

# CIU32F003x5

32 位 ARM Cortex-M0+ 微控制器

## 参考手册

RM1007



北京中电华大电子设计有限责任公司  
CEC Huada Electronic Design Co.,Ltd

# 声 明

本手册的版权属北京中电华大电子设计有限责任公司所有。任何未经授权对本手册进行复印、印刷、出版发行的行为，都将被视为是对北京中电华大电子设计有限责任公司版权的侵害。北京中电华大电子设计有限责任公司保留对此行为诉诸法律的权利。

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。虽然我们已经核对本手册的内容，但是差错有时候难以完全避免，所以，我们会对手册的内容进行定期的审查，并在下一版的文件中作必要的修改。建议您在最终设计前从华大电子获取本文档的最新版本。

## 目录

1	文档约定 .....	20
1.1	寄存器列表符号定义 .....	20
1.2	术语及定义 .....	20
2	简介 .....	22
3	引脚分配与功能描述 .....	23
3.1	引脚分配图 .....	23
3.2	引脚功能描述 .....	25
4	系统与存储器架构 .....	30
4.1	系统架构 .....	30
4.1.1	System Bus .....	31
4.1.2	Bus Matrix .....	31
4.1.3	AHB to APB 总线桥 .....	31
4.2	存储器 .....	32
4.2.1	SRAM .....	32
4.2.2	Flash .....	32
4.3	存储器映射 .....	32
4.4	启动配置 .....	35
5	Flash 存储器 (Flash) .....	36
5.1	简介 .....	36
5.2	Flash 主要特性 .....	36
5.3	Flash 功能描述 .....	36
5.3.1	Flash 存储器组成 .....	36
5.3.2	Flash 读取访问等待周期 .....	37
5.3.3	Flash 解锁 .....	38
5.3.4	User flash 区擦除操作 .....	39
5.3.5	User flash 区编程操作 .....	40
5.4	Option bytes 区操作说明 .....	40
5.4.1	选项字节组成 .....	40
5.4.2	选项字节擦除与编程 .....	41
5.4.3	选项字节加载 .....	42
5.5	Engineer 区 .....	43
5.6	Flash 安全保护机制 .....	43
5.6.1	Flash 读出保护 .....	44
5.6.2	Flash 写入保护 .....	46

5.7	<b>Flash 操作错误</b> .....	46
5.8	<b>Flash 中断</b> .....	47
5.9	<b>Flash 寄存器</b> .....	48
5.9.1	<b>Flash 访问控制寄存器 (FLASH_ACR)</b> .....	48
5.9.2	<b>Flash 控制解锁密钥寄存器 (FLASH_CRKEY)</b> .....	48
5.9.3	<b>Flash 选项字节解锁密钥寄存器 (FLASH_OPTKEY)</b> .....	49
5.9.4	<b>Flash 状态寄存器 (FLASH_SR)</b> .....	49
5.9.5	<b>Flash 控制寄存器 (FLASH_CR)</b> .....	50
5.9.6	<b>Flash 选项字节寄存器 1 (FLASH_OPTR1)</b> .....	52
5.9.7	<b>Flash 选项字节寄存器 2 (FLASH_OPTR2)</b> .....	53
5.9.8	<b>Flash 写保护区寄存器 (FLASH_WRP)</b> .....	53
6	<b>电源管理单元 (PMU)</b> .....	55
6.1	<b>电源</b> .....	55
6.2	<b>电源电压监测</b> .....	55
6.2.1	<b>上电复位和掉电复位</b> .....	55
6.2.2	<b>欠压复位</b> .....	56
6.2.3	<b>内部带隙基准电压</b> .....	56
6.3	<b>低功耗模式</b> .....	56
6.3.1	<b>Run</b> .....	58
6.3.2	<b>Sleep</b> .....	59
6.3.3	<b>Stop</b> .....	59
6.3.4	<b>Deepstop</b> .....	60
6.4	<b>PMU 寄存器</b> .....	62
6.4.1	<b>电源管理控制寄存器 (PMU_CR)</b> .....	62
6.4.2	<b>Flash 唤醒时间寄存器 (PMU_FLASH_WAKEUP)</b> .....	62
7	<b>复位和时钟控制器 (RCC)</b> .....	64
7.1	<b>复位</b> .....	64
7.1.1	<b>电源复位</b> .....	64
7.1.2	<b>系统复位</b> .....	64
7.2	<b>时钟</b> .....	65
7.2.1	<b>RCH 时钟</b> .....	66
7.2.2	<b>RCL 时钟</b> .....	67
7.2.3	<b>EXTCLK 时钟</b> .....	67
7.2.4	<b>系统时钟</b> .....	67
7.2.5	<b>外设异步时钟选择</b> .....	68
7.2.6	<b>IWDG 时钟</b> .....	68



7.2.7	时钟输出功能.....	68
7.2.8	基于 LPTIM1 的时钟测量.....	68
7.3	低功耗模式.....	69
7.4	<b>RCC 寄存器</b> .....	70
7.4.1	时钟控制/状态寄存器 (RCC_CSR1) .....	70
7.4.2	时钟配置寄存器 (RCC_CFG) .....	71
7.4.3	时钟中断使能寄存器 (RCC_IER) .....	73
7.4.4	时钟中断标志寄存器 (RCC_ISR) .....	73
7.4.5	时钟中断清除寄存器 (RCC_ICR) .....	74
7.4.6	GPIO 端口复位寄存器 (RCC_IOPRST) .....	75
7.4.7	AHB 外设复位寄存器 (RCC_AHBRSST) .....	76
7.4.8	APB 外设复位寄存器 1 (RCC_APBRSST1) .....	76
7.4.9	APB 外设复位寄存器 2 (RCC_APBRSST2) .....	77
7.4.10	I/O 端口时钟使能寄存器 (RCC_IOPEN) .....	79
7.4.11	AHB 外设时钟使能寄存器 (RCC_AHBEN) .....	79
7.4.12	APB 外设时钟使能寄存器 1 (RCC_APBEN1) .....	80
7.4.13	APB 外设时钟使能寄存器 2 (RCC_APBEN2) .....	81
7.4.14	外设异步时钟配置寄存器 (RCC_CLKSEL) .....	82
7.4.15	时钟控制/状态寄存器 (RCC_CSR2) .....	83
7.4.16	RCL 校准寄存器 (RCC_RCLCAL) .....	85
7.4.17	RCH 校准寄存器 (RCC_RCHCAL) .....	85
8	<b>通用输入/输出接口 (GPIO)</b> .....	87
8.1	简介 .....	87
8.2	<b>GPIO 主要特性</b> .....	87
8.3	<b>GPIO 功能描述</b> .....	87
8.3.1	通用 I/O (GPIO) .....	89
8.3.2	I/O 引脚功能复用器和映射 .....	89
8.3.3	I/O 端口配置寄存器 .....	90
8.3.4	I/O 端口数据寄存器 .....	91
8.3.5	I/O 数据位操作 .....	91
8.3.6	外部中断/唤醒 .....	91
8.3.7	输入配置 .....	92
8.3.8	输出配置 .....	93
8.3.9	复用功能配置 .....	94
8.3.10	复用功能选择 .....	94
8.3.11	模拟模式配置 .....	95

8.4	GPIO 寄存器 .....	96
8.4.1	GPIO 端口模式寄存器 (GPIOx_MODE) (x = A~C) .....	96
8.4.2	GPIO 端口输出类型寄存器 (GPIOx_OTYPE) (x = A~C) .....	96
8.4.3	GPIO 端口上拉/下拉寄存器 (GPIOx_PUPD) (x = A~C) .....	97
8.4.4	GPIO 端口输入寄存器 (GPIOx_IDR) (x = A~C) .....	98
8.4.5	GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A~C) .....	98
8.4.6	GPIO 端口置位/复位寄存器 (GPIOx_BSR) (x = A~C) .....	99
8.4.7	GPIO 复用功能低位寄存器 (GPIOx_AFL) (x = A~C) .....	99
8.4.8	GPIO 端口位复位寄存器 (GPIOx_BR) (x = A~C) .....	100
9	嵌套向量中断控制器 (NVIC) .....	101
9.1	简介 .....	101
9.2	主要特征.....	101
9.3	中断向量表.....	101
10	扩展中断和事件控制器 (EXTI) .....	103
10.1	简介 .....	103
10.2	EXTI 主要功能.....	103
10.3	EXTI 模块示意图.....	103
10.4	EXTI 的主要信号连接及功能.....	104
10.5	EXTI 通道及信号对应表.....	105
10.6	EXTI 功能描述.....	105
10.6.1	可配置通道.....	105
10.6.2	EXTI 直接通道 .....	106
10.6.3	I/O 多路选择器 .....	107
10.7	EXTI 寄存器.....	108
10.7.1	EXTI 上升沿触发选择寄存器 (EXTI_RTSR) .....	108
10.7.2	EXTI 下降沿触发选择寄存器 (EXTI_FTSR) .....	108
10.7.3	EXTI 中断挂起寄存器 (EXTI_PIR) .....	109
10.7.4	EXTI I/O 选择寄存器 (EXTI_EXTICR1) .....	110
10.7.5	EXTI 中断唤醒屏蔽寄存器 (EXTI_IMR) .....	112
10.7.6	EXTI 事件唤醒屏蔽寄存器 (EXTI_EMR) .....	113
11	循环冗余校验 (CRC) .....	115
11.1	简介 .....	115
11.2	CRC 主要特性.....	115
11.3	CRC 功能描述.....	115
11.3.1	CRC 框图 .....	115
11.3.2	CRC 操作说明 .....	115

11.4	CRC 寄存器.....	117
11.4.1	CRC 控制状态寄存器 (CRC_CSR) .....	117
11.4.2	CRC 结果寄存器 (CRC_RDR) .....	117
11.4.3	CRC 数据寄存器 (CRC_DR) .....	118
12	模数转换器 (ADC) .....	119
12.1	简介 .....	119
12.2	ADC 主要特性.....	119
12.3	ADC 功能描述.....	120
12.3.1	ADC 引脚和内部信号 .....	120
12.3.2	时钟源.....	121
12.3.3	使能/禁止控制.....	121
12.3.4	校准.....	122
12.3.5	转换通道.....	123
12.3.6	转换模式.....	123
12.3.7	启动/停止转换.....	126
12.3.8	转换时序示例.....	128
12.3.9	采样时间 .....	129
12.3.10	数据溢出 .....	130
12.3.11	自动等待模式.....	131
12.3.12	模拟看门狗.....	132
12.3.13	内部参考电压检测.....	133
12.4	ADC 中断.....	135
12.5	ADC 寄存器.....	136
12.5.1	ADC 控制寄存器 (ADC_CR) .....	136
12.5.2	ADC 配置寄存器 1 (ADC_CFG1) .....	137
12.5.3	ADC 配置寄存器 2 (ADC_CFG2) .....	139
12.5.4	ADC 中断和状态寄存器 (ADC_ISR) .....	140
12.5.5	ADC 中断使能寄存器 (ADC_IER) .....	141
12.5.6	ADC 采样时间寄存器 (ADC_SAMPT) .....	142
12.5.7	ADC 通道配置寄存器 (ADC_CHCFG) .....	143
12.5.8	ADC 模拟看门狗配置寄存器 (ADC_AWDGCR) .....	144
12.5.9	ADC 模拟看门狗监控电压阈值寄存器 (ADC_AWDGTR) .....	144
12.5.10	ADC 校准系数 (ADC_CALFACT) .....	145
12.5.11	ADC 数据寄存器 (ADC_DR) .....	145
13	比较器 (COMP) .....	147
13.1	简介 .....	147

13.2	COMP 主要特性 .....	147
13.3	COMP 功能描述 .....	148
13.3.1	COMP 结构框图 .....	148
13.3.2	引脚和内部信号 .....	148
13.3.3	使能/禁止控制 .....	149
13.3.4	窗口比较器 .....	149
13.3.5	输出极性 & 滤波 .....	150
13.3.6	迟滞 .....	151
13.4	COMP 中断 .....	151
13.5	COMP 寄存器 .....	153
13.5.1	COMP 控制寄存器 (COMP_CR) .....	153
13.5.2	COMP1 控制和状态寄存器 (COMP1_CSR) .....	154
13.5.3	COMP2 控制和状态寄存器 (COMP2_CSR) .....	156
14	高级控制定时器 (TIM1) .....	158
14.1	简介 .....	158
14.2	TIM1 主要特性 .....	158
14.3	TIM1 功能描述 .....	159
14.3.1	TIM1 框图 .....	159
14.3.2	TIM1 引脚和内部信号 .....	159
14.3.3	PSC 预分频器 .....	160
14.3.4	计数器 .....	161
14.3.5	重复计数器 .....	170
14.3.6	计数时钟选择 .....	171
14.3.7	比较通道 .....	172
14.3.8	比较输出模式 .....	173
14.3.9	强制输出模式 .....	174
14.3.10	PWM 模式 .....	175
14.3.11	清除 OCxREF 信号 .....	178
14.3.12	单脉冲模式 .....	179
14.3.13	互补输出及死区插入 .....	181
14.3.14	使用断路功能 .....	183
14.3.15	生成 6 步 PWM .....	186
14.3.16	霍尔传感器接口 .....	187
14.3.17	定时器同步从模式 .....	189
14.3.18	定时器同步主模式 .....	191
14.3.19	调试模式 .....	191

14.4	TIM1 中断.....	192
14.5	TIM1 寄存器.....	193
14.5.1	TIM1 控制寄存器 1 (TIM1_CR1) .....	193
14.5.2	TIM1 控制寄存器 2 (TIM1_CR2) .....	195
14.5.3	TIM1 从模式控制寄存器 (TIM1_SMC) .....	197
14.5.4	TIM1 中断使能寄存器 (TIM1_DIER) .....	198
14.5.5	TIM1 状态寄存器 (TIM1_SR) .....	200
14.5.6	TIM1 事件产生寄存器 (TIM1_EVTG) .....	201
14.5.7	TIM1 比较模式寄存器 1 (TIM1_CCM1) .....	203
14.5.8	TIM1 比较模式寄存器 2 (TIM1_CCM2) .....	205
14.5.9	TIM1 比较使能寄存器 (TIM1_CCEN) .....	206
14.5.10	TIM1 计数值寄存器 (TIM1_CNT) .....	208
14.5.11	TIM1 预分频寄存器 (TIM1_PSC) .....	208
14.5.12	TIM1 自动重载值寄存器 (TIM1_ARR) .....	209
14.5.13	TIM1 重复计数器寄存器 (TIM1_RCR) .....	209
14.5.14	TIM1 比较寄存器 1 (TIM1_CC1) .....	210
14.5.15	TIM1 比较寄存器 2 (TIM1_CC2) .....	210
14.5.16	TIM1 比较寄存器 3 (TIM1_CC3) .....	211
14.5.17	TIM1 比较寄存器 4 (TIM1_CC4) .....	211
14.5.18	TIM1 断路和死区寄存器 (TIM1_BDT) .....	212
14.5.19	TIM1 配置寄存器 (TIM1_CFG) .....	214
14.5.20	TIM1 轮换功能寄存器 1 (TIM1_AFI) .....	214
15	通用定时器 (TIM3) .....	217
15.1	简介 .....	217
15.2	TIM3 主要特性.....	217
15.3	TIM3 功能描述.....	218
15.3.1	TIM3 框图.....	218
15.3.2	TIM3 引脚和内部信号.....	218
15.3.3	PSC 预分频器 .....	218
15.3.4	计数器.....	220
15.3.5	计数时钟选择.....	222
15.3.6	捕获/比较通道.....	225
15.3.7	输入捕获模式.....	227
15.3.8	PWM 输入模式.....	227
15.3.9	比较输出模式.....	228
15.3.10	强制输出模式.....	229

15.3.11	PWM 模式.....	230
15.3.12	单脉冲模式.....	231
15.3.13	TIMx 输入异或功能.....	233
15.3.14	TIMx 定时器同步从模式.....	233
15.3.15	定时器同步主模式.....	236
15.3.16	调试模式.....	238
15.4	TIM3 中断.....	239
15.5	TIM3 寄存器.....	240
15.5.1	TIM 控制寄存器 1 (TIMx_CR1) .....	240
15.5.2	TIM 控制寄存器 2 (TIMx_CR2) .....	241
15.5.3	TIM 从模式控制寄存器 (TIMx_SMC) .....	243
15.5.4	TIM 中断使能寄存器 (TIMx_DIER) .....	244
15.5.5	TIM 状态寄存器 (TIMx_SR) .....	245
15.5.6	TIM 事件产生寄存器 (TIMx_EVTG) .....	247
15.5.7	TIM 捕获/比较模式寄存器 1 (TIMx_CCM1) -- (输出比较模式) .....	248
15.5.8	TIM 捕获/比较模式寄存器 1 (TIMx_CCM1) -- (输入捕获模式) .....	250
15.5.9	TIM 捕获/比较模式寄存器 2 (TIMx_CCM2) .....	251
15.5.10	TIM 捕获/比较使能寄存器 (TIMx_CCEN) .....	252
15.5.11	TIM 计数值寄存器 (TIMx_CNT) .....	254
15.5.12	TIM 预分频值寄存器 (TIMx_PSC) .....	255
15.5.13	TIM 自动重载值寄存器 (TIMx_ARR) .....	255
15.5.14	TIM 捕获/比较寄存器 1 (TIMx_CC1) -- (工作模式 0) .....	255
15.5.15	TIM 捕获/比较寄存器 1 (TIMx_CC1) -- (工作模式 1) .....	256
15.5.16	TIM 捕获/比较寄存器 2 (TIMx_CC2) -- (工作模式 0) .....	257
15.5.17	TIM 捕获/比较寄存器 2 (TIMx_CC2) -- (工作模式 1) .....	257
15.5.18	TIM 定时器输入选择寄存器 (TIMx_TISEL) .....	258
16	低功耗定时器 (LPTIM) .....	259
16.1	简介 .....	259
16.2	LPTIM 主要特性 .....	259
16.3	LPTIM 功能描述 .....	259
16.3.1	LPTIM 框图.....	259
16.3.2	LPTIM 内部信号.....	259
16.3.3	时钟源.....	260
16.3.4	计数器模式.....	260
16.3.5	定时器使能.....	261
16.3.6	调试模式.....	261

16.4	LPTIM 低功耗模式 .....	262
16.5	LPTIM 中断 .....	262
16.6	LPTIM 寄存器 .....	263
16.6.1	LPTIM 中断和状态寄存器 (LPTIMx_ISR) .....	263
16.6.2	LPTIM 中断清除寄存器 (LPTIMx_ICR) .....	264
16.6.3	LPTIM 中断使能寄存器(LPTIMx_IER) .....	264
16.6.4	LPTIM 配置寄存器(LPTIMx_CFG).....	265
16.6.5	LPTIM 控制寄存器 (LPTIMx_CR) .....	266
16.6.6	LPTIM 自动重载寄存器 (LPTIMx_ARR) .....	267
16.6.7	LPTIM 计数器寄存器 (LPTIMx_CNT) .....	267
17	红外控制模块 (IRTIM) .....	268
17.1	IRTIM 寄存器 .....	269
18	独立看门狗 (IWDG) .....	270
18.1	简介 .....	270
18.2	IWDG 主要特性 .....	270
18.3	IWDG 功能描述 .....	270
18.3.1	IWDG 框图 .....	270
18.3.2	寄存器访问保护 .....	271
18.3.3	溢出时间设置 .....	271
18.3.4	调试模式 .....	271
18.4	IWDG 低功耗模式 .....	272
18.5	IWDG 寄存器 .....	273
18.5.1	IWDG 控制寄存器 (IWDG_CR) .....	273
18.5.2	IWDG 配置寄存器 (IWDG_CFG) .....	273
18.5.3	IWDG 计数值寄存器 (IWDG_CNT) .....	274
19	SysTick 定时器 (SysTick) .....	275
19.1	简介 .....	275
19.2	SysTick 主要特性 .....	275
19.3	SysTick 功能描述 .....	275
19.4	SysTick 寄存器 .....	276
19.4.1	SysTick 控制和状态寄存器 (SysTick_CTRL) .....	276
19.4.2	SysTick 重载值寄存器 (SysTick_LOAD) .....	277
19.4.3	SysTick 当前计数值寄存器 (SysTick_VAL) .....	277
19.4.4	SysTick 校准值寄存器 (SysTick_CALIB) .....	277
20	I2C 接口 (I2C) .....	279
20.1	简介 .....	279

20.2	I2C 主要特性 .....	279
20.3	I2C 功能描述 .....	279
20.3.1	I2C 框图 .....	279
20.3.2	I2C 引脚 .....	280
20.3.3	I2C 时钟要求 .....	280
20.3.4	I2C 初始化 .....	280
20.3.5	软件复位 .....	282
20.3.6	通信流程 .....	282
20.3.7	错误条件 .....	285
20.4	I2C 中断 .....	286
20.5	I2C 寄存器 .....	287
20.5.1	控制寄存器 1 (I2Cx_CR1) .....	287
20.5.2	控制寄存器 2 (I2Cx_CR2) .....	288
20.5.3	地址寄存器 1 (I2Cx_ADDR1) .....	289
20.5.4	中断和状态寄存器 (I2Cx_ISR) .....	289
20.5.5	中断清除寄存器 (I2Cx_ICR) .....	291
20.5.6	接收数据寄存器 (I2Cx_RDR) .....	292
20.5.7	发送数据寄存器 (I2Cx_TDR) .....	293
21	通用异步收发器 (UART) .....	294
21.1	简介 .....	294
21.2	UART 主要特性 .....	294
21.3	UART 功能描述 .....	295
21.3.1	UART 框图 .....	295
21.3.2	UART 引脚 .....	295
21.3.3	UART 帧格式 .....	295
21.3.4	UART 校验控制 .....	296
21.3.5	UART 发送器 .....	297
21.3.6	UART 接收器 .....	298
21.3.7	UART 波特率生成 .....	302
21.3.8	UART 接收容差 .....	303
21.3.9	UART 单线半双工通信 .....	303
21.4	UART 中断 .....	304
21.5	UART 寄存器 .....	305
21.5.1	UART 控制寄存器 1 (UARTx_CR1) .....	305
21.5.2	UART 控制寄存器 2 (UARTx_CR2) .....	307
21.5.3	UART 控制寄存器 3 (UARTx_CR3) .....	308



21.5.4	UART 波特率分频寄存器 (UARTx_BRR)	309
21.5.5	UART 中断和状态寄存器 (UARTx_ISR)	310
21.5.6	UART 中断标志清零寄存器 (UARTx_ICR)	312
21.5.7	UART 接收数据寄存器 (UARTx_RDR)	313
21.5.8	UART 发送数据寄存器 (UARTx_TDR)	313
22	串行外设接口 (SPI)	315
22.1	简介	315
22.2	SPI 主要特性	315
22.3	SPI 功能说明	316
22.3.1	SPI 框图	316
22.3.2	SPI 引脚	316
22.3.3	单主机通信	316
22.3.4	多主机通信	317
22.3.5	从机通信	318
22.3.6	NSS 管理	318
22.3.7	通信模式	319
22.3.8	状态标志	320
22.3.9	错误标志	321
22.3.10	初始化 SPI	322
22.3.11	数据发送和接收	322
22.3.12	禁止 SPI	323
22.4	SPI 中断	323
22.5	SPI 寄存器	324
22.5.1	SPI 控制寄存器 1 (SPIx_CR1)	324
22.5.2	SPI 控制寄存器 2 (SPIx_CR2)	326
22.5.3	SPI 中断和状态寄存器 (SPIx_ISR)	327
22.5.4	SPI 中断标志清零寄存器 (SPIx_ICR)	328
22.5.5	SPI 数据寄存器 (SPIx_DR)	329
23	调试接口 (DBG)	330
23.1	简介	330
23.2	DBG 功能描述	330
23.3	调试组件 (DBG)	330
23.3.1	对低功耗模式的调试支持	330
23.3.2	对定时器和看门狗的调试支持	331
23.4	DBG 寄存器	332
23.4.1	控制寄存器 (DBG_CR)	332

---

23.4.2	APB 冻结寄存器 1 (DBG_APB_FZ1)	332
23.4.3	APB 冻结寄存器 2 (DBG_APB_FZ2)	333
24	器件电子签名	335
24.1	产品唯一身份识别 (UID) 寄存器 (96 位)	335
24.2	芯片产品型号参数寄存器	335
24.3	Flash 空间参数寄存器	335
24.4	SRAM 空间参数寄存器	335
25	版本历史	336
26	联系方式	337

## 表目录

表 3-1	引脚分配和功能描述.....	25
表 3-2	端口复用功能映射.....	28
表 4-1	存储器地址范围.....	33
表 4-2	外设寄存器地址范围.....	34
表 5-1	Flash 存储器组成.....	36
表 5-2	HCLK 时钟频率与 Flash 读取访问等待周期的对应关系 .....	37
表 5-3	选项字节组成.....	41
表 5-4	校验错误加载默认值对应关系.....	43
表 5-5	产品信息.....	43
表 5-6	出厂校准参数.....	43
表 5-7	RDP 保护等级配置.....	45
表 5-8	WRP 保护区域配置.....	46
表 5-9	Flash 中断.....	47
表 5-10	Flash 基地址.....	48
表 6-1	低功耗模式概述.....	57
表 6-2	低功耗模式下可用外设 <sup>(1)</sup> .....	58
表 6-3	Sleep 说明.....	59
表 6-4	Stop 说明 .....	60
表 6-5	Deepstop 说明 .....	60
表 6-6	PMU 基地址.....	62
表 7-1	时钟关系表.....	66
表 7-2	外设异步时钟来源.....	68
表 7-3	RCC 基地址列表.....	70
表 8-1	GPIO 端口配置表 <sup>(1)</sup> .....	88
表 8-2	GPIO 基地址列表 .....	96
表 9-1	中断向量表 <sup>(1)</sup> .....	101
表 10-1	EXTI 内部信号 .....	104
表 10-2	EXTI 通道及信号对应表.....	105
表 10-3	EXTI 屏蔽功能 .....	105
表 10-4	EXTI 基地址 .....	108
表 11-1	CRC 基地址.....	117
表 12-1	ADC 输入引脚.....	120
表 12-2	ADC 内部信号 .....	120
表 12-3	触发方式及极性.....	126

表 12-4	外部硬件触发源.....	126
表 12-5	内部带隙基准电压校准参数.....	133
表 12-6	ADC 中断.....	135
表 12-7	ADC 基地址.....	136
表 13-1	COMP1 正相输入信号.....	148
表 13-2	COMP1 反相输入信号.....	148
表 13-3	COMP2 正相输入信号.....	149
表 13-4	COMP2 反相输入信号.....	149
表 13-5	COMP 中断.....	152
表 13-6	COMP 基地址.....	153
表 14-1	TIM1 引脚.....	159
表 14-2	TIM1 内部信号.....	160
表 14-3	互补输出信号 OCx 和 OCxN 的配置方式（MOEN 位为 1）.....	181
表 14-4	具有断路功能的互补通道 OCx 和 OCxN 的输出控制位（MOEN 位为 0）.....	184
表 14-5	中断请求.....	192
表 14-6	TIM1 基地址.....	193
表 15-1	TIM3 引脚.....	218
表 15-2	TIM3 内部信号.....	218
表 15-3	中断请求.....	239
表 15-4	TIMx 基地址（x=3）.....	240
表 16-1	LPTIM 内部信号.....	259
表 16-2	低功耗模式对 LPTIM 的影响.....	262
表 16-3	中断请求.....	262
表 16-4	LPTIMx 基地址（x=1）.....	263
表 17-1	IRTIM 基地址列表.....	269
表 18-1	IWDG 溢出周期表.....	271
表 18-2	低功耗模式对 IWDG 的影响.....	272
表 18-3	IWDG 基地址.....	273
表 20-1	I2C 引脚.....	280
表 20-2	中断请求.....	286
表 20-3	I2Cx 基地址（x=1）.....	287
表 21-1	UART 校验控制.....	296
表 21-2	UART 中断请求.....	304
表 21-3	UARTx 基地址列表（x=1、2）.....	305
表 22-1	SPI 引脚.....	316
表 22-2	NSS 配置.....	319

---

表 22-3	通信模式配置.....	319
表 22-4	SPI 中断请求.....	323
表 22-5	SPI 基地址列表.....	324
表 23-1	DBG 基地址 .....	332
表 25-1	版本更改履历.....	336

## 图目录

图 3-1	CIU32F003F5P6-TSSOP20 引脚分配.....	23
图 3-2	CIU32F003F5U6-QFN20 引脚分配.....	23
图 3-3	CIU32F003W5S6-SOP16 引脚分配.....	24
图 3-4	CIU32F003W15S6-SOP16 type1 引脚分配.....	24
图 3-5	CIU32F003J5S6-SOP8 引脚分配.....	24
图 4-1	系统架构示意图.....	31
图 4-2	存储器映射.....	33
图 6-1	电源结构框图.....	55
图 6-2	POR/PDR 和 BOR 阈值.....	56
图 6-3	低功耗模式转换框图.....	57
图 7-1	复位电路框图.....	64
图 7-2	时钟树.....	66
图 8-1	I/O 基本结构.....	88
图 8-2	I/O 引脚输入配置（浮空/上拉/下拉）.....	92
图 8-3	I/O 引脚输出配置.....	93
图 8-4	I/O 引脚复用功能配置.....	94
图 8-5	I/O 引脚高阻态模拟配置.....	95
图 10-1	EXTI 框图.....	104
图 10-2	可配置通道&直接通道的事件输出框图.....	106
图 10-3	可配置通道的中断输出框图.....	106
图 10-4	EXTI 多路复用器.....	107
图 11-1	CRC 框图.....	115
图 11-2	CRC 计算单元框图.....	116
图 11-3	输入数据反转框图.....	116
图 11-4	输出数据反转框图.....	116
图 12-1	ADC 结构框图.....	120
图 12-2	ADC 输入通道.....	123
图 12-3	ADC 转换模式.....	124
图 12-4	停止正在进行的转换.....	128
图 12-5	单次扫描转换模式，软件触发.....	128
图 12-6	循环扫描转换模式，软件触发.....	129
图 12-7	单次扫描转换模式，硬件触发.....	129
图 12-8	循环扫描转换模式，硬件触发.....	129
图 12-9	ADC 转换时序.....	130

图 12-10 数据溢出处理示例 <sup>(1)</sup> .....	131
图 12-11 自动等待模式 <sup>(1)</sup> .....	132
图 12-12 模拟看门狗阈值更新 <sup>(1)</sup> .....	133
图 12-13 V <sub>BGR</sub> 输入通道结构 .....	134
图 13-1 COMP 结构框图 .....	148
图 13-2 窗口比较器结构 <sup>(1)</sup> .....	150
图 13-3 COMP 输出滤波 <sup>(1)</sup> .....	151
图 13-4 COMP 迟滞 .....	151
图 14-1 TIM1 框图 .....	159
图 14-2 预分频器分频系数由 1 变为 2 时生效的时序图 .....	160
图 14-3 预分频器分频系数由 1 变为 4 时的计数器时序图 .....	161
图 14-4 递增计数模式下配置预分频器为 1 分频上溢事件时序图 .....	163
图 14-5 递增计数模式下配置预分频器为 4 分频上溢事件时序图 .....	163
图 14-6 递增计数模式下计数器时序图, ARPE=0 时更新事件 .....	164
图 14-7 递增计数模式下计数器时序图, ARPE=1 时更新事件 .....	164
图 14-8 递减计数模式下配置预分频器为 1 分频下溢事件时序图 .....	165
图 14-9 递减计数模式下配置预分频器为 4 分频下溢事件时序图 .....	166
图 14-10 递减计数模式下计数器时序图, ARPE=0 时更新事件 .....	166
图 14-11 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图 .....	168
图 14-12 中心对齐模式下配置预分频器为 2 分频下溢事件时序图 .....	168
图 14-13 中心对齐模式下, 产生下溢事件, ARPE=1 时的时序图 .....	169
图 14-14 中心对齐模式下, 产生上溢事件, ARPE=1 时的时序图 .....	169
图 14-15 不同模式下设置不同 TIM1_RCR 寄存器值时的更新频率示例 .....	170
图 14-16 选择内部时钟源时的计数时序图 .....	171
图 14-17 选择时钟模式 1 时的时序图 .....	171
图 14-18 比较通道 .....	172
图 14-19 比较通道的输出连接图 (通道 1、通道 2 和通道 3) .....	173
图 14-20 比较通道的输出连接图 (通道 4) .....	173
图 14-21 比较输出模式, 翻转 OC1 (OCxPE 位为 0) .....	174
图 14-22 递增计数下的边沿对齐模式 PWM 波形 (TIM1_ARR 寄存器为 8) .....	176
图 14-23 递减计数下的边沿对齐模式 PWM 波形 (TIM1_ARR 寄存器为 8) .....	177
图 14-24 中心对齐模式 PWM 波形 (TIM1_ARR 寄存器为 8) .....	178
图 14-25 清除 TIM1 的 OCxREF .....	179
图 14-26 单脉冲输出波形 .....	179
图 14-27 单脉冲模式示例 .....	180
图 14-28 带死区插入的互补输出 .....	182

图 14-29 死区时间大于负脉冲宽度的死区波形.....	182
图 14-30 死区时间大于正脉冲宽度的死区波形.....	182
图 14-31 断路电路概述.....	183
图 14-32 BRK 上断路事件的输出状态（OSSI 位为 1） .....	185
图 14-33 BRK 使能后的 PWM 输出状态（OSSI=0） .....	186
图 14-34 COM 事件生成 6 步 PWM 的示例（OSSR=1） .....	187
图 14-35 霍尔传感器接口的示例.....	189
图 14-36 复位模式下的控制时序.....	190
图 14-37 门控模式下的控制时序.....	191
图 14-38 触发模式下的控制时序.....	191
图 15-1 TIMx 框图（x=3） .....	218
图 15-2 预分频器分频系数由 1 变为 2 时生效的时序图.....	219
图 15-3 预分频器分频系数由 1 变为 4 时的计数器时序图.....	219
图 15-4 递增计数模式下配置预分频器为 1 分频上溢事件时序图.....	221
图 15-5 递增计数模式下配置预分频器为 4 分频上溢事件时序图.....	221
图 15-6 递增计数模式下计数器时序图，ARPE=0 时更新事件 .....	222
图 15-7 递增计数模式下计数器时序图，ARPE=1 时更新事件 .....	222
图 15-8 选择内部时钟源时的计数时序图.....	223
图 15-9 选择 TI2 作为时钟模式 1 触发输入的连接示例（上升沿计数） .....	224
图 15-10 选择时钟模式 1 时的时序图.....	224
图 15-11 捕获/比较通道.....	225
图 15-12 捕获/比较通道的输入连接图（通道 1） .....	226
图 15-13 捕获/比较通道的输出连接图（通道 1） .....	226
图 15-14 PWM 输入模式的时序 .....	228
图 15-15 比较输出模式，翻转 OC1（OCxPE 位为 0） .....	229
图 15-16 PWM 波形（TIMx_ARR 寄存器为 8） .....	231
图 15-17 单脉冲输出波形.....	231
图 15-18 单脉冲模式示例.....	232
图 15-19 测量 3 个信号边沿之间的时间间隔.....	233
图 15-20 复位模式下的控制时序.....	234
图 15-21 门控模式下的控制时序.....	235
图 15-22 触发模式下的控制时序.....	236
图 15-23 主从定时器连接示例图.....	236
图 15-24 使用 TIM3 的 OC1REF 信号对 TIM1 实施门控控制 .....	237
图 15-25 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM1 .....	238
图 15-26 使用 TIM3 的门控信号对 TIM1 实施门控控制 .....	238



图 16-1	LPTIMx 框图 (x=1)	259
图 16-2	单次计数模式使能说明	261
图 17-1	IRTIM 框图	268
图 18-1	IWDG 框图	270
图 20-1	I2Cx 框图 (x=1)	279
图 20-2	I2C 初始化流程	281
图 20-3	使能时钟低电平延长发送数据	283
图 20-4	禁止时钟低电平延长发送数据	283
图 20-5	使能时钟低电平延长接收数据	284
图 20-6	禁止时钟低电平延长接收数据	285
图 21-1	UARTx 框图 (x=1、2)	295
图 21-2	帧格式示意图	296
图 21-3	可编程停止位示例	297
图 21-4	发送过程 TC 和 TXE 的动作	298
图 21-5	起始位检测	299
图 21-6	数据采样 (OBS=0)	301
图 22-1	SPIx 框图 (x=1)	316
图 22-2	单主单从应用	316
图 22-3	单主多从应用	317
图 22-4	多主多从应用	317
图 22-5	通信模式时序	320
图 22-6	主从模式 BUSY 状态	321
图 22-7	主模式通信时序	323
图 22-8	从模式通信时序	323

## 1 文档约定

### 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 (r)	软件只能读取这些位
只写 (w)	软件只能写入该位。读取该位时将返回复位值
读取/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/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/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“0”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，读为“0”，写入无效

### 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：

页	512 字节大小的 Flash 空间为一页
字	32 位数据
半字	16 位数据
字节	8 位数据

---

选项字节 (Option Byte)	存储于 Flash 中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
Run	运行模式
Sleep	睡眠模式
Stop	停止模式
Deepstop	深度停止模式

## 2 简介

CIU32F003 系列超值安全 MCU 基于 ARM Cortex-M0+内核，最高频率可达 48MHz，具有多达 24K bytes Flash、3K bytes SRAM，支持 TSSOP20、QFN20、SOP16、SOP8 等多种封装，内部集成 1Msps ADC、2 个低功耗比较器、最高速率可达 24Mbps 的 SPI、2 个 UART、I2C、多种定时器等丰富的外设资源。

CIU32F003 系列超值安全 MCU 应用场景：

- 照明
- 电子烟
- 电动车仪表盘
- BMS 等其他 32 位替代 8/16 位的场景

### 3 引脚分配与功能描述

#### 3.1 引脚分配图

本芯片包含 TSSOP20、QFN20、SOP16、SOP8 多种封装类型，引脚分配见下图所示。

图 3-1 CIU32F003F5P6-TSSOP20 引脚分配

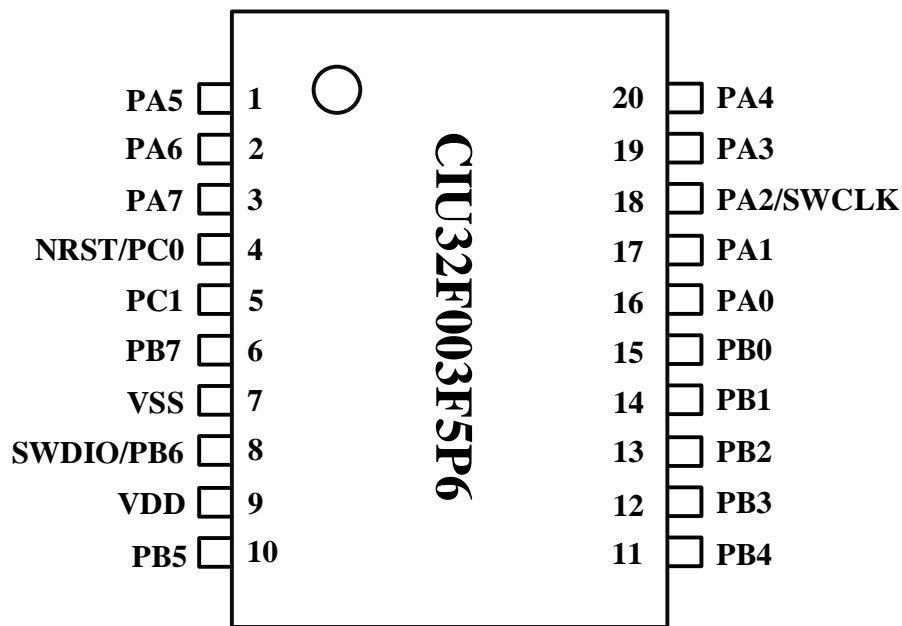
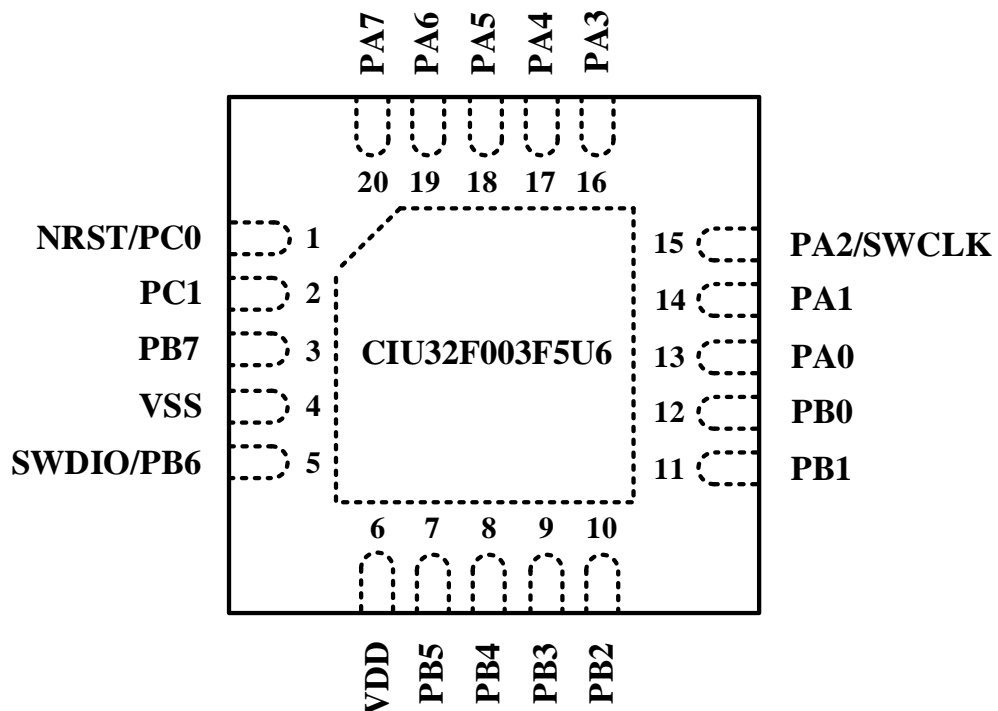
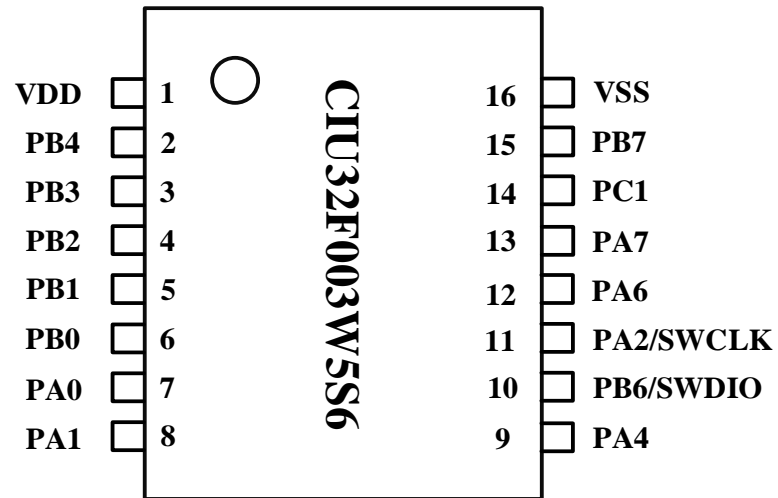
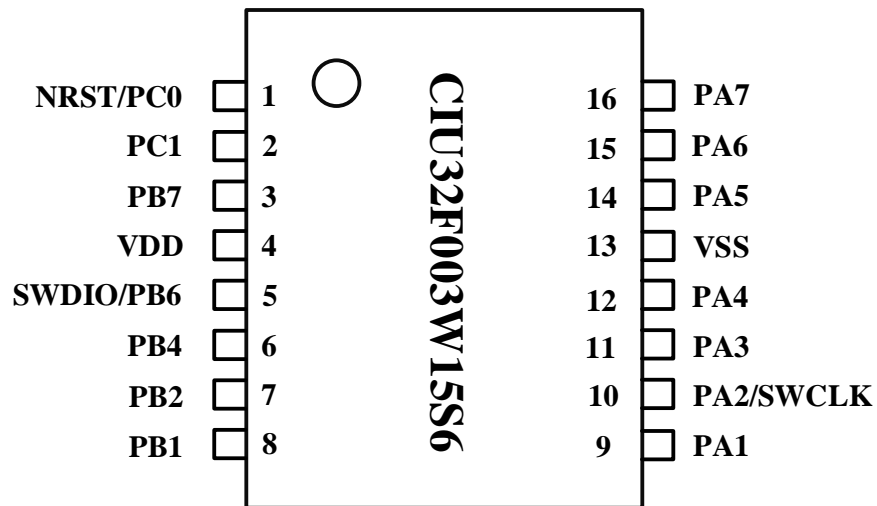
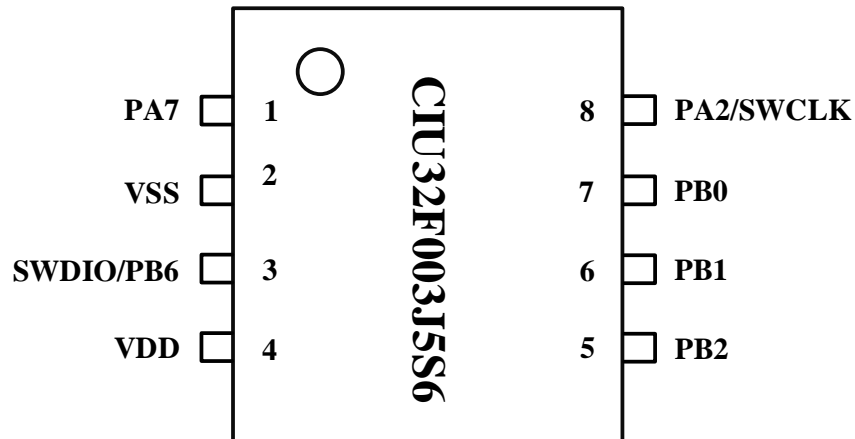


图 3-2 CIU32F003F5U6-QFN20 引脚分配



**图 3-3 CIU32F003W5S6-SOP16 引脚分配**

**图 3-4 CIU32F003W15S6-SOP16 type1 引脚分配**

**图 3-5 CIU32F003J5S6-SOP8 引脚分配**


### 3.2 引脚功能描述

表 3-1 引脚分配和功能描述

引脚编号					引脚名称	引脚类型	附加功能	复用功能
TSSOP20	QFN20	SOP16	SOP16 type1	SOP8				
1	18	-	14	-	PA5	I/O	-	TIM1_CH2 TIM3_CH1 UART2_TX
2	19	12	15	-	PA6	I/O	ADC_IN3	SPI1_NSS UART1_TX TIM3_CH3 SPI1_SCK UART2_RX
3	20	13	16	1	PA7	I/O	ADC_IN4	SPI1_MOSI UART1_TX TIM3_CH2 UART1_RX MCO IR_OUT
4	1	-	1	-	PC0	I/O	NRST <sup>(1)</sup>	SWDIO UART1_TX
5	2	14	2	-	PC1	I/O	EXTCLK	SPI1_MISO TIM1_CH2 TIM3_CH1
6	3	15	3	-	PB7	I/O	-	SPI1_MOSI UART1_RX TIM1_CH1N TIM1_CH2N TIM1_CH4
7	4	16	13	2	VSS	G	-	-
8	5	10	5	3	PB6	I/O	ADC_IN6	SWDIO UART1_TX SPI1_MISO UART2_TX I2C1_SDA MCO
9	6	1	4	4	VDD	P	-	-

引脚编号					引脚名称	引脚类型	附加功能	复用功能
TSSOP20	QFN20	SOP16	SOP16 type1	SOP8				
10	7	-	-	-	PB5	I/O	-	SPI1_NSS UART1_RX TIM1_CH3N TIM3_CH3
11	8	2	6	-	PB4	I/O	-	UART1_TX TIM1_BKIN TIM3_CH4 I2C1_SDA IR_OUT
12	9	3	-	-	PB3	I/O	ADC_IN5	TIM1_CH1N COMP1_OUT I2C1_SCL
13	10	4	7	5	PB2	I/O	-	SPI1_SCK TIM1_CH1 TIM1_CH1N TIM1_CH3 UART2_RX
14	11	5	8	6	PB1	I/O	ADC_IN0 COMP1_INM COMP1_INP COMP2_INP <sup>(2)</sup>	SPI1_NSS TIM1_CH1N TIM1_CH2N TIM1_CH4 MCO
15	12	6	-	7	PB0	I/O	ADC_IN7 COMP1_INP COMP2_INP <sup>(2)</sup>	SPI1_SCK UART1_TX TIM1_CH2 TIM3_CH1
16	13	7	-	-	PA0	I/O	-	SPI1_MOSI TIM1_CH1 TIM3_CH1 TIM1_CH2N TIM1_CH3N
17	14	8	9	-	PA1	I/O	-	SPI1_MISO TIM1_CH2 TIM1_CH3
18	15	11	10	8	PA2	I/O	-	SWCLK



引脚编号					引脚名称	引脚类型	附加功能	复用功能
TSSOP20	QFN20	SOP16	SOP16 type1	SOP8				
								UART1_RX COMP1_OUT I2C1_SCL COMP2_OUT
19	16	-	11	-	PA3	I/O	ADC_IN1 COMP1_INP <sup>(3)</sup> COMP2_INP	UART1_TX TIM1_CH3N TIM3_CH3 UART2_RX
20	17	9	12	-	PA4	I/O	ADC_IN2 COMP1_INP <sup>(3)</sup> COMP2_INM COMP2_INP	UART1_RX TIM1_CH2N TIM3_CH2 UART2_TX

1. 根据选项字节配置决定，上电复位时 PC0 默认为 NRST 功能。
2. PB0 和 PB1 可作为 COMP2\_INP，通过 COMP1\_INP 位和 COMP2\_INPMOD 位配置选择。
3. PA3 和 PA4 可作为 COMP1\_INP，通过 COMP2\_INP 位和 COMP1\_INPMOD 位配置选择。

表 3-2 端口复用功能映射

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	SPI1_MOSI	-	TIM1_CH1	TIM3_CH1	TIM1_CH2N	TIM1_CH3N	-	-
PA1	SPI1_MISO	-	TIM1_CH2	-	TIM1_CH3	-	-	-
PA2	SWCLK	UART1_RX	-	-	COMP1_OUT	-	I2C1_SCL	COMP2_OUT
PA3	-	UART1_TX	TIM1_CH3N	TIM3_CH3	-	UART2_RX	-	-
PA4	-	UART1_RX	TIM1_CH2N	TIM3_CH2	-	UART2_TX	-	-
PA5	-	-	TIM1_CH2	TIM3_CH1	-	UART2_TX	-	-
PA6	SPI1_NSS	UART1_TX	-	TIM3_CH3	SPI1_SCK	UART2_RX	-	-
PA7	SPI1_MOSI	UART1_TX	-	TIM3_CH2	-	UART1_RX	MCO	IR_OUT
PB0	SPI1_SCK	UART1_TX	TIM1_CH2	TIM3_CH1	-	-	-	-
PB1	SPI1_NSS	-	TIM1_CH1N	TIM1_CH2N	TIM1_CH4	-	MCO	-
PB2	SPI1_SCK	-	TIM1_CH1	TIM1_CH1N	TIM1_CH3	UART2_RX	-	-
PB3	-	-	TIM1_CH1N	-	COMP1_OUT	-	I2C1_SCL	-
PB4	-	UART1_TX	TIM1_BKIN	TIM3_CH4	-	-	I2C1_SDA	IR_OUT
PB5	SPI1_NSS	UART1_RX	TIM1_CH3N	TIM3_CH3	-	-	-	-

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PB6	SWDIO	UART1_TX	-	-	SPI1_MISO	UART2_TX	I2C1_SDA	MCO
PB7	SPI1_MOSI	UART1_RX	TIM1_CH1N	TIM1_CH2N	TIM1_CH4	-	-	-
PC0	SWDIO	UART1_TX	-	-	-	-	-	-
PC1	SPI1_MISO	-	TIM1_CH2	TIM3_CH1	-	-	-	-

## 4 系统与存储器架构

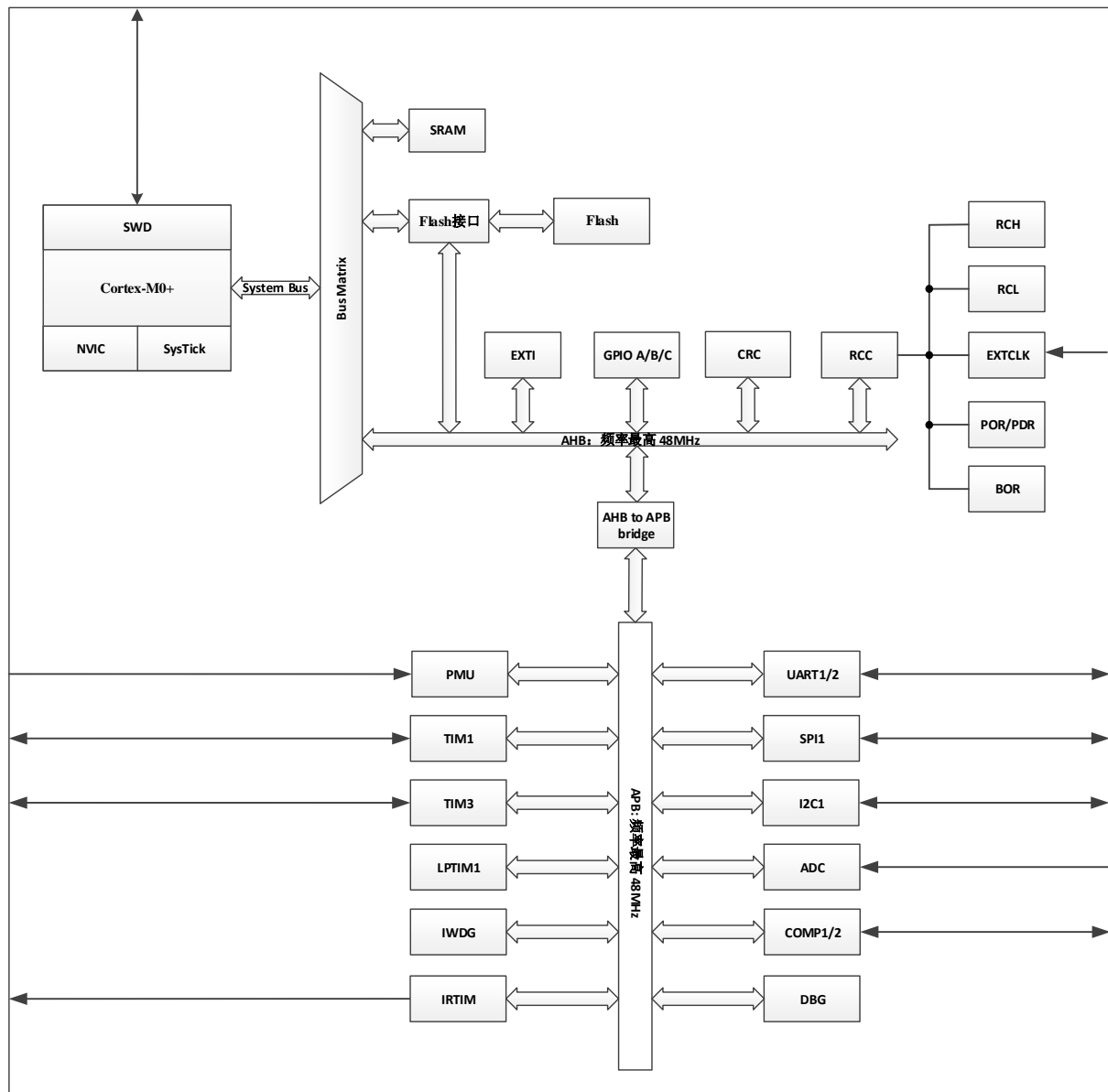
### 4.1 系统架构

主系统包括：

- 一个主机：
  - Cortex-M0+内核
- 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 4-1 系统架构示意图



#### 4.1.1 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。

#### 4.1.2 Bus Matrix

总线矩阵包含主机 CPU 和三个从机（Flash、SRAM、带有 AHB-to-APB 总线桥的 AHB）。

#### 4.1.3 AHB to APB 总线桥

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

## 4.2 存储器

### 4.2.1 SRAM

SRAM 总容量 3KB，支持按字节、半字、字访问。支持以最高系统时钟频率（48MHz）进行寻址，无需插入等待周期。

### 4.2.2 Flash

Flash 存储器区域划分如下：

- User flash 区：高达 24KB，包含应用程序和用户数据，由用户管理；
- Option bytes 区：用于存储外设和存储器保护配置的选项字节；
- Engineer 区：用于存储产品信息及出厂校准参数。

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

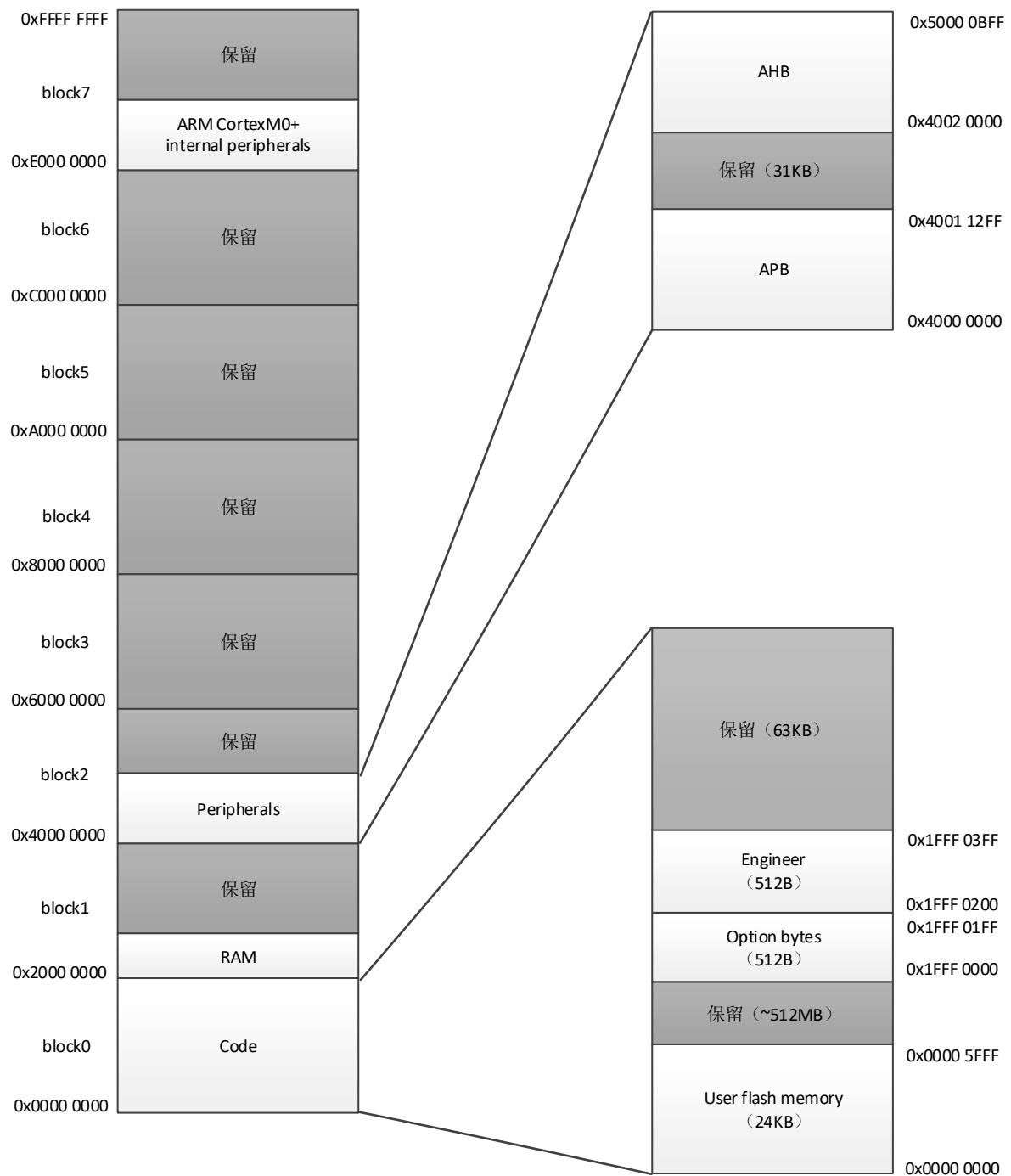
## 4.3 存储器映射

程序存储器、数据存储器、寄存器排列在一个地址连续的 4GB 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，而地址最大的存储单元保存该字的最高位数据。

可寻址的存储空间分为 8 个块，每个块为 512 MB。

图 4-2 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”，对“保留区”的访问会产生总线错误，具体存储映射参见下表。

表 4-1 存储器地址范围

存储类型	地址范围	大小	存储内容
SRAM	0x2000 0C00 - 0x3FFF FFFF	~512MB	Reserved
	0x2000 0000 - 0x20000BFF	3KB	SRAM
Flash	0x1FFF 0400 - 0x1FFF FFFF	63KB	Reserved

	0x1FFF 0200 - 0x1FFF 03FF	512B	Engineer
	0x1FFF 0000 - 0x1FFF 01FF	512B	Option bytes
	0x0000 6000 - 0x1FFE FFFF	~512MB	Reserved
	0x0000 0000 - 0x0000 5FFF	24KB	User flash memory

表 4-2 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
AHB	0x5000 0C00 - 0x5FFF FFFF	~256MB	Reserved
	0x5000 0800 - 0x5000 0BFF	1KB	GPIOC
	0x5000 0400 - 0x5000 07FF	1KB	GPIOB
	0x5000 0000 - 0x5000 03FF	1KB	GPIOA
	0x4002 3400 - 0x4FFF FFFF	~256MB	Reserved
	0x4002 3000 - 0x4002 33FF	1KB	CRC
	0x4002 2400 - 0x4002 2FFF	3KB	Reserved
	0x4002 2000 - 0x4002 23FF	1KB	FLASH
	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 0000 - 0x4002 0FFF	4KB	Reserved
APB	0x4001 5C00 - 0x4001 FFFF	41KB	Reserved
	0x4001 5800 - 0x4001 5BFF	1KB	DBG
	0x4001 3C00 - 0x4001 57FF	7KB	Reserved
	0x4001 3800 - 0x4001 3BFF	1KB	UART1
	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 0200 - 0x4001 03FF	1KB	COMP1/2
	0x4001 01B0 - 0x4001 01FF		Reserved
	0x4001 0000 - 0x4001 01AF		IRTIM



总线	地址范围	大小	外设
	0x4000 8000 - 0x4000 FFFF	32KB	Reserved
	0x4000 7C00 - 0x4000 7FFF	1KB	LPTIM1
	0x4000 7400 - 0x4000 7BFF	2KB	Reserved
	0x4000 7000 - 0x4000 73FF	1KB	PMU
	0x4000 5800 - 0x4000 6FFF	6KB	Reserved
	0x4000 5400 - 0x4000 57FF	1KB	I2C1
	0x4000 4800 - 0x4000 53FF	3KB	Reserved
	0x4000 4400 - 0x4000 47FF	1KB	UART2
	0x4000 3400 - 0x4000 43FF	4KB	Reserved
	0x4000 3000 - 0x4000 33FF	1KB	IWDG
	0x4000 0800 - 0x4000 2FFF	10KB	Reserved
	0x4000 0400 - 0x4000 07FF	1KB	TIM3
	0x4000 0000 - 0x4000 03FF	1KB	Reserved

#### 4.4 启动配置

复位后，芯片从 User flash 启动。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。

## 5 Flash 存储器（Flash）

### 5.1 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。

### 5.2 Flash 主要特性

- 高达 24KB 的用户存储空间，页大小：512 字节
- 32bits 位宽编程，8/16/32bits 位宽读取
- 支持页擦除、批量擦除
- 可配置 2 种读出保护等级（RDP）
- 可配置的写入保护区域（WRP）

### 5.3 Flash 功能描述

#### 5.3.1 Flash 存储器组成

Flash 存储器支持 32bits 编程，支持 8/16/32bits 位宽读访问，可存储指令和数据。

Flash 存储器的组成如下：

- User flash 区：用于存储用户程序和数据，存储空间最大为 24KB，包含 48 个页（Page），每页 512 字节；
- Option bytes 区：用于存储外设和存储器保护配置的选项字节；
- Engineer 区：用于存储产品信息及出厂校准参数。

表 5-1 Flash 存储器组成

区域	地址	空间大小 (字节)	说明
User flash	0x0000 0000 – 0x0000 01FF	512	Page 0
	...	...	...
	0x0000 2E00 – 0x0000 2FFF	512	Page 23
	0x0000 3000 – 0x0000 31FF	512	Page 24
	...	...	...

