



Preliminary

用户手册

User Manual

MM32G0160

基于Arm®Cortex®-M0内核的32位微控制器

版本：0.5

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.....	19
表格目录.....	25
1 系统和存储器架构.....	29
1.1 系统架构简介.....	29
1.2 系统总线.....	29
1.3 DMA 总线.....	30
1.4 USB FS 总线.....	30
1.5 总线矩阵.....	30
2 地址映射.....	31
2.1 存储器映像和寄存器编址.....	31
2.2 内置的 SRAM.....	33
2.3 闪存存储器概述.....	33
3 芯片特定配置.....	34
3.1 中断向量表.....	34
3.2 外部中断映射表.....	35
3.3 DMA 通道分配.....	36
3.4 ADC 通道分配.....	37
3.5 系统模块硬件互联.....	38
3.5.1 TIM 间的互联.....	38
3.5.2 TIM 和 EXTI 触发 ADC.....	38
3.5.3 COMP 触发 TIM.....	39
3.5.4 TIM 刹车事件的触发.....	39
3.5.5 从时钟到 TIM 的互联.....	40
3.6 启动配置.....	40
3.7 引导程序.....	41
4 Embedded FLASH 嵌入式闪存.....	42
4.1 简介.....	42
4.2 闪存构成与说明.....	42
4.2.1 闪存构成.....	42
4.2.2 选项字节说明.....	43
4.2.3 保密空间说明.....	45
4.3 闪存操作与流程.....	45
4.3.1 闪存读操作.....	45
4.3.2 闪存编程方式与操作流程.....	46
4.3.3 对闪存块操作限制的解除与使能.....	48
4.3.4 对选项字节区块操作限制的解除与使能.....	49
4.3.5 主闪存块擦除.....	51
4.3.6 主闪存块编程.....	53
4.3.7 选项字节区块擦除.....	54

4.3.8	选项字节区块编程.....	55
4.3.9	闪存保护	56
4.4	寄存器.....	58
4.4.1	寄存器总览	58
4.4.2	FLASH_ACR 闪存访问控制寄存器	59
4.4.3	FLASH_KEYR FPEC 键寄存器.....	59
4.4.4	FLASH_OPTKEYR 闪存 OPTKEY 寄存器.....	60
4.4.5	FLASH_SR 闪存状态寄存器	60
4.4.6	FLASH_CR 闪存控制寄存器.....	61
4.4.7	FLASH_AR 闪存地址寄存器	62
4.4.8	FLASH_OBR 选项字节寄存器.....	62
4.4.9	FLASH_WRPR 写保护寄存器.....	63
5	PWR 电源控制.....	64
5.1	供电系统.....	64
5.1.1	模拟模块供电.....	64
5.1.2	数字模块供电.....	65
5.1.3	VDD 域	65
5.1.4	1.5V 域.....	65
5.2	电源管理器	65
5.2.1	POR 上电复位和 PDR 掉电复位.....	65
5.2.2	PVD 可编程电压监测器	66
5.3	功耗控制.....	67
5.3.1	功耗控制概述.....	67
5.3.2	运行模式降低系统时钟	68
5.3.3	外设时钟的控制	69
5.3.4	Sleep Mode 睡眠模式.....	69
5.3.5	Stop Mode 停机模式.....	70
5.3.6	DeepStop Mode 深度停机模式.....	71
5.3.7	Standby Mode 待机模式.....	72
5.4	电源控制寄存器.....	73
5.4.1	PWR_CR 电源控制寄存器	74
5.4.2	PWR_CSR 电源控制/状态寄存器.....	75
5.4.3	PWR_CR1 电源控制寄存器 1	76
5.4.4	PWR_SR1 电源状态寄存器.....	77
5.4.5	PWR_SCR 电源状态清除寄存器.....	78
5.4.6	PWR_CFGR 电源配置寄存器	79
6	SYSCFG 系统控制器.....	80
6.1	简介	80
6.2	寄存器.....	80
6.2.1	寄存器总览	80
6.2.2	SYSCFG_CFGR 配置寄存器	80
6.2.3	SYSCFG_EXTICR1 外部中断配置寄存器 1.....	82
6.2.4	SYSCFG_EXTICR2 外部中断配置寄存器 2.....	83
6.2.5	SYSCFG_EXTICR3 外部中断配置寄存器 3.....	83

6.2.6	SYSCFG_EXTICR4 外部中断配置寄存器 4.....	84
6.2.7	SYSCFG_PADHYS PAD 配置寄存器.....	84
7	RCC 时钟和复位.....	86
7.1	复位单元.....	86
7.1.1	简介.....	86
7.1.2	功能框图.....	86
7.1.3	主要特征.....	86
7.1.4	功能描述.....	87
7.2	时钟单元.....	88
7.2.1	简介.....	88
7.2.2	功能框图.....	89
7.2.3	主要特征.....	89
7.2.4	功能描述.....	90
7.3	寄存器.....	98
7.3.1	寄存器总览.....	98
7.3.2	RCC_CR 时钟控制寄存器.....	98
7.3.3	RCC_CFGR 时钟配置寄存器.....	101
7.3.4	RCC_CIR 时钟中断寄存器.....	103
7.3.5	RCC_APB2RSTR APB2 外设复位寄存器.....	105
7.3.6	RCC_APB1RSTR APB1 外设复位寄存器.....	107
7.3.7	RCC_AHBENR AHB 外设时钟使能寄存器.....	109
7.3.8	RCC_APB2ENR APB2 外设时钟使能寄存器.....	110
7.3.9	RCC_APB1ENR APB1 外设时钟使能寄存器.....	112
7.3.10	RCC_BDCR 备份域控制寄存器.....	114
7.3.11	RCC_CSR 控制状态寄存器.....	115
7.3.12	RCC_AHBRSTR AHB 外设复位寄存器.....	117
7.3.13	RCC_CFGR2 时钟配置寄存器 2.....	118
7.3.14	RCC_SYSCFG 系统配置寄存器.....	120
7.3.15	RCC_ICSCR 内部时钟校准寄存器.....	120
7.3.16	RCC_PLL1CFGR PLL1 配置寄存器.....	121
7.3.17	RCC_PLL2CFGR PLL2 配置寄存器.....	122
8	RTC 实时时钟器.....	124
8.1	简介.....	124
8.2	主要特征.....	124
8.3	功能描述.....	125
8.3.1	功能框图.....	125
8.3.2	功能概述.....	125
8.3.3	模块复位.....	126
8.3.4	寄存器读取.....	126
8.3.5	寄存器配置.....	126
8.3.6	标志位产生.....	127
8.3.7	RTC 闹钟描述.....	128
8.3.8	RTC 外部中断事件输出.....	128
8.4	寄存器描述.....	128

8.4.1	寄存器总览	128
8.4.2	控制寄存器高位 (RTC_CRH)	129
8.4.3	控制寄存器低位 (RTC_CRL)	129
8.4.4	预分频装载寄存器高位 (RTC_PRLH)	131
8.4.5	预分频装载寄存器低位 (RTC_PRL)	131
8.4.6	预分频器分频因子寄存器高位 (RTC_DIVH)	131
8.4.7	预分频器分频因子寄存器低位 (RTC_DIVL)	132
8.4.8	计数器寄存器高位 (RTC_CNTH)	132
8.4.9	计数器寄存器低位 (RTC_CNTL)	132
8.4.10	闹钟寄存器高位 (RTC_ALRH)	133
8.4.11	闹钟寄存器低位 (RTC_ALRL)	133
8.4.12	毫秒闹钟寄存器高位 (RTC_MSRH)	133
8.4.13	毫秒闹钟寄存器低位 (RTC_MSRL)	134
9	CRC 循环冗余校验计算单元	135
9.1	简介	135
9.2	主要特征	135
9.3	功能描述	135
9.3.1	功能框图	135
9.3.2	功能概述	136
9.3.3	使用方法	136
9.4	寄存器	137
9.4.1	寄存器总览	137
9.4.2	CRC_DR CRC 数据寄存器	137
9.4.3	CRC_IDR CRC 独立数据寄存器	138
9.4.4	CRC_CR CRC 控制寄存器	138
9.4.5	CRC_MIR CRC 中间数据寄存器	139
10	CRS 时钟回馈系统	140
10.1	简介	140
10.2	主要特征	140
10.3	功能描述	141
10.3.1	功能框图	141
10.3.2	信号描述	141
10.3.3	功能概述	141
10.3.4	同步输入	141
10.3.5	频率误差测量	142
10.3.6	频率误差计算及自动校准	146
10.3.7	CRS 初始化及配置	146
10.4	CRS 低功耗模式	147
10.5	中断	147
10.6	寄存器	148
10.6.1	寄存器总览	148
10.6.2	CRS_CR 控制寄存器	148
10.6.3	CRS_CFGR 配置寄存器	149
10.6.4	CRS_ISR 中断状态寄存器	150

10.6.5	CRS_ICR 中断标志清除寄存器	152
11	HWDIV 硬件除法器	153
11.1	简介	153
11.2	主要特征	153
11.3	功能描述	153
11.4	寄存器描述	153
11.4.1	寄存器总览	153
11.4.2	HWDIV_DVDR 被除数寄存器	154
11.4.3	HWDIV_DVSR 除数寄存器	154
11.4.4	HWDIV_QUOTR 商寄存器	154
11.4.5	HWDIV_RMDR 余数寄存器	155
11.4.6	HWDIV_SR 状态寄存器	155
11.4.7	HWDIV_CR 控制寄存器	155
12	GPIO 通用端口	157
12.1	简介	157
12.2	主要特征	157
12.3	功能描述	158
12.3.1	功能框图	158
12.3.2	GPIO 端口配置	158
12.3.3	复用功能	159
12.3.4	GPIO 锁定机制	159
12.3.5	输入配置	160
12.3.6	输出配置	161
12.3.7	复用功能配置	161
12.3.8	模拟输入配置	162
12.3.9	外部时钟复用 GPIO 端口	163
12.3.10	SWD 复用功能重映射	163
12.4	GPIO 寄存器描述	163
12.4.1	寄存器总览	163
12.4.2	GPIOx_CRL 端口配置低寄存器	164
12.4.3	GPIOx_CRH 端口配置高寄存器	164
12.4.4	GPIOx_IDR 端口输入数据寄存器	165
12.4.5	GPIOx_ODR 端口输出数据寄存器	166
12.4.6	GPIOx_BSRR 端口设置/清除寄存器	166
12.4.7	GPIOx_BRR 端口位清除寄存器	167
12.4.8	GPIOx_LCKR 端口配置锁定寄存器	167
12.4.9	GPIOx_DCR 端口输出开漏控制寄存器	168
12.4.10	GPIOx_AFR1 端口复用功能低位寄存器	168
12.4.11	GPIOx_AFRH 端口复用功能高位寄存器	169
13	EXTI 中断和事件	170
13.1	简介	170
13.2	主要特征	170
13.3	功能描述	170
13.3.1	功能框图	170

13.3.2	中断和异常向量	171
13.3.3	唤醒事件管理	171
13.3.4	中断功能描述	171
13.3.5	硬件中断输出	171
13.3.6	硬件事件输出	172
13.3.7	软件中断与事件输出	172
13.3.8	外部中断映射	172
13.4	寄存器描述	172
13.4.1	寄存器总览	172
13.4.2	中断屏蔽寄存器 (EXTI_IMR)	172
13.4.3	事件屏蔽寄存器 (EXTI_EMR)	173
13.4.4	上升沿触发选择寄存器 (EXTI_RTSR)	173
13.4.5	下降沿触发选择寄存器 (EXTI_FTISR)	174
13.4.6	软件中断事件寄存器 (EXTI_SWIER)	174
13.4.7	软件中断事件挂起寄存器 (EXTI_PR)	175
14	DMA 直接存储器访问控制器	176
14.1	简介	176
14.2	功能框图	176
14.3	主要特征	176
14.4	中断	177
14.5	DMA	177
14.5.1	DMA 请求映像	177
14.6	功能描述	178
14.6.1	DMA 处理	178
14.6.2	仲裁器	179
14.6.3	DMA 通道	179
14.6.4	可编程的数据传输宽度, 对齐方式和数据大小端	180
14.6.5	错误管理	182
14.7	DMA 寄存器描述	182
14.7.1	DMA_ISR DMA 中断状态寄存器	183
14.7.2	DMA_IFCR DMA 中断标志清除寄存器	183
14.7.3	DMA_CCRx DMA 通道 x 配置寄存器 (x=1~7)	184
14.7.4	DMA_CNDTRx DMA 通道 x 传输数量寄存器 (x=1~7)	186
14.7.5	DMA_CPARx DMA 通道 x 外设地址寄存器 (x=1~7)	187
14.7.6	DMA_CMARx DMA 通道 x 存储器地址寄存器 (x = 1~7)	187
15	COMP 比较器	189
15.1	简介	189
15.2	功能框图	189
15.3	主要特征	189
15.4	功能描述	190
15.4.1	比较器时钟和复位	190
15.4.2	比较器开关控制	190
15.4.3	比较器输入和输出	190
15.4.4	比较器通道选择	190

15.4.5	中断和唤醒	191
15.4.6	功耗模式	191
15.4.7	比较器锁定机制	191
15.4.8	迟滞电压	192
15.5	比较器寄存器描述	192
15.5.1	比较器控制状态寄存器 (COMPx_CSR) (x=1, 2)	192
15.5.2	比较器外部参考电压寄存器 (COMP_CRV)	195
15.5.3	比较器轮询寄存器 (COMPx_POLL) (x=1, 2)	196
16	ADC 模拟/数字转换	198
16.1	简介	198
16.2	功能框图	198
16.3	主要特征	199
16.4	中断	199
16.5	DMA	200
16.6	功能描述	200
16.6.1	时钟	200
16.6.2	数据补偿	200
16.6.3	数据对齐	201
16.6.4	可编程分辨率	201
16.6.5	可编程采样时间	202
16.6.6	数据通道寄存器	202
16.6.7	通道选择	202
16.7	ADC 开关	202
16.7.1	普通通道转换	203
16.7.2	任意通道转换	206
16.7.3	注入通道转换	208
16.7.4	ADC 触发信号	211
16.7.5	模拟看门狗	212
16.7.6	内部温度传感器	212
16.7.7	内部电压传感器	213
16.8	寄存器	213
16.8.1	寄存器总览	213
16.8.2	ADC_ADDDATA 数据寄存器	214
16.8.3	ADC_ADCFG 配置寄存器	215
16.8.4	ADC_ADCR 控制寄存器	216
16.8.5	ADC_ADCHS 通道选择寄存器	219
16.8.6	ADC_ADCMPR 模拟看门狗比较寄存器	221
16.8.7	ADC_ADSTA 状态寄存器	221
16.8.8	ADC_ADDRn 通道数据寄存器 (n=0~15)	222
16.8.9	ADC_ADSTA_EXT 扩展状态寄存器	223
16.8.10	ADC_CHANY0 任意通道的通道选择寄存器 0	224
16.8.11	ADC_CHANY1 任意通道的通道选择寄存器 1	225
16.8.12	ADC_ANY_CFG 任意通道配置寄存器	227
16.8.13	ADC_ANY_CR 任意通道控制寄存器	227

16.8.14	ADC_SMPR1 采样配置寄存器 1	230
16.8.15	ADC_SMPR2 采样配置寄存器 2	231
16.8.16	ADC_JOFRn 注入通道数据补偿寄存器 (n=0~3)	232
16.8.17	ADC_JSQR 注入通道序列寄存器	233
16.8.18	ADC_JADDATA 注入数据寄存器	233
16.8.19	ADC_JDRn 注入通道数据寄存器 (n=0~3)	234
16.8.20	ADC_LDATA 上一次转换数据寄存器	235
17	TIM1 高级定时器	237
17.1	简介	237
17.2	功能框图	237
17.3	主要特征	237
17.4	功能描述	238
17.4.1	时钟	238
17.4.2	重复计数器	243
17.4.3	输入捕获	245
17.4.4	比较输出	247
17.4.5	从模式	259
17.4.6	定时器同步	263
17.4.7	定时器异或	266
17.4.8	调试模式	267
17.4.9	中断	268
17.4.10	DMA	268
17.5	寄存器描述	269
17.5.1	TIM1_CR1 控制寄存器 1	269
17.5.2	TIM1_CR2 控制寄存器 2	271
17.5.3	TIM1_SMCR 从模式控制寄存器	272
17.5.4	TIM1_DIER DMA/中断使能寄存器	275
17.5.5	TIM1_SR 状态寄存器	277
17.5.6	TIM1_EGR 事件产生寄存器	279
17.5.7	TIM1_CCMR1 捕获/比较模式寄存器 1	280
17.5.8	TIM1_CCMR2 捕获/比较模式寄存器 2	284
17.5.9	TIM1_CCER 捕获/比较使能寄存器	288
17.5.10	TIM1_CNT 计数器	290
17.5.11	TIM1_PSC 预分频器	291
17.5.12	TIM1_ARR 自动预装载寄存器	291
17.5.13	TIM1_RCR 重复计数寄存器	291
17.5.14	TIM1_CCR1 捕获/比较寄存器 1	292
17.5.15	TIM1_CCR2 捕获/比较寄存器 2	292
17.5.16	TIM1_CCR3 捕获/比较寄存器 3	293
17.5.17	TIM1_CCR4 捕获/比较寄存器 4	293
17.5.18	TIM1_BDTR 刹车和死区寄存器	293
17.5.19	TIM1_DCR DMA 控制寄存器	295
17.5.20	TIM1_DMAR 连续模式 DMA 地址寄存器	296
17.5.21	TIM1_CCMR3 捕获/比较模式寄存器 3	297

17.5.22	TIM1_CCR5 捕获/比较寄存器 5	297
17.5.23	TIM1_PDER PWM 移相/DMA repeat 更新请求使能寄存器	298
17.5.24	TIM1_CCRxFALL PWM 移相递减计数捕获/比较寄存器	298
17.5.25	TIM1_BKINF 刹车输入滤波寄存器	299
18	TIM2 通用定时器	301
18.1	简介	301
18.2	功能框图	301
18.3	主要特征	301
18.4	功能描述	302
18.4.1	时钟	302
18.4.2	输入捕获	307
18.4.3	比较输出	309
18.4.4	从模式	316
18.4.5	定时器同步	320
18.4.6	定时器异或	320
18.4.7	调试模式	321
18.4.8	中断	321
18.4.9	DMA	321
18.5	寄存器描述	322
18.5.1	TIM2_CR1 控制寄存器 1	322
18.5.2	TIM2_CR2 控制寄存器 2	324
18.5.3	TIM2_SMCR 从模式控制寄存器	325
18.5.4	TIM2_DIER DMA/中断使能寄存器	328
18.5.5	TIM2_SR 状态寄存器	329
18.5.6	TIM2_EGR 事件产生寄存器	331
18.5.7	TIM2_CCMR1 捕获/比较模式寄存器 1	332
18.5.8	TIM2_CCMR2 捕获/比较模式寄存器 2	335
18.5.9	TIM2_CCER 捕获/比较使能寄存器	339
18.5.10	TIM2_CNT 计数器	341
18.5.11	TIM2_PSC 预分频器	341
18.5.12	TIM2_ARR 自动预装载寄存器	342
18.5.13	TIM2_CCR1 捕获/比较寄存器 1	342
18.5.14	TIM2_CCR2 捕获/比较寄存器 2	343
18.5.15	TIM2_CCR3 捕获/比较寄存器 3	343
18.5.16	TIM2_CCR4 捕获/比较寄存器 4	344
18.5.17	TIM2_DCR DMA 控制寄存器	344
18.5.18	TIM2_DMAR 连续模式 DMA 地址寄存器	345
18.5.19	TIM2_OR 输入选项寄存器	345
19	TIM3 通用定时器	347
19.1	简介	347
19.2	功能框图	347
19.3	主要特征	347
19.4	功能描述	348
19.4.1	时钟	348

19.4.2	输入捕获	353
19.4.3	比较输出	355
19.4.4	从模式	362
19.4.5	定时器同步	366
19.4.6	定时器异或	366
19.4.7	调试模式	367
19.4.8	中断	367
19.4.9	DMA	367
19.5	寄存器描述	368
19.5.1	TIM3_CR1 控制寄存器 1	368
19.5.2	TIM3_CR2 控制寄存器 2	370
19.5.3	TIM3_SMCR 从模式控制寄存器	371
19.5.4	TIM3_DIER DMA/中断使能寄存器	374
19.5.5	TIM3_SR 状态寄存器	375
19.5.6	TIM3_EGR 事件产生寄存器	377
19.5.7	TIM3_CCMR1 捕获/比较模式寄存器 1	378
19.5.8	TIM3_CCMR2 捕获/比较模式寄存器 2	381
19.5.9	TIM3_CCER 捕获/比较使能寄存器	385
19.5.10	TIM3_CNT 计数器	387
19.5.11	TIM3_PSC 预分频器	387
19.5.12	TIM3_ARR 自动预装载寄存器	388
19.5.13	TIM3_CCR1 捕获/比较寄存器 1	388
19.5.14	TIM3_CCR2 捕获/比较寄存器 2	389
19.5.15	TIM3_CCR3 捕获/比较寄存器 3	389
19.5.16	TIM3_CCR4 捕获/比较寄存器 4	389
19.5.17	TIM3_DCR DMA 控制寄存器	390
19.5.18	TIM3_DMAR 连续模式 DMA 地址寄存器	390
19.5.19	TIM3_OR 输入选项寄存器	391
20	TIM14 基本定时器	392
20.1	简介	392
20.2	功能框图	392
20.3	主要特征	392
20.4	功能描述	393
20.4.1	时钟	393
20.4.2	输入捕获	394
20.4.3	比较输出	395
20.4.4	调试模式	398
20.4.5	中断	398
20.4.6	DMA	399
20.5	寄存器描述	399
20.5.1	TIM14_CR1 控制寄存器 1	399
20.5.2	TIM14_DIER DMA/中断使能寄存器	400
20.5.3	TIM14_SR 状态寄存器	401
20.5.4	TIM14_EGR 事件产生寄存器	402

20.5.5	TIM14_CCMR1 捕获/比较模式寄存器 1.....	403
20.5.6	TIM14_CCER 捕获/比较使能寄存器.....	405
20.5.7	TIM14_CNT 计数器.....	406
20.5.8	TIM14_PSC 预分频器.....	406
20.5.9	TIM14_ARR 自动预装载寄存器.....	407
20.5.10	TIM14_CCR1 捕获/比较寄存器 1.....	407
20.5.11	TIM14_BDTR 刹车和死区寄存器.....	408
21	TIM16/17 基本定时器.....	409
21.1	简介.....	409
21.2	功能框图.....	409
21.3	主要特征.....	409
21.4	功能描述.....	410
21.4.1	时钟.....	410
21.4.2	输入捕获.....	411
21.4.3	比较输出.....	412
21.4.4	定时器同步.....	420
21.4.5	调试模式.....	420
21.4.6	中断.....	420
21.4.7	DMA.....	420
21.5	寄存器描述.....	420
21.5.1	TIMx_CR1 控制寄存器 1.....	421
21.5.2	TIMx_CR2 控制寄存器 2.....	422
21.5.3	TIMx_DIER DMA/中断使能寄存器.....	423
21.5.4	TIMx_SR 状态寄存器.....	424
21.5.5	TIMx_EGR 事件产生寄存器.....	425
21.5.6	TIMx_CCMR1 捕获/比较模式寄存器 1.....	426
21.5.7	TIMx_CCER 捕获/比较使能寄存器.....	429
21.5.8	TIMx_CNT 计数器.....	430
21.5.9	TIMx_PSC 预分频器.....	431
21.5.10	TIMx_ARR 自动预装载寄存器.....	431
21.5.11	TIMx_CCR1 捕获/比较寄存器 1.....	431
21.5.12	TIMx_BDTR 刹车和死区寄存器.....	432
21.5.13	TIMx_BKINF 刹车滤波寄存器.....	434
22	LPTIM 低功耗定时器.....	436
22.1	简介.....	436
22.2	功能框图.....	436
22.3	主要特征.....	436
22.4	中断.....	437
22.5	功能描述.....	437
22.5.1	计数模式.....	437
22.5.2	普通定时器.....	437
22.5.3	Trigger 脉冲触发计数.....	437
22.5.4	TimeOut 模式.....	438
22.5.5	PWM 输出.....	438

22.6	寄存器描述	438
22.6.1	寄存器概览	438
22.6.2	LPT_CFG 配置寄存器	439
22.6.3	LPT_IE 中断使能寄存器	440
22.6.4	LPT_IF 中断标志寄存器	440
22.6.5	LPT_CTRL 控制寄存器	441
22.6.6	LPT_CNT 计数寄存器	441
22.6.7	LPT_CMP 比较值寄存器	441
22.6.8	LPT_TARGET 目标值寄存器	442
23	WWDG 窗口看门狗	443
23.1	简介	443
23.2	功能框图	443
23.3	主要特征	443
23.4	功能描述	444
23.5	窗口看门狗超时时间	445
23.6	寄存器描述	445
23.6.1	控制寄存器 (WWDG_CR)	446
23.6.2	配置寄存器 (WWDG_CFGR)	446
23.6.3	状态寄存器 (WWDG_SR)	447
24	IWDG 独立看门狗	448
24.1	简介	448
24.2	功能框图	449
24.3	主要特征	449
24.4	功能描述	450
24.4.1	流程框图	451
24.4.2	独立看门狗超时时间	451
24.4.3	中断	452
24.5	寄存器描述	452
24.5.1	寄存器总览	452
24.5.2	IWDG_KR 键寄存器	452
24.5.3	IWDG_PR 预分频寄存器	453
24.5.4	IWDG_RLR 重装载寄存器	453
24.5.5	IWDG_SR 状态寄存器	454
24.5.6	IWDG_CR 控制寄存器	455
24.5.7	IWDG_IGEN 中断生成寄存器	455
24.5.8	IWDG_CNT 计数寄存器	456
25	UART 通用异步收发器	457
25.1	简介	457
25.2	功能框图	457
25.3	主要特征	457
25.4	功能概述	458
25.4.1	UART 特性描述	459
25.4.2	分数波特率发生器	459
25.4.3	采样	459

25.4.4	容忍度.....	460
25.4.5	校验控制.....	460
25.4.6	发送器.....	460
25.4.7	接收器.....	462
25.4.8	自动波特率检测.....	463
25.4.9	九位数据通信.....	464
25.4.10	多处理器通信.....	464
25.4.11	单线半双工通信.....	466
25.4.12	智能卡.....	466
25.4.13	红外 IrDA 功能.....	468
25.5	中断.....	469
25.6	DMA.....	470
25.7	寄存器.....	470
25.7.1	寄存器总览.....	470
25.7.2	UART_TDR 发送数据寄存器.....	471
25.7.3	UART_RDR 接收数据寄存器.....	471
25.7.4	UART_CSR 当前状态寄存器.....	471
25.7.5	UART_ISR 中断状态寄存器.....	472
25.7.6	UART_IER 中断使能寄存器.....	474
25.7.7	UART_ICR 中断清除寄存器.....	475
25.7.8	UART_GCR 全局控制寄存器.....	476
25.7.9	UART_CCR 通用空制寄存器.....	477
25.7.10	UART_BRR 波特率寄存器.....	479
25.7.11	UART_FRA 分数波特率寄存器.....	479
25.7.12	UART_RXADDR 接收地址寄存器.....	480
25.7.13	UART_RXMASK 接收掩码寄存器.....	480
25.7.14	UART_SCR SCR 寄存器.....	481
25.7.15	UART_IDLR IDLE 数据长度寄存器.....	481
25.7.16	UART_ABRCR 自动波特率寄存器.....	482
25.7.17	UART_IRDA 红外功能控制寄存器.....	483
26	LPUART 低功耗通用异步收发器.....	484
26.1	简介.....	484
26.2	功能框图.....	484
26.3	主要特征.....	484
26.4	中断.....	485
26.5	DMA.....	486
26.6	功能描述.....	486
26.6.1	特性描述.....	486
26.6.2	接收时序.....	487
26.6.3	发送时序.....	487
26.6.4	调制寄存器建议配置.....	487
26.6.5	奇偶校验控制.....	488
26.6.6	发送器.....	488
26.6.7	接收器.....	489

26.6.8	休眠模式下的数据接收唤醒.....	490
26.6.9	低功耗模式下的数据自动收发.....	490
26.7	寄存器描述.....	491
26.7.1	寄存器概览.....	491
26.7.2	LPUBAUD 波特率寄存器.....	491
26.7.3	MODU 波特率调制控制寄存器.....	492
26.7.4	LPUIF 中断标志寄存器.....	493
26.7.5	LPUSTA 状态寄存器.....	493
26.7.6	LPUCON 控制寄存器.....	495
26.7.7	LPUEN 发送接收使能寄存器.....	496
26.7.8	LPURXD 接收数据寄存器.....	497
26.7.9	LPUTXD 发送数据寄存器.....	497
26.7.10	COMPARE 数据匹配寄存器.....	498
26.7.11	WKCKE 唤醒寄存器.....	498
27	SPI_I2S 串行外设接口.....	500
27.1	SPI_I2S 功能框图.....	500
27.2	SPI_I2S 简述.....	500
27.3	SPI 功能描述.....	501
27.3.1	概述.....	501
27.3.2	SPI 主要特征.....	503
27.3.3	SPI 从模式.....	504
27.3.4	SPI 主模式.....	505
27.3.5	波特率设置.....	506
27.3.6	中断.....	506
27.3.7	DMA 传输.....	507
27.4	I2S 功能描述.....	507
27.4.1	I2S 主要特征.....	507
27.4.2	I2S 总线接口.....	508
27.4.3	数据格式.....	508
27.4.4	通信标准.....	508
27.4.5	从模式.....	511
27.4.6	主模式.....	511
27.4.7	时钟预分频器.....	512
27.4.8	中断.....	514
27.4.9	DMA 传输.....	514
27.5	寄存器.....	515
27.5.1	寄存器总览.....	515
27.5.2	SPI_I2S_TXREG 发送数据寄存器.....	515
27.5.3	SPI_I2S_RXREG 接收数据寄存器.....	515
27.5.4	SPI_I2S_CSTAT 当前状态寄存器.....	516
27.5.5	SPI_I2S_INTSTAT 中断状态寄存器.....	517
27.5.6	SPI_I2S_INTEN 中断使能寄存器.....	518
27.5.7	SPI_I2S_INTCLR 中断清除寄存器.....	519
27.5.8	SPI_I2S_GCTL 全局控制寄存器.....	520

27.5.9	SPI_I2S_CCTL 通用控制寄存器.....	522
27.5.10	SPI_I2S_SPBRG 波特率发生器.....	523
27.5.11	SPI_I2S_RXDNR 接收数据个数寄存器.....	524
27.5.12	SPI_I2S_NSSR 从机片选寄存器.....	524
27.5.13	SPI_I2S_EXTCTL 数据控制寄存器.....	525
27.5.14	SPI_I2S_I2SCFGR I2S 配置寄存器.....	526
28	I2C 内部集成电路接口.....	528
28.1	简介.....	528
28.2	主要特征.....	528
28.3	功能描述.....	529
28.3.1	功能框图.....	529
28.3.2	信号描述.....	530
28.3.3	I2C 协议.....	530
28.3.4	工作模式.....	537
28.3.5	中断.....	542
28.3.6	DMA.....	543
28.4	寄存器.....	543
28.4.1	寄存器总览.....	543
28.4.2	I2C_CR 控制寄存器.....	544
28.4.3	I2C_TAR 目标地址寄存器.....	546
28.4.4	I2C_SAR 从地址寄存器.....	547
28.4.5	I2C_DR 数据命令寄存器.....	547
28.4.6	I2C_SSHR 标准模式时钟高电平计数寄存器.....	548
28.4.7	I2C_SSLR 标准模式时钟低电平计数寄存器.....	548
28.4.8	I2C_FSHR 快速模式时钟高电平计数寄存器.....	549
28.4.9	I2C_FSLR 快速模式时钟低电平计数寄存器.....	549
28.4.10	I2C_ISR 中断状态寄存器.....	549
28.4.11	I2C_IMR 中断屏蔽寄存器.....	550
28.4.12	I2C_RAWISR RAW 中断寄存器.....	550
28.4.13	I2C_RXTLR 接收阈值.....	552
28.4.14	I2C_TXTLR 发送阈值.....	552
28.4.15	I2C_ICR 组合和独立中断清除寄存器.....	553
28.4.16	I2C_RX_UNDER 清除 RX_UNDER 中断寄存器.....	553
28.4.17	I2C_RX_OVER 清除 RX_OVER 中断寄存器.....	553
28.4.18	I2C_TX_OVER 清除 TX_OVER 中断寄存器.....	554
28.4.19	I2C_RD_REQ 清除 RD_REQ 中断寄存器.....	554
28.4.20	I2C_TX_ABRT 清除 TX_ABRT 中断寄存器.....	555
28.4.21	I2C_RX_DONE 清除 RX_DONE 中断寄存器.....	555
28.4.22	I2C_ACTIV 清除 ACTIVITY 中断寄存器.....	555
28.4.23	I2C_STOP 清除 STOP_DET 中断寄存器.....	556
28.4.24	I2C_START 清除 START_DET 中断寄存器.....	556
28.4.25	I2C_GC 清除 GEN_CALL 中断寄存器.....	557
28.4.26	I2C_ENR 使能寄存器.....	557
28.4.27	I2C_SR 状态寄存器.....	558

28.4.28	I2C_TXFLR 发送缓冲级别寄存器	559
28.4.29	I2C_RXFLR 接收缓冲级别寄存器	559
28.4.30	I2C_HOLD SDA 保持时间寄存器	559
28.4.31	I2C_DMA DMA 控制寄存器	560
28.4.32	I2C_SETUP SDA 建立时间寄存器	560
28.4.33	I2C_GCR 广播呼叫 ACK 寄存器	561
28.4.34	I2C_SLVMASK 从地址掩码寄存器	561
28.4.35	I2C_SLVRCVADDR 从器件接收地址寄存器	562
29	USB_FS 全速 USB 控制器	563
29.1	简介	563
29.2	主要特征	563
29.3	功能描述	564
29.3.1	功能框图	564
29.3.2	信号描述	564
29.3.3	USB_FS 典型硬件配置	564
29.3.4	USB_FS 软件编程接口	565
29.3.5	中断	569
29.4	寄存器	570
29.4.1	寄存器总览	570
29.4.2	USB_FS_INTSTAT 中断状态寄存器	570
29.4.3	USB_FS_INTENB 中使能态寄存器	571
29.4.4	USB_FS_ERRSTAT 错误中断状态寄存器	572
29.4.5	USB_FS_ERRENB 错误中断使能寄存器	572
29.4.6	USB_FS_STAT 状态寄存器	573
29.4.7	USB_FS_CTL 控制寄存器	574
29.4.8	USB_FS_ADDR 地址寄存器	574
29.4.9	USB_FS_BDTPAGE1 缓冲区描述符表页寄存器 1	575
29.4.10	USB_FS_FRMNUML 低位帧数寄存器	575
29.4.11	USB_FS_FRMNUMH 高位帧数寄存器	575
29.4.12	USB_FS_BDTPAGE2 缓冲区描述符表页寄存器 2	576
29.4.13	USB_FS_BDTPAGE3 缓冲区描述符表页寄存器 3	576
29.4.14	USB_FS_EPCTL0~7 端点控制寄存器 0~7	576
29.4.15	USB_CTRL USB 控制寄存器	577
30	FlexCAN 控制器局域网	578
30.1	简介	578
30.2	主要特征	578
30.3	功能描述	579
30.3.1	功能框图	579
30.3.2	工作模式	581
30.3.3	传输流程	583
30.3.4	仲裁过程	584
30.3.5	接收流程	586
30.3.6	匹配过程	588
30.3.7	转移流程	591

30.3.8	数据一致性	592
30.3.9	增强型 Rx FIFO	594
30.3.10	传统型 Rx FIFO	598
30.3.11	CAN 协议.....	599
30.3.12	时钟限制	612
30.3.13	复位	614
30.3.14	初始化.....	614
30.3.15	中断	615
30.4	寄存器/内存映射	616
30.4.1	寄存器/内存映射总览.....	617
30.4.2	CAN_MCR 模块配置寄存器	618
30.4.3	CAN_CTRL1 控制寄存器 1.....	621
30.4.4	CAN_TIMER 自由运行计数器寄存器	624
30.4.5	CAN_RXMGMASK Rx 邮箱全局掩码寄存器.....	625
30.4.6	CAN_RX14MASK Rx 14 掩码寄存器	626
30.4.7	CAN_RX15MASK Rx 15 掩码寄存器	626
30.4.8	CAN_ECR 错误计数器寄存器	627
30.4.9	CAN_ESR1 错误和状态寄存器 1.....	628
30.4.10	CAN_IMASK1 中断掩码寄存器 1	631
30.4.11	CAN_IFLAG1 中断标志寄存器 1	632
30.4.12	CAN_CTRL2 控制寄存器 2.....	634
30.4.13	CAN_ESR2 错误和状态 2 寄存器.....	637
30.4.14	CAN_CRCCR CRC 寄存器.....	638
30.4.15	CAN_RXFGMASK 传统型 Rx FIFO 全局掩码寄存器.....	638
30.4.16	CAN_RXFIR 传统型 Rx FIFO 信息寄存器.....	639
30.4.17	CAN_CBT CAN 位时序寄存器	640
30.4.18	CAN_RXIMRn Rx 私有掩码寄存器	641
30.4.19	CAN_FDCTRL CAN FD 控制寄存器	641
30.4.20	CAN_FDCBT CAN FD 位时序寄存器.....	643
30.4.21	CAN_FDCRC CAN FD CRC 寄存器	644
30.4.22	CAN_ERFCR 增强型 Rx FIFO 控制寄存器	644
30.4.23	CAN_ERFIER 增强型 Rx FIFO 中断使能寄存器	646
30.4.24	CAN_ERFSR 增强型 Rx FIFO 状态寄存器	646
30.4.25	CAN_ERFFELn 增强型 Rx FIFO 过滤元素寄存器	647
30.4.26	报文缓冲区结构	648
30.4.27	MB 内存映射	651
30.4.28	传统型 Rx FIFO 结构	652
30.4.29	增强型 Rx FIFO 结构	654
31	BKP 备份寄存器	657
31.1	简介	657
31.2	主要特征	657
31.3	功能描述	657
31.3.1	时钟校准	657
31.3.2	侵入检测	657

31.4	寄存器.....	658
31.4.1	寄存器总览.....	658
31.4.2	时钟校准寄存器 (BKP_RTCCR).....	658
31.4.3	备份控制寄存器 (BKP_CR).....	659
31.4.4	备份控制状态寄存器 (BKP_CSR).....	659
31.4.5	备份数据寄存器 n (BKP_DRn) (n = 1 .. 10).....	660
32	DBG 调试支持.....	661
32.1	简介.....	661
32.2	功能描述.....	661
32.2.1	功能框图.....	661
32.2.2	SWD 内部上拉与下拉.....	662
32.2.3	SWJ 调试端口.....	662
32.3	ID 代码和锁定机制.....	662
32.3.1	微控制器设备 ID 编码.....	662
32.3.2	Cortex JEDEC-106 ID 编码.....	662
32.4	SW 调试端口.....	663
32.4.1	SW 协议介绍.....	663
32.4.2	SW 协议序列.....	663
32.4.3	SW-DP 状态机 (Reset, Idle states, ID code).....	664
32.4.4	DP 和 AP 读 / 写访问.....	664
32.4.5	SW-DP 寄存器.....	664
32.4.6	SW-AP 寄存器.....	665
32.5	MCU 调试模块 (DBGMCU).....	665
32.5.1	低功耗模式的调试支持.....	665
32.5.2	支持定时器、看门狗.....	666
32.6	寄存器.....	666
32.6.1	寄存器总览.....	666
32.6.2	DBG_IDCODE ID 编码寄存器.....	666
32.6.3	DBG_CR 控制寄存器.....	666
33	Device Electronic Signature 器件电子签名.....	669
33.1	简介.....	669
33.2	寄存器描述.....	669
33.2.1	UID1 唯一标识码.....	669
33.2.2	UID2 唯一标识码.....	670
33.2.3	UID3 唯一标识码.....	670
34	修订记录.....	671

附图目录

图 1-1 系统架构框图.....	29
图 4-1 ISP 方式编程流程图.....	47
图 4-2 IAP 编程流程.....	48
图 5-1 电源控制功能框图.....	64
图 5-2 上电复位和掉电复位波形图.....	66
图 5-3 PVD 阈值波形图.....	67
图 7-1 复位功能框图.....	86
图 7-2 时钟树.....	89
图 7-3 高速晶振/陶瓷谐振器.....	90
图 7-4 外部高速输入时钟.....	90
图 8-1 RTC 功能框图.....	125
图 8-2 RTC 秒和闹钟波形图示例, RTC_PRL = 0004, RTC_ALR = 002.....	127
图 8-3 RTC 溢出波形图示例, RTC_PRL = 0004.....	127
图 9-1 CRC 功能框图.....	135
图 10-1 CRS 功能框图.....	141
图 10-2 CRS 计数器状态图.....	146
图 12-1 标准 I/O 端口.....	158
图 12-2 输入浮空/上拉/下拉配置.....	160
图 12-3 输出配置.....	161
图 12-4 复用功能配置.....	162
图 12-5 模拟输入.....	163
图 13-1 EXTI 结构框图.....	170
图 14-1 DMA 功能框图.....	176
图 15-1 比较器框图.....	189
图 15-2 比较器的迟滞.....	192
图 16-1 ADC 系统框图.....	198
图 16-2 数据对齐方式.....	201
图 16-3 单次转换模式时序图.....	203
图 16-4 单周期扫描下使能通道转换时序图 (通道方向从高到低).....	204
图 16-5 单周期扫描下使能通道转换时序图 (通道方向从低到高).....	204
图 16-6 连续扫描模式使能通道转换时序图 (通道方向由低到高).....	205
图 16-7 连续扫描模式使能通道转换时序图 (通道方向由高到低).....	205
图 16-8 单次转换模式下通道转换时序图.....	206
图 16-9 单周期扫描下通道转换时序图.....	207
图 16-10 连续扫描模式下通道转换时序图.....	208
图 16-11 连续扫描模式下动态更新配置时序图.....	208
图 16-12 自动注入通道转换单周期扫描通道转换时序图.....	209
图 16-13 自动注入通道转换连续转换时序图.....	210
图 16-14 任意通道转换时事件注入通道转换时序图 1.....	211
图 16-15 任意通道转换时事件注入通道转换时序图 2.....	211

图 17-1 TIM1 结构图.....	237
图 17-2 时钟选择	239
图 17-3 外部时钟模式 1 下的控制电路	240
图 17-4 外部时钟模式 2 下的控制电路	240
图 17-5 自动预装载.....	241
图 17-6 递增计数模式 (UDIS=0)	241
图 17-7 递增计数模式 (UDIS=1 禁止产生更新事件)	242
图 17-8 递减计数模式 (UDIS=0)	242
图 17-9 递减计数模式 (UDIS=1 禁止产生更新事件)	242
图 17-10 中央计数模式 (UDIS=0)	243
图 17-11 中央计数模式 (UDIS=1 禁止产生更新事件)	243
图 17-12 中央对齐模式重复计数时序图.....	244
图 17-13 边沿对齐模式递增计数时序图.....	244
图 17-14 边沿对齐模式递减计数时序图.....	245
图 17-15 TIM1 输入捕获结构图	245
图 17-16 PWM 输入模式时序	247
图 17-17 比较输出部分结构图	248
图 17-18 比较输出模式, OC1 信号在匹配时翻转	249
图 17-19 边沿对齐递增计数时 PWM 模式 1 的波形	250
图 17-20 边沿对齐递减计数时 PWM 模式 1 的波形	251
图 17-21 中央对齐 PWM 模式 1 的波形.....	252
图 17-22 移相功能示意图	252
图 17-23 使用 COM 事件产生六步 PWM (OSSR = 1)	253
图 17-24 死区插入	254
图 17-25 响应刹车的输出 (OISx=0, OISxN=0)	257
图 17-26 响应刹车的输出 (OISx=0, OISxN=1)	257
图 17-27 响应刹车的输出 (OISx=1, OISxN=0)	257
图 17-28 响应刹车的输出 (OISx=1, OISxN=1)	257
图 17-29 外部事件清除 OCxREF.....	258
图 17-30 单脉冲模式.....	258
图 17-31 编码器模式下的计数器时序图.....	260
图 17-32 IC1FP1 反相编码器接口模式时序图.....	261
图 17-33 复位模式的控制时序图	261
图 17-34 门控模式下的控制时序图.....	262
图 17-35 触发器模式下的控制时序图	262
图 17-36 外部时钟模式 2+从模式 (触发模式) 控制时序图.....	263
图 17-37 定时器间互联	263
图 17-38 使用 TIM3 作为 TIM1 的预分频器.....	264
图 17-39 使用 TIM1 使能 TIM2.....	264
图 17-40 使用 TIM1 的更新事件启动 TIM2.....	265
图 17-41 TIM1 的 TI1 同步启动 TIM1 和 TIM2	265
图 17-42 (TI1 异或输入) 输入捕获波形图	266
图 17-43 霍尔传感器接口实例	267
图 18-1 TIM2 结构图.....	301

图 18-2 时钟选择	303
图 18-3 外部时钟模式 1 下的控制电路	304
图 18-4 外部时钟模式 2 下的控制电路	304
图 18-5 自动预装载.....	305
图 18-6 递增计数模式 (UDIS=0)	305
图 18-7 递增计数模式 (UDIS=1 禁止产生更新事件)	306
图 18-8 递减计数模式 (UDIS=0)	306
图 18-9 递减计数模式 (UDIS=1 禁止产生更新事件)	306
图 18-10 中央计数模式 (UDIS=0)	307
图 18-11 中央计数模式 (UDIS=1 禁止产生更新事件)	307
图 18-12 TIM2 输入捕获结构图	307
图 18-13 PWM 输入模式时序	309
图 18-14 比较输出部分结构图	310
图 18-15 比较输出模式, OC1 信号在匹配时翻转	311
图 18-16 边沿对齐递增计数时 PWM 模式 1 的波形	312
图 18-17 边沿对齐递减计数时 PWM 模式 1 的波形	313
图 18-18 中央对齐 PWM 模式 1 的波形.....	314
图 18-19 外部事件清除 OCxREF.....	315
图 18-20 单脉冲模式.....	315
图 18-21 编码器模式下的计数器时序图.....	317
图 18-22 IC1FP1 反相编码器接口模式时序图.....	317
图 18-23 复位模式的控制时序图	318
图 18-24 门控模式下的控制时序图.....	319
图 18-25 触发器模式下的控制时序图.....	319
图 18-26 外部时钟模式 2+从模式 (触发模式) 控制时序图.....	320
图 18-27 (TI1 异或输入) 输入捕获波形图	320
图 19-1 TIM3 结构图.....	347
图 19-2 时钟选择	349
图 19-3 外部时钟模式 1 下的控制电路	350
图 19-4 外部时钟模式 2 下的控制电路	350
图 19-5 自动预装载.....	351
图 19-6 递增计数模式 (UDIS=0)	351
图 19-7 递增计数模式 (UDIS=1 禁止产生更新事件)	352
图 19-8 递减计数模式 (UDIS=0)	352
图 19-9 递减计数模式 (UDIS=1 禁止产生更新事件)	352
图 19-10 中央计数模式 (UDIS=0)	353
图 19-11 中央计数模式 (UDIS=1 禁止产生更新事件)	353
图 19-12 TIM3 输入捕获结构图	353
图 19-13 PWM 输入模式时序	355
图 19-14 比较输出部分结构图	356
图 19-15 比较输出模式, OC1 信号在匹配时翻转	357
图 19-16 边沿对齐递增计数时 PWM 模式 1 的波形	358
图 19-17 边沿对齐递减计数时 PWM 模式 1 的波形	359
图 19-18 中央对齐 PWM 模式 1 的波形.....	360

图 19-19 外部事件清除 OCxREF.....	361
图 19-20 单脉冲模式.....	361
图 19-21 编码器模式下的计数器时序图.....	363
图 19-22 IC1FP1 反相编码器接口模式时序图.....	363
图 19-23 复位模式的控制时序图.....	364
图 19-24 门控模式下的控制时序图.....	365
图 19-25 触发器模式下的控制时序图.....	365
图 19-26 外部时钟模式 2+从模式（触发模式）控制时序图.....	366
图 19-27（TI1 异或输入）输入捕获波形图.....	366
图 20-1 TIM14 结构图.....	392
图 20-2 自动预装载.....	393
图 20-3 递增计数模式（UDIS=0）.....	393
图 20-4 递增计数模式（UDIS=1 禁止产生更新事件）.....	394
图 20-5 TIM14 输入捕获结构图.....	394
图 20-6 比较输出部分结构图.....	395
图 20-7 比较输出模式，OC1 信号在匹配时翻转.....	397
图 20-8 边沿对齐递增计数时 PWM 模式 1 的波形.....	398
图 20-9 单脉冲模式.....	398
图 21-1 TIMx 结构图.....	409
图 21-2 自动预装载.....	410
图 21-3 递增计数模式（UDIS=0）.....	411
图 21-4 递增计数模式（UDIS=1 禁止产生更新事件）.....	411
图 21-5 TIMx 输入捕获结构图.....	411
图 21-6 比较输出部分结构图.....	413
图 21-7 比较输出模式，OC1 信号在匹配时翻转.....	414
图 21-8 边沿对齐递增计数时 PWM 模式 1 的波形.....	415
图 21-9 死区插入.....	416
图 21-10 响应刹车的输出（OISx=0, OISxN=0）.....	419
图 21-11 响应刹车的输出（OISx=0, OISxN=1）.....	419
图 21-12 响应刹车的输出（OISx=1, OISxN=0）.....	419
图 21-13 响应刹车的输出（OISx=1, OISxN=1）.....	419
图 21-14 单脉冲模式.....	419
图 22-1 功能框图.....	436
图 23-1 WWDG 功能框图.....	443
图 23-2 WWDG 超时时间坐标图.....	445
图 24-1 功能框图.....	449
图 24-2 流程框图.....	451
图 25-1 UART 功能框图.....	457
图 25-2 UART 时序.....	459
图 25-3 RX 引脚采样方法.....	460
图 25-4 发送时状态位变化.....	462
图 25-5 UART 奇偶校验方框图.....	467
图 25-6 UART 采样 NACK 信号方框图.....	468
图 25-7 普通模式下 IrDA 发送和接收图.....	469

图 26-1 LPUART 功能框图.....	484
图 26-2 帧时序.....	486
图 26-3 接收时序.....	487
图 26-4 发送时序.....	487
图 27-1 SPI_I2S 功能框图.....	500
图 27-2 单主和单从应用.....	501
图 27-3 数据时钟时序图.....	502
图 27-4 飞利浦标准示意图.....	509
图 27-5 MSB 对齐标准示意图.....	509
图 27-6 LSB 对齐标准示意图.....	510
图 27-7 PCM 标准示意图.....	510
图 27-8 I2S 从模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器).....	511
图 27-9 I2S 主模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器).....	511
图 27-10 I2S 时钟预分频器示意图.....	512
图 28-1 I2C 功能框图.....	529
图 28-2 起始和停止条件.....	530
图 28-3 7 位地址格式.....	531
图 28-4 10 位地址格式.....	531
图 28-5 主发送协议.....	532
图 28-6 主接收协议.....	532
图 28-7 带 RESTART (SR) 信号的主发送和接收协议.....	533
图 28-8 起始字节传输.....	533
图 28-9 I2C_DR 寄存器.....	534
图 28-10 主发送, TX FIFO 为空.....	534
图 28-11 主接收, TX FIFO 为空.....	534
图 28-12 主发送, 发生 RESTART.....	535
图 28-13 主接收, 发生 RESTART.....	535
图 28-14 两个主器件仲裁.....	535
图 28-15 时钟同步 (示意图).....	536
图 28-16 时钟同步 (时序图).....	536
图 28-17 SCL 生成时序.....	537
图 28-18 流程图 (I2C 接口作为从器件).....	539
图 28-19 流程图 (I2C 接口作为主器件).....	541
图 28-20 I2C 中断机制.....	543
图 29-1 USB_FS 功能框图.....	564
图 29-2 仅从设备硬件连接示意图.....	565
图 29-3 缓冲区描述表.....	566
图 29-4 USB 令牌传输示意图.....	569
图 30-1 FlexCAN 框图.....	580
图 30-2 CAN 帧格式.....	601
图 30-3 CAN FD 报文比特率切换机制.....	602
图 30-4 收发回路延迟测量.....	604
图 30-5 具有最优值的 SSP 位置.....	605
图 30-6 PE 时钟结构.....	607

图 30-7 位时间内的段（使用 CAN_CTRL1 位时序变量的经典 CAN 格式）	608
图 30-8 位时间内的段（使用 CAN FD 格式的 CAN_CBT 和 CAN_FDCBT 位时序变量）	608
图 30-9 匹配和移入时间段	610
图 30-10 仲裁和移出时间段	610
图 30-11 总线关闭结束时的仲裁和移出时间段	610
图 30-12 优化的 Tx 仲裁起点	611
图 30-13 信息寄存器结构示例	648
图 30-14 传统型 Rx FIFO 结构	653
图 30-15 ID 过滤表结构	653
图 30-16 增强型 Rx FIFO 结构	655
图 32-1 调试功能框图	661

Preliminary

表格目录

表 2-1 存储器映像.....	31
表 3-1 异常向量表.....	34
表 3-2 中断向量表.....	34
表 3-3 EXTI 触发源.....	35
表 3-4 DMA 通道分配.....	36
表 3-5 ADC 通道分配.....	37
表 3-6 IP 信号行为描述.....	38
表 3-7 TIM 间的互联.....	38
表 3-8 TIM 和 EXTI 到 ADC 触发的互联.....	38
表 3-9 COMP 输出到 TIM 的互联.....	39
表 3-10 TIM1、TIM16、TIM17 刹车事件的触发.....	39
表 3-11 从时钟源到 TIM2、TIM3 的 CC4 输入的 Remap.....	40
表 3-12 从时钟源到 TIM2、TIM3 的外部触发信号.....	40
表 4-1 Flash 存储空间.....	42
表 4-2 信息块.....	43
表 4-3 选项字节组织结构.....	43
表 4-4 USER 的位含义.....	43
表 4-5 Latency 设置关系.....	45
表 4-6 编程方式.....	46
表 4-7 保护设置的状态变化.....	49
表 4-8 Flash 读保护状态.....	57
表 4-9 Flash 解除读保护状态.....	57
表 4-10 写保护区域.....	58
表 4-11 FLASH 寄存器概览.....	58
表 5-1 低功耗模式列表.....	68
表 5-2 SLEEPNOW 模式.....	69
表 5-3 SLEEPONEXIT 模式.....	69
表 5-4 停机模式.....	70
表 5-5 深度停机模式.....	72
表 5-6 待机模式.....	73
表 5-7 电源控制寄存器概览.....	73
表 6-1 SYSCFG 寄存器概览.....	80
表 7-1 RCC 全局中断表.....	94
表 7-2 MCO 与时钟源对应关系.....	97
表 7-3 RCC 寄存器概览.....	98
表 8-1 RTC 寄存器概览.....	128
表 9-1 CRC 寄存器概览.....	137
表 10-1 CRS 中断请求.....	147
表 10-2 CRS 寄存器概览.....	148

表 11-1 HWDIV 寄存器描述概览	153
表 12-1 端口位配置表 (port0 为例)	158
表 12-2 SWD 复用功能重映射	163
表 12-3 GPIO 寄存器概览.....	163
表 13-1 EXTI 寄存器总览.....	172
表 14-1 DMA 中断请求	177
表 14-2 DMA 各个通道的 DMA 请求一览.....	177
表 14-3 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.....	181
表 14-4 DMA 寄存器概览.....	182
表 15-1 COMP 寄存器概览.....	192
表 16-1 中断事件一览表	199
表 16-2 数据分辨率与数据补偿左对齐关系	200
表 16-3 寄存器概览.....	213
表 17-1 数字滤波器宽度与 ICxF 的对应关系表	246
表 17-2 死区时间计算.....	253
表 17-3 当 MOE=1, OSSR=0/1, OSSI=0 时:	255
表 17-4 当 MOE=1, OSSR=0/1, OSSI=1 时:	255
表 17-5 当 MOE=0, OSSR=0, OSSI=0/1 时:	255
表 17-6 当 MOE=0, OSSR=1, OSSI=0/1 时:	256
表 17-7 计数方向与编码器信号的关系	260
表 17-8 中断事件一览表	268
表 17-9 TIM1 寄存器概览	269
表 17-10 TIM1 内部触发连接.....	275
表 17-11 ICx 极性/电平选择表.....	290
表 18-1 数字滤波器宽度与 ICxF 的对应关系表	308
表 18-2 计数方向与编码器信号的关系	316
表 18-3 中断事件一览表	321
表 18-4 TIM2 寄存器概览	322
表 18-5 TIM2 内部触发连接.....	328
表 18-6 ICx 极性/电平选择表.....	341
表 19-1 数字滤波器宽度与 ICxF 的对应关系表	354
表 19-2 计数方向与编码器信号的关系	362
表 19-3 中断事件一览表	367
表 19-4 TIM3 寄存器概览	368
表 19-5 TIM3 内部触发连接.....	374
表 19-6 ICx 极性/电平选择表.....	387
表 20-1 数字滤波器宽度与 IC1F 的对应关系表	394
表 20-2 中断事件一览表	398
表 20-3 TIM14 寄存器概览	399
表 20-4 IC1 极性/电平选择表.....	406
表 21-1 数字滤波器宽度与 IC1F 的对应关系表	412
表 21-2 死区时间计算.....	415
表 21-3 当 MOE=1, OSSR=0/1, OSSI=0 时:	416
表 21-4 当 MOE=1, OSSR=0/1, OSSI=1 时:	417

表 21-5 当 MOE=0, OSSI=0, OSSR=0/1 时:	417
表 21-6 当 MOE=0, OSSI=1, OSSR=0/1 时:	417
表 21-7 中断事件一览表	420
表 21-8 TIMx 寄存器概览.....	420
表 21-9 IC1 极性/电平选择表.....	430
表 22-1 LPTIM 寄存器概览.....	438
表 23-1 超时时间	445
表 23-2 WWDG 寄存器概览	445
表 24-1 IWDG 超时时间 (40kHz 的输入时钟 (LSI))	451
表 24-2 IWDG 寄存器概览.....	452
表 25-1 UART 中断请求	469
表 25-2 UART 寄存器概览.....	470
表 26-1 LPUART 中断请求.....	485
表 26-2 调制控制寄存器 MCTL 建议配置参数表	487
表 26-3 调制控制寄存器 MCTL 计算方式.....	488
表 26-4 LPUART 寄存器概览	491
表 27-1 波特率公式.....	506
表 27-2 SPI 状态.....	506
表 27-3 I2S 比特率计算公式.....	513
表 27-4 音频采样率与 F _{I2SCLK} (APB 时钟频率) 计算关系.....	513
表 27-5 I2S 音频采样率误差表	513
表 27-6 I2S 中断状态.....	514
表 27-7 SPI_I2S 寄存器概览	515
表 28-1 引脚定义	530
表 28-2 I2C 首字节	531
表 28-3 中断位的置位和清除	542
表 28-4 I2C 寄存器概览.....	543
表 28-5 DISSLAVE 和 MASTER 配置.....	546
表 29-1 术语表.....	563
表 29-2 USB 信号描述表.....	564
表 29-3 BDT 地址计算	567
表 29-4 BDT 地址计算域说明	567
表 29-5 BD 格式.....	567
表 29-6 USB 寄存器概览.....	570
表 30-1 经典 CAN 和 CAN FD 模式功能对比.....	582
表 30-2 本地优先级无效时仲裁值的构成.....	584
表 30-3 本地优先级使能时仲裁值的构成.....	585
表 30-4 匹配架构	588
表 30-5 匹配可能性及由此产生的接收结构	589
表 30-6 标准 ID 过滤元素 (FSCH = 00)	595
表 30-7 标准 ID 过滤元素 (FSCH = 01)	596
表 30-8 标准 ID 过滤元素 (FSCH = 10)	596
表 30-9 扩展 ID 过滤元素 (FSCH = 00)	596
表 30-10 扩展 ID 过滤元素 (FSCH = 01)	596

表 30-11 扩展 ID 过滤元素 (FSCH = 10)	597
表 30-12 ESI 位的写入与传输	602
表 30-13 时间段语法	608
表 30-14 符合 Bosch CAN 2.0B 标准的位时间段设置	609
表 30-15 经典 CAN 格式 CAN 位的最少外设时钟数	612
表 30-16 CAN FD 帧中快速 CAN 位个数	613
表 30-17 CAN FD 帧数据段最大比特率 (增强型 Rx FIFO 禁止)	614
表 30-18 寄存器访问和复位信息	616
表 30-19 FlexCAN 寄存器/内存映射概览	617
表 30-20 MG 位掩码邮箱过滤字段	625
表 30-21 FLEXCAN 状态相关位编码	631
表 30-22 传统型 Rx FIFO 过滤器数量	636
表 30-23 传统型 Rx FIFO 全局掩码位和 IDF 字段的对应关系	639
表 30-24 DMALW 和传输字数	645
表 30-25 NEXIF 和扩展/标准 ID 过滤元素数	645
表 30-26 接收缓冲区代码	648
表 30-27 发送缓冲区代码	649
表 30-28 有效数据字节	651
表 30-29 MB 内存映射	651
表 30-30 ID HIT 偏移地址	655
表 31-1BKP 寄存器概览	658
表 32-1 SWD 调试端口管脚	662
表 32-2 ID 编码	662
表 32-3 8bit 请求包	663
表 32-4 3bit 应答包	663
表 32-5 33bit 数据包	664
表 32-6 SW-DP 寄存器	665
表 32-7 DBG 寄存器概览	666
表 33-1 存储器容量寄存器概览	669
表 34-1 修订记录表	671

1 系统和存储器架构

1.1 系统架构简介

MM32G0160 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。

MM32G0160 采用矩阵总线结构,该矩阵包括三个 AHB 主机: CPU 、 DMA 和 USB, 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。

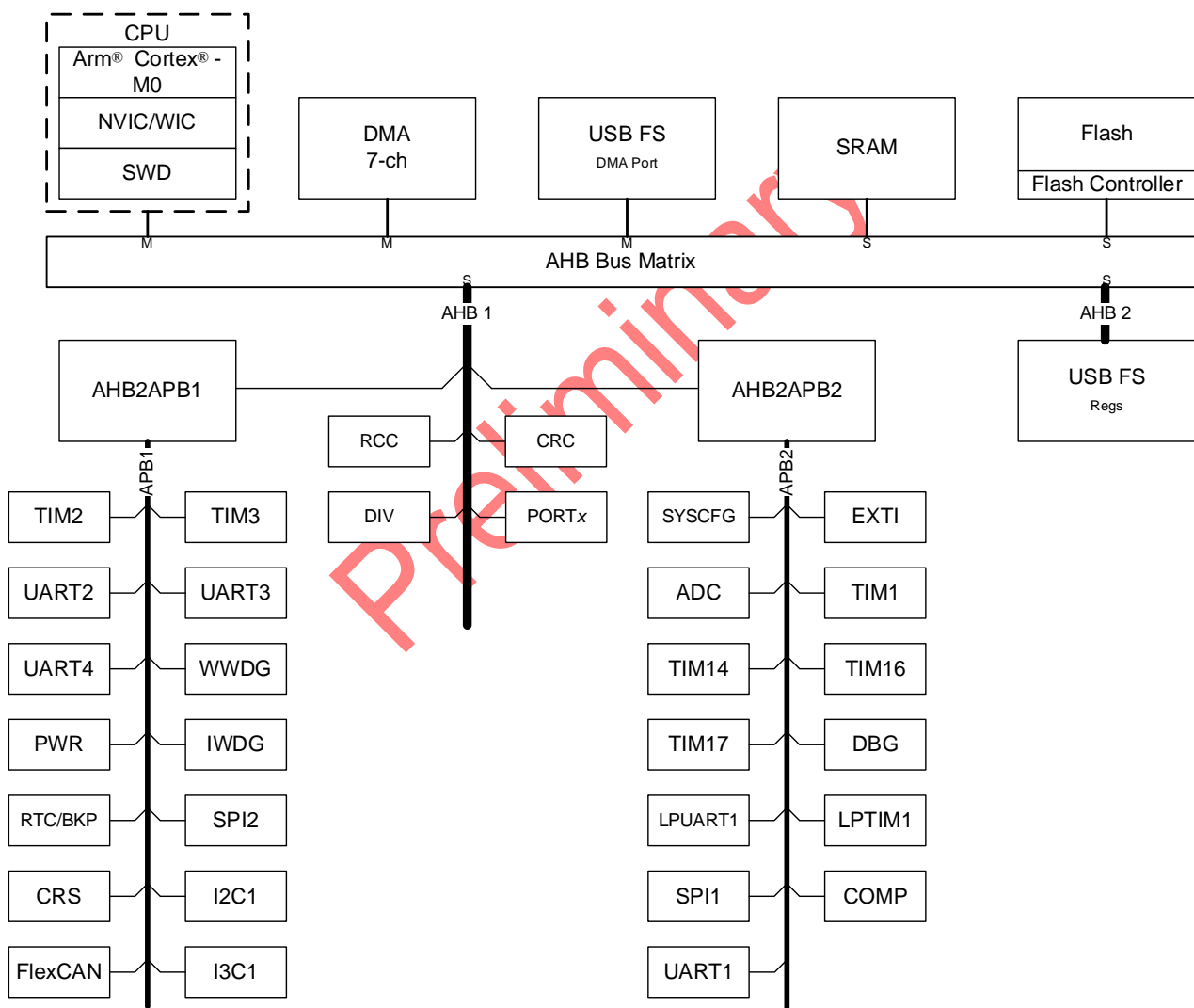


图 1-1 系统架构框图

1.2 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, DMA 和 USB 作为主机驱

动总线，总线矩阵会协调 CPU 内核，DMA 和 USB 之间的访问。

1.3 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。

1.4 USB FS 总线

USB FS 内置本地 DMA，用于同 SRAM 之间数据的传输，占用总线的的一个 Master 端口；USB FS 的寄存器则挂载在 AHB2 上，用于 CPU 配置该外设。

1.5 总线矩阵

总线矩阵包括 1 个 AHB 互联矩阵，AHB1 总线,AHB2 总线和两个桥接的 APB 总线。当 CPU 总线，DMA 总线和 USB FS 总线同时请求时，具备仲裁的功能。AHB1 总线的外设（RCC, HWDIV, GPIO 和 CRC）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 AHB2APB 桥也具备自动拓宽功能。AHB2 总线的外设（USB FS 寄存器配置）通过 AHB 互联矩阵与系统总线连接，仅支持 CPU 访问。

2 地址映射

程序存储器、数据存储器、寄存器和 I/O 接口都位于相同的存储器地址空间（线性 4GB 的地址空间），只是在不同的地址范围。4GB 的地址空间被分为 8 块，每块为 512MB，分配给片上存储器和外设的存储器空间为固定的，不可更改，其余的地址空间为保留的地址空间。

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图。

表 2-1 存储器映像

总线	编址范围	大小	外设
FLASH	0x0000 0000 - 0x0001 FFFF	128 KB	映射为主闪存存储器
	0x0002 0000 - 0x000F FFFF	896 KB	Reserved
	0x0010 0000 - 0x07FF FFFF	127 MB	Reserved
	0x0800 0000 - 0x0801 FFFF	128 KB	主 FLASH 存储区
	0x0802 0000 - 0x080F FFFF	896 KB	Reserved
	0x0810 0000 - 0x1FFD FFFF	382 MB	Reserved
	0x1FFE 0000 - 0x1FFE 01FF	0.5 KB	Reserved
	0x1FFE 0200 - 0x1FFE 0FFF	3.5 KB	Reserved
	0x1FFE 1000 - 0x1FFE 25FF	5.5 KB	保密空间
	0x1FFE 2600 - 0x1FFF F3FF	115.5 KB	Reserved
	0x1FFF F400 - 0x1FFF F7FF	1 KB	系统存储区
	0x1FFF F800 - 0x1FFF F9FF	0.5 KB	选项字节区
	0x1FFF FA00 - 0x1FFF FFFF	1.5 KB	Reserved
SRAM	0x2000 0000 - 0x2000 3FFF	16 KB	SRAM
	0x2000 8000 - 0x200F FFFF	1008 KB	Reserved
	0x2010 8000 - 0x2FFF FFFF	255 MB	Reserved
	0x3000 0000 - 0x3FFF FFFF	256 MB	Reserved
APB1	0x4000 0000 - 0x4000 03FF	1 KB	TIM2
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3
	0x4000 0800 - 0x4000 27FF	8 KB	Reserved
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC_BKP
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG
	0x4000 3400 - 0x4000 37FF	1 KB	Reserved
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2
	0x4000 3C00 - 0x4000 43FF	2 KB	Reserved
	0x4000 4400 - 0x4000 47FF	1 KB	UART2
	0x4000 4800 - 0x4000 4BFF	1 KB	UART3
0x4000 4C00 - 0x4000 4FFF	1 KB	UART4	

总线	编址范围	大小	外设
	0x4000 5000 - 0x4000 53FF	1 KB	Reserved
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1
	0x4000 5800 - 0x4000 6BFF	5 KB	Reserved
	0x4000 6C00 - 0x4000 6FFF	1 KB	CRS
	0x4000 7000 - 0x4000 73FF	1 KB	PWR
	0x4000 7400 - 0x4000 9FFF	29 KB	Reserved
	0x4000 A000 - 0x4000 AFFF	1 KB	I3C1
	0x4000 B000 - 0x4000 BFFF	4 KB	Reserved
	0x4000 C000 - 0x4000 FFFF	16 KB	FLEXCAN
APB2	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI
	0x4001 0800 - 0x4001 0BFF	1 KB	LPUART1
	0x4001 0C00 - 0x4001 23FF	6 KB	Reserved
	0x4001 2400 - 0x4001 27FF	1 KB	ADC
	0x4001 2800 - 0x4001 2BFF	1 KB	LPTIM1
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1
	0x4001 3400 - 0x4001 37FF	1 KB	DBGMCU
	0x4001 3800 - 0x4001 3BFF	1 KB	UART1
	0x4001 3C00 - 0x4001 3FFF	1 KB	COMP
	0x4001 4000 - 0x4001 43FF	1 KB	TIM14
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17
	0x4001 4C00 - 0x4001 FFFF	45 KB	Reserved
AHB1	0x4002 0000 - 0x4002 03FF	1 KB	DMA
	0x4002 0400 - 0x4002 0FFF	3 KB	Reserved
	0x4002 1000 - 0x4002 13FF	1 KB	RCC
	0x4002 1400 - 0x4002 1FFF	3 KB	Reserved
	0x4002 2000 - 0x4002 23FF	1 KB	Flash interface
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved
	0x4002 3000 - 0x4002 33FF	1 KB	CRC
	0x4002 3400 - 0x4002 FFFF	51 KB	Reserved
	0x4003 0000 - 0x4003 03FF	1 KB	HWDIV
	0x4003 0400 - 0x400F FFFF	831 KB	Reserved
	0x4010 0000 - 0x47FF FFFF	127 MB	Reserved
	0x4800 0000 - 0x4800 03FF	1 KB	GPIOA
	0x4800 0400 - 0x4800 07FF	1 KB	GPIOB
	0x4800 0800 - 0x4800 0BFF	1 KB	GPIOC
	0x4800 0C00 - 0x4800 0FFF	1 KB	GPIOD
	0x4800 1000 - 0x4800 FFFF	60 KB	Reserved
	0x4801 0000 - 0x480F FFFF	960 KB	Reserved
	0x4810 0000 - 0x4FFF FFFF	127 MB	Reserved

总线	编址范围	大小	外设
AHB2	0x5000 0000 - 0x5003 FFFF	256 KB	USB_FS
	0x5004 0000 - 0x500F FFFF	768 KB	Reserved
	0x5010 0000 - 0x5FFF FFFF	255 MB	Reserved

2.2 内置的 SRAM

内置最大可到 16K 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 或者 DMA 用最快的系统时钟且不插入任何等待进行访问。

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。
- 由选项字节和系统存储器组成的信息块：
 - ◆ 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）：boot loader 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。

3 芯片特定配置

3.1 中断向量表

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向量与中断向量。

表 3-1 异常向量表

位置	优先级	优先级类别	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 向量	0x0000_0008
	-1	固定	硬件失效（HardFault）	所有类型的失效	0x0000_000C
	-	-	-	保留	0x0000_0010
	-	-	-	保留	0x0000_0014
	-	-	-	保留	0x0000_0018
	-	-	-	保留	0x0000_001C
	-	-	-	保留	0x0000_0020
	-	-	-	保留	0x0000_0024
	-	-	-	保留	0x0000_0028

表 3-2 中断向量表

位置	优先级	优先级类别	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
	-	可设置	-	保留	0x0000_0030
	-	可设置	-	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	WWDG_IWDG	看门狗中断（IWDG 为 EXTI24）	0x0000_0040
1	8	可设置	PVD	电源电压检测（PVD）中断（EXTI16）	0x0000_0044
2	9	可设置	RTC_BKP	RTC 全局中断	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	EXTI[1:0]	EXTI 线[1:0]中断	0x0000_0054
6	13	可设置	EXTI[3:2]	EXTI 线[3:2]中断	0x0000_0058
7	14	可设置	EXTI[15:4]	EXTI 线[15:4]中断	0x0000_005C

位置	优先级	优先级类别	名称	说明	地址
8	15	可设置	HWDIV	HWDIV 中断	0x0000_0060
9	16	可设置	DMA1_CH1	DMA1 通道 1 全局中断	0x0000_0064
10	17	可设置	DMA1_CH[3:2]	DMA1 通道[3:2]全局中断	0x0000_0068
11	18	可设置	DMA1_CH[7:4]	DMA1 通道[7:4]全局中断	0x0000_006C
12	19	可设置	ADC_COMP	ADC 全局中断、连接到 EXTI19 的比较器 1 中断、连接到 EXTI20 的比较器 2 的中断	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	TIM2	TIM2 全局中断	0x0000_007C
16	23	可设置	TIM3	TIM3 全局中断	0x0000_0080
17	24	可设置	LPUART1	LPUART1 全局中断(EXTI22)	0x0000_0084
18	25	可设置	LPTIM1	LPTIM1 全局中断(EXTI23)	0x0000_0088
19	26	可设置	TIM14	TIM14 全局中断	0x0000_008C
20	27	-	-	保留	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	I3C1	I3C1 全局中断	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000_00A4
26	33	可设置	SPI2	SPI2 全局中断	0x0000_00A8
27	34	可设置	UART1	UART1 全局中断	0x0000_00AC
28	35	可设置	UART2	UART2 全局中断	0x0000_00B0
29	36	可设置	UART3_4	UART3 与 UART4 全局中断	0x0000_00B4
30	37	可设置	FlexCAN	FLexCAN 全局中断	0x0000_00B8
31	38	可设置	USB	USB 全局中断与 USB 唤醒中断 (EXTI18)	0x0000_00BC

3.2 外部中断映射表

GPIO 对应的 16 个外部中断/事件映射关系如下表所示：

表 3-3 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PX0(X=A,B,C,D)	SYSCFG_EXTICR1 寄存器中的 EXTI0
EXTI1	PX1(X=A,B,C,D)	SYSCFG_EXTICR1 寄存器中的 EXTI1
EXTI2	PX2(X=A,B,C,D)	SYSCFG_EXTICR1 寄存器中的 EXTI2
EXTI3	PX3(X=A,B,C,D)	SYSCFG_EXTICR1 寄存器中的 EXTI3
EXTI4	PX4(X=A,B,C,D,)	SYSCFG_EXTICR2 寄存器中的 EXTI4
EXTI5	PX5(X=A,B,C,D)	SYSCFG_EXTICR2 寄存器中的 EXTI5

外部中断线	IO 映射	控制位
EXTI6	PX6(X=A,B,C,D)	SYSCFG_EXTICR2 寄存器中的 EXTI6
EXTI7	PX7(X=A,B,C,D)	SYSCFG_EXTICR2 寄存器中的 EXTI7
EXTI8	PX8(X=A,B,C,D)	SYSCFG_EXTICR3 寄存器中的 EXTI8
EXTI9	PX9(X=A,B,C)	SYSCFG_EXTICR3 寄存器中的 EXTI9
EXTI10	PX10(X=A,B,C)	SYSCFG_EXTICR3 寄存器中的 EXTI10
EXTI11	PX11(X=A,B,C)	SYSCFG_EXTICR3 寄存器中的 EXTI11
EXTI12	PX12(X=A,B,C)	SYSCFG_EXTICR4 寄存器中的 EXTI12
EXTI13	PX13(X=A,B,C)	SYSCFG_EXTICR4 寄存器中的 EXTI13
EXTI14	PX14(X=A,B,C)	SYSCFG_EXTICR4 寄存器中的 EXTI14
EXTI15	PX15(X=A,B,C)	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/VDT 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 18 连接到 USB_SUSPEND
- EXTI 线 19 连接到 COMP1 输出
- EXTI 线 20 连接到 COMP2 输出
- EXTI 线 21 保留
- EXTI 线 22 连接到 LPUART 输出
- EXTI 线 23 连接到 LPTIM
- EXTI 线 24 连接到 IWDG

3.3 DMA 通道分配

MM32G0160 使用 7 通道 DMA。

DMA 控制器从外设 ADC、SPIx、UARTx、LPUART、I2C、I3C、FlexCAN 和 TIMx 产生的请求，通过逻辑或输入到 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应有对应的 DMA 使能位，来独立控制外设是否发送 DMA 请求。

表 3-4 DMA 通道分配

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
ADC	ADC ⁽¹⁾	ADC ⁽²⁾					
SPI		SPI1_RX	SPI1_TX	SPI2_RX ⁽¹⁾	SPI2_TX ⁽¹⁾	SPI2_RX ⁽²⁾	SPI2_TX ⁽²⁾
UART		UART1_TX ⁽¹⁾ UART3_TX ⁽¹⁾	UART1_RX ⁽¹⁾ UART3_RX ⁽¹⁾	UART1_TX ⁽²⁾ UART2_TX ⁽¹⁾	UART1_RX ⁽²⁾ UART2_RX ⁽¹⁾	UART2_RX ⁽²⁾ UART3_RX ⁽²⁾ UART4_RX	UART2_TX ⁽²⁾ UART3_TX ⁽²⁾ UART4_TX
LPUART	LPUART_TX	LPUART_RX					
I2C		I2C1_TX ⁽¹⁾	I2C1_RX ⁽¹⁾			I2C1_TX ⁽²⁾	I2C1_RX ⁽²⁾

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
I3C				I3C1_TX	I3C1_RX		
FlexCAN	FlexCAN						
TIM1		TIM1_CH1 ⁽¹⁾	TIM1_CH2 ⁽¹⁾	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 ⁽¹⁾ TIM1_UP TIM1_CH5	TIM1_CH1 ⁽²⁾ TIM1_CH2 ⁽²⁾ TIM1_CH3 ⁽²⁾	
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2 ⁽¹⁾	TIM2_CH4 ⁽¹⁾	TIM2_CH1		TIM2_CH2 ⁽²⁾ TIM2_CH4 ⁽²⁾
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 ⁽¹⁾ TIM3_TRIG ⁽¹⁾		TIM3_CH1 ⁽²⁾ TIM3_TRIG ⁽²⁾	
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⁾					

注:

- SYSCFG_CFGR 寄存器相关映射位被清除, DMA 请求映射在该通道;
- SYSCFG_CFGR 寄存器相关映射位被置位, DMA 请求映射在该通道;

3.4 ADC 通道分配

表 3-5 ADC 通道分配

转换输入通道	ADC1 通道分配
通道 0	ADC_IN0(PA0)
通道 1	ADC_IN1(PA1)
通道 2	ADC_IN2(PA2)
通道 3	ADC_IN3(PA3)
通道 4	ADC_IN4(PA4)
通道 5	ADC_IN5(PA5)
通道 6	ADC_IN6(PA6)
通道 7	ADC_IN7(PA7)
通道 8	ADC_IN8(PB0)
通道 9	ADC_IN9(PB1)
通道 10	ADC_IN10(PB3)
通道 11	ADC_IN11(PB4)
通道 12	ADC_IN12(PB7)
通道 13	Reserved
通道 14	内置 T-Sensor
通道 15	内置 V-Sensor

3.5 系统模块硬件互联

本章介绍 IP 间的信号互联，包括 Timer 间的相互触发、Timer 到 ADC 的触发、比较器到 Timer 的触发等。

为明确 IP 内部的具体信号，参考表 3-6 用于描述各信号的具体行为。

表 3-6 IP 信号行为描述

名称	行为描述
ITRn	来自寄存器比特 TIMx_SMCR.TS
TIMx_TRGO	TIMx IP 的 TRGO 信号，依寄存器比特 TIMx_CR2.MMS 控制
TIMx_ICn	TIMx IP 的第 n 个输入
TIMx_OCnREF	TIMx IP 的第 n 个 compare match 输出（内部信号）
TIMx_BRKIN	TIMx IP 的刹车输入
TRGSEL	来自寄存器比特 ADC_ADCR.TRGSEL
LPTIMx_OUT	LPTIM 的输出信号
LPTIMx_TRIGGER	LPTIM 的触发信号（来自引脚）
OUTSEL	来自寄存器比特 COMPx_CSR.OUTSEL
ETRRMP	来自寄存器比特 TIMx_OR.ETRRMP
TI4RMP	来自寄存器比特 TIMx_OR.TI4RMP

3.5.1 TIM 间的互联

表 3-7 TIM 间的互联

从设备	ITR0	ITR1	ITR2	ITR3
TIM1	N/A	TIM2_TRGO	TIM3_TRGO	TIM17_OC1REF
TIM2	TIM1_TRGO	N/A	TIM3_TRGO	TIM14_OC1REF
TIM3	TIM1_TRGO	TIM2_TRGO	TIM16_OC1REF	TIM14_OC1REF

3.5.2 TIM 和 EXTI 触发 ADC

表 3-8 TIM 和 EXTI 到 ADC 触发的互联

Name	Source	TRGSEL[4:0]
TRG0	TIM1_OC1REF	00000
TRG1	TIM1_OC2REF	00001
TRG2	TIM1_OC3REF	00010
TRG3	TIM2_OC2REF	00011
TRG4	TIM3_TRGO	00100
TRG5	TIM1_OC4REF^TIM1_OC5REF	00101
TRG6	TIM3_OC1REF	00110
TRG7	EXTI[11]	00111
TRG8	TIM1_TRGO	01000

Name	Source	TRGSEL[4:0]
TRG9	EXTI[4]	01001
TRG10	EXTI[5]	01010
TRG11	TIM2_OC1REF	01011
TRG12	TIM3_OC4REF	01100
TRG13	TIM2_TRGO	01101
TRG14	N/A	01110
TRG15	EXTI[15]	01111
TRG16	TIM1_OC4REF	10000
TRG17	TIM1_OC5REF	10001

3.5.3 COMP 触发 TIM

表 3-9 COMP 输出到 TIM 的互联

OUTSEL	Source	Trigger
0000	N/A	N/A
0001	N/A	N/A
0010	COMP1_OUT COMP2_OUT	TIM1_BRKIN
0011	N/A	N/A
0100	N/A	N/A
0101	N/A	N/A
0110	COMP1_OUT COMP2_OUT	TIM1_OCREF_CLR
0111	COMP1_OUT COMP2_OUT	TIM1_IC1
1000	COMP1_OUT COMP2_OUT	TIM2_IC4
1001	COMP1_OUT COMP2_OUT	TIM2_OCREF_CLR
1010	COMP1_OUT COMP2_OUT	TIM3_IC1
1011	COMP1_OUT COMP2_OUT	TIM3_OCREF_CLR
1100	N/A	N/A
1101	N/A	N/A
1110	N/A	N/A
1111	COMP1_OUT COMP2_OUT	LPTIM1_TRIGGER

3.5.4 TIM 刹车事件的触发

TIM1、TIM16 和 TIM17 的刹车输入从以下部分来，详见表 3-10。

- CSS 事件；
- 相应的 BKIN IO；
- 比较器输出（参考表 3-9）。

表 3-10 TIM1、TIM16、TIM17 刹车事件的触发

TIMx	BKINSEL	Source
TIM1	Bit-0	CSS

TIMx	BKINSEL	Source
	Bit-1	TIM1_BKIN1
	Bit-2	TIM1_BKIN2
	Bit-3	TIM1_BKIN3
	Bit-4	TIM1_BKIN4
	Bit-5	Reserved
	Bit-6	COMP1_OUT COMP2_OUT
TIM16	Bit-0	CSS
	Bit-1	TIM16_BKIN1
TIM17	Bit-0	CSS
	Bit-1	TIM17_BKIN1
	Bit-2	TIM17_BKIN2

3.5.5 从时钟到 TIM 的互联

表 3-11 从时钟源到 TIM2、TIM3 的 CC4 输入的 Remap

TIMx	TI4RMP	Source
TIM2	00	TIM2_IC4 COMP_TIM2_IC4
	01	LSI_CLK
	10	Reserved
	11	HSE_CLK_DIV_128
TIM3	00	TIM3_IC4
	01	N/A
	10	N/A
	11	N/A

表 3-12 从时钟源到 TIM2、TIM3 的外部触发信号

TIMx	ETRRMP	Source
TIM2	00	ETR
	01	N/A
	10	N/A
	11	N/A
TIM3	00	ETR
	01	LSI_CLK
	10	Reserved
	11	HSE_CLK_DIV_128

3.6 启动配置

从主闪存存储器启动。

3.7 引导程序

出厂后引导程序存放在系统存储器中，可以通过串口进行 ISP 编程。

Preliminary

4 Embedded FLASH 嵌入式闪存

4.1 简介

嵌入式闪存支持高达 128K 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），还有保留的保密空间，提供了特殊应用的场景下的使用。闪存的控制支持读操作、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 128 页（每页 1K 字节）或 32 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 7	页 28	0x0800 7000 - 0x0800 73FF	1K
		页 29	0x0800 7400 - 0x0800 77FF	1K
		页 30	0x0800 7800 - 0x0800 7BFF	1K
		页 31	0x0800 7C00 - 0x0800 7FFF	1K
	写保护区 8	页 32	0x0800 8000 - 0x080083FF	1K

	写保护区 31	页 124	0x0801 F000 - 0x0801 F3FF	1K
		页 125	0x0801 F400 - 0x0801 F7FF	1K
		页 126	0x0801 F800 - 0x0801 FBFF	1K

模块	区块名称	页名称	地址	大小 (字节)
		页 127	0x0801 FC00 - 0x0801 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。信息存储器可分为保密空间、系统存储器 ISP 和选项字节三部分，保密空间用于存储需特别保护的数据与代码，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	保密空间	0x1FFE 1000 - 0x1FFE 25FF	5.5K
	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
0x1FFF F80C	nWRP2	WRP2	0xFFFF
0x1FFF F80E	nWRP3	WRP3	0xFFFF

表 4-4 USER 的位含义

	Bit	Field	Type	Default	Description	FLASH_OBR
RDP	7: 0	RDP	rw	0xA5	0xA5	

	Bit	Field	Type	Default	Description	FLASH_OBR
nRDP	15: 8	nRDP	rw	0x5A	0x5A	FLASH_OBR. Bit1
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	nRST_STANDBY	rw	0x01	0: 当进入待机模式时产生复位 1: 进入待机模式时不产生复位	FLASH_OBR. Bit4
	3	Reserved	rw	0x01	保留为 0x01	保留
	4	nBOOT1	rw	0x01	0: nBOOT1=0 1: nBOOT1=1	FLASH_OBR. Bit6
	5	Reserved	rw	0x01	保留为 0x01	保留
	6	Reserved	rw	0x01	保留为 0x01	保留
	7	Reserved	rw	0x01	保留为 0x01	保留
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

	Bit	Field	Type	Default	Description	FLASH_OBR
	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 保密空间说明

在保密空间中，Flash 控制器通过指定的密钥配对操作，达到该块内容的读写保护，以避免非法访问。

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打开的。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

