



用户手册

User Manual

MM32F3270

基于Arm®Cortex®-M3内核的32位微控制器

版本：1.1

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
1.1.1 ICode 总线（IBUS）	29
1.1.2 DCode 总线（DBUS）	29
1.1.3 系统总线（SBUS）	29
1.1.4 DMA 总线	29
1.1.5 总线矩阵（BUS Matrix）	29
1.1.6 AHB 到 APB 桥（AHB-APB APBx）	29
1.2 存储器介绍	31
1.2.1 存储器映像和寄存器编址.....	31
1.2.2 内置的 SRAM	33
1.2.3 闪存存储器概述	33
1.2.4 启动配置	34
1.2.5 引导程序	34
2 Embedded FLASH 嵌入式闪存	35
2.1 简介	35
2.2 闪存构成与说明	35
2.2.1 闪存构成	35
2.2.2 选项字节说明	36
2.2.3 保密空间说明	38
2.3 闪存操作与流程	38
2.3.1 闪存读操作	38
2.3.2 闪存编程方式与操作流程.....	39
2.3.3 对闪存块操作限制的解除与使能.....	41
2.3.4 对选项字节区块操作限制的解除与使能.....	42
2.3.5 主闪存块擦除	44
2.3.6 主闪存块编程	46
2.3.7 选项字节区块擦除	47
2.3.8 选项字节区块编程	48
2.3.9 闪存保护	49
2.4 寄存器.....	51
2.4.1 寄存器总览	51
2.4.2 FLASH_ACR 闪存访问控制寄存器.....	52
2.4.3 FLASH_KEYR FPEC 键寄存器	53
2.4.4 FLASH_OPTKEYR 闪存 OPTKEY 寄存器	53
2.4.5 FLASH_SR 闪存状态寄存器	54
2.4.6 FLASH_CR 闪存控制寄存器	54

2.4.7	FLASH_AR 闪存地址寄存器	55
2.4.8	FLASH_OBR 选项字节寄存器.....	56
2.4.9	FLASH_WRPRx 写保护寄存器	56
3	PWR 电源控制.....	58
3.1	供电系统	58
3.1.1	模拟模块供电.....	59
3.1.2	数字模块供电.....	59
3.1.3	电池备份模块供电.....	59
3.1.4	VDD 域	59
3.1.5	备份域.....	59
3.1.6	1.5V 域.....	59
3.2	电源管理器	60
3.2.1	POR 上电复位和 PDR 掉电复位.....	60
3.2.2	PVD 可编程电压监测器	60
3.3	功耗控制	61
3.3.1	功耗控制概述.....	61
3.3.2	运行模式降低系统时钟	63
3.3.3	外设时钟的控制	63
3.3.4	低功耗运行模式	64
3.3.5	睡眠模式 (Sleep Mode)	64
3.3.6	Low Power Sleep Mode 低功耗睡眠模式.....	65
3.3.7	Stop Mode 停机模式.....	66
3.3.8	DeepStop Mode 深度停机模式.....	67
3.3.9	Standby Mode 待机模式.....	69
3.4	调试模式	70
3.5	电源控制寄存器	70
3.5.1	PWR_CR1 电源控制寄存器 1	71
3.5.2	PWR_CSR 电源控制/状态寄存器.....	72
3.5.3	PWR_CR2 电源控制寄存器 2	72
3.5.4	PWR_CR3 电源控制寄存器 3	73
3.5.5	PWR_CR4 电源控制寄存器 4	73
3.5.6	PWR_CR5 电源控制寄存器 5	74
3.5.7	PWR_CR6 电源控制寄存器 6	75
3.5.8	PWR_SR 电源状态寄存器.....	75
3.5.9	PWR_SCR 电源状态清除寄存器.....	76
3.5.10	PWR_CFGR 电源配置寄存器	76
4	RCC 时钟和复位.....	78
4.1	复位单元	78
4.1.1	简介	78
4.1.2	功能框图	78
4.1.3	主要特征	78
4.1.4	功能描述	79
4.2	时钟单元	80
4.2.1	简介	80

4.2.2	功能框图	81
4.2.3	主要特征	81
4.2.4	功能描述	82
4.3	寄存器.....	90
4.3.1	寄存器总览	90
4.3.2	RCC_CR 时钟控制寄存器	91
4.3.3	RCC_CFGR 时钟配置寄存器	93
4.3.4	RCC_CIR 时钟中断寄存器	95
4.3.5	RCC_AHB3RSTR AHB3 外设复位寄存器.....	97
4.3.6	RCC_AHB2RSTR AHB2 外设复位寄存器.....	97
4.3.7	RCC_AHB1RSTR AHB1 外设复位寄存器.....	98
4.3.8	RCC_APB2RSTR APB2 外设复位寄存器.....	99
4.3.9	RCC_APB1RSTR APB1 外设复位寄存器.....	101
4.3.10	RCC_AHB3ENR AHB3 外设时钟使能寄存器	103
4.3.11	RCC_AHB2ENR AHB2 外设时钟使能寄存器	104
4.3.12	RCC_AHB1ENR AHB1 外设时钟使能寄存器	105
4.3.13	RCC_APB2ENR APB2 外设时钟使能寄存器	106
4.3.14	RCC_APB1ENR APB1 外设时钟使能寄存器	108
4.3.15	RCC_BDCR 备份域控制寄存器	110
4.3.16	RCC_CSR 控制状态寄存器.....	112
4.3.17	RCC_SYSCFG 系统配置寄存器	114
4.3.18	RCC_CFGR2 时钟配置寄存器 2.....	115
4.3.19	RCC_PLLCFGR PLL 配置寄存器	116
5	SYSCFG 系统控制器.....	118
5.1	简介	118
5.2	寄存器.....	118
5.2.1	寄存器总览	118
5.2.2	SYSCFG_CFGR1 配置寄存器 1	118
5.2.3	SYSCFG_EXTICR1 外部中断配置寄存器 1	119
5.2.4	SYSCFG_EXTICR2 外部中断配置寄存器 2	120
5.2.5	SYSCFG_EXTICR3 外部中断配置寄存器 3	120
5.2.6	SYSCFG_EXTICR4 外部中断配置寄存器 4	121
5.2.7	SYSCFG_CFGR2 配置寄存器 2	122
5.2.8	SYSCFG_PDETCSR 电源检测配置状态寄存器	122
5.2.9	SYSCFG_VOSDLY 电压调试器输出延时寄存器.....	123
6	EXTI 中断和事件	125
6.1	简介	125
6.2	主要特征	125
6.3	功能描述	126
6.3.1	功能框图	126
6.3.2	中断和异常向量	126
6.3.3	唤醒事件管理	129
6.3.4	中断功能描述	129
6.3.5	硬件中断输出	129

6.3.6	硬件事件输出	129
6.3.7	软件中断与事件输出	130
6.3.8	外部中断映射	130
6.4	寄存器描述	131
6.4.1	寄存器总览	131
6.4.2	EXTI_IMR 中断屏蔽寄存器	131
6.4.3	EXTI_EMR 事件屏蔽寄存器	131
6.4.4	EXTI_RTSR 上升沿触发选择寄存器	132
6.4.5	EXTI_FTSR 下降沿触发选择寄存器	132
6.4.6	EXTI_SWIER 软件中断事件寄存器	133
6.4.7	EXTI_PR 软件中断事件挂起寄存器	133
7	DMA1/2 直接存储器访问控制器	134
7.1	简介	134
7.2	功能框图	134
7.3	主要特征	134
7.4	中断	135
7.5	DMA	135
7.5.1	DMA 请求映像	135
7.6	功能描述	136
7.6.1	DMA 处理	136
7.6.2	仲裁器	137
7.6.3	DMA 通道	137
7.6.4	可编程的数据传输宽度，对齐方式和数据大小端	138
7.6.5	错误管理	140
7.7	DMA 寄存器描述	140
7.7.1	DMA_ISR DMA 中断状态寄存器	140
7.7.2	DMA_IFCR DMA 中断标志清除寄存器	141
7.7.3	DMA_CCRx DMA 通道 x 配置寄存器 ($x=1\sim7$)	142
7.7.4	DMA_CNDTRx DMA 通道 x 传输数量寄存器 ($x=1\sim7$)	144
7.7.5	DMA_CPARx DMA 通道 x 外设地址寄存器 ($x=1\sim7$)	144
7.7.6	DMA_CMARx DMA 通道 x 存储器地址寄存器 ($x = 1\sim7$)	145
8	BKP 备份寄存器	146
8.1	简介	146
8.2	主要特征	146
8.3	功能描述	146
8.3.1	时钟校准	146
8.3.2	侵入检测	146
8.4	寄存器	147
8.4.1	寄存器总览	147
8.4.2	备份数据寄存器 n(BKP_DRn)(n = 1 .. 20)	147
8.4.3	时钟校准寄存器 (BKP_RTCCR)	147
8.4.4	备份控制寄存器 (BKP_CR)	148
8.4.5	备份控制状态寄存器 (BKP_CSR)	149
9	CRS 时钟回馈系统	150

9.1	简介	150
9.2	主要特征	150
9.3	功能描述	151
9.3.1	功能框图	151
9.3.2	信号描述	151
9.3.3	功能概述	151
9.3.4	同步输入	151
9.3.5	频率误差测量	152
9.3.6	频率误差计算及自动校准	156
9.3.7	CRS 初始化及配置	156
9.4	CRS 低功耗模式	157
9.5	中断	157
9.6	寄存器	158
9.6.1	寄存器总览	158
9.6.2	CRS_CR 控制寄存器	158
9.6.3	CRS_CFGR 配置寄存器	159
9.6.4	CRS_ISR 中断状态寄存器	160
9.6.5	CRS_ICR 中断标志清除寄存器	162
10	GPIO 通用端口	163
10.1	简介	163
10.2	主要特征	163
10.3	功能描述	164
10.3.1	功能框图	164
10.3.2	GPIO 端口配置	164
10.3.3	复用功能	165
10.3.4	GPIO 锁定机制	165
10.3.5	输入配置	166
10.3.6	输出配置	167
10.3.7	复用功能配置	167
10.3.8	模拟输入配置	168
10.3.9	外部时钟复用 GPIO 端口	169
10.3.10	SWD 复用功能重映射	169
10.4	GPIO 寄存器描述	169
10.4.1	寄存器总览	169
10.4.2	GPIOx_CRL 端口配置低寄存器	170
10.4.3	GPIOx_CRH 端口配置高寄存器	170
10.4.4	GPIOx_IDR 端口输入数据寄存器	171
10.4.5	GPIOx_ODR 端口输出数据寄存器	172
10.4.6	GPIOx_BSRR 端口设置/清除寄存器	172
10.4.7	GPIOx_BRR 端口位清除寄存器	173
10.4.8	GPIOx_LCKR 端口配置锁定寄存器	173
10.4.9	GPIOx_DCR 端口输出开漏控制寄存器	174
10.4.10	GPIOx_AFRL 端口复用功能低位寄存器	174
10.4.11	GPIOx_AFRH 端口复用功能高位寄存器	175

11	TIM1/8 高级定时器	176
11.1	简介	176
11.2	功能框图	176
11.3	主要特征	177
11.4	功能描述	177
11.4.1	时钟	177
11.4.2	重复计数器	182
11.4.3	输入捕获	184
11.4.4	比较输出	186
11.4.5	从模式	198
11.4.6	定时器同步	202
11.4.7	定时器异或	205
11.4.8	调试模式	206
11.4.9	中断	207
11.4.10	DMA	207
11.5	寄存器描述	208
11.5.1	TIMx_CR1 控制寄存器 1	208
11.5.2	TIMx_CR2 控制寄存器 2	210
11.5.3	TIMx_SMCR 从模式控制寄存器	211
11.5.4	TIMx_DIER DMA/中断使能寄存器	214
11.5.5	TIMx_SR 状态寄存器	216
11.5.6	TIMx_EGR 事件产生寄存器	218
11.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1	219
11.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2	223
11.5.9	TIMx_CCER 捕获/比较使能寄存器	227
11.5.10	TIMx_CNT 计数器	229
11.5.11	TIMx_PSC 预分频器	230
11.5.12	TIMx_ARR 自动预装载寄存器	230
11.5.13	TIMx_RCR 重复计数寄存器	230
11.5.14	TIMx_CCR1 捕获/比较寄存器 1	231
11.5.15	TIMx_CCR2 捕获/比较寄存器 2	231
11.5.16	TIMx_CCR3 捕获/比较寄存器 3	232
11.5.17	TIMx_CCR4 捕获/比较寄存器 4	232
11.5.18	TIMx_BDTR 刹车和死区寄存器	232
11.5.19	TIMx_DCR DMA 控制寄存器	234
11.5.20	TIMx_DMAR 连续模式 DMA 地址寄存器	235
11.5.21	TIMx_CCMR3 捕获/比较模式寄存器 3	236
11.5.22	TIMx_CCR5 捕获/比较寄存器 5	236
11.5.23	TIMx_PDER PWM 移相/DMA repeat 更新请求使能寄存器	237
11.5.24	TIMx_CCRxFALL PWM 移相递减计数捕获/比较寄存器	238
12	TIM3/4 通用定时器	239
12.1	简介	239
12.2	功能框图	239
12.3	主要特征	239

12.4 功能描述	240
12.4.1 时钟	240
12.4.2 输入捕获	245
12.4.3 比较输出	247
12.4.4 从模式	254
12.4.5 定时器同步	258
12.4.6 定时器异或	258
12.4.7 调试模式	259
12.4.8 中断	259
12.4.9 DMA	259
12.5 寄存器描述	260
12.5.1 TIMx_CR1 控制寄存器 1	261
12.5.2 TIMx_CR2 控制寄存器 2	262
12.5.3 TIMx_SMCR 从模式控制寄存器	263
12.5.4 TIMx_DIER DMA/中断使能寄存器	266
12.5.5 TIMx_SR 状态寄存器	268
12.5.6 TIMx_EGR 事件产生寄存器	269
12.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1	270
12.5.8 TIMx_CCMR2 捕获/比较模式寄存器 2	274
12.5.9 TIMx_CCER 捕获/比较使能寄存器	278
12.5.10 TIMx_CNT 计数器	280
12.5.11 TIMx_PSC 预分频器	280
12.5.12 TIMx_ARR 自动预装载寄存器	281
12.5.13 TIMx_CCR1 捕获/比较寄存器 1	281
12.5.14 TIMx_CCR2 捕获/比较寄存器 2	282
12.5.15 TIMx_CCR3 捕获/比较寄存器 3	282
12.5.16 TIMx_CCR4 捕获/比较寄存器 4	282
12.5.17 TIMx_DCR DMA 控制寄存器	283
12.5.18 TIMx_DMAR 连续模式 DMA 地址寄存器	283
13 TIM2/5 通用定时器	285
13.1 简介	285
13.2 功能框图	285
13.3 主要特征	285
13.4 功能描述	286
13.4.1 时钟	286
13.4.2 输入捕获	291
13.4.3 比较输出	293
13.4.4 从模式	300
13.4.5 定时器同步	304
13.4.6 定时器异或	304
13.4.7 调试模式	305
13.4.8 中断	305
13.4.9 DMA	305
13.5 寄存器描述	306

13.5.1	TIMx_CR1 控制寄存器 1	307
13.5.2	TIMx_CR2 控制寄存器 2	308
13.5.3	TIMx_SMCR 从模式控制寄存器.....	309
13.5.4	TIMx_DIER DMA/中断使能寄存器	312
13.5.5	TIMx_SR 状态寄存器	314
13.5.6	TIMx_EGR 事件产生寄存器	315
13.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1.....	316
13.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2	320
13.5.9	TIMx_CCER 捕获/比较使能寄存器.....	324
13.5.10	TIMx_CNT 计数器	326
13.5.11	TIMx_PSC 预分频器.....	326
13.5.12	TIMx_ARR 自动预装载寄存器.....	327
13.5.13	TIMx_CCR1 捕获/比较寄存器 1	327
13.5.14	TIMx_CCR2 捕获/比较寄存器 2	328
13.5.15	TIMx_CCR3 捕获/比较寄存器 3	328
13.5.16	TIMx_CCR4 捕获/比较寄存器 4	329
13.5.17	TIMx_DCR DMA 控制寄存器	329
13.5.18	TIMx_DMAR 连续模式 DMA 地址寄存器	330
13.5.19	TIMx_OR 输入选项寄存器.....	330
14	TIM6/7 基本定时器	332
14.1	简介	332
14.2	功能框图.....	332
14.3	主要特征	332
14.4	功能描述	333
14.4.1	时钟	333
14.4.2	调试模式	334
14.4.3	中断	334
14.4.4	DMA.....	334
14.5	寄存器描述	334
14.5.1	TIMx_CR1 控制寄存器 1	335
14.5.2	TIMx_DIER DMA/中断使能寄存器	335
14.5.3	TIMx_SR 状态寄存器	336
14.5.4	TIMx_EGR 事件产生寄存器	336
14.5.5	TIMx_CNT 计数器	337
14.5.6	TIMx_PSC 预分频器.....	337
14.5.7	TIMx_ARR 自动预装载寄存器.....	337
15	SDIO 安全数字输入输出接口	339
15.1	简介	339
15.2	主要特征	339
15.3	功能描述	340
15.3.1	功能框图	340
15.3.2	信号描述	340
15.3.3	中断事件	340
15.3.4	DMA.....	341

15.3.5	FIFO	341
15.3.6	传输信号时序.....	342
15.3.7	传输控制命令.....	343
15.4	寄存器.....	345
15.4.1	寄存器总览	345
15.4.2	CTRL 控制寄存器	345
15.4.3	IO IO 信号寄存器	347
15.4.4	BYTECNT 数据传输字节计数寄存器.....	348
15.4.5	TRBLOCKCNT 块计数寄存器	349
15.4.6	CRCCTL CRC 控制寄存器	349
15.4.7	CMDCRC 命令 CRC 寄存器.....	350
15.4.8	DATCRC 数据 CRC 寄存器	351
15.4.9	PORT 端口控制寄存器	351
15.4.10	INTEN 中断使能寄存器	352
15.4.11	INTSTA 中断状态寄存器.....	353
15.4.12	CARDSEL 卡选择寄存器.....	354
15.4.13	SIG 信号状态寄存器	355
15.4.14	IOMBCTL 多块控制寄存器	355
15.4.15	BLOCKCNT 数据块传输计数寄存器.....	356
15.4.16	TIMEOUTCNT 数据传输超时寄存器	357
15.4.17	CMDBUF _x (x=0...15) 命令缓冲寄存器.....	357
15.4.18	BUFCTL 数据缓冲控制寄存器.....	358
15.4.19	DATABUF 数据缓冲寄存器	359
16	IWDG 独立看门狗.....	361
16.1	简介	361
16.2	功能框图	362
16.3	主要特征	362
16.4	功能描述	363
16.4.1	流程框图	364
16.4.2	独立看门狗超时时间	364
16.4.3	中断	365
16.5	寄存器描述	365
16.5.1	寄存器总览	365
16.5.2	IWDG_KR 键寄存器	365
16.5.3	IWDG_PR 预分频寄存器	366
16.5.4	IWDG_RLR 重装载寄存器	367
16.5.5	IWDG_SR 状态寄存器	367
16.5.6	IWDG_CR 控制寄存器	368
16.5.7	IWDG_IGEN 中断生成寄存器	369
16.5.8	IWDG_CNT 计数寄存器	369
16.5.9	IWDG_PS 分频计数寄存器	369
17	WWDG 窗口看门狗	371
17.1	简介	371
17.2	功能框图	371

17.3	主要特征	371
17.4	功能描述	372
17.5	窗口看门狗超时时间	373
17.6	寄存器描述	373
17.6.1	控制寄存器 (WWDG_CR)	374
17.6.2	配置寄存器 (WWDG_CFGR)	374
17.6.3	状态寄存器 (WWDG_SR)	375
18	RTC 实时时钟器	376
18.1	简介	376
18.2	主要特征	376
18.3	功能描述	377
18.3.1	功能框图	377
18.3.2	功能概述	377
18.3.3	模块复位	378
18.3.4	寄存器读取	378
18.3.5	寄存器配置	378
18.3.6	标志位产生	379
18.3.7	RTC 闹钟描述	380
18.3.8	RTC 外部中断事件输出	380
18.4	寄存器描述	380
18.4.1	寄存器总览	380
18.4.2	控制寄存器高位 (RTC_CRH)	381
18.4.3	控制寄存器低位 (RTC_CRL)	381
18.4.4	预分频装载寄存器高位 (RTC_PRLH)	383
18.4.5	预分频装载寄存器低位 (RTC_PRLL)	383
18.4.6	预分频器分频因子寄存器高位 (RTC_DIVH)	383
18.4.7	预分频器分频因子寄存器低位 (RTC_DIVL)	384
18.4.8	计数器寄存器高位 (RTC_CNTH)	384
18.4.9	计数器寄存器低位 (RTC_CNTL)	384
18.4.10	闹钟寄存器高位 (RTC_ALRH)	385
18.4.11	闹钟寄存器低位 (RTC_ALRL)	385
18.4.12	毫秒闹钟寄存器高位 (RTC_MSRH)	385
18.4.13	毫秒闹钟寄存器低位 (RTC_MSRL)	386
18.4.14	RTC LSE 配置寄存器 (RTC_LSE_CFG)	386
19	SPI_I2S 串行外设接口	388
19.1	SPI_I2S 功能框图	388
19.2	SPI_I2S 简述	388
19.3	SPI 功能描述	389
19.3.1	概述	389
19.3.2	SPI 主要特征	391
19.3.3	SPI 从模式	392
19.3.4	SPI 主模式	393
19.3.5	波特率设置	394
19.3.6	中断	394

19.3.7 DMA 传输	395
19.4 I2S 功能描述.....	395
19.4.1 I2S 主要特征.....	395
19.4.2 I2S 总线接口.....	396
19.4.3 数据格式	396
19.4.4 通信标准	396
19.4.5 从模式.....	399
19.4.6 主模式.....	399
19.4.7 时钟预分频器.....	400
19.4.8 中断	402
19.4.9 DMA 传输	402
19.5 寄存器.....	403
19.5.1 寄存器总览	403
19.5.2 SPI_I2S_TXREG 发送数据寄存器	403
19.5.3 SPI_I2S_RXREG 接收数据寄存器	403
19.5.4 SPI_I2S_CSTAT 当前状态寄存器.....	404
19.5.5 SPI_I2S_INTSTAT 中断状态寄存器	405
19.5.6 SPI_I2S_INTEN 中断使能寄存器	406
19.5.7 SPI_I2S_INTCLR 中断清除寄存器.....	407
19.5.8 SPI_I2S_GCTL 全局控制寄存器	408
19.5.9 SPI_I2S_CCTL 通用控制寄存器.....	410
19.5.10 SPI_I2S_SPBRG 波特率发生器.....	411
19.5.11 SPI_I2S_RXDNR 接收数据个数寄存器.....	412
19.5.12 SPI_I2S_NSSR 从机片选寄存器.....	412
19.5.13 SPI_I2S_EXTCTL 数据控制寄存器	413
19.5.14 SPI_I2S_I2SCFGR I2S 配置寄存器	413
20 UART 通用异步收发器	415
20.1 简介	415
20.2 功能框图	415
20.3 主要特征	415
20.4 功能描述	416
20.4.1 UART 特性描述	417
20.4.2 分数波特率发生器.....	417
20.4.3 采样	417
20.4.4 容忍度	418
20.4.5 校验控制	418
20.4.6 发送器.....	418
20.4.7 接收器	420
20.4.8 自动波特率检测	421
20.4.9 九位数据通信	422
20.4.10 多处理器通信	422
20.4.11 单线半双工通信	424
20.4.12 智能卡	424
20.4.13 红外 IrDA 功能	426

20.5	中断	427
20.6	DMA.....	428
20.7	寄存器.....	428
20.7.1	寄存器总览	428
20.7.2	UART_TDR 发送数据寄存器	429
20.7.3	UART_RDR 接收数据寄存器.....	429
20.7.4	UART_CSR 当前状态寄存器.....	429
20.7.5	UART_ISR 中断状态寄存器	430
20.7.6	UART_IER 中断使能寄存器	432
20.7.7	UART_ICR 中断清除寄存器	433
20.7.8	UART_GCR 全局控制寄存器	434
20.7.9	UART_CCR 通用控制寄存器.....	435
20.7.10	UART_BRR 波特率寄存器	437
20.7.11	UART_FRA 分数波特率寄存器.....	437
20.7.12	UART_RXADDR 接收地址寄存器	438
20.7.13	UART_RXMASK 接收掩码寄存器	438
20.7.14	UART_SCR SCR 寄存器.....	439
20.7.15	UART_IDLR IDLE 数据长度寄存器	439
20.7.16	UART_ABRCR 自动波特率寄存器	440
20.7.17	UART_IRDA 红外功能控制寄存器.....	441
21	I2C 内部集成电路接口	442
21.1	简介	442
21.2	主要特征	442
21.3	功能描述	443
21.3.1	功能框图	443
21.3.2	信号描述	444
21.3.3	I2C 协议	444
21.3.4	工作模式	451
21.3.5	中断	456
21.3.6	DMA.....	457
21.4	寄存器.....	457
21.4.1	寄存器总览	457
21.4.2	I2C_CR 控制寄存器.....	458
21.4.3	I2C_TAR 目标地址寄存器.....	460
21.4.4	I2C_SAR 从地址寄存器.....	461
21.4.5	I2C_DR 数据命令寄存器	461
21.4.6	I2C_SSHR 标准模式时钟高电平计数寄存器	462
21.4.7	I2C_SSLR 标准模式时钟低电平计数寄存器	462
21.4.8	I2C_FSHR 快速模式时钟高电平计数寄存器	463
21.4.9	I2C_FSLR 快速模式时钟低电平计数寄存器	463
21.4.10	I2C_ISR 中断状态寄存器	463
21.4.11	I2C_IMR 中断屏蔽寄存器	464
21.4.12	I2C_RAWISR RAW 中断寄存器	464
21.4.13	I2C_RXTLR 接收阈值.....	466

21.4.14	I2C_TXTLR 发送阈值	466
21.4.15	I2C_ICR 组合和独立中断清除寄存器	467
21.4.16	I2C_RX_UNDER 清除 RX_UNDER 中断寄存器	467
21.4.17	I2C_RX_OVER 清除 RX_OVER 中断寄存器	467
21.4.18	I2C_TX_OVER 清除 TX_OVER 中断寄存器	468
21.4.19	I2C_RD_REQ 清除 RD_REQ 中断寄存器	468
21.4.20	I2C_TX_ABRT 清除 TX_ABRT 中断寄存器	469
21.4.21	I2C_RX_DONE 清除 RX_DONE 中断寄存器	469
21.4.22	I2C_ACTIV 清除 ACTIVITY 中断寄存器	469
21.4.23	I2C_STOP 清除 STOP_DET 中断寄存器	470
21.4.24	I2C_START 清除 START_DET 中断寄存器	470
21.4.25	I2C_GC 清除 GEN_CALL 中断寄存器	471
21.4.26	I2C_ENR 使能寄存器	471
21.4.27	I2C_SR 状态寄存器	472
21.4.28	I2C_TXFLR 发送缓冲级别寄存器	473
21.4.29	I2C_RXFLR 接收缓冲级别寄存器	473
21.4.30	I2C_HOLD SDA 保持时间寄存器	473
21.4.31	I2C_DMA DMA 控制寄存器	474
21.4.32	I2C_SETUP SDA 建立时间寄存器	474
21.4.33	I2C_GCR 广播呼叫 ACK 寄存器	475
21.4.34	I2C_SLVMASK 从地址掩码寄存器	475
21.4.35	I2C_SLVRCVADDR 从器件接收地址寄存器	476
22	CAN 控制器局域网	477
22.1	简介	477
22.2	主要特征	477
22.3	功能描述	478
22.3.1	功能框图	478
22.3.2	信号描述	479
22.3.3	CAN 工作模式	479
22.3.4	BasicCAN 模式	480
22.3.5	PeliCAN 模式	481
22.3.6	发送处理	484
22.3.7	接收管理	485
22.3.8	标识符过滤	485
22.3.9	报文存储	492
22.3.10	出错管理	495
22.3.11	位时序控制	498
22.3.12	仲裁丢失	499
22.3.13	中断	500
22.4	寄存器	502
22.4.1	寄存器总览	502
22.4.2	CAN_MOD 模式寄存器	503
22.4.3	CAN_CR 控制寄存器	504
22.4.4	CAN_CMRR 命令寄存器	504