区域	地址	空间大小 (字节)	说明
	0x0000 5E00 – 0x0000 5FFF	512	Page 47
Option bytes	0x1FFF 0000 – 0x1FFF 01FF	512	选项字节
Engineer	0x1FFF 0200 – 0x1FFF 03FF	512	产品信息、出厂校准参数

### 5.3.2 Flash 读取访问等待周期

Flash 访问控制寄存器 (FLASH\_ACR) 中的 LATENCY 位, 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见下表。

表 5-2 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 配置
$HCLK \leq 24MHz$	0 HCLK	LATENCY = 0
$HCLK > 24MHz$	1 HCLK	LATENCY = 1

改变 HCLK 频率与修改访问等待周期的流程如下:

- 提高 HCLK 频率的配置步骤:
  - 1) 配置 FLASH\_ACR 寄存器中的 LATENCY 位, 增大 Flash 读取访问的等待周期;
  - 2) 读取 LATENCY 位, 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 \(RCC\\_CFG\)](#) 中的 SYSW[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) 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 \(RCC\\_CFG\)](#) 中的 SYSW[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) 确认系统时钟已切换完成;
  - 3) 配置 FLASH\_ACR 寄存器中的 LATENCY 位, 减小 Flash 读取访问的等待周期;
  - 4) 读取 LATENCY 位, 检查等待周期已配置成功。

### 5.3.3 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。

#### Flash 控制寄存器解锁

复位后，*Flash 控制寄存器 (FLASH\_CR)* 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\_CRKEY 寄存器写入密钥 1：0xE57A 1A85；
- 2) 向 FLASH\_CRKEY 寄存器写入密钥 2：0x7C6E 8391；
- 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，否则将产生总线错误。出错后，直到再次复位后，才能重新对 FLASH\_CR 寄存器进行解锁。

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。

#### Flash 选项字节解锁

复位后，Flash 选项字节处于写保护锁定状态。要对选项字节进行更新，需首先进行解锁操作。

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\_OPTKEY 寄存器写入密钥 1：0x6A89 4D7B；
- 3) 向 FLASH\_OPTKEY 寄存器写入密钥 2：0x7C31 1F5A；
- 4) 检查 FLASH\_CR 寄存器中的 OPTLOCK 位，当该位清零时，表明 Flash 选项字节已解锁。

解锁完成后，才能对 Option bytes 区执行编程与擦除操作。

密钥必须严格按照顺序写入，否则将产生总线错误。出错后，直到再次复位后，才能重新对 Flash 选项字节进行解锁。

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。

#### 5.3.4 User flash 区擦除操作

User flash 区支持以下擦除方式:

- 页擦除 (512 字节)
- 批量擦除

##### User flash 区页擦除步骤

对 User flash 区进行页擦除操作, 可遵循以下步骤:

- 1) 检查 Flash 状态寄存器 (FLASH\_SR) 中的 BSY 标志, 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;
- 4) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 10, 进入页擦除模式;
- 5) 向擦除的目标页任意地址写任意数据(必须是 32 位数据), 启动 Flash 擦除, 同时 BSY 标志将自动置 1;
- 6) 查询并等待 BSY 标志清 0, 表明擦除操作已完成;
- 7) 如果要对多个页执行擦除操作, 可重复执行步骤 5 到 6;
- 8) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 00, 退出页擦除模式;
- 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。

##### User flash 区批量擦除步骤

批量擦除用于擦除整个 User flash 区域, 可遵循以下步骤:

- 1) 检查 Flash 状态寄存器 (FLASH\_SR) 中的 BSY 标志, 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;
- 4) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 11, 进入批量擦除模式;
- 5) 向 User flash 区写任意数据 (必须是 32 位数据), 启动 Flash 擦除, 同时 BSY 标志将自动置 1;
- 6) 查询并等待 BSY 标志清 0, 表明擦除操作已完成;

- 7) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 00，退出擦除模式；
- 8) 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。

### 5.3.5 User flash 区编程操作

Flash 存储器支持按照 32bits 位宽执行编程操作。编程目标地址必须按照字访问，否则写入数据无效，产生总线错误。

#### 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，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；
- 4) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 01，进入 Flash 编程模式；
- 5) 向目标地址写入 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 OP\_MODE[1:0]位域为 00，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。

## 5.4 Option bytes 区操作说明

### 5.4.1 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，用户可根据需求对选项字节进行配置。

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 5-3 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														
0x1FFF 0000	Reserved	NRST_SWD_MODE		Reserved		BOR_LEVEL		BOR_EN		RDP																				
0x1FFF 0004	Reserved																IWDG_STOP	Reserved			RST_STOP									
0x1FFF 0008																	Reserved										WRP			

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\_WRP*：写保护区寄存器

#### 5.4.2 选项字节擦除与编程

选项字节未解锁时，向 Option bytes 区执行擦除或编程操作，相应操作不会启动。

##### 选项字节擦除步骤

对 Option bytes 区的擦除遵循以下步骤：

- 1) 检查 FLASH\_SR 中的 BSY 标志位，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；
- 4) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0] 位域为 10，进入页擦除模式；
- 5) 向 Option bytes 区任意地址写任意数据（必须是 32 位数据），启动选项字节擦除，同时 BSY 标志将自动置 1；

- 6) 查询并等待 BSY 标志清 0，表明擦除操作已完成；
- 7) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 00，退出页擦除模式；
- 8) 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；
- 9) 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。

### 选项字节编程步骤

Option bytes 区仅支持 32bits 编程，低 16bits 存放选项字节，高 16bits 存放选项字节的反码。编程时，选项字节的反码由用户计算并写入。

对选项字节的编程遵循以下步骤：

- 1) 检查 *Flash 状态寄存器 (FLASH\_SR)* 中的 BSY 标志位，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；
- 4) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 01，进入 Flash 编程模式；
- 5) 依次向 Option bytes 区地址写入待编程的数据（软件需保证数据高 16bit 与低 16bit 符合正反码校验），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) 重复步骤 5 和 6，完成对选项字节的编程；
- 8) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 00，退出 Flash 编程模式；
- 9) 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；
- 10) 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。

在完成对选项字节的编程后，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。

#### 5.4.3 选项字节加载

要使 Option bytes 区中选项字节生效，可通过电源复位或系统复位对选项字节进



行加载。

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，错误标志 OPTVERR 置 1，对应的选项字节加载为默认值，具体见下表：

表 5-4 校验错误加载默认值对应关系

选项字节	对应默认值
FLASH_OPTR1	0x0000 0001
FLASH_OPTR2	0x0000 0021
FLASH_WRP	0x0000 003F

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选项字节加载寄存器。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。

## 5.5 Engineer 区

Engineer 区存储产品信息和出厂校准参数，出厂写入，软件可读不可写。

表 5-5 产品信息

存储地址	长度（字）	参数描述
0x1FFF 0340	3	96 位的产品唯一身份识别码
0x1FFF 03A4	1	芯片型号： 0x00000004: CIU32F003
0x1FFF 03A8	1	Flash 空间大小，单位：Kbytes 0x00000018: 24
0x1FFF 03AC	1	SRAM 空间大小，单位：Kbytes 0x00000003: 3

表 5-6 出厂校准参数

存储地址	bit[31:16]	bit[15:0]	参数描述
0x1FFF 0204	~RCHCAL	RCHCAL	RCH 出厂校准值
0x1FFF 020C	~RCLCAL	RCLCAL	RCL 出厂校准值
0x1FFF 03C0	~BGR_CAL	BGR_CAL	BGR 出厂校准值
0x1FFF 03CC	ADC_COMPENSATION_VALUE		ADC 补偿值

## 5.6 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）：通过配置不同 RDP 保护等级，对存储器的操作权限进行限制。
- 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。

### 5.6.1 Flash 读出保护

#### RDP 保护权限

Flash 读出保护（RDP），共有两种保护等级：

##### RDP 0:

- 从 User flash 启动
  - User flash 前 4KB：允许取指、读取、编程、擦除
  - User flash 4KB 后：允许取指、读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - Engineer 区：允许读取
- 调试接口
  - User flash 前 4KB：允许取指、读取、编程、擦除
  - User flash 4KB 后：允许取指、读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - Engineer 区：允许读取

##### RDP 1:

- 从 User flash 启动
  - User flash 前 4KB：允许取指、读取
  - User flash 4KB 后：允许取指、读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - Engineer 区：允许读取
- 调试接口
  - User flash 前 4KB：禁止执行任何操作

- User flash 4KB 后：禁止执行任何操作
- Option bytes 区：允许读取、擦除、编程
- Engineer 区：允许读取

**注意：** 通过 SWD 接口升级到 RDP1 或 RDP1 下连接 SWD 后，需执行一次上电复位。

对于 Flash 存储器，当 Flash 控制器检测到无权限的非法操作时，非法操作会被立刻终止，并产生总线访问错误。

### RDP 保护等级配置

选项字节中的 RDP[7:0]位域，用于配置 RDP 保护等级， FLASH\_OPTR1 寄存器中 RDPRP 位指示当前保护等级。当选项字节加载生效后，RDP 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 5-7 RDP 保护等级配置

RDP 保护等级	RDP[7:0]配置值	RDPRP
RDP 0	0xAA（默认值）	0
RDP 1	除 0xAA 以外其他值	1

RDP 保护等级从 RDP0 升级到 RDP1 不会影响存储器中的内容。

### RDP 保护等级降级操作步骤

对 RDP 保护等级进行降级操作，可遵循以下步骤：

- 1) 检查 Flash 状态寄存器（FLASH\_SR）中的 BSY 标志位，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；
- 4) 执行 Option bytes 区擦除操作，等待擦除完成；
- 5) 配置 FLASH\_CR 寄存器中的 OP\_MODE[1:0]位域为 01，进入 Flash 编程模式；
- 6) 向 Option bytes 区选项字节 OPTR1 的 RDP 位写入 0xAA，BSY 标志将自动置 1；
- 7) 查询并等待 BSY 标志清 0，表明降级操作完成。

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，硬件将继续自动执行以下操作：

- 擦除整个 User flash 区中的内容；
- 选项字节 OPTR1 的 RDP 位更新为 0xAA，其他配置位更新为选项字节加载寄存器中的值；
- 选项字节 OPTR2 更新为选项字节加载寄存器中的值；
- 关闭 WRP 保护区域：WRP[5:0]更新为 0x3F；

### 5.6.2 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，对受保护区域执行擦除或编程操作不会执行，写保护错误标志 WRPERR 置 1。

在 User flash 区，可配置的最小保护区域为 4K，各保护区域相互独立。选项字节中的 WRP 用于配置 WRP 保护区域的范围，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](#)。WRP 保护区域配置见下表：

表 5-8 WRP 保护区域配置

寄存器配置	WRP 保护区域
WRP[0]	0x0000 0000 ~0x0000 0FFF
WRP[1]	0x0000 1000 ~0x0000 1FFF
WRP[2]	0x0000 2000 ~0x0000 2FFF
WRP[3]	0x0000 3000 ~0x0000 3FFF
WRP[4]	0x0000 4000 ~0x0000 4FFF
WRP[5]	0x0000 5000 ~0x0000 5FFF

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。

## 5.7 Flash 操作错误

对 User flash 操作的过程中，可能会出现 WRPERR 错误标志。当错误标志 WRPERR 未被清除时，擦除与编程操作不会启动。

**写保护错误标志 WRPERR**

发生以下情况时，相应操作不会启动，错误标志 WRPERR 置 1：

- 擦除受 WRP 保护的区域时；
- 编程受 WRP 保护的区域时。

#### 校验错误标志 OPTVERR

发生以下情况时，错误标志 OPTVERR 置 1：

- 当 Option bytes 加载值与反其码不匹配时；
- 出厂校准参数与其反码不匹配时。

## 5.8 Flash 中断

- Flash 操作完成中断

配置 FLASH\_CR 寄存器中 EOPIE 位为 1，使能该中断，当 Flash 完成编程操作、擦除操作时，FLASH\_SR 寄存器中 EOP 标志将置 1 并触发中断。

- Flash 操作异常中断

配置 FLASH\_CR 寄存器中 OPERRIE 位为 1，使能该中断，当 WRPERR 位置 1 时，触发中断。

表 5-9 Flash 中断

中断事件	中断标志	中断使能控制位	中断清除方式
Flash 操作完成	EOP	EOPIE	向 EOP 标志位写 1
Flash 操作异常	WRPERR	OPERRIE	向 WRPERR 标志位写 1

## 5.9 Flash 寄存器

Flash 寄存器支持 32 位访问。

表 5-10 Flash 基地址

外设	基地址
Flash	0x4002 2000

### 5.9.1 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.															LATENCY
															rw

位/位域	名称	描述
31:1	保留	写入无效
0	LATENCY	Flash 读取访问等待周期 0: 0 等待周期 1: 1 等待周期

### 5.9.2 Flash 控制解锁密钥寄存器（FLASH\_CRKEY）

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
CRKEY[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
CRKEY[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	CRKEY[31:0]	<p>用于解锁Flash控制寄存器（FLASH_CR）</p> <p>解锁FLASH_CR寄存器中的控制位。</p> <p>解锁时按顺序写入以下密钥：</p> <p>密钥1：0xE57A 1A85</p> <p>密钥2：0x7C6E 8391</p> <p>解锁步骤及注意事项详见：<a href="#">Flash控制寄存器解锁</a>。</p>
------	-------------	--

### 5.9.3 Flash 选项字节解锁密钥寄存器（FLASH\_OPTKEY）

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]	<p>用于解锁选项字节</p> <p>解锁时按顺序写入以下密钥：</p> <p>密钥1：0x6A89 4D7B</p> <p>密钥2：0x7C31 1F5A</p> <p>解锁步骤及注意事项详见：<a href="#">Flash选项字节解锁</a>。</p>

### 5.9.4 Flash 状态寄存器（FLASH\_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.							EOP	Res.							BSY
							rc_wl								r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTVERR	Res.										WRPERR	Res.			
r											rc_wl				

位/位域	名称	描述
31:25	保留	写入无效

24	EOP	Flash操作完成标志 该标志写1清0，写0无效。 当Flash完成编程操作、擦除操作时，此位将置1。 0: Flash操作未完成 1: Flash操作已完成
23:17	保留	写入无效
16	BSY	Flash操作状态标志 当对Flash执行操作时，此位将自动置1，在操作完成后，此位将自动清0。 0: Flash处于空闲状态 1: Flash处于忙状态，操作正在执行
15	OPTVERR	Option bytes校验错误标志 当Option bytes和出厂校准参数与其反码不匹配时，该标志置1。 0: 正常 1: 发生校验错误
14:5	保留	写入无效
4	WRPERR	写保护错误标志 当擦除或编程受WRP保护的区域时，该标志置1。 该标志写1清0，写0无效。 0: 正常 1: 发生写保护错误
3:0	保留	写入无效

### 5.9.5 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.				OPERRIE	EOPIE	Res.							
rs	rs					rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



Res.												OP_MODE[1:0]		Res.
												rw	rw	

位/位域	名称	描述
31	LOCK	<p>FLASH_CR寄存器锁定控制</p> <p>此位置1后，FLASH_CR寄存器将被写保护锁定。</p> <p>当检测到正确的解锁序列时，此位自动清0。</p> <p>解锁步骤及注意事项详见：<a href="#">Flash控制寄存器解锁</a>。</p>
30	OPTLOCK	<p>选项字节锁定控制</p> <p>此位置1后，选项字节处于写保护锁定状态。</p> <p>当检测到正确的解锁序列时，此位自动清0。</p> <p>解锁步骤及注意事项详见：<a href="#">Flash选项字节解锁</a>。</p>
29:26	保留	写入无效
25	OPERRIE	<p>编程/擦除操作异常中断使能控制</p> <p>此位置1后，当FLASH_SR寄存器中的WRPERR置1时，将触发中断。</p> <p>0：禁止</p> <p>1：使能</p>
24	EOPIE	<p>编程/擦除操作完成中断使能控制</p> <p>此位置1后，当FLASH_SR寄存器中的EOP标志置1时，将触发中断。</p> <p>0：禁止</p> <p>1：使能</p>
23:3	保留	写入无效
2:1	OP_MODE[1:0]	<p>Flash操作模式选择</p> <p>00：退出编程/擦除模式</p> <p>01：编程模式</p> <p>10：页擦除模式</p> <p>11：批量擦除模式</p>
0	保留	写入无效

### 5.9.6 Flash 选项字节寄存器 1 (FLASH\_OPTR1)

偏移地址：0x20

复位值：bit: 00000000 00000000 0XX00XXX 0000000X

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.	NRST_SWD_MODE[1:0]		Res.		BOR_LEVEL[1:0]		BOR_EN	Res.							RDPRP
	r	r			r	r	r								r

位/位域	名称	描述
31:15	保留	写入无效
14:13	NRST_SWD_MODE[1:0]	NRST/SWDIO引脚功能选择 0x: PC0: NRST PB6: SWDIO 10: PC0: GPIO PB6: SWDIO 11: PC0: SWDIO PB6: GPIO
12:11	保留	写入无效
10:9	BOR_LEVEL[1:0]	BOR阈值等级 00: level0上升/下降 (2.0/1.9V) 01: level1上升/下降 (2.4/2.3V) 10: level2上升/下降 (2.8/2.7V) 11: level3上升/下降 (3.2/3.1V)
8	BOR_EN	BOR使能控制 0: 禁止BOR 1: 使能BOR
7:1	保留	写入无效
0	RDPRP	RDP保护等级标志 0: RDP0 1: RDP1

### 5.9.7 Flash 选项字节寄存器 2 (FLASH\_OPTR2)

偏移地址：0x24

复位值：bit: 00000000 00000000 00000000 00X0000X

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.										IWDG_STOP	Res.				RST_STOP
										r					r

位/位域	名称	描述
31:6	保留	写入无效
5	IWDG_STOP	独立看门狗在Stop或Deepstop模式下计数器停止控制： 0：独立看门狗在Stop或Deepstop下计数器停止计数 1：独立看门狗在Stop或Deepstop下计数器正常运行
4:1	保留	写入无效
0	RST_STOP	低功耗模式复位 0：执行进入Stop或Deepstop模式操作，产生复位 1：执行进入Stop或Deepstop模式操作，进入Stop或Deepstop模式

### 5.9.8 Flash 写保护区寄存器 (FLASH\_WRP)

偏移地址：0x38

复位值：bit: 00000000 00000000 00000000 00XXXXXX

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.										WRP[5:0]					
										r	r	r	r	r	r

位/位域	名称	描述
------	----	----

---

31:6	保留	写入无效
5:0	WRP [5:0]	<p>WRP保护区域</p> <p>0: 写保护生效</p> <p>1: 写保护失效</p> <p>WRP每bit对应4K保护区域，详见 <a href="#">表: WRP 保护区域配置</a>。</p>



当  $V_{DD}$  电压达到 POR 上电复位阈值时，等待  $t_{RSTTEMPO}$  后释放复位，当  $V_{DD}$  电压低于 PDR 掉电复位阈值时，触发复位。

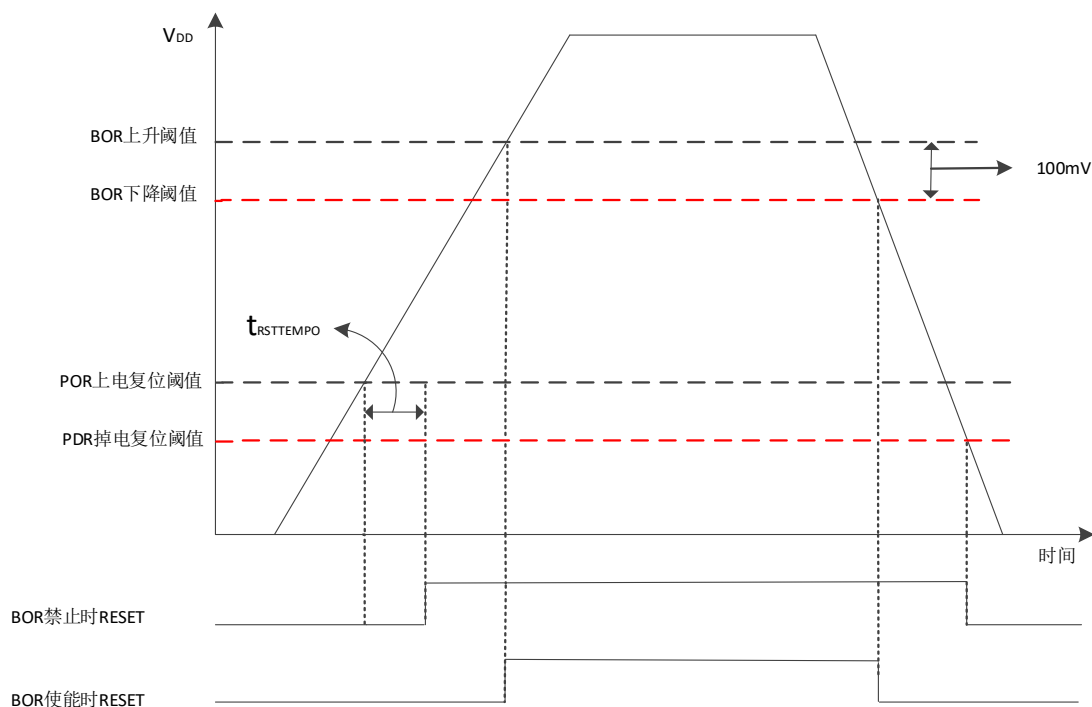
### 6.2.2 欠压复位

可通过选项字节配置 BOR 使能和 BOR 的阈值，BOR 阈值有 4 档可选；参见[选项字节寄存器 1 \(FLASH\\_OPTR1\)](#)。

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，释放复位；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 POR 上电复位阈值时，释放复位。

图 6-2 POR/PDR 和 BOR 阈值



### 6.2.3 内部带隙基准电压

内部带隙基准电压（BGR，典型电压 0.8V）有多种用途，具体如下：

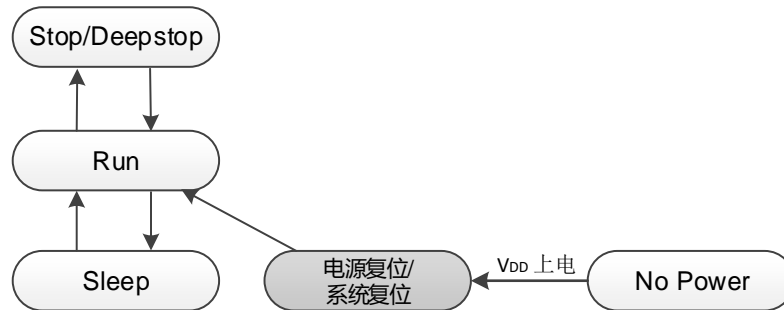
- 可作为 ADC 的内部输入通道；
- 可作为 COMP 的反相输入信号。

## 6.3 低功耗模式

复位后，芯片处于 Run 模式，系统时钟源为 RCHDIV6（8MHz）。芯片提供多种

低功耗模式以降低系统功耗；另外，还可以通过降低系统时钟频率和关闭未使用外设的时钟来降低 Run 模式下的系统功耗。

图 6-3 低功耗模式转换框图



芯片提供 3 种低功耗模式，不同功耗模式差异以及可用外设参见下表：

表 6-1 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	VR 模式
Sleep	仅关闭 CPU 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	MR
		SLEEPDEEP 清 0 WFE	唤醒事件		
Stop	V <sub>CORE</sub> 电源域（包括 CPU、数字外设）时钟关闭；RCH 和 EXTCLK 关闭，RCL 可正常工作；Flash 处于 Standby 状态；SRAM 数据保持；	SLEEPDEEP 置 1 LP_MODE 清 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	RCHDIV6 (8MHz)	MR
		SLEEPDEEP 置 1 LP_MODE 清 0 WFE	与 EXTI 信号线连接的外设生成的唤醒事件		
Deepstop	V <sub>CORE</sub> 电源域（包括 CPU、数字外设）时钟关闭；RCH 和 EXTCLK 关闭，RCL 可正常工作；Flash 处于 PowerDown 状态；SRAM 数据保持；	SLEEPDEEP 置 1 LP_MODE 置 1 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	RCHDIV6 (8MHz)	LPR
		SLEEPDEEP 置 1 LP_MODE 置 1 WFE	与 EXTI 信号线连接的外设生成的唤醒事件		

表 6-2 低功耗模式下可用外设<sup>(1)</sup>

外设	Run	Sleep	Stop/Deepstop	
			功能	唤醒源
CPU	√	×	×	×
Flash	√	√	×	×
SRAM	√	√	√	×
BOR	○	○	○	○
POR/PDR	√	√	√	√
RCH	○	○	×	×
RCL	○	○	○	×
EXTCLK	○	○	×	×
UART1/2	○	○	×	×
I2C1	○	○	×	×
SPI1	○	○	×	×
TIM1/3	○	○	×	×
LPTIM1	○	○	○	○
IRTIM	○	○	×	×
IWDG	○	○	○	○
SysTick	○	○	×	×
ADC	○	○	×	×
COMP1/2	○	○	○	○
GPIO	○	○	○	○
CRC	○	○	×	×

1. √-表示常开，×-表示不可用，○-可配置；

### 6.3.1 Run

芯片复位后进入 Run 模式，默认系统时钟源为 RCHDIV6，时钟频率为 8MHz，最高支持 48MHz，所有外设均可使用。

该模式下，为降低功耗可进行如下配置：

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



### 6.3.2 Sleep

进入 Sleep 模式仅关闭 CPU 的时钟，所有外设均可正常工作；退出 Sleep 模式后，芯片进入 Run 模式，系统时钟源维持不变。

表 6-3 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

### 6.3.3 Stop

Stop 基于 Cortex-M0+ 的深度休眠功能实现，V<sub>CORE</sub> 电源域（包括 CPU、数字外设）时钟关闭，RCH 和 EXTCLK 时钟关闭，RCL 根据配置可保持正常运行。

在 Stop 模式下，VR 处于 MR 模式，SRAM 和寄存器的数据保持，Flash 处于 Standby 状态。如果 Flash 处于编程或擦除过程中，会等待 Flash 操作完成后进入 Stop。

Stop 下可工作的外设参见 [表：低功耗模式下可用外设](#)。

ADC 在 Stop 模式下虽然无法工作，但仍会产生功耗；进入 Stop 前，应该关闭 ADC 可进一步降低功耗。

退出 Stop 模式后，芯片进入 Run 模式，此时系统时钟源为 RCHDIV6。

表 6-4 Stop 说明

行为类型	描述说明
进入 Stop	1. 清 0 电源管理控制寄存器 (PMU_CR) 中 LP_MODE;
	2. SLEEPDEEP 置 1;
	3. 执行 WFI;
	1. 清 0 电源管理控制寄存器 (PMU_CR) 中 LP_MODE;
	2. SLEEPDEEP 置 1;
	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. 清 0 电源管理控制寄存器 (PMU_CR) 中 LP_MODE;
	2. SLEEPDEEP 置 1;
	3. SLEEPONEXIT 置 1;
	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 <a href="#">EXTI 通道及信号对应表</a> ;
	通过 WFE (SEVONPEND 为 0) 方式进入低功耗模式:
	- 与 EXTI 信号线连接的外设中断服务请求 (NVIC 中断使能), 参见 <a href="#">EXTI 通道及信号对应表</a> ;
	- EXTI 生成的事件;
	通过 WFE (SEVONPEND 为 1) 方式进入低功耗模式:
	- 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 <a href="#">EXTI 通道及信号对应表</a> ;
	- EXTI 生成的事件;
退出延迟	RCH 唤醒时间

### 6.3.4 Deepstop

Deepstop 是在 Stop 模式的基础上将 VR 切换至 LPR 模式, 同时 Flash 进入到 Powndown 状态, 从而进一步降低功耗, 但是唤醒时间有所延长。

Deepstop 下可工作的外设与 Stop 模式无差异, 参见 [表: 低功耗模式下可用外设](#)。

表 6-5 Deepstop 说明

行为类型	描述说明
进入 Deepstop	1. 置 1 电源管理控制寄存器 (PMU_CR) 中 LP_MODE;
	2. SLEEPDEEP 置 1;
	3. 执行 WFI;
	1. 置 1 电源管理控制寄存器 (PMU_CR) 中 LP_MODE;
	2. SLEEPDEEP 置 1;

	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. 置 1 电源管理控制寄存器（PMU_CR）中 LP_MODE； 2. SLEEPDEEP 置 1； 3. SLEEPONEXIT 置 1； 4. 退出最低优先级中断服务函数；
退出 Deepstop	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 <a href="#">EXTI 通道及信号对应表</a> ；
	通过 WFE（SEVONPEND 为 0）方式进入低功耗模式： <ul style="list-style-type: none"> <li>与 EXTI 信号线连接的外设中断服务请求（NVIC 中断使能），参见 <a href="#">EXTI 通道及信号对应表</a>；</li> <li>EXTI 生成的事件；</li> </ul>
	通过 WFE（SEVONPEND 为 1）方式进入低功耗模式： <ul style="list-style-type: none"> <li>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 <a href="#">EXTI 通道及信号对应表</a>；</li> <li>EXTI 生成的事件；</li> </ul>
退出延迟	MR 唤醒时间+RCH 唤醒时间+Flash 唤醒时间

## 6.4 PMU 寄存器

PMU 寄存器支持 32 位访问。

表 6-6 PMU 基地址

外设	基地址
PMU	0x4000 7000

### 6.4.1 电源管理控制寄存器（PMU\_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															LP_MODE
															rw

位/位域	名称	描述
31:1	保留	写入无效
0	LP_MODE	低功耗模式选择 0: Stop 1: Deepstop

### 6.4.2 Flash 唤醒时间寄存器（PMU\_FLASH\_WAKEUP）

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.														FLASH_WAKEUP[1:0]	
														rw	rw

位/位域	名称	描述
------	----	----

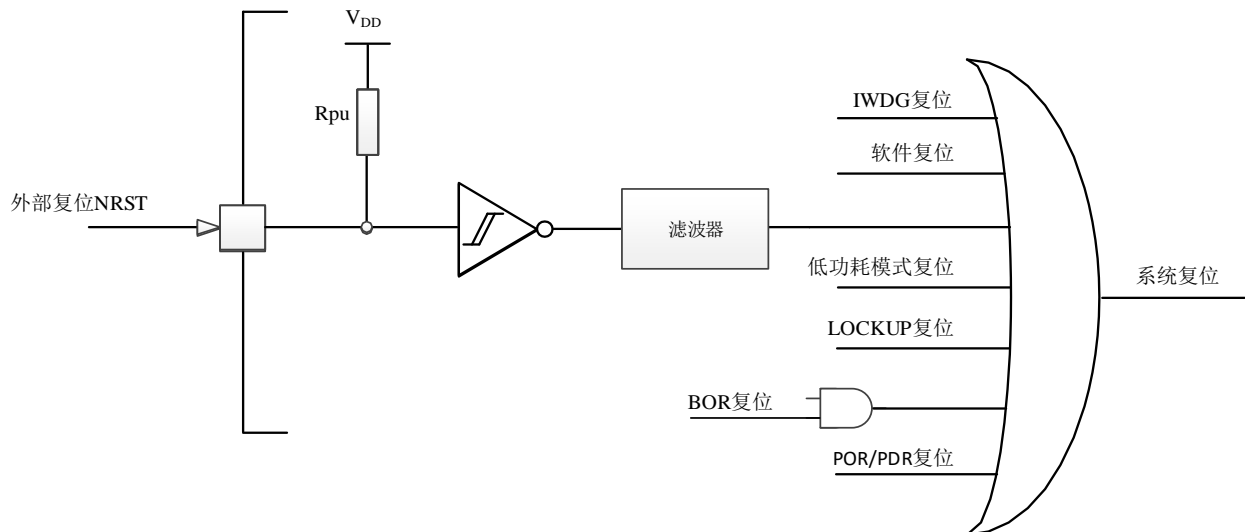
31:2	保留	写入无效
1:0	FLASH_WAKEUP[1:0]	<p>Deepstop唤醒过程中Flash的唤醒等待时间</p> <p>00: 10<math>\mu</math>s</p> <p>01: 8<math>\mu</math>s</p> <p>10: 6<math>\mu</math>s</p> <p>11: 0<math>\mu</math>s</p> <p><i>注意：该位域用于Deepstop唤醒过程中Flash的唤醒等待时间，当该寄存器设置为11时，表明唤醒流程不会进行Flash等待，唤醒后需要从SRAM执行程序且唤醒后10<math>\mu</math>s内不访问Flash。</i></p>

## 7 复位和时钟控制器（RCC）

### 7.1 复位

芯片支持两种类型的复位，分别为电源复位和系统复位。

图 7-1 复位电路框图



#### 7.1.1 电源复位

电源复位会复位所有寄存器，包含以下两种情况：

- 上下电复位（POR/PDR）；
- 欠压复位（BOR）。

#### 7.1.2 系统复位

系统复位将复位除 RCL 校准寄存器、RCH 校准寄存器、DBG 相关寄存器和 RCC\_CSR2 寄存器部分位之外的所有寄存器。

系统复位会重新加载 Flash 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；
- 独立看门狗事件（IWDG 复位）；
- 软件复位；
- 低功耗模式复位；

- LOCKUP 复位。

可通过查询寄存器 `RCC_CSR2` 中的复位标志确定复位源。

### NRST 引脚低电平复位

通过 *Flash 选项字节寄存器 1 (FLASH\_OPTR1)* 中的 `NRST_SWD_MODE[1:0]` 位域可配置 `PC0` 引脚为 NRST 或 GPIO 或 SWDIO 功能。

`PC0` 上电默认功能为 NRST；当电源/系统复位释放后，`PC0` 引脚随选项字节加载的功能才会更新生效。

当 `PC0` 配置为 NRST 功能时，引脚检测到低电平会产生一个系统复位。该引脚内置上拉电阻及毛刺过滤电路。为确保可靠的复位，NRST 引脚低电平保持时间应大于 500μs。详见图：复位电路框图。

### 软件复位

将 Cortex-M0+ 寄存器（`SCB_AIRCR` 寄存器）中的 `SYSRESETREQ` 位置 1，可产生软件复位。

### 低功耗模式复位

为了防止误入低功耗模式，系统提供了低功耗模式复位，如果在选项字节中使能该复位，那么在进入 Stop 或 Deepstop 模式时会产生系统复位。

可以将 *Flash 选项字节寄存器 2 (FLASH\_OPTR2)* 中的 `RST_STOP` 位清零来使能该功能。使能后，只要进入 Stop 或 Deepstop 模式，就会立即触发系统复位，而非进入 Stop 或 Deepstop 模式。

## 7.2 时钟

时钟单元提供了一系列频率的时钟源，包括：

- RCH：内部高速 48MHz RC 振荡器时钟；
- RCL：内部低速 32KHz RC 振荡器时钟；
- EXTCLK：外部输入时钟，支持最高 24MHz 时钟输入。

复位后，`RCHDIV6`（8MHz）作为系统时钟。

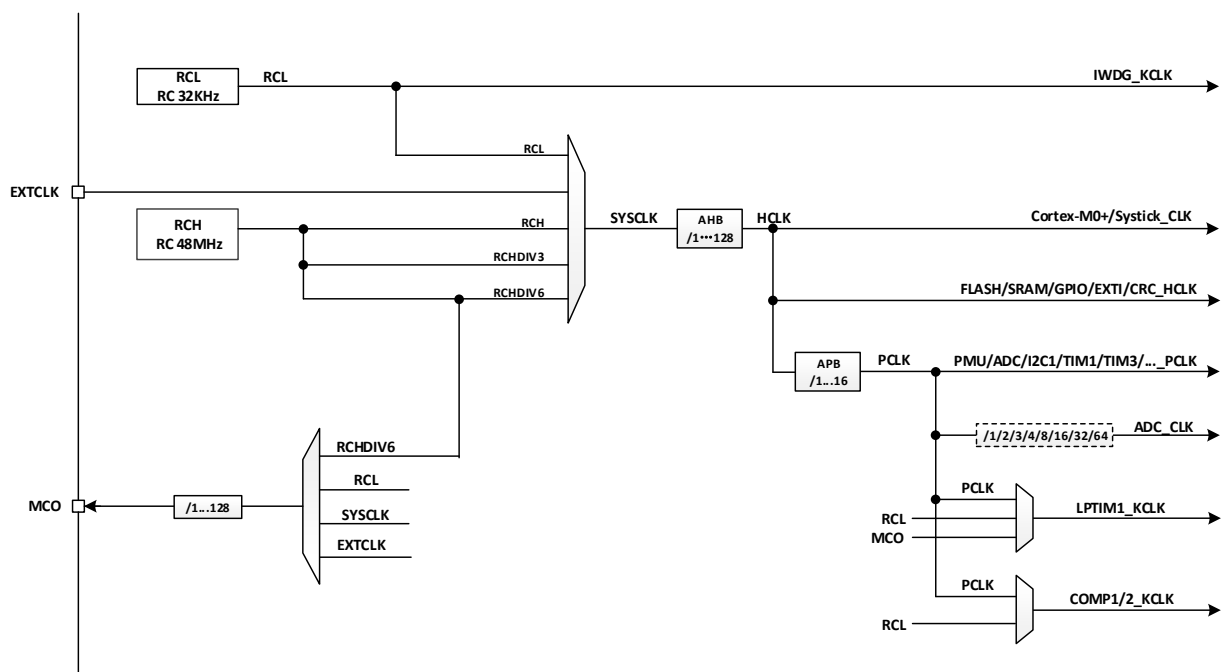
可通过多个预分频器配置 AHB 和 APB 时钟，最大频率为 48MHz。

下表为输入/输出时钟关系表：

表 7-1 时钟关系表

输出时钟	输入时钟	说明
RCHDIV3	RCH	RCH 的 3 分频
RCHDIV6	RCH	RCH 的 6 分频
SYSCLK	RCH、RCHDIV3、RCHDIV6、RCL、EXTCLK	系统时钟，通过寄存器选择时钟源
HCLK	SYSCLK	AHB 时钟，来自 SYSCLK 的 1 到 128 分频
PCLK	HCLK	APB 时钟，来自 HCLK 的 1 到 16 分频

图 7-2 时钟树



### 7.2.1 RCH 时钟

RCH 时钟是由内部的 48MHz RC 振荡器生成的，可用作系统时钟源。

RCH 从启动到稳定时间仅需 1 $\mu$ s(典型值)，全电压全温度范围内的精度为 $\pm 2\%$ ，免外接晶体谐振器。

RCH 可通过 RCC\_CSR1 寄存器中的 RCHON 位来使能或禁止，RCHRDY 标志指示 RCH 是否稳定。如在 RCC\_IER 寄存器中使能中断，则 RCH 稳定时便会产生中断。

RCH 支持用户校准，可通过 RCC\_RCHCAL 寄存器对其进行频率调整；校准可通过 LPTIM1 的级联触发计数模式实现，具体参见：[基于 LPTIM1 的时钟测量](#)。

RCH 出厂校准值保存在 0x1FFF0204 位置，具体格式参见表：[出厂校准参数](#)。



### 7.2.2 RCL 时钟

RCL 时钟频率为 32KHz，可在 Stop 或 Deepstop 模式下保持运行，为 IWDG、LPTIM1 和 COMP1/2 提供工作时钟。

RCL 使能的方式有两种：

- 通过 RCC\_CSR2 寄存器中的 RCLON 位来使能；
- 使能 IWDG 时，RCL 强制使能。

只有 RCLON 寄存器为 0 且禁止 IWDG 时，该时钟才会关闭。

RCC\_CSR2 寄存器中的 RCLRDY 标志指示 RCL 是否稳定。如在 RCC\_IER 寄存器中使能中断，则 RCL 稳定时便会产生中断。

RCL 支持用户校准，可通过 RCC\_RCLCAL 寄存器中 RCL\_CAL[3:0]位域对其进行细调；校准可通过 LPTIM1 的级联触发计数模式实现，具体参见：[基于 LPTIM1 的时钟测量](#)。

RCL 出厂校准值保存在 0x1FFF020C 位置，具体格式参见[表：出厂校准参数](#)。

### 7.2.3 EXTCLK 时钟

EXTCLK 信号仅支持外部时钟输入，时钟频率最大不超过 24MHz。

EXTCLK 可通过 RCC\_CSR1 寄存器中的 EXTCLKON 位来使能或禁止，该时钟由外部直接输入，无需额外的稳定等待时间。

### 7.2.4 系统时钟

可以使用下列不同的时钟源来驱动系统时钟（SYSCLK）：

- RCH
- RCHDIV3
- RCHDIV6
- RCL
- EXTCLK

系统时钟最高频率为 48 MHz。系统复位或 Stop 或 Deepstop 低功耗模式唤醒后，选择 RCHDIV6（8MHz）作为系统时钟。

当时钟切换时，如果选择尚未开启或稳定的时钟源，则切换在该时钟源开启及稳定后才会进行。RCC\_CFG 寄存器中的 SYSWS 用于指示当前的系统时钟源，

作为系统时钟的时钟源无法被关闭。

### 7.2.5 外设异步时钟选择

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

表 7-2 外设异步时钟来源

模块	异步时钟来源
IWDG	RCL
COMP1/2	PCLK 或 RCL
LPTIM1	PCLK 或 RCL 或 MCO

### 7.2.6 IWDG 时钟

IWDG 使能时 RCL 将被强制开启，且不能被关闭。当 RCL 使能并稳定后，将为 IWDG 提供工作时钟。

### 7.2.7 时钟输出功能

可选择以下时钟之一通过 MCO 引脚输出到外部：

- RCHDIV6
- RCL
- SYSCLK
- EXTCLK

由 RCC\_CFG 寄存器中的 MCO\_SEL[2:0]位域选择输出的时钟，所选时钟可以通过 MCO\_PRE[2:0]位域进行分频。

### 7.2.8 基于 LPTIM1 的时钟测量

RCH 和 RCL 的频率都可通过 LPTIM1 的级联触发计数模式进行测量。RCH 可通过 MCO 作为 LPTIM1 的计数时钟源，而 RCL 可直接作为 LPTIM1 的计数时钟源。通过配置 [外设异步时钟配置寄存器（RCC\\_CLKSEL）](#) 来选择 LPTIM1 的计数时钟源。

时钟测量是利用精准的时钟源作为 TIM3 计数时钟，待测时钟 RCH 或 RCL 作为 LPTIM1 的计数时钟源，通过级联触发计数模式启动或暂停计数，从而实现对待测时钟的频率测量，详见：[级联触发计数模式](#)。

为了减少测量误差，TIM3 的计数时间应当尽可能长或者 LPTIM1 的计数值尽可

能大，但是需要注意 TIM3、LPTIM1 的计数值均不能溢出。

### 7.3 低功耗模式

- 可通过软件禁止 AHB 和 APB 未使用的外设时钟；
- Sleep 模式下，仅关闭 CPU 时钟，其他时钟与 Run 模式一致；
- Stop 或 Deepstop 模式下，将关闭 RCH 和 EXTCLK，RCL 时钟不受影响；

如果进入 Stop 或 Deepstop 模式前将[控制寄存器 \(DBG\\_CR\)](#)中的 DBG\_STOP 位置 1, 那么在 Stop 或 Deepstop 模式下 RCHDIV6 将被开启并为系统提供时钟，以实现 Stop 或 Deepstop 模式的调试功能。

## 7.4 RCC 寄存器

RCC 寄存器支持 32 位访问。

表 7-3 RCC 基地址列表

外设	基地址
RCC	0x4002 1000

### 7.4.1 时钟控制/状态寄存器 (RCC\_CSR1)

偏移地址: 0x00

复位值: 0x0000 0500

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

位/位域	名称	描述
31:17	保留	写入无效
16	EXTCLKON	EXTCLK时钟使能 在进入Stop或Deepstop模式时由硬件将该位清0。 如果EXTCLK直接用作系统时钟，则该位不可清0。 0: 禁止 1: 使能
15:11	保留	写入无效
10	RCHRDY	RCH时钟稳定标志 当配置RCHON为0时，RCH_RDY将自动清0。 0: RCH时钟未稳定 1: RCH时钟已稳定
9	保留	写入无效

8                      RCHON                      RCH时钟使能  
 在进入Stop或Deepstop模式时由硬件将该位清0。  
 0: 禁止  
 1: 使能

7:0                      保留                      写入无效

## 7.4.2 时钟配置寄存器 (RCC\_CFG)

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	MCOPRE[2:0]			Res	MCOSEL[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.	PPRE[2:0]			Res.	HPRE[2:0]			Res.		SYSWS[2:0]			SYSW[2:0]		
	rw	rw	rw		rw	rw	rw			r	r	r	rw	rw	rw

位/位域	名称	描述
31	保留	写入无效
30:28	MCOPRE[2:0]	时钟输出预分频 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	写入无效
26:24	MCOSEL[2:0]	时钟输出源选择 000: 禁止MCO输出 001: 选择SYSCLK 011: 选择RCHDIV6

		100: 选择EXTCLK
		110: 选择RCL
		其他: 保留 (禁止MCO输出)
		<i>注意: 使能或切换MCO时钟时, 时钟输出会存在不完整的周期。</i>
23:15	保留	写入无效
14:12	PPRE[2:0]	APB预分频器 用于选择APB时钟 (PCLK) 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
11	保留	写入无效
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
7:6	保留	写入无效
5:3	SYSWS[2:0]	系统时钟源指示 000: RCHDIV6 001: RCHDIV3 010: RCH 011: RCL 100: EXTCLK
2:0	SYSW[2:0]	系统时钟源选择

- 000: RCHDIV6用作系统时钟
- 001: RCHDIV3用作系统时钟
- 010: RCH用作系统时钟
- 011: RCL用作系统时钟
- 100: EXTCLK用作系统时钟
- 其他: 保留 (RCHDIV6用作系统时钟)

### 7.4.3 时钟中断使能寄存器 (RCC\_IER)

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.												RCH_RDYIE	Res.		RCL_RDYIE
												rw			rw

位/位域	名称	描述
31:4	保留	写入无效
3	RCH_RDYIE	RCH稳定中断使能 0: 禁止 1: 使能
2:1	保留	写入无效
0	RCL_RDYIE	RCL稳定中断使能 0: 禁止 1: 使能

### 7.4.4 时钟中断标志寄存器 (RCC\_ISR)

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.												RCH_RDYF	Res.		RCL_RDYF
												r			r

位/位域	名称	描述
31:4	保留	写入无效
3	RCH_RDYF	RCH时钟稳定中断标志 当RCH时钟稳定，且RCH_RDYIE置1时，该位被置1。 将RCH_RDYC位置1，可清除该位。 0：未触发RCH稳定中断 1：触发RCH稳定中断
2:1	保留	写入无效
0	RCL_RDYF	RCL时钟稳定中断标志 当RCL时钟稳定，且RCL_RDYIE置1时，该位被置1。 将RCL_RDYC位置1，可清除该位。 0：未触发RCL稳定中断 1：触发RCL稳定中断

#### 7.4.5 时钟中断清除寄存器（RCC\_ICR）

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.												RCH_RDYC	Res.		RCL_RDYC
												w			w

位/位域	名称	描述
31:4	保留	写入无效
3	RCH_RDYC	RCH时钟稳定中断标志清零



- 0: 无影响  
1: 将RCH\_RDYF标志清零

2:1          保留          写入无效

0          RCL\_RDYC          RCL时钟稳定中断标志清零  
0: 无影响  
1: 将RCL\_RDYF标志清零

#### 7.4.6      **GPIO 端口复位寄存器 (RCC\_IOPRST)**

偏移地址: 0x1C

复位值: 0x0000 0000

**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 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.													GPIOC_ RST	GPIOB_ RST	GPIOA_ RST
													rw	rw	rw

位/位域	名称	描述
31:3	保留	写入无效
2	GPIOC_RST	GPIOC复位 0: 正常工作模式 1: 复位GPIOC
1	GPIOB_RST	GPIOB复位 0: 正常工作模式 1: 复位GPIOB
0	GPIOA_RST	GPIOA复位 0: 正常工作模式 1: 复位GPIOA

### 7.4.7 AHB 外设复位寄存器 (RCC\_AHBRST)

偏移地址：0x20

复位值：0x0000 0000

**注意：** 所有外设复位软件流程为：相应 bit 位置 1，复位外设；清除该位，使外设处于正常工作模式。

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.			CRC_RST	Res.											
			rw												

位/位域	名称	描述
31:13	保留	写入无效
12	CRC_RST	CRC复位 0：正常工作模式 1：复位CRC
11:0	保留	写入无效

### 7.4.8 APB 外设复位寄存器 1 (RCC\_APB1RST1)

偏移地址：0x24

复位值：0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1_RST	Res.										I2C1_RST	Res.			UART2_RST
rw											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TIM3_RST	Res.
														rw	

位/位域	名称	描述
31	LPTIM1_RST	LPTIM1复位 0: 正常工作模式 1: 复位LPTIM1
30:22	保留	写入无效
21	I2C1_RST	I2C1复位 0: 正常工作模式 1: 复位I2C1
20:18	保留	写入无效
17	UART2_RST	UART2复位 0: 正常工作模式 1: 复位UART2
16:2	保留	写入无效
1	TIM3_RST	TIM3复位 0: 正常工作模式 1: 复位TIM3
0	保留	写入无效

#### 7.4.9 APB 外设复位寄存器 2 (RCC\_APBRS2)

偏移地址: 0x28

复位值: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DBG_RST	Res.						ADC_RST	Res.			
				rw							rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UART1_RST	Res.	SPI1_RST	TIM1_RST	Res.										COMP_RST
	rw		rw	rw											rw

位/位域	名称	描述
31:28	保留	写入无效
27	DBG_RST	DBG复位 0: 正常工作模式 1: 复位DBG
26:21	保留	写入无效
20	ADC_RST	ADC复位 0: 正常工作模式 1: 复位ADC
19:15	保留	写入无效
14	UART1_RST	UART1复位 0: 正常工作模式 1: 复位UART1
13	保留	写入无效
12	SPI1_RST	SPI1复位 0: 正常工作模式 1: 复位SPI1
11	TIM1_RST	TIM1复位 0: 正常工作模式 1: 复位TIM1
10:1	保留	写入无效
0	COMP_RST	COMP复位 0: 正常工作模式 1: 复位 COMP1 和 COMP2

### 7.4.10 I/O 端口时钟使能寄存器 (RCC\_IOPEN)

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.													GPIOC EN	GPIOB EN	GPIOA EN
													rw	rw	rw

位/位域	名称	描述
31:3	保留	写入无效
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

### 7.4.11 AHB 外设时钟使能寄存器 (RCC\_AHBEN)

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.			CRC EN	Res.											
			rw												

位/位域	名称	描述
31:13	保留	写入无效
12	CRCEN	CRC时钟使能 0: 禁止 1: 使能
11:0	保留	写入无效

#### 7.4.12 APB 外设时钟使能寄存器 1 (RCC\_APBEN1)

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
LPTIM1 EN	Res.		PMU EN	Res.						I2C1 EN	Res.			UART2 EN	Res.
rw			rw							rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TIM3 EN	Res.
														rw	

位/位域	名称	描述
31	LPTIM1EN	LPTIM1时钟使能 0: 禁止 1: 使能
30:29	保留	写入无效
28	PMUEN	PMU时钟使能 0: 禁止 1: 使能
27:22	保留	写入无效
21	I2C1EN	I2C1时钟使能 0: 禁止 1: 使能

20:18	保留	写入无效
17	UART2EN	UART2时钟使能 0: 禁止 1: 使能
16:2	保留	写入无效
1	TIM3EN	TIM3时钟使能 0: 禁止 1: 使能
0	保留	写入无效

#### 7.4.13 APB 外设时钟使能寄存器 2 (RCC\_APBEN2)

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.				DBG EN	Res.						ADC EN	Res.			
				rw							rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UART1 EN	Res.	SPI1 EN	TIM1 EN	Res.									COMP EN	
	rw		rw	rw											rw

位/位域	名称	描述
31:28	保留	写入无效
27	DBGEN	DBG时钟使能 0: 禁止 1: 使能
26:21	保留	写入无效
20	ADCEN	ADC时钟使能 0: 禁止

		1: 使能
19:15	保留	写入无效
14	UART1EN	UART1时钟使能 0: 禁止 1: 使能
13	保留	写入无效
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	写入无效
0	COMPEN	COMP1和COMP2时钟使能 0: 禁止 1: 使能

#### 7.4.14 外设异步时钟配置寄存器 (RCC\_CLKSEL)

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.												LPTIM1_SEL [1:0]		Res	
												rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													COMP2_SEL	COMP1_SEL	
													rw	rw	

位/位域	名称	描述
31:20	保留	写入无效



19:18	LPTIM1_SEL[1:0]	LPTIM1时钟源选择 00: PCLK 01: RCL 1x: MCO
17:2	保留	写入无效
1	COMP2_SEL	COMP2滤波时钟源选择 0: PCLK 1: RCL
0	COMP1_SEL	COMP1滤波时钟源选择 0: PCLK 1: RCL

#### 7.4.15 时钟控制/状态寄存器 (RCC\_CSR2)

偏移地址: 0x44

复位值: 0xXX00 0000

**注意:** 电源复位可复位该寄存器; 系统复位仅可复位 *bit0*、*bit1*、*bit8* 和 *bit16*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPM_RSTF	Res.	IWDG_RSTF	SW_RSTF	PMU_RSTF	NRST_RSTF	Res.	LOCKUP_RSTF	Res.							RMVF
r		r	r	r	r		r								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							LOCKUP_RSTEN	Res.						RCL_RDY	RCL_ON
							rw							r	rw

位/位域	名称	描述
31	LPM_RSTF	低功耗复位标志 低功耗复位时, 该位被置1。将RMVF位置1可清除该位。 0: 未产生低功耗模式复位 1: 产生非法模式复位
30	保留	写入无效
29	IWDG_RSTF	独立看门狗复位标志

		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	SW_RSTF	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	PMU_RSTF	POR/PDR、BOR复位标志
		POR/PDR或BOR复位时，该位被置1。将RMVF位置1可清除该位。
		0: 未产生POR/PDR或BOR复位
		1: 产生POR/PDR或BOR复位
26	NRST_RSTF	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: 产生来自NRST引脚的复位
25	保留	写入无效
24	LOCKUP_RSTF	LOCKUP复位标志
		当LOCKUP功能使能（LOCKUP_RSTEN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
23:17	保留	写入无效
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。
		0: 无影响
		1: 清除复位标志
15:9	保留	写入无效
8	LOCKUP_RSTEN	LOCKUP复位使能

		0: 禁止 1: 使能
7:2	保留	写入无效
1	RCLRDY	RCL稳定标志 当RCLON位被清0后，RCLRDY将自动清0。 0: RCL未稳定 1: RCL已稳定
0	RCLON	RCL使能 0: 禁止 1: 使能

#### 7.4.16 RCL 校准寄存器 (RCC\_RCLCAL)

偏移地址: 0x50

复位值: 0x0000 000X

**注意:** 仅电源复位时，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.												RCL_CAL[3:0]			
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	写入无效

3:0	RCL_CAL[3:0]	RCL 时钟校准值
-----	--------------	-----------

#### 7.4.17 RCH 校准寄存器 (RCC\_RCHCAL)

偏移地址: 0x54

复位值: 0x0000 0XXX

**注意:** 仅电源复位时，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.					RCH_CAL_COARSE[2:0]			Res.		RCH_CAL_FINE[5:0]					
					rw	rw	rw			rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:11	保留	写入无效
10:8	RCH_CAL_COARSE[2:0]	RCH 时钟粗调值 每增减 1 档，RCH 的输出频率同步增减约 5MHz。
7:6	保留	写入无效
5:0	RCH_CAL_FINE[5:0]	RCH 时钟微调值

## 8 通用输入/输出接口（GPIO）

### 8.1 简介

每组通用 I/O 端口包括 3 个 32 位配置寄存器（GPIOx\_MODE、GPIOx\_OTYPE、GPIOx\_PUPD），2 个 32 位数据寄存器（GPIOx\_IDR、GPIOx\_ODR），1 个 32 位置位/复位寄存器（GPIOx\_BSR）、1 个 32 位复位寄存器（GPIOx\_BR）；每个 GPIO 包括 1 个 32 位复用功能选择寄存器（GPIOx\_AFL）。

### 8.2 GPIO 主要特性

- GPIO 模式
  - 输入
  - 输出
  - 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 或外设复用功能
- 置位和复位寄存器（GPIOx\_BSR），对 GPIOx\_ODR 具有按位写权限

### 8.3 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

下图描述了 I/O 电路的基本结构,GPIO 端口配置表给出了可能的 I/O 配置方案。

图 8-1 I/O 基本结构

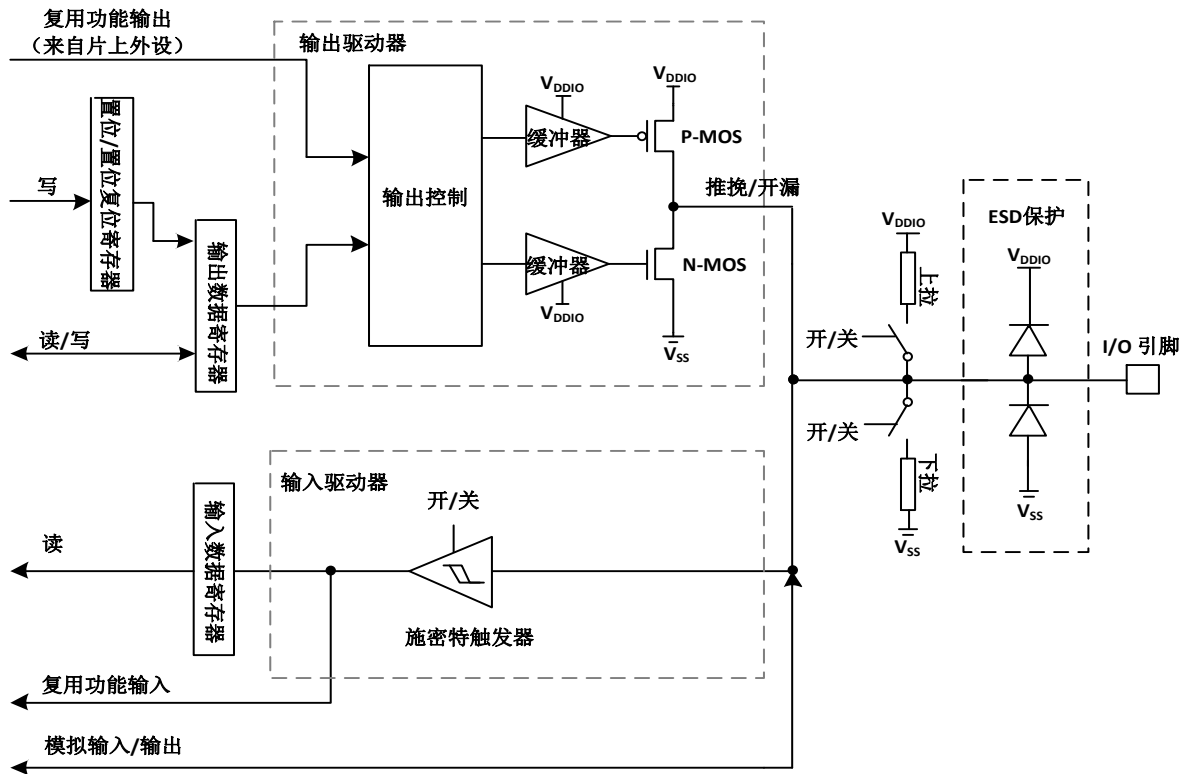


表 8-1 GPIO 端口配置表<sup>(1)</sup>

I/O 端模式组合		I/O 端口配置			
		GPIO 模式 MODE[1:0]	输出类型 OTYPE	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
复用	推挽	10	0	0	0

I/O 端模式组合		I/O 端口配置			
		GPIO 模式 MODE[1:0]	输出类型 OTYPE	上拉/下拉 PUPD[1:0]	
	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
模拟	输入/输出	11	-	上拉/下拉禁用	
			-		
			-		

1. “-”配置无效。

### 8.3.1 通用 I/O (GPIO)

复位释放前/后，除 PC0、PB6、PA2 引脚外，其他引脚均处于模拟模式。

#### ● PC0 引脚

PC0 上电默认功能为 NRST，当复位释放后，根据选项字节配置加载生效：

- FLASH\_OPTR1 寄存器配置 PC0 为 SWD 时，默认复用为 SWDIO、上拉，可配置为 GPIO 或其他复用功能；
- FLASH\_OPTR1 寄存器配置 PC0 为 NRST 时，默认复用为 NRST、上拉；
- FLASH\_OPTR1 寄存器配置 PC0 为 GPIO 时，默认处于模拟模式。

#### ● PA2 引脚

- 复位释放前/后，复用为 SWCLK、下拉。

#### ● PB6 引脚

- FLASH\_OPTR1 寄存器配置 PB6 为 SWD 时，复位释放前/后，复用为 SWDIO、上拉；
- FLASH\_OPTR1 寄存器配置 PB6 为 GPIO 时，复位释放前/后，处于模拟模式。

### 8.3.2 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。

## GPIO

在 I/O 模式寄存器 GPIOx\_MODE 中，将 I/O 配置为输入或输出。

### 复用功能

每个 I/O 电路结构中包含一个引脚复用器，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。

- 芯片复位释放后，复用功能寄存器 GPIOx\_AFL 的值默认选择复用功能 0；
- 复用功能配置。
  - 在 GPIOx\_AFL 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：端口复用功能映射](#)。
  - 通过 GPIOx\_OTYPE、GPIOx\_PUPD 寄存器，分别选择输出类型、上拉/下拉；
  - 在 GPIOx\_MODE 寄存器中将所需 I/O 配置为复用功能。

### 附加功能

附加功能由与之对应的外设进行配置，配置外设附加功能后，GPIO 寄存器中的对应配置无效。

- 模拟功能
  - 对于 ADC、COMP，在 GPIOx\_MODE 寄存器中将所需 I/O 配置为模拟模式，并在 ADC、COMP 寄存器中使能。
- 其他功能
  - 外部时钟输入：使能外部时钟输入（RCC\_CSR1 寄存器的 EXTCLKON 位），由 PC1 输入外部时钟。

### 8.3.3 I/O 端口配置寄存器

每个 I/O 端口包括 3 个 32 位配置寄存器（GPIOx\_MODE、GPIOx\_OTYPE、GPIOx\_PUPD）用于配置端口的每个 GPIO。

- GPIO 模式寄存器 GPIOx\_MODE，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\_OTYPE，用于配置输出类型推挽/开漏；
- 上拉/下拉寄存器 GPIOx\_PUPD 用于配置上拉/下拉。

#### 8.3.4 I/O 端口数据寄存器

每个 I/O 端口有 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，可进行读/写访问；
- GPIOx\_IDR 为只读寄存器，I/O 引脚输入的数据存储到 GPIOx\_IDR。

#### 8.3.5 I/O 数据位操作

置位/复位寄存器 GPIOx\_BSR 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[7:0]和 BR[7:0]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。

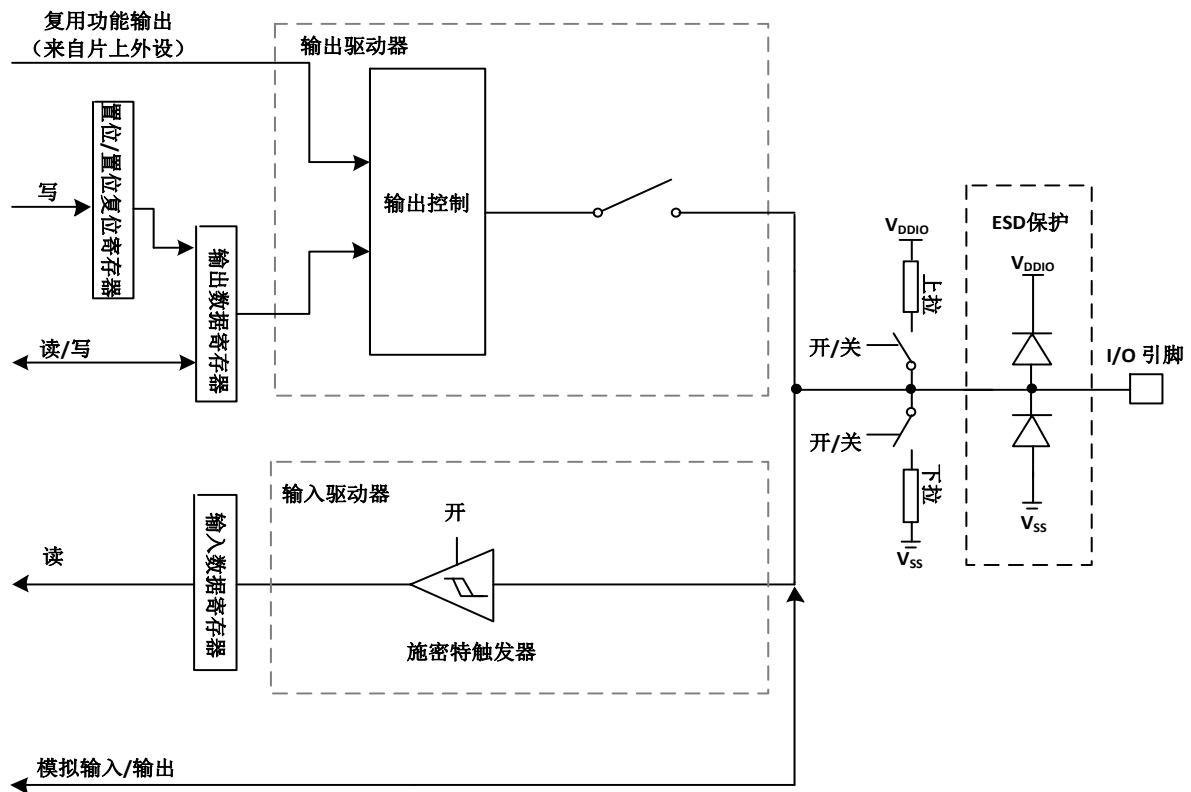
通过写访问 GPIOx\_BSR 寄存器修改 1 个或多个 GPIOx\_ODR 寄存器中的对应位，无需读 GPIOx\_ODR 寄存器后，再次写入，因此修改寄存器值的过程，不会被读取和写入之间产生的中断打断，无需软件关闭中断。

#### 8.3.6 外部中断/唤醒

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 扩展中断和事件控制器](#)。系统应用存在多个 IO 中断的情况下，应选择在不同 EXTI 通道上的 IO，参见 EXTI I/O 选择寄存器。如：PA0、PB0、PC0 在同一个 EXTI 通道上，PA1、PB1、PC1 在同一个 EXTI 通道上。

## 8.3.7 输入配置

图 8-2 I/O 引脚输入配置（浮空/上拉/下拉）

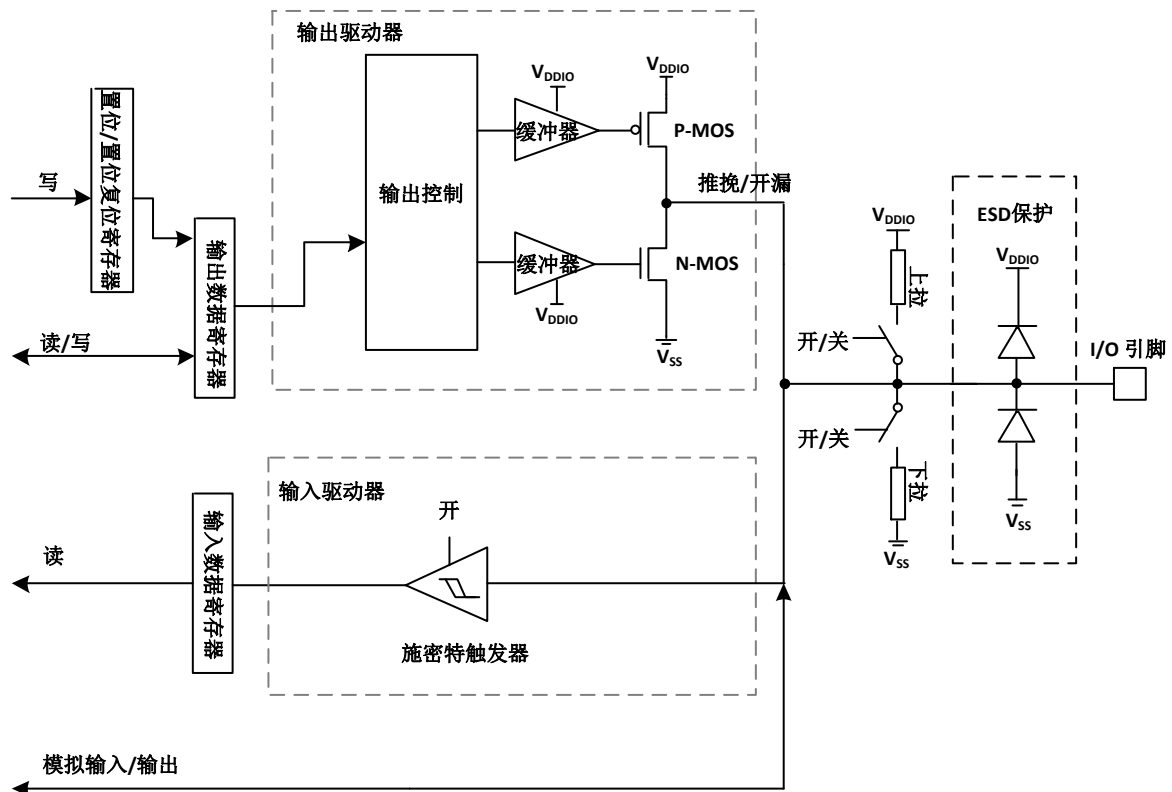


配置 I/O 为输入时，I/O 电路结构示意图如上图所示：

- 施密特触发器输入使能；
- 上拉、下拉电阻的选择由 GPIOx\_PUPD 寄存器中的值决定；
- 输出缓冲器禁用；
- 读访问输入数据寄存器获取 I/O 状态。

## 8.3.8 输出配置

图 8-3 I/O 引脚输出配置

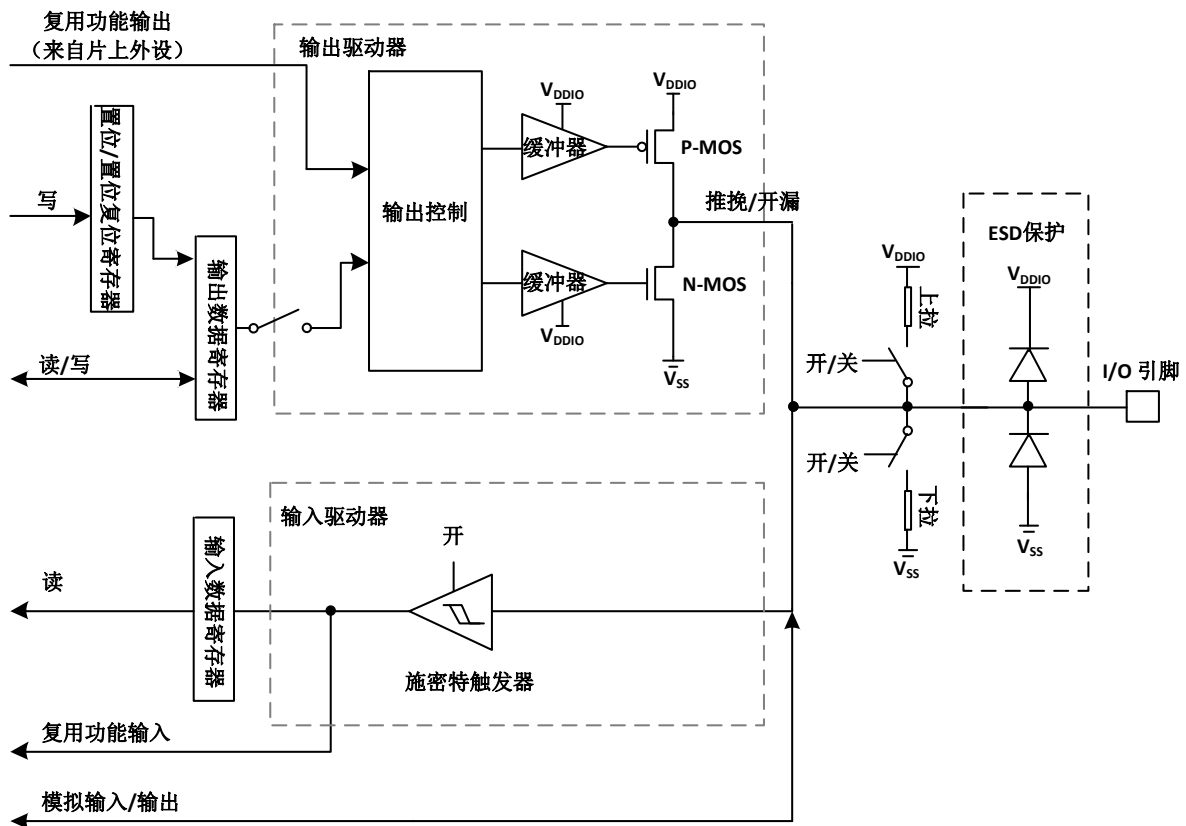


配置 I/O 为输出时，I/O 电路结构示意图如上图所示：

- 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\_PUPD 寄存器的值决定；
- 读访问输入数据寄存器获取 I/O 状态。

## 8.3.9 复用功能配置

图 8-4 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：

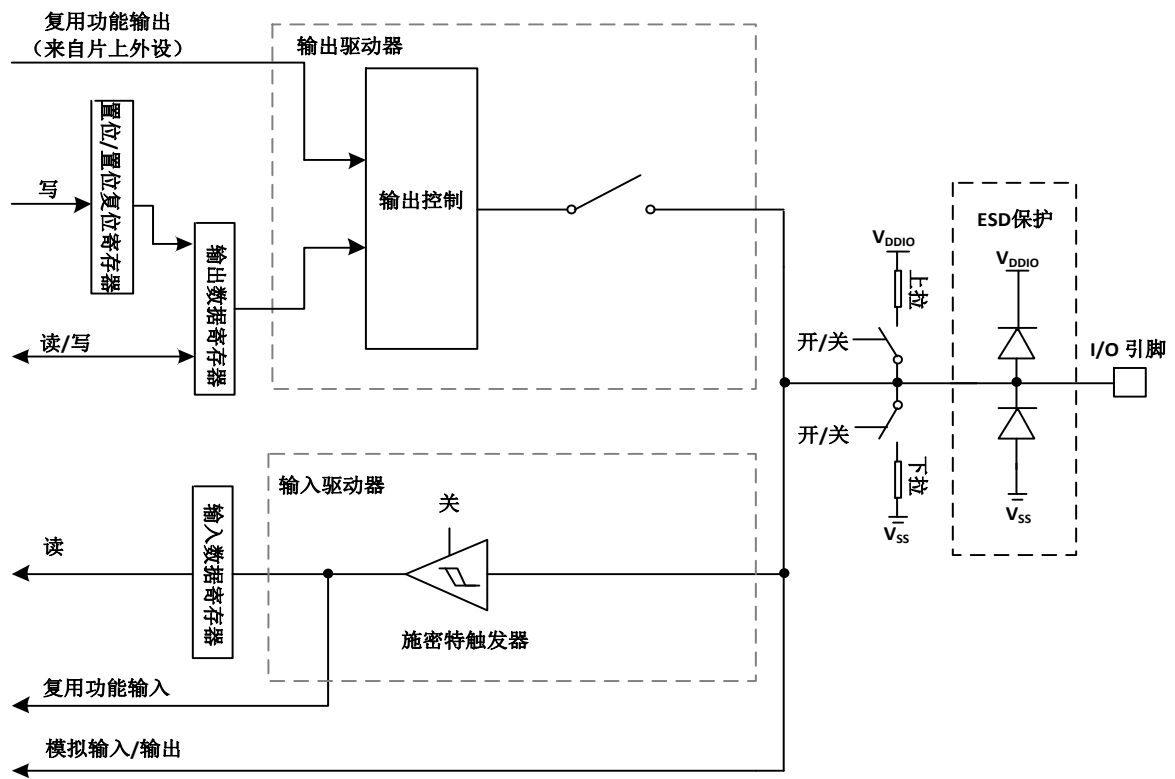
- 施密特触发器输入使能；
- 上拉、下拉电阻的选择由 GPIOx\_PUPD 寄存器中的值决定；
- 读访问输入数据寄存器获取 I/O 状态；
- 输出缓冲器使能。
  - 可配置为开漏或推挽模式；
  - 输出缓冲器由外设信号驱动。（发送器使能信号和输出数据）

## 8.3.10 复用功能选择

用户可通过复用功能寄存器 GPIOx\_AFL 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。外设复用功能参见表：端口复用功能映射。

## 8.3.11 模拟模式配置

图 8-5 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。

## 8.4 GPIO 寄存器

GPIO 寄存器支持 32 位访问。

表 8-2 GPIO 基地址列表

外设	基地址
GPIOA	0x5000 0000
GPIOB	0x5000 0400
GPIOC	0x5000 0800

### 8.4.1 GPIO 端口模式寄存器 (GPIO<sub>x</sub>\_MODE) (x = A~C)

偏移地址: 0x00

复位值: 端口 A      0x0000 FFEF

端口 B      0x0000 EFFF (FLASH\_OPTR1 配置 PB6 为 SWDIO)  
                  0x0000 FFFF (FLASH\_OPTR1 未配置 PB6 为 SWDIO)

端口 C      0x0000 000E (FLASH\_OPTR1 配置 PC0 为 SWDIO)  
                  0x0000 000F (FLASH\_OPTR1 未配置 PC0 为 SWDIO)

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
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	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	写入无效
15:0	MODE <sub>y</sub> [1:0]:	Port <sub>x</sub> 端口I/O引脚 <sub>y</sub> 的I/O模式配置。(x=A~C; y=0~7) 00: 输入模式 01: 输出模式 10: 复用功能模式 11: 模拟模式

### 8.4.2 GPIO 端口输出类型寄存器 (GPIO<sub>x</sub>\_OTYPE) (x = A~C)

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.								OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	写入无效
7:0	OTy[7:0]	Portx端口I/O引脚y的输出类型配置。(x=A~C; y=0~7) 0: 推挽输出 1: 开漏输出