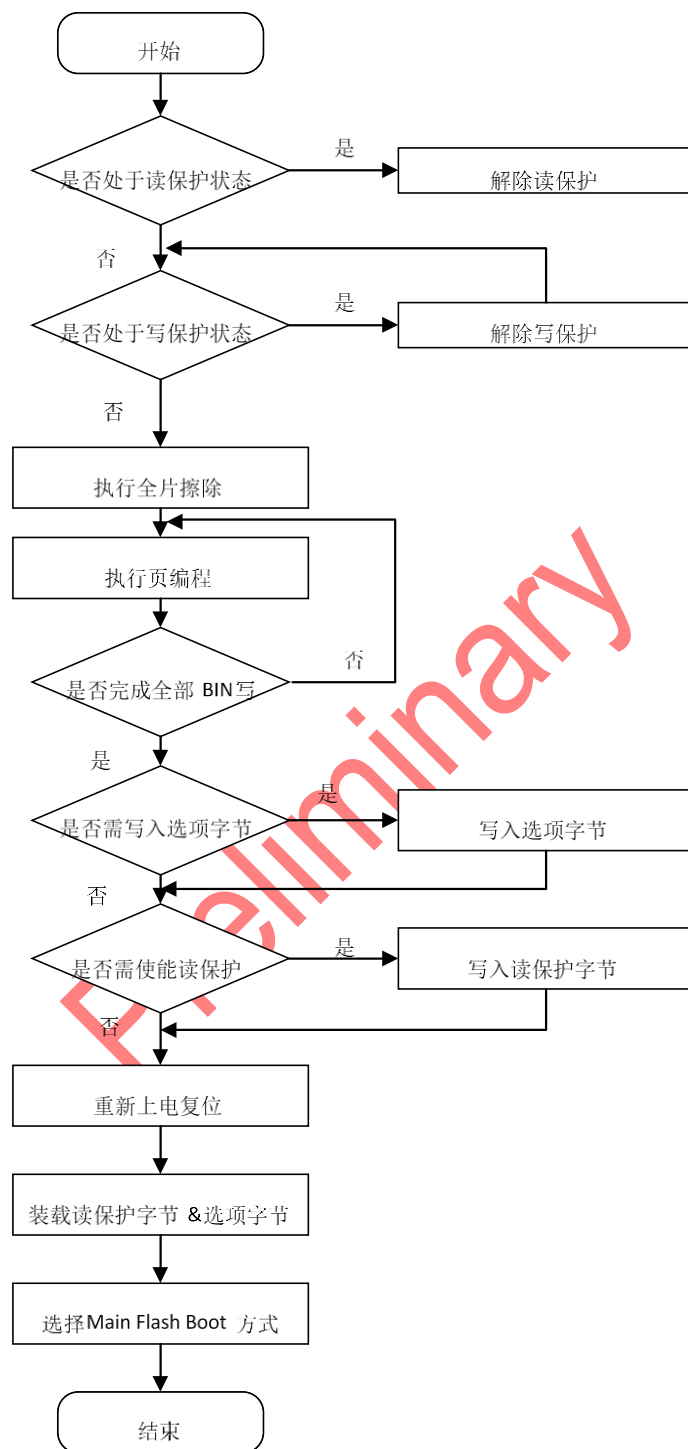


图 4-1 ISP 方式编程流程图

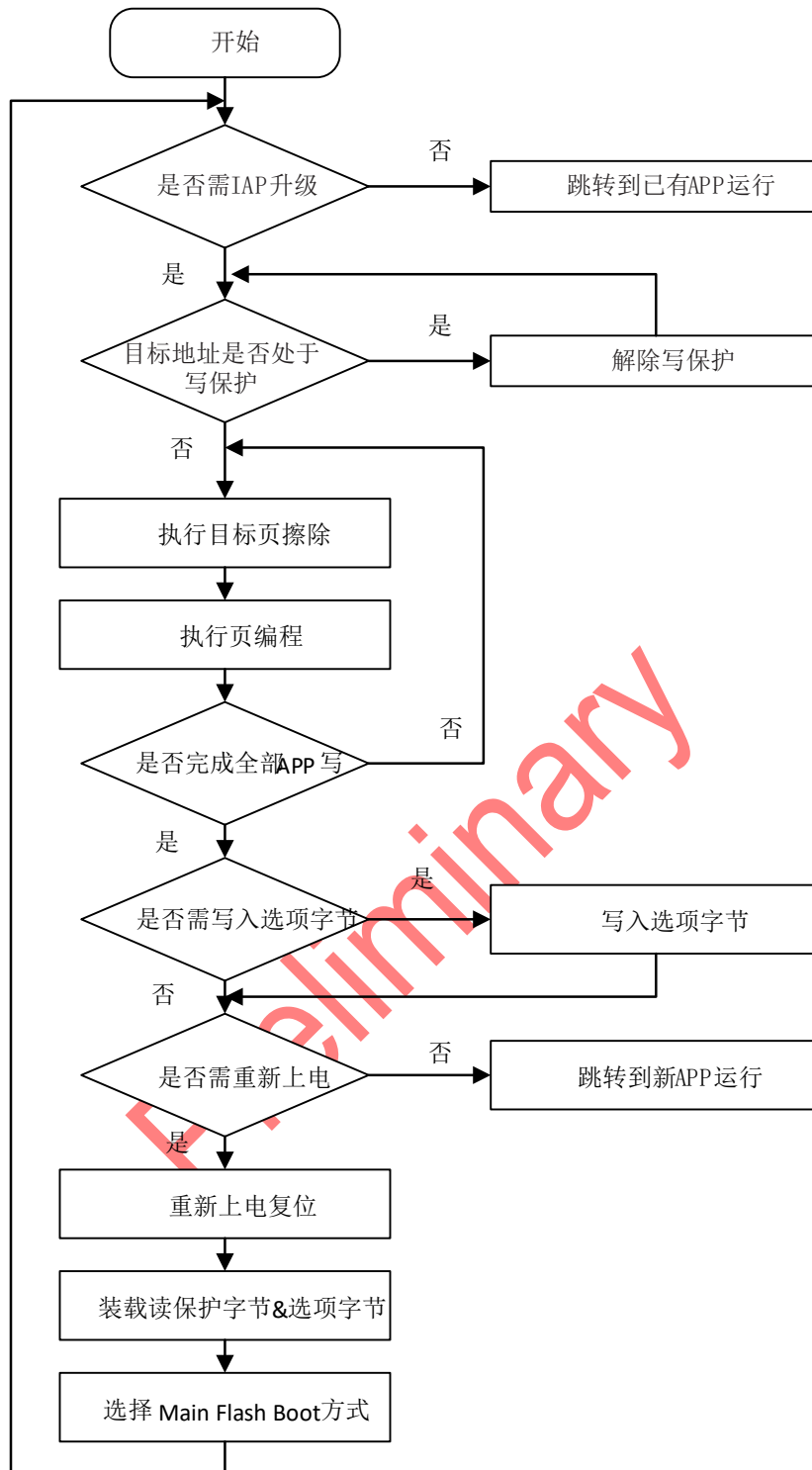


图 4-2 IAP 编程流程

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;
}
```

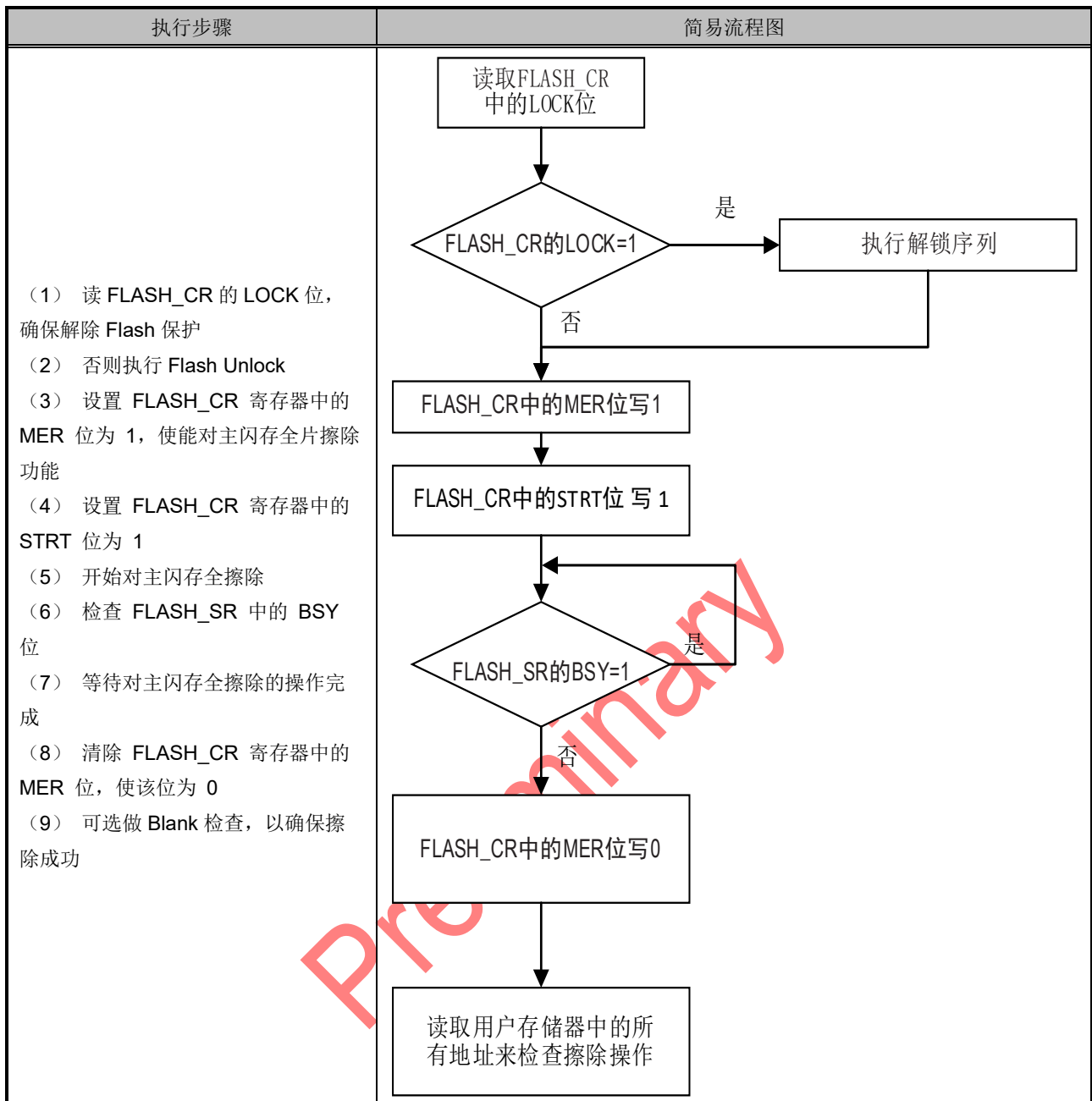
4.3.5 主闪存块擦除

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。

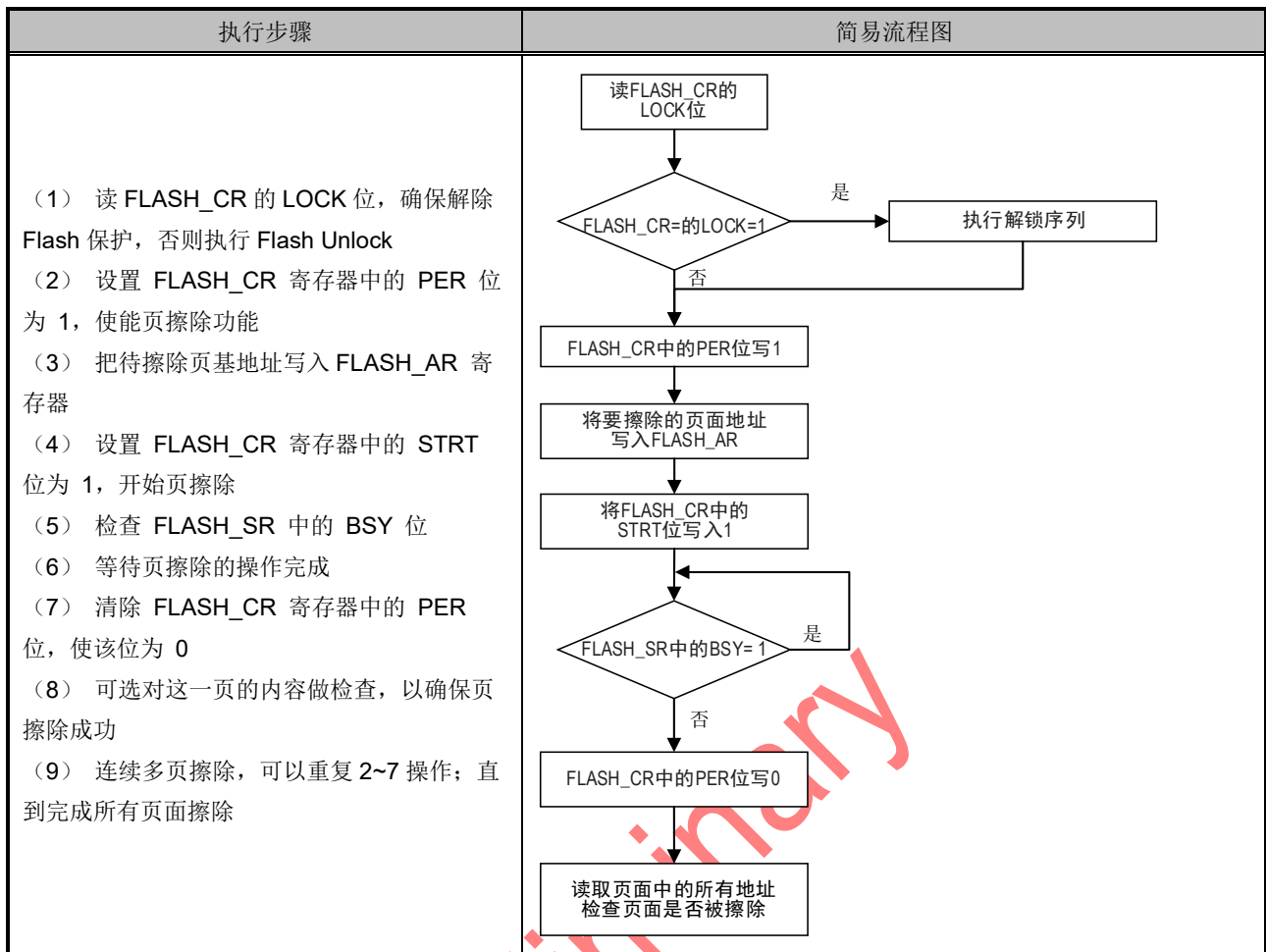
整片擦除功能将初始化主闪存块的所有内容，使所有的值为 0xFFFF，但信息块不会受这个命令影响。

整片擦除操作的寄存器设置，具体步骤如下:

Preliminary



页擦除操作的寄存器设置，具体步骤如下：



4.3.6 主闪存块编程

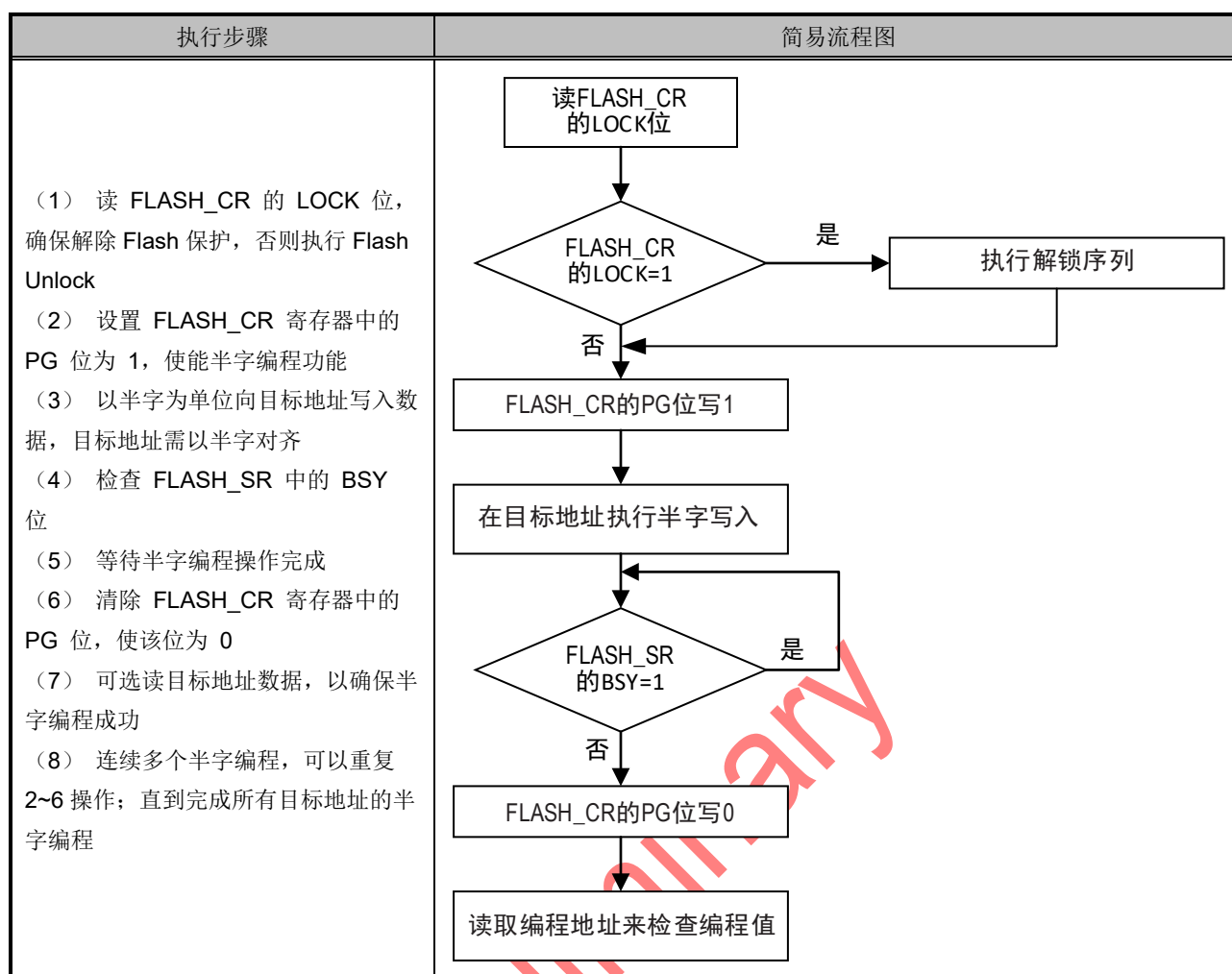
主闪存只支持以 16 位半字编程，用来修改主存储闪存块内容。如果以 32 位整字或 8 字节长度编程，将引起硬件错误中断。当 FLASH_CR 中的 PG 位为 1 时，直接对相应的地址写一个半字（16 位），就是一次编程操作。

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 寄存器中的 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：



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
<p>(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护，否则执行 Flash Unlock</p> <p>(2) 读 FLASH_CR 的 OPTWRE 位，确保解除选项区块保护，否则执行选项区块 Unlock</p> <p>(3) 把待擦除区块基地址写入 FLASH_AR 寄存器</p> <p>(4) 设置 FLASH_CR 寄存器中的 OPTER 位为 1，使能选项字节块擦除功能</p> <p>(5) 设置 FLASH_CR 寄存器中的 STRT 位为 1</p> <p>(6) 检查 FLASH_SR 中的 BSY 位</p> <p>(7) 等待选项字节块擦除操作完成</p> <p>(8) 清除 FLASH_CR 寄存器中的 OPTER 位，设置值为 0</p> <p>(9) 可选对这一区块的内容做检查，以确保对该区块擦除成功</p> <p>(10) 连续多块擦除，可以重复 3~8 操作；直到完成所有选项区块擦除</p>	<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] C --> D{FLASH_CR的OPTWRE=0} B -- 否 --> D D -- 是 --> E[执行解锁OPT序列] E --> F[将闪存选项字节块基地址写入FLASH_AR] F --> G[FLASH_CR中的OPTER位写1] G --> H[FLASH_CR的STRT位写1] H --> I{FLASH_SR的BSY=1} I -- 是 --> I I -- 否 --> J[FLASH_CR中的OPTER位写0] J --> K[通过读取闪存选项存储器中的所有地址来检查闪存选项擦除操作] </pre>

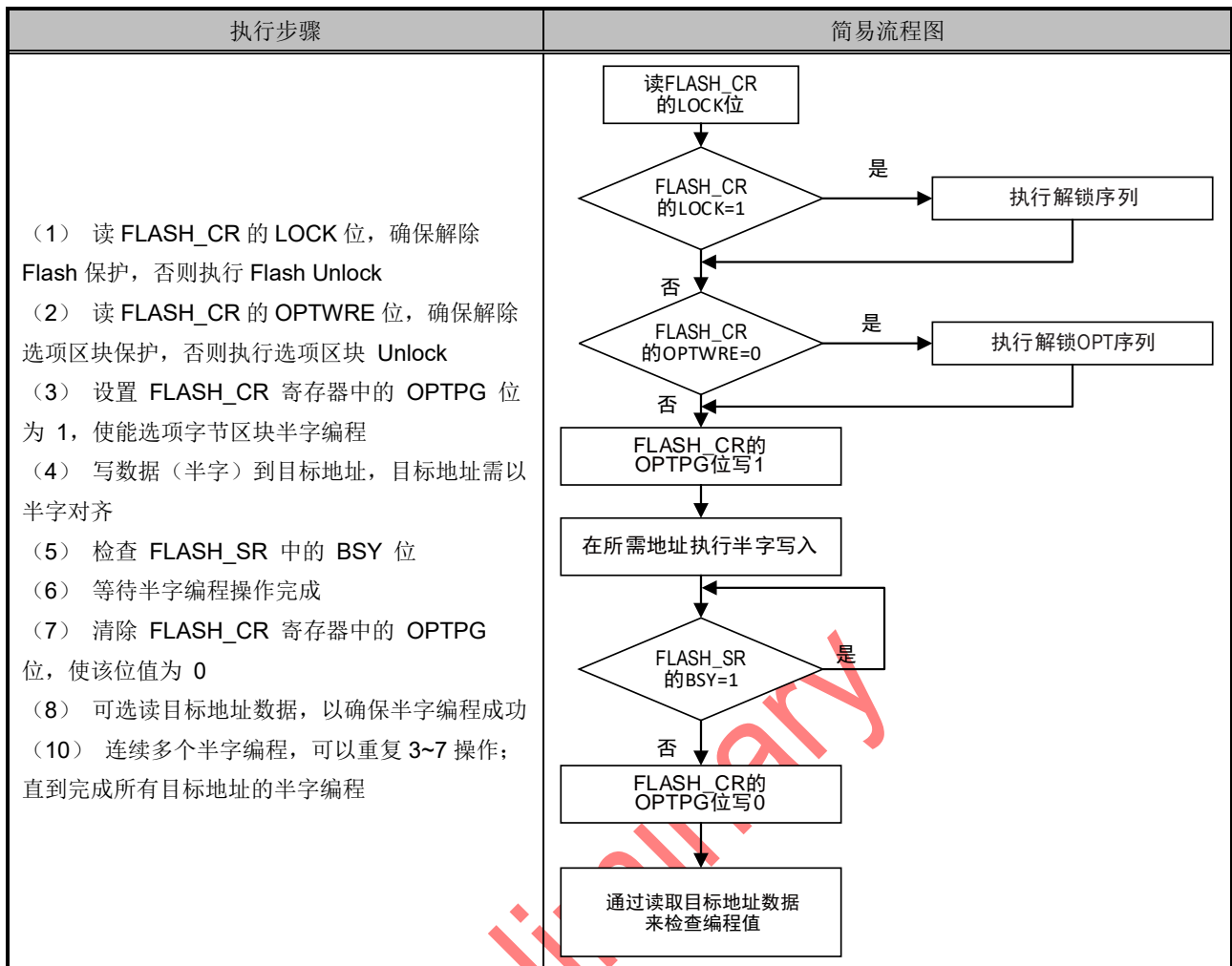
4.3.8 选项字节区块编程

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：



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~WRP3 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	
0x1FFF F80A	nWRP1	WRP1	0xFFFF	
0x1FFF F80C	nWRP2	WRP2	0xFFFF	
0x1FFF F80E	nWRP3	WRP3	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	0xFFFFFFFF

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.										PRFTBS	PRFTBE	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; 注释 2: 当 RCC_CFGR 中 HPRE[3]位设置为非 0 值时, 预取缓冲区硬件自动使能, PRFTBS 位置 1。
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	19	18	17	16	15
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	WRPRERR	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	WRPRERR	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

Bit	Field	Description
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	27	26	25	24	23	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.						OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
						rw		rw	rw	rw	rw		rw	rw	rw

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。

Bit	Field	Description
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						Res.		nBOOT1	Res.	nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR	
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: 7	Res.	保留, 必须保持复位值
6	nBOOT1	nBOOT1

Bit	Field	Description
5	Res.	保留，必须保持复位值
4	nRST_STANDBY	进入待机模式时的复位事件 0: 当进入待机模式时产生复位 1: 进入待机模式时不产生复位
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：该位为只读。

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

Bit	Field	Description
31: 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 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。

5.1 供电系统

芯片的电源分配分为以下四个部分：

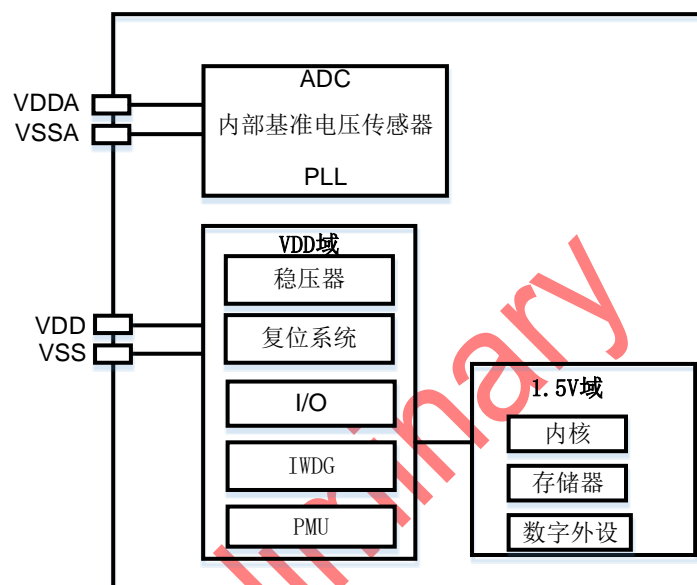


图 5-1 电源控制功能框图

- 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、内部温度传感器和 PLL 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。

5.1.1 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、复位系统和 PLL 等，因此电源的稳定性影响模拟模块的工作的性能。

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

5.1.2 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。

5.1.3 VDD 域

VDD 域主要给 PMU、IWDG 供电，在上电后保持工作状态。

5.1.4 1.5V 域

1.5V 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，芯片会硬件选择关闭该 1.5V 域，在唤醒后芯片会自动开启。主要有以下几种工作状态：

运行模式（Run Mode）：1.5V 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）：1.5V 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）：1.5V 域以低功耗模式工作，只保持寄存器和 SRAM 的内容。

深度停机模式（DeepStop Mode）：1.5V 域以更低功耗模式工作，只保持寄存器和 SRAM 的内容。

待机模式（Standby Mode）：1.5V 域停止供电。除了备用电路和备份域外，其他所有寄存器和 SRAM 的内容全部丢失。

5.2 电源管理器

5.2.1 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。

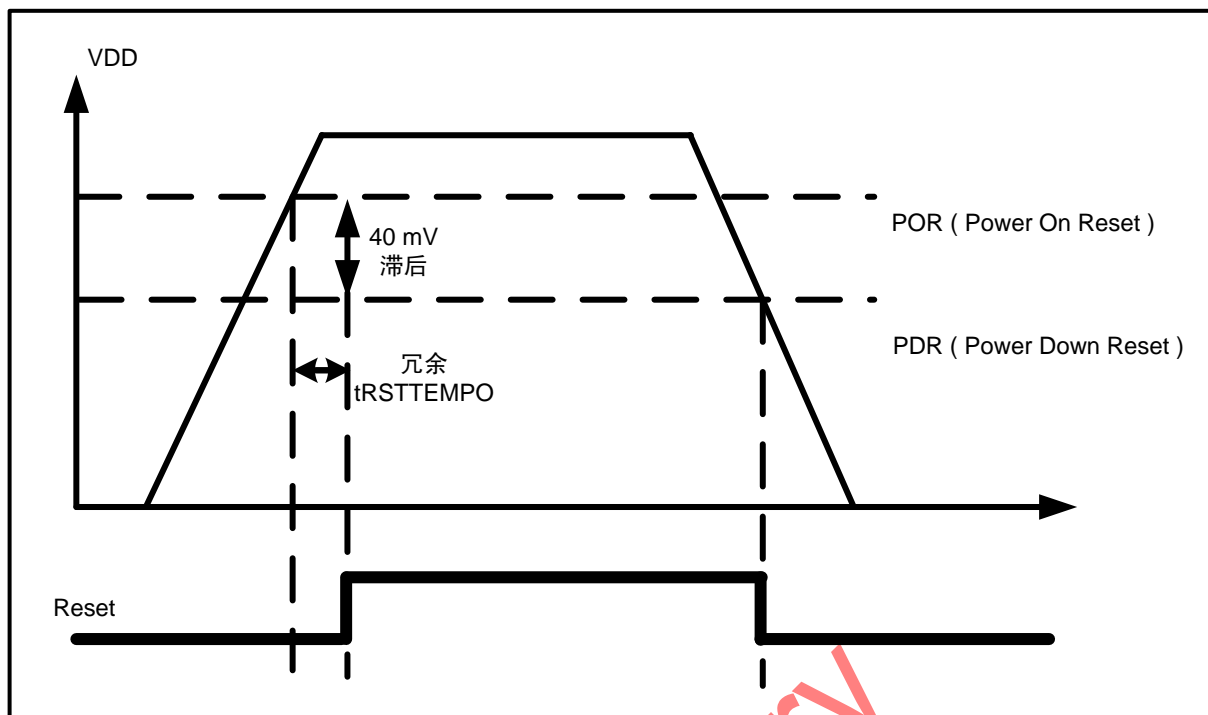


图 5-2 上电复位和掉电复位波形图

5.2.2 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。

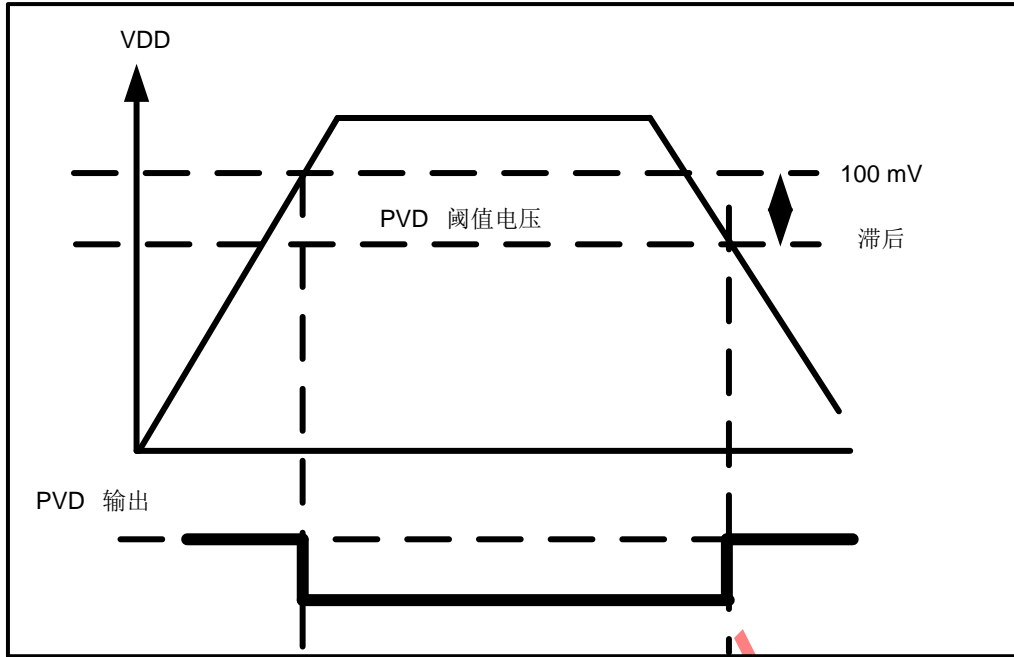


图 5-3 PVD 阈值波形图

5.3 功耗控制

5.3.1 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。

芯片有四种低功耗模式，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。

四种低功耗模式为：

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)： 1.5V 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 SRAM 的内容。

深度停机模式 (DeepStop Mode)： 1.5V 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 SRAM 的内容。

待机模式 (Standby Mode)： 1.5V 域停止供电，寄存器和 SRAM 的内容全部丢失。

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 5-1 低功耗模式列表

模式	进入方式	唤醒方式	对 1.5V 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 1.5V 域的时钟都已关闭	PLL、HSI 和 HSE 的振荡器关闭	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)	清除 PDDS 位；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
待机模式 (Standby Mode)	设置 PDDS 位；SLEEPDEEP 位；WFI 或 WFE；	WKUP 引脚、NRST 引脚上的外部复位、IWDG 复位			关	寄存器和 SRAM 的内容全部丢失	唤醒相当于芯片复位

5.3.2 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、PCLK2）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。

5.3.3 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）、APB2 外设时钟使能寄存器（RCC_APB2ENR）和 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。

5.3.4 Sleep Mode 睡眠模式

5.3.4.1 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。

5.3.4.2 睡眠模式退出

中断或事件发生后，睡眠模式立即被唤醒。

表 5-2 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 5-3 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

SLEEP ON EXIT 模式	说明
	SLEEPONEXIT = 1
退出	中断（参考中断向量表）
唤醒延时	立即唤醒

5.3.5 Stop Mode 停机模式

CPU 深度睡眠模式和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，1.5V 域的所有时钟都被停止，PLL、HSI 和 HSE 振荡器的功能被禁止，SRAM 和寄存器内容将被保留下来。

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。

5.3.5.1 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 PDDS = 0，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 PDDS = 0，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 和 LSIOE 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。

5.3.5.2 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 5-4 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 PDDS 位和 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

5.3.6 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，1.5V 域的所有时钟都被停止，PLL、HSI 和 HSE 振荡器的功能被禁止，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。

5.3.6.1 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 PDDS = 0，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 PDDS = 0，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 LSIOE 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。

5.3.6.2 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 5-5 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 PDDS 和 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 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 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	在进入深度停机模式时需将不使用的 GPIO 设置成模拟输入模式

5.3.7 Standby Mode 待机模式

待机模式是在 CPU 深睡眠模式的基础上关闭电压稳压器。整个 1.5V 可关断域被断电，PLL、HSI 和 HSE 振荡器也被断电，SRAM 和寄存器内容丢失，只有备份域的寄存器和待机电路维持供电。待机模式是芯片的最低功耗模式，唤醒后芯片将复位。

5.3.7.1 待机模式进入

通过对独立的控制位进行编程，待机模式进入方式：

配置电源控制寄存器 (PWR_CR) 的 PDDS = 1；CPU 系统控制寄存器 (SCR) 的 SLEEPDEEP=1。当 WFI/WFE 被执行时，MCU 立即进入待机模式。

进入待机模式时可选择以下功能：

- 独立看门狗 (IWDG)：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，可以通过独立看门狗的复位唤醒方式来唤醒芯片，唤醒后程序重新开始执行；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 LSIOE

位来设置。

5.3.7.2 待机模式退出

当待机模式被唤醒事件唤醒且退出后，产生待机复位。SRAM 和寄存器内容丢失，只有备份域的寄存器保留，系统从待机模式退出时，将会有一段启动延时。

表 5-6 待机模式

待机模式	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； 置位电源控制寄存器 (PWR_CR) 中的 PDDS 位； 复位电源状态寄存器 (PWR_SR1) 中的 WUF 位； 注意：为确保芯片可以被正常唤醒，进入 Standby 模式前需关闭中断使能。
退出	WKUP 引脚上升沿或者下降沿、NRST 引脚外部复位、IWDG 复位
唤醒延时	复位阶段时电压稳压器的启动
注意事项	在进入待机模式时需配置正确的唤醒源，否则进入待机模式后可能无法正常唤醒而造成二次下载

待机模式下的输入/输出端口状态

在待机模式下，所有的 I/O 引脚处于高阻态，除了以下的引脚：

- 复位引脚
- 被使能的唤醒引脚，用户需要正确的配置对应的唤醒边沿信号

调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止或待机模式，将失去调试连接。

5.3.7.3 快速唤醒功能

通过配置 PWR_CR 寄存器的 STDBY_FS_WK 位,可以调节从唤醒事件发生到唤醒事件有效的时间。

5.4 电源控制寄存器

访问：与标准的 APB 读写相比，读写 PWR 寄存器需要额外的 APB 周期

表 5-7 电源控制寄存器概览

Offset	Acronym	RegisterName	Reset
0x00	PWR_CR	电源控制寄存器	0x00000600
0x04	PWR_CSR	电源控制状态寄存器	0x00000000
0x0C	PWR_CR1	电源控制寄存器 1	0x00000000
0x10	PWR_SR1	电源状态寄存器 1	0x00000000
0x14	PWR_SCR	电源状态清除寄存器	0x00000000
0x24	PWR_CFGR	电源配置寄存器	0x00000160

5.4.1 PWR_CR 电源控制寄存器

地址偏移: 0x00

复位值: 0x00000600 (从待机模式唤醒时清除)

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	
STDBY_FS_WK		Res	PLS				Res	Res			PVDE	CSBF	CWUF	PDDS	LPDS	
rw			rw									rw	rc_w1	rc_w1	rw	rw

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 14	STDBY_FS_WK	快速唤醒 STANDBY 模式选择位: 00: 9 个 LSI40K 周期唤醒 standby 01: 7 个 LSI40K 周期唤醒 standby 10: 5 个 LSI40K 周期唤醒 standby 11: 2 个 LSI40K 周期唤醒 standby 注: 使用快速唤醒模式必须先开启 LSI_CLK
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: 1.8V 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	CSBF	清除待机标志位 (Clear Standby Flag) 始终读出为 0 1: 清除 SBF 待机位 (写) 0: 无功效
2	CWUF	清除唤醒标志位 (Clear Wakeup flag) 始终读出为 0 1 = 清除 WUF 唤醒位 (写) 0 = 无功效

Bit	Field	Description
1	PDDS	掉电深睡眠（Power Down Deepsleep） 1: CPU 进入深睡眠时进入待机模式 0: CPU 进入深睡眠时进入停机模式
0	LPDS	深睡眠下的低功耗（Low Power Deepstop） PDDS = 0 时，与 PDDS 位协同操作 1: 进入停机模式时，电压稳压器处于低功耗模式。 0: 进入停机模式时，电压稳压器处于正常功耗模式。 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。

5.4.2 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.		EWUP6	EWUP5	EWUP4	Res.		EWUP2	EWUP1	Res				PVDO	SBF	WUF
		rw	rw	rw			rw	rw					r	r	r

Bit	Field	Description
31: 14	Reserved	保留，始终读为 0
13	EWUP6	使能 WKUP6 引脚（Enable WKUP6 pin） 1 = WKUP6 引脚用于将 CPU 从待机模式唤醒，WKUP6 引脚被强置为输入下拉的配置（WKUP6 引脚上的上升沿将系统从待机模式唤醒） 0 = WKUP6 引脚为通用 I/O。WKUP6 引脚上的事件不能将 CPU 从待机模式唤醒 注：在系统复位时清除这一位。
12	EWUP5	使能 WKUP5 引脚（Enable WKUP5 pin） 1 = WKUP5 引脚用于将 CPU 从待机模式唤醒，WKUP5 引脚被强置为输入下拉的配置（WKUP5 引脚上的上升沿将系统从待机模式唤醒） 0 = WKUP5 引脚为通用 I/O。WKUP5 引脚上的事件不能将 CPU 从待机模式唤醒 注：在系统复位时清除这一位。
11	EWUP4	使能 WKUP4 引脚（Enable WKUP 4pin） 1 = WKUP4 引脚用于将 CPU 从待机模式唤醒，WKUP4 引脚被强置为输入下拉的配置（WKUP4 引脚上的上升沿将系统从待机模式唤醒） 0 = WKUP4 引脚为通用 I/O。WKUP4 引脚上的事件不能将 CPU 从待机模式唤醒 注：在系统复位时清除这一位。
10	Reserved	保留，始终读为 0

Bit	Field	Description
9	EWUP2	使能 WKUP2 引脚 (Enable WKUP2 pin) 1 = WKUP2 引脚用于将 CPU 从待机模式唤醒, WKUP2 引脚被强置为输入下拉的配置 (WKUP2 引脚上的上升沿将系统从待机模式唤醒) 0 = WKUP2 引脚为通用 I/O。WKUP2 引脚上的事件不能将 CPU 从待机模式唤醒 注: 在系统复位时清除这一位。
8	EWUP1	使能 WKUP1 引脚 (Enable WKUP1 pin) 1 = WKUP1 引脚用于将 CPU 从待机模式唤醒, WKUP1 引脚被强置为输入下拉的配置 (WKUP1 引脚上的上升沿将系统从待机模式唤醒) 0 = WKUP1 引脚为通用 I/O。WKUP1 引脚上的事件不能将 CPU 从待机模式唤醒 注: 在系统复位时清除这一位。
7: 3	Reserved	保留, 始终读为 0
2	PVDO	PVD 输出 (PVD output) 当 PVD 被 PVDE 位使能后该位才有效。 1 = VDD/VDDA 低于由 PLS[3: 0]选定的 PVD 阈值 0 = VDD/VDDA 高于由 PLS[3: 0]选定的 PVD 阈值 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。
1	SBF	待机标志 (Standby Flag) 该位由硬件设置, 并只能由 POR/PDR (上电/掉电复位) 或设置电源控制寄存器 (PWR_CR) 的 CSBF 位清除。 1: 系统进入待机模式 0: 系统不在待机模式
0	WUF	唤醒标志 (Wakeup flag) 该位由硬件设置, 并只能由 POR/PDR (上电/掉电复位) 或设置电源控制寄存器 (PWR_CR) 的 CWUF 位清除。 1 = 在 WKUP 引脚上发生唤醒事件 0 = 没有发生唤醒事件 注: 当 WKUP 引脚已经是高电平时, 在 (通过设置 EWUP 位) 使能 WKUP 引脚时, 会检测到一个额外的事件。

5.4.3 PWR_CR1 电源控制寄存器 1

地址偏移: 0x0C

复位值: 0x00000000

寄存器在退出待机模式时不进行复位, 也不会 RCC_APB1RSTR.PWR/DBG 有效后复位

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
Res.										WKP_E DGE6	WKP_E DGE5	WKP_E DGE4	Res.	WKP_E DGE2	WKP_E DGE1

Bit	Field	Description
5	WUF6	唤醒标志 6 当 wkup6 唤醒引脚检测到唤醒事件时设置此位。通过在 PWR_SCR 寄存器的 wuf6_clr 位上写入 1 来清除它。
4	WUF5	唤醒标志 5 当 wkup5 唤醒引脚检测到唤醒事件时设置此位。通过在 PWR_SCR 寄存器的 wuf5_clr 位上写入 1 来清除它。
3	WUF4	唤醒标志 4 当 wkup4 唤醒引脚检测到唤醒事件时设置此位。通过在 PWR_SCR 寄存器的 wuf4_clr 位上写入 1 来清除它。
2	Reserved	保留，始终读为 0
1	WUF2	唤醒标志 2 当 wkup2 唤醒引脚检测到唤醒事件时设置此位。通过在 PWR_SCR 寄存器的 wuf2_clr 位上写入 1 来清除它。
0	WUF1	唤醒标志 1 当 wkup1 唤醒引脚检测到唤醒事件时设置此位。通过在 PWR_SCR 寄存器的 wuf1_clr 位上写入 1 来清除它。

5.4.5 PWR_SCR 电源状态清除寄存器

地址偏移：0x14

复位值：0x00000000

访问：写寄存器需要额外的 3 个 APB 周期，读需要额外的 2 个 APB 周期。

寄存器在退出待机模式时不进行复位，也不会 RCC_APB1RSTR.PWR/DBG 有效后复位

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
Res.										WUF6_ CLR	WUF5_ CLR	WUF4_ CLR	Res.	WUF2_ CLR	WUF1_ CLR
										rw	rw	rw		rw	rw

Bit	Field	Description
31: 6	Reserved	保留，始终读为 0
5	WUF6_CLR	清除唤醒标志 6 将该位置 1 会清除 PWR_SR1 寄存器中的 wuf6 标志。
4	WUF5_CLR	清除唤醒标志 5 将该位置 1 会清除 PWR_SR1 寄存器中的 wuf5 标志。
3	WUF4_CLR	清除唤醒标志 4 将该位置 1 会清除 PWR_SR1 寄存器中的 wuf4 标志。
2	Reserved	保留，始终读为 0

Bit	Field	Description
1	WUF2_CLR	清除唤醒标志 2 将该位置 1 会清除 PWR_SR1 寄存器中的 wuf2 标志。
0	WUF1_CLR	清除唤醒标志 1 将该位置 1 会清除 PWR_SR1 寄存器中的 wuf1 标志。

5.4.6 PWR_CFGR 电源配置寄存器

地址偏移: 0x24

复位值: 0x00000160

31	30	29	28	27	26	25	24	23	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: 6	Reserved	保留, 始终读为 0
9: 5	LSICAL	内部低速时钟校准位 (Internal Low-speed Clock Calibration Bits) 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。
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

6 SYSCFG 系统控制器

6.1 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

6.2 寄存器

6.2.1 寄存器总览

表 6-1 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

6.2.2 SYSCFG_CFGR 配置寄存器

该寄存器具有两位控制位 MEM_MODE，可以用来配置不同存储器到起始地址 0x00000000 的映射。软件配置这两个控制位可用来屏蔽 BOOT 的选择。复位后，此控制位值由实际的 BOOT PIN 配置决定。

偏移地址：0x00

复位值：0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TIM3_T	TIM3_C	TIM2_C	TIM2_C	TIM1_C	TIM1_C	TIM1_C	I2C1_RX	I2C1_TX	UART3_	UART3_	UART2_	UART2_	SPI2_R
		RIG_DM	H1_DMA	H4_DMA	H2_DMA	H3_DMA	H2_DMA	H1_DMA	_DMA_R	_DMA_R	RX_DM	TX_DMA	RX_DM	TX_DMA	X_DMA_
		A_RMP	_RMP	_RMP	_RMP	_RMP	_RMP	_RMP	MP	MP	A_RMP	_RMP	A_RMP	_RMP	RMP
		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
SPI2_TX _DMA_R MP	Res.		TIM17_ DMA_R MP	TIM16_ DMA_R MP	UART1_ RX_DM A_RMP	UART1_ TX_DMA _RMP	ADC_D MA_RM P	Res.						MEM_MODE	

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

Bit	Field	Description
31: 30	Reserved	保留，必须保持复位值
29	TIM3_TRIG_DMA_RMP	TIM3_TRIG 的 DMA 触发重映射 0: TIM3_TRIG 的 DMA 触发功能映射到通道 4 1: TIM3_TRIG 的 DMA 触发功能映射到通道 6
28	TIM3_CH1_DMA_RMP	TIM3_CH1 的 DMA 触发重映射 0: TIM3_CH1 的 DMA 触发功能映射到通道 4 1: TIM3_CH1 的 DMA 触发功能映射到通道 6
27	TIM2_CH4_DMA_RMP	TIM2_CH4 的 DMA 触发重映射 0: TIM2_CH4 的 DMA 触发功能映射到通道 4 1: TIM2_CH4 的 DMA 触发功能映射到通道 7
26	TIM2_CH2_DMA_RMP	TIM2_CH2 的 DMA 触发重映射 0: TIM2_CH2 的 DMA 触发功能映射到通道 3 1: TIM2_CH2 的 DMA 触发功能映射到通道 7
25	TIM1_CH3_DMA_RMP	TIM1_CH3 的 DMA 触发重映射 0: TIM1_CH3 的 DMA 触发功能映射到通道 5 1: TIM1_CH3 的 DMA 触发功能映射到通道 6
24	TIM1_CH2_DMA_RMP	TIM1_CH2 的 DMA 触发重映射 0: TIM1_CH2 的 DMA 触发功能映射到通道 3 1: TIM1_CH2 的 DMA 触发功能映射到通道 6
23	TIM1_CH1_DMA_RMP	TIM1_CH1 的 DMA 触发重映射 0: TIM1_CH1 的 DMA 触发功能映射到通道 2 1: TIM1_CH1 的 DMA 触发功能映射到通道 6
22	I2C1_RX_DMA_RMP	I2C1_RX 的 DMA 触发重映射 0: I2C1_RX 的 DMA 触发功能映射到通道 3 1: I2C1_RX 的 DMA 触发功能映射到通道 7
21	I2C1_TX_DMA_RMP	I2C1_TX 的 DMA 触发重映射 0: I2C1_TX 的 DMA 触发功能映射到通道 2 1: I2C1_TX 的 DMA 触发功能映射到通道 6
20	UART3_RX_DMA_RMP	UART3_RX 的 DMA 触发重映射 0: UART3_RX 的 DMA 触发功能映射到通道 3 1: UART3_RX 的 DMA 触发功能映射到通道 6
19	UART3_TX_DMA_RMP	UART3_TX 的 DMA 触发重映射 0: UART3_TX 的 DMA 触发功能映射到通道 2 1: UART3_TX 的 DMA 触发功能映射到通道 7
18	UART2_RX_DMA_RMP	UART2_RX 的 DMA 触发重映射 0: UART2_RX 的 DMA 触发功能映射到通道 5 1: UART2_RX 的 DMA 触发功能映射到通道 6
17	UART2_TX_DMA_RMP	UART2_TX 的 DMA 触发重映射 0: UART2_TX 的 DMA 触发功能映射到通道 4 1: UART2_TX 的 DMA 触发功能映射到通道 7

Bit	Field	Description
16	SPI2_RX_DMA_RMP	SPI2_RX 的 DMA 触发重映射 0: SPI2_RX 的 DMA 触发功能映射到通道 4 1: SPI2_RX 的 DMA 触发功能映射到通道 6
15	SPI2_TX_DMA_RMP	SPI2_TX 的 DMA 触发重映射 0: SPI2_TX 的 DMA 触发功能映射到通道 5 1: SPI2_TX 的 DMA 触发功能映射到通道 7
14: 13	Reserved	保留, 必须保持复位值
12	TIM17_DMA_RMP	TIM17 DMA 重映射 (TIM17 DMA Remap) 0: TIM17 的 DMA 功能映射到 channel1 1: TIM17 的 DMA 功能重映射到 channel2
11	TIM16_DMA_RMP	TIM16DMA 重映射 (TIM16 DMA Remap) 0: TIM16 的 DMA 功能映射到 channel3 1: TIM16 的 DMA 功能重映射到 channel4
10	UART1_RX_DMA_RMP	UART1 RX DMA 重映射 (UART1 RX DMA Remap) 0: UART1_RX 的 DMA 功能映射到 channel3 1: UART1_RX 的 DMA 功能重映射到 channel5
9	UART1_TX_DMA_RMP	UART1 TX DMA 重映射 (UART1 TX DMA Remap) 0: UART1_TX 的 DMA 功能映射到 channel2 1: UART1_TX 的 DMA 功能重映射到 channel4
8	ADC_DMA_RMP	ADC DMA 重映射 (ADC DMA Remap) 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) 控制存储器内部映射到地址 0x0000 0000。 由软件设置和清除这些位。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

6.2.3 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 0010: PC[x] 管脚 0011: PD[x] 管脚

6.2.4 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 0010: PC[x] 管脚 0011: PD[x] 管脚

6.2.5 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 0010: PC[x] 管脚 0011: PD[x] 管脚

6.2.6 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
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 0011: PD[x] 管脚

6.2.7 SYSCFG_PADHYS PAD 配置寄存器

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.															I2C1_M ODE_SE L 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
15: 0	Reserved	保留, 必须保持复位值

Preliminary

7 RCC 时钟和复位

7.1 复位单元

7.1.1 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。

7.1.2 功能框图

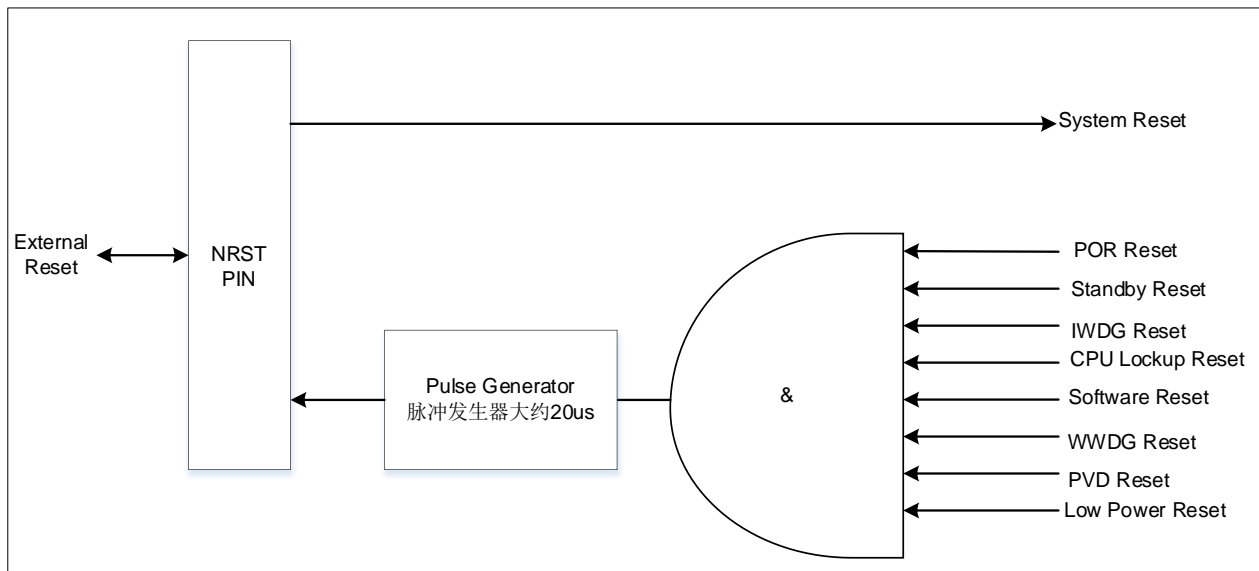


图 7-1 复位功能框图

7.1.3 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）中的待机和唤醒标志、DBG 控制寄存器（DBG_CR）、备份区域中的寄存器不受系统复位影响，其余寄存器都将被系统复位。

7.1.4 功能描述

7.1.4.1 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)
- 待机复位 (Standby Reset)

7.1.4.2 系统复位 (System Reset)

系统复位有以下几种方式:

- 外部复位 (NRST Reset)
- 窗口看门狗复位 (WWDG 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。

窗口看门狗复位 (WWDG Reset):

- 当计数器的值大于配置寄存器 (WWDG_CFGR) 设定的值时喂狗, 将会发生窗口看门狗复位。
- 当计数器的值小于 0x40 时, 将会发生窗口看门狗复位。
- 具体请参考窗口看门狗章节

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_STDBY 位或 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_STDBY 位为 0, 系统将被复位, 而不是进入待机模式。
- ◆ 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或者深度停机模式。
- 具体请参考嵌入式闪存章节

备份域复位 (Backup Reset)

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

7.2 时钟单元

7.2.1 简介

四个可配置的独立系统时钟源:

- 外部高速时钟 (HSE)
- 内部高速时钟 (HSI)
- 锁相环 1 (PLL1)
- 内部低速时钟 (LSI)

可配置的独立外设时钟源：

- 锁相环 2 (PLL2)

7.2.2 功能框图

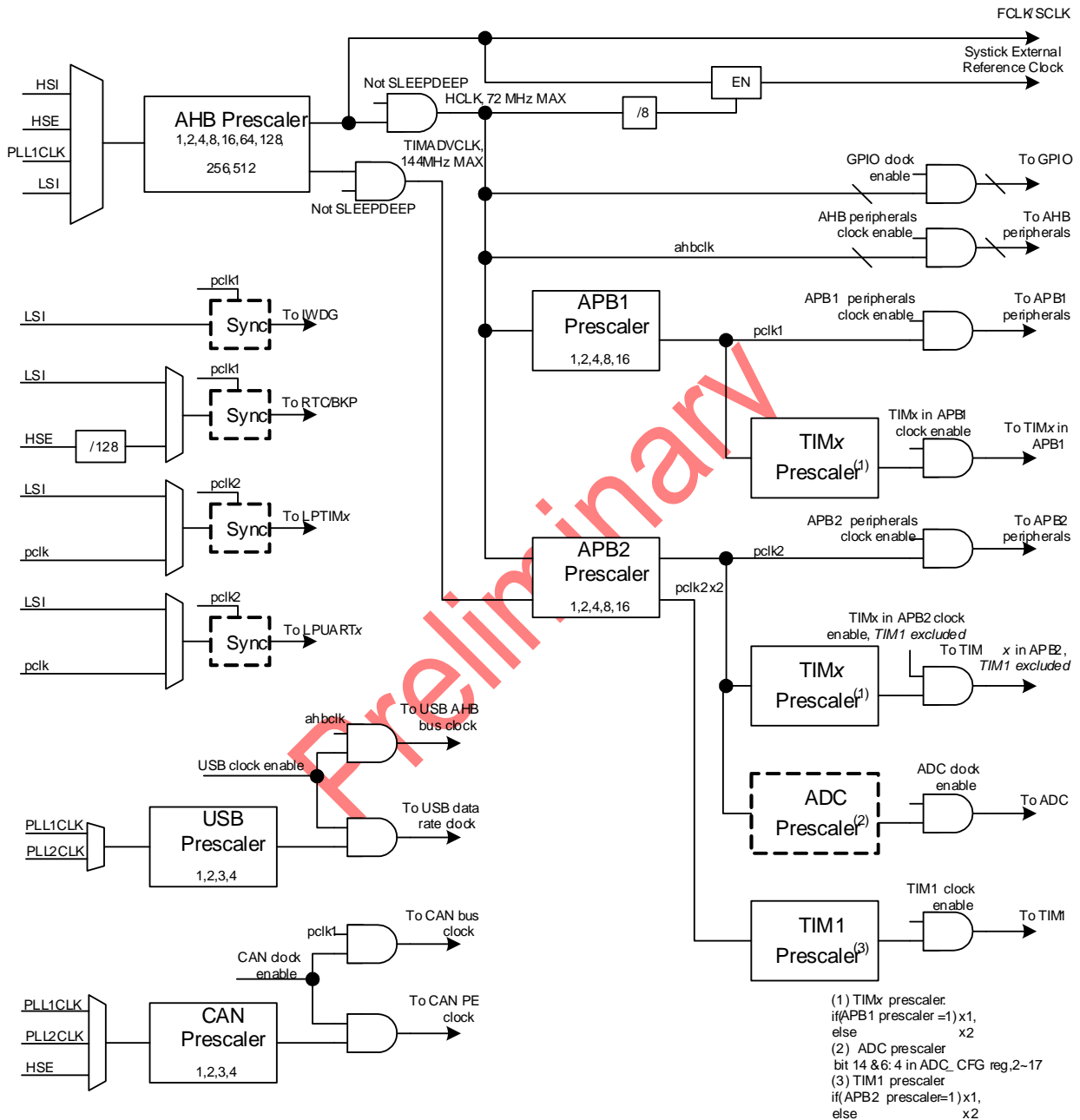


图 7-2 时钟树

7.2.3 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 和 APB1, APB2 总线时钟的最大频率是 72MHz。

7.2.4 功能描述

7.2.4.1 外部高速时钟（HSE）

外部高速时钟的时钟源有以下两种：

- 外部高速晶振/陶瓷谐振器输入时钟

其具有非常精确的特点。

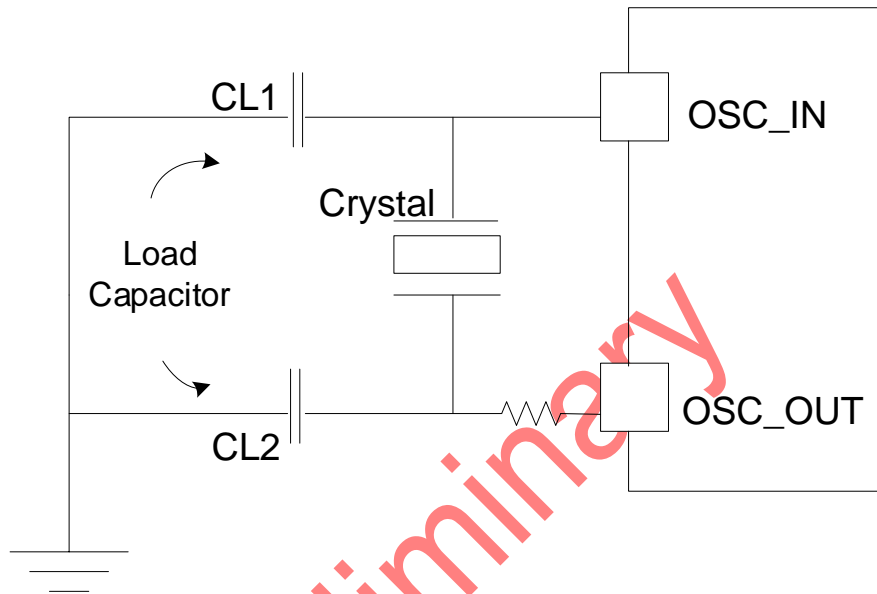


图 7-3 高速晶振/陶瓷谐振器

上图为外部高速晶振/陶瓷谐振器输入模块框图

- 外部高速输入时钟

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

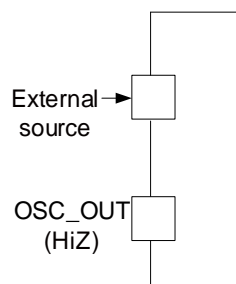


图 7-4 外部高速输入时钟

上图为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或晶体振荡器
- 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
- 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：

- 当使用外部高速输入时钟时，必须将时钟控制寄存器（RCC_CR）的 HSEBYP 位置“1”。
- 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。

7.2.4.2 内部高速时钟（HSI）

HSI 时钟信号由内部 8MHz 振荡器产生，HSI 时钟源在芯片上电后默认启用。

优点：HSI 振荡器能够在不需要任何外部器件的条件下提供系统时钟，启动时间比 HSE 晶体振荡器快。

缺点：即使经过校准，频率也不如外部晶体振荡器或陶瓷谐振器精确。

HSI 具有如下作用：

- 可直接用作系统时钟
- 可以作为 PLL 输入时钟源
- 如果 HSE 晶体振荡器发生故障，HSI 信号也可以用作备用时钟源，参考时钟安全系统(CSS) 章节

使能 HSI 配置步骤：

- 配置时钟控制寄存器（RCC_CR）中的 HSION 位为 1，使能 HSI；
- 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 HSI 启用，HSI 相关配置就不能更改。如需更改配置，必须先禁止 HSI。

7.2.4.3 锁相环 (PLL1)

PLL1 的输入参考时钟有以下三种, 可以通过 PLL1 配置寄存器(RCC_PLL1CFGR)中的 PLL1XTPRE 和 PLL1SRC 位进行选择:

- HSI 振荡器输出时钟
- HSE 晶体输出时钟
- HSE 晶体输出时钟的两分频时钟

PLL1 的输入时钟 f_{PLL1_IN} 和输出时钟 f_{PLL1_OUT} 之间关系为:

$$\frac{f_{PLL1_IN}}{(PLL1DIV[2:0] + 1) * (PLL1PDIV[2:0] + 1)} = \frac{f_{PLL1_OUT}}{PLL1MUL[7:0] + 1}$$

PLL1MUL 和 PLL1PDIV、PLL1DIV 是 PLL1 的倍频器和分频器的分频比设置控制位。

注意事项:

- 在使能 PLL1 之前, 必须进行 PLL1 配置 (选择 PLL1 输入时钟源和选择倍频因子和分频因子)。一旦 PLL1 启用, PLL1 配置就不能被更改。如需更改配置, 必须先禁止 PLL1。
- PLL1 配置寄存器 (RCC_PLL1CFGR) 中 PLL1DIV 建议设置为奇数 (1 或 3 或 5 或 7);

使能 PLL1 配置步骤: (选择 HSI 为 PLL1 时钟源)

- 配置时钟控制寄存器 (RCC_CR) 中的 HSION 位为 1, 使能 HSI;
- 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;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中 PLL1MUL (倍频系数)、PLL1PDIV (PLL1_IN 分频系数) 和 PLL1DIV (分频系数) 控制位;
- 配置时钟控制寄存器 (RCC_CR) 中的 PLL1ON 位为 1, 使能 PLL1;
- 等待时钟控制寄存器 (RCC_CR) 中的 PLL1RDY 位被置位为 1, 表示 PLL1 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。

使能 PLL1 配置步骤: (选择 HSE 为 PLL1 时钟源)

- 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, 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;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中的 PLL1SRC 位为 1, 选择 HSE 时钟用作 PLL1 输入时钟源;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中的 PLL1XTPRE 位, 选择 HSE 或者 HSE 2 分频时钟为 PLL1 输入时钟源;

- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中 PLL1MUL (倍频系数)、PLL1PDIV (PLL1_IN 分频系数) 和 PLL1DIV (分频系数) 控制位;
- 配置时钟控制寄存器 (RCC_CR) 中的 PLL1ON 位为 1, 使能 PLL1;
- 等待时钟控制寄存器 (RCC_CR) 中的 PLL1RDY 位被置位为 1, 表示 PLL1 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。

7.2.4.4 锁相环 2 (PLL2)

PLL2 的输入参考时钟有以下三种, 可以通过 PLL2 配置寄存器 (RCC_PLL2CFGR) 中的 PLL2XTPRE 和 PLL2SRC 位进行选择:

- HSI 振荡器输出时钟
- HSE 晶体输出时钟
- HSE 晶体输出时钟的两分频时钟

PLL2 的输入时钟 f_{PLL2_IN} 和输出时钟 f_{PLL2_OUT} 之间关系为:

$$\frac{f_{PLL2_IN}}{(PLL2DIV[2:0] + 1) * (PLL2PDIV[2:0] + 1)} = \frac{f_{PLL2_OUT}}{PLL2MUL[7:0] + 1}$$

PLL2MUL 和 PLL2PDIV、PLL2DIV 是 PLL2 的倍频器和分频器的分频比设置控制位。

注意事项:

- 在使能 PLL2 之前, 必须进行 PLL2 配置 (选择 PLL2 输入时钟源和选择倍频因子和分频因子)。一旦 PLL2 启用, PLL2 配置就不能被更改。如需更改配置, 必须先禁止 PLL2。
- PLL2 配置寄存器 (RCC_PLL2CFGR) 中 PLL2DIV 建议设置为奇数 (1 或 3 或 5 或 7);

使能 PLL2 配置步骤: (选择 HSI 为 PLL2 时钟源)

- 配置时钟控制寄存器 (RCC_CR) 中的 HSION 位为 1, 使能 HSI;
- 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;
- 配置 PLL2 配置寄存器 (RCC_PLL2CFGR) 中 PLL2MUL (倍频系数)、PLL2PDIV (PLL2_IN 分频系数) 和 PLL2DIV (分频系数) 控制位;
- 配置时钟控制寄存器 (RCC_CR) 中的 PLL2ON 位为 1, 使能 PLL2;
- 等待时钟控制寄存器 (RCC_CR) 中的 PLL2RDY 位被置位为 1, 表示 PLL2 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。

使能 PLL2 配置步骤: (选择 HSE 为 PLL2 时钟源)

- 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，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；
- 配置 PLL2 配置寄存器（RCC_PLL2CFGR）中的 PLL2SRC 位为 1，选择 HSE 时钟用作 PLL2 输入时钟源；
- 配置 PLL2 配置寄存器（RCC_PLL2CFGR）中的 PLL2XTPRE 位，选择 HSE 或者 HSE 2 分频时钟为 PLL2 输入时钟源；
- 配置 PLL2 配置寄存器（RCC_PLL2CFGR）中 PLL2MUL（倍频系数）、PLL2PDIV（PLL2_IN 分频系数）和 PLL2DIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLL2ON 位为 1，使能 PLL2；
- 等待时钟控制寄存器（RCC_CR）中的 PLL2RDY 位被置位为 1，表示 PLL2 稳定，可输出有效时钟，此时才可被选择使用作为外设时钟源。

7.2.4.5 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。

7.2.4.6 中断

表 7-1 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_PLL1RDY	PLL1RDYF	PLL1RDYIE	PLL1RDYC
RCC_HSERDY	HSERDYF	HSERDYIE	HSERDYC
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC
RCC_LSIRDY	LSIRDYF	LSIRDYIE	LSIRDYC

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。

7.2.4.7 系统时钟选择（SWS）

四个系统时钟源：

- 内部高速时钟（HSI 上电后默认）

- 外部高速时钟（HSE）
- 锁相环（PLL1）：当 HSI（或 HSE）作为 PLL1 时钟源时，HSI（或 HSE）不可禁止
- 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（HSI, HSE, PLL1, LSI），每个时钟使能方式不同，具体方式请查看（HSI, HSE, PLL1, 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。

7.2.4.8 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 1us。

7.2.4.9 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_AHBRSTR）来实现相应外设的软件复位。

7.2.4.10 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，如果直接或间接(PLL1 的输入时钟)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(和 PLL1 时钟)。
- 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；
- 配置控制状态寄存器（RCC_CSR）的 LSION 位为 1 和 LSIOE 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；
- 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。

HSE 间接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 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；
- 配置 PLL 配置寄存器（RCC_PLL1CFGR）中的 PLL1SRC 位为 1，选择 HSE 时钟用作 PLL1 输入时钟；
- 配置 PLL1 配置寄存器（RCC_PLL1CFGR）中的 PLL1XTPRE 位，选择 HSE 或者 HSE2 分频为 PLL1 输入时钟；
- 配置 PLL1 配置寄存器（RCC_PLL1CFGR）中 PLL1MUL（倍频系数）、PLL1PDIV（PLL1_IN 分频系数）和 PLL1DIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLL1ON 位为 1，使能 PLL1；
- 等待时钟控制寄存器（RCC_CR）中的 PLL1RDY 位被置位为 1，表示 PLL1 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；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位为 2'b10 来选择 PLL1 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 PLL1；
- 配置控制状态寄存器（RCC_CSR）的 LSION 位为 1 和 LSIOE 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；
- 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。

7.2.4.11 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 7-2 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
011	Reserved
100	SYSCLK
101	HSI
110	HSE
111	PLL1/2

7.2.4.12 RTC 时钟

RTC CLK 时钟源可以是 HSE/128 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。

HSE 和 LSI 时钟不在备份域。因此：

- 如果 LSI 被选为 RTC 时钟：如果 VDD 电源关闭，RTC 状态不确定；
- 如果 HSE 时钟 128 分频后作为 RTC 时钟：如果 VDD 电源关闭或稳压器被关闭，则 RTC 状态不确定。

7.2.4.13 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。

7.2.4.14 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。

7.3 寄存器

7.3.1 寄存器总览

表 7-3 RCC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	时钟控制寄存器	0x00000023
0x04	RCC_CFGR	时钟配置寄存器	0x00000000
0x08	RCC_CIR	时钟中断寄存器	0x00000000
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
0x24	RCC_CSR	控制状态寄存器	0x0C000000
0x28	RCC_AHBRSTR	AHB 外设复位寄存器	0x00000000
0x2C	RCC_CFGR2	时钟配置寄存器 2	0x40030000
0x40	RCC_SYSCFG	系统配置寄存器	0x00000103
0x4C	RCC_ICSCR	内部时钟源校准寄存器	0x02000000
0x50	RCC_PLL1CFGR	PLL1 配置寄存器	0x000E011C
0x54	RCC_PLL2CFGR	PLL2 配置寄存器	0x000E011C

7.3.2 RCC_CR 时钟控制寄存器

偏移地址：0x00

复位值：0x0000 0021

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
Res.		PLL2RD Y	PLL2ON rW	Res.		PLL1RD Y	PLL1ON rW	Res.				CSSON rW	HSEBY P	HSERD Y	HSEON rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Res.	HSIDIV	Res.	HSEOU TPUTS EL	HSEDE GLITCH SEL	HSEDE GLITCH BYP	HSELPPF SEL	HSELPPF BYP	Res.	HSIRDY	HSION
	rw		rw	rw	rw	rw	rw		r	rw

Bit	Field	Description
31: 30	Reserved	保留，必须保持复位值
29	PLL2RDY	PLL2RDY: PLL2 时钟准备就绪标志 (PLL2 clock ready flag) 由硬件设置。 0: PLL2 未稳定 1: PLL2 已稳定 注: 1. PLL2 在上电或者 PLL2_PD(PLL2 使能)启动的时候, 数字的计数逻辑会从 0 开始启动计数 (计数可配), 计数完成后 PLL2 的 clock gating 打开, 产生内部的标志, 放出 PLL2 clock。
28	PLL2ON	PLL2ON: PLL2 使能 (PLL2 enable) 通过软件置 1 或清除。 当进入停止或待机模式时, 由硬件清除。如果 PLL2 时钟用作系统时钟或被选择将成为系统时钟, 则无法重置此位。 0: 禁用 PLL2 1: 使能 PLL2
27: 26	Reserved	保留，必须保持复位值
25	PLL1RDY	PLL1 时钟稳定标志 (PLL1 Clock Ready Flag) 由硬件设置 0: PLL1 未稳定 1: PLL1 已稳定
24	PLL1ON	PLL1 使能 (PLL1 Enable) 通过软件置“1”或清“0”。 当进入停机或待机模式时, 由硬件清除。 如果 PLL1 时钟已经或将要作为系统时钟, 则禁止重置此位。 0: 禁止 PLL1 1: 使能 PLL1
23: 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	HSEBYP	外部高速时钟旁路 (External High-speed Clock Bypass) 由软件置“1”来旁路外部晶体振荡器。 只有当 HSE 振荡器被禁止时, 才能写入 HSEBYP 位。 0: 禁止外部高速晶体振荡器旁路模式 1: 使能外部高速晶体振荡器旁路模式

Bit	Field	Description
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: 14	Reserved	保留, 必须保持复位值
13: 11	HSIDIV	HSIDIV: 内部高速时钟分频系数 (Internal high-speed clock division factor) 000: HSI 不分频 001: HSI 2 分频 010: HSI 4 分频 011: HSI 8 分频 100: HSI 16 分频 101: HSI 32 分频 110: HSI 64 分频 111: HSI 128 分频
10: 9	Reserved	保留, 必须保持复位值
8	HSEOUTPUTSEL	HSEOUTPUTSEL: 输出模式选择 0: 直接输出 1: 滤波输出
7	HSEDEGLITCHSEL	HSEDEGLITCHSEL: 去毛刺宽度控制位 0: 典型值 8ns 1: 典型值 5nS
6	HSEDEGLITCHBYP	HSEDEGLITCHBYP: 去毛刺电路的 bypass 控制位 0: 正常采用去毛刺功能, 1: bypass 掉去毛刺功能
5	HSELPFSEL	HSELPFSEL: LPF 滤波模式控制 0: LPF 滤波后信号放大输出 1: LPF 滤波后输出
4	HSELPFBYP	HSELPFBYP: LPF 输入信号直连输出的使能信号 0: 经过 LPF 滤波输出 1: LPF_IN 直连输出
3: 2	Reserved	保留, 必须保持复位值
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

Bit	Field	Description
0	HSION	<p>内部高速时钟使能（Internal High-speed Clock Enable） 通过软件置“1”或清“0”。</p> <p>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。</p> <p>0：禁止内部高速时钟 1：使能内部高速时钟</p>

7.3.3 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			USBPRE		Res.		USBCL	Res.		
					rw			rw				KSEL			
												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	<p>微控制器时钟输出（Micro Controller Clock Output） 由软件置“1”或清“0” 00x：没有时钟输出 010：LSI 时钟输出 011：Reserved 100：SYSCLK 系统时钟输出 101：HSI 时钟输出 110：HSE 时钟输出 111：PLL1 时钟 2 分频后输出</p> <p>注意： 该时钟输出在启动和切换 MCO 时钟源时可能会被停止。 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。</p>

Bit	Field	Description
23: 22	USBPRE	<p>USBPRE: USB 预分频 (USB prescaler)</p> <p>通过软件指定分频值以输出 48MHz 的 USB 时钟。</p> <p>在使能 RCC_APB1ENR 寄存器中 USB 时钟之前, 此位必须有效</p> <p>00: PLL1 或 PLL2 时钟直接作为 USB 时钟</p> <p>01: PLL1 或 PLL2 时钟 2 分频作为 USB 时钟</p> <p>10: PLL1 或 PLL2 时钟 3 分频作为 USB 时钟</p> <p>11: PLL1 或 PLL2 时钟 4 分频作为 USB 时钟</p>
21: 20	Reserved	保留, 必须保持复位值
19	USBCLKSEL	<p>USBCLKSEL: 选择 USBCLK 时钟源</p> <p>0: 选择系统 PLL1</p> <p>1: 选择 PLL2</p>
18: 14	Reserved	保留, 必须保持复位值。
13: 11	PPRE2	<p>PPRE2: APB2 预分频系数</p> <p>通过软件设置来控制 APB2 时钟 (PCLK2) 预分频系数。</p> <p>0xx: HCLK 不分频</p> <p>100: HCLK 2 分频</p> <p>101: HCLK 4 分频</p> <p>110: HCLK 8 分频</p> <p>111: HCLK 16 分频</p>
10: 8	PPRE1	<p>PPRE1: APB1 预分频系数</p> <p>通过软件设置来控制 APB1 时钟 (PCLK1) 预分频系数。</p> <p>0xx: HCLK 不分频</p> <p>100: HCLK 2 分频</p> <p>101: HCLK 4 分频</p> <p>110: HCLK 8 分频</p> <p>111: HCLK 16 分频</p>
7: 4	HPRE	<p>AHB 预分频系数</p> <p>通过软件设置来控制 AHB 时钟的预分频系数。</p> <p>0xxx: SYSCLK 不分频</p> <p>1000: SYSCLK 2 分频</p> <p>1001: SYSCLK 4 分频</p> <p>1010: SYSCLK 8 分频</p> <p>1011: SYSCLK 16 分频</p> <p>1100: SYSCLK 64 分频</p> <p>1101: SYSCLK 128 分频</p> <p>1110: SYSCLK 256 分频</p> <p>1111: SYSCLK 512 分频</p> <p>注:</p> <p>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。</p>

Bit	Field	Description
3: 2	SWS	系统时钟选择状态 (System Clock Switch Status) 00: 选择 HSI 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 PLL1 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 当从停止或待机模式中返回时, 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 HSI 作为系统时钟。 00: 选择 HSI 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 PLL1 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟

7.3.4 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.		PLL1RD YC	HSERD YC	HSIRDY C	Res.	LSIRDY C
								w1c			w1c	w1c	w1c		w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			PLL1RD YIE	Res.	HSIRDY IE	Res.	LSIRDYI E	CSSF	Res.		PLL1RD YF	HSERD YF	HSIRDY F	Res.	LSIRDY F
			rw		rw		rw	r			r	r	r		r

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: 21	Reserved	保留, 必须保持复位值
20	PLL1RDYC	清除 PLL1 稳定中断 (PLL 1Ready Interrupt Clear) 通过软件置“1”来清除 PLL1 稳定中断标志位 PLL1RDYF。 0: 无效 1: 清除 PLL1 稳定中断标志位 PLL1RDYF

Bit	Field	Description
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	Reserved	保留, 必须保持复位值
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: 13	Reserved	保留, 必须保持复位值
12	PLL1RDYIE	PLL1 稳定中断使能 (PLL 1Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止 PLL1 稳定中断。 0: 禁止 PLL1 稳定中断 1: 使能 PLL1 稳定中断
11	Reserved	保留, 必须保持复位值
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	Reserved	保留, 必须保持复位值
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
6:5	Reserved	保留, 必须保持复位值
4	PLL1RDYF	PLL1 稳定中断标志 (PLL1 Ready Interrupt Flag) 在 PLL1 稳定时, 由硬件置“1”。 通过软件将 PLL1RDYC 位置“1”来清除。 0: 无 PLL1 锁定产生的时钟稳定中断 1: PLL1 锁定导致时钟稳定中断

Bit	Field	Description
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
1	Reserved	保留, 必须保持复位值
0	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

7.3.5 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
LPUART	LPTIM	Res.						DBG	Res.				TIM17	TIM16	TIM14
rw	rw							rw					rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT	UART1	Res.	SPI1	TIM1	Res.	ADC	Res.						SYSCF		
rw	rw		rw	rw		rw							G		
															rw

Bit	Field	Description
31	LPUART	LPUART 复位 (LPUART reset) 由软件置 1 或清‘0’。 0: 无效 1: 重置 LPUART
30	LPTIM	LPTIM 复位 (LPTIM reset) 由软件置 1 或清‘0’。 0: 无效 1: 重置 LPTIM
29: 23	Reserved	保留, 必须保持复位值

Bit	Field	Description
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
14	UART1	UART1 复位 (UART1 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

Bit	Field	Description
0	SYSCFG	SYSCFG 复位 (SYSCFG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

7.3.6 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	CRS	Res.	FLEXCAN	BKP	Res.			I2C1	Res.	UART4	UART3	UART2	Res.
rw			rw	rw		rw	rw				rw		rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	SPI2	Res.		WWDG	Res.				I3C	Res.				TIM3	TIM2	
	rw			rw					rw					rw	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
27	CRS	CRS 复位(CRS reset) 由软件置 1 或清'0'。 0: 无效 1: 复位
26	Reserved	保留, 必须保持复位值
25	FLEXCAN	FLEXCAN: FLEXCAN 复位 (FLEXCAN reset) 通过软件置“1”或清除 0: 无效 1: 复位

Bit	Field	Description
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	Reserved	保留, 必须保持复位值
19	UART4	UART4 复位(UART4 reset) 由软件置 1 或清'0'。 0: 无效 1: 复位
18	UART3	UART3 复位 (UART3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
17	UART2	UART2 复位 (UART2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
16: 15	Reserved	保留, 必须保持复位值
14	SPI2	SPI2 复位 (SPI2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
13: 12	Reserved	保留, 必须保持复位值。
11	WWDG	WWDG 复位 (Window Watchdog Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
10: 7	Reserved	保留, 必须保持复位值
6	I3C	I3C 复位(I3C reset) 由软件置 1 或清'0'。 0: 无效 1: 复位
5: 2	Reserved	保留, 必须保持复位值

Bit	Field	Description
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	TIM2	TIM2 定时器复位 (TIM2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

7.3.7 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.	USB	Res.			GPIOD	GPIOC	GPIOB	GPIOA	Res.
					rw		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
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	Reserved	保留, 必须保持复位值
24	USB	USB 时钟使能(USB clock enable) 由软件置 1 或清'0'。 0: 时钟关闭 1: 时钟开启
23: 21	Reserved	保留, 必须保持复位值
20	GPIOD	GPIOD 时钟使能 (GPIOD Clock Enable) 由软件置“1”或清“0” 0: 时钟关闭 1: 时钟开启

Bit	Field	Description
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

7.3.8 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
LPUART	LPTIM	EXTI	Res.					DBG	Res.			TIM17	TIM16	TIM14	

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

Bit	Field	Description
31	LPUART	LPUART 时钟使能(LPUART clock enable) 由软件置 1 或清'0'。 0: 时钟禁止 1: 时钟使能
30	LPTIM	LPTIM 时钟使能(LPTIM clock enable) 由软件置 1 或清'0'。 0: 时钟禁止 1: 时钟使能
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	COMP	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

Bit	Field	Description
14	UART1	UART1 时钟使能 (UART1 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
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

7.3.9 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	CRS	Res.	FLEXCA N	BKP	Res.			I2C1	Res.	UART4	UART3	UART2	Res.
rw	rw		rw	rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	SPI2	Res.		WWDG	Res.				I3C	Res.				TIM3	TIM2	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	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	CRS	CRS 时钟使能(CRS clock enable) 由软件置 1 或清'0'。 0: 时钟禁止 1: 时钟使能
26	Reserved	保留, 必须保持复位值
25	FLEXCAN	FLEXCAN 时钟使能 (FLEXCAN Clock Enable) 通过软件置"1"或清除 0: 时钟禁止 1: 时钟使能
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
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
19	UART4	UART4 时钟使能(UART4 clock enable) 由软件置 1 或清'0'。 0: 时钟禁止 1: 时钟使能
18	UART3	UART3 时钟使能 (UART3 Clock Enable) 由软件置"1"或清"0" 0: 时钟禁止 1: 时钟使能

Bit	Field	Description
17	UART2	UART2 时钟使能 (UART2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
16: 15	Reserved	保留, 必须保持复位值
14	SPI2	SPI2 时钟使能 (SPI2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
13: 12	Reserved	保留, 必须保持复位值。
11	WWDG	WWDG 时钟使能 (Window Watchdog Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
10: 7	Reserved	保留, 必须保持复位值
6	I3C	I3C 时钟使能(I3C clock enable) 由软件置 1 或清‘0’。 0: 时钟禁止 1: 时钟使能
5: 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	TIM2	TIM2 定时器时钟使能 (TIM2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

7.3.10 RCC_BDCR 备份域控制寄存器

偏移地址: 0x20

复位值: 0x00000000

访问: 只能由备份域复位有效复位支持字, 半字和字节方式访问, 0 到 3 个等待周期, 对该寄存器进行连续访问时需要插入等待状态

注: 备份域控制寄存器中 (RCC_BDCR) 的 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.									
rw							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
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 11: HSE 振荡器在 128 分频后用作 RTC 时钟
7: 0	Reserved	保留，必须保持复位值

7.3.11 RCC_CSR 控制状态寄存器

偏移地址: 0x24

复位值: 0x0C00 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	WWDG	IWDGR	SFTRST	PORRS	PINRST	Res.	RMVF	LOCKU	PVDRS	Res.					
	RSTF	STF	F	TF	F			PF	TF						
r	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	LSIOE	Res.			LSIRDY	LSION

Bit	Field	Description
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	LSIOE	LSI 低压输出使能 (LSI Output Enable) 通过软件置“1”或清“0”，或由电源复位清除。 0: LSI 输出禁止 1: LSI 输出使能 注： 使用 LSI 时需要同时置位 LSION 和 LSIOE,不使用时需要同时关闭 LSION 和 LSIOE
4: 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
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

7.3.12 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.	USB	Res.				GPIOD	GPIOC	GPIOB	GPIOA	Res.
					rw		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	DIVIDER	除法器复位 (DIVIDER 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	USB	USB 复位(USB reset) 由软件置 1 或清‘0’。 0: 无效 1: 复位
23:21	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
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

7.3.13 RCC_CFGR2 时钟配置寄存器 2

偏移地址: 0x2C

复位值：0x40003 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.	LPTIM_CLKSEL		Res.					MCO_PRE				Res.			
	rw							rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CANPRE		Res.			CAN_CLKSEL		Res.					LPUART_CLKSE		
	rw					rw							L		
													rw		

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30: 29	LPTIM_CLKSEL	LPTIM 时钟源选择 00: lse_clk(默认) 01: lsi_clk 10: pclk_lptimer 11: 无时钟
28:24	Reserved	保留，必须保持复位值
23: 20	MCOPRE	MCOPRE: MCO 时钟预分频系数 0xxx: MCO 不分频 1000: MCO 2 分频 1001: MCO 4 分频 1010: MCO 8 分频 1011: MCO 16 分频 1100: MCO 64 分频 1101: MCO 128 分频 1110: MCO 256 分频 1111: MCO 512 分频
19: 14	Reserved	保留，必须保持复位值
13: 12	CANPRE	CANPE 时钟分频 00: 不分频(默认) 01: 2 分频 10: 3 分频 11: 4 分频
11: 10	Reserved	保留，必须保持复位值
9: 8	CAN_CLKSEL	CANPE 时钟源选择 00: PLL1(默认) 01: PLL2 10: HSE 11: 无时钟

Bit	Field	Description
7: 2	Reserved	保留, 必须保持复位值
1: 0	LPUART_CLKSEL	LPUART 时钟源选择 (LPUART clock source selection) 00: Reserved 01: LSI 振荡器作为 LPUART 时钟 10: pclk_lpuart 作为 LPUART 时钟 11: 无时钟

7.3.14 RCC_SYSCFG 系统配置寄存器

偏移地址: 0x40

复位值: 0x0000 0103

访问: 0-3 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HSELPF EN	Res.				HSE_RFB_SEL		Res.						PROG_ CHECK _EN	
	rw					rw								r	

Bit	Field	Description
31: 15	Reserved	保留, 必须保持复位值
14	HSELPFEN	外接晶振低通滤波使能 0: 禁止 1: 使能
13: 10	Reserved	保留, 必须保持复位值
9: 8	HSE_RFB_SEL	HSE_RFB_SEL: 反馈电阻选择: 00: 2M 01: 1M (默认适用 8M~12M 晶振) 10: 500k 11: 200K
7: 1	Reserved	保留, 必须保持复位值
0	PROG_CHECK_EN	写 Flash 时是否检查 Flash 内的数据是否是 0xFF 1: 检查 (硬件固定为 1) 0: 不检查

7.3.15 RCC_ICSCR 内部时钟校准寄存器

偏移地址: 0x4C

复位值：0x0200 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															TRIM_C
															RS_SEL
															rw

Bit	Field	Description
31:1	Reserved	保留，必须保持复位值
0	TRIM_CRS_SEL	TRIM_CRS_SEL: HSITRIM 值是否使用 CRS 模块作为来源 0: 不使用 1: 使用

7.3.16 RCC_PLL1CFGR PLL1 配置寄存器

偏移地址：0x50

复位值：0x000E 011C

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
PLL1PDIV				Res.				PLL1MUL							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PLL1DIV			Res.		PLL1_LDS		PLL1_ICTRL		PLL1XT PRE	PLL1SR C
					rw					rw		rw		rw	rw

Bit	Field	Description
31: 29	PLL1PDIV	PLL1 预分频系数 (PLL1 Pre-divider Factor)
28: 24	Reserved	保留，必须保持复位值
23: 16	PLL1MUL	PLL1 倍频系数 (PLL1 Multiplication Factor)
15: 11	Reserved	保留，必须保持复位值
10: 8	PLL1DIV	PLL1 分频系数 (PLL1 Divide Factor)
7: 6	Reserved	保留，必须保持复位值
5: 4	PLL1_LDS	PLL1 锁定检测器精度选择 (PLL1 Lock Detector Accuracy Select)

Bit	Field	Description
3: 2	PLL1_ICTRL	PLL1 Charge Pump 电流控制信号 (PLL1 CP Current Control Signals) 默认值为 2'b11。当 PLL1 输入时钟源低于 8MHz 时，推荐设置此位为 2'b01。
1	PLL1XTPRE	HSE 时钟用作 PLL1 输入时的分频选择 通过软件置“1”或清除来分频 HSE 时钟后用作 PLL1 输入时钟。 只有当 PLL1 被禁止时，才能写入此位。 0: HSE 时钟不分频 1: HSE 时钟 2 分频
0	PLL1SRC	PLL1 输入时钟源选择 (PLL1 Entry Clock Source Selection) 通过软件置“1”或清除来选择 PLL1 输入时钟源。 只有当 PLL1 被禁止时，才能写入此位。 0: HSI 时钟用作 PLL1 输入时钟 1: HSE 时钟用作 PLL1 输入时钟

注：配置 RCC_PLL1CFGR 寄存器请参考功能描述的 PLL1 章节。

7.3.17 RCC_PLL2CFGR PLL2 配置寄存器

偏移地址：0x54

复位值：0x000E 011C

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
PLL2PDIV			Res.					PLL2MUL							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PLL2DIV			Res.		PLL2_LDS		PLL2_ICTRL		PLL2XT PRE	PLL2SR C
					rw					rw		rw		rw	rw

Bit	Field	Description
31: 29	PLL2PDIV	PLL2 预分频系数 (PLL2 Pre-divider Factor)
28: 24	Reserved	保留，必须保持复位值
23: 16	PLL2MUL	PLL2 倍频系数 (PLL2 Multiplication Factor)
15: 11	Reserved	保留，必须保持复位值
10: 8	PLL2DIV	PLL2 分频系数 (PLL2 Divide Factor)
7: 6	Reserved	保留，必须保持复位值
5: 4	PLL2_LDS	PLL2 锁定检测器精度选择 (PLL2 Lock Detector Accuracy Select)
3: 2	PLL2_ICTRL	PLL2 Charge Pump 电流控制信号 (PLL2 CP Current Control Signals) 默认值为 2'b11。当 PLL2 输入时钟源低于 8MHz 时，推荐设置此位为 2'b01。

Bit	Field	Description
1	PLL2XTPRE	HSE 时钟用作 PLL2 输入时的分频选择 通过软件置“1”或清除来分频 HSE 时钟后用作 PLL2 输入时钟。 只有当 PLL2 被禁止时，才能写入此位。 0: HSE 时钟不分频 1: HSE 时钟 2 分频
0	PLL2SRC	PLL2 输入时钟源选择 (PLL2 Entry Clock Source Selection) 通过软件置“1”或清除来选择 PLL2 输入时钟源。 只有当 PLL2 被禁止时，才能写入此位。 0: HSI 时钟用作 PLL2 输入时钟 1: HSE 时钟用作 PLL2 输入时钟

注：配置 RCC_PLL2CFGR 寄存器请参考功能描述的 PLL2 章节。

Preliminary

8 RTC 实时时钟器

8.1 简介

RTC 模块内部包含一组连续计数的计数器。它作为一个独立的定时器，在相应软件配置下，可提供时钟功能。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或从待机模式唤醒后，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。

8.2 主要特征

- 0 到 220 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 2 种：
 - ◆ HSE 时钟的 128 分频；
 - ◆ 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；

8.3 功能描述

8.3.1 功能框图

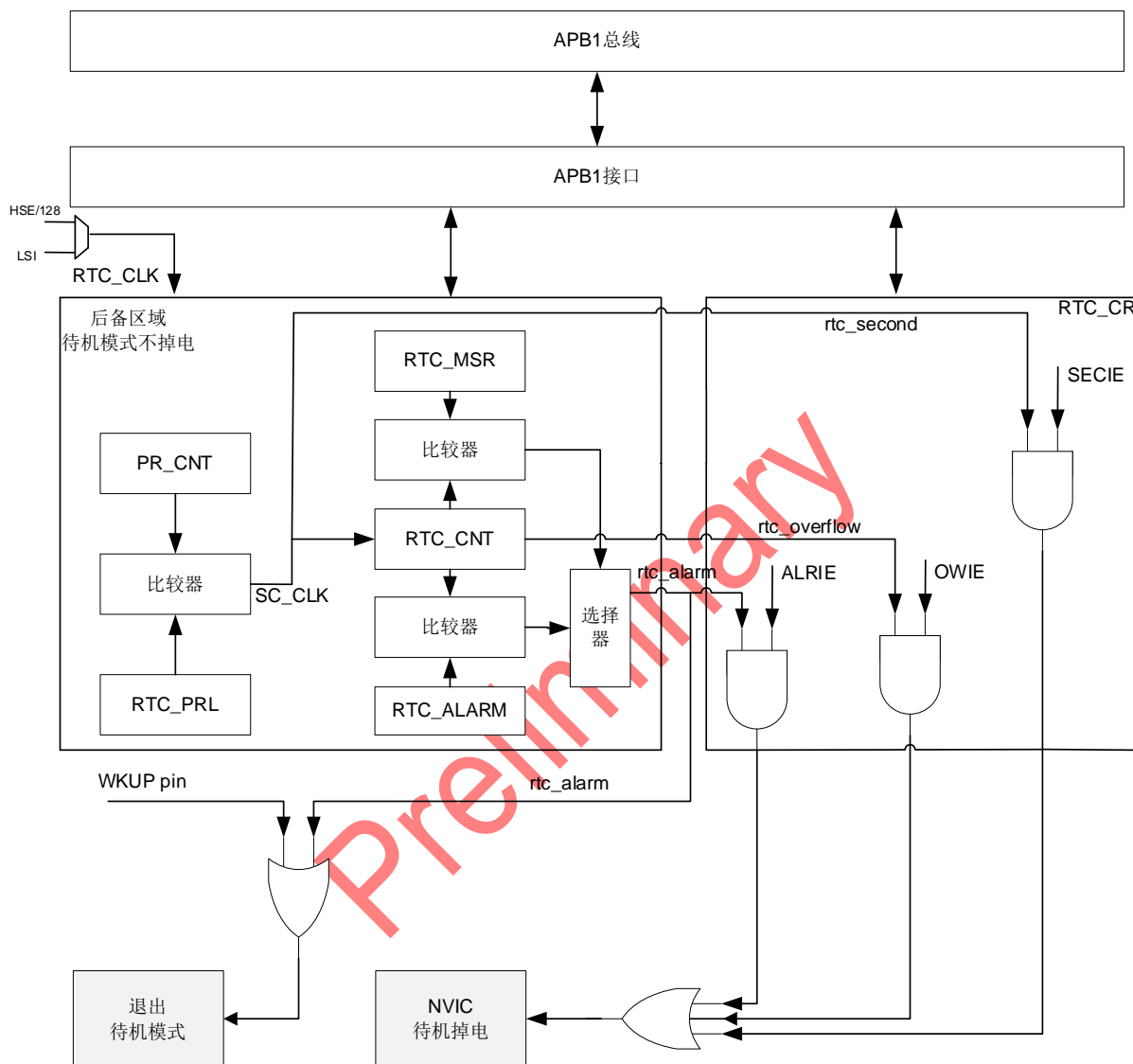


图 8-1 RTC 功能框图

8.3.2 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。

8.3.3 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。

8.3.4 寄存器读取

RTC 内部分频、计数与闹钟部分通过 RTCAPB1 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：

- 从待机模式唤醒系统并立即进行读操作；
- 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、无时钟或断电，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。

8.3.5 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；

8.3.6 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。

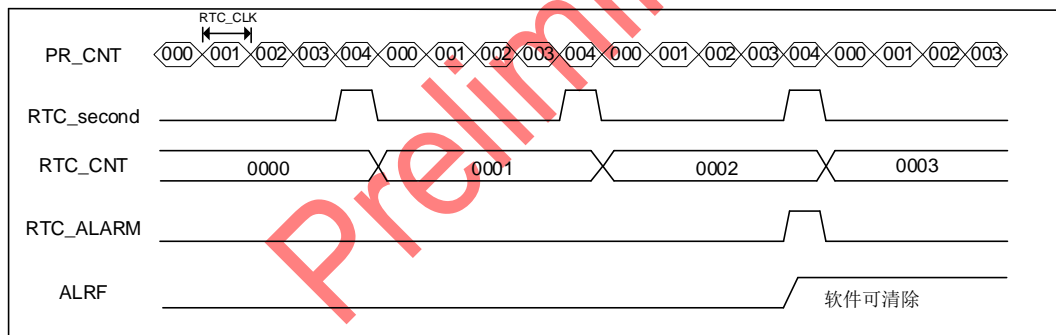


图 8-2 RTC 秒和闹钟波形图示例，RTC_PRL = 0004，RTC_ALR = 002

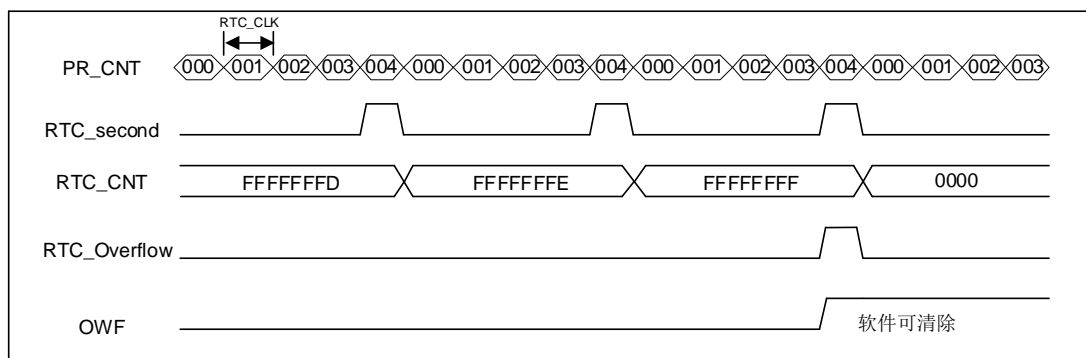


图 8-3 RTC 溢出波形图示例，RTC_PRL = 0004

8.3.7 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。

8.3.8 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。

8.4 寄存器描述

8.4.1 寄存器总览

表 8-1 RTC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RTC_CRH	RTC 控制寄存器高位	0x00000000
0x04	RTC_CRL	RTC 控制寄存器低位	0x00000020
0x08	RTC_PRLH	RTC 预分频装载寄存器高位	0x00000000
0x0C	RTC_PRL	RTC 预分频装载寄存器低位	0x00008000
0x10	RTC_DIVH	RTC 预分频器分频因子寄存器高位	0x00000000
0x14	RTC_DIVL	RTC 预分频器分频因子寄存器低位	0x00000000
0x18	RTC_CNTH	RTC 计数器寄存器高位	0x00000000
0x1C	RTC_CNTL	RTC 计数器寄存器低位	0x00000000
0x20	RTC_ALRH	RTC 闹钟寄存器高位	0x0000FFFF
0x24	RTC_ALRL	RTC 闹钟寄存器低位	0x0000FFFF
0x28	RTC_MSRH	RTC 毫秒寄存器高位	0x00000000
0x2C	RTC_MSRL	RTC 毫秒寄存器低位	0x00000000

8.4.2 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
Res.													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

8.4.3 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
Res.									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

Bit	Field	Description
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
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；

8.4.4 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
Res.												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。

8.4.5 预分频装载寄存器低位 (RTC_PRL)

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 预分频器重载值低位 (RTC Prescaler Reload Value Low)

8.4.6 预分频器分频因子寄存器高位 (RTC_DIVH)

偏移地址: 0x10

复位值: 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												DIV[19: 16]			
r															

Bit	Field	Description
15: 4	Reserved	保留, 必须保持复位值

Bit	Field	Description
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。

8.4.7 预分频器分频因子寄存器低位 (RTC_DIVL)

偏移地址: 0x14

复位值: 0x0000 0000

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。

8.4.8 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]	RTC 计数器高位 (RTC Counter High) 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。 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。

8.4.9 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。</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>

8.4.10 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>

8.4.11 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]	<p>RTC 闹钟值低位 (RTC Alarm Low)</p> <p>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。</p>

8.4.12 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
Res.												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。

8.4.13 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。

9 CRC 循环冗余校验计算单元

9.1 简介

CRC 计算单元利用固定的多项式来计算 8 位、16 位或者 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。

9.2 主要特征

- 统一计算多项式 0x4C11DB7:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

该多项式与以太网计算多项式相同

- 支持 8、16、32 位宽的数据输入寄存器、32 位宽的数据输出寄存器
- 硬件计算时间为 3 个 HCLK 周期
- 带有可存放中间计算过程的 32 位宽的数据寄存器
- 支持 CRC-32 和 CRC-32/MPEG-2 两种算法
- 支持输入数据和输出数据的大小端选择

9.3 功能描述

9.3.1 功能框图

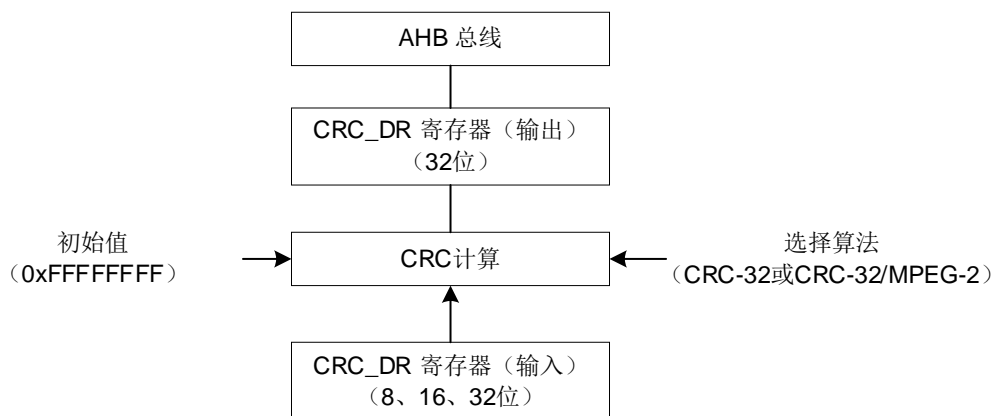


图 9-1 CRC 功能框图

9.3.2 功能概述

- CRC 运算时，对 CRC 数据寄存器（CRC_DR）进行写操作后，新写入的数据会与上一次的 CRC 计算结果进行 CRC 运算，得到一个新的 CRC 结果，这个 CRC 结果会存放于 CRC 数据寄存器（CRC_DR）中。
- 进行 CRC 运算时，可以连续的将需要计算的数据依次写入到 CRC 数据寄存器（CRC_DR）中，最后通过读取 CRC 数据寄存器（CRC_DR）获取 CRC 的运算结果。
- 通过置位 CRC 控制寄存器（CRC_CR）中的 RST 位，可以将 CRC 数据寄存器（CRC_DR）和 CRC 中间数据寄存器（CRC_MIR）的值恢复到默认值 0xFFFFFFFF。但不会影响 CRC 独立数据寄存器（CRC_IDR）的值。
- 提供 8 位独立的数据寄存器（CRC_IDR），作为缓存寄存器，任何时候都可以对其进行读写操作。
- 可以通过 CRC 控制寄存器（CRC_CR）的 AS 位来选择不同的 CRC 计算算法。
- 可以通过 CRC 控制寄存器（CRC_CR）的 ISIZE 位来选择输入数据的位宽；当输入数据为 8 位宽时，CRC 数据寄存器（CRC_DR）的取值范围为[0x00, 0xFF]，当输入数据为 16 位宽时，CRC 数据寄存器（CRC_DR）的取值范围为[0x0000, 0xFFFF]，当输入数据为 32 位宽时，CRC 数据寄存器（CRC_DR）的取值范围为[0x00000000, 0xFFFFFFFF]。
- 可以通过 CRC 控制寄存器（CRC_CR）的 IES 位来设置输入数据的大小端，OES 位来设置输出数据的大小端。
- 可以通过操作 CRC 中间数据寄存器（CRC_MIR），实现多串数据之间的交错 CRC 计算。

9.3.3 使用方法

9.3.3.1 CRC 计算操作步骤

- 使能 CRC 模块时钟；
- 复位 CRC 模块；
- 配置 CRC_CR 寄存器：
 - ◆ 选择算法：CRC-32 或者 CRC-32/MPEG-2；
 - ◆ 选择输入数据位宽：8 位、16 位或者 32 位；

（举例：数值 0x00000012，输入数据 8 位位宽时，CRC_DR 应赋值为 0x12；16 位位宽时，CRC_DR 应赋值为 0x0012；32 位位宽时，CRC_DR 应赋值为 0x00000012）

- ◆ 选择输入、输出的大小端。

（如 0x12345678，大端为 0x78563412，小端为 0x12345678）

- 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。

9.3.3.2 交错 CRC 计算操作步骤

当在第一串数据未计算完插入第二串数据的情况的步骤：

- 将第一串数据计算的中间结果从 CRC 中间数据寄存器（CRC_MIR）中读出并进行存储；
- 通过置位 CRC 控制寄存器（CRC_CR）的 RST 位来复位 CRC 数据寄存器（CRC_DR）；
- 依次将第二串数据写入到 CRC 数据寄存器（CRC_DR）中，在完成 CRC 运算后，将第二串数据的 CRC 运算结果从 CRC 数据寄存器（CRC_DR）中读出并进行存储；
- 再次通过置位 CRC 控制寄存器（CRC_CR）的 RST 位来复位 CRC 数据寄存器（CRC_DR），并把 CRC 计算模式设置成与第一串数据的模式相同；
- 将存储的第一串数据中间计算结果写入 CRC 中间数据寄存器（CRC_MIR）中，然后将第一串数据中剩余的数据依次写入 CRC 数据寄存器（CRC_DR），硬件将自动继续进行计算第一串未完成的数据，待计算完成，可以从 CRC 数据寄存器（CRC_DR）中读取第一串数据的 CRC 运算结果。

9.4 寄存器

9.4.1 寄存器总览

表 9-1 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
0x0C	CRC_MIR	CRC 中间数据寄存器	0xFFFFFFFF

9.4.2 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

9.4.3 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								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。

9.4.4 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										OES	IES	ISIZE		AS	RST
										rw	rw	rw		rw	w

Bit	Field	Description
31: 6	Reserved	保留, 必须保持复位值
5	OES	输出大小端选择 (Output Endian Selection) 0: 小端 (Little-endian) 1: 大端 (Big-endian)

Bit	Field	Description
4	IES	输入大小端选择 (Input Endian Selection) 0: 小端 (Little-endian) 1: 大端 (Big-endian)
3: 2	ISIZE	输入数据宽度 (Input Size) 00: 输入为 32 位 01: 输入为 16 位 10: 输入为 8 位 11: 保留
1	AS	CRC 算法选择 (CRC Algorithm Selection) 0: 算法为 CRC-32/MPEG-2 1: 算法为 CRC-32
0	RST	复位 CRC 计算单元 (CRC Reset) CRC 数据寄存器 (CRC_DR) 和 CRC 中间数据寄存器 (CRC_MIR) 复位为 0xFFFF FFFF 该位只能写“1”，硬件自动清“0”

9.4.5 CRC_MIR CRC 中间数据寄存器

偏移地址: 0x0C

复位值: 0xFFFF FFFF

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

Bit	Field	Description
31: 0	MIR	中间数据寄存器 (Middle Data Register) 用于读取计算的中间结果, 读写都是 32 位操作 如果将读取的值重新写入, 可以继续之前的 CRC 计算。在多个数据序列做 CRC 计算的时候, 可以将计算结果的读出保存, 在其他 CRC 计算完成后, 将保存值写入到寄存器继续之前被中断的计算。

10 CRS 时钟回馈系统

10.1 简介

作为一种先进的数字控制器，时钟回馈系统（CRS）用于校准内部高速时钟 HSI。

微调后精准的 HSI，通过 PLL 倍频得到精准的 48M 时钟提供给 USB 外设，并可支持多种同步信号来源。

10.2 主要特征

- 多种硬件同步源供选择（包括可配置分频系数的预分频器以及信号极性选择）
 - ◆ 直接来自 GPIO 外部管脚的同步源
 - ◆ 来自 USB 外设的 SOF 包，相邻 SOF 包间隔时长由 USB 主机控制为 1ms
- 同步脉冲（SYNC）可以通过软件配置产生
- 硬件自动校准功能，无需软件每次根据实际差值调整微调值
- 可通过手动进行相关配置使得频率更快调整到期望值
- 带有自动误差捕获及数据加载的 16 位频率误差计数器
- 频率误差分析并且产生状态标志的界限值可编程
- 可屏蔽中断
 - ◆ 期望的 0 误差同步事件（ESYNC）
 - ◆ 同步时误差可接受（SYNCOK）
 - ◆ 同步时误差较大（SYNCWARN）
 - ◆ 同步时误差超出调整范围、同步信号丢失、微调值溢出（ERR）

10.3 功能描述

10.3.1 功能框图

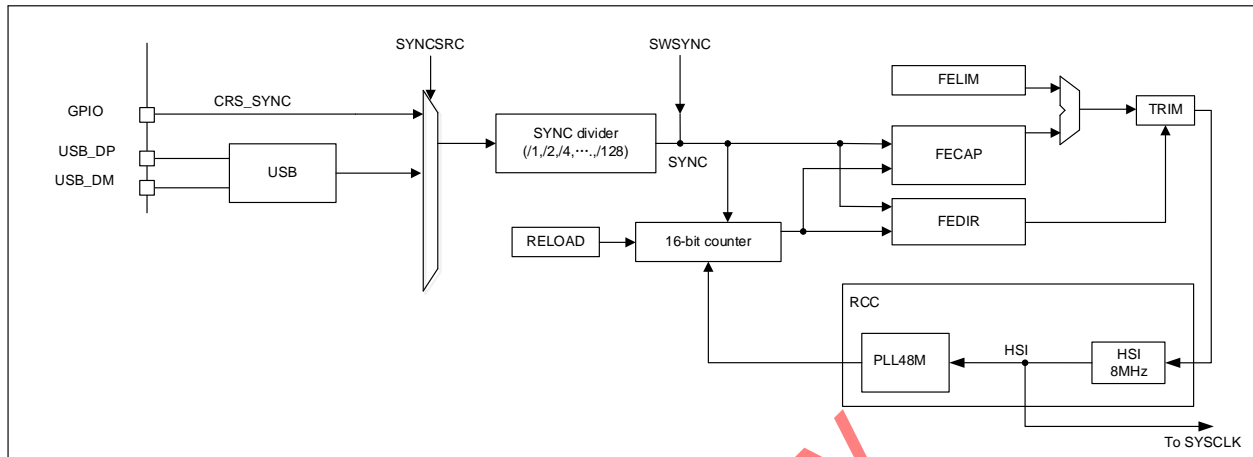


图 10-1 CRS 功能框图

10.3.2 信号描述

信号名	类型	描述
GPIO	输入	来自 PAD 的同步源输入信号
USB_DP/USB_DM	输入	来自 USB 的同步源信号
HSI	输出	校准后的 HSI 时钟输出

10.3.3 功能概述

通过配置 CRS_CFGR 寄存器的 SYNCSRC 位，选择同步源信号；CRS 将实际 48MHz

USB 时钟（来自以 HSI 为参考源的 PLL 输出）与参考的可选同步源进行比较，可以计算出 USB 时钟的实际频率。同时，可根据测得的实际频率与理想频率的差值得到 TRIM 值，用其对 HSI 时钟进行微调，得到精确的 HSI 时钟输出。CRS 可配置自动微调或手动微调。

10.3.4 同步输入

通过 CRS_CFGR 寄存器配置的 CRS 同步源可以来自外部 GPIO 引脚、USB SOF 包。为满足同步输入源有更好的稳健性，CRS 支持一个简单的通过 USB 时钟采样的 2 级数字滤波器，它可以过滤很多小错误。

信号源的极性可配置，且可以通过一个可编程的预分频器分频，以保证其处于一个合适的频率范围内（通常为 1KHz）。

CRS 同步源配置信息详见 CRS 配置寄存器 (CRS_CFGR)。

10.3.5 频率误差测量

频率误差计数器是一个 16 位计数器，可被硬件控制向下或者向上计数。每次同步事件触发时，会使得 RELOAD 值被重新载入该计数器，进行重新计数。在重载后，计数器开始向下重新计数。计数器递减为 0 时，对应 ESYNC 标志位置起。之后计数器自动转换计数方向为向上计数。

在计数器计数的过程中，接收到 SYNC 事件会立即停止计数器计数。此时计数器的值会保存在 CRS_ISR 的 FECAP 位。此时计数器的计数方向会保存在 CRS_ISR 的 FEDIR 位。

根据计数器因 SYNC 事件停止时的计数方向，可以推测出实际时钟频率与期望值的误差：

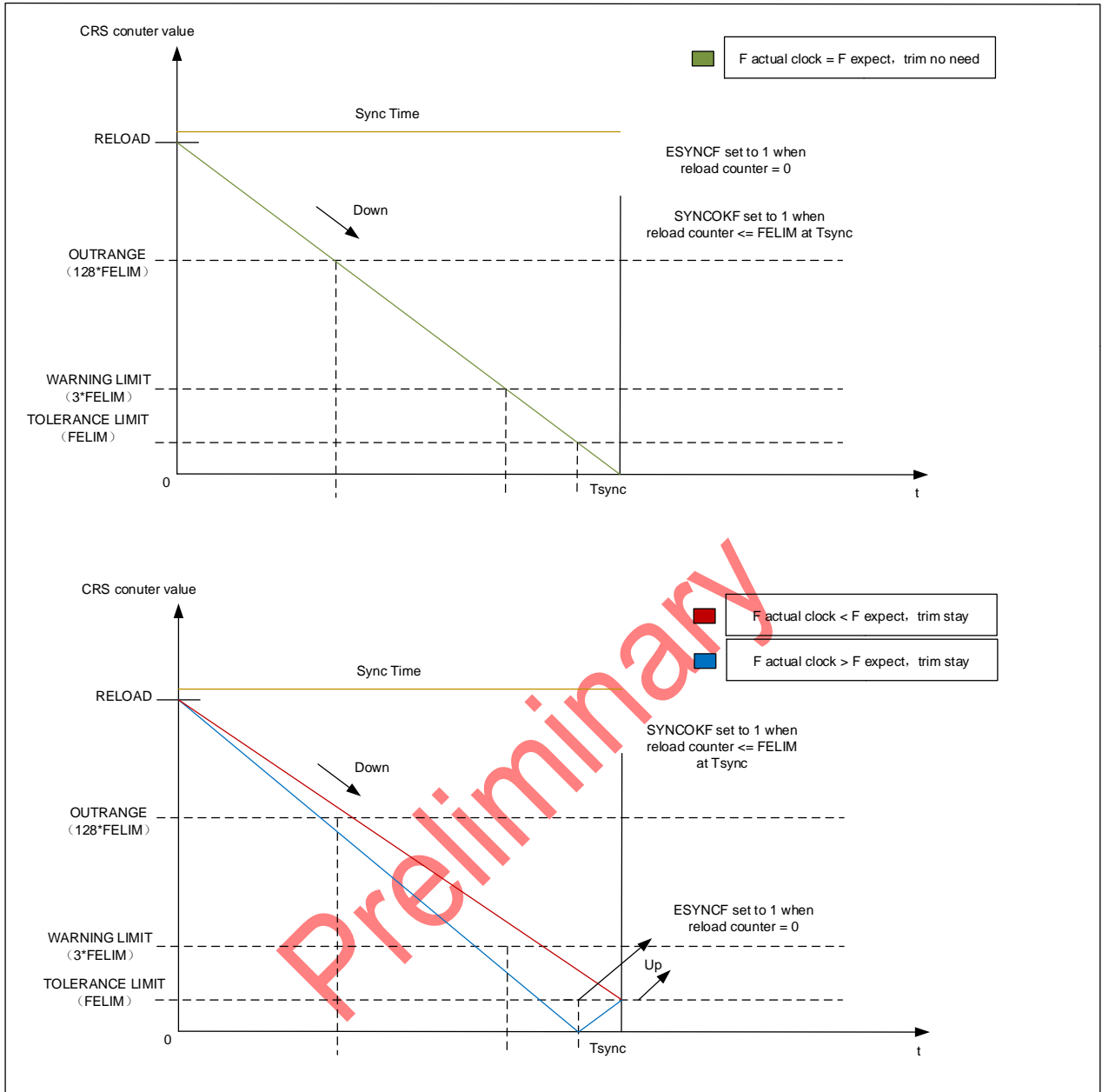
- 停止时 FEDIR=0，即向上计数，实际频率大于期望频率；
- 停止时 FEDIR=1，即向下计数，实际频率小于或等于期望频率。

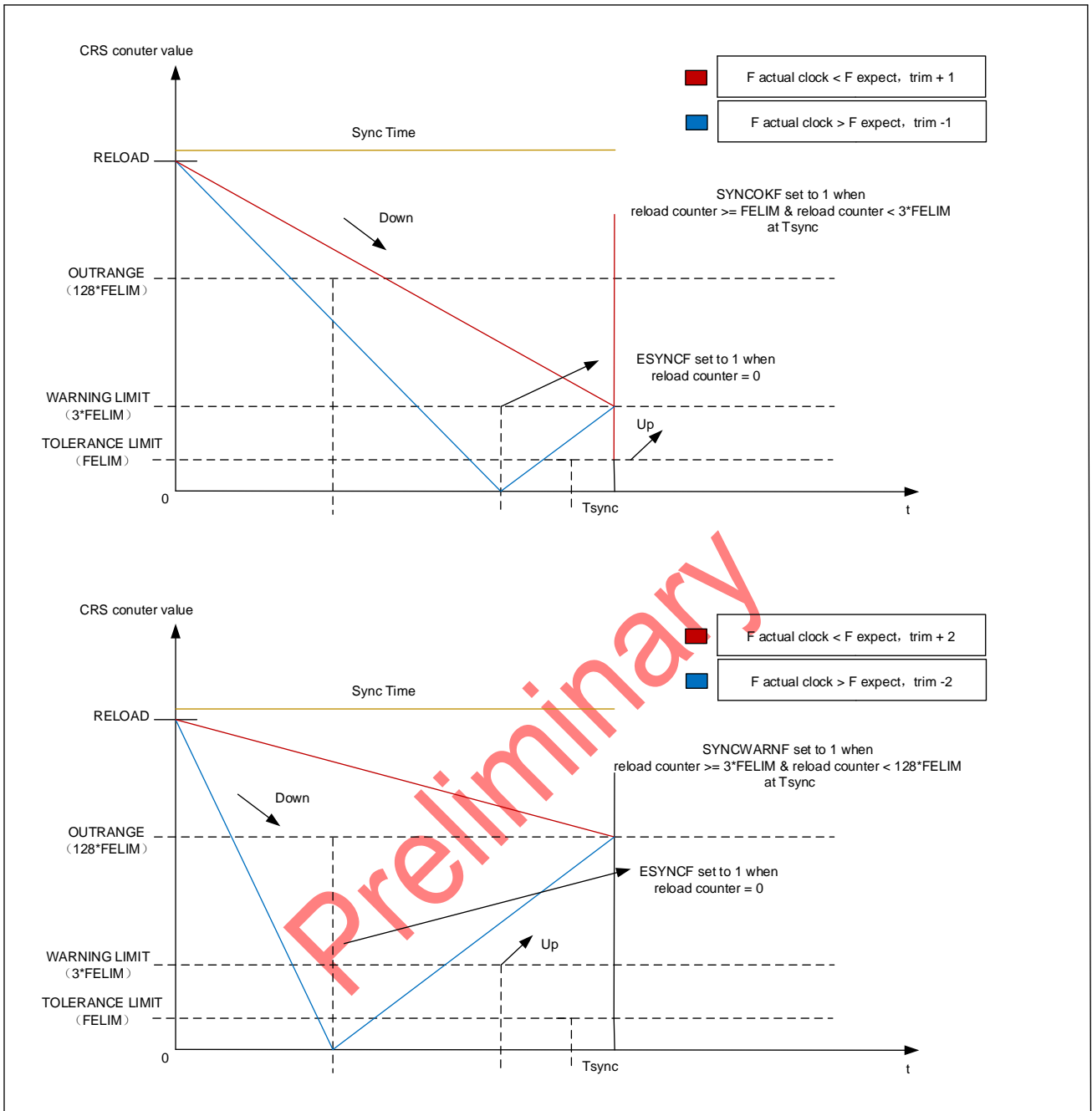
根据计数器因 SYNC 事件停止时的计数值与界限值（界限值可配置，为 CRS_CFGR 寄存器的 FELIM 位）之间的差值，可以得到微调时的步长：

- 停止时 $FECAP < FELIM$ ，可接受范围，不用微调；
- 停止时 $FELIM \leq FECAP < 3 * FELIM$ ，误差稍大，微调步长建议为 1；
- 停止时 $3 * FELIM \leq FECAP < 128 * FELIM$ ，误差较大，微调步长建议为 2；
- 停止时 $128 * FELIM \leq FECAP$ ，误差过大，超出微调范围，微调无意义。

在计数器计数的过程中，一直未接收到 SYNC 事件会一直进行计数，直到计数到设定的最大界限值（最大界限值=128*FELIM），同时标志位 SYNCMISS 置起，表示未接收到 SYNC 事件。

下图列举了计数器测量时可能发生的情况。





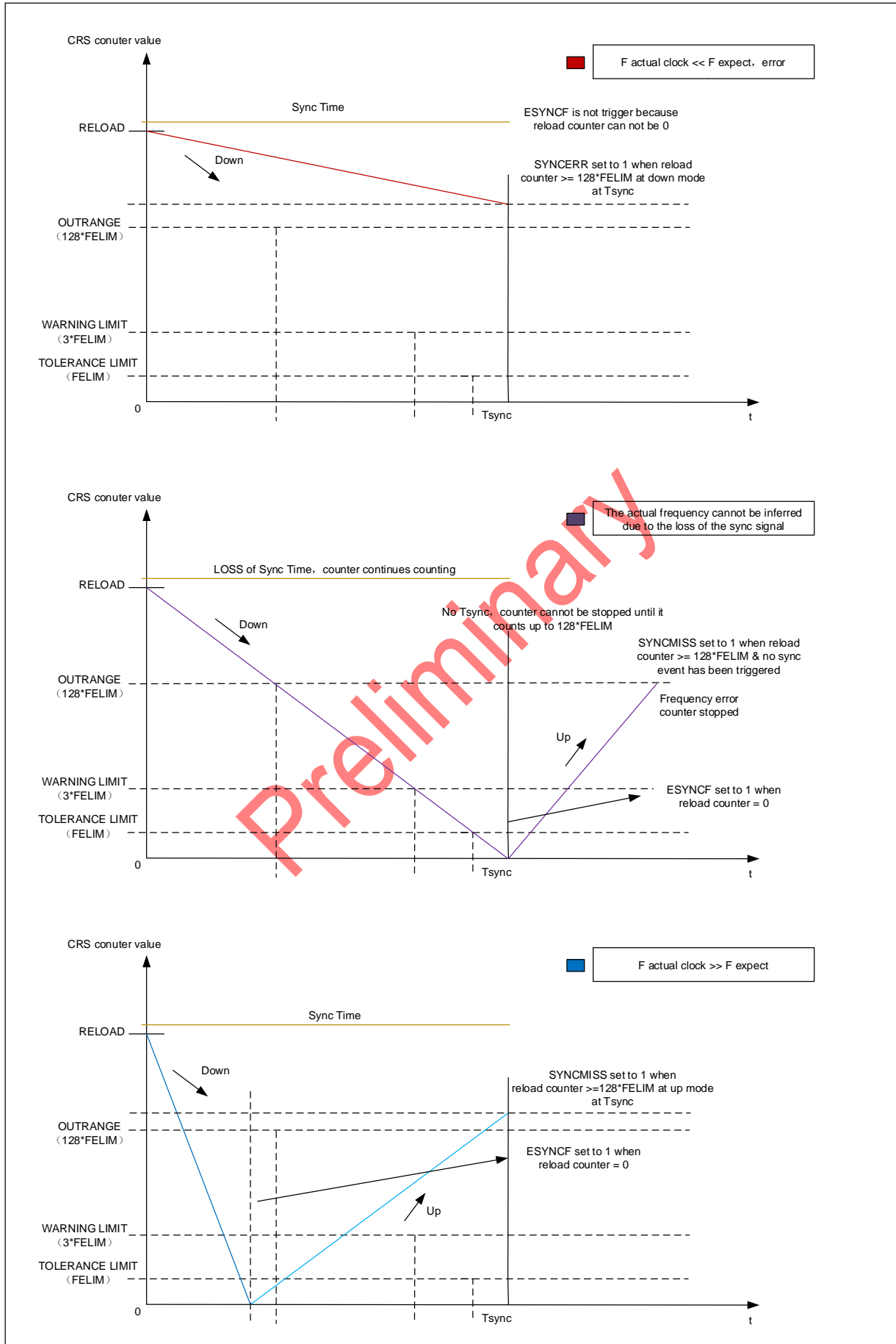


图 10-2 CRS 计数器状态图

10.3.6 频率误差计算及自动校准

通过比较频率误差值与定义的界限区间，可以得到实际频率与期望频率的相对关系：

- 定义 $1 * FELIM$ 为容忍界限 (TOLERANCE LIMIT)
- 定义 $3 * FELIM$ 为警告界限 (WARNING LIMIT)
- 定义 $128 * FELIM$ 为溢出界限 (ERROR LIMIT)

通过判断 FECAP 在哪个区间，对应的状态标志位会置起，同时也会决定自动校准的调整步长（需要使能 CRS_CR 寄存器 AUTOTRIM 位）：

- 当频率误差低于容忍界限，表明实际校准值位于理想范围内，不需要修正。
- 同步正确标志位 (SYNCOK) 被置位
- 自动校准 (AUTOTRIM) 模式下，不需要修正 TRIM 值
- 当频率误差低于警告界限且大于或等于容忍界限，表明需要微小校准，调节器只需调整 1 档就能达到理想的 TRIM 值。
- 同步正确标志位 (SYNCOK) 被置位
- 自动校准 (AUTOTRIM) 模式下 TRIM 数值每次调节一个档位
- 当误差计数大于或等于警告界限，但小于错误界限，表明需要较大校准，并且下个同步周期将不会达到理想的 TRIM 值。
- 同步警告标志位 (SYNCWARN) 被置位
- 自动校准 (AUTOTRIM) 模式下 TRIM 数值每次调节二个档位
- 当误差计数大于或等于错误界限，表明频率误差超出可校准范围，无法校准。当同步信号未清除或丢失时，也将发生上述情况（如：USB SOF 包丢失）。
- 同步错误 (SYNCERR) 或同步丢失 (SYNCMISS) 标志位被置位
- 自动校准 (AUTOTRIM) 模式下 TRIM 数值不变

注：如果实际 TRIM 值已接近边界，继续（自动）校准将导致 TRIM 值溢出，此时 TRIM 值将被锁定在边界，且校准溢出 (TRIMOVF) 标志位被置位。

自动校准 (AUTOTRIM) 模式下，TRIM 位仅由硬件配置且软件只读。

10.3.7 CRS 初始化及配置

10.3.7.1 加载 (RELOAD) 值

RELOAD 值的选定是根据目标频率和分频后的同步源频率比较得出的。RELOAD 每个时钟周期会减少 1，从而希望在零值时刚好发生 SYNC 事件，表明实际频率等于期望频率。公式如下：

$$\text{RELOAD} = (f_{\text{TARGET}} / f_{\text{SYNC}}) - 1$$

重置后的 RELOAD 默认值由目标频率为 48MHz 的时钟和频率为 1KHz（来自 USB 的 SOF 信号）的同步信号计算得到，即 47999（16 进制为 0xBB7F）。

10.3.7.2 FELIM 值

FELIM 的选择需要将 USB 时钟的特性以及它的典型的调节档位相结合。理想的路径数值与调整步长的一半对应，公式如下：

$$\text{FELIM} = (f_{\text{TARGET}} / f_{\text{SYNC}}) * \text{STEP}[\%] / 2$$

为了获得更好的修正反馈，FELIM 应当设置为计算结果最近的整数。如果不希望频繁的调整修正值，可以适当的增加 FELIM 数值。

例如想要让目标频率达到 48MHz，为了匹配 $(f_{\text{TARGET}} / f_{\text{SYNC}}) = 48000$ 和典型的调节档位（STEP 为 0.1%），计算得到 FELIM 值为 24。

注意：错误的配置 RELOAD 和 FELIM 会导致一个不稳定的递减响应，并且没有硬件的保护。理想的运作模式需要适当的 RELOAD 值的设置（根据同步源的频率设置）而且 RELOAD 的值也是大于 FELIM 值的 128 倍。

10.4 CRS 低功耗模式

在系统处于睡眠模式时，对 CRS 的运行没有影响，CRS 的中断服务程序会让设备退出睡眠模式。

当系统处于停止或者待机模式时，CRS 停止操作，寄存器不能被改写，直到停止或待机模式退出，同时 HSI 重启。

10.5 中断

表 10-1 CRS 中断请求

中断事件	事件状态	使能位	清除位
期望的 0 误差同步事件	ESYNCF	ESYNCE	ESYNCC
同步时误差可接受	SYNCOKF	SYNCOKIE	SYNCOKC
同步时误差较大	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同步错误或校准错误 (TRIMOVF, SYNCMISS, SYNCERR)	ERRF	ERRIE	ERRC

10.6 寄存器

10.6.1 寄存器总览

表 10-2 CRS 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CRS_CR	CRS 控制寄存器	0x00020000
0x04	CRS_CFGR	CRS 配置寄存器	0x2022BB7F
0x08	CRS_ISR	CRS 中断状态寄存器	0x00000000
0x0C	CRS_ICR	CRS 中断标志清除寄存器	0x00000000

10.6.2 CRS_CR 控制寄存器

地址偏移: 0x00

复位值: 0x0002_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														TRIM[9: 8]	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIM[7: 0]								SWSYN C	AUTOT RIMEN	CEN	Res.	ESYNCI E	ERRIE	SYNCW ARNIE	SYNCO KIE
rw								rw	rw	rw		rw	rw	rw	rw

Bit	Field	Description
31: 18	Reserved	保留, 必须保持复位值
17: 8	TRIM[9: 0]	<p>TRIM[9: 0]: (HSI Trimming)</p> <p>默认值是 0x200, 这个值是修正档位的中间值。</p> <p>当 RCC_ICSCR 的 TRIM_CRSEL 置 1 时, 写入的值可以重新校正 HSI 频率, 否则写入的值不起作用。</p> <p>当 AUTOTRIMEN 被设置, 此 TRIM 值由硬件控制, 且为只读。</p> <p>全 0: 频率最小</p> <p>全 1: 频率最大</p>
7	SWSYNC	<p>软件同步行为 (Software SYNC Event)</p> <p>由软件设置, 产生一个软件 SYNC 事, 由硬件自动清除。</p> <p>写 0: 无操作</p> <p>写 1: 软件 SYNC 行为被触发 (生成软件 SYNC 事件)</p>

Bit	Field	Description
6	AUTOTRIMEN	自动校准使能 (Automatic Trimming Enable) 根据测量两次同步事件之间的频率误差来计算 TRIM 值。如果 AUTOTRIMEN 位被置位，TRIM 位将被锁定为只读，TRIM 值每次可由硬件调整一个或者两个档位（取决于测量频率误差值）。 0: 禁止自动校准，TRIM 位可被用户修改 1: 使能自动校准，TRIM 位只读且由硬件控制
5	GEN	频率误差计数器使能 (Frequency Error Counter Enable) 0: 频率误差计数器禁止 1: 频率误差计数器使能 当被置位，CRS_CFGR 寄存器被写保护且不能被改写。
4	Reserved	保留，必须保持复位值
3	ESYNCFIE	预期同步中断使能 (Expected SYNC Interrupt Enable) 0: 预期同步 (ESYNCF) 中断禁止 1: 预期同步 (ESYNCF) 中断使能
2	ERRIE	同步及修正错误中断使能 (Synchronization or Trimming Error Interrupt Enable) 0: 同步及修正错误 (ERRF) 中断禁止 1: 同步及修正错误 (ERRF) 中断使能
1	SYNCWARNIE	同步警告中断使能 (SYNC Warning Interrupt Enable) 0: 同步警告 (SYNCWARNF) 中断禁止 1: 同步警告 (SYNCWARNF) 中断使能
0	SYNCOKIE	同步行为 OK 中断使能 (SYNC Event OK Interrupt Enable) 0: 同步行为 OK (SYNCOKF) 中断禁止 1: 同步行为 OK (SYNCOKF) 中断使能

10.6.3 CRS_CFGR 配置寄存器

地址偏移: 0x04

复位值: 0x2022_BB7F

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

Bit	Field	Description
31	SYNCPOL	同步极性选择 (SYNC Polarity Selection) 这位由软件设置与清除，用来选择同步信号源的输入极性。 0: 同步信号上升沿有效 1: 同步信号下降沿有效
30	Reserved	保留，必须保持复位值

Bit	Field	Description
29: 28	SYNCSRC	同步信号源选择 (SYNC Signal Source Selection) 这些位由软件设置与清除, 用来选择信号源。 00: GPIO 选为同步的信号源 01: 保留 10: USB SOF 选为同步的信号源 11: 保留
27	Reserved	保留, 必须保持复位值
26: 24	SYNCDIV	同步信号分频 (SYNC Divider) 这些位由软件设置清除, 用来控制同步信号的分频系数。 000: SYNC 不分频 001: SYNC 2 分频 010: SYNC 4 分频 011: SYNC 8 分频 100: SYNC 16 分频 101: SYNC 32 分频 110: SYNC 64 分频 111: SYNC 128 分频
23: 16	FELIM	频率误差限制 (Frequency Error Limit) FELIM 包含的数值是用来计算捕获的频率误差数值, 频率误差数值存放在 CRS_ISR 寄存器的 FECAP[15: 0]域。
15: 0	RELOAD	计数器加载的数值 (Counter Reload Value) RELOAD 的值在每一次同步行为后都会重新加载进频率误差寄存器中。

10.6.4 CRS_ISR 中断状态寄存器

地址偏移: 0x08

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FECAP																
r																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FEDIR	Res.				TRIMOV	SYNCF	SYNCE	Res.					ESYNC	ERRF	SYNCW	SYNCO
					F	SS	RR						F	ARNF	KF	
r					r	r	r						r	r	r	r

Bit	Field	Description
31: 16	FECAP	频率误差捕获 (Frequency Error Capture) 频率误差计数器的值, 在每次的同步行为结束后被锁存。

Bit	Field	Description
15	FEDIR	<p>频率错误方向 (Frequency Error Direction)</p> <p>频率误差计数器的计数方向, 在每次的同步行为后被锁存。表明了实际频率是高于还是低于目标频率。</p> <p>0: 向上计数, 实际频率高于目标频率</p> <p>1: 向下计数, 实际频率低于目标频率</p>
14: 11	Reserved	保留, 必须保持复位值
10	TRIMOVF	<p>修正溢出 (Trimming Overflow)</p> <p>当自动修正要上溢出或者下溢出的时候, 此标志位被硬件置位。如果 CRS_CR 寄存器的 ERRIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 ERRC 位清除。</p> <p>0: 没有修正溢出错误</p> <p>1: 产生修正溢出错误</p>
9	SYNCMISS	<p>同步信号丢失 (SYNC Missed)</p> <p>当频率误差计数器达到 FELIM 的 128 倍且没有任何 SYNC 事件被触发, 意味着一个 SYNC 脉冲丢失或者是频率误差太大 (内部频率太高) 以至于不能通过调整 TRIM 值来调节, 需要采取其他的措施, SYNCERR 被硬件置位。在这种情况下, 频率误差计数器停止工作 (等待下一个 SYNC)。如果 CRS_CR 寄存器的 ERRIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 ERRC 位清除。</p> <p>0: 没有同步信号丢失错误</p> <p>1: 产生同步信号丢失错误</p>
8	SYNCERR	<p>同步信号错误 (SYNC Error)</p> <p>当同步脉冲输入时, 在触发 ESYNC 行为之前, 测量到的频率误差大于或者等于 FELIM 的 128 倍, SYNCERR 被硬件置位。这说明频率误差太大 (内部频率太低) 以至于不能通过调整 TRIM 值来调节, 需要采取其他的措施。如果 CRS_CR 寄存器的 ERRIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 ERRC 位清除。</p> <p>0: 没有同步信号错误</p> <p>1: 产生同步信号错误</p>
7: 4	Reserved	保留, 必须保持复位值
3	ESYNCF	<p>预期同步标志 (Expected SYNC Flag)</p> <p>当误差频率计数器到达 0 值的时候, 这个标志位被硬件置位。如果 CRS_CR 寄存器的 ESYNCIE 位被置位, 中断产生。通过软件设置 CRS_ICR 寄存器的 ESYNCC 位清除。</p> <p>0: 没有预期同步信号</p> <p>1: 产生预期同步信号</p>
2	ERRF	<p>错误标志 (Error Flag)</p> <p>当产生同步或者修正错误的时候 (TRIMOVF、SYNCMISS 和 SYNCERR), 此标志位被硬件置位。如果 CRS_CR 寄存器的 ERRIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 ERRC 位清除。</p> <p>0: 没有同步或者修正错误信号</p> <p>1: 产生同步或者修正错误信号</p>

Bit	Field	Description
1	SYNCWARNF	同步警告标志 (SYNC Warning Flag) 当测量到的频率误差大于或者等于 FELIM 的 3 倍, 且小于 FELIM 的 128 倍, 此标志位被硬件置位。这意味着需要两个甚至更多的档位才能够修正频率误差。如果 CRS_CR 寄存器的 SYNCWARNIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 SYNCWARNC 位清除。 0: 没有同步警告信号 1: 产生同步警告信号
0	SYNCOKF	同步行为 OK 标志 (SYNC Event OK Flag) 当测量到的频率误差小于 FELIM 的 3 倍, 此标志位被硬件置位。这意味着不需要有校准的行为, 或者一档的调节就足够修正频率误差。如果 CRS_CR 寄存器的 SYNCOKIE 位被置位, 则会产生中断。通过软件设置 CRS_ICR 寄存器的 SYNCOKC 位清除。 0: 没有同步行为 OK 信号 1: 产生同步行为 OK 信号

10.6.5 CRS_ICR 中断标志清除寄存器

地址偏移: 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.												ESYNC	ERRC	SYNCW	SYNCO
												C		ARNC	KC
												W	W	W	W

Bit	Field	Description