22.4.5	CAN_SR 状态寄存器.....	505
22.4.6	CAN_IR 中断寄存器	507
22.4.7	CAN_IER 中断使能寄存器.....	508
22.4.8	GROUP0_ACR 验收代码寄存器组 0.....	509
22.4.9	GROUP0_AMR 验收屏蔽寄存器组 0	510
22.4.10	CAN_BTR0 位时序寄存器 0	511
22.4.11	CAN_BTR1 位时序寄存器 1	511
22.4.12	CAN_TXID0 发送识别码寄存器 0.....	512
22.4.13	CAN_TXID1 发送识别码寄存器 1.....	512
22.4.14	CAN_ALC 仲裁丢失捕捉寄存器	513
22.4.15	CAN_ECC 错误代码捕捉寄存器.....	514
22.4.16	CAN_EWLR 错误报警限制寄存器.....	516
22.4.17	CAN_RXERR RX 错误计数寄存器.....	516
22.4.18	CAN_TXERR TX 错误计数寄存器.....	517
22.4.19	CAN_SFF 发送帧信息寄存器	517
22.4.20	CAN_TXID0 发送识别码寄存器 0.....	518
22.4.21	CAN_TXID1 发送识别码寄存器 1.....	519
22.4.22	CAN_TXDATA0 发送数据寄存器 0.....	519
22.4.23	CAN_TXDATA1 发送数据寄存器 1.....	520
22.4.24	CAN_CDR 时钟分频寄存器.....	520
22.4.25	CAN_AFM0 滤波模式寄存器 0	521
22.4.26	CAN_AFM1 滤波模式寄存器 1	521
22.4.27	CAN_AFM2 滤波模式寄存器 2	522
22.4.28	CAN_FGA0 滤波组使能寄存器 0	522
22.4.29	CAN_FGA1 滤波组使能寄存器 1	522
22.4.30	CAN_FGA2 滤波组使能寄存器 2	523
22.4.31	GROUPx_ACR 验收代码寄存器组 x (x = 1 , ..19)	523
22.4.32	GROUPx_AMR 验收屏蔽寄存器组 x (x = 1 , ..19)	524
23	USB_FS 全速 USB 控制器	526
23.1	简介	526
23.2	主要特征	526
23.3	功能描述	527
23.3.1	功能框图	527
23.3.2	信号描述	527
23.3.3	USB_FS 典型硬件配置.....	527
23.3.4	USB_FS 软件编程接口	529
23.3.5	OTG 与 Host 模式操作方式	533
23.3.6	Host 模式操作示例	533
23.3.7	OTG 操作	536
23.3.8	中断	538
23.4	寄存器.....	539
23.4.1	寄存器总览	539
23.4.2	USB_FS_OTGISTAT OTG 中断状态寄存器.....	540
23.4.3	USB_FS_OTGICTRL OTG 中断控制寄存器.....	541

23.4.4	USB_FS_OTGSTAT OTG 状态寄存器.....	541
23.4.5	USB_FS_OTGCTRL OTG 控制寄存器	542
23.4.6	USB_FS_INTSTAT 中断状态寄存器.....	543
23.4.7	USB_FS_INTENB 中使能态寄存器.....	543
23.4.8	USB_FS_ERRSTAT 错误中断状态寄存器	544
23.4.9	USB_FS_ERRENB 错误中断使能寄存器.....	545
23.4.10	USB_FS_STAT 状态寄存器.....	545
23.4.11	USB_FS_CTL 控制寄存器.....	546
23.4.12	USB_FS_ADDR 地址寄存器	547
23.4.13	USB_FS_BDTPAGE1 缓冲区描述符表页寄存器 1	548
23.4.14	USB_FS_FRMNUML 低位帧数寄存器	548
23.4.15	USB_FS_FRMNUMH 高位帧数寄存器	548
23.4.16	USB_FS_TOKEN 令牌寄存器.....	549
23.4.17	USB_FS_SOFTHLD SOF 阀值寄存器.....	549
23.4.18	USB_FS_BDTPAGE2 缓冲区描述符表页寄存器 2	550
23.4.19	USB_FS_BDTPAGE3 缓冲区描述符表页寄存器 3	550
23.4.20	USB_FS_EPCTL0~15 端点控制寄存器 0~15	551
24	FSMC 存储器接口	552
24.1	设计概述	552
24.1.1	框图	552
24.1.2	接口描述	552
24.1.3	特性	553
24.2	功能描述	553
24.2.1	FSMC 数据位宽	553
24.2.2	FSMC 模式配置	554
24.2.3	NOR Flash/SRAM 模式	555
24.2.4	8080 协议模式	556
24.2.5	6800 协议模式	558
24.3	寄存器描述	559
24.3.1	存储器屏蔽寄存器 (SMSKR0)	560
24.3.2	存储器屏蔽寄存器 (SMSKR1)	561
24.3.3	存储器屏蔽寄存器 (SMSKR2)	562
24.3.4	存储器屏蔽寄存器 (SMSKR3)	563
24.3.5	存储器时序寄存器(SMTMGR_SET0)	564
24.3.6	存储器时序寄存器(SMTMGR_SET1)	565
24.3.7	存储器时序寄存器(SMTMGR_SET2)	565
24.3.8	存储器控制寄存器(SMCLR)	566
25	ADC 模拟/数字转换	568
25.1	简介	568
25.2	功能框图	568
25.3	主要特征	569
25.4	中断	569
25.5	DMA	570
25.6	功能描述	570

25.6.1	时钟	570
25.6.2	数据补偿	570
25.6.3	数据对齐	571
25.6.4	可编程分辨率	571
25.6.5	可编程采样时间	571
25.6.6	数据通道寄存器	572
25.6.7	通道选择	572
25.7	ADC 开关	572
25.7.1	普通通道转换	573
25.7.2	任意通道转换	576
25.7.3	注入通道转换	579
25.7.4	ADC 触发信号	580
25.7.5	模拟看门狗	580
25.7.6	内部温度传感器	581
25.7.7	内部电压传感器	581
25.8	寄存器	582
25.8.1	寄存器总览	582
25.8.2	ADC_ADDATA 数据寄存器	582
25.8.3	ADC_ADCFG 配置寄存器	583
25.8.4	ADC_ADCR 控制寄存器	584
25.8.5	ADC_ADCHS 通道选择寄存器	587
25.8.6	ADC_ADCMPR 模拟看门狗比较寄存器	589
25.8.7	ADC_ADSTA 状态寄存器	590
25.8.8	ADC_ADDRn 通道数据寄存器($n=0\sim 15$)	590
25.8.9	ADC_ADSTA_EXT 扩展状态寄存器	591
25.8.10	ADC_CHANY0 任意通道的通道选择寄存器 0	592
25.8.11	ADC_CHANY1 任意通道的通道选择寄存器 1	594
25.8.12	ADC_ANY_CFG 任意通道配置寄存器	595
25.8.13	ADC_ANY_CR 任意通道控制寄存器	596
25.8.14	ADC_SMPR1 采样配置寄存器 1	598
25.8.15	ADC_SMPR2 采样配置寄存器 2	599
25.8.16	ADC_JOFRn 注入通道数据补偿寄存器($n=0\sim 3$)	600
25.8.17	ADC_JSQR 注入通道序列寄存器	601
25.8.18	ADC_JADDATA 注入数据寄存器	601
25.8.19	ADC_JDRn 注入通道数据寄存器($n=0\sim 3$)	603
26	DAC 数字/模拟转换器	604
26.1	简介	604
26.2	主要特征	604
26.3	功能描述	606
26.3.1	DAC 通道的使能	606
26.3.2	使能 DAC 输出缓存	606
26.3.3	DAC 输出电压	606
26.3.4	DAC 触发源的选择	606
26.3.5	DMA 请求	607

26.3.6	DAC 数据格式.....	607
26.3.7	带三角波生成的 DAC 转换.....	608
26.3.8	带噪声波生成的 DAC 转换.....	609
26.3.9	DAC 转换.....	610
26.3.10	DAC 双通道转换模式.....	611
26.4	寄存器描述	613
26.4.1	DAC_CR 控制寄存器.....	613
26.4.2	DAC_SWTRIGR 软件触发寄存器	617
26.4.3	DAC_DHR12R1 通道 1 的 12 位右对齐数据保持寄存器.....	618
26.4.4	DAC_DHR12L1 通道 1 的 12 位左对齐数据保持寄存器	618
26.4.5	DAC_DHR8R1 通道 1 的 8 位右对齐数据保持寄存器	618
26.4.6	DAC_DHR12R2 通道 2 的 12 位右对齐数据保持寄存器	619
26.4.7	DAC_DHR12L2 通道 2 的 12 位左对齐数据保持寄存器	619
26.4.8	DAC_DHR8R2 通道 2 的 8 位右对齐数据保持寄存器	620
26.4.9	DAC_DHR12LD 双通道的 12 位左对齐数据保持寄存器	620
26.4.10	DAC_DHR12RD 双通道的 12 位右对齐数据保持寄存器.....	621
26.4.11	DAC_DHR8RD 双通道的 8 位右对齐数据保持寄存器.....	621
26.4.12	DAC_DOR1 通道 1 数据输出寄存器.....	621
26.4.13	DAC_DOR2 通道 2 数据输出寄存器.....	622
27	COMP 比较器	623
27.1	简介	623
27.2	功能框图	623
27.3	主要特征	623
27.4	功能描述	624
27.4.1	比较器时钟和复位.....	624
27.4.2	比较器开关控制	624
27.4.3	比较器输入和输出.....	624
27.4.4	比较器通道选择	624
27.4.5	中断和唤醒	625
27.4.6	功耗模式	625
27.4.7	比较器锁定机制	625
27.4.8	迟滞电压	626
27.5	比较器寄存器描述	626
27.5.1	比较器控制状态寄存器 (COMP _x _CSR) (x=1,2)	626
27.5.2	比较器外部参考电压寄存器 (COMP_CRV)	629
27.5.3	比较器轮询寄存器 (COMP _x _POLL) (x=1,2)	630
28	CRC 循环冗余校验计算单元	632
28.1	简介	632
28.2	主要特征	632
28.3	功能描述	632
28.3.1	功能框图	632
28.3.2	功能概述	633
28.3.3	使用方法	633
28.4	寄存器.....	634

28.4.1	寄存器总览	634
28.4.2	CRC_DR CRC 数据寄存器	634
28.4.3	CRC_IDR CRC 独立数据寄存器	635
28.4.4	CRC_CR CRC 控制寄存器	635
28.4.5	CRC_MIR CRC 中间数据寄存器	636
29	DBG 调试支持	637
29.1	简介	637
29.2	功能描述	637
29.2.1	功能框图	637
29.2.2	SWD 内部上拉与下拉	638
29.2.3	SWJ 调试端口	638
29.3	ID 代码和锁定机制	638
29.3.1	微控制器设备 ID 编码	638
29.3.2	Cortex JEDEC-106 ID 编码	639
29.4	SWD 调试端口	639
29.4.1	SWD 协议介绍	639
29.4.2	SW 协议序列	639
29.4.3	SW-DP 状态机 (Reset, Idle states, ID code)	640
29.4.4	DP 和 AP 读 / 写访问	640
29.4.5	SW-DP 寄存器	641
29.4.6	SW-AP 寄存器	641
29.5	JTAG 调试端口	641
29.6	MCU 调试模块 (DBGMCU)	642
29.6.1	低功耗模式的调试支持	642
29.6.2	支持定时器、看门狗	642
29.7	寄存器	643
29.7.1	寄存器总览	643
29.7.2	DBG_IDCODE ID 编码寄存器	643
29.7.3	DBG_CR 控制寄存器	643
30	Device Electronic Signature 器件电子签名	646
30.1	器件电子签名简介	646
30.2	寄存器描述	646
30.2.1	UID1 唯一标识码	646
30.2.2	UID2 唯一标识码	647
30.2.3	UID3 唯一标识码	647
31	修订记录	648

附图目录

图 1-1 系统架构框图	30
图 2-1 ISP 方式编程流程图	40
图 2-2 IAP 编程流程	41
图 3-1 电源控制功能框图	58
图 3-2 上电复位和掉电复位波形图	60
图 3-3 PVD 阈值波形图	61
图 3-4 Standby 模式唤醒滤波功能波形图	70
图 4-1 复位功能框图	78
图 4-2 时钟树	81
图 4-3 高速晶振/陶瓷谐振器	82
图 4-4 外部高速输入时钟	82
图 4-5 低速晶振/陶瓷谐振器	85
图 4-6 外部低速输入时钟	85
图 6-1 EXTI 结构框图	126
图 7-1 DMA 功能框图	134
图 9-1 CRS 功能框图	151
图 9-2 CRS 计数器状态图	156
图 10-1 标准 I/O 端口	164
图 10-2 输入浮空/上拉/下拉配置	166
图 10-3 输出配置	167
图 10-4 复用功能配置	168
图 10-5 模拟输入	169
图 11-1 TIMx 结构图	176
图 11-2 时钟选择	178
图 11-3 外部时钟模式 1 下的控制电路	179
图 11-4 外部时钟模式 2 下的控制电路	179
图 11-5 自动预装载	180
图 11-6 递增计数模式 (UDIS=0)	180
图 11-7 递增计数模式 (UDIS=1 禁止产生更新事件)	181
图 11-8 递减计数模式 (UDIS=0)	181
图 11-9 递减计数模式 (UDIS=1 禁止产生更新事件)	181
图 11-10 中央计数模式 (UDIS=0)	182
图 11-11 中央计数模式 (UDIS=1 禁止产生更新事件)	182
图 11-12 中央对齐模式重复计数时序图	183
图 11-13 边沿对齐模式递增计数时序图	183
图 11-14 边沿对齐模式递减计数时序图	184
图 11-15 TIMx 输入捕获结构图	184
图 11-16 PWM 输入模式时序	186
图 11-17 比较输出部分结构图	187
图 11-18 比较输出模式, OC1 信号在匹配时翻转	188

图 11-19 边沿对齐递增计数时 PWM 模式 1 的波形.....	189
图 11-20 边沿对齐递减计数时 PWM 模式 1 的波形.....	190
图 11-21 中央对齐 PWM 模式 1 的波形.....	191
图 11-22 移相功能示意图.....	191
图 11-23 使用 COM 事件产生六步 PWM (OSSR = 1)	192
图 11-24 死区插入.....	193
图 11-25 响应刹车的输出 (OISx=0, OISxN=0)	196
图 11-26 响应刹车的输出 (OISx=0, OISxN=1)	196
图 11-27 响应刹车的输出 (OISx=1, OISxN=0)	196
图 11-28 响应刹车的输出 (OISx=1, OISxN=1)	196
图 11-29 外部事件清除 OCxREF	197
图 11-30 单脉冲模式	197
图 11-31 编码器模式下的计数器时序图.....	199
图 11-32 IC1FP1 反相编码器接口模式时序图.....	199
图 11-33 复位模式的控制时序图.....	200
图 11-34 门控模式下的控制时序图	201
图 11-35 触发器模式下的控制时序图	201
图 11-36 外部时钟模式 2+从模式 (触发模式) 控制时序图	202
图 11-37 定时器间互联	202
图 11-38 使用 TIM3 作为 TIM1 的预分频器	203
图 11-39 使用 TIM1 使能 TIM2	203
图 11-40 使用 TIM1 的更新事件启动 TIM2	204
图 11-41 TIM1 的 TI1 同步启动 TIM1 和 TIM2.....	204
图 11-42 (TI1 异或输入) 输入捕获波形图.....	205
图 11-43 霍尔传感器接口实例	206
图 12-1 TIMx 结构图	239
图 12-2 时钟选择	241
图 12-3 外部时钟模式 1 下的控制电路	242
图 12-4 外部时钟模式 2 下的控制电路	242
图 12-5 自动预装载.....	243
图 12-6 递增计数模式 (UDIS=0)	243
图 12-7 递增计数模式 (UDIS=1 禁止产生更新事件)	244
图 12-8 递减计数模式 (UDIS=0)	244
图 12-9 递减计数模式 (UDIS=1 禁止产生更新事件)	244
图 12-10 中央计数模式 (UDIS=0)	245
图 12-11 中央计数模式 (UDIS=1 禁止产生更新事件)	245
图 12-12 TIMx 输入捕获结构图	245
图 12-13 PWM 输入模式时序	247
图 12-14 比较输出部分结构图	248
图 12-15 比较输出模式, OC1 信号在匹配时翻转.....	249
图 12-16 边沿对齐递增计数时 PWM 模式 1 的波形	250
图 12-17 边沿对齐递减计数时 PWM 模式 1 的波形	251
图 12-18 中央对齐 PWM 模式 1 的波形.....	252
图 12-19 外部事件清除 OCxREF.....	253

图 12-20 单脉冲模式.....	253
图 12-21 编码器模式下的计数器时序图.....	255
图 12-22 IC1FP1 反相编码器接口模式时序图.....	255
图 12-23 复位模式的控制时序图	256
图 12-24 门控模式下的控制时序图.....	257
图 12-25 触发器模式下的控制时序图	257
图 12-26 外部时钟模式 2+从模式（触发模式）控制时序图.....	258
图 12-27 （TI1 异或输入）输入捕获波形图	259
图 13-1 TIMx 结构图.....	285
图 13-2 时钟选择	287
图 13-3 外部时钟模式 1 下的控制电路	288
图 13-4 外部时钟模式 2 下的控制电路	288
图 13-5 自动预装载.....	289
图 13-6 递增计数模式（UDIS=0）	289
图 13-7 递增计数模式（UDIS=1 禁止产生更新事件）	290
图 13-8 递减计数模式（UDIS=0）	290
图 13-9 递减计数模式（UDIS=1 禁止产生更新事件）	290
图 13-10 中央计数模式（UDIS=0）	291
图 13-11 中央计数模式（UDIS=1 禁止产生更新事件）	291
图 13-12 TIMx 输入捕获结构图	291
图 13-13 PWM 输入模式时序	293
图 13-14 比较输出部分结构图	294
图 13-15 比较输出模式，OC1 信号在匹配时翻转	295
图 13-16 边沿对齐递增计数时 PWM 模式 1 的波形	296
图 13-17 边沿对齐递减计数时 PWM 模式 1 的波形	297
图 13-18 中央对齐 PWM 模式 1 的波形.....	298
图 13-19 外部事件清除 OCxREF.....	299
图 13-20 单脉冲模式.....	299
图 13-21 编码器模式下的计数器时序图.....	301
图 13-22 IC1FP1 反相编码器接口模式时序图.....	301
图 13-23 复位模式的控制时序图	302
图 13-24 门控模式下的控制时序图	303
图 13-25 触发器模式下的控制时序图	303
图 13-26 外部时钟模式 2+从模式（触发模式）控制时序图.....	304
图 13-27 （TI1 异或输入）输入捕获波形图	305
图 14-1 TIMx 结构图	332
图 14-2 自动预装载.....	333
图 14-3 递增计数模式（UDIS=0）	333
图 14-4 递增计数模式（UDIS=1 禁止产生更新事件）	334
图 15-1 SDIO 功能框图	340
图 15-2 SDIO“无响应”和“无数据”操作	342
图 15-3 SDIO（多）数据块读操作	342
图 15-4 SDIO 连续读操作	343
图 15-5 SDIO（多）数据块写操作	343

图 16-1 功能框图	362
图 16-2 流程框图	364
图 17-1 WWDG 功能框图	371
图 17-2 WWDG 超时时间坐标图	373
图 18-1 RTC 功能框图	377
图 18-2 RTC 秒和闹钟波形图示例, RTC_PRL = 0004, RTC_ALR = 002	379
图 18-3 RTC 溢出波形图示例, RTC_PRL = 0004	379
图 19-1 SPI_I2S 功能框图	388
图 19-2 单主和单从应用	389
图 19-3 数据时钟时序图	390
图 19-4 飞利浦标准示意图	397
图 19-5 MSB 对齐标准示意图	397
图 19-6 LSB 对齐标准示意图	398
图 19-7 PCM 标准示意图	398
图 19-8 I2S 从模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器)	399
图 19-9 I2S 主模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器)	399
图 19-10 I2S 时钟预分频器示意图	400
图 20-1 UART 功能框图	415
图 20-2 UART 时序	417
图 20-3 RX 引脚采样方法	418
图 20-4 发送时状态位变化	420
图 20-5 UART 奇偶校验方框图	425
图 20-6 UART 采样 NACK 信号方框图	426
图 20-7 普通模式下 IrDA 发送和接收图	427
图 21-1 I2C 功能框图	443
图 21-2 起始和停止条件	444
图 21-3 7 位地址格式	445
图 21-4 10 位地址格式	445
图 21-5 主发送协议	446
图 21-6 主接收协议	446
图 21-7 带 RESTART (SR) 信号的主发送和接收协议	447
图 21-8 起始字节传输	447
图 21-9 I2C_DR 寄存器	448
图 21-10 主发送, TX FIFO 为空	448
图 21-11 主接收, TX FIFO 为空	448
图 21-12 主发送, 发生 RESTART	449
图 21-13 主接收, 发生 RESTART	449
图 21-14 两个主器件仲裁	449
图 21-15 时钟同步 (示意图)	450
图 21-16 时钟同步 (时序图)	450
图 21-17 SCL 生成时序	451
图 21-18 流程图 (I2C 接口作为从器件)	453
图 21-19 流程图 (I2C 接口作为主器件)	455
图 21-20 I2C 中断机制	457

图 22-1 CAN 网络拓扑结构	477
图 22-2 CAN 功能框图	478
图 22-3 CAN 标识符接收示例	486
图 22-4 接收标准结构信息时的单个滤波器配置	487
图 22-5 单滤波器配置，接收扩展帧信息	488
图 22-6 接收标准结构信息时的双个滤波器配置	489
图 22-7 双滤波器配置，接收扩展帧信息	490
图 22-8 标准帧和扩展帧格式配置在发送缓冲器的列表	494
图 22-9 RXFIFO 中的信息存储举例	495
图 22-10 错误码捕捉功能举例	496
图 22-11 仲裁丢失解释举例	500
图 23-1 USB_FS 功能框图	527
图 23-2 仅主机硬件连接示意图	528
图 23-3 仅从设备硬件连接示意图	528
图 23-4 OTG 硬件连接示意图	529
图 23-5 缓冲区描述表	530
图 23-6 USB 令牌传输示意图	533
图 23-7 OTG A-device 操作方式	536
图 23-8 OTG B-device 操作方式	538
图 24-1 FSMC 框图	552
图 24-2 NOR FLASH/SRAM 写时序	556
图 24-3 NOR FLASH/SRAM 读时序	556
图 24-4 8080 写时序	557
图 24-5 8080 读时序	558
图 24-6 6800 写时序	559
图 24-7 6800 读时序	559
图 25-1 ADC 系统框图	568
图 25-2 数据对齐方式	571
图 25-3 单次转换模式时序图	573
图 25-4 单周期扫描下使能通道转换时序图（通道方向从高到低）	574
图 25-5 单周期扫描下使能通道转换时序图（通道方向从低到高）	574
图 25-6 连续扫描模式使能通道转换时序图（通道方向由低到高）	575
图 25-7 连续扫描模式使能通道转换时序图（通道方向由高到低）	575
图 25-8 单次转换模式下通道转换时序图	576
图 25-9 单周期扫描下通道转换时序图	577
图 25-10 连续扫描模式下通道转换时序图	578
图 25-11 连续扫描模式下动态更新配置时序图	578
图 25-12 自动注入通道转换单周期扫描通道转换时序图	579
图 25-13 自动注入通道转换连续转换时序图	580
图 26-1 DAC 单通道功能框图	605
图 26-2 DAC 单通道模式的数据保持寄存器的不同数据格式	608
图 26-3 DAC 双通道模式的数据保持寄存器的不同数据格式	608
图 26-4 DAC 三角波生成	609
图 26-5 带三角波生成的 DAC 转换（使能软件触发）	609

图 26-6 DAC LFSR 寄存器算法.....	610
图 26-7 带 LFSR 波形生成的 DAC 转换（使能软件触发）.....	610
图 26-8 TEN=0 触发关闭时 DAC 转换的时间示意图.....	611
图 27-1 比较器框图.....	623
图 27-2 比较器的迟滞.....	626
图 28-1 CRC 功能框图	632
图 29-1 调试功能框图	637

表格目录

表 1-1 存储器映像.....	31
表 1-2 启动模式	34
表 2-1 Flash 存储空间	35
表 2-2 信息块.....	36
表 2-3 选项字节组织结构.....	36
表 2-4 USER 的位含义	37
表 2-5 Latency 设置关系.....	39
表 2-6 编程方式	39
表 2-7 保护设置的状态变化	42
表 2-8 Flash 读保护状态.....	50
表 2-9 Flash 解除读保护状态.....	50
表 2-10 写保护区域.....	51
表 2-11 FLASH 寄存器概览	51
表 3-1 低功耗模式列表	62
表 3-2 低功耗运行模式	64
表 3-3 SLEEPNOW 模式	64
表 3-4 SLEEPONEXIT 模式	65
表 3-5 Low Power SLEEPNOW 模式.....	65
表 3-6 Low Power SLEEPONEXIT 模式	66
表 3-7 停机模式	67
表 3-8 深度停机模式.....	68
表 3-9 待机模式	69
表 3-10 电源控制寄存器概览	70
表 4-1 RCC 全局中断表.....	87
表 4-2 MCO 与时钟源对应关系	89
表 4-3 RCC 寄存器概览.....	90
表 5-1 SYSCFG 寄存器概览.....	118
表 6-1 异常向量表.....	126
表 6-2 中断向量表.....	127
表 6-3 EXTI 触发源	130
表 6-4 EXTI 寄存器总览	131
表 7-1 DMA 中断请求.....	135
表 7-2 DMA1 各个通道的 DMA 请求一览.....	135
表 7-3 DMA2 各个通道的 DMA 请求一览.....	136
表 7-4 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	138
表 7-5 DMA 寄存器概览	140
表 8-1 BKP 寄存器概览	147
表 9-1 CRS 中断请求.....	157
表 9-2 CRS 寄存器概览	158

表 10-1 端口位配置表（port0 为例）	158
表 10-2 SWD 复用功能重映射	158
表 10-3 GPIO 寄存器概览.....	158
表 11-1 数字滤波器宽度与 ICxF 的对应关系表.....	185
表 11-2 死区时间计算	192
表 11-3 当 MOE=1, OSSI=0/1, OSSR=0 时:	194
表 11-4 当 MOE=1, OSSI=0/1, OSSR=1 时:	194
表 11-5 当 MOE=0, OSSI=0, OSSR=0/1 时:	194
表 11-6 当 MOE=0, OSSI=1, OSSR=0/1 时:	195
表 11-7 计数方向与编码器信号的关系.....	198
表 11-8 中断事件一览表.....	207
表 11-9 TIMx 寄存器概览.....	208
表 11-10 TIMx 内部触发连接	214
表 11-11 ICx 极性/电平选择表	229
表 12-1 数字滤波器宽度与 ICxF 的对应关系表	246
表 12-2 计数方向与编码器信号的关系	254
表 12-3 中断事件一览表	259
表 12-4 TIMx 寄存器概览.....	260
表 12-5 TIMx 内部触发连接	266
表 12-6 ICx 极性/电平选择表	280
表 13-1 数字滤波器宽度与 ICxF 的对应关系表	292
表 13-2 计数方向与编码器信号的关系	300
表 13-3 中断事件一览表	305
表 13-4 TIMx 寄存器概览.....	306
表 13-5 TIMx 内部触发连接	312
表 13-6 ICx 极性/电平选择表	326
表 14-1 中断事件一览表	334
表 14-2 TIMx 寄存器概览.....	334
表 15-1 SDIO 信号描述表.....	340
表 15-2 SDIO 中断事件	340
表 15-3 IO 寄存器[7: 0]的详细描述	343
表 15-4 IOMBCTL[2: 0]和 IO[7: 6]的详细描述表	344
表 15-5 SDIO 寄存器概览.....	345
表 16-1 IWDG 超时时间（40kHz 的输入时钟（LSI））	364
表 16-2 IWDG 寄存器概览.....	365
表 17-1 超时时间	373
表 17-2 WWDG 寄存器概览	373
表 18-1 RTC 寄存器概览	380
表 19-1 波特率公式.....	394
表 19-2 SPI 状态.....	394
表 19-3 I2S 比特率计算公式	401
表 19-4 I2S 音频采样率 Fs 与 FI2SCLK 计算关系	401
表 19-5 I2S 音频采样率误差表	401
表 19-6 I2S 中断状态	402