#### 8.4.3 GPIO 端口上拉/下拉寄存器 (GPIOx\_PUPD) (x = A~C)

偏移地址：0x0C

复位值：端口 A	0x0000 0020
端口 B	0x0000 1000 (FLASH_OPTR1 配置 PB6 为 SWDIO) 0x0000 0000 (FLASH_OPTR1 未配置 PB6 为 SWDIO)
端口 C	0x0000 0001 (FLASH_OPTR1 配置 PC0 为 SWDIO) 0x0000 0000 (FLASH_OPTR1 未配置 PC0 为 SWDIO)

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
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

位/位域	名称	描述
31:16	保留	写入无效
15:0	PUPDy [1:0]	Portx端口I/O引脚y的上/下拉配置。(x=A~C; y=0~7) 00: 无上拉或下拉 01: 上拉

10: 下拉

11: 保留（写入无效，实际生效为前一次配置有效值）

#### 8.4.4 GPIO 端口输入寄存器（GPIOx\_IDR）（x = A~C）

偏移地址：0x10

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.								ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
								r	r	r	r	r	r	r	r

位/位域	名称	描述
31:8	保留	写入无效
7:0	IDy[7:0]	Portx端口I/O引脚y的输入数据。（x=A~C； y = 0 ~7） 这些位为只读，包含相应 I/O 端口的输入值。

#### 8.4.5 GPIO 端口输出数据寄存器（GPIOx\_ODR）（x = A~C）

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.								OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	写入无效
7:0	ODy	Portx端口I/O引脚y的输出数据。（x=A~C； y = 0 ~7） 这些位可通过软件读取和写入。



#### 8.4.6 GPIO 端口置位/复位寄存器 (GPIO<sub>x</sub>\_BSR) (x = A~C)

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.								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:24	保留	写入无效
23:16	BR <sub>y</sub>	Port <sub>x</sub> 端口I/O引脚 <sub>y</sub> 的复位配置。(x=A~C; y=0~7) 0: 写入无效 1: 复位 <i>注意: 如果同时对BS<sub>y</sub>和BR<sub>y</sub>置位, 则BS<sub>y</sub>的优先级更高。</i>
15:8	保留	写入无效
7:0	BS <sub>y</sub>	Port <sub>x</sub> 端口I/O引脚 <sub>y</sub> 的置位配置。(x=A~C; y=0~7) 0: 写入无效 1: 置位

#### 8.4.7 GPIO 复用功能低位寄存器 (GPIO<sub>x</sub>\_AFL) (x = A~C)

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.	AFSEL7[2:0]			Res.	AFSEL6[2:0]			Res.	AFSEL5[2:0]			Res.	AFSEL4[2:0]		
	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.	AFSEL3[2:0]			Res.	AFSEL2[2:0]			Res.	AFSEL1[2:0]			Res.	AFSEL0[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位/位域	名称	描述
------	----	----

31:0	AFSELY [2:0]	Portx端口I/O引脚y的复用功能选择。(x=A~C; y = 0 ~7)
		000: AF0
		001: AF1
		010: AF2
		011: AF3
		100: AF4
		101: AF5
		110: AF6
		111: AF7

#### 8.4.8 GPIO 端口位复位寄存器 (GPIOx\_BR) (x = A~C)

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.								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	写入无效
7:0	BRy	Portx端口I/O引脚位复位。(x=A~C; y = 0 ~7)
		0: 写入无效
		1: 复位

## 9 嵌套向量中断控制器（NVIC）

### 9.1 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。

### 9.2 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
- 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，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。

### 9.3 中断向量表

表 9-1 中断向量表<sup>(1)</sup>

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	保留	0x0000_0008
-	-1	固定	HardFault_Handler	所有类型的错误	0x0000_000C
-	-	-	保留	保留	0x0000_0010~ 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	-	保留	保留	0x0000_0040
1	8	-	保留	保留	0x0000_0044
2	9	-	保留	保留	0x0000_0048

位置	优先级	优先级类型	缩略语	说明	地址
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[7:4]	EXTI 通道 4 到 7 中断	0x0000_005C
8	15	-	保留	保留	0x0000_0060
9	16	-	保留	保留	0x0000_0064
10	17	-	保留	保留	0x0000_0068
11	18	-	保留	保留	0x0000_006C
12	19	可设置	ADC/COMP	ADC 中断 COMP 中断 (EXTI 16-17)	0x0000_0070
13	20	可设置	TIM1_BRK_UP_ TRIG_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	TIM3	TIM3 全局中断	0x0000_007C
16	23	-	保留	保留	0x0000_0080
17	24	-	保留	保留	0x0000_0084
18	25	-	保留	保留	0x0000_0088
19	26	-	保留	保留	0x0000_008C
20	27	可设置	LPTIM1	LPTIM1 全局中断	0x0000_0090
21	28	可设置	I2C1	I2C1 全局中断	0x0000_0094
22	29	-	保留	保留	0x0000_0098
23	30	可设置	SPI1	SPI1 全局中断	0x0000_009C
24	31	-	保留	保留	0x0000_00A0
25	32	可设置	UART1	UART1 全局中断	0x0000_00A4
26	33	可设置	UART2	UART2 全局中断	0x0000_00A8
27	34	-	保留	保留	0x0000_00AC
28	35	-	保留	保留	0x0000_00B0
29	36	-	保留	保留	0x0000_00B4
30	37	-	保留	保留	0x0000_00B8
31	38	-	保留	保留	0x0000_00BC

1. 灰色部分对应 Cortex-M0+中断。

## 10 扩展中断和事件控制器（EXTI）

### 10.1 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。

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。

### 10.2 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；
  - 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：
  - 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。

### 10.3 EXTI 模块示意图

EXTI 的输入分为可配置通道和直接通道两种。输出分为事件输出和中断输出。

EXTI 信号处理模块内部实现的功能参见：[EXTI 功能描述](#)。

图 10-1 EXTI 框图

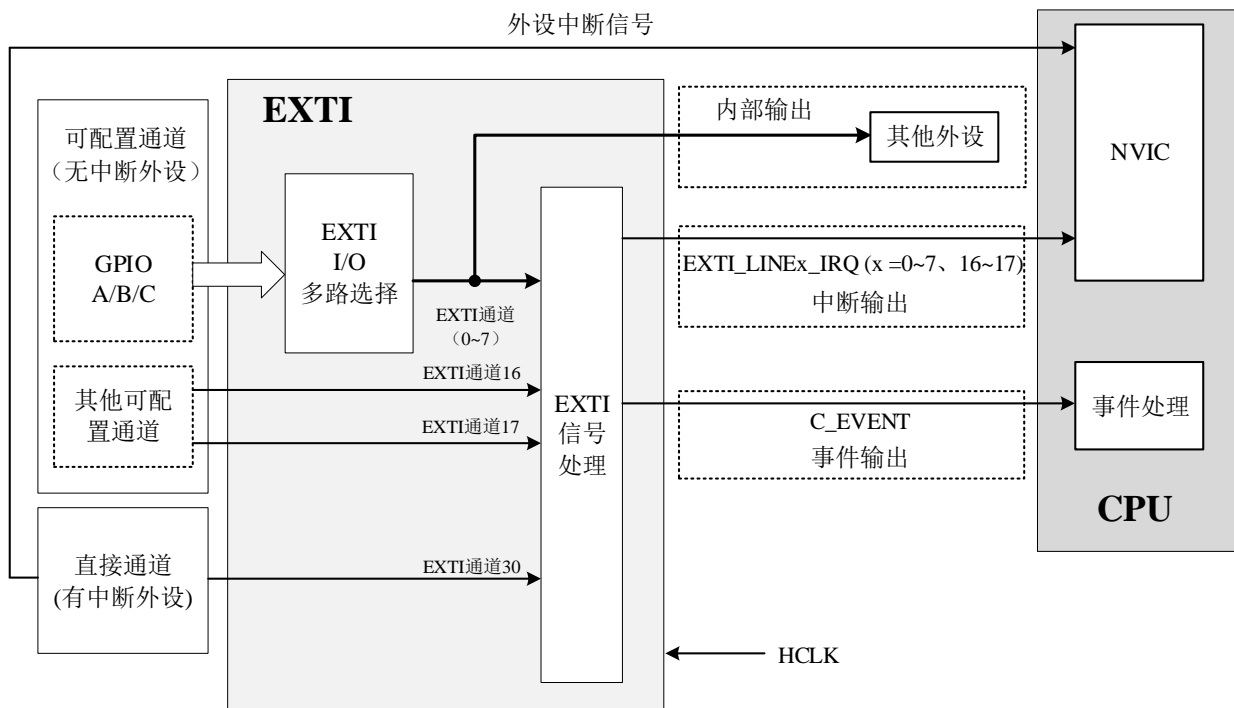


表 10-1 EXTI 内部信号

信号名称	信号类型	说明
EXTI_LINE <sub>x</sub> _IRQ	输出	由可配置通道产生的中断信号（x=0~7、16~17）
C_EVENT	输出	输出到 CPU 的事件输出信号，与 CPU 时钟同步
HCLK	输入	EXTI 时钟，同 AHB 总线时钟

## 10.4 EXTI 的主要信号连接及功能

EXTI 为可配置通道提供了中断挂起标志，挂起标志置 1 代表有中断待处理，需要软件写“1”将标志位清零。EXTI 的中断挂起标志可触发 CPU 中断。

EXTI 中实现了 I/O 多路选择器。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。

本身已实现唤醒事件及中断的外设作为 EXTI 的直接通道。

EXTI 管理来自各个通道的事件，输入到 CPU 的事件处理模块。

EXTI 的事件输出和中断输出可以将 CPU 从低功耗模式下唤醒，唤醒方式主要有以下两种：

- 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。

- 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](#)。

## 10.5 EXTI 通道及信号对应表

表 10-2 EXTI 通道及信号对应表

EXTI 通道	事件源	类型	外设关联的信号
0~7	GPIO	可配置	GPIO 输入信号
16	COMP1	可配置	COMP1 比较器输出结果
17	COMP2	可配置	COMP2 比较器输出结果
30	LPTIM1	直接	中断使能时，自动重载匹配可用于唤醒

## 10.6 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 10-3 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 PIR.PIFn	中断输出	事件输出
0	0	无效	屏蔽	屏蔽
	1	无效	屏蔽	有效
1	0	有效	有效	屏蔽
	1	有效	有效	有效

由可配置通道及直接通道产生的未屏蔽的事件（EXTI\_EMR.EMn=1）可以产生事件请求，也可以作为唤醒源唤醒 CPU。

### 10.6.1 可配置通道

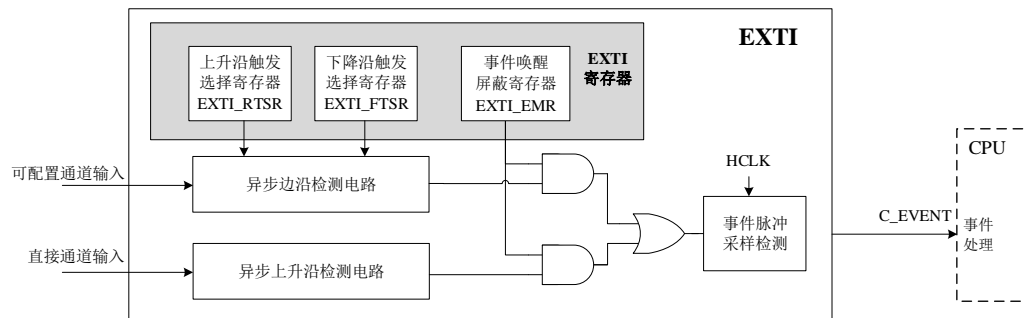
可配置事件的触发沿通过寄存器 EXTI\_RTSR/FTSR 可选为上升沿、下降沿或双沿。

#### 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，其结果输出到 CPU 事件处理模块，可触发 CPU 的事件响应。

参见下图：

图 10-2 可配置通道&直接通道的事件输出框图



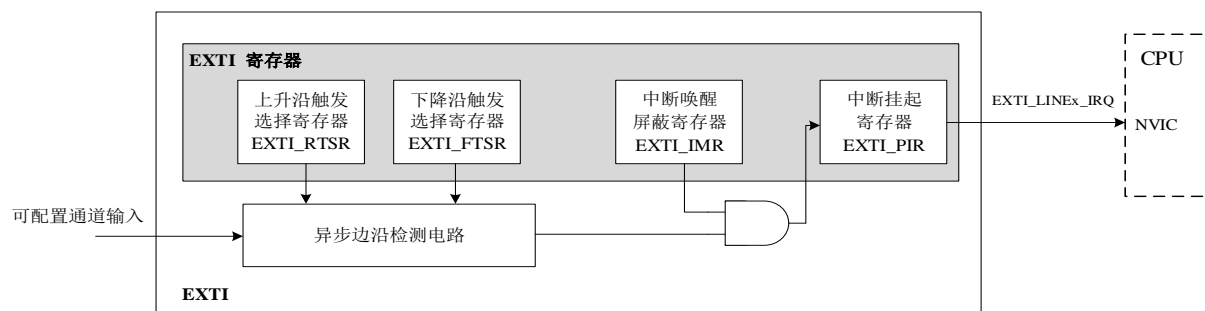
### 可配置通道的中断输出

EXTI 为可配置通道实现中断管理，提供中断挂起寄存器 (EXTI\_PIR)。产生的中断信号 EXTI\_LINE<sub>x</sub>\_IRQ 输入到 NVIC 管理，可触发 CPU 中断。

可配置通道检测到触发沿时，如果中断未屏蔽 (EXTI\_IMR.IM<sub>n</sub>=1)，则中断挂起寄存器 EXTI\_PIR 中的相应 PIF<sub>n</sub> 位被置位，产生中断请求输入到 NVIC，唤醒 CPU 并触发 CPU 中断。后续需将 PIF<sub>n</sub> 位软件写“1”以清除中断请求。

中断挂起寄存器仅响应来自可配置通道且未被屏蔽的 (IM<sub>n</sub>=1) 中断请求。与 EXTI\_EMR 寄存器的配置无关。参见下图：

图 10-3 可配置通道的中断输出框图



## 10.6.2 EXTI 直接通道

来自直接通道的事件或中断都可以唤醒系统。EXTI\_EMR 寄存器和 EXTI\_IMR 寄存器用来设置该事件或中断是否能将系统唤醒。

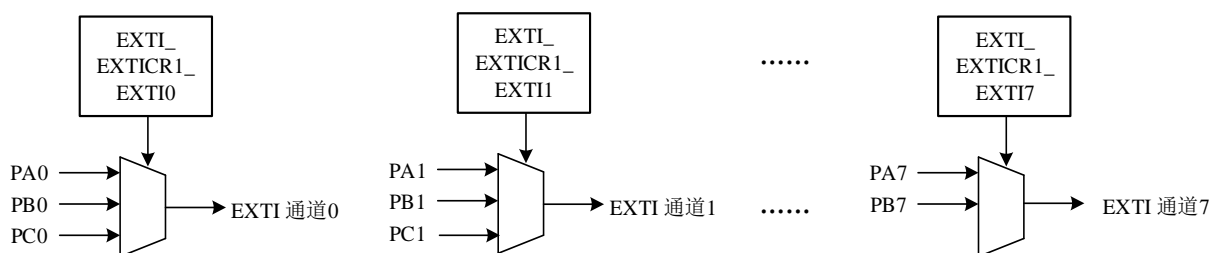
直接通道的中断在外设模块中实现，与 EXTI 无关。直接通道的事件输出逻辑参见图：可配置事件&直接事件的事件输出框图。



### 10.6.3 I/O 多路选择器

EXTI 的 I/O 多路选择器可选任意 GPIO 作为 EXTI 的信号源。所有的 GPIO 作为可配置通道分组接到 8 个多路选择器上，对应 EXTI 通道 0~7。在 EXTI\_EXTICR1 寄存器中选择作为信号源的 GPIO。

图 10-4 EXTI 多路复用器



EXTI 的 I/O 多路选择器的内部输出信号直接连接到其他外设，可作为相关外设的触发源。I/O 多路选择器内部输出到外设的信号与 EXTI\_IMR 和 EXTI\_EMR 寄存器的配置无关。详情参见相关外设模块章节：[模数转换器 \(ADC\)](#)。

## 10.7 EXTI 寄存器

EXTI 寄存器支持 32 位访问。

表 10-4 EXTI 基地址

外设	基地址
EXTI	0x4002 1800

### 10.7.1 EXTI 上升沿触发选择寄存器 (EXTI\_RTSR)

偏移地址: 0x00

复位值: 0x0000 0000

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

位/位域	名称	描述
31:18	保留	写入无效
17:16	RTx	<p>设置EXTI可配置通道 x (x=16~17) 为上升沿触发。</p> <p>每个bit对应一个通道, 使能上升沿触发该通道的事件或中断。</p> <p>0: 禁止</p> <p>1: 使能</p>
15:8	保留	写入无效
7:0	RTx	<p>设置EXTI可配置通道 x (x=0~7) 为上升沿触发。</p> <p>每个bit对应一个通道, 使能上升沿触发该通道的事件或中断。</p> <p>0: 禁止</p> <p>1: 使能</p>

### 10.7.2 EXTI 下降沿触发选择寄存器 (EXTI\_FTSR)

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.														FT17	FT16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	写入无效
17:16	FTx	<p>设置EXTI可配置通道x（x=16~17）为下降沿触发。</p> <p>每个bit对应一个通道，使能下降沿触发该通道的事件或中断。</p> <p>0：禁止</p> <p>1：使能</p>
15:8	保留	写入无效
7:0	FTx	<p>设置EXTI可配置通道x（x=0~7）为下降沿触发。</p> <p>每个bit对应一个通道，使能下降沿触发该通道的事件或中断。</p> <p>0：禁止</p> <p>1：使能</p>

### 10.7.3 EXTI 中断挂起寄存器（EXTI\_PIR）

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.														PIF17	PIF16
														rc_w1	rc_w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w

位/位域	名称	描述
31:18	保留	写入无效

17:16	PIF	在EXTI可配置通道 x (x=16~17) 检测到触发中断并挂起，表示中断待处理。 各个bit在相应通道检测到上升/下降沿中断时置1。 软件写1清零。 0: 未检测到中断挂起请求。 1: 中断请求已挂起。
15:8	保留	写入无效
7:0	PIF	在EXTI可配置通道 x (x=0~7) 检测到触发中断并挂起，表示中断待处理。 各个bit在相应通道检测到上升/下降沿中断时置1。 软件写1清零。 0: 未检测到中断挂起请求。 1: 中断请求已挂起。

#### 10.7.4 EXTI I/O 选择寄存器 (EXTI\_EXTICR1)

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.		EXTI_7[1:0]		Res.		EXTI_6[1:0]		Res.		EXTI_5[1:0]		Res.		EXTI_4[1:0]	
		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.		EXTI_3[1:0]		Res.		EXTI_2[1:0]		Res.		EXTI_1[1:0]		Res.		EXTI_0[1:0]	
		rw	rw			rw	rw			rw	rw			rw	rw

位/位域	名称	描述
31:30	保留	写入无效
29:28	EXTI_7[1:0]	EXTI_7 GPIO端口选择 此位域用于配置EXTI_7的信号源 00: PA7 引脚 01: PB7 引脚 其他: 保留 (写入保留值, 按00处理)
27:26	保留	写入无效

25:24	EXTI_6[1:0]	EXTI_6 GPIO端口选择 此位域用于配置EXTI_6的信号源 00: PA6 引脚 01: PB6 引脚 其他: 保留（写入保留值，按00处理）
23:22	保留	写入无效
21:20	EXTI_5[1:0]	EXTI_5 GPIO端口选择 此位域用于配置EXTI_5的信号源 00: PA5 引脚 01: PB5 引脚 其他: 保留（写入保留值，按00处理）
19:18	保留	写入无效
17:16	EXTI_4[1:0]	EXTI_4 GPIO端口选择 此位域用于配置EXTI_4的信号源 00: PA4 引脚 01: PB4 引脚 其他: 保留（写入保留值，按00处理）
15:14	保留	写入无效
13:12	EXTI_3[1:0]	EXTI_3 GPIO端口选择 此位域用于配置EXTI_3的信号源 00: PA3 引脚 01: PB3 引脚 其他: 保留（写入保留值，按00处理）
11:10	保留	写入无效
9:8	EXTI_2[1:0]	EXTI_2 GPIO端口选择 此位域用于配置EXTI_2的信号源 00: PA2 引脚 01: PB2 引脚 其他: 保留（写入保留值，按00处理）

7:6	保留	写入无效
5:4	EXTI_1[2:0]	EXTI_1 GPIO端口选择 此位域用于配置EXTI_1的信号源 00: PA1 引脚 01: PB1 引脚 10: PC1 引脚 其他: 保留（写入保留值，按00处理）
3:2	保留	写入无效
1:0	EXTI_0[1:0]	EXTI_0 GPIO端口选择 此位域用于配置EXTI_0的信号源 00: PA0 引脚 01: PB0 引脚 10: PC0 引脚 其他: 保留（写入保留值，按00处理）

### 10.7.5 EXTI 中断唤醒屏蔽寄存器 (EXTI\_IMR)

偏移地址: 0x70

复位值: 0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	IM30	Res.												IM17	IM16
	rw													rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
								rw	rw	rw	Rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	写入无效
30	IMx	允许直接通道x的中断唤醒CPU (x=30) 置位表示允许来自此通道的中断将CPU唤醒。清零表示该通道的中断不会唤醒CPU。 0: 禁止

1: 使能

29:18          保留          写入无效

17:16          IMx          允许可配置通道x的中断唤醒CPU（x=17~16）  
置位表示允许来自此通道的中断将CPU唤醒。清零表示该通道的中断不会唤醒CPU。

0: 禁止

1: 使能

15:8          保留          写入无效

7:0          IMx          允许可配置通道x的中断唤醒CPU（x=0~7）  
置位表示允许来自此通道的中断将CPU唤醒。清零表示该通道的中断不会唤醒CPU。

0: 禁止

1: 使能

## 10.7.6      **EXTI 事件唤醒屏蔽寄存器（EXTI\_EMR）**

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.	EM30	Res.												EM17	EM16
	rw													rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
								rw	rw	rw	Rw	rw	rw	rw	rw

位/位域	名称	描述
31	保留	写入无效
30	EMx	允许直接通道x的事件唤醒CPU（x=30） 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。 0: 禁止 1: 使能

29:18	保留	写入无效
17:16	EMx	<p>允许可配置通道x的事件唤醒CPU（x=16~17）</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：禁止</p> <p>1：使能</p>
15:8	保留	写入无效
7:0	EMx	<p>允许可配置通道x的事件唤醒CPU（x=0~7）</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：禁止</p> <p>1：使能</p>



## 11 循环冗余校验（CRC）

### 11.1 简介

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

### 11.2 CRC 主要特性

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

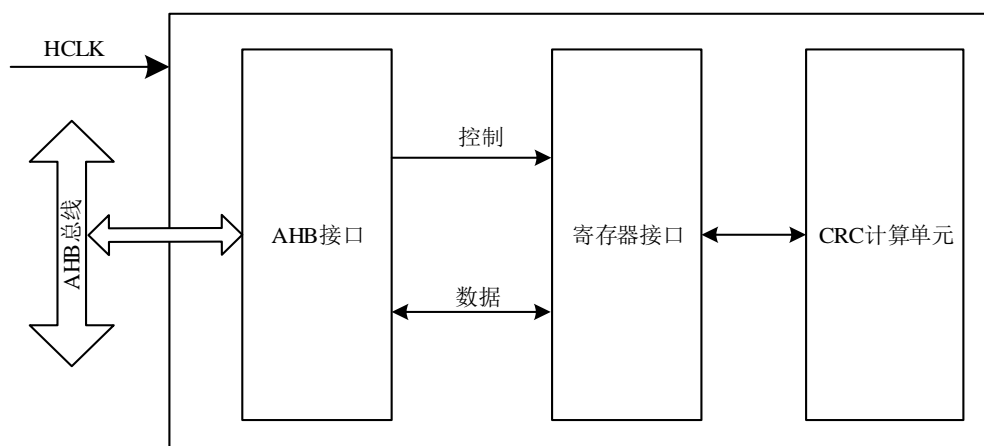
$$X^{16}+X^{12}+X^5+1$$
  - CRC-32 多项式：0x4C11DB7  

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$
- 1 个 AHB 时钟周期完成 8 位 CRC 计算

### 11.3 CRC 功能描述

#### 11.3.1 CRC 框图

图 11-1 CRC 框图

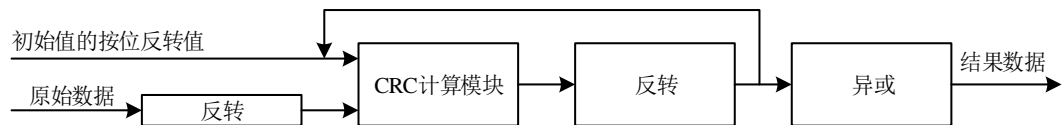


#### 11.3.2 CRC 操作说明

CRC 的数据寄存器 CRC\_DR 用于输入原始数据。CRC 的结果寄存器 CRC\_RDR 用于输出计算生成的 CRC 值和输入初始值。向 CRC\_RDR 寄存器输入初始值时，初始值需要按位反转后写入，参见[图：输出数据反转框图](#)。

CRC 的计算单元将输入到 CRC\_DR 的原始数据反转后，与 CRC\_RDR 初始值的按位反转值(或者前一次计算生成的数据反转后的值)输入到CRC计算模块，计算生成的数据反转后与 0xFFFFFFFF (或者 0xFFFF) 异或，异或生成的结果数据输出到 CRC\_RDR。见下图。

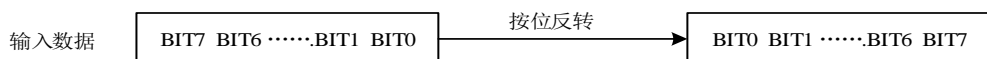
图 11-2 CRC 计算单元框图



将输入数据按 8 位的数据格式依次输入 CRC\_DR 寄存器；1 个 AHB 时钟周期即可完成 8 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。

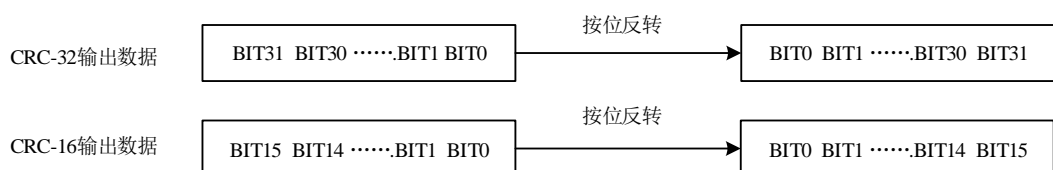
CRC 输入数据按位反转。例如：输入数据 0x11, 0x22, 0x33, 0x44，执行反转后数据为 0x88, 0x44, 0xCC, 0x22。见下图。

图 11-3 输入数据反转框图



CRC 输出数据按位反转并且与 0xFFFFFFFF (或 0xFFFF) 异或。例如：CRC-32 输出数据 0x11223344，执行反转后数据为 0x22CC4488，再执行异或后数据为 0xDD33BB77。见下图。

图 11-4 输出数据反转框图



CRC 计算操作步骤如下：

- 1) 配置多项式，将多项式配置值写入控制状态寄存器 CRC\_CSR 中 POLY\_SIZE 位。CRC-16 多项式配置值为 0，CRC-32 多项式配置值为 1。
- 2) 将初始值按位反转后写入结果寄存器 CRC\_RDR。
- 3) 将待计算的原始数据按 8 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。

## 11.4 CRC 寄存器

CRC 寄存器支持 32 位访问。

表 11-1 CRC 基地址

外设	基地址
CRC	0x4002 3000

### 11.4.1 CRC 控制状态寄存器 (CRC\_CSR)

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.															POLY_SIZE
															rw

位/位域	名称	描述
31:1	保留	写入无效
0	POLY_SIZE	CRC 多项式选择位 0: CRC-16 1: CRC-32

### 11.4.2 CRC 结果寄存器 (CRC\_RDR)

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

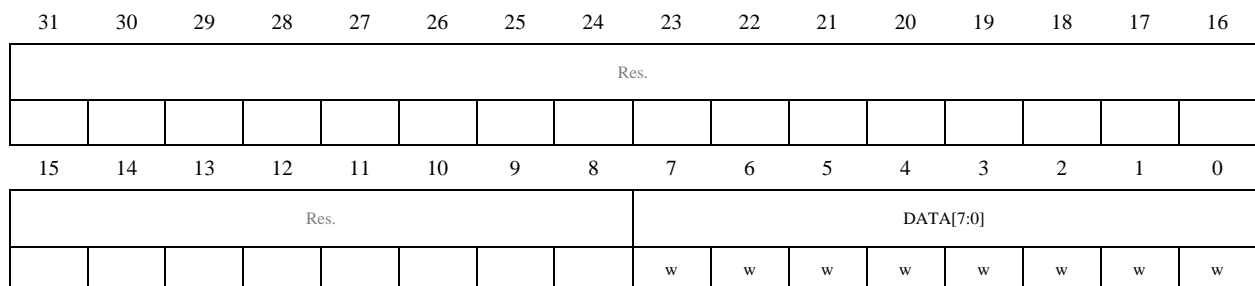
位/位域	名称	描述
------	----	----

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。

### 11.4.3 CRC 数据寄存器 (CRC\_DR)

偏移地址：0x80

复位值：0x0000 0000



位/位域	名称	描述
31:8	保留	写入无效
7:0	DATA[7:0]	输入数据

## 12 模数转换器（ADC）

### 12.1 简介

芯片内置 12 位逐次逼近型模数转换器（SAR ADC），可以对 8 路外部通道和 1 路内部通道的电压信号进行模数转换，可以将通道序列转换配置为单次扫描、循环扫描或循环间断转换模式。ADC 转换结果存储在 12 位数据寄存器。

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

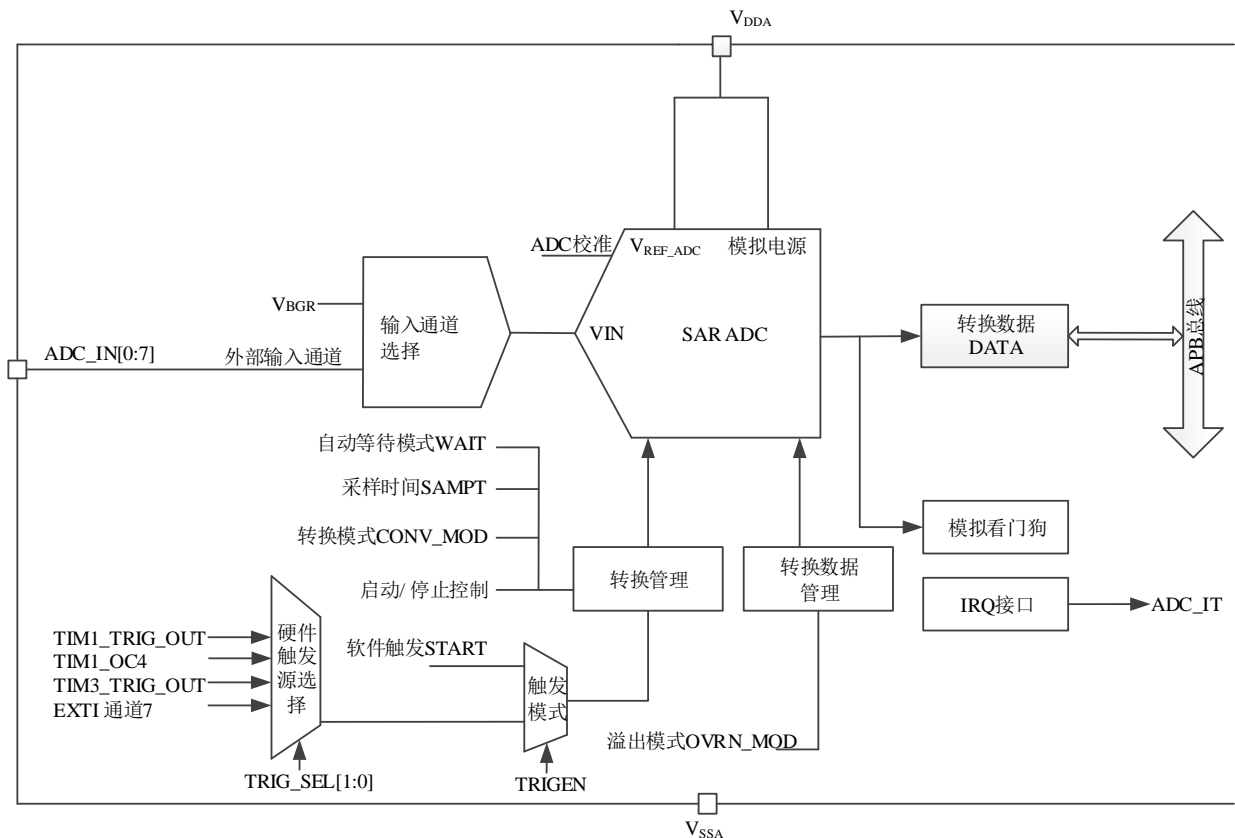
### 12.2 ADC 主要特性

- 工作电压 1.8V~5.5V
- 参考电压  $V_{REF\_ADC}$ :  $V_{DDA}$
- 输入信号幅度  $0 \sim V_{REF\_ADC}$
- 12 位分辨率最高采样率 1Msps
- 模拟输入通道
  - 8 路外部输入通道，可测量高输出阻抗信号；
  - 1 路内部输入通道： $V_{BGR}$ 。
- 转换模式
  - 单次扫描模式；
  - 循环扫描模式；
  - 循环间断模式。
- 启动方式
  - 软件触发；
  - 硬件触发：  
触发源为内部定时器事件或 GPIO 输入事件；  
触发极性可配置（上升沿、下降沿、双沿）。
- 可配置采样时间

- 自动等待模式
- 模拟看门狗

## 12.3 ADC 功能描述

图 12-1 ADC 结构框图



### 12.3.1 ADC 引脚和内部信号

表 12-1 ADC 输入引脚

引脚信号名称	信号类型	说明
V <sub>DDA</sub>	输入，模拟电源	ADC 模拟电源
	输入，参考电压源	ADC 参考电源
V <sub>SSA</sub>	输入，模拟地	ADC 模拟地
ADC_IN <sub>x</sub>	模拟输入	8 路外部输入通道

表 12-2 ADC 内部信号

内部信号名称	信号类型	说明
TIM1_TRIG_OUT、 TIM1_OC4_ADC、	输入	ADC 转换硬件触发源

内部信号名称	信号类型	说明
TIM3_TRIG_OUT、 EXTI 通道 7		
V <sub>BGR</sub>	输入	片内 BGR 输出电压

### 12.3.2 时钟源

ADC 工作时钟 ADC\_CK 源于 APB 总线时钟 PCLK，并可以通过 ADC\_CFG2 寄存器中 PRESC[2:0]位域进行 1、2、3、4、8、16、32、64 分频。

ADC 工作在 16MHz 时的配置方法：

- SYSCLK 选择 RCH, AHB 和 APB 预分频器选择 1 分频，通过 ADC\_CFG2 寄存器的 PRESC[2:0]位选择 3 分频；
- SYSCLK 选择 RCHDIV3, AHB 和 APB 预分频器选择 1 分频，通过 ADC\_CFG2 寄存器的 PRESC[2:0]位选择 1 分频。

ADC 时钟频率与工作电压之间的关系如下：

- $2.2\text{ V} < V_{\text{DDA}} \leq 5.5\text{ V}$  时， $300\text{ KHz} \leq f_{\text{ADC\_CK}} \leq 16\text{ MHz}$
- $1.8\text{ V} \leq V_{\text{DDA}} \leq 2.2\text{ V}$  时， $300\text{ KHz} \leq f_{\text{ADC\_CK}} \leq 8\text{ MHz}$

**注意：** ADC 时钟 ADC\_CK 要求占空比 50%（典型值，占空比范围为 40%~60%）。为此当 PRESC[2:0] 选择 PCLK 的 1 分频作为 ADC 时钟源时，需要选择占空比为 50% 的系统时钟并禁止 AHB 和 APB 预分频器对系统时钟进行分频。

### 12.3.3 使能/禁止控制

ADC 使能、禁止由 ADC\_CR 寄存器的 ADEN 和 ADDIS 位进行控制：

使能 ADC 步骤：

- 1) 将 ADC\_CR 寄存器中 ADEN 位置 1；
- 2) 等待 ADC 启动稳定，启动稳定时间 1μs。

禁止 ADC 步骤：

- 1) 查询 ADC\_CR 寄存器中 START 位为 0，以确认当前未执行任何转换；若 START 位不为 0，可向 ADC\_CR 寄存器 STOP 位写 1，并等待此位清 0，以停止正在进行的转换；
- 2) 将 ADC\_CR 寄存器中 ADDIS 位置 1；
- 3) 查询 ADC\_CR 寄存器中 ADEN 位为 0，确认 ADC 已禁止。

#### 12.3.4 校准

由于生产制造所产生的差异，各芯片 ADC 的偏移误差有所不同。ADC 具有校准功能，用于消除偏移误差。

校准应在使能 ADC 后，启动 ADC 转换之前进行，通过将 ADC\_CR 寄存器的 CALEN 位置 1 启动校准。校准过程中 CALEN 一直保持为 1，校准完成后由硬件清 0，EOCAL 置 1。

校准完成后，可从 ADC\_CALFACT 或 ADC\_DR 寄存器读取校准系数，此校准系数会一直应用于 ADC，即使禁止 ADC 后仍然保留，直至系统复位或 RCC 外设复位时，校准系数会被清除。

如果 ADC 已使能但未进行转换，可向校准系数寄存器 ADC\_CALFACT 写入校准系数。下次转换启动时，校准系数会自动加载到 ADC。校准系数载入过程不会对转换的启动造成延迟。

当 ADC 运行条件发生改变（ $V_{DDA}$  变化是造成偏移误差变化的主要原因，温度变化次之），建议重新进行校准。

ADC 校准步骤：

- 1) ADEN 置 1，并等待 ADC 启动稳定；
- 2) CALEN 置 1，使能校准；
- 3) 查询 EOCAL 位置 1，以等待校准完成。如果已将 ADC\_IER 寄存器中 EOCALIE 位置 1，同时会产生校准中断，可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 或 ADC\_DR 寄存器读取。

为进一步提高 ADC 转换数据精度，需根据出厂补偿值（ADC\_COMPENSATION\_VALUE）对校准系数（ADC\_CALFACT）进行补偿。ADC\_COMPENSATION\_VALUE 为 32 位有符号整型数据，保存地址 0x1FFF03CC，具体格式参见表：出厂校准参数。

校准系数补偿步骤：

- 1) 按照 ADC 校准步骤，获取校准系数 ADC\_CALFACT 初值，该值为 6 位有符号整型数据；
- 2) ADC\_CALFACT 初值扩充为 32 位有符号整型数据，并减去 ADC\_COMPENSATION\_VALUE 作为新的校准系数，写入到 ADC\_CALFACT 寄存器。



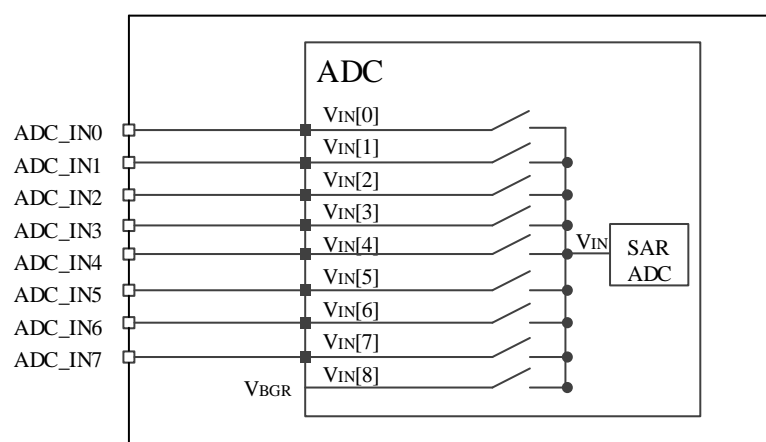
### 12.3.5 转换通道

ADC 模块具有 9 路输入通道：

- 8 路外部输入通道，通过 GPIO 引脚输入（ADC\_IN<sub>x</sub>）；
- 1 路内部输入通道（V<sub>BGR</sub>）：
  - 内部参考电压 V<sub>BGR</sub> 连接至通道 ADC\_VIN[8]；

ADC 输入通道见下图：

图 12-2 ADC 输入通道



### 转换通道选择

ADC 可以转换单个通道，也可以对一组通道序列自动扫描转换。待转换通道需在 ADC\_CHCFG 通道配置寄存器中进行选择。

- 通道序列长度由 ADC\_CHCFG 寄存器中 CHN<sub>x</sub> 位确定
- 通道序列扫描方向可以配置为正向扫描或反向扫描，由 ADC\_CFG1 寄存器中 SDIR 位确定（SDIR=0：正向扫描，即通道 0~8；SDIR=1：反向扫描，即通道 8~0）。

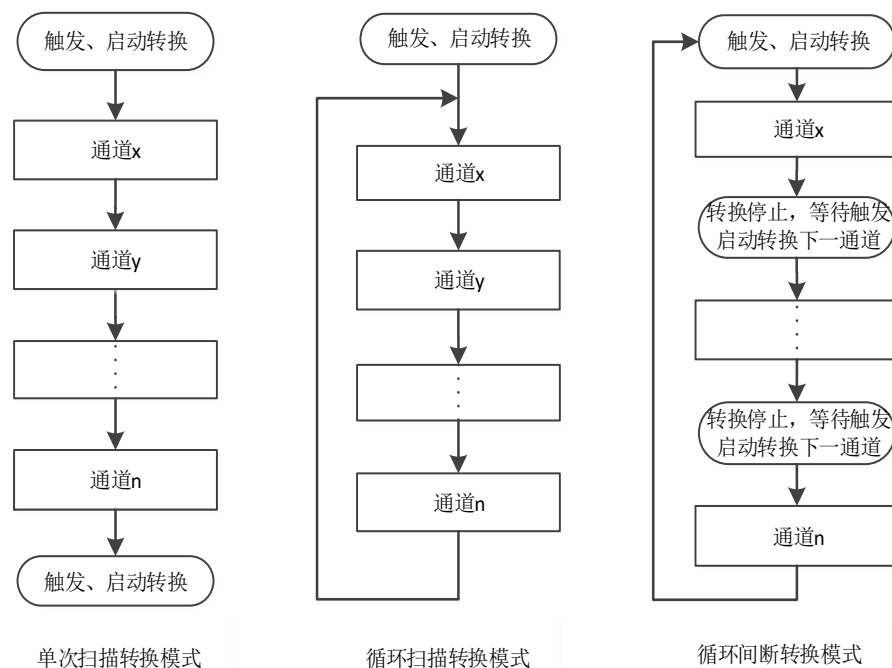
示例：转换通道选择 0、3、7、8，ADC\_CHCFG 寄存器配置如下：

- ADC\_CHCFG 寄存器中 CHN0、CHN3、CHN7、CHN8 位置 1 分别使能通道 0、3、7、8。根据 SDIR 位配置，扫描方向只能选择 0、3、7、8（SDIR = 0）或 8、7、3、0（SDIR = 1）。

### 12.3.6 转换模式

ADC 通道转换可设置为具有单次扫描、循环扫描、循环间断三种转换模式。

图 12-3 ADC 转换模式



### 单次扫描转换模式

单次扫描转换模式下，当软件触发或外部硬件触发事件产生，ADC 对所配置通道序列只进行一次转换。通道序列转换结束后，ADC 停止工作，等待下一次触发事件产生。ADC\_CFG1 寄存器中 CONV\_MOD[1:0]位域设置为“00”，选择此模式。

启动转换方式：

- 软件触发，将 ADC\_CR 寄存器中 START 位置 1，立即启动转换；
- 硬件触发，ADC\_CR 寄存器中 START 位置 1 后，当外部硬件触发事件产生，立即启动转换。

在通道序列转换过程中，每个通道转换完成后，ADC 执行以下操作：

- 转换数据存储在 12 位 ADC\_DR 寄存器；
- 转换通道结束标志（EOC）置 1；
- 如果转换通道结束中断使能（EOCIE 置 1），则产生中断。

在通道序列的所有通道转换完成后，ADC 执行以下操作：

- 转换序列结束标志（EOS）置 1；
- 如果转换序列结束中断使能（EOSIE 置 1），则产生中断。

随后，ADC 停止工作，等待下一次外部硬件触发事件产生，或软件触发 START 位再次置 1。

### 循环扫描转换模式

循环扫描转换模式下，当软件触发或外部硬件触发事件产生，ADC 对所配置通道序列进行循环转换。通道序列转换完成后，ADC 自动循环执行相同的转换序列。ADC\_CFG1 寄存器中 CONV\_MOD[1:0]位域设置为“01”，选择此模式。

启动转换方式：

- 软件触发，将 ADC\_CR 寄存器中 START 位置 1，立即启动转换；
- 硬件触发，ADC\_CR 寄存器中 START 位置 1 后，当外部硬件触发事件产生，立即启动转换。

在通道序列转换过程中，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；
- 转换通道结束标志（EOC）置 1；
- 如果转换通道结束中断使能（EOCIE 置 1），则产生中断。

在每次通道序列中所有通道转换完成后，ADC 执行以下操作：

- 转换序列结束标志（EOS）置 1；
- 如果转换序列结束中断使能（EOSIE 置 1），则产生中断。

随后，ADC 继续对相同的通道序列进行转换。

### 循环间断转换模式

循环间断转换模式下，通道序列中每个通道都需要通过软件触发或外部硬件触发才可以启动转换。当某一通道转换完成，ADC 停止转换，等待下一次触发事件，之后进行下一通道的转换。通道序列转换完成后，ADC 执行相同的转换序列。ADC\_CFG1 寄存器中 CONV\_MOD[1:0]位域设置为“10”，选择此模式。

示例：扫描间断转换模式，待转换通道配置为 0、3、7、8，转换流程如下：

- 第一次触发：转换通道 0，转换完成时产生 EOC 事件；
- 第二次触发：转换通道 3，转换完成时产生 EOC 事件；
- 第三次触发：转换通道 7，转换完成时产生 EOC 事件；
- 第四次触发：转换通道 8，转换完成时，同时产生 EOC 和 EOS 事件；

- 第五次触发：转换通道 0，转换完成时产生 EOC 事件；
- 第六次触发：转换通道 3，转换完成时产生 EOC 事件；
- ...

### 12.3.7 启动/停止转换

#### 启动转换

ADC 启动电压转换有以下两种方式：

- 如果配置为软件触发方式（ADC\_CFG1 寄存器中 TRIGEN[1:0]=00），ADC\_CR 寄存器中 START 位置 1 后立即启动电压转换；
- 如果配置为外部硬件触发方式（ADC\_CFG1 寄存器中 TRIGEN[1:0]≠00），ADC\_CR 寄存器中 START 位置 1 后，根据所配置触发极性，在下一个触发事件产生时，启动电压转换。

当 ADC 正处于电压转换过程中，任何触发事件会被忽略。如果寄存器位 START = 0，则会忽略任何外部硬件触发事件。

ADC\_CFG1 寄存器中 TRIGEN[1:0]位域与触发极性之间的对应关系见下表。仅当 START= 0 时，才可以更改触发方式及极性。

表 12-3 触发方式及极性

触发方式及极性	TRIGEN[1:0]
软件触发方式	00
外部硬件事件上升沿触发	01
外部硬件事件下降沿触发	10
外部硬件事件上升沿和下降沿均触发	11

ADC\_CFG1 寄存器中 TRIG\_SEL[1:0]位域用于选择具体外部硬件事件触发源，见下表。仅当 START = 0 时，才可以更改外部硬件事件触发源。

表 12-4 外部硬件触发源

名称	触发源	TRIG_SEL[1:0]
TRG0	TIM1_TRIG_OUT	00
TRG1	TIM1_OC4_ADC	01
TRG2	TIM3_TRIG_OUT	10
TRG3	EXTI 通道 7	11

ADC\_CR 寄存器中 START 位也用于指示当前 ADC 是否正在处于工作状态。

START 位置 1 指示当前 ADC 处于工作状态；当 START 位清 0 时，指示 ADC 处于空闲状态，此时可以重新配置 ADC。

- 对于所有转换模式、转换启动触发方式（CONV\_MOD[1:0]=XX、TRIGEN[1:0]=XX），软件执行 STOP 流程时，此位清 0；
- 在以下情况，由硬件清 0：
  - 单次扫描转换模式，软件触发启动方式（CONV\_MOD[1:0]=00，TRIGEN[1:0]=00），通道序列转换结束时(EOS=1)，此位清 0。
  - 循环间断转换模式，软件触发启动方式（CONV\_MOD[1:0]=10，TRIGEN[1:0]=00），每个通道转换结束时(EOC=1)，此位清 0。
- 在以下情况，不会由硬件清 0：
  - 循环扫描转换模式（CONV\_MOD[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=1），START 位不会被硬件清 0。
  - 单次扫描转换模式（CONV\_MOD[1:0]=00），当选择外部硬件触发方式时（TRIGEN[1:0]≠00），通道序列转换完成后（EOS=1），START 位不会被硬件清 0。这种情况下，START 位保持为 1，以确保不会错过下一次触发事件。
  - 循环间断转换模式（CONV\_MOD[1:0]=10），当选择外部硬件触发方式时（TRIGEN[1:0]≠00），每个通道转换完成后（EOC=1），START 位不会被硬件清 0。

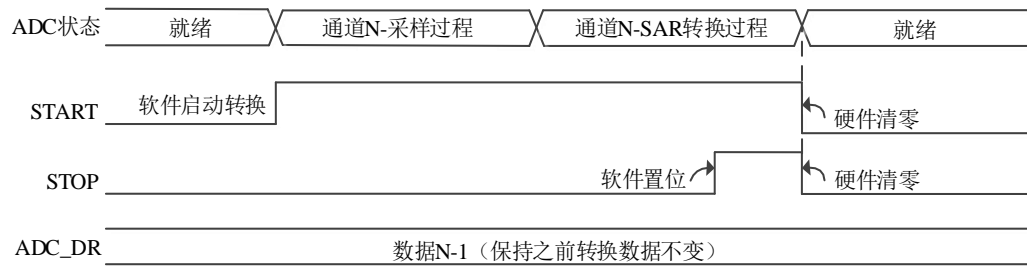
### 停止转换

可通过将 ADC\_CR 寄存器中 STOP 位置 1，停止任何正在进行的转换。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。

STOP 位置 1 后，正在进行的转换会被停止，当前转换结果会被丢弃（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并复位，这意味着重启 ADC 将会重新开始新的转换序列。

一旦停止转换过程完成，ADC\_CR 寄存器中 STOP 位和 START 位均会被硬件清 0。软件必须等待 START 清 0，然后才能开始进行新的转换。

图 12-4 停止正在进行的转换



### 转换结束标志

当通道序列中每个通道转换完成，转换数据结果输出到 ADC\_DR 寄存器后，ADC\_ISR 寄存器中 EOC 标志置 1。此时如果 ADC\_IER 寄存器中转换完成中断使能位 EOCIE 置 1，可产生转换完成中断。EOC 标志可通过软件向其写 1 或者读取 ADC\_DR 寄存器数据的方式清 0。

当 ADC 电压转换的采样阶段结束时，ADC\_ISR 寄存器中 EOSAMP 标志置 1。此时如果 ADC\_IER 寄存器中采样阶段完成中断使能位 EOSAMPIE 置 1，可产生采样结束中断。EOSAMP 标志通过软件向其写 1 清 0。

当通道序列转换结束时（即通道序列中最后一个通道转换完成，在 ADC\_DR 寄存器得到相应转换结果），ADC\_ISR 寄存器中 EOS 标志位置 1。此时如果 ADC\_IER 寄存器中通道序列完成中断使能位 EOSIE 置 1，可产生通道序列转换完成中断。EOS 标志通过软件向其写 1 清 0。

**注意：** STOP 位置 1 后，AWDG、OVRN、EOSAMP、EOC、EOS 均会清零；禁止 ADC 后，所有标志清零。

### 12.3.8 转换时序示例

以下以通道 0、3、7、8 为例，展示 ADC 转换时序。

图 12-5 单次扫描转换模式，软件触发

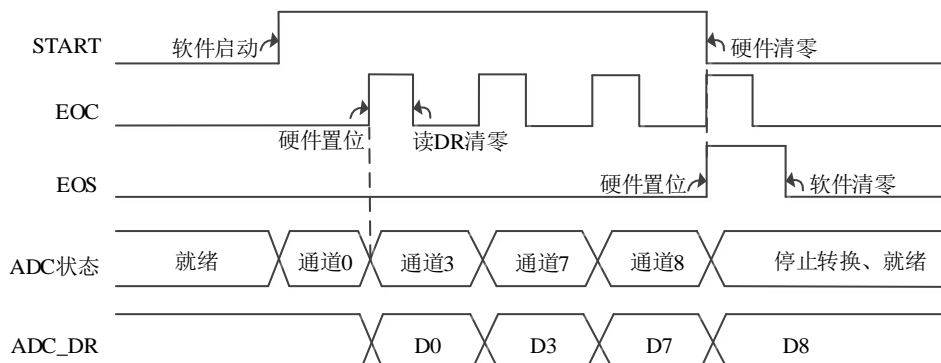


图 12-6 循环扫描转换模式，软件触发

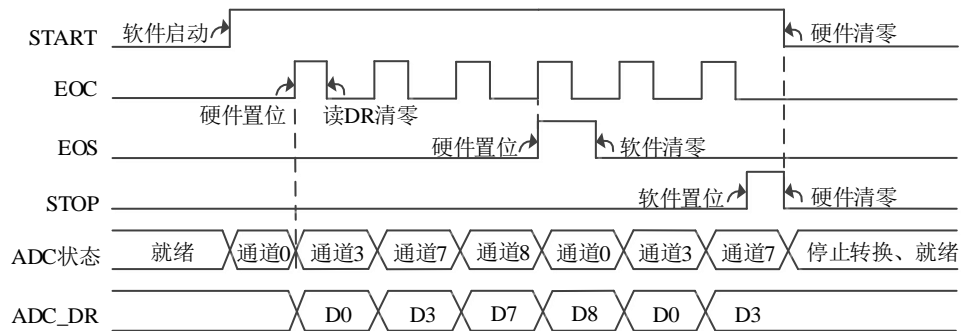


图 12-7 单次扫描转换模式，硬件触发

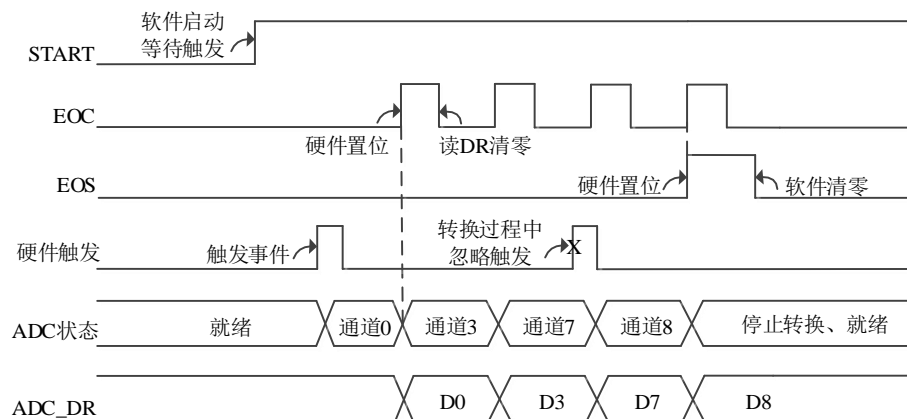
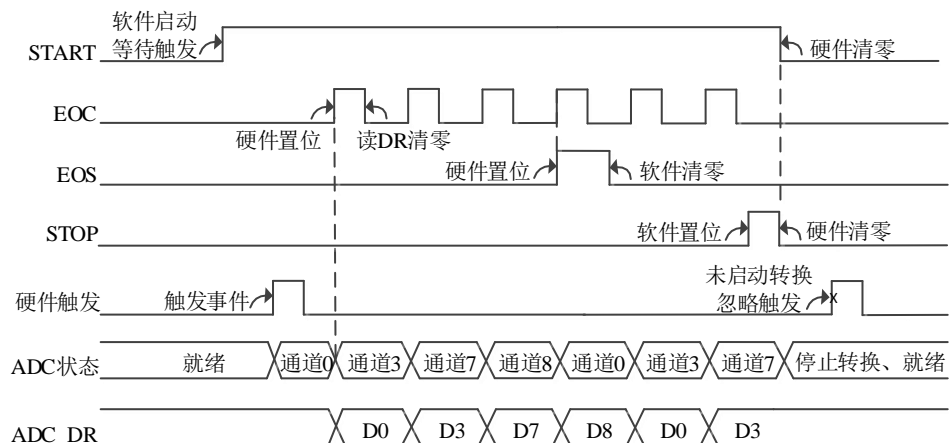


图 12-8 循环扫描转换模式，硬件触发

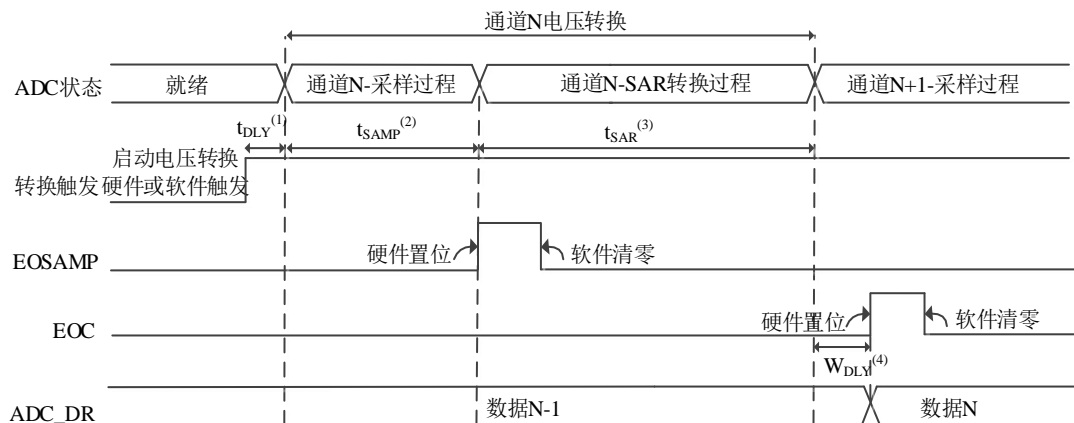


### 12.3.9 采样时间

ADC 电压模数转换过程包括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（ $t_{SAMP}$ ）与逐次逼近时间（ $t_{SAR}$ ）的总和。

采样时间（ $t_{SAMP}$ ）由用户根据输入信号来具体配置，逐次逼近时间（ $t_{SAR}$ ）由 ADC 分辨率确定。

图 12-9 ADC 转换时序



1.  $t_{DLY}$ : 触发延迟;
2.  $t_{SAMP}$ : 采样时间, 由寄存器 `ADC_SAMPT` 配置;
3.  $t_{SAR}$ : 转换时间;
4.  $W_{DLY}$ : 寄存器 `ADC_DR` 写入延迟。

在进行电压转换之前, ADC 需要在待测量信号与内置采样电容之间建立直接连接。用户需根据输入信号的阻抗配置采样时间, 该采样时间必须足以使输入信号电压为采样电容充电, 并将电容电压保持在输入信号电压水平。

ADC 会在一定时钟周期 (`ADC_CK` 周期) 内对输入信号电压进行采样, 该采样时钟周期数由 `ADC_SAMPT` 寄存器对所有通道进行统一设置。

ADC 进行电压模数转换的总转换时间计算公式如下:

$$t_{CONV} = t_{SAMP} + t_{SAR} = t_{SAMP} + 13 \times t_{ADC\_CK}$$

其中:  $t_{SAR}$  为固定值, 即 13 个 `ADC_CK` 时钟周期。

示例:

如果 `ADC_CK` 频率为 16MHz, 采样时间为 3 个 ADC 时钟周期, 则转换时间计算如下:

$$t_{CONV} = (3 + 13) \times t_{ADC\_CK} = 16 \times t_{ADC\_CK} = 1 \mu s$$

### 12.3.10 数据溢出

由于 ADC 所有通道的转换数据都会输出到同一个数据寄存器 `ADC_DR`, 因此如果之前的转换数据未被 CPU 及时处理 (读取转换数据或向 `EOC` 标志写 1 清 0), 当新转换完成而向数据寄存器 `ADC_DR` 输出转换结果时, 会产生数据溢出事件。



当新转换完成，如果此时 EOC 标志仍为“1”，ADC\_ISR 寄存器中数据溢出标志 OVRN 会置 1。如果 ADC\_IER 寄存器中数据溢出中断使能位 OVRNIE 置 1，可产生数据溢出中断。OVRN 标志位通过软件写 1 清 0。

当产生数据溢出事件，ADC 会继续保持工作状态并可以继续电压转换，除非通过软件将 ADC\_CR 寄存器的 STOP 位置 1，从而停止转换并复位转换通道序列。

ADC\_CFG1 寄存器中 OVRN\_MOD 位用于配置发生数据溢出事件时，数据寄存器 ADC\_DR 继续保留之前的转换数据，还是被新的转换数据覆盖。

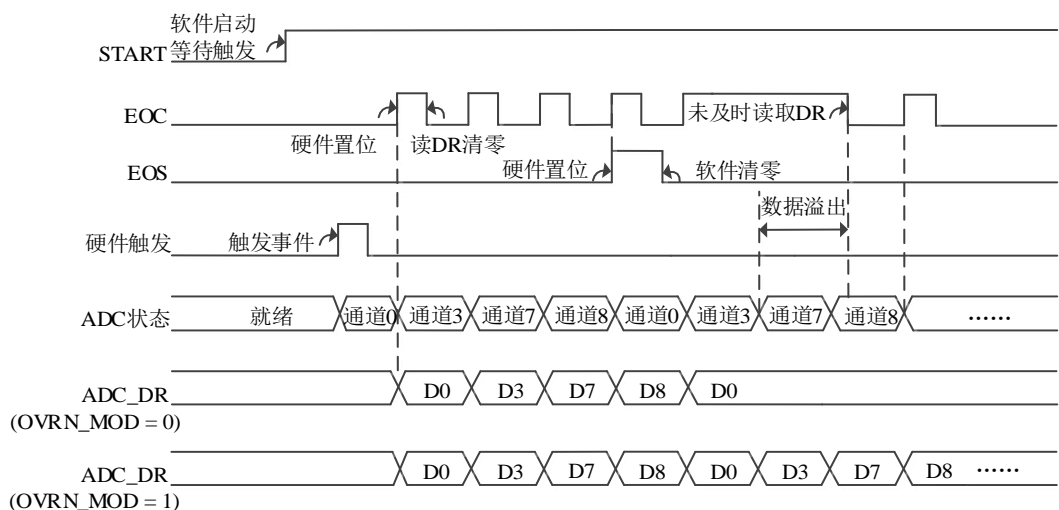
#### ● OVRN\_MOD=0

当产生数据溢出事件，ADC\_DR 数据寄存器中之前的转换数据会被保留，防止其被覆盖，并丢弃新的转换结果。如果 OVRN 位保持为 1，ADC 可以继续转换，但会丢弃后续转换数据。

#### ● OVRN\_MOD=1

当产生数据溢出事件，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 OVRN 位保持为 1，ADC 可以继续转换，ADC\_DR 寄存器始终存储最新转换的数据。

图 12-10 数据溢出处理示例<sup>(1)</sup>



1. 转换通道选择 0、3、7、8，正向扫描，循环扫描转换模式，硬件触发。

### 12.3.11 自动等待模式

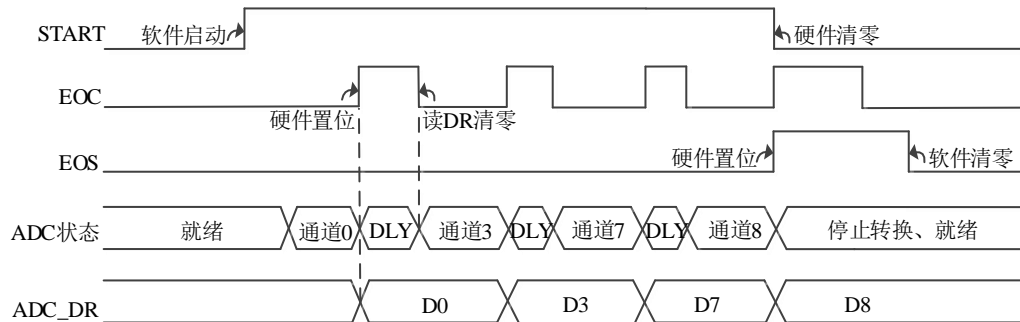
自动等待模式可以自动调整 ADC 通道序列转换过程，使其适应系统读取转换数据的速度，从而简化软件设计。

ADC\_CFG1 寄存器中 WAIT\_MOD 位置 1 使能自动等待模式。此时仅当之前的

转换数据已进行处理，即 ADC\_DR 寄存器已读取或者 EOC 位清 0 后，ADC 才开始新的转换。

**注意：** ADC 转换过程、读数据寄存器或清除 EOC 之前的等待时间内，硬件触发事件会被忽略。

图 12-11 自动等待模式<sup>(1)</sup>



1. 转换通道选择 0、3、7、8，正向扫描，单次扫描转换模式，软件触发。

### 12.3.12 模拟看门狗

#### 模拟看门狗简介

ADC 模拟看门狗，用于监控通道输入电压是否在所配置的阈值范围之内。

通过设置 ADC\_AWDGCR 寄存器中 CHNy (y=0~8) 位，模拟看门狗对所选定转换通道的输入电压进行监控。当 CHNy 位置 1 时，可使能相应通道的模拟看门狗。

模拟看门狗监控通道输入电压的方式是对 12 位转换数据与所设置阈值进行比较。监控电压阈值上下限分别在 ADC\_AWDGTR 寄存器中的 AWDG\_HT[11:0] 和 AWDG\_LT[11:0] 位域进行设置。

如果转换电压低于阈值下限或高于阈值上限，ADC\_ISR 寄存器中模拟看门狗监控电压状态标志位 AWDG 置 1。此时如果 ADC\_IER 寄存器中断使能位 AWDGIE 置 1，会产生模拟看门狗监控电压中断。通过向 AWDG 状态标志位写 1，可清 0 此标志。

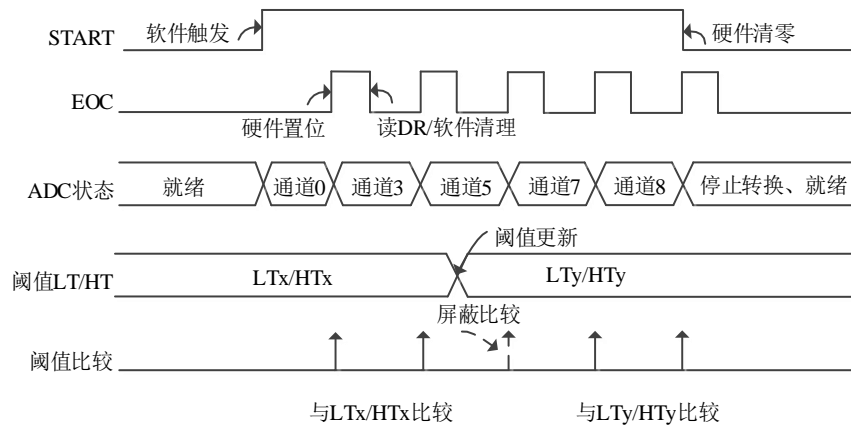
#### 模拟看门狗监控电压阈值控制

模拟看门狗监控电压阈值上下限 AWDG\_HT[11:0]、AWDG\_LT[11:0] 可以在 ADC 电压转换过程中(在转换开始和转换结束之间)进行改变。

如果在 ADC 电压转换过程中改变了监控电压阈值，则模拟看门狗会屏蔽本次监控电压比较。开始新转换时，会清除此屏蔽功能，新的监控电压阈值将会应用于下一次 ADC 转换。

ADC 会在每一次转换结束时进行模拟看门狗监控电压比较。在阈值更新过程中，如果当前 ADC 转换电压超出了新设置的阈值范围，AWDG 标志状态不变，不会产生中断。

图 12-12 模拟看门狗阈值更新<sup>(1)</sup>



1. 转换通道选择 0、3、5、7、8，正向扫描，单次扫描转换模式，软件触发。

ADC 模拟看门狗监控通道 0、3、5、7、8。

### 12.3.13 内部参考电压检测

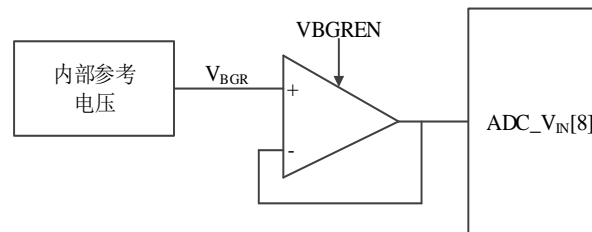
内部带隙基准  $V_{BGR}$  为 ADC 提供了一个稳定的电压输出。 $V_{BGR}$  连接到 ADC\_VIN[8]内部输入通道。由于制造工艺的差异，每颗芯片的  $V_{BGR}$  输出电压不同。在生产过程中对每颗芯片的  $V_{BGR}$  电压单独进行了校准，校准温度、电压条件及校准值数据存储区地址如下表。用户可以使用此精确电压值来计算实际  $V_{REF\_ADC}$  电压。

表 12-5 内部带隙基准电压校准参数

校准参数名称	校准温度、电压等条件	存储区地址
BGR_CAL	温度: 25 °C ( $\pm 2$ °C), $V_{DDA} = V_{REF\_ADC} = 3.3 \text{ V} (\pm 10 \text{ mV})$	0x1FFF03C0

其中 BGR\_CAL 是在  $V_{REF\_ADC} = V_{DDA} = 3.3\text{V}$  条件下，由 ADC 转换得到的内部参考电压输出值  $V_{BGR}$ 。

内部带隙基准电压与 ADC 之间连接结构如下图所示。使用时需通过将 ADC\_CFG2 寄存器中 VBGREN 位置 1 使能 ADC\_VIN[8] ( $V_{BGR}$ ) 的转换。

图 12-13 V<sub>BGR</sub> 输入通道结构

### 使用内部带隙基准电压 V<sub>BGR</sub> 计算实际的 ADC 参考电压 V<sub>REF\_ADC</sub>

ADC 的参考电压可能会有变化，或无法获得准确值。制造过程中在 V<sub>REF\_ADC</sub> = 3.3V 的条件下测试得到的内部参考电压 (V<sub>BGR</sub>)及其校准数据可用于计算实际的 V<sub>REF\_ADC</sub> 电压。

由以下公式可得出为器件供电的实际参考电压 V<sub>REF\_ADC</sub>：

$$V_{REF\_ADC} = 3.3V \times V_{BGR\_CAL} / V_{BGR\_DATA}$$

其中：

- V<sub>BGR\_CAL</sub> 是 V<sub>BGR</sub> 的校准值；
- V<sub>BGR\_DATA</sub> 是在当前 V<sub>REF\_ADC</sub> 电压下由 ADC 转换得到的实际 V<sub>BGR</sub> 输出值。