31: 4	Reserved	保留, 必须保持复位值
3	ESYNCC	清除预期同步标志 (Expected SYNC Clear Flag) 写'1'清除 CRS_ISR 寄存器的 ESYNCF 标志。
2	ERRC	清除错误标志 (Error Clear Flag) 写'1'清除 CRS_ISR 寄存器的 TRIMOVF, SYNCMISS, SYNCERR 位, 同样会清除 ERRF 位。
1	SYNCWARNC	清除同步警告标志 (SYNC Warning Clear Flag) 写'1'清除 CRS_ISR 寄存器的 SYNCWARNF 标志。
0	SYNCOKC	清除同步 OK 标志 (SYNC Event OK Clear Flag) 写'1'清除 CRS_ISR 寄存器的 SYNCOKF 标志。

11 HWDIV 硬件除法器

11.1 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。

11.2 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

11.3 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。

11.4 寄存器描述

11.4.1 寄存器总览

表 11-1 HWDIV 寄存器描述概览

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

11.4.2 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)													

11.4.3 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。													

11.4.4 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)

11.4.5 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)													

11.4.6 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。													

11.4.7 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

Preliminary

12 GPIO 通用端口

12.1 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。

12.2 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

12.3 功能描述

12.3.1 功能框图

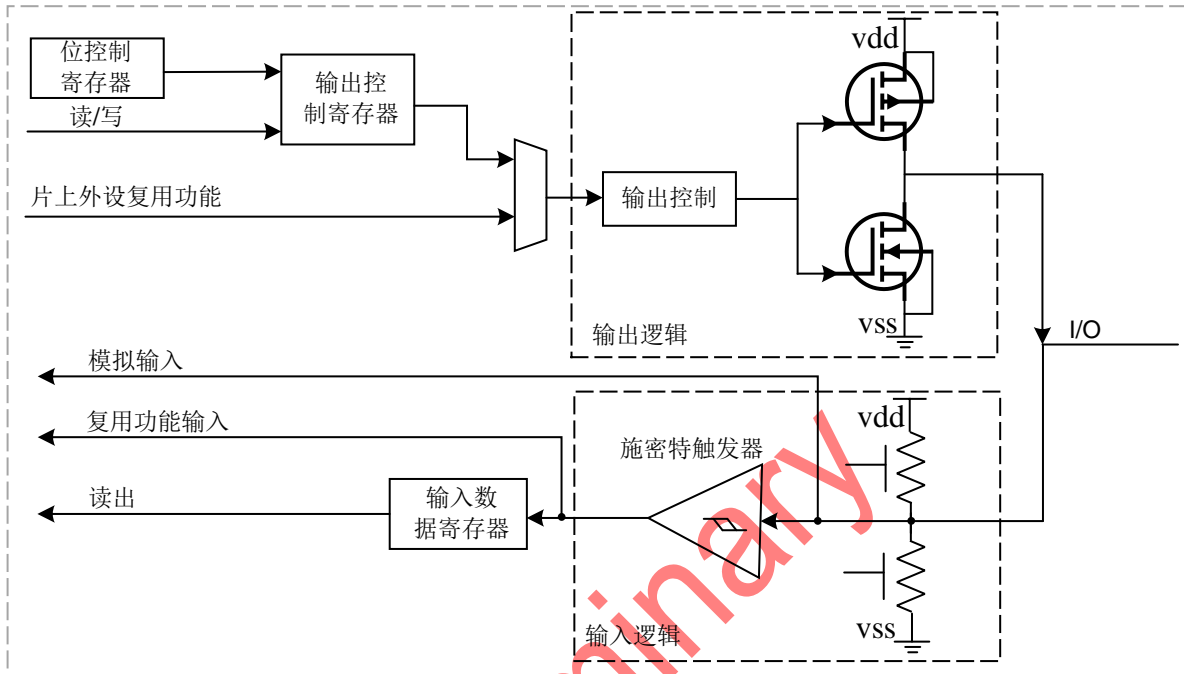


图 12-1 标准 I/O 端口

12.3.2 GPIO 端口配置

表 12-1 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	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	10 11	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

12.3.3 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。

12.3.4 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置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。

12.3.5 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：

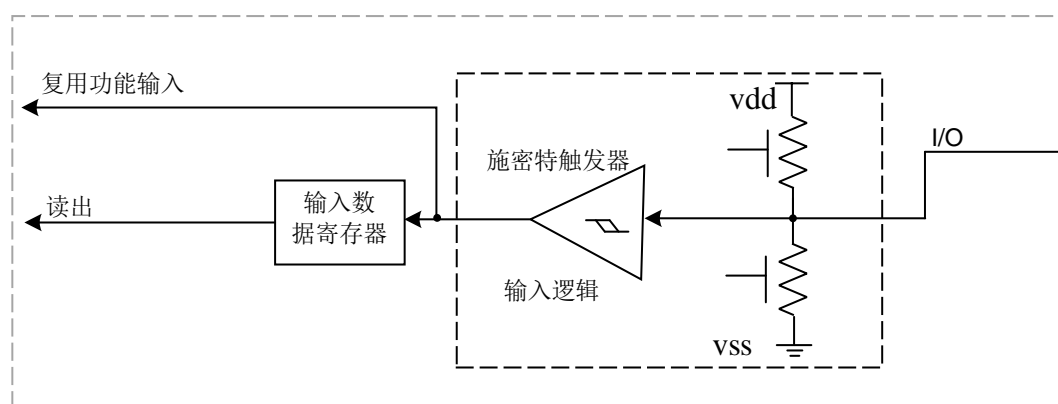


图 12-2 输入浮空/上拉/下拉配置

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。

12.3.6 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：

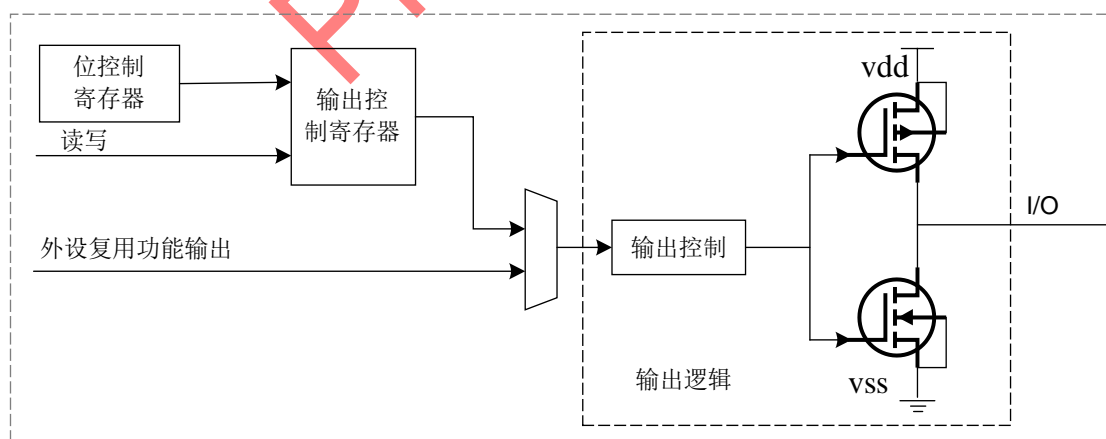


图 12-3 输出配置

12.3.7 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。

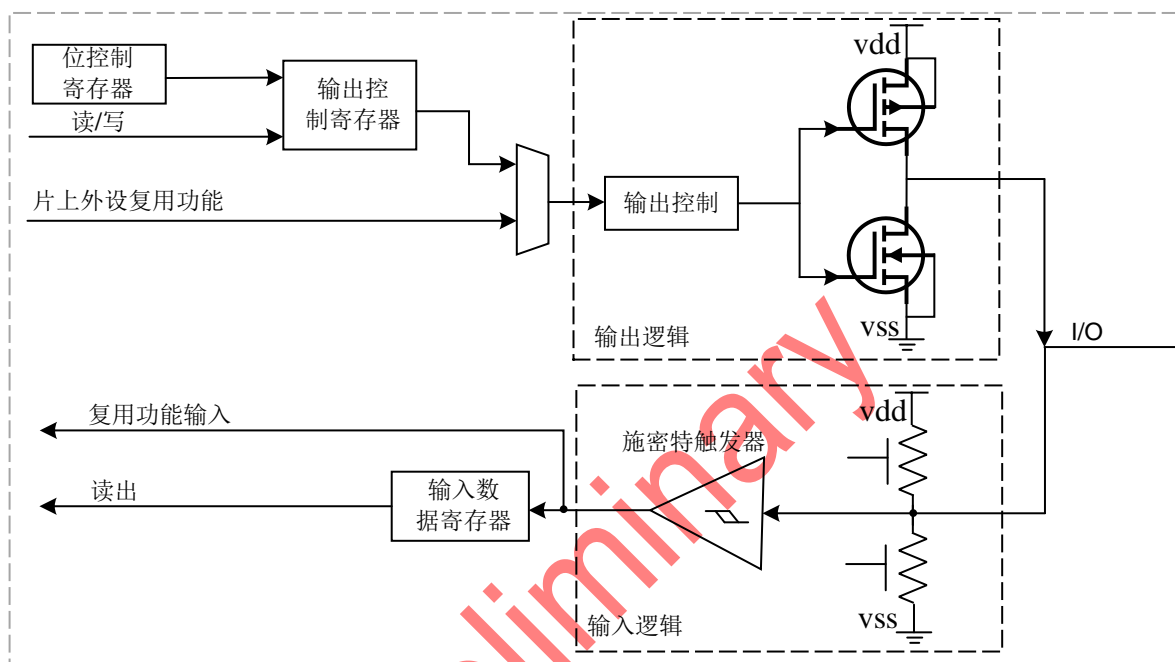


图 12-4 复用功能配置

12.3.8 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

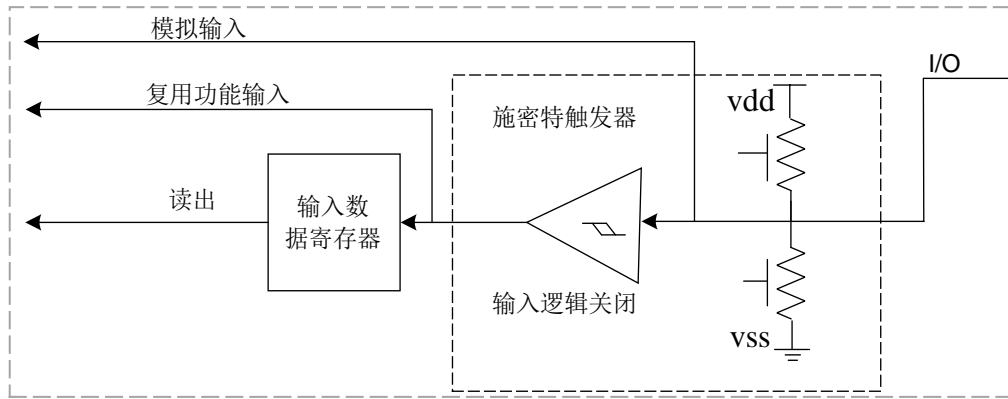


图 12-5 模拟输入

12.3.9 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。

12.3.10 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 12-2 SWD 复用功能重映射

复用功能	GPIO 端口
SWDIO	PA13
SWCLK	PA14

12.4 GPIO 寄存器描述

12.4.1 寄存器总览

表 12-3 GPIO 寄存器概览

Offset	Acronym	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	0x00000000

Offset	Acronym	Register Name	Reset
0x24	GPIOx_AFRH	端口复用功能高位寄存器	0x00000000

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。

12.4.2 GPIOx_CRL 端口配置低寄存器

偏移地址：0x00

复位值：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	端口配置位 (y=7..0)
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	端口输入输出配置 (MODEy) (y = 0..7)
9: 8	MODE2	
5: 4	MODE1	
1: 0	MODE0	

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

配置 MODEy 不等于 0 时，不同配置输出速度不同：
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

软件配置相应的 I/O 端口；参考端口位配置表

其它配置输出速度参考芯片数据手册部分。

12.4.3 GPIOx_CRH 端口配置高寄存器

偏移地址：0x04

复位值：0x4444 4444

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			
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			

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

12.4.4 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

12.4.5 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。

12.4.6 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

12.4.7 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

12.4.8 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
															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

12.4.9 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
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

12.4.10 GPIOx_AFR1 端口复用功能低位寄存器

偏移地址: 0x20

复位值: 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

12.4.11 GPIOx_AFRH 端口复用功能高位寄存器

偏移地址: 0x24

复位值: GPIOx_AFRH (x = B..I): 0xFFFF FFFF, GPIOA_AFRH: 0xF00F FFFF

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
31: 0	AFRy	<p>端口 x 的位 y (y = 8..15) 的复用功能选择位, 软件写配置。</p> <p>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</p>

Preliminary

13 EXTI 中断和事件

13.1 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。

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。

13.2 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

13.3 功能描述

13.3.1 功能框图

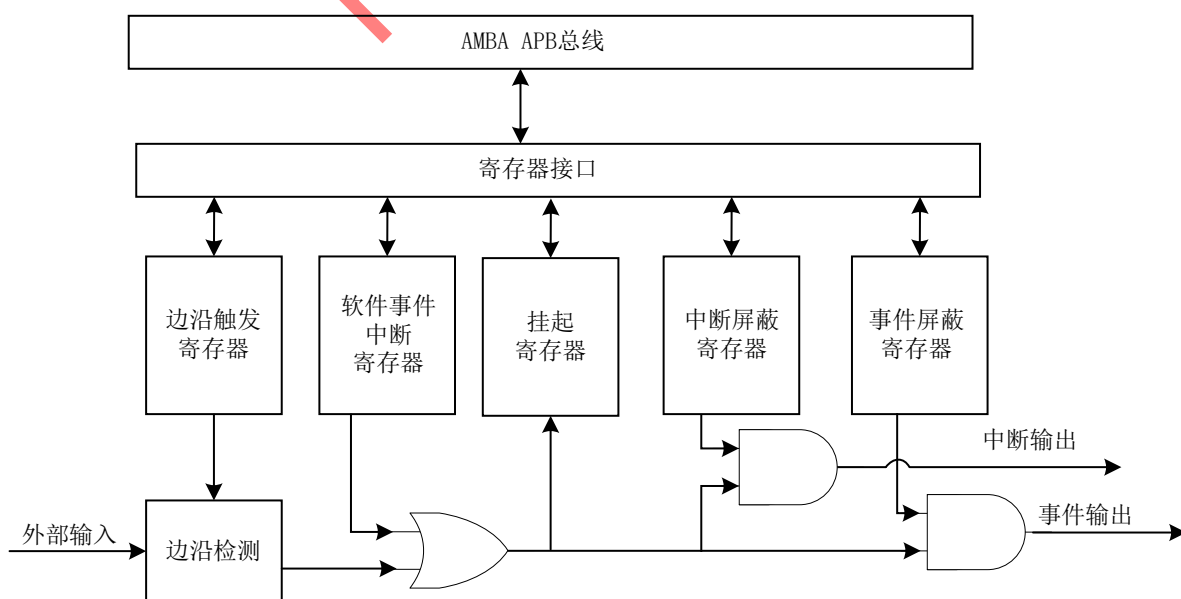


图 13-1 EXTI 结构框图

13.3.2 中断和异常向量

在 Handler 模式下，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，

13.3.3 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。

13.3.4 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。

13.3.5 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 EXTIx（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 EXTIx（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。

13.3.6 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_RTSTR/EXTI_FTSTR）。

13.3.7 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）。

13.3.8 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、RTC、USB、比较器、IWDG）触发。

13.4 寄存器描述

13.4.1 寄存器总览

表 13-1 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_RTSTR	上升沿触发选择寄存器	0x00000000
0x0C	EXTI_FTSTR	下降沿触发选择寄存器	0x00000000
0x10	EXTI_SWIER	软件中断事件寄存器	0x00000000
0x14	EXTI_PR	挂起寄存器	0x00000000

13.4.2 中断屏蔽寄存器（EXTI_IMR）

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

IMRx (x=31~16)

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: 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

13.4.3 事件屏蔽寄存器 (EXTI_EMR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMRx (x=31~16)															
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: 0	IMRx	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

13.4.4 上升沿触发选择寄存器 (EXTI_RTSR)

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
PRx (x=31~16)															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRx (x=15~0)															
rw															

Bit	Field	Description
31: 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

13.4.5 下降沿触发选择寄存器 (EXTI_FTSR)

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
TRx (31~16)															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRx (15~0)															
rw															

Bit	Field	Description
31: 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

13.4.6 软件中断事件寄存器 (EXTI_SWIER)

偏移地址: 0x10

复位值: 0x0000 0000

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

Bit	Field	Description
31: 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

13.4.7 软件中断事件挂起寄存器 (EXTI_PR)

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
PRx (x=31~16)															
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: 0	PRx	<p>线 x 触发挂起位</p> <p>1: 发生了选择的触发请求</p> <p>0: 没有发生触发请求</p> <p>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。</p>

Preliminary

14 DMA 直接存储器访问控制器

14.1 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 7 个通道，多个外设 DMA 请求发送到对应通道上处理。

14.2 功能框图

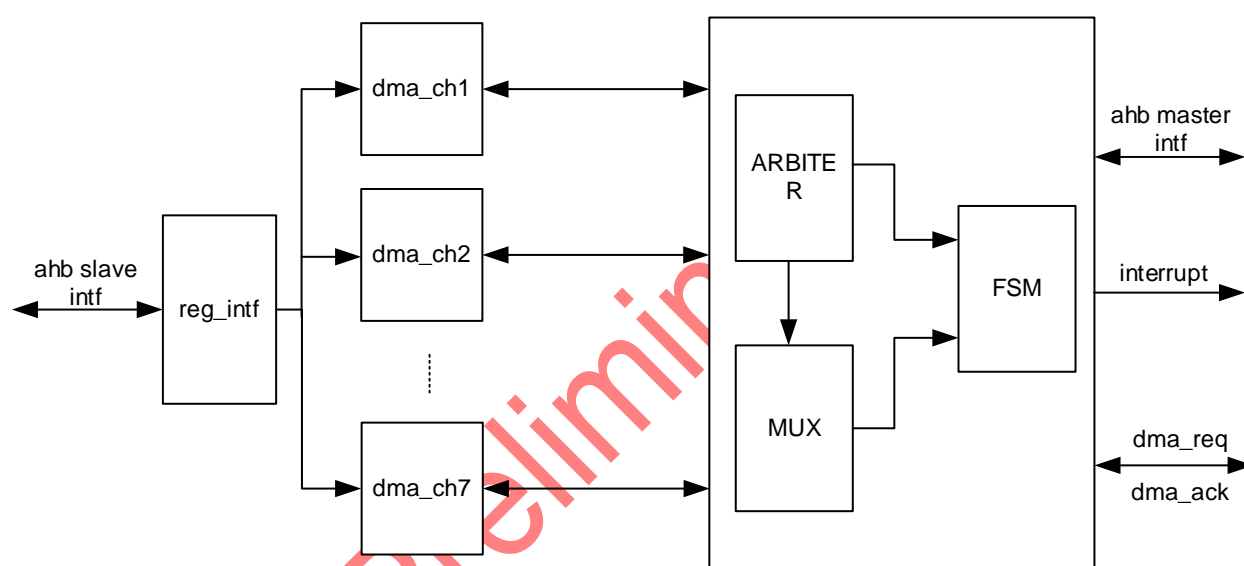


图 14-1 DMA 功能框图

14.3 主要特征

- 7 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 7 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。

14.4 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 14-1 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

14.5 DMA

14.5.1 DMA 请求映像

从外设产生的多个传输请求，通过 DMAMUX 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应有对应的 DMA 使能位，来独立控制外设是否发送传输请求。

表 14-2 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	DMA 通道 4	DMA 通道 5	DMA 通道 6	DMA 通道 7
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	-	-	-	-	-
SPI	-	SPI1_RX	SPI1_TX	SPI2_RX ⁽¹⁾	SPI2_TX ⁽¹⁾	SPI2_RX ⁽²⁾	SPI2_TX ⁽²⁾
UART	-	UART1_TX ⁽¹⁾ UART3_TX ⁽¹⁾	UART1_RX ⁽¹⁾ UART3_RX ⁽¹⁾	UART1_TX ⁽²⁾ UART2_TX ⁽¹⁾	UART1_RX ⁽²⁾ UART2_RX ⁽¹⁾	UART2_RX ⁽²⁾ UART3_RX ⁽²⁾ UART4_RX	UART2_TX ⁽²⁾ UART3_TX ⁽²⁾ UART4_TX
LPUART	LPUART_TX	LPUART_RX	-	-	-	-	-

外设	DMA 通道 1	DMA 通道 2	DMA 通道 3	DMA 通道 4	DMA 通道 5	DMA 通道 6	DMA 通道 7
I2C	-	I2C1_TX ⁽¹⁾	I2C1_RX ⁽¹⁾	-	-	I2C1_TX ⁽²⁾	I2C1_RX ⁽²⁾
I3C	-	-	-	I3C_TX	I3C_RX	-	-
TIM1	-	TIM1_CH1 ⁽¹⁾	TIM1_CH2 ⁽¹⁾	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 ⁽¹⁾ TIM1_UP TIM1_CH5	TIM1_CH1 ⁽²⁾ TIM1_CH2 ⁽²⁾ TIM1_CH3 ⁽²⁾	-
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2 ⁽¹⁾	TIM2_CH4 ⁽¹⁾	TIM2_CH1	-	TIM2_CH2 ⁽²⁾ TIM2_CH4 ⁽²⁾
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 ⁽¹⁾ TIM3_TRIG ⁽¹⁾	-	TIM3_CH1 ⁽²⁾ TIM3_TRIG ⁽²⁾	-
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⁾	-	-	-	-	-
FlexCAN	FlexCAN	-	-	-	-	-	-

- 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。
- 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。

14.6 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。

14.6.1 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 侦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。

14.6.2 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。

14.6.3 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。

14.6.3.1 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。

14.6.3.2 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。

14.6.3.3 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）。

14.6.3.4 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。

14.6.3.5 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。

14.6.4 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 14-3 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 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	8	16	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 00B0[15: 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 0x4 读 B7B6B5B4[31: 0] 在 0x8 读 BBBAB9B8[31: 0] 在 0xC 读 BFBEBDBC[31: 0]	在 0x0 写 B0[7: 0] 在 0x1 写 B4[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]

14.6.4.1 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。

这种方法确实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 $4 \times 8\text{bit} = 32\text{bit}$ 数据，如 0x12 会被扩充为 0x1212_1212。

源数据宽度 16bit，会被复制扩充为 $2 \times 16\text{bit} = 32\text{bit}$ 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 32bit。

14.6.5 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。

14.7 DMA 寄存器描述

表 14-4 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

Offset	Acronym	Register Name	Reset
0x14+20× (n - 1)	DMA_CMARx	DMA 通道 x 存储器地址寄存器	0x00000000

14.7.1 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.				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				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
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 28	Reserved	保留, 始终读为 0
27,23,19,15,11,7,3	TEIFx	通道 x 的传输错误标志 (x=1~7) (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)
26,22,18,14,10,6,2	HTIFx	通道 x 的半传输标志 (x=1~7) (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)
25,21,17,13,9,5,1	TCIFx	通道 x 的传输完成标志 (x=1~7) (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 此标志位。 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)
24,20,16,12,8,4,0	GIFx	通道 x 的全局中断标志 (x=1~7) (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 此标志位。 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

14.7.2 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.				CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31: 28	Reserved	保留，始终读为 0
27,23,19,15,11,7,3	CTEIFx	清除通道 x 的传输错误标志 (x=1~7) (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
26,22,18,14,10,6,2	CHTIFx	清除通道 x 的半传输标志 (x=1~7) (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
25,21,17,13,9,5,1	CTCIFx	清除通道 x 的传输完成标志 (x=1~7) (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
24,20,16,12,8,4,0	CGIFx	清除通道 x 的全局中断标志 (x=1~7) (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

14.7.3 DMA_CCRx DMA 通道 x 配置寄存器 (x=1~7)

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.															BURSTE
															N
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE	MEM2M EM	PL		MSIZE		PSIZE		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
r/w	r/w	r/w		r/w		r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit	Field	Description
31: 17	Reserved	保留，始终读为 0
16	BURSTEN	突发传输使能（Burst transfer enable） 该位由软件置‘1’或清零。 0: 关闭突发传输 1: 使能突发传输 Flexcan dma 传输一定要使能突发传输，其他模块 dma 传输一定要关闭突发传输
15	ARE	自动重装载（Auto reload） 该位由软件置‘1’或清零。 1: 使能自动重装载传输数量，外设地址，存储器地址 0: 禁止自动重装载传输数量，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装载
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

Bit	Field	Description
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
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

14.7.4 DMA_CNDTRx DMA 通道 x 传输数量寄存器 (x=1~7)

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													

Bit	Field	Description
15: 0	NDT	<p>数据传输数量 (Number of data to transfer)</p> <p>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新加载模式时, 寄存器的内容将被自动重新加载为之前配置时的数值。</p> <p>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。</p>

14.7.5 DMA_CPARx DMA 通道 x 外设地址寄存器 (x=1~7)

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
31: 0	PA	<p>外设地址 (Peripheral address)</p> <p>外设数据寄存器的基地址, 作为数据传输的源或目标。</p> <p>当 PSIZE='01' (16 位), 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。</p> <p>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。</p>

14.7.6 DMA_CMARx DMA 通道 x 存储器地址寄存器 (x = 1~7)

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。

Preliminary

15 COMP 比较器

15.1 简介

芯片内嵌 2 个通用比较器 (COMPx, x=1,2)，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。

15.2 功能框图

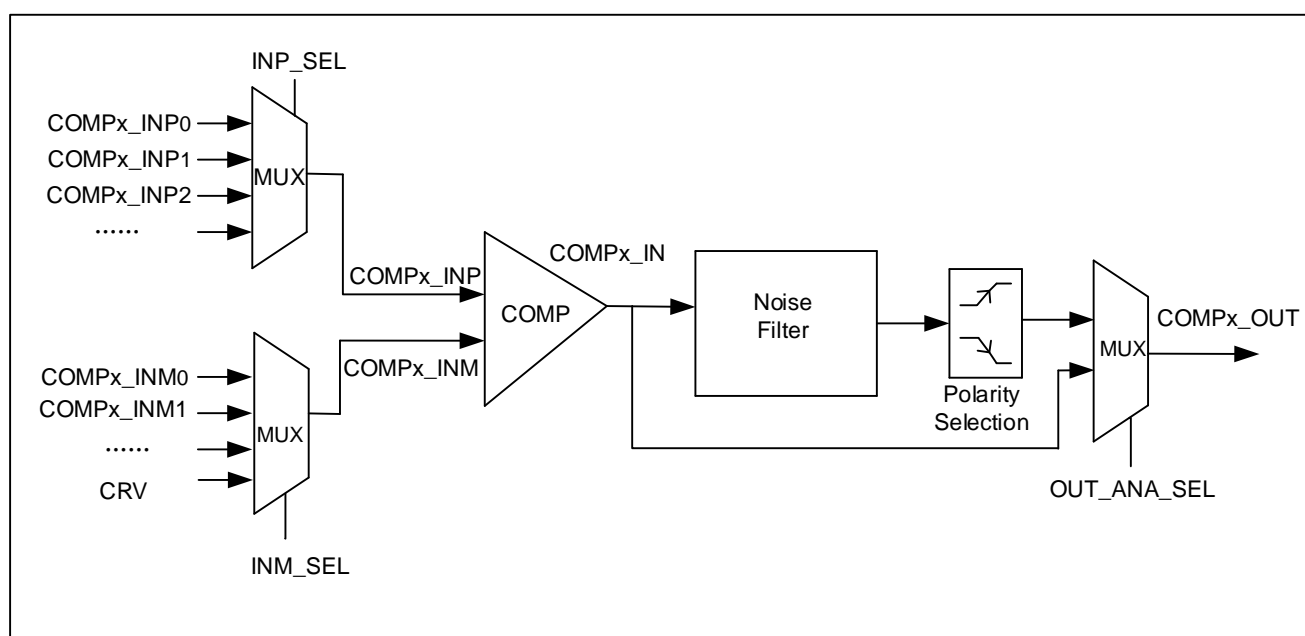


图 15-1 比较器框图

15.3 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；

- 可以实现定周期切换的轮询功能；
- 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。

15.4 功能描述

15.4.1 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。

15.4.2 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。

15.4.3 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。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。

15.4.4 比较器通道选择

每个 COMP 有四个正相输入和五个反相输入通道，正相输入可从四个外部引脚之间选取，反相输入可从四个外部引脚和 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。

15.4.5 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。

15.4.6 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。

15.4.7 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）

15.4.8 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)。

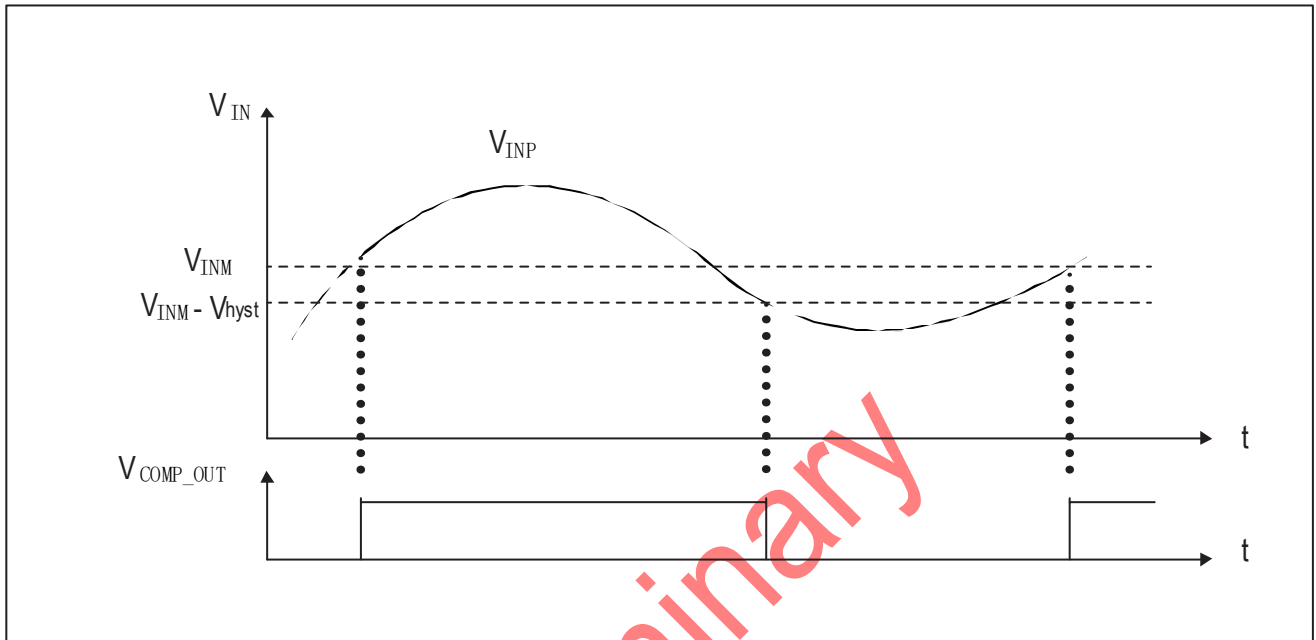


图 15-2 比较器的迟滞

15.5 比较器寄存器描述

表 15-1 COMP 寄存器概览

Offset	Acronym	Register Name	Reset
0x0,0x4	COMPx_CSR (x=1,2)	比较器 x (x=1,2) 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,0x20	COMPx_POLL (x=1, 2)	比较器 x (x=1,2) 轮询寄存器	0x0000000

15.5.1 比较器控制状态寄存器 (COMPx_CSR) (x=1, 2)

偏移地址: 0x0,0x4

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT	OUT_ANA_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	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
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。 Mode=00 时, 11: 85mV 10: 45mV 01: 22mV 00: 0mV 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。

Bit	Field	Description
13 : 10	OUT_SEL	<p>比较器 x 输出选择 (Comparator x Output Selection)</p> <p>这些位用来选择比较器 x 的输出目标。</p> <p>0010: 定时器 1 刹车输入</p> <p>0110: 定时器 1 Ocrefclear 输入</p> <p>0111: 定时器 1 输入捕捉 1</p> <p>1000: 定时器 2 输入捕捉 4</p> <p>1001: 定时器 2 OCrefclear 输入</p> <p>1010: 定时器 3 输入捕捉 1</p> <p>1011: 定时器 3 Ocrefclear 输入</p> <p>1111: LPTIMER TRIG 输入</p> <p>其他: 无选择</p>
9	Reserved	始终读为 0。
8 : 7	INP_SEL	<p>比较器 x 正相输入选择 (Comparator x Normal Phase Input Selection)</p> <p>这些位用于选择连接到比较器 x 正相输入端的信号源。</p> <p>比较器 1:</p> <p>00: COMP1_INP0 (PA1), 即正相输入端通道 0;</p> <p>01: COMP1_INP1 (PA2), 即正相输入端通道 1;</p> <p>10: COMP1_INP2 (PA3), 即正相输入端通道 2;</p> <p>11: COMP1_INP3 (PA4), 即正相输入端通道 3;</p> <p>比较器 2:</p> <p>00: COMP2_INP0 (PA0), 即正相输入端通道 0;</p> <p>01: COMP2_INP1 (PA1), 即正相输入端通道 1;</p> <p>10: COMP2_INP2 (PA2), 即正相输入端通道 2;</p> <p>11: COMP2_INP3 (PA3), 即正相输入端通道 3;</p>
6	Reserved	始终读为 0。
6 : 4	INM_SEL	<p>比较器 x 反相输入选择 (Comparator x Inverting Input Selection)</p> <p>这些位用于选择连接到比较器 x 反相输入端的信号源。</p> <p>比较器 1:</p> <p>000: COMP1_INM0 (PA5), 即反相输入端通道 0;</p> <p>001: COMP1_INM1 (PA6), 即反相输入端通道 1;</p> <p>010: COMP1_INM2 (PA7), 即反相输入端通道 2;</p> <p>011: COMP1_INM3 (PA0), 即反相输入端通道 3;</p> <p>100: COMP1_INM4 (CRV), 即反相输入端通道 4;</p> <p>其他: 无选择</p> <p>比较器 2:</p> <p>000: COMP2_INM0 (PA4), 即反相输入端通道 0;</p> <p>001: COMP2_INM1 (PA5), 即反相输入端通道 1;</p> <p>010: COMP2_INM2 (PA6), 即反相输入端通道 2;</p> <p>011: COMP2_INM3 (PA2), 即反相输入端通道 3;</p> <p>100: COMP2_INM4 (CRV), 即反相输入端通道 4;:</p> <p>其他: 无选择</p>

Bit	Field	Description
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

15.5.2 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_SRC	CRV_EN	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)
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}

15.5.3 比较器轮询寄存器 (COMPx_POLL) (x=1, 2)

偏移地址: 0x1C,0x20

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					POUT			Res.	PERIOD			Res.	FIXN	POLL_C H	POLL_E N
					r				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。

Bit	Field	Description
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。
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

16 ADC 模拟/数字转换

16.1 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。

ADC 有 16 个通道可测量内部或外部信号源，其中 ADC 有 14 路外部输入通道和 2 路内部通道。ADC 的通道可以单次、单周期和连续进行转换。根据不同的方式又可以选择普通通道转换、任意通道转换、注入通道转换。

ADC 的输入时钟不得超过 16M,他是由 APB2 时钟（PCLK2）分频产生。

16.2 功能框图

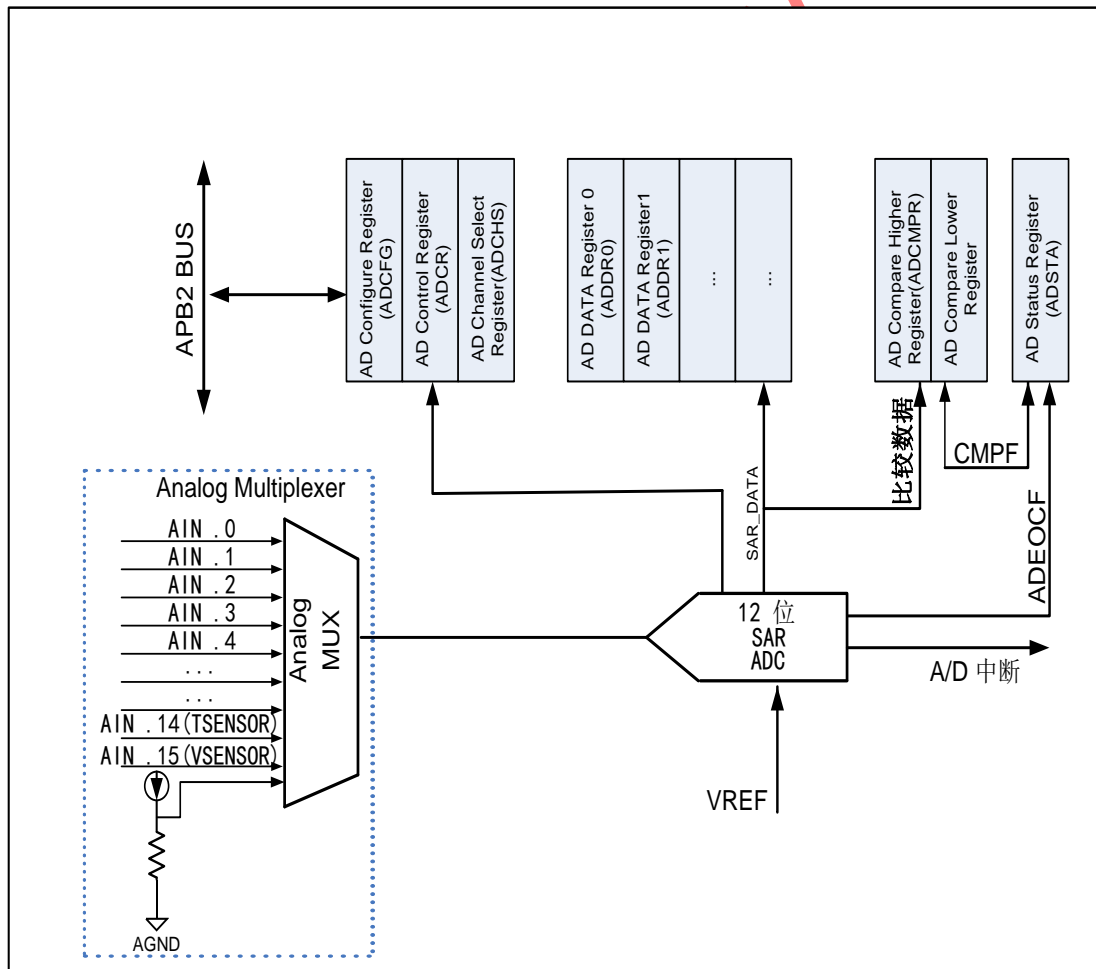


图 16-1 ADC 系统框图

注：T_SENSOR（温度传感器）通道在 ADC 的 AIN14 通道，V_SENSOR（内部参考电压）通道在 ADC 的 AIN15 通道。

16.3 主要特征

- 高达 1Msps 转换速率
- 支持普通通道转换
 - ◆ 单次转换模式：在指定通道完成一次转换
 - ◆ 单周期扫描模式：对所有指定通道（从低序号通道到高序号通道，或从高序号通道到低序号通道）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。若需要修改转换通道只能停止 A/D 转换，等待完成寄存器配置再重新开启转换
- 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
- 支持注入通道转换
 - ◆ 自动注入：在任意通道转换方式下，完成任意通道转换后自动开始进行注入通道工作
 - ◆ 事件注入：在任意通道转换方式下，注入事件发生后（包括软件和触发），如果当前转换正在进行，则立即停止当前任意转换后开始注入通道转换。如果当前无转换，则开始注入通道转换。注入通道转换完成后，继续按照配置完成任意通道转换直到 ADC_ADCR.ADST 位被软件或硬件清除为止。
- 可编程通道采样时间
- 最高 12 位可编程分辨率 SAR
- 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

16.4 中断

ADC 的中断及标志位如下表所示。当配置中断使能后，中断事件发生将产生相应的中断请求。

表 16-1 中断事件一览表

中断	标志位
模拟看门狗比较器中断	AWDIF
通道序列转换结束中断	EOSIF
单通道转换结束中断	EOCIF
采样结束中断	EOSMPIF
注入通道序列转换结束中断	JEOSIF
注入通道转换结束中断	JEOCIF
注入通道采样结束中断	JEOSMPIF

16.5 DMA

A/D 转换结果存储在数据寄存器 ADC_ADDDATA 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_ADDDATA 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。

16.6 功能描述

16.6.1 时钟

ADC 的输入时钟与 PCLK2 同步。在使用 ADC 之前，要先使能 RCC 控制器中的 ADC 时钟控制位。

16.6.2 数据补偿

非注入通道数据没有数据补偿。

注入通道转换数据减去 ADC_JOFRn 寄存器定义的偏移量后保存到注入通道数据寄存器 ADC_JADDATA 和 ADC_JDRn。补偿后的结果可能为负值，因此注入通道转换的结果是有符号数据（数据对齐 SEXT 位是扩展的符号值）。

如果配置 ADC_JOFRn.JOFFSET 不为 0，则注入通道数据寄存器 ADC_JADDATA 和 ADC_JDRn 数据为有符号值，否则为无符号值。

下面表格描述了数据分辨率与数据补偿左对齐关系：

表 16-2 数据分辨率与数据补偿左对齐关系

分辨率选择	模拟转换数据 SAR_DATA, 左移	补偿值	结果	说明
12bit	SAR_DATA[11: 0]	ADC_JOFRn.JOFFESET	12 位有符号数据	-

分辨率选择	模拟转换数据 SAR_DATA, 左移	补偿值	结果	说明
11bit	SAR_DATA[11: 1],0	ADC_JOFRn.JOFFESET	11 位有符号数据	用户必须配置 ADC_JOFRn.JOFFESET[0]为“0”
10bit	SAR_DATA[11: 2],00	ADC_JOFRn.JOFFESET	10 位有符号数据	用户必须配置 ADC_JOFRn.JOFFESET[1:0]为“00”
9bit	SAR_DATA[11: 3],000	ADC_JOFRn.JOFFESET	9 位有符号数据	用户必须配置 ADC_JOFRn.JOFFESET[2:0]为“000”
8bit	SAR_DATA[11: 4],0000	ADC_JOFRn.JOFFESET	8 位有符号数据	用户必须配置 ADC_JOFRn.JOFFESET[3:0]为“0000”

16.6.3 数据对齐

通过配置 ADC_ADCR.ALIGN 位，可以选择转换后数据储存格式为左对齐或右对齐。

如下图所示：

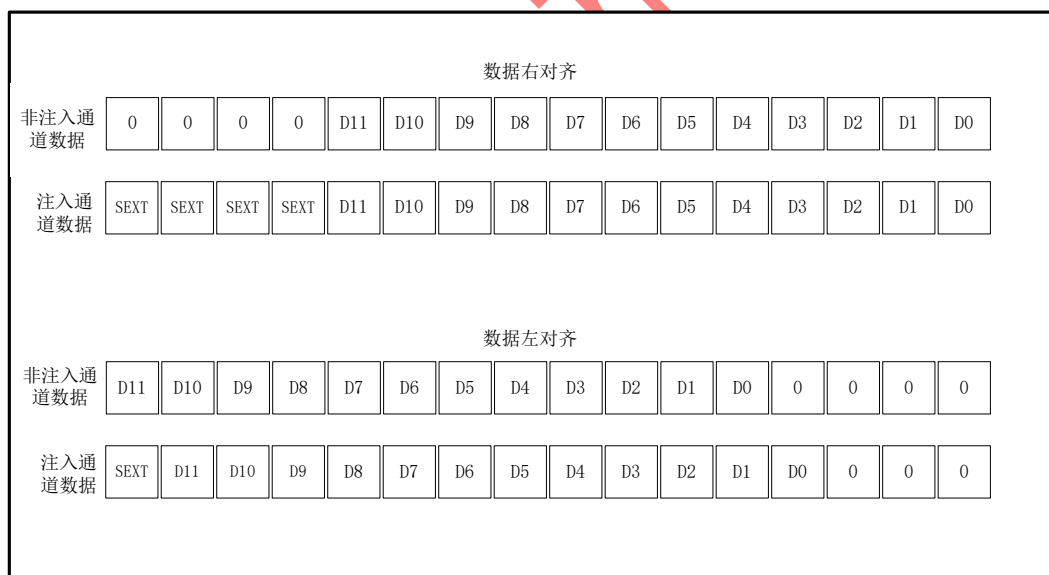


图 16-2 数据对齐方式

16.6.4 可编程分辨率

ADC 转换有效位可通过 ADC_ADCFG.RSLTCTL[2:0]位配置，有效数据位默认是 12 位数据右对齐，通过配置低分辨率可加快数据转换速率。

16.6.5 可编程采样时间

ADC 转换通道的采样时间可通过 ADC_SMPR1 和 ADC_SMPR2 寄存器中的 SAMPCTL0~SAMPCTL15 配置。

ADC 的转换时钟 ADC_CLK 由 PCLK2 分频得到，分频系数可通过设置 ADC_ADCFG.ADCPRE 位来确定，即 $PCLK2 / (ADCPRE + 2)$ 分频后作为 ADC 转换时钟。ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目 m 可以通过配置 ADC_SMPR1 和 ADC_SMPR2 寄存器来更改。设置 ADC 分辨率为 n 位 (n=8, 9, 10, 11, 12)，每个通道采样周期为 $m * T$ (T 为 ADC 模块的时钟周期)。

转换频率计算如下：

$$F_{\text{sample}} = F_{\text{ADC_CLK}} / (m + n + 0.5)。$$

假设分辨率 n 配置为 12bit，每个通道采样周期 m 为 3.5T，则 $F_{\text{sample}} = F_{\text{ADC_CLK}} / 16。$

总转换时间计算如下：

$T_{\text{CONV}} = \text{采样时间} + 12.5 \text{ 个转换周期}。$

例如当 ADC_CLK=16MHz，采样时间为 3.5T， $T_{\text{CONV}} = (3.5 + 12.5) * T = 16 * T$ ，总转换时间为 1 μ s。

16.6.6 数据通道寄存器

ADC 的转换完成后，非注入通道转换结果存储在寄存器 ADC_ADDDATA 中，ADC_ADDDATA.CHANNELSEL 表示当前数据对应的通道号。

注入通道转换结果存储在寄存器 ADC_JADDATA 中，ADC_JADDATA.JCHANNELSEL 表示当前数据对应的注入通道号。

16.6.7 通道选择

ADC 有 14 路外部输入通道 0~13、内部温度传感器通道 14 和内部 1.2V 参考电压通道 15。

不同工作模式下可以通过不同寄存器来使能通道：普通通道转换可设置寄存器 ADC_ADCHS；任意通道转换可设置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1，任意通道转换顺序从寄存器位 CHANY_SEL0~15 优先级由高到低依次转换；注入通道转换可设置寄存器 JSQR，注入通道转换顺序从寄存器位 JSQ0~3 优先级由高到低依次转换。

16.7 ADC 开关

ADC_ADCFG.ADEN 位可控制 A/D 转换模块，ADEN 位为 0 时，模拟转换模块进入掉电模式。设置 ADEN 位为 1，A/D 转换模块从掉电模式唤醒，通过清除 ADC_ADCFG.ADEN 位可以停止转换且 ADC 进入掉电模式，掉电后 ADC 几乎不耗电。

16.7.1 普通通道转换

16.7.1.1 单次采样

配置 ADC_ADCR.ADMODE 为单次转换，配置的通道执行一次转换操作后进入空闲状态，具体流程如下：

- 通过外部触发输入、定时器触发、软件三种方式可以置位 ADC_ADCR.ADST，开始 A/D 转换。
- 采样输出 SAMPL 默认被置位 2.5 个 ADC 时钟后清除。
- A/D 转换完成时，SAR 转换结果 SAR_DATA 将存储于数据寄存器 ADC_ADDDATA 和 ADC_ADDRn 中。
- A/D 转换结束时，状态寄存器 ADC_ADSTA.EOSIF 位置 1，若此时控制寄存器 ADC_ADCR.EOSIE 位置 1，将产生 AD 转换结束中断请求。
- A/D 转换期间，ADC_ADCR.ADST 位保持为 1，A/D 通道转换结束以后，ADC_ADCR.ADST 位自动硬件清 0，A/D 转换器进入空闲状态。

注：在单次转换模式下，如果软件使能多于一个通道，序号最小的通道被转换，其它通道被忽略。

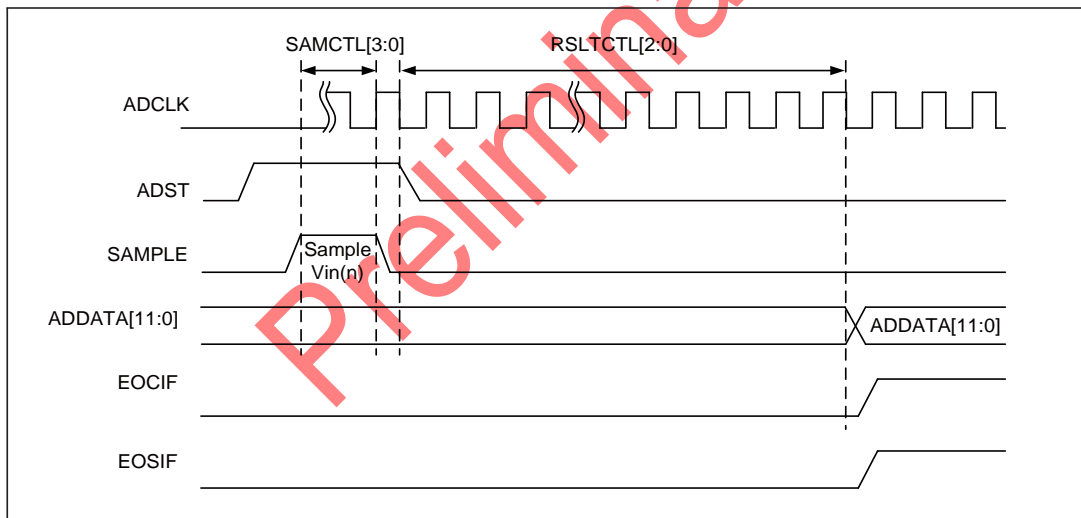


图 16-3 单次转换模式时序图

16.7.1.2 单周期扫描模式

配置 ADC_ADCR.ADMODE 为单周期转换，在单周期扫描模式下，通过配置 ADC_ADCR.SCANDIR 选择扫描通道方向，按 ADC_ADCHS.CHENy (y=0~15) 使能的通道顺序进行 A/D 转换，操作步骤如下：

- 触发或软件置位 ADC_ADCR.ADST 可以开启 A/D 转换，触发事件可配置触发延时，方向设置默认从最小序号通道到最大序号通道的 A/D 转换。
- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器

ADC_ADDDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，ADC_ADSTA.EOSIF 序列转换结束标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。

- 所有 A/D 通道转换结束后，ADC_ADCR.ADST 位自动硬件清 0，A/D 转换器会进入空闲状态。

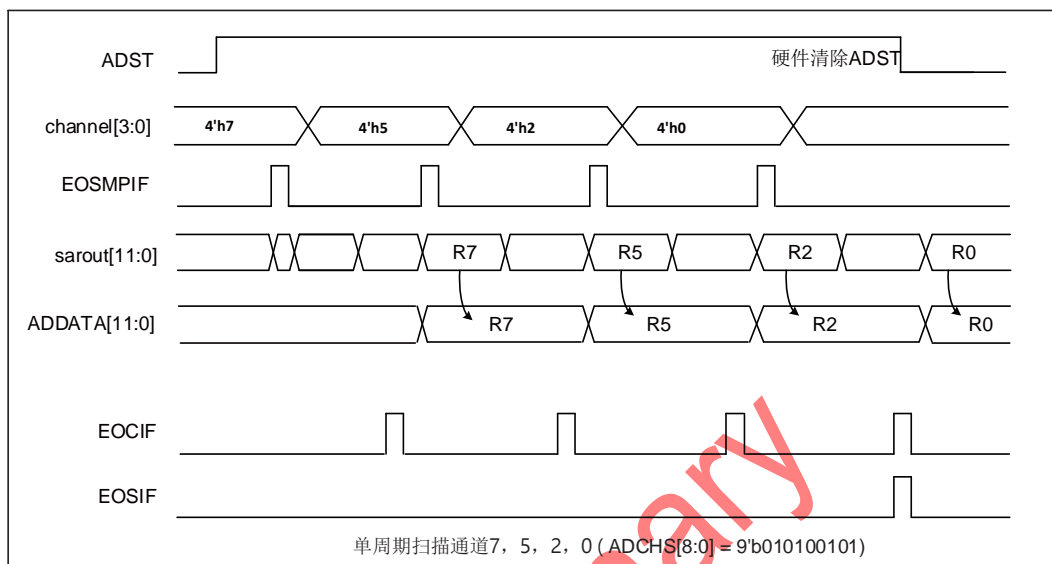


图 16-4 单周期扫描下使能通道转换时序图（通道方向从高到低）

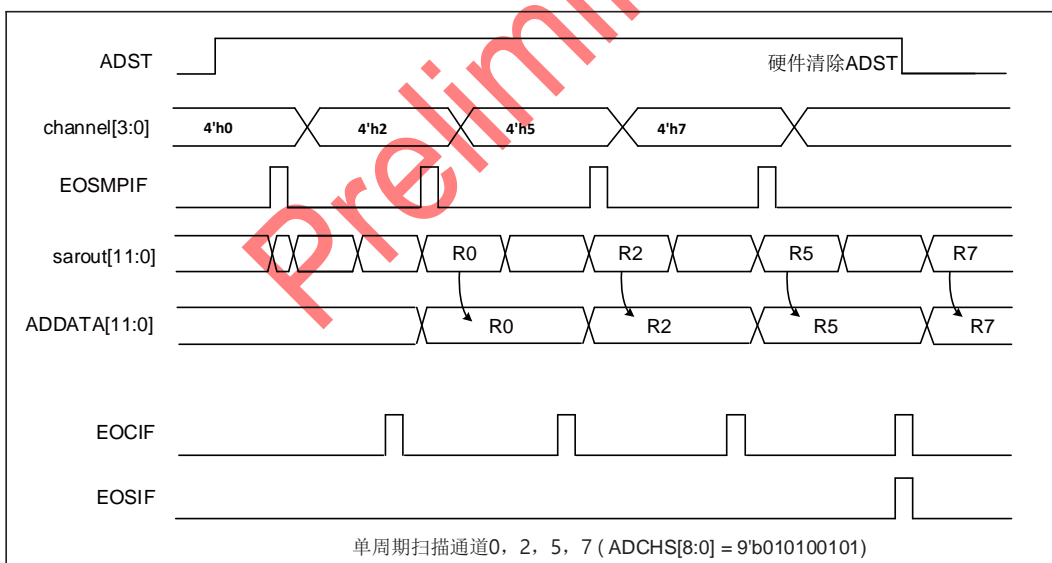


图 16-5 单周期扫描下使能通道转换时序图（通道方向从低到高）

16.7.1.3 连续扫描模式

配置 ADC_ADCR.ADMODE 为连续转换，在连续扫描模式下，通过配置 ADC_ADCR.SCANDIR 选择扫描通道方向，按 ADC_ADCHS.CHENy (y=0~15) 使能的通道顺序进行 A/D 转换，操作步骤如下：

- 触发或软件置位 ADC_ADCR.ADST 可以开启 A/D 转换，触发事件可配置触发延时，方向设置默认从最小序号通道到最大序号通道的 A/D 转换。

- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，ADC_ADSTA.EOSIF 序列转换结束标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。
- 只要 ADC_ADCR.ADST 位保持为 1，就可以持续进行 A/D 转换。当 ADC_ADCR.ADST 位被清除，转换停止，进入空闲状态。当 ADC_ADCR.ADST 位被清除，A/D 转换将结束当前转换。

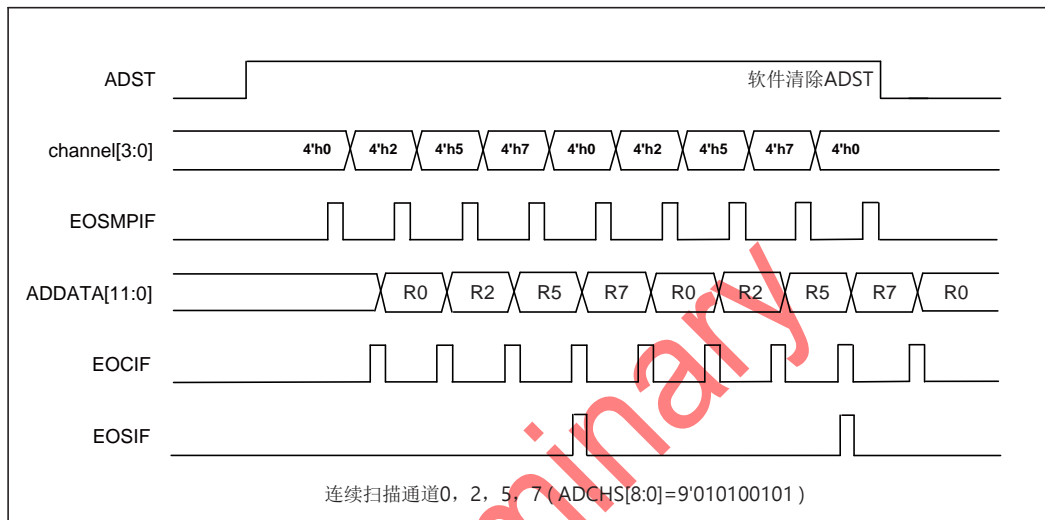


图 16-6 连续扫描模式使能通道转换时序图（通道方向由低到高）

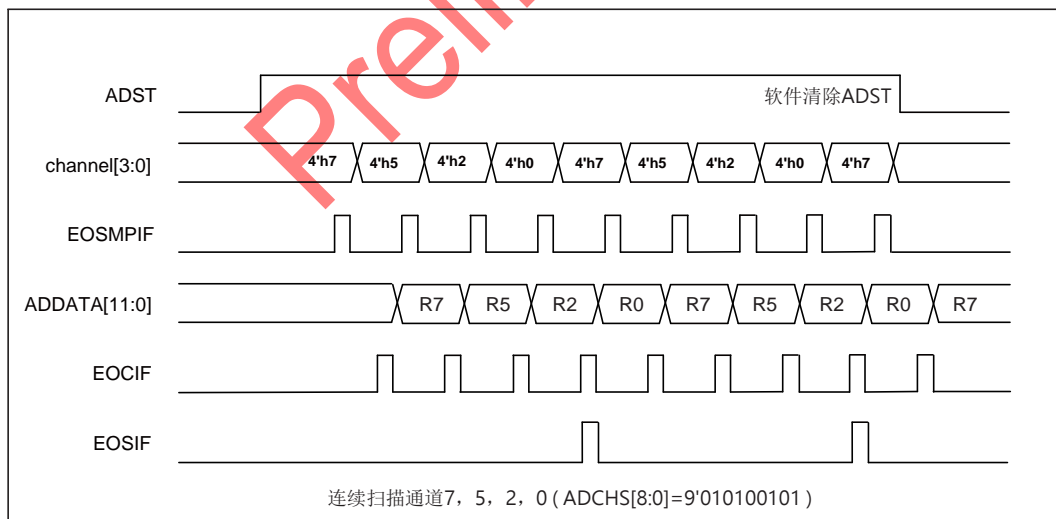


图 16-7 连续扫描模式使能通道转换时序图（通道方向由高到低）

16.7.2 任意通道转换

16.7.2.1 单次转换模式

配置 ADC_ADCR.ADMODE 为单次转换，配置的通道执行一次转换操作后进入空闲状态。具体流程如下：

- 软件配置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1，将需要转换的通道配置结束后置位 ADC_ANY_CR.CHANY_MDEN。
- 通过触发或软件置位 ADC_ADCR.ADST，触发事件可配置触发延时，转换通道选择可通过寄存器 ADC_CHANY0 和 ADC_CHANY1 任意配置。
- 采样输出 SAMPL 默认被置位 2.5 个 ADC 时钟后清除。
- A/D 转换结束后，A/D 转换结果 SAR_DATA 将装载到 ADC_ADDDATA 和 ADC_ADDR0，通道转换结束后 ADC_ADSTA.EOSIF 标志被置位，如果设置使能了转换结束中断，则产生转换结束中断请求。
- 所有 A/D 通道转换结束后，ADST 位将被硬件自动清除，A/D 转换器会进入空闲状态。

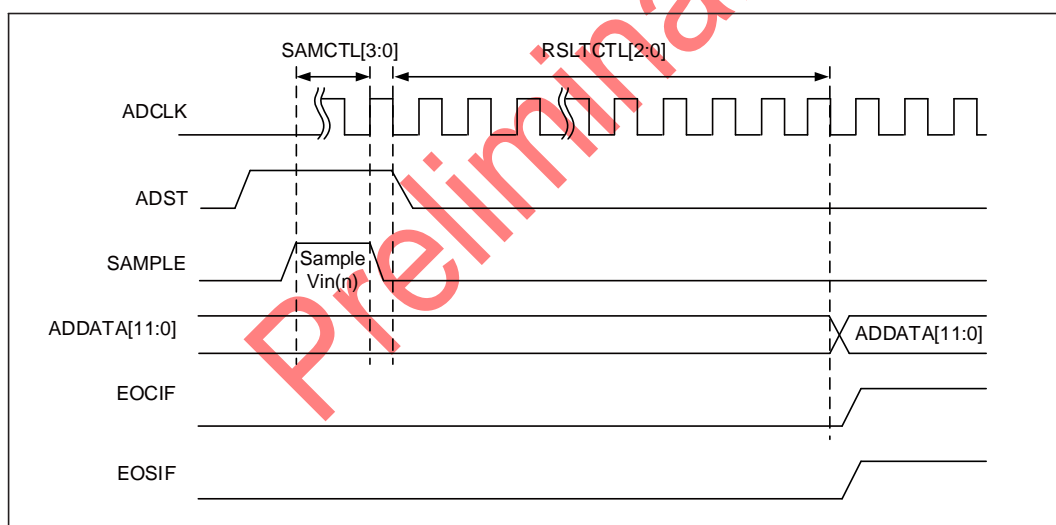


图 16-8 单次转换模式下通道转换时序图

16.7.2.2 单周期扫描模式

配置 ADC_ADCR.ADMODE 为单周期转换，在单周期扫描模式下，A/D 转换通道按照软件配置执行一遍，具体流程如下：

- 软件配置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1，将需要转换的通道和数量配置结束后置位 ADC_ANY_CR.CHANY_MDEN。
- 通过触发或软件置位 ADC_ADCR.ADST，触发事件可配置触发延时。A/D 转换通道数量由 ADC_ANY_CFG.CHANY_NUM 配置，每次转换的输入通道可以由寄存器位 CHANY_SEL0 到

CHANY_SEL15 任意配置，可以完全相同或不同。

- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，ADC_ADSTA.EOSIF 序列转换结束标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。
- 所有 A/D 通道转换结束后，ADST 位硬件自动清 0，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 寄存器后，这些配置不会立即生效，等待所有通道转换结束后生效。软件置位 ADC_ADCR.ADST 开启下一次转换。

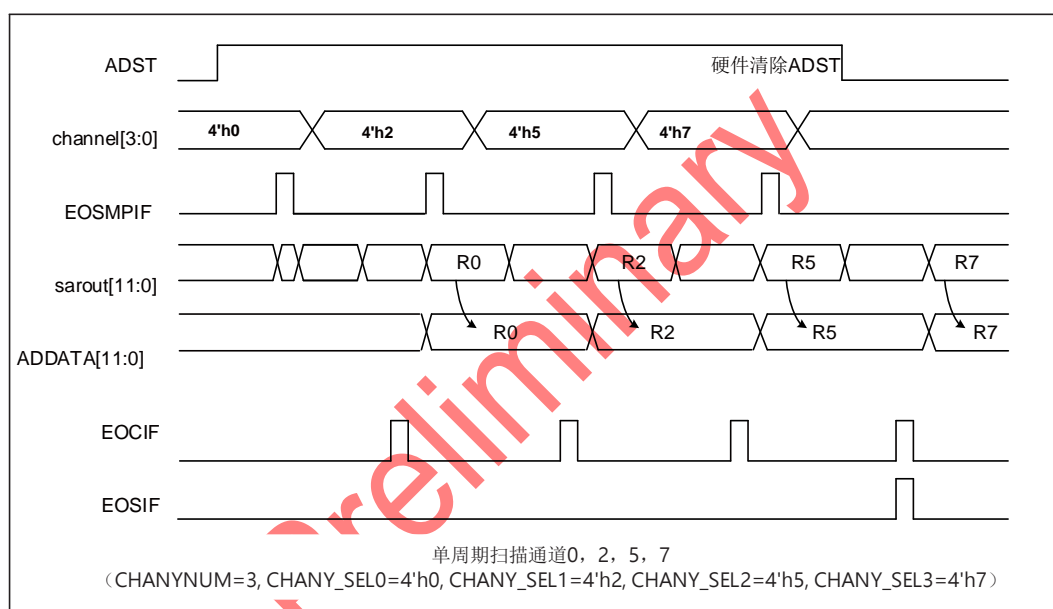


图 16-9 单周期扫描下通道转换时序图

16.7.2.3 连续扫描模式

配置 ADC_ADCR.ADMODE 为连续转换，在连续扫描模式下，A/D 转换通道按照软件配置一直执行，直到软件禁止。具体流程如下：

- 软件配置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1，将需要转换的通道和数量配置结束后置位 ADC_ANY_CR.CHANY_MDEN。
- 通过触发或软件置位 ADC_ADCR.ADST，触发事件可配置触发延时。A/D 转换通道数量由 ADC_ANY_CFG.CHANY_NUM 配置，每次转换的输入通道可以由寄存器位 CHANY_SEL0 到 CHANY_SEL15 任意配置，可以完全相同或不同。
- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器

ADC_ADDDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。

- 只要 ADC_ADCR.ADST 位保持为 1，就可以持续进行 A/D 转换。当 ADC_ADCR.ADST 位被软件清除，当前 A/D 转换结束后停止，A/D 转换器进入空闲状态。
- 若在 A/D 转换过程中，用户更新 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1 寄存器后，这些配置不会立即生效，等待所有通道转换结束后生效，即下一个扫描周期开始新的通道转换。

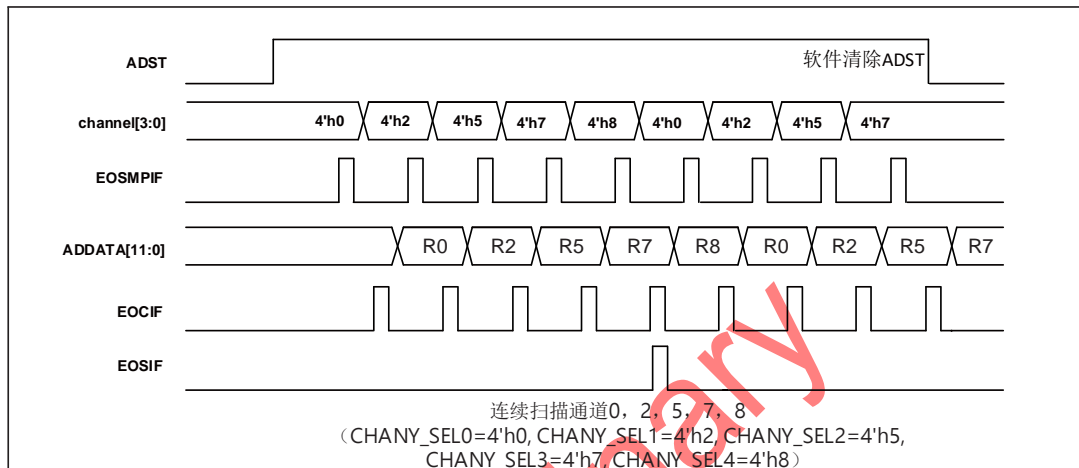


图 16-10 连续扫描模式下通道转换时序图

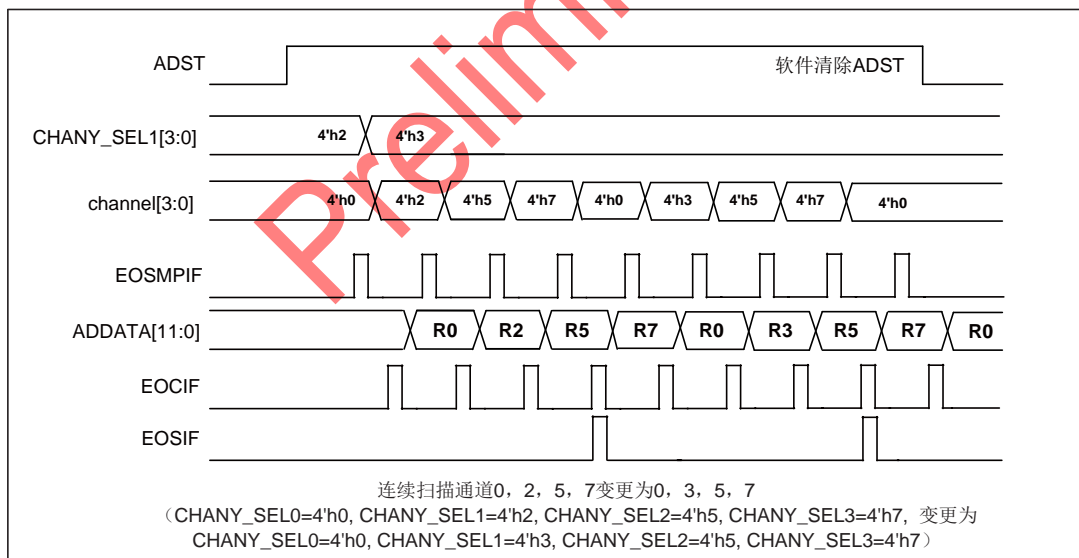


图 16-11 连续扫描模式下动态更新配置时序图

16.7.3 注入通道转换

16.7.3.1 自动注入通道转换

在配置任意通道转换后，如果使能了自动注入通道转换，任意通道转换结束后，自动进入注入通道的

转换。如果任意通道是连续扫描模式，需要清除 ADC_ADCR.ADST 才能停止 A/D 转换。

- 软件配置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1、ADC_JSQR，置位 ADC_ANY_CFG.JAUTO，将需要转换的通道和数量配置结束后置位 ADC_ANY_CR.JCEN、ADC_ANY_CR.CHANY_MDEN。
- 通过触发或软件置位 ADC_ADCR.ADST，触发事件可配置触发延时，A/D 转换任意通道数量由 ADC_ANY_CFG.CHANY_NUM 配置，每次转换的输入通道可以由寄存器位 CHANY_SEL0 到 CHANY_SEL15 任意配置。任意通道转换结束后开始进行注入通道转换，转换通道数量由 JCHANY_NUM 配置，每次转换的输入通道可以由寄存器位 JSQ0 到 JSQ3 配置。
- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDATA 和 ADC_ADDRn、ADC_JDATA 和 ADC_JDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 和 ADC_ADSTA_EXT.JEOCIF 转换结束标志被置位，如果设置使能了转换结束中断，则产生通道转换结束中断请求。连续转换结束后 ADC_ADSTA.EOSIF 和 ADC_ADSTA_EXT.JEOSIF 标志被置位，如果设置使能了序列转换结束中断，产生序列转换结束中断请求。
- 如果是非连续扫描模式，ADC_ADCR.ADST 在转换结束后被硬件自动清除。连续模式下只要 ADC_ADCR.ADST 位保持为 1，就可以持续进行 A/D 转换。当 ADC_ADCR.ADST 位被清除，当前 A/D 转换结束后停止，A/D 转换器进入空闲状态。

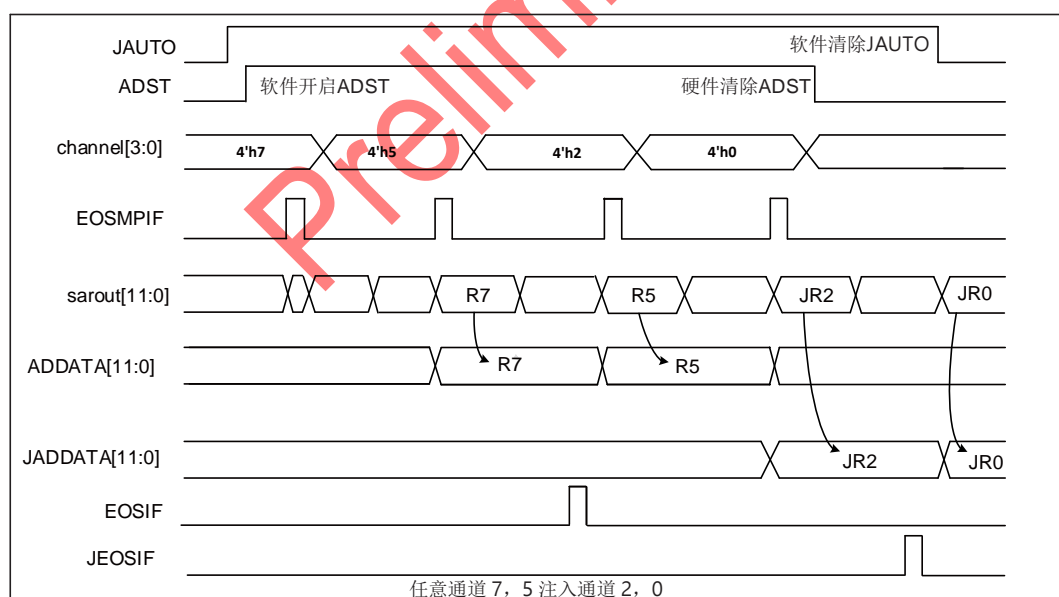


图 16-12 自动注入通道转换单周期扫描通道转换时序图

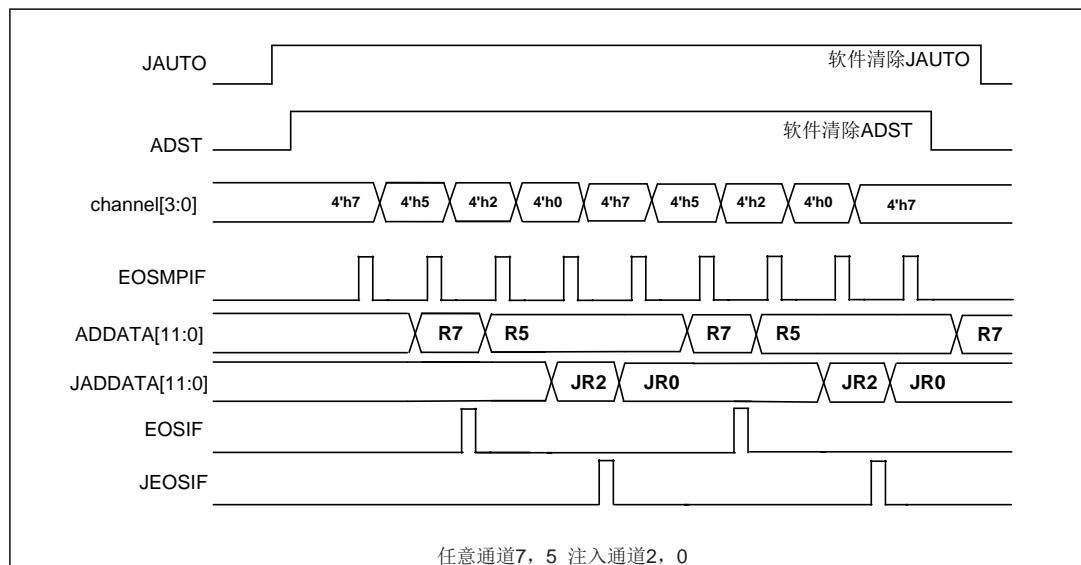


图 16-13 自动注入通道转换连续转换时序图

16.7.3.2 事件注入工作模式

关闭自动注入后，在任意通道转换方式下，如果当前任意通道转换正在进行，注入事件发生后（包括软件和触发），则立即停止当前转换后开始进行注入通道转换，等待所有的注入通道按照寄存器 ADC_JSQR 配置完成注入通道转换后，继续完成剩下的任意通道转换直到 ADC_ADCR.ADST 位被软件或硬件清除为止；如果当前无转换，则直接开始进行注入通道转换。

- 软件配置寄存器 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1、ADC_JSQR，将需要转换的通道和数量配置结束后置位 ADC_ANY_CR.JCEN、ADC_ANY_CR.CHANY_MDEN。
- 通过触发或软件置位 ADC_ADCR.ADST，触发事件可配置触发延时，A/D 转换任意通道数量由 ADC_ANY_CFG.CHANY_NUM 配置，每次转换的输入通道可以由寄存器位 CHANY_SEL0 到 CHANY_SEL15 任意配置。
- 注入通道转换，转换通道数量由 JCHANY_NUM 配置，每次转换的输入通道可以由寄存器位 JSQ0 到 JSQ3 配置。
- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDATA 和 ADC_ADDRn、ADC_JDATA 和 ADC_JDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 和 ADC_ADSTA_EXT.JEOCIF 转换结束标志被置位，如果设置使能了转换结束中断，则产生通道转换结束中断请求。连续转换结束后 ADC_ADSTA.EOSIF 和 ADC_ADSTA_EXT.JEOSIF 标志被置位，如果设置使能了序列转换结束中断，产生序列转换结束中断请求。
- 如果是非连续扫描模式，ADC_ADCR.ADST 在转换结束后被硬件自动清除。连续模式下只要 ADC_ADCR.ADST 位保持为 1，就可以持续进行 A/D 转换。当 ADC_ADCR.ADST 位被清除，

当前 A/D 转换结束后停止，A/D 转换器进入空闲状态。

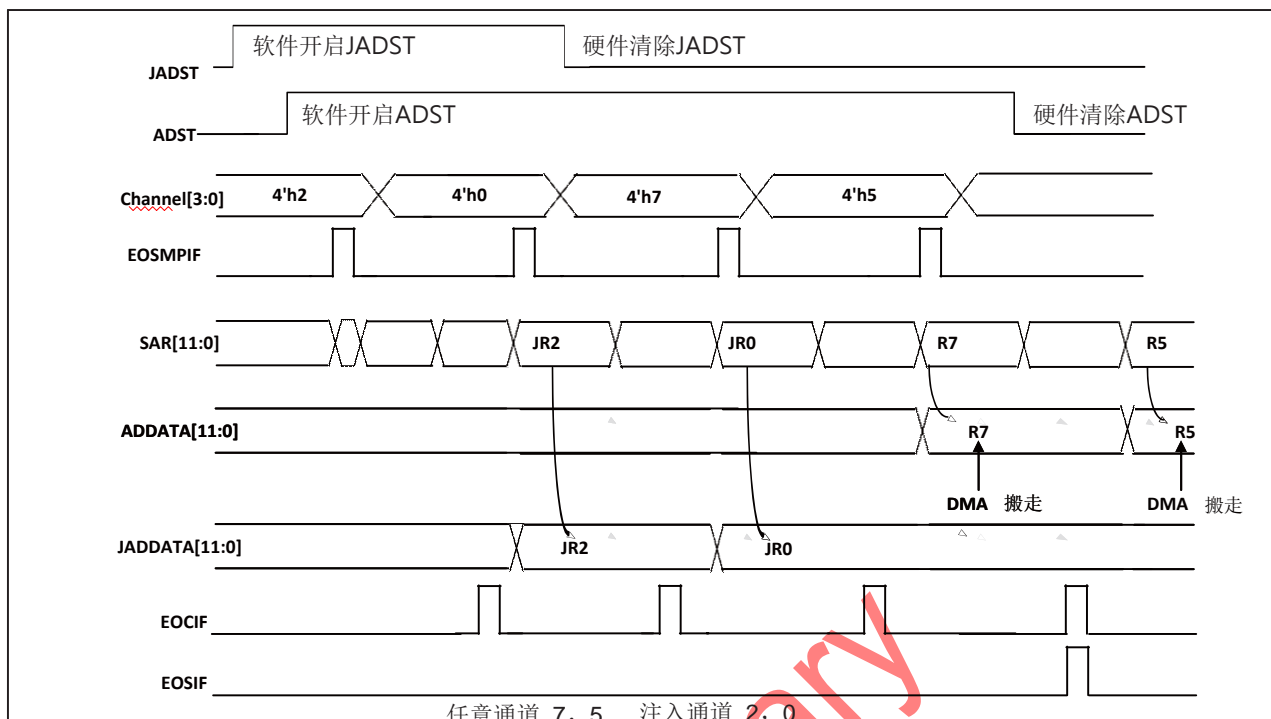


图 16-14 任意通道转换时事件注入通道转换时序图 1

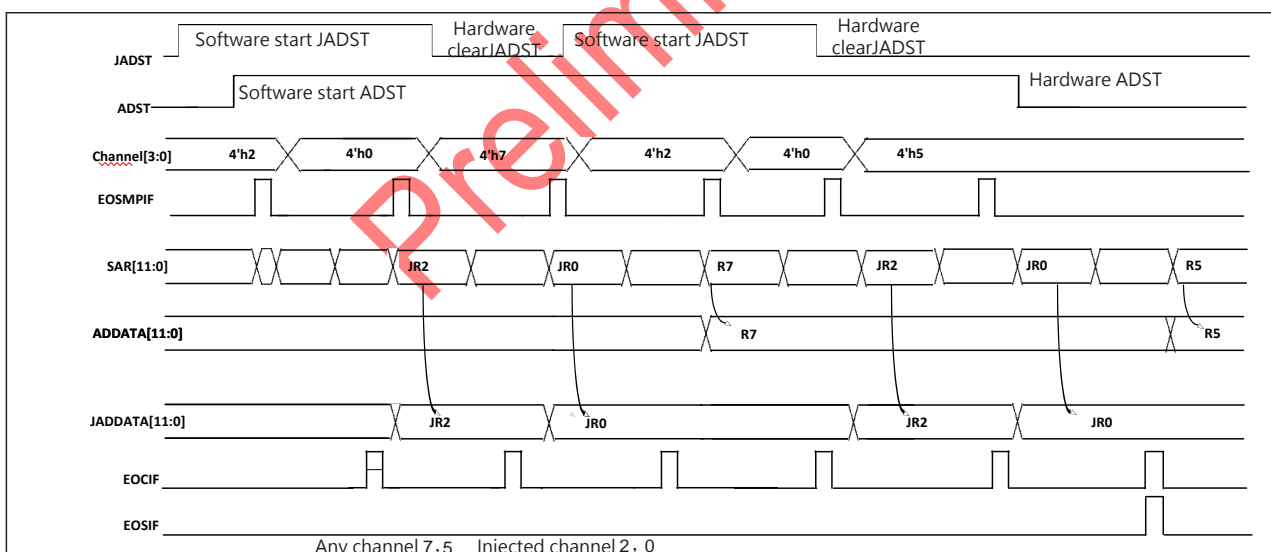


图 16-15 任意通道转换时事件注入通道转换时序图 2

16.7.4 ADC 触发信号

除了软件触发，ADC 转换的触发源还包括定时器和外部事件。

在触发信号产生后，延时 N（可配）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，其余通道采样是在上一个通道转换结束后就立即开始。

通过设置 ADC_ADCR.TRGEN 位可以使用外部事件触发非注入通道转换。

通过设置 ADC_ADCR.TRGSEL 位可以选择非注入通道转换外部触发源。

具体的外部触发源选择情况，可以参考 A/D 控制寄存器 ADC_ADCR.TRGSEL 位的描述。外部触发可设置延时控制，具体参考 AD 控制寄存器 ADC_ADCR.TRGSHIFT 相关位的描述。

16.7.5 模拟看门狗

看门狗比较发生在数据对齐操作前。看门狗比较器提供了上限和下限两个比较寄存器。可配置寄存器位 ADC_ADCR.CMPCH 位选择监控通道。当 ADC_ADCMPR.CPMHDATA 大于或等于 ADC_ADCMPR.CPMLDATA 时，若使能 ADC_ADCFG.AWDEN 非注入通道转换模拟看门狗使能且非注入通道结果大于或等于 ADC_ADCMPR.CMPHDATA 值或者小于 ADC_ADCMPR.CPMLDATA 值，状态寄存器 ADC_ADSTA.AWDIF 位置 1，使能 ADC_ADCFG.JAWDEN 注入通道转换模拟看门狗使能且注入通道结果大于或等于 ADC_ADCMPR.CMPHDATA 值或者小于 ADC_ADCMPR.CPMLDATA 值，状态寄存器 ADC_ADSTA.AWDIF 位也会置 1。

当 ADC_ADCMPR.CPMHDATA 小于 ADC_ADCMPR.CPMLDATA 时，若使能 ADC_ADCFG.AWDEN 非注入通道转换模拟看门狗使能且非注入通道结果等于 ADC_ADCMPR.CPMHDATA 值或者处于两个指定值之间，则状态寄存器 ADC_ADSTA.AWDIF 位会置 1。若使能 ADC_ADCFG.JAWDEN 注入通道转换模拟看门狗使能且注入通道结果等于 ADC_ADCMPR.CPMHDATA 值或者处于两个指定值之间，则状态寄存器 ADC_ADSTA.AWDIF 位也会置 1。

如果控制寄存器 ADC_ADCR.AWDIE 被置位，将产生中断请求。

16.7.6 内部温度传感器

内置的温度传感器仅用来检测器件内部的温度变化 (TA)。如果需要测量精确的温度，则需要使用外置的温度传感器。

通过设置 ADC_ADCFG.TSEN 位可以打开温度传感器，清除 TSEN 位可以关闭温度传感器。

通过设置 ADC_ADCHS.CHENTS 位选择温度传感器通道。

温度数值计算如下：

$$T (^{\circ}\text{C}) = (V_{\text{SENSE}} - V_{25}) / \text{Avg_Slope} + 25$$

V_{25} : 25°C 时的温度传感器转换值，存放在 flash 空间 0x1FFFF7F6。

V_{SENSE} : 温度传感器当前的输出电压

$V_{\text{SENSE}} = \text{Value} * V_{\text{dd}} / 4096$ (Value 是 ADC 的转换结果数据)

Avg_Slope: 温度与 V_{SENSE} 曲线的平均斜率 (以 mV/°C 或 $\mu\text{V}/^{\circ}\text{C}$ 表示)

V_{25} 和 Avg_Slope 的典型值请参考数据手册温度传感器章节。

16.7.7 内部电压传感器

ADC 的内部信号源通道连接了一个 1.2V 内部基准参考电压 V_{ref} ，此通道把 1.2V 的参考电压输出转换为数字值，通过计算获得内部参考电压值。

例如：12bitADC 的电压 V_{DD} 为 3.3V, V_{SENSOR} 通道转换值为 ADC_ADDR (V_{Sensor}) 计算公式如下：

$$3.3/4096 = V_{ref} / ADC_ADDR (V_{Sensor})$$

$$V_{ref} = (ADC_ADDR * 3.3) / 4096$$

内部参考电压在出厂时会被校准，得到的校准值存放在 flash 空间 $0x1FFFF7E0$ ，用户可以根据该校准值反向推出当前系统模拟参考电压值，计算公式如下：

$$V_{ref_calc} / ADC_ADDR (V_{Sensor}) = V_{DDA} / 4096$$

$$V_{DDA} = (V_{ref_calc} * 4096) / ADC_ADDR (V_{Sensor})$$

V_{ref_calc} : 内部参考电压出厂校准值，单位 mv

内部参考电压有单独的使能位，可通过设置寄存器 $ADC_ADCFG.VSEN$ 位开启或关闭。

16.8 寄存器

16.8.1 寄存器总览

表 16-3 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
0x0C	ADC_ADCHS	A/D 通道选择寄存器	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
0x18~0x54	ADC_ADDR 0~15	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
0x70	ADC_SMPR1	A/D 采样配置寄存器 1	0x00000000
0x74	ADC_SMPR2	A/D 采样配置寄存器 2	0x00000000
0x7C~0x88	ADC_JOFR0~3	A/D 注入通道数据补偿寄存器	0x00000000
0x8C	ADC_JSQR	A/D 注入通道序列寄存器	0x00000000
0x90	ADC_JADDATA	A/D 注入数据寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0xB0~0xBC	ADC_JDR0~3	A/D 注入通道数据寄存器	0x00000000
0xF0	ADC_LDATA	A/D 上一次转换数据寄存器	0x00000000

16.8.2 ADC_ADDATA 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_ADDATA 寄存器后该位自动清除。
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_ADDATA 寄存器后该位自动清除。

Bit	Field	Description
19: 16	CHANNELSEL	<p>该 4 位显示当前数据寄存器中的数据所对应的通道 (Channel Selection)</p> <p>0000 : 表示当前数据寄存器所属通道为通道 0</p> <p>0001 : 表示当前数据寄存器所属通道为通道 1</p> <p>0010 : 表示当前数据寄存器所属通道为通道 2</p> <p>0011 : 表示当前数据寄存器所属通道为通道 3</p> <p>0100 : 表示当前数据寄存器所属通道为通道 4</p> <p>0101 : 表示当前数据寄存器所属通道为通道 5</p> <p>0110 : 表示当前数据寄存器所属通道为通道 6</p> <p>0111 : 表示当前数据寄存器所属通道为通道 7</p> <p>1000 : 表示当前数据寄存器所属通道为通道 8</p> <p>1001 : 表示当前数据寄存器所属通道为通道 9</p> <p>1010 : 表示当前数据寄存器所属通道为通道 10</p> <p>1011 : 表示当前数据寄存器所属通道为通道 11</p> <p>1100 : 表示当前数据寄存器所属通道为通道 12</p> <p>1101 : 表示当前数据寄存器所属通道为通道 13</p> <p>1110 : 表示当前数据寄存器所属通道为通道 14 (对应数据为温度传感器的转换数据)</p> <p>1111 : 表示当前数据寄存器所属通道为通道 15 (对应数据为内部参考电压的转换数据)</p>
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。

16.8.3 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.															JAWDEN
															N
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ADCPREH	Res.				RSLTCTL			ADCPREH			VSEN	TSEN	AWDEN	ADEN
	EL					rw			rw			rw	rw	rw	rw
															rw

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值。
16	JAWDEN	A/D 注入通道转换结果狗使能 (Analog Watchdog Enable for Injected Channels) 1: 注入通道 A/D 模拟看门狗使能 0: 注入通道 A/D 模拟看门狗禁止
15	Reserved	保留, 必须保持复位值

Bit	Field	Description
14	ADCPREL	ADC 时钟预分频低位 (ADC Prescaler Low Bits) 预分频系数 $ADCPRE=\{ADCPREH, ADCPREL\}$
13: 10	Reserved	保留, 必须保持复位值
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: $div=(ADCPREH+2)$
3	VSEN	内部参考电压使能 (Voltage Sensor Enable) 1: 内部电压传感器使能 0: 内部电压传感器禁止
2	TSEN	温度传感器使能控制位 (Temperature Sensor Enable) 1: 温度传感器使能 0: 温度传感器禁止
1	AWDEN	非注入通道转换结果模拟看门狗使能 (Analog Watchdog Enable) 1: 非注入通道转换结果模拟看门狗使能 0: 非注入通道转换结果模拟看门狗禁止
0	ADEN	A/D 转换使能 (ADC Enable) 1: 使能 0: 禁止

16.8.4 ADC_ADCR 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.				EOCIE	EOSMPI E	TRG_EDGE		Res.		TRGSHIFT			TRGSELH		SCANDI R
				rw	rw	rw				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	EOSIE
rw				rw	rw		rw		rw			rw	rw	rw	rw

Bit	Field	Description
31: 28	Reserved	保留, 必须保持复位值

Bit	Field	Description
27	EOCIE	A/D 通道转换结束中断使能 (Interrupt Enable for End of Conversion) 1: 使能 A/D 单次转换结束中断 0: 禁止 A/D 单次转换结束中断 如果 EOCIF 置位, A/D 转换结束后产生中断请求。
26	EOSMPIE	A/D 通道采样结束中断使能 (Interrupt Enable for End of Sample) 1: 使能 A/D 采样结束中断 0: 禁止 A/D 采样结束中断 若 EOSMPIF 置位, 则在 A/D 采样结束后产生中断请求。
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: 保留
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。
18 : 17	TRGSELH	外部触发源选择高位 (External Trigger Sources Select For High Bits) TRGSEL={TRGSELH,TRGSELL}
16	SCANDIR	ADC 通道扫描方向 (ADC Scan Direction) 仅在普通通道转换有效 1: ADC 通道选择寄存器按从高到低的顺序扫描 0: ADC 通道选择寄存器按从低到高的顺序扫描

Bit	Field	Description
15 : 12	CMPCH	<p>模拟看门狗通道选择 (Compare Channel Selection For Analog Watchdog)</p> <p>0000 : 选择通道 0 转换结果作为比较对象</p> <p>0001 : 选择通道 1 转换结果作为比较对象</p> <p>0010 : 选择通道 2 转换结果作为比较对象</p> <p>0011 : 选择通道 3 转换结果作为比较对象</p> <p>0100 : 选择通道 4 转换结果作为比较对象</p> <p>0101 : 选择通道 5 转换结果作为比较对象</p> <p>0110 : 选择通道 6 转换结果作为比较对象</p> <p>0111 : 选择通道 7 转换结果作为比较对象</p> <p>1000 : 选择通道 8 转换结果作为比较对象</p> <p>1001 : 选择通道 9 转换结果作为比较对象</p> <p>1010 : 选择通道 10 转换结果作为比较对象</p> <p>1011 : 选择通道 11 转换结果作为比较对象</p> <p>1100 : 选择通道 12 转换结果作为比较对象</p> <p>1101 : 选择通道 13 转换结果作为比较对象</p> <p>1110 : 选择通道 14 转换结果作为比较对象 (即选择比较温度传感器的转换结果为比较对象)</p> <p>1111 : 选择通道 15 转换结果作为比较对象 (即选择所有扫描通道转换结果为比较对象)</p>
11	ALIGN	<p>数据对齐格式 (Data Alignment)</p> <p>1: 左对齐</p> <p>0: 右对齐</p>
10 : 9	ADMD	<p>A/D 转换模式 (ADC Mode)</p> <p>00: 单次转换</p> <p>01: 单周期扫描</p> <p>10: 连续扫描</p> <p>11: 保留</p> <p>当改变转换模式时, 软件要先清除 ADST 位。</p>
8	ADST	<p>A/D 转换开始 (ADC Start)</p> <p>1: 转换开始</p> <p>0: 转换结束或空闲状态</p> <p>ADST 清除有下列两种方式:</p> <p>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。</p> <p>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。</p>
7	Reserved	保留, 必须保持复位值。

Bit	Field	Description
6: 4	TRGSELL	外部触发源选择低位 (External Trigger sources Select for Low bits) TRGSEL={TRGSELH,TRGSELL} ADC 选择外部触发源 TRGSEL: 00000: TIM1_CC1 00001: TIM1_CC2 00010: TIM1_CC3 00011: TIM2_CC2 00100: TIM3_TRGO 00101: TIM1_CC4 和 TIM1_CC5 00110: TIM3_CC1 00111: EXTI 11 01000: TIM1_TRGO 01001: EXTI 4 01010: EXTI 5 01011: TIM2_CC1 01100: TIM3_CC4 01101: TIM2_TRGO 01111: EXTI 15 10000: TIM1_CC4 10001: TIM1_CC5 其它: 无效
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 模拟看门狗中断使能 (Interrupt Enable of Analog Watchdog) 1: 使能 A/D 模拟看门狗中断 0: 禁止 A/D 模拟看门狗中断
0	EOSIE	A/D 非注入通道转换的通道序列转换结束中断使能 (Interrupt Enable for End of Sequence) 1: 使能 A/D 中断 0: 禁止 A/D 中断 如果 EOSIF 置位, A/D 转换结束后会产生中断请求。

16.8.5 ADC_ADCHS 通道选择寄存器

偏移地址: 0x0C

复位值: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHEN15	CHEN14	CHEN13	CHEN12	CHEN11	CHEN10	CHEN9	CHEN8	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15	CHEN15	ADC: 内部参考电压使能 (Voltage Sensor Enable) 1: 使能 0: 禁止
14	CHEN14	ADC: 温度传感器使能 (Temperature Sensor Enable) 1: 使能 0: 禁止
13	CHEN13	模拟输入通道 13 使能 (Analog Input Channel 13 Enable) 1: 使能 0: 禁止
12	CHEN12	模拟输入通道 12 使能 (Analog Input Channel 12 Enable) 1: 使能 0: 禁止
11	CHEN11	模拟输入通道 11 使能 (Analog Input Channel 11 Enable) 1: 使能 0: 禁止
10	CHEN10	模拟输入通道 10 使能 (Analog Input Channel 10 Enable) 1: 使能 0: 禁止
9	CHEN9	模拟输入通道 9 使能 (Analog Input Channel 9 Enable) 1: 使能 0: 禁止
8	CHEN8	模拟输入通道 8 使能 (Analog Input Channel 8 Enable) 1: 使能 0: 禁止
7	CHEN7	模拟输入通道 7 使能 (Analog Input Channel 7 Enable) 1: 使能 0: 禁止
6	CHEN6	模拟输入通道 6 使能 (Analog Input Channel 6 Enable) 1: 使能 0: 禁止
5	CHEN5	模拟输入通道 5 使能 (Analog Input Channel 5 Enable) 1: 使能 0: 禁止
4	CHEN4	模拟输入通道 4 使能 (Analog Input Channel 4 Enable) 1: 使能 0: 禁止

Bit	Field	Description
3	CHEN3	模拟输入通道 3 使能 (Analog Input Channel 3 Enable) 1: 使能 0: 禁止
2	CHEN2	模拟输入通道 2 使能 (Analog Input Channel 2 Enable) 1: 使能 0: 禁止
1	CHEN1	模拟输入通道 1 使能 (Analog Input Channel 1 Enable) 1: 使能 0: 禁止
0	CHEN0	模拟输入通道 0 使能 (Analog Input Channel 0 Enable) 1: 使能 0: 禁止

注： 1) 如果通道使能都关闭，则开启转换时根据 SCANDIR 设置，默认转换通道为 0。

2) 在 ADC_ADSTA.BUSY 无效时才能操作此寄存器。

16.8.6 ADC_ADCMPR 模拟看门狗比较寄存器

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	模拟看门狗比较数值高阈值 (Compare High Threshold Data For Analog Watchdog) 看门狗比较阈值的高限
15: 12	Reserved	保留，必须保持复位值。
11 : 0	CMPLDATA	模拟看门狗比较数值低阈值 (Compare Low Threshold Data For Analog Watchdog) 看门狗比较阈值的低限

16.8.7 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
OVERRUN												VALID			
r												r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID								CHANNEL				Res.	BUSY	AWDIF	EOSIF
r								r					r	rc_w1	rc_w1

Bit	Field	Description
31: 20	OVERRUN	通道 0~11 的数据覆盖标志位 (Overrun Flag)
19: 8	VALID	通道 0~11 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	非注入通道转换忙/空闲 (Busy) 1= A/D 转换器忙碌 0= A/D 转换器空
1	AWDIF	模拟看门狗标志位 (Analog Watchdog Flag) 该标志位写“1”清零。 1: 模拟看门狗事件发生 0: 模拟看门狗事件未发生
0	EOSIF	A/D 通道序列转换结束标志位 (End of Sequential Conversion Flag) 该位由硬件在通道序列转换结束时置位, 由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。

16.8.8 ADC_ADDRn 通道数据寄存器 (n=0~15)

偏移地址: 0x18~0x54

复位值: 0x00000000

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

Bit	Field	Description
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。

16.8.9 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.										JBUSY	JEOSIF	JEOCIF	JEOSM PIF	EOCIF	EOSMPI F
										r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										OVERRUN			VALID		
										r			r		

Bit	Field	Description
31: 22	Reserved	保留，必须保持复位值。
21	JBUSY	注入通道转换忙/空闲（Busy/Idle Flag） 1: A/D 转换器注入通道忙碌 0: A/D 转换器注入通道空闲
20	JEOSIF	A/D 注入通道序列转换结束标志位（End of Injected Sequence Flag） 该位由硬件在通道序列转换结束时置位，由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。
19	JEOCIF	A/D 注入通道转换结束标志位（End of Injected Conversion Flag） 该位由硬件在通道转换结束时置位，由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。

Bit	Field	Description
18	JEOSMPIF	A/D 注入通道采样结束标志位 (End of Injected Sample Flag) 该位由硬件在通道采样结束时置位, 由软件清除。 1: A/D 采样结束 0: A/D 采样未结束 该标志位写“1”清零。
17	EOCIF	A/D 转换结束标志位 (End of Conversion Flag) 该位由硬件在通道转换结束时置位, 由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。
16	EOSMPIF	A/D 采样结束标志位 (End of Sample Flag) 该位由硬件在通道采样结束时置位, 由软件清除。 1: A/D 采样转换结束 0: A/D 采样未结束 该标志位写“1”清零。
15: 8	Reserved	保留, 必须保持复位值。
7: 4	OVERRUN	通道的数据覆盖标志位 (Overrun Flag) 1000: 内部参考电压 (V_SENSOR) 0100: 温度传感器 (T_SENSOR) 0010: 通道 13 0001: 通道 12
3: 0	VALID	通道有效标志位 (Valid Flag) 1000: 内部参考电压 (V_SENSOR) 0100: 温度传感器 (T_SENSOR) 0010: 通道 13 0001: 通道 12

16.8.10 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	任意通道转换序列第 7 次转换选择 (7th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
27: 24	CHANY_SEL6	任意通道转换序列第 6 次转换选择 (6th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
23: 20	CHANY_SEL5	任意通道转换序列第 5 次转换选择 (5th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
19: 16	CHANY_SEL4	任意通道转换序列第 4 次转换选择 (4th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
15: 12	CHANY_SEL3	任意通道转换序列第 3 次转换选择 (3th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
11: 8	CHANY_SEL2	任意通道转换序列第 2 次转换选择 (2th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
7: 4	CHANY_SEL1	任意通道转换序列第 1 次转换选择 (1th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
3: 0	CHANY_SEL0	任意通道转换序列第 0 次转换选择 (0th Conversion Select for Any Channel sequence) 0000~1111: 可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道

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。

16.8.11 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				CHANY_SEL14				CHANY_SEL13				CHANY_SEL12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL11				CHANY_SEL10				CHANY_SEL9				CHANY_SEL8			
rw				rw				rw				rw			

Bit	Field	Description
31: 28	CHANY_SEL15	任意通道转换序列第 15 次转换选择（15th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
27: 24	CHANY_SEL14	任意通道转换序列第 14 次转换选择（14th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
23: 20	CHANY_SEL13	任意通道转换序列第 13 次转换选择（13th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
19: 16	CHANY_SEL12	任意通道转换序列第 12 次转换选择（12th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
15: 12	CHANY_SEL11	任意通道转换序列第 11 次转换选择（11th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
11: 8	CHANY_SEL10	任意通道转换序列第 10 次转换选择（10th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
7: 4	CHANY_SEL9	任意通道转换序列第 9 次转换选择（9th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道
3: 0	CHANY_SEL8	任意通道转换序列第 8 次转换选择（8th Conversion Select for Any Channel sequence） 0000~1111：可配置为通道 0~通道 15 中的任意一个通道为任意通道转换的输入通道

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。

16.8.12 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
3: 0	CHANY_NUM	转换通道数配置 (Number of Any Channel Mode): 0000: CHANY_SEL0 通道 0001: CHANY_SEL0~CHANY_SEL1 通道 0010: CHANY_SEL0~CHANY_SEL2 通道 0011: CHANY_SEL0~CHANY_SEL3 通道 0100: CHANY_SEL0~CHANY_SEL4 通道 0101: CHANY_SEL0~CHANY_SEL5 通道 0110: CHANY_SEL0~CHANY_SEL6 通道 0111: CHANY_SEL 0~CHANY_SEL7 通道 1000: CHANY_SEL0~CHANY_SEL8 通道 1001: CHANY_SEL0~CHANY_SEL9 通道 1010: CHANY_SEL0~CHANY_SEL10 通道 1011: CHANY_SEL0~CHANY_SEL11 通道 1100: CHANY_SEL0~CHANY_SEL12 通道 1101: CHANY_SE 0~CHANY_SEL13 通道 1110: CHANY_SEL0~CHANY_SEL14 通道 1111: CHANY_SEL0~CHANY_SEL15 通道

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。

16.8.13 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.													JTRG_EDGE		
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTRGSHIFT			JTRGSEL					JTRGE N	JADST	JAUTO	JEOSIE	JEOCIE	JEOSM PIE	JCEN	CHANY _MDEN
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: 16	JTRG_EDGE	注入通道转换的外部触发源边沿选择（Trigger sources Edge Selection for Injected Channels） 00: 双沿触发 01: 下沿触发 10: 上沿触发 11: 屏蔽触发
15: 13	JTRGSHIFT	注入通道转换的外部触发源延时采样（External Trigger Shift Sample for Injected Channels） 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，其它通道是在上一个采样结束后立即开始。 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

Bit	Field	Description
12: 8	JTRGSEL	<p>ADC 注入通道转换的外部触发源选择 (External Trigger Sources Selection For Injected Channel)</p> <p>00000: TIM1_CC1 00001: TIM1_CC2 00010: TIM1_CC3 00011: TIM2_CC2 00100: TIM3_TRGO 00101: TIM1_CC4 和 TIM1_CC5 00110: TIM3_CC1 00111: EXTI 11 01000: TIM1_TRGO 01001: EXTI 4 01010: EXTI 5 01011: TIM2_CC1 01100: TIM3_CC4 01101: TIM2_TRGO 01111: EXTI 15 10000: TIM1_CC4 10001: TIM1_CC5 其它: 无效</p>
7	JTRGEN	<p>注入通道转换的外部硬件触发源使能 (External Trigger Enable For Injected Channels)</p> <p>1: 使能外部触发信号启动 A/D 转换 0: 禁止外部触发信号启动 A/D 转换</p>
6	JADST	<p>注入通道转换的 A/D 转换开始 (ADC Start for Injected Channels): 先开启 JCEN</p> <p>1: 注入通道转换开始 0: 注入通道转换结束或进入空闲状态</p> <p>JADST 清除有下列两种方式: 在注入通道转换完成后, JADST 将被硬件自动清除; JCEN 为 0 时 JADST 被清除; 系统复位时 JADST 被清除。</p>
5	JAUTO	<p>自动注入转换 (Automatic Injected Conversion)</p> <p>1: 开启自动注入转换 0: 关闭自动注入转换</p>
4	JEOSIE	<p>A/D 注入通道序列转换结束中断使能 (Interrupt Enable for JEOS)</p> <p>1: 使能 A/D 序列转换结束中断 0: 禁止 A/D 序列转换结束中断</p> <p>如果 JEOSIE 置位, 通道序列 A/D 转换结束后产生中断请求。</p>
3	JEOCIE	<p>A/D 注入通道转换结束中断使能 (Interrupt Enable for JEOC)</p> <p>1: 使能 A/D 转换结束中断 0: 禁止 A/D 转换结束中断</p> <p>如果 JEOCIE 置位, A/D 转换结束后产生中断请求。</p>

Bit	Field	Description
2	JEOSMPIE	A/D 注入通道采样结束中断使能 (Interrupt Enable for JEOSAMP) 1: 使能 A/D 采样结束中断 0: 禁止 A/D 采样结束中断 如果 JEOSMPIF 置位, A/D 转换结束后产生中断请求。
1	JCEN	A/D 注入通道转换使能 (Conversion Enable for Injected Channels) 1: 使能注入转换 0: 禁止注入转换
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

注: 任意通道转换的单周期或连续扫描模式时, 关闭 ADC 必须先等待 ADC_ADCR.ADST 和 ADC_ADSTA.BUSY 为 0, 再清除 ADC_ANY_CR.CHANY_MDEN 位。

16.8.14 ADC_SMPR1 采样配置寄存器 1

偏移地址: 0x70

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP7				SAMP6				SAMP5				SAMP4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP3				SAMP2				SAMP1				SAMP0			
rw				rw				rw				rw			

Bit	Field	Description
31 : 0	SAMP7~ SAMP0	<p>选择通道 0~7 的采样时间选择 (Channel Sample Time Selection)</p> <p>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。</p> <p>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 其它: 保留</p>

16.8.15 ADC_SMPR2 采样配置寄存器 2

偏移地址: 0x74

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP15				SAMP14				SAMP13				SAMP12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP11				SAMP10				SAMP9				SAMP8			
rw				rw				rw				rw			

Bit	Field	Description
31 : 0	SAMP15~ SAMP8	选择通道 8~15 的采样时间选择 (Channel 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。 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 其它: 保留

16.8.16 ADC_JOFRn 注入通道数据补偿寄存器 (n=0~3)

偏移地址: 0x7C~0x88

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.				JOFFSET[11: 0]											
				rw											

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	JOFFSET	12 位 A/D 注入通道 n 的转换结果补偿寄存器 (Data Offset Register for Injected Channel) 注入通道 n 的转换结果减去 JOFFSET 补偿值后, 注入通道转换补偿后的结果保存在寄存器 JADDATA 和 ADC_JDRn. 注: 该位在 JADST=0 时允许进行写操作

16.8.17 ADC_JSQR 注入通道序列寄存器

偏移地址：0x8C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										JNUM		JSQ3			
										rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ3.		JSQ2				JSQ1				JSQ0					
rw		rw				rw				rw					

Bit	Field	Description
23: 22	Reserved	保留，必须保持复位值。
21: 20	JNUM	注入通道转换的通道数配置（Channel Number for Injected Sequence） 00: JSQ0 通道 01: JSQ0~JSQ1 通道 10: JSQ0~JSQ2 通道 11: JSQ0~JSQ3 通道
19: 15	JSQ3	注入通道序列第 3 次转换（3th Conversion for Injected Sequence） 00000~01111: 可配置为通道 0~通道 15 中的任意通道为注入通道 其它: 保留
14: 10	JSQ2	注入通道序列第 2 次转换（2th Conversion for Injected Sequence） 00000~01111: 可配置为通道 0~通道 15 中的任意通道为注入通道 其它: 保留
9: 5	JSQ1	注入通道序列第 1 次转换（1th Conversion for Injected Sequence） 00000~01111: 可配置为通道 0~通道 15 中的任意通道为注入通道。 其它: 保留
4: 0	JSQ0	注入通道序列第 0 次转换（0th Conversion for Injected Sequence） 00000~01111: 可配置为通道 0~通道 15 中的任意通道为注入通道。 其它: 保留

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_JSQR 影子寄存器。在 ADC 工作期间，若更改 ADC_JSQR 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_JSQR，实现通道动态切换。

16.8.18 ADC_JADDATA 注入数据寄存器

偏移地址：0x90

复位值：0x00000000

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

Res.									JVALID	JOVER RUN	Res.	JCHANNELSEL			
									r	r		r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
r															

Bit	Field	Description
31: 23	Reserved	保留，必须保持复位值
22	JVALID	注入通道有效标志位（只读）（Valid Flag for Injected Channels） 1: JDATA[11: 0] 位数据有效 0: JDATA[11: 0] 位数据无效 对应通道转换完成后硬件置位，读 ADC_JADDATA 寄存器自动清除。
21	JOVERRUN	注入通道数据覆盖标志位（只读）（Overrun Flag for Injected Channels） 1: JDATA[11: 0] 数据被覆盖 0: JDATA[11: 0] 数据最近一次转换结果。 新的转换结果装载至寄存器之前，若 JDATA[11: 0] 的数据没有被读取，JOVERRUN 将置位；读 ADC_JADDATA 寄存器硬件自动清除。
20	Reserved	保留，必须保持复位值
19: 16	JCHANNELSEL	该 4 位显示当前数据所对应的注入通道（Injected Channel Selection） 0000 = 通道 0 的转换数据 0001 = 通道 1 的转换数据 0010 = 通道 2 的转换数据 0011 = 通道 3 的转换数据 0100 = 通道 4 的转换数据 0101 = 通道 5 的转换数据 0110 = 通道 6 的转换数据 0111 = 通道 7 的转换数据 1000 = 通道 8 的转换数据 1001 = 通道 9 的转换数据 1010 = 通道 10 的转换数据 1011 = 通道 11 的转换数据 1100 = 通道 12 的转换数据 1101 = 通道 13 的转换数据 1110 = 温度传感器的转换数据 1111 = 内部参考电压的转换数据 其它：无效
15: 0	JDATA	12 位 A/D 当前注入通道转换结果（Transfer Data for Injected Channels） 软件选择对齐方式。

16.8.19 ADC_JDRn 注入通道数据寄存器（n=0~3）

偏移地址：0xB0~0xBC

复位值：0x00000000

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

Bit	Field	Description
31: 23	Reserved	保留，必须保持复位值
22	JVALID	注入通道有效标志位（Valid Flag for Injected Channels） 1: JDATA[11: 0]位数据有效 0: JDATA[11: 0]位数据无效 对应通道转换完成后硬件置位，读 ADC_JADR 寄存器自动清除
21	JOVERRUN	注入通道数据覆盖标志位（Overrun Flag for Injected Channels） 1: JDATA [11: 0]数据被覆盖 0: JDATA [11: 0]数据最近一次转换结果 新的转换结果装载至寄存器之前，若 JDATA[11: 0]的数据没有被读取，JOVERRUN 将置位，读 JDRn 寄存器硬件自动清除。
20: 16	Reserved	保留，必须保持复位值。
15: 0	JDATA	A/D 注入通道 n 的转换结果（Transfer Data for Injected Channels） 软件可选择对齐方式，参考数据对齐章节。

16.8.20 ADC_LDATA 上一次转换数据寄存器

偏移地址：0xF0

复位值：0x00000000

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

Bit	Field	Description
31: 22	Reserved	保留，必须保持复位值。

Bit	Field	Description
21	LVALID	数据有效标志位（只读）（Valid Flag） 1: LCHANNELSEL 和 LDATA[11: 0] 位数据有效 0: LCHANNELSEL 和 LDATA[11: 0] 位数据无效 通道转换完成后硬件置位，读 ADC_LDATA 寄存器后该位自动清除。
20	LOVERRUN	数据覆盖标志位（只读）（Overrun Flag） 1: LDATA[11: 0] 上一次转换数据被覆盖 0: LDATA[11: 0] 上一次转换数据未被覆盖 若数据装载至 LDATA[11: 0] 寄存器之前，上一次 LDATA[11: 0] 没有被读取，则 LOVERRUN 将置位，读 ADC_LDATA 寄存器后该位自动清除。
19: 16	LCHANNELSEL	该 4 位显示上一次数据寄存器中的数据所对应的通道（Last Channel Selection） 0000：表示上一次数据寄存器所属通道为通道 0 0001：表示上一次数据寄存器所属通道为通道 1 0010：表示上一次数据寄存器所属通道为通道 2 0011：表示上一次数据寄存器所属通道为通道 3 0100：表示上一次数据寄存器所属通道为通道 4 0101：表示上一次数据寄存器所属通道为通道 5 0110：表示上一次数据寄存器所属通道为通道 6 0111：表示上一次数据寄存器所属通道为通道 7 1000：表示上一次数据寄存器所属通道为通道 8 1001：表示上一次数据寄存器所属通道为通道 9 1010：表示上一次数据寄存器所属通道为通道 10 1011：表示上一次数据寄存器所属通道为通道 11 1100：表示上一次数据寄存器所属通道为通道 12 1101：表示上一次数据寄存器所属通道为通道 13 1110：表示上一次数据寄存器所属通道为通道 14 1111：表示上一次数据寄存器所属通道为通道 15
15: 0	LDATA	12 位 A/D 上一次通道转换结果（Last Conversion Data）

17 TIM1 高级定时器

17.1 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，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、死区时间可编程的互补输出、单脉冲模式输出等）。

17.2 功能框图

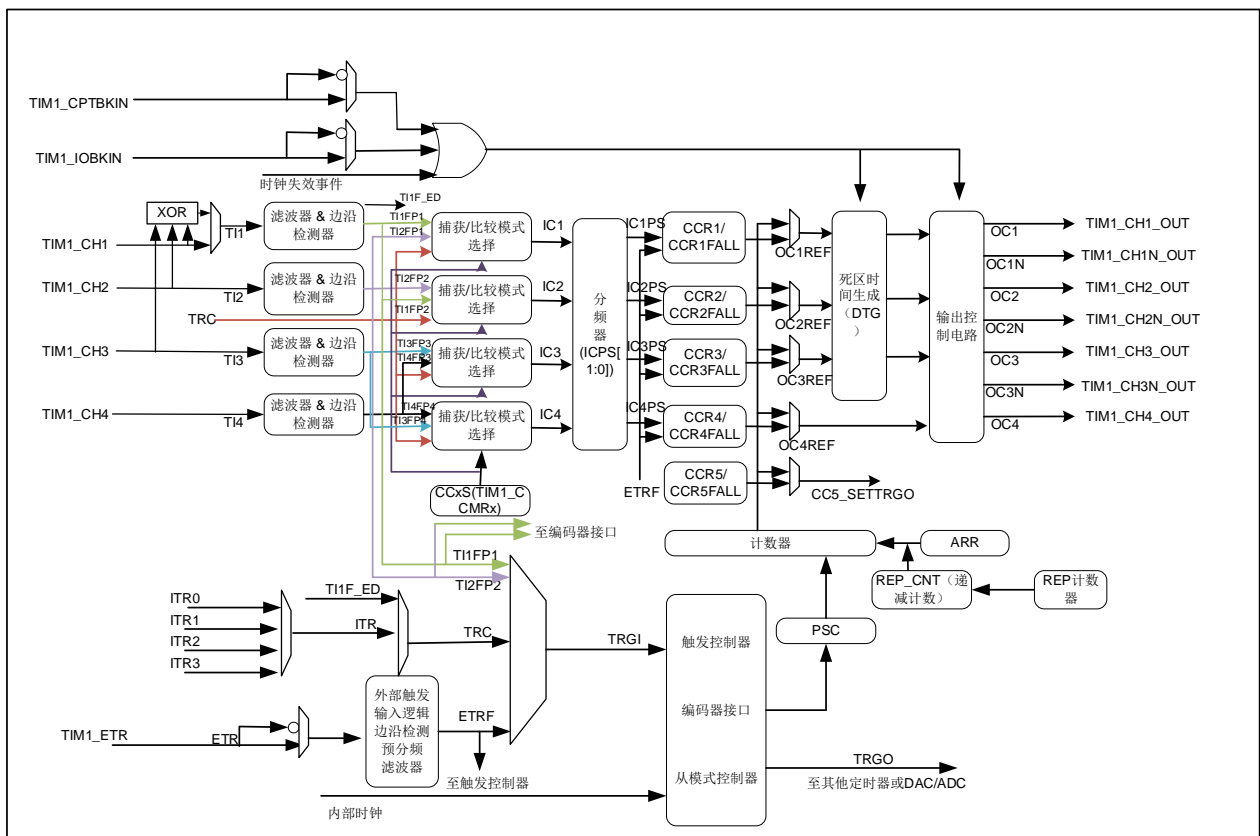


图 17-1 TIM1 结构图

上图为 TIM1 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。

17.3 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）
- 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
- 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，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、输入捕获、比较输出或者刹车输入

17.4 功能描述

17.4.1 时钟

17.4.1.1 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：

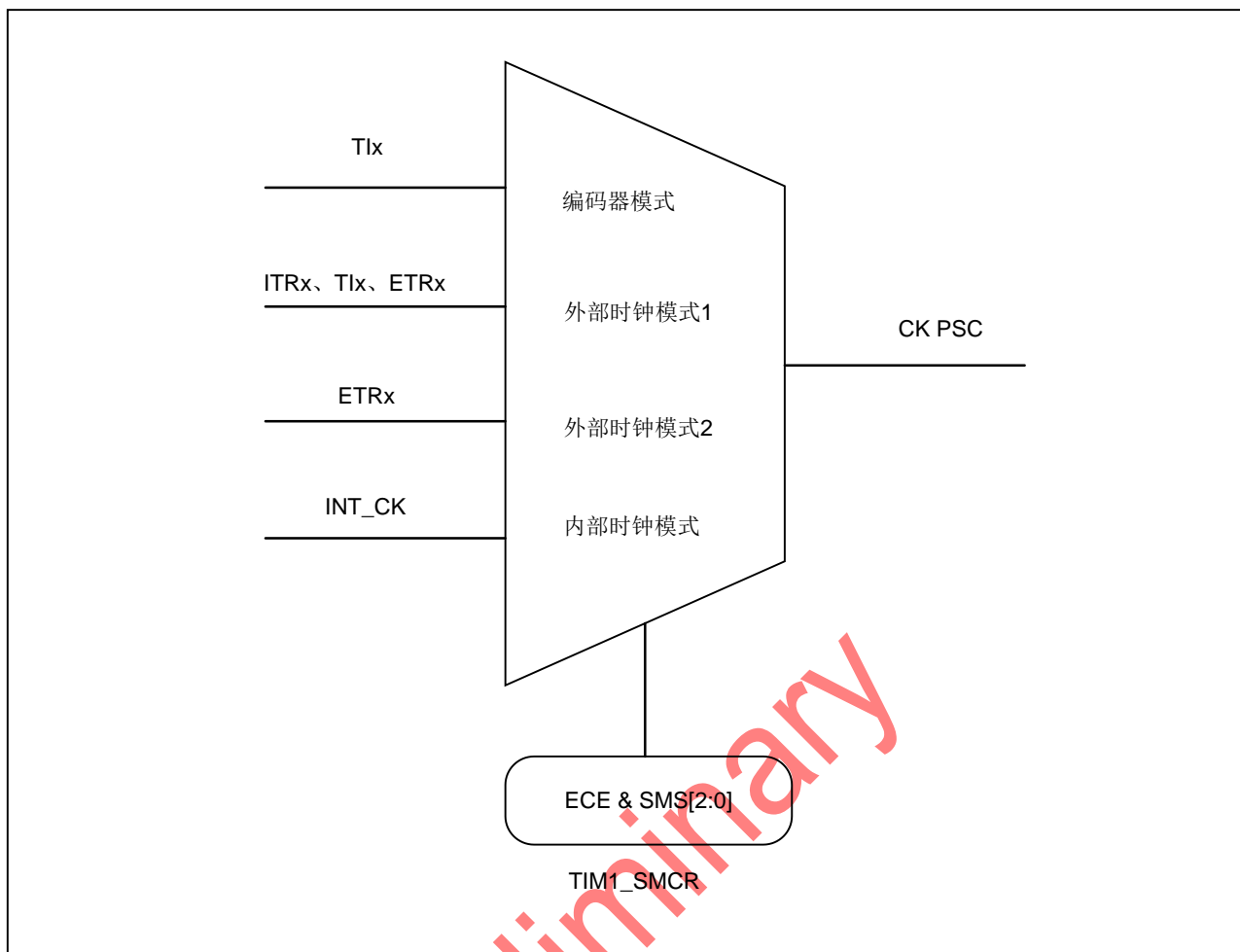


图 17-2 时钟选择

17.4.1.1.1 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。

17.4.1.1.2 外部时钟模式 1 (外部触发输入 TRGI, 包含 Tlx、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 TI1 输入端的上升沿递增计数，具体配置如下：

1. 配置 TIM1_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上；配置 TIM1_CCMR1 寄存器的 IC1F[3: 0]，设置输入滤波器带宽；配置 TIM1_CCER 寄存器 CC1P=0，选择上升沿为有效沿。
2. 配置 TIM1_SMCR 寄存器的 TS=101，选择 TI1 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。

3. 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 TI1 出现有效边沿时, 计数器递增计数一次。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。

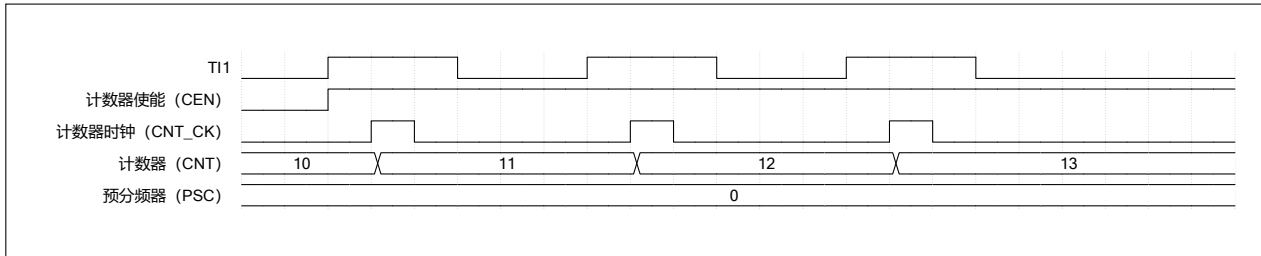


图 17-3 外部时钟模式 1 下的控制电路

17.4.1.1.3 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。

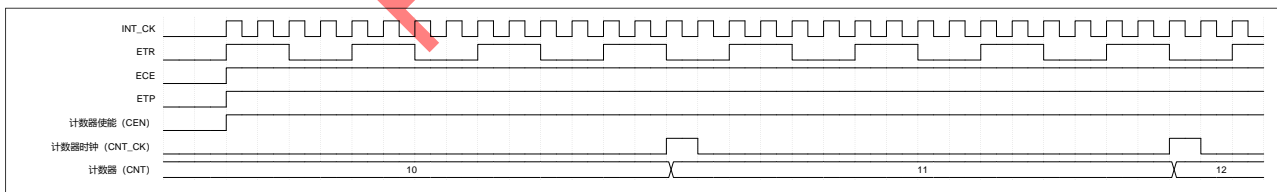


图 17-4 外部时钟模式 2 下的控制电路

17.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

17.4.1.2 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。

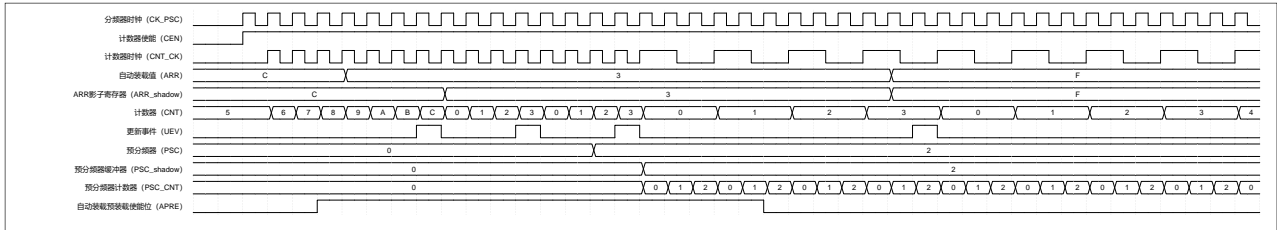


图 17-5 自动预装载

17.4.1.3 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。

17.4.1.3.1 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。

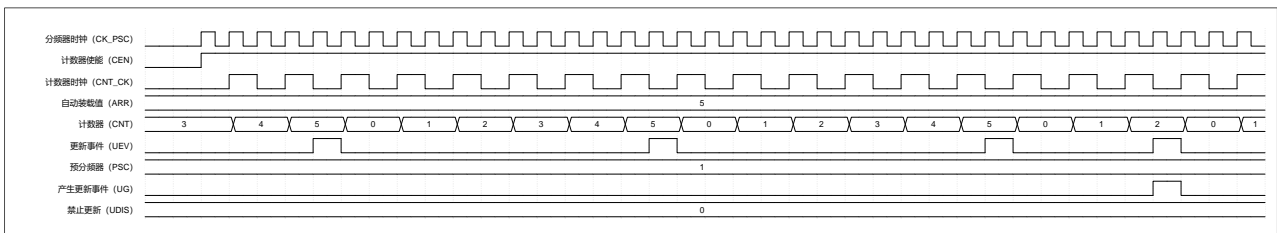


图 17-6 递增计数模式（UDIS=0）

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。

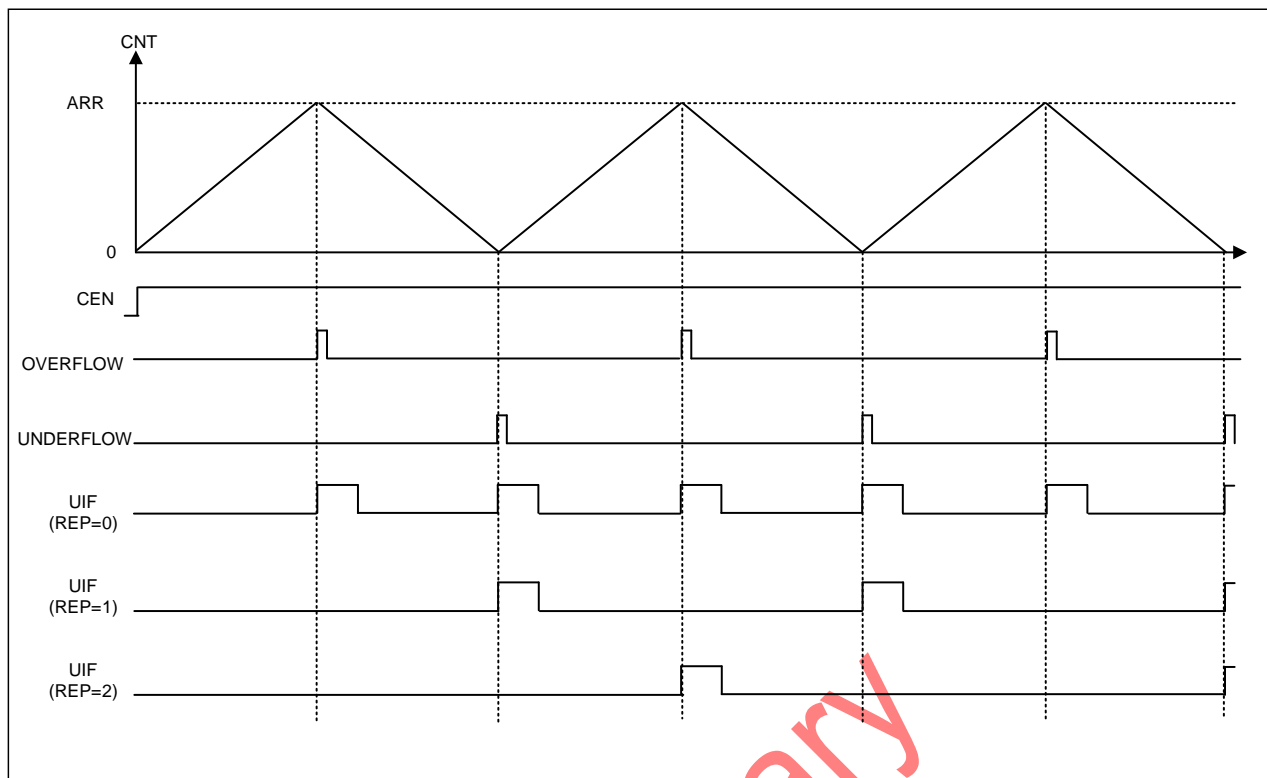


图 17-12 中央对齐模式重复计数时序图

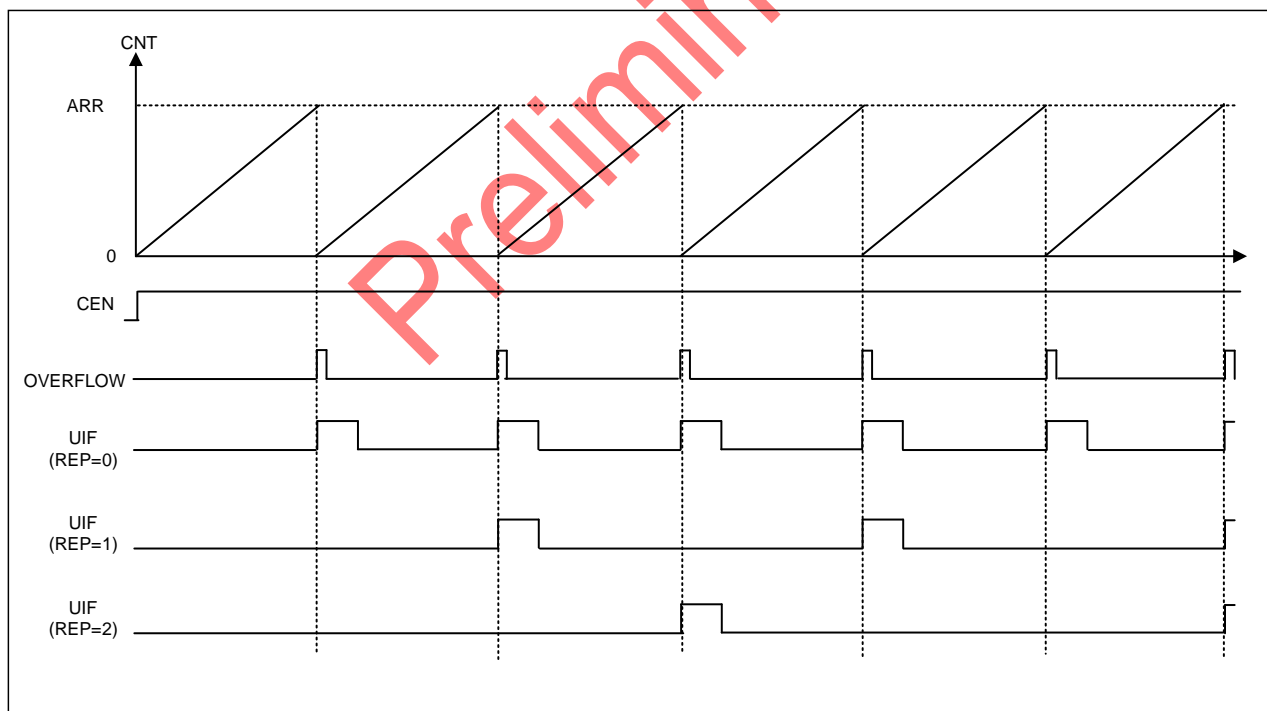


图 17-13 边沿对齐模式递增计数时序图

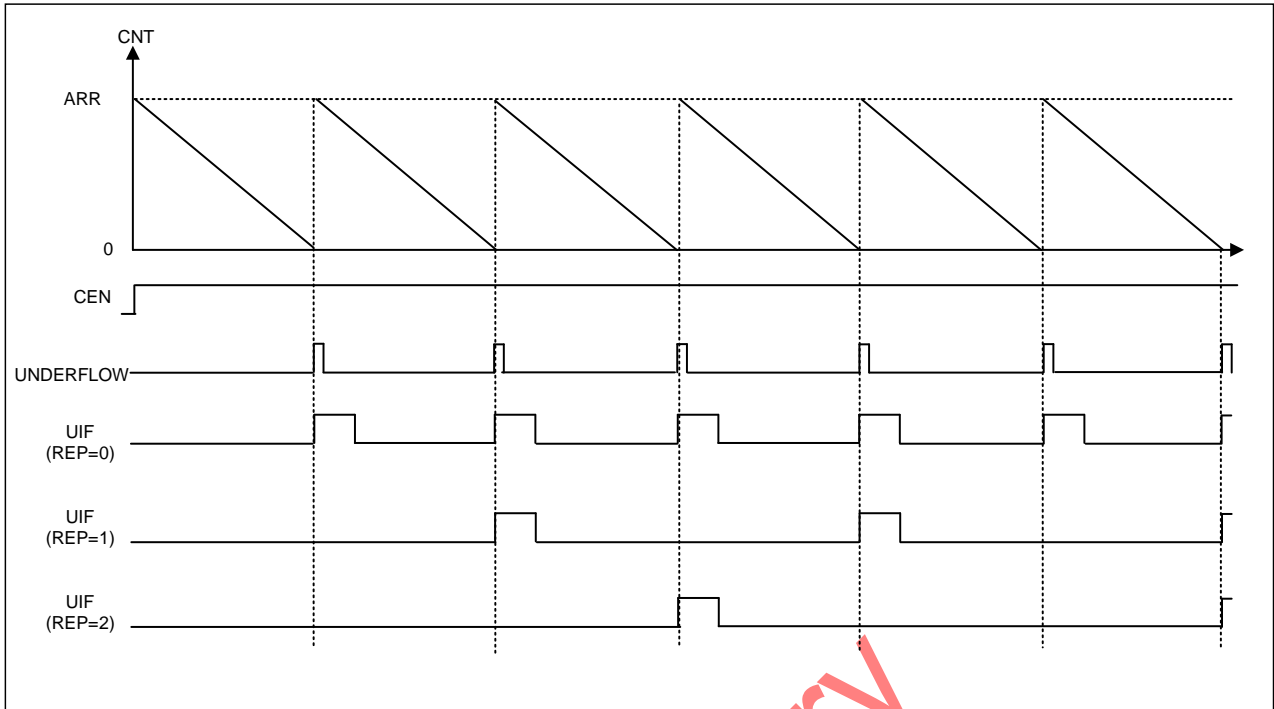


图 17-14 边沿对齐模式递减计数时序图

17.4.3 输入捕获

17.4.3.1 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：

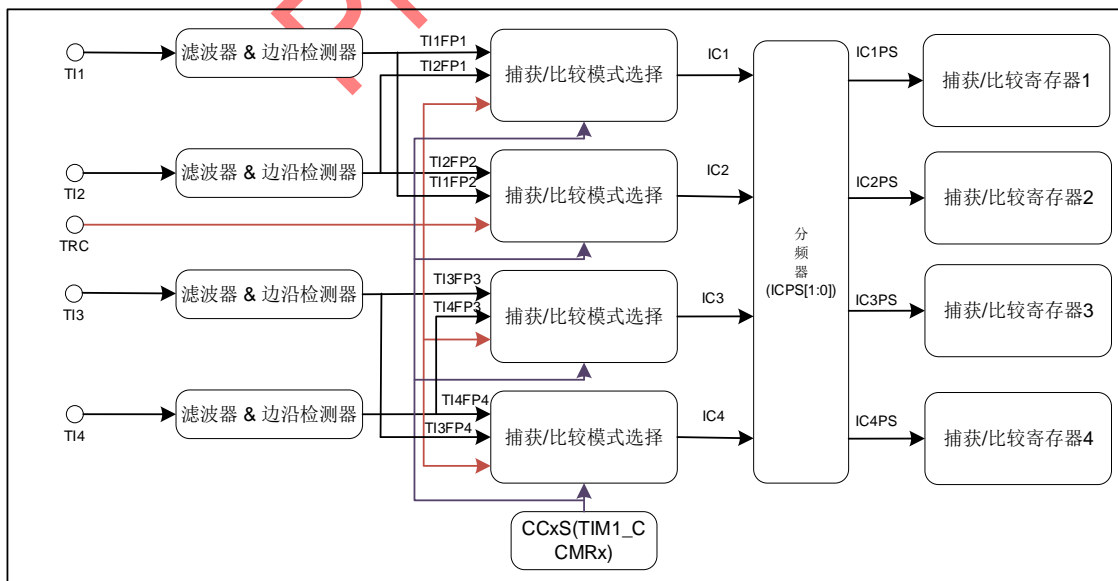


图 17-15 TIM1 输入捕获结构图

通过配置 TIM1_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 $T1x$ 的输入信号采样后，产生一个滤波后的信号 $T1xF$ ，然后通过极性可选的边沿检测器，产生一个有效信号 $T1xFPx$ ，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 17-1 数字滤波器宽度与 $ICxIF$ 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 ($TIM1_SR$) 中的捕获标志位 $CCxIF$ 置 1，通过配置 $CCxIF=0$ 或读取 $TIM1_CCRx$ 中的数据，清除 $CCxIF$ 标志位。当 $CCxIF$ 未被清零时，发生输入捕获事件，重复捕获标志位 $CCxOF$ 将会被置 1，通过配置 $CCxOF=0$ ，可以清除 $CCxOF$ 标志位。

例如，通过采样 $T11$ 输入信号的有效沿，在 $T11$ 的上升沿来到时捕获当前计数器的值，锁存到 $TIM1_CCR1$ 寄存器中，步骤如下：

1. 配置 $TIM1_CCMR1$ 寄存器的 $CC1S=01$ ， $CC1$ 通道被配置为输入， $IC1$ 映射在 $T11$ 上。
2. 配置 $TIM1_CCMR1$ 寄存器的 $IC1F[3: 0]$ ，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 $TIM1_CCER$ 寄存器的 $CC1P=0$ ，选择捕获发生在 $T11$ 信号的上升沿。
4. 配置 $TIM1_CCMR1$ 寄存器的 $IC1PSC[1: 0]$ ，选择预分频系数。
5. 配置 $TIM1_CCER$ 寄存器的 $CC1E = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 $TIM1_DIER$ 寄存器的 $CC1IE=1$ ，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 DMA，配置 $TIM1_DIER$ 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， $TIM1_CCRx$ 寄存器属性变为只读。
- 如果发生了两次以上连续捕获，但 $CCxIF$ 标志未被清零，则重复捕获标志 $CCxOF$ 被置 1。为了避免丢失重复捕获标志 $CCxOF$ 置 1 之前可能产生的捕获信息，建议在读出重复捕获标志之前读取数据。
- 设置 $TIM1_EGR$ 寄存器中相应的 $CCxG$ 位，可以通过软件产生输入捕获中断或 DMA 请求。

17.4.3.2 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 Tix 输入。
- 配置从模式为复位模式，将其中一路 TixFP 作为触发输入信号。

例：测量 TI1 的 PWM 信号的宽度（TIM1_CCR1 寄存器）和占空比（TIM1_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 TIM1_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 TIM1_CCMR1 寄存器的 CC1S = 01，将 IC1 映射在 TI1 上，选择 TIM1_CCR1 的有效输入。
3. 配置 TIM1_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 TIM1_CCR1 中并清除计数器）。
4. 配置 TIM1_CCMR1 寄存器的 CC2S = 10，将 IC2 映射在 TI1 上，选择 TIM1_CCR2 的有效输入。
5. 配置 TIM1_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 TIM1_CCR2 中）。
6. 配置 TIM1_SMCR 寄存器中的 TS = 101，选择 TI1FP1 为有效的触发输入信号。
7. 配置 TIM1_SMCR 中的 SMS = 100，从模式控制器设置为复位模式。
8. 配置 TIM1_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 17-16 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 TIM1_CH1/TIM1_CH2 端口输入信号。

17.4.4 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：

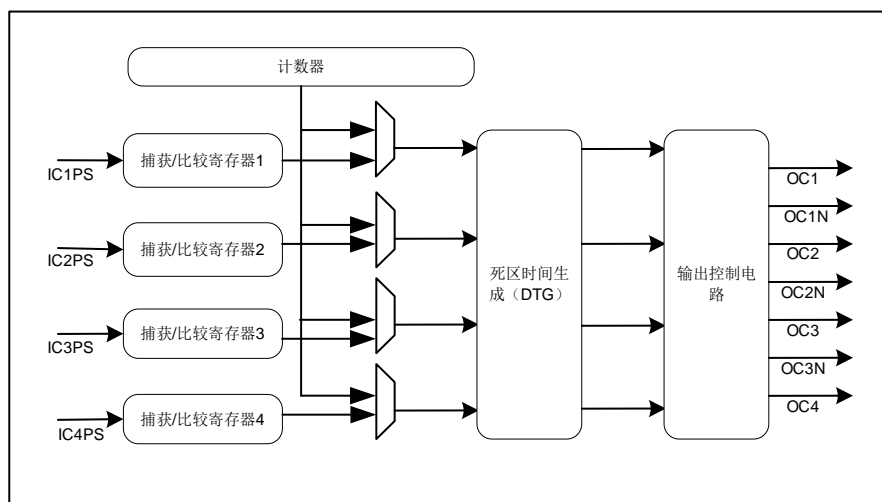


图 17-17 比较输出部分结构图

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。

17.4.4.1 强制输出

配置 TIM1_CCMRx 寄存器的 CCxS = 00，将通道 CCx 设置为输出模式，通过配置 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。

17.4.4.2 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 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. 匹配时中断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，使能捕获/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。

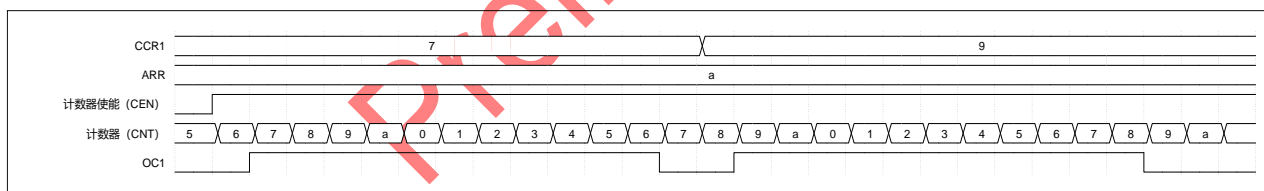


图 17-18 比较输出模式，OC1 信号在匹配时翻转

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。

17.4.4.3 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、OSSI、OSSR 位控制 OCx 的输出使能。配置 TIM1_CR1 寄存器的 CMS 位，可以选择产生边沿对齐或中央对齐的 PWM 信号。

- 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。

17.4.4.3.1 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 TIM1_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，选择 PWM 模式 1，当 TIM1_CNT < TIM1_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装载值 (TIM1_ARR)，则 OCxREF 保持为有效电平。如果比较值为 0，则 OCxREF 保持为无效电平。下图为 CCR1=1，CCR2=4，CCR3=7，CCR4=b，ARR=a 时边沿对齐递增计数时 PWM 模式 1 的波形实例。

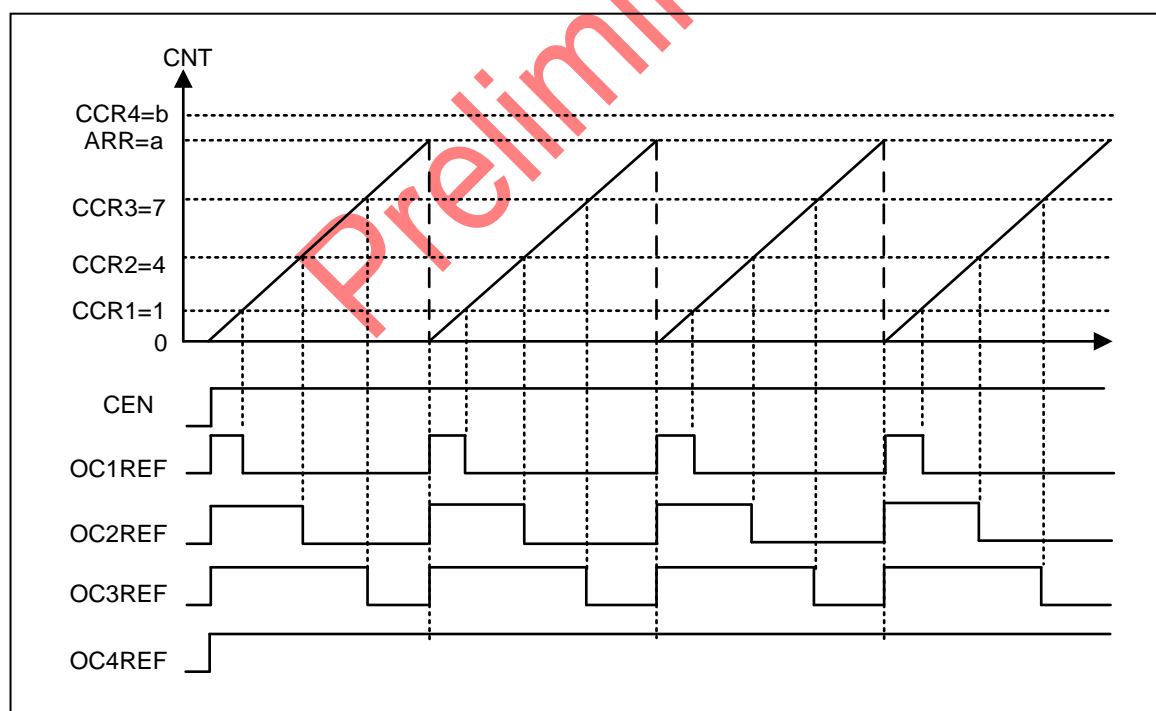


图 17-19 边沿对齐递增计数时 PWM 模式 1 的波形

17.4.4.3.2 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 TIM1_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，

选择 PWM 模式 1，当 $TIM1_CNT > TIM1_CCR_x$ 时通道 x (OC_xREF) 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。

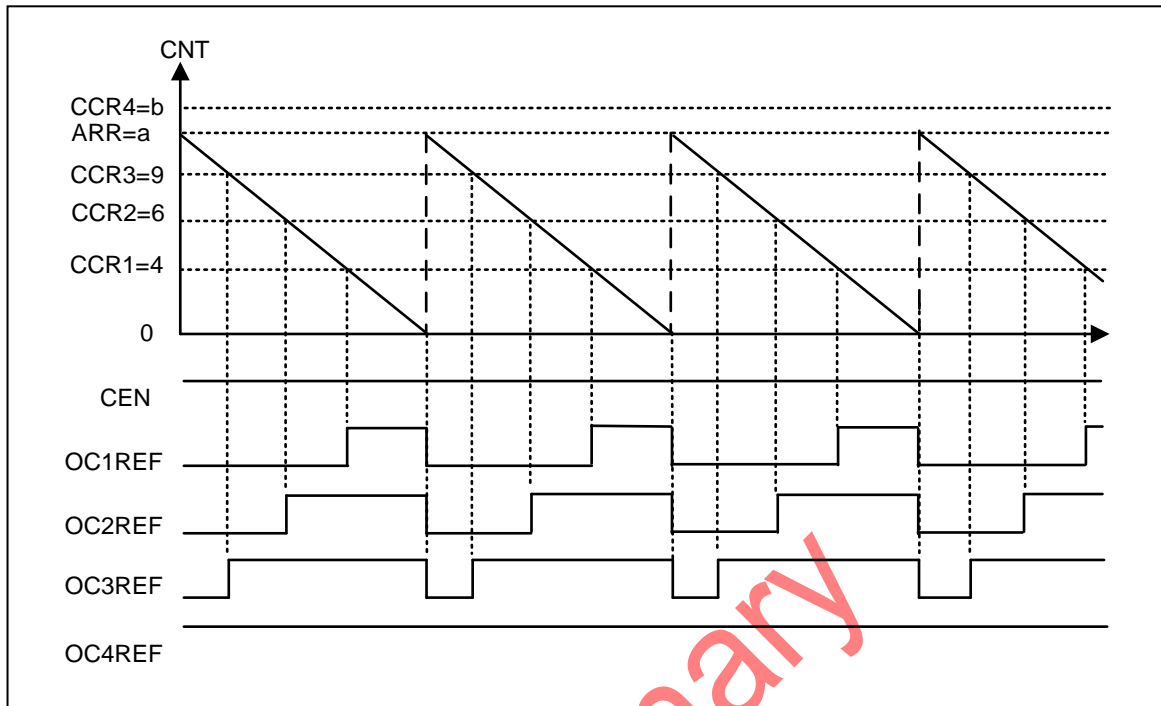


图 17-20 边沿对齐递减计数时 PWM 模式 1 的波形

17.4.4.3.3 PWM 中央对齐模式

首先配置 TIM1 计数器为中央对齐计数模式，配置 $TIM1_CCMR_x$ 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。

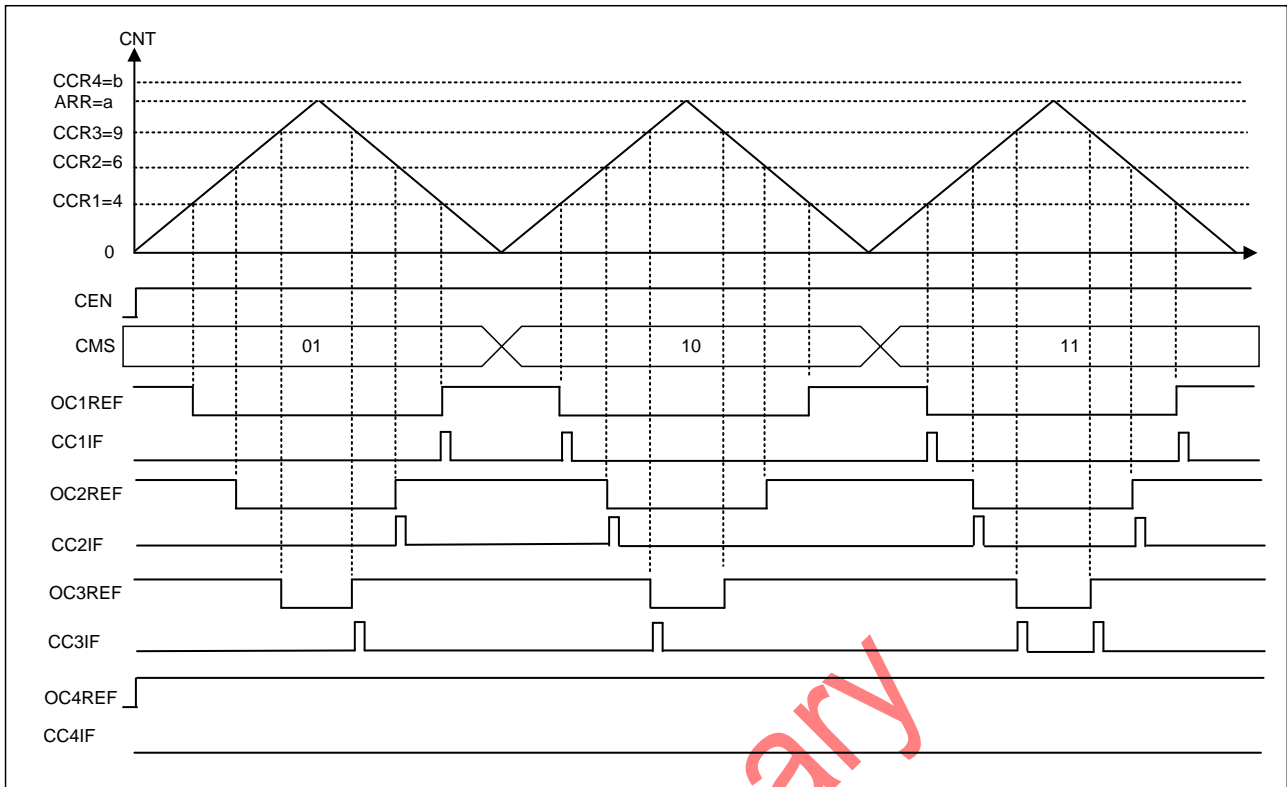


图 17-21 中央对齐 PWM 模式 1 的波形

17.4.4.3.4 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捕获/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。

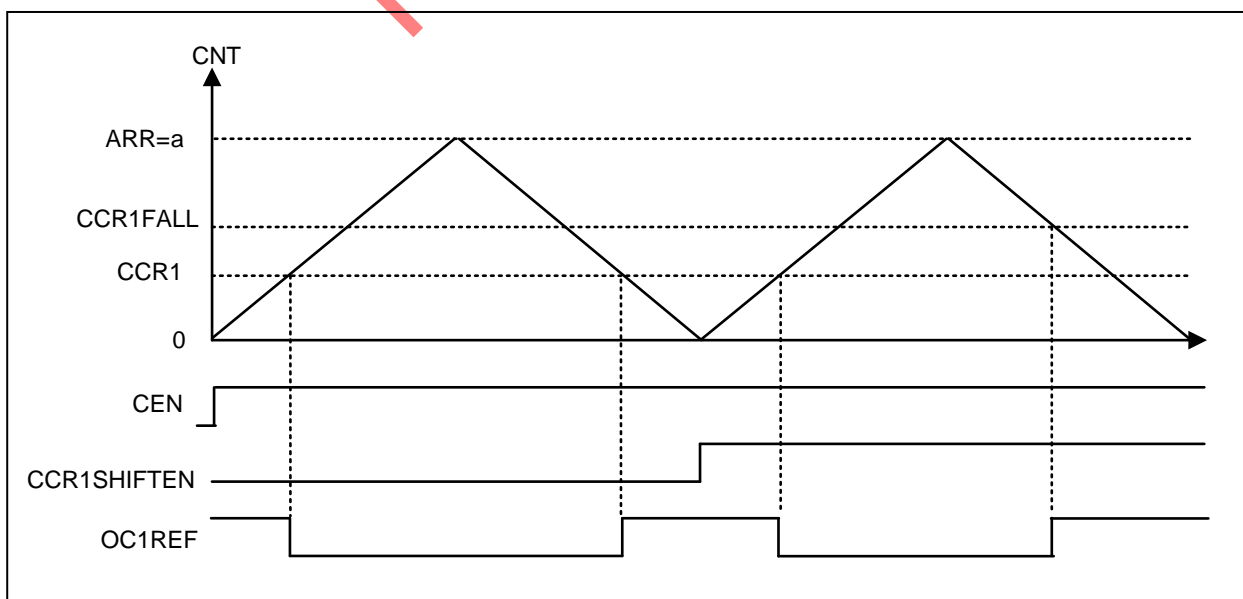


图 17-22 移相功能示意图

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 $> \text{TIM1_ARR}$ ，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。

17.4.4.3.5 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。

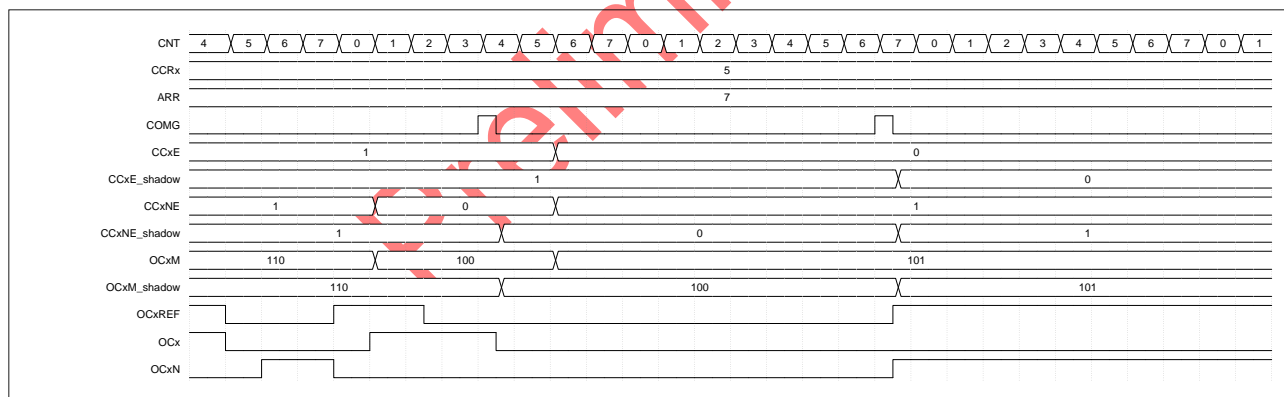


图 17-23 使用 COM 事件产生六步 PWM (OSSR = 1)

17.4.4.4 互补输出和死区插入

OCx 和 OCxN 是一对互补输出通道，TIM1 的通道 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 17-2 死区时间计算

DTG[7: 5]	DT
0xx	$DT = \text{DTG}[7: 0] \times T_{dtg} \quad (T_{dtg} = T_{DTS})$

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

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 $16\mu\text{s}$ 至 31750ns 。
- 若步长时间为 $1\mu\text{s}$ ，死区时间为 $32\mu\text{s}$ 至 $63\mu\text{s}$ 。
- 若步长时间为 $2\mu\text{s}$ ，死区时间为 $64\mu\text{s}$ 至 $126\mu\text{s}$ 。

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 3、表 4、表 5 和表 6 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。

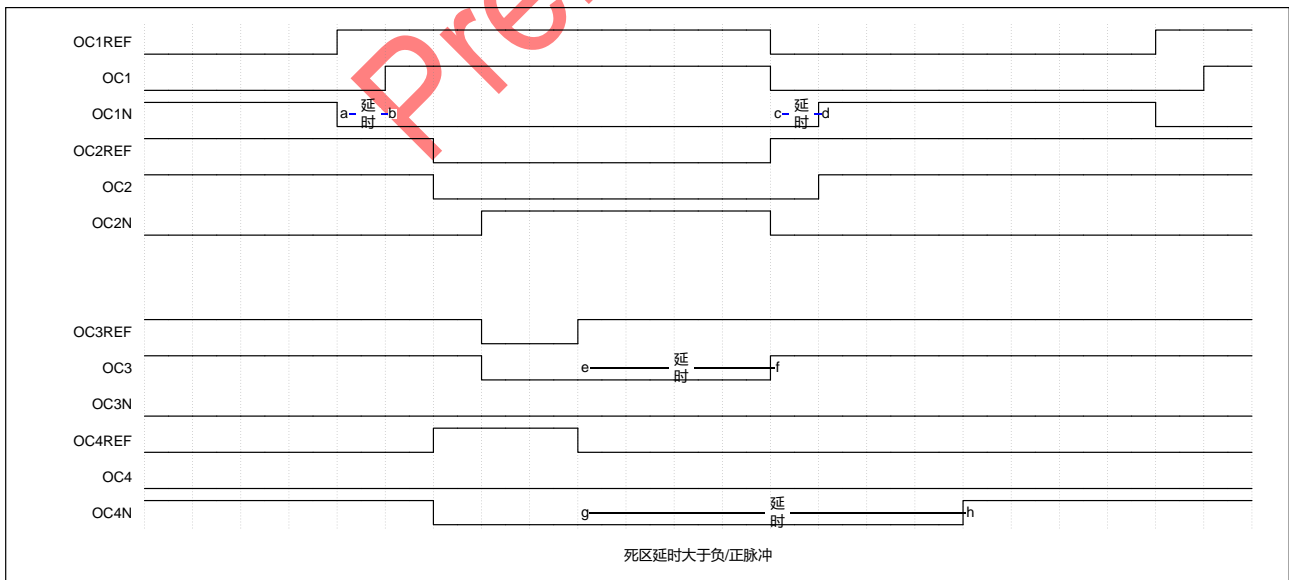


图 17-24 死区插入

17.4.4.5 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 17-3 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 时间, OCx_EN=1	OCxN=OCxREF 反相+Polarity+ 死区时间, OCxN_EN=1

表 17-4 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 17-5 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	

CCxE	CCxNE	OCx	OCxN
		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 17-6 当 MOE=0, OSS1=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：

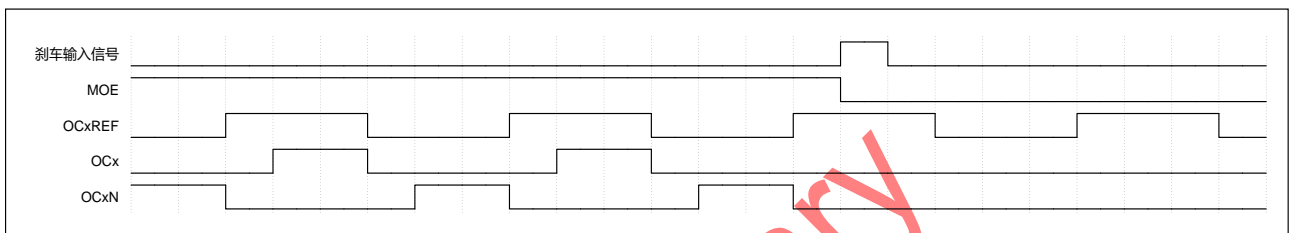


图 17-25 响应刹车的输出（OISx=0, OISxN=0）

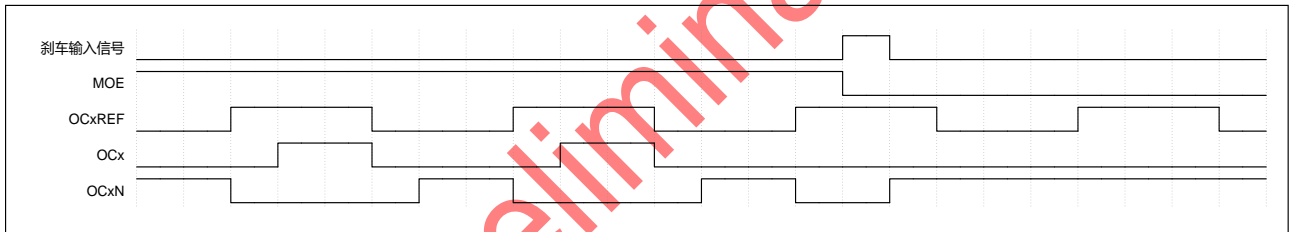


图 17-26 响应刹车的输出（OISx=0, OISxN=1）

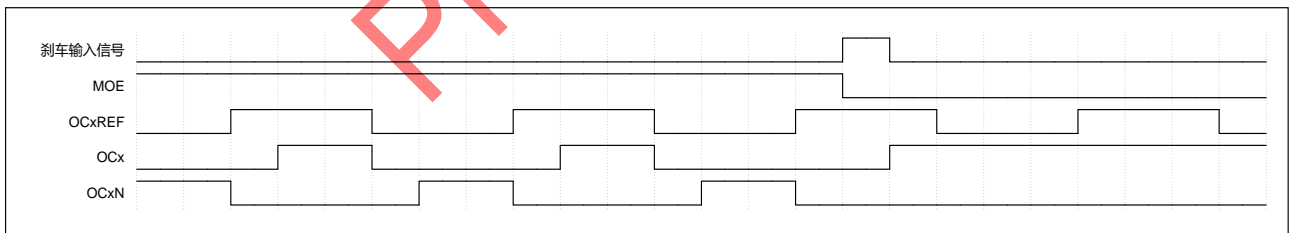


图 17-27 响应刹车的输出（OISx=1, OISxN=0）

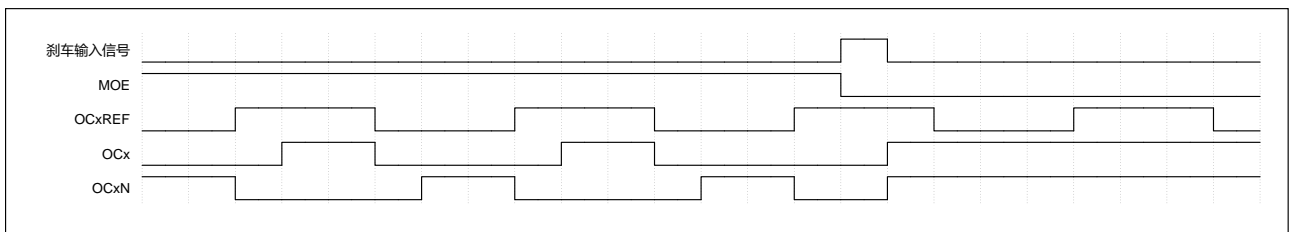


图 17-28 响应刹车的输出（OISx=1, OISxN=1）

17.4.4.6 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)。

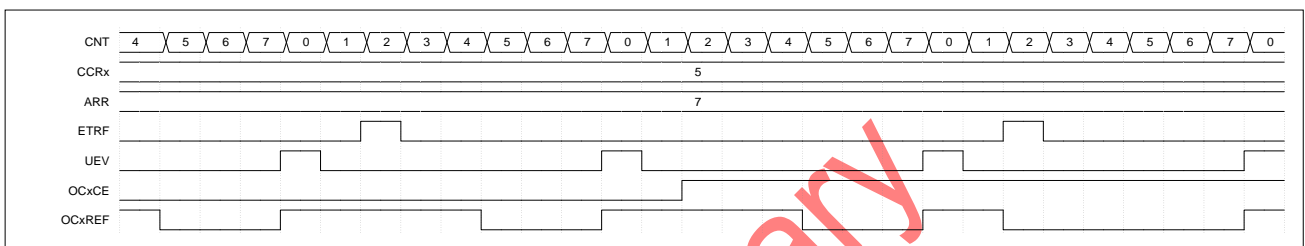


图 17-29 外部事件清除 OCxREF

17.4.4.7 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 $CNT > CCRx$ 。

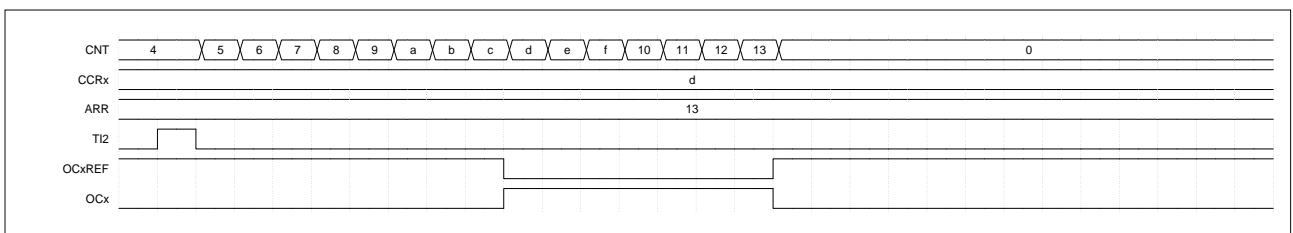


图 17-30 单脉冲模式

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 TIM1_CCMR1 寄存器中的 CC2S = 01, 将 TI2FP2 映射到 TI2。
2. 配置 TIM1_CCER 寄存器中的 CC2P = 0, 检测 TI2FP2 的上升沿。
3. 配置 TIM1_SMCR 寄存器中的 TS = 110, TI2FP2 作为从模式控制器的触发 (TRGI)。

- 配置 TIM1_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 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：

- 配置 TIM1_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
- 配置 TIM1_CCER 寄存器 CC1P = 1，输出低电平有效。
- 配置 TIM1_CCMR1 中 OC1PE = 1 和 TIM1_CR1 寄存器中 ARPE=1，使能预装载寄存器。
- 配置 TIM1_CCR1 寄存器和 TIM1_ARR 寄存器。