表 19-7 SPI_I2S 寄存器概览	403
表 20-1 UART 中断请求	427
表 20-2 UART 寄存器概览.....	428
表 21-1 引脚定义	444
表 21-2 I2C 首字节	445
表 21-3 中断位的置位和清除	456
表 21-4 I2C 寄存器概览	457
表 21-5 DISSLAVE 和 MASTER 配置.....	460
表 22-1 CAN 信号描述表	479
表 22-2 BasicCAN 模式寄存器权限分配表	480
表 22-3 PeliCAN 模式寄存器权限分配表.....	482
表 22-4 BasicCAN 模式里的 RX 和 TX 缓冲器	492
表 22-5 接收时可能出现的错误	497
表 22-6 发送时可能出现的错误	497
表 22-7 BasicCAN 寄存器概览	502
表 22-8 PeliCAN 寄存器概览.....	502
表 23-1 术语表	526
表 23-2 USB 信号描述表	527
表 23-3 BDT 地址计算	531
表 23-4 BDT 地址计算域说明	531
表 23-5 BD 格式.....	531
表 23-6 OTG A-device 操作状态描述表.....	537
表 23-7 OTG B-device 操作状态描述表	538
表 23-8 USB 寄存器概览	539
表 24-1 接口描述	552
表 24-2 NOR FLASH 控制器	553
表 24-3 BANK 地址划分	553
表 24-4 存储器位宽	554
表 24-5 NOR FLASH 16bit 示例	554
表 24-6 SRAM 16bit 示例	555
表 24-7 NOR FLASH/SRAM 接口	555
表 24-8 NOR FLASH/SRAM 时序参数说明.....	555
表 24-9 8080 协议接口	556
表 24-10 8080 地址说明	557
表 24-11 8080 时序参数说明	557
表 24-12 6800 协议接口	558
表 24-13 6800 地址说明	558
表 24-14 6800 时序参数说明	559
表 24-15 寄存器映射表	559
表 25-1 中断事件一览表	569
表 25-2 数据分辨率与数据补偿左对齐关系	570
表 25-3 寄存器概览.....	582
表 26-1 DAC 管脚说明	605
表 26-2 外部触发	606

表 26-3 DAC 寄存器概览	613
表 27-1 COMP 寄存器概览.....	626
表 28-1 CRC 寄存器概览.....	634
表 29-1 SWJ 调试端口管脚	638
表 29-2 ID 编码	638
表 29-3 8bit 请求包	639
表 29-4 3bit 应答包	640
表 29-5 33bit 数据包	640
表 29-6 SW-DP 寄存器	641
表 29-7 JTAG 调试端口寄存器映射	641
表 29-8 JTAG-DP 寄存器.....	642
表 29-9 DBG 寄存器概览	643
表 30-1 存储器容量寄存器概览	646
表 31-1 修订记录表.....	648

1 系统和存储器架构

1.1 系统架构简介

MM32F3270 系列是基于 Arm®Cortex®-M3 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。MM32F3270 系列采用矩阵总线结构，该矩阵包括五个 AHB 主机：CPU、DMA1、DMA2、和 USB，从机分别是 SRAM、闪存存储、FSMC、AHB 总线（含 AHB 到 APB 的总线桥）以及连接在 APB 总线的各种设备。

1.1.1 ICode 总线 (IBUS)

该总线将 CPU 内核的指令总线与闪存指令接口相连接。指令预取在此总线上完成。

1.1.2 DCode 总线 (DBUS)

该总线将 CPU 内核 DCode 总线与闪存存储器的数据接口相连接（常量加载和调试访问）。

1.1.3 系统总线 (SBUS)

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

1.1.4 DMA 总线

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

1.1.5 总线矩阵 (BUS Matrix)

总线矩阵管理着内核系统总线与 DMA 总线的访问仲裁，总线矩阵由主模块总线及从模块总线组成。AHB 外设通过总线矩阵与系统总线相连，允许 DMA 访问。

1.1.6 AHB 到 APB 桥 (AHB-APB APBx)

AHB 到 APB 桥在 AHB 与 APB 总线间提供同步连接。在每次复位之后，所有的外设时钟都关闭（除了 SRAM 及 Flash 外）。在用一个外设前，用户必须打开相应的 RCC_AHBxENR 或

RCC_APBxENR 寄存器中时钟使能位。

注：当对 APB 寄存器进行 8 位或者 16 位访问时，该访问会被自动转换成 32 位的访问：桥会自动将 16 位或者 8 位的数据扩展以配合 32 位的宽度。

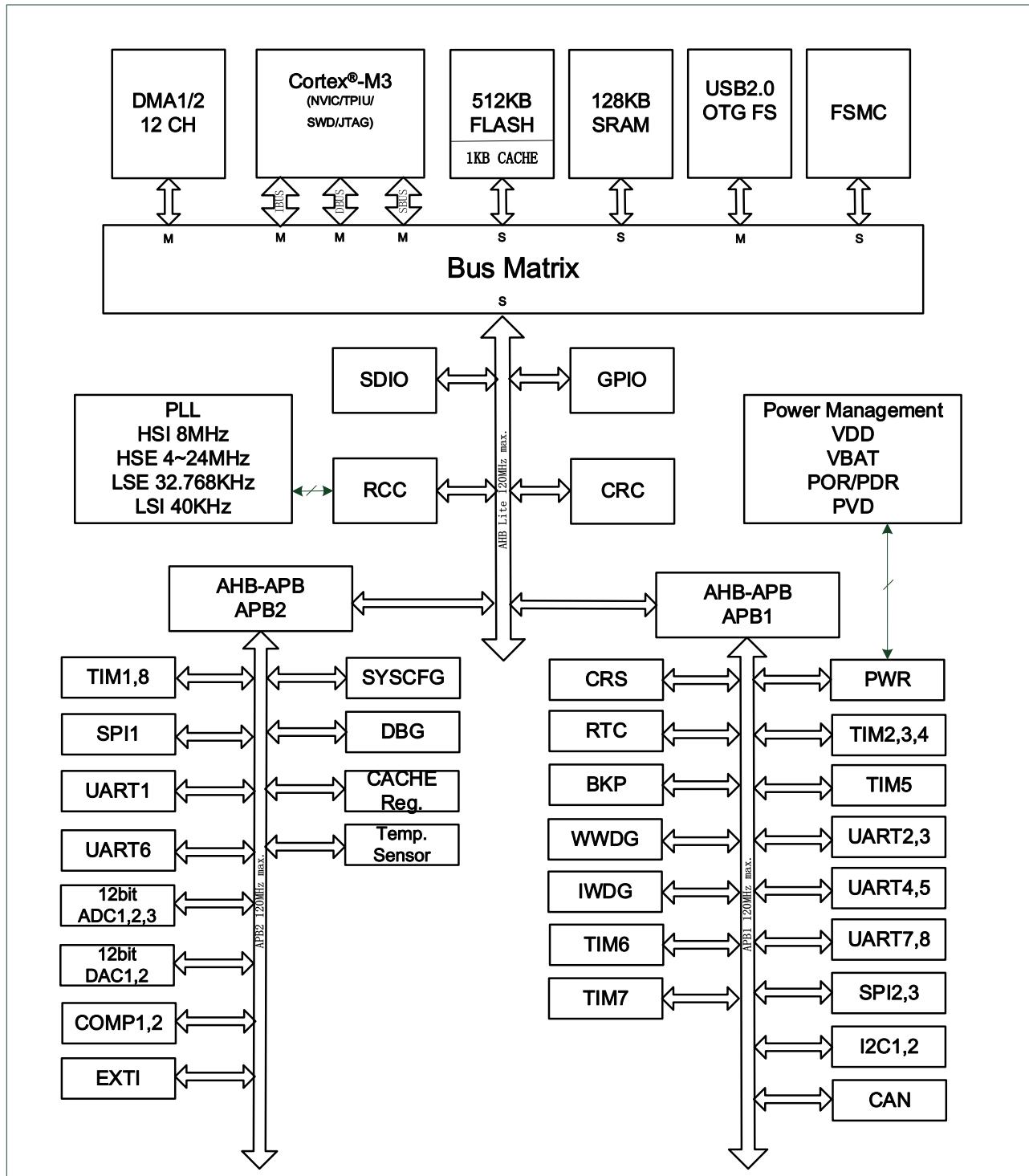


图 1-1 系统架构框图

1.2 存储器介绍

程序存储器、数据存储器、寄存器和 I/O 接口都位于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。

1.2.1 存储器映像和寄存器地址

存储器映像请参考各外设对应章节的存储器映像图。

表 1-1 存储器映像

总线	编址范围	大小	外设
FLASH	0x0000 0000 - 0x0007 FFFF	512 KB	根据 BOOT0/1 引脚的电平可映射到片内 FLASH 存储区、SRAM 或系统存储区中的一个
	0x0008 0000 - 0x07FF FFFF	~127 MB	Reserved
	0x0800 0000 - 0x0807 FFFF	512 KB	片内 FLASH 主存储区
	0x0808 0000 - 0x080F FFFF	512 KB	Reserved
	0x0810 0000 - 0x0810 0FFF	4 KB	Reserved
	0x0810 1000 - 0x0FFF FFFF	~127 MB	Reserved
	0x1000 0000 - 0x1FFD FFFF	~255 MB	Reserved
	0x1FFE 0000 - 0x1FFE 0FFF	4 KB	Reserved
	0x1FFE 1000 - 0x1FFE 1FFF	4 KB	Security memory
	0x1FFE 2000 - 0x1FFF E7FF	114 KB	Reserved
	0x1FFF E800 - 0x1FFF F7FF	4 KB	系统存储区
	0x1FFF F800 - 0x1FFF F9FF	0.5 KB	Option bytes
	0x1FFF FA00 - 0x1FFF FFFF	1.5 KB	Reserved
SRAM	0x2000 0000 - 0x2000 3FFF	16 KB	SRAM-2
	0x2000 4000 - 0x2001 FFFF	112 KB	SRAM-1
	0x2002 0000 - 0x3FFF FFFF	~511 MB	Reserved
APB1	0x4000 0000 - 0x4000 03FF	1 KB	TIM2
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3
	0x4000 0800 - 0x4000 0BFF	1 KB	TIM4
	0x4000 0C00 - 0x4000 0FFF	1 KB	TIM5
	0x4000 1000 - 0x4000 13FF	1 KB	TIM6
	0x4000 1400 - 0x4000 17FF	1 KB	TIM7
	0x4000 1800 - 0x4000 27FF	4 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

总线	编址范围	大小	外设
APB1	0x4000 3C00 - 0x4000 3FFF	1 KB	SPI3
	0x4000 4000 - 0x4000 43FF	1 KB	Reserved
	0x4000 4400 - 0x4000 47FF	1 KB	UART2
	0x4000 4800 - 0x4000 4BFF	1 KB	UART3
	0x4000 4C00 - 0x4000 4FFF	1 KB	UART4
	0x4000 5000 - 0x4000 53FF	1 KB	UART5
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1
	0x4000 5800 - 0x4000 5BFF	1 KB	I2C2
	0x4000 5C00 - 0x4000 63FF	2 KB	Reserved
	0x4000 6400 - 0x4000 67FF	1 KB	CAN
	0x4000 6800 - 0x4000 6BFF	1 KB	Reserved
	0x4000 6C00 - 0x4000 6FFF	1 KB	CRS
	0x4000 7000 - 0x4000 73FF	1 KB	PWR
	0x4000 7400 - 0x4000 77FF	1 KB	DAC
	0x4000 7800 - 0x4000 7BFF	1 KB	UART7
	0x4000 7C00 - 0x4000 7FFF	1 KB	UART8
APB2	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI
	0x4001 0800 - 0x4001 23FF	7 KB	Reserved
	0x4001 2400 - 0x4001 27FF	1 KB	ADC1
	0x4001 2800 - 0x4001 2BFF	1 KB	ADC2
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1
	0x4001 3400 - 0x4001 37FF	1 KB	TIM8
	0x4001 3800 - 0x4001 3BFF	1 KB	UART1
	0x4001 3C00 - 0x4001 3FFF	1 KB	UART6
	0x4001 4000 - 0x4001 43FF	1 KB	COMP
	0x4001 4400 - 0x4001 4BFF	2 KB	Reserved
	0x4001 4C00 - 0x4001 4FFF	1 KB	ADC3
	0x4001 5000 - 0x4001 5FFF	4 KB	Reserved
	0x4001 6000 - 0x4001 63FF	1 KB	CACHE Register
	0x4001 6400 - 0x4001 7FFF	7 KB	Reserved
AHB1	0x4001 8000 - 0x4001 83FF	1 KB	SDIO
	0x4001 8400 - 0x4001 FFFF	31 KB	Reserved
	0x4002 0000 - 0x4002 03FF	1 KB	DMA1
	0x4002 0400 - 0x4002 07FF	1 KB	DMA2
	0x4002 0800 - 0x4002 0FFF	2 KB	Reserved
	0x4002 1000 - 0x4002 13FF	1 KB	RCC
	0x4002 1400 - 0x4002 1FFF	3 KB	Reserved
	0x4002 2000 - 0x4002 23FF	1 KB	Flash memory interface
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved
	0x4002 3000 - 0x4002 33FF	1 KB	CRC

总线	编址范围	大小	外设
AHB2	0x4002 3400 - 0x4002 7FFF	19 KB	Reserved
	0x4002 8000 - 0x4002 9FFF	8 KB	Reserved
	0x4002 A000 - 0x4003 FFFF	88 KB	Reserved
	0x4004 0000 - 0x4004 03FF	1 KB	GPIOA
	0x4004 0400 - 0x4004 07FF	1 KB	GPIOB
	0x4004 0800 - 0x4004 0BFF	1 KB	GPIOC
	0x4004 0C00 - 0x4004 0FFF	1 KB	GPIOD
	0x4004 1000 - 0x4004 13FF	1 KB	GPIOE
	0x4004 1400 - 0x4004 17FF	1 KB	GPIOF
	0x4004 1800 - 0x4004 1BFF	1 KB	GPIOG
	0x4004 1C00 - 0x4004 1FFF	1 KB	GPIOH
	0x4004 2000 - 0x47FF FFFF	~127 MB	Reserved
AHB3	0x5000 0000 - 0x5003 FFFF	256 KB	USB OTG FS
	0x5006 0000 - 0x5006 03FF	1 KB	Reserved
	0x5006 0800 - 0x5006 0BFF	1 KB	Reserved
FSMC	0x6000 0000 - 0x63FF FFFF	64 MB	FSMC Bank
	0x6400 0000 - 0x67FF FFFF	64 MB	FSMC Bank
	0x6800 0000 - 0x6BFF FFFF	64 MB	FSMC Bank
	0x6C00 0000 - 0x6FFFF FFFF	64 MB	FSMC Bank
	0x7000 0000 - 0x9FFFF FFFF	768 MB	Reserved
	0xA000 0000 - 0xA000 0FFF	4 KB	FSMC Register
	0xA000 1000 - 0xA000 13FF	1 KB	Reserved

1.2.2 内置的 SRAM

内置最大可到 128K 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。

1.2.3 闪存存储器概述

闪存存储器分为两个存储区域：

- 由应用数据和用户数据区组成的主闪存存储块
- 由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。

1.2.4 启动配置

在芯片中，可通过 BOOT0 及 BOOT1 脚的配置选择三种不同的启动模式，如下表所示：

表 1-2 启动模式

启动模式	启动模式引脚选择	
	BOOT1	BOOT0
主闪存存储器	x	0
系统存储器	0	1
内置 SRAM	1	1

用户通过设置 BOOT0 和 BOOT1 引脚电平来选择三种启动模式。其中在待机模式中需要保持启动模式引脚的设置值，因为每一次从待机模式唤醒时，CPU 会重新采样 BOOT0 和 BOOT1 引脚电平来重新确定启动模式。

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

启动模式主要分为三种：主闪存存储器，系统存储器和内置 SRAM。

主闪存存储器启动 (BOOT0=0): 主闪存存储器的起始地址是 0x0800 0000，当其被选为启动模式时，被映射到启动存储空间 (0x0000 0000)，但是闪存存储器的内容依旧可以从起始地址 (0x0800 0000) 访问，即当主闪存存储器被选为启动模式，启动地址和起始地址都可以访问闪存存储器。

系统存储器启动 (BOOT1=0 & BOOT0=1): 系统存储器的起始地址是 0x1FFF E800，当其被选为启动模式时，被映射到启动存储空间 (0x0000 0000)，但是系统存储器的内容依旧可以从起始地址 (0x1FFF E800) 访问，即当系统存储器被选为启动模式，启动地址和起始地址都可以访问系统存储器。

内置 SRAM 启动 (BOOT1=1 & BOOT0=1): 内置 SRAM 的起始地址是 0x2000 0000，当其被选为启动模式时，被映射到启动存储空间 (0x0000 0000)，但是内置 SRAM 的内容依旧可以从起始地址 (0x2000 0000) 访问，即当内置 SRAM 被选为启动模式，启动地址和起始地址都可以访问内置 SRAM。

1.2.5 引导程序

出厂引导程序存放在系统存储器中，可以通过串口进行 ISP 编程。

2 Embedded FLASH 嵌入式闪存

2.1 简介

嵌入式闪存支持高达 512K 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。闪存的控制器支持读操作、页擦除、整片擦除，可通过 16 位（半字）方式编程写入闪存，其擦写寿命可达 20000 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。

2.2 闪存构成与说明

2.2.1 闪存构成

- 闪存空间由 128 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 512 页（每页 1K 字节）或 128 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 2-1 所示。

表 2-1 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
	写保护区 8	页 31	0x0800 7C00 - 0x0800 7FFF	1K
		页 32	0x0800 8000 - 0x080083FF	1K
	
	写保护区 127	页 508	0x0807 F000 - 0x0807 F3FF	1K
		页 509	0x0807 F400 - 0x0807 F7FF	1K
		页 510	0x0807 F800 - 0x0807 FBFF	1K

模块	区块名称	页名称	地址	大小(字节)
		页 511	0x0807 FC00 - 0x0807 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进行擦除、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 2-2 信息块

模块	名称	地址	大小(字节)
	保密空间	0x1FFE 1000 - 0x1FFF 1FFF	4K
	系统存储 ISP	0x1FFF E800 - 0x1FFF F7FF	4K
	选项字节	0x1FFF F800 - 0x1FFF F9FF	0.5K

2.2.2 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 2-3 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
0x1FFF F810	nWRP4	WRP4	0xFFFF
0x1FFF F812	nWRP5	WRP5	0xFFFF
0x1FFF F814	nWRP6	WRP6	0xFFFF
0x1FFF F816	nWRP7	WRP7	0xFFFF
0x1FFF F818	nWRP8	WRP8	0xFFFF
0x1FFF F81A	nWRP9	WRP9	0xFFFF

地址	[15: 8]	[7: 0]	默认值
0x1FF F81C	nWRP10	WRP10	0xFFFF
0x1FF F81E	nWRP11	WRP11	0xFFFF
0x1FF F820	nWRP12	WRP12	0xFFFF
0x1FF F822	nWRP13	WRP13	0xFFFF
0x1FF F824	nWRP14	WRP14	0xFFFF
0x1FF F826	nWRP15	WRP15	0xFFFF

表 2-4 USER 的位含义

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

	Bit	Field	Type	Default	Description	FLASH_OBR
	1	DATA1.Bit1	rw	0x01	用户自定义	FLASH_OBR. Bit19
	2	DATA1.Bit2	rw	0x01	用户自定义	FLASH_OBR. Bit20
	3	DATA1.Bit3	rw	0x01	用户自定义	FLASH_OBR. Bit21
	4	DATA1.Bit4	rw	0x01	用户自定义	FLASH_OBR. Bit22
	5	DATA1.Bit5	rw	0x01	用户自定义	FLASH_OBR. Bit23
	6	DATA1.Bit6	rw	0x01	用户自定义	FLASH_OBR. Bit24
	7	DATA1.Bit7	rw	0x01	用户自定义	FLASH_OBR. Bit25

注意：在写保护值中，一个比特位对应四页，即 4096 Bytes。

2.2.3 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。

2.3 闪存操作与流程

2.3.1 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），使能预取指缓冲区后，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 2-5 Latency 设置关系

SYSCLK	AHB DIV	Latency
0MHz < SYSCLK <= 24MHz	1	0
24MHz < SYSCLK <= 48MHz	1	1
48MHz < SYSCLK <= 72MHz	1	2
72MHz < SYSCLK <= 96MHz	1	3
96MHz < SYSCLK <= 120MHz	1	4

2.3.2 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 2-6 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

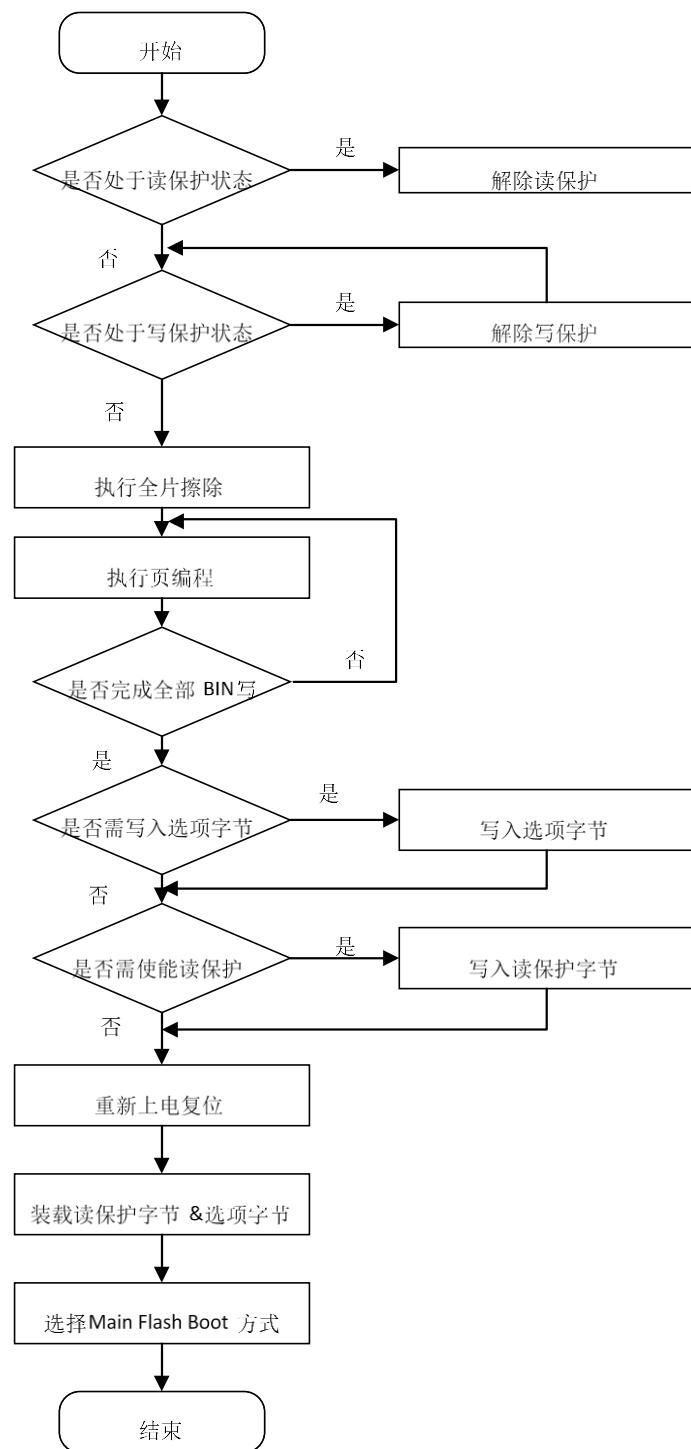


图 2-1 ISP 方式编程流程图

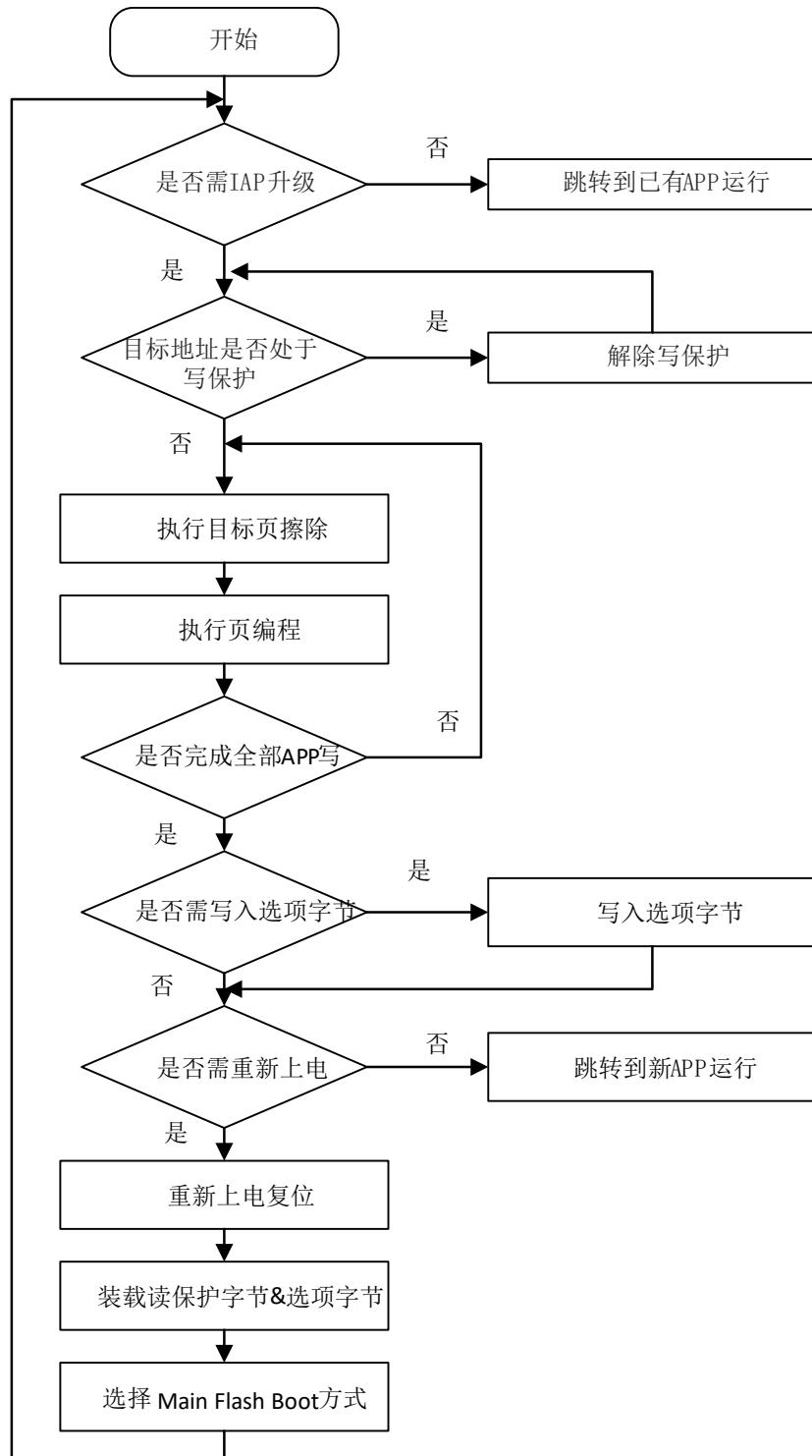


图 2-2 IAP 编程流程

2.3.3 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;
}
```

2.3.4 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与读保护设置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，才做选项字节区与读保护设置区的解锁。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，才能对选项字节区与读保护设置区执行块擦除，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与读保护设置区执行块擦除，半字编程操作。

表 2-7 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			使能对主闪存块的操作保护
闪存控制器状态为 FLASH_CR.LOCK=1	保护	保护	使能对选项字节区与读保护设置区的操作保护

设置与状态	主闪存块	信息块	说明
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			
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	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与读保护设置区的操作保护，不能对选项字节区与读保护设置区执行块擦除，半字编程操作
设置 FLASH_CR.OPTWRE=0 保持 FLASH_CR.LOCK=0	解除保护	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 解除对选项字节区与读保护设置区的操作保护，可对选项字节区与读保护设置区执行块擦除，半字编程操作
设置 FLASH_CR.OPTWRE=0 设置 FLASH_CR.LOCK=1	使能保护	使能保护	仍处于解除对主闪存块的操作保护，使能对选项字节区与读保护设置区的操作保护
			使能了对主闪存块的操作保护，使能对选项字节区与读保护设置区的操作保护

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;
}
```