### 将 ADC 采集值转换为绝对电压值

根据采集到的通道输入电压与参考电压 V<sub>REF\_ADC</sub> 的比例关系，ADC 将通道输入电压转换为数字值。

对于 V<sub>REF\_ADC</sub> 电压值已知的应用，可使用以下公式计算得到该通道输入电压的绝对值：

$$V_{CHANNELX} = \frac{V_{REF\_ADC}}{FULL\_SCALE} \times ADC\_DATAx$$

对于 V<sub>REF\_ADC</sub> 电压值未知的应用，可以使用内部带隙基准电压 V<sub>BGR</sub> 计算实际的 V<sub>REF\_ADC</sub> 电压，因此 V<sub>REF\_ADC</sub> 可替换为 V<sub>BGR</sub> 的表达式，从而得出以下通道输入电压公式：

$$V_{CHANNELX} = \frac{3.3V \times V_{BGR\_CAL} \times ADC\_DATAx}{V_{BGR\_DATA} \times FULL\_SCALE}$$

其中：

- V<sub>BGR\_CAL</sub> 是 V<sub>BGR</sub> 的校准值；
- ADC\_DATAx 是在当前 V<sub>REF+</sub> 电压下由 ADC 在通道 x 上测得的值；

- $V_{BGR\_DATA}$  是在当前  $V_{REF\_ADC}$  电压下由 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。由于分辨率为 12 位，该值为  $2^{12} - 1 = 4095$ 。

## 12.4 ADC 中断

表 12-6 ADC 中断

中断事件	事件标志	使能控制位
校准结束	EOCAL	EOCALIE
采样阶段结束	EOSAMP	EOSAMPIE
通道转换结束	EOC	EOCIE
通道序列转换结束	EOS	EOSIE
模拟看门狗监控电压状态位置	AWDG	AWDGIE
数据溢出	OVRN	OVRNIE

## 12.5 ADC 寄存器

ADC 寄存器支持 32 位访问。

表 12-7 ADC 基地址

外设	基地址
ADC	0x4001 2400

### 12.5.1 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
CALEN	Res.														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											STOP	Res.	START	ADDIS	ADEN
											rs		rs	rs	rs

位/位域	名称	描述
31	CALEN	<p>ADC校准使能</p> <p>此位由软件置1，用于启动ADC校准。校准完成后，由硬件清0。</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表示正处于校准过程</p> <p><i>注意：仅当ADEN=1，ADC稳定后，且START=0、STOP=0、ADDIS=0，可将CALEN位置1。</i></p>
30:5	保留	写入无效
4	STOP	<p>ADC停止转换</p> <p>此位由软件置1，用于停止正在进行的转换，当前转换结果会丢弃。当转换已停止、并且ADC已准备好接收新的启动转换命令时，硬件将此位清0。</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转换。读取值为1表示正在停止转换</p> <p><i>注意：仅当 START=1且ADDIS=0时，可将STOP位置1。</i></p>
3	保留	写入无效

2	START	<p>ADC启动转换</p> <p>此位由软件置1，启动ADC转换。根据TRIGEN[1:0]位域的值，ADC立即开始转换（软件触发方式），或者在发生外部硬件触发事件后开始转换（外部硬件触发方式）。</p> <p>根据转换模式、转换启动触发方式配置，此位通过硬件清0，详见<a href="#">启动停止转换</a>。</p> <p>0：当前未进行ADC转换</p> <p>1：写入1可启动ADC。读取值为1表示ADC正处于转换状态</p> <p><i>注意：仅当 ADEN=1且ADDIS=0，CALEN=0时，可将 START置1。</i></p>
1	ADDIS	<p>禁止ADC</p> <p>此位由软件置1，禁止ADC并使其进入下电状态。</p> <p>ADC禁止后，硬件立即将此位清0，同时也将ADEN硬件清0。</p> <p>0：当前未执行ADDIS命令</p> <p>1：写入1可禁止ADC。读取值为1表示正在禁止ADC</p> <p><i>注意：仅当 ADEN=1且START=0，CALEN=0时，可将ADDIS置1。</i></p>
0	ADEN	<p>使能ADC</p> <p>ADEN位由软件置1，等待启动稳定时间后，ADC处于就绪状态，可进行电压转换。执行ADDIS后，硬件对此位清0。</p> <p>0：ADC未使能</p> <p>1：写入1使能ADC</p> <p><i>注意：仅当ADC_CR寄存器所有位为0时，可将ADEN位置1。</i></p>

## 12.5.2 ADC 配置寄存器 1 (ADC\_CFG1)

偏移地址：0x04

复位值：0x0000 0000

*注意：仅当START=0 时，允许对此位域执行写操作。*

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

	rw	rw	rw	rw	rw	rw							rw		
--	----	----	----	----	----	----	--	--	--	--	--	--	----	--	--

位/位域	名称	描述
31:18	保留	写入无效
17:16	TRIG_SEL[1:0]	外部硬件触发源选择 00: TIM1_TRIG_OUT 01: TIM1_OC4_ADC 10: TIM3_TRIG_OUT 11: EXTI 通道7
15	保留	写入无效
14	WAIT_MOD	自动等待模式使能 0: 禁止 1: 使能
13:12	CONV_MOD[1:0]	转换模式选择 00: 单次扫描转换模式 01: 循环扫描转换模式 10: 循环间断转换模式 11: 保留（默认选择单次扫描转换模式）
11	OV RN_MOD	数据溢出管理方式 0: 数据溢出时，ADC_DR保留上次数据，丢弃本次转换数据 1: 数据溢出时，ADC_DR覆盖上次数据
10:9	TRIGEN[1:0]	触发方式和极性选择 用于选择触发方式及外部硬件触发事件极性，并使能触发。 00: 软件触发方式 01: 外部硬件事件上升沿触发 10: 外部硬件事件下降沿触发 11: 外部硬件事件上升沿和下降沿均触发
8:3	保留	写入无效。
2	SDIR	通道序列扫描方向 0: 正向扫描（CHN0~CHN8）



1: 反向扫描 (CHN8 ~ CHN0)

1:0                      保留                      写入无效

### 12.5.3      **ADC 配置寄存器 2 (ADC\_CFG2)**

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

位/位域	名称	描述
31:27	保留	写入无效
26:24	PRESC[2:0]	ADC时钟源PCLK预分频系数 000: PCLK未分频 001: PCLK 2分频 010: PCLK 3分频 011: PCLK 4分频 100: PCLK 8分频 101: PCLK 16分频 110: PCLK 32分频 111: PCLK 64分频 注意: 仅当ADC已禁止时, 可对此位域执行写操作。
23:17	保留	写入无效
16	VBGREN	V <sub>BGR</sub> 通道输入使能 0: 禁止 1: 使能 注意: 仅当START=0 时, 可对此位域执行写操作。此位域置1后, 软件需延时启动时间 $t_{ADC\_BUF}$ 等待其稳定 ( $t_{ADC\_BUF}$ 参见数据手册)。

15:0          保留          写入无效

#### 12.5.4      **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.			AWDG	Res.		OVRN	EOS	EOC	EOSAMP	Res.
				rc_w1				rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	

位/位域	名称	描述
31:12	保留	写入无效
11	EOCAL	校准结束标志 校准完成时，此位由硬件置1。通过软件写入1将此位清0。 0: 校准未完成 1: 校准已完成
10:8	保留	写入无效
7	AWDG	ADC模拟看门狗监控电压事件标志 当转换电压超出ADC_AWDGTR寄存器中设置的阈值范围时，硬件将该位置1，通过软件写入1将此位清0。 0: 未发生监控电压事件 1: 已发生监控电压事件
6:5	保留	写入无效
4	OVRN	数据溢出标志 在EOC标志已置1时，再次发生新的转换，硬件将此位置1。通过软件写入1将此位清0。 0: 未发生数据溢出事件 1: 已发生数据溢出事件
3	EOS	通道序列转换结束标志

由ADC\_CHCFG寄存器配置的通道序列转换结束时，由硬件将此位置1。通过软件写入1将此位清0。

0: 通道序列转换未完成

1: 通道序列转换已完成

2 EOC

通道转换结束标志

当通道序列中每个通道转换结束，新数据结果出现在ADC\_DR寄存器时，由硬件将此位置1。通过软件写入1，或读取ADC\_DR寄存器都将此位清0。

0: 通道转换未完成

1: 通道转换已完成

1 EOSAMP

采样结束标志

在转换过程中，当采样阶段结束时此位由硬件置1。通过软件写入1将此位清0。

0: 采样阶段未结束

1: 采样阶段已结束

0 保留

写入无效

### 12.5.5 ADC 中断使能寄存器 (ADC\_IER)

偏移地址: 0x14

复位值: 0x0000 0000

**注意:** 仅当  $START=0$  时，允许对此寄存器执行写操作。

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.				EOCALIE	Res.			AWDGIE	Res.		OVRNIE	EOSIE	EOCIE	EOSAMP IE	Res.
				rw				rw			rw	rw	rw	rw	rw

位/位域	名称	描述
31:12	保留	写入无效
11	EOCALIE	校准结束中断使能 0: 禁止

		1: 使能, EOCAL位置1时产生中断
10:8	保留	写入无效
7	AWDGIE	模拟看门狗监控电压事件中断使能 0: 禁止 1: 使能, AWDG位置1时产生中断
6:5	保留	写入无效
4	OVRNIE	数据溢出中断使能 0: 禁止 1: 使能, OVRN位置1时产生中断
3	EOSIE	通道序列转换结束中断使能 0: 禁止 1: 使能, EOS位置1时产生中断
2	EOCIE	通道转换结束中断使能 0: 禁止 1: 使能, EOC位置1时产生中断
1	EOSAMPIE	采样结束中断使能 0: 禁止 1: 使能, EOSAMP位置1时产生中断
0	保留	写入无效

### 12.5.6 ADC 采样时间寄存器 (ADC\_SAMPT)

偏移地址: 0x18

复位值: 0x0000 0000

**注意:** 仅当  $START=0$  时, 允许对此寄存器执行写操作。

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.											SAMPT[3:0]			
											rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	写入无效
3:0	SAMPT [3:0]	采样时间参数 000x: 3 个 ADC_CK时钟周期 0010: 7 个 ADC_CK时钟周期 0011: 12 个 ADC_CK时钟周期 0100: 19 个 ADC_CK时钟周期 0101: 39 个 ADC_CK时钟周期 0110: 79 个 ADC_CK时钟周期 0111: 119 个 ADC_CK时钟周期 1000: 159 个 ADC_CK时钟周期 1001: 239 个 ADC_CK时钟周期 1010: 319 个 ADC_CK时钟周期 1011: 479 个 ADC_CK时钟周期 1100: 639 个 ADC_CK时钟周期 1101: 959 个 ADC_CK时钟周期 1110: 1279 个 ADC_CK时钟周期 1111: 1919 个 ADC_CK时钟周期

### 12.5.7 ADC 通道配置寄存器 (ADC\_CHCFG)

偏移地址: 0x1C

复位值: 0x0000 0000

**注意:** 仅当  $START=0$  时, 允许对此寄存器执行写操作。

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.							CHN8	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	写入无效

8:0 CHN[8:0] 转换通道选择

0: 未选择输入通道x进行转换

1: 已选择输入通道x进行转换

### 12.5.8 ADC 模拟看门狗配置寄存器 (ADC\_AWDGCR)

偏移地址: 0x20

复位值: 0x0000 0000

**注意:** 仅当  $START=0$  时, 允许对此寄存器执行写操作。

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.							CHN8	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	写入无效
8:0	CHN[8:0]	模拟看门狗监控通道选择 每一个寄存器位对应相应通道号。 0: 通道CHN <sub>x</sub> 不被AWDG监控 1: 通道CHN <sub>x</sub> 被AWDG监控

### 12.5.9 ADC 模拟看门狗监控电压阈值寄存器 (ADC\_AWDGTR)

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.				AWDG_HT[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.				AWDG_LT[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	写入无效
27:16	AWDG_HT[11:0]	ADC模拟看门狗监控电压阈值上限
15:12	保留	写入无效
11:0	AWDG_LT[11:0]	ADC模拟看门狗监控电压阈值下限

### 12.5.10 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										CALFACT[5:0]					
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	写入无效
5:0	CALFACT[5:0]	<p>校准系数</p> <p>此位域可由硬件或软件写入。</p> <ul style="list-style-type: none"> <li>校准完成后，会立即由硬件更新该校准系数。</li> <li>软件可向此位域写入新的校准系数。启动新的转换时，会立即应用新校准系数。</li> </ul> <p><i>注意：仅当ADEN=1且START=0时，可对这些位执行写操作。</i></p> <p><i>校准完成后，校准系数也会存储于ADC_DR寄存器的DATA[5:0]。</i></p>

### 12.5.11 ADC 数据寄存器 (ADC\_DR)

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
Res.				DATA[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	写入无效
11:0	DATA[11:0]	转换数据



## 13 比较器（COMP）

### 13.1 简介

芯片内置 2 个模拟电压比较器 COMP1 和 COMP2，可独立使用，可组成窗口比较器。比较器可用于多种功能，包括以下应用：

- 模拟信号触发从低功耗模式唤醒
- 与定时器结合使用，作为断路输入、OCREF\_CLR 事件、捕获事件输入

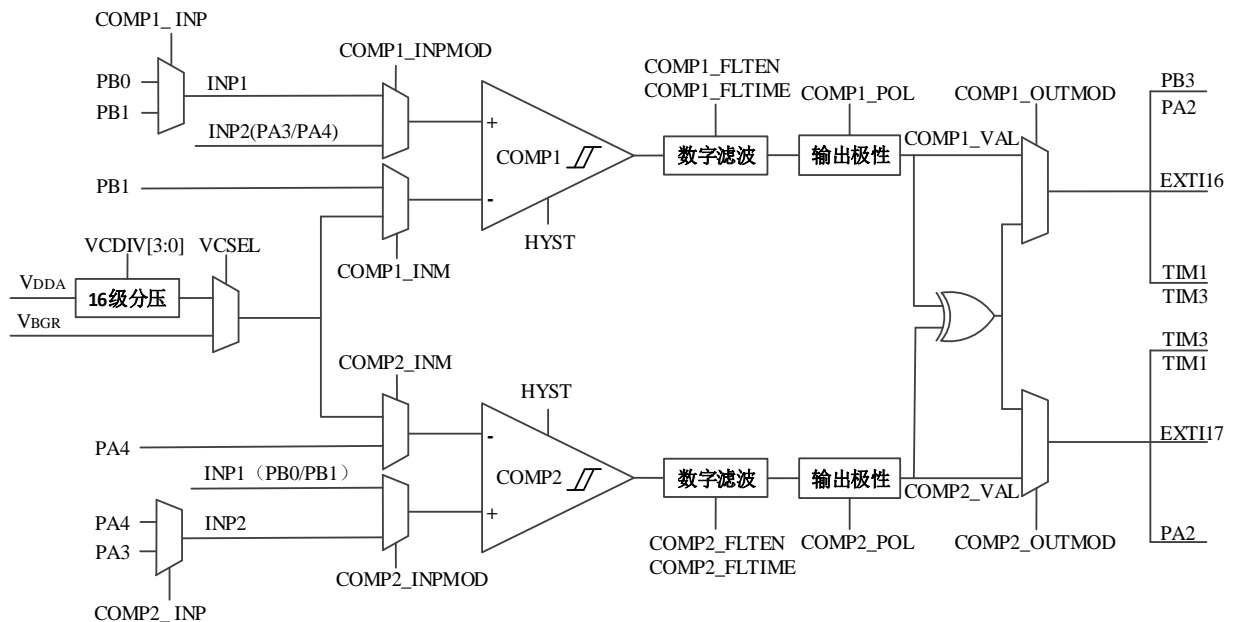
### 13.2 COMP 主要特性

- 输入电压范围  $0 \sim V_{DDA}-1.1V$
- 迟滞可配置
- 输入信号可灵活配置：
  - 具备 COMP 输入功能的 I/O
  - 内部带隙基准电压  $V_{BGR}$
  - 电源电压  $V_{DDA}$  的 16 级分压
- Stop 或 Deepstop 低功耗模式唤醒
- 输出信号可连接到 GPIO、TIM1、TIM3
- 可配置的数字滤波时间
- 窗口比较器

### 13.3 COMP 功能描述

#### 13.3.1 COMP 结构框图

图 13-1 COMP 结构框图



#### 13.3.2 引脚和内部信号

比较器输入可以选择内部参考电压信号（内部带隙基准电压  $V_{BGR}$  或  $V_{DDA}$  的 16 级分压），也可以选择 GPIO 信号，但须将其配置为模拟模式；COMP1/2 正相输入模式位 INPMOD 同时置 1 时，可交换 COMP1/2 的正相输入。

比较器输出可以连接到 GPIO，也可以在片内连接到 TIM1/3 作为输入，用于输入捕获、断路输入、OCREF\_CLR 事件等应用；可以将比较器输出同时连接到 GPIO 和片内外设。

表 13-1 COMP1 正相输入信号

COMP1 正相输入	COMP1_INP	COMP1_INPMOD	COMP2_INP
PB0	0	0	x
PB1	1	0	x
PA3	x	1	0
PA4	x	1	1

表 13-2 COMP1 反相输入信号

COMP1 反相输入	COMP1_INM
内部参考电压	0

COMP1 反相输入	COMP1_INM
PB1	1

表 13-3 COMP2 正相输入信号

COMP2 正相输入	COMP2_INP	COMP2_INPMOD	COMP1_INP
PA3	0	0	x
PA4	1	0	x
PB0	x	1	0
PB1	x	1	1

表 13-4 COMP2 反相输入信号

COMP2 反相输入	COMP2_INM
内部参考电压	0
PA4	1

### 13.3.3 使能/禁止控制

通过将 COMPx\_CSR 寄存器中 EN 位置 1 可使能 COMP，使其上电；将 EN 位清 0 可禁止 COMP，使其下电，并将输出状态 COMPx\_VAL 和输出信号 COMPx\_OUT 清零。

### 13.3.4 窗口比较器

COMP1 和 COMP2 可以组合构成窗口比较器，用于监控输入电压是否处于所设置上/下阈值范围之内。此时这两个比较器的正相输入连接到一起，用于输入待监控的模拟电压；而两个比较器的反相分别输入上/下阈值电压。

使用窗口比较器功能时，通过将其中一个 COMPx 的寄存器 INPMOD 位设置为 0，由 COMPx\_INP 位域选择正相输入信号；将另一个 COMPy 的寄存器 INPMOD 位设置为 1，使其正相输入信号选择 COMPx\_INP 对应的配置。从而使两个比较器正相输入连接，选择同一输入信号。当 COMP1/2 输入 INPMOD 寄存器位都设置 1 时，COMP1/2 正相输入互换。

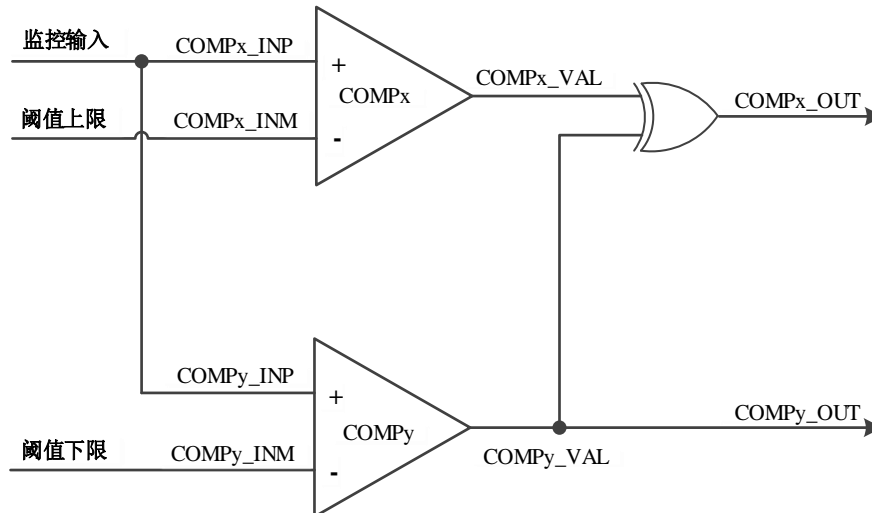
使用窗口比较器功能时，将其中一个 COMPx 的寄存器 OUTMOD 位设置为 1，从而将两个比较器的输出状态进行异或计算，此时比较器输出为 COMPx 输出状态 VAL（滤波和极性控制结果）和 COMPy 输出状态 VAL（滤波和极性控制结果）的异或结果。

比较器输出信号模式设置（OUTMOD 位）与窗口比较器不强制关联：即在窗口比较器模式下两个比较器的输出信号也可以相互独立，此时可由软件对两个比较器的输出信号进行处理；同时非窗口比较器模式下，也可以设置两个比较器

结果异或输出，由软件进行处理。

窗口比较器结构如下图：

图 13-2 窗口比较器结构<sup>(1)</sup>



1. COMPx 的正相输入模式 INPMOD=0，输出信号选择 OUTMOD=1；COMPy 的正相输入模式 INPMOD=1，输出信号选择 OUTMOD=0。

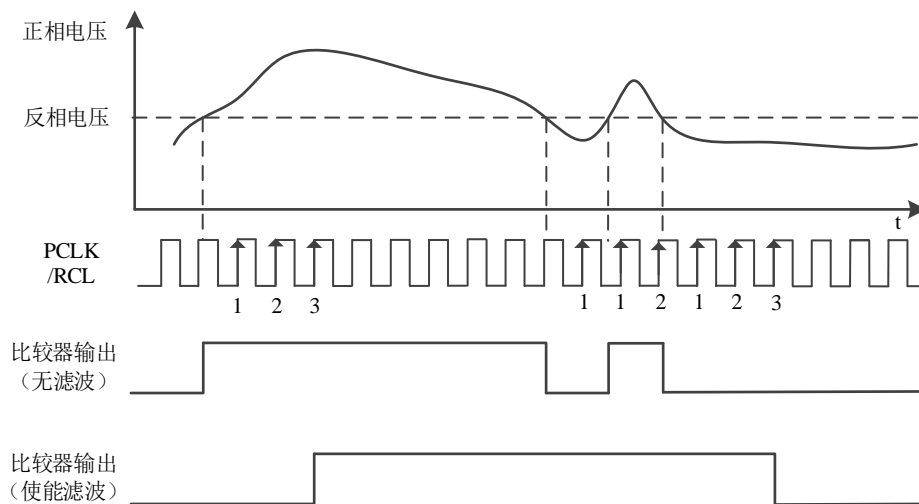
### 13.3.5 输出极性及滤波

比较器输出可进行极性选择，并可进行数字滤波。

通过寄存器 POL 位选择比较器输出极性，比较器输出状态 VAL 如下：

- 寄存器 POL 位置 0 时，比较器输出状态不反相。
  - 正相输入电压大于反相输入电压，输出状态 VAL 为 1；反之为 0；
- 寄存器 POL 位置 1 时，比较器输出状态反相。
  - 正相输入电压大于反相输入电压，输出状态 VAL 为 0；反之为 1；

比较器具有输出滤波功能，用于滤除系统噪声。将寄存器 FLTEN 位置 1 使能输出滤波功能，滤波时间由寄存器 FLTIME[2:0]位域配置。滤波时钟源通过[外设异步时钟配置寄存器（RCC\\_CLKSEL）](#)中 COMPx\_SEL 位选择 PCLK 或 RCL。比较器输出滤波示意图如下：

图 13-3 COMP 输出滤波<sup>(1)</sup>

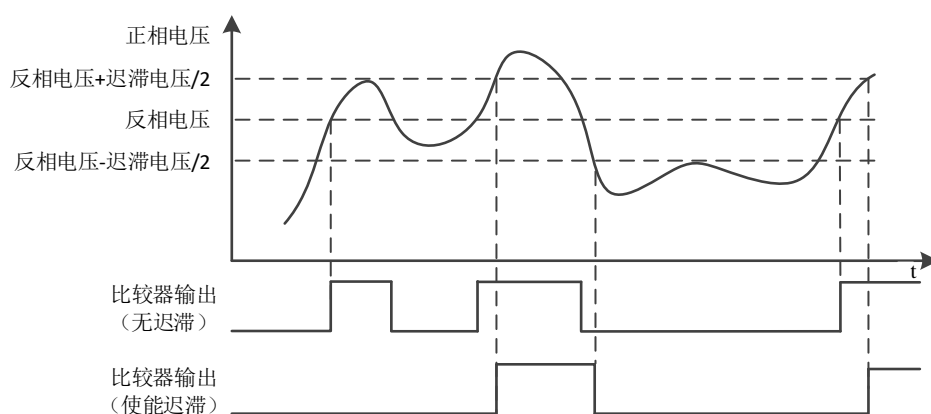
1. COMP 的输出滤波时间设置 3 个时钟周期。

**注意：** 在 *Stop* 或 *Deepstop* 低功耗模式下比较器输出滤波时钟可选择 *RCL*。

### 13.3.6 迟滞

比较器具有迟滞功能，可避免在输入信号有噪声时输出产生非预期变化。通过寄存器 *HYST* 位域可同时设置两个比较器 30mV 迟滞电压值生效。比较器迟滞功能可在不需要时禁止，以便使用外部电阻搭建迟滞电路，设置迟滞值。比较器迟滞功能示意图如下：

图 13-4 COMP 迟滞



## 13.4 COMP 中断

比较器输出可片内连接到 *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。详见[扩展中断和事件](#)

控制器 (EXTI)。

表 13-5 COMP 中断

中断事件	中断标志	中断使能	Sleep 模式唤醒	Stop/Deepstop 模式唤醒
COMP1 输出	COMP1_OUT	通过 EXTI	支持	支持
COMP2 输出	COMP2_OUT	通过 EXTI	支持	支持

## 13.5 COMP 寄存器

COMP 寄存器支持 32 位访问。

表 13-6 COMP 基地址

外设	基地址
COMP	0x4001 0200

### 13.5.1 COMP 控制寄存器 (COMP\_CR)

偏移地址：0x00

复位值：0x0000 0000

**注意：** 仅当 COMP1/2 的 EN 都为 0 时，允许对此寄存器执行写操作；

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

位/位域	名称	描述
31:5	保留	写入无效
16	HYST	比较器1和比较器2迟滞功能控制 0: 禁止 1: 使能
15:7	保留	写入无效
6	VCSEL	内部参考电压源选择 0: $V_{BGR}$ 1: $V_{DDA}$ 的16级分压 <b>注意：</b> 不使用 $V_{DDA}$ 的16级分压时，保持VSEL位为0，可避免分压电路产生的功耗。
5:4	保留	写入无效

3:0            VCDIV[3:0]             $V_{DDA}$  分压选择

0000: 1/16

0001: 2/16

0010: 3/16

0011: 4/16

⋮

1100: 13/16

其他: 保留 (默认 $V_{DDA}$ 的13/16分压)

### 13.5.2 COMP1 控制和状态寄存器 (COMP1\_CSR)

偏移地址: 0x10

复位值: 0x0000 0000

**注意:** 仅当  $EN=0$  时, 允许对此寄存器执行写操作 ( $EN$  位除外)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	VAL	Res.	FLTEN	FLTIME[2:0]			Res.								
	r		rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	OUTMOD	INPMOD	Res.				INP	Res.			INM	Res.			EN
rw	rw	rw					rw				rw				rw

位/位域	名称	描述
31	保留	写入无效
30	VAL	比较器1输出状态 表示比较器滤波及极性控制后的输出结果
29	保留	写入无效
28	FLTEN	比较器1输出滤波使能 0: 禁止 1: 使能
27:25	FLTIME[2:0]	比较器1输出滤波时钟周期选择 000: 1个时钟周期 001: 3个时钟周期 010: 7个时钟周期



		011: 15个时钟周期 100: 31个时钟周期 101: 63个时钟周期 110: 255个时钟周期 111: 1023个时钟周期
24:16	保留	写入无效
15	POL	比较器1输出极性选择 0: 输出不反相 1: 输出反相
14	OUTMOD	比较器1输出信号 (COMP1_OUT) 选择 0: 比较器1输出状态 (COMP1_VAL) 1: 比较器1输出状态 (COMP1_VAL) 和比较器2输出状态 (COMP2_VAL) 的异或结果
13	INPMOD	比较器1正相输入信号选择, 用于配置窗口比较器模式 0: 比较器1正相输入信号由INP位决定 1: 比较器1正相输入信号由比较器2的INP决定
12:9	保留	写入无效
8	INP	比较器1正相输入信号选择 0: PB0 1: PB1
7:5	保留	写入无效
4	INM	比较器1反相输入信号选择 0: 内部参考电压 1: PB1
3:1	保留	写入无效
0	EN	比较器1使能 0: 禁止 1: 使能

### 13.5.3 COMP2 控制和状态寄存器 (COMP2\_CSR)

偏移地址：0x14

复位值：0x0000 0000

**注意：** 仅当  $EN=0$  时，允许对此寄存器执行写操作 ( $EN$  位除外)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	VAL	Res.	FLTEN	FLTIME[2:0]			Res.								
	r		rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	OUTMOD	INPMOD	Res.				INP	Res.			INM	Res.			EN
rw	rw	rw					rw				rw				rw

位/位域	名称	描述
31	保留	写入无效
30	VAL	比较器2输出状态 表示比较器滤波及极性控制后的输出结果
29	保留	写入无效
28	FLTEN	比较器2输出滤波使能 0：禁止 1：使能
27:25	FLTIME[2:0]	比较器2输出滤波时钟周期选择 000：1个时钟周期 001：3个时钟周期 010：7个时钟周期 011：15个时钟周期 100：31个时钟周期 101：63个时钟周期 110：255个时钟周期 111：1023个时钟周期
24:16	保留	写入无效
15	POL	比较器2输出极性选择

		0: 输出不反相 1: 输出反相
14	OUTMOD	比较器2输出信号 (COMP2_OUT) 选择 0: 比较器2输出状态 (COMP2_VAL) 1: 比较器2输出状态 (COMP2_VAL) 和比较器1输出状态 (COMP1_VAL) 的异或结果
13	INPMOD	比较器2正相输入模式选择, 用于配置窗口比较器模式 0: 比较器2正相输入信号由INP位决定 1: 比较器2正相输入信号由比较器1的INP决定
12:9	保留	写入无效
8	INP	比较器2正相输入信号选择 0: PA3 1: PA4
7:5	保留	写入无效
4	INM	比较器2反相输入信号选择 0: 内部参考电压 1: PA4
3:1	保留	写入无效
0	EN	比较器2使能 0: 禁止 1: 使能

## 14 高级控制定时器（TIM1）

### 14.1 简介

高级控制定时器 TIM1 由一个 16 位自动重载计数器构成。可用于生成输出波形（比较输出、单脉冲输出、PWM 和带死区插入的互补 PWM）等多种用途。

高级控制定时器 TIM1 和通用定时器结合使用，可实现定时器互联功能。

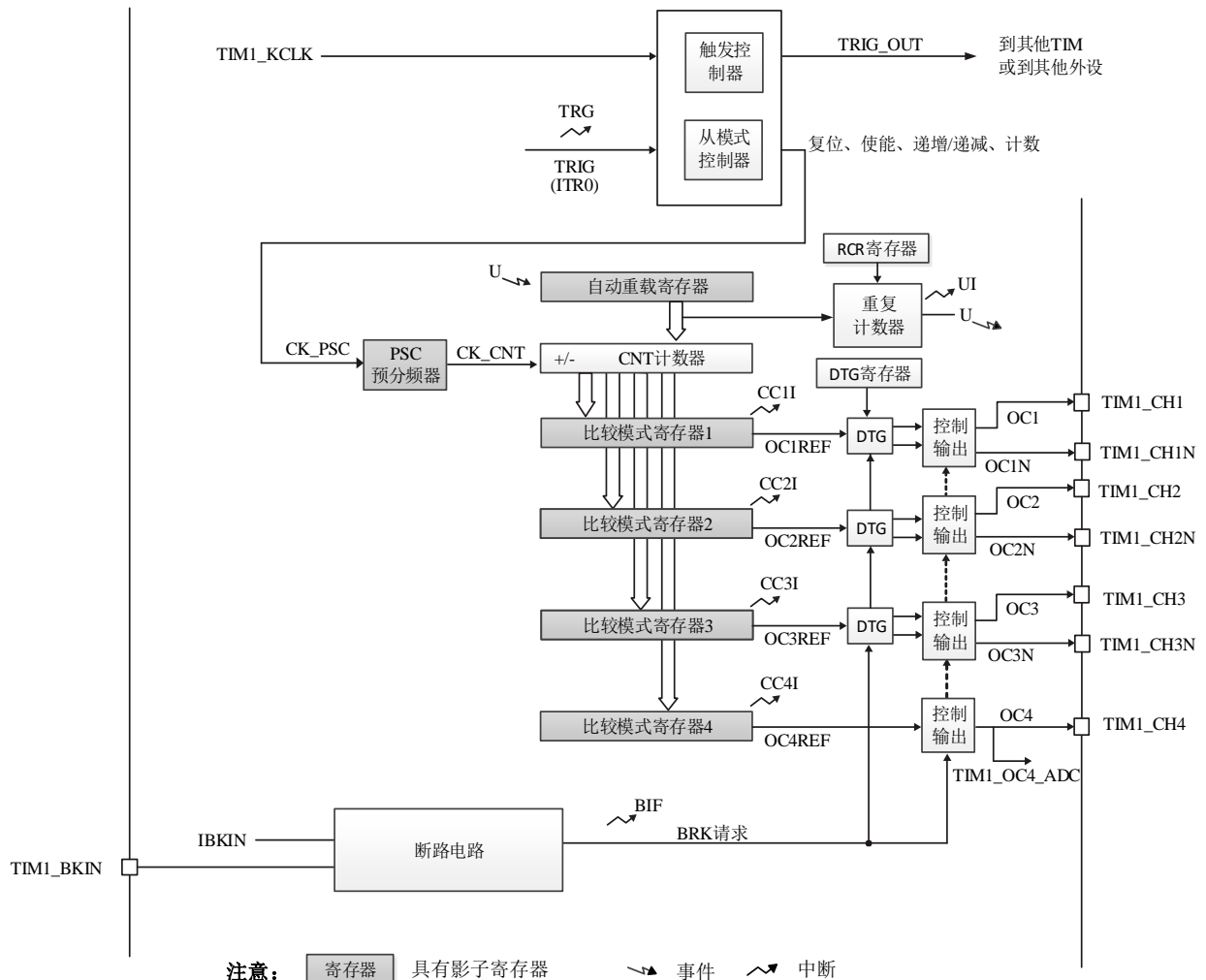
### 14.2 TIM1 主要特性

- 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
- 8 位重复计数器
- 4 个独立通道，可用于：
  - 输出比较
  - PWM 输出
  - 单脉冲模式输出
- 1 个断路输入：
  - 具备复用功能的 GPIO
  - COMP1、COMP2 输出
  - 系统断路（CPU LOCKUP 事件）
- 可编程死区的互补输出
- 可实现定时器的启动、停止、初始化
- 可定时触发 ADC、可多个定时器互联
- 支持霍尔传感器

## 14.3 TIM1 功能描述

### 14.3.1 TIM1 框图

图 14-1 TIM1 框图



### 14.3.2 TIM1 引脚和内部信号

表 14-1 TIM1 引脚

引脚名称	信号类型	说明
TIM1_CH1	输出	通道 1 输出引脚
TIM1_CH1N	输出	通道 1 的互补输出引脚
TIM1_CH2	输出	通道 2 输出引脚
TIM1_CH2N	输出	通道 2 的互补输出引脚
TIM1_CH3	输出	通道 3 输出引脚
TIM1_CH3N	输出	通道 3 的互补输出引脚
TIM1_CH4	输出	通道 4 输出引脚

引脚名称	信号类型	说明
TIM1_BKIN	输入	断路输入信号

表 14-2 TIM1 内部信号

信号名称	信号类型	说明
TIM1_KCLK	输入	TIM1 外设时钟，来自 RCC
ITR	输入	内部触发信号
IBKIN	输入	内部断路输入信号
TIM1_OC4_ADC	输出	通道 4 比较输出信号，连接到 ADC
TRIG_OUT	输出	内部触发输出信号

### 14.3.3 PSC 预分频器

PSC 预分频器对 CNT 计数器的输入时钟进行分频，分频系数介于 1 和 65536 之间。TIM1\_PSC 寄存器中的 PSC[15:0]位域用于配置分频值，实际分频系数为分频值 PSC[15:0]+1。

TIM1\_PSC 寄存器具有缓冲功能（影子寄存器），每次写入的预分频值，会在下一次更新事件发生时生效。

以下各图，举例显示了预分频器更新生效的时序。

图 14-2 预分频器分频系数由 1 变为 2 时生效的时序图

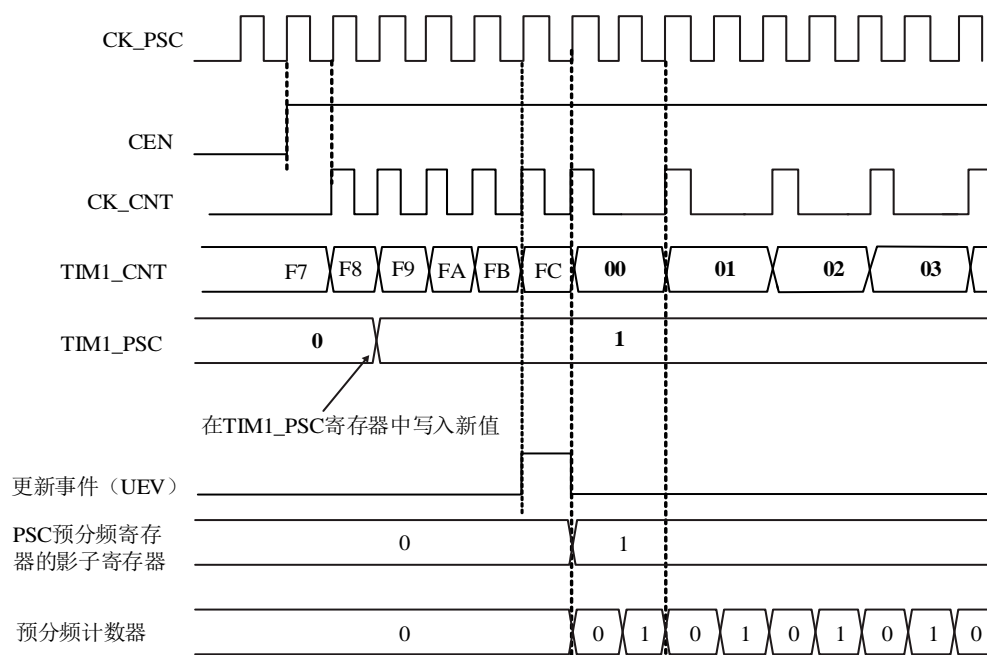
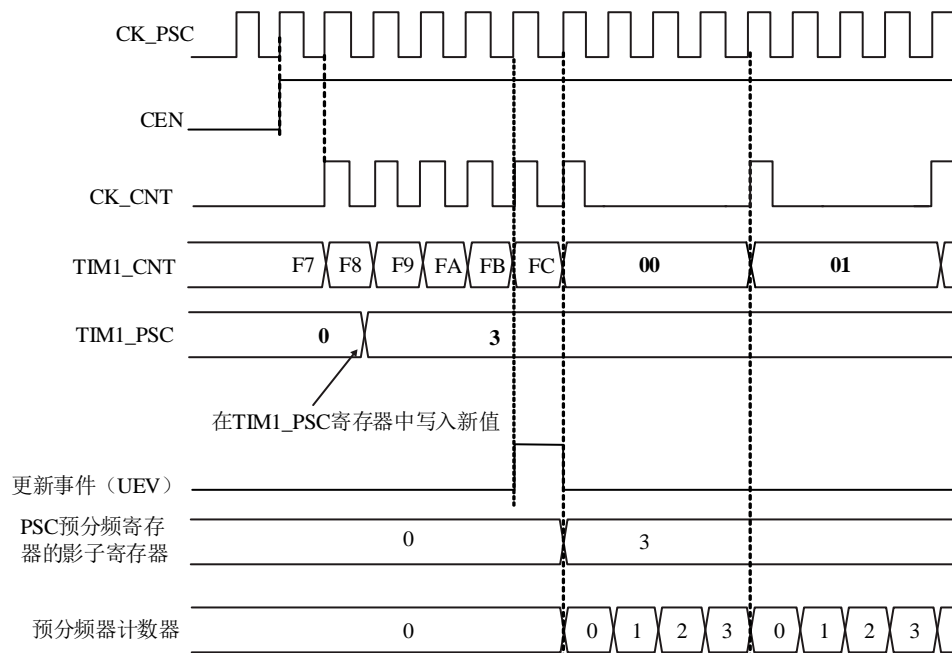


图 14-3 预分频器分频系数由 1 变为 4 时的计数器时序图



#### 14.3.4 计数器

TIM1 内置一个 16 位的计数器，可选择递增计数模式、递减计数模式或中心对齐计数模式。

计数器配置支持动态修改，在计数器运行时，TIM1\_CNT 寄存器、TIM1\_ARR 寄存器和 TIM1\_PSC 寄存器均可进行读写操作。

自动重载值寄存器 (TIM1\_ARR) 中的 ARR[15:0] 位域可用于配置计数器的重载值，该寄存器具有缓冲功能 (影子寄存器)。

如果 TIM1\_CR1 寄存器中的 ARPE 位为 0 时，影子寄存器的更新不受更新事件的影响，TIM1\_ARR 寄存器的值会立即更新。如果 ARPE 位为 1 时，当发生更新事件，则 TIM1\_ARR 寄存器中的值将更新到影子寄存器中。

如果将 TIM1\_CR1 寄存器中的 UDIS 位置 1，可禁止更新 (UEV) 事件，避免对影子寄存器的更新。

当 UDIS 位为 0 时，使能更新事件，此时 URS 位配置为 1，则将 UG 位置 1，会生成更新事件，计数器及预分频计数器均被初始化，但不会将 TIM1\_SR 寄存器中的 UIF 置位。

通过 URS 可以选择更新事件的触发源：

- 0：使能时，以下事件都会产生更新事件。
  - 计数器上溢/下溢 (RCR 设置为 0)；

- RCR 递减到 0，下一个上溢或下溢（RCR 设置为非 0）；
  - 将 TIM1\_EVTG 寄存器中的 UG 位置 1；
  - 通过从模式控制器生成的更新事件（复位模式）。
- 1：使能时，以下事件都会产生更新事件。
    - 计数器上溢/下溢（RCR 设置为 0）；
    - RCR 递减到 0，下一个上溢或下溢（RCR 设置为非 0）。

### 递增计数模式

在递增计数模式下，计数器从 0 开始递增计数，每个 CK\_CNT 周期自动加 1，当计数值等于 TIM1\_ARR 寄存器的值时，将产生计数器上溢事件，TIM1\_SR 寄存器的 UIF 标志将自动置 1，同时计数器将被清 0 并重新开始计数。

如果使用重复计数器，在定时器发生上溢事件时，仅递减重复计数器的值，只有当递增计数的重复次数达到其设定次数加 1 次（TIM1\_RCR+1）后，才会生成更新事件。

发生更新事件时，TIM1\_SR 寄存器中的 UIF 位将自动置 1，并对以下寄存器进行更新：

- TIM1\_RCR 寄存器的值更新到重复计数器中；
- 预装载值（TIM1\_ARR 寄存器）更新到自动重载影子寄存器中；
- 预装载值（TIM1\_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中。

以下各图，显示了递增计数模式下，计数器发生上溢更新事件时的时序，TIM1\_ARR 寄存器配置为 0x36：



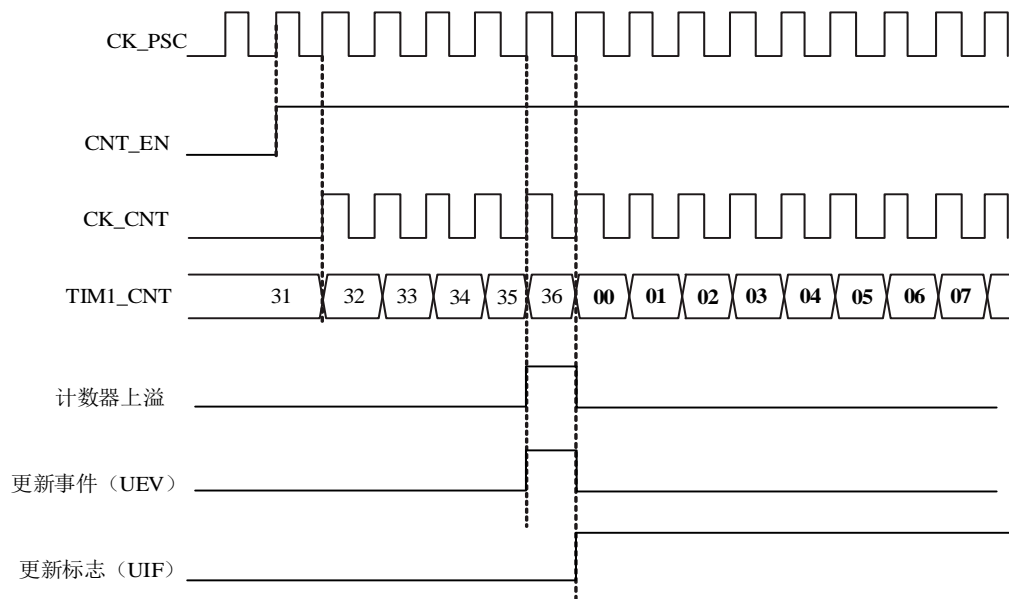
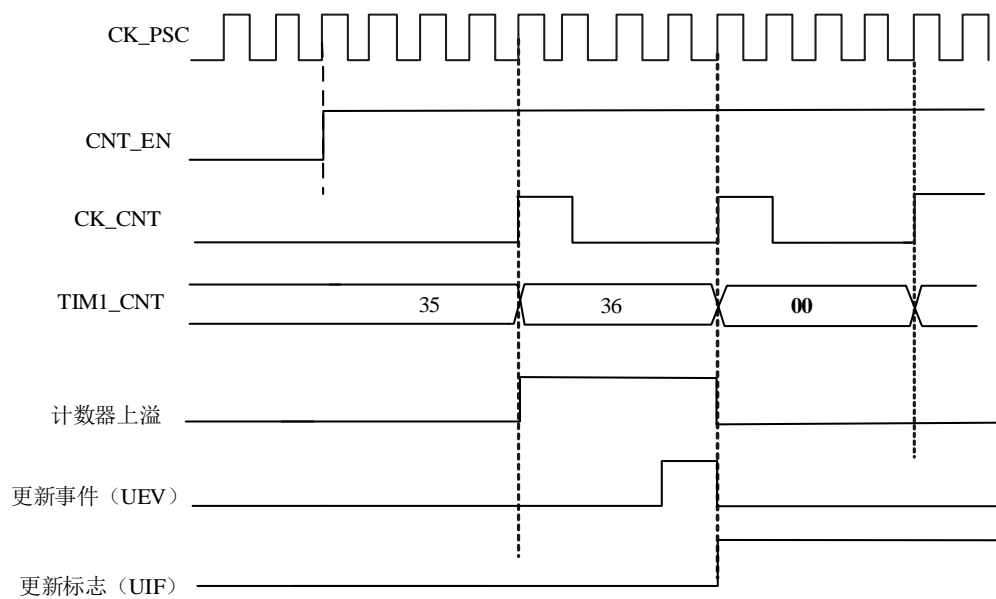
**图 14-4 递增计数模式下配置预分频器为 1 分频上溢事件时序图**

**图 14-5 递增计数模式下配置预分频器为 4 分频上溢事件时序图**


图 14-6 递增计数模式下计数器时序图，ARPE=0 时更新事件

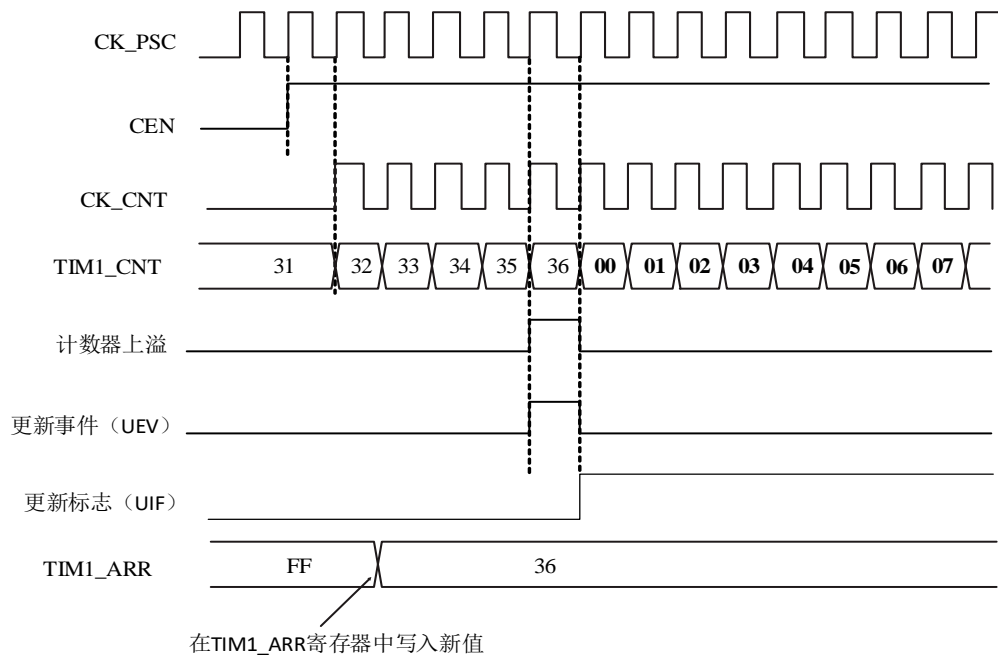
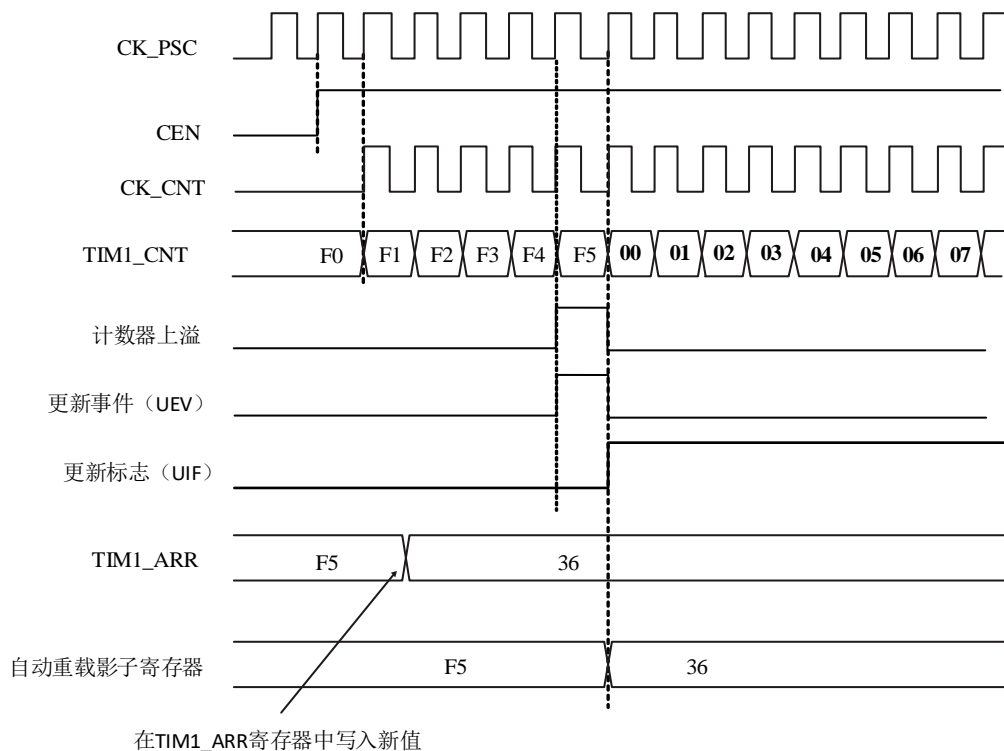


图 14-7 递增计数模式下计数器时序图，ARPE=1 时更新事件



### 递减计数模式

在递减计数模式下，计数器从 TIM1\_ARR 寄存器的值开始递减计数，每个 CK\_CNT 周期自动减 1，当计数器等于 0 时，将产生计数器下溢事件，TIM1\_SR 寄存器的 UIF 标志将自动置 1，同时计数器加载自动重载值，并重新开始递减计数。

如果使用重复计数器，在定时器发生下溢事件时，仅递减重复计数器的值，只有递减计数的重复次数达到其次数加 1 次 (TIM1\_RCR+1) 后，才会产生更新事件。

发生更新事件时，TIM1\_SR 寄存器中的 UIF 位将自动置 1，并对以下寄存器进行更新：

- TIM1\_RCR 寄存器的值更新到重复计数器中；
- 预装载值 (TIM1\_PSC 寄存器) 重新加载到 PSC 预分频器的影子寄存器中；
- 预装载值 (TIM1\_ARR 寄存器) 更新到自动重载影子寄存器中。TIM1\_ARR 寄存器的值在计数器重载之前被更新，因此下一个周期才是预期的值。

以下各图，显示了计数器发生下溢更新事件时的时序，TIM1\_ARR 寄存器配置为 0x36：

图 14-8 递减计数模式下配置预分频器为 1 分频下溢事件时序图

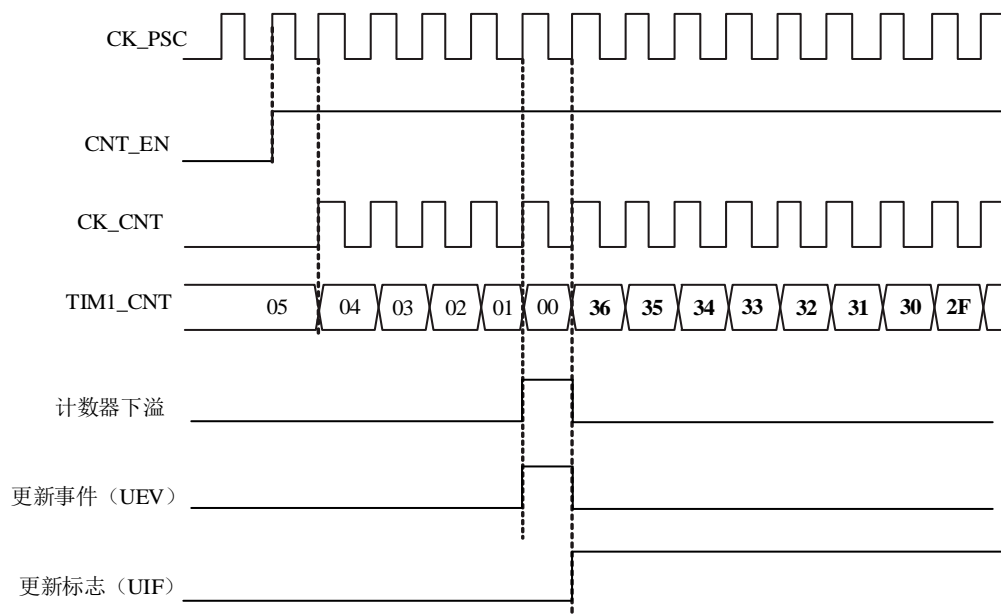


图 14-9 递减计数模式下配置预分频器为 4 分频下溢事件时序图

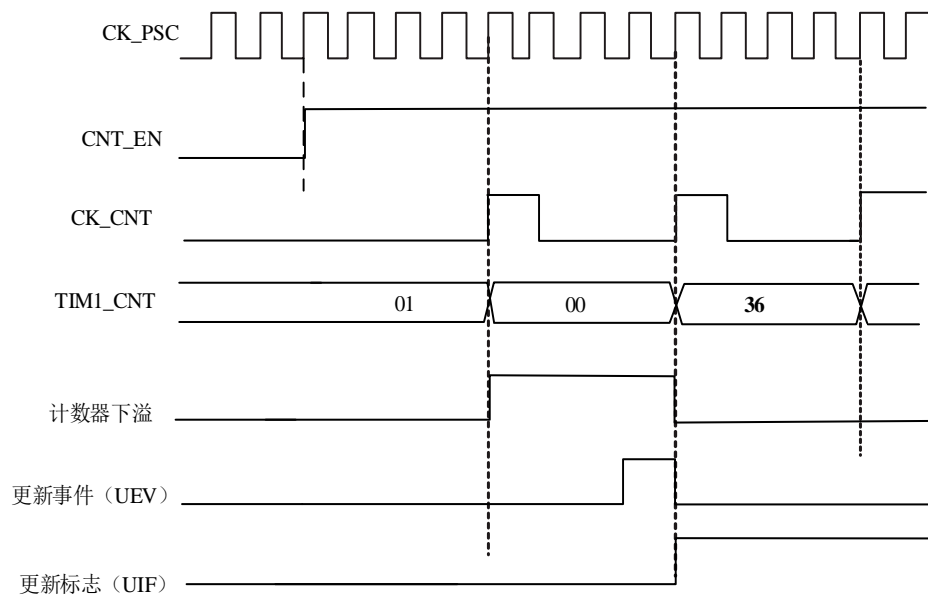
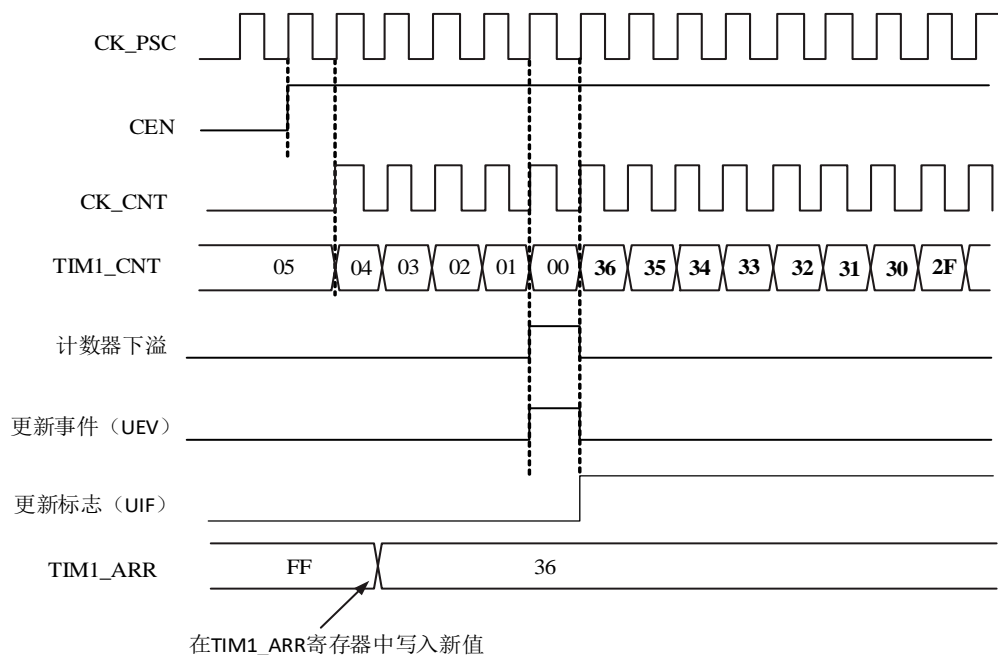


图 14-10 递减计数模式下计数器时序图，ARPE=0 时更新事件



### 中心对齐模式（递增/递减计数）

在中心对齐模式下，计数过程由以下阶段组成：

- 1) 计数器从 TIM1\_CNT 的值开始递增计数到 TIM1\_ARR-1，生成计数器上溢事件；
- 2) 从 TIM1\_ARR 的值开始递减计数到 1，生成计数器下溢事件；
- 3) 从 0 开始递增计数。

当 TIM1\_CR1 寄存器中的 DIR 位为 1 时，启动计数后，计数器从当前 TIM1\_CNT

的值开始递减计数。

启动计数后，无论当前是递增还是递减计数，当 UG 位置 1 时，TIM1\_CNT 寄存器的值和 TIM1\_CR1 寄存器中的 DIR 均变为 0。

当 TIM1\_CR1 寄存器中的 CMS[1:0]位域不为 00 时，计数器将采用中心对齐模式，在此模式下，TIM1\_CR1 寄存器中的 DIR 位不能写入，读取 DIR 位可获取当前计数器的计数方向。中心对齐模式共有以下几种计数模式：

- 中心对齐模式 1（CMS[1:0]位域为 01）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIM1\_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 2（CMS[1:0]位域为 10）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIM1\_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 3（CMS[1:0]位域为 11）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIM1\_SR 寄存器中的 CCxIF 位会置 1。

发生更新事件时，TIM1\_SR 寄存器中的 UIF 位将自动置 1，并对以下寄存器进行更新：

- TIM1\_RCR 寄存器的值更新到重复计数器中；
- 预装载值(TIM1\_PSC 寄存器)重新加载到 PSC 预分频器的影子寄存器中；
- 预装载值（TIM1\_ARR 寄存器）更新到自动重载影子寄存器中。

以下各图，显示了中心对齐模式下，计数器发生上/下溢更新事件时的时序，TIM1\_ARR 寄存器配置为 0x06：

图 14-11 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图

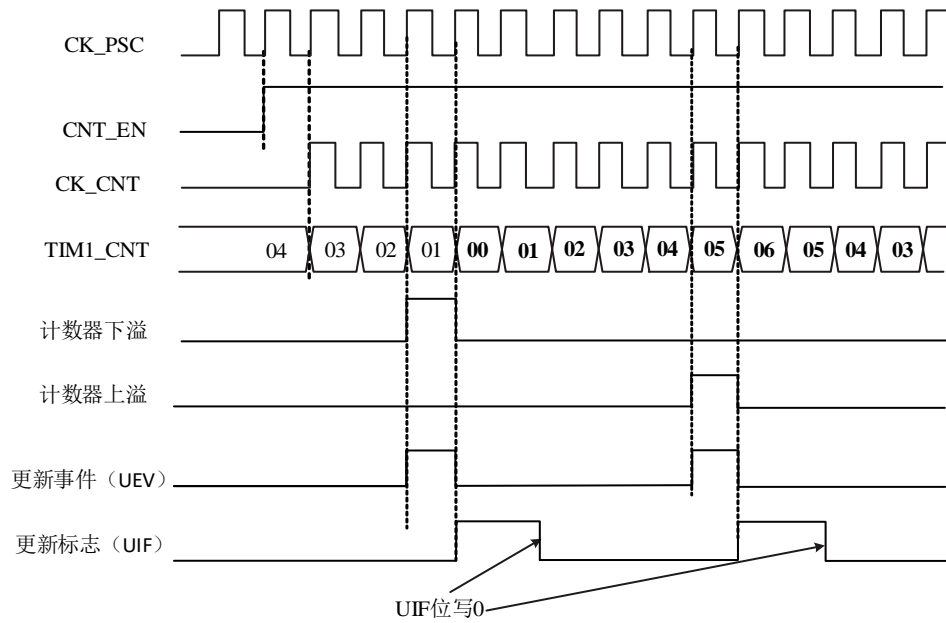


图 14-12 中心对齐模式下配置预分频器为 2 分频下溢事件时序图

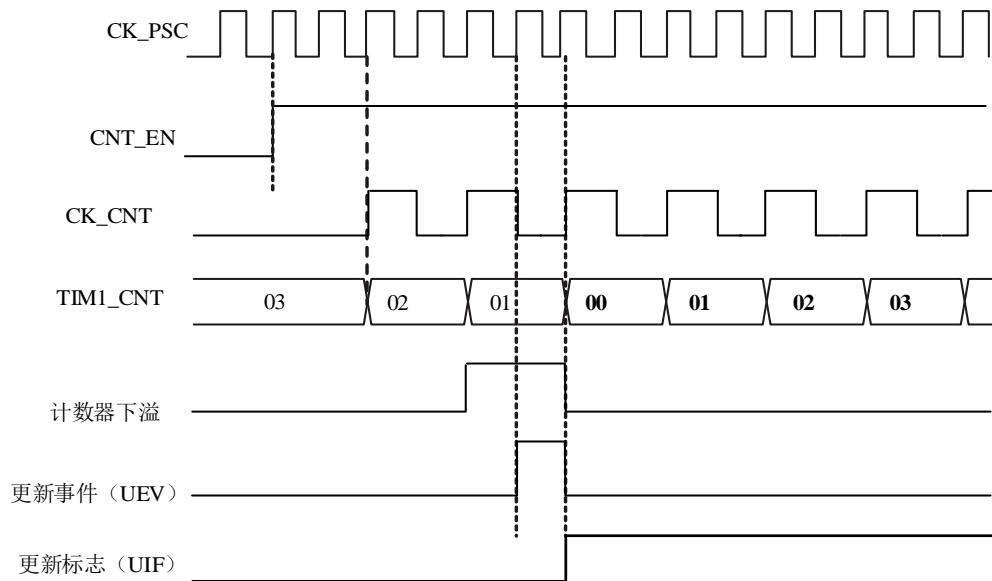


图 14-13 中心对齐模式下，产生下溢事件，ARPE=1 时的时序图

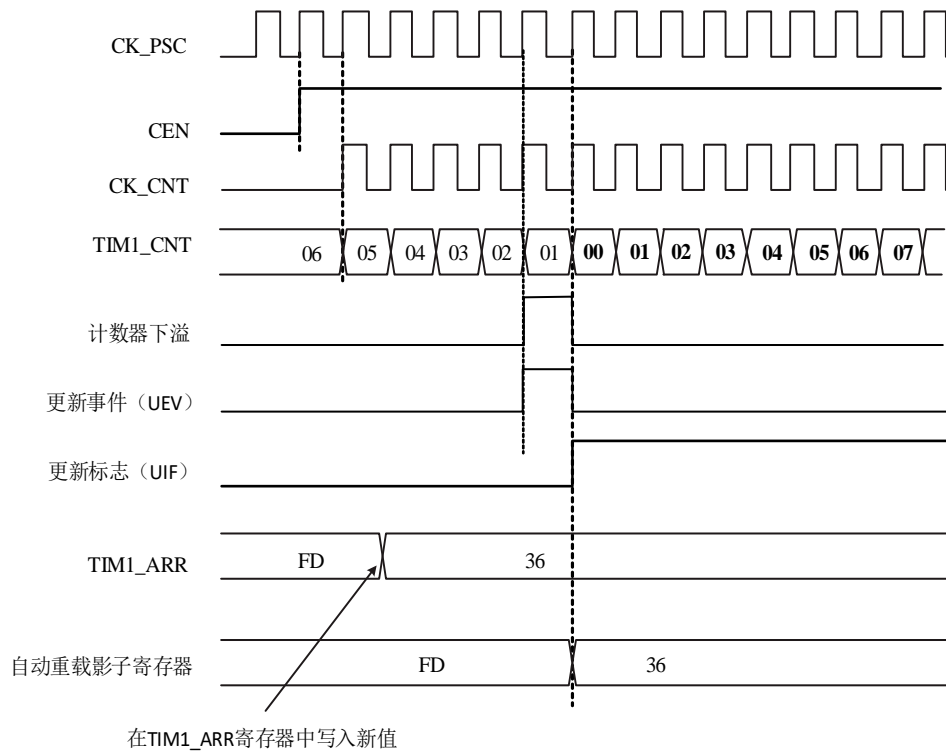
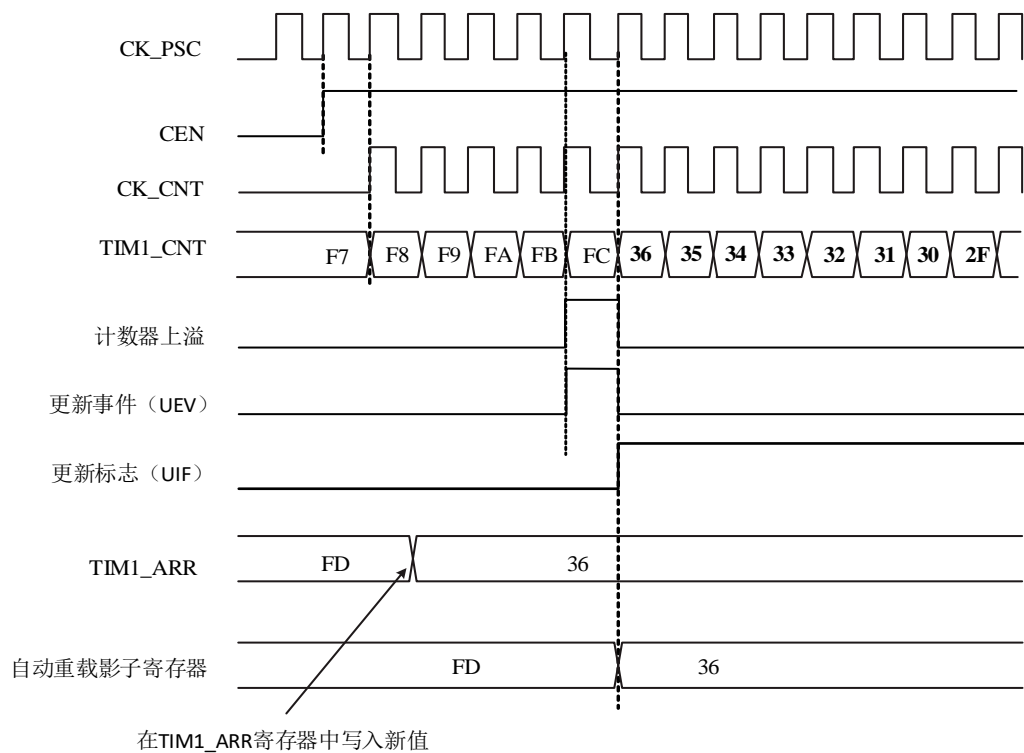


图 14-14 中心对齐模式下，产生上溢事件，ARPE=1 时的时序图



建议使用中心对齐模式时，在启动计数器前置位 TIM1\_EVTG 寄存器中的 UG 位，产生一个更新事件，并且不要在计数器计数期间修改计数器的值。

## 14.3.5 重复计数器

重复计数器由8位递减计数器构成,计数值可通过TIM1\_RCR寄存器的REP[7:0]位域进行配置。

- REP 等于 0: 更新事件在计数器上/下溢时产生;
- REP 不等于 0: 当重复计数器递减到 0, 下一次上溢或下溢产生时, 触发更新事件。

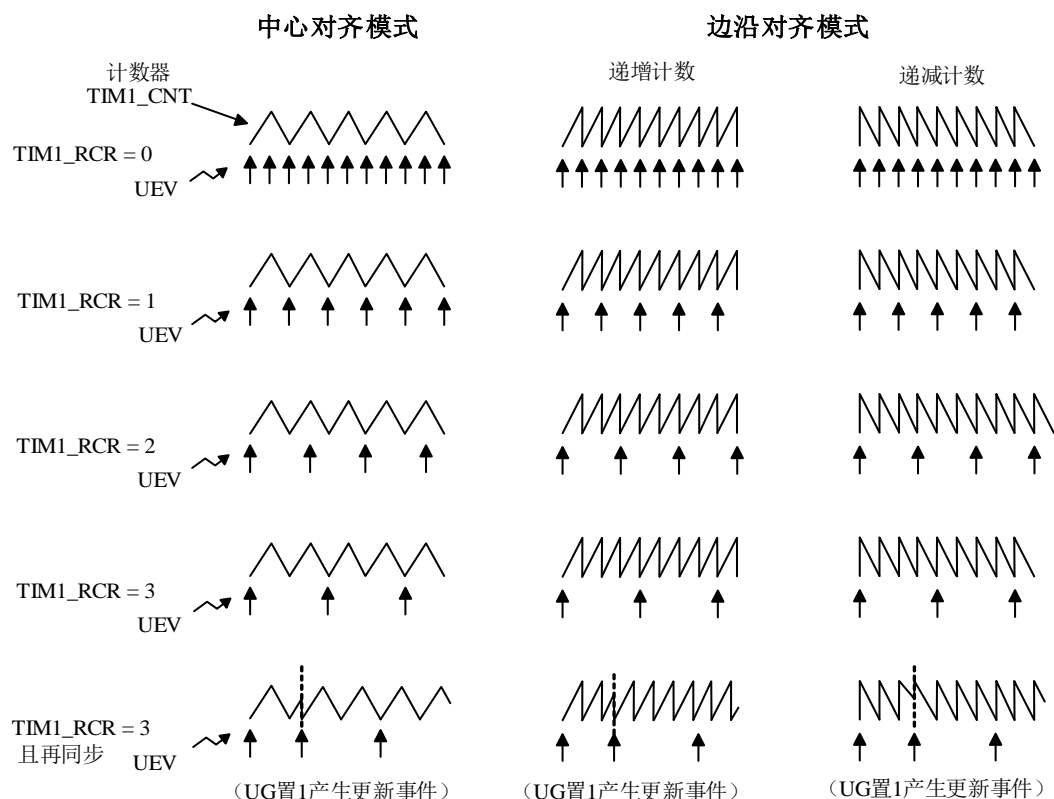
重复计数器在下列情况下自动递减:

- 递增计数模式下的每次计数器上溢;
- 递减计数模式下的每次计数器下溢;
- 中心对齐模式下每次计数器上溢和计数器下溢。

由软件 (通过将 TIM1\_EVTG 寄存器的 UG 位置 1)、硬件 (通过从模式控制器)、计数上/下溢生成更新事件时, 重复计数器会自动加载 TIM1\_RCR 寄存器的值。

下图显示不同模式下, 设置不同 TIM1\_RCR 寄存器值时更新事件的频率示例。

图 14-15 不同模式下设置不同 TIM1\_RCR 寄存器值时的更新频率示例





### 14.3.6 计数时钟选择

计数器的计数时钟可由下列信号源提供：

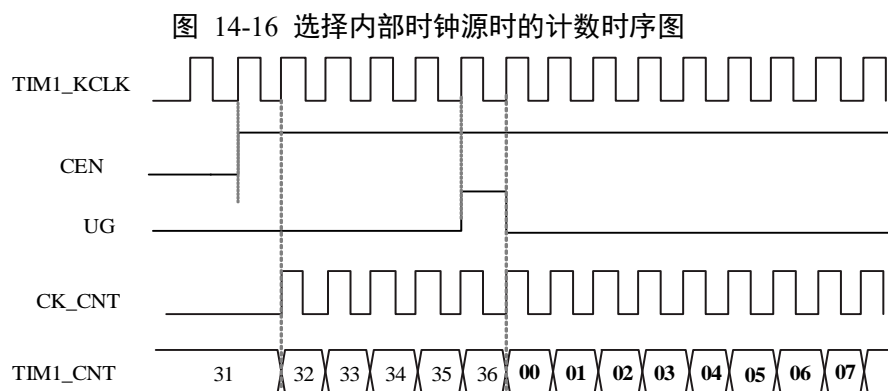
- 内部时钟源：TIM1 外设时钟（TIM1\_KCLK）；
- 时钟模式 1：TRIG 触发输入作为计数时钟。

以上时钟源可通过配置 TIM1\_PSC 寄存器的值进行预分频后，作为计数时钟（CK\_CNT）。

#### 内部时钟源

当配置 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域为 0xx 时，计数器的计数时钟源为 TIM1\_KCLK。

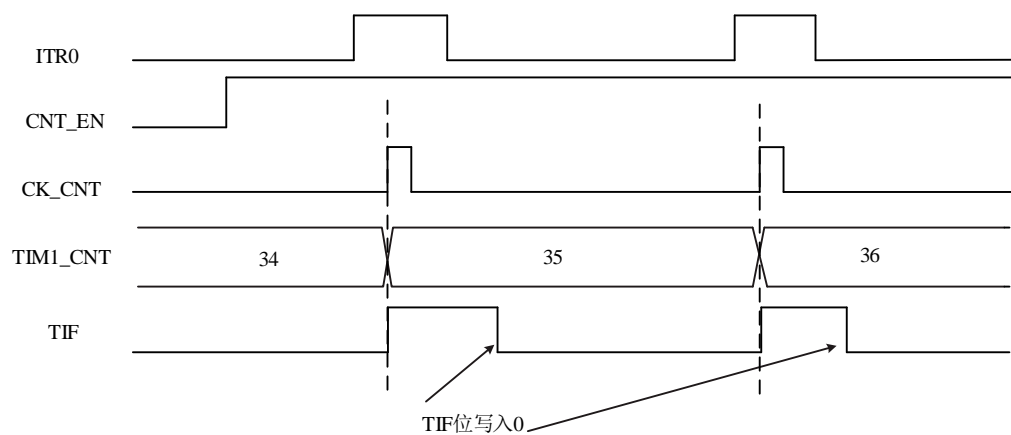
下图显示了选择内部时钟源时的计数时序图。



#### 时钟模式 1

当配置 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域为 111 时，计数器的计数时钟源为 TRIG 触发输入信号，即 ITR0 输入信号。

图 14-17 选择时钟模式 1 时的时序图

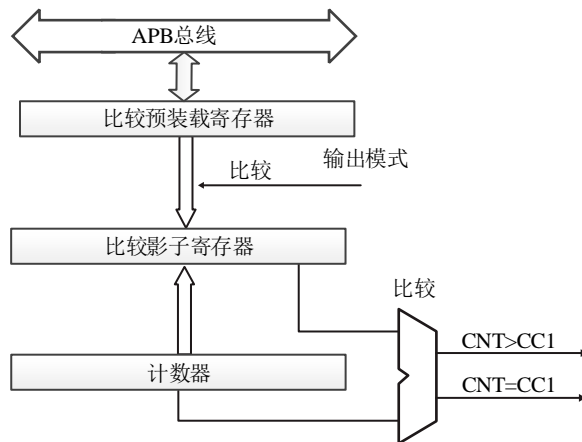


### 14.3.7 比较通道

比较通道包含：

- 比较寄存器（包括一个影子寄存器）
- 输出比较通道（比较器和输出控制）

图 14-18 比较通道



在比较模式下，比较寄存器的内容将加载到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

#### 输出比较模式

输出通道 1 到通道 4 可输出到芯片外部，其中输出通道 1 到通道 3 支持互补输出，OCx 和 OCxN 为互补输出信号。

输出比较模式分为：

- 比较输出模式，详见 [比较输出模式](#)；
- 强制输出模式，详见 [强制输出模式](#)；
- PWM 模式，详见 [PWM 模式](#)；
- 单脉冲模式，详见 [单脉冲模式](#)。

图 14-19 比较通道的输出连接图（通道 1、通道 2 和通道 3）

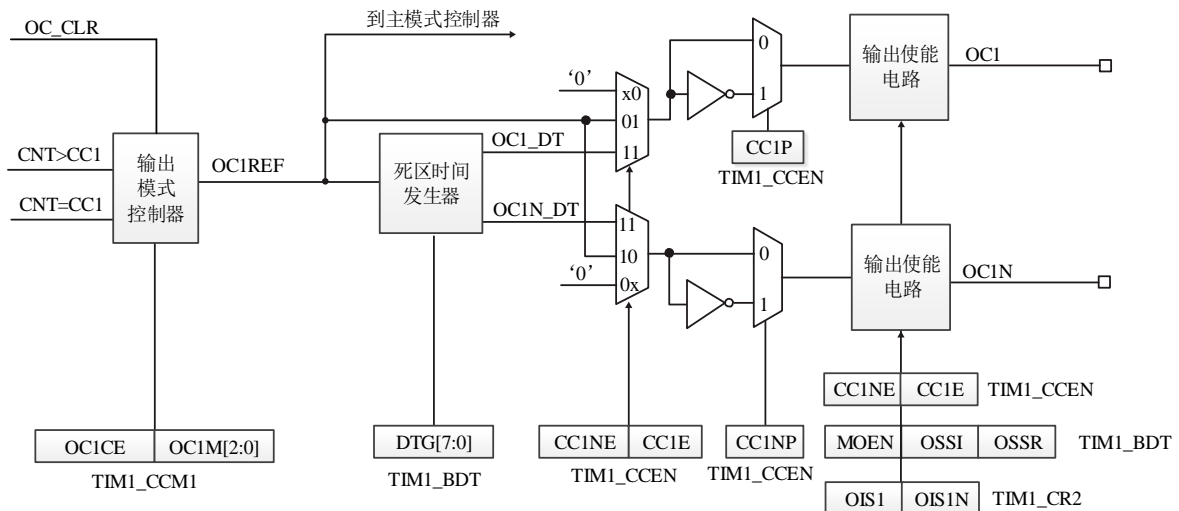
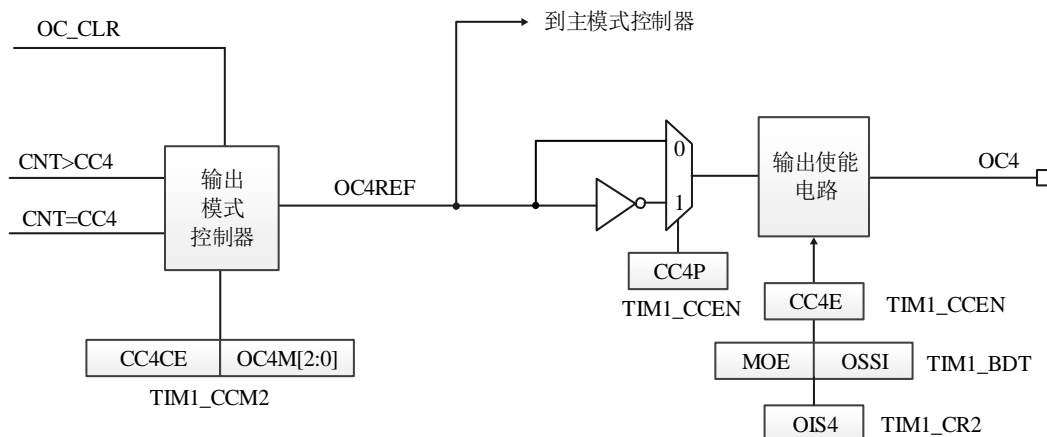


图 14-20 比较通道的输出连接图（通道 4）



### 14.3.8 比较输出模式

该模式包含：匹配输出有效电平、匹配输出无效电平和翻转模式。通过  $TIM1\_CCMx$  寄存器中的  $OCxPE$  位，可使能或禁止  $TIM1\_CCx$  寄存器的预装载功能。

当  $TIM1\_CCx$  的影子寄存器与  $TIM1\_CNT$  寄存器的值匹配时：

- $TIM1\_SR$  寄存器中的  $CCxIF$  位置 1；
- 输出极性由  $TIM1\_CCEN$  寄存器中的  $CCxP$  位决定；
- 如果此时  $TIM1\_DIER$  寄存器中的  $CCxIE$  位为 1，则触发中断；

配置为比较输出模式时，其步骤如下：

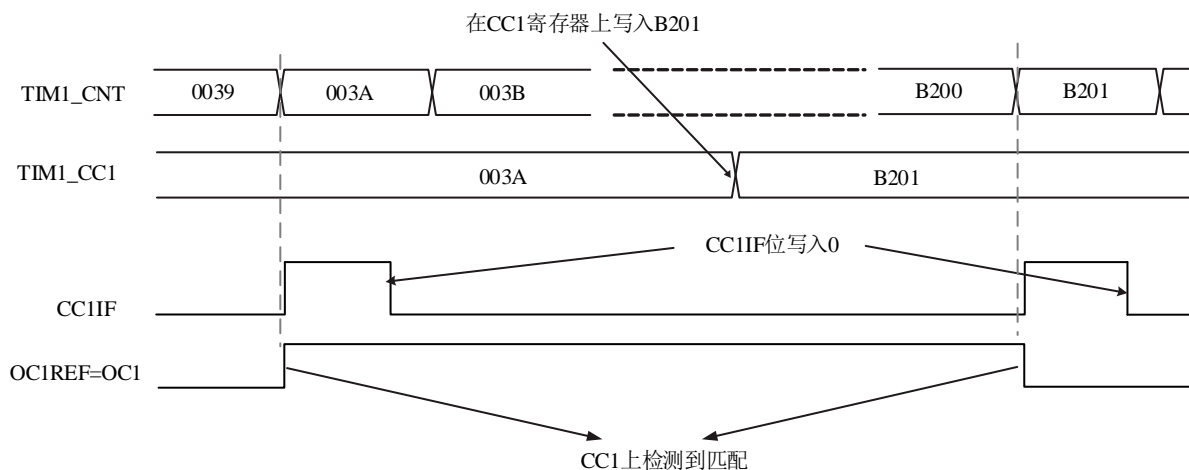
- 1) 配置  $TIM1\_ARR$  寄存器和  $TIM1\_CCx$  寄存器；
- 2) 选择输出模式。例如：

- 配置 TIM1\_CCMx 寄存器中的 OCxM[2:0]位域为 011（当 TIM1\_CNT 寄存器的值与 TIM1\_CCx 寄存器的值匹配时，OCx 输出发生翻转）；
- 禁止预装载寄存器：配置 TIM1\_CCMx 寄存器中的 OCxPE 位为 0；
- 选择高电平有效：配置 TIM1\_CCEN 寄存器中的 CCxP 为 0；
- 使能输出：配置 TIM1\_CCEN 寄存器中的 CCxE 为 1；

3) 使能计数器：配置 TIM1\_CR1 寄存器中的 CEN 位为 1。

当未使能预装载寄存器（TIM1\_CCMx 寄存器中的 OCxPE 位为 0）时，可实时更改 TIM1\_CCx 寄存器的值，以控制输出波形。

图 14-21 比较输出模式，翻转 OC1（OCxPE 位为 0）



### 14.3.9 强制输出模式

在强制输出模式下，可配置输出信号为有效或无效电平，此时输出信号不受 TIM1\_CCx 影子寄存器和计数器之间的比较结果影响。

- 配置 TIM1\_CCMx 寄存器中的 OCxM[2:0] 位域为 101，可将输出信号（OCxREF）强置为高电平（OCxREF 始终为高电平有效）；
- 配置 TIM1\_CCMx 寄存器中的 OCxM[2:0] 位域为 100，可将输出信号（OCxREF）强置为低电平。

OCx 输出信号的极性由 TIM1\_CCEN 寄存器中的 CCxP 位和 CCxNP 位决定。

该模式下，TIM1\_CCx 影子寄存器与计数器之间仍在进行比较，当产生匹配时，相应标志会被置位。

### 14.3.10 PWM 模式

PWM 模式生成的信号，其频率由 TIM1\_ARR 寄存器的值决定，其占空比由 TIM1\_CCx 寄存器的值决定。

每个输出通道都可以独立选择 PWM 模式输出：

- 通过 TIM1\_CCMx 寄存器中的 OCxM[2:0]位域来配置。
  - PWM 模式 1 (OCxM[2:0]位域为 110)：在递增计数模式下，当 TIM1\_CNT < TIM1\_CC1，通道 1 输出为有效电平，否则为无效电平。在递减计数模式下，当 TIM1\_CNT > TIM1\_CC1，通道 1 输出为无效电平，否则为有效电平；
  - PWM 模式 2 (OCxM[2:0]位域为 111)：在递增计数模式下，当 TIM1\_CNT < TIM1\_CC1，通道 1 输出为无效电平，否则为有效电平。在递减计数模式下，当 TIM1\_CNT > TIM1\_CC1，通道 1 输出为有效电平，否则为无效电平；
- 配置 TIM1\_CCMx 寄存器中的 OCxPE 位置 1 使能相应比较寄存器的预装载功能 (TIM1\_CCx)；
- 配置 TIM1\_CR1 寄存器中的 ARPE 位置 1 使能自动重载寄存器的预装载功能 (TIM1\_ARR)。

当使能预装载功能时，在发生更新事件时，预装载寄存器的值才会加载到影子寄存器中，所以启动计数器前，建议将 TIM1\_EVTG 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx/OCxN 的极性可通过 TIM1\_CCEN 寄存器的 CCxP 和 CCxNP 位来配置。通过 TIM1\_CCEN 寄存器中的 CCxE、CCxNE 和 TIM1\_BDT 寄存器中的 MOEN、OSSI 和 OSSR 位组合来选择输出通道的输出方式，详见表：[互补通道 OCx 和 OCxN 的输出控制位 \(MOEN 位为 1\)](#)。

根据 TIM1\_CR1 寄存器中 CMS[1:0]位域的值，选择 PWM 信号的对齐模式：

#### PWM 边沿对齐模式

- 递增计数

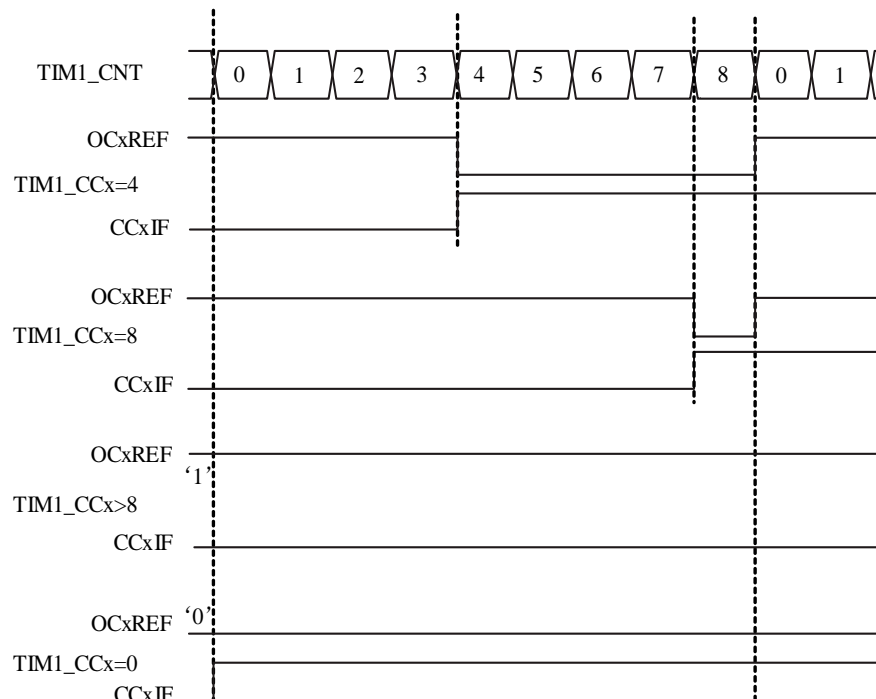
当 TIM1\_CR1 寄存器中的 DIR 位为 0 时执行递增计数。请参考[递增计数模式](#)。

下面是一个 PWM 模式 1 的例子。当 TIM1\_CNT < TIM1\_CCx，OCxREF 信号输出高电平，否则为低电平。如果 TIM1\_CCx 寄存器的值大于

TIM1\_ARR 寄存器的值，则 OCxREF 保持为 1。如果 TIM1\_CCx 寄存器的值为 0，则 OCxREF 保持为 0。

例如：TIM1\_ARR 寄存器的值为 8 时，递增计数下的边沿对齐模式 PWM 波形。

图 14-22 递增计数下的边沿对齐模式 PWM 波形（TIM1\_ARR 寄存器为 8）



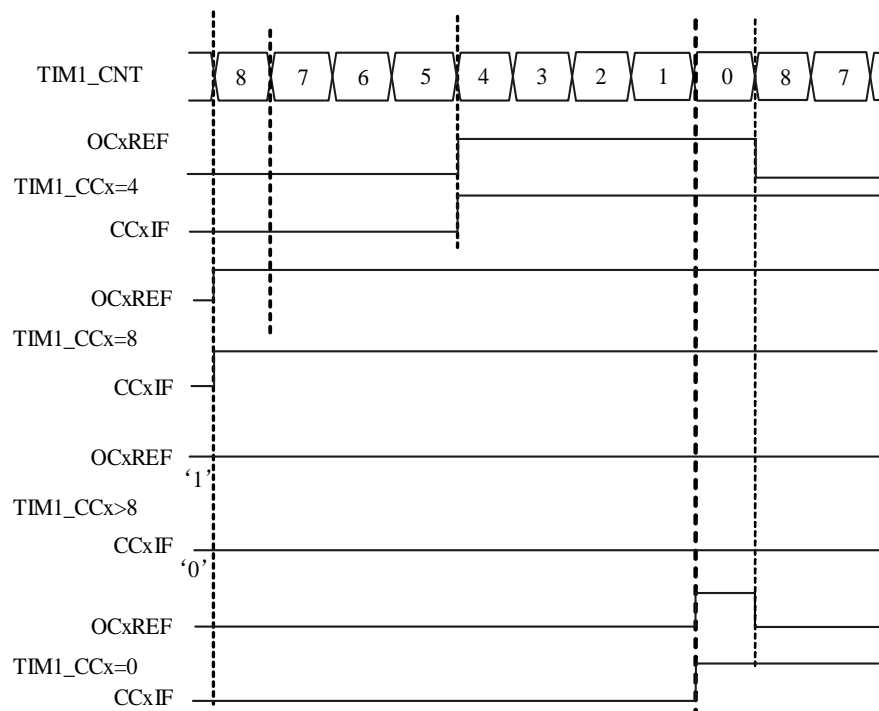
## ● 递减计数

当 TIM1\_CR1 寄存器中的 DIR 位为 1 时执行递减计数。请参考[递减计数模式](#)。

在 PWM 模式 1 下，当  $TIM1\_CNT > TIM1\_CCx$ ，OCxREF 信号输出低电平，否则为高电平。如果 TIM1\_CCx 中的值大于自动重载值（TIM1\_ARR 中的值），则 OCxREF 保持为 1。此模式下不可能产生占空比为 0% 的 PWM 波形。

例如：TIM1\_ARR 寄存器的值为 8 时，递减计数下的边沿对齐模式 PWM 波形。

图 14-23 递减计数下的边沿对齐模式 PWM 波形（TIM1\_ARR 寄存器为 8）



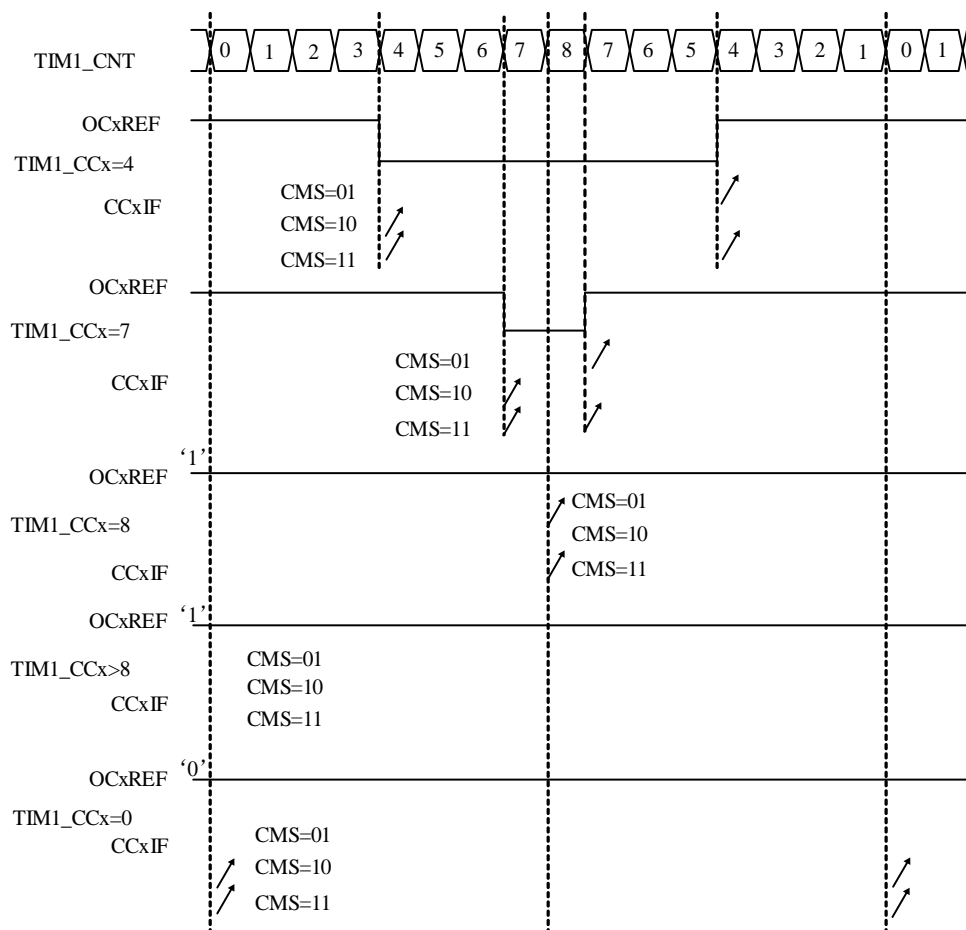
## PWM 中心对齐模式

当 TIM1\_CR1 寄存器中的 CMS[1:0]位域不为 00 时，为中心对齐模式。

如果配置 CMS[1:0]位域为中心对齐模式 3 时，在每个 PWM 周期内，计数器将产生上溢和下溢各一次，所以可更新 2 次 PWM 占空比。请参见[中心对齐模式（递增/递减计数）](#)。

例如：当 PWM 模式为 PWM 模式 1，TIM1\_ARR 寄存器为 8 时，中心对齐模式的 PWM 波形。

图 14-24 中心对齐模式 PWM 波形 (TIM1\_ARR 寄存器为 8)



#### 14.3.11 清除 OCxREF 信号

对于给定的通道，OC\_CLR 信号上的高电平可将 OCxREF 信号复位，OCxREF 信号将保持低电平，直到发生更新事件。

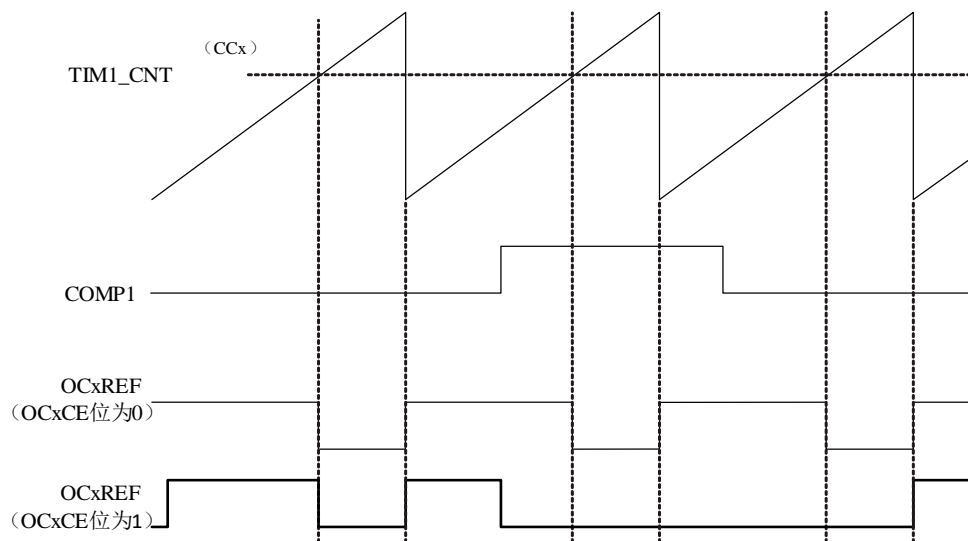
TIM1\_CFG 寄存器中的 OCREF\_CLR 位为 0 时，COMP1 输出连接到 OC\_CLR 输入；OCREF\_CLR 位为 1 时，COMP2 输出连接到 OC\_CLR 输入。

该功能只能用于比较输出模式和 PWM 模式，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。

下图为 OCxCE 不同配置下，当 COMP1 输入变为高电平时 OCxREF 的信号波形。在本例中，TIM1 配置为 PWM 模式。



图 14-25 清除 TIM1 的 OCxREF



### 14.3.12 单脉冲模式

单脉冲模式(OPM)是比较输出模式的一个特例。将 TIM1\_CR1 寄存器中的 OPM 位置 1, 即可选择单脉冲模式。在此模式下, 计数器接收到触发信号, 在一段可编程的延时后产生一个脉宽可编程的脉冲。

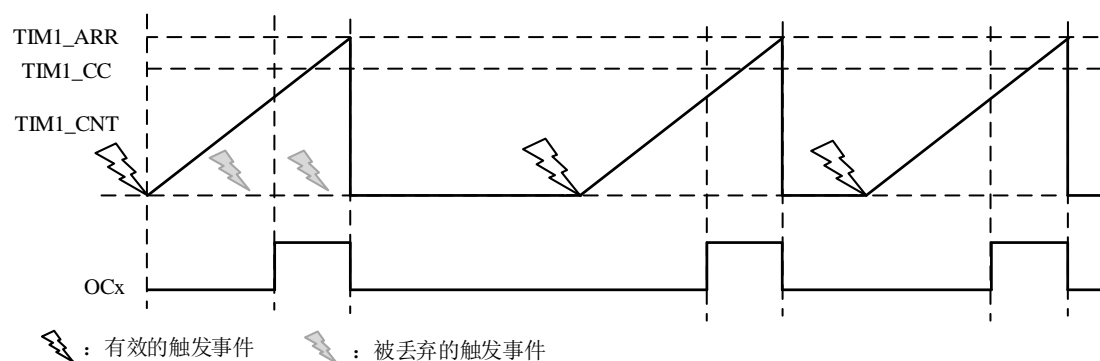
可以通过从模式控制器启动计数器。在比较输出模式或 PWM 模式下生成波形。当发生下一更新事件时, 计数器将自动停止。

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:  $CNT > CCx$

启动计数后, 在一个计数周期内发生的任何触发事件均将被丢弃。如下图所示:

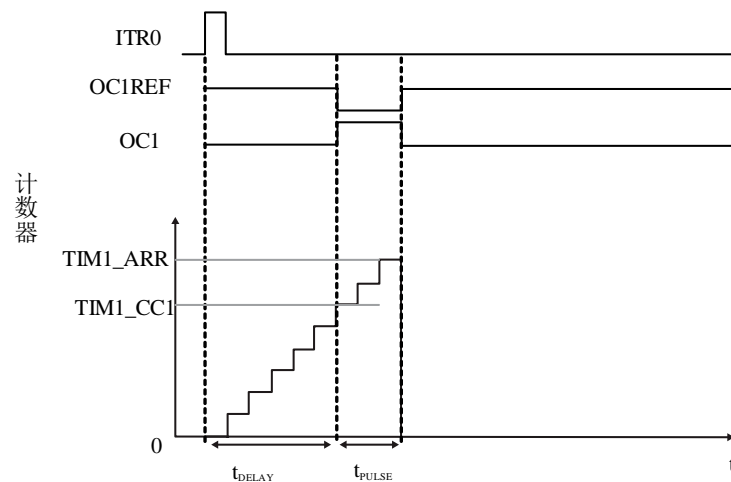
图 14-26 单脉冲输出波形



例如, 在 ITR0 检测到上升沿时, 经过  $t_{DELAY}$  的延迟, 在 OC1 上产生一个宽度

为  $t_{PULSE}$  的正脉冲。

图 14-27 单脉冲模式示例



使用 ITR0 作为触发信号，TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域写入 110（触发模式）。

单脉冲模式的脉冲宽度由当前时钟频率和计数器预分频器决定。

- $t_{DELAY}$  由写入 TIM1\_CC1 寄存器的值定义。
- $t_{PULSE}$  由自动重载值与比较值之差（TIM1\_ARR - TIM1\_CC1）来定义。
- 若产生这样的波形：信号在发生比较匹配时从 0 变 1，在计数器达到自动重载值时由 1 变为 0：
  - OC1 输出极性：CC1P 位为 1
  - 使能 PWM 模式 1（TIM1\_CCM1 寄存器中的 OC1M 写入 110）
  - 如果需要，可使能预装载功能（TIM1\_CCM1 寄存器的 OC1PE 和 TIM1\_CR1 寄存器的 ARPE 位置 1）
  - 在 TIM1\_CC1 寄存器中写入比较值
  - 在 TIM1\_ARR 寄存器中写入自动重载值
  - 将 UG 位置 1，产生一个更新事件，之后等待 ITR0 上的触发事件启动计数器计数。

此例中，TIM1\_CR1 寄存器中的 DIR 和 CMS 位应为 0。

#### OCx 快速使能：

在单脉冲模式下，ITR0 输入的边沿检测会使能计数器（CEN 位自动置 1），之后在计数器值与比较值之间发生比较操作产生输出的转换。但此操作需要一定的

时钟周期，因此它限制了可得到的最小延时  $t_{\text{DELAY}}$ 。

如果要输出延迟时间最短的波形，可以将 TIM1\_CCMx 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF（和 OCx）对触发信号做出响应，而不依赖于比较的结果。其输出的波形与比较匹配时的波形相同。仅在通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

### 14.3.13 互补输出及死区插入

TIM1 的互补输出信号 OCx 和 OCxN，可独立选择极性（配置 TIM1\_CCEN 寄存器中的 CCxP 和 CCxNP 位），其输出控制由以下寄存器的相应位决定：TIM1\_CCEN 寄存器中的 CCxE 和 CCxNE 位及 TIM1\_BDT 寄存器中的 MOEN 和 OSSR 位。

MOEN 为输出使能控制位，该位可由软件置 1，也可通过配置 TIM1\_BDT 寄存器中的 AOEN 位为 1，使 MOEN 位在发生更新事件时自动置 1。

下表为互补输出信号 OCx 和 OCxN 的配置方式：

表 14-3 互补输出信号 OCx 和 OCxN 的配置方式（MOEN 位为 1）

OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
×	0	0	禁止输出（TIM 释放输出控制，与 IO 输出控制电路间为高阻态） <sup>(1)</sup>	
0	0	1	禁止输出（TIM 释放输出控制，与 IO 输出控制电路间为高阻态）	由 OCxREF 及 CCxNP 位决定
0	1	0	由 OCxREF 及 CCxP 位决定	禁止输出（TIM 释放输出控制，与 IO 输出控制电路间为高阻态）
×	1	1	由 OCxREF、CCxP 位及死区时间决定 <sup>(1)</sup>	由 OCxREF 互补项、CCxNP 位及死区时间决定 <sup>(1)</sup>
1	0	1	关闭状态（输出为无效状态）： OCx=CCxP	由 OCxREF 及 CCxNP 位决定
1	1	0	由 OCxREF 及 CCxP 位决定	关闭状态（输出为无效状态） OCxN=CCxNP

1. 如果互补通道均未使用，TIM1\_CR2 寄存器中的 OISx 位、OISxN 位和 TIM1\_CCEN 寄存器中的 CCxP 位、CCxNP 位应配置为 0。

启动计数后，当 TIM1\_CCEN 寄存器中的 CCxE、CCxNE 位和 TIM1\_BDT 寄存器中的 MOEN 位均为 1 时，使能死区插入。

互补通道有一个 8 位的死区发生器，该时钟源为 TIM1\_KCLK，基于该时钟源分频后的时钟频率为  $f_{\text{DTS}}$ 。每个通道的死区时间都是相同的，可通过 TIM1\_BDT 寄存器中的 DTG[7:0] 位域进行配置。

OCxREF 生成 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效，则：

- OCx 与 OCxREF 信号相同，但 OCx 的上升沿相对 OCxREF 的上升沿之间存在死区时间。
- OCxN 与 OCxREF 信号相反，但 OCxN 的上升沿相对 OCxREF 的下降沿之间存在死区时间。

如果死区时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为带死区插入的 OCx/OCxN 与 OCxREF 之间的关系。（假定 CCxP 位为 0、CCxNP 位为 0、MOEN 位为 1、CCxE 和 CCxNE 位为 1）。

图 14-28 带死区插入的互补输出

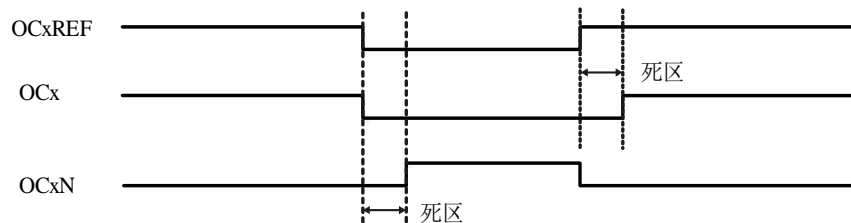


图 14-29 死区时间大于负脉冲宽度的死区波形

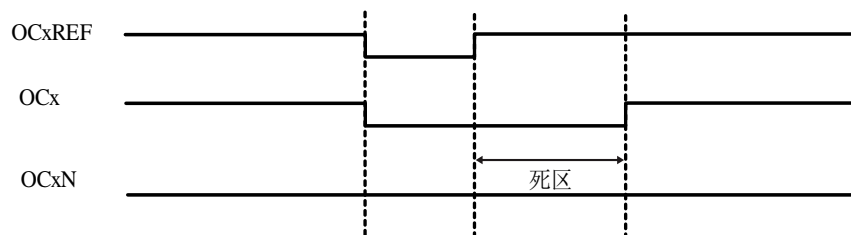
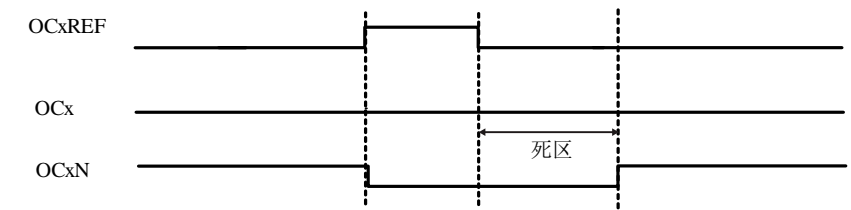


图 14-30 死区时间大于正脉冲宽度的死区波形



### OCxREF 重定向

通过配置 TIM1\_CCEN 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 信号重定向到 OCx 或 OCxN 信号上。

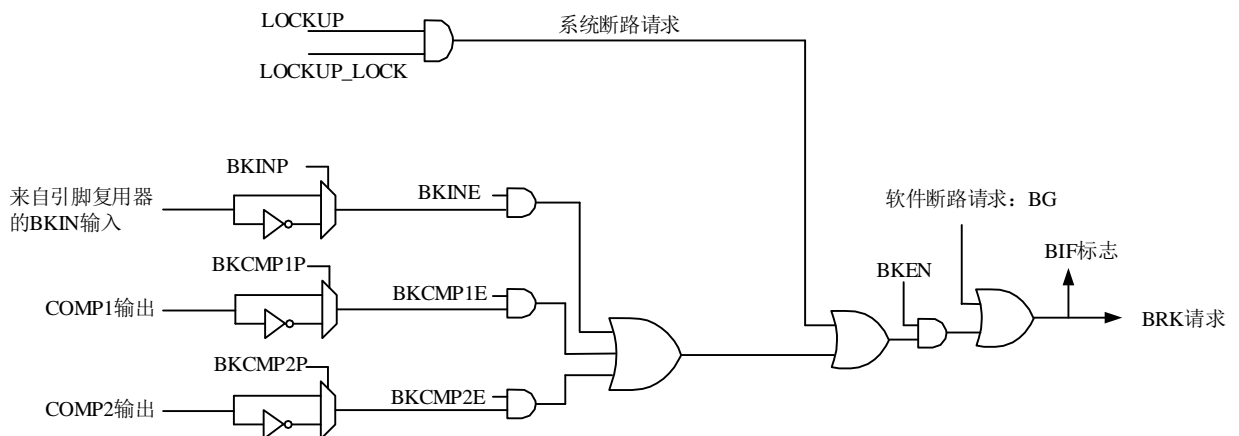
如果仅使能 OCxN（CCxE 位为 0，CCxNE 位为 1），其信号不会反向。当 OCxREF 为高电平，OCxN 即变为高电平。例如，如果 CCxNP 位为 0，则 OCxN 即 OCxREF。如果同时使能 OCx 和 OCxN（CCxE、CCxNE 均置 1），OCxREF 为高电平时，OCx 将变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效（有

效电平由 TIM1\_CCEN 寄存器中的 CCxP 和 CCxNP 位决定)。

#### 14.3.14 使用断路功能

TIM1 有 1 个断路输入。故障发生时关闭 PWM 输出，也可通过芯片内部故障事件关闭输出。

图 14-31 断路电路概述



#### 断路（BRK）通道源：

- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能设定），可通过 TIM1\_BDT BKINP 位配置断路的有效电平
- 内部源：
  - Cortex-M0+ LOCKUP 输出；
  - 比较器的输出，具有可配置的有效电平选择。

使能断路功能可通过 TIM1\_BDT 寄存器中的 BKEN 位置 1 来配置。对 BKEN 位执行写操作时，会在 1 个 TIM1\_KCLK 时钟周期后生效。

通过配置 TIM1\_AF1 寄存器的相应位来使能或禁止断路事件源。

通过 TIM1\_EVTG 寄存器中的 BG 位置 1，可软件生成断路事件，与 BKEN 的值无关。

由于 MOEN 可能为异步信号，所以进行写操作时，必须插入等待周期，之后才能准确读取。

断路（BRK）通道可在死区时间后将输出强制为预定义的电平（有效或无效）。

表 14-4 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位（MOEN 位为 0）

控制位			输出状态
OSSI	CCxE	CCxNE	OCx/OCxN 输出状态
0	×	×	禁止输出（TIM释放输出控制，与IO输出控制电路间为高阻态）
1	0	0	OCx/OCxN的输出首先为无效电平，之后在死区时间后其切换为空闲电平； <i>注意：为了防止短路，当OISx位和OISxN位均配置为有效电平时，OCx和OCxN的输出被强制为无效电平。</i>
	0	1	
	1	0	
	1	1	

当发生断路时：

- MOEN 位异步清 0，输出端所处的状态由 TIM1\_BDT 寄存器中的 OSSI 位来选择。
- 使用互补输出时：
  - 首先输出为无效电平，OCx 和 OCxN 的输出由 TIM1\_CCEN 寄存器中的 CCxP 位和 CCxNP 决定。即使 TIM1 没有时钟，该输出状态仍有效。
  - 如果 TIM1 时钟存在，则将激活死区发生器，死区时间后，OCx 和 OCxN 的输出由 TIM1\_BDT 寄存器中的 OISx 和 OISxN 位决定。因为 MOEN 需进行再同步，因此死区时间会比 TIM1\_BDT 寄存器中 DTG[7:0]位域配置的时间稍长一些。
  - 如果 OSSI 位为 0，TIM1 将释放输出控制（GPIO 控制器接管——高阻态）。
- 将断路状态标志 TIM1\_SR 寄存器中的 BIF 位置 1。如果此时 TIM1\_DIER 寄存器中的 BIE 位置 1，则会产生中断。

当断路输入为有效电平时，MOEN 位无法置 1，同时 TIM1\_SR 寄存器中的 BIF 位也无法清 0。

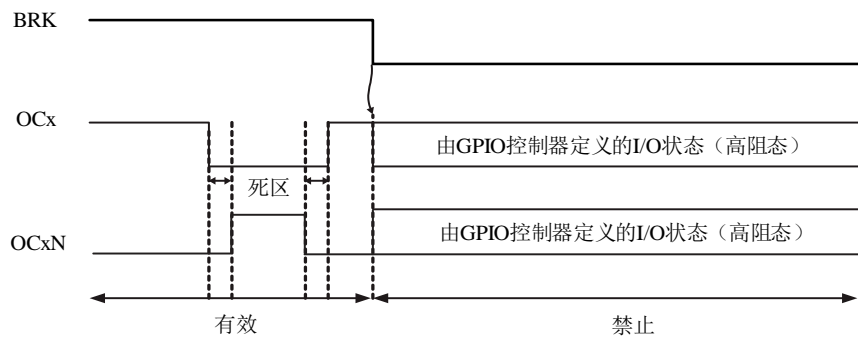
通过 TIM1\_BDT 寄存器中的 LOCK 位实现断路功能相关寄存器的写保护功能，有 3 种保护级别，可对死区时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和有效电平的写操作进行权限保护。

当对 LOCK 位执行一次写操作后，该位不能再被修改，直到下次复位时该位的写权限才被解锁。

下图为当发生 BRK 断路事件时，输出信号的状态。



图 14-33 BRK 使能后的 PWM 输出状态 (OSSI=0)



### 14.3.15 生成 6 步 PWM

具有互补输出功能的通道上，预装载位的配置说明预装载位有 OCxM、CCxE 和 CCxNE 位。发生换相 (COM) 事件时，这些预装载位将装载到影子寄存器位。这样便可预先设置好下一步配置，并在同一时刻修改所有通道的配置。

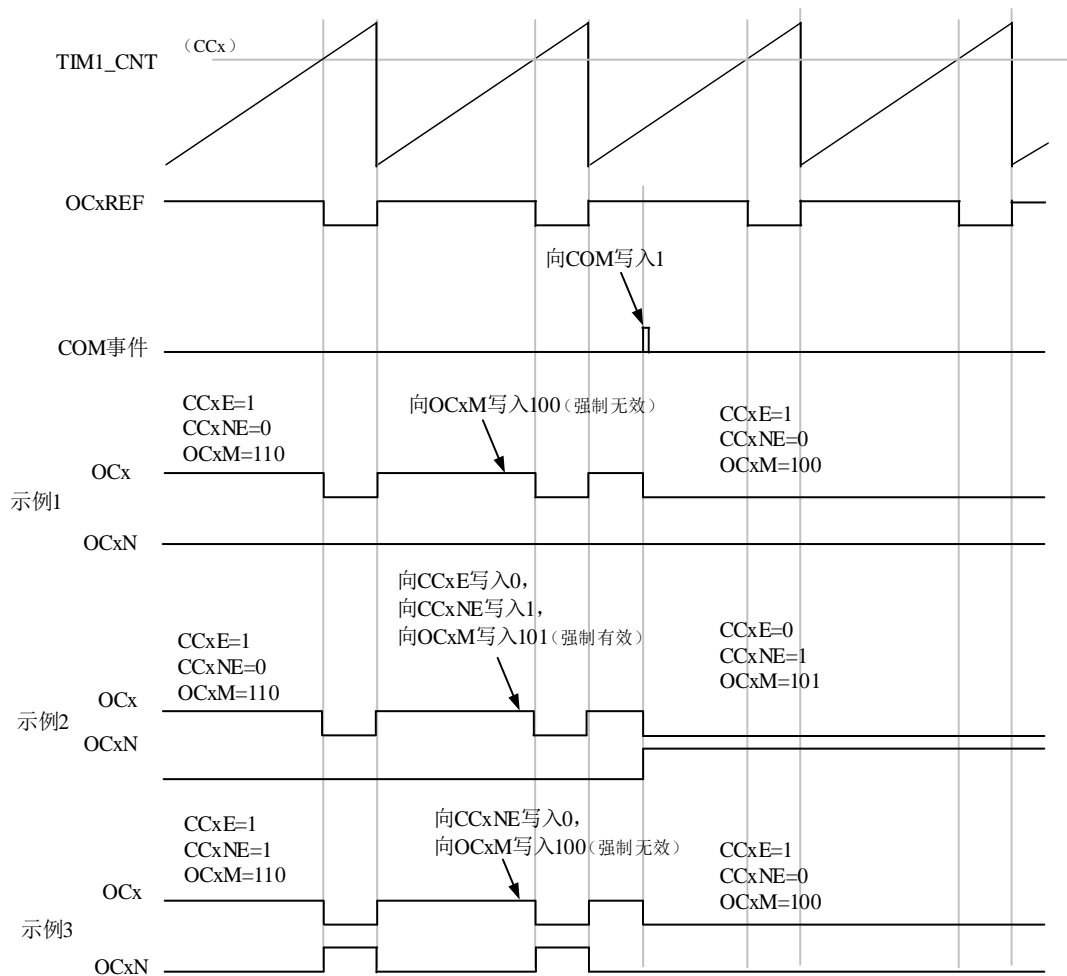
换相事件可通过配置 TIM1\_EVTG 寄存器中的 COMG 位置 1 来触发，也可由 TRIG 上升沿自动触发。

发生换相事件时，TIM1\_SR 寄存器中的 COMIF 位会自动置 1。如果 TIM1\_DIER 寄存器中的 COMIE 位置 1，将产生中断。

下图显示当发生换相事件时，3 种不同配置下的 OCx 和 OCxN 输出行为。



图 14-34 COM 事件生成 6 步 PWM 的示例 (OSSR=1)



### 14.3.16 霍尔传感器接口

使用高级控制定时器 (TIM1) 产生 PWM 信号驱动马达时, 可采用通用 TIM3 定时器作为“接口定时器”来连接霍尔传感器。

TIM3 的 3 个输入引脚 (TIM3\_CH1、TIM3\_CH2 和 TIM3\_CH3) 通过异或门连接到 TI1 输入通道 (配置 TIM3\_CR2 寄存器中的 TI1\_XOR\_SEL 位为 1), 并由 TIM3 进行输入捕获。

TIM3 的从模式控制器配置为复位模式; 选择触发信号为 TI1F\_ED。每当 3 个输入信号中任一输入发生翻转时, 计数器会从 0 开始重新计数。这样将产生一个由霍尔输入端的任何变化都会生成触发事件。

在 TIM3 上, 捕获/比较通道 1 配置为捕获模式, 捕获信号为 TRC。捕获值反映了两次输入变化间的时间, 可提供与电机转速相关的信息。

TIM3 输出通道可用于在一个指定的延时 (比较输出或 PWM 模式) 之后产生一个上升沿电平信号, 此信号可以通过 TRIG\_OUT 输出被送到 TIM1, 作为换向

的触发信号。

示例：霍尔输入连接到 TIM3，每当霍尔输入发生变化后的一个指定时间后，改变 TIM1 的 PWM 配置（包含 CCxE、CCxNE、OCxM 等相应位）。

对 TIM3 的配置，步骤如下：

- TIM3\_CR2 寄存器的 TI1\_XOR\_SEL 位置 1，使 3 个定时器输入经过异或运算后到 TI1 输入通道；
- 配置 TIM3\_ARR 寄存器的值(TIM3\_CNT 的值必须通过 TI1 的变化清 0)。设置预分频值，以得到最大计数器周期，该周期大于传感器上两次变化的间隔时间；
- 将通道 1 配置为捕获模式(选择 TRC)：向 TIM3\_CCM1 寄存器的 CC1S[1:0] 位域写入 11。如果需要，还可配置数字滤波器；
- 配置为复位模式：TIM3\_SMC 寄存器中的 SM\_SEL[2:0]位域写入 100；
- 选择 TI1\_FD 作为输入源：TIM3\_SMC 寄存器中的 TS[1:0]位域写入 01；
- 将通道 2 配置为 PWM2 模式，并向 TIM3\_CCM1 寄存器的 OC2M [2:0]位域写入 111，CC2S[1:0]位域写入 00；
- 选择 OC2REF 作为 TRIG\_OUT 上的触发输出：向 TIM3\_CR2 寄存器的 MM\_SEL[2:0]位域写入 101；

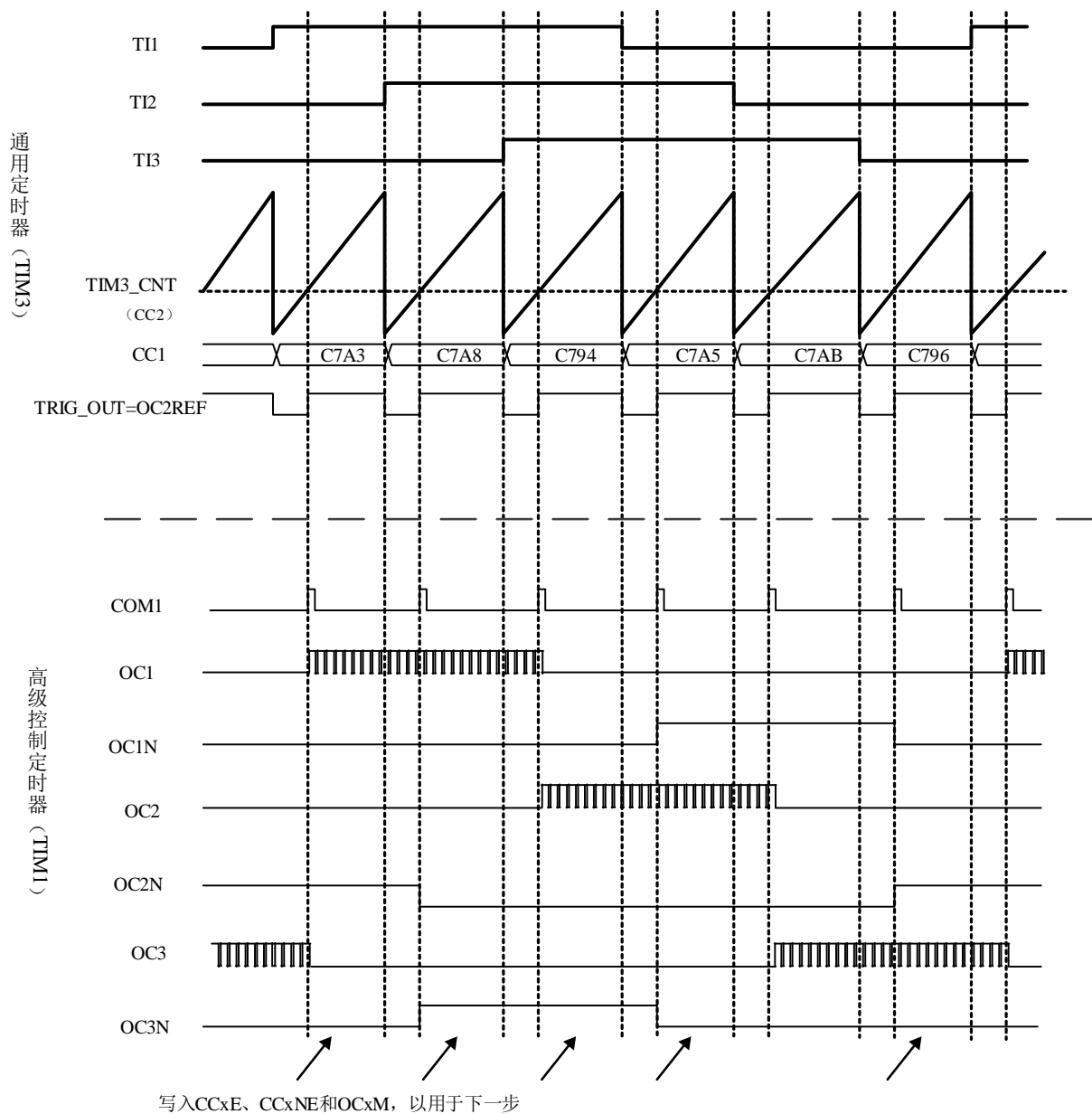
对 TIM1 的配置，步骤如下：

- 触发输入的上升沿生成一个换向事件：TIM1\_CR2 寄存器中的 CCU\_SEL 位置 1；
- 使能比较换相预装载：(TIM1\_CR2 寄存器的 CC\_PRECR 位置 1；
- 如果此时 TIM1\_DIER 寄存器中的 COMIE 位置 1，将产生中断；

在 TIM1 中，定时器配置为 PWM 模式，发生换相事件后，预装载使能位有效时，PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由换相中断子程序中完成。

下图为本示例的示意图。

图 14-35 霍尔传感器接口的示例



### 14.3.17 定时器同步从模式

通过 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域来配置从模式的类型。从模式的计数时钟源只有内部时钟源可用。

以下几种模式为从模式。

#### 复位模式

当触发信号有效时，计数器及其预分频器被重新初始化。

如果 TIM1\_CR1 寄存器中的 URS 位为 0，且 UDIS 位也为 0，则会生成更新事件，且 TIM1\_ARR 和输出比较模式下的 TIM1\_CCx 寄存器的影子寄存器被更

新。

在以下示例中，ITR0 输入信号出现上升沿时，递增计数器清 0：

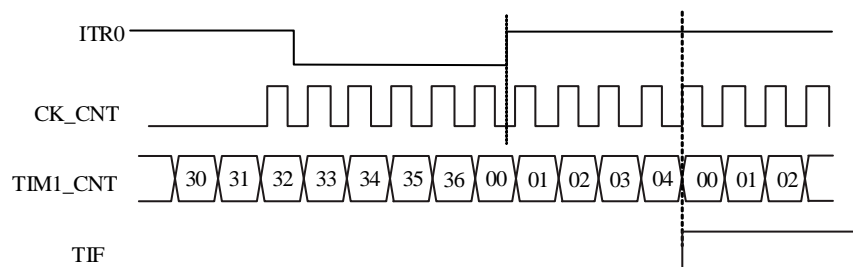
- 配置为复位模式：TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域写入 100。
- 使能计数器：TIM1\_CR1 寄存器中的 CEN 位置 1。

当 ITR0 出现上升沿时，计数器清 0，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；如果使能 TIM1\_DIEN 寄存器中的 TIE 位，则可发送中断。

ITR0 的上升沿与实际计数器复位之间的延迟是由于触发信号 ITR0 输入的重新同步电路引起的。

下图为自动重载值寄存器 TIM1\_ARR=0x36 时的时序。

图 14-36 复位模式下的控制时序



## 门控模式

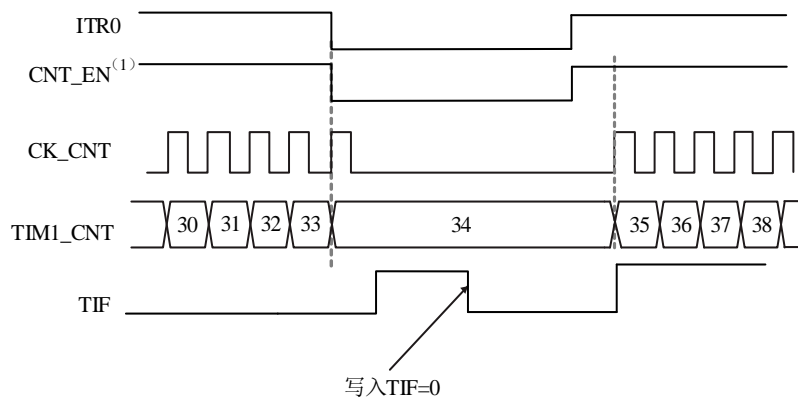
输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 ITR0 输入为高电平时计数：

- 定时器配置为门控模式：TIM1\_SMC 寄存器中的 SM\_SEL[2:0] 位域写入 101；
- 使能计数器：TIM1\_CR1 寄存器中的 CEN 位置 1。

当 ITR0 变为低电平时停止计数。计数器启动或停止时，TIM1\_SR 寄存器中的 TIF 标志置 1。

图 14-37 门控模式下的控制时序



1. CNT\_EN 为 TIM1 内部控制信号。

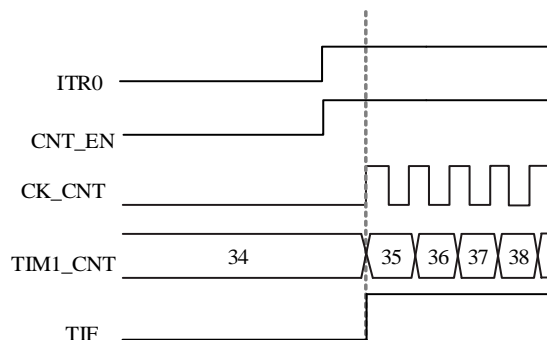
### 触发模式

所选输入端触发信号有效时可以使能计数器（CEN 位为 1）。

在以下示例中，配置为递增计数模式，当 ITR0 输入上出现上升沿时，使能计数器：

- 定时器配置为触发模式：TIM1\_SMC 寄存器中 SM\_SEL[2:0] 位域写入 110；当 ITR0 出现上升沿时，计数器开始根据内部时钟计数，TIM1\_SR 寄存器中的 TIF 标志置 1。

图 14-38 触发模式下的控制时序



#### 14.3.18 定时器同步主模式

TIM1 和内部其他 TIM 连在一起，可实现定时器同步或级联。

详细内容及相关配置请参见：[定时器同步主模式 \(TIM3\)](#)。

#### 14.3.19 调试模式

当 CPU 进入调试模式（Cortex-M0+内核停止）时，TIM1 计数器会根据 [APB 冻结寄存器 \(DBG\\_APB\\_FZ2\)](#) 寄存器中的 TIM1\_HOLD 配置位选择继续计数或者停止计数。当计数器停止（TIM1\_HOLD 置 1）时，为安全起见，输出被禁止（与

MOEN 位被复位的效果一致)。

#### 14.4 TIM1 中断

表 14-5 中断请求

中断事件	事件标志	使能控制位	清除方法
更新事件	UIF	UIE	UIF 位写 0 清除该位
比较 1~4	CCxIF	CCxIE	CCxIF 位写 0 清除该位
触发事件	TIF	TIE	TIF 位写 0 清除该位
换向事件	COMIF	COMIE	COMIF 位写 0 清除该位
断路事件	BIF	BIE	BIF 位写 0 清除该位

## 14.5 TIM1 寄存器

TIM1 寄存器支持 32 位访问。

表 14-6 TIM1 基地址

外设	基地址
TIM1	0x4001 2C00

### 14.5.1 TIM1 控制寄存器 1 (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.						CLK_DIV[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
31:10	保留	写入无效
9:8	CLK_DIV[1:0]	<p>时钟分频</p> <p>定时器时钟频率 (<math>f_{TIM1\_KCLK}</math>) 与死区发生器所使用的采样时钟 (<math>f_{DTS}</math>) 之间的分频比。</p> <p>00: <math>f_{DTS} = f_{TIM1\_KCLK}</math></p> <p>01: <math>f_{DTS} = f_{TIM1\_KCLK}/2</math></p> <p>10: <math>f_{DTS} = f_{TIM1\_KCLK}/4</math></p> <p>11: 保留 (默认为: <math>f_{DTS} = f_{TIM1\_KCLK}</math>)</p>
7	ARPE	<p>TIM1_ARR 寄存器预装载使能控制</p> <p>0: 禁止</p> <p>1: 使能</p>
6:5	CMS[1:0]	<p>计数模式选择</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。 上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置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。 上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置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。 上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置1</p> <p><i>注意：只要计数器处于使能状态（CEN位为1），无法从边沿对齐模式切换为中心对齐模式。</i></p>
4	DIR	<p>计数方向</p> <p>0: 递增计数</p> <p>1: 递减计数</p> <p><i>注意：当TIM1配置为中心对齐模式时，该位为只读状态。</i></p>
3	OPM	<p>单脉冲模式</p> <p>0: 禁止：计数器在发生更新事件时不会停止计数</p> <p>1: 使能：计数器在发生更新事件时停止计数（CEN位自动清0）</p>
2	URS	<p>更新事件源选择</p> <p>UDIS为0时：</p> <p>0: 以下事件都会产生更新事件：</p> <ul style="list-style-type: none"> <li>- 计数器上溢/下溢（RCR设置为0）</li> <li>- RCR递减到0，发生上溢或下溢（RCR设置为非0）</li> <li>- 将UG位置1</li> <li>- 通过从模式控制器生成的更新事件（复位模式）</li> </ul> <p>1: 以下事件都会产生更新事件：</p> <ul style="list-style-type: none"> <li>- 计数器上溢/下溢（RCR设置为0）</li> <li>- RCR递减到0，发生上溢或下溢（RCR设置为非0）</li> </ul> <p><i>注意：当UDIS位为0时，使能更新事件，此时URS位配置为1，则将UG位置1，会生成更新事件，计数器及PSC预分频计数器均被初始化，但不会将UIF置位。</i></p>
1	UDIS	<p>更新事件禁止控制位</p> <p>0: 使能：由URS位决定更新事件的触发源</p>



1: 禁止:

注意: 将UDIS位置1, 如果此时将UG位置1, 或者通过从模式控制器生成的更新事件(复位模式), 或计数器上溢/下溢时, 计数器和PSC预分频计数器被重新初始化, 但不会将UIF置位。

0 CEN 计数器使能

0: 禁止

1: 使能

注意: 在使用时钟模式1或门控模式时, 需软件将CEN置1。而触发模式可自动将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0。

#### 14.5.2 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res.	MM_SEL[2:0]			Res.	CCU_SEL	Res.	CC_PRECR
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw		rw

位/位域	名称	描述
31:15	保留	写入无效
14	OIS4	OC4的空闲电平 请参见OIS1位
13	OIS3N	OC3N的空闲电平 请参见OIS1N位
12	OIS3	OC3的空闲电平 请参见OIS1位
11	OIS2N	OC2N的空闲电平

		请参见OIS1N位
10	OIS2	OC2的空闲电平 请参见OIS1位
9	OIS1N	OC1N的空闲电平 0: 当TIM1_BDT寄存器的MOEN位为0, 经过死区时间后 OC1N输出为低电平 1: 当TIM1_BDT寄存器的MOEN位为0, 经过死区时间后 OC1N输出为高电平 <i>注意: 当TIM1_BDT寄存器中的LOCK位域定义为级别1、2或3 时, 此位即无法修改。</i>
8	OIS1	OC1的空闲电平 0: 当TIM1_BDT寄存器的MOEN位为0, 经过死区时间后OC1输 出低电平 1: 当TIM1_BDT寄存器的MOEN位为0, 经过死区时间后OC1输 出高电平 <i>注意: 当TIM1_BDT寄存器中的LOCK位域定义为级别1、2或3 时, 此位即无法修改。</i>
7	保留	写入无效
6:4	MM_SEL[2:0]	主模式选择 触发输出 (TRIG_OUT) 信号连接到其他外设的控制方式。 000: 复位 — 以下信号都可以用作触发输出 (TRIG_OUT): – TIM1_EVTG寄存器中的UG位 – 触发输入生成 (从模式控制器配置为复位模式), 但 TRIG_OUT上的信号相比实际复位会有一定延时 001: 使能 — 计数器使能信号用作触发输出 (TRIG_OUT) 当TIM1_CR1寄存器的CEN位置位或门控模式信号有效时 产生。为保证当前定时器与从外设实现完美同步, 需选 择主/从模式 (TIM1_SMC寄存器中MS_MOD位置1)。 010: 更新 — 选择更新事件作为触发输出 (TRIG_OUT) 011: 比较脉冲 — 当通道1发生比较匹配时, 输出 (TRIG_OUT) 会发送一个正脉冲 100: 比较 — OC1REF信号用作触发输出 (TRIG_OUT)

		101: 比较 — OC2REF信号用作触发输出 (TRIG_OUT)
		110: 比较 — OC3REF信号用作触发输出 (TRIG_OUT)
		111: 比较 — OC4REF信号用作触发输出 (TRIG_OUT)
3	保留	写入无效
2	CCU_SEL	<p>换相控制更新选择</p> <p>0: 当CC_PRECR置1时, 仅通过TIM1_EVTG寄存器中的COMG位置1, 产生换相事件, 此时对TIM1_CCEN寄存器的CCxE、CCxNE位和TIM1_CCM1寄存器的OCxM位进行更新</p> <p>1: 当CC_PRECR置1时, 可通过将TIM1_EVTG寄存器中的COMG位置1或TRIG的上升沿对TIM1_CCEN寄存器CCxE、CCxNE位和TIM1_CCM1寄存器的OCxM位进行更新。</p> <p><i>注意: 此位仅对具有互补输出的通道有效。</i></p>
1	保留	写入无效
0	CC_PRECR	<p>比较换相预装载控制</p> <p>0: 禁止: CTIM1_CCEN寄存器的CCxE、CCxNE位和TIM1_CCM1寄存器的OCxM位未进行预装载</p> <p>1: 使能: TIM1_CCEN寄存器的CCxE、CCxNE位和TIM1_CCM1寄存器的OCxM位进行预装载, 写入这些位后, 仅当发生换相事件 (TIM1_EVTG寄存器中的COMG位置1或在TRIG上检测到上升沿, 取决于CCU_SEL位) 时才会对这些位进行更新</p> <p><i>注意: 此位仅对具有互补输出的通道有效。</i></p>

### 14.5.3 TIM1 从模式控制寄存器 (TIM1\_SMC)

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.								MS_MOD	Res.				SM_SEL[2:0]		

									rw						rw	rw	rw
--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	----	----	----

位/位域	名称	描述
31:8	保留	写入无效
7	MS_MOD	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（TRIG）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 TRIG_OUT）。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
6:3	保留	写入无效
2:0	SM_SEL[2:0]	从模式选择 触发信号（TRIG）的有效边沿由外部输入端所选的极性决定。 0xx: 禁止从模式 — PSC 预分频器时钟由内部时钟提供，TIM1_CR1 寄存器中的 CEN 位为 1 时，启动计数 100: 复位模式 — 触发输入 TRIG 信号上升沿时，重新初始化计数器并生成更新事件 101: 门控模式 — 触发输入 TRIG 信号为高电平时使能计数器时钟。只要 TRIG 信号变为低电平，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 110: 触发模式 — 触发输入 TRIG 信号出现上升沿时启动计数器（但不复位）。只控制计数器的启动 111: 时钟模式 1 — 触发输入 TRIG 信号的上升沿提供计数器时钟

#### 14.5.4 TIM1 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.								BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

								rw	rw	rw	rw	rw	rw	rw	rw
--	--	--	--	--	--	--	--	----	----	----	----	----	----	----	----

位/位域	名称	描述
31:8	保留	写入无效
7	BIE	断路事件的中断使能 0: 禁止 1: 使能
6	TIE	触发事件的中断使能 0: 禁止 1: 使能
5	COMIE	换相事件的中断使能 0: 禁止 1: 使能
4	CC4IE	通道4比较事件的中断使能 0: 禁止 1: 使能
3	CC3IE	通道3比较事件的中断使能 0: 禁止 1: 使能
2	CC2IE	通道2比较事件的中断使能 0: 禁止 1: 使能
1	CC1IE	通道1比较事件的中断使能 0: 禁止 1: 使能
0	UIE	更新事件的中断使能 0: 禁止 1: 使能

### 14.5.5 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
								rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:8	保留	写入无效
7	BIF	<p>断路事件标志</p> <p>当发生断路事件时，该位自动置1。断路事件消失后，通过写0可清除该位。</p> <p>0：未发生断路事件</p> <p>1：发生断路事件</p>
6	TIF	<p>触发事件标志</p> <ul style="list-style-type: none"> <li>门控模式：该标志将在计数器启动或停止时置1。</li> <li>非门控模式：当使能从模式控制器后，在TRIG信号上检测到有效边沿时，该标志自动置1。</li> </ul> <p>通过写0可清除该位。</p> <p>0：未发生触发事件</p> <p>1：发生触发事件</p>
5	COMIF	<p>换相事件标志</p> <p>当发生换相事件时，该位自动置1，同时TIMx_CCEN寄存器的CCxE位、CCxNE和TIMx_CCMx寄存器的OCxM位被更新。通过写0可清除该位。</p> <p>0：未发生换相事件</p> <p>1：发生换相事件</p>
4	CC4IF	通道4的比较事件标志