- 配置 TIM1_EGR 寄存器 UG=1 产生一个更新事件。
- 等待在 TI2 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。

17.4.4.7.1 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。

17.4.5 从模式

17.4.5.1 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 TIM1_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 TIM1_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 17-7 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 TIM1_CCMR 寄存器的 CC1S=01，将 IC1FP1 映射到 TI1 上。
2. 配置 TIM1_CCMR 寄存器的 CC2S =01，将 IC2FP2 映射到 TI2 上。
3. 配置 TIM1_CCER 寄存器的 CC1P =0，IC1 不反相，此时 IC1=TI1。
4. 配置 TIM1_CCER 寄存器的 CC2P =0，IC2 不反相，此时 IC1=TI2。
5. 配置 TIM1_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 TIM1_CR1 寄存器的 CEN =1，开启计数器。

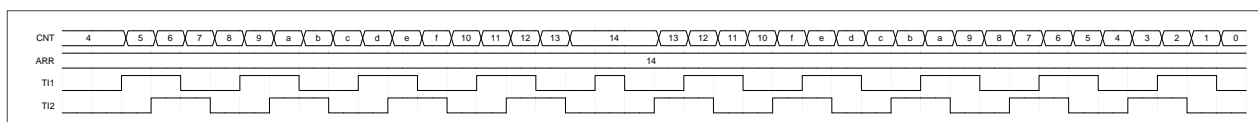


图 17-31 编码器模式下的计数器时序图

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

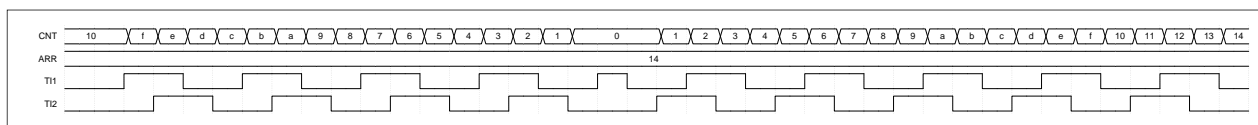


图 17-32 IC1FP1 反相编码器接口模式时序图

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。

17.4.5.2 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，TI2 输入端的下降沿触发计数器重启：

1. 配置 TIM1_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 TIM1_CCER 寄存器的 CC2P=1，检测下降沿。
2. 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 = 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 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 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 TIM1_ARR = 0x13 的时序图。

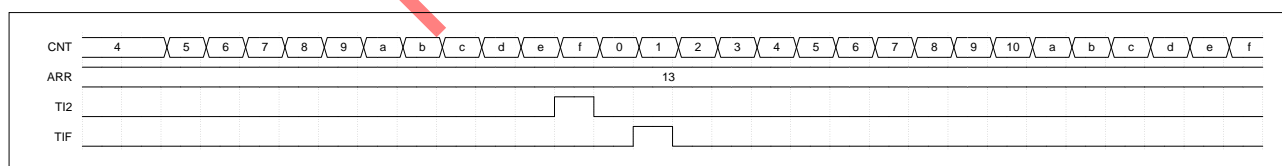


图 17-33 复位模式的控制时序图

17.4.5.3 门控模式

配置 TIM1_SMCR 寄存器 SMS=101，从模式选择门控模式。此模式下，根据 TIM1_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 TIM1_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 TIM1_CCER 寄存器的 CC1P=0，检测 TI1 上的高电平。

2. 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=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 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 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。计数器停止会将 TIF 置 1。

下图为门控模式下 TIM1_ARR=0xf 的时序图。

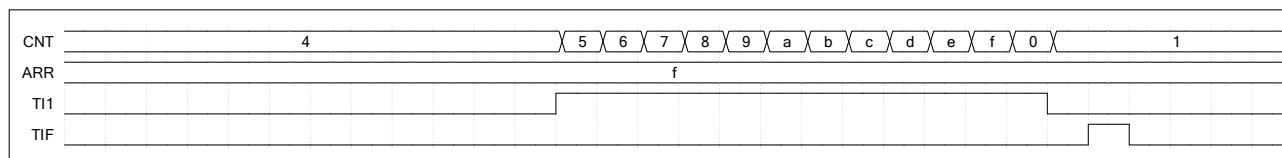


图 17-34 门控模式下的控制时序图

17.4.5.4 触发模式

配置 TIM1_SMCR 寄存器 SMS=110, 从模式选择触发模式。根据 TIM1_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 TIM1_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 TIM1_CCER 寄存器的 CC1P=0, 检测上升沿。

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=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 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 TI1 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。

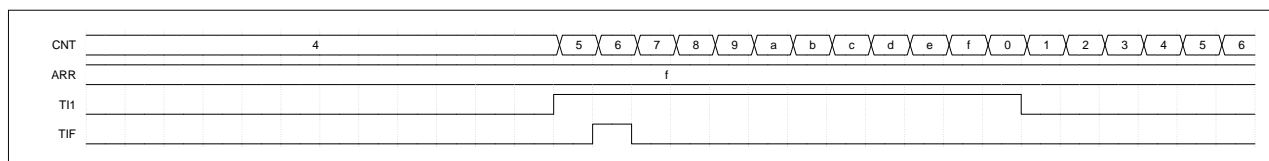


图 17-35 触发器模式下的控制时序图

17.4.5.5 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 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_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 TIM1_CCER 寄存器的 CC1P=0，选择上升沿有效。

3. 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 = 101，选择 TI1 作为输入源。

4. 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 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 TIM1_ARR=13 时的时序图。

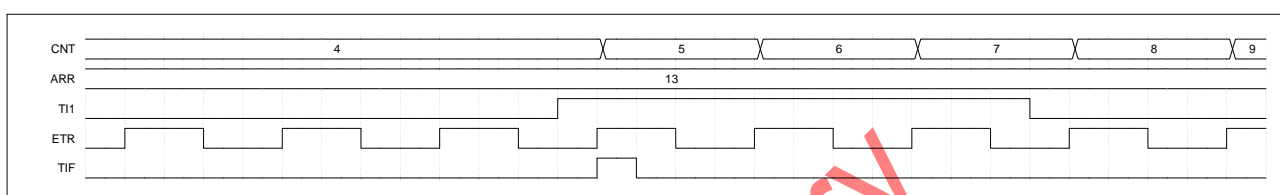


图 17-36 外部时钟模式 2+从模式（触发模式）控制时序图

17.4.6 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：

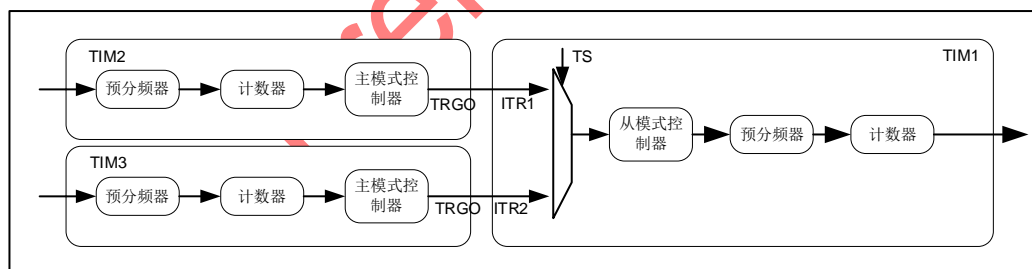


图 17-37 定时器间互联

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：使用 TIM3 作为 TIM1 的预分频器，时序图如下图所示：

1. TIM3 为主模式，配置 TIM3_CR2 寄存器的 MMS=010，TIM3 的更新事件作为触发输出 (TRGO)，TIM3 在每次更新事件时输出一个周期信号。

2. 配置 TIM3_ARR 寄存器，作为 TIM3 的输出周期。

3. 配置 TIM1_SMCR 寄存器的 TS=010，选择 TIM1 的触发源为 TIM3_TRGO。

4. 配置 TIM1_SMCR 寄存器的 SMS=111，从模式选择外部时钟模式 1。

5. 配置 TIM3_CR1 寄存器的 CEN=1，启动 TIM3。

6. 配置 TIM1_CR1 寄存器的 CEN=1，启动 TIM1。

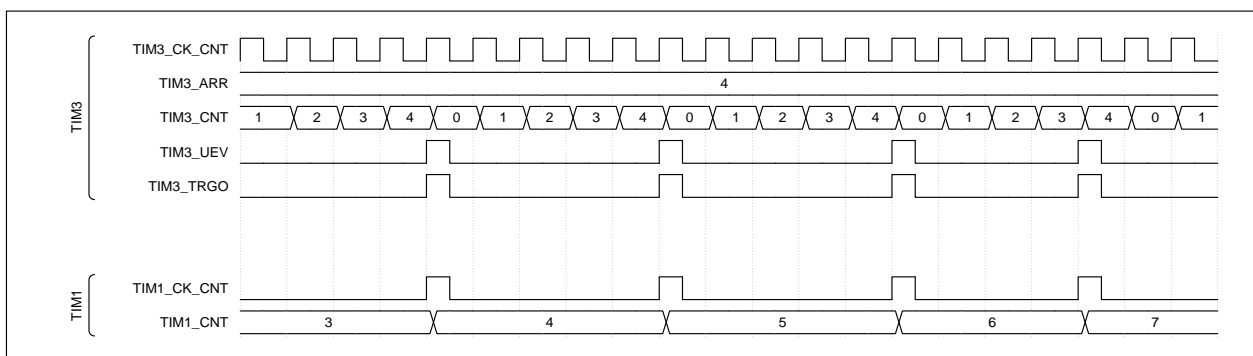


图 17-38 使用 TIM3 作为 TIM1 的预分频器

使用一个定时器使能另一个定时器

例：使用 TIM1 使能 TIM2，时序图如下图所示：

当 TIM1 的 OC1REF 为高时，TIM2 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. TIM1 为主模式，配置 TIM1_CR2 寄存器的 MMS=100，选择 TIM1 的比较输出参考信号(OC1REF) 作为触发输出 (TRGO)。
2. 配置 TIM1_CCR1 寄存器，TIM1_ARR 寄存器，TIM1_CCMR1 寄存器的 OC1M 位，CC1S 位等相关控制位，配置 TIM1 输出信号 TRGO 的输出波形。
3. 配置 TIM2_SMCR 寄存器的 TS=000，选择 TIM1 的 OC1REF 作为 TIM2 的触发输入。
4. 配置 TIM2_SMCR 寄存器的 SMS=101，选择 TIM2 为门控模式。
5. 配置 TIM2_CR1 寄存器的 CEN=1，启动 TIM2。
6. 配置 TIM1_CR1 寄存器的 CEN=1，启动 TIM1。

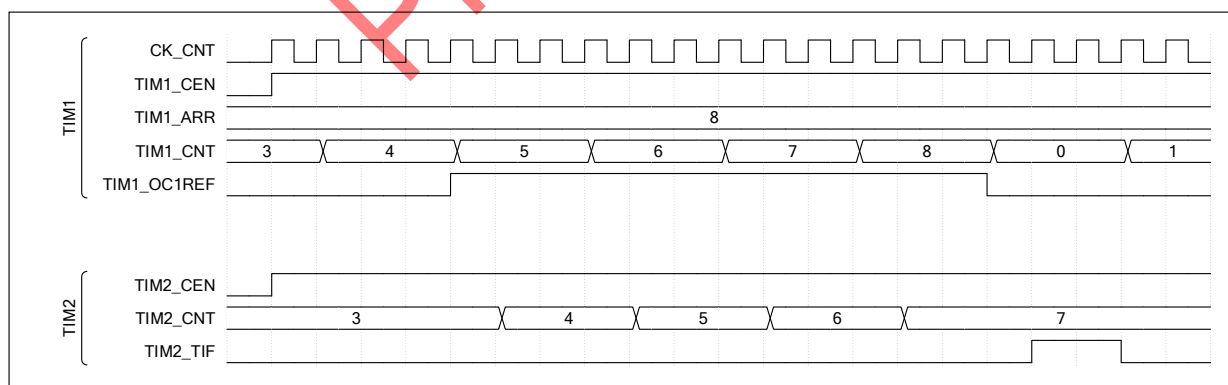


图 17-39 使用 TIM1 使能 TIM2

使用一个定时器启动另一个定时器

例：使用 TIM1 的更新事件启动 TIM2，时序图如下图所示：

当 TIM1 产生更新事件时，TIM2 接收到触发信号，TIM2 的 CEN 由硬件自动置 1，TIM2 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 TIM1_CR2 寄存器的 MMS=010，选择 TIM1 的更新事件为触发输出 (TRGO)；

2. 配置 TIM1_ARR 寄存器，作为更新事件产生的周期；
3. 配置 TIM2_SMCR 寄存器的 TS=000，选择 TIM1 的 TRGO 作为 TIM2 的触发输入；
4. 配置 TIM2_SMCR 寄存器的 SMS=110，选择 TIM2 为触发模式；
5. 配置 TIM1_CR1 寄存器的 CEN=1，启动 TIM1。

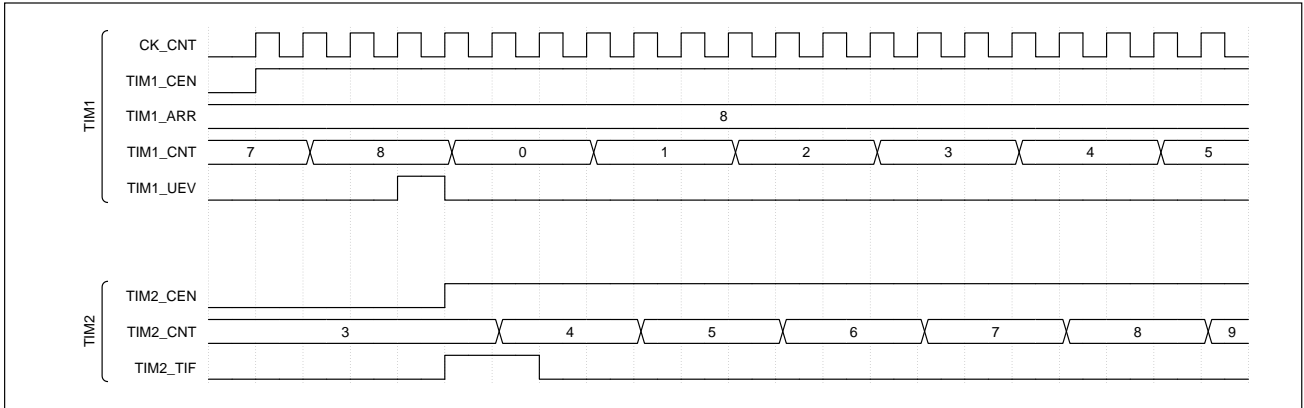


图 17-40 使用 TIM1 的更新事件启动 TIM2

使用一个外部触发同步启动两个定时器

例：使用 TIM1 的 TI1 上升沿，启动 TIM1 的同时启动 TIM2，时序图如下图所示：

为了确保两个定时器同时开启，TIM1 必须在主/从模式下配置。具体配置如下：

1. 配置 TIM1_CR2 寄存器的 MMS=001，设置 TIM1 作为主模式时，将 TIM1 的使能信号 CEN 作为触发输出（TRGO）。
2. 配置 TIM1_SMCR 寄存器的 TS=100，设置 TIM1 作为从模式时，将 TI1 作为触发输入。
3. 配置 TIM1_SMCR 寄存器的 SMS=110，选择 TIM1 为触发模式。
4. 配置 TIM2_SMCR 寄存器的 TS=000，选择 TIM1 的触发输出作为 TIM2 的触发输入。
5. 配置 TIM2_SMCR 寄存器的 SMS=110，选择 TIM2 为触发模式。

当 TIM1 的 TI1 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。

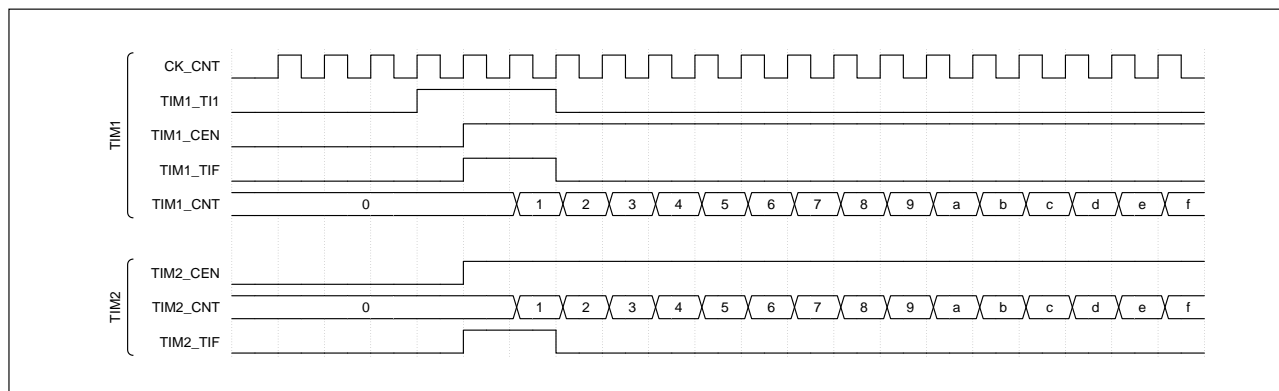


图 17-41 TIM1 的 TI1 同步启动 TIM1 和 TIM2

17.4.7 定时器异或

配置 TIM1_CR2 寄存器的 TI1S =1，将 TIM1_CH1、TIM1_CH2 和 TIM1_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：TIM1_CH1、TIM1_CH2 和 TIM1_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 TIM1_CCR1 寄存器中。具体配置如下：

1. 配置 TIM1_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 TIM1_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 TIM1_CCMR1 寄存器的 IC1F[3: 0]，配置数字滤波器的滤波宽度（按需配置）。
4. 配置 TIM1_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
5. 配置 TIM1_CCMR1 寄存器的 IC1PSC，选择预分频系数。
6. 配置 TIM1_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。
7. 配置 TIM1_CR1 寄存器的 CEN=1，启动计数器。

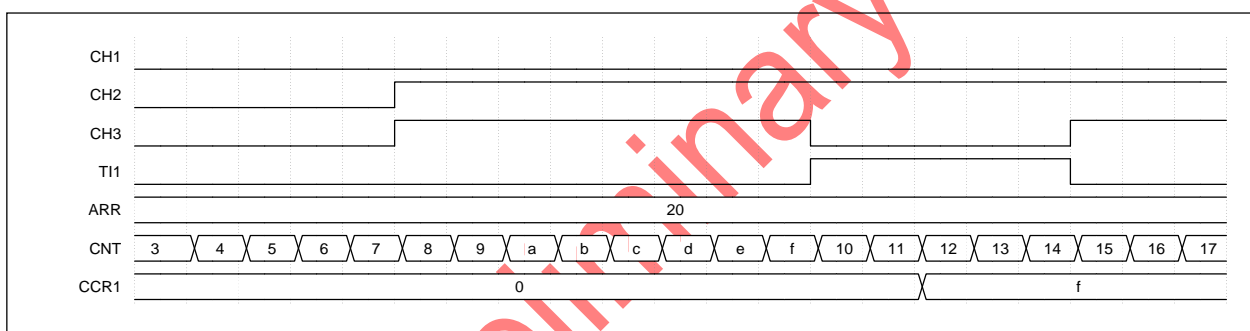


图 17-42 (TI1 异或输入) 输入捕获波形图

17.4.7.1.1 霍尔接口电路

霍尔传感器接口模式是异或功能的一个应用实例，可以用来驱动电机，在使用 TIM1 产生 PWM 信号驱动电机时，可以将另一个计数器（例如 TIM2/TIM3）作为“接口定时器”来连接霍尔传感器，将“接口定时器”的 3 个输入脚（CH1、CH2、CH3）经异或后连接到 TI1 输入通道，“接口定时器”接收这个信号。三个霍尔传感器与“接口定时器”的三路输入捕获引脚对应连接，每个传感器输入一路波形到输入引脚。分析输入捕获信号可以计算电机速度信息。

“接口定时器”在输出模式下可以产生一个用来控制 TIM1_PWM 输出的脉冲，用来驱动电机。所以“接口定时器”在比较输出或 PWM 模式延时一段时间后产生一个正脉冲，然后通过 TRGO 输出到 TIM1。

例：霍尔输入连接到 TIM2 定时器，每次任一霍尔输入上信号变化都会改变 TIM1 的 PWM 配置。

1. 配置 TIM2_CR2 寄存器的 TI1S=1，配置三个定时器输入经异或后连接到 TI1 输入通道。
2. 配置 TIM2_ARR 为其最大值（计数器必须通过 TI1 的变化清零）。配置 PSC，设置计数周期大于传感器上两次变化的间隔时间。
4. 配置 TIM2_CCMR1 的寄存器 CC1S=01，选择通道 1 为捕获模式（选中 TRC）。

5. 配置 TIM2_CCMR1 寄存器的 CC2S=00, OC2M=111, 配置通道 2 为 PWM2 模式, 并具有要求的延时。

6. 配置 TIM2_CR2 寄存器的 MMS=101, 选择 OC2REF 作为 TRGO 上的触发输出。

TIM1 中, ITR 输入必须是 TIM2 的 TRGO, 定时器被编程为可产生 PWM 信号, 捕获/比较控制信号设置为可预装载的 (TIM1_CR2 寄存器中 CCPC = 1), 同时触发输入用于控制 COM 事件 (TIM1_CR2 寄存器中 CCUS = 1)。在一次 COM 事件后, 写入下一步的 PWM 控制位 (CCxE、OCxM), 这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例:

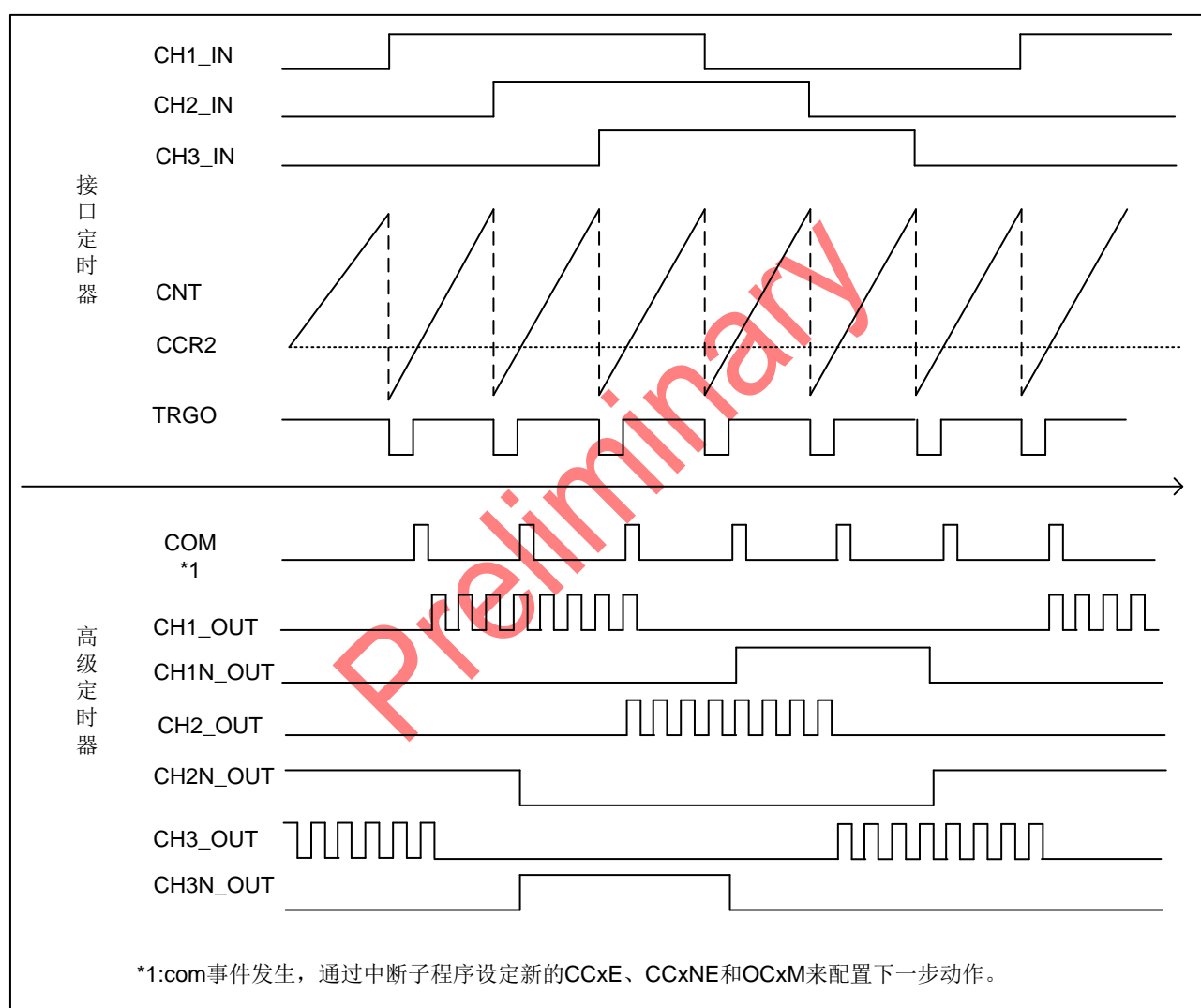


图 17-43 霍尔传感器接口实例

17.4.8 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)

17.4.9 中断

TIM1 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、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 17-8 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
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

17.4.10 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。

17.5 寄存器描述

表 17-9 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	捕获/比较模式寄存器 1	0x0000
0x1C	TIM1_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIM1_CCER	捕获/比较使能寄存器	0x0000
0x24	TIM1_CNT	计数器	0x0000
0x28	TIM1_PSC	预分频率器	0x0000
0x2C	TIM1_ARR	自动装载寄存器	0x0000
0x30	TIM1_RCR	重复计数寄存器	0x0000
0x34	TIM1_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIM1_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIM1_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIM1_CCR4	捕获/比较寄存器 4	0x0000
0x44	TIM1_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	捕获/比较模式寄存器 3	0x0000
0x58	TIM1_CCR5	捕获/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 移相递减计数捕获/比较寄存器	0x0000
0x74	TIM1_BKINF	刹车输入滤波寄存器	0x0000 0000

17.5.1 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
Res.						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, Tlx）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

Bit	Field	Description
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, 计数器和预分频器被初始化; 如果从模式控制器接收到硬件复位, 计数器将被初始化。
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。

17.5.2 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	TI1S		MMS		CCDS	CCUS	Res.	CCPC
	rw	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。

Bit	Field	Description
7	TI1S	<p>TI1 选择 (TI1 selection)</p> <p>0: TIM1_CH1 管脚连到 TI1 输入</p> <p>1: TIM1_CH1、TIM1_CH2 和 TIM1_CH3 管脚经异或后作为 TI1 输入</p>
6: 4	MMS	<p>主模式选择 (Master mode selection)</p> <p>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:</p> <p>000: 复位 TIM1_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。</p> <p>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。</p> <p>010: 更新 更新事件被选为 TRGO。</p> <p>011: 捕获/比较脉冲 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。</p> <p>100: 比较 OC1REF 信号被用于作为触发输出 (TRGO)</p> <p>101: 比较 OC2REF 信号被用于作为触发输出 (TRGO)</p> <p>110: 比较 OC3REF 信号被用于作为触发输出 (TRGO)</p> <p>111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)</p>
3	CCDS	<p>DMA 请求源选择 (Capture/compare DMA selection)</p> <p>0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求</p> <p>1: 发生更新事件时, 发送 CCx 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品</p>
2	CCUS	<p>捕获/比较控制更新源选择 (Capture/compare control update selection)</p> <p>0: CCPC=1 时, 只能配置 COMG=1 更新。</p> <p>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。</p> <p>注: 此位只在通道为互补输出时有效。</p>
1	Reserved	保留, 必须保持复位值。
0	CCPC	<p>捕获/比较预装载控制位 (Capture/compare preloaded control)</p> <p>0: CCxE, CCxNE 和 OCxM 位预装载禁用</p> <p>1: CCxE, CCxNE 和 OCxM 位预装载使能</p> <p>注: 此位只在通道为互补输出时有效。</p>

17.5.3 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	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择 ETR 信号的极性。</p> <p>0: 高电平或上升沿有效</p> <p>1: 低电平或下降沿有效</p> <p>注: 仅适用于支持外部触发的产品</p>
14	ECE	<p>外部时钟使能位 (External clock enable)</p> <p>该位启用外部时钟模式 2。</p> <p>0: 禁止外部时钟模式 2</p> <p>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</p> <p>注 1: 仅适用于支持外部触发的产品。</p> <p>注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。</p> <p>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。</p> <p>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。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler)</p> <p>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。</p> <p>00: 关闭预分频</p> <p>01: ETRP 频率除以 2</p> <p>10: ETRP 频率除以 4</p> <p>11: ETRP 频率除以 8</p> <p>注: 仅适用于支持外部触发的产品。</p>

Bit	Field	Description
11: 8	ETF	<p>外部触发滤波 (External trigger filter)</p> <p>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。</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> <p>注: 仅适用于支持外部触发的产品。</p>
7	MSM	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用</p> <p>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。</p>
6: 4	TS	<p>触发选择 (Trigger selection)</p> <p>触发输入源选择。</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>更多有关 ITRx 的细节, 参见下表。</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</p> <p>注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。</p>

Bit	Field	Description
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), 并且产生一个更新事件。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), 并且产生一个更新事件, 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 17-10 TIM1 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	TIM2_TRGO	TIM3_TRGO	TIM17_OC1REF
TIM2	TIM1_TRGO	-	TIM3_TRGO	TIM14_OC1REF
TIM3	TIM1_TRGO	TIM2_TRGO	TIM16_OC1REF	TIM14_OC1REF

17.5.4 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
Res.													CC5DE	CC5IE	
													r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit	Field	Description
31: 18	Reserved	保留, 必须保持复位值。

Bit	Field	Description
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	BIE	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断 1: 允许刹车中断

Bit	Field	Description
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

17.5.5 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
Res.															CC5IF
rw0c															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
r_w0c							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: 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。

Bit	Field	Description
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	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和外部时钟模式外的其它模式时、在 TRGI 输入端检测到有效边沿, 或门控模式下的计数停止边沿) 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 (捕获/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	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。

Bit	Field	Description
1	CC1IF	<p>捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag)</p> <p>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 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</p> <p>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 TIM1_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIM1_CCR1</p>
0	UIF	<p>更新中断标记 (Update interrupt flag)</p> <p>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</p> <p>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。</p>

17.5.6 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
Res.															CC5G
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

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。

Bit	Field	Description
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	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 CCxE、CCxNE、OCxM 位。 注: 该位只对拥有互补输出的通道有效。
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: 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。

17.5.7 TIM1_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 TIM1_SMCR 寄存器的 TS 位选择）
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) 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 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 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 (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>

Bit	Field	Description
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 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 01: 通道 1 被配置为输入, IC1 映射在 TI1 上 10: 通道 1 被配置为输入, IC1 映射在 TI2 上 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 TIM1_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 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 TIM1_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 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 $f_{sampling}=f_{DTS}/32$, $N=8$</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 $CC1E=0$ ($TIM1_CCER$ 寄存 器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次 捕获 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</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 01: 通道 1 被配置为输入, IC1 映射在 TI1 上 10: 通道 1 被配置为输入, IC1 映射在 TI2 上 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 $TIM1_SMCR$ 寄存器的 TS 位选择)</p>

17.5.8 TIM1_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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

比较输出模式:

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	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 TIM1_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

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode)</p> <p>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。</p> <p>000: 冻结。TIM1_CCR3 与 TIM1_CNT 间的比较结果对 OC3REF 不起作用</p> <p>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</p> <p>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</p> <p>011: 匹配时翻转。当 TIM1_CCR3=TIM1_CNT 时, 翻转 OC3REF 的电平</p> <p>100: 强制为低。强制 OC3REF 为低电平</p> <p>101: 强制为高。强制 OC3REF 为高电平</p> <p>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。</p> <p>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。</p> <p>注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC3S=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, OC3REF 电平才改变。</p>
3	OC3PE	<p>通道 3 比较输出预装载使能 (Output compare 3 preload enable)</p> <p>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</p> <p>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</p> <p>注 1: 当 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC3S =00 (该通道配置成输出) 时, 该位不能被修改。</p> <p>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。</p>

Bit	Field	Description
2	OC3FE	<p>通道 3 比较输出快速使能 (Output compare 3 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 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能</p>
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 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 TIM1_SMCR 寄存器的 TS 位选择)</p>

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	<p>输入捕获 4 滤波器 (Input capture 4 filter)</p> <p>参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler)</p> <p>参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>通道 4 捕获/比较选择 (Capture/Compare 4 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 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 TIM1_SMCR 寄存器的 TS 位选择)</p>

Bit	Field	Description
7: 4	IC3F	<p>通道 3 输入捕获滤波器 (Input capture 3 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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler)</p> <p>该位定义了 IC3 的预分频系数。当 $CC3E=0$ ($TIM1_CCER$ 寄存器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 3 被配置为输出</p> <p>01: 通道 3 被配置为输入, IC3 映射在 TI3 上</p> <p>10: 通道 3 被配置为输入, IC3 映射在 TI4 上</p> <p>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 $TIM1_SMCR$ 寄存器的 TS 位选择)</p>

17.5.9 TIM1_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.	CC 4P	CC 4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw													

Bit	Field	Description
15	CC4NP	通道 4 输入/捕获互补输出极性 (Capture/Compare 4 complementary output 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/捕获互补输出极性 (Capture/Compare 3 complementary output polarity) 参考 CC1NP 的描述。
10	CC3NE	通道 3 输入/捕获互补输出使能 (Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
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/捕获互补输出极性 (Capture/Compare 2 complementary output polarity) 参考 CC1NP 的描述。
6	CC2NE	通道 2 输入/捕获互补输出使能 (Capture/Compare 2 complementary output enable) 参考 CC1NE 的描述。
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。
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 ICx 极性/电平选择表。 注: 当 LOCK 级别 (TIM1_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。

Bit	Field	Description
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 ICx 极性/电平选择表。 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, ICx 的极性/电平选择如下表所示:

表 17-11 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/高电平有效
0	1	保留

17.5.10 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)

17.5.11 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) = fck_psc / (PSC+1) 当发生更新事件时, PSC 的值装入当前预分频寄存器。

17.5.12 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。

17.5.13 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	<p>重复计数器实时写入的值 (Repetition counter value of real-time writing)</p> <p>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。</p> <p>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。</p>
7: 0	REP	<p>重复计数器的值 (Repetition counter value)</p> <p>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。</p> <p>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:</p> <p>在边沿对齐模式下, PWM 周期的数目</p> <p>在中央对齐模式下, PWM 半周期的数目</p>

17.5.14 TIM1_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	<p>通道 1 捕获/比较的值 (Capture/Compare 1 value)</p> <p>通道 1 配置为输入:</p> <p>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)。</p> <p>通道 1 配置为输出:</p> <p>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 TIM1_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。</p>

17.5.15 TIM1_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。

17.5.16 TIM1_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。

17.5.17 TIM1_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。

17.5.18 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
Res.															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。

Bit	Field	Description
11	OSSR	<p>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。</p> <p>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。</p> <p>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。</p>
10	OSSI	<p>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。</p> <p>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。</p> <p>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。</p>
9: 8	LOCK	<p>锁定设置（Lock configuration） 该位定义了寄存器的写保护功能。</p> <p>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（当相关通道通过 CCxS 位设为输出时，CC 极性位是 TIM1_CCER 寄存器的 CCxP/CCxNP 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 CCxS 位设为输出时，CC 控制位是 TIM1_CCMRx 寄存器的 OCxM/OCxPE 位）</p> <p>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。</p>
7: 0	DTG	<p>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。</p> <p>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。</p>

17.5.19 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 17.4.10 DMA 章节。

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

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

17.5.20 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 17.4.10 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。

17.5.21 TIM1_CCMR3 捕获/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
Res.												OC5PE	Res.		
												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。

17.5.22 TIM1_CCR5 捕获/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） CCR5 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 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。

17.5.23 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
Res.										CCR5_	CCR4_	CCR3_	CCR2_	CCR1_	CCDRE
										SHIFT_	SHIFT_	SHIFT_	SHIFT_	SHIFT_	PE
										EN	EN	EN	EN	EN	
										rW	rW	rW	rW	rW	rW

Bit	Field	Description
15: 6	Reserved	保留, 必须保持复位值。
5	CCR5_SHIFT_EN	允许通道 5 输出 PWM 移相使能位 0: 禁止通道 5 输出 PWM 移相 1: 允许通道 5 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
4	CCR4_SHIFT_EN	允许通道 4 输出 PWM 移相使能位 0: 禁止通道 4 输出 PWM 移相 1: 允许通道 4 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
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。

17.5.24 TIM1_CCRxFALL PWM 移相递减计数捕获/比较寄存器

偏移地址: 0x60 ~ 0x70

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捕获/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。

17.5.25 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
Res.															BKIN_S EL
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKIN_SEL											BKINF				BKINFE
rw											rw				rw

Bit	Field	Description
31: 17	Reserved	保留，必须保持复位值。
16: 5	BKIN_SEL	BKINSEL：刹车输入选择（break input sel） xxxxx1xxxxx：若第 6bit 为 1，则选择 CPT 输入信号 xxxxxx0xxxxx：第 5bit 保留，必须保持复位值。 xxxxxxx1xxxx：若第 4bit 为 1，则选择 PA[4]输入信号 xxxxxxxx1xxx：若第 3bit 为 1，则选择 PA[13]输入信号 xxxxxxxxx1xx：若第 2bit 为 1，则选择 PB[12]输入信号 xxxxxxxxxxx1x：若第 1bit 为 1，则选择 PA[6]输入信号 xxxxxxxxxxx1：若第 0bit 为 1，则选择 CSS 信号

Bit	Field	Description
4: 1	BKINF	<p>BKIN 数字滤波采样频率 (break input filter)</p> <p>0000: 2 周期</p> <p>0001: 4 周期</p> <p>0010: 8 周期</p> <p>0011: 16 周期</p> <p>0100: 32 周期</p> <p>0101: 64 周期</p> <p>0110: 128 周期</p> <p>0111: 256 周期</p> <p>1000: 384 周期</p> <p>1001: 512 周期</p> <p>1010: 640 周期</p> <p>1011: 768 周期</p> <p>1100: 896 周期</p> <p>1101: 1024 周期</p> <p>1110: 1152 周期</p> <p>1111: 1280 周期</p> <p>注: 更改档位时应先关闭刹车滤波使能位。</p>
0	BKINFE	<p>BKIN 数字滤波使能 (break input filter enable)</p> <p>1: 使能 BKIN 管脚数字滤波</p> <p>0: 禁止 BKIN 管脚数字滤波</p> <p>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。</p>

18 TIM2 通用定时器

18.1 简介

TIM2 由一个 16 位可实时编程预分频器和一个 32 位计数方向可调的自动重装计数器组成，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）。

18.2 功能框图

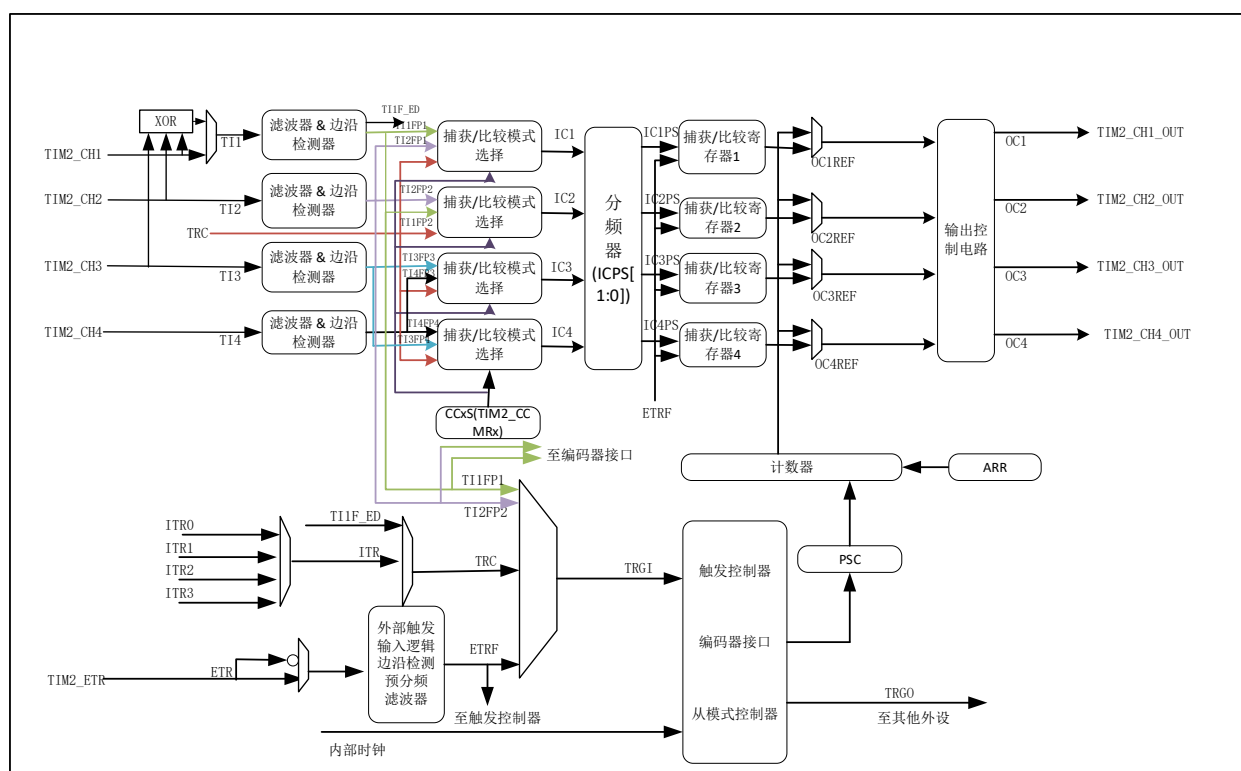


图 18-1 TIM2 结构图

上图为 TIM2 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。

18.3 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（TIX、ETRX），内部触发输入（ITRX）
- 32 位自动重载计数器（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

18.4 功能描述

18.4.1 时钟

18.4.1.1 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 TIx、ITRx、ETRx）
- 外部时钟模式 2：外部触发输入 ETR（包含 ETRx）
- 编码器模式

上述几种时钟选择示意图如下：

Preliminary

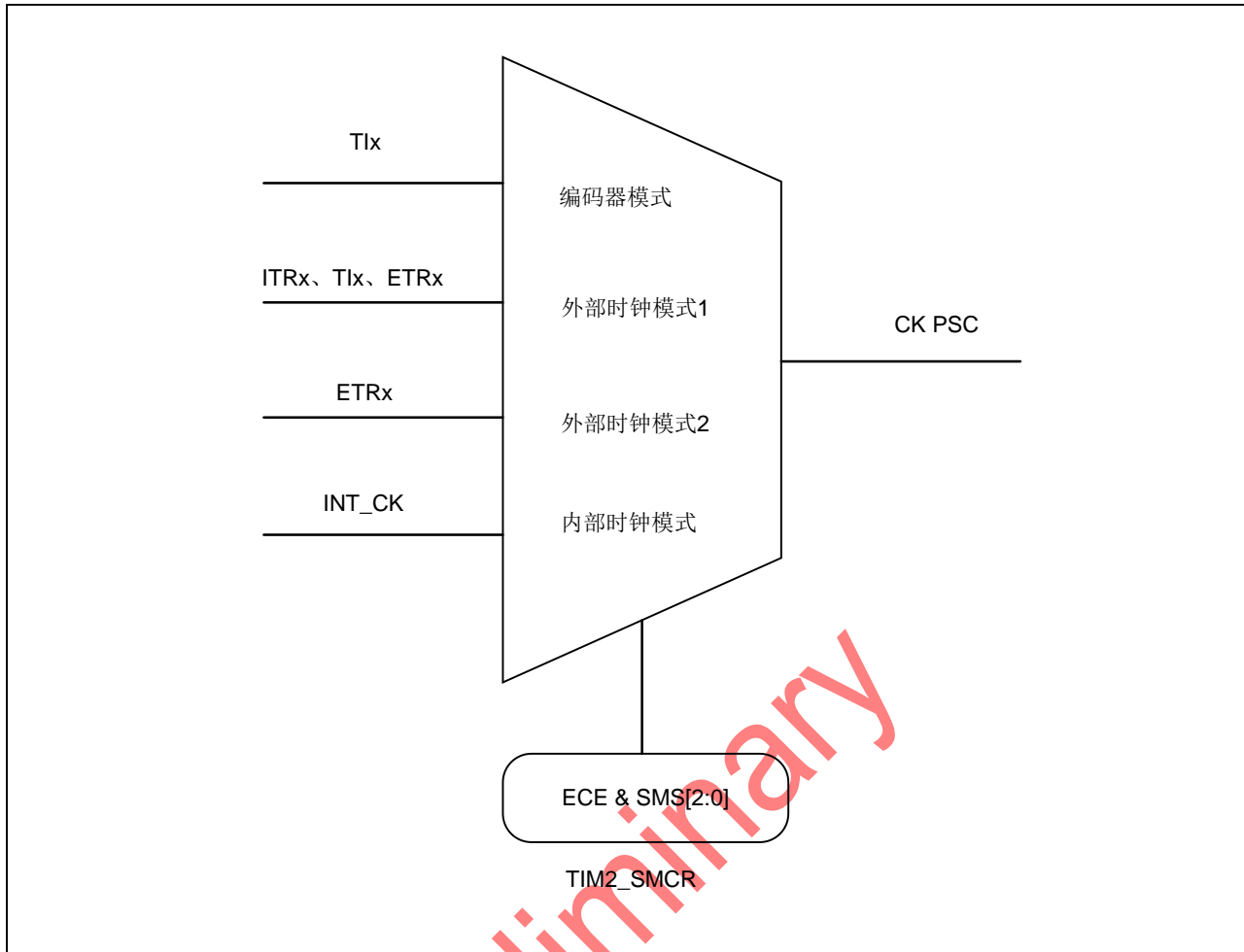


图 18-2 时钟选择

18.4.1.1.1 内部时钟源 (INT_CK)

当配置 TIM2_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。

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

当配置 TIM2_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 TIM2_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上；配置 TIM2_CCMR1 寄存器的 IC1F[3: 0]，设置输入滤波器带宽；配置 TIM2_CCER 寄存器 CC1P=0，选择上升沿为有效沿。
2. 配置 TIM2_SMCR 寄存器的 TS=101，选择 TI1 的作为触发输入源；配置 TIM2_SMCR 寄存器的 SMS=111，选择外部时钟模式 1。

3. 配置 TIM2_CR1 寄存器的 DIR=0，选择递增计数模式，配置 TIM2_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。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。

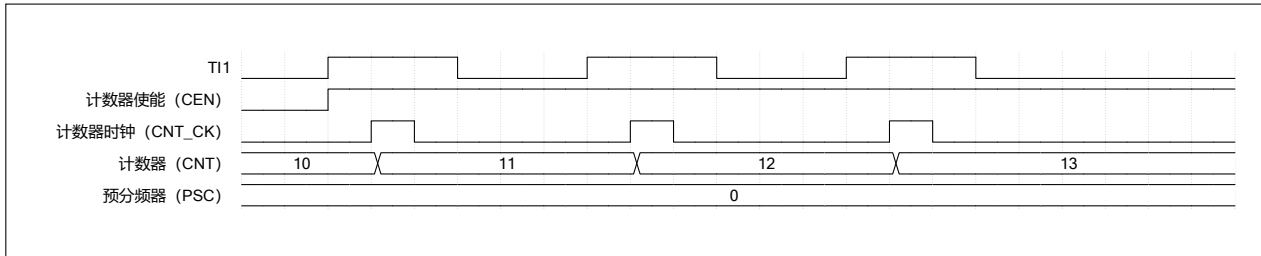


图 18-3 外部时钟模式 1 下的控制电路

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

当配置 TIM2_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 TIM2_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 TIM2_SMCR 寄存器的 ETP=1，选择下降沿有效；配置 TIM2_SMCR 寄存器的 ECE=1，选择外部时钟模式 2。

2. 配置 TIM2_CR1 寄存器的 DIR=0，选择递增计数模式；配置 TIM2_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。

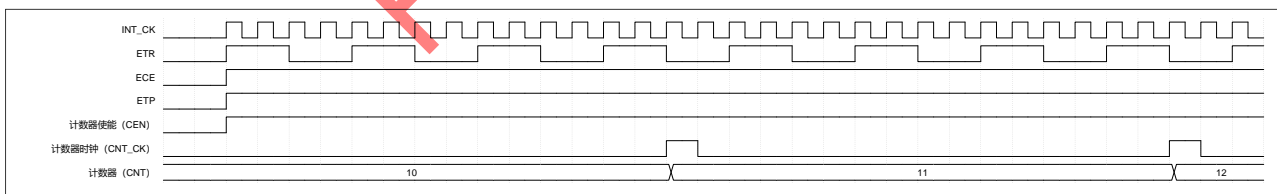


图 18-4 外部时钟模式 2 下的控制电路

18.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

18.4.1.2 时基单元

TIM2 的时基单元主要包括：计数器寄存器（TIM2_CNT）、预分频器寄存器（TIM2_PSC）和自动预装载寄存器（TIM2_ARR）。

计数单元由一个 32 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 32 位影子寄存器，通过设置 TIM2_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。

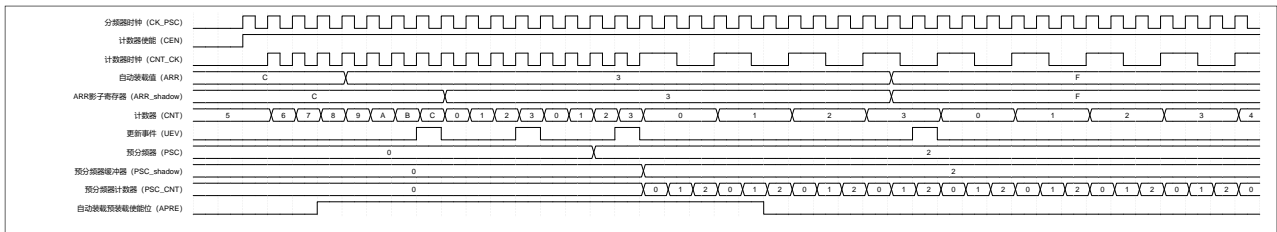


图 18-5 自动预装载

18.4.1.3 计数模式

通过配置 TIM2_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。

18.4.1.3.1 递增计数模式

配置 TIM2_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 TIM2_CR1 寄存器的 CEN 后计数器由 0 开始递增计数，直至 TIM2_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 TIM2_EGR 寄存器的 UG=1，同样可以产生一个更新事件。

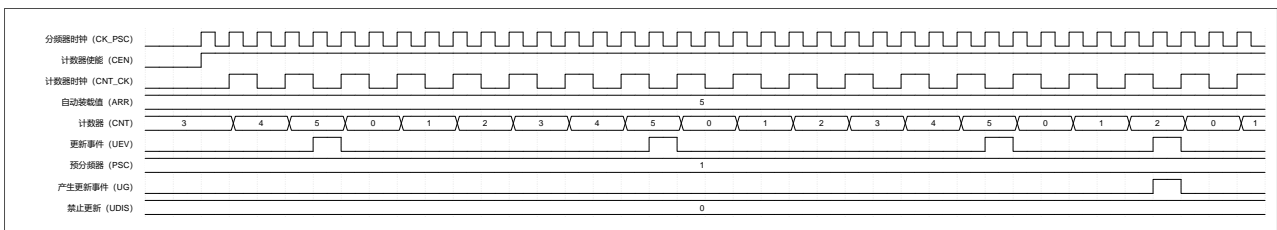


图 18-6 递增计数模式（UDIS=0）

通过配置 TIM2_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。

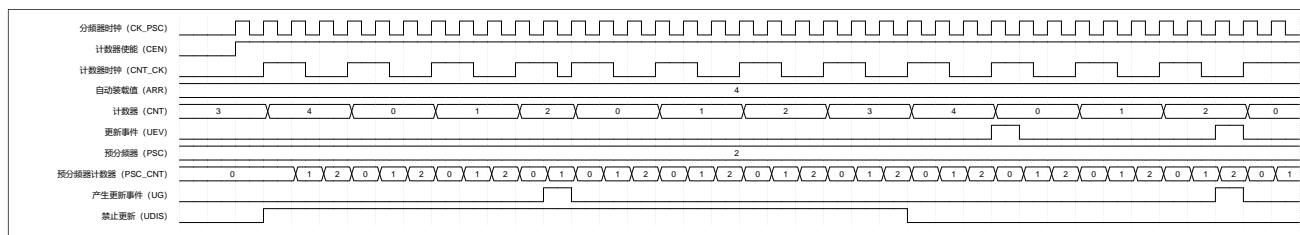


图 18-7 递增计数模式 (UDIS=1 禁止产生更新事件)

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。

18.4.1.3.2 递减计数模式

配置 TIM2_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 TIM2_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 TIM2_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 TIM2_ARR 开始重新递减计数（TIM2_CR1 寄存器 UDIS=0）。

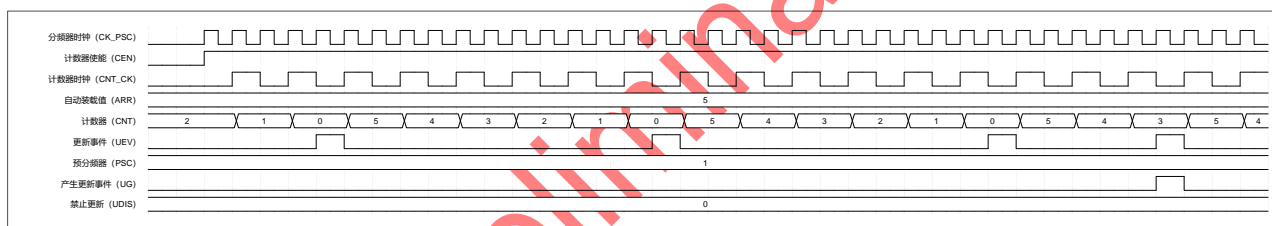


图 18-8 递减计数模式 (UDIS=0)

通过配置 TIM2_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 TIM2_ARR 开始计数。

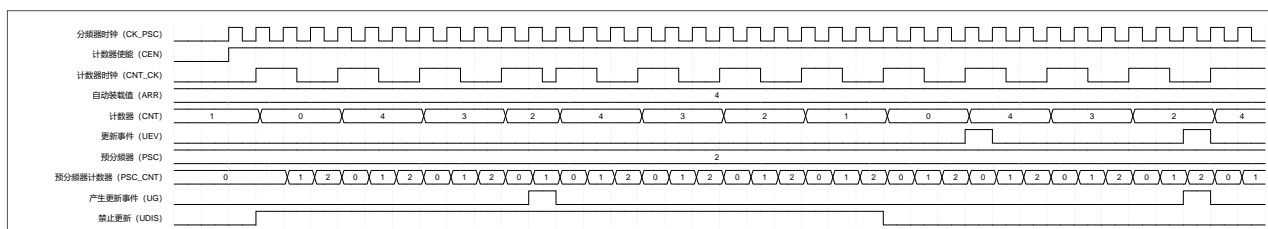


图 18-9 递减计数模式 (UDIS=1 禁止产生更新事件)

18.4.1.3.3 中央计数模式（递增/递减计数模式）

配置 TIM2_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 TIM2_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（TIM2_CR1 寄存器 UDIS=0）。

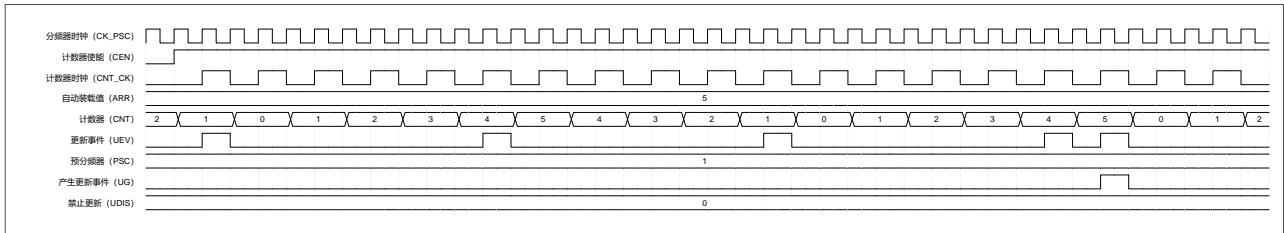


图 18-10 中央计数模式（UDIS=0）

通过配置 TIM2_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。

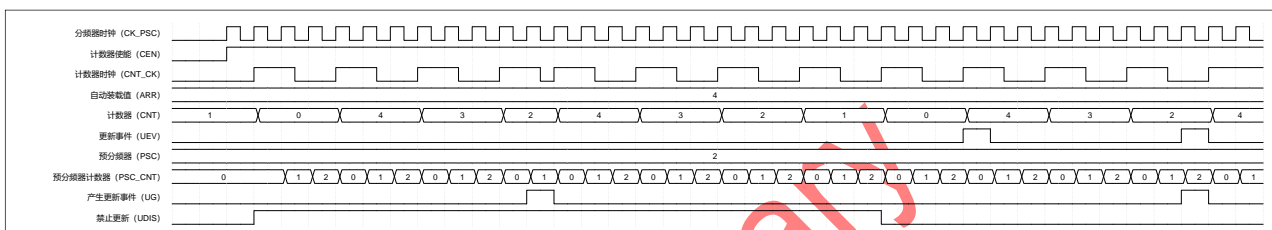


图 18-11 中央计数模式（UDIS=1 禁止产生更新事件）

18.4.2 输入捕获

18.4.2.1 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：

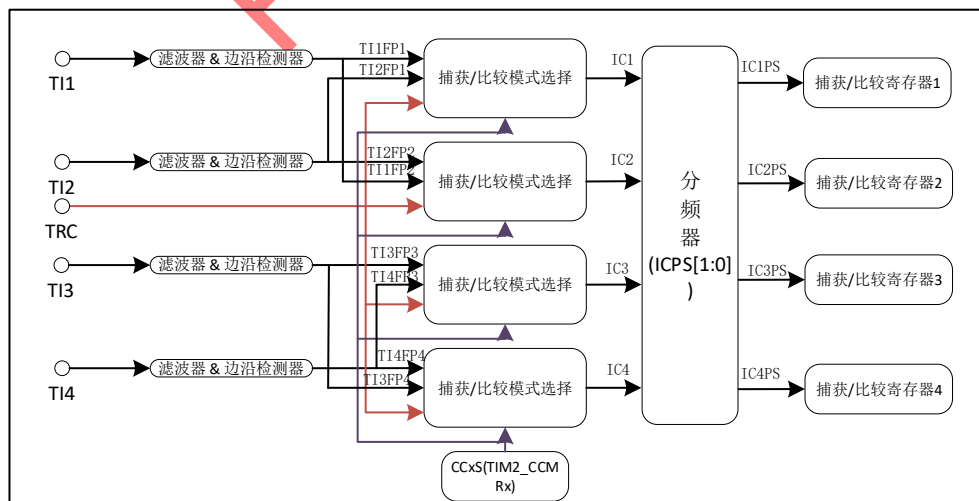


图 18-12 TIM2 输入捕获结构图

通过配置 TIM2_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 18-1 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 ($TIM2_SR$) 中的捕获标志位 $CCxIF$ 置 1，通过配置 $CCxIF=0$ 或读取 $TIM2_CCRx$ 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 $TIM2_CCR1$ 寄存器中，步骤如下：

1. 配置 $TIM2_CCMR1$ 寄存器的 $CC1S=01$ ， $CC1$ 通道被配置为输入， $IC1$ 映射在 $TI1$ 上。
2. 配置 $TIM2_CCMR1$ 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 $TIM2_CCER$ 寄存器的 $CC1P=0$ ，选择捕获发生在 $TI1$ 信号的上升沿。
4. 配置 $TIM2_CCMR1$ 寄存器的 $IC1PSC[1: 0]$ ，选择预分频系数。
5. 配置 $TIM2_CCER$ 寄存器的 $CC1E = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 $TIM2_DIER$ 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 $TIM2_DIER$ 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， $TIM2_CCRx$ 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 $TIM2_EGR$ 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。

18.4.2.2 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 Tix 输入。
- 配置从模式为复位模式，将其中一路 TixFP 作为触发输入信号。

例：测量 TI1 的 PWM 信号的宽度（TIM2_CCR1 寄存器）和占空比（TIM2_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 TIM2_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 TIM2_CCMR1 寄存器的 CC1S = 01，将 IC1 映射在 TI1 上，选择 TIM2_CCR1 的有效输入。
3. 配置 TIM2_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 TIM2_CCR1 中并清除计数器）。
4. 配置 TIM2_CCMR1 寄存器的 CC2S = 10，将 IC2 映射在 TI1 上，选择 TIM2_CCR2 的有效输入。
5. 配置 TIM2_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 TIM2_CCR2 中）。
6. 配置 TIM2_SMCR 寄存器中的 TS = 101，选择 TI1FP1 为有效的触发输入信号。
7. 配置 TIM2_SMCR 中的 SMS = 100，从模式控制器设置为复位模式。
8. 配置 TIM2_CCER 寄存器中 CC1E=1 且 CC2E = 1。开启 CC1 通道和 CC2 通道的捕获使能。

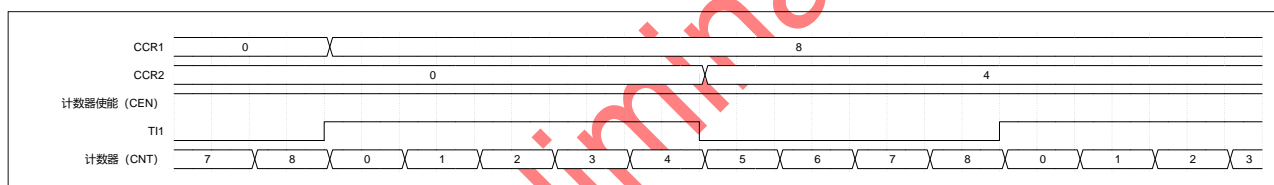


图 18-13 PWM 输入模式时序

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 TIM2_CH1/TIM2_CH2 端口输入信号。

18.4.3 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：

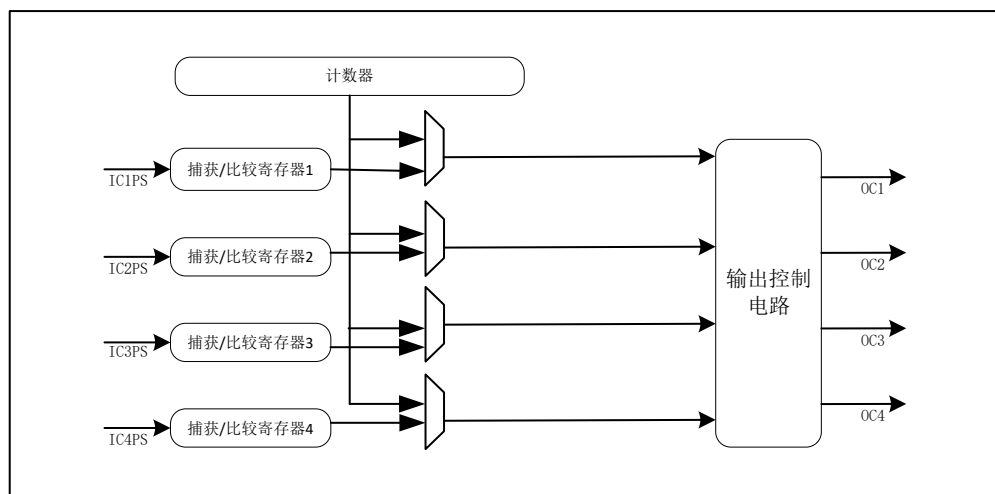


图 18-14 比较输出部分结构图

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。

18.4.3.1 强制输出

配置 TIM2_CCMRx 寄存器的 CCxS = 00，将通道 CCx 设置为输出模式，通过配置 TIM2_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 TIM2_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 TIM2_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 TIM2_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。

18.4.3.2 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 TIM2_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. 匹配时中断状态寄存器中的标志位置 1（TIM2_SR 寄存器中的 CCxIF 位）。

3. 当配置了 TIM2_DIER 寄存器中的 CCxIE =1，匹配时则产生一个中断。
4. 当配置了 TIM2_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 TIM2_ARR 和 TIM2_CCR1 寄存器。
3. 配置 TIM2_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 TIM2_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIM2_CCMR1 寄存器的 OC1PE = 0，禁止 TIM2_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIM2_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIM2_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 TIM2_CR1 寄存器的 CEN =1，启动计数器。

当配置 TIM2_CCMRx 寄存器中 OCxPE=0，禁止 TIM2_CCRx 寄存器的预装载功能时，可以随时写入 TIM2_CCRx 寄存器，并且写入的值立即生效。当配置 TIM2_CCMRx 寄存器中 OCxPE=1，启用 TIM2_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，TIM2_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。

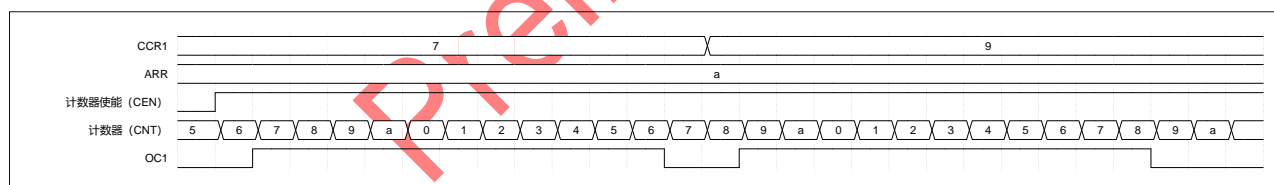


图 18-15 比较输出模式，OC1 信号在匹配时翻转

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 TIM2_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。

18.4.3.3 PWM 输出

在 PWM 模式下，根据 TIM2_ARR 寄存器和 TIM2_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 TIM2_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 TIM2 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 TIM2_CCRx 的预装载功能和 TIM2_ARR 寄存器的预装载功能。写入 TIM2_CCRx 预装载寄存器和

TIM2_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 TIM2_EGR 的 UG=1,产生更新事件用于初始化所有的寄存器。

配置 TIM2_CCER 寄存器的 CCxP 位选择 OCx 的有效极性。配置 TIM2_CCER 寄存器的 CCxE 位控制 OCx 的输出使能。配置 TIM2_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。

18.4.3.3.1 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 TIM2_CCMRx 寄存器的 CCxS=00,选择输出模式,OCxM=110,选择 PWM 模式 1,当 $TIM2_CNT < TIM2_CCR_x$ 时通道 x (OCxREF) 为有效电平,否则为无效电平。如果 TIM2_CCRx 中的比较值大于自动重装载值 (TIM2_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。

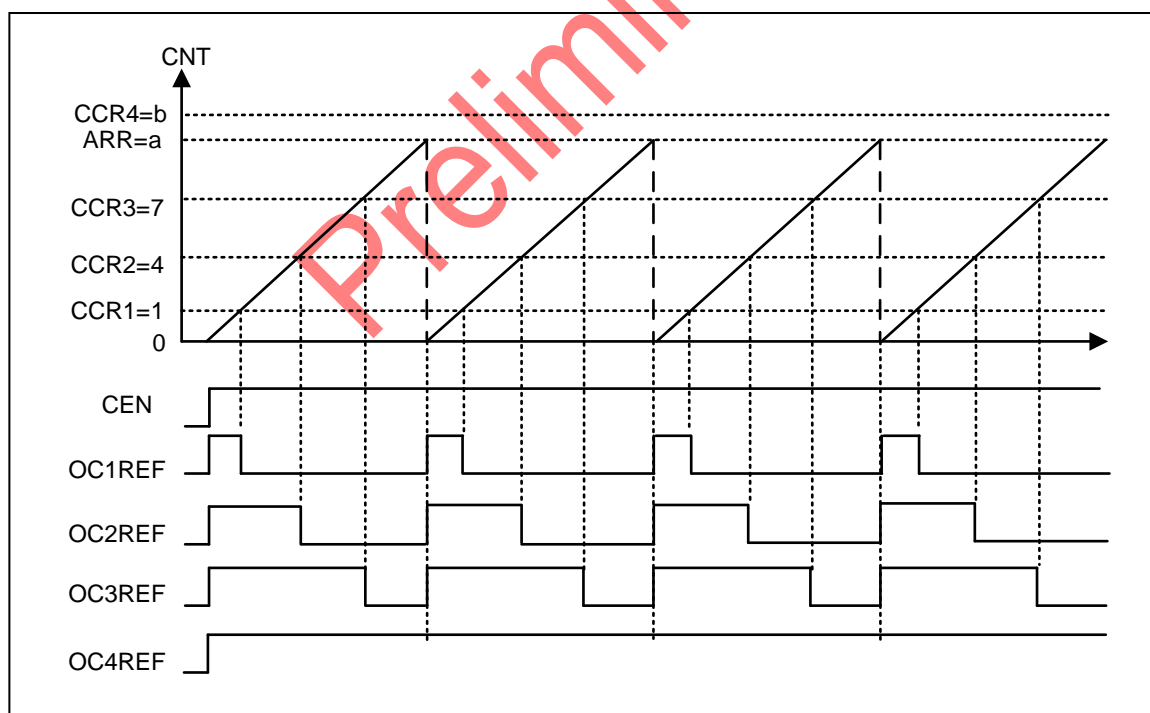


图 18-16 边沿对齐递增计数时 PWM 模式 1 的波形

18.4.3.3.2 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 TIM2_CCMRx 寄存器的 CCxS=00,选择输出模式,OCxM=110,

选择 PWM 模式 1，当 $TIM2_CNT > TIM2_CCR_x$ 时通道 x (OC_xREF) 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。

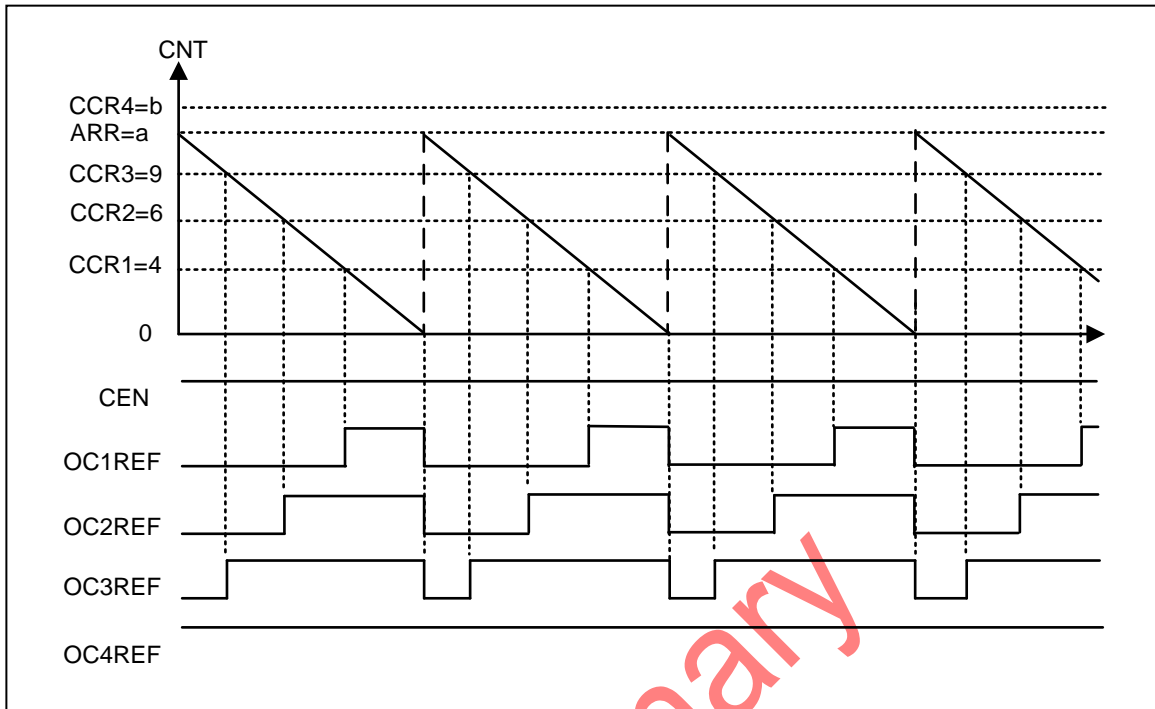


图 18-17 边沿对齐递减计数时 PWM 模式 1 的波形

18.4.3.3.3 PWM 中央对齐模式

首先配置 TIM2 计数器为中央对齐计数模式，配置 $TIM2_CCMR_x$ 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。

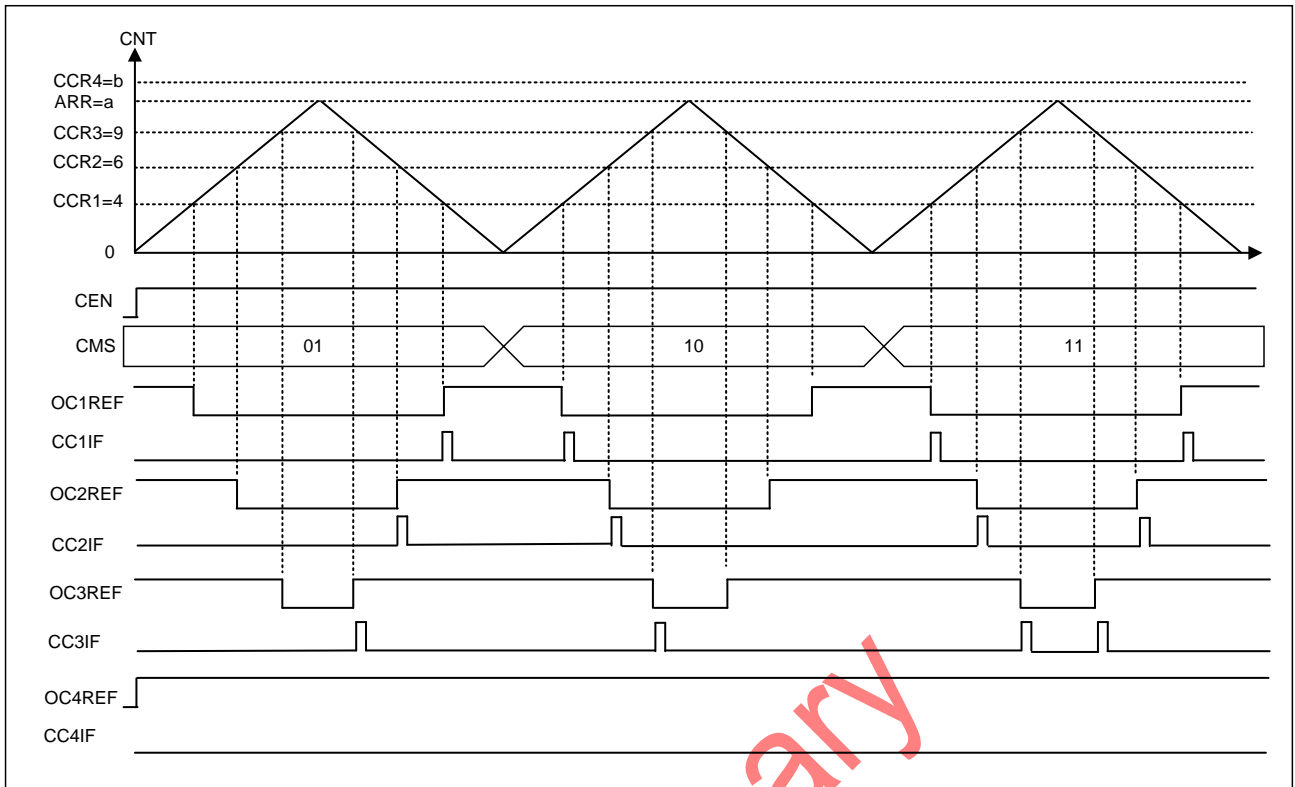


图 18-18 中央对齐 PWM 模式 1 的波形

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 $> \text{TIM2_ARR}$ ，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 TIM2_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。

18.4.3.4 外部事件清除 OCxREF

在配置 TIM2_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 TIM2_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
2. 配置 TIM2_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
3. 配置 TIM2_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）。

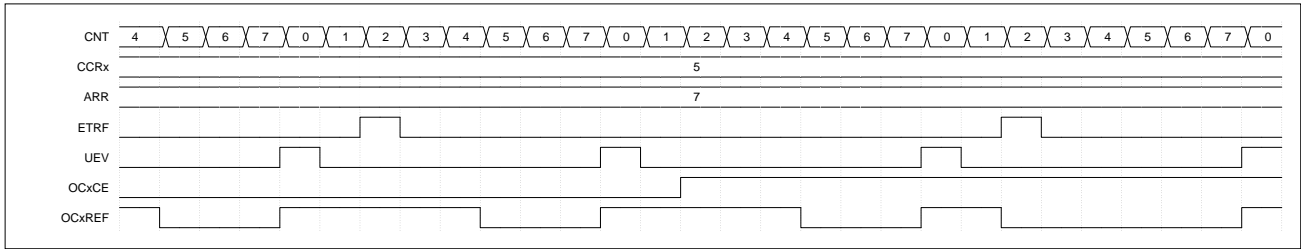


图 18-19 外部事件清除 OCxREF

18.4.3.5 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 TIM2_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 $CNT > CCRx$ 。

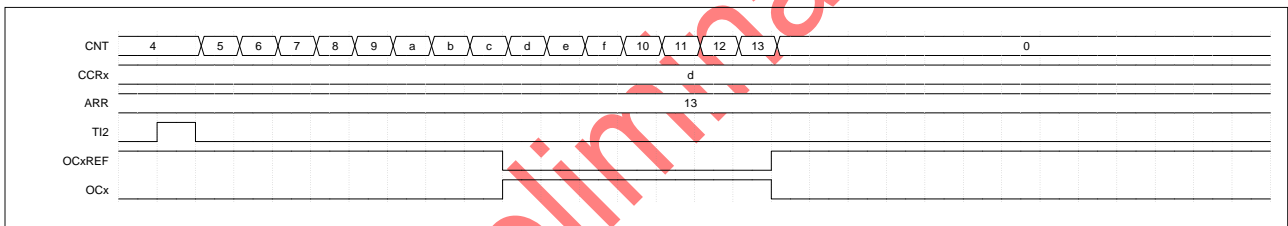


图 18-20 单脉冲模式

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 TIM2_CCMR1 寄存器中的 CC2S = 01，将 TI2FP2 映射到 TI2。
2. 配置 TIM2_CCER 寄存器中的 CC2P = 0，检测 TI2FP2 的上升沿。
3. 配置 TIM2_SMCR 寄存器中的 TS = 110，TI2FP2 作为从模式控制器的触发（TRGI）。
4. 配置 TIM2_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 TIM2_ARR 和 TIM2_CCR1 决定（要考虑时钟频率和计数器预分频器）：由 TIM2_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} ，TIM2_ARR - TIM2_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 TIM2_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
2. 配置 TIM2_CCER 寄存器 CC1P = 1，输出低电平有效。
3. 配置 TIM2_CCMR1 中 OC1PE = 1 和 TIM2_CR1 寄存器中 ARPE=1，使能预装载寄存器。

4. 配置 TIM2_CCR1 寄存器和 TIM2_ARR 寄存器。
5. 配置 TIM2_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，TIM2_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。

18.4.3.5.1 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 TIM2_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。

18.4.4 从模式

18.4.4.1 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 TIM2_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 TIM2_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 18-2 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

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）	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

下列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 TIM2_CCMR 寄存器的 CC1S=01，将 IC1FP1 映射到 TI1 上。
2. 配置 TIM2_CCMR 寄存器的 CC2S=01，将 IC2FP2 映射到 TI2 上。
3. 配置 TIM2_CCER 寄存器的 CC1P=0，IC1 不反相，此时 IC1=TI1。
4. 配置 TIM2_CCER 寄存器的 CC2P=0，IC2 不反相，此时 IC1=TI2。
5. 配置 TIM2_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 TIM2_CR1 寄存器的 CEN=1，开启计数器。

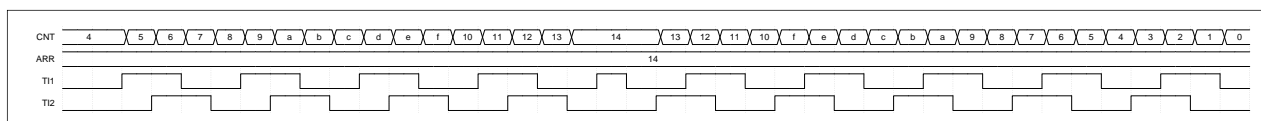


图 18-21 编码器模式下的计数器时序图

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

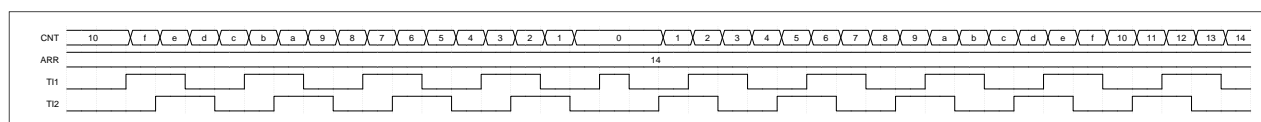


图 18-22 IC1FP1 反相编码器接口模式时序图

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。

18.4.4.2 复位模式

配置 TIM2_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 TIM2_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 TIM2_CCER 寄存器的 CC2P=1，检测下降沿。

2. 配置 TIM2_SMCR 寄存器的 SMS = 100，从模式选择复位模式；配置 TIM2_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 TIM2_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 TIM2_ARR = 0x13 的时序图。

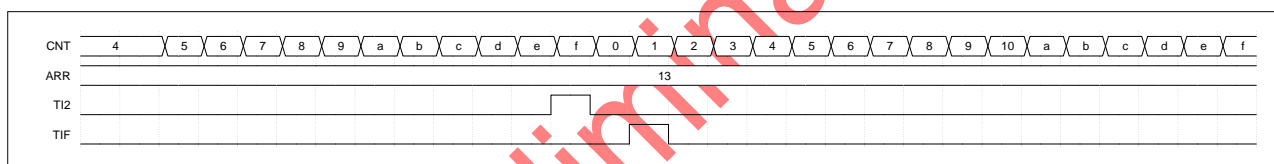


图 18-23 复位模式的控制时序图

18.4.4.3 门控模式

配置 TIM2_SMCR 寄存器 SMS=101，从模式选择门控模式。此模式下，根据 TIM2_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 TIM2_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 TIM2_CCER 寄存器的 CC1P=0，检测 TI1 上的高电平。

2. 配置 TIM2_SMCR 寄存器的 SMS=101，从模式选择为门控模式；配置 TIM2_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 TIM2_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。计数器停止会将 TIF 置 1。

下图为门控模式下 TIM2_ARR=0xf 的时序图。

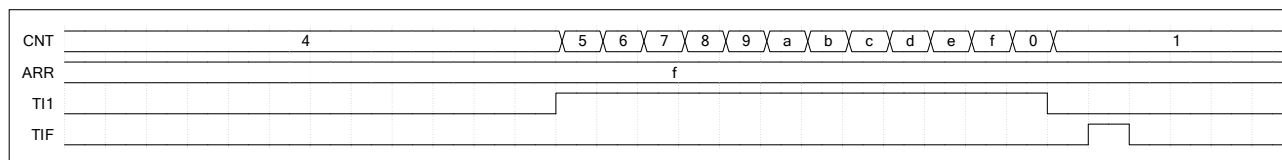


图 18-24 门控模式下的控制时序图

18.4.4.4 触发模式

配置 TIM2_SMCR 寄存器 SMS=110，从模式选择触发模式。根据 TIM2_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 TIM2_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 TIM2_CCER 寄存器的 CC1P=0，检测上升沿。
 2. 配置 TIM2_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIM2_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 TIM2_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 TIM2_ARR=0xf 的时序图。

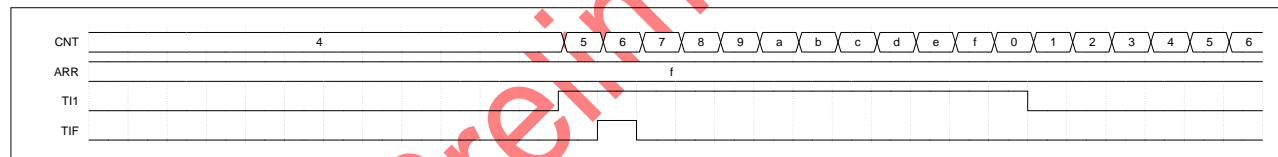


图 18-25 触发器模式下的控制时序图

18.4.4.5 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 TIM2_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIM2_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIM2_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIM2_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。
2. 配置 TIM2_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 TIM2_CCER 寄存器的 CC1P=0，选择上升沿有效。
3. 配置 TIM2_SMCR 寄存器的 SMS = 110，从模式选择为触发模式。配置 TIM2_SMCR 寄存器的 TS = 101，选择 TI1 作为输入源。
4. 配置 TIM2_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 TIM2_ARR=13 时的时序图。

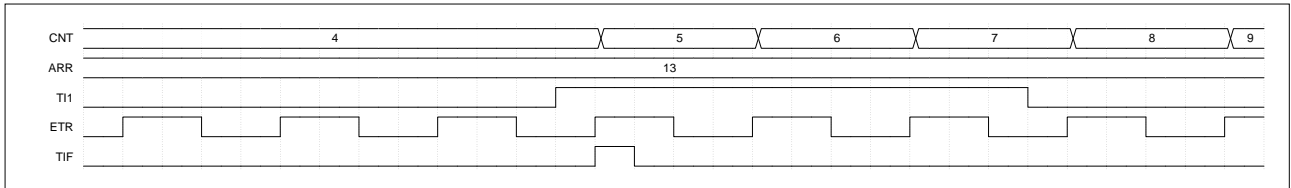


图 18-26 外部时钟模式 2+从模式（触发模式）控制时序图

18.4.5 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。

18.4.6 定时器异或

配置 TIM2_CR2 寄存器的 TI1S =1，将 TIM2_CH1、TIM2_CH2 和 TIM2_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：TIM2_CH1、TIM2_CH2 和 TIM2_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 TIM2_CCR1 寄存器中。具体配置如下：

1. 配置 TIM2_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 TIM2_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 TIM2_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 TIM2_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
5. 配置 TIM2_CCMR1 寄存器的 IC1PSC，选择预分频系数。
6. 配置 TIM2_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。
7. 配置 TIM2_CR1 寄存器的 CEN=1，启动计数器。

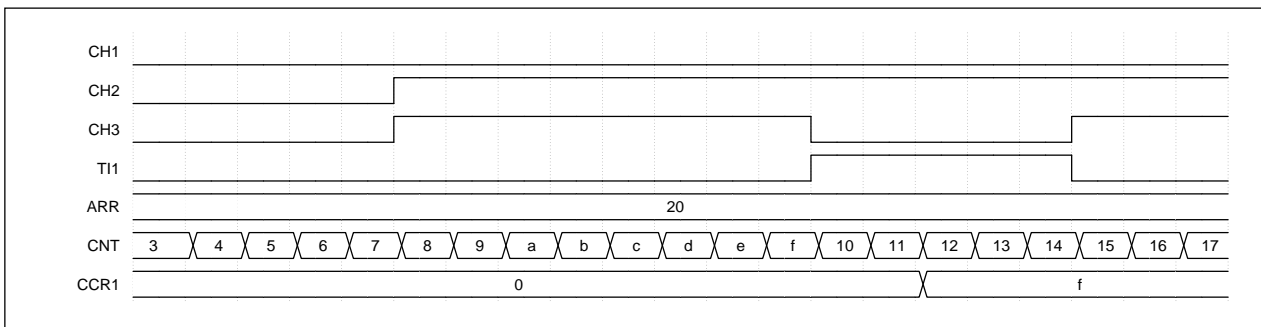


图 18-27 （TI1 异或输入）输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

18.4.7 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIM2_STOP=1，TIM2 计数器停止计数。（详见调试章节）

18.4.8 中断

TIM2 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 18-3 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

18.4.9 DMA

TIM2 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 TIM2 的一部分寄存器，也可以用于按周期读取数个寄存器。

TIM2_DCR 和 TIM2_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 TIM2_DMAR 寄存器。开启 DMA 使能后，在给定的 TIM2 事件发生时，TIM2 会给 DMA 发送请求。对 TIM2_DMAR 寄存器的每次写操作都被重定向到一个 TIM2 寄存器。

TIM2_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 TIM2_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。TIM2_DCR 寄存器的 DBA 位定义了 DMA 传输的基地址，定义从 TIM2_CR1 寄存器地址开始的偏移量（00000 为 TIM2_CR1、00001 为 TIM2_CR2 00110 为 TIM2_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 TIM2_DCR 寄存器的 DBA=01101，配置 DMA 的基地址，选择偏移地址为 TIM2_CCR1 寄存器的地址。
3. 配置 TIM2_DCR 寄存器的 DBL=00010，配置传输长度为 3。
4. 配置 TIM2_DIER 寄存器的 UDE=1，允许更新事件的 DMA 请求。

5. 配置 TIM2_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。

18.5 寄存器描述

表 18-4 TIM2 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIM2_CR1	控制寄存器 1	0x0000
0x04	TIM2_CR2	控制寄存器 2	0x0000
0x08	TIM2_SMCR	从模式控制寄存器	0x0000
0x0C	TIM2_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	TIM2_SR	状态寄存器	0x0000
0x14	TIM2_EGR	事件产生寄存器	0x0000
0x18	TIM2_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIM2_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIM2_CCER	捕获/比较使能寄存器	0x0000
0x24	TIM2_CNT	计数器	0x0000 0000
0x28	TIM2_PSC	预分频率器	0x0000
0x2C	TIM2_ARR	自动装载寄存器	0x0000 0000
0x34	TIM2_CCR1	捕获/比较寄存器 1	0x0000 0000
0x38	TIM2_CCR2	捕获/比较寄存器 2	0x0000 0000
0x3C	TIM2_CCR3	捕获/比较寄存器 3	0x0000 0000
0x40	TIM2_CCR4	捕获/比较寄存器 4	0x0000 0000
0x48	TIM2_DCR	DMA 控制寄存器（仅适用于有内置 DMA 的芯片）	0x0000
0x4C	TIM2_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000
0x50	TIM2_OR	输入选项寄存器	0x0000

18.5.1 TIM2_CR1 控制寄存器 1

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
Res.						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	<p>时钟分频（clock division）</p> <p>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, Tlx）所用的时钟之间的分频比例。</p> <p>00: $t_{DTS} = t_{INT_CK}$</p> <p>01: $t_{DTS} = 2x t_{INT_CK}$</p> <p>10: $t_{DTS} = 4x t_{INT_CK}$</p> <p>11: 保留，不要使用这个配置</p>
7	ARPE	<p>自动重载预装载使能（Auto-reload preload enable）</p> <p>0: 关闭 TIM2_ARR 寄存器的影子寄存器</p> <p>1: 使能 TIM2_ARR 寄存器的影子寄存器</p>
6: 5	CMS	<p>中央对齐模式选择（Center-aligned mode selection）</p> <p>00: 边沿对齐模式。计数方向取决于 DIR 位</p> <p>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</p> <p>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</p> <p>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</p> <p>注：计数过程中，不允许更改此位。</p>
4	DIR	<p>计数方向（Direction）</p> <p>0: 计数器递增计数</p> <p>1: 计数器递减计数</p> <p>注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。</p>
3	OPM	<p>单脉冲模式（One pulse mode）</p> <p>0: 禁止单脉冲模式，在发生更新事件时，计数器继续计数</p> <p>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</p>
2	URS	<p>更新请求源（Update request source）</p> <p>软件配置该位，选择更新事件源。</p> <p>0: 以下事件可产生一个更新中断或 DMA 请求：</p> <ul style="list-style-type: none"> - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 <p>1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求</p>

Bit	Field	Description
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, 计数器和预分频器被初始化; 如果从模式控制器接收到硬件复位, 计数器将被初始化。
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。

18.5.2 TIM2_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.								T11S	MMS			CCDS	Res.			
								rw	rw			rw				

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	T11S	T11 选择 (T11 selection) 0: TIM2_CH1 管脚连到 T11 输入 1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 管脚经异或后作为 T11 输入

Bit	Field	Description
6: 4	MMS	<p>主模式选择 (Master mode selection)</p> <p>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:</p> <p>000: 复位 TIM2_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。</p> <p>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。</p> <p>010: 更新 更新事件被选为 TRGO。</p> <p>011: 捕获/比较脉冲 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。</p> <p>100: 比较 OC1REF 信号被用于作为触发输出 (TRGO)</p> <p>101: 比较 OC2REF 信号被用于作为触发输出 (TRGO)</p> <p>110: 比较 OC3REF 信号被用于作为触发输出 (TRGO)</p> <p>111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)</p>
3	CCDS	<p>DMA 请求源选择 (Capture/compare DMA selection)</p> <p>0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求</p> <p>1: 发生更新事件时, 发送 CCx 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品</p>
2: 0	Reserved	保留, 必须保持复位值。

18.5.3 TIM2_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	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择 ETR 信号的极性。</p> <p>0: 高电平或上升沿有效</p> <p>1: 低电平或下降沿有效</p> <p>注: 仅适用于支持外部触发的产品</p>

Bit	Field	Description
14	ECE	<p>外部时钟使能位 (External clock enable)</p> <p>该位启用外部时钟模式 2。</p> <p>0: 禁止外部时钟模式 2</p> <p>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</p> <p>注 1: 仅适用于支持外部触发的产品。</p> <p>注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。</p> <p>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。</p> <p>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。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须低于 TIM2 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。</p> <p>00: 关闭预分频</p> <p>01: ETRP 频率除以 2</p> <p>10: ETRP 频率除以 4</p> <p>11: ETRP 频率除以 8</p> <p>注: 仅适用于支持外部触发的产品。</p>
11: 8	ETF	<p>外部触发滤波 (External trigger filter)</p> <p>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。</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> <p>注: 仅适用于支持外部触发的产品。</p>

Bit	Field	Description
7	MSM	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用</p> <p>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。</p>
6: 4	TS	<p>触发选择 (Trigger selection)</p> <p>触发输入源选择。</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>更多有关 ITRx 的细节, 参见下表。</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</p> <p>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), 并且产生一个更新事件。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), 并且产生一个更新事件, 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 18-5 TIM2 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	TIM2_TRGO	TIM3_TRGO	TIM17_OC1REF
TIM2	TIM1_TRGO	-	TIM3_TRGO	TIM14_OC1REF
TIM3	TIM1_TRGO	TIM2_TRGO	TIM16_OC1REF	TIM14_OC1REF

18.5.4 TIM2_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
	rw		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	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。

Bit	Field	Description
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

18.5.5 TIM2_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
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.		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。

Bit	Field	Description
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和外部时钟模式外的其它模式时、在 TRGI 输入端检测到有效边沿, 或门控模式下的计数停止边沿) 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 TIM2_CR1.CMS[1: 0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIM2_CNT 的值与 TIM2_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 TIM2_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIM2_CCR1

Bit	Field	Description
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 TIM2_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 TIM2_CR1 寄存器的 UDIS=0、URS=0，当 TIM2_EGR 寄存器的 UG=1 时。 - 若 TIM2_CR1 寄存器的 UDIS=0、URS=0，从模式控制器产生更新事件时。

18.5.6 TIM2_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
Res.									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，TIM2_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。

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, 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。

18.5.7 TIM2_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。

Bit	Field	Description
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 TIM2_SMCR 寄存器的 TS 位选择)</p>
7	OC1CE	<p>通道 1 比较输出清 0 使能 (Output compare 1 clear enable)</p> <p>0: OC1REF 不受 ETR 输入的影响</p> <p>1: 当检测到 ETR 输入有效电平时, OC1REF 清零</p>
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。TIM2_CCR1 与 TIM2_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIM2_CNT 的值与 TIM2_CCR1 的值相同时, 强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIM2_CNT 的值与 TIM2_CCR1 的值相同时, 强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIM2_CCR1=TIM2_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 TIM2_CNT<TIM2_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIM2_CNT > TIM2_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 TIM2_CNT<TIM2_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 TIM2_CNT > TIM2_CCR1 时强制 OC1REF 为高电平, 否则为低电平。</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>

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 TIM2_CCR1 寄存器的预装载功能, 写入 TIM2_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIM2_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_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 (TIM2_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 TIM2_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 TIM2_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 $CC1E=0$ ($TIM2_CCER$ 寄存器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
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 $TIM2_SMCR$ 寄存器的 TS 位选择)</p>

18.5.8 TIM2_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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

比较输出模式：

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	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 TIM2_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

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode)</p> <p>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。</p> <p>000: 冻结。TIM2_CCR3 与 TIM2_CNT 间的比较结果对 OC3REF 不起作用</p> <p>001: 匹配时设置为高。当 TIM2_CNT 的值与 TIM2_CCR3 的值相同时, 强制 OC3REF 为高电平</p> <p>010: 匹配时设置为低。当 TIM2_CNT 的值与 TIM2_CCR3 的值相同时, 强制 OC3REF 为低电平</p> <p>011: 匹配时翻转。当 TIM2_CCR3=TIM2_CNT 时, 翻转 OC3REF 的电平</p> <p>100: 强制为低。强制 OC3REF 为低电平</p> <p>101: 强制为高。强制 OC3REF 为高电平</p> <p>110: PWM 模式 1。在递增计数时, 当 TIM2_CNT < TIM2_CCR3 时强制 OC3REF 为高电平, 否则为低电平; 在递减计数时, 当 TIM2_CNT > TIM2_CCR3 时强制 OC3REF 为低电平, 否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时, 当 TIM2_CNT < TIM2_CCR3 时强制 OC3REF 为低电平, 否则为高电平; 在递减计数时, 当 TIM2_CNT > TIM2_CCR3 时强制 OC3REF 为高电平, 否则为低电平。</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, OC3REF 电平才改变。</p>
3	OC3PE	<p>通道 3 比较输出预装载使能 (Output compare 3 preload enable)</p> <p>0: 禁止 TIM2_CCR3 寄存器的预装载功能, 写入 TIM2_CCR3 寄存器的数值立即生效</p> <p>1: 开启 TIM2_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_CCR3 的预装载值在更新事件到来时生效</p> <p>注: 仅在单脉冲模式下 (TIM2_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。</p>
2	OC3FE	<p>通道 3 比较输出快速使能 (Output compare 3 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 3 比较输出快速使能</p> <p>1: 开启通道 3 比较输出快速使能</p>

Bit	Field	Description
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 3 被配置为输出</p> <p>01: 通道 3 被配置为输入, IC3 映射在 TI3 上</p> <p>10: 通道 3 被配置为输入, IC3 映射在 TI4 上</p> <p>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 TIM2_SMCR 寄存器的 TS 位选择)</p>

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	<p>输入捕获 4 滤波器 (Input capture 4 filter)</p> <p>参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler)</p> <p>参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>通道 4 捕获/比较选择 (Capture/Compare 4 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 4 被配置为输出</p> <p>01: 通道 4 被配置为输入, IC4 映射在 TI4 上</p> <p>10: 通道 4 被配置为输入, IC4 映射在 TI3 上</p> <p>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 TIM2_SMCR 寄存器的 TS 位选择)</p>

Bit	Field	Description
7: 4	IC3F	<p>通道 3 输入捕获滤波器 (Input capture 3 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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler)</p> <p>该位定义了 IC3 的预分频系数。当 $CC3E=0$ ($TIM2_CCER$ 寄存器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 3 被配置为输出</p> <p>01: 通道 3 被配置为输入, IC3 映射在 TI3 上</p> <p>10: 通道 3 被配置为输入, IC3 映射在 TI4 上</p> <p>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 $TIM2_SMCR$ 寄存器的 TS 位选择)</p>

18.5.9 TIM2_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。
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。

Bit	Field	Description
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 18-6 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/高电平有效
0	1	保留

18.5.10 TIM2_CNT 计数器

偏移地址：0x24

复位值：0x0000 0000

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

Bit	Field	Description
31: 0	CNT	计数器的值 (Count value)

18.5.11 TIM2_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。

18.5.12 TIM2_ARR 自动预装载寄存器

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

Bit	Field	Description
31: 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。

18.5.13 TIM2_CCR1 捕获/比较寄存器 1

偏移地址: 0x34

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
31: 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 TIM2_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 TIM2_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

18.5.14 TIM2_CCR2 捕获/比较寄存器 2

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
CCR2															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
31: 0	CCR2	通道 2 捕获/比较的值 (Capture/Compare 2 value) 参考 CCR1 的描述。

18.5.15 TIM2_CCR3 捕获/比较寄存器 3

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

Bit	Field	Description
31: 0	CCR3	通道 3 捕获/比较的值 (Capture/Compare 3 value) 参考 CCR1 的描述。

18.5.16 TIM2_CCR4 捕获/比较寄存器 4

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

Bit	Field	Description
31: 0	CCR4	通道 4 捕获/比较的值 (Capture/Compare 4 value) 参考 CCR1 的描述。

18.5.17 TIM2_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 18.4.9 DMA 章节。

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

Bit	Field	Description
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 TIM2_DMAR 寄存器的第一个地址。DBA 定义为从 TIM2_CR1 寄存器所在地址开始的偏移值： 00000: TIM2_CR1 00001: TIM2_CR2 00010: TIM2_SMCR

18.5.18 TIM2_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 18.4.9 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 TIM2_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： TIM2_CR1 地址 + DBA + DMA 索引, 其中 TIM2_CR1 地址是 TIM2_CR1 寄存器所在的地址, DBA 是 TIM2_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 TIM2_DCR 寄存器中定义的 DBL 值。

18.5.19 TIM2_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
Res.								T14_RMP		Res.				ETR_RMP	
								rw	rw					rw	rw

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。

Bit	Field	Description
7: 6	TI4_RMP	TI4 复用 00: CH4 GPIO 或比较器输入 01: LSI 时钟输入 10: 保留 11: HSE_CLK_DIV_128 时钟输入
5: 2	Reserved	保留，必须保持复位值。
1: 0	ETR_RMP	ETR 复用 00: ETR GPIO 输入 其他保留

Preliminary

19 TIM3 通用定时器

19.1 简介

TIM3 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）。

19.2 功能框图

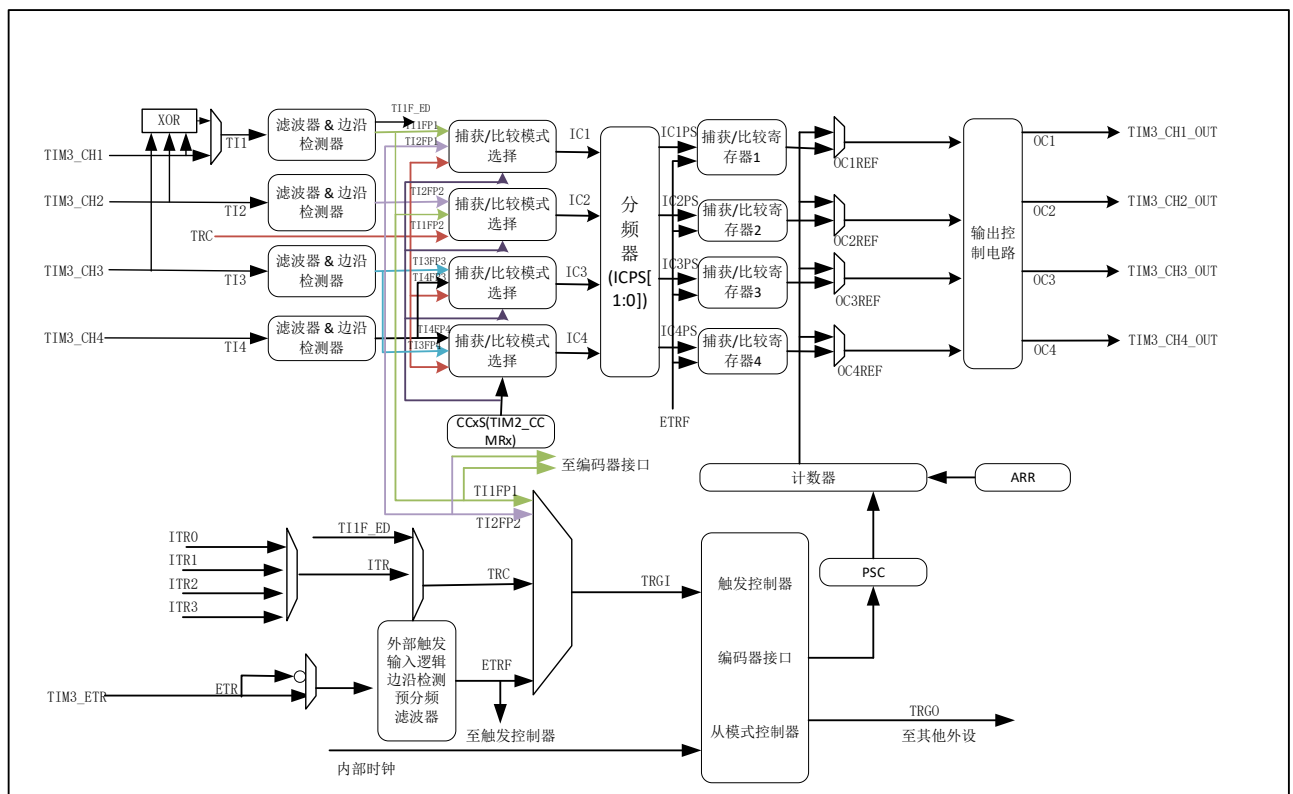


图 19-1 TIM3 结构图

上图为 TIM3 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。

19.3 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（TIx、ETRx），内部触发输入（ITRx）
- 16 位自动重载计数器（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

19.4 功能描述

19.4.1 时钟

19.4.1.1 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 TIX、ITRx、ETRx）
- 外部时钟模式 2：外部触发输入 ETR（ETRx）
- 编码器模式

上述几种时钟选择示意图如下：

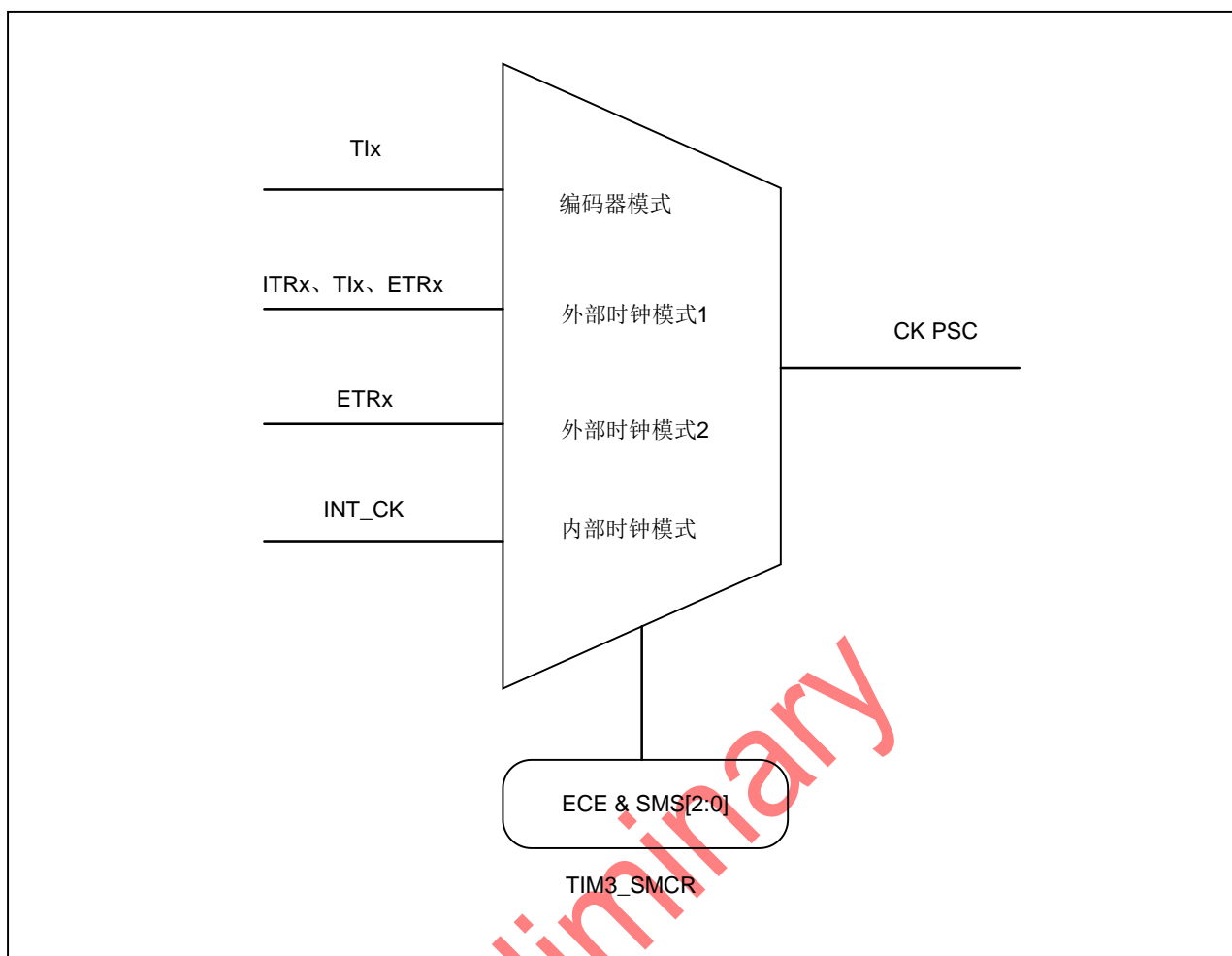


图 19-2 时钟选择

19.4.1.1.1 内部时钟源 (INT_CK)

当配置 TIM3_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。

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

当配置 TIM3_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 TIM3_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上；配置 TIM3_CCMR1 寄存器的 IC1F[3: 0]，设置输入滤波器带宽；配置 TIM3_CCER 寄存器 CC1P=0，选择上升沿为有效沿。
2. 配置 TIM3_SMCR 寄存器的 TS=101，选择 TI1 的作为触发输入源；配置 TIM3_SMCR 寄存器的 SMS=111，选择外部时钟模式 1。

3. 配置 TIM3_CR1 寄存器的 DIR=0, 选择递增计数模式, 配置 TIM3_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。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。

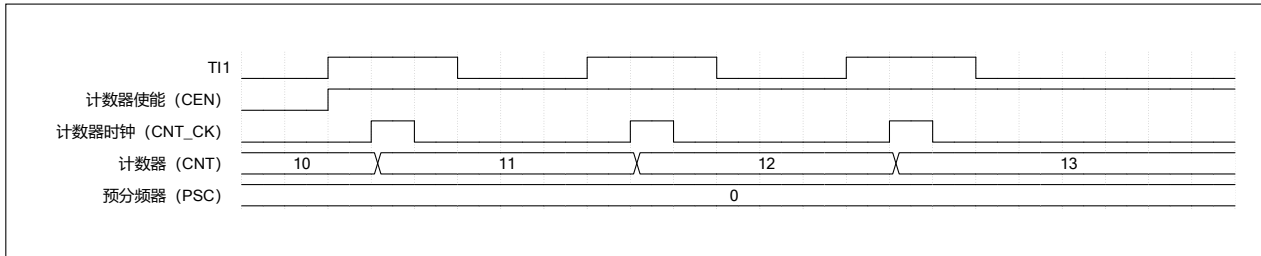


图 19-3 外部时钟模式 1 下的控制电路

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

当配置 TIM3_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 TIM3_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 TIM3_SMCR 寄存器的 ETP=1, 选择下降沿有效; 配置 TIM3_SMCR 寄存器的 ECE=1, 选择外部时钟模式 2。

2. 配置 TIM3_CR1 寄存器的 DIR=0, 选择递增计数模式; 配置 TIM3_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。

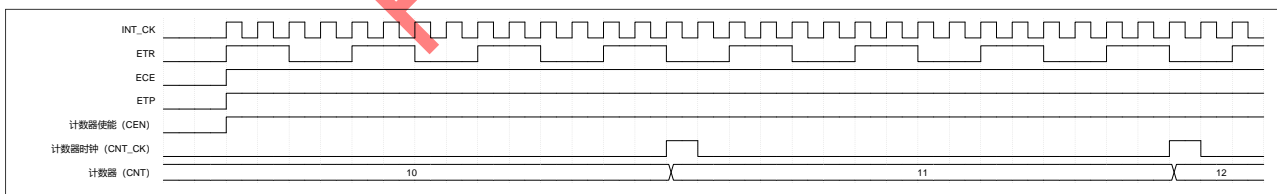


图 19-4 外部时钟模式 2 下的控制电路

19.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

19.4.1.2 时基单元

TIM3 的时基单元主要包括: 计数器寄存器 (TIM3_CNT)、预分频器寄存器 (TIM3_PSC) 和自动预装载寄存器 (TIM3_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 TIM3_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。

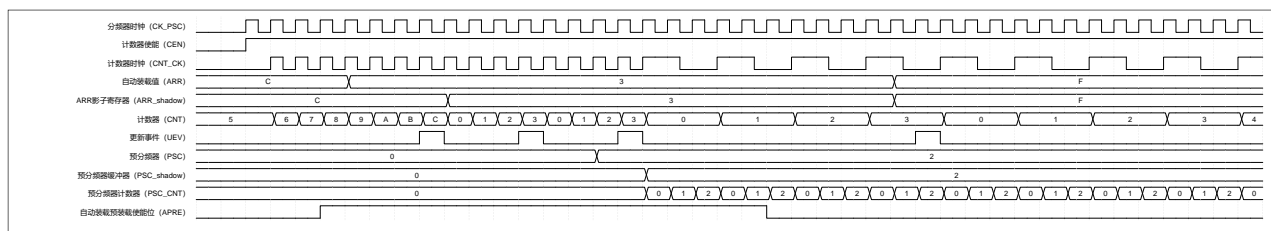


图 19-5 自动预装载

19.4.1.3 计数模式

通过配置 TIM3_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。

19.4.1.3.1 递增计数模式

配置 TIM3_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 TIM3_CR1 寄存器的 CEN 后计数器由 0 开始递增计数，直至 TIM3_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 TIM3_EGR 寄存器的 UG=1，同样可以产生一个更新事件。

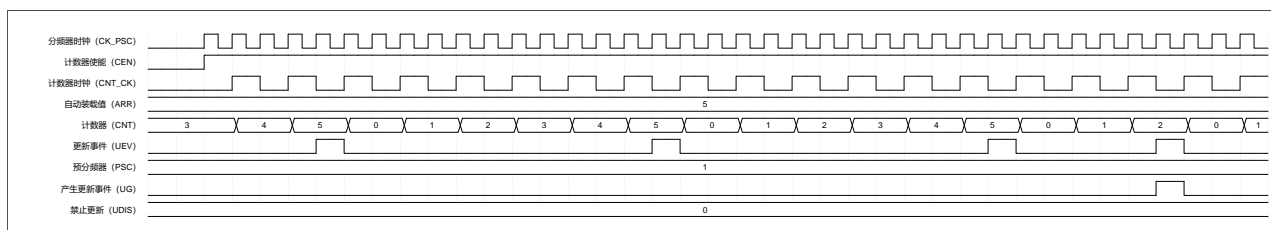


图 19-6 递增计数模式（UDIS=0）

通过配置 TIM3_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。

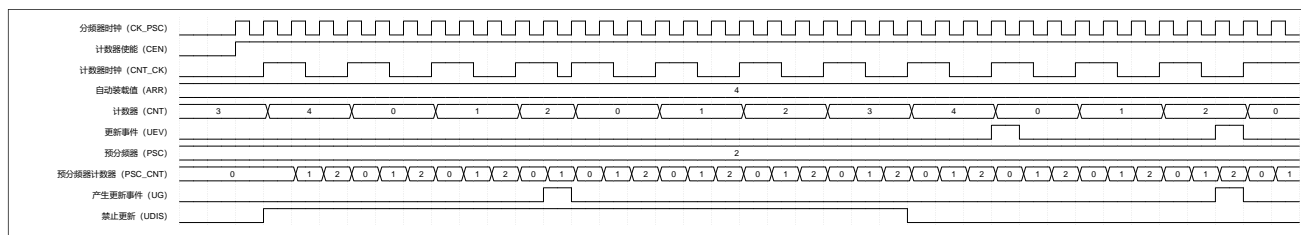


图 19-7 递增计数模式 (UDIS=1 禁止产生更新事件)

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。

19.4.1.3.2 递减计数模式

配置 TIM3_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 TIM3_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 TIM3_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 TIM3_ARR 开始重新递减计数（TIM3_CR1 寄存器 UDIS=0）。

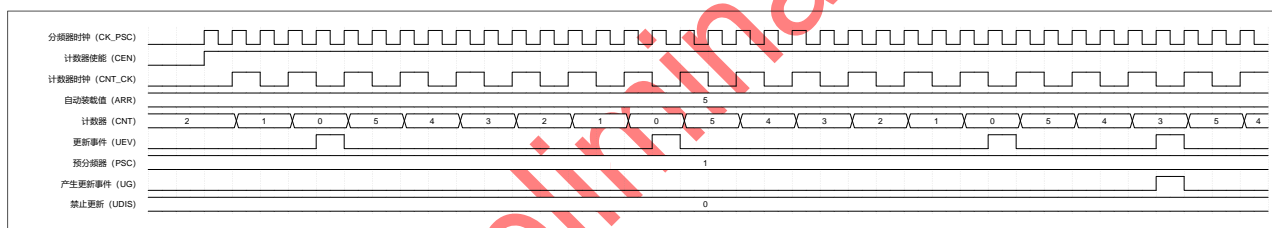


图 19-8 递减计数模式 (UDIS=0)

通过配置 TIM3_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 TIM3_ARR 开始计数。

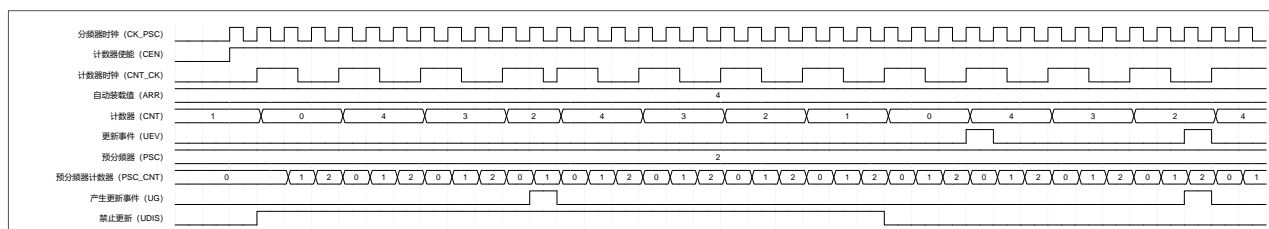


图 19-9 递减计数模式 (UDIS=1 禁止产生更新事件)

19.4.1.3.3 中央计数模式（递增/递减计数模式）

配置 TIM3_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 TIM3_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（TIM3_CR1 寄存器 UDIS=0）。

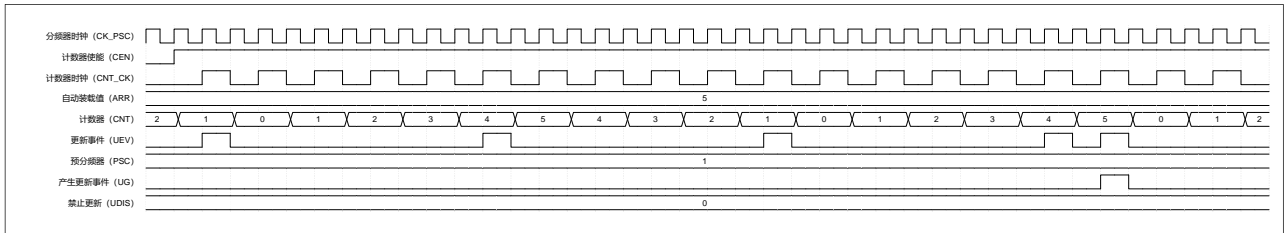


图 19-10 中央计数模式（UDIS=0）

通过配置 TIM3_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。

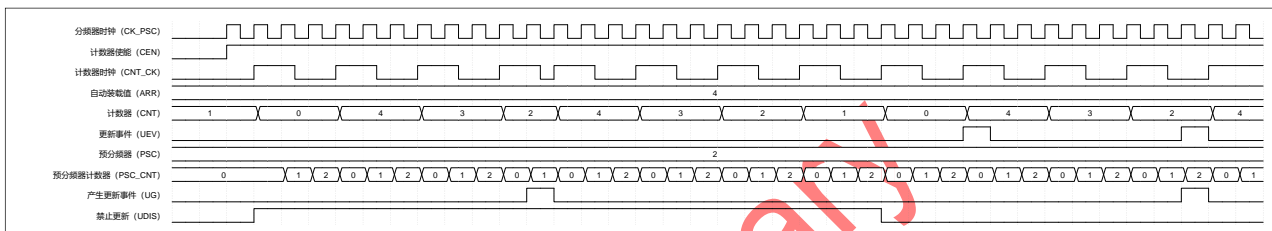


图 19-11 中央计数模式（UDIS=1 禁止产生更新事件）

19.4.2 输入捕获

19.4.2.1 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：

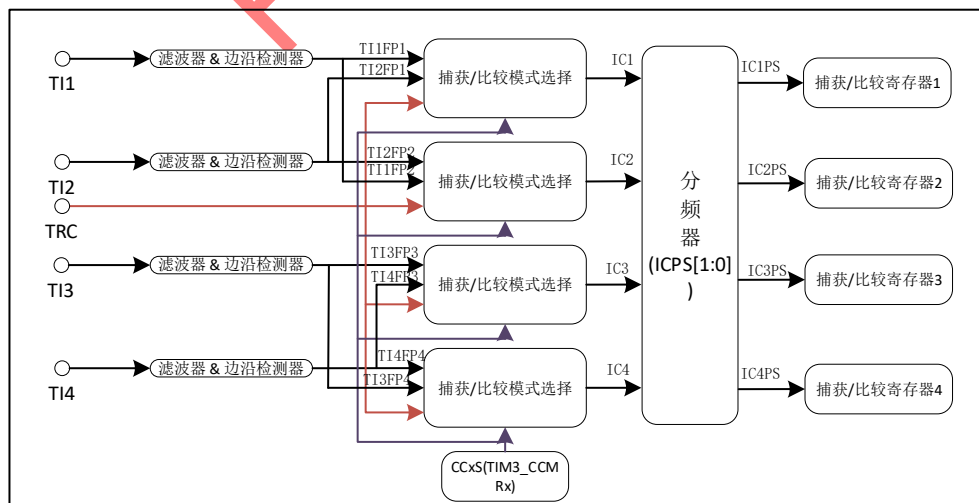


图 19-12 TIM3 输入捕获结构图

通过配置 TIM3_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 19-1 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 ($TIM3_SR$) 中的捕获标志位 $CCxIF$ 置 1，通过配置 $CCxIF=0$ 或读取 $TIM3_CCRx$ 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 $TIM3_CCR1$ 寄存器中，步骤如下：

1. 配置 $TIM3_CCMR1$ 寄存器的 $CC1S=01$ ， $CC1$ 通道被配置为输入， $IC1$ 映射在 $TI1$ 上。
2. 配置 $TIM3_CCMR1$ 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 $TIM3_CCER$ 寄存器的 $CC1P=0$ ，选择捕获发生在 $TI1$ 信号的上升沿。
4. 配置 $TIM3_CCMR1$ 寄存器的 $IC1PSC[1: 0]$ ，选择预分频系数。
5. 配置 $TIM3_CCER$ 寄存器的 $CC1E = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 $TIM3_DIER$ 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 $TIM3_DIER$ 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， $TIM3_CCRx$ 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 $TIM3_EGR$ 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。

19.4.2.2 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 Tix 输入。
- 配置从模式为复位模式，将其中一路 TixFP 作为触发输入信号。

例：测量 TI1 的 PWM 信号的宽度（TIM3_CCR1 寄存器）和占空比（TIM3_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 TIM3_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 TIM3_CCMR1 寄存器的 CC1S = 01，将 IC1 映射在 TI1 上，选择 TIM3_CCR1 的有效输入。
