递减计数时，OC1REFC输出OC2REF。
- 注意：在PWM模式1或PWM模式2下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF电平才会发生更改。*

- 3 OC1PE 输出比较 1 预装载使能
- 0: 禁止与 TIM2/3/4_CCR1 相关的预装载寄存器。可随时向 TIM2/3/4_CCR1 写入数据，写入后将立即使用新值。
- 1: 使能与 TIM2/3/4_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIM2/3/4_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。
- 注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIM2/3/4_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。*
- 2 OC1FE 输出比较1快速使能
- 此位用于加快触发输入事件对CC输出的影响。
- 0: 即使触发开启，CC1也将根据计数器和CC1值正常工作。触发输入出现边沿时，激活CC1输出的最短延迟时间为5个时钟周期。
- 1: 触发输入上出现有效边沿相当于CC1输出上的比较匹配。随后，无论比较结果如何，OC都设置为比较电平。采样触发输入和激活CC1输出的延迟时间缩短为3个时钟周期。仅当通道配置为PWM1或PWM2模式时，OCFE才会起作用。
- 1:0 CC1S 捕获/比较1选择
- 此位域定义通道方向（输入/输出）以及所使用的输入。
- 00: CC1通道配置为输出。
- 01: CC1通道配置为输入，IC1映射到TI1上
- 10: CC1通道配置为输入，IC1映射到TI2上
- 11: CC1通道配置为输入，IC1映射到TRC上。此模式仅在通过TS位（TIM2/3/4_SMCR寄存器）选择内部触发输入时有效
- 注意：只当通道关闭时（TIM2/3/4_CCER中的CC1E=0），才可向CC1S位写入数据。*

21.4.8 TIM2/3/4 捕获/比较模式寄存器 1_输入捕获模式（TIM2/3/4_CCMR1）

偏移地址：0x18

复位值：0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
Res																							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
IC2F[3:0]				IC2PSC[1:0]				CC2S[1:0]				IC1F[3:0]				IC1PSC[1:0]				CC1S[1:0]			
rw				rw				rw				rw				rw							

位/位域	名称	描述
31:16	保留	读入 0，写入无效

15:12	IC2F[3:0]	输入捕获 2 滤波器
11:10	IC2PSC[1:0]	输入捕获 2 预分频器
9:8	CC2S[1:0]	<p>捕获/比较2选择</p> <p>此位域定义通道方向（输入/输出）以及所使用的输入。</p> <p>00: CC2通道配置为输出。</p> <p>01: CC2通道配置为输入，IC2映射到TI2上</p> <p>10: CC2通道配置为输入，IC2映射到TI1上</p> <p>11: CC2通道配置为输入，IC2映射到TRC上。此模式仅在通过TS位（TIM_SMCR寄存器）选择内部触发输入时有效</p> <p><i>注意：仅当通道关闭时（TIM2/3/4_CCER中的CC2E=0），才可向CC2S位写入数据。</i></p>
7:4	IC1F[3:0]	<p>输入捕获1滤波器</p> <p>此位域可定义TI1输入的采样频率和适用于TI1的数字滤波器带宽。数字滤波器由事件计数器组成，每N个连续事件才视为一个有效输出边沿：</p> <p>0000: 无滤波器，按f_{DTS}频率进行采样</p> <p>0001: $f_{SAMPLING}=f_{CK_INT}$, N=2</p> <p>0010: $f_{SAMPLING}=f_{CK_INT}$, N=4</p> <p>0011: $f_{SAMPLING}=f_{CK_INT}$, N=8</p> <p>0100: $f_{SAMPLING}=f_{DTS}/2$, N=6</p> <p>0101: $f_{SAMPLING}=f_{DTS}/2$, N=8</p> <p>0110: $f_{SAMPLING}=f_{DTS}/4$, N=6</p> <p>0111: $f_{SAMPLING}=f_{DTS}/4$, N=8</p> <p>1000: $f_{SAMPLING}=f_{DTS}/8$, N=6</p> <p>1001: $f_{SAMPLING}=f_{DTS}/8$, N=8</p> <p>1010: $f_{SAMPLING}=f_{DTS}/16$, N=5</p> <p>1011: $f_{SAMPLING}=f_{DTS}/16$, N=6</p> <p>1100: $f_{SAMPLING}=f_{DTS}/16$, N=8</p> <p>1101: $f_{SAMPLING}=f_{DTS}/32$, N=5</p> <p>1110: $f_{SAMPLING}=f_{DTS}/32$, N=6</p> <p>1111: $f_{SAMPLING}=f_{DTS}/32$, N=8</p>
3:2	IC1PSC[1:0]	<p>输入捕获 1 预分频器</p> <p>此位域定义 CC1 输入(IC1)的预分频比。</p> <p>只要 CC1E=0（TIM2/3/4_CCER 寄存器），预分频器便立即复位。</p> <p>00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获</p> <p>01: 每发生 2 个事件便执行一次捕获</p> <p>10: 每发生 4 个事件便执行一次捕获</p> <p>11: 每发生 8 个事件便执行一次捕获</p>

1:0 CC1S 捕获/比较1选择
 此位域定义通道方向（输入/输出）以及所使用的输入。
 00: CC1通道配置为输出
 01: CC1通道配置为输入，IC1映射到TI1上
 10: CC1通道配置为输入，IC1映射到TI2上
 11: CC1通道配置为输入，IC1映射到TRC上。此模式仅在通过TS位（TIM3_SMCR寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIM2/3/4_CCER中的CC1E=0），才可向CC1S位写入数据。

21.4.9 TIM2/3/4 捕获/比较模式寄存器 2_输出比较模式（TIM2/3/4_CCMR2）

偏移地址：0x1C

复位值：0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31				30				29				28				27				26				25				24				23				22				21				20				19				18				17				16			
Res																OC4M[3]				Res																OC3M[3]																											
rw																rw				rw																rw																											
15				14				13				12				11				10				9				8				7				6				5				4				3				2				1				0			
OC4CE				OC4M[2:0]				OC4PE				OC4FE				CC4S[1:0]				OC3CE				OC3M[2:0]				OC3PE				OC3FE				CC3S[1:0]																											
rw				rw				rw				rw				rw				rw				rw				rw				rw				rw																											

位/位域	名称	描述
31:25	保留	读入0，写入无效
24	OC4M[3]	输出比较4模式——位3 请参考OC4M说明——位2:0
23:17	保留	读入0，写入无效
16	OC3M[3]	输出比较3模式——位3 请参考OC3M说明——位2:0
15	OC4CE	输出比较4清零使能
24,14,13,12	OC4M[3:0]	输出比较4模式 请参考OC3M说明
11	OC4PE	输出比较4预装载使能
10	OC4FE	输出比较4快速使能
9:8	CC4S[1:0]	捕获/比较4选择

		请参考CC1S说明
7	OC3CE	输出比较3清零使能 请参考OC3CE说明
16,6,5,4	OC3M[3:0]	输出比较3模式 请参考OC1M说明
3	OC3PE	输出比较3预装载使能 请参考OC1PE说明
2	OC3FE	输出比较3快速使能 请参考OC3PE说明
1:0	CC3S	捕获/比较1选择 请参考CC3S说明

21.4.10 TIM2/3/4 捕获/比较模式寄存器 2_输入捕获模式 (TIM2/3/4_CCMR2)

偏移地址: 0x1C

复位值: 0x0000 0000

输入捕获模式和输出比较模式复用寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw				rw		rw		rw				rw		rw	

位/位域	名称	描述
31:16	保留	读入0, 写入无效
15:12	IC4F[3:0]	输入捕获4滤波器 请参考IC1F说明
11:10	IC4PSC[1:0]	输入捕获4预分频器 请参考IC4PSC说明
9:8	CC4S[1:0]	捕获/比较4选择 请参考CC4S说明
7:4	IC3F[3:0]	输入捕获3滤波器 请参考IC1F说明
3:2	IC3PSC[1:0]	输入捕获3预分频器 请参考IC1PSC说明

1:0 CC3S 捕获/比较3选择
请参考CC1S说明

21.4.11 TIM2/3/4 捕获/比较使能寄存器 1 (TIM2/3/4_CCER)

偏移地址: 0x20

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw												

位/位域	名称	描述
31:16	保留	读入 0, 写入无效
15	CC4NP	捕获/比较 4 互补输出极性 请参见 CC1NP 说明
14	保留	读入 0, 写入无效
13	CC4P	捕获/比较 4 输出极性 请参见 CC1P 说明
12	CC4E	捕获/比较 4 输出使能 请参见 CC1E 说明
11	CC3NP	捕获/比较 3 互补输出极性 请参见 CC1NP 说明
10	保留	读入 0, 写入无效读入 0, 写入无效
9	CC3P	捕获/比较 3 输出极性 请参见 CC1P 说明
8	CC3E	捕获/比较 3 输出使能 请参见 CC1E 说明
7	CC2NP	捕获/比较 2 互补输出极性 请参见 CC1NP 说明
6	保留	读入 0, 写入无效

5	CC2P	捕获/比较 2 输出极性 请参见 CC1P 说明
4	CC2E	捕获/比较 2 输出使能 请参见 CC1E 说明
3	CC1NP	捕获/比较1互补输出极性 CC1通道配置为输出: 在这种情况下, CC1NP必须保持清零。 CC1通道配置为输入: 此位与CC1P配合使用, 用以定义TI1FP1/TI2FP1的极性。请参见CC1P说明。
2	保留	读入0, 写入无效
1	CC1P	捕获/比较1输出极性 CC1通道配置为输出: 0: OC1高电平有效 1: OC1低电平有效 CC1通道配置为输入: CC1NP/CC1P位可针对触发或捕获操作选择TI1FP1和TI2FP1的极性。 00: 未反相/上升沿触发 电路对TIxFP1上升沿敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1未反相(在门控模式或编码器模式下执行触发操作)。 01: 反相/下降沿触发 电路对TIxFP1下降沿敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1反相(在门控模式或编码器模式下执行触发操作)。 10: 保留, 不使用此配置。 11: 未反相/上升沿和下降沿均触发 电路对TIxFP1上升沿和下降沿都敏感(在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1未反相(在门控模式下执行触发操作)。编码器模式下不得使用此配置。
0	CC1E	捕获/比较1输出使能 CC1通道配置为输出: 0: 关闭——OC1未激活 1: 开启——在相应输出引脚上输出OC1信号 CC1通道配置为输入: 此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器1 (TIM2/3/4_CCR1) 中。 0: 禁止捕获

1: 使能捕获
21.4.12 TIM2/3/4 计数值寄存器 (TIM2/3/4_CNT)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIFCPY		Res													
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
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

位/位域	名称	描述
31	UIFCPY	UIF副本 该位是 TIM2/3/4_ISR 寄存器中 UIF 位的只读副本。如果 TIM2/3/4_CR1 中的 UIFREMAP 位复位, 则位 31 保留, 读为 0。
30:16	保留	读入 0, 写入无效
15:0	CNT[15:0]	低计数器值

21.4.13 TIM2/3/4 预分频值寄存器 (TIM2/3/4_PSC)

偏移地址: 0x28

复位值: 0x0000 0000

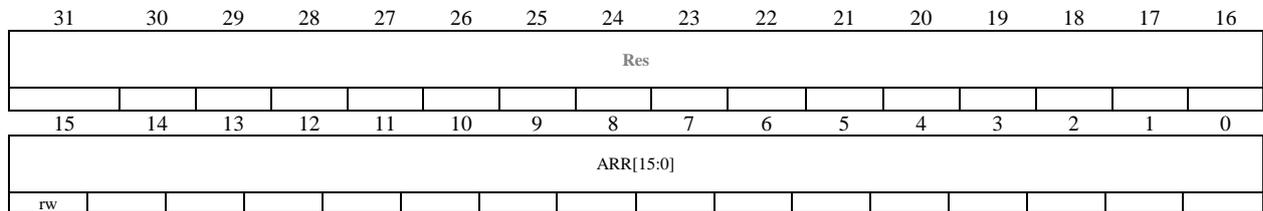
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:16	保留	读入 0, 写入无效
15:0	PSC [15:0]	预分频器值 计数器时钟频率CK_CNT等于 $f_{CK_PSC} / (PSC[15:0]+1)$ 。 PSC包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

21.4.14 TIM2/3/4 自动重载值寄存器 (TIM2/3/4_ARR)

偏移地址：0x2C

复位值：0x0000 FFFF

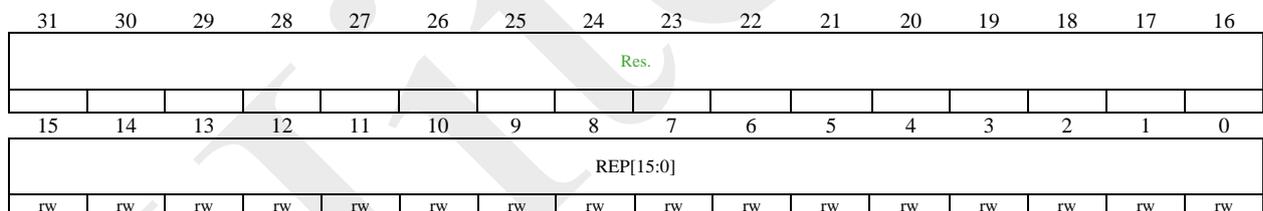


位/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	ARR[15:0]	低自动重载值 ARR为要装载到实际自动重载寄存器的值。 当自动重载值为空时，计数器不工作。

21.4.15 TIM2/3/4 重复计数器寄存器(TIM2/3/4_RCR)

偏移地址：0x30

复位值：0x0000 0000

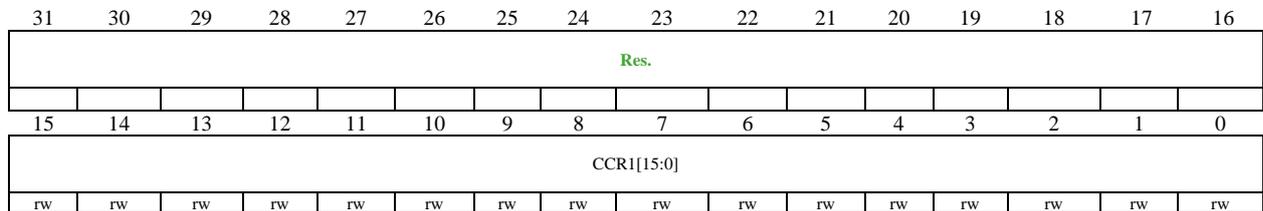


位/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	REP[15:0]	重复计数器值 使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向有效寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 与REP_CNT相关的减计数器每次计数到0时，都将生成一个更新事件并且计数器从REP值重新开始计数。由于只有生成重复更新事件U_RC时，REP_CNT才会重载REP值，因此在生成下一重复更新事件之前，无论向TIM2/3/4_RCR寄存器写入何值都无影响。 这意味着PWM模式下(REP+1)相当于： 边沿对齐模式下的PWM周期数。 中心对齐模式下的PWM半周期数。

21.4.16 TIM2/3/4 捕获/比较寄存器 1 (TIM2/3/4_CCR1)

偏移地址：0x34

复位值：0x0000 0000

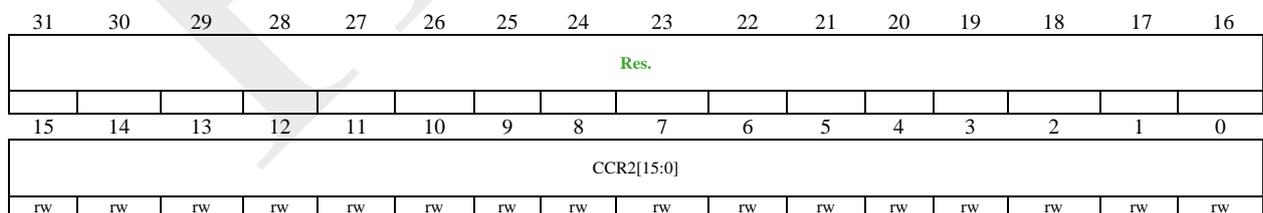


位/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	CCR1[15:0]	捕获/比较1值的低16位 如果通道CC1配置为输出： CCR1是捕获/比较寄存器1的预装载值。 如果没有通过TIM2/3/4_CCMR1寄存器中的OC1PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器1）。 实际捕获/比较寄存器中包含要与计数器TIM2/3/4_CNT进行比较并在OC1输出上发出信号的值。 如果通道CC1配置为输入： CCR1为上一个输入捕获1事件（IC1）发生时的计数器值。 当配置为霍尔传感器接口时，该位数值不可读取。

21.4.17 TIM2/3/4 捕获/比较寄存器 2 (TIM2/3/4_CCR2)

偏移地址：0x38

复位值：0x0000 0000



位/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	CCR2[15:0]	捕获/比较2值的低16位 如果通道CC2配置为输出： CC2是捕获/比较寄存器2的预装载值。 如果没有通过TIM2/3/4_CCMR2寄存器中的OC2PE位来使能预

装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器2）。

实际捕获/比较寄存器中包含要与计数器TIM2/3/4_CNT进行比较并在OC2输出上发出信号的值。

如果通道CC2配置为输入：

CC2为上一个输入捕获2事件（IC2）发生时的计数数值。

当配置为霍尔传感器接口时，该位数值不可读取。

21.4.18 TIM2/3/4 捕获/比较寄存器 3 (TIM2/3/4_CCR3)

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
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/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	CCR3[15:0]	捕获/比较3值的低16位 如果通道CC3配置为输出： CC3是捕获/比较寄存器1的预装载值。 如果没有通过TIM2/3/4_CCMR3寄存器中的OC3PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器1）。 实际捕获/比较寄存器中包含要与计数器TIM2/3/4_CNT进行比较并在OC3输出上发出信号的值。 如果通道CC1配置为输入： CC3为上一个输入捕获1事件（IC3）发生时的计数数值。 当配置为霍尔传感器接口时，该位数值不可读取。

21.4.19 TIM2/3/4 捕获/比较寄存器 4 (TIM2/3/4_CCR4)

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
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/位域	名称	描述
31:16	保留	读入 0，写入无效
15:0	CCR4[15:0]	捕获/比较4值的低16位 如果通道CC4配置为输出： CC4是捕获/比较寄存器4的预装载值。 如果没有通过TIM2/3/4_CCMR4寄存器中的OC4PE位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器2）。 实际捕获/比较寄存器中包含要与计数器TIM2/3/4_CNT进行比较并在OC4输出上发出信号的值。 如果通道CC4配置为输入： CC4为上一个输入捕获2事件（IC4）发生时的计数器值。

21.4.20 TIM2/3/4 DMA 控制寄存器 (TIM2/3/4_DCR)

偏移地址：0x48

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL[4:0]					Res.			DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	DBL[4:0]	DMA连续传送长度 该5位向量定义了DMA的传送次数（当对TIM2/3/4_DMAR寄存器进行读或写时，定时器进行一次连续传送）。 00000：1次传送， 00001：2次传送， 00010：3次传送， ... 10001：18次传送。
7:5	保留	必须保持复位值
4:0	DBA [4:0]	DMA基址 该5位向量定义DMA传输的基址（通过TIM2/3/4_DMAR地址进行读/写访问时）。DBA定义为从TIM2/3/4_CR1寄存器地址开始计算的偏移量。

示例：

00000: TIM2/3/4_CR1,

00001: TIM2/3/4_CR2,

00010: TIM2/3/4_SMCR,

...

示例：以下面的传送为例：DBL=7 次传送且 DBA=TIM2/3/4_CR1。这种情况下将向/从自TIM2/3/4_CR1地址开始的7个寄存器传输数据。

21.4.21 TIM2/3/4 DMA 全传输地址寄存器 (TIM2/3/4_DMAR)

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rw	rw	rw	rw	rw	rw	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
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	DMAB [31:0]	<p>DMA 连续传送寄存器</p> <p>对 DMA 寄存器执行读或写操作将访问位于如下地址的寄存器： (TIM2/3/4_CR1 地址) + (DBA+DMA 索引) × 4 其中 TIM2/3/4_CR1 地址为控制寄存器 1 的地址，DBA 为 TIM2/3/4_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL (TIM2/3/4_DCR 寄存器 中配置的 DBL) 之间。</p>

21.4.22 TIM2/3/4 配置寄存器 1 (TIM2/3/4_OR1)

偏移地址：0x54

复位值：0x00000000

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

位/位域	名称	描述
31:1	保留	读入 0，写入无效

- 0 IOCREF_CLR ocref_clr源选择信号
 0: COMP1的输出连接到OCREF_CLR输入
 1: COMP2 的输出连接到 OCREF_CLR 输入

21.4.23 TIM2/3/4 轮换功能寄存器 1 (TIM2/3/4_AF1)

偏移地址: 0x60

复位值: 0x00000000

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

位/位域	名称	描述
31:18	保留	读入 0, 写入无效
17:14	ETRSEL[3:0]	ETR源选择信号 0000: GPIO连接到ETR输入 0001: COMP1输出连接到ETR输入 0010: COMP2输出连接到ETR输入 0011: LSE 其他值: 保留
13:0	保留	读入 0, 写入无效

21.4.24 TIM2/3 定时器输入选择寄存器 (TIM2/3_TISEL)

偏移地址: 0x68

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				TI2SEL[3:0]				Res				TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	读入 0, 写入无效
11:8	TI2SEL[3:0]	TI2的输入源选择信号 0000: TIM2/3_CH2输入

- 0001: COMP2输出
其他: 保留
- 7:4 保留 读入 0, 写入无效
- 3:0 TI1SEL[3:0] TI1的输入选择信号
0000: TIM2/3_CH1输入
0001: COMP1输出
其他: 保留

21.4.25 TIM4 定时器输入选择寄存器 (TIM4_TISEL)

偏移地址: 0x68

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				TI2SEL[3:0]				Res.				TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	读入 0, 写入无效
11:8	TI2SEL[3:0]	TI2的输入源选择信号 0000: TIM4_CH2输入 0001: COMP2输出 0010: LSE时钟输入 0011: HSE的时钟输入(是否分频由 HSE_DIV 设置) 0100: MCO的时钟输入 0101: LSI的时钟输入 0110: MSI的时钟输入 0111: RTC的计数时钟输入 1000: RTC的唤醒信号输入 1001: TIM2_IC2输出 1010: TIM3_IC2输出 其他: 保留
7:4	保留	读入 0, 写入无效
3:0	TI1SEL[3:0]	TI1的输入选择信号 0000: TIM4_CH1输入 0001: COMP1输出 0010: TIM2_IC1输出

0011: TIM3_IC1输出
其他: 保留

HITENX

22 基本定时器 (TIM5/6/7)

22.1 简介

基本定时器 TIM5、TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器不仅可用作通用定时器以生成时基，还可以专门用于驱动数模转换器 (DAC)。实际上，此类定时器内部连接到 DAC 并能够通过其触发输出驱动 DAC。

这些定时器彼此完全独立，不共享任何资源。

22.2 TIM5/6/7 主要特性

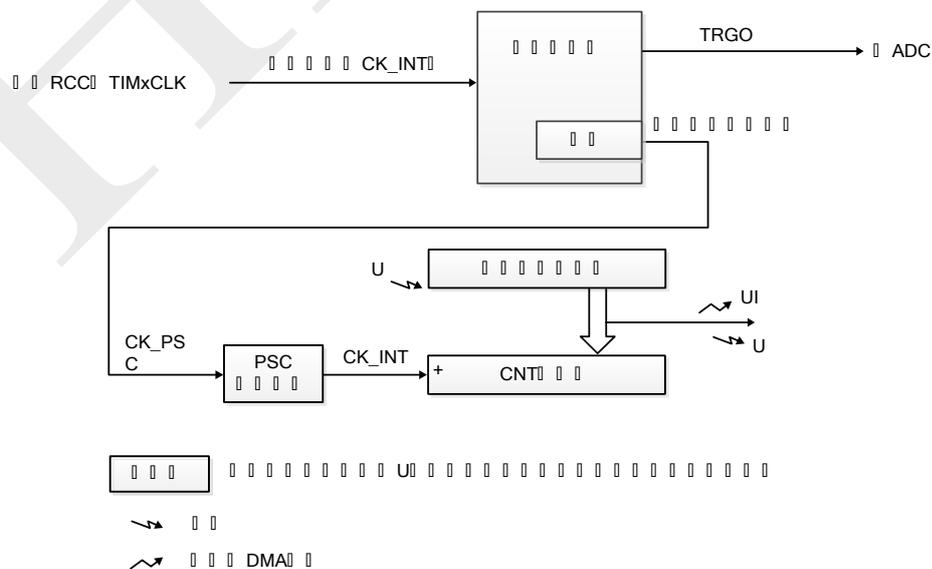
本模块主要功能特性如下：

- 16 位自动装载计数器
- 16 位可编程预分频器(可实时修改),计数时钟频率的分频系数为 1~65535 之间的任意数值
- 同步电路触发 ADC
- 产生中断/DMA 产生更新事件：计数溢出

与 CPU 接口：

- 挂接在 APB1 总线上
- 访问模块内部未定义寄存器空间时，读数据为 0，写数据无效

图 22-1 基本定时器框图



22.3 TIM5/6/7 功能说明

22.3.1 时基单元

可编程定时器的主要模块由一个 16 位及其相关的自动重载寄存器组成。计数器可递增计数。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。

预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 [TIM5/6/7 控制寄存器 \(TIM_CR1\)](#) 中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值 (或者在递减计数时达到下溢值) 并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

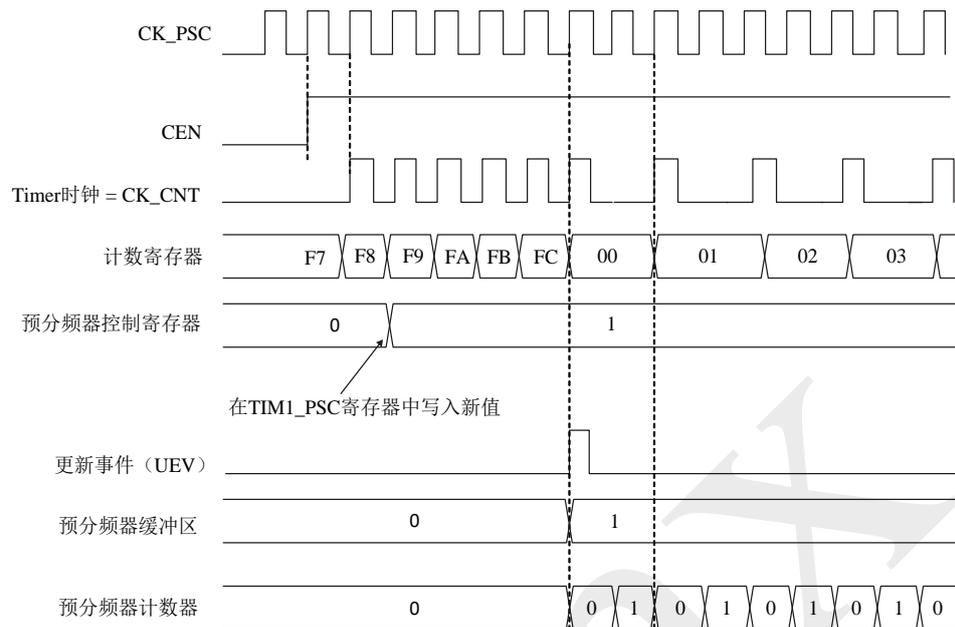
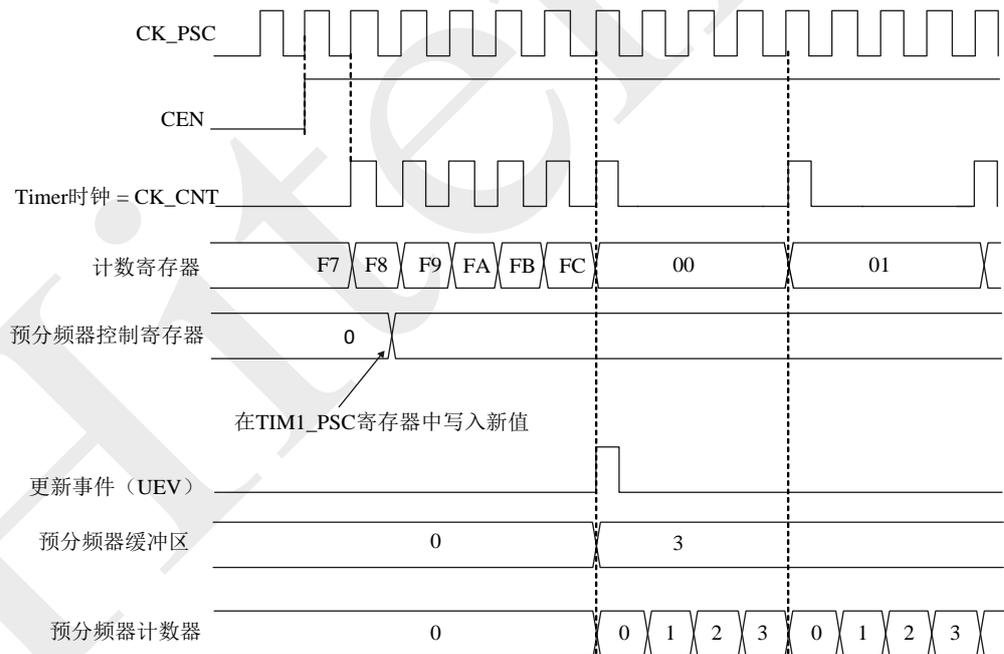
计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器 (有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明)。

注意：实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位寄存器 ([TIM5/6/7 预分频器寄存器 \(TIM_PSC\)](#)) 所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

下图以一些示例说明在预分频比实时变化时计数器的行为：

图 22-2 预分频器分频由 1 变为 2 时的计数器时序图

图 22-3 预分频器分频由 1 变为 4 时的计数器时序图


22.3.2 计数模式

计数器从 0 计数到自动重载值（*TIM5/6/7 自动重载值寄存器 (TIM_ARR)* 的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 *TIM5/6/7 事件产生寄存器 (TIM_EGR)* 中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 *TIM5/6/7 控制寄存器 (TIM_CR1)* 中的 UDIS 位置 1 可禁止 UEV

事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM5/6/7 状态寄存器（TIMx_SR）中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值（TIMx_ARR）更新自动重载影子寄存器

图 22-4 计数器时序图，1 分频内部时钟

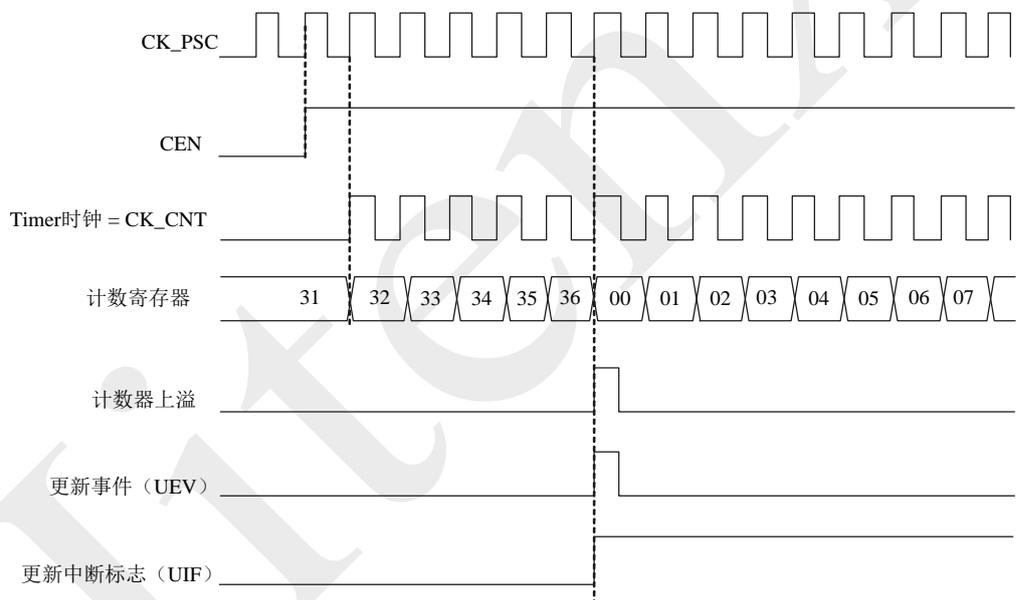


图 22-5 计数器时序图，N 分频内部时钟

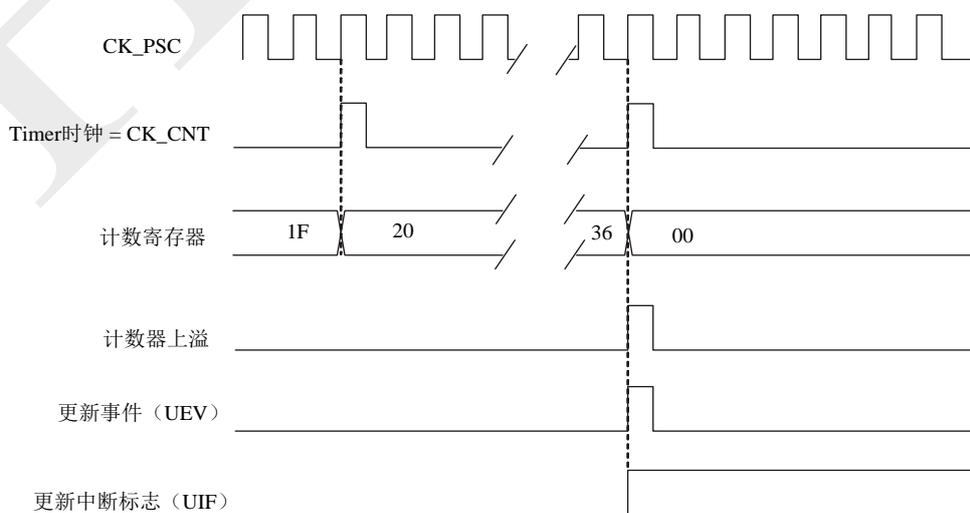
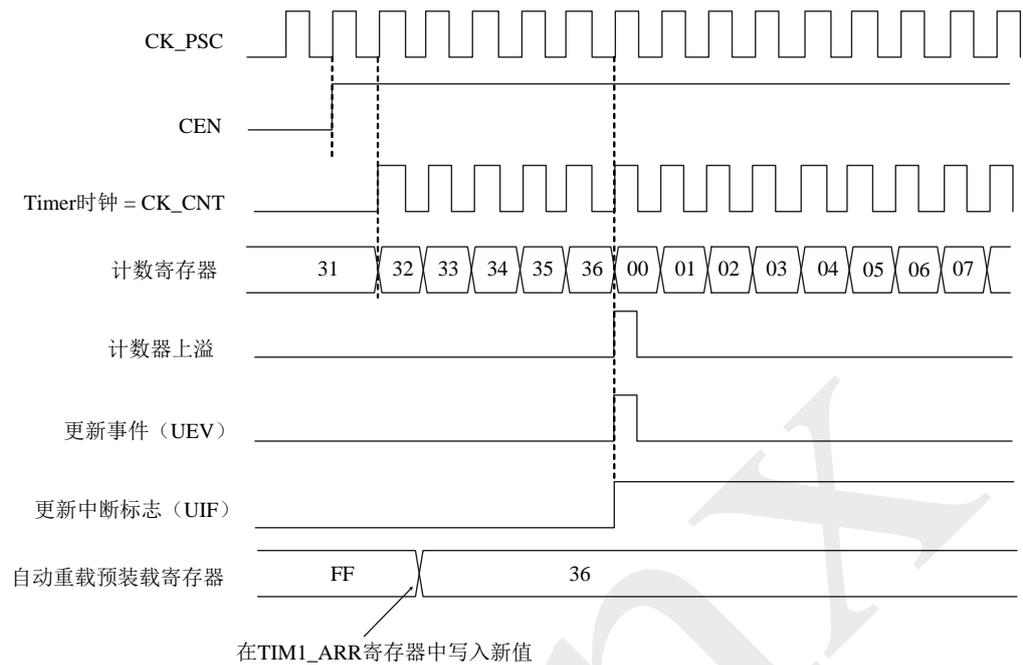
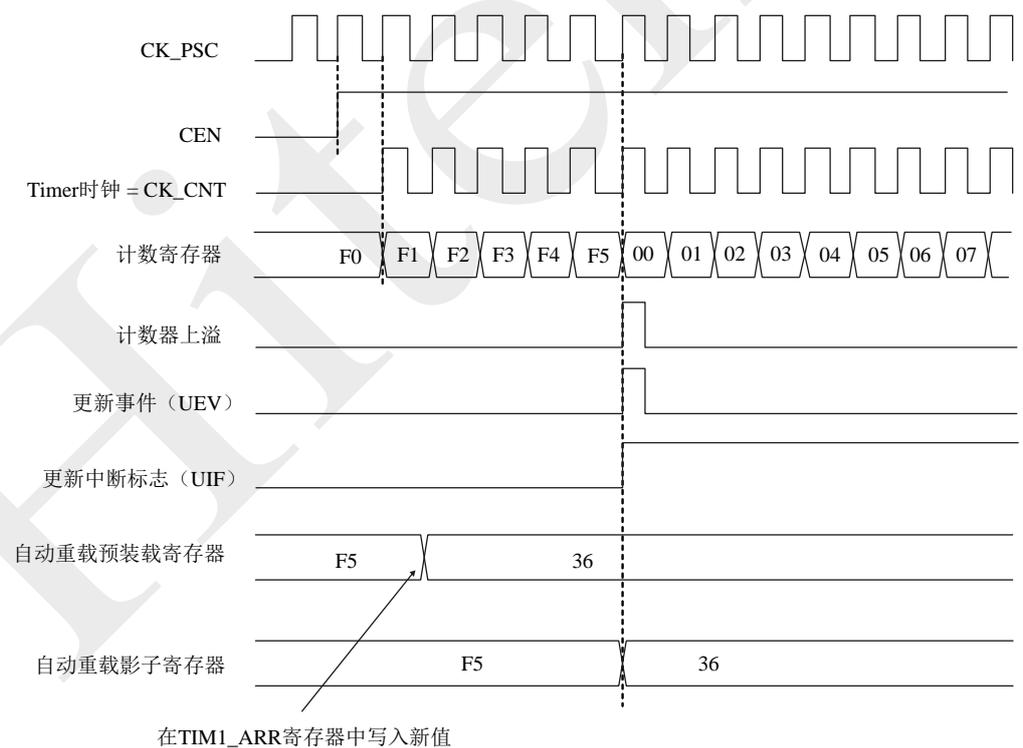


图 22-6 计数器时序图, ARPE=0 时更新事件 (未预装载)

图 22-7 计数器时序图, ARPE=1 时更新事件 (已预装载)


22.3.3 UIF 位重映射

TIM5/6/7 控制寄存器 (TIM_CR1) 中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到 *TIM5/6/7 计数值寄存器 (TIM_CNT)* 的位 31 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务 (计数器读) 和中断 (更新中断) 之间共享处理时产生竞争条件, 从而简化角速度的

计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

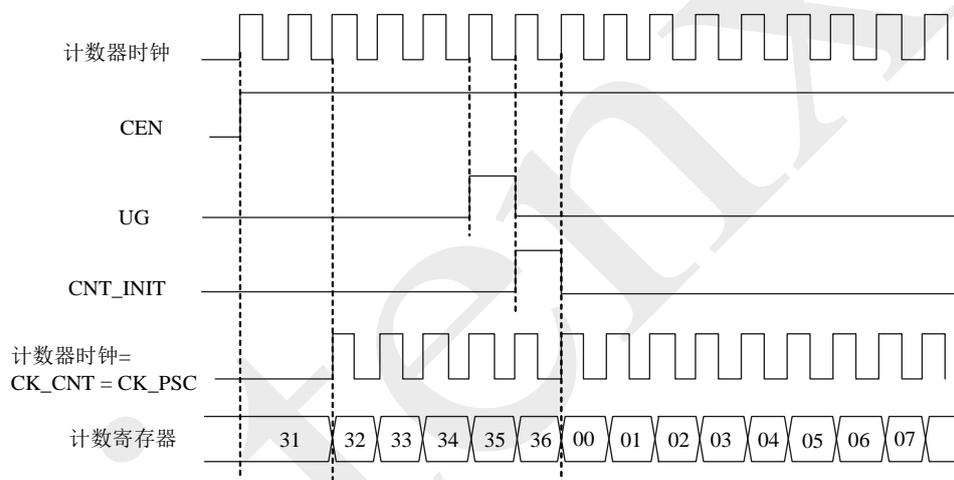
22.3.4 时钟源

TIM5、TIM6 和 TIM7 支持内部时钟源。

TIMx_CR1 寄存器中) 和 UG 位 (TIM_EGR 寄存器中) 为实际控制位, 并且只能通过软件进行更改 (UG 除外, 仍自动清零)。当对 CEN 位写入 1 时, 预分频器的时钟就由内部时钟 CK_INIT 提供。

下图展示了不进行分频时的控制电路和计数器的工作情况

图 22-8 正常模式下的控制电路, 1 分频内部时钟



22.3.5 调试模式

当微控制器进入调试模式时 (Cortex®-M0+ 内核停止), TIM 计数器会根据 **DBG_APB_FZ1** 的 TIMx_HOLD 配置位选择继续正常工作或者停止工作。

22.4 TIM5/6/7 寄存器

22.4.1 TIM5/6/7 控制寄存器 1 (TIM_CR1)

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

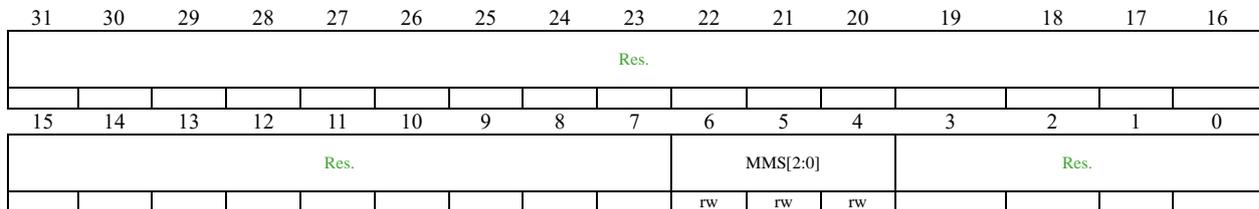
位/位域	名称	描述
31:12	保留	必须保持复位值

11	UIFREMAP	<p>UIF状态位重映射</p> <p>0: UIF状态位不进行重映射</p> <p>1: UIF状态位重映射到TIMx_CNT寄存器的bit31</p>
10:8	保留	必须保持复位值
7	ARPE	<p>自动重载预装载使能</p> <p>0: TIMx_ARR寄存器不进行缓冲</p> <p>1: TIMx_ARR寄存器进行缓冲</p>
6:4	保留	必须保持复位值
3	OPM	<p>单脉冲模式</p> <p>0: 计数器在发生更新事件时不会停止计数</p> <p>1: 计数器在发生下一更新事件时停止计数（将CEN位清零）</p>
2	URS	<p>更新请求源</p> <p>此位由软件置1和清零，用以选择UEV事件源。</p> <p>0: 使能时，所有以下事件都会产生更新中断或DMA请求。此类事件包括：</p> <ul style="list-style-type: none"> - 计数器上溢 - 将UG位置1 - 通过从模式控制器生成的更新事件 <p>1: 使能时，只有计数器上溢会生成更新中断或DMA请求。</p>
1	UDIS	<p>更新禁止</p> <p>此位由软件置1和清零，用以使能/禁止UEV事件生成。</p> <p>0: 使能UEV。更新（UEV）事件可通过以下事件之一产生：</p> <ul style="list-style-type: none"> - 计数器上溢 - 将UG位置1 - 通过从模式控制器生成的更新事件 <p>然后更新影子寄存器的值。</p> <p>1: 禁止UEV。不会生成更新事件，各影子寄存器的值（ARR和PSC）保持不变。但如果将UG位置1，或者从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。</p>
0	CEN	<p>计数器使能</p> <p>0: 禁止计数器</p> <p>1: 使能计数器</p> <p><i>注意：只有事先通过软件将CEN位置1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将CEN位置1。在单脉冲模式下，当发生更新事件时会自动将CEN位清零。</i></p>

22.4.2 TIM5/6/7 控制寄存器 2 (TIM_CR2)

偏移地址：0x04

复位值：0x0000 0000

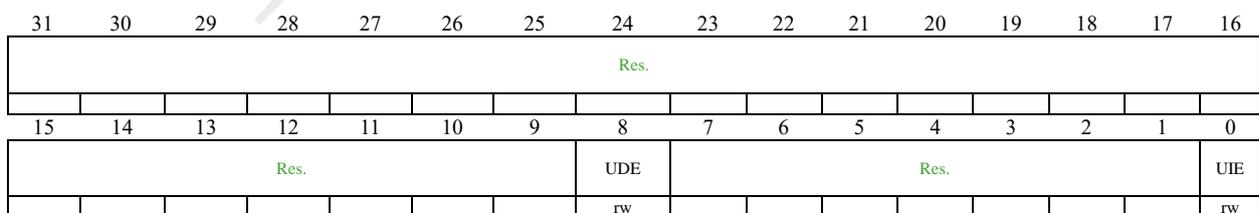


位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	MMS [2:0]	主模式选择 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下： 000: 复位——TIM_EGR寄存器中的UG位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式)，则TRGO上的信号相比实际复位会有延迟。 001: 使能——计数器使能信号CEN用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由CEN控制位与门控模式下的触发输入的逻辑或运算组合而成。 010: 更新——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器
3:0	保留	必须保持复位值

22.4.3 TIM5/6/7 DMA/中断使能寄存器 (TIM_DIER)

偏移地址：0x0C

复位值：0x0000 0000



位/位域	名称	描述
31:10	保留	必须保持复位值
8	UDE	更新DMA请求使能

- 0: 禁止更新DMA请求。
- 1: 使能更新DMA请求。

7:1 保留 必须保持复位值

0 UIE 更新中断使能

- 0: 禁止更新中断。
- 1: 使能更新中断。

22.4.4 **TIM5/6/7 状态寄存器 (TIM_SR)**

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														UIF	
														rc_w0	

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UIF	更新中断标志 该位在发生更新事件时通过硬件置1。但需要通过软件清零。 0: 未发生更新。 1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置1: <ul style="list-style-type: none"> - 上溢或下溢并且当TIMx_CR1寄存器中UDIS=0时; - TIMx_CR1寄存器中的URS=0且UDIS=0, 并且由软件使用TIMx_EGR寄存器中的UG位重新初始化CNT时; - TIMx_CR1寄存器中的URS=0且UDIS=0, 并且CNT由触发事件重新初始化时 (参见 TIM1 从模式控制寄存器 (TIM1_SMCR))。

22.4.5 **TIM5/6/7 事件产生寄存器 (TIM_EGR)**

偏移地址: 0x14

复位值: 0x0000 0000

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

							w	w	w	w	w	w	w	w	w
--	--	--	--	--	--	--	---	---	---	---	---	---	---	---	---

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UG	更新生成 该位可通过软件置1，并由硬件自动清零。 0: 不执行任何操作 1: 重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或DIR=0（递增计数），计数器将清零；如果DIR=1（递减计数），计数器将使用自动重载值（TIMx_ARR）

22.4.6 TIM5/6/7 计数值寄存器 (TIM_CNT)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIFCPY	Res.														
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	UIFCPY	UIF副本 该位是TIMx_SR寄存器中UIF位的只读副本。如果TIMx_CR1中的UIFREMAP位复位，则位31保留，读为0。
30:16	保留	必须保持复位值
15:0	CNT[15:0]	计数器值

22.4.7 TIM5/6/7 预分频值寄存器 (TIM_PSC)

偏移地址: 0x28

复位值: 0x0000 0000

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{CK_PSC} / (PSC[15:0] + 1)$ 。 PSC 包含每次发生更新事件时要装载到有效预分频器寄存器的值。

22.4.8 TIM5/6/7 自动重载值寄存器 (TIM_ARR)

偏移地址: 0x2C

复位值: 0x0000 FFFF

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ARR[15:0]	自动重载值 ARR为要装载到实际自动重载寄存器的值。 当自动重载值为空时，计数器不工作。

23 低功耗定时器（LPTIM）

23.1 简介

LPTIM 是一个 16 位定时器，此模块的开发是为了从降低功耗的发展中受益。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式下保持运行状态。即便是没有内部时钟源，LPTIM 也能运行，鉴于这一特点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，在这种功能模式下系统功耗极低。

LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

23.2 LPTIM 主要特性

本模块主要功能特性如下：

- 16 位向上自动装载计数器
- 3 位计数器预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟：
 - 内部时钟源：LSE、LSI、HSI 或 APB1 时钟
 - LPTIM 输入的外部时钟源（在没有 LP 振荡器运行的情况下工作，由脉冲计数器应用场景下使用）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选软件/硬件输入触发
- 3 位数字滤波时钟分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可编程数字干扰滤波器
- 可配置输出：脉冲和 PWM
- 可配置 I/O 极性
- 正交/非交编码器模式

与 CPU 接口:

- 挂接在 APB1 总线上，只支持 32 位访问
- 访问模块内部未定义寄存器空间时，读数据为 0，写数据无效

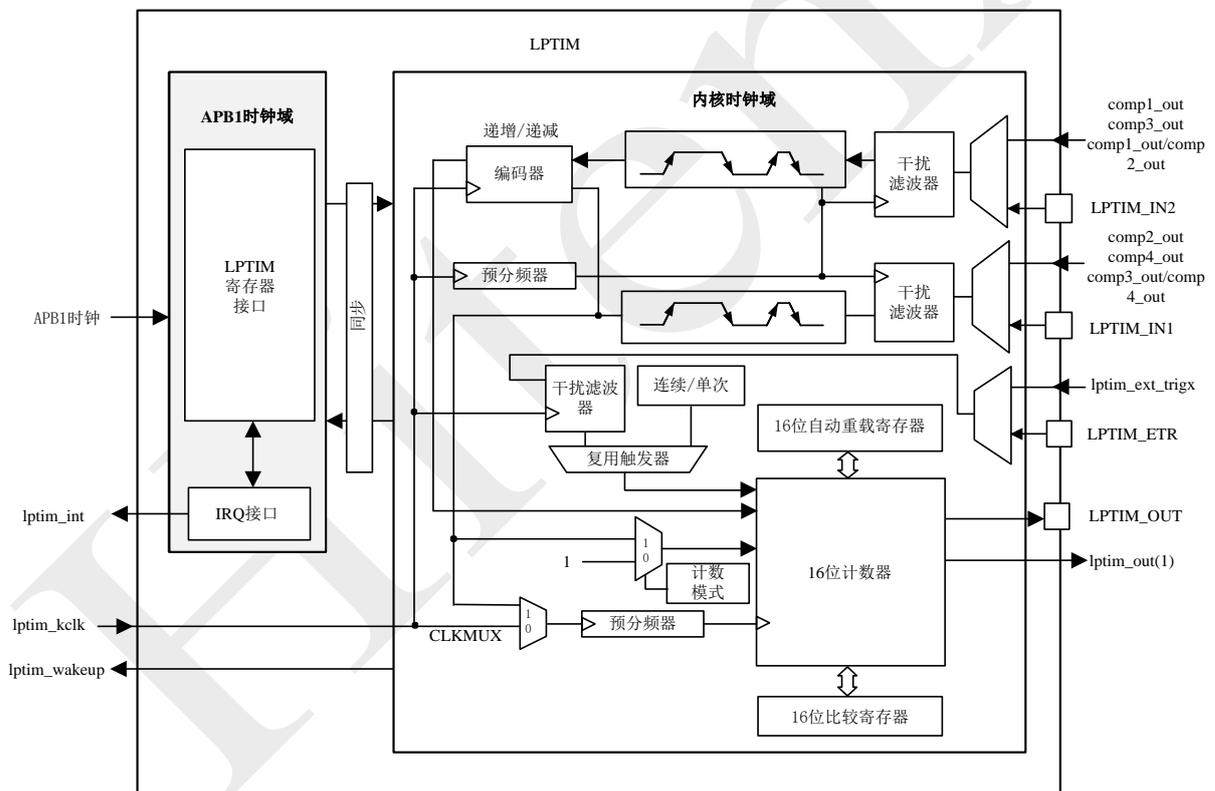
时钟与复位:

- 双时钟域
- 异步复位，低电平有效

23.3 LPTIM 功能说明

23.3.1 LPTIM 框图

图 23-1 低功耗定时器框图



1: lptim_out 是内部输出信号，可以连到 LPTIM 内部外设。

23.3.2 LPTIM 引脚和内部信号

下表分别提供了 LPTIM 引脚和内部信号的列表:

表 23-1 LPTIM 输入输出引脚

信号名	信号类型	描述
LPTIM_IN1	数字输入	LPTIM 输入 1 通道，从 mux0 输入的 GPIO 引脚
LPTIM_IN2	数字输入	LPTIM 输入 2 通道，从 mux0 输入的 GPIO 引脚

信号名	信号类型	描述
LPTIM_ETR	数字输入	LPTIM 由外部触发的 GPIO 引脚
LPTIM_OUT	数字输出	LPTIM 向外输出的 GPIO 引脚

表 23-2 LPTIM 内部信号

信号名	信号类型	描述
lptim_kclk	数字输入	LPTIM 内核时钟
lptim_wakeup	数字输出	LPTIM 唤醒事件
lptim_int	数字输出	LPTIM 全局中断
lptim_in1_mux1	数字输入	COMP1 输出信号
lptim_in1_mux2	数字输入	COMP3 输出信号
lptim_in1_mux3	数字输入	COMP1 和 COMP2 信号的或门输出
lptim_in2_mux1	数字输入	COMP2 输出信号
lptim_in2_mux2	数字输入	COMP4 输出信号
lptim_in2_mux3	数字输入	COMP3 和 COMP4 信号的或门输出
lptim_out	数字输出	LPTIM 计数器输出
lptim_ext_trigx	数字输入	LPTIM 外部触发输入

23.3.3 LPTIM 触发映射

下表详细介绍 LPTIM 的外部触发连接：

表 23-3 LPTIM 外部触发连接

信号名	描述
lptim_ext_trig0	GPIO ETR 输入信号
lptim_ext_trig1	RTC_ALARM
lptim_ext_trig2	保留
lptim_ext_trig3	保留
lptim_ext_trig4	SAC COMP3 输出信号
lptim_ext_trig5	SAC COMP4 输出信号
lptim_ext_trig6	SAC COMP1 输出信号
lptim_ext_trig7	SAC COMP2 输出信号

23.3.4 LPTIM 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过 [外设异步时钟配置寄存器 \(RCC_CCIPR\)](#) 在 APB1、LSI、LSE 或 HSI 时钟源中进行选择。此外，LPTIM 还可通过注入到其外部 IN1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在下述两种可能配置中的其中一种配置下运行：

- 第一种配置：LPTIM 通过外部信号提供计数源，但同时通过 APB1 或 LSE、LSI 和 HSI 等任何其他内置振荡器为 LPTIM 提供内部时钟信号。
- 第二种配置：LPTIM 仅由外部时钟源通过外部 IN1 提供计数源和工作时

钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 *LPTIM 配置寄存器 (LPTIM_CFGR)* 中的 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）；在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的 4 倍。

23.3.5 干扰滤波器

LPTIM 输入（外部或内部）由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。在激活数字滤波器之前，首先应向 LPTIM 提供内部时钟源，这是保证滤波器正常工作的必要条件。

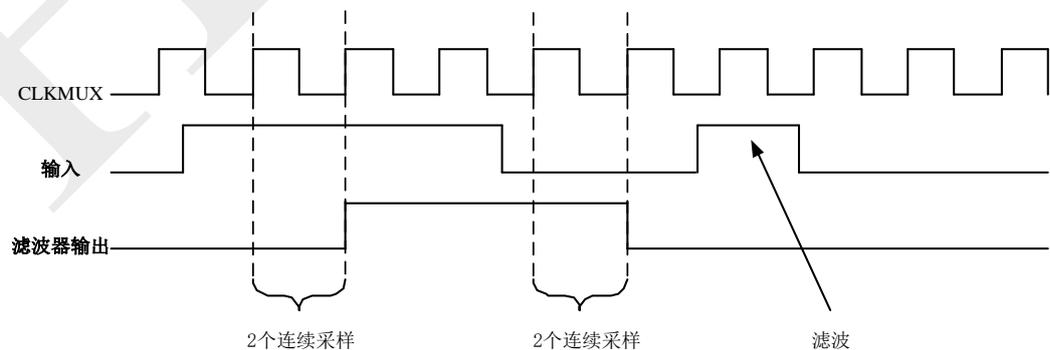
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入，数字滤波器的滤波脉冲数由 CKFLT[1:0] 位控制；数字滤波时钟可通过对 *LPTIM 配置寄存器 (LPTIM_CFGR)* 中 INFLT_PSC[2:0] 位配置。
- 第二组数字滤波器保护 LPTIM 内部触发输入，数字滤波器的滤波脉冲数由 TRIGFLT[1:0] 位控制。

注意： 数字滤波器的滤波脉冲数以组为单位进行控制，无法单独配置同一组内各个数字滤波器的滤波脉冲数。

滤波器的滤波脉冲数会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效切换。下图给出了 2 周期连续采样时，干扰滤波器行为的示例。

图 23-2 干扰滤波器时序图



注意： 不提供内部时钟信号时，必须通过将 CKFLT[1:0] 和 TRIGFLT[1:0] 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。

23.3.6 预分频器

LPTIM16 位计数器前面要有一个可配置的二次幂预分频器。预分频器的分频比由 *LPTIM 配置寄存器 (LPTIM_CFGR)* 中 PRESC[2:0] 的 3 位域进行控制。下表列出了所有分频比：

表 23-4 预分频器分频比

编程	分频系数
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

23.3.7 触发多路复用器

LPTIM 计数器可通过软件启动，也可以在 8 个触发输入之一上检测到有效边沿后启动。LPTIM 配置寄存器 (LPTIM_CFGR) 中 TRIGEN[1:0] 用于确定 LPTIM 触发源：

- TRIGEN[1:0] 等于“00”时，LPTIM 计数器会在通过软件将 *LPTIM 控制寄存器 (LPTIM_CR)* 中 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后延时两个 APB1 时钟和两个内核时钟后启动。
- TRIGEN[1:0] 的其余三个可能值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。

TRIGEN[1:0] 不等于“00”时，LPTIM 配置寄存器 (LPTIM_CFGR) 中 TRIGSEL[2:0] 用于选择 8 个触发输入的其中之一来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个核时钟周期，定时器才能开始运行。如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注意： 必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。

23.3.8 工作模式

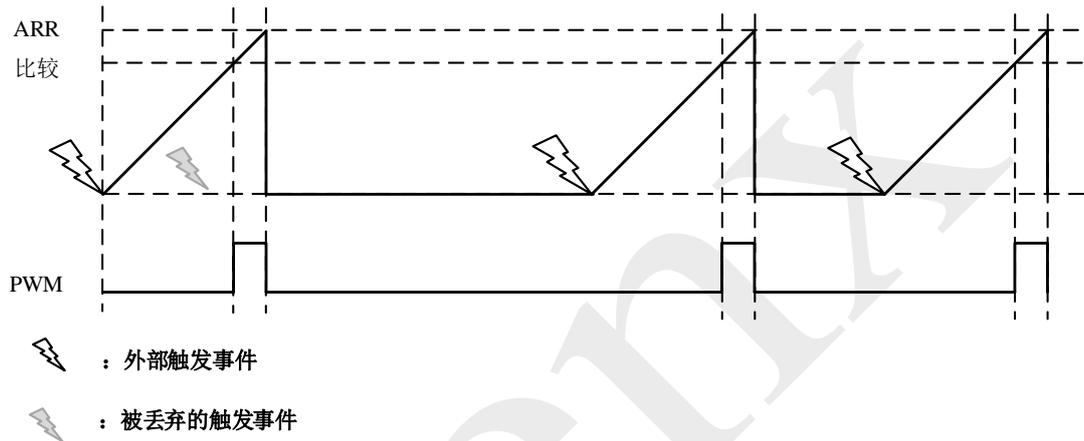
LPTIM 支持以下两种工作模式：

- 单触发模式：定时器由触发事件启动，当达到 ARR 值时停止。
- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止；

23.3.8.1 单触发模式

要使能单触发计数，必须将 SNGSTRT 位置 1。新的触发事件将重新启动定时器。从计数器启动到计数器达到 ARR，这段时间内发生的任何触发事件均将被丢弃。选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如下图所示：

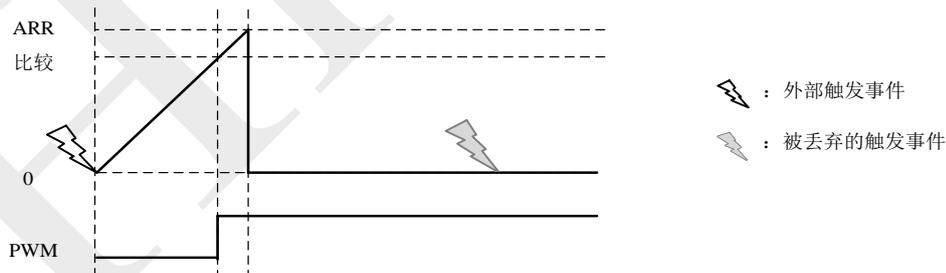
图 23-3 LPTIM 输出波形，单次计数模式配置



置一单触发模式：

当 LPTIM_CFGR 寄存器中的 WAVE 位域置 1 时，将激活置一单触发模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如下图：

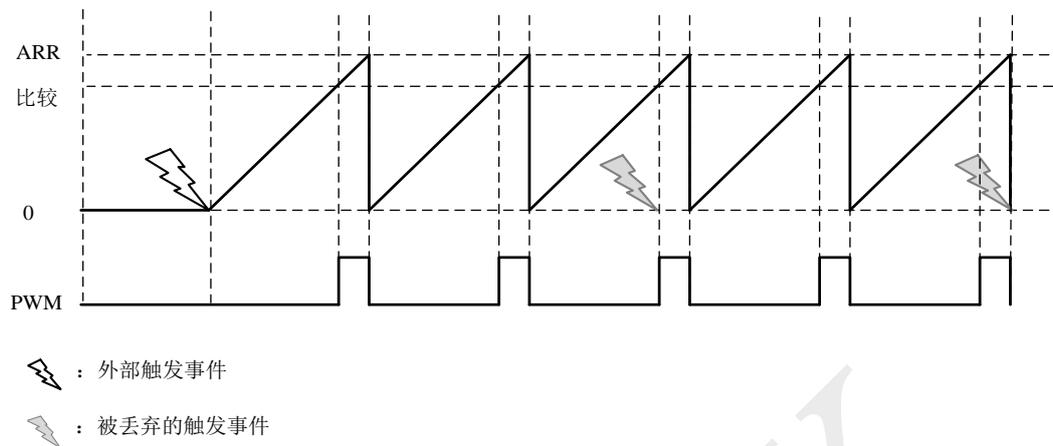
图 23-4 LPTIM 输出波形，单次计数模式配置且激活置一单触发模式（WAVE 位置 1）



若通过软件启动（TRIGEN[1:0]=“00”），将 SNGSTRT 置 1 会使计数器进行单触发计数。

23.3.8.2 连续模式

要使能连续计数，必须将 CNTSTRT 位置 1。若选择外部触发，则在 CNTSTRT 置 1 后到达的外部触发事件将启动计数器进行连续计数；任何后续的外部触发事件都将被丢弃。若通过软件启动（TRIGEN[1:0]=“00”），将 CNTSTRT 置 1 会使计数器开始连续计数，如下图所示：

图 23-5 LPTIM 输出波形、连续计数模式配置


SNGSTRT 和 CNTSTRT 位只能在定时器使能时 (ENABLE 位置 1) 置 1。可以“实时”从单触发模式切换为连续模式。如果之前选择的是连续模式，则将 SNGSTRT 置 1 会使 LPTIM 切换为单触发模式。计数器 (激活时) 将在达到 ARR 后立即停止。如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器 (激活时) 将在达到 ARR 后立即重新启动。

23.3.9 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。第一个触发事件将启动定时器，任何后续的触发事件将复位计数器，且定时器将重新启动。可实现低功耗超时功能，超时值对应于比较值；如果在预期的时间帧内未发生触发事件，MCU 将由比较匹配事件唤醒。

23.3.10 生成波形

LPTIM 自动重载寄存器 (LPTIM_ARR) 和 *LPTIM 比较寄存器 (LPTIM_CMP)* 两个 16 位寄存器，可在 LPTIM 输出上生成多个不同的波形，模式如下：

- **PWM 模式：**若 LPTIM_CMP 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出置 1；若 LPTIM_ARR 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出复位。
- **单脉冲模式：**对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位。
- **置一单触发模式：**除输出保持最后一个信号电平外 (取决于配置的输出极性)，输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIM_ARR 寄存器的值严格大于 LPTIM_CMP 寄存器的值。

LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，

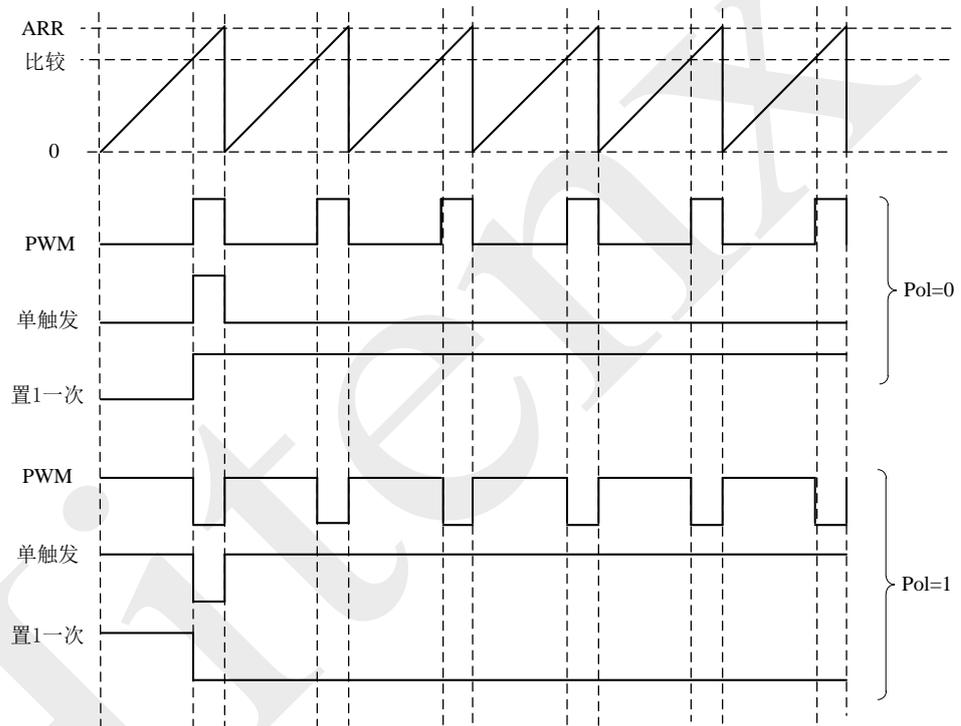
具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。

- 若将 WAVE 位置 1，则会强制 LPTIM 生成置一单触发模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。下图给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 23-6 生成波形



23.3.11 寄存器更新

LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器在 APB1 总线写操作后会立即更新，若定时器已启动，也可在当前周期结束时更新。

LPTIM 配置寄存器 (LPTIM_CFGR) 中 PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式：

- 当 PRELOAD 位复位为 0 时，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在写访问后立即更新。
- 当 PRELOAD 位置 1 时，若定时器已启动，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在当前周期结束时更新。

APB1 总线和 LPTIM 内核逻辑使用的时钟不同，因此在 APB1 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向

这些寄存器执行其他写操作。LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的写操作已完成。向 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

23.3.12 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。*LPTIM 配置寄存器 (LPTIM_CFGR)* 中 CKSEL 位和 COUNTMODE 位用于选择内外源更新计数器。若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0]位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL=0: LPTIM 由内部时钟源提供时钟
 - COUNTMODE=0

将 LPTIM 配置为由内部时钟源进行计时，并将 LPTIM 计数器配置为根据每个内部时钟脉冲进行更新。
 - COUNTMODE=1

LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟采样。因此，为了不丢失任何事件，外部 Input1 信号变化的频率不能超过提供给 LPTIM 的内部时钟的频率，故而不能对 lptim_kclk 分频 (PRESC[2:0]=000)。
- CKSEL=1: LPTIM 由外部时钟源提供时钟

COUNTMODE 值不相关。在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。此配置适合未使能任何内置振荡器的工作模式；对于这种配置，LPTIM 计数器可以在 Input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

由于注入到 LPTIM 外部 Input1 的信号也可用于 LPTIM 的时钟，计数器递增计数前存在一些初始延时（使能 LPTIM 后）；能够确定 LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 LPTIM 后）。

23.3.13 定时器使能

*LPTIM_CR 控制寄存器*中的 ENABLE 位用于使能或者禁止 LPTIM。将 ENABLE 位置 1 后，需要延迟两个核时钟周期，才能真正使能 LPTIM。LPTIM_CFGR 和 LPTIM_IER 寄存器必须在禁止 LPTIM 后才能修改。

23.3.14 定时器计数器复位

LPTIM 实现了两种 LPTIM_CNT 寄存器的复位机制：

- 同步复位机制：

同步复位由 *LPTIM 控制寄存器 (LPTIM_CR)* 中的 COUNTRST 位控制。在将 COUNTRST 位域置 1 后，复位信号在 LPTIM 内核时钟域中传播。因此，重要的是要注意，要在经历几个 LPTIM 内核逻辑时钟脉冲之后再考虑复位。这将使 LPTIM 计数器在复位触发和生效之间额外计数几个脉冲。由于 COUNTRST 位位于 APB1 时钟域中，并且 LPTIM 计数器位于 LPTIM 内核时钟域中，因此当将 1 写入到 COUNTRST 位时，内核时钟需要 3 个时钟周期的延迟用以同步由 APB1 时钟域发出的复位信号。

- 异步复位机制：

异步复位由 *LPTIM 控制寄存器 (LPTIM_CR)* 中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都会将其内容复位为零。应在不提供 LPTIM 内核时钟的时间范围内触发异步复位。例如，当 LPTIM Input1 引脚为外部时钟输入引脚时，只有当足够保证 LPTIM Input1 不会发生反转时，才应用异步复位。

为了实现可靠的 LPTIM_CNT 寄存器内容读取，必须执行两次连续的读访问并进行比较。当两次读访问的值相等时，可认为读访问可靠。然而，当使能了异步复位时，不可能两次读取 LPTIM_CNT 寄存器。

注意： LPTIM 内没有防止两个复位机制同时使用的机制。所以开发人员应该确保这两个机制是排斥地使用。

23.3.15 编码器模式

LPTIM 的编码模式分为正交编码和非交编码。

正交编码器

正交编码器接口模式，有 3 种计数方式：

- 编码器模式 1：LPTIM_CFGR 寄存器中的 CKPOL[1:0]位域为 00 时，计数器在上升沿处计数；
- 编码器模式 2：CKPOL[1:0]位域为 01 时，计数器在下降沿处计数；
- 编码器模式 3：CKPOL[1:0]位域为 10 时，计数器在上升沿和下降沿处均计数。

LPTIM 由内部时钟源提供工作时钟时，可使用正交编码器模式。此模式相当于带有方向选择的外部时钟计数器。计数器仅在 0 到 LPTIM_ARR 之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因

此，在启动前必须先配置 LPTIM_ARR。通过两个外部输入信号 LPTIM_IN1 和 LPTIM_IN2 生成时钟信号作为 LPTIM 计数器时钟，这两个信号间的相位确定计数方向。

计数方向的变化由 LPTIM_ISR 的 DOWN 位和 UP 位来指示。如果计数方向由递增变为递减，则 DOWN 位会被置 1；如果计数方向由递减变为递增，则 UP 位会被置 1。DOWN 位、UP 位由软件将 LPTIM_ICR 寄存器中的 DOWN_CF、UP_CF 位置 1 清 0。如果 LPTIM_IER 中的 DOWN_IE 或 UP_IE 位为 1（使能中断）时，相应方向的变化会触发中断。

LPTIM_IN1 和 LPTIM_IN2 输入上的信号频率不得超过 LPTIM_KCLK 时钟频率 4 分频。

配置 LPTIM_CFGR 寄存器中的 ENC 位为 1，且配置 LPTIM_CFGR 寄存器中的 DUALMODE 为 0，可使能 LPTIM 的正交编码模式。LPTIM_CR 寄存器的 CNTSTRT 位配置为 1，启动正交编码模式下计数。

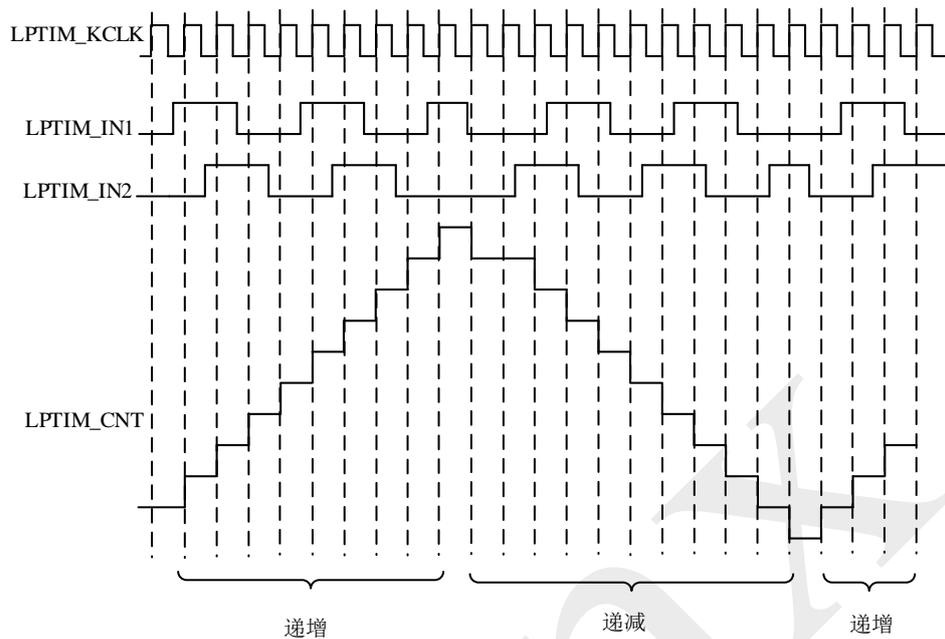
LPTIM_IN1 和 LPTIM_IN2 输入用于连接正交编码器。如果需要，还可对输入滤波器进行配置。根据 CKPOL[1:0]位域的配置，计数器时钟由 LPTIM_IN1 或 LPTIM_IN2 信号上的上升沿、下降沿或双沿提供。请参见下表。

表 23-5 计数器方向与编码器信号的关系

有效边沿	相对信号的电平 (LPTIM_IN1 对应 LPTIM_IN2, LPTIM_IN2 对应 LPTIM_IN1)	LPTIM_IN1 信号		LPTIM_IN2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
双沿（上升 和下降沿）	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

注意： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0]位必须为“000”）。

下图为正交编码器模式下的双沿计数序列。

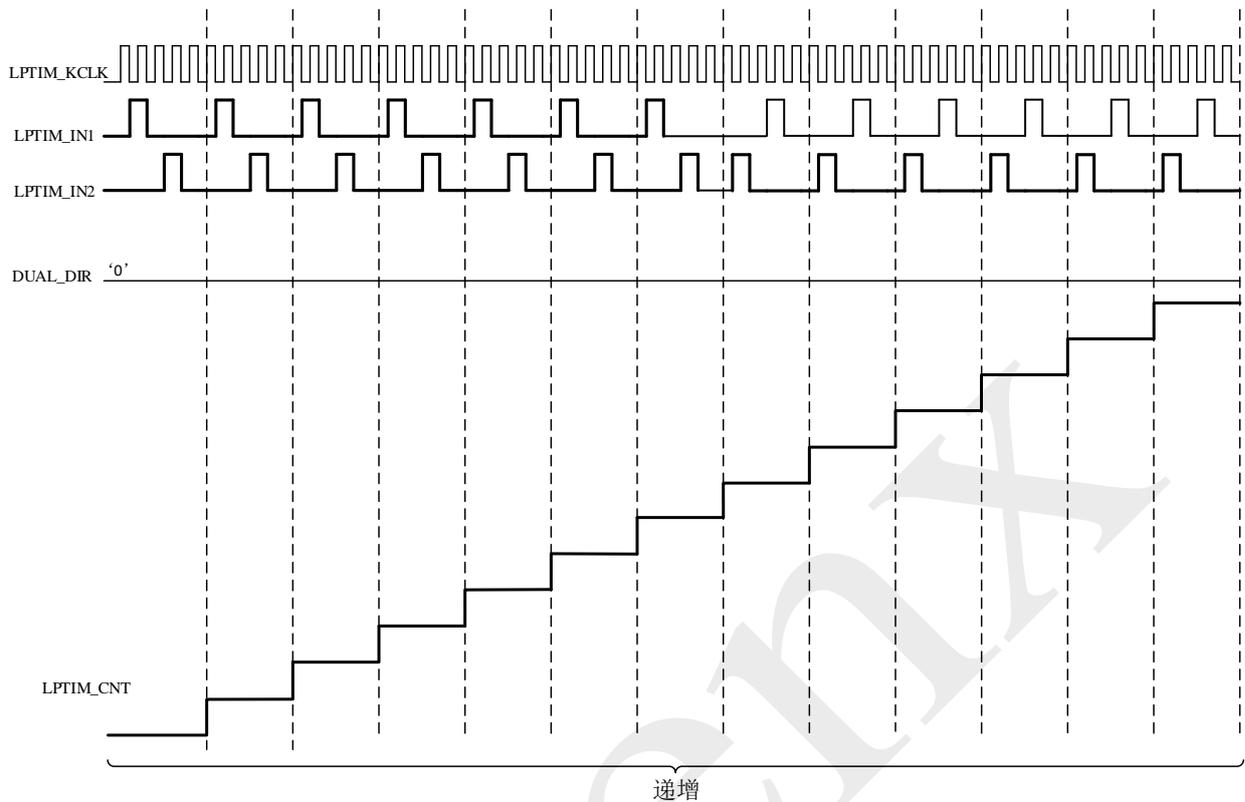
图 23-7 正交编码器模式下的计数序列


非交编码器

配置 LPTIM_CFGR 寄存器中的 DUALMODE 位为 1 时，且配置 LPTIM 寄存器中的 ENC 位清 0，使能非交编码器模式。

通道 1 (LPTIM_IN1) 和通道 2 (LPTIM_IN2) 的极性可通过 LPTIM_CFGR 寄存器中 IN1POL 位和 IN2POL 位配置。

LPTIM 由内部时钟源提供工作时钟时，可使用非交编码器模式。该模式下计数器对通道 1 (LPTIM_IN1) 和通道 2 (LPTIM_IN2) 上的脉冲编码计数，当 LPTIM_IN1 和 LPTIM_IN2 上依次出现两个正脉冲（两个通道脉冲间隔不小于一个 LPTIM_KCLK，两个通道的脉冲先后顺序没有要求），则计数器按照 DUALDIR 位进行一次递增或递减操作。非交编码器模式下的计数序列如下图：