请参见CC1IF说明

3	CC3IF	通道3的比较事件标志 请参见CC1IF说明
2	CC2IF	通道2的比较事件标志 请参见CC1IF说明
1	CC1IF	通道1的比较事件标志 当TIM1_CNT计数器的值与TIM1_CC1寄存器的值匹配时，此标志自动置1，中心对齐模式下除外（请参见TIM1_CR1寄存器中的CMS位说明）。通过写0可清除该位。 0：不匹配 1：匹配
0	UIF	更新事件标志 发生更新事件时自动置1。通过写0可清除该位。 0：未发生更新 1：发生更新： <ul style="list-style-type: none"> <li>当TIM1_CR1寄存器中UDIS位为0，且RCR设置为0，上溢或下溢时</li> <li>当TIM1_CR1寄存器中UDIS位为0，且RCR设置为非0，RCR递减到0，发生上溢或下溢时</li> <li>TIM1_CR1寄存器中的URS位和UDIS位均为0，且TIM1_EVTG寄存器中的UG位置1时</li> <li>TIM1_CR1寄存器中的URS位和UDIS位均为0，且从模式控制器生成的更新事件（复位模式）（参见TIM1_SMC寄存器说明）</li> </ul>

#### 14.5.6 TIM1 事件产生寄存器（TIM1\_EVTG）

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.								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	写入无效
7	BG	生成断路事件 软件置1可生成断路事件，此时MOEN位清0。
6	TG	生成触发事件 软件置1可生成触发事件。
5	COMG	生成换相事件 软件置1可生成换相事件。如果CC_PRECR位置1时，可更新TIMx_CCEN寄存器的CCxE位、CCxNE和TIMx_CCMx寄存器的OCxM位。 <i>注意：此位仅对具有互补输出的通道有效。</i>
4	CC4G	生成通道4比较事件 请参见CC1G说明
3	CC3G	生成通道3比较事件 请参见CC1G说明
2	CC2G	生成通道2比较事件 请参见CC1G说明
1	CC1G	生成通道1比较事件 软件置1可生成比较事件。
0	UG	更新生成 此位由软件置1可生成更新事件。 0：不执行任何操作 1：PSC预分频器计数器将清0（但TIM1_PSC寄存器的值不受影响）；TIM1_CNT被重新初始化。



## 14.5.7 TIM1 比较模式寄存器 1 (TIM1\_CCM1)

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
OC2CE	OC2M[2:0]			OC2PE	OC2FE	Res.		OC1CE	OC1M[2:0]			OC1PE	OC1FE	Res.	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:16	保留	写入无效
15	OC2CE	通道2输出比较清零使能 请参见OC1CE说明
14:12	OC2M[2:0]	通道2输出比较模式选择 请参见OC1M[2:0]说明
11	OC2PE	通道2输出比较预装载使能 请参见OC1PE说明
10	OC2FE	通道2输出比较快速使能 请参见OC1FE说明
9:8	保留	写入无效
7	OC1CE	通道1输出比较清零使能 0: OC1REF不受OC_CLR信号影响 1: OC_CLR信号上检测到高电平时，OC1REF立即清0
6:4	OC1M[2:0]	通道1输出比较模式选择 定义信号OC1REF的行为，OC1REF信号决定OC1和OC1N的值。 OC1REF为高电平有效。OC1和OC1N的有效电平则取决于TIM1_CCEN寄存器的CC1P位和CC1NP位。 000: 冻结 — TIM1_CC1与TIM1_CNT间的比较对输出无影响

（仅作计数器时使用）

001: 匹配输出有效电平 — 当TIM1\_CNT与TIM1\_CC1匹配时，OC1REF信号强制变为有效电平（匹配前OC1REF信号为无效电平）

010: 匹配输出无效电平 — 当TIM1\_CNT与TIM1\_CC1匹配时，OC1REF信号强制变为无效电平（匹配前OC1REF信号为无效电平）

011: 翻转 — 当TIM1\_CNT与TIM1\_CC1匹配时，OC1REF发生翻转

100: 强制变为无效电平 — OC1REF强制变为无效电平

101: 强制变为有效电平 — OC1REF强制变为有效电平

110: PWM模式1:

- 在递增计数模式下，当TIM1\_CNT<TIM1\_CC1，通道1输出为有效电平，否则为无效电平
- 在递减计数模式下，当TIM1\_CNT>TIM1\_CC1，通道1输出为无效电平，否则为有效电平

111: PWM模式2:

- 在递增计数模式下，当TIM1\_CNT<TIM1\_CC1，通道1输出为无效电平，否则为有效电平
- 在递减计数模式下，当TIM1\_CNT>TIM1\_CC1，通道1输出为有效电平，否则为无效电平

*注意：此位域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CC\_PRECR位置1，则仅当生成换相事件时，OC1M有效位才会从预装载位获取新值。*

3	OC1PE	通道1输出比较预装载使能
		0: 禁止：可随时向TIM1_CC1写入数据，写入后将立即生效
		1: 使能：可读/写访问预装载寄存器，TIM1_CC1预装载值在更新事件时才会生效
2	OC1FE	通道1输出比较快速使能
		此位用于加快触发输入事件对OC输出的影响
		0: 禁止：触发输入事件发生时，OC1也将根据计数器和CC1值匹配结果正常输出
		1: 使能：触发输入有效边沿相当于OC1输出上的比较匹配。随后，无论比较结果如何，OC1都设置为比较后电平。仅当通道配置为PWM1或PWM2模式时，OC1FE才会起作用

1:0                      保留                      写入无效

#### 14.5.8      **TIM1 比较模式寄存器 2 (TIM1\_CCM2)**

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
OC4CE	OC4M[2:0]			OC4PE	OC4FE	Res.		OC3CE	OC3M[2:0]			OC3PE	OC3FE	Res.	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:16	保留	写入无效
15	OC4CE	通道4输出比较清零使能 请参见TIM1_CCM1寄存器的OC1CE说明
14:12	OC4M[2:0]	通道4输出比较模式 请参见TIM1_CCM1寄存器的OC1M[2:0]说明
11	OC4PE	通道4输出比较预装载使能 请参见TIM1_CCM1寄存器的OC1PE说明
10	OC4FE	通道4输出比较快速使能 请参见TIM1_CCM1寄存器的OC1FE说明
9:8	保留	写入无效
7	OC3CE	通道3输出比较清零使能 请参见TIM1_CCM1寄存器的OC1CE说明
6:4	OC3M[2:0]	通道3输出比较模式选择 请参见TIM1_CCM1寄存器的OC1M[2:0]说明

3	OC3PE	通道 3 输出比较预装载使能 请参见 TIM1_CCM1 寄存器的 OC1PE 说明
2	OC3FE	通道3输出比较快速使能 请参见TIM1_CCM1寄存器的OC1FE说明
1:0	保留	写入无效

#### 14.5.9 TIM1 比较使能寄存器 (TIM1\_CCEN)

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
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

位/位域	名称	描述
31:14	保留	写入无效
13	CC4P	通道4比较极性 请参见CC1P说明
12	CC4E	通道4比较输出使能 请参见CC1E说明
11	CC3NP	通道3比较互补极性 请参见CC1NP说明
10	CC3NE	通道3比较互补输出使能 请参见CC1NE说明
9	CC3P	通道3比较输极性 请参见CC1P说明

8	CC3E	通道3比较使能 请参见CC1E说明
7	CC2NP	通道2比较互补极性 请参见CC1NP说明
6	CC2NE	通道2比较互补输出使能 请参见CC1NE说明
5	CC2P	通道2比较极性 请参见CC1P说明
4	CC2E	通道2比较使能 请参见CC1E说明
3	CC1NP	通道1比较互补极性 0: OC1N高电平有效 1: OC1N低电平有效 <i>注意: 此位在具有互补输出的通道上具有预装载功能。如果TIM1_CR2寄存器中的CC_PRECR位置1, 则仅当生成换相事件时, CC1NP有效位从预装载位获取新值。</i>
2	CC1NE	通道1比较互补输出使能 0: 禁止 1: 使能 (OC1N输出具体取决于该位和TIM1_BDT寄存器的MOEN、OSSI、OSSR、OIS1N位的组合) <i>注意: 此位将在互补输出的通道上进行预装载。如果TIM1_CR2寄存器中的CC_PRECR位置1, 则仅当发生换相事件时, CC1NE有效位才会从预装载位获取新值。</i>
1	CC1P	通道1比较极性 0: OC1高电平有效 1: OC1低电平有效 <i>注意: 此位在具有互补输出的通道上具有预装载功能。如果TIM1_CR2寄存器中的CC_PRECR位置1, 则仅当发生换向事件时, CC1P有效位从预装载位获取新值。</i>
0	CC1E	通道1比较使能

0：禁止

1：使能（OC1输出具体取决于该位和TIM1\_BDT寄存器的MOEN、OSSI、OSSR、OIS1位。详见表:互补通道OCx和OCxN的输出控制位（MOEN位为1）和表:具有断路功能的互补通道OCx和OCxN的输出控制位（MOEN位为0））

注意：此位可在互补输出的通道上进行预装载。如果TIM1\_CR2寄存器中的CC\_PRECR位置1，则仅当生成换相事件时，CC1E有效位才会从预装载位获取新值。

#### 14.5.10 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
Res.															
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
31:16	保留	写入无效
15:0	CNT[15:0]	计数器值

#### 14.5.11 TIM1 预分频寄存器（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	写入无效
15:0	PSC[15:0]	预分频值 计数器时钟频率（CK_CNT）等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ 。

#### 14.5.12 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	写入无效
15:0	ARR[15:0]	自动重载值 当自动重载值为空时，计数器不工作。

#### 14.5.13 TIM1 重复计数器寄存器（TIM1\_RCR）

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.								REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	写入无效

7:0                  REP[7:0]                  重复计数器值

重复计数器定义了更新事件产生的频率。每次重复计数器递减计数到0时，将生成一个更新事件，此时重复计数器会加载 REP[7:0]的值，并重新开始计数。

#### 14.5.14 TIM1 比较寄存器 1 (TIM1\_CC1)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1[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	写入无效
15:0	CC1[15:0]	通道1的比较值 <ul style="list-style-type: none"> <li>当TIM1_CCM1寄存器中的OC1PE位为1时，使能预装载功能，该位域的新值将在更新事件后生效</li> <li>当TIM1_CCM1寄存器中的OC1PE位为0时，该位域的新值立即生效</li> </ul>

#### 14.5.15 TIM1 比较寄存器 2 (TIM1\_CC2)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC2[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	写入无效
15:0	CC2[15:0]	通道2的比较值 请参见TIM1比较寄存器1（TIM1_CC1）说明

#### 14.5.16 TIM1 比较寄存器 3（TIM1\_CC3）

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
CC3[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	写入无效
15:0	CC3[15:0]	通道3的比较值 请参见TIM1比较寄存器1（TIM1_CC1）说明

#### 14.5.17 TIM1 比较寄存器 4（TIM1\_CC4）

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
CC4[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	写入无效

15:0      CC4[15:0]      通道4的比较值  
请参见TIM1比较寄存器1（TIM1\_CC1）说明

#### 14.5.18 TIM1 断路和死区寄存器（TIM1\_BDT）

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOEN	AOEN	Res.	BKEN	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
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	写入无效
15	MOEN	主输出使能 此位仅对配置为输出的通道有效。当断路发生时，此位自动异步清0。此位由软件置1。 0：禁止OCx和OCxN输出或被强制为空闲状态（由OSSI位域和TIM1_CCEN寄存器的CCxE/CCxNE位决定） 有关配置信息，请参见 <a href="#">表：具有断路功能的互补通道OCx和OCxN的输出控制位（MOEN位为0）</a> 1：如果OCx（OCxN）输出使能位CCxE（CCxNE）置1，则使能OCx（OCxN）输出 <i>注意：只有将BIF位清0后，才能将MOEN位置1。</i>
14	AOEN	自动输出使能 0：MOEN位只能由软件置1 1：MOEN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无效）
13	保留	写入无效
12	BKEN	断路功能使能位

		<p>0: 禁止</p> <p>1: 使能</p>
11	OSSR	<p>运行模式下的关闭状态选择（MOEN位为1时）</p> <p>通道配置为输出模式且具有互补输出时，且互补的两个通道其中之一使能时，该位定义未使能输出通道的输出状态。互补输出的有关配置，请参见表：互补通道OCx和OCxN的输出控制位（MOEN位为1）</p> <p>0: 禁止OCx（OCxN）输出（控制由GPIO接管）</p> <p>1: 使能OCx（OCxN）输出并将其设为无效电平（输出仍由定时器控制）</p>
10	OSSI	<p>空闲模式下的关闭状态选择（MOEN位为0）</p> <p>通道配置为输出模式且具有互补输出时，此位生效。有关配置信息，请参见表：具有断路功能的互补通道OCx和OCxN的输出控制位（MOEN位为0）</p> <p>0: 禁止OCx（OCxN）输出（控制由GPIO接管）</p> <p>1: OCx/OCxN输出强制为其无效电平，之后在死区时间后将其强制为空闲电平（输出仍由定时器控制）</p>
9:8	LOCK[1:0]	<p>锁定配置</p> <p>00: 关闭锁定（不对任何位提供写保护）</p> <p>01: 锁定级别1，此时无法对以下寄存器相应位执行写操作：</p> <ul style="list-style-type: none"> <li>– TIM1_CR2寄存器中的OISx和OISxN位</li> <li>– TIM1_BDT寄存器中的AOEN、BKEN和DTG[7:0]位域</li> <li>– TIM1_AF1寄存器</li> </ul> <p>10: 锁定级别2，此时无法对以下寄存器相应位执行写操作：</p> <ul style="list-style-type: none"> <li>– 锁定级别1中适用的各位域</li> <li>– OSSR和OSSI位</li> <li>– TIM1_CCEN寄存器中的CCxP/CCxNP位</li> </ul> <p>11: 锁定级别3，此时无法对以下寄存器相应位执行写操作：</p> <ul style="list-style-type: none"> <li>– 锁定级别2中适用的各位域</li> <li>– TIM1_CCMx寄存器中的OCxPE和OCxM[2:0]位域</li> </ul> <p>注意：复位后只能对LOCK位执行一次写操作。对TIM1_BDT寄存器执行写操作后其中的内容不能被修改，直到下一次复位。</p>
7:0	DTG[7:0]	配置死区发生器

定义互补输出之间的死区时间。其中DT为死区时间。

DTG[7:5]=0xx:  $DT = DTG[7:0] \times t_{DTG}$ , 其中 $t_{DTG} = t_{DTS}$ 。

DTG[7:5]=10x:  $DT = (64 + DTG[5:0]) \times t_{DTG}$ , 其中 $t_{DTG} = 2 \times t_{DTS}$ 。

DTG[7:5]=110:  $DT = (32 + DTG[4:0]) \times t_{DTG}$ , 其中 $t_{DTG} = 8 \times t_{DTS}$ 。

DTG[7:5]=111:  $DT = (32 + DTG[4:0]) \times t_{DTG}$ , 其中 $t_{DTG} = 16 \times t_{DTS}$ 。

示例: 如果 $t_{DTS} = 125\text{ns}$  (8MHz), 则可能的死区值为:

- 当DTG[7:5]=0xx, 0到15875ns (步长为125ns)
- 当DTG[7:5]=10x, 16 $\mu\text{s}$ 到31750ns (步长为250ns)
- 当DTG[7:5]=110, 32 $\mu\text{s}$ 到63 $\mu\text{s}$  (步长为1 $\mu\text{s}$ )
- 当DTG[7:5]=111, 64 $\mu\text{s}$ 到126 $\mu\text{s}$  (步长为2 $\mu\text{s}$ )

*注意: 使用互补通道时, 需先配置死区时间, 再配置其它位域。*

#### 14.5.19 TIM1 配置寄存器 (TIM1\_CFG)

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_CLR
															rw

位/位域	名称	描述
31:1	保留	写入无效
0	OCREF_CLR	OC_CLR源选择 0: COMP1输出连接到OC_CLR输入 1: COMP2输出连接到OC_CLR输入

#### 14.5.20 TIM1 轮换功能寄存器 1 (TIM1\_AF1)

偏移地址: 0x60

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.				BKCMP2P	BKCMP1P	BKINP	Res.	LOCKUP_LOCK	Res.				BK_CMP2E	BK_CMP1E	BKINE
				rw	rw	rw		rw					rw	rw	rw

位/位域	名称	描述
31:12	保留	写入无效
11	BKCMP2P	COMP2输入极性选择 0: COMP2输入低电平有效 1: COMP2输入高电平有效
10	BKCMP1P	COMP1输入极性选择 0: COMP1输入低电平有效 1: COMP1输入高电平有效
9	BKINP	BKIN输入极性选择 0: BKIN输入低电平有效 1: BKIN输入高电平有效
8	保留	写入无效
7	LOCKUP_LOCK	用于控制LOCKUP与TIM1断路输入的连接。 软件置1，写0无效，直到下次复位时清0 0: LOCKUP输出与TIM1断路输入断开 1: LOCKUP输出与TIM1断路输入连接
6:3	保留	写入无效
2	BKCMP2E	COMP2断路输入使能 0: 禁止 1: 使能
1	BKCMP1E	COMP1断路输入使能 0: 禁止 1: 使能

---

0	BKINE	BKIN断路输入使能
		0: 禁止
		1: 使能

## 15 通用定时器（TIM3）

### 15.1 简介

通用定时器 TIM3 由一个 16 位自动重载计数器构成。可用于测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（比较输出、单脉冲输出和 PWM）等多种用途。

通用定时器 TIM3 和其他定时器结合使用，可实现定时器互联功能。

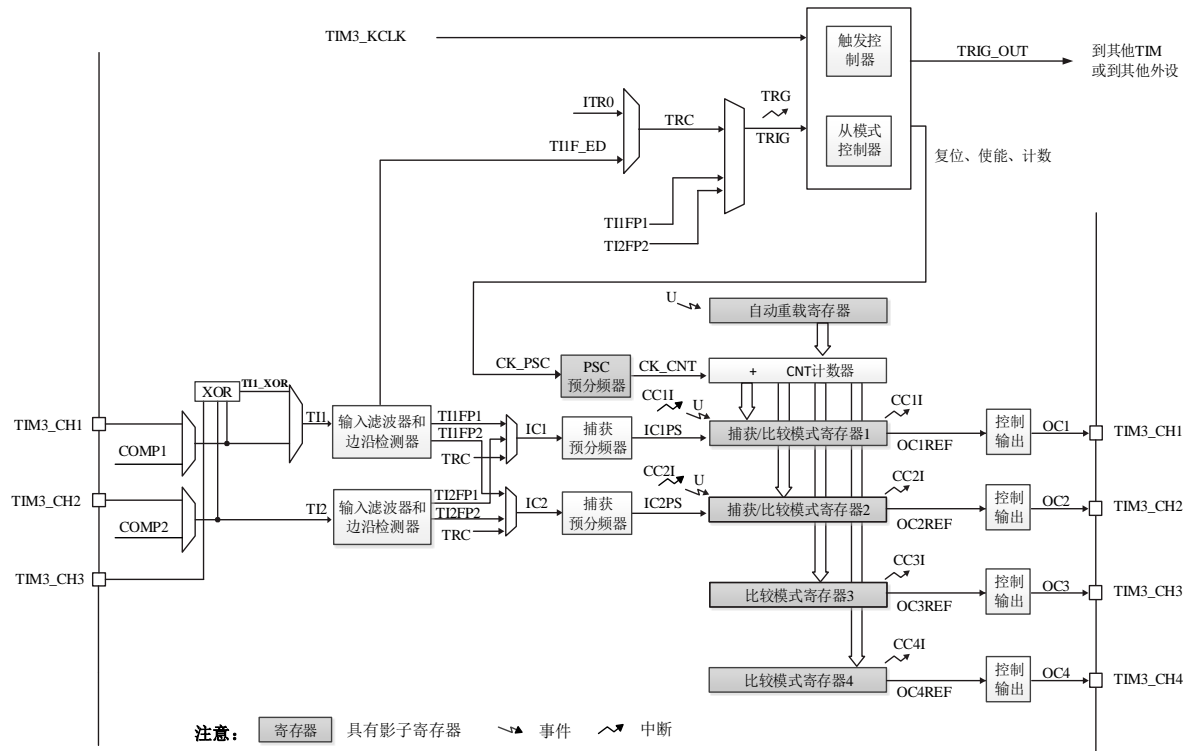
### 15.2 TIM3 主要特性

- 16 位递增自动重载计数器
- 可编程预分频器
- 2 个独立通道输入捕获，4 个独立通道输出比较、PWM 生成和单脉冲模式输出
- 可实现定时器的启动、停止、初始化
- 可定时触发 ADC、可多个定时器互联

## 15.3 TIM3 功能描述

### 15.3.1 TIM3 框图

图 15-1 TIMx 框图 (x=3)



### 15.3.2 TIM3 引脚和内部信号

表 15-1 TIM3 引脚

引脚名称	信号类型	说明
TIM3_CH1	输入/输出	通道 1 输入输出引脚
TIM3_CH2	输入/输出	通道 2 输入输出引脚
TIM3_CH3	输入/输出	通道 3 输入输出引脚
TIM3_CH4	输出	通道 4 输出引脚

表 15-2 TIM3 内部信号

信号名称	信号类型	说明
TIM3_KCLK	输入	TIM3 外设时钟，来自 RCC
ITR	输入	内部触发信号
TRIG_OUT	输出	内部触发输出信号

### 15.3.3 PSC 预分频器

PSC 预分频器对 CNT 计数器的输入时钟进行分频，分频系数介于 1 和 32768 之



间。TIMx\_PSC 寄存器中的 PSC[3:0]位域用于配置分频值，实际分频系数为分频值  $2^{PSC[3:0]}$ 。

TIMx\_PSC 寄存器具有缓冲功能（影子寄存器），每次写入的预分频值，会在下一次更新事件发生时生效。

以下各图，举例显示了预分频器更新生效的时序。

图 15-2 预分频器分频系数由 1 变为 2 时生效的时序图

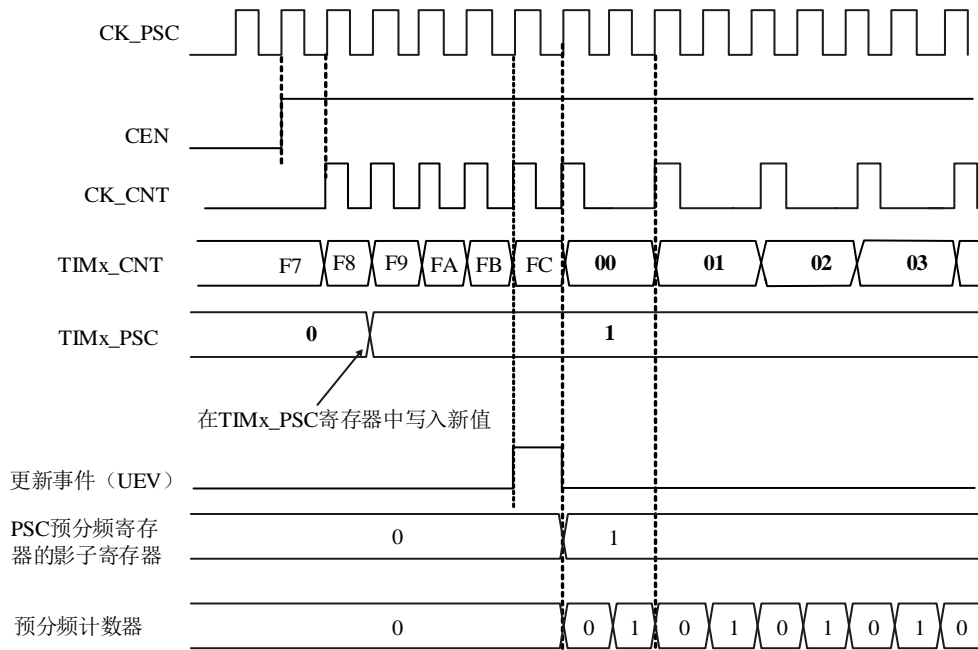
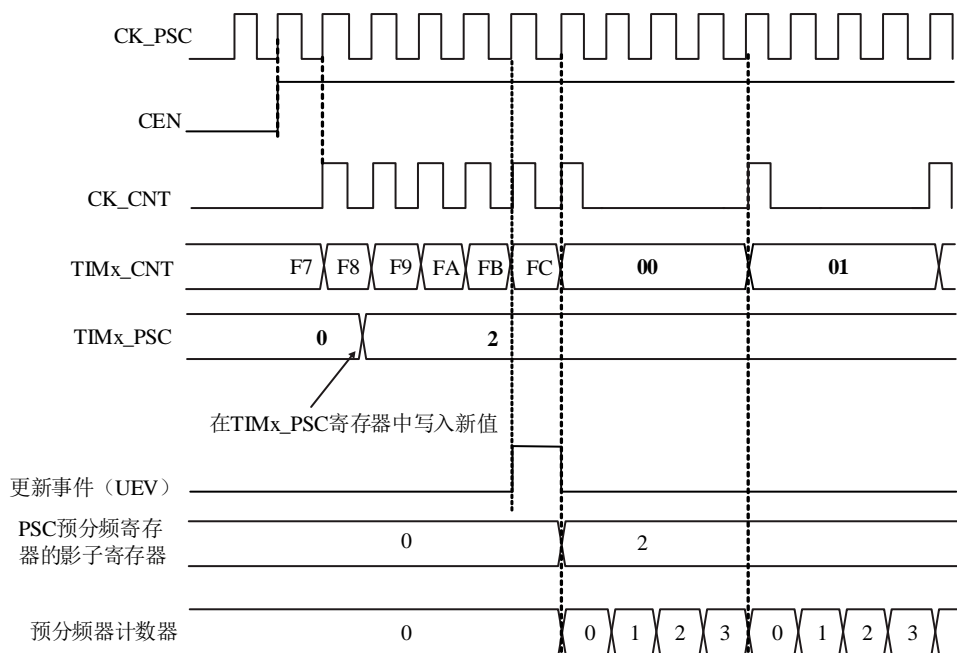


图 15-3 预分频器分频系数由 1 变为 4 时的计数器时序图



#### 15.3.4 计数器

TIMx 内置一个 16 位的递增计数器。

计数器配置支持动态修改，在计数器运行时，TIMx\_CNT 寄存器、TIMx\_ARR 寄存器和 TIMx\_PSC 寄存器均可进行读写操作。

自动重载值寄存器 (TIMx\_ARR) 中的 ARR[15:0] 位域可用于配置计数器的重载值，该寄存器具有缓冲功能 (影子寄存器)。

如果 TIMx\_CR1 寄存器中的 ARPE 位为 0 时，影子寄存器的更新不受更新事件的影响，TIMx\_ARR 寄存器的值会立即更新。如果 ARPE 位为 1 时，当发生更新事件，则 TIMx\_ARR 寄存器中的值将更新到影子寄存器中。

如果将 TIMx\_CR1 寄存器中的 UDIS 位置 1，可禁止更新 (UEV) 事件，避免对影子寄存器的更新。

当 UDIS 位为 0 时，使能更新事件，此时 URS 位配置为 1，则将 UG 位置 1，会生成更新事件，计数器及预分频计数器均被初始化，但不会将 TIMx\_SR 寄存器中的 UIF 置位。

通过 URS 可以选择更新事件的请求源：

- 0：使能时，以下事件都会产生更新事件。
  - 计数器上溢；
  - 将 TIMx\_EVTG 寄存器中的 UG 位置 1；
  - 通过从模式控制器生成的更新事件 (复位模式)。
- 1：使能时，计数器上溢会产生更新事件。

计数器从 0 开始递增计数，每个 CK\_CNT 周期自动加 1，当计数值等于 TIMx\_ARR 寄存器的值时，将产生计数器上溢事件，TIMx\_SR 寄存器的 UIF 标志将自动置 1，同时计数器将被清 0 并重新开始计数。

发生更新事件时，TIMx\_SR 寄存器中的 UIF 位将自动置 1，并对以下寄存器进行更新：

- 预装载值 (TIMx\_ARR 寄存器) 更新到自动重载影子寄存器中
- 预装载值 (TIMx\_PSC 寄存器) 重新加载到 PSC 预分频器的影子寄存器中

以下各图，显示了递增计数模式下，计数器发生上溢更新事件时的时序，TIMx\_ARR 寄存器配置为 0x36：

图 15-4 递增计数模式下配置预分频器为 1 分频上溢事件时序图

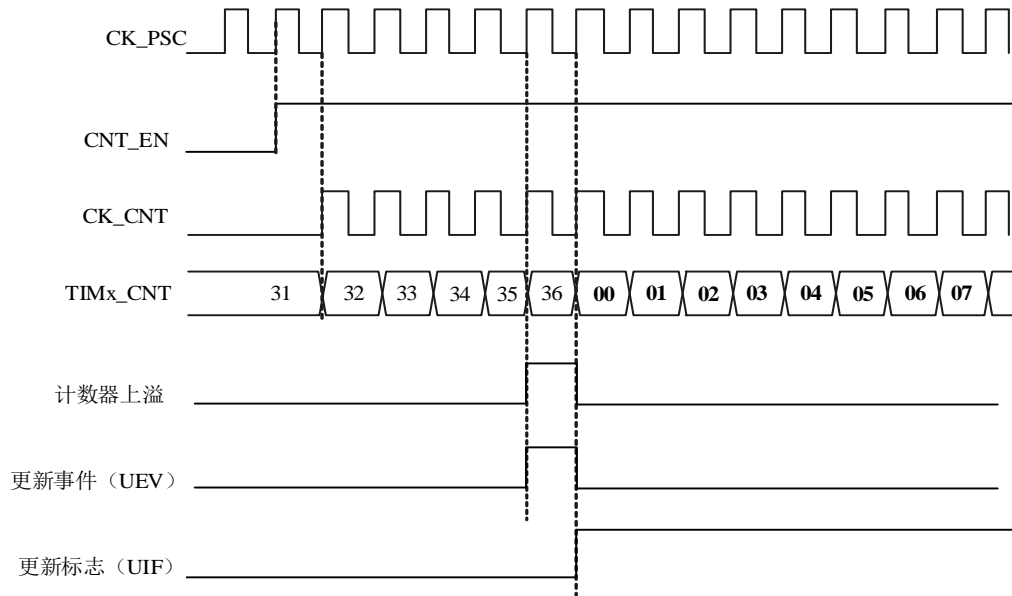


图 15-5 递增计数模式下配置预分频器为 4 分频上溢事件时序图

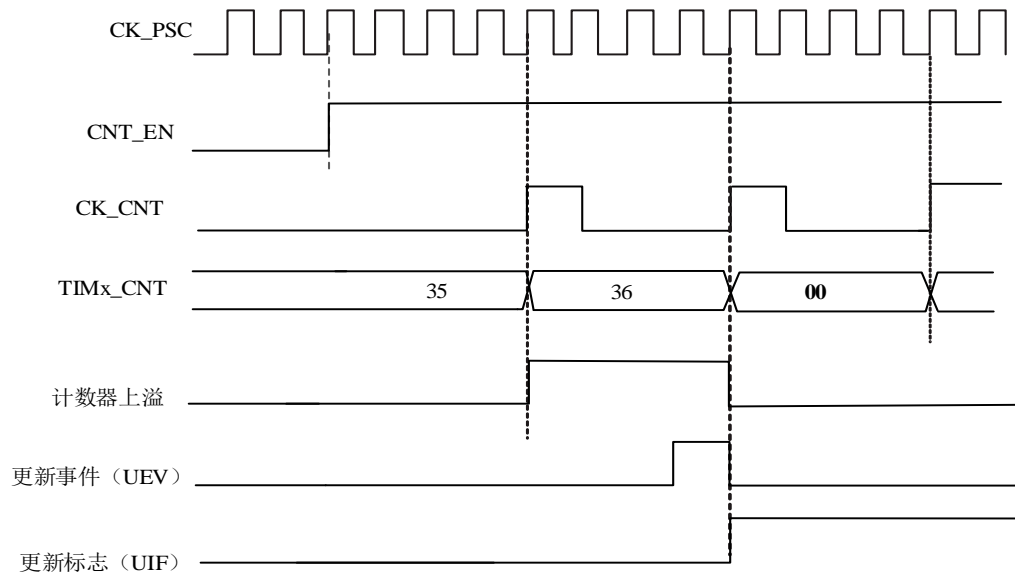


图 15-6 递增计数模式下计数器时序图，ARPE=0 时更新事件

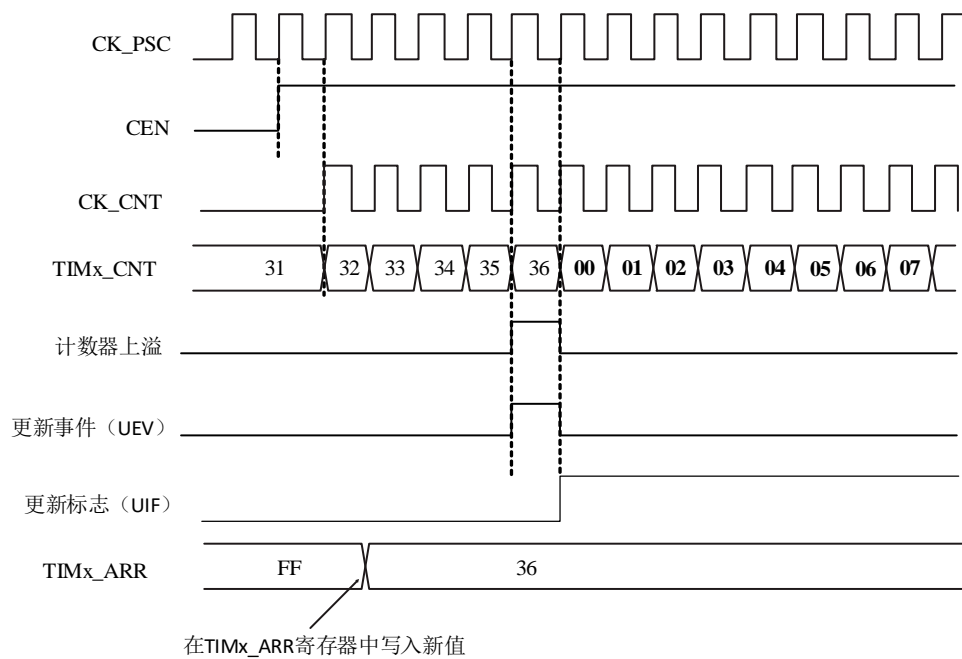
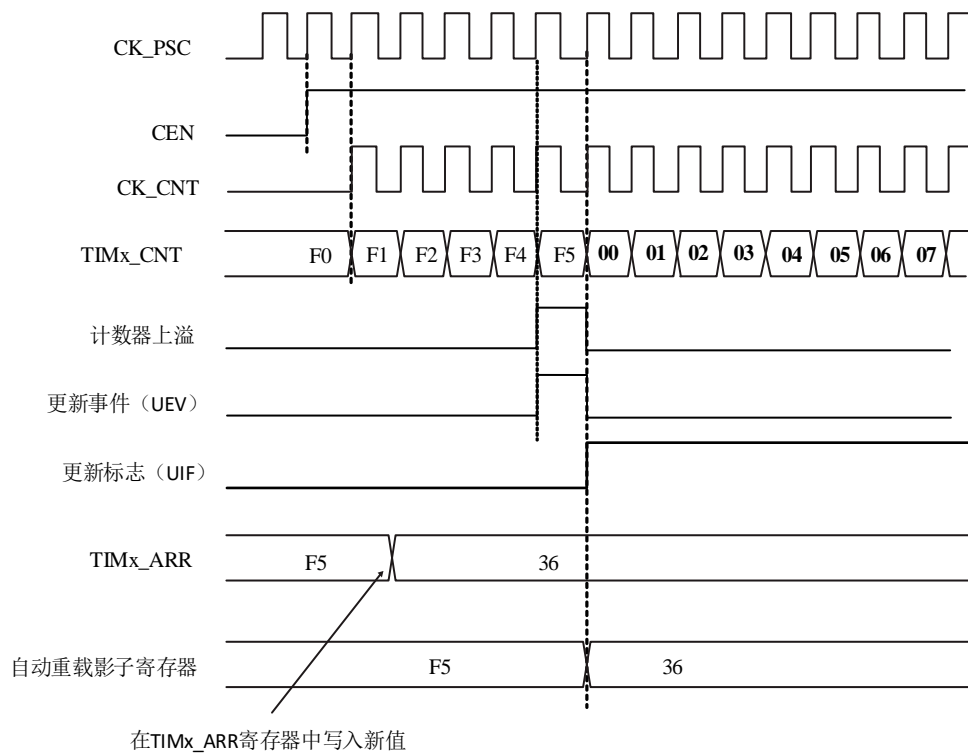


图 15-7 递增计数模式下计数器时序图，ARPE=1 时更新事件



### 15.3.5 计数时钟选择

计数器的计数时钟可由下列信号源提供：

- 内部时钟源：TIMx 外设时钟 (TIMx\_KCLK)；

- 时钟模式 1：TRIG 触发输入作为计数时钟。

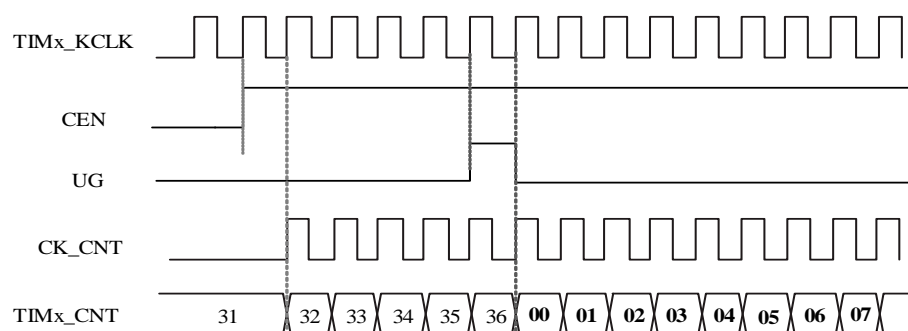
以上时钟源可通过配置 TIMx\_PSC 寄存器的值进行预分频后，作为计数时钟（CK\_CNT）。

### 内部时钟源

当配置 TIMx\_SMC 寄存器中的 SM\_SEL[2:0]位域为 0xx 时，计数器的计数时钟源为 TIMx\_KCLK，来自 RCC，即 TIMx\_PCLK。

下图显示了选择内部时钟源时的计数时序图。

图 15-8 选择内部时钟源时的计数时序图



### 时钟模式 1

当配置 TIMx\_SMC 寄存器中的 SM\_SEL[2:0]位域为 111 时，计数器的计数时钟源为 TRIG 触发输入信号，可通过配置 TIMx\_CCEN 寄存器中的 CCxP 和 CCxNP 位来选择上升沿或下降沿时计数。

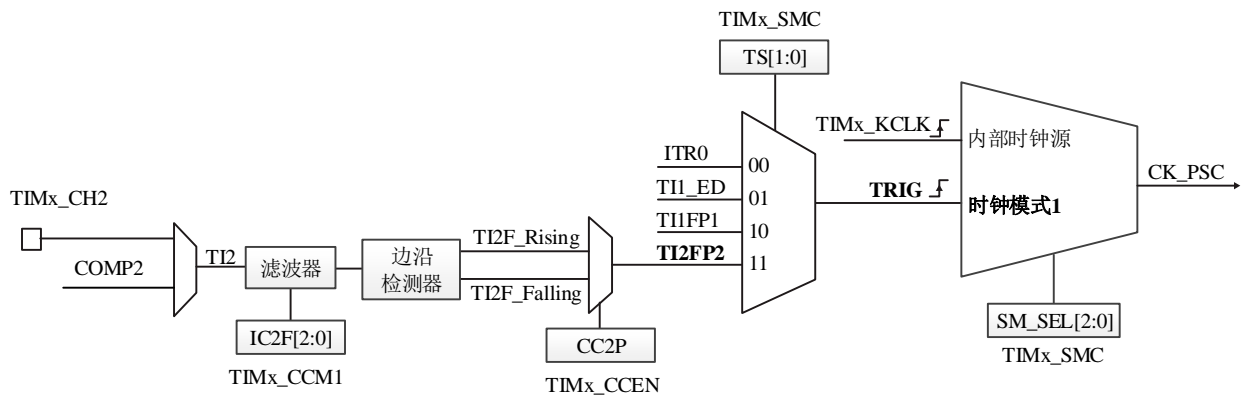
通过配置 TIMx\_SMC 寄存器中的 TS[1:0]位域可选择不同的 TRIG 信号：

- ITR0;
- TI1F\_ED;
- TI1FP1;
- TI2FP2;

TIx 信号与 TIMx\_KCLK 为异步信号，故 TIx 的最大频率为 TIMx\_KCLK 频率的 1/2。

当选择 TI1F\_ED 作为 TRIG 信号时，其信号最大频率为 TIMx\_KCLK 频率的 1/4。

图 15-9 选择 TI2 作为时钟模式 1 触发输入的连接示例（上升沿计数）

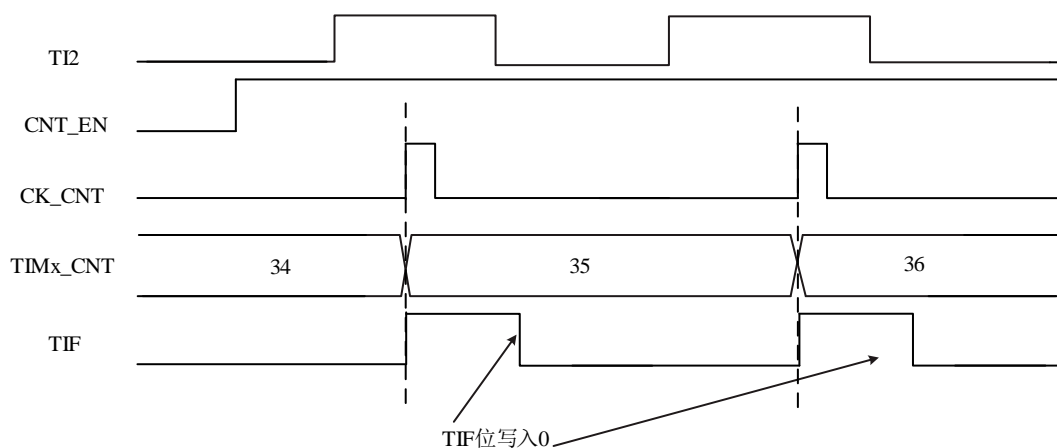


例如，要使计数器在 TI2 输入的上升沿时计数，其步骤如下：

- 1) 选择 TI2 触发源：配置 TIMx\_TISEL 寄存器中的 TI2SEL 位域为 0；
- 2) 配置通道 2 为输入模式：配置 TIMx\_CCM1 寄存器中的 CC2S[1:0] 位域为 01；
- 3) 设置输入滤波器：配置 TIMx\_CCM1 寄存器中 IC2F[2:0] 位域（如果不需要滤波器，请保持 IC2F[2:0] 为 000）；
- 4) 极性选择（上升沿有效）：配置 TIMx\_CCEN 寄存器中的 CC2P 位和 CC2NP 位为 0；
- 5) 选择时钟模式 1：配置 TIMx\_SMC 寄存器中的 SM\_SEL[2:0] 位域为 111；
- 6) 选择 TI2 为触发输入源：配置 TIMx\_SMC 寄存器中的 TS[1:0] 位域为 11；
- 7) 使能计数器：配置 TIMx\_CR1 寄存器中的 CEN 位为 1；

当 TI2 出现上升沿时，计数器便会计数一次，TIMx\_SR 寄存器中的 TIF 被自动置 1。由于受信号同步和滤波器的影响，TI2 的上升沿与计数器时钟之间存在延迟。

图 15-10 选择时钟模式 1 时的时序图

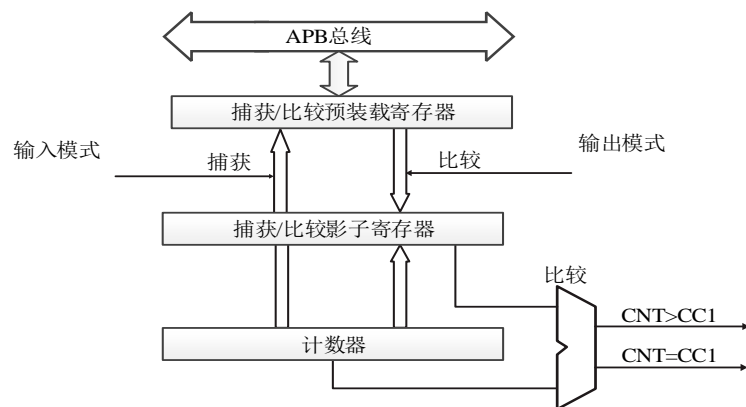


### 15.3.6 捕获/比较通道

捕获/比较通道包含：

- 捕获/比较寄存器（包括一个影子寄存器）
- 输入捕获通道（数字滤波、多路复用和预分频器）
- 输出比较通道（比较器和输出控制）

图 15-11 捕获/比较通道



#### 工作模式 0

当 TIMx\_CR1 寄存器中的 MODE 配置为 0 时，通道 1 和 2 支持 16bit 精度的输入捕获和输出比较。

在捕获模式下，发生捕获时，将计数器 CNT[15:0]的当前值加载到捕获/比较寄存器 TIMx\_CCx[15:0]（x=1、2）中。

在比较模式下，捕获/比较寄存器 TIMx\_CCx[15:0]（x=1、2）的内容将加载到影子寄存器中，然后将影子寄存器的内容与计数器 CNT[15:0]进行比较。

#### 工作模式 1

当 TIMx\_CR1 寄存器中的 MODE 配置为 1 时，通道 1 和 2 支持 8bit 精度的输入捕获，通道 1、2、3、4 支持 8bit 精度的输出比较。

在捕获模式下，发生捕获时，将计数器 CNT[7:0]的当前值加载到捕获/比较寄存器 TIMx\_CCx[7:0]（x=1、2）中。

在比较模式下，捕获/比较寄存器 TIMx\_CCx[7:0]（x=1、2、3、4）的内容将加载到影子寄存器中，然后将影子寄存器的内容与计数器 CNT[7:0]进行比较。

**注意：** 在工作模式 1 下，ARR 和 CNT 配置不得高于 0x00FF。

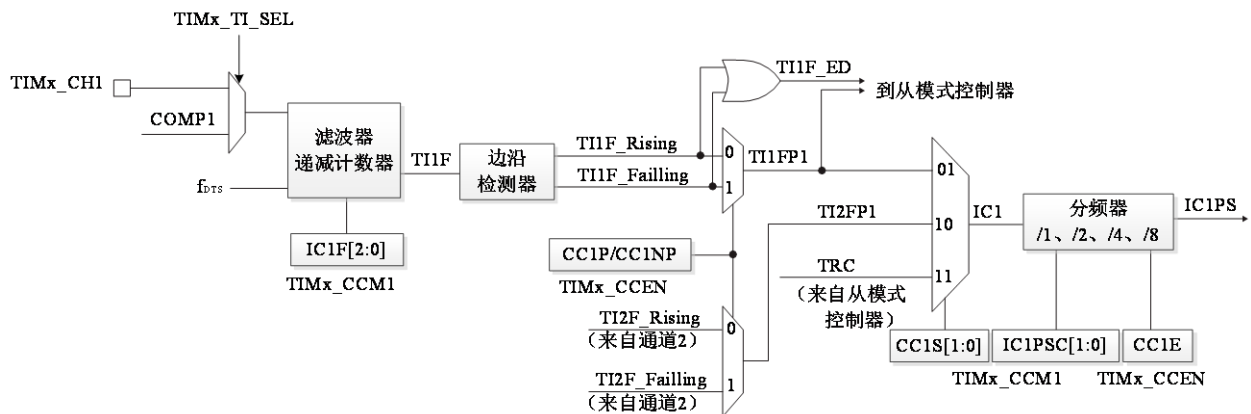
#### 输入捕获模式

TIx 的输入源可通过 TIMx\_TISEL 寄存器的相关位域来配置。输入捕获源选择可通过 TIMx\_CCMx 寄存器中的 CCxS[1:0] 位域来配置。以通道 1 为例, 有 TRC、TI1FP1、TI2FP1。

对 TIx 输入经采样滤波后产生 TIxF 信号, 随后通过一个带极性选择的边沿检测器后产生 ICx 信号。该信号通过捕获预分频器进入捕获/比较寄存器。

各通道的具体组成, 详见下列各图:

图 15-12 捕获/比较通道的输入连接图 (通道 1)



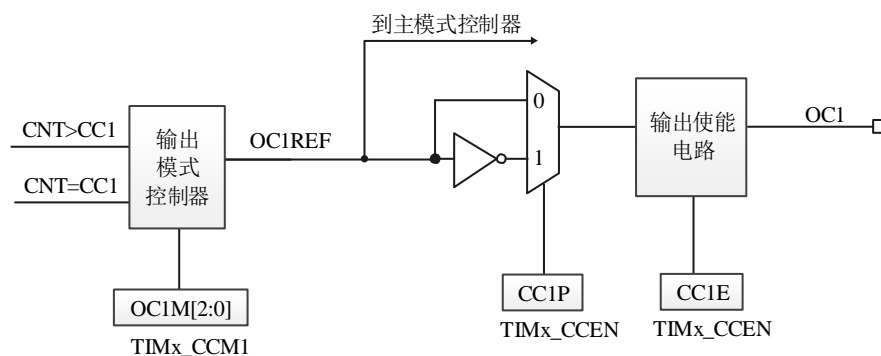
### 输出比较模式

输出通道 1 到通道 4 可输出到芯片外部。

输出比较模式分为:

- 比较输出模式, 详见 [比较输出模式](#);
- 强制输出模式, 详见 [强制输出模式](#);
- PWM 模式, 详见 [PWM 模式](#);
- 单脉冲模式, 详见 [单脉冲模式](#)。

图 15-13 捕获/比较通道的输出连接图 (通道 1)





### 15.3.7 输入捕获模式

在输入捕获模式下，有如下方式生成捕获事件：

- 检测到 ICx 信号的有效触发沿；
- 将 TIMx\_EVTG 寄存器中相应的 CCxG 位置 1。

当产生输入捕获事件，此时会有如下行为：

- TIMx\_CCx 寄存器会捕获当前计数器的值；
- TIMx\_SR 寄存器中的 CCxIF 位置 1；
- 如果此时 TIMx\_DIER 寄存器中的 CCxIE 位为 1，则触发中断。

CCxIF 位为 1 时，再次检测到捕获事件，将产生捕获溢出事件，寄存器 CCxOF 位会自动置 1，同时 TIMx\_CCx 寄存器会重新捕获当前计数器的值。

当 CCxS[1:0]位域的值不为 00（输入捕获模式）时，TIMx\_CCx 寄存器将处于只读状态。

CCxIF 清 0 的方式有：

- 向 CCxIF 写入 0；
- 读取 TIMx\_CCx 寄存器中的已捕获的数据。

例如：在 TI1 输入信号上检测到上升沿时将计数器的值捕获到 TIMx\_CC1 寄存器中。具体配置步骤如下：

- 1) 将 TI1 映射到 IC1：配置 TIMx\_CCM1 寄存器中的 CC1S[1:0]位域为 01；
- 2) 设置输入滤波器：配置 TIMx\_CCM1 寄存器中的 IC1F[2:0]位域为 011；
- 3) 选择 TI1 为上升沿有效：配置 TIMx\_CCEN 寄存器中的 CC1P 位为 0；
- 4) 使能捕获：配置 TIMx\_CCEN 寄存器中的 CC1E 位为 1。

### 15.3.8 PWM 输入模式

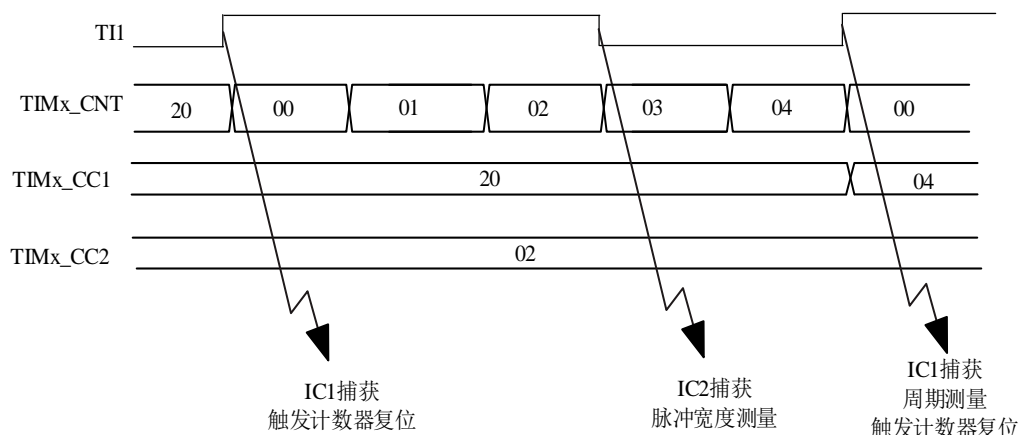
此模式是输入捕获模式的一种特例，但和输入捕获相比，有以下几点区别：

- 一个 TIx 输入信号被映射到两个 ICx 信号上；
- 这两个 ICx 信号为边沿有效，但极性相反；
- 其中一个 TIxFP 信号用作触发输入信号，而从模式控制器配置为复位模式，复位模式详见：[定时器同步从模式](#)。

例如，测量 TI1 上的 PWM 周期和占空比，其配置步骤如下：

- 1) 选择 TI1FP1 的有效极性：配置 TIMx\_CCEN 寄存器中的 CC1NP/CC1P 位为 00（上升沿有效）；
- 2) 将 TI1FP1 映射到 IC1 上：配置 TIMx\_CCM1 寄存器中的 CC1S[1:0]位域为 01；
- 3) 选择 TI1FP2 的有效极性：配置 TIMx\_CCEN 寄存器中的 CC2NP/CC2P 位为 01（下降沿有效）；
- 4) 将 TI1FP2 映射到 IC2 上：配置 TIMx\_CCM1 寄存器中的 CC2S[1:0]位域为 10；
- 5) 选择从模式控制器的触发输入：配置 TIMx\_SMC 寄存器中的 TS[1:0]位域为 10（选择 TI1FP1）；
- 6) 选择从模式控制器为复位模式：配置 TIMx\_SMC 寄存器中的 SM\_SEL[2:0]位域为 100；
- 7) 使能捕获：配置 TIMx\_CCEN 寄存器中的 CC1E 位和 CC2E 位为 1。

图 15-14 PWM 输入模式的时序



### 15.3.9 比较输出模式

该模式包含：匹配输出有效电平、匹配输出无效电平和翻转模式。通过 TIMx\_CCMx 寄存器中的 OCxPE 位，可使能或禁止 TIMx\_CCx 寄存器的预装载功能。

当 TIMx\_CCx 的影子寄存器与 TIMx\_CNT 寄存器的值匹配时：

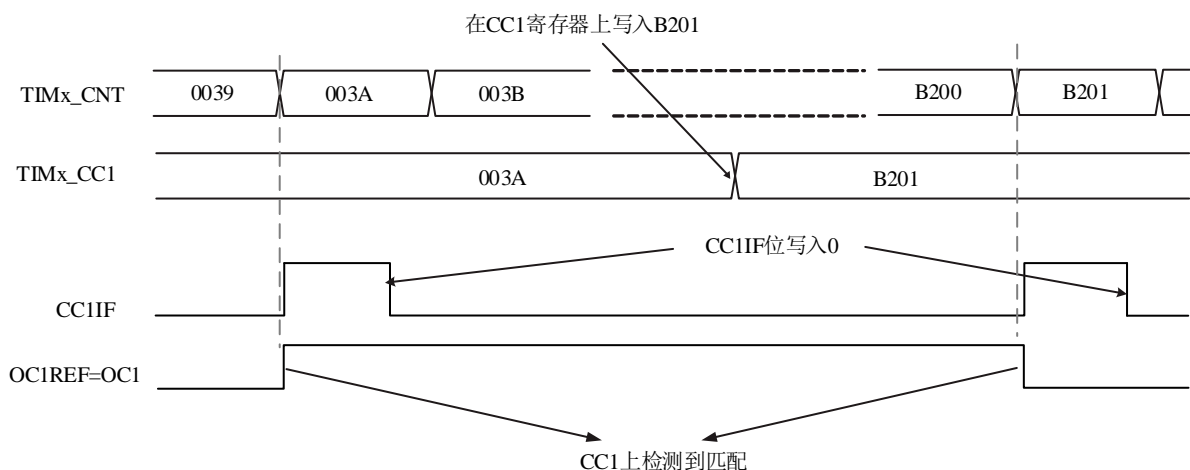
- TIMx\_SR 寄存器中的 CCxIF 位置 1；
- 输出极性由 TIMx\_CCEN 寄存器中的 CCxP 位决定；
- 如果此时 TIMx\_DIER 寄存器中的 CCxIE 位为 1，则触发中断。

配置为比较输出模式时，其步骤如下：

- 1) 配置 TIMx\_ARR 寄存器和 TIMx\_CCx 寄存器；
- 2) 选择输出模式。例如：
  - 配置 TIMx\_CCMx 寄存器中的 OCxM[2:0]位域为 011（当 TIMx\_CNT 寄存器的值与 TIMx\_CCx 寄存器的值匹配时，OCx 输出发生翻转）；
  - 禁止预装载寄存器：配置 TIMx\_CCMx 寄存器中的 OCxPE 位为 0；
  - 选择高电平有效：配置 TIMx\_CCEN 寄存器中的 CCxP 为 0；
  - 使能输出：配置 TIMx\_CCEN 寄存器中的 CCxE 为 1；
- 3) 使能计数器：配置 TIMx\_CR1 寄存器中的 CEN 位为 1。

当未使能预装载寄存器（TIMx\_CCMx 寄存器中的 OCxPE 位为 0）时，可实时更改 TIMx\_CCx 寄存器的值，以控制输出波形。

图 15-15 比较输出模式，翻转 OC1（OCxPE 位为 0）



### 15.3.10 强制输出模式

在强制输出模式下，可配置输出信号为有效或无效电平，此时输出信号不受 TIMx\_CCx 影子寄存器和计数器之间的比较结果影响。

- 配置 TIMx\_CCMx 寄存器中的 OCxM[2:0] 位域为 101，可将输出信号（OCxREF）强置为高电平（OCxREF 始终为高电平有效）；
- 配置 TIMx\_CCMx 寄存器中的 OCxM[2:0] 位域为 100，可将输出信号（OCxREF）强置为低电平。

OCx 输出信号的极性由 TIMx\_CCEN 寄存器中的 CCxP 位决定。

该模式下，TIMx\_CCx 影子寄存器与计数器之间仍在进行比较，当产生匹配时，

相应标志会被置位。

### 15.3.11 PWM 模式

PWM 模式生成的信号，其频率由 TIMx\_ARR 寄存器的值决定，其占空比由 TIMx\_CCx 寄存器的值决定。

每个输出通道都可以独立选择 PWM 模式输出：

- 通过 TIMx\_CCMx 寄存器中的 OCxM[2:0]位域来配置。
  - PWM 模式 1 (OCxM[2:0]位域为 110)：当 TIMx\_CNT < TIMx\_CC1，通道 1 输出为有效电平，否则为无效电平；
  - PWM 模式 2 (OCxM[2:0]位域为 111)：当 TIMx\_CNT < TIMx\_CC1，通道 1 输出为无效电平，否则为有效电平；
- 配置 TIMx\_CCMx 寄存器中的 OCxPE 位置 1 使能相应比较寄存器的预装载功能 (TIMx\_CCx)；
- 配置 TIMx\_CR1 寄存器中的 ARPE 位置 1 使能自动重载寄存器的预装载功能 (TIMx\_ARR)。

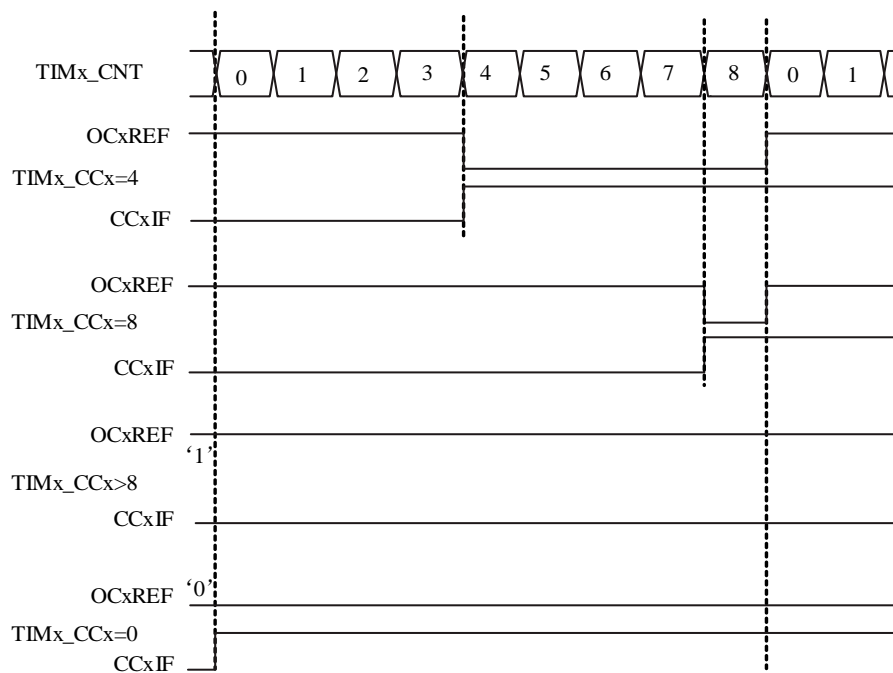
当使能预装载功能时，在发生更新事件时，预装载寄存器的值才会加载到影子寄存器中，所以启动计数器前，建议将 TIMx\_EVTG 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 的极性可通过 TIMx\_CCEN 寄存器的 CCxP 位来配置。

下面是一个 PWM 模式 1 的例子。当 TIMx\_CNT < TIMx\_CCx，OCxREF 信号输出高电平，否则为低电平。如果 TIMx\_CCx 寄存器的值大于 TIMx\_ARR 寄存器的值，则 OCxREF 保持为 1。如果 TIMx\_CCx 寄存器的值为 0，则 OCxREF 保持为 0。

例如：TIMx\_ARR 寄存器的值为 8 时 PWM 波形。

图 15-16 PWM 波形 (TIMx\_ARR 寄存器为 8)



### 15.3.12 单脉冲模式

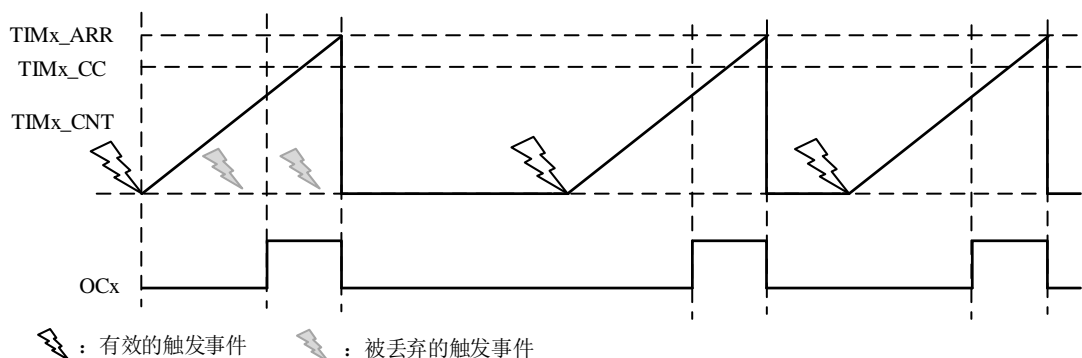
单脉冲模式(OPM)是比较输出模式的一个特例。将 TIMx\_CR1 寄存器中的 OPM 位置 1, 即可选择单脉冲模式。在此模式下, 计数器接收到触发信号, 在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。在比较输出模式或 PWM 模式下生成波形。当发生下一更新事件时 (TIMx\_CNT 的值返回到 0), 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 (定时器等触发时), 必须配置  $CNT < CCx \leq ARR$  (特别注意,  $0 < CCx$ )。

启动计数后, 在一个计数周期内发生的任何触发事件均将被丢弃。如下图所示:

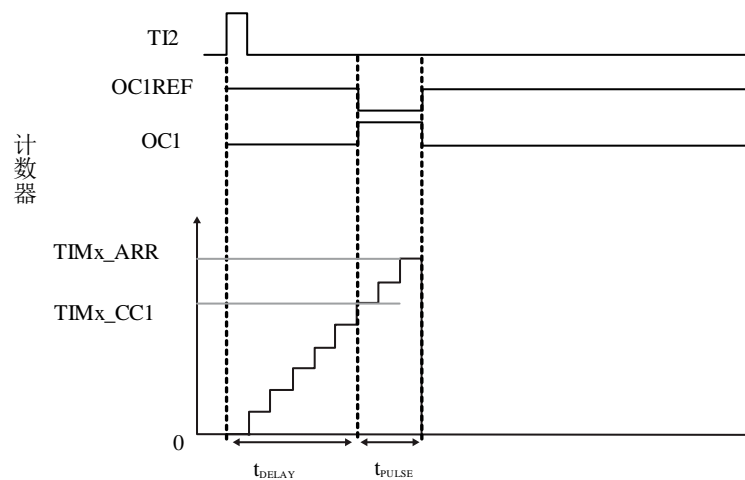
图 15-17 单脉冲输出波形



例如, 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

个宽度为  $t_{PULSE}$  的正脉冲。

图 15-18 单脉冲模式示例



使用 TI2FP2 作为触发信号：

- 1) 选择 TI2 源：配置 TIMx\_TISEL 寄存器中的 TI2\_SEL 位域；
- 2) TI2FP2 映射到 TI2：TIMx\_CCM1 寄存器中的 CC2S[2:0]位域写入 01；
- 3) 检测 TI2FP2 上升沿：TIMx\_CCEN 寄存器中的 CC2P 位和 CC2NP 位清 0；
- 4) TI2FP2 配置为从模式控制器的触发模式 (TRIG)：TIMx\_SMC 寄存器中的 TS[1:0]写入 11；
- 5) TI2FP2 启动计数器：TIMx\_SMC 寄存器中的 SM\_SEL[2:0]位域写入 110 (触发模式)。

单脉冲模式的脉冲宽度由当前时钟频率和计数器预分频器决定。

- $t_{DELAY}$  由写入 TIMx\_CC1 寄存器的值定义。
- $t_{PULSE}$  由自动重载值与比较值之差 (TIMx\_ARR - TIMx\_CC1) 来定义。
- 若产生这样的波形：信号在发生比较匹配时从 0 变 1，在计数器达到自动重载值时由 1 变为 0：
  - OC1 输出极性：CC1P 位为 1；
  - 使能 PWM 模式 1 (TIMx\_CCM1 寄存器中的 OC1M[2:0]写入 110)；
  - 如果需要，可使能预装载功能 (TIMx\_CCM1 寄存器的 OC1PE 和 TIMx\_CR1 寄存器的 ARPE 位置 1)；
  - 在 TIMx\_CC1 寄存器中写入比较值；
  - 在 TIMx\_ARR 寄存器中写入自动重载值；

- 将 UG 位置 1，产生一个更新事件，之后等待 TI2 上的外部触发事件启动计数器计数。

#### OCx 快速使能:

在单脉冲模式下，TIx 输入的边沿检测会使能计数器（CEN 位自动置 1），之后在计数器值与比较值之间发生比较操作产生输出的转换。但此操作需要一定的时钟周期，因此它限制了可得到的最小延时  $t_{\text{DELAY}}$ 。

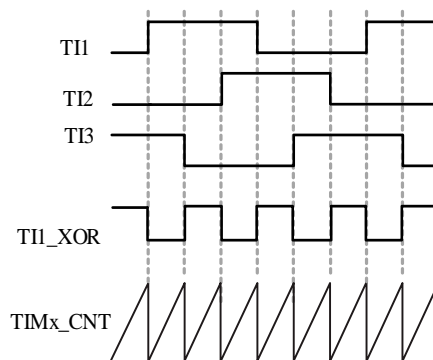
如果要输出延迟时间最短的波形，可以将 TIMx\_CCMx 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF（和 OCx）对触发信号做出响应，而不依赖于比较的结果。其输出的波形与比较匹配时的波形相同。仅在通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

#### 15.3.13 TIMx 输入异或功能

将 TIMx\_CR2 寄存器中的 TI1\_XOR\_SEL 位置 1，可将通道 1 的输入滤波器连接到异或门的输出端，异或门的 3 个输入端为 TI1、TI2 和 TI3。

异或输出用于定时器的输入功能，如触发或输入捕获。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 15-19 测量 3 个信号边沿之间的时间间隔



#### 15.3.14 TIMx 定时器同步从模式

##### 复位模式

当触发信号有效时，计数器及其预分频器被重新初始化。

如果 TIMx\_CR1 寄存器中的 URS 位为 0，且 UDIS 位也为 0，则会生成更新事件，且 TIMx\_ARR 和输出比较模式下的 TIMx\_CCx 寄存器的影子寄存器被更新。

在以下示例中，TI1 输入信号出现上升沿时，递增计数器清 0:

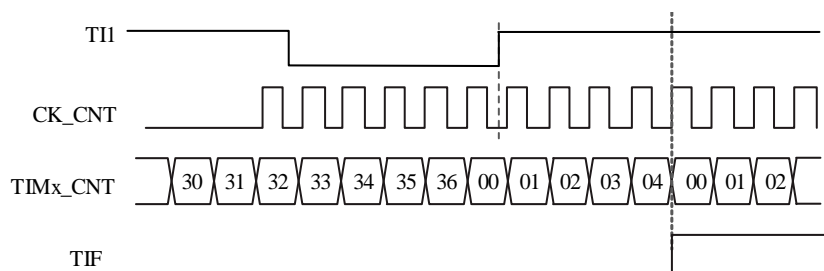
- 1) TI1 映射到 IC1 信号上: TIMx\_CCM1 寄存器中的 CC1S[1:0]位域写入 01;
- 2) 配置 TI1 上升沿有效: TIMx\_CCEN 寄存器中的 CC1P 位 CC1NP 位为 0;
- 3) 配置输入滤波带宽 (本例中不需要任何滤波器, 因此保持 IC1F[2:0]位域为 000);
- 4) 配置为复位模式: TIMx\_SMC 寄存器中的 SM\_SEL[2:0]位域写入 100。
- 5) 选择 TI1FP1 作为触发源: TIMx\_SMC 寄存器中的 TS[1:0]位域写入 10。
- 6) 使能计数器: TIMx\_CR1 寄存器中的 CEN 位置 1。

当 TI1 出现上升沿时, 计数器清 0, 重新从 0 开始计数。同时, 触发标志(TIMx\_SR 寄存器中的 TIF 位) 置 1; 如果使能 TIMx\_DIER 寄存器中的 TIE 位, 则可发送中断。

TI1 的上升沿与实际计数器复位之间的延迟是由于触发信号 TI1 输入的重新同步电路引起的。

下图为自动重载值寄存器 TIMx\_ARR=0x36 时的时序。

图 15-20 复位模式下的控制时序



## 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:

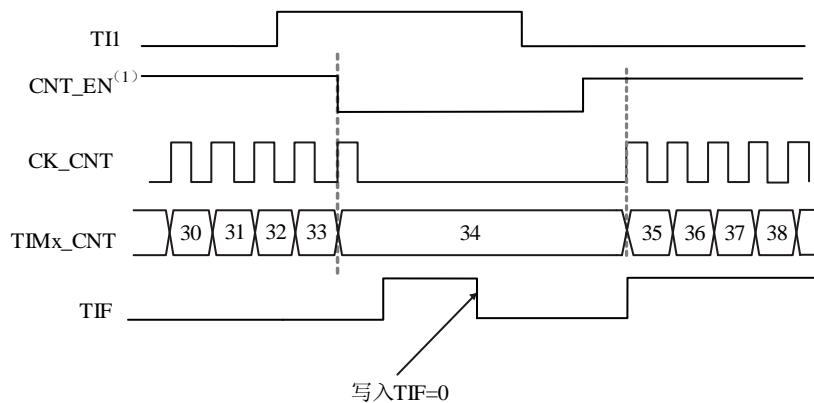
- 1) TI1 映射到 IC1 信号上: TIMx\_CCM1 寄存器中的 CC1S[1:0]位域写入 01;
- 2) 配置 TI1 低电平有效: TIMx\_CCEN 寄存器中的 CC1P 位为 1, CC1NP 位为 0;
- 3) 配置输入滤波带宽 (本例中不需要任何滤波器, 因此保持 IC1F[2:0]位域为 000);
- 4) 定时器配置为门控模式: TIMx\_SMC 寄存器中的 SM\_SEL[2:0] 位域写入 101;
- 5) 选择 TI1FP1 作为触发源: TIMx\_SMC 寄存器中的 TS[1:0]写入 10;
- 6) 使能计数器: TIMx\_CR1 寄存器中的 CEN 位置 1。