解除选项字节区与读保护设置区保护操作代码:

```
#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;
}
```

对选项字节区与读保护设置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;
}
```

2.3.5 主闪存块擦除

闪存控制器支持整片擦除主闪存块和以页为单位擦除主闪存中的页。

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

整片擦除操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
<p>(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护 (2) 否则执行 Flash Unlock (3) 设置 FLASH_CR 寄存器中的 MER 位为 1，使能对主闪存全片擦除功能 (4) 设置 FLASH_CR 寄存器中的 STRT 位为 1 (5) 开始对主闪存全擦除 (6) 检查 FLASH_SR 中的 BSY 位 (7) 等待对主闪存全擦除的操作完成 (8) 清除 FLASH_CR 寄存器中的 MER 位，使该位为 0 (9) 可选做 Blank 检查，以确保擦除成功</p>	<pre> graph TD A[读取FLASH_CR中的LOCK位] --> B{FLASH_CR的LOCK=1} B -- 是 --> C[执行解锁序列] B -- 否 --> D[FLASH_CR中的MER位写1] D --> E[FLASH_CR中的STRT位写1] E --> F{FLASH_SR的BSY=1} F -- 是 --> E F -- 否 --> G[FLASH_CR中的MER位写0] G --> H[读取用户存储器中的所有地址来检查擦除操作] </pre>

页擦除操作的寄存器设置，具体步骤如下：

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

2.3.6 主闪存块编程

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

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

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

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

2.3.7 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_SR 中的 BSY 位</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 -- 否 --> C{FLASH_CR的OPTWRE=0} B -- 是 --> D[执行解锁序列] C -- 否 --> E[将闪存选项字节块地址写入FLASH_AR] C -- 是 --> F[执行解锁OPT序列] E --> 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 -- 否 --> J[FLASH_CR中的OPTER位写0] I -- 是 --> I 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>

2.3.8 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与读保护设置区解除访问限制。完成该操作后，FLASH_CR 寄存器中的 OPTWRE 位会被置 1，才能允许后续的编程操作。

选项字节有效数据为低 8 位，而高 8 位为低 8 位的反码，从而组成为 16 位数据。在编程过程中，软件将高 8 位设置为低 8 位的反码，保证选项字节的写入值总是对的，然后依次写入 16 位数据。

当选项字节被改变时，需要系统上电复位使之生效。

选项字节区块半字编程操作的寄存器设置，具体步骤如下：

执行步骤	简易流程图
<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) 设置 FLASH_CR 寄存器中的 OPTPG 位为 1，使能选项字节区块半字编程</p> <p>(4) 写数据（半字）到目标地址，目标地址需以半字对齐</p> <p>(5) 检查 FLASH_SR 中的 BSY 位</p> <p>(6) 等待半字编程操作完成</p> <p>(7) 清除 FLASH_CR 寄存器中的 OPTPG 位，使该位值为 0</p> <p>(8) 可选读目标地址数据，以确保半字编程成功</p> <p>(10) 连续多个半字编程，可以重复 3~7 操作；直到完成所有目标地址的半字编程</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] B -- 否 --> D{FLASH_CR 的OPTWRE=0} D -- 是 --> E[执行解锁OPT序列] D -- 否 --> F[FLASH_CR的 OPTPG位写1] F --> G[在所需地址执行半字写入] G --> H{FLASH_SR 的BSY=1} H -- 是 --> G H -- 否 --> I[FLASH_CR的 OPTPG位写0] I --> J[通过读取目标地址数据 来检查编程值] </pre>

2.3.9 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）。

2.3.9.1 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 0x1FFF800，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 2-8 Flash 读保护状态

使能读保护操作	读保护状态
对 0xFFFF800 选项区块擦除 写 0x807F 半字到对应地址 0xFFFF800 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 0xFFFF800，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 2-9 Flash 解除读保护状态

解除读保护操作	读保护状态
对 0xFFFF800 选项区块擦除 写 0x5AA5 半字到对应地址 0xFFFF800 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。

2.3.9.2 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~WRP15 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

WRPRTERR 标志位置位。

表 2-10 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	
0x1FFF F810	nWRP4	WRP4	0xFFFF	
0x1FFF F812	nWRP5	WRP5	0xFFFF	
0x1FFF F814	nWRP6	WRP6	0xFFFF	
0x1FFF F816	nWRP7	WRP7	0xFFFF	
0x1FFF F818	nWRP8	WRP8	0xFFFF	
0x1FFF F81A	nWRP9	WRP9	0xFFFF	
0x1FFF F81C	nWRP10	WRP10	0xFFFF	
0x1FFF F81E	nWRP11	WRP11	0xFFFF	
0x1FFF F820	nWRP12	WRP12	0xFFFF	
0x1FFF F822	nWRP13	WRP13	0xFFFF	
0x1FFF F824	nWRP14	WRP14	0xFFFF	
0x1FFF F826	nWRP15	WRP15	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 0xFFFF800；
- 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。

2.4 寄存器

2.4.1 寄存器总览

表 2-11 FLASH 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	FLASH_ACR	闪存访问控制寄存器	0x000000038
0x04	FLASH_KEYR	FPEC 键寄存器	0x000000000
0x08	FLASH_OPTKEYR	闪存 OPTKEY 寄存器	0x000000000
0x0C	FLASH_SR	闪存状态寄存器	0x000000000
0x10	FLASH_CR	闪存控制寄存器	0x000000080
0x14	FLASH_AR	闪存地址寄存器	0x000000000
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

2.4.2 FLASH_ACR 闪存访问控制寄存器

偏移地址：0x00

复位值：0x0000 0038

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

Bit	Field	Description
31: 6	Reserved	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	Reserved	保留，必须保持复位值

Bit	Field	Description
2: 0	LATENCY	<p>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。</p> <p>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 $48\text{MHz} < \text{SYSCLK} \leq 72\text{MHz}$ 011: 三个等待状态, 当 $72\text{MHz} < \text{SYSCLK} \leq 96\text{MHz}$ 100: 四个等待状态, 当 $96\text{MHz} < \text{SYSCLK} \leq 120\text{MHz}$</p>

2.4.3 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	<p>FPEC 键 (Flash key) 这些位用于输入 FPEC 的解锁键。</p>

注: 所有这些位是只写的, 读出时返回 0。

2.4.4 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	<p>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。</p>

注: 所有这些位是只写的, 读出时返回 0。

2.4.5 FLASH_SR 闪存状态寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

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

Bit	Field	Description
31:6	Reserved	保留, 必须保持复位值
5	EOP	操作结束 (End of operation) 当闪存操作 (编程 擦除) 完成时, 硬件设置这位为“1”, 写入“1”可以清除这位状态。
4	WRPRTERR	写保护错误 (Write protection error) 试图对写保护的闪存地址编程时, 硬件设置这位为“1”, 写入“1”可以清除这位状态。
3	Reserved	保留, 必须保持复位值
2	PGERR	编程错误 (Programming error) 试图对内容不是“0xFFFF”的地址编程时, 硬件设置这位为“1”, 写入“1”可以清除这位状态。 注: 进行编程操作之前, 必须先清除 FLASH_CR 寄存器的 STRT 位。
1	Reserved	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”。

2.4.6 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.						OPTWR E	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
						rw		rw	rw	rw	rw		rw	rw	rw

Bit	Field	Description
31:10	Reserved	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	Reserved	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	Reserved	保留, 必须保持复位值
2	MER	全擦除 (Mass erase) 选择擦除所有用户页。
1	PER	页擦除 (Page erase) 选择擦除页。
0	PG	编程 (Programming) 选择编程操作。

2.4.7 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。

2.4.8 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									
						r									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						Res.						nRST_S TDBY	nRST_S TOP	WDG_S W	Res. OPTER R
r						r						r	r	r	r

Bit	Field	Description
31: 26	Reserved	保留, 必须保持复位值
25: 18	Data1	Data1
17: 10	Data0	Data0
9: 5	Reserved	保留, 必须保持复位值
4	nRST_STDBY	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。 注意: 该位为只读。

2.4.9 FLASH_WRP Rx 写保护寄存器

偏移地址: 0x20, 0x24, 0x28, 0x2C

复位值: 0xFFFF 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	<p>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</p>

3 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。

3.1 供电系统

芯片由以下四种电源提供供电：

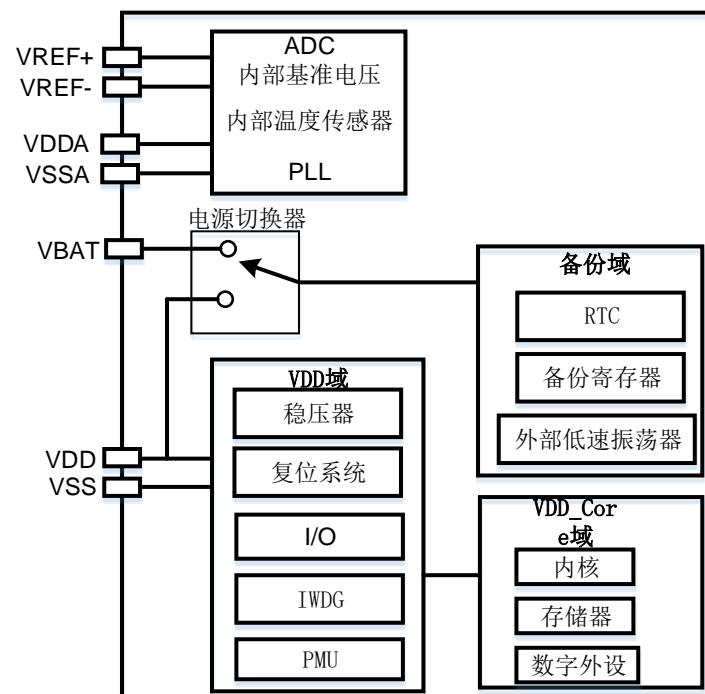


图 3-1 电源控制功能框图

- 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。
- VREF+和 VREF-提供 ADC 的参考电压。
- 由 VDD 和 VSS 提供的数字电源，用于数字部分和 I/O 引脚工作。
- 由 VBAT 提供的电池备份域电源，用于 RTC 和备份模块工作。

注：

- 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，并尽量靠近芯片电源引脚摆放。
- 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。

3.1.1 模拟模块供电

VDDA 主要给芯片内部模拟电路提供电源，主要包括 **ADC** 模块、内部基准电压、内部温度传感器和 **PLL** 等，因此电源的稳定性影响模拟模块的工作性能。

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

- **ADC** 的电源引脚为 **VREF+**
- **ADC** 独立的电源地 **VREF-**

3.1.2 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。

3.1.3 电池备份模块供电

VBAT 是芯片电池供电电源端口，主要为 **RTC**、备份寄存器和外部低速时钟振荡器 **LSE** 供电。**RTC** 和备份寄存器也可通过选通信号选择由 **VDD** 供电。

3.1.4 VDD 域

VDD 域主要给 **PMU**、**IWDG** 供电，在上电后保持工作状态。

3.1.5 备份域

备份域主要为 **RTC**、备份寄存器和外部低速时钟振荡器 **LSE** 供电。

3.1.6 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：**1.5V** 域以正常的功耗模式运行，内存、外设都正常工作。

低功耗运行模式：**1.5V** 域以低功耗模式运行，内存、外设都以低功耗工作。

睡眠模式：**1.5V** 域以正常功耗模式工作，**CPU** 进入睡眠模式，内存、外设都以正常的功耗模式工作。

低功耗睡眠模式：**1.5V** 域以低功耗睡眠模式工作，**CPU** 进入低功耗睡眠模式，内存、外设都以低功耗工作。

停机模式：1.5V 域以低功耗模式工作，只保持寄存器和 RAM 的内容。

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

待机模式：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。

3.2 电源管理器

3.2.1 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。

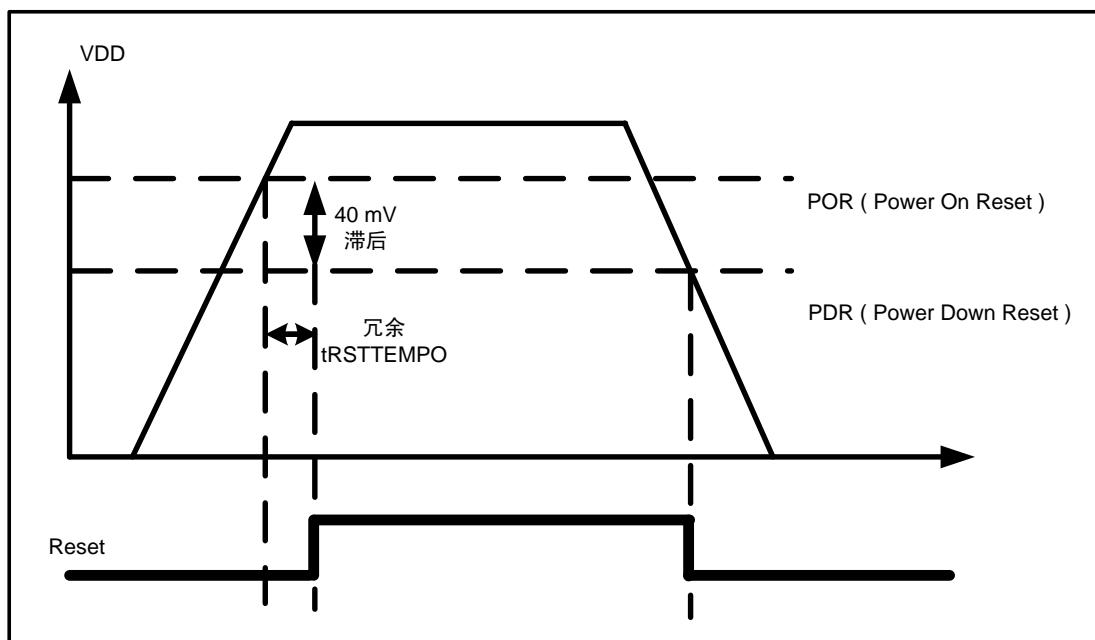


图 3-2 上电复位和掉电复位波形图

3.2.2 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系统电源检测配置寄存器 `SYSCFG_PDETCSR`. `PLS` 位的阈值电压与芯片供电电压进行比较，用来监控电源。通过设置系统电源检测配置寄存器 `SYSCFG_PDETCSR.PVDE` 位来使能 PVD。系统电源检测配置寄存器 `SYSCFG_PDETCSR`. `PVDO` 标志用来表明 `VDD` 是高于还是低于 `SYSCFG_PDETCSR.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。

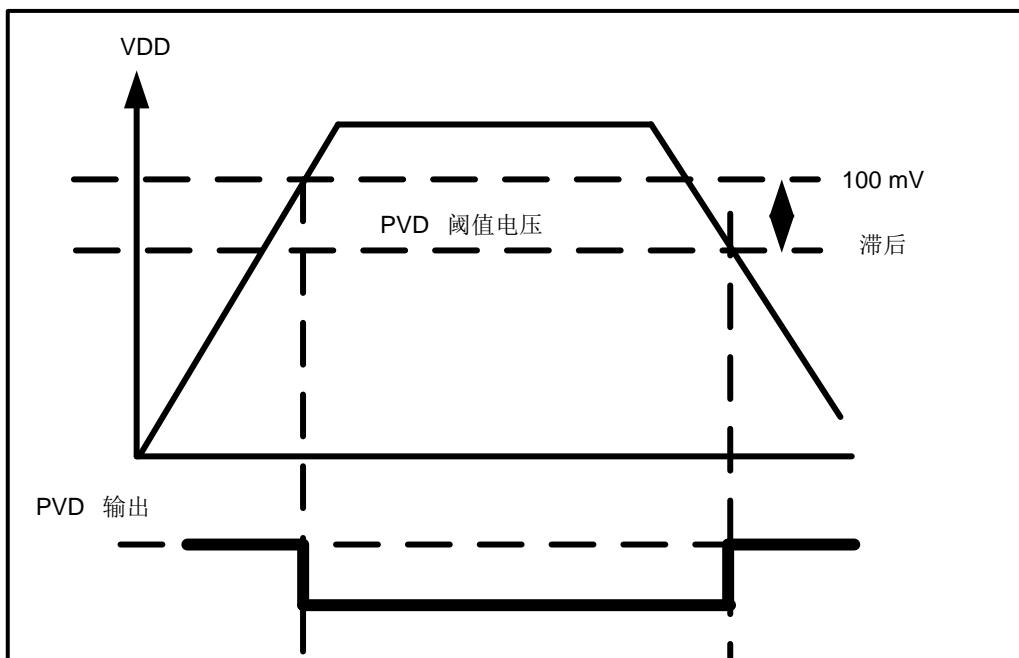


图 3-3 PVD 阈值波形图

3.3 功耗控制

3.3.1 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。

芯片有六种低功耗模式，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。

六种低功耗模式为：

低功耗运行模式：CPU、所有芯片外设包括 CPU 外设如 NVIC、SysTick 等在运行，系统工作时钟不能高于 2MHz。

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

低功耗睡眠模式：CPU 停止，所有芯片外设包括 CPU 的外设，如 NVIC、SysTick 等仍在运行，工作时钟不能高于 2MHz。

停机模式：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 RAM 的内容。

深度停机模式：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 RAM 的内容。

待机模式：1.5V 域停止供电，此域寄存器和 SRAM 的内容全部丢失，备份域除外。

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 3-1 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
低功耗运行模式 (Low Power Run Mode)	设置 LPR 位	清除 LPR 位	PLL,HSE 振荡器关闭 HSI、LSI、LSE 保持工作，系统工作时钟频率不超过 2MHz	无	开		
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					
低功耗睡眠模式 (Low Power)	设置 LPR 位； WFI (Wait for Interrupt)	任一中断	PLL,HSE 振荡器关闭 HSI、LSI、LSE 保持工作，系统	无	开		外设时钟继续维持，寄存器和

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
Sleep Mode)	设置 LPR 位; WFE (Wait for Event)	唤醒事件	工作时钟频率不超过 2MHz				SRAM 的内容保持
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、RTC 闹钟事件、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、RTC 闹钟事件、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、RTC 闹钟事件、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

3.3.2 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。

3.3.3 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。

外设主要挂在 AHB 和 APB 总线，可以通过设置 AHB 外设时钟使能寄存器（RCC_AHB1ENR、RCC_AHB2ENR、RCC_AHB3ENR）、APB 外设时钟使能寄存器（RCC_APB1ENR、RCC_APB2ENR）。

3.3.4 低功耗运行模式

3.3.4.1 低功耗运行模式进入

进一步降低 MCU 在运行模式下的功耗，可以通过配置 PWR_CR1 寄存器的 LPR 位将电压稳压器设置为低功耗模式。该模式下系统频率不应超过 2MHz。

3.3.4.2 低功耗运行模式退出

清除寄存器位 PWR_CR1.LPR 退出低功耗运行模式后可以提高系统时钟。

低功耗运行模式进入退出行为如下表：

表 3-2 低功耗运行模式

低功耗运行模式	说明
进入	在以下条件下置位寄存器位 PWR_CR1.LPR： 将系统频率设置低于 2MHz
退出	清除寄存器位 PWR_CR1.LPR 等待 PWR_CR1.LPR 位清除后，提高系统时钟频率
唤醒延时	稳压器从低功耗模式唤醒的时间

3.3.5 睡眠模式（Sleep Mode）

3.3.5.1 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。

3.3.5.2 睡眠模式退出

中断或事件发生后，睡眠模式立即被唤醒。

表 3-3 SLEEPNOW 模式

SLEEPNOW 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 3-4 SLEEPONEXIT 模式

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

3.3.6 Low Power Sleep Mode 低功耗睡眠模式

3.3.6.1 低功耗睡眠模式进入

通过执行 WFI (Wait For Interrupt) /WFE (Wait for Event) 指令, 请求 MCU 从低功耗运行模式进入低功耗睡眠模式。根据 CPU 系统控制寄存器中的 SLEEPONEXIT 位的值, 有两种可用于选择进入低功耗睡眠模式的机制:

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时, MCU 就立即进入睡眠模式。

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。

3.3.6.2 低功耗睡眠模式退出

中断或事件发生后, 睡眠模式立即被唤醒。

表 3-5 Low Power SLEEPNOW 模式

Low Power 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 LPR=1

Low Power SLEEP NOW 模式	说明
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 3-6 Low Power SLEEPONEXIT 模式

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

3.3.7 Stop Mode 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 和寄存器内容被保留下。

3.3.7.1 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 1 (PWR_CR1) 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_CR1) 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。
- 实时时钟 (RTC)：可以通过 RTC 对应的功能作为停机模式的唤醒源。
- 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 、LSIOE 位来设置。
- 外部低速振荡器 (LSE 振荡器)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 的 LSEON 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。

3.3.7.2 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。

若是从低功耗运行模式进入停机模式，需要在进入停机模式前配置寄存器 RCC_CFGR.HPRE 或者 RCC_CR.HSIDIV。这样可以保证停机模式唤醒后，系统频率不超过 2MHz。

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 3-7 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 1 (PWR_CR1) 中的 PDDS 位和 LPDS； 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 RTC 闹钟事情唤醒、看门狗中断；
唤醒延时	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

3.3.8 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。

3.3.8.1 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 1 (PWR_CR1) 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 1 (PWR_CR1) 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。
- 实时时钟 (RTC)：可以通过 RTC 对应的功能作为停机模式的唤醒源。
- 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 、LSIOE 位来设置。
- 外部低速振荡器 (LSE 振荡器)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 的 LSEON 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。

3.3.8.2 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。

若是从低功耗运行模式进入深度停机模式，需要在进入深度停机模式前配置寄存器 RCC_CFGR.HPRE 或者 RCC_CR.HSIDIV。这样可以保证深度停机模式唤醒后，系统频率不超过 2M。

当电压稳压器处于低功耗模式下，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 3-8 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 1 (PWR_CR1) 中的 PDSS，且置位 LPDS 位； 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 RTC 闹钟事情唤醒、看门狗中断；
唤醒延时	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

3.3.9 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 域被切断，PLL、HSI 和 HSE 振荡器也被关闭，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。

3.3.9.1 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 1 (PWR_CR1) 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。
- 实时时钟 (RTC)：可以通过 RTC 对应的功能作为待机模式的唤醒源。
- 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、LSIOE 位来设置。

3.3.9.2 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 3-9 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 1 (PWR_CR1) 中的 PDDS 位； 复位电源状态寄存器 (PWR_SR) 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、RTC 闹钟事件、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从而造成二次下载

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
- 设置为防侵入或校准输出的 Tamper 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

3.3.9.3 滤波功能

配置 PWR_CR4 和 PWR_CR5 寄存器的 FILTSEL0、FILTSEL1 选择需要过滤的唤醒引脚; FILTCNT0、FILTCNT1 选择唤醒源过滤宽度; 最后配置 FILTE0、FILTE1 使能位打开唤醒源滤波功能。滤波功能开启后可以对唤醒源进行滤波，将滤波阈值（可配置寄存器位 FILTCNT0 调整阈值）以下的干扰滤除，防止误操作唤醒。

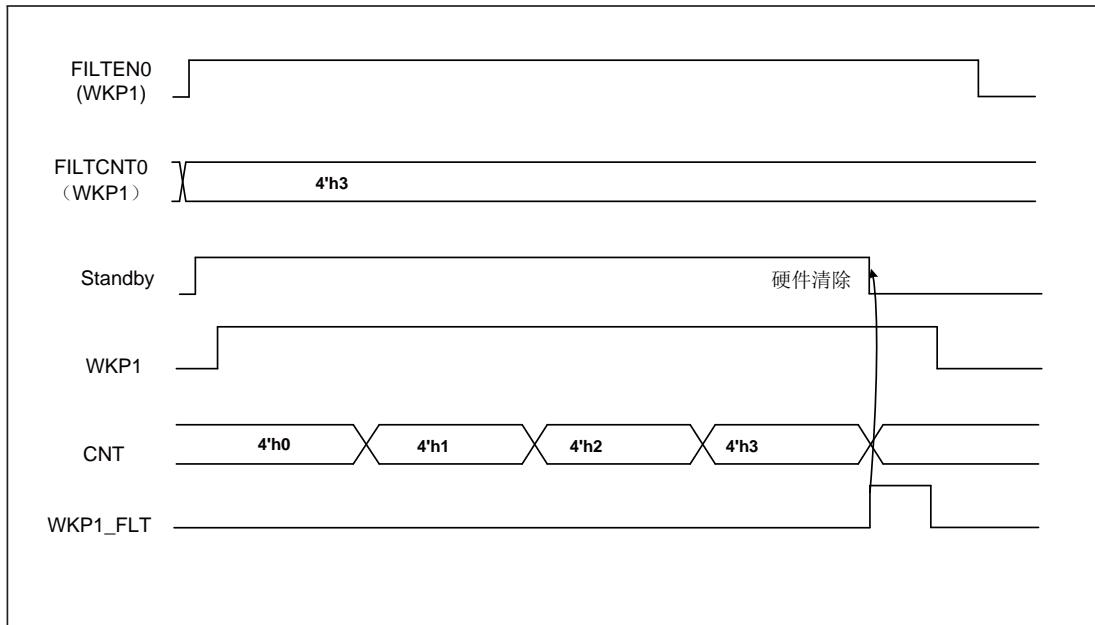


图 3-4 Standby 模式唤醒滤波功能波形图

3.3.9.4 快速唤醒功能

通过配置 PWR_CR6 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。

3.4 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。

3.5 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 3-10 电源控制寄存器概览

Offset	Acronym	RegisterName	Reset
0x00	PWR_CR1	电源控制寄存器 1	0x00000000
0x04	PWR_CSR	电源控制状态寄存器	0x00000000
0x08	PWR_CR2	电源控制寄存器 2	0x00000000
0x0C	PWR_CR3	电源控制寄存器 3	0x00000000

Offset	Acronym	RegisterName	Reset
0x10	PWR_CR4	电源控制寄存器 4	0x00000F00
0x14	PWR_CR5	电源控制寄存器 5	0x00000F00
0x18	PWR_CR6	电源控制寄存器 6	0x00000000
