



用户手册

User Manual

BL32F0120

基于 Arm[®]Cortex[®]-M0 内核的 32 位微控制器

版本：V0.9

保留不通知的情况下，更改相关资料的权利

目录

目录	1
附图目录	14
表格目录	18
1 系统和存储器架构	21
1.1 系统架构简介	21
1.1.1 系统总线	22
1.1.2 DMA 总线	22
1.1.3 总线矩阵	22
1.2 存储器介绍	22
1.2.1 存储器映像和寄存器编址	22
1.2.2 内置的 SRAM	24
1.2.3 闪存存储器概述	24
1.2.4 启动配置	24
1.2.5 引导程序	25
2 PWR 电源控制	26
2.1 供电系统	26
2.1.1 模拟模块供电	27
2.1.2 数字模块供电	27
2.1.3 VDD 域	27
2.1.4 1.2V 域	27
2.2 电源管理器	27
2.2.1 POR 上电复位和 PDR 掉电复位	27
2.2.2 PVD 可编程电压监测器	28
2.3 功耗控制	29
2.3.1 功耗控制概述	29
2.3.2 运行模式降低系统时钟	30
2.3.3 外设时钟的控制	30
2.3.4 Sleep Mode 睡眠模式	31
2.3.5 Stop Mode 停机模式	31
2.3.6 DeepStop Mode 深度停机模式	33
2.4 电源控制寄存器	34
2.4.1 PWR_CR 电源控制寄存器	34
2.4.2 PWR_CSR 电源控制/状态寄存器	35
2.4.3 PWR_CFGR 电源配置寄存器	35
2.4.4 PWR_MEMCR 电源存储控制寄存器	36
3 RCC 时钟和复位	37
3.1 复位单元	37
3.1.1 简介	37
3.1.2 功能框图	37
3.1.3 主要特征	37

3.1.4	功能描述	37
3.2	时钟单元	39
3.2.1	简介	39
3.2.2	功能框图	40
3.2.3	主要特征	40
3.2.4	功能描述	40
3.3	寄存器	46
3.3.1	寄存器总览	46
3.3.2	RCC_CR 时钟控制寄存器	47
3.3.3	RCC_CFGR 时钟配置寄存器	48
3.3.4	RCC_CIR 时钟中断寄存器	49
3.3.5	RCC_APB2RSTR APB2 外设复位寄存器	51
3.3.6	RCC_APB1RSTR APB1 外设复位寄存器	52
3.3.7	RCC_AHBENR AHB 外设时钟使能寄存器	53
3.3.8	RCC_APB2ENR APB2 外设时钟使能寄存器	54
3.3.9	RCC_APB1ENR APB1 外设时钟使能寄存器	56
3.3.10	RCC_BDCR 备份域控制寄存器	57
3.3.11	RCC_CSR 控制状态寄存器	58
3.3.12	RCC_AHBRSTR AHB 外设复位寄存器	60
4	Embedded FLASH 嵌入式闪存	62
4.1	简介	62
4.2	闪存构成与说明	62
4.2.1	闪存构成	62
4.2.2	选项字节说明	63
4.2.3	Data 空间说明	64
4.3	闪存操作与流程	64
4.3.1	闪存读操作	64
4.3.2	闪存编程方式与操作流程	65
4.3.3	对闪存块操作限制的解除与使能	67
4.3.4	对选项字节区块操作限制的解除与使能	68
4.3.5	主闪存块擦除	70
4.3.6	主闪存块编程	72
4.3.7	选项字节区块擦除	73
4.3.8	选项字节区块编程	74
4.3.9	闪存保护	75
4.4	寄存器	77
4.4.1	寄存器总览	77
4.4.2	FLASH_ACR 闪存访问控制寄存器	77
4.4.3	FLASH_KEYR FPEC 键寄存器	78
4.4.4	FLASH_OPTKEYR 闪存 OPTKEY 寄存器	78
4.4.5	FLASH_SR 闪存状态寄存器	79
4.4.6	FLASH_CR 闪存控制寄存器	79
4.4.7	FLASH_AR 闪存地址寄存器	80
4.4.8	FLASH_OBR 选项字节寄存器	80

4.4.9	FLASH_WRP 写保护寄存器.....	81
5	SYSCFG 系统控制器.....	83
5.1	简介.....	83
5.2	寄存器.....	83
5.2.1	寄存器总览.....	83
5.2.2	SYSCFG_CFGR 配置寄存器.....	83
5.2.3	SYSCFG_EXTICR1 外部中断配置寄存器 1.....	84
5.2.4	SYSCFG_EXTICR2 外部中断配置寄存器 2.....	84
5.2.5	SYSCFG_EXTICR3 外部中断配置寄存器 3.....	85
5.2.6	SYSCFG_EXTICR4 外部中断配置寄存器 4.....	85
5.2.7	PAD 配置寄存器 (SYSCFG_PADHYS).....	86
6	DMA 直接存储器访问控制器.....	87
6.1	DMA 简介.....	87
6.2	DMA 功能框图.....	87
6.3	DMA 主要特征.....	87
6.4	中断.....	88
6.5	DMA.....	88
6.5.1	DMA 请求映像.....	88
6.6	功能描述.....	89
6.6.1	DMA 处理.....	89
6.6.2	仲裁器.....	89
6.6.3	DMA 通道.....	90
6.6.4	可编程的数据传输宽度, 对齐方式和数据大小端.....	91
6.6.5	错误管理.....	93
6.7	DMA 寄存器描述.....	93
6.7.1	DMA_ISR DMA 中断状态寄存器.....	93
6.7.2	DMA_IFCR DMA 中断标志清除寄存器.....	94
6.7.3	DMA_CCRx DMA 通道 x 配置寄存器 (x=1~3).....	94
6.7.4	DMA_CNDTRx DMA 通道 x 传输数量寄存器 (x=1~3).....	96
6.7.5	DMA_CPARx DMA 通道 x 外设地址寄存器 (x=1~3).....	96
6.7.6	DMA_CMARx DMA 通道 x 存储器地址寄存器 (x=1~3).....	97
7	GPIO 通用端口.....	98
7.1	简介.....	98
7.2	主要特征.....	98
7.3	功能描述.....	98
7.3.1	功能框图.....	99
7.3.2	GPIO 端口配置.....	99
7.3.3	复用功能.....	100
7.3.4	GPIO 锁定机制.....	100
7.3.5	输入配置.....	101
7.3.6	输出配置.....	102
7.3.7	复用功能配置.....	102
7.3.8	模拟输入配置.....	103
7.3.9	外部时钟复用 GPIO 端口.....	104

7.3.10	SWD 复用功能重映射	104
7.3.11	NRST 复用功能重映射	104
7.4	寄存器	104
7.4.1	寄存器总览	104
7.4.2	GPIOx_CRL 端口配置低寄存器	105
7.4.3	GPIOx_CRH 端口配置高寄存器	105
7.4.4	GPIOx_IDR 端口输入数据寄存器	106
7.4.5	GPIOx_ODR 端口输出数据寄存器	106
7.4.6	GPIOx_BSRR 端口设置/清除寄存器	107
7.4.7	GPIOx_BRR 端口位清除寄存器	107
7.4.8	GPIOx_LCKR 端口配置锁定寄存器	108
7.4.9	GPIOx_DCR 端口输出开漏控制寄存器	108
7.4.10	GPIOx_AFR1 端口复用功能低位寄存器	109
7.4.11	GPIOx_AFR2 端口复用功能高位寄存器	109
8	EXTI 中断和事件	111
8.1	简介	111
8.2	主要特征	111
8.3	功能描述	111
8.3.1	功能框图	111
8.3.2	中断和异常向量	112
8.3.3	唤醒事件管理	113
8.3.4	中断功能描述	113
8.3.5	硬件中断输出	113
8.3.6	硬件事件输出	114
8.3.7	软件中断与事件输出	114
8.3.8	外部中断映射	114
8.4	寄存器	115
8.4.1	寄存器总览	115
8.4.2	EXTI_IMR 中断屏蔽寄存器	115
8.4.3	EXTI_EMR 事件屏蔽寄存器	115
8.4.4	EXTI_RTSR 上升沿触发选择寄存器	116
8.4.5	EXTI_FTSR 下降沿触发选择寄存器	116
8.4.6	EXTI_SWIER 软件中断事件寄存器	117
8.4.7	EXTI_PR 软件中断事件挂起寄存器	117
9	HWDIV 硬件除法器	119
9.1	简介	119
9.2	主要特征	119
9.3	功能描述	119
9.4	寄存器	119
9.4.1	寄存器总览	119
9.4.2	HWDIV_DVDR 被除数寄存器	120
9.4.3	HWDIV_DVSR 除数寄存器	120
9.4.4	HWDIV_QUOTR 商寄存器	120
9.4.5	HWDIV_RMDR 余数寄存器	121

9.4.6	HWDIV_SR 状态寄存器.....	121
9.4.7	HWDIV_CR 控制寄存器	121
10	BKP 备份寄存器.....	123
10.1	简介	123
10.2	主要特征	123
10.3	功能描述	123
10.3.1	时钟校准	123
10.3.2	侵入检测	124
10.4	寄存器	124
10.4.1	寄存器总览	124
10.4.2	备份数据寄存器 n(BKP_DRn)(n = 1 .. 10).....	124
10.4.3	时钟校准寄存器 (BKP_RTCCR).....	124
10.4.4	备份控制寄存器 (BKP_CR).....	125
10.4.5	备份控制状态寄存器 (BKP_CSR).....	125
11	TIM1 高级定时器.....	127
11.1	简介	127
11.2	功能框图	127
11.3	主要特征	128
11.4	功能描述	128
11.4.1	时钟	128
11.4.2	重复计数器	133
11.4.3	比较输出	135
11.4.4	从模式	149
11.4.5	定时器同步	151
11.4.6	调试模式	153
11.4.7	中断	154
11.4.8	DMA	154
11.5	寄存器	155
11.5.1	TIM1_CR1 控制寄存器 1.....	155
11.5.2	TIM1_CR2 控制寄存器 2.....	156
11.5.3	TIM1_SMCR 从模式控制寄存器	158
11.5.4	TIM1_DIER DMA/中断使能寄存器	159
11.5.5	TIM1_SR 状态寄存器	161
11.5.6	TIM1_EGR 事件产生寄存器	162
11.5.7	TIM1_CCMR1 比较模式寄存器 1.....	163
11.5.8	TIM1_CCMR2 比较模式寄存器 2.....	164
11.5.9	TIM1_CCER 比较使能寄存器	166
11.5.10	TIM1_CNT 计数器	167
11.5.11	TIM1_PSC 预分频器	167
11.5.12	TIM1_ARR 自动预装载寄存器	167
11.5.13	TIM1_RCR 重复计数寄存器	168
11.5.14	TIM1_CCR1 比较寄存器 1	168
11.5.15	TIM1_CCR2 比较寄存器 2	168
11.5.16	TIM1_CCR3 比较寄存器 3	169

11.5.17	TIM1_CCR4 比较寄存器 4	169
11.5.18	TIM1_BDTR 刹车和死区寄存器.....	169
11.5.19	TIM1_DCR DMA 控制寄存器.....	171
11.5.20	TIM1_DMAR 连续模式 DMA 地址寄存器	171
11.5.21	TIM1_CCMR3 比较模式寄存器 3.....	172
11.5.22	TIM1_CCR5 比较寄存器 5	172
11.5.23	TIM1_PDER PWM 移相/DMA repeat 更新请求使能寄存器	173
11.5.24	TIM1_CCRxFALL PWM 移相递减计数比较寄存器	173
11.5.25	TIM1_BKINF 刹车输入滤波寄存器	174
12	TIM3 通用定时器.....	176
12.1	简介	176
12.2	功能框图	176
12.3	主要特征	177
12.4	功能描述	177
12.4.1	时钟	177
12.4.2	输入捕获	182
12.4.3	比较输出	185
12.4.4	从模式	193
12.4.5	定时器同步	197
12.4.6	定时器异或	197
12.4.7	调试模式	198
12.4.8	中断	198
12.4.9	DMA	198
12.5	寄存器	199
12.5.1	TIMx_CR1 控制寄存器 1.....	200
12.5.2	TIMx_CR2 控制寄存器 2.....	201
12.5.3	TIMx_SMCR 从模式控制寄存器	201
12.5.4	TIMx_DIER DMA/中断使能寄存器	203
12.5.5	TIMx_SR 状态寄存器	204
12.5.6	TIMx_EGR 事件产生寄存器	206
12.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1.....	206
12.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2.....	209
12.5.9	TIMx_CCER 捕获/比较使能寄存器.....	212
12.5.10	TIMx_CNT 计数器	213
12.5.11	TIMx_PSC 预分频器	213
12.5.12	TIMx_ARR 自动预装载寄存器	214
12.5.13	TIMx_CCR1 捕获/比较寄存器 1	214
12.5.14	TIMx_CCR2 捕获/比较寄存器 2	215
12.5.15	TIMx_CCR3 捕获/比较寄存器 3	215
12.5.16	TIMx_CCR4 捕获/比较寄存器 4.....	215
12.5.17	TIMx_DCR DMA 控制寄存器.....	215
12.5.18	TIMx_DMAR 连续模式 DMA 地址寄存器	216
12.5.19	TIMx_OR 输入选项寄存器.....	216
13	TIM13/14 基本定时器.....	218

13.1	简介	218
13.2	功能框图	218
13.3	主要特征	218
13.4	功能描述	219
13.4.1	时钟	219
13.4.2	输入捕获	220
13.4.3	比较输出	221
13.4.4	调试模式	224
13.4.5	中断	225
13.4.6	DMA	225
13.5	寄存器	225
13.5.1	TIMx_CR1 控制寄存器 1	225
13.5.2	TIMx_DIER DMA/中断使能寄存器	226
13.5.3	TIMx_SR 状态寄存器	227
13.5.4	TIMx_EGR 事件产生寄存器	227
13.5.5	TIMx_CCMR1 捕获/比较模式寄存器 1	228
13.5.6	TIMx_CCER 捕获/比较使能寄存器	230
13.5.7	TIMx_CNT 计数器	231
13.5.8	TIMx_PSC 预分频器	231
13.5.9	TIMx_ARR 自动预装载寄存器	232
13.5.10	TIMx_CCR1 捕获/比较寄存器 1	232
13.5.11	TIMx_BDTR 刹车和死区寄存器	232
14	TIM16/17 基本定时器	234
14.1	简介	234
14.2	功能框图	234
14.3	主要特征	234
14.4	功能描述	235
14.4.1	时钟	235
14.4.2	输入捕获	236
14.4.3	比较输出	238
14.4.4	定时器同步	245
14.4.5	调试模式	245
14.4.6	中断	245
14.4.7	DMA	245
14.5	寄存器	245
14.5.1	TIMx_CR1 控制寄存器 1	246
14.5.2	TIMx_CR2 控制寄存器 2	247
14.5.3	TIMx_DIER DMA/中断使能寄存器	247
14.5.4	TIMx_SR 状态寄存器	248
14.5.5	TIMx_EGR 事件产生寄存器	249
14.5.6	TIMx_CCMR1 捕获/比较模式寄存器 1	250
14.5.7	TIMx_CCER 捕获/比较使能寄存器	252
14.5.8	TIMx_CNT 计数器	253
14.5.9	TIMx_PSC 预分频器	254

14.5.10	TIMx_ARR 自动预装载寄存器	254
14.5.11	TIMx_CCR1 捕获/比较寄存器 1	254
14.5.12	TIMx_BDTR 刹车和死区寄存器.....	255
14.5.13	TIMx_BKINF 刹车滤波寄存器	256
15	IWDG 独立看门狗	259
15.1	IWDG 简介.....	259
15.2	IWDG 主要性能.....	259
15.3	IWDG 功能描述.....	259
15.3.1	硬件看门狗	260
15.3.2	寄存器访问保护	260
15.3.3	调试模式	260
15.4	寄存器	260
15.4.1	寄存器总览	260
15.4.2	IWDG_KR 键寄存器	261
15.4.3	IWDG_PR 预分频寄存器.....	261
15.4.4	IWDG_RLR 重装载寄存器.....	262
15.4.5	IWDG_SR 状态寄存器.....	262
15.4.6	IWDG_CR 控制寄存器	263
15.4.7	IWDG_IGEN 中断生成寄存器	263
15.4.8	IWDG_CNT 计数寄存器.....	264
16	RTC 实时时钟器.....	265
16.1	RTC 简介	265
16.2	主要特征	265
16.3	功能描述	266
16.3.1	功能框图	266
16.3.2	功能概述	266
16.3.3	模块复位	267
16.3.4	寄存器读取	267
16.3.5	寄存器配置	267
16.3.6	标志位产生	268
16.3.7	RTC 闹钟描述.....	269
16.3.8	RTC 外部中断事件输出.....	269
16.4	寄存器描述	269
16.4.1	寄存器总览	269
16.4.2	控制寄存器高位 (RTC_CRH).....	269
16.4.3	控制寄存器低位(RTC_CRL).....	270
16.4.4	预分频装载寄存器高位(RTC_PRLH)	271
16.4.5	预分频装载寄存器低位(RTC_PRLL).....	271
16.4.6	预分频器分频因子寄存器高位(RTC_DIVH)	272
16.4.7	预分频器分频因子寄存器低位(RTC_DIVL).....	272
16.4.8	计数器寄存器高位(RTC_CNTH)	272
16.4.9	计数器寄存器低位(RTC_CNTL).....	273
16.4.10	闹钟寄存器高位(RTC_ALRH)	273
16.4.11	闹钟寄存器低位(RTC_ALRL).....	273

16.4.12	毫秒闹钟寄存器高位 (RTC_MSRH)	274
16.4.13	毫秒闹钟寄存器低位 (RTC_MSRL)	274
16.4.14	RTC LSE 配置寄存器 (RTC_LSE_CFG)	275
17	USART 通用同步异步收发器	276
17.1	介绍	276
17.2	USART 特性	276
17.3	USART 功能描述	277
17.3.1	功能框图	277
17.3.2	信号描述	277
17.3.3	功能概述	278
17.3.4	特性描述	278
17.3.5	波特率发生器	279
17.3.6	采样	280
17.3.7	奇偶校验控制	280
17.3.8	发送器	280
17.3.9	接收器	282
17.3.10	同步模式	283
17.3.11	单线半双工通信	284
17.3.12	硬件流控	284
17.3.13	中断	284
17.3.14	DMA	285
17.4	寄存器	285
17.4.1	寄存器总览	285
17.4.2	USART_SR 状态寄存器	285
17.4.3	USART_DR 数据寄存器	287
17.4.4	USART_BRR 波特率寄存器	287
17.4.5	USART_CR1 控制寄存器 1	288
17.4.6	USART_CR2 控制寄存器 2	289
17.4.7	USART_CR3 控制寄存器 3	290
18	SPI 串行外设接口	292
18.1	简介	292
18.2	功能框图	292
18.3	功能描述	293
18.3.1	概述	293
18.3.2	SPI 主要特征	295
18.3.3	SPI 从模式	296
18.3.4	SPI 主模式	297
18.3.5	波特率设置	298
18.3.6	中断	298
18.3.7	DMA 传输	299
18.4	寄存器	299
18.4.1	SPI_TXREG 发送数据寄存器	300
18.4.2	SPI_RXREG 接收数据寄存器	300
18.4.3	SPI_CSTAT 当前状态寄存器	300

18.4.4	SPI_INTSTAT 中断状态寄存器	301
18.4.5	SPI_INTEN 中断使能寄存器	302
18.4.6	SPI_INTCLR 中断清除寄存器	303
18.4.7	SPI_GCTL 全局控制寄存器	304
18.4.8	SPI_CCTL 通用控制寄存器	306
18.4.9	SPI_SPBRG 波特率发生器	307
18.4.10	SPI_RXDNR 接收数据个数寄存器	307
18.4.11	SPI_NSSR 从机片选寄存器	307
18.4.12	SPI_EXTCTL 数据控制寄存器	308
19	I2C 内部集成电路接口	309
19.1	简介	309
19.2	主要特征	309
19.3	功能描述	310
19.3.1	功能框图	310
19.3.2	信号描述	311
19.3.3	I2C 协议	311
19.3.4	工作模式	319
19.3.5	中断	324
19.3.6	DMA	325
19.4	寄存器	325
19.4.1	寄存器总览	325
19.4.2	I2C_CR 控制寄存器	326
19.4.3	I2C_TAR 目标地址寄存器	327
19.4.4	I2C_SAR 从地址寄存器	328
19.4.5	I2C_DR 数据命令寄存器	328
19.4.6	I2C_SSHR 标准模式时钟高电平计数寄存器	329
19.4.7	I2C_SSLR 标准模式时钟低电平计数寄存器	329
19.4.8	I2C_FSHR 快速模式时钟高电平计数寄存器	330
19.4.9	I2C_FSLR 快速模式时钟低电平计数寄存器	330
19.4.10	I2C_ISR 中断状态寄存器	331
19.4.11	I2C_IMR 中断屏蔽寄存器	331
19.4.12	I2C_RAWISR RAW 中断寄存器	331
19.4.13	I2C_RXTLR 接收阈值	333
19.4.14	I2C_TXTLR 发送阈值	333
19.4.15	I2C_ICR 组合和独立中断清除寄存器	333
19.4.16	I2C_RX_UNDER 清除 RX_UNDER 中断寄存器	334
19.4.17	I2C_RX_OVER 清除 RX_OVER 中断寄存器	334
19.4.18	I2C_TX_OVER 清除 TX_OVER 中断寄存器	335
19.4.19	I2C_RD_REQ 清除 RD_REQ 中断寄存器	335
19.4.20	I2C_TX_ABRT 清除 TX_ABRT 中断寄存器	335
19.4.21	I2C_RX_DONE 清除 RX_DONE 中断寄存器	336
19.4.22	I2C_ACTIV 清除 ACTIVITY 中断寄存器	336
19.4.23	I2C_STOP 清除 STOP_DET 中断寄存器	337
19.4.24	I2C_START 清除 START_DET 中断寄存器	337

19.4.25	I2C_GC 清除 GEN_CALL 中断寄存器.....	337
19.4.26	I2C_ENR 使能寄存器	338
19.4.27	I2C_SR 状态寄存器.....	338
19.4.28	I2C_TXFLR 发送缓冲级别寄存器.....	339
19.4.29	I2C_RXFLR 接收缓冲级别寄存器.....	339
19.4.30	I2C_HOLD SDA 保持时间寄存器.....	340
19.4.31	I2C_DMA DMA 控制寄存器	340
19.4.32	I2C_SETUP SDA 建立时间寄存器.....	341
19.4.33	I2C_GCR 广播呼叫 ACK 寄存器.....	341
19.4.34	I2C_SLVMASK 从地址掩码寄存器.....	341
19.4.35	I2C_SLVRCVADDR 从器件接收地址寄存器	342
20	CSM 通用串行模块.....	343
20.1	简介	343
20.2	功能框图	343
20.3	主要特征	344
20.4	功能描述	344
20.4.1	外部端口	344
20.4.2	数据接收	345
20.4.3	数据发送	345
20.4.4	状态标志	345
20.4.5	波特率设置	345
20.5	CSM 模拟输出	346
20.6	DMA.....	346
20.7	寄存器	346
20.7.1	寄存器总览	346
20.7.2	CSM_TDR1 发送数据寄存器 1	347
20.7.3	CSM_RDR1 接收数据寄存器 1	347
20.7.4	CSM_ISR 中断状态寄存器	347
20.7.5	CSM_IER 中断使能寄存器.....	348
20.7.6	CSM_CR1 控制寄存器 1.....	349
20.7.7	CSM_CFG 配置寄存器	349
20.7.8	CSM_SPBRG 波特率寄存器.....	350
20.7.9	CSM_BCNT 数据位长度寄存器.....	350
21	ADC 模拟/数字转换	352
21.1	简介	352
21.2	主要特征	352
21.3	DMA.....	352
21.4	系统框图	353
21.5	ADC 功能描述.....	353
21.5.1	ADC 开关控制.....	354
21.5.2	通道选择	354
21.6	任意通道工作模式	355
21.6.1	单次转换模式	355
21.6.2	单周期扫描模式	355

21.6.3	连续扫描模式	356
21.7	数据对齐	357
21.7.1	可编程分辨率	358
21.7.2	可编程采样时间	358
21.8	外部触发转换	358
21.9	内部基准参考电压	358
21.10	窗口比较器模式下 AD 转换结果监控	358
21.11	寄存器	359
21.11.1	寄存器总览	359
21.11.2	ADC_ADDDATA 数据寄存器	359
21.11.3	ADC_ADCFG 配置寄存器	360
21.11.4	ADC_ADCR 控制寄存器	361
21.11.5	ADC_ADCMPR 窗口比较寄存器	363
21.11.6	ADC_ADSTA 状态寄存器	363
21.11.7	ADC_ADDRn 通道数据寄存器(n=0~10, 12, 15)	364
21.11.8	ADC_ADSTA_EXT 扩展状态寄存器	364
21.11.9	ADC_CHANY0 任意通道的通道选择寄存器 0	365
21.11.10	ADC_CHANY1 任意通道的通道选择寄存器 1	366
21.11.11	ADC_ANY_CFG 任意通道配置寄存器	366
21.11.12	ADC_ANY_CR 任意通道控制寄存器	367
22	COMP 比较器	368
22.1	简介	368
22.2	功能框图	368
22.3	主要特征	368
22.4	功能描述	369
22.4.1	比较器时钟和复位	369
22.4.2	比较器开关控制	369
22.4.3	比较器输入和输出	369
22.4.4	比较器通道选择	369
22.4.5	中断和唤醒	370
22.4.6	功耗模式	370
22.4.7	比较器锁定机制	370
22.4.8	迟滞电压	371
22.5	寄存器	371
22.5.1	比较器控制状态寄存器 (COMPx_CSR) (x=1)	371
22.5.2	比较器外部参考电压寄存器 (COMP_CRV)	373
22.5.3	比较器轮询寄存器 (COMPx_POLL) (x=1)	374
23	CRC 循环冗余校验计算单元	376
23.1	简介	376
23.2	主要特征	376
23.3	功能描述	376
23.3.1	功能框图	376
23.3.2	功能概述	376
23.3.3	使用方法	377

23.4	寄存器	377
23.4.1	寄存器总览	377
23.4.2	CRC_DR CRC 数据寄存器	377
23.4.3	CRC_IDR CRC 独立数据寄存器	378
23.4.4	CRC_CR CRC 控制寄存器	378
24	DBG 调试支持	379
24.1	介绍	379
24.2	功能描述	379
24.2.1	功能框图	379
24.2.2	SWD 内部上拉与下拉	380
24.2.3	SWJ 调试端口	380
24.3	ID 代码和锁定机制	380
24.3.1	微控制器设备 ID 编码	380
24.3.2	Cortex JEDEC-106 ID 编码	380
24.4	SW 调试端口	381
24.4.1	SW 协议介绍	381
24.4.2	SW 协议序列	381
24.4.3	SW-DP 状态机 (Reset, Idle states, ID code)	382
24.4.4	DP 和 AP 读 / 写访问	382
24.4.5	SW-DP 寄存器	382
24.4.6	SW-AP 寄存器	383
24.5	MCU 调试模块 (DBGMCU)	383
24.5.1	低功耗模式的调试支持	383
24.5.2	支持定时器、看门狗	383
24.6	寄存器	384
24.6.1	寄存器总览	384
24.6.2	DBG_IDCODE ID 编码寄存器	384
24.6.3	DBG_CR 控制寄存器	384
25	Device Electronic Signature 器件电子签名	387
25.1	简介	387
25.2	寄存器描述	387
25.2.1	UID1 唯一标识码	387
25.2.2	UID2 唯一标识码	388
25.2.3	UID3 唯一标识码	388

附图目录

图 1-1 系统架构框图	22
图 2-1 电源控制功能框图	26
图 2-2 上电复位和掉电复位波形图	28
图 2-3 PVD 阈值波形图	29
图 3-1 复位功能框图	37
图 3-2 时钟树	40
图 3-3 外部高速输入时钟	41
图 3-4 低速晶振/陶瓷谐振器	42
图 3-5 外部低速输入时钟	43
图 4-1 ISP 方式编程流程图	66
图 4-2 IAP 编程流程	67
图 6-1 DMA 功能框图	87
图 7-1 模拟输入	104
图 8-1 EXTI 结构框图	112
图 11-1 TIM1 结构图	127
图 11-2 时钟选择	129
图 11-3 外部时钟模式 1 下的控制电路	130
图 11-4 外部时钟模式 2 下的控制电路	130
图 11-5 自动预装载	131
图 11-6 递增计数模式 (UDIS=0)	131
图 11-7 递增计数模式 (UDIS=1 禁止产生更新事件)	132
图 11-8 递减计数模式 (UDIS=0)	132
图 11-9 递减计数模式 (UDIS=1 禁止产生更新事件)	132
图 11-10 中央计数模式 (UDIS=0)	133
图 11-11 中央计数模式 (UDIS=1 禁止产生更新事件)	133
图 11-12 中央对齐模式重复计数时序图	134
图 11-13 边沿对齐模式递增计数时序图	134
图 11-14 边沿对齐模式递减计数时序图	135
图 11-15 比较输出部分结构图	135
图 11-16 比较输出模式, OC1 信号在匹配时翻转	137
图 11-17 边沿对齐递增计数时 PWM 模式 1 的波形	138
图 11-18 边沿对齐递减计数时 PWM 模式 1 的波形	139
图 11-19 中央对齐 PWM 模式 1 的波形	140
图 11-20 移相功能示意图	141
图 11-21 使用 COM 事件产生六步 PWM (OSSR = 1)	141
图 11-22 边沿对齐递增计数模式下 CCx_SETTRGO 输出示例	142
图 11-23 中央对齐模式下 CCx_SETTRGO 输出示例	143
图 11-24 死区插入	144
图 11-25 响应刹车的输出 (OISx=0, OISxN=0)	146
图 11-26 响应刹车的输出 (OISx=0, OISxN=1)	146

图 11-27 响应刹车的输出 (OISx=1, OISxN=0)	147
图 11-28 响应刹车的输出 (OISx=1, OISxN=1)	147
图 11-29 外部事件清除 OCxREF	147
图 11-30 单脉冲模式	148
图 11-31 复位模式的控制时序图	149
图 11-32 门控模式下的控制时序图	150
图 11-33 触发器模式下的控制时序图	150
图 11-34 外部时钟模式 2+从模式 (触发模式) 控制时序图	151
图 11-35 定时器间互联	151
图 11-36 使用主定时器作为从定时器的预分频器	152
图 11-37 使用主定时器使能从定时器	152
图 11-38 使用主定时器的更新事件启动从定时器	153
图 11-39 主定时器的 TI1 同步启动主定时器和从定时器	153
图 12-1 TIMx 结构图	176
图 12-2 时钟选择	178
图 12-3 外部时钟模式 1 下的控制电路	179
图 12-4 外部时钟模式 2 下的控制电路	179
图 12-5 自动预装载	180
图 12-6 递增计数模式 (UDIS=0)	180
图 12-7 递增计数模式 (UDIS=1 禁止产生更新事件)	181
图 12-8 递减计数模式 (UDIS=0)	181
图 12-9 递减计数模式 (UDIS=1 禁止产生更新事件)	181
图 12-10 中央计数模式 (UDIS=0)	182
图 12-11 中央计数模式 (UDIS=1 禁止产生更新事件)	182
图 12-12 TIMx 输入捕获结构图	183
图 12-13 PWM 输入模式时序	184
图 12-14 比较输出部分结构图	185
图 12-15 比较输出模式, OC1 信号在匹配时翻转	186
图 12-16 边沿对齐递增计数时 PWM 模式 1 的波形	188
图 12-17 边沿对齐递减计数时 PWM 模式 1 的波形	189
图 12-18 中央对齐 PWM 模式 1 的波形	190
图 12-19 边沿对齐递增计数模式下 CCx_SETTRGO 输出示例	191
图 12-20 中央对齐模式下 CCx_SETTRGO 输出示例	191
图 12-21 外部事件清除 OCxREF	192
图 12-22 单脉冲模式	192
图 12-23 编码器模式下的计数器时序图	194
图 12-24 IC1FP1 反相编码器接口模式时序图	195
图 12-25 复位模式的控制时序图	195
图 12-26 门控模式下的控制时序图	196
图 12-27 触发器模式下的控制时序图	196
图 12-28 外部时钟模式 2+从模式 (触发模式) 控制时序图	197
图 12-29 (TI1 异或输入) 输入捕获波形图	198
图 13-1 TIMx 结构图	218
图 13-2 自动预装载	219

图 13-3 递增计数模式 (UDIS=0)	220
图 13-4 递增计数模式 (UDIS=1 禁止产生更新事件)	220
图 13-5 TIMx 输入捕获结构图	220
图 13-6 比较输出部分结构图	222
图 13-7 比较输出模式, OC1 信号在匹配时翻转	223
图 13-8 边沿对齐递增计数时 PWM 模式 1 的波形	224
图 13-9 单脉冲模式	224
图 14-1 TIMx 结构图	234
图 14-2 自动预装载	235
图 14-3 递增计数模式 (UDIS=0)	236
图 14-4 递增计数模式 (UDIS=1 禁止产生更新事件)	236
图 14-5 TIMx 输入捕获结构图	236
图 14-6 比较输出部分结构图	238
图 14-7 比较输出模式, OC1 信号在匹配时翻转	239
图 14-8 边沿对齐递增计数时 PWM 模式 1 的波形	240
图 14-9 死区插入	241
图 14-10 响应刹车的输出 (OISx=0, OISxN=0)	243
图 14-11 响应刹车的输出 (OISx=0, OISxN=1)	244
图 14-12 响应刹车的输出 (OISx=1, OISxN=0)	244
图 14-13 响应刹车的输出 (OISx=1, OISxN=1)	244
图 14-14 单脉冲模式	244
图 15-1 独立看门狗框图	259
图 16-1 RTC 功能框图	266
图 16-2 RTC 秒和闹钟波形图示例, RTC_PRL = 0004, RTC_ALR = 002	268
图 16-3 RTC 溢出波形图示例, RTC_PRL = 0004	268
图 17-1 UART 功能框图	277
图 17-2 UART 数据帧类型示意图	279
图 18-1 SPI 功能框图	292
图 18-2 单主和单从应用	293
图 18-3 数据时钟时序图	294
图 19-1 I2C 功能框图	310
图 19-2 起始和停止条件	311
图 19-3 7 位地址格式	312
图 19-4 10 位地址格式	312
图 19-5 主发送协议	313
图 19-6 主接收协议	313
图 19-7 带 RESTART (SR) 信号的主发送和接收协议	314
图 19-8 起始字节传输	314
图 19-9 I2C_DR 寄存器	315
图 19-10 主发送-TX FIFO 为空或产生 STOP	315
图 19-11 主接收-TX FIFO 为空或产生 STOP	316
图 19-12 主发送-产生 RESTART	316
图 19-13 主接收-产生 RESTART	316
图 19-14 两个主器件仲裁	317

图 19-15 时钟同步（示意图）	317
图 19-16 时钟同步（时序图）	318
图 19-17 SCL 生成时序.....	318
图 19-18 流程图（I2C 接口作为从器件）	321
图 19-19 流程图（I2C 接口作为主器件）	323
图 19-20 I2C 中断机制	325
图 20-1 CSM 功能框图.....	343
图 20-2 PD 应用示意图	344
图 21-1 ADC 系统框图	353
图 21-2 ADC 框图	354
图 21-3 单次转换模式时序图	355
图 21-4 单周期扫描下通道转换时序图	356
图 21-5 连续扫描模式通道转换时序图	357
图 21-6 连续扫描模式动态更新配置时序图	357
图 21-7 数据对其方式	357
图 22-1 比较器框图	368
图 22-2 比较器的迟滞	371
图 23-1 CRC 功能框图	376
图 24-1 调试功能框图	379

表格目录

表 1-1 存储器映像	22
表 1-2 启动模式	24
表 2-1 低功耗模式列表	30
表 2-2 SLEEPNOW 模式	31
表 2-3 SLEEPONEXIT 模式	31
表 2-4 停机模式	32
表 2-5 深度停机模式	33
表 2-6 电源控制寄存器概览	34
表 3-1 RCC 全局中断表	43
表 3-2 MCO 与时钟源对应关系	45
表 3-3 RCC 寄存器概览	46
表 4-1 Flash 存储空间	62
表 4-2 信息块	63
表 4-3 选项字节组织结构	63
表 4-4 USER 的位含义	63
表 4-5 Latency 设置关系	65
表 4-6 编程方式	65
表 4-7 保护设置的状态变化	68
表 4-8 Flash 读保护状态	76
表 4-9 Flash 解除读保护状态	76
表 4-10 写保护区域	76
表 4-11 FLASH 寄存器概览	77
表 5-1 SYSCFG 寄存器概览	83
表 6-1 DMA 中断请求	88
表 6-2 DMA 各个通道的 DMA 请求一览	88
表 6-3 可配置的数据传输宽度和大小端操作（当 PINC = MINC = 1），传输数目为 4	91
表 6-4 DMA 寄存器概览	93
表 8-1 异常向量表	112
表 8-2 中断向量表	112
表 8-3 EXTI 触发源	114
表 8-4 EXTI 寄存器总览	115
表 9-1 HWDIV 寄存器概览	119
表 10-1 BKP 寄存器概览	124
表 11-1 死区时间计算	143
表 11-2 当 MOE=1, OSSR=0/1, OSSR=0 时:	144
表 11-3 当 MOE=1, OSSR=0/1, OSSR=1 时:	145
表 11-4 当 MOE=0, OSSR=0, OSSR=0/1 时:	145
表 11-5 当 MOE=0, OSSR=1, OSSR=0/1 时:	145
表 11-6 中断事件一览表	154

表 11-7 TIM1 寄存器概览	155
表 11-8 TIMx 内部触发连接	159
表 12-1 数字滤波器宽度与 ICxF 的对应关系表	183
表 12-2 计数方向与编码器信号的关系	194
表 12-3 中断事件一览表	198
表 12-4 TIMx 寄存器概览	199
表 12-5 TIMx 内部触发连接	203
表 12-6 ICx 极性/电平选择表	213
表 13-1 数字滤波器宽度与 IC1F 的对应关系表	221
表 13-2 中断事件一览表	225
表 13-3 TIMx 寄存器概览	225
表 13-4 IC1 极性/电平选择表	231
表 14-1 数字滤波器宽度与 IC1F 的对应关系表	237
表 14-2 死区时间计算	240
表 14-3 当 MOE=1, OSSI=0/1, OSSR=0 时:	242
表 14-4 当 MOE=1, OSSI=0/1, OSSR=1 时:	242
表 14-5 当 MOE=0, OSSI=0, OSSR=0/1 时:	242
表 14-6 当 MOE=0, OSSI=1, OSSR=0/1 时:	242
表 14-7 中断事件一览表	245
表 14-8 TIMx 寄存器概览	245
表 14-9 IC1 极性/电平选择表	253
表 15-1 IWDG 超时时间 (以 LSI 时钟频率 40kHz 为例)	260
表 15-2 IWDG 寄存器概览	260
表 16-1 RTC 寄存器概览	269
表 17-1 UART 中断请求	284
表 17-2 USART 寄存器概览	285
表 18-1 波特率公式	298
表 18-2 SPI 状态	298
表 18-3 SPI 寄存器概览	299
表 19-1 引脚定义	311
表 19-2 I2C 首字节	312
表 19-3 中断位的置位和清除	324
表 19-4 I2C 寄存器概览	325
表 19-5 DISSLAVE 和 MASTER 配置	327
表 20-1 CSM 寄存器概览	346
表 21-1 ADC 寄存器概览	359
表 22-1 COMP 寄存器概览	371
表 23-1 CRC 寄存器概览	377
表 24-1 SWD 调试端口管脚	380
表 24-2 ID 编码	380
表 24-3 8bit 请求包	381
表 24-4 3bit 应答包	381
表 24-5 33bit 数据包	382
表 24-6 SW-DP 寄存器	383

表 24-7 DBG 寄存器概览.....	384
表 25-1 存储器容量寄存器概览	387

1 系统和存储器架构

1.1 系统架构简介

BL32F0120 是基于 ARM Cortex-M0 处理器开发的 32 位微控制器产品，它同时具备了高性能和低功耗的特点。BL32F0120 采用矩阵总线结构，该矩阵包括两个 AHB 主机：CPU 和 DMA，三个从机分别是 SRAM、闪存存储、AHB 总线（含 AHB 到 APB 的总线桥）以及连接在 APB 总线的各种设备。

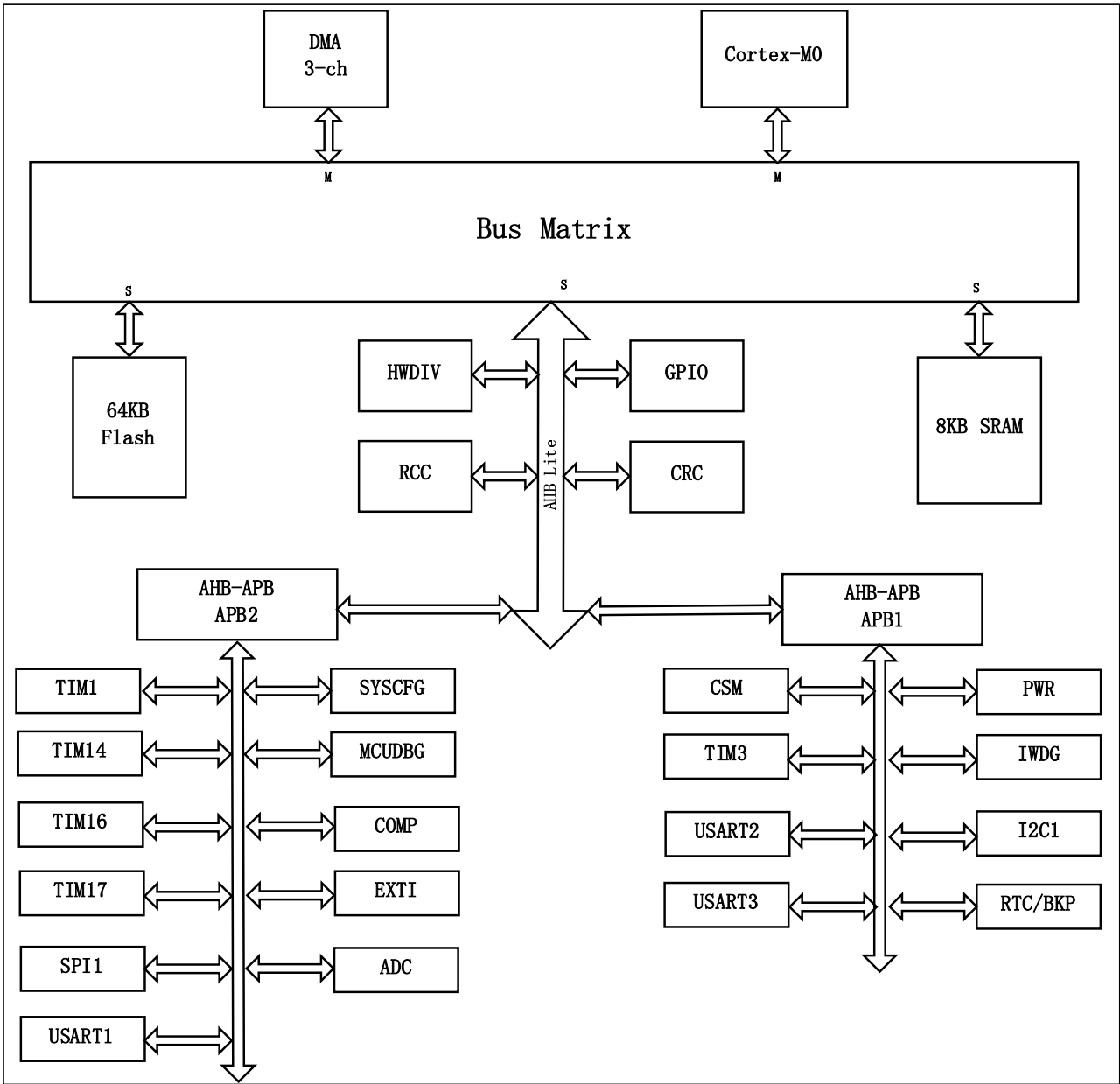


图 1-1 系统架构框图

1.1.1 系统总线

系统总线连接 CPU 内核和总线矩阵，从而达到数据传输的作用。CPU 和 DMA 作为主机驱动总线，总线矩阵会协调 CPU 内核和 DMA 之间的访问。

1.1.2 DMA 总线

DMA 总线作用是连接 DMA 和总线矩阵，从而达到数据传输的作用，总线矩阵协调着主机 DMA 到 从机 SRAM，闪存和连接在 APB 线上的各种外设的访问控制。

1.1.3 总线矩阵

总线矩阵包括一个 AHB 互联矩阵，一个 AHB 总线和两个桥接的 APB 总线。当 CPU 总线和 DMA 总线同时请求时，具备仲裁的功能。AHB 总线的外设（RCC，GPIO 和 DIV）通过 AHB 互联矩阵与系统总线连接。在 APB 和 AHB 总线之间连接通过 AHB2APB 桥进行数据交换。当 APB 寄存器进行 8 位 16 位访问，APB 会自动拓宽成 32 位，同样的，AHB2APB 桥也具备自动拓宽功能。

1.2 存储器介绍

程序存储器，数据存储器，寄存器和 I/O 接口都位于不同地址范围的存储器地址空间（线性 4GB 的地址空间）。4GB 的地址空间被分为 8 块，每块为 512MB。

1.2.1 存储器映像和寄存器编址

存储器映像请参考各外设对应章节的存储器映像表格。

表 1-1 存储器映像

总线	编址范围	大小	外设
Flash	0x0000 0000 - 0x0000 FFFF	64 KB	可映射为主闪存存储器、数据存储器、系统存储器或 SRAM，有赖于 BOOT 的配置
	0x0000 8000 - 0x07FF FFFF	~127 MB	Reserved
	0x0800 0000 - 0x0800 FFFF	64 KB	主存储区
	0x0801 0000 - 0x1FDF FFFF	~381 MB	Reserved
	0x1FE0 0000 - 0x1FE0 0BFF	3KB	数据存储器
	0x1FE0 0C00 - 0x1FFF F3FF	~2MB	Reserved
	0x1FFF F400 - 0x1FFF F7FF	1 KB	系统存储区
	0x1FFF F800 - 0x1FFF F9FF	0.5KB	选项字节
	0x1FFF FA00 - 0x1FFF FFFF	1.5KB	Reserved

总线	编址范围	大小	外设
SRAM	0x2000 0000 – 0x2000 1FFF	8 KB	SRAM
	0x2000 2000 – 0x2FFF FFFF	~255 MB	Reserved
APB1	0x4000 0000 – 0x4000 03FF	1KB	Reserved
	0x4000 0400 – 0x4000 07FF	1KB	TIM3
	0x4000 0800 – 0x4000 27FF	8KB	Reserved
	0x4000 2800 – 0x4000 2BFF	1KB	RTC/BKP
	0x4000 2C00 – 0x4000 2FFF	1KB	Reserved
	0x4000 3000 – 0x4000 33FF	1KB	IWDG
	0x4000 3400 – 0x4000 43FF	4KB	Reserved
	0x4000 4400 – 0x4000 47FF	1KB	USART2
	0x4000 4800 – 0x4000 4BFF	1KB	USART3
	0x4000 4C00 – 0x4000 53FF	2KB	Reserved
	0x4000 5400 – 0x4000 57FF	1KB	I2C1
	0x4000 5800 – 0x4000 67FF	4KB	Reserved
	0x4000 6800 – 0x4000 6BFF	1KB	CSM
	0x4000 6C00 – 0x4000 6FFF	1KB	Reserved
	0x4000 7000 – 0x4000 73FF	1KB	PWR
	0x4000 7400 – 0x4000 FFFF	35KB	Reserved
APB2	0x4001 0000 – 0x4001 03FF	1KB	SYSCFG
	0x4001 0400 – 0x4001 07FF	1KB	EXTI
	0x4001 0800 – 0x4001 23FF	7KB	Reserved
	0x4001 2400 – 0x4001 27FF	1KB	ADC1
	0x4001 2800 – 0x4001 2BFF	1KB	Reserved
	0x4001 2C00 – 0x4001 2FFF	1KB	TIM1
	0x4001 3000 – 0x4001 33FF	1KB	SPI1
	0x4001 3400 – 0x4001 37FF	1KB	DBG
	0x4001 3800 – 0x4001 3BFF	1KB	USART1
	0x4001 3C00 – 0x4001 3FFF	1KB	COMP
	0x4001 4000 – 0x4001 43FF	1KB	TIM14
	0x4001 4400 – 0x4001 47FF	1KB	TIM16
	0x4001 4800 – 0x4001 4BFF	1KB	TIM17
	0x4001 4C00 – 0x4001 FFFF	45KB	Reserved
AHB	0x4002 0000 – 0x4002 03FF	1KB	DMA
	0x4002 0400 – 0x4002 0FFF	3KB	Reserved
	0x4002 1000 – 0x4002 13FF	1KB	RCC
	0x4002 1400 – 0x4002 1FFF	3KB	Reserved
	0x4002 2000 – 0x4002 23FF	1KB	Flash Interface
	0x4002 2400 – 0x4002 2FFF	3KB	Reserved
	0x4002 3000 – 0x4002 33FF	1KB	CRC
	0x4002 3400 – 0x4002 FFFF	51KB	Reserved
	0x4003 0000 – 0x4003 03FF	1KB	DIV
	0x4003 0400 – 0x47FF FFFF	~128MB	Reserved
	0x4800 0000 – 0x4800 03FF	1KB	PORT A
	0x4800 0400 – 0x4800 07FF	1KB	PORT B
	0x4800 0800 – 0x4800 0BFF	1KB	PORT C
	0x4800 0C00 – 0x4800 0FFF	1KB	PORT D

1.2.2 内置的 SRAM

内置最大可到 8K 字节的静态 SRAM。它可以以字节（8 位）、半字（16 位）或字（32 位）进行访问。SRAM 起始地址为 0x2000 0000。

SRAM 可以被 CPU 用最快的系统时钟且不插入任何等待进行访问。

1.2.3 闪存存储器概述

闪存存储器分为两个存储区域：

- 由应用数据和用户数据区组成的主闪存存储块。
- 由数据存储区、选项字节和系统存储器组成的信息块：
- 数据存储区：用来存储非易失数据，或放置额外的用户程序
- 选项字节（Option bytes）：包括硬件和存储保护用户配置选项。
- 系统存储器（System memory）：Bootloader 代码。

闪存接口基于 AHB 协议执行指令和数据存取。闪存接口的预取缓冲功能可加速 CPU 执行代码的速度。

1.2.4 启动配置

在芯片中，可通过 BOOT0 引脚电平状态及选项字节中的 BOOT0SEL、nBOOT0 和 nBOOT1 位的配置选择三种不同的启动模式，如下表所示：

表 1-2 启动模式

启动模式	启动模式配置			
	nBOOT1	BOOT0 引脚	nBOOT0	BOOT0SEL
主闪存存储器	x	0	x	1
系统存储器	1	1	x	1
内置 SRAM	0	1	x	1
主闪存存储器	x	x	1	0
系统存储器	1	x	0	0
内置 SRAM	0	x	0	0

用户通过设置 BOOT0 引脚值和 BOOT0SEL、nBOOT0 和 nBOOT1 位来选则启动模式，在器件复位后，芯片根据不同的启动模式配置，从而确定启动模式。

器件复位后，CPU 先从 0x0000 0000 地址开始获取栈顶值，再从 0x0000 0004 地址获取引导代码的基地址，并且从基地址开始执行程序。

启动模式主要分为三种：主闪存存储器启动、系统存储器启动和内置 SRAM 启动。

主闪存存储器启动：主闪存存储器的起始地址是 0x0800 0000，当其被选为启动模式时，被映射到启动存储空间（0x0000 0000），但是闪存存储器的内容依旧可以从起始地址（0x0800 0000）访问，即当主闪存存储器被选为启动模式，启动地址和起始地址都可以访问闪存存储器。

系统存储器启动：系统存储器的起始地址是 0x1FFF F400，当其被选为启动模式时，被映射到启动存储空间（0x0000 0000），但是系统存储器的内容依旧可以从起始地址（0x1FFF F400）访问，即当系统存储器被选为启动模式，启动地址和起始地址都可以访问系统存储器。

内置 SRAM 启动：内置 SRAM 的起始地址是 0x2000 0000，当其被选为启动模式时，被映射到启动存储空间（0x0000 0000），但是内置 SRAM 的内容依旧可以从起始地址（0x2000 0000）访问，即当内置 SRAM 被选为启动模式，启动地址和起始地址都可以访问内置 SRAM。

1.2.5 引导程序

出厂后引导程序存放在系统存储器中，可以通过串口（如 USART1）进行 ISP 编程。

2 PWR 电源控制

电源控制 PWR（Power Controller）主要涉及芯片的供电系统、电源管理器和低功耗模式等功能。

2.1 供电系统

芯片的电源分配分为以下四个部分：

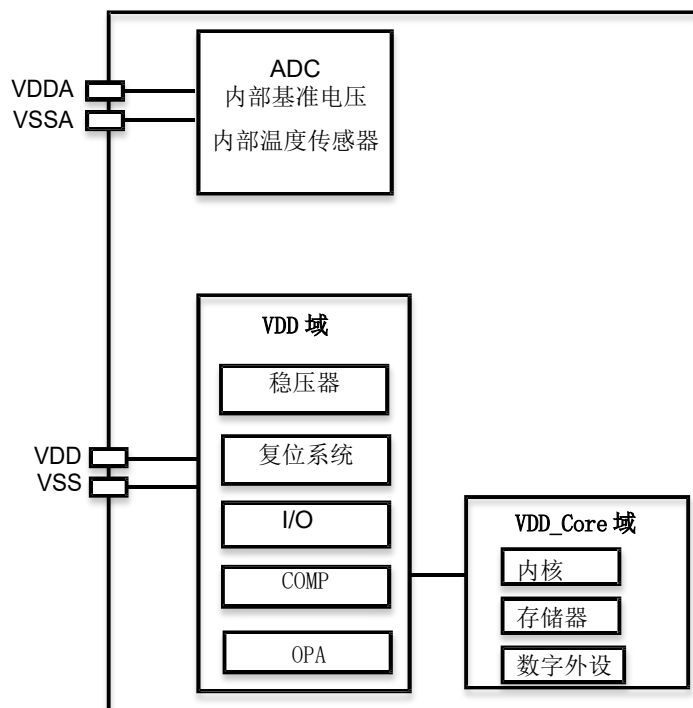


图 2-1 电源控制功能框图

- 由 VDDA 和 VSSA 提供的模拟电源，为芯片模拟模块提供电压，用于 ADC 模块、内部基准电压、内部温度传感器等。
- 由 VDD 和 VSS 提供的数字电源，用于数字部分和 I/O 引脚工作。
- 在供电系统中要求在相应的电源引脚上外接 10uF 和 100nF 的电容，并尽量靠近引脚摆放。

注：

- 在有 VDDA 和 VSSA 的封装中，VDDA 和 VSSA 不可悬空，且 VDD 和 VDDA 电压差要小于 50mV。
- 在没有 VDDA 和 VSSA 的封装中，已经在封装内部将 VDD 和 VDDA 连接，VSSA 和 VSS 连接。

2.1.1 模拟模块供电

模拟模块供电主要给 MCU 内部模拟电路提供电源，主要包括 ADC 模块、复位系统等，因此电源的稳定性影响模拟模块的工作性能。

ADC 的精度有一部分取决于 ADC 模块供电的稳定性。针对有需要高精度的 ADC 的应用，为了过滤和屏蔽来自印刷电路板上的毛刺对 ADC 采样的干扰、提高 ADC 的转换精度，ADC 需要一个独立稳定的外部电源供电。

- ADC 的电源引脚为 VDDA
- ADC 独立的电源地 VSSA

2.1.2 数字模块供电

VDD、VSS 是芯片数字模块供电电源端口，主要为 IO 供电以及通过稳压器为内核、内置数字外设和存储器等供电。

2.1.3 VDD 域

VDD 域主要给 PMU 的模拟部分（LDO 及电源检测）、ADC/COMP/OPA 的模拟部分供电，在上电后保持工作状态。

2.1.4 1.2V 域

1.2V 域主要给芯片的内核、内存和外设提供供电，在上电后默认是开启状态，在进入低功耗待机模式时，芯片会硬件选择进入稳压器的低功耗模式，在唤醒后芯片会自动切换进入稳压器的正常功耗模式。主要有以下几种工作状态：

运行模式（Run Mode）：1.2V 域以正常的功耗模式运行，内存、外设都正常工作。

睡眠模式（Sleep Mode）：1.2V 域以正常的功耗模式工作，CPU 进入睡眠模式，内存、外设都以正常的功耗模式工作。

停机模式（Stop Mode）：1.2V 域以低功耗模式工作，只保持寄存器和 RAM 的内容。

深度停机模式（DeepStop Mode）：1.2V 域以更低功耗模式工作，只保持寄存器和 RAM 的内容。

2.2 电源管理器

2.2.1 POR 上电复位和 PDR 掉电复位

芯片有一个完整的上电复位（POR）和掉电复位（PDR）电路。当供电电压达到芯片最低的工作电压后系统能正常工作，当供电电压低于芯片最低的工作电压时，系统处于非工作状态。在对芯片进行上电或者掉电操作时，上电操作的供电电压达到芯片最低的工作电压时，芯片产生的上电复位会被释放；

芯片在掉电操作时，电压低于最低工作电压时，芯片会产生掉电复位。

当 VDD 低于指定的 POR/PDR 限位电压时，系统保持为复位状态，NRST 复位引脚处于低电平，关于复位持续时间（ $t_{RSTTEMPO}$ ）等上电复位和掉电复位的细节请参考数据手册的电气特性部分。

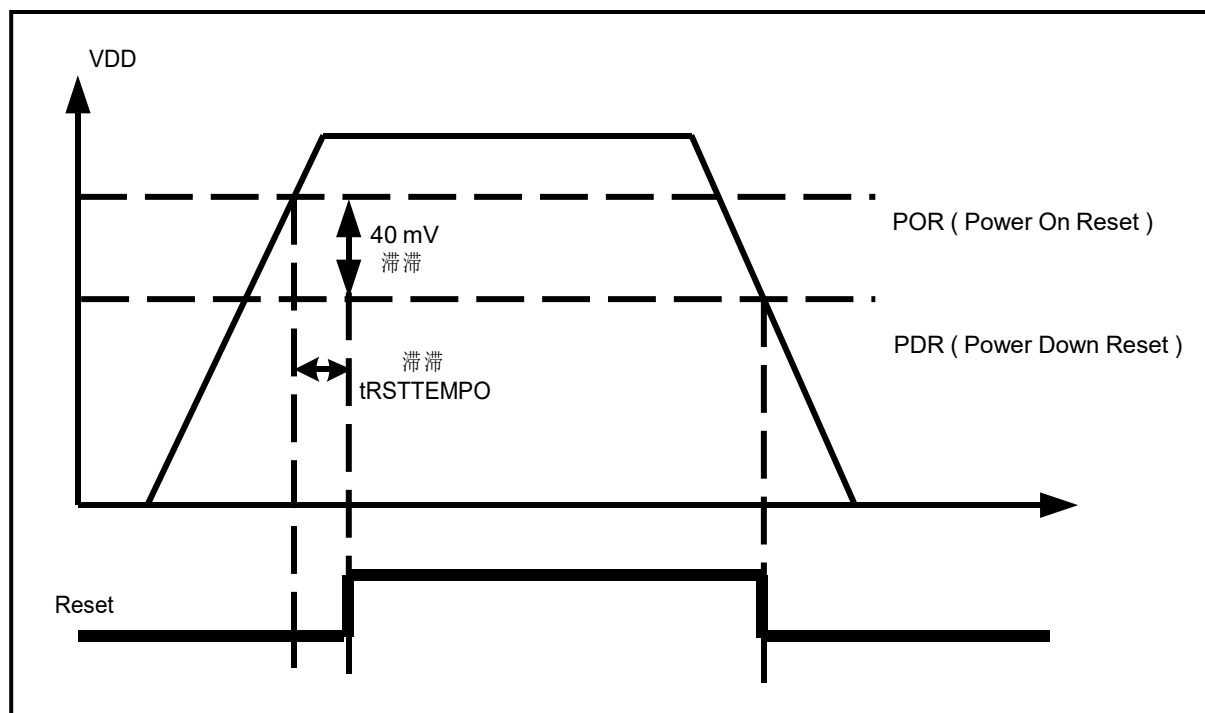


图 2-2 上电复位和掉电复位波形图

2.2.2 PVD 可编程电压监测器

可编程电压监测器 PVD (Programmable Voltage Detector) 可以用来监视芯片的供电电压，在供电电压下降到给定的阈值以下时，产生一个中断，软件可以做紧急处理。当供电电压又恢复到给定的阈值以上时，也会产生一个中断，软件处理供电恢复。供电下降的阈值与供电上升的阈值有一个固定的差值，这就是 PVD 迟滞电压，通过列出的 PVD 阈值数据可以看到这个差别。引入这个差值的目的是为了防止电压在阈值上下小幅抖动，而频繁地产生中断。

用户可以通过软件设置电源控制寄存器 PWR_CR 中的 PLS 位的阈值电压与芯片供电电压进行比较，用来监控电源。

通过设置电源控制寄存器 PWR_CR 中的 PVDE 位来使能 PVD。电源控制/状态寄存器 PWR_CSR 中的 PVDO 标志用来表明 VDD 是高于还是低于 PWR_CR 中的 PLS 位选择的阈值电压。

PVD 中断对应到外部中断 EXTI 16，如果用户有配置外部中断 EXTI 16，该事件就会产生中断，进入相应中断服务函数。当 VDD 下降到 PVD 阈值以下或当 VDD 升到 PVD 阈值以上时，根据设置的外部中断 EXTI 16 的上升/下降边沿触发，就会产生 PVD 中断（也可以通过软件配置产生 PVD 复位）。用户可以在中断中做一些对应的操作，例如：当条件触发且掉电的速率慢于中断中处理程序的执行时间

时，如果系统需要进入特别保护状态，那么可以通过执行紧急关闭任务，以保存系统一些重要数据，同时也对外设进行相应的保护等操作。

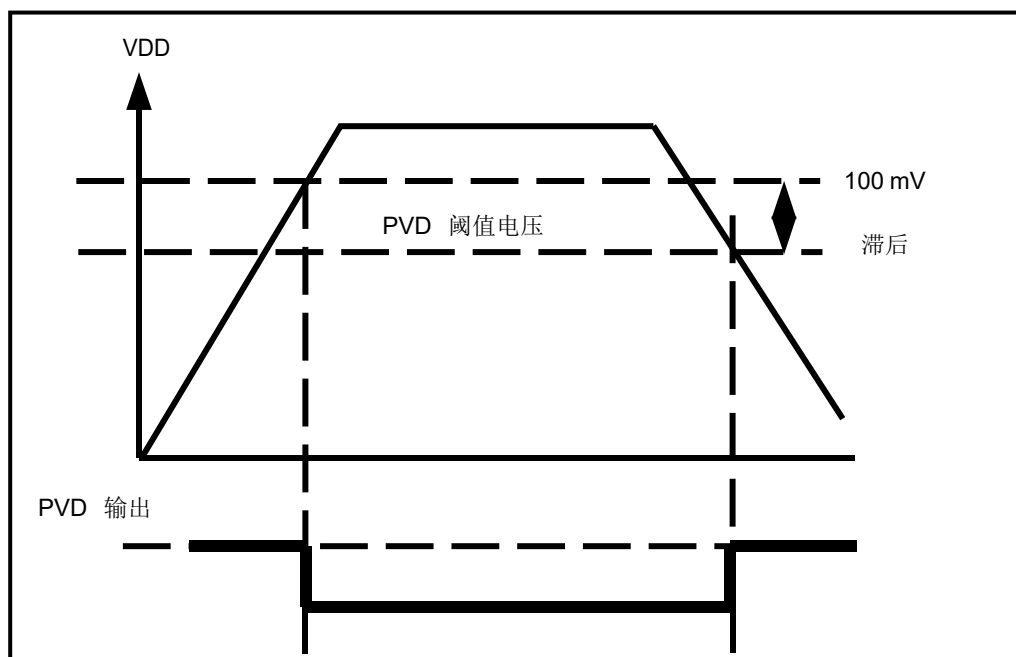


图 2-3 PVD 阈值波形图

2.3 功耗控制

2.3.1 功耗控制概述

为了延长电池供电类产品寿命，在 MCU 不需要工作时，可以利用 MCU 提供的多种低功耗模式来节省功耗；当需要 MCU 重新开始工作时，可以通过多种唤醒源唤醒 MCU，从而达到节省芯片电流消耗的目的。

芯片有三种低功耗模式，这些模式电源消耗不同、唤醒时间不同、唤醒源不同，用户可以根据应用需求，选择最佳的低功耗模式。

三种低功耗模式为：

睡眠模式（Sleep Mode）：CPU 停止，所有芯片外设包括 CPU 的外设，如 NVIC、SysTick 等仍在运行。

停机模式（Stop Mode）：1.2V 域以低功耗模式工作，CPU 及所有外设时钟都停止，只保持寄存器和 RAM 的内容。

深度停机模式（DeepStop Mode）：1.2V 域以更低功耗模式工作，CPU 及所有外设时钟都停止，只保持寄存器和 RAM 的内容。

此外，在运行模式下，可以通过以下方式中的一种降低功耗：

- **降低系统时钟频率：**在满足应用需求的同时可以选择低速时钟频率或采用高速时钟和低速时钟

循环切换的方式来节省功耗。

- 关闭 APB 和 AHB 总线上未被使用的外设时钟：用户只使能应用需要的时钟，其他多余的时钟都选择关闭。
- 选择低电压供电：供电电压越高芯片的耗电越大，所以应用中在芯片安全的供电电压范围内可以选择合适的供电电压。

表 2-1 低功耗模式列表

模式	进入方式	唤醒方式	对 1.2V 域时钟的影响	对 VDD 域时钟的影响	电压稳压器	对数据和寄存器的影响	注意事项
睡眠模式 (Sleep Mode)	WFI (Wait for Interrupt)	任一中断	CPU 时钟关，对其他时钟和 ADC 时钟无影响	无	开		外设时钟继续维持，寄存器和 SRAM 的内容保持
	WFE (Wait for Event)	唤醒事件					
停机模式 (Stop Mode)	清除 LPDS 位；设置 SLEEPDEEP 位；WFI 或 WFE；	任一外部中断（在外部中断寄存器中设置）或事件、IWDG 中断（非复位）唤醒	所有使用 1.2V 域的时钟都已关闭	HSI 关闭	开	寄存器和 SRAM 的内容保持	进入低功耗模式前不使用的 GPIO 应该设置模拟输入状态
深度停机模式 (DeepStop Mode)	设置 LPDS 位；SLEEPDEEP 位；WFI 或 WFE；	任一外部中断（在外部中断寄存器中设置）或事件、IWDG 中断（非复位）唤醒			开	寄存器和 SRAM 的内容保持	进入低功耗前不使用的 GPIO 应该设置模拟输入状态

2.3.2 运行模式降低系统时钟

在满足应用需求的同时可以选择低速时钟频率或采用高速时钟和低速时钟循环切换的方式来节省功耗。

芯片的系统时钟可以灵活配置，用户可以选择不同的时钟源作为系统时钟；也可以通过配置不同时钟分频器来降低系统时钟（SYSCLK、HCLK、PCLK1）的频率。

进入睡眠模式前，降低外设的时钟频率可以有效节省睡眠模式下的功耗。

2.3.3 外设时钟的控制

在芯片执行程序过程中，可以通过关闭外设时钟来降低功耗。

使用睡眠模式时，在执行 WFI 或 WFE 指令前可以关闭外设的时钟，有效地降低睡眠模式下外设的电流消耗。

外设的时钟主要挂在 AHB 外设时钟使能寄存器（RCC_AHBENR）、APB1 外设时钟使能寄存器（RCC_APB1ENR）总线上，用户可以单独配置寄存器外设控制位关闭外设时钟。

2.3.4 Sleep Mode 睡眠模式

2.3.4.1 睡眠模式进入

通过执行 WFI (Wait For Interrupt) /WFE (Wait for Event) 指令，可以请求 MCU 进入睡眠模式。根据 CPU 系统控制寄存器 (SCB->SCR) 中的 SLEEPONEXIT 位的值，有两种可用于选择进入睡眠模式的机制：

SLEEPNOW: 如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，MCU 立即进入睡眠模式。

SLEEPONEXIT: 如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出后 MCU 立即进入睡眠模式。

在睡眠模式下，所有的 I/O 引脚都保持在运行模式时的状态。

2.3.4.2 睡眠模式退出 中断或事件发生后，睡

眠模式立即被唤醒。

表 2-2 SLEEPNOW 模式

SLEEP NOW 模式	说明
进入	在以下条件下执行 WFI (Wait for Interrupt) 或 WFE (Wait for Event) 指令： SLEEPDEEP = 0 SLEEPONEXIT = 0
退出	如果执行 WFI 进入睡眠模式：中断 (参考中断向量表) 如果执行 WFE 进入睡眠模式：唤醒事件 (参考唤醒事件管理)
唤醒延时	立即唤醒

表 2-3 SLEEPONEXIT 模式

SLEEP ON EXIT 模式	说明
进入	在以下条件下执行 WFI (Wait for Interrupt) 指令： SLEEPDEEP = 0 SLEEPONEXIT = 1
退出	中断 (参考中断向量表)
唤醒延时	立即唤醒

2.3.5 Stop Mode 停机模式

CPU 深度睡眠模式+外设的时钟控制组成了停机模式。停机模式下，CPU 进入深度睡眠模式，1.2V 域的所有时钟都被停止，HSI 被禁止，SRAM 和寄存器内容被保留下来。

在停机模式下，所有的 I/O 引脚都保持在运行模式时的状态。

2.3.5.1 停机模式进入

通过对独立的控制位进行编程，停机模式根据唤醒方式的不同有两种进入方式：

- 等待外部中断线 WFI 方式进入停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 0 ； CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFI 被执行时，MCU 立即进入停机模式。
- 等待外部事件 WFE 方式进入停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 0 ； CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFE 被执行时，MCU 立即进入停机模式。

进入停机模式时可选择以下功能：

- 独立看门狗（IWDG）：可通过写入独立看门狗的键寄存器或硬件选择来启动独立看门狗，独立看门狗可以选择中断或者复位方式唤醒芯片，中断方式唤醒芯片后 MCU 继续执行进入低功耗前的程序，复位方式唤醒后 MCU 执行复位；用户可以选择关闭 LSI 时钟源从而关闭独立看门狗。
- 内部低速振荡器（LSI 振荡器）：通过 RCC 控制/状态寄存器（RCC_CSR）的 LSION 位来设置。

在停机模式下，如果在进入该模式前 ADC 没有被关闭，那么 ADC 仍然消耗电流。通过设置寄存器 ADC_ADCFG 的 ADEN 位为 0 可关闭这个外设。其它没有使用的 GPIO 需要设置模拟输入模式，否则有电流消耗。

2.3.5.2 停机模式退出

当停机模式被中断或事件唤醒且退出后，系统时钟源硬件自动选择为 HSI 振荡器，如果选择其它时钟源作为系统时钟需用户重新配置。

当电压稳压器处于运行模式下，系统从停机模式退出时，将会有一段额外的启动延时。

表 2-4 停机模式

停机模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt）或 WFE（Wait for Event）指令： 置位 CPU 系统控制寄存器中的 SLEEPDEEP 位； 复位电源控制寄存器（PWR_CR）中的 LPDS 位； 系统时钟切换至 LSI 或 HSI； 注：为了进入停机模式，所有的外部中断的请求位（中断事件挂起寄存器 EXTI_PR）标志都必须被清除，否则停机模式的进入流程将会被跳过，程序继续运行。
退出	在以下条件下执行 WFI（Wait for Interrupt）指令： 任一外部中断线被设置为中断模式（相应的外部中断向量在 NVIC 中必须使能），参见中断向量表 Wait for Event； 在以下条件下执行 WFE（Wait for Event）指令：任一外部中断线被设置为事件模式，例如看门狗中断；
唤醒延时	LSI 或 HSI 的唤醒时间和电压稳压器唤醒产生的额外时间
注意事项	在进入停机模式时需将不使用的 GPIO 设置成模拟输入模式

2.3.6 DeepStop Mode 深度停机模式

深度停机是在 CPU 深度睡眠模式的基础上结合了外设的时钟控制和电压稳压器控制机制的一种低功耗模式。在深度停机模式下，1.2V 域的所有时钟都被停止，HSI 功能被禁止，SRAM 和寄存器内容被保留下来。

在深度停机模式下，所有的 I/O 引脚都保持在运行模式时的状态。

2.3.6.1 深度停机模式进入

通过对独立的控制位进行编程，深度停机模式根据唤醒的方式不同有两种进入方式：

- 等待外部中断线 WFI 方式进入深度停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 1；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFI 被执行时，MCU 立即进入深度停机模式。
- 等待外部事件 WFE 方式进入深度停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 1；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFE 被执行时，MCU 立即进入深度停机模式。

进入停机模式时可选择以下功能：

- 独立看门狗（IWDG）：可通过写入独立看门狗的键寄存器或硬件选择来启动独立看门狗，独立看门狗可以选择中断或者复位方式唤醒芯片，中断方式唤醒芯片后 MCU 继续执行进入低功耗前的程序，复位方式唤醒后 MCU 执行复位；用户可以选择关闭 LSI 时钟源从而关闭独立看门狗。
- 内部低速振荡器（LSI 振荡器）：通过 RCC 控制/状态寄存器（RCC_CSR）的 LSION 和 LSI_OEN_LV 位来设置。

在深度停机模式下，如果在进入该模式前 ADC 没有被关闭，那么 ADC 仍然消耗电流。通过设置寄存器 ADC_ADCFG 的 ADEN 位为 0 可关闭这个外设。其他没有使用的 GPIO 需要设置模拟输入模式，否则有电流消耗。

2.3.6.2 深度停机模式退出

当深度停机模式被中断或事件唤醒且退出后，系统时钟为 HSI 振荡器，如果选择其它时钟源作为系统时钟需用户重新配置。

当电压稳压器处于运行低功耗模式下，系统从深度停机模式退出时，将会有一段额外的启动延时。

表 2-5 深度停机模式

深度停机模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt）或 WFE（Wait for Event）指令： 置位 CPU 系统控制寄存器中的 SLEEPDEEP 位； 复位电源控制寄存器（PWR_CR）中的 LPDS 位； 系统时钟切换至 LSI 或 HSI 注：为了进入深度停机模式，所有的外部中断的请求位（中断事件挂起寄存器

深度停机模式	说明
	EXTI_PR) 标志都必须被清除，否则深度停机模式的进入流程将会被跳过，程序继续运行。
退出	在以下条件下执行 WFI (Wait for Interrupt) 指令： 任一外部中断引线被设置为中断模式（相应的外部中断向量在 NVIC 中必须使能），参见中断向量表 Wait for Event； 在以下条件下执行 WFE (Wait for Event) 指令：任一 一外部中断线被设置为事件模式，例如看门狗中断；
唤醒延时	LSI 或 HSI 的唤醒时间和电压稳压器唤醒产生的额外时间
注意事项	在进入深度停机模式时需将不使用的 GPIO 设置成模拟输入模式

2.4 电源控制寄存器

表 2-6 电源控制寄存器概览

Offset	Acronym	RegisterName	Reset
0x00	PWR_CR	电源控制寄存器	0x00000000
0x04	PWR_CSR	电源控制状态寄存器	0x00000000
0x24	PWR_CFGR	电源配置寄存器	0x00000040
0x30	PWR_MEMCR	电源存储控制寄存器	0x00000000

2.4.1 PWR_CR 电源控制寄存器

地址偏移：0x00 复位值：

0x00000600

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

Bit	Field	Description
31:13	Reserved	保留，始终读为 0
12: 9	PLS	PVD 电平选择 (PVD level selection) 这些位用于选择电源电压监测器的电压阈值。 0000: 保留 0100: 3.0V 1000: 4.2V 0001: 2.1V 0101: 3.3V 1001: 4.5V 0010: 2.4V 0110: 3.6V 1010: 4.8V 0011: 2.7V 0111: 3.9V 其他: 保留 注： 详细说明参见数据手册中的电气特性部分。
8	Reserved	保留，始终读为 0
7:5	Reserved	保留，始终读为 0
4	PVDE	电源电压监测器 (PVD) 使能 (Power voltage detector enable) 1 = 开启 PVD 0 = 禁止 PVD
3:1	Reserved	保留，始终读为 0

Bit	Field	Description
0	LPDS	<p>深睡眠下的低功耗（Low Power Deepstop）</p> <p>1：进入停机模式时，电压稳压器处于低功耗模式。</p> <p>0：进入停机模式时，电压稳压器处于正常功耗模式。</p> <p>当进入停机模式时，LPDS = 1 时的电流小于 LPDS = 0 时的电流。详见该芯片对应的数据手册。</p>

2.4.2 PWR_CSR 电源控制/状态寄存器

地址偏移：0x04 复位值：0x00000000（从待机模式唤醒时不被清除）

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												PVDO		Res	
												r			

Bit	Field	Description
31:3	Reserved	保留，始终读为 0
2	PVDO	<p>PVD 输出（PVD output）</p> <p>当 PVD 被 PVDE 位使能后该位才有效。</p> <p>1 = VDD/VDDA 低于由 PLS[3: 0]选定的 PVD 阈值</p> <p>0 = VDD/VDDA 高于由 PLS[3: 0]选定的 PVD 阈值</p> <p>注：在待机模式下 PVD 被停止。因此，待机模式后或复位后，直到设置 PVDE 位之前，该位为 0。</p>
1:0	Reserved	保留，始终读为 0

2.4.3 PWR_CFGR 电源配置寄存器

地址偏移：0x24 复位值：

0x00000040

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						LSICAL						LSICALSEL			
						rw						w			

Bit	Field	Description
31:10	Reserved	保留，始终读为 0
9:5	LSICAL	<p>内部低速时钟校准位（Internal Low-speed Clock Calibration Bits）</p> <p>在系统启动时，这些位被自动初始化为出厂校准值，用户可以写入其他校准值，但读出始终为出厂校准值。如果 LSICALSEL=0x1F，写入的值可以重新校正 LSI 频率，否则写入的值不起作用</p>

Bit	Field	Description
4:0	LSICALSEL	内部低速时钟校准值选择方式 (Internal Low-speed Calibration Select) 初值为 0, 当写入不同值时: 写入 1F: 选择寄存器 LSICAL 的值, 写入 1F 后仍读出为 0。 写入其他: 选择出厂校准值

2.4.4 PWR_MEMCR 电源存储控制寄存器

地址偏移: 0x30 复位值:

0x00000000

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

Bit	Field	Description
31:13	Reserved	保留, 始终读为 0
12	Deepstop_sram	0:deepstop 模式下, sram 进入 retention 1:deepstop 模式下, sram 进入 shutdown
11:5	Reserved	保留, 始终读为 0
4:0	MEM_LOCK	PWR_MEMCR 配置锁 对该寄存器其他位进行配置时, 必须解锁 MEM_LOCK=0x15 时, 可以配置其他控制位; 否则, 其他位无法被更改

3 RCC 时钟和复位

3.1 复位单元

3.1.1 简介

系统共有三大类复位：电源复位、系统复位和备份域复位。

3.1.2 功能框图

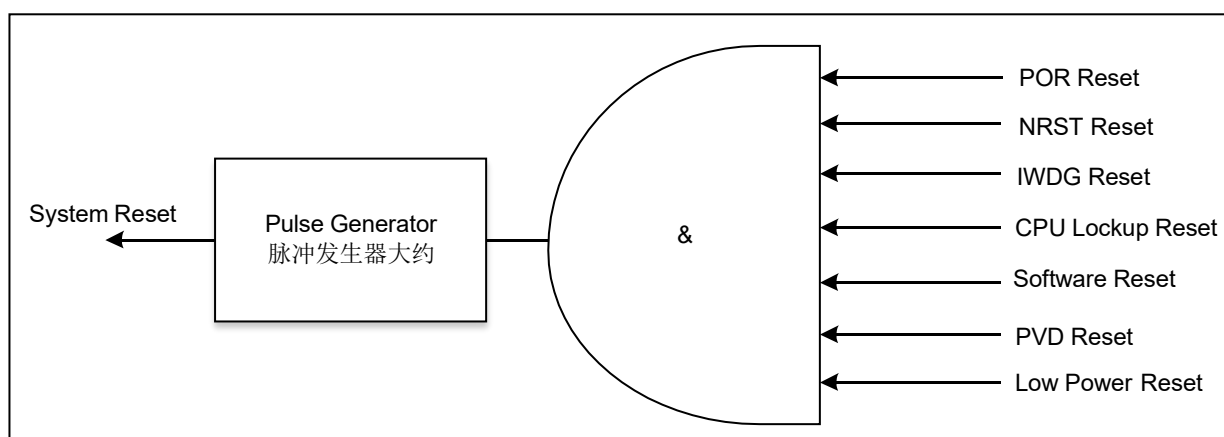


图 3-1 复位功能框图

3.1.3 主要特征

- 复位事件判定：通过控制状态寄存器（RCC_CSR）中的复位标志位来进行判断。
- 电源复位：复位所有寄存器。
- 系统复位：除了时钟控制寄存器（RCC_CSR）中的复位标志以及内部低速振荡器使能标志、电源控制寄存器（PWR_CSR）中的唤醒标志、DBG 控制寄存器（DBG_CR）不受系统复位影响，其余寄存器都将被系统复位。

3.1.4 功能描述

3.1.4.1 电源复位（POR Reset）

电源复位有以下方式：

- 上电复位 (Power-on Reset)

- 掉电复位 (Power-down Reset)

3.1.4.2 系统复位 (System Reset)

系统复位有以下几种方式:

- 外部复位 (NRST Reset)
- 独立看门狗复位 (IWDG Reset)
- 软件复位 (Software Reset)
- CPU 死锁复位 (CPU Lockup Reset)
- PVD 复位 (PVD Reset)
- 低功耗复位 (Low Power Reset)

外部复位 (NRST Reset):

- 当通过 NRST Pin 输入低电平时, 将会发生外部复位。

独立看门狗复位 (IWDG Reset):

- 计数器开始从其复位值 0xFFFF 开始递减, 当递减到达 0x0000 时, 将会发生独立看门狗复位。
- 比较/输出如果程序异常, 无法正常喂狗, 将会发生独立看门狗复位。
- 具体请参考独立看门狗章节

软件复位 (Software Reset):

- 可以通过将 SCB_AIRCR[SYSRESETREQ]置 1, 将会发生软件复位

CPU 死锁复位 (CPU Lockup Reset):

- 配置控制状态寄存器 (RCC_CSR) 的 LOCKUPEN 位为 1, CPU 死锁复位使能;
- 当 CPU 进入锁定状态将会发生 CPU 死锁复位。

PVD 复位 (PVD Reset):

- 配置控制状态寄存器 (RCC_CSR) 的 PVDRSTEN 位为 1, PVD 复位使能;
- 配置电源控制寄存器 (PWR_CR) 的 PVDE 位为 1, 使能 PVD;
- 配置电源控制寄存器 (PWR_CR) 的 PLS 位进行 PVD 阈值选择;
- 检测 VDD 电源, 当 VDD 电源低于选择的阈值电压时将会发生 PVD 复位。

低功耗复位 (Low Power Reset):

- 为防止应用程序误进入低功耗模式，可以通过配置选项字节空间 `nRST_STOP` 位为 0，在误进入低功耗模式前进行系统复位。
- 通过配置选项字节空间 `nRST_STOP` 位为 0，系统将被复位，而不是进入停机模式。
- 具体请参考嵌入式闪存章节

低功耗复位 (Low Power Reset):

- 上电复位
- 配置备份区域控制寄存器 `RCC_BDCR` 中的 `BDRST` 位

3.2 时钟单元

3.2.1 简介

四个可配置的独立系统时钟源:

- 内部高速时钟 (HSI144)
- 内部高速时钟 (HSI8)
- 外部高速时钟 (HSE)
- 内部低速时钟 (LSI)

一个可配置的独立外设时钟源:

- 外部低速时钟 (LSE)

3.2.2 功能框图

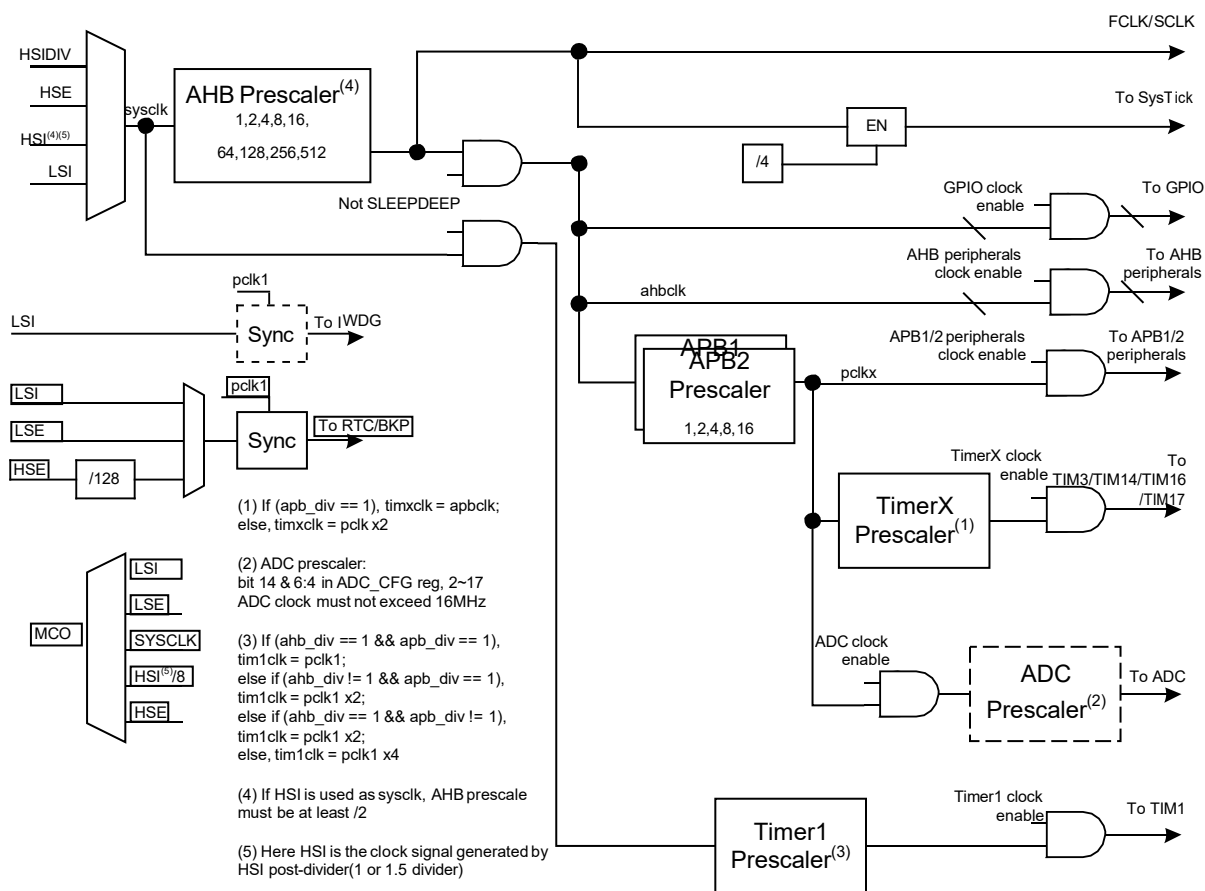


图 3-2 时钟树

3.2.3 主要特征

通过时钟配置寄存器（RCC_CFGR）的预分频控制位来分别配置 AHB、APB1 和 APB2 总线的时钟频率。AHB、APB1 和 APB2 总线时钟的最大频率是 72MHz。

3.2.4 功能描述

3.2.4.1 外部高速时钟（HSE） 外部高

速时钟的时钟源有以下一种：

- 外部高速输入时钟

通过配置时钟控制寄存器（RCC_CR）的 HSEON 位来选择此时钟输入方式。

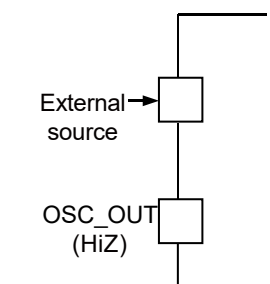


图 3-3 外部高速输入时钟

上图外部高速输入时钟模块框图

外部高速时钟具有以下特征：

- 必须提供外部高速时钟源
- 输入频率范围 4~24MHz
- 推荐外部高速时钟信号为 50%占空比（方波、正弦波），详情请参阅数据手册中有关电气特性部分
- 在使能 HSE 时，必须驱动 OSC_IN 管脚，而 OSC_OUT 引脚在未被使用时应该保持悬空

注意事项：

- 一旦 HSE 启用，HSE 相关配置就不能更改。如需更改配置，必须先禁止 HSE。

启用 HSE 配置流程如下：

- 配置时钟控制寄存器（RCC_CR）中的 HSEON 位为 1，使能 HSE；
- 等待时钟控制寄存器（RCC_CR）中的 HSERDY 位被置位为 1，表示 HSE 稳定，会输出有效时钟信号，此时才可被选择使用作为系统时钟或外设时钟源。

3.2.4.2 内部高速时钟（HSI144）

HSI144 时钟信号由内部 144MHz 振荡器产生，HSI144 时钟源在芯片上电后默认启用。使能 HSI144 配置步骤：

- 配置时钟控制寄存器（RCC_CR）中的 HSION 位为 1，使能 HSI144；
- 等待时钟控制寄存器（RCC_CR）中的 HSIRDY 位被置位为 1，表示 HSI 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源。

注意事项：

- 一旦 HSI144 启用，HSI144 相关配置就不能更改。如需更改配置，必须先禁止 HSI144。
- HSI144 可以由 RCC_CR 中的 HSI144DIV 位控制选择不分频/1.5 分频。
- 当系统时钟选择 HSI144 时，RCC_CFGR 中 HPRE 至少要配成 SYSCLK 2 分频

3.2.4.3 内部高速时钟（HSI8）

- HSI8 时钟信号由内部 HSI144 分频到 8MHz 而来，HSI8 时钟源在芯片上电后默认启用。
- 使能 HSI8 配置步骤同 HSI144，HSI144 使能后 HSI8 自动使能。

注意事项：

- HSI8 不受 RCC_CR 中的 HSI144DIV 位影响，始终由 HSI144 分频到 8MHz

3.2.4.4 内部低速时钟（LSI）

LSI 作为一个低功耗时钟源，为独立看门狗提供时钟源。时钟中心频率在 40kHz 左右。详情请参阅数据手册中有关电气特性部分。

使能 LSI 配置步骤：

- 配置控制状态寄存器（RCC_CSR）的 LSION 位置位为 1，使能 LSI；
- 等待控制状态寄存器（RCC_CSR）中的 LSIRDY 位被置位为 1，表示 LSI 稳定，可输出有效时钟。

注意事项：

- 一旦 LSI 启用，LSI 相关配置就不能更改。如需更改配置，必须先禁止 LSI。

3.2.4.5 外部低速时钟（LSE） 外部低速时钟的时钟源有以

下两种：

- 外部低速晶振/陶瓷谐振器输入时钟（32.768kHz 晶体振荡器产生） 其为实时时钟单元（RTC）或者其他定时功能提供低功耗且高精度的时钟源。

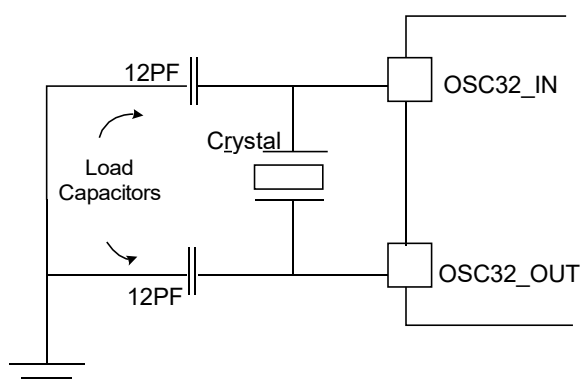


图 3-4 低速晶振/陶瓷谐振器

上图为外部低速晶振/陶瓷谐振器输入模块示意图，具体请参考数据手册。

- 外部低速输入时钟

通过配置备份域控制寄存器（RCC_BDCR）的 LSEBYP 和 LSEON 位来选择此时钟输入方式。

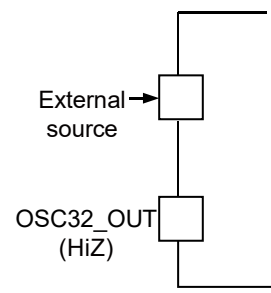


图 3-5 外部低速输入时钟

上图为外部低速输入时钟模块框图

外部低速时钟具有以下特征：

- 必须提供外部低速时钟源或晶体振荡器
- 输入频率为 32.768kHz
- 推荐外部低速时钟信号为 50%占空比（方波、正弦波），详情请参阅数据手册中有关电气特性部分
- 在使能 LSE 时，必须驱动 OSC32_IN 管脚，而 OSC32_OUT 引脚在未被使用时应该保持悬空

使能 LSE 配置步骤：

- 配置备份域控制寄存器（RCC_BDCR）的 BDRST 位为 1，使备份域软件复位；
- 配置备份域控制寄存器（RCC_BDCR）的 BDRST 位为 0，清除备份域软件复位；
- 配置电源控制寄存器（RCC_BDCR）的 DBP 位为 1，取消备份域的写保护；
- 配置备份域控制寄存器（RCC_BDCR）的 LSEON 位为 1，使能 LSE；
- 等待备份域控制寄存器（RCC_BDCR）的 LSERDY 位被置位为 1，表示 LSE 稳定，可输出有效时钟。

注意事项：

- 当使用外部低速输入时钟时，必须将备份域控制寄存器（RCC_BDCR）的 LSEBYP 位置“1”。
- 一旦 LSE 启用，LSE 相关配置就不能更改。如需更改配置，必须先禁止 LSE。

3.2.4.6 中断

表 3-1 RCC 全局中断表

中断事件	事件标志位	使能控制位	标志清除位
RCC_HSERDY	HSERDYF	HSERDYIE	HSERDYC
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC

中断事件	事件标志位	使能控制位	标志清除位
RCC_LSIRDY	LSIRDYF	LSIRDYIE	LSIRDYC
RCC_LSERDY	LSERDYF	LSERDYIE	LSERDYC

注：如上标志位/控制位/清除位都可通过时钟中断寄存器（RCC_CIR）进行配置。

3.2.4.7 系统时钟选择（SWS）

四个系统时钟源：

- 内部高速时钟 HSI8（上电后默认）
- 内部高速时钟 HSI144
- 外部高速时钟 HSE
- 内部低速时钟 LSI

系统时钟配置步骤：

- 使能需要的系统时钟源（HSI144，HSI8，HSE，LSI），每个时钟使能方式不同，具体方式请查看（HSI144，HSI8，HSE，LSI 章节）；
- 等待被选择的时钟源 RDY 信号被置位为 1，表示系统时钟源稳定（当目标时钟源稳定后，系统时钟才可以切换）；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位来选择系统时钟；
- 通过读取时钟配置寄存器（RCC_CFGR）的 SWS 位，判断当前系统时钟的时钟源。

3.2.4.8 系统时钟频率切换 系统时钟频率从低速到高速切换，或者从高速到低速切换，推荐先切换到

中速频率过度，高速低速

切换间隔至少 1 μ s。

3.2.4.9 外设复位

可以通过 APB2 外设复位寄存器（RCC_APB2RSTR）、APB1 外设复位寄存器（RCC_APB1RSTR）和 AHB 外设复位寄存器（RCC_AHBSTR）来实现相应外设的软件复位。

3.2.4.10 时钟安全系统（CSS） 时钟安全系统的具体工作方式包括软件启动和中断

处理。

软件启动：通过时钟控制寄存器（RCC_CR）的 CSSON 置位为 1 启动时钟安全系统。时钟检测器在 HSE 振荡器启动并稳定后延迟启用，当此 HSE 振荡器关闭时禁止此安全系统。

触发中断：当检测到 HSE 时钟故障时，CSS 将自行禁止 HSE 振荡器，将时钟故障事件发送到高级定时器的刹车输入端，并触发时钟安全系统中断，请求软件执行救援操作。CSS 中断连接到 CPU 的 NMI 中断。

中断判断标志及清除方式：当时钟中断寄存器（RCC_CIR）中的 CSSF 位由硬件置位为 1，表示 HSE 时钟失效导致了时钟安全系统中断。通过软件将时钟中断寄存器（RCC_CIR）中 CSSC 位写 1 来清除安全系统中断标志位 CSSF。

注意事项：

- 一旦启用 CSS，如果 HSE 时钟发生故障，CSS 中断发生，并自动生成 NMI 中断。除非 CSS 中断挂起位被清除，否则 NMI 将被不停地执行。因此，在 NMI 的处理程序中，用户必须通过设置时钟中断寄存器（RCC_CIR）的 CSSC 位为 1 来清除 CSS 中断。
- 当故障发生时，如果直接使用 HSE 振荡器作为系统时钟，系统时钟将自动切换到内部高速振荡器 HSI，并禁止 HSE 振荡器。
- 在启动时钟安全系统前，需使能 LSI，详情请参考内部低速时钟章节。

HSE 当系统时钟配置使用步骤：

- 配置时钟控制寄存器（RCC_CR）中的 HSEON 置位为 1，使能 HSE；
- 等待时钟控制寄存器（RCC_CR）中的 HSERDY 位被置位为 1，表示 HSE 稳定，可输出有效时钟；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位为 2'b01 来选择 HSE 为系统时钟；
- 通过读取时钟配置寄存器（RCC_CFGR）的 SWS 位，判断当前系统时钟的时钟源是否为 HSE；
- 等待控制状态寄存器（RCC_CSR）中的 LSIRDY 位被置位为 1，表示 LSI 稳定，可输出有效时钟；
- 配置时钟控制寄存器（RCC_CR）中的 CSSON 位为 1，时钟安全系统使能。

3.2.4.11 微控制器时钟输出（MCO）

微控制器时钟输出（MCO）允许时钟输出到外部 MCO 引脚上。相应 GPIO 端口的配置寄存器必须被配置为复用输出功能。可以选择以下五个时钟信号中的一个作为 MCO 输出时钟：

表 3-2 MCO 与时钟源对应关系

时钟配置寄存器（RCC_CFGR）的 MCO 位	时钟源
00x	没有时钟输出
010	LSI

时钟配置寄存器 (RCC_CFGR) 的 MCO 位	时钟源
011	LSE
100	SYSCLK
101	HSI/8
110	HSE

3.2.4.12 RTC 时钟

RTC CLK 时钟源可以是 HSE/128、LSE 或 LSI 时钟，通过配置备份域控制寄存器 (RCC_BDCR) 中 RTCSEL 位来选择。如果备份域没有被复位，则无法修改此选择。备份域被复位后，必须设置 DBP 位（取消备份域的写保护）为 1 才可修改此选择。

3.2.4.13 定时器时钟

当时钟配置寄存器 (RCC_CFGR) 的 PPRE1 为 3'b0xx（不分频）时，挂在 APB1 上的定时器的时钟频率和 APB1 总线频率一样；否则挂在 APB1 上的定时器的时钟频率是 APB1 总线频率的两倍；挂在 APB2 上的定时器同理。

当时钟配置寄存器 (RCC_CFGR) 的 HPRE 为 4'b0xxx（不分频）时，高级定时器 (TIM1) 的时钟频率和挂在 APB2 上的定时器的时钟频率一样，否则高级定时器 (TIM1) 的时钟频率是挂在 APB2 上的定时器的时钟频率的两倍。

3.2.4.14 独立看门狗时钟

硬件启动独立看门狗，LSI 振荡器将被自动开启，并且不能被关闭；

软件启动独立看门狗，则 LSI 振荡器需通过软件使能开启，在 LSI 振荡器稳定输出后，时钟供应给 IWDG，LSI 可以被软件关闭。

3.3 寄存器

3.3.1 寄存器总览

表 3-3 RCC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	时钟控制寄存器	0x00000001
0x04	RCC_CFGR	时钟配置寄存器	0x00000000
0x08	RCC_CIR	时钟中断寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x0C	RCC_APB2RSTR	APB2 外设复位寄存器	0x00000000
0x10	RCC_APB1RSTR	APB1 外设复位寄存器	0x00000000
0x14	RCC_AHBENR	AHB 外设时钟使能寄存器	0x00000014
0x18	RCC_APB2ENR	APB2 外设时钟使能寄存器	0x20000000
0x1C	RCC_APB1ENR	APB1 外设时钟使能寄存器	0x40000000
0x20	RCC_BDCR	备份域控制寄存器	0x00000000
0x24	RCC_CSR	控制状态寄存器	0x08000000
0x28	RCC_AHBRSTR	AHB 外设复位寄存器	0x00000000

3.3.2 RCC_CR 时钟控制寄存器

偏移地址：0x00 复位值：

0x0000 0001

访问：无等待状态，字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CSSON	Reserved	HSERDY	HSEON
												rw		r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HSI144 DIV	Reserved									HSIRDY	HSION
				rw										r	rw

Bit	Field	Description
31: 20	Reserved	保留，必须保持复位值
19	CSSON	时钟安全系统使能（Clock Security System Enable） 通过软件置“1”或清“0”。 0：禁止时钟安全监测器 1：如果外部高速振荡器稳定，使能时钟安全监测器
18	Reserved	保留，必须保持复位值
17	HSERDY	外部高速时钟稳定标志（External High-speed Clock Ready Flag） 由硬件设置。 0：外部高速晶体振荡器未稳定 1：外部高速晶体振荡器已稳定
16	HSEON	外部高速时钟使能（External High-speed Clock Enable） 通过软件置“1”或清“0”。当进入待机或停机模式时，此位由硬件清“0”。当系统时钟已经或将要使用 HSE 作为时钟源时，则禁止重置此位。 0：禁止外部高速晶体振荡器 1：使能外部高速晶体振荡器
15: 12	Reserved	保留，必须保持复位值
11	HSI144DIV	HSI144DIV：内部高速时钟分频系数（Internal high-speed clock division factor） 0：HSI144 不分频 1：HSI144 1.5 分频 注：HSI8 不受此位影响，始终由 HSI144 分频为 8MHz
10: 2	Reserved	保留，必须保持复位值

Bit	Field	Description
1	HSIRDY	内部高速时钟稳定标志 (Internal High-speed Clock Ready Flag) 由硬件置“1”，表示内部时钟已经稳定。 在 HSION 位被清除后，HSIRDY 在 3 个 AHB 时钟周期后变“0”。 0: 内部高速时钟未稳定 1: 内部高速时钟已稳定
0	HSION	内部高速时钟使能 (Internal High-speed Clock Enable) 通过软件置“1”或清“0”。当退出待机或停机模式或外部振荡器用作系统时钟并且发生故障时，此位由硬件置“1”，来迫使内部振荡器使能。当系统时钟已经或将要使用 HSI 作为时钟源时，则禁止重置此位。 0: 禁止内部高速时钟 1: 使能内部高速时钟

3.3.3 RCC_CFGR 时钟配置寄存器

偏移地址: 0x04 复位值:

0x0000 0000

访问: 无等待状态, 字, 半字和字节访问 只有当访问发生在时钟切换时, 才会

插入 1 或 2 个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					MCO			Res.							
					rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		PPRE2			PPRE1			HPRE				SWS		SW	
		rw			rw			rw				r		rw	

Bit	Field	Description
31: 27	Reserved	保留, 必须保持复位值
26: 24	MCO	微控制器时钟输出 (Micro Controller Clock Output) 由软件置“1”或清“0” 010: LSI 时钟输出 011: LSE 时钟输出 100: SYSCLK 时钟输出 101: HSI 时钟 8 分频输出 110: HSE 时钟输出 其他: 没有时钟输出 注意: 该时钟输出在启动和切换 MCO 时钟源时可能会被停止。 系统时钟通过 MCO 管脚输出时, 请保证输出时钟频率不超过 50MHz。
23: 14	Reserved	保留, 必须保持复位值
13: 11	PPRE2	PPRE2: APB2 预分频系数 通过软件设置来控制 APB2 时钟 (PCLK2) 预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频

Bit	Field	Description
10: 8	PPRE1	PPRE1: APB1 预分频系数 通过软件设置来控制 APB1 时钟 (PCLK1) 预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频
7: 4	HPRE	AHB 预分频系数 通过软件设置来控制 AHB 时钟的预分频系数。 0xxx: SYSCLK 不分频 1000: SYSCLK 2 分频 1001: SYSCLK 4 分频 1010: SYSCLK 8 分频 1011: SYSCLK 16 分频 1100: SYSCLK 64 分频 1101: SYSCLK 128 分频 1110: SYSCLK 256 分频 1111: SYSCLK 512 分频 注: 当 AHB 时钟的预分频系数大于 1 时, 必须开启预取缓冲器, 详见闪存访问控制寄存器章节。 当 SYSCLK 选择 HSI144 时, HPRE 至少要配为 SYSCLK 2 分频。
3: 2	SWS	系统时钟选择状态 (System Clock Switch Status) 00: 选择 HSI8 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 HSI144 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟
1: 0	SW	系统时钟选择 (System Clock Switch) 通过软件配置来选择系统时钟源 当从停止模式中返回时, 当直接或间接作为系统时钟的 HSE 出现故障时, 硬件 都会强制选择 HSI8 作为系统时钟。 00: 选择 HSI8 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 HSI144 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟