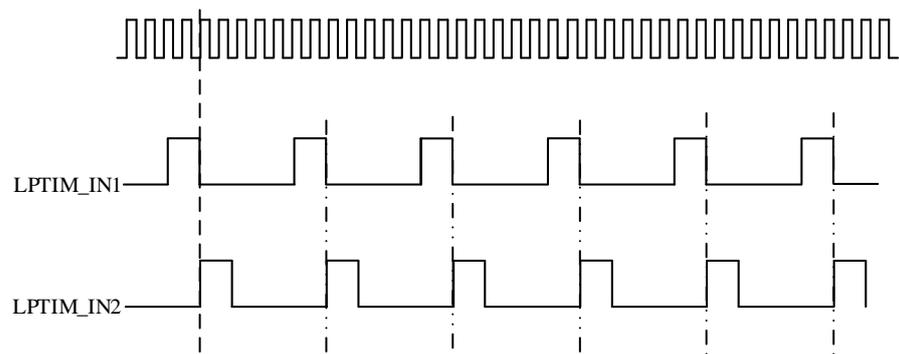
图 23-8 非交编码器模式下的计数序列


注意： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0；另外，PSC 预分频器分频比必须等于其复位值 1（PRESC[2:0] 位必须为“000”）。

非交编码器下面几种情况无法解码，主要归纳如下：

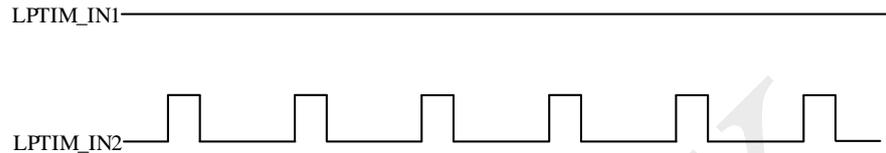
- 通道 1（LPTIM_IN1）、通道 2（LPTIM_IN2）依次出现连续脉冲，且脉冲间无间隔，导致非交编码器无法解码（见下图）。

LPTIM_ISR 寄存器中的 INB2B 位、DE 位将置 1。如果 LPTIM_IER 寄存器中的 INB2BIE 位、DEIE 位使能，将触发双通道脉冲连续错误中断、非交波形错误中断。INB2B 位、DE 位由 LPTIM_ICR 寄存器中的 INB2BCF 位、DECF 位置 1 清 0。

图 23-9 非交编码器背靠背波形无法解码


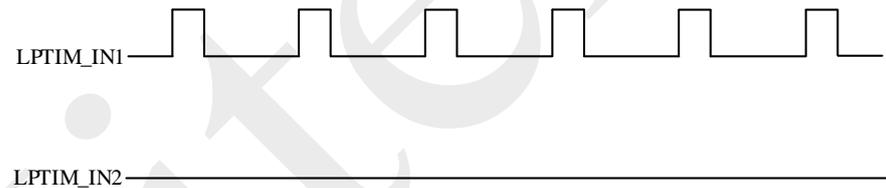
- 通道 1 (LPTIM_IN1) 信号缺失, 导致非交编码器无法解码 (见下图)。LPTIM_ISR 寄存器中的 IN1I 位、DE 位将置 1。如果 LPTIM_IER 寄存器中的 IN1IE 位、DEIE 位使能, 将触发通道 1 缺失中断、非交波形错误中断。IN1I 位、DE 位由 LPTIM_ICR 寄存器中的 IN1ICF 位、DECF 位置 1 清 0。

图 23-10 非交编码器通道 1 缺失

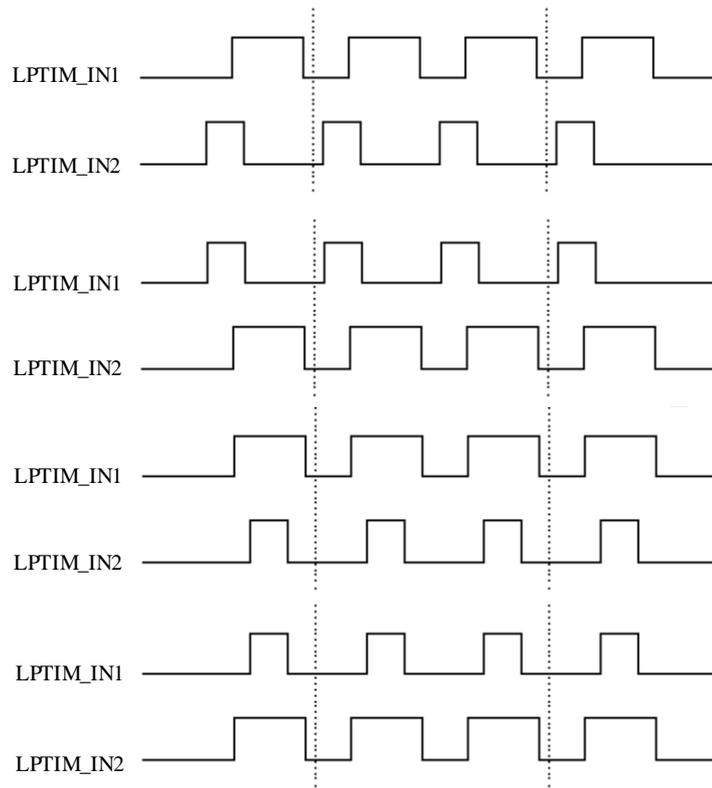


- 通道 2 (LPTIM_IN2) 信号缺失, 导致非交编码器无法解码(见下图)。LPTIM_ISR 寄存器中的 IN2I 位、DE 位将置 1。如果 LPTIM_IER 寄存器中的 IN2IE 位、DEIE 位使能, 将触发通道 2 缺失中断、非交波形错误中断。IN2I 位、DE 位由 LPTIM_ICR 寄存器中的 IN2ICF 位、DECF 位置 1 清 0。

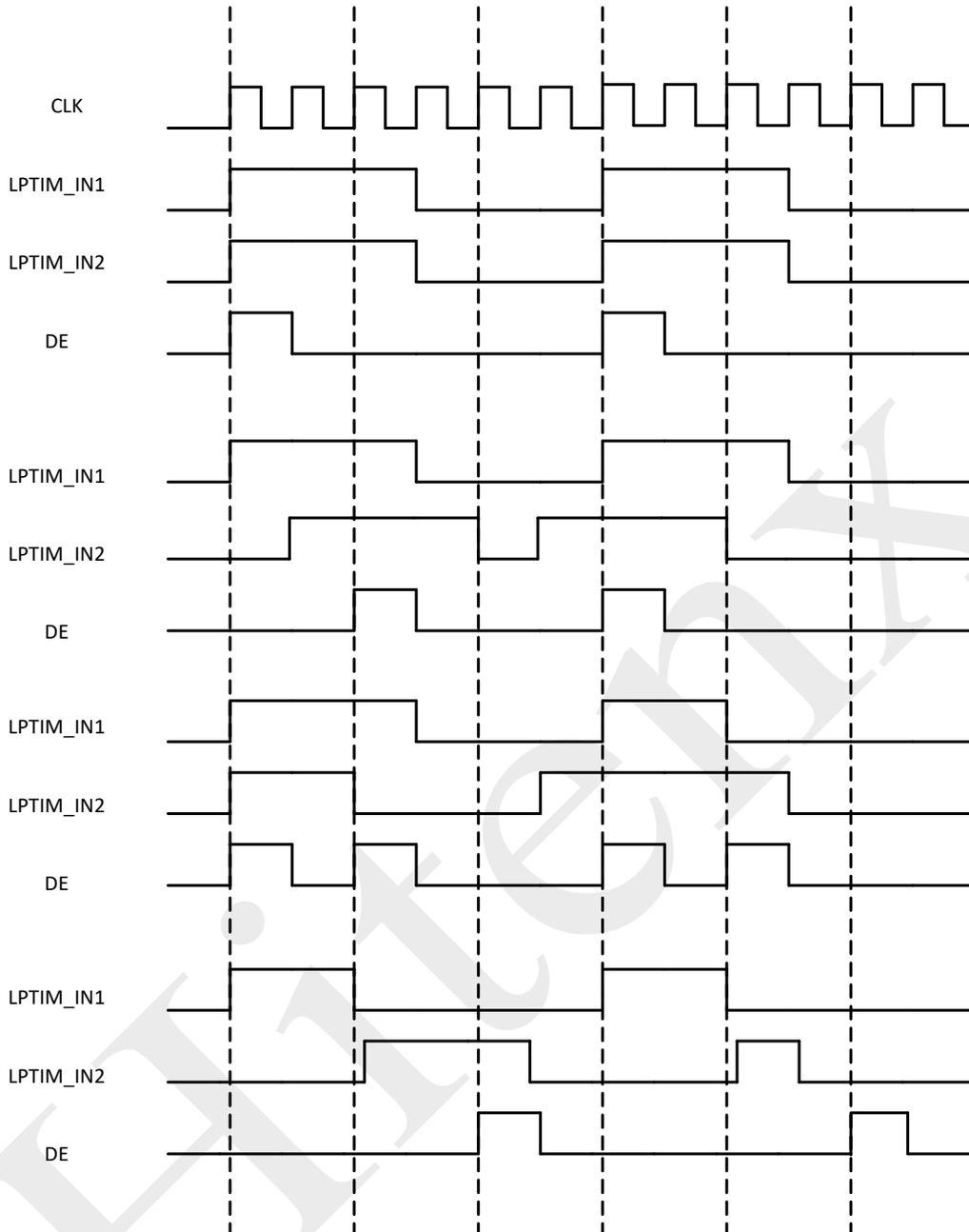
图 23-11 非交编码器通 2 道缺失



- 两个通道 (LPTIM_IN1、LPTIM_IN2) 出现同时为高的情况, 导致非交编码器无法解码 (见下图)。LPTIM_ISR 寄存器中的 DE 位将置 1。如果 LPTIM_IER 寄存器中的 DEIE 位使能, 将触发非交波形错误中断。DE 位由 LPTIM_ICR 寄存器中的 DECF 位置 1 清 0。

图 23-12 非交编码器无法计数波形

图 23-13 非交编码模式错误中断标志置 1

非交编码模式错误中断标志（DE）在下图情况置 1，下图每当 DE 置 1 后，通过 DECF 位清。



23.4 调试模式

当控制器进入调试模式（内核停止）时，LPTIM 计数器要么继续正常工作，要么停止模式，这取决于 DBG 模块中的 DBG_LPTIM1_STOP 配置位。

23.5 LPTIM 低功耗模式

表 23-6 低功耗模式对 LPTIM 的影响

模式	说明
Sleep	无影响，LPTIM 中断可使器件退出睡眠模式。
LPRun	无影响。

模式	说明
LPSleep	无影响，LPTIM 中断可使器件退出睡眠模式。
Stop	当 LSE 或 LSI 锁定 LPTIM 时无影响，LPTIM 中断使能会导致器件退出 Stop。

23.6 中断 LPTIM

若以下事件通过 *LPTIM 使能寄存器 (LPTIM_IER)* 使能，则这些事件会生成中断/唤醒事件：

- 比较匹配
- 自动重载匹配（编码模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成

注意： 只要 *LPTIM_IER* 寄存器（中断使能寄存器）中的位在 *LPTIM_ISR* 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断。

表 23-7 中断事件

中断事件	说明
比较匹配	当计数器寄存器（LPTIM_CNT）的内容与比较寄存器（LPTIM_CMP）的内容匹配时，生成中断标志
自动重载匹配	当 <i>LPTIM 计数器寄存器 (LPTIM_CNT)</i> 的内容与自动重载寄存器（LPTIM_ARR）的内容匹配时，生成中断标志
外部触发事件	当检测到外部触发事件时，会生成中断标志
自动重载寄存器更新成功	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志
比较寄存器更新成功	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志

23.7 LPTIM 寄存器

23.7.1 LPTIM 中断和状态寄存器（LPTIM_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DE	IN B2B	IN2I	IN1I	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM	
					r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
------	----	----

31:11	保留	必须保持复位值。
10	DE	非交编码模式，非交波形错误标志（波形脉冲有重叠会触发此事件通道1缺失和通道2缺失也会同步触发此错误）。
9	INB2B	非交编码模式，双通道脉冲连续标志状态（背靠背两通道脉冲无间隔）。
8	IN2I	非交编码模式，通道2缺失标志状态。
7	IN1I	非交编码模式，通道1缺失标志状态。
6	DOWN	计数方向从递增变为递减 正交编码模式，由硬件将DOWN位置1时，会通知应用计数方向由递增变为递减。
5	UP	计数方向从递减变为递增 正交编码模式，由硬件将UP位置1时，会通知应用计数方向由递减变为递增。
4	ARROK	自动重载寄存器更新成功 由硬件将ARROK置1时，会通知应用LPTIM_ARR寄存器的APB1总线写操作已成功完成。如果这样，便可启动新的写操作。
3	CMPOK	比较寄存器更新成功 由硬件将CMPOK置1时，会通知应用LPTIM_CMP寄存器的APB1总线写操作已成功完成。如果这样，便可启动新的写操作。
2	EXTTRIG	外部触发边沿事件 由硬件将EXTTRIG置1时，会通知应用所选的外部触发输入上产生有效边沿。如果由于定时器已启动而忽略触发事件，则不会将此标志置1。
1	ARRM	自动重载匹配 由硬件将ARRM置1时，会通知应用LPTIM_CNT寄存器的值已达到LPTIM_ARR寄存器的值。
0	CMPM	比较匹配 由硬件将CMPM位置1时，会通知应用LPTIM_CNT寄存器的值已达到LPTIM_CMP寄存器的值。

23.7.2 LPTIM 中断清零寄存器（LPTIM_ICR）

偏移地址：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					DECF	IN B2BCF	IN2 ICF	IN1 ICF	DOWN CF	UPCF	ARROK CF	CMPOK CF	EXTTRIG CF	ARRM CF	CMPM CF
					w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DECF	非交波形错误中断清零（波形脉冲有重叠会触发此事件通道1缺失和通道2缺失也会同步触发此错误）
9	INB2BCF	非交编码模式，双通道脉冲连续错误中断清零（两通道脉冲无间隔）
8	IN2ICF	非交编码模式，通道2缺失错误中断清零
7	IN1ICF	非交编码模式，通道1缺失错误中断清零
6	DOWNCF	方向变为递减清零标志 将1写入此位时，LPTIM_ISR寄存器中的DOWN标志将清零
5	UPCF	计数方向从递减变为递增清零标志 将1写入此位时，LPTIM_ISR寄存器中的UP标志将清零
4	ARROKCF	自动重载寄存器更新成功清零标志 将1写入此位时，LPTIM_ISR寄存器中的ARROK标志将清零
3	CMPOKCF	比较寄存器更新成功清零标志 将1写入此位时，LPTIM_ISR寄存器中的CMPOK标志将清零。
2	EXTTRIGCF	外部触发有效边沿清零标志 将1写入此位时，LPTIM_ISR寄存器中的EXTTRIG标志将清零
1	ARRMCF	自动重载匹配清零标志 将1写入此位时，LPTIM_ISR寄存器中的ARRM标志将清零
0	CMPMCF	比较匹配清零标志 将1写入此位时，LPTIM_ISR寄存器中的CMP标志将清零

23.7.3 LPTIM 中断使能寄存器(LPTIM_IER)

偏移地址：0x08

复位值：0x00000000

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					DEIE	IN B2BIE	IN2 IE	IN1 IE	DOWN IE	UPIE	ARROK IE	CMPOK IE	EXTTRIG IE	ARRM IE	CMPM IE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DEIE	非交波形错误中断使能（波形脉冲有重叠会触发此事件通道1缺失和通道2缺失也会同步触发此错误）
9	INB2BIE	非交编码模式，双通道脉冲连续错误中断使能（两通道脉冲无间隔）
8	IN2IE	非交编码模式，通道2缺失错误中断使能
7	IN1IE	非交编码模式，通道1缺失错误中断使能
6	DOWNIE	方向变为递减中断使能 0: 禁止DOWN中断 1: 使能DOWN中断
5	UPIE	方向变为递增中断使能 0: 禁止UP中断 1: 使能UP中断
4	ARROKIE	自动重载寄存器更新成功中断使能 0: 禁止ARROK中断 1: 使能ARROK中断
3	CMPOKIE	比较寄存器更新成功中断使能 0: 禁止CMPOK中断 1: 使能CMPOK中断
2	EXTTRIGIE	外部触发有效边沿中断使能 0: 禁止EXTTRIG中断 1: 使能EXTTRIG中断
1	ARRMIE	自动重载匹配中断使能 0: 禁止ARRM中断 1: 使能ARRM中断
0	CMPMIE	比较匹配中断使能 0: 禁止CMPM中断 1: 使能CMPM中断

23.7.4 LPTIM 配置寄存器(LPTIM_CFGR)

偏移地址: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IFPSC[2:0]			IN2 POL	IN1 POL	DUAL MODE	DUAL DIR	ENC	COUNT MODE	PRELOAD	WAV POL	WAVE	TIM OUT	TRIG EN[1:0]		Res.
rw	rw	rw	rw	rw	rw	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
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRIGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	IFPSC[2:0]	输入滤波配置 对通道输入信号 (lptim_in1和lptim_in2) 滤波时钟分频, 分频系数可从以下分频系数中选择: 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
28	IN2POL	LPTIM_IN2输入极性选择位 0: 不取反 1: 取反 注: LPTIM_IN2极性选择仅在非交编码模式生效
27	IN1POL	LPTIM_IN1输入极性选择位 0: 不取反 1: 取反 注: LPTIM_IN1极性选择仅在非交编码模式生效
26	DUALMODE	非交编码器模式使能位 0: 禁止 1: 使能
25	DUALDIR	非交编码器模式方向选择位 0: 计数方向为递增 1: 计数方向为递减
24	ENC	正交编码器模式使能位 0: 禁止 1: 使能
23	COUNTMODE	计数时钟源选择 0: 计数器在内部时钟每个有效边缘递增 1: 计数器在输入到LPTIM_IN1的外部时钟每个有效边缘递增

22	PRELOAD	寄存器更新模式选择位 用于选择LPTIM_ARR寄存器和LPTIM_CMP寄存器的更新方式 0: 寄存器实时更新 1: 寄存器在当前LPTIM计数周期结束时更新
21	WAVPOL	波形极性选择位 0: 当LPTIM_CNT小于LPTIM_CMP时, 输出低电平; 当LPTIM_CNT大于LPTIM_CMP且小于LPTIM_ARR时, 输出高电平 1: 当LPTIM_CNT大于LPTIM_CMP时, 输出高电平; 当LPTIM_CNT大于LPTIM_CMP且小于LPTIM_ARR时, 输出低电平
20	WAVE	波形位控制输出波形 0: 停用置一单触发模式和PWM/单脉冲波形 1: 激活置一单触发模式
19	TIMOUT	超时使能位控制超时功能 0: 定时器已启动时到达的触发事件将被忽略 1: 定时器已启动时到达的触发事件将复位并重新启动计数器
18:17	TRIGEN[1:0]	触发使能和极性位控制LPTIM计数器是否由外部触发信启动。 如果已选择由外部触发信号启动, 触发有效边沿的配置有以下三种: 00: 软件触发 (由软件启动计数) 01: 上升沿为有效边沿 10: 下降沿为有效边沿 11: 上升沿和下降沿均为有效边沿
16	保留	必须保持复位值。
15:13	TRIGSEL[2:0]	触发源选择器TRIGSEL位用于选择作为LPTIM触发事件的触发源, 可用触发源包括以下: 000: lptim_ext_trig0 001: lptim_ext_trig1 010: lptim_ext_trig2 011: lptim_ext_trig3 100: lptim_ext_trig4 101: lptim_ext_trig5 110: lptim_ext_trig6 111: lptim_ext_trig7
12	保留	必须保持复位值。
11:9	PRESC[2:0]	时钟预分频器位配置预分频器的分频系数。分频系数可从以下分频系数中选择: 000: 1 001: 2 010: 4 011: 8 100: 16

		101: 32 110: 64 111: 128
8	保留	必须保持复位值。
7:6	TRIGFLT[1:0]	<p>触发信号的可配置数字滤波器TRIGFLT值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能：</p> <p>00: 任何触发信号有效电平变化均视为有效触发</p> <p>01: 触发信号有效电平变化必须至少稳定2个时钟周期，才能将其视为有效触发。</p> <p>10: 触发信号有效电平变化必须至少稳定4个时钟周期，才能将其视为有效触发。</p> <p>11: 触发信号有效电平变化必须至少稳定8个时钟周期，才能将其视为有效触发。</p>
5	保留	必须保持复位值。
4:3	CKFLT[1:0]	<p>外部时钟的可配置数字滤波器CKFLT值用于设置连续相同采样的数量，若在外时钟信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能</p> <p>00: 任何外部时钟信号电平变化均视为有效切换</p> <p>01: 外部时钟信号电平变化必须至少稳定2个时钟周期，才能将其视为有效切换。</p> <p>10: 外部时钟信号电平变化必须至少稳定4个时钟周期，才能将其视为有效切换。</p> <p>11: 外部时钟信号电平变化必须至少稳定8个时钟周期，才能将其视为有效切换。</p>
2:1	CKPOL[1:0]	<p>时钟极性，如果LPTIM由外部时钟源提供时钟：</p> <p>当LPTIM由外部时钟源提供时钟时，CKPOL位用于配置计数所使用的有效边沿：</p> <p>00: 上升沿为用于计数的有效边沿</p> <p>01: 下降沿为用于计数的有效边沿</p> <p>10: 上升沿和下降沿均为有效边沿</p> <p>当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。</p> <p>11: 不允许</p>
0	CKSEL	<p>时钟选择器，CKSEL位选择LPTIM将使用的时钟源：</p> <p>0: LPTIM由内部时钟源（APB1时钟或任意内置振荡器）提供时钟</p> <p>1: LPTIM由外部时钟源通过LPTIM外部Input1提供时钟</p>

23.7.5 LPTIM 控制寄存器 (LPTIM_CR)

偏移地址：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
											rw	rs	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	RSTARE	读使能后复位 此位由软件置1和清零，当RSTARE设置为1之后，任何一个对LPTIM_CNT寄存器的读操作都会异步复位LPTIM_CNT寄存器的内容。
3	COUNTRST	计数器复位 此位由软件置1，硬件清零。此位置1后会产生一个LPTIM_CNT寄存器的同步复位，因为是同步复位，复位只会发生在3个LPTimer内核时钟周期的同步延迟之后。（LPTimer的内核时钟可能和APB1时钟不同） <i>注意：当此位被硬件清零之前，软件不可对其进行写1操作，所以软件在写1之前必须检验此位为0。</i>
2	CNTSTRT	定时器以连续模式启动 此位通过软件置1，硬件清零。 若通过软件启动（TRIGEN[1:0]=“00”），将此位置1会使LPTIM以连续模式启动；如果禁止软件启动（TRIGEN[1:0]不等于“00”），将此位置1会使定时器在检测到外部触发信号后立即以连续模式启动；如果在进行单脉冲模式计数时将此位置1，则在LPTIM_ARR寄存器和LPTIM_CNT寄存器下一次匹配时定时器不会停止，LPTIM计数器将继续以连续模式计数。 <i>注意：只有在使能LPTIM时，才能将此位置1。此位由硬件自动复位。</i>
1	SNGSTRT	LPTIM以单次模式启动 此位通过软件置1，通过硬件清零。 若通过软件启动（TRIGEN[1:0]=“00”），将此位置1会使LPTIM以单脉冲模式启动；如果禁止软件启动（TRIGEN[1:0]不等于“00”），将此位置1会使LPTIM在检测到外部触发信号后立即以单脉冲模式启动；如果在LPTIM处于连续计数模式时将此位置1，LPTIM将在LPTIM_ARR寄存器和LPTIM_CNT寄存器下一次匹配时停止。 <i>注意：只有在使能LPTIM时，才能将此位置1。此位由硬件自动</i>

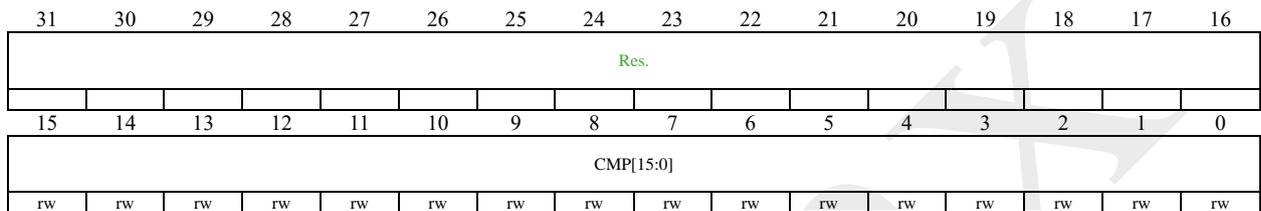
复位。

0	ENABLE	LPTIM使能ENABLE位由软件置1和清零。 0: 禁止LPTIM 1: 使能LPTIM
---	--------	---

23.7.6 LPTIM 比较寄存器 (LPTIM_CMP)

偏移地址: 0x14

复位值: 0x00000000

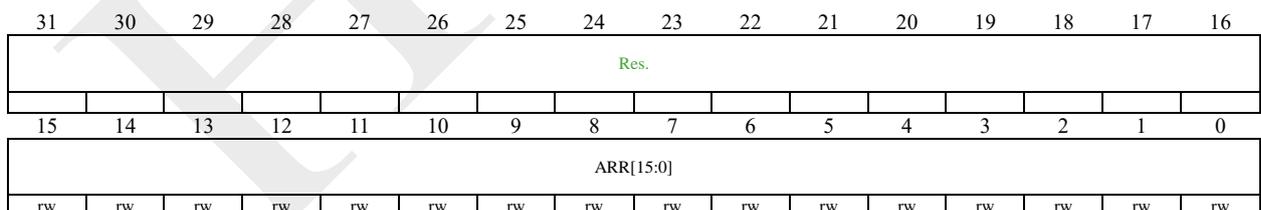


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CMP[15:0]	比较值 CMP为LPTIM所使用的比较值。 必须在LPTIM已使能时 (ENABLE位置1) 才能修改LPTIM_CMP寄存器的内容。

23.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR)

偏移地址: 0x18

复位值: 0x00000001



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ARR[15:0]	自动重载值 此值必须严格大于 CMP[15:0]的值。必须在 LPTIM 已使能时 (ENABLE 位置 1) 才能修改 LPTIM_ARR 寄存器的内容。

23.7.8 LPTIM 计数器寄存器 (LPTIM_CNT)

偏移地址: 0x1C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	计数器值 当 LPTIM 通过异步时钟运行时, 读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下, 必须连续执行读访问两次, 并验证两次返回的值是否相同。

23.7.9 LPTIM 配置寄存器 2 (LPTIM_CFGR2)

偏移地址: 0x20

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										IN2SEL[1:0]		Res.		IN1SEL[1:0]	
										rw	rw			rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	IN2SEL[1:0]	LPTIM input 2选择信号 IN2SEL位控制LPTIM input 2的输入选择器, 将LPTIM input 2连接到一个有效的输入 00: lptim_in2_mux0

		01: lptim_in2_mux1 10: lptim_in2_mux2 11: lptim_in2_mux3 详细输入信号选择请参见： 表LPTIM输入输出引脚和内部信号
3:2	保留	必须保持复位值。
1:0	IN1SEL[1:0]	LPTIM input 1选择信号 IN1SEL位控制LPTIM input 1的输入选择器，将LPTIM input 1连接到一个有效的输入 00: lptim_in1_mux0 01: lptim_in1_mux1 10: lptim_in1_mux2 11: lptim_in1_mux3 详细输入信号选择请参见： 表LPTIM输入输出引脚和内部信号

24 SysTick 定时器 (SysTick)

24.1 简介

ARM® Cortex®-M0+内核提供了一个硬件定时器 SysTick。

在 OS 中多任务调度时需要周期性执行上下文切换，可以借助 SysTick 实现此功能。每当定时器中断产生时，处理器都会在异常处理中断中进行 OS 任务调度及维护工作。

24.2 主要特性

- 24 位硬件定时器
- 减计数
- 提供异常处理中断
- 为 RTOS 提供可编程系统定时器
- 无系统时可作为定时器，计时器或周期性中断源

24.3 功能描述

SysTick 是一个 24 位硬件定时器。通过将 SYST_CVR 寄存器的 ENABLE 位置 1 使能时，计数器从当前计数寄存器 SYST_CVR 的值开始减计数，每当减到 0 时，会在下一个时钟沿重新装载 SYST_RVR 寄存器的数值到 SYST_CVR 中，并再次进入减计数。

每次计数到 0 时，SysTick 的 COUNTFLAG 标志位将置位，读此位时标志清零。

写入 SYST_CVR 寄存器时，SYST_CVR 寄存器复位，同时 COUNTFLAG 标志位也将清零。写入 SYST_CVR 寄存器后的下一个时钟沿，SYST_CVR 寄存器会装载 SYST_RVR 中的计数值，重新开始计数。写入 SYST_CVR 寄存器不会使溢出标志位 COUNTFLAG 置位，也不会触发中断。读 SYST_CVR 寄存器时返回值为当前计数器的值。

向 SYST_RVR 寄存器中写“0”时，计数器在当前计数完成后停止计数，停止后计数器的值保持为“0”。

注意： 当前计数值寄存器 `SYST_CVR` 在复位后是未知状态，所以在首次使能 `SysTick` 前建议首先写入 `SYST_RVR` 寄存器，再写入 `SYST_CVR`。这样的操作顺序保证了正确的重载值被写入，并同时清空 `SYST_CVR`。之后 `SysTick` 使能时直接从已知的 `SYST_RVR` 开始计数，避免了从未知的 `SYST_CVR` 开始计数带来的隐患。

`SYST_CALIB` 校准值寄存器中 `TENMS[23:0]` 位域提供了校准值，可以通过此校准值配置计数器每隔 1ms 生成中断请求。

当 MCU 处于调试模式内核停止时，`SysTick` 定时器也会停止计数。

24.4 SysTick 寄存器

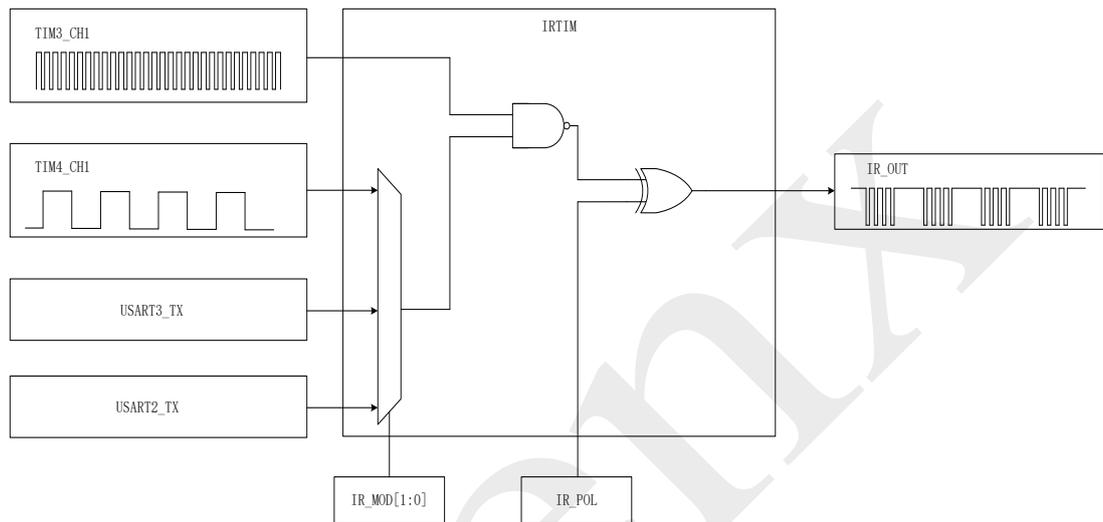
参见内核寄存器——[SysTick 寄存器](#) 章节。

25 红外控制模块 (IRTIM)

红外控制模块 (IRTIM) 能够产生红外控制信号, 可以配合红外 LED 发射器一同使用, 以实现红外远程控制功能。

IRTIM 在芯片内部与 USART2、USART3、TIM3、TIM4 相连, 详见下图。

图 25-1 IRTIM 内部硬件连接



通过配置不同的载波和调制信号, 可以产生符合任意红外脉冲调制标准的红外控制信号。

TIM3 定时器的通道 1 (TIM3_CH1) 用于产生高频载波信号。为了避免毛刺和干扰, IR 信号通过 TIM3_CH1 的载波同步。

通过配置 [系统控制寄存器 \(SYSCFG_CR\)](#) 中的 IR_MODE[1:0] 位域, 可以选择 TIM4 定时器的通道 1 (TIM4_OC1)、USART2 或 USART3 中的一路信号作为调制信号。

IRTIM 输出信号的极性, 由 [系统控制寄存器 \(SYSCFG_CR\)](#) 中的 IR_POL 控制位进行选择, 可以根据需要将输出极性反转。

IRTIM 产生的红外控制信号由 IR_OUT 引脚输出, 通过 GPIO 引脚 (详见 datasheet) 的复用功能进行配置。

26 独立看门狗 (IWDG)

26.1 简介

MCU 内置独立看门狗，可检测并解决芯片发生的异常，并在计数器达到给定的超时值时触发系统复位。

独立看门狗 (IWDG) 由其内部低速时钟 (LSI) 驱动，因此在系统时钟发生故障时仍然保持工作状态。

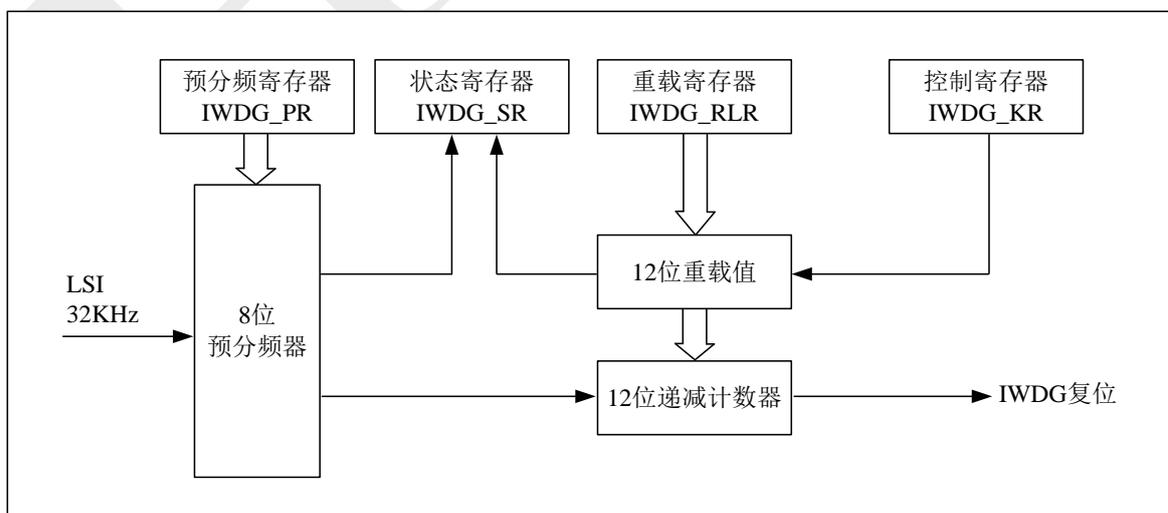
26.2 功能特性

- 递减计数器
- 工作时钟为 32kHz LSI 提供，可在 Stop 模式下工作
- 复位条件（如果看门狗已使能）
 - 当递减计数器值小于 0x000 时复位
 - 在窗口之外重载递减计数器时复位
- 可通过选项字节设置，来控制上电时是否使能独立看门狗。

26.3 功能说明

26.3.1 结构框图

图 26-1 独立看门狗结构框图



独立看门狗功能在 Stop 模式下仍能工作。

通过将 0x0000 CCCC 写到控制寄存器 (IWDG_KR) 来使能独立看门狗时, 计数器开始从配置值 (默认值 0xFFFF) 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号 (IWDG 复位)。

任何时候将 0x0000 AAAA 写到控制寄存器 (IWDG_KR) 中, 重载寄存器 (IWDG_RLR) 的值就会被重载到递减计数器, 从而避免产生看门狗复位。

IWDG 一旦运行, 其时钟源 (LSI) 无法停止; IWDG 使能后, LSI 自动打开。

注意: 当重载值、预分频值、窗口值正在被使用时, 在更改重载值前要等待 RVU 位更新完成; 在更改预分频值前要等待 PVU 位更新完成; 在更改窗口值前要等待 WVU 位更新完成。然而, 在更新预分频值、重载值、窗口值后, 不需要分别等到 RVU、PVU、WVU 复位后再继续执行代码, 除非是低功耗模式进入的情况。

26.3.2 窗口功能

通过在窗口寄存器 (IWDG_WINR) 中设置合适的窗口, IWDG 也可以用作窗口看门狗。

当计数器值大于窗口寄存器 (IWDG_WINR) 中的值时, 如果执行重载操作, 则会产生复位。

窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF, 因此, 如果不更新此默认值, 将禁止窗口功能。

窗口值一经更改, 便执行重载操作 (将递减计数器复位为重载寄存器值)。

26.3.2.1 使能窗口功能时配置 IWDG

使能窗口功能时, 可按以下步骤配置 IWDG:

- 1) 通过将 0x0000 CCCC 写入控制寄存器 (IWDG_KR) 来使能 IWDG;
- 2) 通过将 0x0000 5555 写入控制寄存器 (IWDG_KR) 来使能寄存器访问;
- 3) 在预分频器寄存器 (IWDG_PR) 中配置预分频值;
- 4) 将重载值写入重载寄存器 (IWDG_RLR);
- 5) 等待状态寄存器更新 (IWDG_SR 为 0x0000 0000);
- 6) 将窗口值写入窗口寄存器 (IWDG_WINR), 这会自动刷新计数器值为重载寄存器 (IWDG_RLR) 的值。

26.3.2.2 禁止窗口功能时配置 IWDG

不禁止窗口功能时, 可按以下步骤配置 IWDG:

- 1) 通过将 0x0000 CCCC 写入控制寄存器 (IWDG_KR) 来使能 IWDG;
- 2) 通过将 0x0000 5555 写入控制寄存器 (IWDG_KR) 来使能寄存器访问;
- 3) 在预分频器寄存器 (IWDG_PR) 中配置预分频值;

- 4) 将重载值写入重载寄存器 (IWDG_RLR);
- 5) 等待状态寄存器更新 (IWDG_SR 为 0x0000 0000);
- 6) 通过将 0x0000 AAAA 写入控制寄存器 (IWDG_KR) 使计数器值刷新为重载寄存器 (IWDG_RLR) 的值。

26.3.3 硬件方式使能看门狗

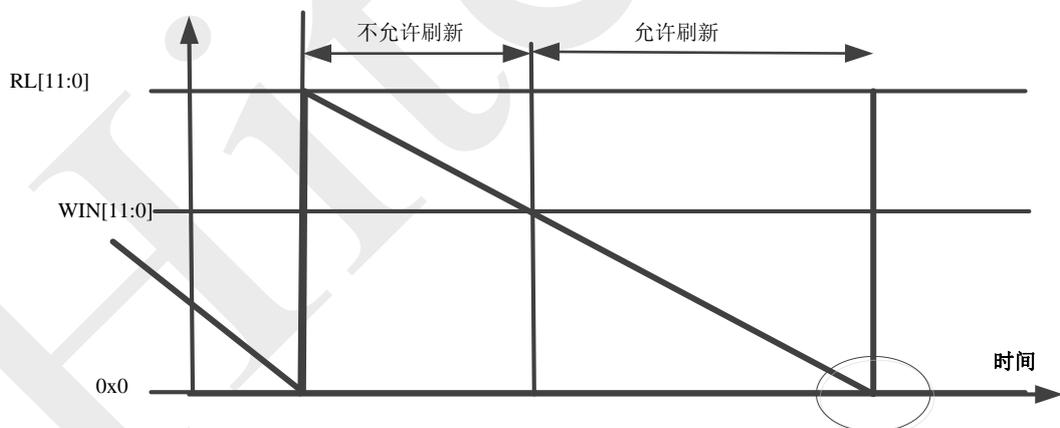
如果通过 [选项字节寄存器 1](#) 使能“硬件看门狗”功能，上电时将使能看门狗；如果在计数器计数结束前，若软件没有向控制寄存器 (IWDG_KR) 写入相应的值，或者在窗口之外重载了递减计数器，则系统会产生复位。

26.3.4 寄存器访问保护

预分频器寄存器 (IWDG_PR)、重载寄存器 (IWDG_RLR) 和窗口寄存器 (IWDG_WINR) 具有写访问保护功能。若要修改这些寄存器，用户必须首先对控制寄存器 (IWDG_KR) 写入 0x0000 5555，而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这意味着重载操作 (即向控制寄存器 (IWDG_KR) 写入 0x0000 AAAA) 也会启动写保护功能。

26.3.5 独立看门狗超时设置

图 26-2 独立看门狗超时时间图



超时值的计算公式如下：

$$T_{IWDG} = T_{LSI} \times 2^{PR[2:0]+2} \times (RL[11:0]+1) \text{ (ms)} \quad (PR[2:0] < 7)$$

$$T_{IWDG} = T_{LSI} \times 2^{PR[2:0]+1} \times (RL [11:0]+1) \text{ (ms)} \quad (PR[2:0] = 7)$$

其中：

T_{IWDG} ：IWDG超时值

$PR[2:0]$ ：预分频值

T_{LSI} ：单位为ms

例如，将 $PR[2:0]$ 设置为1，并将 $RL [11:0]$ 设置为0xFFFF：

$$T_{IWDG} = (1/32) \times 2^{1+2} \times (4095+1) = 1024(\text{ms})$$

26.3.6 调试模式

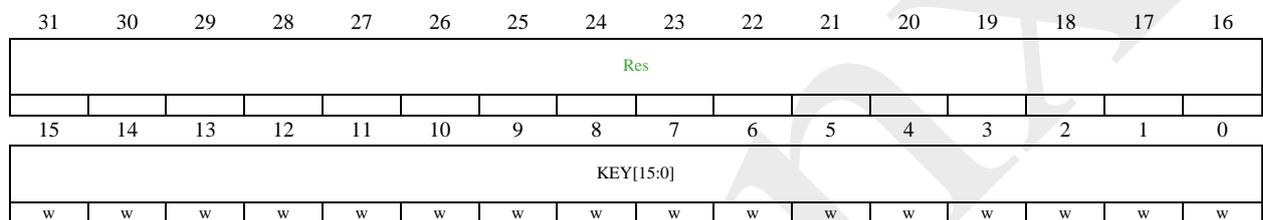
当 MCU 进入调试模式调试暂停时，IWDG 计数器会根据 DBG 模块中的 DBG_IWDG_STOP 选择继续正常工作或者停止工作。

26.4 IWDG 寄存器

26.4.1 IWDG 控制寄存器 (IWDG_KR)

偏移地址：0x00

复位值：0x0000 0000;

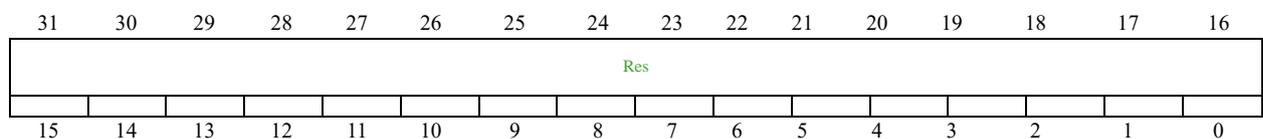


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	KEY[15:0]	<p>控制值</p> <p>只能写入，读为 0x0000。</p> <p>必须定期对这些位写入 0x0000 AAAA，使递减计数器值刷新为重载寄存器 (IWDG_RLR) 的值；否则当递减计数器计数到 0 时，看门狗会产生复位。</p> <p>写入 0x0000 5555 可使能对预分频寄存器 (IWDG_PR)、重载寄存器 (IWDG_RLR) 和窗口寄存器 (IWDG_WINR) 的访问。</p> <p>写入 0x0000 CCCC 可使能看门狗（硬件方式使能看门狗的情况除外）。</p> <p><i>注意：若使能窗口功能（窗口值写入窗口寄存器），这会自动刷新计数器值为重载寄存器 (IWDG_RLR) 的值。</i></p>

26.4.2 IWDG 预分频寄存器 (IWDG_PR)

偏移地址：0x04

复位值：0x0000 0007;



Res												PR[2:0]		
												rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
2:0	PR[2:0]	<p>预分频值</p> <p>这些位受写访问保护，通过设置这些位来选择计数器时钟的预分频值。预分频值必须完成更新时（IWDG_SR的PVU位必须为0），才能更改预分频器的分频系数。</p> <p>000: 4分频 001: 8分频 010: 16分频 011: 32分频 100: 64分频 101: 128分频 110: 256分频 111: 256分频</p> <p><i>注意：只有在预分频值更新完成（IWDG_SR寄存器中的PVU位为0）时，从寄存器读取的值才有效，否则读取的值可能不是最新的。</i></p>

26.4.3 IWDG 重载寄存器 (IWDG_RLR)

偏移地址: 0x08

复位值: 0x0000 0FFF;

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

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	RL[11:0]	<p>重载值</p> <p>每次对控制寄存器（IWDG_KR）写入 0xAAAA 时，RL[11:0]就会重载到独立看门狗计数器中。之后，计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。</p> <p>重载值必须完成更新时（IWDG_SR的RVU位必须为0），才能更改重载寄存器值。</p> <p><i>注意：只有在重载值更新完成（IWDG_SR寄存器中的RVU位为0）时，从寄存器读取的值才有效，否则读取的值可能不是最新的。</i></p>

26.4.4 IWDG 状态寄存器 (IWDG_SR)

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
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													WVU	RVU	PVU
													r	r	r

位/位域	名称	描述
31:3	保留	必须保持复位值
2	WVU	看门狗计数器窗口值更新 当对窗口寄存器写操作时该位自动置 1，完成写操作后自动清 0。 该位由硬件置 1 表示窗口值正在更新。当在完成重载值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。 窗口值只有在 WVU 位为 0 时才可更新。
1	RVU	看门狗计数器重载值更新 当对重载寄存器写操作时该位自动置 1，完成写操作后自动清 0。 该位由硬件置 1 以指示重载值正在更新。当在 V _{DD} 电压域下完成重载值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。 重载值只有在 RVU 位为 0 时才可更新。
0	PVU	看门狗计数器预分频值更新 当对预分频寄存器写操作时该位自动置 1，完成写操作后自动清 0。 该位由硬件置 1 以指示预分频值正在更新。当在 V _{DD} 电压域下完成预分频值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。 预分频值只有在 PVU 位为 0 时才可更新。

26.4.5 IWDG 窗口寄存器 (IWDG_WINR)

偏移地址: 0x10

复位值: 0x0000 0FFF;

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				WIN[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WIN[11:0]	<p>计数器窗口值</p> <p>这些位包含用于与递减计数器进行比较的窗口值上限。为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载。</p> <p>窗口值必须完成更新时（IWDG_SR 的 WVU 位必须为 0），才能更改窗口寄存器值。</p> <p><i>注意：只有在窗口值更新完成（IWDG_SR 寄存器中的 WVU 位为 0）时，从寄存器读取的值才有效，否则读取的值可能不是最新的。</i></p>

27 窗口看门狗 (WWDG)

27.1 简介

窗口看门狗 (WWDG) 通常被用来检测并解决芯片发生的异常，并在计数器达到给定的超时值时触发系统复位。

WWDG 时钟由 APB1 时钟经预分频后提供，通过可配置的时间窗口来检测异常操作。

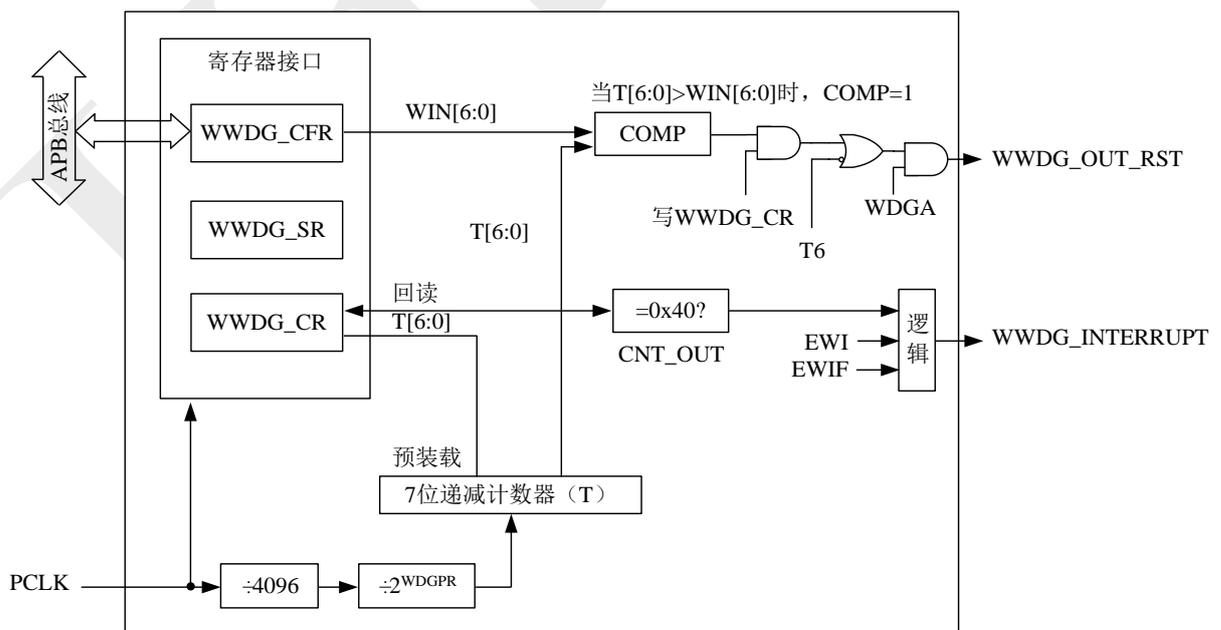
27.2 功能特性

- 递减计数器
- 复位条件 (如果看门狗已使能)
 - 当递减计数器值小于 0x40 时复位
 - 在窗口之外重载递减计数器时复位
- 提前唤醒中断 (EWI): 当计数器递减到 0x40 时触发

27.3 功能说明

27.3.1 结构框图

图 27-1 窗口看门狗结构框图



在正常运行过程中必须定期地向控制寄存器 (WWDG_CR) 写入计数值，来刷新窗口看门狗的计数值，以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值且高于 0x3F 时，才能执行定期写入控制寄存器 (WWDG_CR) 操作。

注意: 要写入控制寄存器 (WWDG_CR) 中的 T[6:0] 位域必须介于 0x7F 和 0x40 之间。

27.3.2 递减计数器

当窗口看门狗使能时, 无法禁止, 直至下次复位。如果使能窗口看门狗 (控制寄存器 (WWDG_CR) 中的 WDGA 位置 1), 则当 7 位递减计数器 (T[6:0]位域) 从 0x40 递减到 0x3F (T[6]已清零) 时会产生复位, 因此当使能窗口看门狗时, T[6:0]位域必须大于 0x40 (T[6]位置 1), 以防止立即复位。

窗口值通过配置寄存器 (WWDG_CFR) 来配置, 当计数器值大于窗口值时, 如果此时重载计数器, 会立即产生复位, 所以必须在小于窗口值且大于 0x3F 时重载计数值。

T[5:0]位包含了窗口看门狗的计数值, 最大计数周期由 APB1 时钟周期、配置寄存器 (WWDG_CFR) 中的 WDG TB[2:0]位域和控制寄存器 (WWDG_CR) 中的 T[5:0]位域决定。

注意: 可使用 T[6] 位产生软件复位 (将控制寄存器的 WDGA 位置 1 并将 T[6] 位清 0)。

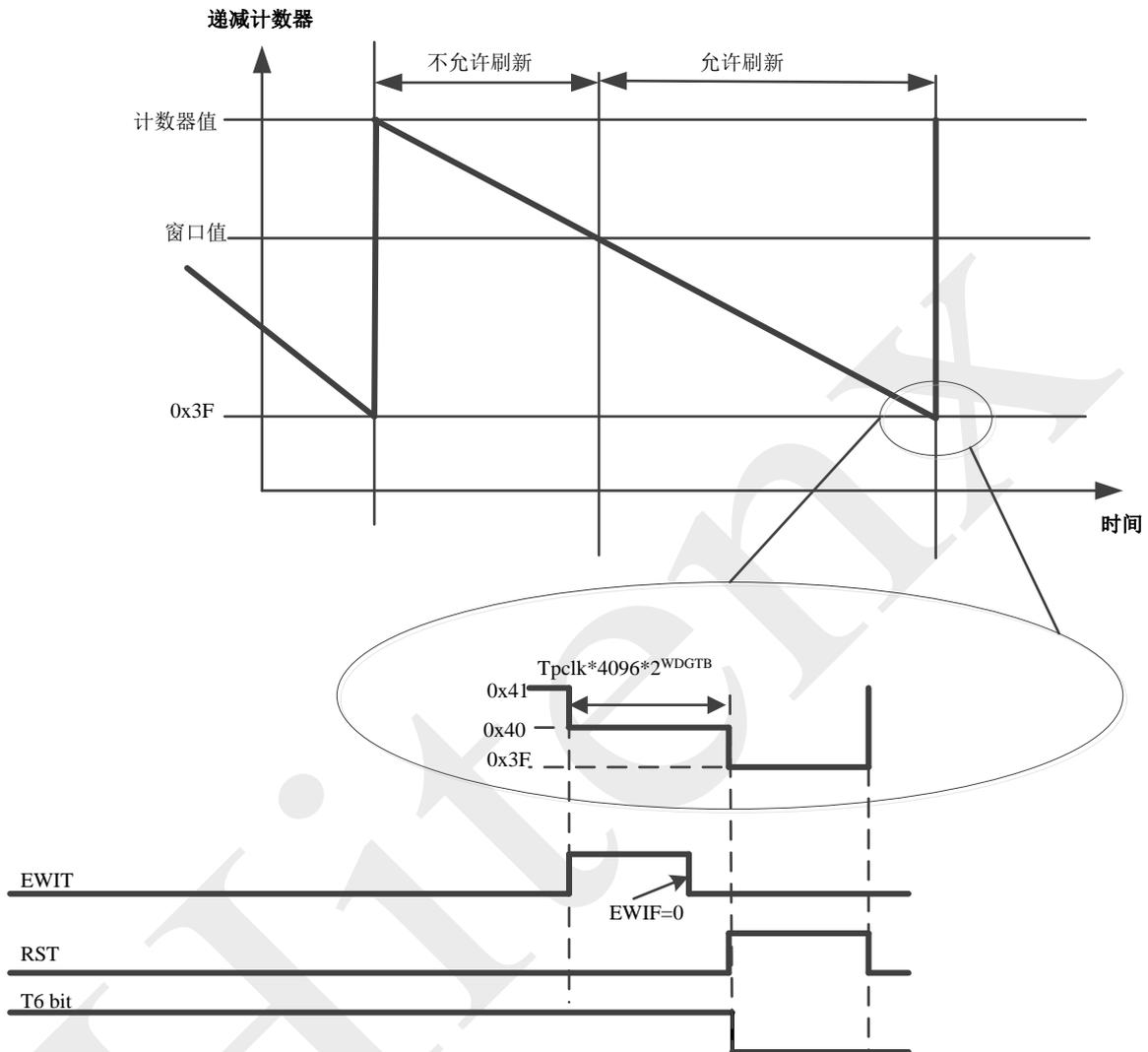
27.3.3 看门狗中断

如果在产生复位之前必须执行相关的数据保存操作, 可使用提前唤醒中断 (EWI)。通过将配置寄存器 (WWDG_CFR) 中的 EWI 位置 1 来使能 EWI 中断。当计数器的值递减为 0x40 时, 将产生 EWI 中断。在相应的中断服务程序中执行复位前的特定操作; 若不希望生成 WWDG 复位, 可在相应的中断服务程序重载 WWDG 计数器。

注意: 配置寄存器 (WWDG_CFR) 中的 EWI 位只有在复位后才由硬件清零。

通过状态寄存器 (WWDG_SR) 的 EWIF 位清 0 来清除 EWI 中断。

注意: 当有更高优先级中断发生时导致 EWI 中断无法及时响应, 最终会发生 WWDG 复位。

27.3.4 窗口看门狗超时设置
图 27-2 窗口看门狗超时时间图


超时值的计算公式如下：

$$t_{\text{WWDG}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}[2:0]} \times (\text{T}[5:0] + 1) \text{ (ms)}$$

其中：

t_{WWDG} ：WWDG超时值

t_{PCLK} ：APB1时钟周期，单位为ms

4096：对应于内部分频器的值

例如：假设APB频率等于48MHz，将WDGTB[2:0]设置为1，并将T[5:0]设置为20：

$$t_{\text{WWDG}} = (1/48000) \times 4096 \times 2^1 \times (20 + 1) = 3.58 \text{ (ms)}$$

27.3.5 调试模式

当 MCU 进入调试模式调试暂停时，WWDG 计数器会根据 DBG 模块中的 DBG_WWDG_STOP 选择继续正常工作或者停止工作。

27.4 WWDG 寄存器

27.4.1 WWDG 控制寄存器 (WWDG_CR)

偏移地址：0x00

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								WDGA	T[6:0]						
								rs	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7	WDGA	使能位 此位由软件置1，只有复位后才由硬件清零，软件写0无效。当WDGA置1时，看门狗可产生复位。 0：禁止看门狗 1：使能看门狗
6:0	T[6:0]	7 位计数器 (MSB 到 LSB) 窗口看门狗计数器的值。它每隔 $(4096 \times 2^{WDGTB[2:0]})$ PCLK 个周期递减一次。当它从 0x40 递减到 0x3F (T[6]位为 0) 时会产生复位。

27.4.2 WWDG 配置寄存器 (WWDG_CFR)

偏移地址：0x04

复位值：0x0000 387F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		WDGTB			Res	EWI	Res		W[6:0]						
		rw	rw	rw		rs			rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

31:14	保留	必须保持复位值
13:11	WDGTB[13:11]	计数时钟预分频 PCLK 的 4096 分频后的预分频设置（参见 框图 27-1 ） 000: 1 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频
10	保留	必须保持复位值
9	EWI	提前唤醒中断 置 1 后，只要计数器值达到 0x40 就会产生中断。
8:7	保留	必须保持复位值
6:0	W[6:0]	7位窗口值 这些位包含用于与递减计数器进行比较的窗口值。

27.4.3 WWDG 状态寄存器 (WWDG_SR)

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
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															EWIF
															rc_w0

位/位域	名称	描述
31:1	保留	必须保持复位值
0	EWIF	提前唤醒中断标志 当计数值递减到0x40时此位被置1，由软件通过写0来清除。写1无效。 如果不使能EWI中断，此位也会被置1

28 I2C 接口 (I2C)

28.1 简介

I2C 接口实现 MCU 和外部 I2C 设备的同步通信，支持标准模式 (Standard-mode (Sm))、快速模式 (Fast-mode (Fm)) 和超快速模式 (Fast-mode Plus (Fm+))。实现多主模式功能，可以控制时钟同步和总线仲裁。

28.2 主要特性

- 兼容 I2C 总线规范 Rev.3:
 - 主模式和从模式
 - 多主模式
 - 标准模式 (最高 100 KHz)
 - 快速模式 (最高 400 KHz)
 - 超快速模式 (最高 1 MHz)
 - 7 位和 10 位寻址模式
 - 多个 7 位从地址 (2 个从设备地址寄存器，1 个具有可配置的匹配位数)
 - 所有 7 位地址应答
 - 广播地址
 - 可配置的数据建立和保持时间
 - 从模式可配置时钟延长
 - 方便易用的事件管理
 - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可配置的数字噪声滤波器

I2C1 还实现以下扩展功能, I2C2 未实现以下扩展功能。

- 支持独立时钟源，可使 I2C 通信速度不受 PCLK 时钟频率更改的影响
- 地址匹配时从停止模式唤醒

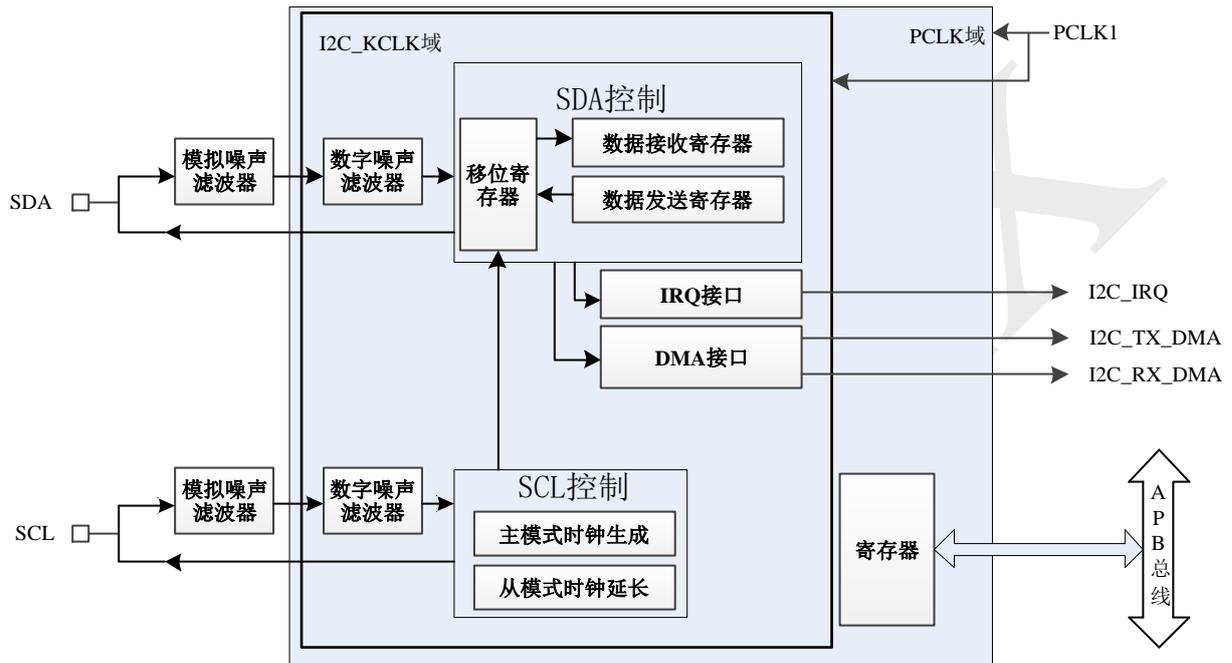
28.3 I2C 功能描述

该接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。

28.3.1 I2C 框图

I2C 模块框图如下图。

图 28-1 I2C 框图



28.3.2 I2C 引脚和内部信号

表 28-1 I2C 引脚

引脚名称	引脚类型	说明
SCL	输入/输出	I2C 同步时钟引脚
SDA	输入/输出	I2C 数据引脚

表 28-2 I2C 内部信号

引脚名称	引脚类型	说明
I2C_KCLK	输入	I2C 内核时钟
PCLK	输入	I2C 的 APB 时钟
I2C_IRQ	输出	I2C 中断信号
I2C_TX_DMA	输出	I2C 发送数据 DMA 请求信号
I2C_RX_DMA	输出	I2C 接收数据 DMA 请求信号
I2C_WAKEUP	输出	I2C 唤醒信号

28.3.3 I2C 时钟要求

I2C 内核的时钟由 I2C_KCLK 提供，I2C_KCLK 周期 t_{I2C_KCLK} 必须遵循以下条

件:

$$t_{I2C_KCLK} < (t_{LOW}) / 4 \text{ 且 } t_{I2C_KCLK} < t_{HIGH}$$

其中 t_{LOW} : SCL 低电平时间;

t_{HIGH} : SCL 高电平时间;

数字滤波器延时为 $DNF \times t_{I2C_KCLK}$ 。

PCLK 时钟周期 t_{PCLK} 必须遵循以下条件:

$$t_{PCLK} < 4/3 t_{SCL}$$

其中 t_{SCL} : SCL 周期。

注意: 当 I2C 内核时钟由 PCLK 提供时, PCLK 必须遵循 t_{I2C_KCLK} 的条件。

28.3.4 模式选择

I2C 接口有四种工作模式:

- 从模式发送
- 从模式接收
- 主模式发送
- 主模式接收

I2C 接口默认工作在从模式。当需要进行数据传输时, 当 I2C_CR2 寄存器的 START 位置 1 生成起始位后会由从模式切换为主模式, 在出现仲裁丢失或生成停止位时从主模式切换为从模式, 从而实现多主模式功能。

通信流程

在主模式下, I2C 接口会启动数据传输并生成时钟信号。串行数据传输是以起始位开始, 停止位结束。I2C_CR2 寄存器的 START 位置 1 生成起始位和 I2C_CR2 寄存器的 STOP 位置 1 生成停止位。总线在起始位后处于忙的状态, 在停止位后总线被释放, 处于空闲状态。如果数据传输后无停止位, 主设备再次发送起始位传输数据, 该起始位为重复起始位, 总线在此过程中一直是忙的状态。

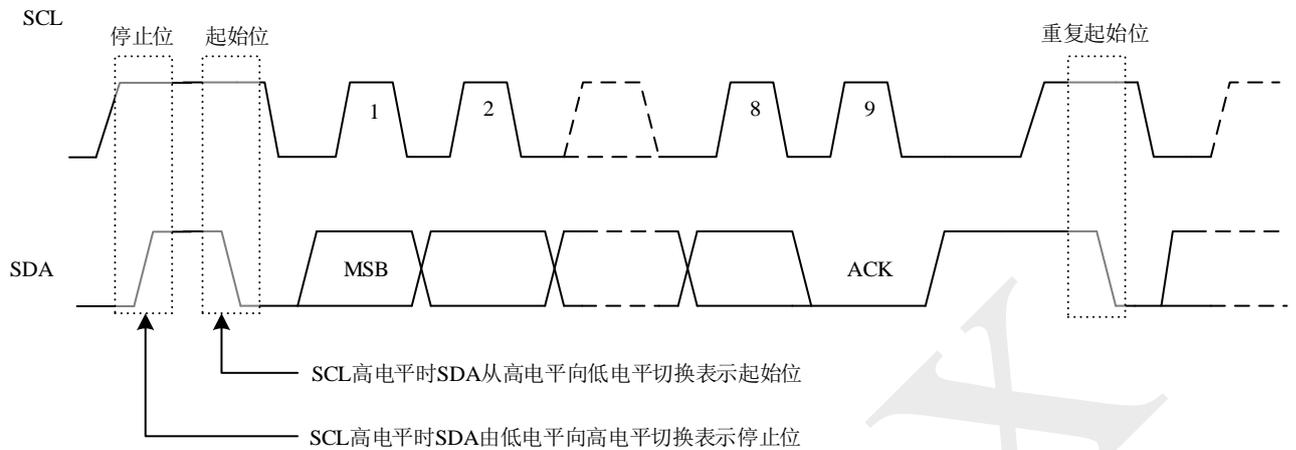
在从模式下, 该接口能够识别其自身地址 (7 或 10 位) 以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。

SDA 线上的数据必须在 SCL 时钟线高电平周期内保持稳定, 数据线的高或低电平状态只有在 SCL 线的时钟信号是低电平周期内才能改变, 数据和地址均以 8 位字节传输, MSB 在前。起始位后紧随地址字节 (7 位地址占据一个字节, 10 位地址占据两个字节), 地址始终在主模式下传送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲, 在此期间接收端必须向发送端

发送一个应答位。

图 28-2 I2C 总线协议



28.3.5 I2C 初始化

图 28-3 I2C 初始化流程



使能和关闭外设

首先在 RCC 中配置 I2C 的时钟源，并使能 I2C 时钟，然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I2C。

当禁止 I2C(PE 清 0)时，I2C 将执行软件复位。详细信息，请参见 [软件复位](#)。

噪声滤波器

提供可配置的数字噪声滤波器，用于实现输入信号滤波。

注意： I2C 使能状态下，不允许更改噪声滤波器配置。

- 数字滤波器

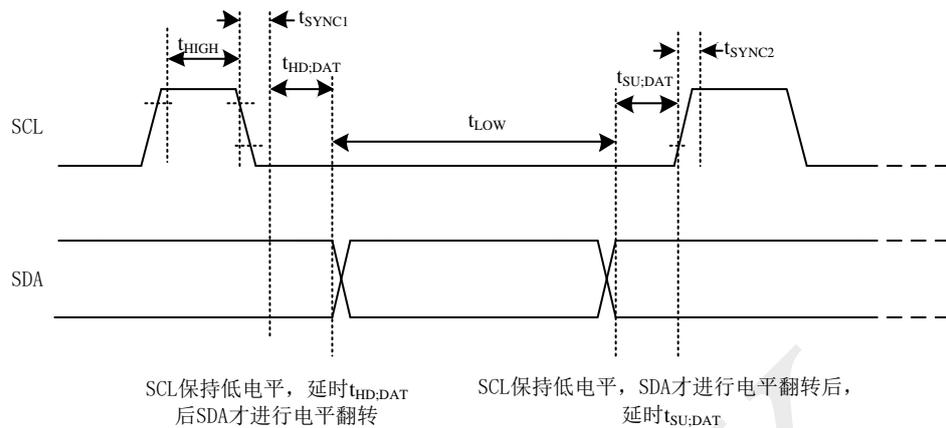
默认情况下，SDA 和 SCL 数字噪声滤波器是关闭的，通过配置 I2C_CR1 寄存器中的 DNF[3:0]位来使能数字滤波器。使能数字滤波器时，SCL 或 SDA 的电平只有在电平稳定时间超过 DNF 个 I2C 时钟周期后才会认为电平是有效地变化，从而可抑制的尖峰脉宽在 1~15 个 I2C 时钟周期的噪声。

I2C 时序

在主模式下必须配置时序，必须通过配置 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0]和 SCLL[7:0]位来配置 SCL 时钟的高电平和低电平，配置 SCLDEL[3:0]和 SDADEL[3:0]位以便保证主模式和从模式下使用正确的数据保持和建立时间，符合 I2C 规范中的时序要求。

表 28-3 I2C 规范时序要求

符号	参数	标准模式		快速模式		超快速模式		单位
		最小值	最大值	最小值	最大值	最小值	最大值	
t _{HD;DAT}	数据保持时间	0	-	0	-	0	-	us
t _{VD;DAT}	数据有效时间	-	3.45	-	0.9	-	0.45	us
t _{SU;DAT}	数据建立时间	250	-	100	-	50	-	ns
t _{HD;STA}	(重复) 起始位的保持时间	4.0	-	0.6	-	0.26	-	us
t _{SU;STA}	起始位的建立时间	4.7	-	0.6	-	0.26	-	us
t _{SU;STO}	停止位的建立时间	4.0	-	0.6	-	0.26	-	us
t _{BUF}	停止位和起始位之间的空闲时间	4.7	-	1.3	-	0.5	-	us
t _{HIGH}	SCL 低电平时间	4.0	-	0.6	-	0.26	-	us
t _{LOW}	SCL 高电平时间	4.7	-	1.3	-	0.5	-	us
t _r	上升沿时间	-	1000	-	300	-	120	ns
t _f	下降沿时间	-	300	-	300	-	120	ns

图 28-4 I2C_TIMINGR 寄存器配置生成 SCL 时序


当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时，该延时为 $t_{SDADEL} = (SDADEL + 2) \times t_{PRESC}$ ($t_{PRESC} = (PRESC + 1) \times t_{I2C_KCLK}$)，影响数据保持时间 $t_{HD;DAT}$ ，如图 28-4 I2C_TIMINGR 寄存器配置生成 SCL 时序。SDA 输出总延时为： $t_{SYNC1} + t_{SDADEL}$ 。 t_{SYNC1} 和 t_{SYNC2} 时间取决于以下参数：

- SCL 下降沿斜率
 - 数字滤波器使能后引入的延时 $t_{DNF} = DNF \times t_{I2C_KCLK}$
 - SCL 与 I2C_KCLK 时钟建立同步产生的 2-3 个 I2C_KCLK 时钟周期延时
- 用户在编程 SDADEL 时遵循以下条件：

$$\{t_{f(max)} + t_{HD;DAT(min)} - [(DNF + 3) \times t_{I2C_KCLK}]\} / t_{PRESC} \leq SDADEL$$

$$SDADEL \leq \{t_{HD;DAT(max)} - [(DNF + 4) \times t_{I2C_KCLK}]\} / t_{PRESC}$$

标准模式、快速模式和超快速模式下的 $t_{HD;DAT}$ 的最大值分别为 3.45us、0.9us 和 0.45us，但必须小于 $t_{VD;DAT}$ 最大值。只有器件未延长 SCL 信号的低电平时间时，才能满足最大值条件。如果延长 SCL，数据必须在建立时间内保持有效，之后才能释放 SCL。在这种情况下，上述公式变为如下形式：

$$SDADEL \leq \{t_{VD;DAT(max)} - t_{r(max)} - [(DNF + 4) \times t_{I2C_KCLK}]\} / t_{PRESC}$$

注意： $NOSTRETCH=0$ 时会违反该条件，器件会根据 SCLDEL 值来延长 SCL 低电平是时间，保证数据建立时间。

在 t_{SDADEL} 延时后，在数据未写入 I2C_TXDR 寄存器而导致从设备必须延长时钟的情况下发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 $t_{SCLDEL} = (SCLDEL + 3) \times t_{PRESC}$ ($t_{PRESC} = (PRESC + 1) \times t_{I2C_KCLK}$)，影响数据建立时间 $t_{SU;DAT}$ ，如 [图 I2C_TIMINGR 寄存器配置生成 SCL 时序](#)。用户在编程 SCLDEL 时遵循以下条件：

$$(t_{r(max)} + t_{SU;DAT(min)} / t_{PRESC}) - 1 \leq SCLDEL$$

注意： 在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主设备或从设备会至少在 $(SDADEL+SCLDEL+1) \times t_{PRESC}$ 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。

如果从模式下 NOSTRETCH=1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

通过配置 SCLH[7:0]来设置 SCL 高电平时间：当内部检测到 SCL 上升沿时，会将 SCL 释放为高电平，高电平时间为 $t_{SCLH}=(SCLH+3) \times t_{PRESC}$ ($t_{PRESC}=(PRESC+1) \times t_{I2C_KCLK}$)，影响高电平时间 t_{HIGH} ，如 [图 I2C_TIMINGR 寄存器配置生成 SCL 时序](#)。

通过配置 SCLL[7:0]来设置 SCL 低电平时间：当内部检测到 SCL 下降沿时，会拉低 SCL，低电平时间为 $t_{SCLL}=(SCLL+3) \times t_{PRESC}$ ($t_{PRESC}=(PRESC+1) \times t_{I2C_KCLK}$)，影响低电平时间 t_{LOW} ，如 [图 I2C_TIMINGR 寄存器配置生成 SCL 时序](#)。

注意： 使能 I2C 后，不允许更改时序配置。

I2C_TIMINGR 寄存器配置示例

下表提供 I2C_TIMINGR 寄存器初始化时配置的值，来实现 I2C 不同的通讯速率。

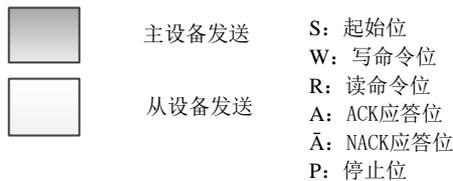
表 28-4 I2C_KCLK=18MHz 配置示例

参数	100KHz	400KHz	1MHz
PRESC[3:0]	3	1	1
t _{PRESC}	215ns	108ns	108ns
SCLDEL[3:0]	0x04	0x00	0x00
t _{SCLDEL}	6×215ns=1290ns	2×108ns=216ns	2×108ns=216ns
SDADEL[3:0]	0x01	0x01	0x00
t _{SDADEL}	3×215ns=645ns	3×108ns=324ns	2×108ns=216ns
SCLH[7:0]	0x11	0x04	0x01
t _{SCLH}	19×215ns=4085ns	6×108ns=648ns	3×108ns=324ns
SCLL[7:0]	0x12	0x0b	0x03
t _{SCLL}	20×215ns=4250ns	13×108ns=1404ns	5×108ns=540ns

地址模式

I2C 有 7 位地址模式和 10 位地址寻址模式，在总线上两种地址模式兼容。

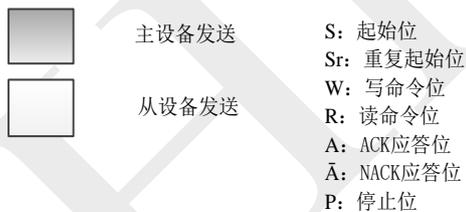
图 28-5 7 位地址读写过程
7位地址写数据过程

7位地址读数据过程

图 28-6 10 位地址读写过程
10位地址写数据过程


11110XX

10位地址读数据过程


11110XX



I2C 模块通过地址寄存器 OAR1 和地址寄存器 OAR2 支持 7 位地址模式和 10 位地址模式。

地址寄存器 1 支持 7 位和 10 位地址模式，OA1MODE 为 1 时，OA1[9:0]是 10 位地址模式的从模式地址；OA1MODE 为 0 时，OA1[7:1]是 7 位地址模式的从模式地址。

地址寄存器 2 只支持 7 位地址模式，OA2[7:1]是 7 位地址模式的从模式地址。可配置 OA2MSK[2:0]的值，调整地址匹配的地址位数。如果 OA2MSK[2:0]的值不为 0，即使地址匹配也不会对 I2C 规范的保留地址(0b0000xxx 和 0b1111xxx)应答。

通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

时钟低电平延长

- 使能时钟延长的从模式 (NOSTRETCH 为 0)