0x1C	PWR_SR	电源状态寄存器	0x00000000
0x20	PWR_SCR	电源状态清除寄存器	0x00000000
0x24	PWR_CFGR	电源配置寄存器	0x00000160

3.5.1 PWR_CR1 电源控制寄存器 1

地址偏移: 0x00

复位值: 0x00000000 (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
VOS	LPR	Res										CSBF	Res	PDDS	LPDS
rw	rw											rw		rw	rw

Bit	Field	Description
31:16	Reserved	保留, 始终读为 0
15:14	VOS	调制器电压输出选择 (Voltage Output Selection) 00: 保留 01: 1.6V 10: 1.7V 11: 保留 注: 如果系统时钟需要达到最大频率 120MHz, 需要将 VOS 设置为 2'b10 即 1.7V。
13	LPR	低功耗配置位, 置位后电压稳压器从正常模式切换到低功耗模式。 1: 低功耗运行模式 0: 正常运行模式
7:4	Reserved	保留, 始终读为 0
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	Reserved	保留, 始终读为 0
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

Bit	Field	Description
0	LPDS	<p>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。</p>

3.5.2 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	VOSRD Y	Res				Res						SBF	Res		
	r											r			

Bit	Field	Description
31:15	Reserved	保留, 始终读为 0
14	VOSRDY	<p>电压调制器输出准备完成标志位 1: 准备完成 0: 准备未完成</p>
13:2	Reserved	保留, 始终读为 0
1	SBF	<p>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 1 (PWR_CR1) 的 CSBF 位清除。 1: 系统进入待机模式 0: 系统不在待机模式</p>
0	Reserved	保留, 始终读为 0

3.5.3 PWR_CR2 电源控制寄存器 2

地址偏移: 0x08

复位值: 0x00000000

寄存器在退出待机模式时不进行复位，也不会在 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								EWUP5	EWUP4	EWUP3	EWUP2	EWUP1	EWUP0		
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5:0	EWUP5~ EWUP1	使能唤醒引脚 0~5 (Enable WakeUp Pin0~5) 1: WKUP 引脚用于唤醒待机模式, WKUP 引脚需要设置为输入下拉 (WKUP 引脚上的上升沿将系统从待机模式唤醒) 0 : WKUP 引脚上的事件不能唤醒待机模式, 该引脚为通用 I/O。

3.5.4 PWR_CR3 电源控制寄存器 3

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, 也不会在 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								WP5	WP4	WP3	WP2	WP1	WP0		
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5:0	WP5~WP0	唤醒引脚 0~5 的极性检测 1: 低电平检测 (下降沿), 唤醒引脚必须先配置为高电平 0: 高电平检测 (上升沿), 唤醒引脚必须先配置为低电平

3.5.5 PWR_CR4 电源控制寄存器 4

地址偏移: 0x10

复位值: 0x00000F00

寄存器在退出待机模式时不进行复位, 也不会在 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
FILTCNT0								Res		FILTF0	FILTE0		FILTSEL0		
										w1c					

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
15:8	FILTCNT0	唤醒引脚滤波器 0 计数器值: 0、1、2、3...、254、255
7:5	Reserved	保留, 始终读为 0
4	FILTFO	唤醒引脚滤波器 0 滤波有效标志位 1: 唤醒源经过滤波器 0 0: 唤醒源不经过滤波器 0
3:2	FILTE0	唤醒引脚滤波器 0 使能 (Filter Enable) 00: 使能滤波器 0 无效 01: 使能滤波器 0 上沿滤波 10: 使能滤波器 0 下沿滤波 其他: 保留
1:0	FILTSEL0	唤醒引脚滤波选择 0 (WakeUp Pin Filter Select 0) 00: 选择 WKUP 引脚 0 01: 选择 WKUP 引脚 1 10: 选择 WKUP 引脚 2 其他: 保留

3.5.6 PWR_CR5 电源控制寄存器 5

地址偏移: 0x14

复位值: 0x00000F00

寄存器在退出待机模式时不进行复位, 也不会在 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
FILTCNT1								Res		FILTF1	FILTE1		FILTSEL1		
rw										w1c	rw		rw		

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
15:8	FILTCNT1	唤醒滤波器 1 计数器值: 0、1、2、3...、254、255
7:5	Reserved	保留, 始终读为 0
4	FILTF1	唤醒引脚滤波器 1 滤波有效标志位 1: 唤醒源经过滤波器 1 0: 唤醒源不经过滤波器 1

Bit	Field	Description
3:2	FILTE1	唤醒引脚滤波器 1 使能 (Filter Enable) 00: 使能滤波器 1 无效 01: 使能滤波器 1 上沿滤波 10: 使能滤波器 1 下沿滤波 其他: 保留
1:0	FILTSEL1	唤醒引脚滤波选择 1 (WakeUp Pin Filt Select 1) 00: 选择 WKUP 引脚 3 01: 选择 WKUP 引脚 4 10: 选择 WKUP 引脚 5 其他: 保留

3.5.7 PWR_CR6 电源控制寄存器 6

地址偏移: 0x18

复位值: 0x00000000

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

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															
STDBY_FS_WK															
rw															

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
2:0	STDBY_FS_WK	STDBY_FS_WK: 快速唤醒 Standby 时间选择位 000: 保留, 禁止使用。 001: 2 个 LSI 周期唤醒 010: 3 个 LSI 周期唤醒 011: 4 个 LSI 周期唤醒 000: 5 个 LSI 周期唤醒 101: 6 个 LSI 周期唤醒 110: 7 个 LSI 周期唤醒 111: 8 个 LSI 周期唤醒 使用待机模式前需要配置该寄存器为非“0”值

3.5.8 PWR_SR 电源状态寄存器

地址偏移: 0x1C

复位值: 0x00000000

寄存器在退出待机模式时不进行复位，也不会在 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															
										WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
										r	r	r	r	r	r

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5:0	WUF5~WUF0	唤醒引脚 0~5 的唤醒标志位 (WakeUp Source0~5 Flag) 1: 检测到唤醒引脚有唤醒事件 0: 未检测到唤醒引脚有唤醒事件

3.5.9 PWR_SCR 电源状态清除寄存器

地址偏移: 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															
										CWUF5	CWUF4	CWUF3	CWUF2	CWUF1	CWUF0
										w	w	w	w	w	w

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5:0	CWUF5~ CWUF0	清除唤醒引脚 0~5 的唤醒标志位 (Clear WakeUp Source0~5 Flag) 1: 清除 PWR_SR 寄存器对应的唤醒标志位 0: 无

3.5.10 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															

Res	LSICAL	LSICALSEL
	rw	w

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
9:5	LSICAL	<p>内部低速时钟校准位 (Internal Low-speed Clock Calibration Bits)</p> <p>在系统启动时, 这些位被自动初始化为出厂校准值, 用户可以写入其他校准值, 但读出始终为出厂校准值。如果 LSICALSEL=0x1F, 写入的值可以重新校正 LSI 频率, 否则写入的值不起作用</p>
4:0	LSICALSEL	<p>内部低速时钟校准值选择方式 (Internal Low-speed Calibration Select)</p> <p>初值为 0, 当写入不同值时:</p> <p>写入 1F: 选择寄存器 LSICAL 的值, 写入 1F 后仍读出为 0。</p> <p>写入其他: 选择出厂校准值</p>

4 RCC 时钟和复位

4.1 复位单元

4.1.1 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。

4.1.2 功能框图

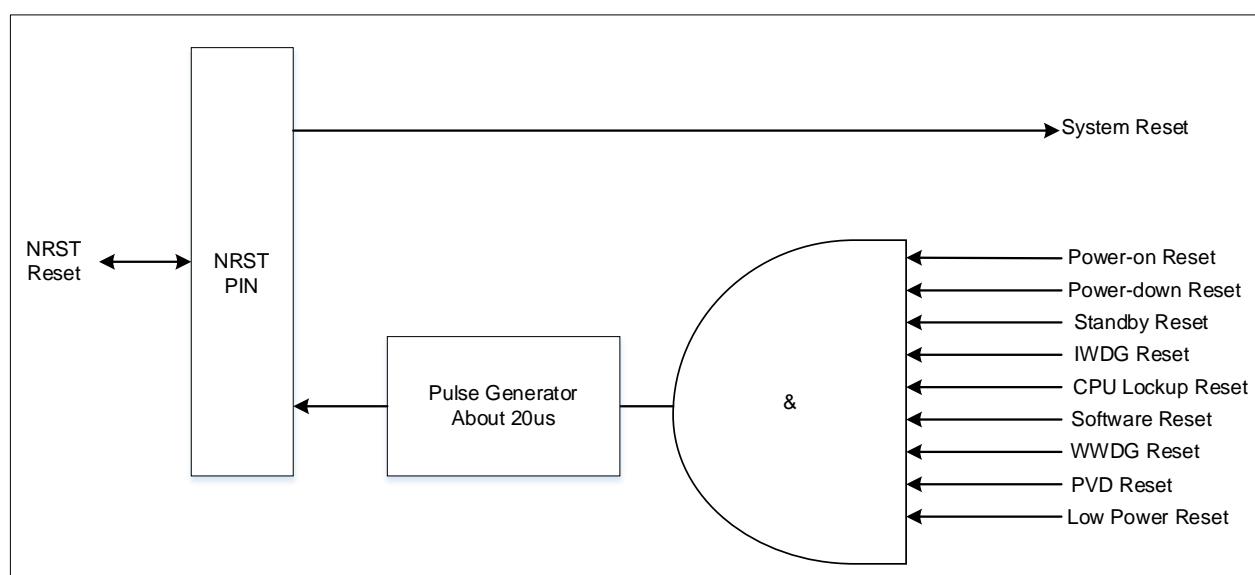


图 4-1 复位功能框图

4.1.3 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，包括备份域寄存器。
- 系统复位：除了 RCC_CSR 中的复位标志、内部低速振荡器使能位及 LSI 时钟输出使能位、PWR_CSR 中的待机和唤醒标志、DBG_CR、RTC 部分寄存器（RTC_PRLH、RTC_PRLL、RTC_DIVH、RTC_DIVL、RTC_CNTH、RTC_CNTL、RTC_ALRH、RTC_ALRL、RTC_MSRH、RTC_MSRL）、备份寄存器不受系统复位影响，其余寄存器都将被系统复位。
- 备份域复位：复位备份域寄存器（除了 RTC_CRH、RTC_CRL 和 RTC_LSE_CFG）

4.1.4 功能描述

4.1.4.1 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)

4.1.4.2 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。
- 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；
- 配置电源检测配置状态寄存器 (SYSCFG_PDETCSR) 的 PVDE 位为 1，使能 PVD；
- 配置电源检测配置状态寄存器 (SYSCFG_PDETCSR) 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

4.1.4.3 备份域复位 (Backup Domain Reset)

备份域复位有以下方式：

- 配置备份域控制寄存器 RCC_BDCR 中的 BDRST 位

4.2 时钟单元

4.2.1 简介

五个可配置的独立时钟：

- 外部高速时钟 (HSE)
- 内部高速时钟 (HSI)
- 锁相环 (PLL)
- 外部低速时钟 (LSE)

- 内部低速时钟 (LSI)

4.2.2 功能框图

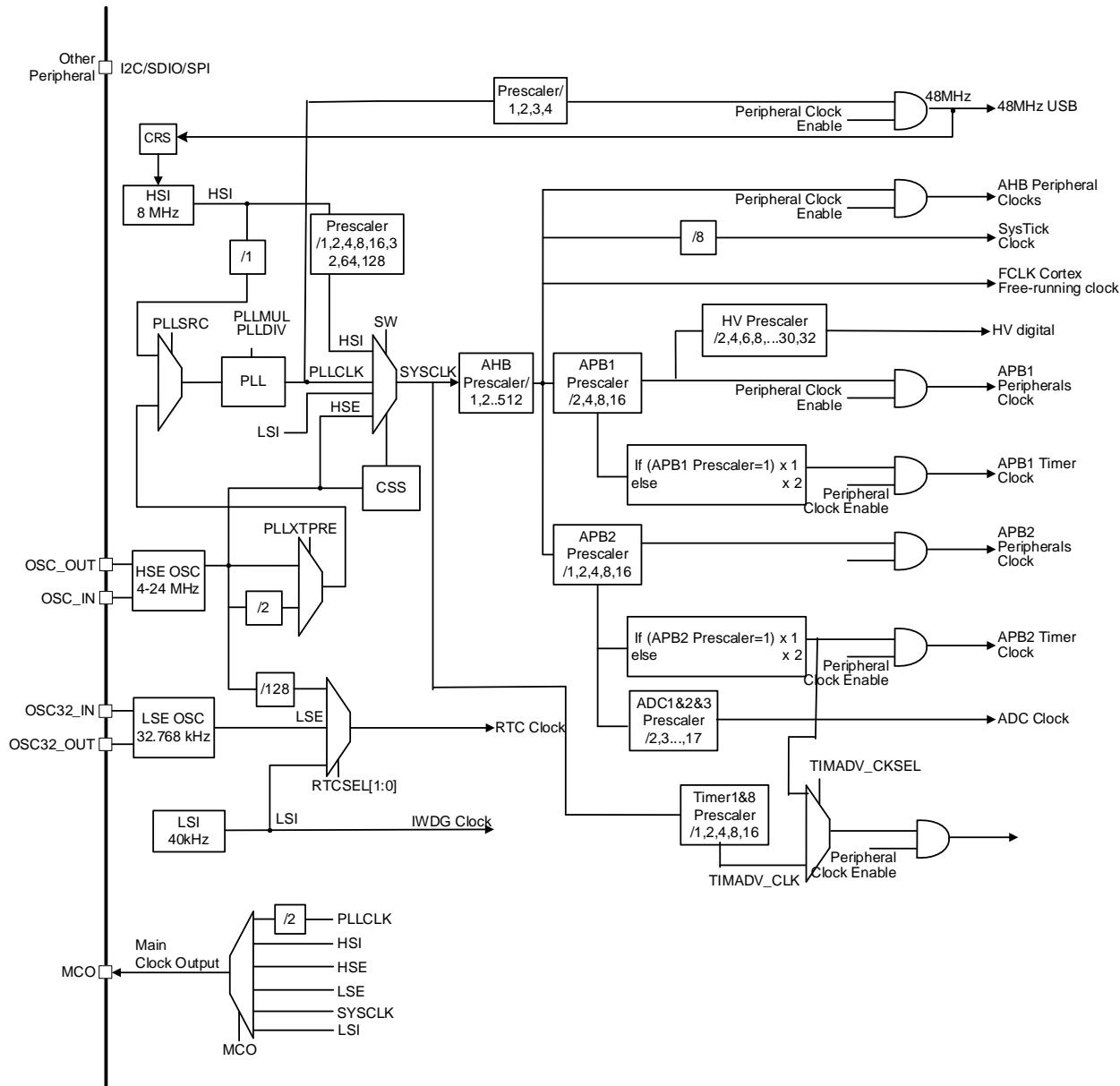


图 4-2 时钟树

4.2.3 主要特征

通过时钟配置寄存器 (RCC_CFR) 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 120MHz。

4.2.4 功能描述

4.2.4.1 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。

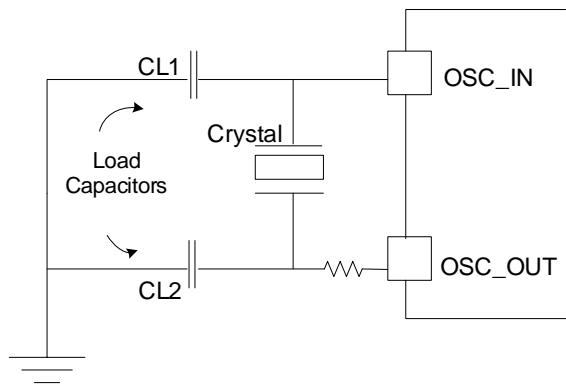


图 4-3 高速晶振/陶瓷谐振器

上图为外部高速晶振/陶瓷谐振器输入模块示意图，具体请参考数据手册。

- 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。

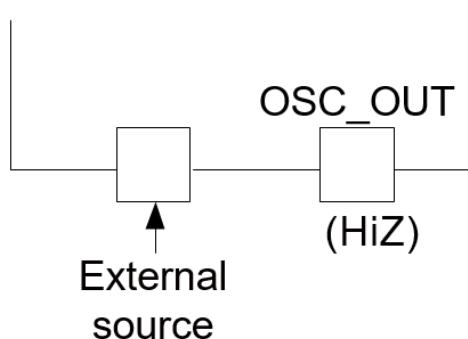


图 4-4 外部高速输入时钟

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

- 在使能 HSE 时，必须驱动 OSC_IN 管脚，而 OSC_OUT 引脚在未被使用时应该保持悬空

注意事项：

- 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。

4.2.4.2 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。

4.2.4.3 锁相环（PLL）

PLL 的输入参考时钟有以下三种，可以通过 PLL 配置寄存器（RCC_PLLCFGR）中的 PLLXTPRE 和

PLLSRC 位进行选择：

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

PLL 的输入时钟 f_{PLL_IN} 和输出时钟 f_{PLL_OUT} 之间关系为：

$$\frac{f_{PLL_IN}}{PLLDIV[2 : 0] + 1} = \frac{f_{PLL_OUT}}{PLLMUL[6 : 0] + 1}$$

PLLMUL 和 PLLDIV 是 PLL 的倍频器和分频器的分频比设置控制位。

注意事项：

- 在使能 PLL 之前，必须进行 PLL 配置（选择 PLL 输入时钟源和选择倍频因子和分频因子）。一旦 PLL 启用，PLL 配置就不能被更改。如需更改配置，必须先禁止 PLL。
- 如果在应用中需要使用 USB 接口，则需要 48MHz 的 USB 时钟，因此 PLL 必须被配置为 48MHz 或 96MHz。
- PLL 配置寄存器（RCC_PLLCFGR）中 PLLDIV 建议设置为奇数（1 或 3 或 5 或 7）；

使能 PLL 配置步骤：（选择 HSI 为 PLL 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 PLL 配置寄存器（RCC_PLLCFGR）中 PLLMUL（倍频系数）和 PLLDIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLLON 位为 1，使能 PLL；
- 等待时钟控制寄存器（RCC_CR）中的 PLLRDY 位被置位为 1，表示 PLL 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 PLL 配置步骤：（选择 HSE 为 PLL 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 PLL 配置寄存器（RCC_PLLCFGR）中的 PLLSRC 位为 1，选择 HSE 时钟用作 PLL 输入时钟源；
- 配置 PLL 配置寄存器（RCC_PLLCFGR）中的 PLLXTPRE 位，选择 HSE 或者 HSE 2 分频时钟为 PLL 输入时钟源；
- 配置 PLL 配置寄存器（RCC_PLLCFGR）中 PLLMUL（倍频系数）和 PLLDIV（分频系数）控

制位；

- 配置时钟控制寄存器（RCC_CR）中的 PLLON 位为 1，使能 PLL；
- 等待时钟控制寄存器（RCC_CR）中的 PLLRDY 位被置位为 1，表示 PLL 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。

4.2.4.4 外部低速时钟（LSE）

外部低速时钟的时钟源有以下两种：

- 外部低速晶振/陶瓷谐振器输入时钟（32.768kHz 晶体振荡器产生）

其为实时时钟单元（RTC）或者其他定时功能提供低功耗且高精度的时钟源。

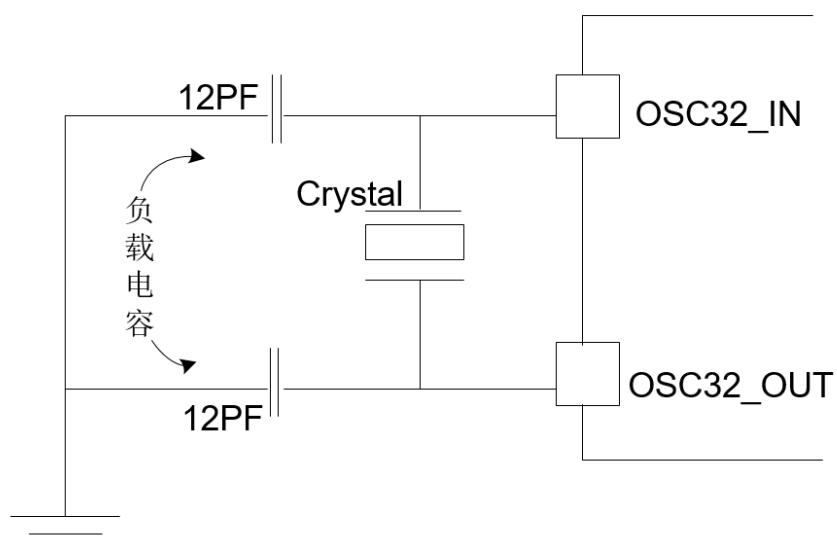


图 4-5 低速晶振/陶瓷谐振器

上图为外部低速晶振/陶瓷谐振器输入模块示意图，具体请参考数据手册。

- 外部低速输入时钟

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

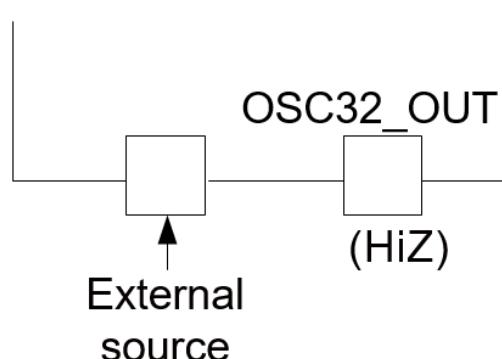


图 4-6 外部低速输入时钟

上图为外部低速输入时钟模块框图

外部低速时钟具有以下特征：

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

使能 LSE 配置步骤：

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

注意事项：

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

4.2.4.5 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及系统在停机或待机模式下的 RTC 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。

4.2.4.6 中断

表 4-1 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_PLLRDY	PLLRDYF	PLLRDYIE	PLLRDYC
RCC_HSERDY	HSERDYF	HSERDYIE	HSERDYC
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC
RCC_LSERDY	LSERDYF	LSERDYIE	LSERDYC
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。

4.2.4.7 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（PLL）：当 HSI（或 HSE）作为 PLL 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, PLL, LSI），每个时钟使能方式不同，具体方式请查看（HSI, HSE, PLL, 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。

4.2.4.8 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渡，切换间隔至少 1us。

4.2.4.9 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 AHB1 外设复位寄存器（RCC_AHB1RSTR）、AHB2 外设复位寄存器（RCC_AHB2RSTR）、AHB3 外设复位寄存器（RCC_AHB3RSTR）来实现相应外设的软件复位。

4.2.4.10 时钟安全系统（CSS）

时钟安全系统的具体工作方式包括软件启动和中断处理。

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

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

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

注意事项：

- 一旦启用 CSS，如果 HSE 时钟发生故障，CSS 中断发生，并自动生成 NMI 中断。除非 CSS 中断挂起位被清除，否则 NMI 将被不停地执行。因此，在 NMI 的处理程序中，用户必须通过设置时钟中断寄存器（RCC_CIR）的 CSSC 位为 1 来清除 CSS 中断。
- 当故障发生时，如果直接使用 HSE 振荡器作为系统时钟，系统时钟将自动切换到内部高速振荡器 HSI，并禁止 HSE 振荡器；如果 HSE 振荡器用作 PLL 输入时钟，PLL 时钟用作系统时钟，则 PLL 也被同时禁止。
- 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_PLLCFGR）中的 PLLSRC 位为 1，选择 HSE 时钟用作 PLL 输入时钟；
- 配置 PLL 配置寄存器（RCC_PLLCFGR）中的 PLLXTPRE 位，选择 HSE 或者 HSE2 分频为 PLL 输入时钟；
- 配置 PLL 配置寄存器（RCC_PLLCFGR）中 PLLMUL（倍频系数）、PLLDIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLLON 位为 1，使能 PLL；
- 等待时钟控制寄存器（RCC_CR）中的 PLLRDY 位被置位为 1，表示 PLL 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 PLL 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 PLL；
- 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。

4.2.4.11 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。可以选择以下六个时钟信号中的一个作为 MCO 输出时钟：

表 4-2 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	LSE
100	SYSCLK
101	HSI
110	HSE
111	PLL/2

4.2.4.12 RTC 时钟

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

消备份域的写保护) 为 1 才可修改此选择。

LSE 时钟在备份域里, 而 **HSE** 和 **LSI** 时钟不在备份域。因此:

- 如果 **LSE** 被选为 **RTC** 时钟: 只要 **VBAT** 维持供电, 即使 **VDD** 供电被切断, **RTC** 仍继续工作;
- 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。

4.2.4.13 定时器时钟

当时钟配置寄存器 (**RCC_CFGR**) 的 **PPRE1** 位[10:8]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 2 (**RCC_CFGR2**) 的 **TIMADV_CKSEL** 位为 1, 高级定时器 (**TIM1/8**) 的时钟频率直接由系统时钟经过配置时钟配置寄存器 2 (**RCC_CFGR2**) 的 **TIMADV_CLK** 预分频位得到。

4.2.4.14 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。

4.3 寄存器

4.3.1 寄存器总览

表 4-3 RCC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	时钟控制寄存器	0x00000001
0x04	RCC_CFGR	时钟配置寄存器	0x00000000
0x08	RCC_CIR	时钟中断寄存器	0x00000000
0x0C	RCC_AHB3RSTR	AHB3 外设复位寄存器	0x00000000
0x10	RCC_AHB2RSTR	AHB2 外设复位寄存器	0x00000000
0x14	RCC_AHB1RSTR	AHB1 外设复位寄存器	0x00000000
0x18	RCC_APB2RSTR	APB2 外设复位寄存器	0x00000000
0x1C	RCC_APB1RSTR	APB1 外设复位寄存器	0x00000000
0x20	RCC_AHB3ENR	AHB3 外设时钟使能寄存器	0x00000000
0x24	RCC_AHB2ENR	AHB2 外设时钟使能寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x28	RCC_AHB1ENR	AHB1 外设时钟使能寄存器	0x00006000
0x2C	RCC_APB2ENR	APB2 外设时钟使能寄存器	0x00000000
0x30	RCC_APB1ENR	APB1 外设时钟使能寄存器	0x00000000
0x34	RCC_BDCR	备份域控制寄存器	0x00000000
0x38	RCC_CSR	控制状态寄存器	0x0C000000
0x3C	RCC_SYSCFG	系统配置寄存器	0x00000101
0x40	RCC_CFGR2	时钟配置寄存器 2	0x00031F00
0x48	RCC_PLLCFGR	PLL 配置寄存器	0x0018031C

4.3.2 RCC_CR 时钟控制寄存器

偏移地址: 0x00

复位值: 0x0000 0001

访问: 无等待状态, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						PLLRDY	PLلون	Res.				CSSON	HSEBY	HSERD	HSEON
						r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HSIDIV				Res.										HSIRDY
	rw														HSION

Bit	Field	Description
31: 26	Reserved	保留, 必须保持复位值
25	PLLRDY	PLL 时钟稳定标志 (PLL Clock Ready Flag) 由硬件设置 0: PLL 未稳定 1: PLL 已稳定
24	PLلون	PLL 使能 (PLL 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 PLL 时钟已经或将要作为系统时钟, 则禁止重置此位。 0: 禁止 PLL 1: 使能 PLL
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

Bit	Field	Description
18	HSEBYP	<p>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</p>
17	HSERDY	<p>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</p>
16	HSEON	<p>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</p>
15: 14	Reserved	保留，必须保持复位值
13: 11	HSIDIV	<p>HSI 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 注：停机或深度停机模式被唤醒后，系统将强制使用 HSI 作为系统时钟。在低功耗运行或低功耗睡眠模式下，需要配置内部高速时钟分频系数以降低系统时钟频率。</p>
10: 2	Reserved	保留，必须保持复位值
1	HSIRDY	<p>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</p>
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”。 当退出待机或停机模式或外部振荡器用作系统时钟并且发生故障时，此位由硬件置“1”，来迫使内部振荡器使能。当系统时钟已经或将要使用 HSI 作为时钟源时，则禁止重置此位。 0: 禁止内部高速时钟 1: 使能内部高速时钟</p>

4.3.3 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.					
					rw			rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		PPRE2			PPRE1			HPRE				SWS		SW	
		rw			rw			rw				r		rw	

Bit	Field	Description
31: 27	Reserved	保留, 必须保持复位值
26: 24	MCO	微控制器时钟输出 (Micro Controller Clock Output) 由软件置“1”或清“0” 00x: 没有时钟输出 010: LSI 时钟输出 011: LSE 时钟输出 100: SYSCLK 系统时钟输出 101: HSI 时钟输出 110: HSE 时钟输出 111: PLL 时钟 2 分频后输出 注意: 该时钟输出在启动和切换 MCO 时钟源时可能会被停止。 系统时钟通过 MCO 管脚输出时, 请保证输出时钟频率不超过 50MHz。
23: 22	USBPRE	USB 预分频 (USB Prescaler) 通过软件设置分频值以输出 48MHz 的 USB 时钟。 在配置 RCC_AHB2ENR 寄存器使能 USB 时钟之前, 此位必须有效。 00: PLL 时钟直接作为 USB 时钟 01: PLL 时钟 2 分频作为 USB 时钟 10: PLL 时钟 3 分频作为 USB 时钟 11: PLL 时钟 4 分频作为 USB 时钟
21: 15	Reserved	保留, 必须保持复位值
14	Reserved	保留, 必须保持复位值。

Bit	Field	Description
13: 11	PPRE2	<p>APB2 预分频系数 通过软件设置来控制 APB2 时钟（PCLK2）预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频</p>
10: 8	PPRE1	<p>PPRE1: APB1 预分频系数 通过软件设置来控制 APB1 时钟（PCLK1）预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频</p>
7: 4	HPRE	<p>AHB 预分频系数 通过软件设置来控制 AHB 时钟的预分频系数。 0xxx: SYSCLK 不分频 1000: SYSCLK 2 分频 1001: SYSCLK 4 分频 1010: SYSCLK 8 分频 1011: SYSCLK 16 分频 1100: SYSCLK 64 分频 1101: SYSCLK 128 分频 1110: SYSCLK 256 分频 1111: SYSCLK 512 分频 注： 当 AHB 时钟的预分频系数大于 1 时，必须开启预取缓冲器，详见闪存访问控制寄存器章节。</p>
3: 2	SWS	<p>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 PLL 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟</p>
1: 0	SW	<p>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 PLL 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟</p>

4.3.4 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.		PLLRDY	HSERD	HSIRDY	LSERD	LSIRDY
								C	Res.		YC	C	YC	C	