3.3.4 RCC_CIR 时钟中断寄存器

偏移地址: 0x08 复位值:

0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								CSSC	Res.			HSERD YC	HSIRDY C	LSERD YC	LSIRDY C
								w1c				w1c	w1c	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				HSERD YIE	HSIRDY IE	LSERD YIE	LSIRDY IE	CSSF	Res.			HSERD YF	HSIRDY F	LSERD YF	LSIRDY F
				rw	rw	rw	rw	r				r	r	r	r

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31: 24	Reserved	保留，必须保持复位值
23	CSSC	时钟安全系统中断清除（Clock Security System Interrupt Clear） 通过软件置“1”来清除安全系统中断标志位 CSSF。 0：无效 1：清除安全系统中断标志位 CSSF
22: 20	Reserved	保留，必须保持复位值
19	HSERDYC	清除 HSE 稳定中断（HSE Ready Interrupt Clear） 通过软件置“1”来清除 HSE 稳定中断标志位 HSERDYF。 0：无效 1：清除 HSE 稳定中断标志位 HSERDYF
18	HSIRDYC	清除 HSI 稳定中断（HSI Ready Interrupt Clear） 通过软件置“1”来清除 HSI 稳定中断标志位 HSIRDYF。 0：无效 1：清除 HSI 稳定中断标志位 HSIRDYF
17	LSERDYC	LSERDYC：清除 LSE 就绪中断（LSE ready interrupt clear）通过软件置 1 来清除 LSE 就绪中断标志位 LSERDYF。0：无效 0：无效 1：清除 LSE 就绪中断标志位 LSERDYF
16	LSIRDYC	清除 LSI 稳定中断（LSI Ready Interrupt Clear） 通过软件置“1”来清除 LSI 稳定中断标志位 LSIRDYF。 0：无效 1：清除 LSI 稳定中断标志位 LSIRDYF
15: 12	Reserved	保留，必须保持复位值
11	HSERDYIE	HSE 稳定中断使能（HSE Ready Interrupt Enable） 通过软件置“1”来使能或清“0”来禁止外部振荡器稳定中断。 0：禁止 HSE 稳定中断 1：使能 HSE 稳定中断
10	HSIRDYIE	HSI 稳定中断使能（HSI Ready Interrupt Rnable） 通过软件置“1”来使能或清“0”来禁止内部振荡器稳定中断。 0：禁止 HSI 稳定中断 1：使能 HSI 稳定中断
9	LSERDYIE	LSERDYIE：LSE 就绪中断使能（LSE ready interrupt enable） 通过软件置 1 来使能或禁用外部 32KHz 振荡器就绪中断。 0：禁用 LSE 就绪中断 1：使能 LSE 就绪中断
8	LSIRDYIE	LSI 稳定中断使能（LSI Ready Interrupt Enable） 通过软件置“1”来使能或清“0”来禁止内部 40KHz 振荡器稳定中断。 0：禁止 LSI 稳定中断 1：使能 LSI 稳定中断
7	CSSF	时钟安全系统中断标志（Clock Security System Interrupt Flag） 在外部振荡器时钟出现故障时，由硬件置“1”。 通过软件将 CSSC 位置“1”来清除。 0：无 HSE 时钟失效产生的安全系统中断 1：HSE 时钟失效导致时钟安全系统中断
6: 4	Reserved	保留，必须保持复位值
3	HSERDYF	HSE 稳定中断标志（HSE Ready Interrupt Flag） 在外部高速时钟稳定时，由硬件置“1”。 通过软件将 HSERDYC 位置“1”来清除。 0：无外部振荡器产生的时钟稳定中断 1：外部振荡器导致时钟稳定中断
2	HSIRDYF	HSI 稳定中断标志（HSI Ready Interrupt Flag） 在内部高速时钟稳定时，由硬件置“1”。 通过软件将 HSIRDYC 位置“1”来清除。 0：无内部 HSI 振荡器产生的时钟稳定中断 1：内部 HSI 振荡器导致时钟稳定中断

Bit	Field	Description
1	LSERDYF	LSERDYF: LSE 就绪中断标志 (LSE ready interrupt flag) 在外部低速时钟就绪时, 由硬件置“1”。通过软件将 LSERDYC 位置“1”来清除。 0: 无外部 32KHz 振荡器产生的时钟就绪中断 1: 外部 32KHz 振荡器导致时钟就绪中断
0	LSIRDYF	LSIRDYF: LSI 稳定中断标志 (LSI Ready Interrupt Flag) 在内部低速时钟稳定时, 由硬件置“1”。通过软件将 LSIRDYC 位置“1”来清除。 0: 无内部 40KHz 振荡器产生的时钟稳定中断 1: 内部 40KHz 振荡器导致时钟稳定中断

3.3.5 RCC_APB2RSTR APB2 外设复位寄存器

偏移地址: 0x0C 复位值:

0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									DBG	Res.			TIM17	TIM16	TIM14
									rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT rw	USART 1 rw	Res.	SPI1 rw	TIM1 rw	Res.	ADC	Res.							SYSCFG	
						rw								rw	

Bit	Field	Description
31: 23	Reserved	保留, 必须保持复位值
22	DBG	DBG 复位 (DBG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
21: 19	Reserved	保留, 必须保持复位值
18	TIM17	TIM17 定时器复位 (TIM17 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
17	TIM16	TIM16 定时器复位 (TIM16 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
16	TIM14	TIM14 定时器复位 (TIM14 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
15	CPT	比较器复位 (Comparator reset) 由软件置“1”或清“0”。 0: 无效 1: 复位

Bit	Field	Description
14	USART1	USART1 复位 (USART1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
13	Reserved	保留, 必须保持复位值
12	SPI1	SPI1 复位 (SPI1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
11	TIM1	TIM1 定时器复位 (TIM1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
10	Reserved	保留, 必须保持复位值
9	ADC	ADC 复位 (ADC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
8: 1	Reserved	保留, 必须保持复位值
0	SYSCFG	SYSCFG 复位 (SYSCFG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

3.3.6 RCC_APB1RSTR APB1 外设复位寄存器

偏移地址: 0x10 复位值:

0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC	Res.		PWR	Res.	CSM	Res.	BKP	Res.		I2C1	Res.		USART3	USART2	Res.
rw			rw		rw		rw			rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TIM3	Res.
														rw	

Bit	Field	Description
31	RTC	RTC 复位(RTC reset) 由软件置 1 或清‘0’。 0: 无效 1: 复位
30: 29	Reserved	保留, 必须保持复位值
28	PWR	PWR 复位 (Power Interface Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

Bit	Field	Description
27	Reserved	保留，必须保持复位值
26	CSM	CSM 复位 (CSM Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
25	Reserved	保留，必须保持复位值
24	BKP	BKP 复位 (BKP reset) 由软件置 1 或清“0”。 0: 无效 1: 复位
23: 22	Reserved	保留，必须保持复位值
21	I2C1	I2C1 复位 (I2C1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
20: 19	Reserved	保留，必须保持复位值
18	USART3	USART3 复位 (USART3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
17	USART2	USART2 复位 (USART2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
16: 2	Reserved	保留，必须保持复位值
1	TIM3	TIM3 定时器复位 (TIM3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
0	Reserved	保留，必须保持复位值

3.3.7 RCC_AHBENR AHB 外设时钟使能寄存器

偏移地址: 0x14 复位值:

0x0000 0014

访问: 无等待周期, 字, 半字和字节访问

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

Bit	Field	Description
31: 27	Reserved	保留，必须保持复位值

Bit	Field	Description
26	HWDIV	除法器时钟使能 (HWDIV clock enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
25: 21	Reserved	保留, 必须保持复位值
20	GPIOB	GPIOB 时钟使能 (GPIOB Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
19	GPIOC	GPIOC 时钟使能 (GPIOC Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
18	GPIOB	GPIOB 时钟使能 (GPIOB Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
17	GPIOA	GPIOA 时钟使能 (GPIOA Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
16: 7	Reserved	保留, 必须保持复位值
6	CRC	CRC 时钟使能 (CRC Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
5	Reserved	保留, 必须保持复位值
4	Flash	FLASH 时钟使能 (FLASH Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
3	Reserved	保留, 必须保持复位值
2	SRAM	SRAM 时钟使能 (SRAM Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启
1	Reserved	保留, 必须保持复位值
0	DMA	DMA 时钟使能 (DMA clock enable) 由软件置“1”或清“0”。 0: 时钟关闭 1: 时钟开启

3.3.8 RCC_APB2ENR APB2 外设时钟使能寄存器

偏移地址: 0x18 复位值:

0x2000 0000

访问: 无等待周期, 字, 半字和字节访问 注: 当外设时钟没有启动时, 软件不能读出外设寄存器的数值

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		EXTI	Res.						DBG	Res.			TIM17	TIM16	TIM14
		rw							rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT	USART1	Res.	SPI1	TIM1	Res.	ADC	Res.							SYSCFG	rw
rw	rw		rw	rw		rw									

Bit	Field	Description
31: 30	Reserved	保留，必须保持复位值
29	EXTI	EXTI 时钟使能 (EXTI Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
28:23	Reserved	保留，必须保持复位值
22	DBG	DBG 时钟使能 (DBG Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
21: 19	Reserved	保留，必须保持复位值
18	TIM17	TIM17 定时器时钟使能 (TIM17 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
17	TIM16	TIM16 定时器时钟使能 (TIM16 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
16	TIM14	TIM14 定时器时钟使能 (TIM14 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
15	CPT	比较器时钟使能 (Comparator Clock Enable) 由软件置“1”或清“0”。 0: 时钟禁止 1: 时钟使能
14	USART1	USART1 时钟使能 (USART1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
13	Reserved	保留，必须保持复位值
12	SPI1	SPI1 时钟使能 (SPI1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
11	TIM1	TIM1 定时器时钟使能 (TIM1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
10	Reserved	保留，必须保持复位值
9	ADC	ADC 时钟使能 (ADC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
8: 1	Reserved	保留，必须保持复位值

Bit	Field	Description
0	SYSCFG	SYSCFG 时钟使能 (SYSCFG Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

3.3.9 RCC_APB1ENR APB1 外设时钟使能寄存器

偏移地址: 0x1C 复位值:

0x4000 0000

访问: 无等待周期, 字, 半字和字节访问 注: 当外设时钟没有启动时, 软件不能读出外设寄存器的数值

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC	IWDG	Res.	PWR	Res.	CSM	Res.	BKP	Res.		I2C1	Res.		USART3	USART2	Res.
rw	rw		rw		rw		rw			rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TIM3	Res.
														rw	

Bit	Field	Description
31	RTC	RTC 时钟使能 (RTC Clock Enable) 由软件置“1”或清“0”。 0: 时钟禁止 1: 时钟使能
30	IWDG	IWDG 时钟使能 (IWDG Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
29	Reserved	保留, 必须保持复位值
28	PWR	PWR 时钟使能 (Power Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
27	Reserved	保留, 必须保持复位值
26	CSM	CSM 时钟使能 (CSM Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
25	Reserved	保留, 必须保持复位值
24	BKP	BKP 时钟使能 (Backup interface clock enable) 由软件置 1 或清“0”。 0: 时钟禁止 1: 时钟使能
23: 22	Reserved	保留, 必须保持复位值

Bit	Field	Description
21	I2C1	I2C1 时钟使能 (I2C1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
20: 19	Reserved	保留, 必须保持复位值
18	USART3	USART3 时钟使能 (USART3 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
17	USART2	USART2 时钟使能 (USART2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
16: 2	Reserved	保留, 必须保持复位值
1	TIM3	TIM3 定时器时钟使能 (TIM3 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
0	Reserved	保留, 必须保持复位值

3.3.10 RCC_BDCR 备份域控制寄存器

偏移地址: 0x20 复位值:

0x00000000

支持字, 半字和字节方式访问。该寄存器在读写操作对应 IP (RTC/LSE/BKP) 以后, 下一次读写操作前要等待对应 IP (RTC/LSE/BKP) 所在总线 (APB1/APB2, 其中 RTC 及 LSE 对应 BKP 所在 APB 总线) 的 5 个周期以上。

注: 备份域控制寄存器中 (RCC_BDCR) 的 LSEON、LSEBYP、RTCSEL 和 RTCEN 位在备份域内。因此, 在重置之后, 这些位处于写保护的状态, 在修改这些位之前, 必须将 (RCC_BDCR) DBP 位置 1。任何内部或外部重置都不会对这些位产生影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							DBP	Res.							BDRST
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.					RTCSEL		Res.					LSEBYP	LSE RDY	LSEON
rw						rw							rw	r	rw

Bit	Field	Description
31:25	Reserved	保留, 必须保持复位值
24	DBP	DBP: 取消后被区域的写保护, 在复位后, 后备寄存器处于被保护状态以防意外写入。设置这一位允许写入这些寄存器。由软件置 1 或清'0'。 0: 禁止对 RTC 和后备寄存器的访问 1: 允许对 RTC 和后备寄存器的访问
23:17	Reserved	保留, 必须保持复位值

Bit	Field	Description
16	BDRST	BDRST: 备份域软件复位 (Backup domain software reset) 通过软件置 1 或清除 0: 复位未激活 1: 复位整个备份域
15	RTCEN	RTCEN: RTC 时钟使能 (RTC clock enable) 通过软件置 1 或清除 0: 禁用 RTC 时钟 1: 使能 RTC 时钟
14:10	Reserved	保留, 必须保持复位值
9: 8	RTCSEL	RTCSEL: RTC 时钟源选择 (RTC clock source selection) 由软件设置为选择 RTC 的时钟源。一旦选择了 RTC 时钟源, 就不能再更改它, 除非备份域被重置。可以使用 BDRST 位来重置它们。 00: 没有时钟 01: LSE 振荡器用作 RTC 时钟 10: LSI 振荡器用作 RTC 时钟 11: HSE 振荡器在 128 分频后用作 RTC 时钟
7:3	Reserved	保留, 必须保持复位值
2	LSEBYP	LSEBYP: 外部低速振荡器旁路 (External low-speed oscillator bypass) 在调试模式下通过软件置 1 来旁路 LSE。只有在外部 32KHz 振荡器被禁用时, 才能写入此位。 0: 禁用 LSE 振荡器旁路模式 1: 使能 LSE 振荡器旁路模式
1	LSERDY	LSERDY: 外部低速 LSE 就绪 (External low-speed oscillator ready) 通过软件置 1 或清除来指示外部 32KHz 振荡器是否稳定。在 LSEON 被清除后, 在 3 个 AHB 时钟周期后变低。 0: 外部 32KHz 振荡器未稳定 1: 外部 32KHz 振荡器已稳定
0	LSEON	LSEON: 外部低速振荡器使能 (External low-speed oscillator enable) 通过软件置 1 或清除 0: 禁用外部 32KHz 振荡器 1: 使能外部 32KHz 振荡器

3.3.11 RCC_CSR 控制状态寄存器

偏移地址: 0x24 复位值:

0x0800 0000

访问: 0-3 等待周期, 字, 半字和字节访问 当连续对该寄存器进行访问时, 将插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRR STF	Res.	IWDGR STF	SFTRST F	PORRS TF	PINRST F	Res.	RMVF	LOCKU PF	PVDRS TF	Res.					
r		r	r	r	r		w1c	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								LOCKU PEN	PVDRS TEN	Res.				LSIRDY	LSION
								rw	rw					r	rw

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31	LPWRRSTF	低功耗管理复位标志 (Low Power Reset Flag) 在低功耗管理复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无低功耗管理复位 1: 发生低功耗管理复位
30	Reserved	保留，必须保持复位值
29	IWDGRSTF	独立看门狗复位标志 (Independent Watchdog Reset Flag) 在独立看门狗复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无独立看门狗复位发生 1: 发生独立看门狗复位
28	SFTRSTF	软件复位标志 (Software Reset Flag) 在软件复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无软件复位发生 1: 发生软件复位
27	PORRSTF	上电/掉电复位标志 (POR/PDR Reset Flag) 在上电/掉电复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无上电/掉电复位发生 1: 发生上电/掉电复位
26	PINRSTF	NRST 管脚复位标志 (PIN Reset Flag) 在 NRST 管脚复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 NRST 管脚复位发生 1: 发生 NRST 管脚复位
25	Reserved	保留，必须保持复位值
24	RMVF	清除复位标志 (Remove Reset Flag) 由软件置“1”来清除复位标志。 0: 无效 1: 清除复位标志
23	LOCKUPF	CPU 死锁复位标志 (CPU Lockup Reset Flag) 在 CPU 发生死锁复位时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 CPU 死锁复位发生 1: 发生 CPU 死锁复位
22	PVDRSTF	PVD 复位标志 (PVD Reset Flag) 在 PVD 复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 PVD 复位发生 1: 发生 PVD 复位
21: 8	Reserved	保留，必须保持复位值
7	LOCKUPEN	CPU 死锁复位使能 (CPU Lockup Reset Enable) 0: 禁止 CPU 死锁复位 1: 使能 CPU 死锁复位
6	PVDRSTEN	PVD 复位使能 (PVD Reset Enable) 0: 禁止 PVD 产生复位 1: 使能 PVD 产生复位
5: 2	Reserved	保留，必须保持复位值
1	LSIRDY	内部低速时钟稳定 (Internal Low-speed Oscillator Ready) 由硬件置“1”或清“0”来指示内部 40KHz 振荡器是否稳定。 在 LSION 清“0”后，3 个 AHB 时钟后 LSIRDY 被清“0”。 0: 内部 40KHz 振荡器时钟未稳定 1: 内部 40KHz 振荡器时钟稳定

Bit	Field	Description
0	LSION	内部低速振荡器使能 (Internal Low-speed Oscillator Enable) 通过软件置“1”或清“0”，或由电源复位清除。 0: 禁止内部 40KHz 振荡器 1: 使能内部 40KHz 振荡器

3.3.12 RCC_AHBRSTR AHB 外设复位寄存器

偏移地址: 0x28 复位值:

0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

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

Bit	Field	Description
31: 27	Reserved	保留, 必须保持复位值
26	HWDIV	除法器复位 (HWDIV Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
25:21	Reserved	保留, 必须保持复位值
20	GPIOD	GPIOD 复位 (GPIOD Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
19	GPIOC	GPIOC 复位 (GPIOC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
18	GPIOB	GPIOB 复位 (GPIOB Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
17	GPIOA	GPIOA 复位 (GPIOA Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
16: 7	Reserved	保留, 必须保持复位值
6	CRC	CRC 复位 (CRC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

Bit	Field	Description
5: 1	Reserved	保留，必须保持复位值
0	DMA	DMA 复位（DMA Reset） 由软件置“1”或清“0”。 0：无效 1：复位

4 Embedded FLASH 嵌入式闪存

4.1 简介

嵌入式闪存支持高达 64K Bytes 的片内 Main Flash，还提供了选项字节块与系统启动块（支持芯片 Boot 引导），还有保留的 Data 空间。闪存的控制器支持读操作、页擦除、整片擦除，可通过 16 位（半字）方式编程写入闪存，其擦写寿命可达 100000 次。闪存控制器在读取数据时，支持带预取缓冲器的数据接口，以支持 MCU 运行在更高的主频。

4.2 闪存构成与说明

4.2.1 闪存构成

- 闪存空间由 64 位宽的存储单元组成，既可以存代码又可以存数据。
- 主闪存块按 64 页（每页 1K 字节）或 16 个写保护块（每块 4K 字节）划分。
- 主闪存块可按页（每 1K 字节）擦除（Page Erase）。
- 以 4 页（4K 字节）为单位作为 1 个写保护块来设置写保护。
- 整个片内 Flash 由两部分组成：一部分是主存储块，另一部分是信息存储块。
- 主存储块用于存储用户代码和数据，用户代码可以对主存储器进行擦除、编程和读取操作。每个 1K 字节在主存储块中称为一页，可以执行最小单位的擦除；另外以 1 个写保护区为单位（4K 字节，4 页=1 个写保护块）进行写保护分配，如
- 表 4-1 所示。

表 4-1 Flash 存储空间

模块	区块名称	页名称	地址	大小（字节）
主存储块	写保护区 0	页 0	0x0800 0000 - 0x0800 03FF	1K
		页 1	0x0800 0400 - 0x0800 07FF	1K
		页 2	0x0800 0800 - 0x0800 0BFF	1K
		页 3	0x0800 0C00 - 0x0800 0FFF	1K

	写保护区 3	页 12	0x0800 3000 - 0x0800 33FF	1K
		页 13	0x0800 3400 - 0x0800 37FF	1K
		页 14	0x0800 3800 - 0x0800 3BFF	1K
		页 15	0x0800 3C00 - 0x0800 3FFF	1K

	写保护区 15	页 60	0x0800 F000 - 0x0800 F3FF	1K
		页 61	0x0800 F400 - 0x0800 F7FF	1K
		页 62	0x0800 F800 - 0x0800 FBFF	1K

模块	区块名称	页名称	地址	大小（字节）
		页 63	0x0800 FC00 - 0x0800 FFFF	1K

信息存储块中，除了“系统存储器 ISP”区域出厂锁定，用户不可写入外，其他区域在一定条件下用户可进行读写操作。信息存储器可分为 Data 空间、系统存储器 ISP 和选项字节三部分，Data 空间用于存储非易失数据（EEPROM），系统存储 ISP 区用于存储出厂的 ISP Bootloader，选项字节（Option byte）部分中的前 12 个字节是主存储器的写和读保护信息，剩余字节可用于存放用户特殊的数据。对于选项字节部分，用户可以通过规定的流程对其擦除、编程和读取。仅 ISP 部分由于用于固化 ISP 升级的代码，不支持用户进行擦除和编程。

表 4-2 信息块

模块	名称	地址	大小（字节）
信息块	Data 空间	0x1FE0 0000- 0x1FE0 0BFF	3K
	系统存储 ISP	0x1FFF F400 - 0x1FFF F7FF	1K
	选项字节	0x1FFF F800 - 0x1FFF F9FF	0.5K

4.2.2 选项字节说明

在选项字节页中，内容主要有写保护使能，硬件看门狗使能等。Flash 控制器可以通过选项字节中值的设置，达到使能主存储器禁止写入功能，以避免非法写入；还可以使能硬件看门狗。相关信息存储在选项字节中，修改选项字节中内容后，需要复位或重新上电后才生效，写入时需按半字高低字节反码方式写入，如 nUser，nData 等。每次系统复位后，选项字节会重新装载选项字节信息块的数据，并做相应的判断与状态改变，这些状态保存在选项字节寄存器（FLASH_OBR 及 FLASH_WRPR）中。在信息块中每个选择位都有对应的反码位，在加载选择位时反码位用于验证选择位是否正确，如果在加载过程中发现有差别，将产生一个选项字节错误标志（OPTERR），如果开启了对应的中断，将触发中断。

选项字节块中选项字节的组织结构如下表所示（位 15~8 中的值为位 7~0 中选项字节的反码）：

表 4-3 选项字节组织结构

地址	[15: 8]	[7: 0]	默认值
0x1FFF F800	nRDP	RDP	0x5AA5
0x1FFF F802	nUSER	USER	0xFFFF
0x1FFF F804	nData0	Data0	0xFFFF
0x1FFF F806	nData1	Data1	0xFFFF
0x1FFF F808	nWRP0	WRP0	0xFFFF
0x1FFF F80A	nWRP1	WRP1	0xFFFF

表 4-4 USER 的位含义

	Bit	Field	Type	Default	Description	FLASH_OBR
RDP	7: 0	RDP	rw	0xA5	0xA5	FLASH_OBR. Bit1
nRDP	15: 8	nRDP	rw	0x5A	0x5A	
User Byte	0	WDG_SW	rw	0x01	0: 硬件看门狗 1: 软件看门狗	FLASH_OBR. Bit2
	1	nRST_STOP	rw	0x01	0: 当进入停机（STOP）模式时产生复位 1: 进入停机（STOP）模式时不产生复位	FLASH_OBR. Bit3
	2	Reserved	rw	0x01	保留为 0x01	保留
	3	Reserved	rw	0x01	保留为 0x01	保留

	Bit	Field	Type	Default	Description	FLASH_OBR
	4	nBOOT1	rw	0x01	0: nBOOT1=0 1: nBOOT1=1	FLASH_OBR. Bit6
	5	OBR_nRST	rw	0x01	NRST 复用 GPIO 1: NRST 功能 0: GPIO 功能	FLASH_OBR. Bit7
	6	nBOOT0	rw	0x01	BOOT 选项, 见芯片配置章节	FLASH_OBR. Bit8
	7	BOOT0SEL	rw	0x01	BOOT 选项, 见芯片配置章节	FLASH_OBR. Bit9
DATA0 Byte	0	DATA0.Bit0	rw	0x01	用户自定义	FLASH_OBR. Bit10
	1	DATA0.Bit1	rw	0x01	用户自定义	FLASH_OBR. Bit11
	2	DATA0.Bit2	rw	0x01	用户自定义	FLASH_OBR. Bit12
	3	DATA0.Bit3	rw	0x01	用户自定义	FLASH_OBR. Bit13
	4	DATA0.Bit4	rw	0x01	用户自定义	FLASH_OBR. Bit14
	5	DATA0.Bit5	rw	0x01	用户自定义	FLASH_OBR. Bit15
	6	DATA0.Bit6	rw	0x01	用户自定义	FLASH_OBR. Bit16
	7	DATA0.Bit7	rw	0x01	用户自定义	FLASH_OBR. Bit17
DATA1 Byte	0	DATA1.Bit0	rw	0x01	用户自定义	FLASH_OBR. Bit18
	1	DATA1.Bit1	rw	0x01	用户自定义	FLASH_OBR. Bit19
	2	DATA1.Bit2	rw	0x01	用户自定义	FLASH_OBR. Bit20
	3	DATA1.Bit3	rw	0x01	用户自定义	FLASH_OBR. Bit21
	4	DATA1.Bit4	rw	0x01	用户自定义	FLASH_OBR. Bit22
	5	DATA1.Bit5	rw	0x01	用户自定义	FLASH_OBR. Bit23
	6	DATA1.Bit6	rw	0x01	用户自定义	FLASH_OBR. Bit24
	7	DATA1.Bit7	rw	0x01	用户自定义	FLASH_OBR. Bit25

注意：在写保护值中，一个比特位对应四页，即 4096 Bytes。

4.2.3 Data 空间说明

在 Data 空间中，可以用来存储非易失数据。Data 空间的擦除、编程操作流程与选项字节区完全一致。Data 空间除地址与主闪存块不连续外，读取/取指行为同主闪存块完全相同。

4.3 闪存操作与流程

4.3.1 闪存读操作

用户代码和数据存储于主存储块中，闪存控制器可以按照 8bit/16bit/32bit 位读取数据或指令。主闪存模块与普通外设一样统一寻址访问。基于读保护与写保护的要求，任何对主存储块的内容的读写操作都须经过特定的判断过程，以防止非法读取与写入。

闪存按 Flash 访问控制寄存器（FLASH_ACR）中的设定的方式，通过 AHB 总线执行取指令和取数据。结合 AHB 时钟，设定相应的访问时延（Latency），使能预取值缓冲区后，可提高 CPU 的取指令速度，从而提高 CPU 的运行速度。访问时延（Latency）在 SYSCLK 低于等于 24MHz，可以设定为 0，此后每增加 24MHz，需要增加一个时延。

上电复位后，闪存控制器默认设定预取指缓冲区是关闭的。如需要关闭或重新打开预取指缓冲功能，必须设定 SYSCLK 低于 24MHz，并且 AHB 时钟没有经过任何分频的条件下（SYSCLK 必须等于 HCLK）才可以关闭或重新打开预取指缓冲功能。

为了保护对 Flash 的正确读取，必须在 Flash 访问控制寄存器中的 LATENCY[2: 0] 中指定预取指控制器的速度比，这个数值等于每次访问 Flash 后到下次访问之间所需插入的等待周期的个数。复位后，这个值默认为零，也就是没有插入等待周期的状态，相应的系统时钟也复位为使用内置时钟 HSI=8MHz。复位后如果需要修改系统时钟，必须先配置好安全的 LATENCY[2: 0] 值，而当 AHB 时钟的预分频器大于 1 时，预取指缓冲区也需设定相应的访问时延（Latency）。

表 4-5 Latency 设置关系

SYSCLK	AHB DIV	Latency
0MHz < SYSCLK <=24MHz	1	0
24MHz < SYSCLK <= 48MHz	1	1
48MHz < SYSCLK <= 72MHz	1	2

4.3.2 闪存编程方式与操作流程

嵌入式闪存支持如下三种编程方式。

表 4-6 编程方式

编程方式	编程说明
在电路编程（ICP）	ICP 是指通过特定烧写器，利用 SWD 接口，改变 Flash 的内容，将用户代码烧录到 MCU 中。
在系统编程（ISP）	ISP 是指通过 ISP Firmware，结合指定的 UART 接口，改变 Flash 的内容，将用户代码烧录到 MCU 中。
在应用编程（IAP）	与 ICP 和 ISP 的方法不同的是，IAP（在应用编程）能够使用 MCU 支持的任何通信接口（UART, I2C, SPI, CAN, USB 等）下载程序或者数据。IAP 允许用户在运行程序的过程中重写应用程序，前提是一部分应用程序必须预先用 ICP 或 ISP 的方法烧写进去。

烧写和擦除操作在整个产品工作电压范围内都可以完成，在对 Flash 空间做写操作或擦除操作时，内部振荡器（HSI）必须处于开启状态，还需确保 AHB 时钟大于等于 8MHz。

只要 CPU 不去访问 Flash 空间，进行中的 Flash 写操作不会妨碍 CPU 的运行（从 RAM 或 ISP 中运行）。在对 Flash 进行写操作或擦除操作时，对 Flash 的读访问都会遇到总线停顿，直到写操作或擦除操作完成后才会继续执行，因此在写操作或擦除 Flash 时，不可以对它取指和访问数据。

闪存的编程操作由一系列的动作组合而成，主要包括：

- 对 Flash 操作的解锁与保护
- 对 Flash 擦除（页擦除与整片擦除）
- 对 Flash 编程（半字编程）

- 对信息块中各空间（如选项字节）操作的解锁与保护
- 对信息块中各空间（如选项字节）擦除
- 对信息块中各空间（如选项字节）编程（半字编程）

ISP、IAP 方式编程流程

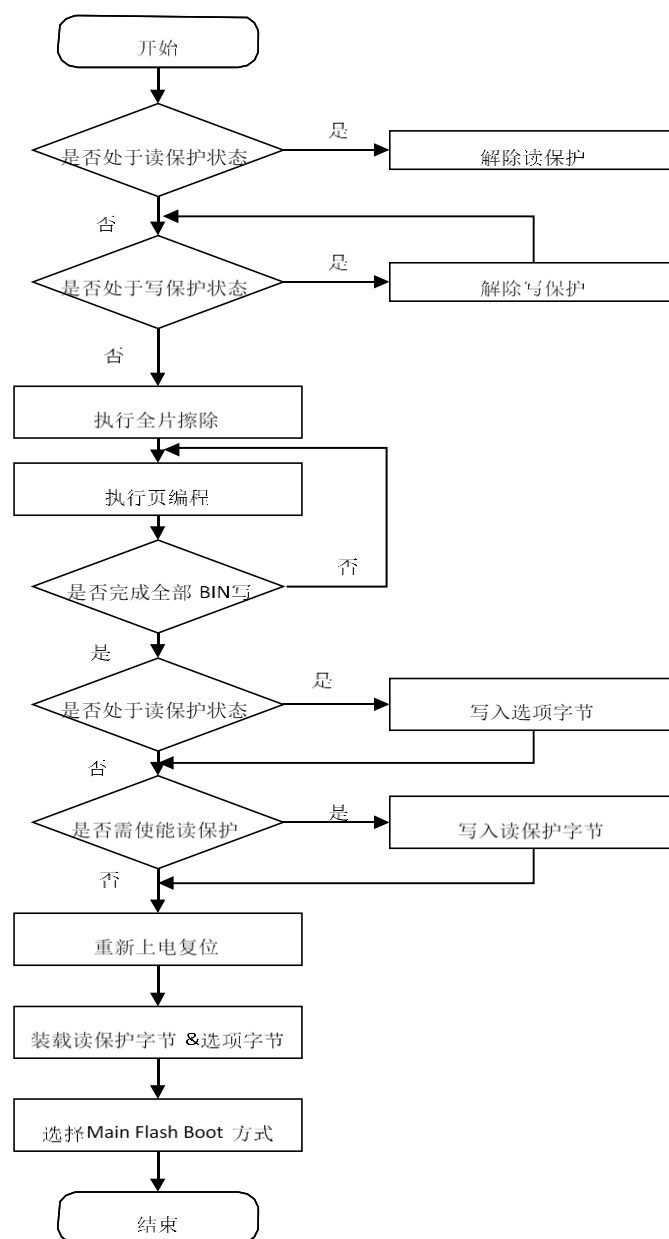


图 4-1 ISP 方式编程流程图

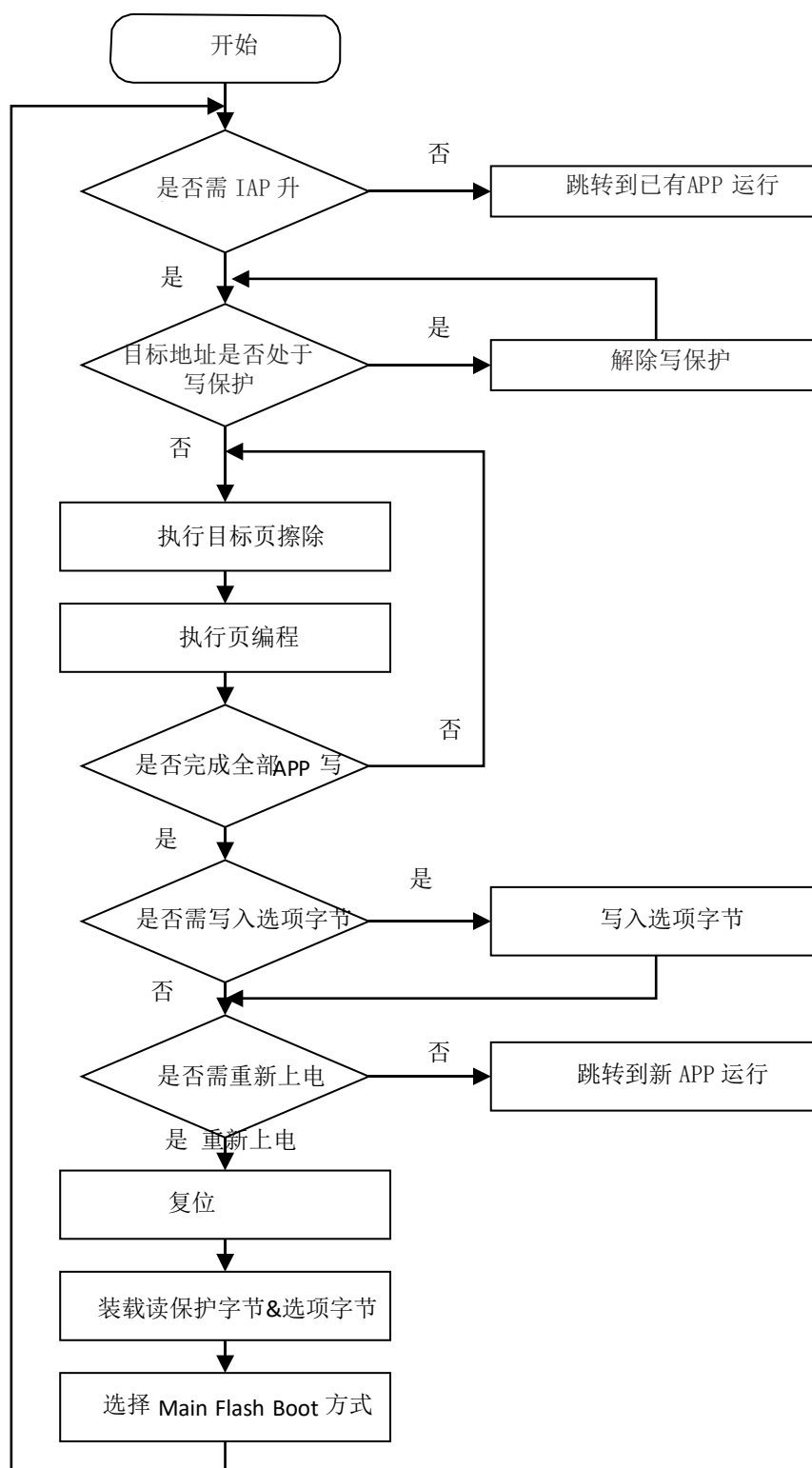


图 4-2 IAP 编程流程

4.3.3 对闪存块操作限制的解除与使能

嵌入式闪存存在复位后是处于受保护状态的，可避免意外的页擦除、全片擦除和写值等破坏 Flash 存

存储空间的操作。复位后，FLASH_CR 寄存器进入锁定状态，FLASH_CR 的 LOCK 位被控制器模块置为 1。只有通过先后向 FLASH_KEYR 寄存器写入 0x45670123 和 0xCDEF89AB 做解锁操作后，FLASH_CR 的 LOCK 位置为 0，才能开启对 FLASH_CR 的访问权限，否则 FLASH_CR 寄存器不允许被改写。

可以通过软件设置 FLASH_CR 的 LOCK 位置为 1 再次锁定，使 Flash 存储器处于受保护状态。
解除保护操作代码：

```
#define FLASH_KEY1      ((unsigned int)0x45670123)
#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)
void FLASH_Unlock()
{
    FLASH->KEYR = ((unsigned int)0x45670123);
    FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

不符合上述顺序的操作与写入错误的值，将会锁死 FLASH_CR，并引发一个总线错误，直至下次复位。

使能保护操作代码：

```
#define FLASH_CR_LOCK_Pos      (7)
#define FLASH_CR_LOCK          (0x01U << FLASH_CR_LOCK_Pos)
void FLASH_Lock(void)
{
    FLASH->CR |= FLASH_CR_LOCK;
}
```

4.3.4 对选项字节区块操作限制的解除与使能

闪存控制器在复位后，它的选项字节区块默认是处于写保护的，并且任何时候都是可读的。同样是为了避免对选项字节区做块擦除和写值等破坏性操作，复位后，FLASH_CR 寄存器进入锁定状态，FLASH_CR 的 LOCK 位被控制器模块置为 1，而 OPTWRE 位被控制器模块清除为 0；因此需先后向 FLASH_KEYR 寄存器写入 0x45670123 和 0xCDEF89AB 做解锁 FLASH 操作，FLASH_CR 的 LOCK 位置为 0 后，才做选项字节区的解锁。通过向 FLASH_OPT_KEYR 寄存器先后写入 0x45670123 和 0xCDEF89AB，从而使硬件将 FLASH_CR 寄存器的 OPTWRE 位置 1，才能对选项字节区执行块擦除，半字编程操作。可将 FLASH_CR 寄存器的 OPTWRE 位置 0，从而禁止对选项字节区执行块擦除，半字编程操作。

表 4-7 保护设置的状态变化

设置与状态	主闪存块	信息块	说明
上电复位 闪存控制器状态为	保护	保护	使能对主闪存块的操作保护 使能对选项字节区的操作保护

设置与状态	主闪存块	信息块	说明
FLASH_CR.LOCK=1 FLASH_CR.OPTWRE=0			
设置 FLASH_KEYR=0x45670123 FLASH_KEYR=0xCDEF89AB 闪存控制器状态变为 FLASH_CR.LOCK=0 FLASH_CR.OPTWRE=0	解除保护	保护	解除对主闪存块的操作保护，可对主闪存块执行全片擦除，页擦除，半字编程 还保持使能对选项字节区的操作保护，不能对选项字节区执行块擦除，半字编程操作
FLASH_KEYR=0x45670123 FLASH_KEYR=0xCDEF89AB FLASH_OTPKEYR=0x45670123 FLASH_OTPKEYR=0xCDEF89AB 闪存控制器状态变为 FLASH_CR.LOCK=0 FLASH_CR.OPTWRE=1	解除保护	解除保护	解除对主闪存块的操作保护，可对主闪存块执行全片擦除，页擦除，半字编程 解除对选项字节区的操作保护，可对选项字节区执行块擦除，半字编程操作
设置 FLASH_CR.OPTWRE=0 保持 FLASH_CR.LOCK=0	解除保护	使能保护	仍处于解除对主闪存块的操作保护，使能对选项字节区的操作保护
设置 FLASH_CR.OPTWRE=0 设置 FLASH_CR.LOCK=1	使能保护	使能保护	使能了对主闪存块的操作保护，使能对选项字节区的操作保护

解除保护操作代码：

```
#define FLASH_KEY1      ((unsigned int)0x45670123)

#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)

void FLASH_Unlock(void)
{
    FLASH->KEYR = ((unsigned int)0x45670123);
    FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

使能保护操作代码：

```
#define FLASH_CR_LOCK_Pos      (7)

#define FLASH_CR_LOCK          (0x01U << FLASH_CR_LOCK_Pos)

void FLASH_Lock(void)
{
    FLASH->CR |= FLASH_CR_LOCK;
}
```

解除选项字节区保护操作代码：

```
#define FLASH_KEY1      ((unsigned int)0x45670123)

#define FLASH_KEY2      ((unsigned int)0xCDEF89AB)

void FLASH_OPT_Unlock (void)
{
    FLASH->OPTKEYR = FLASH_KEY1;
    FLASH->OPTKEYR = FLASH_KEY2;
}
```

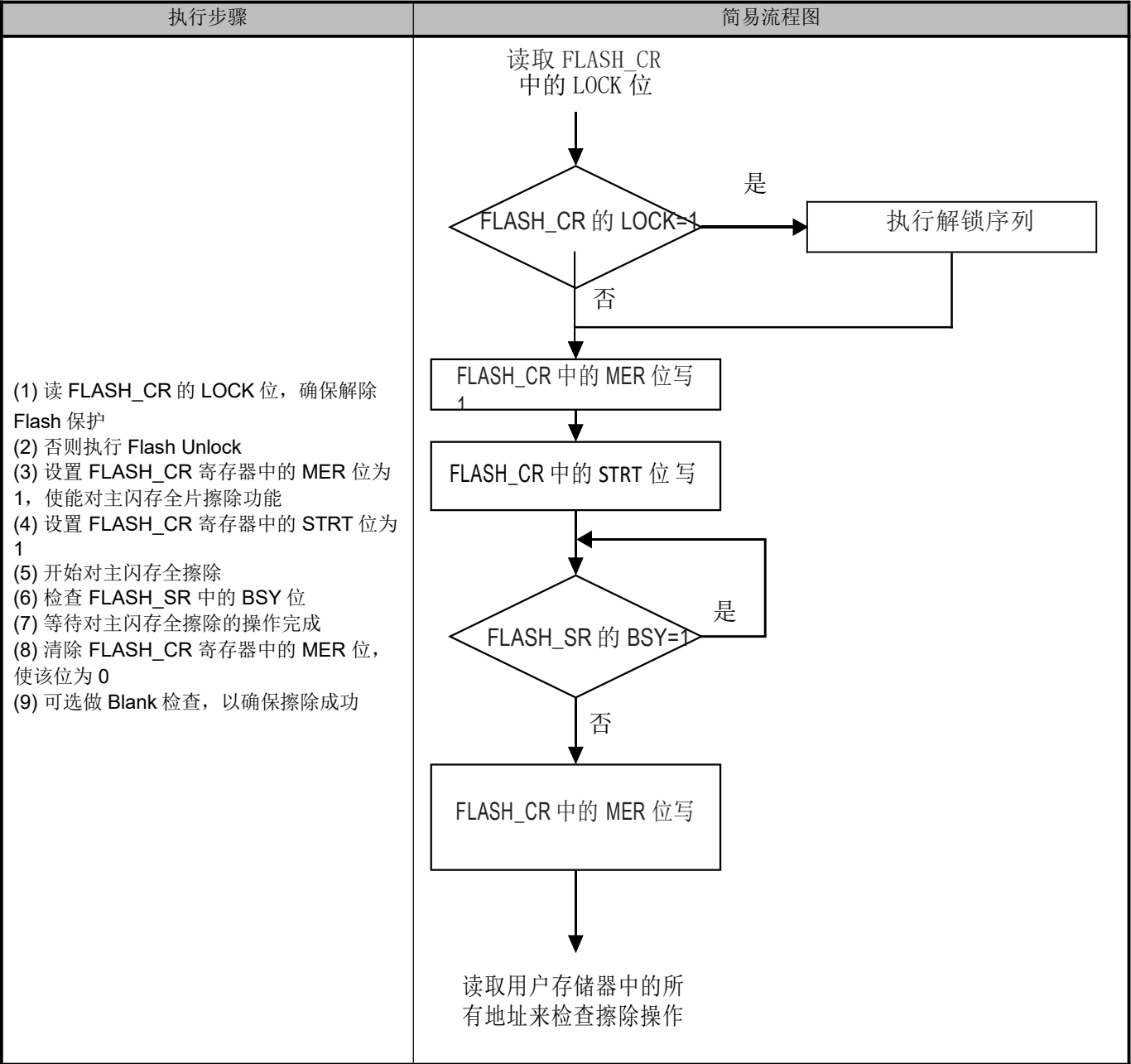
```
}  
对选项字节区保护使能操作代码：  
#define FLASH_CR_OPTWRE_Pos    (9)  
#define FLASH_CR_OPTWRE        (0x01U << FLASH_CR_OPTWRE_Pos)  
void FLASH_OPT_Lock(void)  
{  
    FLASH->CR &= ~FLASH_CR_OPTWRE;  
}
```

4.3.5 主闪存块擦除

闪存控制器支持整片擦除主闪存块和以页为单位擦除主闪存中的页。 整片擦除功能将初始化主闪存块的所有内容，使所有的值为 0xFFFF ，但信息块不会受这个命令影响。

整片擦除操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
------	-------



页擦除操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
------	-------

执行步骤	简易流程图
<div>(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护，否则执行 Flash Unlock</div> <div>(2) 设置 FLASH_CR 寄存器中的 PER 位为 1，使能页擦除功能</div> <div>(3) 把待擦除页基地址写入 FLASH_AR 寄存器</div> <div>(4) 设置 FLASH_CR 寄存器中的 STRT 位为 1，开始页擦除</div> <div>(5) 检查 FLASH_SR 中的 BSY 位</div> <div>(6) 等待页擦除的操作完成</div> <div>(7) 清除 FLASH_CR 寄存器中的 PER 位，使该位为 0</div> <div>(8) 可选对这一页的内容做检查，以确保页擦除成功</div> <div>(9) 连续多页擦除，可以重复 2~7 操作；直到完成所有页面擦除</div>	<pre>graph TD Start([开始]) --> ReadLock[读 FLASH_CR 的 LOCK 位] ReadLock --> IsLock{FLASH_CR 的 LOCK=1} IsLock -- 是 --> Unlock[执行解锁序列] IsLock -- 否 --> WritePER[FLASH_CR 中的 PER 位写] WritePER --> WriteAddr[将要擦除的页面地址写入 FLASH_AR] WriteAddr --> WriteSTRT[将 FLASH_CR 中的 STRT 位写入 1] WriteSTRT --> IsBSY{FLASH_SR 中的 BSY=1} IsBSY -- 是 --> IsBSY IsBSY -- 否 --> WritePER0[FLASH_CR 中的 PER 位写 0] WritePER0 --> ReadBack[读取页面中的所有地址 检查页面是否被擦除] ReadBack --> End([结束])</pre>

4.3.6 主闪存块编程

主闪存只支持以 16 位半字编程，用来修改主存储闪存块内容。如果以 32 位整字或 8 位字节的长度编程，将引起硬件错误中断。当 FLASH_CR 中的 PG 位为 1 时，直接对相应的地址写一个半字（16 位），就是一次编程操作。

主闪存控制器会预读待编程半字是否为全 1（即是否为 0xFFFF），如果不是，这次编程操作会自动取消，并且在 FLASH_SR 寄存器的 PGERR 位上提示编程错误警告。

如果待编程地址所对应的写保护块在 FLASH_WRPR 中的写保护位有效，同样也不会有编程动作，同样也会产生编程错误警告，编程动作结束后，FLASH_SR 寄存器中得 EOP 位会给出提示。

注意：当 CPU 进入省电模式时，通过 SWD 接口，对闪存操作将产生错误。避免在主闪存中运行中断程序时进行擦除或编程操作。

主闪存块编程操作的寄存器设置，具体步骤如下：

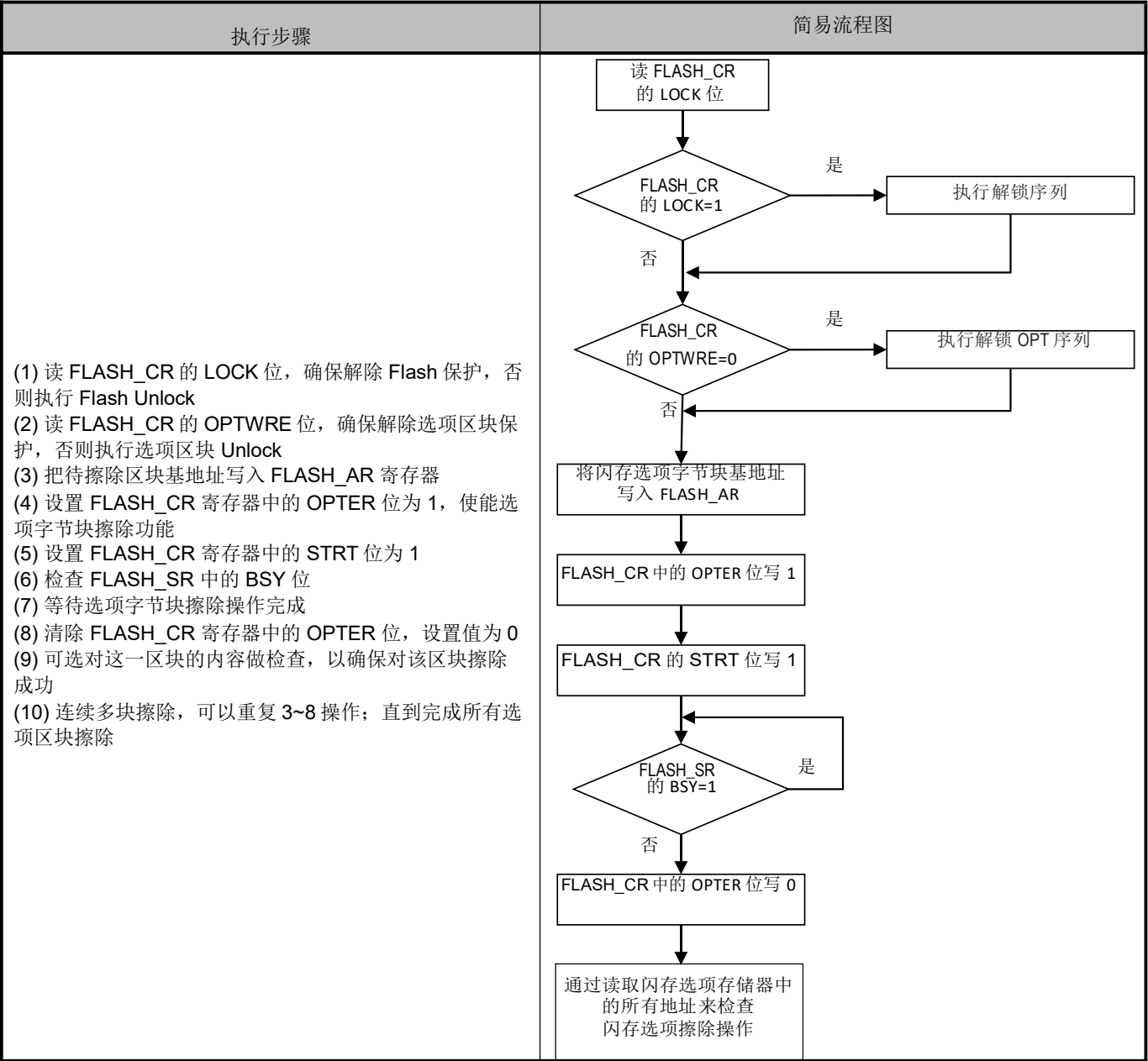
执行步骤	简易流程图
------	-------

执行步骤	简易流程图
<div>(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护，否则执行 Flash Unlock</div> <div>使能半字编程功能</div> <div>(3) 以半字为单位向目标地址写入数据，目标地址需以半字对齐</div> <div>(4) 检查 FLASH_SR 中的 BSY 位</div> <div>(5) 等待半字编程操作完成</div> <div>(6) 清除 FLASH_CR 寄存器中的 PG 位，使该位为 0</div> <div>(7) 可选读目标地址数据，以确保半字编程成功</div> <div>(8) 连续多个半字编程，可以重复 2~6 操作；直到完成所有目标地址的半字编程</div>	<pre>graph TD; A[读 FLASH_CR 的 LOCK 位] --> B{FLASH_CR 的 LOCK=1}; B -- 是 --> C[执行解锁序列]; B -- 否 --> D[FLASH_CR 的 PG 位写 1]; C --> D; D --> E[在目标地址执行半字写入]; E --> F{FLASH_SR 的 BSY=1}; F -- 是 --> F; F -- 否 --> G[FLASH_CR 的 PG 位写 0]; G --> H[读取编程地址来检查编程值];</pre>

4.3.7 选项字节区块擦除

选项字节区块擦除操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
------	-------

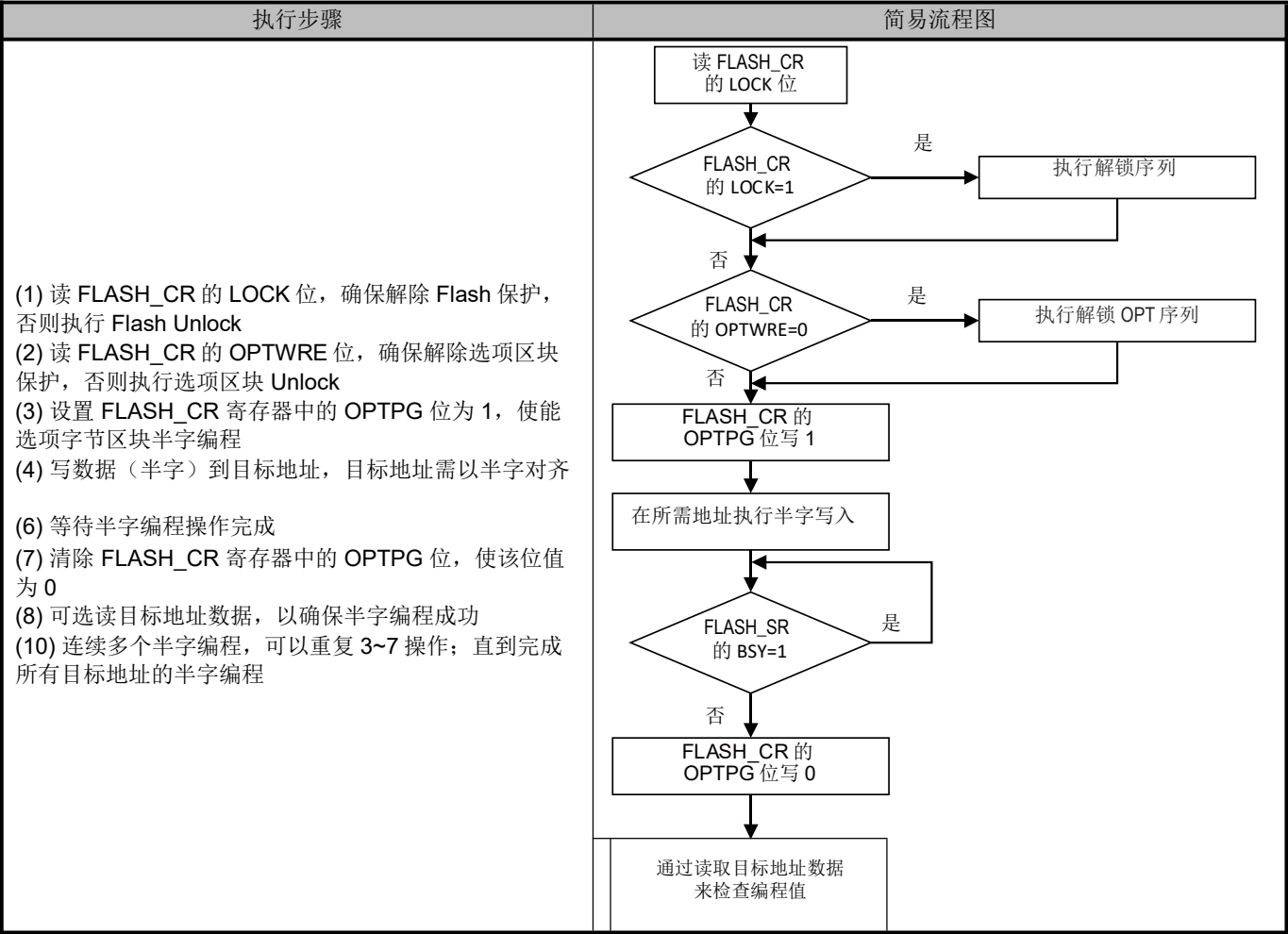


4.3.8 选项字节区块编程

选项字节区块的编程与主闪存块地址的编程不同，因其写入值复位后加载到配置选项，需要更加严格的保护。解除对闪存控制器的访问限制后，还需要对选项字节区块解除访问限制。完成该操作后，FLASH_CR 寄存器中的 OPTWRE 位会被置 1，才能允许后续的编程操作。

选项字节有效数据为低 8 位，而高 8 位为低 8 位的反码，从而组成为 16 位数据。在编程过程中，软件将高 8 位设置为低 8 位的反码，保证选项字节的写入值总是对的，然后依次写入 16 位数据。当选项字节被改变时，需要系统上电复位使之生效。选项字节区块半字编程操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
------	-------



4.3.9 闪存保护

主闪存块使能读写保护可以防范主闪存块的代码被不可信的代码读出，也可以防范在程序跑飞的时候对主闪存块的意外擦除与编程。使能读保护的范围是整个主闪存块，而使能写保护的最小单位是一个写保护块（即 4 页）。

4.3.9.1 主闪存块读保护

主闪存块使能或解除读保护是通过从内置 SRAM 或 ICP、ISP 方式设置 RDP 半字，然后系统重新上电复位，加载了新的 RDPs 后起作用的。设置读保护，需要执行一次上电复位，而不是系统复位，才能起作用。

使能读保护

按选项字节区块半字编程的操作方式，按顺序写 RDP 半字到对应地址

- 设置 FLASH AR 地址值为 0x1FFFF800，执行该选项区块擦除。
- 按选项字节区块半字编程的操作方式，按顺序写 0x807F 半字到对应地址。
- 进行上电复位以重新加载选项字节，此时读保护被使能。

当 RDP 字包含下列数值时，且被重新上电复位后主闪存块被置于保护状态。

表 4-8 Flash 读保护状态

使能读保护操作	读保护状态
对 0x1FFFF800 选项区块擦除 写 0x807F 半字到对应地址 0x1FFFF800 重新上电复位，读保护被使能	保护

当读保护半字被写入相应的值以后：

- 1. 只允许从用户代码执行对主闪存存储器的读操作（以非调试方式从主闪存存储器启动）。
- 2. 读保护后，调试模式下（SRAM boot 和 debug 模式）禁止对 Flash 进行操作。
- 3. MCU 可以通过在主闪存存储器中执行的代码进行编程（实现 IAP 或数据存储等功能），但不允许在调试模式下或从内部 SRAM 启动后执行主闪存块写或页擦除操作（整片擦除除外）。
- 4. 所有通过 SWD 向内置 SRAM 装载代码并执行代码的功能依然有效，亦可以通过 SWD 从内置 SRAM 启动，这个功能可以用来解除读保护。
- 5. 通过从内置 SRAM 执行代码访问主闪存存储器的操作，通过 DMA、SWD（串行线调试）对闪存的访问都将被禁止。

解除读保护

从内置 SRAM 或 ICP 方式解除读保护的过程是：

- 1. 设置 FLASH_AR 地址值为 0x1FFFF800，执行该选项区块擦除。
- 2. 按选项字节区块半字编程的操作方式，按流程写 0x5AA5 半字到对应地址。
- 3. 设置 FLASH_AR 地址值为 0x08000000，执行主 Flash 全片擦除。
- 4. 进行上电复位以重新加载选项字节，此时读保护被解除。

表 4-9 Flash 解除读保护状态

解除读保护操作	读保护状态
对 0x1FFFF800 选项区块擦除 写 0x5AA5 半字到对应地址 0x1FFFF800 对 0x08000000 的主 Flash 全片擦除 重 新上电复位，读保护被解除	解除读保护

注：1.如选项字节块对应的地址值为非 0xFFFF，需先执行擦除选项字节块的动作，执行擦除选项字节块的动作不会导致自动的整片擦除操作，不会改变读保护状态。2.必需对 0x08000000 的主 Flash 全片擦除。

4.3.9.2 主闪存块写保护

使能写保护

写保护通过设置选项字节区块中的 WRP0 中的 WRP 位为 0，来设置写保护，系统复位后将加载新选项字节，使能写保护。如果试图写入或擦除一个受写保护的页，会引起 FLASH_SR 中的 WRPRERR 标志位置位。

表 4-10 写保护区域

地址	[15: 8]	[7: 0]	默认值	注释
0x1FFF F808	nWRP0	WRP0	0xFFFF	

解除写保护

解除写保护有下述 2 种情形：

1. 情形 1：解除写保护，同时解除读保护：

- a. 使用闪存控制寄存器（FLASH_CR）的 OPTER 位擦除整个选项字节区块；写 0x5AA5 半字到对应地址 0x1FFFF800；
- b. 对 0x08000000 的主 Flash 全片擦除；
- c. 进行系统复位，重装载选项字节（包含新的 WRP 字节），写保护被解除。

使用这种方法，将解除全片主闪存模块的写保护同时擦除全片主闪存块。

2. 情形 2：解除写保护，同时保持读保护有效，这种情况常见于用户自己实现在程序中编程的启动程序中：

- a. 使用闪存控制寄存器（FLASH_CR）的 OPTER 位擦除整个选项字节区块；
- b. 进行系统复位，重装载选项字节（包含新的 WRP 字节），写保护被解除。

使用这种方法，将解除整个主闪存模块的写保护，同时保持读保护有效。

4.4 寄存器

4.4.1 寄存器总览

表 4-11 FLASH 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	FLASH_ACR	闪存访问控制寄存器	0x00000038
0x04	FLASH_KEYR	FPEC 键寄存器	0x00000000
0x08	FLASH_OPTKEYR	闪存 OPTKEY 寄存器	0x00000000
0x0C	FLASH_SR	闪存状态寄存器	0x00000000
0x10	FLASH_CR	闪存控制寄存器	0x00000080
0x14	FLASH_AR	闪存地址寄存器	0x00000000
0x1C	FLASH_OBR	选项字节寄存器	0x03FFFC1C
0x20	FLASH_WRPR	写保护寄存器	0x0000FFFF

注意：Flash 寄存器只支持以 32 位的方式访问

4.4.2 FLASH_ACR 闪存访问控制寄存器

偏移地址：0x00 复位值：

0x0000 0038

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.										PRFTB S	PRFTB E	Res.	LATENCY		
										r	rw		rw		

Bit	Field	Description
31: 6	Res.	保留，必须保持复位值
5	PRFTBS	预取缓冲区状态（Prefetchbufferstatus） 0：预取缓冲区关闭 1：预取缓冲区开启
4	PRFTBE	预取缓冲区使能（Prefetch buffer enable） 0：关闭预取缓冲区 1：启用预取缓冲区 注释 1：只有当 LATENCY 为 0 时，通过此位才能控制预取缓冲区开关；
3	Res.	保留，必须保持复位值
2: 0	LATENCY	时延（Latency） 这些位表示 SYSCLK（系统时钟）周期与闪存访问时间的比例。 000：零等待状态，当 $0 < \text{SYSCLK} \leq 24\text{MHz}$ 001：一个等待状态，当 $24\text{MHz} < \text{SYSCLK} \leq 48\text{MHz}$ 010：两个等待状态，当 $48\text{MHz} < \text{SYSCLK} \leq 72\text{MHz}$

4.4.3 FLASH_KEYR FPEC 键寄存器

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

Bit	Field	Description
31: 0	FKEYR	FPEC 键（Flash key） 这些位用于输入 FPEC 的解锁键。

注：所有这些位是只写的，读出时返回 0。

4.4.4 FLASH_OPTKEYR 闪存 OPTKEY 寄存器

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

w

Bit	Field	Description
31: 0	OPTKEYR	选择字节键 (Option byte key) 这些位用于输入选项字节的键以解除 OPTWRE。

注：所有这些位是只写的，读出时返回 0。

4.4.5 FLASH_SR 闪存状态寄存器

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.										EOP	WRPRT ERR	Res.	PGERR	Res.	BSY
										rw	rw		rw		r

Bit	Field	Description
31:6	Res.	保留，必须保持复位值
5	EOP	操作结束 (End of operation) 当闪存操作 (编程 擦除) 完成时，硬件设置这位为“1”，写入“1”可以清除这位状态。
4	WRPRTERR	写保护错误 (Write protection error) 试图对写保护的闪存地址编程时，硬件设置这位为“1”，写入“1”可以清除这位状态。
3	Res.	保留，必须保持复位值
2	PGERR	编程错误 (Programming error) 试图对内容不是“0xFFFF”的地址编程时，硬件设置这位为“1”，写入“1”可以清除这位状态。 注：进行编程操作之前，必须先清除 FLASH_CR 寄存器的 STRT 位。
1	Res.	保留，必须保持复位值
0	BSY	忙 (Busy) 该位指示闪存操作正在进行。在闪存操作开始时，该位被置为“1”；在操作结束或发生错误时该位被清除为“0”。

4.4.6 FLASH_CR 闪存控制寄存器

偏移地址：0x10 复位值：

0x0000 0080

31	30	29	28	28	27	26	25	24	23	22	21	20	19	18	17
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						OPTWR E	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
						rw		rw	rw	rw	rw		rw	rw	rw

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31:10	Res.	保留，必须保持复位值
9	OPTWRE	允许写选项字节（Option byte write enable） 当该位为“1”时，允许对选项字节进行编程操作。当在 FLASH_OPTKEYR 寄存器写入正确的键序列后，该位被置为“1”。软件写 0 可清除此位。
8	Res.	保留，必须保持复位值
7	LOCK	锁（Lock） 只能写“1”。当该位为“1”时表示 FPEC 和 FLASH_CR 被锁住。在检测到正确的解锁序列后，硬件自动清除此位为“0”。在一次不成功的解锁操作后，下次系统复位前，该位不能再被改变。
6	STRT	开始（Start） 当该位为“1”时将触发一次擦除操作。该位只可由软件置为“1”并在 BSY 变为“1”时自动清“0”。
5	OPTER	擦除选项字节（Option byte erase） 擦除选项字节。
4	OPTPG	烧写选项字节（Option byte programming） 对选项字节编程。
3	Res.	保留，必须保持复位值
2	MER	全擦除（Mass erase） 选择擦除所有用户页。
1	PER	页擦除（Page erase） 选择擦除页。
0	PG	编程（Programming） 选择编程操作。

4.4.7 FLASH_AR 闪存地址寄存器

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

Bit	Field	Description
31: 0	FAR	用户闪存地址（Flash Address） 当进行页擦除时选择要擦除的页。注意：当 FLASH_SR 中的 BSY 位为“1”时，不能写这个寄存器。

由硬件修改为当前最后使用的地址。页擦除操作中，必须修改这个寄存器以指定要擦除的页。

4.4.8 FLASH_OBR 选项字节寄存器

偏移地址：0x1C 复位值：

0x03FF FC1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								Data1				Data0			

								r						r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						BOOT0SEL	nBOOT0	OBR_nRST	nBOOT1	Res.		nRST_STOP	WDG_SW	RDPRT	OPTERR
r						r	r	r	r			r	r	r	r

Bit	Field	Description
31: 26	Res.	保留，必须保持复位值
25: 18	Data1	Data1
17: 10	Data0	Data0
9	BOOT0SEL	BOOT0SEL
8	nBOOT0	nBOOT0
7	OBR_nRST	NRST 复用 GPIO 1: NRST 功能 0: GPIO 功能
6	nBOOT1	nBOOT1
5	Res.	保留，必须保持复位值
4	Res.	保留，必须保持复位值
3	nRST_STOP	进入停机模式时的复位事件 0: 当进入停机（STOP）模式时产生复位 1: 进入停机（STOP）模式时不产生复位
2	WDG_SW	选择看门狗事件 0: 硬件看门狗 1: 软件看门狗
1	RDPRT	读保护（Read protection level status） 当设置为“1”，表示闪存存储器被读保护。 注：该位为只读。
0	OPTERR	选项字节错误（Option byte error） 当该位为“1”时表示选项字节和它的反码不匹配。 注意：该位为只读。

4.4.9 FLASH_WRP 写保护寄存器

偏移地址：0x20 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
WRP															
r															

Bit	Field	Description
31:16	Res.	保留，必须保持复位值
15:0	WRP	写保护（Write protect） 该寄存器包含由 OBL 加载的写保护选项字节。 0: 写保护生效 1: 写保护失效

5 SYSCFG 系统控制器

5.1 简介

该芯片具有一组系统配置寄存器。这些寄存器的主要功能如下：

- 管理连接到 GPIO 口的外部中断（引脚配置）
- 重映射存储器到代码起始区域
- 部分外设的系统级配置

5.2 寄存器

5.2.1 寄存器总览

表 5-1 SYSCFG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR	SYSCFG 配置寄存器	0x0000000X
0x08	SYSCFG_EXTICR1	SYSCFG 外部中断配置寄存器 1	0x00000000
0x0C	SYSCFG_EXTICR2	SYSCFG 外部中断配置寄存器 2	0x00000000
0x10	SYSCFG_EXTICR3	SYSCFG 外部中断配置寄存器 3	0x00000000
0x14	SYSCFG_EXTICR4	SYSCFG 外部中断配置寄存器 4	0x00000000
0x18	SYSCFG_PADHYS	SYSCFG PAD 配置寄存器	0x00000000

5.2.2 SYSCFG_CFGR 配置寄存器

该寄存器专门用于配置内存起始区域映射和 DMA 请求重映射。具有两个可配置内存起始 0x0000 0000 地址存储区类型的控制位，这两个控制位可软件配置来屏蔽 BOOT 的选择。复位后，这两个控制位为实际的 BOOT 模式配置。

偏移地址：0x00

复位值：0x0000 000X（X 为实际 BOOT 模式的选择控制位）

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
TIM17_DMA_RMP	TIM16_DMA_RMP	TIM1_CH3_DMA_RMP	CSM_DMA_RMP	USART3_TX_DMA_RMP	USART2_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	Res.						MEM_MODE	
rw	rw	rw	rw	rw	rw	rw	rw							rw	

Bit	Field	Description
31: 16	Reserved	保留，始终读为 0
15	TIM17_DMA_RMP	0: TIM17 的 DMA 功能映射到 channel2 1: TIM17 的 DMA 功能重映射到 channel3
14	TIM16_DMA_RMP	0: TIM16 的 DMA 功能映射到 channel1 1: TIM16 的 DMA 功能重映射到 channel2
13	TIM1_CH3_DMA_RMP	0: 映射到 channel1 1: 重映射到 channel3
12	CSM_DMA_RMP	0: 映射到 channel1 1: 重映射到 channel3
11	USART3_TX_DMA_RMP	0: USART3_TX 的 DMA 功能映射到 channel1 1: USART3_TX 的 DMA 功能重映射到 channel2
10	USART2_RX_DMA_RMP	0: USART2_RX 的 DMA 功能映射到 channel2 1: USART2_RX 的 DMA 功能重映射到 channel3
9	USART1_TX_DMA_RMP	0: USART1_TX 的 DMA 功能映射到 channel1 1: USART1_TX 的 DMA 功能重映射到 channel3
8	ADC_DMA_RMP	0: ADC 的 DMA 功能映射到 channel1 1: ADC 的 DMA 功能重映射到 channel2
7: 2	Reserved	保留，必须保持复位值
1: 0	MEM_MODE	存储映射选择位 (Memory selection bit) 由软件设置和清除这些位。它控制存储器内部映射到地址 0x0000 0000。当复位后这些位值由 BOOT0 的引脚配置值和 nBOOT1 bit 值决定。x0: 主闪存存储器映射到 0x0000 0000 01: 系统闪存映射到 0x0000 0000 11: 嵌入式 RAM 映射到 0x0000 0000

5.2.3 SYSCFG_EXTICR1 外部中断配置寄存器 1

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
EXTI3				EXTI2				EXTI1				EXTI0			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置 (x=0...3) (EXTIx configuration) 选择 EXTIx 外部中断的输入源。 0000: PA[x]管脚 0001: PB[x] 管脚 0011: PD[x] 管脚

5.2.4 SYSCFG_EXTICR2 外部中断配置寄存器 2

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
EXTI7				EXTI6				EXTI5				EXTI4			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置（x=4...7）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000: PA[x]管脚 0001: PB[x]管脚 0011: PD[x]管脚

5.2.5 SYSCFG_EXTICR3 外部中断配置寄存器 3

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
EXTI11				EXTI10				EXTI9				EXTI8			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置（x=8...11）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000: PA[x] 管脚 0001: PB[x] 管脚

5.2.6 SYSCFG_EXTICR4 外部中断配置寄存器 4

偏移地址: 0x014 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
EXTI15				EXTI14				EXTI13				EXTI12			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值

Bit	Field	Description
15:0	EXTIx	EXTIx 配置 (x=12...15) (EXTIx configuration) 选择 EXTIx 外部中断的输入源。 0000: PA[x] 管脚 0001: PB[x] 管脚 0010: PC[x] 管脚 (x=13、14、15)

5.2.7 PAD 配置寄存器 (SYSCFG_PADHYS)

偏移地址: 0x018 复位值:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															I2C1_m ode_sel
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

Bit	Field	Description
31:17	Reserved	保留, 必须保持复位值
16	I2C1_MODE_SEL	I2C1 端口模式选择位 0: 开漏模式 1: 推挽模式 注: I2C 作为从机时不建议使用推挽模式; I2C 作为主机使用推挽模式时, 进行通信的从机无法拉低 SCL
15:0	Reserved	保留, 必须保持复位值

6 DMA直接存储器访问控制器

6.1 DMA 简介

DMA 控制器通过共享系统总线，实现无需 CPU 参与的快速自动数据传输。

DMA 控制器有 3 个通道，多个外设 DMA 请求发送到对应通道上处理。

6.2 DMA 功能框图

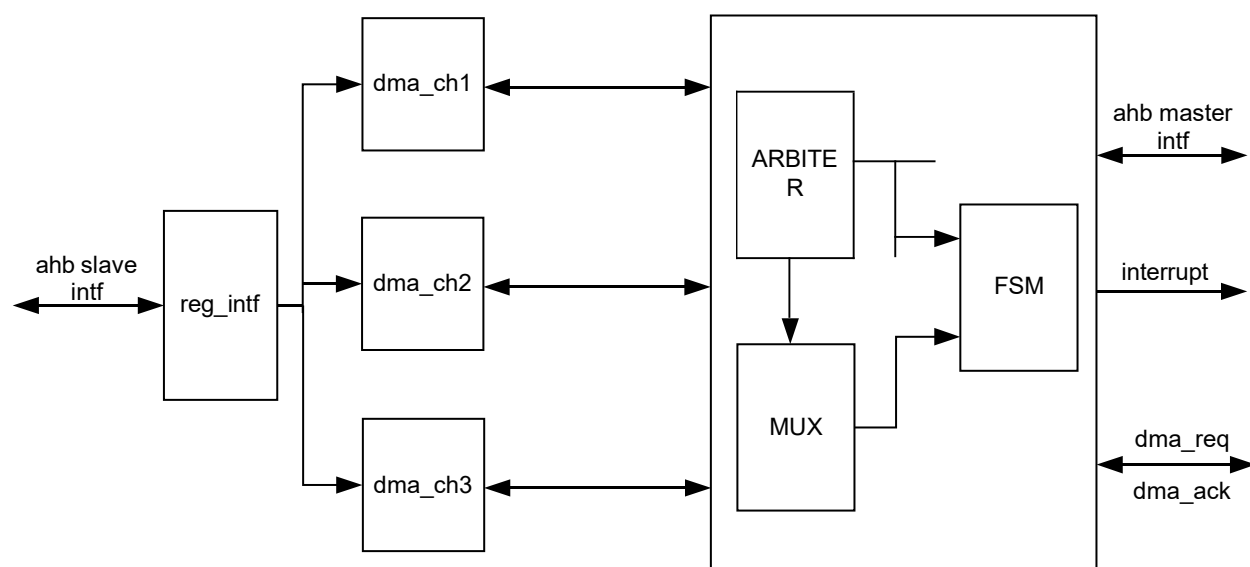


图 6-1 DMA 功能框图

6.3 DMA 主要特征

- 3 个独立的通道，可通过寄存器配置相关功能。
- 硬件发出的 DMA 请求与对应专用 DMA 通道直连。通过软件配置寄存器的方式也可以触发 DMA 通道请求。
- 可以通过软件的方式配置寄存器决定 3 个通道请求之间的处理优先级（共有四级：很高、高、中等和低），若优先级相同，则由硬件自动决定处理顺序（低编号通道请求优先处理）。
- 数据源头与目的地的传输宽度可独立配置为字节、半字、全字。
- 按照数据源的宽度配置进行打包，并在目的地按照目的地的宽度配置进行拆包。要求源和目标

地址必须根据各自配置的数据传输宽度对齐。

- 支持循环缓冲器控制。
- 每个通道支持 DMA 半传输，DMA 传输完成和 DMA 传输出错 3 种事件标志。各通道单独的中断请求由这 3 种事件标志逻辑或起来。
- 支持存储器对存储器传输。
- 支持数据传输方向为外设到存储器，存储器到外设。
- 数据访问的源和目标可以是：SRAM、APB1、APB2 和 AHB 总线上的外设。
- 数据的传输数量可以通过软件配置对应寄存器，最大值为 65535。

6.4 中断

DMA 半传输，DMA 传输完成和 DMA 传输出错为每个 DMA 通道都会产生的 3 种事件标志。各通道单独的中断请求由这 3 种事件标志逻辑或起来。

可以配置寄存器的对应位来使能这些中断，以满足程序的不同需求。

表 6-1 DMA 中断请求

中断事件	事件标志位	使能控制位
半传输	HTIF	HTIE
传输结束	TCIF	TCIE
传输出错	TEIF	TEIE

6.5 DMA

6.5.1 DMA 请求映像

从外设产生的多个传输请求，通过 DMAMAP 输入到 DMA 控制器，为了避免冲突，在一个通道中，同时只能有一个外设 DMA 请求有效。参见下图的 DMA 请求映像。

外设本身的控制寄存器有对应的 DMA 使能位，来独立控制外设是否发送传输请求。

表 6-2 DMA 各个通道的 DMA 请求一览

外设	DMA 通道 1	DMA 通道 2	DMA 通道 3
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	
SPI		SPI1_RX	SPI1_TX
USART	USART1_TX ⁽¹⁾ USART2_TX USART3_TX ⁽¹⁾	USART1_RX USART2_RX ⁽¹⁾ USART3_TX ⁽²⁾	USART1_TX ⁽²⁾ USART2_RX ⁽²⁾ USART3_RX
I2C		I2C1_TX	I2C1_RX
CSM	CSM ⁽¹⁾		CSM ⁽²⁾
TIM1	TIM1_CH1 TIM1_CH2 TIM1_CH3 ⁽¹⁾	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 ⁽²⁾ TIM1_UP TIM1_CH5
TIM3	TIM3_CH3	TIM3_CH4	TIM3_CH1

外设	DMA 通道 1	DMA 通道 2	DMA 通道 3
		TIM3_UP	TIM3_TRIG
TIM16	TIM16_CH1 ⁽¹⁾ TIM16_UP ⁽¹⁾	TIM16_CH1 ⁽²⁾ TIM16_UP ⁽²⁾	
TIM17		TIM17_CH1 ⁽¹⁾ TIM17_UP ⁽¹⁾	TIM17_CH1 ⁽²⁾ TIM17_UP ⁽²⁾

(1)如果 SYSCFG_CFGR 寄存器的对应映射位被复位，DMA 请求被映射在这个 DMA 通道。

(2)如果 SYSCFG_CFGR 寄存器的对应映射位被置位，DMA 请求被映射在这个 DMA 通道。

6.6 功能描述

DMA 与 CPU 都是通过系统总线实现对存储器或外设数据的访问。当 CPU 和 DMA 访问冲突时，DMA 请求可能会占用系统总线，此时 CPU 只能等待 DMA 传输完成释放总线。为了防止总线一直被 DMA 占用导致 CPU 无法工作，总线仲裁器会执行相关的循环调度，以此保证 CPU 至少可以获得一半的系统总线控制权。

6.6.1 DMA 处理

外设产生一个相关事件后，会将 DMA 请求信号发送到 DMA 控制器对应通道。按照软件配置的 DMA 通道优先级，或者硬件默认规则，DMA 控制器依次处理这些请求。DMA 响应外设请求，通过总线访问外设的同时，DMA 控制器会发送给外设一个应答信号，告知外设本次请求已响应。外设得到 DMA 的应答信号后，会立即释放掉本次请求。DMA 检测到外设请求消失后，对应的应答信号也会随之释放掉，本次 DMA 传输完成。

综上，每个 DMA 传送由 3 个操作组成：

- 加载源地址数据，地址由软件配置。
- 存储数据到目的地址，地址由软件配置。
- 执行一次 DMA 传输，计数器 DMA_CNDTRx 从配置的传输数量开始递减，表示剩余还有多少次 DMA 传输。

每次传输的操作顺序为：

- 1.开启 DMA 使能
- 2.开启外设 DMA 请求
- 3.待传输完毕后关闭外设 DMA 请求
- 4.重复步骤 1-3

6.6.2 仲裁器

仲裁器决定 DMA 控制器优先解决哪个 DMA 请求。优先级分软硬件 2 种逻辑控制：

- 软件：4 个等级优先级，每个通道的优先级可在 DMA_CCRx 寄存器配置：

- 最高优先级
- 高优先级
- 中等优先级
- 低优先级
- 硬件：先处理软件优先级高的请求，软件优先级配置相同则默认更低编号的通道优先。

6.6.3 DMA 通道

外设寄存器与存储器的固定地址通过 DMA 通道进行 DMA 传输。数据的传输数量可以通过软件配置对应寄存器，最大值为 65535。从配置的传输数量开始，每次传输后 DMA_CNDTRx 都会递减，指示剩余还需多少次 DMA 传输。

6.6.3.1 可编程数据宽度

配置 DMA_CCRx 寄存器中的 PSIZE 和 MSIZE 位，可以控制外设和存储器对应的传输数据宽度。

6.6.3.2 指针增量

配置 DMA_CCRx 寄存器中 PINC 和 MINC 标志位，外设与存储器的访问地址可以按照步长累加，不需要每次都去设置访问地址。

清零增量模式寄存器则每次 DMA 传输固定访问同一个地址。配置为增量模式时，下一个要传输的地址将是前一个地址加上步长，步长取决与所选的数据宽度 1

(8 位)、2 (16 位) 或 4 (32 位)。首个传输的地址存放在 DMA_CPARx / DMA_CMARx 寄存器中。

通道配置为非循环模式，DMA_CNDTRx 递减为 0 后，不会继续进行 DMA 传输。

6.6.3.3 通道配置

以下为 DMA 通道 x 的配置流程 (x 表示通道编号)：

- 操作 DMA_CPARx 寄存器，配置外设寄存器的地址。DMA 传输时该外设地址为源或目标地址取决于 DMA 传输方向。
- 操作 DMA_CMARx 寄存器，配置数据存储器的地址。DMA 传输时需要从该存储器地址加载或者存储数据取决于 DMA 传输方向。
- 操作 DMA_CNDTRx，配置 DMA 传输数量。DMA 传输完成一次，该值减 1，且在 DMA 传输期间该寄存器不可被软件写操作。
- 操作 DMA_CCRx 寄存器的 PL[1:0] 位，配置通道的优先级。
- 操作 DMA_CCRx 寄存器，配置数据传输方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度、中断产生种类。

- 操作 DMA_CCRx 寄存器的 ENABLE 位，使能这个通道。该通道使能后，就可以进行正常的 DMA 工作，响应外设请求，进行 DMA 传输。

半传输标志（HTIF）被硬件置‘1’，表示当前 DMA 传输数量为配置传输数量的一半。若想产生中断，则需使能半传输中断位（HTIE）。

传输完成标志（TCIF）被硬件置‘1’，表示当前 DMA 配置的传输数据已全部传输完毕。若想产生中断，则需使能传输完成中断位（TCIE）。

6.6.3.4 循环模式

如果需要循环读写缓冲区或者是进行连续的数据传输（如 ADC 的扫描模式），可以进入循环模式。置‘1’DMA_CCRx 寄存器中的 CIRC 位，使能循环模式。在循环模式下，DMA_CNDTRx 被递减为 0 时，会自动重新加载先前配置的数值，随后重新进行递减操作，DMA 会继续传输数据。

6.6.3.5 存储器到存储器模式

DMA 支持存储器到存储器的访问，不需要外设的参与。置‘1’DMA_CCRx 寄存器中的 MEM2MEM 位，同时置‘1’DMA_CCRx 寄存器中的通道使能位，即可开始 DMA 传输。若 DMA_CNDTRx 递减为 0，则 DMA 传输结束。

存储器到存储器的访问不支持循环模式。

6.6.4 可编程的数据传输宽度，对齐方式和数据大小端

当 PSIZE 和 MSIZE 不相同，DMA 模块按照下表进行数据对齐。 源数据比特位对齐写入目标地址， 若目标数据传输宽度大于源数据传输宽度，则目标数据宽度多余位补 0 处理。 若目标数据传输宽度小于源数据传输宽度，则源数据宽度多余部分截断处理。

表 6-3 可配置的数据传输宽度和大小端操作（当 PINC = MINC = 1），传输数目为 4

组合类型	传输宽度		传输操作	
	源端	目标	源（地址 数据）	目标（地址 数据）
源端传输宽度 等于 目标传输宽度 地址步长、数据 宽度一致	8	8	在 0x0 读 B0[7:0] 在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x0 写 B0[7:0] 在 0x1 写 B1[7:0] 在 0x2 写 B2[7:0] 在 0x3 写 B3[7:0]
	16	16	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 B1B0[15:0] 在 0x2 写 B3B2[15:0] 在 0x4 写 B5B4[15:0] 在 0x6 写 B7B6[15:0]
	32	32	在 0x0 读 B3B2B1B0[31:0] 在 0x4 读 B7B6B5B4[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B3B2B1B0[31:0] 在 0x4 写 B7B6B5B4[31:0] 在 0x8 写 BBBAB9B8[31:0] 在 0xC 写 BFBEBDBC[31:0]
源端传输宽度	8	16	在 0x0 读 B0[7:0]	在 0x0 写 00B0[15:0]

组合类型	传输宽度		传输操作	
小于 目标传输宽度 地址步长为传输 宽度 ÷ 8 目标数据多余 bit 位补 0			在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x2 写 00B1[15:0] 在 0x4 写 00B2[15:0] 在 0x6 写 00B3[15:0]
	8	32	在 0x0 读 B0[7:0] 在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x0 写 000000B0[31:0] 在 0x4 写 000000B1[31:0] 在 0x8 写 000000B2[31:0] 在 0xC 写 000000B3[31:0]
	16	32	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 0000B1B0[31:0] 在 0x4 写 0000B3B2[31:0] 在 0x8 写 0000B5B4[31:0] 在 0xC 写 0000B7B6[31:0]
源端传输宽度 大于 目标传输宽度 地址步长为传输 宽度 ÷ 8 目标数据不足 bit 位截断	16	8	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 B0[7:0] 在 0x1 写 B2[7:0] 在 0x2 写 B4[7:0] 在 0x3 写 B6[7:0]
	32	8	在 0x0 读 B3B2B1B0[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B0[7:0] 在 0x2 写 B8[7:0] 在 0x3 写 BC[7:0]
	32	16	在 0x0 读 B3B2B1B0[31:0] 在 0x4 读 B7B6B5B4[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B1B0[15:0] 在 0x2 写 B5B4[15:0] 在 0x4 写 B9B8[15:0] 在 0x6 写 BDBC[15:0]

6.6.4.1 操作一个不支持字节或半字写的 AHB 设备

AHB 总线传输通过 HSIZE 表示传输数据的宽度，目标设备不支持字节/半字操作意味着对应从设置没有处理 HSIZE 的逻辑，目标设备统一认为传输数据为一个字 32bit。

AHB 设备通常都支持字（32bit）操作，而当有的 AHB 设备不支持字节（8bit）或者半字（16bit）写操作时，DMA 会将数据处理扩展为字（32bit）。同时对应目标地址步长应变改与 32bit 对应的 4，目标数据宽度应配置为 32bit。

举例说明，存储器配置为数据源，传输宽度为 8bit，待传输数据为 0xDA，我们希望能够传输到目标设备地址的 0x2 上面，由于对应从设备不支持 8bit/16bit 操作，意味着从设备会认为自己接收的是 32bit 的 0x0000_00DA，并且写 32bit 的 0x0000_00DA 到 0x0 地址上（因为是 32bit 操作，地址单位为 0x4），也就是说 0x2 的地址对应的数据为 0x00。

因此需要对数据进行复制扩展操作，0xDA 会被复制扩展为 4 个 8bit 组合为一个 32bit 数据 0xDADA_DADA，目标地址的 0x2，从设备接收到 32bit 的 0xDADA_DADA，会写 32bit 数据 0xDADA_DADA 到 0x0 地址上。这样目的地址 0x2 上的数据即为想要的 0xDA。这种方法会写冗余的数据到目标寄存器上，但是可以保证写入目的地址上的数据是程序想要的，不会发生错误。

源数据宽度 8bit，会被复制扩充为 4 x 8bit = 32bit 数据，如 0x12 会被扩充为 0x1212_1212。源数据宽度 16bit，会被复制扩充为 2 x 16bit = 32bit 数据，如 0x1234 会被扩充为 0x1234_1234。只支持 32bit 传输，不支持 8bit / 16bit 的从设备数据宽度应配为 8bit / 16bit。

6.6.5 错误管理

地址空间会存在不允许被访问的保留区域，DMA 传输地址自动递增或者指定地址时有可能会访问到这些保留地址区域。DMA 传输错误标志（TEIF）会在 DMA 操作一个保留的地址空间时置‘1’，同时该 DMA 通道对应的使能位会被硬件清零，以停止该通道上的错误传输。此时，在 DMA_IFT 寄存器中对应该通道的传输错误中断标志位（TEIF）将被置位。若想产生中断，需配置 DMA_CCRx 寄存器中对应的传输错误中断使能位。

6.7 DMA 寄存器描述

表 6-4 DMA 寄存器概览

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

6.7.1 DMA_ISR DMA 中断状态寄存器

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.				TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31:12	Reserved	保留，始终读为 0
11,7,3	TEIFx	通道 x 的传输错误标志（x=1~3）（Channel x transfer error flag） 该位软件只读，由硬件写 1 或清 0。置‘1’DMA_IFCR 寄存器对应位，可以清 0 此标志位。 0: 对应通道 x 的 DMA 传输正常（TE） 1: 对应通道 x 的 DMA 访问保留地址，传输错误（TE）
10,6,2	HTIFx	通道 x 的半传输标志（x=1~3）（Channel x half transfer flag） 该位软件只读，由硬件写 1 或清 0。置‘1’DMA_IFCR 寄存器对应位，可以清 0 此标志位。 0: 对应通道 x 的 DMA 传输未到一半（HT） 1: 对应通道 x 的 DMA 传输已到一半（HT）

Bit	Field	Description
9,5,1	TCIFx	通道 x 的传输完成标志 (x=1~3) (Channel x transfer complete flag) 该位软件只读, 由硬件写 1 或清 0。置'1'DMA_IFCR 寄存器对应位, 可以清 0 此标志。 0: 对应通道 x 的 DMA 传输未完成 (TC) 1: 对应通道 x 的 DMA 传输完毕 (TC)
8,4,0	GIFx	通道 x 的全局中断标志 (x=1~3) (Channel x global interrupt flag) 该位软件只读, 由硬件写 1 或清 0。置'1'DMA_IFCR 寄存器对应位, 可以清 0 此标志。 0: 对应通道 x 上 TE、HT、TC 事件都没有产生 1: 对应通道 x 有 TE、HT、TC 事件中的任一事件产生

6.7.2 DMA_IFCR DMA 中断标志清除寄存器

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.				CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
				w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:12	Reserved	保留, 始终读为 0
11,7,3	CTEIFx	清除通道 x 的传输错误标志 (x=1~3) (Channel x transfer error clear) 该位由软件置'1'或清零。 0: 无效 1: 清'0'DMA_ISR 寄存器中的对应 TEIF 标志
10,6,2	CHTIFx	清除通道 x 的半传输标志 (x=1~3) (Channel x half transfer clear) 该位由软件置'1'或清零。 0: 无效 1: 清'0'DMA_ISR 寄存器中的对应 HTIF 标志
9,5,1	CTCIFx	清除通道 x 的传输完成标志 (x=1~3) (Channel x transfer complete clear) 该位由软件置'1'或清零。 0: 无效 1: 清'0'DMA_ISR 寄存器中的对应 TCIF 标志
8,4,0	CGIFx	清除通道 x 的全局中断标志 (x=1~3) (Channel x global interrupt clear) 该位由软件置'1'或清零。 0: 无效 1: 清'0'DMA_ISR 寄存器中的对应的 GIF、TEIF、HTIF 和 TCIF 标志

6.7.3 DMA_CCRx DMA 通道 x 配置寄存器 (x=1~3)

偏移地址: 0x08+20x (通道编号-1) 复位值:

0x0000 0000

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

ARE	MEM2MEM	PL	MSIZE	PSIZE	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	Reserved	保留，始终读为 0
15	ARE	自动重载（Auto reload） 该位由软件置‘1’或清零。 1:使能自动重载传输数量，外设地址，存储器地址 0:禁止自动重载传输数量，外设地址，存储器地址 注：当 ARE 配置为 1，DMA 通道使能关闭后就会触发自动重载
14	MEM2MEM	存储器到存储器模式（Memory to memory mode） 该位由软件置‘1’或清零。 0:关闭存储器到存储器模式 1:使能存储器到存储器模式
13:12	PL	通道优先级（Channel priority level） 该位由软件置‘1’或清零。 00:低 01:中 10:高 11:最高
11:10	MSIZE	存储器数据宽度（Memory size） 该位由软件置‘1’或清零。 00:8 bit 01:16 bit 10:32 bit 11:保留，未定义
9:8	PSIZE	外设数据宽度（Peripheral size） 该位由软件置‘1’或清零。 00:8 bit 01:16 bit 10:32 bit 11:保留，未定义
7	MINC	存储器地址递增模式（Memory increment mode） 该位由软件置‘1’或清零。 0:关闭存储器地址递增操作 1:使能存储器地址递增操作
6	PINC	外设地址递增模式（Peripheral increment mode） 该位由软件置‘1’或清零。 0:关闭外设地址递增操作 1:使能外设地址递增操作
5	CIRC	循环模式（Circular mode） 该位由软件置‘1’或清零。 0:关闭循环操作 1:使能循环操作
4	DIR	数据传输方向（Data transfer direction） 该位由软件置‘1’或清零。 0:从外设读 1:从存储器读
3	TEIE	传输错误中断使能（Transfer error interrupt enable） 该位由软件置‘1’或清零。 0:关闭 TE 中断 1:使能 TE 中断
2	HTIE	半传输中断使能（Half transfer interrupt enable） 该位由软件置‘1’或清零。 0:关闭 HT 中断 1:使能 HT 中断

Bit	Field	Description
1	TCIE	传输完成中断使能（Transfer complete interrupt enable） 该位由软件置‘1’或清零。 0:关闭 TC 中断 1:使能 TC 中断
0	EN	通道使能（Channel enable） 该位由软件置‘1’或清零。 0:通道关闭 1:通道使能 注：EN 位清零并不能使得此通道正在进行的 DMA 传输停止。如需停止传输，需要参照特定软件操作流程。

6.7.4 DMA_CNDTRx DMA 通道 x 传输数量寄存器（x=1~3）

偏移地址：0x0C+20x（通道编号-1） 复位值：

0x0000 0000

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

Bit	Field	Description
31:16	Reserved	保留，始终读为 0
15:0	NDT	数据传输数量（Number of data to transfer） 数据传输数量为 0~65535。这个寄存器只能在通道关闭（DMA_CCRx 的 EN=0）时写入。通道使能后该寄存器软件变为不可写只读，表示剩余多少次 DMA 传输。每次 DMA 传输后，该寄存器数值递减。寄存器数值递减为 0，表示数据全部传输完毕。此时若通道配置为自动重加载模式时，寄存器的内容将被自动重新加载为之前配置时的数值。 与通道是否使能无关，只要该寄存器为 0，DMA 就不会传输数据。

6.7.5 DMA_CPARx DMA 通道 x 外设地址寄存器（x=1~3）

偏移地址：0x10+20x（通道编号-1） 复位值：

0x0000 0000

当使能通道（DMA_CCRx 的 EN=1）时不能写该寄存器。

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

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31:0	PA	外设地址（Peripheral address） 外设数据寄存器的基地址，作为数据传输的源或目标。 当 PSIZE='01'（16 位），地址基本单位为 0x2，最低位 PA[0]不必使用。操作自动地与半字地址对齐。 当 PSIZE='10'（32 位），地址基本单位为 0x4，倒数 2 位 PA[1:0]不必使用。操作自动地与字地址对齐。

6.7.6 DMA_CMARx DMA 通道 x 存储器地址寄存器（x = 1~3）

偏移地址：0x14+20x（通道编号-1） 复位值：

0x0000 0000

当开启通道（DMA_CCRx 的 EN=1）时不能写该寄存器。

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

Bit	Field	Description
31:0	MA	存储器地址（Memory address） 存储器地址作为数据传输的源或目标。 当 MSIZE='01'（16 位），地址基本单位为 0x2，最低位 MA[0] 不必使用。操作自动地与半字地址对齐。 当 MSIZE='10'（32 位），地址基本单位为 0x4，倒数 2 位 MA[1:0]不必使用。操作自动地与字地址对齐。

7 GPIO 通用端口

7.1 简介

每个通用 I/O 端口都可以通过两个 32 位的控制寄存器（GPIOx_CRL/GPIOx_CRH）和两个 32 位的复用控制寄存器（GPIOx_AFRL/GPIOx_AFRH）配置为 8 种模式：模拟输入、浮空输入、上拉输入、下拉输入、推挽输出、开漏输出、复用推挽输出和复用开漏输出。

可以自由编程控制每个 I/O 端口，支持字（32 位）、半字（16 位）或字节（8 位）访问所有寄存器。GPIO 寄存器组有 GPIOx_BSRR 和 GPIOx_BRR 位控制寄存器，通过写操作这两个寄存器可以独立的按位控制 GPIOx_ODR 输出 0 或 1。

7.2 主要特征

- 每次 AHB 的写操作，可以更改 GPIOx_ODR 对应的一位或多位
- 所有 I/O 支持编程 EXTI 配置寄存器输出外部触发中断
- 支持配置 GPIO 锁定机制
- 输入支持浮空、上拉、下拉、模拟
- 输出支持推挽与开漏上拉或开漏下拉
- 默认浮空输入，输入输出方向可配
- I/O 输出速度可配

7.3 功能描述

7.3.1 功能框图

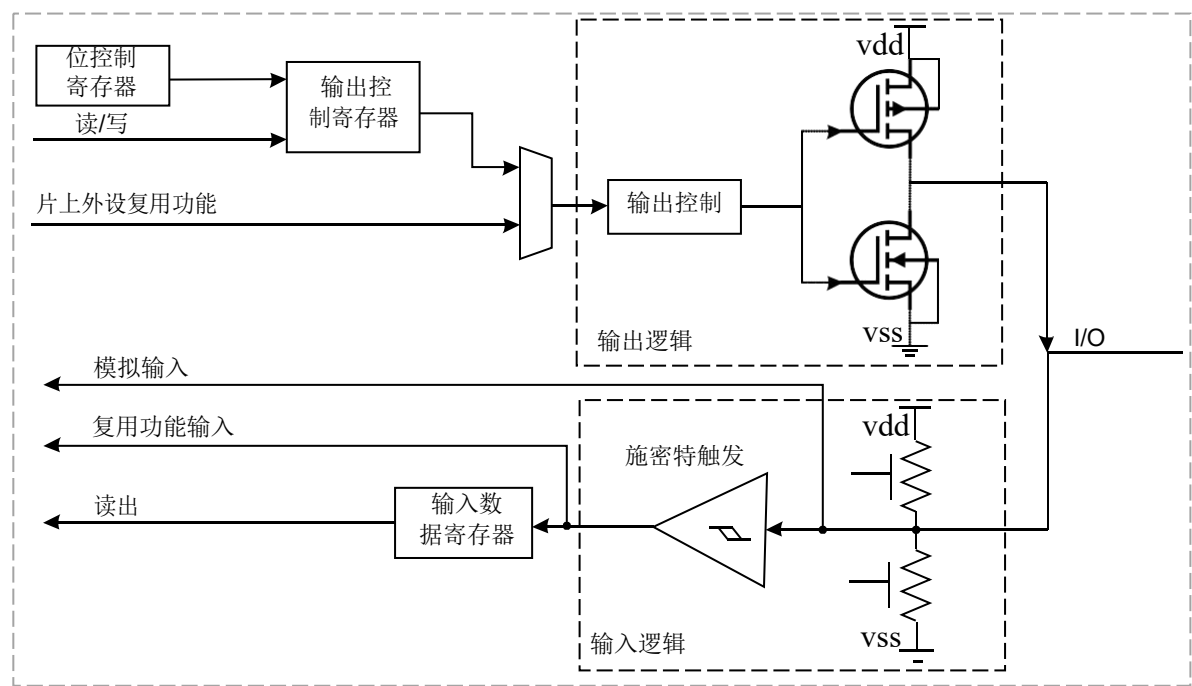


图 7-1 标准 I/O 端口

7.3.2 GPIO 端口配置

表 7-1 端口位配置表（port0 为例）

引脚模式		上下拉	DCR[1:0]		CNF0		MODE0	ODRx
模拟输入		x	x	x	0	0	00	x
通用输入		浮空	x	x	0	1		x
复用输入		上拉	x	x	1	0		1
		下拉	x	x	1	0		0
通用输出	推挽	x	x	x	0	0	01 10 11	0 or 1
	开漏	浮空	x	0	0	1		0 or 1
		上拉	1	1	0	1		0 or 1
		下拉	0	1	0	1		0 or 1
复用输出	推挽	x	x	x	1	0		x
	开漏	浮空	x	0	1	1		x
		上拉	1	1	1	1		x
		下拉	0	1	1	1		x

注：x 表示 I/O 在对应的模式下不用关心，ODR0 代表输出数据寄存器第 0 位。 输入输出参考配置如下：

- 通用输入：
用户只需配置 GPIOx_CRL 中的 CNF0 选择输入模式

- 通用输出：

推挽输出：用户配置 MODE0 选择输出速度，配置 CNF0=00；

开漏输出：用户配置 MODE0 选择输出速度，配置 CNF0=01，如果对 pin 上下拉有要求，需要单独配置 GPIOx_DCR 寄存器，非开漏输出模式，上下拉失效。

- 复用功能：

配置 AFRLx[3:0]与 AFRHx[3:0]寄存器选择复用功能： 推挽复用输

出：用户配置 MODE0 选择输出速度，配置 CNF0=10； 推挽开漏

输出：用户配置 MODE0 选择输出速度，配置 CNF0=11。

如果输出模式下对 IO 上下拉有要求，需要单独配置 GPIOx_DCR 寄存器，非开漏输出模式，上下拉失效。

在复位期间或复位之后，GPIO 端口被配置成浮空输入模式，串行线调试端口（Serial-Wired Debug pins）默认为输入 PU/PD 模式。

配置为通用输出模式后，输出数据寄存器（GPIOx_ODR）的值会输出到相应的 I/O 引脚。在每个 AHB 时钟周期，输入数据寄存器（GPIOx_IDR）捕捉 I/O 引脚上的数据。 注：并不是所有芯片都包括 JTAG 和 SWD 调试端口，芯片具体配置可参考芯片数据手册。

- PA14: SWCLK 置于下拉模式
- PA13: SWDIO 置于上拉模式

7.3.3 复用功能

配置复用功能寄存器打开 IO 对应的复用功能。

- 配置 IO 为复用输入功能时，端口选择上拉、下拉或浮空输入。
- 配置 IO 为复用输出功能时，端口选择推挽或开漏输出模式。
- IO 配置为双向复用功能时，端口选择推挽或开漏输出模式，输入变为浮空输入，开漏模式下可配置 GPIOx_DCR 寄存器选择弱上拉或下拉电阻。