当 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx\_SR 寄存器中的 TIF 标志置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于触发信号 TI1 输入的重新同步电路引起的。

图 15-21 门控模式下的控制时序



1. CNT\_EN 为 TIMx 内部控制信号。

### 触发模式

所选输入端触发信号有效时可以使能计数器。

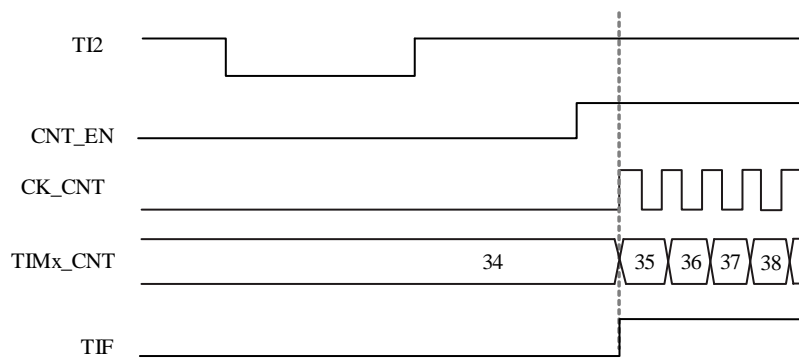
在以下示例中，当 TI2 输入上出现上升沿时，使能计数器：

- 1) TI2 映射到 IC2 信号上：TIMx\_CCM1 寄存器中的 CC2S[1:0]位域写入 01；
- 2) 配置 TI2 上升沿有效：TIMx\_CCEN 寄存器中的 CC2P 位和 CC2NP 位为 0，上升沿有效；
- 3) 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F[2:0]位域为 000）；
- 4) 定时器配置为触发模式：TIMx\_SMC 寄存器中 SM\_SEL[2:0] 位域写入 110；
- 5) 选择 TI2FP2 作为触发源：TIMx\_SMC 寄存器中的 TS[1:0]位域 11。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，TIMx\_SR 寄存器中的 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触发信号 TI2 输入的重新同步电路引起的。

图 15-22 触发模式下的控制时序

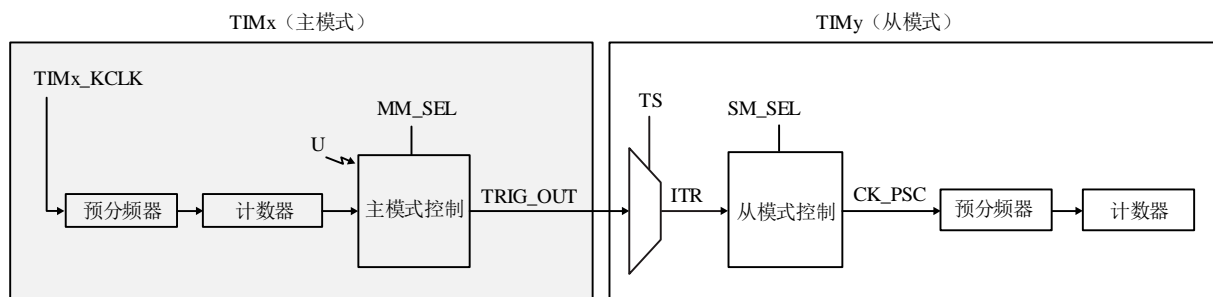


### 15.3.15 定时器同步主模式

TIMx 和内部其他 TIM 连在一起，可实现定时器同步或级联。

下图显示了配置为主/从定时器的连接框图。

图 15-23 主从定时器连接示例图



#### TIMx 配置为 TIMy 的预分频器

例如，将 TIM3 配置为 TIM1 的预分频器。其具体步骤如下：

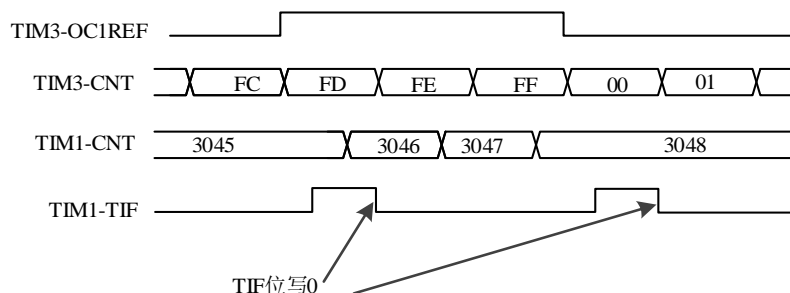
- 1) 设置 TIM3 为主模式：配置 TIM3\_CR2 寄存器中的 MM\_SEL[2:0]为 010，则每次生成更新事件时，TRIG\_OUT 都会输出一个上升沿；
- 2) 设置 TIM1 的从模式控制器设为时钟模式 1：配置 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域为 111，此时 TIM1 的时钟由 TIM3 周期性触发信号的上升沿提供；
- 3) 使能 TIM1 计数器：配置 TIM1\_CR1 寄存器中的 CEN 位为 1；
- 4) 使能 TIM3 计数器：配置 TIM3\_CR1 寄存器中的 CEN 位为 1。

#### TIMx 使能 TIMy

例如：TIM3 的输出比较 1 使能 TIM1，连接图如下所示。当 TIM3 的 OC1REF 为高电平时，TIM1 开始计数。

- 1) 设置 TIM3 为主模式，且输出比较 1 的参考信号 (OC1REF) 为触发输出：配置 TIM3\_CR2 寄存器中的 MM\_SEL[2:0]位域为 100；
- 2) 设置 TIM3 的 OC1REF 波形：配置 TIM3\_CCM1 寄存器；
- 3) 设置 TIM1 为门控模式：配置 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域为 101；
- 4) 使能 TIM1 计数器：配置 TIM1\_CR1 寄存器中的 CEN 位为 1；
- 5) 使能 TIM3 计数器：配置 TIM3\_CR1 寄存器中的 CEN 位为 1；

图 15-24 使用 TIM3 的 OC1REF 信号对 TIM1 实施门控控制



上图中，TIM1 的计数器和 PSC 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 TIM3 前，通过 UG 置 1 的方式复位 TIM，在从指定值开始计数。

### 使用一个外部触发同步启动 2 个定时器

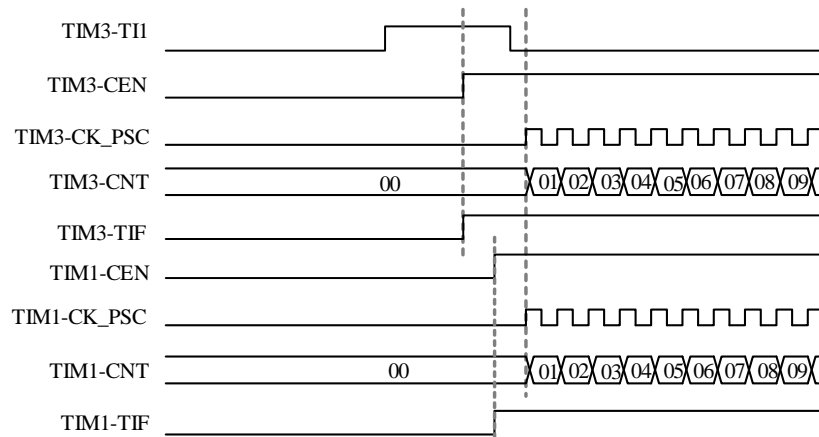
例如：TIM3 的 TI1 输入信号的上升沿使能 TIM3，同时也使能 TIM1。此时 TIM3 必须为主/从模式（对于 TI1 而言，TIM3 为从模式；对于 TIM1 而言，TIM3 为主模式）：

- 1) 设置 TIM3 的触发输出 TRIG\_OUT：配置 TIM3\_CR2 寄存器中的 MM\_SEL[2:0]位域为 001；
- 2) 设置 TIM3 用于接收 TI1 的输入触发 TRIG：配置 TIM3\_SMC 寄存器中的 TS[1:0]位域为 10；
- 3) 设置 TIM3 为触发模式：配置 TIM3\_SMC 寄存器中的 SM\_SEL[2:0]位域为 110；
- 4) 设置 TIM3 为主模式：配置 TIM3\_SMC 寄存器中的 MS\_MOD 位为 1；
- 5) 设置 TIM1 为触发模式：配置 TIM1\_SMC 寄存器中的 SM\_SEL[2:0]位域为 110。

当 TIM3 的 TI1 出现上升沿时，两个计数器开始同步计数，并且两个 TIF 标志都会置 1。

两个定时器在启动计数前进行了初始化（通过将各自的 UG 位置 1），两个计数器从 0 开始计数。如果计数器之前需要一个计数偏移量，可实时对其中一个计数器的 TIM\_CNT 寄存器进行写操作。

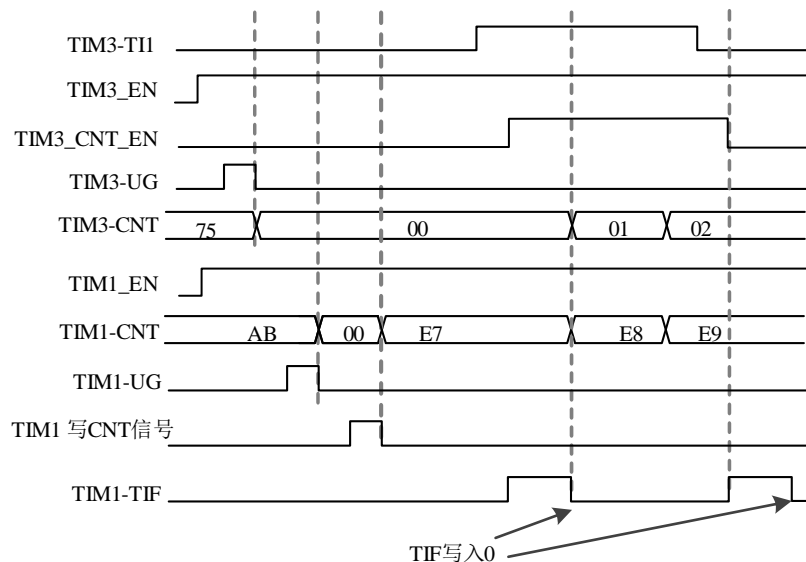
图 15-25 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM1



如果配置 TIM3 从模式控制器的触发模式为门控模式时，则当 TI1 为高电平，两个计数器开始计数；当 TI1 为低电平，两个计数器停止计数。

下列中 TIM3 为主模式，从 0 开始计数。TIM1 为从模式，从 0xE7 开始计数。两个 TIM 的预分频比相同。

图 15-26 使用 TIM3 的门控信号对 TIM1 实施门控控制



### 15.3.16 调试模式

当 CPU 进入调试模式时（Cortex-M0+内核停止），TIMx 计数器会根据 [APB 冻结寄存器 \(DBG\\_APB\\_FZ1\)](#) 寄存器中的 TIMx\_HOLD 配置位选择继续计数或者停止计数。

## 15.4 TIM3 中断

表 15-3 中断请求

中断事件	事件标志	使能控制位	清除方法
更新事件	UIF	UIE	UIF 位写 0 清除该位
捕获/比较 1~4	CCxIF	CCxIE	CCxIF 位写 0 或读取 TIMx_CCx 寄存器的值，可清除该位；
触发事件	TIF	TIE	TIF 位写 0 清除该位

## 15.5 TIM3 寄存器

TIMx 寄存器支持 32 位访问。

表 15-4 TIMx 基地址 (x=3)

外设	基地址
TIM3	0x4000 0400

### 15.5.1 TIM 控制寄存器 1 (TIMx\_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
MODE	Res.					CLK_DIV[1:0]		ARPE	Res.			OPM	URS	UDIS	CEN
rw						rw	rw	rw				rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	写入无效
15	MODE	工作模式配置选择 0: 工作模式 0, 2 通道输入, 2 通道输出模式, 精度为 16bit 1: 工作模式 1, 2 通道输入, 4 通道输出模式, 精度为 8bit 模式工作参见: <a href="#">捕获/比较通道</a> 。
14:10	保留	写入无效
9:8	CLK_DIV[1:0]	时钟分频 定时器时钟频率 ( $f_{TIMx\_KCLK}$ ) 与数字滤波器所使用的采样时钟 ( $f_{DTS}$ ) 之间的分频比。 00: $f_{DTS} = f_{TIMx\_KCLK}$ 01: $f_{DTS} = f_{TIMx\_KCLK}/2$ 10: $f_{DTS} = f_{TIMx\_KCLK}/4$ 11: 保留 (默认 $f_{DTS} = f_{TIMx\_KCLK}$ )

7	ARPE	TIMx_ARR寄存器预装载使能控制 0: 禁止 1: 使能
6:4	保留	写入无效
3	OPM	单脉冲模式 0: 禁止: 计数器在发生更新事件时不会停止计数 1: 使能: 计数器在发生更新事件时停止计数 (CEN位自动清0)
2	URS	更新事件请求源 UDIS为0时: 0: 以下事件都会产生更新事件: - 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 (复位模式) 1: 计数器上溢会产生更新事件: <i>注意: 当UDIS位为0时, 使能更新事件, 此时URS位配置为1, 则将UG位置1, 会生成更新事件, 计数器及PSC预分频计数器均被初始化, 但不会将UIF置位。</i>
1	UDIS	更新事件禁止控制位 0: 使能: 由URS位决定更新事件的触发源 1: 禁止: <i>注意: 将UDIS位置1, 如果此时将UG位置1, 或者从从模式控制器接收到一个硬件复位, 或计数器上溢时, 计数器和PSC预分频计数器被重新初始化, 但不会将UIF置位。</i>
0	CEN	计数器使能 0: 禁止 1: 使能 <i>注意: 在使用时钟模式1或门控模式时, 需软件将CEN置1。而触发模式可自动将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0。</i>

### 15.5.2 TIM 控制寄存器 2 (TIMx\_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.								TI1_XOR_SEL	MM_SEL[2:0]				Res.			
								rw	rw	rw	rw					

位/位域	名称	描述
31:8	保留	写入无效
7	TI1_XOR_SEL	<p>TI1选择</p> <p>0: TIMx_CH1引脚连接到TI1输入</p> <p>1: TIMx_CH1、CH2和CH3引脚异或组合后连接到TI1输入</p>
6:4	MM_SEL[2:0]	<p>主模式选择</p> <p>触发输出（TRIG_OUT）信号连接到其他外设的控制方式。</p> <p>000: 复位 — 以下信号都可以用作触发输出（TRIG_OUT）：</p> <ul style="list-style-type: none"> <li>TIMx_EVTG寄存器中的UG位</li> <li>触发输入生成（从模式控制器配置为复位模式），但TRIG_OUT上的信号相比实际复位会有一定延时</li> </ul> <p>001: 使能 — 计数器使能信号用作触发输出（TRIG_OUT）。计数器使能信号由TIMx_CR1寄存器的CEN控制位与门控模式下的触发输入的逻辑“与”运算组合而成。为保证当前定时器与从外设实现完美同步，需选择主/从模式（TIMx_SMC寄存器中MS_MOD位置1）</p> <p>010: 更新 — 选择更新事件作为触发输出（TRIG_OUT）</p> <p>011: 比较脉冲 — 当发生捕获或比较匹配时，CC1IF标志置1（即使已为高），此时触发输出（TRIG_OUT）都会发送一个正脉冲</p> <p>100: 比较 — OC1REF信号用作触发输出（TRIG_OUT）</p> <p>101: 比较 — OC2REF信号用作触发输出（TRIG_OUT）</p> <p>110: 比较 — OC3REF信号用作触发输出（TRIG_OUT）</p> <p>111: 比较 — OC4REF信号用作触发输出（TRIG_OUT）</p>
3:0	保留	写入无效



### 15.5.3 TIM 从模式控制寄存器 (TIMx\_SMC)

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.								MS_MOD	Res.	TS[1:0]		Res.	SM_SEL[2:0]		
								rw		rw	rw		rw	rw	rw

位/位域	名称	描述
31:8	保留	写入无效
7	MS_MOD	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（TRIG）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TRIG_OUT）。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
6	保留	写入无效
5:4	TS[1:0]	触发源选择 00：内部触发 0（ITR0） 01：TI1 边沿检测器 TI1F_ED 10：定时器输入 1（TI1FP1） 11：定时器输入 2（TI2FP2） <i>注意：当配置为从模式后，该位域不能被修改。</i>
3	保留	写入无效
2:0	SM_SEL[2:0]	从模式选择 触发信号（TRIG）的有效边沿由外部输入端所选的极性决定。 0xx：禁止从模式 — PSC 预分频器时钟由内部时钟提供，TIMx_CR1 寄存器中的 CEN 位为 1 时，启动计数

- 100: 复位模式 — 触发输入 TRIG 信号上升沿时, 重新初始化计数器并生成更新事件
- 101: 门控模式 — 触发输入 TRIG 信号为高电平时使能计数器时钟。只要 TRIG 信号变为低电平, 计数器立即停止计数 (但不复位)。计数器启动和停止都被控制
- 110: 触发模式 — 触发输入 TRIG 信号出现上升沿时启动计数器 (但不复位)。只控制计数器的启动
- 111: 时钟模式 1 — 触发输入 TRIG 信号的上升沿提供计数器时钟

#### 15.5.4 TIM 中断使能寄存器 (TIMx\_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.								TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE	
								rw		rw	rw	rw	rw	rw	

位/位域	名称	描述
31:7	保留	写入无效
6	TIE	触发事件的中断使能 0: 禁止 1: 使能
5	保留	写入无效
4	CC4IE	通道4比较事件的中断使能 0: 禁止 1: 使能
3	CC3IE	通道3比较事件的中断使能 0: 禁止 1: 使能

2	CC2IE	通道2捕获/比较事件的中断使能 0: 禁止 1: 使能
1	CC1IE	通道1捕获/比较事件的中断使能 0: 禁止 1: 使能
0	UIE	更新事件的中断使能 0: 禁止 1: 使能

### 15.5.5 TIM 状态寄存器 (TIMx\_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.					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

位/位域	名称	描述
31:11	保留	写入无效
10	CC2OF	通道2的重复捕获标志 请参见CC1OF位说明
9	CC1OF	通道1的重复捕获标志 通过写入0可将该位清0。 0: 未检测到重复捕获 1: CC1IF标志已置1时, 再次检测到捕获; TIMx_CC1寄存器捕获到新的计数器值
8:7	保留	写入无效

6	TIF	<p>触发事件标志</p> <ul style="list-style-type: none"> <li>– 门控模式：该标志将在计数器启动或停止时置1。</li> <li>– 非门控模式：当使能从模式控制器后，在TRIG信号上检测到有效边沿时，该标志自动置1。</li> </ul> <p>通过写0可清除该位。</p> <p>0：未发生触发事件</p> <p>1：发生触发事件</p>
5	保留	写入无效
4	CC4IF	<p>通道4的比较事件标志</p> <p>请参见CC1IF位输出比较说明</p>
3	CC3IF	<p>通道3的比较事件标志</p> <p>请参见CC1IF位输出比较说明</p>
2	CC2IF	<p>通道2的捕获/比较事件标志</p> <p>请参见CC1IF位说明</p>
1	CC1IF	<p>通道1的捕获/比较事件标志</p> <p>如果通道1配置为输出：</p> <p>当TIMx_CNT计数器的值与TIMx_CC1寄存器的值匹配时，此标志自动置1。通过写0可清除该位。</p> <p>0：不匹配</p> <p>1：匹配</p> <p>如果通道1配置为输入：</p> <p>发生捕获事件时自动置1，通过写0或读取TIMx_CC1寄存器将清除该位。</p> <p>0：未发生输入捕获事件</p> <p>1：发生输入捕获事件，TIMx_CC1寄存器中已捕获到计数器值</p>
0	UIF	<p>更新事件标志</p> <p>发生更新事件时自动置1。通过写0可清除该位。</p> <p>0：未发生更新</p> <p>1：发生更新：</p> <ul style="list-style-type: none"> <li>– 当TIMx_CR1寄存器中UDIS位为0，上溢时</li> <li>– TIMx_CR1寄存器中的URS位和UDIS位均为0，且TIMx_EVTG寄存器中的UG位置1时</li> </ul>

- TIMx\_CR1寄存器中的URS位和UDIS位均为0，且从模式控制器生成的更新事件(复位模式)(参见TIMx\_SMC寄存器说明)

### 15.5.6 TIM 事件产生寄存器 (TIMx\_EVTG)

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	w	w

位/位域	名称	描述
31:7	保留	写入无效
6	TG	生成触发事件 软件置1可生成触发事件。
5	保留	写入无效
4	CC4G	生成通道4比较事件 请参见CC1G位说明
3	CC3G	生成通道3比较事件 请参见CC1G位说明
2	CC2G	生成通道2捕获/比较事件 请参见CC1G位说明
1	CC1G	生成通道1捕获/比较事件 软件置1可生成捕获/比较事件。 如果通道配置为输出: TIMx_SR寄存器的CC1IF标志位置1 如果通道配置为输入: - TIMx_SR寄存器的CC1IF标志位置1; 如果 CC1IF 已置位,

则CC1OF 标志将置 1

- TIMx\_CC1寄存器中将捕获到计数器当前值

0	UG	更新生成
		此位由软件置1可生成更新事件。
		0: 不执行任何操作
		1: PSC预分频器计数器将清0（但TIMx_PSC寄存器的值不受影响）；TIMx_CNT被重新初始化。

### 15.5.7 TIM 捕获/比较模式寄存器 1 (TIMx\_CCM1) -- (输出比较模式)

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.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	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	写入无效
14:12	OC2M[2:0]	通道2输出比较模式选择 请参见OC1M[2:0]位域说明
11	OC2PE	通道2输出比较预装载使能 请参见OC1PE位说明
10	OC2FE	通道2输出比较快速使能 请参见OC1FE位说明
9:8	CC2S[1:0]	通道2捕获/比较选择 仅当通道关闭时（TIMx_CCEN中的CC2E位为0），CC2S位才可写入。 00: 通道2配置为输出 01: 通道2配置为输入，TI2映射到IC2上 10: 通道2配置为输入，TI1映射到IC2上

		11: 通道2配置为输入, TRC映射到IC2上。此模式仅在通过TIMx_SMC寄存器中TS[1:0]位域选中内部触发输入时有效
7	保留	写入无效
6:4	OC1M[2:0]	<p>通道1输出比较模式选择</p> <p>定义信号OC1REF的行为, OC1REF信号决定OC1的值。OC1REF为高电平有效。OC1的有效电平则取决于TIMx_CCEN寄存器的CC1P位。</p> <p>000: 冻结 — TIMx_CC1与TIMx_CNT间的比较对输出无影响(仅作计数器时使用)</p> <p>001: 匹配输出有效电平 — 当TIMx_CNT与TIMx_CC1匹配时, OC1REF信号强制变为有效电平(匹配前OC1REF信号为无效电平)</p> <p>010: 匹配输出无效电平 — 当TIMx_CNT与TIMx_CC1匹配时, OC1REF信号强制变为无效电平(匹配前OC1REF信号为无效电平)</p> <p>011: 翻转 — 当TIMx_CNT与TIMx_CC1匹配时, OC1REF发生翻转</p> <p>100: 强制变为无效电平 — OC1REF强制变为无效电平</p> <p>101: 强制变为有效电平 — OC1REF强制变为有效电平</p> <p>110: PWM模式1: 当TIMx_CNT &lt; TIMx_CC1, 通道1输出为有效电平, 否则为无效电平</p> <p>111: PWM模式2: 当TIMx_CNT &lt; TIMx_CC1, 通道1输出为无效电平, 否则为有效电平</p>
3	OC1PE	<p>通道1输出比较预装载使能</p> <p>0: 禁止: 可随时向TIMx_CC1写入数据, 写入后将立即生效</p> <p>1: 使能: 可读/写访问预装载寄存器, TIMx_CC1预装载值在更新事件时才会生效</p>
2	OC1FE	<p>通道1输出比较快速使能</p> <p>此位用于加快触发输入事件对OC输出的影响</p> <p>0: 禁止: 触发输入事件发生时, OC1也将根据计数器和CC1值匹配结果正常输出</p> <p>1: 使能: 触发输入有效边沿相当于OC1输出上的比较匹配。随后, 无论比较结果如何, OC1都设置为比较后电平。仅在PWM1或PWM2模式时, OC1FE才会起作用</p>

**1:0**                      **CC1S[1:0]**                      通道1捕获/比较选择

仅当通道关闭时（TIMx\_CCEN中的CC1E位为0），才可向CC1S位写入数据。

00：通道1配置为输出

01：通道1配置为输入，TI1映射到IC1上

10：通道1配置为输入，TI2映射到IC1上

11：通道1配置为输入，TRC映射到IC1上。此模式仅在通过TIMx\_SMC寄存器中TS[1:0]位域选中内部触发输入时有效

### 15.5.8 TIM 捕获/比较模式寄存器 1 (TIMx\_CCM1) -- (输入捕获模式)

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.	IC2F[2:0]			IC2PSC[1:0]		CC2S[1:0]		Res.	IC1F[2:0]			IC1PSC[1:0]		CC1S[1:0]	
	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	写入无效
14:12	IC2F[2:0]	通道2输入捕获滤波器配置 请参见 IC1F[2:0]位域说明
11:10	IC2PSC[1:0]	通道2输入捕获预分频器 请参见IC1PSC [1:0]位域说明
9:8	CC2S[1:0]	通道2捕获/比较选择 请参见 <a href="#">TIMx捕获 /比较模式寄存器1 (TIMx_CCM1) -- (输出比较模式)</a> CC2S[1:0]说明
7	保留	写入无效
6:4	IC1F[2:0]	通道1输入捕获滤波器配置 配置 TI1 输入的采样频率和数字滤波宽度。连续 4 个采样周期



均采样到有效电平时，才视为一个有效电平：

000：无滤波器，按  $f_{DTS}$  频率进行采样

001：  $f_{SAMPLING} = f_{TIMx\_KCLK}$

010：  $f_{SAMPLING} = f_{DTS}/2$

011：  $f_{SAMPLING} = f_{DTS}/4$

100：  $f_{SAMPLING} = f_{DTS}/8$

101：  $f_{SAMPLING} = f_{DTS}/16$

110：  $f_{SAMPLING} = f_{DTS}/32$

111：  $f_{SAMPLING} = f_{DTS}/32$

3:2 IC1PSC[1:0]

通道1输入捕获预分频器

只要TIMx\_CCEN寄存器的CC1E位为0，预分频器便立即复位。

00：无预分频器，捕获输入上信号上每检测到一个有效边沿便执行捕获

01：每发生2个事件执行一次捕获

10：每发生4个事件执行一次捕获

11：每发生8个事件执行一次捕获

1:0 CC1S [1:0]

通道1捕获/比较选择

请参见 [TIMx捕获/比较模式寄存器1 \(TIMx\\_CCM1\)](#)--(输出比较模式) CC1S[1:0]说明

### 15.5.9 TIM 捕获/比较模式寄存器 2 (TIMx\_CCM2)

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.	OC4M[2:0]			OC4PE	OC4FE	Res.			OC3M[2:0]			OC3PE	OC3FE	Res.	
	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw		

位/位域	名称	描述
31:15	保留	写入无效
14:12	OC4M[2:0]	通道4输出比较模式 请参见OC1M[1:0]位域说明

11	OC4PE	通道4输出比较预装载使能 请参见OC1PE位说明
10	OC4FE	通道4输出比较快速使能 请参见OC1FE位说明
9:7	保留	写入无效
6:4	OC3M[2:0]	通道3输出比较模式选择 请参见OC1M[1:0]位域说明
3	OC3PE	通道 3 输出比较预装载使能 请参见 OC1PE 位说明
2	OC3FE	通道3输出比较快速使能 请参见OC1FE位说明
1:0	保留	写入无效

#### 15.5.10 TIM 捕获/比较使能寄存器 (TIMx\_CCEN)

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
Res.	CC4P	CC4E	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E		
	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw		

位/位域	名称	描述
31:14	保留	写入无效
13	CC4P	通道4比较极性 请参见CC1P位说明
12	CC4E	通道 4 比较使能

		请参见 CC1E 位说明
11:10	保留	写入无效
9	CC3P	通道3比较输极性 请参见CC1P位说明
8	CC3E	通道3比较使能 请参见CC1E位说明
7	CC2NP	通道2捕获/比较互补极性 请参见CC1NP位说明，仅在配置为输入有效
6	保留	写入无效
5	CC2P	通道2捕获/比较极性 请参见CC1P位说明
4	CC2E	通道2捕获/比较使能 请参见CC1E位说明
3	CC1NP	通道1捕获/比较互补极性 通道1配置为输出： CC1NP必须保持清0 通道1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	写入无效
1	CC1P	通道1捕获/比较极性 通道1配置为输出： 0: OC1高电平有效 1: OC1低电平有效 通道1配置为输入： CC1NP/CC1P位选择TI1FP1和TI2FP1的有效极性： – CC1NP=0, CC1P=0: TIxFP1上升沿有效（在复位模式、时钟模式1或触发模式下

生效)

TIxFP1未反相（在门控模式下生效）

- CC1NP=0, CC1P=1:

TIxFP1下降沿有效（在复位模式、时钟模式1或触发模式下生效）

TIxFP1反相（在门控模式下生效）

- CC1NP=1, CC1P=1:

TIxFP1上升沿和下降沿都有效（在复位模式、时钟模式1或触发模式下生效）

TIxFP1未反相（在门控模式下生效）。

- CC1NP=1, CC1P=0: 保留（默认未反相/上升沿有效）

0	CC1E	通道1捕获/比较使能
		通道1配置为输出:
		0: 禁止
		1: 使能
		通道1配置为输入:
		输入捕获使能位
		0: 禁止
		1: 使能

### 15.5.11 TIM 计数值寄存器 (TIMx\_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
Res.															
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
31:16	保留	写入无效
15:0	CNT[15:0]	计数器值

### 15.5.12 TIM 预分频值寄存器 (TIMx\_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
Res.												PSC[3:0]			
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	写入无效
3:0	PSC [3:0]	预分频值 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (2^{(psc[3:0])})$ 。

### 15.5.13 TIM 自动重载值寄存器 (TIMx\_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	写入无效
15:0	ARR[15:0]	自动重载值 当自动重载值为空时, 计数器不工作。

### 15.5.14 TIM 捕获/比较寄存器 1 (TIMx\_CC1) -- (工作模式 0)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1_MODE0[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	写入无效
15:0	CC1_MODE0[15:0]	<p>通道1的捕获/比较值</p> <p>通道1配置为输出比较模式：</p> <ul style="list-style-type: none"> <li>当TIMx_CCM1寄存器中的OC1PE位为1时，使能预装载功能，该位域的新值将在更新事件后生效</li> <li>当TIMx_CCM1寄存器中的OC1PE位为0时，该位域的新值立即生效</li> </ul> <p>通道1配置为输入捕获模式：</p> <p>该寄存器为只读状态。当发生输入捕获事件时，将捕获当前计数器的值。</p>

### 15.5.15 TIM 捕获/比较寄存器 1 (TIMx\_CC1) -- (工作模式 1)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3_MODE1[7:0]								CC1_MODE1[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	写入无效
15:8	CC3_MODE1[7:0]	<p>通道3的比较值</p> <ul style="list-style-type: none"> <li>当TIMx_CCM2寄存器中的OC3PE位为1时，使能预装载功能，该位域的新值将在更新事件后生效</li> </ul>

- 当TIMx\_CCM2寄存器中的OC3PE位为0时,该位域的新值立即生效

7:0	CC1_MODE1[7:0]	通道1的捕获/比较值
		通道1配置为输出比较模式:
		- 当TIMx_CCM1寄存器中的OC1PE位为1时,使能预装载功能,该位域的新值将在更新事件后生效
		- 当TIMx_CCM1寄存器中的OC1PE位为0时,该位域的新值立即生效
		通道1配置为输入捕获模式:
		该寄存器为只读状态。当发生输入捕获事件时,将捕获当前计数器的值。

### 15.5.16 TIM 捕获/比较寄存器 2 (TIMx\_CC2) -- (工作模式 0)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC2_MODE0[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	写入无效
15:0	CC2_MODE0[15:0]	通道2的捕获/比较值 请参见CC1_MODE0[15:0]位域说明

### 15.5.17 TIM 捕获/比较寄存器 2 (TIMx\_CC2) -- (工作模式 1)

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4_MODE1[15:0]								CC2_MODE1[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	写入无效
15:8	CC4_MODE1[7:0]	通道4的比较值 请参见CC3_MODE1[7:0]位域说明
7:0	CC2_MODE1[7:0]	通道2的捕获/比较值 请参见CC1_MODE1[7:0]位域说明

### 15.5.18 TIM 定时器输入选择寄存器 (TIMx\_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.							TI2_SEL	Res.							TI1_SEL
							rw								rw

位/位域	名称	描述
31:9	保留	写入无效
8	TI2_SEL	TI2 输入源选择 0: CH2 1: COMP2 输出
7:1	保留	写入无效
0	TI1_SEL	TI1 输入源选择 0: CH1 1: COMP1 输出



## 16 低功耗定时器（LPTIM）

### 16.1 简介

LPTIM 是一个 16 位定时器，可选择 RCL、PCLK 或 MCO 作为工作时钟源。可在 Stop 或 Deepstop 模式下工作，具有定时唤醒功能。

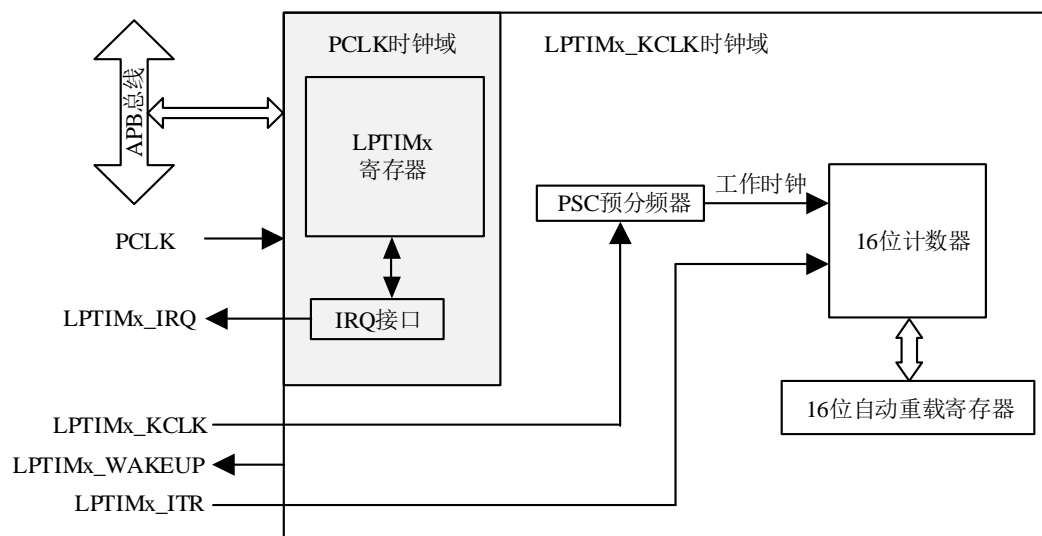
### 16.2 LPTIM 主要特性

- 16 位递增计数器
- 3 位计数预分频器
- 支持单次计数模式，连续计数模式和级联触发计数模式
- 可选工作时钟：RCL、PCLK 或 MCO
- 支持低功耗模式唤醒

### 16.3 LPTIM 功能描述

#### 16.3.1 LPTIM 框图

图 16-1 LPTIM<sub>x</sub> 框图 (x=1)



#### 16.3.2 LPTIM 内部信号

表 16-1 LPTIM 内部信号

信号名称	信号类型	描述
LPTIM <sub>x</sub> _KCLK	输入	LPTIM 外设时钟，来自 RCC

信号名称	信号类型	描述
LPTIMx_ITR	输入	LPTIM 级联计数控制，来自 TIM3_TRIG_OUT
LPTIMx_WAKEUP	输出	LPTIM 唤醒源信号
LPTIMx_IRQ	输出	LPTIM 全局中断

### 16.3.3 时钟源

LPTIMx\_KCLK 时钟源通过 [外设异步时钟配置寄存器 \(RCC\\_CLKSEL\)](#) 中 LPTIM1\_SEL[1:0]位域配置，可选择 PCLK、RCL 或 MCO 作为 LPTIM 的时钟源。

### 16.3.4 计数器模式

使能 LPTIM 后，LPTIMx\_CR 寄存器中的 SNGSTRT 位或 CNTSTRT 位置位才能生效。

- 连续计数模式：启动计数后，一直计数直到 LPTIM 被关闭才会停止；
- 单次计数模式：启动计数后，计数计完一个周期后停止。

LPTIM 可以在连续计数模式和单次计数模式间切换。如果之前是连续计数模式，将 SNGSTRT 置位会切换为单次计数模式，计数器计完一个计数周期后立即停止。如果之前是单次计数模式，将 CNTSTRT 置位会切换为连续计数模式。计数器将连续计数，直到 LPTIM 被关闭。

若 CNTSTRT、SNGSTRT 位被同时置位，此时计数器工作在连续计数模式下。

#### 连续计数模式

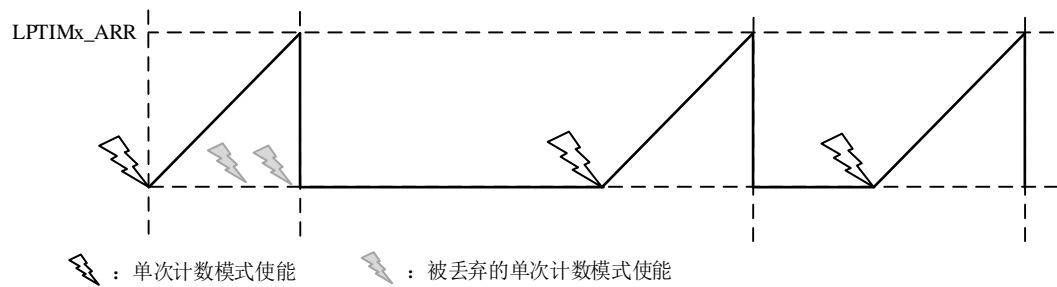
使能连续计数模式计数，需要将 LPTIMx\_CR 寄存器中的 CNTSTRT 位置位。CNTSTRT 置位后由硬件立即自动清零。连续计数期间再次将 CNTSTRT 置位，将被丢弃。

若 LPTIMx\_CNT 寄存器值与 LPTIMx\_ARR 寄存器值匹配，则 LPTIMx\_ISR 寄存器中的 ARRM 位置位。ARRM 位由 LPTIMx\_ICR 寄存器中的 ARRM\_CF 置位清零。

#### 单次计数模式

使能单次计数模式计数，需要将 LPTIMx\_CR 寄存器中的 SNGSTRT 位置位。SNGSTRT 置位后，由硬件立即自动清零。在当前计数周期内，SNGSTRT 位多次置位将被丢弃，如下图所示。

图 16-2 单次计数模式使能说明



### 级联触发计数模式

配置 LPTIMx\_CFG 寄存器中的 ITREN 位, 可使能 LPTIMx\_ITR 信号对 LPTIMx 的计数控制, 该模式下无需配置 CNTSTRT 或 SNGSTRT 位, LPTIM 工作在连续计数模式。

ITREN 位在禁止 LPTIM 后才能修改:

- ITREN 位为 1: 当 LPTIMx\_ITR 信号为 1 时, LPTIM 启动计数; 当 LPTIMx\_ITR 信号为 0 时, LPTIM 暂停计数;
- ITREN 位为 0: LPTIM 的计数不受 LPTIMx\_ITR 信号的控制。

LPTIMx\_ITR 信号发生变化时, LPTIMx\_ISR 寄存器中的 ITRF 位置位。如果 LPTIMx\_IER 寄存器中的 ITRF\_IE 位置位, 则触发中断。向 LPTIMx\_ICR 寄存器中的 ITRF\_CF 位写 1, 可清除该位。

#### 16.3.5 定时器使能

配置 LPTIMx\_CR 寄存器中的 ENABLE 位, 使能或者禁止 LPTIM。将 ENABLE 位置位后, 在 2 个 LPTIMx\_KCLK 时钟周期后使能生效。

LPTIMx\_CFG 寄存器必须在禁止 LPTIM 后才能修改; LPTIMx\_ARR 必须在使能 LPTIM 后才能修改。

当禁止 LPTIM 时, LPTIMx\_CNT 和 LPTIMx\_ISR 寄存器被清零。

#### 16.3.6 调试模式

当 CPU 进入调试模式 (Cortex-M0+内核停止) 时, 计数器会根据 [APB 冻结寄存器 \(DBG\\_APB\\_FZ1\)](#) 寄存器中的 LPTIMx\_HOLD 配置位选择继续正常工作或者停止工作。当计数器停止 (LPTIMx\_HOLD 置位) 时, 输出保持当前电平。

## 16.4 LPTIM 低功耗模式

表 16-2 低功耗模式对 LPTIM 的影响

模式	说明
Sleep	无影响，LPTIM 中断可使芯片退出 Sleep 模式。
Stop 或 Deepstop	当 LPTIM 由 RCL 提供内部时钟源时无影响，LPTIM 的 ARRM 中断可使芯片退出 Stop 或 Deepstop 模式。

## 16.5 LPTIM 中断

表 16-3 中断请求

中断事件	事件标志	使能控制位	清除方法
自动重载匹配	ARRM	ARRM_IE	ARRM_CF 位置位
级联触发标志	ITRF	ITRF_IE	ITRF_CF 位置位

## 16.6 LPTIM 寄存器

LPTIM 寄存器支持 32 位访问。

表 16-4 LPTIMx 基地址 (x=1)

外设	基地址
LPTIM1	0x4000 7C00

### 16.6.1 LPTIM 中断和状态寄存器 (LPTIMx\_ISR)

偏移地址: 0x00

复位值: 0x0000 0000

**注意:** 禁止 LPTIM (*ENABLE=0*) 时, 该寄存器被清 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
ITRF	Res.													ARRM	Res.
r														r	

位/位域	名称	描述
31:16	保留	写入无效
15	ITRF	<p>级联触发标志</p> <p>该标志在级联计数启动或停止时置位, 向 LPTIMx_ICR 寄存器中 ITRF_CF 位写 1 时该标志清 0。</p> <p>0: 未发生触发</p> <p>1: 发生触发</p> <p><b>注意:</b> 当 <i>ITREN</i> 为 0 时, 该位自动清 0。</p>
14:2	保留	写入无效
1	ARRM	<p>自动重载匹配标志</p> <p>当 LPTIMx_CNT 寄存器的值计到 LPTIMx_ARR 寄存器的值时, 该位自动置位。LPTIMx_ICR 寄存器中 ARRM_CF 位置位, 该位清 0。</p> <p>0: 未发生自动重载匹配事件</p>

1: 发生自动重载匹配事件

0                      保留                      写入无效

## 16.6.2 LPTIM 中断清除寄存器 (LPTIMx\_ICR)

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
ITRF_CF	Res.													ARRM_CF	Res.
w														w	

位/位域	名称	描述
31:16	保留	写入无效
15	ITRF_CF	级联触发标志清零 该位置位，LPTIMx_ISR寄存器中的ITRF位清零。
14:2	保留	写入无效
1	ARRM_CF	自动重载匹配标志清零 该位置位，LPTIMx_ISR寄存器中的ARRM位清零。
0	保留	写入无效

## 16.6.3 LPTIM 中断使能寄存器(LPTIMx\_IER)

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
ITRF_IE	Res.													ARRM_IE	Res.

rw														rw	
----	--	--	--	--	--	--	--	--	--	--	--	--	--	----	--

位/位域	名称	描述
31:16	保留	写入无效
15	ITRF_IE	级联触发中断使能位 0: 禁止 1: 使能
14:2	保留	写入无效
1	ARRM_IE	自动重载匹配中断使能位 0: 禁止 1: 使能
0	保留	写入无效

#### 16.6.4 LPTIM 配置寄存器(LPTIMx\_CFG)

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

位/位域	名称	描述
31:17	保留	写入无效
16	ITREN	级联触发使能 0: 禁止 1: 使能

15:12	保留	写入无效
11:9	PRESC[2:0]	PSC预分频器的分频系数选择 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
8:0	保留	写入无效

#### 16.6.5 LPTIM 控制寄存器 (LPTIMx\_CR)

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.													CNTS TRT	SNGS TRT	ENABLE
													w	w	rw

位/位域	名称	描述
31:3	保留	写入无效
2	CNTSTRT	连续计数模式使能控制 软件置位，硬件自动清零。
1	SNGSTRT	单次计数模式使能控制 软件置位，硬件自动清零。
0	ENABLE	LPTIM使能控制



0: 禁止

1: 使能

### 16.6.6 LPTIM 自动重载寄存器 (LPTIMx\_ARR)

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.															
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	写入无效
15:0	ARR[15:0]	自动重载值

### 16.6.7 LPTIM 计数器寄存器 (LPTIMx\_CNT)

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
CNT [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

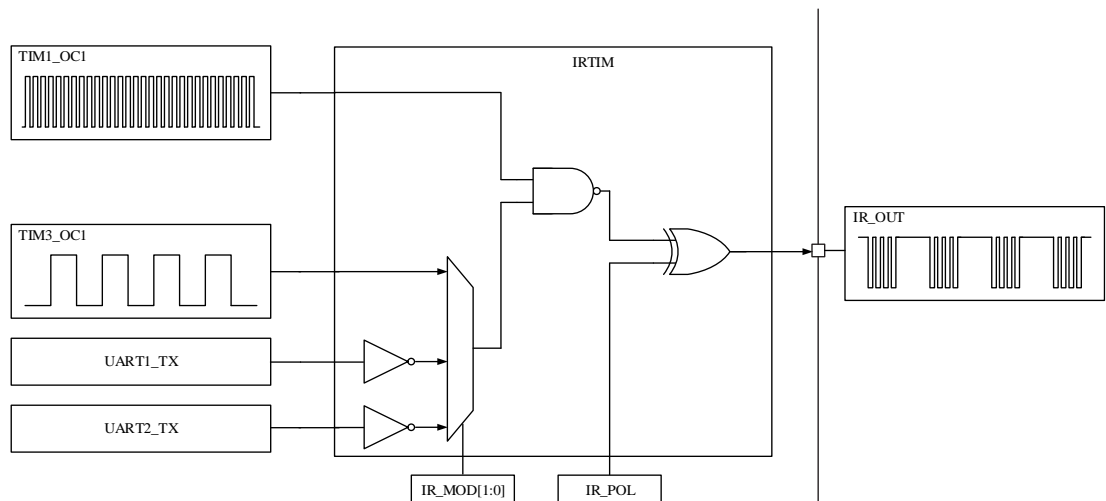
位/位域	名称	描述
31:16	保留	写入无效
15:0	CNT[15:0]	计数器值。 读取计数值时应连续读取2次以上, 为相同值才认为是稳定的结果。

## 17 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 UART1、UART2、TIM1、TIM3 相连，详见下图。

图 17-1 IRTIM 框图



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。

TIM1 定时器的通道 1（TIM1\_OC1）用于产生高频载波信号。

通过配置 IRTIM 控制寄存器（IRTIM\_CR）中的 IR\_MODE[1:0] 位域，可以选择 TIM3 定时器的通道 1（TIM3\_OC1）、UART1 或 UART2 中的一路信号作为调制信号。

IRTIM 输出信号的极性，由 IRTIM 控制寄存器（IRTIM\_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 引脚的复用功能进行配置。

## 17.1 IRTIM 寄存器

IRTIM 寄存器支持 32 位访问。

表 17-1 IRTIM 基地址列表

外设	基地址
IRTIM	0x4001 0000

### 17.1.1 IRTIM 控制寄存器 (IRTIM\_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											IR_MODE[1:0]		IR_POL	Res.	
											rw	rw	rw		

位/位域	名称	描述
31:5	保留	写入无效
4:3	IR_MODE[1:0]	IRTIM调制信号选择 00: TIM3_OC1 01: UART1_TX 10: UART2_TX 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	保留	写入无效

## 18 独立看门狗（IWDG）

### 18.1 简介

芯片内置独立看门狗，可用于检测并解决芯片发生的异常，并在计数器由给定的超时值递减到 0 时触发系统复位。

独立看门狗由内部低速时钟 RCL 驱动，因此在系统时钟发生故障时仍然保持工作状态，提高系统可靠性。

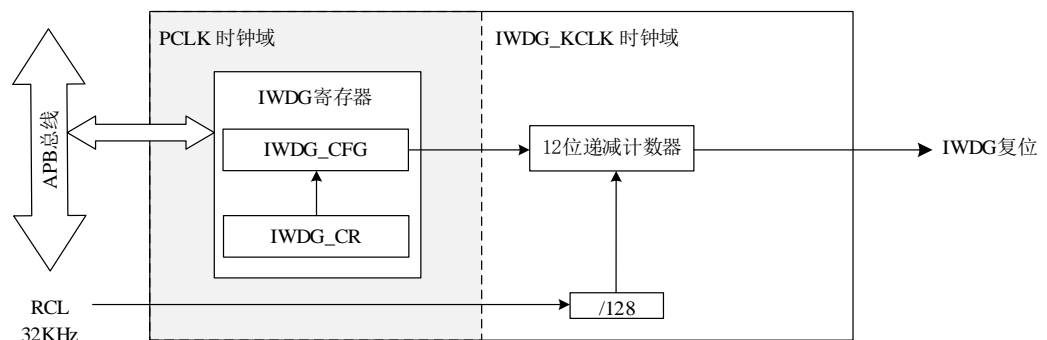
### 18.2 IWDG 主要特性

- 12 位递减计数器
- 工作时钟由 RCL 提供，可在 Stop 或 Deepstop 模式下工作
- 当达到溢出时间时复位

### 18.3 IWDG 功能描述

#### 18.3.1 IWDG 框图

图 18-1 IWDG 框图



IWDG 功能在 Stop 或 Deepstop 模式下仍能工作。

通过将 0x0000 CCCC 写到控制寄存器（IWDG\_CR）来使能独立看门狗时，计数器从复位值（0xFFFF）开始递减计数。当计数达到 0x000 时会产生 IWDG 复位。

IWDG 使能后任何时候将 0x0000 AAAA 写到控制寄存器（IWDG\_CR）中，IWDG\_CFG 中设置的计数溢出时间值将被重载到计数器中，从而避免产生看门狗复位。

IWDG 一旦运行，其时钟源（RCL）无法停止；IWDG 使能后，RCL 自动打

开。

### 18.3.2 寄存器访问保护

配置寄存器（IWDG\_CFG）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\_CR）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。

### 18.3.3 溢出时间设置

IWDG 内部预分频为 128，分频后的计数溢出时间共 8 个可用档位，溢出时间与计数周期之间的关系如下：

$$t_{IWDG} = t_{RCL} * 128 * N$$

表 18-1 IWDG 溢出周期表

RCL	计数周期 (N)	溢出时间 (ms)
32KHz	32	128
	64	256
	128	512
	256	1024
	512	2048
	1024	4096
	2048	8192
	4096	16384

实时修改溢出时间，可按以下步骤配置：

- 1) 确保 IWDG 正在运行；
- 2) 将 0x0000 5555 写入控制寄存器（IWDG\_CR），解除写保护；
- 3) 修改配置寄存器（IWDG\_CFG）中的计数溢出时间；
- 4) 读取该计数溢出时间选择值，确保写入正确；
- 5) 将 0x0000 AAAA 写入控制寄存器（IWDG\_CR），进行喂狗操作。

### 18.3.4 调试模式

当芯片进入调试模式时，IWDG 计数器会根据 DBG 模块中的 [APB 冻结寄存器 \(DBG\\_APB\\_FZI\)](#) 中的 IWDG\_HOLD 选择继续计数或者停止计数。

## 18.4 IWDG 低功耗模式

表 18-2 低功耗模式对 IWDG 的影响

模式	说明
Sleep	无影响
Stop 或 Deepstop	无影响，依据 <a href="#">选项字节寄存器 2 (FLASH_OTPR2)</a> 的 IWDG_STOP 进行计数或停止计数。

## 18.5 IWDG 寄存器

IWDG 寄存器支持 32 位访问。

表 18-3 IWDG 基地址

外设	基地址
IWDG	0x4000 3000

### 18.5.1 IWDG 控制寄存器 (IWDG\_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
Res.															
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
31:16	保留	写入无效
15:0	KEY[15:0]	控制值 必须定期对这些位写入 0x0000 AAAA，使递减计数器重新载入超时值；否则递减到 0 时，看门狗会产生复位。 写入 0x0000 5555 可使能对配置寄存器 (IWDG_CFG) 的访问。 写入 0x0000 CCCC 可使能看门狗。

### 18.5.2 IWDG 配置寄存器 (IWDG\_CFG)

偏移地址: 0x04

复位值: 0x0000 0007

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
														OVP[2:0]	



位/位域	名称	描述
31:3	保留	写入无效
2:0	OVP[2:0]	<p>计数溢出时间选择</p> <p>该位域受写访问保护，用于选择计数器溢出时间。</p> <p>000: 128ms</p> <p>001: 256ms</p> <p>010: 512ms</p> <p>011: 1.024s</p> <p>100: 2.048s</p> <p>101: 4.096s</p> <p>110: 8.192s</p> <p>111: 16.384s</p>

偏移地址: 0x14

复位值: 0x0000 0fff

位/位域	名称	描述
31:12	保留	写入无效
11:0	CNT[11:0]	<p>当前计数值</p> <p>由于计数器工作时钟与 APB 总线为异步关系，软件读取计数值时应连续读取 2 次以上，为相同值时才认为是稳定结果。</p>



## 19 SysTick 定时器 (SysTick)

### 19.1 简介

ARM Cortex-M0+内核提供了一个定时器 SysTick。

### 19.2 SysTick 主要特性

- 24 位计数器
- 递减计数
- 提供异常处理中断

### 19.3 SysTick 功能描述

SysTick 是一个 24 位定时器。通过将 SysTick\_CTRL 寄存器的 ENABLE 位置 1 使能时，计数器从当前计数值寄存器 SysTick\_VAL 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 SysTick\_LOAD 寄存器的值到 SysTick\_VAL 中，并再次进行递减计数。

每次计数到 0 时，SysTick\_CTRL 的 COUNTFLAG 标志位将置位，读此位时标志清零。

向 SysTick\_VAL 寄存器写入任意值时，SysTick\_VAL 寄存器和 COUNTFLAG 标志位都将清零，在下一个时钟沿 SysTick\_VAL 寄存器会装载 SysTick\_LOAD 中的计数值，重新开始计数，此过程不会触发中断。读 SysTick\_VAL 寄存器时返回值为当前计数器的值。

向 SysTick\_LOAD 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”。

在首次使能 SysTick 前首先写入 SysTick\_LOAD 寄存器，再写入 SysTick\_VAL。这样的操作顺序保证了正确的重载值被写入，并同时清空 SysTick\_VAL。之后 SysTick 使能时直接从 SysTick\_LOAD 开始计数。

当芯片处于调试模式内核停止时，SysTick 也会停止计数。

## 19.4 SysTick 寄存器

### 19.4.1 SysTick 控制和状态寄存器 (SysTick\_CTRL)

地址：0xE000 E010

复位值：0x0000 0004

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

位/位域	名称	描述
31:17	保留	读为0，写无效
16	COUNTFLAG	<p>溢出标志位</p> <p>上一次读SysTick_CTRL寄存器后，SysTick是否出现过溢出（计数到0）。</p> <p>此位在计数值从1减为0时置位，读取该位或者SysTick_VAL寄存器写入任意值后，该标志位自动清零。</p> <p>0：未发生过溢出</p> <p>1：发生过溢出</p>
15:3	保留	读为0，写无效
2	CLKSOURCE	<p>SysTick时钟源选择</p> <p>无外部参考时钟，计数时钟源始终为内核时钟HCLK</p> <p>读为1，写无效</p>
1	TICKINT	<p>SysTick中断使能</p> <p>0：禁止</p> <p>1：使能</p>
0	ENABLE	<p>SysTick使能计数</p> <p>0：禁止</p> <p>1：使能</p>

#### 19.4.2 SysTick 重载值寄存器 (SysTick\_LOAD)

地址：0xE000 E014

复位值：0x00XX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								RELOAD[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	写入无效
23:0	RELOAD[23:0]	SysTick的重载计数值 当计数值递减到0时，该值装载到SysTick_VAL寄存器

#### 19.4.3 SysTick 当前计数值寄存器 (SysTick\_VAL)

地址：0xE000 E018

复位值：0x00XX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								CURRENT[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	写入无效
23:0	CURRENT[23:0]	计数器当前值 读此寄存器时获取SysTick的当前计数值。 写任意值到该寄存器，该寄存器和SysTick_CTRL寄存器的COUNTFLAG标志都清零。

#### 19.4.4 SysTick 校准值寄存器 (SysTick\_CALIB)

地址：0xE000 E01C

复位值：0x8000 BB7F

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	计数时钟标志 1: SysTick总是使用内核时钟
30	SKEW	TENMS精度指示 0: TENMS校准值代表精确的1ms
29:24	保留	写入无效
23:0	TENMS[23:0]	1ms校准计数值0xBB7F 时钟源HCLK配置为48MHz，SysTick计数间隔为1ms的计数值

## 20 I2C 接口 (I2C)

### 20.1 简介

I2C 接口实现芯片和外部 I2C 设备的同步通信,支持标准模式(Standard-mode)、快速模式 (Fast-mode) 和超快速模式 (Fast-mode Plus)。

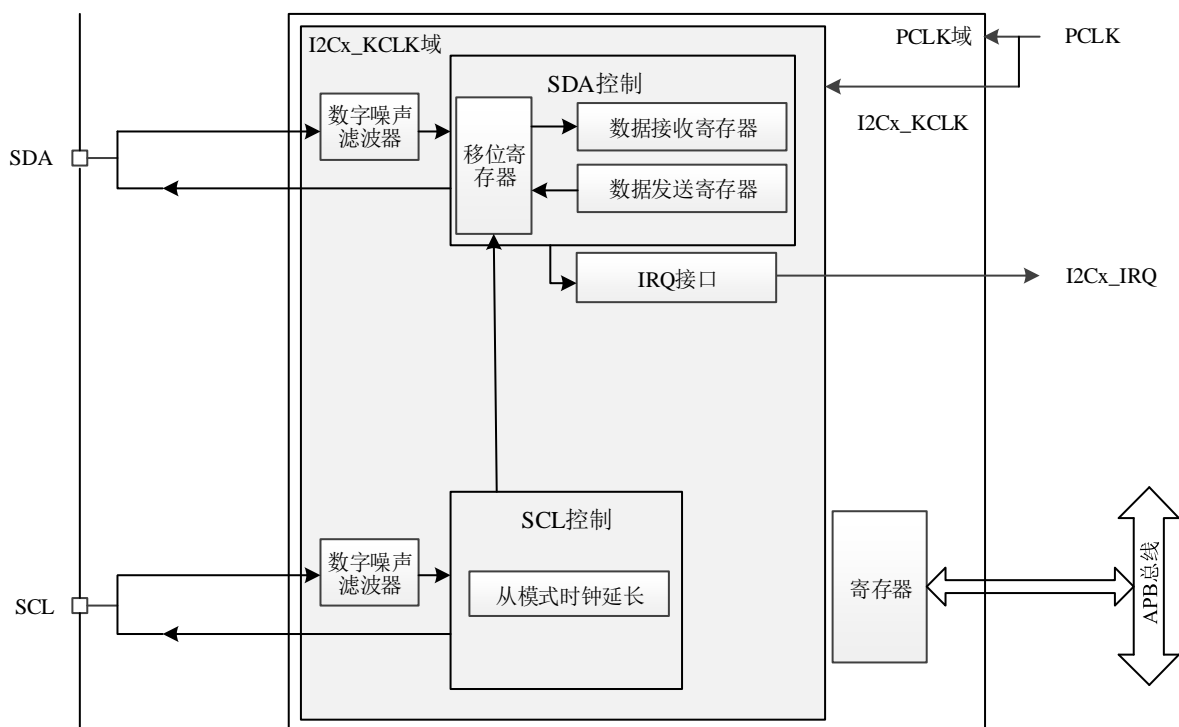
### 20.2 I2C 主要特性

- 支持从模式
- 支持标准模式 (最高 100KHz)、快速模式 (最高 400KHz)、超快速模式 (最高 1MHz)
- 支持 7 位从地址和广播地址
- 从模式可配置时钟延长
- 可配置的数字噪声滤波器

### 20.3 I2C 功能描述

#### 20.3.1 I2C 框图

图 20-1 I2Cx 框图 (x=1)



### 20.3.2 I2C 引脚

表 20-1 I2C 引脚

引脚名称	引脚类型	说明
SCL	输入/输出	I2Cx 同步时钟引脚
SDA	输入/输出	I2Cx 数据引脚

### 20.3.3 I2C 时钟要求

I2C 内核的时钟由 PCLK 提供，I2Cx\_KCLK 周期  $t_{I2Cx\_KCLK}$  必须遵循以下条件：

$$t_{I2Cx\_KCLK} < (t_{LOW} - t_{FILTERS}) / 8 \text{ 且 } t_{I2Cx\_KCLK} < t_{HIGH}$$

其中  $t_{LOW}$ ：SCL 低电平时间；

$t_{HIGH}$ ：SCL 高电平时间；

$t_{FILTERS}$ ：噪声滤波器使能时，该值为数字噪声滤波器引入的延时总和（数字滤波器延时为  $DNF \times t_{I2Cx\_KCLK}$ ）。

### 20.3.4 I2C 初始化

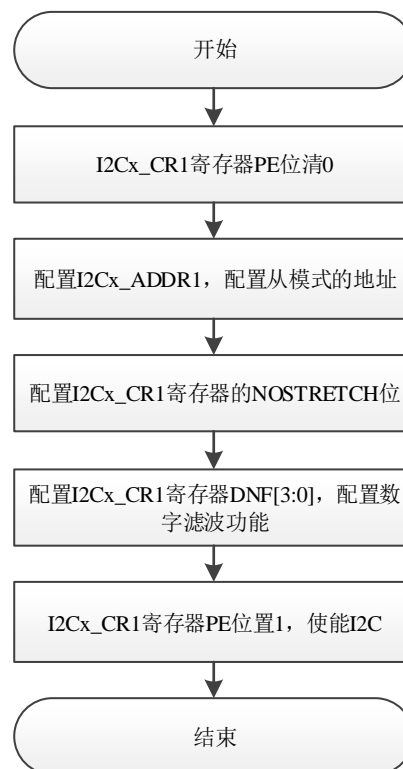
#### 使能和关闭外设

在 RCC 中配置使能 I2C 时钟，然后可通过将 I2Cx\_CR1 寄存器中的 PE 位置 1 以使能 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](#)。

I2C 初始化流程如下：

图 20-2 I2C 初始化流程



## 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。

- 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 I2Cx\_CR1 寄存器中的 DNF[3:0] 位来使能数字滤波器。使能数字滤波器后，SCL 或 SDA 的电平只有在电平稳定时间超过 DNF 个 I2Cx\_KCLK 时钟周期后才会认为电平有效，从而抑制在 1~15 个 I2Cx\_KCLK 时钟周期的噪声。

## 时钟低电平延长

- 使能时钟延长 (NOSTRETCH 为 0)

在默认模式下，I2C 从设备会在以下情况下延长 SCL 时钟线：

当接收到的地址和从地址匹配时，ADDR 标志置 1，在发送完 ACK 后，SCL 时钟线被延长。向 I2Cx\_ICR 寄存器的 ADDR CF 位写 1 清除 ADDR 标志，将释放 SCL 时钟线。

在发送过程中，当发送数据寄存器 I2Cx\_TDR 和移位寄存器中均无有效数据时，SCL 时钟线被延长。向 I2Cx\_TDR 寄存器中写入数据时，将释放 SCL 时钟线。

在接收过程中，当接收数据寄存器 I2Cx\_RDR 中还有未被读取的数据，且移位寄存器又接收完成新数据时，SCL 时钟线被延长。读取 I2Cx\_RDR 时，将释放 SCL 时钟线。

- 禁止时钟延长 (NOSTRETCH 为 1)

当 I2Cx\_CR1 寄存器中的 NOSTRETCH 为 1 时，I2C 从设备不会延长 SCL 时钟线。

ADDR 标志置 1 时，不会延长 SCL 时钟线。

### 20.3.5 软件复位

可通过将 I2Cx\_CR1 寄存器中的 PE 位清 0 来执行软件复位。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：

- I2Cx\_CR2 寄存器： NACK；
- I2Cx\_ISR 寄存器： BUSY、TXE、TXIS、RXNE、ADDR、DIR、NACKF、STOPF、BERR 和 OVR。

### 20.3.6 通信流程

I2C 支持 7 位地址，用户通过地址寄存器 I2Cx\_ADDR1 配置自身从地址。I2C 接收到的地址和自身地址一致时，硬件发送应答位 ACK，ADDR 标志置 1，DIR 传输方向更新，如果 I2Cx\_CR1 寄存器中的 EVTIE 位置 1，同时会生成中断。从地址匹配后，接收到停止位，STOPF 标志置 1，如果 I2Cx\_CR1 寄存器中的 EVTIE 位置 1，同时会生成中断。

#### 发送数据时序

当 I2Cx\_TDR 寄存器为空时，TXE 置 1。在发送模式下，当从地址匹配且 I2Cx\_TDR 寄存器为空时，TXIS 将置 1，如果 I2Cx\_CR1 寄存器中的 BUFIE 位置 1，同时会生成中断。I2Cx\_TDR 寄存器中写入待发送的数据时，TXIS 和 TXE 位将被清 0。

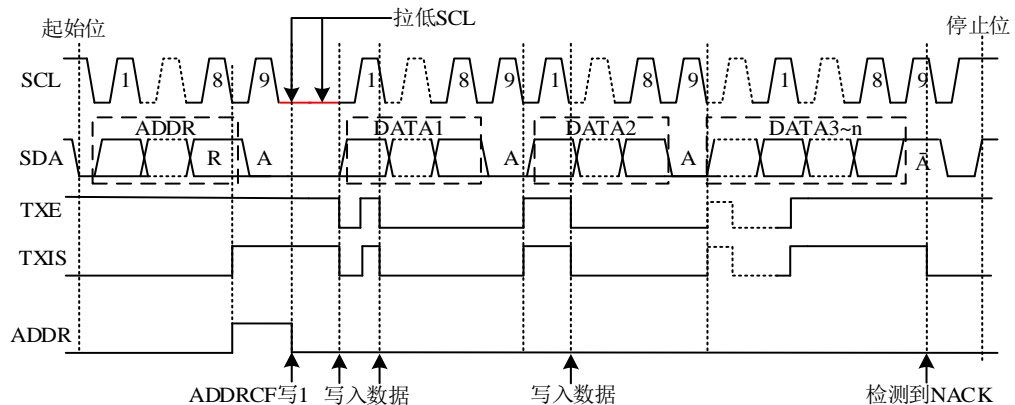
在数据传输前，如果 I2Cx\_TDR 寄存器已有数据存在，可以通过将 I2Cx\_ISR 寄存器中的 TXE 置 1，使原数据失效重新写入新的数据。

数据传输过程中，接收到 NACK 时，I2Cx\_ISR 寄存器中的 NACKF 位将置 1，如果 I2Cx\_CR1 寄存器中的 EVTIE 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。

使能低电平延长（NOSTRETCH 清 0）发送数据时序图：



图 20-3 使能时钟低电平延长发送数据

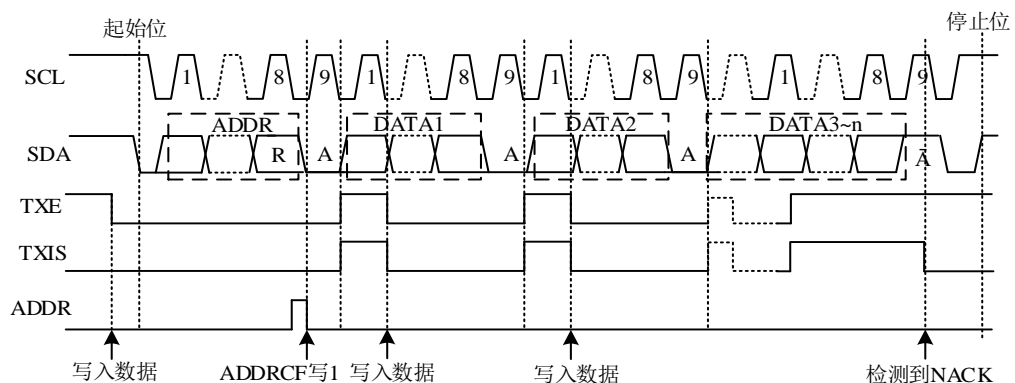


操作流程如下：

- 1) 等待 ADDR 置 1，查询 DIR 值，确认数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
- 2) 等待 TXIS 置 1，DATA1 写入 I2Cx\_TDR 寄存器；
- 3) 重复步骤 2，依次将 DATA2~DATA<sub>n</sub> 写入 I2Cx\_TDR 寄存器；
- 4) 等待 STOPF 置 1，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址匹配事件前将待发送的第一个字节数据写入 I2Cx\_TDR 寄存器，如下图：

图 20-4 禁止时钟低电平延长发送数据



操作流程如下：

- 1) 等待 TXE 置 1，DATA1 写入 I2Cx\_TDR 寄存器；
- 2) 等待 ADDR 置 1，查询 DIR 值，确认数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
- 3) 等待 TXIS 置 1，DATA2 写入 I2Cx\_TDR 寄存器；

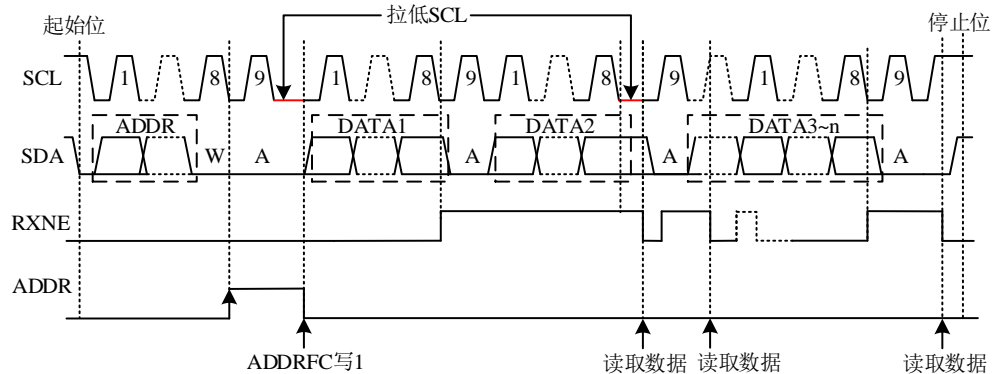
- 4) 重复步骤 3，依次将 DATA3~DATA<sub>n</sub> 写入 I2Cx\_TDR 寄存器；
- 5) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