在默认模式下，I2C 从设备会在以下情况下延长 SCL 时钟：

ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDR CF 位置 1 以清除 ADDR 标志时，将释放该时钟延长。

发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清 0 且 TXE 为 1 时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延长。

接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延长。

当 SBC 置 1 从设备字节控制模式且 RELOAD 置 1 重载模式 TCR 置 1 时，最后一个数据字节已传输完成。通过向 NBYTES[7:0]位域写入一个非 0 值将 TCR 清 0，将释放该时钟延长。

- 禁止时钟延长的从模式 (NOSTRETCH 为 1)

当 I2C_CR1 寄存器中的 NOSTRETCH 为 1 时，I2C 从设备不会延长 SCL 时钟信号。

ADDR 标志置 1 时，不会延长 SCL 时钟信号。

发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，同时会生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清 0）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清 0 上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。

接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，同时会生成中断。

28.3.6 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清 0 来执行软件复位，必须使 PE 为 0 持续至少 3 个 APB 时钟周期，才能成功执行软件复位。在这种情况下，SCL 和 SDA 被释放为高电平，内部状态机复位，通信控制位和状态位恢复为复位值。下面列出了受影响的寄存器位：

- I2C_CR2 寄存器：START、STOP 和 NACK。

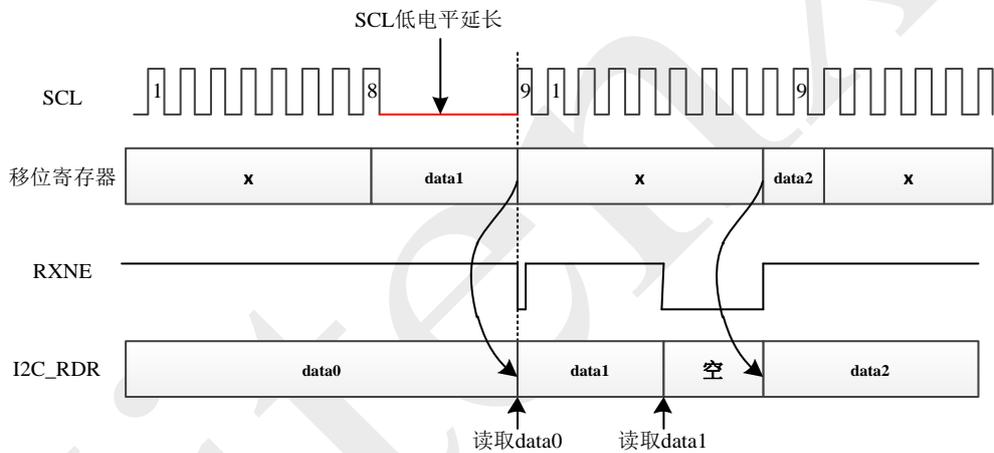
- I2C_ISR 寄存器: BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR。

28.3.7 数据传输

接收

SDA 接收的数据输入到移位寄存器, 接收到完整的数据字节时在第 8 个 SCL 脉冲后, 如果 I2C_RXDR 寄存器为空(RXNE 为 0), 则移位寄存器的内容会复制到其中。如果尚未读 I2C_RXDR 中接收的数据字节 (RXNE 为 1), 则将在第 8 个和第 9 个 SCL 脉冲之间 (应答脉冲之前) 延长 SCL 的低电平时间, 直到读取了 I2C_RXDR 的数据为止。

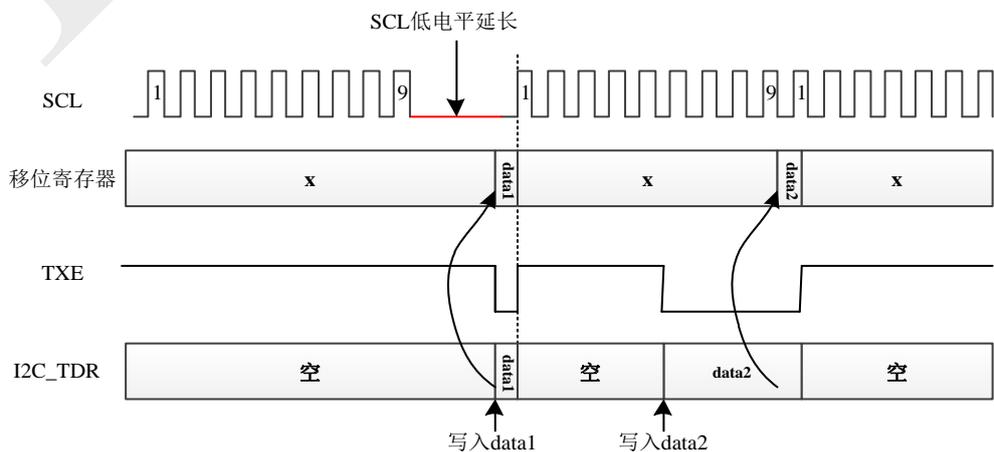
图 28-7 数据接收



发送

如果 I2C_TXDR 寄存器非空(TXE 为 0), 则其内容会在第 9 个 SCL 脉冲 (应答脉冲) 后复制到移位寄存器中。然后移位寄存器的内容会依次发送到 SDA 线上。如果 I2C_TXDR 内尚未写入任何数据 (TXE 为 1), 则将在第 9 个 SCL 脉冲后延长 SCL 线的低电平时间, 直到待发送数据写入了 I2C_TXDR 为止。

图 28-8 数据发送



硬件管理传输

I2C 模块内置了字节计数器，以便在主从模式下管理字节传输和结束通信：

- 主模式

- 接收最后一字节数据时发送 NACK
- 接收或发送数据后发送停止位
- 数据传输完成拉低 SCL，可配置 START 位发送重复起始位

- 从模式

从模式字节计数器默认为禁止状态，通过将 I2C_CR1 寄存器中的 SBC（从模式字节控制）位置 1 来使能。

- 接收每字节数据后拉低 SCL，可配置 NACK 位发送 ACK 或 NACK

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0]位进行编程。如果待传输的字节数大于 255，须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1，在该模式下，完成 NBYTES[7:0]中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会拉低等待，当向 NBYTES[7:0]写入一个非 0 值，TCR 标志清 0，释放 SCL。

注意：在向 NBYTE 中写入最后一次传输的字节数前，必须把 RELOAD 位清 0。

当主模式下 RELOAD 为 0，自动结束模式（I2C_CR2 寄存器中的 AUTOEND 为 1）时，一旦完成 NBYTES[7:0]中所写入字节数的数据传输，主设备便会自动发送停止位。

当主模式下 RELOAD 为 0，软件结束模式（I2C_CR2 寄存器中的 AUTOEND 为 0）时，完成 NBYTES[7:0]中字节数的数据传输后，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 便会拉低等待，需要软件操作释放 SCL。把 I2C_CR2 寄存器中的 START 或 STOP 置 1 时，TC 标志将被清 0。当主设备要发送重复起始位时，须使用该模式。

注意：当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

28.3.8 从模式

从模式发送数据时序

当 I2C_TXDR 寄存器为空时，TXIS 将置 1，如果 I2C_CR1 寄存器中的 TXIE 位置 1，同时会生成中断。I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清 0。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，同时会生成中断，从设备自动释放 SCL 和 SDA。收到

NACK 时，TXIS 位不会置 1。

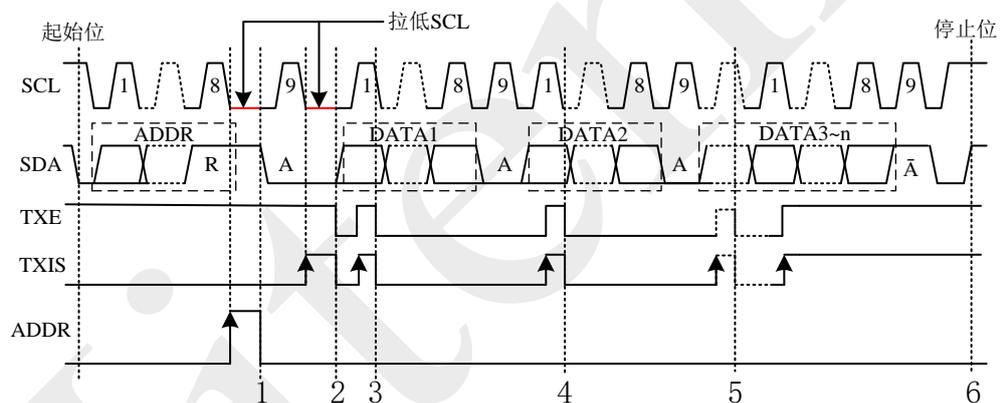
当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。

如果地址匹配时（ADDR 置 1）TXE 为 0，可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器发送新的数据字节。如果 NOSTRETCH 置 1，当 ADDR 标志置 1 时不会延长 SCL，无法在 ADDR 中断服务程序中刷新 I2C_TXDR 寄存器的内容（更新第一个数据字节），必须在 I2C_TXDR 寄存器中提前写入待发送的第一个数据。

如果需要 TXIS 事件（发送中断或发送 DMA 请求），将 TXE 位和 TXIS 位均置 1，生成 TXIS 事件。

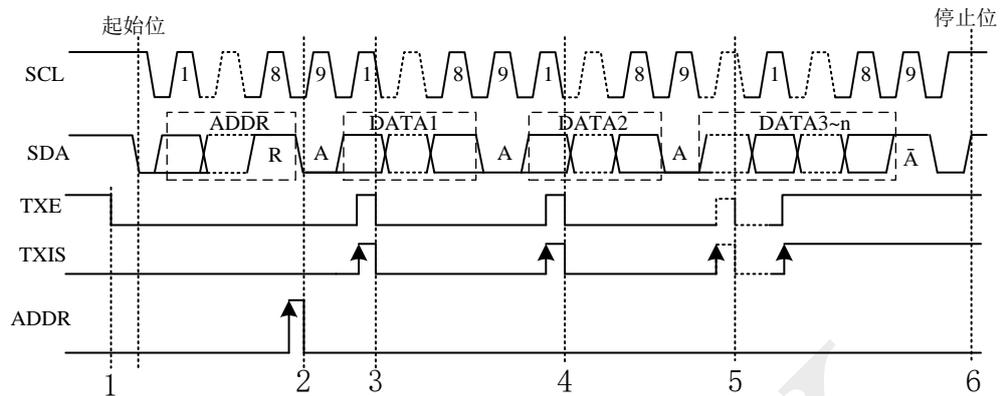
使能低电平延长（NOSTRETCH 清 0）发送数据时序图：

图 28-9 从模式使能时钟低电平延长发送数据



1. 查询ADDRCODE[6:0]和DIR值，确认匹配的地址和数据传输方向，ADDRCF置1清除ADDR标志；
2. DATA1写入I2C_TXDR寄存器；
3. DATA2写入I2C_TXDR寄存器；
4. DATA3写入I2C_TXDR寄存器；
5. DATA4~DATA_n写入I2C_TXDR寄存器；
6. 清除STOPF标志，结束本次数据传输。

禁止低电平延长功能（NOSTRETCH 置 1）发送数据时需要在地地址匹配事件前将待发送的第一个字节数据写入 I2C_TXDR 寄存器，序图：

图 28-10 从模式禁止时钟低电平延长发送数据


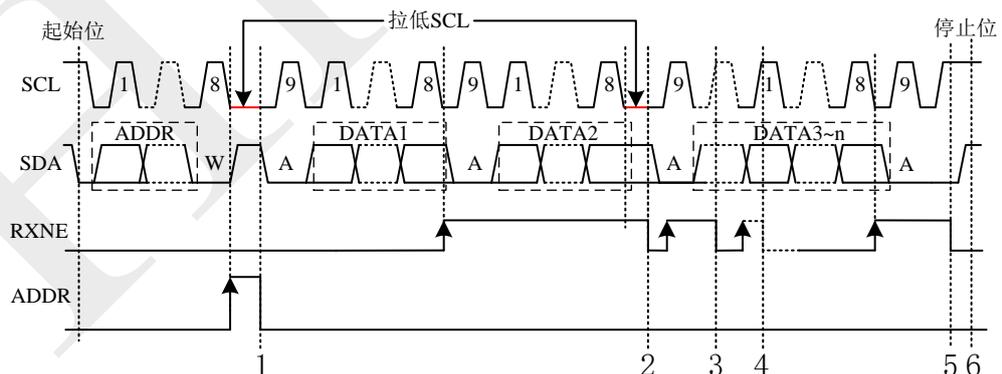
- 1.DATA1写入I2C_TDR寄存器;
- 2.查询ADDCODE[6:0]和DIR值, 确认匹配的地址和数据传输方向, ADDRRCF置1清除ADDR标志;
- 3.DATA2写入I2C_TXDR寄存器;
- 4.DATA3写入I2C_TXDR寄存器;
- 5.DATA4~DATAn写入I2C_TXDR寄存器;
- 6.清除STOPF标志, 结束本次数据传输。

从模式接收数据时序

I2C_RXDR 寄存器接收到数据, I2C_ISR 中的 RXNE 将置 1, 如果 I2C_CR1 中的 RXIE 置 1, 同时会生成中断。读取 I2C_RXDR 时, 将清 0 RXNE。

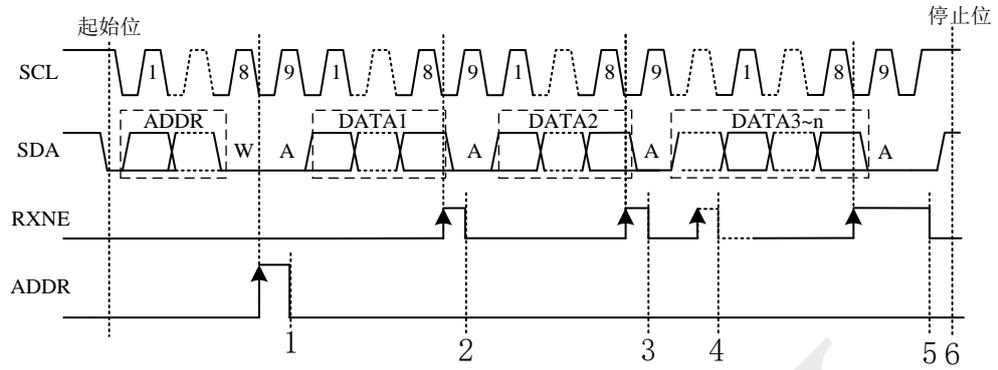
I2C_CR1 寄存器中的 STOPIE 置 1, 当接收到停止位, I2C_ISR 中的 STOPF 将置 1 并生成中断。

使能低电平延长功能 (NOSTRETCH 为 0) 接收数据时序图:

图 28-11 从模式使能时钟低电平延长接收数据


- 1.查询ADDCODE[6:0]和DIR值, 确认匹配的地址和数据传输方向, ADDRRCF置1清除ADDR标志;
- 2.从I2C_RXDR寄存器读取DATA1;
- 3.从I2C_RXDR寄存器读取DATA2;
- 4.从I2C_RXDR寄存器读取DATA3~DATA[n-1];
- 5.从I2C_RXDR寄存器读取DATAn;
- 6.清除STOPF标志, 结束本次数据传输。

禁止低电平延长功能 (NOSTRETCH 置 1) 接收数据时序图:

图 28-12 从模式禁止时钟低电平延长接收数据


1. 查询ADDCODE[6:0]和DIR值，确认匹配的地址和数据传输方向，ADDRCF置1清除ADDR标志；
2. 从I2C_RXDR寄存器读取DATA1；
3. 从I2C_RXDR寄存器读取DATA2；
4. 从I2C_RXDR寄存器读取DATA3~DATA[n-1]；
5. 从I2C_RXDR寄存器读取DATA_n；
6. 清除STOPF标志，结束本次数据传输。

28.3.9 主模式

要发起通信，必须配置 I2C_CR2 寄存器中以下标志位参数：

- 7 位或 10 位寻址模式：ADD10；
- 待发送的从地址：SADD[9:0]；
- 传输方向：RD_WRN；
- 读取 10 位地址时：必须对 HEAD10R 进行相应配置，以指示传输方向变化时发送完整的地址序列，还是只发送 10 位地址高字节。

图 28-13 HEAD10R 置 1 或清 0 在 10 位地址模式先发送再接收数据过程中的作用

HEAD10R为0先发送数据再接收数据过程



HEAD10R置1先发送数据再接收数据过程



主设备发送



从设备发送

S: 起始位
 Sr: 重复起始位
 W: 写命令位
 R: 读命令位
 A: ACK应答位
 Ā: NACK应答位
 P: 停止位

- 待传输的字节数：NBYTES[7:0]。

然后，用户将 I2C_CR2 寄存器中的 START 位置 1，START 位置 1 时，不允许更改上述所有控制位。之后，当主设备检测到总线空闲时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从设备地址。

注意： 无论接收到的 ACK 还是 NACK，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。如果当 START 位置 1 时，I2C 作为从设备被寻址（ADDR 置 1），则 I2C 将切换为从模式，START 位将在 ADDR_{CF} 位置 1 时清 0。该过程同样适用于重复起始位，在此过程中，BUSY 置 1。

主模式发送数据时序

发送数据时，在发送完每个字节接收到 ACK 后，TXIS 标志将置 1，如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入下一个待发送的数据时，该标志将被清 0。收到 NACK 时，TXIS 标志不会置 1。

字节计数器 NBYTES[7:0]可配置的最大值为 255，如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0]数据传输完成时，TCR 标志将置 1，并且

SCL 线的低电平将被延长，直到 NBYTES[7:0] 被写入非 0 值。

- 当 RELOAD 为 0 且 NBYTES[7:0] 数据传输完成：

自动结束模式 (AUTOEND 置 1)，将自动发送停止位。

软件结束模式 (AUTOEND 为 0)，TC 标志将置 1 且延长 SCL 低电平时间，执行以下操作：

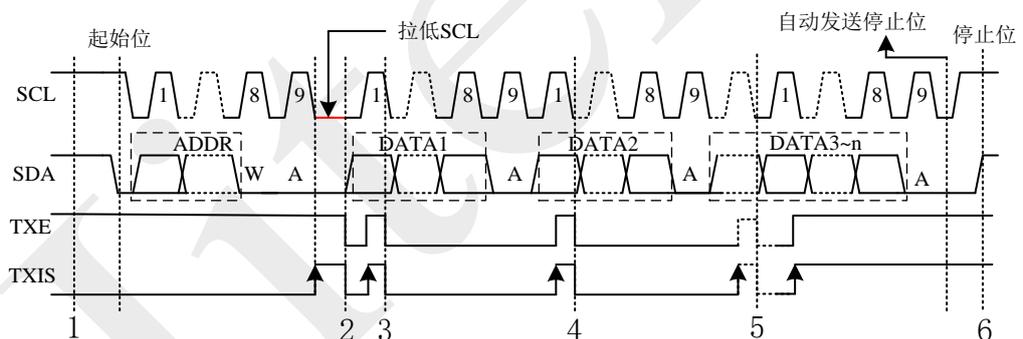
- 可将 I2C_CR2 寄存器中的 START 位置 1 发送重复起始位和从地址。将 START 位置 1 会将 TC 标志清 0；
- 可将 I2C_CR2 寄存器中的 STOP 位置 1 来停止位。将 STOP 位置 1 会将 TC 标志清 0。

- 接收到 NACK：

TXIS 标志不会置 1，自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

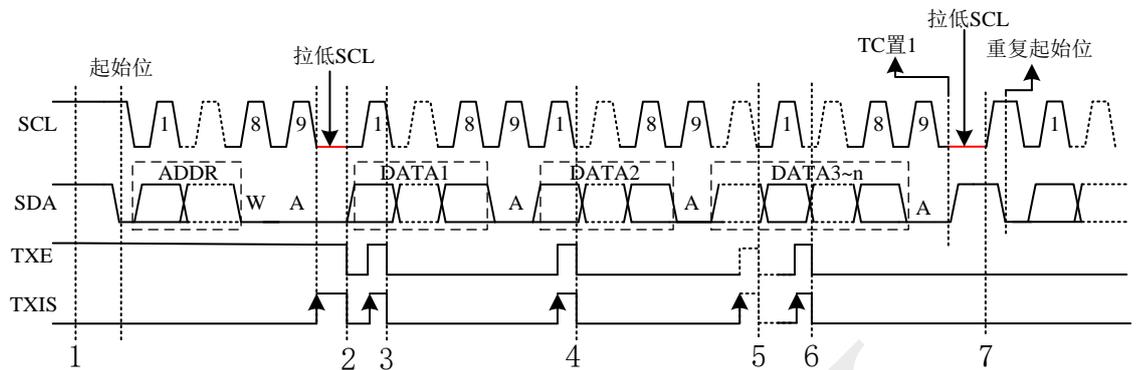
自动结束模式发送数据时序图 (AUTOEND 置 1)：

图 28-14 主模式自动结束模式发送数据时序



1. 设置从设备地址，设置发送字节数写入 NBYTES[7:0]，设置自动结束模式 (AUTOEND=1)，START 位置 1 发送起始位；
2. DATA1 写入 I2C_TXDR 寄存器；
3. DATA2 写入 I2C_TXDR 寄存器；
4. DATA3 写入 I2C_TXDR 寄存器；
5. DATA4~DATA_n 写入 I2C_TXDR 寄存器；
6. 清除 STOPF 标志，结束本次数据传输。

软件结束模式发送数据时序图 (AUTOEND 清 0)：

图 28-15 主模式软件结束模式发送数据时序


1. 设置从设备地址，设置发送字节数写入NBYTES[7:0]，设置软件结束模式（AUTOEND清0），START位置1发送起始位；
2. DATA1写入I2C_TXDR寄存器；
3. DATA2写入I2C_TXDR寄存器；
4. DATA3写入I2C_TXDR寄存器；
5. DATA4~DATA_n写入I2C_TXDR寄存器；
6. 将下次待发送的数据写入I2C_TXDR寄存器；
7. 将START位置1，发送重复起始位，进行下次数据的传输。

主模式接收数据时序

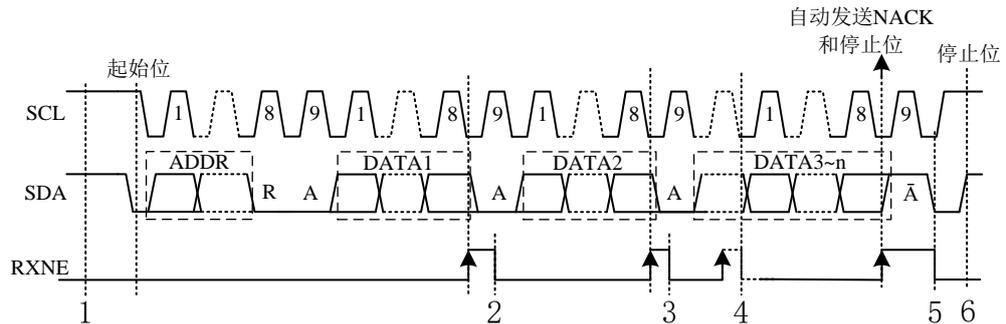
读数据时，在接收到每个字节的第 8 个 SCL 脉冲后，RXNE 标志将置 1，如果 I2C_CR1 寄存器中的 RXIE 位置 1，同时会生成中断。读取 I2C_RXDR 时，将清 0 该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0]数据传输完成时，TCR 标志将置 1，并且延长 SCL 的低电平时间，NBYTES[7:0]写入非 0 值，释放 SCL。

当 RELOAD 为 0 且 NBYTES[7:0]数据传输完成时：

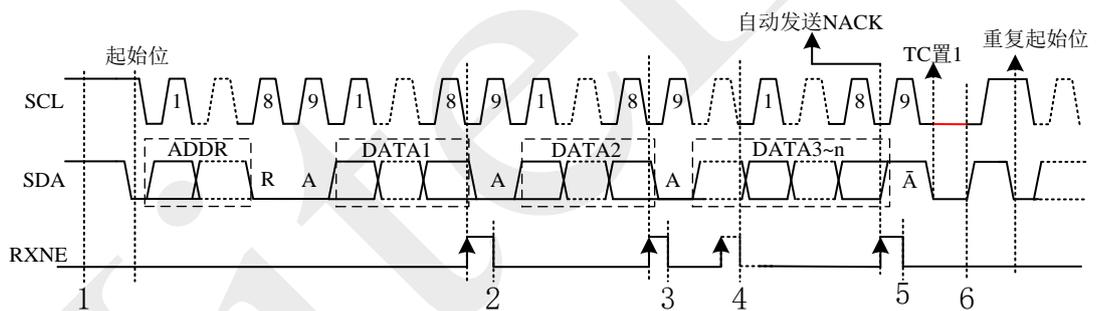
- 自动结束模式（AUTOEND 置 1）接收到最后一个字节后，将自动发送 NACK 和停止位。
- 软件结束模式（AUTOEND 为 0），接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1，延长 SCL 的低电平时间，以便执行以下软件操作：
 - 可将 I2C_CR2 寄存器中的 START 位置 1 发送重复起始位和从地址，将 START 位置 1 会将 TC 标志清 0；
 - 可将 I2C_CR2 寄存器中的 STOP 位置 1 来停止位。将 STOP 位置 1 会将 TC 标志清 0。

自动结束模式发送数据时序图（AUTOEND 置 1）：

图 28-16 主模式自动结束模式接收数据时序


1. 设置从设备地址，设置接收字节数写入NBYTES[7:0]，设置自动结束模式（AUTOEND=1），START位置1发送起始位；
2. 从I2C_RXDR寄存器读取DATA1；
3. 从I2C_RXDR寄存器读取DATA2；
4. 从I2C_RXDR寄存器读取DATA3~DATA[n-1]；
5. 从I2C_RXDR寄存器读取DATA_n；
6. 清除STOPF标志，结束本次数据传输。

软件结束模式发送数据时序图（AUTOEND 清 0）：

图 28-17 主模式软件结束模式接收数据时序


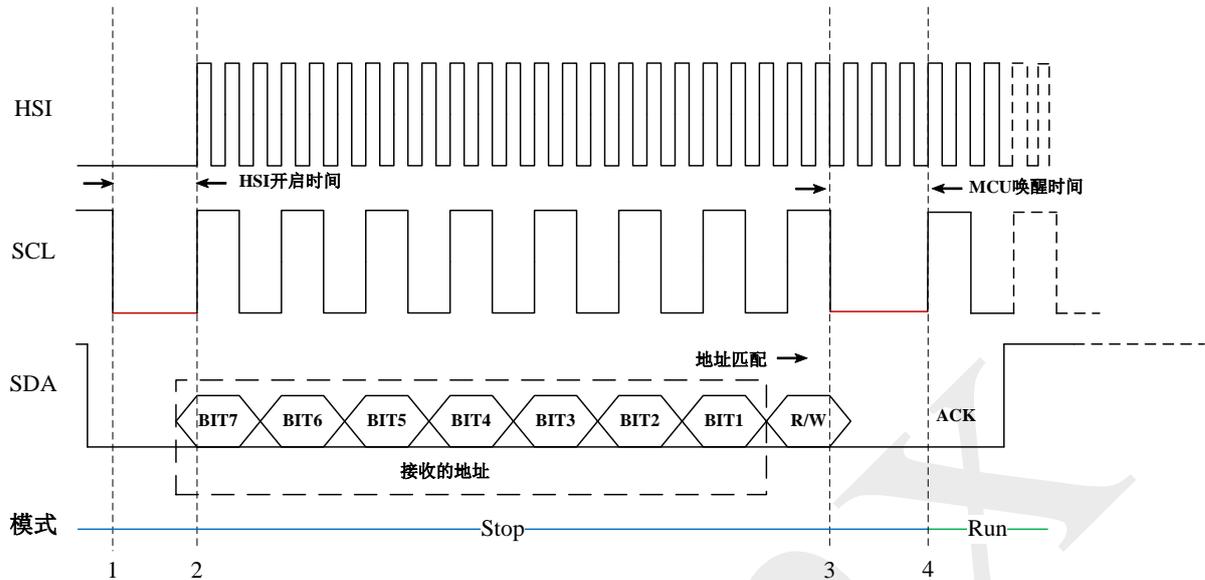
1. 设置从设备地址，设置接收字节数写入NBYTES[7:0]，设置软件结束模式（AUTOEND清0），START位置1发送起始位；
2. 从I2C_RXDR寄存器读取DATA1；
3. 从I2C_RXDR寄存器读取DATA2；
4. 从I2C_RXDR寄存器读取DATA3~DATA[n-1]；
5. 从I2C_RXDR寄存器读取DATA_n；
6. START置1发送重启位和从设备地址，进行后续数据传输。

28.3.10 低功耗模式唤醒

从模式下，当被寻址且地址匹配时，能够将 MCU 从 Stop 模式唤醒，支持以下寻址模式：

- 接收的地址与地址寄存器 1 或地址寄存器 2 匹配；
- 使能广播地址响应（I2C_CR1 寄存器的 GCEN 置 1），接收到广播地址；

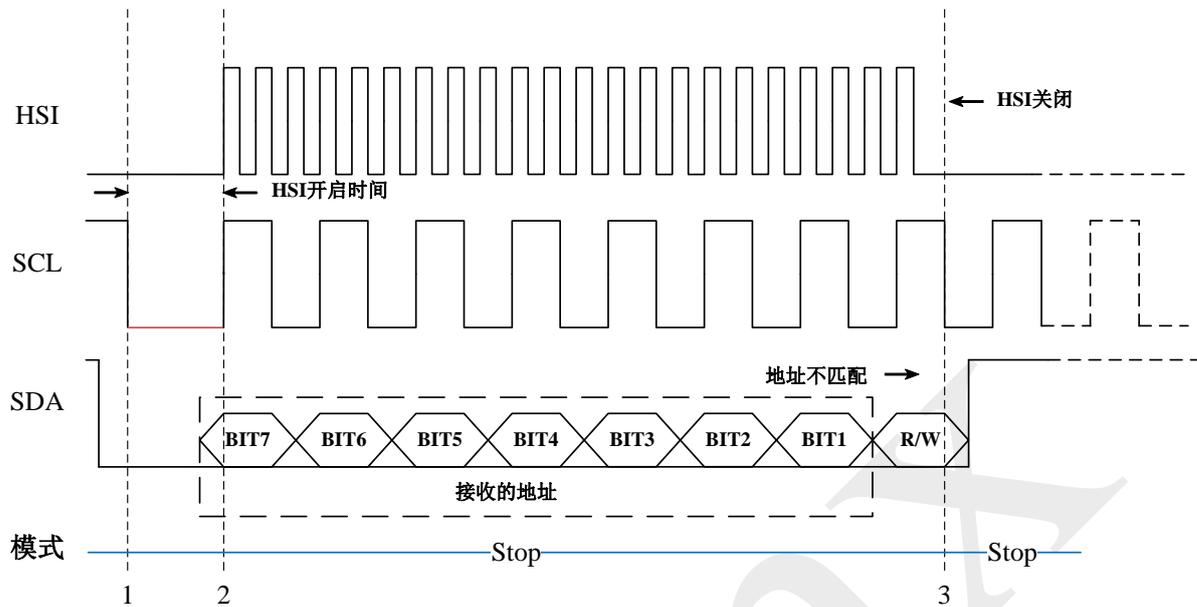
选择 HSI 作为 I2C_KCLK 时钟源，将 I2C_CR1 寄存器中 NOSTETCH 位清 0 使能 SCL 低电平延长功能，WUPEN 位置 1 使能唤醒功能，可以使能从 Stop 模式唤醒功能。

图 28-18 低功耗模式唤醒时序


- 1.从设备接收到起始信号，拉低SCL，启动FRC时钟；
- 2.HSI启动后释放SCL，接收地址；
- 3.接收的地址匹配，拉低SCL唤醒MCU；
- 4.MCU唤醒完成，释放SCL，进行后续的处理。

如上图，在 Stop 模式时，HSI 关闭。当检测到起始位，主设备拉低 SCL 后，I2C 外设将拉低 SCL 请求开启 HSI 时钟，直到 HSI 唤醒释放 SCL，进行地址接收。地址匹配后，再次拉低 SCL，唤醒整个 MCU 系统。通过软件清除 ADDR 标志释放 SCL，进行后续操作。

如果接收的地址不匹配，HSI 再次关闭，MCU 不被唤醒，如下图所示。

图 28-19 低功耗模式未唤醒时序


1. 从设备接收到起始信号，拉低SCL，启动HSI时钟；
2. HSI启动后释放SCL，接收地址；
3. 接收的地址不匹配，关闭HSI，未唤醒MCU。

注意： 如果I2C的时钟是SYS_CLK或PCLK，或者WUPEN为0，则接收到起始位后，HSI不会开启。必须使能时钟延长（NOSTRETCH为0）才能确保从停止模式唤醒功能正常。

注意： 使能低功耗唤醒功能进入Stop模式前，必须关闭数字滤波器。否则不能唤醒MCU。

注意： 从模式地址接收阶段，地址未接收完成，HSI将一直处于开启的状态，直到地址接收完成；接收到起始位或重复起始位将重新接收地址；接收到停止位结束本次的地址接收。

28.3.11 DMA 传输

DMA 发送

将I2C_CR1寄存器中的TXDMAEN位置1来使能DMA（请参见：[直接存储器访问控制器（DMA）](#)）发送。当TXIS位置1时，DMA将源地址的数据写入I2C_TXDR寄存器。

DMA只能传输数据，不能传输地址、应答位、起始位和停止位：

- 主模式：从地址、和起始位均通过操作I2C寄存器实现（无法通过DMA实现）。必须在START置1之前初始化DMA，传输结束由NBYTES计数器来管理。
- 从模式：

- 当 NOSTRETCH 为 0 时，需要在地址匹配事件之前初始化 DMA，或在地址匹配中断程序中清除 ADDR 标志前初始化 DMA；
- 当 NOSTRETCH 为 1 时，必须在地址匹配事件之前初始化 DMA。

注意： 如果使用 DMA 进行发送，则无需使能 TXIE 位。

DMA 接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 来使能 DMA（请参见：[直接存储器访问控制器 \(DMA\)](#)）接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器写入 DMA 目的地址。数据字节采用 DMA 进行传输。

- 主模式：从地址、和起始位均通过操作 I2C 寄存器实现（无法通过 DMA 实现）。必须在 START 置 1 之前初始化 DMA，传输结束由 NBYTES 计数器来管理。
- 从模式：
 - 当 NOSTRETCH 为 0 时，需要在地址匹配事件之前初始化 DMA，或在地址匹配中断程序中清除 ADDR 标志前初始化 DMA；
 - 当 NOSTRETCH 为 1 时，必须在地址匹配事件之前初始化 DMA。

注意： 如果使用 DMA 进行接收，则无需使能 RXIE 位。

28.3.12 错误条件

以下错误条件会导致通信失败。

- 总线错误

主设备或地址匹配后的从设备进入数据传输阶段，在每个字节的 9 个脉冲期间检测到起始位（参见[通讯流程](#)），生成总线错误。检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，同时会生成中断。

在主模式地址发送阶段，在每个字节的 9 个脉冲期间检测到起始位会生成总线错误，BERR 标志置 1。

在从模式地址接收阶段，检测到起始位或重复起始位时，将重新接收地址，接收到停止位结束本次地址接收，但 BERR 标志不会置 1。

由于 I2C 通讯流程中要求在 SCL 为高电平期间，SDA 保持电平稳定。在数据传输过程中：若 SCL 高电平期间 SDA 出现下降沿则为起始位产生的总线错误；若 SCL 高电平期间 SDA 出现上升沿则为停止位产生的总线错误。

主模式数据传输阶段检测到起始位，BERR 置 1 生成总线错误，复位

内部状态机，清空移位寄存器，开始地址接收。

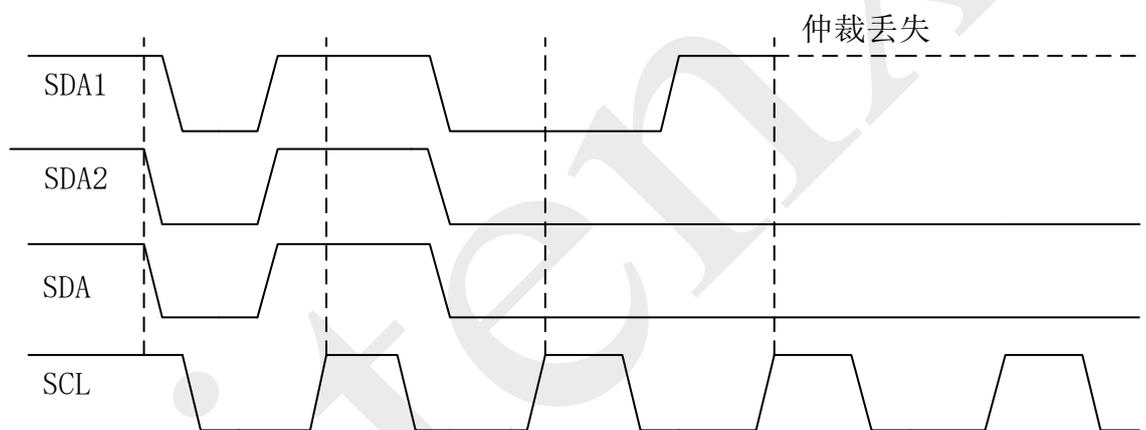
从模式数据传输阶段检测到起始位，**BERR** 置 1 生成总线错误，复位内部状态机，清空移位寄存器，开始地址接收。如果地址接收未完成（未接收到 8bit 数据），则一直处于地址接收模式，等待地址接收完成。

从模式数据传输阶段检测到停止位，**BERR** 置 1 生成总线错误，停止数据传输，复位内部状态机，等待接收下一个的起始位。

● 仲裁丢失

当多个主设备同时在空闲总线上发送起始位进行数据传输，需通过时钟同步和仲裁，来决定哪个主设备获得总线控制权。仲裁过程如下图：

图 28-20 I2C 总线仲裁



仲裁的时钟位为同步后的时钟 **SCL**，两个主设备可能在起始位的最短保持时间内在总线上产生一个有效的起始位，仲裁逐位进行，在每一位的仲裁期间，当 **SCL** 为高时，每个主机都检查 **SDA** 的电平是否和自己发送的相同。仲裁的过程可能需要持续很多位。如果一个主设备发送高电平但检测到 **SDA** 电平为低，则认为自己仲裁丢失并关闭自己的 **SDA** 输出驱动，而另一个主机则继续完成自己的传输。

当检测到仲裁丢失时，**I2C_ISR** 寄存器中的 **ARLO** 标志将置 1，如果 **I2C_CR1** 寄存器中的 **ERRIE** 位置 1，同时会生成中断。

- 主模式：仲裁发生在地址发送阶段、数据传输阶段和应答阶段；当检测到仲裁丢失时 **SDA** 和 **SCL** 被释放，起始位由硬件清 0，主设备自动切换为从模式。
- 从模式：仲裁在数据阶段和应答阶段；当检测到仲裁丢失时传输停止，**SCL** 和 **SDA** 被释放。

● 上溢、下溢错误

仅在从模式下且 **I2C_CR1** 寄存器 **NOSTRETCH** 为 1 时会检测到上溢或下

溢错误。当检测到溢出错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，同时会生成中断。

当接收到一个新的字节，但 RDR 寄存器的值还未被读取时，将产生上溢错误，接收的新字节丢弃，自动发送 NACK 来响应丢弃的字节。

在发送过程中，出现下列情况将产生下溢错误：

- 当上一次数据传输结束标志尚未清除（STOPF 为 1），本次地址匹配后应发送第一个数据字节时，如果 I2C_TXDR 寄存器有数据则发送该数据，否则发送 0xFF；
- 当主设备开始读取数据，但从设备发送的下一个字节尚写入 I2C_TXDR 寄存器数据时，将发送 0xFF。

● 超时错误

主模式检测到超时，将自动发送停止位。

从模式检测到超时，将自动释放 SDA 和 SCL。

28.4 低功耗模式说明

不同低功耗模式下 I2C 外设的状态如下表所示。

表 28-5 低功耗模式说明

模式	说明
Sleep 和 LPSleep	对 I2C 通信无影响，I2C 中断可使器件退出 Sleep 和 LPSleep
Stop	I2C 外设的寄存器仍被保持，地址匹配时可唤醒（参见 低功耗模式唤醒 ）

28.5 中断

I2C 中断请求如下：

表 28-6 中断请求和状态清除

中断事件	事件标志	中断使能位	事件标志或中断清除方法	从 Sleep 或 LPSleep 模式唤醒	从 Stop 模式唤醒
接收数据寄存器非空	RXNE	RXIE	读出 I2C_RXDR 寄存器的数据	√	×
发送数据寄存器中断状态	TXIS	TXIE	向 I2C_TXDR 寄存器写入数据	√	×
接收到停止位	STOPF	STOPF	I2C_ICR 中 STOPCF 置 1	√	×
传输完成等待重载	TCR	TCIE	I2C_CR2 寄存器 NBYTES[7:0] 写入非 0 值	√	×

中断事件	事件标志	中断使能位	事件标志或中断清除方法	从 Sleep 或 LPSleep 模式唤醒	从 Stop 模式唤醒
传输完成	TC		I2C_CR2 寄存器 START 置 1 或 STOP 置 1	√	×
地址匹配	ADDR	ADDRIE	I2C_ICR 寄存器 ADDRCF 置 1	√	√ ⁽²⁾
接收到 NACK	NACKF	NACKIE	I2C_ICR 寄存器 NACKCF 置 1	√	×
总线错误	BERR	ERRIE	I2C_ICR 寄存器 BERRCF 置 1	√	×
仲裁丢失	ARLO		I2C_ICR 寄存器 ARLOCF 置 1	√	×
上溢、下溢错误	OVR		I2C_ICR 寄存器 OVRCF 置 1	√	×

- “√”表示可以唤醒，“×”表示不能唤醒；
- 只有 I2C1 的地址匹配事件能够从 Stop 模式唤醒 MCU；

28.6 I2C 寄存器

28.6.1 控制寄存器 1 (I2C_CR1)

偏移地址：0x00

复位值：0x0000 0000

访问：由于时钟同步，写访问最长生效时间可达 $3 \times t_{PCLK1} + 4 \times t_{I2C_KCLK}$ 。当对该寄存器执行写访问未完成后，再次执行写访问，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												GCEN	WUPEN	NO STRETCH	SBC
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.		DNF[3:0]				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	GCEN	广播地址使能 0: 禁止广播地址，不响应0b00000000。 1: 使能广播地址，响应0b00000000。
18	WUPEN	从Stop模式唤醒使能 0: 禁止从Stop模式唤醒功能。

		1: 使能从Stop模式唤醒功能。 <i>注意: 如果不支持唤醒功能, 该位保持复位值。 只有关闭数字滤波器 (DNF位为0b0000) 时, 该位才能置1。</i>
17	NOSTRETCH	时钟低电平延长控制 该位用于在从模式下禁止时钟延长, 在主模式下必须保持为0。 0: 使能时钟低电平延长。 1: 禁止时钟低电平延长。 <i>注意: 该位只能在I2C禁止状态 (PE为0) 时写入。</i>
16	SBC	从模式字节计数控制使能 该位用在从模式下使能硬件字节计数器NBYTES[7:0]。 0: 禁止从模式下硬件字节计数器。 1: 使能从模式下硬件字节计数器。
15	RXDMAEN	DMA接收请求使能 0: 禁止DMA接收请求 1: 使能DMA接收请求
14	TXDMAEN	DMA发送请求使能 0: 禁止DMA发送请求 1: 使能DMA发送请求
13:12	保留	必须保持复位值。
11:8	DNF[3:0]	数字噪声滤波器 配置SCL和SDA输入端的数字噪声滤波器功能, 可滤除的脉宽为小于DNF[3:0]×I2C_KCLK以下的脉冲。 0000: 禁止数字噪声滤波器 0001: 开启数字噪声滤波器, 滤除脉宽小于1×I2C_KCLK的噪声。 : : 1110: 开启数字噪声滤波器, 滤除脉宽小于14×I2C_KCLK的噪声。 1111: 开启数字噪声滤波器, 滤除脉宽小于15×I2C_KCLK的噪声。 <i>注意: 这些位只能在I2C禁止状态 (PE为0) 时写入。</i>
7	ERRIE	错误中断检测使能 0: 禁止错误检测中断 1: 使能错误检测中断

6	TCIE	传输完成中断使能 0: 禁止传输完成中断 1: 使能传输完成中断
5	STOPIE	停止位检测中断使能 0: 禁止停止位检测中断 1: 使能停止位检测中断
4	NACKIE	NACK应答中断使能 0: 禁止NACK中断 1: 使能NACK中断
3	ADDRIE	地址匹配中断使能 0: 禁止地址匹配中断 1: 使能地址匹配中断
2	RXIE	接收中断使能 0: 禁止接收中断 1: 使能接收中断
1	TXIE	发送中断使能 0: 禁止发送中断 1: 使能发送中断
0	PE	I2C使能 0: 禁止I2C 1: 使能I2C <i>注意: 当PE清0时, 会执行软件复位: 将释放SCL和SDA, 内部状态机和状态位均恢复为复位值。PE清0必须保持至少3个APB时钟周期。</i>

28.6.2 控制寄存器 2 (I2C_CR2)

偏移地址: 0x04

复位值: 0x0000 0000

访问: 由于时钟同步, 写访问最长生效时间可达 $3 \times t_{PCLK1} + 4 \times t_{I2C_CLK}$ 。当对该寄存器执行写访问未完成后, 再次执行写访问, 会在第二个写访问中插入等待周期, 直到前一个写访问完成为止。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					Res.	AUTO END	RELOAD	NBYTES[7:0]							
					rw	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
NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	AUTOEND	<p>主模式自动结束模式 该位由软件置1和清0。</p> <p>0: 软件结束模式（当NBYTES数据传输完成时，TC标志将置1，驱动SCL低电平并延长等待，等待相应的软件操作） 1: 自动结束模式（当NBYTES数据传输完成时，将自动发送停止位）</p> <p><i>注意：在从模式下，该位无效；当RELOAD为1时，该位无效。</i></p>
24	RELOAD	<p>重载模式 该位由软件置1和清0。</p> <p>0: 传输完NBYTES字节数据后，NBYTES[7:0]不需要重新写入非0值。 1: 传输完NBYTES数据后，TCR标志将置1，驱动SCL驱动为低电平并延长等待，直到NBYTES[7:0]重新写入非0值。</p>
23:16	NBYTES[7:0]	<p>字节计数器 待发送或接收的字节数。 在从模式下，当SBC为0时，该位域无效。 <i>注意：START位置1时，不允许更改该位域。</i></p>
15	NACK	<p>从模式NACK生成 该位由软件置1，并可在发送NACK、接收到停止位、地址匹配、或PE清0时，该位自动清0。</p> <p>0: 在当前接收的字节后发送ACK 1: 在当前接收的字节后发送NACK</p> <p><i>注意：向该位写0无效。</i></p> <p><i>该位仅在从模式下使用；在主模式接收数据时，与NACK位的值无关，最后一个字节后都将自动生成NACK，其余字节自动生成ACK。</i></p> <p><i>当NOSTRETCH为1，从模式接收数据发生上溢错误时，将自动生成NACK，不受NACK位的控制。</i></p>
14	STOP	<p>主模式停止位生成 该位由软件置1，在检测到停止位时或PE清0时由硬件清0。</p> <p>0: 不生成停止位。 1: 在当前字节传输完成后生成停止位。</p> <p><i>注意：向该位写0无效。</i></p>
13	START	<p>主模式起始位生成 该位由软件置1，并可在发送起始位和地址之后、发送仲裁丢</p>

失、出现超时错误或PE清0时，该位自动清0。

I2C_ICR的ADDRCF位写入1，该位也会清0。

将该位置1会在总线释放后生成起始位。如果在主模式下 AUTOEND为0（软件结束模式）、RELOAD为0（非重载模式），将该位置1会在NBYTES[7:0]传输结束后，生成重复起始位。

0：不生成起始位。

1：生成起始位或重复起始位。

注意：向该位写0无效；当RELOAD为1时，该位无效。

处于从模式，也可将该位置1，以便当总线空闲后发送起始位并切换为主模式。

总线忙将该位置1会等待总线空闲后再发送起始位。

12 HEAD10R

主模式发送转接收10位地址头发送控制

主模式发送转接收时，只发送10位地址的前7位地址头字节。

0：主设备发送完整的10位地址读序列：起始位+带写命令10位地址高字节+10位地址低字节+重复起始位+带读命令的10位地址高字节，参见[地址模式](#)。

1：主设备读数据时只发送10位地址的前7位和读命令。

注意：START位置1时，不允许更改该位。

11 ADD10

主模式10位寻址模式

0：主设备工作在7位寻址模式。

1：主设备工作在10位寻址模式。

注意：START位置1时，不允许更改该位。

10 RD_WRN

主模式数据传输方向

0：主设备发送数据。

1：主设备接收数据。

注意：START位置1时，不允许更改该位。

9:0 SADD[9:0]

主模式待发送的从设备地址

7位地址模式（ADD10为0）：

SADD[7:1]：待发送的7位从设备地址。

10位地址模式（ADD10为1）：

SADD[9:0]：待发送的10位从设备地址。

注意：START位置1时，不允许更改该位。

28.6.3 地址寄存器 1 (I2C_OAR1)

偏移地址：0x08

复位值：0x0000 0000

访问：由于时钟同步，写访问最长生效时间可达 $3 \times t_{PCLK1} + 4 \times t_{I2C_KCLK}$ 。当对该

寄存器执行写访问未完成后，再次执行写访问，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.				OA1 MODE	OA1[9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OA1EN	从模式地址1使能 0: 接收到地址1不应答ACK。 1: 接收到地址1应答ACK。
14:11	保留	必须保持复位值。
10	OA1MODE	从模式地址1模式 0: OA1为7位地址模式。 1: OA1为10位地址模式。 <i>注意: 在OA1EN为0时才可以写入这些位。</i>
9:0	OA1[9:0]	从模式地址1 7位地址模式 (OA1MODE为0): OA1[7:1]: 7位从模式地址。 10位地址模式 (OA1MODE为1): OA1[9:0]: 10位从模式地址。 <i>注意: 在OA1EN为0时才可以写入该位域。</i>

28.6.4 地址寄存器 2 (I2C_OAR2)

偏移地址: 0x0C

复位值: 0x0000 0000

访问: 由于时钟同步，写访问最长生效时间可达 $3 \times t_{PCLK1} + 4 \times t_{I2C_KCLK}$ 。当对该寄存器执行写访问未完成后，再次执行写访问，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.				OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OA2EN	从模式地址2使能 0: 接收到地址2不应答ACK。 1: 接收到地址2应答ACK。
10:8	OA2MSK[2:0]	从模式地址2屏蔽位 000: OA2[7:1]全部比较。 001: OA2[1]被屏蔽, 比较OA2[7:2]。 010: OA2[2:1] 被屏蔽, 比较OA2[7:3]。 011: OA2[3:1] 被屏蔽, 比较OA2[7:4]。 100: OA2[4:1] 被屏蔽, 比较OA2[7:5]。 101: OA2[5:1] 被屏蔽, 比较OA2[7:6]。 110: OA2[6:1] 被屏蔽, 比较OA2[7]。 111: OA2[7:1]不进行比较, 对接收到的全部7位地址(保留的I2C地址除外)应答。 <i>注意: 在OA2EN为0时才可以写入这些位。 OA2MSK有屏蔽位(不为0b000), 即使从模式地址2比较匹配, 对保留的I2C地址(0b0000xxx和0b1111xxx)也不会应答ACK。</i>
7:1	OA2[7:1]	从模式地址2 7位从模式地址。 <i>注意: 在OA2EN为0时才可以写入这些位。</i>
0	保留	必须保持复位值。

28.6.5 时钟配置寄存器 (I2C_TIMINGR)

偏移地址: 0x10

复位值: 0x0030 3D5B

注意: 该寄存器只能PE为0时才可以写入。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.				SCLDEL[3:0]				SDADEL[3:0]			
r/w	r/w	r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
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

位/位域	名称	描述
31:28	PRESC[3:0]	时钟预分频 用于对I2C_KCLK进行预分频, 生成用于数据建立时间SCLDEL、数据保持时间SDASCL、SCL高电平时间SCLH、SCL

低电平时间SCLL的计时时钟，时钟周期如下：

$$t_{PRESC} = (PRESC + 1) \times t_{I2C_KCLK}$$

27:24	保留	必须保持复位值。
23:20	SCLDEL[3:0]	<p>数据建立时间</p> <p>在SDA的边沿和SCL上升沿之间生成延时t_{SCLDEL}。当NOSTRETCH=0时，延长SCL低电平t_{SCLDEL}时间。</p> $t_{SCLDEL} = (SCLDEL + 1) \times t_{PRESC}$ <p>请参见图I2C_TIMINGR寄存器配置生成SCL时序。 注意：t_{SCLDEL}用来生成$t_{SU:DAT}$。</p>
19:16	SDADEL[3:0]	<p>数据保持时间</p> <p>在SCL下降沿和SDA的边沿之间生成延时t_{SDADEL}。当NOSTRETCH=0时，延长SCL低电平t_{SDADEL}时间。</p> $t_{SDADEL} = SDADEL \times t_{PRESC}$ <p>请参见图I2C_TIMINGR寄存器配置生成SCL时序。 注意：t_{SDADEL}用来生成$t_{HD:DAT}$。</p>
15:8	SCLH[7:0]	<p>SCL高电平时间</p> <p>生成SCL的高电平时间t_{SCLH}。</p> $t_{SCLH} = (SCLH + 1) \times t_{PRESC}$ <p>请参见图I2C_TIMINGR寄存器配置生成SCL时序。 注意：t_{SCLH}用来生成$t_{SU:STO}$和$t_{HD:STA}$。</p>
7:0	SCLL[7:0]	<p>SCL低电平时间</p> <p>生成SCL的低电平时间t_{SCLL}。</p> $t_{SCLL} = (SCLL + 1) \times t_{PRESC}$ <p>请参见图I2C_TIMINGR寄存器配置生成SCL时序。 注意：t_{SCLL}用来生成t_{BUF}和$t_{SU:STA}$。</p>

28.6.6 中断和状态寄存器 (I2C_ISR)

偏移地址：0x18

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								ADDCODE[6:0]						DIR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	TIMEOUT	Res.	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE	
r		r	r		r	r	r	r	r	r	r	r	r	rs	rs

位/位域	名称	描述
------	----	----

31:24	保留	必须保持复位值。
23:17	ADDCODE[6:0]	从模式接收的地址 发生地址匹配事件时 (ADDR置1)，该位域更新为接收到的地址。在10位地址模式时，该位域更新为接收到10位地址高字节地址。
16	DIR	从模式数据传输方向 在发生地址匹配事件ADDR置1时更新。 0: 从模式接收数据。 1: 从模式发送数据。
15	BUSY	总线忙 该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置1。当检测到停止位或PE清0时，该位自动清0。
14:13	保留	必须保持复位值。
12	TIMEOUT	超时错误标志 发生总线超时错误或时钟累积超时错误时，该标志由硬件置1，将TIMEOUTCF位置1，该位自动清0。 <i>注意：当PE清0，该位自动清0。 该位保持复位值。</i>
11	保留	必须保持复位值。
10	OVR	从模式溢出错误标志 在从模式下且NOSTRETCH为1时，如果发生上溢/下溢错误，该标志由硬件置1，将OVRDCF位置1，该位自动清0。 <i>注意：当PE为0，该位自动清0。</i>
9	ARLO	仲裁丢失标志 发生仲裁丢失时，该标志由硬件置1，将ARLOCF位置1，该位自动清0。 <i>注意：当PE为0，该位自动清0。</i>
8	BERR	总线错误标志 数据传输中检测到总线错误时，该标志置1，将BERRDCF位置1，该位自动清0。在从模式地址接收阶段，该标志不会置1。 <i>注意：当PE为0，该位自动清0。</i>
7	TCR	传输完成，等待重新写入NBYTES[7:0] 当RELOAD为1且NBYTES[7:0]数据传输完成时，该标志由硬件置1，当NBYTES[7:0]写入非0值时，该位自动清0。 <i>注意：当PE为0，该位自动清0。 仅当SBC为1时，才能用于从模式。</i>

6	TC	<p>主模式传输完成</p> <p>当RELOAD为0、AUTOEND为0且NBYTES[7:0]数据传输完成时，该标志置1，当START或STOP位置1时，该标志由软件清0。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>
5	STOPF	<p>停止位标志</p> <p>主模式发出停止位后，在总线上检测到停止位，该标志置1；从模式地址匹配后，在总线上检测到停止位，该标志由硬件置1。将STOPCF位置1，该位自动清0。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>
4	NACKF	<p>接收到NACK标志</p> <p>当前字节发送完后接收到NACK，该标志由硬件置1，将NACKCF位置1，该位自动清0。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>
3	ADDR	<p>从模式地址匹配标志</p> <p>从模式接收的地址与使能的从设备地址寄存器匹配时，该标志由硬件置1，将ADDRCF位置1，该位自动清0。</p> <p><i>注意：当PE为0，该位自动清0。</i></p>
2	RXNE	<p>接收数据寄存器非空</p> <p>I2C_RXDR寄存器非空，该标志由硬件置1；读取I2C_RXDR数据后，该位自动清0。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>
1	TXIS	<p>发送中断状态</p> <p>当I2C_TXDR为空时，该标志置1；将下一个要发送的数据写入I2C_TXDR后，该位自动清0。</p> <p>当NOSTRETCH为1时，可由软件写1，用于生成TXIS事件触发中断或DMA请求。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>
0	TXE	<p>发送数据寄存器为空</p> <p>当I2C_TXDR寄存器为空时，该标志由硬件置1，将下一个要发送的数据写入I2C_TXDR后，该位自动清0。</p> <p>该位可软件写1，用于使能待发送数据写入到I2C_TXDR寄存器。</p> <p><i>注意：当PE清0，该位自动清0。</i></p>

28.6.7 中断清除寄存器 (I2C_ICR)

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OVRCF		ARLOCF	BERRCF	Res.			STOPCF	NACKCF	ADDRCF	Res.	
		w		w	w	w	w			w	w	w			

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	OVRCF	溢出标志清除 该位写入1，I2C_ISR寄存器中的OVR标志清0。
9	ARLOCF	仲裁丢失标志清除 该位写入1，I2C_ISR寄存器中的ARLO标志清0。
8	BERRCF	总线错误标志清除 该位写入1，I2C_ISR寄存器中的BERR标志清0。
7:6	保留	必须保持复位值。
5	STOPCF	停止位标志清除 该位写入1，I2C_ISR寄存器中的STOPF标志清0。
4	NACKCF	NACK位标志清除 该位写入1，I2C_ISR寄存器中的NACKF标志清0。
3	ADDRCF	从模式地址匹配标志清除 该位写入1，I2C_ISR寄存器中的ADDR标志清0，同时I2C_CR2寄存器中的START位也清0。
2:0	保留	必须保持复位值。

28.6.8 接收数据寄存器 (I2C_RXDR)

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								RXDATA[7:0]							
								r	r	r	r	r	r	r	r

位/位域	名称	描述
31:8	保留	必须保持复位值。

7:0 RXDATA[7:0] 接收数据寄存器

28.6.9 发送数据寄存器 (I2C_TXDR)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.								TXDATA[7:0]								
								rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TXDATA[7:0]	发送数据寄存器 注意: 仅可在I2C_ISR寄存器的TXE为1时, 该位域才可以写入数据。

29 通用异步收发器 (USART)

29.1 简介

通用同步/异步收发器 USART 支持全双工或半双工，同步或异步的通信方式，符合通用工业标准 NRZ 异步串行编码格式，内部集成小数级波特率发生器，波特率范围可配置。

USART 还支持单线半双工通信，IrDA 的 SIR 编码标准通信，多机通信和硬件流控(CTS/RTS)。

USART 支持使用 DMA 通信。

29.2 主要特性

- 持全双工半双工，同步或异步的通信方式
- 符合 NRZ 标准格式
- 支持 16 倍或 8 倍过采样
- 内置波特率发生器
- 可配置双时钟域，可工作在独立于 PCLK 的专用时钟源
- 发送器/接收器配有独立的使能位
- TX/RX 引脚可交换
- 发送/接收的信号极性可独立控制
- 内置两个深度为 8 的接收 FIFO 和发送 FIFO
- 自动波特率检测功能
- 支持数据长度：7、8 或 9 位
- 可编程数据传输顺序，支持 MSB 或 LSB 优先
- 可编程停止位位数:0.5 位，1 位，1.5 位或 2 位
- 奇偶校验可配置为：奇校验、偶校验、无校验
- 支持通过 DMA 进行连续通信
- 支持主/从模式同步通信
- 支持单线半双工通信
- 支持硬件流控 (CTS/RTS) 以及 RS485 收发控制
- 支持多机通信，支持通过空闲检测或地址匹配检测从静默模式唤醒

29.3 USART 扩展功能

- 支持 3/16bit 脉宽的 IrDA SIR 编解码器
- 支持 Modbus 通信，提供超时检测功能及 CR/LF 字符识别功能

29.4 USART 功能列表

表 29-1 USART 功能列表

USART 功能	USART1	USART2\3
硬件流控	√	√
DMA 通信	√	√
多机通信	√	√
同步模式（主/从）	√	√
智能卡模式	×	×
单线半双工通信	√	√
IrDA SIR ENDEC 模块	√	×
双时钟域，从低功耗模式唤醒	√	×
接收器超时中断	√	×
Modbus 通信	√	×
自动波特率检测	√	×
驱动器使能	√	√
USART 数据长度	7 位、8 位和 9 位	
Tx/Rx FIFO	√	×
Tx/Rx FIFO 大小	8	×
预分频	√	×

注意：“√”代表支持此功能，“×”代表不支持此功能。

29.5 USART 功能概述

29.5.1 USART 框图

图 29-1 USART 框图

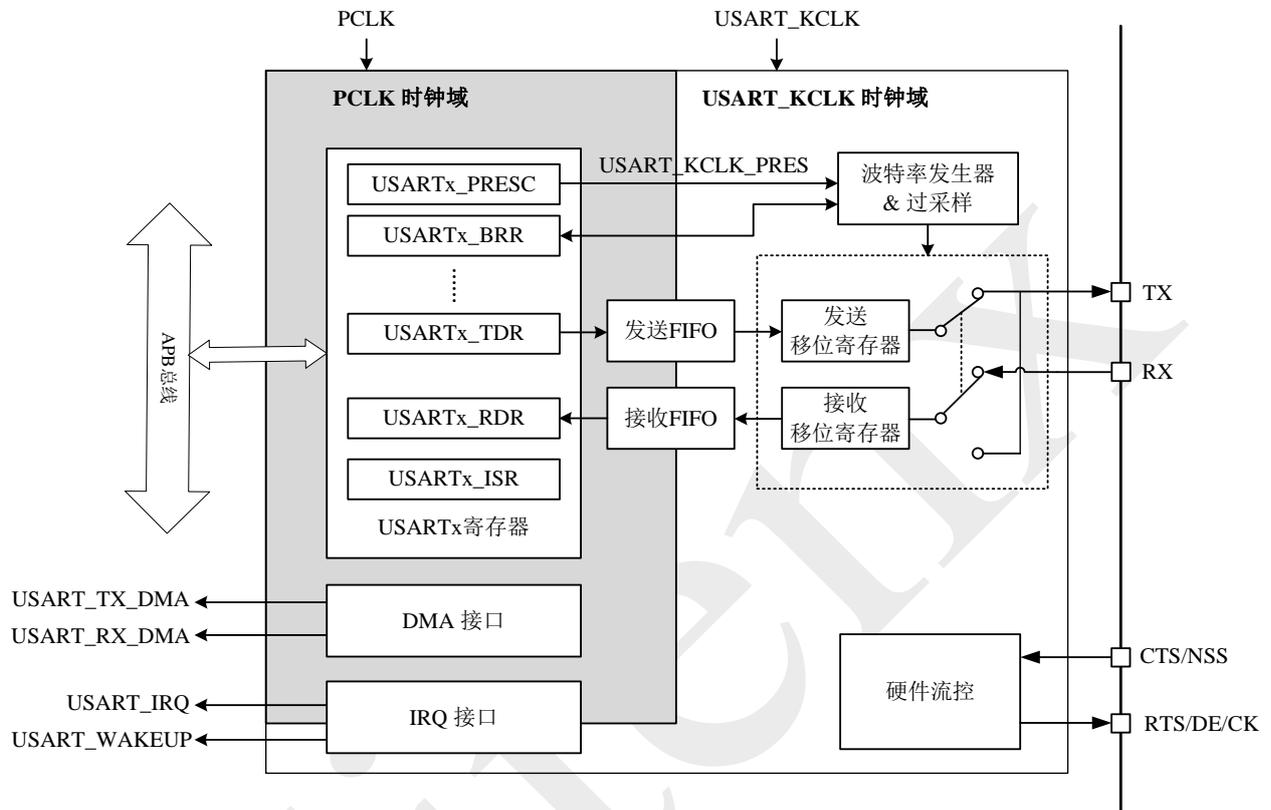


表 29-2 USART1 框图信号说明表

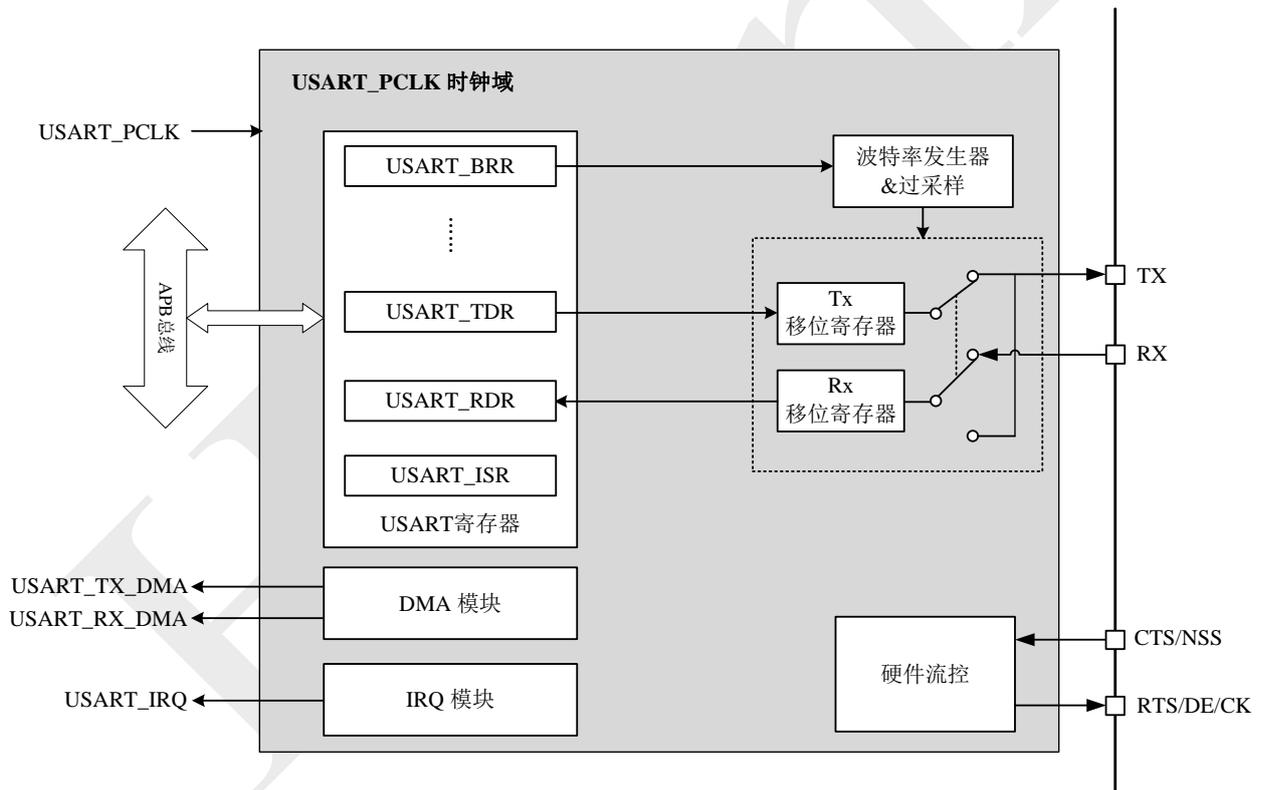
信号名称	类型	说明
USART_PCLK	输入	USART_PCLK 时钟域的时钟源，同 APB 总线时钟
USART_KCLK	输入	USART_KCLK 时钟域的时钟源。 可配置为：PCLK、LSE、HSI 或 SYSCLK
USART_KCLK_PRES	输入	通信时钟源，由 USART_KCLK 经过预分频后得到
USART_TX_DMA	输出	USART 发送 DMA 请求信号
USART_RX_DMA	输出	USART 接收 DMA 请求信号
USART_IRQ	输出	USART 中断请求信号
USART_WAKEUP	输出	USART 唤醒请求信号

如上图所示，USART1 模块工作在两个时钟域：

- USART_PCLK 时钟域的时钟源来自 APB 总线时钟，修改寄存器时需提供 USART_PCLK。
- USART_KCLK 时钟源由 RCC 提供，可配置为 PCLK、LSE、HSI 或 SYSCLK。当时钟源配置为 HSI 或 LSE 时支持从 Stop 模式唤醒。
- 由于 USART 寄存器属于 USART_PCLK 时钟域，所以在 USART_KCLK 时钟停止时，仍然可以读写寄存器。
- USART_KCLK 时钟配置为 PCLK 时，USART_KCLK 与 USART_PCLK 相同。

USART 工作在同步从模式时，使用外部主设备通过 CK 引脚输入的 SCLK 时钟信号来处理数据流，此时约束 USART_KCLK 需至少为 SCLK 时钟频率的 3 倍。

图 29-2 USART2\3 框图



如上图所示，USART2\3 模块不支持双时钟域，仅工作在 USART_PCLK 时钟域，此时 USART_KCLK 与 USART_PCLK 相同。信号说明请参见下表：

表 29-3 USART2\3 框图信号说明表

信号名称	类型	说明
USART_PCLK	输入	USART_PCLK 时钟域的时钟源，同 APB 总线时钟

信号名称	类型	说明
USART_TX_DMA	输出	USART 发送 DMA 请求信号
USART_RX_DMA	输出	USART 接收 DMA 请求信号
USART_IRQ	输出	由 USART 中断信号

USART1~2 工作在同步从模式时，使用外部主设备通过 CK 引脚输入的 SCLK 时钟信号来处理数据流，此时约束 USART_PCLK 需至少为 SCLK 时钟频率的 3 倍。

USART 引脚说明

USART 双向通信：

USART 双向通信时至少需要两个通信引脚：数据接收(RX)，和数据发送(TX)：

- RX 引脚，数据接收：
是串行数据的接收引脚，提供过采样功能有效区分数据与噪声。
- TX 引脚，数据发送：
是数据发送引脚。禁止发送时 TX 引脚恢复到可编程 I/O 状态。如果发送使能但是没有发送数据时 TX 引脚为高电平 “H”。在单线通信模式时，此引脚用于数据收发。

RS232 硬件流程控制：

RS232 流控中需使用以下引脚：

- CTS (Clear To Send)：
此引脚为输入，接收到高电平时，当前传输完成后数据发送被阻止
- RTS (Request To Send)：
此引脚为输出，USART 准备好接收数据时将此引脚拉低

RS485 硬件流程控制：

RS485 流控中需使用以下引脚：

- DE (Drive Enable)：
此信号用于开启外部收发器的发送模式。

同步主/从模式

同步主/从通信模式下用到以下引脚：

- CK
此引脚在同步主模式下作为时钟输出，在同步从模式下作为时钟输入

- 在同步主模式下，此引脚用于输出同步模式下使用的数据传输时钟，以 SPI 主模式的形式进行同步发送，起始位和停止位无对应时钟脉冲，可选择是否在末尾数据位发送时钟脉冲，时钟相位和极性均可编程。详见 [同步模式](#)。同时 RX 引脚上可同步接收数据
- 同步从模式下作为时钟输入
- NSS

软件使能时，在同步从模式下此引脚作为从机片选，同步主模式下不使用该引脚。

注意： DE、RTS 和 CK 复用同一引脚。

NSS 和 CTS 复用同一引脚。

29.5.2 USART 帧格式

可通过 USART_CR1 寄存器中的 M 位域（M0: bit12, M1: bit28）编程字符长度，参见：[字符组成示意图](#)。

- 7 位字符长度：M[1:0] = “10”；
- 8 位字符长度：M[1:0] = “00”；
- 9 位字符长度：M[1:0] = “01”；

注意： 自动波特率检测只支持 8 位，9 位字符长度。

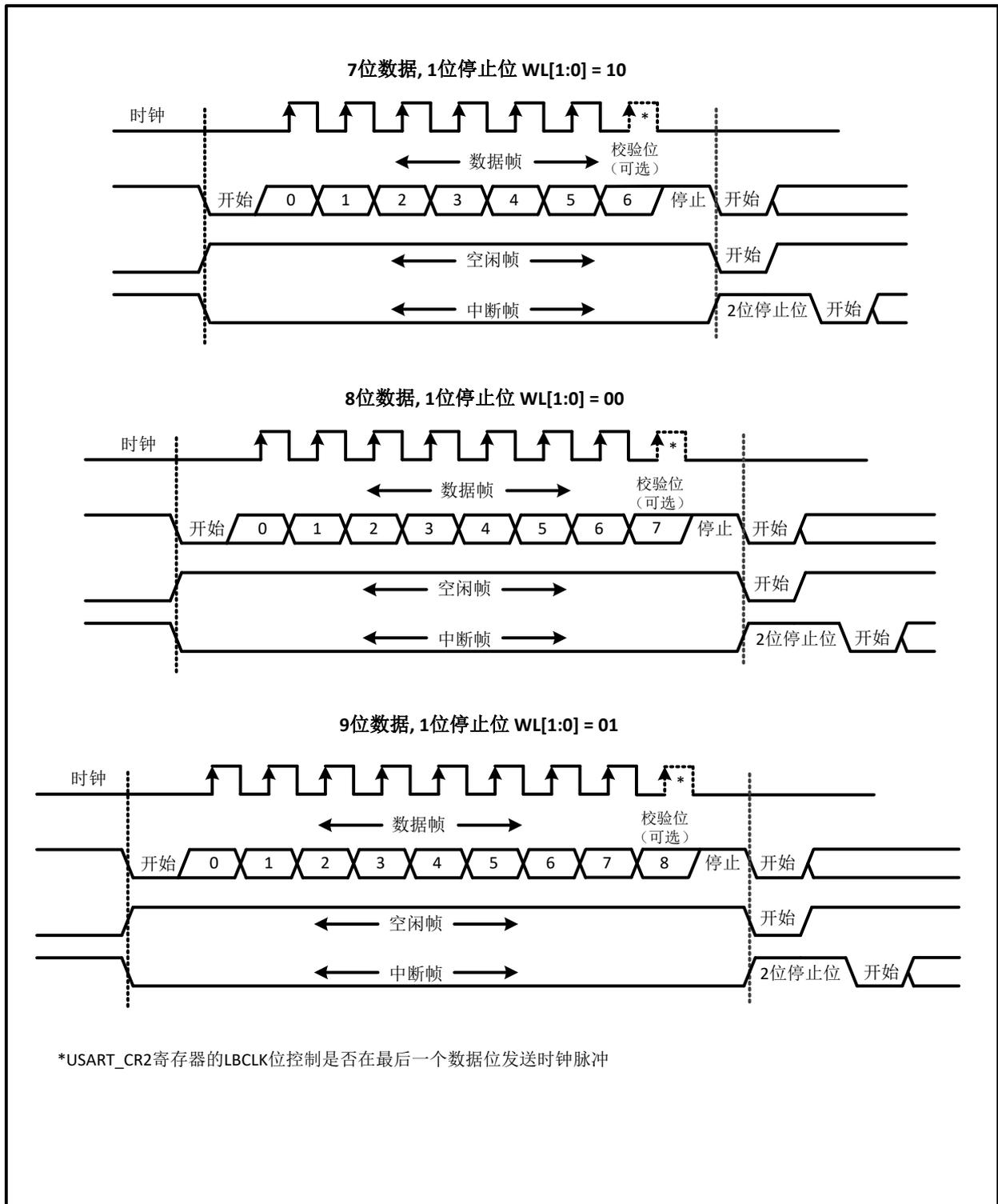
空闲帧 (Idle Character) 整帧全为 “1”。

中断帧 (Break Character) 整帧全为 “0”。中断帧的末尾由发送器插入 2 位停止位。

发送和接收操作由通用波特率发生器驱动。

当发送器或接收器的使能位（TE/RE）置 1 时，产生发送时钟或接收时钟。

默认情况下，TX 和 RX 引脚在起始位时都是低电平，停止位期间都是高电平。发送和接收信号的极性可独立配置是否翻转，参见 [寄存器 USART_CR2](#) 中 TXINV,RXINV 位的说明。详见下图：

图 29-3 字符组成示意图


29.5.3 USART 校验控制

表 29-4 USART 校验控制

字符长度/M[1:0]		PEN 位 校验控制 (置 1 使能)	USART 帧格式 ^(注)
8 位	00	0	起始位 8 位数据 停止位
		1	起始位 7 位数据 校验位 停止位
9 位	01	0	起始位 9 位数据 停止位
		1	起始位 8 位数据 校验位 停止位
7 位	10	0	起始位 7 位数据 停止位
		1	起始位 6 位数据 校验位 停止位

注意: 在数据寄存器中, 校验位位于 MSB

偶校验:

使能偶校验时 (USART_CR1 寄存器中 PS 位 = 0), 一个字符中 (含校验位) “1” 的总数为偶数,

例如, 数据 “0101 1010” 中已包含偶数个 “1”, 则使能偶校验时, 校验位为 “0”。

奇校验

使能奇校验时 (USART_CR1 寄存器中 PS 位 = 1), 一个字符中 (含校验位) “1” 的数量为奇数。

例如, 数据 “0101 1010” 中包含偶数个 “1”, 则使能奇校验时, 校验位为 “1”。

29.5.4 USART FIFO 和阈值设置

USART 配有深度为 8 的一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。将 USART_CR1 寄存器中的 FIFOEN 位置 1 即可使能 FIFO 模式。FIFO 模式仅支持在 USART1 模块使用。

由于最大数据字长为 9 位, 所以 TXFIFO 的宽度为 9 位。但是 RXFIFO 的默认宽度为 12 位, 这是因为接收器在 FIFO 中不仅存储数据还存储与每个字符相关的错误标志 (奇偶校验错误、噪声错误和帧错误标志)。