当配置端口为复用输出功能时，端口与片上外设输出信号连接。如果仅仅通过软件方式配置 GPIO 引脚为复用输出功能，外设没有被激活，此时输出不确定。

7.3.4 GPIO 锁定机制

GPIO 存在锁定机制，能够保持设定 IO 配置不被改变。当对某一端口执行锁定机制后，在下一次复位之前，不能改变端口对应的配置。锁定键写序列为：

- GPIOx_LCKR[16]='1'+LCKR[15:0]。
- GPIOx_LCKR[16]='0'+LCKR[15:0]。
- GPIOx_LCKR[16]='1'+LCKR[15:0]。

使能 GPIOA 的 PA[0]端口锁定参考配置如下：

- GPIOA->GPIOA_LCKR=0x10001。
- GPIOA->GPIOA_LCKR=0x00001。
- GPIOA->GPIOA_LCKR=0x10001。

当执行完上述三个步骤后，GPIOA_LCKR 寄存器的第 16 位置 1，在下一次软件复位之前，写 GPIOA_LCKR 寄存器无效，GPIOA_LCKR 寄存器的第 16 保持为 1，不会被更改，PA[0]会一直保持锁定之前的配置不变。

当端口被锁定后，只能在软件复位之后才能再次更改端口位的配置，GPIOx_LCKR 寄存器的一个锁定位锁定端口配置寄存器（GPIOx_CRL）与（GPIOx_CRH）中的 4 个位。

注意事项：

以上配置只是锁定了 PA[0]的配置，对于 PA[15:1]以及其它 GPIO 控制寄存器的配置操作依然有效。

7.3.5 输入配置

当 I/O 端口配置为输入时：

- 施密特触发输入使能。
- 输出缓冲被禁用。
- 可以选择浮空、上拉或下拉输入模式。
- I/O 脚上的数据在每个 AHB 时钟被采样到输入数据寄存器。
- 读访问输入数据寄存器可得到 I/O 状态。

下图给出了 I/O 端口的输入配置：

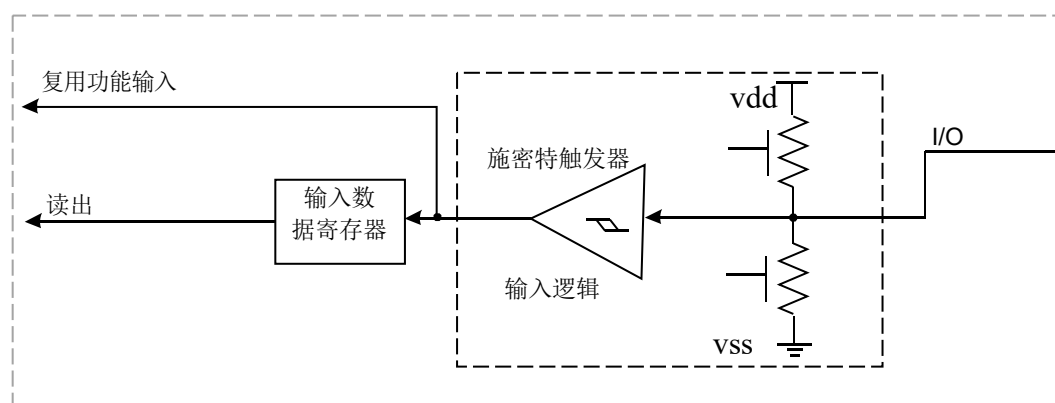


图 7-2 输入浮空/上拉/下拉配置

GPIOA 的 PA[0]端口输入上拉参考配置如下：

- GPIOA->GPIOA_ODR=0x0001。
- GPIOA->GPIOA_CRL=0x00000008。

GPIOA 的 PA[0]端口输入下拉参考配置如下：

- GPIOA->GPIOA_ODR=0x0000。

- GPIOA->GPIOA_CRL=0x00000008。

注意事项：

当端口配置上拉输入时，需要首先配置对应端口的 GPIO_ODR 寄存器对应位输出 1。当端口配置下拉输入时，需要首先配置对应端口的 GPIO_ODR 寄存器对应位输出 0。

7.3.6 输出配置

当 GPIO 配置为输出时：

- 施密特触发输入使能。
- 输出缓冲使能。
- 通用输出模式下，弱上拉和弱下拉电阻被禁用。
- 开漏模式：端口输出数据寄存器配置为 0 时，对应的引脚输出低电平，端口输出数据寄存器配置为 1 时，对应的管脚处于高阻态。
- 推挽模式：输出寄存器配置为 0 时，对应的引脚输出低电平，输出寄存器配置为 1 时，对应的管脚输出高电平。
- 对端口输出数据寄存器读操作，返回上次写入值。
- 对端口输入数据寄存器进行读操作，获得当前 I/O 的状态。

下图为 I/O 端口的输出配置：

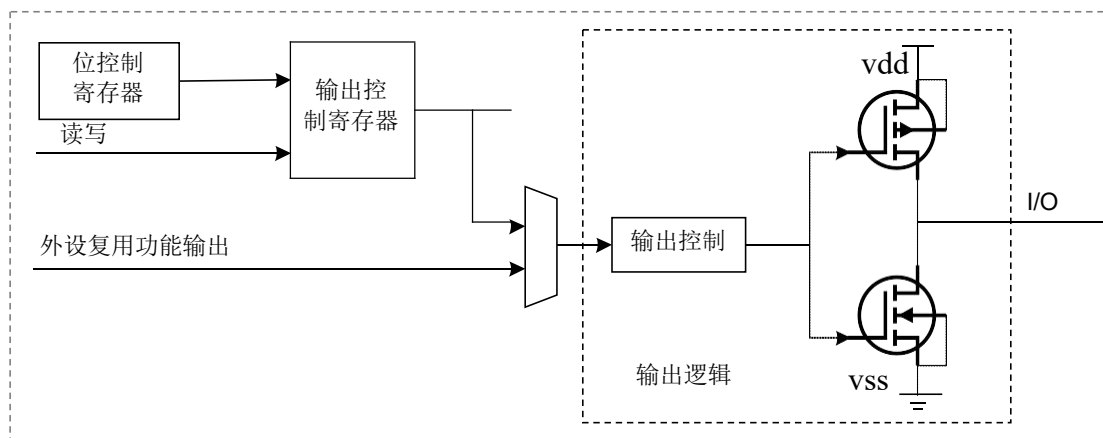


图 7-3 输出配置

7.3.7 复用功能配置

当配置引脚为复用功能时：

- 施密特触发输入使能。
- 输出缓冲器可以配置为开漏或推挽。
- 在开漏输出模式下，通过配置 GPIOx_DCR 寄存器选择弱上拉或下拉电阻。

- 当配置为输入时，可选弱上拉或弱下拉电阻。
- I/O 脚上数据在每个 AHB 时钟周期被采样到输入数据寄存器。

下图为 I/O 端口复用功能的配置，具体见 AFRL 与 AFRH 寄存器与数据手册部分。

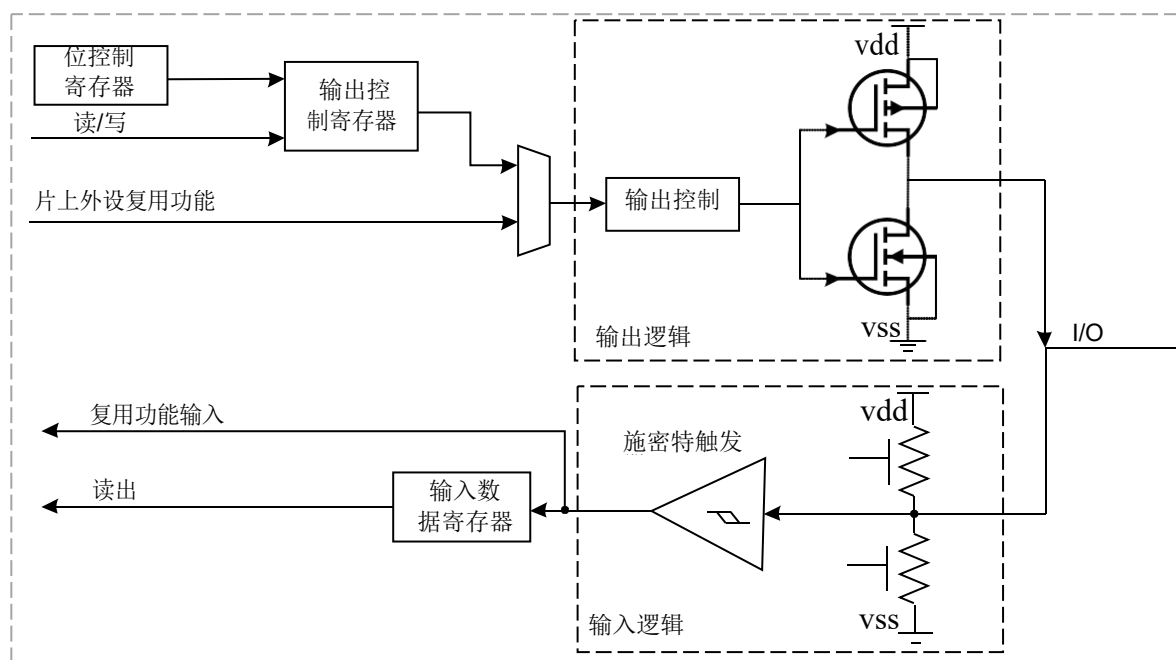


图 7-4 复用功能配置

7.3.8 模拟输入配置

当 I/O 端口被配置成模拟输入配置时：

- 输出缓冲器禁用。
- 施密特触发输入禁用。
- 弱上拉与弱下拉电阻禁用。
- 端口输入数据寄存器保持为 0。

下图为 I/O 端口的模拟输入配置

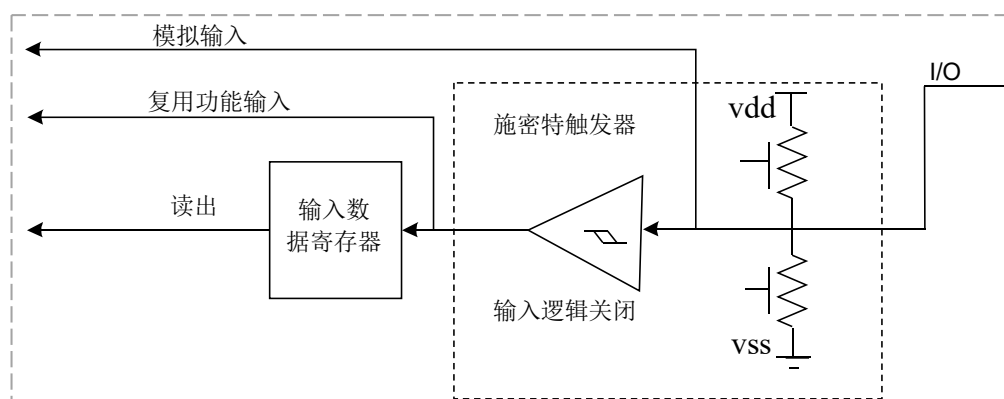


图 7-1 模拟输入

7.3.9 外部时钟复用 GPIO 端口

外部 HSE 时钟复用 GPIO，当对应的时钟 PAD 用做 GPIO 功能时，需先关闭外部时钟，再按照正常的 GPIO 功能操作，具体映射关系参考芯片数据手册部分。

7.3.10 SWD 复用功能重映射

SWD 调试接口信号被映射到 GPIO 端口上，如下表所示：

表 7-2 SWD 复用功能重映射

复用功能	GPIO 端口
SWDIO	PA13
SWCLK	PA14

7.3.11 NRST 复用功能重映射

包含了 NRST 功能的 GPIO，默认用作 NRST 功能，当需要将其用作 GPIO 的功能时，需要首先配置闪存选项字节寄存器(FLASH_OBR)的 OBR_RSTN 位等于 0，然后经系统复位后，配置生效。

包含了 NRST 功能的 GPIO，芯片内部默认接上拉电阻，当用作 NRST 功能时，禁止配置该引脚相关的 GPIO 控制器，避免影响复位功能，保持默认配置即可。

7.4 寄存器

7.4.1 寄存器总览

表 7-3 GPIO 寄存器概览

Offset	Acronvm	Register Name	Reset
0x00	GPIOx_CRL	端口配置低寄存器	0x44444444
0x04	GPIOx_CRH	端口配置高寄存器	0x44444444
0x08	GPIOx_IDR	端口输入数据寄存器	0x0000XXXX
0x0C	GPIOx_ODR	端口输出数据寄存器	0x00000000
0x10	GPIOx_BSRR	端口设置/清除寄存器	0x00000000
0x14	GPIOx_BRR	端口位清除寄存器	0x00000000
0x18	GPIOx_LCKR	端口配置锁定寄存器	0x00000000
0x1C	GPIOx_DCR	端口输出开漏控制寄存器	0x00000000
0x20	GPIOx_AFR1	端口复用功能低位寄存器	0xFFFFFFFF
0x24	GPIOx_AFRH	端口复用功能高位寄存器	0xFFFFFFFF

注：GPIOx 中“x”的可能范围是 A 到 I，但并不是所有芯片均包括所有 GPIOA 到 GPIOI 组，各芯片 的具体配置可参考各芯片的数据手册。

7.4.2 GPIOx_CRL 端口配置低寄存器

偏移地址：0x00

复位值：GPIOA_CRL: 0x4444 4444, GPIOB_CRL: 0x4444 4444

GPIOC_CRL: 0x0000 0000, GPIOD_CRL: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7		MODE7		CNF6		MODE6		CNF5		MODE5		CNF4		MODE4	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3		MODE3		CNF2		MODE2		CNF1		MODE1		CNF0		MODE0	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF7	<p>端口配置位 (y=7..0)</p> <p>配置 MODEy 等于 0，端口为输入模式，此时配置 CNFy 位选择输入模式： 00: 模拟输入模式 01: 浮空输入模式 10: 上拉/下拉输入模式 11: 保留</p> <p>配置 MODEy 不等于 0，端口为输出模式，此时配置 CNFy 位选择输出模式： 00: 通用推挽输出模式 01: 通用开漏输出模式 10: 复用功能推挽输出模式 11: 复用功能开漏输出模式</p> <p>端口输入输出配置 (MODEy) (y = 0..7) 软件配置相应的 I/O 端口；参考端口位配置表</p> <p>配置 MODEy 不等于 0 时，不同配置输出速度不同： 00: 输入模式； 其它配置输出速度参考芯片数据手册部分。</p>
27:26	CNF6	
23:22	CNF5	
19:18	CNF4	
15:14	CNF3	
11:10	CNF2	
7:6	CNF1	
3:2	CNF0	
29:28	MODE7	
25:24	MODE6	
21:20	MODE5	
17:16	MODE4	
13:12	MODE3	
9:8	MODE2	
5:4	MODE1	
1:0	MODE0	

7.4.3 GPIOx_CRH 端口配置高寄存器

偏移地址：0x04

复位值：GPIOA_CRH: 0x4444 4444, GPIOB_CRH: 0x4444 4444

GPIOC_CRH: 0x4440 0000, GPIOD_CRH: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15		MODE15		CNF14		MODE14		CNF13		MODE13		CNF12		MODE12	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11		MODE11		CNF10		MODE10		CNF9		MODE9		CNF8		MODE8	

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

Bit	Field	Description
31:30	CNF15	<p>端口配置位 ($y=15..8$)</p> <p>配置 $MODEy$ 等于 0, 端口为输入模式, 此时配置 $CNFy$ 位选择输入模式:</p> <p>00: 模拟输入模式</p> <p>01: 浮空输入模式</p> <p>10: 上拉/下拉输入模式</p> <p>11: 保留</p> <p>配置 $MODEy$ 不等于 0, 端口为输出模式, 此时配置 $CNFy$ 位选择输出模式:</p> <p>00: 通用推挽输出模式</p> <p>01: 通用开漏输出模式</p> <p>10: 复用功能推挽输出模式</p> <p>11: 复用功能开漏输出模式</p> <p>端口输入输出配置 ($MODEy$) ($y = 15..8$)</p> <p>软件配置相应的 I/O 端口; 参考端口位配置表</p> <p>配置 $MODEy$ 不等于 0 时, 不同配置输出速度不同:</p> <p>00: 输入模式;</p> <p>其它配置输出速度参考芯片数据手册部分。</p>
27:26	CNF14	
23:22	CNF13	
19:18	CNF12	
15:14	CNF11	
11:10	CNF10	
7:6	CNF9	
3:2	CNF8	
29:28	MODE15	
25:24	MODE14	
21:20	MODE13	
17:16	MODE12	
13:12	MODE11	
9:8	MODE10	
5:4	MODE9	
1:0	MODE8	

7.4.4 GPIOx_IDR 端口输入数据寄存器

偏移地址: 0x08 复位值:

0x0000 XXXX

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
IDRy($y=15..0$)															
r															

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	IDRy	端口输入数据 ($y=15..0$) 读出的值代表对应的 I/O 状态

7.4.5 GPIOx_ODR 端口输出数据寄存器

偏移地址: 0xC 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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

ODRy(y=15~0)

rw

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	ODRy	端口输出数据 (y=15..0) 配置为通用输出模式时, 写入值输出到对应的 IO 注: 操作 GPIOx_BSRR (x=A..F) 寄存器可以分别独立的对各个 ODR 位置 1 或清 0。

7.4.6 GPIOx_BSRR 端口设置/清除寄存器

偏移地址: 0x10 复位值:

0x0000 000

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

BRy(y=15~0)

w

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

BSy(y=15~0)

w

Bit	Field	Description
31:16	BRy	端口清除位 y (y=15..0) 写 0 相应的 ODRy 位保持不变 写 1 清除对应的 ODRy 位为 0
15:0	BSy	端口置位 y (y=15..0) 写 0 相应的 ODRy 位保持不变 写 1 置位对应的 ODRy 位为 1 注: 同时写 BSy 位与 BRy 位为 1 时, BSy 的优先级高于 BRy

7.4.7 GPIOx_BRR 端口位清除寄存器

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

BRy(y=15~0)

w

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	BRy	端口清除位 y (y=15..0) 写 0 相应的 ODRY 位保持不变 写 1 清除对应的 ODRY 位为 0

7.4.8 GPIOx_LCKR 端口配置锁定寄存器

地址偏移: 0x18 复位值:

0x0000 0000

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

Bit	Field	Description
31:17	Reserved	始终读为 0
16	LCKK	锁键 (Lock key) 该位可随时读出,它只可通过锁键写入序列修改。 0: 端口配置锁键位未被激活 1: 端口配置锁键位被激活,下次软件复位前 GPIOx_LCKR 寄存器被锁住 锁键序列: 写 1->写 0->写 1
15:0	LCKy	端口 x 的锁位 y (y = 15..0) 这些位可读可写但只能在 LCKK 位为 0 时写入。 0: 不锁定端口的配置 1: 锁定端口的配置

7.4.9 GPIOx_DCR 端口输出开漏控制寄存器

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
PX15		PX14		PX13		PX12		PX11		PX10		PX9		PX8	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PX7		PX6		PX5		PX4		PX3		PX2		PX1		PX0	
rw															

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31:2	PX15-PX1	见 PX0
1:0	PX0	PX0[1: 0]: 11: 开漏输出模式下, 端口上拉 01: 开漏输出模式下, 端口下拉 x0: 开漏输出模式下, 端口无上下拉

7.4.10 GPIOx_AFRL 端口复用功能低位寄存器

偏移地址: 0x20

复位值: GPIOA_AFRL: 0xFFFF FFFF, GPIOB_AFRL: 0xFFFF FFFF GPIOC_AFRL:
0x0000 0000, GPIOD_AFRL: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7				AFR6				AFR5				AFR4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3				AFR2				AFR1				AFR0			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	端口 x 的位 y (y = 0..7) 的复用功能选择位, 软件写配置。 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

7.4.11 GPIOx_AFRH 端口复用功能高位寄存器

偏移地址: 0x24

复位值: GPIOA_AFRH: 0xFFFF FFFF, GPIOB_AFRH: 0xFFFF FFFF GPIOC_AFRH:
0xFFFF0 0000, GPIOD_AFRH: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15				AFR14				AFR13				AFR12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11				AFR10				AFR9				AFR8			
rw				rw				rw				rw			

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31:0	AFRy	端口 x 的位 y (y = 8..15) 的复用功能选择位，软件写配置。 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

8 EXTI 中断和事件

8.1 简介

嵌套向量中断控制器（NVIC）连接处理器核，管理低延迟的异常和中断处理。NVIC 内部包含 2 位的中断优先级配置位，从而可提供 4 个中断优先级等级。EXTI 模块包括边沿检测电路，能够产生中断请求或者唤醒事件，边沿检测支持上升沿、下降沿、任意边沿配置。每一个边沿检测电路支持独立的使能与屏蔽。

8.2 主要特征

- 独立触发与屏蔽每个中断
- 软件配置中断/事件输出
- 产生唤醒事件唤醒低功耗模式
- 挂起寄存器保存对应每条中断线的状态
- 所有 GPIO 支持配置为 EXTI 的触发源
- 支持上升沿触发，下降沿触发和任意边沿触发

8.3 功能描述

8.3.1 功能框图

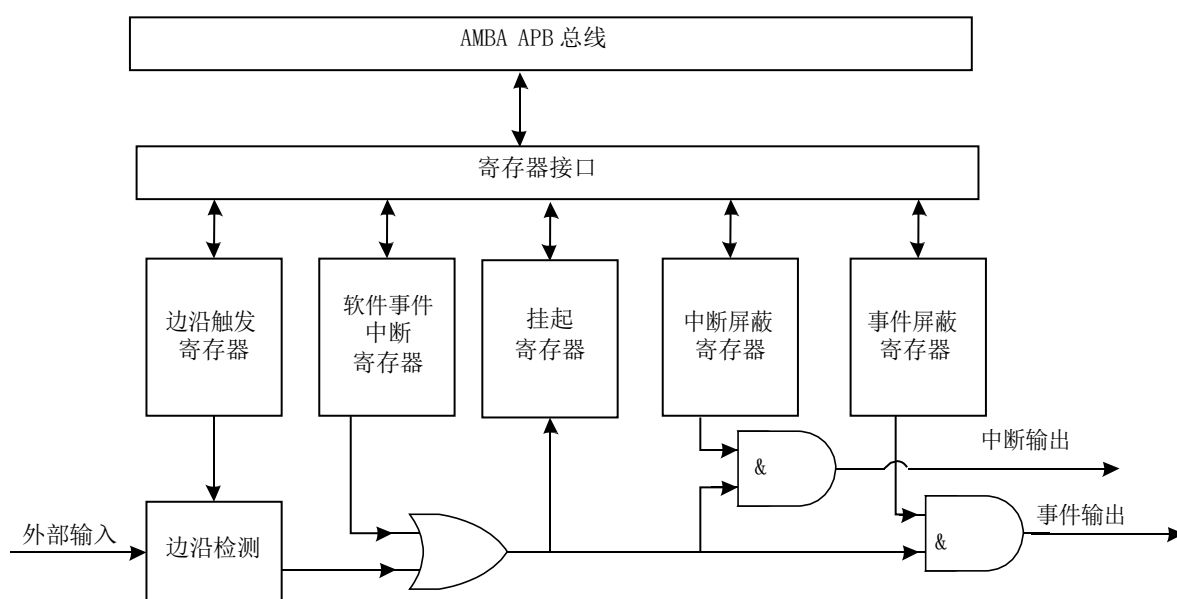


图 8-1 EXTI 结构框图

8.3.2 中断和异常向量

在 Handler 模式下，Cortex-M0 处理器与内嵌中断向量控制（NVIC）对所有的异常进行优先级区分处理。当异常发生时，系统会将当前处理的工作压栈，执行完中断服务程序后出栈。取向量与当前工作的压栈并行进行的，提高了中断的效率，下表分别列出了异常类型与中断向量。

表 8-1 异常向量表

位置	优先级	优先级类型	名称	说明	地址
				保留	0x0000 0000
	-3	固定	Reset	复位	0x0000 0004
	-2	固定	NMI	不可屏蔽中断 RCC 时钟安全系统(CSS)连接到 NMI	0x0000 0008
	-1	固定	硬件失效 (HardFault)	所有类型的失效	0x0000 000C

表 8-2 中断向量表

位置	优先级	优先级类型	名称	说明	地址
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000 002C
				保留	0x0000 0030
				保留	0x0000 0034
	5	可设置	PendSV	可挂起的系统服务	0x0000 0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	IWDG	连到 EXTI24 的独立看门狗中断	0x0000_0040
1	8	可设置	PVD	连到 EXTI16 的电源电压检测 (PVD)中断	0x0000_0044
2	9	可设置	RTC_BKP	RTC_BKP 全局中断与 RTC ALARM 连接到 EXTI17 的中断	0x0000_0048
3	10	可设置	Flash	闪存全局中断	0x0000_004C
4	11	可设置	RCC	RCC 全局中断	0x0000_0050
5	12	可设置	EXTI0_1	EXTI 线[1: 0]中断	0x0000_0054
6	13	可设置	EXTI2_3	EXTI 线[3: 2]中断	0x0000_0058
7	14	可设置	EXTI4_15	EXTI 线[15: 4]中断	0x0000_005C
8	15	可设置	HWDIV	Hard_Divider	0x0000_0060
9	16	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_0064
10	17	可设置	DMA1 通道 2/3	DMA1 通道 2/3 全局中断	0x0000_0068
11	18	可设置	Reserved	Reserved	0x0000_006C
12	19	可设置	ADC_COMP	ADC 全局中断 连到 EXTI19 的比较器 1 输出	0x0000_0070
13	20	可设置	TIM1_BRK_UP_TRG_COM	TIM1 刹车、更新、触发、COM 中断	0x0000_0074
14	21	可设置	TIM1_CC	TIM1 捕捉比较中断	0x0000_0078
15	22	可设置	Reserved	Reserved	0x0000_007C
16	23	可设置	TIM3	TIM3 全局中断	0x0000_0080
17	24	可设置	Reserved	Reserved	0x0000_0084
18	25	可设置	Reserved	Reserved	0x0000_0088
19	26	可设置	TIM14	TIM14 全局中断	0x0000_008C

位置	优先级	优先级类型	名称	说明	地址
20	27	可设置	Reserved	Reserved	0x0000_0090
21	28	可设置	TIM16	TIM16 全局中断	0x0000_0094
22	29	可设置	TIM17	TIM17 全局中断	0x0000_0098
23	30	可设置	I2C1	I2C1 全局中断	0x0000_009C
24	31	可设置	Reserved	Reserved	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000_00A4
26	33	可设置	Reserved	Reserved	0x0000_00A8
27	34	可设置	USART1	USART1 全局中断	0x0000_00AC
28	35	可设置	USART2	USART2 全局中断	0x0000_00B0
29	36	可设置	USART3	USART3 全局中断	0x0000_00B4
30	37	可设置	Reserved	Reserved	0x0000_00B8
31	38	可设置	CSM	CSM 全局中断	0x0000_00BC

8.3.3 唤醒事件管理

EXTI 模块支持产生中断或者事件用于将系统从低功耗模式下唤醒，用户执行 WFE 指令进入相应的低功耗模式后，可以通过配置 EXTI 线产生事件输出唤醒系统，用户执行 WFI 进入低功耗模式后，可以通过配置 EXTI 线产生中断输出唤醒系统，具体详细配置参考电源控制章节。

8.3.4 中断功能描述

要使能中断功能，产生中断，首先配置边沿检测触发寄存器为需要的触发类型，打开相应的中断屏蔽寄存器的对应位允许中断请求。在对应的外部中断线检测到配置的触发条件时，产生一个中断请求，挂起寄存器对应位置 1，通过对挂起寄存器对应位写 1，将清除中断。

配置产生事件，首先配置边沿检测触发寄存器为需要的触发类型，打开相应的事件屏蔽寄存器的对应位允许事件请求。在对应的外部中断线检测到配置的触发条件时，产生一个事件请求。

使能软件中断/事件寄存器的对应位，也能够产生中断/事件请求。

8.3.5 硬件中断输出

配置硬件中断源的具体步骤如下：

- 打开对应中断线的屏蔽位（EXTI_IMR），使能中断。
- 配置对应中断线的触发寄存器位（EXTI_RTSR/EXTI_FTSR）。
- 打开对应连接到 NVIC 的中断通道，使得中断请求能够传递到 CPU，被正确的响应。

当配置 EXTI_x（x=31~0）线产生中断输出后，EXTI_PR 寄存器的对应位会置 1，需要清除 EXTI_PR 寄存器的对应挂起位才能再次检测 EXTI_x（x=31~0）线的翻转并产生中断。

清除 EXTI_PR 寄存器挂起位有以下三种方式：

- EXTI_PR 寄存器的挂起位写 1。

- 如果配置了上升沿触发选择寄存器（EXTI_RTSR），对应位写 0 会清除挂起位。如果配置了下降沿触发选择寄存器（EXTI_FTSR），对应位写 0 会清除挂起位。
- 通过改变 EXTI 线的边沿检测极性清除。

8.3.6 硬件事件输出

配置硬件事件源的具体步骤如下：

- 打开对应事件线的屏蔽位（EXTI_EMR）。
- 配置对应事件线的触发寄存器位（EXTI_RTSR/EXTI_FTSR）。

8.3.7 软件中断与事件输出

支持通过软件的方式配置产生中断与事件，具体步骤如下：

- 使能事件或中断使能位（EXTI_IMR，EXTI_EMR）。
- 配置软件中断事件寄存器对应位为 1（EXTI_SWIER）。

8.3.8 外部中断映射

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

表 8-3 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PA0;PB0;PD0	SYSCFG_EXTICR1 寄存器中的 EXTI0
EXTI1	PA1;PB1; PD1	SYSCFG_EXTICR1 寄存器中的 EXTI1
EXTI2	PA2;PB2; PD2	SYSCFG_EXTICR1 寄存器中的 EXTI2
EXTI3	PA3;PB3; PD3	SYSCFG_EXTICR1 寄存器中的 EXTI3
EXTI4	PA4;PB4; PD4	SYSCFG_EXTICR2 寄存器中的 EXTI4
EXTI5	PA5;PB5; PD5	SYSCFG_EXTICR2 寄存器中的 EXTI5
EXTI6	PA6;PB6; PD6	SYSCFG_EXTICR2 寄存器中的 EXTI6
EXTI7	PA7;PB7; PD7	SYSCFG_EXTICR2 寄存器中的 EXTI7
EXTI8	PA8;PB8	SYSCFG_EXTICR3 寄存器中的 EXTI8
EXTI9	PA9;PB9	SYSCFG_EXTICR3 寄存器中的 EXTI9
EXTI10	PA10;PB10	SYSCFG_EXTICR3 寄存器中的 EXTI10
EXTI11	PA11;PB11	SYSCFG_EXTICR3 寄存器中的 EXTI11
EXTI12	PA12;PB12	SYSCFG_EXTICR4 寄存器中的 EXTI12
EXTI13	PA13;PB13;PC13	SYSCFG_EXTICR4 寄存器中的 EXTI13
EXTI14	PA14;PB14;PC14	SYSCFG_EXTICR4 寄存器中的 EXTI14
EXTI15	PA15;PB15;PC15	SYSCFG_EXTICR4 寄存器中的 EXTI15

其他的外部中断/事件控制器的连接如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 输出
- EXTI 线 19 连接到比较器 1 输出
- EXTI 线 24 连接到 IWDG 输出

8.4 寄存器

8.4.1 寄存器总览

表 8-4 EXTI 寄存器总览

Offset	Acronym	Register Name	Reset
0x00	EXTI_IMR	中断屏蔽寄存器	0x00000000
0x04	EXTI_EMR	事件屏蔽寄存器	0x00000000
0x08	EXTI_RTSR	上升沿触发选择寄存器	0x00000000
0x0C	EXTI_FTSR	下降沿触发选择寄存器	0x00000000
0x10	EXTI_SWIER	软件中断事件寄存器	0x00000000
0x14	EXTI_PR	挂起寄存器	0x00000000

8.4.2 EXTI_IMR 中断屏蔽寄存器

偏移地址：0x0 复位值：
0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							IMRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	IMRx	线 x 中断使能位 1: 配置该位为 1，使能线 x 对应的中断 0: 配置该位为 0，禁止线 x 对应的中断

8.4.3 EXTI_EMR 事件屏蔽寄存器

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.							EMRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	EMRx	线 x 事件使能位 1: 配置该位为 1，使能线 x 对应的事件 0: 配置该位为 0，禁止线 x 对应的事件

8.4.4 EXTI_RTSR 上升沿触发选择寄存器

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.							TRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	TRx	线 x 对应中断或事件的触发极性 1: 配置该位为 1，使能线 x 对应的上升沿触发中断或事件 0: 配置该位为 0，禁止线 x 对应的上升沿触发中断或事件

8.4.5 EXTI_FTSR 下降沿触发选择寄存器

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.							TRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TRx(x=15~0)

rw

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	TRx	线 x 对应中断或事件的触发极性 1: 配置该位为 1，使能线 x 对应的下降沿触发中断或事件 0: 配置该位为 0，禁止线 x 对应的下降沿触发中断或事件

8.4.6 EXTI_SWIER 软件中断事件寄存器

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.							SWIERx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIERx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	SWIERx	线 x 上的软件配置中断或事件使能 写 1 将设置 EXTI_PR 寄存器中相应的挂起位，同时配置 EXTI_IMR 或 EXTI_EMR 中对应位为 1，能够产生中断或事件。 注：向 EXTI_PR 寄存器的对应位写 1，可以清除该位

8.4.7 EXTI_PR 软件中断事件挂起寄存器

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.							PRx(x=24~16)								
							rc_w1								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRx(x=15~0)															
rc_w1															

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	PRx	线 x 触发挂起位 1：发生了选择的触发请求 0：没有发生触发请求 外部中断线上出现选择的边沿事件时，该位被置 1，写 1 清除该位，也可以通过改变边沿检测的极性清除。

9 HWDIV 硬件除法器

9.1 简介

硬件除法器能自动执行有符号或者无符号的 32 位整数除法运算。

9.2 主要特征

- 32 位除数和被除数，输出 32 位的商和余数
- 8 个 HCLK 周期完成一次除法运算
- 如果除数为零，会产生溢出中断标志位
- 写除数寄存器自动执行除法运算
- 读商和余数寄存器时硬件自动等待运算结束
- 有符号或者无符号整数除法运算

9.3 功能描述

硬件除法单元包括 4 个 32 位数据寄存器，分别为被除数，除数，商和余数，可以做有符号或者无符号的 32 位除法运算。通过硬件除法控制寄存器 HWDIV_CR 的 USIGN 位可以选择是有符号除法还是无符号除法。

每次写除数寄存器，会自动触发除法运算，在运算结束后，结果会写入到商和余数寄存器里。如果在运算结束前读商寄存器、余数寄存器或者状态寄存器，读操作会保持，直到当前运算结束才返回运算结果。如果除数为零，会产生溢出中断标志位。

每次必须先初始化除数与被除数寄存器，才能读取商、余数、状态寄存器的值。

9.4 寄存器

9.4.1 寄存器总览

表 9-1 HWDIV 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	HWDIV_DVDR	被除数寄存器	0x00000000
0x04	HWDIV_DVSR	除数寄存器	0x00000001
0x08	HWDIV_QUOTR	商寄存器	0x00000000
0x0C	HWDIV_RMDR	余数寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x10	HWDIV_SR	状态寄存器	0x00000000
0x14	HWDIV_CR	控制寄存器	0x00000001

9.4.2 HWDIV_DVDR 被除数寄存器

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

Bit	Field	Description
31:0	DIVIDEND	被除数寄存器位 (Dividend data)

9.4.3 HWDIV_DVSR 除数寄存器

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

Bit	Field	Description
31:0	DIVISOR	除数寄存器位 (Divisor data) 写完该寄存器后，自动触发除法运算。

9.4.4 HWDIV_QUOTR 商寄存器

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

QUOTIENT

r

Bit	Field	Description
31:0	QUOTIENT	商寄存器位 (Quotient data)

9.4.5 HWDIV_RMDR 余数寄存器

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

Bit	Field	Description
31:0	REMAINDER	余数寄存器位 (Remainder data)

9.4.6 HWDIV_SR 状态寄存器

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.															OVF
															r

Bit	Field	Description
31:0	Reserved	保留, 始终读为 0
0	OVF	除零溢出状态标志位 (overflow) 在下次除法操作前由软件写 1 清除 1: 当前操作除数为零。 0: 当前操作除数不为零。

9.4.7 HWDIV_CR 控制寄存器

偏移地址: 0x14 复位值:

0x0000 0001

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

Bit	Field	Description
31:2	Reserved	保留，始终读为 0
1	OVFE	除零溢出中断使能（Oerflow interrupt enable） 1：除零溢出中断使能 0：除零溢出中断不使能
0	USIGN	无符号除法使能（Unsigned enable） 1：无符号除法 0：有符号除法

10 BKP 备份寄存器

10.1 简介

备份寄存器含有 10 个 16 位的数据寄存器，用户端可用于存储 20 个字节的应用程序数据。当发生系统复位时，备份数据寄存器 BKP_DRn 不会被复位。

BKP 内部含有一组控制寄存器，可用于侵入事件检测和 RTC 时钟校准。复位后，硬件禁止访问备份寄存器与 RTC，以保护备份域，防止可能存在的意外的写操作。

可以通过如下步骤开启备份寄存器以及 RTC 的访问权限。

- 配置时钟和复位 RCC 的 APB1 外设时钟使能寄存器 RCC_APB1ENR 的 PWREN 和 BKPEN 位为 1，即使能 PWR 和 BKP 时钟
- 配置时钟和复位 RCC 的备份域控制寄存器 RCC_BDCR 的 DBP 位为 1，即使能对备份寄存器和 RTC 的访问。

10.2 主要特征

- 20 字节数据备份寄存器
- 提供存储 RTC 校验值的校验寄存器
- 提供中断使能和侵入检测的状态与控制寄存器
- 当 PC13 管脚不用于侵入检测时，可以用于输出 RTC 校准时钟、RTC 闹钟脉冲或者秒脉冲

10.3 功能描述

10.3.1 时钟校准

将 RTC 时钟经 64 分频输出到侵入检测引脚 TAMP 上可以测量该时钟。如果配置 RTC 校验寄存器 BKP_RTCCR 中的 CCO 位为 1，将会开启时钟输出功能。

ppm (parts per million) 误差的计算公式为：ppm 误差 = 偏差 / 基准值 * 10^6 。

若校准值为 1，则 RTC 校准时，每 2^{20} 个时钟周期扣除 1 个时钟脉冲，这相当于 $0.954\text{ppm}(1/2^{20} \times 10^6 = 0.954)$ 。而 BKP_RTCCR 寄存器校准值 CAL 最大为 127，所以最大可以减慢 $121\text{ppm}(0.954\text{ppm} \times 127 = 121)$ 。

10.3.2 侵入检测

上升沿侵入检测，配置 BKP_CR 的 TPE 位等于 1，BKP_CR 的 TPAL 位等于 0，当 TAMP 引脚出现 0 到 1 的电平翻转，会产生侵入检测事件，此时，会清除备份寄存器中的所有数据。

下降沿侵入检测，配置 BKP_CR 的 TPE 位等于 1，BKP_CR 的 TPAL 位等于 1，当 TAMP 引脚出现 1 到 0 的电平翻转，会产生侵入检测事件，此时，会清除备份寄存器中的所有数据。

检测到侵入事件，同时配置 BKP_CSR 寄存器的 TPIE 位为 1，产生中断输出。

PC13 管脚的侵入功能应该在侵入事件被检测且清除后关闭。为了防止软件在发生侵入事件期间对备份寄存器进行写操作，需要在写备份控制寄存器前，重新置位 BKP_CR 的 TPE 位来开启侵入检测功能，去监控 TAMP 引脚。

10.4 寄存器

10.4.1 寄存器总览

表 10-1 BKP 寄存器概览

Offset	Acronym	Register Name	Reset
0x50+4*(n-1)	BKP_DRn	备份数据寄存器 n	0x00000000
0x40	BKP_RTCCR	RTC 时钟校准寄存器	0x00000000
0x44	BKP_CR	备份控制寄存器	0x00000000
0x48	BKP_CSR	备份控制状态寄存器	0x00000000

10.4.2 备份数据寄存器 n(BKP_DRn)(n = 1 .. 10)

偏移地址：0x50 + 4 * (备份数据寄存器编号 - 1) 复位值：
0x0000

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

Bit	Field	Description
15:0	BKP	备份数据，用来存放用户数据。 注：当发生系统复位时，BKP_DRn 不会复位，可以由备份域复位或侵入引脚复位（当侵入引脚功能开启时）。

10.4.3 时钟校准寄存器 (BKP_RTCCR)

偏移地址：0x40 复位值：
0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ASOS rw	ASOE rw	CCO rw	CAL[6:0] rw						

Bit	Field	Description
15:10	Reserved	保留，必须保持复位值
9	ASOS	闹钟或秒输出选择（Alarm Clock Or Second Pulse） 配置 ASOE 位为 1，ASOS 位用于选择在 TAMP 引脚上输出的输出信号。 0：输出 RTC 闹钟脉冲 1：输出秒脉冲
8	ASOE	使能输出闹钟或秒脉冲（Alarm Or Second Output Enable） 根据 ASOS 位的设置，ASOE 位用于使能闹钟或秒脉冲输出到 TAMP 引脚。 注：输出脉冲的宽度为一个 RTC 时钟的周期。设置了 ASOE 位时不能开启 TAMP 的功能。只能由后备区域复位清除该位。
7	CCO	校准时钟输出（Calibration Clock Output） 0：无影响 1：配置 CCO 位为 1，在侵入检测引脚输出 64 分频后的 RTC 时钟 注：当 VDD 掉电时，清除该位。当 CCO 位置 1 时，必须关闭侵入检测功能以避免检测到无用的侵入信号。
6:0	CAL[6:0]	校准值（Calibration value） 校准值表示时钟脉冲跳过数量（每 2^{20} 个时钟脉冲）。可以用来对 RTC 进行校准，以 $1/2^{20} * 10^6$ ppm 的比例减慢时钟。 注:RTC 时钟可以被减慢 0~121ppm。

10.4.4 备份控制寄存器 (BKP_CR)

偏移地址：0x44 复位值：

0x0000

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

Bit	Field	Description
15:2	Reserved	保留，必须保持复位值
1	TPAL	侵入检测引脚有效电平（TAMP Pin Active Level） 0：TPE 位为 1，侵入检测 TAMP 引脚上的高电平会清除备份寄存器所有数据 1：TPE 位为 1，侵入检测 TAMP 引脚上的低电平会清除备份寄存器所有数据
0	TPE	侵入检测引脚使能（TAMP Pin Enable） 0：TAMP 引脚作为普通 IO 1：开启侵入检测引脚

10.4.5 备份控制状态寄存器 (BKP_CSR)

偏移地址：0x48 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TIF	TEF	Reserved					TPIE	CTI	CTE

	r	r		rw	w	w
--	---	---	--	----	---	---

Bit	Field	Description
15:10	Reserved	保留，必须保持复位值
9	TIF	<p>侵入中断标志（TAMP Interrupt Flag）</p> <p>当 TPIE 位为 1 并且检测到侵入事件时，该位由硬件置'1'。向 CTI 位写'1'来清除此标志位与中断（如果 TPIE 位被清除，则此位也会被清除）。</p> <p>0：无侵入中断</p> <p>1：产生侵入中断</p>
8	TEF	<p>侵入事件标志（TAMP Event Flag）</p> <p>当发生侵入事件时，此位由硬件置'1'。向 CTE 位写'1'可清除该标志位。</p> <p>0：无侵入事件</p> <p>1：产生侵入事件</p>
7:3	Reserved	保留，必须保持复位值
2	TPIE	<p>使能侵入引脚中断（TAMP Pin Interrupt Enable）</p> <p>0：禁止侵入检测中断</p> <p>1：允许侵入检测中断（必须配置 BKP_CR 寄存器的 TPE 位为'1'）</p> <p>注 1：侵入中断无法唤醒处于低功耗模式的内核。</p> <p>注 2：只有系统复位后才复位该位</p>
1	CTI	<p>清除侵入检测中断（Clear TAMP Interrupt）</p> <p>该位只能写入，读出值为 0。</p> <p>0：无效</p> <p>1：清除侵入检测中断和侵入检测中断标志（TIF）</p>
0	CTE	<p>清除侵入检测事件标志（Clear TAMP Event Flag），该位只能写入，读出值为 0。</p> <p>0：无效</p> <p>1：清除侵入检测事件标志（TEF），同时复位侵入检测器</p>

11 TIM1 高级定时器

11.1 简介

TIM1 由一个 16 位可实时编程预分频器和一个 16 位计数方向可调的自动重装载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。高级定时器具有多种用途，能够提供 PWM 输出、死区时间可编程的互补输出、单脉冲模式输出等多种功能。

11.2 功能框图

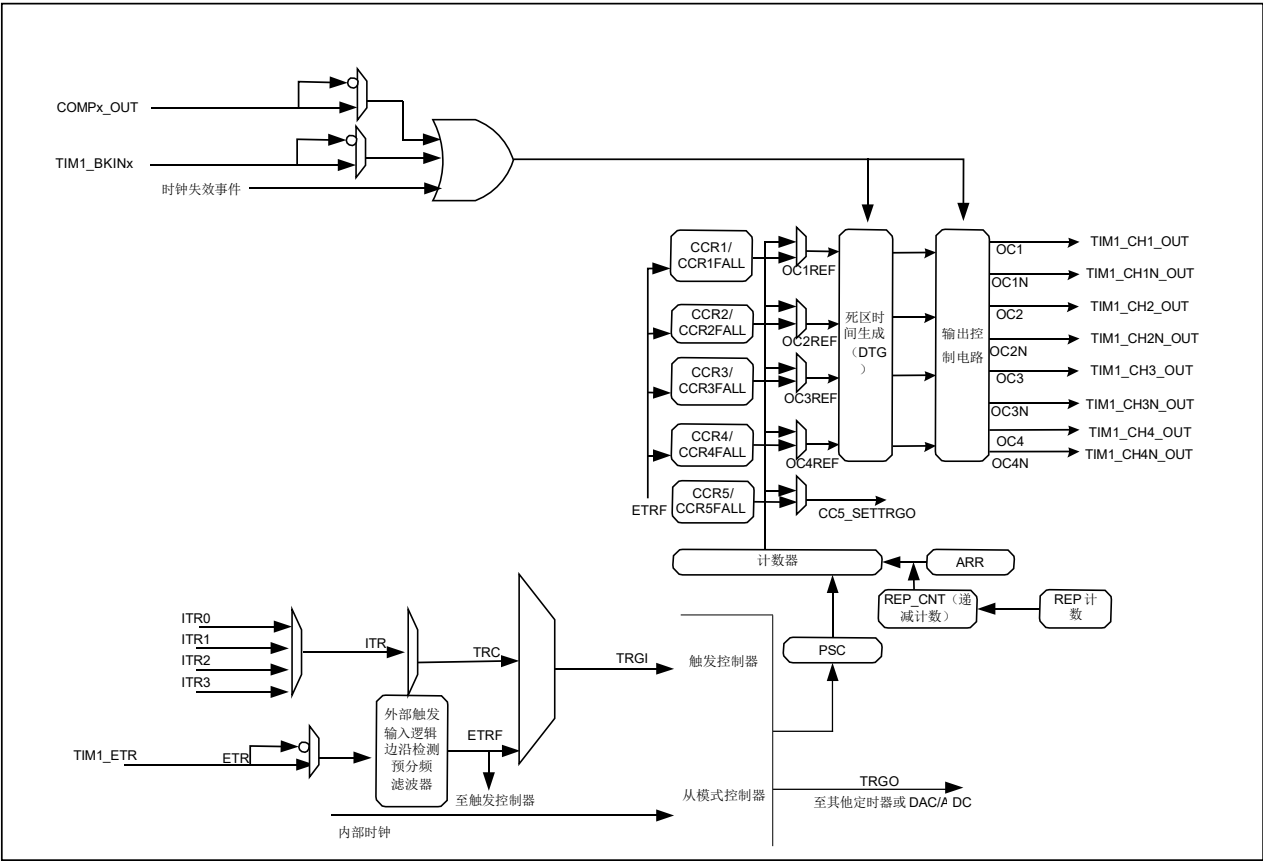


图 11-1 TIM1 结构图

上图为 TIM1 的结构框图，主要由触发单元、输出单元、时基单元、比较模块、刹车单元等结构组成。

11.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 时钟源可选：内部时钟源，外部时钟输入（ETR_x），内部触发输入(ITR_x)
- 16 位自动重装载计数器（计数方向：递增、递减、递增/递减）
- 8 位可编程重复计数功能，重复计数器可自动重装载（定时器到指定时间后自动更新重复寄存器）
- 外部信号控制定时器并且能够实现定时器间互连的同步电路
- 触发输入可以作为外部时钟或者逐周期管理
- 5 个比较通道(通道 1~4 支持输出，通道 5 仅提供中断及内部触发信号)
- 4 个输出通道，通道 1/2/3 有互补输出通道，通道 4 无互补输出通道
- 比较输出（控制输出波形或指示定时器已经计时结束）
- PWM 输出（死区时间可调；边沿对齐或中央对齐模式）
- 刹车输入可将计时器的输出信号置于安全状态（复位态或已知态，用户可选）
- 单脉冲输出
- 产生中断/DMA 请求的事件：更新事件、触发事件、比较输出或者刹车输入

11.4 功能描述

11.4.1 时钟

11.4.1.1 时钟选择 计数器的时钟

源有以下几种：

- 内部时钟（INT_CK）
- 外部时钟模式 1：外部触发输入 TRGI（包含 ITR_x、ETR_x）
- 外部时钟模式 2：外部触发输入 ETR（包含 ETR_x）

上述几种时钟选择示意图如下：

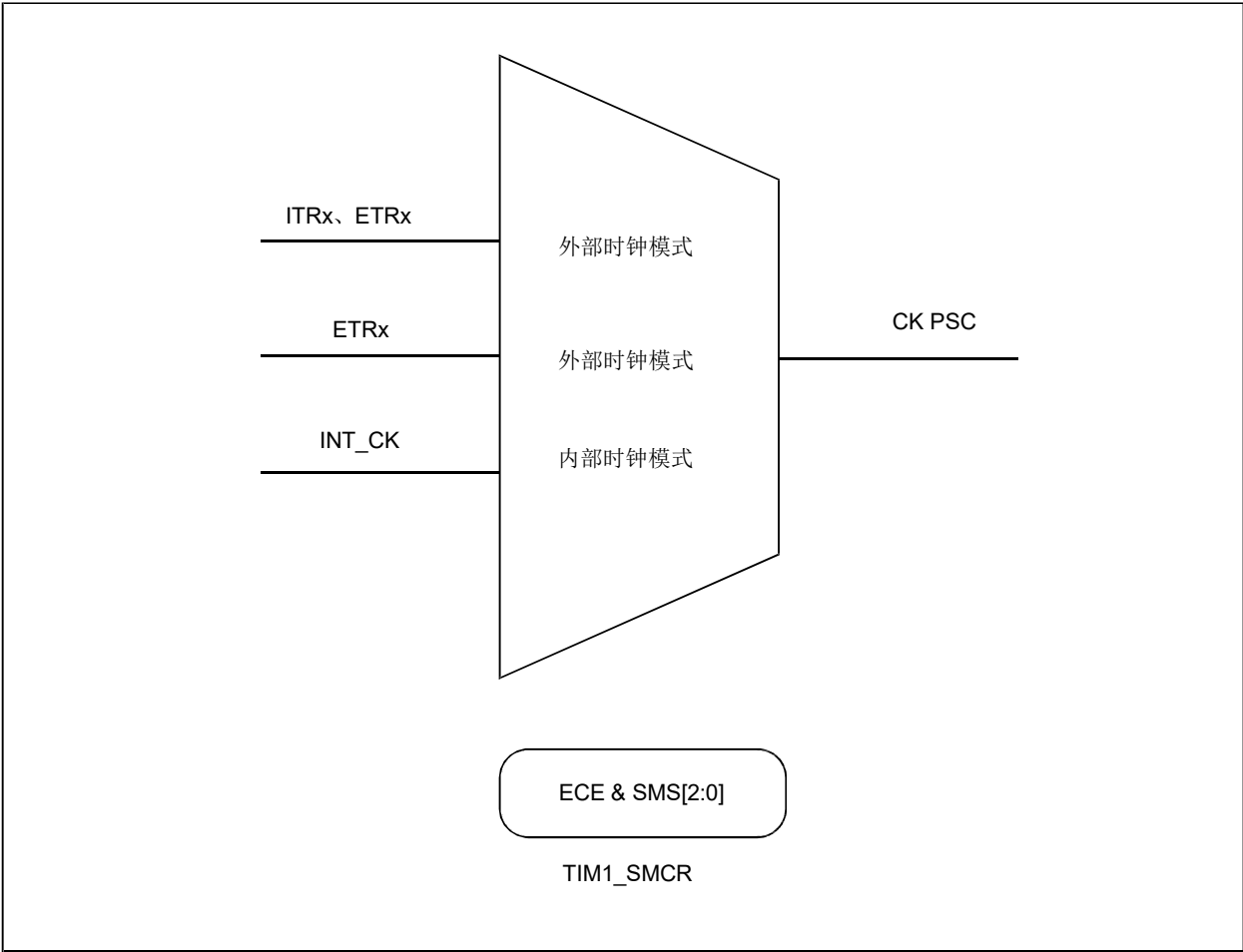


图 11-2 时钟选择

11.4.1.1.1 内部时钟源（INT_ CK）

当配置 TIM1_ SMCR 寄存器的 SMS=000、关闭从模式时，计数器使能打开，预分频器的时钟直接由内部时钟驱动。此时计数器时钟为内部时钟分频后的时钟。

11.4.1.1.2 外部时钟模式 1（外部触发输入 TRGI，包含 ITRx、ETRx）

当配置 TIM1_ SMCR 寄存器的 SMS = 111 时，选择外部时钟模式 1（TRGI）。计数器由选定的输入信号的每个上升沿或下降沿驱动。

例：计数器在 ITR1 输入端的上升沿递增计数，具体配置如下：

1. 配置 TIM1_ SMCR 寄存器的 TS=001，选择 ITR1 的作为触发输入源；配置 TIM1_ SMCR 寄存

器的 SMS=111，选择外部时钟模式 1。

2. 配置 TIM1_CR1 寄存器的 DIR=0，选择递增计数模式，配置 TIM1_CR1 寄存器的 CEN=1，启动计数器。

当 ITR1 出现有效边沿时，计数器递增计数一次且 TIF 标志位由硬件置 1。ITR1 的有效边沿和计数

器的实际时钟之间的延时取决于 ITR1 输入端同步电路设计。

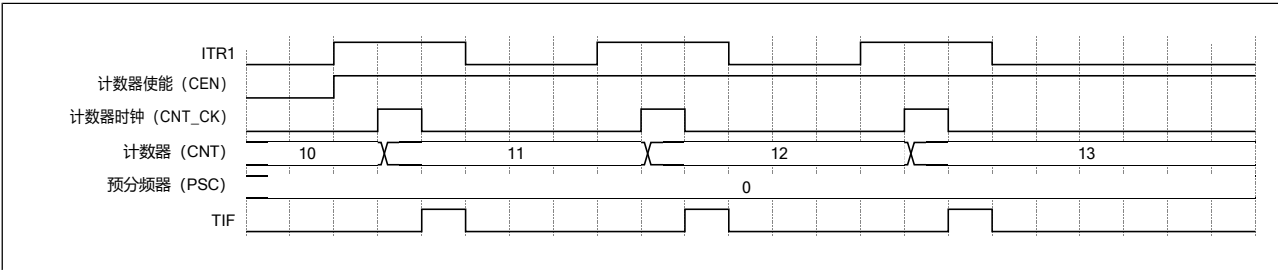


图 11-3 外部时钟模式 1 下的控制电路

11.4.1.1.3 外部时钟模式 2（外部触发输入 ETR，包含 ETRx）

当配置 TIM1_SMCR 寄存器的 ECE=1 时，使能外部时钟模式 2，计数器由 ETR 信号上的有效边沿驱动。

例：ETR 的每 4 个下降沿计数一次，递增计数，具体配置如下：

1. 配置 TIM1_SMCR 寄存器的 ETF[3: 0] = 0010，每 4 个 ETR 信号的有效边沿驱动计数器计数一次；配置 TIM1_SMCR 寄存器的 ETP=1，选择下降沿有效；配置 TIM1_SMCR 寄存器的 ECE=1，选择外部时钟模式 2。
2. 配置 TIM1_CR1 寄存器的 DIR=0，选择递增计数模式；配置 TIM1_CR1 寄存器的 CEN=1，启动计数器。

在 ETR 的下降沿和计数器实际时钟之间的延时取决于在 ETR 信号端的同步电路设计。

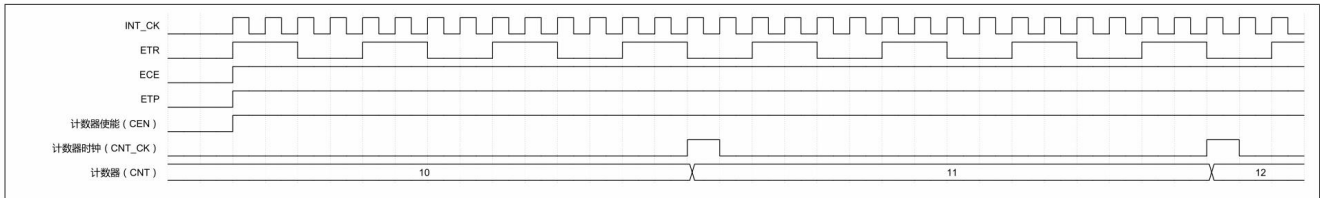


图 11-4 外部时钟模式 2 下的控制电路

11.4.1.2 时基单元

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

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

计数器的时钟由预分频器提供，预分频器由预分频计数器和对应的寄存器组成，分频系数为 1-65536，可以随时写入，在下一次更新事件时生效。

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

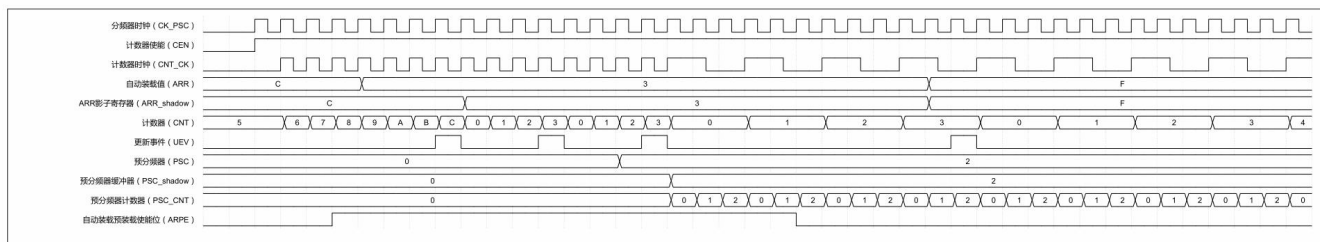


图 11-5 自动预装载

11.4.1.3 计数模式

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

11.4.1.3.1 递增计数模式

配置 TIM1_CR1 寄存器 CMS=0，DIR=0，选择递增计数模式。

递增计数模式下，在使能 TIM1_CR1 寄存器的 CEN 后计数器由 0 开始递增计数，直至 TIM1_ARR 的值，产生一个计数器上溢事件（更新事件），并从 0 开始重新递增计数。当用户启用了重复计数功能，重复计数器在每次上溢事件时递减计数，只有当重复计数器从设定值递减到 0 时，才会产生更新事件。设置 TIM1_EGR 寄存器的 UG=1，同样可以产生一个更新事件。

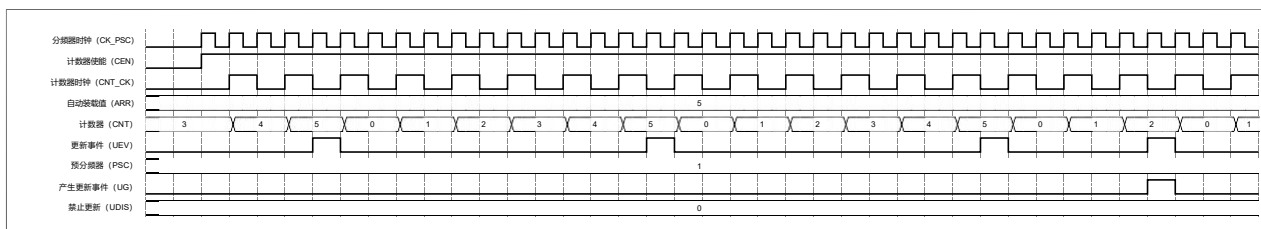


图 11-6 递增计数模式（UDIS=0）

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

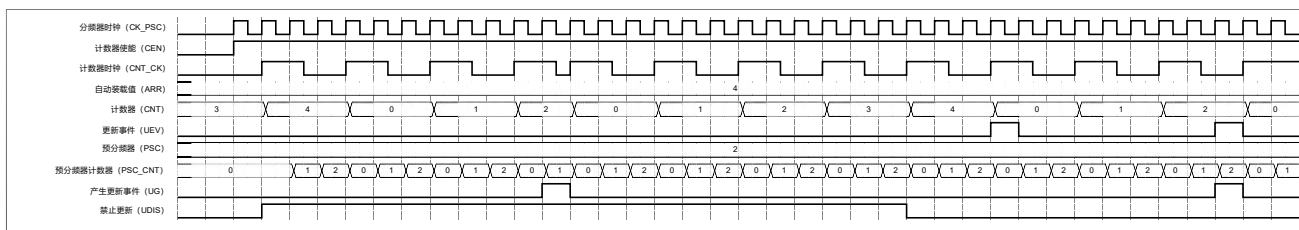


图 11-7 递增计数模式（UDIS=1 禁止产生更新事件）

注：发生更新事件时

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

11.4.1.3.2 递减计数模式

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

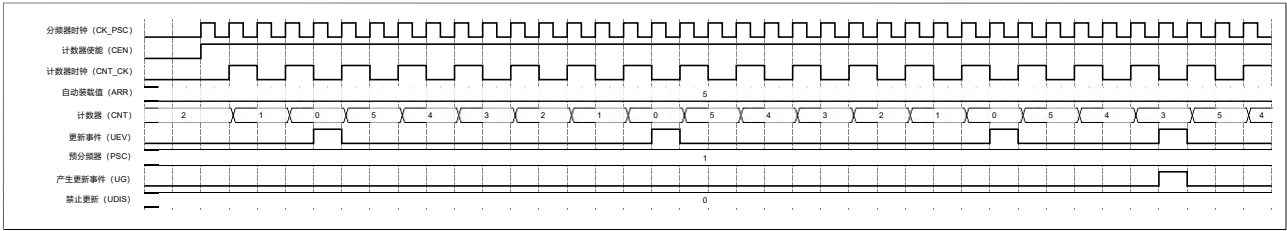


图 11-8 递减计数模式（UDIS=0）

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

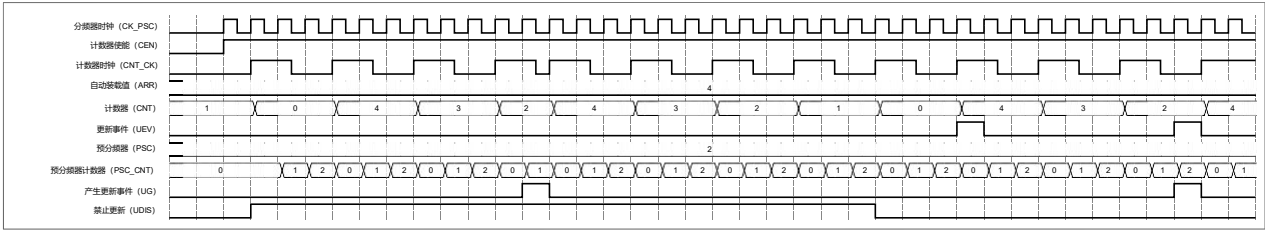


图 11-9 递减计数模式（UDIS=1 禁止产生更新事件）

11.4.1.3.3 中央计数模式（递增/递减计数模式）

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

中央对齐计数模式，递增计数和递减计数交替进行。递增计数到 ARR-1 时，产生一个上溢事件，然后从 ARR 开始递减计数到 1，产生一个下溢事件，再从 0 开始递增计数。

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

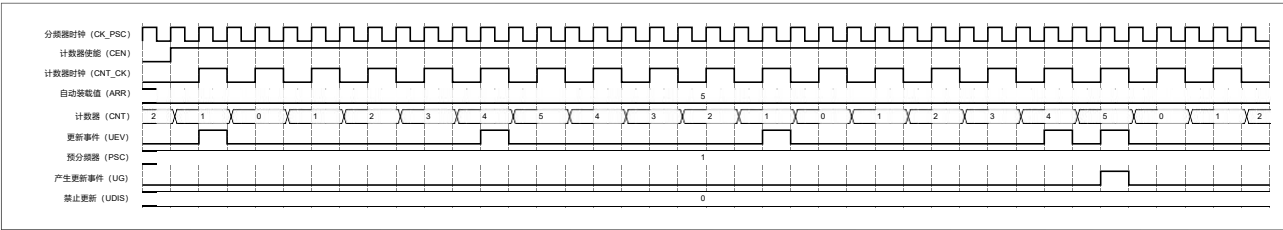


图 11-10 中央计数模式（UDIS=0）

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

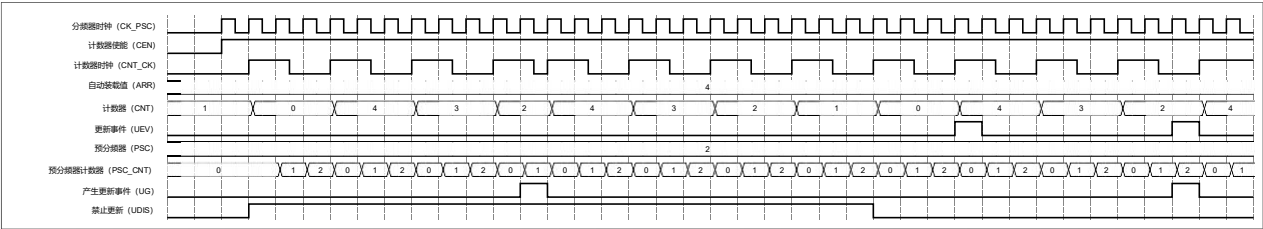


图 11-11 中央计数模式（UDIS=1 禁止产生更新事件）

11.4.2 重复计数器

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

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

通过配置 TIM1_PDER 寄存器的 CCDREPE，可以调整 DMA 请求的发生频率。CCDREPE=0，DMA 请求需要根据重复计数寄存器的值来产生；CCDREPE=1，DMA 请求在每次上溢或下溢时都会产生。

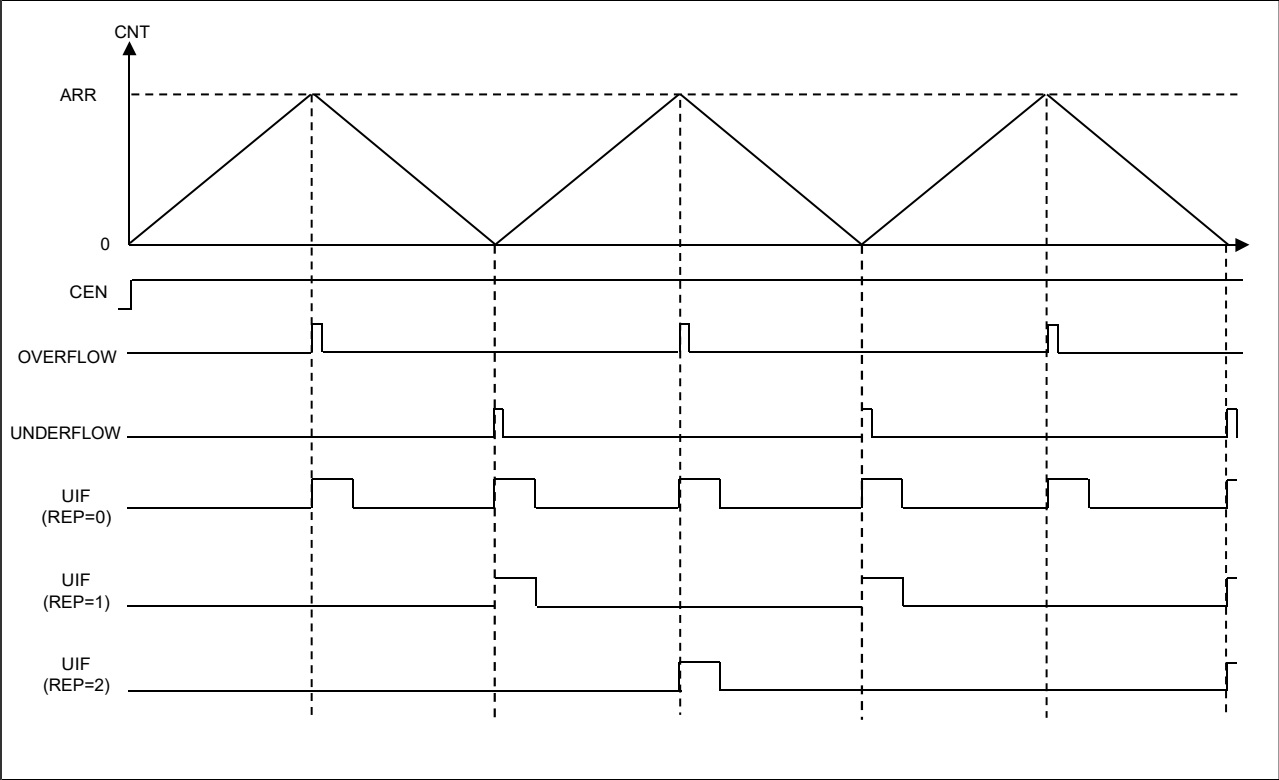


图 11-12 中央对齐模式重复计数时序图

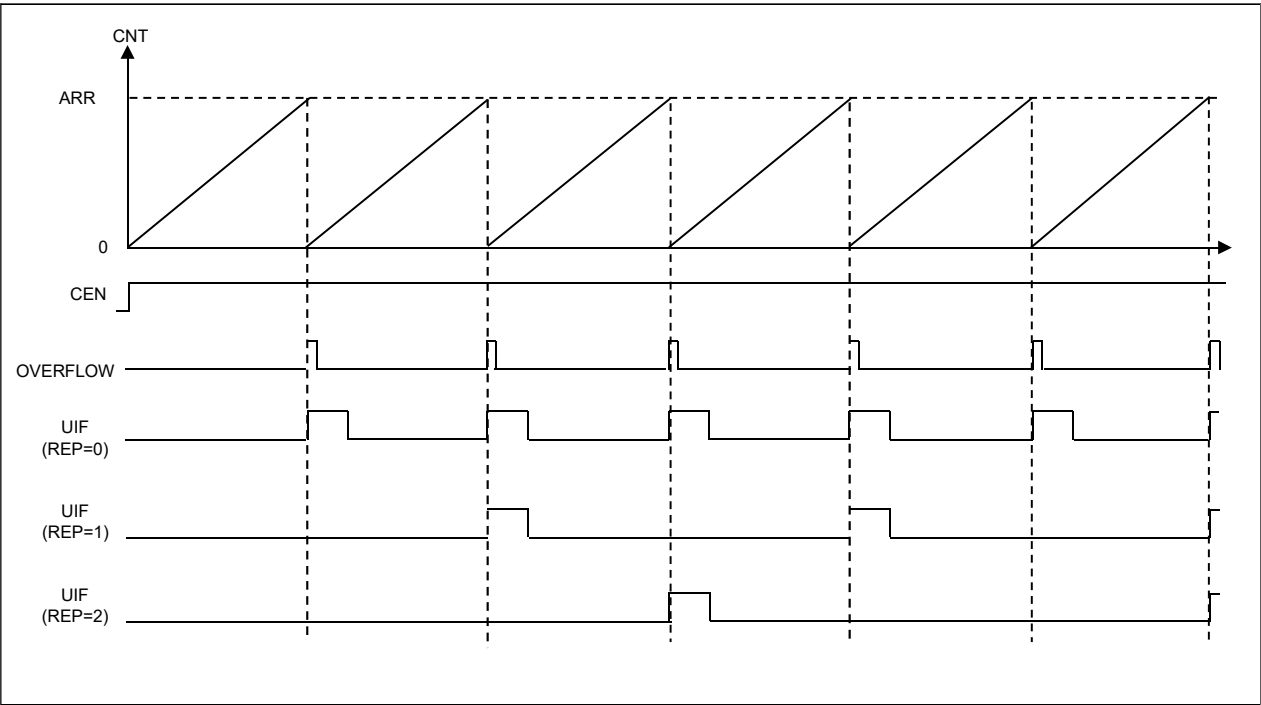


图 11-13 边沿对齐模式递增计数时序图

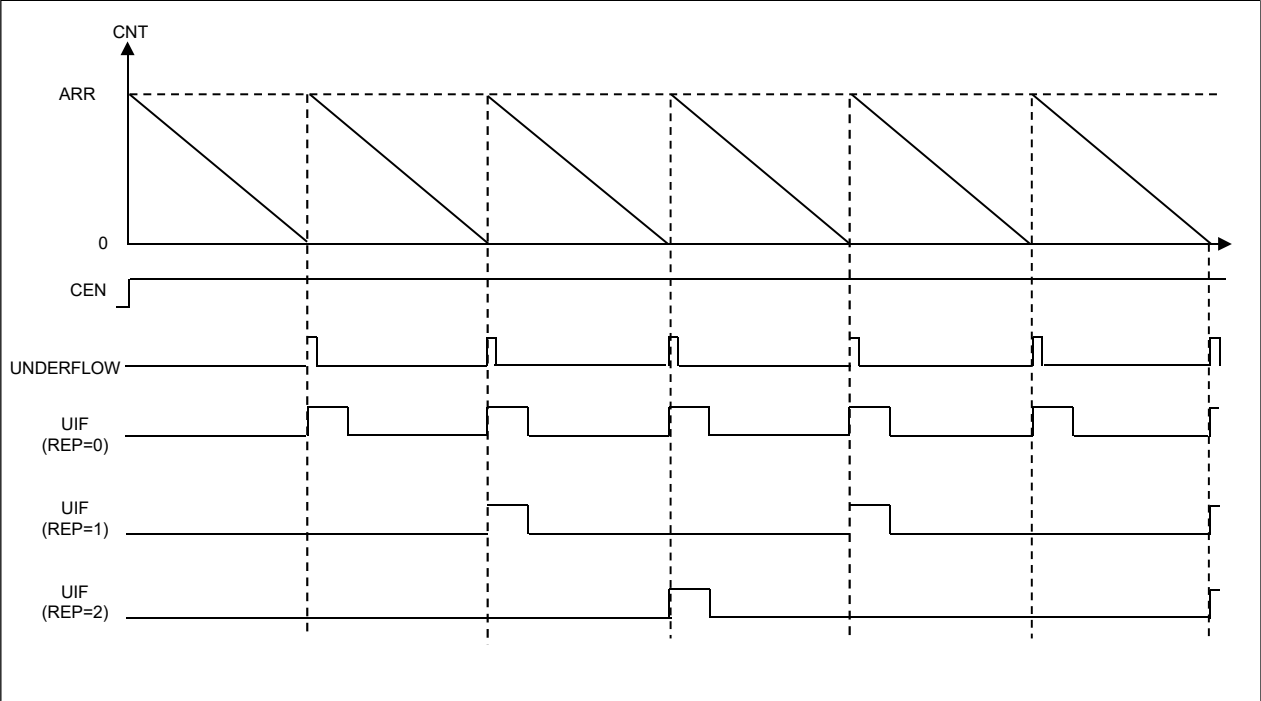


图 11-14 边沿对齐模式递减计数时序图

11.4.3 比较输出

比较通道的比较输出部分由比较器、输出控制电路和比较寄存器组成，其结构图如下图所示：

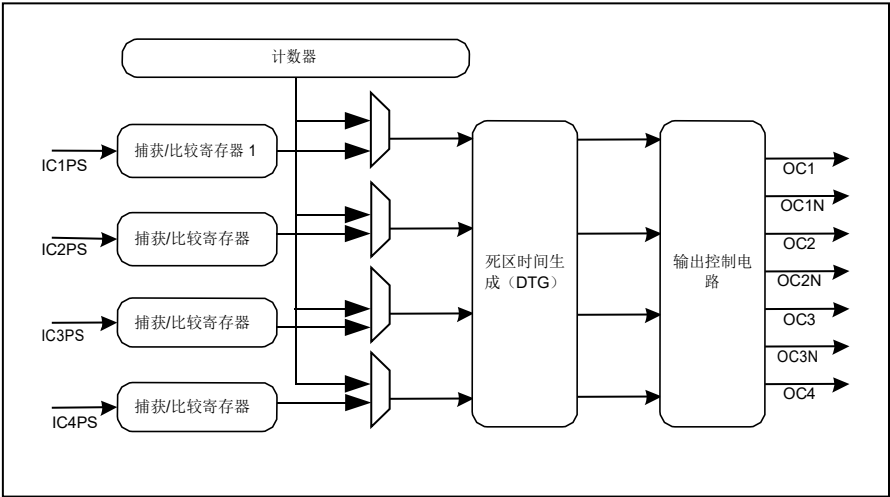


图 11-15 比较输出部分结构图

在比较输出模式下，比较寄存器的内容被载入到影子寄存器中，然后影子寄存器的内容和计数器当前值进行比较。比较模块包括一个比较寄存器（预装载寄存器）和一个影子寄存器，读写过程仅操作比较寄存器。

11.4.3.1 强制输出

配置 TIM1_CCMRx 寄存器 OCxM 位，可以直接将比较输出信号直接强制为有效或无效状态，不依赖于比较结果。配置 TIM1_CCMRx 寄存器 OCxM = 100，强置比较输出信号为无效状态。此时 OCxREF 被强置为低电平。配置 TIM1_CCMRx 寄存器 OCxM = 101，强置比较输出信号为有效状态。此时 OCxREF 被强置为高电平（OCxREF 始终为高电平有效）。

注：强制输出模式下，在 TIM1_CCRx 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断和 DMA 请求，仍会产生对应的中断和 DMA 请求。

11.4.3.2 比较输出

比较输出模式下，当计数器与比较寄存器值相同时，可以根据 TIM1_CCMRx 寄存器的 OCxM 位的配置用来输出不同的波形。

例如，当计数器与比较寄存器的内容匹配时，比较输出模式下的操作如下：

1. 在比较匹配时，OCxM 的值不同，输出通道 x 信号 OCx 的操作不同：
 - ◆ OCxM = 000：OCx 信号保持它的电平
 - ◆ OCxM = 001：OCx 信号被设置成有效电平
 - ◆ OCxM = 010：OCx 信号被设置成无效电平
 - ◆ OCxM = 011：OCx 信号进行翻转
2. 匹配时状态寄存器中的标志位置 1（TIM1_SR 寄存器中的 CCxIF 位）。
3. 当配置了 TIM1_DIER 寄存器中的 CCxIE =1，匹配时则产生一个中断。
4. 当配置了 TIM1_DIER 寄存器中的 CCxDE =1，匹配时则产生一个 DMA 请求（仅适用于有内置 DMA 的产品）。

比较输出模式也可以用来输出一个单脉冲（单脉冲输出模式）。例如，通道 1 的比较输出模式的配置步骤如下：

1. 配置计数器的时钟（选择时钟源，配置预分频系数）。
2. 配置 TIM1_ARR 和 TIM1_CCR1 寄存器。
3. 配置 TIM1_DIER 寄存器的 CC1IE =1，使能比较 1 中断。
4. 配置输出模式：
 - ◆ 配置 TIM1_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIM1_CCMR1 寄存器的 OC1PE = 0，禁止 TIM1_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIM1_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIM1_CCER 寄存器的 CC1E = 1，开启输出/比较 1 输出使能，OC1 信号输出到对应的输出引脚。
5. 配置 TIM1_CR1 寄存器的 CEN =1，启动计数器。

当配置 TIM1_CCMRx 寄存器中 OCxPE=0，禁止 TIM1_CCRx 寄存器的预装载功能时，可以随时写入 TIM1_CCRx 寄存器，并且写入的值立即生效。当配置 TIM1_CCMRx 寄存器中 OCxPE=1，启用 TIM1_CCRx 寄存器的预装载功能时，读写仅对预装载寄存器进行操作，TIM1_CCRx 预装载寄存器的值在下次更新事件到来时生效。下图给出了一个例子。

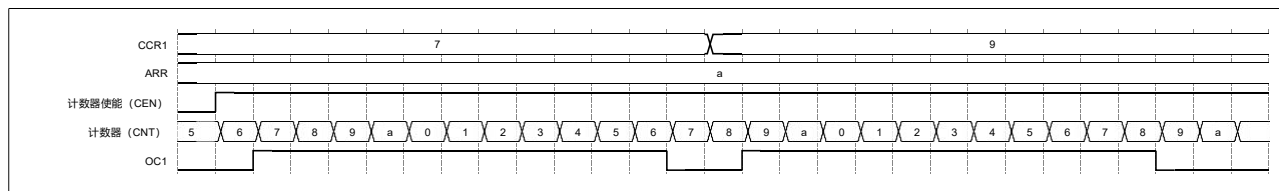


图 11-16 比较输出模式，OC1 信号在匹配时翻转

注：比较输出模式下，更新事件不会对输出结果产生影响。强制输出模式下，在 TIM1_CCRx 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断和 DMA 请求，仍会产生对应的中断和 DMA 请求。

11.4.3.3 PWM 输出

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

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

配置 TIM1_CCER 寄存器的 CCxP 选择 OCx 的有效极性。配置 TIM1_CCER 寄存器的 CCxE、CCxNE 位和 TIM1_BDTR 寄存器的 MOE、OSS1、OSSR 位控制 OCx 的输出使能。配置 TIM1_CR1 寄存器的 CMS 位，可以选择产生边沿对齐或中央对齐的 PWM 信号。

- CMS=00，边沿对齐模式，再进一步配置 DIR，选择递增或递减计数模式。
- CMS=01，中央对齐模式 1。
- CMS=10，中央对齐模式 2。
- CMS=11，中央对齐模式 3。

11.4.3.3.1 PWM 边沿对齐模式——递增计数模式

在递增计数模式配置的基础上，配置 OCxM=110，选择 PWM 模式 1，当 TIM1_CNT < TIM1_CCRx 时通道 x (OCxREF) 为有效电平，否则为无效电平。如果 TIM1_CCRx 中的比较值大于自动重载值 (TIM1_ARR)，则 OCxREF 保持为有效电平。如果比较值为 0，则 OCxREF 保持为无

效电平。下图为 CCR1=0x1, CCR2=0x4, CCR3=0x7, CCR4=0xb, ARR=0xa 时边沿对齐递增计数时 PWM 模式 1 的波形实例。

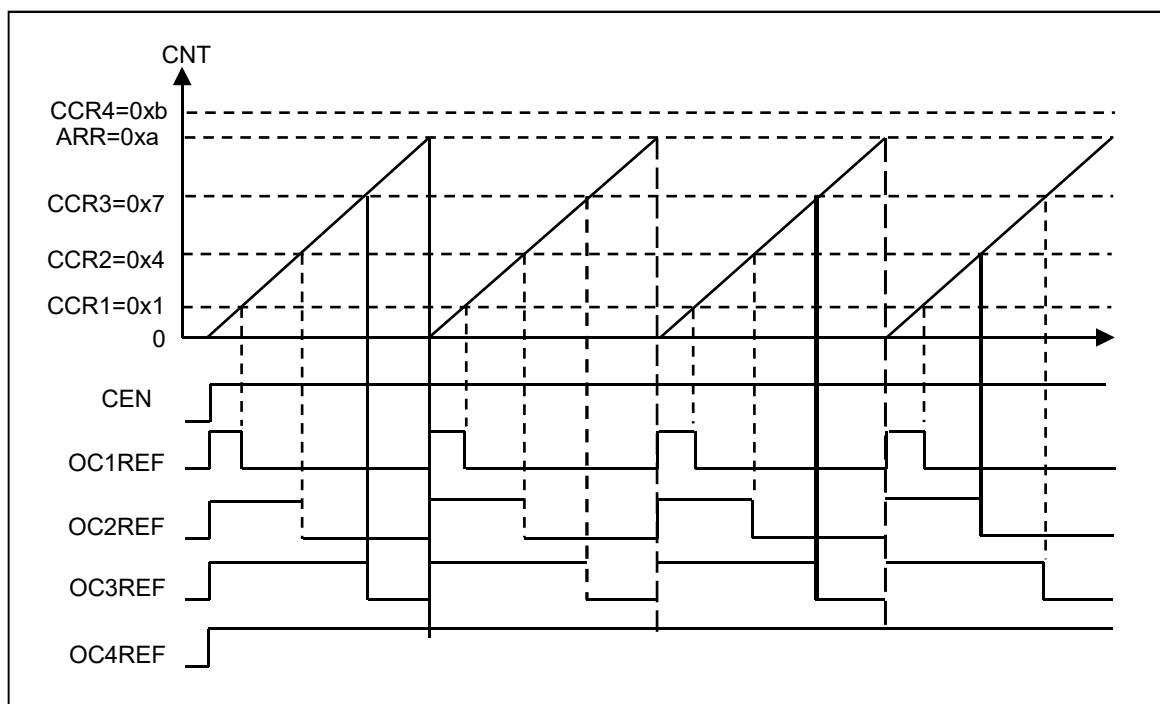


图 11-17 边沿对齐递增计数时 PWM 模式 1 的波形

11.4.3.3.2 PWM 边沿对齐模式——递减计数模式

在递减计数模式配置的基础上，配置 OCxM=110，选择 PWM 模式 1，当 TIM1_CNT > TIM1_CCRx 时通道 x (OCxREF) 为无效电平，否则有效电平。下图为 CCR1=0x4, CCR2=0x6, CCR3=0x9, CCR4=0xb, ARR=0xa 时边沿对齐递减计数时 PWM 模式 1 的波形实例。

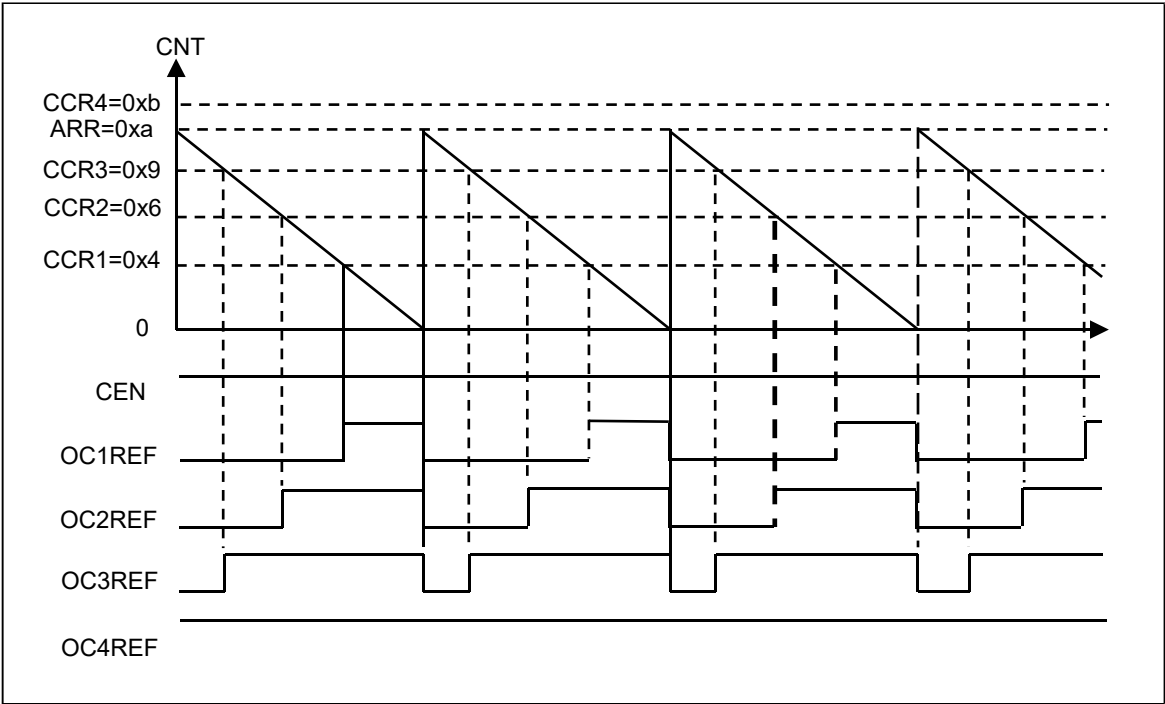


图 11-18 边沿对齐递减计数时 PWM 模式 1 的波形

11.4.3.3.3 PWM 中央对齐模式

首先配置 TIM1 计数器为中央对齐计数模式，根据配置不同的 CMS，比较输出中断标志位在计数器递减计数时被设置 (CMS=01)、在计数器递增计数时被设置 (CMS=10)、或在计数器递增或递减计数时被设置 (CMS=11)。下图为 CCR1=0x4, CCR2=0x6, CCR3=0x9, CCR4=0xb, ARR=0xa 时中央对齐 PWM 模式 1 的波形实例。

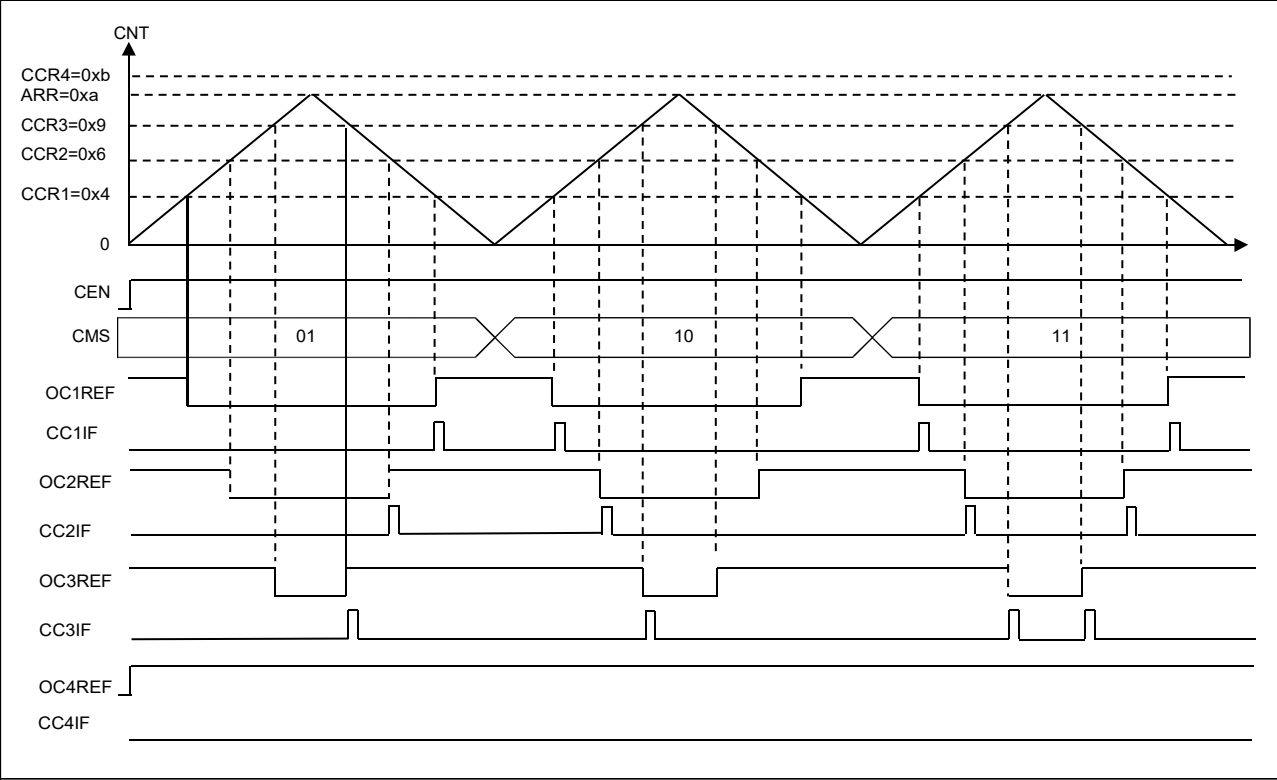


图 11-19 中央对齐 PWM 模式 1 的波形

注：

- 进入中央对齐模式时，使用当前的递增/递减计数配置，计数方向取决于当前的 DIR 的值。
- 在中央对齐模式下，最好不要修改计数器的值，可能会产生不可预知的结果。当计数器处于递增计数时，写入计数器的值 > TIM1_ARR，计数器会继续递增计数。直接写入 0 或 ARR，会立即更新计数方向，但不会产生更新事件。
- 建议使用中央对齐模式时，在启动计数器之前配置 TIM1_EGR 寄存器的 UG=1，产生一个软件更新，更新所有寄存器，启动计数器后不要修改计数器的值。

11.4.3.3.4 PWM 中央对齐模式下移相功能

设置 PDER 寄存器（通道 x 输出 PWM 移相使能位）和 CCRxFALL 寄存器（通道 x 在 PWM 中央对齐模式递减计数时的比较值），可以实现各通道输出 PWM 移相。根据需要移动相位，配置 CCRxFALL 以及 CCRx，即可实现 PWM 输出可编程的移相波形，可左移或是右移。

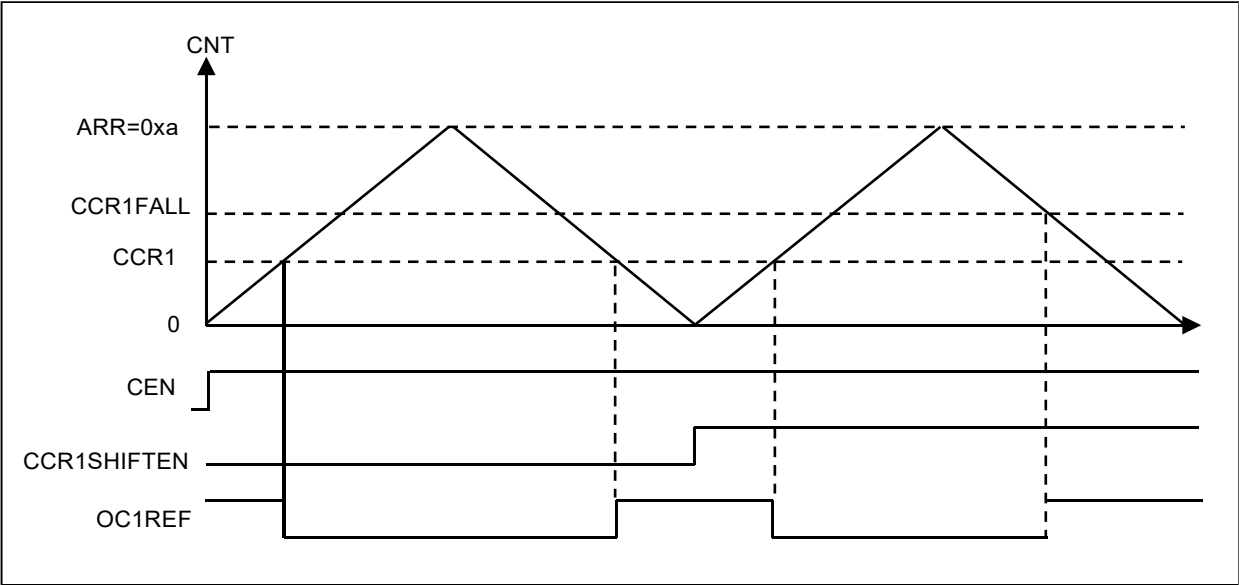


图 11-20 移相功能示意图

11.4.3.3.5 六步 PWM 输出

通过配置 OCxM 选择输出模式，CCxE=1 和 CCxNE=1 打开通道 x 和互补通道的输出使能，可以在通道 x 产生互补输出，这几个功能位为预装载位，在发生 COM 换相事件时，这些预装载位被装载到对应的影子寄存器中。这样可以在写入这些位时不会影响现在的输出，并可以同时载入所有通道配置。配置 TIM1_EGR 寄存器的 COMG=1 或在 TRGI 上升沿都可以产生 COM 事件。

发生 COM 事件时，COM 中断标记会被硬件置 1；当配置了 TIM1_DIER 寄存器 COMIE=1 和 COMDE = 1，发生 COM 事件会产生一个 COM 中断和一个 DMA 请求（产生 DMA 请求仅适用于有内置 DMA 的产品）。

下图显示当发生 COM 事件时，不同配置下 OCx 和 OCxN 输出。

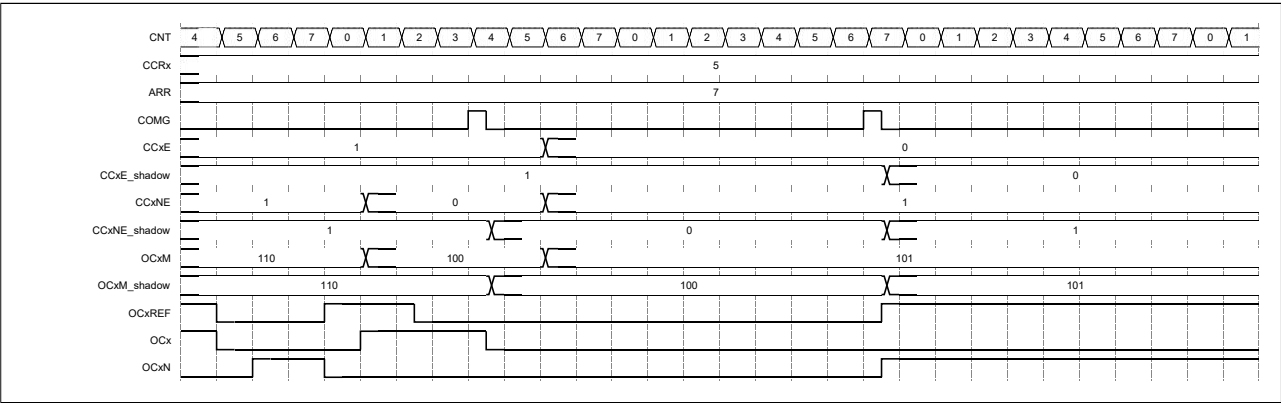


图 11-21 使用 COM 事件产生六步 PWM（OSSR = 1）

11.4.3.3.6 触发源输出

PWM 模式下，CCx_SETTRGO 信号可以用于触发 ADC 等模块。本章节仅介绍 CCx_SETTRGO 信号的触发逻辑，详细触发源选择，触发边沿选择等信息请参考 ADC 章节。

边沿对齐模式下，每次比较匹配（TIMx_CNT 当前计数值等于 TIMx_CCRx）时，CCx_SETTRGO 发生一次翻转。下图为边沿对齐递增计数模式下 CCx_SETTRGO 输出示例。

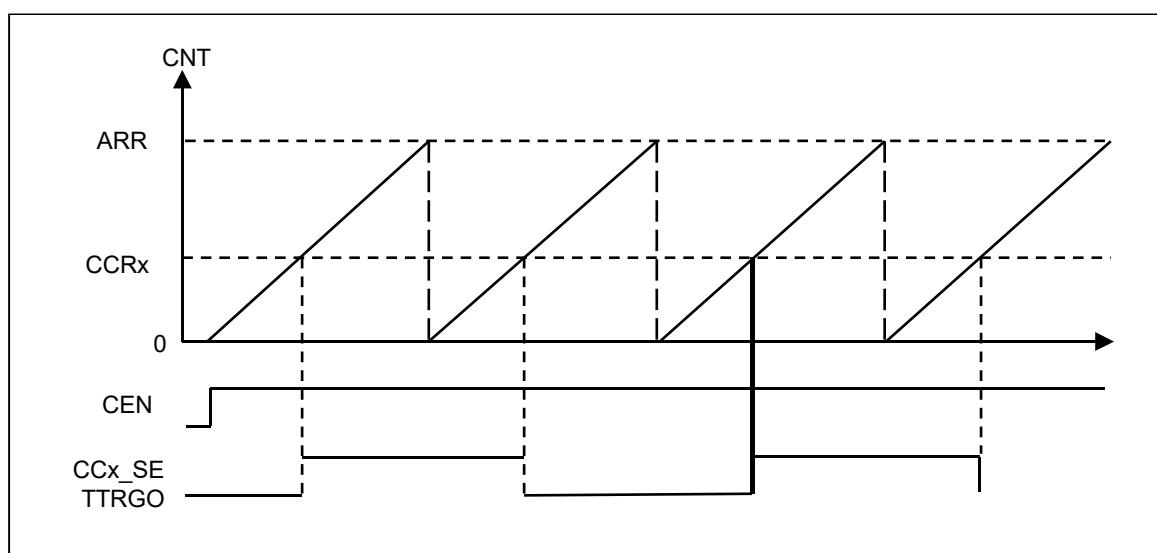


图 11-22 边沿对齐递增计数模式下 CCx_SETTRGO 输出示例

中央对齐模式 1 下，在递减计数周期比较匹配时 CCx_SETTRGO 发生一次翻转。中央对齐模式 2 下，在递增计数周期比较匹配时 CCx_SETTRGO 发生一次翻转。中央对齐模式 3 下，在递增计数周期或递减计数周期比较匹配时 CCx_SETTRGO 发生一次翻转。下图为中央对齐模式下 CCx_SETTRGO 输出示例。

注：使用移相模式时，递减计数比较匹配目标为 CCRxFALL。

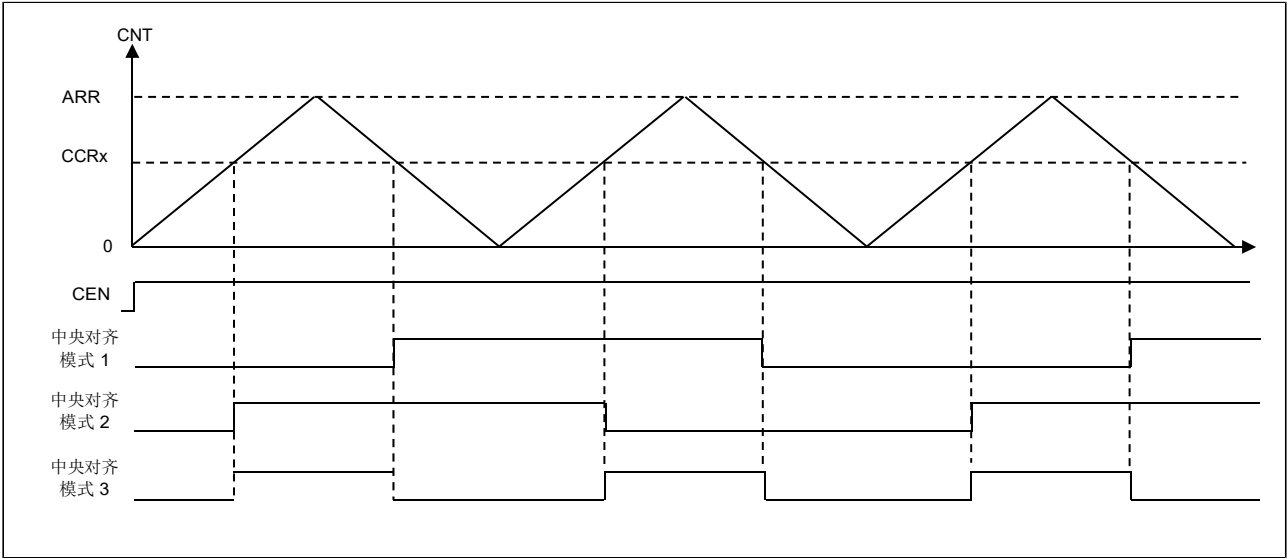


图 11-23 中央对齐模式下 CCx_SETTRGO 输出示例

11.4.3.4 互补输出和死区插入

OCx 和 OCxN 是一对互补输出通道，TIM1 通道 1/2/3 能够输出三路可以管理瞬时关断和瞬时接通的互补信号，同时具有可调的死区时间。用户根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

TIM1_BDTR 寄存器 DTG[7: 0]位定义了插入互补输出之间的死区持续时间，具体计算方式如下表：

表 11-1 死区时间计算

DTG[7: 5]	DT
0xx	$DT = DTG[7: 0] \times T_{dtg} \text{ (} T_{dtg} = T_{DTS} \text{)}$
10x	$DT = (64 + DTG[5: 0]) \times T_{dtg} \text{ (} T_{dtg} = 2 \times T_{DTS} \text{)}$
110	$DT = (32 + DTG[4: 0]) \times T_{dtg} \text{ (} T_{dtg} = 8 \times T_{DTS} \text{)}$
111	$DT = (32 + DTG[4: 0]) \times T_{dtg} \text{ (} T_{dtg} = 16 \times T_{DTS} \text{)}$

例如，如果 $T_{DTS}=125\text{ns}$ ，可能的死区时间为：

- 若步长时间为 125ns，死区时间为 0 至 15875ns。
- 若步长时间为 250ns，死区时间为 16μs 至 31750ns。
- 若步长时间为 1μs，死区时间为 32μs 至 63μs。
- 若步长时间为 2μs，死区时间为 64μs 至 126μs。

当不存在刹车电路时，同时配置 CCxE=1 和 CCxNE=1，开启死区插入，否则还需要配置 MOE=1。配置 TIM1_CCER 寄存器的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性（主输出 OCx 或互补输出 OCxN）。

通过配置 TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，TIM1_BDTR 和 TIM1_CR2 寄存器中的

MOE、OISx、OISxN、OSSI 和 OSSR 位的不同组合可以控制互补信号 OCx 和 OCxN 的输出。具体的组合控制配置见本章表 2、表 3、表 4 和表 5 的互补输出通道 OCx 和 OCxN 的控制位。