### 接收数据时序

I2Cx\_RDR 寄存器接收到数据，I2Cx\_ISR 中的 RXNE 将置 1，如果 I2Cx\_CR1 中的 BUFIE 置 1，同时会生成中断。读取 I2Cx\_RDR 时，RXNE 标志将被清 0。

通常情况下应由主设备控制数据的传输，从设备若需主动结束数据的接收，可以将 I2Cx\_CR2 控制寄存器中的 NACK 位置 1，从设备在完成当前接收的字节后发送 NACK 响应位，自动释放 SCL 和 SDA，主设备在接收到 NACK 响应位后，将自动发送停止位结束本次数据传输。使能低电平延长功能（NOSTRETCH 为 0）接收数据时序图：

图 20-5 使能时钟低电平延长接收数据



操作流程如下：

- 1) 等待 ADDR 置 1，查询 DIR 值，确认数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
- 2) 等待 RXNE 置 1，从 I2Cx\_RDR 寄存器读取 DATA1；
- 3) 重复步骤 2，依次从 I2Cx\_RDR 寄存器读取 DATA2~DATA<sub>n</sub>；
- 4) 等待 STOPF 置 1，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：



寄存器 I2Cx\_TDR 和移位寄存器中均无有效数据，将产生下溢错误，并自动发送数据 0xFF。

## 20.4 I2C 中断

表 20-2 中断请求

中断事件	事件标志	中断使能位	事件标志或中断清除方法
接收数据寄存器非空	RXNE	BUFIE	读出 I2Cx_RDR 寄存器的数据
发送数据寄存器中断状态	TXIS		向 I2Cx_TDR 寄存器写入数据
地址匹配	ADDR	EVTIE	I2Cx_ICR 寄存器 ADDRCF 置 1
接收到 NACK	NACKF		I2Cx_ICR 寄存器 NACKCF 置 1
接收到停止位	STOPF		I2Cx_ICR 中 STOPCF 置 1
总线错误	BERR	ERRIE	I2Cx_ICR 寄存器 BERRCF 置 1
上溢、下溢错误	OVR		I2Cx_ICR 寄存器 OVRCF 置 1

## 20.5 I2C 寄存器

I2C 寄存器支持 32 位访问。

表 20-3 I2Cx 基地址 (x=1)

外设	基地址
I2C1	0x4000 5400

### 20.5.1 控制寄存器 1 (I2Cx\_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.												GCEN	Res.	NO STRETCH	Res.
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DNF[3:0]				ERRIE	Res.			EVTIE	Res.	BUFIE	PE
				rw	rw	rw	rw	rw				rw		rw	rw

位/位域	名称	描述
31:20	保留	写入无效
19	GCEN	广播地址使能 0: 禁止 1: 使能
17	NOSTRETCH	时钟低电平延长控制 0: 使能 1: 禁止 <i>注意: 该位只能在I2C禁止状态 (PE为0) 时写入。</i>
16:13	保留	写入无效
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]×I2Cx_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×I2Cx_KCLK的

噪声

:

1110: 开启数字噪声滤波器, 滤除脉宽小于 $14 \times I2Cx\_KCLK$ 的噪声

1111: 开启数字噪声滤波器, 滤除脉宽小于 $15 \times I2Cx\_KCLK$ 的噪声

注意: 该位域只能在I2C禁止状态 (PE为0) 时写入。

7                      ERRIE                      错误中断检测使能  
0: 禁止  
1: 使能

6:4                      保留                      写入无效

3                      EVTIE                      事件中断使能  
0: 禁止  
1: 使能

1                      BUFIE                      缓冲器中断使能  
0: 禁止  
1: 使能

0                      PE                      I2C使能  
0: 禁止  
1: 使能

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。

## 20.5.2 控制寄存器 2 (I2Cx\_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
NACK	Res.														
rs															

位/位域	名称	描述
31:16	保留	写入无效
15	NACK	<p>NACK发送控制</p> <p>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</p> <p>1：在当前接收的字节后发送NACK</p> <p><i>注意：当NOSTRETCH为1，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>
14:0	保留	写入无效

### 20.5.3 地址寄存器 1 (I2Cx\_ADDR1)

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.								ADDR1[7:1]							Res.
								rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:8	保留	写入无效
7:1	ADDR1[7:1]	<p>I2C设备地址</p> <p>ADDR1[7:1]：7位地址</p>
0	保留	写入无效

### 20.5.4 中断和状态寄存器 (I2Cx\_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.															DIR
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.				OVR	Res.	BERR	Res.		STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r					r		r			r	r	r	r	r	rs

位/位域	名称	描述
31:17	保留	写入无效
16	DIR	数据传输方向 在发生地址匹配事件ADDR置1时更新。当检测到起始位、停止位或PE清0时，该位自动清0 0: I2C接收数据 1: I2C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。
14:11	保留	写入无效
10	OVR	溢出错误标志 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	保留	写入无效
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>
7:6	保留	写入无效
5	STOPF	停止位标志 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。



注意：当PE清0，该位自动清0。

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>注意：当PE清0，该位自动清0。</p>
3	ADDR	<p>地址匹配标志</p> <p>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>注意：当PE为0，该位自动清0。</p>
2	RXNE	<p>接收数据寄存器非空</p> <p>I2Cx_RDR寄存器非空，该标志由硬件置1；读取I2Cx_RDR数据后，该位自动清0。</p> <p>注意：当PE清0，该位自动清0。</p>
1	TXIS	<p>发送中断状态</p> <p>当处于发送模式时，当I2Cx_TDR为空，该标志置1；将数据写入I2Cx_TDR后，该位自动清0。</p> <p>注意：当PE清0，该位自动清0。</p>
0	TXE	<p>发送数据寄存器为空</p> <p>当I2Cx_TDR寄存器为空时,该标志由硬件置1，将数据写入I2Cx_TDR后，该位自动清0。</p> <p>该位可软件写1，清除I2Cx_TDR寄存器中已写入的数据。</p> <p>注意：当PE清0时，该位硬件置1。</p>

### 20.5.5 中断清除寄存器（I2Cx\_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	Res.	BERRCF	Res.		STOPCF	NACKCF	ADDRCF	Res.		
					w		w			w	w	w			

位/位域	名称	描述
31:11	保留	写入无效
10	OVRCF	溢出标志清除 该位写入1，I2Cx_ISR寄存器中的OVR标志清0。
9	保留	写入无效
8	BERRCF	总线错误标志清除 该位写入1，I2Cx_ISR寄存器中的BERR标志清0。
7:6	保留	写入无效
5	STOPCF	停止位标志清除 该位写入1，I2Cx_ISR寄存器中的STOPF标志清0。
4	NACKCF	NACK位标志清除 该位写入1，I2Cx_ISR寄存器中的NACKF标志清0。
3	ADDRCF	地址匹配标志清除 该位写入1，I2Cx_ISR寄存器中的ADDR标志清0。
2:0	保留	写入无效

### 20.5.6 接收数据寄存器（I2Cx\_RDR）

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                  写入无效

7:0                  RXDATA[7:0]                  接收数据寄存器

## 20.5.7      发送数据寄存器 (I2Cx\_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.								TXDATA[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
------	----	----

31:8	保留	写入无效
------	----	------

7:0	TXDATA[7:0]	发送数据寄存器
-----	-------------	---------

注意: 仅可在I2Cx\_ISR寄存器的TXE为1时, 该位域才可以写入数据。

## 21 通用异步收发器 (UART)

### 21.1 简介

异步收发器 UART 支持全双工或半双工，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。

UART 支持单线半双工通信。

提供了 2 个异步收发器 UART1/2。

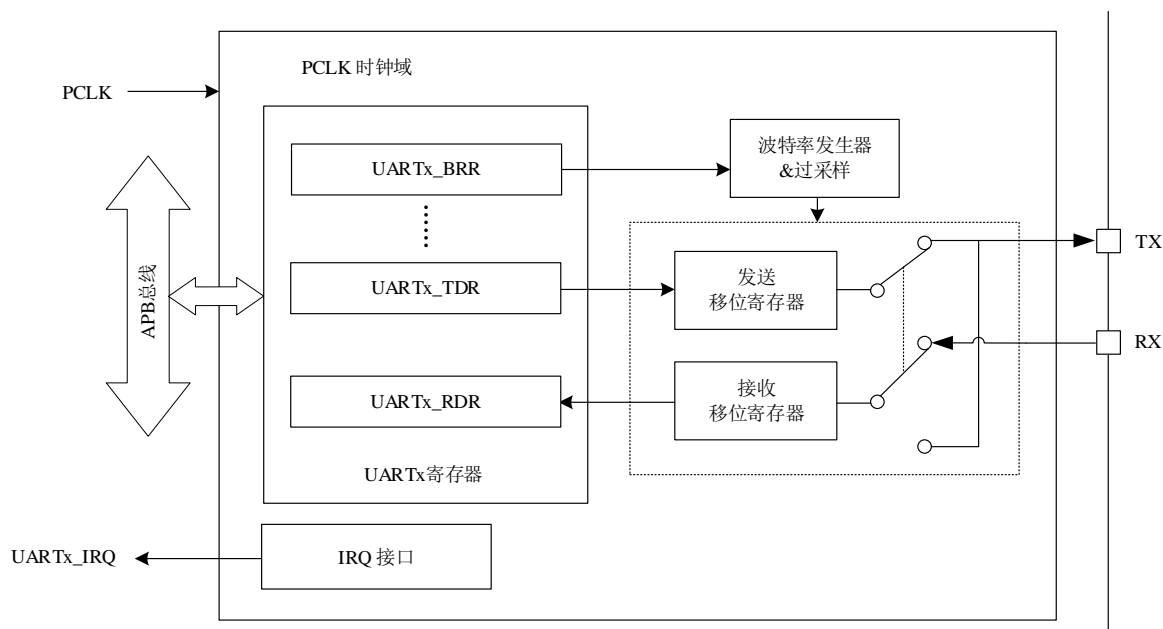
### 21.2 UART 主要特性

- 可编程波特率发生器
  - 系统时钟 48MHz，16 倍过采样，最高速度可达 3Mbps
  - 小数级波特率发生器
- 帧格式
  - 数据长度：8 位、9 位
  - 停止位位数：1 位或 2 位
  - 奇校验、偶校验、无校验
  - MSB 或 LSB 优先
- 单线半双工通信
- TX/RX 引脚可交换
- 发送/接收的信号极性可独立控制

## 21.3 UART 功能描述

### 21.3.1 UART 框图

图 21-1 UARTx 框图 (x=1、2)



### 21.3.2 UART 引脚

UART 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)。支持 TX 引脚 RX 引脚互换功能，在 UARTx\_CR2 的 SWAP 位配置。

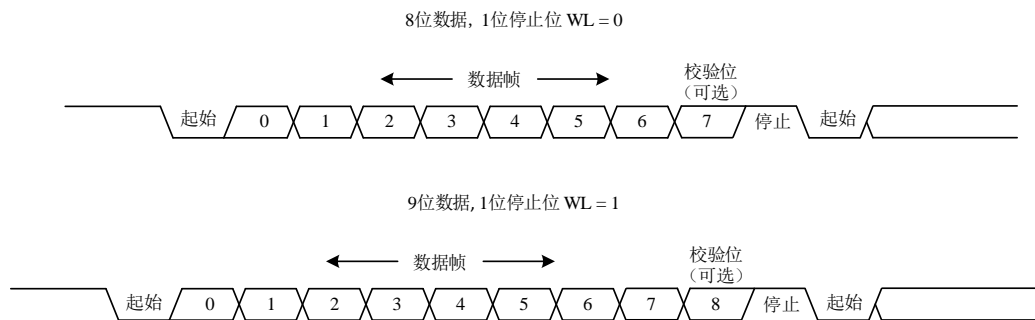
### 21.3.3 UART 帧格式

可通过 UARTx\_CR1 寄存器中的 WL 位编程字符长度，参见：[帧格式示意图](#)。

- 8 位字符长度：WL=0
- 9 位字符长度：WL=1

默认情况下，TX 和 RX 引脚在起始位期间都是低电平，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 [UART 控制寄存器 2 \(UARTx\\_CR2\)](#) 中 TXIVC, RXIVC 位的说明。详见下图：

图 21-2 帧格式示意图



### 21.3.4 UART 校验控制

表 21-1 UART 校验控制

字符长度		校验控制 PEN	UART 帧格式
8 位	WL=0	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	WL=1	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

#### 偶校验:

使能偶校验时, 一个字符中 (含校验位) “1” 的总数为偶数。(UARTx\_CR1 寄存器中 PTS 位=0)。

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, 一个字符中 (含校验位) “1” 的总数为奇数。(UARTx\_CR1 寄存器中 PTS 位=1)。

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”。

#### 接收检测奇偶校验

UARTx\_CR1 寄存器中的 PEN 位置位使能奇偶校验, 使能后, 如果奇偶校验失败, 则 UARTx\_ISR 寄存器中的奇偶校验错误标志 PE 位置位, 如果 UARTx\_CR1 寄存器中 PEIE 位置位, 则触发中断。向 UARTx\_ICR 寄存器中的 PECF 位写 1, 可清零 PE 标志。

#### 发送生成奇偶校验

UARTx\_CR1 寄存器中的 PEN 位置位使能奇偶校验, 使能后, 数据寄存器中数

据的 MSB 会替换为奇偶校验位。如果选择偶校验 (PTS=0)，则一个字符中“1”的总数为偶数；如果选择奇校验 (PTS=1)，则一个字符中“1”的总数为奇数。

### 21.3.5 UART 发送器

根据 WL 的配置，发送的字符长度可以是 8 位或 9 位。发送使能位 (TE) 置位即可使能发送器。发送移位寄存器的数据由 TX 引脚输出。

#### 字符发送

通过 UARTx\_CR1 寄存器的 TE 位置位使能发送器。

UART 发送期间，TX 引脚首先移出数据的最低有效位 (默认 LSB 优先)。发送数据寄存器 UARTx\_TDR 为内部总线和发送移位寄存器之间提供缓冲区。

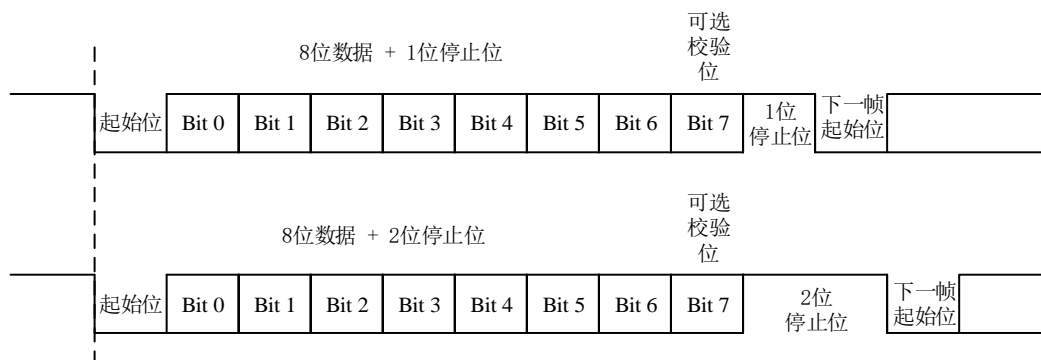
每个字符的发送都以起始位开始，以停止位结束，停止位的位数在 UARTx\_CR2 寄存器的 STOPBIT 位域可配置为 1 或 2 位。

#### 停止位

可以在 UARTx\_CR2 的 STOPBIT 中编程停止位的位数，可配置为 1 位、2 位。

下图为不同停止位配置示例：

图 21-3 可编程停止位示例



#### 字符发送配置步骤

- 1) 配置 UARTx\_BRR 寄存器选择通信波特率；
- 2) 配置 UARTx\_CR1 中的 WL 位定义字符长度；
- 3) 配置 UARTx\_CR2 中的 STOPBIT 设置停止位数；
- 4) 通过向 UARTx\_CR1 寄存器中的 UE 位写入“1”使能 UART；
- 5) 将 UARTx\_CR1 中的 TE 位置位，使能发送器；
- 6) 在 UARTx\_TDR 寄存器中写入准备发送的数据；向 UARTx\_TDR 写入数据会将 TXE 标志清零，表示 UARTx\_TDR 发送寄存器中数据已满。
- 7) 当最后一个待发送的数据写入 UARTx\_TDR 寄存器后，需等待发送完成直

到 UARTx\_ISR 中的发送完成标志位 TC 置位。TC 置位时表示最后一个字符已发送完成。

### 字符发送过程

UARTx\_TDR 寄存器写入 1 个字符后，发送数据空标志 TXE 位清零。

发送数据寄存器 UARTx\_TDR 为空时，TXE 标志位自动置位，此时表示：

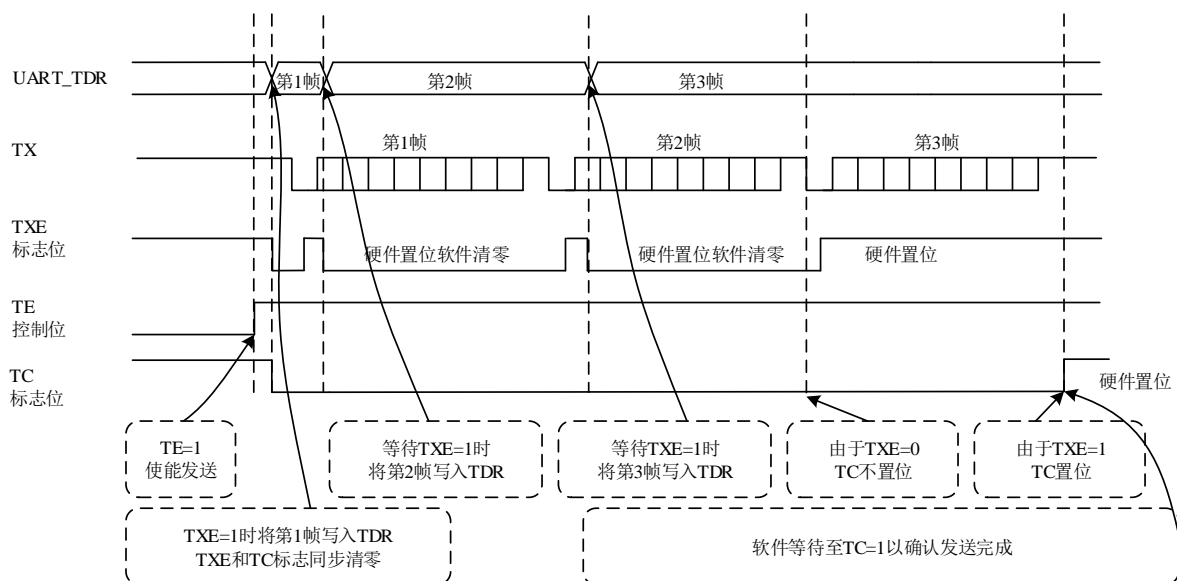
- 数据已由 UARTx\_TDR 寄存器移入移位寄存器，开始发送数据；
- 可以在 UARTx\_TDR 寄存器中写入数据，前一数据不会被覆盖。

TXEIE 位置位使能中断时，TXE 标志置位会触发中断。

在发送未进行时 (TE=0)，写入 UARTx\_TDR 寄存器后，发送数据非空标志 TXE 位清零，数据发送开始后，TXE 位立即置位，指示可以写入下一数据。

发送时的 TC 和 TXE 的动作请参见下图：

图 21-4 发送过程 TC 和 TXE 的动作



### 21.3.6 UART 接收器

UART 可接收 8 位或 9 位长度的字符，可在 UARTx\_CR1 寄存器中 WL 位配置。

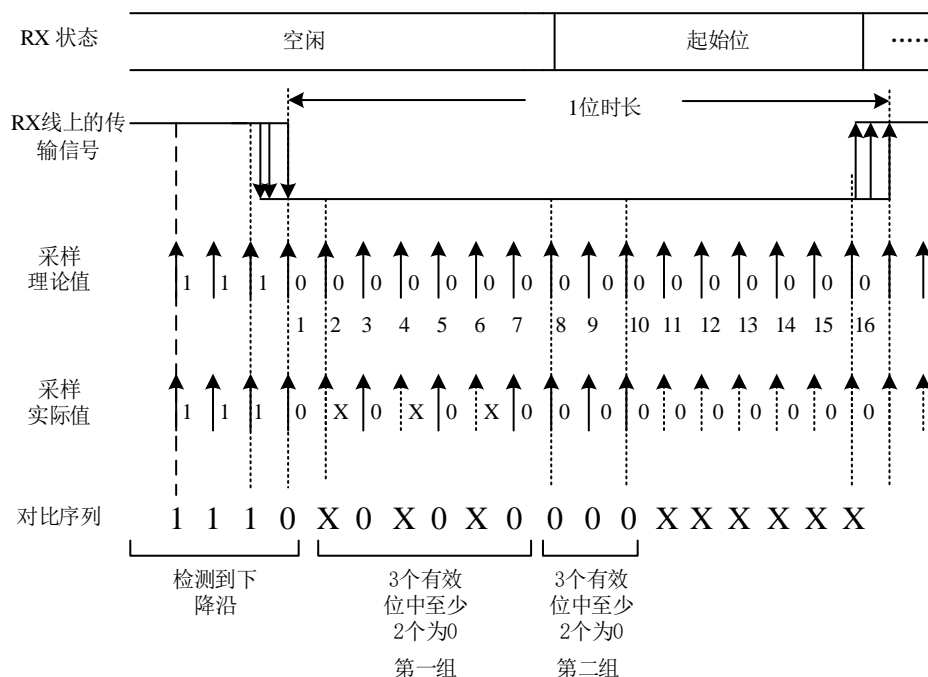
#### 起始位检测

在 UART 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 0 X 0 0 0 X X X X X X



图 21-5 起始位检测



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”，第二组 3 个采样位，第 8、9 和第 10 位仍检测到全为“0”，可确认接收到起始位。

满足以下条件时，检测到起始位的同时，噪声标志位 NOISE 置位：

- 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”。

上述描述包含了检测到起始位无噪声，和检测到起始位含噪声两种情况，如果两种情况均不满足，则起始位检测失败，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 UARTx\_BRR 寄存器选择通信波特率。
- 2) 配置 UARTx\_CR1 中的 WL 位定义字符长度。
- 3) 配置 UARTx\_CR2 中的 STOPBIT 位域设置停止位数。
- 4) 向 UARTx\_CR1 寄存器中的 UE 位写入“1”使能 UART。
- 5) UARTx\_CR1 寄存器 RE 位置位，使能接收。

接收字符时：

- 接收非空标志 RXNE 置位时，表示接收移位寄存器中的数据已经传送到 UARTx\_RDR 寄存器中，指示已接收到数据。

由于 UARTx\_RDR 寄存器中仅有 1 字符空间，读 UARTx\_RDR 寄存器时 RXNE 标志位清零。

- 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。

### 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 位置位，如果中断使能，即 UARTx\_CR1 中的 RXNEIE 或 EIE 位置位，则触发中断。

通过向 UARTx\_ICR 寄存器中的 ORECF 位写 1 来清零 ORE 位。

可以通过将 UARTx\_CR3 中的 ORED 位置位禁止上溢错误检测，禁止后 ORE 标志保持为 0。

每接收到一个字符后，接收非空标志位 RXNE 置位。当 RXNE 标志为 1 时，又接收到新的字符，就会产生上溢错误。上溢错误标志 ORE 位清零前，数据无法从移位寄存器传到接收数据寄存器 UARTx\_RDR 中。

发生上溢错误时有以下动作：

- 上溢错误标志 ORE 位置位；
- 移位寄存器被新数据覆盖，ORE 位清零前接收到的数据都将丢失。
- 上溢错误发生后，在 ORE 位清零前，UARTx\_RDR 寄存器中的内容不会被覆盖。读取 UARTx\_RDR 寄存器时，得到溢出前接收到的最后一个数据。
- UARTx\_CR1 中的 RXNEIE 或 EIE 位置位时，会触发中断。

ORE 位置位时表示至少已有一个数据丢失。

### 过采样模式

接收器配备了 16 倍过采样功能，可在噪声环境中提取有效数据。采样时钟是波特率时钟的 16 倍，参见 [图：数据采样](#)。对每个 bit 采样 16 次，此情况下最大通信速率为  $UARTx\_PCLK/16$ 。

可通过 UARTx\_CR3 寄存器中的 OBS 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 标志位置位。

- 在接收位的中心进行 1 次采样，无 NOISE 标志输出。

根据应用场景可选：

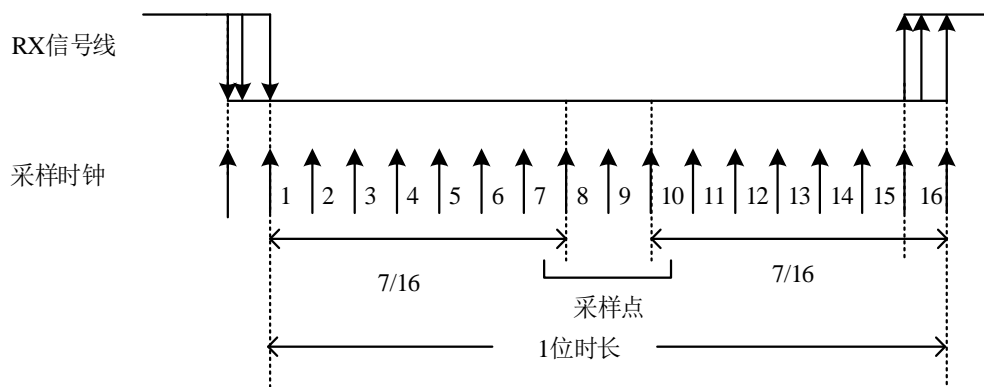
- 噪声较大的环境下，建议选择三次采样多数表决法(OBS=0)；如果检测到了噪声，则表示采样过程中有噪声干扰，为保证数据准确，推荐丢弃检测到噪声的数据；
- 在线路无噪声影响时，可选择单次采样法(OBS=1)，可以提高接收容差(参见：[UART 接收容差](#))。

接收数据检测到噪声时：

- 在接收非空标志 RXNE 位置位的上升沿时 NOISE 位置位。
- 无效数据从移位寄存器传送到 UARTx\_RDR 寄存器。
- NOISE 位置位时，如果 UARTx\_CR3 寄存器中的 EIE 位置位则触发中断。NOISE 位置位时，RXNE 位同时置位，如果 RXNEIE 位置位，则触发中断。

使用 UARTx\_ICR 寄存器中的 NOISECF 位写“1”清零 NOISE 位。

图 21-6 数据采样 (OBS=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 UARTx\_RDR 寄存器。

- FE 位置位时，如果 UARTx\_CR3 寄存器中的 EIE 位置位，则触发错误中断。FE 置位时，RXNE 位也被置位，如果 RXNEIE 位置位会触发接收非空中断。

向 UARTx\_ICR 寄存器中的 FECF 位写 1，清零 FE 位。

### 配置接收停止位

在 UARTx\_CR2 中的 STOPBIT 位域中，配置预计接收的停止位的位数。

采样逻辑电平的方式选择 3 次采样即 OBS=0 时，有以下几个选项：

- 1 位停止位：在第 8、第 9 和第 10 个采样沿采样 1 位停止位。
- 2 位停止位：在第一个停止位的第 8、第 9 和第 10 个采样沿采样。

采样逻辑电平的方式选择 1 次采样，即 OBS=1 时，有以下几个选项：

- 1 个停止位：在第一个停止位的中间采样一次。
- 2 个停止位：在第一个停止位的中间位置采样一次。

## 21.3.7 UART 波特率生成

接收和发送的波特率通过编程 UARTx\_BRR 寄存器得到。

计算公式如下：

标准 UART 的波特率，计算公式如下：

$$\text{Tx或Rx的波特率} = \frac{\text{UART\_PCLK}}{\text{BRR}}$$

BRR 是一个无符号小数，存储在 UARTx\_BRR 寄存器中。

**注意：** BRR 必须大于等于 16。

### 根据时钟频率和波特率计算 BRR

例 1：在 UARTx\_PCLK=8MHz 时得到 9600 的波特率

$$\text{BRR} = 8\,000\,000 / 9600$$

$$\text{BRR} = 0\text{d}833 = 0\text{x}341$$

例 2：在 UARTx\_PCLK=48MHz 时得到 921.6K 的波特率

$$\text{BRR} = 48\,000\,000 / 921\,600$$

$$\text{BRR} = 0\text{d}52 = 0\text{x}34$$

### 21.3.8 UART 接收容差

仅当时钟系统总偏差小于 UART 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：

$$DTR\_ERR + DQU\_ERR + DRX\_OSERR + DL\_ERR < \text{UART 接收容差}。$$

影响接收容差的条件有以下几点：

- 由 UARTx\_CR1 寄存器中的 WL 位设置的 8 或 9 位字符长度，使用 8 位字符长度时接收容差更大。
- UARTx\_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 BRR 不小于 16,反映到 UARTx\_BRR 寄存器中为“0x10”。BRR[3:0]如果等于“0000”则 BRR 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 UARTx\_CR3 寄存器中 OBS 位配置的单次采样（OBS=1）或 3 次采样多数表决（OBS=0）的方式，使用单次采样方式时接收容差更大。

### 21.3.9 UART 单线半双工通信

UART 支持单线半双工通信协议，通过将 UARTx\_CR3 寄存器中的 HDEN 位置位可开启单线半双工通信模式。

向 HDEN 位写 1 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。

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 UARTx\_CR2 中的 SWAP 为置位时，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。

## 21.4 UART 中断

表 21-2 UART 中断请求

中断事件	事件标志	中断使能位	事件标志或中断清除方法
发送数据寄存器空	TXE	TXEIE	向发送数据寄存器写入数据
接收数据寄存器非空	RXNE	RXNEIE	读取接收数据寄存器
发送数据寄存器空且移位寄存器空	TC	TCIE	向发送数据寄存器写入数据或 TCCF 位写 1
奇偶校验错误	PE	PEIE	PECF 位写 1
上溢错误	ORE	RXNEIE、EIE	ORECF 位写 1
检测到噪声	NOISE	EIE	NOISECF 写 1
检测到帧错误	FE		FECF 位写 1

## 21.5 UART 寄存器

UART 寄存器支持 32 位访问。

表 21-3 UARTx 基地址列表 (x=1、2)

UART 序号	基地址
UART1	0x4001 3800
UART2	0x4000 4400

### 21.5.1 UART 控制寄存器 1 (UARTx\_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.			WL	Res.	PEN	PTS	PEIE	TXE IE	TCIE	RXNE IE	Res.	TE	RE	Res.	UE
			rw		rw	rw	rw	rw	rw	rw		rw	rw		rw

位/位域	名称	描述
31:13	保留	写入无效。
12	WL	<p>字符长度</p> <p>此位用于配置字符长度。</p> <p>0: 8位数据位</p> <p>1: 9位数据位</p> <p><i>注意：在禁止UART (UE=0) 时才能写入此位。</i></p>
11	保留	写入无效
10	PEN	<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 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（参见图：帧格式示意图）。</p> <p>此位置位后，奇偶校验从下一字节生效。</p> <p>0: 禁止</p> <p>1: 使能</p>

*注意：使能奇偶校验，占用1个数据位。*

*在禁止UART（UE=0）时才能写入此位。*

9	PTS	<p>奇偶校验选择</p> <p>此位用于在奇偶校验使能时（PEN位置位）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><i>注意：在禁止UART（UE=0）时才能写入此位。</i></p>
8	PEIE	<p>奇偶校验错误中断使能</p> <p>0：禁止</p> <p>1：使能</p>
7	TXEIE	<p>发送数据寄存器空中断使能</p> <p>0：禁止</p> <p>1：使能</p>
6	TCIE	<p>传输完成中断使能</p> <p>0：禁止</p> <p>1：使能</p>
5	RXNEIE	<p>接收数据非空中断使能</p> <p>0：禁止</p> <p>1：使能</p>
4	保留	写入无效
3	TE	<p>发送器使能</p> <p>0：禁止</p> <p>1：使能</p>
2	RE	<p>接收器使能</p> <p>0：禁止</p> <p>1：使能</p>
1	保留	写入无效



- 0            UE            UART 使能
- 此位清零后，UART 配置保留；TDR、RDR 寄存器清零，UARTx\_ISR 的所有状态标志均被复位。
- 0: 禁止
- 1: 使能

### 21.5.2    UART 控制寄存器 2 (UARTx\_CR2)

偏移地址：0x04

复位值：0x0000 0000

**注意：**    在禁止 UART ( $UE=0$ ) 时才能修改此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												MSB FIRST	DATA IVC	TX IVC	RX IVC
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP BIT	Res.												
rw		rw													

位/位域	名称	描述
31:20	保留	写入无效
19	MSBFIRST	MSB优先 0: 发送/接收时LSB优先 1: 发送/接收MSB优先
18	DATAIVC	数据极性反相 0: 按正向逻辑处理发送/接收的数据 (1=H, 0=L) 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。)
17	TXIVC	TX引脚有效电平反相 0: TX引脚使用标准逻辑电平 ( $V_{DD}=1$ , $GND=0$ ) 1: TX引脚信号反向 ( $V_{DD}=0$ , $GND=1$ )
16	RXIVC	RX引脚有效电平反相 0: RX引脚信号使用标准逻辑电平 ( $V_{DD}=1$ , $GND=0$ ) 1: 对RX引脚信号反相 ( $V_{DD}=0$ , $GND=1$ )

15	SWAP	<p>交换TX/RX引脚</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UART交叉连接。</p>
14	保留	写入无效
13	STOPBIT	<p>停止位</p> <p>此位域用于编程停止位的位数。</p> <p>0: 1位停止位</p> <p>1: 2位停止位</p>
12:0	保留	写入无效

### 21.5.3 UART 控制寄存器 3 (UART<sub>x</sub>\_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			ORED	OBS	Res.							HDEN	Res.		EIE
			rw	rw								rw			rw

位/位域	名称	描述
31:13	保留	写入无效
12	ORED	<p>禁止接收上溢</p> <p>0: 使能上溢检测</p> <p>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置位</p> <p>1: 禁止上溢功能, ORE标志不会置位</p> <p>注意: 在禁止UART (UE=0) 时才能写入此位。</p>
11	OBS	采样方式选择

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，无噪声检测功能

*注意：OBS功能仅适用于数据位采样和停止位的采样，不适用于起始位的采样。*

*在禁止UART (UE=0) 时才能写入此位。*

10:4          保留          写入无效

3              HDEN          使能单线半双工

0: 禁止

1: 使能

*注意：在禁止UART (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 (UARTx\_ISR寄存器中FE=1、ORE=1或NOISE=1) 时，此位置位会触发中断。

0: 禁止

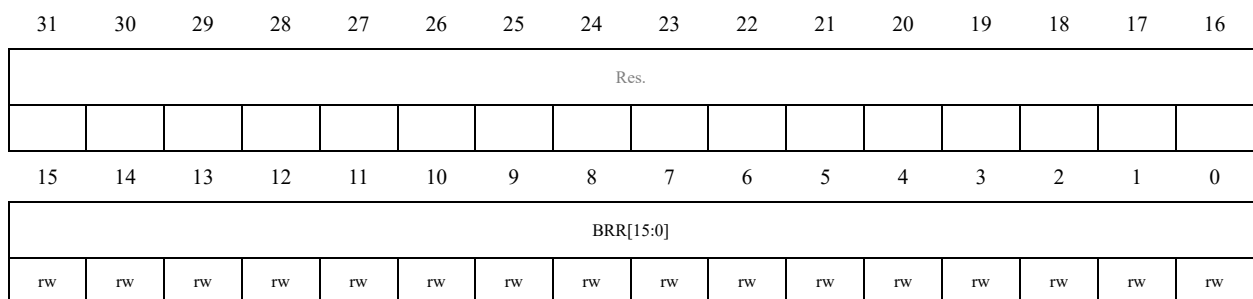
1: 使能

#### 21.5.4      **UART 波特率分频寄存器 (UARTx\_BRR)**

偏移地址: 0x0C

复位值: 0x0000 0000

*注意：在禁止 UART (UE=0) 时才能修改此寄存器*



位/位域	名称	描述
31:16	保留	写入无效
15:0	BRR[15:0]	BRR[15:0]位域为波特率分频值。

详情参见：[UART 波特率生成](#)。

## 21.5.5 UART 中断和状态寄存器 (UARTx\_ISR)

偏移地址：0x1C

复位值：0x0000 00C0

**注意：** 在禁止 UART ( $UE=0$ ) 时，TXE 和 TC 会自动置 1，其他位会自动清 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															BUSY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TXE	TC	RXNE	Res.	ORE	NOISE	FE	PE
								r	r	r		r	r	r	r

位/位域	名称	描述
31:17	保留	写入无效
16	BUSY	忙标志 当 RX 线路有接收的数据时，从检测到起始位开始，此位置位。 在接收结束时，无论是否成功，此位均清零。 0: UART 处于空闲状态 1: 正在接收数据
15:8	保留	写入无效
7	TXE	发送数据寄存器为空 当 UARTx_TDR 寄存器的内容已发送时，TXE 置位。如果 UARTx_CR1 寄存器中 TXEIE 位置位时，则触发中断。 写入 UARTx_TDR 寄存器时，此位清零。 0: 发送数据寄存器 UARTx_TDR 非空 1: 发送数据寄存器 UARTx_TDR 为空
6	TC	发送完成标志 此标志位置位，表示写入到 UARTx_TDR 中的最后一个数据已从移位寄存器中发出。如果已完成所有数据的发送，且 TXE 置位，则此位置位。

		<p>如果UARTx_CR1寄存器中TCIE位置位时，则会触发中断。</p> <p>通过向UARTx_ICR寄存器中的TCCF写1，或写UARTx_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：为确保最后一个字符完整发送，向TDR寄存器中写入最后一个数据后，在禁止UART (UE=0)，或禁止发送器 (TE=0)，或是进入stop模式前，需等待发送完成标志TC置位。</i></p>
5	RXNE	<p>接收数据寄存器非空</p> <p>当接收到数据时，RXNE位置位。如果UARTx_CR1寄存器中RXNEIE位置位时，则会触发中断。</p> <p>读UARTx_RDR寄存器时，此位清零。</p> <p>0: 未接收到数据</p> <p>1: 已接收到数据</p>
4	保留	写入无效
3	ORE	<p>上溢错误标志</p> <p>当RXNE标志为1时，又接收到新的字符，就会产生上溢错误，则此位置位。</p> <p>如果中断使能，UARTx_CR1寄存器中RXNEIE=1或EIE=1，则会触发中断。</p> <p>通过向UARTx_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：UARTx_CR3寄存器中的ORED位置位时，上溢错误检测被禁止，则此位清零（无上溢检测）。</i></p>
2	NOISE	<p>噪声检测标志</p> <p>当接收中检测到噪声时，此位自动置位。</p> <p>如果UARTx_CR1寄存器中的RXNEIE位置位，或UARTx_CR3寄存器中EIE位置位时，则会触发中断。</p> <p>向UARTx_ICR寄存器中的NOISECF写入1，此位清零。</p> <p>0: 未检测到噪声</p>

1: 检测到噪声

注意：当线路无噪声时，可以使用单次采样的方式（OBS=1）来提高UART的接收容差，此时无噪声检测，无NOISE标志输出。

1 FE

帧错误标志

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，此位置位。

如果 UARTx\_CR1 寄存器中 EIE=1，则会触发中断。

向 UARTx\_ICR 寄存器中的 FECF 写 1，此位清零。

0: 未检测到帧错误

1: 检测到帧错误

0 PE

奇偶校验错误标志

当在接收时发生奇偶校验错误时，此位置位。如果 UARTx\_CR1 寄存器中 PEIE 位置位时，则会触发中断。

向 UARTx\_ICR 寄存器中的 PECF 写入 1，此位清零。

0: 无奇偶校验错误

1: 奇偶校验错误

## 21.5.6 UART 中断标志清零寄存器（UARTx\_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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									TCCF	Res.		ORECF	NOISECF	FECF	PECF
									w			w	w	w	w

位/位域	名称	描述
31:7	保留	写入无效
6	TCCF	发送完成标志清零 向此位写1时，UARTx_ISR寄存器中TC标志清零。

5:4	保留	写入无效
3	ORECF	上溢错误标志清零 向此位写入1时，UARTx_ISR寄存器中ORE标志清零。
2	NOISECF	噪声检测标志清零 向此位写入1时，UARTx_ISR寄存器中NOISE标志清零。
1	FECF	帧错误标志清零 向此位写入1时，UARTx_ISR寄存器中FE标志清零。
0	PECF	奇偶校验错误标志清零 向此位写入1时，UARTx_ISR寄存器中PE标志清零。

### 21.5.7 UART 接收数据寄存器 (UARTx\_RDR)

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								RDR[8:0]							
								r	r	r	r	r	r	r	r

位/位域	名称	描述
31:9	保留	写入无效
8:0	RDR[8:0]	接收数据寄存器 在使能奇偶校验的时，接收数据的MSB为奇偶校验位。

### 21.5.8 UART 发送数据寄存器 (UARTx\_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]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:9	保留	写入无效
8:0	TDR[8:0]	发送数据寄存器 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。



## 22 串行外设接口（SPI）

### 22.1 简介

串行外设接口协议支持与外部设备进行同步全双工串行通信。

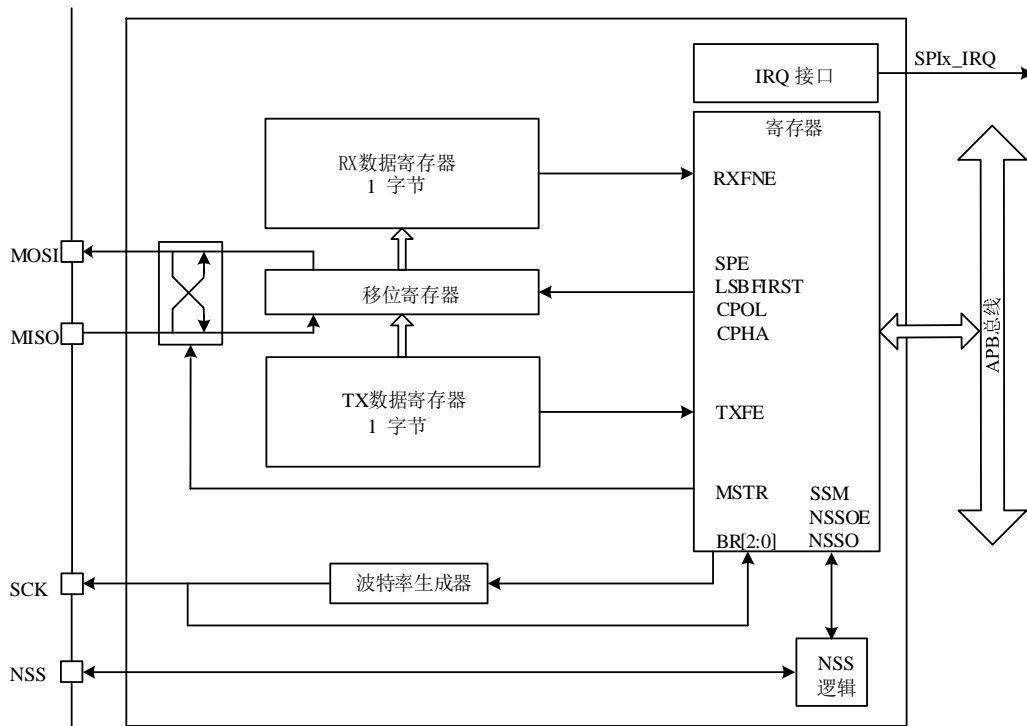
### 22.2 SPI 主要特性

- 支持 NSS、SCK、MISO 和 MOSI 标准四线的同步全双工通信
- 可配置为主机模式或从机模式
- 多主机模式功能
- 从模式支持软件片选
- 主机模式支持 7 种通信速率，可达 PCLK/2
- 可编程时钟极性和相位
- 传输数据大小固定为 8bit
- 可编程数据传输顺序：MSB 或 LSB
- 支持 SPI Motorola 协议
- 可触发中断的错误事件：主机模式冲突、上溢

## 22.3 SPI 功能说明

### 22.3.1 SPI 框图

图 22-1 SPIx 框图 (x=1)



### 22.3.2 SPI 引脚

表 22-1 SPI 引脚

引脚名称	信号类型	说明
MISO	输入/输出	主入/从出数据
MOSI	输入/输出	主出/从入数据
SCK	输入/输出	串行时钟
NSS	输入/输出	从机选择

### 22.3.3 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 22-2 单主单从应用

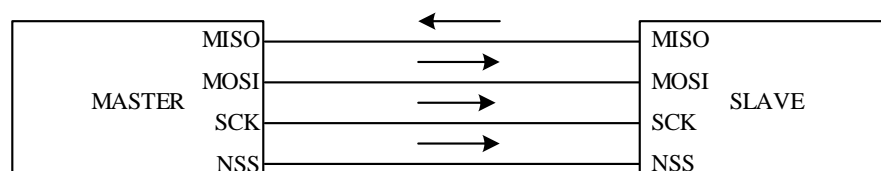
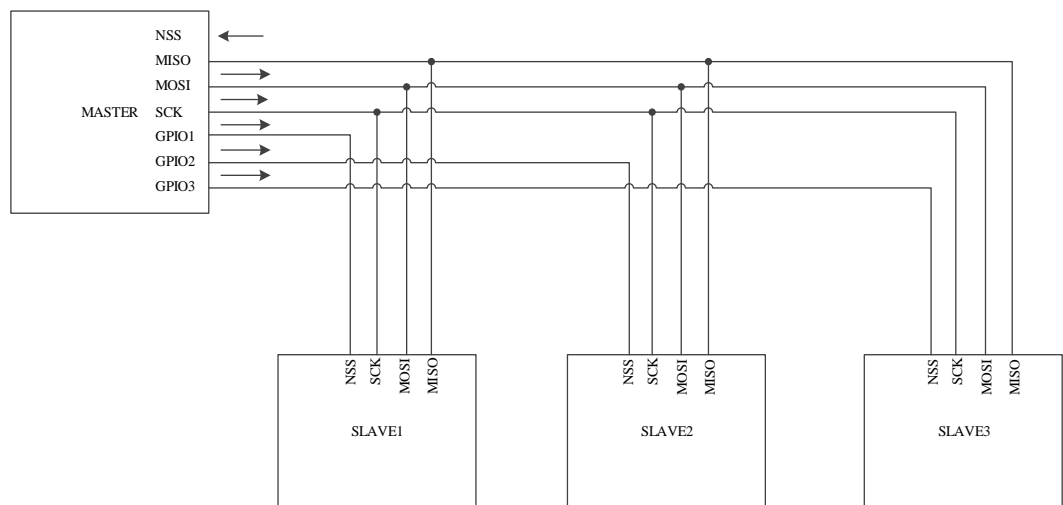


图 22-3 单主多从应用



当 SPIx\_CR1 寄存器的 MSTR 置 1 时，SPI 接口工作于主机模式。

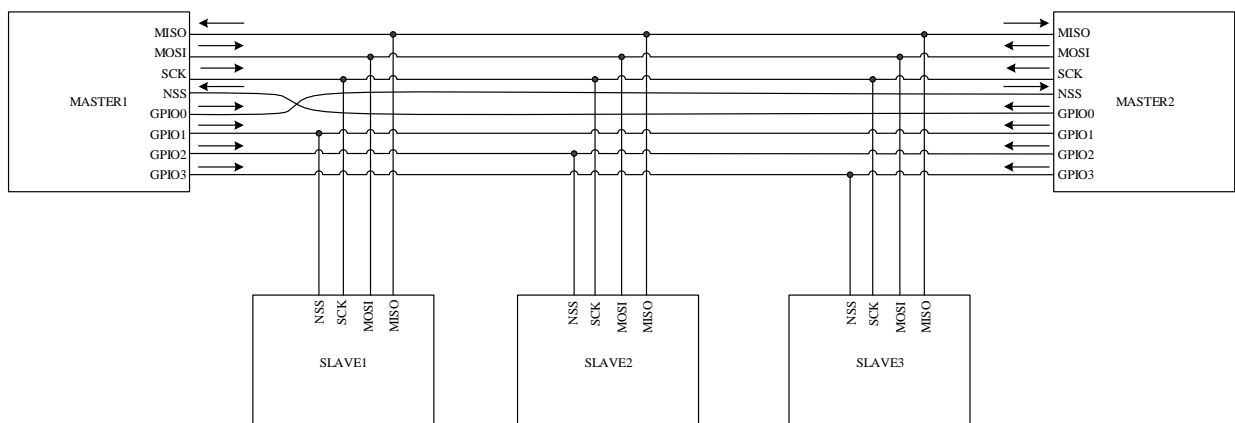
SCK 由主机提供，其波特率由 SPIx\_CR1 寄存器的 BR[2:0]位域控制，范围为  $f_{PCLK}/2 \sim f_{PCLK}/128$ 。

NSS 使用参见：[NSS 管理](#)。

#### 22.3.4 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 22-4 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](#)）。发生冲突后，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](#)。

### 22.3.5 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，从机应在主机时钟的第 1 个时钟沿到来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[图：单主单从应用](#)。

### 22.3.6 NSS 管理

在从机模式下，NSS 可配置为硬件片选和软件片选两种方式：

- 硬件片选方式下，SSM 需保持清 0，被 NSS 引脚片选的从机才能够与主机进行通信。
- 软件片选方式下，SSM 保持置 1，忽略 NSS 引脚的输入，从机始终保持被片选的状态。

在主机模式下，NSS 可用作输出或输入：

- 用作输出时，应用于单主机模式，可以通过 NSS 驱动单个从机的从机选择信号，也可以使用若干 GPIO 驱动多个从机的从机选择信号。
  - 若用 GPIO 作为片选引脚，NSSOE 保持置 1，按照 SPI 通信协议操作 GPIO 进行片选，然后清 0 NSSO 位后发数。
  - 若选择 NSS 作为片选引脚，NSSOE 保持置 1，按照 SPI 通信协议操作 NSSO 位进行片选。
- 用作输入时，NSS 检测总线占用冲突，用于多主机模式。

- NSS 引脚输入为高，NSSOE 保持清 0，SPI 为主机模式，按照 SPI 通信协议操作 GPIO 进行片选，然后清 0 NSSO 位后发数。
- NSS 引脚输入为低，NSSOE 保持清 0，SPI 进入主模式冲突错误。

NSS 的配置应用见下表：

表 22-2 NSS 配置

主从模式	NSS 引脚	NSSOE	NSSO	SSM	说明
从机	输入	0	1	0	从机 NSS 为输入，用于确定从机的选择是否有效。
				1	从机始终处于被选中状态，忽略 NSS 引脚输入。
主机	输入	0	0/1	0	NSS 输入高电平时，GPIO 片选后也要操作 NSSO 位。
	输出	1	0/1	0	NSS 为输出信号，NSS 输出电平由 NSSO 控制。

### 22.3.7 通信模式

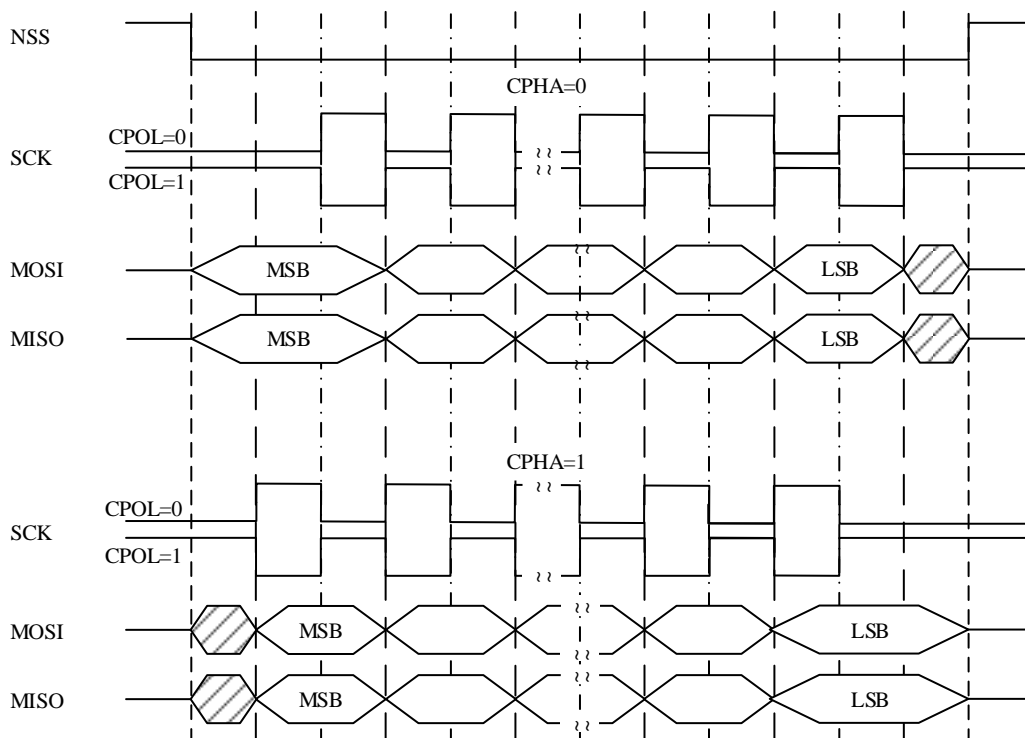
SPI 通信过程中，将同时执行接收和发送操作。数据线上的信息的移位和采样与串行时钟同步。四种通信格式取决于时钟相位和时钟极性，具体见下表。

表 22-3 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。

图 22-5 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。

### 22.3.8 状态标志

提供发送数据寄存器空状态、接收数据寄存器非空状态和总线忙状态，通过这些状态管理数据的发送和接收。

- 发送数据寄存器空标志 (TXFE)

当发送数据寄存器中没有数据时，TXFE 标志置 1。如果发送数据寄存器非空时，TXFE 标志由硬件自动清 0。

- 接收数据寄存器非空标志 (RXFNE)

当接收数据寄存器非空时，RXFNE 标志置 1。只有将接收数据寄存器中的数据读出，RXFNE 标志由硬件自动清 0。

- 忙标志 (BUSY)

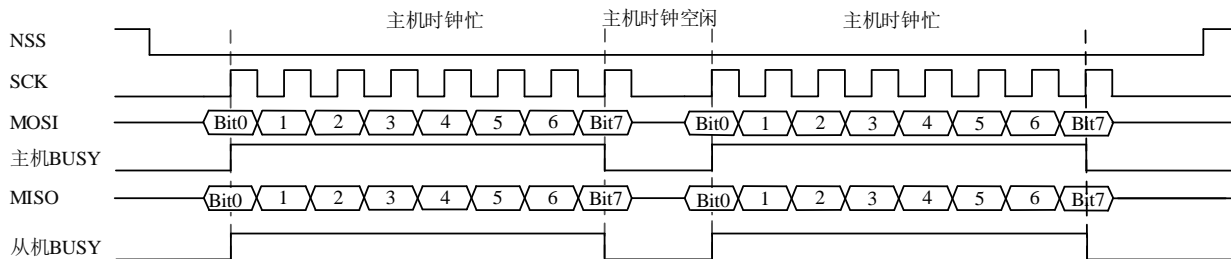
BUSY 标志由硬件置 1 和清 0。

当 BUSY 置 1 时，表示 SPI 总线上正在进行数据传输，BUSY 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，BUSY 标志将清 0：

- 在主机/从机应用下，上一个数据帧最后的采样时钟边沿与下一个数据帧起始的采样时钟边沿之间；
- 在主机应用下，发送数据寄存器已为空且最后一个数据发送完成时；
- 在多主机应用下，检测到主机模式冲突时。

图 22-6 主从模式 BUSY 状态



### 22.3.9 错误标志

以下错误标志中，如有一个置 1 且错误中断使能 ERRIE 位置 1，则将生成 SPI 错误中断。

#### ● 上溢标志（OVR）

当移位寄存器已接收到数据，但接收数据寄存器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数据寄存器中之前的值，新接收的数据将被丢弃。将 SPIx\_ICR 寄存器中 OVRDCF 位置 1，OVR 标志将同步清 0。

#### ● 主机模式冲突标志（MMF）

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。

为避免包含多个芯片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。

### 22.3.10 初始化 SPI

除 BR[2:0]和 NSSOE 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 MSTR 位，选择主从模式。
  - 配置 NSSOE 位，选择 NSS 状态。从机模式无需配置此项。
- 4) 将 SPIx\_CR1 寄存器的 SPE 位置 1。

### 22.3.11 数据发送和接收

#### 查询和中断方式

SPIx\_DR 的写访问可通过 TXFE 状态管理。当 TXFE 置 1 时，可以对 SPIx\_DR 执行写访问。

SPIx\_DR 的读访问可通过 RXFNE 状态管理。当 RXFNE 置 1 时，可以对 SPIx\_DR 执行读访问。

TXFE 和 RXFNE 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](#)。

为确保 SPI 最后一个数据通信的完整性，正确步骤如下：

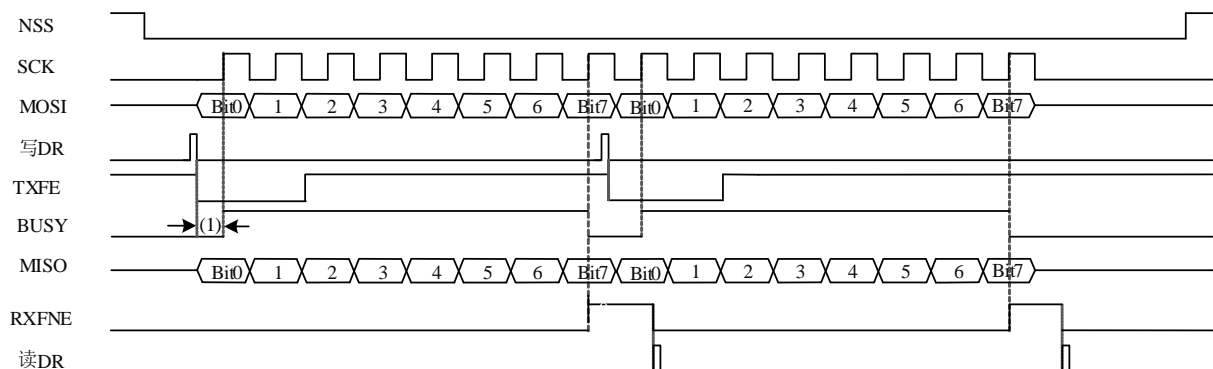
- 1) 等待发送数据寄存器为空 TXFE 为 1。
- 2) 等待 BUSY 为 0，最后一帧数据已传输完成。
- 3) 读出接收数据直至 RXFNE 为 0。

时钟信号由主机设备提供，直至发送数据寄存器和移位寄存器为空，之后时钟停止，直至主机再次发送数据。通信过程参见下图。



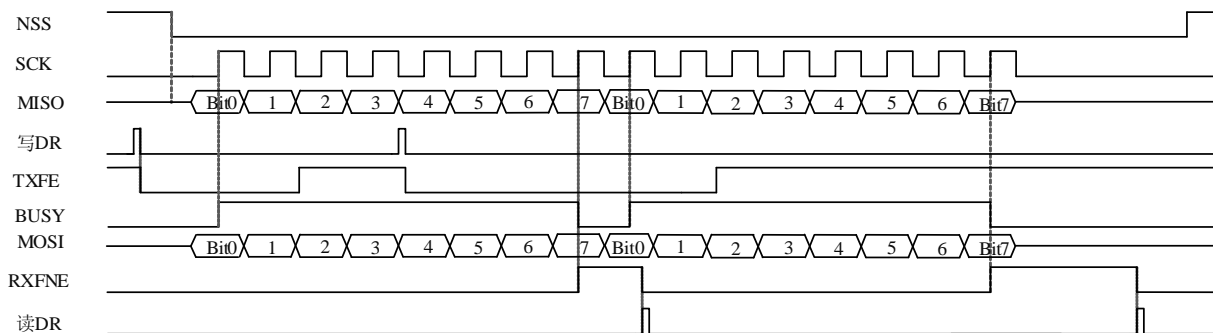
下图为 SPI 主机模式，传输 2 字节数据的通信时序，通信参数配置为：时钟空闲低电平，在第一个边沿采样。

图 22-7 主模式通信时序



下图为 SPI 从机模式，传输 2 字节数据的通信时序，通信参数配置为：时钟空闲低电平，在第一个边沿采样。

图 22-8 从模式通信时序



### 22.3.12 禁止 SPI

当禁止 SPI(SPE=0)时，SPIx\_ISR 寄存器和状态寄存器中所有标志恢复为上电默认值，清空 SPIx\_DR 寄存器。

## 22.4 SPI 中断

表 22-4 SPI 中断请求

中断事件	事件标志	使能控制位	清除方法
发送数据寄存器空	TXFE	TXFEIE	写 SPIx_DR 寄存器，使发送数据寄存器不为空
接收数据寄存器非空	RXFNE	RXFNEIE	读 SPIx_DR 寄存器，使接收数据寄存器为空
上溢	OVR	ERRIE	SPIx_ICR 寄存器 OVRCF 置 1
主机模式冲突	MMF		SPIx_ICR 寄存器 MMFCF 置 1

## 22.5 SPI 寄存器

SPI 寄存器支持 32 位访问。

表 22-5 SPI 基地址列表

SPI 序号	基地址
SPI1	0x4001 3000

### 22.5.1 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.						TXFEIE	Res.				RXFNEIE	Res.	ERRIE	Res.	
						rw					rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				SSM	NSSOE	Res.	LSB FIRST	Res.	MSTR	CPOL	CPHA	BR[2:0]			SPE
				rw	rw		rw		rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:26	保留	写入无效
25	TXFEIE	发送数据寄存器为空中断使能 0: 禁止 1: 使能
24:21	保留	写入无效
20	RXFNEIE	接收数据寄存器非空中断使能 0: 禁止 1: 使能
19	保留	写入无效
18	ERRIE	错误中断使能 如果使能错误中断，MMF、OVR 任一事件将产生中断。

		0: 禁止 1: 使能
17:12	保留	写入无效
11	SSM	从机模式 NSS 管理 0: 从机选择由 NSS 引脚输入决定是否被片选 1: 从机选择始终处于被选中状态，并忽略 NSS 引脚输入 <i>注意：此位仅在从机模式下设置有效，主机模式下硬件清0。</i>
10	NSSOE	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，从机模式下硬件清0。</i>
9	保留	写入无效
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	写入无效
6	MSTR	SPI 主从模式 0: 从机模式 1: 主机模式 <i>注意：应先配置时钟极性，再配置主从模式，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: $f_{CLK}/2$ 001: $f_{CLK}/4$ 010: $f_{CLK}/8$ 011: $f_{CLK}/16$ 100: $f_{CLK}/32$ 101: $f_{CLK}/64$ 110: $f_{CLK}/128$ 111: 保留 (写入无效, 保持为前一次的配置值) <i>注意: 仅在禁止SPI (SPE为0) 时才可对此位执行写操作。</i>
0	SPE	SPI 使能控制 0: 禁止 1: 使能

## 22.5.2 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.															NSSO
															rw

位/位域	名称	描述
31:1	保留	写入无效
0	NSSO	NSS 输出电平 该位结合 NSSOE 使用, NSSOE 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，从机模式下硬件置 1。*

### 22.5.3 SPI 中断和状态寄存器 (SPIx\_ISR)

偏移地址：0x0C

复位值：0x0000 0100

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.							TXFE	Res.	RXFNE	Res.	BUSY	Res.	OVR	Res.	MMF
							r		r		r		r		r

位/位域	名称	描述
31:9	保留	写入无效
8	TXFE	<p>发送数据寄存器空标志</p> <p>发送数据寄存器为空时置 1，否则由硬件清 0。</p> <p>0: 发送数据寄存器非空</p> <p>1: 发送数据寄存器为空</p> <p><i>注意：当 SPE 为 0，该位自动置 1。</i></p>
7	保留	写入无效
6	RXFNE	<p>接收数据寄存器非空标志</p> <p>接收数据寄存器非空时置 1，否则硬件清 0。</p> <p>0: 接收数据寄存器为空，未接收到数据</p> <p>1: 接收数据寄存器非空，已接收到数据</p> <p><i>注意：当 SPE 为 0，该位自动清 0。</i></p>
5	保留	写入无效
4	BUSY	<p>SPI 总线传输状态标志</p> <p>由硬件置 1 和清 0。</p>

0: SPI 总线空闲

1: SPI 总线正在传输数据

注意：当 *SPE* 为 0，该位自动清 0。

3 保留

写入无效

2 OVR

上溢标志

接收数据寄存器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 *OVR*CF 位置 1，此位清 0。

0: 未发生上溢

1: 发生上溢

注意：当 *SPE* 为 0，该位自动清 0。

1 保留

写入无效

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 *MMF*CF 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，*MSTR*和*SPE*同时由硬件清0。此标志不清除，无法置位*MSTR*和*SPE*。

## 22.5.4 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.													OVR	Res.	MMF
													w		w

位/位域 名称 描述

31:3 保留 写入无效

2	OVRCF	上溢标志清除 写 1 清除 SPIx_ISR 寄存器中 OVR 标志。
1	保留	写入无效
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。

### 22.5.5 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	写入无效
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数据寄存器；对数据寄存器执行读操作时，将返回接收数据寄存器中的已接收的数据。

## 23 调试接口（DBG）

### 23.1 简介

芯片采用 Cortex-M0+内核，具有硬件调试功能，支持复杂的调试操作。可以通过 SWD 调试接口为程序设置断点，使内核在获取特定指令（指令断点）或访问特定数据（数据断点）时停止。内核停止时，内核的内部状态和外设的状态都可以通过调试接口查询。当芯片连接到调试器后，可通过调试接口进行调试操作。

### 23.2 DBG 功能描述

Cortex-M0+内置的调试功能是 ARM CoreSight 设计套件的一部分，集成了 SW-DP（串行调试接口）、DWT（数据观察点触发）和 BPU（断点单元），具体如下：

- 调试支持串行线（SW）协议；
- BPU 提供 3 个断点寄存器；
- DWT 提供 1 个观察点寄存器组。

### 23.3 调试组件（DBG）

芯片调试组件为以下调试功能提供支持：

- 低功耗模式；
- 断点期间定时器和看门狗的时钟控制。

#### 23.3.1 对低功耗模式的调试支持

支持 Sleep 模式、Stop 模式和 Deepstop 模式的调试功能。

内核不允许在调试期间关闭 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。

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 模式下，HCLK 仍有效，对调试功能没有任何限制。
- 调试 Stop 或 Deepstop 模式时，DBG\_CR 寄存器的 DBG\_STOP 位置 1，系统时钟自动切换为 RCHDIV6 给 HCLK 提供时钟。



### 23.3.2 对定时器和看门狗的调试支持

当芯片处于调试模式内核停止时，可选择定时器或看门狗计数器是否继续计数：

- 可以配置 TIM3\_HOLD 控制 TIM3 继续计数或停止计数；
- 可以配置 TIM1\_HOLD 控制 TIM1 继续计数或停止计数；
- 可以配置 LPTIM1\_HOLD 控制 LPTIM1 继续计数或停止计数；
- 可以配置 IWDG\_HOLD 控制 IWDG 继续计数或停止计数。

## 23.4 DBG 寄存器

DBG 寄存器支持 32 位访问。

表 23-1 DBG 基地址

DBG 外设	基地址
DBG	0x4001 5800

### 23.4.1 控制寄存器（DBG\_CR）

偏移地址：0x00

复位值：0x0000 0000

**注意：** 仅电源复位时，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
31:1	保留	写入无效
0	DBG_STOP	Stop模式调试使能 0：禁止 1：使能

### 23.4.2 APB 冻结寄存器 1（DBG\_APB\_FZ1）

偏移地址：0x04

复位值：0x0000 0000

**注意：** 仅电源复位时，复位该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		LPTIM1_HOLD	Res.										IWDG_HOLD	Res.	
		rw											rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													TIM3_HOLD	Res.	



位/位域	名称	描述
31:30	保留	写入无效
29	LPTIM1_HOLD	内核停止时，LPTIM1计数控制 0: 正常计数 1: 停止计数
28:19	保留	写入无效
18	IWDG_HOLD	内核停止时，IWDG计数控制 0: 正常计数 1: 停止计数
15:2	保留	写入无效
1	TIM3_HOLD	内核停止时，TIM3计数控制 0: 正常计数 1: 停止计数
0	保留	写入无效

偏移地址: 0x08

复位值: 0x0000 0000

**注意:** 仅电源复位时, 复位该寄存器。

位/位域	名称	描述
31:1	保留	写入无效

0	TIM1_HOLD	内核停止时，TIM1计数控制
		0：正常计数
		1：停止计数

## 24 器件电子签名

器件电子签名可通过 SWD 或者 CPU 读取。它包含的芯片识别信息在出厂时编写, 用户程序或者外部设备可以读取电子签名, 用以自动匹配微处理器的特性。

### 24.1 产品唯一身份识别 (UID) 寄存器 (96 位)

96 位的唯一设备标识符提供了一个对于任何器件和任何上下文都唯一的参考号码。用户不能改变这些位, 其典型应用场景为:

- 用作序列号
- 对 Flash 编程前将 UID 和软件加密原语及协议结合使用, 作为安全密钥的一部分以提高 Flash 代码的安全性
- 激活安全启动过程等

存储地址: 0x1FFF 0340, 参见 [表: 产品信息](#)。

### 24.2 芯片产品型号参数寄存器

存储地址: 0x1FFF 03A4, 参见 [表: 产品信息](#)。

### 24.3 Flash 空间参数寄存器

存储地址: 0x1FFF 03A8, 参见 [表: 产品信息](#)。

### 24.4 SRAM 空间参数寄存器

存储地址: 0x1FFF 03AC, 参见 [表: 产品信息](#)。

## 25 版本历史

表 25-1 版本更改履历

日期	版本号	修改范围
2024-4-18	V1.0	初版
2024-10-9	V1.1	1、SPI: 更新 SPI 初始化流程, 明确时钟极性与主从模式的先后配置顺序, 完善 SPI_CR1 寄存器 MSTR 控制位描述 2、I2C: 更正图 20-3 和图 20-4 中 TXIS 的行为, 完善 I2Cx_ISR 寄存器中 TXE 控制位置 1 的行为描述 3、ADC: 补充 ADC_CFG3 寄存器相关描述 4、TIM1: 更正 TIM1_AF1 寄存器的复位值
2024-11-18	V1.2	1、ADC: 完善 ADC 使用流程描述 2、Flash: 完善读出保护描述
2025-1-13	V1.3	1、修正文档目录格式 2、PMU: 完善 PMU_FLASH_WAKEUP 寄存器各档位时间配置说明
2025-5-30	V1.4	1、引脚分配与功能描述: 完善表 3-1 中 COMP 附加功能描述 2、COMP: 更新结构框图, 明确 COMP1/2 可选择的正相输入 IO; 完善滤波时钟描述
2025-6-27	V1.5	引脚分配与功能描述: 增加 SOP16 type1 封装相关内容

## 联系方式

网址: [www.hed.com.cn](http://www.hed.com.cn)

地址: 北京市昌平区北七家未来科技城南区中国电子网络安全和信息化产业基地 C 栋

邮编: 102209

如果您在购买与使用过程中有任何意见或建议, 请随时与我们联系。