3. 配置 TIM3_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 TIM3_CCR1 中并清除计数器）。
4. 配置 TIM3_CCMR1 寄存器的 CC2S = 10，将 IC2 映射在 TI1 上，选择 TIM3_CCR2 的有效输入。
5. 配置 TIM3_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 TIM3_CCR2 中）。
6. 配置 TIM3_SMCR 寄存器中的 TS = 101，选择 TI1FP1 为有效的触发输入信号。
7. 配置 TIM3_SMCR 中的 SMS = 100，从模式控制器设置为复位模式。
8. 配置 TIM3_CCER 寄存器中 CC1E=1 且 CC2E = 1。开启 CC1 通道和 CC2 通道的捕获使能。

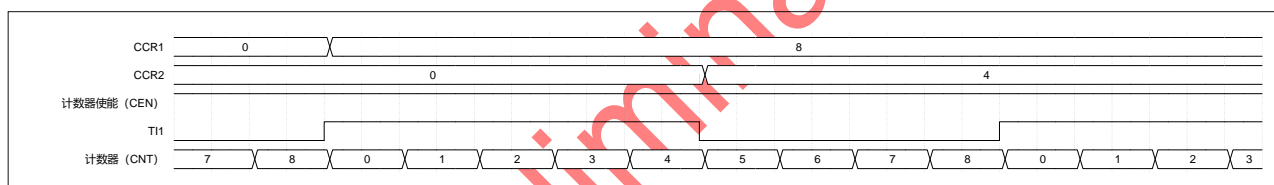


图 19-13 PWM 输入模式时序

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 TIM3_CH1/TIM3_CH2 端口输入信号。

19.4.3 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：

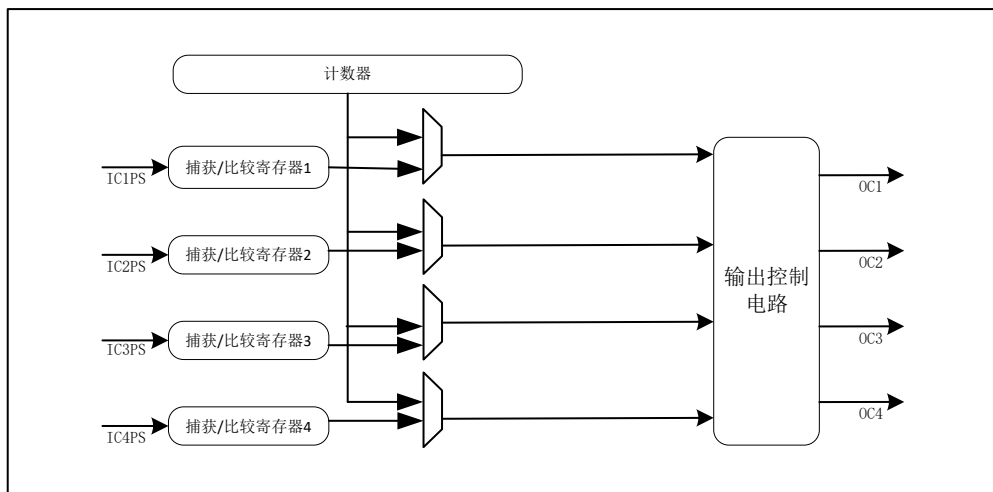


图 19-14 比较输出部分结构图

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。

19.4.3.1 强制输出

配置 TIM3_CCMRx 寄存器的 CCxS = 00，将通道 CCx 设置为输出模式，通过配置 TIM3_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 TIM3_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 TIM3_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 TIM3_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。

19.4.3.2 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 TIM3_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. 匹配时中断状态寄存器中的标志位置 1（TIM3_SR 寄存器中的 CCxIF 位）。

3. 当配置了 TIM3_DIER 寄存器中的 CCxIE =1，匹配时则产生一个中断。
4. 当配置了 TIM3_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 TIM3_ARR 和 TIM3_CCR1 寄存器。
3. 配置 TIM3_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 TIM3_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIM3_CCMR1 寄存器的 OC1PE = 0，禁止 TIM3_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIM3_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIM3_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 TIM3_CR1 寄存器的 CEN =1，启动计数器。

当配置 TIM3_CCMRx 寄存器中 OCxPE=0，禁止 TIM3_CCRx 寄存器的预装载功能时，可以随时写入 TIM3_CCRx 寄存器，并且写入的值立即生效。当配置 TIM3_CCMRx 寄存器中 OCxPE=1，启用 TIM3_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，TIM3_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。

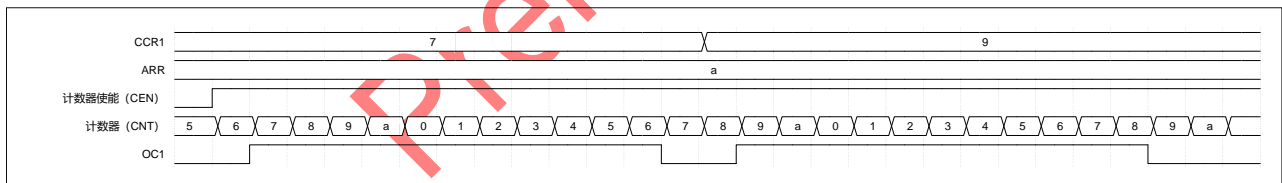


图 19-15 比较输出模式，OC1 信号在匹配时翻转

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 TIM3_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。

19.4.3.3 PWM 输出

在 PWM 模式下，根据 TIM3_ARR 寄存器和 TIM3_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 TIM3_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 TIM3 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 TIM3_CCRx 的预装载功能和 TIM3_ARR 寄存器的预装载功能。写入 TIM3_CCRx 预装载寄存器和

TIM3_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 TIM3_EGR 的 UG=1,产生更新事件用于初始化所有的寄存器。

配置 TIM3_CCER 寄存器的 CCxP 位选择 OCx 的有效极性。配置 TIM3_CCER 寄存器的 CCxE 位控制 OCx 的输出使能。配置 TIM3_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。

19.4.3.3.1 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 TIM3_CCMRx 寄存器的 CCxS=00,选择输出模式,OCxM=110,选择 PWM 模式 1,当 $TIM3_CNT < TIM3_CCR_x$ 时通道 x (OCxREF) 为有效电平,否则为无效电平。如果 TIM3_CCRx 中的比较值大于自动重装载值 (TIM3_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。

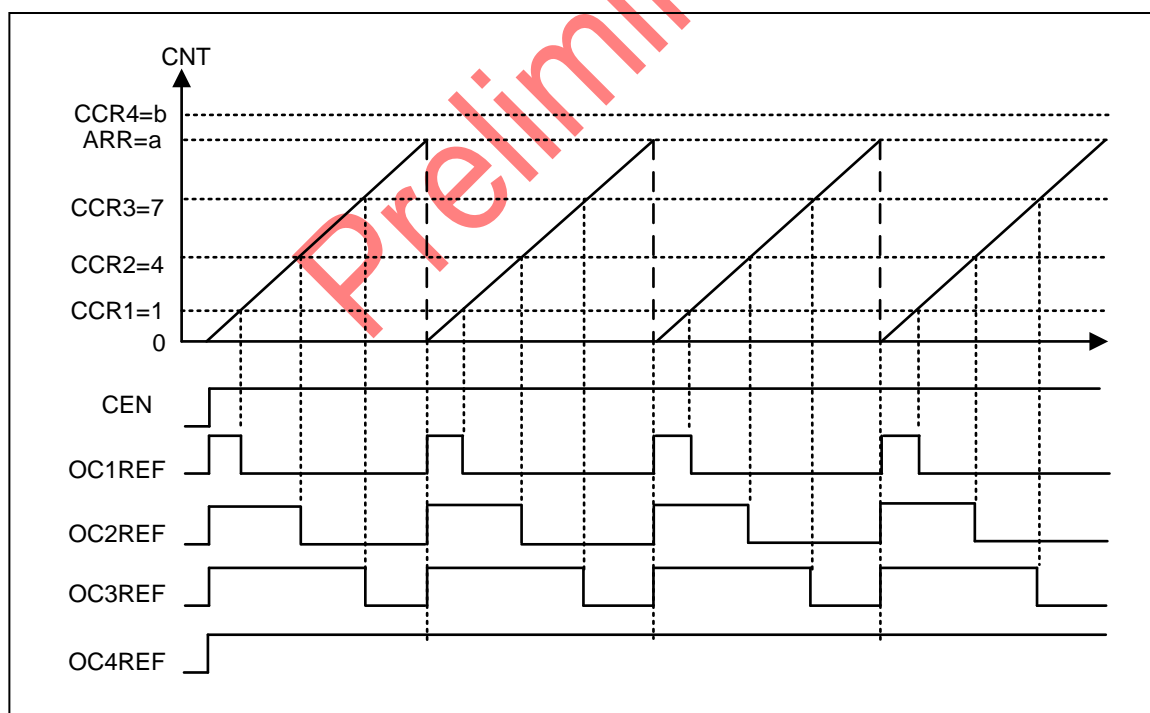


图 19-16 边沿对齐递增计数时 PWM 模式 1 的波形

19.4.3.3.2 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 TIM3_CCMRx 寄存器的 CCxS=00,选择输出模式,OCxM=110,

选择 PWM 模式 1，当 $TIM3_CNT > TIM3_CCR_x$ 时通道 x (OC_xREF) 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。

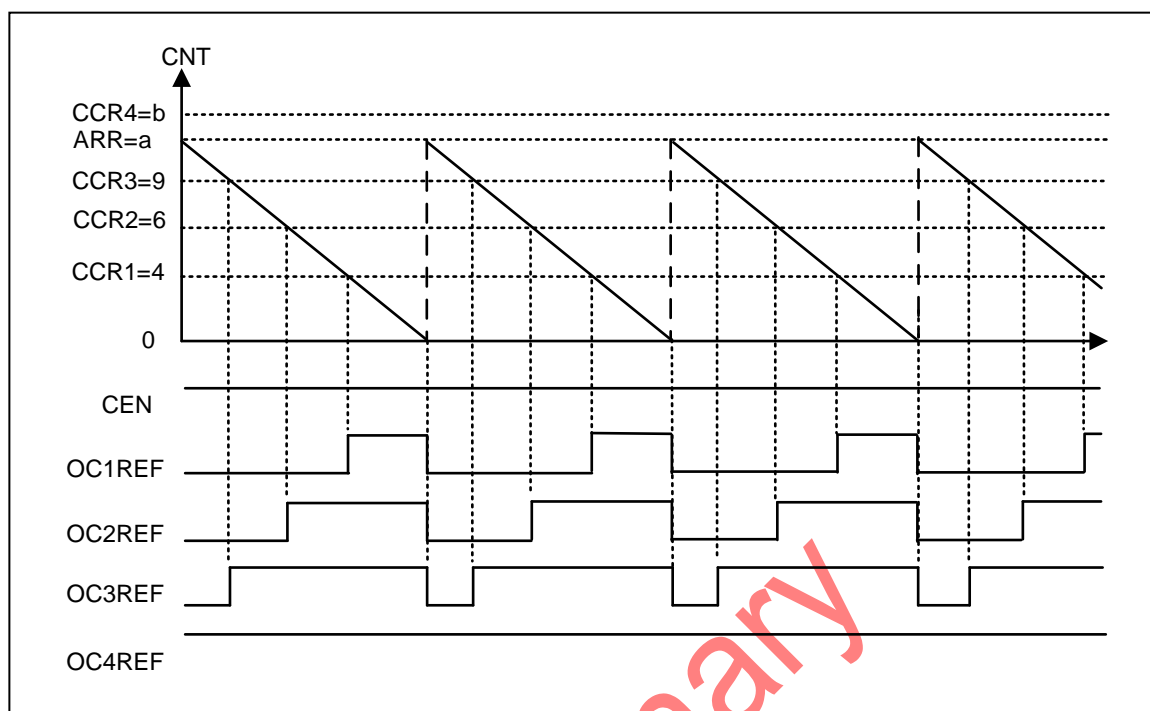


图 19-17 边沿对齐递减计数时 PWM 模式 1 的波形

19.4.3.3.3 PWM 中央对齐模式

首先配置 TIM3 计数器为中央对齐计数模式，配置 $TIM3_CCMR_x$ 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。

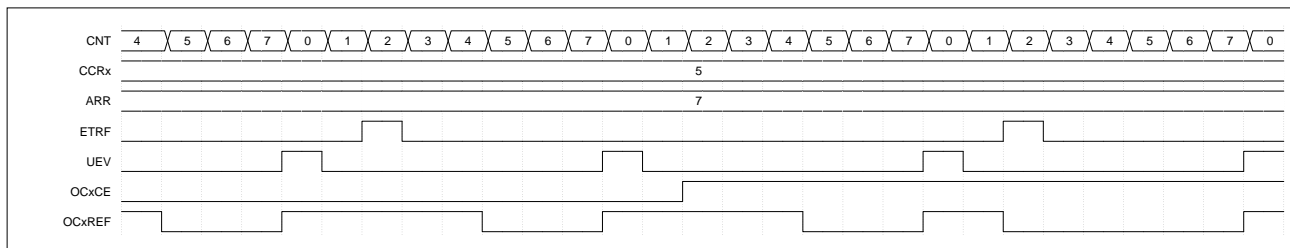


图 19-19 外部事件清除 OCxREF

19.4.3.5 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 TIM3_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 $CNT > CCRx$ 。

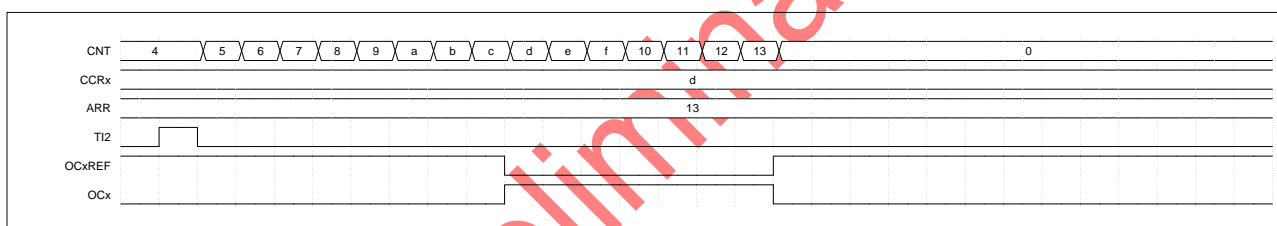


图 19-20 单脉冲模式

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 TIM3_CCMR1 寄存器中的 CC2S = 01，将 TI2FP2 映射到 TI2。
2. 配置 TIM3_CCER 寄存器中的 CC2P = 0，检测 TI2FP2 的上升沿。
3. 配置 TIM3_SMCR 寄存器中的 TS = 110，TI2FP2 作为从模式控制器的触发（TRGI）。
4. 配置 TIM3_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 TIM3_ARR 和 TIM3_CCR1 决定（要考虑时钟频率和计数器预分频器）：由 TIM3_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} ，TIM3_ARR - TIM3_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 TIM3_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
2. 配置 TIM3_CCER 寄存器 CC1P = 1，输出低电平有效。
3. 配置 TIM3_CCMR1 中 OC1PE = 1 和 TIM3_CR1 寄存器中 ARPE=1，使能预装载寄存器。

4. 配置 TIM3_CCR1 寄存器和 TIM3_ARR 寄存器。
5. 配置 TIM3_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，TIM3_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。

19.4.3.5.1 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 TIM3_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。

19.4.4 从模式

19.4.4.1 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 TIM3_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 TIM3_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 19-2 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

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）	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

下列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 TIM3_CCMR 寄存器的 CC1S=01，将 IC1FP1 映射到 TI1 上。
2. 配置 TIM3_CCMR 寄存器的 CC2S=01，将 IC2FP2 映射到 TI2 上。
3. 配置 TIM3_CCER 寄存器的 CC1P=0，IC1 不反相，此时 IC1=TI1。
4. 配置 TIM3_CCER 寄存器的 CC2P=0，IC2 不反相，此时 IC1=TI2。
5. 配置 TIM3_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 TIM3_CR1 寄存器的 CEN=1，开启计数器。

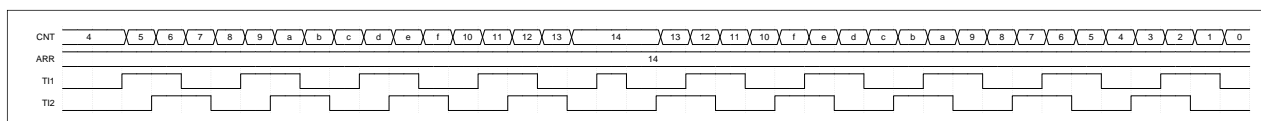


图 19-21 编码器模式下的计数器时序图

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

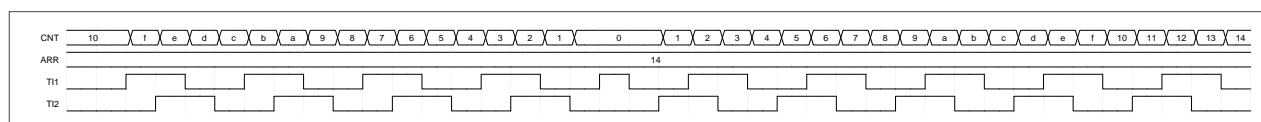


图 19-22 IC1FP1 反相编码器接口模式时序图

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。

19.4.4.2 复位模式

配置 TIM3_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 TIM3_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 TIM3_CCER 寄存器的 CC2P=1，检测下降沿。

2. 配置 TIM3_SMCR 寄存器的 SMS = 100，从模式选择复位模式；配置 TIM3_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 TIM3_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 TIM3_ARR = 0x13 的时序图。

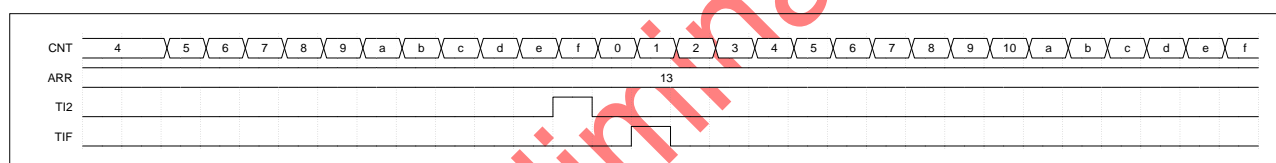


图 19-23 复位模式的控制时序图

19.4.4.3 门控模式

配置 TIM3_SMCR 寄存器 SMS=101，从模式选择门控模式。此模式下，根据 TIM3_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 TIM3_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 TIM3_CCER 寄存器的 CC1P=0，检测 TI1 上的高电平。

2. 配置 TIM3_SMCR 寄存器的 SMS=101，从模式选择为门控模式；配置 TIM3_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 TIM3_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。计数器停止会将 TIF 置 1。

下图为门控模式下 TIM3_ARR=0xf 的时序图。

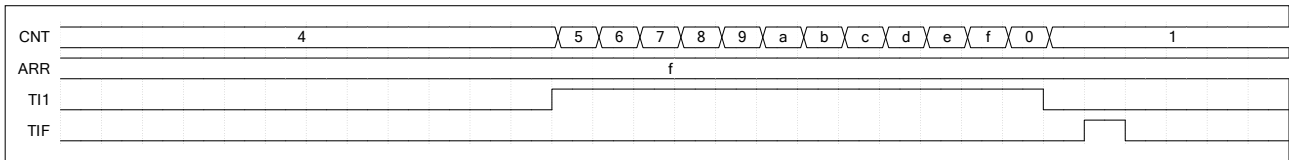


图 19-24 门控模式下的控制时序图

19.4.4.4 触发模式

配置 TIM3_SMCR 寄存器 SMS=110，从模式选择触发模式。根据 TIM3_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 TIM3_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 TIM3_CCER 寄存器的 CC1P=0，检测上升沿。
 2. 配置 TIM3_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIM3_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 TIM3_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 TIM3_ARR=0xf 的时序图。

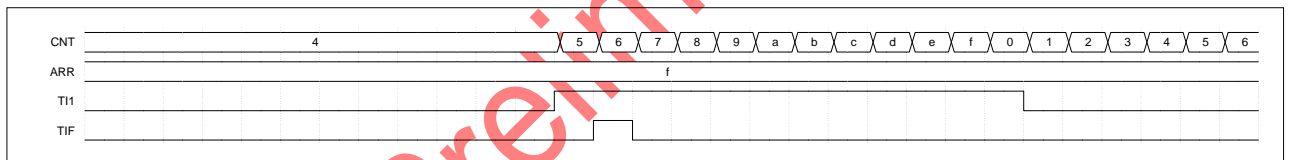


图 19-25 触发器模式下的控制时序图

19.4.4.5 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 TIM3_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIM3_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIM3_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIM3_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。
2. 配置 TIM3_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 TIM3_CCER 寄存器的 CC1P=0，选择上升沿有效。
3. 配置 TIM3_SMCR 寄存器的 SMS = 110，从模式选择为触发模式。配置 TIM3_SMCR 寄存器的 TS = 101，选择 TI1 作为输入源。
4. 配置 TIM3_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 TIM3_ARR=13 时的时序图。

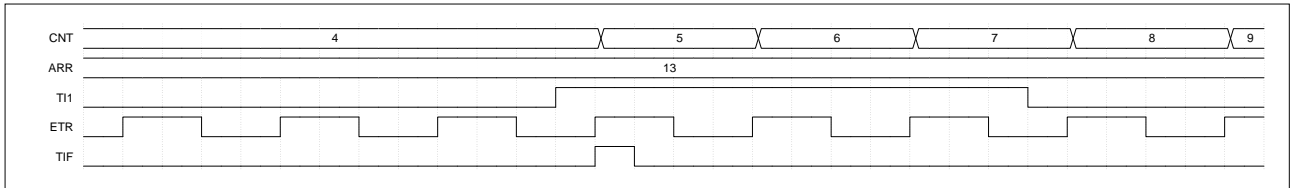


图 19-26 外部时钟模式 2+从模式（触发模式）控制时序图

19.4.5 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。

19.4.6 定时器异或

配置 TIM3_CR2 寄存器的 TI1S =1，将 TIM3_CH1、TIM3_CH2 和 TIM3_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：TIM3_CH1、TIM3_CH2 和 TIM3_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 TIM3_CCR1 寄存器中。具体配置如下：

1. 配置 TIM3_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 TIM3_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 TIM3_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 TIM3_CCER 寄存器的 CC1P=0，选择捕获发生在 TI1 信号的上升沿。
5. 配置 TIM3_CCMR1 寄存器的 IC1PSC，选择预分频系数。
6. 配置 TIM3_CCER 寄存器的 CC1E = 1，开启输入/捕获通道 1 的捕获使能。
7. 配置 TIM3_CR1 寄存器的 CEN=1，启动计数器。

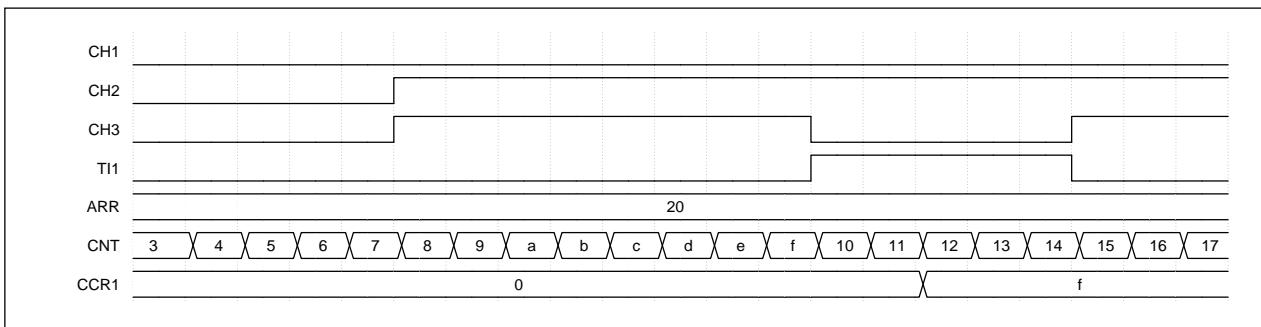


图 19-27 （TI1 异或输入）输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

19.4.7 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIM3_STOP=1，TIM3 计数器停止计数。（详见调试章节）

19.4.8 中断

TIM3 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 19-3 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

19.4.9 DMA

TIM3 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 TIM3 的一部分寄存器，也可以用于按周期读取数个寄存器。

TIM3_DCR 和 TIM3_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 TIM3_DMAR 寄存器。开启 DMA 使能后，在给定的 TIM3 事件发生时，TIM3 会给 DMA 发送请求。对 TIM3_DMAR 寄存器的每次写操作都被重定向到一个 TIM3 寄存器。

TIM3_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 TIM3_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。TIM3_DCR 寄存器的 DBA 位定义了 DMA 传输的基地址，定义从 TIM3_CR1 寄存器地址开始的偏移量（00000 为 TIM3_CR1、00001 为 TIM3_CR2 00110 为 TIM3_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 TIM3_DCR 寄存器的 DBA=01101，配置 DMA 的基地址，选择偏移地址为 TIM3_CCR1 寄存器的地址。
3. 配置 TIM3_DCR 寄存器的 DBL=00010，配置传输长度为 3。
4. 配置 TIM3_DIER 寄存器的 UDE=1，允许更新事件的 DMA 请求。

5. 配置 TIM3_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。

19.5 寄存器描述

表 19-4 TIM3 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIM3_CR1	控制寄存器 1	0x0000
0x04	TIM3_CR2	控制寄存器 2	0x0000
0x08	TIM3_SMCR	从模式控制寄存器	0x0000
0x0C	TIM3_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	TIM3_SR	状态寄存器	0x0000
0x14	TIM3_EGR	事件产生寄存器	0x0000
0x18	TIM3_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIM3_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIM3_CCER	捕获/比较使能寄存器	0x0000
0x24	TIM3_CNT	计数器	0x0000
0x28	TIM3_PSC	预分频率器	0x0000
0x2C	TIM3_ARR	自动装载寄存器	0x0000
0x34	TIM3_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIM3_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIM3_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIM3_CCR4	捕获/比较寄存器 4	0x0000
0x48	TIM3_DCR	DMA 控制寄存器（仅适用于有内置 DMA 的芯片）	0x0000
0x4C	TIM3_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000
0x50	TIM3_OR	输入选项寄存器	0x0000

19.5.1 TIM3_CR1 控制寄存器 1

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
Res.						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	<p>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, Tlx）所用的时钟之间的分频比例。</p> <p>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</p>
7	ARPE	<p>自动重载预装载使能（Auto-reload preload enable） 0: 关闭 TIM3_ARR 寄存器的影子寄存器 1: 使能 TIM3_ARR 寄存器的影子寄存器</p>
6: 5	CMS	<p>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。</p>
4	DIR	<p>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。</p>
3	OPM	<p>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</p>
2	URS	<p>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</p>

Bit	Field	Description
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, 计数器和预分频器被初始化; 如果从模式控制器接收到硬件复位, 计数器将被初始化。
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。

19.5.2 TIM3_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.								T11S	MMS			CCDS	Res.			
								rw	rw			rw				

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	T11S	T11 选择 (T11 selection) 0: TIM3_CH1 管脚连到 TI1 输入 1: TIM3_CH1、TIM3_CH2 和 TIM3_CH3 管脚经异或后作为 TI1 输入

Bit	Field	Description
6: 4	MMS	<p>主模式选择 (Master mode selection)</p> <p>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:</p> <p>000: 复位 TIM3_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。</p> <p>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。</p> <p>010: 更新 更新事件被选为 TRGO。</p> <p>011: 捕获/比较脉冲 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。</p> <p>100: 比较 OC1REF 信号被用于作为触发输出 (TRGO)</p> <p>101: 比较 OC2REF 信号被用于作为触发输出 (TRGO)</p> <p>110: 比较 OC3REF 信号被用于作为触发输出 (TRGO)</p> <p>111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)</p>
3	CCDS	<p>DMA 请求源选择 (Capture/compare DMA selection)</p> <p>0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求</p> <p>1: 发生更新事件时, 发送 CCx 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品</p>
2: 0	Reserved	保留, 必须保持复位值。

19.5.3 TIM3_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	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择 ETR 信号的极性。</p> <p>0: 高电平或上升沿有效</p> <p>1: 低电平或下降沿有效</p> <p>注: 仅适用于支持外部触发的产品</p>

Bit	Field	Description
14	ECE	<p>外部时钟使能位 (External clock enable)</p> <p>该位启用外部时钟模式 2。</p> <p>0: 禁止外部时钟模式 2</p> <p>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</p> <p>注 1: 仅适用于支持外部触发的产品。</p> <p>注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。</p> <p>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。</p> <p>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。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须低于 TIM3 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。</p> <p>00: 关闭预分频</p> <p>01: ETRP 频率除以 2</p> <p>10: ETRP 频率除以 4</p> <p>11: ETRP 频率除以 8</p> <p>注: 仅适用于支持外部触发的产品。</p>
11: 8	ETF	<p>外部触发滤波 (External trigger filter)</p> <p>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。</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> <p>注: 仅适用于支持外部触发的产品。</p>

Bit	Field	Description
7	MSM	<p>主/从模式 (Master/slave mode)</p> <p>0: 无作用</p> <p>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。</p>
6: 4	TS	<p>触发选择 (Trigger selection)</p> <p>触发输入源选择。</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>更多有关 ITRx 的细节, 参见下表。</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</p> <p>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), 并且产生一个更新事件。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), 并且产生一个更新事件, 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 19-5 TIM3 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	TIM2_TRGO	TIM3_TRGO	TIM17_OC1REF
TIM2	TIM1_TRGO	-	TIM3_TRGO	TIM14_OC1REF
TIM3	TIM1_TRGO	TIM2_TRGO	TIM16_OC1REF	TIM14_OC1REF

19.5.4 TIM3_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
	rw		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	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。

Bit	Field	Description
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

19.5.5 TIM3_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
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.		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。

Bit	Field	Description
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和外部时钟模式外的其它模式时、在 TRGI 输入端检测到有效边沿, 或门控模式下的计数停止边沿) 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 TIM3_CR1.CMS[1: 0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIM3_CNT 的值与 TIM3_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 TIM3_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIM3_CCR1

Bit	Field	Description
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 TIM3_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 TIM3_CR1 寄存器的 UDIS=0、URS=0，当 TIM3_EGR 寄存器的 UG=1 时。 - 若 TIM3_CR1 寄存器的 UDIS=0、URS=0，从模式控制器产生更新事件时。

19.5.6 TIM3_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
Res.									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，TIM3_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。

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, 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。

19.5.7 TIM3_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。

Bit	Field	Description
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 TIM3_SMCR 寄存器的 TS 位选择)</p>
7	OC1CE	<p>通道 1 比较输出清 0 使能 (Output compare 1 clear enable)</p> <p>0: OC1REF 不受 ETR 输入的影响</p> <p>1: 当检测到 ETR 输入有效电平时, OC1REF 清零</p>
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。TIM3_CCR1 与 TIM3_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIM3_CNT 的值与 TIM3_CCR1 的值相同时, 强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIM3_CNT 的值与 TIM3_CCR1 的值相同时, 强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIM3_CCR1=TIM3_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 TIM3_CNT<TIM3_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIM3_CNT > TIM3_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 TIM3_CNT<TIM3_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 TIM3_CNT > TIM3_CCR1 时强制 OC1REF 为高电平, 否则为低电平。</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>

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 TIM3_CCR1 寄存器的预装载功能, 写入 TIM3_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIM3_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM3_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 (TIM3_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 TIM3_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 TIM3_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 $CC1E=0$ (TIM3_CCER 寄存器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
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 TIM3_SMCR 寄存器的 TS 位选择)</p>

19.5.8 TIM3_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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

比较输出模式：

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	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 TIM3_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

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode)</p> <p>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。</p> <p>000: 冻结。TIM3_CCR3 与 TIM3_CNT 间的比较结果对 OC3REF 不起作用</p> <p>001: 匹配时设置为高。当 TIM3_CNT 的值与 TIM3_CCR3 的值相同时, 强制 OC3REF 为高电平</p> <p>010: 匹配时设置为低。当 TIM3_CNT 的值与 TIM3_CCR3 的值相同时, 强制 OC3REF 为低电平</p> <p>011: 匹配时翻转。当 TIM3_CCR3=TIM3_CNT 时, 翻转 OC3REF 的电平</p> <p>100: 强制为低。强制 OC3REF 为低电平</p> <p>101: 强制为高。强制 OC3REF 为高电平</p> <p>110: PWM 模式 1。在递增计数时, 当 TIM3_CNT < TIM3_CCR3 时强制 OC3REF 为高电平, 否则为低电平; 在递减计数时, 当 TIM3_CNT > TIM3_CCR3 时强制 OC3REF 为低电平, 否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时, 当 TIM3_CNT < TIM3_CCR3 时强制 OC3REF 为低电平, 否则为高电平; 在递减计数时, 当 TIM3_CNT > TIM3_CCR3 时强制 OC3REF 为高电平, 否则为低电平。</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, OC3REF 电平才改变。</p>
3	OC3PE	<p>通道 3 比较输出预装载使能 (Output compare 3 preload enable)</p> <p>0: 禁止 TIM3_CCR3 寄存器的预装载功能, 写入 TIM3_CCR3 寄存器的数值立即生效</p> <p>1: 开启 TIM3_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM3_CCR3 的预装载值在更新事件到来时生效</p> <p>注: 仅在单脉冲模式下 (TIM3_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。</p>
2	OC3FE	<p>通道 3 比较输出快速使能 (Output compare 3 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 3 比较输出快速使能</p> <p>1: 开启通道 3 比较输出快速使能</p>

Bit	Field	Description
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 3 被配置为输出</p> <p>01: 通道 3 被配置为输入, IC3 映射在 TI3 上</p> <p>10: 通道 3 被配置为输入, IC3 映射在 TI4 上</p> <p>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 TIM3_SMCR 寄存器的 TS 位选择)</p>

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	<p>输入捕获 4 滤波器 (Input capture 4 filter)</p> <p>参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler)</p> <p>参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>通道 4 捕获/比较选择 (Capture/Compare 4 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 4 被配置为输出</p> <p>01: 通道 4 被配置为输入, IC4 映射在 TI4 上</p> <p>10: 通道 4 被配置为输入, IC4 映射在 TI3 上</p> <p>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 TIM3_SMCR 寄存器的 TS 位选择)</p>

Bit	Field	Description
7: 4	IC3F	<p>通道 3 输入捕获滤波器 (Input capture 3 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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler)</p> <p>该位定义了 IC3 的预分频系数。当 $CC3E=0$ ($TIM3_CCER$ 寄存器中) 时, 预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每 2 个事件触发一次捕获</p> <p>10: 每 4 个事件触发一次捕获</p> <p>11: 每 8 个事件触发一次捕获</p>
1: 0	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 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 3 被配置为输出</p> <p>01: 通道 3 被配置为输入, IC3 映射在 TI3 上</p> <p>10: 通道 3 被配置为输入, IC3 映射在 TI4 上</p> <p>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 $TIM3_SMCR$ 寄存器的 TS 位选择)</p>

19.5.9 TIM3_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。
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。

Bit	Field	Description
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 19-6 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/高电平有效
0	1	保留

19.5.10 TIM3_CNT 计数器

偏移地址：0x24

复位值：0x0000

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

19.5.11 TIM3_PSC 预分频器

偏移地址：0x28

复位值：0x0000

Bit	Field	Description
15: 0	PSC	预分频器的值 (Prescaler value)

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。

19.5.12 TIM3_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。

19.5.13 TIM3_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 TIM3_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 TIM3_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。

19.5.14 TIM3_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。

19.5.15 TIM3_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。

19.5.16 TIM3_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。

19.5.17 TIM3_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 19.4.9 DMA 章节。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL					Res.			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 TIM3_DMAR 寄存器的第一个地址。DBA 定义为从 TIM3_CR1 寄存器所在地址开始的偏移值： 00000: TIM3_CR1 00001: TIM3_CR2 00010: TIM3_SMCR

19.5.18 TIM3_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 19.4.9 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 TIM3_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: TIM3_CR1 地址 + DBA + DMA 索引, 其中 TIM3_CR1 地址是 TIM3_CR1 寄存器所在的地址, DBA 是 TIM3_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 TIM3_DCR 寄存器中定义的 DBL 值。

19.5.19 TIM3_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
Res.								TI4_RMP		Res.				ETR_RMP	
								rw	rw					rw	rw

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7: 6	TI4_RMP	TI4 复用 00: CH4 GPIO 输入 其他保留
5: 2	Reserved	保留, 必须保持复位值。
1: 0	ETR_RMP	ETR 复用 00: ETR GPIO 输入 01: LSI 时钟输入 10: 保留 11: HSE_CLK_DIV_128 时钟输入

20 TIM14 基本定时器

20.1 简介

TIM14 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）。

20.2 功能框图

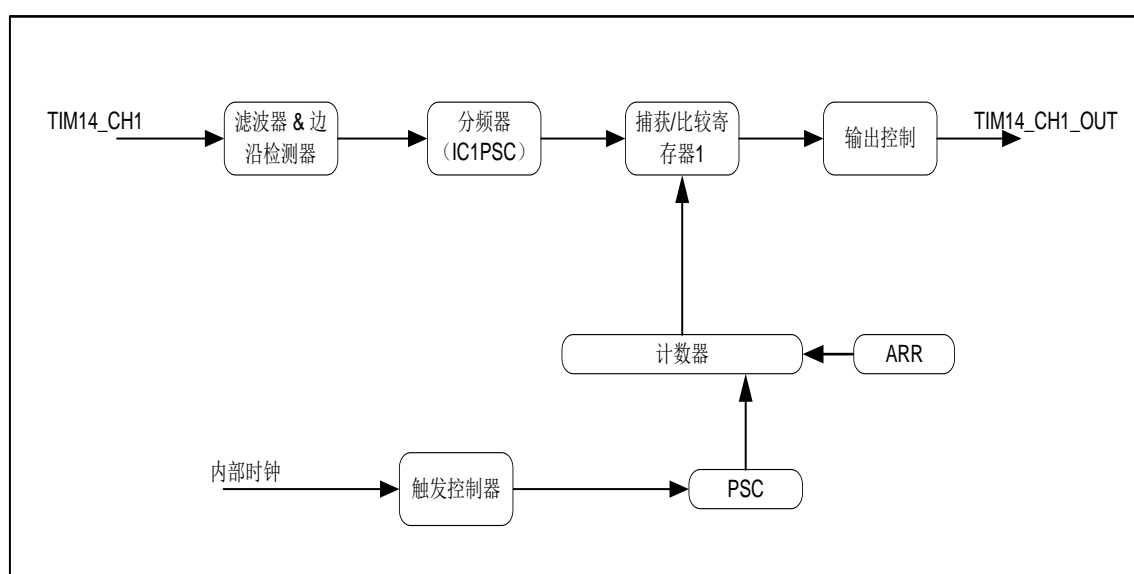


图 20-1 TIM14 结构图

上图为 TIM14 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。

20.3 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

20.4 功能描述

20.4.1 时钟

20.4.1.1 时钟选择

计数器的时钟由内部时钟（INT_CK）提供。

20.4.1.2 时基单元

TIM14 的时基单元主要包括：计数器寄存器（TIM14_CNT）、预分频器寄存器（TIM14_PSC）和自动预装载寄存器（TIM14_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 TIM14_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。

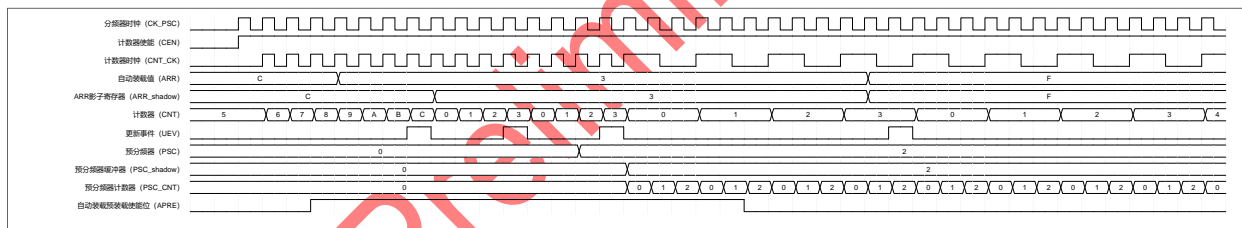


图 20-2 自动预装载

20.4.1.3 计数模式

TIM14 仅支持递增计数模式。在使能 TIM14_CR1 寄存器的 CEN 后计数器由 0 开始递增计数，直至 TIM14_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 TIM14_EGR 寄存器的 UG=1，同样可以产生一个更新事件。

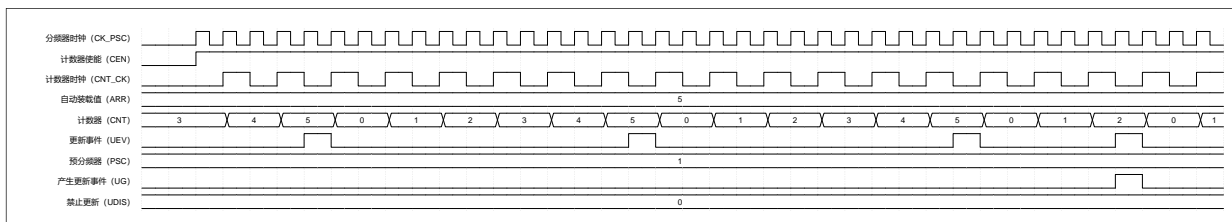


图 20-3 递增计数模式（UDIS=0）

通过配置 TIM14_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

IC1F[3: 0]	采样频率和滤波宽度	IC1F[3: 0]	采样频率和滤波宽度
0101	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/2$, $N=8$	1101	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/32$, $N=5$
0110	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/4$, $N=6$	1110	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/32$, $N=6$
0111	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/4$, $N=8$	1111	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/32$, $N=8$

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(TIM14_SR)中的捕获标志位 CC1IF 置 1,通过配置 CC1IF=0 或读取 TIM14_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 TIM14_CCR1 寄存器中,步骤如下:

1. 配置 TIM14_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 TIM14_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 TIM14_CCER 寄存器的 CC1P=0, 选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIM14_CCMR1 寄存器的 IC1PSC[1: 0], 选择预分频系数。
5. 配置 TIM14_CCER 寄存器的 CC1E = 1, 开启输入/捕获通道 1 的捕获使能。
6. 配置 TIM14_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 TIM14_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, TIM14_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 TIM14_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。

20.4.3 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:

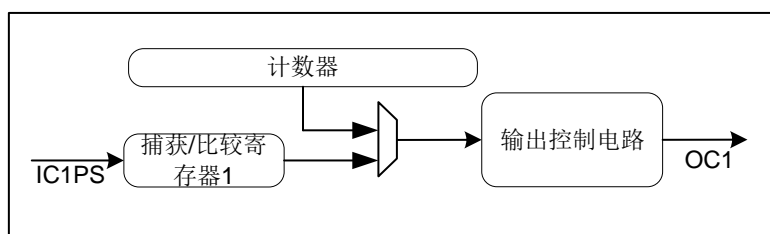


图 20-6 比较输出部分结构图

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。

20.4.3.1 强制输出

配置 TIM14_CCMR1 寄存器的 CC1S = 00，将通道 CC1 设置为输出模式，通过配置 TIM14_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 TIM14_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 TIM14_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 TIM14_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。

20.4.3.2 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 TIM14_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（TIM14_SR 寄存器中的 CC1IF 位）。
3. 当配置了 TIM14_DIER 寄存器中的 CC1IE = 1，匹配时则产生一个中断。
4. 当配置了 TIM14_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 TIM14_ARR 和 TIM14_CCR1 寄存器。
3. 配置 TIM14_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 TIM14_CCMR1 寄存器的 OC1M = 011，OC1 比较匹配时翻转。
 - ◆ 配置 TIM14_CCMR1 寄存器的 OC1PE = 0，禁止 TIM14_CCR1 寄存器的预装载功能。
 - ◆ 配置 TIM14_CCER 寄存器的 CC1P = 1，OC1 低电平有效。
 - ◆ 配置 TIM14_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 TIM14_CR1 寄存器的 CEN =1, 启动计数器。

当配置 TIM14_CCMR1 寄存器中 OC1PE=0, 禁止 TIM14_CCR1 寄存器的预装载功能时, 可以随时写入 TIM14_CCR1 寄存器, 并且写入的值立即生效。当配置 TIM14_CCMR1 寄存器中 OC1PE=1, 启用 TIM14_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, TIM14_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。

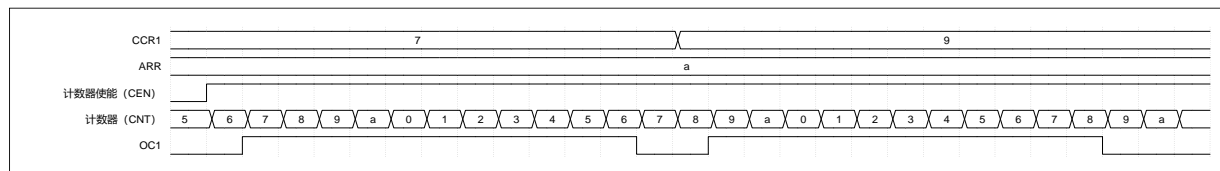


图 20-7 比较输出模式, OC1 信号在匹配时翻转

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 TIM14_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。

20.4.3.3 PWM 输出

在 PWM 模式下, 根据 TIM14_ARR 寄存器和 TIM14_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 TIM14_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, TIM14 可以产生 1 个独立占空比的 PWM 输出信号。PWM 模式下可开启 TIM14_CCR1 的预装载功能和 TIM14_ARR 寄存器的预装载功能。写入 TIM14_CCR1 预装载寄存器和 TIM14_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 TIM14_EGR 的 UG=1, 产生更新事件用于初始化所有的寄存器。

配置 TIM14_CCER 寄存器的 CC1P 位选择 OC1 的有效极性。配置 TIM14_CCER 寄存器的 CC1E 位控制 OC1 的输出使能。

20.4.3.3.1 PWM 边沿对齐模式——递增计数模式

配置 TIM14_CCMR1 寄存器的 CC1S=00, 选择输出模式, OC1M=110, 选择 PWM 模式 1, 当 TIM14_CNT < TIM14_CCR1 时通道 1 (OC1REF) 为有效电平, 否则为无效电平。如果 TIM14_CCR1 中的比较值大于自动重装载值 (TIM14_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。

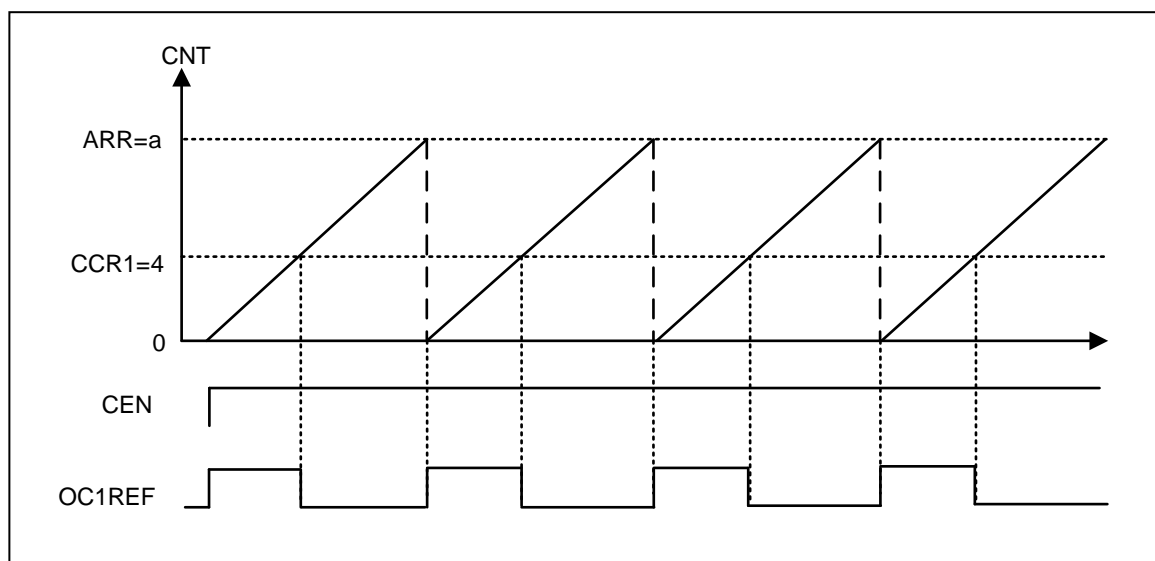


图 20-8 边沿对齐递增计数时 PWM 模式 1 的波形

20.4.3.4 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 TIM14_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。

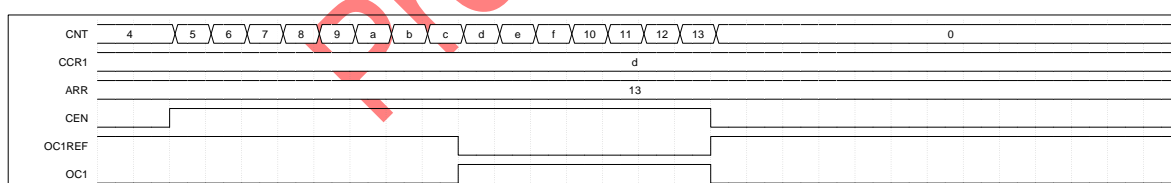


图 20-9 单脉冲模式

20.4.4 调试模式

在调试模式下，配置 DBG_CR 寄存器中 DBG_TIM14_STOP=1，TIM14 计数器停止计数。（详见调试章节）

20.4.5 中断

TIM14 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 20-2 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

20.4.6 DMA

TIM14 能够在发生捕获/比较事件或更新事件时生成 DMA 请求。

20.5 寄存器描述

表 20-3 TIM14 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIM14_CR1	控制寄存器 1	0x0000
0x0C	TIM14_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	TIM14_SR	状态寄存器	0x0000
0x14	TIM14_EGR	事件产生寄存器	0x0000
0x18	TIM14_CCMR1	捕获/比较模式寄存器 1	0x0000
0x20	TIM14_CCER	捕获/比较使能寄存器	0x0000
0x24	TIM14_CNT	计数器	0x0000
0x28	TIM14_PSC	预分频率器	0x0000
0x2C	TIM14_ARR	自动装载寄存器	0x0000
0x34	TIM14_CCR1	捕获/比较寄存器 1	0x0000
0x44	TIM14_BDTR	刹车和死区寄存器	0x0000

20.5.1 TIM14_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
Res.						CKD		ARPE	Res.			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	自动重载预装载使能 (Auto-reload preload enable) 0: 关闭 TIM14_ARR 寄存器的影子寄存器 1: 使能 TIM14_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

20.5.2 TIM14_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.						CC1DE	UDE	Res.						CC1IE	UIE
						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: 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

20.5.3 TIM14_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
Res.						CC1OF	Res.						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。

Bit	Field	Description
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: TIM14_CNT 的值与 TIM14_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 TIM14_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIM14_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。

20.5.4 TIM14_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
Res.													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

20.5.5 TIM14_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	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。
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。TIM14_CCR1 与 TIM14_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIM14_CNT 的值与 TIM14_CCR1 的值相同时，强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIM14_CNT 的值与 TIM14_CCR1 的值相同时，强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转，当 TIM14_CCR1=TIM14_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 TIM14_CNT<TIM14_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 TIM14_CNT<TIM14_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>

Bit	Field	Description
3	OC1PE	通道 1 比较输出预装载使能 (Output compare 1 preload enable) 0: 禁止 TIM14_CCR1 寄存器的预装载功能, 写入 TIM14_CCR1 寄存器的数值立即生效。 1: 开启 TIM14_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM14_CCR1 的预装载值在更新事件到来时生效。 注: 若该位置 1, 在单脉冲模式下 (TIM14_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。
2	Reserved	保留, 必须保持复位值。
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 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 1 被配置为输出 01: 通道 1 被配置为输入 10: 保留 11: 保留

输入捕获模式:

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7: 4	IC1F	通道 1 输入捕获滤波器 (Input capture 1 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	IC1PSC	通道 1 输入/捕获预分频器 (Input capture 1 prescaler) 该位定义了 IC1 的预分频系数。当 CC1E=0 (TIM14_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	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 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 1 被配置为输出 01: 通道 1 被配置为输入 10: 保留 11: 保留

20.5.6 TIM14_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
Res.												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

Bit	Field	Description
15: 4	Reserved	保留, 必须保持复位值。
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 IC1 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 IC1 极性/电平选择表。

Bit	Field	Description
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 通道 1 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 20-4 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/高电平有效
0	1	保留

20.5.7 TIM14_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)													

20.5.8 TIM14_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。

20.5.9 TIM14_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。

20.5.10 TIM14_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 TIM14_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 TIM14_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。

20.5.11 TIM14_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	Res.														
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 (TIM14_CCER 寄存器的 CC1E), 则开启 OC1 输出
14: 0	Reserved	保留, 必须保持复位值。

Preliminary

21 TIM16/17 基本定时器

TIM16、TIM17 具有相同的功能，本章节统一表述为 TIMx，部分示意图以 TIM16 为例展示。

21.1 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）。

21.2 功能框图

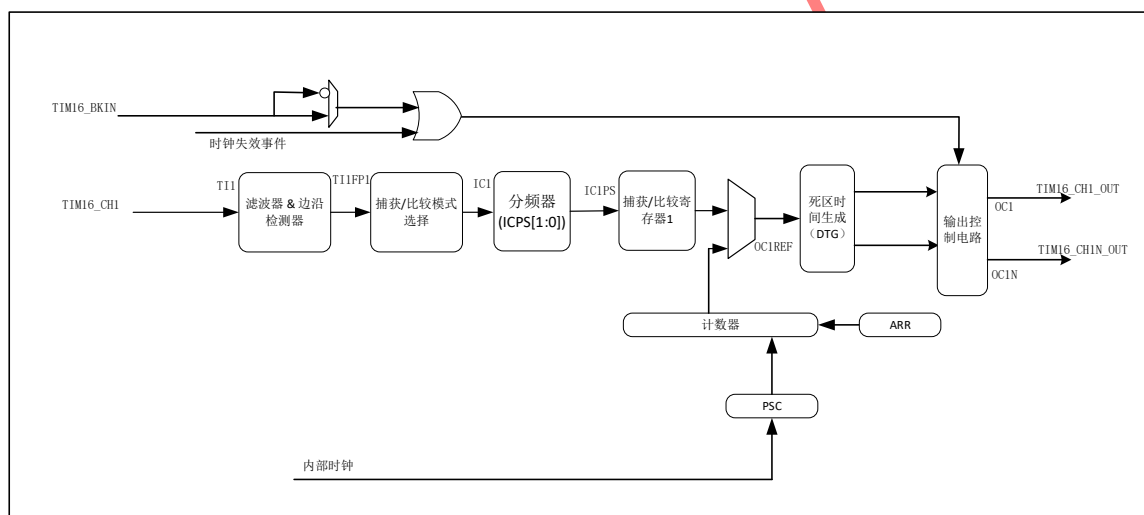


图 21-1 TIMx 结构图

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。

21.3 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

21.4 功能描述

21.4.1 时钟

21.4.1.1 时钟选择

计数器的时钟由内部时钟（INT_CK）提供。

21.4.1.2 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。

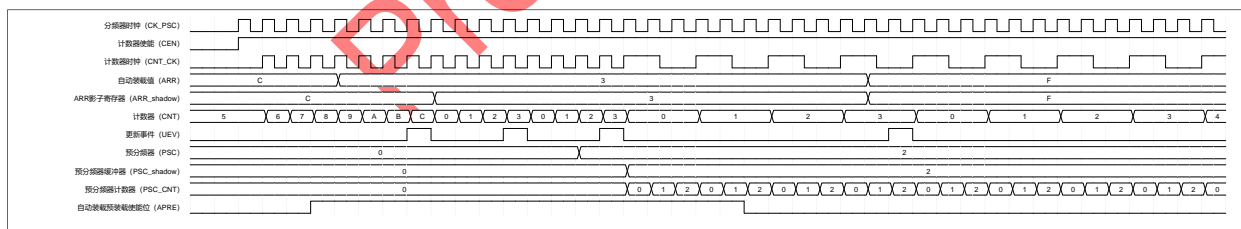


图 21-2 自动预装载

21.4.1.3 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。

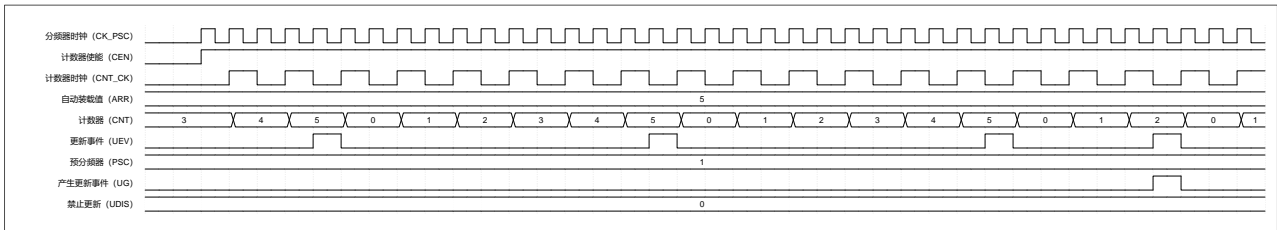


图 21-3 递增计数模式 (UDIS=0)

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。

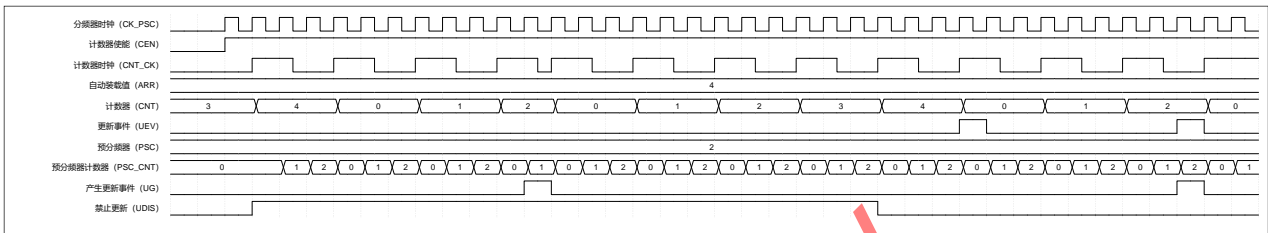


图 21-4 递增计数模式 (UDIS=1 禁止产生更新事件)

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。

21.4.2 输入捕获

21.4.2.1 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：

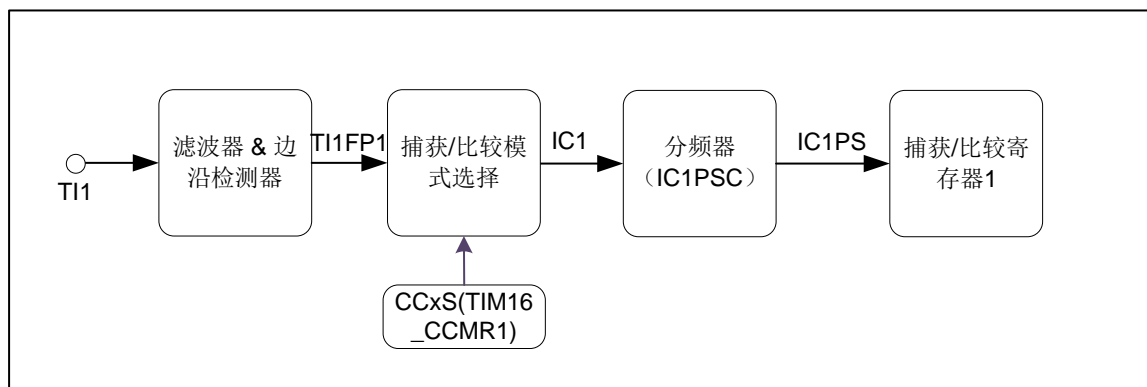


图 21-5 TIMx 输入捕获结构图

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 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 21-1 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。

21.4.3 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：

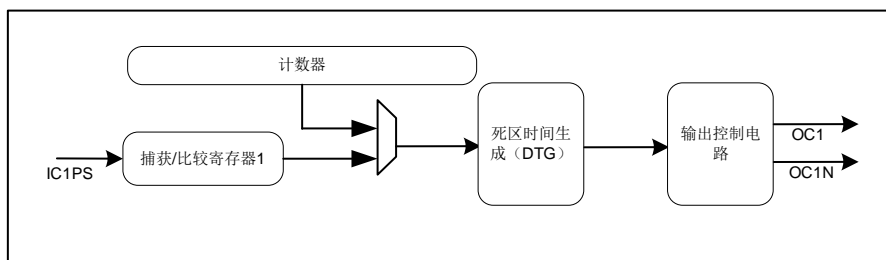


图 21-6 比较输出部分结构图

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。

21.4.3.1 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。

21.4.3.2 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。

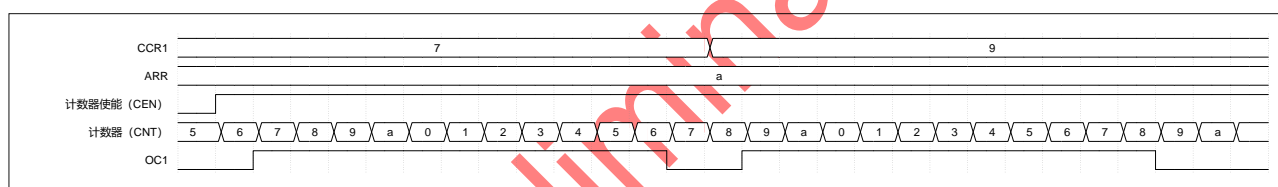


图 21-7 比较输出模式，OC1 信号在匹配时翻转

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。

21.4.3.3 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。

21.4.3.3.1 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。

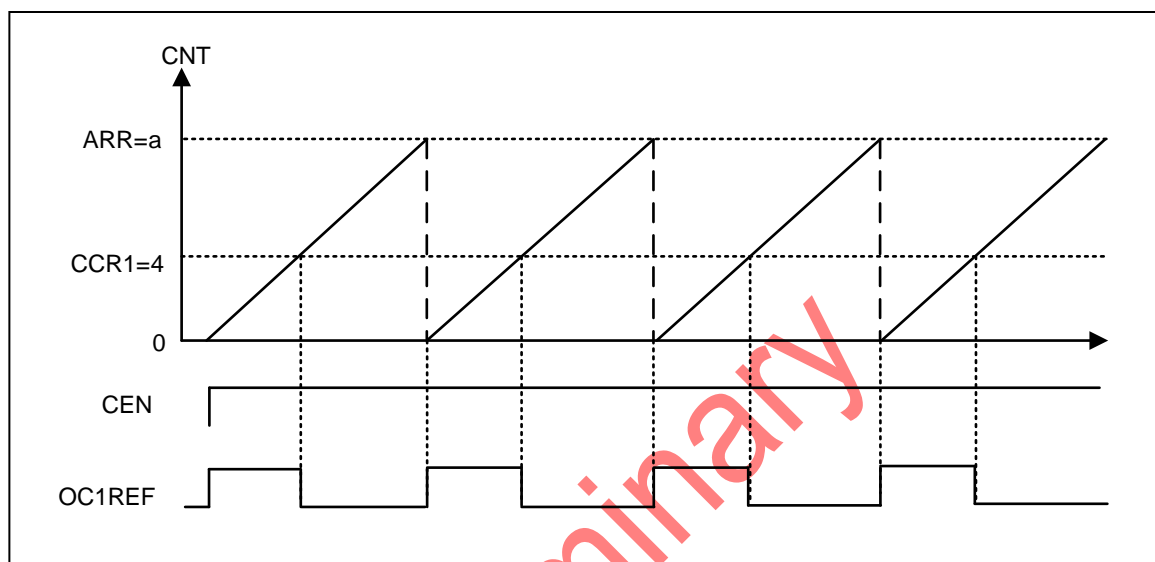


图 21-8 边沿对齐递增计数时 PWM 模式 1 的波形

21.4.3.4 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 21-2 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 $T_{DTS}=125\text{ns}$,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 16 μs 至 31750ns。
- 若步长时间为 1 μs ,死区时间为 32 μs 至 63 μs 。

- 若步长时间为 $2\mu\text{s}$ ，死区时间为 $64\mu\text{s}$ 至 $126\mu\text{s}$ 。

当不存在刹车电路时，同时配置 $\text{CC1E}=1$ 和 $\text{CC1NE}=1$ ，开启死区插入，否则还需要配置 $\text{MOE}=1$ 。

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。

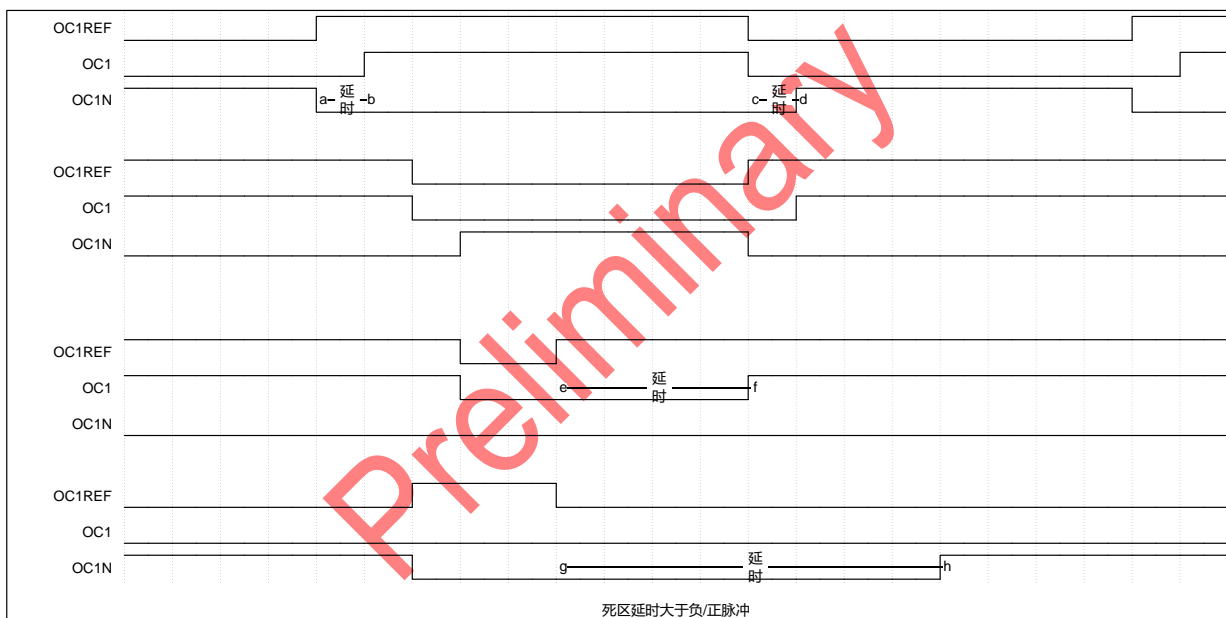


图 21-9 死区插入

21.4.3.5 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 21-3 当 $\text{MOE}=1$ ， $\text{OSSI}=0/1$ ， $\text{OSSR}=0$ 时：

CC1E	CC1NE	OC1	OC1N
0	0	$\text{OC1}=0$, $\text{OC1_EN}=0$	$\text{OC1N}=0$, $\text{OC1N_EN}=0$

CC1E	CC1NE	OC1	OC1N
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 21-4 当 MOE=1, OSS1=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 21-5 当 MOE=0, OSS1=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 21-6 当 MOE=0, OSS1=1, OSSR=0/1 时:

CC1E	CC1NE	OC1	OC1N
0	0	OC1_EN=1, OC1N_EN=1 异步的: OC1=CC1P, OC1N=CC1NP	

CC1E	CC1NE	OC1	OC1N
		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	
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：

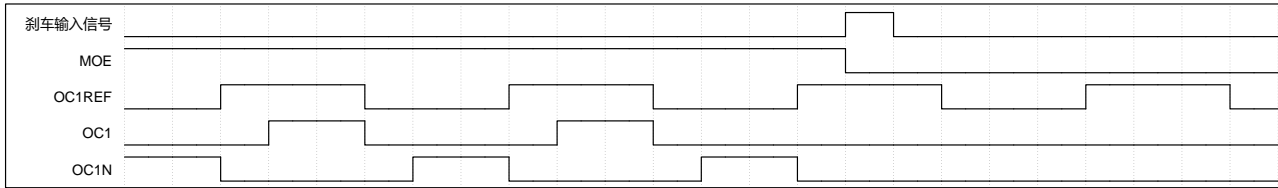


图 21-10 响应刹车的输出（OISx=0, OISxN=0）

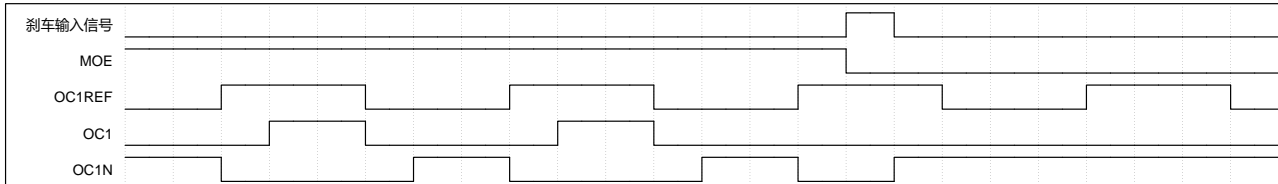


图 21-11 响应刹车的输出（OISx=0, OISxN=1）

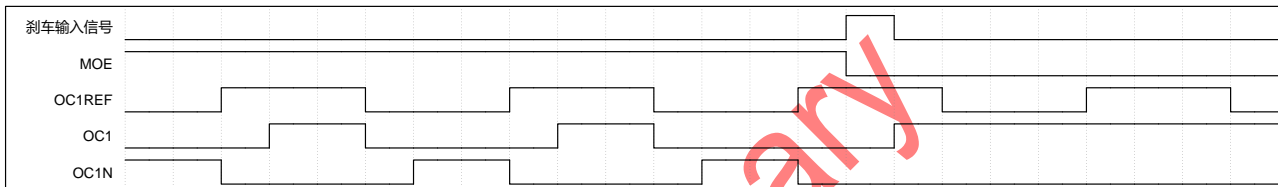


图 21-12 响应刹车的输出（OISx=1, OISxN=0）

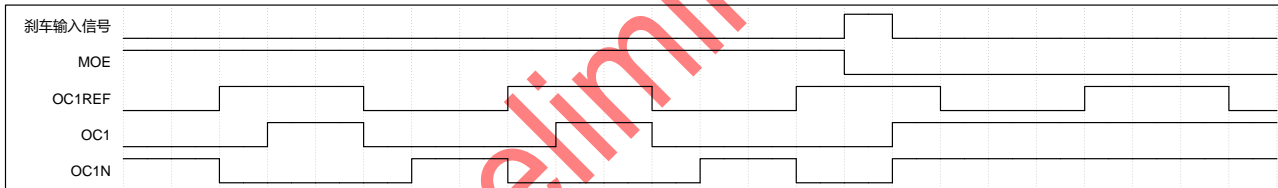


图 21-13 响应刹车的输出（OISx=1, OISxN=1）

21.4.3.6 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。

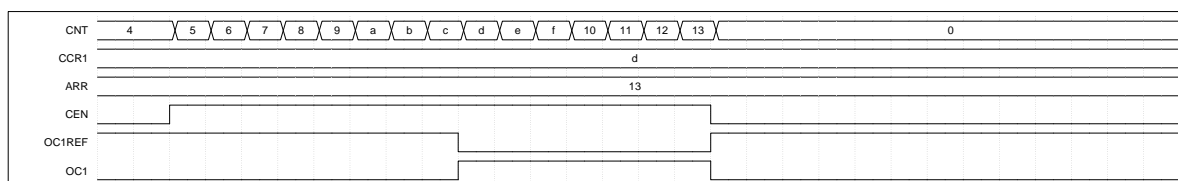


图 21-14 单脉冲模式

21.4.3.6.1 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。

21.4.4 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。

21.4.5 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）

21.4.6 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 21-7 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

21.4.7 DMA

TIMx 能够在发生捕获/比较事件或更新事件时生成 DMA 请求。

21.5 寄存器描述

表 21-8 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

Offset	Acronym	Register Name	Reset
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
0x74	TIMx_BKINF	刹车滤波寄存器	0x0000 0000

21.5.1 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
Res.						CKD		ARPE	Res.			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: $t_{DTS} = 2x t_{INT_CK}$ 10: $t_{DTS} = 4x t_{INT_CK}$ 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

Bit	Field	Description
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

21.5.2 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
Res.						OIS1N	OIS1	Res.				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

Bit	Field	Description
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。

21.5.3 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
Res.						CC1DE	UDE	BIE	Res.	COMIE	Res.			CC1IE	UIE
						rw	rw	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	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。

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

21.5.4 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
Res.						CC1OF	Res.	BIF	Res.	COMIF	Res.			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。
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。

Bit	Field	Description
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。

21.5.5 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
Res.								BG	Res.	COMG	Res.			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。

21.5.6 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	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。
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>

Bit	Field	Description
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 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 01: 通道 1 被配置为输入 10: 保留 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。
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 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 $f_{sampling}=f_{DTS}/32$, $N=8$</p>

Bit	Field	Description
3: 2	IC1PSC	通道 1 输入/捕获预分频器 (Input capture 1 prescaler) 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 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	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 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 1 被配置为输出 01: 通道 1 被配置为输入 10: 保留 11: 保留

21.5.7 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。
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。

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。 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 21-9 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/高电平有效
0	1	保留

21.5.8 TIMx_CNT 计数器

偏移地址: 0x24

复位值: 0x0000

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

21.5.9 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。

21.5.10 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。

21.5.11 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
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。

21.5.12 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						
r/w	r/w	r/w	r/w	r/w	r/w	r/w			r/w						

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。

Bit	Field	Description
12	BKE	<p>刹车功能使能 (Break enable)</p> <p>0: 禁止刹车输入</p> <p>1: 开启刹车输入</p> <p>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。</p> <p>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。</p>
11	OSSR	<p>运行模式下“关闭状态”选择 (Off-state selection for Run mode)</p> <p>该位仅适用于当 MOE = 1 且通道为互补输出。</p> <p>0: 当定时器不工作时, 禁止 OC1/OC1N 输出</p> <p>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。</p> <p>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。</p>
10	OSSI	<p>空闲模式下“关闭状态”选择 (Off-state selection for Idle mode)</p> <p>该位仅适用于当 MOE = 0 且通道设为输出时。</p> <p>0: 当定时器不工作时, 禁止 OC1/OC1N 输出。</p> <p>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。</p> <p>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。</p>
9: 8	LOCK	<p>锁定设置 (Lock configuration)</p> <p>该位定义了寄存器的写保护功能。</p> <p>00: 写保护功能关闭, 寄存器无写保护</p> <p>01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OIS1/OIS1N 位</p> <p>10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位 (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</p> <p>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)</p> <p>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。</p>
7: 0	DTG	<p>死区发生器设置 (Dead-time generator setup)</p> <p>这些位定义了插入互补输出之间的死区持续时间。</p> <p>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。</p>

21.5.13 TIMx_BKINF 刹车滤波寄存器

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
Res															BKIN_S EL
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKIN_SEL											BKINF				BKINFE
rw											rw				rw

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值。
16: 5	BKIN_SEL	<p>BKINSEL: 刹车输入选择 (break input sel)</p> <p>TIM16: xxxxxxxxx1x: 若第 1bit 为 1, 则选择 PB[5]输入信号 xxxxxxxxx1: 若第 0bit 为 1, 则选择 CSS 信号</p> <p>TIM17: xxxxxxxxx1xx: 若第 2bit 为 1, 则选择 PB[4]输入信号 xxxxxxxxx1x: 若第 1bit 为 1, 则选择 PA[10]输入信号 xxxxxxxxx1: 若第 0bit 为 1, 则选择 CSS 信号</p>
4: 1	BKINF	<p>BKINF: BKIN 数字滤波采样频率 (break input filter)</p> <p>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</p> <p>注: 更改档位时应先关闭刹车滤波使能位。</p>

Bit	Field	Description
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。

Preliminary

22 LPTIM 低功耗定时器

22.1 简介

LPTIM 由一个 16 位计数器组成，可以为用户提供便捷的计数和定时功能。LPTIM 运行在 CORE 电源域，可以工作在低功耗模式下，具有低功耗的特点。LPTIM 的时钟也可由外部时钟提供。可以在休眠模式下实现外部脉冲计数功能。通过外部输入的触发信号，能够实现低功耗超时唤醒。LPTIM 具有外部时钟计数，超时唤醒功能和 PWM 输出等多种用途。

22.2 功能框图

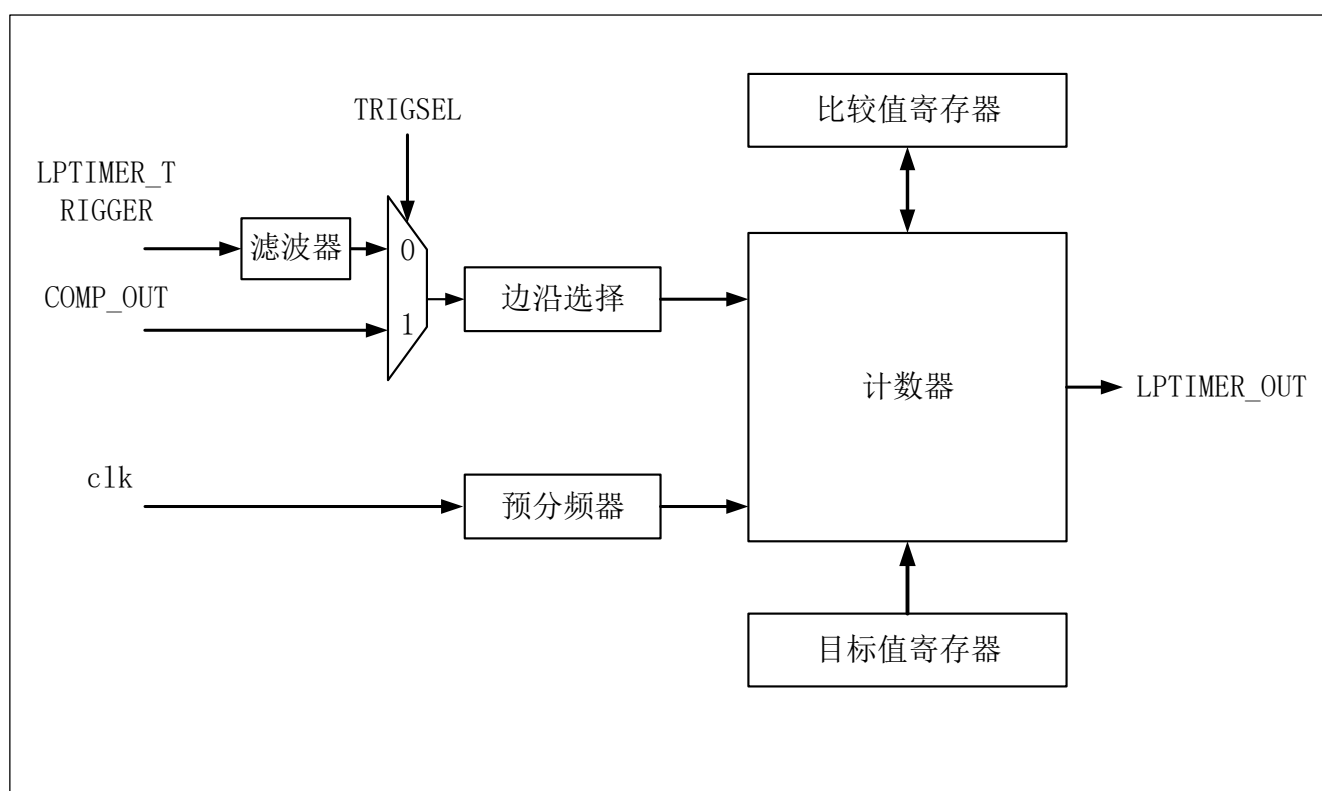


图 22-1 功能框图

上图为 LPTIM 的结构框图，主要由触发源选择、输出单元和计数模块等结构组成。

22.3 主要特征

- 16 位递增计数器
- 3-bit 异步时钟预分频器，对应的分频系数分别为 1、2、4、8、16、32、64、128
- 16-bit 比较寄存器和目标值寄存器

- 触发源可选：硬件触发、软件触发
- 输入极性可选
- 外部脉冲计数（无时钟时）
- 低功耗超时唤醒
- PWM 输出

22.4 中断

LPTIM 的中断包括：外部触发中断，比较匹配中断，计数器溢出中断，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。

22.5 功能描述

22.5.1 计数模式

配置 LPT_CFG 寄存器 MODE 位，可以选择 LPTIM 的计数模式，连续计数模式（溢出后重新从 0 开始计数）或单次计数模式（溢出后停止计数）

22.5.2 普通定时器

- 配置 LPT_CFG 寄存器 TMODE=00，选择普通计数器模式
- 计数器时钟可选择外部时钟或内部时钟
- 配置 LPT_CTRL 寄存器 LPTEN=1，使能计数器
- 计数器使能后有两个周期的同步过程
- 同步完成后，计数器开始工作

22.5.3 Trigger 脉冲触发计数

- 配置 LPT_CFG 寄存器 TMODE=01，选择 Trigger 脉冲触发计数模式
- 计数器时钟选择内部时钟
- 配置 LPT_CFG 寄存器 TRIGCFG，选择外部触发信号的有效沿
- 配置 LPT_IE 寄存器，打开比较匹配中断使能和计数器溢出中断使能，当计数器值和比较值匹配或计数器溢出时产生相应的中断
- 配置 LPT_CTRL 寄存器 LPTEN=1，使能计数器
- 计数器使能后有两个周期的同步过程
- 同步完成后，计数器开始工作

22.5.4 TimeOut 模式

- 配置 LPT_CFG 寄存器 TMODE=11，选择 TimeOut 模式
- 计数器时钟可以选择内部时钟或外部时钟
- 配置 LPT_CFG 寄存器 TRIGSEL=0，触发源选择外部引脚触发
- 外部触发源第一次触发启动计数器，再次检测到触发源时，复位计数器并重新计数
- 配置 LPT_IE 寄存器 OVIE=1，打开计数器溢出中断使能，在计数器溢出时产生溢出中断，计数器停止计数，清除使能位

注：

- 使能后有两个计数时钟周期的同步，同步完成后启动计数器
- 在此模式下，若下次触发信号在达到比较值之前没有到来，则产生超时中断，可在低功耗模式下唤醒 MCU

22.5.5 PWM 输出

- 配置 LPT_CFG 寄存器 PWM=1，选择 PWM 输出模式
- 配置 LPT_CFG 寄存器 TMODE=00，MODE=0，选择普通计数器模式，连续计数模式
- 计数器时钟可选择外部时钟或内部时钟
- 配置 LPT_CMP 和 LPT_TARGET 寄存器，设定比较值和目标值，PWM 的占空比由比较值和目标值决定，输出在计数器值等于比较值翻转为 1，在等于目标值时翻转为 0；
- 配置 LPT_CTRL 寄存器 LPTEN=1，使能计数器

22.6 寄存器描述

22.6.1 寄存器概览

表 22-1 LPTIM 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	LPT_CFG	配置寄存器	0x0000
0x04	LPT_IE	中断使能寄存器	0x0000
0x08	LPT_IF	中断标志寄存器	0x0000
0x0C	LPT_CTRL	控制寄存器	0x0000
0x10	LPT_CNT	计数寄存器	0x0000
0x14	LPT_CMP	比较值寄存器	0x0000
0x18	LPT_TARGET	目标值寄存器	0x0000

22.6.2 LPT_CFG 配置寄存器

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
FLTEN	Res.				DIVSEL			TRIGCFG		TRIGSEL	POLARITY	PWM	TMODE		MODE
rw					rw			rw		rw	rw	rw	rw		rw

Bit	Field	Description
15	FLTEN	输入信号滤波使能位 0: 输入信号滤波禁止 1: 输入信号滤波使能
14: 11	Reserved	保留，始终读为 0
10: 8	DIVSEL	计数器时钟分频选择 000: 1 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频 注：时钟分频设置在计数器停止状态
7: 6	TRIGCFG	外部触发边沿选择 00: 外部输入信号上升沿触发 01: 外部输入信号下降沿触发 10: 外部输入信号上升沿/下降沿触发 11: 外部输入信号上升沿/下降沿触发
5	TRIGSEL	触发输入源选择 0: 外部引脚触发 1: COMP 触发
4	POLARITY	比较匹配波形极性选择 0: 正极性波形，即第一次计数值=比较值时产生输出波形上升沿 1: 负极性波形，即第一次计数值=比较值时产生输出波形下降沿
3	PWM	脉冲宽度调制 0: 周期方波输出模式 1: PWM 输出模式
2: 1	TMODE	工作模式选择 00/10: 带波形输出的普通定时器模式 01: Trigger 脉冲触发计数模式 11: Timeout 模式

Bit	Field	Description
0	MODE	计数模式 0: 连续计数模式: 计数器被触发后保持运行, 直到被关闭为止。计数器达到目标之后回到 0 重新开始计数, 并产生溢出中断 1: 单次计数模式: 计数器被触发后计数达到目标值后回到 0, 并自动停止, 产生溢出中断。

22.6.3 LPT_IE 中断使能寄存器

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.													COMPI E	TRIGIE	OVIE
													rw		rw

Bit	Field	Description
15: 3	Reserved	保留, 始终读为 0
2	COMPIE	比较匹配中断使能位 0: 比较匹配中断禁止 1: 比较匹配中断使能
1	TRIGIE	外部触发中断使能位 0: 外部触发中断禁止 1: 外部触发中断使能
0	OVIE	计数器溢出中断使能位 0: 计数器溢出中断禁止 1: 计数器溢出中断使能

22.6.4 LPT_IF 中断标志寄存器

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
Res.													COMPIF	TRIGIF	OVIF
													w1c	w1c	w1c

Bit	Field	Description
15: 4	Reserved	保留, 始终读为 0

Bit	Field	Description
2	COMPIF	比较匹配中断标志位，写 1 清零 0: 无中断产生 1: 比较匹配中断产生
1	TRIGIF	外部触发中断标志位，写 1 清零 0: 无中断产生 1: 外部触发中断产生
0	OVIF	计数器溢出中断标志位，写 1 清零 0: 无中断产生 1: 计数器溢出中断产生

22.6.5 LPT_CTRL 控制寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

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

Bit	Field	Description
15: 1	Reserved	保留，始终读为 0
0	LPTEN	LPTIM 使能位 0: 禁止计数器计数 1: 使能计数器计数

22.6.6 LPT_CNT 计数寄存器

偏移地址: 0x10

复位值: 0x0000 0000

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

Bit	Field	Description
15: 0	CNT	LPTIM 计数器值

22.6.7 LPT_CMP 比较值寄存器

偏移地址: 0x14

复位值: 0x0000 0000

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

COMPARE_REG[15: 0]

rw

Bit	Field	Description
15: 0	COMPARE_REG	LPTIM 比较值寄存器

22.6.8 LPT_TARGET 目标值寄存器

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

TARGET_REG[15: 0]

rw

Bit	Field	Description
15: 0	TARGET_REG	LPTIM 目标值寄存器, PWM 输出频率 = 时钟频率 / (TARGET_REG+1)

Preliminary

23 WWDG 窗口看门狗

23.1 简介

窗口看门狗是喂狗时间有上下限范围的看门狗，主要用于检测由外部干扰和不可预测的条件导致程序跑飞而引起的软件问题。

根据程序正常执行的时间来设置刷新看门狗的一个时间窗口，保证不会提前刷新看门狗也不会滞后刷新看门狗，这样就可以检测出程序有没有按照正常的路径运行或者出现非正常地跳过某些程序段的情况。

23.2 功能框图

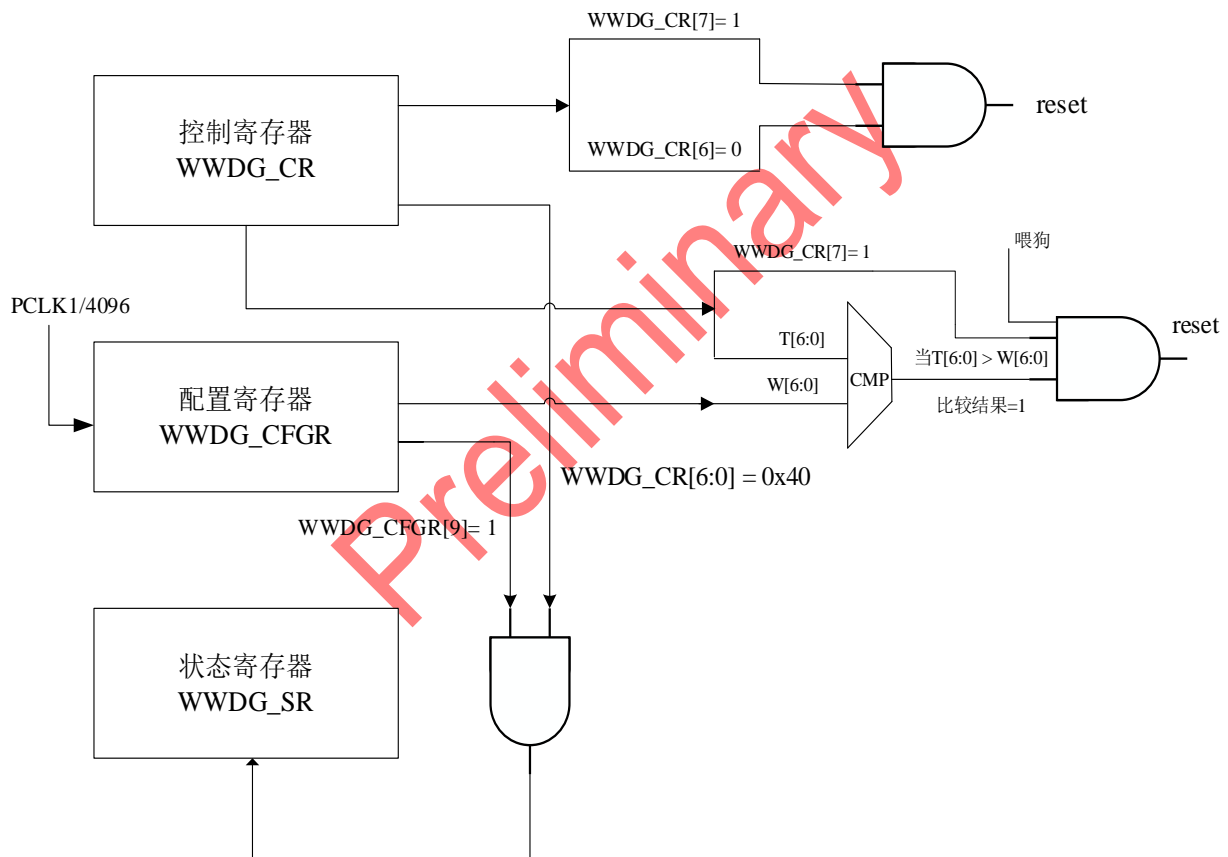


图 23-1 WWDG 功能框图

23.3 主要特征

- 可编程的自由运行的 7 位递减计数器（0xFF 和 0xC0）
- 窗口外喂狗产生复位

上限：当计数器的值大于配置寄存器（WWDG_CFGR）设定的值时喂狗，将会产生复位

下限：当计数器的值小于 0x40 时，产生复位

- 中断：

如果启动了窗口看门狗并且允许中断,当递减计数器递减到 0x40 时,将产生提前唤醒中断(EWI),可以在中断处理函数中,向 WWDG_CR 重新装载计数器的值来达到喂狗的目的,从而防止复位。

23.4 功能描述

- 当控制寄存器 WWDG_CR 中的第 7bit 位 WDGA 被置“1”时,启动窗口看门狗,若计数器的值从下限固定值 0x40 递减到 0x3F (WWDG_CR 中的第 6bit 位清零),将会产生一个复位。或者计数器的值大于配置寄存器 WWDG_CFGR 设定的值时喂狗,也会产生复位。
- 窗口看门狗和独立看门狗一样,都是在应用程序正常运行中做定时喂狗操作,来防止 MCU 发生复位。区别在于:窗口看门狗喂狗必须在计数器值小于窗口寄存器值的时候操作,也只有在这种情况下,喂狗才不会产生复位。其中,写入控制寄存器 WWDG_CR 的值必须在 0xFF 和 0xC0 之间。
- 窗口看门狗在系统复位后不会自动运行,需要设置控制寄存器 WWDG_CR 中的 WDGA 位开启窗口看门狗。因为 WDGA 位只能由硬件复位清零,所以一旦软件置位后,窗口看门狗会一直开启。当窗口看门狗被打开后,需要配置控制寄存器 WWDG_CR 的第 6bit 位置“1”,以防止立即产生窗口看门狗复位。WWDG_CR[6: 0]代表看门狗复位前的计数值,由于预分频值软件可配,且复位前的延时与预分频值的配置有关,因此复位前的延时会在一个最小值和最大值之间变化。
- 利用提前唤醒中断(EWI)可以重载计数器来避免复位。设置配置寄存器 WWDG_CFGR 中的 EW1 位来使能中断,当计数器数值到达 0x40 且产生中断后,在中断处理函数中向控制寄存器 WWDG_CR 重写计数器的值来达到喂狗目的从而防止复位。在状态寄存器 WWDG_SR 中软件写“0”来清除中断。

23.5 窗口看门狗超时时间

T[6:0]CNT递减计数器

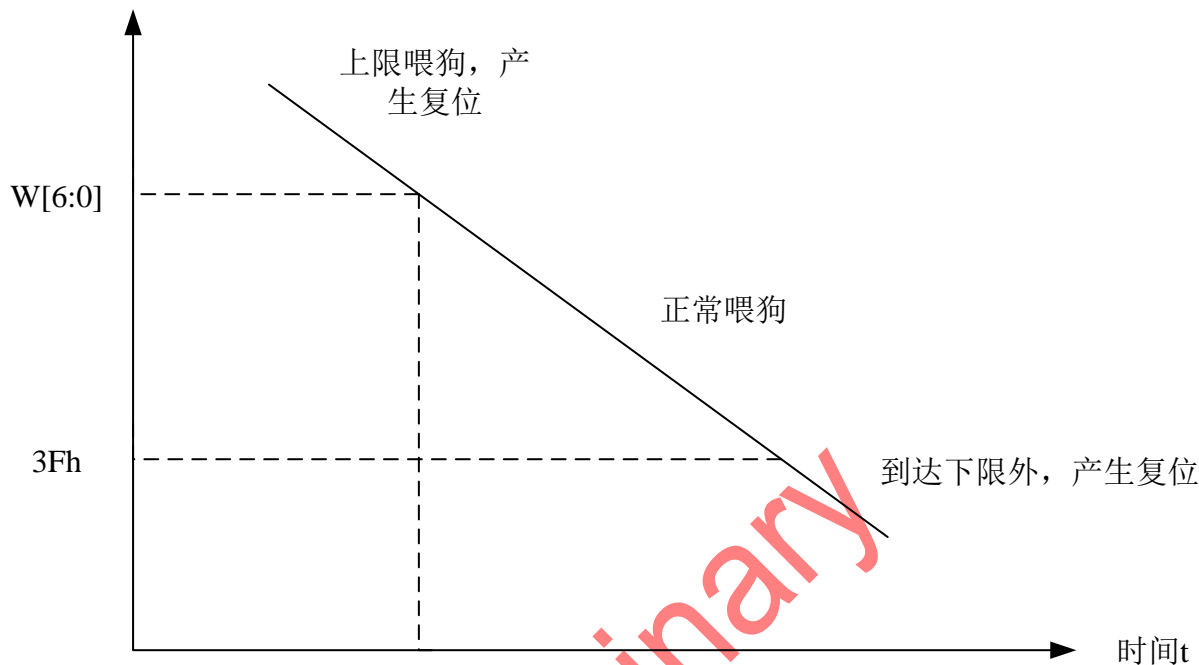


图 23-2 WWDG 超时时间坐标图

表 23-1 超时时间

时基 (WDGTB)	最短时间	最长时间
0	56.5us	3.64ms
1	113.5us	7.28ms
2	227.5us	14.56ms
3	455us	29.12ms

超时计算公式:

$$T = T_{\text{pclk}} * 4096 * 2^{\text{WDGTB}} * (\text{T}[5:0] + 1)$$

WDGTB: 分频器的时基

T_{pclk} : APB1 的时钟间隔

23.6 寄存器描述

表 23-2 WWDG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	WWDG_CR	控制寄存器	0x0000007F
0x04	WWDG_CFGR	配置寄存器	0x0000007F
0x08	WWDG_SR	状态寄存器	0x00000000

23.6.1 控制寄存器 (WWDG_CR)

偏移地址: 0x00

复位值: 0x0000 007F

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	WDGA	激活位 (Activation Bit) 此位由软件置 1, 但仅能由硬件在复位后清 0。当 WDGA 为 1 时, 看门狗被启动。 0: 关闭看门狗 1: 启动看门狗
6: 0	T[6: 0]	7 位递减计数器 (MSB 至 LSB) (7 - Bit Counter) 用来存储看门狗的计数器值。每 (4096×2^{WDGTB}) 个 PCLK1 周期减 1, 当计数器值从 40h 递减为 3Fh 时 (T[6]变成 0), 产生看门狗复位。

23.6.2 配置寄存器 (WWDG_CFGR)

偏移地址: 0x04

复位值: 0x0000 007F

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

Bit	Field	Description
31: 10	Reserved	保留, 必须保持复位值。
9	EWI	提前唤醒中断使能 (Early Wakeup Interrupt) 1: 当计数器值达到 40h, 产生中断。 0: 看门狗中断禁止 该位只能由硬件在复位后清除。

Bit	Field	Description
8: 7	WDGTB	时基 (Timer Base) 预分频器的时基可根据如下修改： 00: 计时器时钟 (PCLK1 除以 4096) 除以 1 01: 计时器时钟 (PCLK1 除以 4096) 除以 2 10: 计时器时钟 (PCLK1 除以 4096) 除以 4 11: 计时器时钟 (PCLK1 除以 4096) 除以 8
6: 0	W[6: 0]	7 位窗口值 (7-Bit Window Value) 窗口看门狗的上限窗口值。

23.6.3 状态寄存器 (WWDG_SR)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															EWIF
															w0c
Bit	Field	Description													
31: 1	Reserved	保留, 必须保持复位值。													
0	EWIF	EWIF: 提前唤醒中断标志 (Early Wakeup Interrupt Flag) 当计数器值达到 40h, 此位由硬件置 1; 通过软件写 0 来清除, 写 1 无效。													

24 IWDG 独立看门狗

24.1 简介

独立看门狗的设计初衷是为了检测 and 解决由软件错误所引起的故障，它的原理可简述为：当独立看门狗（IWDG）计数器不断递减到达给定数值时，产生一个系统复位信号使系统复位，从而提高系统整体安全性能。

独立看门狗适合应用于那些需要看门狗作为一个处于主程序之外，能够完全独立工作，并且对时间精度要求低的场合。

独立看门狗是由内部低速的时钟（LSI）驱动的，保证当主时钟发生故障的时候，独立看门狗依旧可以继续工作。

Preliminary

24.2 功能框图

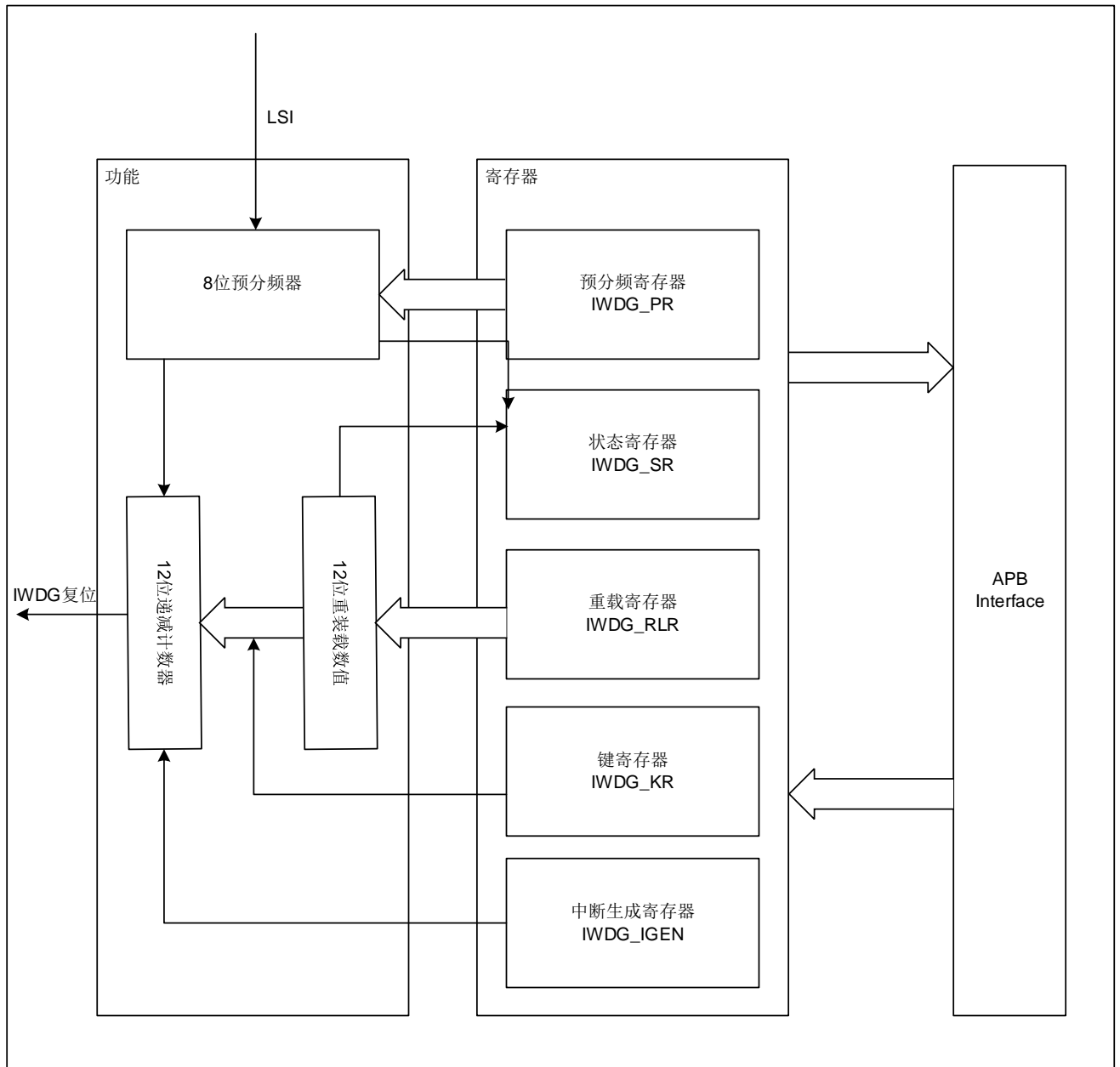


图 24-1 功能框图

24.3 主要特征

- 芯片默认为软件看门狗模式。
- 通过闪存烧写复位选项字节寄存器中的 WDG_SW 位可以启动硬件看门狗。硬件看门狗启动后在系统复位上电后自动启动，内部计数器开始递减。
- LSI 可在停机模式和待机模式下继续进行工作；

- 看门狗内部是自由运行的 12 位递减计数器，当计数到达 0x0000 产生一个系统复位或者中断信号。

24.4 功能描述

- 在键值寄存器 (IWDG_KR) 中写入 0XCCCC，开启独立看门狗。与此同时，计数器开始从其复位值 0Xfff 开始递减，当递减到达 0x0000 时会产生一个系统复位信号，或者递减到 IWDG_IGEN 的值后产生中断，两种结果取决于 IRQ_SEL 的配置。
- 任何时候写入 0XAAAA 到 IWDG_KR，就会把重载寄存器 (IWDG_RLR) 中的值重新加载到计数器中 (通常说的喂狗)，从而避免复位信号或中断信号的产生。
- 比较/输出如果程序异常，无法正常喂狗，就会产生复位信号或中断信号，系统复位或系统中断。
- IWDG_PR, IWDG_RLR, IWDG_IGEN 寄存器具有访问保护功能。只有在键值寄存器 (IWDG_KR) 写入 0x5555，才可以修改以上被保护的寄存器的值。当以其他的值写入键值寄存器，会打乱操作顺序，寄存器依旧处于保护状态。当进行重载操作时，也会处于保护状态。
- 独立看门狗的时钟由 LSI 提供，因此可以工作在停止和待机模式下。
- 独立看门狗可以在低功耗模式下正常计数，它的复位能够使系统退出 Standby 模式。
- 在低功耗模式下，可以通过配置 RCC 寄存器，选择在进入 Stop 模式后是否关闭 LSI 时钟，从而关闭软件看门狗。

24.4.1 流程框图

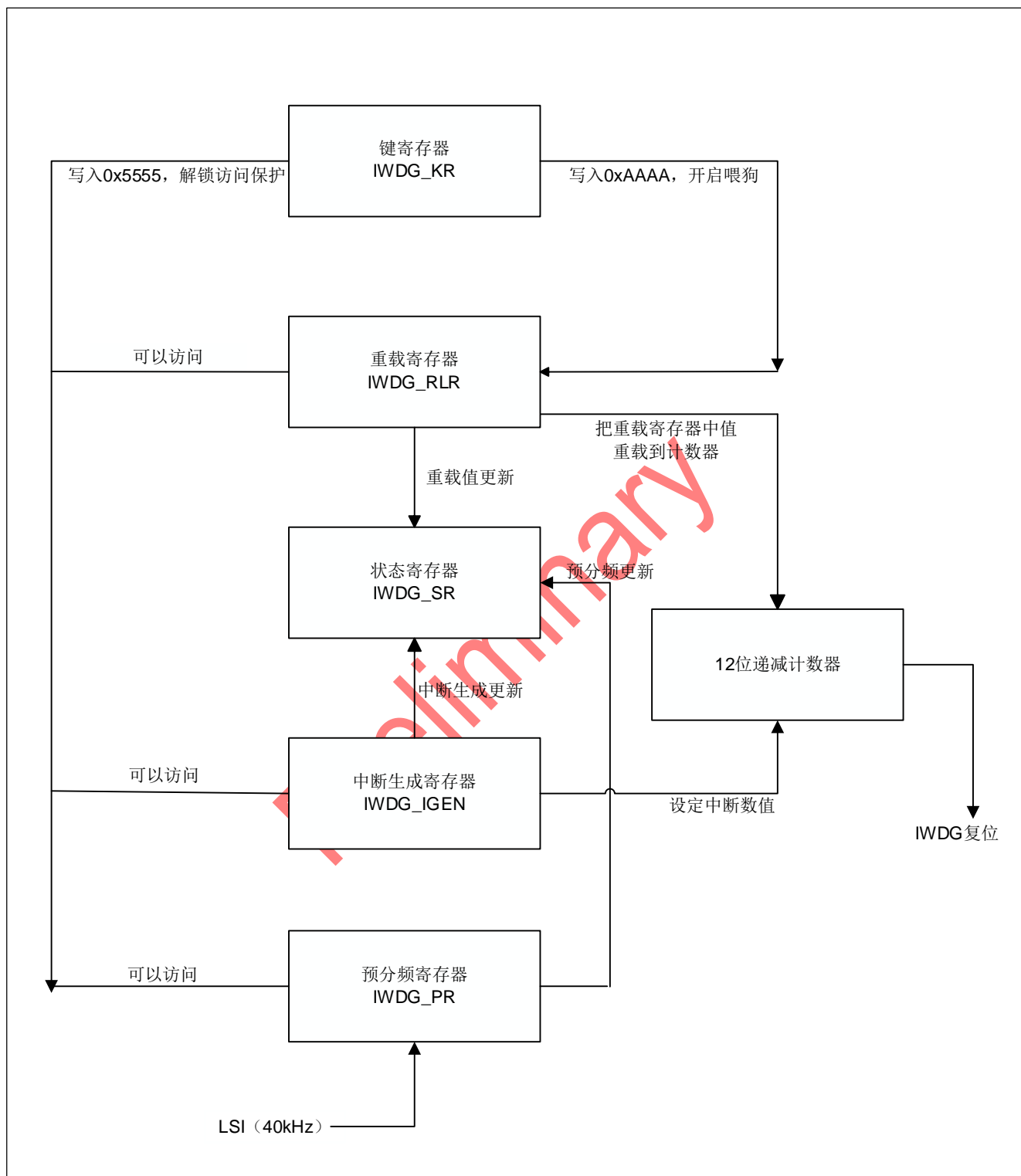


图 24-2 流程框图

24.4.2 独立看门狗超时时间

表 24-1 IWDG 超时时间 (40kHz 的输入时钟 (LSI))

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

超出（溢出）时间计算：

$$T_{out} = ((4 \times 2^{PR}) \times RLR) / 40$$

其中：T_{out} 的单位为毫秒。

时钟频率 LSI = 40K，一个看门狗时钟周期就是最短超时时间。

最长超时时间 = (IWDG_RLR 寄存器最大值) X 看门狗时钟周期。

24.4.3 中断

- IWDG 在 CR.IRQ_SEL 配置为 1 条件下，当计数器溢出后产生中断。

24.5 寄存器描述

24.5.1 寄存器总览

表 24-2 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
0x14	IWDG_IGEN	中断生成寄存器	0x00000FFF
0x18	IWDG_CNT	计数寄存器	0x00000001

24.5.2 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
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Res.

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）（Key value） 软件每隔一段时间，写进 0XAAAA 进行喂狗操作，否则当计数器递减到 0x0000 时，会产生一个复位信号，使系统复位。 软件写入 0x5555 表示解除保护，可以访问其他配置寄存器（IWDG_PR、IWDG_RLR、IWDG_CR（bit0），IWDG_IGEN） 软件写入 0xCCCC，开启看门狗。

24.5.3 IWDG_PR 预分频寄存器

偏移地址：0x04

复位值：0x0000 0000（在待机模式复位）

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
Res.													PR		
Res.													rw		

Bit	Field	Description
31: 3	Reserved	始终读为 0
2: 0	PR	预分频因子（Prescaler divider） 通过设置这些位来选择 LSI 时钟的预分频因子 要改变预分频因子，需要先解除保护（向 IWDG_KE 中写入 0x5555）才能写入，当预分频因子被更新完成后，PUV 寄存器位会变为 0，此时读出数据才是有效的。 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

24.5.4 IWDG_RLR 重装载寄存器

偏移地址：0x08

复位值：0x0000 0FFF（在待机模式复位）

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
Res.				RL											
Res.				rw											

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11: 0	RL	重装载值（Watchdog counter reload value） 配置看门狗计数器的重载值，每当喂狗（向 IWDG_KR 寄存器中写入 0xAAAA）时，会把此位数值更新到计时器中，然后从此值开始递减。更改重载值需要解除保护（向 IWDG_KR 寄存器中写入 0x5555），当重载值更新完毕后，RUV 寄存器位会清 0，也就在此时读出值才是有效的。看门狗的超时周期可以通过重装载值和预分频值来计算。

24.5.5 IWDG_SR 状态寄存器

偏移地址：0x0C

复位值：0x0000 0000（待机模式不复位）

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
Res.												UPDAT	IVU	RVU	PVU
Res.												r	r	r	r

Bit	Field	Description
31: 4	Reserved	保留，必须保持复位值
3	UPDATE	看门狗重装载值更新标志 当 IWDG_KR 寄存器中写入 0XAAAA 时，update 置位，当看门狗计数器被更新，重装载值写入到计数器中，update 自动清零。
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。

Bit	Field	Description
1	RVU	看门狗计数器重载值更新 (Watchdog counter reload value update) 重载值的更新正在进行中, 此位置 1。 当重载更新结束后, 此位清'0' (最多需要 5 个 40KHz 的振荡器周期) 重载值只有在 RVU 位被清'0'后才可更新。
0	PVU	看门狗预分频更新 (Watchdog prescaler value update) 预分频值的更新正在进行中时, 此位置 1。 当预分频值更新结束后, 此位由清'0' (最多需要 5 个 40KHz 的振荡器周期) 预分频值只有在 PVU 位被清'0'后才可更新。

注: 如果在应用程序中使用多个重载值、预分频值或中断生成值, 先解除寄存器保护 (向 IWDG_KR 中写入 0x5555), 然后配置 IWDG_PR IWDG_RLR, IWDG_IGEN 寄存器, 等待对应的状态寄存器位 (PVU, RVU, IVU) 清零, 表示已经配置好重载值, 预分频值, 和中断生成值, 接下来就可以执行喂狗操作, 或者等待计数器自动递减产生复位或者中断信号。

24.5.6 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_CL	IRQ_SE
														R	L
														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: 无效操作, 中断标志位依旧挂起 注意: 这个 bit 的写操作不需要设置 KEY 解除保护
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

24.5.7 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	IWDG 中断生成值（Watchdog Interrupt Generate value） 用于定义看门狗中断生成值，每当计数器值递减等于该值时，会产生中断。 要改变该位数值，需要先解除保护。 当改变数值被更新完成后 IWDG_SR 寄存器中的 IVU 位为 清 0 此时，读出数据才是有效的。

24.5.8 IWDG_CNT 计数寄存器

偏移地址：0x18

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													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，必须保持复位值
18: 8	IWDG_CNT	IWDG 计数器 counter 的值
7: 0	IWDG_PS	IWDG 时钟分频计数器的值

25 UART 通用异步收发器

25.1 简介

通用异步收发器 (UART) 可以灵活地与外部设备进行全双工数据交换。通过分数波特率发生器, UART 可以选择宽范围的波特率。异步单向通信和半双工单线通信, 以及调制解调器 (CTS/RTS) 操作、IrDA 红外功能也能够被支持。另外, UART 也支持多处理器之间的通信。

对于高速数据通信, 可以通过使用多缓冲器配置的 DMA 方式来实现。

25.2 功能框图

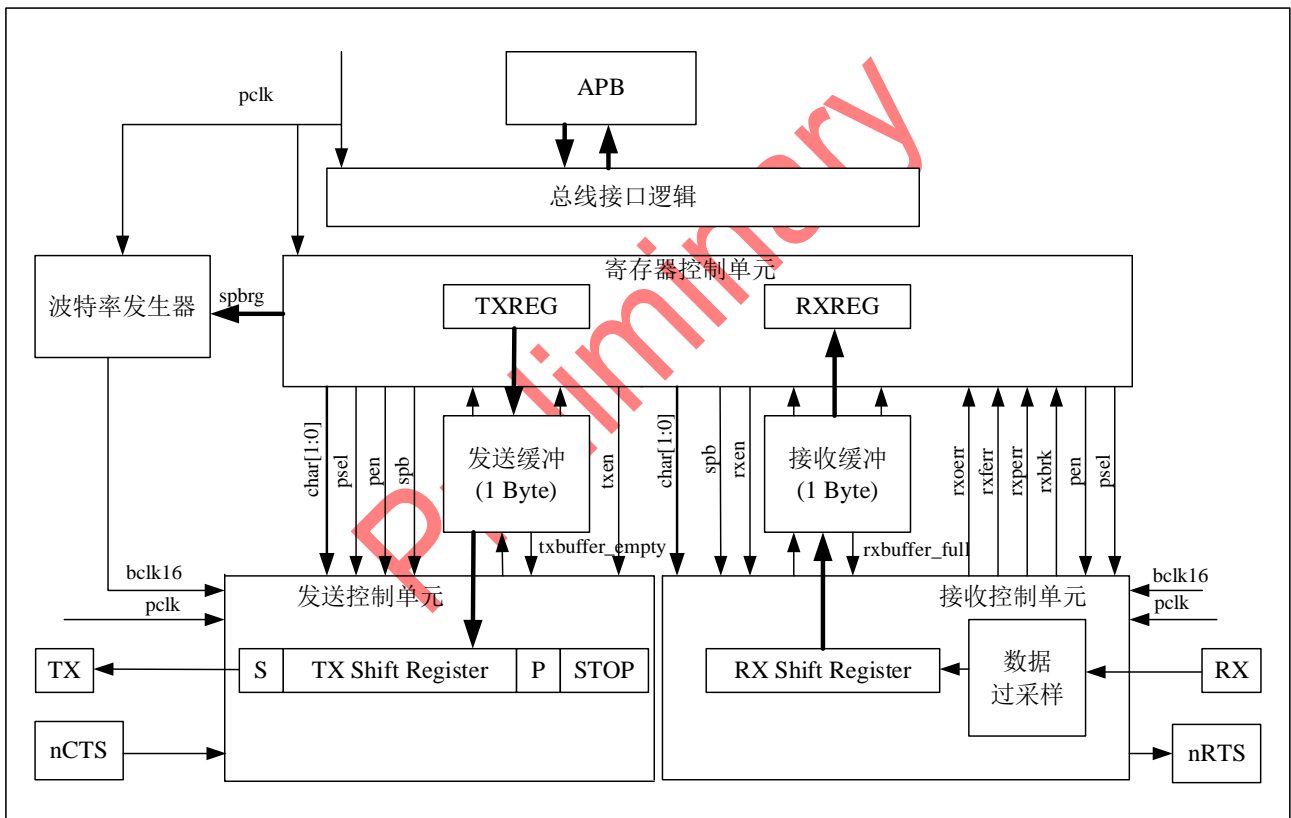


图 25-1 UART 功能框图

25.3 主要特征

- 支持异步方式下 RS-232S 协议, 符合异步串口通信 16550 芯片标准
- 支持 DMA 请求
- 全双工异步操作
- 分数波特率发生器

- 可配置波特率，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
- 内置 1 字节发送和 1 字节接收缓冲
- 发送和接收数据低位在前，一个起始位开始，后面接数据位，数据长度可为 5~8 位，最后为停止位；另外可选择是否有奇偶校验位（奇偶校验位在数据位之后，停止位之前）
- 第 9 位可做同步帧配置
- 支持硬件奇/偶校验的产生和检测
- 断开符号的产生和检测
- 空闲符号的产生和检测
- 支持 LIN 协议下收发断开符号
- 支持信号收发互换，接收和发送取反
- 支持波特率自适应功能
- 支持硬件自动流控制
- 支持 IrDA SIR ENDEC 规范的红外功能

25.4 功能概述

在全双工通信的情况下，至少需要分配两个引脚给 **UART**：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 **UART** 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：**UART** 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。当发送器被除能时，**TX** 引脚恢复到它的 I/O 端口配置状态。

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。

一个数据（5，6，7 或 8 位），发送和接收顺序为从最低位（LSB）到最高位。

停止位用‘1’表示一帧的结束，位数可配置为 0.5、1、1.5、2 位。

分数波特率发生器：可配置 16 位的整数以及 4 位的小数，从而提高波特率的精度。

通过使能 **UART_GCR** 寄存器的 **SWAP** 位，可以交换接收和发送端的信号。

通过使能 **UART_GCR** 寄存器的 **RXTOG/ TXTOG** 位，可以将接收/发送端的信号取反（包含起始位、停止位也都被取反后再输入/输出）。

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。

25.4.1 UART 特性描述

通过配置 UART_CCR.CHAR 位，可调整字符位长度为 5~8 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，定义为一个空闲符号。下一个数据帧的起始位跟在空闲符号之后。

包括停止位在内，一个完全由‘0’组成的完整数据帧，定义为一个断开符号。在断开符号结束后，发送端会再发送一个‘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和接收器的使能位置位控制之后，供给发送器或接收器使用。

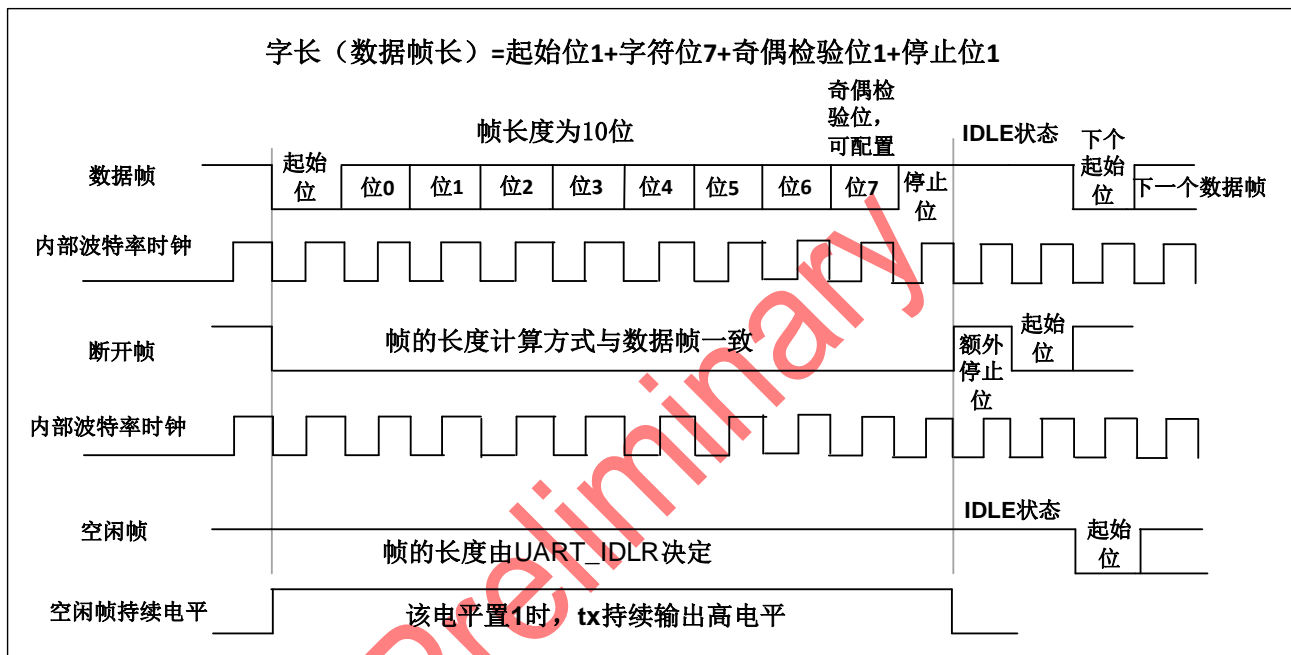


图 25-2 UART 时序

25.4.2 分数波特率发生器

设置 BRR 和 FRA 寄存器，可设置相应波特率，参考如下公式：

$$f_{baudrate} = \frac{f_{PCLK}}{16 \times UARTDIV}$$

$$UARTDIV = BRR + \frac{FRA}{16}$$

$$f_{baudrate} = \frac{f_{PCLK}}{16 \times BRR + FRA}$$

其中 BRR 寄存器最小值为 1，FRA 的值可配置在 0~15 范围。

25.4.3 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 16 倍数据波特率的

时钟采样 RX 引脚的数据，取中间第 7，8，9 个时钟下降沿的采样值。

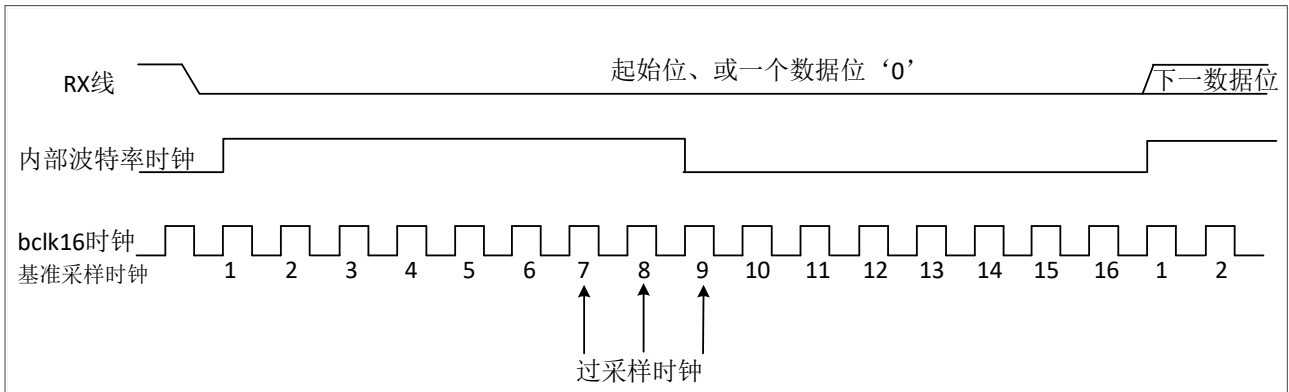


图 25-3 RX 引脚采样方法

25.4.4 容忍度

时钟偏差等因素会导致收发双方波特率的偏差，为了提高接收端对波特率变化的容忍度，在 STOP 状态时，当三次数据采样完成后，接收端就开始准备下一帧数据的接收。当收发双方波特率偏差处于 $\pm 3\%$ 以内时，仍然能保证数据正常传输。

25.4.5 校验控制

奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）可以通过配置 UART_CCR.PEN 位为‘1’来激活。如果奇偶校验出错，无效数据仍然会从移位寄存器传输到 UART_RDR 寄存器。配置 UART_ICR.RXPERR 位为‘1’来清除 UART_ISR.RXPERR_INTF 标志。

智能卡自动应答有效（UART_SCR.SCAEN=‘1’）时，在 STOP 状态检测到奇偶校验错误，会产生 1 位的低电平信号（RXNACK），指示帧错误。

偶校验：校验位加上数据中 1 的个数为偶数。

奇校验：校验位加上数据中 1 的个数为奇数。

奇偶校验位的输出或接收判断，由硬件自动完成，无需软件配置。

例如，数据为 8'b1000_0110，有 3 个‘1’。

偶校验（UART_CCR.PSEL =‘1’）：校验位将是‘1’。

奇校验（UART_CCR.PSEL =‘0’）：校验位将是‘0’。

奇偶校验使能时，奇偶校验位的发送跟在数据 MSB 位之后；接收时奇偶校验错误，硬件自动置位 UART_ISR.RXPERR_INTF 标志。若想产生中断，需要提前配置中断使能寄存器。

注意：如果使能 9bit 功能，则奇偶检验无效。

25.4.6 发送器

UART_CCR.CHAR 位的配置决定发送数据的位数（5~8 位）。配置 UART_CCR.TXEN 位为‘1’，使能

发送器，数据会串行输出到 TX 引脚上。

25.4.6.1 字符发送

在 UART 发送期间，数据从 UART_TDR 寄存器写入，经过一字节的缓冲器缓冲，通过发送移位寄存器以最低字节到最高字节的顺序，串行在 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。

通过配置 UART_CCR.CHAR[1: 0] 位来配置字符长度；通过配置 UART_CCR.SPB[1: 0] 位来配置停止位的位数。

当前数据输出未完成前不能清零 UART_GCR.TXEN 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。

25.4.6.2 配置步骤

1. 配置 UART_GCR.UARTEN 位为‘1’，使能 UART。
2. 配置 UART_CCR.CHAR[1: 0] 位，设置字符长度。
3. 配置 UART_CCR.SPB[1: 0] 位，设置停止位的位数。
4. 配置 UART_BRR 和 UART_FRA，产生需要的波特率时钟。
5. 配置 UART_GCR.TXEN 位为‘1’，使能发送器。

6. 将待发送数据写入 UART_TDR 寄存器（此时硬件自动清‘0’UART_ISR.TX_INTF 位）。由于缓冲器为一字节，每次只能写入一个数据，且需要等待缓冲器为空时才能写 UART_TDR 寄存器。

25.4.6.3 通信过程

UART_ISR.TX_INTF 只能由硬件写‘1’和清‘0’。当该位被硬件置‘1’时，表明：

1. 数据发送已经开始，数据已经从 UART_TDR 寄存器传输到移位寄存器
2. 可向 UART_TDR 寄存器写入下一个传输数据（不会影响当前数据的输出）

UART 发送器处于空闲状态时，软件写 UART_TDR 寄存器，数据被立即搬运至移位寄存器进行串行输出，且置位 UART_ISR.TX_INTF 标志和 UART_CSR.TXEPT 标志；如果 UART_IER.TX_IEN 位已配置为‘1’，会产生中断。

可以通过 TX_INTF 中断向 UART_TDR 寄存器写入下一个传输数据，实现连续发送。

当停止位发送完成，且没有写入新的发送数据，会置位 UART_ISR.TXC_INTF 标志和 UART_CSR.TXC 标志，表明发送已完成；如果 UART_IER.TXC_IEN 位已配置为‘1’，也会产生中断。

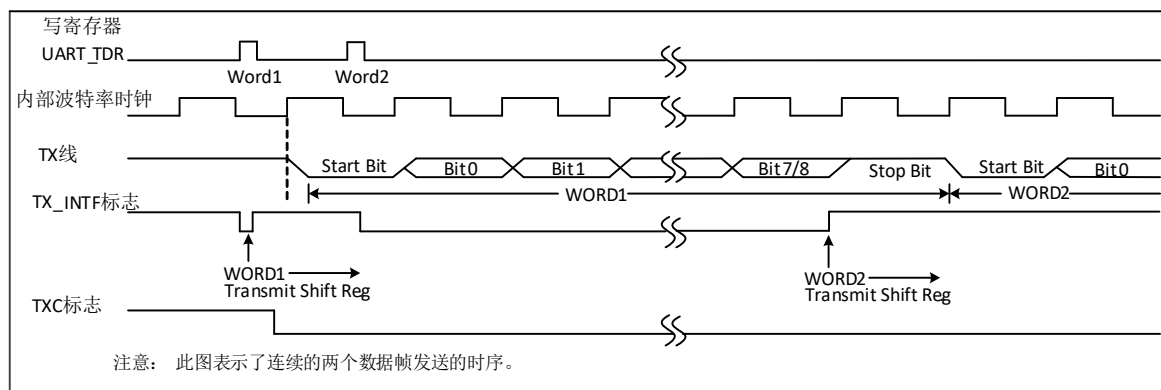


图 25-4 发送时状态位变化

25.4.6.4 断开符号

断开符号的发送只能在 `UART_CSR.TXC` 为 1 时发起。配置 `UART_CCR.BRK` 位为 '1' 即可发送断开符号。断开符号发送完成后，硬件自动清 '0' `BRK` 位，并发送一位的高电平用于接收端下一帧数据起始位的检测。

断开符号的总长度由 '起始位 + 数据位 + (校验位) + 停止位' 构成，长度不固定，会随着数据位、校验位、停止位的设置改变。断开符号发送完成时，会置位 `UART_ISR.TXBRK_INTF` 标志。

25.4.6.5 空闲符号

空闲符号的长度由 `UART_IDLR` 寄存器的值决定。当 `UART_IDLR` 不为 0 时，在使能 `TXEN` 后，会立即发送一帧空闲符号，再进行有效数据的输出。

25.4.7 接收器

25.4.7.1 字符接收

对于 `UART` 的接收器，`RX` 串行输入数据，接收顺序从数据最低位到最高位。内置一个接收移位寄存器和一个接收数据寄存器形成双缓冲结构，从而可以实现数据的连续接收。

配置步骤：

1. 配置 `UART_GCR.UARTEN` 位为 1，使能 `UART`。
2. 配置 `UART_CCR.CHAR` 位来设置字符长度。
3. 配置 `UART_CCR.SPB` 位来设置停止位的位数。
4. 配置 `UART_BRR` 和 `UART_FRA`，产生需要的波特率时钟。
5. 配置 `UART_GCR.RXEN` 位为 1，使能接收器，等待检测起始位。
6. 当接收到一个完整帧时，接收移位寄存器的数据被传输到 `RDR` 寄存器，硬件置位 `UART_ISR.RX_INTF` 标志和 `UART_CSR.RXAVL` 标志，表明数据已经被接收并且可以被读出。如果硬件

检测到数据接收错误时，会置位相应的错误标志。

7. 通过配置 `UART_IER.RX_IEN` 位为‘1’，使能接收中断。
8. 在接收期间如果检测到帧错误、溢出错误或奇偶校验错误，硬件置位相应的错误标志。
9. 软件读 `UART_RDR` 寄存器，或配置 `UART_ICR.RX_ICLR` 位为‘0’，实现清‘0’`UART_ISR.RX_INTF` 位。为了防止溢出错误，在下一字符接收结束前 `RX_INTF` 位必须被清零。