例：OCx 和 OCxN 都为高有效，PWM 模式下，发生匹配时，输出参考信号 OCxREF 信号翻转，输出信号 OCx 与参考信号相同，但是 OCx 信号的上升沿对于参考信号的上升沿有一个延时；互补输出信号 OCxN 与参考信号相反，OCxN 信号的上升沿对于参考信号的下降沿同样有一个延时。

注：死区时间不能大于或等于 OCx 或 OCxN 信号的占空比，否则 OCx 或 OCxN 信号一直为无效值。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。

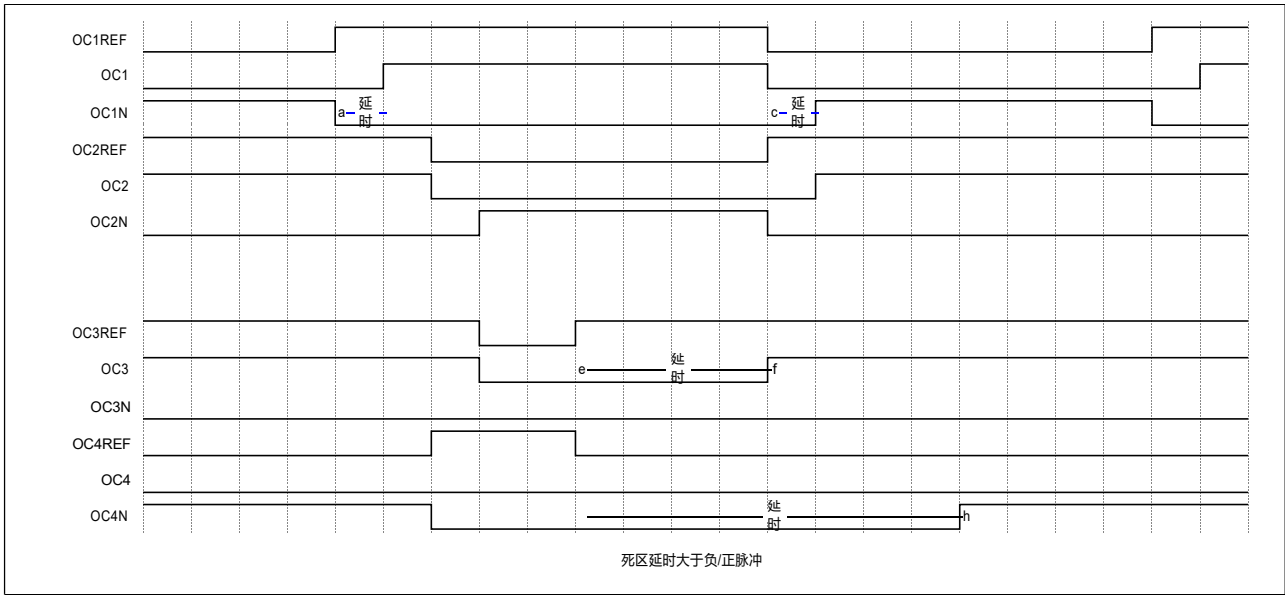


图 11-24 死区插入

11.4.3.5 刹车功能

TIM1 的刹车源有引脚输入、时钟失效事件、和比较器输出三种类型。时钟失效事件由复位时钟控制器中的时钟安全系统产生。

使用刹车功能时，OCx 和 OCxN 输出信号电平被以下功能位组合控制：TIM1_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIM1_CR2 寄存器中的 OISx 和 OISxN 位。发生刹车事件时，OCx 和 OCxN 输出不能同时有效。具体的带刹车功能的互补输出通道 OCx 和 OCxN 的输出状态如下列表格所示。

表 11-2 当 MOE=1，OSSI=0/1，OSSR=0 时：

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0, OCx_EN=0	OCxN=0, OCxN_EN=0
0	1	OCx=0, OCx_EN=0	OCxN=OCxREF+Polarity, OCxN_EN=1
1	0	OCx=OCxREF+Polarity, OCx_EN=1	OCxN=0, OCxN_EN=0
1	1	OCx=OCxREF+Polarity+死区	OCxN=OCxREF 反相+Polarity+

CCxE	CCxNE	OCx	OCxN
		时间, OCx_EN=1	死区时间, OCxN_EN=1

表 11-3 当 MOE=1, OSSI=0/1, OSSR=1 时:

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0, OCx_EN=0	OCxN=0, OCxN_EN=0
0	1	OCx=CCxP, OCx_EN=1	OCxN=OCxREF+Polarity, OCxN_EN=1
1	0	OCx=OCxREF+Polarity, OCx_EN=1	OCxN=CCxNP, OCxN_EN=1
1	1	OCx=OCxREF+Polarity+死区时间, OCx_EN=1	OCxN=OCxREF 反相+Polarity+死区时间, OCxN_EN=1

表 11-4 当 MOE=0, OSSI=0, OSSR=0/1 时:

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=0, OCxN_EN=0 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
0	1	OCx_EN=0, OCxN_EN=0 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
1	0	OCx_EN=0, OCxN_EN=0 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
1	1	OCx_EN=0, OCxN_EN=0 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	

表 11-5 当 MOE=0, OSSI=1, OSSR=0/1 时:

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=1, OCxN_EN=1 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
0	1	OCx_EN=1, OCxN_EN=1 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
1	0	OCx_EN=1, OCxN_EN=1 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	
1	1	OCx_EN=1, OCxN_EN=1 异步的: OCx=CCxP, OCxN=CCxNP 若时钟存在: 经过一个死区时间后, OCx=OISx, OCxN=OISxN OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	

注: 当通道的输出和互补输出都关闭时, OISx, OISxN, CCxP 和 CCxNP 都必须配置为 0。

系统复位后，MOE=0，刹车功能禁止，需要配置 TIM1_BKINF 寄存器的 BKIN_SEL，选择刹车源，支持选择多路刹车源，任意一路刹车信号有效都会触发刹车。配置 TIM1_BKINF 寄存器的 BKINFE，选择刹车信号滤波功能是否有效。配置 TIM1_BKINF 寄存器的 BKINF，选择刹车数字滤波的采样频率。更改刹车数字滤波采样频率前应先关闭刹车滤波功能。配置 TIM1_BDTR 寄存器的 BKE=1，使能刹车功能信号。配置 TIM1_BDTR 寄存器的 BKP 位选择刹车输入信号的极性。BKP 和 BKE 可以同时写入，且会在一个时钟周期后生效。

由于 MOE 被异步清除，因此在实际信号和同步控制位间插入了一个再同步电路，用于在同步信号和异步信号间产生延迟（当 MOE 状态为 0 时写入 1，写入后读取前需要插入一个空指令用于延时，否则无法保证正确读取）。

发生刹车事件时，MOE 被异步清零，此时根据 OSS1 的配置 OCx/OCxN 的输出将置于无效状态、空闲状态或复位状态；MOE=0 时，输出由 TIM1_CR2 寄存器的 OISx 位决定，OSS1=0 时，定时器关闭输出使能，否则打开输出使能。当使用互补输出时，输出首先置于复位状态，然后死区重新生成，在死区之后输出电平由 OISx 和 OISxN 决定。

配置 TIM1_DIER 寄存器的 BIE=1，当发生刹车事件时，产生一个刹车中断；配置 TIM1_BDTR 寄存器的 AOE = 1，则在下一个更新事件到来时自动置位 MOE 位。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能（自动地或者通过软件）设置 MOE，并且状态标志 BIF 不能被清除。

刹车电路中实现了写保护以保证应用程序的安全，允许用户锁住死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性等参数。通过 TIM1_BDTR 寄存器中的 LOCK 位，可以选择 lock 等级（总共三级 lock）。Lock 在系统复位后只能修改一次。

下图显示响应刹车的输出实例：

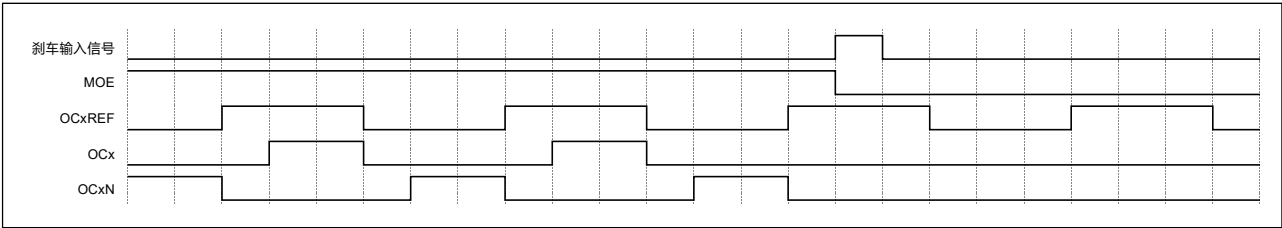


图 11-25 响应刹车的输出（OISx=0，OISxN=0）

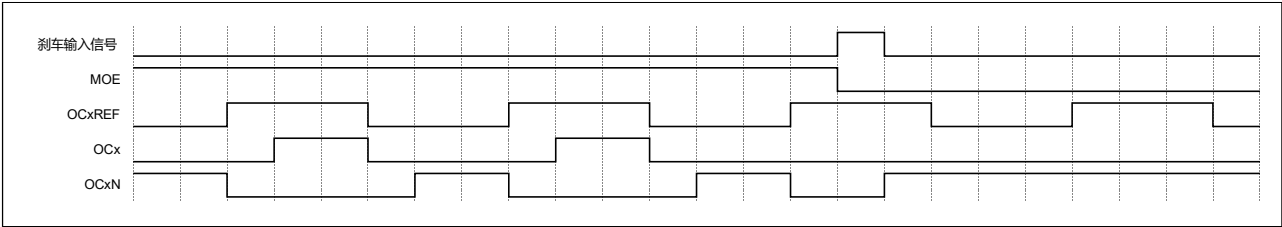


图 11-26 响应刹车的输出（OISx=0，OISxN=1）

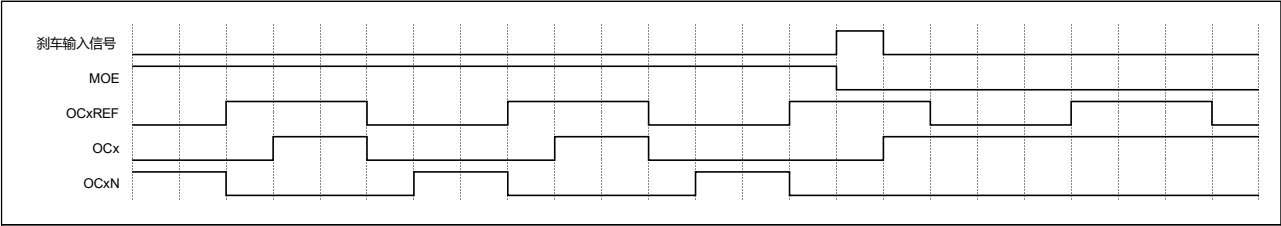


图 11-27 响应刹车的输出（OISx=1，OISxN=0）

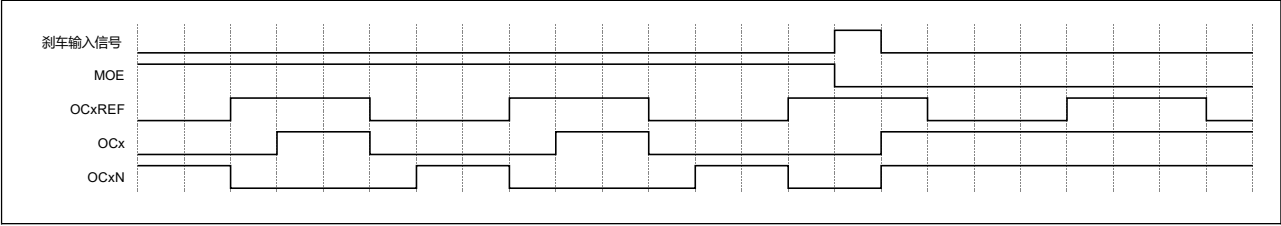


图 11-28 响应刹车的输出（OISx=1，OISxN=1）

11.4.3.6 外部事件清除 OCxREF

在配置 TIM1_CCMR 寄存器的 OCxCE=1 时，OCxREF 可以被 ETR 输入端的有效电平拉低直到发生下一次更新事件（UEV）。此功能只能用于比较输出模式和 PWM 模式，不能用于强制输出模式。

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

- 1. 配置 TIM1_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
- 2. 配置 TIM1_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
- 3. 配置 TIM1_SMCR 寄存器 ETF[3: 0]和 ETP，配置 ETR 信号的触发极性和滤波宽度。 下图显示了当 ETR 输入变化触发 ETRF 为高时，对应不同 OCxCE 的值，OCxREF 信号的动作（PWM 模式）。

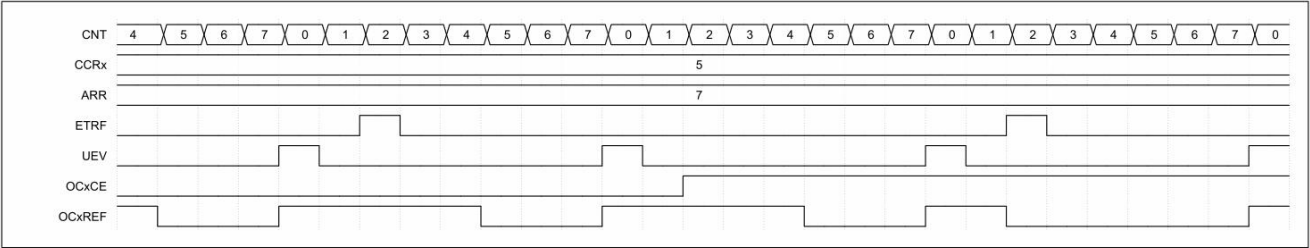


图 11-29 外部事件清除 OCxREF

11.4.3.7 单脉冲输出

单脉冲模式（OPM）下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 TIM1_CR1 寄存器的 OPM=1，选择单脉冲模式，触发信号有效沿或配置 CEN=1 都可以启动计数器，直到下个更新事

件发生或配置 CEN=0 时，计数器停止计数。产生脉冲的必要条件是比较值与计数器的初始值不同。所以在计数器启动之前的必要配置如下：

- 递增计数方式：计数器 $CNT < CCRx \leq ARR$ 。
- 递减计数方式：计数器 $CNT > CCRx$ 。

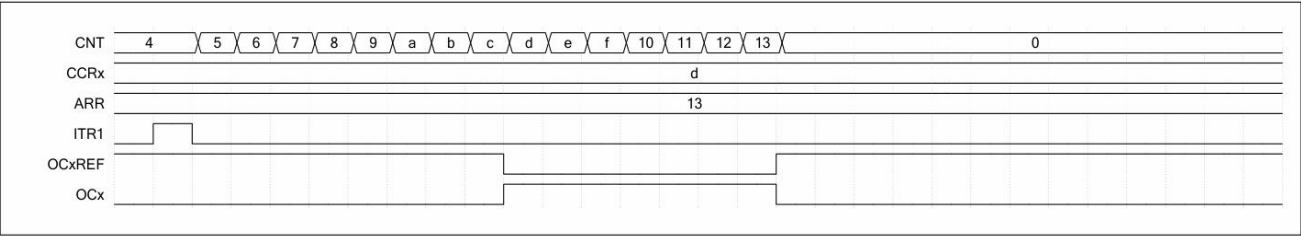


图 11-30 单脉冲模式

例如，在 ITR1 检测到上升沿，延迟 t_{DELAY} 之后，在 OC2 上产生一个长度为 t_{PULSE} 的正脉冲。配置 ITR1 作为触发源：

1. 配置 TIM1_SMCR 寄存器中的 TS = 001，ITR1 作为从模式控制器的触发（TRGI）。
2. 配置 TIM1_SMCR 寄存器中的 SMS = 110，选择触发模式，ITR1 使能计数器工作。

OPM 的波形由 TIM1_ARR 和 TIM1_CCR1 决定（要考虑时钟频率和计数器预分频器）：由 TIM1_CCR1 寄存器的值和 CNT 初始值决定触发信号与单脉冲开始之间的延迟 t_{DELAY} ，TIM1_ARR - TIM1_CCR1 的值为脉冲的宽度 t_{PULSE} 。

下面是一个产生负脉冲的例子，即发生比较匹配时产生从 1 到 0 的波形，计数器达到预装载值时产生一个从 0 到 1 的波形：

1. 配置 TIM1_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
2. 配置 TIM1_CCER 寄存器 CC1P = 1，输出低电平有效。
3. 配置 TIM1_CCMR1 中 OC1PE = 1 和 TIM1_CR1 寄存器中 ARPE=1，使能预装载寄存器。
4. 配置 TIM1_CCR1 寄存器和 TIM1_ARR 寄存器。
5. 配置 TIM1_EGR 寄存器 UG=1 产生一个更新事件。
6. 等待在 ITR1 上的一个外部触发事件。

此例中，TIM1_CR1 寄存器中的 DIR=0、CMS=0、OPM= 1，在下一个更新事件（当计数器从自动装载值返回到 0）时停止计数。

11.4.3.7.1 OCx 快速使能

OCx 快速使能，是单脉冲模式的一种特殊情况。在单脉冲模式下，通过设置 TIM1_CCMR 寄存器的 OCxFE=1，强制 OCxREF 直接响应激励而不是依赖计数器和比较值之间的比较结果，输出波形和比较匹配时的波形一样。这样可以去除比较的时间，快速输出比较结果。OCx 快速输出使能只在 PWM 模式下生效。

11.4.4 从模式

11.4.4.1 复位模式

配置 TIM1_SMCR 寄存器的 SMS=100，从模式选择复位模式。此模式下，TRGI 输入事件会使计数器清零重启。

例如，ITR1 触发计数器重启：

- 1. 配置 TIM1_SMCR 寄存器的 SMS = 100，从模式选择复位模式；配置 TIM1_SMCR 寄存器的 TS = 001，选择 ITR1 作为同步计数器的触发输入。
- 2. 配置 TIM1_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

计数器的时钟源由内部时钟提供，当检测到 ITR1 的上升沿，计数器被清零重启。此时触发器中断标记被硬件置 1。

下图为复位模式下 TIM1_ARR = 0x13 的时序图。

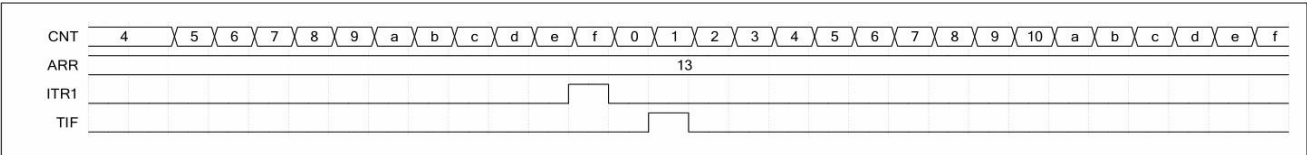


图 11-31 复位模式的控制时序图

11.4.4.2 门控模式

配置 TIM1_SMCR 寄存器 SMS=101，从模式选择门控模式。TRGI 输入为有效电平时，计数器始终开启，否则计数器停止（但不发生复位操作），计数器的开启和停止可控。

例如，计数器只在 ITR1 为高时计数：

- 1. 配置 TIM1_SMCR 寄存器的 SMS=101，从模式选择为门控模式；配置 TIM1_SMCR 寄存器的 TS=001，选择 ITR1 作为同步计数器的触发输入。
- 2. 配置 TIM1_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

计数器的时钟源由内部时钟提供，当检测到 ITR1 的高电平，计数器开始计数，当 ITR1 为低电平时，计数器停止计数。计数器开启或停止都会将 TIF 置 1。

下图为门控模式下 TIM1_ARR=0xf 的时序图。

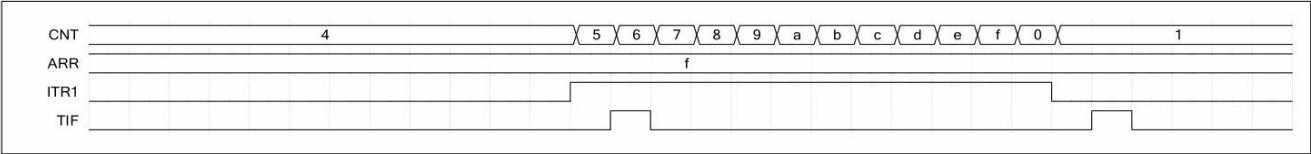


图 11-32 门控模式下的控制时序图

11.4.4.3 触发模式

配置 TIM1_SMCR 寄存器 SMS=110，从模式选择触发模式。TRGI 输入为有效边沿时，计数器开始计数。计数器的启动可控，停止不可控。

例如，计数器在 ITR1 的上升沿开始计数：

1. 配置 TIM1_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIM1_SMCR 寄存器的 TS=001，选择 ITR1 作为计数器的触发输入。
2. 配置 TIM1_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。计数器的时钟源由内部时钟提供，当检测到 ITR1 的上升沿，计数器开始计数。下图为触发模式下 TIM1_ARR=0xf 的时序图。

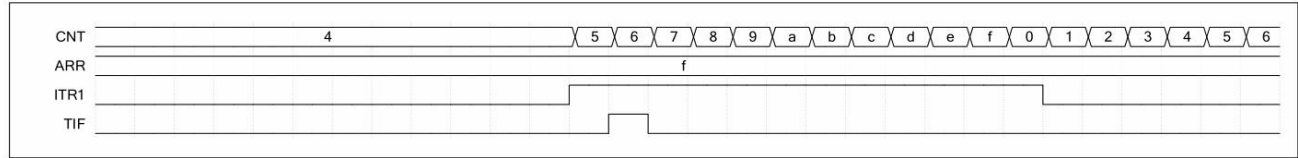


图 11-33 触发器模式下的控制时序图

11.4.4.4 外部时钟模式 2+从模式

当时钟源选择外部时钟模式 2、ETR 信号被用作外部时钟的输入时，可以与从模式一起使用。这种使用方式时，从模式仅支持复位模式、门控模式、触发模式，不支持外部时钟模式 1。

例如，从模式选择触发模式，计数器在 ETR 的每一个上升沿计数一次：

1. 配置 TIM1_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIM1_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIM1_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIM1_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。
2. 配置 TIM1_SMCR 寄存器的 SMS = 110，从模式选择为触发模式。配置 TIM1_SMCR 寄存器的 TS = 001，选择 ITR1 作为输入源。
3. 配置 TIM1_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。计数器在 ITR1 的上升沿开始计数，并将 TIF 置 1。ETR 信号的上升沿和计数器实际计数时钟间的延时取决于 ETR 输入端的同步电路设计。

下图为外部时钟模式 2+从模式（触发模式）下 TIM1_ARR=13 时的时序图。

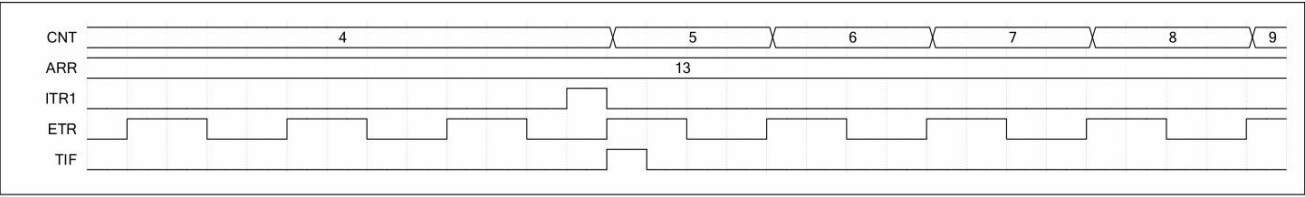


图 11-34 外部时钟模式 2+从模式（触发模式）控制时序图

11.4.5 定时器同步

不同的 TIM1 定时器在内部连接，可以实现定时器之间的级联或同步。 定时器间的同步互联示意图如下：

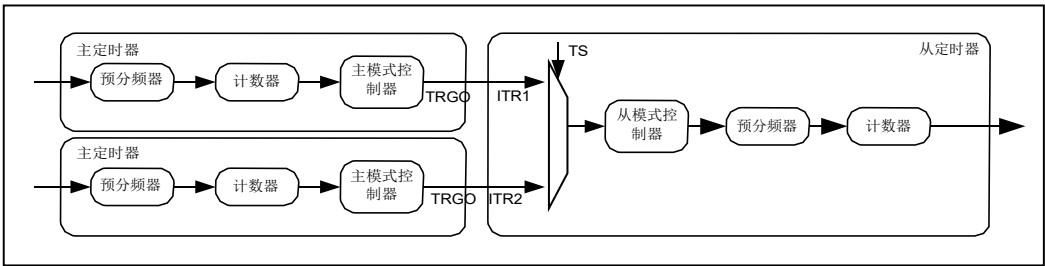


图 11-35 定时器间互联

以下是几种典型的互联应用。

使用一个定时器作为另一个定时器的预分频器 例：主定时器作为从定时器的预分频器，时序图如下图所示：

- 1. 配置主定时器 CR2 寄存器的 MMS=010，主定时器的更新事件作为触发输出（TRGO），主定时器在每次更新事件时输出一个周期信号。
- 2. 配置主定时器 ARR 寄存器，作为主定时器的输出周期。
- 3. 配置从定时器 SMCR 寄存器的 TS，选择从定时器的触发源为主定时器 TRGO。
- 4. 配置从定时器 SMCR 寄存器的 SMS=111，从模式选择外部时钟模式 1。
- 5. 配置主定时器 CR1 寄存器的 CEN=1，启动主定时器。
- 6. 配置从定时器 CR1 寄存器的 CEN=1，启动从定时器。

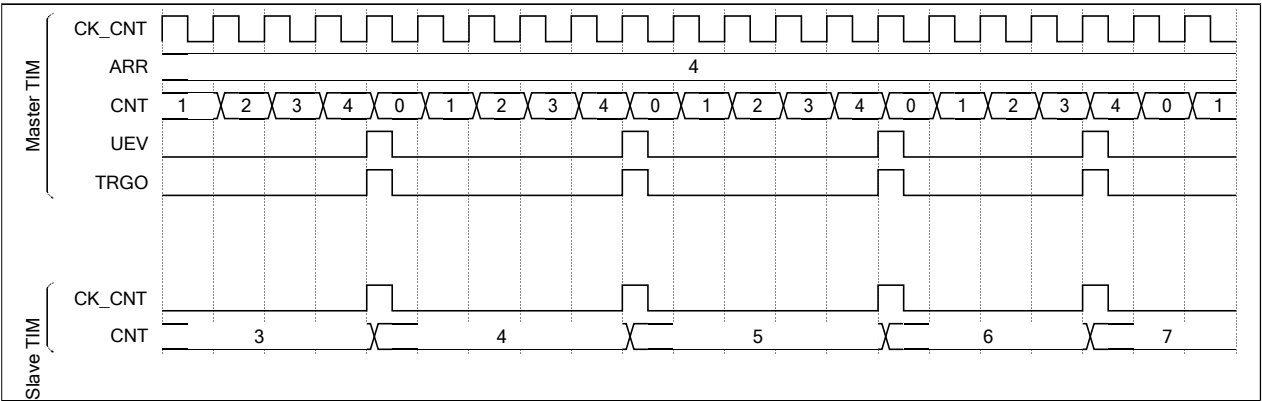


图 11-36 使用主定时器作为从定时器的预分频器 使用一个定时器使能另一个定时器 例：使用主定时器使能从定时器，时序图如下图所示：

当主定时器的 OC1REF 为高时，从定时器的计数器才开始计数。两个 TIM 的计数时钟为 CK_CNT 的三分频。具体配置如下：

- 1. 配置主定时器 CR2 寄存器的 MMS=100，选择主定时器的比较输出参考信号（OC1REF）作为触发输出（TRGO）。
- 2. 配置主定时器 CCR1 寄存器、ARR 寄存器、CCMR1 寄存器的 OC1M 位等相关控制位，配置主定时器输出信号 TRGO 的输出波形。
- 3. 配置从定时器 SMCR 寄存器的 TS，选择主定时器的 OC1REF 作为从定时器的触发输入。
- 4. 配置从定时器 SMCR 寄存器的 SMS=101，选择从定时器为门控模式。
- 5. 配置从定时器 CR1 寄存器的 CEN=1，启动从定时器。
- 6. 配置主定时器 CR1 寄存器的 CEN=1，启动主定时器。

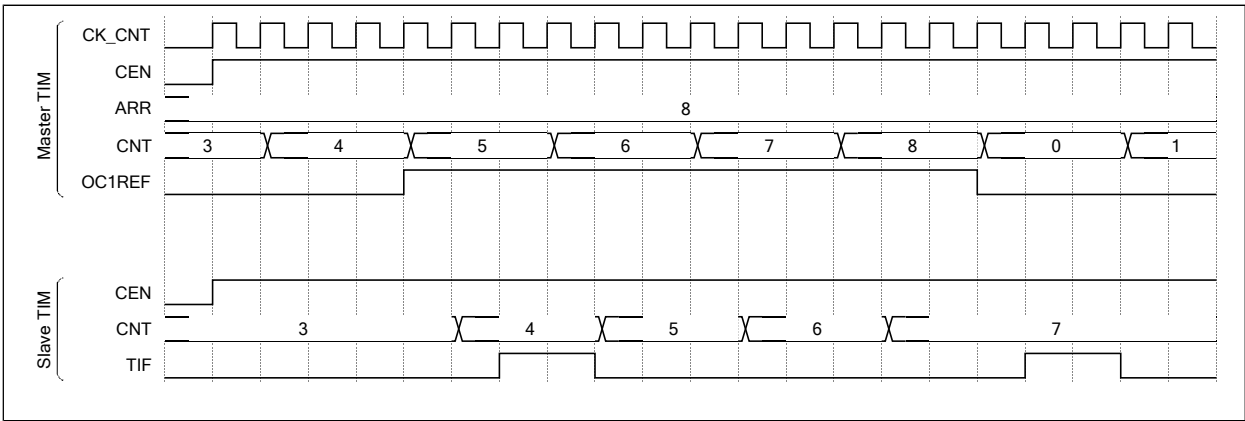


图 11-37 使用主定时器使能从定时器 使用一个定时器启动另一个定时器 例：使用主定时器的更新事件启动从定时器，时序图如下图所示：

当主定时器产生更新事件时，从定时器接收到触发信号，从定时器的 CEN 由硬件自动置 1，从定时器的计数器开始计数。两个 TIM 的计数时钟为 CK_CNT 的三分频。具体配置如下：

- 1. 配置主定时器 CR2 寄存器的 MMS=010，选择主定时器的更新事件为触发输出（TRGO）；
- 2. 配置主定时器 ARR 寄存器，作为更新事件产生的周期；
- 3. 配置从定时器 SMCR 寄存器的 TS，选择主定时器的 TRGO 作为从定时器的触发输入；
- 4. 配置从定时器 SMCR 寄存器的 SMS=110，选择从定时器为触发模式；
- 5. 配置主定时器 CR1 寄存器的 CEN=1，启动主定时器。

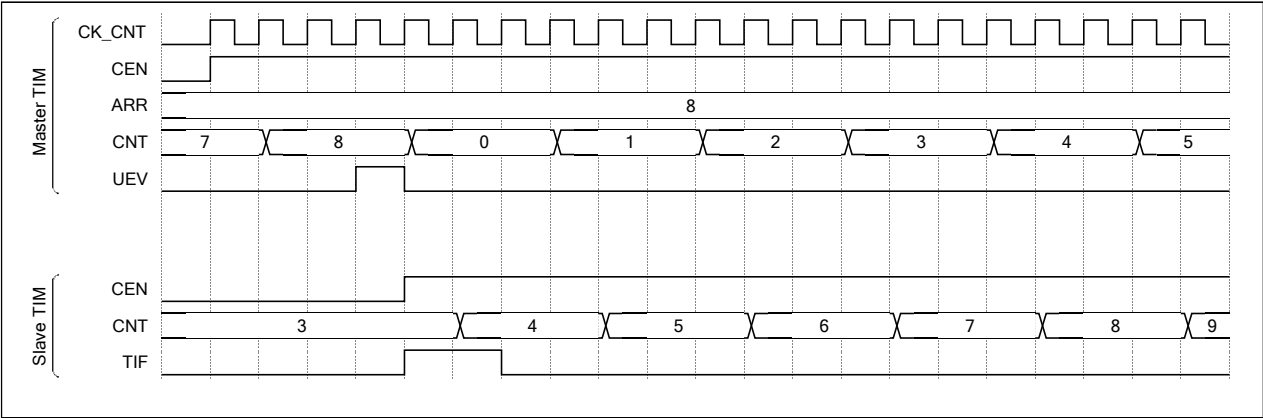


图 11-38 使用主定时器的更新事件启动从定时器

使用一个外部触发同步启动两个定时器

例：使用主定时器的 **ETR** 上升沿，启动主定时器的同时启动从定时器，时序图如下图所示：

为了确保两个定时器同时开启，主定时器必须在主/从模式下配置。具体配置如下：

1. 配置主定时器 **CR2** 寄存器的 **MMS=001**，作为主模式时，将主定时器的使能信号 **CEN** 作为触发输出（**TRGO**）。
2. 配置主定时器 **SMCR** 寄存器的 **TS=111**，设置主定时器作为从模式时，将 **ETR** 作为触发输入。
3. 配置主定时器 **SMCR** 寄存器的 **SMS=110**，选择主定时器为触发模式。
4. 配置从定时器 **SMCR** 寄存器的 **TS**，选择主定时器的触发输出作为从定时器的触发输入。
5. 配置从定时器 **SMCR** 寄存器的 **SMS=110**，选择从定时器为触发模式。

当主定时器的 **ETR** 出现上升沿时，两个定时器同步启动（按照内部时钟），计数器开始计数，两个定时器的 **TIF** 标志也同时置 1。

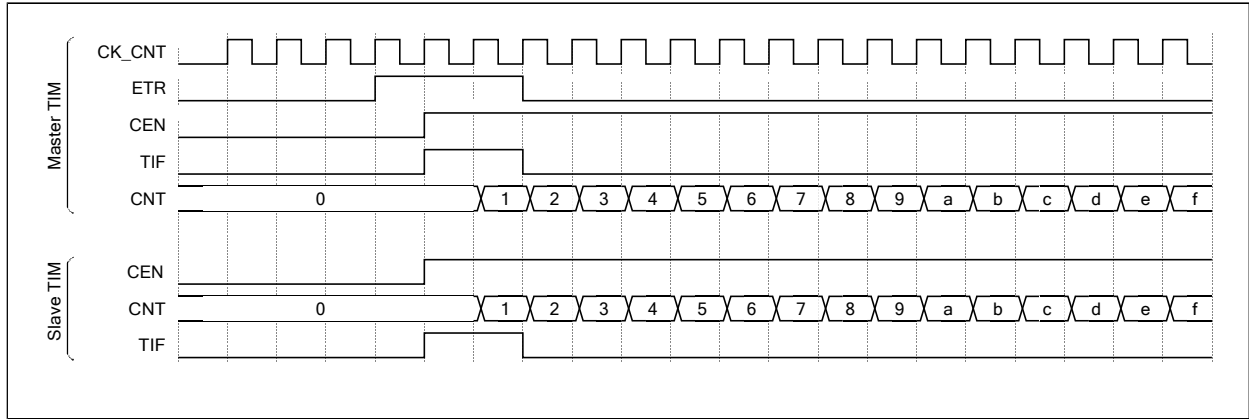


图 11-39 主定时器的 T11 同步启动主定时器和从定时器

11.4.6 调试模式

在调试模式下，配置 **DBG_CR** 寄存器中 **DBG_TIM1_STOP=1**，TIM1 计数器停止计数。（详见调

试章节)

11.4.7 中断

TIM1 的中断包括：比较 1 中断、比较 2 中断、比较 3 中断、比较 4 中断、比较 5 中断、更新中断、COM 中断、触发中断和刹车中断，当相应的中断使能位打开，发生相应的事件时，产生相应的中断。

表 11-6 中断事件一览表

中断事件	标志位	使能位
比较 1 中断	CC1IF	CC1IE
比较 2 中断	CC2IF	CC2IE
比较 3 中断	CC3IF	CC3IE
比较 4 中断	CC4IF	CC4IE
比较 5 中断	CC5IF	CC5IE
更新中断	UIF	UIE
COM 中断	COMIF	COMIE
触发中断	TIF	TIE
刹车中断	BIF	BIE

11.4.8 DMA

TIM1 能够在发生单个事件时生成一个或连续多个 DMA 请求。主要目的是在没有软件开销的情况下，多次重新编程 TIM1 的一部分寄存器，也可以用于按周期读取数个寄存器。

TIM1_DCR 和 TIM1_DMAR 寄存器跟 DMA 模式相关。DMA 控制器的目标是唯一的，必须指向 TIM1_DMAR 寄存器。开启 DMA 使能后，在给定的 TIM1 事件发生时，TIM1 会给 DMA 发送请求。对 TIM1_DMAR 寄存器的每次写操作都被重定向到一个 TIM1 寄存器。

TIM1_DCR 寄存器的 DBL 位定义了 DMA 连续传送的长度，即传输寄存器数量。当对 TIM1_DMAR 进行读写操作时，定时器识别 DBL，确定需要传输的寄存器数量。TIM1_DCR 寄存器的 DBA 位定义了 DMA 传输的基地址，定义从 TIM1_CR1 寄存器地址开始的偏移量（00000 为 TIM1_CR1、00001 为 TIM1_CR2 00110 为 TIM1_CCMR1 等）。

例：DMA 连续传送模式用于在发生更新事件时更新 CCR1、CCR2、CCR3 寄存器的内容。具体配置如下：

1. 配置相应的 DMA 通道。
2. 配置 TIM1_DCR 寄存器的 DBA=01101，配置 DMA 的基地址，选择偏移地址为 TIM1_CCR1 寄存器的地址。
3. 配置 TIM1_DCR 寄存器的 DBL=00010，配置传输长度为 3。
4. 配置 TIM1_DIER 寄存器的 UDE=1，允许更新事件的 DMA 请求。
5. 配置 TIM1_CR1 寄存器的 CEN=1，启动计数器。
6. 使能 DMA 通道。

此例中发生一次更新事件，DMA 将相应存储器地址中准备好的数据传输到 CCR1、CCR2、CCR3 寄存器中，或将 CCR1、CCR2、CCR3 寄存器的值传输到相应存储器地址中。

11.5 寄存器

表 11-7 TIM1 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIM1_CR1	控制寄存器 1	0x0000
0x04	TIM1_CR2	控制寄存器 2	0x0000
0x08	TIM1_SMCR	从模式控制寄存器	0x0000
0x0C	TIM1_DIER	DMA/中断使能寄存器（DMA 部分仅适用于有内置 DMA 的芯片）	0x0000 0000
0x10	TIM1_SR	状态寄存器	0x0000 0000
0x14	TIM1_EGR	事件产生寄存器	0x0000 0000
0x18	TIM1_CCMR1	比较模式寄存器 1	0x0000
0x1C	TIM1_CCMR2	比较模式寄存器 2	0x0000
0x20	TIM1_CCER	比较使能寄存器	0x0000
0x24	TIM1_CNT	计数器	0x0000
0x28	TIM1_PSC	预分频率器	0x0000
0x2C	TIM1_ARR	自动装载寄存器	0x0000
0x30	TIM1_RCR	重复计数寄存器	0x0000
0x34	TIM1_CCR1	比较寄存器 1	0x0000
0x38	TIM1_CCR2	比较寄存器 2	0x0000
0x3C	TIM1_CCR3	比较寄存器 3	0x0000
0x40	TIM1_CCR4	比较寄存器 4	0x0000
0x44	TIM1_BDTR	刹车和死区寄存器	0x0000 0000
0x48	TIM1_DCR	DMA 控制寄存器（仅适用于有内置 DMA 的芯片）	0x0000
0x4C	TIM1_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000
0x54	TIM1_CCMR3	比较模式寄存器 3	0x0000
0x58	TIM1_CCR5	比较寄存器 5	0x0000
0x5C	TIM1_PDER	PWM 移相/DMA repeat 更新请求使能寄存器	0x0000
0x60 ~ 0x70	TIM1_CCRxFALL	PWM 移相递减计数比较寄存器	0x0000
0x74	TIM1_BKINF	刹车输入滤波寄存器	0x0000 0000

11.5.1 TIM1_CR1 控制寄存器 1

偏移地址：0x00 复位值：

0x0000

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

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（ETR）所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT_CK}$ 01: $t_{DTS} = 2 \times t_{INT_CK}$ 10: $t_{DTS} = 4 \times t_{INT_CK}$ 11: 保留，不要使用这个配置
7	ARPE	自动重载预装载使能（Auto-reload preload enable） 0: 关闭 TIM1_ARR 寄存器的影子寄存器 1: 使能 TIM1_ARR 寄存器的影子寄存器
6: 5	CMS	中央对齐模式选择（Center-aligned mode selection） 00: 边沿对齐模式。计数方向取决于 DIR 位 01: 中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式，只在计数器递减计数时比较中断标志位被置 1 10: 中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式，只在计数器递增计数时比较中断标志位被置 1 11: 中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式，在计数器递增和递减计数时比较中断标志位均被置 1 注：计数过程中，不允许更改此位。
4	DIR	计数方向（Direction） 0: 计数器递增计数 1: 计数器递减计数 注：当计数器配置为中央对齐模式时，该位为只读。
3	OPM	单脉冲模式（One pulse mode） 0: 禁止单脉冲模式，在发生更新事件时，计数器继续计数 1: 使能单脉冲模式，在发生下一次更新事件或软件清除 CEN 位时，计数器停止计数
2	URS	更新请求源（Update request source） 软件配置该位，选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求： - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新（Update disable） 该位用来允许或禁止更新事件的产生 0: 允许更新事件（UEV） 1: 禁止更新事件。不产生更新事件，影子寄存器（ARR、PSC、CCR _x ）保持值不变。如果设置了 EGR_UG 位为 1，或者从模式控制器接收到硬件复位，计数器和预分频器被初始化。
0	CEN	计数器使能（Counter enable） 0: 禁止计数器 1: 使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

11.5.2 TIM1_CR2 控制寄存器 2

偏移地址：0x04 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res.	MMS			CCDS	CCUS	Res.	CCPC

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

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14	OIS4	输出空闲状态 4（OC4 输出）。参见 OIS1 位。
13	OIS3N	输出空闲状态 3（OC3N 输出）。参见 OIS1N 位。
12	OIS3	输出空闲状态 3（OC3 输出）。参见 OIS1 位。
11	OIS2N	输出空闲状态 2（OC2N 输出）。参见 OIS1N 位。
10	OIS2	输出空闲状态 2（OC2 输出）。参见 OIS1 位。
9	OIS1N	输出空闲状态 1（OC1N 输出）（Output Idle state 1） 0：当 MOE = 0 时，死区后 OC1N = 0 1：当 MOE = 0 时，死区后 OC1N = 1 注：已经设置了 LOCK（TIM1_BKR 寄存器）级别 1、2 或 3 后，该位不能被修改。
8	OIS1	输出空闲状态 1（OC1 输出）（Output Idle state 1） 0：当 MOE=0，若 OC1N 有效，则在死区时间后 OC1 = 0 1：当 MOE=0，若 OC1N 有效，则在死区时间后 OC1 = 1 注：已经设置了 LOCK（TIM1_BKR 寄存器）级别 1、2 或 3 后，该位不能被修改。
7	Reserved	保留，必须保持复位值。
6: 4	MMS	主模式选择（Master mode selection） 这些位控制 TRGO 信号的选择，用于选择在主模式下送到从定时器的同步信息： 000：复位 TIM1_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001：使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出（TRGO），计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式。 010：更新 更新事件被选为 TRGO。 011：比较脉冲 通道 1 发生一次比较成功时，触发输出送出一个 TRGO 信号。 100：比较 OC1REF 信号被用于作为触发输出（TRGO） 101：比较 OC2REF 信号被用于作为触发输出（TRGO） 110：比较 OC3REF 信号被用于作为触发输出（TRGO） 111：比较 OC4REF 信号被用于作为触发输出（TRGO）
3	CCDS	DMA 请求源选择（compare DMA selection） 0：当 CCx 发生比较事件时，发送 CCx 的 DMA 请求 1：发生更新事件时，发送 CCx 的 DMA 请求 注：仅适用于有内置 DMA 的产品
2	CCUS	比较控制更新源选择（compare control update selection） 0：CCPC=1 时，只能配置 COMG=1 更新。 1：CCPC=1 时，可以通过配置 COMG=1 或检测到 TRGI 上的一个上升沿更新。 注：此位只在通道为互补输出时有效。
1	Reserved	保留，必须保持复位值。
0	CCPC	比较预装载控制位（compare preloaded control） 0：CCxE，CCxNE 和 OCxM 位预装载禁用 1：CCxE，CCxNE 和 OCxM 位预装载使能 注：此位只在通道为互补输出时有效。

11.5.3 TIM1_SMCR 从模式控制寄存器

偏移地址：0x08 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS		ETF				MSM	TS			OCCS	SMS		
rw	rw	rw		rw				rw	rw			rw	rw		

Bit	Field	Description
15	ETP	外部触发极性 (External trigger polarity) 该位选择 ETR 信号的极性。 0: 高电平或上升沿有效 1: 低电平或下降沿有效 注: 仅适用于支持外部触发的产品
14	ECE	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2。 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2, ETRF 信号上的任意有效沿驱动计数器计数 注 1: 仅适用于支持外部触发的产品。 注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。 注 3: TS ≠ 111 时, 复位模式, 门控模式和触发模式可以与外部时钟模式 2 同时使用。 注 4: 同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入是 ETR。
13: 12	ETPS	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须低于 TIM1 主时钟 PCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8 注: 仅适用于支持外部触发的产品。
11: 8	ETF	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器, 以 f _{DTS} 采样 0001: 采样频率 f _{sampling} =f _{INT_CK} , N=2 0010: 采样频率 f _{sampling} =f _{INT_CK} , N=4 0011: 采样频率 f _{sampling} =f _{INT_CK} , N=8 0100: 采样频率 f _{sampling} =f _{DTS} /2, N=6 0101: 采样频率 f _{sampling} =f _{DTS} /2, N=8 0110: 采样频率 f _{sampling} =f _{DTS} /4, N=6 0111: 采样频率 f _{sampling} =f _{DTS} /4, N=8 1000: 采样频率 f _{sampling} =f _{DTS} /8, N=6 1001: 采样频率 f _{sampling} =f _{DTS} /8, N=8 1010: 采样频率 f _{sampling} =f _{DTS} /16, N=5 1011: 采样频率 f _{sampling} =f _{DTS} /16, N=6 1100: 采样频率 f _{sampling} =f _{DTS} /16, N=8 1101: 采样频率 f _{sampling} =f _{DTS} /32, N=5 1110: 采样频率 f _{sampling} =f _{DTS} /32, N=6 1111: 采样频率 f _{sampling} =f _{DTS} /32, N=8 注: 仅适用于支持外部触发的产品。

Bit	Field	Description
7	MSM	主/从模式 (Master/slave mode) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟, 以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步, 该功能可以把几个定时器同步到一个单一的外部事件。
6: 4	TS	触发选择 (Trigger selection) 触发输入源选择。 000: 内部触发 0 (ITR0) 001: 内部触发 1 (ITR1) 010: 内部触发 2 (ITR2) 011: 内部触发 3 (ITR3) 100: 保留 101: 保留 110: 保留 111: 外部触发输入 (ETR) 注: 从模式使能后这些位不能修改。
3	OCCS	比较输出信号 (OCxREF) 清除选择 (Output compare clear selection) 在 PWM 模式下, 清除比较输出 (OCxREF)。 0: 外部触发信号作为清除信号 1: 比较器 (COMP) 输出作为清除信号 注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。
2: 0	SMS	从模式选择 (Slave mode selection) 当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。 000: 关闭从模式 - 如果 CEN = 1, 则预分频器直接由内部时钟驱动。 001: 保留 010: 保留 011: 保留 100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新事件。 101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器开始计数。当触发输入变为低时, 计数器停止计数 (但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 - 计数器在触发输入 TRGI 的上升沿启动 (但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。

表 11-8 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	-	TIM3_TRGO	TIM17_OC1REF
TIM3	TIM1_TRGO	-	TIM16_OC1REF	TIM14_OC1REF

11.5.4 TIM1_DIER DMA/中断使能寄存器

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
Reserved														CC5DE rw	CC5IE rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res. rw	TDE rw	COMDE rw	CC4DE rw	CC3DE rw	CC2DE rw	CC1DE rw	UDE rw	BIE rw	TIE rw	COMIE rw	CC4IE rw	CC3IE rw	CC2IE rw	CC1IE rw	UIE rw

Bit	Field	Description
31: 18	Reserved	保留，必须保持复位值。
17	CC5DE	允许比较 5 的 DMA 请求 (Compare 5 DMA request enable) 0: 禁止比较 5 的 DMA 请求 1: 允许比较 5 的 DMA 请求 注：仅适用于有内置 DMA 且有通道 5 的产品。
16	CC5IE	允许比较 5 中断 (Compare 5 interrupt enable) 0: 禁止比较 5 中断 1: 允许比较 5 中断
15	Reserved	保留，必须保持复位值。
14	TDE	允许触发 DMA 请求 (Trigger DMA request enable) 0: 禁止触发 DMA 请求 1: 允许触发 DMA 请求 注：仅适用于有内置 DMA 的产品。
13	COMDE	允许 COM 的 DMA 请求 (COM DMA request enable) 0: 禁止 COM 的 DMA 请求 1: 允许 COM 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
12	CC4DE	允许比较 4 的 DMA 请求 (Compare 4 DMA request enable) 0: 禁止比较 4 的 DMA 请求 1: 允许比较 4 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
11	CC3DE	允许比较 3 的 DMA 请求 (Compare 3 DMA request enable) 0: 禁止比较 3 的 DMA 请求 1: 允许比较 3 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
10	CC2DE	允许比较 2 的 DMA 请求 (Compare 2 DMA request enable) 0: 禁止比较 2 的 DMA 请求 1: 允许比较 2 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
9	CC1DE	允许比较 1 的 DMA 请求 (Compare 1 DMA request enable) 0: 禁止比较 1 的 DMA 请求 1: 允许比较 1 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
8	UDE	允许更新 DMA 请求 (Update DMA request enable) 0: 禁止更新 DMA 请求 1: 允许更新 DMA 请求 注：仅适用于有内置 DMA 的产品。
7	BIE	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断 1: 允许刹车中断
6	TIE	允许触发中断 (Trigger interrupt enable) 0: 禁止触发中断 1: 允许触发中断
5	COMIE	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断 1: 允许 COM 中断
4	CC4IE	允许比较 4 中断 (Compare 4 interrupt enable) 0: 禁止比较 4 中断 1: 允许比较 4 中断

Bit	Field	Description
3	CC3IE	允许比较 3 中断 (Compare 3 interrupt enable) 0: 禁止比较 3 中断 1: 允许比较 3 中断
2	CC2IE	允许比较 2 中断 (Compare 2 interrupt enable) 0: 禁止比较 2 中断 1: 允许比较 2 中断
1	CC1IE	允许比较 1 中断 (Compare 1 interrupt enable) 0: 禁止比较 1 中断 1: 允许比较 1 中断
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

11.5.5 TIM1_SR 状态寄存器

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
Reserved															CC5IF
															rw0c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
															r_w0c

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值。
16	CC5IF	比较 5 中断标记 (Compare 5 interrupt flag) 参考 CC1IF 描述。
15: 8	Reserved	保留, 必须保持复位值。
7	BIF	刹车中断标记 (Break interrupt flag) 当刹车输入有效, 由硬件对该位置 1。如果刹车输入无效, 则该位可由软件清 0 0: 无刹车事件产生 1: 刹车输入上检测到有效电平
6	TIF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在 TRGI 输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断产生
5	COMIF	COM 中断标记 (COM interrupt flag) 当产生 COM 事件 (比较控制位 CCxE、CCxNE、OCxM 已被更新) 时该位由硬件置 1。它由软件清 0。 0: 无 COM 事件产生 1: COM 中断产生
4	CC4IF	比较 4 中断标记 (Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	比较 3 中断标记 (Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	比较 2 中断标记 (Compare 2 interrupt flag) 参考 CC1IF 描述。

Bit	Field	Description
1	CC1IF	比较 1 中断标记 (Compare 1 interrupt flag) 当计数器值与比较值匹配时该位由硬件置 1 (在中央对齐模式下根据 TIM1_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIM1_CNT 的值与 TIM1_CCR1 的值匹配
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新 时该位由硬件置 1: - 若 TIM1_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 -若 TIM1_CR1 寄存器的 UDIS=0、URS=0, 当 TIM1_EGR 寄存器的 UG=1 时。 - 若 TIM1_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。

11.5.6 TIM1_EGR 事件产生寄存器

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
Reserved															CC5G
															w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值。
16	CC5G	产生比较 5 事件 (Compare 5 generation) 参考 CC1G 描述。
15: 8	Reserved	保留, 必须保持复位值。
7	BG	产生刹车事件 (Break generation) 0: 无动作 1: 产生一个刹车事件, 此时 MOE=0, BIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA, 由硬件清除。
6	TG	产生触发事件 (Trigger generation) 0: 无动作 1: 产生触发事件, TIM1_SR 寄存器的 TIF = 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA, 由硬件自动清 0。
5	COMG	比较事件, 产生控制更新 (Compare control update generation) 0: 无动作 1: 比较事件控制更新产生, 由硬件自动清 0, 当 CCPC=1, 允许更新 CCxE、CCxNE、OCxM 位。 注: 该位只对拥有互补输出的通道有效。
4	CC4G	产生比较 4 事件 (Compare 4 generation) 参考 CC1G 描述。
3	CC3G	产生比较 3 事件 (Compare 3 generation) 参考 CC1G 描述。

Bit	Field	Description
2	CC2G	产生比较 2 事件 (Compare 2 generation) 参考 CC1G 描述。
1	CC1G	产生通道 1 比较事件 (Compare 1 generation) 该位由软件置 1，用于产生一个比较事件，由硬件自动清 0。 0: 无动作 1: 通道 CC1 上产生一个比较事件： CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
0	UG	产生更新事件 (Update generation) 0: 无动作 1: 初始化计数器，并产生一个更新事件。由硬件自动清 0，如果选择了中央对齐或递增计数模式，计数器被清 0；如果选择递减计数模式，计数器将载入自动重载值。预分频计数器将同时被清除。

11.5.7 TIM1_CCMR1 比较模式寄存器 1

偏移地址：0x18 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	Reserved		OC1CE	OC1M			OC1PE	OC1FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能 (Output compare 2 clear enable) 参考 OC1CE 的描述。
14: 12	OC2M	通道 2 比较输出模式 (Output compare 2 mode) 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能 (Output compare 2 preload enable) 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能 (Output compare 2 fast enable) 参考 OC1FE 的描述。
9: 8	Reserved	保留，必须保持复位值。
7	OC1CE	通道 1 比较输出清 0 使能 (Output compare 1 clear enable) 0: OC1REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入有效电平时，OC1REF 清零

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode)</p> <p>该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。TIM1_CCR1 与 TIM1_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIM1_CNT 的值与 TIM1_CCR1 的值相同时, 强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIM1_CNT 的值与 TIM1_CCR1 的值相同时, 强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIM1_CCR1=TIM1_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为低。强制 OC1REF 为低电平。</p> <p>101: 强制为高。强制 OC1REF 为高电平。</p> <p>110: PWM 模式 1。在递增计数时, 当 TIM1_CNT<TIM1_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIM1_CNT > TIM1_CCR1 时强制 OC1REF 为低电平, 否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时, 当 TIM1_CNT<TIM1_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平; 在递减计数时, 当 TIM1_CNT > TIM1_CCR1 时强制 OC1REF 为高电平, 否则为低电平。</p> <p>注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 时, 该位不能被修改。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable)</p> <p>0: 禁止 TIM1_CCR1 寄存器的预装载功能, 写入 TIM1_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIM1_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1 的预装载值在更新事件到来时生效。</p> <p>注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 并且时, 该位不能被修改。</p> <p>注 2: 若该位置 1, 在单脉冲模式下 (TIM1_CR1 寄存器的 OPM=1), 是否设定预装载寄存器无影响; 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。</p>
2	OC1FE	<p>通道 1 比较输出快速使能 (Output compare 1 fast enable)</p> <p>该位为 1 时, 若通道配置为 PWM 模式, 会加快比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。</p> <p>0: 禁止通道 1 比较输出快速使能</p> <p>1: 开启通道 1 比较输出快速使能</p>
1: 0	Reserved	保留, 必须保持复位值。

11.5.8 TIM1_CCMR2 比较模式寄存器 2

偏移地址: 0x1C 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	Reserved		OC3CE	OC3M			OC3PE	OC3FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
15	OC4CE	通道 4 比较输出清零使能 (Output compare 4 clear enable) 参考 OC3CE 的描述
14: 12	OC4M	通道 4 比较输出模式 (Output compare 4 mode) 参考 OC3M 的描述
11	OC4PE	通道 4 比较输出预装载使能 (Output compare 4 preload enable) 参考 OC3PE 的描述
10	OC4FE	通道 4 比较输出快速使能 (Output compare 4 fast enable) 参考 OC3FE 的描述
9: 8	Reserved	保留, 必须保持复位值。
7	OC3CE	通道 3 比较输出清 0 使能 (Output compare 3 clear enable) 0: OC3REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入有效电平时, OC3REF 清零
6: 4	OC3M	通道 3 比较输出模式 (Output compare 3 mode) 该位定义了输出参考信号 OC3REF 的动作, 而 OC3REF 决定了 OC3、OC3N 的值。OC3REF 是高电平有效, 而 OC3、OC3N 的有效电平取决于 CC3P、CC3NP 位。 000: 冻结。TIM1_CCR3 与 TIM1_CNT 间的比较结果对 OC3REF 不起作用 001: 匹配时设置为高。当 TIM1_CNT 的值与 TIM1_CCR3 的值相同时, 强制 OC3REF 为高电平 010: 匹配时设置为低。当 TIM1_CNT 的值与 TIM1_CCR3 的值相同时, 强制 OC3REF 为低电平 011: 匹配时翻转。当 TIM1_CCR3=TIM1_CNT 时, 翻转 OC3REF 的电平 100: 强制为低。强制 OC3REF 为低电平 101: 强制为高。强制 OC3REF 为高电平 110: PWM 模式 1。在递增计数时, 当 TIM1_CNT<TIM1_CCR3 时强制 OC3REF 为高电平, 否则为低电平;在递减计数时, 当 TIM1_CNT > TIM1_CCR3 时强制 OC3REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIM1_CNT<TIM1_CCR3 时强制 OC3REF 为低电平, 否则为高电平;在递减计数时, 当 TIM1_CNT>TIM1_CCR3 时强制 OC3REF 为高电平, 否则为低电平。 注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 时, 该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能 (Output compare 3 preload enable) 0: 禁止 TIM1_CCR3 寄存器的预装载功能, 写入 TIM1_CCR3 寄存器的数值立即生效 1: 开启 TIM1_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR3 的预装载值在更新事件到来时生效 注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 时, 该位不能被修改。 注 2: 仅在单脉冲模式下 (TIM1_CR1 寄存器的 OPM= 1), 无需设定预装载寄存器, 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。
2	OC3FE	通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。 0: 禁止通道 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能
1: 0	Reserved	保留, 必须保持复位值。

11.5.9 TIM1_CCER 比较使能寄存器

偏移地址：0x20 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC 4P	CC 4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw															

Bit	Field	Description
15: 14	Reserved	保留，必须保持复位值。
13	CC4P	通道 4 输出极性 (Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	通道 4 输出使能 (Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	通道 3 互补输出极性 (Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	CC3NE	通道 3 互补输出使能 (Compare 3 complementary output enable) 参考 CC1NE 的描述。
9	CC3P	通道 3 输出极性 (Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	通道 3 输出使能 (Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	通道 2 互补输出极性 (Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	CC2NE	通道 2 互补输出使能 (Compare 2 complementary output enable) 参考 CC1NE 的描述。
5	CC2P	通道 2 输出极性 (Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	通道 2 输出使能 (Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	通道 1 互补输出极性 (Compare 1 complementary output polarity) 此位定义了输出信号极性： 0: OC1N 高电平有效 1: OC1N 低电平有效 注：当 LOCK 级别 (TIM1_BDTR 寄存器中的 LCCK 位) 设为 3 或 2 时，该位不能被修改。
2	CC1NE	通道 1 互补输出使能 (Compare 1 complementary output enable) 0: 关闭通道 1 互补输出。OC1N 禁止输出。 1: 开启通道 1 互补输出。 OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	通道 1 输出极性 (Compare 1 output polarity) 此位定义了输出信号极性： 0: OC1 高电平有效 1: OC1 低电平有效 注：当 LOCK 级别 (TIM1_BDTR 寄存器中的 LCCK 位) 设为 3 或 2 时，该位不能被修改。
0	CC1E	通道 1 输出使能 (Compare 1 output enable) 0: 关闭。OC1 禁止输出 1: 开启。OC1 信号输出到对应的输出引脚 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。

11.5.10 TIM1_CNT 计数器

偏移地址：0x24 复位值：

0x0000

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

Bit	Field	Description
15: 0	CNT	计数器的值 (Count value)

11.5.11 TIM1_PSC 预分频器

偏移地址：0x28 复位值：

0x0000

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

Bit	Field	Description
15: 0	PSC	预分频器的值 (Prescaler value) 计数器的时钟频率 (ck_cnt) = $f_{CK_PSC} / (PSC + 1)$ 当发生更新事件时, PSC 的值装入当前预分频寄存器。

11.5.12 TIM1_ARR 自动预装载寄存器

偏移地址：0x2C 复位值：

0x0000

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

Bit	Field	Description
15: 0	ARR	自动预装载值 (Auto-reload value) 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时, 计数器不工作。

11.5.13 TIM1_RCR 重复计数寄存器

偏移地址：0x30 复位值：

0x0000

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

Bit	Field	Description
15: 8	REP_CNT	重复计数器实时写入的值（Repetition counter value of real-time writing）在重复计数模式下，写入该位可以实时地将更新中断标志位（UIF）的检测点移位。 注：在更新事件后写入该位，在更新事件前写入 REP_CNT 将会被 REP 的值覆盖，使移位无效。
7: 0	REP	重复计数器的值（Repetition counter value） 重复计数器的值定义了更新事件的产生速率。重复计数器计数值递减为 0 时产生更新事件。如果允许产生更新中断，则会同时影响产生更新中断的速率。 对 REP 值的写入在下次更新事件发生时生效，所以在 PWM 模式中，（REP+1）对应着：在边沿对齐模式下，PWM 周期的数目 在中央对齐模式下，PWM 半周期的数目

11.5.14 TIM1_CCR1 比较寄存器 1

偏移地址：0x34 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15: 0	CCR1	通道 1 比较的值（compare 1 value） 如果在 TIM1_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会立即传输至对应的当前比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前比较影子寄存器中。当前比较影子寄存器参与同计数器 TIM1_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

11.5.15 TIM1_CCR2 比较寄存器 2

偏移地址：0x38 复位值：

0x0000

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

CCR2															
rw															

Bit	Field	Description
15: 0	CCR2	通道 2 比较的值（Compare 2 value） 参考 CCR1 的描述。

11.5.16 TIM1_CCR3 比较寄存器 3

偏移地址：0x3C 复位值：
0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
15: 0	CCR3	通道 3 比较的值（Compare 3 value） 参考 CCR1 的描述。

11.5.17 TIM1_CCR4 比较寄存器 4

偏移地址：0x40 复位值：
0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
15: 0	CCR4	通道 4 比较的值（Compare 4 value） 参考 CCR1 的描述。

11.5.18 TIM1_BDTR 刹车和死区寄存器

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
Reserved															DOE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK		DTG							
rw	rw	rw	rw	rw	rw	rw		rw							

注：根据锁定设置，DOE、AOE、BKP、BKE、OSSI、OSSR 和 DTG 位均可被写保护，有必要在第一次写入 TIM1_BDTR 寄存器时对它们进行配置，详见互补输出和死区插入章节。

Bit	Field	Description
31: 17	Reserved	保留，必须保持复位值。
16	DOE	直接输出（Direct output enable） 当刹车有效、MOE 置零后，有效。 0：刹车输入后，等待一个死区时间后输出空闲状态（输出使能信号关闭） 1：立即输出空闲状态（输出使能信号关闭） 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
15	MOE	主输出使能（Main output enable） 当通道 x 配置为输出时，根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。当刹车输入有效时，该位被硬件异步清 0。 0：禁止 OCx 和 OCxN 输出或强制为空闲状态（输出使能信号关闭） 1：如果设置了相应的使能位（TIM1_CCER 寄存器的 CCxE、CCxNE 位），则开启 OCx 和 OCxN 输出
14	AOE	自动输出使能（AutoMatic output enable） 0：MOE 不能被硬件置 1 1：MOE 能被软件置 1 或刹车无效时在下一个更新事件被硬件自动置 1 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
13	BKP	刹车输入极性（Break Polarity） 0：刹车输入低电平有效 1：刹车输入高电平有效 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
12	BKE	刹车功能使能（Break enable） 0：禁止刹车输入 1：开启刹车输入 注 1：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。 注 2：刹车输入包括引脚输入、比较器比较结果输入和 CSS 时钟失效三类事件，刹车使能前应先配置 TIM1_BKINF 寄存器中 BKIN_SEL 位，选择刹车源。
11	OSSR	运行模式下“关闭状态”选择（Off-state selection for Run mode） 该位仅适用于当 MOE = 1 且通道为互补输出。 0：当定时器不工作时，禁止 OC/OCN 输出 1：当定时器不工作时，如果 CCxE = 1 或 CCxNE = 1，首先开启 OC/OCN 并输出无效电平，然后置位 OC/OCN 输出使能信号。 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。
10	OSSI	空闲模式下“关闭状态”选择（Off-state selection for Idle mode） 该位仅适用于当 MOE = 0 且通道设为输出时。 0：当定时器不工作时，禁止 OC/OCN 输出。 1：当定时器不工作时，如果 CCxE = 1 或 CCxNE = 1，首先 OC/OCN 输出无效电平，然后置位 OC/OCN 输出使能信号。 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。

Bit	Field	Description
9: 8	LOCK	锁定设置 (Lock configuration) 该位定义了寄存器的写保护功能。 00: 写保护功能关闭, 寄存器无写保护 01: 锁定级别 1, 不能写入 TIM1_BDTR 寄存器的 DOE、DTG、BKE、BKP、AOE 位和 TIM1_CR2 寄存器的 OISx/OISxN 位 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位以及 OSSR/OSSI 位 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位 注: 在系统复位后, LOCK 位只能写一次, 当写入 TIM1_BDTR 寄存器后, LOCK 被写保护。
7: 0	DTG	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。 注: 当 LOCK 级别 (TIM1_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3 时, 不能修改这些位。

11.5.19 TIM1_DCR DMA 控制寄存器

偏移地址: 0x48 复位值:

0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 11.4.8 DMA 章节。

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

Bit	Field	Description
15: 13	Reserved	保留, 必须保持复位值。
12: 8	DBL	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的访问寄存器的数量 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输
7: 5	Reserved	保留, 必须保持复位值。
4: 0	DBA	DMA 基地址 (DMA base address) 这些位定义了 DMA 在连续模式下访问 TIM1_DMAR 寄存器的第一个地址。DBA 定义为从 TIM1_CR1 寄存器所在地址开始的偏移值: 00000: TIM1_CR1 00001: TIM1_CR2 00010: TIM1_SMCR

11.5.20 TIM1_DMAR 连续模式 DMA 地址寄存器

偏移地址: 0x4C

复位值：0x0000

注：本寄存器仅适用于有内置 DMA 的产品，详见 11.4.8 DMA 章节。

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

Bit	Field	Description
15: 0	DMAB	DMA 连续传送地址寄存器（DMA address register for burst accesses） 对 TIM1_DMAR 寄存器的读写操作会导致对以下地址所在寄存器的存取操作： TIM1_CR1 地址 + DBA + DMA 索引，其中 TIM1_CR1 地址是 TIM1_CR1 寄存器所在的地址，DBA 是 TIM1_DCR 寄存器中定义的基地址，DMA 索引是 DMA 自动控制的偏移量，它取决于 TIM1_DCR 寄存器中定义的 DBL 值。

11.5.21 TIM1_CCMR3 比较模式寄存器 3

偏移地址：0x54 复位值：

0x0000 通道仅适用于比较输出

模式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OC5PE	Reserved		
												rw			

Bit	Field	Description
15: 4	Reserved	保留，必须保持复位值。
3	OC5PE	比较输出 5 预装载使能（Output compare 5 preload enable） 0：禁止 TIM1_CCR5 寄存器的预装载功能，写入 TIM1_CCR5 寄存器的数值立即生效。 1：开启 TIM1_CCR5 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CCR5 的预装载值在更新事件到来时生效。 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）时，该位不能被修改。
2: 0	Reserved	保留，必须保持复位值。

11.5.22 TIM1_CCR5 比较寄存器 5

偏移地址：0x58 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5															
rw															

Bit	Field	Description
15: 0	CCR5	比较 5 的值 (Compare 5 value) CC5 通道只能配置为输出： 如果在 TIM1_CCMR3 寄存器 (OC5PE 位) 中未选择预装载功能，写入的数值会立即传输至对应的当前比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前比较影子寄存器中。当前比较影子寄存器参与同计数器 TIM1_CNT 的比较，由于 CC5 通道为内部通道，无法输出至引脚，比较结果用于内部触发事件。

11.5.23 TIM1_PDER PWM 移相/DMA repeat 更新请求使能寄存器

偏移地址：0x5C 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CCR3_SHIFT_EN	CCR2_SHIFT_EN	CCR1_SHIFT_EN	CCDREPE
												rw	rw	rw	rw

Bit	Field	Description
15: 4	Reserved	保留，必须保持复位值。
3	CCR3_SHIFT_EN	允许通道 3 输出 PWM 移相使能位 0: 禁止通道 3 输出 PWM 移相 1: 允许通道 3 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
2	CCR2_SHIFT_EN	允许通道 2 输出 PWM 移相使能位 0: 禁止通道 2 输出 PWM 移相 1: 允许通道 2 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
1	CCR1_SHIFT_EN	允许通道 1 输出 PWM 移相使能位 0: 禁止通道 1 输出 PWM 移相 1: 允许通道 1 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
0	CCDREPE	使能 DMA 在每次 underflow 或是 overflow 时都发出更新请求 0: DMA 发生更新请求需要根据重复计数寄存器的值来产生。 1: 使能 DMA 在每次 underflow 或 overflow 都发出更新请求。

11.5.24 TIM1_CCRxFALL PWM 移相递减计数比较寄存器

偏移地址：0x60 ~ 0x68

复位值：0x0000

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

Bit	Field	Description
15: 0	CCRxFALL	通道 x 在 PWM 中央对齐模式递减计数时的比较值 PWM 移相功能：开启 PDER 寄存器的 PWM 移相使能，根据需要移动相位，配置 CCRxFALL 以及 CCRx，即可实现 PWM 输出可编程的移相波形，可左移或是右移。

11.5.25 TIM1_BKINF 刹车输入滤波寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	COMPBKIN_SEL	IOBKIN_SEL								CSSBKIN_SEL	BKINF				BKINF
	rw	rw								rw	rw				rw

Bit	Field	Description
31: 15	Reserved	保留，必须保持复位值。
14	COMPBKIN_SEL	COMP 刹车输入选择(COMP break input sel) COMPBKIN_SEL[0]: 0: COMP1_OUT 刹车输入无效 1: COMP1_OUT 刹车输入有效
13: 6	IOBKIN_SEL	IO 刹车输入选择(IO break input sel) IOBKIN_SEL[7]: 0: TIM1_BKIN8 刹车输入无效 1: TIM1_BKIN8 刹车输入有效 IOBKIN_SEL[6]: 0: TIM1_BKIN7 刹车输入无效 1: TIM1_BKIN7 刹车输入有效 IOBKIN_SEL[5]: 0: TIM1_BKIN6 刹车输入无效 1: TIM1_BKIN6 刹车输入有效 IOBKIN_SEL[4]: 0: TIM1_BKIN5 刹车输入无效 1: TIM1_BKIN5 刹车输入有效 IOBKIN_SEL[3]: 0: TIM1_BKIN4 刹车输入无效 1: TIM1_BKIN4 刹车输入有效 IOBKIN_SEL[2]: 0: TIM1_BKIN3 刹车输入无效 1: TIM1_BKIN3 刹车输入有效 IOBKIN_SEL[1]: 0: TIM1_BKIN2 刹车输入无效 1: TIM1_BKIN2 刹车输入有效 IOBKIN_SEL[0]: 0: TIM1_BKIN1 刹车输入无效 1: TIM1_BKIN1 刹车输入有效

Bit	Field	Description
5	CSSBKIN_SEL	CSS 刹车输入选择(CSS break input sel) CSSBKIN_SEL: 0: CSS 刹车输入无效 1: CSS 刹车输入有效
4: 1	BKINF	BKIN 数字滤波采样频率(break input filter) 0000: 2 周期 0001: 4 周期 0010: 8 周期 0011: 16 周期 0100: 32 周期 0101: 64 周期 0110: 128 周期 0111: 256 周期 1000: 384 周期 1001: 512 周期 1010: 640 周期 1011: 768 周期 1100: 896 周期 1101: 1024 周期 1110: 1152 周期 1111: 1280 周期 注: 更改档位时应先关闭刹车滤波使能位。
0	BKINFE	BKIN 数字滤波使能(break input filter enable) 1: 使能 BKIN 管脚数字滤波 0: 禁止 BKIN 管脚数字滤波 注: 滤波使能信号在刹车条件配置之后打开。 滤波功能用于过滤电平。

12 TIM3 通用定时器

16 位通用定时器包含 TIM3、TIM4，具有相同的功能，本产品仅搭载 TIM3。本章节统一表述为 TIMx，部分示意图以 TIM3 为例展示。

12.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 16 位计数方向可调的自动重装载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。通用定时器具有多种用途，如输入功能（测量输入信号的脉冲宽度、频率，PWM 输入等），输出功能（PWM 输出、单脉冲模式输出等）。

12.2 功能框图

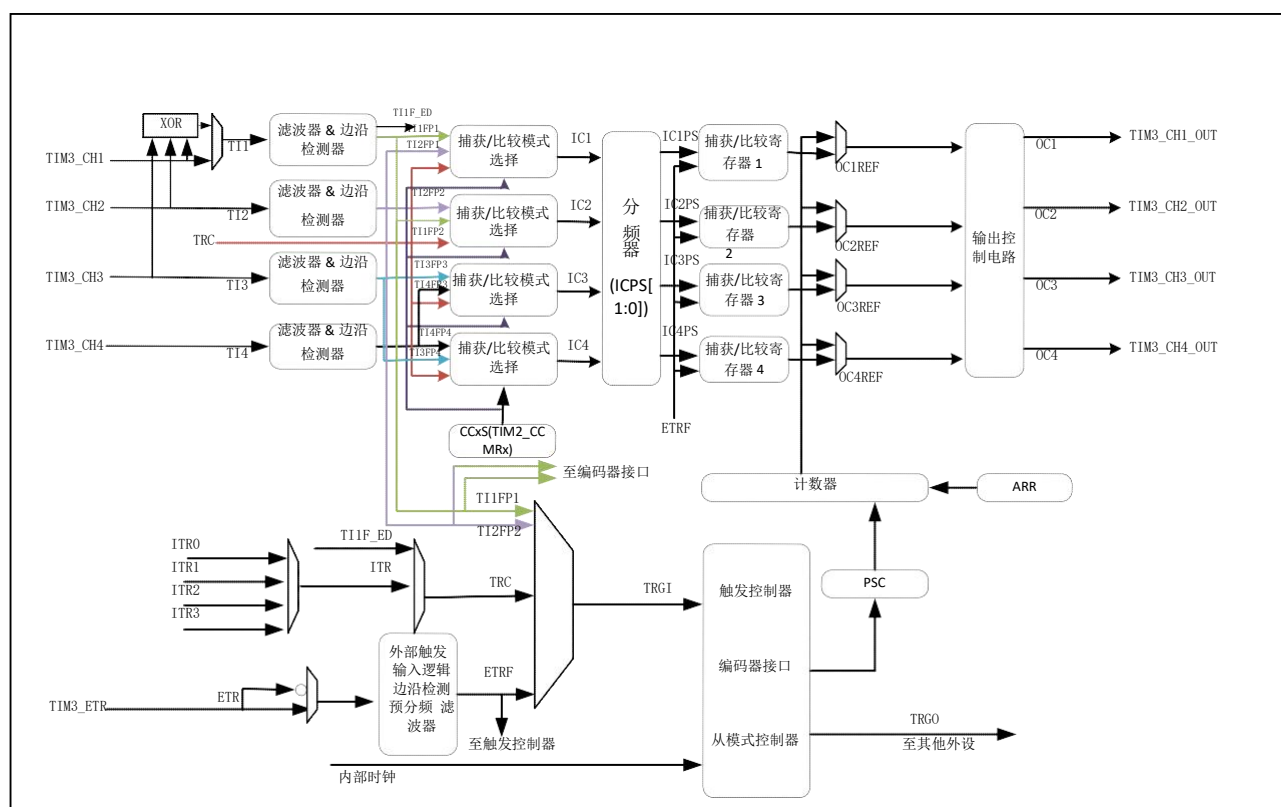


图 12-1 TIMx 结构图

上图为 TIMx 的结构框图，主要由输入单元、输出单元、时基单元、捕获/比较模块等结构组成。

12.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 时钟源可选：内部时钟源，外部时钟输入（Tl_x、ETR_x），内部触发输入(ITR_x)
- 16 位自动重装载计数器（计数方向：递增、递减、递增/递减）
- 输入捕获：输入信号的脉冲宽度、周期的测量
- 触发输入可以作为外部时钟或者逐周期管理
- 支持编码器、霍尔传感器等接口
- 4 个输出通道
- 比较输出（控制输出波形或指示定时器已经计时结束）
- PWM 输出（边沿对齐或中央对齐模式）
- 单脉冲输出
- 产生中断/DMA 请求的事件：更新事件、触发事件、输入捕获、比较输出

12.4 功能描述

12.4.1 时钟

12.4.1.1 时钟选择 计数器的时钟

源有以下几种：

- 内部时钟（INT_CK）
- 外部时钟模式 1：外部触发输入 TRGI（包含 Tl_x、ITR_x、ETR_x）
- 外部时钟模式 2：外部触发输入 ETR（包含 ETR_x）
- 编码器模式 上述几种时钟

选择示意图如下：

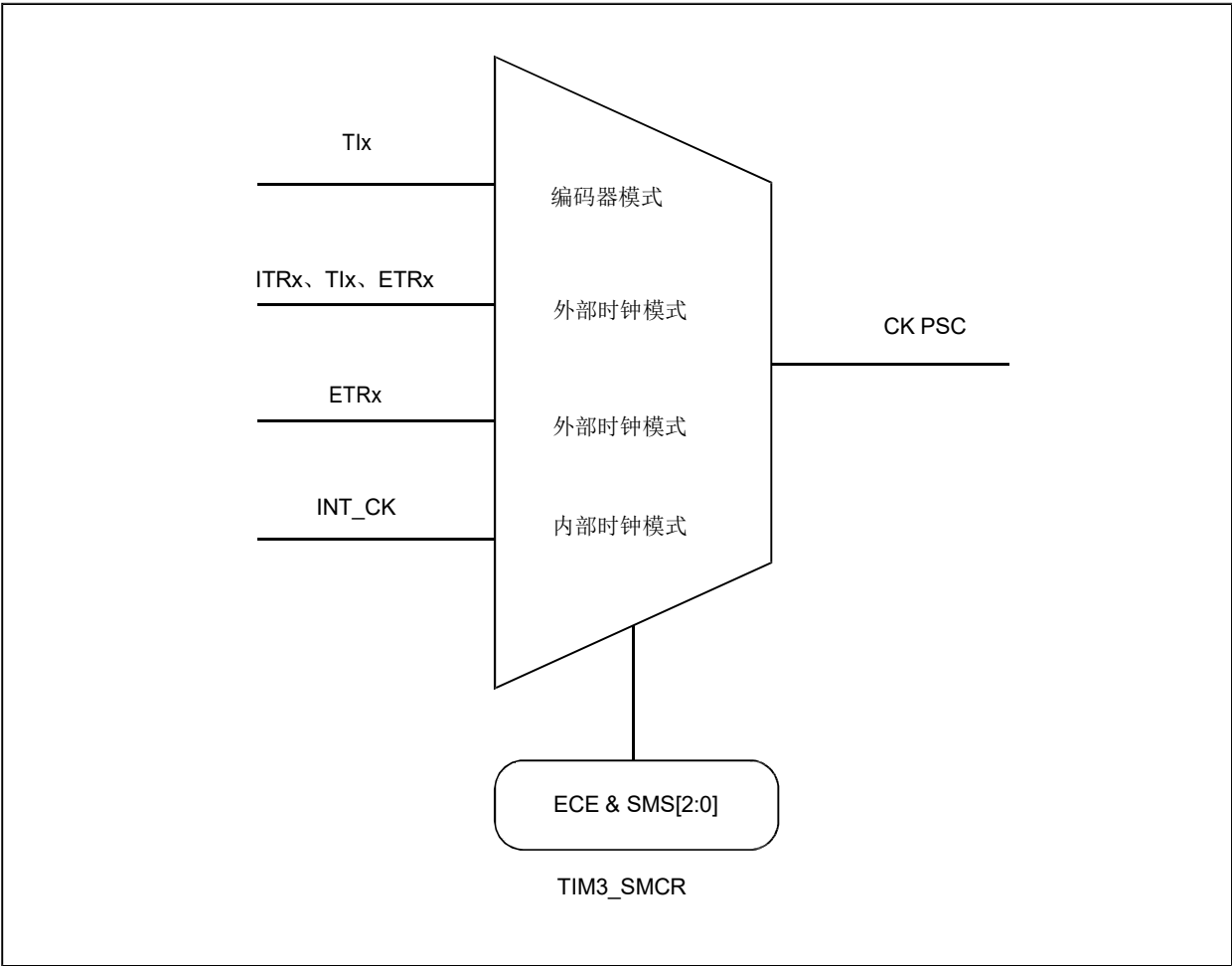


图 12-2 时钟选择

12.4.1.1.1 内部时钟源 (INT_CK)

当配置 TIMx_SMCR 寄存器的 SMS=000、关闭从模式时，计数器使能打开，预分频器的时钟直接由内部时钟驱动。此时计数器时钟为内部时钟分频后的时钟。

12.4.1.1.2 外部时钟模式 1（外部触发输入 TRGI，包含 TIx、ITRx、ETRx）

当配置 TIMx_SMCR 寄存器的 SMS = 111 时，选择外部时钟模式 1（TRGI）。计数器由选定的输入信号的每个上升沿或下降沿驱动。

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上；配置 TIMx_CCMR1 寄存器的 IC1F[3: 0]，设置输入滤波器带宽；配置 TIMx_CCER 寄存器 CC1P=0，选择上升沿为有效沿。
2. 配置 TIMx_SMCR 寄存器的 TS=101，选择 TI1 的作为触发输入源；配置 TIMx_SMCR 寄存器的 SMS=111，选择外部时钟模式 1。

3. 配置 TIMx_CR1 寄存器的 DIR=0，选择递增计数模式，配置 TIMx_CR1 寄存器的 CEN=1，启动计数器。

当 TI1 出现有效边沿时，计数器递增计数一次且 TIF 标志位由硬件置 1。TI1 的有效边沿和计数器的实际时钟之间的延时取决于 TI1 输入端同步电路设计。

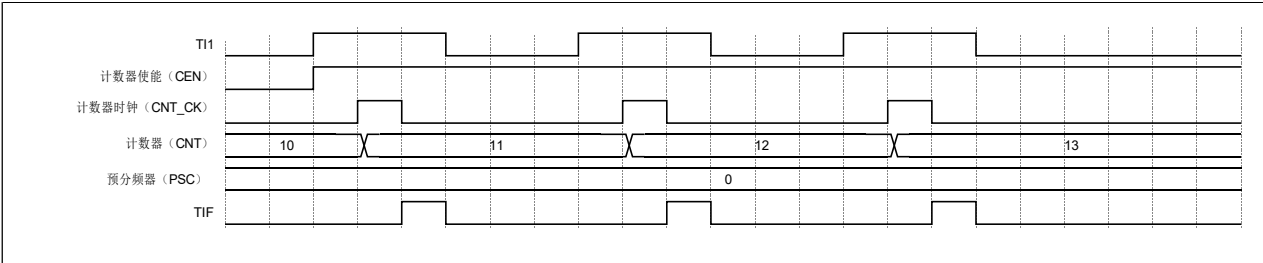


图 12-3 外部时钟模式 1 下的控制电路

12.4.1.1.3 外部时钟模式 2（外部触发输入 ETR，包含 ETRx）

当配置 TIMx_SMCR 寄存器的 ECE=1 时，使能外部时钟模式 2，计数器由 ETR 信号上的有效边沿驱动。

例：ETR 的每 4 个下降沿计数一次，递增计数，具体配置如下：

1. 配置 TIMx_SMCR 寄存器的 ETF[3: 0] = 0010，每 4 个 ETR 信号的有效边沿驱动计数器计数一次；配置 TIMx_SMCR 寄存器的 ETP=1，选择下降沿有效；配置 TIMx_SMCR 寄存器的 ECE=1，选择外部时钟模式 2。

2. 配置 TIMx_CR1 寄存器的 DIR=0，选择递增计数模式；配置 TIMx_CR1 寄存器的 CEN=1，启动计数器。

在 ETR 的下降沿和计数器实际时钟之间的延时取决于在 ETR 信号端的同步电路设计。

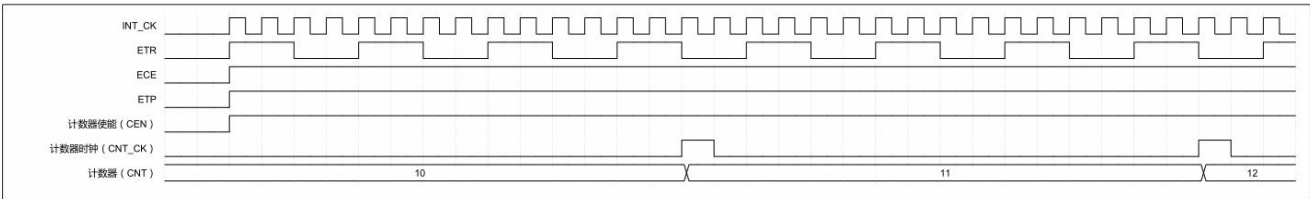


图 12-4 外部时钟模式 2 下的控制电路

12.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

12.4.1.2 时基单元

TIMx 的时基单元主要包括：计数器寄存器（TIMx_CNT）、预分频器寄存器（TIMx_PSC）和自动

预装载寄存器（TIMx_ARR）。

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

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

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

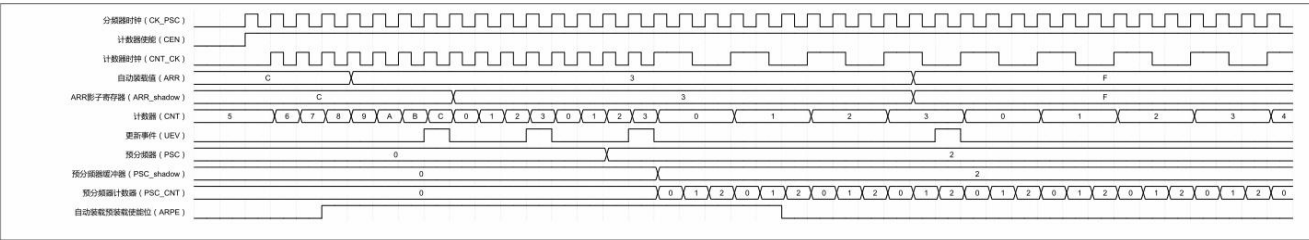


图 12-5 自动预装载

12.4.1.3 计数模式

通过配置 TIMx_CR1 寄存器的 DIR 位和 CMS 位可以选择计数器的计数模式，可以分为三种计数模式，递增计数模式、递减计数模式和中央对齐计数模式（递增/递减计数模式），下面对每种计数模式做详细介绍。

12.4.1.3.1 递增计数模式

配置 TIMx_CR1 寄存器 CMS=0，DIR=0，选择递增计数模式。

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

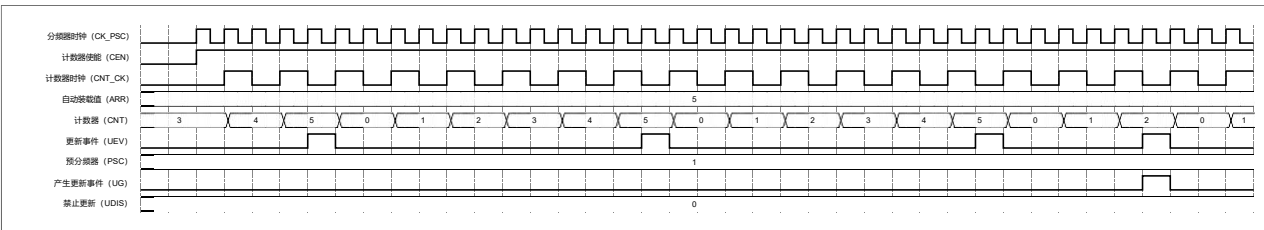


图 12-6 递增计数模式（UDIS=0）

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

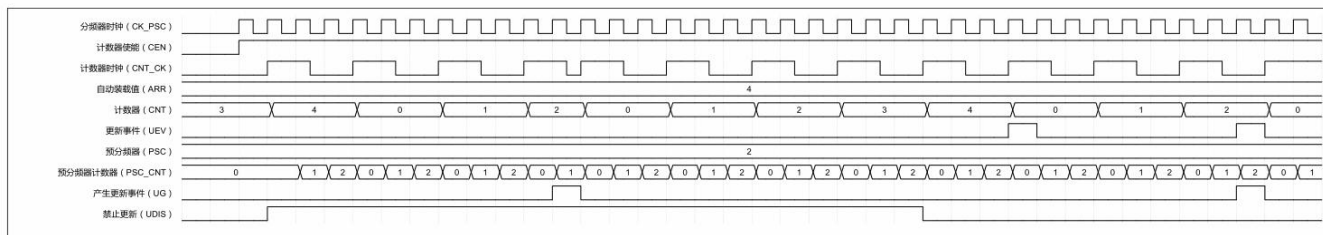


图 12-7 递增计数模式 (UDIS=1 禁止产生更新事件)

注：发生更新事件时

- ARR 寄存器中的值被载入 ARR 影子寄存器中。
- 预分频器的预装载值生效。

12.4.1.3.2 递减计数模式

配置 TIMx_CR1 寄存器的 CMS=0, DIR=1, 选择递减计数模式。递减计数模式下, 计数器从自动预装载值 TIMx_ARR 开始递减计数, 计数到 0 时, 产生一个下溢

事件 (更新事件)。设置 TIMx_EGR 寄存器的 UG=1, 同样可以产生一个更新事件, 更新事件后计数器从自动预装载值 TIMx_ARR 开始重新递减计数 (TIMx_CR1 寄存器 UDIS=0)。

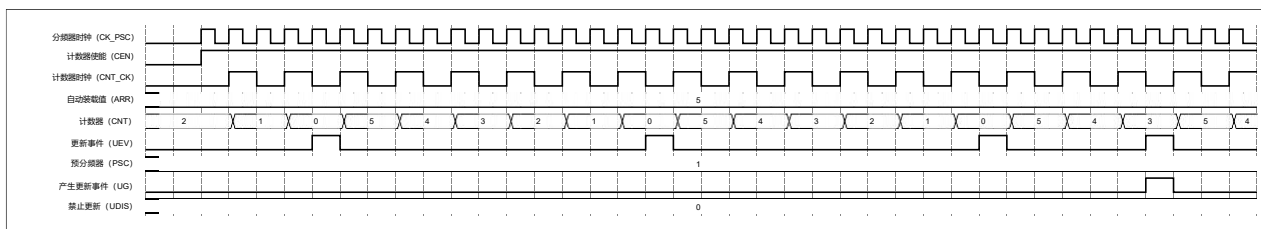


图 12-8 递减计数模式 (UDIS=0)

通过配置 TIMx_CR1 寄存器的 UDIS=1, 可禁止产生更新事件, 当计数器发生下溢事件时, 不产生更新事件。此时若配置 UG=1, 同样不产生更新事件, 但是计数器和预分频器计数器会被初始化, 从 TIMx_ARR 开始计数。

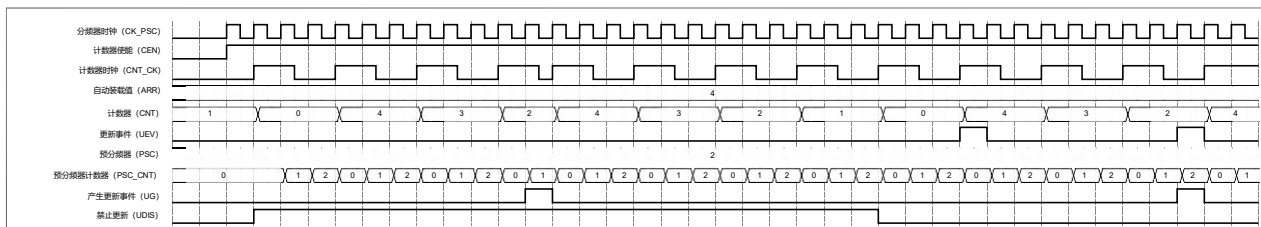


图 12-9 递减计数模式 (UDIS=1 禁止产生更新事件)

12.4.1.3.3 中央计数模式 (递增/递减计数模式)

配置 TIMx_CR1 寄存器的 CMS ≠ 0 (此时写入 DIR 无效), 选择中央对齐计数模式。

中央对齐计数模式，递增计数和递减计数交替进行。递增计数到 **ARR-1** 时，产生一个上溢事件，然后从 **ARR** 开始递减计数到 **1**，产生一个下溢事件，再从 **0** 开始递增计数。

设置 **TIMx_EGR** 寄存器的 **UG=1**，同样可以产生一个更新事件，更新事件后计数器从 **0** 开始重新递增计数（**TIMx_CR1** 寄存器 **UDIS=0**）。

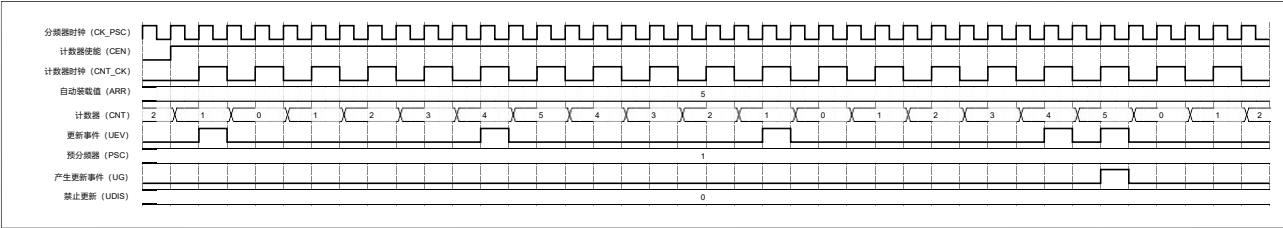


图 12-10 中央计数模式（UDIS=0）

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

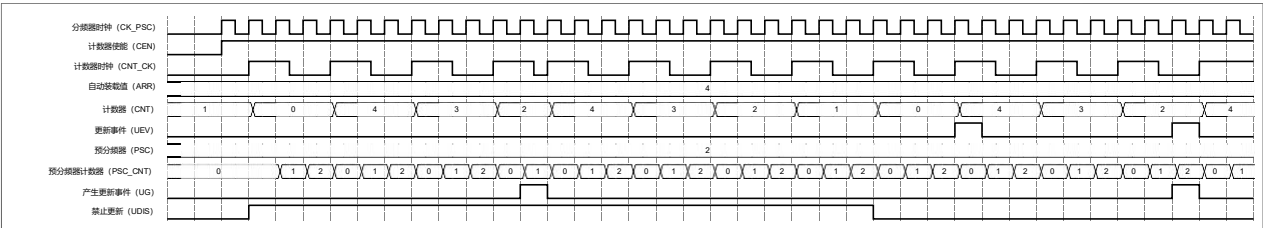


图 12-11 中央计数模式（UDIS=1 禁止产生更新事件）

12.4.2 输入捕获

12.4.2.1 输入捕获 输入捕获部分包括数字滤波器、多路复用、预分频器等，其结构

如下图所示：

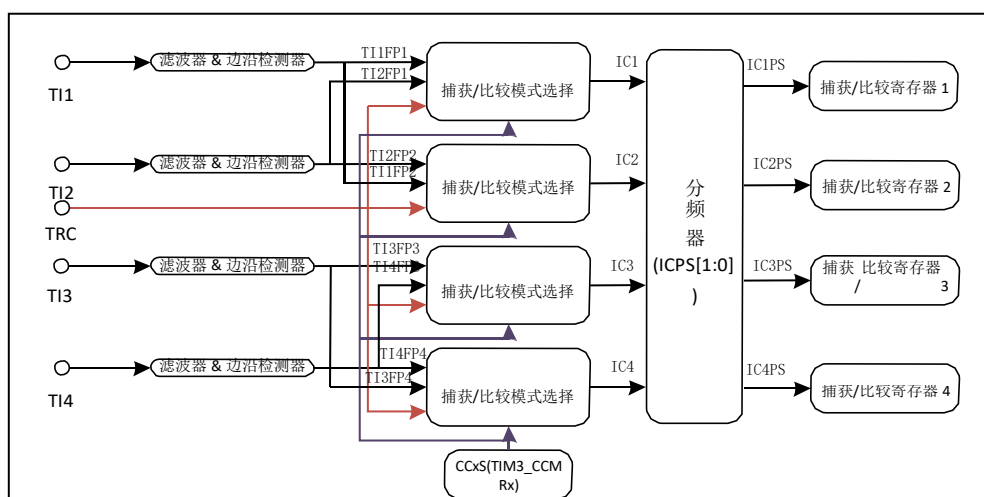


图 12-12 TIMx 输入捕获结构图

通过配置 TIMx_CCMRx 寄存器的 ICxF，可以设置数字滤波器的滤波宽度（滤波器的采样频率及数字滤波宽度如下表所示），当数字滤波器的输入信号宽度大于滤波宽度时，输入信号有效；数字滤波器对输入引脚 TIx 的输入信号采样后，产生一个滤波后的信号 TIxF，然后通过极性可选的边沿检测器，产生一个有效信号 TIxFPx，这个信号可以作为从模式控制器的触发输入信号，同时该信号经过预分频器产生一个信号 ICxPS，用于触发输入捕获事件。

表 12-1 数字滤波器宽度与 ICxF 的对应关系表

IC1F[3: 0]	采样频率和滤波宽度	IC1F[3: 0]	采样频率和滤波宽度
0000	无滤波器，以 f _{DTS} 采样	1000	采样频率 f _{sampling} =f _{DTS} /8, N=6
0001	采样频率 f _{sampling} =f _{INT CK} , N=2	1001	采样频率 f _{sampling} =f _{DTS} /8, N=8
0010	采样频率 f _{sampling} =f _{INT CK} , N=4	1010	采样频率 f _{sampling} =f _{DTS} /16, N=5
0011	采样频率 f _{sampling} =f _{INT CK} , N=8	1011	采样频率 f _{sampling} =f _{DTS} /16, N=6
0100	采样频率 f _{sampling} =f _{DTS} /2, N=6	1100	采样频率 f _{sampling} =f _{DTS} /16, N=8
0101	采样频率 f _{sampling} =f _{DTS} /2, N=8	1101	采样频率 f _{sampling} =f _{DTS} /32, N=5
0110	采样频率 f _{sampling} =f _{DTS} /4, N=6	1110	采样频率 f _{sampling} =f _{DTS} /32, N=6
0111	采样频率 f _{sampling} =f _{DTS} /4, N=8	1111	采样频率 f _{sampling} =f _{DTS} /32, N=8

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

例如，通过采样 TI1 输入信号的有效沿，在 TI1 的上升沿来到时捕获当前计数器的值，锁存到 TIMx_CCR1 寄存器中，步骤如下：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 IC1F[3: 0]，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 TIMx_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 IC1PSC[1:0]，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。

6. 配置 TIMx_DIER 寄存器的 CC1IE=1，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 DMA，配置 TIMx_DIER 寄存器的 CC1DE=1，允许捕获/比较通道 1 的 DMA 请求。

注：

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

12.4.2.2 PWM 捕获

PWM 输入模式的操作配置与一般输入捕获有以下不同点：

- 两个边沿有效且极性相反的 ICx 信号被映射至同一个 Tlx 输入。
- 配置从模式为复位模式，将其中一路 TlxFP 作为触发输入信号。

例：测量 TI1 的 PWM 信号的宽度（TIMx_CCR1 寄存器）和占空比（TIMx_CCR2 寄存器），测量值取决于内部时钟 INT_CK 的频率和预分频器的值。具体步骤如下：

1. 配置 TIMx_CR1 寄存器 DIR=0，选择计数器计数模式为递增计数模式。
2. 配置 TIMx_CCMR1 寄存器的 CC1S = 01，将 IC1 映射在 TI1 上，选择 TIMx_CCR1 的有效输入。
3. 配置 TIMx_CCER 寄存器的 CC1P =0，选择 TI1FP1 的有效极性（上升沿有效）（将计数器的值捕获到 TIMx_CCR1 中并清除计数器）。
4. 配置 TIMx_CCMR1 寄存器的 CC2S =10，将 IC2 映射在 TI1 上，选择 TIMx_CCR2 的有效输入。
5. 配置 TIMx_CCER 寄存器的 CC2P =1，选择 TI2FP2 的有效极性（下降沿有效）（将计数器的值捕获到 TIMx_CCR2 中）。
6. 配置 TIMx_SMCR 寄存器中的 TS = 101，选择 TI1FP1 为有效的触发输入信号。
7. 配置 TIMx_SMCR 中的 SMS = 100，从模式控制器设置为复位模式。
8. 配置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E = 1。开启 CC1 通道和 CC2 通道的捕获使能。

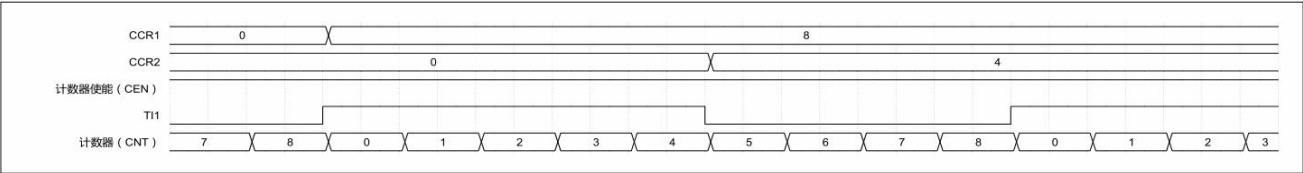


图 12-13 PWM 输入模式时序

注：由于从模式控制器只连接了 TI1FP1 和 TI2FP2，所以 PWM 输入模式只适用于 TIMx_CH1/TIMx_CH2 端口输入信号。

12.4.3 比较输出

捕获比较通道的比较输出部分由比较器、输出控制电路和捕获/比较寄存器组成，其结构图如下图所示：

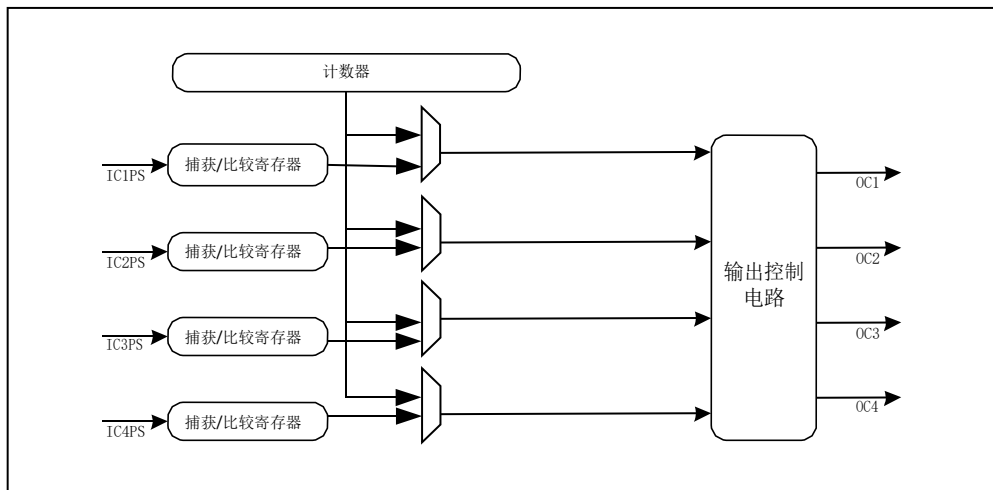


图 12-14 比较输出部分结构图

在比较输出模式下，捕获比较寄存器的内容被载入到影子寄存器中，然后影子寄存器的内容和计数器当前值进行比较。捕获/比较模块包括一个捕获/比较寄存器（预装载寄存器）和一个影子寄存器，读写过程仅操作捕获/比较寄存器。