								w1c	w1c		w1c	w1c	w1c	w1c	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			PLLRDY	HSERD	HSIRDY	LSERD	LSIRDY	CSSF	Res.		PLLRDY	HSERD	HSIRDY	LSERD	LSIRDY
			IE	YIE	IE	YIE	E		F		YF	F	YF	F	
			RW	RW	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	PLLRDYC	清除 PLL 稳定中断 (PLL Ready Interrupt Clear) 通过软件置“1”来清除 PLL 稳定中断标志位 PLLRDYF。 0: 无效 1: 清除 PLL 稳定中断标志位 PLLRDYF
19	HSERDYC	清除 HSE 稳定中断 (HSE Ready Interrupt Clear) 通过软件置“1”来清除 HSE 稳定中断标志位 HSERDYF。 0: 无效 1: 清除 HSE 稳定中断标志位 HSERDYF
18	HSIRDYC	清除 HSI 稳定中断 (HSI Ready Interrupt Clear) 通过软件置“1”来清除 HSI 稳定中断标志位 HSIRDYF。 0: 无效 1: 清除 HSI 稳定中断标志位 HSIRDYF
17	LSERDYC	清除 LSE 稳定中断 (LSE Ready Interrupt Clear) 通过软件置“1”来清除 LSE 稳定中断标志位 LSERDYF。 0: 无效 1: 清除 LSE 稳定中断标志位 LSERDYF
16	LSIRDYC	清除 LSI 稳定中断 (LSI Ready Interrupt Clear) 通过软件置“1”来清除 LSI 稳定中断标志位 LSIRDYF。 0: 无效 1: 清除 LSI 稳定中断标志位 LSIRDYF

Bit	Field	Description
15: 13	Reserved	保留, 必须保持复位值
12	PLL RDY IE	<p>PLL 稳定中断使能 (PLL Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止 PLL 稳定中断。 0: 禁止 PLL 稳定中断 1: 使能 PLL 稳定中断</p>
11	HSE RDY IE	<p>HSE 稳定中断使能 (HSE Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止外部振荡器稳定中断。 0: 禁止 HSE 稳定中断 1: 使能 HSE 稳定中断</p>
10	HSI RDY IE	<p>HSI 稳定中断使能 (HSI Ready Interrupt Enable) 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</p>
9	LSE RDY IE	<p>LSE 稳定中断使能 (LSE Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止外部 32KHz 振荡器稳定中断。 0: 禁止 LSE 稳定中断 1: 使能 LSE 稳定中断</p>
8	LSI RDY IE	<p>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</p>
7	CSSF	<p>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</p>
6: 5	Reserved	保留, 必须保持复位值
4	PLL RDY F	<p>PLL 稳定中断标志 (PLL Ready Interrupt Flag) 在 PLL 稳定时, 由硬件置“1”。 通过软件将 PLLRDYC 位置“1”来清除。 0: 无 PLL 锁定产生的时钟稳定中断 1: PLL 锁定导致时钟稳定中断</p>
3	HSE RDY F	<p>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</p>
2	HSI RDY F	<p>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</p>

Bit	Field	Description
1	LSERDYF	<p>LSE 稳定中断标志 (LSE Ready Interrupt Flag) 在外部低速时钟稳定时, 由硬件置“1”。 通过软件将 LSERDYC 位置“1”来清除。 0: 无外部 32KHz 振荡器产生的时钟稳定中断 1: 外部 32KHz 振荡器导致时钟稳定中断</p>
0	LSIRDYF	<p>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</p>

4.3.5 RCC_AHB3RSTR AHB3 外设复位寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

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

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值
0	FSMC	<p>FSMC 复位 (FSMC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位</p>

4.3.6 RCC_AHB2RSTR AHB2 外设复位寄存器

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.	USBOT GFS rw	Res.
------	--------------------	------

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	USBOTGFS	USBOTGFS 复位 (USBOTGFS Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
6: 0	Reserved	保留, 必须保持复位值

4.3.7 RCC_AHB1RSTR AHB1 外设复位寄存器

偏移地址: 0x14

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.										DMA2	DMA1	Res.			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CRC	Res.	SDIO	Res.		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
rw			rw			rw		rw							

Bit	Field	Description
31: 23	Reserved	保留, 必须保持复位值
22	DMA2	DMA2 复位 (DMA2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
21	DMA1	DMA1 复位 (DMA1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
20: 13	Reserved	保留, 必须保持复位值
12	CRC	CRC 复位 (CRC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
11	Reserved	保留, 必须保持复位值

Bit	Field	Description
10	SDIO	SDIO 复位 (SDIO Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
9: 8	Reserved	保留, 必须保持复位值
7	GPIOH	GPIOH 复位 (GPIOH Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
6	GPIOG	GPIOG 复位 (GPIOG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
5	GPIOF	GPIOF 复位 (GPIOF Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
4	GPIOE	GPIOE 复位 (GPIOE Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
3	GPIOD	GPIOD 复位 (GPIOD Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
2	GPIOC	GPIOC 复位 (GPIOC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
1	GPIOB	GPIOB 复位 (GPIOB Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
0	GPIOA	GPIOA 复位 (GPIOA Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

4.3.8 RCC_APB2RSTR APB2 外设复位寄存器

偏移地址: 0x18

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.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP	SYSCFG	Res.	SPI1	Res.	ADC3	ADC2	ADC1	Res.		UART6	UART1	Res.		TIM8	TIM1
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	COMP	COMP 复位 (Comparator Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
14	SYSCFG	SYSCFG 复位 (SYSCFG 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	Reserved	保留，必须保持复位值
10	ADC3	ADC3 复位 (ADC3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
9	ADC2	ADC2 复位 (ADC2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
8	ADC1	ADC1 复位 (ADC1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
7: 6	Reserved	保留，必须保持复位值
5	UART6	UART6 复位 (UART6 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

Bit	Field	Description
4	UART1	UART1 复位 (UART1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
3: 2	Reserved	保留, 必须保持复位值
1	TIM8	TIM8 定时器复位 (TIM8 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
0	TIM1	TIM1 定时器复位 (TIM1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

4.3.9 RCC_APB1RSTR APB1 外设复位寄存器

偏移地址: 0x1C

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
UART8	UART7	DAC	PWR/D BG	BKP	Res.	CAN	CRS	Res.	I2C2	I2C1	UART5	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
SPI3	SPI2	Res.	WWDG	Res.					TIM7	TIM6	TIM5	TIM4	TIM3	TIM2	Res.
rw	rw								rw	rw	rw	rw	rw	rw	

Bit	Field	Description
31	UART8	UART8 复位 (UART8 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
30	UART7	UART7 复位 (UART7 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
29	DAC	DAC 复位 (DAC Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

Bit	Field	Description
28	PWR/DBG	PWR/DBG 复位 (Power Interface/DBG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
27	BKP	BKP 复位 (Backup Interface 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	CAN	CAN 复位 (CAN Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
24	CRS	CRS 复位 (CRS Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
23	Reserved	保留, 必须保持复位值
22	I2C2	I2C2 复位 (I2C2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
21	I2C1	I2C1 复位 (I2C1 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
20	UART5	UART5 复位 (UART5 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
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	Reserved	保留, 必须保持复位值

Bit	Field	Description
15	SPI3	SPI3 复位 (SPI3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
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: 6	Reserved	保留, 必须保持复位值
5	TIM7	TIM7 复位 (Timer7 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
4	TIM6	TIM6 复位 (Timer6 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
3	TIM5	TIM5 复位 (Timer5 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
2	TIM4	TIM4 复位 (Timer4 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
1	TIM3	TIM3 复位 (Timer3 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
0	TIM2	TIM2 复位 (Timer2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位

4.3.10 RCC_AHB3ENR AHB3 外设时钟使能寄存器

偏移地址: 0x20

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.

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

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值
0	FSMC	FSMC 时钟使能 (FSMC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

4.3.11 RCC_AHB2ENR AHB2 外设时钟使能寄存器

偏移地址: 0x24

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.

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	USBOTGFS	USBOTGFS (USBOTGFS Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
6: 0	Reserved	保留, 必须保持复位值

4.3.12 RCC_AHB1ENR AHB1 外设时钟使能寄存器

偏移地址: 0x28

复位值: 0x0000 6000

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.									DMA2	DMA1	Res.				
									RW	RW					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SRAM	Flash	CRC	Res.	SDIO	Res.		GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
	RW	RW	RW		RW			RW							

Bit	Field	Description
31: 23	Reserved	保留, 必须保持复位值
22	DMA2	DMA2 时钟使能 (DMA2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
21	DMA1	DMA1 时钟使能 (DMA1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
20: 15	Reserved	保留, 必须保持复位值
14	SRAM	SRAM 时钟使能 (SRAM Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
13	Flash	FLASH 时钟使能 (FLASH Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
12	CRC	CRC 时钟使能 (CRC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
11	Reserved	保留, 必须保持复位值
10	SDIO	SDIO 时钟使能 (SDIO Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
9: 8	Reserved	保留, 必须保持复位值

Bit	Field	Description
7	GPIOH	GPIOH 时钟使能 (GPIOH Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
6	GPIOG	GPIOG 时钟使能 (GPIOG Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
5	GPIOF	GPIOF 时钟使能 (GPIOF Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
4	GPIOE	GPIOE 时钟使能 (GPIOE Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
3	GPIOD	GPIOD 时钟使能 (GPIOD Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
2	GPIOC	GPIOC 时钟使能 (GPIOC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
1	GPIOB	GPIOB 时钟使能 (GPIOB Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
0	GPIOA	GPIOA 时钟使能 (GPIOA Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

4.3.13 RCC_APB2ENR APB2 外设时钟使能寄存器

偏移地址: 0x2C

复位值: 0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

注: 当外设时钟没有启动时, 软件不能读出外设寄存器的数值

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP	SYSCFG G	Res.	SPI1	Res.	ADC3	ADC2	ADC1	Res.	UART6	UART1	Res.	TIM8	TIM1		
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	COMP	COMP 时钟使能 (COMP Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
14	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
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	Reserved	保留, 必须保持复位值
10	ADC3	ADC3 接口时钟使能 (ADC3 Interface Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
9	ADC2	ADC2 接口时钟使能 (ADC2 Interface Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
8	ADC1	ADC1 接口时钟使能 (ADC1 Interface Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
7: 6	Reserved	保留, 必须保持复位值
5	UART6	UART6 时钟使能 (UART6 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
4	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

Bit	Field	Description
3: 2	Reserved	保留，必须保持复位值
1	TIM8	TIM8 时钟使能 (Timer8 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
0	TIM1	TIM1 时钟使能 (Timer1 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

4.3.14 RCC_APB1ENR APB1 外设时钟使能寄存器

偏移地址: 0x30

复位值: 0x0000 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
UART8	UART7	DAC	PWR/D BG	BKP	Res.	CAN	CRS	Res.	I2C2	I2C1	UART5	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
SPI3	SPI2	Res.	WWDG	Res.						TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
rw	rw									rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	UART8	UART8 时钟使能 (UART8 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
30	UART7	UART7 时钟使能 (UART7 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
29	DAC	DAC 时钟使能 (DAC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

Bit	Field	Description
28	PWR/DBG	PWR/DBG 时钟使能 (Power Interface/DBG Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
27	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
26	Reserved	保留, 必须保持复位值
25	CAN	CAN 时钟使能 (CAN Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
24	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
23	Reserved	保留, 必须保持复位值
22	I2C2	I2C2 时钟使能 (I2C2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
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	UART5	UART5 时钟使能 (UART5 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
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
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	Reserved	保留, 必须保持复位值

Bit	Field	Description
15	SPI3	SPI3 时钟使能 (SPI3 clock enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
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: 6	Reserved	保留, 必须保持复位值
5	TIM7	TIM7 时钟使能 (Timer7 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
4	TIM6	TIM6 时钟使能 (Timer6 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
3	TIM5	TIM5 时钟使能 (Timer5 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
2	TIM4	TIM4 时钟使能 (Timer4 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
1	TIM3	TIM3 时钟使能 (Timer3 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
0	TIM2	TIM2 时钟使能 (Timer2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能

4.3.15 RCC_BDCR 备份域控制寄存器

偏移地址: 0x34

复位值: 0x0000 0000, 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 LSEON、LSEBYP、RTCSEL 和 RTCEN 位在备份域内。因此, 在复位之后, 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”。任何内部或外部复位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.							
RW									RW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RTCEN	Res.				RTCSEL		Res.				LSEBYP	LSERD Y	LSEON			
					RW									RW	r	w

Bit	Field	Description
31: 25	Reserved	保留, 必须保持复位值
24	DBP	取消备份域的写保护, 在复位后, 备份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 和备份寄存器的访问 1: 允许对 RTC 和备份寄存器的访问
23: 17	Reserved	保留, 必须保持复位值
16	BDRST	备份域软件复位 (Backup Domain Software Reset) 通过软件置“1”或清“0” 0: 复位未激活 1: 复位整个备份域
15	RTCEN	RTC 时钟使能 (RTC Clock Enable) 通过软件置“1”或清“0” 0: 禁止 RTC 时钟 1: 使能 RTC 时钟
14: 10	Reserved	保留, 必须保持复位值
9: 8	RTCSEL	RTC 时钟源选择 (RTC Clock Source Selection) 由软件设置来选择 RTC 的时钟源。一旦选择了 RTC 时钟源, 就不能再更改它, 除非备份域被复位。可以使用 BDRST 位来复位它们。 00: 没有时钟 01: LSE 振荡器用作 RTC 时钟 10: LSI 振荡器用作 RTC 时钟 11: HSE 振荡器在 128 分频后用作 RTC 时钟
7: 3	Reserved	保留, 必须保持复位值

Bit	Field	Description
2	LSEBYP	外部低速振荡器旁路 (External Low-speed Oscillator Bypass) 通过软件置“1”来旁路 LSE。 只有在外部 32KHz 振荡器被禁止时，才能写入此位。 0: 禁止 LSE 振荡器旁路模式 1: 使能 LSE 振荡器旁路模式
1	LSERDY	外部低速时钟稳定 (External Low-speed Oscillator Ready) 通过软件置“1”或清“0”来指示外部 32KHz 振荡器是否稳定。 在 LSEON 被清除后，在 3 个 AHB 时钟周期后变低。 0: 外部 32KHz 振荡器未稳定 1: 外部 32KHz 振荡器已稳定
0	LSEON	外部低速振荡器使能 (External Low-speed Oscillator Enable) 通过软件置“1”或清“0” 0: 禁止外部 32KHz 振荡器 1: 使能外部 32KHz 振荡器

4.3.16 RCC_CSR 控制状态寄存器

偏移地址: 0x38

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
LPWRRSTF	WWDG	IWDGR	SFTRST	PORRS	PINRST		RMVF	LOCKU	PVDRS						
r	r	r	r	r	r	Res.		PF	TF						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								LOCKU	PVDRS	LSIOE			LSIRDY	LSION	
								PEN	TEN						r
								rw	rw	rw					rw

Bit	Field	Description
31	LPWRRSTF	低功耗管理复位标志 (Low Power Reset Flag) 在低功耗管理复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无低功耗管理复位 1: 发生低功耗管理复位

Bit	Field	Description
30	WWDGRSTF	窗口看门狗复位标志 (Window Watchdog Reset Flag) 在窗口看门狗复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无窗口看门狗复位发生 1: 发生窗口看门狗复位
29	IWDGRSTF	独立看门狗复位标志 (Independent Watchdog Reset Flag) 在独立看门狗复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无独立看门狗复位发生 1: 发生独立看门狗复位
28	SFTRSTF	软件复位标志 (Software Reset Flag) 在软件复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无软件复位发生 1: 发生软件复位