在接收数据时，软件不应该清零 `RXEN` 位，否则还在 `RX` 引脚上的数据将不再被接收。

25.4.7.2 断开符号

`UART` 接收器识别到一个断开符号时，会置位 `RXBRK_INTF` 标志，且数据不会被写入接收缓冲器。断开符号由‘起始位+数据位+（校验位）+停止位’构成，长度不固定，会随着数据位、校验位、停止位设置而变化。

25.4.7.3 空闲符号

空闲符号长度由 `UART_IDLR` 寄存器决定（默认值为 `0x0C`）。当 `UART_IDLR` 不为 0 时，接收到一个空闲符号，硬件会置位 `UART_ISR.RXIDLE_INTF` 标志，数据不会被写入接收缓冲器。

25.4.7.4 溢出错误

如果 `UART_RDR` 没有被读出，又接收到一个字符则发生溢出错误。当产生溢出错误时，表明：

1. 硬件将置位 `UART_ISR.RXOERR_INTF` 标志。
2. 未读出的 `UART_RDR` 内容不会丢失，读 `UART_RDR` 寄存器仍然得到此数据。
3. 后接收到的字符将被丢失。

如果 `UART_IER.RXOERR_IEN` 位已配置为‘1’，会产生中断。

25.4.7.5 帧错误

当停止位没有在预期的时间上被识别出，接收器会产生帧错误。当帧错误被检测到时，表明：

1. 硬件将置位 `UART_ISR.RXFERR_INTF` 标志。
2. `UART_RDR` 寄存器不会被更新。

如果 `UART_IER.RXFERR_IEN` 位已配置为‘1’，会产生中断。

25.4.8 自动波特率检测

`UART` 能够根据接收到的信号，自动检测并重置 `UART_BRR` 寄存器。自动波特率配置如下：

1. 配置 `UART_ABRCCR` 寄存器的 `Former_edge` 和 `Latter_edge`，选择自动波特率开始及结束的边沿方式（上升或下降）。

2. 配置 UART_ABRCR.Abr_bitcnt, 选择自动波特率的数据位长度。
3. 配置 UART_ABRCR.Abren 位为‘1’, 打开自动波特率功能。
4. 配置 UART_IER.ABREND_IEN 位为‘1’, 自动波特率结束后会产生中断。
5. 配置 UART_IER.ABRERR_IEN 位为‘1’, 自动波特率检测发生错误, 会产生错误中断。

示例 1: 设置 Former_edge=1 (上升沿), Latter_edge=0 (下降沿), Abr_bitcnt=2 (4 位长度)。

当接收到数据 0x0F 时, 将对 RX 线上数据 (起始位->MSB 位: 0_1111_0000) 的低四位 0xF 进行波特率检测, 然后重置 UART_BRR 寄存器, 并置位 UART_ISR.ABREND_INTF 标志。

示例 2: 设置 Former_edge=0 (下降沿), Latter_edge=1 (上升沿), Abr_bitcnt=2 (4 位长度)。

当接收到数据 0x08 时, 将对 RX 线上数据 (起始位->MSB 位: 0_0001_0000) 的‘起始位+最低三位数据’进行波特率检测, 然后重置 UART_BRR 寄存器, 并置位 UART_ISR.ABREND_INTF 标志。

注意: 检测边沿和数据长度的配置, 应该与接收数据相匹配。例如, 当 Former_edge=1, 而接收数据的最低位不为‘1’, 硬件会置位 UART_ISR.ABRERR_INTF 标志。

另外, 当自动波特率检测开始前, 通过软件初始配置的波特率与发送侧波特率相差过大, 可能会导致自动波特率功能失效。

25.4.9 九位数据通信

通过配置 UART_CCR 寄存器的相关配置, 来实现 9 位数据通信。具体如下:

B8EN 为‘1’, 将使能 9 位数据的发送和接收。

在 B8EN 使能后, 奇偶校验使能位 PEN 无效。

数据发送时, 需要先设置 B8TXD; B8TXD 作为发送数据的 MSB 和 UART_TDR 的值一起发送。B8TXD 与 B8POL 相同时, 表示该数据作为地址帧或者同步帧。B8TOG 为‘1’, 当前帧发送结束后 B8TXD 会自动翻转, 下一个数据帧无需再配置 B8TXD。

数据接收时, 接收数据的第 9 位可以从 B8RXD 读到; B8RXD 与 B8POL 相同时, 硬件置位 UART_ISR.RXB8_INTF 标志。

25.4.10 多处理器通信

多个处理器通信可以通过将多个 UART 连在一个网络里实现。作为主设备的 UART, 它的 TX 输出连接其余从设备的 RX 输入; 而其余从设备的 TX 输出逻辑相与之后, 连接到主设备的 RX 输入。

主机通过寻址方式找到目的从机, 而其他非目的从机应该保持在一个较低功耗的模式, 以避免资源浪费。

当从设备不是目的从机时, 配置此从机进入静默模式来降低功耗。在静默模式里:

1. 硬件不会置位所有的接收相关状态标志位。
2. 不会产生所有接收相关的中断。

软硬件均可改写 UART_CCR.RWU 位。配置 UART_CCR.RWU 位为‘1’, 使接收器进入静默模式。在

设置地址标记唤醒时，如果接收缓冲器非空（接收地址未匹配从机），则软件不能配置 RWU 位为‘0’来退出静默模式。

通过配置 UART_CCR.WAKE 位，选择退出静默模式的检测方式：

1. WAKE=0：空闲总线检测。
2. WAKE=1：地址标记检测。

25.4.10.1 空闲总线检测

静默模式下，UART_CCR.WAKE=0，从设备进行空闲符号检测。如果在 RX 线上检测空闲符号，硬件在自动清‘0’UART_CCR.RWU 位，并退出静默模式。由于没有正常帧输入，硬件不会置位 UART_ISR.RX_INTF 标志。

25.4.10.2 地址标记（Address Mark）检测

静默模式下，UART_CCR.WAKE=1，从设备进行地址标记检测。

接收字节有效位的最高位作为地址帧或数据帧的识别标记，因此，最高位等于 B8POL 则代表该字节为地址帧；反之，则为数据帧。

从机接收器将 UART_RXADDR 寄存器的配置值作为该从机地址，与接收到的地址帧比较。应注意，UART_RXMASK 寄存器具备地址掩码功能，只有当它某一位为 1 时，相应的地址位才会参与比较；为 0 则直接认为该位匹配成功。

若从机接收的地址帧经过地址掩码后匹配失败，硬件置位 UART_CCR.RWU 位并进入静默模式。由于本从机处于静默模式，接收到的地址帧将不会导致相关的硬件响应（比如触发标志位、中断请求、DMA 读取请求等）。

若从机接收的地址帧经过地址掩码后匹配成功，硬件自动清‘0’UART_CCR.RWU 位并退出静默模式。由于本从机已经退出静默模式，此地址帧会触发 UART_ISR.RX_INTF 标志，之后接收器正常工作。

25.4.10.3 配置步骤

多处理器通信的实现，建立在九位数据通信使能的基础上，需以下步骤。

主机发送时，可参考（发送器）进行，另注意如下：

1. 在使能 TXEN 前，配置 B8EN 位为‘1’，并配置 B8TXD 与 B8POL 位。如果配置 B8TXD=B8POL=1，从设备检测到最高位的高电平认为是地址帧；如果配置 B8TXD=B8POL=0，从设备检测到最高位的低电平认为是地址帧。此时如果配置 B8TOG 位为‘1’，无需步骤 3 中的 B8TXD 配置。
2. 使能 TXEN 后，再写 TDR 数据作为地址值，与 B8TXD 一起作为地址帧发送到 TX 线上。
3. 之后再配置 B8TXD 为 B8POL 的相反值，并写 TDR，将紧接着发送数据帧。
4. 根据发送的数据类型（地址帧或数据帧），按照步骤 3 所示操作，改写 B8TXD 和 TDR，实现主机的连续发送（地址帧+数据帧+... ..）。

从机接收时，可参考（接收器）进行，另注意如下：

1. 在使能 RXEN 前，配置 B8EN 位为‘1’，同时配置 B8POL 位（作为判断当前字符为地址帧或数据帧的标志，与 B8RXD 位进行比较，相同则认为是地址帧，否则为数据帧）。
2. 在使能 RXEN 前，还需配置 WAKE 位来选择从机退出静默模式的检测方式。如果配置 WAKE=1，还需配置本机的从机地址（配置 RXADDR 和 RXMASK 寄存器）。
3. 使能 RXEN 后，如果 WAKE=0，通过写 RWU=1 使从机进入静默模式，之后从机接收器将在检测到空闲帧时自动退出静默模式（硬件会清‘0’RWU 位）。
4. 使能 RXEN 后，如果 WAKE=1，可通过软件配置 RWU 位为‘1’使从机进入静默模式，之后从机接收器将进行地址标记检测。另外硬件将会在地址匹配失败时置位 RWU 并进入静默模式。

25.4.11 单线半双工通信

配置 UART_SCR.HDSEL 位为‘1’，选择单线半双工模式，芯片内部逻辑会将 TX 与 RX 互连。需注意 UART_SCR.SCEN 位应配置为‘0’。

单线半双工时：

1. RX 引脚悬空，不参与传输。UART 的 TX 直接连接另一个 UART 的 TX。
2. 在传输数据时，TX 一直被占用，直到停止位发送完成。
3. 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 UART 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一致。

在没有通信前，两个 UART 的 UART_GCR.RXEN 都开启，处于等待接收状态。当需要通信时，两个 UART 需要约定好谁来发送；发送方 RXEN 关闭，TXEN 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 UART 的发送：当发送使能位 TXEN 开启，只要写 UART_TDR，TX 就会发送数据）。

25.4.12 智能卡

配置 UART_SCR.SCEN 位为‘1’进入智能卡模式，硬件将自动为半双工模式（不用设置 UART_SCR.HDSEL 位）。

该模式下的 UART 符合 ISO7816-3 标准，在 TXEN 或 RXEN 使能前还需以下步骤：

1. 8 位数据位+校验位：UART_CCR 寄存器配置 CHAR=11、PEN=1。
2. 发送和接收均推荐为 1.5 个停止位：UART_CCR 寄存器配置 SPB1=1、SPB0=1。
3. 如需使能 NACK 信号：UART_SCR 寄存器配置 SCAEN=1。
4. 智能卡保护时间配置：UART_SCR 寄存器配置 SCFCNT。

下图举例说明，有校验错误和无校验错误的两种情况下，TX 线上信号区别。

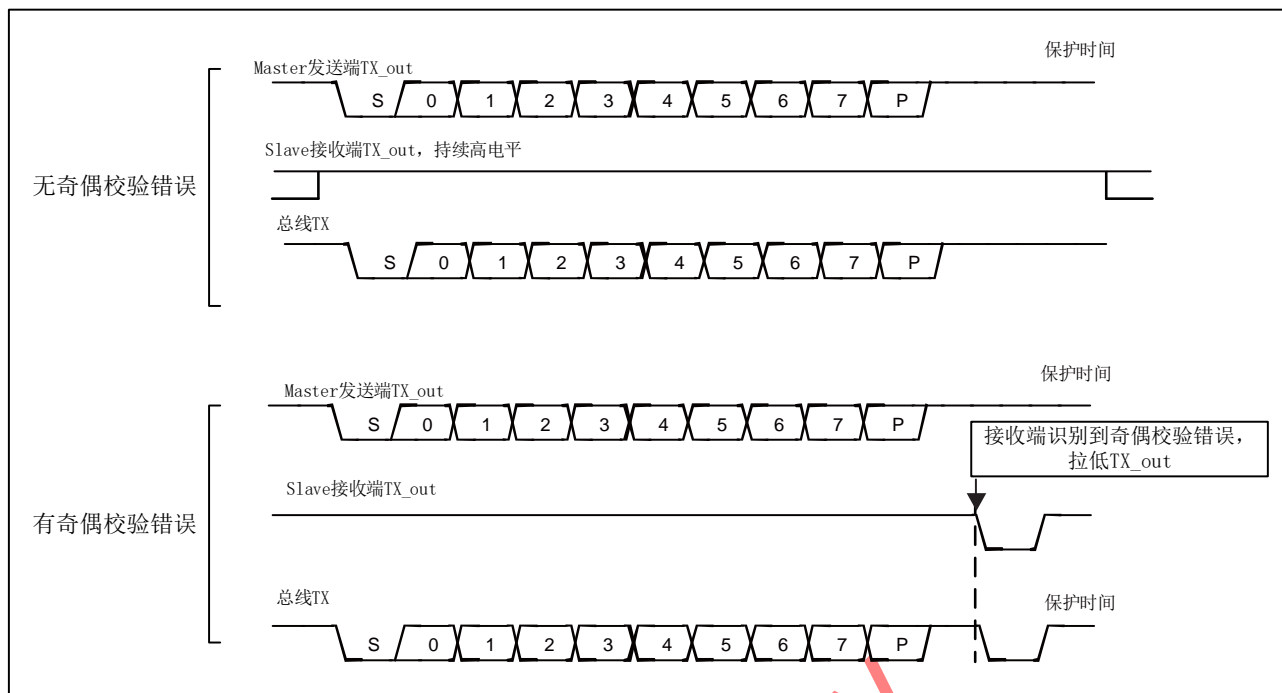


图 25-5 UART 奇偶校验方框图

UART 作为主机，为了能够识别从机方向反馈的校验错误信号（1 位低电平：NACK 信号），UART 发送数据时需使能 TXEN，而在停止位发送完毕后应该除能 TXEN，同时 TX 被拉到高电平（TX 应被配置成开漏模式）。

UART 作为主机，需要用 TX 单线驱动智能卡端的双向线，TX 与 RX 实质上使用的同一个 IO 口。

智能卡通信协议：

1. 从机数据被传输至移位寄存器，再到 TX 引脚上串行输出，相比正常模式下的发送时间会多至少 1/2 波特时钟周期。正常模式下，TX 引脚出现数据的时间是在发送移位寄存器满时的下一个波特时钟沿；而智能卡模式 TX 引脚出现数据的时间会比正常模式慢 1/2 波特时钟。

2. 从机接收器在数据帧的停止位（推荐为 1.5 位）接收期间，检测到了奇偶校验错误，会拉低数据线一个波特时钟周期（NACK 信号）。由于数据线被拉低，相当于主机发送了一个帧错误（停止位被拉低成 '0'）。主机收到反馈后，软件利用这个反馈信号可根据协议处理重发数据。注意，需要输出该 NACK 信号，应配置 UART_SCR.SCAEN 位为 '1'。

3. 智能卡模式下存在保护时间，即通过配置 UART_SCR.SCFcnt[7: 0] 来延长 UART_CSR.TXC 标志的置位时间；奇偶校验结束后，对应的保护时间计数器就会启动累加，当计数器的值累加到等于配置的保护时间寄存器的值时，TXC 标志才会被置位。而在正常传输模式（非智能卡）下，硬件置位 TXC 的条件是发送缓冲器为空且没有后续数据写入 TDR。

4. 智能卡模式不会影响硬件写 TDR 时，自动清 '0' TXC 标志的时序。

5. 如果主机收到从机的 NACK 信号，主机的接收器不会把 NACK 当作起始位检测。

6. 从机接收器检测到校验错误后，其反馈给主机的 NACK 信号也不会被认为是起始位。

注意：

a. 智能卡模式中没有断开符号的定义；一个包括停止位在内的全 '0' 数据帧将被识别为帧错误。

b. 智能卡模式中没有空闲符号的定义；当来回切换 TXEN 位时，不会认为接收到空闲符号。

下图说明了 UART 对于 NACK 信号的采样逻辑，使能 UART 的接收器，用于检查数据的完整性和 NACK 信号（停止位为 1.5 位，其中 1 个 NACK 位处于停止位的后段）。

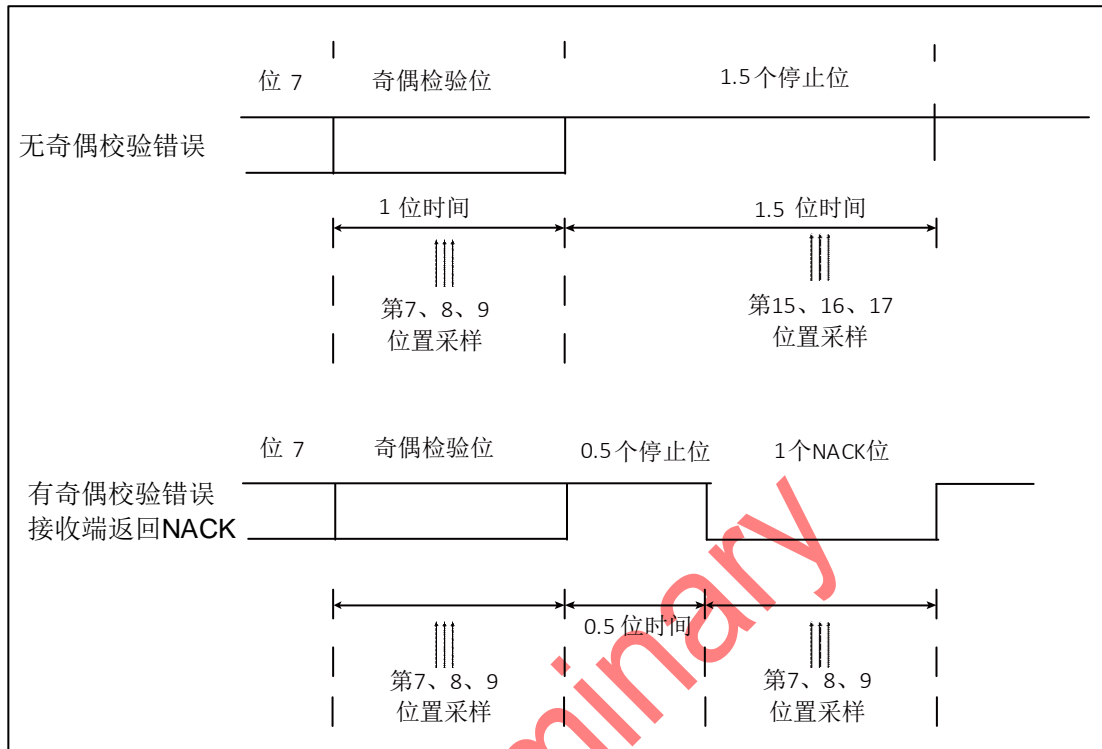


图 25-6 UART 采样 NACK 信号方框图

25.4.13 红外 IrDA 功能

UART 模块支持 IrDA (infrared data association) SIR ENDEC 规范；IrDA SIR 物理层规定使用反相归零调制方案 (RZI)。

该方案用一个红外光脉冲代表逻辑“0”。正常模式下，“0”的脉宽为 3/16 的波特率；低功耗模式下，频率范围应配置在 ($1.42 \text{ MHz} < \text{sirlp_clk} < 2.12 \text{ MHz}$)，脉冲的宽度是 $\text{sirlp_clk} * 3$ ，sirlp_clk 可通过配置 UART_IRDA.PSC_REG[7: 0] 来将系统时钟 pclk 分频以达到这个值。

SIR 发送编码器和 SIR 接收解码器实现 UART 数据流与红外脉冲流的转换。

IrDA 是一个半双工通信协议，编解码不可同时进行；在 IrDA 模式里，UART_CCRL 寄存器上的停止位必须配置成 1 个。

在 TXEN 或 RXEN 使能前还需以下步骤：

1. 使能红外功能：UART_IRDA 寄存器配置 Siren=1。
2. 低功耗模式时：提前配置 UART_IRDA 寄存器 PSC_REG 和 Sirlp=1。

下图示意了数据格式转换的样式：

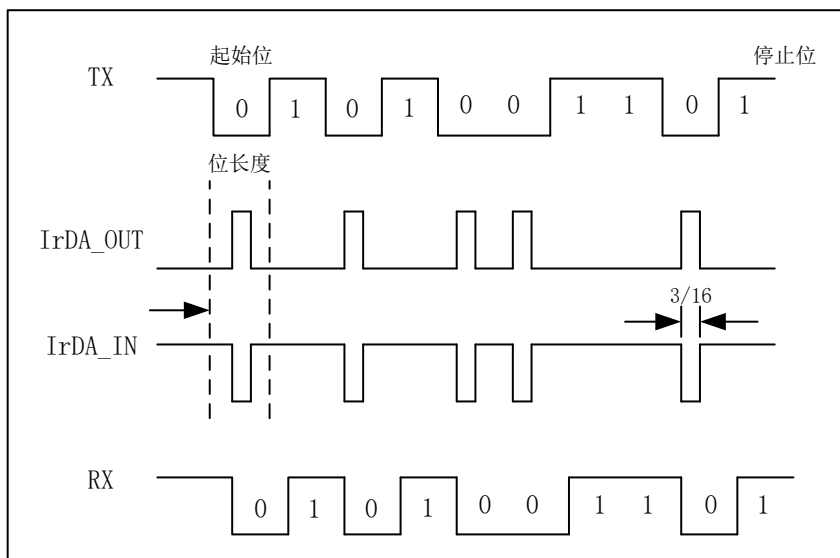


图 25-7 普通模式下 IrDA 发送和接收图

25.5 中断

支持下面中断源：

- 发送缓冲空
- 接收数据有效
- 接收溢出错误
- 帧错误
- 奇偶校验错误
- 接收断开帧
- 发送移位寄存器完成
- 发送断开帧完成
- 接收同步帧
- 检测到空闲帧
- 自动波特率结束
- 自动波特率错误

表 25-1 UART 中断请求

中断事件	中断状态	使能位
发送缓冲空	TX_INTF	TXIEN
接收到有效数据	RX_INTF	RXIEN
发送移位寄存器完成	TXC_INTF	TXC_EN
接收溢出错误	RXOERR_INTF	RXOERREN
奇偶校验错误	RXPERR_INTF	RXPERREN
帧错误	RXFERR_INTF	RXFERREN

中断事件	中断状态	使能位
接收断开帧	RXBRK_INTF	RXBRKEN
发送断开帧	TXBRK_INTF	TXBRK_EN
接收同步帧	RXB8_INTF	RXB8_EN
接收空闲帧	RXIDLE_INTF	RXIDLEN
自动波特率结束	ABREND_INTF	ABRENDIEN
自动波特率错误	ABRERR_INTF	ABRERRIEN

如果设置了对应的中断使能控制位，这些设置就可以产生各自对应的中断。

25.6 DMA

UART 可以利用 DMA 来搬运数据；使能 UART_GCR.DMAMODE 位激活 DMA 模式。

利用 DMA 发送：DMA 搬送数据到 UART_TDR。

使能 UARTEN, TXEN, DAMMODE 后，只要 TXFIFO 为空，就请求 DMA 发送。

在利用 DMA 发送时，需要提前配置好 DMA 的源地址（存储器地址）和目的地址（UART_TDR），传输的数据量以及 DMA 通道。

利用 DMA 接收：DMA 将 UART_RDR 数据搬走。

使能 UARTEN, RXEN, DMAMODE 后，只要 RXFIFO 有数据，即不为空，就请求 DMA 接收。

在利用 DMA 接收时，需要提前配置好 DMA 的源地址（UART_RDR）和目的地址（存储器地址），传输的数据量以及 DMA 通道。

25.7 寄存器

25.7.1 寄存器总览

表 25-2 UART 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	UART_TDR	UART 发送数据寄存器	0x00000000
0x04	UART_RDR	UART 接收数据寄存器	0x00000000
0x08	UART_CSR	UART 当前状态寄存器	0x00000009
0x0C	UART_ISR	UART 中断状态寄存器	0x00000000
0x10	UART_IER	UART 中断使能寄存器	0x00000000
0x14	UART_ICR	UART 中断清除寄存器	0x00000000
0x18	UART_GCR	UART 全局控制寄存器	0x00000000
0x1C	UART_CCR	UART 通用控制寄存器	0x00000000
0x20	UART_BRR	UART 波特率寄存器	0x00000001
0x24	UART_FRA	UART 分数波特率寄存器	0x00000000
0x28	UART_RXADDR	UART 接收地址寄存器	0x00000000
0x2C	UART_RXMASK	UART 接收掩码寄存器	0x000000FF

Offset	Acronym	Register Name	Reset
0x30	UART_SCR	UART SCR 寄存器	0x00000000
0x34	UART_IDLR	UART IDLE 数据长度寄存器	0x0000000C
0x38	UART_ABRCR	UART ABRCR 自动波特率控制寄存器	0x00000000
0x3C	UART_IRDA	UART IRDA 红外功能控制寄存器	0x00000100

25.7.2 UART_TDR 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							TXREG								
rw															
Bit	Field	Description													
31: 9	Reserved	保留, 始终读为 0													
8: 0	TXREG	发送数据寄存器 (Transmit data register)													

25.7.3 UART_RDR 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							RXREG								
r															
Bit	Field	Description													
31: 9	Reserved	保留, 始终读为 0													
8: 0	RXREG	接收数据寄存器 (Receive data register) 该寄存器只读													

25.7.4 UART_CSR 当前状态寄存器

偏移地址: 0x08

复位值：0x0000 0009

31	30	29	28	27	26	25	24	23	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.												TXEPT	TXFULL	RXAVL	TXC
												r	r	r	r

Bit	Field	Description
31: 4	Reserved	保留，始终读为 0
3	TXEPT	发送缓冲空标识位（Transmit buffer empty flag bit） 0：发送缓冲不为空 1：发送缓冲为空
2	TXFULL	发送缓冲满标识位（Transmit buffer full flag bit） 0：发送缓冲不满 1：发送缓冲满
1	RXAVL	接收有效字节数据标识位（Receive valid data flag bit） 当接收缓冲接收了一个完整字节的数据时置位该位。 0：接收缓冲为空 1：接收缓冲接收了一个完整有效的字节数据
0	TXC	发送完成标识位（Transmit complete flag bit） 0：移位寄存器发送未完成 1：移位寄存器数据发送完成 注：此标识位在智能卡（ISO7816）时与保护时间（UART_SCR.SCFCNT 配置）相关。

25.7.5 UART_ISR 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	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.				ABRER	ABREN	RXIDLE	RXB8_I	TXBRK_	RXBRK	RXFER	RXPER	RXOER	TXC_IN	RX_INT	TX_INT	
				R_INTF	D_INTF	_INTF	NTF	INTF	_INTF	R_INTF	R_INTF	R_INTF	TF	F	F	
				r	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	ABRERR_INTF	UART 自动波特率错误中断标志位（Auto baud rate error interrupt flag bit） 0：无自动波特率错误 1：自动波特率错误

Bit	Field	Description
10	ABREND_INTF	UART 自动波特率结束中断标志位 (Auto baud rate end interrupt flag bit) 0: 自动波特率未结束 1: 自动波特率结束
9	RXIDLE_INTF	UART 接收空闲帧中断标志位 (Receive frame idle interrupt flag bit) 在停止位后 RX 引脚在一段时间内接收到若干 (UART_IDLR 数量) 高电平。 0: 没有检测到空闲帧 1: 检测到空闲帧
8	RXB8_INTF	UART 同步帧中断标志位 在 9 位通讯模式下, 当接收到数据的第九位与寄存器 CCR.B8POL 相同时, RXB8_INT 位置。该位可以作为中断请求信号 0: 没有接收到同步帧 1: 接收到同步帧
7	TXBRK_INTF	UART 断开帧发送完成中断标志位 1: 移位寄存器断开帧数据发送完成 0: 移位寄存器空或正在移位发送备注: 不能连续发送断开帧。
6	RXBRK_INTF	UART 接收断开帧中断标志位 (Receive frame break interrupt flag bit) 在异常停止位后 RX 引脚在一段时间内接收到 10 个或大于 10 位的低电平。 0: 没有检测到断开帧 1: 检测到断开帧
5	RXFERR_INTF	帧错误中断标志位 (Frame error interrupt flag bit) 帧错误发生在当检测到异常停止位。 0: 没有检测到帧错误 1: 检测到帧错误
4	RXPERR_INTF	奇偶校验错误中断标志位 (Parity error interrupt flag bit) 0: 没有检测到奇偶校验错误 1: 检测到奇偶校验错误
3	RXOERR_INTF	接收溢出错误中断标志位 (Receive overflow error interrupt flag bit) 仅当 autoflowen=0 时会置位。 0: 没有溢出错误 1: 接收溢出错误
2	TXC_INTF	UART 发送移位寄存器完成中断标志位 0: 移位寄存器非空或正在移位发送 1: 移位寄存器数据发送完成 注: 此标志位在智能卡 (ISO7816) 时与保护时间 (UART_SCR.SCFCNT 配置) 相关。
1	RX_INTF	接收有效数据中断标志位 (Receive valid data interrupt flag bit) 当接收缓冲接收了一个完整字节的数据时置位该位。 0: 接收缓冲为空 1: 接收缓冲有效字节数据
0	TX_INTF	发送缓冲空中断标志位 (Transmit buffer empty interrupt flag bit) 0: 发送缓冲不为空 1: 发送缓冲空

25.7.6 UART_IER 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ABRER R_IEN	ABREN D_IEN	RXIDLE _IEN	RXB8_I EN	TXBRK_ IEN	RXBRK _IEN	RXFERR R_IEN	RXPERR R_IEN	RXOERR R_IEN	TXC_IEN	RX_IEN	TX_IEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 12	Reserved	保留，始终读为 0
11	ABRERR_IEN	自动波特率错误中断使能（Auto baud rate error enable bit） 0：中断禁止 1：中断使能
10	ABREND_IEN	自动波特率结束中断使能（Auto baud rate end enable bit） 0：中断禁止 1：中断使能
9	RXIDLE_IEN	接收空闲帧中断使能位（Receive frame idle interrupt enable bit） 0：中断禁止 1：中断使能
8	RXB8_IEN	UART 同步帧中断使能控制位 0：禁止接收同步帧中断 1：使能接收同步帧中断
7	TXBRK_IEN	UART 断开帧发送完成中断使能控制位 0：禁止发送断开帧完成中断 1：使能发送断开帧完成中断
6	RXBRK_IEN	UART 接收断开帧中断使能位（Receive frame break interrupt enable bit） 0：中断禁止 1：中断使能
5	RXFERR_IEN	帧错误中断使能位（Frame error interrupt enable bit） 0：中断禁止 1：中断使能
4	RXPERR_IEN	奇偶校验错误中断使能位（Parity error interrupt enable bit） 0：中断禁止 1：中断使能
3	RXOERR_IEN	接收溢出错误中断使能位（Receive overflow error interrupt enable bit） 0：中断禁止 1：中断使能

Bit	Field	Description
2	TXC_IEN	UART 发送移位寄存器完成中断使能控制位 0: 中断禁止 1: 中断使能
1	RX_IEN	接收缓冲中断使能位 (Receive buffer 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 buffer empty interrupt enable bit) 0: 中断禁止 1: 中断使能

25.7.7 UART_ICR 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ABRER	ABREN	RXIDLE	RXB8_I	TXBRK	RXBRK	RXFER	RXPER	RXOER	TXC_IC	RX_ICL	TX_ICL
				R_ICLR	D_ICLR	_ICLR	CLR	_ICLR	_ICLR	R_ICLR	R_ICLR	R_ICLR	LR	R	R
				W	W	W	W	W	W	W	W	W	W	W	W

Bit	Field	Description
31: 12	Reserved	保留, 始终读为 0
11	ABRERR_ICLR	UART 自动波特率错误中断清除位 (Auto baud rate error clear bit) 0: 写 0 无意义 1: 写 1 清除中断
10	ABREND_ICLR	UART 自动波特率结束中断清除位 (Auto baud rate end clear bit) 0: 写 0 无意义 1: 写 1 清除中断
9	RXIDLE_ICLR	UART 接收空闲帧中断清除位 (Receive frame idle interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
8	RXB8_ICLR	UART 同步帧中断标志清除控制位 0: 写 0 无意义 1: 清除接收同步帧中断标志
7	TXBRK_ICLR	UART 断开帧发送完成中断标志清除控制位 0: 写 0 无意义 1: 清除断开帧发送完成中断标志

Bit	Field	Description
6	RXBRK_ICLR	UART 接收断开帧中断清除位 (Receive frame break interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
5	RXFERR_ICLR	帧错误中断使能位 (Frame error interrupt enable bit) 0: 写 0 无意义 1: 写 1 清除中断
4	RXPERR_ICLR	奇偶校验错误中断清除位 (Parity error interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
3	RXOERR_ICLR	接收溢出错误中断清除位 (Receive overflow error interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
2	TXC_ICLR	发送完成中断清除位 (Transmit complete interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
1	RX_ICLR	接收中断清除位 (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 (Transmit buffer empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断

25.7.8 UART_GCR 全局控制寄存器

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.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.					TXTOG	RXTOG	SWAP	SELB8	Res.			TXEN	RXEN	AUTOF LOWEN	DMAMO DE	UARTE N
					rw	rw	rw	rw				rw	rw	rw	rw	rw

Bit	Field	Description
31: 11	Reserved	保留, 始终读为 0
10	TXTOG	发送取反位 0: 无效 1: 发送取反

Bit	Field	Description
9	RXTOG	接收取反 0: 无效 1: 接收取反
8	SWAP	输入与输出交换 0: 无效 1: 输入输出交换 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。
7	SELB8	选择 B8 数据接收或发送是否有效 0: 无效 1: B8 数据收发 UART_CCR 位 B8EN 有效时, 发送时发送 9 位数据; 接收时接收寄存器位 CHAR+1 的数据长度。
6: 5	Reserved	保留, 始终读为 0
4	TXEN	发送使能位 (Enable transmit) 0: 发送禁止。可以清除 TX BUFFER 1: 发送使能
3	RXEN	接收使能位 (Enable receive) 0: 接收禁止。可以清除 RX BUFFER. 1: 接收使能
2	AUTOFLOWEN	自动流控制使能位 (Automatic flow control enable bit) 0: 自动流控制禁止 1: 自动流控制使能
1	DMAMODE	DMA 方式选择位 (DMA mode selection bit) 0: 选择正常方式 1: 选择 DMA 方式
0	UARTEN	UART 模块选择位 (UART mode selection bit) 0: UART 模块禁止 1: UART 模块使能

25.7.9 UART_CCR 通用控制寄存器

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LIN	WAKE	RWU	B8EN	B8TOG	B8POL	B8TXD	B8RXD	SPB1	CHAR		BRK	SPB0	PSEL	PEN

	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw
--	----	----	----	----	----	----	----	---	----	----	----	----	----	----

Bit	Field	Description
31: 15	Reserved	保留，始终读为 0
14	LIN	UART LIN 协议收发断开帧（UART LIN enable bit） 0: LIN 协议无效 1: LIN 协议有效
13	WAKE	唤醒方法。 该位决定把 UART 唤醒的方法。 0: 空闲总线唤醒 1: 地址标记唤醒
12	RWU	接收静默。 该位用来决定是否把 UART 置于静默模式。该位可以由软件设置或清除。当唤醒序列到来时，硬件也会自动将其清零。 0: 接收器处于正常工作模式 1: 接收器处于静默模式 在设置地址标记唤醒时，如果接收 buffer 非空则不能软件修改。
11	B8EN	UART 同步帧第九位使能控制位。 该位使能后校验使能位 PEN 不起作用。 0: 禁止同步帧第九位发送 1: 使能同步帧第九位发送
10	B8TOG	UART 同步帧发送第九位自动翻转控制位。 0: 禁止第九位自动翻转 1: 使能第九位自动翻转 注: 在 B8TXD 和 B8POL 的值相同时，在配置完寄存器后传输的第二个数据开始翻转，第一个数据默认为地址位。
9	B8POL	UART 同步帧第九位极性控制位。 0: 同步帧第九位低电平有效 1: 同步帧第九位高电平有效
8	B8TXD	UART 同步帧发送数据第九位。 0: 发送同步帧第九位为低电平 1: 发送同步帧第九位为高电平
7	B8RXD	UART 同步帧接收数据第九位。 只读。 0: 接收同步帧第九位为低电平 1: 接收同步帧第九位为高电平
6	SPB1	停止位选择位，与 SPB0 结合设置停止位位数。
5: 4	CHAR	UART 数据宽度位（UART width bit） 00: 5 位；01: 6 位 10: 7 位；11: 8 位
3	BRK	UART 发送断开帧（UART transmit frame break） 0: 不输出断开帧 1: 串行强制输出逻辑'0'（断开帧）

Bit	Field	Description
2	SPB0	停止位选择 (Stop bit selection) 设置发送停止位位数。 SPB1, SPB0 00: 1 个停止位 SPB1, SPB0 01: 2 个停止位 SPB1, SPB0 10: 0.5 个停止位 SPB1, SPB0 11: 1.5 个停止位
1	PSEL	校验选择位 (Parity selection bit) 当校验使能后, 该位用于选择是采用偶校验还是奇校验。 0: 奇校验 1: 偶校验
0	PEN	校验使能位 (Parity enable bit) 0: 禁止奇偶校验 1: 发送接收使能奇偶校验

25.7.10 UART_BRR 波特率寄存器

偏移地址: 0x20

复位值: 0x0000 0001

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	DIV_Mantissa	UARTDIV 的整数部分 这 16 位定义了 UART 分频器除法因子 (UARTDIV) 的整数部分。 DIV_Mantissa 最小值为 1

25.7.11 UART_FRA 分数波特率寄存器

偏移地址: 0x24

复位值: 0x0000 0000

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	DIV_Mantissa	UARTDIV 的整数部分 这 16 位定义了 UART 分频器除法因子 (UARTDIV) 的整数部分。 DIV_Mantissa 最小值为 1

Res.	DIV_Fraction
	rw

Bit	Field	Description
31: 4	Reserved	保留，始终读为 0
3: 0	DIV_Fraction	UARTDIV 的小数部分 这 4 位定义了 UART 分频器除法因子 (UARTDIV) 的小数部分。

25.7.12 UART_RXADDR 接收地址寄存器

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

Bit	Field	Description
31: 8	Reserved	保留，始终读为 0
7: 0	RXADDR	UART 同步帧数据本机匹配地址。 如果 RXMASK =0xFF 时，接收到的同步帧数据与本机匹配地址相同时，产生 RXB8_INTF。 地址 0 是广播地址，收到后都会响应。

25.7.13 UART_RXMASK 接收掩码寄存器

偏移地址: 0x2C

复位值: 0x0000 00FF

31	30	29	28	27	26	25	24	23	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.								RXMASK							
rw															

Bit	Field	Description
31: 8	Reserved	保留，始终读为 0

Bit	Field	Description
7: 0	RXMASK	数据位全为“0”时，接收到任何数据都产生同步帧中断请求。 如果数据位为“1”，RDR 和 RXADDR 的相应位匹配时，产生同步帧中断请求。

25.7.14 UART_SCR SCR 寄存器

偏移地址：0x30

复位值：0x0000 0000

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

Bit	Field	Description
31: 13	Reserved	保留，始终读为 0
12	HDSEL	单线半双工模式选择。 0: 禁止半双工模式 1: 使能半双工模式
11: 4	SCFCNT	ISO7816 保护计数器。 发送数据时，通过该寄存器配置当前数据发送完成时延迟 TXC 标志置位的时间。在未达到此计数周期时，发送器将禁止发送下一个数据的起始位。 1: 禁止此设定 2~255: 保护时间为 2~255 个波特率时间（单位为波特率时间） 注：当设置计数器为 0x0 或 0xFF 时，均为 255 个波特率的计数时间。
3	Reserved	保留，始终读为 0
2	NACK	主接收帧应答位。 保护期间收到低电平置位
1	SCAEN	ISO7816 校验自动应答位。 0: 禁止自动应答 1: 使能自动应答
0	SCEN	ISO7816 使能控制位。 0: 禁止 ISO7816 功能 1: 使能 ISO7816 功能

25.7.15 UART_IDLR IDLE 数据长度寄存器

偏移地址：0x34

复位值：0x0000 000C

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

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

Bit	Field	Description
31: 16	Reserved	保留，始终读为 0
15: 0	IDLR	UART idle 数据长度寄存器 (Idle data length register) 数据长度不可配置为 0。

25.7.16 UART_ABRCCR 自动波特率寄存器

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.											Latter_e	Former_	Abr_bitcnt		Abren
											dge	edge			
											rw	rw	rw		rw

Bit	Field	Description
31: 5	Reserved	保留，始终读为 0
4	Latter_edge	自动波特率后一个边沿选择。 0: 下降沿 1: 上升沿
3	Former_edge	自动波特率前一个边沿选择。 0: 下降沿 1: 上升沿
2: 1	Abr_bitcnt	自动波特率检测长度。 检测前一个边沿和后一个边沿之间的位长。 00: 1 位 01: 2 位 10: 4 位 11: 8 位
0	Abren	自动波特率使能。 只能在 UART 空闲时使能自动波特率，使能后检测接收信号的边沿，完成自动波特率检测后，硬件自动设置 UART_BRR 和 UART_FRA 寄存器。 0: 自动波特率禁止 1: 自动波特率使能

25.7.17 UART_IRDA 红外功能控制寄存器

偏移地址：0x3C

复位值：0x0000 0100

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

Bit	Field	Description
31: 16	Reserved	保留，始终读为 0
15: 8	PSC_REG	预分频寄存器（Prescaler value） 在红外低功耗模式下，对 UART 源时钟（pclk）分频获得低功耗模式下频率 sirIp_clk： 0000 0000：保留，不能写入该值 0000 0001：对源时钟 1 分频 0000 0010：对源时钟 2 分频 0000 0011：对源时钟 3 分频 注：保证分频后的时钟频率在（1.42 MHz < sirIp_clk < 2.12 MHz）之间。
7: 2	Reserved	保留，始终读为 0
1	SirIp	红外低功耗模式（IrDA low_power） 0：普通模式 1：低功耗模式
0	Siren	IrDA 红外模式使能（IrDA mode enable） 0：不使能红外模式 1：使能红外模式

26 LPUART 低功耗通用异步收发器

26.1 简介

LPUART 为低功耗通用异步收发器, 相比 UART, 其功耗极低, 支持在低功耗模式运行以及唤醒芯片, 支持唤醒的低功耗模式参考 PMU 模块。

26.2 功能框图

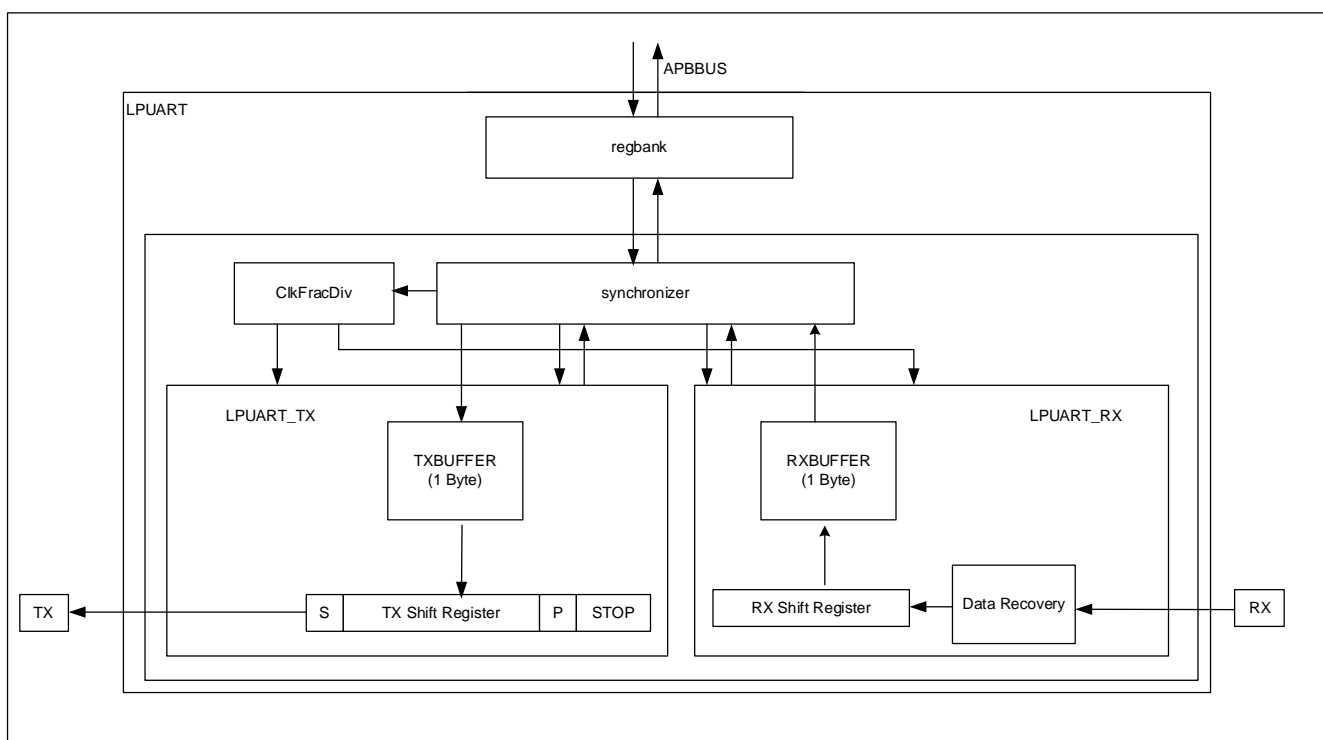


图 26-1 LPUART 功能框图

26.3 主要特征

- 支持 UART 帧格式的异步数据收发
- 全双工异步操作
- 支持输入任意频率时钟源, 配置参考 RCC 模块 RCC_CFGR2 寄存器描述。
- 支持可编程的波特率数据传输, 计算方法参考本章节表 3。
- 内置独立的 1 字节发送和 1 字节接收缓冲
- 发送和接收数据低位在前
- 一个起始位开始, 后面接数据位, 输出的数据长度可为 7 或 8 位, 最后为停止位。另外可选择是

否有加奇偶校验位，奇偶校验位在数据位之后停止位之前。

- 支持硬件奇数或者偶数校验产生和侦测
- 支持信号接收和发送取反
- 支持下面中断源：
 - 接收缓冲溢出
 - 帧错误
 - 奇偶校验错误
 - 接收器检测到起始位
 - 接收器检测到下降沿
 - 接收器完整接收 1byte 数据
 - 接收器完整接收数据且与预设数据匹配
 - 发送器数据完成发送
 - 发送器缓冲空
 - 支持低功耗模式以下唤醒源：
 - 接收器检测到下降沿唤醒
 - 接收器检测到起始位唤醒
 - 接收器 1 字节接收完成唤醒
 - 接收器 1 字节数据接收并匹配唤醒

26.4 中断

表 26-1 LPUART 中断请求

中断事件	中断状态	使能
RX 检测到 START 位中断	LPUSTA_START	LPUCON_RXIE & LPUCON_RXEV=00
RX 接收完成 1byte 数据中断	LPUIF_RXIF	LPUCON_RXIE & LPUCON_RXEV=01
RX 接收数据匹配成功中断	LPUSTA_MATCH	LPUCON_RXIE & LPUCON_RXEV=10
RX 检测到下降沿中断	LPUIF_RXNEGIF	LPUCON_RXIE & LPUCON_RXEV=11
发送 buffer 空	LPUIF_TXIF	LPUCON_TXIE
发送一帧数据完成	LPUIF_TC_IF	LPUCON_TCIE
检验位错误中断	LPUSTA_PERR	LPUCON_ERRIE
帧格式错误中断	LPUSTA_FERR	LPUCON_ERRIE
接收 buffer 溢出中断	LPUSTA_RXOV	LPUCON_ERRIE

如果设置了对应的中断使能控制位，这些设置就可以产生各自对应的中断。

26.5 DMA

LPUART 可以利用 DMA 来搬运数据；使能 LPUEN 的 DMAR 与 DMAT 位激活 DMA 模式；

利用 DMA 发送：DMA 搬送数据到 LPUTXD

使能 TXEN、DMAT 后，只要 TXFIFO 为空，就请求 DMA 发送。

在利用 DMA 发送时，要配置好 DMA 的源地址（存储器地址）和目的地址（LPUTXD），传输的数据量以及 DMA 通道。

利用 DMA 接收：DMA 将 LPURXD 数据搬走

使能 RXEN、DMAR 后，只要 RXFIFO 有数据，即不为空，就请求 DMA 接收。

在利用 DMA 接收时，要配置好 DMA 的源地址（LPURXD）和目的地址（存储器地址），传输的数据量以及 DMA 通道。

26.6 功能描述

LPUART 为全双工通信，需要分配两个脚给 LPUART：接收数据输入（RX）和发送数据输出（TX）。

LPUART 数据帧符合 UART 协议，空闲帧与断开帧未定义。

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。

一个数据帧（7 或 8 位），发送和接收顺序从最低位到最高位。

停止位用 1 表示一帧的结束，位数可配置为 1、2 个。

可配置有无奇偶检验位，可配置奇校验或者偶校验。

通过使能 LPUCON 寄存器的 RXPOL/ TXPOL 位，可以将接收/发送端的信号取反。

26.6.1 特性描述

数据长度可以编程为 7 或 8 位，需要配置 LPUCON 寄存器中的 DL 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。

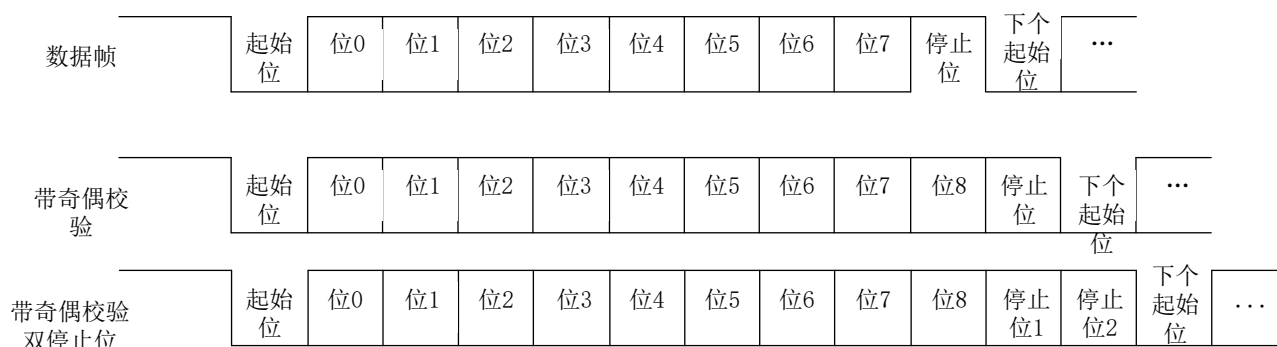


图 26-2 帧时序

26.6.2 接收时序

由于 LPUART 工作时钟不是波特率的整数倍，采用固定分频系数的话会引入累积误差，在接收时采用 3、4 分频交替进行接收，确保在每个 bit 的中间位置采样，每个 bit 采样一次。每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：

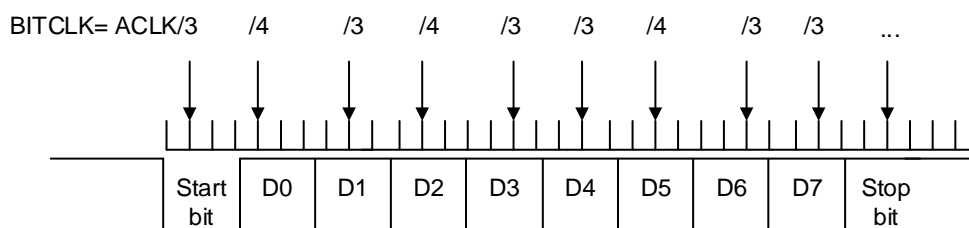


图 26-3 接收时序

26.6.3 发送时序

类似于 LPUART 接收，LPUART 工作时钟不是波特率的整数倍，采用固定分频系数同样会引入累积误差，在发送时也采用了 3、4 分频交替进行发送，每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：

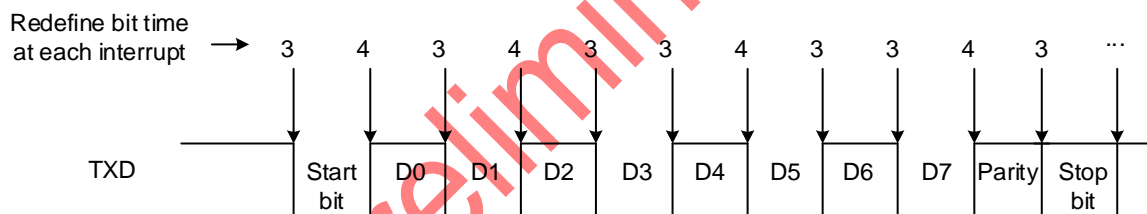


图 26-4 发送时序

26.6.4 调制寄存器建议配置

当 LPUART 工作时钟为准确的 32768Hz 时，软件可配置 BREN 为 0，然后需要根据通信波特率的不同合理配置调制控制寄存器 MCTL，建议的配置参数表如下：

表 26-2 调制控制寄存器 MCTL 建议配置参数表

Baud	MCTL											
	Bit0 (start)	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

当 LPUART 工作时钟不是 32768Hz 时，软件可配置 BREN 为 1，并根据工作时钟计算 BR 和 MODU 的值，以时钟源 40KHz 和波特率 9600 为例计算方式如下：

表 26-3 调制控制寄存器 MCTL 计算方式

i	通信帧 bit	累计周期 S	单 bit 采样周期 N	BR	MCTL
1	START bit0	4	4	3	0
2	bit1	8	4		0
3	bit2	13	5		1
4	bit3	17	4		0
5	bit4	21	4		0
6	bit5	25	4		0
7	bit6	29	4		0
8	bit7	33	4		0
9	bit8	38	5		1
10	bit9	42	4		0
11	bit10	46	4		0
12	bit11	50	4		0

$$S = \text{ROUND} (F_{\text{uclk}}/F_{\text{bps}}*i,0);$$

$$N = S_i - S_{i-1};$$

$$BR = \text{MIN} (N) - 1;$$

$$\text{MCTL} = N - BR - 1;$$

26.6.5 奇偶校验控制

奇偶控制(发送时生成一个奇偶位,接收时进行奇偶校验)可以通过设置 LPUCON 寄存器上的 PAREN 位而激活。如果奇偶校验出错,无效数据仍然会从移位寄存器传送到 LPURXD 寄存器, LPUSTA 的 PERR 位硬件自动置 1。LPUSTA 的 PERR 位软件写“1”可清零该状态标志位。

偶校验: 校验位加上数据总共的 1 的个数为偶数。

奇校验: 校验位加上数据总共的 1 的个数为奇数。

待发送数据的 1 的个数是不可改变的, 通过配置校验位为 1 或者 0, 满足奇偶校验的要求。

例如: 数据= 1000_0110, 有 3 个‘1’

偶校验: (LPUCON 中的 PTYP = 0), 校验位将是‘1’。

奇校验: (LPUCON 中的 PTYP = 1), 校验位将是‘0’。

在带奇偶校验的传输中, 奇偶校验位的发送将跟在数据发送完最高有效位的后面。

26.6.6 发送器

发送 buffer 深度位 1 个字节。

DL 位的配置决定发送器发送数据的位数 (7~8 位)。发送数据需要先置起发送使能位 (TXEN), 之后数据随着发送移位寄存器串行输出到 TX 引脚上。

26.6.6.1 字符发送

在 LPUART 发送期间，数据从 LPUTXD 寄存器写入，经过一字节缓冲器缓冲，最后通过发送移位寄存器，以最低字节到最高字节的顺序，串行在 TX 引脚上输出。

一位低电平的起始位会先于字符发送，而在字符发送完成的最后会发送高电平的停止位。停止位的位数可通过寄存器配置。

当前数据传输未完成前不能清零 TXEN 位，否则此时波特率发生器会停止产生时钟，发送器由于没有时钟驱动导致发送数据后部分丢失，数据不完整。

26.6.6.2 可配置的停止位

通过 SL 位进行编程，可以控制发送器每帧输出停止位的位数 1 或者 2。

26.6.6.3 配置步骤

- 配置 LPUBAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUCON 寄存器，选择帧格式、极性、中断参数等
- 配置 LPUEN 寄存器打开发送使能
- 将待发送数据写入 LPUTXD 寄存器
- 等待中断事件
- 当发送完成时，LPUSTA 的 TXE 标志位会被硬件置起，表示数据已传入移位寄存器，发送 buffer 为空。此时可往 LPUTXD 写入下一个数据。软件向发送 buffer 写数据时 TXE 标志位自动清零。

26.6.7 接收器

接收 buffer 深度位 1 个字节。

26.6.7.1 字符接收

对于 LPUART 的接收器，RX 串行输入数据，接收顺序从数据最低有效位到最高有效位。在接收器中，内部总线和接收移位寄存器之间插入了 LPURXD 寄存器包含的缓冲器。

26.6.7.2 配置步骤

- 配置 LPUBAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器

- 配置 LPUCON 寄存器，选择帧格式、极性、中断参数等
- 配置 LPUEN 寄存器打开接收使能和中断使能
- 等待相应中断事件

当接收一个完整帧时，LPUSTA 的 RXF 标志位置起，表示已完整接收数据，此时软件可读取 LPURXD 读出接收到的数据。软件读 LPUDATA 寄存器时，RXF 标志位自动清零。为了防止溢出错误，在下一字符接收结束前，RXF 位必须被清零。

在接收数据时，不应该软件清零 RXEN 位。如果在接收时清零 RXEN 位，还在 RX 引脚上的数据将不再被接收。

26.6.7.3 溢出错误

如果在 LPURXD 前一个字符没有被读出前又接收到一个字符，则发生溢出错误。当溢出错误产生时：

- LPUSTA 的 RXOV 位被置位。
- 新的数据将会刷新 LPURXD 寄存器，上一个未读数据将会丢失。
- 若想产生中断，需配 LPUCON 的 ERRIE 置位。

26.6.7.4 帧错误

当停止位没有在预期的时间上接收和识别出来时检测到帧错误。当帧错误被检测到时：

- LPUSTA 的 FERR 位被硬件置起。
- 新的数据将会刷新 LPURXD 寄存器。
- 若想产生中断，需配 LPUCON 的 ERRIE 置位。

26.6.8 休眠模式下的数据接收唤醒

LPUART 支持在休眠模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

- 配置 LPUBAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUCON 寄存器，选择帧格式、极性，通过 LPUCON.RXEV 选择唤醒事件为 START 位、一帧接收完成、一帧数据匹配或 RXD 下降沿检测
- 配置 LPUEN 寄存器打开接收使能
- 软件进入休眠模式

26.6.9 低功耗模式下的数据自动收发

通过 DMA 功能，软件可以实现低功耗模式下一定数据量的 LPUART 自动收发，而无需 CPU 干预，

低功耗模式参考 PMU 模块。

- 配置 LPUBAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUCON 寄存器，选择帧格式、极性、中断参数等
- 配置 DMA 通道 TX 和通道 RX 对应控制寄存器，选择 LPUART 收发
- 如果需要发送数据，将待发数据写入 RAM 中指定位置
- 配置 DMA 数据收发长度和 RAM 指针
- 将系统主时钟选为 LSCLK
- 软件进入 LPRUN
- 配置 LPUEN 寄存器打开发送接收使能
- 如 CPU 无额外工作，可以主动进入 WFI/WFE，等待中断唤醒

26.7 寄存器描述

26.7.1 寄存器概览

表 26-4 LPUART 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	LPUBAUD	波特率寄存器	0x00000000
0x04	MODU	波特率调制控制寄存器	0x00000000
0x08	LPUIF	中断标志寄存器	0x00000002
0x0C	LPUSTA	状态寄存器	0x00000030
0x10	LPUCON	控制寄存器	0x00000000
0x14	LPUEN	发送接收使能寄存器	0x00000000
0x18	LPURXD	接收数据寄存器	0x00000000
0x1C	LPUTXD	发送数据寄存器	0x00000000
0x20	COMPARE	数据匹配寄存器	0x00000000
0x24	WKCKE	唤醒寄存器	0x00000000

26.7.2 LPUBAUD 波特率寄存器

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
BR															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							BREN		Res.				BAUD		

Bit	Field	Description
31: 16	BR	波特率分频 时钟源到波特率的分频系数 BR+1 BR 不可配置为 0 和 1。
15: 9	Reserved	保留，始终读为 0
8	BREN	波特率分频使能 0: 波特率有 BAUD 寄存器控制，此时只支持 32.768KHz 的时钟作为时钟源 1: 波特率有 BR 寄存器控制，可以支持 32.768KHz 以及其他频率的时钟作为时钟源
7: 3	Reserved	保留，始终读为 0
2: 0	BAUD	波特率控制，仅在 BREN=0 时有效 000b : 9600bps 001b : 4800bps 010b : 2400bps 011b : 1200bps 100b : 600bps 101b / 110b / 111b : 300bps 注意：当 BREN = 1 时，需要配置 BAUD ! = 0

26.7.3 MODU 波特率调制控制寄存器

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.				MCTL											
Res.				rw											

Bit	Field	Description
31: 12	Reserved	保留，始终读为 0
11: 0	MCTL	LPUART 每个 bit 的调制控制信号

26.7.4 LPUIF 中断标志寄存器

偏移地址：0x08

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	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.												TC_IF	RXNEGIF	TXIF	RXIF
												r/w1c	r/w1c	r/w1c	r/w1c

Bit	Field	Description
31: 4	Reserved	保留，始终读为 0
3	TC_IF	发送完成中断标志 1: 一帧数据发送完后中断产生 0: 无中断产生 该位写 1 清零
2	RXNEGIF	RXD 下降沿中断标志 1: RXD 下降沿中断产生 0: 无中断产生 该位写 1 清零
1	TXIF	发送 buffer 空中断标志 1: 发送 buffer 空后中断产生 0: 无中断产生 该位写 1 清零
0	RXIF	接收完成中断标志 1: 接收完一帧数据后中断产生 0: 无中断产生 该位写 1 清零 注意：需要与 WKCKE 寄存器配合使用

26.7.5 LPUSTA 状态寄存器

偏移地址：0x0C

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	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	PERR	TC	TXE	RXF	MATCH	FERR	RXOV
	r/w1c	r/w1c	r	r	r	r/w1c	r/w1c	r/w1c

Bit	Field	Description
31: 8	Reserved	保留，始终读为 0
7	START	起始位检测标志 1: 检测到起始位 0: 未检测到起始位 该位写 1 清零
6	PERR	校验位错误 1: 检测到校验位错误 0: 未检测到校验位错误 该位写 1 清零
5	TC	发送空闲状态标志 1: 当发送过程处于等待状态下且发送 buffer 为空 0: 数据发送过程正在进行或发送 buffer 非空 该位在发送 buffer 非空或数据发送开始时硬件自动清零
4	TXE	发送 buffer 空标志 1: 发送 buffer 为空 0: 发送 buffer 非空 该位硬件置位，在软件向 LPUTXD 写数据时自动清零
3	RXF	接收缓冲满 1: 接收 buffer 为满 0: 接收 buffer 非满 该位在软件读 LPURXD 寄存器时自动清零
2	MATCH	数据匹配标志 1: 接收缓冲区内的数据与数据匹配寄存器相同 0: 接收缓冲区内的数据与数据匹配寄存器不同 该位写 1 清零
1	FERR	帧格式错误 1: 检测到帧格式错误 0: 未检测到帧格式错误 该位写 1 清零
0	RXOV	接收缓冲溢出 1: 接收缓冲发生溢出错误 0: 接收缓冲未发生溢出错误 该位写 1 清零

26.7.6 LPUCON 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.			TXPOL	RXPOL	PAREN	PTYP	SL	DL	RXEV		ERRIE	TCIE	TXIE	NEDET	RXIE
			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
31: 13	Reserved	保留, 始终读为 0
12	TXPOL	数据发送极性 0: 非取反 1: 取反
11	RXPOL	接收极性 0: 非取反 1: 取反
10	PAREN	校验位使能 0: 数据帧无奇偶校验位 1: 数据帧有奇偶校验位
9	PTYP	校验位类型 0: 偶校验 1: 奇校验
8	SL	停止位长度 0: 1bit 1: 2bits
7	DL	数据长度 0: 8bits 1: 7bits
6: 5	RXEV	接收中断事件配置 用于控制何种事件下向 CPU 提供接收中断 00: START 位检测唤醒 01: 1byte 数据接收完成 10: 接收数据匹配成功 11: 下降沿检测唤醒
4	ERRIE	错误中断使能 0: 禁止接收错误中断 1: 允许接收错误中断

Bit	Field	Description
3	TCIE	发送完成中断使能 0: 禁止发送完成中断 1: 允许发送完成中断
2	TXIE	发送 buffer 空中断使能 0: 禁止发送 buffer 空中断 1: 允许发送 buffer 空中断
1	NEDET	下降沿采样使能位 0: 不使用 32k 时钟下降沿检测 start bit 1: 使用 32k 时钟下降沿检测 start bit
0	RXIE	接收中断使能 0: 禁止接收中断 1: 允许接收中断

26.7.7 LPUEN 发送接收使能寄存器

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												DMAR	DMAT	RXEN	TXEN
												rw	rw	rw	rw

Bit	Field	Description
31: 4	Reserved	保留, 始终读为 0
3	DMAR	DMA 接收使能 0: 关闭 DMA 接收 1: 打开 DMA 接收 软件写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作 (因为有同步操作)
2	DMAT	DMA 发送使能 0: 关闭 DMA 发送 1: 打开 DMA 发送 软件写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作 (因为有同步操作)

Bit	Field	Description
1	RXEN	接收使能 0: 关闭 LPUART 接收 1: 打开 LPUART 接收 软件写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作 (因为有同步操作)
0	TXEN	发送使能 0: 关闭 LPUART 发送 1: 打开 LPUART 发送 软件写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作 (因为有同步操作)

26.7.8 LPURXD 接收数据寄存器

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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DATA							
								r							
Bit	Field	Description													
31: 8	Reserved	保留, 始终读为 0													
7: 0	DATA	接收数据寄存器 该寄存器只读													

26.7.9 LPUTXD 发送数据寄存器

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

Bit	Field	Description
31: 8	Reserved	保留, 始终读为 0
7: 0	DATA	发送数据寄存器 该寄存器只写, 读返回 0

26.7.10 COMPARE 数据匹配寄存器

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

Bit	Field	Description
31: 8	Reserved	保留, 始终读为 0
7: 0	COMPARE	比较数据 如果配置 LPUCON 的 RXEV=10, 当接收缓冲区内的数据与 COMPARE 相同时, 触发接收完成中断

26.7.11 WKCKE 唤醒寄存器

偏移地址: 0x24

复位值: 0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留, 始终读为 0

Bit	Field	Description
0	WKCKE	<p>1 字节唤醒休眠模式控制位</p> <p>当 1 字节接收完成或 1 字节数据匹配时, 可将芯片从休眠模式唤醒;</p> <p>1: 如果接收到的 1 字节数据有帧错误或奇偶校验错误时, 不能唤醒, 且对应标志位 RXIF 与 MATCH 不置起</p> <p>0: 不管接收到的 1 字节数据是否存在错误, 都可唤醒</p>

Preliminary

27 SPI_I2S 串行外设接口

27.1 SPI_I2S 功能框图

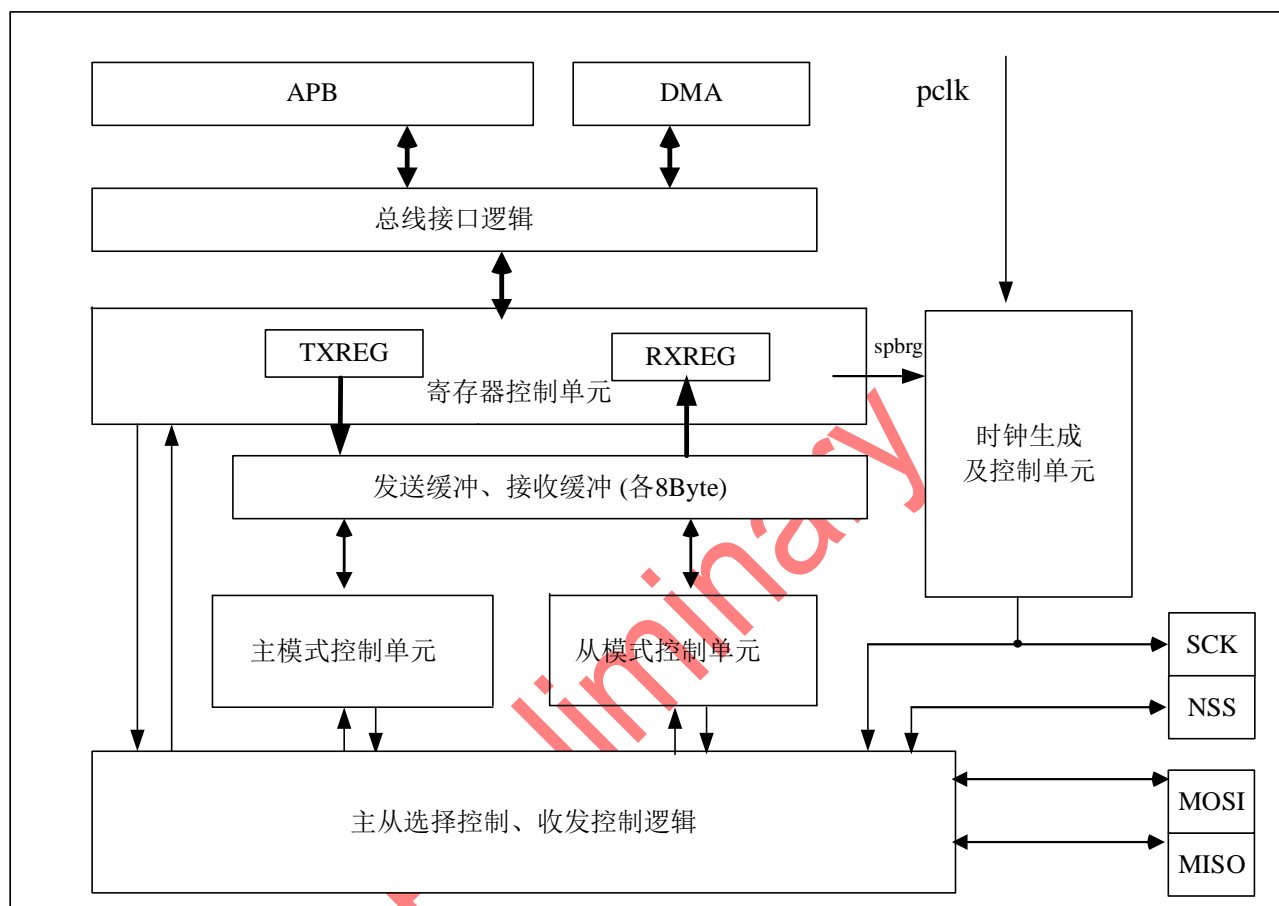


图 27-1 SPI_I2S 功能框图

27.2 SPI_I2S 简述

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。

I2S (Inter-IC Sound) 总线，也称为集成电路内置音频总线，是飞利浦公司针对数字音频设备之间的音频数据传输而制定的一种总线标准。它使用独立的导线分别传输时钟与数据信号，因而避免了因时差诱发的失真，广泛应用于多媒体系统上各音频设备之间的数据传输。

27.3 SPI 功能描述

27.3.1 概述

SPI 支持同时接收和发送 1~32 位数据，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_I2S_GCTL.MODE 为‘1’使 SPI 工作于主模式，再配置 SPI_I2S_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_I2S_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：

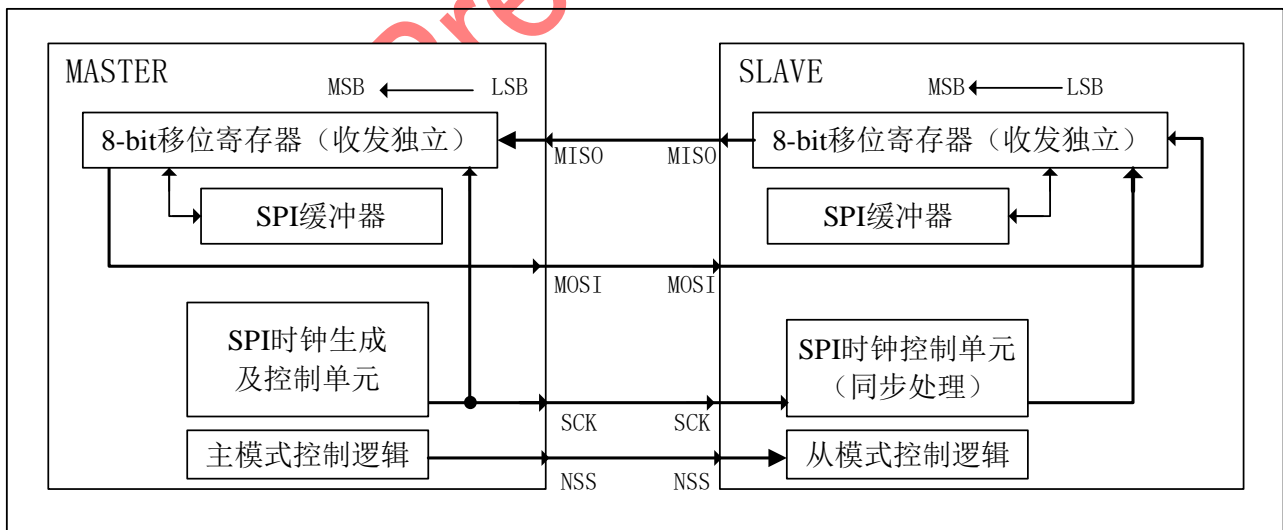


图 27-2 单主和单从应用

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。

27.3.1.1 时钟信号的极性和相位

SPI_I2S_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。

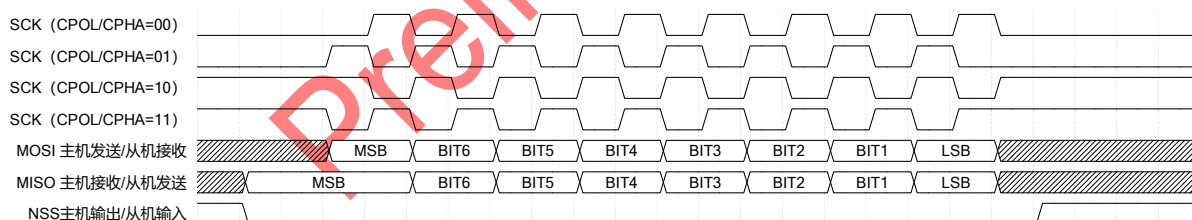


图 27-3 数据时钟时序图

需注意，配置 SPI_I2S_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_I2S_GCTL.NSS=0，NSS 输出状态则由从机片选寄存器 SPI_I2S_NSSR 的 NSS 位进行控制（需软件切换 NSS 引脚的输出状态）。

27.3.1.2 高速传输

针对高速传输时板级延时敏感的情况，可配置 SPI_I2S_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（>=10MHz）；低速传输，针对主设备波特率发生器配置在 SPBRG>4 且输出 SCK 时钟较慢（<10MHz）。

- 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。

27.3.1.3 数据帧格式

配置 SPI_I2S_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_I2S_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_I2S_GCT.DW8_32=1，和寄存器 SPI_I2S_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。

27.3.2 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 PCLK/2、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、固定帧长的数据收发都支持）
- 8 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

27.3.3 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_I2S_SPBRG（从设备下无效）。

27.3.3.1 配置步骤

1. 配置 SPI_I2S_GCTL.SPILEN，定义数据帧格式为 7 位或者 8 位；配置 SPI_I2S_GCTL.DW8_32 为‘1’，可配置 SPI_I2S_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_I2S_CCTL 中的 CPOL、CPHA/CPHASEL 位，以确定时序模式。
 3. 配置 SPI_I2S_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_I2S_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_I2S_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_I2S_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。

27.3.3.2 数据发送过程

写数据到发送数据寄存器 SPI_I2S_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_I2S_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_I2S_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_I2S_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。

27.3.3.3 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；硬件也会同时置位 SPI_I2S_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_I2S_RXREG，就能从接收缓冲中获取该数据。

- 软件配置 SPI_I2S_INTEN.RX_IEN 位为‘1’来打开中断使能，利用 CPU 中断获取接收数据。

27.3.4 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。

27.3.4.1 配置步骤

1. 配置 SPI_I2S_SPBREG 寄存器，定义串行时钟波特率。
2. 配置 SPI_I2S_CCTL 寄存器中 CPOL、CPHA/CPHASEL 位，确定时序模式。
3. 配置 SPI_I2S_CCTL.SPILEN 来定义 8 或 7 位数据帧格式；配置 SPI_I2S_GCTL.DW8_32 为‘1’，可配置 SPI_I2S_EXTCTL 寄存器来定义成任意帧格式（SPILEN 需固定为‘1’）。
4. 配置 SPI_I2S_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_I2S_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_I2S_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_I2S_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_I2S_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。

27.3.4.2 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_I2S_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_I2S_INTEN.TX_IEN 位为‘1’来产生 CPU 中断）。

27.3.4.3 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；硬件也同时会置位 SPI_I2S_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_I2S_RXREG，就能从接收缓冲中获取该数据。
- 软件配置 SPI_I2S_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_I2S_INTEN.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）。

27.3.5 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_I2S_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 27-1 波特率公式

模式	公式
SPI 模式	波特率 = Fpclk/X

27.3.6 中断

27.3.6.1 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_I2S_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_I2S_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 27-2 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

分类	状态标志	缓冲和信号状态
	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
	TXEPT	发送缓冲和发送移位寄存器都为空
	RXAVL	接收缓冲非空

27.3.7 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_I2S_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，即发送缓冲未滿时就发起 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 27.3.6.1 状态标志）。

27.4 I2S 功能描述

27.4.1 I2S 主要特征

- 半双工通信（仅发送器或接收器）
- 主操作或从操作
- 9 位可配置线性预分频器，以达到精确的音频采样频率（8KHz~192KHz）
- 数据帧格式可配置为 16 位、24 位或 32 位
- 数据包帧固定为 16 位（16 位有效数据）或 32 位（16 位、24 位、32 位有效数据）
- 可配置时钟极性（稳定状态）
- 发送模式下具有下溢标志（仅从机），接收模式下具有上溢标志（主/从机）和发送/接收模式下的帧错误标志（仅从机）
- 用于传输和接收的 32 位寄存器为两个声道分时复用
- 数据方向始终是 MSB 优先

- 支持 I2S 协议：
 - ◆ 飞利浦标准
 - ◆ MSB 对齐标准（MSB 位向左对齐）
 - ◆ LSB 对齐标准（LSB 位向右对齐）
 - ◆ PCM 标准（具有短帧同步模式、长帧同步模式的两种方式）
- 利用 DMA 请求传输数据（32 位宽）
- 可配置 MCLK 时钟输出来驱动外部音频组件，其比率固定在 $256 \times F_s$ （其中 F_s 为音频采样频率）

27.4.2 I2S 总线接口

I2S 与 SPI 共用以下引脚：

- SD: 串行数据（映射在 MOSI 引脚上），用于发送或接收两次多路数据通道（仅在半双工模式下）。
- WS: 声道选择（映射在 NSS 引脚上），是主模式控制数据的输出信号，或从模式的输入。
- CK: 串行时钟（映射在 SCK 引脚上），是主模式串行时钟的输出，或从模式串行时钟的输入。
- MCK: 可选的驱动时钟（映射在 MISO 引脚上），用于驱动外部音频组件（仅当外部音频设备需要时钟输入时使用，由主模式提供）。

27.4.3 数据格式

三线总线处理音频数据，必须经过分时复用两个声道：右声道和左声道。因为只有一个 32 位寄存器用于传输或接收，所以软件应依次配置寄存器 TXREG 为各声道的数据，或依次读取寄存器 RXREG 为各声道的数据。按照 I2S 协议，总是先发送左声道，然后发送右声道。

注意：在 PCM 协议下 SPI_I2S_CSTAT.CHSIDE 状态标志无效。

数据采用以下格式发送：

- 16 位数据打包在 16 位数据包帧
- 16 位有效数据打包在 32 位数据包帧（软件配置无效位为'0'）
- 24 位有效数据打包在 32 位数据包帧（软件配置无效位为'0'）
- 32 位数据打包在 32 位数据包帧

当使用 32 位帧上发送 16 位数据时，前 16 位（MSB）是有效的位，16 位 LSB 强制为 0，无需任何软件操作，通过硬件实现。其他格式相似。

27.4.4 通信标准

I2S 接口支持四种音频标准，通过配置寄存器 SPI_I2S_I2SCFGR 中的 I2SSTD[1: 0]、PCMSYNC 位进行切换；数据格式则通过配置 DATLEN[1: 0]、CHLEN 来进行选择。

对于所有通信标准及数据格式，总是先发送最高位（MSB 优先）。

27.4.4.1 飞利浦标准

对于飞利浦标准，WS 信号用于指示正在传输的声道，它在 CK 的下降沿被锁定。发射器在 CK 的下降沿锁存数据，接收器在 CK 的上升沿读取数据。对于本标准格式的信号，无论有多少位有效数据，每个数据包帧的最高有效位总是出现在 WS 变化后的第 2 个 CK 脉冲周期处。

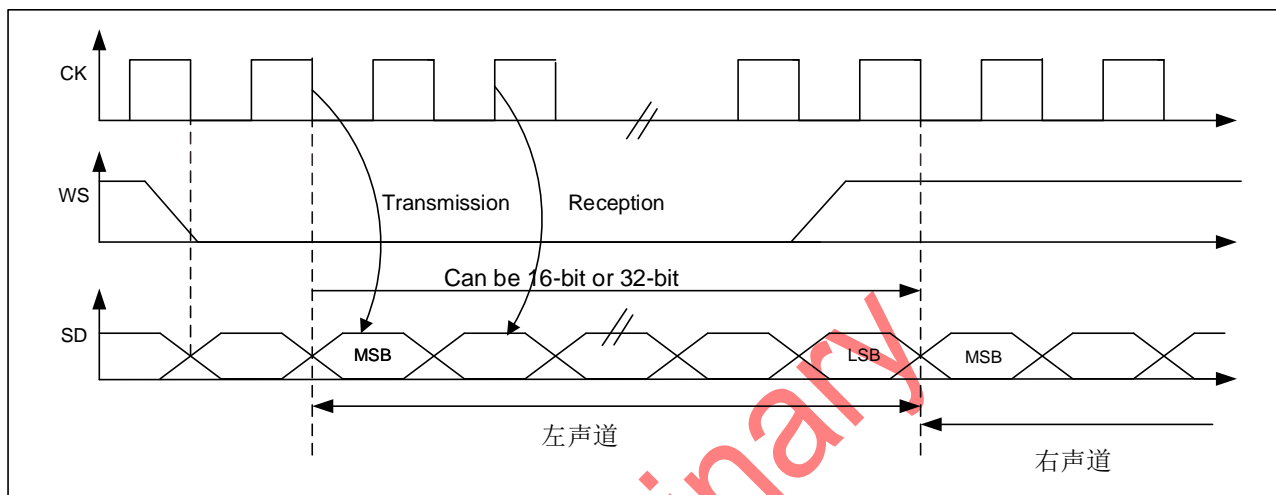


图 27-4 飞利浦标准示意图

27.4.4.2 MSB 对齐标准

对于 MSB 对齐标准，每个数据包帧的最高有效位（MSB 位）总是出现在 WS 变化后的第 1 个 CK 脉冲周期处。

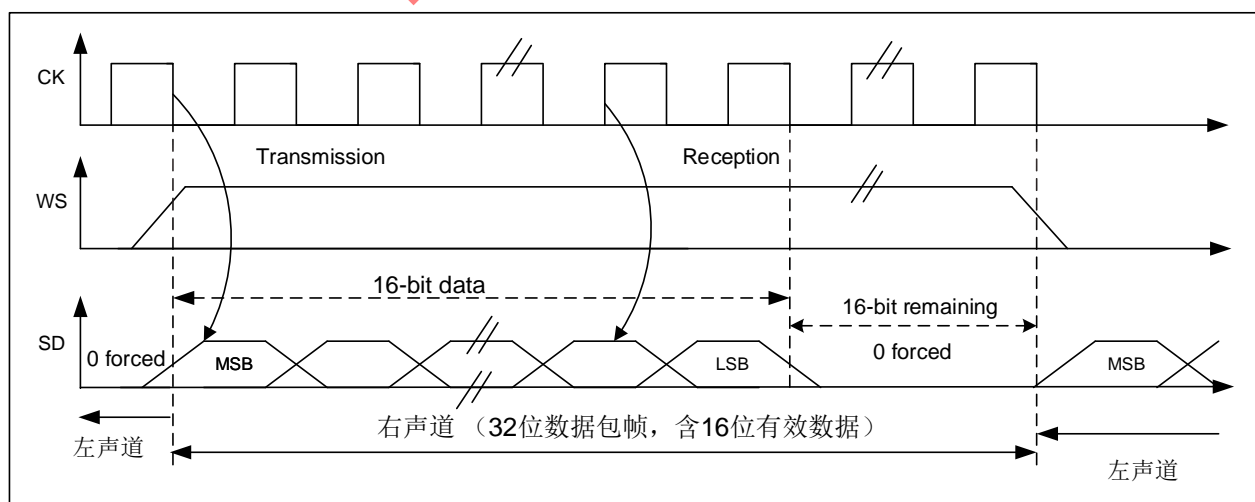


图 27-5 MSB 对齐标准示意图

27.4.4.3 LSB 对齐标准

对于 LSB 对齐标准，每个数据包帧的最低有效位总（LSB 位）是出现在 WS 变化前的 1 个 CK 脉冲周期处。

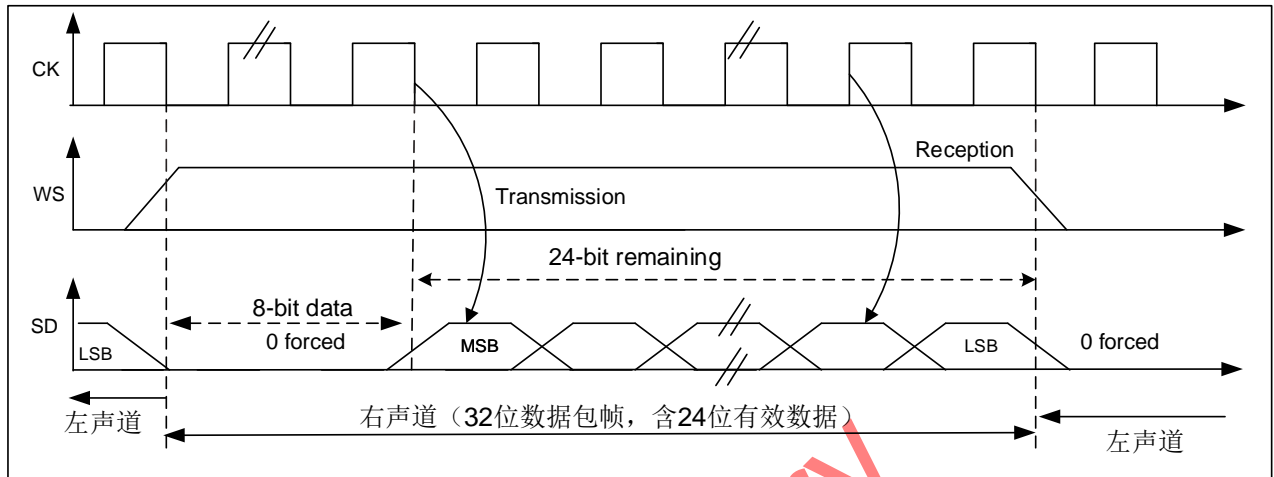


图 27-6 LSB 对齐标准示意图

27.4.4.4 PCM 标准

对于 PCM 标准，WS 引脚不作为声道信息使用（CHSIDE 标志位无效）。

PCM 标准有两个模式，分别为短帧模式和长帧模式；通过配置 SPI_I2S_I2SCFGR.PCMSYNC 位进行切换。在 PCM 模式下，输出信号（WS，SD）在 CK 时钟的上升沿锁存，输入信号（WS，SD）在 CK 时钟的下降沿采样。在主模式下应配置 CK 时钟和 WS 引脚为输出。

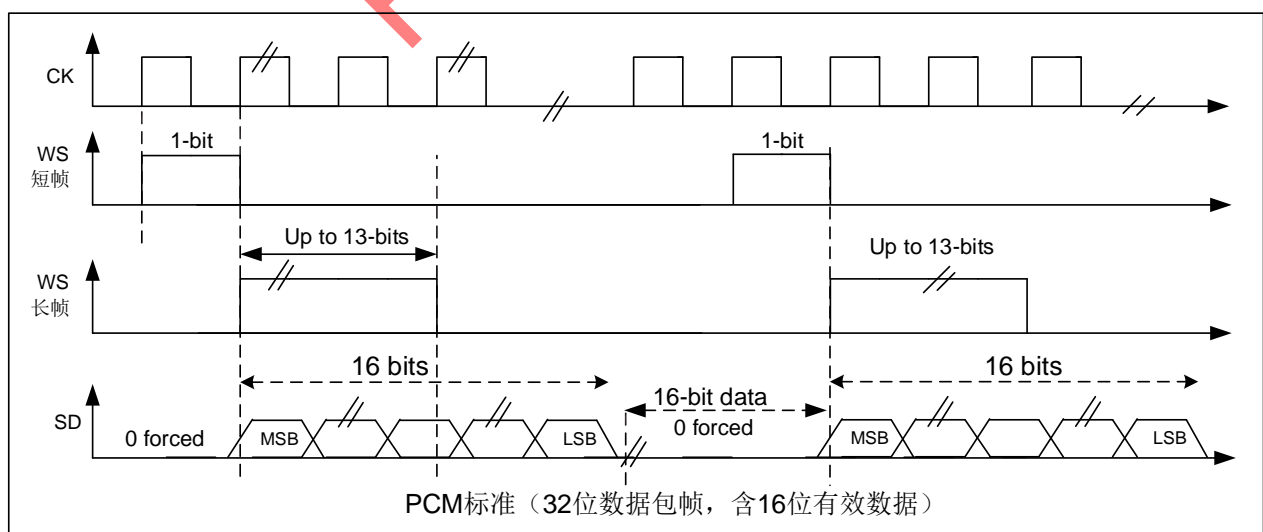


图 27-7 PCM 标准示意图

27.4.5 从模式

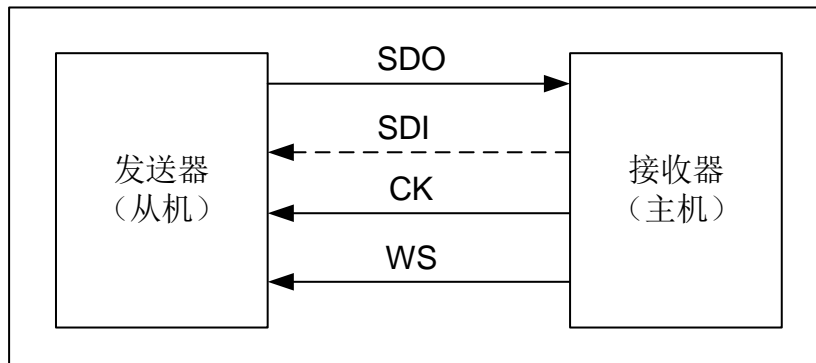


图 27-8 I2S 从模式（SD 配置成 SDO 为发送器，配置成 SDI 为接收器）

从模式支持发送或接收，由于 I2S 只能支持半双工，SD 引脚只能配置成 SDO 输出或 SDI 输入。收发状态、实现方法类同 SPI 从模式，配置流程如下：

1. 配置 SPI_I2S_GCTL.SPIEN 位为‘1’，开启模块使能；
2. 配置 SPI_I2S_GCTL.MODE 位为‘0’，使模块功能为从模式；
4. 配置寄存器 SPI_I2S_I2SCFGR 中的 I2SDIV[8: 0]、DATLEN 和 CHLEN 位，以符合希望得到的音频采样频率及数据包帧格式，计算方法请参考 27.4.7 时钟预分频器；
5. 配置 SPI_I2S_I2SCFGR.SPI_I2S 位为‘1’，使能 I2S 传输功能；
6. 配置寄存器 SPI_I2S_I2SCFGR 中的 I2SSTD[1: 0]、PCMSYNC 位，选择 I2S 传输时使用的通信标准；
7. 配置 SPI_I2S_GCTL.DMAMODE 位为‘1’，以启用 DMA 传输；
8. 开启半双工传输许可，即配置寄存器 SPI_I2S_GCTL 中的 TXEN 或 RXEN 位为‘1’（TXEN、RXEN 不可同时配置为‘1’）。

注意，从模式下发送时，在检测到 WS 的边沿之前，需要对寄存器 SPI_I2S_TXREG 进行 1 次数据写入操作；而且，从模式下接收时，在配置 RXEN 位为‘1’之前，需要一直维持 WS 输入信号在高电平。

27.4.6 主模式

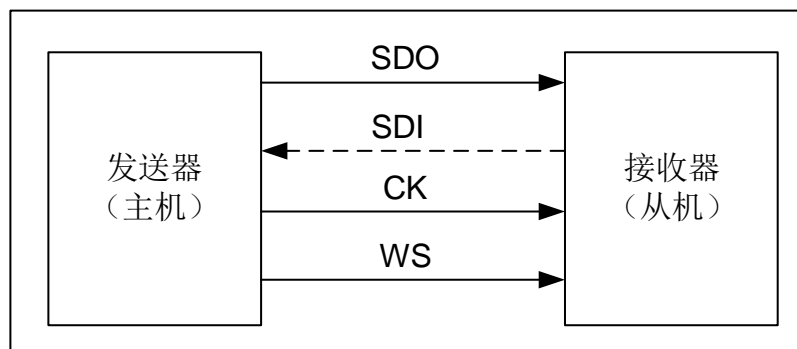


图 27-9 I2S 主模式（SD 配置成 SDO 为发送器，配置成 SDI 为接收器）

主模式也支持发送或接收，且半双工机制下 SD 引脚只能配置成 SDO 输出或 SDI 输入。收发状态、实现方法类同 SPI 从模式，配置流程如下：

1. 配置 SPI_I2S_GCTL.SPIEN 位为‘1’，开启模块使能；
2. 配置 SPI_I2S_GCTL.MODE 位为‘1’，使模块功能为主模式；
3. 配置 SPI_I2S_CCTL.CPOL 位，选择时钟空闲状态时的输出电平；
4. 配置 SPI_I2S_I2SCFGR.MCKOE 位，选择是否向外部器件提供 MCK 驱动时钟；
5. 配置寄存器 SPI_I2S_I2SCFGR 中的 I2SDIV[8: 0]、DATLEN 和 CHLEN 位，以符合希望得到的音频采样频率及数据包帧格式，计算方法请参考 27.4.7 时钟预分频器；
6. 配置 SPI_I2S_I2SCFGR.SPI_I2S 位为‘1’，使能 I2S 传输功能；
7. 配置寄存器 SPI_I2S_I2SCFGR 中的 I2SSTD[1: 0]、PCMSYNC 位，选择 I2S 传输时使用的通信标准；
8. 配置 SPI_I2S_GCTL.DMAMODE 位为‘1’，以启用 DMA 传输；
9. 开启半双工传输许可，即配置寄存器 SPI_I2S_GCTL 中的 TXEN 或 RXEN 位为‘1’(TXEN、RXEN 不可同时配置为‘1’)。

注意：主模式接收时，当接收到的字节个数达到寄存器 SPI_I2S_RXDNR 配置值时，将立即停止传输（CK 时钟输出结束，并固定在 CPOL 位配置的电平状态）。

27.4.7 时钟预分频器

I2SCLK 时钟由系统 APB 时钟提供，下图示意了 I2S 模块的预分频器电路结构：

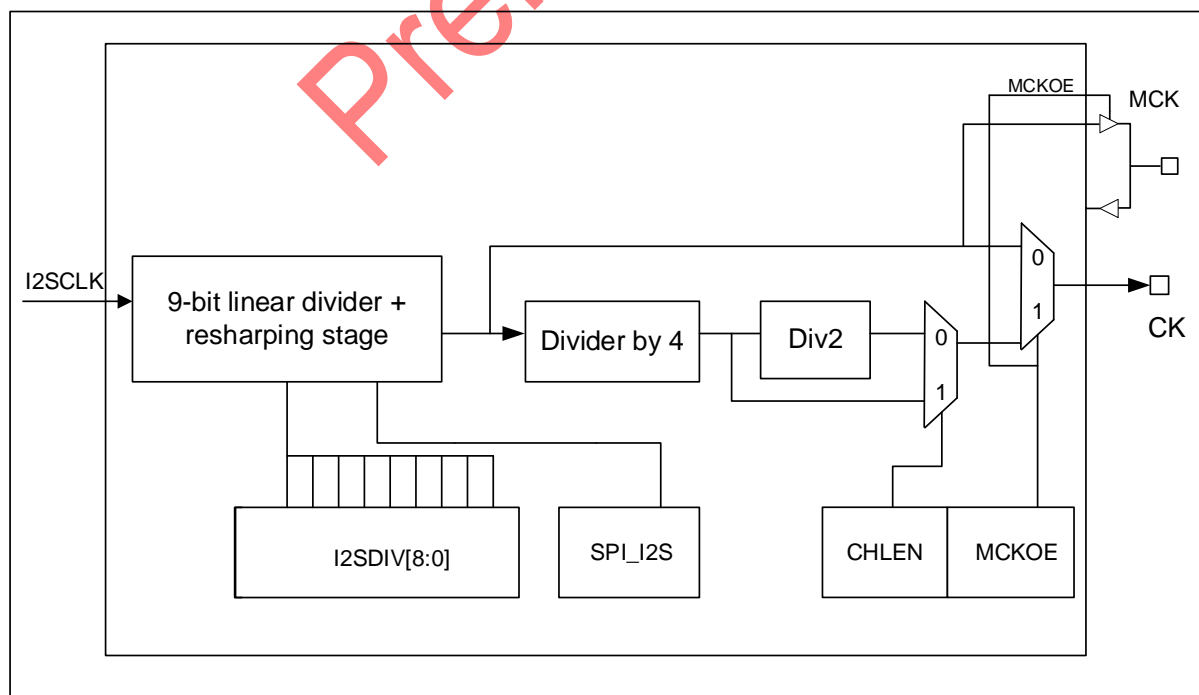


图 27-10 I2S 时钟预分频器示意图

由上图看出，当 MCKOE 位为‘0’时芯片不需要输出 MCK 时钟，预分频器直接将 I2SCLK 分频到 CK；

当 MCKOE 位为‘1’时芯片会输出 MCK 时钟，预分频器将 I2SCLK 分频后得到 MCK，然后再经过分频处理才得到 CK（分频倍数由 CHLEN 选择为 4 或 8）。

音频采样率一般常用 192KHz, 96 KHz, 48 KHz, 44.1 KHz, 32 KHz, 22.05 KHz, 16 KHz, 11.025 KHz, 8 KHz。因此可根据 I2S 时钟分频器的电路功能式样，配置寄存器 I2SCFGR 中的 I2SDIV[8: 0]、MCKOE 和 CHLEN 位来得到期望的音频采样率。

I2S 传输数据时，比特率计算公式如下表所示（CK 输出一个时钟周期对应传输 1 比特数据，因此比特率 = CK 频率 F_{CK} ）。

表 27-3 I2S 比特率计算公式

MCKOE	CHLEN	比特率计算公式
0	0	$F_{CK} = F_{I2SCLK} / I2SDIV[8: 0]$
0	1	$F_{CK} = F_{I2SCLK} / I2SDIV[8: 0]$
1	0	$F_{CK} = F_{I2SCLK} / (I2SDIV[8: 0] \times 4 \times 2)$
1	1	$F_{CK} = F_{I2SCLK} / (I2SDIV[8: 0] \times 4)$

音频采样率（ F_s ）和 I2S 比特率的关系由如下的公式定义：

$$F_s = \text{I2S 比特率} / (\text{通道长度} \times \text{通道数}) = F_{CK} / (\text{通道长度} \times \text{通道数})$$

注：通道长度，即数据包帧长度，可配置为 16 位或 32 位；通道数为左右声道，值固定为 2。

综上所述，根据 I2SDIV[8: 0]、MCKOE 和 CHLEN 位的配置情况，得到音频采样率与 F_{I2SCLK} （APB 时钟频率）的关系如下：

表 27-4 音频采样率与 F_{I2SCLK} （APB 时钟频率）计算关系

MCKOE	CHLEN	音频采样率 F_s 与 F_{I2SCLK} 计算关系	注意事项
0	0	$F_{I2SCLK} = I2SDIV[8: 0] \times 32 \times F_s$	I2SDIV[8: 0]应配置在 2~511 范围内
0	1	$F_{I2SCLK} = I2SDIV[8: 0] \times 64 \times F_s$	I2SDIV[8: 0]应配置在 2~511 范围内
1	0	$F_{I2SCLK} = I2SDIV[8: 0] \times 256 \times F_s$	I2SDIV[8: 0]应配置在 1~511 范围内
1	1	$F_{I2SCLK} = I2SDIV[8: 0] \times 256 \times F_s$	I2SDIV[8: 0]应配置在 1~511 范围内

当 MCKOE 位为‘1’，芯片会输出 MCK 驱动时钟，且满足 $F_{MCLK} = 256 \times F_s$ ；此时假设 $F_s = 192\text{KHz}$ ， $I2SDIV[8: 0] = 1$ ，则要求 APB 时钟（ F_{I2SCLK} ）= $1 \times 256 \times 192\text{KHz} = 49.152\text{MHz}$ 。

针对常用的音频采样率，当按照 $F_{I2SCLK} = 48\text{MHz}$ 计算时，误差如下：

表 27-5 I2S 音频采样率误差表

F_s	声道位宽	MCKOE=1				MCKOE=0			
		I2SDIV理想值	I2SDIV配置值	真实 F_s	误差 (%)	I2SDIV理想值	I2SDIV配置值	真实 F_s	误差 (%)
192000	32	0.976563	1	187500	-2.34375	3.90625	4	187500	-2.34375
192000	16	0.976563	1	187500	-2.34375	7.8125	8	187500	-2.34375
96000	32	1.953125	2	93750	-2.34375	7.8125	8	93750	-2.34375
96000	16	1.953125	2	93750	-2.34375	15.625	16	93750	-2.34375
48000	32	3.90625	4	46875	-2.34375	15.625	16	46875	-2.34375
48000	16	3.90625	4	46875	-2.34375	31.25	31	48387.1	0.806452
44100	32	4.251701	4	46875	6.292517	17.0068	17	44117.65	0.040016
44100	16	4.251701	4	46875	6.292517	34.01361	34	44117.65	0.040016

		MCKOE=1				MCKOE=0			
32000	32	5.859375	6	31250	-2.34375	23.4375	23	32608.7	1.902174
32000	16	5.859375	6	31250	-2.34375	46.875	47	31914.89	-0.26596
22050	32	8.503401	9	20833.33333	-5.51776	34.01361	34	22058.82	0.040016
22050	16	8.503401	9	20833.33333	-5.51776	68.02721	68	22058.82	0.040016
16000	32	11.71875	12	15625	-2.34375	46.875	47	15957.45	-0.26596
16000	16	11.71875	12	15625	-2.34375	93.75	94	15957.45	-0.26596
11025	32	17.0068	17	11029.41176	0.040016	68.02721	68	11029.41	0.040016
11025	16	17.0068	17	11029.41176	0.040016	136.0544	136	11029.41	0.040016
8000	32	23.4375	23	8152.173913	1.902174	93.75	94	7978.723	-0.26596
8000	16	23.4375	23	8152.173913	1.902174	187.5	188	7978.723	-0.26596

27.4.8 中断

27.4.8.1 状态标志

I2S 与 SPI 共用同一个中断向量。

表 27-6 I2S 中断状态

中断事件	中断寄存器标志
TX BUFFER 空	TX_INTF
RX BUFFER 非空	RX_INTF
TX BUFFER 向下溢出	UNDERRUN_INTF
RX BUFFER 向上溢出	RXOERR_INTF
帧传输错误	FRE_INTF

TX BUFFER 空 (TX_INTF): 表示当前发送缓冲中的数据已经被读走。

RX BUFFER 非空 (RX_INTF): 表示当前接收缓冲中的数据有效, APB 总线可读。

TX BUFFER 向下溢出 (UNDERRUN_INTF): 表示从模式下发送缓冲中的数据在 I2S 总线上被传输了两次。

RX BUFFER 向上溢出 (RXOERR_INTF): 表示上一个接收数据未被读走, 且已经被覆盖。

帧传输错误 (FRE_INTF): 表示 I2S 传输协议不匹配 (仅从模式下有效)。

此外, I2S 模式下有以下 2 个专用的状态标志:

忙标志 (BSY): 表示 I2S 正在传输中。

声道标志 (CHSIDE): 表示正在传输的声道为左声道还是右声道。

27.4.9 DMA 传输

在 I2S 模式, DMA 的工作方式与 SPI 模式完全相同。

27.5 寄存器

27.5.1 寄存器总览

表 27-7 SPI_I2S 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SPI_I2S_TXREG	发送数据寄存器	0x00000000
0x04	SPI_I2S_RXREG	接收数据寄存器	0x00000000
0x08	SPI_I2S_CSTAT	当前状态寄存器	0x00002001
0x0C	SPI_I2S_INTSTAT	中断状态寄存器	0x00000000
0x10	SPI_I2S_INTEN	中断使能寄存器	0x00000000
0x14	SPI_I2S_INTCLR	中断清除寄存器	0x00000000
0x18	SPI_I2S_GCTL	全局控制寄存器	0x00000004
0x1C	SPI_I2S_CCTL	通用控制寄存器	0x00000008
0x20	SPI_I2S_SPBRG	波特率发生器	0x00000002
0x24	SPI_I2S_RXDNR	接收数据个数寄存器	0x00000001
0x28	SPI_I2S_NSSR	从机片选寄存器	0x000000FF
0x2C	SPI_I2S_EXTCTL	数据控制寄存器	0x00000008
0x30	SPI_I2S_I2SCFGR	I2S 配置寄存器	0x00010000

27.5.2 SPI_I2S_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	Type	Reset	Description
31: 0	TXREG	rw	0x0000 0000	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

27.5.3 SPI_I2S_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	Type	Reset	Description
31: 0	RXREG	r	0x0000 0000	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。

27.5.4 SPI_I2S_CSTAT 当前状态寄存器

偏移地址：0x08

复位值：0x0000 2001

31	30	29	28	27	26	25	24	23	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.		CHSIDE	BUSY	RXFADDR				TXFADDR				RXAVL_4BYTE	TXFULL	RXAVL	TXEPT
		r	r	r				r				r	r	r	r

Bit	Field	Type	Reset	Description
31: 14	Reserved			始终读为 0
13	CHSIDE	r	0x01	声道标志位 0：表示正在传输的声道为左声道 1：表示正在传输的声道为右声道 注：SPI 模式下不可用；且 I2S 的 PCM 标准模式下无意义。
12	BUSY	r	0x00	忙标志位 表示 I2S 或 SPI 正在传输中
11: 8	RXFADDR	r	0x00	当前接收缓冲中有效字节个数
7: 4	TXFADDR	r	0x00	当前发送缓冲中有效字节个数

Bit	Field	Type	Reset	Description
3	RXAVL_4BYTE	r	0x00	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 注: 工作在 I2S 模式下时, 该位在接收到一个声道数据后置位 (如 CHLEN=0, 在接收到 16bit 数据后置位)。
2	TXFULL	r	0x00	发送缓冲满标志位 (Transmitter FIFO full status bit) 0: 发送缓冲未滿 1: 发送缓冲滿
1	RXAVL	r	0x00	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	r	0x01	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。

27.5.5 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								FRE_IN TF	TXEPT_ INTF	RXFULL_ INTF	RXMAT CH_INT F	RXOER R_INTF	UNDER RUN_IN TF	RX_INT F	TX_INT F
								r	r	r	r	r	r	r	r

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留, 始终读为 0
7	FRE_INTF	r	0x00	帧传输错误中断标志位: 表示 I2S/TI 模式下传输协议不匹配 (仅从模式有效)
6	TXEPT_INTF	r	0x00	发送端空中断标志位 (Transmitter empty interrupt flag bit) 硬件自动置位, 写 INTCLR.TXEPT_ICLR 位为'1'清除。 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: 该位是中断状态信号, TXEPT 是状态信号。

Bit	Field	Type	Reset	Description
5	RXFULL_INTF	r	0x00	接收缓冲满中断标志位 (RX FIFO full interrupt flag bit) 硬件自动置位, 写 INTCLR.RXFULL_ICLR 位为'1'清除。 0: RX 缓冲未满 1: RX 缓冲满
4	RXMATCH_INTF	r	0x00	接收到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.RXMATCH_ICLR 位为'1'清除。 0: 未完成 RXDNR 寄存器指定的字节数 1: 接收了 RXDNR 寄存器指定的字节数
3	RXOERR_INTF	r	0x00	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.RXOERR_ICLR 位为'1'清除。 0: 没有溢出错误 1: 溢出错误
2	UNDERRUN_INTF	r	0x00	SPI 从机模式下溢标志位 (SPI underrun interrupt flag bit) 硬件自动置位, 写 INTCLR.UNDERRUN_ICLR 位为'1'清除。 0: 没有下溢错误 1: 下溢错误
1	RX_INTF	r	0x00	接收缓冲数据有效中断标志位 (Receive data available interrupt flag bit) 硬件自动置位, 写 INTCLR.RX_ICLR 位为'1'清除。当接收缓冲接收了一个完整有效数据时置位。 0: 接收缓冲空 1: 接收缓冲接收到一个完整有效数据 注: 有效数据包含的字节数, 参考 27.3.6.1 状态标志。
0	TX_INTF	r	0x00	发送缓冲空中断标志位 (缓冲为空, 可写 TXREG) (Transmit FIFO available interrupt flag bit) 硬件自动置位, 发送缓冲不为空时自动清零。 0: 发送缓冲不为空 1: 发送缓冲为空

27.5.6 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Res.	FRE_IEN	TXEPT_IEN	RXFULL_IEN	RXMATCH_IEN	RXOERR_IEN	UNDERUN_IEN	RX_IEN	TX_IEN
	rW	rW	rW	rW	rW	rW	rW	rW

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留，始终读为 0
7	FRE_IEN	rw	0x00	帧传输错误中断使能 (FRE_IEN): 0: 中断禁止 1: 中断使能
6	TXEPT_IEN	rw	0x00	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	rw	0x00	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	rw	0x00	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
3	RXOERR_IEN	rw	0x00	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	rw	0x00	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	rw	0x00	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	rw	0x00	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

27.5.7 SPI_I2S_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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								FRE_ICLR	TXEPT_ICLR	RXFULL_ICLR	RXMATCH_ICLR	RXOERR_ICLR	UNDERRUN_ICLR	RX_ICLR	TX_ICLR
								w	w	w	w	w	w	w	w

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留，始终读为 0
7	FRE_ICLR	w	0x00	帧传输错误中断清除 0: 写 0 无意义 1: 写 1 清除中断
6	TXEPT_ICLR	w	0x00	发送端空中断清除位 (Transmitter empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
5	RXFULL_ICLR	w	0x00	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	w	0x00	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	w	0x00	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	w	0x00	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	w	0x00	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	w	0x00	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

27.5.8 SPI_I2S_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
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Res.														PAD_SEL	
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD_SEL			Res.	DW8_32	NSS	DMAMO DE	Res.				RXEN	TXEN	MODE	INTEN	SPIEN
rw				rw	rw	rw					rw	rw	rw	rw	rw

Bit	Field	Description																																																																																																																													
31: 18	Reserved	保留，始终读为 0																																																																																																																													
17: 13	PAD_SEL	<p>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：</p> <table border="1"> <thead> <tr> <th>PAD_SEL</th><th>SCL</th><th>MOSI</th><th>NSS</th><th>MISO</th></tr> </thead> <tbody> <tr><td>0</td><td>PAD0</td><td>PAD1</td><td>PAD2</td><td>PAD3</td></tr> <tr><td>1</td><td>PAD0</td><td>PAD1</td><td>PAD3</td><td>PAD2</td></tr> <tr><td>2</td><td>PAD0</td><td>PAD2</td><td>PAD1</td><td>PAD3</td></tr> <tr><td>3</td><td>PAD0</td><td>PAD2</td><td>PAD3</td><td>PAD1</td></tr> <tr><td>4</td><td>PAD0</td><td>PAD3</td><td>PAD1</td><td>PAD2</td></tr> <tr><td>5</td><td>PAD0</td><td>PAD3</td><td>PAD2</td><td>PAD1</td></tr> <tr><td>6</td><td>PAD1</td><td>PAD0</td><td>PAD2</td><td>PAD3</td></tr> <tr><td>7</td><td>PAD1</td><td>PAD0</td><td>PAD3</td><td>PAD2</td></tr> <tr><td>8</td><td>PAD1</td><td>PAD2</td><td>PAD0</td><td>PAD3</td></tr> <tr><td>9</td><td>PAD1</td><td>PAD2</td><td>PAD3</td><td>PAD0</td></tr> <tr><td>10</td><td>PAD1</td><td>PAD3</td><td>PAD0</td><td>PAD2</td></tr> <tr><td>11</td><td>PAD1</td><td>PAD3</td><td>PAD2</td><td>PAD0</td></tr> <tr><td>12</td><td>PAD2</td><td>PAD0</td><td>PAD1</td><td>PAD3</td></tr> <tr><td>13</td><td>PAD2</td><td>PAD0</td><td>PAD3</td><td>PAD1</td></tr> <tr><td>14</td><td>PAD2</td><td>PAD1</td><td>PAD0</td><td>PAD3</td></tr> <tr><td>15</td><td>PAD2</td><td>PAD1</td><td>PAD3</td><td>PAD0</td></tr> <tr><td>16</td><td>PAD2</td><td>PAD3</td><td>PAD0</td><td>PAD1</td></tr> <tr><td>17</td><td>PAD2</td><td>PAD3</td><td>PAD1</td><td>PAD0</td></tr> <tr><td>18</td><td>PAD3</td><td>PAD0</td><td>PAD1</td><td>PAD2</td></tr> <tr><td>19</td><td>PAD3</td><td>PAD0</td><td>PAD2</td><td>PAD1</td></tr> <tr><td>20</td><td>PAD3</td><td>PAD1</td><td>PAD0</td><td>PAD2</td></tr> <tr><td>21</td><td>PAD3</td><td>PAD1</td><td>PAD2</td><td>PAD0</td></tr> <tr><td>22</td><td>PAD3</td><td>PAD2</td><td>PAD0</td><td>PAD1</td></tr> <tr><td>23</td><td>PAD3</td><td>PAD2</td><td>PAD1</td><td>PAD0</td></tr> </tbody> </table>	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
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																																																																																																																											
12	Reserved	预留，必须保持复位值。																																																																																																																													
11	DW8_32	<p>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：I2S 模式下固定为 1。</p>																																																																																																																													

Bit	Field	Description
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 注: I2S 模式下固定为 0。
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。
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/I2S 中断使能位 (SPI/I2S interrupt enable bit) 0: 禁止 SPI/I2S 中断 1: 使能 SPI/I2S 中断
0	SPIEN	SPI/I2S 选择位 (SPI/I2S select bit) 0: SPI/I2S 禁止 (复位状态) 1: SPI/I2S 使能

27.5.9 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									CPHAS EL	TXEDG E	RXEDG E	SPILEN	LSBFE	CPOL	CPHA
									rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 7	Reserved	保留, 始终读为 0

Bit	Field	Description
6	CPHASEL	CPHA 极性取反选择 (CPHA polarity invert select) 0: CPHA 设置保持不变 1: 将 CPHA 设置值取反 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 注: I2S 模式下固定为 1。
5	TXEDGE	发送数据相位调整位 (从模式) (Transmit data edge select) 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 (从模式输入时钟较慢)。 1: 发送数据立即发送到数据总线 可用于高速传输时 (从模式输入时钟较快, 超过 10MHz) 注: 建议该位配置为 1, 以免通信速率较快时不满足 AC 特性, 导致从模式下发送数据不符合主机采样时序要求。
4	RXEDGE	接收数据采样时钟沿选择位 (主模式) (Receive data edge select) 0: 在传输数据位的中间采样数据 1: 在传输数据位的尾时钟沿采样数据 (用于高速传输, 后移采样时间点以应对板级布线及从模式发送数据的延迟) 注: 建议该位配置为 1, 以免通信速率较快时不满足 AC 特性, 导致主模式下接收数据不正确。
3	SPILEN	SPI 数据宽度位 (SPI character length bit) 该位在 DW8_32=0 时有效; DW8_32=1 时, 该位需保持为 1。 0: 7 位数据 1: 8 位数据 (缺省) 注: I2S 模式下固定为 1。
2	LSBFE	LSBFE: LSB 在前使能位 (LSI first enable bit) 0: 数据传输或接收时最高位在前 1: 数据传输或接收时最低位在前 注: I2S 模式下固定为 0。
1	CPOL	时钟极性标志位 (Clock polarity select bit) 0: 时钟在空闲状态为低电平 (两次传输之间) 1: 时钟在空闲状态为高电平 (两次传输之间)
0	CPHA	时钟相位选择位 (Clock phase select bit) 0: 第一个数据位采样从第二个时钟边沿开始 1: 第一个数据位采样从第一个时钟边沿开始 注: I2S 模式下固定为 0。

27.5.10 SPI_I2S_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
Res.															

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

Bit	Field	Type	Reset	Description
31: 16	Reserved			保留，始终读为 0
15: 0	SPBRG	rw	0x0002	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： 波特率 = Fpclk / SPBRG（Fpclk 是 APB 时钟频率） 注：不能对该寄存器写值为 0、1。

27.5.11 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDNR															
rw															

Bit	Field	Type	Reset	Description
31: 16	Reserved			保留，始终读为 0
15: 0	RXDNR	rw	0x0001	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。

27.5.12 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															NSS

rw

Bit	Field	Type	Reset	Description
31: 1	Reserved			保留，必须保持复位值。
0	NSS	rw	0x1	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

27.5.13 SPI_I2S_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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											EXTLEN				
rw															

Bit	Field	Type	Reset	Description
31: 5	Reserved			保留，始终读为 0
4: 0	EXTLEN	rw	0x08	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_I2S_GCTL.DW8_32=1 时有效；当 DW8_32=0 时，必须保持初始值 5'h8)。I2S 模式下此配置为无效： I2S 模式下，当 CHLEN=1, EXTLEN 值固定为 5'b00000(32 位)； I2S 模式下，当 CHLEN=0, EXTLEN 值固定为 5'b10000(16 位)。 EXTLEN[4: 0] 应在 SPI_I2S_GCTL.SPIEN 位为'0'时配置，且在通信过程中不能被修改。

27.5.14 SPI_I2S_I2SCFGR I2S 配置寄存器

偏移地址：0x30

复位值：0x00010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							I2SDIV								
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				MCKOE	SPI_I2S	Res.			PCMSY N	I2SSTD		Res.	DATLEN		CHLEN
				rw	rw				rw	rw			rw		rw

Bit	Field	Description
31: 25	Reserved	保留，始终读为 0
24: 16	I2SDIV	I2S 预分频器的分频系数 注：主模式下才有效，且不可配置该位值为 0；当主模式下不输出 MCK 驱动时钟，即 MCKOE=0 时，不可配置该位值为 1（不支持 1 分频）。
15: 12	Reserved	保留，始终读为 0
11	MCKOE	主时钟输出使能 0：主模式驱动时钟 MCK 输出禁止 1：主模式驱动 MCK 输出使能
10	SPI_I2S	模块功能选择 0：启用 SPI 功能 1：启用 I2S 功能 注：当配置该位为 1，即 I2S 模式下，以下寄存器功能位会被硬件固定，无需软件配置。 GCTL 寄存器中的 DW8_32、NSS 位； CCTL 寄存器中的 CPHASEL、SPILEN、LSBFE、CPHA 位； EXTCTL 寄存器中的 EXTLEN 位。
9: 7	Reserved	保留，始终读为 0
6	PCMSYNC	PCM 标准帧同步模式 0：短帧同步模式 1：长帧同步模式
5: 4	I2SSTD	I2S 标准 00：PHILIPS 标准 01：MSB 对齐标准 10：LSB 对齐标准 11：PCM 标准
3	Reserved	保留，始终读为 0

Bit	Field	Description
2: 1	DATLEN	数据长度 00: 16 位宽 01: 24 位宽 10: 32 位宽 11: 禁止设置该值 注: 当 CHLEN = 0 时, DATLEN 值由硬件固定为 2'b00。
0	CHLEN	声道长度 (每个声道包含的数据位数) 0: 16 位宽 1: 32 位宽

Preliminary

28 I2C 内部集成电路接口

28.1 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）。

28.2 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）

28.3 功能描述

28.3.1 功能框图

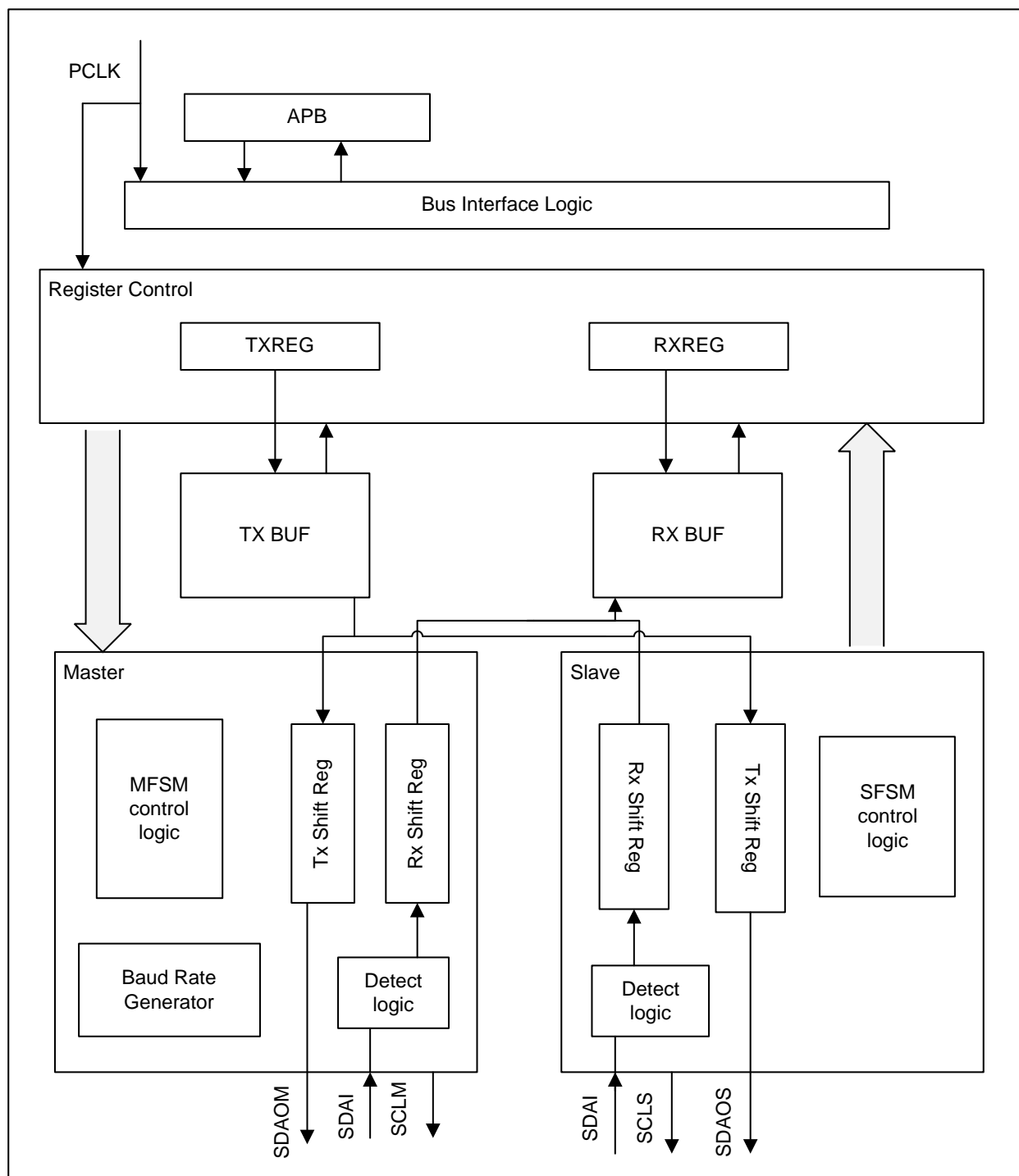


图 28-1 I2C 功能框图

28.3.2 信号描述

表 28-1 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。

28.3.3 I2C 协议

28.3.3.1 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。

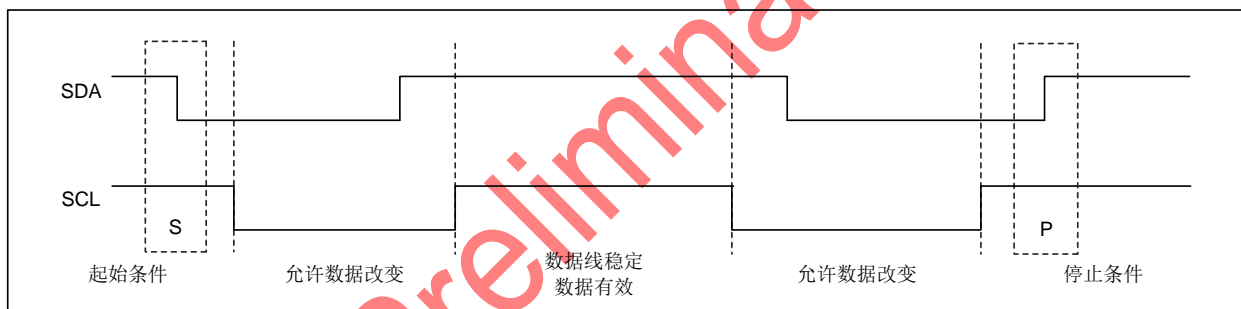


图 28-2 起始和停止条件

28.3.3.2 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。

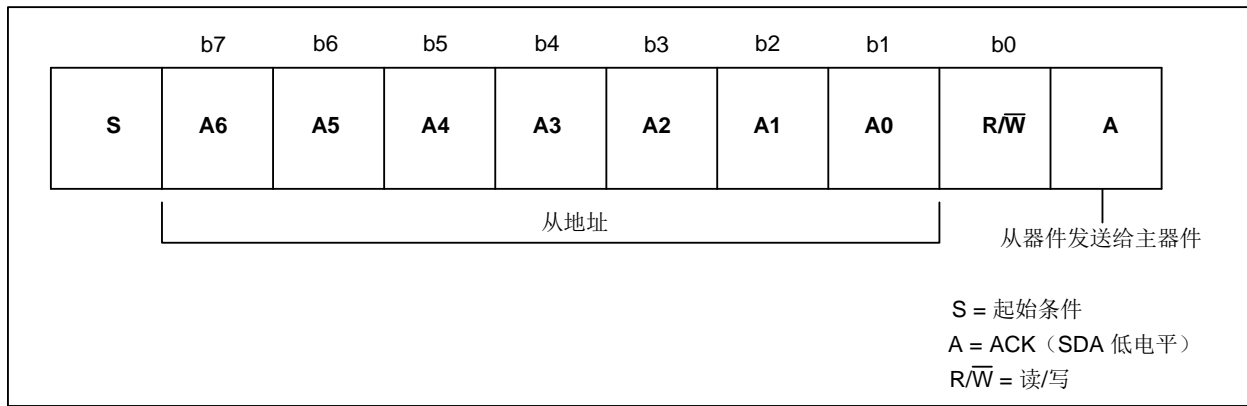


图 28-3 7 位地址格式

- 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：

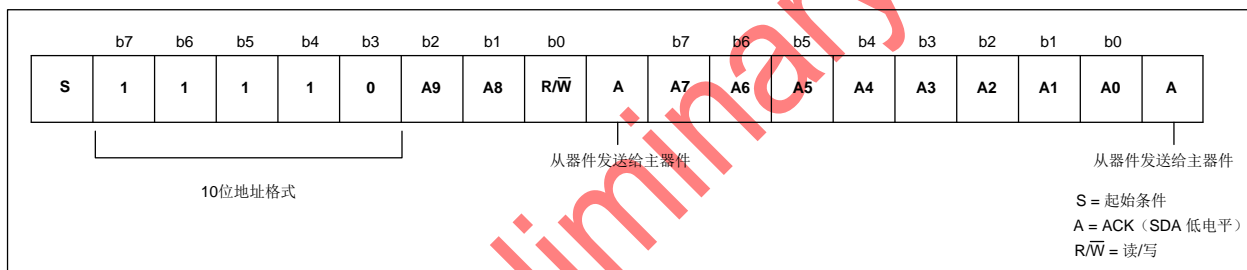


图 28-4 10 位地址格式

下表定义了 I2C 首字节的特殊用途和保留地址。

表 28-2 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

28.3.3.3 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。

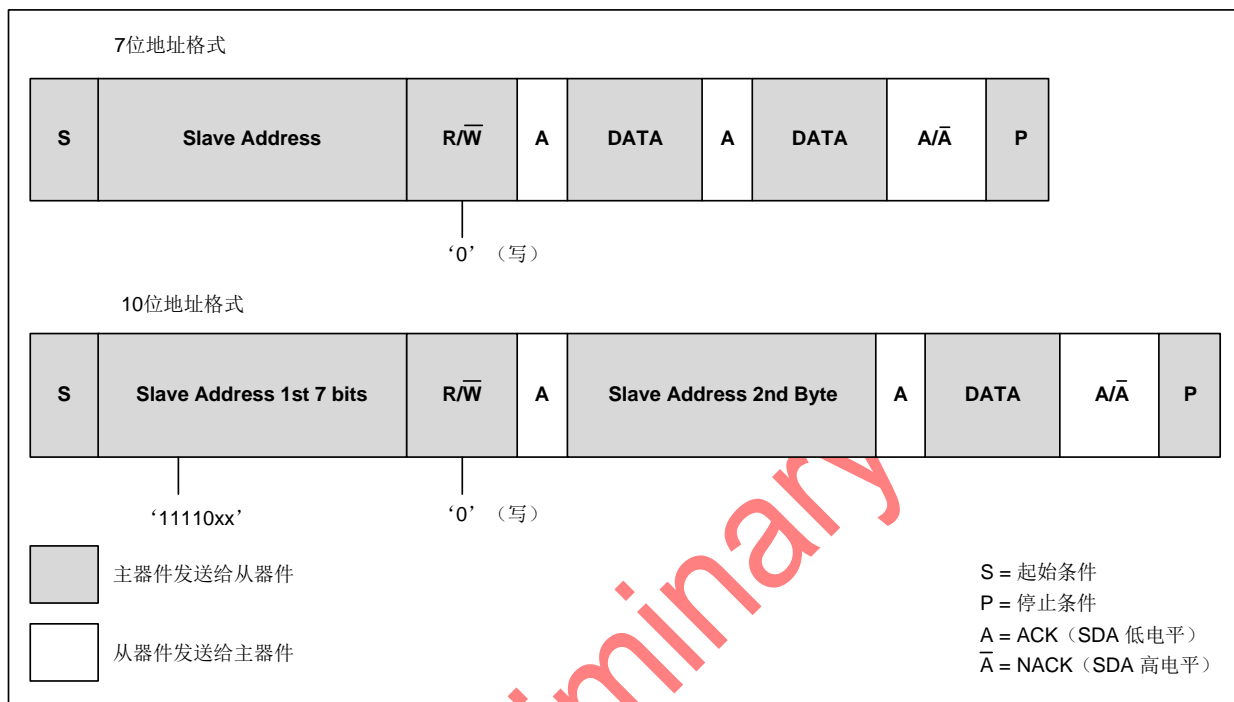


图 28-5 主发送协议

● 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。

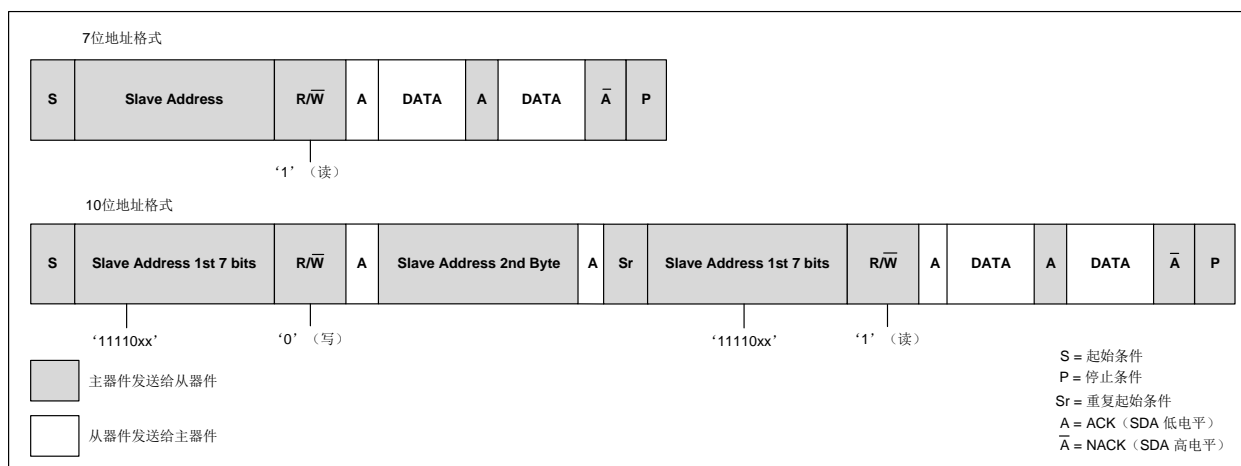


图 28-6 主接收协议

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件通

信。

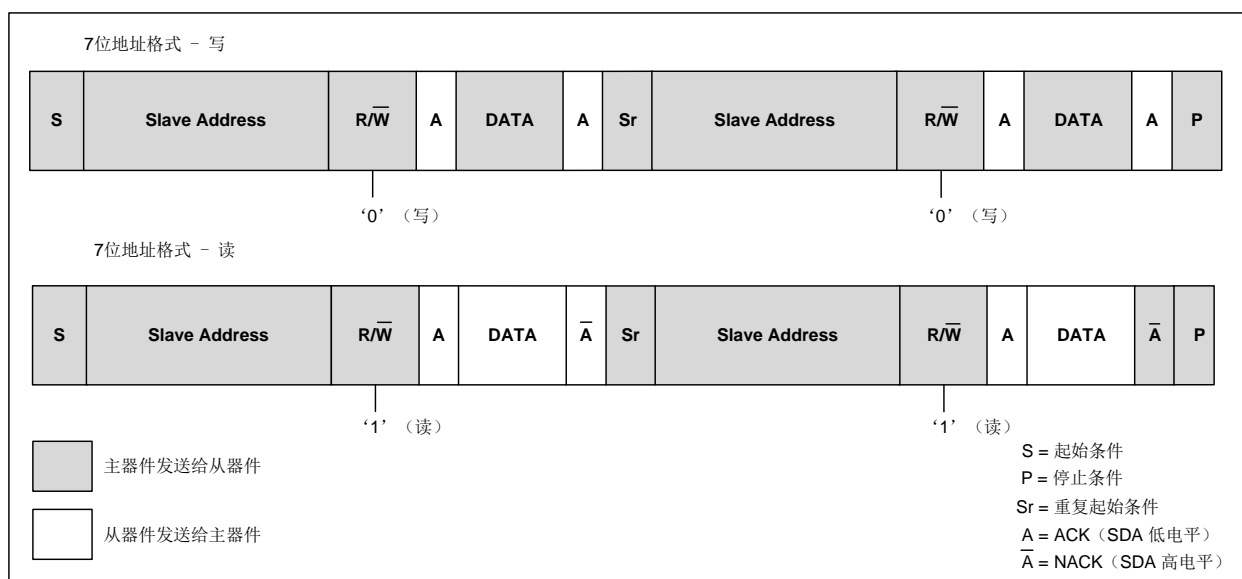


图 28-7 带 RESTART (SR) 信号的主发送和接收协议

● 起始字节传输协议

起始字节传输协议用于无专用 I2C 硬件模块的系统。当 I2C 接口模块作为主器件时，在每次传输开始，可以为需要的从器件产生起始字节输出。

如下图所示，该协议由 7 个 0 以及 1 个 1 组成。处理器可以在地址阶段用低速采样来查询总线。一旦检测到 0，处理器可以从低速采样切换到主器件的正常速率。

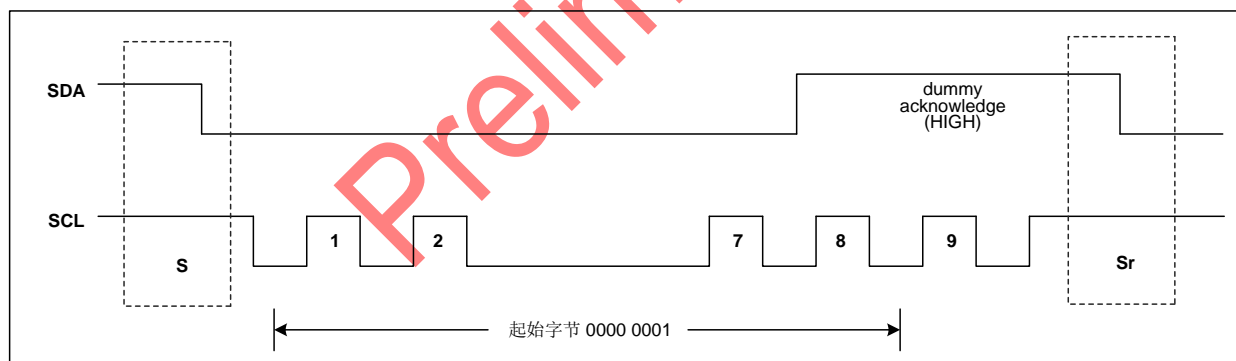


图 28-8 起始字节传输

起始字节程序流程如下：

- 1) 主器件产生起始条件
- 2) 主器件发送起始字节 (0000 0001)
- 3) 主器件发送应答相关的时钟脉冲 (为了符合总线上的字节处理格式)
- 4) 没有从器件响应 ACK 信号
- 5) 主器件产生重复起始条件 (RESTART)

I2C 硬件接收器无需响应起始字节，因为这是一个保留地址，且地址会在 RESTART 后复位。

28.3.3.4 发送缓冲管理以及起始、停止和重复起始条件产生

发送为空时 I2C 不产生停止条件，而是拉低 SCL 使总线停止，直到有新数据为止。只有写 1 到 I2C_CR.STOP（位 9）或 I2C_ENR.ABORT（位 1）时才会产生停止条件。

下图显示了 I2C_DR 寄存器的位。

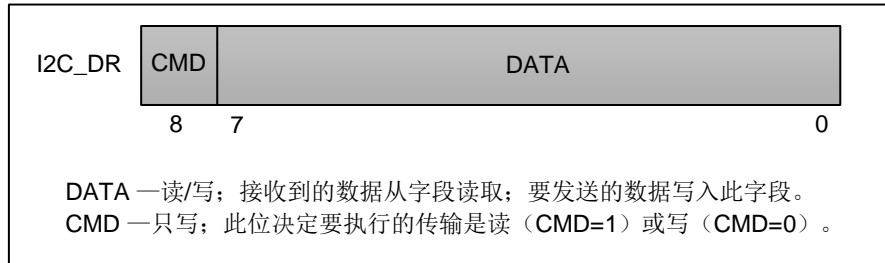


图 28-9 I2C_DR 寄存器

下图描述了 I2C 模块工作在主发送模式下，TX FIFO 变为空时的时序。

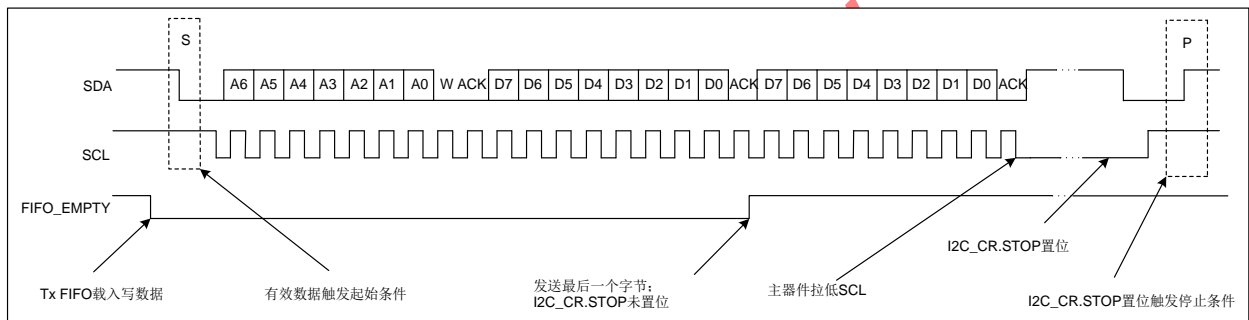


图 28-10 主发送，TX FIFO 为空

下图描述了 I2C 模块工作在主接收模式下，TX FIFO 变为空时的时序。

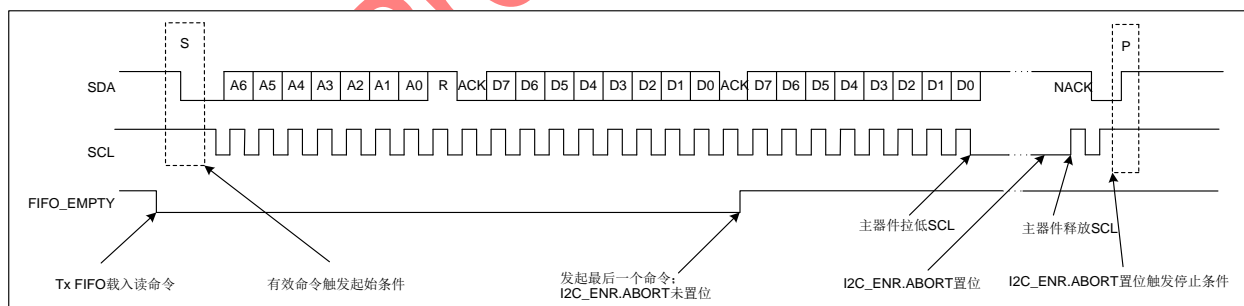


图 28-11 主接收，TX FIFO 为空

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。

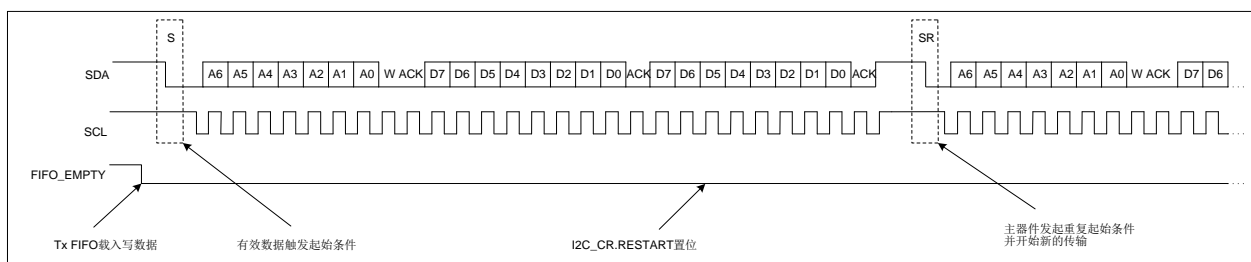


图 28-12 主发送，发生 RESTART

下图描述了 I2C 模块工作在主接收模式下，发生 RESTART 时的时序。

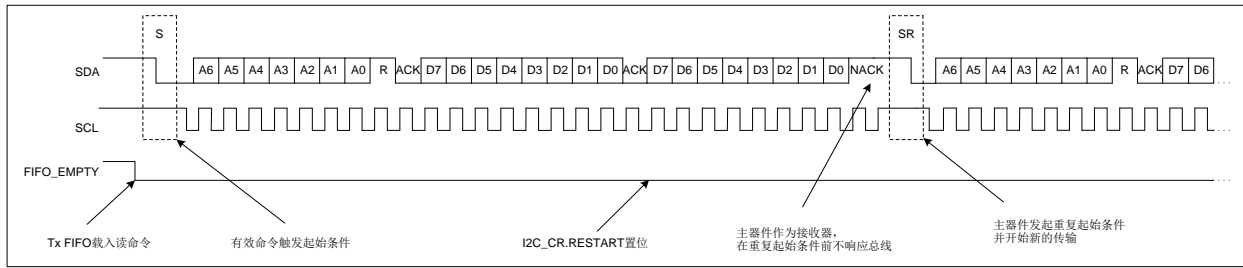


图 28-13 主接收，发生 RESTART

28.3.3.5 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。

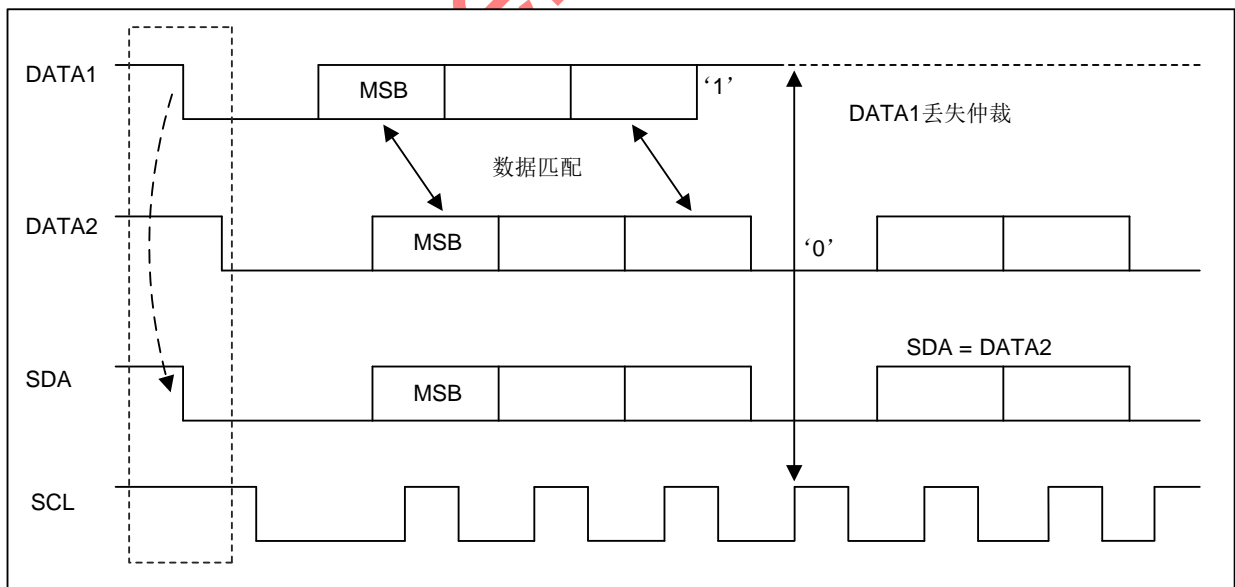


图 28-14 两个主器件仲裁

28.3.3.6 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。

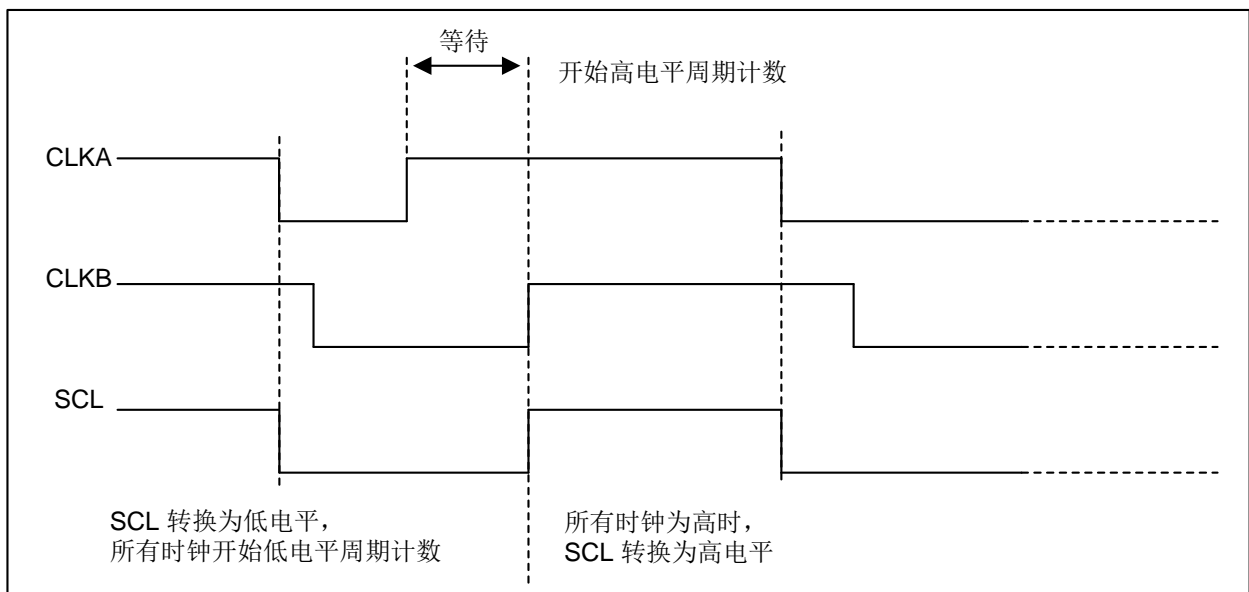


图 28-15 时钟同步（示意图）

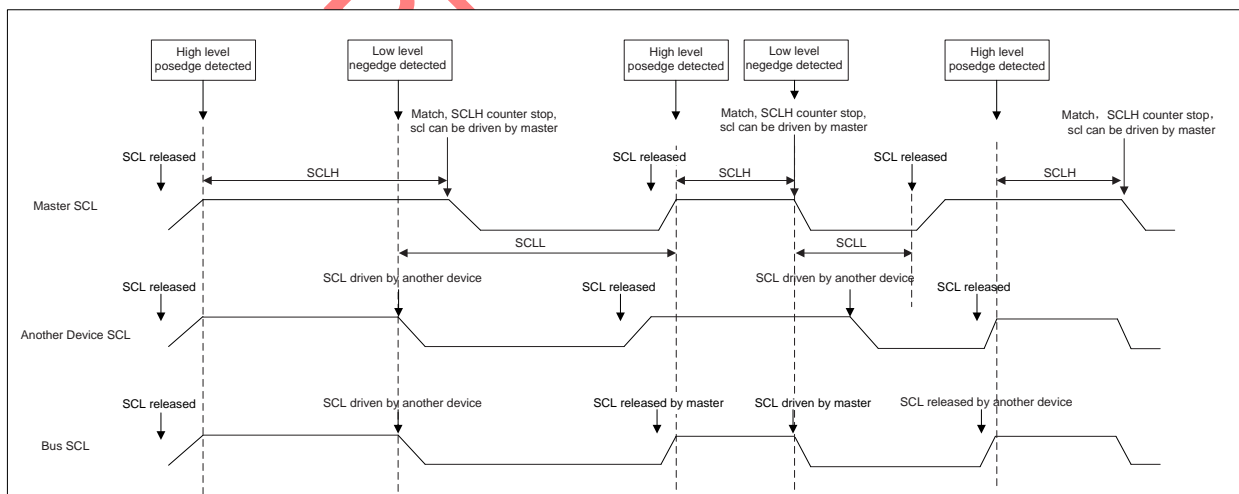


图 28-16 时钟同步（时序图）

28.3.3.7 SCL 配置

I2C 的 SCL 可参考如下配置：

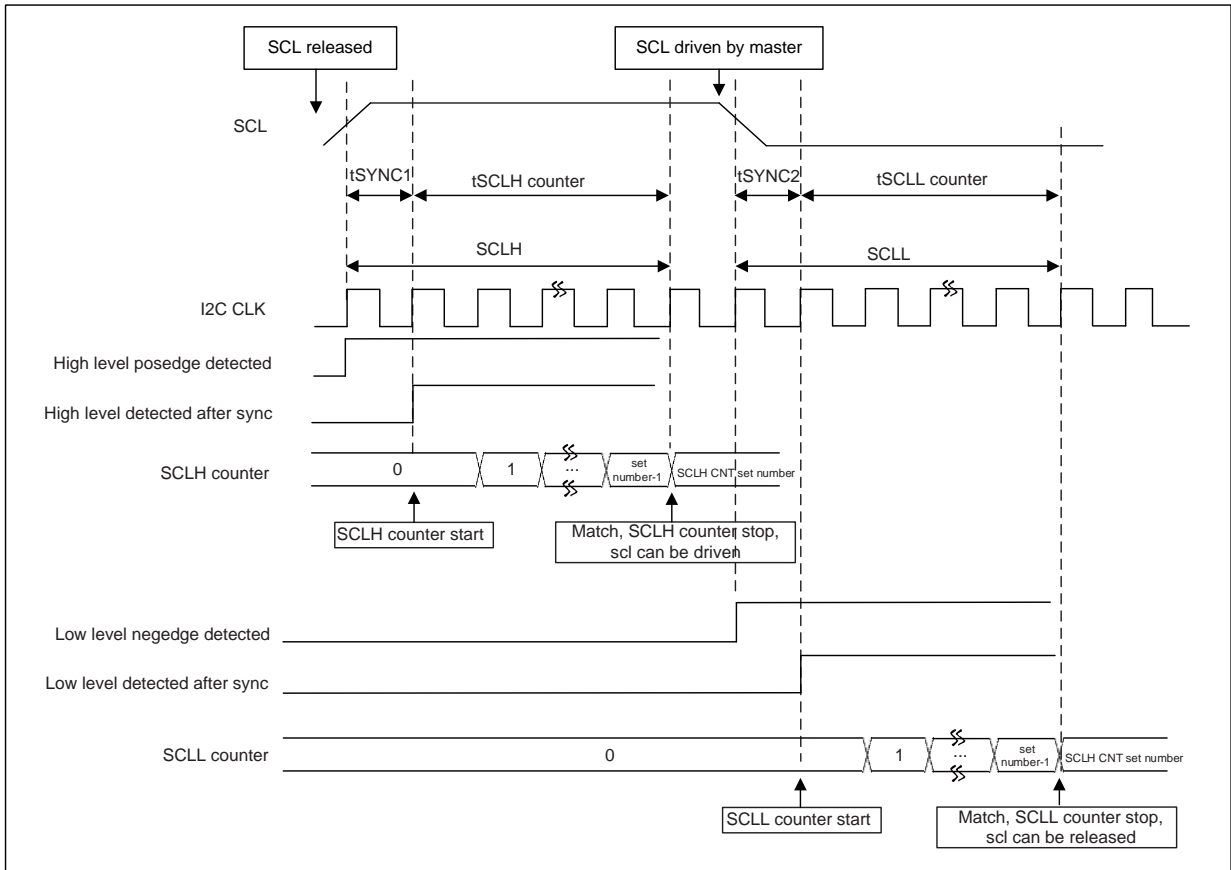


图 28-17 SCL 生成时序

- 标准模式:

$$SCLH = (SSHR + 12) \times I2C\ CLK + tSYNC1$$

$$SCLL = (SSLR + 1) \times I2C\ CLK + tSYNC2$$

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

28.3.4 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）。

28.3.4.1 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：

- a. 读到 I2C_RAWISR.RD_REQ（位 5）为 1 等效于产生了 RD_REQ 中断
- b. 软件必须采取行动以满足 I2C 传输的要求
- c. 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:

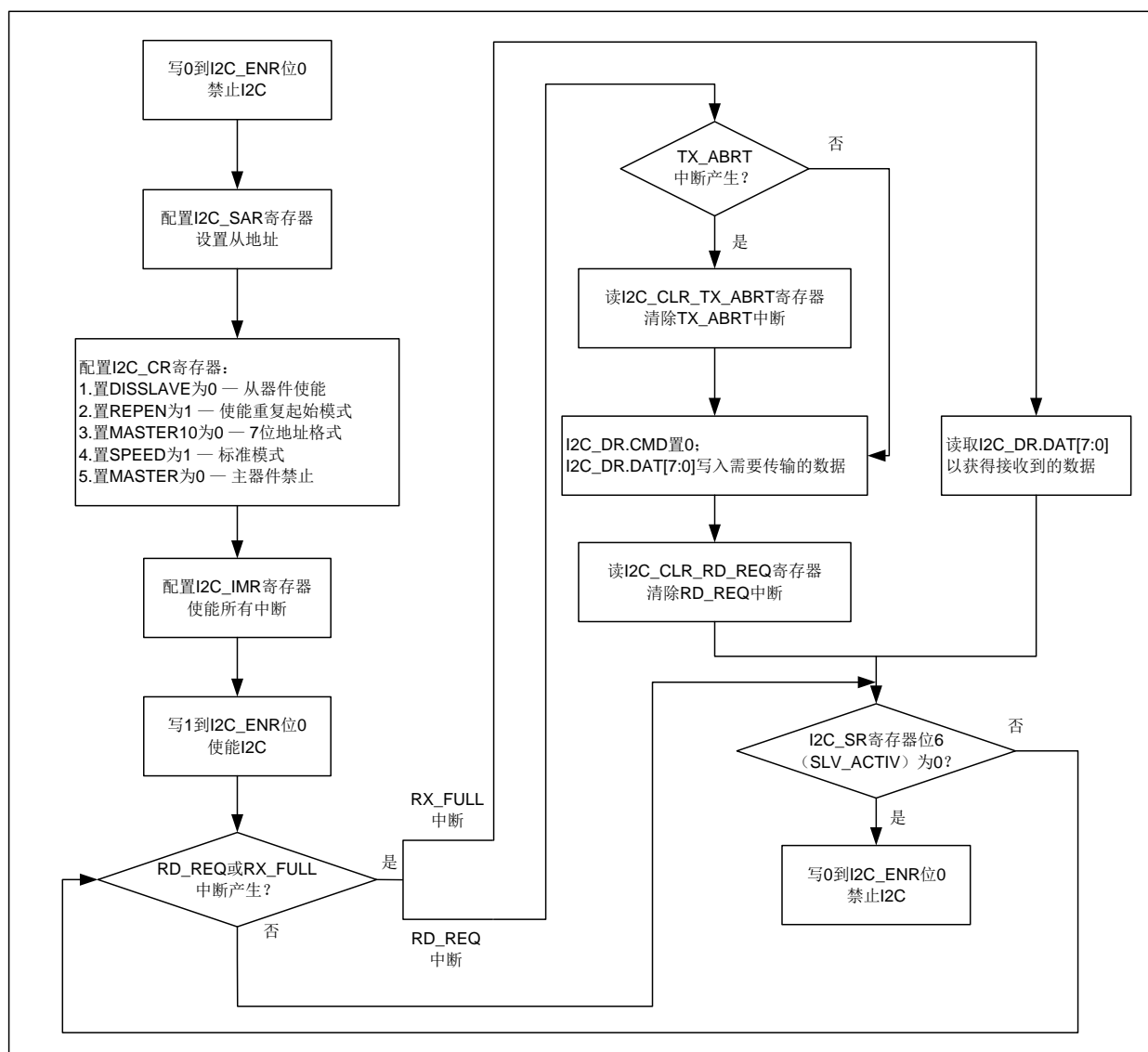


图 28-18 流程图 (I2C 接口作为从器件)

● 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。

28.3.4.2 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.DISSLAVE (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：

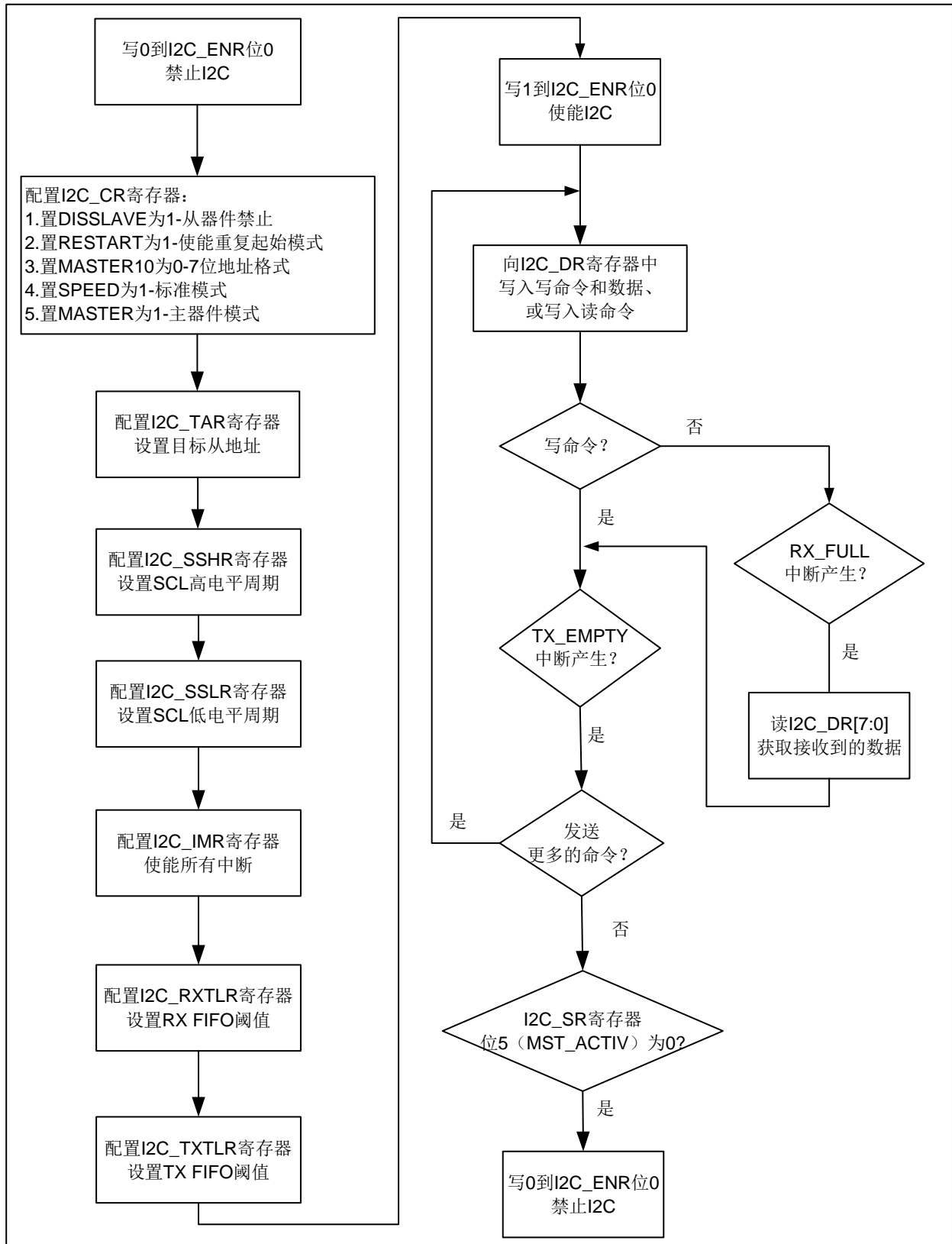


图 28-19 流程图（I2C 接口作为主器件）

28.3.4.3 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