12.4.3.1 强制输出

配置 `TIMx_CCMRx` 寄存器的 `CCxS = 00`，将通道 `CCx` 设置为输出模式，通过配置 `TIMx_CCMRx` 寄存器 `OCxM` 位，可以直接将比较输出信号直接强制为有效或无效状态，不依赖于比较结果。配置 `TIMx_CCMRx` 寄存器 `OCxM = 100`，强置比较输出信号为无效状态。此时 `OCxREF` 被强置为低电平。配置 `TIMx_CCMRx` 寄存器 `OCxM = 101`，强置比较输出信号为有效状态。此时 `OCxREF` 被强置为高电平（`OCxREF` 始终为高电平有效）。

注：强制输出模式下，在 `TIMx_CCRx` 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断或 DMA 请求，仍会产生对应的中断或 DMA 请求。

12.4.3.2 比较输出

比较输出模式下，当计数器与捕获比较寄存器值相同时，可以根据 `TIMx_CCMRx` 寄存器的 `OCxM` 位的配置用来输出不同的波形。例如，当计数器与捕获/比较寄存器的内容匹配时，

比较输出模式下的操作如下： 1. 在比较匹配时，`OCxM` 的值不同，输出通道 `x` 信号 `OCx` 的操作不同：

- ◆ `OCxM = 000`：`OCx` 信号保持它的电平
- ◆ `OCxM = 001`：`OCx` 信号被设置成有效电平

- ◆ OCxM = 010: OCx 信号被设置成无效电平
 - ◆ OCxM = 011: OCx 信号进行翻转
2. 匹配时状态寄存器中的标志位置 1 (TIMx_SR 寄存器中的 CCxIF 位)。
 3. 当配置了 TIMx_DIER 寄存器中的 CCxIE =1, 匹配时则产生一个中断。
 4. 当配置了 TIMx_DIER 寄存器中的 CCxDE =1, 匹配时则产生一个 DMA 请求 (仅适用于有内置 DMA 的产品)。

比较输出模式也可以用来输出一个单脉冲 (单脉冲输出模式)。 例如, 通道 1 的比较输出模式的配置步骤如下:

1. 配置计数器的时钟 (选择时钟源, 配置预分频系数)。
2. 配置 TIMx_ARR 和 TIMx_CCR1 寄存器。
3. 配置 TIMx_DIER 寄存器的 CC1IE =1, 使能捕获/比较 1 中断。
4. 配置输出模式:
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1M = 011, OC1 比较匹配时翻转。
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1PE = 0 , 禁止 TIMx_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1P = 1, OC1 低电平有效。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1E = 1, 开启输出/比较 1 输出使能, OC1 信号输出到对应的输出引脚。
5. 配置 TIMx_CR1 寄存器的 CEN =1, 启动计数器。

当配置 TIMx_CCMRx 寄存器中 OCxPE=0, 禁止 TIMx_CCRx 寄存器的预装载功能时, 可以随时写入 TIMx_CCRx 寄存器, 并且写入的值立即生效。当配置 TIMx_CCMRx 寄存器中 OCxPE=1, 启用 TIMx_CCRx 寄存器的预装载功能时, 读写仅对预装载寄存器进行操作, TIMx_CCRx 预装载寄存器的值在下次更新事件到来时生效。下图给出了一个例子。

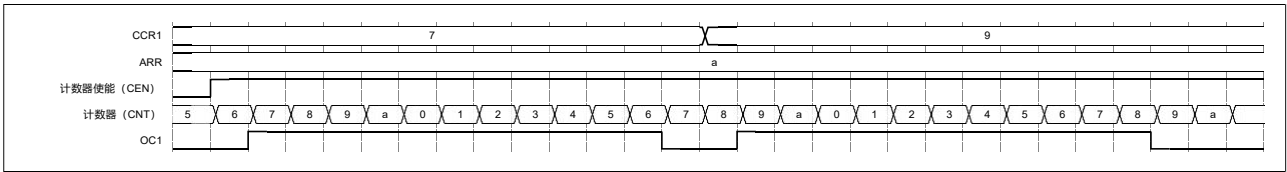


图 12-15 比较输出模式, OC1 信号在匹配时翻转

注: 比较输出模式下, 更新事件不会对输出结果产生影响。强制输出模式下, 在 TIMx_CCRx 影子寄存器和计数器之间的比较输出仍在进行, 比较结果的相应标志位也会被修改, 如果开启了对应的中断或 DMA 请求, 仍会产生对应的中断或 DMA 请求。

12.4.3.3 PWM 输出

在 PWM 模式下, 根据 TIMx_ARR 寄存器和 TIMx_CCRx 寄存器的值, 产生一个频率、占空比可

控的 PWM 波形。

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

配置 $TIMx_CCER$ 寄存器的 $CCxP$ 位选择 OCx 的有效极性。配置 $TIMx_CCER$ 寄存器的 $CCxE$ 位控制 OCx 的输出使能。配置 $TIMx_CR1$ 寄存器的 CMS 位，可以选择产生边沿对齐或中央对齐的 PWM 信号。

- $CMS=00$ ，边沿对齐模式，再进一步配置 DIR ，选择递增或递减计数模式。
- $CMS=01$ ，中央对齐模式 1。
- $CMS=10$ ，中央对齐模式 2。
- $CMS=11$ ，中央对齐模式 3。

12.4.3.3.1 PWM 边沿对齐模式——递增计数模式

在递增计数模式配置的基础上，配置 $TIMx_CCMRx$ 寄存器的 $CCxS=00$ ，选择输出模式， $OCxM=110$ ，选择 PWM 模式 1，当 $TIMx_CNT < TIMx_CCRx$ 时通道 x ($OCxREF$) 为有效电平，否则为无效电平。如果 $TIMx_CCRx$ 中的比较值大于自动重装载值 ($TIMx_ARR$)，则 $OCxREF$ 保持为有效电平。如果比较值为 0，则 $OCxREF$ 保持为无效电平。下图为 $CCR1=1$ ， $CCR2=4$ ， $CCR3=7$ ， $CCR4=b$ ， $ARR=a$ 时边沿对齐递增计数时 PWM 模式 1 的波形实例。

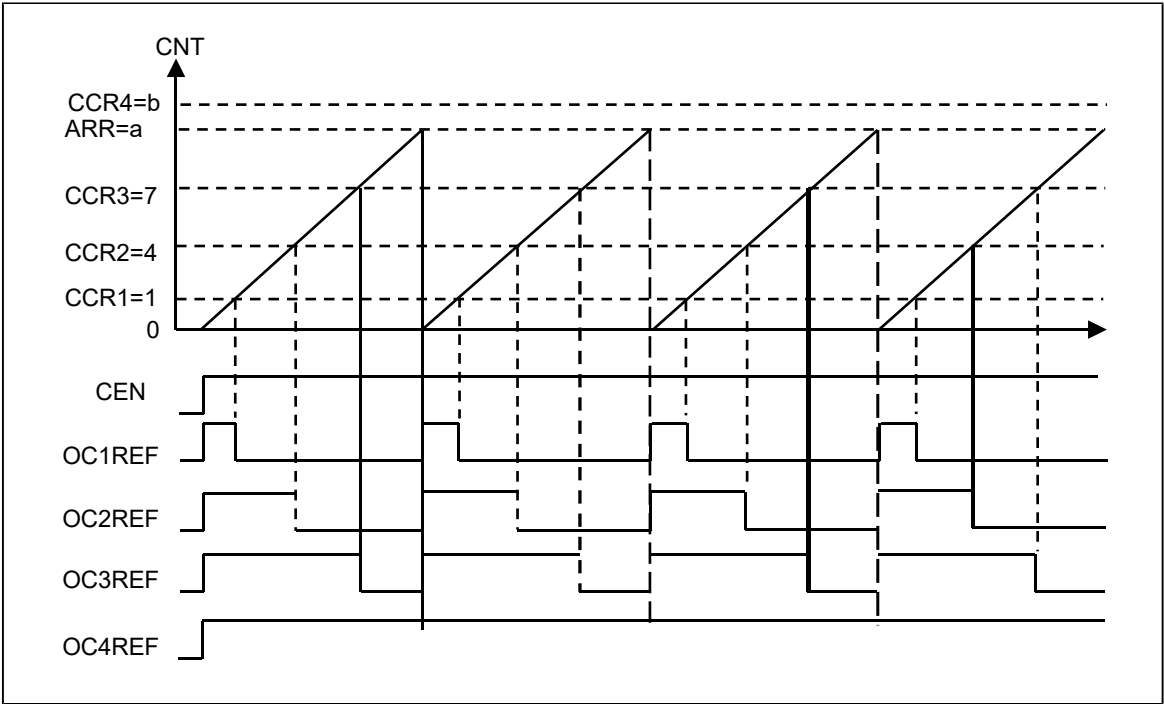


图 12-16 边沿对齐递增计数时 PWM 模式 1 的波形

12.4.3.3.2 PWM 边沿对齐模式——递减计数模式

在递减计数模式配置的基础上，配置 TIMx_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，选择 PWM 模式 1，当 TIMx_CNT > TIMx_CCRx 时通道 x (OCxREF) 为无效电平，否则有效电平。下图为 CCR1=4，CCR2=6，CCR3=9，CCR4=b，ARR=a 时边沿对齐递减计数时 PWM 模式 1 的波形实例。

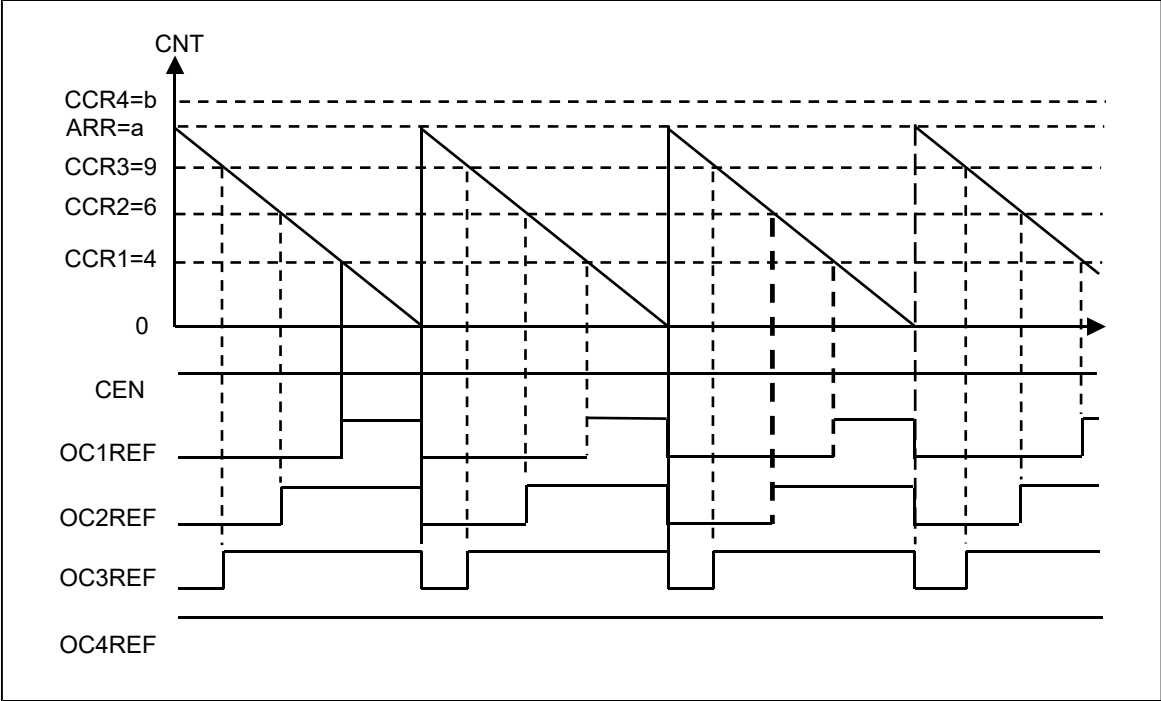


图 12-17 边沿对齐递减计数时 PWM 模式 1 的波形

12.4.3.3.3 PWM 中央对齐模式

首先配置 TIMx 计数器为中央对齐计数模式，配置 TIMx_CCMRx 寄存器的 CCxS=00，选择输出模式，根据配置不同的 CMS，比较输出中断标志位在计数器递减计数时被设置（CMS=01）、在计数器递增计数时被设置（CMS=10）、或在计数器递增或递减计数时被设置（CMS=11）。下图为 CCR1=4，CCR2=6，CCR3=9，CCR4=b，ARR=a 时中央对齐 PWM 模式 1 的波形实例。

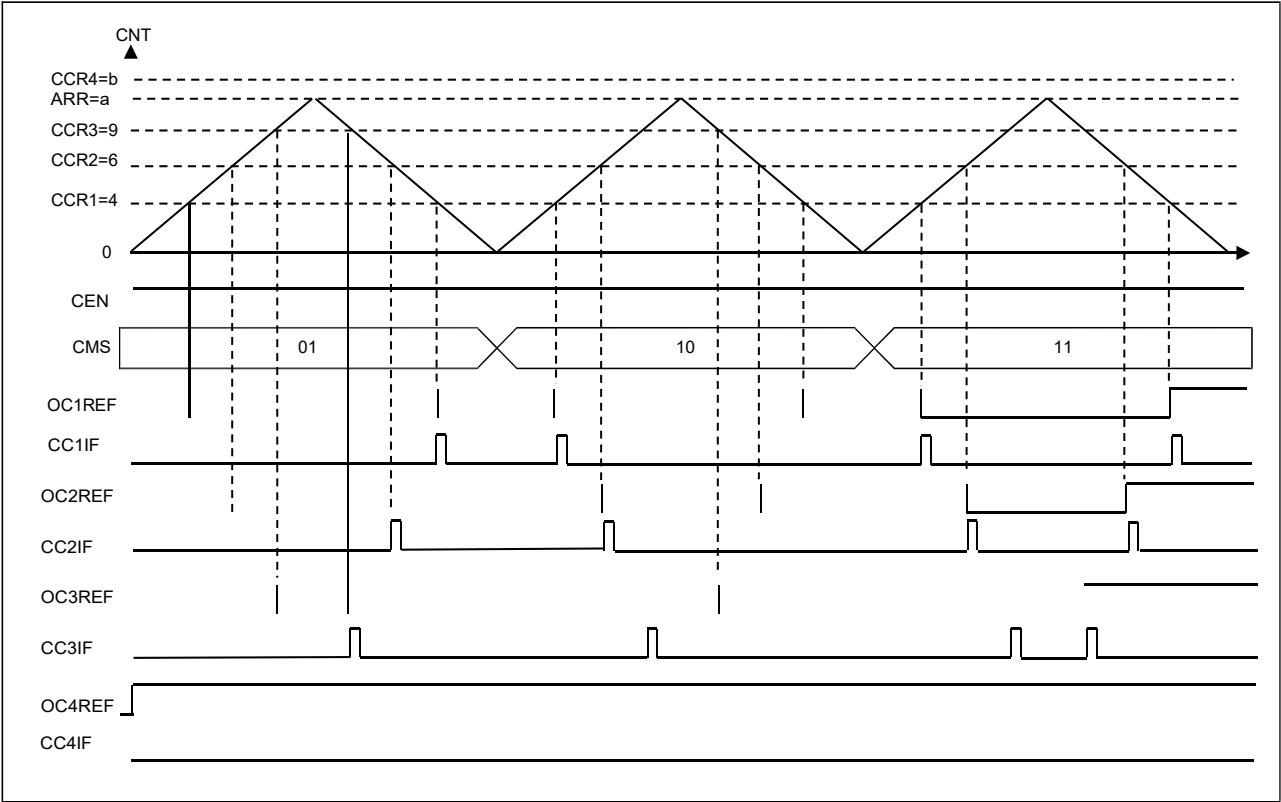


图 12-18 中央对齐 PWM 模式 1 的波形

注：

- 进入中央对齐模式时，使用当前的递增/递减计数配置，计数方向取决于当前的 DIR 的值。
- 在中央对齐模式下，最好不要修改计数器的值，可能会产生不可预知的结果。当计数器处于递增计数时，写入计数器的值 $> \text{TIMx_ARR}$ ，计数器会继续递增计数。直接写入 0 或 ARR，会立即更新计数方向，但不会产生更新事件。
- 建议使用中央对齐模式时，在启动计数器之前配置 TIMx_EGR 寄存器的 UG=1，产生一个软件更新，更新所有寄存器，启动计数器后不要修改计数器的值。

12.4.3.3.4 触发源输出

PWM 模式下，CCx_SETTRGO 信号可以用于触发 ADC 等模块。本章节仅介绍 CCx_SETTRGO 信号的触发逻辑，详细触发源选择，触发边沿选择等信息请参考 ADC 章节。

边沿对齐模式下，每次比较匹配（TIMx_CNT 当前计数值等于 TIMx_CCRx）时，CCx_SETTRGO 发生一次翻转。下图为边沿对齐递增计数模式下 CCx_SETTRGO 输出示例。

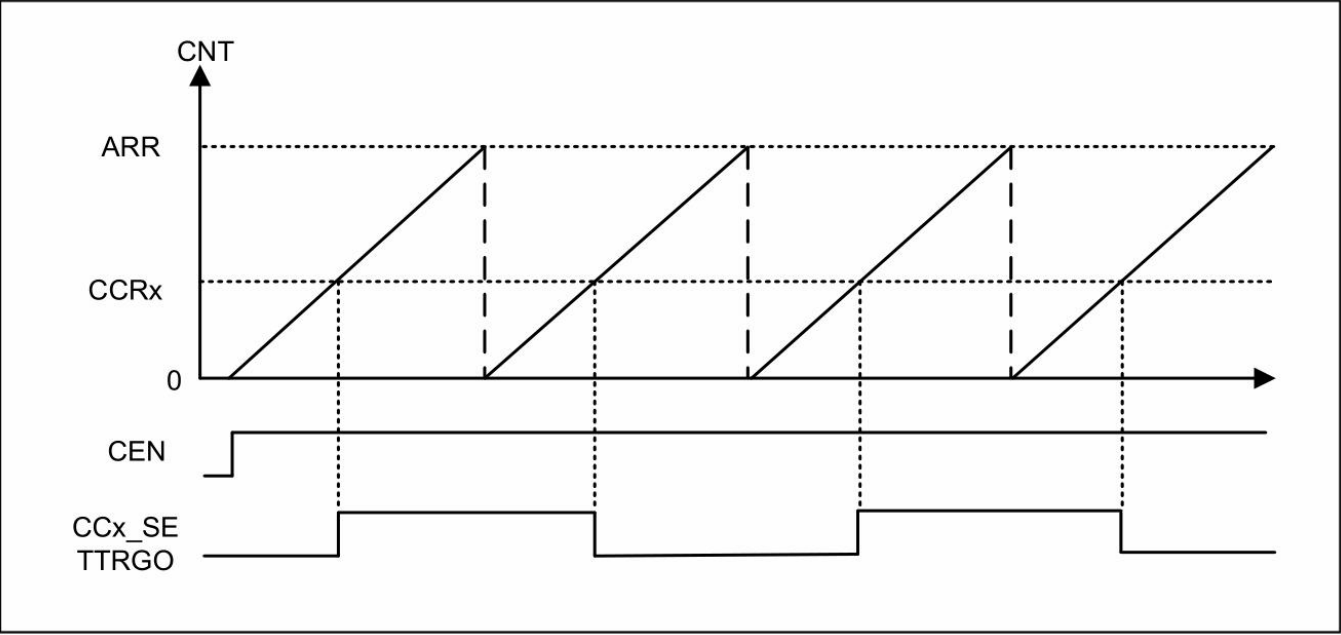


图 12-19 边沿对齐递增计数模式下 CC_x_SETTRGO 输出示例

中央对齐模式 1 下，在递减计数周期比较匹配时 CC_x_SETTRGO 发生一次翻转。中央对齐模式 2 下，在递增计数周期比较匹配时 CC_x_SETTRGO 发生一次翻转。中央对齐模式 3 下，在递增计数周期或递减计数周期比较匹配时 CC_x_SETTRGO 发生一次翻转。下图为中央对齐模式下 CC_x_SETTRGO 输出示例。

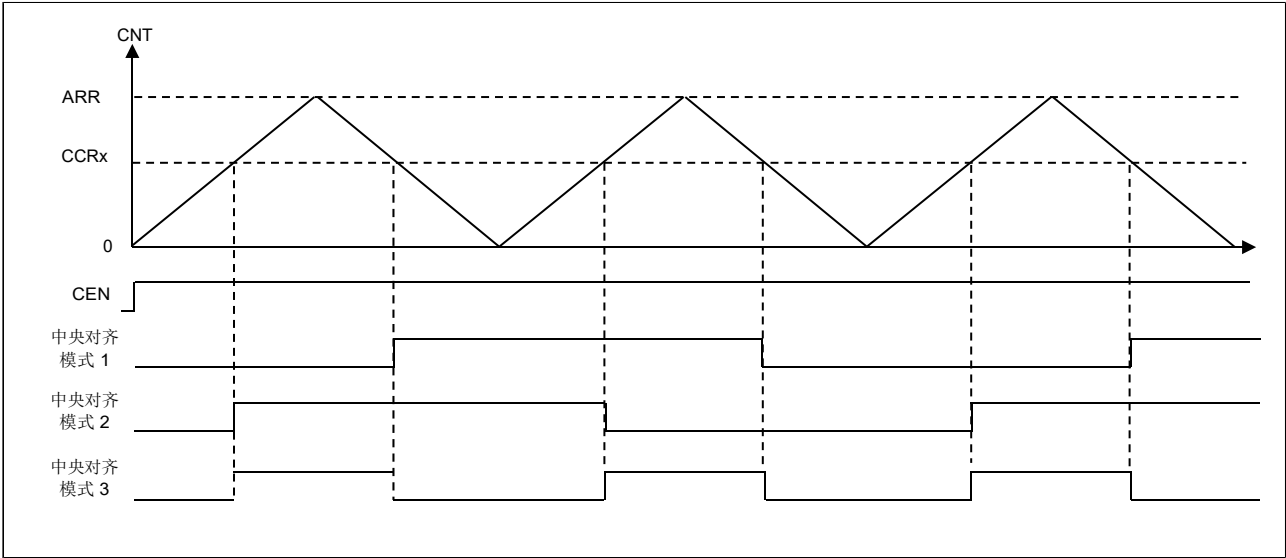


图 12-20 中央对齐模式下 CC_x_SETTRGO 输出示例

12.4.3.4 外部事件清除 OCxREF

在配置 TIMx_CCMR 寄存器的 OCxCE=1 时，OCxREF 可以被 ETR 输入端的有效电平拉低直到发生下一次更新事件（UEV）。此功能只能用于比较输出模式和 PWM 模式，不能用于强制输出模式。

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

- 1. 配置 TIMx_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
- 2. 配置 TIMx_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
- 3. 配置 TIMx_SMCR 寄存器 ETF[3: 0]和 ETP，配置 ETR 信号的触发极性和滤波宽度。 下图显示了当 ETR 输入变化触发 ETRF 为高时，对应不同 OCxCE 的值，OCxREF 信号的动作（PWM 模式）。

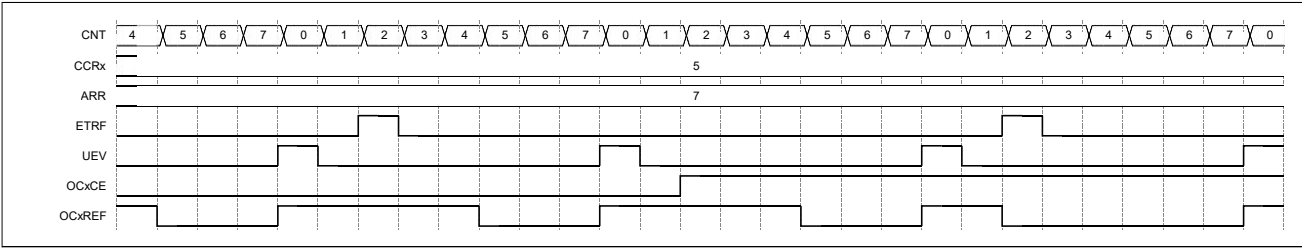


图 12-21 外部事件清除 OCxREF

12.4.3.5 单脉冲输出

单脉冲模式（OPM）下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 TIMx_CR1 寄存器的 OPM=1，选择单脉冲模式，触发信号有效沿或配置 CEN=1 都可以启动计数器，直到下个更新事件发生或配置 CEN=0 时，计数器停止计数。

产生脉冲的必要条件是比较值与计数器的初始值不同。所以在计数器启动之前的必要配置如下：

- 递增计数方式：计数器 $CNT < CCRx \leq ARR$ 。
- 递减计数方式：计数器 $CNT > CCRx$ 。

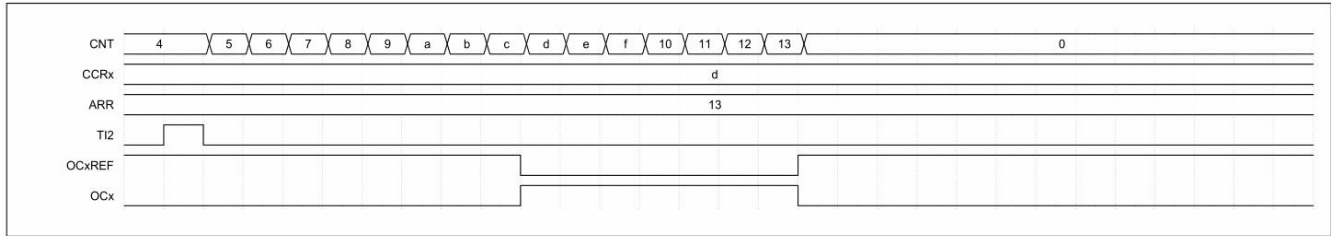


图 12-22 单脉冲模式

例如，在 TI2 检测到上升沿，延迟 t_{DELAY} 之后，在 OC2 上产生一个长度为 t_{PULSE} 的正脉冲。

配置 TI2FP2 作为触发源：

1. 配置 TIMx_CCMR1 寄存器中的 CC2S = 01，将 TI2FP2 映射到 TI2。
2. 配置 TIMx_CCER 寄存器中的 CC2P = 0，检测 TI2FP2 的上升沿。
3. 配置 TIMx_SMCR 寄存器中的 TS = 110，TI2FP2 作为从模式控制器的触发（TRGI）。
4. 配置 TIMx_SMCR 寄存器中的 SMS = 110，选择触发模式，TI2FP2 使能计数器工作。

OPM 的波形由 TIMx_ARR 和 TIMx_CCR1 决定（要考虑时钟频率和计数器预分频器）：由 TIMx_CCR1 寄存器的值和 CNT 初始值决定触发信号与单脉冲开始之间的延迟 t_{DELAY} ，TIMx_ARR - TIMx_CCR1 的值为脉冲的宽度 t_{PULSE} 。

下面是一个产生负脉冲的例子，即发生比较匹配时产生从 1 到 0 的波形，计数器达到预装载值时产生一个从 0 到 1 的波形：

1. 配置 TIMx_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
2. 配置 TIMx_CCER 寄存器 CC1P = 1，输出低电平有效。
3. 配置 TIMx_CCMR1 中 OC1PE = 1 和 TIMx_CR1 寄存器中 ARPE=1，使能预装载寄存器。
4. 配置 TIMx_CCR1 寄存器和 TIMx_ARR 寄存器。
5. 配置 TIMx_EGR 寄存器 UG=1 产生一个更新事件。
6. 等待在 TI2 上的一个外部触发事件。

此例中，TIMx_CR1 寄存器中的 DIR=0、CMS=0、OPM= 1，在下一个更新事件（当计数器从自动装载值返回到 0）时停止计数。

12.4.3.5.1 OCx 快速使能

OCx 快速使能，是单脉冲模式的一种特殊情况。在单脉冲模式下，通过设置 TIMx_CCMR 寄存器的 OCxFE=1，强制 OCxREF 直接响应激励而不是依赖计数器和比较值之间的比较结果，输出波形和比较匹配时的波形一样。这样可以去除比较的时间，快速输出比较结果。OCx 快速输出使能只在 PWM 模式下生效。

12.4.4 从模式

12.4.4.1 编码器接口

编码器接口模式就是计数器在 TI1 和 TI2 正交信号相互作用下计数，在输入源改变期间，计数方向被硬件自动修改。通过配置 TIMx_SMCR 寄存器 SMS 位可以选择输入源，根据输入源的不同，可以将编码器接口模式分为 3 种模式，SMS=001，编码器接口模式 1；SMS=010，编码器接口模式 2；SMS=011，编码器接口模式 3；三种模式具体计数操作如下表所示。两个输入 TI1 和 TI2 被用来作为正交编码器的接口。

编码器模式下，计数器开启之前必须先配置好 ARR 寄存器，因为使用编码器接口模式相当于使用了一个带有方向选择的外部时钟。计数器在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数（递增

计数和递减计数由外部时钟控制)。 注：编码器模式不支持外部时钟模式 2。

编码器接口模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 12-2 计数方向与编码器信号的关系

计数模式	相对电平（TI1FP1 相对于 TI2，TI2FP2 相对于 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
编码器接口模式 1（只在 TI2 计数）	高电平	-	-	递增计数	递减计数
编码器接口模式 1（只在 TI2 计数）	低电平	-	-	递减计数	递增计数
编码器接口模式 2（只在 TI1 计数）	高电平	递减计数	递增计数	-	-
编码器接口模式 2（只在 TI1 计数）	低电平	递增计数	递减计数	-	-
编码器接口模式 3（在 TI1 和 TI2 计数）	高电平	递减计数	递增计数	递增计数	递减计数
编码器接口模式 3（在 TI1 和 TI2 计数）	低电平	递增计数	递减计数	递减计数	递增计数

下例是计数器在编码器接口模式下的配置和时序图，从图中可以看出计数信号的产生和方向控制。 具体配置如下：

- 1. 配置 TIMx_CCMR 寄存器的 CC1S=01，将 IC1FP1 映射到 TI1 上。
- 2. 配置 TIMx_CCMR 寄存器的 CC2S =01，将 IC2FP2 映射到 TI2 上。
- 3. 配置 TIMx_CCER 寄存器的 CC1P =0，IC1 不反相，此时 IC1=TI1。
- 4. 配置 TIMx_CCER 寄存器的 CC2P =0，IC2 不反相，此时 IC1=TI2。
- 5. 配置 TIMx_SMCR 寄存器的 SMS =011，选择编码器模式 3，根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿计数。
- 6. 配置 TIMx_CR1 寄存器的 CEN =1，开启计数器。

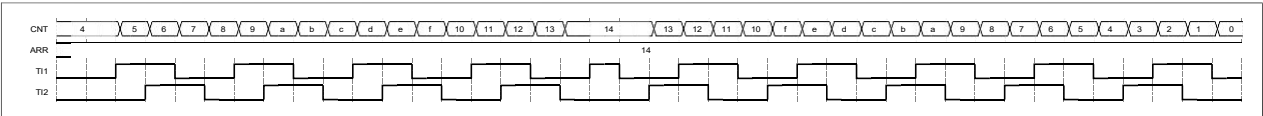


图 12-23 编码器模式下的计数器时序图

下图为当 IC1FP1 反相时计数器的时序图（CC1P = 1，其他配置不变）

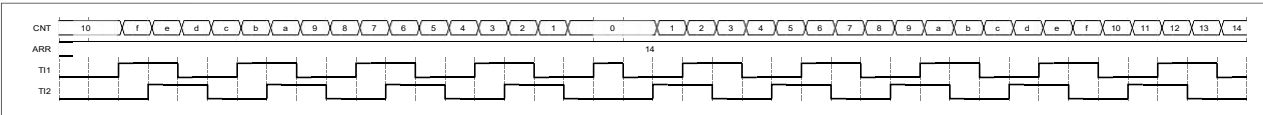


图 12-24 IC1FP1 反相编码器接口模式时序图

编码器接口模式下，计数器可以提供传感器当前位置的信息。通过使用另一个配置在捕获模式的定时器测量两个编码器事件的间隔周期来获得动态的信息（速度，加速度，减速度）。根据两个编码器事件的间隔周期，可以定期读取计数器。可以通过把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期性的并且可以由另一个定时器产生）来实现计数器的定期读取。若芯片内置 DMA，还可以通过 DMA 请求来读取它的值。

12.4.4.2 复位模式

配置 TIMx_SMCR 寄存器的 SMS=100，从模式选择复位模式。此模式下，TRGI 输入事件会使计数器清零重启。

例如，TI2 输入端的下降沿触发计数器重启：

1. 配置 TIMx_CCMR1 寄存器的 CC2S=01，CC2 通道被配置为输入模式；IC2 映射在 TI2 上，配置 TIMx_CCER 寄存器的 CC2P=1，检测下降沿。

2. 配置 TIMx_SMCR 寄存器的 SMS = 100，从模式选择复位模式；配置 TIMx_SMCR 寄存器的 TS = 110，选择滤波后的定时器输入 2（TI2FP2）作为同步计数器的触发输入。

3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

计数器的时钟源由内部时钟提供，当检测到 TI2 的下降沿，计数器被清零重启。此时触发器中断标记被硬件置 1。

下图为复位模式下 TIMx_ARR = 0x13 的时序图。

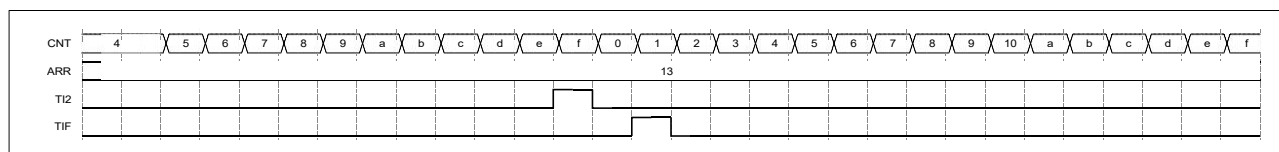


图 12-25 复位模式的控制时序图

12.4.4.3 门控模式

配置 TIMx_SMCR 寄存器 SMS=101，从模式选择门控模式。此模式下，根据 TIMx_CCER 寄存器 CCxP 的值来选择有效电平（0：高电平有效，1：低电平有效）。TRGI 输入为有效电平时，计数器始终开启，否则计数器停止（但不发生复位操作），计数器的开启和停止可控。

例如，计数器只在 TI1 为高时计数：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入模式，IC1 映射在 TI1 上；配置 TIMx_CCER 寄存器的 CC1P=0，检测 TI1 上的高电平。

2. 配置 TIMx_SMCR 寄存器的 SMS=101，从模式选择为门控模式；配置 TIMx_SMCR 寄存器

的 TS=101，选择滤波后的定时器输入 1（TI1FP1）作为同步计数器的触发输入。

3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

计数器的时钟源由内部时钟提供，当检测到 TI1 的高电平，计数器开始计数，当 TI1 为低电平时，计数器停止计数。计数器开启或停止都会将 TIF 置 1。

下图为门控模式下 TIMx_ARR=0xf 的时序图。

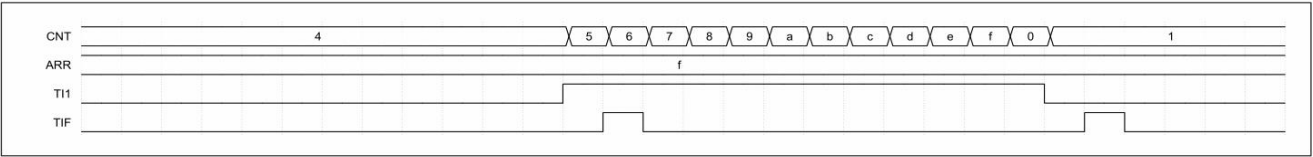


图 12-26 门控模式下的控制时序图

12.4.4.4 触发模式

配置 TIMx_SMCR 寄存器 SMS=110，从模式选择触发模式。根据 TIMx_CCER 寄存器 CCxP 的值来选择有效边沿（0：上升沿有效，1：下降沿有效），TRGI 输入为有效边沿时，计数器开始计数。计数器的启动可控，停止不可控。

例如，计数器在 TI1 输入的上升沿开始计数：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入模式，IC1 映射在 TI1 上，配置 TIMx_CCER 寄存器的 CC1P=0，检测上升沿。

2. 配置 TIMx_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIMx_SMCR 寄存器的 TS=101，选择滤波后的定时器输入 1（TI1FP1）作为计数器的触发输入。

3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。

计数器的时钟源由内部时钟提供，当检测到 TI1 的上升沿，计数器开始计数。 下图为触发模式下 TIMx_ARR=0xf 的时序图。

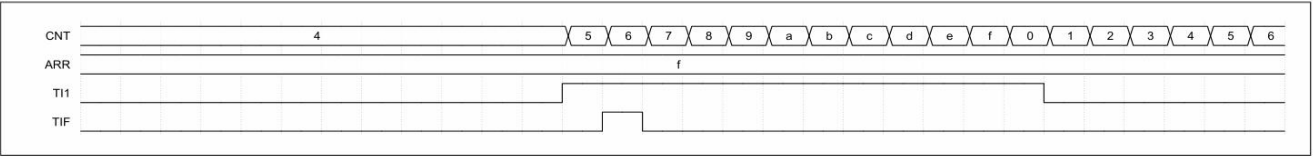


图 12-27 触发器模式下的控制时序图

12.4.4.5 外部时钟模式 2+从模式

当时钟源选择外部时钟模式 2、ETR 信号被用作外部时钟的输入时，可以与从模式一起使用。这种使用方式时，从模式仅支持复位模式、门控模式、触发模式，不支持外部时钟模式 1 和编码器模式。

例如，从模式选择触发模式，计数器在 ETR 的每一个上升沿计数一次：

1. 配置 TIMx_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIMx_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIMx_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIMx_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。

2. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上作为输入捕获源；配置 TIMx_CCER 寄存器的 CC1P=0，选择上升沿有效。

3. 配置 TIMx_SMCR 寄存器的 SMS = 110，从模式选择为触发模式。配置 TIMx_SMCR 寄存器的 TS = 101，选择 TI1 作为输入源。

4. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。计数器在 TI1 的上升沿开始计数，并将 TIF 置 1。ETR 信号的上升沿和计数器实际计数时钟间的延时取决于 ETR 输入端的同步电路设计。

下图为外部时钟模式 2+从模式（触发模式）下 TIMx_ARR=13 时的时序图。

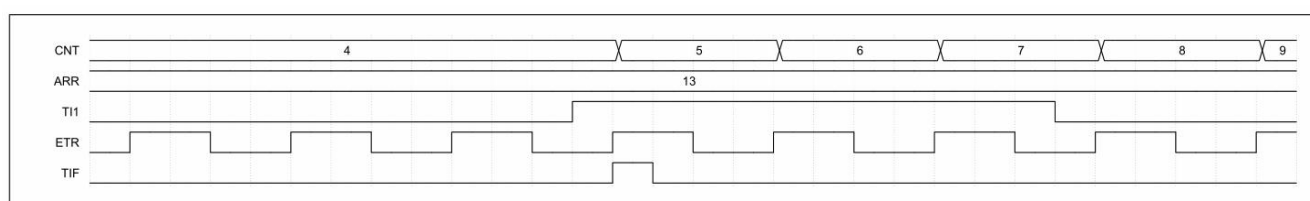


图 12-28 外部时钟模式 2+从模式（触发模式）控制时序图

12.4.5 定时器同步

不同的定时器在内部连接，可以实现定时器之间的级联或同步。详细描述请参考 TIM1 相关章节。

12.4.6 定时器异或

配置 TIMx_CR2 寄存器的 TI1S = 1，将 TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连接到 TI1 的输入端，用于定时器的所有输入模式。

例：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连接到 TI1 的输入端，采样 TI1 输入信号的有效沿，在 TI1 的上升沿来到时捕获当前计数器的值，锁存到 TIMx_CCR1 寄存器中。具体配置如下：

1. 配置 TIMx_CR2 寄存器的 TI1S=1，配置定时器的三个输入经异或后连接到 TI1 输入通道。
2. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上。
3. 配置 TIMx_CCMR1 寄存器的 IC1F[3: 0]，配置数字滤波器的滤波宽度（按需配置）。
4. 配置 TIMx_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
5. 配置 TIMx_CCMR1 寄存器的 IC1PSC，选择预分频系数。
6. 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。

7. 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器。

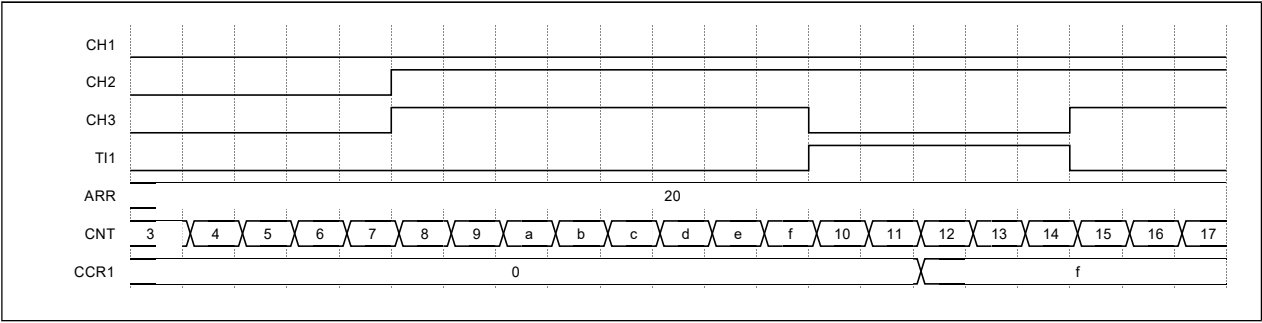


图 12-29（TI1 异或输入）输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

12.4.7 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIMx_STOP=1，TIMx 计数器停止计数。（详见调 试 章节）

12.4.8 中断

TIMx 的中断包括：捕获/比较 1 中断、捕获/比较 2 中断、捕获/比较 3 中断、捕获/比较 4 中断、更 新 中断、触发中断，当相应的中断使能位打开，发生相应的事件时，产生相应的中断。

表 12-3 中断事件一览表

中断事件	标志位	使能位
捕获/比较 1 中断	CC1IF	CC1IE
捕获/比较 2 中断	CC2IF	CC2IE
捕获/比较 3 中断	CC3IF	CC3IE
捕获/比较 4 中断	CC4IF	CC4IE
更新中断	UIF	UIE
触发中断	TIF	TIE

12.4.9 DMA

TIMx 能够在发生单个事件时生成一个或连续多个 DMA 请求。主要目的是在没有软件开销的情况 下，多次重新编程 TIMx 的一部分寄存器，也可以用于按周期读取数个寄存器。

TIMx_DCR 和 TIMx_DMAR 寄存器跟 DMA 模式相关。DMA 控制器的目标是唯一的，必须指向 TIMx_DMAR 寄存器。开启 DMA 使能后，在给定的 TIMx 事件发生时，TIMx 会给 DMA 发送请求。对 TIMx_DMAR 寄存器的每次写操作都被重定向到一个 TIMx 寄存器。

TIMx_DCR 寄存器的 DBL 位定义了 DMA 连续传送的长度，即传输寄存器数量。当对

TIMx_DMAR 进行读写操作时，定时器识别 DBL，确定需要传输的寄存器数量。TIMx_DCR 寄存器的 DBA 位定义了 DMA 传输的基地址，定义从 TIMx_CR1 寄存器地址开始的偏移量（00000 为 TIMx_CR1、00001 为 TIMx_CR2 00110 为 TIMx_CCMR1 等）。

例：DMA 连续传送模式用于在发生更新事件时更新 CCR1、CCR2、CCR3 寄存器的内容。具体配置如下：

1. 配置相应的 DMA 通道。
2. 配置 TIMx_DCR 寄存器的 DBA=01101，配置 DMA 的基地址，选择偏移地址为 TIMx_CCR1 寄存器的地址。
3. 配置 TIMx_DCR 寄存器的 DBL=00010，配置传输长度为 3。
4. 配置 TIMx_DIER 寄存器的 UDE=1，允许更新事件的 DMA 请求。
5. 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器。
6. 使能 DMA 通道。

此例中发生一次更新事件，DMA 将相应存储器地址中准备好的数据传输到 CCR1、CCR2、CCR3 寄存器中，或将 CCR1、CCR2、CCR3 寄存器的值传输到相应存储器地址中。

12.5 寄存器

表 12-4 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x04	TIMx_CR2	控制寄存器 2	0x0000
0x08	TIMx_SMCR	从模式控制寄存器	0x0000 0000
0x0C	TIMx_DIER	DMA/中断使能寄存器（DMA 部分仅适用于有内置 DMA 的芯片）	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIMx_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIMx_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIMx_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIMx_CCR4	捕获/比较寄存器 4	0x0000
0x48	TIMx_DCR	DMA 控制寄存器（仅适用于有内置 DMA 的芯片）	0x0000
0x4C	TIMx_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000
0x50	TIMx_OR	输入选项寄存器	0x0000

12.5.1 TIMx_CR1 控制寄存器 1

偏移地址: 0x00 复位值:

0x0000

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

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。
9: 8	CKD	时钟分频 (clock division) 定义定时器时钟 (INT_CK) 频率与死区时间计数器、数字滤波器 (ETR, TIx) 所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT_CK}$ 01: $t_{DTS} = 2 \times t_{INT_CK}$ 10: $t_{DTS} = 4 \times t_{INT_CK}$ 11: 保留, 不要使用这个配置
7	ARPE	自动重载预装载使能 (Auto-reload preload enable) 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 5	CMS	中央对齐模式选择 (Center-aligned mode selection) 00: 边沿对齐模式。计数方向取决于 DIR 位 01: 中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式, 只在计数器递减计数时比较中断标志位被置 1 10: 中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式, 只在计数器递增计数时比较中断标志位被置 1 11: 中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式, 在计数器递增和递减计数时比较中断标志位均被置 1 注: 计数过程中, 不允许更改此位。
4	DIR	计数方向 (Direction) 0: 计数器递增计数 1: 计数器递减计数 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。
3	OPM	单脉冲模式 (One pulse mode) 0: 禁止单脉冲模式, 在发生更新事件时, 计数器继续计数 1: 使能单脉冲模式, 在发生下一次更新事件或软件清除 CEN 位时, 计数器停止计数
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC、CCRx) 保持值不变。如果设置了 EGR_UG 位为 1, 或者从模式控制器接收到硬件复位, 计数器和预分频器被初始化。

Bit	Field	Description
0	CEN	计数器使能 (Counter enable) 0: 禁止计数器 1: 使能计数器 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

12.5.2 TIMx_CR2 控制寄存器 2

偏移地址: 0x04 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S rw	MMS rw			CCDS rw	Reserved		

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	TI1S	TI1 选择 (TI1 selection) 0: TIMx_CH1 管脚连到 TI1 输入 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入
6: 4	MMS	主模式选择 (Master mode selection) 这些位控制 TRGO 信号的选择, 用于选择在主模式下送到从定时器的同步信息: 000: 复位 TIMx_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001: 使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO), 计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新 更新事件被选为 TRGO。 011: 捕获/比较脉冲 通道 1 发生一次捕获或一次比较成功时, 触发输出送出一个 TRGO 信号。 100: 比较 OC1REF 信号被用于作为触发输出 (TRGO) 101: 比较 OC2REF 信号被用于作为触发输出 (TRGO) 110: 比较 OC3REF 信号被用于作为触发输出 (TRGO) 111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)
3	CCDS	DMA 请求源选择 (Capture/compare DMA selection) 0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求 1: 发生更新事件时, 发送 CCx 的 DMA 请求 注: 仅适用于有内置 DMA 的产品
2: 0	Reserved	保留, 必须保持复位值。

12.5.3 TIMx_SMCR 从模式控制寄存器

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

ETP	ECE	ETPS	ETF	MSM	TS[2:0]	OCCS	SMS
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值。
15	ETP	外部触发极性 (External trigger polarity) 该位选择 ETR 信号的极性。 0: 高电平或上升沿有效 1: 低电平或下降沿有效 注: 仅适用于支持外部触发的产品
14	ECE	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2。 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2, ETRF 信号上的任意有效沿驱动计数器计数 注 1: 仅适用于支持外部触发的产品。 注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。 注 3: TS ≠ 111 时, 复位模式, 门控模式和触发模式可以与外部时钟模式 2 同时使用。 注 4: 同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入是 ETR。
13: 12	ETPS	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须低于 TIMx 主时钟 PCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8 注: 仅适用于支持外部触发的产品。
11: 8	ETF	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT_CK}$, N=2 0010: 采样频率 $f_{sampling}=f_{INT_CK}$, N=4 0011: 采样频率 $f_{sampling}=f_{INT_CK}$, N=8 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, N=6 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, N=8 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, N=6 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, N=8 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, N=6 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, N=8 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, N=5 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, N=6 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, N=8 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, N=5 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, N=6 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, N=8 注: 仅适用于支持外部触发的产品。
7	MSM	主/从模式 (Master/slave mode) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟, 以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步, 该功能可以把几个定时器同步到一个单一的外部事件。

Bit	Field	Description
6: 4	TS[2:0]	<p>触发选择 (Trigger selection)</p> <p>000: 内部触发 0 (ITR0)</p> <p>001: 内部触发 1 (ITR1)</p> <p>010: 内部触发 2 (ITR2)</p> <p>011: 内部触发 3 (ITR3)</p> <p>100: TI1 的边沿检测器 (TI1F_ED)</p> <p>101: 滤波后的定时器输入 1 (TI1FP1)</p> <p>110: 滤波后的定时器输入 2 (TI2FP2)</p> <p>111: 外部触发输入 (ETR)</p> <p>others: 保留</p> <p>更多有关 ITRx 的细节, 参见内部触发连接一览表。</p> <p>注: 从模式使能后这些位不能修改。</p>
3	OCCS	<p>比较输出信号 (OCxREF) 清除选择 (Output compare clear selection)</p> <p>在 PWM 模式下, 清除比较输出 (OCxREF)。</p> <p>0: 外部触发信号作为清除信号</p> <p>1: 比较器 (COMP) 输出作为清除信号 注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。</p>
2: 0	SMS	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。</p> <p>000: 关闭从模式 - 如果 CEN = 1, 则预分频器直接由内部时钟驱动。</p> <p>001: 编码器模式 1- 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿递增/递减计数。</p> <p>010: 编码器模式 2- 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿递增/递减计数。</p> <p>011: 编码器模式 3 - 根据另一个输入的电平, 计数器在 TI1FP1 和 TI2FP2 的边沿递增/递减计数。</p> <p>100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新事件。</p> <p>101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器开始计数。当触发输入变为低时, 计数器停止计数 (但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式 - 计数器在触发输入 TRGI 的上升沿启动 (但不复位), 只有计数器的启动是受控的。</p> <p>111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。</p> <p>注: 如果 TI1F_ED 被选为触发输入 (TS = 100) 时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。</p>

表 12-5 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	-	TIM3_TRGO	TIM17_OC1REF
TIM3	TIM1_TRGO	-	TIM16_OC1REF	TIM14_OC1REF

12.5.4 TIMx_DIER DMA/中断使能寄存器

偏移地址: 0x0C 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	r/w		r/w	r/w	r/w	r/w	r/w		r/w		r/w	r/w	r/w	r/w	r/w

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14	TDE	允许触发 DMA 请求（Trigger DMA request enable） 0：禁止触发 DMA 请求 1：允许触发 DMA 请求 注：仅适用于有内置 DMA 的产品。
13	Reserved	保留，必须保持复位值。
12	CC4DE	允许捕获/比较 4 的 DMA 请求（Capture/Compare 4 DMA request enable） 0：禁止捕获/比较 4 的 DMA 请求 1：允许捕获/比较 4 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
11	CC3DE	允许捕获/比较 3 的 DMA 请求（Capture/Compare 3 DMA request enable） 0：禁止捕获/比较 3 的 DMA 请求 1：允许捕获/比较 3 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
10	CC2DE	允许捕获/比较 2 的 DMA 请求（Capture/Compare 2 DMA request enable） 0：禁止捕获/比较 2 的 DMA 请求 1：允许捕获/比较 2 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
9	CC1DE	允许捕获/比较 1 的 DMA 请求（Capture/Compare 1 DMA request enable） 0：禁止捕获/比较 1 的 DMA 请求 1：允许捕获/比较 1 的 DMA 请求 注：仅适用于有内置 DMA 的产品。
8	UDE	允许更新 DMA 请求（Update DMA request enable） 0：禁止更新 DMA 请求 1：允许更新 DMA 请求 注：仅适用于有内置 DMA 的产品。
7	Reserved	保留，必须保持复位值。
6	TIE	允许触发中断（Trigger interrupt enable） 0：禁止触发中断 1：允许触发中断
5	Reserved	保留，必须保持复位值。
4	CC4IE	允许捕获/比较 4 中断（Capture/Compare 4 interrupt enable） 0：禁止捕获/比较 4 中断 1：允许捕获/比较 4 中断
3	CC3IE	允许捕获/比较 3 中断（Capture/Compare 3 interrupt enable） 0：禁止捕获/比较 3 中断 1：允许捕获/比较 3 中断
2	CC2IE	允许捕获/比较 2 中断（Capture/Compare 2 interrupt enable） 0：禁止捕获/比较 2 中断 1：允许捕获/比较 2 中断
1	CC1IE	允许捕获/比较 1 中断（Capture/Compare 1 interrupt enable） 0：禁止捕获/比较 1 中断 1：允许捕获/比较 1 中断
0	UIE	允许更新事件中断（Update interrupt enable） 0：禁止更新事件中断 1：允许更新事件中断

12.5.5 TIMx_SR 状态寄存器

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r_w0c						r_w0c		r_w0c				

Bit	Field	Description
15: 13	Reserved	保留，必须保持复位值。
12	CC4OF	捕获/比较 4 重复捕获标记 (Capture/Compare 4 overcapture flag) 参考 CC1OF 描述。
11	CC3OF	捕获/比较 3 重复捕获标记 (Capture/Compare 3 overcapture flag) 参考 CC1OF 描述。
10	CC2OF	捕获/比较 2 重复捕获标记 (Capture/Compare 2 overcapture flag) 参考 CC1OF 描述。
9	CC1OF	捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当通道 1 被配置为输入捕获，CC1IF 已经为 1 后，捕获事件再次发生时，该标记可由硬件置 1。写 0 可清除该位。 0：无重复捕获产生 1：重复捕获产生
8: 7	Reserved	保留，必须保持复位值。
6	TIF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置 1。它由软件清 0。 0：无触发器事件产生 1：触发器中断产生
5	Reserved	保留，必须保持复位值。
4	CC4IF	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式： 当计数器值与比较值匹配时该位由硬件置 1（在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位）。它由软件清 0。 0：无匹配发生 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式： 当发生捕获事件时该位由硬件置 1，由软件清 0 或读取 TIMx_CCR1 的值清 0。 0：无输入捕获产生 1：计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0：无更新中断发生 1：发生更新中断 当寄存器被更新时该位由硬件置 1： - 若 TIMx_CR1 寄存器的 UDIS=0，且 REP_CNT=0，当计数器产生上溢/下溢事件时。 -若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，从模式控制器产生更新事件时。

12.5.6 TIMx_EGR 事件产生寄存器

偏移地址：0x14 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									W		W	W	W	W	W

Bit	Field	Description
15: 7	Reserved	保留，必须保持复位值。
6	TG	产生触发事件（Trigger generation） 0：无动作 1：产生触发事件，TIMx_SR 寄存器的 TIF = 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA，由硬件自动清 0。
5	Reserved	保留，必须保持复位值。
4	CC4G	产生捕获/比较 4 事件（Capture/Compare 4 generation） 参考 CC1G 描述。
3	CC3G	产生捕获/比较 3 事件（Capture/Compare 3 generation） 参考 CC1G 描述。
2	CC2G	产生捕获/比较 2 事件（Capture/Compare 2 generation） 参考 CC1G 描述。
1	CC1G	产生通道 1 捕获/比较事件（Capture/Compare 1 generation） 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0：无动作 1：通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出：CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入：当前的计数器值被捕获至 TIMx_CCR1 寄存器，CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF = 1。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0，如果选择了中央对齐或递增计数模式，计数器被清 0；如果选择递减计数模式，计数器将载入自动重载值。预分频计数器将同时被清除。

12.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址：0x18 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	CC2S		OC1CE	OC1M			OC1PE	OC1FE	CC1S	
IC2F				IC2PSC		CC2S		IC1F				IC1PSC		CC1S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CCxS 定义。该寄存器 CCxS 以外其它位的作用在输入模式和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx

描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能 (Output compare 2 clear enable) 参考 OC1CE 的描述。
14: 12	OC2M	通道 2 比较输出模式 (Output compare 2 mode) 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能 (Output compare 2 preload enable) 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能 (Output compare 2 fast enable) 参考 OC1FE 的描述。
9: 8	CC2S	通道 2 捕获/比较选择 (Capture/Compare 2 selection) 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00: 通道 2 被配置为输出 01: 通道 2 被配置为输入，IC2 映射在 TI2 上 10: 通道 2 被配置为输入，IC2 映射在 TI1 上 11: 通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)
7	OC1CE	通道 1 比较输出清零使能 (Output compare 1 clear enable) 0: OC1REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入有效电平时，OC1REF 清零
6: 4	OC1M	通道 1 比较输出模式 (Output compare 1 mode) 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。 000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为高电平。 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为低电平。 011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100: 强制为低。强制 OC1REF 为低电平。 101: 强制为高。强制 OC1REF 为高电平。 110: PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平，否则为高电平。 111: PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平，否则为高电平；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。 注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。

Bit	Field	Description
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable)</p> <p>0: 禁止 TIMx_CCR1 寄存器的预装载功能, 写入 TIMx_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时生效。</p> <p>注: 若该位置 1, 在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1), 是否设定预装载寄存器无影响; 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。</p>
2	OC1FE	<p>通道 1 比较输出快速使能 (Output compare 1 fast enable)</p> <p>该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。</p> <p>0: 禁止通道 1 比较输出快速使能</p> <p>1: 开启通道 1 比较输出快速使能</p>
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入, IC1 映射在 TI1 上</p> <p>10: 通道 1 被配置为输入, IC1 映射在 TI2 上</p> <p>11: 通道 1 被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)</p>

输入捕获模式:

Bit	Field	Description
15: 12	IC2F	<p>输入捕获 2 滤波器 (Input capture 2 filter)</p> <p>参考 IC1F 的描述</p>
11: 10	IC2PSC	<p>输入/捕获 2 预分频器 (Input capture 2 prescaler)</p> <p>参考 IC1PSC 的描述</p>
9: 8	CC2S	<p>通道 2 捕获/比较选择 (Capture/Compare 2 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 2 被配置为输出</p> <p>01: 通道 2 被配置为输入, IC2 映射在 TI2 上</p> <p>10: 通道 2 被配置为输入, IC2 映射在 TI1 上</p> <p>11: 通道 2 被配置为输入, IC2 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)</p>

Bit	Field	Description
7: 4	IC1F	<p>通道 1 输入捕获滤波器 (Input capture 1 filter)</p> <p>数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, 以 f_{DTS} 采样</p> <p>0001: 采样频率 $f_{sampling}=f_{INT_CK}$, N=2</p> <p>0010: 采样频率 $f_{sampling}=f_{INT_CK}$, N=4</p> <p>0011: 采样频率 $f_{sampling}=f_{INT_CK}$, N=8</p> <p>0100: 采样频率 $f_{sampling}=f_{DTS}/2$, N=6</p> <p>0101: 采样频率 $f_{sampling}=f_{DTS}/2$, N=8</p> <p>0110: 采样频率 $f_{sampling}=f_{DTS}/4$, N=6</p> <p>0111: 采样频率 $f_{sampling}=f_{DTS}/4$, N=8</p> <p>1000: 采样频率 $f_{sampling}=f_{DTS}/8$, N=6</p> <p>1001: 采样频率 $f_{sampling}=f_{DTS}/8$, N=8</p> <p>1010: 采样频率 $f_{sampling}=f_{DTS}/16$, N=5</p> <p>1011: 采样频率 $f_{sampling}=f_{DTS}/16$, N=6</p> <p>1100: 采样频率 $f_{sampling}=f_{DTS}/16$, N=8</p> <p>1101: 采样频率 $f_{sampling}=f_{DTS}/32$, N=5</p> <p>1110: 采样频率 $f_{sampling}=f_{DTS}/32$, N=6</p> <p>1111: 采样频率 $f_{sampling}=f_{DTS}/32$, N=8</p>
3: 2	IC1PSC	<p>通道 1 输入/捕获预分频器 (Input capture 1 prescaler)</p> <p>该位定义了 IC1 的预分频系数。当 CC1E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入, IC1 映射在 TI1 上</p> <p>10: 通道 1 被配置为输入, IC1 映射在 TI2 上</p> <p>11: 通道 1 被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)</p>

12.5.8 TIMx_CCMR2 捕获/比较模式寄存器 2

偏移地址: 0x1C 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	CC4S		OC3CE	OC3M			OC3PE	OC3FE	CC3S	
IC4F				IC4PSC		CC4S		IC3F				IC3PSC		CC3S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

比较输出模式:

Bit	Field	Description
15	OC4CE	<p>通道 4 比较输出清零使能 (Output compare 4 clear enable)</p> <p>参考 OC3CE 的描述</p>

Bit	Field	Description
14: 12	OC4M	通道 4 比较输出模式 (Output compare 4 mode) 参考 OC3M 的描述
11	OC4PE	通道 4 比较输出预装载使能 (Output compare 4 preload enable) 参考 OC3PE 的描述
10	OC4FE	通道 4 比较输出快速使能 (Output compare 4 fast enable) 参考 OC3FE 的描述
9: 8	CC4S	通道 4 捕获/比较选择 (Capture/Compare 4 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入, IC4 映射在 TI4 上 10: 通道 4 被配置为输入, IC4 映射在 TI3 上 11: 通道 4 被配置为输入, IC4 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)
7	OC3CE	通道 3 比较输出清 0 使能 (Output compare 3 clear enable) 0: OC3REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入有效电平时, OC3REF 清零
6: 4	OC3M	通道 3 比较输出模式 (Output compare 3 mode) 该位定义了输出参考信号 OC3REF 的动作, 而 OC3REF 决定了 OC3 的值。OC3REF 是高电平有效, 而 OC3 的有效电平取决于 CC3P 位。 000: 冻结。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为高电平 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为低电平 011: 匹配时翻转。当 TIMx_CCR3=TIMx_CNT 时, 翻转 OC3REF 的电平 100: 强制为低。强制 OC3REF 为低电平 101: 强制为高。强制 OC3REF 为高电平 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平;在递减计数时, 当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平;在递减计数时, 当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平。 注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能 (Output compare 3 preload enable) 0: 禁止 TIMx_CCR3 寄存器的预装载功能, 写入 TIMx_CCR3 寄存器的数值立即生效 1: 开启 TIMx_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR3 的预装载值在更新事件到来时生效 注: 仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM= 1), 无需设定预装载寄存器, 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。

Bit	Field	Description
2	OC3FE	通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。 0: 禁止通道 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	输入捕获 4 滤波器 (Input capture 4 filter) 参考 IC3F 的描述
11: 10	IC4PSC	输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述
9: 8	CC4S	通道 4 捕获/比较选择 (Capture/Compare 4 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入, IC4 映射在 TI4 上 10: 通道 4 被配置为输入, IC4 映射在 TI3 上 11: 通道 4 被配置为输入, IC4 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)
7: 4	IC3F	通道 3 输入捕获滤波器 (Input capture 3 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$

Bit	Field	Description
3: 2	IC3PSC	通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 CC3E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

12.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20 复位值:

0x0000

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

Bit	Field	Description
15	CC4NP	通道 4 输入捕获极性 (Capture 4 polarity) 参考 CC1NP 的描述。
14	Reserved	保留, 必须保持复位值。
13	CC4P	通道 4 输入/捕获输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	通道 4 输入/捕获输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	通道 3 输入捕获极性 (Capture 3 polarity) 参考 CC1NP 的描述。
10	Reserved	保留, 必须保持复位值。
9	CC3P	通道 3 输入/捕获输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	通道 3 输入/捕获输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	通道 2 输入捕获极性 (Capture 2 polarity) 参考 CC1NP 的描述。
6	Reserved	保留, 必须保持复位值。
5	CC2P	通道 2 输入/捕获输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	通道 2 输入/捕获输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。

Bit	Field	Description
3	CC1NP	通道 1 输入捕获极性（Capture 1 polarity） 通道 1 配置为输出时，此位无效。 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 ICx 极性/电平选择表。
2	Reserved	保留，必须保持复位值。
1	CC1P	通道 1 输入/捕获输出极性（Capture/Compare 1 output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 ICx 极性/电平选择表。
0	CC1E	通道 1 输入/捕获输出使能（Capture/Compare 1 output enable） 通道 1 配置为输出时： 0：关闭。OC1 禁止输出 1：开启。OC1 信号输出到对应的输出引脚 CC1 通道配置为输入： 该位决定了输入捕获功能是否启用。 0：捕获禁止 1：捕获使能

输入模式下，ICx 的极性/电平选择如下表所示：

表 12-6 ICx 极性/电平选择表

CCxP	CCxNP	ICx 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/低电平有效
0	1	保留

12.5.10 TIMx_CNT 计数器

偏移地址：0x24 复位值：

0x0000

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

Bit	Field	Description
15: 0	CNT	计数器的值（Count value）

12.5.11 TIMx_PSC 预分频器

偏移地址：0x28 复位值：

0x0000

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

PSC

rw

Bit	Field	Description
15: 0	PSC	预分频器的值 (Prescaler value) 计数器的时钟频率 (ck_cnt) = $f_{CK_PSC} / (PSC + 1)$ 当发生更新事件时, PSC 的值装入当前预分频寄存器。

12.5.12 TIMx_ARR 自动预装载寄存器

偏移地址: 0x2C 复位值:

0x0000

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

Bit	Field	Description
15: 0	ARR	自动预装载值 (Auto-reload value) 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时, 计数器不工作。

12.5.13 TIMx_CCR1 捕获/比较寄存器 1

偏移地址: 0x34 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15: 0	CCR1	通道 1 捕获/比较的值 (Capture/Compare 1 value) 通道 1 配置为输入: 上一次捕获事件发生时捕获的计数器值存放于 CCR1 (此时 CCR1 寄存器为只读)。 通道 1 配置为输出: 如果在 TIMx_CCMR1 寄存器 (OC1PE 位) 中未选择预装载功能, 写入的数值会立即传输至对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时, 此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。

12.5.14 TIMx_CCR2 捕获/比较寄存器 2

偏移地址：0x38 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
15: 0	CCR2	通道 2 捕获/比较的值 (Capture/Compare 2 value) 参考 CCR1 的描述。

12.5.15 TIMx_CCR3 捕获/比较寄存器 3

偏移地址：0x3C 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
15: 0	CCR3	通道 3 捕获/比较的值 (Capture/Compare 3 value) 参考 CCR1 的描述。

12.5.16 TIMx_CCR4 捕获/比较寄存器 4

偏移地址：0x40 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
15: 0	CCR4	通道 4 捕获/比较的值 (Capture/Compare 4 value) 参考 CCR1 的描述。

12.5.17 TIMx_DCR DMA 控制寄存器

偏移地址：0x48 复位值：

0x0000

注：本寄存器仅适用于有内置 DMA 的产品，详见 12.4.9 DMA 章节。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DBL rw					Res.			DBA rw				

Bit	Field	Description
15: 13	Reserved	保留，必须保持复位值。
12: 8	DBL	DMA 连续传送长度（DMA burst length） 这些位定义了 DMA 在连续模式下的访问寄存器的数量 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输
7: 5	Reserved	保留，必须保持复位值。
4: 0	DBA	DMA 基地址（DMA base address） 这些位定义了 DMA 在连续模式下访问 TIMx_DMAR 寄存器的第一个地址。DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移值： 00000: TIMx_CR1 00001: TIMx_CR2 00010: TIMx_SMCR

12.5.18 TIMx_DMAR 连续模式 DMA 地址寄存器

偏移地址：0x4C 复位值：

0x0000

注：本寄存器仅适用于有内置 DMA 的产品，详见 12.4.9 DMA 章节。

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

Bit	Field	Description
15: 0	DMAB	DMA 连续传送地址寄存器（DMA address register for burst accesses） 对 TIMx_DMAR 寄存器的读写操作会导致对以下地址所在寄存器的存取操作： TIMx_CR1 地址 + DBA + DMA 索引，其中 TIMx_CR1 地址是 TIMx_CR1 寄存器所在的地址，DBA 是 TIMx_DCR 寄存器中定义的基地址，DMA 索引是 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL 值。

12.5.19 TIMx_OR 输入选项寄存器

偏移地址：0x50

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI4_RMP		Reserved				ETR_RMP	
								rw						rw	

Bit	Field	Description
15:8	Reserved	保留，必须保持复位值。
7:6	TI4_RMP	Timer3 TI4 复用 00: CH4 GPIO 输入 01: LSI 时钟输入 10: 保留 11: HSE_CLK_D128 时钟输入
5:2	Reserved	保留，必须保持复位值。
1:0	ETR_RMP	Timer3 ETR 复用 00: ETR GPIO 输入 01: LSI 时钟输入 10: 保留 11: HSE_CLK_D128 时钟输入

13 TIM13/14 基本定时器

16 位单通道基本定时器包含 TIM13、TIM14，具有相同的功能，本产品仅搭载 TIM14。本章节统一表述为 TIMx，部分示意图以 TIM14 为例展示。

13.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 16 位自动重载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。基本定时器具有多种用途，如输入功能（测量输入信号的脉冲宽度、频率等），输出功能（PWM 输出、比较输出等）。

13.2 功能框图

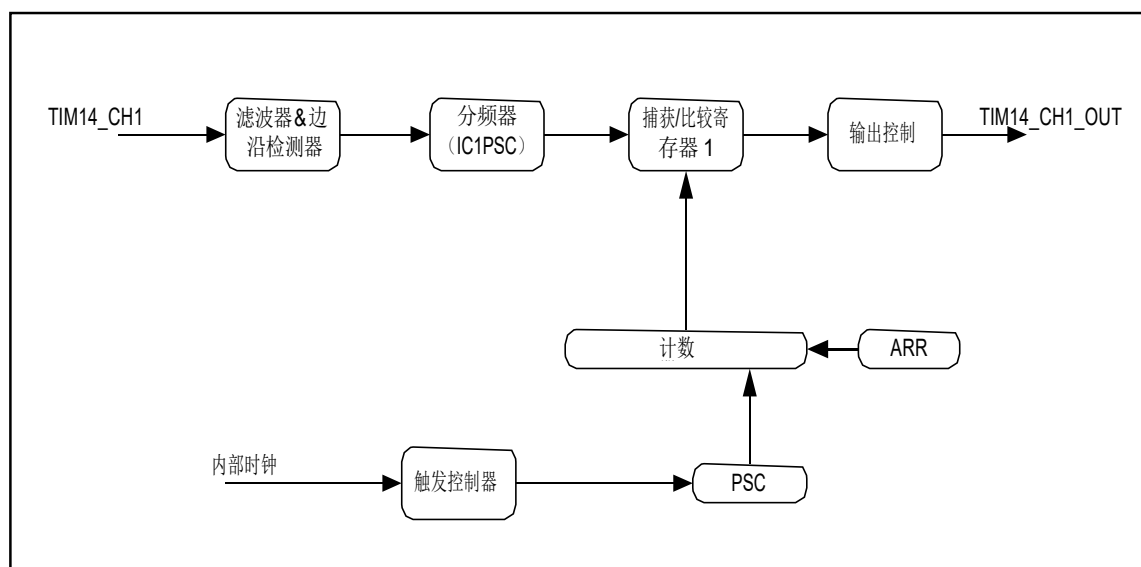


图 13-1 TIMx 结构图

上图为 TIMx 的结构框图，主要由输入单元、输出单元、时基单元、捕获/比较模块等结构组成。

13.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 16 位自动重载计数器（计数方向：递增）
- 输入捕获：输入信号的脉冲宽度、周期的测量

- 比较输出（控制输出波形或指示定时器已经计时结束）
- 单脉冲模式
- PWM 输出（边沿对齐模式）
- 产生中断/DMA 请求的事件：更新事件、输入捕获、比较输出

13.4 功能描述

13.4.1 时钟

13.4.1.1 时钟选择 计数器的时钟由内部时钟（INT_CK）提

供。

13.4.1.2 时基单元

TIMx 的时基单元主要包括：计数器寄存器（TIMx_CNT）、预分频器寄存器（TIMx_PSC）和自动预装载寄存器（TIMx_ARR）。

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

计数器的时钟由预分频器提供，预分频器由预分频计数器和对应的寄存器组成，分频系数为 1-65536，可以随时写入，在下一次更新事件时生效。

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

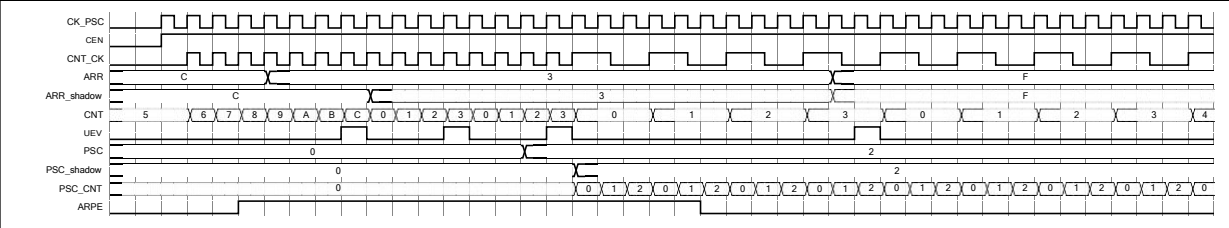


图 13-2 自动预装载

13.4.1.3 计数模式

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

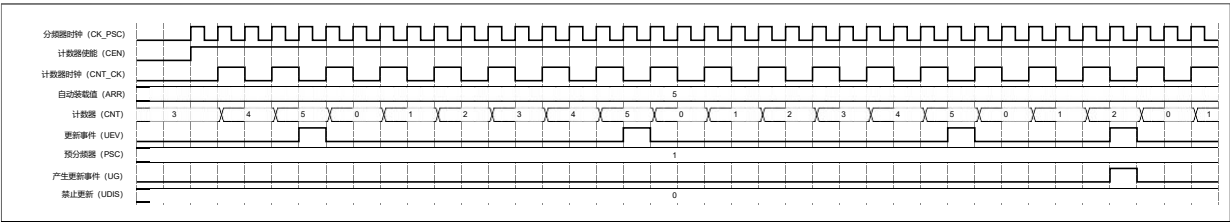


图 13-3 递增计数模式（UDIS=0）

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

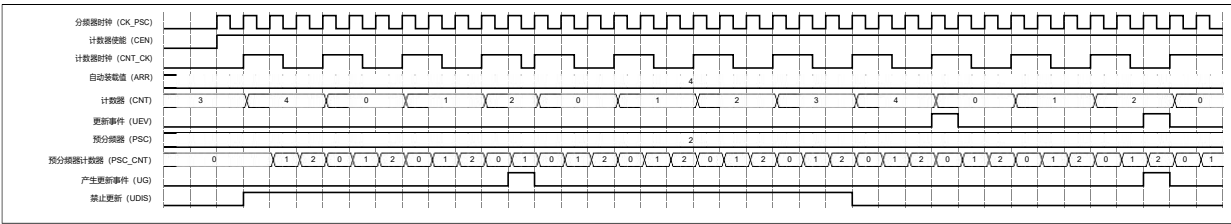


图 13-4 递增计数模式（UDIS=1 禁止产生更新事件）

注：发生更新事件时

- ARR 寄存器中的值被载入 ARR 影子寄存器中。
- 预分频器的预装载值生效。

13.4.2 输入捕获

13.4.2.1 输入捕获 输入捕获部分包括数字滤波器、边沿检测器、预分频器和捕获比较寄存器等，其结构

如下图所示：

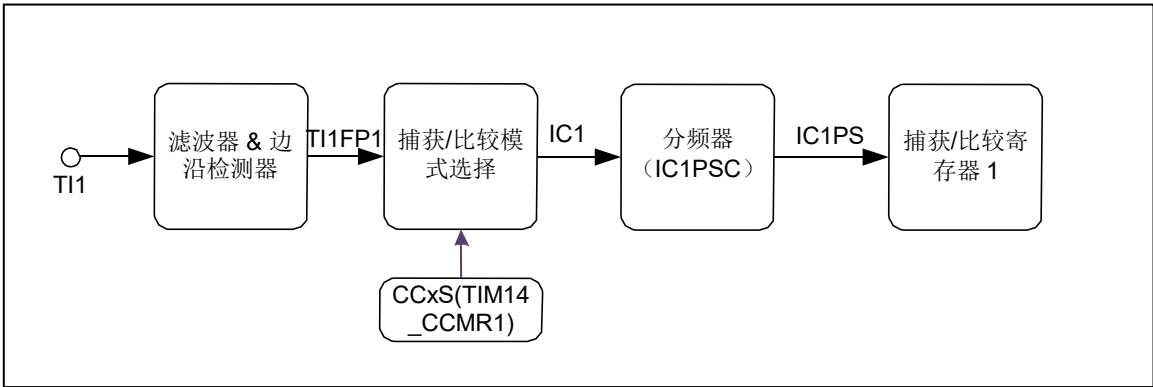


图 13-5 TIMx 输入捕获结构图

通过配置 TIMx_CCMR1 寄存器的 IC1F，可以设置数字滤波器的滤波宽度（滤波器的采样频率及数字滤波宽度如下表所示），当数字滤波器的输入信号宽度大于滤波宽度时，输入信号有效；数字滤波器对输入引脚 TI1 的输入信号采样后，产生一个滤波后的信号 TI1F，然后通过极性可选的边沿检测器，

产生一个有效信号 IC1，该信号经过预分频器产生一个信号 IC1PS，用于触发输入捕获事件。

表 13-1 数字滤波器宽度与 IC1F 的对应关系表

IC1F[3: 0]	采样频率和滤波宽度	IC1F[3: 0]	采样频率和滤波宽度
0000	无滤波器，以 f_{DTS} 采样	1000	采样频率 $f_{sampling}=f_{DTS}/8$ ， $N=6$
0001	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=2$	1001	采样频率 $f_{sampling}=f_{DTS}/8$ ， $N=8$
0010	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=4$	1010	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=5$
0011	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=8$	1011	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=6$
0100	采样频率 $f_{sampling}=f_{DTS}/2$ ， $N=6$	1100	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=8$
0101	采样频率 $f_{sampling}=f_{DTS}/2$ ， $N=8$	1101	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=5$
0110	采样频率 $f_{sampling}=f_{DTS}/4$ ， $N=6$	1110	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=6$
0111	采样频率 $f_{sampling}=f_{DTS}/4$ ， $N=8$	1111	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=8$

输入捕获模式下，当检测到信号 IC1 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断或 DMA 使能，发生捕获事件时，将产生相应的中断或 DMA 请求。发生捕获事件时，会将状态寄存器（TIMx_SR）中的捕获标志位 CC1IF 置 1，通过配置 CC1IF=0 或读取 TIMx_CCR1 中的数据，清除 CC1IF 标志位。当 CC1IF 未被清零时，发生输入捕获事件，重复捕获标志位 CC1OF 将会被置 1，通过配置 CC1OF=0，可以清除 CC1OF 标志位。

例如，通过采样 TI1 输入信号的有效沿，在 TI1 的上升沿来到时捕获当前计数器的值，锁存到 TIMx_CCR1 寄存器中，步骤如下：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 IC1F[3: 0]，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 TIMx_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 IC1PSC[1:0]，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。
6. 配置 TIMx_DIER 寄存器的 CC1IE=1，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 DMA，配置 TIMx_DIER 寄存器的 CC1DE=1，允许捕获/比较通道 1 的 DMA 请求。

注：

- 当通道配置为输入模式时，TIMx_CCR1 寄存器属性变为只读。
- 如果发生了两次以上连续捕获，但 CC1IF 标志未被清零，则重复捕获标志 CC1OF 被置 1。为了避免丢失重复捕获标志 CC1OF 置 1 之前可能产生的捕获信息，建议在读出重复捕获标志之前读取数据。
- 设置 TIMx_EGR 寄存器中相应的 CC1G 位，可以通过软件产生输入捕获中断或 DMA 请求。

13.4.3 比较输出

捕获比较通道的比较输出部分由比较器、输出控制电路和捕获/比较寄存器组成，其结构图如下图所示：

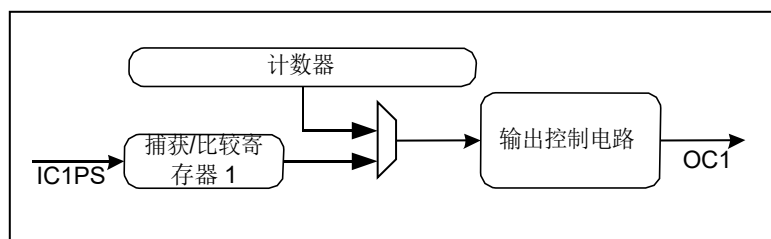


图 13-6 比较输出部分结构图

在比较输出模式下，捕获比较寄存器的内容被载入到影子寄存器中，然后影子寄存器的内容和计数器当前值进行比较。捕获/比较模块包括一个捕获/比较寄存器（预装载寄存器）和一个影子寄存器，读写过程仅操作捕获/比较寄存器。

13.4.3.1 强制输出

配置 `TIMx_CCMR1` 寄存器的 `CC1S = 00`，将通道 `CC1` 设置为输出模式，通过配置 `TIMx_CCMR1` 寄存器 `OC1M` 位，可以直接将比较输出信号直接强制为有效或无效状态，不依赖于比较结果。配置 `TIMx_CCMR1` 寄存器 `OC1M = 100`，强置比较输出信号为无效状态。此时 `OC1REF` 被强置为低电平。配置 `TIMx_CCMR1` 寄存器 `OC1M = 101`，强置比较输出信号为有效状态。此时 `OC1REF` 被强置为高电平（`OC1REF` 始终为高电平有效）。

注：强制输出模式下，在 `TIMx_CCR1` 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断和 DMA 请求，仍会产生对应的中断和 DMA 请求。

13.4.3.2 比较输出

比较输出模式下，当计数器与捕获比较寄存器值相同时，可以根据 `TIMx_CCMR1` 寄存器的 `OC1M` 位的配置用来输出不同的波形。例如，当计数器与捕获/比较寄存器的内容匹配时，

比较输出模式下的操作如下： 1. 在比较匹配时，`OC1M` 的值不同，输出通道 1 信号 `OC1` 的操作不同：

- ◆ `OC1M = 000`：OC1 信号保持它的电平
- ◆ `OC1M = 001`：OC1 信号被设置成有效电平
- ◆ `OC1M = 010`：OC1 信号被设置成无效电平
- ◆ `OC1M = 011`：OC1 信号进行翻转

2. 匹配时中断状态寄存器中的标志位置 1（`TIMx_SR` 寄存器中的 `CC1IF` 位）。

3. 当配置了 `TIMx_DIER` 寄存器中的 `CC1IE = 1`，匹配时则产生一个中断。

4. 当配置了 `TIMx_DIER` 寄存器中的 `CC1DE = 1`，匹配时则产生一个 DMA 请求（仅适用于有内置 DMA 的产品）。

比较输出模式也可以用来输出一个单脉冲（单脉冲输出模式）。例如，通

道 1 的比较输出模式的配置步骤如下：

- 1. 配置计数器的时钟（选择时钟源，配置预分频系数）。
- 2. 配置 TIMx_ARR 和 TIMx_CCR1 寄存器。
- 3. 配置 TIMx_DIER 寄存器的 CC1IE =1，使能捕获/比较 1 中断。
- 4. 配置输出模式：
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1PE = 0，禁止 TIMx_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输出/比较 1 输出使能，OC1 信号输出到对应的输出引脚。
- 5. 配置 TIMx_CR1 寄存器的 CEN =1，启动计数器。

当配置 TIMx_CCMR1 寄存器中 OC1PE=0，禁止 TIMx_CCR1 寄存器的预装载功能时，可以随时写入 TIMx_CCR1 寄存器，并且写入的值立即生效。当配置 TIMx_CCMR1 寄存器中 OC1PE=1，启用 TIMx_CCR1 寄存器的预装载功能时，读写仅对预装载寄存器进行操作，TIMx_CCR1 预装载寄存器的值在下次更新事件到来时生效。下图给出了一个例子。

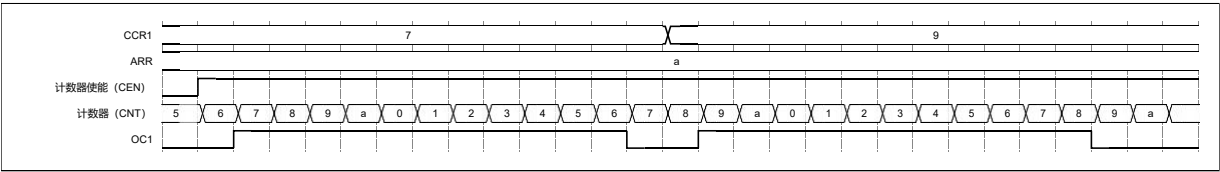


图 13-7 比较输出模式，OC1 信号在匹配时翻转

注：比较输出模式下，更新事件不会对输出结果产生影响。强制输出模式下，在 TIMx_CCR1 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断和 DMA 请求，仍会产生对应的中断和 DMA 请求。

13.4.3.3 PWM 输出

在 PWM 模式下，根据 TIMx_ARR 寄存器和 TIMx_CCR1 寄存器的值，产生一个频率、占空比可控的 PWM 波形。

配置与通道 1 对应的 TIMx_CCMR1 寄存器的 OC1M=110 或 OC1M=111，选择通道 1 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下，计数器和 CCR1 会一直进行比较，根据配置和比较结果，通道 1 输出不同的信号，TIMx 可以产生 1 个独立占空比的 PWM 输出信号。PWM 模式下可开启 TIMx_CCR1 的预装载功能和 TIMx_ARR 寄存器的预装载功能。写入 TIMx_CCR1 预装载寄存器和 TIMx_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 TIMx_EGR 的 UG=1，产生更新事件用于初始化所有的寄存器。

配置 TIMx_CCER 寄存器的 CC1P 位选择 OC1 的有效极性。配置 TIMx_CCER 寄存器的 CC1E 位控制 OC1 的输出使能。

13.4.3.3.1 PWM 边沿对齐模式——递增计数模式

配置 TIMx_CCMR1 寄存器的 CC1S=00，选择输出模式，OC1M=110，选择 PWM 模式 1，当 TIMx_CNT < TIMx_CCR1 时通道 1（OC1REF）为有效电平，否则为无效电平。如果 TIMx_CCR1 中的比较值大于自动重装载值（TIMx_ARR），则 OC1REF 保持为有效电平。如果比较值为 0，则 OC1REF 保持为无效电平。下图为 CCR1=4，ARR=a 时边沿对齐递增计数时 PWM 模式 1 的波形实例。

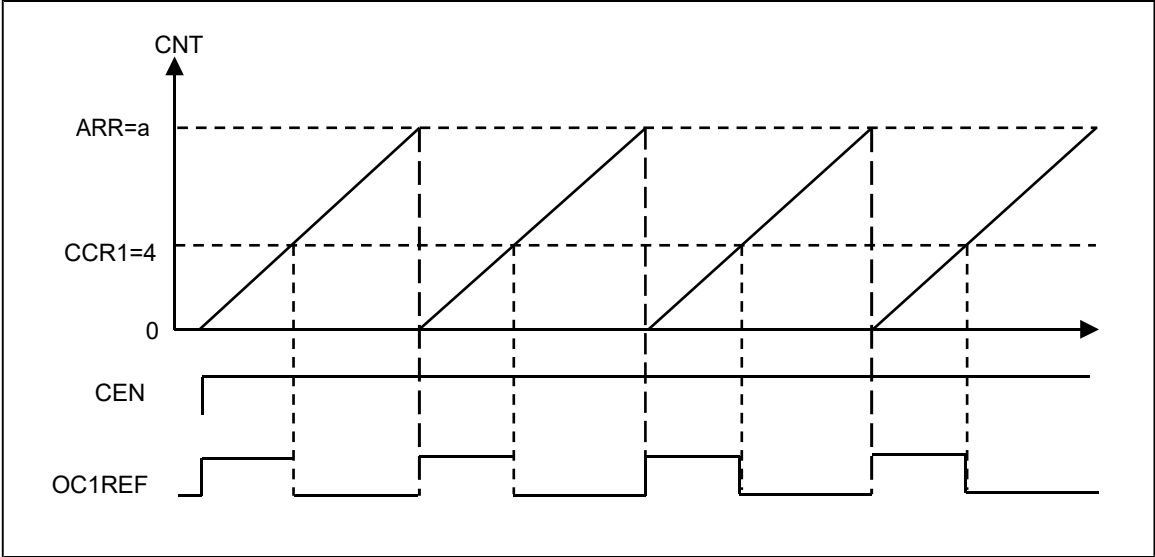


图 13-8 边沿对齐递增计数时 PWM 模式 1 的波形

13.4.3.4 单脉冲输出

单脉冲模式（OPM）下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 TIMx_CR1 寄存器的 OPM=1，选择单脉冲模式，配置 CEN=1 启动计数器，直到下个更新事件发生或配置 CEN=0 时，计数器停止计数。

产生脉冲的必要条件是比较值与计数器的初始值不同。所以在计数器启动之前的必要配置如下：

- 递增计数方式：计数器 $CNT < CCR1 \leq ARR$ 。

下图是单脉冲输出示例。

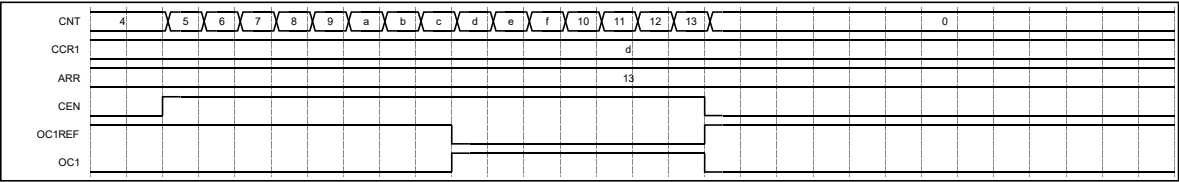


图 13-9 单脉冲模式

13.4.4 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIMx_STOP=1，TIMx 计数器停止计数。（详见调

试章节)

13.4.5 中断

TIMx 的中断包括：捕获/比较 1 中断和更新中断，当相应的中断使能位打开，发生相应的事件时，产生相应的中断。

表 13-2 中断事件一览表

中断事件	标志位	使能位
捕获/比较 1 中断	CC1IF	CC1IE
更新中断	UIF	UIE

13.4.6 DMA

TIMx 能够在发生捕获/比较事件或更新事件时生成 DMA 请求。

13.5 寄存器

表 13-3 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x0C	TIMx_DIER	DMA/中断使能寄存器（DMA 部分仅适用于有内置 DMA 的芯片）	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x44	TIMx_BDTR	刹车和死区寄存器	0x0000

13.5.1 TIMx_CR1 控制寄存器 1

偏移地址：0x00 复位值：

0x0000

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

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。

Bit	Field	Description
9: 8	CKD	时钟分频 (clock division) 定义定时器时钟 (INT_CK) 频率与数字滤波器 (TI1) 所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT_CK}$ 01: $t_{DTS} = 2 \times t_{INT_CK}$ 10: $t_{DTS} = 4 \times t_{INT_CK}$ 11: 保留, 不要使用这个配置
7	ARPE	自动重装载预装载使能 (Auto-reload preload enable) 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 4	Reserved	保留, 必须保持复位值。
3	OPM	单脉冲模式 (One pulse mode) 0: 禁止单脉冲模式, 在发生更新事件时, 计数器继续计数 1: 使能单脉冲模式, 在发生下一次更新事件或软件清除 CEN 位时, 计数器停止计数
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢 - 设置 UG 位 1: 只有计数器上溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC、CCR1) 保持值不变。如果设置了 EGR_UG 位为 1, 计数器和预分频器被初始化。
0	CEN	计数器使能 (Counter enable) 0: 禁止计数器 1: 使能计数器

13.5.2 TIMx_DIER DMA/中断使能寄存器

偏移地址: 0x0C 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1DE	UDE	Reserved						CC1IE	UIE
						rw	rw							rw	rw

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。
9	CC1DE	允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较 1 的 DMA 请求 1: 允许捕获/比较 1 的 DMA 请求 注: 仅适用于有内置 DMA 的产品。
8	UDE	允许更新 DMA 请求 (Update DMA request enable) 0: 禁止更新 DMA 请求 1: 允许更新 DMA 请求 注: 仅适用于有内置 DMA 的产品。
7: 2	Reserved	保留, 必须保持复位值。

Bit	Field	Description
1	CC1IE	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

13.5.3 TIMx_SR 状态寄存器

偏移地址: 0x10 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Reserved							CC1IF	UIF
						r_w0c								r_w0c	r_w0c

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。
9	CC1OF	捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当通道 1 被配置为输入捕获, CC1IF 已经为 1 后, 捕获事件再次发生时, 该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生 1: 重复捕获产生
8: 2	Reserved	保留, 必须保持复位值。
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1, 它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0 或读取 TIMx_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 更新事件包含计数器上溢、设置 UG=1。

13.5.4 TIMx_EGR 事件产生寄存器

偏移地址: 0x14 复位值:

0x0000

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

Bit	Field	Description
15: 2	Reserved	保留，必须保持复位值。
1	CC1G	产生通道 1 捕获/比较事件（Capture/Compare 1 generation） 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0：无动作 1：通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出：CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入：CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF = 1。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0

13.5.5 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址：0x18 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Res.	OC1M			OC1PE	Res.	CC1S	
								IC1F				IC1PSC		CC1S	
								rw	rw			rw		rw	rw

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CC1S 定义。该寄存器 CC1S 以外其它位的作用在输入模式和输出模式下不同。OC1x 描述了通道在输出模式下的功能，IC1x 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15: 7	Reserved	保留，必须保持复位值。

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode)</p> <p>该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。</p> <p>000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转, 当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为低。强制 OC1REF 为低电平。</p> <p>101: 强制为高。强制 OC1REF 为高电平。</p> <p>110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平。</p> <p>111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平。</p> <p>注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable)</p> <p>0: 禁止 TIMx_CCR1 寄存器的预装载功能, 写入 TIMx_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时生效。</p> <p>注: 若该位置 1, 在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1), 是否设定预装载寄存器无影响; 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。</p>
2	Reserved	保留, 必须保持复位值。
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入</p> <p>10: 保留</p> <p>11: 保留</p>

输入捕获模式:

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。

Bit	Field	Description
7: 4	IC1F	<p>通道 1 输入捕获滤波器 (Input capture 1 filter)</p> <p>数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, 以 f_{DTS} 采样</p> <p>0001: 采样频率 $f_{sampling}=f_{INT_CK}$, N=2</p> <p>0010: 采样频率 $f_{sampling}=f_{INT_CK}$, N=4</p> <p>0011: 采样频率 $f_{sampling}=f_{INT_CK}$, N=8</p> <p>0100: 采样频率 $f_{sampling}=f_{DTS}/2$, N=6</p> <p>0101: 采样频率 $f_{sampling}=f_{DTS}/2$, N=8</p> <p>0110: 采样频率 $f_{sampling}=f_{DTS}/4$, N=6</p> <p>0111: 采样频率 $f_{sampling}=f_{DTS}/4$, N=8</p> <p>1000: 采样频率 $f_{sampling}=f_{DTS}/8$, N=6</p> <p>1001: 采样频率 $f_{sampling}=f_{DTS}/8$, N=8</p> <p>1010: 采样频率 $f_{sampling}=f_{DTS}/16$, N=5</p> <p>1011: 采样频率 $f_{sampling}=f_{DTS}/16$, N=6</p> <p>1100: 采样频率 $f_{sampling}=f_{DTS}/16$, N=8</p> <p>1101: 采样频率 $f_{sampling}=f_{DTS}/32$, N=5</p> <p>1110: 采样频率 $f_{sampling}=f_{DTS}/32$, N=6</p> <p>1111: 采样频率 $f_{sampling}=f_{DTS}/32$, N=8</p>
3: 2	IC1PSC	<p>通道 1 输入/捕获预分频器 (Input capture 1 prescaler)</p> <p>该位定义了 IC1 的预分频系数。当 CC1E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次 捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入</p> <p>10: 保留</p> <p>11: 保留</p>

13.5.6 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

Bit	Field	Description
15: 4	Reserved	保留, 必须保持复位值。
3	CC1NP	<p>通道 1 输入捕获极性 (Capture 1 polarity)</p> <p>通道 1 配置为输出时, 此位无效。</p> <p>通道 1 配置为输入时, CC1P/CC1NP 配合使用定义了输入信号极性和电平, 详细参考 IC1 极性/电平选择表。</p>
2	Reserved	保留, 必须保持复位值。

Bit	Field	Description
1	CC1P	通道 1 输入/捕获输出极性（Capture/Compare 1 output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 IC1 极性/电平选择表。
0	CC1E	通道 1 输入/捕获输出使能（Capture/Compare 1 output enable） 通道 1 配置为输出时： 0：关闭。OC1 禁止输出 1：开启。OC1 信号输出到对应的输出引脚 CC1 通道配置为输入： 该位决定了输入捕获功能是否启用。 0：捕获禁止 1：捕获使能

输入模式下，IC1 的极性/电平选择如下表所示：

表 13-4 IC1 极性/电平选择表

CC1P	CC1NP	IC1 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/低电平有效
0	1	保留

13.5.7 TIMx_CNT 计数器

偏移地址：0x24 复位值：

0x0000

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

Bit	Field	Description
15: 0	CNT	计数器的值（Count value）

13.5.8 TIMx_PSC 预分频器

偏移地址：0x28 复位值：

0x0000

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

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
15: 0	PSC	预分频器的值 (Prescaler value) 计数器的时钟频率 (ck_cnt) = $f_{CK_PSC} / (PSC+1)$ 当发生更新事件时, PSC 的值装入当前预分频寄存器。

13.5.9 TIMx_ARR 自动预装载寄存器

偏移地址: 0x2C 复位值:

0x0000

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

Bit	Field	Description
15: 0	ARR	自动预装载值 (Auto-reload value) 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时, 计数器不工作。

13.5.10 TIMx_CCR1 捕获/比较寄存器 1

偏移地址: 0x34 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15: 0	CCR1	通道 1 捕获/比较的值 (Capture/Compare 1 value) 通道 1 配置为输入: 上一次捕获事件发生时捕获的计数器值存放于 CCR1 (此时 CCR1 寄存器为只读)。通道 1 配置为输出: 如果在 TIMx_CCMR1 寄存器 (OC1PE 位) 中未选择预装载功能, 写入的数值会立即传输至对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时, 此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。

13.5.11 TIMx_BDTR 刹车和死区寄存器

偏移地址: 0x44 复位值:

0x0000

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

MOE	Reserved
rw	

Bit	Field	Description
15	MOE	主输出使能 (Main output enable) 0: 禁止 OC1 输出或强制为空闲状态 1: 如果设置了相应的使能位 (TIMx_CCER 寄存器的 CC1E), 则开启 OC1 输出
14: 0	Reserved	保留, 必须保持复位值。

14 TIM16/17 基本定时器

16 位互补通道基本定时器包含 TIM16、TIM17，具有相同的功能，本产品搭载 TIM16 和 TIM17。本章节统一表述为 TIMx，部分示意图以 TIM16 为例展示。

14.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 16 位自动重载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。基本定时器具有多种用途，如输入功能（测量输入信号的脉冲宽度、频率等），输出功能（PWM 输出、死区时间可编程的互补输出、单脉冲模式输出等）。

14.2 功能框图

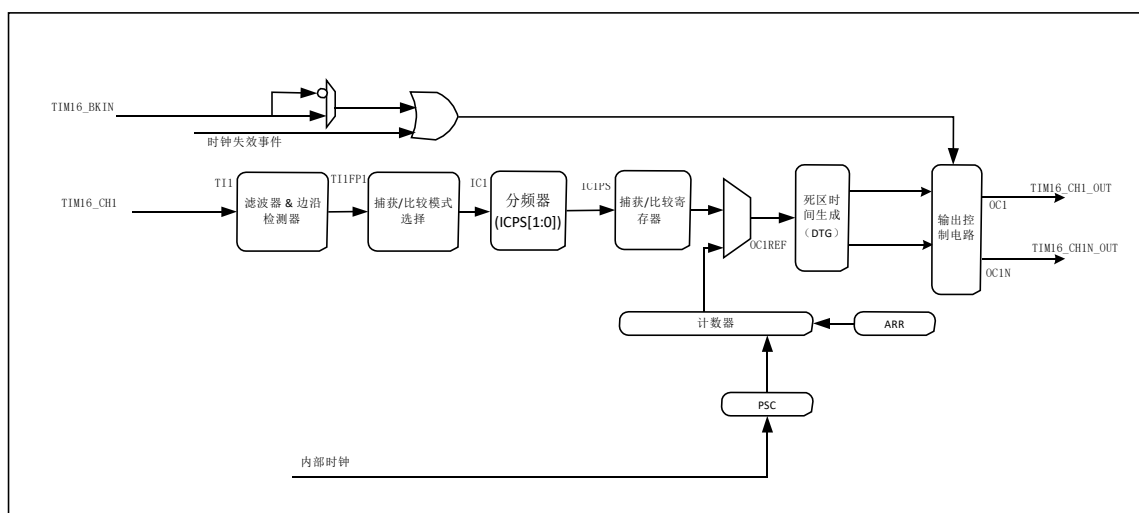


图 14-1 TIMx 结构图

上图为 TIMx 的结构框图，主要由输入单元、输出单元、时基单元、捕获/比较模块、刹车单元等结构组成。

14.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 16 位自动重载计数器（计数方向：递增）
- 输入捕获：输入信号的脉冲宽度、周期的测量

- 比较输出（控制输出波形或指示定时器已经计时结束）
- PWM 输出（死区时间可调；边沿对齐模式）
- 刹车输入可将计时器的输出信号置于安全状态（复位态或已知态，用户可选）
- 单脉冲输出
- 产生中断请求的事件：更新事件、COM 事件、输入捕获、比较输出或者刹车输入
- 产生 DMA 请求的事件：更新事件、输入捕获、比较输出

14.4 功能描述

14.4.1 时钟

14.4.1.1 时钟选择 计数器的时钟由内部时钟（INT_CK）提

供。

14.4.1.2 时基单元

TIMx 的时基单元主要包括：计数器寄存器（TIMx_CNT）、预分频器寄存器（TIMx_PSC）、自动预装载寄存器（TIMx_ARR）。

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

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

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

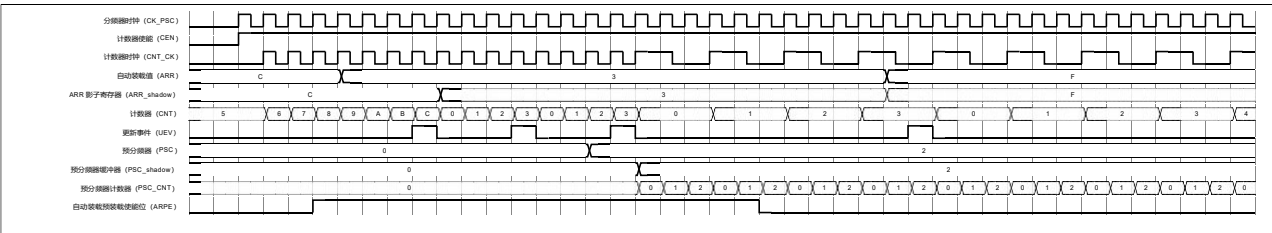


图 14-2 自动预装载

14.4.1.3 计数模式

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

通过配置 TIMx_CCMR1 寄存器的 IC1F，可以设置数字滤波器的滤波宽度（滤波器的采样频率及数字滤波宽度如下表所示），当数字滤波器的输入信号宽度大于滤波宽度时，输入信号有效；数字滤波器对输入引脚 TI1 的输入信号采样后，产生一个滤波后的信号 TI1F，然后通过极性可选的边沿检测器，产生一个有效信号 TI1FP1，这个信号可以作为从模式控制器的触发输入信号，同时该信号经过预分频器产生一个信号 IC1PS，用于触发输入捕获事件。

表 14-1 数字滤波器宽度与 IC1F 的对应关系表

IC1F[3: 0]	采样频率和滤波宽度	IC1F[3: 0]	采样频率和滤波宽度
0000	无滤波器，以 f_{DTS} 采样	1000	采样频率 $f_{sampling}=f_{DTS}/8$ ， $N=6$
0001	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=2$	1001	采样频率 $f_{sampling}=f_{DTS}/8$ ， $N=8$
0010	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=4$	1010	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=5$
0011	采样频率 $f_{sampling}=f_{INT_CK}$ ， $N=8$	1011	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=6$
0100	采样频率 $f_{sampling}=f_{DTS}/2$ ， $N=6$	1100	采样频率 $f_{sampling}=f_{DTS}/16$ ， $N=8$
0101	采样频率 $f_{sampling}=f_{DTS}/2$ ， $N=8$	1101	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=5$
0110	采样频率 $f_{sampling}=f_{DTS}/4$ ， $N=6$	1110	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=6$
0111	采样频率 $f_{sampling}=f_{DTS}/4$ ， $N=8$	1111	采样频率 $f_{sampling}=f_{DTS}/32$ ， $N=8$