27	PORRSTF	上电/掉电复位标志 (POR/PDR Reset Flag) 在上电/掉电复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无上电/掉电复位发生 1: 发生上电/掉电复位
26	PINRSTF	nRST 管脚复位标志 (PIN Reset Flag) 在 nRST 管脚复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 nRST 管脚复位发生 1: 发生 nRST 管脚复位
25	Reserved	保留，必须保持复位值
24	RMVF	清除复位标志 (Remove Reset Flag) 由软件置“1”来清除复位标志。 0: 无效 1: 清除复位标志
23	LOCKUPF	CPU 死锁复位标志 (CPU Lockup Reset Flag) 在 CPU 发生死锁复位时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 CPU 死锁复位发生 1: 发生 CPU 死锁复位
22	PVDRSTF	PVD 复位标志 (PVD Reset Flag) 在 PVD 复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0: 无 PVD 复位发生 1: 发生 PVD 复位
21: 8	Reserved	保留，必须保持复位值

Bit	Field	Description
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 时钟输出使能 (LSI Output Enable) 通过软件置“1”或清“0”，或由电源复位清除。 0: LSI 时钟输出禁止 1: LSI 时钟输出使能 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

4.3.17 RCC_SYSCFG 系统配置寄存器

偏移地址: 0x3C

复位值: 0x0000 0101

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					HSE_RFB_SEL							DATAP	PROG_	
	EN												REFET	CHECK	
	RW					RW							CH	_EN	
													RW		r

Bit	Field	Description
31: 15	Reserved	保留，必须保持复位值

Bit	Field	Description
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	反馈电阻选择: 00: 2M 01: 1M (默认适用 8M~12M 晶振) 10: 500K 11: 200K
7: 3	Reserved	保留, 必须保持复位值
2	DATAPREFETCH	DATA 预取模块使能位 0: 使能 1: 禁止
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 0: 不检查 1: 检查 (硬件固定为 1)

4.3.18 RCC_CFGR2 时钟配置寄存器 2

偏移地址: 0x40

复位值: 0x0003 1F00

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.												APB1_CLK_HV_PRE			
												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												TIMADV_PRE		TIMADV_CKSEL	
												rw		rw	

Bit	Field	Description
31: 20	Reserved	保留, 必须保持复位值

Bit	Field	Description
19: 16	APB1_CLK_HV_PRE	APB1_HV 输出时钟分频系数 0011: 8 分频 0100: 10 分频 0101: 12 分频 0110: 14 分频 0111: 16 分频 1000: 18 分频 1001: 20 分频 1010: 22 分频 1011: 24 分频 1100: 26 分频 1101: 28 分频 1110: 30 分频 1111: 32 分频 注: 配置值必须介于 0x03-0x0F
15: 13	Reserved	保留, 必须保持复位值
12: 8	Reserved	保留, 必须保持复位值。
7: 4	Reserved	保留, 必须保持复位值
3: 1	TIMADV_PRE	TIMADV_CLK 预分频 由软件来控制系统时钟 (SYSCLK) 的预分频系数。 0xx: TIMADV_CLK 不分频 100: TIMADV_CLK 2 分频 101: TIMADV_CLK 4 分频 110: TIMADV_CLK 8 分频 111: TIMADV_CLK 16 分频
0	TIMADV_CKSEL	TIMADV 时钟选择 0: 选择 PCLK2 时钟源 APB2 时钟不分频时, 高级定时器频率等于 APB2 时钟 (PCLK2) 频率; APB2 时钟分频时, 高级定时器频率等于 APB2 时钟 (PCLK2) 的 2 倍频 1: 选择 SYSCLK 经过配置 TIMADV_PRE 分频后的 TIMADV_CLK 时钟

4.3.19 RCC_PLLCFGR PLL 配置寄存器

偏移地址: 0x48

复位值: 0x0018 031C

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

Res.	PLLDIV	Res.	PLL_ICTRL	PLLXTPRE	PLLSRC
	rw			rw	rw

Bit	Field	Description
31: 23	Reserved	保留, 必须保持复位值
22: 16	PLLMUL	PLL 倍频系数 (PLL Multiplication Factor)
15: 11	Reserved	保留, 必须保持复位值
10: 8	PLLDIV	PLL 分频系数 (PLL Divide Factor)
7: 6	Reserved	保留, 必须保持复位值
5: 4	Reserved	保留, 必须保持复位值。
3: 2	PLL_ICTRL	PLL Charge Pump 电流控制信号 (PLL CP Current Control Signals) 默认值为 2'b11。当 PLL 输入时钟源低于 8MHz 时, 推荐设置此位为 2'b01。
1	PLLXTPRE	HSE 时钟用作 PLL 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 PLL 输入时钟。 只有当 PLL 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	PLLSRC	PLL 输入时钟源选择 (PLL Entry Clock Source Selection) 通过软件置“1”或清除来选择 PLL 输入时钟源。 只有当 PLL 被禁止时, 才能写入此位。 0: HSI 时钟用作 PLL 输入时钟 1: HSE 时钟用作 PLL 输入时钟

注: 配置 RCC_PLLCFGR 寄存器请参考功能描述的 PLL 章节

5 SYSCFG 系统控制器

5.1 简介

该芯片具有一组系统配置寄存器。这些寄存器的主要功能如下：

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

5.2 寄存器

5.2.1 寄存器总览

表 5-1 SYSCFG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR1	SYSCFG 配置寄存器 1	0x2000000X
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_CFGR2	SYSCFG 配置寄存器 2	0x00000000
0x1C	SYSCFG_PDETCR	SYSCFG 电源检测配置状态寄存器	0x00000000
0x20	SYSCFG_VOSDLY	SYSCFG 电压调试器输出延时寄存器	0x000001F4

5.2.2 SYSCFG_CFGR1 配置寄存器 1

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。复位后，此控制位值为实际的 BOOT PIN 配置决定。

偏移地址：0x00

复位值：0x2000 000X

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

Bit	Field	Description
31	Reserved	保留, 必须保持复位值
30:29	FC_MODESEL	FSMC 模式选择 (FSMC Mode Select) 01: 兼容 8080 协议接口 00: 兼容 NOR FLASH 接口 1x: 保留
28	FC_ODATAEN	FSMC 地址数据复用引脚只能作为数据使用(FSMC Addr/Data Pin Only Used as Data Enable) 1: 仅作为数据引脚使用 0: 允许作为数据或地址使用
27:2	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

5.2.3 SYSCFG_EXTICR1 外部中断配置寄存器 1

偏移地址: 0x08

复位值: 0x0000 0000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3				EXTI2				EXTI1				EXTI0			
rw				rw				rw				rw			

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

Bit	Field	Description
15:0	EXTIx	<p>EXTIx 配置 ($x=0\ldots3$) (EXTIx configuration) 选择 EXTIx 外部中断的输入源。</p> <p>0000: PA[x] 管脚 0001: PB[x] 管脚 0010: PC[x] 管脚 0011: PD[x] 管脚 0100: PE[x] 管脚 0101: PF[x] 管脚 0110: PG[x] 管脚 0111: PH[x] 管脚</p>

5.2.4 SYSCFG_EXTICR2 外部中断配置寄存器 2

偏移地址: 0x0C

复位值: 0x0000 0000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7				EXTI6				EXTI5				EXTI4			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留, 必须保持复位值
15:0	EXTIx	<p>EXTIx 配置 ($x=4\ldots7$) (EXTIx configuration) 选择 EXTIx 外部中断的输入源。</p> <p>0000: PA[x] 管脚 0001: PB[x] 管脚 0010: PC[x] 管脚 0011: PD[x] 管脚 0100: PE[x] 管脚 0101: PF[x] 管脚 0110: PG[x] 管脚 0111: PH[x] 管脚</p>

5.2.5 SYSCFG_EXTICR3 外部中断配置寄存器 3

偏移地址: 0x10

复位值: 0x0000 0000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11				EXTI10				EXTI9				EXTI8			
RW				RW				RW				RW			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置 (x=8...11) (EXTIx configuration) 选择 EXTIx 外部中断的输入源。 0000: PA[x] 管脚 0001: PB[x] 管脚 0010: PC[x] 管脚 0011: PD[x] 管脚 0100: PE[x] 管脚 0101: PF[x] 管脚 0110: PG[x] 管脚 0111: PH[x] 管脚

5.2.6 SYSCFG_EXTICR4 外部中断配置寄存器 4

偏移地址: 0x014

复位值: 0x0000 0000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15				EXTI14				EXTI13				EXTI12			
RW				RW				RW				RW			

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

Bit	Field	Description
15:0	EXTIx	<p>EXTIx 配置 ($x=12\ldots15$) (EXTIx configuration)</p> <p>选择 EXTIx 外部中断的输入源。</p> <p>0000: PA[x] 管脚 0001: PB[x] 管脚 0010: PC[x] 管脚 0011: PD[x] 管脚 0100: PE[x] 管脚 0101: PF[x] 管脚 0110: PG[x] 管脚 0111: PH[x] 管脚</p>

5.2.7 SYSCFG_CFRG2 配置寄存器 2

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.															

Bit	Field	Description
31:18	Reserved	保留, 必须保持复位值
17	I2C2_MODE_SEL	<p>I2C2 端口模式选择位</p> <p>0: 开漏模式 1: 推挽模式</p>
16	I2C1_MODE_SEL	<p>I2C1 端口模式选择位</p> <p>0: 开漏模式 1: 推挽模式</p>
15:0	Reserved	保留, 必须保持复位值

5.2.8 SYSCFG_PDETCSR 电源检测配置状态寄存器

偏移地址: 0x1C

复位值: 0x00000000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VBAT_D IV3_EN rw	Res.				PVDO	PLS				PVDE	rw	rw	rw	

Bit	Field	Description
31:12	Reserved	保留, 必须保持复位值
11	VBAT_DIV3_EN	VBATDIV3_EN: ADC 检测 VBAT_DIV3 分压值使能 1: 使用 ADC 检测 VBAT_DIV3 分压值 0: 不使用 ADC 检测 VBAT_DIV3 分压值
10:6	Reserved	保留, 必须保持复位值。
5	PVDO	PVDO: PVD 输出状态 1: PVD 输出高 0: PVD 输出低 注: PVDE 必须使能。
4:1	PLS	PLS: PVD 阈值选择 0000: 1.8V 0001: 2.1V 0010: 2.4V 0011: 2.7V 0100: 3.0V 0101: 3.3V 0110: 3.6V 0111: 3.9V 1000: 4.2V 1001: 4.5V 1010: 4.8V
0	PVDE	PVDE: PVD 使能 1: PVD 使能 0: PVD 禁止

5.2.9 SYSCFG_VOSDLY 电压调试器输出延时寄存器

偏移地址: 0x20

复位值: 0x0000 01F4

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

Res.

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

Res.	VOSDLY_CNT
	rw

Bit	Field	Description
31:10	Reserved	保留, 必须保持复位值
9:0	VOSDLY_CNT	电压调试器输出改变的延迟时间 (Delay Counter of Voltage Output Selection Change)

6 EXTI 中断和事件

6.1 简介

嵌套向量中断控制器（NVIC）连接处理器核，管理低延迟的异常和中断处理。NVIC 内部包含 3 位的中断优先级配置位，从而可提供 8 个中断优先级等级，其它更多的异常与 NVIC 编程的细节请参考《Cortex®-M3 技术参考手册》。

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。

6.2 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
- 除 GPIOH 外的所有 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

6.3 功能描述

6.3.1 功能框图

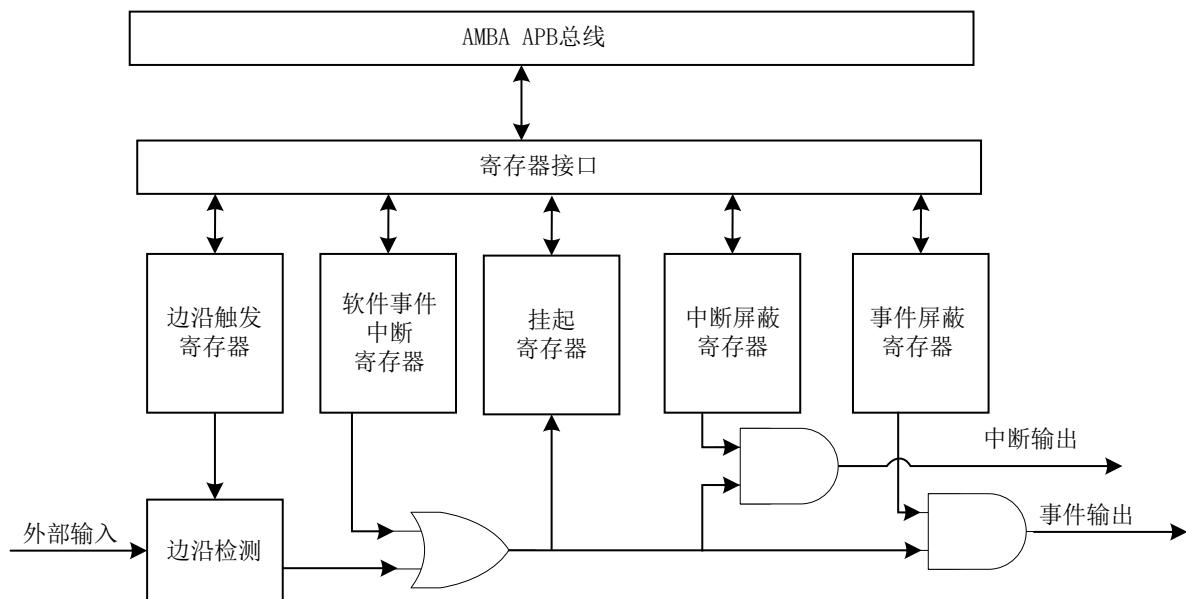


图 6-1 EXTI 结构框图

6.3.2 中断和异常向量

Cortex®-M3 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。获取向量与压栈操作是并行进行的，提高了中断的效率，下表分别列出了异常类型与中断向量。

表 6-1 异常向量表

位置	优先级	优先级类型	名称	说明	地址
				保留	0x0000 0000
	-3	固定	Reset	复位	0x0000 0004
	-2	固定	NMI	不可屏蔽中断 RCC 时钟安全系统(CSS)连接到 NMI	0x0000 0008
	-1	固定	硬件失效 (HardFault)	所有类型的失效	0x0000 000C
	0	可设置	存储管理 (MemManage)	存储器管理	0x0000 0010
	1	可设置	总线错误 (BusFault)	预取指失败，存储器访问失败	0x0000 0014
	2	可设置	错误应用 (UsageFault)	未定义的指令或非法状态	0x0000 0018
				保留	0x0000 001C~0x0000 002B

表 6-2 中断向量表

位置	优先级	优先级类型	名称	说明	地址
	3	可设置	SVCALL	通过 SWI 指令的系统服务调用	0x0000_002C
	4	可设置	调试监控 (DebugMonitor)	调试监控器	0x0000_0030
				保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG_IWDG	窗口看门狗中断与连到 EXTI21 的独立看门狗中断	0x0000_0040
1	8	可设置	PVD_VDT	连到 EXTI16 的电源电压检测(PVD)与 CORE 电压检测(VDT)中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	RTC	RTC 全局中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	RCC_CRS	RCC 和 CRS 全局中断	0x0000_0054
6	13	可设置	EXTI0	EXTI 线 0 中断	0x0000_0058
7	14	可设置	EXTI1	EXTI 线 1 中断	0x0000_005C
8	15	可设置	EXTI2	EXTI 线 2 中断	0x0000_0060
9	16	可设置	EXTI3	EXTI 线 3 中断	0x0000_0064
10	17	可设置	EXTI4	EXTI 线 4 中断	0x0000_0068
11	18	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_006C
12	19	可设置	DMA1 通道 2	DMA1 通道 2 全局中断	0x0000_0070
13	20	可设置	DMA1 通道 3	DMA1 通道 3 全局中断	0x0000_0074
14	21	可设置	DMA1 通道 4	DMA1 通道 4 全局中断	0x0000_0078
15	22	可设置	DMA1 通道 5	DMA1 通道 5 全局中断	0x0000_007C
16	23	可设置	DMA1 通道 6	DMA1 通道 6 全局中断	0x0000_0080
17	24	可设置	DMA1 通道 7	DMA1 通道 7 全局中断	0x0000_0084
18	25	可设置	ADC1_ADC2	ADC1 和 ADC2 全局中断	0x0000_0088
19	26	可设置	Cache	Cache 全局中断	0x0000_008C
20	27	-	-	保留	0x0000_0090
21	28	可设置	CAN1	CAN1 全局中断	0x0000_0094
22	29	-	-	保留	0x0000_0098
23	30	可设置	EXTI9_5	EXTI 线[9:5]中断	0x0000_009C
24	31	可设置	TIM1_BRK	TIM1 刹车中断	0x0000_00A0
25	32	可设置	TIM1_UP	TIM1 更新中断	0x0000_00A4
26	33	可设置	TIM1_TRG_COM	TIM1 触发、COM 中断	0x0000_00A8
27	34	可设置	TIM1_CC	TIM1 捕捉比较中断	0x0000_00AC
28	35	可设置	TIM2	TIM2 全局中断	0x0000_00B0
29	36	可设置	TIM3	TIM3 全局中断	0x0000_00B4
30	37	可设置	TIM4	TIM4 全局中断	0x0000_00B8
31	38	可设置	I2C1	I2C1 全局中断	0x0000_00BC

位置	优先级	优先级类型	名称	说明	地址
32	39	-	-	保留	0x0000_00C0
33	40	可设置	I2C2	I2C2 全局中断	0x0000_00C4
34	41	-	-	保留	0x0000_00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2 全局中断	0x0000_00D0
37	44	可设置	UART1	UART1 全局中断	0x0000_00D4
38	45	可设置	UART2	UART2 全局中断	0x0000_00D8
39	46	可设置	UART3	UART3 全局中断	0x0000_00DC
40	47	可设置	EXTI15_10	EXTI 线[15:10]中断	0x0000_00E0
41	48	可设置	RTC Alarm	连接 EXTI 17 的 RTC 闹钟中断	0x0000_00E4
42	49	-	-	保留	0x0000_00E8
43	50	可设置	TIM8_BRK	TIM8 刹车中断	0x0000_00EC
44	51	可设置	TIM8_UP	TIM8 更新中断	0x0000_00F0
45	52	可设置	TIM8_TRG_COM	TIM8 触发、COM 中断	0x0000_00F4
46	53	可设置	TIM8_CC	TIM8 捕获比较中断	0x0000_00F8
47	54	可设置	ADC3	ADC3 全局中断	0x0000_00FC
48	55	-	-	保留	0x0000_0100
49	56	可设置	SDIO	SDIO 全局中断	0x0000_0104
50	57	可设置	TIM5	TIM5 全局中断	0x0000_0108
51	58	可设置	SPI3	SPI3 全局中断	0x0000_010C
52	59	可设置	UART4	UART4 全局中断	0x0000_0110
53	60	可设置	UART5	UART5 全局中断	0x0000_0114
54	61	可设置	TIM6	TIM6 全局中断	0x0000_0118
55	62	可设置	TIM7	TIM7 全局中断	0x0000_011C
56	63	可设置	DMA2 通道 1	DMA2 通道 1 全局中断	0x0000_0120
57	64	可设置	DMA2 通道 2	DMA2 通道 2 全局中断	0x0000_0124
58	65	可设置	DMA2 通道 3	DMA2 通道 3 全局中断	0x0000_0128
59	66	可设置	DMA2 通道 4	DMA2 通道 4 全局中断	0x0000_012C
60	67	可设置	DMA2 通道 5	DMA2 通道 5 全局中断	0x0000_0130
-	-	-	-	保留	0x0000_0134 ~ 0x0000_013C
64	71	可设置	COMP1_2	连接 EXTI 19/20 的比较器 1/2 中断	0x0000_0140
-	-	-	-	保留	0x0000_0144 ~ 0x0000_0148
67	74	可设置	USB OTGFS	USB OTGFS 全局中断	0x0000_014C
-	-	-	-	保留	0x0000_0150 ~ 0x0000_0158
71	78	可设置	UART6	UART6 全局中断	0x0000_015C
-	-	-	-	保留	0x0000_0160 ~ 0x0000_0184
82	89	可设置	UART7	UART7 全局中断	0x0000_0188

位置	优先级	优先级类型	名称	说明	地址
83	90	可设置	UART8	UART8 全局中断	0x0000_018C
-	-	-	-	保留	0x0000_0190 ~ 0x0000_01B0

6.3.3 唤醒事件管理

所有 **EXTI** 线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。

6.3.4 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。

6.3.5 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**。

当配置 **EXTIx** ($x=31\sim0$) 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** ($x=31\sim0$) 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。

6.3.6 硬件事件输出

配置硬件事件源的具体步骤如下：

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

6.3.7 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）。

6.3.8 外部中断映射

除 GPIOH 外的所有 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。具体存在的连接关系如下表所示。

表 6-3 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PA0;PB0;PC0;PD0;PE0;PF0;PG0	SYSCFG_EXTICR1 寄存器中的 EXTI0
EXTI1	PA1;PB1;PC1;PD1;PE1;PF0;PG1	SYSCFG_EXTICR1 寄存器中的 EXTI1
EXTI2	PA2;PB2;PC2;PD2;PE2;PF2;PG2	SYSCFG_EXTICR1 寄存器中的 EXTI2
EXTI3	PA3;PB3;PC3;PD3;PE3;PF3;PG3	SYSCFG_EXTICR1 寄存器中的 EXTI3
EXTI4	PA4;PB4;PC4;PD4;PE4;PF4;PG4	SYSCFG_EXTICR2 寄存器中的 EXTI4
EXTI5	PA5;PB5;PC5;PD5;PE5;PF5;PG5	SYSCFG_EXTICR2 寄存器中的 EXTI5
EXTI6	PA6;PB6;PC6;PD6;PE6;PF6;PG6	SYSCFG_EXTICR2 寄存器中的 EXTI6
EXTI7	PA7;PB7;PC7;PD7;PE7;PF7;PG7	SYSCFG_EXTICR2 寄存器中的 EXTI7
EXTI8	PA8;PB8;PC8;PD8;PE8;PF8;PG8	SYSCFG_EXTICR3 寄存器中的 EXTI8
EXTI9	PA9;PB9;PC9;PD9;PE9;PF9;PG9	SYSCFG_EXTICR3 寄存器中的 EXTI9
EXTI10	PA10;PB10;PC10;PD10;PE10;PF10;PG10	SYSCFG_EXTICR3 寄存器中的 EXTI10
EXTI11	PA11;PB11;PC11;PD11;PE11;PF11;PG11	SYSCFG_EXTICR3 寄存器中的 EXTI11
EXTI12	PA12;PB12;PC12;PD12;PE12;PF12;PG12	SYSCFG_EXTICR4 寄存器中的 EXTI12
EXTI13	PA13;PB13;PC13;PD13;PE13;PF13;PG13	SYSCFG_EXTICR4 寄存器中的 EXTI13
EXTI14	PA14;PB14;PC14;PD14;PE14;PF14;PG14	SYSCFG_EXTICR4 寄存器中的 EXTI14
EXTI15	PA15;PB15;PC15;PD14;PE15;PF15;PG15	SYSCFG_EXTICR4 寄存器中的 EXTI15

其他的外部中断/事件控制器的连接如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 19 连接到比较器 1 输出
- EXTI 线 20 连接到比较器 2 输出
- EXTI 线 21 连接到 IWDG 中断

6.4 寄存器描述

6.4.1 寄存器总览

表 6-4 EXTI 寄存器总览

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

6.4.2 EXTI_IMR 中断屏蔽寄存器

偏移地址: 0x0

复位值: 0x0000 0000

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

6.4.3 EXTI_EMR 事件屏蔽寄存器

偏移地址: 0x04

复位值: 0x0000 0000

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

EMRx(x=15~0)

rw

Bit	Field	Description
31:0	EMRx	线 x 事件使能位 1: 配置该位为 1, 使能线 x 对应的事件 0: 配置该位为 0, 禁止线 x 对应的事件

6.4.4 EXTI_RTSR 上升沿触发选择寄存器

偏移地址: 0x08

复位值: 0x0000 0000

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

6.4.5 EXTI_FTSR 下降沿触发选择寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

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

6.4.6 EXTI_SWIER 软件中断事件寄存器

偏移地址: 0x10

复位值: 0x0000 0000

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

6.4.7 EXTI_PR 软件中断事件挂起寄存器

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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	线 x 触发挂起位 1: 发生了选择的触发请求 0: 没有发生触发请求 外部中断线上出现选择的边沿事件时，该位被置 1，写 1 清除该位，也可以通过改变边沿检测的极性清除。													

7 DMA1/2 直接存储器访问控制器

7.1 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。

DMA1、DMA2 具有相同的功能，DMA1 有 7 个通道，DMA2 有 5 个通道，本章节以 DMA1 表述。

7.2 功能框图

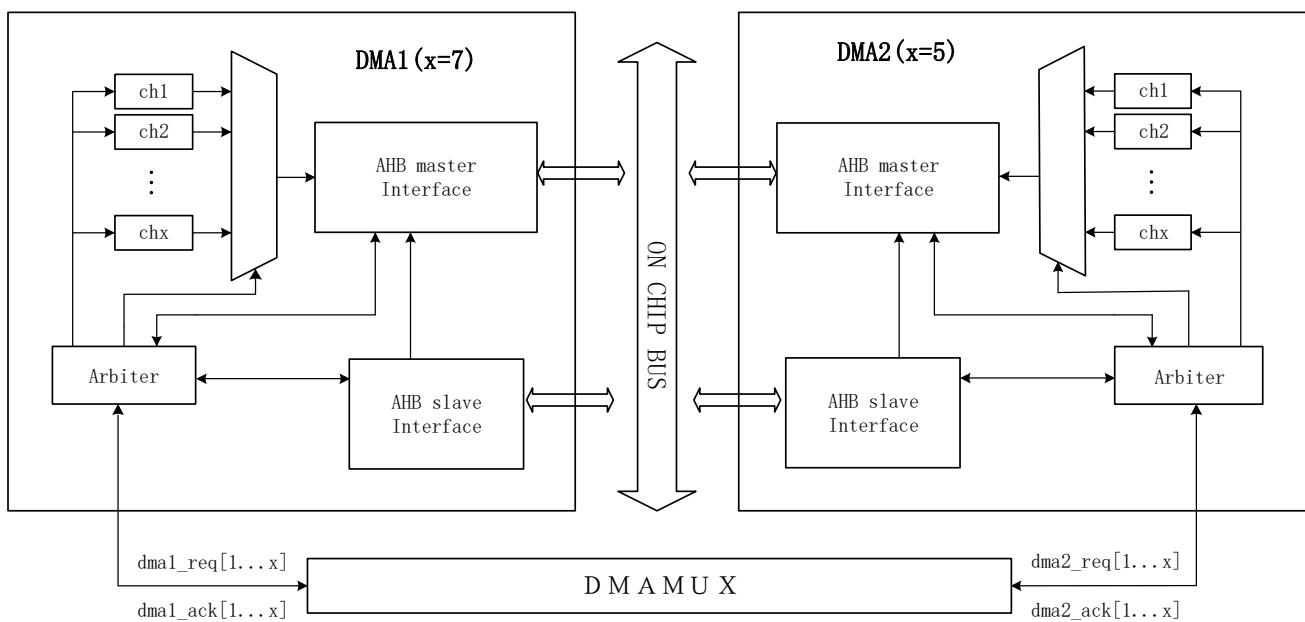


图 7-1 DMA 功能框图

7.3 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。
- 按照数据源头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 。

7.4 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 7-1 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

7.5 DMA

7.5.1 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 7-2 DMA1 各个通道的 DMA 请求一览

外设	DMA1 通道 1	DMA1 通道 2	DMA1 通道 3	DMA1 通道 4	DMA1 通道 5	DMA1 通道 6	DMA1 通道 7
ADC	ADC1	ADC2					
SPI/I2S		SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX		