28.3.5 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 28-3 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。

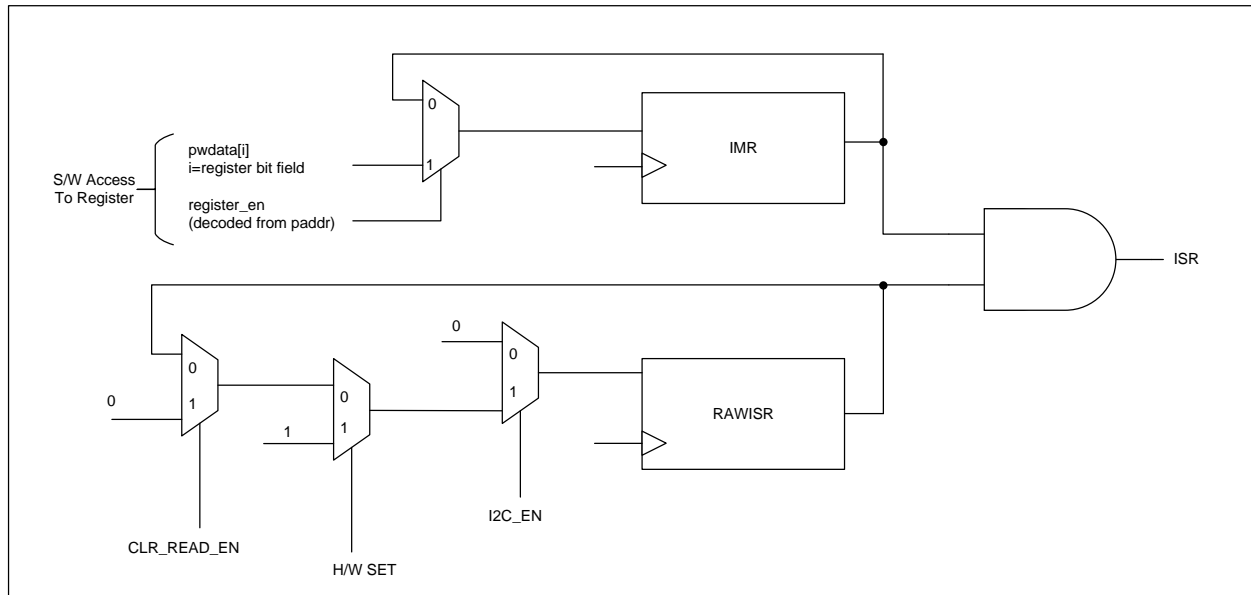


图 28-20 I2C 中断机制

28.3.6 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。

28.3.6.1 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。

28.3.6.2 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。

28.4 寄存器

28.4.1 寄存器总览

表 28-4 I2C 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	I2C_CR	控制寄存器	0x0000007F

Offset	Acronym	Register Name	Reset
0x04	I2C_TAR	目标地址寄存器	0x00000055
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

28.4.2 I2C_CR 控制寄存器

偏移地址：0x00

复位值：0x0000 007F

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			PAD_SE L	SLV_TX _ABRT_ DIS	RESTA RT	STOP	EMPINT	STOPIN T	DISSLA VE	REPEN	MASTE R10	SLAVE1 0	SPEED		MASTE R
			rw	rw	rw	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	SLV_TX_ABRT_DIS	I2C 作为从发送器时 0: 接收到 RD_REQ 信号后清除 TX FIFO 1: 接收到 RD_REQ 信号后禁止清除 TX FIFO
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。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: 发送模式：如果 TX FIFO 为空，则在当前字节后产生 STOP。如果 TX FIFO 不为空，则在完成剩余传输后产生 STOP 接收模式：在当前字节后产生 STOP。如果 TX FIFO 不为空，则主器件立即发起一个新的传输并进行总线仲裁
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

Bit	Field	Description
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 1: 标准模式 (最大 100Kbps) 2: 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 28-5 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

28.4.3 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。

28.4.4 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。

28.4.5 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.								CMD		DAT					
								w		rw					

Bit	Field	Description
31: 9	Reserved	保留, 必须保持复位值
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

28.4.6 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。

28.4.7 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

Bit	Field	Description
15: 0	CNT	标准模式下 SCL 时钟低电平周期，最小值为 8。

28.4.8 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。

28.4.9 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。

28.4.10 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_DO NE	TX_ABR T	RD_RE Q	TX_EM PTY	TX_OV ER	RX_FUL L	RX_OV ER	RX_UN DER
				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

28.4.11 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_DO NE	TX_ABR T	RD_RE Q	TX_EM PTY	TX_OV ER	RX_FUL L	RX_OV ER	RX_UN DER
				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

28.4.12 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_DO NE	TX_ABR T	RD_RE Q	TX_EM PTY	TX_OV ER	RX_FUL L	RX_OV ER	RX_UN DER
				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

Bit	Field	Description
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。

Bit	Field	Description
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。

28.4.13 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。

28.4.14 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。

28.4.15 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。

28.4.16 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])

28.4.17 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])

28.4.18 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])

28.4.19 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])

28.4.20 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
															r

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。

28.4.21 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_DO NE
															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]）

28.4.22 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。

28.4.23 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]）

28.4.24 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

Bit	Field	Description
0	START	读该寄存器清零 START 中断 (I2C_RAWISR[10])

28.4.25 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])

28.4.26 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	ENAB LE
														rw	rw

Bit	Field	Description
31: 2	Reserved	保留, 必须保持复位值
1	ABORT	<p>I2C 传输中止 (I2C transfer abort)</p> <p>0: 中止没有发生或者已经结束</p> <p>1: 中止操作正在进行</p> <p>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。</p>

Bit	Field	Description
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

28.4.27 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_AC TIV	MST_A CTIV	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
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: 接收缓冲未空 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	发送缓冲未空 0: 发送缓冲满 1: 发送缓冲未空
0	ACTIV	I2C 活动状态 MST_ACTIV 位与 SLV_ACTIV 位相或的结果。

28.4.28 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)

28.4.29 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	
														r	

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)

28.4.30 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。

28.4.31 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

28.4.32 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

Bit	Field	Description
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。

28.4.33 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)

28.4.34 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

28.4.35 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

Preliminary

29 USB_FS 全速 USB 控制器

29.1 简介

USB_FS 控制器支持全速设备控制器，支持 12Mb/s 的全速传输，且内置 USB 全速 PHY。

表 29-1 术语表

英文缩写	说明
FS	全速
USB	通用串行总线
PHY	物理层
BDT	缓冲区描述表
BD	缓冲区描述符
SIE	串行接口引擎

29.2 主要特征

- 全速设备控制器兼容 USB 1.1 和 2.0 规范
- 8 个双向端点 (Endpoint)
- FIFO 数据流接口
- DMA 数据传输，无需 CPU 参与完成 USB FIFO 和系统存储器的之间的数据交互
- FIFO 满、FIFO 空、读写访问结束时可发起 DMA 请求

29.3 功能描述

29.3.1 功能框图

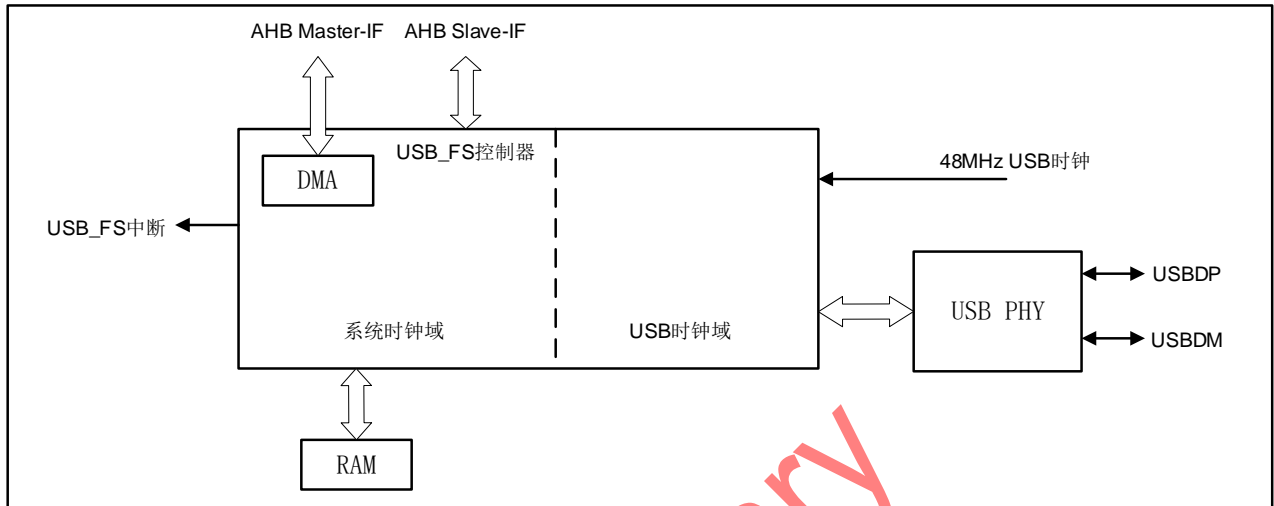


图 29-1 USB_FS 功能框图

29.3.2 信号描述

表 29-2 USB 信号描述表

信号名	类型	描述
USBDP	输入输出	差分数据 D+ 信号
USBDM	输入输出	差分数据 D- 信号

29.3.3 USB_FS 典型硬件配置

下图为设备模式硬件连接示意图。

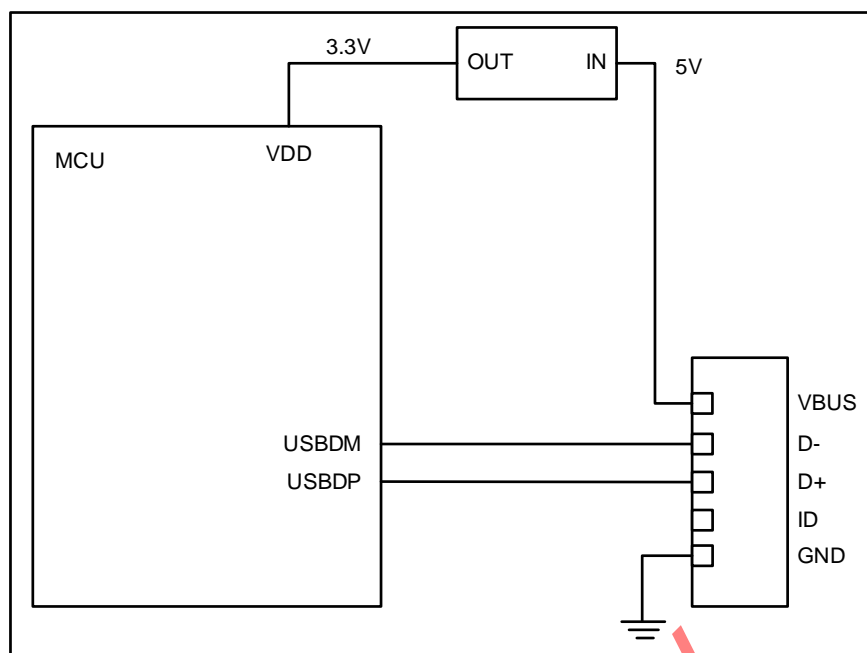


图 29-2 仅从设备硬件连接示意图

29.3.4 USB_FS 软件编程接口

29.3.4.1 USB_FS 数据传输方向

Rx（接收）表示将数据从 USB 移动到内部存储方向，Tx（发送）表示将数据从内部存储移动到 USB 方向。USB 令牌中数据方向是相对于主机而言，所以主机的 IN 令牌和设备的 OUT/SETUP 令牌数据方向为 Rx，主机的 OUT/SETUP 令牌或设备的 IN 令牌的数据方向为 Tx。

29.3.4.2 缓冲区描述表（Buffer Descriptor Table）

为了有效地管理 USB 端点通信，USB_FS 在系统存储器中使用缓冲区描述表（BDT）。BDT 起始地址位于系统内存的 512 字节的边界上，并由 BDTPAGE 寄存器确定地址。每个端点方向需要两个八字节的缓冲区来存放描述符条目。每个端点方向由两个缓冲区描述符条目（Buffer Descriptor Entries）组成（EVEN BD 和 ODD BD）。从而实现双缓冲 BD，提高 USB 传输数据的吞吐量。

微处理器和 USB_FS 都可以访问缓冲区，所以使用一个简单的信号量机制来区分谁可以更新系统内存中的 BDT 和缓冲区。当 BD 由微处理器“拥有”时，信号位 OWN 为‘0’，此时微处理器允许对 BD 条目和系统存储器中的缓冲区进行读写访问。当 OWN 为‘1’时，BD 条目和系统存储器中的缓冲区由 USB_FS 控制，此时 USB_FS 具有完全的读写访问权，微处理器不应修改 BD 或其相应的数据缓冲区。

BD 还包含间接地址指针，指向实际缓冲区驻留在系统内存中的位置。这个间接寻址机制如下图所示。

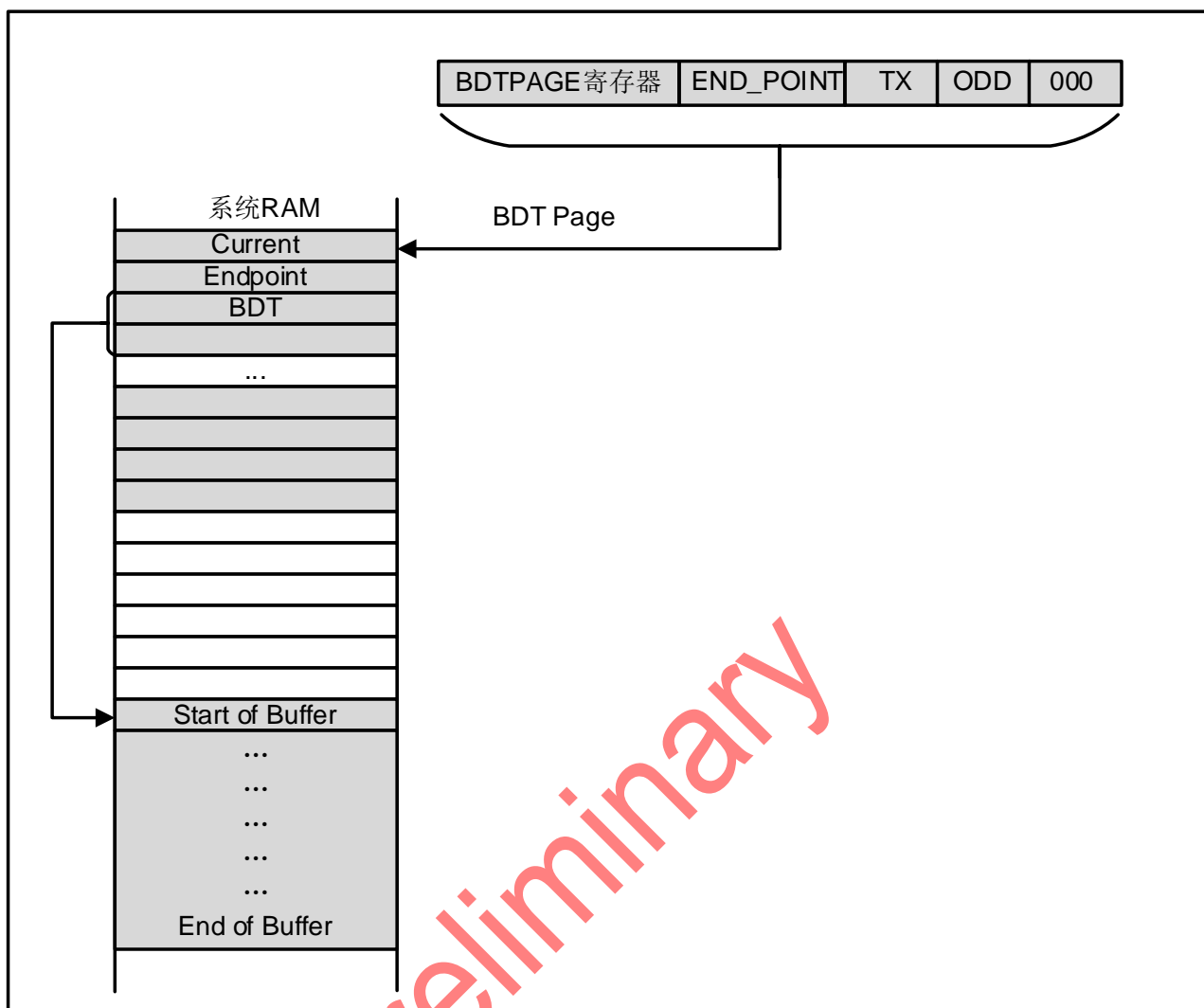


图 29-3 缓冲区描述表

29.3.4.3 缓冲区描述表寻址机制

要通过 USB_FS 或微处理器访问端点数据，必须理解缓冲区描述符表的寻址机制。如前所述，每个 USB 端点方向需要 16 个字节。BDT 页面寄存器指向 BDT 的起始位置。BDT 起始地址位于系统内存的 512 字节的边界上。所有启用的 TX 和 RX 端点的 BD 条目都被索引到 BDT 中，以便通过 USB_FS 或微处理器轻松访问。

当 USB_FS 在启用的端点上接收到 USB 令牌时，它将使用其集成的内置 DMA 控制器来询问 BDT。USB_FS 必须读取相应的端点 BD 条目并确定它是否拥有系统存储器中的 BD 和相应的缓冲区的访问权限。

为计算 BDT 的入口地址，BDTPAGE 寄存器与当前端点以及 TX 和 ODD 字段连接在一起形成一个 32 位地址。该地址机制如下表所示：

表 29-3 BDT 地址计算

bit 31: 24	bit 23: 16	bit 15: 9	bit 8: 5	bit 4	bit 3	bit 2: 0
BDTPAGE3	BDTPAGE2	BDTPAGE 1[7: 1]	END_POINT	IN	ODD	Res.

表 29-4 BDT 地址计算域说明

Field	说明
BDTPAGE	BDTPAGE 寄存器
END_POINT	USB 令牌的中的 Endpoint 域
TX	'1' 为 TX 发送传输, '0'为 RX 接收传输
ODD	对应于当前正在使用的缓冲区, 有 USB_FS SIE 以 ping-pong 方式维护。

- 缓冲区描述符 (BD) 格式

缓冲区描述符 (BD) 为 USB_FS 和微处理器提供端点缓冲区控制信息。基于谁在内存中读取 BD, 缓冲区描述符具有不同的含义。

USB_FS 控制器使用存储在 BD 中的数据来确定:

- 谁拥有系统内存中的缓冲区
- DATA0 或 DATA1 (PID)
- 在数据包完成时释放自己
- 无地址递增 (FIFO 模式)
- 启用数据切换同步机制 (Data toggle synchronization)
- 传输或接收的数据量
- 缓冲区驻留在系统内存中的地址

微处理器使用存储在 BD 中的数据来确定:

- 谁拥有系统内存中的缓冲区
- DATA0 或 DATA1 (PID)
- 收到的令牌 PID
- 传输或接收的数据量
- 缓冲区驻留在系统内存中的地址

BD 的具体格式如下表所示:

表 29-5 BD 格式

31: 26	25: 16	15: 8	7	6	5	4	3	2	1	0
Res.	BC	Res.	OWN	DATA0/1	KEEP/ TOK_PID[3]	NINC/ TOK_PID[2]	DTS/ TOK_PID[1]	BDT_STALL/ TOK_PID[0]	0	0
Buffer Address (32-Bits)										

Field	说明
BC	字节计数位。USB_FS SIE 将在完成指定字节的 RX 传输后更改此字段。
OWN	如果 OWN = 1, USB_FS 可以独占访问 BD。如果 OWN = 0, 则微处理器可以独占访问 BD。这个 OWN 位决定了谁当前拥有缓冲区。 SIE 在完成接收一个令牌时一般将 '0' 写入该位, 除非 KEEP = 1。 当 OWN = 0 时, USB_FS 忽略 BD 中的所有其他字段, 微处理器可以访问整个 BD。BD 的这个字节应该始终是微处理器在初始化 BD 时更新的最后一个字节。 一旦 BD 由 USB_FS 拥有, 微处理器不应该以任何方式改变它。
DATA0/1	发送或接收 DATA0 或 DATA1 包。它不由 USB_FS 改变。
KEEP	如果 KEEP 等于 1, 一旦 OWN 位被置 '1', BD 将永远由 USB_FS 持有。KEEP 必须等于 '0' 才能允许 USB_FS 在令牌处理完成后释放 BD。ISO 端点正在填充一个 FIFO 时需设置此位。微处理器不会被告知令牌已被处理, 数据被简单地传送或接收到 FIFO。当 KEEP = 1 时, NINC 位通常也被置 '1', 以防止地址递增。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 3 会被 USB_FS 写回到 BD 中。
NINC	NINC 位禁用 DMA 地址递增。这将强制 DMA 从相同的地址读取或写入数据。当数据需要从单一位置 (如 FIFO) 读取或写入单个地址时, 这对端点很有用。通常情况下, 该位与 KEEP 位一同设置, 对于连接到 FIFO 的 ISO 端点。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 2 会被 USB_FS 写回 BD。
DTS	设置该位将使 USB_FS 执行数据切换同步机制。当该位为 0 时, 不执行数据切换同步。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 1 会被 USB_FS 写回 BD。
BDT_STALL	如果 SIE 使用此地址的 BDT 接收令牌, 则设置此位将导致 USB_FS 发出 STALL 握手。当设置 BDT_STALL 位时, SIE 不改变 BDT (OWN 保持, BDT 的其余部分保持不变)。
TOK_PID[3: 0]	当传输完成时, 当前令牌 PID 由 USB_FS 写回 BD。写回的值是来自 USB 规范的令牌 PID。具体请参考 USB 规范。
ADDR [31: 0]	地址位表示系统存储器中的 32 位缓冲区地址。由用户设置。

29.3.4.4 USB 传输

当 USB_FS 发送或接收数据时, 首先计算 BDT 地址。一旦 BDT 被读取, 并且 OWN 位为 '1', SIE 就会将数据通过 DMA 的方式发送或接收到 BD 寄存器 ADDR 所指向的缓冲区。当令牌包传输完成时, 如果 KEEP 为 '0', 则 USB_FS 将更新 BDT 并将 OWN 位更改为 '0'。STAT 寄存器更新并产生 TOK_DNE 中断。当微处理器处理 TOK_DNE 中断时, 首先读取 STAT 寄存器, 这给微处理器提供处理端点所需的全部信息。此时, 微处理器需要分配一个新的 BD, 以便可以为该端点发送或接收额外的 USB 数据, 然后处理最后一个 BD。下图显示了一个典型的 USB 令牌如何处理的时间线。

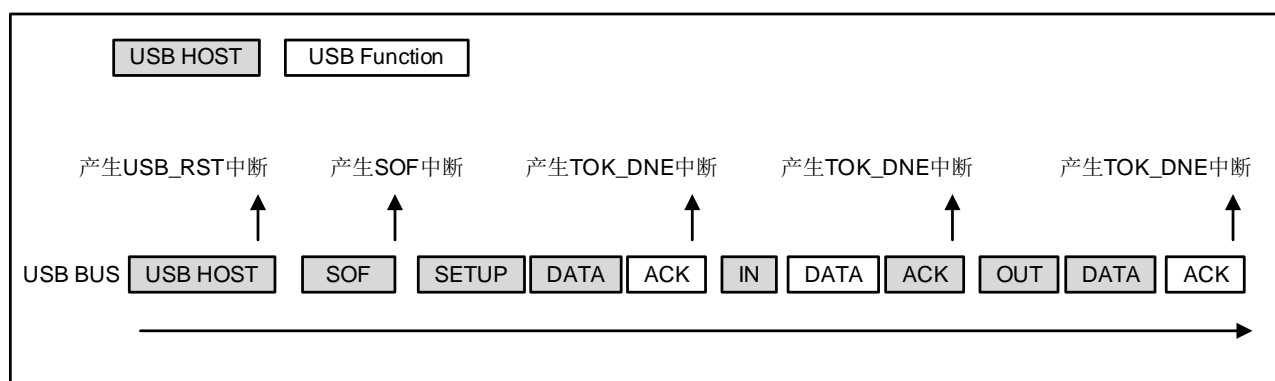


图 29-4 USB 令牌传输示意图

硬件性能问题如内部存储器延迟过大导致 FIFO 溢出，或者软件错误如接收到的数据包超过 MaxPacket size 设定，将会导致 DMA 上溢错误。

由于硬件性能问题引起的 DMA 上溢错误，USB 将根据传输种类不同响应 NAK 或 BTO。在设备模式下，ERR_STAT 寄存器的 DMA_ERR 位都将被置 '1'。通过设置 INTENB 和 ERRENB 寄存器，允许产生 DMA 错误中断。且在设备模式 BDT 不会被写回，也不会触发 TOK_DNE 中断。

为了保护内存中的数据，USB 对超过 MaxPacket size 的数据包会将数据裁剪到 MaxPacket size，存入内部存储器，对于非同步传输返回 ACK。ERR_STAT 寄存器的 DMA_ERR 位将被置 '1'。通过寄存器设定中断许可，允许产生 DMA 错误中断，并且 TOK_DNE 中断也将会被触发。回写到 BDT 的数据包长度将是 MaxPacket size 值，用于表示实际写入内存的数据长度。软件可以取消传输，禁止断点等操作。

29.3.5 中断

USB_FS 控制器全局中断支持以下几种中断源，发生任何一个，都可产生 USB_FS 全局中断。

- STALL 中断
- RESUME 中断
- SLEEP 中断
- TOK_DNE 中断
- ERROR 中断
- USB_RST 中断

29.4 寄存器

29.4.1 寄存器总览

表 29-6 USB 寄存器概览

Offset	Acronym	Register Name	Reset
0x80	USB_FS_INTSTAT	Interrupt Status Register	0x0001
0x84	USB_FS_INTENB	Interrupt Enable Register	0x0000
0x88	USB_FS_ERRSTAT	Error Interrupt Status Register	0x0000
0x8C	USB_FS_ERRENB	Error Interrupt Enable Register	0x0000
0x90	USB_FS_STAT	Status Register	0x0000
0x94	USB_FS_CTL	Control Register	0x0040
0x98	USB_FS_ADDR	Address Register	0x0000
0x9C	USB_FS_BDTPAGE1	BDT Page Register 1	0x0000
0xA0	USB_FS_FRMNUML	Frame Number Register	0x0000
0xA4	USB_FS_FRMNUMH	Frame Number Register	0x0000
0xB0	USB_FS_BDTPAGE2	BDT Page Register 2	0x0000
0xB4	USB_FS_BDTPAGE3	BDT Page Register 3	0x0000
0xC0	USB_FS_EPCTL0	Endpoint Control Register 0	0x0000
0xC4	USB_FS_EPCTL1	Endpoint Control Register 1	0x0000
0xC8	USB_FS_EPCTL2	Endpoint Control Register 2	0x0000
0xCC	USB_FS_EPCTL3	Endpoint Control Register	0x0000
0xD0	USB_FS_EPCTL4	Endpoint Control Register 4	0x0000
0xD4	USB_FS_EPCTL5	Endpoint Control Register 5	0x0000
0xD8	USB_FS_EPCTL6	Endpoint Control Register 6	0x0000
0xDC	USB_FS_EPCTL7	Endpoint Control Register 7	0x0000
0x100	USB_CTRL	USB Control register	0x0000

29.4.2 USB_FS_INTSTAT 中断状态寄存器

偏移地址：0x80

复位值：0x0000

中断状态寄存器（INTSTAT）包含 USB_FS 中每个中断源的位。这些位中的每一位都使用各自的中断使能位进行控制（参见中断使能寄存器）。寄存器的所有位逻辑“或”在一起形成微处理器的单个中断源。中断位置位时，可以通过写 1 清零。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								STALL	Res.	RESUM E	SLEEP	TOK_DN E	SOF_TO K	ERROR	USB_RS T
								rw		rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	STALL	该位在 SIE 发送 STALL 握手包时置位。
6	Reserved	保留，必须保持复位值。
5	RESUME	该位用于发送 Remote Wakeup 信号。 不处于挂起状态时，需禁止该中断。
4	SLEEP	如果 USB_FS 在 USB 总线信号上检测到 3ms 空闲，则该位置'1'。
3	TOK_DNE	当正在处理的令牌完成时，该位被置位。微处理器应立即读取 STAT 寄存器以确定相关的端点和 BD。 将该位清零会导致 STAT 寄存器被清零或 STAT 保持寄存器被加载到 STAT 寄存器。
2	SOF_TOK	如果 USB_FS 收到 SOF 令牌，则此位置位。
1	ERROR	当 ERRSTAT 寄存器中的任何错误条件发生时，该位置位。微处理器必须读取 ERRSTAT 寄存器来确定错误的来源
0	USB_RST	当 USB_FS 解码有效的 USB 复位时，该位置位。这将通知微处理器将 0x00 写入地址寄存器并使能端点 0。一旦检测到 USB 复位 2.5 微秒，USB_RST 将被置位。在 USB 复位条件被移除之前，它不会被再次置位。

29.4.3 USB_FS_INTENB 中使能态寄存器

偏移地址：0x84

复位值：0x0000

设置这些位中的任何一位将使能 INTSTAT 寄存器中的相应中断源。复位后该寄存器值为 0x00 即禁止所有中断。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								STALL	Res.	RESUME	SLEEP	TOK_DNE	SOF_TOK	ERROR	USB_RST
								r/w		r/w	r/w	r/w	r/w	r/w	r/w

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	STALL	设置该位为 1 使能 STALL 中断。
6	Reserved	保留，必须保持复位值。
5	RESUME	设置该位为 1 使能 RESUME 中断。
4	SLEEP	设置该位为 1 使能 SLEEP 中断。
3	TOK_DNE	设置该位为 1 使能 TOK_DNE 中断。
2	SOF_TOK	设置该位为 1 使能 SOF_TOK 中断。
1	ERROR	设置该位为 1 使能 ERROR 中断。
0	USB_RST	设置该位为 1 使能 USB_RST 中断。

29.4.4 USB_FS_ERRSTAT 错误中断状态寄存器

偏移地址：0x88

复位值：0x0000

这些位中的每一位都使用各自的错误使能位进行控制（请参见错误使能寄存器页面）。结果进行“或”运算并发送到 INTSTAT 寄存器的 ERROR 位。通过写 1 清除对应的状态位。一旦检测到错误条件，对应的状态位就会置位。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BTS_ERR	Res.	DMA_ERR	BTO_ERR	DFN8	CRC16	CRC5/EOF	PID_ERR
								rw		rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	BTS_ERR	检测到位填充错误时置位。相应的数据包将被拒绝。
6	Reserved	保留，必须保持复位值。
5	DMA_ERR	如果 USB_FS 请求 DMA 访问来读取新的 BDT，但在 USB_FS 需要接收或发送数据之前没有给出总线，则该位置‘1’，如果处理 TX 传输，将导致发送数据下溢，或者如果处理一个 Rx 传输，这会引发接收数据溢出。 如果发向主机的数据包大于在 BDT 中分配的缓冲区大小，则该位也被置位。在这种情况下，数据包在被放入缓冲存储器时被截断。
4	BTO_ERR	如果发生总线周转超时错误，则该位置位。此 USB_FS 使用总线周转定时器来跟踪 SETUP 或 OUT 的令牌和数据阶段或 IN TOKEN 的数据和握手阶段之间经过的时间。如果在 IDLE 转换之前，从先前的 EOP 中计数，多于 16 位的时间，则会出现总线周转超时错误。
3	DFN8	收到的数据字段不是 8 位。如果数据字段不是整数字节，则该位将被置位。
2	CRC16	CRC16 校验失败。如果数据包由于 CRC16 错误而被拒绝，则该位置位。
1	CRC5/EOF	该中断将检测主机生成的令牌数据包中的 CRC5 错误。如果令牌数据包由于 CRC5 错误而被拒绝，则该位置位。
0	PID_ERR	PID 自校验错误。

29.4.5 USB_FS_ERRENB 错误中断使能寄存器

偏移地址：0x8C

复位值：0x0000

错误中断使能寄存器包含 USB_FS 中每个错误中断源的使能位。设置任何这些位将使能 ERR_ISTAT 寄存器中的相应错误中断源。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BTS_ERR	Res.	DMA_ERR	BTO_ERR	DFN8	CRC16	CRC5/EOF	PID_ERR
								rw		rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	BTS_ERR	设置该位为 1 使能 BTS_ERR 中断。
6	Reserved	保留，必须保持复位值。
5	DMA_ERR	设置该位为 1 使能 DMA_ERR 中断。
4	BTO_ERR	设置该位为 1 使能 BTO_ERR 中断。
3	DFN8	设置该位为 1 使能 DFN8 中断。
2	CRC16	设置该位为 1 使能 CRC16 中断。
1	CRC5/EOF	设置该位为 1 使能 CRC5/EOF 中断。
0	PID_ERR	设置该位为 1 使能 PID_ERR 中断。

29.4.6 USB_FS_STAT 状态寄存器

偏移地址：0x90

复位值：0x0000

当微处理器收到 TOK_DNE 中断时，应读取状态寄存器以确定先前端点通信的状态。当 TOK_DNE 中断位置位时，状态寄存器中的数据有效。

STAT 寄存器实际上是 USB_FS 维护的状态 FIFO 的读取窗口。当 USB_FS 使用 BD 时，它会更新状态寄存器。如果在 TOK_DNE 中断服务之前执行了另一个 USB 事务，则 USB_FS 将把下一个事务的状态存储在 STAT FIFO 中。

因此，STAT 寄存器实际上是一个四字节 FIFO，它允许微处理器在 SIE 处理下一个事务时处理一个事务。

清零 INTSTAT 寄存器中的 TOK_DNE 位会导致 SIE 使用下一个 STAT 值的内容更新 STAT 寄存器。如果 STAT 保持寄存器中的数据有效，则 SIE 将立即重新置位 TOK_DNE 中断。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								ENDP			TX	ODD	Res.		
								r			r	r			

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 4	ENDP	表示前一个令牌的端点地址。这允许微处理器确定哪个 BDT 条目由最后的 USB 事务更新。
3	TX	该位指示上次更新的 BDT 的传输方向。 TX = 1 表示发送数据传输。 TX = 0 表示接收数据传输。
2	ODD	该位表示更新的最后一个缓冲区描述符位于 BDT 的奇数组中，请参阅前面的部分以获取有关 BDT 地址生成的更多信息。
1: 0	Reserved	保留，必须保持复位值。

29.4.7 USB_FS_CTL 控制寄存器

偏移地址：0x94

复位值：0x0000

控制寄存器为 USB_FS 提供各种控制和配置信息。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.								JSTATE	SE0	TxdSuspendTokenBusy	Res.	Res.	RESUME	ODD_RST	USB_EN		
								r	r	rw					rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	JSTATE	USB 差分接收器接收到 JSTATE 信号。
6	SE0	USB 接收到 SE0 信号。
5	TxdSuspendTokenBusy	该位用于通知处理器 SIE 已禁用数据包发送和接收。清零该位允许 SIE 继续令牌处理。当接收到 SETUP 令牌包时，该位由 SIE 置位，允许软件在恢复令牌处理之前使 BDT 中的任何挂起的数据包事务出队。
4: 3	Reserved	保留，必须保持复位值。
2	RESUME	设置该位将允许 USB_FS 执行恢复信号。这将允许 USB_FS 执行远程唤醒。软件必须将 RESUME 设置为 1 达所需的时间，然后将其清零。
1	ODD_RST	设置该位将重置所有的 BDT ODD ping / pong 位为 0，然后指定偶数 BDT。
0	USB_EN	设置该位为 1 将使 USB_FS 工作，清除它将禁用 USB_FS。将该位置'1'将导致 SIE 复位 BDT 的 ODD 位。因此，设置该位会重置 SIE 中的大部分逻辑。

29.4.8 USB_FS_ADDR 地址寄存器

偏移地址：0x98

复位值：0x0000

地址寄存器包含 USB_FS 在设备模式下解码的唯一 USB 地址。

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

Bit	Field	Description
15: 7	Reserved	保留，必须保持复位值。
6: 0	ADDR	USB_FS 在设备模式下解码的 USB 地址。

29.4.9 USB_FS_BDTPAGE1 缓冲区描述符表页寄存器 1

偏移地址：0x9C

复位值：0x0000

用于计算当前缓冲区描述符表（BDT）驻留在系统存储器中的地址。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 1	BDT_BA	该 7 位值提供 BDT 基地址的位 15 至 9，其定义了缓冲器描述符表在系统存储器中的位置。内存中的 32 位 BDT 基地址总是与 512 字节边界对齐。
0	Reserved	保留，必须保持复位值。

29.4.10 USB_FS_FRMNUML 低位帧数寄存器

偏移地址：0xA0

复位值：0x0000

帧编号寄存器包含 11 位帧编号。帧编号寄存器需要两个 8 位寄存器来实现，低位字节包含在 FRMNUML 中，高位字节包含在 FRMNUMH 中。当接收到 SOF TOKEN 时，这些寄存器用当前帧号更新。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 0	FRM	这些位表示 11 位帧数的低 8 位。

29.4.11 USB_FS_FRMNUMH 高位帧数寄存器

偏移地址：0xA4

复位值：0x0000

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

Bit	Field	Description
15: 3	Reserved	保留，必须保持复位值。
2: 0	FRM	这些位表示 11 位帧数的高 3 位。

29.4.12 USB_FS_BDTPAGE2 缓冲区描述符表页寄存器 2

偏移地址：0xB0

复位值：0x0000

用于计算当前缓冲区描述符表（BDT）驻留在系统存储器中的地址。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 0	BDT_BA	该 8 位值提供 BDT 基地址的位 23 至 16，其定义了缓冲器描述符表在系统存储器中的位置。内存中的 32 位 BDT 基地址总是与 512 字节边界对齐。

29.4.13 USB_FS_BDTPAGE3 缓冲区描述符表页寄存器 3

偏移地址：0xB4

复位值：0x0000

用于计算当前缓冲区描述符表（BDT）驻留在系统存储器中的地址。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 0	BDT_BA	该 8 位值提供 BDT 基地址的位 31 至 24，其定义了缓冲器描述符表在系统存储器中的位置。内存中的 32 位 BDT 基地址总是与 512 字节边界对齐。

29.4.14 USB_FS_EPCTL0~7 端点控制寄存器 0~7

偏移地址：0xC0~0xDC

复位值：0x0000

在收到 USB_RST 中断后，微处理器应将 ENDPT0 设置为 0x0D。

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

Res.	EP_CTL_DIS	EP_RX_EN	EP_TX_EN	EP_STALL	EP_HSHK
	rw	rw	rw	rw	rw

Bit	Field	Description
15: 5	Reserved	保留，必须保持复位值。
4: 2	EP_CTL_DIS EP_RX_EN EP_TX_EN	这三位定义了端点是否被使能以及端点的方向，具体如下： EP_CTL_DIS EP_RX_EN EP_TX_EN : 使能/方向控制 X X 0 0 : 端点被禁用 X 0 1 : TX X 1 0 : RX 0 1 1 : 控制传输，TX 和 RX 1 1 1 : TX 和 RX
1	EP_STALL	优先于 EP_HSHK 位的设定，且仅在端点没有被禁用时有效。 0: 不返回 STALL 握手信号 1: 返回 STALL 握手信号
0	EP_HSHK	0: 该端点不返回握手信号 1: 该端点返回握手信号

29.4.15 USB_CTRL USB 控制寄存器

偏移地址: 0x100

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.								SUSPE	Res.						VBUSORE	VBUSORV
								rw							rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	SUSPE	片内 USB PHY 挂起状态使能位，用于 USB 控制器处于挂起状态时，设置此位可以使片内 PHY 处于挂起状态，进一步节省功耗。清除该位使片内 USB PHY 处于正常工作状态。SUSPE=1 是，允许 Resume 操作唤醒系统 STOP 模式。 0: 片内 USB PHY 处于正常工作状态 1: 片内 USB PHY 处于挂起状态，允许 Resume 或 Remote Wakeup 操作唤醒系统 STOP 模式
6: 2	Reserved	保留，必须保持复位值。

30 FlexCAN 控制器局域网

30.1 简介

FlexCAN 模块是一个通信控制器，遵循 ISO 11898-1 标准、CAN FD 和 CAN 2.0B 协议规范。CAN 协议主要被设计用作车载串行总线，在以下方面满足规范要求：

- 实时处理
- 车辆在电磁干扰环境下的可靠操作
- 成本效益
- 带宽要求

该模块支持标准和扩展帧，支持最大 64 字节有效负载，传输速率高达 8Mbps，并且具有非常灵活的用于传输和接收的邮箱系统。邮箱系统由 32 个报文缓冲区（MB）组成。报文缓冲区用于存储配置、控制数据、时间戳、报文 ID 以及数据（参见章节“报文缓冲区结构”）。和 MB 相对应的内存空间可以通过配置支持传统型 Rx FIFO 接收机制，该机制能够结合 ID 过滤表（高达 104 个扩展 ID 或 208 个标准 ID 或 416 个 8 位 ID）检测接收帧，并为高达 32 个 ID 表元素提供了私有掩码寄存器。

FlexCAN 支持 FIFO（经典 CAN 帧使用传统型 Rx FIFO；CAN FD 帧使用增强型 Rx FIFO）和邮箱同时接收。对于邮箱接收，通过匹配算法可以将接收到的帧只存储到具有相同 ID 的 MB 中。掩码机制能够将 MB 中设置的 ID 与一系列接收帧的 ID 进行匹配。传输时，仲裁算法基于报文的 ID（提供本地优先级可选）或 MB 的编号来决定待传输 MB 的优先级。

30.2 主要特征

- 完全支持 CAN FD 和 CAN 2.0B 协议：
 - a) 标准帧
 - b) 扩展帧
 - c) 0 ~ 64 字节数据长度
 - d) 可编程比特率，最高 8Mbps
 - e) 基于内容的寻址方式
- 符合 ISO 11898-1 标准
- 每个邮箱都可配置为接收或发送，均支持标准和扩展报文
- 每个邮箱都有私有接收掩码寄存器
- 全功能的传统型和增强型 Rx FIFO，存储容量各多达 6 帧，自动进行内部指针处理且支持 DMA
- 传输中止功能
- 支持只听模式

- 可编程的回环模式，支持自测试
- 可编程的传输优先级机制：最低 ID、最小邮箱编号或最高优先级
- 基于 16 位自由运行计时器的时间戳机制
- 全局网络时间（由特定的报文进行同步）
- 中断掩码
- 高优先级仲裁机制，延迟时间短
- 远程帧可以由软件处理或自动处理
- Tx 邮箱状态（最低优先级缓冲区或空缓冲区）
- ID 接收过滤命中指示（IDHIT）寄存器
- 发送报文的 CRC 状态
- 传统型 Rx FIFO 全局掩码寄存器
- 匹配过程中邮箱或 Rx FIFO 优先级可选
- 高效的 Rx FIFO ID 过滤功能，传统型 Rx FIFO 支持 104 个 ID 过滤表元素；增强型 Rx FIFO 支持 2 个 ID 过滤元素
- PE 时钟源可编程为外设时钟或振荡器时钟
- 收发器延迟补偿（TDC）功能

30.3 功能描述

30.3.1 功能框图

FlexCAN 的功能框图如下，包括用于存储报文缓冲区（MB）、接收全局掩码寄存器、接收私有掩码寄存器、接收（Rx）FIFO 过滤器以及接收 FIFO 标识符过滤器的内存。

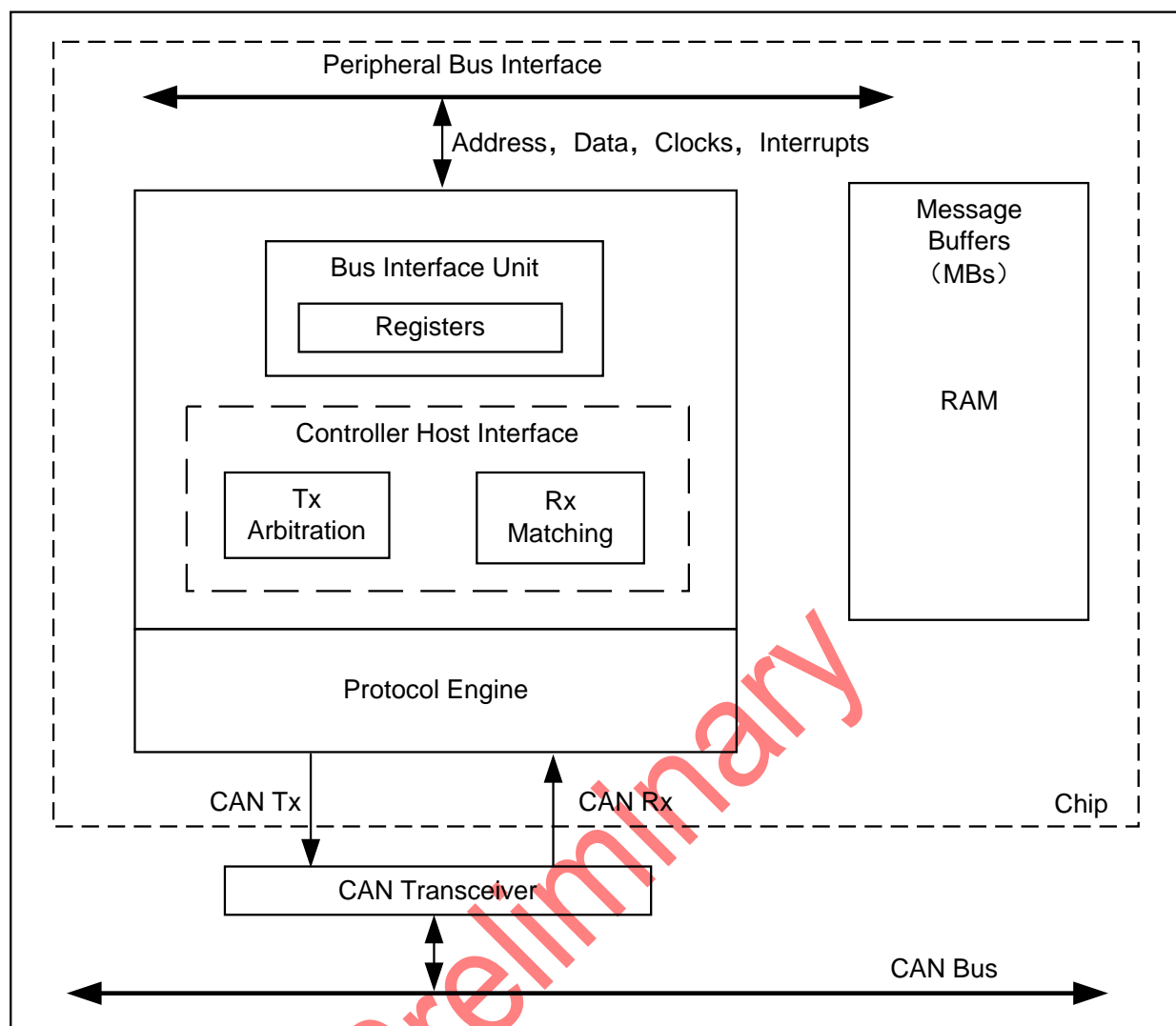


图 30-1 FlexCAN 框图

协议引擎（PE）子模块管理 CAN 总线上的串行通信：

- 请求存取 RAM 接收和传输帧
- 验收接收到的报文
- 进行错误处理
- 检测 CAN FD 报文

控制器主机接口（CHI）子模块负责选择接收和传输的报文缓冲区，以及对报文的仲裁和 ID 匹配算法。

总线接口单元（BIU）子模块控制内部接口总线的访问，建立与 CPU 和其他模块的连接。时钟、地址和数据总线、中断输出、DMA 都通过 BIU 进行访问。

30.3.2 工作模式

FlexCAN 具有以下 5 种功能模式：

- 正常模式
- 冻结模式
- 回环模式
- 只听模式
- CAN FD 活动模式

30.3.2.1 正常模式（用户或管理员模式）

FlexCAN 收发帧，正常处理错误，CAN 协议的功能全部使能。用户模式和管理员模式在访问一些受限的控制寄存器时有所不同，参见表格“寄存器访问和复位信息”。

30.3.2.2 冻结模式

使能冻结模式（置位 CAN_MCR.FRZ）后，可通过以下方式之一请求进入冻结模式：

- 非低功耗模式下，置位 CAN_MCR.HALT
- 非低功耗模式下，芯片进入调试模式

FlexCAN 硬件置位 CAN_MCR.FRZACK 以响应上述请求。当请求和响应的条件都满足时，CPU 认为模块处于冻结模式。该模式下收发帧停止，与 CAN 总线的同步丢失。

请求冻结模式时，FlexCAN 执行以下操作：

- 等待模块处于间歇（Intermission）、被动错误、总线关闭或空闲状态
- 等待所有内部活动（仲裁、匹配、移入和移出等）完成，待处理的移入不会阻止模块进入冻结模式
- 忽略 Rx 引脚的输入，驱动 Tx 引脚为隐性
- 停止预分频器，由此中止所有 CAN 协议活动
- 允许对错误计数寄存器执行写操作（该寄存器在其他模式下为只读）
- 置位 CAN_MCR.NOTRDY 和 CAN_MCR.FRZACK

请求冻结模式后，用户在执行任何其他操作前，必须等待 CAN_MCR.FRZACK 被置位。否则，FlexCAN 的动作无法预测。冻结模式下，所有内存映射的寄存器都可以被访问（CAN_CTRL1.CLKSRC 除外，该位为只读）。

退出冻结模式的方式有以下两种：

- CPU 清零 CAN_MCR.FRZ
- 芯片退出调试模式和/或 CAN_MCR.HALT 被清零

当 PE 识别到冻结模式请求无效后，CAN_MCR.FRZACK 被硬件清零。退出冻结模式时，FlexCAN 尝试通过等待 11 个连续隐性位与 CAN 总线重新同步。

30.3.2.3 回环模式

CAN_CTRL1.LPB 被置位时，FlexCAN 将进入回环模式。该模式下，Rx 引脚的输入被忽略，Tx 引脚的输出处于隐性状态（逻辑 1）。发送器的输出在内部反馈给接收器的输入，可用于模块自测试。发送报文时，FlexCAN 将自身发送的报文视为从远程节点接收到的报文。为保证能正确接收到自身发送的报文，FlexCAN 会忽略 CAN 帧应答场 ACK 槽（ACK SLOT）发送的位。发送和接收中断均会产生。

30.3.2.4 只听模式

CTRL1.LOM 被置位时，FlexCAN 将进入只听模式。该模式下，FlexCAN 工作在 CAN 错误被动模式，所有的错误计数器都被冻结。发送被禁止，只能接收被其他 CAN 节点应答的报文。如果 FlexCAN 检测到一个未被应答的报文，则标记为 BIT0 错误（不改变接收错误计数值）。

30.3.2.5 CAN FD 活动模式

当 FlexCAN 处于冻结模式时，可通过设置 CAN_MCR.FDEN 将 FlexCAN 设置为 CAN FD 活动模式。该模式下，FlexCAN 能够收发所有按照 CAN FD 和 CAN 2.0 协议格式化的报文。

下表为 CAN FD 活动模式（CAN_MCR.FDEN = 1）和经典 CAN 模式（CAN_MCR.FDEN = 0）功能上的区别。

表 30-1 经典 CAN 和 CAN FD 模式功能对比

Feature	经典 CAN	CAN FD
传统型 Rx FIFO	支持	不支持
传统型 Rx FIFO DMA	支持	不支持
增强型 Rx FIFO	支持	支持
增强型 Rx FIFO DMA	支持	支持

FlexCAN 具有以下低功耗模式：

- 模块禁止模式

30.3.2.6 模块禁止模式

一种低功耗模式，通常用于临时停止 FlexCAN 模块，不耗电。CPU 置位 CAN_MCR.MDIS 以请求模块禁止，FlexCAN 置位 CAN_MCR.LPMACK 响应请求。当请求和响应的条件都满足时，CPU 认为模块处于禁止模式。

如果在冻结模式下禁止 FlexCAN，则模块请求停止 PE 和 CHI 子模块的时钟，并置位

CAN_MCR.LPMACK 和 CAN_MCR.FRZACK。

如果在传输或接收期间禁止 FlexCAN，则模块执行以下操作：

- 等待模块处于总线关闭或空闲状态，或等待 Intermission 的第三位为隐性
- 等待所有内部活动（仲裁、匹配、移入和移出等）完成，不考虑待处理的移入
- 忽略 Rx 引脚的输入，驱动 Tx 引脚为隐性
- 关闭 PE 和 CHI 子模块的时钟
- 置位 CAN_MCR.NOTRDY 和 CAN_MCR.LPMACK

总线接口单元继续运行，允许 CPU 访问内存映射寄存器（除 RXMGMASK、RX14MASK、RX15MASK 和 RXFGMASK）。模块禁止模式下，不能访问 RAM 内的寄存器 CAN_RXFIR、MBs、CAN_RXIMRn 和预留字。

清零 CAN_MCR.MDIS 可以退出模块禁止模式。FlexCAN 请求恢复时钟，并在 PE 子模块识别到 CPU 取消禁止模式后清零 CAN_MCR.LPMACK。

30.3.3 传输流程

传输 CAN 帧时，CPU 必须为传输准备报文缓冲区，步骤如下：

- 1) 检查相应的中断标志位是否被置位，若被置位则将其清零
- 2) 如果 MB 为激活状态（传输待处理），则向控制和状态（C/S）字的代码（CODE）字段写入中止代码（0b1001）以请求中止传输。通过轮询 CAN_IFLAG1 寄存器，等待相应的 IFLAG 置位；或使能相应的 IMASK 位，通过中断请求来实现。接着，回读 CODE 字段以检查传输是否被中止（参见章节“传输中止机制”），并清零相应的中断标志

注：如果一个报文缓冲区既能参与匹配又能参与仲裁过程，则称该报文缓冲区处于激活状态（ACTIVE）。CODE 字段为 0b0000 的接收 MB 处于非激活状态（参见表格“接收缓冲区代码”）；代码为 0b1000 或 0b1001 的发送 MB 也处于非激活状态（参见表格“发送缓冲区代码”）。

- 3) 写入 ID，包括本地优先级（若 CAN_MCR.LPRIOEN 被置位）
- 4) 写入数据字节
- 5) 根据需要配置 C/S：ID 类型（IDE）、远程传输请求（RTR）、帧格式（EDL）、比特率切换（BRS）、数据长度（DLC），写入代码 0xC 以激活 MB 进行 CAN 帧传输

激活的 MB 将参与仲裁过程，并最终依照优先级进行传输。用于传输的 MB 中存储的 DLC 值大于相应的 MB 负载大小时，FlexCAN 使用 0xCC 添加必要的字节数，以完成预期的 DLC。

传输成功结束时：

- 自由运行计时器的值被写入时间戳字段
- C/S 的 CODE 字段被更新
- CAN_CRC 和 CAN_FDCRC 寄存器被更新
- CAN_IFLAG1 的状态标志置位；若相应的中断使能，则产生中断

传输后的新 CODE 字段取决于用于激活 MB 的代码（参见表格“发送缓冲区代码”）。

中止功能使能（CAN_MCR.AEN 置位）时，邮箱（被配置为发送缓冲区）的中断标志被置位后，CPU 无法更新邮箱。因此，开始为新传输或接收准备该 MB 前，CPU 必须清零相应的 CAN_IFLAG1 位。

30.3.4 仲裁过程

仲裁过程按照邮箱编号由低到高进行扫描，搜索用于下一次传输的邮箱。该邮箱存放了下一次要传输的报文，被称为仲裁获胜邮箱。下列事件会触发仲裁过程：

- CAN 帧的 CRC 字段，起始点取决于 CAN_CTRL2.TASD 字段的值
- CAN 帧的错误界定符（Error Delimiter）字段期间
- CAN 帧的过载界定符（Overload Delimiter）字段期间
- 获胜邮箱非激活，且 CAN 总线尚未到达 Intermission 字段首位
- CPU 写获胜邮箱的 C/S，且 CAN 总线尚未到达 Intermission 字段的首位
- CHI 处于空闲状态，且 CPU 写任意 MB 的 C/S
- FlexCAN 退出总线关闭状态
- 离开冻结模式或低功耗模式

如果在 CAN 总线到达 Intermission 字段的首位前，仲裁过程没能评估所有邮箱，则临时获胜邮箱无效，且 FlexCAN 不会参与下一次的 CAN 总线竞争。仲裁过程在扫描结束时，会根据 CAN_CTRL1.LBUF 和 CAN_MCR.LPRIOEN 的配置，从激活的 Tx 邮箱中选择获胜邮箱。

30.3.4.1 最小邮箱编号优先

CAN_CTRL1.LBUF 被置位时，CAN_MCR.LPRIOEN 位无效，搜索到的第一个（最小编号）激活的 Tx 邮箱为仲裁获胜邮箱。

30.3.4.2 最高优先级邮箱优先

CAN_CTRL1.LBUF 为 0 时，仲裁过程根据最高优先级搜索激活的 Tx 邮箱。当多个外部节点同时竞争总线时，该邮箱的帧赢得 CAN 总线仲裁的几率更高。

用于此次仲裁的位序列被称为邮箱仲裁值。在所有 Tx 邮箱中，优先级最高的 Tx 邮箱仲裁值最小。如果两个或多个邮箱具有相同的仲裁值，则编号最小的邮箱为仲裁获胜邮箱。

仲裁值的构成取决于 CAN_MCR.LPRIOEN 的配置。

- 本地优先级禁止

如果 CAN_MCR.LPRIOEN 为 0，则本地优先级无效，根据 CAN 帧的位传输顺序确立仲裁值。

表 30-2 本地优先级无效时仲裁值的构成

格式	邮箱仲裁值 (32 位)				
标准 (IDE = 0)	标准 ID (11 位)	RTR (1 位)	IDE (1 位)	— (18 位)	— (1 位)
扩展 (IDE = 1)	扩展 ID [28: 18] (11 位)	SRR (1 位)	IDE (1 位)	扩展 ID [17: 0] (18 位)	RTR (1 位)

- 本地优先级使能

如果 CAN_MCR.LPRIOEN 被置位，则本地优先级使能，邮箱的 PRIO 字段纳入仲裁值的最左侧。

表 30-3 本地优先级使能时仲裁值的构成

格式	邮箱仲裁值 (35 位)					
标准 (IDE = 0)	PRIO (3 位)	标准 ID (11 位)	RTR (1 位)	IDE (1 位)	— (18 位)	— (1 位)
扩展 (IDE = 1)	PRIO (3 位)	扩展 ID [28: 18] (11 位)	SRR (1 位)	IDE (1 位)	扩展 ID [17: 0] (18 位)	RTR (1 位)

PRIO 字段为仲裁值的最高位，因此 PRIO 值低的邮箱优先级更高，与剩余仲裁值无关。

注：PRIO 字段不属于 CAN 总线帧，仅影响内部仲裁过程。

30.3.4.3 仲裁完成

找出仲裁获胜邮箱后，其内容会被复制到一个名为 Tx 串行报文缓冲区 (Tx SMB) 的隐藏辅助 MB 中，该 MB 与普通 MB 结构相同，但用户无法访问。此复制操作称为移出 (Move Out)。移出完成后，无法写相应 MB 的 C/S (CAN_MCR.AEN 被置位时)。以下事件可使写操作恢复：

- MB 被传输后，且相应的 IFLAG 位被 CPU 清零
- FlexCAN 进入冻结模式或总线关闭
- FlexCAN 丢失总线仲裁或出现传输错误

当 CAN 总线上出现第一个机会窗口时，Tx SMB 内的报文依据 CAN 协议规则进行传输。

30.3.4.4 仲裁启动和停止条件

仲裁在下列情况下被触发：

- Rx 帧和 Tx 帧期间，从 CAN CRC 字段至帧结束。配置 CAN_CTRL2.TASD 可优化仲裁起点
- CAN 总线关闭状态，从 TX_ERR_CNT 的值为 124 到 128。配置 CAN_CTRL2.TASD 可优化仲裁起点
- 总线空闲状态，CPU 写 C/S。第一个 C/S 写操作启动仲裁过程，同一个仲裁期间的第二个 C/S 写操作会重启流程。如果执行了其他 C/S 写操作，则 Tx 仲裁进程被挂起。如果在仲裁过程结束后没有获胜者，则 Tx 仲裁机启动一个新的仲裁过程。如果存在待处理的仲裁，且总线空闲状态启动，则仲裁过程被触发。此时，总线空闲状态下的第一个和第二个 C/S 写操作不会重启仲裁过程。在等待总线空闲 (Wait For Bus Idle) 状态并且下一个状态为空闲时，可能没有足够的时间完成仲裁。此情况下，扫描不会中断，而是在总线空闲状态下完成。仲裁期间，C/S 写操作

不会导致仲裁重启

- 仲裁获胜邮箱在有效仲裁期间失活
- 退出冻结模式（等待总线空闲状态的首位）。如果在等待总线空闲状态下进行重同步，则仲裁过程会重启

仲裁在下列情况下停止：

- 所有邮箱都被扫描过
- 最低缓冲区（lowest buffer）功能使能时，发现一个激活的 Tx 邮箱
- 仲裁获胜邮箱在任何仲裁期间失活或中止
- 没有足够的时间完成 Tx 仲裁过程。此情况下，仲裁过程被挂起
- 总线错误或过载
- 总线空闲状态下请求低功耗或冻结模式

仲裁在下列情况下被挂起：

- 无法及时完成仲裁过程
- 仲裁期间执行 C/S 写操作，如果执行写操作的 MB 编号小于 Tx 仲裁指针
- 没有进行中的 Tx 仲裁过程时，执行任何 C/S 写操作
- Rx 匹配刚刚更新了 Rx 代码至 Tx 代码
- 进入总线关闭状态

仲裁期间写 C/S 会产生下列影响：

- 如果仲裁获胜邮箱执行 C/S 写操作，则立即重启新流程
- 如果在编号比 Tx 仲裁指针大的 MB 中执行 C/S 写操作，则进行中的仲裁过程将正常扫描该 MB

30.3.5 接收流程

为确保邮箱正常接收帧，CPU 必须执行下列步骤以准备接收邮箱：

1) 如果邮箱（Tx 或 Rx）为激活状态，需要使邮箱失活（参见章节“邮箱失活”），最好选用安全的失活方式（参见章节“传输中止机制”）

2) 写 ID

3) 向 C/S 的代码字段写入空代码（0b0100）以激活邮箱

MB 激活后，能够接收与过滤器匹配的帧。接收成功结束时，通过以下移入流程（参见章节“移入”）更新邮箱：

1) 接收到的数据字段（CAN 格式报文最多 8 字节；CAN FD 格式报文最多 64 字节）被存储

2) 接收到的 ID 段被存储

3) 帧标识符字段第二位时自由运行计时器的值，被写入邮箱的时间戳字段

- 4) 接收到的 SRR、IDE、RTR 和 DLC 字段被存储
- 5) C/S 的代码字段被更新（参见表格“接收缓冲区代码”和“发送缓冲区代码”）
- 6) 中断标志寄存器的状态位被置位。如果相应的中断掩码使能，则生成中断

建议 CPU 按照以下方式处理（读取）邮箱中接收到的帧：

- 1) 读取邮箱的 C/S
- 2) 检查 BUSY 位是否为 0，表明邮箱被上锁。重复步骤 1) 直到该位清零。参见章节“邮箱上锁机制”
- 3) 读取邮箱内容。邮箱上锁时，移入流程不会更改其内容（参见章节“移入”）
- 4) 确认 CAN_IFLAG1 寄存器的相应标志位
- 5) 读取自由运行计时器以解锁邮箱

CPU 应该通过检查 CAN_IFLAG1 寄存器中，特定邮箱的状态标志位来轮询帧接收，而不是轮询该邮箱的代码字段。轮询代码字段无效，因为一旦接收到帧，CPU 处理邮箱（解锁邮箱后读取 C/S），代码字段不会转变为 EMPTY，而是仍然为 FULL（如表格“接收缓冲区代码”所示）。如果 CPU 试图在未提前安全地使邮箱失活的情况下，在读取邮箱后通过强制向 C/S 写 EMPTY 来解决这一问题，则与相应邮箱过滤器相匹配的新接收帧可能丢失。因此，禁止通过直接读取邮箱的 C/S 进行轮询，而应该读取 CAN_IFLAG1 寄存器。

接收帧的标识符始终存储在匹配的邮箱中。因掩码出现的匹配，相应邮箱中 ID 段的内容可能会发生改变。

当 CAN_MCR.SRXDIS 被置位时，FlexCAN 不会存储自身传输的任何 MB 内的帧，即使模块包含一个匹配的 Rx 邮箱，且不会产生中断标志或中断信号；当 CAN_MCR.SRXDIS 为 0 时，如果有匹配的 Rx 邮箱，FlexCAN 可以接收到自身发送的帧。

为了能够通过传统型 Rx FIFO 接收 CAN 帧，CPU 必须在冻结模式下使能并配置传统型 Rx FIFO（参见章节“传统型 Rx FIFO”）。接收到 Rx FIFO 有可用帧的中断时（CAN_IFLAG1.BUF5I），CPU 应通过下列程序处理接收到的帧：

- 1) 读取 C/S（可选，仅在 IDE 和 RTR 位使用掩码时才是必须的）
- 2) 读取 ID 段（可选，仅在使用掩码时才是必须的）
- 3) 读取数据字段
- 4) 读取 CAN_RXFIR 寄存器（可选）
- 5) 通过向 CAN_IFLAG1.BUF5I 写 1 来清零传统型 Rx FIFO 有可用帧的中断（必须，释放 MB 并允许 CPU 读取下一个传统型 Rx FIFO 条目）

当 CAN_MCR.DMA 被置位时，传统型 Rx FIFO 接收到帧，CAN_IFLAG1.BUF5I 不生成 CPU 中断，而是生成 DMA 请求（参见章节“传统型 Rx FIFO 的 DMA 操作”）。传统型 Rx FIFO 区域的 CAN_IMASK1 未被使用。

DMA 通过下列程序处理接收到的帧：

- 1) 读取 C/S（读取 0x80 地址，可选）

- 2) 读取 ID 段 (读取 0x84 地址, 可选)
- 3) 读取所有数据字节 (从 0x88 地址开始读取, 可选)
- 4) 读取最后一个数据字节 (读取 0x8C 地址, 必须)

30.3.6 匹配过程

匹配过程扫描 MB 内存, 搜索与从 CAN 总线上接收到的帧具有相同 ID 的 Rx MB。如果传统型或增强型 Rx FIFO 使能, 则可在邮箱和 FIFO 过滤器中选择扫描优先级。匹配从较低编号的 MB 向较高编号的 MB 进行。如果在第一种结构内未发现匹配, 则继续扫描下一个结构。如果 FIFO 已满, 则匹配算法在 FIFO 区域以外寻找匹配的 MB。

正在被接收的帧将存储在名为 Rx 串行报文缓冲区 (Rx SMB) 的隐藏辅助 MB 内, 该 MB 与普通 MB 结构相同。

匹配过程的开始点取决于以下条件:

- 如果接收到的是远程帧, 开始点为帧的 CRC 字段
- 如果接收到的是 DLC 字段为 0 的数据帧, 开始点为帧的 CRC 字段
- 如果接收到的是 DLC 字段不为 0 的数据帧, 开始点为帧的数据字段

如果在 FIFO 或邮箱中发现匹配的 ID, Rx SMB 的内容会通过移入流程传输至 FIFO 或匹配的邮箱。如果检测到任何 CAN 协议错误, 则在接收结束时没有匹配结果传输至 FIFO 或匹配的邮箱。

匹配过程扫描 Rx FIFO (使能时) 和激活的 Rx 邮箱 (代码字段为 EMPTY、FULL、OVERRUN 或 RANSWER) 内的所有匹配元素, 以寻找一次和 Rx SMB 内匹配元素的成功比较。与成功比较的匹配元素相关联的接收结构 (Rx FIFO 或邮箱) 称为匹配结构。扫描结束时, 从匹配结构中选择匹配获胜者, 具体取决于下表所列条件。

表 30-4 匹配架构

结构	SMB. RTR	CTRL2. RRS	CTRL2. EACEN	MB.IDE	MB.RTR	MB.ID ⁽¹⁾	MB.CODE
邮箱	0	—	0	cmp ⁽²⁾	cmp_msk ⁽³⁾	cmp_msk ⁽⁴⁾	EMPTY、FULL、OVERRUN
邮箱	0	—	1	cmp_msk	cmp_msk	cmp_msk	EMPTY、FULL、OVERRUN
邮箱	1	0	—	cmp	no_cmp	cmp	RANSWER
邮箱	1	1	0	cmp	no_cmp	cmp_msk	EMPTY、FULL、OVERRUN
邮箱	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY、FULL、OVERRUN
传统 型 Rx FIFO ⁽⁵⁾	—	—	—	cmp_msk	cmp_msk	cmp_msk	—

1) 对于邮箱结构, SMB.IDE 被置位时 ID 为 29 位 (标准 ID + 扩展 ID); SMB.IDE 为 0 时, ID 为 11 位 (标准 ID)。对于传统型 Rx FIFO 结构, ID 取决于 CAN_MCR.IDAM

2) cmp: 比较 Rx SMB 和 MB 的内容, 忽略掩码

- 3) no_cmp: 不比较 Rx SMB 和 MB 的内容
- 4) cmp_msk: 比较 Rx SMB 和 MB 的内容, 考虑掩码
- 5) 当 CAN_MCR.IDAM 为格式 C 时, 不考虑 SMB.IDE 和 SMB.RTR

满足以下任一条件时, 接收结构为自由接收:

- 邮箱代码字段为 EMPTY
- 邮箱代码字段为 FULL 或 OVERRUN, 且该邮箱已被处理过 (CPU 读取 C/S 并解锁, 参见章节“邮箱上锁机制”)
- 邮箱代码字段为 FULL 或 OVERRUN, 且该邮箱已失活 (参见章节“邮箱失活”)
- 传统型或增强型 Rx FIFO 未滿

邮箱和传统型 Rx FIFO 的扫描顺序为, 从最小编号的匹配元素到最大编号的匹配元素。

匹配获胜邮箱受 CAN_MCR.IRMQ 位影响。如果该位为 0, 则匹配获胜邮箱为第一个匹配的邮箱, 无论是否为自由接收邮箱; 如果该位被置位, 则根据以下优先级来选择匹配获胜邮箱:

- 1) 第一个自由接收的匹配邮箱
- 2) 最后一个非自由接收的匹配邮箱

可通过 CAN_CTRL2.MRP 位选择扫描优先级: 邮箱优先或传统型 Rx FIFO 优先。选择 Rx FIFO 优先时:

- 如果 Rx FIFO 是匹配结构且自由接收, 则 Rx FIFO 为匹配获胜者, 不再扫描邮箱
- 如果 Rx FIFO 不是匹配结构或不是自由接收, 则在上述邮箱中搜索匹配获胜者

选择邮箱优先时:

- 如果找到自由接收的匹配邮箱, 则该邮箱为匹配获胜者, 不再扫描 Rx FIFO
- 如果未找到匹配邮箱, 则在 Rx FIFO 中搜索匹配获胜者

如果在上述两种条件均未能满足的情况下, 找到了一个非自由接收的匹配邮箱, 则匹配获胜者由 CAN_MCR.IRMQ 位决定:

- CAN_MCR.IRMQ 为 0 时, 匹配获胜者为第一个匹配的邮箱
- CAN_MCR.IRMQ 被置位时, 如果 Rx FIFO 为自由接收的匹配结构, 则匹配获胜者是 Rx FIFO; 否则, 匹配获胜者是最后一个非自由接收的匹配邮箱

下表总结了所有匹配可能:

表 30-5 匹配可能性及由此产生的接收结构

RFEN	IRMQ	MRP	MB 匹 配	FIFO 匹 配	接收结构	描述
无 FIFO, 只有 MB, 匹配始终为 MB 优先:						
0	0	X ⁽¹⁾	无 ⁽²⁾	— ⁽³⁾	无	因无匹配而丢失帧
0	0	X	自由 ⁽⁴⁾	—	First MB	
0	1	X	无	—	无	因无匹配而丢失帧
0	1	X	自由	—	First MB	

RFEN	IRMQ	MRP	MB 匹 配	FIFO 匹 配	接收结构	描述
0	1	X	非自由	—	Last MB	溢出
FIFO 使能, FIFO 中无匹配, 如同 FIFO 不存在:						
1	0	X	无	无 ⁽⁵⁾	无	因无匹配而丢失帧
1	0	X	自由	无	First MB	
1	1	X	无	无	无	因无匹配而丢失帧
1	1	X	自由	无	First MB	
1	1	X	非自由	无	Last MB	溢出
FIFO 使能, 队列禁止:						
1	0	0	X	非满 ⁽⁶⁾	FIFO	
1	0	0	无	满 ⁽⁷⁾	无	因 FIFO 满而丢失帧 (FIFO 溢出)
1	0	0	自由	满	First MB	
1	0	0	非自由	满	First MB	
1	0	1	无	非满	FIFO	
1	0	1	无	满	无	因 FIFO 满而丢失帧 (FIFO 溢出)
1	0	1	自由	X	First MB	
1	0	1	非自由	X	First MB	溢出
FIFO 使能, 队列使能:						
1	1	0	X	非满	FIFO	
1	1	0	无	满	无	因 FIFO 满而丢失帧 (FIFO 溢出)
1	1	0	自由	满	First MB	
1	1	0	非自由	满	Last MB	溢出
1	1	1	无	非满	FIFO	
1	1	1	自由	X	First MB	
1	1	1	非自由	非满	FIFO	
1	1	1	非自由	满	Last MB	溢出

1) X: “don't care”条件

2) MB 匹配“无”表示帧没有匹配任何 MB (自由接收或非自由接收)

3) —: 禁止条件

4) MB 匹配“自由”表示帧至少匹配一个自由接收 MB, 无论是否匹配非自由接收 MB

5) FIFO 匹配“无”表示帧没有匹配任何 Rx FIFO 过滤器, 如同 Rx FIFO 不存在 (CAN_CTRL2.RFEN = 0)

6) FIFO 匹配“未滿”表示帧已匹配 FIFO 过滤器, 且有空余接收空间

7) FIFO 匹配“满”表示帧已匹配 FIFO 过滤器, 但 FIFO 没有空余接收空间, 无法存储

如果在匹配过程中发生了不安全的邮箱失活操作 (参见章节“邮箱失活”), 且失活的邮箱为临时匹配获胜邮箱, 则临时匹配获胜邮箱无效。匹配元素扫描不会停止或重启, 而是正常继续。后果是, 当前的匹配过程继续运行, 邮箱失活前的匹配元素比较仿佛不存在, 因此可能会丢失一个报文。

例如: 假设 Rx FIFO 被禁用, IRMQ 被置位, 2 号和 5 号 MB 具有相同的 ID, FlexCAN 开始接收具有该 ID 的报文。当第一个报文到达时, 首先匹配到 2 号 MB, 该 MB 的代码为 EMPTY, 因此报

文存储于此。当第二条报文到达时，再次匹配到 2 号 MB，由于该 MB 为非自由接收，匹配算法继续寻找，找到 5 号 MB 后将报文存储于此。如果又有一条具有相同 ID 的报文到达，匹配算法发现没有自由接收的匹配 MB，因而决定覆盖最后一个匹配 MB，即 5 号 MB。覆盖的同时，设置 MB 代码字段为 **OVERRUN**。

为多个 MB 编程相同的 ID，能够实现队列接收（除全功能 FIFO 外），以便 CPU 有更多时间处理 MB。CPU 通过检查 MB 的时间戳字段可以确定报文到达的顺序。

FlexCAN 的每个 MB 都支持私有掩码（CAN_RXIMRn 寄存器），使用掩码可以匹配一系列的 ID。在匹配算法阶段，如果掩码被置位，则比较相应的 ID 位；如果掩码位为 0，则不比较相应的 ID 位。私有掩码寄存器位于 RAM 中，因此复位后不会被初始化，并且只能在 FlexCAN 处于冻结模式时被编程。

FlexCAN 还支持一个仅有四个掩码寄存器（CAN_RXFGMASK、CAN_RXMGMASK、CAN_RX14MASK 和 CAN_RX15MASK）的替代掩码方案，该方案在 CAN_MCR.IRMQ 为 0 时启用。

30.3.7 转移流程

转移流程有两种类型：移入和移出。

30.3.7.1 移入

移入流程将 Rx SMB 接收到的报文复制到匹配的 Rx 邮箱或 FIFO 中。如果移入的目标是传统型 Rx FIFO，则报文的属性也会被复制至 CAN_RXFIR FIFO 内。每个 Rx SMB 均有独立的移入流程，但在给定的时间内只执行一个移入操作。只有当 Rx SMB 中的报文存在对应的匹配获胜者（参见章节“匹配过程”），且以下条件都满足时，才开始移入：

- CAN 总线已经到达或让位于：
 - a) 靠近携带 Rx SMB 信息帧的 Intermission 字段的第二位，或者
 - b) 靠近携带 Rx SMB 信息帧的过载帧的第一位
- 没有正在进行的匹配过程
- 目标邮箱未被 CPU 锁定
- 其他 Rx SMB 没有正在进行的移入流程。如果同时启动一个以上的移入流程，则两个都可以执行，并用最新的替代最旧的

本章节使用“待处理的移入”表示尚未满足上述所有条件的、将要开始的移入流程

如果满足以下任一条件，则移入流程被取消，且 Rx SMB 能够接收其他报文：

- CAN 总线到达携带报文的帧附近的 Intermission 字段首位，且相应的匹配过程完成后，目标邮箱失活
- 先前有一个待处理的移入，并且该移入具有相同的目标邮箱
- Rx SMB 正在接收 FlexCAN 传输的帧，且自接收禁止（CAN_MCR.SRXDIS 被置位）
- 检测到任何 CAN 协议错误

注：如果 FlexCAN 进入冻结或低功耗模式，待处理的移入流程不会被取消，而是处于等待状态。等待从冻结或低功耗模式退出并被解锁。如果 MB 在冻结模式下被解锁，则移入流程即刻执行。

FlexCAN 按照以下步骤执行移入流程：

- 1) 如果报文的目标是传统型 Rx FIFO，则将 IDHIT 移入 RXFIR FIFO
- 2) 根据为 Rx 存储元素选定的有效负载数，从 Rx SMB 读取所有数据字
- 3) 根据为 Rx 存储元素选定的有效负载数，向 Rx 邮箱写入所有数据字。如果存储元素的数据数小于报文 DLC 字段的有效负载数，则有效负载被截断，不符合目标大小的高阶字节丢失
- 4) 从 Rx SMB 读取 C/S 和 ID
- 5) 将 C/S 和 ID 写入 Rx 邮箱，并更新代码字段

移入流程并不是自动的，该流程会因目标邮箱失活（参见章节“邮箱失活”）而被立即取消。此情况可能导致邮箱被部分更新，因而出现不一致。例外情况是，如果移入目标为传统型或增强型 Rx FIFO，则流程不能被取消。当移入流程正在进行时，会置位目标报文缓冲区的 BUSY 位（代码字段的最低有效位），以警告 CPU 报文缓冲区的内容暂时不一致。

30.3.7.2 移出

移出流程是指，传输报文可用时，将报文从 Tx 邮箱复制到 Tx SMB 的过程（参见章节“仲裁过程”）。移出在以下情况发生：

- Intermission 字段的首位
- 总线关闭期间，TX 错误计数器的值在 124 到 128 之间
- 总线空闲状态期间
- 等待总线空闲状态期间

移出流程并不是自动的。总线空闲状态以外，只有 CPU 具有并发访问内存的优先级。在总线空闲状态，对于并发的内存访问，移出流程优先级最低。

30.3.8 数据一致性

为了保持数据的一致性以及 FlexCAN 的正确运行，CPU 必须遵守“传输流程”和“接收流程”所描述的规则。

30.3.8.1 传输中止机制

中止机制提供了一种安全的方式，以请求中止待处理的传输。如果传输被中止，或帧不能被中止而是被传输了，提供反馈机制通知 CPU。

中止传输必须满足两个主要条件：

- CAN_MCR.AEN 必须被置位

- CPU 的第一个动作必须是向 C/S 的代码字段写入中止代码 (0b1001)

为传输配置的激活的 MB 在更新前必须先中止。如果向当前正在传输的邮箱，或已载入 Tx SMB 中用于传输的邮箱写中止代码，则写操作无效，传输不受干扰。但中止请求会被捕获并保持挂起，直到满足以下条件之一：

- FlexCAN 丢失仲裁
- 传输期间出现错误
- FlexCAN 进入冻结模式
- FlexCAN 进入总线关闭状态
- 有一个过载帧

如果上述条件均未满足，则正确传输 MB，并置位相应的中断标志 (CAN_IFLAG1)；生成中断（如果使能）。当中断标志被置位时，中止请求自动清零；如果满足了上述条件之一，则帧不会被传输，中止代码被写入代码字段，相应的中断标志 (CAN_IFLAG1) 置位；生成中断（可选）。

如果 CPU 在内部传输开始前写入中止代码，则 MB 被更新，中断标志被置位。通过这种方式，CPU 只需读取中止代码即可确保 MB 安全失活。尽管 CAN_MCR.AEN 被置位且 CPU 写入了中止代码，但此时 MB 处于失活状态而非中止，因为传输尚未开始。中止邮箱的条件是中止请求被捕获且保持挂起状态，直到满足先前所述的条件之一。

中止流程总结如下：

- CPU 检查相应的 IFLAG 并清零
- CPU 向 C/S 的代码字段写入 0b1001
- CPU 等待相应的 IFLAG，以了解帧是被传输还是被中止
- CPU 读代码字段，以检查帧是被传输（代码 = 0b1000）或是被中止（代码 = 0b1001）
- 清除相应的 IFLAG，以允许 MB 被重新配置

30.3.8.2 邮箱失活

提供失活机制是为了保护邮箱不被 FlexCAN 内部流程所更新，从而允许 CPU 在更新后信赖邮箱数据的一致性，即使是在正常模式下。

只有当 MCR.AEN 位被清零时才能使能传输邮箱失活机制。

失活的邮箱在重新激活前，不会参与仲裁或匹配过程。参见章节“传输流程”和“接收流程”，以了解更多关于失活和重新激活邮箱的描述。

为了使邮箱失活，CPU 必须更新代码字段为 INACTIVE (0b0000 或 0b1000)。

因为 CPU 无法和 FlexCAN 内部流程同步更新代码字段，使邮箱失活可能会导致以下后果：

- 与失活的 Rx 邮箱过滤相匹配的总线帧，可能在未通知的情况下丢失，即使有其他具有相同过滤的邮箱
- 失活的 Tx 邮箱内包含报文的帧可能会在未设置相应 IFLAG 的情况下传输

为了确保安全失活并避免 Tx 邮箱产生上述后果，CPU 必须采用传输中止机制（参见章节“传输中止机制”）。

失活机制自动解锁邮箱（参见章节“邮箱上锁机制”）。

注：作为传统型或增强型 Rx FIFO 一部分的报文缓冲区不能被失活。FlexCAN 的传统型 Rx FIFO 区域没有写保护。当 RFEN 被置位时，CPU 必须保持传统型 Rx FIFO 区域的数据一致性。

30.3.8.3 邮箱上锁机制

除邮箱失活外，FlexCAN 还有另外一种适用于接收流程的数据一致性机制。当 CPU 从代码为 FULL 或 OVERRUN 的 Rx MB 读取 C/S 时，FlexCAN 假定 CPU 想以自动的方式读取整个 MB，因此为该 MB 设置内部上锁标志。当 CPU 读取自由运行计时器（全局解锁操作），或读取其他 MB 的 C/S（无论代码如何）时，锁定解除。CPU 写 C/S 也能解锁 MB，但不推荐用于正常解锁，因为这种方式会取消待处理的转移，可能丢失一个接收到的报文。MB 上锁可以防止在 CPU 读取某个 MB 时将新帧写入到该 MB 中。

注：锁定机制仅适用于 Rx MB，该 MB 不属于传统型 Rx FIFO 的一部分，且相应的代码不为 INACTIVE (0b0000) 或 EMPTY (0b0100)。Tx MB 不能被锁定。

例如，假定传统型 Rx FIFO 被禁用，2 号和 5 号 MB 被编程为相同的 ID，且 FlexCAN 已接收到报文并将其存入两个 MB 中。假设 CPU 决定读取 5 号 MB，与此同时具有相同 ID 的其他报文到达。当 CPU 读取 5 号 MB 的 C/S 时，该 MB 被锁定。新报文到达且匹配算法发现没有自由接收的 MB，因此决定覆盖 5 号 MB。但该 MB 被上锁，新报文无法写入。新报文将继续留在 Rx SMB，等待 MB 被解锁，然后才能被写入 MB。如果又有一条具有相同 ID 的报文到达时，MB 没有及时解锁，则新的报文的覆盖掉 Rx SMB 中的报文，且 MB 的代码字段或错误和状态寄存器中，不会有报文丢失的提示。

当报文从 Rx SMB 移入 MB 时，代码字段的 BUSY 位被置位。如果 CPU 读取 C/S 并发现 BUSY 位被置位，则应该延缓访问 MB 直到 BUSY 位被清零。

注：如果 BUSY 位被置位或者 MB 为空，则读取 C/S 不会锁定 MB。

失活优先于上锁。如果 CPU 失活一个锁定的 Rx MB，则其锁定状态被忽略且 MB 在本轮匹配中被标记为无效。Rx SMB 中的任何未被处理的报文不会传输至 MB。当 CPU 读取自由运行计时器或其他 MB 的 C/S 时，MB 被解锁。锁定和解锁机制在正常模式和冻结模式下具有相同的功能。

正常或冻结模式下解锁会导致移入未处理的报文。如果解锁在低功耗模式下进行，则移入过程会被延后，且只能在模块回到正常或冻结模式时发生。

30.3.9 增强型 Rx FIFO

置位 CAN_EFRCCR.ERFEN 可启用增强型 Rx FIFO，以代替传统型 Rx FIFO，两者不能同时启用。

增强型 Rx FIFO 支持通过 CAN_EFRCCR.ERFWM 配置水线（watermark）。如果 CAN_EFRCCR.ERFWM 为 1，则只在 FIFO 中存储了最少的报文时通知 CPU；当存储的报文数大于 CAN_EFRCCR.ERFWM 设定值时，CAN_ERFSR.ERFWMII 将由硬件置位。CAN_ERFIER.ERFWMIII 置

位时可触发中断；CAN_MCR.DMA 置位时可触发 DMA 传输。

为使用增强型 Rx FIFO 进行接收，CPU 必须执行以下配置程序：

- 1) 进入冻结模式
- 2) 置位 CAN_ERFCR.ERFEN 使能增强型 Rx FIFO（尚未使能时）

注：增强型 Rx FIFO 使能时，CAN_MCR.RFEN 必须为 0。

- 3) 置位 CAN_ERFSR.ERFCLR 以复位增强型 Rx FIFO
- 4) 清除 CAN_ERFSR.ERFUFW、CAN_ERFSR.ERFOVF、CAN_ERFSR.ERFWM1 和 CAN_ERFSR.ERFDA

- 5) 通过 CAN_ERFCR.NEXIF 配置过滤元素为扩展 ID 或标准 ID
- 6) 通过 CAN_ERFCR.ERFWM 配置增强型 Rx FIFO 水线
- 7) 通过 CAN_ERFIER 使能所需中断
- 8) 需要使用 DMA 时，请置位 CAN_MCR.DMA 并通过 CAN_ERFCR.DMALW 配置传输字数
- 9) 通过 CAN_ERFFELn 配置过滤元素

注：CAN_ERFFELn 位于 RAM 中，接收前需要进行初始化

- 10) 退出冻结模式

需更改增强型 Rx FIFO 的任何配置时，必须执行上述步骤。

增强型 Rx FIFO 支持扩展 ID 和标准 ID 两种过滤元素。每个扩展 ID 过滤元素存储于两个 CAN_ERFFELn 寄存器中；每个标准 ID 过滤元素存储于一个 CAN_ERFFELn 寄存器中。

30.3.9.1 增强型 Rx FIFO 匹配过程

CAN_ERFCR.ERFEN 为 1 时，FlexCAN 扫描 CAN_ERFFELn 存储区。如果至少有一个过滤元素满足匹配标准，CAN 报文内容将传输至增强型 Rx FIFO。

注：如果多个过滤器与传入的报文 ID 匹配，则匹配过程找到的第一个匹配过滤器将在 ID HIT 中指示。

每个 CAN_ERFFELn 寄存器可存储一个标准过滤元素。匹配标准由 CAN_ERFFELn.FSCH 定义：

1) FSCH = 00 时，过滤方案基于掩码和过滤器。仅当满足以下条件时，CAN 报文才与标准 ID 过滤元素匹配：

- CAN 报文为标准帧格式 (IDE = 0)
- ID[n] = 标准 ID 过滤器[n] 或标准 ID 掩码[n] = 0, n 为 0 ~ 10
- RTR = RTR 过滤器或 RTR 掩码为 0

表 30-6 标准 ID 过滤元素 (FSCH = 00)

31	30	29	28	27	26 ~ 16	15 ~ 12	11	10 ~ 0
FSCH = 00		预留		RTR 过滤器	标准 ID 过滤器	预留	RTR 掩码	标准 ID 掩码

2) FSCH = 01 时, 过滤方案基于范围。仅当满足以下条件时, CAN 报文才与标准 ID 过滤元素匹配:

- CAN 报文为标准帧格式 (IDE = 0)
- ID ≥ 标准 ID 过滤器 1
- ID ≤ 标准 ID 过滤器 2
- RTR = RTR 过滤器或 RTR 掩码为 0

表 30-7 标准 ID 过滤元素 (FSCH = 01)

31	30	29	28	27	26 ~ 16	15 ~ 12	11	10 ~ 0
FSCH = 01	预留			RTR 过滤器	标准 ID 过滤器 2	预留	RTR 掩码	标准 ID 过滤器 1

3) FSCH = 10 时, 过滤方案基于两个不带掩码的过滤器。仅当满足以下条件时, CAN 报文才与标准 ID 过滤元素匹配:

- CAN 报文为标准帧格式 (IDE = 0)
- ID[n] = 标准 ID 过滤器 1[n] 或 ID[n] = 标准 ID 过滤器 2[n], n 为 0 ~ 10
- RTR = RTR 过滤器 1 或 RTR = RTR 过滤器 2

表 30-8 标准 ID 过滤元素 (FSCH = 10)

31	30	29	28	27	26 ~ 16	15 ~ 12	11	10 ~ 0
FSCH = 10	预留			RTR 过滤器 2	标准 ID 过滤器 2	预留	RTR 过滤器 1	标准 ID 过滤器 1

每对 CAN_ERFFELn 寄存器可存储一个扩展过滤元素。匹配标准由 CAN_ERFFELn.FSCH 定义:

1) FSCH = 00 时, 过滤方案基于掩码和过滤器。仅当满足以下条件时, CAN 报文才与扩展 ID 过滤元素匹配:

- CAN 报文为扩展帧格式 (IDE = 1)
- ID[n] = 扩展 ID 过滤器[n] 或扩展 ID 掩码[n] = 0, n 为 0 ~ 28
- RTR = RTR 过滤器或 RTR 掩码 = 0

表 30-9 扩展 ID 过滤元素 (FSCH = 00)

31	30	29	28 ~ 0
FSCH = 00	RTR 过滤器		扩展 ID 过滤器
预留	RTR 掩码		扩展 ID 掩码

2) FSCH = 01 时, 过滤方案基于范围。仅当满足以下条件时, CAN 报文才与扩展 ID 过滤元素匹配:

- CAN 报文为扩展帧格式 (IDE = 1)
- ID ≥ 扩展 ID 过滤器 1
- ID ≤ 扩展 ID 过滤器 2
- RTR = RTR 过滤器或 RTR 掩码 = 0

表 30-10 扩展 ID 过滤元素 (FSCH = 01)

31	30	29	28 ~ 0
FSCH = 01		RTR 过滤器	扩展 ID 过滤器 2
预留		RTR 掩码	扩展 ID 过滤器 1

3) FSCH = 10 时, 过滤方案基于两个不带掩码的过滤器。仅当满足以下条件时, CAN 报文才与扩展 ID 过滤元素匹配:

- CAN 报文为扩展帧格式 (IDE=1)
- ID[n] = 扩展 ID 过滤器 1[n] 或 ID[n] = 扩展 ID 过滤器 2[n], n 为 0 ~ 28
- RTR = RTR 过滤器 1 或 RTR = RTR 过滤器 2

表 30-11 扩展 ID 过滤元素 (FSCH = 10)

31	30	29	28 ~ 0
FSCH = 10		RTR 过滤器 2	扩展 ID 过滤器 2
预留		RTR 过滤器 1	扩展 ID 过滤器 1

30.3.9.2 增强型 Rx FIFO 的 DMA 操作

置位 CAN_ERFCR.ERFEN 和 CAN_MCR.DMA 可使能增强型 Rx FIFO 的 DMA 功能。DMA 控制器通过读取增强型 Rx FIFO 输出接口地址范围 0x2000 ~ 0x204C 的报文缓冲区结构来读取接收到的报文。

DMA 操作期间 CPU 禁止访问增强型 Rx FIFO 输出接口地址范围。CAN_MCR.DMA 置位前, CPU 必须处理增强型 Rx FIFO 状态位。否则, 这些位可能会显示 FIFO 中有待处理的数据, 并错误地生成 DMA 请求。

CAN_MCR.DMA 清零前, CPU 必须首先清除 CAN_ERFSR 寄存器的 ERFUFW、ERFOVF、ERMWMI 和 ERFDA 字段, 之后写 1 至 CAN_ERFSR.ERFCLR 以清除增强型 Rx FIFO。

当增强型 Rx FIFO 中至少有一帧可被读取时, CAN_ERFSR.ERFDA 由硬件置位; 当 FIFO 中存储的帧数大于 CAN_ERFCR.ERFWM 设置的水线时, CAN_ERFSR.ERFWMI 由硬件置位, 同时产生 DMA 请求。一旦接收到请求, DMA 控制器能够通过增强型 Rx FIFO 的输出读取报文。DMA 控制器必须执行由 CAN_ERFCR.ERFWM 设定的读取次数, 且每个读取过程必须在 CAN_ERFCR.DMALW 设定的地址处结束。

DMA 传输不能被动态更改, 因此需要编程 CAN_ERFCR.DMALW 以便增强型 Rx FIFO 存储 CAN 总线上最大的 CAN 报文。DLC 字段决定有效数据字节, 参见表格“有效数据字节”。

每次 DMA 控制器从增强型 Rx FIFO 读取一条报文时, FlexCAN 清除 CAN_ERFSR.ERFDA, 至少有一条报文存储在 FIFO 中时该位再次置位。报文被读取时 CAN_ERFSR.ERFWMI 由硬件清除, FIFO 中的报文数大于 CAN_ERFCR.ERFWM 的设定值时该位重新置位。

30.3.9.3 增强型 Rx FIFO 清除操作

当 CAN_ERFCR.ERFEN 置位时, CPU 可通过写 1 至 CAN_ERFSR.ERFCLR 以清除增强型 Rx

FIFO。清除操作复位内部 FIFO 指针，不改变存储在 RAM 中的 FIFO 内容。该操作只能在冻结模式下执行，其他模式下被硬件锁定。清除操作不会清除 CAN_EFRSR 的 ERFUFW、ERFOVF、ERFDA 和 ERFWMI 字段，因此必须在执行该操作前处理这些字段。

30.3.10 传统型 Rx FIFO

CAN_MCR.RFEN 被置位时传统型 Rx FIFO 使能。传统型 Rx FIFO 有 6 条报文的深度。传统型 Rx FIFO 结构占有的内存区域（包括报文缓冲区和传统型 Rx FIFO 引擎），参见章节“传统型 Rx FIFO 结构”。通过在传统型 Rx FIFO 的输出接口重复地读取报文缓冲区结构，CPU 可以按照报文被接收的顺序进行读取。

注：CAN FD 使能时不能使用传统型 Rx FIFO。

当传统型 Rx FIFO 中至少有一个帧可被读取时，CAN_IFLAG1.BUF5I（传统型 Rx FIFO 中有可用帧）被置位。使能相应的掩码位可生成中断。一旦接收到中断，CPU 能够读取报文（以报文缓冲区的方式访问传统型 Rx FIFO 的输出）和 CAN_RXFIR 寄存器，之后清除中断。如果传统型 Rx FIFO 中有更多的报文，则清除中断的行为，会将 FIFO 输出更新为下一条报文，并将 CAN_RXFIR 更新为该报文的属性，再次向 CPU 发出中断。否则，标志位会被清除。传统型 Rx FIFO 的输出仅在 CAN_IFLAG1.BUF5I 被置位时有效。

接收到新报文后，传统型 Rx FIFO 中未读取的报文数从 4 增加至 5 时，CAN_IFLAG1.BUF6I（传统型 Rx FIFO 警告）被置位，表明传统型 Rx FIFO 将满。该标志位会一直保持置位，直到 CPU 将其清零。

因传统型 Rx FIFO 满而导致新来的报文丢失时，CAN_IFLAG1.BUF7I（传统型 Rx FIFO 溢出）被置位。当传统型 Rx FIFO 满且新报文被邮箱捕获时，CAN_IFLAG1.BUF7I 不会被置位。BUF7I 会一直保持置位，直到 CPU 将其清零。

清除这三个标志位中的任何一个都不会影响其他两个的状态。

如果 IFLAG 位被置位且相应的掩码位也被置位，则生成中断。

过滤标准通过编程最多 104 个 32 位寄存器来制定，根据 CAN_CTRL2.RFFN 的设定，可被配置为下列格式之一：

- 格式 A: 104 IDAFs（包含 IDE 和 RTR 的标准或扩展 ID）
- 格式 B: 208 IDAFs（包含 IDE 和 RTR 的标准 ID 和扩展 ID 的 14 位）
- 格式 C: 416 IDAFs（标准或扩展 ID 的 8 位）

注：选定的格式适用于过滤表的所有条目。表格里的格式不能混用。参见章节“传统型 Rx FIFO 结构”。

传统型 Rx FIFO 中的每个可用帧均有一个相应的 IDHIT（ID 接收过滤命中指示），可从 C/S 的 IDHIT 字段中读取。CPU 还可通过访问 CAN_RXFIR 寄存器获得该信息。CAN_RXFIR.IDHIT 字段在 CAN_IFLAG1.BUF5I 标志被置位时有效，与传统型 Rx FIFO 输出的报文相关联。CAN_RXFIR 寄存器必须在标志被清零前读取，以确保信息指示传统型 Rx FIFO 里正确的帧。

根据 CAN_CTRL2.RFFN 的设置，过滤表中高达 32 个元素分别受到私有掩码寄存器

(CAN_RXIMRn)的影响,方便定义很高效的过滤条件。如果 CAN_MCR.IRMQ 为 0,则传统型 Rx FIFO 过滤表受 CAN_RXFGMASK 影响。

30.3.10.1 传统型 Rx FIFO 的 DMA 操作

通过置位 CAN_MCR.RFEN 和 CAN_MCR.DMA 可以使能传统型 Rx FIFO 的 DMA 功能。

DMA 控制器可以通过读取传统型 Rx FIFO 输出接口 0x80 ~ 0x8C 地址范围的报文缓冲区结构来读取接收到的报文。

当 CAN_MCR.DMA 被置位时, CPU 禁止访问传统型 Rx FIFO 输出接口地址范围。CAN_MCR.DMA 置位前, CPU 必须处理传统型 Rx FIFO 区域被置位的 IFLAG。否则,这些 IFLAG 可能会显示传统型 Rx FIFO 中有待处理的数据,并错误地生成 DMA 请求。CAN_MCR.DMA 被清零前, CPU 必须执行传统型 Rx FIFO 清除操作。

当传统型 Rx FIFO 中至少有一个帧可被读取时, CAN_IFLAG1.BUF5I (传统型 Rx FIFO 中有可用帧)被置位,同时生成 DMA 请求。一旦接收到请求, DMA 控制器能够读取报文 (以报文缓冲区的方式访问传统型 Rx FIFO 的输出)。DMA 读取流程必须以读地址 0x8C 结束,该操作会清零 CAN_IFLAG1.BUF5I,并将传统型 Rx FIFO 输出更新为下一条报文 (如果传统型 Rx FIFO 不为空),将 CAN_RXFIR 更新为该报文的属性。如果传统型 Rx FIFO 内还存储了更多的报文,则 CAN_IFLAG1.BUF5I 被重新置位并发出新的 DMA 请求。否则, CAN_IFLAG1.BUF5I 仍为清零状态。

注: DMA 完成传统型 Rx FIFO 读取后,不能读取 CAN_RXFIR 寄存器的内容。IDHIT 信息还可从地址 0x080 的 C/S 获取 (参见章节“传统型 Rx FIFO 结构”)。

DMA 功能使能时, CAN_IFLAG1.BUF6I 和 CAN_IFLAG1.BUF7I 不可用。

当 FlexCAN 与 DMA 协作时, CPU 不会收到任何传统型 Rx FIFO 中断,且不会清零相关的 IFLAG。此外,相关的 IMASK 并不用于屏蔽 DMA 请求的生成。

30.3.10.2 传统型 Rx FIFO 清除操作

当 CAN_MCR.RFEN 被置位时,传统型 Rx FIFO 清除操作用于清空传统型 Rx FIFO 的内容。CAN_MCR.RFEN 被置位后,当 CPU 向 CAN_IFLAG1.BUF0I 写 1 时,传统型 Rx FIFO 被清除。该操作只能在冻结模式下进行,且不会清零传统型 Rx FIFO 相关的 IFLAG。因此, CPU 在执行传统型 Rx FIFO 清除操作前,必须处理所有传统型 Rx FIFO 相关的 IFLAG。当传统型 Rx FIFO 与 DMA 协作时,传统型 Rx FIFO 清除操作会清零 CAN_IFLAG1.BUF5I 标志, DMA 请求被取消。

注:除非 CAN_MCR.DMA 被置位,否则传统型 Rx FIFO 清除操作不会清零 IFLAG。当 CAN_MCR.DMA 被置位时,只有 CAN_IFLAG1.BUF5I 标志会被清零。

30.3.11 CAN 协议

本章节描述了 CAN 协议相关的功能。

30.3.11.1 CAN FD 帧

ISO 11898-1 标准指定了符合 ISO 11898-1 (2003) 标准的经典帧格式，并引入了 CAN FD (Flexible Data Rate) 帧格式。经典帧格式支持高达 1Mbps 的比特率，以及每帧 8 字节的有效负载。FD 帧格式支持超过 1Mbps 的比特率，以及每帧超过 8 字节的有效负载。FlexCAN 可以收发 CAN FD 和经典 CAN 格式交替的报文。

CAN FD 帧中有三个附加的控制位：

- 扩展数据长度位 (EDL)：支持更多的数据负载
- 比特率切换位 (BRS)：决定 CAN FD 帧是否切换比特率
- 错误状态指示 (ESI)：错误主动节点发送显性；错误被动节点发送隐性

CAN FD 格式不支持远程帧，远程帧总是以经典 CAN 格式传输。接收到 FD 帧并匹配邮箱时，接收报文缓冲区的 RTR 位将被无效化。

CAN FD 报文数据字段可超过 8 字节，支持 12 至 64 字节。CAN FD 报文可切换比特率，使 CAN 帧的控制字段、数据字段和 CRC 字段比特率高于帧的开始和结束。各种不同的帧格式如下图所示：

Preliminary

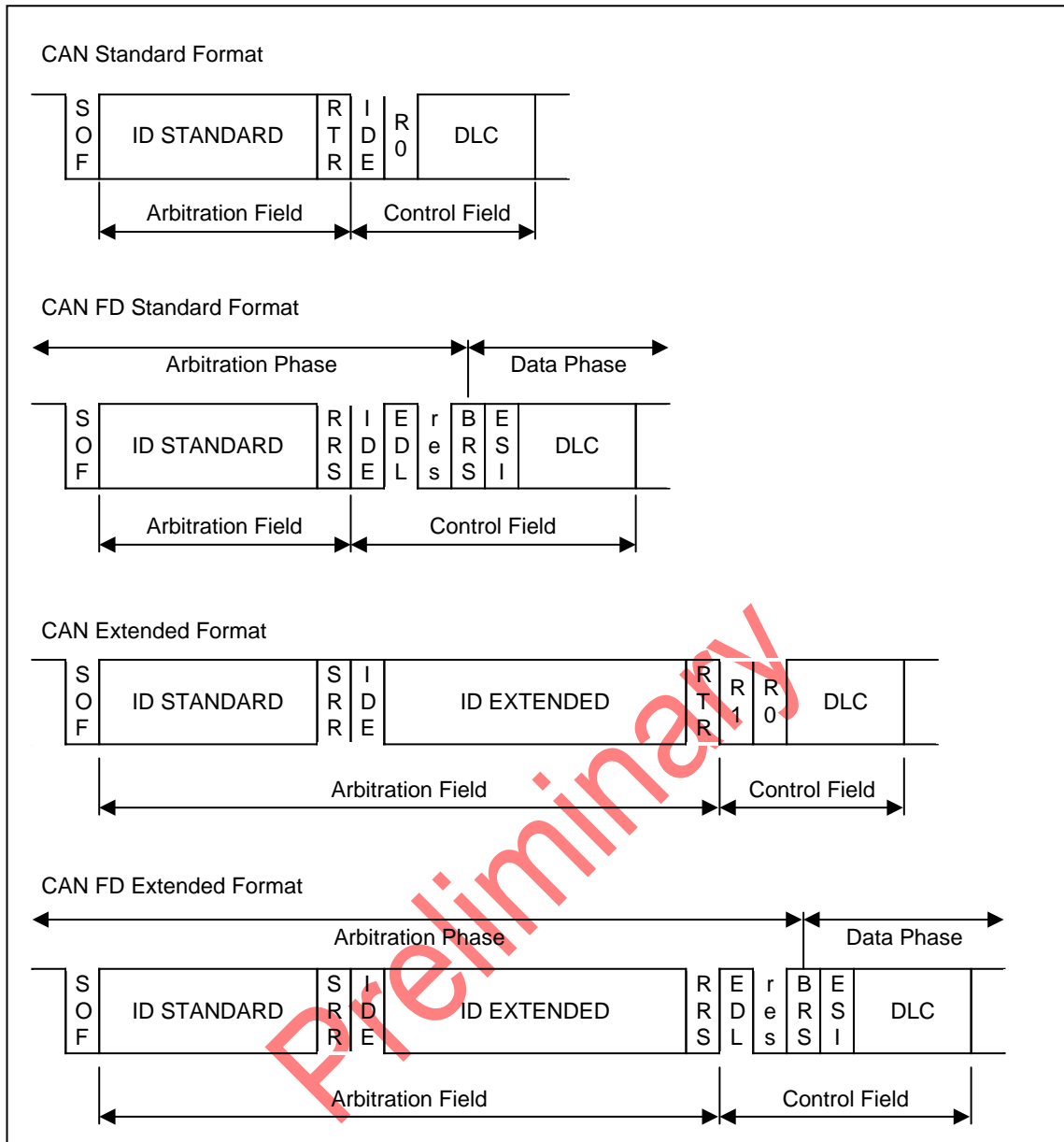


图 30-2 CAN 帧格式

置位 `CAN_MCR.FDEN` 将使能 CAN FD 报文的收发。CAN 帧的隐性 R0 位（标准格式）或隐性 R1 位（扩展格式），都被解码为 EDL 位。CAN FD 帧由隐性 EDL 位识别；经典 CAN 帧由显性 EDL 位识别。

CAN FD 帧从 SOF（帧起始）到 BRS 的仲裁段，以标称比特率传输；从 BRS 到 CRC 界定符的数据段，以数据比特率传输；从 CRC 界定符到 Intermission 位，传输恢复为标称比特率。如果 CAN FD 帧中 BRS 为隐性，则位时序在 BRS 的采样点发生变化。BRS 位之前，CAN FD 仲裁段的标称位时序由 `CAN_CBT` 或 `CAN_CTRL1` 寄存器定义；检测到隐性 BRS 时，数据位时序由 `CAN_FDCBT` 寄存器定义。

注：如果标称和数据位时序中的 Tq 不同，则仲裁段的量化误差（不超过 $1Tq$ ）可能作为相位误差出现。此情况可能在仲裁段切换到数据段后发生，并持续到下一个同步事件。因此，标称和数据位时序的 Tq 应该相同，由此最小化 CAN 总线上出现错误帧的机会，并优化使用 CAN FD 帧的网络的时钟容忍度。

所选 Tx MB 的 BRS 置位时，CAN_FDCTRL.FDRATE 使能所有发送帧的比特率切换。如果 CAN_FDCTRL.FDRATE 为 0，则发送以标称速率进行，与 BRS 无关。该位可随时写入，但只对接收或发送的下一条报文起效。

在 CRC 界定符位的采样点或检测到错误时恢复标称位时序。下图描述了 BRS 位为隐性时，进入和离开数据段的机制。

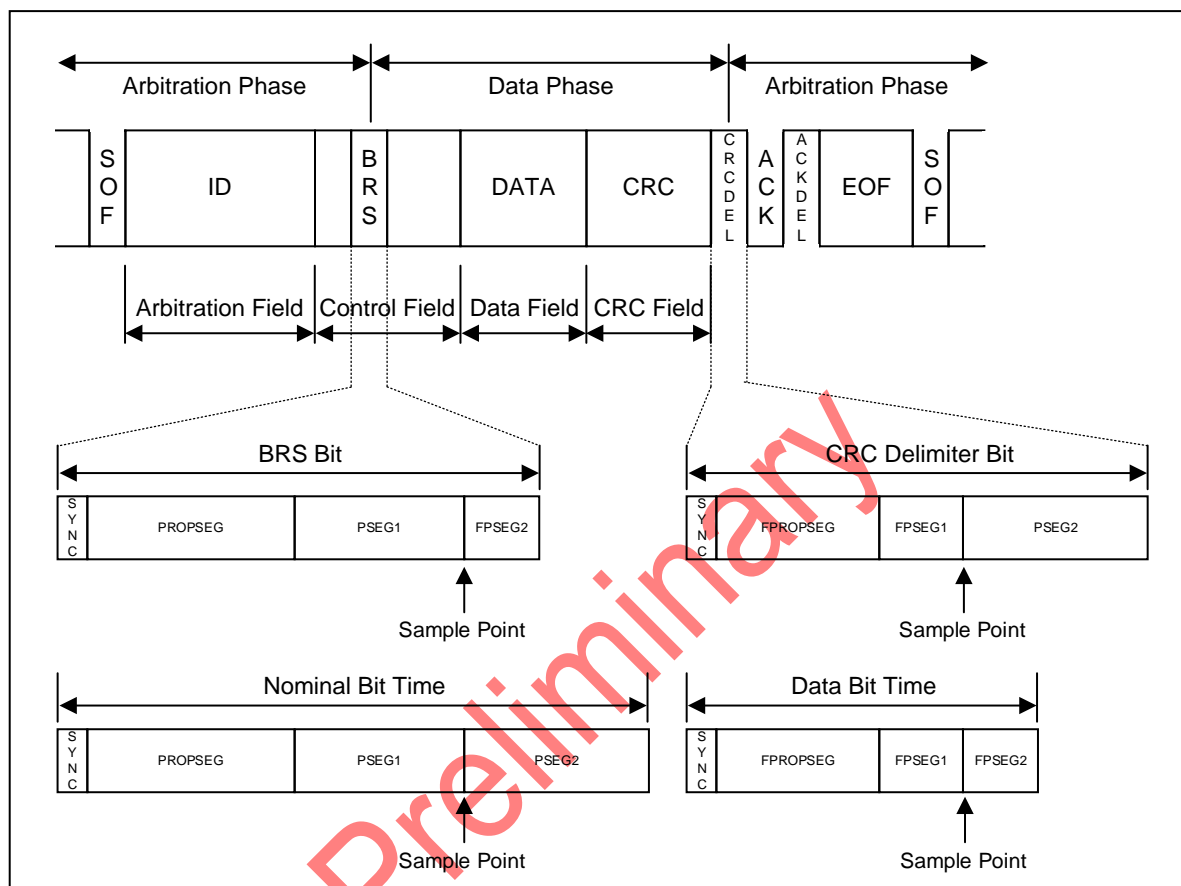


图 30-3 CAN FD 报文比特率切换机制

注：经典 CAN 帧中，CRC 界定符为一个隐性位；CAN FD 帧中，CRC 界定符可由一个或两个隐性位组成。FlexCAN 只发送一个隐性位作为 CRC 界定符，但接受从隐性到显性变化沿（ACK 槽起始）前的两个隐性位。作为接收器的 FlexCAN 在第一个 CRC 界定符后发送响应位。CAN FD 帧中，FlexCAN 接受一个两位的显性 ACK 槽作为有效 ACK，以补偿接收器间的相位偏移。

CAN FD 数据段可配置的最大比特率取决于 PE 时钟频率。例如，PE 时钟为 40MHz，可配置的最小位时间为 5Tq，则数据段比特率为 8Mbps。

如果帧由 FlexCAN 节点发起，则 ESI 位由传输开始时 FlexCAN 的错误状态决定；如果 FlexCAN 作为报文的网关，则 ESI 位由原始的传输节点决定。如果发送端为错误被动，则 ESI 为隐性；否则为显性，详见下表。

表 30-12 ESI 位的写入与传输

帧起始时 FlexCAN 的故障界定状态	Tx MB 的 ESI 位	被发送的 ESI
错误主动	0	0（错误主动）
错误被动	0	1（错误被动）

帧起始时 FlexCAN 的故障界定状态	Tx MB 的 ESI 位	被发送的 ESI
错误主动	1	1 (错误被动)
错误被动	1	1 (错误被动)

不同的 CAN 帧格式有不同的 CRC 多项式，每个多项式的汉明距离为 6。

- CRC_15 (0xC599) 用于经典 CAN 帧: $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- CRC_17 (0x3685B) 用于数据字段最大 16 字节的 CAN FD 帧: $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1$
- CRC_21 (0x302899) 用于数据字段大于 16 字节的 CAN FD 帧: $x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1$

帧起始时同时计算三个 CRC 多项式，根据 EDL 位和 DLC 字段的值选择用于传输的 CRC 序列。FlexCAN 接收到报文后，对 EDL 和 DLC 进行解码，选择合适的 CRC 多项式来检查 CRC 错误。

CAN FD 帧将填充位包含在位流中进行 CRC 计算；经典 CAN 帧则不包括填充位。传输完与 CRC 计算相关的最后一位后，传输报文（CAN FD 帧和非 FD 帧）的 CRC 计算值存储于 CAN_FDCRC 寄存器中，该值位宽与报文类型相匹配。CAN_CRCCR 寄存器只报告经典 CAN 报文的有效 CRC。

CAN FD 帧改变了 CRC 序列的位填充方法，将填充位插入至固定位置。当 FlexCAN 发送 CAN FD 帧时，即使前一个字段的最后一位不满足 CAN 填充条件，也会在 CRC 序列的第一位前插入一个固定的填充位。之后在 CRC 序列的每四位后插入额外的填充位，所有固定填充位的值都是其前一位的相反值。FlexCAN 接收到 CAN FD 帧时，将丢弃固定填充位后进行 CRC 检查。如果固定填充位与其前一位具有相同值，则检测到位填充错误。

FlexCAN 检测 CAN FD 帧错误的方法与经典 CAN 帧相同。CAN_ECR.RXERRCNT 和 TXERRCNT 分别计数 Rx 和 Tx 错误；CAN_ECR.RXERRCNTFAST 和 TXERRCNTFAST 分别计数 BRS 置位时 CAN FD 帧数据段的 Tx 和 Rx 错误。CAN FD 帧和非 FD 帧更新错误计数器的规则相同，细节见寄存器 CAN_ECR 的描述。

CAN_ESR1 寄存器中的错误标志 BITERR1、BITERR0、ACKERR、CRCERR、FRMERR 和 STFERR 指明 CAN FD 帧和非 FD 帧中的错误。CAN_CTRL1.ERRMSK 置位时将产生 ERRINT 中断。CAN_ESR1 寄存器提供了额外的错误标志（BITERR1FAST、BITERR0FAST、CRCERRFAST、FRMERRFAST 和 STFERRFAST），用于指示 BRS 置位时 CAN FD 帧数据段发生的错误。CAN FD 帧数据段不检测 ACKERR。CAN_ESR1.FLTCONF 指示 CAN FD 帧和非 FD 帧的故障界定状态，且只受 RXERRCNT 和 TXERRCNT 错误计数器影响。RXERRCNTFAST 和 TXERRCNTFAST 计数器可被视为一种状态，帮助检测与比特率相关的错误性质。

当 FlexCAN 在数据段收发 CAN FD 报文并检测到错误时，将立即切换回仲裁段，并以标称速率发起一个错误标志。

CAN FD 帧重同步和硬同步方式与经典 CAN 帧相同。CAN FD 帧中 EDL 到 R0 的隐性至显性变化沿也会执行硬同步。FlexCAN 在 CAN FD 数据段传输时不会重同步。

30.3.11.2 收发器延迟补偿

当报文的 BRS 位置位时，CAN FD 协议允许以比仲裁段使用的标称速率更高的比特率收发数据，最高 8 Mbps。

在 CAN FD 帧的数据段，如果发送器无法在其最新传输的位的采样点接收到该位，则会检测到位错误。当比特率切换使能（BRS 置位）时，数据段的 CAN 位时间长度可能会短于收发器的回路延迟，因此会妨碍在当前 CAN 位时间间隔内正确地比较发送位和接收位。

FlexCAN 支持收发器延迟补偿（TDC）机制，该机制定义了二次采样点，在该采样点正确地比较发送位与接收位，以检查位错误。

TDC 机制通过 CAN_FDCTRL.TDCEN 使能，仅在 BRS 置位的 CAN FD 帧的数据段有效。TDC 既不影响非 FD 帧，也不影响以正常比特率传输的 FD 帧。TDC 从 BRS 位的采样点到 CRC 界定符位的采样点之间一直处于活动状态，前提是传输中的各个报文都置位了 BRS。TDC 活动时，在实际接收到的位和延迟的发送位之间进行比较，其中，延迟基于收发器环路延迟测量值进行计算。

注：使用 TDC 机制的发送器会忽略 CRC 界定符位的实际值。CRC 字段末尾的全局错误将导致接收器发送错误帧，发送器将在确认帧或帧结束时检测到错误帧。

对于每个 BRS 置位的发送 FD 帧，延迟测量由隐性 EDL 位到显性 R0 位的转换触发（如下图所示）。从发送的 EDL - R0 边沿到接收到的 EDL - R0 边沿，以 PE 时钟周期测量环路延迟。测量到的环路延迟时间加上 CAN_FDCTRL.TDCOFF 中指定的偏移量，决定了二次采样点的位置。CAN_FDCTRL.TDCVAL（最大 63 个时钟周期）字段存储计算结果。

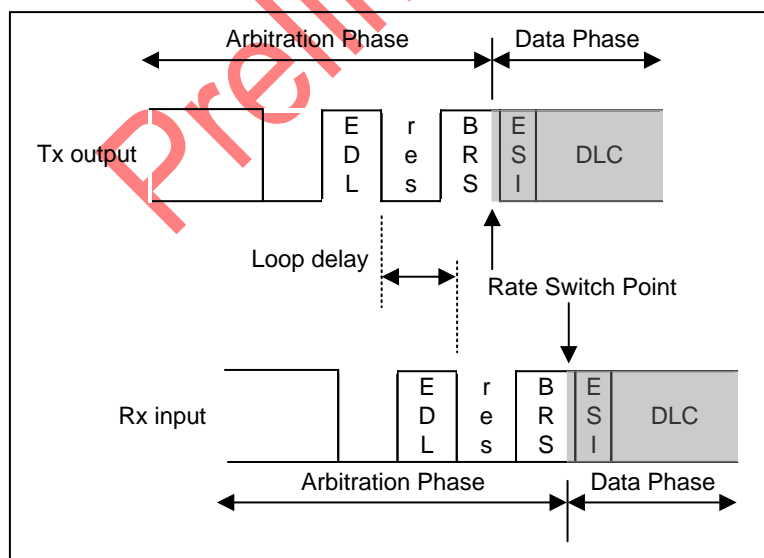


图 30-4 收发回路延迟测量

测量到的环路延迟与 CAN 位的边沿有关，因此不足以用来决定二次采样点。TDCOFF 或 ETDCOFF 用于将二次采样点从边沿移至位时间的中间点（例如，数据段位时间的一半），因此 TDCOFF 或 ETDCOFF 的值不能大于数据段的 CAN 位持续时间。

如果二次采样点被设置得非常靠近 CAN 位边沿（SYNC 字段），则数据段的位采样过程可能会出现问题。为了确保在最佳区域进行位采样，TDC 偏移量应按照以下等式进行配置：

$$\text{Offset} = (\text{FPSEG1} + \text{FPROPSEG} + 2) \times (\text{FPRESDIV} + 1)$$

下图显示了配置后二次采样点 (SSP) 的位置。

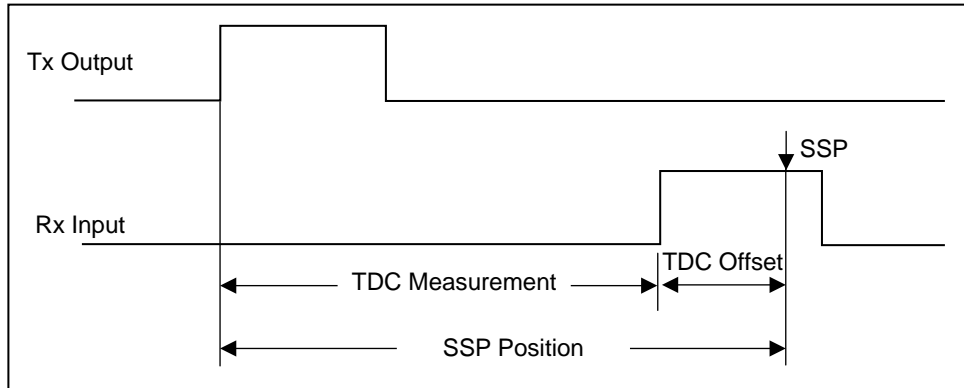


图 30-5 具有最优值的 SSP 位置

在比特率切换使能的 CAN FD 帧的数据段，每个 Tx CAN 位开始时，暂存 Tx 位于缓冲区。基于 CAN_FDCTRL.TDCVAL 的倒计时启动，最后比较接收到的 Rx 位（经过外部环路延迟以及指定的偏移量）与存储的 Tx 位。如果在二次采样点检测到一个位错误，FlexCAN 会在下一个采样点向 CAN 总线发出错误标志。

在仲裁段，延迟补偿始终被禁止。FlexCAN 收发器延迟补偿在数据段可补偿的最大延迟为 3 个 CAN 位时间 - 2T_q。当 TDC 机制超出范围且无法补偿时，CAN_FDCTRL.TDCFAIL 标志被置位。

30.3.11.3 远程帧

CPU 可以通过将邮箱传输时的 RTR 位设置为 '1' 来编程邮箱为远程帧。远程帧成功传输后，邮箱变为具有相同 ID 的接收报文缓冲区。

FlexCAN 将结果应答帧接收到和传输远程帧相同的报文缓冲区中。由于应答帧的 RTR = 0，而远程帧的 RTR = 1，在比较传入的应答帧时，对该报文缓冲区的过滤不能考虑 RTR 位。要排除 RTR 位，请确保以下位中的一个或两个都为 0：

- 各掩码寄存器的 RTR 位，可以是私有掩码寄存器 RXIMRn[31] 或全局掩码寄存器 RXMGMASK[31]，取决于 CAN_MCR.IRMQ。CAN_MCR.IRMQ = 1 时，RXIMRn[31] 禁用报文缓冲区 n 的 RTR 比较；CAN_MCR.IRMQ = 0 时，RXMGMASK[31] 禁用所有报文缓冲区的 RTR 比较
- CAN_CTRL2.EACEN = 0 可全局地禁止比较传入帧的 RTR 位，忽略报文缓冲区的 RTR 掩码位

当 FlexCAN 接收到远程帧时，根据远程请求存储位 (CAN_CTRL2.RRS)、传统型 Rx FIFO 使能位 (CAN_MCR.RFEN) 和增强型 Rx FIFO 使能位 (CAN_MCR.ERFEN) 的配置，可采取以下四种方式进行处理：

- 如果 RRS 为 0，则帧的 ID 与代码字段为 0b1010 的传输报文缓冲区的 ID 相比较。如果有匹配的 ID，则该邮箱被传输。注：如果匹配邮箱的 RTR 位被置位，则 FlexCAN 传输远程帧进行

响应。接收到的远程帧不会存入接收缓冲区，而是仅用于触发应答帧的传输。掩码寄存器不用于远程帧的匹配，接收到的帧的所有 ID 位（除了 RTR）都应匹配。如果接收到远程帧并且匹配了一个邮箱，则报文缓冲区立即进入内部仲裁过程，但会被认为是一个正常的没有高优先级的 Tx 邮箱。该帧的数据长度与启动传输的远程帧的 DLC 字段无关。

- 如果 RRS 被置位，则帧的 ID 与代码字段为 0b0100、0b0010 或 0b0110 的接收邮箱的 ID 相比较。如果有匹配的 ID，则该邮箱以与数据帧相同的方式存储远程帧。不会自动产生远程响应帧；掩码寄存器被用于匹配过程。
- 如果 RFEN 被置位，则 FlexCAN 不会自动响应与传统型 Rx FIFO 过滤条件匹配的远程帧。如果远程帧匹配目标 ID 中的任何一个，则会被存储于传统型 Rx FIFO 并提交给 CPU。

注：对于过滤格式 A 和 B，可以选择是否接收远程帧；对于格式 C，远程帧总是被接收（如果 ID 匹配）。远程帧被认为是普通帧，当接收成功且传统型 Rx FIFO 已满时，会产生传统型 Rx FIFO 溢出。

- 如果 ERFEN 被置位，则 FlexCAN 不会自动响应与增强型 Rx FIFO 过滤条件匹配的远程帧。远程帧被认为是普通帧，当接收成功且增强型 Rx FIFO 已满时，会产生增强型 Rx FIFO 溢出。

注：CAN FD 格式中没有远程帧，RTR 位被固定的显性 RRS 位取代。FlexCAN 以经典 CAN 格式收发远程帧。

30.3.11.4 过载帧

FlexCAN 在 CAN 总线上检测到以下条件时会传输过载帧：

- 在 Intermission 的第一/第二位检测到一个显性位
- 在 Rx 帧结束字段的第 7 位（最后一位）检测到一个显性位
- 在错误帧界定符或过载帧界定符的第 8 位（最后一位）检测到一个显性位

30.3.11.5 时间戳

自由运行计时器的值为 CAN 总线上 ID 段起始时的采样值，在移入结束时存入报文缓冲区的时间戳字段，为网络行为提供时间参考。

自由运行计时器通过 FlexCAN 的位时钟计时，位时钟定义了 CAN 总线上的波特率。报文传输/接收时，每传输或接收一位，计时器就加 1。当总线上没有报文时，使用先前编程的波特率计数。

模块禁止模式和冻结模式时自由运行计时器不会增加。计时器可以在接收到特定帧时复位，使网络时间同步。

30.3.11.6 协议时序

下图为产生 PE 时钟的电路结构。时钟源选择位（CAN_CTRL1.CLKSRC）定义了内部时钟为异步时钟或同步时钟。其中，同步时钟为外设时钟；异步时钟的时钟源可选（细节请参考 RCC 章节 RCC_CFGR2 寄存器）。为保证可靠运行，应在模块禁止模式时（CAN_MCR.MDIS 置位）选择时钟源。

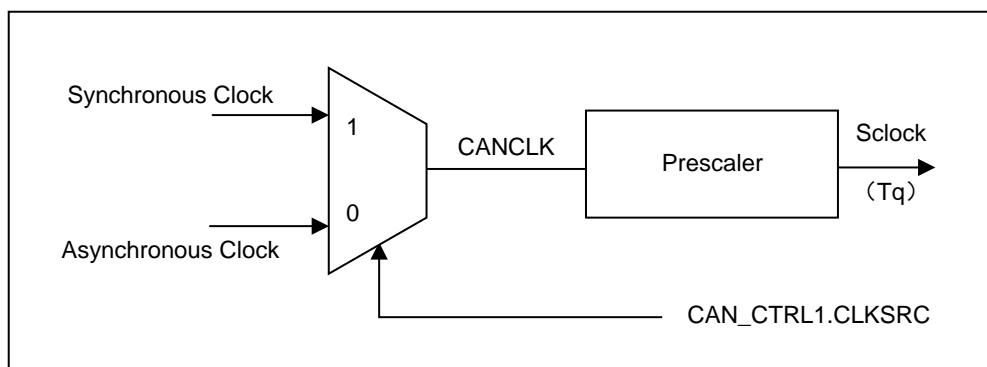


图 30-6 PE 时钟结构

当 CAN 总线时序需要一个严格的容差（高达 0.1%）时，应选择异步时钟，并配置异步时钟的时钟源为振荡器时钟。振荡器时钟比外设时钟具有更好的抖动性能。

FlexCAN 支持多种方式来设置 CAN 协议所要求的位时序参数。控制寄存器 1 (CAN_CTRL1) 有各种用于控制位时序参数的字段：PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW。CAN 位时序寄存器 (CAN_CBT) 扩展了 CAN_CTRL1 中 CAN 位时序变量的范围。CAN_FDCBT 提供了用于 BRS 置位的 CAN FD 帧数据段的位时序变量。

注：CAN FD 使能时，应始终置位 CAN_CBT.BTF 或 CAN_CTRL2.BTE，并在 CAN_CBT 中配置 CAN 位时序变量。

PRES DIV 字段（及其扩展范围 EPRES DIV 和用于 CAN FD 报文数据段的 FDPRES DIV）定义了串行时钟 (Sclock) 的预分频（见下列方程）。串行时钟的周期定义了用于构成 CAN 波形的时间单位 Tq (Time Quantum)。Tq 为 CAN 引擎所能处理的最小时间单元。

$$Tq = \frac{(PRES DIV + 1)}{f_{CANCLK}}$$

比特率定义了接收或传输 CAN 报文的速率，公式如下：

$$CAN \text{ Bit Time} = (\text{Number of Time Quanta in 1 bit time}) * Tq$$

$$Bit \text{ Rate} = \frac{1}{CAN \text{ Bit Time}}$$

位时间可以细分为三个部分：

- 同步段 (SYNC_SEG)：1Tq 的固定长度；信号边沿出现在该段内
- 时间段 1：包括 CAN 标准的传播段和相位段 1。该段可通过设置 CAN_CTRL1 寄存器的 PROPSEG 和 PSEG1 字段来编程，其总和 (+2) 为 2~16Tq。当 CAN_CBT.BTF 被置位时，FlexCAN 使用来自 CAN_CBT 寄存器的 EPROPSEG 和 EPSEG1 字段，其总和 (+2) 为 2~96Tq。对于 BRS 置位的 CAN FD 报文，FlexCAN 使用 CAN_FDCBT 寄存器的 FDPROPSEG 和 FDPSEG1 字段，其总和为 2~39Tq。
- 时间段 2：CAN 标准的相位段 2。该段可通过设置 CAN_CTRL1 寄存器的 PSEG2 字段来编程，其值 (+1) 为 2~8Tq。当 CAN_CBT.BTF 被置位时，FlexCAN 使用来自 CAN_CBT 寄存器的 EPSEG2 字段，其值 (+1) 为 2~32Tq。对于 BRS 置位的 CAN FD 报文，FlexCAN 使用 CAN_FDCBT 寄存器的 FDPSEG2 字段，其值 (+1) 为 2~8Tq。时间段 2 不能小于信息处理时间 (IPT)，IPT 在 FlexCAN 中为 2Tq。

注：上述时间段定义的位时间必须不小于 $5T_q$ 。对于位时间计算，使用 $2T_q$ 的信息处理时间（IPT），该值在 FlexCAN 中实现。

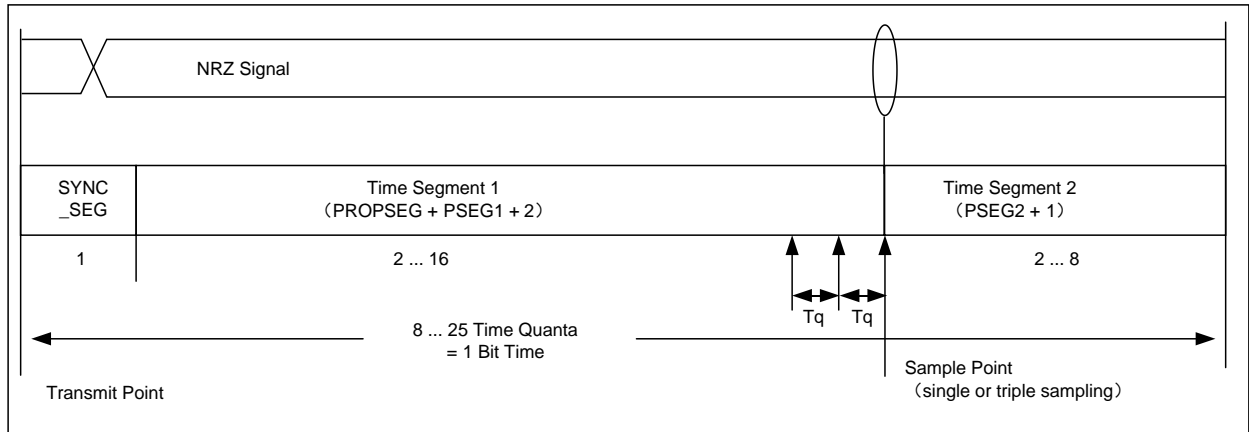


图 30-7 位时间内的段（使用 CAN_CTRL1 位时序变量的经典 CAN 格式）

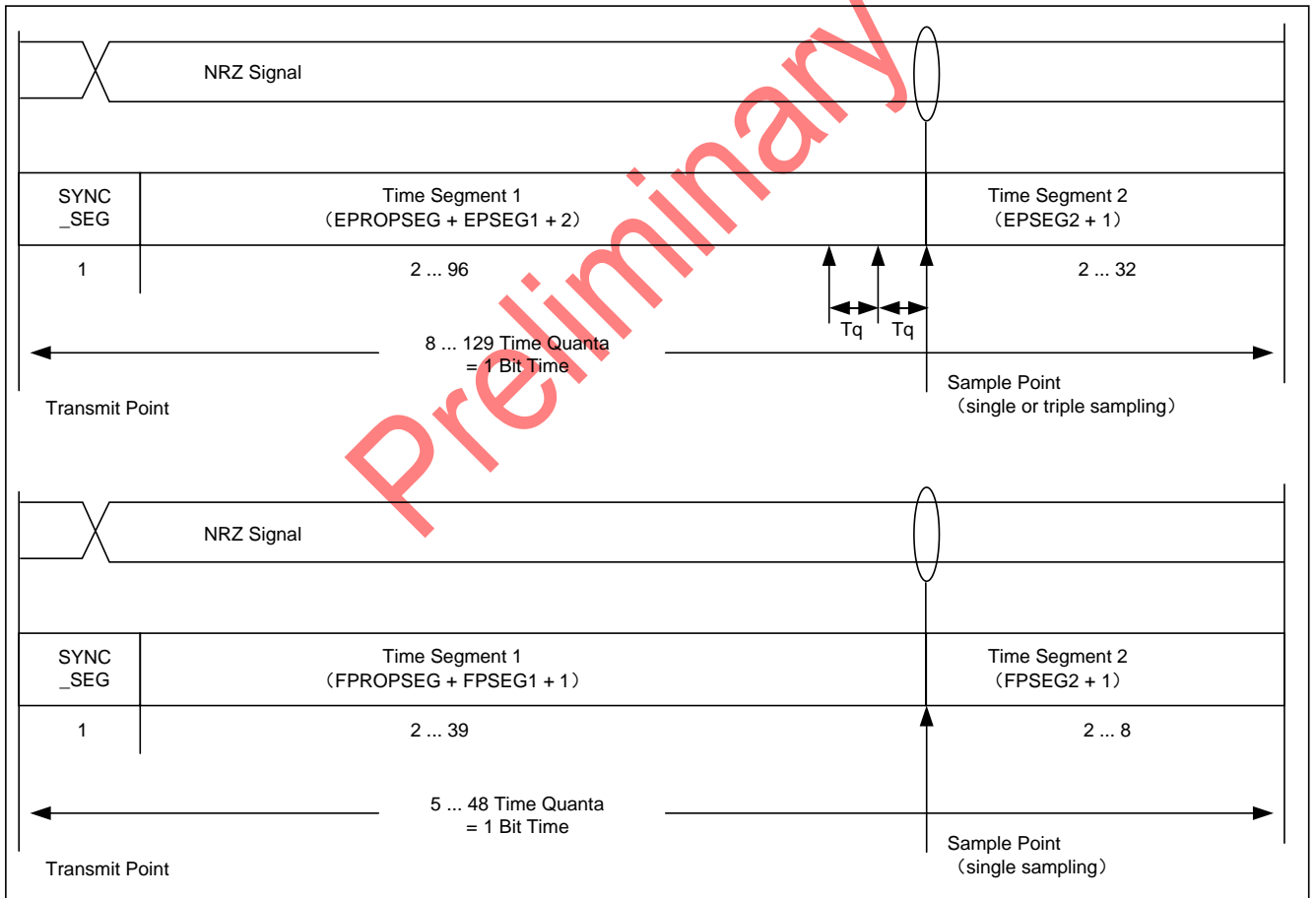


图 30-8 位时间内的段（使用 CAN FD 格式的 CAN_CBT 和 CAN_FDCBT 位时序变量）

表 30-13 时间段语法

语法	描述
SYNC_SEG	系统期望在此期间总线上发生转换
TSEG1	对应于 PROPSEG 和 PSEG1 的总和

语法	描述
TSEG2	对应于 PSEG2 的值
传输点	传输模式下的节点在该点传输新值到 CAN 总线上
采样点	节点在该点采样总线。如果选择每位采样三次，则该点位于第三次采样的位置

下表列举了一些经典 CAN 格式（Bosch CAN 2.0B）报文的时间段设置示例。

表 30-14 符合 Bosch CAN 2.0B 标准的位时间段设置

时间段 1	时间段 2	重同步补偿宽度
5 ~ 10	2	1 ~ 2
4 ~ 11	3	1 ~ 3
5 ~ 12	4	1 ~ 4
6 ~ 13	5	1 ~ 4
7 ~ 14	6	1 ~ 4
8 ~ 15	7	1 ~ 4
9 ~ 16	8	1 ~ 4

注：用户必须确保位时间的设置符合 CAN 协议标准（ISO 11898-1）。

当采用 CAN 位作为持续时间的衡量标准时（例如，评估报文中的 CAN 位事件），一个 CAN 位的外设时钟个数（NumClkBit）为：

$$\text{NUMClkBit} = \frac{f_{\text{SYS}}}{f_{\text{CANCLK}}} \times (\text{PRES DIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

其中：

- NumClkBit 为一个 CAN 位的外设时钟个数
- f_{CANCLK} 为 PE 时钟，单位 Hz。
- f_{SYS} 为系统（CHI）时钟频率，单位 Hz。
- PSEG1 为 CAN_CTRL1.PSEG1 的值
- PSEG2 为 CAN_CTRL1.PSEG2 的值
- PROPSEG 为 CAN_CTRL1.PROPSEG 的值
- PRES DIV 为 CAN_CTRL1.PRES DIV 的值

上述公式也适用于 CAN 位时序寄存器（CAN_CBT）所述的 CAN 位时间变量。

30.3.11.7 仲裁和匹配时序

在正常接收和传输期间，匹配、仲裁、移入和移出流程都在 CAN 帧的特定时间段被执行，如下图所示。

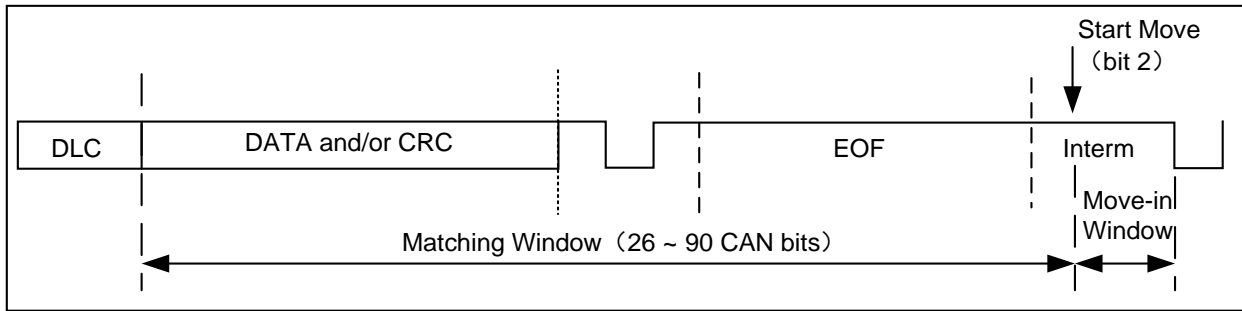


图 30-9 匹配和移入时间段

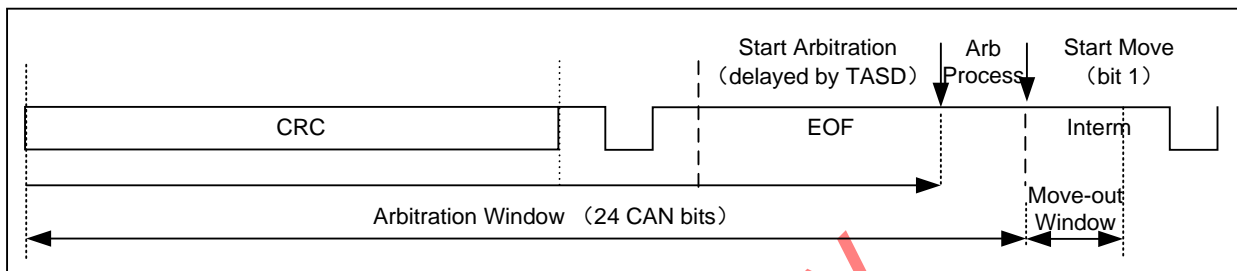


图 30-10 仲裁和移出时间段

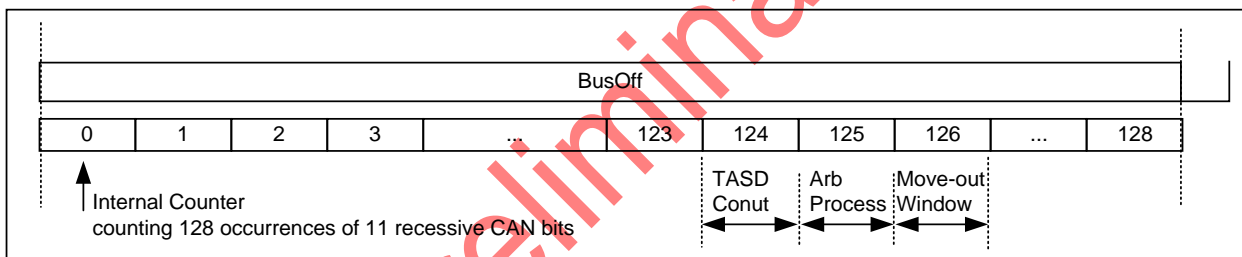


图 30-11 总线关闭结束时的仲裁和移出时间段

注：上图中，匹配和仲裁时序未考虑 CPU 或 FlexCAN 内部其他子模块同时访问内存所引起的延迟。

30.3.11.8 Tx 仲裁启动延迟

Tx 仲裁启动延迟 (CAN_CTRL2.TASD) 位字段，指示了 FlexCAN 从当前帧的 CRC 字段首位开始，延迟 Tx 仲裁过程起点所用的 CAN 位个数。CAN_CTRL2.TASD 只能在冻结模式下写入，其他模式下被硬件锁定。

CPU 在内部仲裁过程结束后重新配置传输 MB 的能力，影响了传输性能。如果仲裁在 Intermission 字段的首位前结束得过早，则 CPU 有机会重新配置部分 Tx MB，获胜 MB 不再是传输的最佳候选者。

TASD 可用于通过定义仲裁起点来优化传输性能，基于以下要素：

- 决定 CAN 位速率的 CAN 位时序变量
- 匹配和仲裁过程所采用的 MB 数量
- CHI 时钟（外设时钟）和 PE 时钟比

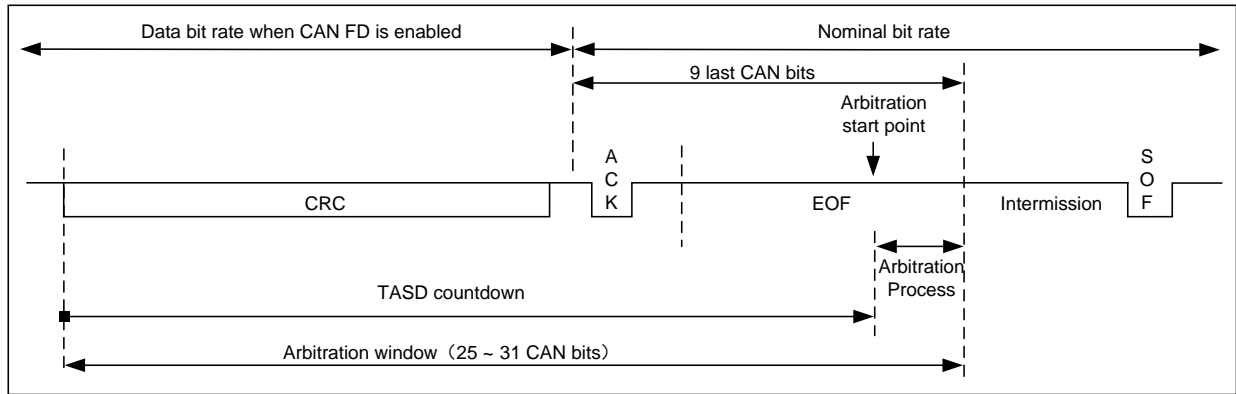


图 30-12 优化的 Tx 仲裁起点

从 CAN 位的角度看，仲裁过程的持续时间与可用的 MB 数量和 CAN 位速率成正比，与外设时钟频率成反比。

最优的仲裁时序，是在 CAN 帧 Intermission 字段首位前扫描最后一个 MB。例如，如果只有少量的 MB，且 CAN 波特率低，则仲裁可以更靠近帧的末端，增加起点的延迟，反之亦然。如果 T ASD 被设置为 0，则仲裁起点不延迟，且更多的时间会被预留给仲裁。此外，如果 T ASD 接近 24，则 CPU 可以稍后配置 Tx MB，并减少仲裁预留时间。如果为仲裁预留的时间太少，则 FlexCAN 可能无法及时找到获胜邮箱，并在最佳时机发送以赢得总线仲裁。

优化 T ASD 的值计算如下：

CAN FD 帧， $MAXMB + 1 \leq NMB_{END}$ 时：

$$TASD = 31 - \frac{2 \times (MAXMB + 1) + 4}{CPCB_N}$$

CAN FD 帧， $MAXMB + 1 > NMB_{END}$ 时：

$$TASD = 22 - \frac{2 \times (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

非 FD 帧：

$$TASD = 25 - \frac{2 \times (MAXMB + 1) + 4}{CPCB}$$

其中：

$$NMB_{END} = \frac{(9 \times CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRES DIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRES DIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

- MAXMB 为 CAN_MCR.MAXMB 的值

- NMB_{END} 为仲裁过程在帧结束时最后 9 个 CAN 位可以扫描到的 MB 数量，见上图
- $BITRATE_N$ 是由 CAN 标称位时间变量计算出的 CAN 位速率，单位 bps
- $BITRATE_F$ 是由 CAN 数据位时间变量计算出的 CAN 位速率，单位 bps
- $CPCB_N$ 是 CAN FD 帧标称比特率相应的每个 CAN 位的外设时钟数量
- $CPCB_F$ 是 CAN FD 帧数据比特率相应的每个 CAN 位的外设时钟数量
- $CPCB$ 是非 FD 帧每个 CAN 位的外设时钟数量
- f_{CANCLK} 为 PE 时钟，单位 Hz
- f_{SYS} 为外设时钟，单位 Hz
- EPSEG1 为 CAN_CBT.EPSEG1 的值（也可使用 CAN_CTRL1.PSEG1）
- EPSEG2 为 CAN_CBT.EPSEG2 的值（也可使用 CAN_CTRL1.PSEG2）
- EPROPSEG 为 CAN_CBT.EPROPSEG 的值（也可使用 CAN_CTRL1.PROPSEG）
- EPRESDIV 为 CAN_CBT.EPRESDIV 的值（也可使用 CAN_CTRL1.PRESDIV）

更多细节参见章节“协议时序”。

30.3.12 时钟限制

FlexCAN 支持 CHI 和 PE 子模块为同步或异步时钟：

- CHI 子模块时钟：由外设时钟供给
- PE 子模块时钟：时钟源可选（细节请参考 RCC 章节 RCC_CFGFR2 寄存器）。当 PE 时钟由振荡器时钟供给时，可在总线上实现非常低的抖动

当 CAN_CTRL1.CLKSRC 置位时，CHI 和 PE 子模块时钟都由外设时钟供给，发生同步操作。

当 CHI 和 PE 子模块时钟频率和/或相位不同时，CHI 时钟（外设时钟）频率必须始终大于 PE 时钟频率。

进行匹配和仲裁时，FlexCAN 需要在一个 CAN 帧的时间段内，扫描整个 MB 内存。该时间段由多个 CAN 位组成，为了获得充足的时间，CAN 位需要满足以下限制：

- CHI 时钟（外设时钟）频率不能低于 PE 时钟频率
- 每个 CAN 位需要满足最少外设时钟数的限制

表 30-15 经典 CAN 格式 CAN 位的最少外设时钟数

邮箱数量	CAN_MCR.RFEN	CAN_ERFCR.ERFEN	CAN 位的最少外设时钟数
32	0	0	16
32	1	0	17
32	0	1	19

经典 CAN 帧的比特率取决于 CAN 位的 T_q 数，可通过 CAN_CTRL1 或 CAN_CBT 寄存器进行调整。 T_q 的定义参见章节“协议时序”。每个 CAN 位的最少 T_q 数为 8。

对于 CAN FD 帧，有一些约束需要满足。每个标称比特率（NumClkNomBit）的 CAN 位的外设时钟

数可通过以下公式计算：

$$\text{NumClkNomBit} = \frac{f_{\text{SYS}}}{f_{\text{CANCLK}}} \times (\text{PRES DIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

$$= \frac{f_{\text{SYS}}}{\text{NomBitRate}}$$

其中 PRES DIV、PSEG1 和 PSEG2 为 CAN_CTRL1 寄存器的设定值。也可以使用 CAN_CBT 寄存器的 EPRES DIV、EPSEG1 和 EPSEG2 代替。

BRS 置位的 FD 帧中，数据段 CAN 位（快速 CAN 位）的个数取决于负载数据的字节数。数据字节数越少，快速 CAN 位个数 (NumOfFastBits) 就越少，FlexCAN 在内部匹配和仲裁过程中扫描整个 MB 内存的时间就越少。

表 30-16 CAN FD 帧中快速 CAN 位个数

最小数据字节数	DLC 字段	快速 CAN 位个数
0	0x0	21
1	0x1	29
2	0x2	37
3	0x3	45
4	0x4	53
5	0x5	61
6	0x6	69
7	0x7	77
8	0x8	85
12	0x9	117
16	0xA	149
20	0xB	186
24	0xC	218
32	0xD	282
48	0xE	410
64	0xF	538

CAN FD 帧数据段的比特率比仲裁段快，每个快速 CAN 位的最小外设时钟数 (MinNumClkFastBit) 可按照以下等式计算，以保证 FlexCAN 在接收和传输过程中有足够的时间扫描 MB 内存。

$$\text{MinNumClkFastBit}_A = \frac{(8.5 \times \text{MaxNumOfMB}) + [\text{ERFEN} \times (2 \times \text{NFE} + 4)] + 64 - (9 \times \text{NumClkNomBit})}{\text{NumOfFastBits}}$$

其中 MaxNumOfMB 为 CAN_MCR.MAXMB 定义的最大可用邮箱数。ERFEN 和 NFE 为 CAN_ERFCR 寄存器中的字段。

CHI 和 PE 子模块间的时钟域交叉电路，为保证握手机制在不丢失状态信息的情况下正常工作，要求每个快速 CAN 位的最小外设时钟数为：

$$\text{MinNumClkFastBit}_B = 3 \times \left(1 + \frac{f_{\text{SYS}}}{f_{\text{CANCLK}}} \right)$$

因此，每个快速 CAN 位的最小外设时钟数由以上两个计算值中较大的值决定：

$\text{MinNumClkFastBit} = \text{Maximum} (\text{MinNumClkFastBitA}, \text{MinNumClkFastBitB})$

CAN FD 帧数据段的最大 CAN 比特率 (DataBitRateMAX) 计算如下:

$$\text{DataBitRate}_{\text{MAX}} = \frac{f_{\text{CANCLK}}}{\text{ROUNDUP} \left(\frac{\text{MinNumClkFastBit} \times f_{\text{CANCLK}}}{f_{\text{SYS}}} \right)}$$