输入捕获模式下，当检测到信号 IC1 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断或 DMA 使能，发生捕获事件时，将产生相应的中断或 DMA 请求。发生捕获事件时，会将状态寄存器（TIM1_SR）中的捕获标志位 CC1IF 置 1，通过配置 CC1IF=0 或读取 TIMx_CCR1 中的数据，清除 CC1IF 标志位。当 CC1IF 未被清零时，发生输入捕获事件，重复捕获标志位 CC1OF 将会被置 1，通过配置 CC1OF=0，可以清除 CC1OF 标志位。

例如，通过采样 TI1 输入信号的有效沿，在 TI1 的上升沿来到时捕获当前计数器的值，锁存到 TIMx_CCR1 寄存器中，步骤如下：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 IC1F[3: 0]，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 TIMx_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 IC1PSC[1:0]，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。
6. 配置 TIMx_DIER 寄存器的 CC1IE=1，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 DMA，配置 TIMx_DIER 寄存器的 CC1DE=1，允许捕获/比较通道 1 的 DMA 请求。

注：

- 当通道配置为输入模式时，TIMx_CCR1 寄存器属性变为只读。
- 如果发生了两次以上连续捕获，但 CC1IF 标志未被清零，则重复捕获标志 CC1OF 被置 1。为了避免丢失重复捕获标志 CC1OF 置 1 之前可能产生的捕获信息，建议在读出重复捕获标志之前读取数据。
- 设置 TIMx_EGR 寄存器中相应的 CC1G 位，可以通过软件产生输入捕获中断或 DMA 请求。

14.4.3 比较输出

捕获比较通道的比较输出部分由比较器、输出控制电路和捕获/比较寄存器组成，其结构图如下图所示：

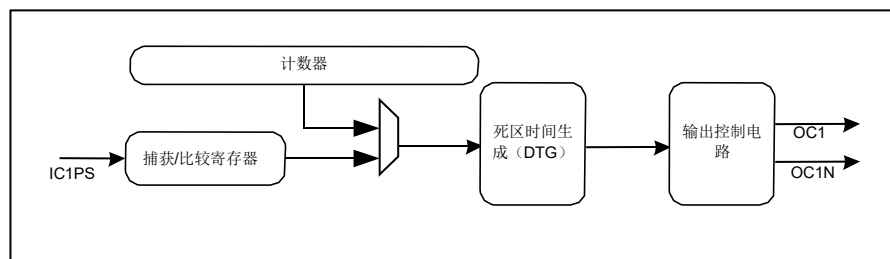


图 14-6 比较输出部分结构图

在比较输出模式下，捕获比较寄存器的内容被载入到影子寄存器中，然后影子寄存器的内容和计数器当前值进行比较。捕获/比较模块包括一个捕获/比较寄存器（预装载寄存器）和一个影子寄存器，读写过程仅操作捕获/比较寄存器。

14.4.3.1 强制输出

配置 `TIMx_CCMR1` 寄存器的 `CC1S = 00`，将通道 `CC1` 设置为输出模式，通过配置 `TIMx_CCMR1` 寄存器 `OC1M` 位，可以直接将比较输出信号直接强制为有效或无效状态，不依赖于比较结果。配置 `TIMx_CCMR1` 寄存器 `OC1M = 100`，强置比较输出信号为无效状态。此时 `OC1REF` 被强置为低电平。配置 `TIMx_CCMR1` 寄存器 `OC1M = 101`，强置比较输出信号为有效状态。此时 `OC1REF` 被强置为高电平（`OC1REF` 始终为高电平有效）。

注：强制输出模式下，在 `TIMx_CCR1` 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断或 DMA 请求，仍会产生对应的中断或 DMA 请求。

14.4.3.2 比较输出

比较输出模式下，当计数器与捕获比较寄存器值相同时，可以根据 `TIMx_CCMR1` 寄存器的 `OC1M` 位的配置用来输出不同的波形。例如，当计数器与捕获/比较寄存器的内容匹配时，

比较输出模式下的操作如下： 1. 在比较匹配时，`OC1M` 的值不同，输出通道 1 信号 `OC1` 的操作不同：

- ◆ `OC1M = 000`：OC1 信号保持它的电平
 - ◆ `OC1M = 001`：OC1 信号被设置成有效电平
 - ◆ `OC1M = 010`：OC1 信号被设置成无效电平
 - ◆ `OC1M = 011`：OC1 信号进行翻转
2. 匹配时中断状态寄存器中的标志位置 1（`TIMx_SR` 寄存器中的 `CC1IF` 位）。
 3. 当配置了 `TIMx_DIER` 寄存器中的 `CC1IE = 1`，匹配时则产生一个中断。

4. 当配置了 TIMx_DIER 寄存器中的 CC1DE =1，匹配时则产生一个 DMA 请求（仅适用于有内置 DMA 的产品）。

比较输出模式也可以用来输出一个单脉冲（单脉冲输出模式）。例如，通道 1 的比较输出模式的配置步骤如下：

- 1. 配置计数器的时钟（选择时钟源，配置预分频系数）。
- 2. 配置 TIMx_ARR 和 TIMx_CCR1 寄存器。
- 3. 配置 TIMx_DIER 寄存器的 CC1IE =1，使能捕获/比较 1 中断。
- 4. 配置输出模式：
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIMx_CCMR1 寄存器的 OC1PE = 0，禁止 TIMx_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIMx_CCER 寄存器的 CC1E = 1，开启输出/比较 1 输出使能，OC1 信号输出到对应的输出引脚。
- 5. 配置 TIMx_CR1 寄存器的 CEN =1，启动计数器。

当配置 TIMx_CCMR1 寄存器中 OC1PE=0，禁止 TIMx_CCR1 寄存器的预装载功能时，可以随时写入 TIMx_CCR1 寄存器，并且写入的值立即生效。当配置 TIMx_CCMR1 寄存器中 OC1PE=1，启用 TIMx_CCR1 寄存器的预装载功能时，读写仅对预装载寄存器进行操作，TIMx_CCR1 预装载寄存器的值在下次更新事件到来时生效。下图给出了一个例子。

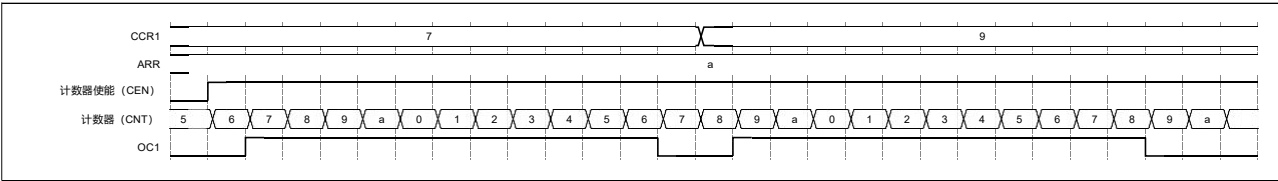


图 14-7 比较输出模式，OC1 信号在匹配时翻转

注：比较输出模式下，更新事件不会对输出结果产生影响。强制输出模式下，在 TIMx_CCR1 影子寄存器和计数器之间的比较输出仍在进行，比较结果的相应标志位也会被修改，如果开启了对应的中断或 DMA 请求，仍会产生对应的中断或 DMA 请求。

14.4.3.3 PWM 输出

在 PWM 模式下，根据 TIMx_ARR 寄存器和 TIMx_CCR1 寄存器的值，产生一个频率、占空比可控的 PWM 波形。

配置与通道 1 对应的 TIMx_CCMR1 寄存器的 OC1M=110 或 OC1M=111，选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下，计数器和 CCR1 会一直进行比较，根据配置和比较结果，通道 1 输出不同的信号，因此 TIMx 可以产生 1 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 TIMx_CCR1 的预装载功能和 TIMx_ARR 寄存器的预装载功能。写入 TIMx_CCR1 预装载寄存器

和 TIMx_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 TIMx_EGR 的 UG=1，产生更新事件用于初始化所有的寄存器。

配置 TIMx_CCER 寄存器的 CC1P 选择 OC1 的有效极性。配置 TIMx_CCER 寄存器的 CC1E、CC1NE 位和 TIMx_BDTR 寄存器的 MOE、OSSI、OSSR 位控制 OC1 的输出使能。

14.4.3.3.1 PWM 边沿对齐模式——递增计数模式

在递增计数模式配置的基础上，配置 TIMx_CCMR1 寄存器的 CC1S=00，选择输出模式，OC1M=110，选择 PWM 模式 1，当 TIMx_CNT < TIMx_CCR1 时通道 1（OC1REF）为有效电平，否则为无效电平。如果 TIMx_CCR1 中的比较值大于自动重装载值（TIMx_ARR），则 OC1REF 保持为有效电平。如果比较值为 0，则 OC1REF 保持为无效电平。下图为 CCR1=4，ARR=a 时边沿对齐递增计数时 PWM 模式 1 的波形实例。

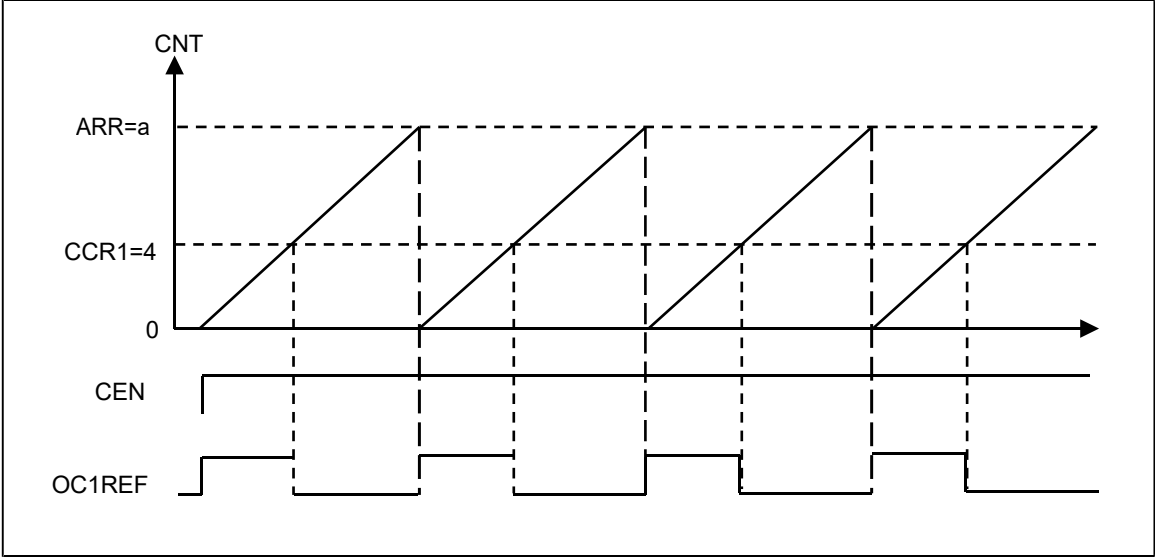


图 14-8 边沿对齐递增计数时 PWM 模式 1 的波形

14.4.3.4 互补输出和死区插入

OC1 和 OC1N 是一对互补输出通道，TIMx 的通道 1 能够输出 1 路可以管理瞬时关断和瞬时接通的互补信号，同时具有可调的死区时间。用户根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

TIMx_BDTR 寄存器 DTG[7: 0]位定义了插入互补输出之间的死区持续时间，具体计算方式如下表：

表 14-2 死区时间计算

DTG[7: 5]	DT
0xx	$DT = DTG[7: 0] \times T_{dtg} \quad (T_{dtg} = T_{DTS})$
10x	$DT = (64 + DTG[5: 0]) \times T_{dtg} \quad (T_{dtg} = 2 \times T_{DTS})$
110	$DT = (32 + DTG[4: 0]) \times T_{dtg} \quad (T_{dtg} = 8 \times T_{DTS})$
111	$DT = (32 + DTG[4: 0]) \times T_{dtg} \quad (T_{dtg} = 16 \times T_{DTS})$

例如，如果 $T_{DTS}=125ns$ ，可能的死区时间为：

- 若步长时间为 125ns，死区时间为 0 至 15875ns。
- 若步长时间为 250ns，死区时间为 16μs 至 31750ns。
- 若步长时间为 1μs，死区时间为 32μs 至 63μs。
- 若步长时间为 2μs，死区时间为 64μs 至 126μs。

当不存在刹车电路时，同时配置 CC1E=1 和 CC1NE=1，开启死区插入，否则还需要配置 MOE=1。

配置 TIMx_CCER 寄存器的 CC1P 和 CC1NP 位，可以为每一个输出独立地选择极性（主输出 OC1 或互补输出 OC1N）。

通过配置 TIMx_CCER 寄存器的 CC1E 和 CC1NE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OIS1、OIS1N、OSSI 和 OSSR 位的不同组合可以控制互补信号 OC1 和 OC1N 的输出。具体的组合控制配置见本章表 3、表 4、表 5 和表 6 的互补输出通道 OC1 和 OC1N 的控制位。

例：OC1 和 OC1N 都为高有效，PWM 模式下，发生匹配时，输出参考信号 OC1REF 信号翻转，输出信号 OC1 与参考信号相同，但是 OC1 信号的上升沿对于参考信号的上升沿有一个延时；互补输出信号 OC1N 与参考信号相反，OC1N 信号的上升沿对于参考信号的下降沿同样有一个延时。

注：死区时间不能大于或等于 OC1 或 OC1N 信号的占空比，否则 OC1 或 OC1N 信号一直为无效值。

下列几张图显示了死区发生器的输出信号和当前参考信号 OC1REF 之间的关系。

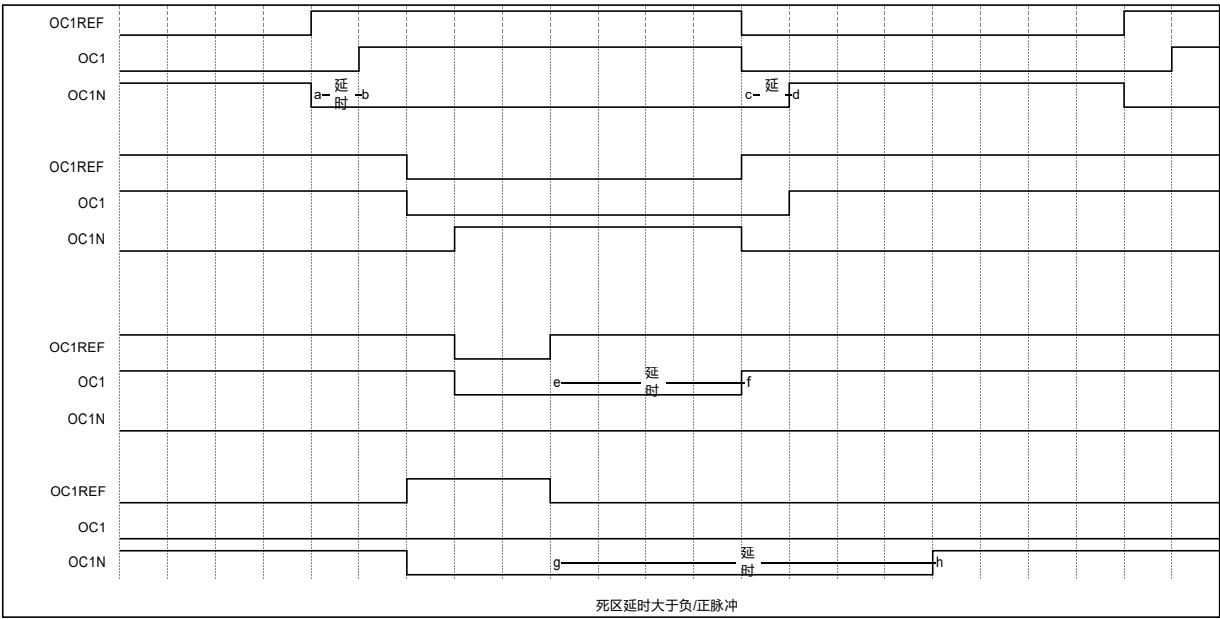


图 14-9 死区插入

14.4.3.5 刹车功能

TIMx 的刹车源有引脚输入和时钟失效事件两种类型。时钟失效事件由复位时钟控制器中的时钟安全系统产生。

使用刹车功能时，OC1 和 OC1N 输出信号电平被以下功能位组合控制：TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OIS1 和 OIS1N 位。发生刹车事件时，OC1 和

OC1N 输出不能同时有效。具体的带刹车功能的互补输出通道 OC1 和 OC1N 的输出状态如下列表格所示。

表 14-3 当 MOE=1, OSSI=0/1, OSSR=0 时:

CC1E	CC1NE	OC1	OC1N
0	0	OC1=0, OC1_EN=0	OC1N=0, OC1N_EN=0
0	1	OC1=0, OC1_EN=0	OC1N=OC1REF+Polarity, OC1N_EN=1
1	0	OC1=OC1REF+Polarity, OC1_EN=1	OC1N=0, OC1N_EN=0
1	1	OC1=OC1REF+Polarity+死区时间, OC1_EN=1	OC1N=OC1REF 反相+Polarity+死区时间, OC1N_EN=1

表 14-4 当 MOE=1, OSSI=0/1, OSSR=1 时:

CC1E	CC1NE	OC1	OC1N
0	0	OC1=0, OC1_EN=0	OC1N=0, OC1N_EN=0
0	1	OC1=CC1P, OC1_EN=1	OC1N=OC1REF+Polarity, OC1N_EN=1
1	0	OC1=OC1REF+Polarity, OC1_EN=1	OC1N=CC1NP, OC1N_EN=1
1	1	OC1=OC1REF+Polarity+死区时间, OC1_EN=1	OC1N=OC1REF 反相+Polarity+死区时间, OC1N_EN=1

表 14-5 当 MOE=0, OSSI=0, OSSR=0/1 时:

CC1E	CC1NE	OC1	OC1N
0	0	OC1_EN=0, OC1N_EN=0 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	
0	1	OC1_EN=0, OC1N_EN=0 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	
1	0	OC1_EN=0, OC1N_EN=0 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	
1	1	OC1_EN=0, OC1N_EN=0 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	

表 14-6 当 MOE=0, OSSI=1, OSSR=0/1 时:

CC1E	CC1NE	OC1	OC1N
0	0	OC1_EN=1, OC1N_EN=1 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	
0	1	OC1_EN=1, OC1N_EN=1 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	

CC1E	CC1NE	OC1	OC1N
1	0	OC1_EN=1, OC1N_EN=1 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	
1	1	OC1_EN=1, OC1N_EN=1 异步的: OC1=CC1P, OC1N=CC1NP 若时钟存在: 经过一个死区时间后, OC1=OIS1, OC1N=OIS1N OIS1 和 OIS1N 都不对应 OC1 和 OC1N 的有效电平	

注：当通道的输出和互补输出都关闭时，OIS1，OIS1N，CC1P 和 CC1NP 都必须配置为 0。系统复位后，MOE=0，刹车功能禁止，需要配置 TIMx_BKINF 寄存器的 BKIN_SEL，选择刹车源，支持选择多路刹车源，任意一路刹车信号有效都会触发刹车。配置 TIMx_BKINF 寄存器的 BKINFE，选择刹车信号滤波功能是否有效。配置 TIMx_BKINF 寄存器的 BKINF，选择刹车数字滤波的采样频率。更改刹车数字滤波采样频率前应先关闭刹车滤波功能。配置 TIMx_BDTR 寄存器的 BKE=1，使能刹车功能信号。配置 TIMx_BDTR 寄存器的 BKP 位选择刹车输入信号的极性。BKP 和 BKE 可以同时写入，且会在一个时钟周期后生效。

由于 MOE 被异步清除，因此在实际信号和同步控制位间插入了一个再同步电路，用于在同步信号和异步信号间产生延迟（当 MOE 状态为 0 时写入 1，写入后读取前需要插入一个空指令用于延时，否则无法保证正确读取）。

发生刹车事件时，MOE 被异步清零，此时根据 OSS1 的配置 OC1/OC1N 的输出将置于无效状态、空闲状态或复位状态；MOE=0 时，输出由 TIMx_CR2 寄存器的 OIS1 位决定，OSS1=0 时，定时器关闭输出使能，否则打开输出使能。当使用互补输出时，输出首先置于复位状态，然后死区重新生成，在死区之后输出电平由 OIS1 和 OIS1N 决定。

配置 TIMx_DIER 寄存器的 BIE=1，当发生刹车事件时，产生一个刹车中断；配置 TIMx_BDTR 寄存器的 AOE = 1，则在下一个更新事件到来时自动置位 MOE 位。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能（自动地或者通过软件）设置 MOE，并且状态标志 BIF 不能被清除。

刹车电路中实现了写保护以保证应用程序的安全，允许用户锁住死区长度，OC1/OC1N 极性和被禁止的状态，OC1M 配置，刹车使能和极性等参数。通过 TIMx_BDTR 寄存器中的 LOCK 位，可以选择 lock 等级（总共三级 lock）。Lock 在系统复位后只能修改一次。

下图显示响应刹车的输出实例：

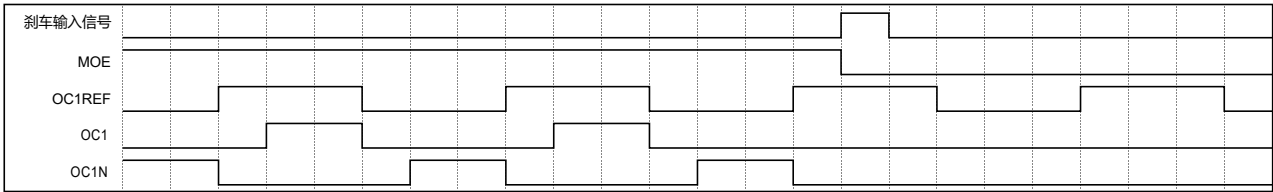


图 14-10 响应刹车的输出（OISx=0，OISxN=0）

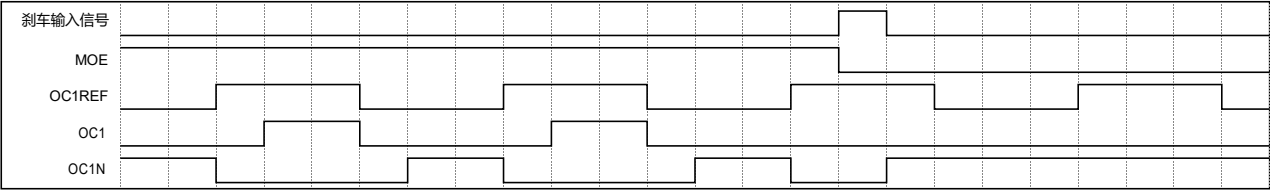


图 14-11 响应刹车的输出（OISx=0，OISxN=1）

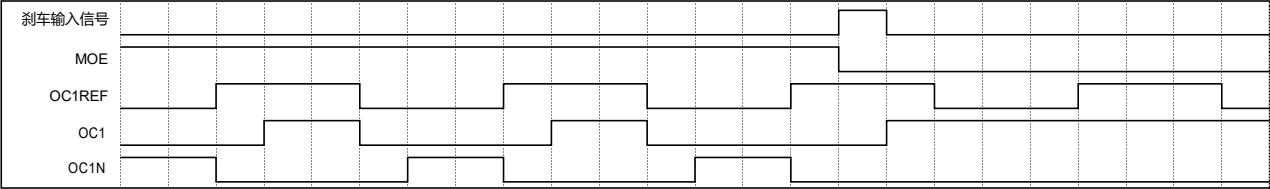


图 14-12 响应刹车的输出（OISx=1，OISxN=0）

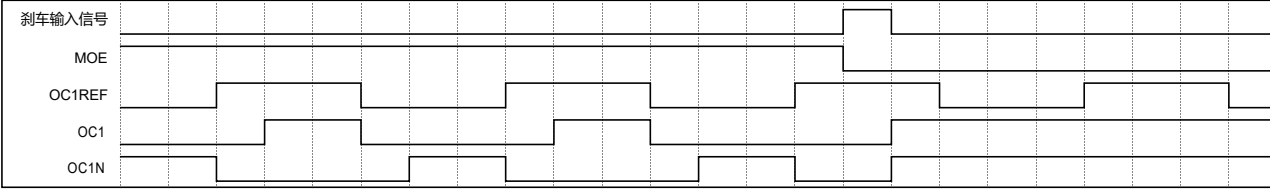


图 14-13 响应刹车的输出（OISx=1，OISxN=1）

14.4.3.6 单脉冲输出

单脉冲模式（OPM）下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 TIMx_CR1 寄存器的 OPM=1，选择单脉冲模式，配置 CEN=1 启动计数器，直到下个更新事件发生或配置 CEN=0 时，计数器停止计数。

产生脉冲的必要条件是比较值与计数器的初始值不同。所以在计数器启动之前的必要配置如下：

- 递增计数方式：计数器 $CNT < CCR1 \leq ARR$ 。

下图是单脉冲输出示例。

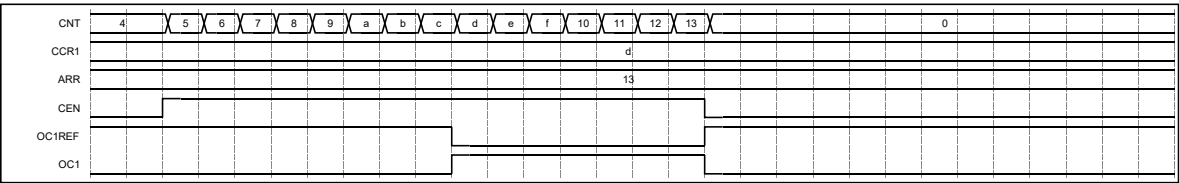


图 14-14 单脉冲模式

14.4.3.6.1 OC1 快速使能

OC1 快速使能，是单脉冲模式的一种特殊情况。在单脉冲模式下，通过设置 TIMx_CCMR 寄存器的 OC1FE=1，强制 OC1REF 直接响应激励而不是依赖计数器和比较值之间的比较结果，输出波形和

比较匹配时的波形一样。这样可以去除比较的时间，快速输出比较结果。OC1 快速输出使能只在 PWM 模式下生效。

14.4.4 定时器同步

不同的定时器在内部连接，可以实现定时器之间的级联或同步。 详细描述请参考 TIM1 相关章节。

14.4.5 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIMx_STOP=1，TIMx 计数器停止计数。（详见调 试 章节）

14.4.6 中断

TIMx 的中断包括：捕获/比较 1 中断、更新中断、COM 中断和刹车中断，当相应的中断使能位打 开，发生相应的事件时，产生相应的中断。

表 14-7 中断事件一览表

中断事件	标志位	使能位
捕获/比较 1 中断	CC1IF	CC1IE
更新中断	UIF	UIE
COM 中断	COMIF	COMIE
刹车中断	BIF	BIE

14.4.7 DMA

TIMx 能够在发生捕获/比较事件或更新事件时生成 DMA 请求。

14.5 寄存器

表 14-8 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x04	TIMx_CR2	控制寄存器 2	0x0000
0x0C	TIMx_DIER	DMA/中断使能寄存器（DMA 部 分仅适用于有内置 DMA 的芯 片）	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000

Offset	Acronym	Register Name	Reset
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x44	TIMx_BDTR	刹车和死区寄存器	0x0000
0x74	TIMx_BKINF	刹车滤波寄存器	0x0000 0000

14.5.1 TIMx_CR1 控制寄存器 1

偏移地址：0x00 复位值：

0x0000

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

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（TI1）所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT_CK}$ 01: $t_{DTS} = 2 \times t_{INT_CK}$ 10: $t_{DTS} = 4 \times t_{INT_CK}$ 11: 保留，不要使用这个配置
7	ARPE	自动重载预装载使能（Auto-reload preload enable） 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 4	Reserved	保留，必须保持复位值。
3	OPM	单脉冲模式（One pulse mode） 0: 禁止单脉冲模式，在发生更新事件时，计数器继续计数 1: 使能单脉冲模式，在发生下一次更新事件或软件清除 CEN 位时，计数器停止计数
2	URS	更新请求源（Update request source） 软件配置该位，选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求： - 计数器上溢 - 设置 UG 位 1: 只有计数器上溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新（Update disable） 该位用来允许或禁止更新事件的产生 0: 允许更新事件（UEV） 1: 禁止更新事件。不产生更新事件，影子寄存器（ARR、PSC、CCR1）保持值不变。如果设置了 EGR_UG 位为 1，计数器和预分频器被初始化。
0	CEN	计数器使能（Counter enable） 0: 禁止计数器 1: 使能计数器

14.5.2 TIMx_CR2 控制寄存器 2

偏移地址: 0x04 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						OIS1N	OIS1	Reserved				CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。
9	OIS1N	输出空闲状态 1 (OC1N 输出) (Output Idle state 1) 0: 当 MOE = 0 时, 死区后 OC1N = 0 1: 当 MOE = 0 时, 死区后 OC1N = 1 注: 已经设置了 LOCK (TIMx_BKR 寄存器) 级别 1、2 或 3 后, 该位不能被修改。
8	OIS1	输出空闲状态 1 (OC1 输出) (Output Idle state 1) 0: 当 MOE=0, 若 OC1N 有效, 则在死区时间后 OC1 = 0 1: 当 MOE=0, 若 OC1N 有效, 则在死区时间后 OC1 = 1 注: 已经设置了 LOCK (TIMx_BKR 寄存器) 级别 1、2 或 3 后, 该位不能被修改。
7: 4	Reserved	保留, 必须保持复位值。
3	CCDS	DMA 请求源选择 (Capture/compare DMA selection) 0: 当 CC1 发生捕获/比较事件时, 发送 CC1 的 DMA 请求 1: 发生更新事件时, 发送 CC1 的 DMA 请求 注: 仅适用于有内置 DMA 的产品
2	CCUS	捕获/比较控制更新源选择 (Capture/compare control update selection) 0: CCPC=1 时, 只能配置 COMG=1 更新。 1: CCPC=1 时, 可以通过配置 COMG=1 或检测到 TI1 的一个上升沿更新。 注: 此位只在通道为互补输出时有效。
1	Reserved	保留, 必须保持复位值。
0	CCPC	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CC1E, CC1NE 和 OC1M 位预装载禁用 1: CC1E, CC1NE 和 OC1M 位预装载使能 注: 此位只在通道为互补输出时有效。

14.5.3 TIMx_DIER DMA/中断使能寄存器

偏移地址: 0x0C 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1DE	UDE	BIE	Res.	COMIE	Reserved			CC1IE	UIE
						rw	rw	rw		rw				rw	

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。

Bit	Field	Description
9	CC1DE	允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较 1 的 DMA 请求 1: 允许捕获/比较 1 的 DMA 请求 注: 仅适用于有内置 DMA 的产品。
8	UDE	允许更新 DMA 请求 (Update DMA request enable) 0: 禁止更新 DMA 请求 1: 允许更新 DMA 请求 注: 仅适用于有内置 DMA 的产品。
7	BIE	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断 1: 允许刹车中断
6	Reserved	保留, 必须保持复位值。
5	COMIE	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断 1: 允许 COM 中断
4: 2	Reserved	保留, 必须保持复位值。
1	CC1IE	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

14.5.4 TIMx_SR 状态寄存器

偏移地址: 0x10 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Res.	BIF	Res.	COMIF	Reserved			CC1IF	UIF
						r_w0c		r_w0c		r_w0c				r_w0c	

Bit	Field	Description
15: 10	Reserved	保留, 必须保持复位值。
9	CC1OF	捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当通道 1 被配置为输入捕获, CC1IF 已经为 1 后, 捕获事件再次发生时, 该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生 1: 重复捕获产生
8	Reserved	保留, 必须保持复位值。
7	BIF	刹车中断标记 (Break interrupt flag) 当刹车输入有效, 由硬件对该位置 1。如果刹车输入无效, 则该位可由软件清 0 0: 无刹车事件产生 1: 刹车输入上检测到有效电平
6	Reserved	保留, 必须保持复位值。

Bit	Field	Description
5	COMIF	COM 中断标记 (COM interrupt flag) 当产生 COM 事件 (捕获/比较控制位 CC1E、CC1NE、OC1M 已被更新) 时该位由硬件置 1。它由软件清 0。 0: 无 COM 事件产生 1: COM 中断产生
4: 2	Reserved	保留, 必须保持复位值。
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1, 它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0 或读取 TIMx_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 更新事件包含计数器上溢、设置 UG=1。

14.5.5 TIMx_EGR 事件产生寄存器

偏移地址: 0x14 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	Res.	COMG	Reserved			CC1G	UG
								w		w				w	w

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	BG	产生刹车事件 (Break generation) 0: 无动作 1: 产生一个刹车事件, 此时 MOE=0, BIF=1, 若开启对应的中断, 则产生相应的中断请求, 由硬件清除。
6	Reserved	保留, 必须保持复位值。
5	COMG	捕获/比较事件, 产生控制更新 (Capture/Compare control update generation) 0: 无动作 1: 捕获/比较事件控制更新产生, 由硬件自动清 0, 当 CCPC=1, 允许更新 CC1E、CC1NE、OC1M 位。 注: 该位只对拥有互补输出的通道有效。
4: 2	Reserved	保留, 必须保持复位值。

Bit	Field	Description
1	CC1G	产生通道 1 捕获/比较事件 (Capture/Compare 1 generation) 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0：无动作 1：通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出：CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 若通道 CC1 配置为输入：CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF = 1。
0	UG	产生更新事件 (Update generation) 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0。

14.5.6 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址：0x18 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Res.								Res.	OC1M			OC1PE	OC1FE	CC1S				
								IC1F						IC1PSC		CC1S		
								rw	rw			rw			rw			

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CC1S 定义。该寄存器 CC1S 以外其它位的作用在输入模式和输出模式下不同。OC1x 描述了通道在输出模式下的功能，IC1x 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15: 7	Reserved	保留，必须保持复位值。

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode)</p> <p>该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。</p> <p>000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。</p> <p>100: 强制为低。强制 OC1REF 为低电平。</p> <p>101: 强制为高。强制 OC1REF 为高电平。</p> <p>110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平。</p> <p>111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平。</p> <p>注 1: 当 LOCK 级别设为 3 (TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S = 00 (该通道配置成输出) 时, 该位不能被修改。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable)</p> <p>0: 禁止 TIMx_CCR1 寄存器的预装载功能, 写入 TIMx_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时生效。</p> <p>注 1: 当 LOCK 级别设为 3 (TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S = 00 (该通道配置成输出) 时, 该位不能被修改。</p> <p>注 2: 若该位置 1, 在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1), 是否设定预装载寄存器无影响; 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。</p>
2	OC1FE	<p>通道 1 比较输出快速使能 (Output compare 1 fast enable)</p> <p>该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。</p> <p>0: 禁止通道 1 比较输出快速使能</p> <p>1: 开启通道 1 比较输出快速使能</p>
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入</p> <p>10: 保留</p> <p>11: 保留</p>

输入捕获模式:

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。

Bit	Field	Description
7: 4	IC1F	<p>通道 1 输入捕获滤波器 (Input capture 1 filter)</p> <p>数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, 以 f_{DTS} 采样</p> <p>0001: 采样频率 $f_{sampling}=f_{INT_CK}$, N=2</p> <p>0010: 采样频率 $f_{sampling}=f_{INT_CK}$, N=4</p> <p>0011: 采样频率 $f_{sampling}=f_{INT_CK}$, N=8</p> <p>0100: 采样频率 $f_{sampling}=f_{DTS}/2$, N=6</p> <p>0101: 采样频率 $f_{sampling}=f_{DTS}/2$, N=8</p> <p>0110: 采样频率 $f_{sampling}=f_{DTS}/4$, N=6</p> <p>0111: 采样频率 $f_{sampling}=f_{DTS}/4$, N=8</p> <p>1000: 采样频率 $f_{sampling}=f_{DTS}/8$, N=6</p> <p>1001: 采样频率 $f_{sampling}=f_{DTS}/8$, N=8</p> <p>1010: 采样频率 $f_{sampling}=f_{DTS}/16$, N=5</p> <p>1011: 采样频率 $f_{sampling}=f_{DTS}/16$, N=6</p> <p>1100: 采样频率 $f_{sampling}=f_{DTS}/16$, N=8</p> <p>1101: 采样频率 $f_{sampling}=f_{DTS}/32$, N=5</p> <p>1110: 采样频率 $f_{sampling}=f_{DTS}/32$, N=6</p> <p>1111: 采样频率 $f_{sampling}=f_{DTS}/32$, N=8</p>
3: 2	IC1PSC	<p>通道 1 输入/捕获预分频器 (Input capture 1 prescaler)</p> <p>该位定义了 IC1 的预分频系数。当 CC1E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次 捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC1S	<p>通道 1 捕获/比较选择 (Capture/Compare 1 selection)</p> <p>该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</p> <p>00: 通道 1 被配置为输出</p> <p>01: 通道 1 被配置为输入</p> <p>10: 保留</p> <p>11: 保留</p>

14.5.7 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

Bit	Field	Description
15: 4	Reserved	保留, 必须保持复位值。

Bit	Field	Description
3	CC1NP	通道 1 输入 / 捕获互补输出极性（Capture/Compare 1 complementary output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1N 高电平有效 1：OC1N 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 IC1 极性/电平选择表。 注：当 LOCK 级别（TIMx_BDTR 寄存器中的 LCCK 位）设为 3 或 2 且 CC1S = 00（通道配置为输出）时，该位不能被修改。
2	CC1NE	通道 1 输入 / 捕获互补输出使能（Capture/Compare 1 complementary output enable） 0：关闭通道 1 互补输出。OC1N 禁止输出。 1：开启通道 1 互补输出。 OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	通道 1 输入/捕获输出极性（Capture/Compare 1 output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 IC1 极性/电平选择表。 注：当 LOCK 级别（TIMx_BDTR 寄存器中的 LCCK 位）设为 3 或 2 时，该位不能被修改。
0	CC1E	通道 1 输入/捕获输出使能（Capture/Compare 1 output enable） 通道 1 配置为输出时： 0：关闭。OC1 禁止输出 1：开启。OC1 信号输出到对应的输出引脚 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了输入捕获功能是否启用。 0：捕获禁止 1：捕获使能

输入模式下，IC1 的极性/电平选择如下表所示：

表 14-9 IC1 极性/电平选择表

CC1P	CC1NP	IC1 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/低电平有效
0	1	保留

14.5.8 TIMx_CNT 计数器

偏移地址：0x24 复位值：

0x0000

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

Bit	Field	Description
15: 0	CNT	计数器的值 (Count value)

14.5.9 TIMx_PSC 预分频器

偏移地址: 0x28 复位值:

0x0000

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

Bit	Field	Description
15: 0	PSC	预分频器的值 (Prescaler value) 计数器的时钟频率 (ck_cnt) = $f_{CK_PSC} / (PSC + 1)$ 当发生更新事件时, PSC 的值装入当前预分频寄存器。

14.5.10 TIMx_ARR 自动预装载寄存器

偏移地址: 0x2C 复位值:

0x0000

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

Bit	Field	Description
15: 0	ARR	自动预装载值 (Auto-reload value) 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时, 计数器不工作。

14.5.11 TIMx_CCR1 捕获/比较寄存器 1

偏移地址: 0x34 复位值:

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
15: 0	CCR1	通道 1 捕获/比较的值 (Capture/Compare 1 value) 通道 1 配置为输入： 上一次捕获事件发生时捕获的计数器值存放于 CCR1 (此时 CCR1 寄存器为只读)。通道 1 配置为输出： 如果在 TIMx_CCMR1 寄存器 (OC1PE 位) 中未选择预装载功能，写入的数值会立即传输至对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

14.5.12 TIMx_BDTR 刹车和死区寄存器

偏移地址：0x44 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK		DTG							
rw	rw	rw	rw	rw	rw	rw		rw							

注：根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG 位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置，详见互补输出和死区插入章节。

Bit	Field	Description
15	MOE	主输出使能 (Main output enable) 当通道 1 配置为输出时，根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。当刹车输入有效时，该位被硬件异步清 0。 0: 禁止 OC1 和 OC1N 输出或强制为空闲状态 (输出使能信号关闭) 1: 如果设置了相应的使能位 (TIMx_CCER 寄存器的 CC1E、CC1NE 位)，则开启 OC1 和 OC1N 输出
14	AOE	自动输出使能 (AutoMatic output enable) 0: MOE 不能被硬件置 1 1: MOE 能被软件置 1 或刹车无效时在下一个更新事件被硬件自动置 1 注：当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。
13	BKP	刹车输入极性 (Break Polarity) 0: 刹车输入低电平有效 1: 刹车输入高电平有效 注：当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。
12	BKE	刹车功能使能 (Break enable) 0: 禁止刹车输入 1: 开启刹车输入 注 1: 当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。 注 2: 刹车输入包括引脚输入和 CSS 时钟失效两类事件，刹车使能前应先配置 TIMx_BKINF 寄存器中 BKIN_SEL 位，选择刹车源。

Bit	Field	Description
11	OSSR	运行模式下“关闭状态”选择（Off-state selection for Run mode） 该位仅适用于当 MOE = 1 且通道为互补输出。 0：当定时器不工作时，禁止 OC1/OC1N 输出 1：当定时器不工作时，如果 CC1E = 1 或 CC1NE = 1，首先开启 OC1/OC1N 并输出无效电平，然后置位 OC1/OC1N 输出使能信号。 注：当 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。
10	OSSI	空闲模式下“关闭状态”选择（Off-state selection for Idle mode） 该位仅适用于当 MOE = 0 且通道设为输出时。 0：当定时器不工作时，禁止 OC1/OC1N 输出。 1：当定时器不工作时，如果 CC1E = 1 或 CC1NE = 1，首先 OC1/OC1N 输出无效电平，然后置位 OC1/OC1N 输出使能信号。 注：当 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。
9: 8	LOCK	锁定设置（Lock configuration） 该位定义了寄存器的写保护功能。 00：写保护功能关闭，寄存器无写保护 01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OIS1/OIS1N 位 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位（通过 CC1S 位设为输出时，CC 极性位是 TIMx_CCER 寄存器的 CC1P/CC1NP 位）以及 OSSR/OSSI 位 11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位（当相关通道通过 CC1S 位设为输出时，CC 控制位是 TIMx_CCMR1 寄存器的 OC1M/OC1PE 位） 注：在系统复位后，LOCK 位只能写一次，当写入 TIMx_BDTR 寄存器后，LOCK 被写保护。
7: 0	DTG	死区发生器设置（Dead-time generator setup） 这些位定义了插入互补输出之间的死区持续时间。 注：当 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）设为 1、2 或 3 时，不能修改这些位。

14.5.13 TIMx_BKINF 刹车滤波寄存器

偏移地址：0x74 复位值：

0x0000 0000 TIM16:

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。

Bit	Field	Description
7: 6	IOBKIN_SEL	IO 刹车输入选择(IO break input sel) IOBKIN_SEL[1]: 0: TIM16_BKIN2 刹车输入无效 1: TIM16_BKIN2 刹车输入有效 IOBKIN_SEL[0]: 0: TIM16_BKIN1 刹车输入无效 1: TIM16_BKIN1 刹车输入有效
5	CSSBKIN_SEL	CSS 刹车输入选择(CSS break input sel) CSSBKIN_SEL: 0: CSS 刹车输入无效 1: CSS 刹车输入有效
4: 1	BKINF	BKINF: BKIN 数字滤波采样频率 (break input filter) 0000: 2 周期 0001: 4 周期 0010: 8 周期 0011: 16 周期 0100: 32 周期 0101: 64 周期 0110: 128 周期 0111: 256 周期 1000: 384 周期 1001: 512 周期 1010: 640 周期 1011: 768 周期 1100: 896 周期 1101: 1024 周期 1110: 1152 周期 1111: 1280 周期 注: 更改档位时应先关闭刹车滤波使能位。
0	BKINFE	BKINFE: BKIN 数字滤波使能 (break input filter enable) 1: 使能 BKIN 管脚数字滤波 0: 禁止 BKIN 管脚数字滤波 注: 滤波使能信号在刹车条件配置之后打开。

TIM17:

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

Bit	Field	Description
31: 9	Reserved	保留, 必须保持复位值。
8: 6	IOBKIN_SEL	IO 刹车输入选择(IO break input sel) IOBKIN_SEL[2]: 0: TIM17_BKIN3 刹车输入无效 1: TIM17_BKIN3 刹车输入有效 IOBKIN_SEL[1]: 0: TIM17_BKIN2 刹车输入无效 1: TIM17_BKIN2 刹车输入有效 IOBKIN_SEL[0]: 0: TIM17_BKIN1 刹车输入无效 1: TIM17_BKIN1 刹车输入有效

Bit	Field	Description
5	CSSBKIN_SEL	CSS 刹车输入选择(CSS break input sel) CSSBKIN_SEL: 0: CSS 刹车输入无效 1: CSS 刹车输入有效
4: 1	BKINF	BKINF: BKIN 数字滤波采样频率 (break input filter) 0000: 2 周期 0001: 4 周期 0010: 8 周期 0011: 16 周期 0100: 32 周期 0101: 64 周期 0110: 128 周期 0111: 256 周期 1000: 384 周期 1001: 512 周期 1010: 640 周期 1011: 768 周期 1100: 896 周期 1101: 1024 周期 1110: 1152 周期 1111: 1280 周期 注: 更改档位时应先关闭刹车滤波使能位。
0	BKINFE	BKINFE: BKIN 数字滤波使能 (break input filter enable) 1: 使能 BKIN 管脚数字滤波 0: 禁止 BKIN 管脚数字滤波 注: 滤波使能信号在刹车条件配置之后打开。

15 IWDG 独立看门狗

15.1 IWDG 简介

内置独立看门狗，提供了更高的安全性、时间的精确性和使用的灵活性。可用来检测 and 解决由软件错误引起的故障，当计数器达到给定的超时值时，产生系统复位。独立看门狗 (IWDG) 由专门的低速时钟 (LSI) 驱动，即使主时钟发生故障它也仍然有效。

IWDG 最适合应用于那些需要看门狗作为一个正在主程序外，能够完全独立工作，并且对时间精度要求低的场合。

15.2 IWDG 主要性能

- 自由运行的递减计数器
- 时钟由独立的振荡器提供 (可在停机模式下工作)
- 看门狗被激活后，则在计数器计数至 0x0000 时产生复位。

15.3 IWDG 功能描述

下图为独立看门狗模块的功能框图。

在键寄存器 (IWDG_KR) 中写入 0xCCCC。开始启动独立看门狗；此时计数器开始从其复位值 0xFFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号 (IWGD_RESET)。 无论何时，只要在键寄存器 IWDG_KR 中写入 0xAAAA，IWDG_RLR 中的值就会被重新加载到计数器，从而避免产生看门狗复位。

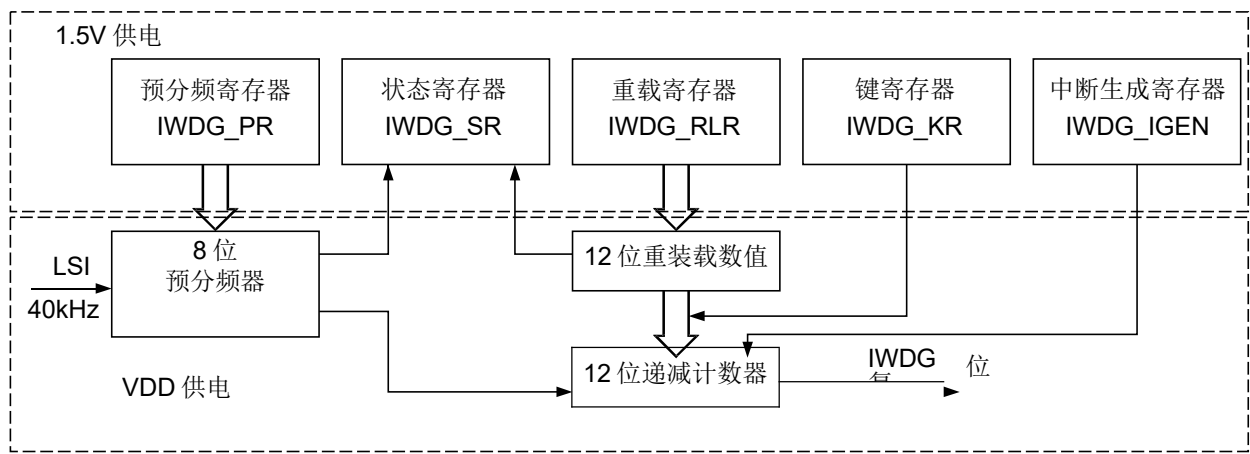


图 15-1 独立看门狗框图

注：看门狗功能处于 VDD 供电区，即在停机时仍能正常工作。

表 15-1 IWDG 超时时间（以 LSI 时钟频率 40kHz 为例）

预分频系数	PR[2:0] 位	最短时间 (ms) RL[11:0]=0x000	最长时间 (ms)
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 或 7)	6.4	26214.4

注：这些时间是按照 40KHz 时钟给出。实际上，MCU 内部的振荡器频率会在 30KHz 到 60KHz 之间变化。此外，即使振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与振荡器时钟之间的相位差，因此总会有一个完整的振荡器周期是不确定的。

15.3.1 硬件看门狗

如果用户在选择字节中 (请参考“嵌入式闪存”章节) 启动了‘硬件看门狗’功能，在系统上电复位后，看门狗会自动开始运行；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，则系统会产生复位。

15.3.2 寄存器访问保护

IWDG_PR、IWDG_RLR 和 IWDG_IGEN 寄存器具有写保护功能。要修改这三个寄存器的值，必须先向 IWDG_KR 寄存器中写入 0x5555。以不同的值写入这个寄存器将会打乱操作顺序，寄存器将重新被保护。重装载操作 (即写入 0xAAAA) 也会启动写保护功能。状态寄存器指示预分频值和递减计数器是否正在被更新。

15.3.3 调试模式

当微控制器进入调试模式时 (CPU 核心停止)，根据调试模块中的 DBG_IWDG_STOP 配置位的状态，IWDG 的计数器能够继续工作或停止。详见调试模块的章节。

15.4 寄存器

15.4.1 寄存器总览

表 15-2 IWDG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	IWDG_KR	键寄存器	0x00000000
0x04	IWDG_PR	预分频寄存器	0x00000000
0x08	IWDG_RLR	重装载寄存器	0x00000FFF
0x0C	IWDG_SR	状态寄存器	0x00000000
0x10	IWDG_CR	控制寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x14	IWDG_IGEN	中断生成寄存器	0x00000FFF
0x18	IWDG_CNT	计数寄存器	0x00000000

15.4.2 IWDG_KR 键寄存器

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

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	KEY	键值 (只写寄存器，读出值为 0x0000)(Key value) 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。 写入 0x5555 表示允许访问 IWDG_PR、IWDG_RLR、IWDG_IGEN 和 IWDG_CR[IRQ_SEL] 寄存器。 写入 0xCCCC，启动看门狗工作。

15.4.3 IWDG_PR 预分频寄存器

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
Reserved32															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved32													PR		
													rw		

Bit	Field	Description
31:3	Reserved	始终读为 0

Bit	Field	Description
2: 0	PR	<p>预分频因子 (Prescaler divider)</p> <p>这些位具有写保护设置。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子，IWDG_SR 寄存器的 PVU 位必须为 0。</p> <p>000: 预分频因子 = 4 100: 预分频因子 = 64 001: 预分频因子 = 8 101: 预分频因子 = 128 010: 预分频因子 = 16 110: 预分频因子 = 256 011: 预分频因子 = 32 111: 预分频因子 = 256 注意：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有对 IWDG_SR 寄存器的 PVU 位为 0 时，读出的值才有效。</p>

15.4.4 IWDG_RLR 重装载寄存器

偏移地址：0x08 复位值：

0x0000 0FFF

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

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11: 0	RL	<p>看门狗计数器重装载值 (Watchdog counter reload value)</p> <p>这些位具有写保护。用于定义看门狗计数器的重装载值，每当向 IWDG_KR 寄存器写入 0xAAAA 时，重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过次重装载值和时钟预分频值来计算。注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 RUV 位为 0 时，读出的值才有效。</p>

15.4.5 IWDG_SR 状态寄存器

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.												UPDAT E	IVU	RVU	PVU
												r	r	r	r

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值

Bit	Field	Description
3	UPDATE	看门狗重装值更新标记 此位由硬件置'1'用来指示 IWDG_KR 寄存器中写入 0xAAAA。当看门狗的重装值写入计数器后，计数器被更新，此位由硬件清'0'。
2	IVU	看门狗中断生成值更新 (Watchdog Interrupt Generate value update) 此位由硬件置'1'用来指示中断生成值的更新正在进行中。当在 VDD 域中的中断生成值更新结束后，此位由硬件清'0'(最多需要 5 个 40KHz 的振荡器周期) 中断生成值只有在 IVU 位被清'0'后才可更新。
1	RVU	看门狗计数器重装值更新 (Watchdog counter reload value update) 此位由硬件置'1'用来指示重装值的更新正在进行中。当在 VDD 域中的重装值更新结束后，此位由硬件清'0'(最多需要 5 个 40KHz 的振荡器周期) 重装值只有在 RVU 位被清'0'后才可更新。
0	PVU	看门狗预分频更新 (Watchdog prescaler value update) 此位由硬件置'1'用来指示预分频值的更新正在进行中。当在 VDD 域中的预分频值更新结束后，此位由硬件清'0'(最多需要 5 个 40KHz 的振荡器周期) 预分频值只有在 RVU 位被清'0'后才可更新。

注：如果在应用程序中使用多个重装值、预分频值或中断生成值，则必须在 RVU 位被清除后才能重新改变预装载值，在 PVU 位被清除后才能重新改变预分频值，必须在 IVU 位被清除后才能重新改变中断生成值。然而，在预分频或重装值更新后，不必等待 RVU 或 PVU 复位，可以继续执行下面的代码。

15.4.6 IWDG_CR 控制寄存器

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.														IRQ_CLR	IRQ_SEL
														rw	rw

Bit	Field	Description
31:2	Reserved	保留，必须保持复位值
1	IRQ_CLR	IWDG 中断清除 1：写 1 清除中断 0：无操作
0	IRQ_SEL	IWDG 溢出操作选择 1：溢出后产生中断 0：溢出后产生复位

15.4.7 IWDG_IGEN 中断生成寄存器

偏移地址：0x14 复位值：

0x0000 0FFF

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

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

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11: 0	IGEN	<p>IWDG 中断生成值 (Watchdog Interrupt Generate value)</p> <p>这些位具有写保护。用于定义看门狗计数器产生中断的阈值，每当计数器值递减等于阈值时，会产生 中断。</p> <p>只有当 IWDG_SR 寄存器中的 IVU 位为 0 时，才能对此寄存器进行修改。注：对此寄存器进行读操作，将从 VDD 电压域返回中断生成值。如果写操作正在进行，返回值可能无效。只有当 IWDG_SR 寄存器的 IVU 位为 0 时，返回值才能保证有效。</p>

15.4.8 IWDG_CNT 计数寄存器

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

Bit	Field	Description
31: 19	Reserved	保留，始终读为 0。
18: 8	IWDG_CNT	IWDG 计数器 counter 的当前值
7: 0	IWDG_PS	IWDG 时钟分频计数器的当前值

16 RTC 实时时钟器

16.1 RTC 简介

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

RTC 模块处于备份域中，即在当芯片被系统复位后，RTC 设置保持不变，正常计时。为避免对备份域的意外写操作，系统复位后禁止访问 RTC 和备份寄存器。如需开启 RTC 和备份寄存器的访问权限，需要配置 RCC_BDCR 寄存器的 DBP 位为 1。

16.2 主要特征

- 0 到 2^{20} 的可编程控制预分频系数；
- 32 位的可编程计数器；
- 包含 2 个独立的时钟：RTC 时钟以及用于 APB1 接口的 PCLK1，其中 RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一以上；
- RTC 的时钟源有 3 种：
 - ◆ HSE 时钟的 128 分频；
 - ◆ 外部低速 LSE 振荡器时钟；
 - ◆ 内部低速 LSI 振荡器时钟；
- 3 个可屏蔽中断：
 - ◆ 可编程控制产生的闹钟中断；
 - ◆ 可编程控制产生的秒中断（最长时间为 1 秒）；
 - ◆ 计数溢出中断，内部可编程计数器溢出并返回为 0 的状态；
- 2 个独立的复位系统：
 - ◆ 预分频器、闹钟、毫秒闹钟、计数器和分频器（RTC 核心部分）只能通过上电复位或备份域软件复位；
 - ◆ APB1 接口部分通过系统复位；

16.3 功能描述

16.3.1 功能框图

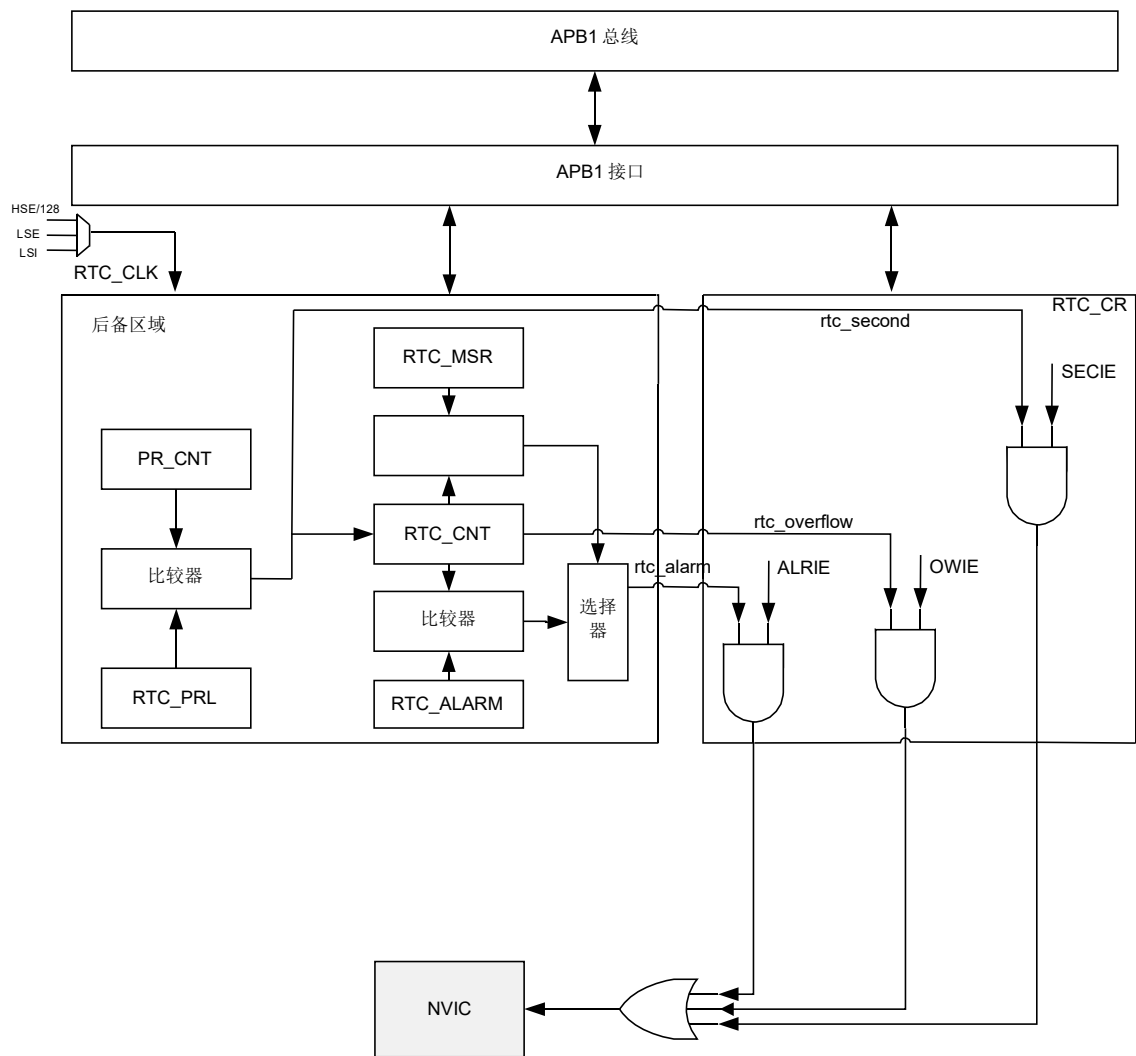


图 16-1 RTC 功能框图

16.3.2 功能概述

RTC 主要包含两个部分，APB1 接口部分与可编程计数部分。

APB1 接口部分用于连接 APB1 总线，该部分包含一组 16 位寄存器。可编程预分频器，最大可编程时钟周期 SC_CLK 为 1 秒。如果在 RTC_CR 控制寄存器中设置相应的中断使能位，则在每个 SC_CLK 时钟周期中会产生一个中断（秒中断）。

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

16.3.3 模块复位

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

16.3.4 寄存器读取

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

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

可能造成这种情况的几种情形如下：

- 从停机模式唤醒系统并立即进行读操作；
- 发生系统复位或电源复位并立即进行读操作；

当发生以上情况时：即有复位、无时钟，导致 APB1 接口被禁止，但 RTC 备份域电路仍保持运行状态。因此，如果 RTC 的 APB1 接口处于被禁止的状态，读取 RTC 寄存器时，则软件必须等待硬件置位 RTC_CRL 寄存器中的同步标志位（RSF），读操作才能继续。

注：睡眠或低功耗睡眠模式不影响 APB1 接口。

16.3.5 寄存器配置

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

具体配置过程如下：

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

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

16.3.6 标志位产生

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

当计数器到达 0x00000000 之前的最后一个 RTC 时钟周期时，RTC 溢出标志（OWF）置 1。

RTC_CNT 开始计数后，在计数到闹钟寄存器值加 1（RTC_ALR+1）之前的时钟周期中，配置 RTC_ALARM 与闹钟标志 ALRF。

- 当满足 RTC_CNT 计数值到达 RTC_ALR+1 值前的一个 RTC 时钟周期时，RTC_ALARM 标志位置 1；
- 当满足 RTC_CNT 计数值到达 RTC_MSR+1 值前的一个 RTC 时钟周期时，RTC_ALARM 标志位置 1；
- 当满足 RTC_CNT 计数值等于 RTC_ALR+1 值且 RTC_DIV 计数到 MSR+1 时，RTC_ALARM 标志位置 1；

写 RTC 闹钟只能通过如下的两种方式与秒标志同步：

- 只有在进入中断之后，RTC_CNT、RTC_ALR、RTC_MSR 这几个寄存器才能被更新。
- 只有在 RTC_CRL.SECF 位置 1 时，RTC_CNT、RTC_ALR、RTC_MSR 这几个寄存器才能被更新。

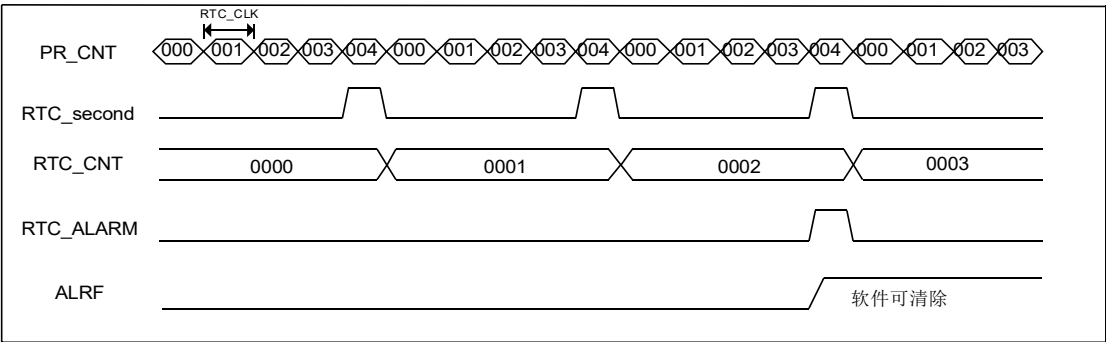


图 16-2 RTC 秒和闹钟波形图示例，RTC_PRL = 0004，RTC_ALR = 002

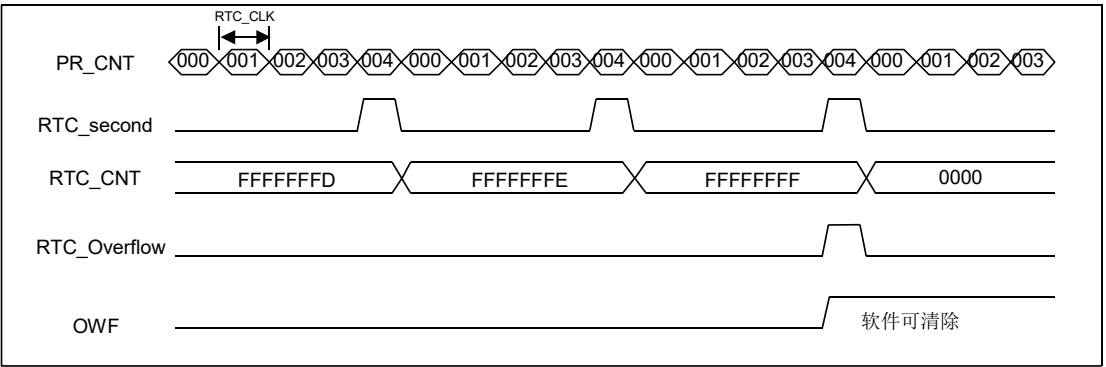


图 16-3 RTC 溢出波形图示例，RTC_PRL = 0004

16.3.7 RTC 闹钟描述

RTC 闹钟分为两部分：

毫秒计时闹钟：此时需 RTC_ALR 寄存器为 0，通过配置 RTC_MSR 来定时毫秒时间。当定时时间到达，闹钟标志位置位。

秒计时闹钟：此时需 RTC_MSR 寄存器为 0，通过配置 RTC_ALR 来定时秒钟时间。当定时时间到达，闹钟标志位置位。

可同时使用秒和毫秒闹钟用于非整秒定时。

RTC 闹钟循环：使用 RTC_CRL 的 ALPEN 位来配置闹钟单次/循环发生。

16.3.8 RTC 外部中断事件输出

配置 RTC_CRL 寄存器中的 ALRF 位为 1，同时使能 EXTI 17，则允许产生 RTC 闹钟中断；配置 EXTI 17 为事件模式，会产生一个 RTC 事件脉冲，但不会进入中断。

配置 RTC_CRH 寄存器中的 ALRIE 与 RTC_CRL 寄存器中的 ALRF 位为 1，允许产生 RTC 全局中断。同时使能 EXTI 17，则允许产生 RTC 全局中断和 RTC 闹钟中断。

16.4 寄存器描述

16.4.1 寄存器总览

表 16-1 RTC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RTC_CRH	RTC 控制寄存器高位	0x0000
0x04	RTC_CRL	RTC 控制寄存器低位	0x0020
0x08	RTC_PRLH	RTC 预分频装载寄存器高位	0x0000
0x0C	RTC_PRLH	RTC 预分频装载寄存器低位	0x8000
0x10	RTC_DIVH	RTC 预分频器分频因子寄存器高位	0x0000
0x14	RTC_DIVL	RTC 预分频器分频因子寄存器低位	0x0000
0x18	RTC_CNTH	RTC 计数器寄存器高位	0x0000
0x1C	RTC_CNTL	RTC 计数器寄存器低位	0x0000
0x20	RTC_ALRH	RTC 闹钟寄存器高位	0xFFFF
0x24	RTC_ALRL	RTC 闹钟寄存器低位	0xFFFF
0x28	RTC_MSRH	RTC 毫秒寄存器高位	0x0000
0x2C	RTC_MSRL	RTC 毫秒寄存器低位	0x0000
0x3C	RTC_LSE_CFG	RTC LSE 配置寄存器	0x0198

16.4.2 控制寄存器高位 (RTC_CRH)

偏移地址：0x0

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													OWIE	ALRIE	SECIE
													rw	rw	rw

Bit	Field	Description
15:3	Reserved	保留，必须保持复位值
2	OWIE	溢出中断使能位（Overflow Interrupt Enable） 0：溢出中断禁止 1：溢出中断使能
1	ALRIE	闹钟中断使能位（Alarm Interrupt Enable） 0：闹钟中断禁止 1：闹钟中断使能
0	SECIE	秒中断使能位（Second Interrupt Enable） 0：秒中断禁止 1：秒中断使能

16.4.3 控制寄存器低位(RTC_CRL)

偏移地址：0x04 复位值：

0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ALPEN	RTOFF	CNF	RSF	OWF	ALRF	SECF
									rw	r	rw	w0c	w0c	w0c	w0c

Bit	Field	Description
15:7	Reserved	保留，必须保持复位值
6	ALPEN	RTC 闹钟循环使能（RTC Alarm Loop Enable） 0：单次发生闹钟事件 1：循环发生闹钟事件
5	RTOFF	RTC 操作状态（RTC Operation OFF） 指示对寄存器最后一次写操作是否完成。若此位为'0'，则表示无法对任何的 RTC 寄存器进行读写操作。此位只读。 0：对 RTC 寄存器的写操作仍在进行 1：完成对 RTC 寄存器的写操作
4	CNF	配置标志（Configuration Flag） 只能通过软件置'1'进入配置模式，之后可以向 RTC_CNTL/H、RTC_ALRL/H、RTC_MSRL/H 或 RTC_PRL/H 寄存器写入数据。由软件清'0'后，才能进入下一次对寄存器的写操作过程。 0：退出配置模式 1：进入配置模式
3	RSF	寄存器同步标志（Registers Synchronized Flag） 由软件清零，由硬件置 1。 用户必须等待这位被硬件置'1'，才能进行读操作。确保同步后的 RTC_CNT、RTC_ALR、RTC_PRL、RTC_DIV 以及 RTC_MSR 寄存器的值正确。 0：寄存器没有同步 1：寄存器已经同步

Bit	Field	Description
2	OWF	溢出标志（Overflow Flag） 当可编程计数器溢出时，硬件置'1'，通过配置 RTC_CRH 寄存器中的 OWIE 位为 1，可以产生中断。 软件清除该位 0：无溢出 1：32 位可编程计数器溢出
1	ALRF	闹钟标志（Alarm Flag） 可编程计数器等于 RTC_ALR 寄存器设定值时，硬件置'1'，通过配置 RTC_CRH 寄存器中的 ALRIE 位为 1，可以产生中断。 软件清除该位 0：无闹钟产生 1：有闹钟产生
0	SECF	秒标志（Second Flag） 预分频器溢出时，硬件置'1'该位，RTC 计数器加 1。该标志为可编程的 RTC 计数器提供计数时钟（常用周期为 1 秒）通过配置 RTC_CTH 寄存器中的 SECIE 位为 1，可以产生中断。软件清除该位 0：没有秒标志 1：产生秒标志

注：

- 当状态位 RTOFF=0 时，表示上一次写操作没有完成，此时不能对 RTC_CR 寄存器进行写操作；
- 复位时禁止所有中断，无任何挂起的中断请求，可以写 RTC 寄存器；
- 硬件置位 OWF、ALRF、SECF 和 RSF 等标志位，软件清零；
- 若关闭 APB1 时钟，OWF、ALRF、SECF 和 RSF 位值保持不变；
- 标志位置位后将保持挂起状态，直到中断响应且软件配置 RTC_CR 清除该标志位；

16.4.4 预分频装载寄存器高位(RTC_PRLH)

偏移地址：0x08 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRL[19:16]			
												w			

Bit	Field	Description
15:4	Reserved	保留，必须保持复位值
3:0	PRL[19:16]	RTC 预分频器重装载值高位（RTC Prescaler Reload Value High） 计数器的时钟频率计算公式如下： $f_{SC_CLK} = f_{RTC_CLK} / (PRL[19:0] + 1)$ 注：不推荐配置为 0，会产生错误的 RTC 中断和标志位。

16.4.5 预分频装载寄存器低位(RTC_PRLI)

偏移地址：0x0C

复位值：0x8000

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

Bit	Field	Description
15:0	PRL[15:0]	RTC 预分频器重装值低位（RTC Prescaler Reload Value Low）

16.4.6 预分频器分频因子寄存器高位(RTC_DIVH)

偏移地址：0x10 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DIV[19:16]			
												r			

Bit	Field	Description
15:4	Reserved	保留，必须保持复位值
3:0	DIV[19:16]	RTC 时钟分频器分频因子高位（RTC Clock Divider High） 只读寄存器，在 RTC_PRL 或 RTC_CNT 寄存器被软件更新后，由硬件重新装载。 在每个 SC_CLK 时钟周期里，RTC 预分频器的值会被 RTC_PRL 寄存器的值重置。通过 读取 RTC_DIV 寄存器可以得到预分频计数器的当前值或当前精确的时间。

16.4.7 预分频器分频因子寄存器低位(RTC_DIVL)

偏移地址：0x14 复位值：

0x0000

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

Bit	Field	Description
15:0	DIV[15:0]	RTC 时钟分频器分频因子低位 (RTC Clock Divider Low) 只读寄存器，在 RTC_PRL 或 RTC_CNT 寄存器被软件更新后，由硬件清零。 在每个 SC_CLK 时钟周期里，RTC 预分频器的值会被 RTC_PRL 寄存器的值重置。通过 读取 RTC_DIV 寄存器可以得到预分频计数器的当前值或当前精确的时间。

16.4.8 计数器寄存器高位(RTC_CNTH)

偏移地址：0x18 复位值：

0x0000

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

Bit	Field	Description
15:0	CNT[31:16]	<p>RTC 计数器高位 (RTC Counter High)</p> <p>通过读 RTC_CNTH 寄存器获取 RTC 计数器当前值的高位部分。先进入配置模式。才能对该寄存器执行写操作。</p> <p>该寄存器受 RTC_CR 的位 RTOFF 写保护。</p> <p>写入该寄存器的值，会直接加载到对应的编程计数器。软件可通过读取该寄存器的值，获得计数器计数值或显示系统时间。</p>

16.4.9 计数器寄存器低位(RTC_CNTL)

偏移地址: 0x1C 复位值:

0x0000

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

Bit	Field	Description
15:0	CNT[15:0]	<p>RTC 计数器低位 (RTC Counter Low)</p> <p>通过读 RTC_CNTL 寄存器获取 RTC 计数器当前值的低位部分。先进入配置模式。才能对该寄存器执行写操作。</p> <p>该寄存器受 RTC_CR 的位 RTOFF 写保护。 写入该寄存器的值，会直接加载到对应的编程计数器。软件可通过读取该寄存器的值，获得计数器计数值或显示系统时间。</p>

16.4.10 闹钟寄存器高位(RTC_ALRH)

偏移地址: 0x20 复位值:

0xFFFF

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

Bit	Field	Description
15:0	ALR[31:16]	<p>RTC 闹钟值高位 (RTC Alarm High)</p> <p>软件写入的闹钟时间的高位部分。先进入配置模式，才能对该寄存器执行写操作。</p>

16.4.11 闹钟寄存器低位(RTC_ALRL)

偏移地址: 0x24

复位值：0xFFFF

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

Bit	Field	Description
15:0	ALR[15:0]	RTC 闹钟值低位（RTC Alarm Low） 软件写入的闹钟时间的低位部分。先进入配置模式，才能对该寄存器执行写操作。

16.4.12 毫秒闹钟寄存器高位 (RTC_MSRH)

偏移地址：0x28 复位值：

0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MSR[19:16]			
												rw			

Bit	Field	Description
15:4	Reserved	保留，必须保持复位值
3:0	MSR[19:16]	RTC 闹钟值高位（RTC Msec High） 软件写入的毫秒闹钟时间的高位部分 注：进入配置模式，才能对该寄存器执行写操作。

16.4.13 毫秒闹钟寄存器低位 (RTC_MSRL)

偏移地址：0x2C 复位值：

0x0000

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

Bit	Field	Description
15:0	MSR[15:0]	RTC 闹钟值低位（RTC Msec Low） 软件写入的毫秒闹钟时间的低位部分 注：进入配置模式，才能对该寄存器执行写操作。

- 若 RTC_ALR 为 0，RTC_MSR 不为 0：当可编程计数器的值与 RTC_MSR 中的 32 位值相等时，即触发一次毫秒事件，并发生闹钟中断；
- 若 RTC_ALR 不为 0，RTC_MSR 不为 0：当秒事件触发后，毫秒事件也触发，则会发生闹钟中断；
- 若 RTC_ALR 不为 0，RTC_MSR=0：则当秒事件触发后会发生闹钟中断；

- 此寄存器受 RTC_CR 寄存器里的 RTOFF 位写保护，仅当 RTOFF=1 时，允许对该寄存器写操作。

16.4.14 RTC LSE 配置寄存器 (RTC_LSE_CFG)

偏移地址：0x3C 复位值：

0x0198

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							LSE_RFB_SEL			Res.		LSE_IB			
							rw					rw			

Bit	Field	Description
15:9	Reserved	保留，必须保持复位值。
8:6	LSE_RFB_SEL	反馈电阻选择： 000: 4MΩ 100: 7MΩ 110: 10MΩ 111: 13MΩ
5:4	Reserved	保留，必须保持复位值。
3:0	LSE_IB	LSE 偏置电流调节： 从 0000~1111 调节逐渐变大，默认配置为 1000

17 USART 通用同步异步收发器

17.1 介绍

本通用同步/异步收发器（USART）可以灵活地与外部设备进行全双工数据交换。通过内置波特率（包含整数及小数设定）发生器，USART 可以支持宽范围的波特率。

USART 支持异步模式（UART）、同步模式。其中 UART 支持单线半双工通信，UART 和同步模式支持调制解调器（CTS/RTS）操作。

本产品支持 3 通道 USART 接口，其中 USART1、USART2 支持时钟同步功能，USART3 不支持时钟同步功能。

通道	单线半双工	硬件流控	异步模式（UART）	同步模式
USART1	√	√	√	√
USART2	√	√	√	√
USART3	√	√	√	-

注：“√”表示支持此功能；“-”表示不支持此功能。

17.2 USART 特性

- 支持 DMA
- 支持全双工异步通信，全双工时钟同步通信
- 波特率发生器（包含整数及分数配置）
 - ◆ 可编程波特率，供发送器和接收器使用（最小分频系数为 1）
- 独立的发送和接收缓冲寄存器，且发送器和接收器可单独使能。
- 支持 LSB、MSB 收发模式
- 数据位长度可编程（8 或 9 位）
- 停止位可配置（1/2 位）
- 校验位功能可配置（奇校验、偶校验，无校验）
- 支持空闲帧的产生（TE 使能时自动输出）和接收检测
- 支持硬件自动流控制（配备 CTS，RTS 引脚功能）
- 支持信号收发引脚的互换，接收和发送信号的电平取反
- 支持下面中断源：
 - ◆ 发送端数据寄存器为空（TXE）
 - ◆ CTS 标志（CTS）

- ◆ 发送完成（TC）
- ◆ 接收端数据有效（RXNE）
- ◆ 接收缓冲器溢出（OVR）
- ◆ 接收空闲帧完成（IDLE）
- ◆ 奇偶校验错误（PE）
- ◆ 噪声标志（NF）和帧错误（FE）

17.3 USART 功能描述

17.3.1 功能框图

USART 的功能框图可参考如下，可分为寄存器相关的控制单元、收发数据控制器、时钟控制器、硬件流控制单元以及引脚控制逻辑单元。

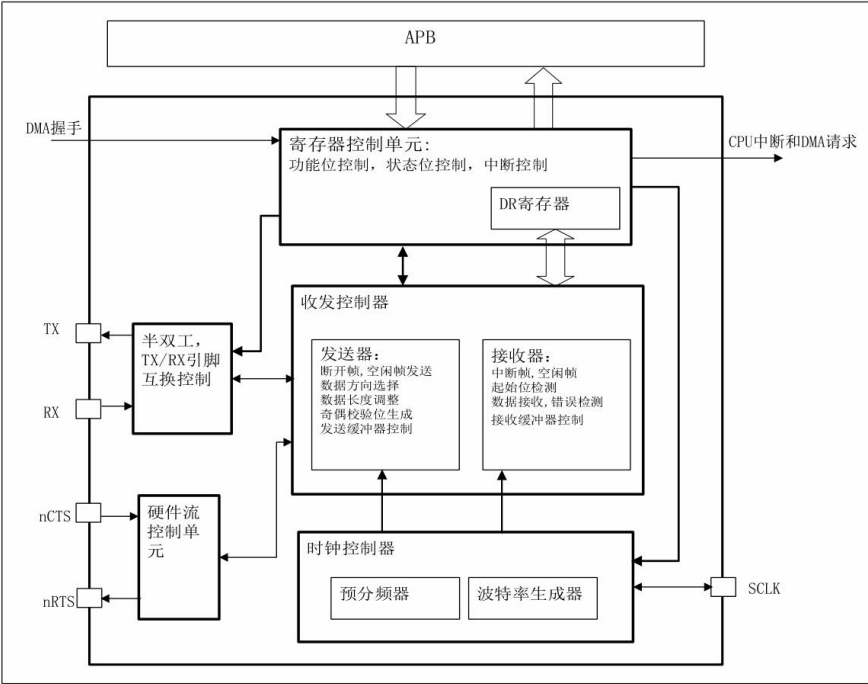


图 17-1 USART 功能框图

17.3.2 信号描述

信号名	类型	描述
USART_SCLK	输出或输入	同步模式下的输入或输出时钟引脚、
USART_TX	输出或输入	发送数据引脚，或半双工收发数据引脚

信号名	类型	描述
USART_RX	输入	接收数据引脚（全双工时）
USART_nCTS	输入	清除发送引脚
USART_nRTS	输出	请求发送引脚

17.3.3 功能概述

在全双工通信的情况下，至少需要分配两个脚给 USART：接收数据输入（RX）和发送数据输出（TX）。

RX：外部串行数据通过该引脚，传送给 USART 接收器。对于传输过程中产生的噪音，可以使用 过采样的技术将其与数据区分并剔除，得到原本的数据。

TX：USART 发送器内部产生的串行数据通过该引脚发送输出。当发送器被使能，并且无数据发送时，TX 引脚输出高电平。

空闲状态为总线在开始发送或者开始接收前的初始状态。 起始位

为一位，用‘0’表示。

在 USART 通信中，一个数据（8 或 9 位）的发送和接收顺序可配置为从最低位到最高位（LSB），或最高位到最低位（MSB）。

停止位用‘1’表示一帧的结束，位数可配置为 1、2 位。

通过编程波特率寄存器 USART_BRR 来控制内部的波特率发生器以得到期望的数据传输波特率（参考 1.3.5 章节的波特率计算公式）。

通过配置 USART_CR2 寄存器的 SWAP 位，可以交换接收和发送端的引脚。

通过配置 USART_CR2 寄存器的 RXTOG/TXTOG 位，可以将接收/发送端的电平信号取反（包含 起始位、停止位）。

在硬件流控模式中还需要下列引脚：

- **nCTS 清除发送：**当其为高电平时，表明当前接收端不能进行数据接收，发送端应停止之后的发送。
- **nRTS 请求发送：**当其为低电平时，表明当前接收端可以接收数据。

另外，USART 模块还支持同步模式（不同于 UART），此时需要下列引脚：

- **SCLK 发送器时钟输出或时钟输入：**该引脚用于同步模式，同步模式下，支持时钟的输入和输出功能，且时钟极性和相位可通过软件配置。

17.3.4 特性描述

通过配置 USART_CR1.DL 位，可调整字符位长度为 8 或 9 位。发送器会在发送起始位时拉低 TX 引脚，在发送停止位时拉高 TX 引脚。

包括停止位在内，一个完全由‘1’组成的完整数据帧，定义为一个空闲帧。下一个数据帧的起始位跟在空闲帧之后。

包括停止位在内，一个完全由‘0’组成的完整数据帧，定义为一个断开帧。在断开帧结束后，发送器会再发送一个‘1’的停止位，使得下一帧的起始位能够被识别到（产生下降沿被接收端检测到）。

波特率发生器产生的时钟经过发送器或接收器的使能位置位控制之后，供给发送器或接收器使用。

下图为数据帧格式，断开帧，以及空闲帧的样例。

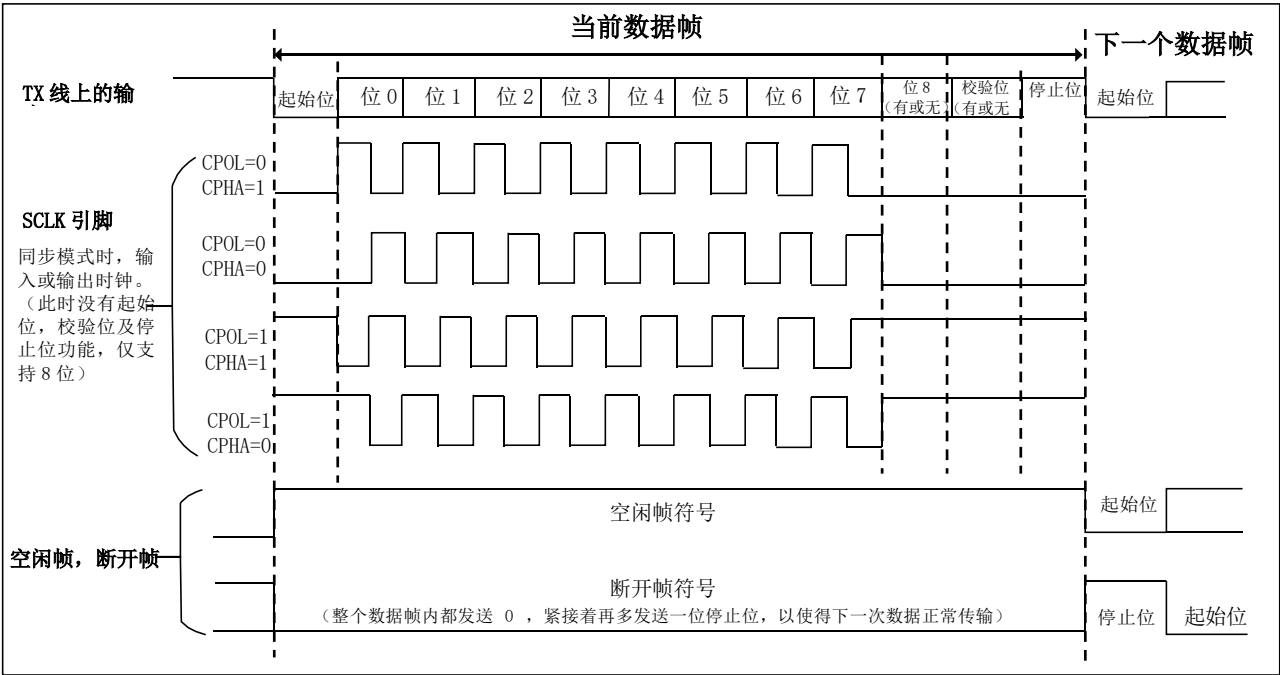


图 17-2 UART 数据帧类型示意图

17.3.5 波特率发生器

各通信模式下，可按照以下公式配置波特率。

- 异步（UART）模式： 通信波特率为：

$$f_{baudrate} = \frac{PCLK}{N \times (MFD + FFD/N)}$$

误差 E (%) 为：

$$E(\%) = \{ \frac{PCLK}{f_{baudrate} \times N \times (MFD + FFD/N)} - 1 \} \times 100$$

上述公式中，PCLK 为内部时钟源的频率；MFD、FFD 为 波特率寄存器 USART_BRR 中的整数和小数分频；N=8x(2-OVER8)，其中 OVER8 为控制寄存器 USART_CR1 中的过采样模式选择。当 OVER8=1（8 倍过采样），FFD[3:0] 只使用低 3 位，用户应配置 FFD[3] 位为 1'h0。

- 同步模式（小数分频配置无效，软件配置 DIV_fraction[3:0] 位为 4'h0）：

通信波特率为：

$$f_{baudrate} = \frac{PCLK}{4 \times MFD}$$

上述公式中，PCLK 为内部时钟源的频率；MFD 为波特率寄存器 USART_BRR 中整数分频。同步模式下小数分频（FFD）无效，用户应配置 FFD[3:0] 位为 4'h0。

17.3.6 采样

UART 内置检测电路检测一帧数据的开始，并对 RX 引脚进行采样，UART 采用 8 或 16 倍数据波特率的时钟采样 RX 引脚的数据。

可配置 USART_CR1.OVER8 位，来选择 USART 采用 16 倍或 8 倍数据波特率的时钟进行 RX 引脚的数据采样。

当选择 8 倍过采样（OVER8=1）时，可获得更高的速度（高达 $f_{PCLK}/8$ ），但接收器对时钟偏差的最大容差将会降低。

17.3.7 奇偶校验控制

配置 USART_CR1 寄存器中的 PCE 位为‘1’来使能奇偶校验控制，PS 位用来选择奇校验或偶校验。

偶校验：校验位加上数据中‘1’的总数为偶数。

奇校验：校验位加上数据中‘1’的总数为奇数。奇偶校验控制有效

时：

- 发送器会自动生成一个奇偶校验位，并在停止位前输出。
- 接收器会对奇偶校验位进行检测并判断该位是否错误。如果奇偶校验位错误则硬件自动置位 USART_SR.PE 标志，但当前接收数据仍然会从移位寄存器传送到 USART_DR 寄存器。

17.3.8 发送器

配置 USART_CR1.TE 位为‘1’，使能发送器，数据会串行输出到 TX 引脚上。发送数据寄存器 USART_DR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。其中 UART 模式可以通过设置 USART_CR1.DL 位来选择长度（8 或 9 位）。

17.3.8.1 字符发送

在 USART 发送期间，数据从 USART_DR 寄存器写入，通过发送移位寄存器将数据帧字节以最低位到最高位（USART_CR1.MLS=0），或最高位到最低位（USART_CR1.MLS=1）的顺序在 TX 引脚上输出。

发送数据顺序：1 位起始位，字符，1 位奇偶校验位（有或无），停止位。

可通过寄存器 `USART_CR2.STOP[1:0]` 来配置停止位的个数（可配置为 1 或 2 位停止位）。当前数据传输未完成前不能清零 `USART_CR1.TE` 位，否则波特率发生器会停止产生时钟，导致该数据后部分丢失。

17.3.8.2 发送断开帧

配置 `USART_CR1.SBK` 位为‘1’即可发送断开帧。如果在数据传输过程中配置 `SBK=1`，则当前数据发送完成后才会输出断开帧到 TX 引脚上。

断开帧发送完成时，硬件自动清零 `SBK` 位，并发送一位高电平的停止位（以保证相连的下一帧数据的起始位被检测到）。

断开帧的长度取决于数据帧长度（`CR2.DL`），奇偶校验使能位（`CR1.PS`），以及停止位（`CR2.STOP`）。例如，无奇偶校验且停止位为 1 位时，`CR2.DL=0` 则断开帧为 10 位连续的‘0’，`CR2.DL=1` 则断开帧为 11 位连续的‘0’。

17.3.8.3 发送配置步骤

可参考下面的步骤，来配置 USART 进行数据帧的发送：

1. 配置 USART 所需要使用的引脚功能。
2. 使能 USART（`USART_CR1.UE=1`）。
3. 配置 `USART_BRR` 寄存器。
4. 根据传输数据帧等需求配置 `USART_CR1`，`USART_CR2`，`USART_CR3` 寄存器。
5. 使能发送器（`USART_CR1.TE=1`），如果需要使用发送数据寄存器空中断，则设置 `USART_CR1.TXEIEN=1`。
6. 等待发送数据寄存器空，写通信数据到 `USART_DR`，数据传输到发送移位寄存器，发送开始（CTS 功能有效时，`USART_CTS` 输入为低电平时数据传输到发送移位寄存器，发送开始）。
7. 如果需要连续发送数据时，重复步骤 6。
8. 通过确认 `USART_SR.TC` 位确认发送是否完成。如果配置 `TCIEN=1`，则最后一帧数据发送结束后，产生发送完成中断。

注：USART 的发送器支持两种中断，即发送数据寄存器空中断 TXE 和发送完成中断 TC，可通过 `USART_SR` 寄存器中的状态位查询。配置 `TXEIEN=1`，当 `USART_DR` 寄存器的值传送到发送移位寄存器时产生 TXE 中断。配置 `TCIEN=1`，当发送数据的最后一位时 `USART_DR.DR` 寄存器没有写入更新数据则产生 TC 中断。

17.3.9 接收器

数据寄存器 USART_DR 寄存器和内部的接收移位寄存器组成双缓冲器结构，可以连续接收数据。

其中 UART 模式可以通过设置 USART_CR1.DL 位来选择数据长度，8 位或 9 位。

接收器使能位 USART_CR.RE 置'1'并检测到开始位后，RX 管脚上数据接收到接收移位寄存器；当接收到一帧数据，数据从接收移位寄存器传送到数据寄存器 USART_DR 中，同时，状态标志 RXNE 将置'1'。配置 RXNEIEN=1 则许可该中断请求。

CPU 或 DMA 利用该请求读取接收数据时，一次请求只能读取一次数据。接收数据的顺序为：开始位->数据位（MSB/LSB）->校验位（有或者无）->停止位。

17.3.9.1 接收断开帧

USART 接收器识别到一个断开帧时，会置位 USART_SR.FE 标志（等同于在停止位接收到'0'）。

17.3.9.2 接收空闲帧

UART 正常工作时，接收器接收到了一个空闲帧会置位 USART_SR.IDLE 标志。

配置 IDLEIEN=1 来许可 IDLE 中断请求。

17.3.9.3 接收配置步骤

可参考下面的步骤，来配置 USART 进行数据帧的接收：

1. 设定 UART 所需要使用的功能引脚。
2. 使能 USART（USART_CR1.UE=1）。
3. 配置 USART_BRR 寄存器。
4. 根据数据帧等需求配置来设定 USART_CR1，USART_CR2，USART_CR3 寄存器。
5. USART_BRR 寄存器设定通信波特率（时钟源为外部时钟源时不需要设定）。
6. 使能接收器（USART_CR1.RE=1），如果需要使用接收中断，则设置 USART_CR1.RXNEIEN=1。

7. 当检测到开始位后，接收器将数据接收到接收移位寄存器，并检查校验位和停止位。一共有三种错误标志状态：PE，FE，ORE。当没有错误发生时，接收到的数据从接收移位寄存器传送到 USART_DR 寄存器，并置 RXNE 标志位为'1'。

8. 可通过 RXNE 中断读取接收到的数据，重复步骤 7 即可连续接收数据。

9. 在接收期间如果检测到接收错误时，对应的错误标志将被置位。注：为了防止溢出错误，在下一字符接收结束前，RXNE 位必须被清零（软件读取数据寄存器 USART_DR）。当发生 PE，FE，ORE 中任何一种接收错误都不能再进行数据的接收，但可以通过将

所有的错误标志清零来重启数据接收。

- 发生上溢错误时接收到的数据丢失，ORE 状态位置‘1’，但 RXNE 中断不发生。
- 发生奇偶校验错误时接收到的数据传送给 USART_DR，PE 状态位置‘1’，但 RXNE 中断不发生。
- 发生帧错误时接收到的数据传送给 USART_DR，FE 状态位置‘1’，但 RXNE 中断不发生。

17.3.10 同步模式

通过配置 USART_CR1.SAS 位为‘1’来使能同步模式（时钟引脚功能将同时有效）。在同步模式下，USART_CR3.HDSEL 位应配置为‘0’。同步模式支持主模式和从模式：主模式时使用内部波特率生成器生成的时钟，同时输出时钟；从模

式时由 SCLK 引脚输入时钟。USART 在同步模式下，能与 SPI 实现数据通信（此时，用户应配置 SPI 与 USART 的时钟极性、时钟相位为一致）。

17.3.10.1 时钟描述

配置 USART_CR2.CLKEN 位为‘1’来使能时钟引脚功能，同时根据 USART_CR3.CKINE 位配置来选择使用内部波特率时钟或从 SCLK 引脚输入时钟，以进行数据通信。

当选择内部波特率时钟时，可通过 SCLK 引脚输出同步时钟。

1 帧数据的收发包含 8 个时钟脉冲。

当 RE 和 TE 都为‘0’，时钟输出会停止，并固定在 USART_CR2.CPOL 配置的电平。

通过配置 USART_CR2.CPOL 位选择时钟极性；通过配置 USART_CR2.CPHA 位选择外部时钟相位。

17.3.10.2 时钟同步功能描述

SCLK 引脚作为发送器的时钟输出时，仅在数据段输出时钟，一帧数据输出 8 个时钟脉冲，最后一位发送完后，通信线保持最后一位的值，时钟输出固定在高电平或低电平（由 CPOL 位决定）。

USART 接收器在同步模式下的工作方式与异步模式下不同。如果 RE=1，则数据在 SCLK 变化边沿上采样（上升或下降沿，取决于 CPOL 和 CPHA 位配置情况），而不会进行任何过采样。此时必须确保足够的建立时间和保持时间，以符合时序要求（类同于 SPI 协议）。

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$f_{baudrate} = \frac{PCLK}{4 \times MFD}$$

其中通信波特率的单位为 Mbps；PCLK 为内部时钟源的频率；MFD 为波特率寄存器 USART_BRR 中整数分频（注意，在同步模式下应配置 $MFD \geq 2$ ，且小数分频 FFD 无效，用户应配置 FFD[3:0] 位为 4'h0）。

使用内部时钟源且 MFD=2 时，同步模式的最高波特率为 PCLK/8（MBps）。外部时钟源时，要求外部输入时钟的最大频率为 PCLK/8（MHz），此时最高波特率也为 PCLK/8（MBps）。

17.3.11 单线半双工通信

配置 USART_CR3.HDSEL 位为‘1’，进入单线半双工模式。 单线半双工模式下的芯片内部逻辑会将 TX 与 RX 互连，同时：

- RX 引脚悬空，不参与传输。传输时 USART 的 TX 直接连接另一个 USART 的 TX。
- 在传输数据时，TX 一直被占用，直到停止位发送完成。
- 在没有传输数据时，TX 处于被释放状态。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口；TX 对应 I/O 在不被 USART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除了单线引脚的配置外，其余配置和正常传输时一致。

在没有通信前，两个 USART 的 RXEN 都开启，处于等待接收状态，当需要通信时，两个 USART 要约定好谁来发送，发送方的 USART_CR1 寄存器中 RE 关闭，TE 使能。如果两边 UART 都试图发送数据，将产生发送冲突（硬件不会阻碍 USART 的发送：当发送使能位 TE 开启，只要写 USART_DR，TX 就会发送数据）。

17.3.12 硬件流控

USART 模块支持使用 nCTS 输入和 nRTS 输出，以控制 2 个器件间的串行数据流。

配置 USART_CR3 寄存器中的 RTSE 位和 CTSE 位，分别使能 RTS 和 CTS 流控制。

如果使能 RTS 流控制，只要 USART 接收器准备好接收新数据，则 nRTS 输出低电平，请求发送。当接收寄存器已满或产生接收错误（上溢错误，帧错误和奇偶校验错误）时，则 nRTS 输出高电平，指示发送方中止下一帧数据的发送。

如果使能 CTS 流控制，则发送器会在发送下一帧数据前检查 nCTS。如果 nCTS 输入为低电平，则会立即发送下一帧数据（假设数据已准备好发送，即 TXE=0）；否则不会进行发送。在发送过程中 nCTS 的变化不会影响当前帧数据的发送。

当 USART_CR3.CTSE=1 时，只要 nCTS 发生变化，USART_SR.CTS 标志位便会由硬件自动置位，此时如果已配置 USART_CR3.CTSIEN 位为‘1’，则会产生中断。

17.3.13 中断

本 USART 模块可以支持下面的中断源：

表 17-1 UART 中断请求

中断事件	中断状态位	使能位	UART	同步模式
发送数据寄存器为空	TXE	TXEIE	√	√

中断事件	中断状态位	使能位	UART	同步模式
CTS 标志	CTS	CTSIEN	√	√
发送完成	TC	TCIEN	√	√
接收数据寄存器满	RXNE	RXNEIEN	√	√
检测到空闲线路	IDLE	IDLEIEN	√	-
奇偶校验错误	PE	PEIEN	√	-
噪声标志	NF	ERRIEN	√	-
上溢错误	ORE	ERRIEN	√	√
帧错误	FE	ERRIEN	√	-

注：“√”表示使用该中断。“-”表示不使用该中断。

17.3.14 DMA

USART 可以利用 DMA 来搬运数据（需提前配置好 DMA），需要配置 USART_CR3.DMAMODE 位为‘1’来激活 DMA 模式。

利用 DMA 发送：DMA 搬运发送数据到 USART_DR 。使能 USART_CR1 的 UE，TE 后，只要发 送缓冲为空，就请求 DMA 搬运发送数据到 USART_DR。

利用 DMA 接收：DMA 将 USART_DR 中的接收数据搬走。使能 USART_CR1 的 UE，RE 后，只 要接收缓冲器中存在有效数据（USART_SR.RXNE=1），就请求 DMA 进行搬运操作。

17.4 寄存器

17.4.1 寄存器总览

表 17-2 USART 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	USART_SR	状态寄存器	0x0000_00C0
0x04	USART_DR	数据收发寄存器	0x0000_01FF
0x08	USART_BRR	波特率控制寄存器	0x0000_0000
0x0C	USART_CR1	控制寄存器 1	0x0000_0000
0x10	USART_CR2	控制寄存器 2	0x0000_0000
0x14	USART_CR3	控制寄存器 3	0x0000_6000

17.4.2 USART_SR 状态寄存器

偏移地址：0x00 复位值：

0x0000 00C0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rcw0		r	r	rc_w0	r	r	r	r	r

Bit	Field	Description
31:10	Reserved	保留，必须保持复位值
9	CTS ^{*注 1}	CTS 标志 0: CTS 线上未 发生状态变化 1: CTS 线上发生状态变化
8	Reserved	保留，必须保持复位值
7	TXE	发送数据寄存器为空标志（Transmit data register Empty） 0: 发送数据寄存器非空（数据未传送到移位寄存器） 1: 发送数据寄存器为空（数据已送到移位寄存器） TXE 位由硬件自动置‘1’及清‘0’，数据未传送到移位寄存器时硬件自动清零该位（写入 DR 寄存器时）；当 TE=0 或数据由 DR 传送到移位寄存器时硬件自动置位该位。
6	TC	发送完成标志（Transmission complete） 0: 发送未完成 1: 发送完成 TC 清零的条件： 在 TE=1 时向数据寄存器写入发送数据。 TC 的置位条件： TE=0 或数据帧最后一位送出时没有写 USART_DR 来更新发送数据寄存器。
5	RXNE ^{*注 1}	接收数据寄存器不为空标志（Read data register not empty） 0: 未接收到有效数据 1: 接收到有效数据 注：RXNE 位由硬件置‘1’及清‘0’，用户也可以通过向该位写入‘0’来清零。接收到有效数据时硬件自动将 RXNE 置‘1’，读取接收数据后硬件自动将 RXNE 清零。
4	IDLE ^{*注 2}	空闲帧检测标志（IDLE frame detected） 0: 未检测到空闲帧 1: 检测到空闲帧 UART 检测到空闲帧时，该位由硬件自动置‘1’。
3	ORE ^{*注 2}	接收溢出错误标志（Overrun error） 0: 无接收溢出错误 1: 检测到接收溢出错误 注：在 RXNE=1（已有可读数据），又接收到一帧新的数据，该位将由硬件自动置‘1’。
2	NF ^{*注 2}	噪声检测标志（Noise detected flag） 0: 未检测到噪声 1: 检测到噪声 注：当在接收的信号线上检测到噪声时，该位由硬件自动置‘1’。
1	FE ^{*注 2}	接收帧错误标志（Framing error） 0: 无接收帧错误 1: 发生接收帧错误 该位由硬件自动置‘1’，置位条件： 异步模式（UART）时，接收数据帧的停止位为低电平。 注：FE=1 时，接收的数据会从移位寄存器传送到数据寄存器，但不产生 RXNE 中断请求，同时将停止后续的数据接收动作。
0	PE ^{*注 2}	奇偶校验错误标志（Parity error） 0: 无奇偶校验错误 1: 奇偶校验错误 在接收数据时检测到了奇偶校验错误，该位由硬件自动置‘1’。 注：PE=1 时，接收的数据会从移位寄存器传送到数据寄存器，但不产生 RXNE 中断请求，同时将停止后续的数据接收动作。

*注 1：通过软件写‘0’可将该位清零。

*注 2：通过软件序列可将该位清零（读取状态寄存器，然后对 USART_DR 数据寄存器执行读访

问)。

17.4.3 USART_DR 数据寄存器

偏移地址：0x04 复位值：

0x0000 01FF

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

Bit	Field	Description
31:9	Reserved	保留，必须保持复位值
8:0	DR[8:0]	发送/接收数据寄存器 包含接收到的数据字符，或要发送的数据字符，取决于所执行的操作是“读取”操作还是“写入”操作：读取时，表示接收到的数据；写入时，表示准备要发送的数据。 注：最高位 DR[8] 只在异步模式（UART）且数据长度为 9 位（DL=1）时有效。

17.4.4 USART_BRR 波特率寄存器

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

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
19:4	MFD [15:0]	波特率整数分频（Mantissa frequency division of baudrate） 这 16 位用于定义 USART 波特率的整数分频。 使能发送或接收（TE 或 RE 设定为 1）之前，用户应根据波特率需求配置该整数分频值。 注：需配置 MFD≥1。

Bit	Field	Description
3:0	FFD[3:0]	波特率小数分频 (Fraction frequency division of baudrate) 这 4 位用于定义 USART 波特率的小数分频。 使能发送或接收 (TE 或 RE 配置为‘1’) 之前, 用户应根据波特率需求配置该小数分频值。 注: 配置 FFD[3:0]=4'h0 则小数分频无效。异步模式 (UART) 且 USART_CR1.OVER8=1 时, 最高位 FFD[3] 无效, 用户应配置 FFD[3]=0。在同步模式时, 小数分频无效, 用户应配置 FFD[3:0]=4'h0。