注意: 接收的数据和相关错误标志一起存储在 RXFIFO 中, 但读取 RDR 时仅读取到数据。状态标志位于 USART_ISR 寄存器中。

可以由 TXFIFO 或 RXFIFO 设置的阈值触发中断。阈值的设置在 USART_CR3 寄存器中的 RXFTCFG 和 TXFTCFG 位域编程。

在使能阈值时:

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时, 触

发中断。此时，USART_ISR 寄存器中的阈值标志 RXFT 置 1，表示已接收到 RXFTCFG 个数据，此时接收数据寄存器 USART_RDR 中存有 1 个数据，RXFIFO 中存有(RXFTCFG - 1) 个数据。

例如，通过 RXFTCFG 位域编程为“010”将阈值设定为 RXFIFO 总深度的 1/2。则接收到 RXFIFO 总深度一半的数据时认为到达阈值，RXFT 标志置 1。此时 RXFIFO 中存有 $(8 \times 0.5 - 1 = 3)$ 3 个数据，USART_RDR 中存有 1 个数据。

- 当 TXFIFO 中的未使用空间数达到在 TXFTCFG 位域中编程的阈值时，会触发中断，USART_ISR 寄存器中的 TXFT 标志位置 1。

29.5.5 USART 发送器

根据 M[1:0]的配置，发送的字符长度可以是 7 位、8 位或 9 位。发送使能位 (TE) 置 1 即可使能发送器。发送移位寄存器的数据由 TX 引脚输出，相应的时钟脉冲由 CK 引脚输出。

字符发送：

通过 USART_CR1 寄存器的 TE 位置 1 使能发送器。

USART 发送期间，TX 引脚首先移出数据的最低有效位（默认 LSB 优先）。接收数据寄存器 USART_TDR 为内部总线和发送移位寄存器之间提供缓冲区 (TDR)。

使能 FIFO 时，写入发送数据寄存器 (USART_TDR) 中的数据会预存放在 TXFIFO 中。每个字符的发送都以起始位开始，以停止位结束，停止位的位数在 USART_CR2 寄存器的 STOP[1:0]位域可配置为 0.5、1、1.5 或 2 位。

注意： 向 USART_TDR 中写入发送的数据前，发送使能位 TE 需要先置 1。

数据发送期间清零 TE 位会冻结波特率计数器，影响 TX 引脚上的数据，当前发送的数据可能丢失，所以不建议在发送过程中复位 TE 位。

TE 位置 1 后会先发送一个空闲帧。

停止位：

可以在 USART_CR2 的 STOP[1:0]中编程停止位的位数，停止位跟随在每一个字符后发送。停止位可配置为以下选项：

- 0.5 个停止位
- 1 个停止位：默认配置。
- 1.5 个停止位
- 2 个停止位：常用于标准 USART 通信和单线半双工。

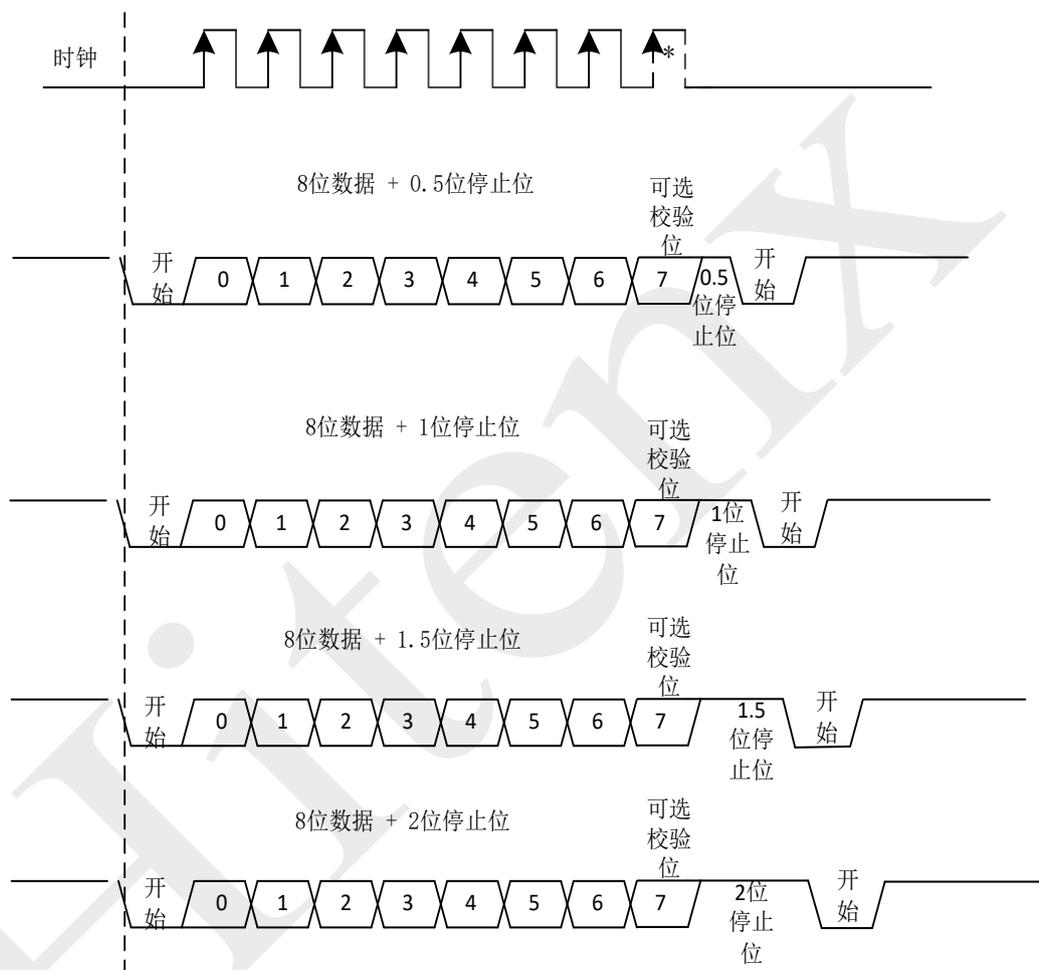
空闲帧中也包含停止位。

中断帧可能包含 10 个低电平位（M[1:0] = “00” 时）、11 个低电平位（M[1:0] = “01”）或 9 个低电平位（M[1:0] = “10” 时），结尾接 2 位停止位。

数据末尾位可选为奇偶校验位。奇偶校验位与停止位不同，不在末尾追加而是占用一位数据位。

下图为三种不同停止位配置示例：

图 29-4 可编程停止位示例



*注：可通过USART_CR2 寄存器中LBCL位配置末位是否有对应时钟沿

字符发送步骤：

字符发送配置步骤：

- 1) 配置 USART_BRR 寄存器选择通信波特率。
- 2) 配置 USART_CR1 中的 M[1:0]定义字符长度长度。
- 3) 配置 USART_CR2 中的 STOP[1:0]设置停止位数。
- 4) 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
- 5) 使用 DMA 通信时请选择 USART_CR3 中的 DMA 使能位 (DMAT)。请参照 [USART 多机通信](#)中的说明配置 DMA 寄存器。
- 6) 将 USART_CR1 中的 TE 位置 1，使能发送器（首次发送时先发一个空

闲帧)。

- 7) 在 USART_TDR 寄存器中写入准备发送的数据。
 - 禁用 FIFO 时, 向 USART_TDR 写入数据会将 TXE 标志清零, 表示 USART_TDR 发送寄存器中数据已满。
 - 使能 FIFO 模式时, 向 USART_TDR 写入数据会增添在 TXFIFO 中。USART_ISR 标志寄存器的 TXFNF 标志位为“1”期间代表 TXFIFO 未
满, 此时可以写入 USART_TDR。如此标志清零, 表示 TXFIFO 已满, 不能继续写入 USART_TDR。
- 8) 当最后需发送的数据写入 USART_TDR 寄存器后, 需等待发送完成直到 USART_ISR 的发送完成标志位 TC 置 1。TC 置位时代表以下内容:
 - 禁用 FIFO 时表示最后一个字符已发送完成。
 - 使能 FIFO 时表示 TXFIFO 和移位寄存器都已清空。为保证最后一个字符内容完成发送, 需检查 TC 标志位置 1 后, 再禁用 USART 或进入低功耗模式。

单字节通信:

- 禁用 FIFO 时:

由于 USART_TDR 寄存器只有 1 字节空间, 所以写入 1 字节数据后 USART_TDR 寄存器已满, 发送数据非空标志 TXE 位清零。

发送数据寄存器 USART_TDR 为空时, TXE 标志位自动置 1, 此时表示:

- 数据已由 USART_TDR 寄存器移入移位寄存器, 开始发送数据;
- 可以在 USART_TDR 寄存器中写入数据, 前一数据不会被覆盖。

TXEIE 位置 1 使能中断时, TXE 标志置 1 会触发中断。

发送时, 写入的待发数据存储于 USART_TDR 寄存器中作为缓冲。新数据在当前发送结束时复制到移位寄存器中。

在发送未进行时, 写入 USART_TDR 寄存器的数据实际存放在移位寄存器中, 数据发送开始后, 发送数据非空标志 TXE 位立即置 1, 指示可以写入下一数据。

- 使能 FIFO 时, USART_ISR 寄存器中的发送 FIFO 未
满标志位 TXFNF 置位, 指示以下状态:
 - USART_TDR 寄存器为空;
 - 可以向 USART_TDR 寄存器中写入数据, 前一数据不会被覆盖。在发送进行时写 USART_TDR 寄存器会将数据存储于 TXFIFO 中。当前发送结束时下一数据从 TXFIFO 中复制到移位寄存器中。

TXFIFO 未滿时，写 USART_TDR 寄存器不会将 TXFNF 标志清零，此标志位保持置 1，直到 TXFIFO 满时清零。USART_CR1 中 TXFNFIE 位置 1 使能中断时，TXFNF 置 1 会触发中断。

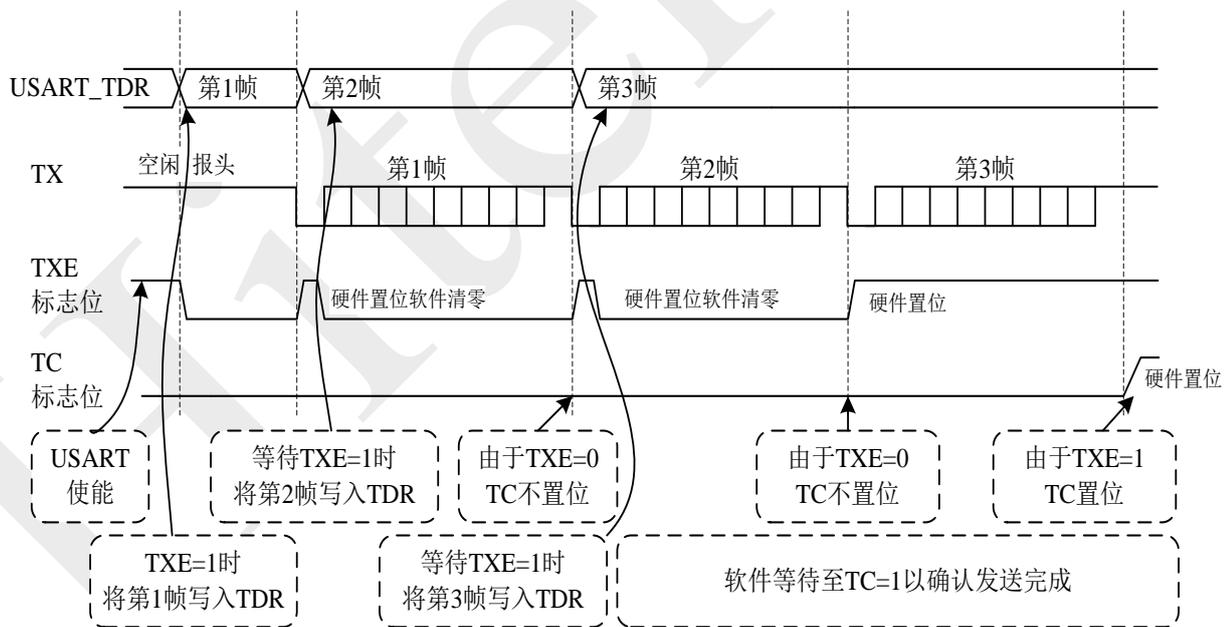
使用 FIFO 阈值时，配置当 TXFIFO 中空位数达到阈值时触发中断，再将数据写入 TXFIFO。这样的配置可以对 USART_TDR 连续写入多次，写入次数取决于阈值的大小。

随着停止位发送完毕，一帧数据发送完成，如果此时发送数据非空标志 TXE（使能 FIFO 时对应 TXFE）置 1，指示后续没有等待发送的数据，发送过程完成，随即发送完成标志位 TC 置位。如果 USART_CR1 寄存器中 TCIE 位置 1 使能中断，则 TC 标志置位同时将触发中断。

发送时的 TC 和 TXE 的动作请参见下图：

注意： 为确保最后一个字符完整发送，向 USART_TDR 寄存器中写入最后一个数据后，需要等待发送完成标志 TC 置 1 后，才可以禁用 USART 或进入低功耗模式。

图 29-5 发送过程 TC 和 TXE 的动作（FIFO 禁用）



注意： 当 FIFO 使能时数据按照标志位 TXFNF 传输（图中 TXE）。

中断帧：

将请求寄存器 USART_RQR 的 SBKRQ 位置 1，则在当前发送结束后发送一个中断帧。中断帧的长度取决于 M[1:0] 位域的设置，参见：[字符组成示意图](#)。

SBKRQ 位置位后，当前字符发送完成后将在 TX 线路上发送一个中断帧。此位由软件编程置位，在中断帧发送完成后的停止位期间自动清零。

USART 在中断帧末尾加入两个停止位以确保下一帧的起始位可以正确识别。

使能 FIFO 时即使 TXFIFO 已满，发送中断帧的优先级也高于数据的发送，仍会先发中断帧再继续数据的发送。所以如果需要完成数据发送后再发送中断帧，则需要等待发送数据寄存器空标志 TXE(TXFNF)置 1 后，再置位 SBKRQ。

空闲帧

将 TE 位置 1 则 USART 在发送第一个字符帧之前先发一个空闲帧。

29.5.6 USART 接收器

USART 可接收 7 位、8 位或 9 位长度的字符，在 USART_CR1 寄存器中 M[1:0] 位域可配置。

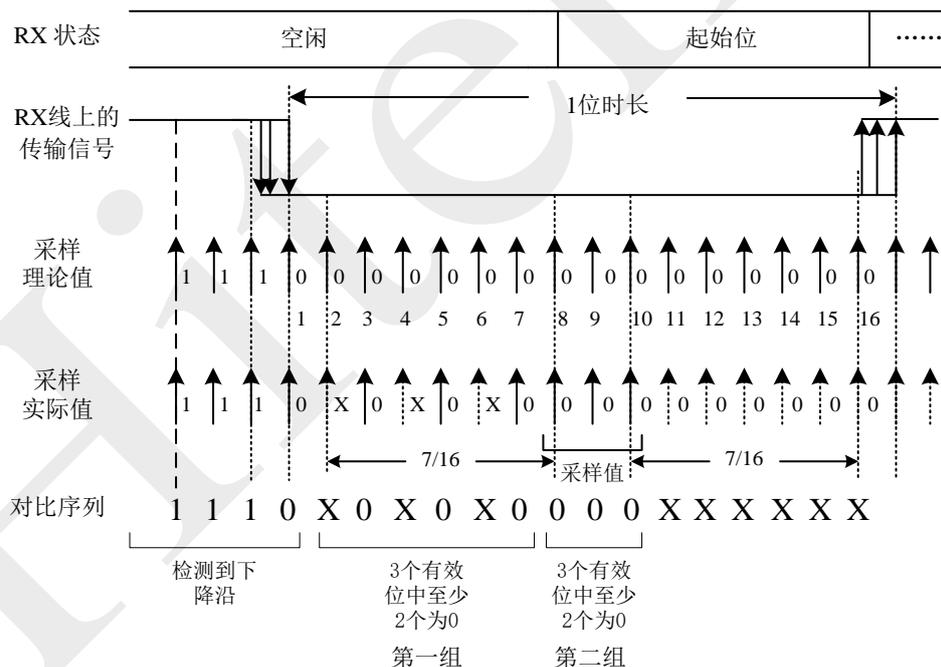
起始位检测：

在 USART 中识别出特定序列的采样结果时识别为起始位。

此序列为：111 0 X 0 X 0X 0X 0 X0X 0。

8 倍和 16 倍过采样的起始位检测方式相同。

图 29-6 8 倍/16 倍过采样时的起始位检测



注意： 如果采样序列不完整，则终止起始位检测，RX 回到空闲状态直到检测到下一个下降沿。

如果第一组 3 个采样位均为“0”，即第 3、5 和第 7 位首次采样时检测到的全为“0”且第二组 3 个采样位，第 8、9 和第 10 位仍检测到全为“0”，可确认收到起始位，此时接收非空标志 RXNE 位置 1，如果 RXNEIE = 1 则触发中断。在使能 FIFO 时，为 RXFNE 标志置 1，RXFNEIE = 1 时产生中断。

满足以下条件时，检测到起始位的同时，噪声标志位 NOISE 置 1：

- 两组采样的 3 个采样位中只有两位为 0 时。例如，第 3、5、7 位采样结果为“010”，第 8、9、10 位采样结果为“001”；
- 两组采样中任意一组的 3 个采样位中有两位为 0 时。例如，对第 3、5、7 位进行采样为“000”，但对第 8、9、10 位采样结果为“001”。

以上条件均不满足，则起始位检测终止，接收器返回空闲状态，无标志位置位。

字符接收

接收配置如下：

- 1) 配置 USART_BRR 寄存器选择通信波特率。
- 2) 配置 USART_CR1 中的 M[1:0] 定义字符长度。
- 3) 配置 USART_CR2 中的 STOP[1:0] 位域设置停止位数。
- 4) 向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
- 5) 使用 DMA 通信时，置位 USART_CR3 中的 DMA 使能位 (DMAR)。参照 [USART 多机通信](#) 章节配置 DMA 寄存器。
- 6) USART_CR1 寄存器 RE 位置 1，使能接收。
- 7) 等待 REACK 标志置位以确认接收器准备完毕。

接收字符时：

- 不使用 DMA 通信：
 - 禁用 FIFO 时，接收非空标志 RXNE 置 1 时，表示接收移位寄存器中的数据已经传送到 USART_RDR 寄存器中，指示已接收到数据，可以读取。

由于 USART_RDR 寄存器中仅有 1 字符空间，读 USART_RDR 寄存器时 RXNE 标志位清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 位写 1，使用清空 RXFIFO 的方式将 RXNE 标志清零。

- 使能 FIFO 时，接收 FIFO 非空标志 RXFNE 位置 1 时，表示 RXFIFO 中已接收到数据，读 USART_RDR 寄存器时获取接收数据。接收到的数据和相应错误标志位一起存储在 RXFIFO 中。当 RXFNEIE 位置 1 时，触发中断。

RXFIFO 为空时，RXFNE 标志位清零，也可以通过向 USART_RQR 中的 RXFRQ 位写 1，用清空 RXFIFO 的方式将 RXFNE 标志清零，相当于采用丢弃 RXFIFO 中剩余数据的来避免出现上溢错误。

还可以配置为达到 RXFIFO 阈值时触发中断，这样 MCU 可以根据 FIFO 阈值的设置批量读取数据。

- 使用 DMA 通信：
 - 禁用 FIFO 时，由于 USART_RDR 寄存器中仅有 1 字节空间，每收到

1 个字符, 接收非空标志位 RXNE 都会置 1, DMA 读取 USART_RDR 后 RXNE 位清零。

- FIFO 使能时, RXFIFO 非空时 RXFNE 位置 1, 每次 DMA 读请求都从 RXFIFO 中读取一个数据。DMA 读请求可由 RXFIFO 非空事件触发, 这样即可在 FIFO 收到数据时触发 DMA 读请求。

- 接收期间如果检测到帧错误、噪声错误或上溢错误, 相应错误标志位置位。

中断帧:

接收到中断帧时 USART 按帧错误处理。

空闲帧:

检测到空闲帧时处理方式和接收数据基本一致, 区别在于空闲帧出现时, IDLE 标志位置 1, 如果 USART_CR1 中的 IDLEIE 位置 1 会触发中断。

上溢错误:

- 禁用 FIFO 时:

每接收到一个字符后, 接收非空标志位 RXNE 置 1。当 RXNE 标志为 1 时, 或上一个 DMA 读请求未处理时, 又接收到新的字符, 就会产生上溢错误。RXNE 位清零前, 数据无法从移位寄存器传到接收数据寄存器 USART_RDR 中。

发生上溢错误时有以下动作:

- 上溢错误标志 ORE 位置 1;
- USART_RDR 寄存器中的内容不会丢失。读取 USART_RDR 寄存器时, 得到溢出前接收到的最后一个数据。
- 移位寄存器被新数据覆盖, 在上溢期间接收到的数据都将丢失。
- USART_CR1 中的 EIE 位置 1 时, 会触发中断。

- 使能 FIFO 时:

移位寄存器已准备好传输但 RXFIFO 已满时, 会发生上溢错误。

在 RXFIFO 中出现一个空闲位置之前, 数据无法从移位寄存器传送到 USART_RDR 寄存器。当出现空闲位置时, RXFIFO 非空, 接收 FIFO 非空标志位 RXFNE 置 1。

发生上溢错误时:

- 上溢错误标志 ORE 位置 1。
- RXFIFO 中接收到的第一个数据不会丢失。读取 USART_RDR 寄存器得到此数据。

- 移位寄存器将被覆盖。在上溢期间接收到的任何数据都会丢失。
- 如果 EIE 位置 1 时，会触发中断。

通过向 USART_ICR 寄存器中的 ORECF 位写 1 来清零 ORE 位。

注意： ORE 位置 1 时表示至少已有 1 个数据丢失。

ORE 位置 1 且禁用 FIFO 时，有以下两种可能：

- $RXNE = 1$ ，表示最后一个有效数据已存储在 USART_RDR 可以读取。
- $RXNE = 0$ ，RDR 寄存器中已没有需要读取的数据。在读取 RDR 寄存器中最后一个有效数据的同时接收到新数据，也会产生上溢，此新数据丢失。

时钟源和过采样模式的选择：

时钟源可通过时钟控制系统进行选择，详见时钟和复位控制（RCC）章节。需在使能 USART（USART_CR1 寄存器的 UE 位置位）之前选择时钟源。

依据以下两个条件选择时钟源：

- 可在低功耗模式下使用 USART；
- 通信速率；

USART1 支持双时钟域。可在 RCC_CCPIR 寄存器中将 USART_KCLK 的时钟源选择为 HSI、LSE、SYSCLK 或 PCLK。时钟源选择为 HSI 或 LSE 时，USART1 可在 Stop 模式下接收数据，并支持将 MCU 从 Stop 模式唤醒。

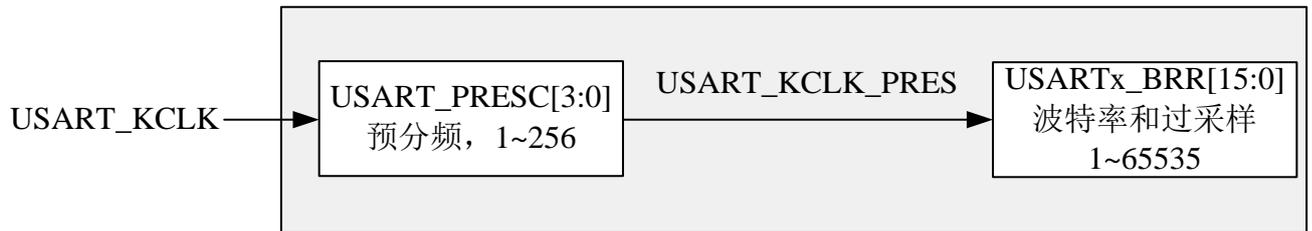
MCU 唤醒后，已接收的数据可以通过读 USART_RDR 寄存器，或通过 DMA 传输。如果选择了 SYSCLK 和 PCLK，这些时钟源在 Stop 模式下不工作，则 USART 也不能通信。

USART2~3 的时钟源为 PCLK，不支持双时钟域，也不能将 MCU 从低功耗模式下唤醒。

时钟源直接影响通信速率，限制了最大通信速率。

USART_KCLK 时钟通过 USART_PRESC 寄存器中定义的系数进行预分频，得到 USART_KCLK_PRESC。

注意： 仅 USART1 配有 USART_PRESC 寄存器。USART2~3 中没有此 USART_PRESC 寄存器，仅支持使用 PCLK 时钟源。

图 29-7 USART_KCLK 时钟分频框图(USART1)


接收器配备了 16 倍或 8 倍过采样功能（同步模式除外），可在噪声环境中提取有效数据，在最大通信速率与时钟的抗噪声之间获得最优平衡。

采样时钟可以是波特率时钟的 16 倍或 8 倍，过采样方式在 USART_CR1 寄存器中的 OVER8 位可选，有以下两种选项，参见图：[数据采样-8 倍过采样](#)和图：[数据采样-16 倍过采样及连接](#)。

- 选择 8 倍过采样 (OVER8 = 1)，对每个 bit 采样 8 次，比 16 倍过采样支持更高的通信速率（最高 USART_KCLK_PRES/8）。此情况下接收容差较低，适用于噪声小的系统中。
- 选择 16 倍过采样 (OVER8=0)，对每个 bit 采样 16 次，增加了接收容差。此情况下最大通信速率被限制在 USART_KCLK_PRES/16。

可通过 USARTx_CR3 寄存器中的 ONEBIT 位选择采样逻辑电平的方式，有以下两种选项：

- 在接收位的中心进行 3 次采样，使用多数表决的方式确定逻辑值。此情况下，如果参与多数表决的 3 次采样结果不同，则 NOISE 标志位置 1。
- 在接收位的中心进行 1 次采样，无 NOISE 标志输出。

注意： IrDA 模式不支持使用 8 倍过采样。使能这些模式时 OVER8 位由硬件强制清零。

根据应用场景可选：

- 噪声较大的环境下建议选择三次采样多数表决法 (ONEBIT=0)；如果检测到了噪声，则表示采样过程中有噪声干扰，为保证数据准确，推荐丢弃检测到噪声的数据（请参见表：[采样数据和噪声检测](#)）。
- 在线路无噪声影响时可选择单次采样法 (ONEBIT=1)，可以提高接收容差(参见[接收器对时钟偏差的容差](#)章节)。

接收帧检测到噪声时：

- 在接收非空标志 RXNE 位（使能 FIFO 时为 RXFNE 位）置位的上升沿时 NOISE 位置 1。
- 无效数据从移位寄存器传送到 USARTx_RDR 寄存器。

- NOISE 位置 1 时，如果 USART_CR3 寄存器中的 EIE 位置 1 则触发中断。NOISE 位置 1 时，RXNE 位（使能 FIFO 模式时为 RXFNE 位）同时置 1，如果 RXNEIE=1 则触发中断

使用 USART_ICR 寄存器中的 NOISECF 位写“1”清零 NOISE 位。

注意： 同步模式下不支持噪声错误检测。

IrDA 模式不支持使用 8 倍过采样。使能这些模式时 OVER8 位由硬件强制清零。

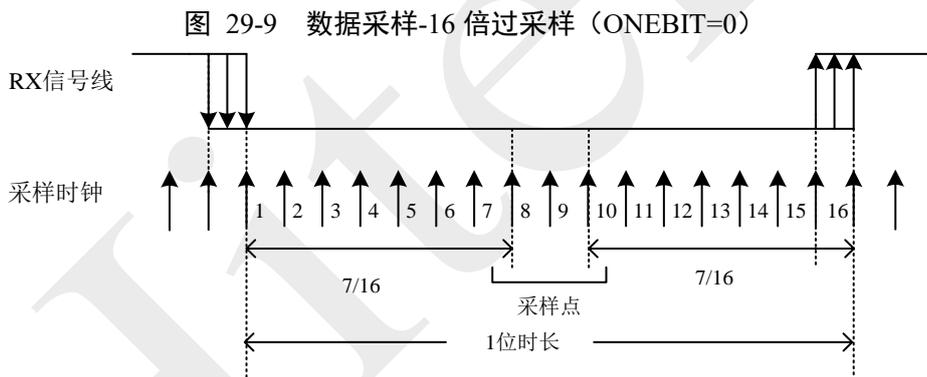
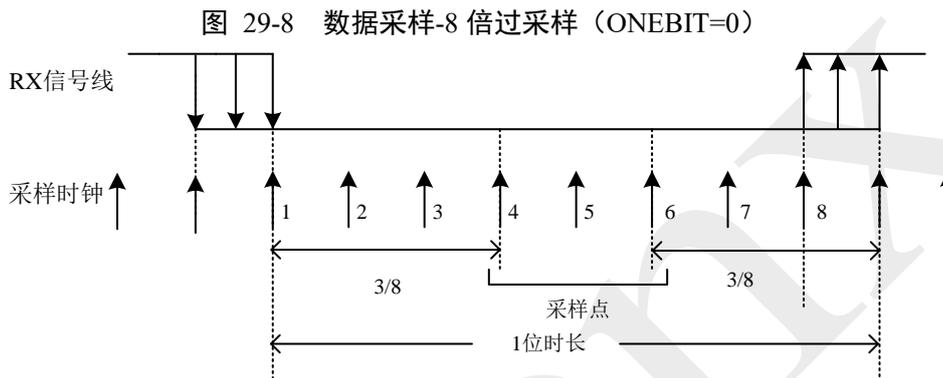


表 29-5 采样数据和噪声检测 (ONEBIT=0)

采样值	实际接收值	NOISE 位状态
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

帧错误：

接收数据时如果在预期时间内未能识别出停止位，则检测为帧错误，同步失败或噪声过大的情况下可能引发此错误。

检测到帧错误时有以下动作：

- 帧错误标志 FE 位置位。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器（使能 FIFO 时为 RXFIFO）。
- FE 位置 1 时，如果 USART_CR3 寄存器中的 EIE 位置 1 则触发中断。FE 置 1 时，RXNE 位（使用 FIFO 模式时为 RXFNE 位）也被置位，如果 RXNE 位置 1 会触发中断。

向 USARTx_ICR 寄存器中的 FECF 位写 1，清零 FE 位。

注意：同步模式不支持帧错误检测。

配置接收停止位：

在 USART_CR2 中的 STOP[1:0]位域中，配置预计接收的停止位的位数。

有以下几个选项：

- **0.5 位停止位：**选择 0.5 个停止位时，不会检测到帧错误和中断帧。
- **1 位停止位：**在第 8、第 9 和第 10 个采样沿采样 1 位停止位。
- **1.5 位停止位：**对 1.5 位停止位的采样在第 16、第 17 和第 18 个采样时钟进行。
- **2 位停止位：**对 2 位停止位的采样在对第一个停止位的第 8、9、10 次采样。如果在第一个停止位期间检测到帧错误，则帧错误标志位 FE 置 1。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位末尾时置 1。发生帧错误时不检测第 2 位停止位。

29.5.7 USART 波特率生成

接收和发送的波特率由编程 USARTx_BRR 寄存器得到。

有以下公式：

公式 1：标准 USART 的波特率，含同步模式(OVER8 = 0 或 1)

$$16 \text{ 倍过采样时(OVER8 = 0): Tx或Rx的波特率} = \frac{USART_KCLK_PRES}{USART_DIV}$$

$$8 \text{ 倍过采样时(OVER8 = 1): Tx 或 Rx 的波特率} = \frac{2 \times USART_KCLK_PRES}{USART_DIV}$$

公式 2：IrDA 模式下的波特率(OVER8 = 0)

$$Tx或Rx的波特率 = \frac{USART_KCLK_PRES}{USART_DIV}$$

USART_DIV 是一个无符号固定小数，存储在 USART_BRR 寄存器中。

- 当 OVER8 = 0 时， $BRR = USART_DIV$ 。
- 当 OVER8 = 1 时
 - $BRR[2:0] = USART_DIV[3:0] \gg 1$;
 - BRR[3] 需保持清零;
 - $BRR[15:4] = USART_DIV[15:4]$;

注意： 写 USARTx_BRR 寄存器后波特率计数器立即更新。因此不应在通信时更改波特率寄存器的值。

16 倍或 8 倍过采样时，USART_DIV 必须大于等于 16。

根据 USART_BRR 中的值计算 USART_DIV：

例 1：在 USART_KCLK_PRES = 16 MHz 时得到 9600 的波特率

- 16 倍过采样时：
 $USART_DIV = 16\ 000\ 000/9600$
 $BRR = USART_DIV = 0d1667 = 0x683$
- 8 倍过采样时：
 $USART_DIV = 2 \times 16\ 000\ 000/9600$
 $USART_DIV = 0d3333 = 0xD05$
 $BRR[3:0] = 0x5 \gg 1 = 0x2$
 $BRR = 0xD02$

例 2：在 USART_KCLK_PRES = 48MHz 时得到 921.6 K 的波特率

- 16 倍过采样时：
 $USART_DIV = 48\ 000\ 000/921\ 600$
 $BRR = USART_DIV = 0d52 = 0x34$
- 8 倍过采样时：
 $USART_DIV = 2 \times 48\ 000\ 000/921\ 600$
 $USART_DIV = 0d104 = 0x68$
 $BRR[3:0] = USART_DIV[3:0] \gg 1 = 0x8 \gg 1 = 0x4$
 $BRR = 0x64$

29.5.8 USART 接收容差

仅当时钟系统总偏差小于 USART 接收容差时，接收器才可以正常工作。影响

系统总时钟偏差的因素有以下几点：

- **DTR_ERR**：由发送器误差引起的偏差,其中包含发送器本地振荡器的偏差。
- **DQU_ERR**：接收器的波特率量化误差。
- **DRX_OSERR**：接收器本地振荡器的偏差。
- **DL_ERR**：传输线路引入的偏差（通常是由于收发器引入了上升沿下降沿的时序不对称）

需以下条件：

$DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR + DWKUP < USART$ 接收容差。

其中：

DWKUP 是从低功耗唤醒时，采样点偏差产生的误差。

在 $M[1:0] = 01$ （9 位字符长度）时： $DWKUP = \frac{t_{WUUSART}}{11 \times Tbit}$ ；

在 $M[1:0] = 00$ （8 位字符长度）时： $DWKUP = \frac{t_{WUUSART}}{10 \times Tbit}$

在 $M[1:0] = 10$ （7 位字符长度）时： $DWKUP = \frac{t_{WUUSART}}{9 \times Tbit}$

其中 $t_{WUUSART}$ 为从低功耗模式唤醒的时间(参见数据手册)。

USART 接收器在表 *BRR [3:0] = 0000 时的 USART 接收容差*和表 *BRR [3:0] ≠ 0000 时的 USART 接收容差*中说明了可正确接收数据的最大接收容差，影响接收容差的条件有以下几点：

- 由 USARTx_CR1 寄存器中的 M 位域设置的 9、10 或 11 位字符长度。
- 由 USARTx_CR1 寄存器中的 OVER8 位设置的 8 倍或 16 倍过采样。
- USARTx_BRR 寄存器的 BRR[3:0] 位是否等于“0000”。由于需要支持过采样时钟，所以约束 USART_DIV 不小于 16，反映到 USARTx_BRR 寄存器中为“10h”。BRR[3:0]如果等于“0000” USART_DIV 为 16 的整数倍，BRR[3:0]不等于“0000”时引入了小数部分，所以 BRR[3:0]等于“0000”时接收容差更大。
- 在 USARTx_CR3 寄存器中 ONEBIT 位配置的单次采样（ONEBIT=1）或 3 次采样多数表决（ONEBIT=0）的方式。

表 29-6 BRR [3:0] = 0000 时的 USART 接收容差

M[1:0]	OVER8=0, 16 倍过采样		OVER8=1, 8 倍过采样	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00 (8bits)	3.75%	4.375%	2.50%	3.75%

01 (9bits)	3.41%	3.97%	2.27%	3.41%
10 (7bits)	4.16%	4.86%	2.77%	4.16%

表 29-7 BRR [3:0] ≠ 0000 时的 USART 接收容差

M[1:0]	OVER8=0, 16 倍过采样		OVER8=1, 8 倍过采样	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00 (8bits)	3.33%	3.88%	2%	3%
01 (9bits)	3.03%	3.53%	1.82%	2.73%
10 (7bits)	3.7%	4.31%	2.22%	3.33%

注意：当接收帧中恰好包含 10 bits (M 位 = 00 时)、11 bits (M 位 = 01 时) 或 9 bits (M 位 = 10 时) 的空闲帧时，以上表格中的数据可能略有不同。

29.5.9 USART 自动波特率检测

USART 可以根据一个接收字符来检测波特率，并自动调整 USARTx_BRR 寄存器的值。在自动波特率模式下，字符接收期间波特率被多次测量并自动调整 详见 [图：波特率自动检测的四种模式](#)。

自动波特率检测适用以下两种应用场景：

- 与通信速率未知的系统通信；
- 系统自身的时钟源精度较低时，使用自动波特率功能无需测量时钟偏差，即可得到正确的波特率；

使能自动波特率检测功能需要 USART 时钟源支持预期的通信速率：

- 16 倍过采样下支持的波特率范围为：
USART_KCLK_PRE/65535 ~ USART_KCLK_PRE /16。
- 8 倍过采样时，波特率范围为：
USART_KCLK_PRE/65535 ~ USART_KCLK_PRE /8。

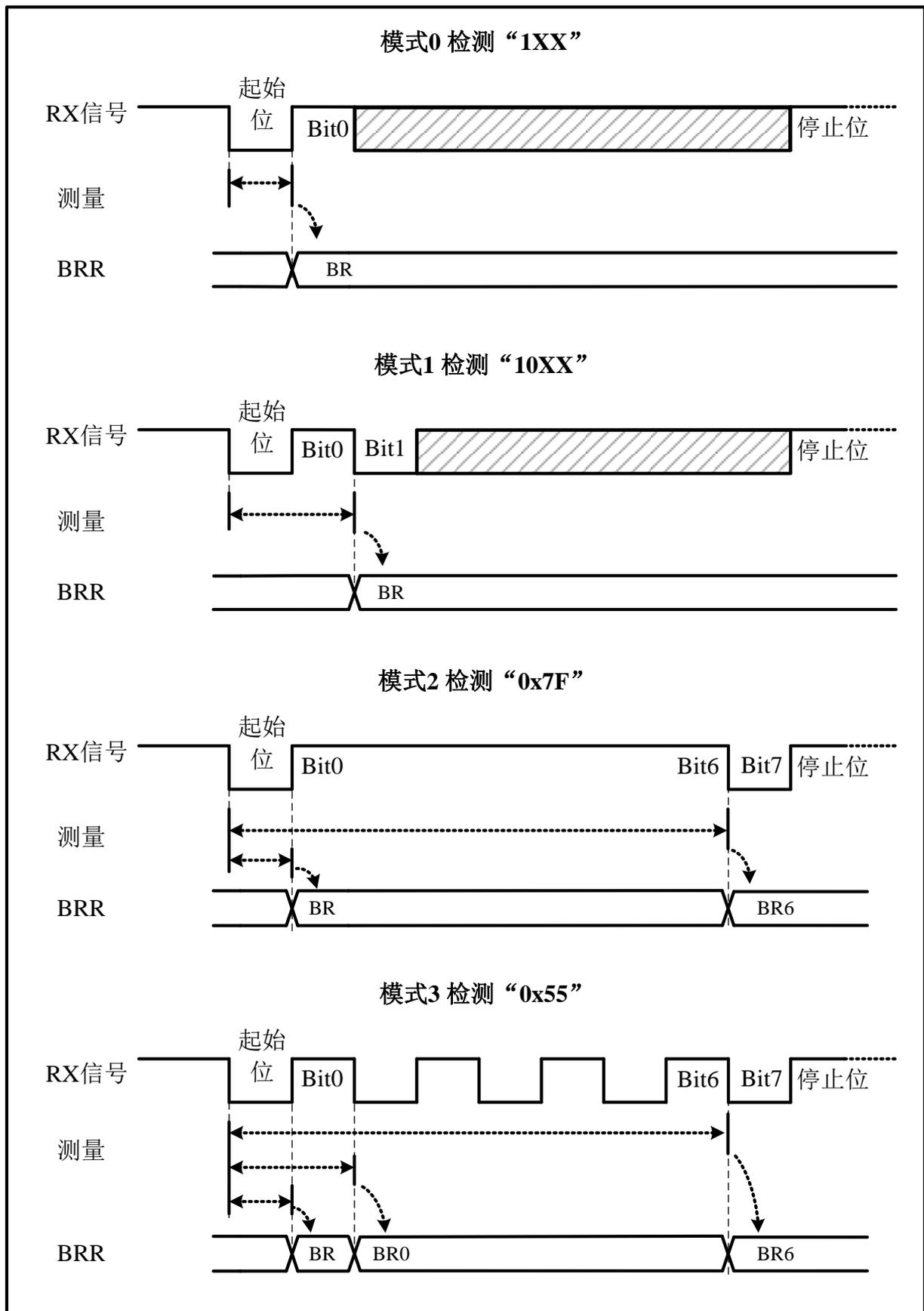
使能自动波特率检测时，先配置 USART_CR2 寄存器中的 ABRMOD[1:0] 位域选择检测模式后，才可以置位 ABREN 使能。

检测模式有以下四种，参见 [图：波特率自动检测的四种模式](#)：

- 模式 0：首位为“1”的字符。
此模式测量起始位的持续时间（由下降沿到上升沿），并在起始位结束时更新 BRR 寄存器。

- 模式 1：以“10xx”开头的字符。
此模式测量起始位和首个数据位的持续时间，即测量下降沿到下降沿的间隔，适用于信号沿斜率小的场景，可以得到较高的精度。
- 模式 2：“0x7F”字符
LSB 优先时为“0x7F”，MSB 优先时为“0xFE”。
此模式下，波特率在起始位结束时更新为 BR，然后在 Bit6 结束时根据下降沿到下降沿测得的 BR6，再次更新波特率为 BR6。
Bit0~Bit6 的采样使用 BR，Bit6 结束后使用 BR6 采样后续字符。
- 模式 3：“0x55”字符
此模式下，波特率在起始位结束时更新为 BR，然后在 Bit0 结束时根据下降沿到下降沿测得的 BR0 再次更新，最后在位 Bit6 结束时更新波特率到 BR6。
Bit0 的采样使用 BR，Bit1~Bit6 的采样使用 BR0，最后在 Bit6 结束后使用 BR6 采样后续字符。此模式还会对 RX 线路的逻辑翻转执行另外的检查。如果 RX 线上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

图 29-10 波特率自动检测的四种模式



使能自动波特率检测前，需将 USARTx_BRR 寄存器预设任意值作为初始化。

USARTx_CR2 寄存器中的 ABREN 位置 1 时开启自动波特率检测。此后 USART 开始等待 RX 线路上的字符。

自动波特率检测完成后 USARTx_ISR 寄存器中的 ABRF 标志自动置 1。

线路噪声过大时，波特率检测可能无法正常运行，BRR 的值可能受到影响甚至被破坏，此时错误标志位 ABRE 置 1。当通信速率不在自动波特率检测范围内时，也会置位错误标志位 ABRE。例如，在 16 倍过采样时 1bit 持续时间不在 16 ~ 65536 个时钟周期之内，或 8 倍过采样时不在 8~65536 个时钟周期之内。

将 USART_RQR 寄存器的 ABRRQ 位置 1 可重启自动波特率检测，ABRF 标志自动清零。

禁用 FIFO 发生自动波特率错误时，ABRE 标志和接收非空标志 RXNE 以及帧错误标志 FE 位同时置 1。

使能 FIFO 发生自动波特率错误时 ABRE 标志和接收 FIFO 非空标志 RXFNE 以及帧错误标志 FE 位同时置位。

使能 FIFO 时应使用 RXFIFO 中最先接收到数据进行自动波特率检测。因此，在使能自动波特率检测之前，需要查询 USART_ISR 寄存器的 RXFNE 标志为 0 以确认 RXFIFO 为空。

注意： 如果在进行自动波特率检测期间禁用 USART ($UE=0$)，则 BRR 的值可能被破坏。

29.5.10 USART 多机通信

USART 多机器通信，即多个 USART 连接在同一个网络中。

例如，一个 USART 作为主设备，其 TX 引脚输出到其他 USART 的 RX 引脚；作为从设备的多个 USART 将各自的 TX 引脚连接在一起做线与，连接到 USART 主设备的 RX 引脚。

在多机通信中，最理想的状态是，只有期待的接收器响应，并主动接收完整的消息内容，这样可以减少由未被寻址的接收器造成的冗余开销。

多机通信目前仅支持无校验的方式。

静默模式就是为此目标而设计，多机通信时，地址不匹配的接收器进入静默模式。通过 USART_CR1 寄存器的静默模式使能位 MME 置 1，使能静默模式。

注意： 使能 FIFO 且 MME 已置 1 时，不推荐清零 MME 位再快速置 1（在两个 USART_KCLK_PRE 周期内），此操作可能因清零状态无法被识别而导致静默模式持续有效。

静默模式使能时：

- 接收状态位不会被置 1；
- 任何接收中断都被禁止；

- USART_ISR 寄存器中的 RWU 位(静默模式状态指示位)不会立即置 1。在不同的场景下, RWU 位可由软件置位(向 USART_RQR 寄存器中的 MMRQ 写 1), 或由硬件自动控制, 详情请参见图: [使用地址匹配检测时的静默模式](#);

静默模式的唤醒选择条件可由 USART_CR1 寄存器中的 WAKE 配置, 具体如下:

- WAKE 位为 0 时, 使用空闲帧检测;
- WAKE 位为 1 时, 使用地址匹配检测。

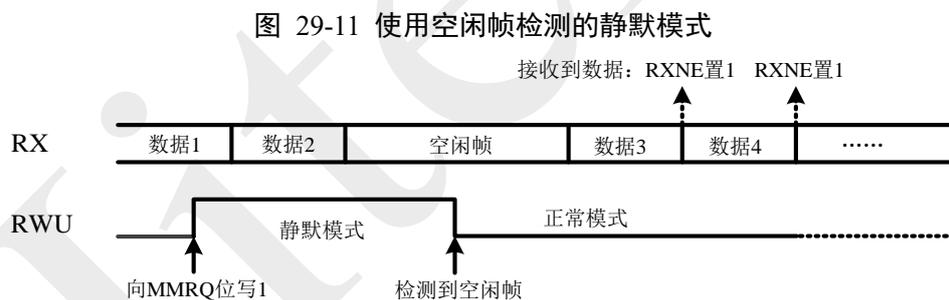
使用空闲帧检测 (WAKE=0)

向 USART_RQR 寄存器的 MMRQ 位写 1 进入静默模式, RWU 位自动置 1, (指示 USART 已进入静默模式)。

当检测到空闲帧时, USART 退出静默模式。此时 RWU 位自动清零, 但 USARTx_ISR 寄存器中的空闲线路标志 IDLE 位不会置 1。

使用空闲帧检测的网络中, 主设备发送空闲帧时所有的从设备同时从静默模式退出, 实现多从设备同步。

使用空闲帧检测时, 静默模式的动作参见下图:



注意: 需在下一空闲帧到来之前将 MMRQ 位置 1 进入静默模式。由于检测到下一空闲帧时, 执行从静默模式退出, 随后判断是否重新进入静默模式。所以在空闲状态到来后, 再将 MMRQ 位置 1 不会进入静默模式, RWU 也不被置 1。实际上 USART 不只在接收一个空闲帧后可能检测到空闲状态, 如果在线路处于空闲状态时使能 USART, 只要在一个空闲帧的时间中检测到空闲, USART 也识别为空闲状态。

主设备也可以不通过 TE (翻转发送器使能位) 的方式主动产生空闲字符, 只要发送完一帧数据后延时不小于 1 个空闲帧的时间, 从机也会检测到空闲状态。

使用地址匹配检测 (WAKE=1)

在此模式下, 如果字符的 MSB 为 1, 则将此字符识别为“地址”, 否则识别为“数据”。

在地址字符中，检测地址可以为 4bits 或 6、7、8bits（LSB）。接收器本地的地址在 USARTx_CR2 寄存器的 ADD 位域中设置，设置方法如下：

- ADDM7 为 0 时，地址长度为 4bits，保存在 ADD[3:0]；
- ADDM7 为 1 时，7bits 字符长度时，地址长度为 6bits，保存在 ADD [5:0] ；
8bits 字符长度时，地址长度为 7bits，保存在 ADD [6:0]；
9bits 字符长度时，地址长度为 8bits，保存在 ADD [7:0]；

当接收的地址字符与本地地址不匹配时，USART 接收器进入静默模式。此时，RWU 位由硬件置 1。进入静默模式后，RXNE 位不会被此地址字符置 1，也不会触发中断或产生 DMA 请求。使能 FIFO 时，应确认在进入静默模式之前，RXFIFO 中至少有一个未使用空间。

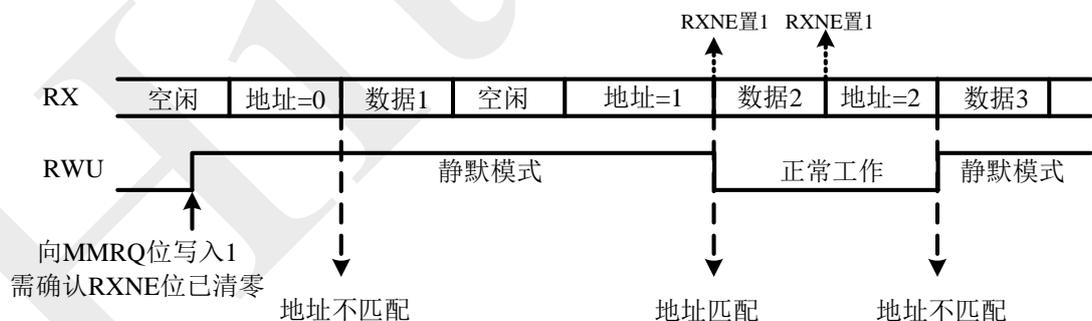
当向 MMRQ 位写 1 时，USART 进入静默模式，此时 RWU 位也自动置 1。

当接收到的地址与本地地址匹配时，USART 接收器退出静默模式，RWU 位清零，此后可以开始正常接收后续字符。由于 RWU 位已清零，RXNE/RXFNE 位会被此地址字符置 1。

注意： 使能 FIFO 时，如果在接收器对字符的最后一位采样时 MMRQ 置 1，则进入静默模式之前有可能接收到该数据。

下图中给出了使用地址匹配检测时的示例：

图 29-12 使用地址匹配检测时的静默模式



29.5.11 USART Modbus 通信

USART 提供对 Modbus/RTU 和 Modbus/ASCII 协议的基本支持。

Modbus 是一个半双工块传输协议。协议的控制功能需由软件实现，如地址识别、块完整性控制和命令解析等。

USART 为块尾检测提供基本功能支持，无需软件或其他资源的开销。

Modbus/RTU:

在此模式下，一个块的块尾通过超过 2 个字符时间的“静默”（总线空闲）来识别。此功能可通过配置超时时间实现。

超时功能和中断分别通过 USARTx_CR2 寄存器中的 RTOEN 位和 USARTx_CR1 寄存器中的 RTOIE 位使能。USART_RTOR 寄存器的 RTO 位域需配置为对应 2 个字符时间（例如 22 个 bits 持续时间）的值。如果在此期间接收到线路空闲，则在接收到最后一个停止位后触发中断，指示当前块接收已完成。

Modbus/ASCII:

在此模式下，一个块的块尾通过特定字符序列(CR/LF)识别。USART 通过字符匹配功能实现此机制。

LF 的 ASCII 码写入在 ADD[7:0] 位域中，使能字符匹配中断 (CMIE=1)，接收到 LF 时触发中断，使能 DMA 通信时，接收到中断后可检查 DMA 缓存区中的 CR/LF。

29.5.12 USART 同步模式

主模式

将 USART_CR2 寄存器中的 CLKEN 位置 1 使能同步主模式。

使能同步模式需将 USART_CR3 寄存器中的 HDSEL 和 IREN 位清零。

同步主模式下，USART 作为主设备控制双向串行通信，有如下配置：

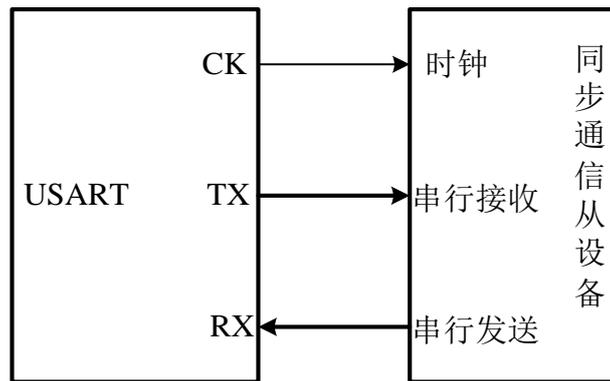
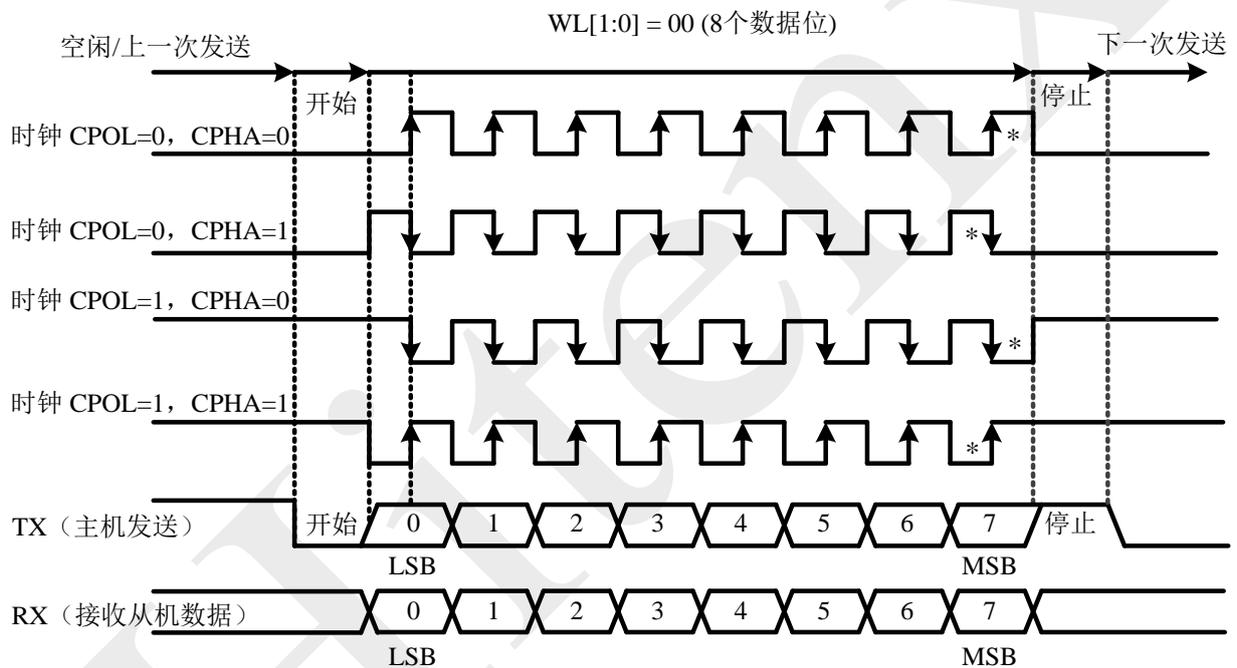
- CK 引脚是 USART 时钟输出，在起始位或停止位期间无时钟脉冲输出。
- USART_CR2 寄存器的 LBCL 位，用于控制字符的最后一个数据位期间是否输出时钟脉冲（例如用于地址匹配时）。
- USART_CR2 寄存器中的 CPOL 位用于选择时钟极性。
- USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位。

（详情请见：[同步主模式数据-时钟示意图 1](#)；[同步主模式数据-时钟示意图 2](#)。）

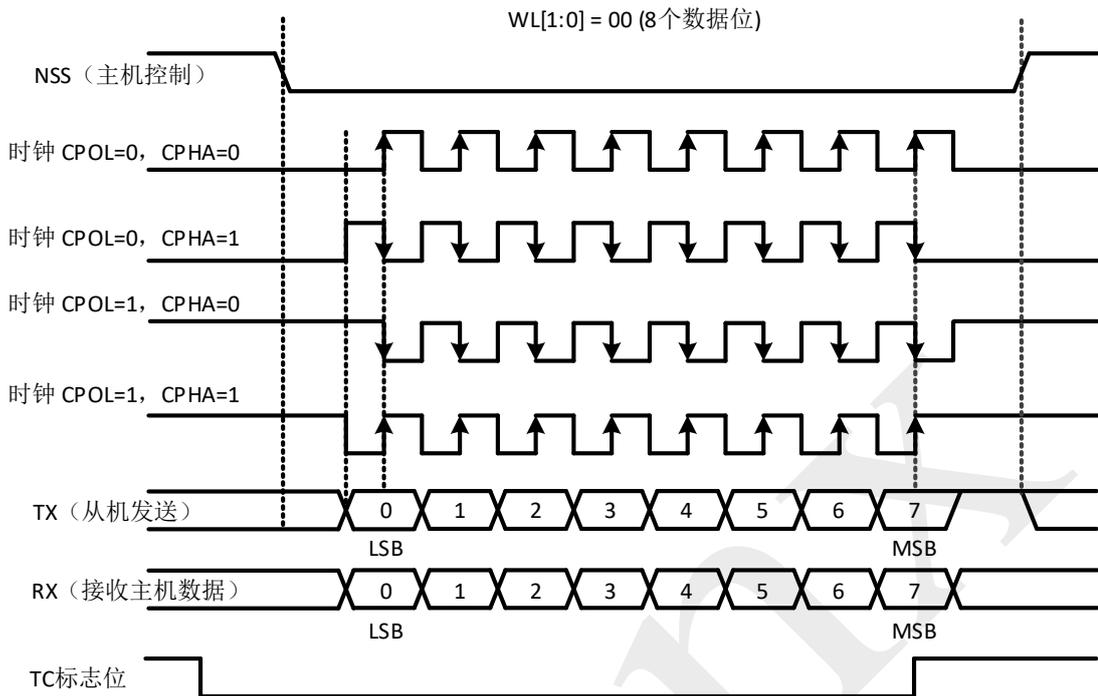
在空闲状态、空闲帧和发送中断帧期间，CK 时钟无输出。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1 使能接收，则数据在 CK 引脚输入的时钟沿上采样，根据 CPOL 和 CPHA 的设置，采样沿可配置为上升沿或下降沿。同步模式无过采样功能。

可使用 GPIO 做从机片选控制。

图 29-13 同步通信主模式下的连接例

图 29-14 同步主模式数据-时钟示意图 1 (M[1:0]=“00”)


*注： USART_CR2寄存器中的LBCL位配置是否在最后一个数据位发送时钟脉冲，箭头表示数据采样边沿

图 29-15 同步主模式数据-时钟示意图 2 (M[1:0]=“01”)


注：箭头表示数据采集边沿，
在同步从模式下，只能通过TC来判断数据是否已经从移位寄存器中移出到TX线上

从模式

通过将 USART_CR2 寄存器中的 SLVEN 位置 1 使能同步从模式。同步从模式下，需将以下配置清零：

- USART_CR2 寄存器中的 CLKEN 位。
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

同步从模式下 CK 引脚为输入。

注意： 作为同步从设备时，USART 时钟 USART_KCLK_PRES 的频率必须是 CK 引脚输入时钟频率的 3 倍以上。

在同步从模式下，只能通过判断 TC 标志位把数据写入 RDR 寄存器，在此模式下 TXE/TXFNF 标志位失效。

在同步从模式下，不支持使用 DMA 方式将数据从内存搬运至发送寄存器 TDR 进行发送。

在同步从模式下，当使用第二边沿 $cpha=1$ 时，不支持相邻两个发送字节的最高数据位由 0 至 1 变化。

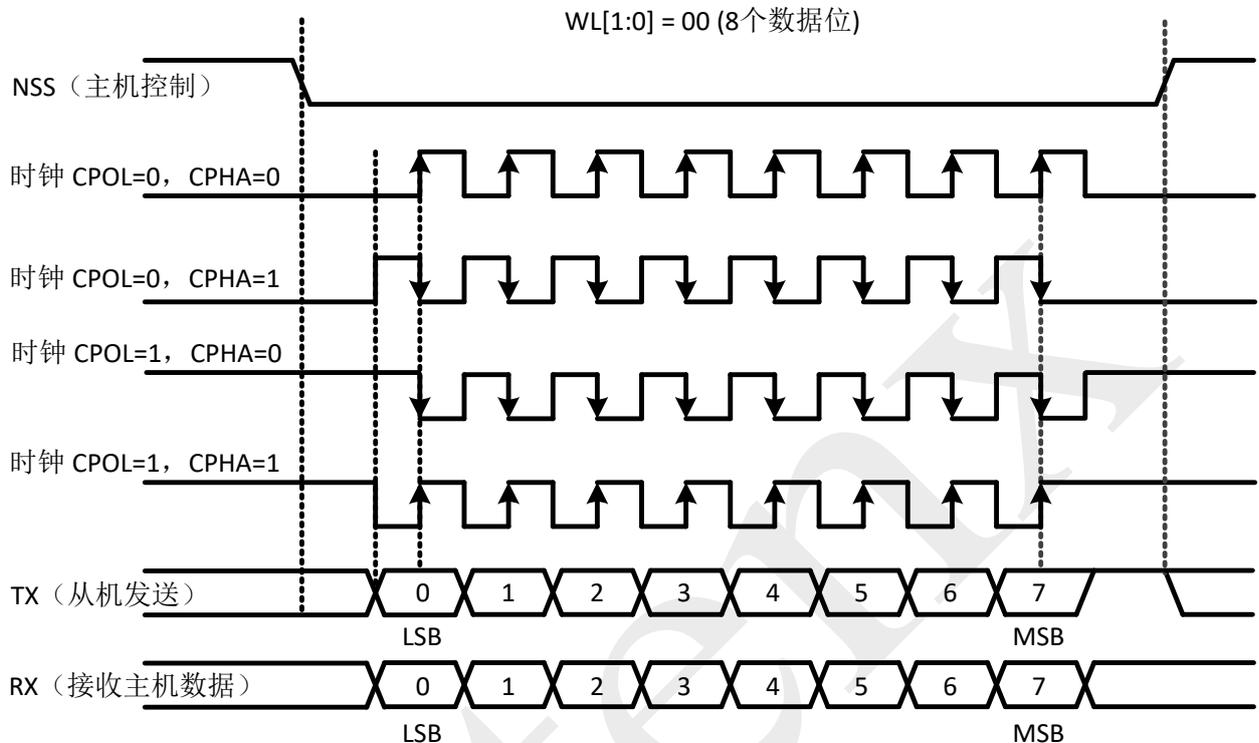
USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟的极性和相位。参见：[同步从模式数据-时钟示意图](#)。

从发送模式支持下溢错误检测。如果在 USART_TDR 寄存器未写入数据前，出

现第一个数据发送时钟脉冲，则下溢错误标志位 UDR 置 1。

从模式下支持配置为硬件或软件片选 (NSS)。

图 29-16 同步从模式数据-时钟示意图 (M[1:0]=“00”)



注：箭头表示数据采集边沿

从设备片选引脚(NSS)管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位，可将从设备片选管理配置为硬件管理或软件管理：

- 使用软件 NSS 管理 (DIS_NSS = 1)，
同步从设备一直处于被选中的状态，并忽略 NSS 引脚输入。
NSS 引脚空闲可作为其他用途。
- 使用硬件 NSS 管理 (DIS_NSS = 0)
同步从设备片选取决于 NSS 引脚输入。NSS 为低电平时选中，高电平时取消选中。

注意： 为了确保时钟正常工作，注意需在 USART 禁用 (UE=0) 时，配置 LBCL (仅用于同步主模式)、CPOL 和 CPHA。

在同步从模式下，需在主设备发起通信之前，或是有稳定时钟下，且需在相邻帧之间使能 USART。否则，如果 USART 从设备在主设备处于某一帧期间使能，从设备与主设备无法同步。在通信时钟的第一个边沿到来之前，以及当前通信结束之前，从设备的发送数据寄存器必须准备就绪，否则从设备会发送 0。

从设备下溢错误

在从设备发送过程中，如果第一个时钟沿到来时，USART_TDR 中尚未填充数据，就会生成下溢错误。

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志置 1，从设备持续发送最后一个字符，直到下溢错误标志由软件清零。

如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

向 USART_ICR 寄存器中的 UDRCF 位写 1，可清零下溢错误标志 UDR。

发生下溢错误时，可将新数据写入 TDR 寄存器，清除下溢错误标志后新数据才可以发出。如果发生下溢错误且没有新数据被写入 TDR，则 TC 标志在帧结束时置 1。

注意：如果第一个发送时钟的边沿过于接近向 USART_TDR 写入数据的时间点，也可能发生下溢错误。为避免发生下溢错误，应在第一个发送时钟边沿前至少 3 个 USART_KCLK 周期写入 USART_TDR。

29.5.13 USART 单线半双工通信

通过将 USARTx_CR3 寄存器中的 HDSEL 位置 1 来开启单线半双工通信模式。在此模式下需将以下位清零：

- USARTx_CR2 寄存器中的 CLKEN 位。
- USARTx_CR3 寄存器中的 IREN 位。

USART 可支持单线半双工通信协议，此时 TX 和 RX 信号在内部相连接。

向 HDSEL 位写 1 后使能单线半双工模式，有以下动作：

- TX 和 RX 信号在内部相连接。
- RX 引脚不使用，可做 GPIO 使用。
- 无数据传输时 TX 引脚释放，表现为处于空闲或接收状态的 I/O。需要将 TX 引脚配置为复用开漏模式，并外接上拉电阻。

除以上几点以外，通信协议与正常 USART 通信类似。通信线路上的冲突需由软件管理，尤其要注意，只要数据在 TE 位置 1 时写入数据寄存器 TDR，发送就会持续进行，硬件不会主动阻止发送过程。

29.5.14 USART 接收超时

通过 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能接收超时功能。

超时间隔在 USART_RTOR 寄存器中的 RTO[15:0] 位域编程。

接收超时计数器开始计数的时间点如下描述：

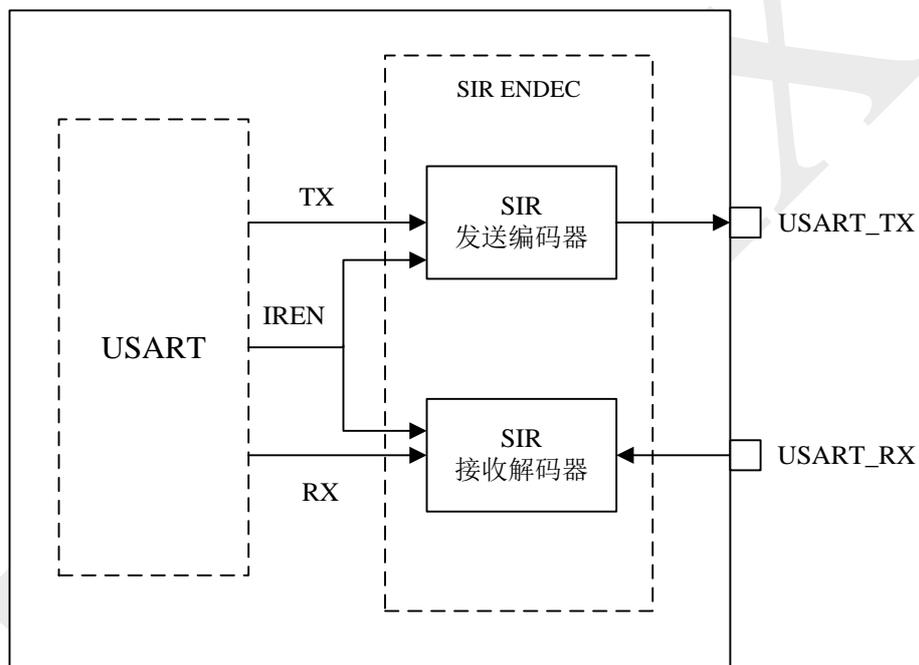
- STOP = “00” (1 位停止位)、STOP = “11” (1.5 位停止位) 和 STOP = “10” (2 位停止位) 时，在停止位结束时开始计数。
- STOP = “01” (0.5 位停止位) 时从停止位开始时开始计数。

超时间隔结束后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则触发超时中断。

29.5.15 USART IrDA SIR 编解码模块

本章内容仅适用于 IrDA 通信模式。

图 29-17 IrDA 功能使能时 USART 内部结构框图



通过将 USART_CR3 寄存器中的 IREN 位置 1 使能 IrDA 模式。在 IrDA 模式下需清零以下寄存器：

- USART_CR2 寄存器中的 STOP 和 CLKEN 位。
- USART_CR3 寄存器中的 HDSEL 位。

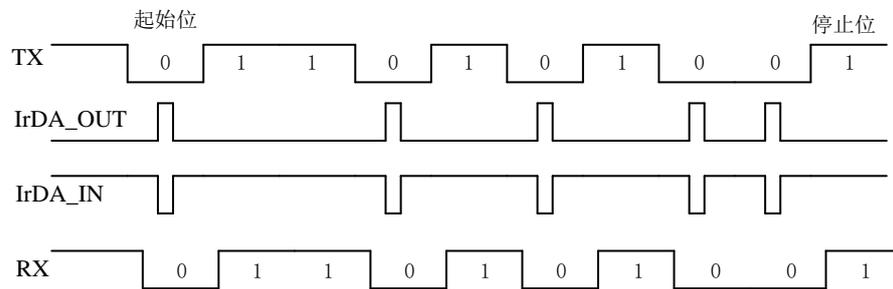
IrDA SIR 物理层规定使用反相归零 (RZI) 的调制方式，并以红外光脉冲表示逻辑 “0”，参见 [图：标准模式下 IrDA SIR 数据调制](#)。

SIR 发送编码器用于将 USART 的发送数据流调制为非归零 (NRZ) 位流。发送脉冲位流输出到外部驱动器并驱动红外线 LED。USART 仅支持最高 115.2 Kbps 的 SIR 编解码比特率。在正常模式下，发送的脉冲宽度约定为一位周期的 3/16。

SIR 接收解码器将红外接收器输出的归零位流转换为 NRZ 串行位流，并输出到 USART。在空闲状态下，接收解码器输入为常高。发送编码器输出的极性与

解码器输入相反。当解码器输入低电平时检测到起始位。参见下图：

图 29-18 标准模式下 IrDA SIR 数据调制



- IrDA 协议为半双工通信协议。USART 向 IrDA 编码器发送数据时，IrDA 解码器会忽略接收线上的数据。USART 接收 IrDA 解码器数据时，IrDA 编码器不会响应 USART 发送到 TX 线上的数据。所以软件应避免在接收数据的同时执行发送，这样会破坏即将发送的数据。
- 发送编码器将“0”作为高电平脉冲发送，“1”作为低电平发送。正常模式下脉冲宽度规定为 1bit 周期的 3/16。
- ISIR 接收解码器将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 空闲时发送编码器输出为低电平，与接收解码器的输入反相
- IrDA 规范要求脉冲的许容值要大于 1.41 μ s。脉冲宽度的许容值可通过 USART_GTPR 寄存器可配置。接收器端的滤波器会滤除宽度小于 2 个采样时钟的脉冲，波特率时钟在 USART_GTPR 中 PSC[4:0]位域可配置。宽度小于 1 个波特率时钟周期的脉冲都将被滤除，宽度介于 1 个到 2 个波特率时钟周期之间的脉冲有可能被接收也有可能被滤除，宽度大于 2 个波特率时钟周期的脉冲作为有效脉冲被接收。当 PSC[7:0]=0 时 IrDA 编码器/解码器不工作，不建议使用。
- 支持与低功耗发送器通信。
- 在 IrDA 模式下仅支持配置为 1 位停止位。
- IrDA 下仅支持 16 倍过采样。

IrDA 低功耗模式

- 发送器

低功耗模式下，脉冲宽度不是 1bit 周期的 3/16，而是低功耗波特率的 3 倍。波特率最小可为 1.42 MHz。标准模式下对应波特率为 1.8432 MHz (115.2 Kbps*16) 波特率范围控制在 1.42 MHz < PSC < 2.12 MHz。低功耗模式下通过可编程分频器对系统时钟分频，以达到波特率范围。

- 接收器

低功耗模式下的接收与正常模式下的接收类似。为实现抗干扰，USART 会滤除时间短于 1 个波特时钟周期的脉冲，只有当宽度大于 2 个低功耗波特时钟周期时才识别为有效低电平。

注意： 宽度介于 1 个到 2 个 PSC 周期之间的脉冲可能被接收也可能被拒绝。
接收器的建立时间由软件配置。由于是半双工协议，根据 IrDA 物理层规定发送和接收之间至少要有 10 ms 的间隔。

29.5.16 USART 使用 DMA 通信

USART 支持通过 DMA 进行通信。DMA 对接收缓冲区和发送缓冲区分别发起独立的访问请求。

使用 DMA 发送：

将 USART_CR3 寄存器中的 DMAT 位置 1，使能 DMA 发送。当 TXE 标志（使能 FIFO 时为 TXFNF 标志）置 1 时，可将数据从 DMA 中预设好的存储区加载到 USART_TDR 寄存器。参考以下步骤建立从 USART 到 DMA 的映射（x 表示通道编号）：

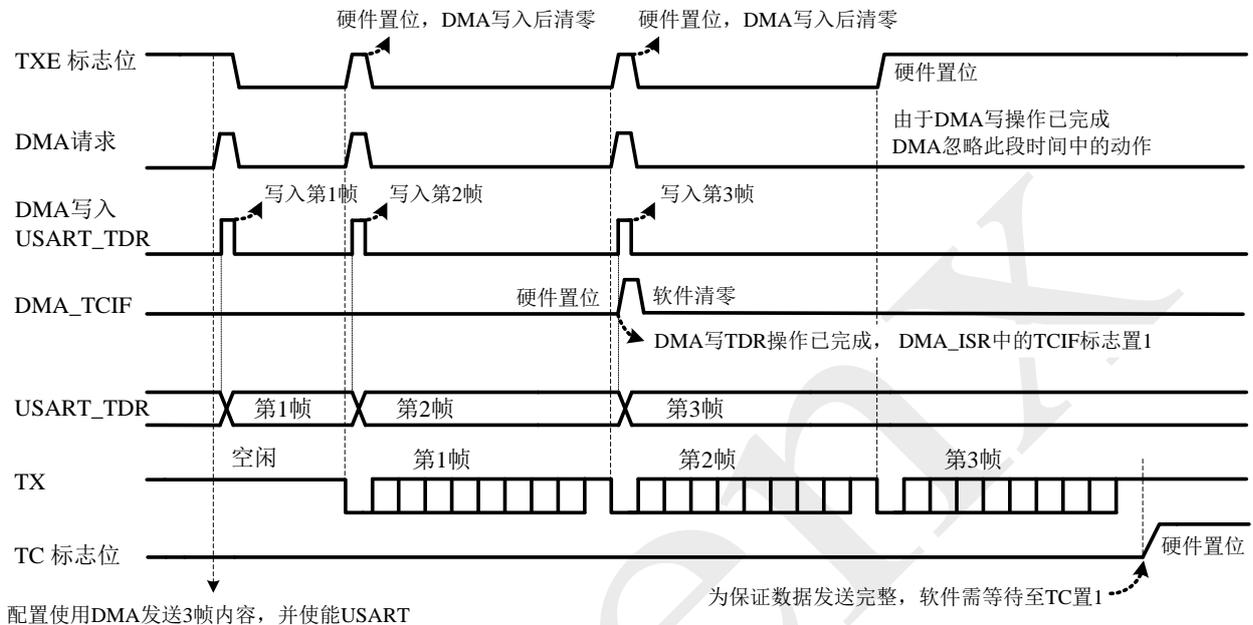
- 在 DMA 通道 x 控制寄存器 DMA_CCRx 中配置通道的优先级。
- 配置 DMAMUX 输出请求。参见：*DMAMUX 配置流程*。
- 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要传输的总字节数。
- 在 DMA 源地址寄存器 DMA_CSARx 中配置存储区地址作为传输源地址，每次 TXE（使能 FIFO 模式时为 TXFNF）置位时，数据都从此存储区加载到 USART_TDR 寄存器中。
- 在 DMA 目的地址寄存器 DMA_CDARx 中配置 USART_TDR 寄存器为传输目的地址。每次 TXE（使能 FIFO 模式时为 TXFNF）置位时，数据加载到 USART_TDR 寄存器。
- 根据具体应用配置中断触发的条件。
- 向 USART_ICR 寄存器中的 TCCF 位写 1，清零 USART_ISR 寄存器中的 TC 标志。
- 将 DMA_CCRx 寄存器中的 EN 位置 1 以使能通道。

当达到 DMA 控制器中设置传输数据次数时，会在 DMA 相应通道上产生中断。在发送模式下，DMA 完成了对所有需要发送的数据的写操作时，DMA_ISR 寄存器中的 TCIF 标志置 1，此外，还可以通过检测 USART_ISR 寄存器中的 TC 标志位确认 USART 通信是否完成。TC 标志在数据发送期间保持清零，在最后一个字符发送结束后由自动置 1。需要注意，在禁用 USART 之前或进入低功耗