FlexCAN 在 CAN FD 模式下可达到的最大数据比特率, 受外设和振荡器时钟频率、最大邮箱数和预期的标称比特率影响。此外, 数据比特率取决于 FD 帧的最小有效负载大小。

根据时钟频率、有效负载大小和可用邮箱数的不同, CAN FD 的最大数据速率如下表所示。

表 30-17 CAN FD 帧数据段最大比特率 (增强型 Rx FIFO 禁止)

外设时钟频率 (MHz)	负载大小	可用邮箱数	最大数据速率 (Mbps)
40	0 ~ 64	32	6.667
50	0 ~ 64	32	6.667
60	0 ~ 64	32	8.0
75	0 ~ 64	32	8.0

30.3.13 复位

FlexCAN 支持三种复位方式:

- 芯片硬件复位: 异步复位所有内存映射的寄存器
- FlexCAN 软件复位 (CAN_MCR.SOFTTRST): 同步复位部分内存映射的寄存器, 参见表格“寄存器访问和复位信息”
- 芯片软件复位: 效果同 CAN_MCR.SOFTTRST

软件复位是同步的, 当软件复位待处理时, CAN_MCR.SOFTTRST 保持置位。因此, 轮询该位可得知复位何时完成。低功耗模式时钟关闭时不能应用软件复位, 复位前应退出低功耗模式并恢复时钟。

当 FlexCAN 处于模块禁止模式时, 应选择时钟源 (CAN_CTRL1.CLKSRC)。选择时钟源并使能 FlexCAN 时 (CAN_MCR.MDIS 位清零), FlexCAN 自动进入冻结模式。在冻结模式下, FlexCAN 未与 CAN 总线同步, CAN_MCR.HALT 和 CAN_MCR.FRZ 被置位, 内部状态机停止, 且 CAN_MCR.FRZACK 和 CAN_MCR.NOTRDY 位被置位。Tx 管脚为隐性, FlexCAN 不会启动任何 CAN 帧的传输或接收。注: MB 和 Rx 私有掩码寄存器不受复位影响, 因此不会自动被初始化。

30.3.14 初始化

任何配置的更改/初始化, 都要求 FlexCAN 必须进入冻结模式 (参见章节“冻结模式”)。以下为 FlexCAN 的通用初始化步骤:

- 1) 初始化模块配置寄存器 (CAN_MCR):
 - 置位 IRMQ, 使能各 MB 的私有过滤和接收队列功能
 - 置位 WRNEN, 使能警告中断

- 根据需要，置位 SRXDIS 以停止帧的自接收
- 置位 RFEN 或 ERFEN，使能 Rx FIFO
- 如果 Rx FIFO 已使能且需使用 DMA，则置位 DMA 位
- 置位 AEN 位，使能中止机制
- 置位 LPRIOEN 位，使能本地优先级功能

2) 初始化控制寄存器 1 (CAN_CTRL1)、CAN 位时序寄存器 (CAN_CBT) (可选)、CAN FD 位时序寄存器 (CAN_FDCBT)

- 确定位时序参数: PROPSEG、PSEG1、PSEG2 和 RJW (或 EPROPSEG、EPSEG1、EPSEG2 和 ERJW)
- 确定 CAN FD 位时序参数: FPROPSEG、FPSEG1、FPSEG2 和 FRJW
- 编程 PRES DIV (或 EPRES DIV) 字段以配置位速率
- 编程 FPRES DIV 字段以配置 CAN FD 位速率
- 确定内部仲裁模式 (LBUF 位)

3) 初始化报文缓冲区:

- 所有 MB 的 C/S 必须被初始化
- 如果 Rx FIFO 使能，则 ID 过滤表必须被初始化
- 各 MB 的其他条目应按要求进行初始化

4) 初始化 Rx 私有掩码寄存器 (CAN_RXIMRn)

5) 置位所需中断掩码位: CAN_CTRL1/CAN_CTRL2 (总线关闭和错误中断)、CAN_IMASKn (所有 MB 中断)

6) 清除 CAN_MCR.HALT 位

最后一个步骤后，FlexCAN 尝试同步至 CAN 总线。

30.3.15 中断

FlexCAN 支持很多中断源: 由 MB 引起的中断、总线关闭、总线关闭完成、错误、快速错误 (BRS 置位的 CAN FD 帧数据段中检测到的错误)、Tx 警告和 Rx 警告。

如果设置了相应的 IMASK 位，每个 MB 都可以作为中断来源。每个缓冲区在 CAN_IFLAG1 寄存器中都有指定的标志位，当对应的缓冲区成功完成传输时置位; 当 CPU 写 1 至该位时清零 (除非在同一时间产生了另一个中断)。

注: 必须保证 CPU 仅清零引起当前中断的位。因此，位操作指令 (BSET) 不能用于清除中断标志，这些指令可能会导致进入当前中断服务程序后设置的中断标志被意外清除。

传统型 Rx FIFO 使能 (CAN_MCR.RFEN = 1) 且 DMA 禁止 (CAN_MCR.DMA = 0) 时，对应于 MB0 ~ 7 的中断含义不同:

- CAN_IFLAG1 寄存器位 7 为“传统型 Rx FIFO 溢出”标志

- 位 6 为“传统型 Rx FIFO 警告”标志
- 位 5 为“传统型 Rx FIFO 中有可用帧”标志
- 位 4~0 未使用

参加章节“中断标志寄存器 1 (CAN_IFLAG1)”以获得更多信息。

如果传统型 Rx FIFO 和 DMA 都被使能 (CAN_MCR.RFEN=1 且 CAN_MCR.DMA=1), FlexCAN 不会生成任何传统型 Rx FIFO 中断。CAN_IFLAG1 寄存器位 5 仍表示“传统型 Rx FIFO 中有可用帧”, 并生成 DMA 请求。位 7、6 和 4~0 未使用。

注: 使能 CAN FD 时, 不可使用传统型 Rx FIFO。

对于组合中断 (多个 MB 中断源的“或”), 任何一个相关的 MB (或 FIFO) 产生中断时, 就会生成组合中断。此时, CPU 必须通过读取 CAN_IFLAG1 寄存器来确定造成中断的 MB (或 FIFO) 是哪一个。

总线关闭、总线关闭完成、错误、快速错误、Tx 警告和 Rx 警告中断, 可以从 CAN_ESR1 寄存器读取。总线关闭、错误、Tx 警告、Rx 警告中断的掩码位位于 CAN_CTRL1 寄存器。

30.4 寄存器/内存映射

CPU 访问 FlexCAN 寄存器需遵循下列规则:

- 无限制地读写管理员权限的寄存器 (下表中, 管理员模式下访问类型为 S/U 的寄存器; 或访问类型为 S 的寄存器) 会导致访问错误
- 读写预留地址空间会导致访问错误
- 对当前所有位都为只读的地址执行写操作会导致访问错误。如果至少有一个位不是只读, 则不会发生访问错误。某地址或其部分位的写权限会随操作模式或临时状态而改变, 细节请参考寄存器和位描述
- 如果 CAN_MCR.MAXMB 的值小于可用的 MB 数, 则可以将未使用的内存空间用作通用 RAM 空间

寄存器的访问类型可以是管理员 (S) 或无限制 (U), 大部分的寄存器都可通过编程 CAN_MCR.SUPV 配置为管理员或无限制访问。

表 30-18 寄存器访问和复位信息

寄存器	访问类型	硬件复位	软件复位
CAN_MCR 模块配置寄存器	S	是	是
CAN_CTRL1 控制寄存器 1	S/U	是	否
CAN_TIMER 自由运行计时器	S/U	是	是
CAN_RXMGMASK Rx 邮箱全局掩码寄存器	S/U	否	否
CAN_RX14MASK Rx 14 掩码寄存器	S/U	否	否
CAN_RX15MASK Rx 15 掩码寄存器	S/U	否	否
CAN_ECR 错误计数寄存器	S/U	是	是
CAN_ESR1 错误和状态寄存器 1	S/U	是	是
CAN_IMASK1 中断掩码寄存器 1	S/U	是	是

寄存器	访问类型	硬件复位	软件复位
CAN_IFLAG1 中断标志寄存器 1	S/U	是	是
CAN_CTRL2 控制寄存器 2	S/U	是	否
CAN_ESR2 错误和状态寄存器 2	S/U	是	是
CAN_CRCCR CRC 寄存器	S/U	是	是
CAN_RXFGMASK 传统型 Rx FIFO 全局掩码寄存器	S/U	否	否
CAN_RXFIR 传统型 Rx FIFO 信息寄存器	S/U	否	否
CAN_CBT CAN 位时序寄存器	S/U	是	否
报文缓冲区	S/U	否	否
CAN_RXIMRn Rx 私有掩码寄存器	S/U	否	否
CAN_FDCTRL CAN FD 控制寄存器	S/U	是	否
CAN_FDCBT CAN FD 位时序寄存器	S/U	是	否
CAN_FDCRC CAN FD CRC 寄存器	S/U	是	是
CAN_ERFCR 增强型 Rx FIFO 控制寄存器	S/U	是	是
CAN_ERFIER 增强型 Rx FIFO 中断使能寄存器	S/U	是	是
CAN_ERFSR 增强型 Rx FIFO 增强型 Rx FIFO 状态寄存器	S/U	是	是
CAN_ERFFELn 增强型 Rx FIFO 过滤元素寄存器	S/U	否	否

30.4.1 寄存器/内存映射总览

FlexCAN 可使用报文缓冲区和 Rx FIFO 结构存储 CAN 报文，用于传输与接收。RAM 空间从偏移地址 0x80 开始。

下表为 FlexCAN 的内存映射，包括 32 个 128 位的报文缓冲区（MB），占据了偏移地址 0x80 至 0x27F。偏移地址 0x2000 到 0x204C 为增强型 Rx FIFO 的输出，参见章节“增强型 Rx FIFO 结构”。

表 30-19 FlexCAN 寄存器/内存映射概览

偏移地址	寄存器	访问权限	复位值
00h	CAN_MCR 模块配置寄存器	R/W	D880_000Fh
04h	CAN_CTRL1 控制寄存器 1	R/W	0000_0000h
08h	CAN_TIMER 自由运行计时器	R/W	0000_0000h
10h	CAN_RXMGMASK Rx 邮箱全局掩码寄存器	R/W	自定义
14h	CAN_RX14MASK Rx 14 掩码寄存器	R/W	自定义
18h	CAN_RX15MASK Rx 15 掩码寄存器	R/W	自定义
1Ch	CAN_ECR 错误计数寄存器	R/W	0000_0000h
20h	CAN_ESR1 错误和状态寄存器 1	R/W	0000_0000h
28h	CAN_IMASK1 中断掩码寄存器 1	R/W	0000_0000h
30h	CAN_IFLAG1 中断标志寄存器 1	R/W	0000_0000h
34h	CAN_CTRL2 控制寄存器 2	R/W	00A0_0000h
38h	CAN_ESR2 错误和状态寄存器 2	R/W	0000_0000h
44h	CAN_CRCCR CRC 寄存器	R	0000_0000h
48h	CAN_RXFGMASK 传统型 Rx FIFO 全局掩码寄存器	R/W	自定义
4Ch	CAN_RXFIR 传统型 Rx FIFO 信息寄存器	R	自定义

偏移地址	寄存器	访问权限	复位值
50h	CAN_CBT CAN 位时序寄存器	R/W	0000_0000h
880h ~ 8FCh	CAN_RXIMRn Rx 私有掩码寄存器 (n = 0 ~ 31)	R/W	自定义
C00h	CAN_FDCTRL CAN FD 控制寄存器	R/W	8000_0100h
C04h	CAN_FDCBT CAN FD 位时序寄存器	R/W	0000_0000h
C08h	CAN_FDCRC CAN FD CRC 寄存器	R	0000_0000h
C0Ch	CAN_ERFCR 增强型 Rx FIFO 控制寄存器	R/W	0000_0000h
C10h	CAN_ERFIER 增强型 Rx FIFO 中断使能寄存器	R/W	0000_0000h
C14h	CAN_ERFSR 增强型 Rx FIFO 状态寄存器	R/W	0000_0000h
3000h ~ 3004h	CAN_ERFFELn 增强型 Rx FIFO 过滤元素寄存器 (n=0 ~ 1)	R/W	自定义

30.4.2 CAN_MCR 模块配置寄存器

偏移地址: 0x00

复位值: 0xD880 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MDIS	FRZ	RFEN	HALT	NOTRD Y	Res.	SOFT R ST	FRZAC K	SUPV	Res.	WRNEN	LPMAC K	Res.		SRXDIS	IRMQ
rw	rw	rw	rw	r		rw	r	rw		rw	r			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA	Res.	LPRIOE N	AEN	FDEN	Res.	IDAM		Res.	MAXMB						
rw		rw	rw	rw		rw			rw						

字段	名称	描述
31	MDIS	1: 禁止 FlexCAN, 停止 PE 和 CHI 子模块的时钟 0: 使能 FlexCAN 该位不受软件复位影响。
30	FRZ	1: 使能冻结模式 0: 禁止冻结模式 当 CAN_MCR.HALT 置位或芯片请求调试模式时, FRZ 用于指定 FlexCAN 的行为。FRZ 被置位时, FlexCAN 将会进入冻结模式; 该位置 0 将使 FlexCAN 退出冻结模式。
29	RFEN	1: 使能传统型 Rx FIFO 0: 禁止传统型 Rx FIFO 置位 RFEN 时, MB0 ~ 5 不能用于正常的接收和传输, 因为相应的内存空间(0x80 ~ 0xDC) 被 FIFO 引擎使用。额外的 MB (最多 26 个, 取决于 CAN_CTRL2.RFFN 的设置) 用于传统型 Rx FIFO ID 过滤表元素。 RFEN 只能在冻结模式写入。 注: CAN FD 使能时, 不能置位设置 RFEN (见 FDEN 位描述)。

字段	名称	描述
28	HALT	<p>1: 如果 FRZ 被置位, 则进入冻结模式</p> <p>0: 没有冻结模式请求</p> <p>该位置位会使 FlexCAN 进入冻结模式。CPU 在初始化报文缓冲区和控制寄存器 (CAN_CTRL1 和 CAN_CTRL2) 后应将该位清零。FlexCAN 在 HALT 清零前不会进行接收或传输。模块处于低功耗模式时, 不能进入冻结模式。</p>
27	NOTRDY	<p>1: FlexCAN 处于禁止或冻结模式</p> <p>0: FlexCAN 处于正常、只听或回环模式</p> <p>该位只读, 在 FlexCAN 退出禁止或冻结模式后清零, 不受软件复位影响。</p>
26	Reserved	保留, 必须保持复位值
25	SOFTRST	<p>软件复位:</p> <p>1: 复位受软件复位影响的寄存器</p> <p>0: 无复位请求</p> <p>该位置位时, FlexCAN 复位内部状态机和部分内存映射寄存器。</p> <p>该位可直接由 CPU 写 CAM_MCR 置位; 当在芯片级请求全局软件复位时, 也会被置位。</p> <p>如果复位没有完成, SOFTRST 将保持置位, 完成复位后自动清零。因此软件可以轮询该位以判断软件复位是否完成。</p> <p>在低功耗模式下, 如果时钟被关闭, 则软件复位将不起作用。FlexCAN 应该首先退出低功耗模式, 然后软件复位才会起作用。</p> <p>该位不受软件复位影响。</p>
24	FRZACK	<p>冻结模式应答:</p> <p>1: FlexCAN 处于冻结模式, 预分频器停止</p> <p>0: FlexCAN 未处于冻结模式, 预分频器运行</p> <p>该位只读。直到当前传输或者接收过程完成, 冻结模式请求才会被执行。因此软件可通过轮询该位以了解 FlexCAN 是否已确切地进入冻结模式。如果冻结模式请求被取消, 则一旦 FlexCAN 模块预分频器又重新运行, 该位被清零。如果 FlexCAN 在低功耗模式时请求冻结模式, 则只有当模块退出低功耗模式时 FRZACK 位才会被置位。参见章节“冻结模式”。</p> <p>该位不受软件复位影响。</p>
23	SUPV	<p>1: FlexCAN 处于管理员模式, 受影响的寄存器只允许管理员访问</p> <p>0: FlexCAN 处于用户模式, 受影响的寄存器允许管理员和无限制访问</p> <p>受 SUPV 影响的寄存器在表格“寄存器访问和复位信息”的访问类型列被标记为 S/U。</p> <p>SUPV 复位值为 1, 因此受影响的寄存器只能从管理员访问开始。该位只能在冻结模式下写入, 其他模式下被硬件锁定。</p>
22	Reserved	保留, 必须保持复位值

字段	名称	描述
21	WRNEN	警告中断使能： 1：当相应的错误计数器从小于 96 上升至大于等于 96 时，TWRNINT 和 RWRNINT 位被置位 0：TWRNINT 和 RWRNINT 位为零，与错误计数器的值无关 WRNEN 被置位时，错误和状态寄存器 1（CAN_ESR1）产生 TWRNINT 和 RWRNINT 标志。如果 WRNEN 被清零，则 TWRNINT 和 RWRNINT 标志将一直为零，与错误计数器的值无关，且不会生成警告中断。 WRNEN 只能在冻结模式下写入，其他模式下被硬件锁定。
20	LPMACK	低功耗模式应答： 1：FlexCAN 模块处于低功耗模式 0：FlexCAN 模块未处于低功耗模式 该位只读。直到所有当前传输或接收流程已完成，FlexCAN 才能进入低功耗模式。因此，CPU 可通过轮询该位以了解 FlexCAN 是否已确切地进入低功耗模式。 LPMACK 不受软件复位影响。
19: 18	Reserved	保留，必须保持复位值
17	SRXDIS	1：禁止自接收 0：使能自接收 该位控制 FlexCAN 是否接收自身传输的帧。如果 SRXDIS 被置位，由 FlexCAN 传输的帧不会存储在任 MB 中，无论 MB 的 ID 是否与帧匹配，且接收帧时不会生成中断标志或中断信号。 SRXDIS 只能在冻结模式下写入，其他模式下被硬件锁定。
16	IRMQ	1：使能私有 Rx 掩码和队列功能 0：禁止私有 Rx 掩码和队列功能 IRMQ 控制 Rx 匹配过程是基于私有掩码和队列功能或基于 CAN_RXMGMASK、CAN_RX14MASK、CAN_RX15MASK 和 CAN_RXFGMASK 构成的掩码机制。 IRMQ 只能在冻结模式下写入，其他模式下被硬件锁定。
15	DMA	1：使能 Rx FIFO DMA 功能 0：禁止 Rx FIFO DMA 功能 DMA 功能只能在 Rx FIFO 下使用，因此，CAN_MCR.RFEN 或 CAN_ERFCR.ERFEN 必须被置位。置位了 DMA 和 RFEN 时，CAN_IFLAG1.BUF5I 会生成 DMA 请求，而不是生成 Rx FIFO 中断。 DMA 位只能在冻结模式下写入，其他模式下被硬件锁定。
14	Reserved	保留，必须保持复位值
13	LPRIOEN	1：使能本地优先级 0：禁止本地优先级 本地优先级用于在仲裁期间扩展 ID，仲裁过程基于完整的 32 位字完成。但实际传输的 ID 仍为标准帧的 11 位、扩展帧的 29 位 ID。 LPRIOEN 只能在冻结模式下写入，其他模式下被硬件锁定。

字段	名称	描述
12	AEN	1: 使能中止 0: 禁止中止 Tx 中止机制确保安全地中止未完成的传输, 因此在没有通知的情况下 CAN 总线上不会有帧传输, 参见章节“传输中止机制”。 AEN 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: 当 AEN 被置位时, 中止机制只能用于更新传输邮箱, 将中止代码写入 Rx 邮箱会导致不可预测的结果。
11	FDEN	1: CAN FD 使能。FlexCAN 能够以 CAN FD 和 CAN 2.0 两种格式收发报文 0: CAN FD 禁止。FlexCAN 能够以 CAN 2.0 格式收发报文 FDEN 只能在冻结模式下写入。 注: 如果 FDEN 置位, 则不能使能传统型 Rx FIFO (RFEN)。
10	Reserved	保留, 必须保持复位值
9: 8	IDAM	传统型 Rx FIFO ID 过滤表元素格式: 00: 格式 A — 每个 ID 过滤表元素有一个完整的 ID (标准和扩展) 01: 格式 B — 每个 ID 过滤表元素都有两个完整的标准 ID 或两个不完整的 14 位扩展 ID 10: 格式 C — 每个 ID 过滤表元素有四个不完整的 8 位标准 ID 11: 格式 D — 拒绝所有帧 所有表元素同时被 IDAM 配置 (采用相同格式), 参见章节“传统型 Rx FIFO 结构”。 IDAM 只能在冻结模式下写入, 其他模式下被硬件锁定。
7	Reserved	保留, 必须保持复位值
6: 0	MAXMB	最后一个将参与匹配和仲裁过程的 MB 编号 复位值为 0x0F, 即配置了 16 个报文缓冲区。 MAXMB 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: MAXMB 必须小于等于可用 MB 的数量。此外, MAXMB 的值必须考虑传统型 Rx FIFO 占用的 MB 以及由 CAN_CTRL2.RFFN 定义的 ID 过滤表空间。

30.4.3 CAN_CTRL1 控制寄存器 1

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
PRESDIV								RJW		PSEG1			PSEG2		
rw								rw		rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOFFM SK	ERRMS K	CLKSR C	LPB	TWRNM SK	RWRN MSK	Res.		SMP	BOFFR EC	TSYN	LBUF	LOM	PROPSEG		
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

字段	名称	描述
31: 24	PRES DIV	<p>预分频器分频系数 — 定义 PE 时钟频率与串行时钟 (Sclock) 频率之间的比率</p> $\text{Sclock 频率} = \text{PE 时钟频率} / (\text{PRES DIV} + 1)$ <p>Sclock 周期定义了 CAN 协议的 Tq。对于复位值, Sclock 频率等于 PE 时钟频率。PRES DIV 的最大值为 0xFF, 因此 Sclock 的最小时钟频率为 PE 时钟频率除以 256, 参见章节“协议时序”。</p> <p>PRES DIV 只能在冻结模式下写入, 其他模式下被硬件锁定。</p>
23: 22	RJW	<p>重同步补偿 — 定义了一次重同步补偿的最大 Tq 数</p> $\text{重同步补偿宽度} = \text{RJW} + 1$ <p>RJW 有效值为 0 ~ 3, 只能在冻结模式下写入, 其他模式下被硬件锁定。</p>
21: 19	PSEG1	<p>相位段 1 — 定义位时间 (bit time) 中相位段 1 的长度</p> $\text{相位段 1} = (\text{PSEG1} + 1) \times \text{Tq}$ <p>PSEG1 有效值为 0 ~ 7, 只能在冻结模式下写入, 其他模式下被硬件锁定。</p>
18: 16	PSEG2	<p>相位段 2 — 定义位时间中相位段 2 的长度</p> $\text{相位段 2} = (\text{PSEG2} + 1) \times \text{Tq}$ <p>PSEG2 有效值为 0 ~ 7, 只能在冻结模式下写入, 其他模式下被硬件锁定。</p>
15	BOFFMSK	<p>总线关闭中断 (CAN_ESR1.BOFFINT) 掩码:</p> <p>1: 总线关闭中断使能</p> <p>0: 总线关闭中断禁止</p>
14	ERRMSK	<p>错误中断 (CAN_ESR1.ERRINT) 掩码:</p> <p>1: 错误中断使能</p> <p>0: 错误中断禁止</p>
13	CLKSRC	<p>FlexCAN 通信时钟源选择:</p> <p>1: 同步时钟</p> <p>0: 异步时钟</p> <p>只能在模块禁止模式下写入, 其他模式下被硬件锁定。</p> <p>注: 通信时该位必须置 1。</p>
12	LPB	<p>1: 回环模式使能</p> <p>0: 回环模式禁止</p> <p>在回环模式下, FlexCAN 进行内部循环用于自检。发送器的输出从内部反馈到接收器的输入。Rx CAN 输入引脚被忽略, Tx CAN 输出处于隐性状态 (逻辑 1)。发送报文时, FlexCAN 和正常传输时一样, 将自身传输的报文看作从远程节点接收到的报文。</p> <p>在回环模式下, FlexCAN 忽略 CAN 帧应答字段的 ACK, 生成内部应答位以确保能正确接收自身发送的报文, 并产生发送和接收中断。</p> <p>LPB 只能在冻结模式下写入, 其他模式下被硬件锁定。</p> <p>注: 回环模式下, CAN_MCR.SRXDIS 不能被置位, 否则会妨碍自接收。</p>
11	TWRNMSK	<p>Tx 警告中断 (CAN_ESR1.TWRNINT) 掩码:</p> <p>1: 警告中断使能</p> <p>0: 警告中断禁止</p> <p>当 CAN_MCR.WRNEN 为 0 时, 该位读数为 0。</p> <p>只有 CAN_MCR.WRNEN 被置位时, 才能写入该位。</p>

字段	名称	描述
10	RWRNMSK	Rx 警告中断 (CAN_ESR1.RWRNINT) 掩码: 1: 警告中断使能 0: 警告中断禁止 当 CAN_MCR.WRNEN 为 0 时, 该位读出为 0。 只有 CAN_MCR.WRNEN 被置位时, 才能写入该位。
9: 8	Reserved	保留, 必须保持复位值
7	SMP	Rx 输入的 CAN 位采样模式: 0: 仅采样一次来决定 CAN 位的值 1: 采样三次来决定 CAN 位的值。一次常规采样 (采样点) 和 2 次之前的采样, 使用多数决规则。 SMP 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: 为确保正确运行, 置位 SMP 时, 必须保证 CAN_CTRL1.PSEG1 (或 CAN_CBT.EPSEG1) 最小 2Tq。CAN FD 使能 (CAN_MCR.FDEN = 1) 时, SMP 不能被置位。
6	BOFFREC	FlexCAN 如何从总线关闭状态中恢复: 0: 自动从总线关闭状态恢复使能 1: 自动从总线关闭状态恢复禁止 如果 BOFFREC 为 0, 则根据 CAN 2.0B 自动从总线关闭状态中恢复; 如果 BOFFREC 被置位, 则禁止自动恢复, FlexCAN 保持在总线关闭状态, 直到 BOFFREC 被软件清零。如果清零发生在 CAN 总线上检测到 128 次连续的 11 个隐性位之前, 则总线关闭恢复, 如同 BOFFREC 从未被置位一样; 如果清零发生在 128 次 11 个隐性位之后, 则 FlexCAN 通过等待 11 个隐性位来重新同步总线。 该位清零后, 在总线关闭期间可再次被重新置位, 但只有在 FlexCAN 下一次进入总线关闭状态时才会生效。如果 BOFFREC 在 FlexCAN 进入总线关闭时被清零, 总线关闭期间置位 BOFFREC 不会影响当前的总线关闭恢复。
5	TSYN	定时器同步 — MB0 每接收到一次报文就复位自由运行计时器的机制: 1: 定时器同步使能 0: 定时器同步禁止 定时器同步功能为同步多个 FlexCAN 站点提供特定的 SYNC 报文 (全局网络时间)。如果 CAN_MCR.RFEN 被置位 (传统型 Rx FIFO 使能), 则第一个可用的邮箱 (通过 CAN_CTRL2.RFFN 设置) 取代 MB0, 被用于定时器同步。 TSYN 只能在冻结模式下写入, 其他模式下被硬件锁定。
4	LBUF	报文缓冲区传输的顺序: 1: 最小编号 MB 优先传输 0: 最高优先级 MB 优先传输 当 LBUF 被置位时, CAN_MCR.LPRIOEN 位不影响优先级仲裁。 LBUF 只能在冻结模式下写入, 其他模式下被硬件锁定。

字段	名称	描述
3	LOM	<p>配置 FlexCAN 在只听模式下运行： 1: FlexCAN 在只听模式下运行 0: 只听模式已停用</p> <p>在只听模式下，传输停止，CAN_ECR 描述的所有错误计数器都被冻结，且 FlexCAN 在 CAN 错误被动模式下运行。只有被其他 CAN 节点应答了的报文才可以被接收。如果 FlexCAN 检测到一个尚未被应答的报文，则标记为一个 BIT0 错误，而不改变接收错误计数器 (CAN_ECR.RXERRCNT)，如同试图应答报文一样。</p> <p>只听模式由指示被动错误的 CAN_ESR1.FLTCONF 字段的状态应答，请求和应答之间可能有一些延迟。</p> <p>LOM 只能在冻结模式下写入，其他模式下被硬件锁定。</p>
2: 0	PROPSEG	<p>传播时间段 — 定义位时间中传播段的长度 传播时间 = (PROPSEG + 1) × Tq PROPSEG 有效值为 0~7，只能在冻结模式下写入，其他模式下被硬件锁定。</p>

注 1: CAN 位时序变量 (PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW) 还可以在 CAN_CBT 寄存器中配置，该寄存器扩展了所有相关变量的范围。如果 CAN_CBT.BTF 被置位，则 CAN_CTRL1 中上述位时序变量字段变为只读。

注 2: CAN_CTRL1 和 CAN_CBT 的 CAN 位时序变量存储在同一个寄存器内。

注 3: CAN FD 使能时，不使用 CAN_CTRL1 的位时序变量，而是使用 CAN_CBT 寄存器的 EPRES DIV、ERJW、EPSEG1、EPSEG2 和 EPROPSEG 字段。

30.4.4 CAN_TIMER 自由运行计数器寄存器

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

字段	名称	描述
31: 16	Reserved	保留，必须保持复位值
15: 0	TIMER	计时器值：包含自由运行计数器的值

CAN_TIMER 寄存器代表一个 16 位自由运行的计数器，可通过 CPU 进行读写。该计数器复位值为 0x0，计数到 0xFFFF 后返回，重新开始计数。

计时器通过 CAN 位时钟（位时钟决定 CAN 总线波特率）来计数。报文传输/接收期间，每接收或传输一次，计时器加一；总线上没有报文时，计时器使用先前编程的波特率来计数；在冻结、禁止模式下计

时器不增加。

当任何帧的 ID 段的第二位在 CAN 总线上时，计时器的值被捕获。捕获的值在报文成功被接收或传输后，被写入报文缓冲区的时间戳字段。

如果 CAN_CTRL1.TSYN 被置位，则每当第一个可用邮箱接收到报文时（根据 CAN_CTRL2.RFFN 的设置）计时器将会被复位。

CPU 可以在任何时候写该寄存器，但如果写操作和第一个邮箱的接收导致计时器复位同时发生，那么写入的值将被丢弃。

30.4.5 CAN_RXMGMASK Rx 邮箱全局掩码寄存器

CAN_RXMGMASK 寄存器位于 RAM 空间，用于掩码所有 Rx MB 的过滤字段，不包括具有私有掩码寄存器的 MB14 ~ 15。

CAN_MCR.IRMQ 为 0 时，CAN_RXMGMASK 有效；CAN_MCR.IRMQ 被置位时，CAN_RXMGMASK 无效。

偏移地址：0x10

复位值：x

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

字段	名称	描述
31: 0	MG	<p>Rx 邮箱全局掩码位：</p> <p>0: 过滤器相应位“don't care”</p> <p>1: 过滤器相应位被检查</p> <p>只能在冻结模式下写入，其他模式下被硬件锁定。</p> <p>注：MG 与邮箱 ID 的对齐并不完美，因为最高两个 MG 位会影响 RTR 和 IDE 字段（位于邮箱的 C/S）。</p> <p>下表详细介绍了 MG 位是如何掩码各邮箱过滤字段的。</p>

表 30-20 MG 位掩码邮箱过滤字段

SMB.RTR ⁽¹⁾	CAN_CTRL2. RRS	CAN_CTRL2. EACEN	邮箱过滤字段			
			MB.RTR	MB.IDE	MB.ID	预留
0	—	0	注 ⁽²⁾	注 ⁽³⁾	MG[28: 0]	MG[31: 29]
0	—	1	MG[31]	MG[30]	MG[28: 0]	MG[29]
1	0	—	—	—	—	MG[31: 0]
1	1	0	—	—	MG[28: 0]	MG[31: 29]
1	1	1	MG[31]	MG[30]	MG[28: 0]	MG[29]

- 1) 接收帧的 RTR 位, 被存放在 Rx SMB 中
- 2) 如果 CAN_CTRL2.EACEN 为 0, 则邮箱的 RTR 位永远不会和接收帧的 RTR 位比较
- 3) 如果 CAN_CTRL2.EACEN 为 0, 则邮箱的 IDE 位总是会和接收帧的 IDE 位比较

30.4.6 CAN_RX14MASK Rx 14 掩码寄存器

CAN_RX14MASK 寄存器位于 RAM 空间, 用于掩码 MB14 的过滤字段。

CAN_MCR.IRMQ 被置位时, CAN_RX14MASK 无效。

偏移地址: 0x14

复位值: x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX14M[31: 16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX14M[15: 0]															
rw															

字段	名称	描述
31: 0	RX14M	Rx MB14 掩码位: 0: 过滤器相应位“don't care” 1: 过滤器相应位被检查 每位用作 MB14 相应过滤字段的掩码, 与 CAN_RXMGMASK 的设置方式相同。 只能在冻结模式下写入, 其他模式下被硬件锁定。

30.4.7 CAN_RX15MASK Rx 15 掩码寄存器

CAN_RX15MASK 寄存器位于 RAM 空间, 用于掩码 MB15 的过滤字段。

CAN_MCR.IRMQ 被置位时, CAN_RX15MASK 无效。

偏移地址: 0x18

复位值: x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX15M[31: 16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX15M[15: 0]															
rw															

字段	名称	描述
31: 0	RX15M	<p>Rx MB15 掩码位:</p> <p>0: 过滤器相应位“无关”</p> <p>1: 过滤器相应位被检查</p> <p>每位用作 MB15 相应过滤字段的掩码, 与 CAN_RXMGMASK 的设置方式相同。</p> <p>只能在冻结模式下写入, 其他模式下被硬件锁定。</p>

30.4.8 CAN_ECR 错误计数器寄存器

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
RXERRCNTFAST								TXERRCNTFAST							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXERRCNT								TXERRCNT							
rw								rw							

字段	名称	描述
31: 24	RXERRCNTFAST	<p>快速位接收错误计数器 — 计算在接收到的 BRS 置位的 CAN FD 报文数据段中检测到的所有错误</p> <p>非冻结模式下, RXERRCNTFAST 计数器为只读</p>
23: 16	TXERRCNTFAST	<p>快速位传输错误计数器 — 计算在传输的 BRS 置位的 CAN FD 报文数据段中检测到的所有错误</p> <p>非冻结模式下, TXERRCNTFAST 计数器为只读</p>
15: 8	RXERRCNT	<p>接收错误计数器 — 计算在接收到的报文中检测到的所有错误</p> <p>非冻结模式下, RXERRCNT 计数器为只读</p>
7: 0	TXERRCNT	<p>传输错误计数器 — 计算在传输的报文中检测到的所有错误</p> <p>非冻结模式下, TXERRCNT 计数器为只读</p>

TXERRCNT 和 RXERRCNT 计数器统计 CAN FD 和非 FD 报文中的所有错误; TXERRCNTFAST 和 RXERRCNTFAST 用于统计 BRS 置位的 CAN FD 帧数据段所发生的错误。

故障界定状态 (CAN_ESR1.FLTCONF) 基于 TXERRCNT 和 RXERRCNT 计数器更新。TXERRCNT 和 RXERRCNT 计数器的自加和自减规则见 CAN 协议。

FlexCAN 总线状态转换的基本原则:

- 如果 TXERRCNT 或 RXERRCNT 增加至 ≥ 128 , 则 CAN_ESR1.FLTCONF 字段将会更新, 以反映错误被动状态
- 如果 FlexCAN 为错误被动状态, TXERRCNT 或 RXERRCNT 自减至 ≤ 127 , 且另一个计数器已满足该条件, 则 CAN_ESR1.FLTCONF 字段将会更新, 以反映错误主动状态

- 如果 TXERRCNT 增加至 >255，则 CAN_ESR1.FLTCONF 字段将会更新，以反映总线关闭状态，且可能会产生中断。随后 TXERRCNT 复位为 0
- 如果 FlexCAN 为总线关闭状态，TXERRCNT 与另一个内部计数器级联计数，计数到总线上出现 128 次 11 个连续隐性位时，TXERRCNT 复位为 0。计数的方式为，内部计数器计数 11 位后重新计数；同时 TXERRCNT 增加。当 TXERRCNT 增加至 128 时，CAN_ESR1.FLTCONF 字段将会更新为错误主动状态，且两个错误计数器都复位为 0。在少于 11 个连续隐性位之后出现显性位时，内部计数器复位为 0，且不影响 TXERRCNT 的值
- 如果系统启动期间，只有一个节点在运行，由于应答错误 (CAN_ESR1.ACKERR)，TXERRCNT 在试图发送的每一个报文中增加。转换为错误被动状态后，TXERRCNT 不再因应答错误而增加。因此，设备不会进入总线关闭状态
- RXERRCNT 增加至 127 以上后不会再增加，即使接收器检测到更多的错误。在下次成功接收报文时，将计数器的值设置在 119 至 127 之间，以恢复错误主动状态
- TXERRCNTFAST 和 RXERRCNTFAST 计数值的增减取决于在 BRS 置位的 CAN FD 帧数据段中检测到的错误，遵循与 TXERRCNT 和 RXERRCNT 计数器相同的增减规则，并在计数至最大值 255 后停止。当 FlexCAN 处于总线关闭状态时，计数停止并保持计数值不变；当 FlexCAN 离开总线关闭状态时，计数器会被重置；当 FlexCAN 返回至错误主动状态时，计数器将重新开始计数

30.4.9 CAN_ESR1 错误和状态寄存器 1

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
BIT1ERRFAST	BIT0ERRFAST	Res.	CRCERRFAST	FRMERRFAST	STFERFAST	Res.				ERRR	ERRINTFAST	BOFFDONEINT	SYNCH	TWRNINT	RWRNINT
r	r		r	r	r					rw	rw	rw	r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFER	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	

字段	名称	描述
31	BIT1ERRFAST	BRS 置位的 CAN FD 帧 Bit1 错误 — 表示传输位和接收位出现不一致： 1: 至少一个隐性发送位被接收为显性
30	BIT0ERRFAST	BRS 置位的 CAN FD 帧 Bit0 错误 — 表示传输位和接收位出现不一致： 1: 至少一个显性发送位被接收为隐性
29	Reserved	保留，必须保持复位值
28	CRCERRFAST	BRS 置位的 CAN FD 帧 CRC 校验错误 — 表示接收节点检测到 CRC 错误（即计算的 CRC 值不同于接收到的 CRC）：

字段	名称	描述
		1: 自上次读取该寄存器后发生了 CRC 错误
27	FRMERRFAST	BRS 置位的 CAN FD 帧格式错误 — 表示接收节点检测到格式错误（即固定格式的位字段包括至少一个非法位）： 1: 自上次读取该寄存器后发生了格式错误
26	STFERRFAST	BRS 置位的 CAN FD 帧填充错误 — 表示接收节点检测到填充错误： 1: 自上次读取该寄存器后发生了填充错误
25: 22	Reserved	保留，必须保持复位值
21	ERROVR	错误溢出 — 表示在已有错误标志被置位的情况下又发生了错误： 1: 溢出 0: 未溢出 写 1 清零
20	ERRINTFAST	BRS 置位的 CAN FD 报文数据段错误中断： 1: 表示至少有一个错误标志（BIT1ERRFAST、BIT0ERRFAST、CRCERRFAST、FRMERRFAST 或 STFERRFAST）被置位 如果相应的掩码位（CAN_CTRL2.ERRMSKFAST）被置位，则产生中断。 写 1 清零，写 0 无效。
19	BOFFDONEINT	总线关闭完成中断： 1: FlexCAN 已完成总线关闭流程 当 Tx 错误计数器（TXERRCNT）在 CAN 总线上计数到 128 次 11 个连续的隐性位，并准备离开总线关闭状态时，BOFFDONEINT 被置位。如果相应的掩码位（CAN_CTRL2.BOFFDONEMSK）被置位，则 CPU 生成中断。 写 1 清零，写 0 无效。
18	SYNCH	CAN 同步状态： 1: FlexCAN 已同步至 CAN 总线，并能够参与相应的通信流程 0: FlexCAN 未同步至 CAN 总线 该位只读，由 FlexCAN 置位和清零。
17	TWRNINT	Tx 警告中断标志： 1: Tx 错误计数器从小于 96 增至超过或等于 96 如果 CAN_MCR.WRNEN 被置位，当 TXWRN 标志由 0 变 1 时，TWRNINT 被置位，表示 Tx 错误计数器达到 96。如果相应的掩码位（CAN_CTRL1.TWRNMSK）被置位，则产生中断； 当 CAN_MCR.WRNEN 被清零时，TWRNINT 标志被掩码。CPU 必须在 CAN_MCR.WRNEN 被清零前清除 TWRNINT 标志。否则，当 CAN_MCR.WRNEN 再次被置位时，TWRNINT 将被置位。 TWRNINT 不会在总线关闭状态置位，且不会在冻结模式下更新。 写 1 清零，写 0 无效。
16	RWRNINT	Rx 警告中断标志： 1: Rx 错误计数器从小于 96 增至超过或等于 96 如果 CAN_MCR.WRNEN 被置位，当 RXWRN 标志由 0 变 1 时，RWRNINT 被置位，表示 Rx 错误计数器达到 96。如果相应的掩码位（CAN_CTRL1.RWRNMSK）被置位，则产生中断。

字段	名称	描述
		<p>当 CAN_MCR.WRNEN 被清零时，RWRNINT 标志被掩码。CPU 必须在 CAN_MCR.WRNEN 被清零前清除 RWRNINT 标志。否则，当 CAN_MCR.WRNEN 再次被置位时，RWRNINT 将被置位。</p> <p>RWRNINT 不会在冻结模式下更新。</p> <p>写 1 清零，写 0 无效。</p>
15	BIT1ERR	<p>Bit1 错误 — 表示非 FD 报文、CAN FD 报文仲裁或数据段的传输位和接收位出现不一致：</p> <p>1: 至少一个隐性发送位被接收为显性</p> <p>注：在仲裁字段或 ACK SLOT，或节点发送显性的被动错误标志时，BIT1ERR 不会被发送器设置。</p>
14	BIT0ERR	<p>Bit0 错误 — 表示非 FD 报文、CAN FD 报文仲裁或数据段的传输位和接收位出现不一致：</p> <p>1: 至少一个显性发送位被接收为隐性</p>
13	ACKERR	<p>应答错误 — 表示发送节点检测到应答错误（即在 ACK SLOT 未检测到显性位）：</p> <p>1: 自上次读取该寄存器后发生了 ACK 错误</p>
12	CRCERR	<p>CRC 校验错误 — 表示接收节点检测到非 FD 报文、CAN FD 报文仲裁或数据段有 CRC 错误（即计算的 CRC 值不同于接收到的 CRC）：</p> <p>1: 自上次读取该寄存器后发生了 CRC 错误</p>
11	FRMERR	<p>格式错误 — 表示接收节点检测到非 FD 报文、CAN FD 报文仲裁或数据段有格式错误（即固定格式的位字段包括至少一个非法位）：</p> <p>1: 自上次读取该寄存器后发生了格式错误</p>
10	STFERR	<p>填充错误 — 表示接收节点检测到非 FD 报文、CAN FD 报文仲裁或数据段有填充错误：</p> <p>1: 自上次读取该寄存器后发生了填充错误</p>
9	TXWRN	<p>TX 错误警告 — 表示报文传输期间发生重复的错误：</p> <p>1: TXERRCNT 超过或等于 96</p> <p>TXWRN 只受 CAN_ECR.TXERRCNT 的值影响，冻结模式下不会更新。</p>
8	RXWRN	<p>Rx 错误警告 — 表示报文接收期间发生重复的错误：</p> <p>1: RXERRCNT 超过或等于 96</p> <p>RXWRN 只受 CAN_ECR.RXERRCNT 的值影响，冻结模式下不会更新。</p>
7	IDLE	<p>1: CAN 总线为 IDLE 状态</p> <p>参见表格“FLEXCAN 状态相关位编码”。</p>
6	TX	<p>1: FlexCAN 正在传输报文</p> <p>0: FlexCAN 未传输报文</p> <p>参见表格“FLEXCAN 状态相关位编码”。</p>
5: 4	FLTCONF	<p>FlexCAN 的故障界定状态：</p> <p>00: 主动错误</p> <p>01: 被动错误</p> <p>1x: 总线关闭</p> <p>如果 CAN_CTRL1.LOM 被置位，经过延迟后（延迟时间取决于 CAN 位时序），FLTCONF 字段会标志为“被动错误”。同样的延迟会影响 FLTCONF 将更新反映到 CPU 的 CAN_ECR 寄存器，可能需要一个 CAN 位时间来使两者保持一致。</p>

字段	名称	描述
		FLTCONF 受软件复位影响，但如果 CAN_CTRL1.LOM 被置位，则 FLTCONF 的复位值仅持续一个 CAN 位，之后 FLTCONF 指示被动错误。
3	RX	1: FlexCAN 正在接收报文 0: FlexCAN 未接收报文 参见表格“FLEXCAN 状态相关位编码”。
2	BOFFINT	总线关闭中断： 1: FlexCAN 进入总线关闭状态 当 FlexCAN 进入总线关闭状态时，BOFFINT 被置位。如果相应的掩码位（CAN_CTRL1.BOFFMSK）被置位，则产生中断。 写 1 清零，写 0 无效。
1	ERRINT	错误中断： 1: 表明 CAN_ESR1 中至少有一个错误标志（BIT1ERR、BIT0ERR、ACKERR、CRCERR、FRMERR 或 STFERR）被置位 如果相应的掩码位（CAN_CTRL1.ERRMSK）被置位，则产生中断。 写 1 清零，写 0 无效。
0	Reserved	保留，必须保持复位值

在单个 CAN 帧中检测到的错误可通过一个或多个错误标志报告。此外，错误报告可以累计，以防 CPU 读取 CAN_ESR1 之前的后续帧中还有错误事件。

建议 CPU 在处理相应位的中断请求时采用以下流程：

- 读取 CAN_ESR1 以确定所有错误条件和状态位。此操作会清除自上次读取访问以来置位的相应位
- 写 1 以清除触发中断请求的中断位
- 若 ERR_OVR 被置位，则写 1 以清除该位

从清除所有错误标记开始，第一个错误事件置位 ERRINT（如果相应的掩码位被置位）。如果在 CPU 处理中断请求前，后续帧发生其他错误事件，则 ERR_OVR 被置位以指示已累计来自不同帧的错误。

表 30-21 FLEXCAN 状态相关位编码

SYNCH	IDLE	TX	RX	FlexCAN 状态
0	0	0	0	未同步至 CAN 总线
1	1	x	x	空闲
1	0	1	0	传输
1	0	0	1	接收

30.4.10 CAN_IMASK1 中断掩码寄存器 1

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
BUF31TO0M															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUF31TO0M															
rw															

字段	名称	描述
31: 0	BUF31TO0M	<p>MBi 掩码位 — 使能或禁止 MB31 至 MB0 的相应中断:</p> <p>1: 相应的 MB 中断使能</p> <p>0: 相应的 MB 中断禁止</p> <p>注: 如果相应的 CAN_IFLAG1 位被置位, 则置位或清零 CAN_IMASK1 可以产生或屏蔽中断请求。</p>

30.4.11 CAN_IFLAG1 中断标志寄存器 1

偏移地址: 0x30

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BUF31TO8I															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
rw								rw	rw	rw	rw				rw

字段	名称	描述
31: 8	BUF31TO8I	<p>MBi 中断标志位 — 每个位对应 MB31 至 MB8</p> <p>1: 相应的缓冲区已成功传输或接收</p> <p>0: 相应的缓存区未成功传输或接收</p>
7	BUF7I	<p>MB7 中断或传统型 Rx FIFO 溢出:</p> <p>当 CAN_MCR.RFEN 为 0 时,</p> <p>1: MB7 完成传输/接收</p> <p>0: MB7 未完成传输/接收</p> <p>当 CAN_MCR.RFEN 为 1 (传统型 Rx FIFO 使能) 时,</p> <p>1: 传统型 Rx FIFO 溢出, 表示信息丢失, 因为传统型 Rx FIFO 已满</p> <p>0: 传统型 Rx FIFO 未溢出</p> <p>传统型 Rx FIFO 已满且报文被邮箱捕获时, BUF7I 不会置位。</p> <p>注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF7I。</p>
6	BUF6I	<p>MB6 中断或传统型 Rx FIFO 警告:</p> <p>当 CAN_MCR.RFEN 为 0 时,</p> <p>1: MB6 完成传输/接收</p> <p>0: MB6 未完成传输/接收</p> <p>当 CAN_MCR.RFEN 为 1 (传统型 Rx FIFO 使能) 时,</p> <p>1: 传统型 Rx FIFO 几乎为满, 表示未读报文数因接收到新报文而从 4 增加至 5</p>

字段	名称	描述
		<p>0: 传统型 Rx FIFO 未几乎为满 如果未读报文数超过 4 时 BUF6I 被清零, 则 BUF6I 不会再次被置位, 直到传统型 Rx FIFO 中未读报文数降至 4 或更少。 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF6I。</p>
5	BUF5I	<p>MB5 中断或传统型 Rx FIFO 中有可用的帧: 当 CAN_MCR.RFEN 为 0 时, 1: MB5 完成传输/接收 0: MB5 未完成传输/接收 当 CAN_MCR.RFEN 为 1 (传统型 Rx FIFO 使能) 时, 1: 传统型 Rx FIFO 中有可用的帧, 表示传统型 Rx FIFO 中至少有一个帧可读 0: 传统型 Rx FIFO 中无可用的帧 当 CAN_MCR.RFEN 和 CAN_MCR.DMA 被置位时, BUF5I 用于生成 DMA 请求, 且 CPU 不能通过写 1 清除该位。 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF5I。</p>
4: 1	BUF4TO1I	<p>MB4 ~ 1 中断或预留: 当 CAN_MCR.RFEN 为 0 时, 1: 相应的缓存区成功完成传输或接收 0: 相应的缓存区未成功完成传输或接收 当 CAN_MCR.RFEN 为 1 (传统型 Rx FIFO 使能) 时, BUF4TO1I 为预留位 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF4TO1I。</p>
0	BUF0I	<p>MB0 中断或清除传统型 Rx FIFO 当 CAN_MCR.RFEN 为 0 时, 1: MB0 成功完成传输或接收 0: MB0 未成功完成传输或接收 当 CAN_MCR.RFEN 为 1 (传统型 Rx FIFO 使能) 时, BUF0I 触发 FIFO 清除操作, 该操作清空传统型 Rx FIFO 的内容。执行该操作前, CPU 必须处理所有传统型 Rx FIFO 相关的 IFLAG。当 CAN_MCR.DMA 被置位 (DMA 使能) 时, 该操作还会清除 BUF5I 标志并中止 DMA 请求。FIFO 清除操作发生在 CPU 写 1 到 BUF0I 时, 仅在冻结模式下发生, 其他情况下该位被硬件锁定。</p>

每个缓冲区都有一个对应的中断标志位, 每次成功传输或接收, 都会置位相应的 CAN_IFLAG1 位。如果相应的 CAN_IMASK1 位被置位, 则会产生中断。中断标志必须通过写 1 清零, 写 0 无效。

更新 CAN_MCR.MAXMB 字段前, CPU 必须处理 MB 编号大于 CAN_MCR.MAXMB 更新值的 CAN_IFLAG1 位, 否则标志位会保持置位, 且与可用的 MB 数不一致。

传统型 Rx FIFO DMA 使能后的例外情况如下所述:

30.4.11.1 传统型 Rx FIFO

CAN_MCR.RFEN = 1 且 CAN_MCR.DMA = 0 时:

- BUF7I、BUF6I 和 BUF5I 表示传统型 Rx FIFO 的运行条件

- BUF0I 用于清空传统型 Rx FIFO
- BUF4I 至 BUF1I 预留

置位 CAN_MCR.RFEN 前，CPU 必须处理传统型 Rx FIFO 区域被置位的 IFLAG（参见章节“传统型 Rx FIFO 结构”）。否则，这些 IFLAG 将错误地显示相关 MB（现在用于 FIFO）有内容需要处理。

当 CAN_MCR.RFEN 被清零时，传统型 Rx FIFO 标志必须被清零。当选定的 CAN_CTRL2.RFFN 值使传统型 Rx FIFO 过滤器超出 MB7 时，同样需要注意。例如当 RFFN 为 0x3 时，MB0~13 被传统型 Rx FIFO 过滤器占据，相应的 IFLAG 位必须被清零。

注：CAN_MCR.FDEN 置位时必须禁用传统型 Rx FIFO。

30.4.11.2 传统型 Rx FIFO DMA

CAN_MCR.RFEN = 1 且 CAN_MCR.DMA = 1（使能传统型 Rx FIFO DMA 功能）时：

- BUF7I、BUF6I 和 BUF4I ~ BUF1I 未使用
- BUF5I 表示传统型 Rx FIFO 的运行条件
- BUF5I 不生成 CPU 中断，而是生成 DMA 请求
- BUF0I 用于清空传统型 Rx FIFO

当 CAN_MCR.DMA 被置位时，不考虑传统型 Rx FIFO 区域的 IMASK1 位。此外，当 DMA 使能时，CPU 不能清零 BUF5I。置位 CAN_MCR.DMA 前，CPU 必须处理 Rx FIFO 区域被置位的 IFLAG。当 CAN_MCR.DMA 被清零时，传统型 Rx FIFO 必须为空。

30.4.12 CAN_CTRL2 控制寄存器 2

CAN_CTRL2 寄存器为冻结模式下的内存写访问、扩展传统型 Rx FIFO 过滤器数量、调整 FlexCAN 内部流程（如匹配和仲裁）提供控制位。

偏移地址：0x34

复位值：0x00A0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRMS KFAST	BOFFD	Res.		RFFN				TASD					MRP	RRS	EACEN
	ONEMS K			rw				rw					rw	rw	rw
rw	rw			rw				rw					rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREXC EN	Res.	ISOCAN FDEN	EDFLTD IS	Res.										
	rw		rw	rw											

字段	名称	描述
31	ERRMSKFAST	快速错误中断 (CAN_ESR1.ERRINTFAST) 掩码: 1: ERRINTFAST 错误中断使能 0: ERRINTFAST 错误中断禁止
30	BOFFDONEMSK	总线关闭完成中断 (CAN_ESR1.BOFFDONEINT) 掩码: 1: 总线关闭完成中断使能 0: 总线关闭完成中断禁止
29: 28	Reserved	保留, 必须保持复位值
27: 24	RFFN	传统型 Rx FIFO 过滤器数量, 参见表格“传统型 Rx FIFO 过滤器数量” RFFN 的值不能使传统型 Rx FIFO 和传统型 Rx FIFO ID 过滤器占有的 MB 数超过由 CAN_MCR.MAXMB 定义的邮箱数量。 每组八个过滤器占用相当于两个 MB 的内存空间, 因此过滤器越多可用的邮箱越少。考虑到传统型 Rx FIFO 占用了原本为 MB0 ~ 5 预留的内存空间, RFFN 的值应不大于可用内存字的数量: $(\text{SETUP_MB} - 6) \times 4$ 其中: SETUP_MB 为最大邮箱数 32 和 CAN_MCR.MAXMB 中的较小值。 剩余可用邮箱数量: $(\text{SETUP_MB} - 8) - (\text{RFFN} \times 2)$ 如果 RFFN 设定的传统型 Rx FIFO 过滤器数量超过 SETUP_MB, 则超出部分无效。 注: 最后一个剩余可用邮箱的编号为最大邮箱数 32 - 1 和 CAN_MCR.MAXMB 之间的最小值。 如果 Rx 私有掩码寄存器未使能, 则所有传统型 Rx FIFO 过滤器受传统型 Rx FIFO 全局掩码影响。 RFFN 只能在冻结模式下写入, 其他模式下被硬件锁定。
23: 19	TASD	Tx 仲裁启动延迟 — 定义从 CAN 总线的 CRC 字段首位开始, Tx 仲裁过程起始点延迟的 CAN 位数, 参见章节“Tx 仲裁启动延迟”。 TASD 只能在冻结模式下写入, 其他模式下被硬件锁定。
18	MRP	邮箱接收优先级: 1: 匹配从邮箱开始, 若没有发生匹配, 则在传统型 Rx FIFO 中继续匹配 0: 匹配从传统型 Rx FIFO 开始, 在邮箱继续 MRP 只能在冻结模式下写入, 其他模式下被硬件锁定。
17	RRS	远程请求存储: 1: 远程帧被存储 0: 产生远程响应帧 如果 RRS 被置位, 则将远程帧提交至匹配过程, 并以和数据帧相同的方式存储在相应的报文缓冲区中, 不会自动生成远程响应帧。 如果 RRS 被清零, 则将远程帧提交到匹配过程, 且如果一个代码为 0b1010 的报文缓冲区具有相同的 ID, 则自动生成远程响应帧。 RRS 只能在冻结模式下写入, 其他模式下被硬件锁定。
16	EACEN	Rx 邮箱全帧仲裁字段比较: 1: 匹配过程比较 Rx 邮箱过滤器的 IDE、RTR 位和接收帧的相应位, 掩码位适用 0: 匹配过程比较 Rx 邮箱过滤器的 IDE 位, 但不比较 RTR, 与掩码位无关 EACEN 不影响传统型 Rx FIFO 的匹配。 EACEN 只能在冻结模式下写入, 其他模式下被硬件锁定。
15	Reserved	保留, 必须保持复位值

字段	名称	描述
14	PREXCEN	1: 协议例外使能 0: 协议例外禁止 PREXCEN 只能在冻结模式下写入。
13	Reserved	保留, 必须保持复位值
12	ISOCANFDEN	1: ISO CAN FD 使能 (ISO 11898 协议标准) 0: ISO CAN FD 禁止 (Bosch CAN FD 协议标准) ISOCANFDEN 通过使能以下功能以符合 ISO 11898 标准: 1) 填充位的计数从帧起始至数据段的最后一位, 且填充位的模 8 计数和各自的奇偶校验位被组合为 4 位的位填充计数 (Stuff Count) 字段, 插入至 CRC 字段之前 2) CRC 计算扩展至数据段结束, 将 Stuff Count 字段考虑在内 ISOCANFDEN 只能在冻结模式下写入。
11	EDFLTDIS	总线集成状态时: 1: 边沿滤波器禁止 0: 边沿滤波器使能 边沿滤波器使能时, 需要连续两个具有显性总线状态的标称 Tq, 以检测引起同步的边沿。同步发生时, 重新计数 11 个连续的隐性位。边沿滤波器防止小于标称位时间的显性脉冲 (出现在 FD 帧的数据段) 被误认为空闲状态。 EDFLTDIS 只能在冻结模式下写入。
10: 0	Reserved	保留, 必须保持复位值

表 30-22 传统型 Rx FIFO 过滤器数量

RFFN[3: 0]	传统型 Rx FIFO 过滤元素数量	传统型 Rx FIFO 和 ID 过滤表占用的报文缓冲区	剩余可用邮箱	受 Rx 私有掩码影响的传统型 Rx FIFO ID 过滤表元素	受传统型 Rx FIFO 全局掩码影响的 Rx FIFO ID 过滤表元素
0x0	8	MB0 ~ 7	MB8 ~ 31	元素 0 ~ 7	无
0x1	16	MB0 ~ 9	MB10 ~ 31	元素 0 ~ 9	元素 10 ~ 15
0x2	24	MB0 ~ 11	MB12 ~ 31	元素 0 ~ 11	元素 12 ~ 23
0x3	32	MB0 ~ 13	MB14 ~ 31	元素 0 ~ 13	元素 14 ~ 31
0x4	40	MB0 ~ 15	MB16 ~ 31	元素 0 ~ 15	元素 16 ~ 39
0x5	48	MB0 ~ 17	MB18 ~ 31	元素 0 ~ 17	元素 18 ~ 47
0x6	56	MB0 ~ 19	MB20 ~ 31	元素 0 ~ 19	元素 20 ~ 55
0x7	64	MB0 ~ 21	MB22 ~ 31	元素 0 ~ 21	元素 22 ~ 63
0x8	72	MB0 ~ 23	MB24 ~ 31	元素 0 ~ 23	元素 24 ~ 71
0x9	80	MB0 ~ 25	MB26 ~ 31	元素 0 ~ 25	元素 26 ~ 79
0xA	88	MB0 ~ 27	MB28 ~ 31	元素 0 ~ 27	元素 28 ~ 87
0xB	96	MB0 ~ 29	MB30 ~ 31	元素 0 ~ 29	元素 30 ~ 95
0xC	104	MB0 ~ 31	无	元素 0 ~ 31	元素 32 ~ 103

30.4.13 CAN_ESR2 错误和状态 2 寄存器

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.									LPTM						
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VPS	IMB	Res.												
	r	r													

字段	名称	描述
31: 23	Reserved	保留, 必须保持复位值
22: 16	LPTM	最低优先级的 Tx 邮箱 如果 CAN_ESR2.VPS 被置位, 则 LPTM 指示编号最小的非活动邮箱 (见 CAN_ESR2.IMB 位描述)。如果没有非活动邮箱, 则所示邮箱取决于 CAN_CTRL1.LBUF 的值: CAN_CTRL1.LBUF 为 0 时, 指示具有最大仲裁值的邮箱 (参见章节“最高优先级有效优先”); CAN_CTRL1.LBUF 被置位时, 指示编号最大的活动 Tx 邮箱 如果 Tx 邮箱正在被传输, 则 LPTM 不考虑该邮箱。如果 CAN_ESR2.IMB 未被置位, 且帧被成功传输, 则 LPTM 更新为其邮箱号。
15	Reserved	保留, 必须保持复位值
14	VPS	CAN_ESR2.IMB 和 CAN_ESR2.LPTM 的内容当前是否有效: 1: IMB 和 LPTM 的内容有效 0: IMB 和 LPTM 的内容无效 VPS 在每次完成 Tx 仲裁过程时被置位, 除非 CPU 向已被扫描的邮箱写入 C/S (在 Tx 仲裁期间, 位于 Tx 仲裁点后)。如果没有非活动邮箱且只有一个 Tx 邮箱被传输, 则 VPS 不会被置位。 VPS 在每次启动 Tx 仲裁进程或向任何邮箱写入 C/S 时被清零。 注: CPU 向被中止机制锁定的 MB 写入 C/S 不影响 VPS。当 CAN_MCR.AEN 被置位时, 向正在被传输的 MB 的 C/S 写入中止代码 (中止待处理), 或试图写 CAN_IFLAG1 被置位的 Tx MB, 写操作都会被锁定。
13	IMB	非活动邮箱 如果 CAN_ESR2.VPS 被置位, IMB 指示是否有任何非活动邮箱 (代码字段为 0b1000 或 0b0000): 1: 如果 CAN_ESR2.VPS 被置位, 则至少有一个非活动邮箱。CAN_ESR2.LPTM 为第一个邮箱的编号 0: 如果 CAN_ESR2.VPS 被置位, 则 CAN_ESR2.LPTM 不是非活动邮箱 IMB 在下列情况下被置位: 仲裁期间, 如果 CAN_ESR2.LPTM 被找到且非活动 如果 CAN_ESR2.MB 未被置位, 且帧被成功传输

字段	名称	描述
		IMB 在仲裁开始时被清零（参见章节“仲裁过程”）。 注：如果 MB 被成功传输且 CAN_ESR2.IMB=0（无非活动邮箱），则 CAN_ESR2.VPS 和 CAN_ESR2.IMB 被置位，该 MB 的编号被写入 CAN_ESR2.LPTM。
12: 0	Reserved	保留，必须保持复位值

30.4.14 CAN_CRCCR CRC 寄存器

CAN_CRCCR 为发送的非 FD 报文提供 CRC 信息。对于需要 17 或 21 位 CRC 的 CAN FD 帧，必须使用 CAN_FDCRC 寄存器，因为 CAN_CRCCR 只报告 CRC 的低 15 位。Tx 中断标志被置位的同时，CAN_CRCCR 被更新。

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.									MBCRC						
									r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXCRC														
r															

字段	名称	描述
31: 23	Reserved	保留，必须保持复位值
22: 16	MBCRC	CRC 邮箱 — 表示与 CAN_CRCCR.TXCRC 字段对应的邮箱编号
15	Reserved	保留，必须保持复位值
14: 0	TXCRC	传输的 CRC 值 — 表示最后一次传输的非 FD 报文的 CRC 值

30.4.15 CAN_RXFGMASK 传统型 Rx FIFO 全局掩码寄存器

CAN_RXFGMASK 位于 RAM 空间。传统型 Rx FIFO 使能时，用于掩码传统型 Rx FIFO ID 过滤表元素（依照 CAN_CTRL2.RFFN 的设定，没有相应 CAN_RXIMR 的元素）。

偏移地址：0x48

复位值：x

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

字段	名称	描述
31: 0	FGM	传统型 Rx FIFO 全局掩码位 — 掩码 ID 过滤表元素的相应位： 1: 过滤表的相应位被检查 0: 过滤表的相应位“don't care”，细节见下表 只能在冻结模式下写入，其他模式下被硬件锁定。

表 30-23 传统型 Rx FIFO 全局掩码位和 IDF 字段的对应关系

传统型 Rx FIFO ID 过滤表元素格式 (CAN_MCR.IDAM)	ID 接收过滤器字段					
	RTR	IDE	RXIDA	RXIDB ⁽¹⁾	RXIDC ⁽²⁾	预留
A	FGM[31]	FGM[30]	FGM[29: 1]	—	—	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	—	FGM[29: 16] , FGM[13: 0]	—	—
C	—	—	—	—	FGM[31: 24], FGM[23: 16], FGM[15: 8], FGM[7: 0]	—

1) 如果选用格式 B，则接收帧 ID 只有高 14 位与传统型 Rx FIFO 过滤器相比较

2) 如果选用格式 C，则接收帧 ID 只有高 8 位与传统型 Rx FIFO 过滤器相比较

30.4.16 CAN_RXFIR 传统型 Rx FIFO 信息寄存器

偏移地址：0x4C

复位值：x

31	30	29	28	27	26	25	24	23	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.							IDHIT								
r															

字段	名称	描述
31: 9	Reserved	保留，必须保持复位值
8: 0	IDHIT	ID 接收过滤命中指示 — 指示被传统型 Rx FIFO 中接收到的报文命中的 ID 接收过滤器 (IDAF) 如果多个过滤器与接收到的报文 ID 相匹配，则指示匹配过程中第一个匹配的 IDAF (最小编号)。 IDHIT 仅在 CAN_IFLAG1.BUF5I 被置位时有效。

CPU 通过 CAN_RXFIR 访问位于 RAM 的传统型 RXFIR FIFO 的输出。当一个新报文被移入传统型 Rx FIFO 时，FlexCAN 会写该寄存器；当传统型 Rx FIFO 的输出被下一条报文更新时，该寄存器的输出也被更新。CAN_RXFIR 读取说明参见章节“传统型 Rx FIFO”。

30.4.17 CAN_CBT CAN 位时序寄存器

CAN_CBT 寄存器是 CAN_CTRL1 中 CAN 位时序变量的扩展。

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
BTF		EPRES DIV										ERJW			
rw		rw										rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPROPSEG						EPSEG1						EPSEG2			
rw						rw						rw			

字段	名称	描述
31	BTF	位时序格式 — 使用扩展的 CAN 位时序字段 EPRES DIV、EPROPSEG、EPSEG1、EPSEG2 和 ERJW，取代 CAN_CTRL1 定义的 CAN 位时序变量： 1: 使能扩展的位时序定义 0: 禁止扩展的位时序定义 CAN FD 使能 (CAN_MCR.FDEN 置位) 时，需始终置位 BTF。BTF 只能在冻结模式写入，其他模式下被硬件锁定。
30: 21	EPRES DIV	扩展的预分频器分频系数 当 CAN_CBT.BTF 被置位时，EPRES DIV 定义了 PE 时钟频率和串行时钟 (Sclock) 频率之间的比率。EPRES DIV 扩展了 CAN_CTRL1.PRES DIV 的数值范围。 $Sclock \text{ 频率} = PE \text{ 时钟频率} / (EPRES DIV + 1)$ Sclock 周期定义了 CAN 协议的 Tq。对于复位值，Sclock 频率等于 PE 时钟频率 (参见章节“协议时序”)。 当 CAN_CBT.BTF 为 0 时，EPRES DIV 无效。 EPRES DIV 只能在冻结模式下写入，其他模式下被硬件锁定。
20: 16	ERJW	扩展的重同步补偿 当 CAN_CBT.BTF 被置位时，ERJW 定义一次重同步补偿的最大 Tq。ERJW 扩展了 CAN_CTRL1.RJW 的数值范围。 重同步补偿宽度 = ERJW + 1, Tq = Sclock 周期 当 CAN_CBT.BTF 被为 0 时，ERJW 无效。 ERJW 只能在冻结模式下写入，其他模式下被硬件锁定。
15: 10	EPROPSEG	扩展传播段 当 CAN_CBT.BTF 被置位时，EPROPSEG 定义传播段的位时间长度。EPROPSEG 扩展了 CAN_CTRL1.PROPSEG 的数值范围。 传播段时间 = (EPROPSEG + 1) × Tq, Tq = Sclock 周期 当 CAN_CB.BTF 为 0 时，EPROPSEG 无效。 EPROPSEG 只能在冻结模式下写入，其他模式下被硬件锁定。
9: 5	EPSEG1	扩展相位段 1 当 CAN_CBT.BTF 被置位时，EPSEG1 定义位时间中相位段 1 的长度。EPSEG1 扩展了 CAN_CTRL1.EPSEG1 的数值范围。

字段	名称	描述
		相位段 1 = (EPSEG1 + 1) × Tq, Tq = Sclock 周期 当 CAN_CBT.BTF 为 0 时, EPSEG1 无效。 EPSEG1 只能在冻结模式下写入, 其他模式下被硬件锁定。
4: 0	EPSEG2	扩展相位段 2 当 CAN_CBT.BTF 被置位时, EPSEG2 定义位时间中相位段 2 的长度。EPSEG2 扩展了 CAN_CTRL1.PSEG2 的数值范围。 相位段 2 = (EPSEG2 + 1) × Tq, Tq = Sclock 周期 当 CAN_CBT.BTF 为 0 时, EPSEG2 无效。 EPSEG2 只能在冻结模式下写入, 其他模式下被硬件锁定。

注: CAN_CTRL1 和 CAN_CBT 的 CAN 位变量存储在相同的寄存器内。

30.4.18 CAN_RXIMRn Rx 私有掩码寄存器

CAN_RXIMRn 位于 RAM 中, 必须在任何接收之前进行初始化。用于存储 Rx MB 和传统型 Rx FIFO 的 ID 过滤接收掩码。

偏移地址: 0x880 + 4 * n

复位值: x

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

字段	名称	描述
31: 0	MI	私有掩码位: 1: 过滤器的相应位被检查 0: 过滤器的相应位“don't care” 传统型 Rx FIFO 禁止 (CAN_MCR.RFEN = 0) 时, 一一对应地为每个可用的 Rx 邮箱提供私有掩码, 见 RXMGMASK 寄存器描述; 传统型 Rx FIFO 使能 (CAN_MCR.RFEN = 1) 时, 一一对应地为每个传统型 Rx FIFO ID 过滤表元素提供私有掩码, 见 RXFGMASK 寄存器描述。具体取决于 CAN_CTRL2.RFFN 的设置 (参见章节“传统型 Rx FIFO”)。 只能在冻结模式下由 CPU 访问, 不受复位影响。

30.4.19 CAN_FDCTRL CAN FD 控制寄存器

偏移地址: 0xC00

复位值: 0x8000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FDRATE	Res.													MBDSR0		
rw																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TDCEN	TDCFAIL	Res.	TDCOFF					Res.		TDCVAL						
rw		rw		rw							r					

字段	名称	描述
31	FDRATE	<p>比特率切换使能：</p> <p>1：如果 Tx MB 中的 BRS 位为隐性，则发送具有比特率切换的帧</p> <p>0：无视 Tx MB 中的 BRS 位，以标称速率传输帧</p> <p>FDRATE 可随时写入，但只有当 CAN 总线处于等待总线空闲、总线空闲或总线关闭状态，或当前收发的帧到达帧间隔时，该位设定才会生效。FDRATE 为 0 时，CPU 强制所有 CAN FD 报文以标称比特率传输，不受 Tx MB 的 BRS 位影响。</p>
30: 18	Reserved	保留，必须保持复位值
17: 16	MBDSR0	<p>配置数据大小：</p> <p>00：每个报文缓冲区 8 字节</p> <p>01：每个报文缓冲区 16 字节</p> <p>10：每个报文缓冲区 32 字节</p> <p>11：每个报文缓冲区 64 字节</p> <p>MBDSR0 只能在冻结模式下写入。</p>
15	TDCEN	<p>1：收发器延迟补偿（TDC）使能</p> <p>0：收发器延迟补偿（TDC）禁止</p> <p>TDCEN 只能在冻结模式下写入，回环模式必须关闭 TDC。</p>
14	TDCFAIL	<p>收发器延迟补偿失败 — TDC 机制超出范围，无法补偿回路延迟并成功比较延迟的接收位与发送位：</p> <p>1：回路延迟测量值超出范围</p> <p>0：回路延迟测量值在范围内</p> <p>TDCFAIL 在 FlexCAN 第一次检测到超出范围的情况时被设置。要清除 TDCFAIL，请向其写入 1。</p> <p>参见章节“收发器延迟补偿”。</p>
13	Reserved	保留，必须保持复位值
12: 8	TDCOFF	<p>收发器延迟补偿偏移 — 加至收发器环路延迟测量值上的偏移量，以便在比特率切换时定义延迟比较点的位置。</p> <p>以 PE 时钟周期为单位，必须小于数据比特率中 CAN 位持续时间。参见章节“收发器延迟补偿”。</p> <p>TDCOFF 只能在冻结模式下写入，不建议配置为 0</p>
5: 0	TDCVAL	<p>收发器延迟补偿值 — 从发送的 EDL - R0 边沿到接收到相应边沿的收发环路延迟测量值加上 TDCOFF</p> <p>以 PE 时钟周期为单位。参见章节“收发器延迟补偿”。</p>

30.4.20 CAN_FDCBT CAN FD 位时序寄存器

CAN_FDCBT 寄存器存储了 CAN_FDCTRL.FDRATE 置位时，CAN FD 报文数据段使用的 CAN 位时序变量，与 CAN FD 规范兼容。

偏移地址：0xC04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		FPRES DIV										Res.	FRJW		
		rw											rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		FPROPSEG				Res.		FPSEG1			Res.		FPSEG2		
		rw						rw							

字段	名称	描述
31: 30	Reserved	保留，必须保持复位值
29: 20	PPRES DIV	快速预分频器分频系数 FPRES DIV 定义了 BRS 置位的 CAN FD 帧数据比特率部分的 PE 时钟频率和串行时钟（Sclock）频率之间的比率。Sclock 周期定义了 CAN FD 协议数据比特率的 Tq。 Sclock 频率 = PE 时钟频率 / (FPRES DIV + 1) 注：为避免处理 FD 帧时出错，FPRES DIV 和 PRES DIV (CAN_CBT 或 CAN_CTRL1) 请使用相同的值。 FPRES DIV 只能在冻结模式下写入，其他模式下被硬件锁定。
19	Reserved	保留，必须保持复位值
18: 16	FRJW	快速重同步补偿 FRJW 定义了 BRS 置位的 CAN FD 帧数据比特率部分，在一次重同步过程中允许补偿误差的最大 Tq。 重同步补偿宽度 = FRJW + 1, Tq = Sclock 周期
15	Reserved	保留，必须保持复位值
14: 10	FPROPSEG	快速传播段 FPROPSEG 定义了 BRS 置位的 CAN FD 帧数据比特率部分传播段的位时间长度。 传播时间 = FPROPSEG × Tq, Tq = Sclock 周期 FPROPSEG 只能在冻结模式下写入，其他模式下被硬件锁定。
9: 8	Reserved	保留，必须保持复位值
7: 5	FPSEG1	快速相位段 1 FPSEG1 定义了 BRS 置位的 CAN FD 帧数据比特率部分，位时间中相位段 1 的长度。 相位段 1 = (FPSEG1 + 1) × Tq, Tq = Sclock 周期 FPSEG1 只能在冻结模式下写入，其他模式下被硬件锁定。
4: 3	Reserved	保留，必须保持复位值
2: 0	FPSEG2	快速相位段 2

字段	名称	描述
		<p>FPSEG2 定义了 BRS 置位的 CAN FD 帧数据比特率部分，位时间中相位段 2 的长度。</p> <p>相位段 2 = (FPSEG2 + 1) × Tq, Tq = Sclock 周期</p> <p>FPSEG2 只能在冻结模式下写入，其他模式下被硬件锁定。</p>

注：FPROPSEG、FPSEG1 的和必须至少是 2Tq。

30.4.21 CAN_FDCRC CAN FD CRC 寄存器

Tx 中断标志被置位的同时，CAN_FDCRC 被更新。

偏移地址：0xC08

复位值：0x0000 0000

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

字段	名称	描述
31	Reserved	保留，必须保持复位值
31: 24	FDMBCRC	CRC 邮箱 — 表示与 FDTXCRC 字段对应的邮箱编号 该字段适用于 CAN FD 帧和非 FD 帧，与 CAN_CRCCR.MBCRC 相同
23: 21	Reserved	保留，必须保持复位值
20: 0	FDTXCRC	扩展的传输 CRC 值 — 表示最后一次传输的报文的 CRC 值 对于 CRC_15，高 6 位为 0，值与 CAN_CRCCR 相同；对于 CRC_17，高 4 位为零。

30.4.22 CAN_ERFCR 增强型 Rx FIFO 控制寄存器

偏移地址：0xC0C

复位值：0x0000 0000

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

字段	名称	描述
31	ERFEN	1: 增强型 Rx FIFO 使能 0: 增强型 Rx FIFO 禁止 ERFEN 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: CAN_MCR.RFEN 为 1 时, 不能置位 ERFEN。
30: 26	DMALW	DMA 最后一个字 — 定义每个增强型 Rx FIFO 元素最后的 DMA 地址, 参见表格“DMALW 和传输字数”。 DMALW 只能在冻结模式下写入, 其他模式下被硬件锁定。
25: 17	Reserved	保留, 必须保持复位值
16	NEXIF	扩展 ID 过滤元素数 — 定义在增强型 Rx FIFO 匹配过程中使用的扩展 ID 过滤元素个数 参见表格“NEXIF 和扩展/标准 ID 过滤元素数” NEXIF 只能在冻结模式下写入, 其他模式下被硬件锁定。
15: 3	Reserved	保留, 必须保持复位值
2: 0	ERFWM	增强型 Rx FIFO 水线 — 定义存储于增强型 Rx FIFO 中 CAN 报文的最少数量 最少 CAN 报文数 = ERFWM + 1, 达到 ERFWM + 1 时 ERFSR.ERFWM1 置位 ERFWM 只能在冻结模式下写入, 其他模式下被硬件锁定。

表 30-24 DMALW 和传输字数

DMALW	传输的 32 位字数	最后一个 FIFO 的地址
0	1	0x2000
1	2	0x2004
2	3	0x2008
3	4	0x200C
4	5	0x2010
5	6	0x2014
6	7	0x2018
7	8	0x201C
8	9	0x2020
9	10	0x2024
10	11	0x2028
11	12	0x202C
12	13	0x2030
13	14	0x2034
14	15	0x2038
15	16	0x203C
16	17	0x2040
17	18	0x2044
18	19	0x2048
19	20	0x204C

注: 上表中未提及的 DMALW 值为预留值, 不可使用。

表 30-25 NEXIF 和扩展/标准 ID 过滤元素数

NEXIF	扩展 ID 过滤元素数	标准 ID 过滤元素数
0	0	2
1	1	0

30.4.23 CAN_ERFIER 增强型 Rx FIFO 中断使能寄存器

偏移地址: 0xC10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERFUF WIE	ERFOV FIE	ERFWM IIE	ERFDAI E	Res.											
rW	rW	rW	rW												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

字段	名称	描述
31	ERFUFWIE	1: 增强型 Rx FIFO 下溢中断 (ERFSR.ERFUFW) 使能 0: 增强型 Rx FIFO 下溢中断禁止
30	ERFOVFIE	1: 增强型 Rx FIFO 溢出中断 (ERFSR.ERFOVF) 使能 0: 增强型 Rx FIFO 溢出中断禁止
29	ERFWMIE	1: 增强型 Rx FIFO 水线中断 (ERFSR.ERFWM) 使能 0: 增强型 Rx FIFO 水线中断禁止
28	ERFDAIE	1: 增强型 Rx FIFO 数据可用中断 (ERFSR.ERFDA) 使能 0: 增强型 Rx FIFO 数据可用中断禁止
27: 0	Reserved	保留, 必须保持复位值

30.4.24 CAN_ERFSR 增强型 Rx FIFO 状态寄存器

偏移地址: 0xC14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
ERFUF W	ERFOV F	ERFWM I	ERFDA	ERFCL R	Res.										ERFE	ERFF
rW	rW	rW	rW	W											r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.														ERFEL		
r																

字段	名称	描述
31	ERFUFW	1: 增强型 Rx FIFO 下溢。CAN_ERFIER.ERFUFWIE = 1 时产生相应中断 0: 增强型 Rx FIFO 未下溢 写 1 清零。
30	ERFOVF	1: 增强型 Rx FIFO 溢出。CAN_ERFIER.ERFOVFIE = 1 时产生相应中断 0: 增强型 Rx FIFO 未溢出 写 1 清零。
29	ERFWMI	1: 增强型 Rx FIFO 中的报文数大于 CAN_ERFCR.ERFWM 设定值。CAN_ERFIER.ERFWMIIE = 1 时产生相应中断 0: 增强型 Rx FIFO 中的报文数不大于 CAN_ERFCR.ERFWM 设定值 写 1 清零。
28	ERFDA	1: 至少有一条报文存储在增强型 Rx FIFO 中。CAN_ERFIER.ERFDAIE = 1 时产生相应中断 0: 增强型 Rx FIFO 中无报文 写 1 清零。
27	ERFCLR	增强型 Rx FIFO 清除 冻结模式时, 向该位写 1 将清除增强型 Rx FIFO; 非冻结模式时, 写该位无效
26: 18	Reserved	保留, 必须保持复位值
17	ERFE	1: 增强型 Rx FIFO 为空 0: 增强型 Rx FIFO 非空
16	ERFF	1: 增强型 Rx FIFO 满 0: 增强型 Rx FIFO 未满
15: 3	Reserved	保留, 必须保持复位值
2: 0	ERFEL	表示存储在增强型 Rx FIFO 中的 CAN 报文数量

30.4.25 CAN_ERFFELn 增强型 Rx FIFO 过滤元素寄存器

CAN_ERFFELn 寄存器位于 RAM 中, 接收前必须初始化。每个 CAN_ERFFELn 存储一个标准 ID 过滤元素; 每对 CAN_ERFFELn 存储一个扩展 ID 过滤元素。

偏移地址: $0x3000 + 4 * n$

复位值: x

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

字段	名称	描述
31: 0	FEL	用于匹配过程的过滤元素。如果满足匹配条件, 则将报文存储在增强型 Rx FIFO 中。该字段只能在冻结模式下写入, 其他模式下被硬件锁定。

30.4.26 报文缓冲区结构

下图为 FlexCAN 所使用的报文缓冲区结构，包括 CAN 2.0B 的两种帧格式：扩展帧（29 位 ID）和标准帧（11 位 ID）。每个报文缓冲区（MB）由 16、24、40 或 72 字节组成，其中包括 8、16、32 或 64 字节的数据。邮箱使用 0x80 ~ 0x27F 的内存区域。

	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
0x0	EDL	BRS	ESI	预留	CODE				预留	SR	ID	RT	DLC				时间戳															
0x4	PRIO			ID (标准/扩展)								ID (扩展)																				
	预留																															
0x8	数据字节0				数据字节1				数据字节2				数据字节3																			
0xC	数据字节4				数据字节5				数据字节6				数据字节7																			
0x10	数据字节8				数据字节9				数据字节10				数据字节11																			
0x14	数据字节12				数据字节13				数据字节14				数据字节15																			
0x18	数据字节16				数据字节17				数据字节18				数据字节19																			
0x1C	数据字节20				数据字节21				数据字节22				数据字节23																			
0x20	数据字节24				数据字节25				数据字节26				数据字节27																			
0x24	数据字节28				数据字节29				数据字节30				数据字节31																			
0x28	数据字节32				数据字节33				数据字节34				数据字节35																			
0x2C	数据字节36				数据字节37				数据字节38				数据字节39																			
0x30	数据字节40				数据字节41				数据字节42				数据字节43																			
0x34	数据字节44				数据字节45				数据字节46				数据字节47																			
0x38	数据字节48				数据字节49				数据字节50				数据字节51																			
0x3C	数据字节52				数据字节53				数据字节54				数据字节55																			
0x40	数据字节56				数据字节57				数据字节58				数据字节59																			
0x44	数据字节60				数据字节61				数据字节62				数据字节63																			

图 30-13 信息寄存器结构示例

- EDL — 扩展数据长度

EDL 位区分 CAN 帧和 CAN FD 帧。配置为 RANSWER（代码字段为 0b1010）的报文缓冲区不能置位 EDL，细节见下表。

- BRS — 比特率切换，定义是否在 CAN FD 帧内切换比特率
- ESI — 错误状态指示，表示发送节点是错误主动还是错误被动
- CODE — 报文缓冲区代码

CODE 字段可以被 CPU 和 FlexCAN 读写，用作报文缓冲区匹配和仲裁过程的一部分。编码如下表所示。

表 30-26 接收缓冲区代码

CODE	接收到新帧前的 Rx code	SRV ⁽¹⁾	成功接收后的 Rx code ⁽²⁾	RRS ⁽³⁾	注解
0b0000: INACTIVE MB 未激活	INACTIVE	—	—	—	MB 不参与匹配过程
0b0100: EMPTY MB 激活且为空	EMPTY	—	FULL	—	成功接收到帧时（在移入过程之后），CODE 自动更新为 FULL
0b0010: FULL MB 满	FULL	Yes	FULL	—	读取 C/S 并解锁 MB（SRV）不会使代码返回 EMPTY。如果在 MB 被处理后，有新帧被移入 MB，则代码仍保持 FULL。细节参见章节“匹配过程”。
		No	OVERRUN	—	如果 MB 为 FULL，新帧在 CPU 使用该 MB 前被移入，则 CODE 会自动更新为 OVERRUN。细节参见章节“匹配过程”。
0b0110: OVERRUN MB 被覆盖到满的缓冲区中	OVERRUN	Yes	FULL	—	如果 CODE 为 OVERRUN 且 CPU 已处理了 MB，则当一个新帧移动到 MB 时，代码返回 FULL。
		No	OVERRUN	—	如果 CODE 为 OVERRUN，且必须移入另一个新帧，则 MB 将再次被覆盖，代码将保持 OVERRUN。细节参见章节“匹配过程”。
0b1010: RANSWER ⁽⁴⁾ 帧被配置为识别远程帧并回复响应帧 ⁽⁵⁾	RANSWER	—	TANSWER (1110)	0	远程应答被配置用于识别接收到的远程帧。之后，MB 被配置为发送响应帧，代码自动变更为 TANSWER（0b1110）。细节参见章节“匹配过程”。如果 CAN_CTRL2.RRS 被清零，则在收到相同 ID 的远程帧时发送响应帧。
		—	—	1	匹配与仲裁阶段，忽略此代码。细节参见章节“匹配过程”。
CODE[0] = 1: BUSY FLexCAN 正在更新 MB，CPU 禁止访问	BUSY ⁽⁶⁾	—	FULL	—	表示正在更新 MB。自动清零且不影响下一个 CODE。
		—	OVERRUN	—	

1) SRV: 已处理的 MB。MB 被读取，并通过读 CAN_TIMER 或其他 MB 被解锁

2) 帧被移入 MB 则认为该帧被成功接收。参见章节“移入”

3) 远程请求存储位，细节参见寄存器 CAN_CTRL2

4) CODE 0b1010 不被认作 Tx，具有该 CODE 的 MB 不应被中止

5) CODE 0b1010 必须在配置为 CAN FD 格式的报文缓冲区中使用，同时设置 EDL 位

6) 对于 Tx MB，读取时应忽略 BUSY 位，除非 CAN_MCR.AEN 被置位。如果该位被置位，则相应的 MB 不参与匹配过程

表 30-27 发送缓冲区代码

CODE	发送帧前的 Tx code	MB RTR	成功传输后的 Tx code	注解
0b1000: INACTIVE MB 未激活	INACTIVE	—	—	MB 不参与匹配过程
0b1001: ABORT MB 被丢弃	ABORT	—	—	MB 不参与匹配过程
0b1100: DATA MB 为 Tx 数据帧 (MB RTR 必须为 0)	DATA	0	INACTIVE	无条件地发送数据帧一次。传输后, MB 自动返回 INACTIVE 状态
0b1100: REMOTE MB 为远程帧 (MB RTR 必须为 1)	REMOTE	1	EMPTY	无条件地发送远程帧一次。传输后, MB 自动转换为具有相同 ID 的接收空 MB
0b1110: TANSWER MB 为远程帧的响应帧	TANSWER	—	RANSWER	中间代码, 匹配远程帧后由 CHI 自动写 入 MB。远程响应帧将无条件地传输一 次, 之后代码将自动返回到 RANSWER (0b1010)。 CPU 写该 Code 可以产生相同的效果。 远程响应帧可以是数据帧, 也可以是另一 个远程帧, 取决于 RTR 位的值。参见章 节“匹配过程”和“仲裁过程”。

- SRR — 替代远程请求

1: 扩展帧格式传输时, 必须使用隐性位

0: 扩展帧格式传输时, 显性位无效

固定的隐性位, 只用于扩展帧格式。传输时 (发送缓冲区) 该位必须设置为 1, 且将会和从 CAN 总线上接收到的值一起存储于接收缓冲区。该位可以被接收为隐性或显性, 如果 FlexCAN 以显性位接收, 则认为仲裁丢失。

- IDE — ID 扩展位

1: 扩展帧

0: 标准帧

- RTR — 远程传输请求

1: 如果是发送 MB, 则表示当前 MB 可能有一个远程帧待发送; 如果是接收 MB, 则接收到的远程帧将会被存储;

0: 表示当前的 MB 中有一个数据帧待传输。在接收 MB 中, 可能会被用于匹配过程。

如果 FlexCAN 传输 1 (隐性), 接收到 0 (显性), 则认为仲裁丢失。如果 RTR 传输 0 (显性), 接收到 1 (隐性), 则认为是位错误。如果接收到的值与发送值相同, 则被认为是一次成功的位传输。

注: 配置 CAN FD 帧时 RTR 位必须为 0。

- DLC — 数据字节长度

该 4 位字段为发送/接收数据的长度 (以字节为单位), 位于偏移地址为 0x8 到 0xF 的 MB 空间。

接收阶段，该字段由 FlexCAN 写入，从接收帧的 DLC（Data Length Code）字段复制而得；传输阶段，该字段由 CPU 写入，且与要传输的帧的 DLC 字段相对应。当 RTR = 1 时，被传输的帧为远程帧，不包含数据字段（DLC 字段的设置无效，参见表格“有效数据字节”）。

- TIME STAMP — 自由运行计时器时间戳

该 16 位字段为自由运行计时器的复制，当标识符字段开头出现在 CAN 总线上时进行捕获。

- PRIO — 本地优先级

该 3 位字段只有当 MCR.LPRIO_EN 被置位时才有效，且只针对传输邮箱。用于附加到 ID 来定义传输优先级，不会被传输。参见章节“仲裁过程”。

- ID — 帧标识符

标准帧格式，只有高 11 位（28~18）用于识别接收或发送帧，忽略低 18 位。扩展帧格式，所有位都用于识别传输或接收帧。

- DATA BYTE0 ~ 63 — 数据字段

数据帧最多可以使用 64 个字节，取决于为 MB 选择的有效负载大小。从总线上接收到的帧以该帧被接收时的格式进行存放。只有 n 小于 DLC 时，DATA BYTE (n) 才有效。

表 30-28 有效数据字节

DLC	有效数据字节
0	无
1	数据字节 0
2	数据字节 0 ~ 1
3	数据字节 0 ~ 2
4	数据字节 0 ~ 3
5	数据字节 0 ~ 4
6	数据字节 0 ~ 5
7	数据字节 0 ~ 6
8	数据字节 0 ~ 7
9	数据字节 0 ~ 11
10	数据字节 0 ~ 15
11	数据字节 0 ~ 19
12	数据字节 0 ~ 23
13	数据字节 0 ~ 31
14	数据字节 0 ~ 47
15	数据字节 0 ~ 63

30.4.27 MB 内存映射

FlexCAN 内存缓冲区的内存映射如下表所示。

表 30-29 MB 内存映射

MB	偏移地址 (8 字节负载)	偏移地址 (16 字节负载)	偏移地址 (32 字节负载)	偏移地址 (64 字节负载)
MB0	0x0080	0x0080	0x0080	0x0080
MB1	0x0090	0x0098	0x00A8	0x00C8
MB2	0x00A0	0x00B0	0x00D0	0x0110
MB3	0x00B0	0x00C8	0x00F8	0x0158
MB4	0x00C0	0x00E0	0x0120	0x01A0
MB5	0x00D0	0x00F8	0x0148	0x01E8
MB6	0x00E0	0x0110	0x0170	0x0230
MB7	0x00F0	0x0128	0x0198	-
MB8	0x0100	0x0140	0x01C0	-
MB9	0x0110	0x0158	0x01E8	-
MB10	0x0120	0x0170	0x0210	-
MB11	0x0130	0x0188	0x0238	-
MB12	0x0140	0x01A0	-	-
MB13	0x0150	0x01B8	-	-
MB14	0x0160	0x01D0	-	-
MB15	0x0170	0x01E8	-	-
MB16	0x0180	0x200	-	-
MB17	0x0190	0x218	-	-
MB18	0x01A0	0x230	-	-
MB19	0x01B0	0x248	-	-
MB20	0x01C0	0x260	-	-
MB21	0x01D0	-	-	-
MB22	0x01E0	-	-	-
MB23	0x01F0	-	-	-
MB24	0x0200	-	-	-
MB25	0x0210	-	-	-
MB26	0x0220	-	-	-
MB27	0x0230	-	-	-
MB28	0x0240	-	-	-
MB29	0x0250	-	-	-
MB30	0x0260	-	-	-
MB31	0x0270	-	-	-

30.4.28 传统型 Rx FIFO 结构

当 CAN_MCR.RFEN 被置位时, 传统型 Rx FIFO 引擎使用从 0x80 至 0xDC 的内存空间 (通常被 MB 0 ~ 5 占用)。

0x80 ~ 0x8C 包含传统型 Rx FIFO 的输出, 必须作为报文消息缓冲区由 CPU 读取。传统型 Rx FIFO 输出包含已被接收但未被读取的最旧的报文; 0x90 ~ 0xDC 为传统型 Rx FIFO 引擎内部使用预留。

一个额外的内存空间，从 0xE0 开始，可扩展至 0x27C（通常被 MB6 ~ 31 占用），取决于 CAN_CTRL2.RFFN 的设置，包括 ID 过滤表（可配置 8~104 个表元素），该表指定了传统型 Rx FIFO 接收帧的过滤条件。

复位时，可变的 ID 过滤表内存空间默认为 0xE0，仅可扩展至 0xFC，对应于 RFFN = 0 时的 MB 6~7。

	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
0x80	IDHIT										S R R	I D E	R T R	DLC				时间戳														
0x84	预留		ID (标准)										ID (扩展)																			
0x88	数据字节0					数据字节1					数据字节2					数据字节3																
0x8C	数据字节4					数据字节5					数据字节6					数据字节7																
0x90	预留																															
~																																
0xDC																																
0xE0	ID过滤表元素0																															
0xE4	ID过滤表元素1																															
0xE8	ID过滤表元素2																															
~																																
0x274	ID过滤表元素101																															
0x278	ID过滤表元素102																															
0x27C	ID过滤表元素103																															

图 30-14 传统型 Rx FIFO 结构

各 ID 过滤表元素占有整个 32 位字，且可由 1、2 或 4 个标识符接收过滤器（IDAF）组成，取决于 CAN_MCR.IDAM 的设置。下图显示了三个不同格式的 ID 表元素，所有表元素必须具有相同的格式。参见章节“传统型 Rx FIFO”。

	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
格式A	R T R	I D E	RXIDA (标准: 29 ~ 19, 扩展: 29 ~ 1)																								预留					
格式B	R T R	I D E	RXIDB_0 (标准: 29 ~ 19, 扩展: 29 ~ 16)										R T R	I D E	RXIDB_1 (标准: 13 ~ 3, 扩展: 13 ~ 0)																	
格式C	RXIDC_0 (标准/扩展: 31 ~ 24)								RXIDC_1 (标准/扩展: 23 ~ 16)								RXIDC_2 (标准/扩展: 15 ~ 8)								RXIDC_3 (标准/扩展: 7 ~ 0)							

图 30-15 ID 过滤表结构

- RTR — 远程帧，指定如果远程帧匹配目标 ID，是否被接收到传统型 Rx FIFO 中：
 - 1: 接收远程帧，拒绝数据帧
 - 0: 拒绝远程帧，接收数据帧
- IDE — 扩展帧，指定传统型 Rx FIFO 是否接收匹配目标 ID 的扩展帧或标准帧：
 - 1: 接收扩展帧，拒绝标准帧
 - 0: 拒绝扩展帧，接收标准帧

- **RXIDA** — Rx 帧标识符（格式 A）

指定作为传统型 Rx FIFO 接收条件的 ID。标准帧格式，只有高 11 位（29 ~ 19）用作帧标识符；扩展帧格式，所有位都被使用。

- **RXIDB_0、RXIDB_1** — Rx 帧标识符（格式 B）

指定作为传统型 Rx FIFO 接收条件的 ID。标准帧格式，只有高 11 位（29 ~ 19 和 13 ~ 3）用作帧标识符；扩展帧格式，所有 14 位都与接收到的 ID 的高 14 位比较。

- **RXIDC_0、RXIDC_1、RXIDC_2、RXIDC_3** — Rx 帧标识符（格式 C）

指定作为传统型 Rx FIFO 接收条件的 ID。标准帧和扩展帧格式下，所有的 8 位都与接收到的 ID 的高 8 位相比较。

- **IDHIT** — 标识符接收过滤器命中指示

9 位的 IDHIT 字段表示传统型 Rx FIFO 输出的接收报文命中的标识符接收过滤器。参见章节“传统型 Rx FIFO”。

30.4.29 增强型 Rx FIFO 结构

增强型 Rx FIFO 通过置位 CAN_ERFCR.ERFEN 使能。0x2000 ~ 0x204C 包含了增强型 Rx FIFO 的输出，作为报文缓冲区被 CPU 读取。此输出包含已接收但尚未读取的最旧的报文。下图显示了增强型 Rx FIFO 的数据结构。

Preliminary

	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
0x2000	EDL	BR	ESI	预留								SRR	ID	RT	DLC				时间戳													
0x2004	预留			ID (标准)									ID (扩展)																			
0x2008	数据字节0				数据字节1				数据字节2				数据字节3																			
0x200C	数据字节4				数据字节5				数据字节6				数据字节7																			
0x2010	数据字节8				数据字节9				数据字节10				数据字节11																			
0x2014	数据字节12				数据字节13				数据字节14				数据字节15																			
0x2018	数据字节16				数据字节17				数据字节18				数据字节19																			
0x201C	数据字节20				数据字节21				数据字节22				数据字节23																			
0x2020	数据字节24				数据字节25				数据字节26				数据字节27																			
0x2024	数据字节28				数据字节29				数据字节30				数据字节31																			
0x2028	数据字节32				数据字节33				数据字节34				数据字节35																			
0x202C	数据字节36				数据字节37				数据字节38				数据字节39																			
0x2030	数据字节40				数据字节41				数据字节42				数据字节43																			
0x2034	数据字节44				数据字节45				数据字节46				数据字节47																			
0x2038	数据字节48				数据字节49				数据字节50				数据字节51																			
0x203C	数据字节52				数据字节53				数据字节54				数据字节55																			
0x2040	数据字节56				数据字节57				数据字节58				数据字节59																			
0x2044	数据字节60				数据字节61				数据字节62				数据字节63																			
IH_OFF	预留																							ID HIT								
IH_OFF + 4	预留																															
0x2050	5 个增强型 FIFO 元素																															
~																																
0x21DC																																

图 30-16 增强型 Rx FIFO 结构

注：ID HIT 偏移地址随 DLC 动态改变，如下表所示。

表 30-30 ID HIT 偏移地址

DLC	ID HIT offset (IH_OFF)
0	0x2008
1 ~ 4	0x200C
5 ~ 8	0x2010
9	0x2014
10	0x2018
11	0x201C
12	0x2020
13	0x2028
14	0x2038
15	0x2048

- EDL — 扩展数据长度，用于区分 CAN 帧和 CAN FD 帧：

1: CAN FD 帧格式

0: 经典 CAN 帧格式

- BRS — 比特率切换，指示 CAN FD 帧内是否切换比特率：

1: CAN FD 帧内比特率切换

0: CAN FD 帧内比特率不切换

- ESI — 错误状态指示（EDL 为 1 时该位才有意义），指示发送节点是错误主动还是错误被动：

1: 发送节点为错误被动

0: 发送节点为错误主动

- SRR — 替代远程请求

只在扩展格式中使用的固定隐性位。发送节点总是以隐性的方式发送，接收节点可以以隐性或显性的方式接收。如果 FlexCAN 接收显性的 SRR 位，则认为仲裁丢失。

- IDE — 扩展帧，指定帧格式为扩展帧或标准帧：

1: 扩展帧

0: 标准帧

- RTR — 远程帧，指定帧格式为数据帧或远程帧：

1: 远程帧

0: 数据帧

- DLC — 数据长度代码

4 位的 DLC 字段定义了 CAN 帧数据字段的字节数。当 RTR = 1 时，该帧为不包含数据字段的远程帧，忽略 DLC 字段。

- 时间戳 — 16 位时间戳

16 位时间戳字段是在 CAN 帧期间捕获的自由运行计时器的复制。

- ID — 帧标识符

标准帧格式中，高 11 位用于帧识别，低 18 位被忽略。

扩展帧格式中，所有 29 位都用于帧识别。

- 数据字节 0 ~ 63 — 数据字段

数据字段最多可存储 64 个字节。

- ID HIT — 标识符接收过滤器命中指示

ID HIT 字段表示增强型 Rx FIFO 输出的接收报文命中的过滤元素（CAN_ERFFELn）。

31 BKP 备份寄存器

31.1 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。当发生系统复位或系统待机模式下唤醒时，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。

31.2 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

31.3 功能描述

31.3.1 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： $\text{ppm 误差} = \text{偏差} / \text{基准值} * 10^6$ 。

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 0.954ppm ($1/2^{20} * 10^6 = 0.954$)。而 BKP_RTCCR 寄存器校准值 CAL 最大为 127，所以最大可以减慢 121ppm ($0.954\text{ppm} * 127 = 121$)。

31.3.2 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。

注：当 VDD 掉电时，侵入检测功能仍然开启。为了避免误操作导致备份寄存器中的数据被复位，需保证 TAMP 引脚连接到正确的电平。

31.4 寄存器

31.4.1 寄存器总览

表 31-1BKP 寄存器概览

Offset	Acronym	Register Name	Reset
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
0x50+4* (n-1)	BKP_DRn	备份数据寄存器 n	0x00000000

31.4.2 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
Res.						ASOS	ASOE	CCO	CAL[6: 0]						
						rw	rw	rw	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

Bit	Field	Description
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。

31.4.3 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
Res.														TPAL	TPE
														rw	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

31.4.4 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	
Res.						TIF	TEF	Res.						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	侵入中断标志（TAMP Interrupt Flag） 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）。 0: 无侵入中断 1: 产生侵入中断
8	TEF	侵入事件标志（TAMP Event Flag） 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。 0: 无侵入事件 1: 产生侵入事件
7: 3	Reserved	保留，必须保持复位值
2	TPIE	使能侵入引脚中断（TAMP Pin Interrupt Enable） 0: 禁止侵入检测中断 1: 允许侵入检测中断（必须配置 BKP_CR 寄存器的 TPE 位为'1'） 注 1: 侵入中断无法唤醒处于低功耗模式的内核。 注 2: 只有系统复位或由待机模式唤醒后才复位该位
1	CTI	清除侵入检测中断（Clear TAMP Interrupt） 该位只能写入，读出值为 0。 0: 无效 1: 清除侵入检测中断和侵入检测中断标志（TIF）
0	CTE	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。 0: 无效 1: 清除侵入检测事件标志（TEF），同时复位侵入检测器

31.4.5 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：当发生系统复位或系统待机模式下唤醒时，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）。

32 DBG 调试支持

32.1 简介

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。

32.2 功能描述

32.2.1 功能框图

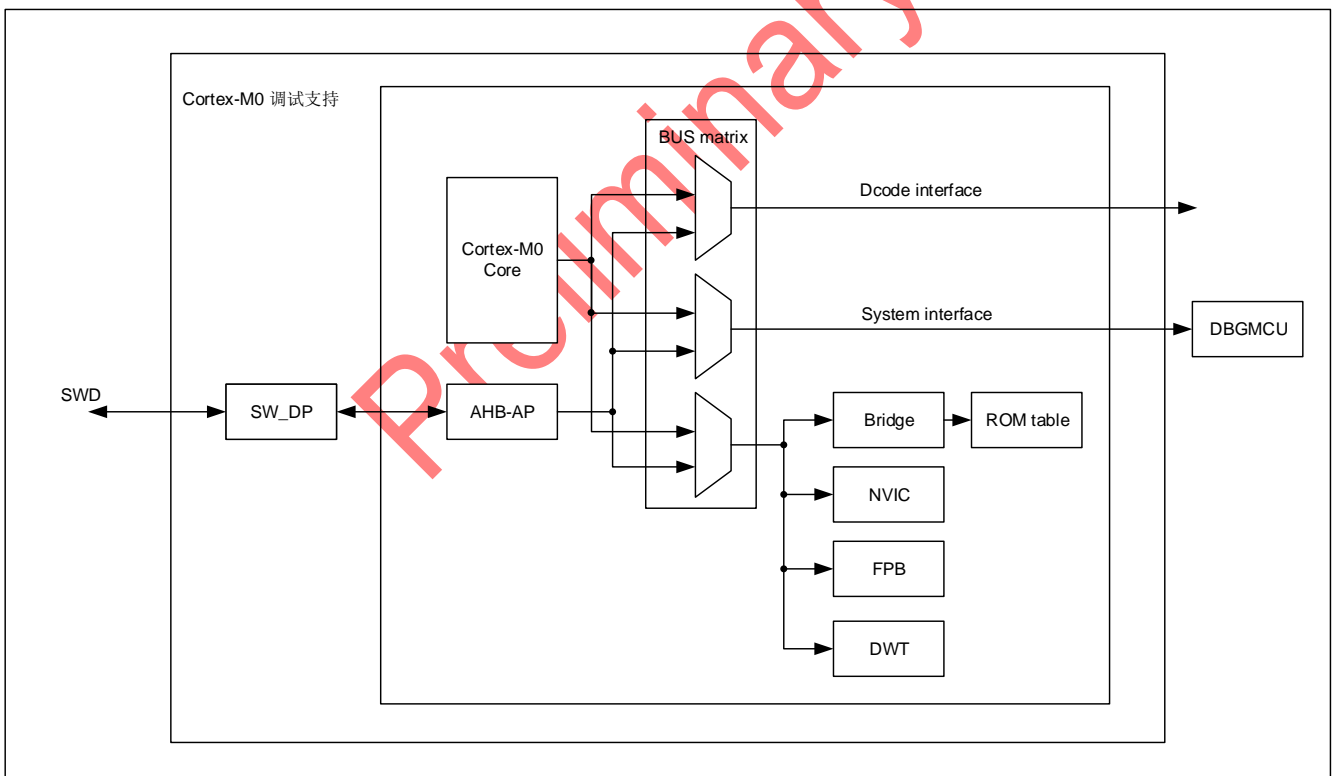


图 32-1 调试功能框图

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

32.2.2 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。

32.2.3 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 32-1 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

32.3 ID 代码和锁定机制

在芯片内部有多个 ID 编码，如下表：

表 32-2 ID 编码

ID 名	芯片
DEV_ID	0X4C50E900
CPU TAP SW ID	0x0BB11477

32.3.1 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。

32.3.2 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 0xE00FF000_0xE00FFFFF 的 4KB ROM 表中。

32.4 SW 调试端口

32.4.1 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。

32.4.2 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 32-3 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 32-4 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 32-5 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。

32.4.3 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;

32.4.4 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;

32.4.5 SW-DP 寄存器

当 APnDP=0 时, 可以访问以下这些寄存器。

表 32-6 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。

32.4.6 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

32.5 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

32.5.1 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。
- 当进入待机模式时，须配置 DBG_STANDBY 位，该配置下数字电路将不会断电，并且将会激活

HSI 为 FCLK 和 HCLK 提供时钟。

32.5.2 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

32.6 寄存器

32.6.1 寄存器总览

表 32-7 DBG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID 编码寄存器	0X4C50E900
0x04	DBG_CR	DBG 控制寄存器	0x00200000

32.6.2 DBG_IDCODE ID 编码寄存器

偏移地址：0x40013400（只支持 32 位访问，只读）

复位值：0X4C50E900

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

32.6.3 DBG_CR 控制寄存器

偏移地址：0x40013404（只支持 32 位访问，只读）

复位值：0x0020 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.							IPG_DE BUG	Res.	IPS_TE ST_AC CESS	IPS_SU PERVIS OR_AC CESS	Res.		DBG_TI M14_ST OP	DBG_TI M17_ST OP	DBG_TI M16_ST OP
							rw		rw	rw			rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBG_TI M3_ST OP	DBG_TI M2_ST OP	DBG_TI M1_ST OP	DBG_W WDG_S TOP	DBG_I WDG_S TOP	Res.				DBG_S TOP_F OR_LD O	DBG_S TANDB Y	DBG_S TOP	DBG_S LEEP
			rw	rw	rw	rw	rw					rw	rw	rw	rw

Bit	Field	Description
31: 25	Reserved	保留，必须保持复位值
24	IPG_DEBUG	FCAN Debug Mode 请求 0: 不请求 FCAN debug 模式 1: 请求 FCAN debug 模式
23	Reserved	保留，必须保持复位值
22	IPS_TEST_ACCESS	IPS TEST ACCESS 0: 指示当前访问不具有测试模式特权 1: 指示当前访问具有测试模式特权
21	IPS_SUPERVISOR_ACCESS	IPS_SUPERVISOR_ACCESS 0: 指示当前访问具有 supervisor mode 特权 1: 指示当前访问不具有 supervisor mode 特权
20: 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 16 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: 13	Reserved	保留，必须保持复位值

Bit	Field	Description
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	DBG_TIM2_STOP	当内核进入调试状态时计数器 2 停止工作 (TIM2 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
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	DBG_WWDG_STOP	当内核进入调试状态时调试窗口看门狗停止工作 (Debug Window Watchdog Stopped When Core is Halted) 0: 窗口看门狗计数器仍然正常工作 1: 窗口看门狗计数器停止工作
8	DBG_IWDG_STOP	独立看门狗停止工作 (Debug Independent Watchdog Stopped) 此位与内核是否进入调试状态无关 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_LDO	调试停机模式 LDO 状态 (Debug Stop Mode For LDO) 0: LDO 进入低功耗状态, 正常进入 STOP 模式 1: LDO 不进入低功耗状态, PLL 保持供电, 不能进入真正 STOP 模式。CPU 进入 DEEPSLEEP, HCLK 关闭
2	DBG_STANDBY	调试待机模式 (Debug Standby Mode) 0: 数字电路部分完全断电, HCLK 与 FCLK 关闭。除指示退出待机模式的标志位, STANDBY 模式的退出与复位行为相同 1: 数字电路不断电, 时钟 FCLK 与 HCLK 开启, 由 HSI 提供时钟。STANDBY 模式的退出与复位行为相同。
1	DBG_STOP	调试停机模式 (Debug Stop Mode) 0: 在停机模式时, 时钟控制器禁止一切时钟 (包括 HCLK 和 FCLK)。当从 STOP 模式退出时, 时钟配置与复位之后的配置一致。如果需要重新使能 PLL, HSE 等时钟, 必须通过软件重新配置 1: 在停机模式时, FCLK 与 HCLK 开启, 由 HSI 提供时钟。当从 STOP 模式退出时, 如果需要重新使能 PLL, HSE 等时钟, 必须通过软件重新配置
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, 都保持原先配置好的时钟。

33 Device Electronic Signature 器件电子签名

33.1 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

33.2 寄存器描述

基地址: 0x1FFF F7E8

表 33-1 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

33.2.1 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															
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)												

33.2.2 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)												

33.2.3 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)												

34 修订记录

表 34-1 修订记录表

修订时间	修订版本	修订内容
2023/02/22	V0.5	初步版本

Preliminary