17.4.5 USART_CR1 控制寄存器 1

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
Reserved														SAS	MLS
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	UE	DL	Res.	PCE	PS	PEIEN	TXEIE	TCIEN	RXNEIE N	IDLEIEN	TE	RE	Res.	SBK
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31:18	Reserved	保留, 必须保持复位值
17	SAS	通信模式选择位 (synchronous/asynchronous mode selection) 0: 异步模式 (UART) (asynchronous) 1: 同步模式 (synchronous) 注: 用户应在 TE=0 和 RE=0 时配置此位。
16	MLS	MSB/LSB 方式选择位 0: LSB 方式 1: MSB 方式 注: 用户应在 TE=0 和 RE=0 时配置此位。
15	OVER8	UART 过采样模式 (Oversampling mode) 0: 16 倍过采样 1: 8 倍过采样 注: 用户应在 TE=0 和 RE=0 时配置此位。
14	Reserved	保留, 必须保持复位值
13	UE	USART 使能 (USART enable) 0: 禁止 USART 预分频器和引脚输出 1: 使能 USART 该位清零后, USART 预分频器和输出将停止, 并会结束当前传输以降低功耗。 注: 该位由用户置 ‘1’ 和清零。
12	DL	数据长度 (Data length) 0: 8 位 1: 9 位 注: 用户应在 TE=0 和 RE=0 时配置此位。
11	Reserved	保留, 必须保持复位值
10	PCE	奇偶校验控制使能 (Parity control enable) 0: 禁止奇偶校验功能 1: 使能奇偶校验功能 注: 该位由用户置 ‘1’ 和清零。同步模式时应配置 PCE=0 (复位值)。

Bit	Field	Description
9	PS	校验模式 0: 偶校验 1: 奇校验 注: 该位由用户置 '1' 和 清零, 只在 PCE=1 时有效。
8	PEIEN	PE 中断使能 (PE interrupt enable) 0: 禁止 PE 中断请求 1: 允许 PE 中断请求 注: 该位由用户置 '1' 和 清零。
7	TXEIEIEN	TXE 中断使能 (TXE interrupt enable) 0: 禁止 TXE 中断请求 1: 允许 TXE 中断请求 注: 该位由用户置 '1' 和 清零。
6	TCIEN	传送完成中断使能 (Transmission complete Interrupt enable) 0: 禁止 TC 中断请求 1: 允许 TC 中断请求 注: 该位由用户置 '1' 和 清零。
5	RXNEIEN	RXNE 中断使能 (RXNE interrupt enable) 0: 禁止 RXNE 中断请求 1: 允许 RXNE 中断请求 注: 该位由用户置 '1' 和 清零。
4	IDLEIEN	IDLE 中断使能 (IDLE interrupt enable) 0: 禁止 IDLE 中断请求 1: 允许 IDLE 中断请求 注: 该位由用户置 '1' 和 清零。
3	TE	发送器使能 (Transmitter enable) 0: 禁止发送器 1: 使能发送器 注: 该位由用户置 '1' 和 清零。在同步模式下, 如需同时收发, 用户必须同时配置 TE 和 RE 位, 以保证时钟与数据收发的时序正常。
2	RE	接收器使能 (Receiver enable) 0: 禁止接收器 1: 使能接收器 注: 该位由用户置 '1' 和 清零。在同步模式下, 如需同时收发, 用户必须同时配置 RE 和 TE 位, 以保证时钟与数据收发的时序正常。
1	Reserved	保留, 必须保持复位值
0	SBK	发送断路 (Send break) 0: 不发送断开帧 1: 发送断开帧 该位用于发送断开帧, 可由用户置 '1', 并在发送完断开帧后由硬件自动清零。

17.4.6 USART_CR2 控制寄存器 2

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	CPOL	CPHA	Res.								
rw		rw			rw	rw									

Bit	Field	Description
31:15	Reserved	保留，必须保持复位值
15	SWAP	输入与输出引脚交换 0: IO 引脚功能不交换 1: IO 引脚功能的输入与输出交换 注: SWAP 置位后, GPIOx_CRL 寄存器的 MODE 需要更改, 如: 原输入模式变为输出模式。
14	Reserved	保留，必须保持复位值
13:12	STOP[1:0]	停止位 (STOP bit) UART 模式: 00: 1 个停止位 10: 2 个停止位 01: 保留 11: 保留
11	Reserved	保留，必须保持复位值
10	CPOL	时钟极性 (Clock POLarity) 0: 空闲时, 时钟为低电平。 1: 空闲时, 时钟为高电平。 注: 该位与 CPHA 位结合使用可获得所需的时钟/数据关系 (仅在同步时钟模式下有效)。
9	CPHA	时钟相位 (Clock PHase) 0: 在时钟第一个变化沿捕获数据 1: 在时钟第二个变化沿捕获数据 注: 该位与 CPOL 位结合使用可获得所需的时钟/数据关系 (仅在同步时钟模式下有效)。
8:0	Reserved	保留，必须保持复位值

17.4.7 USART_CR3 控制寄存器 3

偏移地址: 0x14 复位值:

0x0000 6000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TXTOG	RXTOG	Reserved											CKINE
		rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ONEBIT	CTSIEN	CTSE	RTSE	Res.	DMAMODE	Res.		HDSEL	Res.		ERRIEN
				rw	rw	rw	rw		rw			rw			rw

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29	TXTOG	发送取反位 0: 发送取反功能无效 1: 发送信号电平取反
28	RXTOG	接收取反 0: 接收取反功能无效 1: 接收信号电平取反
27:17	Reserved	保留，必须保持复位值

Bit	Field	Description
16	CKINE	同步模式下时钟输入许可 此位用于控制时钟是否由外部输入（CR1.SAS 位为‘1’时该位配置有效）： 0：时钟不由外部输入 1：时钟由外部输入 软件应在 TE=0 和 RE=0 时配置此位。
15:12	Reserved	保留，必须保持复位值
11	ONEBIT	UART 单次采样方式使能（One sample bit method enable） 0：三次采样（多数判决） 1：单次采样 注：用户应在 TE=0 和 RE=0 时配置此位。当选择一个采样位方法后，噪声检测标志（USART_SR.NF）失效。
10	CTSIEN	CTS 中断使能（CTS interrupt enable） 0：禁止 CTS 中断请求 1：允许 CTS 中断请求
9	CTSE	CTS 使能（CTS enable） 0：禁止 CTS 硬件流控制 1：使能 CTS 硬件流控制
8	RTSE	RTS 使能（RTS enable） 0：禁止 RTS 硬件流控制 1：使能 RTS 硬件流控制
7	Reserved	保留，必须保持复位值
6	DMAMODE	DMA 方式使能位（DMA mode enable bit） 0：选择轮询或中断方式 1：选择 DMA 方式
5:4	Reserved	保留，必须保持复位值
3	HDSEL	单线半双工选择（Half-duplex selection） 0：全双工模式 1：半双工模式
2:1	Reserved	保留，必须保持复位值
0	ERRIEN	错误中断使能（Error interrupt enable） 0：禁止错误中断请求 1：允许错误中断请求 错误中断包括 FE，ORE，NF 三种。 注：当使用 DMA 对 USART 进行 DR 的读写（USART_CR3.DMAMODE=1）时，应配置 ERRIEN=1，以许可向 CPU 发出通信异常的中断请求。

18 SPI 串行外设接口

18.1 简介

SPI（Serial Peripheral Interface）接口广泛用于不同设备之间的板级通讯，如扩展串行 Flash，ADC 等。许多 IC 制造商生产的器件都支持 SPI 接口。

SPI 允许 MCU 与外部设备以全双工、同步、串行方式通信。应用软件可以通过查询状态或 SPI 中断来通信。

18.2 功能框图

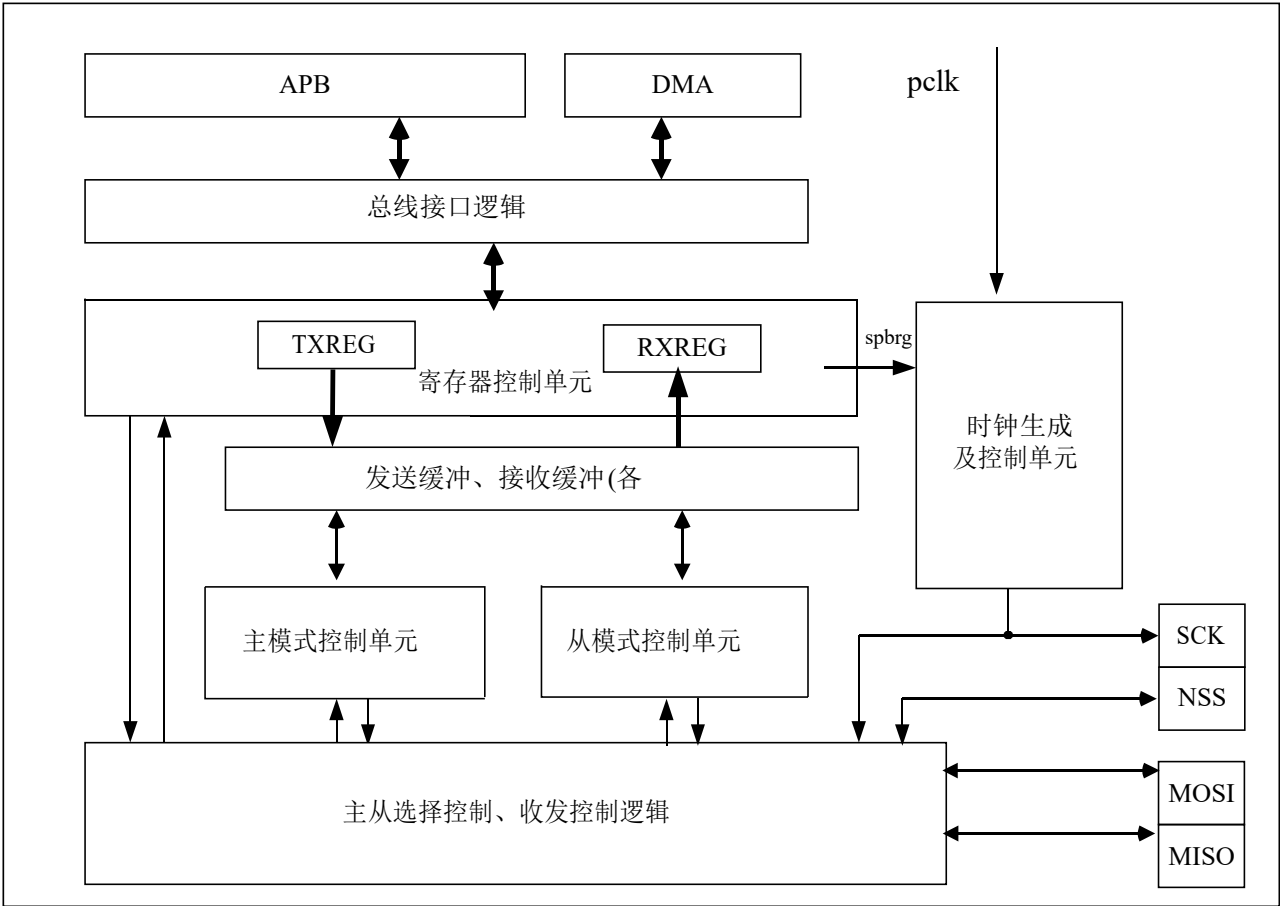


图 18-1 SPI 功能框图

18.3 功能描述

18.3.1 概述

SPI 支持同时接收和发送 1~32 位数据，但是仅在 8bit 传输时支持 LSB 和 MSB，1~32 位传输只支持 LSB 传输，可以被配置为从模式或在一个主机环境下的主模式。软件通过配置通用控制寄存器（CCTL）中的 CPOL 位和 CPHA 位，选择时钟与数据之间有四种不同的传输时序；并可配置 LSBFE 位选择使用 MSB 在前或者 LSB 在前来传输数据。

SPI 在时钟 SCK 的上升沿或下降沿发送数据，在相反的时钟有效沿接收数据。

使用 SPI 交换数据，要求必须在接收数据完成时进行数据读取操作，即使该数据不是有效数据；并且要求主机和与其通信的从机之间具有相同的时钟相位和极性。

SPI 通过 4 个引脚与外部器件相连：

- MISO：主设备输入、从设备输出引脚。传输方向为从设备发送到主设备。
- MOSI：主设备输出、从设备输入引脚。传输方向为主设备发送到从设备。
- SCK：串口时钟，由主设备产生并提供给从设备。
- NSS：从设备选择。该引脚为 SPI 协议下的可选功能，用于主设备选择与之通信的从设备（多从设备时），实现主从设备间的一对一单独通信，并能避免数据线上设备之间的冲突。当软件配置激活 NSS 引脚功能后，配置 SPI_GCTL.MODE 为‘1’使 SPI 工作于主模式，再配置 SPI_NSSR.NSS 为‘0’使 NSS 引脚输出低电平，从而允许与主设备连接的从设备（配置 SPI_GCTL.MODE 为‘0’）和它进行数据通信。

下图示意主、从设备之间一对一通信时的连接情况：

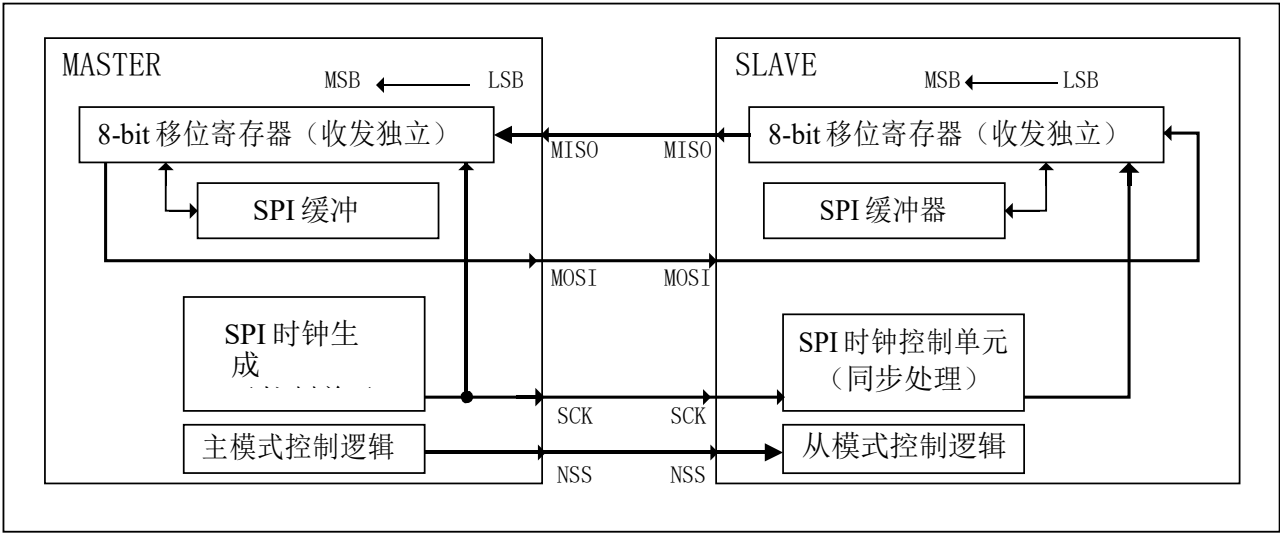


图 18-2 单主和单从应用

SPI 主、从设备的同名引脚互连，图中示意数据方向为从最高位到最低位的方式串行通信。主设备负责发起通信请求，从设备负责响应，从设备通过 SCK 引脚得到主设备提供的时钟信号，

从而使得主、从设备均使用同一个时钟进行同步的全双工通信。

对于从设备而言，MOSI 引脚输入来自主设备的发送数据，MISO 引脚输出响应数据传给主设备。

18.3.1.1 时钟信号的极性和相位

SPI_CCTL 寄存器的 CPOL 和 CPHA 位分别控制时钟的极性和相位，通过软件配置可分别得到 4 种不同的时钟/数据时序关系。

时钟极性指的是 SCK 时钟空闲状态下电平保持何种状态：如果配置 CPOL 位为‘0’，在空闲状态下 SCK 时钟保持为低电平；反之，则在空闲状态下 SCK 时钟保持为高电平。主、从设备都会受到 CPOL 控制位的影响。

时钟相位决定输入数据采样的时序：如果配置 CPHA 位为‘0’，第一个数据位会在 SCK 的第二个时钟边沿被采样；反之，则第一个数据位会在 SCK 的第一个时钟边沿被采样。

另外 CPHASEL 位在系统上电复位后为‘0’，当软件调整此位配置为‘1’后将会切换数据采样时序，也就是使得 CPHA 位的功能发生切换。例如，CPHASEL=1，CPHA=0 时：第一个数据位会在 SCK 的第一个时钟边沿被采样（CPOL 位为‘0’为上升沿；为‘1’则是下降沿）。

因此，需要根据 CPOL、CPHA 及 CPHASEL 位的组合配置来期望使用的时钟/数据时序关系。 时序配置需要注意以下几点：

- SPI 工作时不能修改 CPOL/CPHA（如需修改，首先关闭 SPI 的使能位 SPIEN）。
- 主、从设备同步通信，因此双方时序配置应保持一致。
- 空闲状态时 SCK 的电平情况必须和 CPOL 为配置的极性一致。

下图示意在 SPI 传输下的不同 CPHA 与 CPOL 位组合的 4 种情况（配置 CPHASEL 位为‘1’时），以及主/从设备的 SCK/MISO/MOSI/NSS 引脚的时序关系。

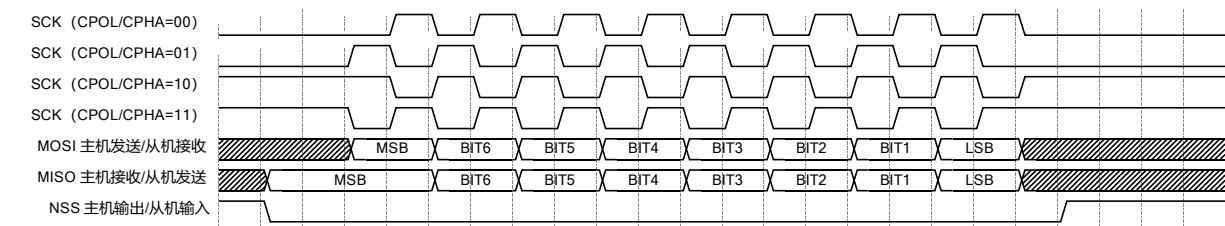


图 18-3 数据时钟时序图

需注意，配置 SPI_GCTL.NSS=1，即 NSS 引脚功能由硬件自动控制时，在数据通信完成后硬件将自动拉高此引脚（上图例中所示）；当配置 SPI_GCTL.NSS=0，NSS 输出状态则由从机片选寄存器 SPI_NSS 的 NSS 位进行控制（需软件切换 NSS 引脚的输出状态）。

18.3.1.2 高速传输

针对高速传输时板级延时敏感的情况，可配置 SPI_CCTL 寄存器中 TXEDGE/RXEDGE 位对数据发送/接收采样的时间点进行调整。高速传输，针对主设备的波特率发生器配置在 SPBRG<=4 且输出

SCK 时钟较快 ($\geq 10\text{MHz}$)；低速传输，针对主设备波特率发生器配置在 $\text{SPBRG} > 4$ 且输出 SCK 时钟较慢 ($< 10\text{MHz}$)。

- 在从模式下，TXEDGE 位为‘1’时，软件一旦配置 TXREG 写入数据，不等待 SCK 时钟输入的边沿而是立即发送到 MISO 引脚线上；TXEDGE 位为‘0’时，从设备总是等到一个有效时钟边沿才发送数据到 MISO 引脚线上。
- 在主模式下，RXEDGE 位为‘1’时，将后移采样数据的时间点，在传输数据位的尾时钟沿采样数据；RXEDGE 位为‘0’时，在传输数据位的中间点上采样数据。数据帧格式

配置 SPI_CCTL 寄存器中的 LSBFE 位，决定数据输出的优先顺序。LSBFE 位为‘1’会从最低位到最高位的顺序收发数据，为‘0’（缺省）会从最高位到最低位的顺序收发数据。

配置 SPI_CCTL 寄存器的 SPILEN 位，决定数据帧的数据长度。SPILEN 位为‘1’（缺省）则数据帧长是 8 位，为‘0’则数据帧长是 7 位。SPI 的发送和接收都受到数据帧格式配置的控制。

另外，可通过配置 SPI_GCTL.DW8_32=1，和寄存器 SPI_EXTCTL 来实现任意帧长的数据格式（帧长范围在 1~32 位之间）；任意帧长配置时也支持 LSBFE 位的功能（LSB 或 MSB 优先）。

在配合 DMA 进行数据传输时，需要将 DMA 的数据长度配置为 8bit。

18.3.2 SPI 主要特征

- 完全兼容 Motorola 的 SPI 规格
- 支持 DMA 请求
- 在 3 根线上支持全双工同步传输
- 16 位的可配置波特率生成器
- 支持主机模式和从机模式
- 支持一个主机与多个从机通信
- SPI 在主、从机模式下，时钟最快分别可达 $\text{PCLK}/2$ 、 $\text{PCLK}/4$ （PCLK 为 APB 时钟）
- 可配置的时钟极性和相位
- 可配置的数据帧长度（固定 8 位或 7 位帧长，1~32 位任意帧长）
- 可配置的数据顺序，MSB 在前或者 LSB 在前（1~32 位任意帧长支持 LSB、8 位数据收发都支持）
- 4 个字节的接收/发送缓冲，同时具备下列中断事件或状态供软件配置使用：
 - ◆ 发送缓冲为空
 - ◆ 发送缓冲和发送移位寄存器同时为空
 - ◆ 发送端下溢
 - ◆ 接收到有效字节
 - ◆ 接收缓冲上溢
 - ◆ 接收缓冲满

◆ 主模式下接收到指定的字节个数

18.3.3 SPI 从模式

SPI 作为从设备时，SCK 引脚输入来自主设备的串行时钟，因此从设备工作时不使用波特率发生器，不需配置寄存器 SPI_SPBRG（从设备下无效）。

18.3.3.1 配置步骤

1. 配置 SPI_GCTL.SPILEN，定义数据帧格式为 7 位或者 8 位；配置 SPI_GCTL.DW8_32 为‘1’，可配置 SPI_EXTCTL 寄存器来定义成任意帧格式（SPILEN 需固定为‘1’）。
2. 配置寄存器 SPI_CCTL 中的 CPOL、CPHA/CPHASEL 位，以确定时序模式。
3. 配置 SPI_CCTL.LSBFE，确定数据帧的收发顺序（LSB 或 MSB 位优先）。
4. 配置寄存器 SPI_GCTL 中的 MODE 位为‘0’（从模式），SPIEN 位为‘1’（SPI 功能使能），并配置 SPI 工作所需的 GPIO 功能引脚。
5. 配置寄存器 SPI_GCTL 中 TXEN、RXEN 位为‘1’，打开发送、接收的许可（发送时需提前写入数据到寄存器 SPI_TXREG），SPI 将在从模式下接收 MOSI 引脚数据，并从 MISO 引脚输出数据。

注意：必须配置主、从设备的时序模式和数据帧收发顺序为一致，以保证数据能正常传输。

18.3.3.2 数据发送过程

写数据到发送数据寄存器 SPI_TXREG 后，整个数据会一起传输到发送缓冲。

当从设备接收到 SCK 引脚上的时钟信号，也同时接收到 MOSI 引脚传来的第一个数据位；从设备利用 SCK 的变化边沿，把发送数据逐位发送到 MISO 引脚上。发送数据的过程符合数据/时钟的相关时序（由 CPOL、CPHA/CPHASEL 位来决定）。

但在高速传输时（配置 SPI_CCTL.TXEDGE=1），数据将不再按照输入的 SCK 时钟边沿而变化，而会提前以内部 PCLK 的时钟边沿把数据送到 MISO 引脚上（此提前量不会早于前一位数据接收采样的 SCK 时钟边沿）。

当数据第一位被发送时，硬件会置位 SPI_INTSTAT.TX_INTF 标志，软件可利用此标志来写 TXREG 以实现数据的连续发送（配置 SPI_INTEN.TX_IEN 位为‘1’来产生 CPU 中断）。注意：从机时钟信号由主机提供，因此，连续传输的前提必须是主机能提供连续不断的时钟。

18.3.3.3 数据接收过程

从设备接收到 MOSI 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也同时会置位 SPI_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_RXREG，就能从接收缓冲中获取该数据。

- 软件配置 SPI_INTEN.RX_IEN 位为‘1’来打开中断使能，利用 CPU 中断获取接收数据。

18.3.4 SPI 主模式

SPI 作为主设备时，输出串行时钟到 SCK 引脚上，供从设备使用。

18.3.4.1 配置步骤

1. 配置 SPI_SPBRG 寄存器，定义串行时钟波特率。
2. 配置 SPI_CCTL 寄存器中 CPOL、CPHA/CPHASEL 位，确定时序模式。
3. 配置 SPI_CCTL.SPILEN 来定义 8 或 7 位数据帧格式；配置 SPI_GCTL.DW8_32 为‘1’，可配置 SPI_EXTCTL 寄存器来定义成任意帧格式（SPILEN 需固定为‘1’）。
4. 配置 SPI_CCTL.LSBFE 来确定数据收发的顺序（LSB 或 MSB 位优先）。
5. 如果只接收而不发送数据，可配置 SPI_RXDNR 寄存器来定义需要接收的字节数（当接收到指定个数字节后，SCK 时钟输出会结束并保持在 CPOL 位配置的状态上）。
6. 配置寄存器 SPI_GCTL 中的 MODE 位为‘1’（主模式）、SPIEN 位为‘1’（SPI 功能使能），并配置 SPI 工作所需的 GPIO 功能引脚。
7. 配置寄存器 SPI_GCTL 中 TXEN、RXEN 位为‘1’，打开发送、接收的许可（发送时打开 TXEN 后写入数据到寄存器 SPI_TXREG），SPI 将在主模式下输出时钟 SCK 和同步数据 MOSI 到引脚上，并从 MISO 引脚上采样输入数据；NSS 是主设备可选的输出功能。

注意：必须配置主、从设备的时序模式和数据帧收发顺序为一致，以保证数据能正常传输。

18.3.4.2 数据发送过程

配置 TXEN 位为‘1’后，写数据到发送数据寄存器 TXREG，此数据将传输到发送缓冲，主设备开始发送。主设备按照预先配置好的波特率串行输出 SCK 时钟和 MOSI 数据到引脚上，此过程符合数据/时钟的相关时序（由 CPOL、CPHA/CPHASEL 位决定）；而且 LSBFE 位决定了数据串行传输顺序。

当数据第一位被发送时，硬件会置位 SPI_INTSTAT.TX_INTF 标志，软件利用此标志来写 TXREG 以实现数据的连续发送（配置 SPI_INTEN.TX_IEN 位为‘1’来产生 CPU 中断）。

18.3.4.3 数据接收过程

主设备接收到 MISO 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也会同时置位 SPI_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_RXREG，就能从接收缓冲中获取该数据。
- 软件配置 SPI_INTEN.RX_IEN 位为‘1’可打开中断使能，利用 CPU 中断获取接收数据。
- 只接收时，接收完 RXDNR 定义的字节个数后，硬件将置位 SPI_INTSTAT.RXMATCH_INTF

标志，同时主设备不再发送时钟信号，SCK 输出将保持在 CPOL 位配置的状态上（固定高或 低电平）。

18.3.5 波特率设置

SCK 引脚输出的时钟频率符合波特率配置，它由内部时钟 PCLK 按照 SPI_SPBRG 寄存器的配置 值分频得到。寄存器 SPBREG 控制一个 16 位计数器的计数周期。

按照期望的波特率和 Fpclk（APB 模块 PCLK 时钟频率），使用下表公式可计算出给寄存器 SPBRG 的配置值（下表中的 X），X 在 2~65535 范围之内。

表 18-1 波特率公式

模式	公式
SPI 模式	波特率 = Fpclk/X

18.3.6 中断

18.3.6.1 状态标志

为了软件操作的方便，应用程序可以通过 4 个当前状态标志和 7 个中断状态标志来监控 SPI 总线 的状态。

当前状态标志是只读，由硬件自动置位和清除。 中断状态标志会在事件发生时置位，可在中断使能时 产生 CPU 中断请求，并由软件清除。

SPI 内部分别有一个 8 字节的发送缓冲和接收缓冲，CPU 可根据 SPI_GCTL 寄存器中 DW8_32 位 的配置，每次读写 1 个或 4 个字节。根据 DW8_32 的配置，发送和接收缓冲分别有 1 个字节或者 1 个 有效数据的状态标志。

注意：配置 SPI_GCTL.DW8_32=1，收发缓冲至多有两个有效数据；此时，当帧长配置为 8bit 及 以下时一个有效数据为 1 个字节；帧长配置在 9~16bit 范围时一个有效数据为 2 个字节；帧长配置在 17~24bit 范围时一个有效数据为 3 个字节；帧长配置在 25~32bit 范围时一个有效数据为 4 个字节。

表 18-2 SPI 状态

分类	状态标志	缓冲和信号状态
中断状态	TX_INTF	发送缓冲为空，根据 DW8_32 配置能完成一次发送数据寄存器 TXREG 的写操作
	RX_INTF	根据 DW8_32 设置，至少有一个有效数据的数据， 能完成一次接收数据寄存器 RXREG 的读操作
	UNDERRUN_INTF	发送缓冲空且重复发送
	RXOERR_INTF	接收缓冲非空且被覆盖
	RXMATCH_INTF	非空，指定个数的最后 1 个数据传到接收缓冲中（主模式下有效）
	RXFULL_INTF	接收缓冲满，不能再接收新的数据
	TXEPT_INTF	发送缓冲和发送移位寄存器都为空
当前状态	RXAVL_4BYTE	接收缓冲有超过 4 字节有效数据
	TXFULL	发送缓冲满

分类	状态标志	缓冲和信号状态
	TXEPT	发送缓冲和发送移位寄存器都为空
	RXAVL	接收缓冲非空

18.3.7 DMA 传输

SPI 可以利用 DMA 来搬运数据，包含读取接收数据和写入发送数据。DMA 请求及应答机制，能监控收发缓冲的空满状态，并提高对 SPI 收发数据寄存器 RXREG、TXREG 的读写速率，从而加快 SPI 通信速度。

配置 SPI_GCTL.DMAMODE 位为‘1’，来实现 SPI 模块与 DMA 之间的信号交互。当发送缓冲有空闲空间，即请求 DMA 写入 TXREG；当接收缓冲有可读的有效数据，即请求 DMA 读取 RXREG。

- 发送时：当配置 DW8_32 位为‘0’，发送缓冲有大于等于 1 个空闲数据（1 字节）的空间，即发送缓冲未满时就发起 DMA 请求；当配置 DW8_32 位为‘1’，发送缓冲有 1 个有效数据的空闲空间时，会发起 DMA 请求。每次请求只进行一次 DMA 传输，且传输数据字节数由 DW8_32 位决定。
- 接收时：当配置 DW8_32 位为‘0’，接收缓冲有大于等于 1 个有效数据（1 字节）时就发起 DMA 请求；当配置 DW8_32 位为‘1’，接收缓冲有 1 个有效数据后才发起 DMA 请求。每次请求只进行一次 DMA 传输。且传输数据字节数由 DW8_32 位决定。
- 注意：DW8_32=1 时，1 个有效数据的字节数由帧长位数决定（可参考 18.3.6.1 状态标志）。

18.4 寄存器

表 18-3 SPI 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SPI_TXREG	发送数据寄存器	0x00000000
0x04	SPI_RXREG	接收数据寄存器	0x00000000
0x08	SPI_CSTAT	当前状态寄存器	0x00000001
0x0C	SPI_INTSTAT	中断状态寄存器	0x00000000
0x10	SPI_INTEN	中断使能寄存器	0x00000000
0x14	SPI_INTCLR	中断清除寄存器	0x00000000
0x18	SPI_GCTL	全局控制寄存器	0x00000004
0x1C	SPI_CCTL	通用控制寄存器	0x00000008
0x20	SPI_SPBRG	波特率发生器	0x00000002
0x24	SPI_RXDNR	接收数据个数寄存器	0x00000001
0x28	SPI_NSSR	从机片选寄存器	0x000000FF
0x2C	SPI_EXTCTL	数据控制寄存器	0x00000008

18.4.1 SPI_TXREG 发送数据寄存器

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

Bit	Field	Description
31: 0	TXREG	发送数据寄存器 (Transmit data register) 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，TXREG[31: 0]都有效

18.4.2 SPI_RXREG 接收数据寄存器

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

Bit	Field	Description
31: 0	RXREG	接收数据寄存器 (Receive data register) 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，RXREG[31: 0]都有效 注：该寄存器可读不可写。

18.4.3 SPI_CSTAT 当前状态寄存器

偏移地址：0x08 复位值：

0x0000 0001

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

Res.	RXFADDR	Res.	TXFADDR	RXAVL_4BYTE	TXFULL	RXAVL	TXEPT
	r		r	r	r	r	r

Bit	Field	Description
31: 11	Reserved	始终读为 0
10: 8	RXFADDR	当前接收缓冲中有效字节个数
7	Reserved	始终读为 0
6: 4	TXFADDR	当前发送缓冲中有效字节个数
3	RXAVL_4BYTE	接收缓冲中有效数据达到 4 个字节标志 (Receive available 4 byte data message) 0: 接收缓冲中数据小于 4 个字节 1: 接收缓冲中有超过 4 个字节
2	TXFULL	发送缓冲满标志位 (Transmitter FIFO full status bit) 0: 发送缓冲未满 1: 发送缓冲满
1	RXAVL	接收有效数据标志位 (Receive available byte data message) 当接收缓冲中接收到一个字节数据时置位该位。 0: 接收缓冲空 1: 接收缓冲非空 注: 该位只读, 由硬件自动置位和清除。
0	TXEPT	发送端空标志位 (Transmitter empty bit) 0: 发送缓冲或发送移位寄存器不为空 1: 发送缓冲和发送移位寄存器都为空 注: 该位只读, 由硬件自动置位和清除。

18.4.4 SPI_INTSTAT 中断状态寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TXEPT_INTF	RXFULL_INTF	RXMAT_CH_INTF	RXOERR_INTF	UNDER_RUN_INTF	RX_INTF	TX_INTF
									r	r	r	r	r	r	r

Bit	Field	Description
31:7	Reserved	始终读为 0。
6	TXEPT_INTF	发送端空中断标志位 (Transmitter empty interrupt flag bit) 硬件自动置位, 写 INTCLR 寄存器 TXEPT_ICLR 位清除。 1: 发送端缓冲器和 TX 移位寄存器为空 0: 发送端不为空 注意: 该位是中断状态信号, TXEPT 是状态信号。
5	RXFULL_INTF	接收端缓冲器满中断标志位 (RX FIFO full interrupt flag bit) 硬件自动置位, 写 INTCLR 寄存器 RXFULL_ICLR 位清除。 1: RX 缓冲器满 0: RX 缓冲器未满

Bit	Field	Description
4	RXMATCH_INTF	接收指定字节数中断标志位 (Receive data match the RXDNR number, the receive process will be completed and generate the interrupt) 硬件自动置位, 写 INTCLR 寄存器 RXMATCH_ICLR 位清除。 1: 接收了 RXDNR 寄存器指定的字节数 0: 未完成 RXDNR 寄存器指定的字节数
3	RXOERR_INTF	接收端溢出错误中断标志位 (Receive overrun error interrupt flag bit) 硬件自动置位, 写 INTCLR 寄存器 RXOERR_ICLR 位清除。 1: 溢出错误 0: 没有溢出错误
2	UNDERRUN_INTF	SPI 从机模式下溢标志位 (SPI underrun interrupt flag bit) 硬件自动置位, 写 INTCLR 寄存器 UNDERRUN_ICLR 位清除。 1: 下溢错误 0: 没有下溢错误
1	RX_INTF	接收端数据有效中断标志位 (Receive data available interrupt flag bit) 硬件自动置位, 写 INTCLR 寄存器 RX_ICLR 位清除。 当接收端缓冲器接收了一个完整字节数据。 1: 接收端缓冲器有有效字节数据 0: 接收端缓冲器空
0	TX_INTF	发送缓冲器有效中断标志位 (发送了一个字节的数据) (Transmit FIFO available interrupt flag bit) 硬件自动置位, 发送缓冲器不为空自动清零。 1: 发送端缓冲器有效 0: 发送端缓冲器无效

18.4.5 SPI_INTEN 中断使能寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TXEPT_IEN	RXFULL_IEN	RXMATCH_IEN	RXOERR_IEN	UNDERRUN_IEN	RX_IEN	TX_IEN
									rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 7	Reserved	保留, 始终读为 0
6	TXEPT_IEN	发送端空中断使能位 (Transmit empty interrupt enable bit) 0: 中断禁止 1: 中断使能
5	RXFULL_IEN	接收缓冲满中断使能位 (Receive FIFO full interrupt enable bit) 0: 中断禁止 1: 中断使能
4	RXMATCH_IEN	接收指定字节数中断使能位 (Receive data complete interrupt enable bit) 0: 中断禁止 1: 中断使能

Bit	Field	Description
3	RXOERR_IEN	接收端溢出错误中断使能位 (Overrun error interrupt enable bit) 0: 中断禁止 1: 中断使能
2	UNDERRUN_IEN	SPI 从机模式下溢中断使能位 (SPI 从机模式) (Transmitterunderrun interrupt enable bit (SPI slave mode only)) 0: 中断禁止 1: 中断使能
1	RX_IEN	接收端数据中断使能位 (Receive FIFO interrupt enable bit) 0: 中断禁止 1: 中断使能
0	TX_IEN	发送缓冲空中断使能位 (Transmit FIFO empty interrupt enable bit) 0: 中断禁止 1: 中断使能

18.4.6 SPI_INTCLR 中断清除寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TXEPT_ICLR	RXFULL_ICLR	RXMATCH_ICLR	RXOERR_ICLR	UNDERRUN_ICLR	RX_ICLR	TX_ICLR
									w	w	w	w	w	w	w

Bit	Field	Description
31: 7	Reserved	保留, 始终读为 0
6	TXEPT_ICLR	发送缓冲空中断清除位 (Transmitter empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
5	RXFULL_ICLR	接收缓冲满中断清除位 (Receiver buffer full interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
4	RXMATCH_ICLR	接收指定字节数中断清除位 (Receive completed interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
3	RXOERR_ICLR	接收端溢出错误中断清除位 (Overrun error interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
2	UNDERRUN_ICLR	SPI 从机模式下溢中断清除位 (SPI 从机模式) (Transmitter underrun interrupt clear bit (SPI slave mode only)) 0: 写 0 无意义 1: 写 1 清除中断
1	RX_ICLR	接收端数据中断清除位 (Receive interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
0	TX_ICLR	发送缓冲空中断清除位 (Transmitter FIFO empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断

18.4.7 SPI_GCTL 全局控制寄存器

偏移地址：0x18 复位值：
0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														PAD_SEL	
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD_SEL			NSSTO G	DW8_32	NSS	DMAMO DE	Reserved				RXEN	TXEN	MODE	INTEN	SPIEN
rw			rw	rw	rw	rw					rw	rw	rw	rw	rw

Bit	Field	Description
31: 18	Reserved	保留，始终读为 0

Bit	Field	Description
17: 13	PAD_SEL	PAD_SEL SCL MOSI NSS MISO 0 PAD0 PAD1 PAD2 PAD3 1 PAD0 PAD1 PAD3 PAD2 2 PAD0 PAD2 PAD1 PAD3 3 PAD0 PAD2 PAD3 PAD1 4 PAD0 PAD3 PAD1 PAD2 5 PAD0 PAD3 PAD2 PAD1 6 PAD1 PAD0 PAD2 PAD3 7 PAD1 PAD0 PAD3 PAD2 8 PAD1 PAD2 PAD0 PAD3 9 PAD1 PAD2 PAD3 PAD0 10 PAD1 PAD3 PAD0 PAD2 11 PAD1 PAD3 PAD2 PAD0 12 PAD2 PAD0 PAD1 PAD3 13 PAD2 PAD0 PAD3 PAD1 14 PAD2 PAD1 PAD0 PAD3 15 PAD2 PAD1 PAD3 PAD0 16 PAD2 PAD3 PAD0 PAD1 17 PAD2 PAD3 PAD1 PAD0 18 PAD3 PAD0 PAD1 PAD2 19 PAD3 PAD0 PAD2 PAD1 20 PAD3 PAD1 PAD0 PAD2 21 PAD3 PAD1 PAD2 PAD0 22 PAD3 PAD2 PAD0 PAD1 23 PAD3 PAD2 PAD1 PAD0
		PAD0, PAD1, PAD2, PAD3 分别对应引脚分布中的 SCK、MOSI、NSS、MISO 引脚。信号 SCL、MOSI、NSS、MISO 和 PAD 的映射变换如下：
12	NSSTOG	从设备选择信号自动翻转 (Slave select toggle) 1: NSS 信号在传输完每个数据后自动翻转 0: NSS 信号不翻转 注： 该位只在主模式下有效
11	DW8_32	发送和接收数据寄存器有效数据选择 (Valid byte or double word data select signal) 0: 只有低 8 位有效 1: 32 位数据都有效 注：必须 用指定数据格式访问。
10	NSS	硬件或软件控制主模式下的 NSS 输出 (NSS select signal that from software or hardware) 0: 由 NSSR 寄存器值控制 1: 进行数据传输时硬件自动控制
9	DMAMODE	DMA 方式 (DMA Mode selection bit) 0: 正常模式 1: 开启 DMA 模式
8: 5	Reserved	预留，必须保持复位值。

Bit	Field	Description
4	RXEN	接收使能位 (Receive enable bit) 0: 接收禁止。同时可以清空 RX 缓冲 1: 接收使能 注: 当 SPI 只工作在主机接收模式时, TXEN 必须设置为 0。
3	TXEN	发送使能位 (Transmit enable bit) 0: 发送禁止。同时可以清空 TX 缓冲 1: 发送使能 注: 当在主机模式下发送和接收同时发生。
2	MODE	主机模式位 (Master mode bit) 0: 从机模式 (串行时钟来自外部主机) 1: 主机模式 (由内部 BRG 产生串行时钟)
1	INTEN	SPI 中断使能位 (SPI interrupt enable bit) 1: 使能 SPI 中断 0: 禁止 SPI 中断
0	SPIEN	SPI 选择位 (SPI select bit) 0: SPI 禁止 (复位状态) 1: SPI 使能

18.4.8 SPI_CCTL 通用控制寄存器

偏移地址: 0x1C 复位值:

0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								CPHAS		TXEDG	RXEDG				
								EI		E	E	SPILEN	LSBFE	CPOL	CPHA
								rw		rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:7	Reserved	始终读为 0。
6	CPHASEL	CPHA 极性取反选择 (CPHA polarity invert select) 1 : CPHA 设置取反。 CPHA 为 1 时，第一个数据位采样从第二个时钟边沿开始。 CPHA 为 0 时，第一个数据位采样从第一个时钟边沿开始。 0 : CPHA 设置保持不变。
5	TXEDGE	发送数据相位调整位 (从模式)(Transmit data edge select) 1 : 发送数据立即发送到数据总线 可用于高速模式时 (SPBRG = 4)。 0 : 发送数据在一个有效时钟边沿后发送到数据总线可用于低速模式时 (SPBRG > 4)。
4	RXEDGE	接收数据采样时钟沿选择位 (主模式)(Receive data edge select) 1 : 在传输数据位的尾时钟沿采样数据 (用于高速模式) 0 : 在传输数据位的中间采样数据
3	SPILEN	SPI 数据宽度位 (SPI character length bit) 该位在 DW8_32 置位后 (DW8_32=0) 配置后起作用。 1 : 8 位数据 (缺省) 0 : 7 位数据
2	LSBFE	LSBFE: LSB 在前使能位 (LSI first enable bit) 1 : 数据传输或接收最低位在前 0 : 数据传输或接收最高位在前
1	CPOL	时钟极性标志位 (Clock polarity select bit) 1 : 时钟在空闲状态为高电平 (两次传输之间) 0 : 时钟在空闲状态为低电平 (两次传输之间)

Bit	Field	Description
0	CPHA	时钟相位选择位 (Clock phase select bit) 1: 第一个数据位采样从第一个时钟边沿开始 0: 第一个数据位采样从第二个时钟边沿开始

18.4.9 SPI_SPBRG 波特率发生器

偏移地址: 0x20 复位值:

0x0000 0002

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

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	SPBRG	SPI 波特率控制寄存器用于产生波特率 (SPI baud rate control register for baud rate) 波特率公式: $\text{波特率} = \text{Fpclk} / \text{SPBRG}$ (Fpclk 是 APB 时钟频率) 注: 不能对该寄存器写值为 0、1。

18.4.10 SPI_RXDNR 接收数据个数寄存器

偏移地址: 0x24 复位值:

0x0000 0001

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

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	RXDNR	该寄存器用于存储下次接收过程需要接收字节的个数 (The register is used to hold a count of to be received bytes in next receive process) 注: 该寄存器的值仅在 SPI 为主机接收模式下有效; 缺省值是 1。该寄存器值通过软件来改变, 且不能对该寄存器写值为 0。

18.4.11 SPI_NSSR 从机片选寄存器

偏移地址: 0x28

复位值：0x0000 00FF

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值。
0	NSS	主模式下片选输出信号。低有效，从模式下该位无效（Chip select output signal in Master mode）。0：从器件被选中（允许从器件与主模式建立通信） 1：从器件未选中

18.4.12 SPI_EXTCTL 数据控制寄存器

偏移地址：0x2C 复位值：

0x0000 0008

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

Bit	Field	Description
31: 5	Reserved	保留，始终读为 0
4: 0	EXTLEN	控制 SPI 数据帧长度的选择 0 0000: 32 bit 0 0001: 1 bit 0 0010: 2 bit 0 0011: 3 bit 1 1100: 28 bit 1 1101: 29 bit 1 1110: 30 bit 1 1111: 31 bit 注：仅当 SPI_GCTL 寄存器 DW8_32 位为'0'时有效，且 SPI_CCTL 寄存器的 LSBFE 位必须配置为'1'， SPILEN 位也必须配置为'1'。

19 I2C 内部集成电路接口

19.1 简介

微控制器通过 I2C 总线接口实现芯片间的串行互联。所有 I2C 总线特定的序列、协议仲裁和时序，都可以通过 I2C 提供的多主功能来控制。

I2C 总线是一种两线串行接口，串行数据线（SDA）和串行时钟（SCL）在连接到总线的器件间传递信息。每个器件都通过一个唯一的地址进行识别，且都可以作为发送或接收器。此外，器件在执行数据传输时也可以被看作是主器件或从器件。主器件是在总线上发起数据传输，并产生允许该传输的时钟信号的器件。此时，任何被寻址的器件都被认为是从器件。

I2C 有两种速率模式可供选择：标准模式（数据传输速率最大为 100Kbps）、快速模式（数据传输速率最大为 400Kbps）。

19.2 主要特征

- I2C 总线协议转换器/并行总线
- 半双工同步操作
- 支持主从模式
- 支持 7 位和 10 位地址格式
- 支持起始（START）、停止（STOP）、重新起始（RESTART）以及应答（ACK）信号的生成和检测
- 支持标准模式（最大 100Kbps）、快速模式（最大 400Kbps）
- 分别有 2 字节的发送和接收缓冲
- 支持过滤毛刺功能
- 支持 DMA 操作
- 支持中断和查询操作
- 支持多个从地址（详细见 I2C_SLVMASK 寄存器描述）

19.3 功能描述

19.3.1 功能框图

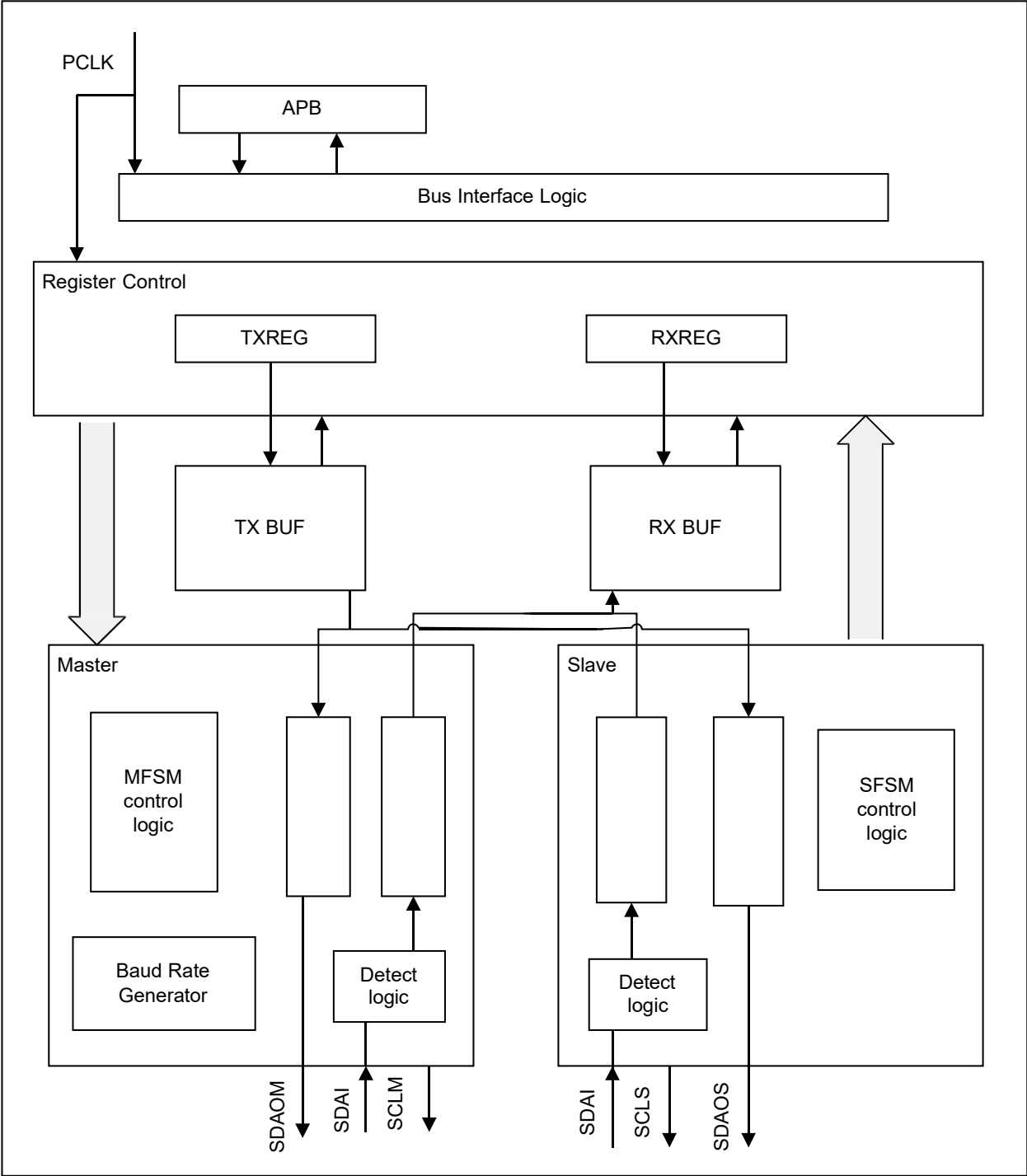


图 19-1 I2C 功能框图

19.3.2 信号描述

表 19-1 引脚定义

引脚名	属性	描述
I2C_SCL	I/O	I2C 时钟
I2C_SDA	I/O	I2C 数据

注：使用时引脚均需配置为开漏模式，配置方法请参考 GPIO 章节。

19.3.3 I2C 协议

19.3.3.1 起始和停止条件

总线处于空闲状态时，SCL 和 SDA 同时被外部上拉电阻拉为高电平。主器件启动数据传输时，必须先产生起始条件。在 SCL 线为高电平时，SDA 线从高电平向低电平切换表示起始条件。主器件结束传输时要发送停止条件。在 SCL 线为高电平时，SDA 线由低电平向高电平切换表示停止条件。下图显示了起始和停止条件的时序。数据传输过程中，当 SCL 为 1 时，SDA 必须保持稳定。

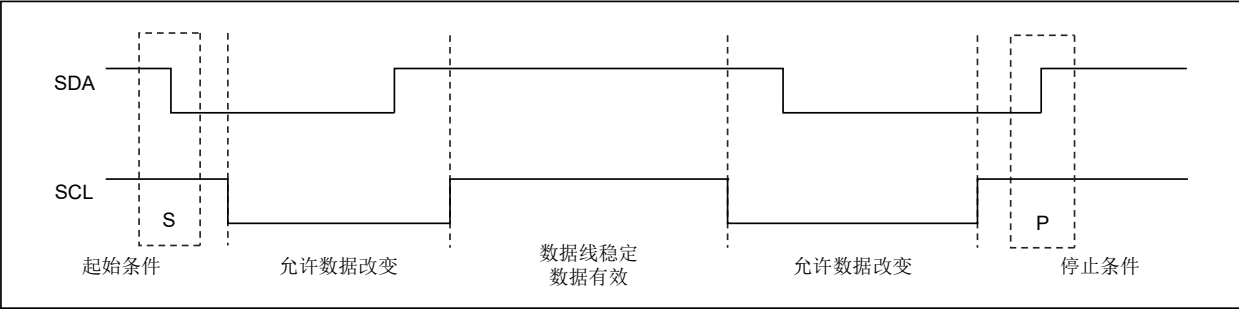


图 19-2 起始和停止条件

19.3.3.2 寻址协议

I2C 有两种地址格式：7 位地址格式和 10 位地址格式。

- 7 位地址格式 下图中，起始条件（S）后发送的第一个字节的前七位（b7：1）为从地址，最低位（b0）为数据

方向位。b0 为 0 表示主器件写数据到从器件，b0 为 1 表示主器件从从器件读数据。

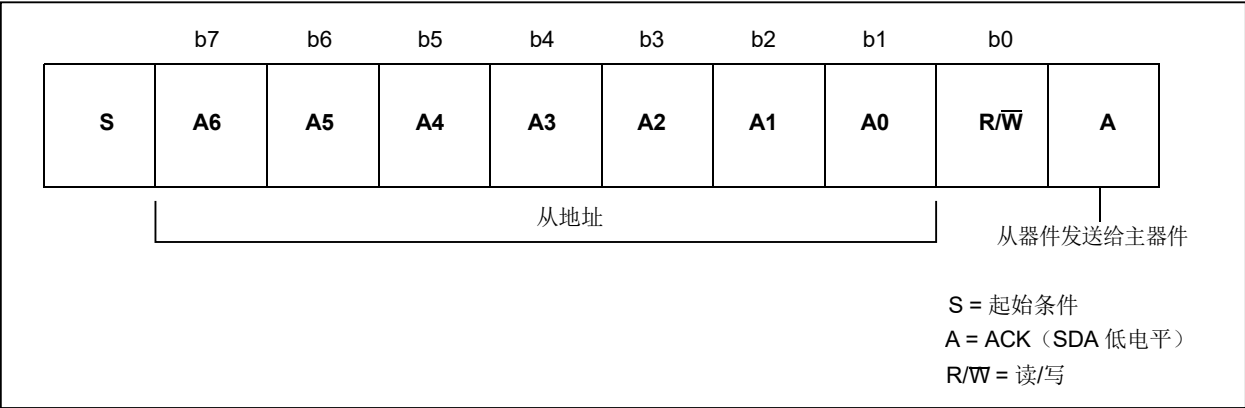


图 19-3 7 位地址格式

● 10 位地址格式

在 10 位地址格式中，需要发送 2 个字节来传输 10 位地址。第一个字节的描述如下：前五位（b7：3）用于通知从器件传输为 10 位地址格式；后两位（b2：1）为从地址的 b9：8；最低位（b0）为数据方向位（R/W）。第二个字节为 10 位地址的低八位。

具体如下图所示：

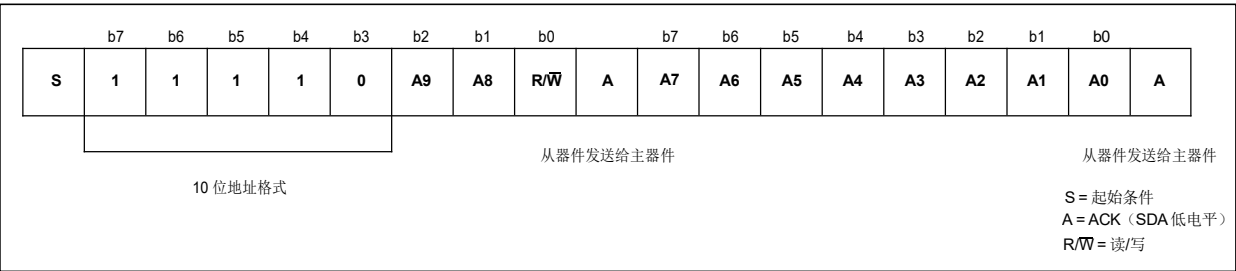


图 19-4 10 位地址格式

下表定义了 I2C 首字节的特殊用途和保留地址。

表 19-2 I2C 首字节

从地址	R/W 位	描述
0000 000	0	广播呼叫地址。I2C 将数据放入接收缓冲，并产生广播呼叫中断
0000 000	1	起始字节
0000 001	x	CBUS 地址。I2C 接口忽略该访问
0000 010	x	保留
0000 011	x	保留
0000 1xx	x	保留
1111 1xx	x	保留
1111 0xx	x	10 位从器件寻址

19.3.3.3 发送和接收协议

主器件可以发起数据传输，作为主发送器或主接收器，向总线发送数据或从总线接收数据。从器件响应主器件的请求，充当从发送器或从接收器。

- 主发送和从接收 所有数据都以字节格式传输，对每次传输的字节数没有限制。主器件发送完地址和 R/W 位或者一

个字节的的数据到从器件后，从接收器必须产生一个响应信号（**ACK**）。从接收器不能产生响应信号（**ACK**）时，主器件将会产生一个停止条件中止传输。从器件不能响应时，必须释放 **SDA** 为高电平，以便主器件产生停止条件。

当主发送器如下图所示传输数据时，从接收器在接收到的每个字节后产生一个 **ACK** 来响应主发送器。

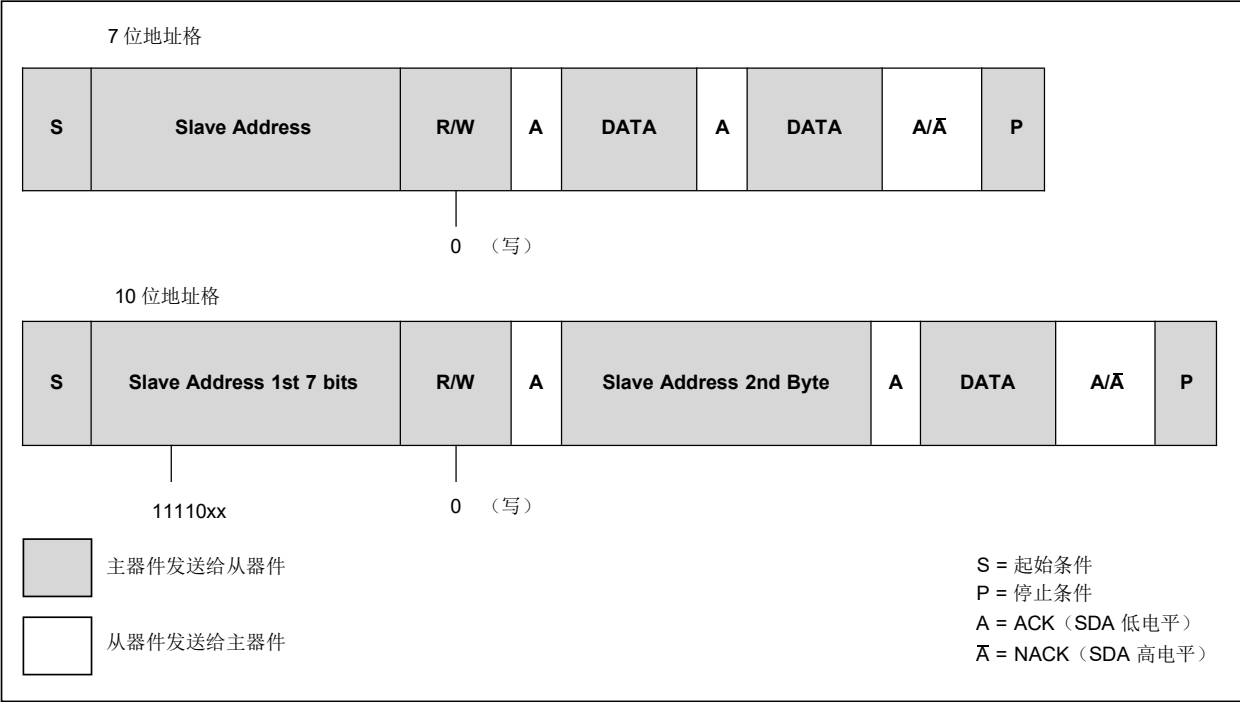


图 19-5 主发送协议

● 主接收和从发送

当主器件如下图所示接收数据时，主器件在每次接收到一个字节数据后响应从发送器，最后一个字节除外。通过这种方式，主接收器能够通知从发送器是否为最后一个字节。从发送器在检测到 **NACK** 时必须释放 **SDA**，以便主器件产生停止条件。

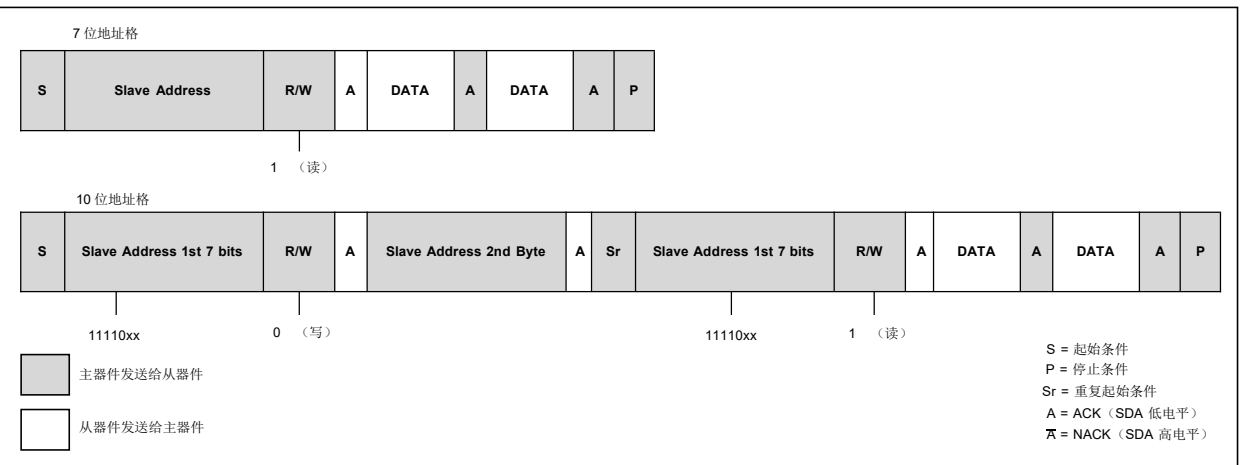


图 19-6 主接收协议

主器件不想因产生停止条件而释放总线时，可以产生一个重复起始条件。重复起始条件与起始条件相同，但重复起始条件在 **ACK** 后产生。在主模式下，**I2C** 接口可以使用不同的传输方向与相同的从器

图 19-11 主接收-TX FIFO 为空或产生 STOP

下图描述了 I2C 模块工作在主发送模式下，发生 RESTART (I2C_CR.REPEN 为 1) 时的时序。I2C_CR.REPEN 为 0 时，则先生成 STOP 再生成 START，以此代替 RESTART。

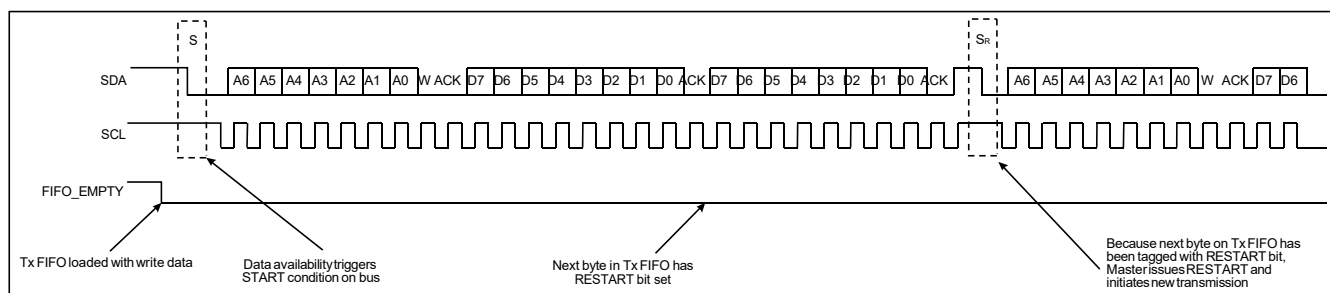


图 19-12 主发送-产生 RESTART

下图描述了 I2C 模块工作在主接收模式下，发生 RESTART 时的时序。

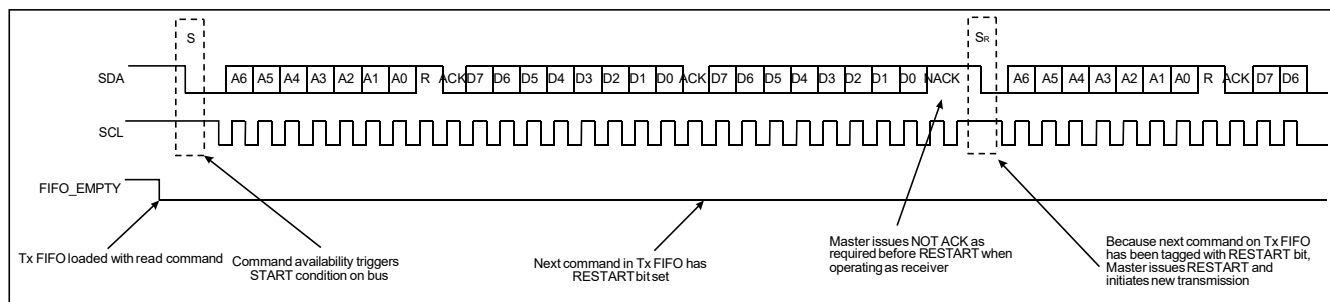


图 19-13 主接收-产生 RESTART

19.3.3.5 仲裁

如果总线上有多个主器件试图通过同时生成 START 条件来控制总线，那么就会有一个仲裁过程（多个主器件同时尝试控制总线，但只允许其中一个控制总线并使报文不被破坏的过程）。一旦其中一个主器件已经控制了总线，那么直到该主器件发送停止条件并将总线释放为空闲状态后，其他主器件才能控制总线。

当 SCL 线为高电平时，仲裁在 SDA 线发生。如果两个或多个主器件尝试发送信息到总线，在其他主器件都产生 0 的情况下，首先产生 1 的主器件将丢失仲裁。丢失仲裁的主器件可以继续产生时钟脉冲直到字节传输结束。如果每个主器件都尝试寻址相同的器件，仲裁会继续在数据阶段进行。

检测到丢失仲裁后，I2C 接口会停止产生 SCL 信号。

下图显示了两个主器件仲裁的总线时序。

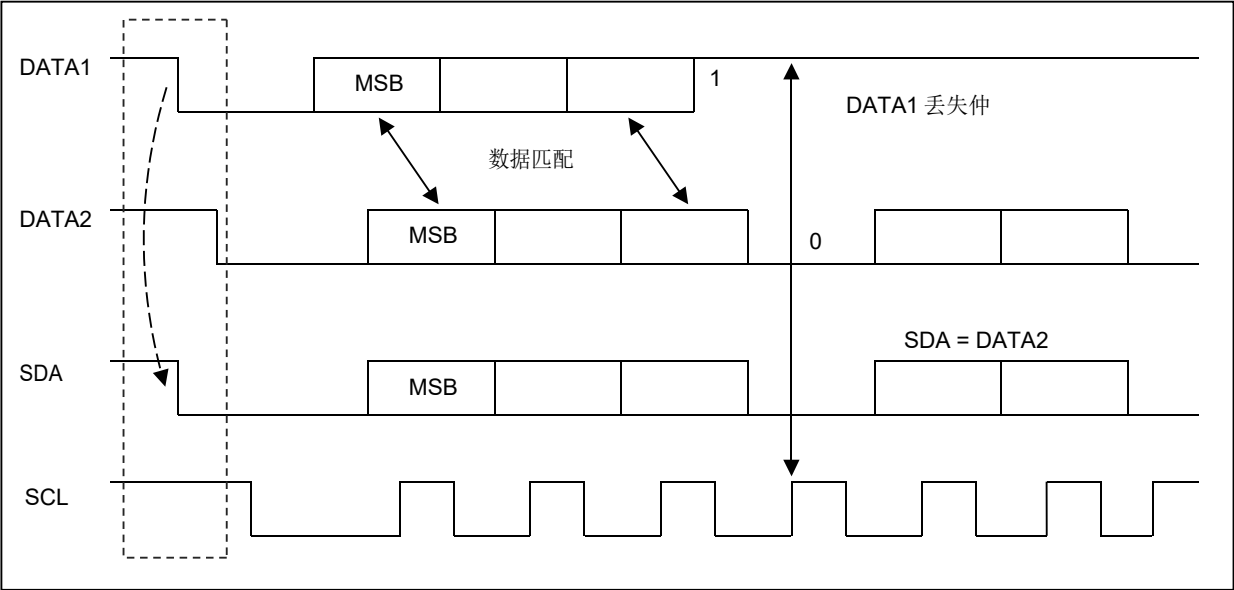


图 19-14 两个主器件仲裁

19.3.3.6 时钟同步

当两个或多个主器件试图同时在总线上传输信息时，必须仲裁和同步 SCL 时钟。所有主器件都各自产生时钟来传输消息。数据只在时钟的高电平有效。时钟同步通过 SCL 信号的线“与”连接进行。当主器件把 SCL 时钟转换为 0 时，主器件会计算 SCL 低电平的时间，在下一个时钟周期开始时把 SCL 转换为 1。但如果另一个主器件把 SCL 保持为 0，那么这个主器件会进入等待状态直到 SCL 变为 1。

之后，所有主器件计算各自的高电平时间，最短高电平时间的主器件把 SCL 转换为 0。接着，所有主器件计算各自的低电平时间，最长低电平时间的主器件强制其他主器件进入等待状态。由此产生一个同步后的 SCL 时钟，如下图所示。

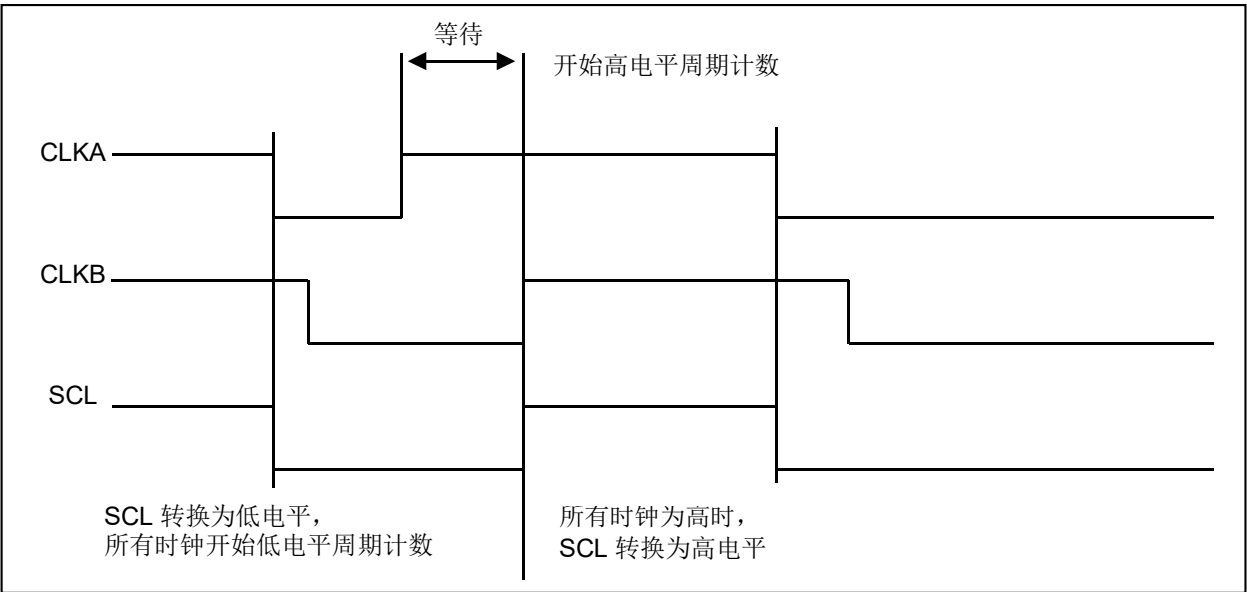


图 19-15 时钟同步（示意图）

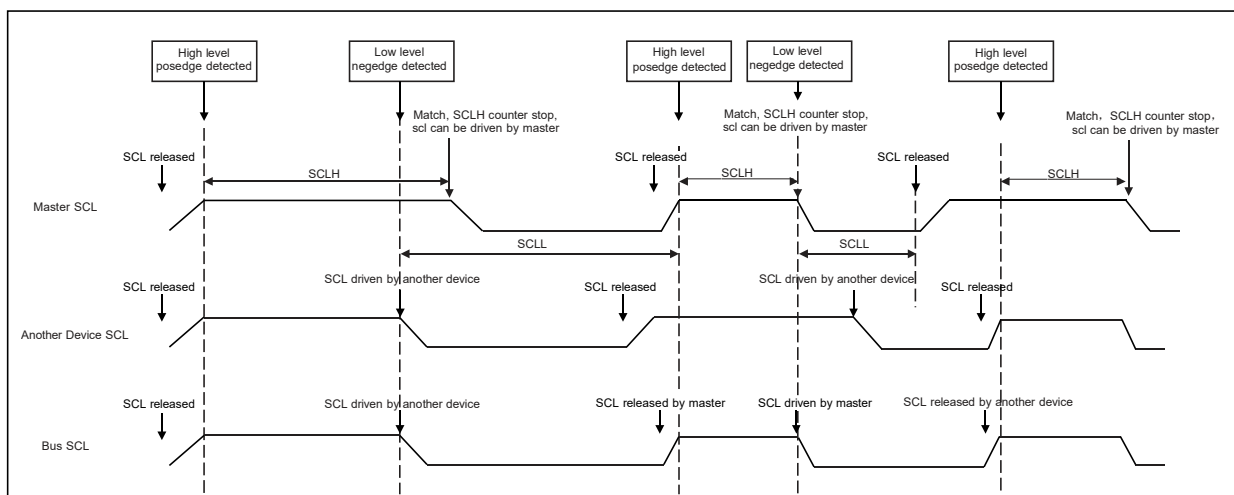


图 19-16 时钟同步（时序图）

19.3.3.7 SCL 配置

I2C 的 SCL 可参考如下配置：

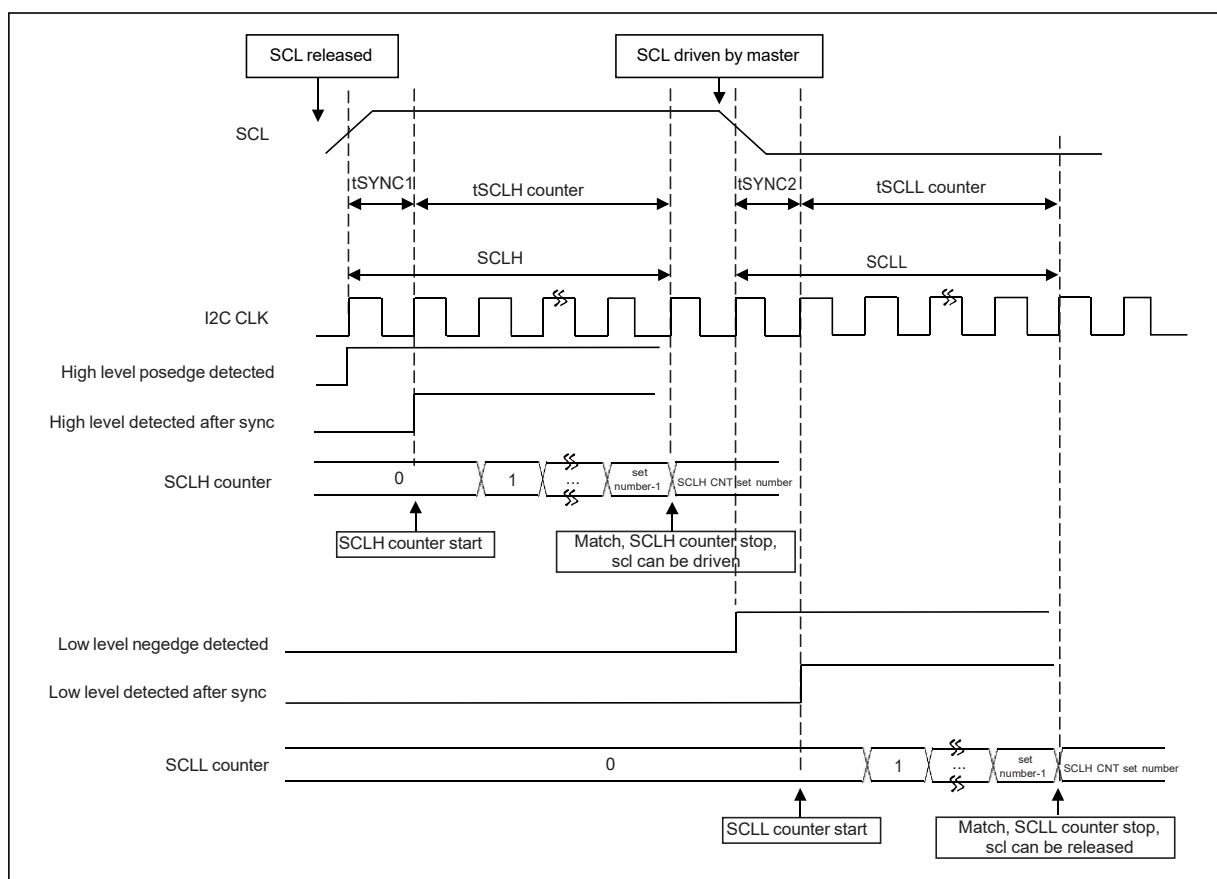


图 19-17 SCL 生成时序

- 标准模式：

$$\text{SCLH} = (\text{SSHR} + 12) \times \text{I2C CLK} + \text{tSYNC1}$$

$$\text{SCLL} = (\text{SSLR} + 1) \times \text{I2C CLK} + \text{tSYNC2}$$

注：tSYNC1 为 0~1 个 I2C CLK，tSYNC2 为 0~1 个 I2C CLK

- 快速模式：

$$SCLH = (FSHR + 12) \times I2C\ CLK + tSYNC1$$

$$SCLL = (FSLR + 1) \times I2C\ CLK + tSYNC2$$

注：tSYNC1 为 2~3 个 I2C CLK，tSYNC2 为 2~3 个 I2C CLK

19.3.4 工作模式

I2C 接口可以以下述 4 种方式之一运行：

- 从发送模式
- 从接收模式
- 主发送模式
- 主接收模式

注：I2C 接口模块只能工作在主模式或从模式，不能同时工作在两种模式下。因此需要确保寄存器 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）不能分别设置为 0 和 1（或者分别设置为 1 和 0）。

19.3.4.1 从模式

- 初始化配置

1) 写 0 到 I2C_ENR.ENABLE（位 0）以禁止 I2C

2) 通过配置 I2C_SAR 寄存器指定从地址（该地址为 I2C 接口所响应的地址）

3) 通过配置 I2C_CR.SLAVE10（位 3）指定地址格式（7 位或 10 位地址）；写 0 到 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）

4) 置位 I2C_ENR.ENABLE（位 0），使能 I2C 接口模块

- 从发送的单字节操作

I2C 接口被其他 I2C 主器件寻址并请求数据时，I2C 接口工作在从发送模式，工作流程如下：

1) 其他 I2C 主器件发起 I2C 传输，发送地址与 I2C_SAR 寄存器的从地址匹配

2) I2C 接口响应发送的地址，识别传输方向以表明其工作在从发送模式

3) I2C 接口产生 RD_REQ 中断（I2C_RAWISR 位 5），并将 SCL 线拉低；总线一直处于等待状态直到软件响应。若 RD_REQ 中断被屏蔽（I2C_IMR 位 5 为 0），建议 CPU 定期查询 I2C_RAWISR 寄存器：

- 读到 I2C_RAWISR.RD_REQ（位 5）为 1 等效于产生了 RD_REQ 中断
- 软件必须采取行动以满足 I2C 传输的要求
- 时间间隔通常在 10 个 SCL 时钟周期左右（例如，400kbps 的时间间隔是 25us）

4) 如果在接收到读请求前 TX FIFO 中仍有数据, I2C 接口会产生 TX_ABRT 中断 (I2C_RAWISR 位 6), 以清空 TX FIFO 中的数据 (I2C_CR.SLV_TX_ABRT_DIS 为 0 时) 因此, 在试图写入 TX FIFO 前, 软件有必要通过读取 I2C_TX_ABRT 寄存器来清除 TX_ABRT 中断。若 TX_ABRT 中断被屏蔽 (I2C_IMR 位 6 为 0), 建议 CPU 定期查询 I2C_RAWISR 寄存器。读到 I2C_RAWISR.TX_ABRT (位 6) 为 1 等效于产生了 TX_ABRT 中断。

5) 软件写数据到 I2C_DR 寄存器 (其中位 8 设置为 0)

6) 软件必须先清除 I2C_RAWISR.RD_REQ (位 5) 和 I2C_RAWISR.TX_ABRT (位 6) 中断

7) I2C 接口释放 SCL, 并发送数据字节

8) 主器件发送重复起始条件控制总线, 或发送停止条件释放总线

- 从接收的单字节操作

其他主器件寻址 I2C 接口并发送数据时, I2C 接口工作在从接收模式, 工作流程如下:

1) 其他 I2C 主器件发起 I2C 传输, 发送地址与 I2C_SAR 寄存器的从地址匹配

2) I2C 接口响应发送的地址, 识别传输方向以表明其工作在从接收模式

3) I2C 接口接收主器件发送的数据并将其存储在接收缓冲中

4) I2C 接口产生 RX_FULL 中断 (I2C_RAWISR 位 2)

若 RX_FULL 中断被屏蔽 (I2C_IMR 位 2 为 0), 建议 CPU 定期查询 I2C_SR 寄存器。读到 I2C_SR.RFNE (位 3) 为 1 等效于产生了 RX_FULL 中断

5) 软件通过读 I2C_DR 寄存器 (位 7:0) 来获得接收到的数据

6) 主器件发送重复起始条件控制总线, 或发送停止条件释放总线

- 程序流程图

下图为 I2C 接口作为从器件时的程序示例:

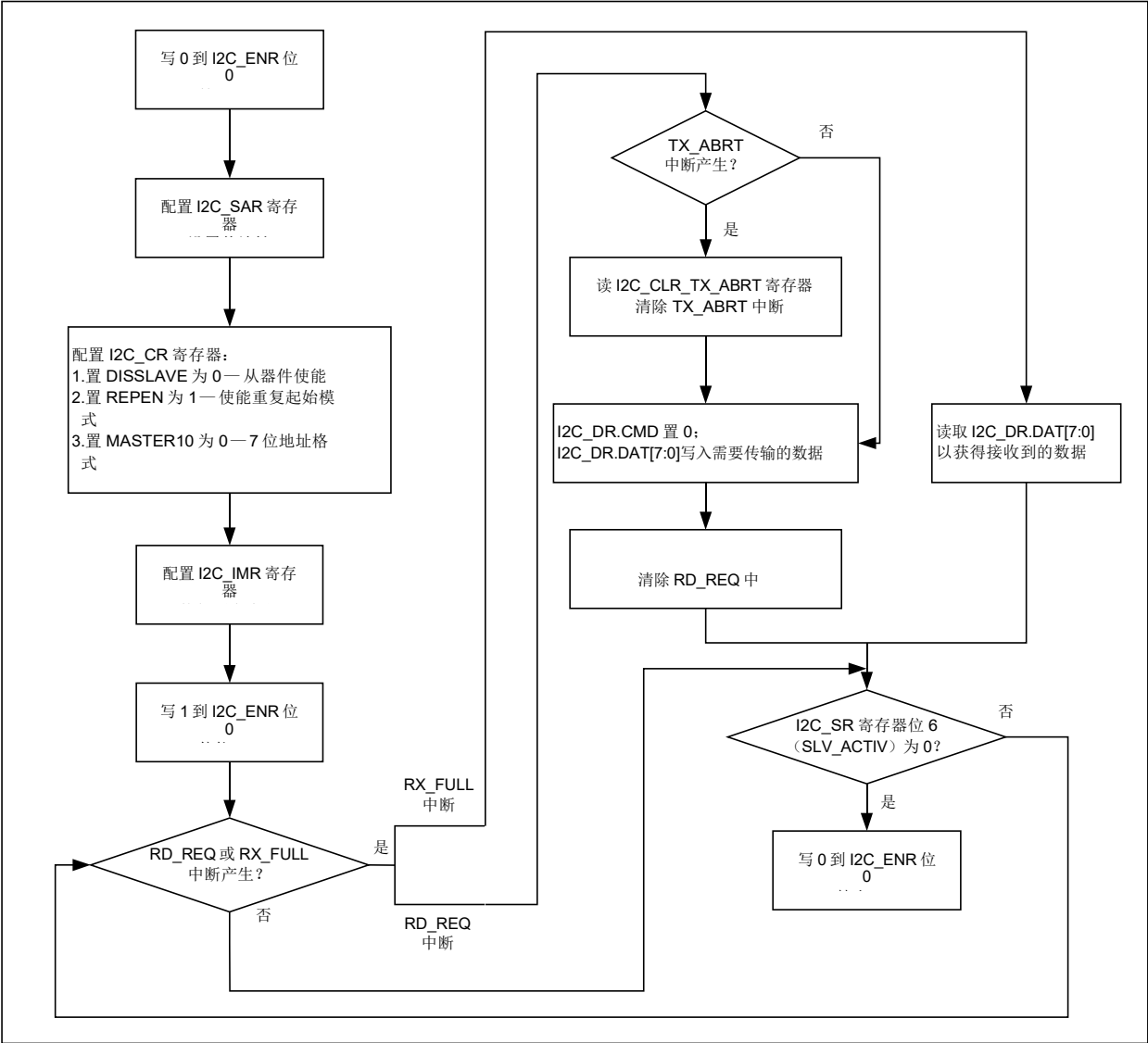


图 19-18 流程图（I2C 接口作为从器件）

● 从器件的块传输操作

标准 I2C 协议中，所有的数据处理都是单字节的处理，程序通过写一个字节到从器件的 TX FIFO 来响应主器件的读请求。当从器件（从发送）接收到主器件（主接收）的读请求（RD_REQ）时，至少应在从发送的 TX FIFO 中放置一个数据。I2C 接口模块可以处理 TX FIFO 中的多个数据，因此，后续的读请求无需再产生中断来获取数据。这极大地减少了因每次数据中断而导致的等待时间。

此模式仅适用于 I2C 接口作为从发送器的情况。如果接收到主器件发送的 ACK，从器件的 TX FIFO 中没有数据，I2C 接口将拉低总线的 SCL 线直到读请求中断（RD_REQ）产生并且 TX FIFO 的数据准备好后才释放 SCL 线。

如果 RX_REQ 中断被屏蔽（I2C_ISR 位 5 为 0），软件可以定期查询 I2C_RAWISR 寄存器。读到 I2C_RAWISR.RX_REQ（位 5）为 1 等效于产生了 RX_REQ 中断。

RD_REQ 中断由读请求产生，和中断一样必须在退出中断服务程序（ISR）时清除。在中断服务程序中（ISR）可以写一个或多个字节的数据到 TX FIFO。在将这些字节传输给主器件的过程中，如果

主器件响应了最后一个字节，则说明主器件正在要求更多的数据，那么从器件必须再次产生 RD_REQ 中断请求。

如果主器件要从 I2C 接口接收 n 个字节，但程序写到 TX FIFO 中的数据个数大于 n，那么从器件在发送完要求的 n 个字节数据后，将会清空 TX FIFO 并忽略额外的字节。

19.3.4.2 主模式

- 初始化配置

- 1) 写 0 到 I2C_ENR.ENALE (位 0) 以禁止 I2C

- 2) 通过配置 I2C_CR.SPEED (位 2:1) 指定 I2C 工作的速率模式 (标准模式、快速模式); 同时确保 I2C_CR.DISSLAIVE (位 6) 和 I2C_CR.MASTER (位 0) 都为 1

- 3) 将要寻址的 I2C 器件地址写入 I2C_TAR (该寄存器还可以配置为广播呼叫地址或起始字节命令)

- 4) 置位 I2C_ENR.ENALE (位 0), 使能 I2C 接口模块

- 5) 将传输方向和数据写入 I2C_DR, I2C 接口由此产生起始条件并发送地址字节

如果在使能 I2C 接口前配置了 I2C_DR 寄存器, 数据和命令都会丢失, 因为在 I2C 接口禁止的情况下缓冲是清空的。

- 主发送和主接收

I2C 接口支持读写的动态切换。发送数据时, 写数据到 I2C_DR 的低字节中, 配置 I2C_DR.CMD (位 8) 为 0 产生写操作。随后的读命令无需配置 I2C_DR 的低字节, 只需要配置 I2C_DR.CMD (位 8) 为 1 即可。若 TX FIFO 为空, I2C 模块拉低 SCL 直到下个命令写入到 TX FIFO 中。

- 程序流程图

下图为 I2C 接口作为主器件时的程序示例:

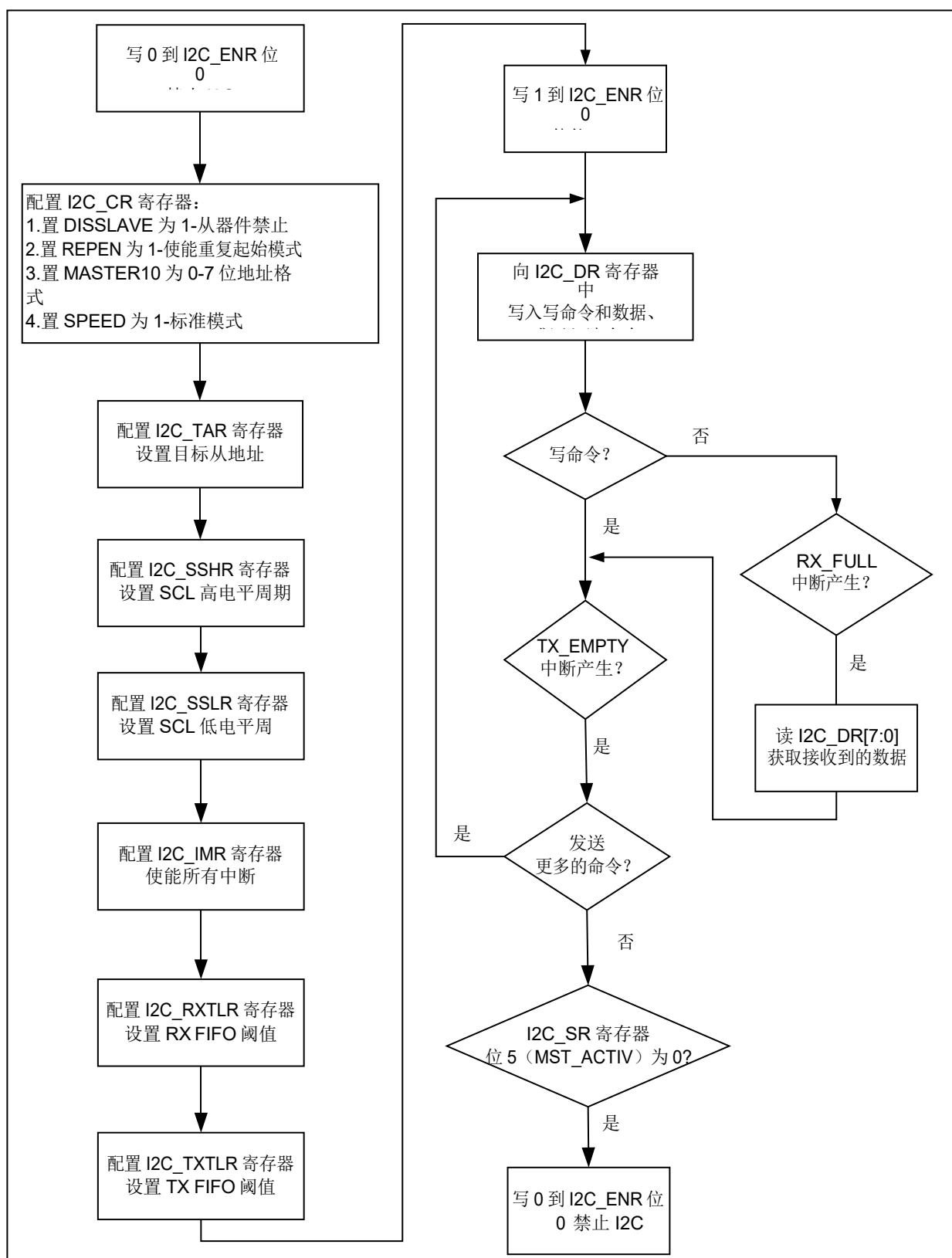


图 19-19 流程图（I2C 接口作为主器件）

19.3.4.3 中止传输

I2C_ENR.ABORT（位 1）允许软件在完成 TX FIFO 中的传输命令之前放弃 I2C 总线。作为 ABORT 请求的响应，I2C 模块发出停止条件到 I2C 总线，同时清空 TX FIFO。中止传输只允许在主模式下执行。

- 程序流程
- 1) 停止向 TX FIFO（I2C_DR）中写入新命令
- 2) 在 DMA 模式下工作时，通过置 I2C_DMA.TXEN（位 1）为 0 来禁止发送 DMA
- 3) 置 I2C_ENR.ABORT（位 1）为 1
- 4) 等待 TX_ABRT 中断

19.3.5 中断

下表列出了 I2C 的中断位及其设置和清除方式。部分位由硬件置位并由软件清除；另一部分位由 硬件置位和清除。

表 19-3 中断位的置位和清除

中断位	硬件置位/软件清除	硬件置位和清除
GEN_CALL	√	x
START_DET	√	x
STOP_DET	√	x
ACTIVITY	√	x
RX_DONE	√	x
TX_ABRT	√	x
RD_REQ	√	x
TX_EMPTY	x	√
TX_OVER	√	x
RX_FULL	x	√
RX_OVER	√	x
RX_UNDER	√	x

下图描述了中断寄存器中，中断位被硬件置位和软件清除的操作。

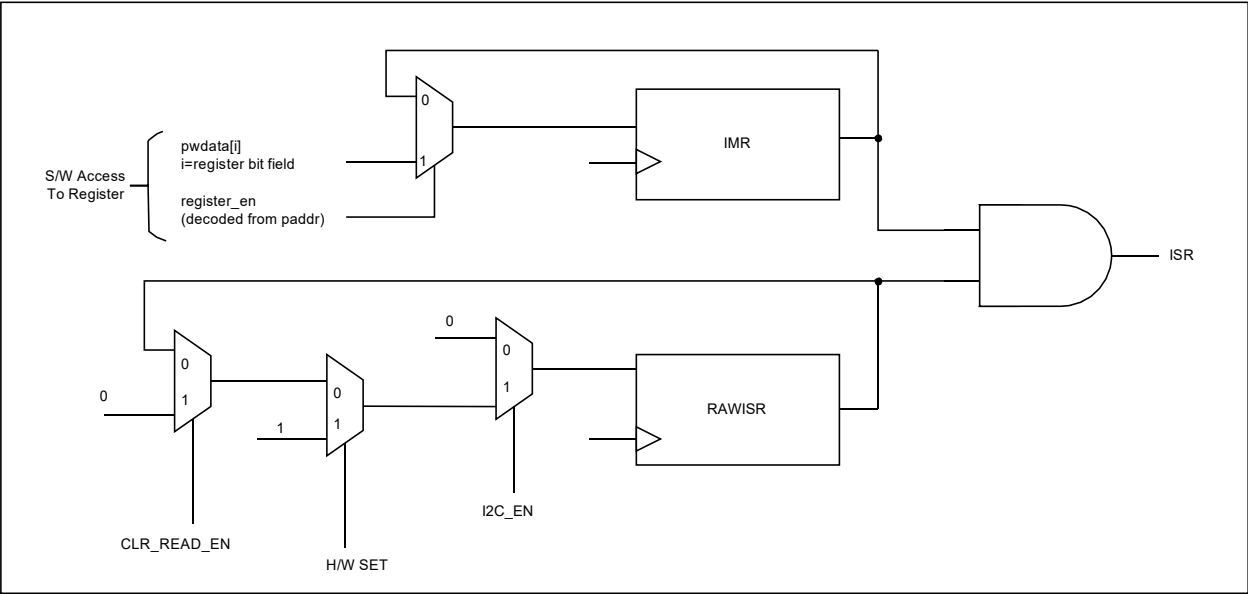


图 19-20 I2C 中断机制

19.3.6 DMA

I2C 接口支持使用 DMA 发送和接收数据。通过设置 I2C_DMA 寄存器的对应位可以单独开启 DMA 发送或 DMA 接收。发送时数据寄存器变空或接收时数据寄存器变满，则产生 DMA 请求。DMA 请求必须在当前字节传输结束之前被响应。

19.3.6.1 DMA 发送

通过设置 I2C_DMA.TXEN（位 1）可以激活 DMA 发送模式。为 I2C 分配好 DMA 通道后，发送数据时，DMA 控制器会将数据从预置的存储区装载进 I2C_DR 寄存器。

19.3.6.2 DMA 接收

通过设置 I2C_DMA.RXEN（位 0）可以激活 DMA 接收模式。为 I2C 分配好 DMA 通道后，每次接收到数据字节时，DMA 控制器会将数据从 I2C_DR 寄存器传送到预置的存储区。

19.4 寄存器

19.4.1 寄存器总览

表 19-4 I2C 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	I2C_CR	控制寄存器	0x0000007F
0x04	I2C_TAR	目标地址寄存器	0x00000055

Offset	Acronym	Register Name	Reset
0x08	I2C_SAR	从地址寄存器	0x00000055
0x10	I2C_DR	数据命令寄存器	0x00000000
0x14	I2C_SSHR	标准模式时钟高电平计数寄存器	0x00000190
0x18	I2C_SSLR	标准模时钟低电平计数寄存器	0x000001D6
0x1C	I2C_FSHR	快速模式时钟高电平计数寄存器	0x0000003C
0x20	I2C_FSLR	快速模时钟低电平计数寄存器	0x00000082
0x2C	I2C_ISR	中断状态寄存器	0x00000000
0x30	I2C_IMR	中断屏蔽寄存器	0x000008FF
0x34	I2C_RAWISR	RAW 中断寄存器	0x00000000
0x38	I2C_RXTLR	接收阈值	0x00000000
0x3C	I2C_TXTLR	发送阈值	0x00000000
0x40	I2C_ICR	组合和独立中断清除寄存器	0x00000000
0x44	I2C_RX_UNDER	清除 RX_UNDER 中断寄存器	0x00000000
0x48	I2C_RX_OVER	清除 RX_OVER 中断寄存器	0x00000000
0x4C	I2C_TX_OVER	清除 TX_OVER 中断寄存器	0x00000000
0x50	I2C_RD_REQ	清除 RD_REQ 中断寄存器	0x00000000
0x54	I2C_TX_ABRT	清除 TX_ABRT 中断寄存器	0x00000000
0x58	I2C_RX_DONE	清除 RX_DONE 中断寄存器	0x00000000
0x5C	I2C_ACTIV	清除 ACTIVITY 中断寄存器	0x00000000
0x60	I2C_STOP	清除 STOP_DET 中断寄存器	0x00000000
0x64	I2C_START	清除 START_DET 中断寄存器	0x00000000
0x68	I2C_GC	清除 GEN_CALL 中断寄存器	0x00000000
0x6C	I2C_ENR	使能寄存器	0x00000000
0x70	I2C_SR	状态寄存器	0x00000006
0x74	I2C_TXFLR	发送缓冲级别寄存器	0x00000000
0x78	I2C_RXFLR	接收缓冲级别寄存器	0x00000000
0x7C	I2C_HOLD	SDA 保持时间寄存器	0x00000001
0x88	I2C_DMA	DMA 控制寄存器	0x00000000
0x94	I2C_SETUP	SDA 建立时间寄存器	0x00000064
0x98	I2C_GCR	广播呼叫 ACK 寄存器	0x00000001
0xB0	I2C_SLVMASK	从地址掩码寄存器	0x000003FF
0xB4	I2C_SLVRCVADDR	从器件接收地址寄存器	0x00000000

19.4.2 I2C_CR 控制寄存器

偏移地址：0x00 复位值：

0x0000 007D

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.			PAD_SE L	Res.			EMPINT	STOPIN T	DISSLA VE	REPEN	MASTE R10	SLAVE1 0	SPEED		MASTE R
			rw				rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31: 13	Reserved	保留，必须保持复位值
12	PAD_SEL	PAD_MUX 交换 SCL、SDA 对应 PAD 0: PAD0 ⇌ SCL; PAD1 ⇌ SDA 1: PAD0 ⇌ SDA; PAD1 ⇌ SCL
11: 9	Reserved	保留，必须保持复位值
8	EMPINT	该位控制 TX_EMPTY 中断的产生，细节请参考 I2C_RAWISR 寄存器
7	STOPINT	从模式下，是否产生 STOP 中断 0: 无论地址是否匹配，都产生 STOP 中断 1: 仅在地址匹配时产生 STOP 中断 注：广播地址寻址时，如果该位置位，从器件不产生 STOP 中断。STOP 中断 仅在发送地址与从地址（I2C_SAR）匹配时产生。
6	DISSLAVE	该位控制是否禁用 I2C 从器件 0: 从器件使能 1: 从器件禁止
5	REPEN	I2C 作为主器件时是否发送 RESTART 条件 0: 禁止 1: 使能 RESTART 条件可以替换为先发送停止条件再发送起始条件。 当 RESTART 被禁止时，I2C 接口作为主器件不能执行以下功能： 发送起始字节 组合格式模式下改变传输方向 10 位地址格式的读操作 执行上述操作会置位 I2C_RAWISR.TX_ABRT（位 6）。
4	MASTER10	I2C 作为主器件时的地址格式 0: 7 位地址格式 1: 10 位地址格式
3	SLAVE10	I2C 作为从器件时，该位控制是否响应 7 位或 10 位地址 0: 7 位寻址。I2C 接口忽略 10 位寻址；对于 7 位寻址，仅比较 I2C_SAR 寄存器的低 7 位 1: 10 位寻址。I2C 仅响应 10 位寻址，接收地址与 I2C_SAR 的 10 位比较
2: 1	SPEED	该两位控制 I2C 接口工作在主模式时的速率 01: 标准模式（最大 100Kbps） 10: 快速模式（最大 400Kbps）
0	MASTER	该位控制是否使能主器件 0: 主器件禁止 1: 主器件使能

I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）的配置组合如下表所示：

表 19-5 DISSLAVE 和 MASTER 配置

DISSLAVE (I2C_CR[6])	MASTER (I2C_CR[0])	状态
0	0	从器件
0	1	禁止配置
1	0	禁止配置
1	1	主器件

19.4.3 I2C_TAR 目标地址寄存器

偏移地址：0x04 复位值：

0x0000 0055

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.				SPECIAL	GC	ADDR									
				rw	rw	rw									

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11	SPECIAL	该位指示软件执行的是否为特殊命令（广播呼叫或起始字节命令） 0：忽略 GC（位 10），正常使用 ADDR（位 9:0） 1：执行特殊 I2C 命令，细节请参考 GC 位的描述
10	GC	位 11 置位时，该位显示 I2C 执行的是广播呼叫还是起始字节命令 0：广播呼叫 发起广播呼叫后只能执行写操作。I2C 接口一直工作在广播呼叫模式下，直到 SPECIAL（位 11）被清零 1：起始字节命令
9: 0	ADDR	主模式时的目标地址 发起广播呼叫时，该字段被忽略。要产生开始字节命令，CPU 只需要对该字段写一次。

19.4.4 I2C_SAR 从地址寄存器

偏移地址：0x08 复位值：

0x0000 0055

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.						ADDR									
						rw									

Bit	Field	Description
31: 10	Reserved	保留，必须保持复位值
9: 0	ADDR	I2C 接口的从地址。对于 7 位地址格式，ADDR 只有[6: 0]有效。

19.4.5 I2C_DR 数据命令寄存器

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.	RESTA RT w	STOP w	CMD w	DAT rw
------	------------------	-----------	----------	-----------