耗模式之前，需要确认最后一个数据发送完成。为避免破坏最后一个字符的传输，需要等待 TC 置位后才可以禁用 USART 或进入低功耗模式。

图 29-19 USART 通过 DMA 发送示意图

图中描述了USART通过DMA发送3帧内容的过程：



注意： 使能FIFO时DMA请求由发送FIFO未满足事件触发，即TXFNF=1。

使用 DMA 接收：

将 USART_CR3 寄存器中的 DMAR 位置 1，使能 DMA 接收。

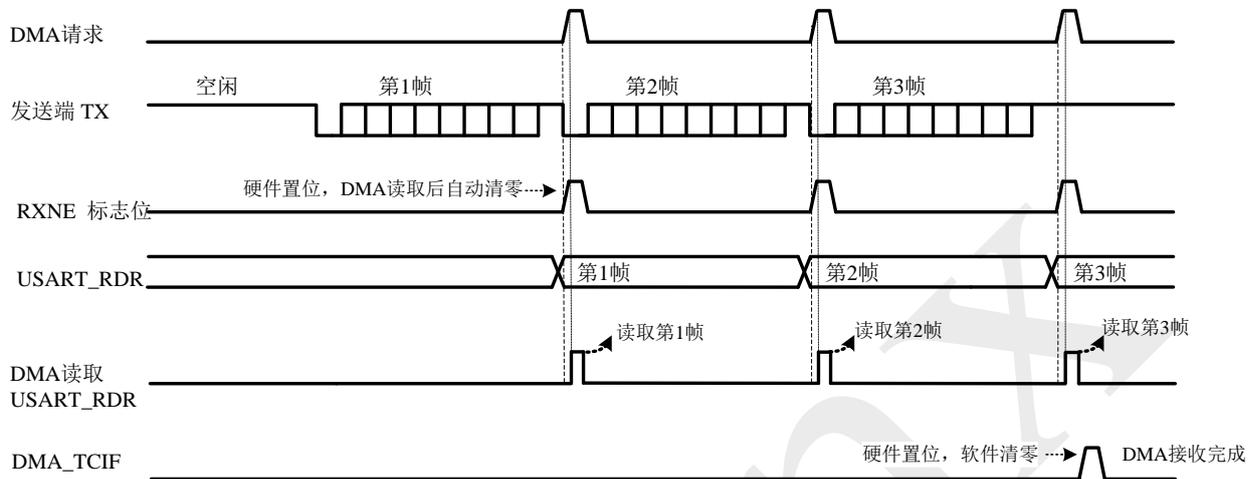
接收字符时，数据从 USART_RDR 寄存器加载到 DMA 配置的存储区域中。参考以下步骤建立 USART 到 DMA 通道到的映射（x 表示通道编号）：

- 在 DMA 通道 x 控制寄存器 DMA_CCRx 中配置通道的优先级。
- 配置 DMAMUX 输出请求。参见：DMAMUX 配置流程。
- 在 DMA 控制寄存器中配置需要接收的总字节数。
- 将 USART_RDR 寄存器地址写入源地址寄存器 DMA_CSARx 中，配置为传输源地址。每次发生接收非空 RXNE（使能 FIFO 模式时为 RXFNE）置 1 时，数据都从源地址复制到 DMA 配置的存储区。
- 将存储区地址写入目的地址寄存器 DMA_CDARx 中，配置为目的地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都由从 USART_RDR 加载到此存储区域。
- 根据具体应用配置触发中断的条件。
- 将 DMA_CCRx 寄存器中的 EN 位置 1 以使能通道。

当达到在 DMA 控制器中设置传输数据次数时，会在 DMA 相应通道上产生中断。

图 29-20 USART 通过 DMA 接收数据示例

本图对应发送示意图发送过程中，USART通过DMA接收三帧内容的过程：



注意：使能FIFO时DMA请求由接收FIFO未满足事件触发，即RXFNF=1。

DMA 通信模式下的错误标志及中断：

在使用 DMA 通信发生错误时，在当前字节传输结束后将错误标志置位，如果中断使能位置 1，则同时触发中断。字符接收过程中，出现帧错误，上溢错误或噪声错误时这些错误和 RXNE（使能 FIFO 模式时为 RXFNE）一同置位，USART 为错误中断配备有单独的中断使能位，位于 USART_CR3 寄存器中的 EIE 位，此位置 1 时，产生任一错误都会在当前字节结束后触发中断。

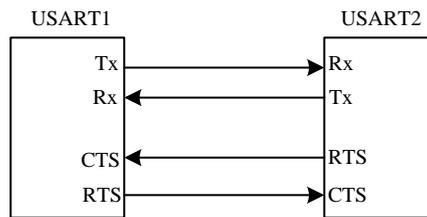
在 USART_CR3 寄存器的 DDRE 位可配置在接收出错时禁止 DMA。默认情况下（DDRE=0）接收出错时不禁止 DMA，这种情况下，相应的错误标志置 1，但 RXNE 保持为 0，不产生 DMA 请求，避免了上溢，错误数据不会被接收，但下一个正确数据可以被接收。DDRE=1 时，接收出错后，相应的错误标志和 RXNE 均置 1。但 DMA 请求会被屏蔽，直到错误标志清零后才会产生 DMA 请求。这意味着出错时，软件需先禁止 DMA 请求(DMAR = 0)或者将 RXNE 清零，然后再将错误标志清零后才可以继续接收。

29.5.17 RS232 硬件流控和 RS485 控制

通过 CTS 和 RTS 可以控制 2 个设备间通信的串行数据流。

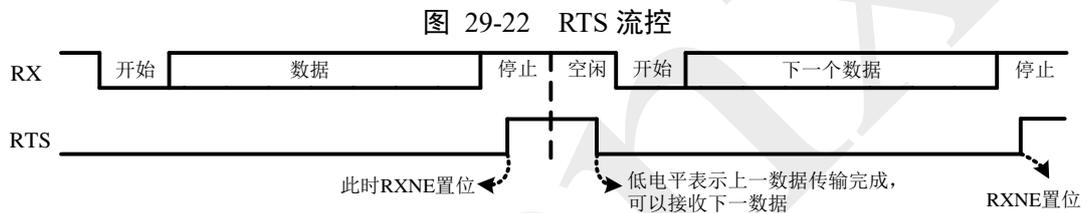
USART_CR3 寄存器中的 RTSE 位和 CTSE 位置 1，可分别使能 RTS 和 CTS 流控。

参见下图连接示意图：

图 29-21 两个 USART 间串行通信连接例


RS232 RTS 流控

将 RTSE 位置 1 使能 RTS 流控后，USART 准备好接收新数据时会将 RTS 置有效（低电平）。接收数据寄存器满，无法继续接收数据时，将 RTS 置为无效（高电平），此时表示传输应在当前帧结束后停止。使用 RTS 流控的示例如下图：

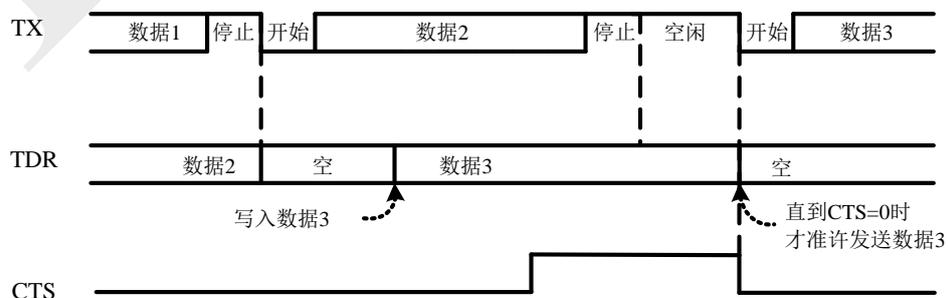


注意：使能 FIFO 时由 RXFIFO 状态控制 RTS。

RS232 CTS 流控

使能 CTS 流控（CTSE=1）时发送器会在发下一帧前检查 CTS。如果 CTS 有效（低电平），则允许下一数据发送，此时只要发送数据准备好，即 TXE/TXFNF=0 时，即开始发送，否则暂停发送。如果在发送过程中 CTS 变为无效（高电平），则完成当前发送后停止发送。

使能 CTS 流控（CTSE=1）时，只要 CTS 发生变化，CTSIF 状态位会随 CTS 的变化由硬件自动置位，置 1 时表示接收器已准备好进行通信，置 0 时表示接收端不能接收。USART_CR3 寄存器中的 CTSIE 位置 1 时会触发中断。CTS 流控制使能时的通信示例参见下图：

图 29-23 CTS 流控示例


注意：为保证正常工作，使能 CTS 需在当前字符结束前至少 3 个 USART 时钟周期内完成。此外还应注意，当 CTS 上的脉冲小于 2 个 PCLK 周期时无法将 CTSCF

置位。

RS485 收发器控制:

将 USART_CR3 控制寄存器中的 DEM 位置 1,使能 RS485 收发器控制功能。此功能下用户可使用 DE (Driver Enable) 信号控制外部 RS485 收发器。

USART_CR1 寄存器的 DEAT [4:0] 位域用于编程 DE 信号的建立时间: 从使能 DE (DE 拉高) 到发送起始位之间的时间间隔。

USART_CR1 寄存器的 DEDT [4:0] 位域用于编程 DE 信号的保持时间: 从最后一帧的停止位, 到停用 DE 拉低之间的时间。

DE 信号的极性在 USART_CR3 寄存器中的 DEP 位可配置。DEAT 和 DEDT 的单位为采样时钟周期, 根据过采样的设置, 可以为 1bit 时间的 1/8 或 1/16。

29.5.18 USART 低功耗管理

USART 支持在低功耗模式下工作,即使在停用 USART_PCLK 时钟时也可以保证数据正常传输。

在 USART 时钟源使用 HSI 或 LSE 时, 将 UEWK 位置 1, 满足唤醒条件时可以将 MCU 从停止模式唤醒。

如果 USART_PCLK 停止, 但有些操作需使用 USART_PCLK 时钟, 可以由 USART 的唤醒信号 USART_WAKEUP 将 USART_PCLK 重新使能, 以便接收 RXFIFO 或填充 TXFIFO。

- 禁用 FIFO 时, 需要使能 USART_PCLK 后才可以读取 USART 接收数据寄存器。这种情况下 USART_WAKEUP 信号的触发条件为 RXNE 位置 1, 即接收到数据时产生唤醒信号。RXNEIE 位需在进入低功耗模式前置 1,
- 使能 FIFO 时, 此时以下事件可以作为 USART_WAKEUP 信号的触发条件:
 - RxFIFO 已满。RXFFIE 位需要在进入低功耗模式前置 1, 在低功耗模式下可以接收的数据个数与 RXFIFO 总深度相同, 接收至 RXFIFO 满时, RXFF 标志不置位。
 - RXFIFO 非空。RXFNEIE 位需在进入低功耗模式前置 1。
 - TXFIFO 为空。TXFNFIE 位需在进入低功耗模式之前置 1。

低功耗模式下收发数据, 为避免上溢/下溢错误, USART_WAKEUP 信号的触发条件可以设置为以下事件:

- TXFIFO 达到阈值。TXFTIE 位需在进入低功耗模式前置 1。
- RXFIFO 达到阈值。RXFTIE 位需在进入低功耗模式前置 1。

使用 FIFO 相关的中断 (RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 达阈值) 唤醒 MCU 可以尽可能延长 USART 在低功耗

模式下的运行时间，利于优化功耗。

- 除了以上提到的中断以外，WUF 位置 1 也可以产生 USART_WAKEUP 信号，WUF 的置 1 的条件在 WUS[1:0] 位域配置，有以下几个选项：
 - 地址匹配时
 - 检测到起始位时
 - RXNE 置 1 时

如果 WUFIE 位置 1，则触发中断。只有将 WUFIE 位置 1 时，WUS 标志置位才可以将 MCU 从低功耗模式唤醒。

可以将处理器从低功耗模式唤醒。

注意： 在进入低功耗模式之前需确认 USART 没有正在进行的传输。仅检查 BUSY 标志不能保证没有正在进行的数据接收。

无论 MCU 处于低功耗模式还是正常工作模式，WUF 标志在检测到唤醒事件时都会置位。

初始化完成并使能接收后，在进入低功耗模式前，需检查 REACK 位以确认 USART 接收已使能。

使用 DMA 接收时，需在进入低功耗模式前停用 DMA，并在退出低功耗模式后重新使能。

使能 FIFO，仅在进入静默模式时才支持由地址匹配触发低功耗唤醒功能。

低功耗模式和静默模式

如果 USART 在进入低功耗模式前处于静默模式：

- 由于空闲帧检测无法在低功耗模式下工作，所以空闲帧检测不能用于从静默模式唤醒。
- 如果静默模式的唤醒源配置为地址匹配，则低功耗模式的唤醒源也需同样配置为地址匹配。在进入静默模式前 RXNE 需清零，如果在进入低功耗模式时 RXNE 标志置 1，地址匹配发生时无法从静默模式退出，但是地址匹配仍然可以将 MCU 从低功耗模式下唤醒，只不过唤醒后无法从静默模式退出。

注意： 使能 FIFO 时，静默模式可与从低功耗模式唤醒灵活搭配，无限制条件，上述两点约束条件不适用于 FIFO 使能时。

USART 内核时钟 (USART_KCLK) 停止时从 Stop 模式唤醒

如果在 Stop 模式下 USART_KCLK 时钟停止，在 USART 接收线上检测到下降沿时，借助 USART_KCLK_REQ 请求信号开启 USART_KCLK 时钟。USART_KCLK 随后用于接收。

此后如果检测到了有效的唤醒事件，MCU 从 Stop 模式唤醒并正常接收数据。

如果唤醒事件无效，则 USART_KCLK 再次停止，MCU 不被唤醒，保持在 Stop 模式下，内核时钟开启请求被释放。

以下示例展示了唤醒事件为“地址匹配检测”且不使能 FIFO 的情况。

下图所示为唤醒事件有效时的 USART 行为。

图 29-24 检测到有效唤醒事件

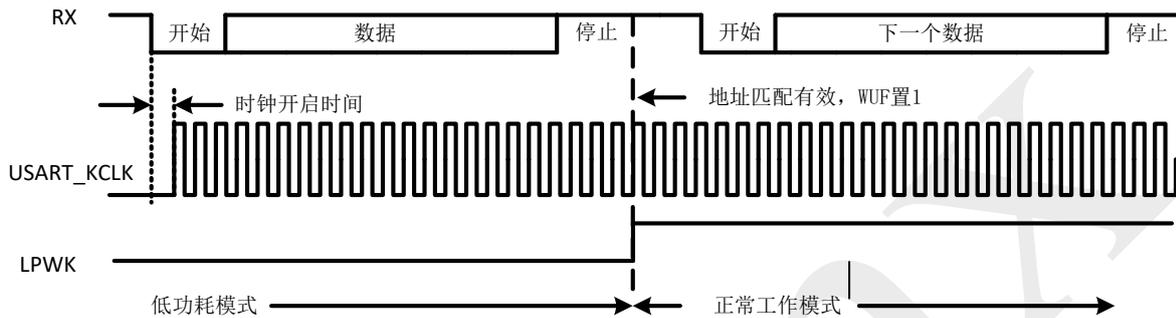
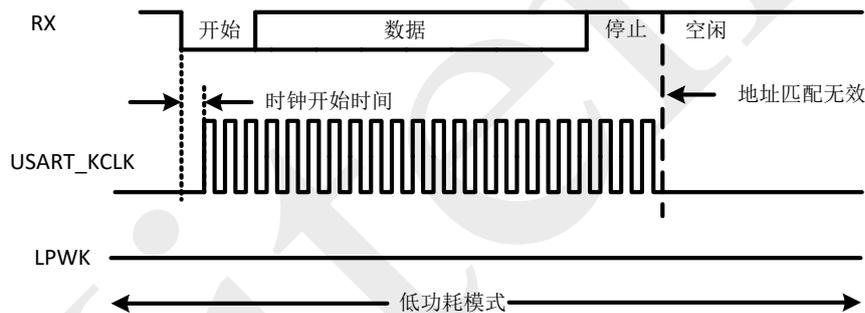


图 29-25 未检测到有效唤醒事件



注意： 唤醒事件为地址匹配或任一接收帧时以上两图适用。

唤醒事件为起始位检测时 USART 在起始位结束时间向 MCU 发送唤醒事件信号。

确定从低功耗模式唤醒 MCU 的最大 USART 波特率

从低功耗模式正确唤醒微控制器的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 USART 接收器容差（请参见 [USART 接收器对时钟偏差的容差章节](#)）。

举例来说：OVER8=0，M 位 = “01”，ONEBIT=0，BRR [3:0]=0000。（16 倍过采样，9bits 字符长度，3 次采样表决）。

在这些条件下，根据 [BRR=0 时的 USART 接收容差](#)：BRR [3:0] = 0000 时的 USART 接收器容差为 3.41%。

$DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR + DWKUP < USART$ 接收器的容差

$$DWKUP_{max} = t_{WUUSART} / (11 \times T_{bit\ Min})$$

$$T_{\text{bit Min}} = t_{\text{WUUSART}} / (11 \times \text{DWKUPmax})$$

其中 t_{WUUSART} 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTR_ERR、DQU_ERR、DRX_OSERR 和 DL_ERR 为 0%，则 DWKUP 的最大值为 3.41%。实际上，计算时至少需要考虑 USART_KCLK 时钟源的精度。

例如，如果将 HSI 用作 USART_KCLK，HSI 的精确度为 1%：

$$t_{\text{WUUSART}} = 3\mu\text{s} \quad (\text{仅供参考，正确参数请参见数据手册})。$$

$$\text{DWKUPmax} = 3.41\% - 1\% = 2.41\%$$

$$T_{\text{bit Min}} = 3\mu\text{s} / (11 \times 2.41\%) = 11.32 \mu\text{s}$$

得到从低功耗唤醒时支持的最大波特率为： $1/11.32\mu\text{s} = 88.36 \text{ Kbauds}$

29.5.19 USART 中断请求表

USART 可以产生两种类型的中断信号：USART_IRQ 和 USART_WAKEUP。其中 USART_WAKEUP 信号可用于低功耗唤醒。如果唤醒时使能 USART_IRQ，USART_IRQ 由 NVIC 管理，MCU 在唤醒后响应中断。

表 29-8 USART 中断请求表

中断标志	触发事件	使能控制位	中断清除方法
TXE	发送数据寄存器为空	TXEIE	TXE 在 TDR 中被写入数据时清零。
TXFNF	发送 FIFO 未滿	TXFNFIE	TXFNF 在 TXFIFO 已滿时清零。
TXFE	发送 FIFO 为空	TXFEIE	TXFNF 在 TXFIFO 包含至少一个数据时自动清零。
TXFT	达到发送 FIFO 阈值	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。
CTSIF	CTSIF 由软件通过将 CTSCF 位置“1”来清零。	CTSIE	CTSIF 由软件通过将 CTSCF 位置“1”来清零。
TC	TC 在 TDR 中被写入数据时清零，或通过 TCCF 位置“1”来清零。	TCIE	TC 在 TDR 中被写入数据时清零，或通过 TCCF 位置“1”来清零。
RXNE	RXNE 通过读取 RDR 或通过 RXFRQ 位置“1”来清零。	RXNEIE	RXNE 通过读取 RDR 或通过 RXFRQ 位置“1”来清零。
RXFNE	RXFNE 在 RXFIFO 为空时清零，或通过 RXFRQ 位置“1”来清零。	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过 RXFRQ 位置“1”来清零。
RXFF	RXFF 在 RXFIFO 至少包含一个数据时清零。	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。
RXFT	达到接收 FIFO 阈值	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。

中断标志	触发事件	使能控制位	中断清除方法
ORE	检测到溢出错误	RXNEIE/ RXFNEIE	ORE 通过将 ORECF 位置“1”清零。
IDLE	总线空闲检测	IDLEIE	通过将 IDLECF 位的置位来清零 IDLE 标志。
PE	奇偶校验错误	PEIE	PE 通过将 PECF 位置“1”来清零。
NOISE 或 ORE 或 FE	多缓冲区通信中的噪声标志、 溢出错误和帧错误。	EIE	NOISE 通过将 NOISECF 位置“1”来清零。ORE 通过将 ORECF 位置“1”来清零。FE 标志通过将 FECF 位置“1”来清零。
CMF	地址字符匹配	CMIE	CMF 通过将 ACMCF 位置“1”来清零。
RTOF	接收器超时	RTOFIE	RTOF 通过将 RTOCCF 位置“1”来清零。
WUF	从低功耗模式唤醒	WUFIE	WUF 通过将 WUCF 位置“1”来清零。
UDR	SPI 从设备下溢错误	EIE	UDR 通过将 UDRCF 位置“1”来清零。

注意: RXFF 标志在 USART 收到 $n+1$ 个数据的时候置位, 即接收到 RXFIFO 深度 n 个数据加上 USART_RDR 中的 1 个数据。在 STOP 模式下, USART_RDR 时钟源已停, 所以不会有数据接收到 USART_RDR 中, 所以 RXFIFO 接收到和深度相同数量的数据时, 就会触发 RXFF 中断, 而 RXFF 标志位不会被置位。

29.6 USART 寄存器

每个寄存器为 32bits, 仅支持以字 (Word) 为单位访问。

29.6.1 USART 控制寄存器 1 (USART_CR1) -- 【FIFO 使能时】

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
RXFF IE	TXFE IE	FIFO EN	M1	Res.	RTO IE	DEAT[4:0]				DEDT[4:0]					
rw	rw	rw	rw		rw	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
OVER 8	CM IE	MME	M0	WAKE	PCE	PS	PEIE	TXFNF IE	TCIE	RXFNE IE	IDLE IE	TE	RE	UE SM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	RXFFIE	RXFIFO 已满中断使能 0: 禁止中断。 1: 当 USART_ISR 寄存器中的 RXFF=1 时, 触发中断。

30	TXFEIE	TXFIFO为空中断使能 0: 禁止中断。 1: 当USART_ISR寄存器中的TXFE=1时, 触发中断。
29	FIFOEN	FIFO使能 0: 禁用FIFO。 1: 使能FIFO。 在禁止USART(UE=0)时才能写入此位域。 <i>注意: FIFO模式支持在标准UART通信、同步主/从模式模式下使用, 不支持IrDA模式。</i>
28	M1	字符长度 此位和bit 12(M0)用于配置字符长度, 由软件置1或清零。 M[1:0] = 00: 1个起始位, 8个数据位, n个停止位 M[1:0] = 01: 1个起始位, 9个数据位, n个停止位 M[1:0] = 10: 1个起始位, 7个数据位, n个停止位 在禁用USART(UE=0)时才能写入此位。 <i>注意: 自动波特率检测只支持8位或9位字符长度。</i>
27	保留	读为0且必须保持为复位值。
26	RTOIE	接收超时中断使能 0: 禁止中断 1: USARTx_ISR寄存器中的RTOF位置1时, 触发中断。 <i>注意: USART不支持接收器超时功能时, 此位保留并由硬件强制清零。</i>
25: 21	DEAT[4:0]	RS485模式下, DE信号的建立时间 用于配置RS485收发器使能信号DE拉高与发送起始位之间的间隔时间, 单位为采样时钟周期, 根据过采样设置, 可为1/8bit或1/16bit时间。 在禁用USART(UE=0)时才能写入此位域。 <i>注意: 如果不支持RS485收发器控制功能, 此位保留, 读为0写入无效。</i>
20: 16	DEDT[4:0]	RS485模式下, 禁止驱动器时DE信号的保持时间 从最后一帧的停止位到DE拉低之间的时间。单位为采样时钟周期, 根据过采样设置, 可以是1/8或1/16位的时间。 如果在DEDT期间内在USARTx_TDR寄存器写入新数据, 则新数据仅在经过DEDT和DEAT后才会发送。 在禁用USART(UE=0)后才能写入此位域。 <i>注意: 如果不支持RS485收发器控制功能, 此位保留, 读为0写入无效。</i>

15	OVER8	<p>过采样模式，</p> <p>0: 16倍过采样</p> <p>1: 8倍过采样</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意：在IrDA模式下，此位需保持清零。</i></p>
14	CMIE	<p>字符匹配中断使能</p> <p>0: 禁止中断</p> <p>1: USARTx_ISR寄存器中的CMF位置1时，触发中断。</p>
13	MME	<p>静默模式使能</p> <p>用于使能USART进入静默模式，此位置1时，USART按WAKE位配置的方式在正常模式与静默模式之间切换。</p> <p>0: 接收器仅工作在正常工作模式。</p> <p>1: 允许接收器在静默模式和正常工作模式间切换。</p>
12	M0	<p>字符长度</p> <p>此位和位28(M1)用于设置字符长度。详情参见M1的说明。</p> <p>在禁用USART(UE=0)时才能写入此位。</p>
11	WAKE	<p>静默模式唤醒方式选择</p> <p>此位用于选择USART退出静默模式的方式</p> <p>0: 空闲帧检测</p> <p>1: 地址匹配</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
10	PCE	<p>奇偶校验控制使能</p> <p>此位选择硬件奇偶校验控制，包括发送时生成校验位和接收时对校验位的检测。使能奇偶校验时，发送数据中奇偶校验位插入到MSB位置（参见：字符组成示意图），并对接收到的数据检查奇偶校验位。</p> <p>此位置1后，奇偶校验控制从下一字节生效。</p> <p>0: 禁用奇偶校验</p> <p>1: 使能奇偶校验</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
9	PS	<p>奇偶校验选择</p> <p>此位用于在奇偶校验使能时（PCE位置1）时，选择奇校验或偶校验。</p> <p>此位选择后从下一字节开始生效。</p> <p>0: 偶校验</p> <p>1: 奇校验</p> <p>在禁用USART(UE=0)后才能写入此位。</p>
8	PEIE	<p>奇偶校验错误中断使能</p>

		0: 禁止中断 1: 当USARTx_ISR寄存器中PE=1时， 触发中断。
7	TXFNFIE	TXFIFO非空中断使能 0: 禁止中断 1: 当USARTx_ISR寄存器中TXFNF=1时， 触发中断
6	TCIE	传输完成中断使能 0: 禁止中断 1: 当USART_ISR寄存器中的TC=1时， 触发中断
5	RXFNEIE	RXFIFO非空中断使能 0: 禁止中断 1: 当USARTx_ISR寄存器中的ORE=1或RXFNE=1时， 触发中断
4	IDLEIE	空闲帧检测中断使能 0: 禁止中断 1: 当 USARTx_ISR 寄存器中的 IDLE=1 时， 触发 USART 中断
3	TE	发送器使能 0: 禁用发送器 1: 使能发送器 <i>注意: 传送期间TE 位上的“0”脉冲, 即由“L”变为“H”时, 会在当前字节发送完毕后发送一个空闲帧。为了正确发送空闲帧, TE 置 0 后需要持续一段时间才可以再写入 1, 可由软件查询 USARTx_ISR 寄存器中的 TEACK 位确认有效间隔时间。</i>
2	RE	接收器使能 0: 禁止接收器 1: 使能接收器
1	UESM	使能 USART 将 MCU 从低功耗模式唤醒 当此位清零时, USART 无法将 MCU 从低功耗模式(停止模式)唤醒。 当此位置 1 时, USART 可以将 MCU 从低功耗模式(停止模式)唤醒, 需要配置 USART_KCLK 为 FIRC 的二分频或 LSE, 时钟源在 RCC_CCIPR 寄存器中可配置。 0: USART 不能将 MCU 从低功耗醒。 1: USART 能够将 MCU 从低功耗模式唤醒。 <i>注意: 建议在进入停止模式前将 UESM 位置 1, 并在退出停止模式时将其清零。</i> <i>如果 USART 不支持从停止模式唤醒功能, 则此位保留并由硬件强制清零。</i>

0

UE

USART 使能

此位清零后,USART 预分频器立即停止,当前工作丢弃。USART 配置被保留,但 USART_ISR 的所有状态标志均被复位。

0: 禁止 USART

1: 使能 USART

注意: 禁用 USART 之前需要将 TE 位清零, 为避免传输错误, 需检查 USART_ISR 中的 TC 位置 1, 确认传输完成后再将 TE 位清零。

UE = 0 时会同时复位 DMA 请求, 因此复位 UE 位前还需禁止 DMA 通道。

29.6.2 USART 控制寄存器 1 (USART_CR1) -- 【FIFO 禁用】

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
Res.		FIFO EN	M1	Res.	RTO IE	DEAT[4:0]					DEDT[4:0]				
		rw	rw		rw	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
OVER 8	CM IE	MME	M0	WAKE	PCE	PS	PEIE	TXE IE	TCIE	RXNE IE	IDLE IE	TE	RE	UE SM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域
名称
描述

31: 30

保留

必须保持复位值。

29

FIFOEN

FIFO使能

0: 禁用FIFO

1: 使能FIFO

只有在USART未使能(UE=0)时才能写入此位。

注意: FIFO模式只能在标准UART通信、同步(SPI)主/从设备模式模式下使用, 不支持在IrDA模式下使能。

28

M1

字符长度,

此位和bit 12(M0)用于配置字符长度。

M[1:0] = 00: 1个起始位, 8个数据位, n个停止位

M[1:0] = 01: 1个起始位, 9个数据位, n个停止位

M[1:0] = 10: 1个起始位, 7个数据位, n个停止位

在禁用USART(UE=0)时才能写入此位。

注意: 自动波特率检测只支持8位, 9位字符长度

27

保留

读为0且必须保持为复位值。

26

RTOIE

接收器超时中断使能

0: 禁止中断

1: USARTx_ISR寄存器中的RTOF位置1时, 触发中断。

注意: USART不支持接收器超时功能时, 此位保留并由硬件强

制清零。

25: 21	DEAT[4:0]	<p>RS485模式下，DE信号的建立时间</p> <p>用于配置RS485收发器使能信号DE拉高与发送起始位之间的间隔时间，单位为采样时钟周期，根据过采样设置，可为1/8bit或1/16bit时间。</p> <p>在禁用USART(UE=0)时才能写入此位域。</p> <p><i>注意：如果不支持RS485驱动器使能功能，此位保留，读为0写入无效。</i></p>
20: 16	DEDT[4:0]	<p>RS485模式下，禁止驱动器时DE信号的保持时间</p> <p>从最后一帧的停止位到DE拉低之间的时间。单位为采样时钟周期，根据过采样设置，可以是1/8或1/16位的时间。</p> <p>如果在DEDT期间内在USARTx_TDR寄存器写入新数据，则新数据仅在经过DEDT和DEAT后才会发送。</p> <p>在禁用USART(UE=0)后才能写入此位域。</p> <p><i>注意：如果不支持RS485使能功能，此位保留，读为0写入无效。</i></p>
15	OVER8	<p>过采样模式</p> <p>0: 16倍过采样</p> <p>1: 8倍过采样</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意：在IrDA模式下，此位需保持清零。</i></p>
14	CMIE	<p>字符匹配中断使能，</p> <p>0: 禁止中断</p> <p>1: USARTx_ISR寄存器中的CMF位置1时，触发中断。</p>
13	MME	<p>静默模式使能，</p> <p>用于使能USART进入静默模式，此位置1时，USART按WAKE位配置的方式在正常模式与静默模式之间切换。</p> <p>0: 接收器仅工作在正常工作模式。</p> <p>1: 允许接收器在静默模式和正常工作模式间切换。</p>
12	M0	<p>字符长度</p> <p>此位和位28(M1)用于确定字符长度。</p> <p>详情请参见位28(M1)的说明。</p> <p>在禁用USART(UE=0)时才能写入此位。</p>
11	WAKE	<p>从静默模式退出方式</p> <p>此位用于选择USART从静默模式退出的方式</p> <p>0: 空闲帧检测</p> <p>1: 地址匹配</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
10	PCE	<p>奇偶校验控制使能</p> <p>此位使能硬件奇偶校验控制，包括发送时生成校验位和接收时</p>

		对校验位的检测。使能奇偶校验时，发送数据中奇偶校验位插入到MSB（参见： 字符组成示意图 ），并对接收到的数据检查奇偶校验位。 此位置1后，奇偶校验从下一字节生效。 0：禁用奇偶校验 1：使能奇偶校验 在禁用USART(UE=0)后才能写入此位域。
9	PS	奇偶校验选择 此位用于在奇偶校验使能时（PCE位置1）时，选择奇校验或偶校验。 此位选择后从下一字节开始生效。 0：偶校验 1：奇校验 在禁用USART(UE=0)后才能写入此位域。
8	PEIE	奇偶校验错误中断使能 0：禁止中断 1：当USARTx_ISR寄存器中PE=1时，触发中断。
7	TXEIE	发送数据寄存器空中断使能 0：禁止中断 1：当USARTx_ISR寄存器中TXE=1时，触发中断。
6	TCIE	传输完成中断使能 0：禁止中断 1：当USART_ISR寄存器中的TC=1时，触发中断。
5	RXNEIE	接收数据非空中断使能 0：禁止中断 1：当USARTx_ISR寄存器中的ORE=1或RXNE=1时，触发中断。
4	IDELIE	空闲帧检测中断使能，标志位为IDLE。 0：禁止中断 1：当USARTx_ISR寄存器中的IDLE=1时，触发中断
3	TE	发送器使能 0：禁用发送器 1：使能发送器 <i>注意：传送期间TE位上的“0”脉冲，即由“L”变为“H”时，会在当前字节发送完毕后发送一个空闲帧。为了正确发送空闲字符，TE置0后需要持续一段时间才可以写入1，可由软件轮询USARTx_ISR寄存器中的TEACK位的方式确认有效间隔时间。</i>
2	RE	接收器使能 该位使能接收器。

0: 禁止接收器
1: 使能接收器

- 1 UESM 使能 USART 将 MCU 从低功耗模式唤醒
 当此位清零时, USART 无法将 MCU 从低功耗模式(停止模式)唤醒。
 当此位置 1 时, USART 可以将 MCU 从低功耗模式(停止模式)唤醒, 需要配置 USART_KCLKFIRC 的二分频或 LSE, 时钟源在 RCC_CCIPR 寄存器中可配置。
 0: USART 不能将 MCU 从低功耗醒。
 1: USART 能够将 MCU 从低功耗模式唤醒。
注意: 建议在进入停止模式前将 UESM 位置 1, 并在退出停止模式时将其清零。停止模式
如果 USART 不支持从唤醒功能, 则此位保留并由硬件强制清零。
- 0 UE USART 使能
 此位清零后, USART 预分频器立即停止, 当前工作丢弃。USART 配置被保留, 但 USART_ISR 的所有状态标志均被复位。
 0: 禁止 USART
 1: 使能 USART
注意: 禁用 USART 之前需要将 TE 位清零, 为避免传输错误, 需检查 USART_ISR 中的 TC 位置 1, 确认传输完成后再将 TE 位清零。
UE = 0 时会同时复位 DMA 请求, 因此复位 UE 位前还需禁止 DMA 通道。

29.6.3 USART 控制寄存器 2 (USART_CR2)

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
ADD[7:0]								RTO EN	ABRMOD[1:0]		ABR EN	MSB FIRST	DATA INV	TXIN V	RXIN V
rw	rw	rw	rw	rw	rw	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
SWAP	Res.	STOP[1:0]		CLK EN	CPOL	CPHA	LBCL	Res.			ADD M7	DIS_NSS	Res.		SLV EN
rw		rw	rw	rw	Rw	rw	rw				rw	rw			rw

位/位域	名称	描述
31:28	ADD[7:4]	USART本地节点地址 此位域用于配置USART本地地址或需要检测的字符。 在多机通信时, 用于通过地址匹配, 可将MCU从静默模式或低功耗模式下唤醒。地址字符的MSB为1。 此位域还可用于字符检测 (例如, ModBus协议中的块结束检

		测)。这种情况下，接收到的整个字符（8bits）将与ADD[7:0]的值进行比较，如果一致则CMF置1。 仅在禁止接收(RXE=0)或停用USART(UE=0)时才能写入此位域。
27:24	ADD[3:0]	USART本地节点地址 此位域用于指定USART本地地址或要识别的字符。 在多机通信时，用于通过地址匹配将MCU从静默模式或低功耗模式下唤醒。 仅在禁止接收(RE = 0)或禁止USART(UE=0)时才能写入该位域。 ADD[7:0]配合USART_CR2中的ADDM7位的设置，实现地址匹配模式。 ADDM7: 地址匹配模式选择 ADDM7为0: 地址长度为4bits，保存在ADD[3:0]; ADDM7为1: 7bits字符长度时，地址长度为6bits，保存在ADD [5:0] ; 8bits字符长度时，地址长度为7bits，保存在ADD [6:0]; 9bits字符长度时，地址长度为8bits，保存在ADD [7:0];
23	RTOEN	接收超时使能 0: 禁止接收超时功能。 1: 使能接收超时功能。 使能接收超时后，如果RX线路在RTO（接收器超时寄存器）中编程的时间内无接收，持续处于空闲状态，则USARTx_ISR寄存器中的RTOF标志置1。 <i>注意：如果USART不支持接收器超时功能，该位保留并由硬件强制清零。</i>
22:21	ABRMOD[1:0]	自动波特率检测模式选择 00: 模式 0，检测起始位 01: 模式 1，检测“10xx”开头的字符 10: 模式 2，检测字符“0x7F” 11: 模式 2，检测字符“0x55” 仅在ABREN = 0时或禁止USART(UE=0)时才能写入此位域。 <i>注意：DATAINV和MSBFIRST的设置需要与线路约定一致。例如MSBFIRST=1时，0x55匹配的发送数据可能为0xAA，如果USART不支持自动波特率检测功能，则此位保留并由硬件强制清零。</i>
20	ABREN	自动波特率检测使能 0: 禁止自动波特率检测。 1: 使能自动波特率检测。 <i>注意：如果USART不支持自动波特率功能，则此位保留并由硬件强制清零。</i>

19	MSBFIRST	<p>MSB优先</p> <p>0: 发送/接收时LSB(bit0)优先。</p> <p>1: 发送/接收MSB优先。</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
18	DATAINV	<p>二进制数据极性反向</p> <p>0: 按正向逻辑处理发送/接收的数据。(1=H, 0=L)</p> <p>1: 按反向逻辑处理发送/接收数的数据。(1=L, 0=H)。奇偶校验位也取反。</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
17	TXINV	<p>TX引脚有效电平反向</p> <p>0: TX引脚使用标准逻辑电平 ($V_{DD} = 1, GND = 0$)</p> <p>1: TX引脚信号反向。($V_{DD} = 0, GND=1$)。</p> <p>在禁用USART(UE=0)后才能写入此位。</p>
16	RXINV	<p>RX引脚有效电平反向</p> <p>0: RX引脚信号使用标准逻辑电平 ($V_{DD} = 1, GND = 0$) 工作</p> <p>1: 对RX引脚信号反向。($V_{DD} = 0, GND = 1$)。</p> <p>在禁用USART(UE=0)后才能写入此位。</p>
15	SWAP	<p>TX/RX引脚交换</p> <p>0: 按标准引脚排列使用TX/RX引脚</p> <p>1: TX和RX引脚交换</p> <p>此功能可以方便的支持与其他USART交叉连接。</p> <p>在禁用USART(UE=0)后才能写入此位。</p>
14	保留	<p>必须保持复位值。</p>
13: 12	STOP[1:0]	<p>停止位</p> <p>此位域用于编程停止位。</p> <p>00: 1位停止位</p> <p>01: 0.5位停止位</p> <p>10: 2位停止位</p> <p>11: 1.5位停止位</p> <p>在禁用USART(UE=0)后才能写入此位域。</p>
11	CLKEN	<p>时钟使能</p> <p>使能CK引脚的输出。</p> <p>0: 禁用CK引脚的时钟输出</p> <p>1: 使能CK引脚的时钟输出</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意: 如果不支持同步模式, 则该位保留并由硬件强制清零。</i></p>

10	CPOL	<p>时钟极性</p> <p>此位用于选择在同步模式下CK引脚上时钟输出的极性</p> <p>0: 空闲时CK引脚为低电平</p> <p>1: 空闲时CK引脚为高电平</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意: 如果不支持同步模式, 此位保留并由硬件强制清零。</i></p>
9	CPHA	<p>时钟相位</p> <p>此位用于选择在同步模式下CK引脚上时钟输出的相位</p> <p>0: 第一边沿采样数据</p> <p>1: 第二边沿采样数据</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意: 如果不支持同步模式, 该位保留并由硬件强制清零。</i></p>
8	LBCL	<p>最后一位对应时钟脉冲</p> <p>此位用于在同步模式下选择是否在最后一个数据位(MSB)发送时钟脉冲。</p> <p>0: 最后一个数据位的时钟脉冲不输出</p> <p>1: 输出最后一个数据位的时钟脉冲</p> <p><i>注意: 最后一位为发送的第7个、第8个或第9个数据位, 具体取决于USARTx_CR1寄存器中M位域的配置。</i></p> <p><i>在禁用USART(UE=0)时才能写入此位。</i></p>
7: 5	保留	<p>必须保持复位值。</p>
4	ADDM7	<p>地址匹配模式选择</p> <p>0: 地址长度为4bits, 保存在ADD[3:0];</p> <p>1: 7bits字符长度时, 地址长度为6bits, 保存在ADD [5:0] ;</p> <p>8bits字符长度时, 地址长度为7bits, 保存在ADD [6:0];</p> <p>9bits字符长度时, 地址长度为8bits, 保存在ADD [7:0];</p> <p>在禁用USART(UE=0)时才能写入该位</p>
3	DIS_NSS	<p>当DIS_NSS位置1时, 忽略NSS引脚输入。</p> <p>0: 同步从设备片选取取决于 NSS 引脚输入</p> <p>1: 同步从设备一直处于被选中的状态, 并忽略 NSS 引脚输入。</p> <p><i>注意: 不支持同步模式时, 该位保留且需保持复位值。</i></p>
2: 1	保留	<p>必须保持复位值。</p>

- 0 SLVEN 同步从模式使能
 SLVEN位置1时，使能同步从模式。
 0：禁止从模式。
 1：使能从模式。
 注意：不支持同步模式时，该位保留且需保持复位值。

29.6.4 USART 控制寄存器 3 (USART_CR3)

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	
TXFTCFG[2:0]						RXFTIE		RXFTCFG[2:0]						Res.		TXFTIE		WUFIE		WUS[1:0]				Res.								
rw		rw		rw		rw		rw		rw		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	
DEP		DEM		DDRE		OVRDIS		ONEBIT		CTSIE		CTSE		RTSE		DMAT		DMAR		Res.		Res.		HDSE L		IRLP		IREN		EIE		
rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		Rw		rw		rw		rw		

位/位域	名称	描述
31:29	TXFTCFG[2:0]	发送FIFO阈值配置 000：发送FIFO未使用空间达到其深度的1/8 001：发送FIFO未使用空间达到其深度的1/4 010：发送FIFO未使用空间达到其深度的1/2 011：发送FIFO未使用空间达到其深度的3/4 100：发送FIFO未使用空间达到其深度的7/8 101：发送FIFO全部空间未使用 其它：保留
28	RXFTIE	接收FIFO阈值中断使能 0：禁止接收FIFO阈值中断 1：使能接收FIFO阈值中断
27: 25	RXFTCFG[2:0]	接收 FIFO 阈值配置 000：接收 FIFO 已使用空间达到其深度的 1/8 001：接收 FIFO 已使用空间达到其深度的 1/4 010：接收 FIFO 已使用空间达到其深度的 1/2 011：接收 FIFO 已使用空间达到其深度的 3/4 100：接收 FIFO 已使用空间达到其深度的 7/8 101：接收 FIFO 全部空间已使用 其它：保留
24	保留	读为0且必须保持为复位值。
23	TXFTIE	发送 FIFO 阈值中断使能

		0: 禁止发送 FIFO 阈值中断 1: 使能发送 FIFO 阈值中断
22	WUFIE	<p>从低功耗模式唤醒信号中断使能</p> <p>0: 当检测到唤醒事件, 即USARTx_ISR寄存器中的WUF=1时, 不触发USART中断</p> <p>1: 当检测到唤醒事件, 即USARTx_ISR寄存器中的WUF=1时, 触发中断</p> <p><i>注意: WUFIE需在进入低功耗模式前置1。</i></p> <p><i>WUF中断仅在低功耗模式下有效。</i></p> <p><i>如果USART不支持从低功耗模式唤醒功能, 则此位保留并由硬件强制清零</i></p>
21:20	WUS[1:0]	<p>从低功耗模式唤醒中断标志选择</p> <p>此位域用于选择将WUF位置位的事件, 选中的事件发生时将低功耗唤醒标志置位。</p> <p>00: 地址匹配时, WUF位置1 (按ADD[7:0]和ADDM7所定义)</p> <p>01: 保留</p> <p>10: 检测到起始位时, WUF位置1</p> <p>11: RXNE置1时, WUF位置1</p> <p>在禁用USART(UE=0)后才能写入此位域。</p> <p><i>注意: 如果USART不支持从停止模式唤醒功能, 则此位保留并由硬件强制清零。</i></p>
19:17	保留	读为0且必须保持为复位值。
16	保留	必须保持复位值
15	DEP	<p>RS485收发器使能信号极性选择</p> <p>0: DE信号高电平有效。</p> <p>1: DE信号低电平有效。</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意: 如果不支持RS485收发器使能功能, 该位保留并且必须保持清零。</i></p>
14	DEM	<p>RS485收发器使能控制</p> <p>0: 禁止DE功能。</p> <p>1: 使能DE功能。DE信号在RTS引脚上输出。</p> <p>在禁用USART(UE=0)时才能写入此位。</p> <p><i>注意: 如果不支持RS485收发器控制功能, 该位保留并且必须保持清零。</i></p>
13	DDRE	<p>接收出错时禁止DMA</p> <p>0: 接收出错时不禁止DMA。相应的错误标志置1, 但RXNE保</p>

- 持为0，所以不产生DMA请求，同时避免了上溢，错误数据不会被接收，但是下一个正确数据可以被接收。
- 1：接收出错后禁止DMA。相应的错误标志以及RXNE均置1。屏蔽DMA请求，直到错误标志清零。这意味着软件必须首先禁止DMA请求(DMAR = 0)或者将RXNE清零，然后再将错误标志清零后才可以继续接收。
- 在禁用USART(UE=0)时才能写入此位。
- 注：接收错误包括：奇偶校验错误、帧错误或噪声错误。*
- 12 OVRDIS 禁止接收上溢
- 0：使能上溢检测，在有未读已接收的数据（即RXNE标志置1）时接收新数据，则上溢错误标志位ORE置1。
- 1：禁止上溢功能。如果在RXNE标志置1时接收到新数据，ORE标志不会置1，且新接收的数据会覆盖USARTx_RDR寄存器的内容。
- 在禁用USART(UE=0)时才能写入此位。
- 11 ONEBIT 采样方式选择
- 使用单次采样时，噪声检测标志(NOISE)无效。
- 0：对每个bit进行三次采样，使用多数表决的方法确认采样值
- 1：对每个bit进行单次采样
- 在禁用USART(UE=0)时才能写入此位。
- 注意：ONEBIT功能仅适用于数据位采样，不适用于起始位采样。*
- 10 CTSIE CTS中断使能
- 0：禁止CTS中断
- 1：当USARTx_ISR寄存器中的CTSIF=1时，触发中断
- 注意：如果不支持硬件流控功能，该位保留并由硬件强制清零。*
- 9 CTSE CTS使能
- 0：禁止CTS硬件流控
- 1：使能CTS输入，仅当CTS输入有效（低电平）时才允许数据发送。如果在发送数据时CTS输入为高电平，则在完成当前发送后停止发送。如果使CTS拉高时数据已写入发送数据寄存器，则此数据将延迟发送，直到CTS重新有效时才开始发送。
- 在禁用USART(UE=0)时才能写入该位。
- 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。*
- 8 RTSE RTS使能
- 0：禁止RTS硬件流控制
- 1：使能RTS输出，仅当接收缓冲区中有空间时才会请求接收数据。如果USART不能继续接收数据，将RTS输出为1；可以接

		收数据时，RTS输出为0。 在禁用USART(UE=0)时才能写入此位。 <i>注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。</i>
7	DMAT	使能DMA发送 1: 使能DMA发送 0: 禁用DMA发送
6	DMAR	使能DMA接收 1: 使能DMA接收 0: 禁用DMA接收
5	保留	读为0且必须保持为复位值。
4	保留	读为0且必须保持为复位值。
3	HDSEL	使能单线半双工 0: 禁用单线半双工模式 1: 使能单线半双工模式 在禁用USART(UE=0)时才能写入此位
2	IRLP	IrDA低功耗模式使能 0: 使能IrDA正常模式 1: 使能IrDA使用低功耗模式 在禁用USART(UE=0)时才能写入此位。 <i>注意：如果不支持IrDA模式，该位保留并由硬件强制清零。</i>
1	IREN	IrDA模式使能 0: 禁用IrDA 1: 使能IrDA 在禁用USART(UE=0)时才能写入此位。 <i>注意：如果不支持IrDA模式，该位保留并由硬件强制清零。</i>
0	EIE	错误中断使能 如果发生帧错误、上溢错误或出现噪声标志（USARTx_ISR寄存器中FE = 1或ORE = 1或NE = 1）时，使能此位会触发中断。 0: 禁止中断 1: 相应错误标志置位时，触发中断

29.6.5 USART 波特率分频寄存器 (USART_BRR)

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
Res.																
BRR[15:0]																

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	BRR[15:4]	BRR[15:4] = USART_DIV[15:4] 详情参见: USART 波特率生成章节
3:0	BRR[3:0]	当 OVER8 = 0 时: BRR[3:0] = USART_DIV[3:0]。 当 OVER8 = 1 时: BRR[2:0] = USART_DIV[3:0] >>1 BRR[3]需保持为零。 详情参见: USART 波特率生成章节

29.6.6 USART 保护时间和预分频器寄存器 (USART_GTPR)

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
Res.																
Res.								PSC[7:0]								

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	保留	读为0且必须保持为复位值。
7:0	PSC[7:0]	预分频器值 在 IrDA 低功耗和 IrDA 正常模式下: PSC[7:0] 的值为 IrDA 正常模式及低功耗模式的波特率。 此位域用于编程预分频器, 对 USART_KCLK_PRE 进行分频以 以获得较低的功耗: 使用寄存器中给出的值 (8 个有效位) 对源时钟进行分频: 00000000: 保留-不编程此值 00000001: 时钟源 1 分频

00000010: 时钟源 2 分频
 00000011: 时钟源 3 分频
 ...
 11111111: 时钟源 255 分频

29.6.7 USART 超时及块传输长度寄存器 (USARTx_RTOR)

偏移地址: 0x14

复位值: 0x0000 0000

注意: 如果不支持接收器超时功能, 此寄存器保留并由硬件强制为“0x00000000”。
 详情请参考[USART 功能列表](#) 章节

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
rw	rw	rw	rw	rw	rw	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
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	BLEN	块长度 块长度计数器可以由以下两个条件中任意一个清零: 1, 接收器禁止时即 RE=0 <i>注意: 这个位域可在块接收开始后可编程, 编程为起始字段中 LEN 中的数据。每个接收到的块只能对此位域编程一次。</i>
23:0	RTO[23:0]	接收器超时时间 此位域用于提供接收器的超时时间 (以 1 数据位时间为单位)。在标准模式下, 如果在接收到最后一个字符后, 在 RTO 对应的时间内未检测到新的起始位, 则 RTOF 标志置 1。 <i>注意: 每个接收到的字符只能对此值编程一次。 可以实时写入 RTO。如果新值小于或等于当前计数器的值, RTOF 标志置 1。</i>

29.6.8 USART 请求寄存器 (USART_RQR)

偏移地址: 0x18

复位值: 0x0000 0000

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

Res.										TX FRQ	RX FRQ	MMR Q	SBKR Q	ABRR Q
										w	w	w	w	w

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFRQ	发送数据清空请求， 禁用 FIFO 时， 向此位写 1 会清空发送数据， TXE 标志置 1。 此操作会丢弃发送数据。 使能 FIFO 时， 向此位写入 1 会清空 TXFIFO， TXNF 标志置位。 清空 TXFIFO 功能在所有使能 FIFO 的模式下均适用。
3	RXFRQ	接收数据清空请求 向该位写1时将清空接收数据， 未被读取的数据会被丢弃， 可用于避免上溢。 禁用FIFO时， RXNE标志清零。 使能FIFO时， RXFIFO被清空， RXFNE标志清零
2	MMRQ	静默模式请求 向此位写 1， USART 进入静默模式， 并将 RWU 标志置 1。
1	SBKRQ	发送中断帧请求 向此位写入 1 可将 SBKF 标志置 1， 并在发送设备准备好后立即请求在线路上发送一个中断帧。 <i>注意： 如果需要在一系列数据后再插入中断帧， 其中由部分数据还没有发送完， 这种情况下， 软件应等到TXE标志置位后再将SBKRQ位置1。</i>
0	ABRRQ	波特率自适应请求 向此位写入 1 可复位 USARTx_ISR 寄存器中的 ABRF 标志， 并请求对下一个接收到的数据帧进行自动波特率测量。 <i>注意： 如果 USART 不支持自动波特率功能， 则此位保留并由硬件强制清零。</i>

29.6.9 USART 中断和状态寄存器 (USART_ISR) -- 【FIFO 使能】

偏移地址： 0x1C

复位值： 0x0XX0 00C0

禁用 FIFO 时， XX = 00

使能 FIFO 时， XX = 88

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				TXFT	RXFT	Res.	RXFF	TXFE	RE ACK.	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r		r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	Res.	RTOF	CTS	CTSIF	Res.	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
r	r	r		r	r	r		r	r	r	r	r	r	r	r

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	TXFT	TXFIFO 阈值标志 发送FIFO未使用空间大于或等于阈值时置1。如果USART_CR3寄存器中的TXFTIE位=1（位31），则会触发中断。 0: TXFIFO 未达到设置阈值 1: TXFIFO 已达到设置阈值
26	RXFT	RXFIFO 阈值标志 RXFIFO 已接收数据长度大于或等于阈值时置 1，否则由硬件清 0。 0: RXFIFO 未达到设置阈值 1: RXFIFO 已达设置阈值 <i>注意：当 RXFTCFG 阈值配置为“100”时，如果有 8 个数（RXFIFO 中有 7 个数据，USART_RDR 中有 1 个数据，则 RXFT 标志将置 1。因此，接收到的第 9 个数据不导致上溢错误。接收到第 10 个数据后会发生上溢错误。</i>
25	保留	此位保留且保持复位值
24	RXFF	RXFIFO 已满 当接收到 RXFIFO 大小+1 个数据时， （即 RXFIFO 大小+USART_RDR 寄存器中的 1 个数据），此位由硬件置 1。 如果 USART_CR1 寄存器中 RXFFIE 位置 1，则会触发中断。 0: RXFIFO 未滿。 1: RXFIFO 已滿。
23	TXFE	TXFIFO 空标志 当 TXFIFO 为空时，此位由硬件置 1。 当 TXFIFO 中至少由一个数据时，此标志自动清零。 也可以通过向 USART_RQR 寄存器中的 TXFRQ 写 1 将 TXFE 标志置 1。 如果中断使能，即 USART_CR1 寄存器中的 TXFEIE 位=1，则会触发中断。 0: TXFIFO 非空。 1: TXFIFO 为空。

22	REACK	<p>接收使能确认标志</p> <p>此位由硬件置 1/清零</p> <p>用于确认进入低功耗模式前 USART 是否已准备好接收数据。</p>
21	TEACK	<p>发送使能确认标志</p> <p>此位由硬件置 1/清零</p> <p>当发送器准备开始发送时，此位置 1。</p> <p>当通过向 TE 清零再置 1 来生成空闲帧请求时，此位用于指示发送空闲帧请求的 TE 脉冲的最小宽度，TE 清零后检查此位置 1 后，再将 TE 置 1，</p>
20	WUF	<p>从低功耗模式唤醒标志</p> <p>当检测到唤醒事件时，此位由硬件置 1。</p> <p>事件源在 WUS[1:0]位域中选择。</p> <p>若需清除标志位，需先读取接受寄存器 RDR 中的接受数据，再通过向 USARTx_ICR 寄存器中的 WUCF 写 1 清零。</p> <p>中断使能时，即 USARTx_CR3 寄存器中 WUFIE=1，则会触发中断。</p> <p><i>注意：当UEWK清零时，WUF标志也清零。</i></p> <p><i>如果USART不支持从停止模式唤醒功能，则此位保留由硬件强制清零。</i></p>
19	RWU	<p>静默模式状态指示位</p> <p>该位指示 USART 是否处于静默模式。</p> <p>当从静默模式退出时此位清零，继续处于静默时标志为 1。</p> <p>静默模式退出方式有地址匹配和空闲帧检测两种，在 USARTx_CR1 寄存器中的 WAKE 位可配置。</p> <p>此位由硬件清零/置 1。选择空闲检测即 IDLE 标志唤醒时，此位只能通过向 USARTx_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。</p> <p>0: 处于正常工作模式</p> <p>1: 处于静默模式</p>
18	SBKF	<p>中断帧发送标志</p> <p>此位指示已请求发送中断帧。</p> <p>向写入 USARTx_RQR 寄存器中的 SBKRQ 位写 1，即产生发送中断帧请求时，此位置 1。</p> <p>此位在中断帧的停止位期间由硬件自动复位。</p> <p>0: 无中断帧发送请求</p> <p>1: 有中断帧发送请求</p>

17	CMF	<p>字符匹配标志</p> <p>接收到由ADD[7:0]定义的字符时此位由硬件置1。通过向USARTx_ICR寄存器中的CMCF写1清零。</p> <p>如果中断使能，即USARTx_CR1寄存器中CMIE=1时，会触发中断。</p> <p>0: 未检测到字符匹配 1: 检测到字符匹配</p>
16	BUSY	<p>忙标志</p> <p>此位由硬件置1和复位。</p> <p>当RX线路有接收的数据，从检测到起始位开始，此位置1。在接收结束时，无论是否成功此位均复位。</p> <p>0: 当前没有接收数据，USART处于空闲状态 1: 正在接收数据</p>
15	ABRF	<p>自动波特率检测标志</p> <p>此位由硬件置1，置位条件有两种可能：</p> <p>1, 自动波特率检测成功，此时RXNE也置1，如果RXNEIE = 1则触发中断。</p> <p>2, 自动波特率已完成但是未成功，此时，ADBREN、RXNE和FE均置1。</p> <p>向USARTx_RQR寄存器中的ABRQ写1，发起新的自动波特率检测请求后，此位清零。</p> <p><i>注意：如果USART不支持自动波特率检测功能，则此位保留并由硬件强制清零。</i></p>
14	ABRE	<p>自动波特率检测错误标志</p> <p>如果自动波特率检测失败，例如，波特率超出检测范围或字符匹配失败等，此位由硬件置1。</p> <p>向USARTx_RQR寄存器中的ABRQ位写1，重新发起自动波特率检测请求，则此位清零。</p> <p><i>注意：如果USART不支持自动波特率功能，则此位保留并由硬件强制清零。</i></p>
13	UDR	<p>同步从设备下溢出错误标志</p> <p>在作为同步从设备发送时，如果在USART_TDR寄存器未写入数据前，出现第一个数据发送时钟脉冲，则此标志位置1。</p> <p>向USART_ICR中的UDRCF位写1，此位清零。</p> <p>0: 未检测到下溢出 1: 检测到下溢出</p> <p><i>注意：如果USART不支持自动波特率功能，则此位保留并由硬件强制清零。</i></p>

12	保留	此位保留且保持复位值
11	RTOF	<p>接收器超时标志</p> <p>接收器在等待在 RTO 中编程的超时时间期间，没有接收到任何数据时，此位由硬件置 1。</p> <p>向 USARTx_ICR 寄存器中的 RTOCF 写入 1，此位清零。</p> <p>如果接收超时中断使能，即 USARTx_CR1 寄存器中 RTOIE=1，时会触发中断。</p> <p>0: 未达到超时时间</p> <p>1: 接收超时</p> <p><i>注意：如果 2 个字符之间的间隔时间等于 RTOT 中的编程时间，则 RTOF 不置位。如果此间隔时间大于 (RTO 中编程时间+2) 个采样时间 (2/16 或 2/8，具体取决于过采样模式)，则 RTOF 标志置 1。即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。</i></p>
10	CTS	<p>CTS 标志</p> <p>此位由硬件置 1/复位。此位状态是对 CTS 输入引脚的状态取反</p> <p>0: CTS 线置 1</p> <p>1: CTS 线置 0</p> <p><i>注意：如果不支持硬件流控，此位保留并由硬件强制清零。</i></p>
9	CTSIF	<p>CTS 中断标志</p> <p>在使能 CTS，即 CTSE 位置 1 时，当 CTS 输入出现电平翻转时，此位由硬件置 1。通过向 USARTx_ICR 寄存器中的 CTSCF 位写 1，此位清零。</p> <p>如果中断使能，即 USARTx_CR3 寄存器中 CTSIE=1 时，触发中断。</p> <p>0: CTS 状态线上未发生变化</p> <p>1: CTS 状态线上发生变化</p> <p><i>注意：如果不支持硬件流控，该位保留并由硬件强制清零。</i></p>
8	保留	必须保持复位值。
7	TXFNF	<p>TXFIFO 未滿</p> <p>使能 FIFO 模式时，TXFNF 会在 TXFIFO 未滿时由硬件置 1，表示可以向 USART_TDR 中写入数据。每次写 USART_TDR 寄存器都会将数据存入 TXFIFO 中。</p>

此标志位保持置 1 直到 TXFIFO 已满。当 TXFIFO 已满时此标志清零，表示不能向 USART_TDR 中写入数据。

如果中断使能，即 USART_CR1 寄存器中 TXFNFIE 位=1，则触发中断。

0: TXFIFO 已满。

1: TXFIFO 未滿。

注意：在清空发送数据请求 (TXFRQ) 期间，TXFNF 保持为 0，直到 TXFIFO 被清空后置 1。在写入 TXFIFO 之前需先检查 TXFNF。清空发送数据请求 (TXFRQ) 后，TXFNF 和 TXFE 同时置位。

6 TC

发送完成标志

此标志位表示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成一帧中所有数据的发送，且 TXE/TXFE 置 1，则此位由硬件置 1。

如果中断使能即，USART_CR1 寄存器中 TCIE = 1，则会触发中断。

通过向 USART_ICR 寄存器中的 TCCF 写 1，或写入 USART_TDR 寄存器时，TC 位清零。

0: 传送未完成

1: 传送已完成

注意：如果 TE 位被置 0，且当前无正在进行的发送正在，TC 位会立即置 1。

5 RXFNE

RXFIFO 非空

RXFIFO 非空时，RXFNE 位由硬件置 1，表示已接收到数据。每次读 USART_RDR 都会从 RXFIFO 中释放一个位置。

RXFNE 在 RXFIFO 为空时清零。

也可以通过向 USART_RQR 寄存器中的 RXFRQ 位写 1，通过清空接收数据的方式将 RXFNE 标志清零。

如果 USART_CR1 寄存器中 RXFNEIE=1，则会触发中断。

0: 接收 FIFO 为空，未接收到数据

1: 接收 FIFO 非空，已接收到数据

4 IDLE

检测到空闲帧

检测到空闲线路时，此位由硬件置 1。如果中断使能，即 USARTx_CR1 寄存器中 IDLEIE=1，则会触发中断。向 USARTx_ICR 寄存器中的 IDLECF 写 1，位清零。

0: 未检测到空闲帧

1: 检测到空闲帧

注意：只有接收到数据后再检测到新的空闲状态时，即RXFNE位置1后，IDLE位才会被再次置1。

使能静默模式(MME=1)后，如果USART未处于静默状态(RWU=0)，则IDLE可以置1，如果处于静默模式(RWU=1)，IDLE不置1。

3 ORE

溢出错误标志

在 $RXNE = 1$ 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 RDR 寄存器时，此位由硬件置 1。通过向 USARTx_ICR 寄存器中的 ORECF 写入 1，此位清零。

如果中断使能 USARTx_CR1 寄存器中 $RXNEIE=1$ 或 $EIE=1$ ，则会触发中断。

0: 无溢出错误

1: 检测到溢出错误

注意：当此位置1时，RDR寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE位置1后，

如果在多缓冲区通信中ORE标志置1，则会触发中断。

USARTx_CR3寄存器中的OVRDIS位置1时，此位将被永久强制清零（无上溢检测）

2 NE

噪声检测标志

当接收中检测到噪声时，此位由硬件置 1。

向 USARTx_ICR 寄存器中的 NECF 写入 1，此位清零。

0: 未检测到噪声

1: 检测到噪声

注意：禁用FIFO时，此位与RXNE位同时置位，由RXNE产生中断，NOISE不再产生中断。

EIE位置1后，如果在DMA通信时NOISE标志置1，则会触发中断。

当线路无噪声时，可以配置ONEBIT=1，使用单次采样的方式提高USART的接收容差，并禁止噪声检测，无NOISE标志输出。

1 FE

帧错误标志

当检测到未同步信号、噪声过大或中断帧时，此位由硬件置 1。

向 USARTx_ICR 寄存器中的 FECF 写 1，此位清零。

如果 USARTx_CR1 寄存器中 $EIE = 1$ ，则会触发中断。

0: 未检测到帧错误

1: 检测到帧错误或中断帧

0 PE

奇偶校验错误标志

当在接收器模式下发生奇偶校验错误时，此位由硬件置 1。向

USARTx_ICR 寄存器中的 PECF 写入 1，此位清零。
 如果 USARTx_CR1 寄存器中 PEIE = 1，则会触发中断。
 0: 无奇偶校验错误
 1: 奇偶校验错误

29.6.10 USART 中断和状态寄存器 (USART_ISR) -- 【FIFO 未使能】

偏移地址: 0x1C

复位值: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	Res.	RTOF	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r		r	r	r		r	r	r	r	r	r	r	r

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	保留	此位保留且保持复位值
24:23	保留	必须保持复位值。
22	REACK	接收使能确认标志 此位由硬件置 1/清零 用于确认进入低功耗模式前 USART 是否已准备好接收数据。
21	TEACK	发送使能确认标志 此位由硬件置 1/清零 当发送器准备开始发送时，此位置 1。 当通过向 TE 清零再置 1 来生成空闲帧请求时，此位用于指示发送空闲帧请求的 TE 脉冲的最小宽度，TE 清零后检查此位置 1 后，再将 TE 置 1。
20	WUF	从低功耗模式唤醒标志 当检测到唤醒事件时，此位由硬件置 1。 事件源在 WUS[1:0]位域中选择。 此位通过向 USARTx_ICR 寄存器中的 WUCF 写 1 清零。 中断使能时，即 USARTx_CR3 寄存器中 WUFIE=1，则会触发中断。 <i>注意：当UESM清零时，WUF标志也清零。</i> <i>如果USART不支持从停止模式唤醒功能，则此位保留由</i>

硬件强制清零。

19	RWU	<p>静默模式状态指示位</p> <p>该位指示 USART 是否处于静默模式。</p> <p>当从静默模式退出时此位清零，继续处于静默时标志为 1。</p> <p>静默模式退出方式有地址匹配和空闲帧检测两种，在 USARTx_CR1 寄存器中的 WAKE 位可配置。</p> <p>此位由硬件清零/置 1。选择空闲检测即 IDLE 标志唤醒时，此位只能通过向 USARTx_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。</p> <p>0: 处于正常工作模式</p> <p>1: 处于静默模式</p>
18	SBKF	<p>中断帧发送标志</p> <p>此位指示已请求发送中断帧。</p> <p>向写入 USARTx_RQR 寄存器中的 SBKRQ 位写 1，即产生发送中断帧请求时，此位置 1。</p> <p>此位在中断帧的停止位期间由硬件自动复位。</p> <p>0: 无中断帧发送请求</p> <p>1: 有中断帧发送请求</p>
17	CMF	<p>字符匹配标志</p> <p>接收到由 ADD[7:0] 定义的字符时此位由硬件置 1。通过向 USARTx_ICR 寄存器中的 CMCF 写 1 清零。</p> <p>如果中断使能，即 USARTx_CR1 寄存器中 CMIE=1 时，会触发中断。</p> <p>0: 未检测到字符匹配</p> <p>1: 检测到字符匹配</p>
16	BUSY	<p>忙标志</p> <p>此位由硬件置 1 和复位。</p> <p>当 RX 线路有接收的数据，从检测到起始位开始，此位置 1。在接收结束时，无论是否成功此位均复位。</p> <p>0: 当前没有接收数据，USART 处于空闲状态</p> <p>1: 正在接收数据</p>
15	ABRF	<p>自动波特率检测标志</p> <p>此位由硬件置 1，置位条件有两种可能：</p> <p>1, 自动波特率检测成功，此时 RXNE 也置 1，如果 RXNEIE = 1 则触发中断。</p> <p>2, 自动波特率已完成但是未成功，此时，ADBREN、RXNE 和 FE 均置 1。</p>

		向 USARTx_RQR 寄存器中的 ABRRQ 写 1，发起新的自动波特率检测请求后，此位清零。 <i>注意：如果 USART 不支持自动波特率检测功能，则此位保留并由硬件强制清零。</i>
14	ABRE	自动波特率检测错误标志 如果自动波特率检测失败，例如，波特率超出检测范围或字符匹配失败等，此位由硬件置 1。 向 USARTx_RQR 寄存器中的 ABRRQ 位写 1，重新发起自动波特率检测请求，则此位清零。 <i>注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。</i>
13	UDR	同步从设备下溢出错误标志 在作为同步从设备发送时，如果在 USART_TDR 寄存器未写入数据前，出现第一个数据发送时钟脉冲，则此标志位置 1。 向 USART_ICR 中的 UDRCF 位写 1，此位清零。 0：未检测到下溢出 1：检测到下溢出 <i>注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。</i>
12	保留	此位保留且保持复位值
11	RTOF	接收器超时标志 接收器在等待在 RTO 中编程的超时时间期间，没有接收到任何数据时，此位由硬件置 1。 向 USARTx_ICR 寄存器中的 RTOCF 写入 1，此位清零。 如果接收超时中断使能，即 USARTx_CR1 寄存器中 RTOIE=1，时会触发中断。 0：未达到超时时间 1：接收超时 <i>注意：如果 2 个字符之间的间隔时间等于 RTOT 中的编程时间，则 RTOF 不置位。如果此间隔时间大于 (RTO 中编程时间+2) 个采样时间 (2/16 或 2/8，具体取决于过采样模式)，则 RTOF 标志置 1。即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。 如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。</i>
10	CTS	CTS 标志

		<p>此位由硬件置 1/复位。此位状态是对 CTS 输入引脚的状态取反</p> <p>0: CTS 线置 1</p> <p>1: CTS 线置 0</p> <p><i>注意: 如果不支持硬件流控, 此位保留并由硬件强制清零。</i></p>
9	CTSIF	<p>CTS 中断标志</p> <p>在使能 CTS, 即 CTSE 位置 1 时, 当 CTS 输入出现电平翻转时, 此位由硬件置 1。通过向 USARTx_ICR 寄存器中的 CTSCF 位写 1, 此位清零。</p> <p>如果中断使能, 即 USARTx_CR3 寄存器中 CTSIE=1 时, 触发中断。</p> <p>0: CTS 状态线上未发生变化</p> <p>1: CTS 状态线上发生变化</p> <p><i>注意: 如果不支持硬件流控, 该位保留并由硬件强制清零。</i></p>
8	保留	<p>必须保持复位值。</p>
7	TXE	<p>发送数据寄存器为空</p> <p>禁用 FIFO 时, 当 USART_TDR 寄存器的内容已传输到移位寄存器时, TXE 由硬件置 1。写入 USART_TDR 寄存器时, 此位清零。还可以向 USART_RQR 寄存器中的 TXFRQ 写 1, 通过清空发送数据的方式将 TXE 标志置 1。</p> <p>如果 USART_CR1 寄存器中 TXEIE/TXFNIE 位=1, 则会触发中断。</p> <p>0: 发送数据寄存器 USART_TDR 已满</p> <p>1: 发送数据寄存器 USART_TDR 未滿</p>
6	TC	<p>发送完成标志</p> <p>此标志位表示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。</p> <p>如果已完成所有数据的发送, 且 TXE/TXFE 为 1, 则此位由硬件置 1。</p> <p>如果中断使能, 即 USART_CR1 寄存器中 TCIE = 1, 则触发中断。</p> <p>通过向 USART_ICR 寄存器中的 TCCF 写入 1, 或写入 USART_TDR 寄存器时, TC 位清零。</p> <p>0: 发送未完成</p> <p>1: 发送已完成</p> <p><i>注意: 如果 TE 位被置 0 且当前无任何发送正在进行, TC 位会立即置 1。</i></p>
5	RXNE	<p>读取数据寄存器非空</p>

当接收移位寄存器的内容已传输到USART_RDR寄存器时，RXNE位由硬件置1。读USART_RDR寄存器时，此位清零。

也可以向USART_RQR寄存器中的RXFRQ位写1，通过清空接收数据的方式将RXNE标志位清零。

如果USART_CR1寄存器中RXNEIE=1，则会触发中断。

0: 未接收到数据

1: 已接收到数据

4 IDLE

检测到空闲帧

检测到空闲线路时，此位由硬件置1。如果中断使能，即USARTx_CR1寄存器中IDLEIE=1，则会触发中断。向USARTx_ICR寄存器中的IDLECF写1，位清零。

0: 未检测到空闲帧

1: 检测到空闲帧

注意：只有接收到数据后再检测到新的空闲状态时，即RXFNE位置1后，IDLE位才会被再次置1。

使能静默模式(MME=1)后，如果USART未处于静默状态(RWU=0)，则IDLE可以置1，如果处于静默模式(RWU=1)，IDLE不置1。

3 ORE

溢出错误标志

在RXNE=1的情况下，当移位寄存器中当前正在接收的数据准备好传输到RDR寄存器时，此位由硬件置1。通过向USARTx_ICR寄存器中的ORECF写入1，此位清零。

如果中断使能USARTx_CR1寄存器中RXNEIE=1或EIE=1，则会触发中断。

0: 无溢出错误

1: 检测到溢出错误

注意：当此位置1时，RDR寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE位置1后，

如果在多缓冲区通信中ORE标志置1，则会触发中断。

USARTx_CR3寄存器中的OVRDIS位置1时，此位将被永久强制清零（无上溢检测）

2 NE

噪声检测标志

当接收中检测到噪声时，此位由硬件置1。

向USARTx_ICR寄存器中的NECF写入1，此位清零。

0: 未检测到噪声

1: 检测到噪声

注意：禁用FIFO时，此位与RXNE位同时置位，由RXNE产生中断，NOISE不再产生中断。

EIE位置1后，如果在DMA通信时NOISE标志置1，则会触发中断。

当线路无噪声时，可以配置ONEBIT=1，使用单次采样的方式提高USART的接收容差，并禁止噪声检测，无NOISE标志输出。

- | | | |
|---|----|---|
| 1 | FE | <p>帧错误标志</p> <p>当检测到未同步信号、噪声过大或中断帧时，此位由硬件置1。向 USARTx_ICR 寄存器中的 FECF 写1，此位清零。</p> <p>如果 USARTx_CR1 寄存器中 EIE = 1，则会触发中断。</p> <p>0: 未检测到帧错误</p> <p>1: 检测到帧错误或中断帧</p> |
| 0 | PE | <p>奇偶校验错误标志</p> <p>当在接收器模式下发生奇偶校验错误时，此位由硬件置1。向 USARTx_ICR 寄存器中的 PECF 写入1，此位清零。</p> <p>如果 USARTx_CR1 寄存器中 PEIE = 1，则会触发中断。</p> <p>0: 无奇偶校验错误</p> <p>1: 奇偶校验错误</p> |

29.6.11 USART 中断标志清零寄存器(USART_ICR)

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	
											w			w		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	UDRCF	Res.	RTOCF	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF	
		w			w			w			w	w	w	w	w	

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUCF	从低功耗模式唤醒标志清零 向此位写1时，USARTx_ISR寄存器中WUF标志清零。 <i>注意：如果USART不支持低功耗模式唤醒功能，此位保留并由硬件强制清零。</i>
19:18	保留	必须保持复位值。
17	CMCF	字符匹配标志清零 向此位写1时，USARTx_ISR寄存器中CMF标志清零。

16:14	保留	必须保持复位值
13	UDRCF	下溢标志清零 向此位写1时，USART_ISR寄存器中UDRF标志清零。 <i>注意：如果USART不支持同步从设备模式，该位保留且需保持复位值。</i>
12	保留	必须保持复位值
11	RTOCF	接收超时标志清零 向此位写1时，USARTx_ISR寄存器中RTOF标志清零。 <i>注：如果USART不支持接收器超时功能，该位保留并由硬件强制清零。</i>
10	保留	必须保持复位值
9	CTSCF	CTS清零标志 向此位写1时，USARTx_ISR寄存器中CTSIF标志清零。 <i>注：如果不支持硬件流控制功能，该位保留并由硬件强制清零。</i>
8	保留	必须保持复位值
7	保留	必须保持复位值
6	TCCF	发送完成标志清零 向此位写1时，USARTx_ISR寄存器中TC标志清零。
5	保留	必须保持复位值。
4	IDLECF	空闲线路标志清零 向此位写1时，USARTx_ISR寄存器中IDLE标志清零。
3	ORECF	上溢错误标志清零 此位写入1时，USARTx_ISR寄存器中ORE标志清零。
2	NECF	噪声检测标志清零 向此位写1时，USARTx_ISR寄存器中NE标志清零。
1	FECF	帧错误标志清零 此位写入1时，USARTx_ISR寄存器中FE标志清零。
0	PECF	奇偶校验错误标志清零 此位写入1时，USARTx_ISR寄存器中PE标志清零。

29.6.12 USART 接收数据寄存器(USARTx_RDR)

偏移地址：0x24

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res	Res	Res.	Res.	Res.	Res	Res.	RDR[8:0]										
							r	r	r	r	r	r	r	r	r		

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDR[8:0]	接收数据寄存器 RDR寄存器在输入移位寄存器和内部总线之间提供了并行接口的缓冲区。 在使能奇偶校验的时，接收数据的MSB为奇偶校验位。

29.6.13 USART 发送数据寄存器(USARTx_TDR)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
								Res.								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.								TDR[8:0]								
							r	r	r	r	r	r	r	r	r	

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDR[8:0]	发送数据寄存器 TDR寄存器为内部总线和输出移位寄存器之间提供了并行接口缓冲区。 在使能奇偶校验时，发送数据中写入的MSB自动填充为奇偶校验位，此时写入MSB的值无效。 <i>注意：只能在发送数据寄存器未满足TXE=1时写入此寄存器。</i>

29.6.14 USART 预分频器寄存器(USARTx_PRESC)

偏移地址：0x2C

复位值：0x0000 0000

注意：在禁用 USART(UE=0)时才能修改此寄存器。
仅适用于 USART1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												PRESCALER [3:0]			
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PRESCALER[3:0]	<p>时钟预分频器</p> <p>USART_KCLK 可通过预分频系数进行分频：</p> <p>0000：输入时钟未分频</p> <p>0001：输入时钟 2 分频</p> <p>0010：输入时钟 4 分频</p> <p>0011：输入时钟 6 分频</p> <p>0100：输入时钟 8 分频</p> <p>0101：输入时钟 10 分频</p> <p>0110：输入时钟 12 分频</p> <p>0111：输入时钟 16 分频</p> <p>1000：输入时钟 32 分频</p> <p>1001：输入时钟 64 分频</p> <p>1010：输入时钟 128 分频</p> <p>1011：输入时钟 256 分频</p> <p>其余组合：保留</p> <p>注意：PRESCALER 写入未定义组合时，预分频值将为“1011”，即输入时钟256分频。</p>