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
10	RESTART	发送或接收字节前，是否产生 RESTART 0: 如果 REPEN 为 1，则仅在前一个命令改变传输方向时才产生 RESTART； 如果 REPEN 为 0，则先产生 STOP 再产生 START 1: 如果 REPEN 为 1，则在数据接收或发送（根据 CMD 的值）前产生 RESTART，无论前一个命令是否改变数据的传输方向；如果 REPEN 为 0，则先产生 STOP 再产生 START
9	STOP	发送或接收字节后，是否产生 STOP 0: 当前字节之后不产生 STOP，无论 TX FIFO 是否为空。如果 TX FIFO 不为空，主器件继续当前传输（根据 CMD 的值发送或接收数据）。如果 TX FIFO 为空，主器件将拉低 SCL 线，挂起总线直至 TX FIFO 收到新命令 1: 当前字节之后产生 STOP，无论 TX FIFO 是否为空。如果 TX FIFO 不为空，主机将立即通过发送 START 来尝试发起一次新的传输。
8	CMD	控制在主模式下执行读或写操作 0: 写 1: 读 向 TX FIFO 中输入命令时，该位用于区分读和写命令。从接收模式下，对该位的写操作被忽略；从发送模式下，写 0 表示发送 I2C_DR 寄存器的数据。
7: 0	DAT	I2C 总线待发送或接收到的数据

19.4.6 I2C_SSHR 标准模式时钟高电平计数寄存器

偏移地址：0x14 复位值：

0x0000 0190

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															
rw															

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	CNT	标准模式下 SCL 时钟高电平周期 注：该寄存器只可配置在 6 至 65525 之间，这是由于 I2C 接口使用了一个 16 位的计数器，该计数器值等于 CNT+10 时标志 I2C 总线处于空闲状态。

19.4.7 I2C_SSLR 标准模式时钟低电平计数寄存器

偏移地址：0x18 复位值：

0x0000 01D6

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

rw

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	CNT	标准模式下 SCL 时钟低电平周期，最小值为 8。

19.4.8 I2C_FSHR 快速模式时钟高电平计数寄存器

偏移地址：0x1C 复位值：

0x0000 003C

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

rw

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	CNT	快速模式下 SCL 时钟高电平周期，最小值为 6。

19.4.9 I2C_FSLR 快速模式时钟低电平计数寄存器

偏移地址：0x20 复位值：

0x0000 0082

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

rw

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	CNT	快速模式下 SCL 时钟低电平周期，最小值为 8。

19.4.10 I2C_ISR 中断状态寄存器

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.				GC	START	STOP	ACTIV	RX_DON	TX_ABR	RD_RE	TX_EM	TX_OV	RX_FUL	RX_OV	RX_UN
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11: 0	ISR	每位的具体描述可以参考 I2C_RAWISR 寄存器

19.4.11 I2C_IMR 中断屏蔽寄存器

偏移地址：0x30 复位值：

0x0000 08FF

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.				GC	START	STOP	ACTIV	RX_DON	TX_ABR	RD_RE	TX_EM	TX_OV	RX_FUL	RX_OV	RX_UN
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11: 0	IMR	每一位屏蔽与 I2C_ISR 寄存器对应的中断状态位

19.4.12 I2C_RAWISR RAW 中断寄存器

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
Res.				GC	START	STOP	ACTIV	RX_DON	TX_ABR	RD_RE	TX_EM	TX_OV	RX_FUL	RX_OV	RX_UN
				r	r	r	r	r	r	r	r	r	r	r	r

	r	r	r	r	r	r	r	r	r	r	r	r
--	---	---	---	---	---	---	---	---	---	---	---	---

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11	GC	广播呼叫（General call） 接收到广播呼叫地址并响应时置位。 禁止 I2C 接口或当 CPU 读 I2C_GC.GC（位 0）时清零。 I2C 将接收到的数据存储在接收缓冲中。
10	START	起始条件检测 无论 I2C 接口工作在主模式或从模式，一旦检测到 I2C 接口上发生了起始或重复起始条件即置位该位。
9	STOP	停止条件检测 该位状态取决于 I2C_CR.STOPINT（位 7）的状态： STOPINT = 0： 无论 I2C 接口工作在主模式或从模式，一旦检测到 I2C 接口上发生了停止条件时即置位该位。从模式下，无论从器件是否被寻址都会产生 STOP 中断。 STOPINT = 1： 主模式下（MASTER = 1），该位指示 I2C 接口是否发生停止条件； 从模式下（MASTER = 0），仅当从器件被寻址时产生 STOP 中断。
8	ACTIV	该位用于捕捉 I2C 模块的活动状态。置位后只能由以下四种方式清零： <ul style="list-style-type: none"> 禁止 I2C 接口 读 I2C_ACTIV 寄存器 读 I2C_ICR 寄存器 系统复位 即使 I2C 处于空闲状态，该位也仍然保持为高直到被清零。
7	RX_DONE	从发送结束（Transmit done） 当 I2C 作为从发送器时，如果主器件没有响应发送的字节，则该位置位。 该情况发生在传输的最后一个字节则表示传输结束。
6	TX_ABRT	发送中止（Transmit abort） 当 I2C 接口作为发送器，不能发送完缓冲中的数据时置位。 注意：发送中止会将 I2C 接口的接收和发送缓冲清空。发送缓冲会处于刷新状态直到读 I2C_TX_ABRT 寄存器。一旦该读操作执行后，发送缓冲就可以接收 APB 接口上的新数据。
5	RD_REQ	读请求（Read request） 当 I2C 作为从器件，其他主器件试图从 I2C 接口读取数据时置位。 I2C 接口会使总线保持在等待状态（SCL = 0）直到中断被处理。这表明 I2C 接口被其他主器件寻址成功且被要求发送数据。处理器必须响应该中断然后写数据到 I2C_DR 寄存器中。该位在处理器读 I2C_RD_REQ 寄存器后清零。
4	TX_EMPTY	发送缓冲空（Transmit buffer empty） 该位状态取决于 I2C_CR.EMPINT（位 8）的状态： EMPINT = 0，发送缓冲区数据个数小于等于阈值时置位； EMPINT = 1，发送缓冲区数据个数小于等于阈值且前一个发送地址/数据的内部移位寄存器发送结束时置位。 发送缓冲区数据个数大于阈值时由硬件自动清零。
3	TX_OVER	发送缓冲过载（Transmit buffer over） 发送缓冲满时处理器写入新数据导致溢出时置位。
2	RX_FULL	接收缓冲满（Receive buffer full） 接收缓冲区数据大于阈值时置位。 接收缓冲区数据小于等于阈值时由硬件清零。
1	RX_OVER	接收缓冲过载（Receive buffer over） 接收缓冲满时收到新数据时置位。此时 I2C 接口会响应，但新的数据会丢失。
0	RX_UNDER	接收缓冲欠载（Receive buffer under） RX FIFO 为空时处理器读 I2C_DR 寄存器则该位置位。

19.4.13 I2C_RXTLR 接收阈值

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
Res.								TL							
								rw							

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	TL	接收 FIFO 阈值（Receive FIFO threshold level） 控制 RX_FULL 中断触发。

19.4.14 I2C_TXTLR 发送阈值

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
Res.								TL							
								rw							

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	TL	发送 FIFO 阈值（Transmit FIFO threshold level） 控制 TX_EMPTY 中断触发。

19.4.15 I2C_ICR 组合和独立中断清除寄存器

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.	ICR
	r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	ICR	读该寄存器将会清除所有组合中断、独立中断。 该位不清除硬件可自动清除的中断，仅清除软件可清除中断。

19.4.16 I2C_RX_UNDER 清除 RX_UNDER 中断寄存器

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
Res.															RX_UN DER
															r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RX_UNDER	读该寄存器清零 RX_UNDER 中断（I2C_RAWISR[0]）

19.4.17 I2C_RX_OVER 清除 RX_OVER 中断寄存器

偏移地址：0x48 复位值：

0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RX_OVER	读该寄存器清零 RX_OVER 中断（I2C_RAWISR[1]）

19.4.18 I2C_TX_OVER 清除 TX_OVER 中断寄存器

偏移地址：0x4C 复位值：

0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	TX_OVER	读该寄存器清零 TX_OVER 中断（I2C_RAWISR[3]）

19.4.19 I2C_RD_REQ 清除 RD_REQ 中断寄存器

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.															RD_RE Q
															r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RD_REQ	读该寄存器清零 RD_REQ 中断（I2C_RAWISR[5]）

19.4.20 I2C_TX_ABRT 清除 TX_ABRT 中断寄存器

偏移地址：0x54 复位值：

0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	TX_ABRT	读该寄存器清零 TX_ABRT 中断（I2C_RAWISR[6]） 同时将 TX FIFO 从刷新/复位状态中释放，以便接收写入的数据。

19.4.21 I2C_RX_DONE 清除 RX_DONE 中断寄存器

偏移地址：0x58 复位值：

0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RX_DONE	读该寄存器清零 RX_DONE 中断（I2C_RAWISR[7]）

19.4.22 I2C_ACTIV 清除 ACTIVITY 中断寄存器

偏移地址：0x5C 复位值：

0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	ACTIV	I2C 不活动时，读该寄存器清零 ACTIV 中断（I2C_RAWISR[8]） 如果 I2C 仍然活动，那么 ACTIV 中断将继续置位。当 I2C 模块禁止或 I2C 总线不再活动时该位由硬件清零。可以通过读该寄存器得到 I2C_RAWISR.ACTIV（位 8）的状态。

19.4.23 I2C_STOP 清除 STOP_DET 中断寄存器

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.															STOP
															r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	STOP	读该寄存器清零 STOP 中断（I2C_RAWISR[9]）

19.4.24 I2C_START 清除 START_DET 中断寄存器

偏移地址：0x64 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.															START
															r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	START	读该寄存器清零 START 中断（I2C_RAWISR[10]）

19.4.25 I2C_GC 清除 GEN_CALL 中断寄存器

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.															GC
															r

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	GC	读该寄存器清零 GC 中断（I2C_RAWISR[11]）

19.4.26 I2C_ENR 使能寄存器

偏移地址：0x6C 复位值：

0x0000 0000

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

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1	ABORT	I2C 传输中止（I2C transfer abort） 0：中止没有发生或者已经结束 1：中止操作正在进行 I2C 模块作为主器件时，软件可以置位该位以中止 I2C 传输。一旦置位不能立即清除。置位后 I2C 模块控制逻辑会在完成当前传输后产生 STOP 条件并清空发送缓冲，中止操作之后产生 TX_ABRT 中断。 该 ABORT 位会在中止操作结束后自动清零。
0	ENABLE	I2C 模块使能 0：禁止 I2C 模块（发送和接收缓冲保持擦除状态） 1：使能 I2C 模块

19.4.27 I2C_SR 状态寄存器

偏移地址：0x70 复位值：

0x0000 0006

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.									SLV_ACTIV	MST_ACTIV	RFF	RFNE	TFE	TFNF	ACTIV
									r	r	r	r	r	r	r

Bit	Field	Description
31: 7	Reserved	保留，必须保持复位值

Bit	Field	Description
6	SLV_ACTIV	从状态机（Slave FSM）活动状态 0: 从状态机处于 IDLE 状态，I2C 从器件部分不活动 1: 从状态机不处于 IDLE 状态，I2C 从器件部分活动
5	MST_ACTIV	主状态机（Master FSM）活动状态 0: 主状态机处于 IDLE 状态，I2C 主器件部分不活动 1: 主状态机不处于 IDLE 状态，I2C 主器件部分活动
4	RFF	接收缓冲满 0: 接收缓冲未满 1: 接收缓冲满
3	RFNE	接收缓冲非空 0: 接收缓冲空 1: 接收缓冲非空
2	TFE	发送缓冲空 0: 发送缓冲非空 1: 发送缓冲空
1	TFNF	发送缓冲未满 0: 发送缓冲满 1: 发送缓冲未满
0	ACTIV	I2C 活动状态 MST_ACTIV 位与 SLV_ACTIV 位相或的结果。

19.4.28 I2C_TXFLR 发送缓冲级别寄存器

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

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1: 0	CNT	发送缓冲中有效数据的个数（0~2）

19.4.29 I2C_RXFLR 接收缓冲级别寄存器

偏移地址：0x78 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.													CNT		

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1: 0	CNT	接收缓冲中有效数据的个数（0~2）

19.4.30 I2C_HOLD SDA 保持时间寄存器

偏移地址：0x7C 复位值：

0x0000 0001

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

Bit	Field	Description
31: 24	Reserved	保留，必须保持复位值
23: 16	RX_HOLD	I2C 作为接收器时，配置 SDA 内部保持时间（SCL 为高电平时有效），单位为 APB 时钟周期。
15: 0	TX_HOLD	I2C 作为发送器时，配置 SDA 保持时间（SCL 从高电平转换为低电平后），单位为 APB 时钟周期。

19.4.31 I2C_DMA DMA 控制寄存器

偏移地址：0x88 复位值：

0x0000 0000

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

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1	TXEN	0: DMA 发送禁止 1: DMA 发送使能
0	RXEN	0: DMA 接收禁止 1: DMA 接收使能

19.4.32 I2C_SETUP SDA 建立时间寄存器

偏移地址：0x94 复位值：
0x0000 0064

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.								CNT							
								rw							

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	CNT	SDA 建立时间，最小值为 2。 如果所需延迟时间为 1000ns，当 APB 时钟频率为 10MHz 时，建议将该寄存器设为 11。

19.4.33 I2C_GCR 广播呼叫 ACK 寄存器

偏移地址：0x98 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.														GC	
														rw	

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	GC	广播呼叫 ACK 0: 接收到广播呼叫后不响应（NACK），也不产生中断 1: 接收到广播呼叫后响应（ACK）

19.4.34 I2C_SLVMASK 从地址掩码寄存器

偏移地址：0xB0 复位值：
0x0000 03FF

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.						MASK									
						rw									

Bit	Field	Description
31: 10	Reserved	保留，必须保持复位值
9: 0	MASK	从地址掩码 0: I2C_SAR 寄存器的相应位被掩码忽略，不需要比较 1: I2C_SAR 寄存器的相应位需要比较

19.4.35 I2C_SLVRCVADDR 从器件接收地址寄存器

偏移地址：0xB4 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.						ADDR									
						r									

Bit	Field	Description
31: 10	Reserved	保留，必须保持复位值
9: 0	ADDR	从器件实际接收到的地址

20 CSM 通用串行模块

20.1 简介

串行收发模块(CSM)主要用于串行数据的接收和发送，数据可由比较器输入，通过内部设置的波特率采样后转换为 32 位数据，可用 CPU 或者 DMA 保存在存储器。

20.2 功能框图

下图为 CSM 的功能框图，通过移位寄存器将输入的串行信号转变为 32 位寄存器数据，或者将 32 位发送数据寄存器移位作为串行输出。

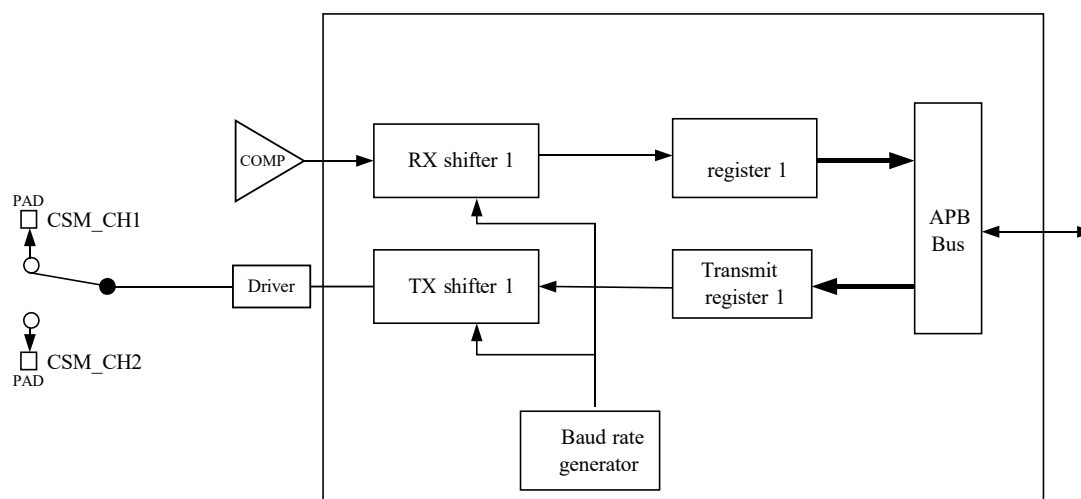


图 20-1 CSM 功能框图

CSM_CH1 和 CSM_CH2 可作为内部比较器输入信号，也可被分压电路 Driver 驱动选择其一进行输出。ADC 模块中的内部参考电压(Voltage Sensor)为 Driver 提供输出电压基准，CSM 模块控制 Driver 的使能和输出。如果配置 CSM 输出 1KHz 方波，由于 ADC 的内部参考电压恒为 1.2 V，那么在选择输出的 CSM_CH1 或 CSM_CH2 上输出频率为 1KHz、电压值为 1.2 V 方波；ADC 可以监测 CSM_CH1 和 CSM_CH2 的电压。

PD 应用示意图如下所示。

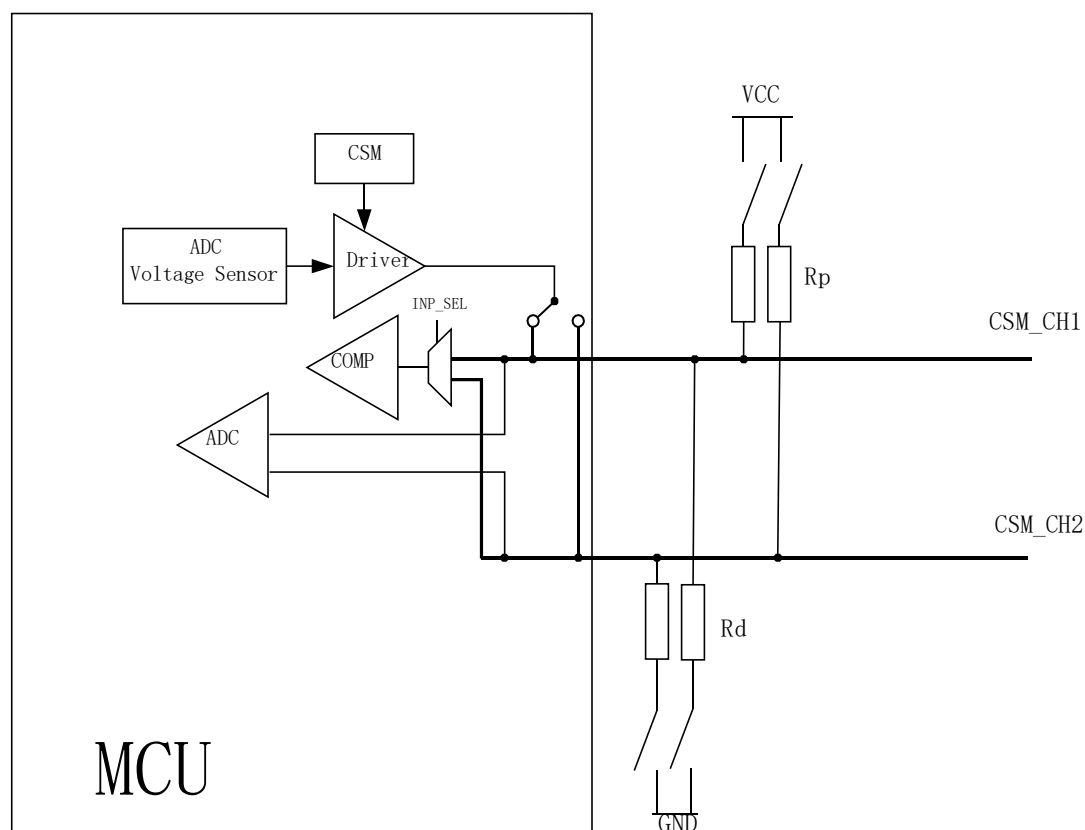


图 20-2 PD 应用示意图

20.3 主要特征

- 可以通过比较器输入
- 支持发送电压值为 1.2 V 的信号
- 只支持异步的接收和发送
- 支持 DMA 请求
- 16 位的可编程波特率生成器
- 可编程的空闲总线状态（低/高）

20.4 功能描述

20.4.1 外部端口

外部端口输入由比较器输入。输出信号电压为固定 1.2 V。

20.4.2 数据接收

接收状态下，最低有效位先从外部输入。每接收 32 位之后，数据会从移位寄存器搬到相应的接收数据寄存器中。如果接收中断 `CSM_IER.RX_IEN` 使能，则产生中断。如果 DMA 使能，则产生 DMA 请求信号。

串行数据接收时，通过 `STARTSEL` 位设置采样时刻：通道使能后一直采样或者检测到起始条件后开始采样；当选择检测起始条件开始采样时，起始位不算数据，从起始位之后，开始数据的采样。这就要求发送端在发送数据之前，要先发一个起始位，供接收端识别。

起始条件：在出现非空闲电平后开始。如果空闲电平是高电平，则出现低电平后开始采样。如果空闲电平是低电平，则出现高电平后开始采样。

结束条件：通过 `STOPSEL` 位配置结束条件；如果是固定长度的数据，则在接收设置的位数后结束；如果是不定长度的数据，则在接收设置的空闲电平数量后结束。数据接收结束后通道使能由硬件自动关闭。

20.4.3 数据发送

发送状态下，写入 32 位数据到发送数据寄存器，数据会搬到发送移位寄存器，最低有效位先输出。数据搬到移位寄存器后，如果发送缓冲器中断 `CSM_IER.TX_IEN` 使能，则产生中断；如果 DMA 使能，则产生 DMA 请求信号。当移位寄存器数据全部输出完毕后，如果发送完成中断 `CSM_IER.TXC_IEN` 使能，则会产生中断。

使能 `CSM_CR.EN` 后，写 `CSM_TDR` 后就开始发送，不会发送起始位；在发送完设置的位数后，发送结束。数据发送结束后通道使能由硬件自动关闭。

20.4.4 状态标志

为了软件操作的方便，应用程序可以通过中断状态标志来监控总线的状态。当前状态标志由硬件自动置位，写 1 清除相应的中断状态位。中断状态标志位在事件发生时置位，并在中断使能时产生 CPU 中断，由软件清除。

20.4.5 波特率设置

波特率是生成的采样时钟的频率，一般是 `PCLK` 的分频。`CSM_SPBRG` 寄存器控制 16 位计数器的计数周期。

提供期望的波特率和 f_{PCLK} ，参考如下公式：

$$\text{波特率} = f_{PCLK} / (X + 1)$$

其中 X 的值为写入 `CSM_SPBRG` 寄存器的值（2 ~ 65535）。

20.5 CSM 模拟输出

使能 ADC 模块中的内部参考电压(Voltage Sensor)为 Driver 提供电压基准，CSM 模块的输出经过 Driver 模块，增加驱动能力后由 CSM_CR.OCSEL 选择在 PA1 或 PA2 上输出，相关配置如下：

- 使能 ADC_ADCFG.VSEN，以提供 CSM 模块需要的电压基准
- 使能 CSM_CFG.VTXEN，使能 Driver 分压电路驱动
- 配置 CSM_CR.OCSEL 选择在 PA1 或 PA2 上输出
- 设置通道为发送模式并使能 CSM_CR. EN，然后写 CSM_TDR 寄存器，使 CSM 模块输出数据
- 内部将输出线连接到 Driver 模块，经 Driver 模块后在 PA1 或 PA2 引脚上输出数据

注意：引脚初始状态应为高阻态，即不能有任何驱动

注意：任意通道切换至发送模式时，写 CSM_TDR 寄存器前还必须保证提供 Driver 充足的建立时间（15us）

20.6 DMA

为了达到最大通信速度，需要及时往发送寄存器填数据，同样接收寄存器中的数据也必须及时读走以防止溢出。为了方便高速率的数据传输，实现了一种采用简单的请求/应答的 DMA 机制。

当 DMAEN 位使能时，CSM 模块可以发出 DMA 传输数据的请求。当发送移位寄存器为空时，会发出 DMA 请求，DMA 将存储器中数据搬至发送数据寄存器中；当接收缓冲器接收到一个完整字数据时，会发出 DMA 请求，DMA 将数据搬至存储器中。

20.7 寄存器

20.7.1 寄存器总览

表 20-1 CSM 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CSM_TDR1	CSM 发送数据寄存器 1	0x00000000
0x08	CSM_RDR1	CSM 接收数据寄存器 1	0x00000000
0x10	CSM_ISR	CSM 中断状态寄存器	0x00000000
0x14	CSM_IER	CSM 中断使能寄存器	0x00000000
0x18	CSM_CR1	CSM 控制寄存器 1	0x00000000
0x20	CSM_CFG	CSM 配置寄存器	0x00000000
0x24	CSM_SBPRG	CSM 波特率寄存器	0x00000002
0x28	CSM_BCNT	CSM 数据位长度寄存器	0x00000000

20.7.2 CSM_TDR1 发送数据寄存器 1

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
TDR1 [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
TDR1 [15: 0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	TDR1	发送数据寄存器 1(Transmit data register 1)

20.7.3 CSM_RDR1 接收数据寄存器 1

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
RDR1 [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
RDR1 [15: 0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	RDR1	接收数据寄存器 1(Receive data register 1)

20.7.4 CSM_ISR 中断状态寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STOP_I NTF	START_ INTF	TXC_IN TF	RX_ INTF	TX_ INTF

						r_w1c	r_w1c	r_w1c	r_w1c	r_w1c
--	--	--	--	--	--	-------	-------	-------	-------	-------

Bit	Field	Description
31:5	Reserved	保留，始终读为 0
4	STOP_INTF	停止中断标志位 当检测到停止的状态，硬件自动置位，写 1 清除。 1 = 停止接收有效数据 0 = 没有停止标志
3	START_INTF	开始中断标志位 当检测到起始的状态，硬件自动置位，写 1 清除。 1 = 开始接收有效数据 0 = 没有起始标志
2	TXC_INTF	发送移位寄存器完成中断标志位(Transmit complete interrupt flag bit) 当发送数据缓冲和移位寄存器都空时，硬件自动置位，写 1 清除。 1 = 移位寄存器数据发送完成 0 = 移位寄存器空或正在移位发送
1	RX_INTF	接收数据有效中断标志位(Receive data available interrupt flag bit) 当接收端缓冲器接收了一个完整数据时，硬件自动置位，写 1 清除。 1 = 接收端缓冲器有有效数据 0 = 接收端缓冲器空
0	TX_INTF	发送缓冲器有效中断标志位(Transmit FIFO available interrupt flag bit) 发送数据缓冲空时硬件自动置位，写 1 清除。 1 = 发送端缓冲器有效 0 = 发送端缓冲器无效 注：该标志位适用于大于 32bit 数据的发送

20.7.5 CSM_IER 中断使能寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STOP_IEN	START_IEN	TXC_IEN	RX_IEN	TX_IEN
											rw	rw	rw	rw	rw

Bit	Field	Description
31:5	Reserved	保留，始终读为 0
4	STOP_IEN	停止接收数据中断使能位 1 = 中断使能 0 = 禁止中断
3	START_IEN	开始接收数据中断使能位 1 = 中断使能 0 = 禁止中断
2	TXC_IEN	发送移位寄存器完成中断使能位(Transmit complete interrupt enable bit) 1 = 中断使能 0 = 禁止中断

Bit	Field	Description
1	RX_IEN	接收数据有效中断使能位(Receive data available interrupt enable bit) 1 = 中断使能 0 = 禁止中断
0	TX_IEN	发送缓冲器有效中断使能位(Transmit FIFO available interrupt enable bit) 1 = 中断使能 0 = 禁止中断

20.7.6 CSM_CR1 控制寄存器 1

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OCSEL	TXSEL1	DMAEN1	Res.	IDLEP1	EN1
										rW	rW	rW		rW	rW

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5	OCSEL	分压电路输出通道选择 1= 选择 PA[2] 作为模拟输出 0= 选择 PA[1] 作为模拟输出
4	TXSEL1	发送和接收数据选择控制 1 = 发送数据 0 = 接收数据
3	DMAEN1	DMA 使能控制 1 = 开启 0 = 关闭
2	Reserved	保留, 始终读为 0
1	IDLEP1	空闲状态电平 1 = 在空闲状态为高电平 0 = 在空闲状态为低电平
0	EN1	数字通道使能 该位用于使能通道内部的数字控制逻辑, 数据传输结束后自动清除 1 = 使能 0 = 关闭

20.7.7 CSM_CFG 配置寄存器

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
MAXBIT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												VTXEN	Res.	STOPSEL	STARTSEL
												rw	rw	rw	rw

Bit	Field	Description
31:16	MAXBIT[15:0]	收发数据的长度，以位（bit）为单位 STOPSEL=0 时表示采样到空闲电平的长度后结束 STOPSEL=1 时表示采样到数据的长度后结束
15:4	Reserved	保留，始终读为 0
3	VTXEN	发送分压电路使能位 1：使能 0：禁止
2	Reserved	保留，始终读为 0
1	STOPSEL	采样结束条件 1 = 采样到给定长度数据后结束 0 = 采样到给定长度的空闲电平后结束，MAXBIT=0 表示一直采样不结束
0	STARTSEL	起始条件检测 从空闲状态电平变为非空闲状态电平起始条件 1 = 不检测起始条件，使能后开始采样 0 = 检测起始条件后开始采样

20.7.8 CSM_SPBRG 波特率寄存器

偏移地址：0x24 复位值：

0x0000 0002

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

Bit	Field	Description
31:16	Reserved	保留，始终读为 0
15:0	SPBRG[15: 0]	波特率控制寄存器用于产生采样时钟（baud rate control register for simple data） 波特率公式： 波特率 = $f_{PCLK} / (SPBRG + 1)$ （ f_{PCLK} 是 APB 时钟频率）

20.7.9 CSM_BCNT 数据位长度寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31:16	Reserved	保留，始终读为 0
15:0	BCNT[15:0]	当前接收或者发送数据位长度，只读

21 ADC 模拟/数字转换

21.1 简介

ADC 是 12 位的逐次逼近型 (SAR) 模拟数字转换器, 可以将模拟信号转换成数字信号。

A/D 转换器支持多种工作模式: 单次转换和连续转换模式, 并且可以选择通道自动扫描及扫描方向。A/D 转换的启动方式有软件设定、外部引脚触发以及各个定时器启动。窗口比较器 (模拟看门狗) 允许应用程序检测输入电压是否超出了用户设定的高/低阈值值。ADC 的输入时钟不得超过 16MHz, 它是由 PCLK2 经分频产生。

21.2 主要特征

- 高达 1Msps 转换速率
- 支持任意通道转换
 - ◆ 单次转换模式: 在指定通道完成一次转换
 - ◆ 单周期扫描模式: 在所有指定通道按照通道设置完成一个周期转换
 - ◆ 连续扫描模式: 连续执行单周期扫描模式直到软件停止 A/D 转换。若需要转换期间修改通道, 用户不必停止转换, 可配置相应通道寄存器, 配置的新通道将在下一个扫描周期进行转换
- 可编程通道采样时间
- 最高 12 位可编程分辨率 SAR 多达 12 路外部输入通道和 1 路内部通道
- 支持 DMA 传输
- A/D 转换开始条件
 - ◆ 软件启动
 - ◆ 触发启动, 可配置触发延时
- 模拟看门狗功能。转换结果与指定的阈值区间进行比较, 当转换值超出设定的阈值区间时, 如果 ADC_ADCR.AWDIE 置位, 则产生中断

21.3 DMA

A/D 转换结果存储在数据寄存器 ADC_ADDATA 中, 当多个通道连续转换时可以使用 DMA 访问保存数据, 避免转换数据丢失。

DMA 使能开启, 当通道转换结束后将产生 DMA 请求, 将转换数据从 ADC_ADDATA 寄存器传输

到软件指定的目的地址。

每个通道有对应的数据寄存器 ADC_ADDRn，可以访问这些寄存器获取各通道的转换结果。

21.4系统框图

ADC 系统框图如下：

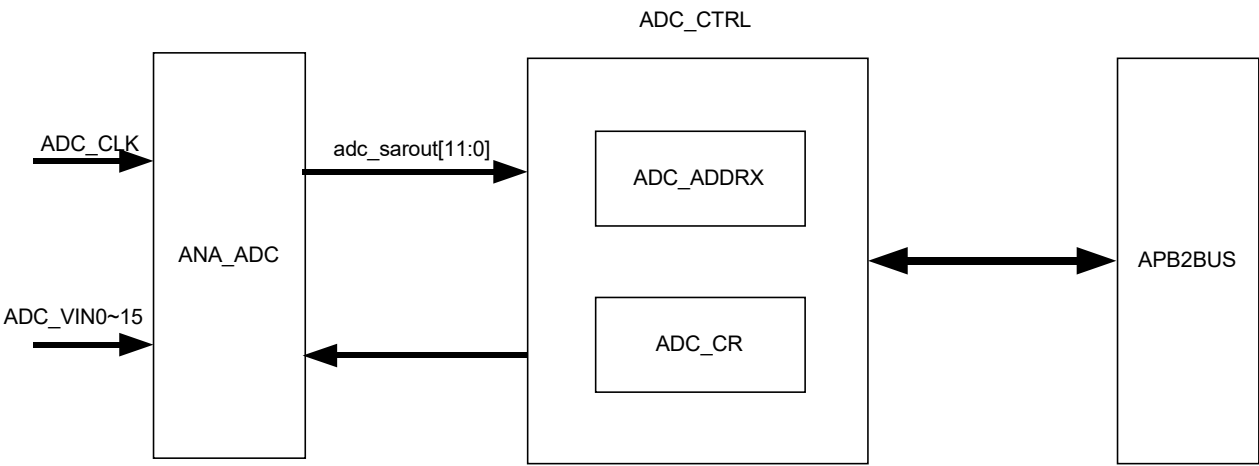


图 21-1 ADC 系统框图

21.5ADC 功能描述

下图显示了 ADC 框图

注: V_SENSOR(内部参考电压) 通道在 ADC 的 AIN15 通道。

21.6 任意通道工作模式

21.6.1 单次转换模式

在单次转换模式下，A/D 转换相应通道上只执行一次，具体流程如下：

软件设置寄存器 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，设置转换通道，置位 CHANY_MDEN。(单次转换模式，只需设置 CHANY_SEL0)通过软件、外部触发输入及定时器溢出置位 ADCR 寄存器的 ADST，开始 A/D 转换。A/D 转换完成时，A/D 转换的数据值将存储于数据寄存器 ADDATA 和 ADDRn 中。A/D 转换完成时，状态寄存器 ADSTA 的 ADIF 位置 1。若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。A/D 转换期间，ADST 位保持为 1。A/D 转换结束时，ADST 位自动清 0，A/D 转换器进入空闲模式。

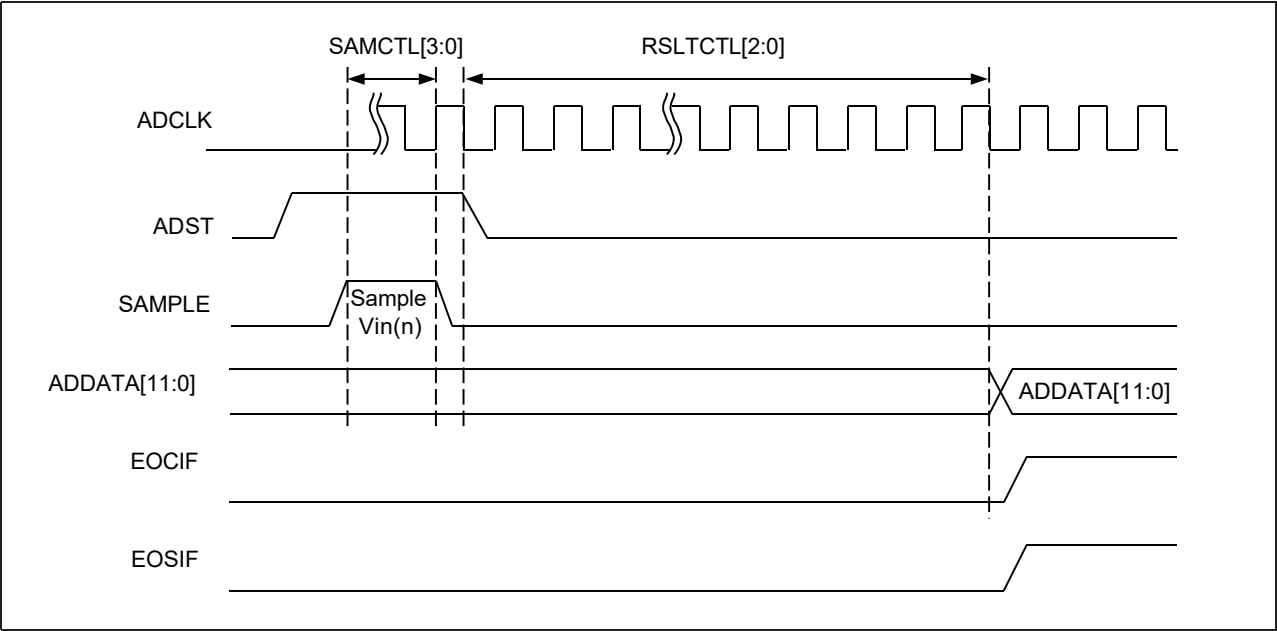


图 21-3 单次转换模式时序图

21.6.2 单周期扫描模式

在单周期扫描模式下，A/D 转换通道依软件配置执行一遍，具体流程如下：

软件设置寄存器 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，将需要转换的通道、数量设置好，然后置位 CHANY_MDEN。

通过软件、外部触发置位 ADCR 寄存器的 ADST，外部触发可软件配置触发延时，A/D 转换方向从 CHANY_SEL0 到 CHANY_SEL10，CHANY_SEL12，CHANY_SEL15 转换通道数量由 CHANY_NUM 配置，且 CHANY_SEL0 到 CHANY_SEL10，CHANY_SEL12，CHANY_SEL15 是任意配置的，可以完全相同，或完全不相同。

每路 A/D 转换完成时，A/D 转换的数据值将有序装载到相应通道的数据寄存器中，ADIF 转换结束标志被设置，若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。

A/D 转换结束后，ADST 位自动清 0，A/D 转换器进入空闲模式。

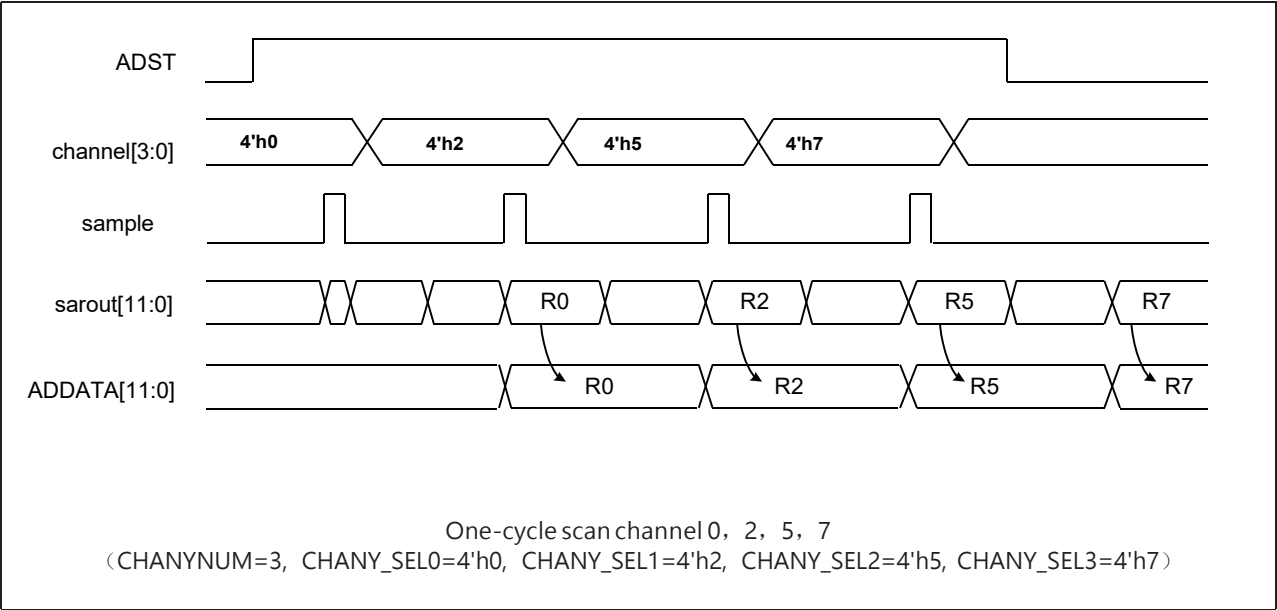


图 21-4 单周期扫描下通道转换时序图

21.6.3 连续扫描模式

在连续扫描模式下，A/D 转换通道依软件配置一直执行，直到软件禁止。具体流程如下： 软件设置寄存器 ADC_ANY_CFG, ADC_CHANY0, ADC_CHANY1, 将需要转换的通道、数量设置好，然后置位 CHANY_MDEN

通过软件、外部触发置位 ADCR 寄存器的 ADST，外部触发可软件配置触发延时，A/D 转换方向从 CHANY_SELO 到 CHANY_SEL10, CHANY_SEL12, CHANY_SEL15 转换通道数量由 CHANY_NUM 配置，且 CHANY_SELO 到 CHANY_SEL10, CHANY_SEL12, CHANY_SEL15 是任意配置的，可以完全相同，或完全不相同。

每路 A/D 转换完成时，A/D 转换的数据值将有序装载到相应通道的数据寄存器中，ADIF 转换结束标志被设置，若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。

只要 ADST 位保持为 1，持续进行 A/D 转换。当 ADST 位被清 0，A/D 转换停止，A/D 转换器进入空闲状态。

若在 A/D 转换过程中，软件更新 ADC_ANY_CFG, ADC_CHANY0, ADC_CHANY1，硬件不会立即更新这些配置，只会在当前设置的通道都转换结束时更新，即下一个扫描周期开始新的通道转换。

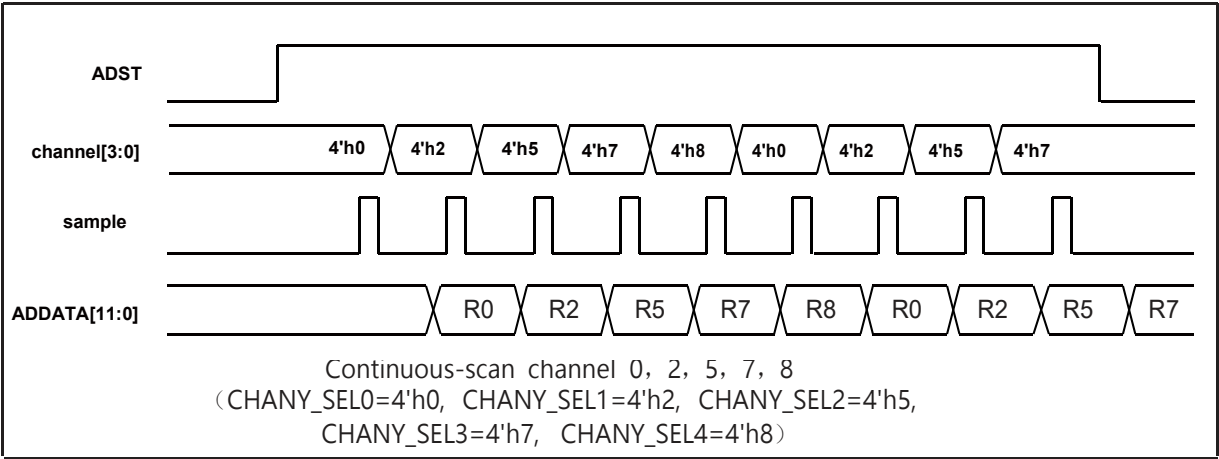


图 21-5 连续扫描模式通道转换时序图

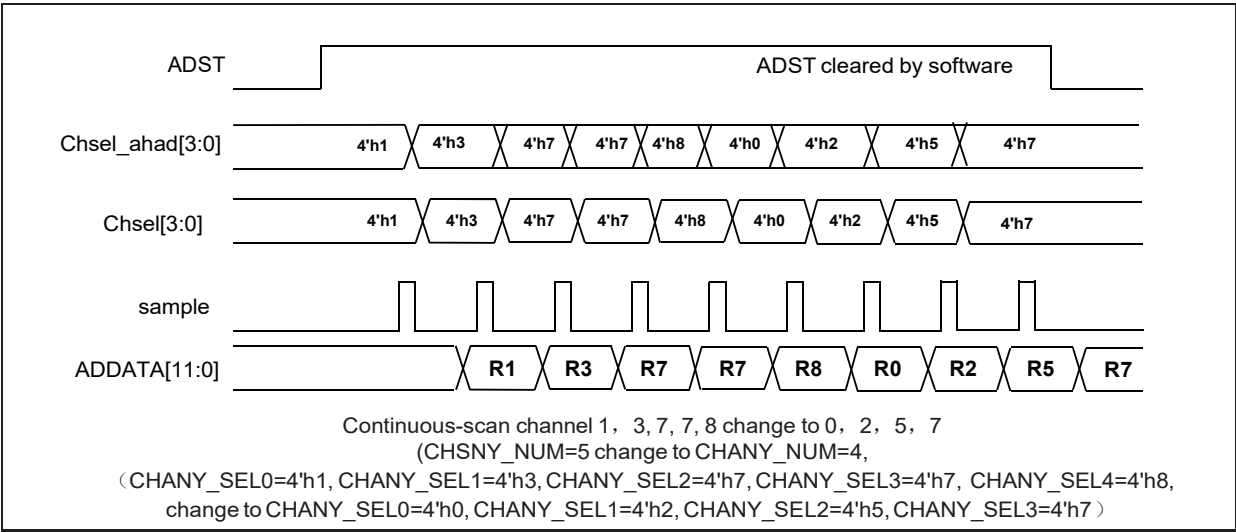


图 21-6 连续扫描模式动态更新配置时序图

21.7 数据对齐

ADCR 寄存器中的 ALIGN 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐，如下图所示。

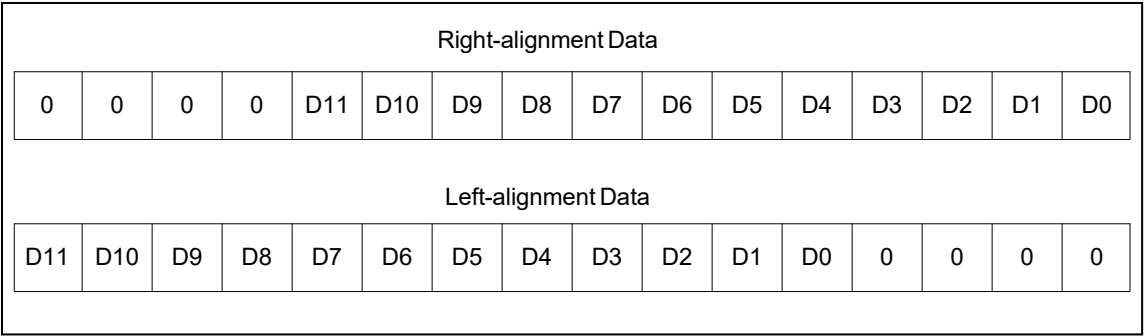


图 21-7 数据对其方式

21.7.1 可编程分辨率

ADC 转换有效位数可通过 ADC_CFG 寄存器中的 RSLTCTL[2: 0] 位更改，以便加快数据转换速率，有效数据位是在 12 位数据高位对齐。

21.7.2 可编程采样时间

ADC 的时钟 ADCLK 由 PCLK2 分频得到，分频系数可通过设置 ADCFG 寄存器的 ADCPRE 位来确定，即 $PCLK2 / (N + 2)$ 分频后作为 ADC 时钟。ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_CFG 寄存器中的 SAMCTL[3:0] 位更改。

设置 ADC 分辨率为 n 位 (n=8,9,10,11,12)，每个通道采样周期为 m。采样频率采样时间计算如下：

$$F_{\text{sample}} = F_{\text{ADCLK}} / (m + n + 0.5)。$$

假设分辨率配置为 12Bit，每个通道采样周期为 3.5T，则 $F_{\text{sample}} = F_{\text{ADCLK}} / 16$ 。总转换时间如下计算：

$T_{\text{CONV}} = \text{采样时间} + 12.5 \text{ 个转换周期}$ 例

如：

当 $ADCCLK = 16\text{MHz}$ ，采样时间为 3.5 周期 $T_{\text{CONV}} = 3.5 + 12.5 = 16$ ，周期 = $1\mu\text{s}$

21.8 外部触发转换

ADC 转换可以由外部事件触发 (例如定时器捕获，EXTI 线)。如果设置了 ADCR 寄存器的

TRGEN 位，就可以使用外部事件触发转换。通过设置 TRGSEL 位可以选择外部触发源。具体的外部触发源选择情况，可以参考 AD 控制寄存器相关位的描述。

外部触发可设置延时控制，具体参考 ADCR[21:19] 的 TRGSHIFT 的描述。在触发信号产生后，

延时 N 个 PCLK2 的时钟周期再开始采样。如果是触发扫描模式，只有第一个通道采样被延时，其余通道是在上一个采样结束后立即开始。

21.9 内部基准参考电压

ADC 的内部信号源通道连接了一个内部基准参考电压，大小为 1.2V，此通道把 1.2v 的参考电压输出转换为数字值。

内部参考电压有单独的始能位，可通过设置寄存器的相应位开启或关闭。

21.10 窗口比较器模式下 AD 转换结果监控

比较模式下提供了上限和下限两个比较寄存器。可通过软件设定 CMPCH 位选择监控通道。当

CPMHDATA ≥ CPMLDATA 时，比较结果大于或等于 ADCMPR 寄存器的 CPMHDATA

指定值或者小于 CMPLDATA 指定值，状态寄存器 ADSTA 的 ADWIF 位置 1。

当 CPMHDATA < CPMLDATA 时，比较结果如果等于 CPMHDATA 指定值或者处于两个指定值之间，则状态寄存器 ADSTA 的 ADWIF 位置 1。如果控制寄存器 ADCR 的 ADWIE 置位，将产生中断请求。

21.11 寄存器

21.11.1 寄存器总览

表 21-1 ADC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDDATA	A/D 数据寄存器	0x00000000
0x04	ADC_ADCFG	A/D 配置寄存器	0x00000000
0x08	ADC_ADCR	A/D 控制寄存器	0x00000000
0x10	ADC_ADCMPR	A/D 模拟看门狗比较寄存器	0x00000000
0x14	ADC_ADSTA	A/D 状态寄存器	0x00000000
0x18~0x40 0x48 0x54	ADC_ADDR 0~10,12,15	A/D 通道数据寄存器	0x00000000
0x58	ADC_ADSTA_EXT	A/D 扩展状态寄存器	0x00000000
0x5C	ADC_CHANY0	A/D 任意通道通道选择寄存器 0	0x00000000
0x60	ADC_CHANY1	A/D 任意通道通道选择寄存器 1	0x00000000
0x64	ADC_ANY_CFG	A/D 任意通道配置寄存器	0x00000000
0x68	ADC_ANY_CR	A/D 任意通道控制寄存器	0x00000000

21.11.2 ADC_ADDDATA 数据寄存器

偏移地址：0x00 复位值：

0x00000000

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

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值。
21	VALID	数据有效标志位（只读）（Valid Flag） 1： DATA[11:0] 位数据有效 0： DATA[11:0] 位数据无效 通道转换完成后硬件置位，读 ADC_ADDDATA 寄存器后该位自动清除。

Bit	Field	Description
20	OVERRUN	数据覆盖标志位（只读）（ Overrun Flag ） 1：DATA[11:0] 上一次转换数据被覆盖 0：DATA[11:0] 上一次转换数据未被覆盖 若数据装载至 DATA[11:0] 数据寄存器位之前，若上一次 DATA[11:0] 没有被读取，则 OVERRUN 将置位，读 ADC_ADDDATA 寄存器后该位自动清除。
19:16	CHANNELSEL	该 4 位显示当前数据寄存器中的数据所对应的通道（ Channel Selection ） 0000：表示当前数据寄存器所属通道为通道 0 0001：表示当前数据寄存器所属通道为通道 1 0010：表示当前数据寄存器所属通道为通道 2 0011：表示当前数据寄存器所属通道为通道 3 0100：表示当前数据寄存器所属通道为通道 4 0101：表示当前数据寄存器所属通道为通道 5 0110：表示当前数据寄存器所属通道为通道 6 0111：表示当前数据寄存器所属通道为通道 7 1000：表示当前数据寄存器所属通道为通道 8 1001：表示当前数据寄存器所属通道为通道 9 1010：表示当前数据寄存器所属通道为通道 10 1100：表示当前数据寄存器所属通道为通道 12 1111：表示当前数据寄存器所属通道为通道 15 其他：无效
15:0	DATA	12 位 A/D 当前通道转换结果（ Conversion Data ） 软件选择对齐方式。

21.11.3 ADC_ADCFG 配置寄存器

偏移地址：0x04 复位值：

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ADCPREL	SAMCTL				RSLTCTL			ADCPREH			VSEN	Res.	ADWEN	ADEN
	rw	rw				rw			rw			rw		rw	rw

Bit	Field	Description
31:15	Reserved	保留，必须保持复位值。
14	ADCPREL	ADC 时钟预分频低位（ ADC Prescaler Low Bits ） 预分频系数 $ADCPRE = \{ADCPREH, ADCPREL\}$
13:10	SAMCTL	选择通道 x 的采样时间（ Channel x Sample time selection ） 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。采样时间参考数据手册 ADC 特性 0000：2.5 周期 0100：42.5 周期 0001：8.5 周期 0101：56.5 周期 0010：14.5 周期 0110：72.5 周期 0011：29.5 周期 0111：240.5 周期 1000：3.5 周期 1001：4.5 周期 1010：5.5 周期 1011：6.5 周期 1100：7.5 周期 其他：保留

Bit	Field	Description
9: 7	RSLTCTL	ADC 转换数据分辨率选择 (Data Resolution Selection) 000: 12 位有效 001: 11 位有效 010: 10 位有效 011: 9 位有效 100: 8 位有效 其它: 保留
6: 4	ADCPREH	ADC 时钟预分频高位 (ADC Prescaler High Bits) 预分频系数 $ADCPRE = \{ADCPREH, ADCPREL\}$ ADC 时钟分频: $div = (ADCPRE + 2)$
3	VSEN	内部参考电压传感器使能 (Voltage Sensor Enable) 1: 内部电压传感器使能 0: 内部电压传感器禁用
2	Reserved	保留, 必须保持复位值。
1	ADWEN	A/D 窗口比较器使能 (ADC window comparison enable) 1: A/D 窗口比较器使能 0: A/D 窗口比较器禁用
0	ADEN	A/D 转换使能 (ADC Enable) 1: 使能 0: 禁止

21.11.4 ADC_ADCCR 控制寄存器

偏移地址: 0x08 复位值:

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						TRG_EDGE		Res.		TRGSHIFT			TRGSELH		Res.
						rw				rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCH				ALIGN	ADMD		ADST	Res.	TRGSELL			DMAEN	TRGEN	AWDIE	ADIE
rw				rw	rw		rw		rw			rw	rw	rw	rw

Bit	Field	Description
31:26	Reserved	保留, 必须保持复位值
25: 24	TRG_EDGE	触发源边沿选择 (Trigger Sources Edge Selection) 00: 双沿触发 01: 下沿触发 10: 上沿触发 11: 屏蔽触发
23:22	Reserved	保留, 必须保持复位值。
21: 19	TRGSHIFT	外部触发源采样延时 (External Trigger Sources Shift For Sample) 在触发产生后, 延时 N 个 PCLK2 的时钟周期再开始采样 (N=1, 2, 3...)。 000: 不延时 001: 4 个周期 010: 16 个周期 011: 32 个周期 100: 64 个周期 101: 128 个周期 110: 256 个周期 111: 512 个周期 如果是触发扫描, 则其它通道采样是在上一个转换结束后立即开始。

Bit	Field	Description
18 : 17	TRGSELH	外部触发源选择高位 (External Trigger Sources Select For High Bits) TRGSEL={TRGSELH,TRGSELL}
16	Reserved	保留, 必须保持复位值。
15 : 12	CMPCH	窗口比较通道选择 (Window comparison channel selection) 0000 = 选择比较通道 0 转换结果 0001 = 选择比较通道 1 转换结果 0010 = 选择比较通道 2 转换结果 0011 = 选择比较通道 3 转换结果 0100 = 选择比较通道 4 转换结果 0101 = 选择比较通道 5 转换结果 0110 = 选择比较通道 6 转换结果 0111 = 选择比较通道 7 转换结果 1000 = 选择比较通道 8 转换结果 1001 = 选择比较通道 9 转换结果 1010 = 选择比较通道 10 转换结果 1100 = 选择比较通道 12 转换结果 1111 = 所有扫描通道 其他: 无效
11	ALIGN	数据对齐格式 (Data Alignment) 1: 左对齐 0: 右对齐
10 : 9	ADMD	A/D 转换模式 (ADC Mode) 00: 单次转换 01: 单周期扫描 10: 连续扫描 11: 保留 当改变转换模式时, 软件要先清除 ADST 位。
8	ADST	A/D 转换开始 (ADC Start) 1: 转换开始 0: 转换结束或空闲状态 ADST 清除有下列两种方式: 在单次模式或者单周期模式下, 转换完成后 ADST 将硬件自动清除。 在连续扫描模式下, ADST 由软件或系统复位清除, 否则 A/D 转换将一直进行。
7	Reserved	保留, 必须保持复位值。
6 : 4	TRGSELL	外部触发源选择 (External trigger selection), 位 [18:17,6:4] 选择外部触发源 00000: TIM1_CC1 00001: TIM1_CC2 00010: TIM1_CC3 00011: Reserved 00100: TIM3_TRGO 00101: TIM1_CC4 和 TIM1_CC5 00110: TIM3_CC1 00111: EXTI 线 11 01000: TIM1_TRGO 01001: EXTI 线 4 01010: EXTI 线 5 01011: Reserved 01100: TIM3_CC4 01101: Reserved 01110: Reserved 01111: EXTI 线 15 10000: TIM1_CC4 10001: TIM1_CC5 其他: 无效

Bit	Field	Description
3	DMAEN	DMA 使能 (Direct Memory Access Enable) 1: DMA 请求使能 0: DMA 请求禁止
2	TRGEN	外部硬件触发源使能 (External Trigger Sources Enable) 1: 使用外部触发信号启动 A/D 转换 0: 禁止外部触发信号启动 A/D 转换
1	AWDIE	A/D 窗口比较器中断使能 (ADC window comparator interrupt enable) 1 = 使能 A/D 窗口比较器中断 0 = 禁用 A/D 窗口比较器中断
0	ADIE	A/D 中断使能 (ADC interrupt enable) 1 = 使能 A/D 中断 0 = 禁用 A/D 中断 如果 ADIF 置位, A/D 转换结束后产生中断请求。

21.11.5 ADC_ADCMPR 窗口比较寄存器

偏移地址: 0x10 复位值:

0x00000000

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

Bit	Field	Description
31:28	Reserved	保留, 必须保持复位值
27: 16	CMPHDATA	比较数值上限 (Compare data high limit) 该 12 位数值将和指定通道的转换结果相比较。
15:12	Reserved	保留, 必须保持复位值。
11: 0	CMPLDATA	比较数值下限 (Compare data low limit) 该 12 位数值将和指定通道的转换结果相比较。

21.11.6 ADC_ADSTA 状态寄存器

偏移地址: 0x14 复位值:

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OVERRUN											Res.	VALID		
	r												r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID								CHANNEL				Res.	BUSY	AWDIF	ADIF
r								r					r	rc_w1	rc_w1

Bit	Field	Description
-----	-------	-------------

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30:20	OVERRUN	通道 0~10 的数据覆盖标志位（Overrun Flag）
19	Reserved	保留，必须保持复位值
18:8	VALID	通道 0~10 的数据有效标志位（Valid Flag）
7:4	CHANNEL	当前转换通道（Current Convert Channel） 注：BUSY=1 时表示当前转换中的通道；BUSY=0 时表示即将转换的通道。
3	Reserved	保留，必须保持复位值
2	BUSY	通道转换忙/空闲（Busy） 1：A/D 转换器忙碌 0：A/D 转换器空闲
1	AWDIF	比较标志位（ADC window comparator interrupt flag） 选择的 A/D 转换通道，结果大于等于 ADCMPHR 或小于 ADCMPLR，该位置‘1’。该标志位写‘1’清零。
0	ADIF	A/D 转换结束标志位（ADC interrupt flag） 该位由硬件在通道组转换结束时设置，由软件清除。 1：A/D 转换完成 0：A/D 转换未完成 该标志位写‘1’清零

21.11.7 ADC_ADDRn 通道数据寄存器(n=0~10, 12, 15)

偏移地址：0x18~0x40, 0x48, 0x54 复位值：

0x00000000

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

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值。
21	VALID	有效标志位（只读）（Valid Flag） 1：DATA[11:0]位数据有效 0：DATA[11:0]位数据无效 对应通道转换完成后硬件置位，读 ADC_ADDRn 寄存器自动清除。
20	OVERRUN	数据覆盖标志位（只读）（Overrun Flag） 1：DATA[11:0]数据被覆盖 0：DATA[11:0]数据最近一次转换结果 若在装载至 DATA[11:0] 数据寄存器之前，上一次 DATA[11:0]数据没有被读取，OVERRUN 将置位。读 ADC_ADDRn 寄存器硬件自动清除。
19:16	Reserved	保留，必须保持复位值。
15:0	DATA	通道 n 的 12 位 A/D 转换结果（Covert Data） 软件选择对齐方式。

21.11.8 ADC_ADSTA_EXT 扩展状态寄存器

偏移地址：0x58

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OVERR UN15	Res.		OVERR UN12	VALID15	Res.		VALID12
								r			r	r			

Bit	Field	Description
31:8	Reserved	保留，必须保持复位值。
7	OVERRUN15	通道 15 的数据覆盖标志位 (Overrun flag)
6:5	Reserved	保留，必须保持复位值。
4	OVERRUN12	通道 12 的数据覆盖标志位 (Overrun flag)
3	VALID15	通道 15 的有效标志位 (Valid flag)
2:1	Reserved	保留，必须保持复位值。
0	VALID12	通道 12 的有效标志位 (Valid flag)

21.11.9 ADC_CHANY0 任意通道的通道选择寄存器 0

偏移地址：0x5C 复位值：

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL7				CHANY_SEL6				CHANY_SEL5				CHANY_SEL4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL3				CHANY_SEL2				CHANY_SEL1				CHANY_SEL0			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL7	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
27:24	CHANY_SEL6	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
23:20	CHANY_SEL5	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
19:16	CHANY_SEL4	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
15:12	CHANY_SEL3	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
11:8	CHANY_SEL2	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
7:4	CHANY_SEL1	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
3:0	CHANY_SEL0	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_CHANY0 影子寄存器。在 ADC 工作期间，若更改 ADC_CHANY0 的值，只会更新到其影子寄存器中，在 ADC 转换最后一个通道时，影子寄存器的值才会更新至 ADC_CHANY0，实现通道动态切换。

21.11.10 ADC_CHANY1 任意通道的通道选择寄存器 1

偏移地址：0x60 复位值：

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL15				Res.								CHANY_SEL12			
rw												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CHANY_SEL10				CHANY_SEL9				CHANY_SEL8			
				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL15	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
27:24	Reserved	保留，必须保持复位值。
23:20	Reserved	保留，必须保持复位值。
19:16	CHANY_SEL12	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
15:12	Reserved	保留，必须保持复位值。
11:8	CHANY_SEL10	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
7:4	CHANY_SEL9	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。
3:0	CHANY_SEL8	可配置为 ch0 ~ ch10,ch12,ch15 中的任意通道。

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_CHANY1 影子寄存器。在 ADC 工作期间，若更改 ADC_CHANY1 的值，只会更新到其影子寄存器中，在 ADC 转换最后一个通道时，影子寄存器的值才会更新至 ADC_CHANY1，实现通道动态切换。

21.11.11 ADC_ANY_CFG 任意通道配置寄存器

偏移地址：0x64 复位值：

0x00000000

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

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值

Bit	Field	Description
3:0	CHANY_NUM	转换通道数配置(Number of Any Channel Mode): 0000: 0 通道 0001: 0~1 通道 0010: 0~2 通道 0011: 0~3 通道 0100: 0~4 通道 0101: 0~5 通道 0110: 0~6 通道 0111: 0~7 通道 1000: 0~8 通道 1001: 0~9 通道 1010: 0~10 通道 1100: 0~10,12 通道 1111: 0~10,12,15 通道 其他: 无效

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_NUM 影子寄存器。在 ADC 工作期间，若更改 ADC_NUM 的值，只会更新到其影子寄存器中，在 ADC 转换最后一个通道时，影子寄存器的值才会更新至 ADC_NUM，实现通道动态切换。

21.11.12 ADC_ANY_CR 任意通道控制寄存器

偏移地址：0x68 复位值：

0x00000000

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

Bit	Field	Description
31:1	Reserved	保留，必须保持复位值
0	CHANY_MDEN	任意通道转换使能位(Any Channel Mode Enable): 1: 使能任意通道转换 0: 禁止任意通道转换 在使能后，配置 ADC 通道功能发生变化。由两部分共同控制，CHANY_NUM 是配置 CH0 ~ CH10, CH12, CH15 中的通道数，然后 CH0 ~ CH10, CH12, CH15 分别由 CHANY_SEL0 ~ CHANY_SEL15 配置为任意 ADC 通道。

注：在任意通道模式，且单周期/连续扫描模式下，关闭 ADC 时，必须先禁止 ADC_ADCR 的 ADST 位，然后判断 ADC_ADSTA 的 BUSY 位是否为 0，即等到 ADC 转换完成，再禁止 ADC_ANY_CR 的 CHANY_MDEN 位。

22 COMP 比较器

22.1 简介

芯片内嵌 1 个通用比较器（COMP_x， $x=1$ ），比较器集成数字滤波器，其比较结果可输出至定时器或产生中断，或产生触发低功耗模式唤醒事件。也可与定时器结合使用，与定时器输出的 PWM 相结合，组成周期性的电流控制回路。

22.2 功能框图

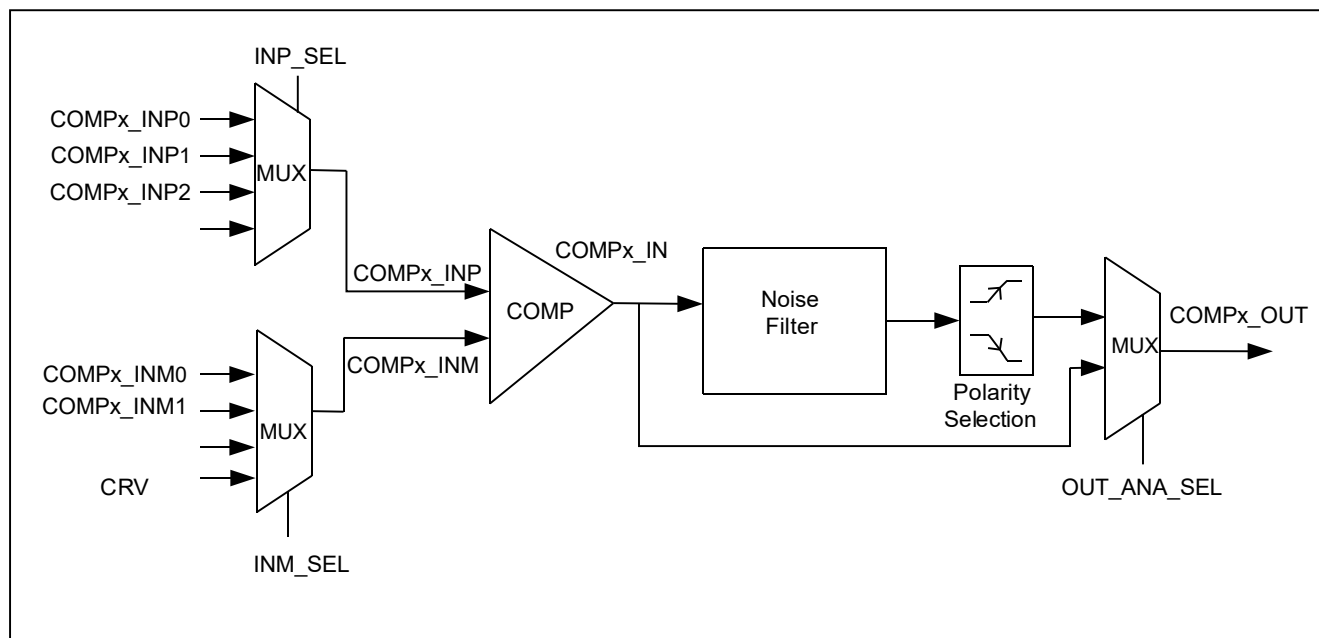


图 22-1 比较器框图

22.3 主要特征

- 模拟输入为可复用的 I/O 引脚；
- 可编程迟滞电压；
- 支持多种速率和功耗；
- 支持比较结果的滤波功能，滤波周期可配置；
- 输出送至 I/O 引脚或定时器；
- 支持通过外部事件 EXTI 把 CPU 从睡眠和停机模式唤醒；

- 每个比较器支持 4 个正相输入和 5 个反相输入，带有轮询功能；
- 可以实现定期切换的轮询功能；
- 可控制轮询通道 1/2/3 或 1/2；
- 正向输入和反向输入可同步轮询，或可选固定反向输入端。

22.4 功能描述

22.4.1 比较器时钟和复位

COMP 的输入时钟与 APB2 CLK 同步。在使用比较器之前，要先通过设置 RCC 控制器中的对应比较器时钟使能位来使能比较器时钟。配置 RCC 控制器中对应的比较器复位控制位可进行比较器的软件复位操作。

22.4.2 比较器开关控制

在使用比较器之前，需要置位 COMPx_CSR 寄存器的 EN 位给 COMP 上电。设置 EN 位为 1 时，它将 COMP 从断电状态唤醒，清除 EN 位（EN 位置 0）可停止比较器工作。

22.4.3 比较器输入和输出

当 I/O 用作比较器输入时，必须在 GPIO 寄存器中将其设置为模拟输入模式。模拟比较器比较的结果经过 Filter 滤波模块，再经过极性配置后输出，送入其他模块使用；当 OUT_ANA_SEL=1 时，模拟比较器的比较结果不经过任何模块，直接输出。比较器的输出可选择滤波功能（参考 COMPx_CSR 寄存器的 OFLT 的配置），可以作为内部各种定时器的输入（参考 COMPx_CSR 寄存器的 OUT_SEL 的配置），也可以输出到 I/O。

22.4.4 比较器通道选择

每个 COMP 有 4 个正相输入和 5 个反相输入通道，正相输入可从四个外部引脚之间选取，反相输入可从 4 个外部引脚和 CRV 电压分压值选取。CRV 的电压可选择 VDDA 或者内部 1.2V 基准电压（V_{REFINT}）的分压。

COMP 可以在普通工作模式下通过软件选择输入通道，也可以在轮询工作模式下通过硬件轮询的方式分时监测多个通道的比较结果，从逻辑上类似于多个比较器同时工作。

在普通工作模式下，比较器比较所选择的 INP 和 INM 端口上的信号，具体流程如下：

配置 COMPx_CSR 寄存器的 INP_SEL 位和 INM_SEL 位，选择所要比较的信号；

配置 COMPx_CSR 寄存器的 EN 位，比较器开始上电工作；

比较的结果存放于 COMPx_CSR 寄存器的 OUT 位。

另外，当 COMP 的 INM_SEL 选择 CRV 时，需要配置 COMP_CRV 寄存器的 CRV_SEL 位，然后将 CRV_EN 置位（在上面步骤 2 之前）。

在轮询工作模式下，COMP 的 INP 端口上的信号将会周期性的轮询变化，而 INM 端口的信号可以配置 COMPx_POLL 寄存器的 FIXN 位选择跟随 INP 端口变化或者由 COMPx_CSR 的 INM_SEL 位来配置。需要注意的是，当启动轮询功能以后，COMPx_CSR 的 INP_SEL 位将失去作用，同样的，如果 COMPx_POLL 寄存器的 FIXN 位选择 INM 端口跟随 INP 轮询变化，COMPx_CSR 的 INM_SEL 位也将失去作用。具体流程如下：

配置 COMPx_POLL 寄存器的 PERIOD 位来选择所需要的轮询等待周期；

配置 COMPx_POLL 寄存器的 FIXN 位来决定 INM 端口的信号是否跟随 INP 端口轮询变化； 配

置 COMPx_POLL 寄存器的 POLL_CH 位决定所需要轮询的通道是 1/2/3 或者 1/2；

配置 COMPx_POLL 寄存器的 POLL_EN 位，启动轮询功能；

配置 COMPx_CSR 寄存器的 EN 位，比较器开始上电工作；

轮询比较的结果存放于 COMPx_POLL 寄存器的 POUT 位，其中 POUT[2]、POUT[1]、POUT[0] 位分别存放轮询通道 3/2/1 的比较结果。

22.4.5 中断和唤醒

比较器的输出可以芯片内部硬件连接到事件控制器。每个比较器有自己的 EXTI 信号，能产生事件来退出低功耗模式。详细内容可以参考手册的中断和事件部分。

22.4.6 功耗模式

在具体应用中可以通过调整比较器功耗和响应时间得到最优的结果。

COMPx_CSR 寄存器的 MODE 位有下面 4 种设置：

- 00：高速/高功耗；
- 01：中速/中等功耗；
- 10：低速/低功耗；
- 11：极低速/极低功耗。

22.4.7 比较器锁定机制

比较器能用于安全的用途，比如过流或者过热保护。在某些特定的应用中，有必要保证比较器设置不能被随意改写或者程序计数器破坏所改变。

为了这个目的，比较器控制和状态寄存器可以设为只读。

一旦设置完成，LOCK 位被设置为 1，这导致整个 COMPx_CSR 寄存器变成只读，包括 LOCK 位在内，只能通过 MCU 复位所清除。（详细参考 COMPx_CSR 关于 LOCK 位的配置）

22.4.8 迟滞电压

为了避免噪声信号导致的无效输入，比较器支持可配置的迟滞电压，（详细参考 COMPx_CSR 的 HYST 位的配置）。

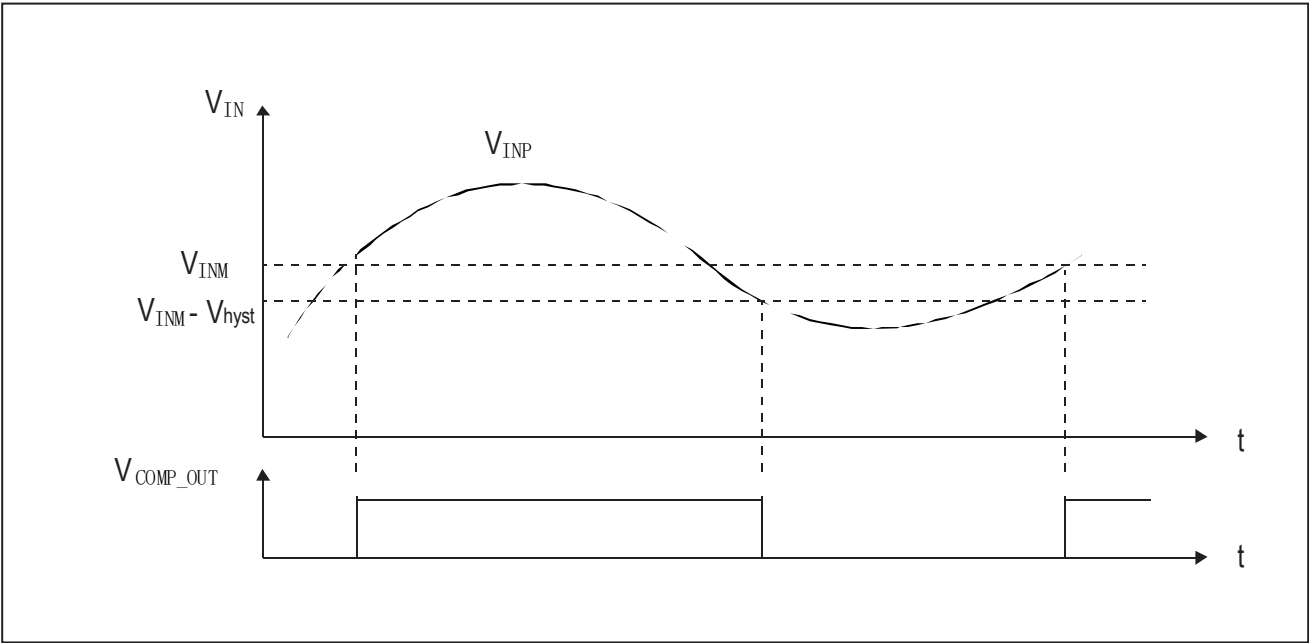


图 22-2 比较器的迟滞

22.5 寄存器

表 22-1 COMP 寄存器概览

Offset	Acronym	Register Name	Reset
0x0	COMPx_CSR (x=1)	比较器 x (x=1) 控制和状态寄存器	0x0000000
0x18	COMP_CRV	比较器外部参考电压寄存器	0x0000000
0x1C	COMPx_POLL (x=1)	比较器 x (x=1) 轮询寄存器	0x0000000

22.5.1 比较器控制状态寄存器（COMPx_CSR）（x=1）

偏移地址：0x0 复位值：

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT	OUT_A NA_SEL	Res.								OFLT			HYST	
rw	r	rw									rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	Res.	OUT_SEL				Res.	INP_SEL		INM_SEL			MODE		Res.	EN

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

Bit	Field	Description
31	LOCK	比较器寄存器锁 (Comparator Register Lock) 该位只能写一次, 由软件置'1', 由系统复位清零。一旦设置, 比较器 x 的所有控制位为只读。 1: COMPx_CSR 只读 0: COMPx_CSR 可读可写
30	OUT	比较器 x 输出 (Comparator x Output) 反映比较器 x 输出状态。 1: 高输出 (正相输入高于反相输入) 0: 低输出 (正相输入低于反相输入)
29	OUT_ANA_SEL	比较器 x 输出来源选择 (Comparator x output source selection) 1: 选择模拟输出信号 0: 选择经过同步的模拟输出信号 备注: 当需要 COMP 在低功耗模式下产生中断唤醒 CPU 时, 可以将 OUT_ANA_SEL 置 1。
28 : 21	Reserved	始终读为 0。
20 : 18	OFLT	比较器 x 输出滤波周期 (Comparator Output Filter Period) 这些位控制比较器 x 的输出滤波周期, 当比较器输出信号小于滤波周期宽度时, 被认为是无效信号而被过滤掉, 否则认为是有效信号, 其中 n 代表 COMPx_POLL -> PERIOD 配置的轮询等待周期。 111: 128 * n 个时钟周期 110: 64 * n 个时钟周期 101: 32 * n 个时钟周期 100: 16 * n 个时钟周期 011: 8 * n 个时钟周期 010: 4 * n 个时钟周期 001: 2 * n 个时钟周期 000: 无滤波
17 : 16	HYST	比较器 x 迟滞电压 (Comparator x Hysteresis) 这些位控制比较器 x 的迟滞电压。 当 MODE = 00, 11: 85mV, 10: 45mV 01: 22mV 00: 0mV 当 MODE = 01、10 或 11 时 11: 60mV, 10: 32mV 01: 15mV 00: 0mV
15	POL	比较器 x 输出极性控制 (Comparator x Output Polarity) 该位用于选择比较器 x 输出极性。 1: 输出端反相输出 0: 输出端同相输出
14	Reserved	始终读为 0。
13 : 10	OUT_SEL	比较器 x 输出选择 (Comparator x Output Selection) 这些位用来选择比较器 x 的输出目标。 0010: 定时器 1 刹车输入 0110: 定时器 1 Ocrefclear 输入 1010: 定时器 3 输入捕捉 1 1011: 定时器 3 Ocrefclear 输入 其他: 无选择
9	Reserved	始终读为 0。
8 : 7	INP_SEL	比较器 x 正相输入选择 (Comparator x Normal Phase Input Selection) 这些位用于选择连接到比较器 x 正相输入端的信号源。 00: COMP1_INP0 (PA1), 即正相输入端通道 0; 01: COMP1_INP1 (PA2), 即正相输入端通道 1; 10: COMP1_INP2 (PA3), 即正相输入端通道 2; 11: COMP1_INP3 (PA4), 即正相输入端通道 3;

Bit	Field	Description
6: 4	INM_SEL	比较器 x 反相输入选择 (Comparator x Inverting Input Selection) 这些位用于选择连接到比较器 x 反相输入端的信号源。 000: COMP1_INM0 (PA5), 即反相输入端通道 0; 001: COMP1_INM1 (PA6), 即反相输入端通道 1; 010: COMP1_INM2 (PA7), 即反相输入端通道 2; 011: COMP1_INM3 (PD6), 即反相输入端通道 3; 100: COMP1_INM4 (CRV), 即反相输入端通道 4; 其他: 无选择
3: 2	MODE	比较器 x 模式 (Comparator x Mode) 比较器 x 的工作模式控制位, 允许调整速率和功耗。 11: 极低功率/极低速 10: 低功率/低速 01: 中等功率/中速 00: 高功率/高速
1	Reserved	始终读为 0。
0	EN	比较器 x 使能 (Comparator x Enable) 该位是比较器开关使能位。 1: 比较器 x 使能 0: 比较器 x 禁止

22.5.2 比较器外部参考电压寄存器 (COMP_CRV)

偏移地址: 0x18 复位值:

0x00000000

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

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5	CRV_SRC	比较器参考电压源选择 (Comparator Reference Voltage Source Select) 0: VREFINT (内部电压) 1: VDDA (外部电压) 备注: 当选用内部电压作比较器的参考电压源时, 需要提前配置 ADC_ADCFG 来打开内部电压传感器使能
4	CRV_EN	比较器参考电压使能 (Comparator Reference Voltage Enable) 1: 比较器参考电压 (CRV) 使能 0: 比较器参考电压 (CRV) 禁止

Bit	Field	Description
3:0	CRV_SEL	比较器参考电压选择 (Comparator Reference Voltage Select) 0000: 1/20 VDDA 或 1/20 V _{REFINT} 0001: 2/20 VDDA 或 2/20 V _{REFINT} 0010: 3/20 VDDA 或 3/20 V _{REFINT} 0011: 4/20 VDDA 或 4/20 V _{REFINT} 0100: 5/20 VDDA 或 5/20 V _{REFINT} 0101: 6/20 VDDA 或 6/20 V _{REFINT} 0110: 7/20 VDDA 或 7/20 V _{REFINT} 0111: 8/20 VDDA 或 8/20 V _{REFINT} 1000: 9/20 VDDA 或 9/20 V _{REFINT} 1001: 10/20 VDDA 或 10/20 V _{REFINT} 1010: 11/20 VDDA 或 11/20 V _{REFINT} 1011: 12/20 VDDA 或 12/20 V _{REFINT} 1100: 13/20 VDDA 或 13/20 V _{REFINT} 1101: 14/20 VDDA 或 14/20 V _{REFINT} 1110: 15/20 VDDA 或 15/20 V _{REFINT} 1111: 16/20 VDDA 或 16/20 V _{REFINT}

22.5.3 比较器轮询寄存器 (COMPx_POLL) (x=1)

偏移地址: 0x1C 复位值:

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					POUT			Res.	PERIOD			Res.	FIXN	POLL_C H	POLL_E N
					r			rw	rw			rw	rw	rw	rw

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	POUT	轮询通道输出 (Polling Output) 只读, 反映轮询通道输出状态, POUT[0]对应通道 1, POUT[1]对应通道 2, POUT[2]对应通道 3。 1: 高输出 (正相输入高于反相输入) 0: 低输出 (正相输入低于反相输入)
7	Reserved	始终读为 0。
6:4	PERIOD	轮询等待周期 (Polling Wait Cycle) 每 n 个 PCLK2 周期切换到下一个轮询通道。 111: 128 个时钟周期 110: 64 个时钟周期 101: 32 个时钟周期 100: 16 个时钟周期 011: 8 个时钟周期 010: 4 个时钟周期 001: 2 个时钟周期 000: 1 个时钟周期
3	Reserved	始终读为 0。

Bit	Field	Description
2	FIXN	轮询通道反相输入端固定 (Polling Inverting Input Fix) 1: 轮询通道反相输入固定。由 CSR 寄存器 INM_SEL 决定, 此时, INP_SEL 无效。 0: 轮询通道反相输入不固定。与 INP 通道同时变化, 此时 INM_SEL 和 INP_SEL 无效。
1	POLL_CH	比较器轮询通道 (Comparator Polling Channel) 1: 轮询通道 1/2/3 0: 轮询通道 1/2
0	POLL_EN	比较器轮询模式使能 (Comparator Polling Enable) 1: 比较器轮询模式使能 0: 比较器轮询模式禁止

23 CRC 循环冗余校验计算单元

23.1 简介

CRC 计算单元利用固定的多项式来 32 位数据的 CRC 校验值，用于对数据传输或数据存储的完整性进行验证。

23.2 主要特征

- 支持 CRC-32/MPEG-2（以太网）多项式：0x4C11DB7

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

- 支持 32 位宽的数据寄存器用于输入/输出
- 硬件计算时间为 3 个 HCLK 周期
- 8 位独立数据寄存器，用于存放临时数据。

23.3 功能描述

23.3.1 功能框图

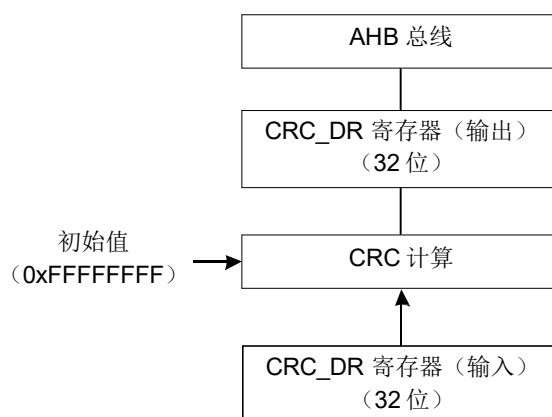


图 23-1 CRC 功能框图

23.3.2 功能概述

- CRC 计算单元含有 1 个 32 位数据寄存器：
- 对该寄存器进行写操作时，作为输入寄存器，可以输入要进行 CRC 计算的新数据。

- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。
- 每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合 (对整个 32 位字进行 CRC 计算，而不是逐字节地计算)。
- 在 CRC 计算期间会暂停写操作，因此可以对寄存器 CRC_DR 进行背靠背写入或者连续地写•读操作。
- 可以通过设置寄存器 CRC_CR 的 RST 位来重置数据寄存器 CRC_DR 为 0xFFFFFFFF。该操作不影响寄存器 CRC_IDR 内的数据。

23.3.3 使用方法

23.3.3.1 CRC 计算操作步骤

- 使能 CRC 模块时钟；
- 复位 CRC 模块；
- 通过配置 CRC 控制寄存器（CRC_CR）的 RST 位，将 CRC 恢复到初始状态；
- 将数据依次写入 CRC 数据寄存器（CRC_DR）；
- 读取 CRC 数据寄存器（CRC_DR），得到 CRC 计算结果。

23.4 寄存器

23.4.1 寄存器总览

表 23-1 CRC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CRC_DR	CRC 数据寄存器	0xFFFFFFFF
0x04	CRC_IDR	CRC 独立数据寄存器	0x00000000
0x08	CRC_CR	CRC 控制寄存器	0x00000000

23.4.2 CRC_DR CRC 数据寄存器

偏移地址：0x00 复位值：
0xFFFF FFFF

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

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

Bit	Field	Description
31: 0	DR	数据寄存器（Data Register） 写入时，作为输入寄存器，将写入的数据和前一次的结果做 CRC 计算 读取时，返回 CRC 计算的结果

23.4.3 CRC_IDR CRC 独立数据寄存器

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	IDR	8 位通用数据寄存器（General-purpose 8-bit Data Register） 临时存放 1 个字节的数据空间。 不受 CRC 控制寄存器（CRC_CR）的 RST 位影响。

23.4.4 CRC_CR CRC 控制寄存器

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RST	复位 CRC 计算单元（CRC Reset） CRC 数据寄存器（CRC_DR）复位为 0xFFFF FFFF 该位只能写“1”，硬件自动清“0”

24 DBG 调试支持

24.1 介绍

芯片内核包含硬件调试模块，主要用于功能的调试。当内核在取指（指令断点）或访问数据（数据断点）时，硬件调试模块可以控制内核停止，此时用户可以查询内核的内部状态和系统的外部状态。查询完成后，内核可以继续执行当前程序。

当芯片与调试器连接开始调试时，调试器自动调用内核的调试模块进行调试操作。

24.2 功能描述

24.2.1 功能框图

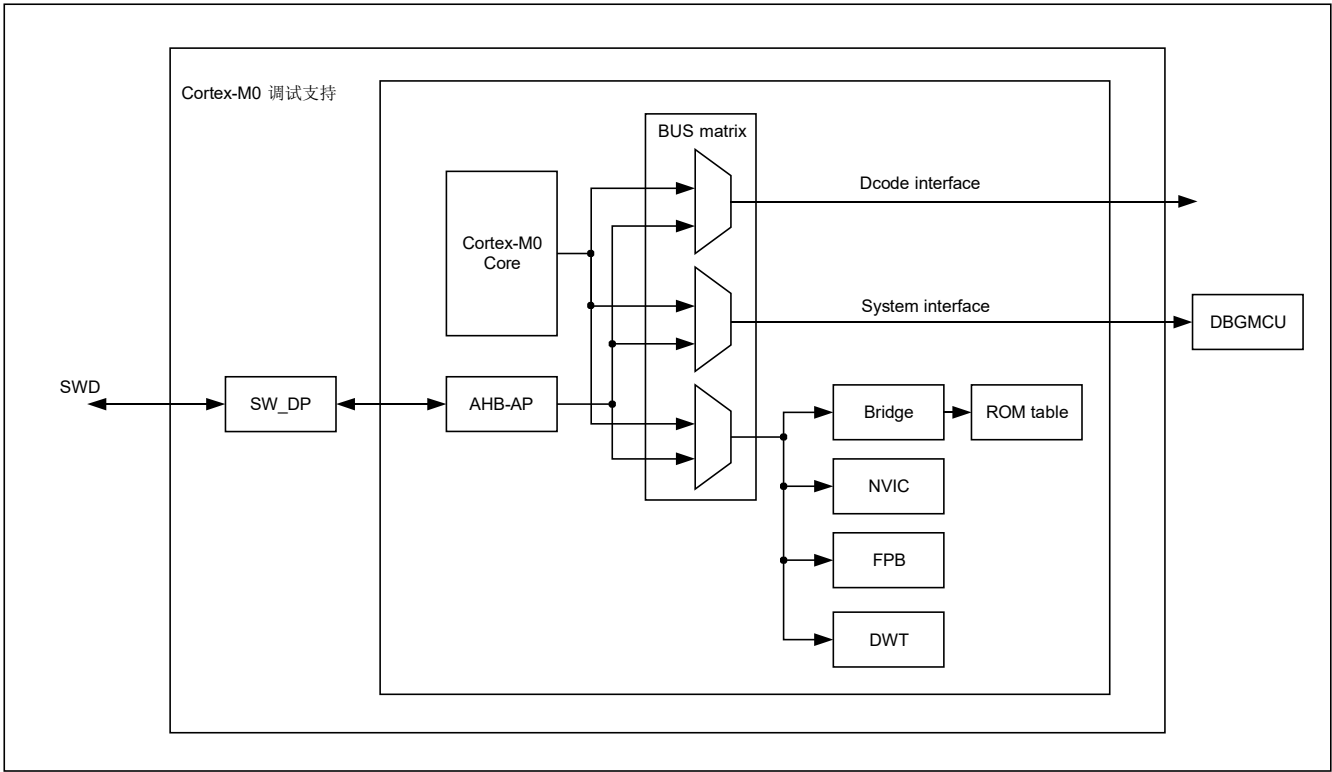


图 24-1 调试功能框图

Cortex-M0 内核含有调试单元，该单元由以下部分组成：

- SWDP: SW 调试端口
- BPU: 断点调试单元
- DWT: 数据观察点和跟踪

24.2.2 SWD 内部上拉与下拉

SWD 引脚输入直接控制调试模式，不能悬空。为了保证 I/O 电平可控，SWD 引脚上内嵌了上拉和下拉电阻。

- SWDIO: 内部上拉
- SWCLK: 内部下拉

软件可以把这些 I/O 口作为普通的 I/O 口使用，此时默认的上下拉功能关闭，参考通用端口 GPIO 章节。

24.2.3 SWJ 调试端口

该芯片的 2 个普通 I/O 口可用作 SWD-DP 接口引脚，不同封装都支持 SWD 调试端口。

表 24-1 SWD 调试端口管脚

SWD-DP 端口引脚名称	SW 调试接口		引脚分配
	类型	调试功能	
SWDIO	输入/输出	串行数据输入/输出	PA13
SWCLK	输入	串行时钟	PA14

24.3 ID 代码和锁定机制

在芯片内部有多个 ID 编码，如下表：

表 24-2 ID 编码

ID 名	芯片
DEV_ID	0x4C553800
CPU TAP SW ID	0x0BC11477

24.3.1 微控制器设备 ID 编码

微控制器内部包含设备 ID 编码，此 ID 定义了微控制器的硅片版本，并且映射到外部 APB 总线上。通过用户代码与调试接口均能够获取此 ID 编码。

24.3.2 Cortex JEDEC-106 ID 编码

微控制器有一个 JEDEC-106ID 编码。它位于映射到内部 PPB 总线地址为 4KB ROM 表中。

24.4 SW 调试端口

24.4.1 SW 协议介绍

此同步串行协议使用 2 个引脚：主机到目标的时钟信号（SWCLK）与双向数据信号（SWDIO）。

SWDIO 作为双向数据线，需连接上拉电阻（ARM 建议值 100K）。SWDIO 引脚内嵌了上拉电阻， 无需额外的外接电阻。

数据从低位开始传输，允许读写寄存器组 DPACC 与 APACC。

根据协议，当 SWDIO 改变方向时，同时需要插入一个转换时间（默认一个 Bit 时间，具体可以通过 SWCLK 调整），这段间内，任何设备不能驱动此信号线。

24.4.2 SW 协议序列

一次序列包含三个阶段：

- 主机发送 8 位请求包；
- 目标发送 3 位确认应答；
- 根据配置方向，主机或目标发送 33 位（包含一位校验位）数据；

表 24-3 8bit 请求包

比特位	名称	描述
0	起始	必须为 1
1	APnDP	0：访问 DP 1：访问 AP
2	RnW	0：写请求 1：读请求
4: 3	A[3 : 2]	DP 或 AP 寄存器的地址
5	Parity	前面比特位的校验位
6	Stop	0
7	Park	不能由主机驱动，由于有上拉，目标永远读为 1

注：每一个请求包紧跟一个 Bit 转换时间。更多关于 DAPCC 与 APACC 寄存器的信息，查看 ARM 相关的 CPU 技术参考手册。

表 24-4 3bit 应答包

比特位	名称	描述
0: 2	ACK	001：失败 010：等待 100：成功

注：当应答信号（ACK）处于上表的情况之一时，应答位后有一个转换时间。

表 24-5 33bit 数据包

比特位	名称	描述
0: 31	WDATA/RDATA	写或读的数据
32	Parity	32 位数据的奇偶校验位

注：读数据位结束后等待一个转换时间。

24.4.3 SW-DP 状态机 (Reset, Idle states, ID code)

SW-DP 状态机通过内部的 ID 码识别 SW_DP，遵守 JEP-106 标准,具体信息请参考 ARM 相关手册。

直到调试器读取 ID 之前，SW-DP 的状态机不会工作。

- 当出现了上电复位，或 DP 从 JTAG 切换到 SWD 后，或超过 50 个周期的高电平时，SW-DP 状态机将处于复位状态；
- 如果 RESET 状态之后出现了至少 2 个周期的低电平，状态机会切换到 IDLE 状态；
- 状态机开始处于复位态，工作时必须先切换到 IDLE 态，先执行读 DP-SW ID 寄存器的操作。否则，调试器无法进行其它正常的传输，会出现 ACK Fault；

24.4.4 DP 和 AP 读 / 写访问

- 对 DP 的读操作没有延时：调试器将直接获得数据（如果 ACK 返回成功状态），或者处于等待状态（如果 ACK 返回等待状态）；
- 对 AP 的读操作具有延时。这意味着前一次读操作的结果只能在下一次操作时获得。如果下一次的操作不是对 AP 的访问，则必须读 DP-RDBUFF 寄存器来获得上一次读操作的结果；
- DP-CTRL/STAT 寄存器的 READOK 标志位会在每次 AP 读操作和 RDBUFF 读操作后更新，以通知调试器 AP 的读操作是否成功；
- SW-DP 具有写缓冲区（DP 和 AP 都有写缓冲），这使得其它传输进行时，仍然可以接受写操作。如果写缓冲区满，调试器将获得一个等待的 ACK 响应。读 IDCODE 寄存器，读 CTRL/STAT 寄存器和写 ABORT 寄存器操作在写缓冲区满时仍被接受；
- 由于 SWCLK 和 HCLK 的异步性，需要在写操作后（在奇偶校验位后）插入 2 个额外的 SWCLK 周期，以确保内部写操作正确完成。这两个额外的时钟周期需要在 IDLE 状态下插入。这个操作步骤在写 CTRL/STAT 寄存器以提出一个上电请求时尤其重要，否则下一个操作（在内核上电后才有效的操作）会立即执行，这将会导致操作失败；

24.4.5 SW-DP 寄存器

当 APnDP=0 时，可以访问以下这些寄存器。

表 24-6 SW-DP 寄存器

A[3:2]	读 / 写	SELECT 寄存器的 CTRLSEL 位	寄存器	描述
00	读		IDCODE	固定为 0x0BB1 1477（用于识别 SW-DP）。
00	写		ABORT	
01	读/写	0	DP-CTRL /STAT	请求一个系统或调试的上电操作；配置 AP 访问的操作式；控制比较，校验操作； 读取一些状态位（溢出，上电响应）。
01	读/写	1	WIRE CONTROL	配置串行通信物理层协议（如转换时间长度等）。
10	读		READ RESEND	允许从一个错误的调试传输中恢复数据而不用重复最初的 AP 传输。
10	写		SELECT	选择当前的访问端口和有效的 4 字长寄存器窗口。
11	读/写		READ BUFFER	这个寄存器会从 AP 捕获上一次读操作的数据结果，因此可以获得数据而不必再启动一个新的 AP 传输。

24.4.6 SW-AP 寄存器

当 APnDP=1 时，可以访问 AP 寄存器的访问地址由以下两部分组成：

- A[3: 2]的值
- DP SELECT 寄存器的当前值

24.5 MCU 调试模块 (DBGMCU)

MCU 调试模块提供以下调试器协助功能：

- 支持低功耗模式
- 断点时定时器与看门狗的时钟控制

24.5.1 低功耗模式的调试支持

MCU 具有多种低功耗模式，能够关闭 CPU 时钟，降低 CPU 的功耗，通过执行 WFE 或 WFI 指令进入低功耗模式。CPU Free-Run 时钟 FCLK 与 AHB 总线时钟 HCLK 对于调试操作时必须的，不能关闭，同时 MCU 可以通过配置一些寄存器来改变低功耗模式特性，从而支持在低功耗模式下调试代码， 具体的配置如下。

- 当进入睡眠模式时，为了能够提供 HCLK 同 FCLK 相同的时钟，调试器必须先置位 DBG_CR 寄存器的 DBG_SLEEP 位。
- 当进入停机模式时，必须先配置 DBG_STOP 位，该操作会激活内部振荡器 HSI，从而为 FCLK 与 HCLK 提供时钟。

24.5.2 支持定时器、看门狗

当产生断点时，根据定时器和看门狗的应用不同来选择计数器的工作模式；

- 计数器可以选择继续计数，通常应用在输出 PWM 波控制电机
- 计数器可以选择停止计数，通常应用在看门狗计数

24.6 寄存器

24.6.1 寄存器总览

基地址：0x4001 3400

表 24-7 DBG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID 编码寄存器	0x4C553800
0x04	DBG_CR	DBG 控制寄存器	0x00000000

24.6.2 DBG_IDCODE ID 编码寄存器

偏移地址：0x00 （只支持 32 位访问，只读） 复位值：
0x4C553800

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

Bit	Field	Description
31:0	DEV_ID	设备识别编码 (Device Identifier) 只读寄存器，始终读为复位值

24.6.3 DBG_CR 控制寄存器

偏移地址：0x04 （只支持 32 位访问）
复位值：0x0000 0000 （POR 复位，不被系统复位所复位）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													DBG_TI M14_ST OP	DBG_TI M17_ST OP	DBG_TI M16_ST OP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DBG_TIM3_PWM_OFF	Res.	DBG_TIM1_PWM_OFF	DBG_TIM3_STOP	Res.	DBG_TIM1_STOP	Res.	DBG_IWDG_STOP	Res.	DBG_STOP_FOR_LD_O	Res.	DBG_STOP	DBG_SLEEP
rw		rw	rw		rw		rw		rw		rw	rw

Bit	Field	Description
31:19	Reserved	保留，必须保持复位值
18	DBG_TIM14_STOP	当内核进入调试状态时计数器 14 停止工作 (TIM14 Counter Stopped When Core is Halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
17	DBG_TIM17_STOP	当内核进入调试状态时计数器 17 停止工作 (TIM17 Counter Stopped When Core is Halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
16	DBG_TIM16_STOP	当内核进入调试状态时计数器 6 停止工作 (TIM16 Counter Stopped When Core is Halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
15	DBG_TIM3_PWM_OFF	TIM3 在调试模式下 PWM 输出全部为 0 0: TIM3 保持正常输出 1: TIM3 在 debug halt 时输出全部为 0
14	Reserved	保留，必须保持复位值
13	DBG_TIM1_PWM_OFF	TIM1 在调试模式下 PWM 输出全部为 0 0: TIM1 保持正常输出 1: TIM1 在 debug halt 时输出全部为 0
12	DBG_TIM3_STOP	当内核进入调试状态时计数器 3 停止工作 (TIM3 Counter Stopped When Core is Halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
11	Reserved	保留，必须保持复位值
10	DBG_TIM1_STOP	当内核进入调试状态时计数器 1 停止工作 (TIM1 Counter Stopped When Core is Halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
9	Reserved	保留，必须保持复位值
8	DBG_IWDG_STOP	当内核进入调试状态时看门狗停止工作 (Debug independent watchdog stopped when core is halted) 0: 看门狗计数器仍然正常工作 1: 看门狗计数器停止工作
7:4	Reserved	保留，必须保持复位值
3	DBG_STOP_FOR_LD_O	调试停机模式(Debug Stop mode) 0: 进入 stop 模式时, CORE 域切换到 LP 档位 1: 进入 stop 模式时, CORE 域仍处于 HP 档位, CPU 进入 deepsleep, CLK 关闭
2	Reserved	保留，必须保持复位值
1	DBG_STOP	调试停机模式(Debug Stop Mode For LDO) 0:(FCLK 关, HCLK 关) 在停机模式时, 时钟控制器禁止一切时钟(包括 HCLK 和 FCLK)。如果 HSI 时钟 18 分频后作为 sysclk 输入时钟, 当从 STOP 模式退出时, 时钟配置与复位之后的配置一致, 软件不需要重新配置时钟控制系统。如果是其他时钟作为 sysclk 输入时钟, 当从 STOP 模式退出时, 软件必须重新配置时钟控制系统。 1:(FCLK 关, HCLK 关) 在停机模式时, FCLK 和 HCLK 时钟都由内部振荡器提供。当退出停机模式时, 软件必须重新配置时钟控制系统。

Bit	Field	Description
0	DBG_SLEEP	调试睡眠模式 (Debug Sleep mode) 0: 在睡眠模式时, 时钟 FCLK 开启, FCLK 保持默认配置的系统时钟, HCLK 则关闭。睡眠模式不会复位配置好的时钟系统, 因此退出睡眠模式时, 软件不需重新配置系统时钟 1: 在睡眠模式时, FCLK 和 HCLK 时钟开启 都保持由原先配置好的系统时钟提供。

25 Device Electronic Signature 器件电子签名

25.1 简介

器件电子签名是存放在闪存存储器系统存储区域内用来唯一识别一颗微控制器的身份标识码（96 bits）。在任何情况下，用户都不可以修改器件电子签名。

可以通过软件读取器件电子签名，用来实现以下功能：

- 作为密码使用，在编程闪存时，通过器件电子签名与软件加密算法结合使用，提高代码在闪存中的安全性
- 作为序列号，用作终端应用中的序列号
- 激活安全引导流程

25.2 寄存器描述

基地址：0x1FFF F7E8

表 25-1 存储器容量寄存器概览

Offset	Acronym	Register Name	Reset
0x00	UID1	唯一标识码 1	0xFFFFFFFF
0x04	UID2	唯一标识码 2	0xFFFFFFFF
0x08	UID3	唯一标识码 3	0xFFFFFFFF

25.2.1 UID1 唯一标识码

偏移地址：0x00 复位值：其
值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (31: 16)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (15: 0)															
r															
Bit	Field		Description												
31: 0	U_ID (31: 0)		U_ID: 唯一身份标志 31: 0 位 (31: 0 unique ID bits)												

25.2.2 UID2 唯一标识码

偏移地址：0x04 复位值：其值

在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (63: 48)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (47: 32)															
r															
Bit	Field		Description												
31: 0	U_ID (63: 32)		U_ID: 唯一身份标志 63: 32 位 (63: 32 unique ID bits)												

25.2.3 UID3 唯一标识码

偏移地址：0x08 复位值：其值

在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (95: 80)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (79: 64)															
r															
Bit	Field		Description												
31: 0	U_ID (95: 64)		U_ID: 唯一身份标志 95 : 64 位 (95: 64 unique ID bits)												