30 低功耗通用异步收发器 (LPUART)

30.1 简介

低功耗通用异步收发器 LPUART(Low-power Universal Asynchronous Receiver Transmitter)提供了灵活方便的串行数据交换接口。支持全双工或半双工，同步或异步的传输方式。LPUART 提供了可编程的波特率发生器，能对系统时钟进行分频产生 LPUART 发送和接收所需的特定频率。

LPUART 是低功耗设计的 USART，提供相应的硬件支持。在实现双向 USART 通信的同时保证低功耗。仅用 32.768 kHz LSE 低速时钟源就能实现高达 9600 波特率的 USART 通信。在比 LSE 速率更高的时钟源下可支持更高波特率的通信。

即使当微控制器处于低功耗模式，在能耗极低的状态下 LPUART 也可以接收到通信数据。

LPUART 支持单线半双工通信，还支持多机通信和硬件流控(CTS/RTS)。支持 DMA 功能，可实现高速率的数据通信。

模块提供 APB 总线接口，可用于 32 位平台。

30.2 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可编程波特率
- 32.768 kHz 时钟源下支持波特率范围 300 波特/s 到 9600 波特/s，使用高频时钟源可实现更高的波特率
- 双时钟域，带有独立于 PCLK 的专用内核时钟支持外设；
- 内置两个深度为 8 的接收 FIFO 和发送 FIFO
- 数据字长度可编程（7 位、8 位或 9 位）
- 数据顺序可编程，支持 MSB 或 LSB 优先
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 使用 DMA 实现连续通信
- 为发送/接收配有独立的使能位；
- 可单独控制收/发的信号极性
- Tx/Rx 引脚配置可交换
- 通信控制/错误检测标志
- 提供传输检测标志
 - 接收缓冲区已满
 - 发送缓冲区已空
 - BUSY 标志和发送结束标志
- 奇偶校验控制：
 - 发送奇偶校验位

- 对接收数据进行奇偶校验
- 四个错误检测标志:
 - 上溢错误
 - 噪声检测
 - 帧错误
 - 奇偶校验错误
- 带有中断标志位的中断源
- 支持硬件流控(CTS/RTS)以及 RS485 收发控制
- 多机通信时, 支持从静默模式唤醒(通过空闲线检测或地址标记检测)。

30.3 LPUART 功能一览

表 30-1 LPUART 支持功能一览

USART 全功能	LPUART 功能
硬件流控	√
使用 DMA 进行连续通信	√
多机通信	√
单线半双工通信	√
智能卡模式	×
IrDA SIR ENDEC 模块	×
双时钟域, 从低功耗模式唤醒	√
接收器超时中断	×
Modbus 通信	×
自动波特率检测	×
RS485 收发器控制	√
USART 数据长度	7, 8 和 9 位
Tx/Rx FIFO	√
Tx/Rx FIFO 大小	8

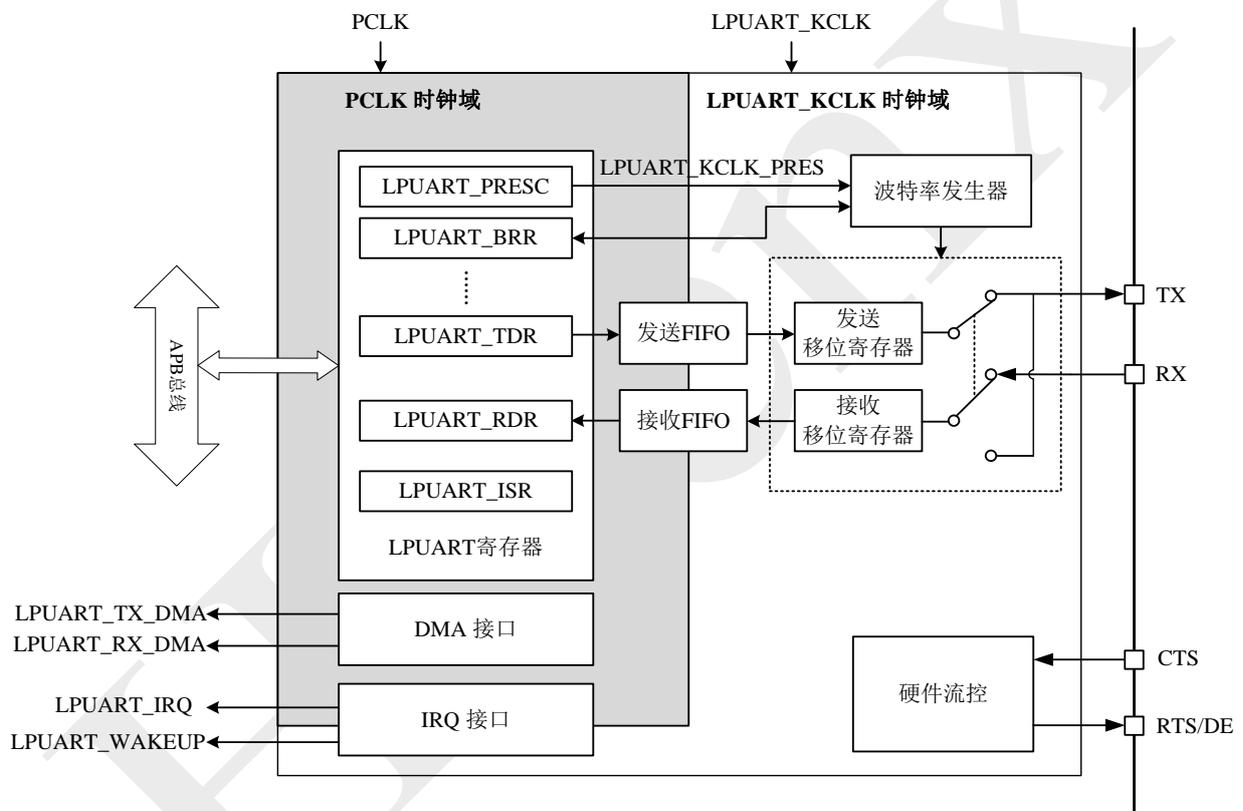
USART 全功能	LPUART 功能
预分频	√

注意：“√”代表支持此功能，“×”代表不支持此功能

30.4 LPUART 功能详解

30.4.1 LPUART 框图

图 30-1 LPUART 框图



如图所示表示两个完全独立的时钟域：

- LPUART_pclk 时钟域，连接 APB 总线时钟，读写寄存器需要此时钟源。
- LPUART_KCLK 时钟域以 LPUART_KCLK 为时钟源，此时钟源可独立于 LPUART_pclk。所以在 LPUART_KCLK 时钟停止时依然可以对 LPUART_pclk 时钟域的寄存器区域进行读/写操作。
- 不启用双时钟域时 LPUART_KCLK 等同于 LPUART_pclk。

LPUART_ker_clk 和 LPUART_pclk 是完全独立的，相互之间没有限制关系，可独立配置速率。

RS232 硬件流程控制：

RS232 流控中需要用到以下引脚：

- CTS (Clear To Send):
此引脚为输入，接收到高电平时，当前传输完成后数据发送被阻止。
- RTS (Request To Send):
此引脚为输出，LPUART 准备好接收数据时，将此引脚拉低。

RS485 硬件流程控制：

RS485 流控中需使用以下引脚：

- DE (Drive Enable):
此引脚输出信号用于开启外部收发器的发送模式。

注意： *DE 和 RTS 复用同一引脚。*

30.4.2 LPUART 信号详解

LPUART 双向通信至少需要两个引脚：数据接收--Rx 和数据发送--Tx：

- RX 数据接收（输入）：串行输入
- TX 数据发送（输出）：
禁止发送时 TX 引脚恢复到可编程 I/O 状态。如果发送使能但是没有数据需要发时 TX 引脚为高电平 “H”。在单线通信模式此引脚用于数据的发送和接收。

30.4.3 LPUART 字符详解

可通过对 LPUART_CR1 寄存器中的 M0 (12bit) 和 M1 (28bit) 将字长设置为 7 位、8 位或 9 位（请参见图 316）。

- M[1:0] = “10” 7 位字长；
- M[1:0] = “00” 8 位字长；
- M[1:0] = “01” 9 位字长；

Tx 和 Rx 在起始位时默认都是低电平，停止位期间都是高电平。信号的极性可以分别通过独立的极性配置取反。

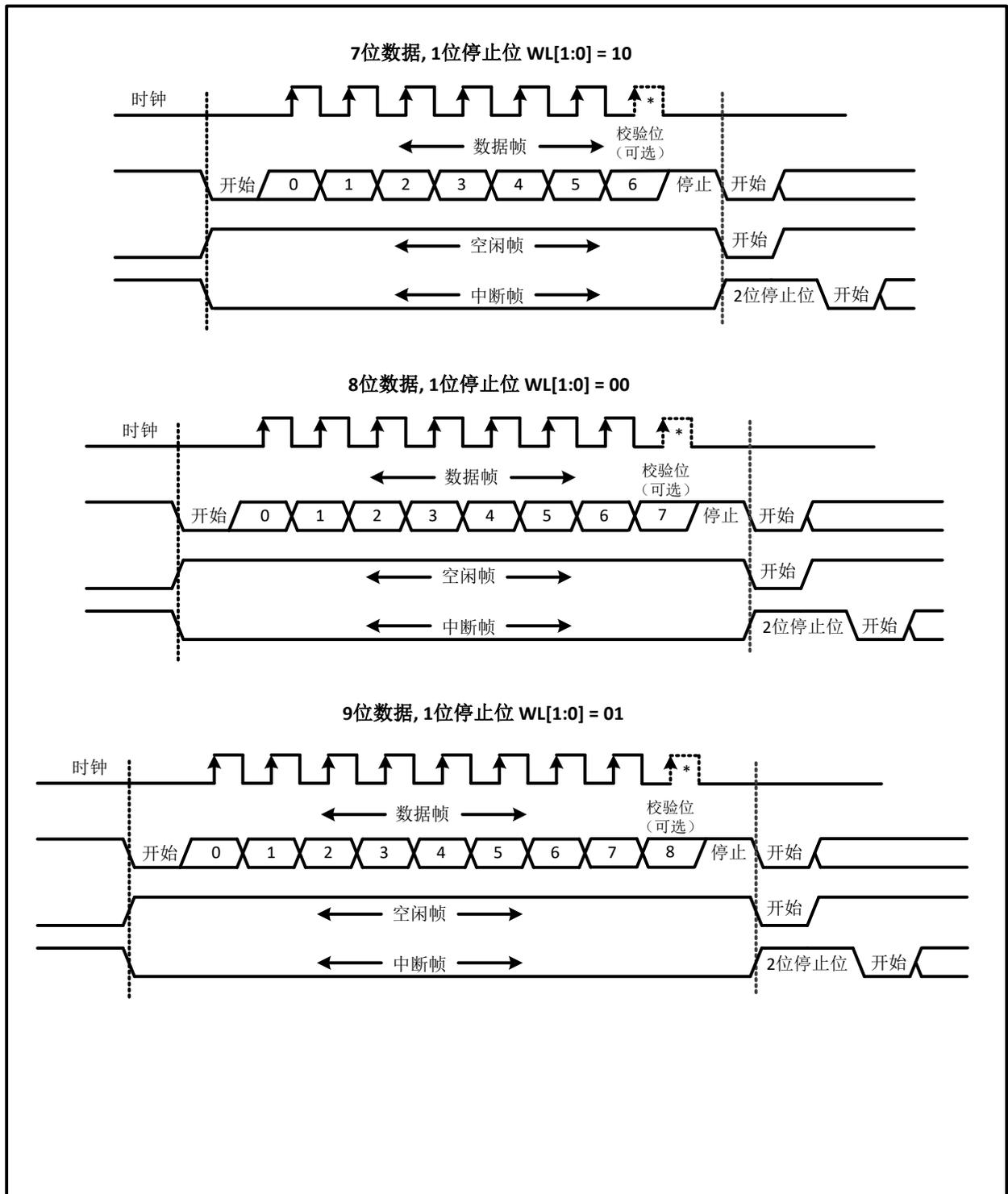
空闲字符 (Idle Character) 可理解为一整帧全“1”（包括停止位的电平也是“1”）。

中断字符 (Break Character) 可理解为在一帧的时间内接收到了全“0”。中断字符帧的末尾由发送器插入 2 位停止位。

发送和接收操作由通用波特率发生器驱动。当发送器或接收器的使能位置 1 时产生发送时钟或接收时钟。

详细说明请参见下图。

图 30-2 LPUART 字长详解



30.4.4 LPUART FIFO 和阈值设置

LPUART 配有深度为 8 的一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。将 LPUART_CR1 寄存器中的 FIFOEN 位置 1，即可使能 FIFO 模式。

由于最大字符长度为 9 位，所以 TXFIFO 的宽度为 9 位。但是 RXFIFO 的默认宽度为 12 位，这是因为在接收 FIFO 中，不仅存储数据还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注意： 接收的数据和相关错误标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取到数据。状态标志位于 LPUART_ISR 寄存器中。

可以由 TXFIFO 或 RXFIFO 设置的阈值触发中断。阈值的设置在 LPUART_CR3 寄存器中的 RXFTCFG[2:0]和 TXFTCFG[2:0]位域编程。

在使能阈值时：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG[2:0]位域中编程的阈值时，触发中断。此时，LPUART_ISR 寄存器中的阈值标志 RXFT 置 1，表示已接收到 RXFTCFG[2:0]个数据，此时接收数据寄存器 LPUART_RDR 中存有 1 个数据，RXFIFO 中存有(RXFTCFG - 1) 个数据。

例如，通过 RXFTCFG[2:0]位域编程为“010”将阈值设定为 RXFIFO 总深度的 1/2。则接收到 RXFIFO 总深度一半的数据时认为到达阈值，RXFT 标志置 1。此时 RXFIFO 中存有 $(8 \times 0.5 - 1 = 3)$ 3 个数据，LPUART_RDR 中存有 1 个数据。

- 当 TXFIFO 中的未使用空间数达到在 TXFTCFG[2:0]位域中编程的阈值时，会触发中断，LPUART_ISR 寄存器中的 TXFT 标志位置 1。

30.4.5 LPUART 发送器

根据 M[1:0]的配置状态，发送的数据字长可以是 7 位、8 位或 9 位。发送使能位 (TE) 置 1 即可激活发送器。发送移位寄存器中的数据由 TX 引脚输出。

字符发送：

LPUART 发送期间，首先通过 TX 引脚移出数据的最低有效位 LSB（默认）。在该模式下，LPUART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

启用 FIFO 时，需要写入发送数据寄存器 (LPUART_TDR) 中的数据会预先顺序存放在 TXFIFO 中。发送时每个字符都以起始位（1 位逻辑低电平）引领，字符末尾以停止位结尾，停止位的数量可配置为 1 或 2 位。

使用 DMA 通信时，每次 TXE（使能 FIFO 模式时为 TXFNF）置位时，数据都从 DMA 中配置的存储区加载到 LPUART_TDR 寄存器中。

注意： 向 LPUART_TDR 中写入要发送的数据前，TE 位需要先置为“1”。数据发送期间复位 TE 位会冻结波特率计数器，进而破坏 TX 引脚上的数据，当前发送的数据随即丢失，所以不建议在发送期间复位 TE 位。

TE 位使能时会先发送一个空闲帧。

配置停止位:

可以在 LPUART_CR2 的 STOP[1:0]位编程停止位的数量，停止位将安插在每一个字符后发送。停止位的配置有以下几点常用选项：

- 1 个停止位：停止位数默认值。
- 2 个停止位：正常 UART 模式、单线半双工模式。

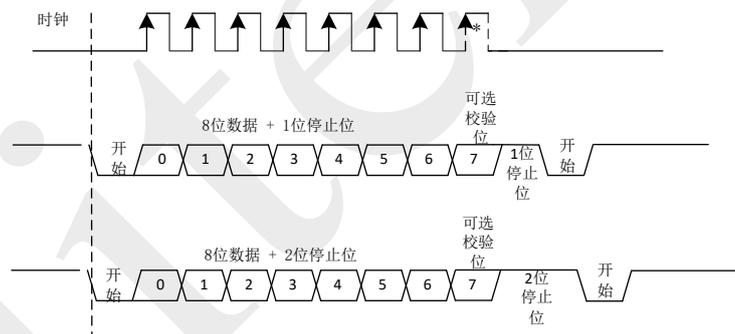
发送的空闲帧中也包含停止位。

中断帧发送 10 个低电平位 (M[1:0] = “00” 时)、11 个低电平位 (M[1:0] = “01”) 或 9 个低电平位 (M[1:0] = “10” 时) 结尾包含 2 位停止位。过长的中断帧 (长度大于 9/10/11 个低电平位) 无法发送。

数据末尾位可选为奇偶校验位。奇偶校验位与停止位不同，不在末尾追加而是占用一位数据位。

下图 32-3 为不同停止位数量配置示例：

图 30-3 配置停止位



字符发送配置步骤:

- 配置 LPUART_CR1 中的 M[1:0]定义字长。
- 配置 LPUART_BRR 寄存器选择通信波特率。
- 配置 LPUART_CR2 中的 STOP[1:0]设置停止位数。
- 通过向 LPUART_CR1 寄存器中的 UE 位写入 “1”使能 LPUART。
- 进行多缓冲区通信时请选择 LPUART_CR3 中的 DMA 使能位 (DMAT)。按照 [29.5.10 USART 多机通信](#) 中的说明配置 DMA 寄存器。
- 将 LPUART_CR1 中的 TE 位置 1 自动先发一个空闲帧。
- 在 LPUART_TDR 寄存器中写入准备发送的数据；
 - 禁止 FIFO 时，向 LPUART_TDR 写入数据会将 TXE 标志清零，表示 LPUART_TDR 发送寄存器中数据已满。

- 使能 FIFO 模式时，向 LPUART_TDR 写入数据会移入 TXFIFO 中。LPUART_ISR 标志寄存器的 TXFNF 标志位为“1”期间，表示 TXFIFO 未
满，此时可以写入 LPUART_TDR。如果此标志清零，表示 TXFIFO 已满，
不能继续写入 LPUART_TDR。

- 当最后一个待发送的数据写入 LPUART_TDR 寄存器后，需等待发送完成直到 LPUART_ISR 中的发送完成标志位 TC 置 1。TC 置位时代表以下内容：
 - 禁止 FIFO 时表示最后一个字符已发送完成。
 - 使能 FIFO 时表示 TXFIFO 和移位寄存器都已清空。
- 在禁用 LPUART 或是进入 CPU_HALT 模式之前需要检查 TC 标志，以保证最后一帧完成发送。

字符发送过程：

- 禁止 FIFO 时：

由于 LPUART_TDR 寄存器只有 1 字符空间，所以写入 1 个字符后，发送数据非空标志 TXE 位清零。

发送数据寄存器 LPUART_TDR 为空时，TXE 标志位自动置 1，此时表示：

- 数据已由 LPUART_TDR 寄存器移入移位寄存器，开始发送数据；
- 可以在 LPUART_TDR 寄存器中写入数据，前一数据不会被覆盖。

TXEIE 位置 1 使能中断时，TXE 标志置 1 时会触发中断。

发送时，写入的待发数据存储在 LPUART_TDR 寄存器中作为缓冲。新数据在当前发送结束时复制到移位寄存器中。

在发送未进行 (TE=0) 时，写入 LPUART_TDR 寄存器的数据实际存放在移位寄存器中，数据发送开始后，发送数据非空标志 TXE 位立即置 1，指示可以写入下一数据。

- 使能 FIFO 时，LPUART_ISR 寄存器中的发送 FIFO 未空标志位 TXFNF 置 1，指示以下状态：
 - LPUART_TDR 寄存器为空；
 - 可以向 LPUART_TDR 寄存器中写入数据，前一数据不会被覆盖。在发送进行时写 LPUART_TDR 寄存器，会将数据存储在 TXFIFO 中。当前发送结束时，下一数据从 TXFIFO 中复制到移位寄存器中。

TXFIFO 未空时，写 LPUART_TDR 寄存器不会将 TXFNF 标志清零，此标志位保持置 1，直到 TXFIFO 满时清零。LPUART_CR1 中 TXFNFIE 位置 1 使能中断时，TXFNF 置 1 会触发中断。

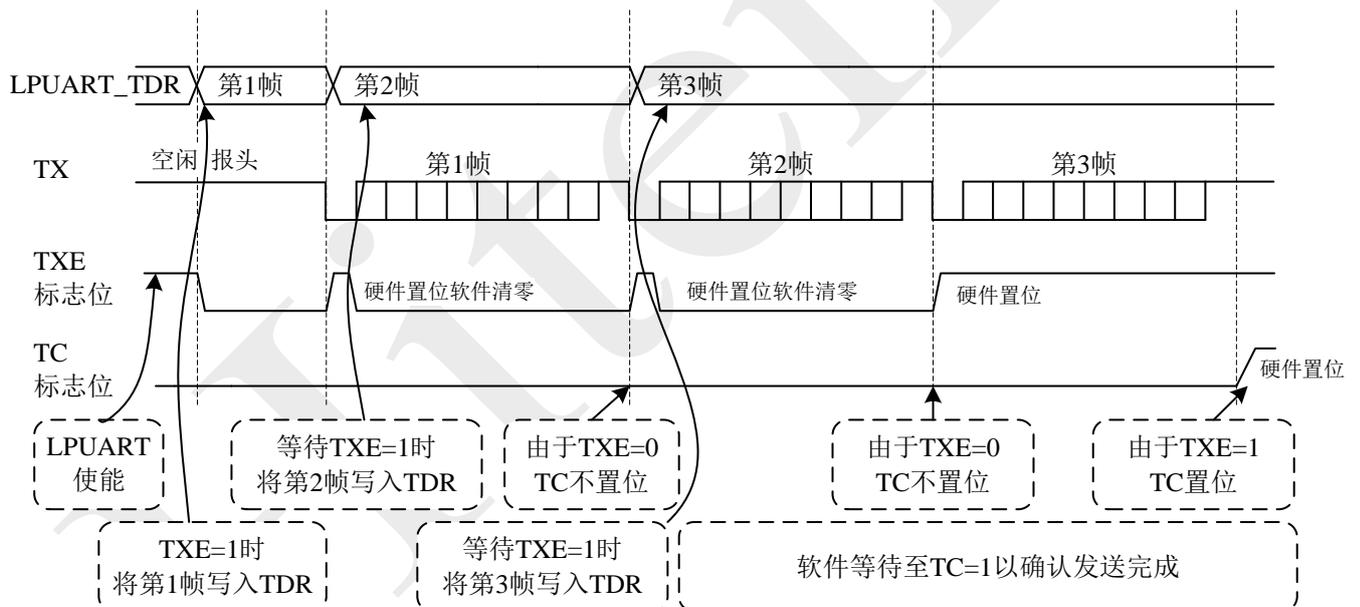
使用 FIFO 阈值时，配置当 TXFIFO 中未使用的空间数达到阈值时触发中断，响应中断时，再将数据写入 TXFIFO。这样的配置可以对 LPUART_TDR 连续写入多次，写入次数取决于阈值的大小。

随着停止位发送完毕，一帧数据发送完成，如果此时发送数据非空标志 TXE（使能 FIFO 时对应 TXFE）置 1，表示后续没有等待发送的数据，发送过程完成，随即发送完成标志位 TC 置位。如果 LPUART_CR1 寄存器中 TCIE 位置 1 使能中断，则 TC 标志置位同时将触发中断。

发送时的 TC 和 TXE 的动作请参见下图：

注意： 为确保最后一个字符完整发送，向 LPUART_TDR 寄存器中写入最后一个数据后，需要等待发送完成标志 TC 置 1 后，才可以禁止 LPUART 或进入低功耗模式。其中，Stop 模式下，如果选用可以正常工作的时钟源，则 LPUART 也可以继续发送数据，无此限制。在 Sleep 模式下 LPUART 可以正常工作，无此限制。

图 30-4 发送过程 TC 和 TXE 的动作（FIFO 禁止时）



注意： 当 FIFO 使能时数据传输标志位为 TXFNF

中断帧：

将请求寄存器 LPUART_RQR 的 SBKRQ 位置 1 则在当前发送结束时发送一个中断帧。中断帧的长度取决于 M[1:0] 位（请参见图 32-4）。

SBKRQ 位置 1 后，当前字符发送完成后将在 TX 线路上发送一个中断帧。此位由软件置位，在中断帧发送完成后的停止位期间由硬件自动清零。

LPUART 在中断帧末尾插入两个停止位以确保下一帧的起始位被识别。

使能 FIFO 时，即使 TXFIFO 已满，发送中断帧的优先级也高于数据的发送，

仍会先发中断帧再继续数据的发送。所以如果需要完成数据发送后再发送中断帧，则需要等待发送数据寄存器空标志 TXFE 置 1 后，再置位 SBKRQ。

空闲帧

将 TE 位置 1 则 LPUART 在第一个数据帧之前先发一个空闲帧。

30.4.6 LPUART 接收器

LPUART 可接收 7 位、8 位或 9 位长度的字符，可在 LPUART_CR1 寄存器中 M[1:0] 位域配置。

起始位检测：

Rx 接收线上出现下降沿时启动起始位检测，后续会对起始位中间部分进行 1 次采样，确保后续也保持在逻辑“0”。如果起始位采样出现了逻辑“1”，则噪声标志 NOISE 为置位，起始位被丢弃，继续等待新的起始位。如果正常检测到起始位，则继续对后续数据进行采样。

字符接收过程：

接收过程中 LSB 先收，LPUART_RDR 作为内部总线和接收移位寄存器之间的缓冲，相关配置详见以下步骤：

- 配置 LPUART_CR1 中的 M[1:0] 定义字长。
- 配置 LPUART_BRR 寄存器选择通信波特率。
- 配置 LPUART_CR2 中的 STOP[1:0] 设置停止位数。
- 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
- 进行多缓冲区通信时请选择 LPUART_CR3 中的 DMA 使能位 (DMAT)。按照 [29.5.10 USART 多机通信](#) 中的说明配置 DMA 寄存器。
- LPUART_CR1 寄存器 RE 位置 1，使能接收器开始搜索起始位。
- 等待 REACK 标注置位以确认接收器准备完毕。

接收字符时：

- 不使用 DMA 通信：
 - 禁止 FIFO 时，接收非空标志 RXNE 置 1 时，表示接收移位寄存器中的数据已经传送到 LPUART_RDR 寄存器中，指示已接收到数据。
由于 LPUART_RDR 寄存器中仅有 1 字符空间，读 LPUART_RDR 寄存器时 RXNE 标志位清零。也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 位写 1，使用清空接收数据的方式将 RXNE 标志清零。
 - 使能 FIFO 时，接收 FIFO 非空标志 RXFNE 位置 1 时，表示 RXFIFO 中已接收到数据，读 LPUART_RDR 寄存器时获取接收数据。接收到

的数据和相应错误标志位一起存储在 RXFIFO 中。当 RXFNEIE 位置 1 时，触发中断。

RXFIFO 为空时，RXFNE 标志位清零，也可以通过向 LPUART_RQR 中的 RXFRQ 位写 1，用清空 RXFIFO 的方式将 RXFNE 标志清零，RXFIFO 中的数据被丢弃。

如果配置为达到 RXFIFO 阈值时触发中断，则可以根据 FIFO 阈值的设置批量读取数据。

- 使用 DMA 通信：
 - 禁止 FIFO 时，由于 LPUART_RDR 寄存器中仅有 1 字符空间，每收到 1 个字符，接收非空标志位 RXNE 都会置 1，DMA 读取 LPUART_RDR 后 RXNE 位清零。
 - FIFO 使能时，RXFIFO 非空时 RXFNE 位置 1，每次 DMA 读请求都从 RXFIFO 中读取一个数据。
- 接收期间如果检测到帧错误、噪声错误上溢错误或奇偶校验错误，相应错误标志位置位。

中断字符：

接收到中断字符时，LPUART 按帧错误进行处理。

空闲帧

检测到空闲帧时处理方式和接收数据基本一致，区别在于空闲帧出现时，IDLE 标志位置 1，如果 LPUART_CR1 中的 IDLEIE 位置 1，会触发中断。

上溢错误

上溢错误检测默认开启，当检测到上溢错误时，上溢错误标志 ORE 位置 1，如果中断使能，即 LPUART_CR1 中的 EIE 位置 1，则触发中断。

通过向 LPUART_ICR 寄存器中的 ORECF 位写 1 来清零 ORE 位。

可以通过将 LPUART_CR3 中的 ORED 位置 1 禁止上溢错误检测，禁止后 ORE 标志保持为 0。

满足以下条件时生成上溢错误：

- 禁止 FIFO 时：

每接收到一个字符后，接收非空标志位 RXNE 置 1。当 RXNE 标志为 1 时，或上一个 DMA 读请求未处理时，又接收到新的字符，就会产生上溢错误。RXNE 位清零前，数据无法从移位寄存器传到接收数据寄存器 LPUART_RDR 中。

发生上溢错误时有以下动作：

- 上溢错误标志 ORE 位置 1；
- LPUART_RDR 寄存器中的内容不会丢失。读取 LPUART_RDR 寄存器时，得到溢出前接收到的最后一个数据。
- 移位寄存器被新数据覆盖，在上溢期间接收到的数据都将丢失。
- LPUART_CR1 中的 EIE 位置 1 时，会触发中断。

ORE 位置 1 时，有以下两种可能：

- RXNE = 1，表示最后一个有效数据已存储在 LPUART_RDR 可以读取。
- RXNE = 0，RDR 寄存器中已没有需要读取的数据。这种情况可能是由于在读取 RDR 寄存器中最后一个有效数据的同时接收到新数据，此新数据丢失。

- 使能 FIFO 时：

当 RXFIFO 已满 (RXFF=1) 时又接收到新的数据，就会发生上溢错误。

在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 LPUART_RDR 寄存器。当出现空闲位置时，RXFIFO 非空，接收 FIFO 非空标志位 RXFNE 置 1。

发生上溢错误时：

- 上溢错误标志 ORE 位置 1。
- RXFIFO 中接收到的数据不会丢失。
- 移位寄存器将被覆盖。在上溢期间接收到的任何数据都会丢失。
- 如果 EIE 位置 1 时，会触发中断。

注意：ORE 位置 1 时，表示至少已有 1 个数据丢失。

配合过采样模式选择时钟源：

时钟源可通过时钟控制系统进行选择，详见 [RCC 章节：异步时钟配置寄存器](#)。需在使能 LPUART (LPUART_CR1 寄存器的 UE 位置 1) 之前选择时钟源。

依据以下两个条件选择时钟源：

- 可在低功耗模式下使用 LPUART；
- 通信速率；

LPUART 支持双时钟域。可在 RCC_CCIPR 寄存器中将 LPUART_KCLK 的时钟源选择为 HSI、LSE、SYSCLK 或 PCLK。

时钟源选择为 HSI 或 LSE 时，LPUART 可在 Stop 模式下接收数据，并支持将芯片从 Stop 模式唤醒。如果时钟源选择为 SYSCLK 或 PCLK，由于这些时钟源

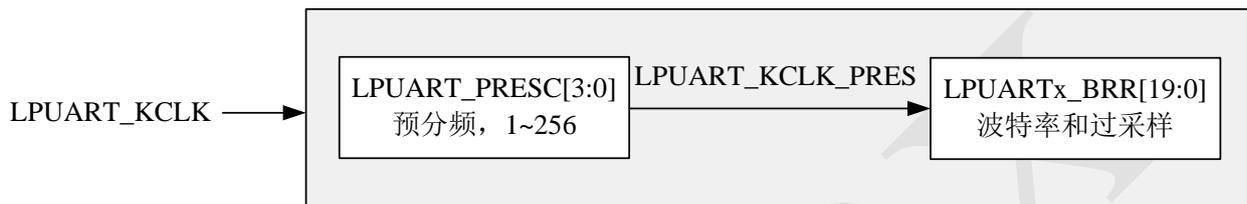
在 Stop 模式下不工作，所以 LPUART 也不能通信。

芯片唤醒后，已接收的数据可以通过读 LPUART_RDR 寄存器，或通过 DMA 传输获取。

时钟源直接影响通信速率，限制了最大通信速率。

LPUART_KCLK 时钟通过 LPUART_PRESC 寄存器中定义的系数进行预分频，得到 LPUART_KCLK_PRES。

图 30-5 LPUART_KCLK 时钟分频框图



帧错误:

接收数据时如果在预期时间内未能识别出停止位，则检测为帧错误，同步失败或噪声过大的情况下可能引发此错误。

检测到帧错误时有以下动作：

- 帧错误标识 FE 位由硬件置位。
- 无效数据从移位寄存器传送到 LPUART_RDR 寄存器（使用 FIFO 模式时为 RXFIFO）。
- FE 位置 1 时，如果 LPUART_CR3 寄存器中的 EIE 位置 1，则触发中断。FE 置 1 时，RXNE 位（使用 FIFO 模式时为 RXFNE 位）也被置位，如果 RXNE 位置 1 会触发中断。

FE 位清零的方式为将 LPUART_ICR 寄存器中的 FECF 位置 1。

配置接收停止位:

可通过 LPUART_CR2 中的 STOP 控制位可以配置接收时停止位的数量，可以是 1 个或 2 个停止位。

- 1 个停止位：在第 8、第 9 和第 10 个采样沿采样 1 位停止位。
- 2 个停止位：对 2 个停止位的采样发生在对第二个停止位的中间位置。如果采样检测到帧错误，则帧错误标志位 FE 和 RXNE 标志将置 1。不对第一个停止位检测帧错误。

30.4.7 LPUART 波特率产生

通过 LPUART_BRR 设置发送和接收的时钟频率：

$$\text{Tx 或 Rx 的波特率} = \frac{256 * \text{LPUARTckpres}}{\text{LPUARTDIV}}$$

LPUARTDIV 的值在 LPUART_BRR 寄存器中设置。

LPUARTDIV 取值范围：

$$0x1000 \cong \text{LPUARTDIV} \cong 0x300$$

注意： 对 LPUART_BRR 执行写操作后，波特率计数器随即更新。因此不应在通信时更改此寄存器的值。

LPUART_BRR 寄存器赋值小于 0x300 为非法操作。

时钟频率 f_{CK} 的范围需在 3 倍波特率到 4096 倍波特率之间。

在时钟源为 LSE 时波特率上限为 9600, 需要更高的波特率时需要将时钟源切换为更高速的时钟。例如使用 48MHz 时钟源可以实现的最大通信波特率为 16M。

表 30-2 LPUART_KCLK_PRE=32.768KHz 时的波特率误差

波特率目标值	波特率实际值	BRR 寄存器的值	误差 (%)
300bps	300bps	0x6D3A	0
600bps	600bps	0x369D	0
1200bps	1200.087bps	0x1B4E	0.007
2400bps	2400.17 bps	0xDA7	0.007
4800bps	4801.72 bps	0x6D3	0.035
9600bps	9608.94 bps	0x369	0.093

30.4.8 LPUART 接收容差

仅当时钟系统总偏差小于 LPUART 接收容差时，接收器才可以正常工作。影响系统总时钟偏差的因素有以下几点：

- DTR_ERR: 由发送器误差引起的偏差,其中包含发送器本地振荡器的偏差。
- DQU_ERR: 接收器的波特率量化误差。
- DRX_OSERR: 接收器本地振荡器的偏差。
- DL_ERR: 传输线路引入的偏差（通常是由于收发器引入了上升沿下降沿的时序不对称）

需以下条件：

$\text{DTR_ERR} + \text{DQU_ERR} + \text{DRX_OSERR} + \text{DL_ERR} + \text{DWKUP} < \text{LPUART 接收容差}$ 。

其中 DWKUP 是从低功耗唤醒时，采样点偏差产生的误差。

[LPUART 的接收容差表](#)中说明了可正确接收数据的最大接收容差，影响接收容差的条件有以下几点：

- LPUART_CR1 寄存器中的 M[1:0]位域设置的 9、10 或 11 位字符长度。
- LPUART_CR1 寄存器中的 STOP[1:0]位域中设置的停止位位数。
- LPUARTx_BRR 寄存器的配置。

表 30-3 LPUART 的接收容差

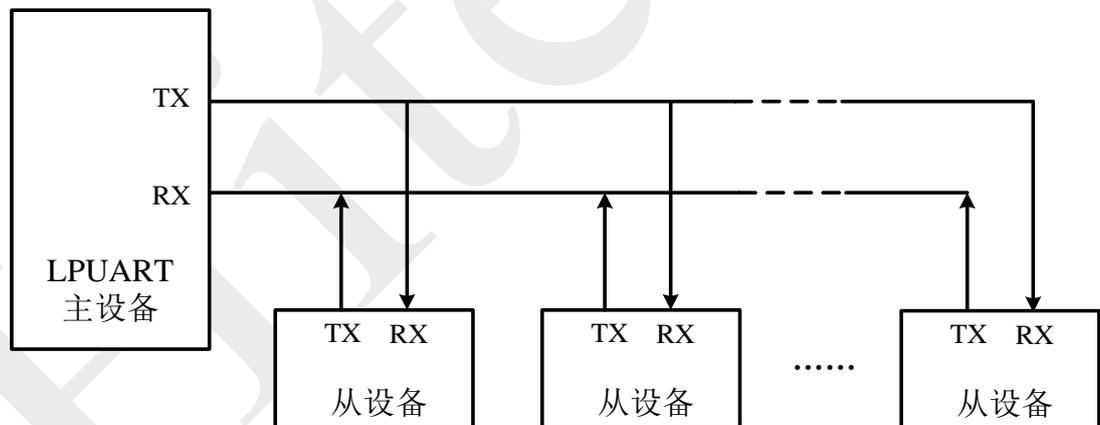
停止位	字符长度	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
1 位 STOP=00	8 bits (M=00)	1.82%	2.56%	3.90%	4.42%
	9 bits (M=01)	1.69%	2.33%	2.53%	4.14%
	7 bits (M=10)	2.08%	2.86%	4.35%	4.42%
2 位 STOP=10	8 bits (M=00)	2.08%	2.86%	4.35%	4.42%
	9 bits (M=01)	1.82%	2.56%	3.90%	4.42%
	7 bits (M=10)	2.34%	3.23%	4.92%	4.42%

30.4.9 LPUART 多机通信模式

LPUART 多机通信，即多个 LPUART 连接在同一个网络中。

例如，一个 LPUART 作为主设备，其 TX 引脚输出到其它 LPUART 的 RX 引脚；作为从设备的多个 LPUART 将各自的 TX 引脚线与后，连接到 LPUART 主设备的 RX 引脚，参见下图：

图 30-6 多机通信连接示意图



在多机通信中，最理想的状态是，只有期待的接收器响应，并主动接收完整的消息内容，这样可以减少由未被寻址的接收器造成的冗余开销。

静默模式就是为此目标而设计，多机通信时，地址不匹配的接收器进入静默模式。通过 LPUART_CR1 寄存器的静默模式使能位 MME 置 1，使能静默模式。

注意： MME 已置 1 时，不推荐清零 MME 位后，在 2 个 LPUART_KCLK_PRE 周期内再快速置 1，此操作可能因清零状态无法被识别而导致静默模式持续有效。

静默模式使能时：

LPUART_ISR 寄存器中的 RWU 位（静默模式状态指示位）不会立即置 1。在不同的场景下，RWU 位可由软件置位（向 LPUART_RQR 寄存器中的 MMRQ

写 1)，或由硬件自动控制，详情请参见图：使用地址匹配检测时的静默模式。
进入静默模式后：

- 接收状态位不会被置 1；
- 任何接收中断都被禁止；

从静默模式退出的条件在 LPUART_CR1 寄存器中的 WAKE 位配置，具体如下：

- WAKE 位为 0 时，使用空闲帧检测。
- WAKE 位为 1 时，使用地址匹配检测。

使用空闲帧检测 (WAKE=0)

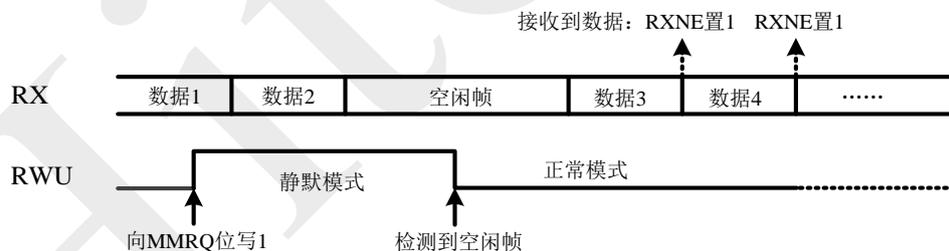
向 LPUART_RQR 寄存器的 MMRQ 位写 1 进入静默模式，RWU 位自动置 1，（指示 LPUART 已进入静默模式）。

当检测到空闲帧时，LPUART 退出静默模式。此时 RWU 位自动清零，但 LPUARTx_ISR 寄存器中的空闲标志 IDLE 位不会置 1。

使用空闲帧检测的多机通信网络中，主设备发送空闲帧时，所有的从设备同时从静默模式退出，实现多从设备同步。

使用空闲帧检测时，静默模式的动作参见下图：

图 30-7 使用空闲帧检测的方式从静默模式唤醒



注意：从检测到空闲帧开始，到收到下一数据的起始位期间，接收器从静默模式退出，在此期间将 MMRQ 位置 1 无效，不能重新进入静默模式（RWU 不置 1）。需要在收到下一数据的起始位以后，到下一空闲帧到来之前向 MMRQ 位写 1，才可以重新进入静默模式。

主设备可以不通过翻转 TE（先清零再置 1）的方式产生空闲帧，只要保持空闲状态不小于 1 个空闲帧的时间，从设备也会检测到空闲帧。

使用地址匹配检测 (WAKE=1)

在此模式下，MSB 为 1 的字符识别为“地址”，否则识别为“数据”。

在地址字符中，检测地址长度可以为 4bits 或 6、7、8bits。从设备本地地址在 LPUART_CR2 寄存器的 ADD[7:0]位域中设置，设置方法如下：

- ADDM7 为 0 时：地址长度为 4bits，保存在 ADD[3:0]；

- ADDM7 为 1 时：
 - 7bits 字符长度时，地址长度为 6bits，保存在 ADD [5:0];
 - 8bits 字符长度时，地址长度为 7bits，保存在 ADD [6:0];
 - 9bits 字符长度时，地址长度为 8bits，保存在 ADD [7:0];

向 LPUART_RQR 寄存器的 MMRQ 位写 1 时，LPUART 进入静默模式，RWU 位自动置 1。

当接收的地址字符与本地地址不匹配时，LPUART 接收器不会退出静默模式，RWU 位置 1；RXNE 位不会因不匹配的地址字符置 1，不会触发中断，也不会产生 DMA 请求。

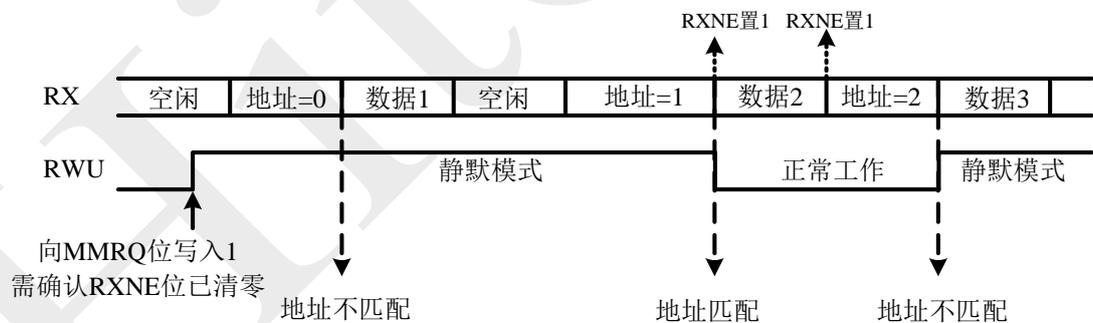
当接收到的地址与本地地址匹配时，LPUART 退出静默模式，RWU 位清零，由于 RWU 位已清零，RXNE/RXFNE 位因接收到匹配地址字符置 1。此后可以开始正常接收后续字符。

再次接收到与本地地址不匹配的地址时，LPUART 接收器重新进入静默模式。

注意： 如果在对接收字符的最后一位采样时 MMRQ 置 1，则进入静默模式之前有可能接收到该数据。

使用地址匹配检测时的示例参见下图：

图 30-8 使用地址匹配检测时的静默模式



30.4.10 LPUART 校验控制

表 30-4 LPUART 校验控制

字长/M[1:0]		PEN 位: 校验控制 (使能:"1")	LPUART 帧格式 ^(*)
8 位	00	0	起始位 8 位数据 停止位
	00	1	起始位 7 位数据 校验位 停止位
9 位	01	0	起始位 9 位数据 停止位
	01	1	起始位 8 位数据 校验位 停止位
7 位	10	0	起始位 7 位数据 停止位

字长/M[1:0]	PEN 位: 校验控制 (使能:"1")	LPUART 帧格式 ^(注)
10	1	起始位 6 位数据 校验位 停止位

注: 在数据寄存器中校验位位于 MSB (根据 M 位的设置可为第 8 位或第 7 位)

偶校验:

偶校验时, 奇偶校验位的值维持一帧内容中包含校验位“1”的数量为偶数 (帧内容可以是 6 个、7 个或 8 个 LSB 位, 具体取决于 M[1:0] 的值)。

例如, 数据为“0011 1010”其中含有 4 个“1”, 则在启用偶校验 (LPUART_CR1 寄存器中 PS 位置 0) 时校验位为“0”。

奇校验

奇校验时, 奇偶校验位的值维持一帧中 (包含校验位)“1”的数量为奇数 (帧可由 6 个、7 个或 8 个 LSB 位组成, 具体取决于 M 位的值)。

例如, 数据为“0011 1010”其中含有 4 个“1”, 则在启用奇校验 (LPUART_CR1 寄存器中 PS 位置 1) 时校验位为“1”。

接收时进行奇偶校验:

如果奇偶校验检查失败, 则 LPUART_ISR 寄存器中的 PE 标志置 1; 如果 LPUART_CR1 寄存器中 PEIE 位置 1, 则会触发中断。通过软件对 LPUART_ICR 寄存器中的 PECF 位写“1”来清零 PE 标志。

发送时生成奇偶校验:

如果 LPUART_CR1 寄存器中的 PEN 位置 1, 则在数据寄存器中写入的数据的 MSB 会替换为奇偶校验位发送。如果选择偶校验 (PS=0), 则“1”的数量为偶数; 如果选择奇校验 (PS=1), 则“1”的数量为奇数。

30.4.11 LPUART 单线半双工通信

LPUART 支持单线半双工通信协议, 通过将 LPUART_CR3 寄存器中的 HDSEL 位置 1 可开启单线半双工通信模式, 此时 TX 和 RX 信号在内部相连接。

向 HDSEL 位写 1 使能单线半双工模式后, 有以下动作:

- TX 和 RX 信号在内部相连接。

默认使用 TX 引脚进行数据收发, RX 引脚不使用, 可做 GPIO 使用。

当 LPUART_CR2 中的 SWAP 为置 1 时, TX/RX 引脚交换使能, 使用 RX 引脚进行数据收发。

- 无数据传输时 TX 处于接收状态。

需要将 TX 引脚配置为复用开漏模式，并配置内置上拉电阻，或连接外部上拉电阻。

除以上几点以外，通信协议与正常 LPUART 通信类似。

注意： 只要数据在 TE 位置 1 时写入数据寄存器 TDR，发送就会持续进行，硬件不会主动阻止发送过程。

单线半双工模式下，不支持 TX、RX 引脚电平反相功能（LPUART_CR2 寄存器中的 TXIVC 和 RXINV 位配置无效）。

30.4.12 LPUART 使用 DMA 通信

LPUART 支持通过 DMA 进行通信。接收和发送有独立的 DMA 请求。

DMA 发送

将 LPUART_CR3 寄存器中的 DMAT 位置 1，使能 DMA 发送。当 TXE 标志（使能 FIFO 时为 TXFNF 标志）置 1 时，DMA 可将数据由预设存储器加载到 LPUART_TDR 寄存器。参考以下步骤建立从 LPUART 到 DMA 的映射（x 表示通道编号）：

- 1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见：[DMA 配置流程](#)。
- 2) 配置 DMAMUX 输出请求。参见：[DMAMUX 配置流程](#)。
- 3) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要发送的总字节数。
- 4) 在 DMA 源地址寄存器 DMA_CSARx 中，配置存储区地址，作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，写入 LPUART_TDR 寄存器地址，作传输目的地址；每次 TXE（使能 FIFO 模式时为 TXFNF）置位时，数据都从存储区加载到 LPUART_TDR 寄存器中。
- 5) 根据具体应用配置中断触发的条件。
- 6) 向 LPUART_ICR 寄存器中的 TCCF 位写 1，清零 LPUART_ISR 寄存器中的 TC 标志。
- 7) 将 DMA_CCx 寄存器中的 EN 位置 1 以使能通道。

当达到 DMA 控制器中设置传输数据次数时，会在 DMA 相应通道上产生中断。

在发送模式下，DMA 完成了对所有需要发送的数据的写操作时，DMA_ISR 寄存器中的 TFx 标志置 1，此外，还可以通过检测 LPUART_ISR 寄存器中的 TC 标志位确认 LPUART 通信是否完成。TC 标志在数据发送期间保持清零，在最后一个字符发送结束后自动置 1。

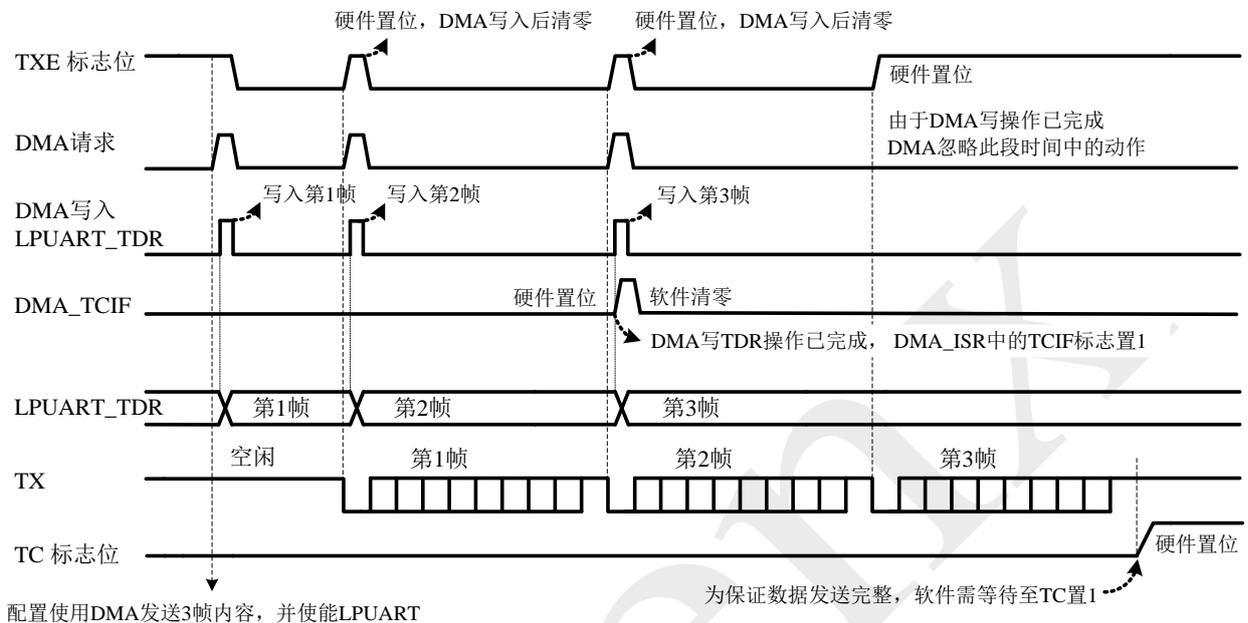
注意： 为避免破坏最后一个字符的传输，在禁止 LPUART 之前或进入低功耗模式之前，需要等待 TC 置位以确认最后一个数据发送完成。

DMA 发送时的时序参见下图，下图中描述了 LPUART 通过 DMA 发送 3 帧内

容的过程：

图 30-9 LPUART 通过 DMA 发送示意图

图中描述了LPUART通过DMA发送3帧内容的过程：



3. **注意：**启用FIFO时DMA请求由发送FIFO未满足事件触发，即TXFNF=1。

DMA 接收

将 LPUART_CR3 寄存器中的 DMAR 位置 1，使能 DMA 接收。

接收字符时，数据从 LPUART_RDR 寄存器加载到 DMA 配置的存储区域中。

参考以下步骤建立从 LPUART 到 DMA 的映射（x 表示通道编号）：

- 1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见：[DMA 配置流程](#)。
- 2) 配置 DMAMUX 输出请求。参见：[DMAMUX 配置流程](#)。
- 3) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要接收的总字节数。
- 4) 在 DMA 源地址寄存器 DMA_CSARx 中，写入 LPUART_RDR 寄存器地址，
- 5) 作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，配置存储区地址，作传输目的地址；每次发生接收非空 RXNE（使能 FIFO 模式时为 RXFNE）置 1 时，数据都从源地址 LPUART_RDR 复制到 DMA 配置接收缓冲区中。
- 6) 根据具体应用配置触发中断的条件。
- 7) 将 DMA_CCx 寄存器中的 EN 位置 1 以使能通道。

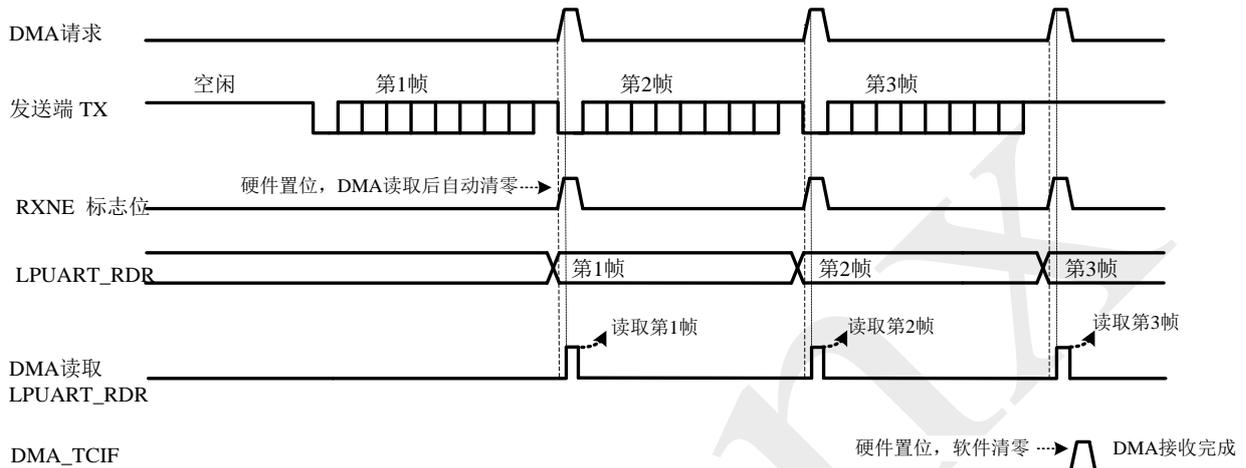
当达到在 DMA 控制器中设置传输数据次数时，在 DMA 相应通道上可以产生

中断。

DMA 接收时的时序参见下图，下图中描述了 LPUART 通过 DMA 接收 3 帧内容的过程：

图 30-10 LPUART 通过 DMA 接收示意图

本图对应发送示意图发送过程中，LPUART通过DMA接收三帧内容的过程：



4. **注意：** 使能 FIFO 时 DMA 请求由接收 FIFO 未满足事件触发，即 $RXFNE=1$ 。

DMA 通信模式下的错误标志及中断

字符接收过程中，出现帧错误，上溢错误或噪声错误时，这些错误标志和 RXNE（使能 FIFO 模式时为 RXFNE）一同置位，将 LPUART_CR3 寄存器中的 EIE 位置 1 时，产生任一错误都会在当前字符结束后触发错误中断。出现校验错误时，LPUART_CR3 寄存器中的 PEIE 置 1 时，在当前字符结束后触发校验错误中断。

在 LPUART_CR3 寄存器的 DDRE 位可配置在接收出错时禁止 DMA。

默认 DDRE=0，接收出错时不禁止 DMA，相应的错误标志置 1，但 RXNE 保持为 0，不产生 DMA 请求，错误数据不会被接收到 LPUART_RDR 寄存器，避免了上溢，但下一个正确数据可以被接收。

DDRE=1 时，接收出错后相应的错误标志和 RXNE 均置 1，错误数据会被接收到 LPUART_RDR 寄存器。但 DMA 请求会被屏蔽。此种情况下接收出错时，软件需禁止 DMA 请求(DMAR = 0)或者将 RXNE 清零，然后再将错误标志清零后才可以继续开启 DMA 接收。

当发生上溢错误时，在错误标志清零前接收到的数据只会刷新移位寄存器，不会覆盖 RDR 中的最后一个数据，所以无论 DDRE 位是否置 1，都需要清除错误标志后才可以继续接收数据。

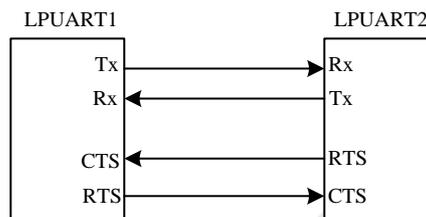
30.4.13 LPUART RS232 硬件流控和 RS485 驱动器

RS232 硬件流控 RTS/CTS 用于通信流量控制，RTS 为输出引脚，CTS 为输入引脚，低电平有效，RTS 和 CTS 引脚连接关系参见下图。

CTS 和 RTS 硬件流控可通过 LPUART_CR3 寄存器中的 RTSE 位和 CTSE 独立配置，置位使能，清零禁止。

下图为此种情况下的连接示意：

图 30-11 两个 LPUART 间串行通信连接例



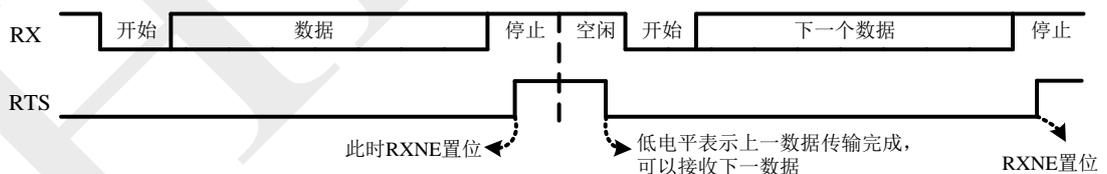
RS232 RTS 流控

使能 RTS 流控 (RTSE=1) 后，接收数据寄存器 LPUART_RDR 空 (RXNE=0) 时，RTS 引脚输出低电平，开始接收数据；接收数据寄存器非空 (RXNE = 1) 时，RTS 引脚输出高电平，停止接收数据。

- RTS 引脚输出低电平时，表示本机 LPUART 准备好接收数据，通知对端设备发送数据。
- 若使能 FIFO，仅当接收 FIFO 已满的时，RTS 输出高电平。

RTS 流控通信示例参见下图：

图 30-12 RTS 动作示例



注意： 使能 FIFO 时由 RXFIFO 状态控制 RTS。

RS232 CTS 流控

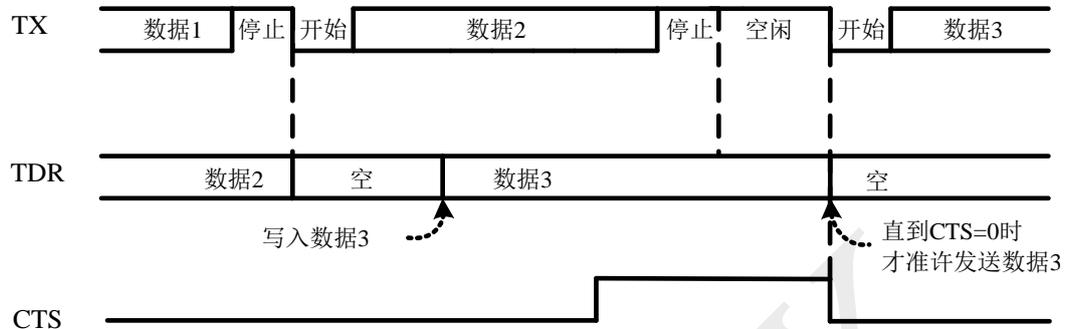
使能 CTS 流控 (CTSE=1) 时，发送器会在发下一字符前检查 CTS。只要 CTS 电平发生变化 (由低到高或由高到低)，CTSIF 位均置 1，LPUART_CR3 寄存器中的 CTSIE 位置 1 时会触发中断。

- 发送端 LPUART 发送器检测到 CTS 引脚低电平时，可发送数据。
- 发送端 LPUART 发送器检测到 CTS 引脚高电平时，当前正在发送的数据发送完成后停止发送。
- 发送端 LPUART 发送器检测到 CTS 引脚高电平时，正在写入发送数据寄存

器中的数据不向外发送，当发送器再次检测到 CTS 低电平时，恢复发送。

CTS 流控通信示例参见下图：

图 30-13 CTS 动作示例



注意：为保证正常工作，使能 CTS 需在当前字符结束前，至少提前 1 个 LPUART_KCLK 时钟周期内完成。此外还应注意，当 CTS 上的脉冲宽度小于 1 个 PCLK 周期时，可能导致 CTSIF 位不被置位。CTS 输入的最小脉冲宽度为 1 个 KCLK，如果 KCLK 周期小于 PCLK 时，有可能电平变化被识别但 CTSIF 标志位不被置位。

RS485 收发器控制

将 LPUART_CR3 控制寄存器中的 DEM 位置 1，可启用 RS485 收发器控制功能。此功能下由硬件通过 DE (Driver Enable) 引脚控制外部 RS485 收发器。

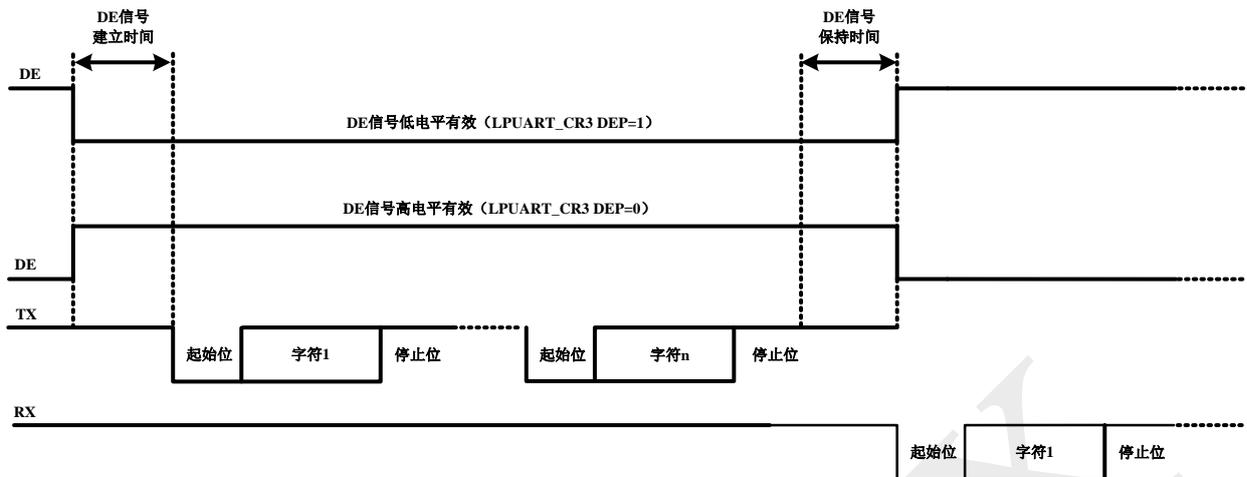
LPUART_CR1 寄存器的 DEAT[4:0]位域用于编程 DE 信号的建立时间：从使能 DE (DE 拉高) 到发送起始位之间的时间。

LPUART_CR1 寄存器的 DEDT [4:0] 位域用于编程 DE 信号的保持时间：从最后一个字符的停止位到 DE 拉低之间的时间。

DE 信号的极性可在 LPUART_CR3 寄存器中的 DEP 位配置。

DEAT[4:0]和 DEDT[4:0] 的计算方法如下，其中 $P = BRR[19:11]$ ：

- DE 信号建立时间
 - $P \neq 0$ 时，为 $(1 + (DEAT \times P)) \times f_{LPUART_KCLK}$ ；
 - $P = 0$ 时，为 $(1 + DEAT) \times f_{LPUART_KCLK}$ ；
- DE 信号保持时间
 - $P \neq 0$ 时，为 $(1 + (DEDT \times P)) \times f_{LPUART_KCLK}$ ；
 - $P = 0$ 时，为 $(1 + DEDT) \times f_{LPUART_KCLK}$ ；

图 30-14 RS485 DE 信号发送/接收控制时序


30.5 LPUART 低功耗模式

30.5.1 LPUART 对低功耗模式的支持

LPUART 支持在低功耗模式下工作，支持将芯片从低功耗模式（Stop 和 Sleep）唤醒。

选用 Stop 模式下可以工作的时钟（HSI 或 LSE）作为 LPUART 的时钟源时，即使停用 PCLK 时钟，也可以保证数据正常传输，并支持将芯片从 Stop 模式唤醒。

LPUART_WAKEUP 信号为 LPUART 发出的唤醒信号，此信号可以唤醒时钟（包括 PCLK）和 CPU（详情参见 [EXTI 章节](#)），支持将芯片从低功耗模式唤醒。可以通过使能相关中断来配置 LPUART_WAKEUP 信号的触发事件源，有以下几种情况：

- 禁止 FIFO 时，使能 PCLK 后，才可以读取 LPUART 接收数据寄存器。这种情况下 LPUART_WAKEUP 信号的触发事件可配置为 RXNE 位置 1，即接收到数据时产生唤醒信号。RXNEIE 位需在进入从低功耗模式前置 1。
- 使能 FIFO 时，此时可以使用以下事件作为 LPUART_WAKEUP 信号的触发事件：
 - 使用 RXFIFO 已满。则 RXFFIE 位需要在进入 Stop 模式前置 1，在 Stop 模式下可以接收的数据个数与 RXFIFO 总深度相同。接收至 RXFIFO 满时，由于 Stop 模式下 PCLK 停止，所以 RXFF 标志不置位。唤醒以后 PCLK 恢复，RXFF 标志置位。
 - 使用 RXFIFO 非空。则 RXFNEIE 位需在进入低功耗模式前置 1。
 - 使用 TXFIFO 为空。则 TXFEIE 位需在进入低功耗模式前置 1。

Stop 模式下收发数据时，为避免上溢/下溢错误，LPUART_WAKEUP 信号的触发事件还可以使用以下事件：

- 使用 TXFIFO 达到阈值。则 TXFTIE 位需在进入低功耗模式前置 1。
- 使用 RXFIFO 达到阈值。则 RXFTIE 位需在进入低功耗模式前置 1。

使用 FIFO 相关的中断（RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 达阈值）唤醒芯片可以尽可能延长 LPUART 在低功耗模式下的运行时间，利于优化功耗。

- 除了以上提到的事件以外，WUF 位（从低功耗模式唤醒标志）置 1 也可以作为 LPUART_WAKEUP 信号的触发事件，WUF 置 1 的条件在 WUS[1:0] 位域配置，有以下几个选项：

- 地址匹配。
- 检测到起始位。
- RXNE 置 1。

如果 WUFIE 位置 1，则触发中断。只有将 WUFIE 位置 1 时，WUF 标志置位才可以将芯片从低功耗模式唤醒。WUFIE 位需在低功耗模式前置 1。

注意：

在进入 Stop 模式之前，需确认 LPUART 无正在进行的传输。

初始化完成并使能接收 (RE=1) 后，在进入低功耗模式前，需检查 REACK 位置 1 以确认 LPUART 接收已使能。

使用 DMA 接收时，需在进入 Stop 模式前停用 DMA，并在退出低功耗模式后重新使能。

30.5.2 低功耗模式和静默模式

如果 LPUART 在进入低功耗前处于静默模式：

- 空闲帧检测无法在 Stop 模式下工作，所以进入 Stop 模式后不能使用空闲帧检测从静默模式退出。

Sleep 模式下空闲帧检测可以正常工作，支持使用空闲帧检测从静默模式退出。

- 如果退出静默模式的条件配置为地址匹配，则需在 WUS[1:0] 位域将低功耗模式的唤醒源也同样配置为地址匹配。

此种配置下，如下两种条件不能退出静默模式：

FIFO 禁止时，在进入静默模式前，RXNE 标志需清零，如果未清零，在进入低功耗模式后，当发生地址匹配时，可以将芯片从 Stop 模式唤醒，唤醒后不会退出静默模式。

FIFO 使能时，进入静默模式前需保证 RXFIFO 中至少有一个未使用空间（RXFNF=1），否则当地址匹配发生时，LPUART 无法从静默模式退出，但是地址匹配事件可以将芯片从低功耗模式下唤醒。

30.5.3 LPUART 内核时钟（LPUART_KCLK）停止时从 Stop 模式唤醒

如果在 Stop 模式下 LPUART_KCLK 时钟停止，在 LPUART 接收线上检测到下降沿时，借助 LPUART_KCLK_REQ 请求信号开启 LPUART_KCLK 时钟。

此后如果检测到了有效的唤醒事件，芯片从 Stop 模式唤醒并正常接收数据。

如果唤醒事件无效，则 LPUART_KCLK 再次关闭，芯片不被唤醒，保持在 Stop 模式下。内核时钟开启请求被释放。

以下示例展示了唤醒事件为“地址匹配检测”且不使能 FIFO 的情况。

图 30-15 检测到有效唤醒事件

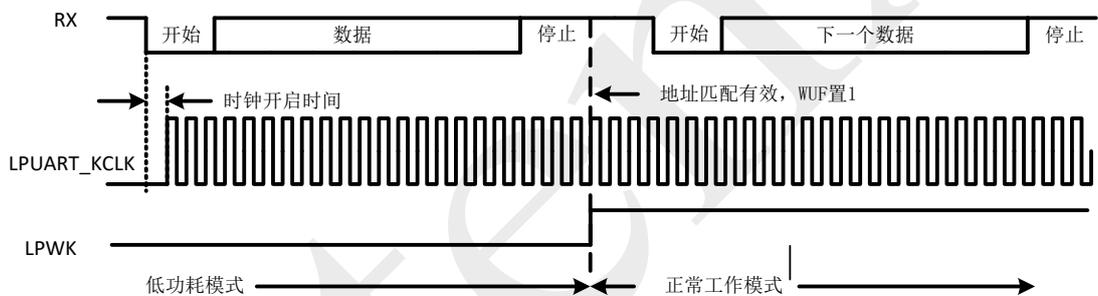
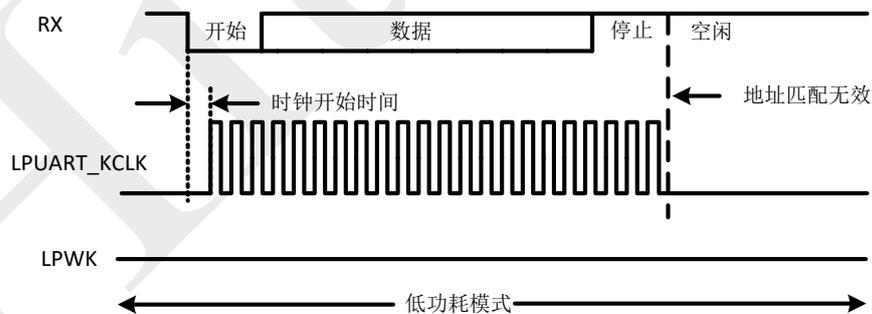


图 30-16 未检测到有效唤醒事件



注意： 唤醒事件为地址匹配或任一接收帧时，以上两图适用。

唤醒事件为起始位检测，则 LPUART 在起始位结束时发送唤醒事件。

不同低功耗模式下 LPUART 外设的状态如下表所示：

表 30-5 低功耗模式说明

模式	说明
Sleep	对 LPUART 通信无影响，LPUART 输出的唤醒请求信号或中断请求信号可使芯片退出 Sleep，参见： LPUART 中断请求表
Stop	LPUART 的寄存器仍被保持，LPUART 输出的唤醒请求信号或中断请求信号可在此模式下唤醒芯片。需将时钟源配置为 HSI

模式	说明
	或 LSE。

30.5.4 确定从低功耗模式唤醒芯片的最高波特率

从低功耗模式唤醒芯片的最大波特率取决于唤醒时间参数（参见芯片数据手册）和 LPUART 接收器容差（详情参见 [LPUART 接收容差章节](#)）。

举例说明：M 位 = “00”，STOP[1:0]= “10”（8bits 字符长度，2 位停止位）。

在上述条件下，根据 [LPUART 接收容差表](#) LPUART 接收器容差为 2.08%。

$DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR + DWKUP < LPUART$ 接收器的容差

$$DWKUP_{max} = t_{WULPUART} / (11 \times T_{bit\ Min})$$

$$T_{bit\ Min} = t_{WULPUART} / (11 \times DWKUP_{max})$$

其中 $t_{WULPUART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTR_ERR、DQU_ERR、DRX_OSERR 和 DL_ERR 为 0%，则 DWKUP 的最大值为 2.08%。实际上，计算时至少需要考虑 LPUART_KCLK 时钟源的精度。

例如，如果将 HSI 用作 LPUART_KCLK，HSI 的精确度为 2%：

$$t_{WULPUART} = 3\mu s \text{（仅供参考，正确参数请参见数据手册）。}$$

$$DWKUP_{max} = 2.08\% - 2\% = 0.8\%$$

$$T_{bit\ Min} = 3\mu s / (11 \times 1.41\%) = 34.09 \mu s$$

得到从低功耗唤醒时支持的最大波特率为： $1/34.09\mu s = 29.33\text{Kbauds}$

30.5.5 LPUART 中断请求表

参见下表 32-5

表 30-6 LPUART 中断请求表

中断标志	触发事件	中断使能位	中断清除方法	输出信号	
				LPUART_IRQ	LPUART_WAKEUP
TXE	发送数据寄存器空	TXEIE	向发送数据寄存器写入数据	√	×
TXFNF	发送 FIFO 未滿	TXFNFIE	发送 FIFO 滿时清零	√	×

中断标志	触发事件	中断使能位	中断清除方法	输出信号	
				LPUART_IRQ	LPUART_WAKEUP
TXFE	发送 FIFO 空	TXFEIE	发送 FIFO 中包含至少一个数据时自动清零，或 TXFRQ 位写 1。	√	√
TXFT	达到发送 FIFO 阈值	TXFTIE	发送 FIFO 中数据数量小于阈值时自动清除	√	√
RXNE	接收数据寄存器非空	RXNEIE	读取接收数据寄存器，或 RXFRQ 位写 1	√	√
RXFNE	接收 FIFO 非空	RXFNEIE	读取数据直到接收 FIFO 空，或 RXFRQ 位写 1	√	√
RXFF	接收 FIFO 满	RXFFIE	至少读取一个数据，或 RXFRQ 位写 1	√	√
RXFT	达到接收 FIFO 阈值	RXFTIE	读取接收 FIFO 中数据直到少于阈值或 RXFRQ 位写 1	√	√
TC	发送数据寄存器空且移位寄存器空	TCIE	向发送数据寄存器写入数据或 TCCF 位写 1	√	×
CTSIF	CTS 电平发生变化	CTSIE	CTSCF 位写 1	√	×
IDLE	总线空闲检测	IDLEIE	IDLECF 位写 1	√	×
PE	奇偶校验错误	PEIE	PECF 位写 1	√	×
NOISE	检测到噪声	EIE	NOISECF 写 1	√	×
FE	检测到帧错误		FECF 位写 1	√	×

中断标志	触发事件	中断使能位	中断清除方法	输出信号	
				LPUART_IRQ	LPUART_WAKEUP
ORE	接收溢出错误		ORECF 位写 1	√	×
CMF	检测到字符匹配	CMIE	CMCF 位写 1	√	×
WUF	检测到唤醒事件	WUFIE	WUF 通过将 WUCF 位置“1”来清零。	√	√

30.6 LPUART 寄存器

每个外设寄存器为 32 位，支持以字（Word）为单位访问。

30.6.1 LPUART 控制寄存器 1 (LPUART_CR1) -- (FIFO 使能)

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
RXFFIE	TXFEIE	FIFOEN	M1	Res	DEAT[4:0]				DEDT[4:0]						
rw	rw	rw	rw		rw	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
Res	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	RXFFIE	RXFIFO已满中断使能 0：禁止中断 1：使能中断
30	TXFEIE	TXFIFO为空中断使能 0：禁止中断 1：使能中断
29	FIFOEN	FIFO使能 0：禁止 1：使能 在禁止LPUART(UE=0)时才能写入此位域。
28	M1	字符长度 此位和bit 12(M0)用于配置字符长度。 M[1:0] = 00：8位数据位 M[1:0] = 01：9位数据位 M[1:0] = 10：7位数据位

		M[1:0] = 11: 配置无效, 此位域的值不发生变化。 在禁止LPUART(UE=0)时才能写入此位域。
27:26	保留	写入无效。
25:21	DEAT[4:0]	RS485模式下, DE信号的建立时间 用于配置RS485收发器使能信号DE有效(拉高), 与发送起始位之间的时间。 在禁止LPUART(UE=0)时才能写入此位域。 计算方法参见 RS485收发器控制章节
20:16	DEDT[4:0]	RS485模式下, 禁止驱动器时DE信号的保持时间 从最后一帧的停止位到DE失效(拉低)之间的时间。 如果在DEDT期间内在LPUART_TDR寄存器写入新数据, 则新数据仅在DE再次使能后才会发送, 即新数据需经过当前DEDT和下一次DE使能后的DEAT后才会发送。 在禁止LPUART(UE=0)时才能写入此位域。 计算方法参见 RS485收发器控制章节
15	保留	写入无效。
14	CMIE	字符匹配中断使能 0: 禁止中断 1: 使能中断
13	MME	静默模式使能 用于使能LPUART静默模式, 不同配置下静默模式的进入和退出条件参见: 多机通信 章节。 0: 接收器仅工作在运行模式 1: 允许接收器在静默模式和运行模式间切换
12	M0	字符长度 此位和位28(M1)用于设置字符长度。详情参见M1的说明 在禁止LPUART(UE=0)时才能写入此位
11	WAKE	从静默模式退出方式选择 0: 空闲帧检测 1: 地址匹配 在禁止LPUART(UE=0)时才能写入此位。
10	PCE	奇偶校验控制使能 此位选择硬件奇偶校验控制, 包括发送时生成校验位和接收时对校验位的检测。使能奇偶校验时, 校验位位于数据寄存器的MSB(参见: 帧格式示意图)。 此位置1后, 奇偶校验从下一字节生效。

		<p>0: 禁止奇偶校验</p> <p>1: 使能奇偶校验</p> <p>在禁止LPUART(UE=0)时才能写入此位域。</p> <p><i>注意: 使能奇偶校验, 占用1个数据位。</i></p>
9	PS	<p>奇偶校验选择</p> <p>此位用于在奇偶校验控制使能时 (PCE位置1) 时, 选择奇校验或偶校验。</p> <p>此位选择后从下一字节开始生效。</p> <p>0: 偶校验</p> <p>1: 奇校验</p> <p>在禁止LPUART(UE=0)时才能写入此位域。</p>
8	PEIE	<p>奇偶校验错误中断使能</p> <p>0: 禁止中断</p> <p>1: 使能中断</p>
7	TXFNIE	<p>TXFIFO非空中断使能</p> <p>0: 禁止中断</p> <p>1: 使能中断</p>
6	TCIE	<p>传输完成中断使能</p> <p>0: 禁止中断</p> <p>1: 使能中断</p>
5	RXFNEIE	<p>RXFIFO非空中断使能</p> <p>0: 禁止中断</p> <p>1: 使能中断</p>
4	IDLEIE	<p>空闲帧检测中断使能</p> <p>0: 禁止中断</p> <p>1: 使能中断</p>
3	TE	<p>发送器使能</p> <p>0: 禁止发送器</p> <p>1: 使能发送器</p> <p><i>注意: 传送期间TE 位上的“0”脉冲 (将TE 清零后再置1), 会在当前字节发送完毕后发送一个空闲帧。为了空闲帧可以正确发送, TE 清零后需要间隔一段时间才可以再写入1, 可由软件查询LPUART_ISR 寄存器中的TEACK 位清零来确认有效间隔时间。</i></p>
2	RE	<p>接收器使能</p> <p>0: 禁止接收</p> <p>1: 使能接收</p>

- 1 UESM LPUART 在低功耗模式下的唤醒功能使能控制
 0: LPUART 不能将芯片从低功耗模式唤醒
 1: LPUART 能够将芯片从低功耗模式唤醒, 从停止模式唤醒时, 需将 LPUART_KCLK 时钟配置为 HSI 或 LXTAL。
 注意: 建议在进入停止模式前将 UESM 位置 1, 并在退出停止模式时将其清零。
- 0 UE LPUART 使能
 此位清零后, LPUART 预分频器立即停止, 当前工作丢弃; LPUART 配置保留; TXFIFO、RXFIFO、LPUART_TDR 以及 LPUART_RDR 被清空; LPUART_ISR 的所有状态标志均被复位。
 0: 禁止
 1: 使能
 注意: 为避免传输错误, 禁止 LPUART 之前需要将 TE 位清零, 在 TE 位清零前, 为确认传输完成, 需检查 LPUART_ISR 中的 TC 位置 1 后再将 TE 位清零。
 UE = 0 时会同时复位 DMA 请求, 因此清零 UE 前还需禁止 DMA 通道。

30.6.2 LPUART 控制寄存器 1 (LPUART_CR1) -- (FIFO 禁止)

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
Res.	FIFO ON	M1	Res	Res	DEAT[4:0]						DEDT[4:0]				
	rw	rw	-	-	rw	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
Res.	MIE	MME	M0	WAKE	PCE	PS	PEIE	TXE IE	TCIE	RXNE IE	IDLE IE	TE	RE	UESM	UE
-	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31: 30	保留	写入无效。
29	FIFOEN	FIFO使能 0: 禁止 1: 使能 在禁止LPUART(UE=0)时才能写入此位域。
28	M1	字符长度 此位和bit 12(M0)用于配置字符长度。 M[1:0] = 00: 8位数据位 M[1:0] = 01: 9位数据位

		M[1:0] = 10: 7位数据位 M[1:0] = 11: 配置无效, 此位域的值不发生变化。 在禁止LPUART(UE=0)时才能写入此位域。
27: 26	保留	写入无效。
25: 21	DEAT[4:0]	RS485模式下, DE信号的建立时间 用于配置RS485收发器使能信号DE有效(拉高), 与发送起始位之间的时间。 在禁止LPUART(UE=0)时才能写入此位域。 计算方法参见 RS485收发器控制章节
20: 16	DEDT[4:0]	RS485模式下, 禁止驱动器时DE信号的保持时间 从最后一帧的停止位到DE失效(拉低)之间的时间。 DE信号的极性在LPUART_CR3寄存器中的DEP位可配置。 如果在DEDT期间内在LPUART_TDR寄存器写入新数据, 则新数据仅在经过DEDT和DEAT后才会发送。 在禁止LPUART(UE=0)时才能写入此位域。 计算方法参见 RS485收发器控制章节
15	保留	写入无效
14	CMIE	字符匹配中断使能 0: 禁止中断 1: 使能中断
13	MME	静默模式使能 用于使能LPUART进入静默模式, 此位置1时, LPUART按WAKE位配置的方式在运行模式与静默模式之间切换。 0: 接收器仅工作在运行模式 1: 允许接收器在静默模式和运行模式间切换
12	M0	字符长度 此位和位28(M1)用于设置字符长度。详情参见M1的说明 在禁止LPUART(UE=0)时才能写入此位。
11	WAKE	从静默模式退出方式选择 0: 空闲帧检测 1: 地址匹配 在禁止LPUART(UE=0)时才能写入此位。
10	PCE	奇偶校验控制使能 此位选择硬件奇偶校验控制, 包括发送时生成校验位和接收时对校验位的检测。使能奇偶校验时, 校验位位于数据寄存器的MSB(参见: 字符组成示意图)。

		<p>此位置1后，奇偶校验从下一字节生效。</p> <p>0：禁止奇偶校验</p> <p>1：使能奇偶校验</p> <p>在禁止LPUART(UE=0)时才能写入此位。</p> <p><i>注意：使能奇偶校验，占用1个数据位。</i></p>
9	PS	<p>奇偶校验选择</p> <p>此位用于在奇偶校验控制使能时（PCE位置1）时，选择奇校验或偶校验。</p> <p>此位选择后从下一字节开始生效。</p> <p>0：偶校验</p> <p>1：奇校验</p> <p>在禁止LPUART(UE=0)时才能写入此位。</p>
8	PEIE	<p>奇偶校验错误中断使能</p> <p>0：禁止中断</p> <p>1：使能中断</p>
7	TXEIE	<p>数据寄存器空中断使能</p> <p>0：禁止中断</p> <p>1：使能中断</p>
6	TCIE	<p>传输完成中断使能</p> <p>0：禁止中断</p> <p>1：使能中断</p>
5	RXNEIE	<p>接收数据非空中断使能</p> <p>0：禁止中断</p> <p>1：使能中断</p>
4	IDLEIE	<p>空闲帧检测中断使能</p> <p>0：禁止中断</p> <p>1：使能中断</p>
3	TE	<p>发送器使能</p> <p>0：禁止发送器</p> <p>1：使能发送器</p> <p><i>注意：传送期间TE位上的“0”脉冲（将TE清零后再置1），会在当前字节发送完毕后发送一个空闲帧。为了空闲帧可以正确发送，TE清零后需要间隔一段时间才可以再写入1，可由软件查询LPUART_ISR寄存器中的TEACK位清零来确认有效间隔时间。</i></p>
2	RE	<p>接收器使能</p> <p>0：禁止接收</p>

1: 使能接收

1 UESM

LPUART 在低功耗模式下的唤醒功能使能控制

0: LPUART 不能将芯片从低功耗模式唤醒

1: LPUART 能够将芯片从低功耗模式唤醒, 从停止模式唤醒时, 需将 LPUART_KCLK 时钟配置为 HSI 或 LXTAL。

注意: 建议在进入停止模式前将 UEWK 位置 1, 并在退出停止模式时将其清零。

0 UE

LPUART 使能

此位清零后, LPUART 预分频器立即停止, 当前工作丢弃; LPUART 配置保留; TXFIFO、RXFIFO、LPUART_TDR 以及 LPUART_RDR 被清空; LPUART_ISR 的所有状态标志均被复位。

0: 禁止

1: 使能

注意: 为避免传输错误, 禁止 LPUART 之前需要将 TE 位清零, 在 TE 位清零前, 为确认传输完成, 需检查 LPUART_ISR 中的 TC 位置 1 后再将 TE 位清零。

UE = 0 时会同时复位 DMA 请求, 因此清零 UE 前还需禁止 DMA 通道。

30.6.3 LPUART 控制寄存器 2 (LPUART_CR2)

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	
ADD[7:0]								Res	Res	Res	Res	MSB FIRST	DATA INV	TXIN V	RXIN V	
rw	rw	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	
SWAP	Res.	STOP[1:0]		Res.							ADD M7	Res.				
rw		rw	rw	-	-	-	-	-	-	-	rw	-	-	-	-	

位/位域

名称

描述

31:28

ADD[7:4]

LPUART本地节点地址

此位域用于指定LPUART本地节点的地址或需要识别的字符。适用于多机通信时, 通过7位地址匹配从静默模式或低功耗模式下唤醒。发送器发送作为地址的字符时MSB为1。此位域还可用于正常工作下的字符检测(例如, ModBus协议中的块结束检测)。这种情况下, 接收到的整个字符(8位)将与ADD[7:0]值进行比较, 如果一致则CMF置1。

仅在禁止接收(RXE=0)或停用LPUART(UE=0)时才能写入此位域。

27:24	ADD[3:0]	<p>LPUART本地节点地址</p> <p>此位域用于指定LPUART节点的地址或要识别的字符代码。此位域在多机通信时于静默模式或停止模式下使用，以通过地址标记检测进行唤醒。</p> <p>仅在禁止接收(RE = 0)或禁止LPUART(UE=0)时才能写入该位域。</p> <p>ADD[7:0]配合LPUART_CR2中的ADDM7位的设置，实现地址匹配模式。</p> <p>ADDM7: 7bit/4bit地址匹配模式选择</p> <p>ADDM7为0: 检测4bit地址 ADD[3:0]</p> <p>ADDM7为1: 检测6bit 地址ADD[5:0](7bit字长)</p> <p>检测7bit地址 ADD[6:0](8bit字长)</p> <p>检测8bit地址 ADD[7:0] (9bit字长)</p>
23:20	保留	必须保持复位值。
19	MSBFIRST	<p>MSB优先</p> <p>此位由软件置位/清零。</p> <p>在停用LPUART(UE=0)时才能更改此位域。</p> <p><i>注意: 如果不支持RS485驱动器使能功能, 该位保留并且需保持清零。</i></p>
18	DATAINV	<p>二进制数据极性取反</p> <p>此位由软件置位/清零。</p> <p>0: 按正向逻辑处理发送/接收的数据。(1=H, 0=L)</p> <p>1: 按反向逻辑处理发送/接收数的数据。(1=L, 0=H)。奇偶校验位也取反。</p> <p>在停用LPUART(UE=0)后才能写入此位域。</p>
17	TXINV	<p>TX引脚有效电平反向</p> <p>此位由软件置位/清零。</p> <p>0: TX引脚使用标准逻辑电平 ($V_{DD} = 1, GND = 0$)</p> <p>1: TX引脚信号取反。($V_{DD} = 0, GND = 1$)。</p> <p>允许在TX线路上使用外部反相器。</p> <p>在停用LPUART(UE=0)后才能写入此位域。</p>
16	RXINV	<p>RX引脚有效电平反向</p> <p>此位由软件置位/清零。</p> <p>0: RX引脚信号使用标准逻辑电平 ($V_{DD} = 1, GND = 0$) 工作</p> <p>1: 对RX引脚信号取反。($V_{DD} = 0, GND = 1$)。</p> <p>允许在RX线路上使用外部反相器。</p> <p>在停用LPUART(UE=0)后才能写入此位域。</p>

- 15 SWAP 交换TX/RX引脚
此位由软件置位/清零。
0: 按标准引脚排列使用TX/RX引脚
1: TX和RX引脚交换
此功能可以方便的支持与其他LPUART的交叉连接。
在停用LPUART(UE=0)后才能写入此位域。
- 14 保留 必须保持复位值。
- 13: 12 STOP[1:0] 停止位
此位域用于编程停止位。
00: 1个停止位
01: 保留
10: 2个停止位
11: 保留
在停用LPUART(UE=0)后才能写入此位域。
- 11:5 保留 必须保持复位值。
- 4 ADDM7 7位地址检测/4位地址检测选择
用于选择4位地址检测或7位地址检测。
0: 4位地址检测
1: 7位地址检测（在8位数据模式下）
在停用LPUART(UE=0)时才能写入该位
注意：在7位和9位数据模式下，地址检测分别在6位和8位地址上完成（ADD[5:0]和ADD[7:0]）。
- 3: 0 保留 必须保持复位值。

30.6.4 LPUART 控制寄存器 3 (LPUART_CR3)

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	
TXFTCFG[2:0]						RXFTIE		RXFTCFG[2:0]						Res.		TXFTIE		WUFIE		WUS[1:0]				Res.											
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	rw		
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0					
DEP		DEM		DDRE		OVRDIS		Res.		CTSIE		CTSE		RTSE		DMAT		DMAR		Res.				HDSEL		Res.		Res.		EIE					
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	rw		

位/位域	名称	描述
31:29	TXFTCFG[2:0]	发送FIFO阈值配置 000: 发送FIFO未使用空间达到其深度的1/8

		001: 发送FIFO未使用空间达到其深度的1/4
		010: 发送FIFO未使用空间达到其深度的1/2
		011: 发送FIFO未使用空间达到其深度的3/4
		100: 发送FIFO未使用空间达到其深度的7/8
		101: 发送FIFO全部空间未使用
		其它: 保留
		在禁止LPUART(UE=0)时才能写入此位域。
		例: 发送FIFO总深度为8, 则TXFTCFG[2:0]=“010”时, 当TXFIFO中有4个未使用空间时达到阈值, TXFT=1, 如果中断使能 (TXFTIE=1), 则触发中断。
28	RXFTIE	接收FIFO阈值中断使能
		0: 禁止接收FIFO阈值中断
		1: 使能接收FIFO阈值中断
27: 25	RXFTCFG[2:0]	接收 FIFO 阈值配置
		000: 接收 FIFO 已使用空间达到其深度的 1/8
		001: 接收 FIFO 已使用空间达到其深度的 1/4
		010: 接收 FIFO 已使用空间达到其深度的 1/2
		011: 接收 FIFO 已使用空间达到其深度的 3/4
		100: 接收 FIFO 已使用空间达到其深度的 7/8
		101: 接收 FIFO 全部空间已使用
		其它: 保留
		在禁止LPUART(UE=0)时才能写入此位域。
		例: 接收FIFO总深度为8, 则RXFTCFG[2:0]位域编程为“010”时, 接收到4个数据时认为达到阈值, 此时RXFIFO 中存有 (8*0.5-1=3) 3个数据, LPUART_RDR 中存有 1 个数据。达到接收阈值时RXFT 标志置1, 如果中断使能 (RXFTIE=1), 则触发中断。
24	保留	写入无效。
23	TXFTIE	发送 FIFO 阈值中断使能
		0: 禁止发送 FIFO 阈值中断
		1: 使能发送FIFO阈值中断
22	WUFIE	从低功耗模式唤醒信号中断使能
		0: 检测到唤醒事件, 即LPUART_ISR寄存器中的WUF=1时, 不触发中断
		1: 检测到唤醒事件, 即LPUART_ISR寄存器中的WUF=1时, 触发中断
		<i>注意: WUFIE 需在进入低功耗模式前置1。</i>
21:20	WUS[1:0]	从低功耗模式唤醒信号源选择

		此位域用于选择将WUF位置位的事件
		00: 地址匹配时, WUF位置1 (按ADD[7:0]和ADDM7所定义)
		01: 保留
		10: 检测到起始位时, WUF位置1
		11: RXNE (FIFO使能时为RXFNE) 置1时, WUF位置1在禁止LPUART(UE=0)时才能写入此位域。
19:16	保留	写入无效
15	DEP	RS485收发器使能信号极性选择 0: DE信号高电平有效 1: DE信号低电平有效 在禁止LPUART(UE=0)时才能写入此位。
14	DEM	RS485收发器使能控制 0: 禁止DE功能 1: 使能DE功能 在禁止LPUART(UE=0)时才能写入此位。
13	DDRE	接收出错时禁止DMA 0: 接收出错时不禁DMA 相应错误标志置1, 但RXNE (使能FIFO时为RXFNE) 保持为0, 所以不产生DMA请求, 同时避免了上溢, 错误数据不会被接收, 但是下一个正确数据可以被接收。 1: 接收出错后禁止DMA 相应的错误标志以及RXNE均置1。不产生DMA请求, 直到错误标志清零。这意味着软件必须首先禁止DMA请求 (DMAR = 0) 或者将RXNE (使能FIFO时为RXFNE) 清零, 然后再将错误标志清零才可以继续接收。 在禁止LPUART(UE=0)时才能写入此位。 <i>注意: 接收错误包括: 奇偶校验错误、帧错误或噪声错误。</i>
12	OVRDIS	禁止接收上溢 0: 使能上溢检测 禁止FIFO时, 接收数据未读取 (RXNE=1) 时, 又接收到新数据时, 上溢错误标志位ORE置 1。 使能FIFO时, 接收FIFO已满 (RXFF=1) 时, 又接收到新数据时, 上溢错误标志位ORE置 1。 1: 禁止上溢功能, ORE标志不会置1 在禁止LPUART(UE=0)时才能写入此位。
11	保留	写入无效。
10	CTSIE	CTS中断使能

		0: 禁止中断 1: 使能中断
9	CTSE	CTS使能 0: 禁止CTS硬件流控功能 1: 使能CTS硬件流控功能，仅当CTS输入有效（低电平）时才允许数据发送。如果在发送数据时，CTS输入为高电平，则在完成当前发送后停止发送。如果CTS拉高时数据已写入发送数据寄存器，则此数据将延迟发送，直到CTS重新有效时才开始发送。 在禁止LPUART(UE=0)时才能写入此位。
8	RTSE	RTS使能 0: 禁止RTS硬件流控功能 1: 使能RTS输出，LPUART可以接收数据时，RTS输出为低电平，不能继续接收数据时，RTS输出为高电平 在禁止LPUART(UE=0)时才能写入此位域。
7	DMAT	使能DMA发送请求 0: 禁止 1: 使能
6	DMAR	使能DMA接收请求 0: 禁止 1: 使能
5: 4	保留	写入无效。
3	HDSEL	使能单线半双工 0: 禁止单线半双工模式 1: 使能单线半双工模式 在禁止LPUART(UE=0)时才能写入此位。
2:1	保留	写入无效。
0	EIE	错误中断使能 如果发生帧错误、上溢错误或噪声错误时（LPUART_ISR寄存器中FE = 1、ORE = 1或NE = 1）时，此位置1会触发中断。 0: 禁止中断 1: 使能中断

30.6.5 LPUART 波特率分频寄存器 (LPUART_BRR)

偏移地址: 0x0C

复位值：0x0000 0000

注意：在禁止 LPUART(UE=0)时修改此寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												BRR[19:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值
19:0	BRR[19:0]	<p>注意：禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。</p> <p>如果 LPUART_BRR 必须 $\geq 0x300$ 且 LPUART_BRR 为 20 位,则使用高 fck 值生成高波特率时应十分谨慎.fck 必须在 [3 x 波特率到 4096 x 波特率] 的范围内。</p>

30.6.6 LPUART 请求寄存器 (LPUART_RQR)

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.												TX FRQ	RX FRQ	MMRQ	SBKRQ	Res.
												w	w	w	w	-

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFRQ	<p>发送数据清空请求，</p> <p>禁止 FIFO 时，向此位写 1 会清空发送数据，TXE 标志置 1。</p> <p>使能 FIFO 时，向此位写入 1 会清空 TXFIFO，TXFIFO 清空后，TXNF 标志置 1。</p> <p>此操作会丢弃发送数据，使能 FIFO 时会清空发送 FIFO。</p>
3	RXFRQ	<p>接收数据清空请求</p> <p>向该位写 1 时清空接收数据，未被读取的数据会被丢弃，可用于避免上溢。</p> <p>禁止 FIFO 时，RXNE 标志清零。</p> <p>使能 FIFO 时，RXFIFO 被清空，RXFNE 标志清零。</p>

		<p>当接收到 RXFIFO 大小+1 个数据时， (即 RXFIFO 大小+LPUART_RDR 寄存器中的 1 个数据)，此位置 1。 如果 LPUART_CR1 寄存器中 RXFFIE 位置 1，则会触发中断。</p> <p>0: RXFIFO 未满 1: RXFIFO 已满</p>
23	TXFE	<p>TXFIFO 空标志 当 TXFIFO 为空时，此位置 1。 当 TXFIFO 中至少有一个数据时，此标志自动清零。 也可以通过向 LPUART_RQR 寄存器中的 TXFRQ 写 1 将 TXFE 标志置 1。 如果 LPUART_CR1 寄存器中的 TXFEIE 位置 1，则会触发中断。</p> <p>0: TXFIFO 非空 1: TXFIFO 为空</p>
22	REACK	<p>接收使能确认标志 0: 接收器未准备好 1: 接收器准备完成，可以接收数据</p>
21	TEACK	<p>发送使能确认标志: TE 清零时，发送器禁止完成后此位清零；当发送器已准备好发送时，此位置 1。 0: 发送器未准备好， 1: 发送器准备完成 <i>注意：TEACK 为 0 时才可以将 TE 置 1。向 TE 写 0 后检查此位清零，再将 TE 置 1，此时可以实现以最快的方式发送空闲帧。</i></p>
20	WUF	<p>从低功耗模式唤醒标志 当检测以下到唤醒事件时，此位置 1。 如果 LPUART_CR3 寄存器中 WUFIE 位置 1，则会触发中断。 相应事件在 WUS[1:0]位域中选择，有以下几个选项：</p> <ul style="list-style-type: none"> - 地址匹配时 - 检测到起始位时 - RXNE 置 1 时 <p>通过向 LPUART_ICR 寄存器中的 WUCF 写 1，此位清零。 <i>注意：当 UESM 清零时，WUF 标志也清零。</i></p>
19	RWU	<p>静默模式状态指示位 该位指示 LPUART 是否处于静默模式。 当从静默模式退出时此位清零，处于静默时，此位为 1。</p>

从静默模式退出的方式有地址匹配和空闲帧检测两种，在 LPUART_CR1 寄存器中的 WAKE 位可配置。

0: 处于运行模式
1: 处于静默模式

18	SBKF	<p>中断帧发送标志</p> <p>此位表示已发送中断帧。</p> <p>向写入 LPUART_RQR 寄存器中的 SBKRQ 位写 1，产生发送中断帧请求时，此位置 1。</p> <p>此位在中断帧的停止位期间自动清零。</p> <p>0: 中断帧已发送 1: 有中断帧发送请求</p>
17	CMF	<p>字符匹配标志</p> <p>接收到由 ADD[7:0] 定义的字符时此位由硬件置 1。</p> <p>如果 LPUART_CR1 寄存器中 CMIE=1 位置 1，则会触发中断。</p> <p>通过向 LPUART_ICR 寄存器中的 CMCF 写 1 清零。</p> <p>0: 未检测到字符匹配 1: 检测到字符匹配</p>
16	BUSY	<p>忙标志</p> <p>当 RX 线路有接收的数据时，从检测到起始位开始，此位置 1。</p> <p>在接收结束时，无论是否成功，此位均清零。</p> <p>0: LPUART 处于空闲状态 1: 正在接收数据</p>
15: 11	保留	写入无效。
10	CTS	<p>CTS 标志</p> <p>此位状态是对 CTS 输入引脚的状态取反</p> <p>0: CTS 引脚为高电平 1: CTS 引脚为低电平</p>
9	CTSIF	<p>CTS 中断标志</p> <p>在使能 CTS，即 CTSE 位置 1 时，当 CTS 输入出现电平翻转时，此位置 1。</p> <p>如果 LPUART_CR3 寄存器中 CTSIE 位置 1，则会触发中断。</p> <p>通过向 LPUART_ICR 寄存器中的 CTSCF 位写 1，此位清零。</p> <p>0: CTS 引脚电平未发生变化 1: CTS 引脚电平发生变化</p>
8	保留	写入无效。
7	TXFNF	<p>TXFIFO 未滿</p> <p>此位在 TXFIFO 未滿时自动置 1，表示可以向 LPUART_TDR 中</p>

- 写入数据。
 如果 LPUART_CR1 寄存器中 RXFFIE 位置 1，则会触发中断。
 如果中断使能，即 LPUART_CR1 寄存器中 TXFNFIE 位置 1，
 则触发中断。
 当 TXFIFO 已满时，此标志清零，表示不能向 LPUART_TDR 中
 写入数据。
 0: TXFIFO 已满。
 1: TXFIFO 未满。
*注意：在发送清空发送数据请求 (TXFRQ=1) 后，TXFNF 不会
 立即置 1，此位先保持为 0，直到 TXFIFO 被清空后和
 TXFE 同时置 1。*
- 6 TC 发送完成标志
 此标志位置 1，表示写入到 LPUART_TDR 中的最后一个数据已
 完成发送，且 TXFF 置 1，则此位置 1。
 如果 LPUART_CR1 寄存器中 TCIE 位置 1 时，则会触发中断。
 通过向 LPUART_ICR 寄存器中的 TCCF 写 1，或写
 LPUART_TDR 寄存器时，TC 位清零。
 0: 发送未完成
 1: 发送已完成
*注意：如果 TE 位被清零，且当前无发送进行时，TC 位会立即
 置 1。*
- 5 RXFNE RXFIFO 非空
 RXFIFO 非空时，RXFNE 位置 1，表示已接收到数据。每次读
 LPUART_RDR 都会从 RXFIFO 中获取一个数据。
 如果 LPUART_CR1 寄存器中 RXFFIE 位置 1，则会触发中断。
 RXFNE 在 RXFIFO 为空时清零。
 也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 位写 1，通过清
 空接收数据的方式将 RXFNE 标志清零。
 0: 接收 FIFO 为空，未接收到数据
 1: 接收 FIFO 非空，已接收到数据
- 4 IDLE 检测到空闲帧
 检测到空闲帧时，此位置 1。
 如果 LPUART_CR1 寄存器中 IDLEIE 位置 1，则会触发中断。
 向 LPUART_ICR 寄存器中的 IDLECF 写 1，此位清零。
 0: 未检测到空闲帧
 1: 检测到空闲帧
*注意：清空此标志位以后，只有 RXFNE 位置 1 后，IDLE 位才会
 被再次置 1。
 使能静默模式 (MME=1) 后，如果 LPUART 未处于静默模
 式 (RWU=0)，则 IDLE 可以置 1，如果处于静默模式
 (RWU=1)，IDLE 标志位不置 1。*

3	ORE	<p>上溢错误标志</p> <p>在 RXFIFO 已满时 (RXFF=1) 又接收到新的数据, 则此位置 1。如果 LPUART_CR1 寄存器中 RXFNEIE=1 或 EIE=1, 则会触发中断。</p> <p>通过向 LPUART_ICR 寄存器中的 ORECF 写入 1, 此位清零。</p> <p>0: 无上溢错误 1: 检测到上溢错误</p> <p><i>注意: 当此位置1时, RDR 寄存器的内容不会丢失, 但移位寄存器会被覆盖。</i></p> <p><i>LPUART_CR3 寄存器中的OVRDIS位置1时, 上溢错误检测被禁止, 则此位清零 (无上溢检测)。</i></p>
2	NE	<p>噪声检测标志</p> <p>当接收中检测到噪声时, 此位自动置 1。</p> <p>如果 LPUART_CR1 寄存器中 EIE 位置 1, 则会触发中断。向 LPUART_ICR 寄存器中的 NECF 写入 1, 此位清零。</p> <p>0: 未检测到噪声 1: 检测到噪声</p> <p><i>注意: 如果此位与RXFNE位同时置1, 由RXFNE产生中断, NE标志不再产生中断。</i></p> <p><i>当禁止RXFNE中断时, EIE位置1后, NE 位置1时会触发中断。</i></p>
1	FE	<p>帧错误标志</p> <p>接收数据时, 如果在预期时间内未能识别出停止位, 则检测为帧错误, 此位置 1。接收到中断帧, 同步失败或噪声过大时可能产生此错误。</p> <p>如果 LPUART_CR1 寄存器中 EIE 位置 1, 则会触发中断。向 LPUART_ICR 寄存器中的 FECF 写 1, 此位清零。</p> <p>0: 未检测到帧错误 1: 检测到帧错误或中断帧</p>
0	PE	<p>奇偶校验错误标志</p> <p>当在接收时发生奇偶校验错误时, 此位置 1。如果 LPUART_CR1 寄存器中 PEIE 位置 1, 则会触发中断。向 LPUART_ICR 寄存器中的 PECF 写入 1, 此位清零。</p> <p>0: 无奇偶校验错误 1: 奇偶校验错误</p>

30.6.8 LPUART 中断和状态寄存器 (LPUART_ISR) -- (FIFO 禁止)

偏移地址: 0x1C

复位值: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r		r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	写入无效。
22	REACK	接收使能确认标志 0: 接收器未准备好 1: 接收器准备完成, 可以接收数据
21	TEACK	发送使能确认标志: TE 清零时, 发送器禁止完成后此位清零; 当发送器已准备好发送时, 此位置 1。 0: 发送器未准备好, 1: 发送器准备完成 <i>注意: TEACK 为 0 时才可以将 TE 置 1。向 TE 写 0 后检查此位清零, 再将 TE 置 1, 此时可以实现以最快的方式发送空闲帧。</i>
20	WUF	从低功耗模式唤醒标志 当检测以下到唤醒事件时, 此位置 1。 如果 LPUART_CR3 寄存器中 WUFIE 位置 1, 则会触发中断。 相应事件在 WUS[1:0]位域中选择, 有以下几个选项: - 地址匹配时 - 检测到起始位时 - RXNE 置 1 时 通过向 LPUART_ICR 寄存器中的 WUCF 写 1, 此位清零。 <i>注意: 当 UESM 清零时, WUF 标志也清零。</i>
19	RWU	静默模式状态指示位 该位指示 LPUART 是否处于静默模式。 当从静默模式退出时此位清零, 处于静默时, 此位为 1。 从静默模式退出的方式有地址匹配和空闲帧检测两种, 在 LPUART_CR1 寄存器中的 WAKE 位可配置。 0: 处于运行模式 1: 处于静默模式
18	SBKF	中断帧发送标志 此位表示已发送中断帧。

		<p>向写入 LPUART_RQR 寄存器中的 SBKRQ 位写 1，产生发送中断帧请求时，此位置 1。</p> <p>此位在中断帧的停止位期间自动清零。</p> <p>0: 中断帧已发送</p> <p>1: 有中断帧发送请求</p>
17	CMF	<p>字符匹配标志</p> <p>接收到由 ADD[7:0] 定义的字符时此位由硬件置 1。</p> <p>如果 LPUART_CR1 寄存器中 CMIE=1 位置 1，则会触发中断。</p> <p>通过向 LPUART_ICR 寄存器中的 CMCF 写 1 清零。</p> <p>0: 未检测到字符匹配</p> <p>1: 检测到字符匹配</p>
16	BUSY	<p>忙标志</p> <p>当 RX 线路有接收的数据时，从检测到起始位开始，此位置 1。</p> <p>在接收结束时，无论是否成功，此位均清零。</p> <p>0: LPUART 处于空闲状态</p> <p>1: 正在接收数据</p>
15:11	保留	写入无效。
10	CTS	<p>CTS 标志</p> <p>此位状态是对 CTS 输入引脚的状态取反</p> <p>0: CTS 引脚为高电平</p> <p>1: CTS 引脚为低电平</p>
9	CTSIF	<p>CTS 中断标志</p> <p>在使能 CTS，即 CTSE 位置 1 时，当 CTS 输入出现电平翻转时，此位置 1。</p> <p>如果 LPUART_CR3 寄存器中 CTSIE 位置 1，则会触发中断。</p> <p>通过向 LPUART_ICR 寄存器中的 CTSCF 位写 1，此位清零。</p> <p>0: CTS 引脚电平未发生变化</p> <p>1: CTS 引脚电平发生变化</p>
8	保留	写入无效。
7	TXE	<p>发送数据寄存器为空</p> <p>在 LPUART_TDR 寄存器的内容已发送时，此位自动置 1，表示可以向 LPUART_TDR 中写入数据。如果 LPUART_CR1 寄存器中 TXEIE 位置 1，则触发中断。</p> <p>写入 LPUART_TDR 寄存器时，此位清零，表示不能继续向 TDR 中写入数据。还可以向 LPUART_RQR 寄存器中的 TXFRQ 写 1，通过清空发送数据的方式将 TXE 标志置 1。</p> <p>0: 发送数据寄存器 LPUART_TDR 已满</p> <p>1: 发送数据寄存器 LPUART_TDR 未滿</p>

6	TC	<p>发送完成标志</p> <p>此标志位置1，表示写入到LPUART_TDR中的最后一个数据已完成发送，且TXFF置1，则此位置1。</p> <p>如果LPUART_CR1寄存器中TCIE位置1时，则会触发中断。</p> <p>通过向LPUART_ICR寄存器中的TCCF写1，或写LPUART_TDR寄存器时，TC位清零。</p> <p>0: 发送未完成</p> <p>1: 发送已完成</p> <p><i>注意：如果TE位被清零，且当前无发送进行时，TC位会立即置1。</i></p>
5	RXNE	<p>接收数据寄存器非空</p> <p>当接收到数据时，RXNE位置1。如果LPUART_CR1寄存器中RXNEIE位置1，则会触发中断。</p> <p>读LPUART_RDR寄存器时，此位清零。</p> <p>也可以向LPUART_RQR寄存器中的RXFRQ位写1，通过清空接收数据的方式将RXNE标志位清零。</p> <p>0: 未接收到数据</p> <p>1: 已接收到数据</p>
4	IDLE	<p>检测到空闲帧</p> <p>检测到空闲帧时，此位置1。如果中断使能，即LPUART_CR1寄存器中IDLEIE=1，则会触发中断。</p> <p>向LPUART_ICR寄存器中的IDLECF写1，此位清零。</p> <p>0: 未检测到空闲帧</p> <p>1: 检测到空闲帧</p> <p><i>注意：清空此标志位以后，只有RXNE位置1后，IDLE位才会被再次置1。</i></p> <p><i>使能静默模式(MME=1)后，如果LPUART未处于静默模式(RWU=0)，则IDLE可以置1，如果处于静默模式(RWU=1)，IDLE标志位不置1。</i></p>
3	ORE	<p>上溢错误标志</p> <p>在RXFIFO已满时(RXFF=1)又接收到新的数据，则此位置1。如果LPUART_CR1寄存器中RXFNEIE=1或EIE=1，则会触发中断。</p> <p>通过向LPUART_ICR寄存器中的ORECF写入1，此位清零。</p> <p>0: 无上溢错误</p> <p>1: 检测到上溢错误</p> <p><i>注意：当此位置1时，RDR寄存器的内容不会丢失，但移位寄存器会被覆盖。</i></p> <p><i>LPUART_CR3寄存器中的OVERDIS位置1时，上溢错误检测被禁止，则此位清零(无上溢检测)。</i></p>

2	NE	<p>噪声检测标志</p> <p>当接收中检测到噪声时，此位自动置1。</p> <p>使能中断 EIE 位置 1 后，则会触发中断。</p> <p>向 LPUART_ICR 寄存器中的 NECF 写入 1，此位清零。</p> <p>0: 未检测到噪声</p> <p>1: 检测到噪声</p> <p><i>注意：如果此位与RXNE位同时置1，由RXNE产生中断，NE标志不再产生中断。</i></p>
1	FE	<p>帧错误标志</p> <p>接收数据时，如果在预期时间内未能识别出停止位，则检测为帧错误，此位置 1。接收到中断帧，同步失败或噪声过大时可能产生此错误。</p> <p>如果 LPUART_CR1 寄存器中 EIE 位置 1，则会触发中断。</p> <p>向 LPUART_ICR 寄存器中的 FECF 写 1，此位清零。</p> <p>0: 未检测到帧错误</p> <p>1: 检测到帧错误或中断帧</p>
0	PE	<p>奇偶校验错误标志</p> <p>当在接收时发生奇偶校验错误时，此位置 1。如果 LPUART_CR1 寄存器中 PEIE 位置 1，则会触发中断。</p> <p>向 LPUART_ICR 寄存器中的 PECF 写入 1，此位清零。</p> <p>0: 无奇偶校验错误</p> <p>1: 奇偶校验错误</p>

30.6.9 LPUART 中断标志清零寄存器(LPUART_ICR)

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res						
-	-	-	-	-	-	-	-	-	-	-	w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	CTSCF	Res	Res	TCCF	Res	IDLECF	ORECF	NECF	FECF	PECF
-	-	-	-	-	-	w	-	-	w	-	w	w	w	w	w

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUCF	<p>从低功耗模式唤醒标志清零</p> <p>此位写入1时，LPUARTx_ISR寄存器中WUF标志将清零。</p> <p><i>注意：如果LPUART不支持从低功耗模式唤醒，此位保留并由硬件强制清零。</i></p>

19: 18	保留	必须保持复位值。
17	CMCF	字符匹配标志清零 此位写入1时，LPUARTx_ISR寄存器中CMF标志将清零。
16:10	保留	必须保持复位值。
9	CTSCF	CTS标志清零 此位写入1时，LPUART_ISR寄存器中CTSIF标志将清零。
8:7	保留	必须保持复位值。
6	TCCF	发送完成标志清零 此位写入1时，LPUARTx_ISR寄存器中TC标志将清零。
5	保留	必须保持复位值。
4	IDLECF	空闲线路标志清零 此位写入1时，LPUARTx_ISR寄存器中IDLE标志将清零。
3	ORECF	上溢错误标志清零 此位写入1时，LPUARTx_ISR寄存器中ORE标志将清零。
2	NECF	噪声检测标志清零 此位写入1时，LPUARTx_ISR寄存器中NOISE标志将清零。
1	FECF	帧错误标志清零 此位写入1时，LPUARTx_ISR寄存器中FE标志将清零。
0	PECF	奇偶校验错误标志清零 此位写入1时，LPUARTx_ISR寄存器中PE标志将清零。

30.6.10 LPUART 接收数据寄存器(LPUART_RDR)

偏移地址: 0x24

复位值: 0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res.	Res.	Res.	Res	Res.	RDR[8:0]								
-	-	-	-	-	-	-	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDR[8:0]	接收数据

包含接收到的数据字符。

RDR寄存器在输入移位寄存器和内部总线之间提供了并行接口缓冲区。

在使能奇偶校验的时，接收数据的MSB位为接收到的奇偶校验位。

30.6.11 LPUART 发送数据寄存器(LPUARTx_TDR)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res	Res	Res.	Res.	Res.	Res	Res.	TDR[8:0]										
							rw	rw	rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDR[8:0]	<p>发送数据值 包含要发送的数据字符。</p> <p>TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口缓冲区。</p> <p>在使能奇偶校验时，即 LPUARTx_CR1 寄存器中的 PCE 位置 1 时，发送数据中写入 MSB 的值被奇偶校验位取代，此种情况下写入 MSB 的值无效。</p> <p><i>注意：只能在发送数据寄存器未满即TXE=1时写入此寄存器。</i></p>

30.6.12 LPUART 预分频器寄存器(LPUART_PRESC)

偏移地址：0x2C

复位值：0x0000 0000

注意：在禁用LPUART(UE=0)时才能更改此位域

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PRESCALER [3:0]														
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PRESCALER[3:0]	<p>时钟预分频器</p> <p>LPUART 输入时钟可通过预分频系数进行分频：</p> <p>0000：输入时钟未分频</p> <p>0001：输入时钟 2 分频</p> <p>0010：输入时钟 4 分频</p> <p>0011：输入时钟 6 分频</p> <p>0100：输入时钟 8 分频</p> <p>0101：输入时钟 10 分频</p> <p>0110：输入时钟 12 分频</p> <p>0111：输入时钟 16 分频</p> <p>1000：输入时钟 32 分频</p> <p>1001：输入时钟 64 分频</p> <p>1010：输入时钟 128 分频</p> <p>1011：输入时钟 256 分频</p> <p>其余组合：保留</p> <p><i>注意：PRESCALER 写入未定义组合时，预分频值将为“1011”，即输入时钟除以256。</i></p>

31 串行外设接口 (SPI)

31.1 简介

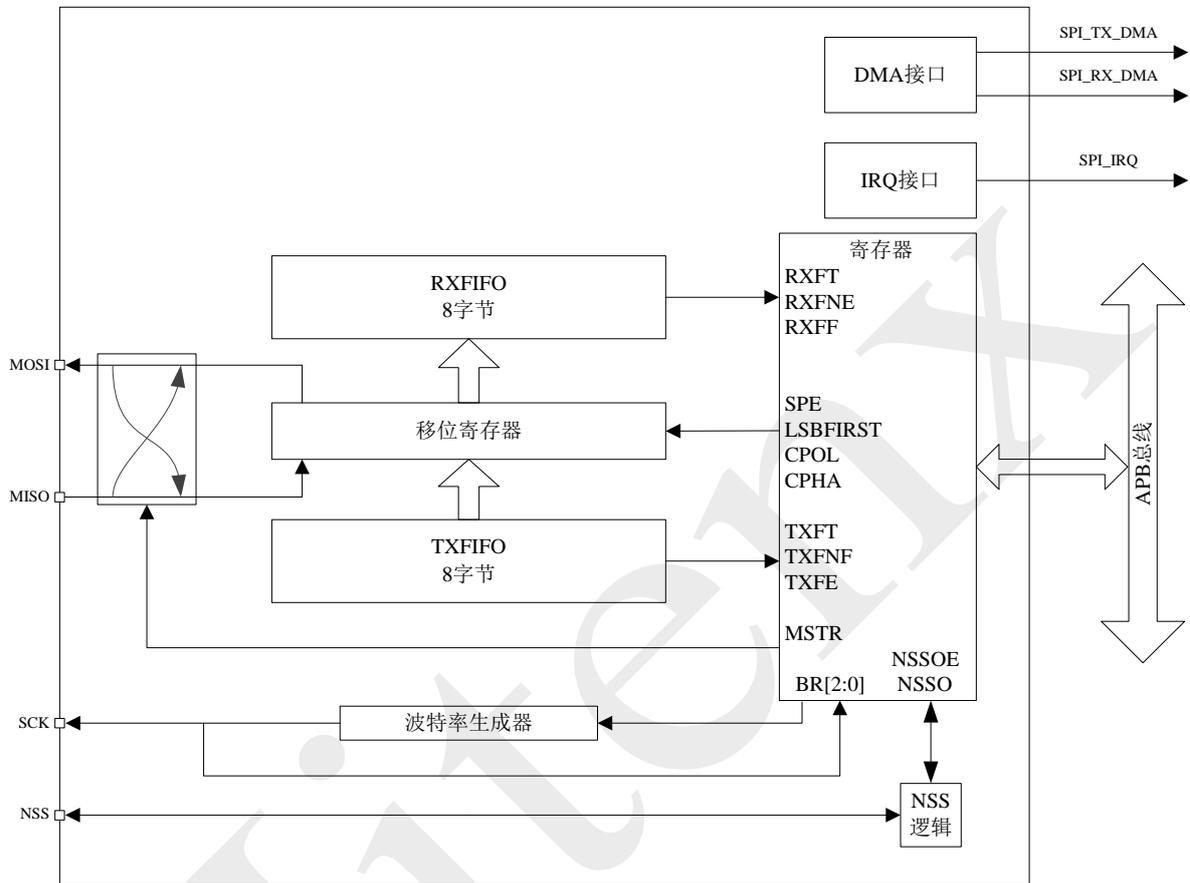
串行外设接口(SPI)协议支持与外部设备进行同步全双工串行通信。SPI 可选主机或从机模式，当配置为主机模式时，可为外部从设备提供通信时钟(SCK)。

31.2 SPI 主要特性

- 可配置为主机模式或从机模式
- 多主机模式功能
- 基于 NSS、SCK、MISO 和 MOSI 标准四线的同步全双工通信
- 帧数据大小固定为 8 bit
- 主机模式支持 7 种波特率分频，最大分频系数为 PCLK/8，最高通信速率为 6M bps
- 从机模式最大分频系数为 PCLK/8，最高通信速率为 6M bps
- 可编程时钟极性和相位
- 可编程数据传输顺序：MSB 或 LSB
- 可触发中断的错误事件：主机模式冲突、上溢、从模式故障和从模式下溢
- 支持 DMA 发送和接收
- 内置两个 8 字节接收 FIFO 和发送 FIFO

31.3 SPI 功能说明

31.3.1 SPI 框图

图 31-1 SPI 框图


31.3.2 SPI 引脚和信号

表 31-1 SPI 引脚

引脚名称	信号类型	说明
MISO	输入/输出	主入/从出数据。 主机模式：输入信号，用于接收数据； 从机模式：输出信号，用于发送数据。
MOSI	输入/输出	主出/从入数据。 主机模式：输出信号，用于发送数据； 从机模式：输入信号，用于接收数据。
SCK	输入/输出	串行时钟。 主机模式：输出信号，用于发送时钟； 从机模式：输入信号，用于接收时钟。
NSS	输入/输出	从机选择。 主机模式：输出或输入信号 - 单主机模式应设置为输出，用于从机选择；

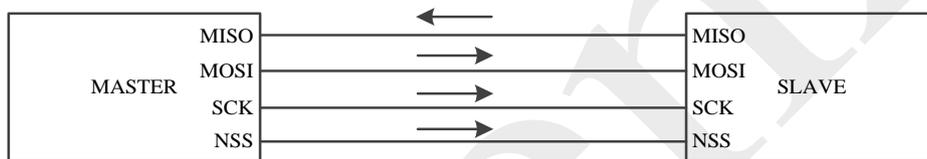
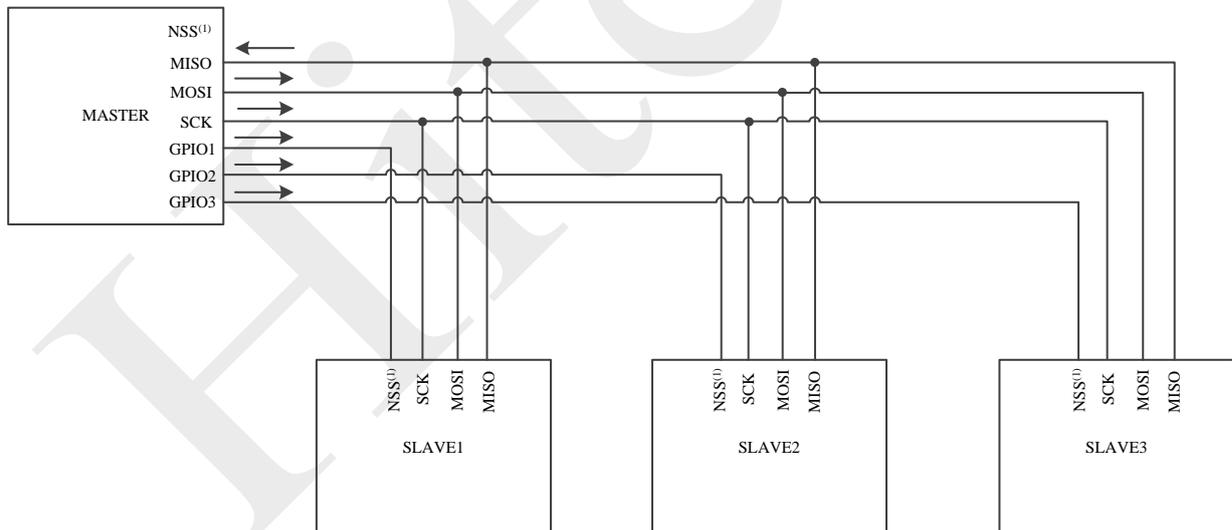
引脚名称	信号类型	说明
		- 多主机模式应设置为输入，用于多主冲突检测； 从机模式：输入信号，用于有效选择。

表 31-2 SPI 内部信号

信号名称	信号类型	说明
SPI_IRQ	输出	SPI 中断信号。
SPI_TX_DMA	输出	SPI 发送 DMA 请求信号。
SPI_RX_DMA	输出	SPI 接收 DMA 请求信号。

31.3.3 单主机通信

SPI 是同步全双工通信接口，主从设备通过 MOSI 和 MISO 两条数据线进行数据传输。单主单从和单主多从连接关系见下图。

图 31-2 单主单从应用

图 31-3 一主多从应用


1. 主机的 NSS 引脚未使用，但必须配置输出。从机的 NSS 输入由主机的 GPIO 驱动控制。

当 SPIx_CR1 寄存器的 MSTR 置 1 时，SPI 接口工作于主机模式。

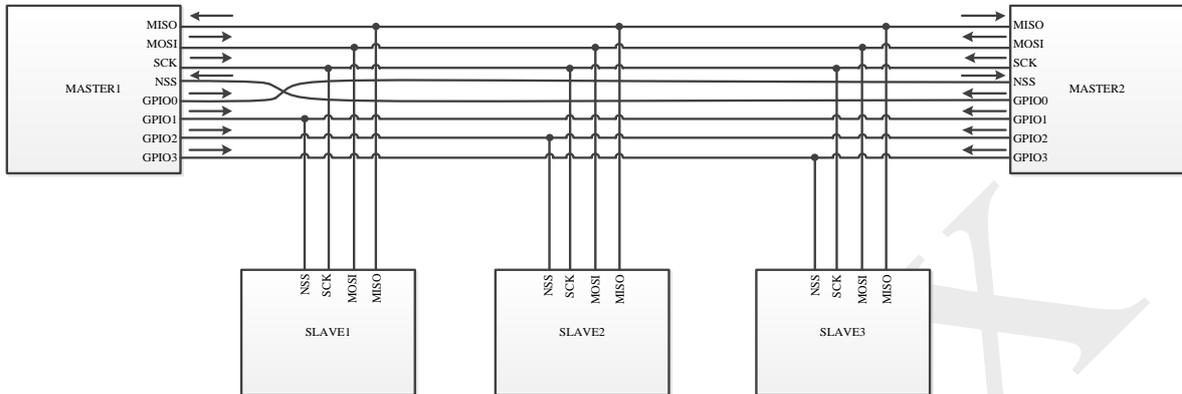
SCK 由主机提供，其波特率由 SPIx_CR1 寄存器的 BR[2:0]控制，范围为 $f_{PCLK}/2 \sim f_{PCLK}/128$ 。

NSS 使用参见：[NSS 管理](#)。

31.3.4 多主机通信

SPI 总线包含两个主机时，用户可使用主机冲突功能来检测的主机间是否存在冲突，从而实现多主机通信。多主机连接关系见下图。

图 31-4 多主多从应用



当总线上的主机设备无数据传输时，均保持为禁止状态（SPE 为 0），GPIO1~3 为输入状态。主从设备进行数据传输时，按照如下步骤操作：

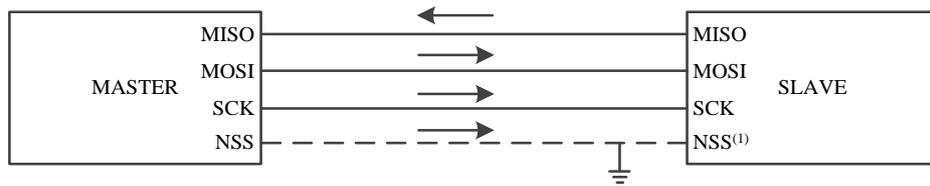
- 1) 当 MASTER1 要接管对总线的控制，将自身切换到主机模式。如果此时其 NSS 输入为低，则总线已被 MASTER2 占用，MASTER1 将会生成主机模式冲突；
- 2) MASTER1 通过 GPIO0 向 MASTER2 的 NSS 施加低电平，通知其总线已占用；
- 3) MASTER1 通过 GPIO1 输出低电平选择与 SLAVE1 进行通信；
- 4) 传输完成后，MASTER1 由主机模式切换至禁止状态；
- 5) MASTER1 通过 GPIO0 向 MASTER2 的 NSS 施加高电平，通知其总线已空闲；
- 6) MASTER1 的 GPIO1 切换至输入状态。

如果两个主机同时发出总线占用的控制请求，则会出现总线冲突（参见：[主机模式冲突](#)）。发送冲突后，用户需通过应用程序进行冲突仲裁处理，例如，通过延迟尝试请求占用总线。

除 NSS 配置不同外，多主模式配置与单主机配置相同。NSS 配置差异参见：[NSS 管理](#)。

31.3.5 从机通信

当 SPIx_CR1 寄存器的 MSTR 清 0 时，SPI 接口工作于从机模式。从机模式下，数据收发流程受控于主机时钟，从机应在时钟发出之前将发送数据写入 SPIx_DR 寄存器，以保证数据正常传输。从机应用连接关系见下图。

图 31-5 从机应用


1. 从机 NSS 输入可选择由主机提供，或者固定为低电平。

注意： 从机模式且 NSS 固定为低电平时，仅支持 CPHA=1 的通信模式。

注意： 多个从机情况下，从机 MISO 建议配置为上拉开漏模式，防止意外短路发生。

31.3.6 NSS 管理

在从机模式下，NSS 用作从机选择输入信号，被选中的从机才能够与主机进行通信。

在主机模式下，NSS 可用作输出或输入：

- 用作输出时，可以驱动单个从机的从机选择信号，用于单主机模式；
- 用作输入时，可以检测总线占用冲突，用在多主机模式；

NSS 的配置应用见下表：

表 31-3 NSS 配置⁽¹⁾

主从模式	NSS 引脚	NSSE	NSSP	说明
从机	输入	×	×	从机 NSS 为输入，用于确定从机的选择是否有效。
主机	输入	0	×	NSS 为输入信号,此时输出电平控制位 NSSP 无效。
	输出	1	0/1	NSS 为输出信号，NSS 输出电平由 NSSP 控制。

×-表示无效。

31.3.7 通信模式

SPI 通信过程中，将同时执行接收和发送操作。数据线上的信息的移位和采样与串行时钟同步。四种通信格式取决于时钟相位和时钟极性，具体见下表。

表 31-4 通信模式配置

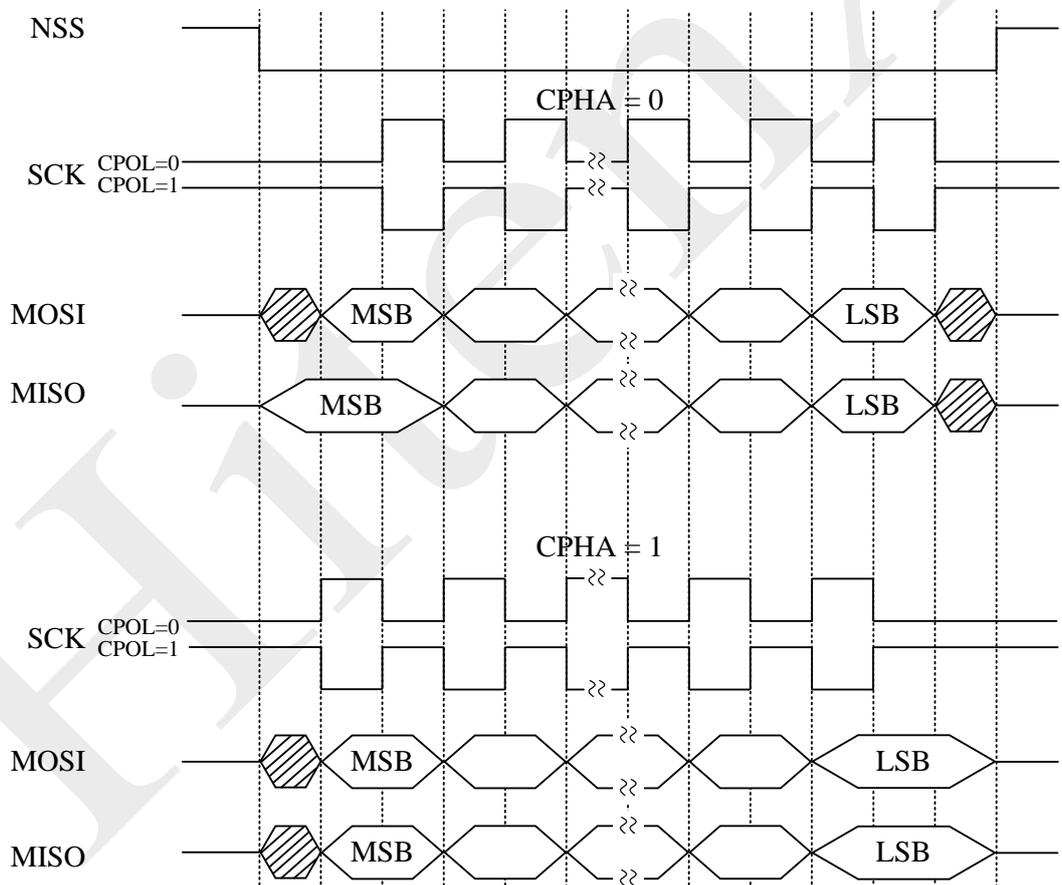
SPI 模式	CPOL	CPHA	说明
模式 0	0	0	时钟空闲为低电平，在时钟第一个边沿进行数据采样，即数据在时钟的上升沿采样，下降沿变化。
模式 1	0	1	时钟空闲为低电平，在时钟第二个边沿进行数据

SPI 模式	CPOL	CPHA	说明
			采样，即数据在时钟的下降沿采样，上升沿变化。
模式 2	1	0	时钟空闲为高电平，在时钟第一个边沿进行数据采样，即数据在时钟的下降沿采样，上升沿变化。
模式 3	1	1	时钟空闲为高电平，在时钟第二个边沿进行数据采样，即数据在时钟的上升沿采样，下降沿变化。

四种通信模式通过 SPIx_CR1 寄存器的时钟极性 CPOL 位和时钟相位 CPHA 位配置。

注意： SCK 的空闲状态必须与 SPIx_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

图 31-6 通信模式时序



SPI 内部移位寄存器支持可配置的数据传输顺序；通过 SPIx_CR1 寄存器的 LSBFIRST 位选择，LSBFIRST 为 0，MSB 优先；LSBFIRST 为 1，LSB 优先。

31.3.8 状态标志

提供发送 FIFO 状态、接收 FIFO 状态和总线忙状态，通过这些状态管理数据的发送和接收。

- 发送 FIFO 阈值标志

发送 FIFO 未使用空间大于或等于 SPIx_CR1 寄存器中 TXFC[2:0]设置阈值时, TXFT 标志置 1。未使用空间小于设置阈值时, TXFT 标志将由硬件自动清 0。

- 发送 FIFO 非满标志

当发送 FIFO 有未使用空间时, TXFNF 标志置 1。当发送 FIFO 满时, TXFNF 标志由硬件自动清 0。

- 发送 FIFO 空标志

当发送 FIFO 中没有数据时, TXFE 标志置 1。如果发送 FIFO 非空时, TXFE 标志由硬件自动清 0。

- 接收 FIFO 阈值标志

接收 FIFO 中已接收的数据的长度大于或等于 SPIx_CR1 寄存器中 RXFC[2:0]设置阈值时, RXFT 标志置 1。未读取数据长度小于设置阈值时, RXFT 标志由硬件自动清 0。

- 接收 FIFO 非空标志

当接收 FIFO 非空时, RXFNE 标志置 1。只有将接收 FIFO 中的数据全部读出, RXFNE 标志由硬件自动清 0。

- 接收 FIFO 满标志

当接收 FIFO 满时, RXFF 标志置 1。接收 FIFO 非满时, RXFF 标志由硬件自动清 0。此标志置 1 时, 尽快读出数据, 否则可能造成接收数据溢出。

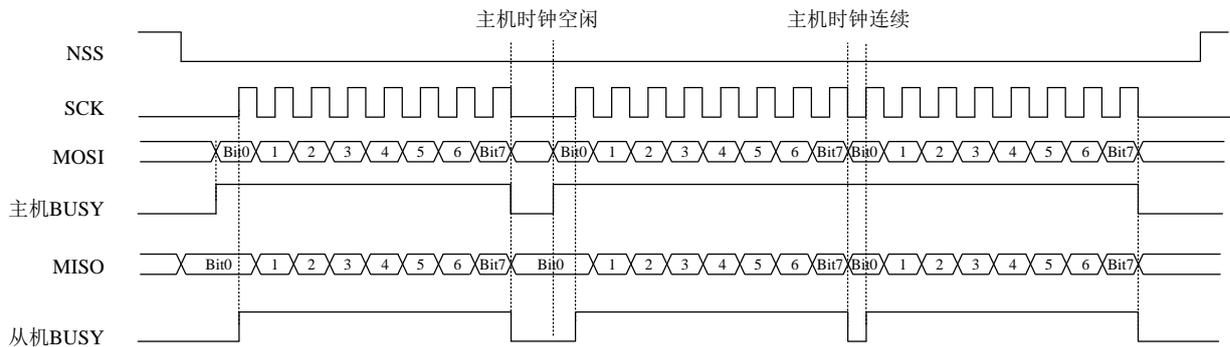
- 忙标志 (BSY)

BSY 标志由硬件置 1 和清 0。

当 BSY 置 1 时, 表示 SPI 总线上正在进行数据传输, BSY 标志检测一帧数据是否传输完成, 用以正确禁止 SPI, 避免破坏数据的完整性。

在以下任意一种条件下, BSY 标志将清 0:

- 在主机应用下, 发送 FIFO 已为空且最后一个数据全部发送完成时;
- 在从机应用下, 上一个数据帧最后的时钟边沿与下一个数据帧起始的时钟边沿之间 (相邻数据之间的至少半个 SPI 时钟周期);
- 在多主机应用下, 检测到主机模式冲突时。

图 31-7 主从模式 BSY 状态


注意： 当主机模式发送 FIFO 数据连续时，总线时钟连续，主机设备 BSY 标志始终置 1。但从机模式并非 BSY 一直为高。

31.3.9 错误或状态标志

以下错误或状态标志中，如有一个置 1 且对应中断使能 MMFIE、OVEIE、UDRIE 或 SMEIE 位置 1，则将生成 SPI 中断。

- 从机模式下溢标志

当 SPI 为从机模式，如果发送 FIFO 已空，而主机还在读取数据，此时 UDR 标志置 1。将 SPIx_ICR 寄存器中 UDRCF 位置 1，UDR 标志将同步清 0。

- 上溢标志

当移位寄存器已接收到数据但接收 FIFO 都没有空间保存此数据，此时 OVR 标志置 1。当出现上溢的情况时，新接收的数据不会覆盖接收 FIFO 中之前的值，新接收的数据将被丢弃。将 SPIx_ICR 寄存器中 OVRCF 位置 1，OVR 标志将同步清 0。

- 从机模式状态标志

当 SPI 为从机模式时，如果总线正在进行数据传输时，NSS 输入信号被拉高，此时 SME 标志置 1。未接收完整的数据被丢弃，接收 FIFO 中的未读的数据不受影响。将 SPIx_ICR 寄存器中 SMECF 位置 1，SME 标志将同步清 0。

- 主机模式冲突标志

当 SPI 主机模式且 NSS 配置为输入时，NSS 信号被拉低，将发生主机模式冲突，此时 MMF 标志置 1。多应用于多主机通信系统。

发生主机模式冲突时，硬件将自动执行如下动作：

- SPE 位清 0，禁止 SPI 接口；
- MSTR 位清 0，进入从机模式。

将 SPIx_ICR 寄存器中 MMFCF 位置 1，MMF 标志将同步清 0。

为避免包含多个 MCU 的系统中发生主机冲突，必须在 MMF 位清 0 期间将输出给其他从机的 NSS 引脚拉高。在 MMF 清 0 后，可以将 SPE 和 MSTR 位重新置 1。在从机模式中，MMF 位不会置 1，但由主机模式冲突引起的自动进入从机模式的情况除外。

31.3.10 初始化 SPI

除 BR[2:0]和 NSSE 外，主机模式和从机模式的配置步骤相同。对于具体的配置，参看相应功能介绍的内容。SPI 初始化步骤如下：

- 1) 将对应的 GPIO 配置为 NSS、SCK、MOSI 和 MISO 功能，参考：[通用输入/输出接口 \(GPIO\)](#)。主机模式下，不使用 SPI 控制 NSS 情况下，可将 NSS 配置为 GPIO 功能。
- 2) 将 SPIx_CR1 寄存器的 SPE 位清 0。
- 3) 写 SPIx_CR1 寄存器：
 - 通过 BR[2:0]位配置串行时钟波特率。从机模式无需配置此项。
 - 配置 CPOL 位和 CPHA 位，选择 SPI 模式。
 - 配置 LSBFIRST 位，选择数据位传输顺序。
 - 配置 NSSE 位，选择 NSS 状态。从机模式无需配置此项。
- 4) 同时将 SPIx_CR1 寄存器的 SPE 位置 1、配置 MSTR 位（选择主从模式）、配置 SSM（NSS 的软件设置使能）。

31.3.11 数据发送和接收

SPI 内置 8 字节发送 FIFO 和 8 字节接收 FIFO。对 SPIx_DR 寄存器执行写访问时，会将要发送数据移入发送 FIFO。对 SPIx_DR 寄存器执行读访问时，会返回尚未读取的接收 FIFO 中存储的最早的值。

SPIx_DR 的写访问可通过 TXFT、TXFNF 和 TXFE 状态管理。当 TXFT 置 1 时，可以连续对 SPIx_DR 执行多次写访问，写入次数取决于 TXFC[2:0]的配置。当 TXFNF 置 1 时，至少可对 SPIx_DR 执行 1 次写访问。当 TXFE 事件置 1 时，可以连续对 SPIx_DR 执行 8 次写访问。

SPIx_DR 的读访问可通过 RXFT、RXFNE 和 RXFF 状态管理。当 RXFT 置 1 时，可以连续对 SPIx_DR 执行多次读访问，读出次数取决于 RXFC[2:0]的配置。当 RXFNE 事件置 1 时，至少可对 SPIx_DR 执行 1 次读访问。当 RXFF 事件置 1 时，可以连续对 SPIx_DR 执行 8 次读访问。

TXFT、TXFNF、TXFE、RXFT、RXFNE 和 RXFF 事件可以通过查询或者中断的方式进行判断和处理。当使用中断方式时，将 SPx_CR1 的相关事件中断使能

置 1，发生事件时产生中断，参见：[SPI 中断](#)。

除查询和中断方式外，SPI 还支持 DMA 方式。

SPIx_CR1 寄存器中 DMA 方式发送使能 (DMAT 为 1) 时，每次 TXFNF 为 1，SPI 会发出 DMA 请求。

SPIx_CR1 寄存器中 DMA 方式接收使能 (DMA 为 1) 时，每次 RXFNE 为 1，SPI 会发出 DMA 请求。

当 SPI 仅用于发送数据时，可以只使能 SPI 发送的 DMA 通道。在这种情况下，SPI 的 OVR 标志会置 1，因为未读取接收的数据。当 SPI 仅用于接收数据时，可以只使能 SPI 接收的 DMA 通道。

在发送模式下，DMA 传输完成标志置 1 后，可以对 TXFE 和 BSY 标志进行查询，以确保 SPI 通信已完成。在关闭 SPI 必须执行此步骤，以避免损坏最后一次发送。

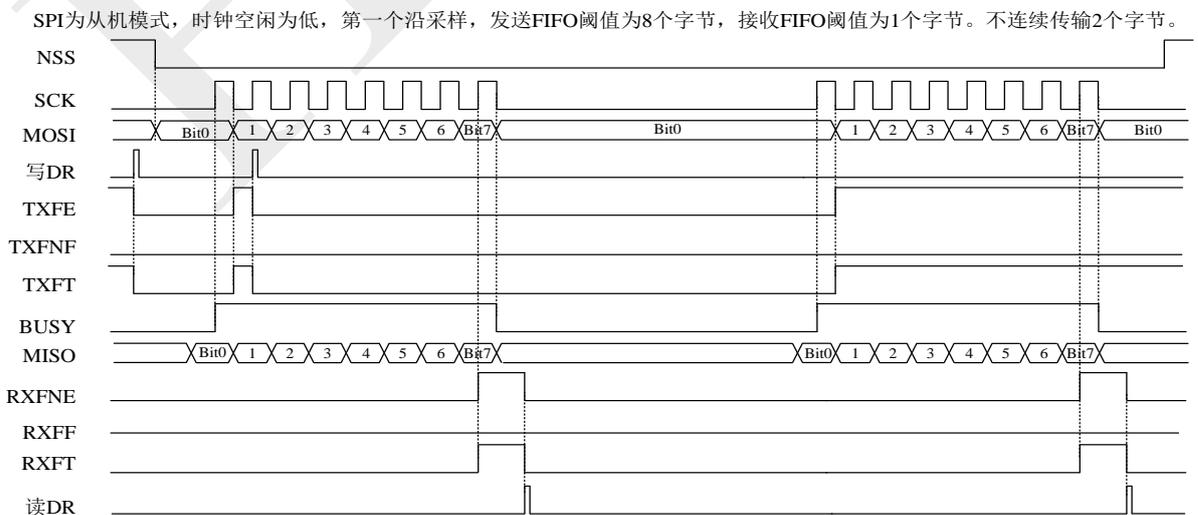
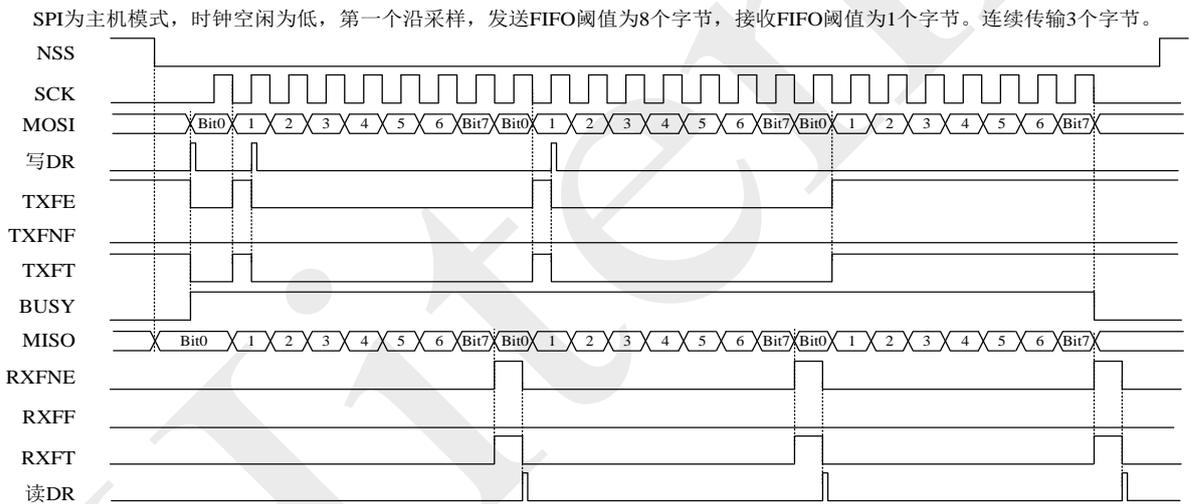
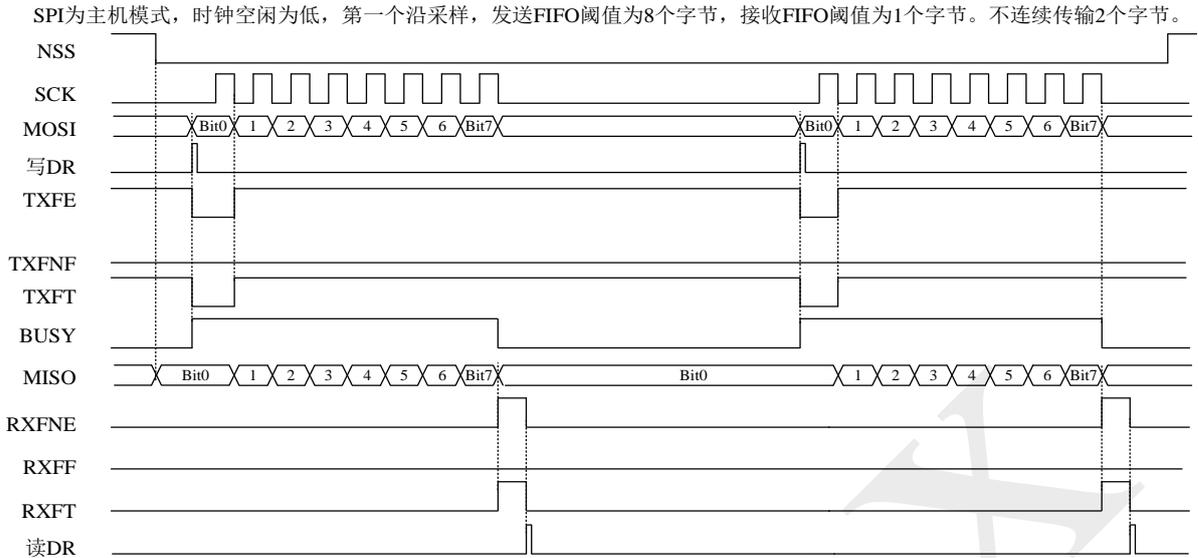
通过 DMA 发送和接收数据时，必须按顺序执行以下步骤：

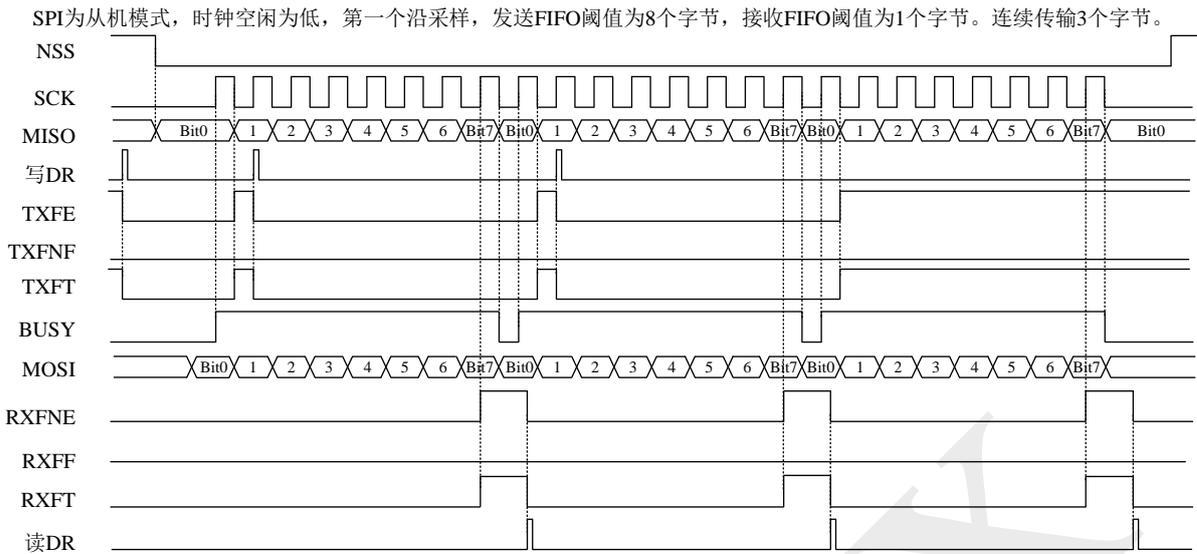
- 1) 配置 DMA 的数据发送通道和数据接收通道，参见：[直接存储器访问控制器 \(DMA\)](#)。
- 2) 使能 SPI 接收的 DMA 通道。
- 3) SPI_CR1 寄存器中 DMA 位置 1，启动 SPI DMA 方式接收。
- 4) 使能 SPI 发送的 DMA 通道。
- 5) SPI_CR1 寄存器中 DMAT 位置 1，启动 SPI DMA 方式发送。

要关闭 DMA 发送和接收数据，必须按顺序执行以下步骤：

- 1) 将 SPI_CR1 寄存器中 DMA 和 DMAT 位清 0。
- 2) 禁止 DMA 的数据发送通道和数据接收通道，参见：[直接存储器访问控制器 \(DMA\)](#)。

时钟信号由主机设备提供，直至发送 FIFO 和移位寄存器为空，之后时钟停止，直至主机再次发送数据。通信过程参见下图。

图 31-8 主从模式通信时序




31.3.12 禁止 SPI

当禁止 SPI 时，必须按照下述步骤进行操作，否则会损坏正在进行的交互。

正确的禁止步骤如下：

- 1) 等待数据发送完成直至 TXFE 为 1。
- 2) 等待 BSY 为 0，最后一帧数据已传输完成。
- 3) 读出接收数据直至 RXFNE 为 0 或清空接收 FIFO。
- 4) 将 SPIx_CR1 寄存器的 SPE 位清 0。禁止 SPI。

31.4 SPI 中断

在 SPI 通信过程中，SPI 中断可由不同事件生成。不同事件可配置使能或禁止生成中断。SPI 中断详情参见下表。

表 31-5 SPI 中断

中断事件	事件标志	使能控制位	清除方法
发送 FIFO 阈值	TXFT	TXFTIE	写 SPIx_DR 寄存器，使发送 FIFO 未使用空间小于设置阈值
发送 FIFO 非满	TXFNF	TXFNFIE	写 SPIx_DR 寄存器，使发送 FIFO 为满
发送 FIFO 空	TXFE	TXFEIE	写 SPIx_DR 寄存器，使发送 FIFO 不为空
接收 FIFO 阈值	RXFT	RXFTIE	读 SPIx_DR 寄存器，使接收 FIFO 未读数据长度小于设置阈值
接收 FIFO 非空	RXFNE	RXFNEIE	读 SPIx_DR 寄存器，使接收 FIFO 为空
接收 FIFO 满	RXFF	RXFFIE	读 SPIx_DR 寄存器，使接收 FIFO

中断事件	事件标志	使能控制位	清除方法
			不满
从机模式下溢	UDR	UDRIE	SPIx_ICR 寄存器 UDRCF 置 1
上溢	OVR	OVRIE	SPIx_ICR 寄存器 OVRDCF 置 1
从机模式错误	SME	SMEIE	SPIx_ICR 寄存器 SMECF 置 1
主机模式冲突	MMF	MMFIE	SPIx_ICR 寄存器 MMFCF 置 1

31.5 SPI 寄存器

31.5.1 SPI 控制寄存器 1 (SPIx_CR1)

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
Res.	TXFC[2:0]			TXFTIE	TXFNFIE	TXFEIE	RXFC[2:0]			RXFTIE	RXFNEIE	RXFFIE	Res.	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	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
MMFIE	OVRIE	UDRIE	SMEIE	SSM	NSSE	Res.	LSBFIRST	Res.	MSTR	CPOL	CPHA	BR[2:0]		SPE	
rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	TXFC[2:0]	发送 FIFO 阈值配置 000: 发送 FIFO 未使用空间达到其深度的 1/8 001: 发送 FIFO 未使用空间达到其深度的 1/4 010: 发送 FIFO 未使用空间达到其深度的 1/2 011: 发送 FIFO 未使用空间达到其深度的 3/4 100: 发送 FIFO 未使用空间达到其深度的 7/8 101: 发送 FIFO 全部空间未使用 其它: 保留
27	TXFTIE	发送 FIFO 阈值中断使能 0: 禁止发送 FIFO 阈值中断 1: 使能发送 FIFO 阈值中断
26	TXFNFIE	发送 FIFO 非满中断使能 0: 禁止发送 FIFO 非满中断 1: 使能发送 FIFO 非满中断
25	TXFEIE	发送 FIFO 为空中断使能 0: 禁止发送 FIFO 为空中断

		1: 使能发送 FIFO 为空中断
24:22	RXFC[2:0]	接收 FIFO 阈值配置 000: 接收 FIFO 已使用空间达到其深度的 1/8 001: 接收 FIFO 已使用空间达到其深度的 1/4 010: 接收 FIFO 已使用空间达到其深度的 1/2 011: 接收 FIFO 已使用空间达到其深度的 3/4 100: 接收 FIFO 已使用空间达到其深度的 7/8 101: 接收 FIFO 全部空间已使用 其它: 保留
21	RXFTIE	接收 FIFO 阈值中断使能 0: 禁止接收 FIFO 阈值中断 1: 使能接收 FIFO 阈值中断
20	RXFNEIE	接收 FIFO 非空中断使能 0: 禁止接收 FIFO 非空中断 1: 使能接收 FIFO 非空中断
19	RXFFIE	接收 FIFO 为满中断使能 0: 禁止接收 FIFO 为满中断 1: 使能接收 FIFO 为满中断
18	保留	必须保持复位值。
17	TXDMAEN	DMA 方式发送使能 0: 禁止 DMA 方式发送 1: 使能 DMA 方式发送
16	RXDMAEN	DMA 方式接收使能 0: 禁止 DMA 方式接收 1: 使能 DMA 方式接收
15	MMFIE	MMF 错误中断使能 0: 禁止 1: 使能
14	OVRIE	上溢错误(OVR)中断使能 0: 禁止 1: 使能
13	UDRIE	下溢错误(UDR)中断使能

		0: 禁止 1: 使能
12	SMEIE	从模式 NSS 状态中断使能 0: 禁止 1: 使能
11	SSM	NSS 的软件设置使能（只在 slave 模式有效） 1: 由软件直接将 NSS 输入在模块内部强制设置为 0 0: 由外部 NSS 管脚控制 NSS 的电平
10	NSSE	NSS 输出使能 0: NSS 为输入模式 1: NSS 为输出模式 <i>注意：此位仅在主机模式下设置有效。</i>
9	保留	必须保持复位值。
8	LSBFIRST	数据传输顺序 0: 发送/接收数据时 MSB 在前 1: 发送/接收数据时 LSB 在前 <i>注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。</i>
7	保留	必须保持复位值。
6	MSTR	SPI 主从模式 0: 从机模式 1: 主机模式 <i>注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。</i>
5	CPOL	时钟极性 0: SCK 空闲状态为低电平 1: SCK 空闲状态为高电平 <i>注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。</i>
4	CPHA	时钟相位配置 0: 第一边沿采样数据 1: 第二边沿采样数据 <i>注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。</i>
3:1	BR[2:0]	波特率控制 000: $f_{\text{PCLK}}/2$ 001: $f_{\text{PCLK}}/4$

- 010: $f_{PCLK}/8$
- 011: $f_{PCLK}/16$
- 100: $f_{PCLK}/32$
- 101: $f_{PCLK}/64$
- 110: $f_{PCLK}/128$
- 111: 保留

注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。

- 0 SPE SPI 使能控制
- 0: 禁止
- 1: 使能

31.5.2 SPI 控制寄存器 2 (SPIx_CR2)

偏移地址：0x04

复位值：0x0000 0001

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

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	NSSP	NSS 输出电平 该位结合 NSSE 使用，NSSE 为 1 时，NSS 输出有效。 0: NSS 输出低电平 1: NSS 输出高电平 注意：此位仅在主机模式下设置有效。

31.5.3 SPI FIFO 清空寄存器 (SPIx_FIFOCLR)

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

7	RXFT	接收 FIFO 阈值标志 接收 FIFO 已接收数据长度大于或等于阈值时置 1，否则由硬件清 0。 0: 接收 FIFO 未达到设置阈值 1: 接收 FIFO 已达到设置阈值
6	RXFNE	接收 FIFO 非空标志 接收 FIFO 非空时置 1，否则硬件清 0。 0: 接收 FIFO 为空，未接收到数据 1: 接收 FIFO 非空，已接收到数据
5	RXFF	接收 FIFO 已满标志 接收 FIFO 为满时置 1，否则硬件清 0。 0: 接收 FIFO 非满 1: 接收 FIFO 已满
4	BSY	SPI 总线传输状态标志 由硬件置 1 和清 0。 0: SPI 总线空闲 1: SPI 总线正在传输数据
3	UDR	从机模式下溢标志 在从机模式下，发送 FIFO 已空而主机仍在读取数据时，此位置 1。将 SPIx_ICR 寄存器中的 UDRCF 位置 1，此位清 0。 0: 未发生下溢 1: 发生下溢
2	OVR	上溢标志 接收 FIFO 已满而移位寄存器又接收到数据时，此位置 1。将 SPIx_ICR 寄存器中的 OVRCF 位置 1，此位清 0。 0: 未发生上溢 1: 发生上溢
1	SME	从机模式 NSS 状态标志 从机模式正在通信中，NSS 被拉高时，此位置 1。将 SPIx_ICR 寄存器中的 SMECF 位置 1，此位清 0。 0: 未发生从机模式 NSS 电平从低拉到高状态 1: 发生从机模式 NSS 电平从低拉到高状态
0	MMF	主机模式冲突标志

在主机模式下，NSS 输入电平为低，此位置 1。将 SPIx_ICR 寄存器中的 MMFCF 位置 1，此位清 0。

0: 未发生主机模式冲突

1: 发生主机模式冲突

注意：发生主机模式冲突时，MSTE 和 SPE 同时由硬件清 0。此标志不消除，无法置位 MSTE 和 SPE。

31.5.5 SPI 中断标志清零寄存器 (SPIx_ICR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												UDRCF	OVRCF	SMECF	MMFCF
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	UDRCF	从机模式下溢标志清除 写 1 清除 SPIx_ISR 寄存器中 UDR 标志。
2	OVRCF	上溢标志清除 写 1 清除 SPIx_ISR 寄存器中 OVR 标志。
1	SMECF	从机模式 NSS 状态标志清除 写 1 清除 SPIx_ISR 寄存器中 SME 标志。
0	MMFCF	主机模式冲突标志清除 写 1 清除 SPIx_ISR 寄存器中 MMF 标志。

31.5.6 SPI 数据寄存器 (SPIx_DR)

偏移地址：0x14

复位值：0x0000 0000

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

Res.								DR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	DR[7:0]	数据寄存器 对数据寄存器执行写操作时，将要发送数据写入发送 FIFO； 对数据寄存器执行读操作时，将返回接收 FIFO 中的已接收的数据。

32 实时时钟计数器 (RTC)

32.1 简介

RTC 实时时钟，具有日历计时功能，并能在任意低功耗模式下进行自动唤醒。同时 RTC 也是独立的二进制码十进制数 (Binary-Coded Decimal, BCD) 定时器/计数器，提供了中断可配置的日历闹钟。无论芯片运行在什么状态 (运行状态、低功耗状态、复位状态)，只要供电电压保持在可工作范围内，RTC 就不会停止工作。

32.2 RTC 主要特性

RTC 主要特性如下：

- 日历采用 BCD 格式，包含秒、分钟、小时 (12/24 小时制)、星期、日期、月份和年份等信息；
- 月底自动修正月份的日期，28、29 (闰年)、30 和 31 日；
- 包含 1 个可配置闹钟；
- 具有 RTC 时钟同步功能，能对 RTC 时钟进行 1 到 32767 个时钟脉冲的实时修正；
- 含有分辨率可达 0.96ppm 的数字校准电路，能够有效校准石英晶振的误差。
- 包含 1 个 16 位自动重载唤醒定时器 (WUT)，具有可配置的分辨率和定时周期；

RTC 外设的时钟源可以是：

- 32.768kHz 外部晶体振荡器 (LSE)；
- 内部低功耗 RC 振荡器 (LSI，典型频率为 32kHz)；
- 经过 RCC 预分频后的外部高速时钟 (HSE)；

当 RTC 时钟由 LSE 或 LSI 提供时，RTC 可在所有低功耗模式下工作。

所有 RTC 事件 (闹钟、唤醒定时器) 都可以产生中断并将设备从低功耗模式唤醒

32.3 RTC 功能说明

实时时钟的时钟源可配置为外部低速晶振(LSE)、外部高速晶振(HSE)或内部慢速 RC (LSI)，上电默认 RTC 的时钟源为无时钟。控制器 RTC_CR0、RTC_CR1 与 RTC_COMPENR 只受上电复位控制，其它复位源不能复位这三个控制寄存器。其它数据寄存器上电状态不定，上电后需要初始化，不受任何复位影响。

所有软件写入和读取的日期时间值都为 BCD 码，无须十六进制转换为十进制。任何无效的日期时间都无法写入，比如 32 日、25 时、61 秒、13 月等。

32.3.1 RTC 框图

图 32-1 RTC 结构框图

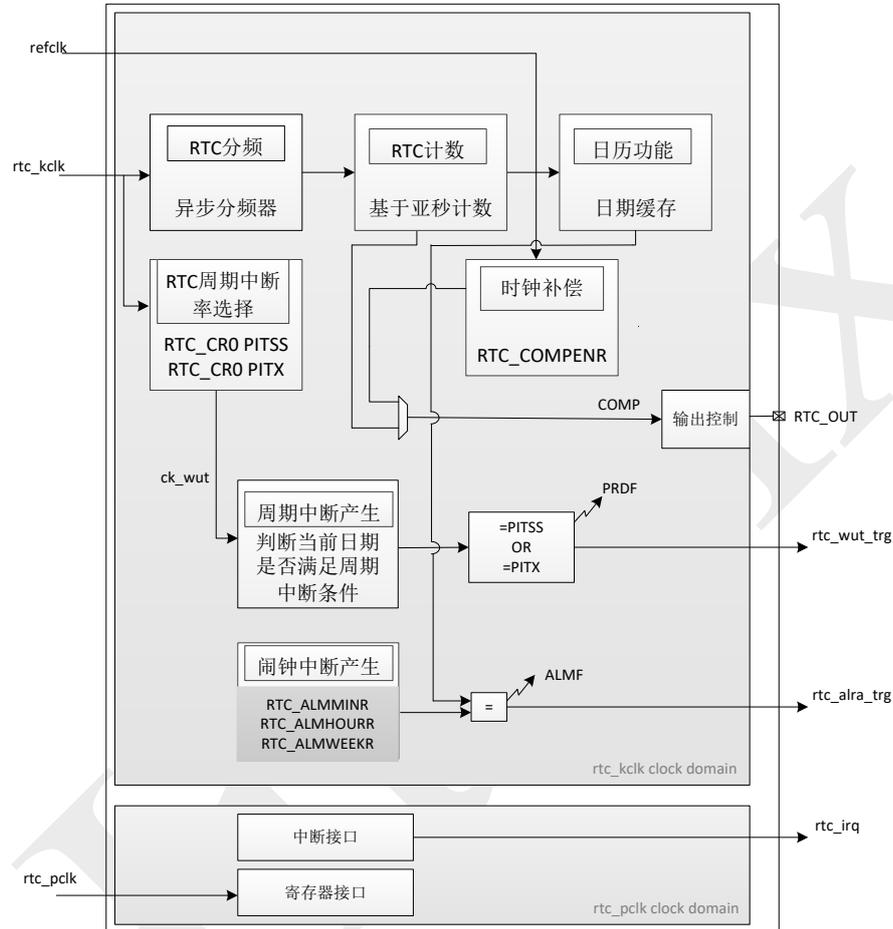


表 32-1 RTC 输入/输出引脚

引脚名称	信号类型	说明
RTC_OUT	输出	RTC 1Hz 时钟输出

表 32-2 RTC 内部输入/输出信号

引脚名称	信号类型	说明
rtc_kclk	输入	RTC 内核时钟，在本文档中也称为 RTCCLK
rtc_pclk	输入	RTC APB 时钟
refclk	输入	校准时钟源
rtc_irq	输出	RTC 中断
rtc_alra_trg	输出	RTC 闹钟事件检测触发
rtc_wut_trg	输出	RTC 唤醒事件检测触发

32.3.2 RTC 初始化和配置

RTC 寄存器为 32 位寄存器。RTC 在上电之后复位一次，在系统不掉电的情况下，各种复位请求都不能复位 RTC，RTC 会一直处于计数状态。在上电之后，需要软件按以下步骤设定 RTC 的时钟源选择、日历初始值、闹钟设置、RTC 输出时钟误差校准、中断使能等之后，启动 RTC。

- 1) 设定 `RTC_CR0.START=0`, 计数停止;
- 2) 设定 `RTC_CR0.AMPM` 和 `RTC_CR0.PITS`, `RTC_CR0.PITX` 设定定时制和中断周期;
- 3) 设定 `RCC_BDCR.RTCSEL`, 选择 RTC 的计时时钟;
- 4) 设定秒、分、时、周、日、月、年的日历计数寄存器;
- 5) 需要进行时钟校准时, 设定计数时钟校准寄存器 `RTC_COMPENR`;
- 6) 清除中断标志位 `RTC_CR1.ALMF`, `RTC_CR1.PITF`, 并使能中断;
- 7) 设定 `RTC_CR0.START=1`, 计数开始。

32.3.3 RTC 低功耗模式切换

在 RTC 计数开始后，如果系统需要立即切换为低功耗模式时，请执行下列任何一种方式后再进行模式切换：

- 设置 RCC 控制寄存器(`RCC_BDCR`)的 `RTCLPW` 位;
- 在 `RTC_CR0.START=1` 设定后，经过 2 个以上 RTC 计数时钟后再进行切换。
- `RTC_CR0.START=1` 设定后，设定 `RTC_CR1.WAIT=1`，查询

RTC_CR1.WAITF=1.再设定 RTC_CR1.WAIT=0, 查询 RTC_CR1.WAITF=0 后再进行切换。

注意：在 RTC 低功耗模式下，RTC 寄存器不能读写。

32.3.4 RTC 读取日历

有三种读取日历寄存器方式：

方式一、任意时刻读取方式 1

- 1) 设定 RTC_CR1.WAIT=1,停止日历寄存器计数，进入读写模式；
- 2) 查询直到 RTC_CR1.WAITF=1；
- 3) 读出秒、分、时、周、日、月、年计数寄存器值；
- 4) 设定 RTC_CR1.WAIT=0，计数器重新开始计数；
- 5) 查询直到 RTC_CR1.WAITF=0；

方式二、任意时刻读取方式 2

- 1) 读出分、时、周、日、月、年计数寄存器值
- 2) 读出秒计数寄存器值
- 3) 再次读出秒计数寄存器值
- 4) 判断两次秒的读出值是否相等，不同的话重新从第一步开始，直到读取结束；

方式三、中断读取方式

在 RTC 中断服务中读出秒、分、时、周、日、月、年计数寄存器值。因为中断发生后到下次数据改变至少 0.5 秒的时间。

32.3.5 RTC 写入日历寄存器

- 1) 设定 RTC_CR1.WAIT=1,停止日历寄存器计数，进入读写模式；
- 2) 查询直到 RTC_CR1.WAITF=1；
- 3) 写入秒、分、时、周、日、月、年计数寄存器值；
- 4) 设定 RTC_CR1.WAIT=0，计数器重新开始计数。注意：必须在 1 秒内完成所有操作；
- 5) 查询直到 RTC_CR1.WAITF=0；

注意：在 RTC 未启动情况下写秒、分、时、周、日、月、年计数寄存器不需要等 WAIT。

在计数模式下更改秒寄存器会复位秒计数，写、时、周、日、月、年计数寄存器不会影响 RTC 计数。

32.3.6 RTC 闹钟设定

- 1) 设定 RTC_CR1.ALMEN=0, 闹钟禁止;
- 2) 设定 RTC_CR1.ALMIE=1, 闹钟中断许可;
- 3) 分闹钟 RTC_ALMMINR、时闹钟 RTC_ALMHOURR、周闹钟 RTC_ALMWEEKR 设定;
- 4) 设定 RTC_CR1.ALMEN=1, 闹钟使能;
- 5) 等待发生中断
- 6) 由于闹钟中断和周期中断共用中断请求, 当 RTC_CR1.ALMF=1 时, 进入闹钟中断处理, 否则进入周期中断处理。

32.3.7 RTC 输出时钟校准

由于外部晶振在各种温度条件下工作存在输出频率偏差, RTC 在需要输出高精度时, 需要对产生的 1Hz 时钟进行校准。校准方法参照 32.4.13 RTC 时钟校准寄存器。

32.3.8 RTC 时钟输出

当选择时钟源频率为 32.768k 情况下, 秒计数器 RTC_SECR 计数变化 1 时实际间隔 1s。当选择其他时钟频率时, 秒计数器 RTC_SECR 计数变化 1 实际间隔不一定为 1s。

RTC 可输出 3 种精度的 1Hz 时钟, 分别为未校准的普通精度时钟、每 32 秒内平均校准的分布式时钟及每秒校准的均匀式时钟。其中:

普通精度的 1Hz 输出设定如下:

- 1) 设定 RTC_CR0.START=0, 计数停止;
- 2) RTC 输出引脚设定;
- 3) RTC_CR0.OUTEN=1, 时钟输出使能;
- 4) 设定 RTC_CR0.START=1, 计数开始;
- 5) 等待 2 个计数周期以上;
- 6) 1Hz 输出开始。

分布式 1Hz 输出设定如下:

- 1) 设定 RTC_CR0.START=0, 计数停止;
- 2) RTC 输出引脚设定;
- 3) RTC_CR0.OUTEN=1, 时钟输出使能;
- 4) 时钟校准寄存器 RTC_COMPENR.CALR 值设定;
- 5) 时钟校准寄存器 RTC_COMPENR.CALEN=1, 误差校准有效;
- 6) 设定 RTC_CR0.START=1, 计数开始;
- 7) 等待 2 个计数周期以上;

8) 1Hz 输出开始。

均匀式 1Hz 输出设定如下：

- 1) 设定 RTC_CR0.START=0, 计数停止；
- 2) RTC 输出引脚设定；
- 3) RTC_CR0.OUTEN=1, 时钟输出使能；
- 4) RTC_CR0.CALCM=1, 选择输出均匀式 1Hz 时钟；
- 5) 配置高速校准时钟 RTC_FREQ_ADJUST
- 6) 时钟校准寄存器 RTC_COMPENR.CALR 值设定；
- 7) 时钟校准寄存器 RTC_COMPENR.CALEN=1, 误差校准使能；
- 8) 设定 RTC_CR0.START=1, 计数开始；
- 9) 等待 2 个计数周期以上；
- 10) 1Hz 输出开始。

32.3.9 RTC 中断

RTC 支持两种中断类型，即闹钟中断和周期中断。闹钟中断与周期中断共用一个中断信号；

1) RTC 闹钟中断

当 RTC_CR1.ALMIE=1 时，若当前日历时间与分闹钟寄存器(RTC_ALMMINR)、时闹钟寄存器(RTC_ALMHOURLR)、周闹钟寄存器(RTC_ALMWEEKR)相等时，触发闹钟中断。

2) RTC 周期中断

当 RTC_CR1.ALMIE=1 时，选择的周期间隔到达后，触发周期唤醒中断，由于闹钟和周期共用中断，通过控制寄存器 1(RTC_CR1)的 ALMF 和 PITF 位来区分。

32.4 RTC 寄存器

32.4.1 RTC 控制寄存器 0 (RTC_CR0)

偏移地址：0x00

复位值：0x0000 0000（只有上电对该寄存器复位有效）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	PITSS	PITX					STAR T	HZ1S	HZ1O E	Res.	AMPM	PITS			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	PITSS	RTC周期中断源选择 0: 使用PITS所设定的周期中断时间间隔 1: 使用PITX所设定的周期中断时间间隔
13:8	PITX	设置产生周期中断的时间间隔，可设定的范围为0.5秒到32秒，步进为0.5秒 000000: 0.5秒 000001: 1秒 111110: 31.5秒 111111: 32秒
7	START	RTC计数器启动 0: 停止RTC计数器 1: 启动RTC计数器
6	HZ1S	0: RTC 1Hz分布式校准 1: RTC 1Hz均匀式校准 注：配合RTC_COMPENR的EN=1使用
5	HZ1OE	0: 禁止RTC输出 1: 使能RTC输出
4	保留	必须保持复位值。
3	AMPM	小时格式 0: AM/PM小时格式(12小时) 1: 24小时/天格式
2:0	PITS	RTC中断周期选择 000:不产生周期中断 001:每0.5秒 010:每1秒 011:每1分钟 100:每1小时 101:每1天(每日00时00分00秒) 11x:每1月(每月1日00时00分00秒) 注意：在START=1计数过程中写入周期选择时，为防止误动作请在NVIC中关闭RTC中断，并在写入选择位后将RTC中断标志清除。

32.4.2 RTC 控制寄存器 1 (RTC_CR1)

偏移地址：0x04

复位值：0x0000 0000（只有上电对该寄存器复位有效）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								ALMEN	ALMIE	Res	ALMF	PITF	Res	WAITF	WAIT
								rw	rw		rw	rw		r	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	ALMEN	0:禁止闹钟 1:使能闹钟 在START=1日历计数过程中并且ALMIE=1中断使能的情况下使能ALMEN时，为防止误动作请将系统中断关闭。使能后请将ALMF标志位清除。
6	ALMIE	0:禁止闹钟中断 1:使能闹钟中断
5	保留	必须保持复位值。
4	ALMF	0: 未发生闹钟中断 1: 已发生闹钟中断 注意：该位仅在ALMEN=1时有效。闹钟匹配时，32.768KHz一个时钟后置1.写0时清除标志，写1无效。
3	PITF	0: 未发生周期中断 1: 已发生周期中断
2	保留	必须保持复位值。
1	WAITF	0: 非写入/读出状态 1: 写入/读出状态 注意：WAIT位设定是否有效标志。在写入/读出前请确认该位是否为1。计数过程中，在WAIT位清0后等待写入完成后该位才能清0。
0	WAIT	0: 正常计数模式 1: 写入/读出模式 注意：在写入/读出日历寄存器时将该位置为1。由于计数器在连续计数，请在1秒的时间内完成写入/读出操作并将该位清零。

32.4.3 RTC 秒计数寄存器 (RTC_SECR)

偏移地址：0x08

复位值：0x0000 00XX

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

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	SECH	秒计数十位值
3:0	SECL	秒计数个位值。

注：这个寄存器表示0~59秒，采用十进制计数。请写入十进制0~59的BCD码，写入错误值时，写入值将被忽略。

32.4.4 RTC 分计数寄存器 (RTC_MINR)

偏移地址：0x0C

复位值：0x0000 00XX

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

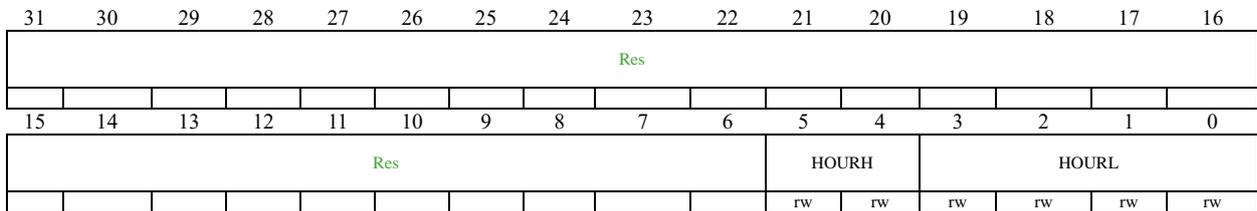
位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MINH	分计数十位值
3:0	MINL	分计数个位值。

注：这个寄存器表示0~59分，采用十进制计数。请写入十进制0~59的BCD码，写入错误值时，写入值将被忽略。

32.4.5 RTC 时计数寄存器 (RTC_HOUR)

偏移地址：0x10

复位值：0x0000 00XX



位/位域	名称	描述
------	----	----

31:6	保留	必须保持复位值。
------	----	----------

5:4	HOURL	时计数十位值
-----	-------	--------

3:0	HOURL	时计数个位值。
-----	-------	---------

注：这个寄存器24小时制时表示0~23小时。12小时制时，b5=0表示AM，则01H~21H表示上午，b5=1表示PM，则21H~32H表示下午。

请根据控制寄存器AMPM位的值，设定正确十进制0~23或者01~12、21:32的BCD码。写入超过范围的值将被忽略。

24小时制	AMPM=1	12小时制	AMPM=0
时间	寄存器值 (H)	时间	寄存器值 (H)
00时	00	AM 12时	12
01时	01	AM 01时	01
02时	02	AM 02时	02
03时	03	AM 03时	03
04时	04	AM 04时	04
05时	05	AM 05时	05
06时	06	AM 06时	06
07时	07	AM 07时	07
08时	08	AM 08时	08
09时	09	AM 09时	09
10时	10	AM 10时	10
11时	11	AM 11时	11
12时	12	PM 12时	32
13时	13	PM 01时	21
14时	14	PM 02时	22
15时	15	PM 03时	23
16时	16	PM 04时	24
17时	17	PM 05时	25
18时	18	PM 06时	26
19时	19	PM 07时	27

20时	20	PM 08时	28
21时	21	PM 09时	29
22时	22	PM 10时	30
23时	23	PM 11时	31

32.4.6 RTC 日计数寄存器 (RTC_DAYR)

偏移地址: 0x14

复位值: 0x0000 00XX

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

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	DAYH	日计数十位值
3:0	DAYL	日计数个位值。

注: 这个寄存器表示第1~31日, 采用十进制计数。自动计算闰年和月份, 具体表示如下:

月份	日计数表示
2月(普通年)	01~28
2月(闰年)	01~29
4、6、9、11月	01~30
1、3、5、7、8、10、12月	01~31

32.4.7 RTC 周计数寄存器 (RTC_WEEKR)

偏移地址: 0x18

复位值: 0x0000 000X

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

位/位域	名称	描述
31:3	保留	必须保持复位值。

2:0 WEEK 周计数值。

注：这个寄存器0~6表示周日~周六，请写入正确的十进制0~6的BCD码，写入其它值，将被忽略。周计数值对应关系如下：

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

32.4.8 RTC 月计数寄存器 (RTC_MONR)

偏移地址：0x1C

复位值：0x0000 00XX

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

位/位域	名称	描述
------	----	----

31:6	保留	必须保持复位值。
------	----	----------

4:0	MON	月计数值。
-----	-----	-------

注：这个寄存器十进制1~12表示1~12月，请写入正确的十进制1~12的BCD码，写入其他值，将被忽略。

32.4.9 RTC 年计数寄存器 (RTC_YEARR)

偏移地址：0x20

复位值：0x0000 00XX

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

位/位域	名称	描述
------	----	----

31:8	保留	必须保持复位值。
------	----	----------

7:4 YEARH 年计数十位值

3:0 YEARL 年计数个位值。

注：这个寄存器十进制0~99表示第0~99年，根据月进位计数。自动计算闰年如：00、04、08、...、92、96等。请写入正确的十进制年计数值，写入其他值，将被忽略。

32.4.10 RTC 分闹钟寄存器 (RTC_ALMMINR)

偏移地址：0x24

复位值：0x0000 00XX

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

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MINH	分闹钟匹配十位值
3:0	MINL	分闹钟匹配个位值。

注：这个寄存器表示0~59分，采用十进制计数。请写入十进制0~59的BCD码，写入错误值时，不会发生闹钟匹配。

32.4.11 RTC 时闹钟寄存器 (RTC_ALMHOURR)

偏移地址：0x28

复位值：0x0000 00XX

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

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	HOURH	时闹钟匹配十位值

3:0 HOURL 时闹钟匹配个位值。
 请根据时制设定正确的闹钟匹配值，否则不会发生闹钟匹配。

32.4.12 RTC 周闹钟寄存器 (RTC_ALMWEKR)

偏移地址：0x2C

复位值：0x0000 00XX

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

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:0	WEEK	周闹钟匹配值。 Bit0~bit6分别对应周日~周六，对应位置1时，则代表每周该日闹钟有效。如bit0=1,bit5=1表示周日和周五闹钟设定有效。

32.4.13 RTC 时钟校准寄存器 (RTC_COMPENR)

偏移地址：0x30

复位值：0x0000 0020（只有上电复位对该寄存器复位有效）

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	EN	时钟误差校准使能 0: 禁止时钟误差校准 1: 使能时钟误差校准
14:9	保留	必须保持复位值。
8:0	CR	时钟校准值。 通过校准值设定，可针对每秒进行+/-0.96ppm的精度校

准。校准值为9位带小数点的2的补码，后5位为小数部分，可校准范围为-274.6ppm~212.6ppm。

具体校准精度参考下表：

校准值设定									校准数
EN	CR[8:0]								
1	1	0	0	0	0	0	0	0	-274.6ppm
	1	0	0	0	0	0	0	1	-273.7ppm

	0	0	0	0	1	1	1	1	-0.95ppm
	0	0	0	1	0	0	0	0	0ppm

	0	1	1	1	1	1	1	1	+211.7ppm
	0	1	1	1	1	1	1	1	+212.6ppm
0	x	x	x	x	x	x	x	x	无校准

33 可编程逻辑阵列 (PLA)

33.1 简介

可编程逻辑阵列 (PLA) 模块提供了多个用户可编程的数字逻辑块, 在没有 CPU 干预的情况下操作。它由四个专用的独立的可编程逻辑单元 (PLU) 组成。它们支持用户可编程的异步和同步的布尔逻辑操作。内部和外部信号都可以用作 PLU 的输入, 输出可以连接到端口 I/O 引脚或直接选择外围输入作为输出。

33.2 功能特性

可编程逻辑阵列 (PLA) 的主要特点如下:

- 四个可编程逻辑单元 (PLU), 具有直接引脚和内部逻辑连接
- 每个单元支持 256 种不同的组合逻辑功能 (AND、OR、XOR、混合等), 以及包括一个时钟触发器以便进行同步操作。
- 单元可以进行同步操作, 也可以进行异步操作
- 单元模块可以级联在一起, 以便实现更复杂的逻辑功能
- 可以与串行外设进行连接, 比如 USART、SPI 或者定时外设 (高级定时器、通用定时器和基本定时器)
- 可以用于同步和触发多个片上资源 (ADC、DAC、定时器等)
- 异步输出可以用于低功耗状态唤醒

33.3 功能说明

33.3.1 结构框图

图 33-1 PLA 顶层框图

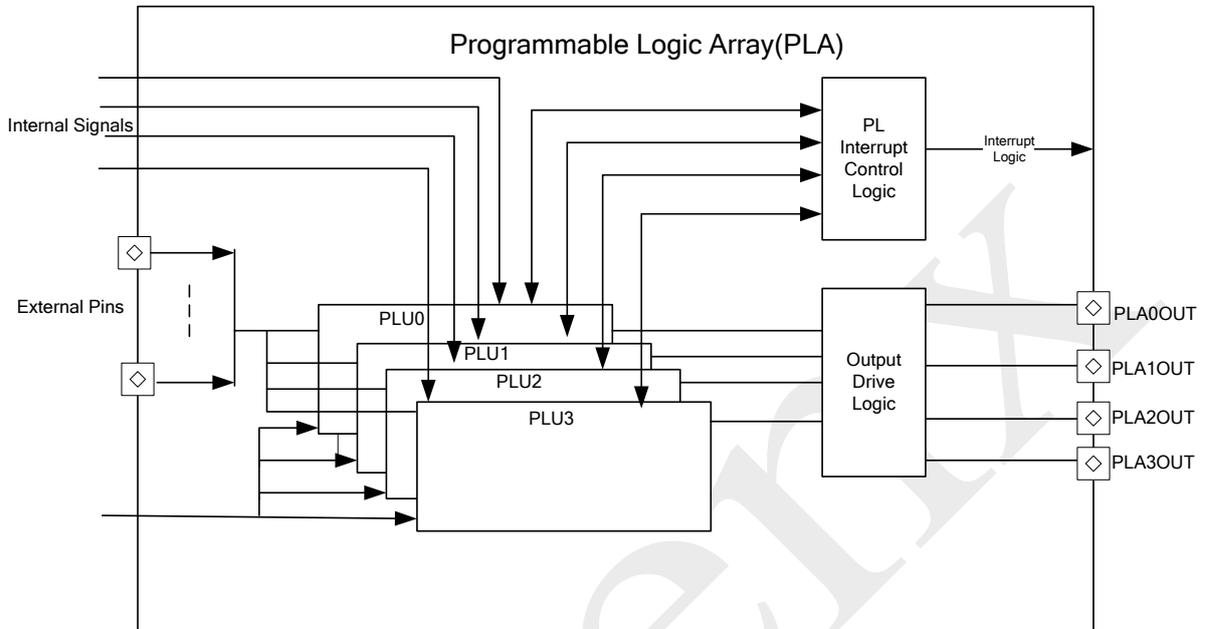
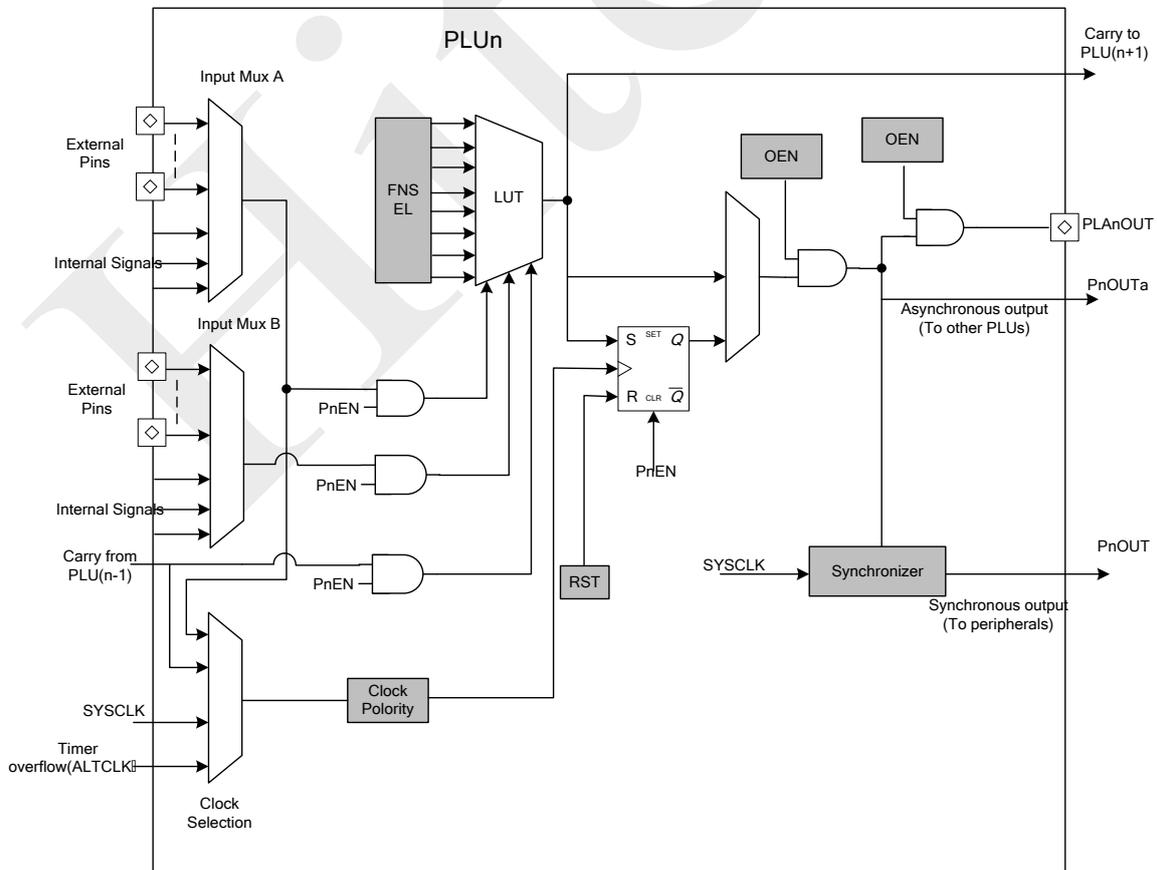


图 33-2 PLA 内部框图



33.3.2 PLA 配置顺序

固件应该在启用单个 PLU 之前配置功能选择、mux 输入和输出功能。PLU 的初始化包括以下一般步骤：

- 1、在 PLA_CFGRx.MXy 中选择 A 和 B 的输入
- 2、利用 PLA_CFGRx.FNSEL 选择 LUT 的功能
- 3、通过 PLA_CFGRx.OUTSEL 配置 PLU
- 4、如果为 PLU 选择了 D 触发器输出 (OUTSEL=1)，则建议还设置 RST=1 以将触发器输出重置为 0。
- 5、设置 PLA_CR 中所需要的任何中断。每个模块的下降和上升边缘中断分别使用 PxFIE 和 PxFIE 位启用。
- 6、通过在 PLA_CR 中设置 PxEN 位来启用 PLU。固件可以设置多个位来同时启用多个 PLU。
- 7、如果需要直接引脚输出，固件可以通过在 PLA_CFGRx 中设置 OEN 位来启用输出

33.3.3 输入多路复用器选择

每个 PLU 都有两个主要逻辑输入 (A 和 B) 和一个承载输入(C)。A 和 B 输入是由 PLA_CFGRx 寄存器中的 MXA 和 MXB 字段选择，并且可能是许多不同的内部和外部信号之一。当选择另一个 PLU 输出作为输入时，将使用来自该 PLU 的异步输出，从而可以实现更复杂的布尔逻辑函数。

注意：当使用计时器溢出事件作为输入时，计时器溢出事件是一个脉冲，在一个 SYSCLK 周期内逻辑高，在剩余的计时器周期内逻辑低。

承载输入 C 是前一个 PLU 的 LUT 输出。例如，PLU1 上的携带输入是 PLU0 的 LUT 输出。PLU0 比较特殊，它的携带输入是比较器 3 的输出。

对 PLU 输入的引脚输入没有 SYSCLK 同步。到 PLU 输入的其他内部外设（如定时器）是 SYSCLK 同步的，因为这些外设是 SYSCLK 同步的。一个脉冲需要至少有 1 个 SYSCLK 宽，以保证在捕获模式下的计时器，以保证捕获边缘。然而，仍然有可能捕获到一个更窄的脉冲。因此，包含 PLU 的固件不能依赖于计时器不捕获小于 1 个 SYSCLK 周期宽的脉冲。

33.3.3.1 PLU 多路复用器输入选择

表 33-1 PLA_CFGRx.MXA

PLA_CFGRx.MXA	PLU0A	PLU1A	PLU2A	PLU3A
0000	Uart1_tx	P0OUTa	P1OUTa	P2OUTa
0001	eventout	eventout	eventout	eventout

0010	Rtc_wut_trig	Rtc_wut_trig	Rtc_wut_trig	Rtc_wut_trig
0011	Spi1_miso	Spi1_miso	Spi1_miso	Spi1_miso
0100	Tim3_ch1	Tim3_ch2	Tim3_ch3	Tim3_ch4
0101	Tim2_ch1	Tim2_ch2	Tim2_ch3	Tim2_ch4
0110	Tim1_ch1	Tim1_ch2	Tim1_ch3	Tim1_ch4
0111	Tim4_ch1	Tim4_ch2	Tim4_ch3	Tim4_ch4
1000	PLAIN1/PC0	PLAIN3/PA6	PLAIN1/PC0	PLAIN1/PC0
1001	PLAIN2/PC1	PLAIN4/PA7	PLAIN2/PC1	PLAIN2/PC1
1010	PLAIN3/PA6	PLAIN5/PA8	PLAIN3/PA6	PLAIN3/PA6
1011	PLAIN4/PA7	PLAIN6/PA9	PLAIN4/PA7	PLAIN4/PA7
1100	PLAIN6/PA9	PLAIN7/PC4	PLAIN8/PC5	PLAIN5/PA8
1101	PLAIN7/PC4	PLAIN8/PC5	PLAIN9/PC6	PLAIN6/PA9
1110	PLAIN8/PC5	PLAIN9/PC6	PLAIN10/PC7	PLAIN7/PC4
1111	PLAIN9/PC6	PLAIN10/PC7	PLAIN11/PB15	PLAIN8/PC5

表 33-2 PLA_CFGRx.MXB

PLA_CFGRx.MXB	PLU0B	PLU1B	PLU2B	PLU3B
0000	Lpuart_tx	Uart2_tx	Uart3_tx	Lpuart_tx
0001	Adc_awd	Spi2_miso	Adc_eoc	Adc_awd
0010	Comp1_out	Comp1_out	Comp2_out	Comp1_out
0011	Lptim_out	Lptim_out	Lptim_out	Lptim_out
0100	ADC_START	ADC_START	ADC_START	ADC_START
0101	Tim1_ch2	Tim2_ch2	Tim3_ch2	Tim4_ch2
0110	Tim1_ch3	Tim2_ch3	Tim3_ch3	Tim4_ch3
0111	Tim1_ch4	Tim2_ch4	Tim3_ch4	Tim4_ch4
1000	PLAIN10/PC7	PLAIN11/PB15	PLAIN13/PD9	PLAIN9/PC6
1001	PLAIN11/PB15	PLAIN12/PB0	PLAIN14/PB1	PLAIN10/PC7
1010	PLAIN14/PB1	PLAIN13/PD9	PLAIN15/PB2	PLAIN1/PB15
1011	PLAIN15/PB2	PLAIN14/PB1	PLAIN16/PB6	PLAIN14/PB1
1100	PLAIN16/PB6	PLAIN15/PB2	PLAIN17/PA10	PLAIN15/PB2
1101	PLAIN17/PA10	PLAIN16/PB6	PLAIN18/PA11	PLAIN16/PB6
1110	PLAIN19/PA15	PLAIN18/PA11	PLAIN19/PA15	PLAIN19/PA15
1111	PLAIN20/PB13	PLAIN19/PA15	PLAIN20/PB13	PLAIN20/PB13

每个 PLU 都向系统提供一个异步和同步（同步到 SYSCLK）的输出。固件可以通过读取 PLA_FR 寄存器来随时读取同步输出。PLU 输出可以直接从 LUT 获得，或从锁存的 D 型触发器输出，由 PLA_CFGRx 中的 OUTSEL 位控制。当一个 PLU 被禁用(PLA_CR 中的 PxEN 为 0)时，它的两个输出都将保持在逻辑 0。

D 触发器时钟可以从四个源中配置，由 PLA_CFGRx 中的 CLKSEL 字段选择。触发器时钟可以选择反转，使用 CLKINV 位。每个 PLU 都有以下选项来计时其触发器：

- CARRY_IN: 来自前一个 PLU 的携带(C)输入。第一个 PLU 使用来自比较器 3 的输出。
- MXA_INPUT: 对 PLU 的 A 的输入，由 MXA 寄存器字段定义。
- SYSCLK: 系统时钟。
- ALTCLK: PLU0 的计时器 1 溢出，PLU1 的计时器 2 溢出，PLU2 的计时器 3 溢出，PLU3 的计时器 4 溢出

当使用 D 触发器输出时，触发器可以通过写入 PLA_CFGRx 中的 RST 位 1 在任何时候重置为逻辑 0。输出将不会处于此重置状态(重置发生后 RST 返回到 0)。

PLU 输出也可以出现在选定的引脚上。到内部外设的 PLU 输出信号（另一个 PLU 输入除外）为 SYSCLK 同步。到任何 PLU 输入的 PLU 输出信号都不是 SYSCLK 同步的。

33.3.4 LUT 配置

每个 PLU 中的布尔逻辑函数由 LUT 确定，并且可以通过寄存器 PLA_CFGRx 中的 FNSEL 字段来更改。LUT 为一个 8 输入的多路复用器。FNSEL 的位映射到 8 个多路复用器输入，LUT 的输出由 A、B 和 C 输入的组合选择。

表 33-3 LUT 真值表

A Input	B Input	C Input	LUT Output
0	0	0	FNSEL.0
0	0	1	FNSEL.1
0	1	0	FNSEL.2
0	1	1	FNSEL.3
1	0	0	FNSEL.4
1	0	1	FNSEL.5
1	1	0	FNSEL.6
1	1	1	FNSEL.7

使用 LUT 可以实现任何 3 个输入的布尔逻辑函数。为了确定给定的逻辑函数编程成 FNSEL 的值，可以使用 LUT 真值表。例如，要实现布尔函数(A AND B)，对于 A 和 B 为 1 的任何组合，LUT 输出应该为 1，对于所有其他组合，LUT 输出应该为 0。表中的最后两行(对应于 FNSEL.7 和 FNSEL.6)满足此标准，因此 FNSEL 应该被编程为 1100000b 或 0xC0。

如果要实现函数(A XOR B), FNSSEL.2、FNSSEL.3、FNSSEL.4 和 FNSSEL.5 应该置为 1, FNSSEL.0、FNSSEL.1、FNSSEL.6 和 FNSSEL 7 应该置为 0, 因此 FNSSEL 应该被编程为 00111100b 或 0x3C。

33.4 PLA 寄存器

33.4.1 PLA 控制寄存器 (PLA_CR)

偏移地址: 0x00

复位值: 0x00000000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												P3EN	P2EN	P1EN	POEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								P3RIE	P3FIE	P2RIE	P2FIE	P1RIE	P1FIE	P0RIE	P0FIE
								rw							

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	P3EN	PLU3 使能 0: 不使能 PLU3, PLU3 输出保持逻辑 0 1: 使能 PLU3
18	P2EN	PLU2 使能 0: 不使能 PLU2, PLU2 输出保持逻辑 0 1: 使能 PLU2
17	P1EN	PLU1 使能 0: 不使能 PLU1, PLU1 输出保持逻辑 0 1: 使能 PLU1
16	P0EN	PLU0 使能 0: 不使能 PLU0, PLU0 输出保持逻辑 0 1: 使能 PLU0
15:8	保留	必须保持复位值。
7	P3RIE	PLU3 上升沿(使用 SYSCLK 同步)中断使能 0: 禁止, 对于 PLU3 的上升沿事件, 将不会产生中断 1: 使能, 对于 PLU3 的上升沿事件, 将产生中断

6	P3FIE	<p>PLU3 下降沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU3 的下降沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU3 的下降沿事件, 将产生中断</p>
5	P2RIE	<p>PLU2 上升沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU2 的上升沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU2 的上升沿事件, 将产生中断</p>
4	P2FIE	<p>PLU2 下降沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU2 的下降沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU2 的下降沿事件, 将产生中断</p>
3	PIRIE	<p>PLU1 上升沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU1 的上升沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU1 的上升沿事件, 将产生中断</p>
2	PIFIE	<p>PLU1 下降沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU1 的下降沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU1 的下降沿事件, 将产生中断</p>
1	PORIE	<p>PLU0 上升沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU0 的上升沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU0 的上升沿事件, 将产生中断</p>
0	POFIE	<p>PLU0 下降沿(使用 SYSCLK 同步)中断使能</p> <p>0: 禁止, 对于 PLU0 的下降沿事件, 将不会产生中断</p> <p>1: 使能, 对于 PLU0 的下降沿事件, 将产生中断</p>

33.4.2 PLA 标志寄存器 (PLA_SR)

偏移地址: 0x04

复位值: 0x0000 0000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								P3RIF	P3FIF	P2RIF	P2FIF	P1RIF	P1FIF	PORIF	POFIF
Res.								rc_w0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												P3OUT	P2OUT	P1OUT	POOUT
Res.												ro	ro	ro	ro

位/位域	名称	描述
31:24	保留	必须保持复位值。

23	P3RIF	PLU3 上升沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
22	P3FIF	PLU3 下降沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
21	P2RIF	PLU2 上升沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
20	P2FIF	PLU2 下降沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
19	P1RIF	PLU1 上升沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
18	P1FIF	PLU1 下降沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
17	P0RIF	PLU0 上升沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
16	P0FIF	PLU0 下降沿中断标志 0: 未检测到新的中断标志 1: 检测到新的中断标志, 该位由软件写 0 清零
15:4	保留	必须保持复位值。
3	P3OUT	PLU3 输出状态 该位代表 PLU3 经过 SYSCLK 同步后的输出状态
2	P2OUT	PLU2 输出状态 该位代表 PLU2 经过 SYSCLK 同步后的输出状态
1	P1OUT	PLU1 输出状态

该位代表PLU1经过SYSCLK同步后的输出状态

0 P0OUT PLU0 输出状态
该位代表 PLU0 经过 SYSCLK 同步后的输出状态

33.4.3 PLU0 配置寄存器 (PLA_CFGR0)

偏移地址: 0x0C

复位值: 0x0000 0000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MXA				MXB				FNSEL							
rw	rw	rw	rw	rw	rw	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
Res.								OUTSEL	OEN		RST	CLKINV	CKSEL		
								rw	rw			rw	rw	rw	rw

位/位域	名称	描述
31:28	MXA	PLU0 MUX 的 A 端口输入选择
27:24	MXB	PLU0 MUX 的 B 端口输入选择
23:16	FNSEL	PLU0 查找表(LUT)功能选择 LUT 是一个 8 输入的多功能复用器, 受 FNSEL 信号控制, 复用器的输入信号是 MXA、MXB 和 Carry-in 例如: FNSEL =0xC0,实现 MXA&MXB 功能; FNSEL =0xE4 实现(Carry & MXA) ((not Carry)&(MXB)
15:8	保留	必须保持复位值。
7	OUTSEL	PLU0 输出选择, 0: 选择 DFF 作为 PLU 的输出 1: 选择 LUT 作为 PLU 的输出
6	OEN	PLU0 输出使能, 将异步输出信号输出到 GPIO 管脚 0: 禁止 1: 使能
5:4	保留	必须保持复位值。
3	RST	PLU0 DFF 复位 写 1 将立即复位 DFF, 复位后该位自动清零

- 2 CLKINV PLU0 DFF 输入时钟反向
 0: 正常, 输入时钟不反向输入到 DFF
 1: 反向, 将输入时钟反向后输入到 DFF
- 1:0 CLKSEL PLU0 DFF 输入时钟选择
 00: CARRY_IN (carry-in 信号)
 01: MXA_INPUT(MXA 输入)
 10: SYSCLK
 11: ALTCLK(timer overflow)

33.4.4 PLU1 配置寄存器 (PLA_CFGR1)

偏移地址: 0x14

复位值: 0x0000 0000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MXA				MXB				FNSEL							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OUTSEL	OEN			RST	CLKINV	CKSEL	

位/位域	名称	描述
31:28	MXA	PLU1 MUX 的 A 端口输入选择
27:24	MXB	PLU1 MUX 的 B 端口输入选择
23:16	FNSEL	PLU1 查找表(LUT)功能选择 LUT 是一个 8 输入的多功能复用器, 受 FNSEL 信号控制, 复用器的输入信号是 MXA、MXB 和 Carry-in 例如: FNSEL =0xC0,实现 MXA&MXB 功能; FNSEL =0xE4 实现(Carry & MXA) ((not Carry)&(MXB)
15:8	保留	必须保持复位值。
7	OUTSEL	PLU1 输出选择, 0: 选择 DFF 作为 PLU 的输出 1: 选择 LUT 作为 PLU 的输出
6	OEN	PLU1 输出使能, 将异步输出信号输出到 GPIO 管脚

		0: 禁止 1: 使能
5:4	保留	必须保持复位值。
3	RST	PLU1 DFF 复位 写 1 将立即复位 DFF，复位后该位自动清零
2	CLKINV	PLU1 DFF 输入时钟反向 0: 正常，输入时钟不反向输入到 DFF 1: 反向，将输入时钟反向后输入到 DFF
1:0	CLKSEL	PLU1 DFF 输入时钟选择 00: CARRY_IN (carry-in 信号) 01: MXA_INPUT(MXA 输入) 10: SYSCLK 11: ALTCLK(timer overflow)

33.4.5 PLU2 配置寄存器 (PLA_CFGR2)

偏移地址: 0x1C

复位值: 0x0000 0000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
MXA				MXB				FNSEL									
										rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.								OUTSEL	OEN			RST	CLKINV	CKSEL			
								rw	rw			rw	rw	rw			

位/位域	名称	描述
31:28	MXA	PLU2 MUX 的 A 端口输入选择
27:24	MXB	PLU2 MUX 的 B 端口输入选择
23:16	FNSEL	PLU2 查找表(LUT)功能选择 LUT 是一个 8 输入的多功能复用器，受 FNSEL 信号控制，复用器的输入信号是 MXA、MXB 和 Carry-in 例如： FNSEL =0xC0,实现 MXA&MXB 功能； FNSEL =0xE4 实现(Carry & MXA) ((not Carry)&(MXB)

15:8	保留	必须保持复位值。
7	OUTSEL	PLU2 输出选择, 0: 选择 DFF 作为 PLU 的输出 1: 选择 LUT 作为 PLU 的输出
6	OEN	PLU2 输出使能, 将异步输出信号输出到 GPIO 管脚 0: 禁止 1: 使能
5:4	保留	必须保持复位值。
3	RST	PLU2 DFF 复位 写 1 将立即复位 DFF, 复位后该位自动清零
2	CLKINV	PLU2 DFF 输入时钟反向 0: 正常, 输入时钟不反向输入到 DFF 1: 反向, 将输入时钟反向后输入到 DFF
1:0	CLKSEL	PLU2 DFF 输入时钟选择 00: CARRY_IN (carry-in 信号) 01: MXA_INPUT(MXA 输入) 10: SYSCLK 11: ALTCLK(timer overflow)

33.4.6 PLU3 配置寄存器 (PLA_CFGR3)

偏移地址: 0x24

复位值: 0x0000 0000(该寄存器支持按字节读写)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MXA				MXB				FNSEL							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OUTSEL	OEN		RST	CLKINV	CKSEL		
								rw	rw		rw	rw	rw	rw	

位/位域	名称	描述
31:28	MXA	PLU3 MUX 的 A 端口输入选择
27:24	MXB	PLU3 MUX 的 B 端口输入选择

23:16	FNSEL	<p>PLU3 查找表(LUT)功能选择</p> <p>LUT 是一个 8 输入的多功能复用器，受 FNSEL 信号控制，复用器的输入信号是 MXA、MXB 和 Carry-in</p> <p>例如：</p> <p>FNSEL =0xC0,实现 MXA&MXB 功能；</p> <p>FNSEL =0xE4 实现(Carry & MXA) ((not Carry)&(MXB))</p>
15:8	保留	必须保持复位值。
7	OUTSEL	<p>PLU3 输出选择，</p> <p>0: 选择 DFF 作为 PLU 的输出</p> <p>1: 选择 LUT 作为 PLU 的输出</p>
6	OEN	<p>PLU3 输出使能，将异步输出信号输出到 GPIO 管脚</p> <p>0: 禁止</p> <p>1: 使能</p>
5:4	保留	必须保持复位值。
3	RST	<p>PLU3 DFF 复位</p> <p>写 1 将立即复位 DFF，复位后该位自动清零</p>
2	CLKINV	<p>PLU3 DFF 输入时钟反向</p> <p>0: 正常，输入时钟不反向输入到 DFF</p> <p>1: 反向，将输入时钟反向后输入到 DFF</p>
1:0	CLKSEL	<p>PLU3 DFF 输入时钟选择</p> <p>00: CARRY_IN (carry-in 信号)</p> <p>01: MXA_INPUT(MXA 输入)</p> <p>10: SYSCLK</p> <p>11: ALTCLK(timer overflow)</p>

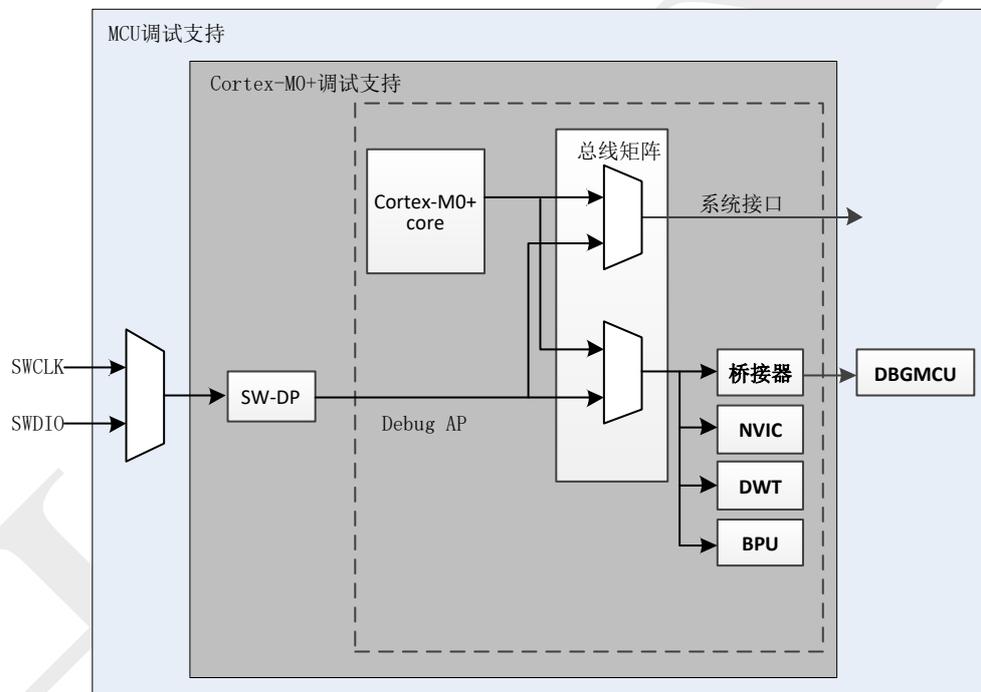
34 调试接口 (Debug Support)

34.1 概述

本产品使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指令(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

当调试器与 MCU 连接并进行调试时，将使用调试功能。

图 34-1 MCU 调试支持框图



Cortex®-M0+内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持，它包括：SW-DP（串行线）、DWT（数据观察点触发）和 BPU（断点单元）。

低功耗模式时，支持对外设时钟的控制。

34.2 参考文档

- Cortex®-M0+ Technical Reference Manual (TRM)
- Arm Debug Interface V5
- Arm CoreSight Design Kit revision r1p1 Technical Reference Manual

34.3 SWD 接口

34.3.1 SWD 接口引脚

两个 GPIO 可复用为 SWD 的接口的 SWDIO 和 SWCLK，所有封装都提供该引脚。

表 34-1 SWD 接口引脚分配

SWD 接口名称	功能说明	引脚分配
SWDIO	串行数据输入/输出	PA12
SWCLK	串行时钟	PA13

34.3.2 SWD 引脚分配

复位后,PA12 将复用为带上拉电阻 SWDIO,PA13 将复用为带下拉电阻 SWCLK。软件可禁止 SWD 接口,释放 PA12 和 PA13 为 GPIO,软件释放后,PA12 和 PA13 释放后,由 GPIO 模块控制,更多详细信息请参见 datasheet。

34.4 SWD 协议

34.4.1 协议简介

此同步串行协议使用 SWDIO 和 SWCLK 引脚,可以同时读取和写入两组寄存器组: DPACC 寄存器组和 APACC 寄存器组。

传输数据时,LSB 在前。

每次在协议中更改 SWDIO 的传输方向时,都会插入一个 SWCLK 时钟周期的转换时间,此时线路即不受主机驱动也不受目标驱动。可通过配置 SWCLK 频率调整转换时间。

34.4.2 SWD 协议序列

每个序列包含三个阶段:

- 1) 调试器发送数据包请求 (8bit)
- 2) MCU 发送的 ACK 响应 (3bit)
- 3) 调试器或 MCU 发送的数据传输阶段 (33bit)

表 34-2 数据包请求

位	名称	说明
0	Start	必须为 1
1	APnDP	0: DP 访问

位	名称	说明
		1: AP 访问
2	RnW	0: 写请求 1: 读请求
[4:3]	A[3:2]	DP 或 AP 寄存器的地址位域
5	Parity	{APnDP, RnW, A[3:2]} 这四位数的奇偶校验位
6	Stop	0
7	Park	调试器和 MCU 都不驱动, 此时存在上拉电阻, 读为 1

有关 DPACC 和 APACC 寄存器的详细说明, 请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间, 此时调试器和 MCU 都不会驱动 SWDIO。

表 34-3 ACK 响应

位	名称	说明
[0:2]	ACK	001: FAULT 010: WAIT 100: OK

当调试器读数据或接收到 WAIT 或 FAULT 响应时, ACK 响应后是转换时间。

表 34-4 数据传输

位	名称	说明
[0:31]	WDATA or RDATA	写入或读取的数据
32	Parity	32 个数据位的奇偶校验

当调试器读数据时, 数据传输后是转换时间。

34.4.3 SW-DP 状态机 (复位、空闲状态、ID 代码)

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 Arm 代码, 设置为 0x0BC11477 (相当于 Cortex®-M0+)。

- 在读取 ID 代码前, SW-DP 状态机时不工作的。
- 在上电复位后或 SWDIO 处于高电平超过 50 个时钟周期后, SW-DP 状态机处于复位状态。
- 如果在复位状态后 SWDIO 处于低电平至少两个时钟周期, SW-DP 状态机处于空闲状态。

复位状态后, 该状态机必须首先进入空闲状态, 然后对 SW-DP IDCODE 寄存器执行读访问。否则, 执行其他的访问时会接收 FAULT 相应。

34.4.4 DP 和 AP 的读写访问

不延迟对 DP 的读访问: 可以立即发送目标响应 (如果 ACK=OK), 也可以延迟

发送目标响应（如果 ACK=WAIT）。

延迟对 AP 的读访问：这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。

每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。

SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其他操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则会响应 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ALVRT 写入除外，这几项操作在写缓冲区已满时也会被接受。

由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

需要在写 CTRL/STAT 寄存器前发送 power-up 请求。否则，下一个操作（在内核上电后才有效的操作）立即执行，这将导致失败。

34.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器。

表 34-5 SW-DP 寄存器

A[3:2]	R/W	SELECT 寄存器的 CTRSXTL 位	寄存器	说明
00	读		IDCODE	制造商代码设置为 Cortex®-M0+ 的默认 ARM 代码：0x0BC11477（标识 SW-DP）
00	写		ALVRT	
01	读/写	0	DP-CTRL/STAT	目的： - 请求系统或调试上电 - 配置 AP 访问的传输操作 - 控制比较和验证操作 - 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行接口协议（如转换时间的持续时间）
10	读		READ RESEND	使读取的数据能够从损坏的调试器传输中恢复，无需重复原始的 AP 传输
10	写		SELECT	用于选择当前访问接口和活动的 4 字节寄存器窗口
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 访问时提供读取 AP 请

A[3:2]	R/W	SELECT 寄存器的 CTRSXTL 位	寄存器	说明
				求的结果)。 此读取缓冲区捕获 AP 中的数据, 显示为前一次读取的结果, 无需启动新操作。

34.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器。

有多个 AP 寄存器, 通过 A[3:2] 和 DP-SELECT 寄存器的值进行寻址。

表 34-6 SW-AP 寄存器

地址	A[3:2] 值	说明
0x0	00	必须保持复位值。
0x4	01	DP-CTRL/STAT 寄存器。用于: <ul style="list-style-type: none"> - 请求系统或调试上电 - 配置 AP 访问的传输操作 - 控制比较和验证操作 - 读取一些状态标志 (上溢和上电确认)
0x8	10	DP-SELECT 寄存器: 用于选择当前访问接口和活动的 4 字节寄存器窗口。 <ul style="list-style-type: none"> - 31:24 APSEL 选择当前 AP - 23:8 保留 - 7:4 APBBANKSEL 在当前 AP 上选择活动的 4 字节寄存器窗口 - 3:0 保留
0xC	11	DPREBUFF 寄存器: 用于通过调试器在执行一系列操作后获取最后结果 (无需请求新的 JTAG-DP 操作)

34.4.7 内核调试

通过内核调试寄存器调试内核。通过调试访问接口访问这些寄存器, 它由 4 个寄存器组成。

表 34-7 内核调试寄存器

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器: 此寄存器提供有关处理器状态的信息, 能够使内核进入调试停止状态并提供处理器步进供能。
DCRSR	17 位调试内核寄存器选择器寄存器: 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器: 此寄存器保存在寄存器与 DCRSR 寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试停止控制和状态寄存器: 此寄存器提供向量捕获和调试监视控制。

这些寄存器的复位操作只能通过上电复位，系统复位时不会复位。

为了在复位后立即使内核进入调试停止状态，必须使能调试和异常监视控制寄存器的 bit0 (VC_CORRESET)，使能调试停止控制和状态寄存器的 bit0 (C_DEBUGEN)。

34.5 BPU (断点单元)

Cortex®-M0+ BPU 实现提供 4 个断点寄存器。

34.5.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M Arm 和 Arm CoreSight Components Technical Reference Manual。

34.6 DWT (数据观察点)

Cortex®-M0+ DWT 实现提供 2 个观察点寄存器组。

34.6.1 DWT 功能

处理器观察点实现了数据地址和机缘 PC 的观察点功能 (即 PC 采样寄存器)，并支持比较器地址掩码，如 ARMv6-M Arm 中所述。

34.6.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。更多信息，请参见 ARMv6-M Arm。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

34.7 MCU 调试组件

MCU 调试组件为以下调试功能提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗和 I2C 的时钟控制

34.7.1 对低功耗模式的调试支持

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。DBG 模块提供支持，允许用户在低功耗模式下调试软件。

必须设置一些调试寄存器，以便在低功耗模式进行调试：

- 在 Sleep 和 LPSleep 模式下，FCLK 和 HCLK 仍有效，对调试功能没有任何限制。
- 在 Stop 模式下，必须先将 DBG_CR 寄存器的 DBG_STOP 位置 1，这样在 Stop 模式自动使能 HSI 时钟，以便在低功耗模式调试时为 FCLK 和 HCLK 提供时钟。

34.7.2 对定时器、看门狗和 I2C 的调试支持

断点期间，必须选择定时器、看门狗计数器是否继续计数：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。例如，使能看门狗时需要这种方式。

34.8 DBG 寄存器

34.8.1 数据寄存器 (DBG_DAT)

配置该寄存器能够存放调试使用的暂存数据。

偏移地址：0x00

POR 复位值：0x0000 0000

注意： 该寄存器只能通过POR复位，系统复位不会复位该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_DAT[31:16]															
rW	rW	rW	rW	rW	rW	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
DBG_DAT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:0	DBG_DAT	调试使用的暂存数据。

34.8.2 控制寄存器 (DBG_CR)

配置该寄存器能够在 STOP 模式下对 MCU 进行调试。

偏移地址：0x04

POR 复位值：0x0000 0000

注意： 该寄存器只能通过POR复位，系统复位不会复位该寄存器。

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

位/位域	名称	描述
31:30	保留	必须保持复位值。
0	DBG_STOP	调试Stop模式 1: 使能在Stop模式调试MCU功能(HCLK ON, FCLK ON)。 0: 禁止在Stop模式调试MCU功能(HCLK OFF, FCLK OFF)。

34.8.3 APB1 冻结寄存器 (DBG_APB_FZ1)

当 MCU 处于调试模式内核停止(CPU_HALT)时，配置该寄存器可使能或停止 APB1 外设的计数功能。

偏移地址：0x08

POR 复位值：0x0000 0000

注意： 该寄存器只能通过POR复位，系统复位不会复位该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DBG_LPTIM1_STOP	Res.										DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP
		rw											rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	DBG_LPTIM1_STOP	LPTIM1计数停止 0: 内核停止时，LPTIM1正常计数 1: 内核停止时，停止LPTIM1计数
28:19	保留	必须保持复位值。

18	DBG_IWDG_STOP	IWDG计数停止 0: 内核停止时, IWDG正常计数 1: 内核停止时, 停止IWDG计数
17	DBG_WWDG_STOP	WWDG计数停止 0: 内核停止时, WWDG正常计数 1: 内核停止时, 停止WWDG计数
16	DBG_RTC_STOP	RTC计数停止 0: 内核停止时, RTC正常计数 1: 内核停止时, 停止RTC计数
15:6	保留	必须保持复位值。
5	DBG_TIM7_STOP	TIM7计数停止 0: 内核停止时, TIM7正常计数 1: 内核停止时, 停止TIM7计数
4	DBG_TIM6_STOP	TIM6计数停止 0: 内核停止时, TIM6正常计数 1: 内核停止时, 停止TIM6计数
3	DBG_TIM5_STOP	TIM5计数停止 0: 内核停止时, TIM5正常计数 1: 内核停止时, 停止TIM5计数
2	DBG_TIM4_STOP	TIM4计数停止 0: 内核停止时, TIM4正常计数 1: 内核停止时, 停止TIM4计数
1	DBG_TIM3_STOP	TIM3计数停止 0: 内核停止时, TIM3正常计数 1: 内核停止时, 停止TIM3计数
0	DBG_TIM2_STOP	TIM2计数停止 0: 内核停止时, TIM2正常计数 1: 内核停止时, 停止TIM2计数

34.8.4 APB2 冻结寄存器 (DBG_APB_FZ2)

当 MCU 处于调试模式内核停止时, 配置该寄存器可使能或停止 APB2 外设的计数功能。

偏移地址: 0x0C

POR 复位值: 0x0000 0000

注意: 该寄存器只能通过POR复位，系统复位不会复位该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															DBG_TIM1_STOP
															rw

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	DBG_TIM1_STOP	TIM1计数停止 0: 内核停止时，TIM1正常计数 1: 内核停止时，停止TIM1计数

35 Cortex®-M0+内核寄存器

本章节主要介绍本文档中涉及到的内核寄存器，内核寄存器的介绍可参见《DUI0662B_cortex_m0p_r0p1_dgug.pdf》

35.1 系统控制寄存器

35.1.1 应用中断和复位控制寄存器

地址：0xE000 ED0C

复位值：0xFA05 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEY[15:0]															
rw	rw	rw	rw	rw	rw	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
ENDIANNESS	Res.												SYSRESETREQ	VECTCLRACTIVE	Res.
r														w	rw

位/位域	名称	描述
31:16	VECTKEY[15:0]	寄存器密钥 必须写入0x05FA到VECTKEY[15:0]，否则对当前寄存器的写操作被忽略。 读VECTKEY[15:0]时，固定为0xFA05。
15	ENDIANNESS	数据大小端 该位固定为0，表示采用小端排序。
14:3	保留	必须保持复位值
2	SYSRESETREQ	系统复位请求 0：无作用 1：请求系统复位
1	VECTCLRACTIVE	预留调试使用 写入必须为0，写其他值会有不可预知的影响。
0	保留	必须保持复位值

35.1.2 系统控制寄存器（SCR）

地址：0xE000 ED10

复位值：0x0000 0000

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

Res.										SEVON PEND	Res.	SLEEP DEEP	SLEEP ON EXIT	Res.
										rw		rw	rw	

位/位域	名称	描述
31:5	保留	必须保持复位值
4	SEVONPEND	NVIC中断挂起状态置位时是否生成唤醒事件 0: 未使能的NVIC中断不能唤醒CPU, 只有使能的中断或唤醒事件信号 1: 未使能的中断可以唤醒CPU
3	保留	必须保持复位值
2	SLEEPDEEP	CPU低功耗模式选择: 0: 睡眠模式 1: 深度睡眠模式
1	SLEEPONEXIT	低功耗进入方式选择 用于在 0: 退出最低优先级中断服务函数时不进入低功耗模式 1: 退出最低优先级中断服务函数时进入低功耗模式
0	保留	必须保持复位值

35.2 SysTick 寄存器

35.2.1 SysTick 控制和状态寄存器 (SYST_CSR)

地址: 0xE000 E010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	COUNT FLAG
Res.															r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.												CLK SOURCE	TICK INT	ENABLE		
												rw	rw	rw		

位/位域	名称	描述
31:17	保留	需保持复位值
16	COUNTFLAG	溢出标志位 上一次读CSR寄存器后, 计数器是否出现过溢出 (计数到0)。 此位在计数器从1减为0时置位 读该标志位时, 该标志位自动清0; SYST_CVR寄存器CURRENT[23:0]写入任何值清除该标志位。

		0: SysTick定时器未发生过溢出 1: SysTick定时器发生过溢出
15:3	保留	需保持复位值
2	CLKSOURCE	SysTick时钟源选择: 0: SysTick时钟源为外部参考时钟HCLK/8 1: SysTick使用内核时钟HCLK
1	TICKINT	SysTick中断使能 0: 禁止中断 1: 使能中断 SYST_CVR寄存器CURRENT[23:0]写入任何值对该位无影响。
0	ENABLE	SysTick使能 0: 禁止SysTick 1: 使能SysTick

35.2.2 SysTick 重载值寄存器 (SYST_RVR)

地址: 0xE000 E014

复位值: 0x00xxxxxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								RELOAD[23:16]							
15	14	13	12	11	10	9	8	Rw	rw	rw	rw	rw	rw	rw	rw
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	需保持复位值
23:0	RELOAD	SysTick定时器的重载计数值

35.2.3 SysTick 当前计数值寄存器 (SYST_CVR)

地址: 0xE000 E018

复位值: 0x00xxxxxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								CURRENT[23:16]							
15	14	13	12	11	10	9	8	rw	rw	rw	rw	rw	rw	rw	rw
CURRENT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	需保持复位值
23:0	CURRENT	计数器当前值 读此寄存器时获取SysTick计数器的当前计数值。 写任意值到该寄存器，清零该寄存器和SYST_CSR寄存器的COUNTFLAG标志。

35.2.4 SysTick 校准值寄存器 (SYST_CALIB)

地址: 0xE000 E01C

复位值: 0x00001F40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
NOREF	SKEW	Res.						TENMS[23:16]								
r	r							r	r	r	r	r	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TENMS[15:0]																
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位/位域	名称	描述
31	NOREF	参考时钟指示位: 0: 使用外部参考时钟HCLK/8 1: SysTick使用内部时钟HCLK
30	SKEW	TENMS精度指示 0: TENMS校准值代表精确的1ms 1: TENMS校准值代表粗略的1ms
29:24	保留	需保持复位值
23:0	TENMS	1ms校准计数值 时钟源选择参考时钟HCLK/8, HCLK配置为48MHz, SysTick定时器计数间隔为1ms的计数值

35.3 NVIC 寄存器

35.3.1 NVIC 中断使能寄存器 (NVIC_ISER)

地址: 0xE000 E100

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETENA[31:16]															
rw	rw	rw	rw	rw	rw	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
SETENA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	SETENA	每个位对应32个对应中断号的中断的使能 0: 不使能 1: 使能（写0无效）

35.3.2 NVIC 中断清使能寄存器 (NVIC_ICER)

地址: 0xE000 E180

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
rw	rw	rw	rw	rw	rw	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
CLRENA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	CLRENA	每个位对应32个对应中断号的中断的清使能 0: 不使能 1: 清中断（写0无效）

35.3.3 NVIC 中断挂起使能寄存器 (NVIC_ISPR)

地址: 0xE000 E200

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETPEND[31:16]															
rw	rw	rw	rw	rw	rw	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
SETPEND[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	SETPEND	每个位对应32个对应中断号的挂起使能 0: 不使能 1: 使能（写0无效）

35.3.4 NVIC 中断挂起清寄存器 (NVIC_ICPR)

地址: 0xE000 E280

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRPEND[31:16]															
rw	rw	rw	rw	rw	rw	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
CLRPEND[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:0	CLRPEND	每个位对应32个对应中断号的中断挂起清使能 0: 使能 1: 清中断挂起（写0无效）

35.3.5 NVIC 优先级寄存器 (NVIC_IPRn)

地址: 0xE000 E400~E000 E41C(IPRn:n 为 0~7)

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIN3		Res.						PRIN2		Res.					
rw	rw							rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIN1		Res.						PRIN0		Res.					
rw	rw							rw	rw						

位/位域	名称	描述
31:30	PRIN3	中断号N+3的优先级(对应PRI3、PRI7、PRI11、PRI15、PRI19、PRI23、PRI27、PRI31) PRI数值越低, 优先级越高
29:24	保留	保持为0
23:22	PRIN2	中断号N+3的优先级(对应PRI2、PRI6、PRI10、PRI14、PRI18、PRI22、PRI26、PRI30) PRI数值越低, 优先级越高
21:16	保留	保持为0
15:14	PRIN1	中断号N+2的优先级(对应PRI1、PRI5、PRI9、PRI13、PRI17、PRI21、PRI25、PRI29) PRI数值越低, 优先级越高
13:8	保留	保持为0
7:6	PRIN0	中断号N+1的优先级(对应PRI0、PRI4、PRI8、PRI12、PRI16、PRI20、PRI24、PRI28) PRI数值越低, 优先级越高
5:0	保留	保持为0

36 器件电子签名

电子签名存放在闪存存储器的系统存储区域，可以通过 SWD 或者 CPU 读取。它所包含的芯片识别信息在出厂时写入，用户固件或者外部设备可以读取电子签名，用于自动匹配不同配置的处理器。

产品唯一的身份标识非常合适：

- 用来作为序列码号；
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器的安全性；
- 用来激活带安全机制的自举过程。

96 位的产品唯一身份标识所提供的号码对任意一个序列微控制器，在任何情况下都是唯一的。用户在任何情况下，都无法修改这个身份标识。

这个 96 位的产品唯一身份标识，按照用户不同的用法，可以以字节、半字、字读取。

读取地址基址：0x1FFF 0000（UID[95:0]）。