UART	UART6_RX	UART3_TX	UART3_RX	UART1_TX	UART1_RX	UART2_RX	UART2_TX
I2C				I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
SDIO							
TIM1		TIM1_CC1	TIM1_CC2	TIM1_CC4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CC3	
TIM2	TIM2_CC3	TIM2_UP			TIM2_CC1		TIM2_CC2 TIM2_CC4

外设	DMA1 通道 1	DMA1 通道 2	DMA1 通道 3	DMA1 通道 4	DMA1 通道 5	DMA1 通道 6	DMA1 通道 7
TIM3		TIM3_CC3	TIM3_CC4 TIM3_UP			TIM3_CC1 TIM3_TRIG	
TIM4	TIM4_CC1			TIM4_CC2	TIM4_CC3		TIM4_UP

- 1. 如果 **SYSCFG_CFGR** 寄存器的对应映射位被复位, DMA 请求被映射在这个 DMA 通道。
- 2. 如果 **SYSCFG_CFGR** 寄存器的对应映射位被置位, DMA 请求被映射在这个 DMA 通道。

表 7-3 DMA2 各个通道的 DMA 请求一览

外设	DMA2 通道 1	DMA2 通道 2	DMA2 通道 3	DMA2 通道 4	DMA2 通道 5
ADC					ADC3
SPI/I2S	SPI3_RX	SPI3_TX			
UART	UART5_RX/ UART7_RX	UART5_TX/ UART7_TX	UART4_RX/ UART8_RX	UART6_TX	UART4_TX/ UART8_TX
SDIO				SDIO	
TIM5	TIM5_CC4 TIM5_TRIG	TIM5_CC3 TIM5_UP		TIM5_CC2	TIM5_CC1
TIM6			TIM6_UP/ DAC_CH1		
TIM7				TIM7_UP/ DAC_CH2	
TIM8	TIM8_CC3 TIM8_UP	TIM8_CC4 TIM8_TRIG TIM8_COM	TIM8_CC1		TIM8_CC2

7.6 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。

7.6.1 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。

7.6.2 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。

7.6.3 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。

7.6.3.1 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。

7.6.3.2 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。

7.6.3.3 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）。

7.6.3.4 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。

7.6.3.5 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。

7.6.4 可编程的数据传输宽度，对齐方式和数据大小端

当 PSIZE 和 MSIZE 不相同时，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 7-4 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 0x0 写 B0[7:0]
			在 0x1 读 B1[7:0]	在 0x1 写 B1[7:0]
			在 0x2 读 B2[7:0]	在 0x2 写 B2[7:0]
	16	16	在 0x3 读 B3[7:0]	在 0x3 写 B3[7:0]
			在 0x0 读 B1B0[15:0]	在 0x0 写 B1B0[15:0]
			在 0x2 读 B3B2[15:0]	在 0x2 写 B3B2[15:0]
	32	32	在 0x4 读 B5B4[15:0]	在 0x4 写 B5B4[15:0]
			在 0x6 读 B7B6[15:0]	在 0x6 写 B7B6[15:0]
			在 0x0 读 B3B2B1B0[31:0]	在 0x0 写 B3B2B1B0[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 0x4 读 B7B6B5B4[31:0]	在 0x4 写 B7B6B5B4[31:0]
			在 0x8 读 BBBAB9B8[31:0]	在 0x8 写 BBBAB9B8[31:0]
			在 0xC 读 BFBEBDBC[31:0]	在 0xC 写 BFBEBDBC[31:0]
			在 0x0 读 B0[7:0]	在 0x0 写 00B0[15:0]
	8	32	在 0x1 读 B1[7:0]	在 0x2 写 00B1[15:0]
			在 0x2 读 B2[7:0]	在 0x4 写 00B2[15:0]
			在 0x3 读 B3[7:0]	在 0x6 写 00B3[15:0]
			在 0x0 读 B0[7:0]	在 0x0 写 000000B0[31:0]
	16	32	在 0x1 读 B1[7:0]	在 0x4 写 000000B1[31:0]
			在 0x2 读 B2[7:0]	在 0x8 写 000000B2[31:0]
			在 0x3 读 B3[7:0]	在 0xC 写 000000B3[31:0]
			在 0x0 读 B1B0[15:0]	在 0x0 写 0000B1B0[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 0x2 读 B3B2[15:0]	在 0x4 写 0000B3B2[31:0]
			在 0x4 读 B5B4[15:0]	在 0x8 写 0000B5B4[31:0]
			在 0x6 读 B7B6[15:0]	在 0xC 写 0000B7B6[31:0]
			在 0x0 读 B1B0[15:0]	在 0x0 写 B0[7:0]
	32	8	在 0x2 读 B3B2B1B0[31:0]	在 0x1 写 B2[7:0]
			在 0x4 读 B7B6B5B4[31:0]	在 0x2 写 B4[7:0]
			在 0x8 读 BBBAB9B8[31:0]	在 0x3 写 B6[7:0]
			在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B0[7:0]
	32	16	在 0x0 读 B3B2B1B0[31:0]	在 0x1 写 B4[7:0]
			在 0x4 读 B7B6B5B4[31:0]	在 0x2 写 B8[7:0]
			在 0x8 读 BBBAB9B8[31:0]	在 0x3 写 BC[7:0]
			在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B1B0[15:0]

7.6.4.1 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更改与 32bit 对应的 4，目标数据宽度应配置为 32bit。

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

因此需要对数据进行复制扩展操作，0xDA 会被复制扩展为 4 个 8bit 组合为一个 32bit 数据 0xDADA_DADA，目标地址的 0x2，从设备接收到 32bit 的 0xDADA_DADA，会写 32bit 数据 0xDADA_DADA 到 0x0 地址上。这样目的地址 0x2 上的数据即为想要的 0xDA。

这种方法会写冗余的数据到目标寄存器上，但是可以保证写入目的地址上的数据是程序想要的，不会发生错误。

源数据宽度 8bit，会被复制扩充为 $4 \times 8\text{bit} = 32\text{bit}$ 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。

7.6.5 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。

7.7 DMA 寄存器描述

表 7-5 DMA 寄存器概览

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

7.7.1 DMA_ISR DMA 中断状态寄存器

偏移地址：0x00

复位值: 0x0000 0000

注: 该寄存器包含了所有 7 个 DMA 通道的寄存器配置, DMA2 只有 5 个 DMA 通道, 则通道 6, 7 的寄存器为预留位, 需保持复位值不变。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
				Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF16	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				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	
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 ($x=1\sim7$) (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 ($x=1\sim7$) (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 ($x=1\sim7$) (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 ($x=1\sim7$) (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

7.7.2 DMA_IFCR DMA 中断标志清除寄存器

偏移地址: 0x04

复位值: 0x0000 0000

注: 该寄存器包含了所有 7 个 DMA 通道的寄存器配置, DMA2 只有 5 个 DMA 通道, 则通道 6, 7 的寄存器为预留位, 需保持复位值不变。

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

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 ($x=1\sim7$) (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 ($x=1\sim7$) (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 ($x=1\sim7$) (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 ($x=1\sim7$) (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

7.7.3 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

注：该寄存器包含了所有 7 个 DMA 通道的寄存器配置，DMA2 只有 5 个 DMA 通道，则通道 6, 7 的寄存器为预留位，需保持复位值不变。

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE	MEM2MEM	PL		MSIZE		PSIZE		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
RW	RW	RW		RW		RW		RW	RW	RW	RW	RW	RW	RW	RW

Bit	Field	Description
31:16	Reserved	保留，始终读为 0

Bit	Field	Description
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 0:禁止自动重装载传输数量
14	MEM2MEM	存储器到存储器模式 (Memory to memory mode) 该位由软件置‘1’或清零。 0:关闭存储器到存储器模式 1:使能存储器到存储器模式
13:12	PL	通道优先级 (Channel priority level) 该位由软件置‘1’或清零。 00:低 01:中 10:高 11:最高
11:10	MSIZE	存储器数据宽度 (Memory size) 该位由软件置‘1’或清零。 00:8 bit 01:16 bit 10:32 bit 11:保留, 未定义
9:8	PSIZE	外设数据宽度 (Peripheral size) 该位由软件置‘1’或清零。 00:8 bit 01:16 bit 10:32 bit 11:保留, 未定义
7	MINC	存储器地址递增模式 (Memory increment mode) 该位由软件置‘1’或清零。 0:关闭存储器地址递增操作 1:使能存储器地址递增操作
6	PINC	外设地址递增模式 (Peripheral increment mode) 该位由软件置‘1’或清零。 0:关闭外设地址递增操作 1:使能外设地址递增操作
5	CIRC	循环模式 (Circular mode) 该位由软件置‘1’或清零。 0:关闭循环操作 1:使能循环操作
4	DIR	数据传输方向 (Date transfer direction) 该位由软件置‘1’或清零。 0:从外设读 1:从存储器读

Bit	Field	Description
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

7.7.4 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
15:0	NDT	数据传输数量 (Number of data to transfer) 数据传输数量为 0~65535。这个寄存器只能在通道关闭 (DMA_CCRx 的 EN=0) 时写入。通道使能后该寄存器变为不可写只读, 表示剩余多少次 DMA 传输。每次 DMA 传输后, 该寄存器数值递减。寄存器数值递减为 0, 表示数据全部传输完毕。此时若通道配置为自动重加载模式时, 寄存器的内容将被自动重新加载为之前配置时的数值。 与通道是否使能无关, 只要该寄存器为 0, DMA 就不会传输数据。

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

7.7.6 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。

8 BKP 备份寄存器

8.1 简介

备份寄存器含有 20 个 16 位的数据寄存器，用户端可用于存储 40 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。

8.2 主要特征

- 40 字节数据备份寄存器
- 提供存储 RTC 校准值的校准寄存器
- 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

8.3 功能描述

8.3.1 时钟校准

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

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

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

8.3.2 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。

8.4 寄存器

8.4.1 寄存器总览

表 8-1 BKP 寄存器概览

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

8.4.2 备份数据寄存器 n(**BKP_DRn**)(n = 1 .. 20)

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）。

8.4.3 时钟校准寄存器 (**BKP_RTCCR**)

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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
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 220 个时钟脉冲)。可以用来对 RTC 进行校准, 以 $1/220 * 106 \text{ ppm}$ 的比例减慢时钟。 注: RTC 时钟可以被减慢 0~121ppm。

8.4.4 备份控制寄存器 (BKP_CR)

偏移地址: 0x44

复位值: 0x0000

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

8.4.5 备份控制状态寄存器 (BKP_CSR)

偏移地址: 0x48

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	<p>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)。</p> <p>0: 无侵入中断 1: 产生侵入中断</p>
8	TEF	<p>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。</p> <p>0: 无侵入事件 1: 产生侵入事件</p>
7:3	Reserved	保留, 必须保持复位值
2	TPIE	<p>使能侵入引脚中断 (TAMP Pin Interrupt Enable) 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</p>
1	CTI	<p>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)</p>
0	CTE	<p>清除侵入检测事件标志 (Clear TAMP Event Flag), 该位只能写入, 读出值为 0。 0: 无效 1: 清除侵入检测事件标志 (TEF), 同时复位侵入检测器</p>

9 CRS 时钟回馈系统

9.1 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。

9.2 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
 - ◆ LSE 时钟输出
- 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）

9.3 功能描述

9.3.1 功能框图

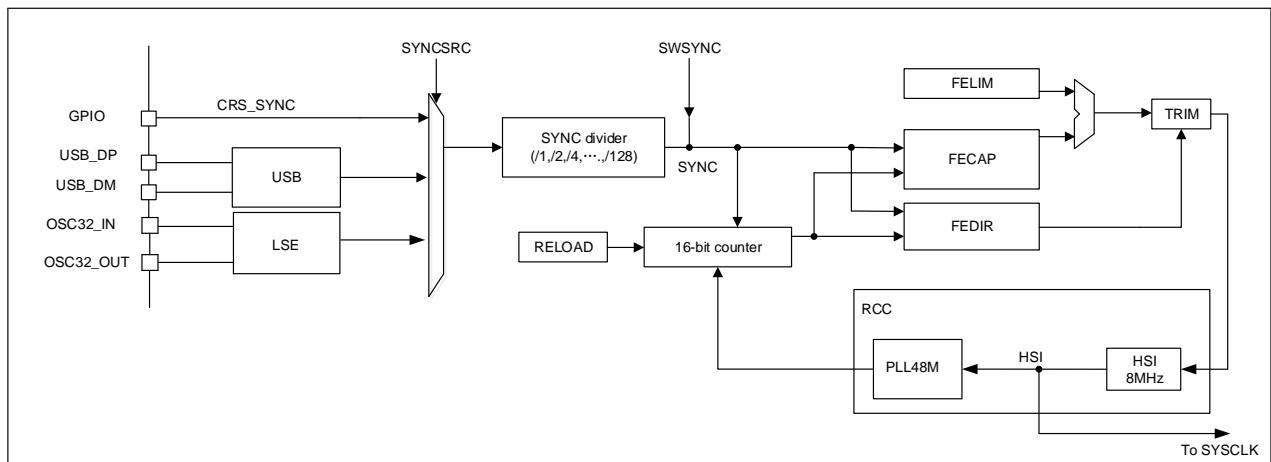


图 9-1 CRS 功能框图

9.3.2 信号描述

信号名	类型	描述
GPIO	输入	来自 GPIO 的同步源输入信号
USB_DP/USB_DM	输入	来自 USB 的同步源信号
OSC32_IN/OSC32_OUT	输入	来自 LSE 输出的同步源输入信号
HSI	输出	校准后的 HSI 时钟输出

9.3.3 功能概述

通过配置 CRS_CFGR 寄存器的 SYNC_SRC 位，选择同步源信号；CRS 将实际 48MHz 的 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，可以计算测出 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。

9.3.4 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 包或 LSE 时钟。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)。

9.3.5 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：

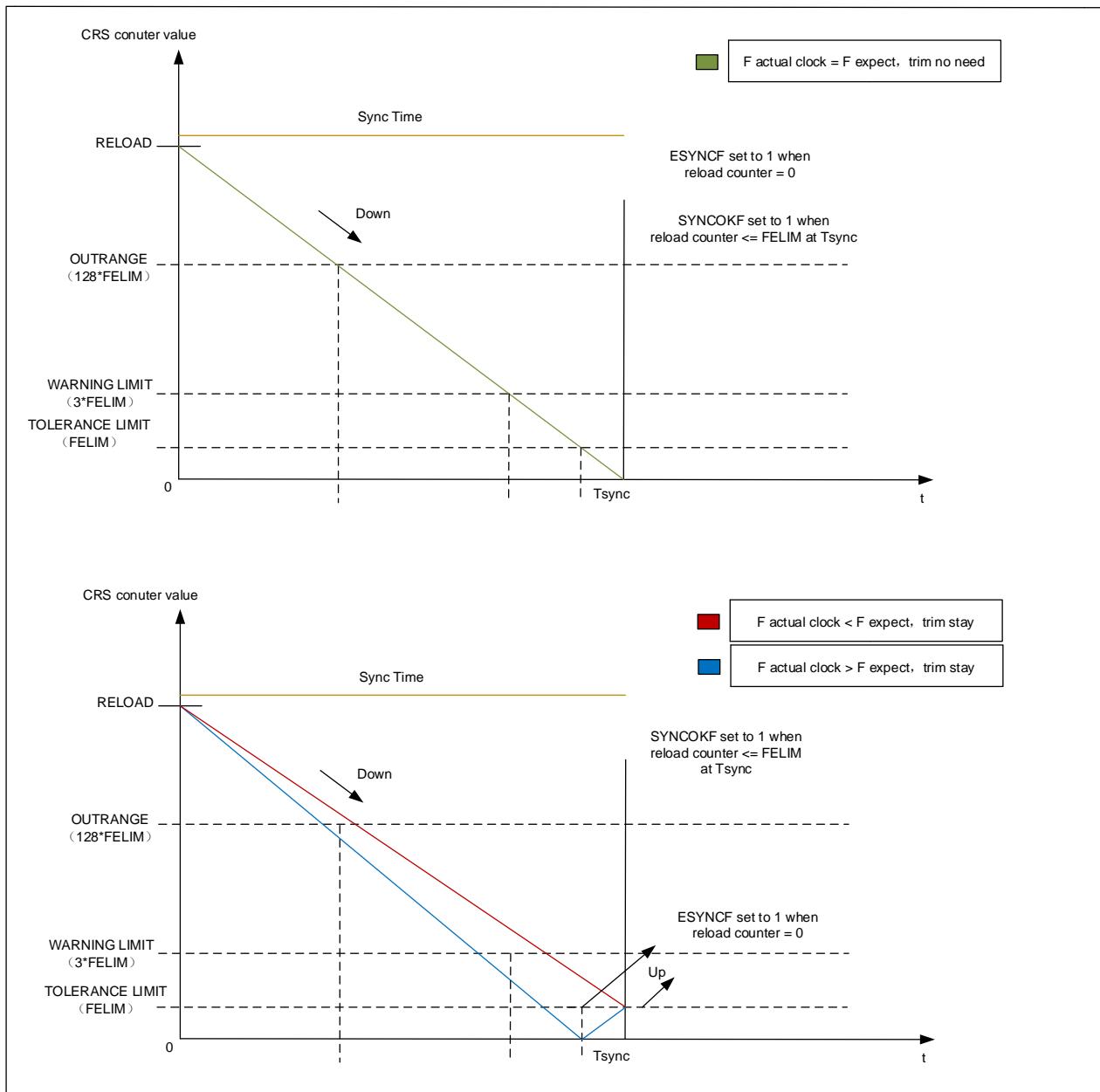
- 停止时 FEDIR=0，即向上计数，实际频率大于期望频率；
- 停止时 FEDIR=1，即向下计数，实际频率小于或等于期望频率。

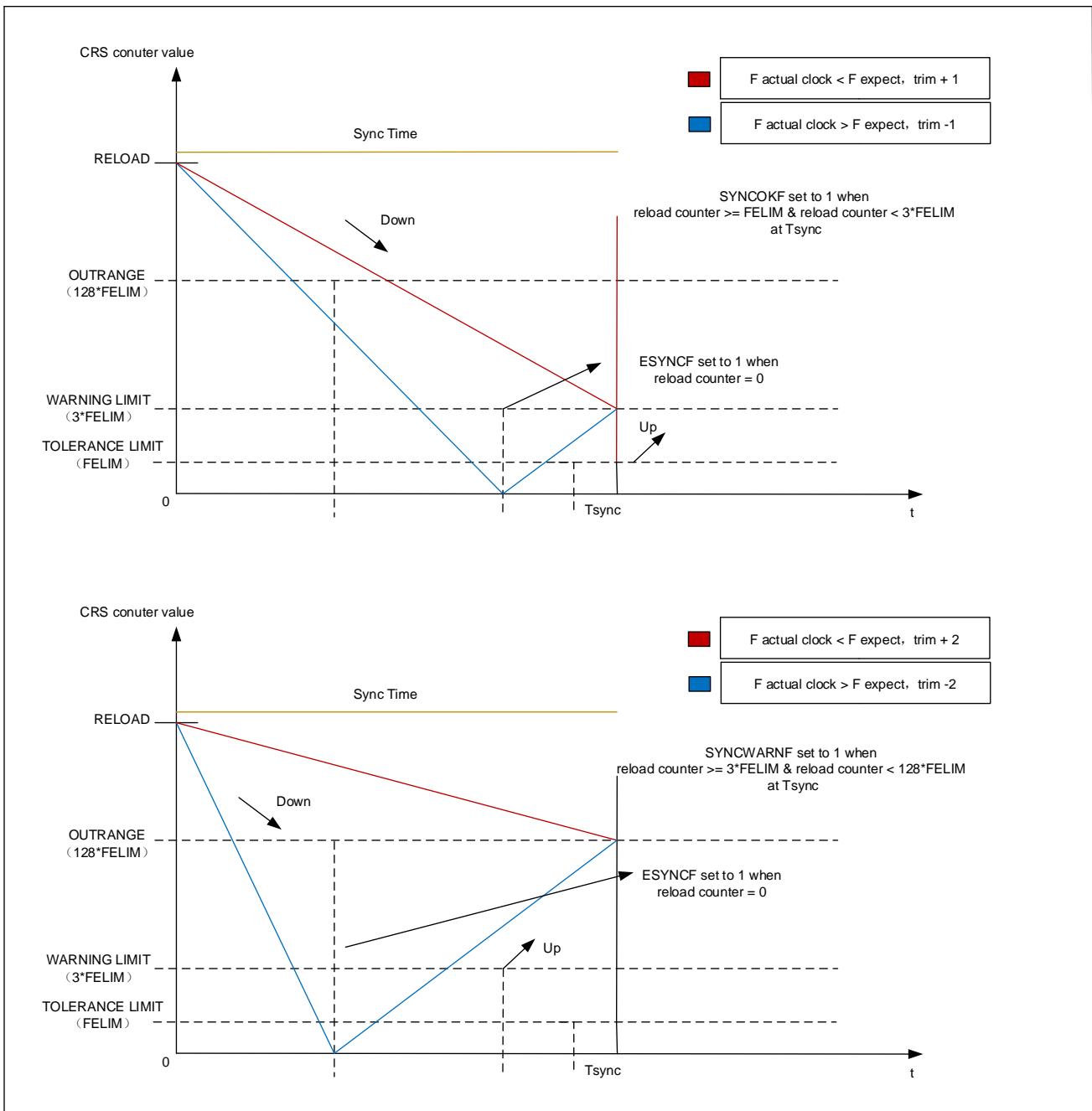
根据计数器因 SYNC 事件停止时的计数值与界限值（界限值可配置，为 CRS_CFGR 寄存器的 FELIM 位）之间的差值，可以得到微调时的步长：

- 停止时 FECAP<FELIM，可接受范围，不用微调；
- 停止时 FELIM<=FECAP<3*FELIM，误差稍大，微调步长建议为 1；
- 停止时 3*FELIM<=FECAP<128*FELIM，误差较大，微调步长建议为 2；
- 停止时 128*FELIM<=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各种情况。





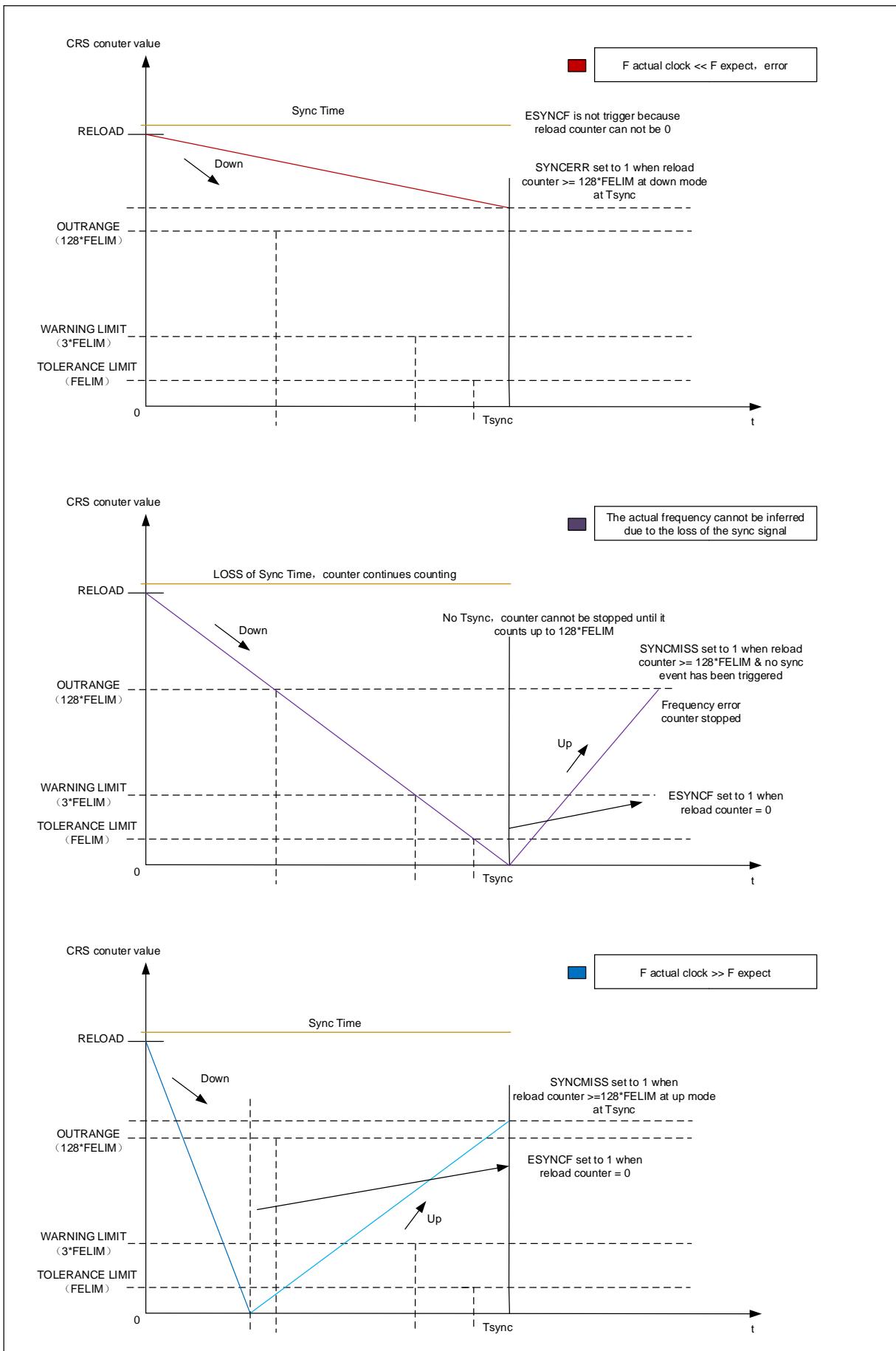


图 9-2 CRS 计数器状态图

9.3.6 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 $1 \times FELIM$ 为容忍界限 (TOLERANCE LIMIT)
- 定义 $3 \times FELIM$ 为警告界限 (WARNING LIMIT)
- 定义 $128 \times FELIM$ 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。

9.3.7 CRS 初始化及配置

9.3.7.1 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)。

9.3.7.2 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 FELIM 数值与调整步长的一半对应，公式如下：

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

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 $(f_{TARGET} / f_{SYNC}) = 48000$ 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。

9.4 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。

9.5 中断

表 9-1 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	ESYNCIE	ESYNCC
同步时误差可接受	SYNCOKF	SYNCOKE	SYNOKC
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

9.6 寄存器

9.6.1 寄存器总览

表 9-2 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

9.6.2 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														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) 默认值是 0x200, 这个值是修正档位的中间值。 当 RCC_ICSCR 的 TRIM_CRS_SEL 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 全 1: 频率最大</p>
7	SWSYNC	<p>软件同步行为 (Software SYNC Event) 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 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	CEN	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	ESYNCIE	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

9.6.3 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: LSE 时钟作为同步的信号源 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。

9.6.4 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.				TRIMOV	SYNCMI	SYNCE	Res.				ESYNC	ERRF	SYNCW	SYNCO
r					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 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 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 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 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 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 1: 产生同步或者修正错误信号</p>

Bit	Field	Description
1	SYNCWARNF	<p>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。</p> <p>0: 没有同步警告信号 1: 产生同步警告信号</p>
0	SYNCOKF	<p>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。</p> <p>0: 没有同步行为 OK 信号 1: 产生同步行为 OK 信号</p>

9.6.5 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	
													ESYNC C	ERRC	SYNCW ARNC	SYNCO KC
													W	W	W	W

Bit	Field	Description
31:4	Reserved	保留, 必须保持复位值
3	ESYNCC	<p>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。</p>
2	ERRC	<p>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。</p>
1	SYNCWARNC	<p>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。</p>
0	SYNCOKC	<p>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。</p>

10 GPIO 通用端口

10.1 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。

10.2 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
- 除 GPIOH 外的所有 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

10.3 功能描述

10.3.1 功能框图

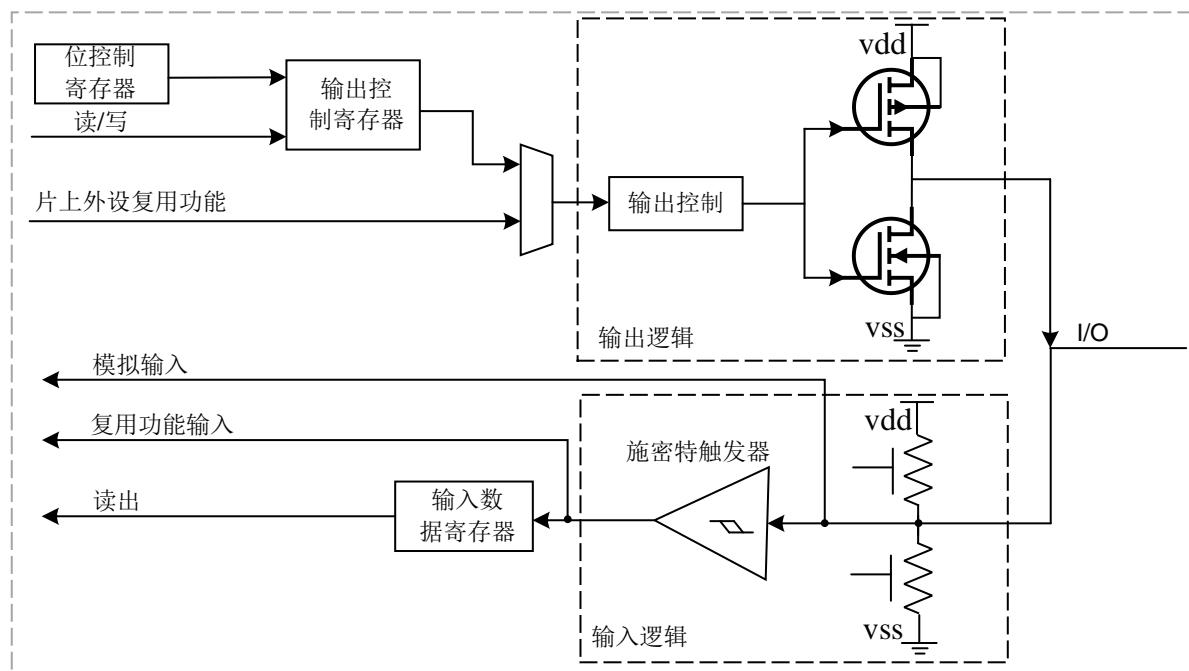


图 10-1 标准 I/O 端口

10.3.2 GPIO 端口配置

表 10-1 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 复用输入		浮空	x	x	0	1		x	
		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 CNFO 选择输入模式，配置 ODR 选择上下拉。

- 通用输出：

推挽输出：用户配置 MODE0 选择输出速度，配置 CNFO=00；

开漏输出：用户配置 MODE0 选择输出速度，配置 CNFO=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 CNFO=10；

推挽开漏输出：用户配置 MODE0 选择输出速度，配置 CNFO=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-Wire Debug pins)默认为输入 PU/PD 模式。

配置为通用输出模式后，输出数据寄存器(GPIOx_ODR)的值会输出到相应的 I/O 引脚。在每个 AHB 时钟周期，输入数据寄存器(GPIOX_IDR)捕捉 I/O 引脚上的数据。

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

10.3.3 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。

10.3.4 GPIO 锁定机制

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

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

- $\text{GPIO}_x\text{_LCKR}[16] = '1' + \text{LCKR}[15:0]$ 。

使能 GPIOA 的 PA[0] 端口锁定参考配置如下：

- $\text{GPIOA} \rightarrow \text{GPIOA_LCKR} = 0x10001$ 。
- $\text{GPIOA} \rightarrow \text{GPIOA_LCKR} = 0x00001$ 。
- $\text{GPIOA} \rightarrow \text{GPIOA_LCKR} = 0x10001$ 。

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， $\text{GPIO}_x\text{_LCKR}$ 寄存器的一个锁定位锁定端口配置寄存器（ $\text{GPIO}_x\text{_CRL}$ ）与（ $\text{GPIO}_x\text{_CRH}$ ）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。

10.3.5 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：

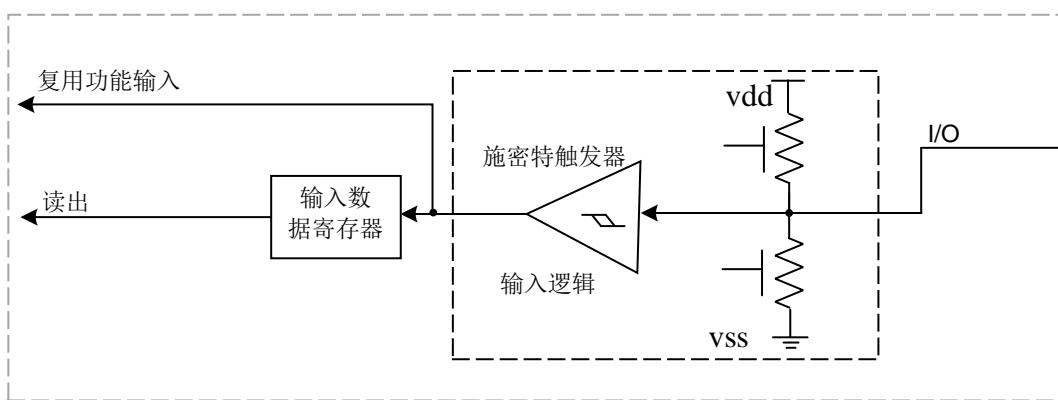


图 10-2 输入浮空/上拉/下拉配置

GPIOA 的 PA[0] 端口输入上拉参考配置如下：

- $\text{GPIOA} \rightarrow \text{GPIOA_ODR} = 0x0001$ 。
- $\text{GPIOA} \rightarrow \text{GPIOA_CRL} = 0x00000008$ 。

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。

10.3.6 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：

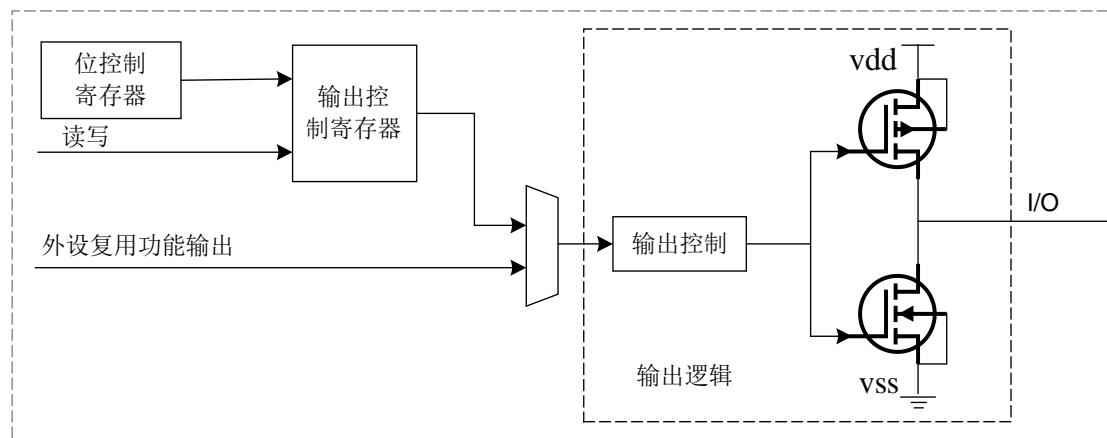


图 10-3 输出配置

10.3.7 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或不使用上下拉电阻。
- 当配置为输入时，可选弱上拉或弱下拉电阻或不使用上下拉电阻。
- 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。

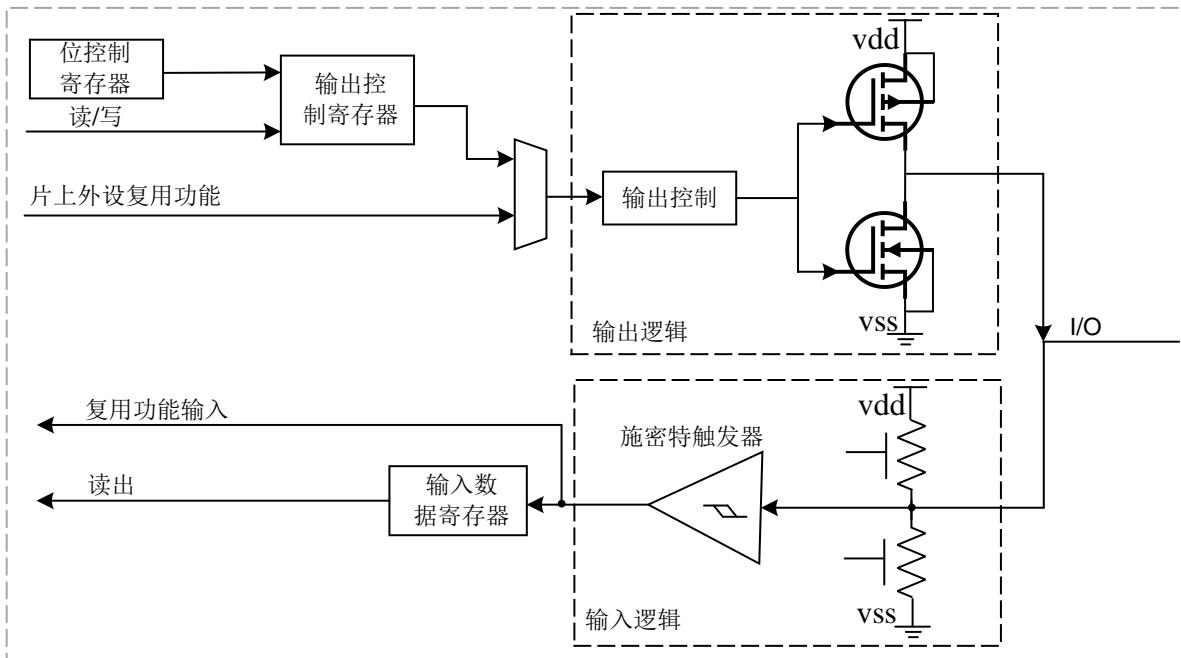


图 10-4 复用功能配置

10.3.8 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

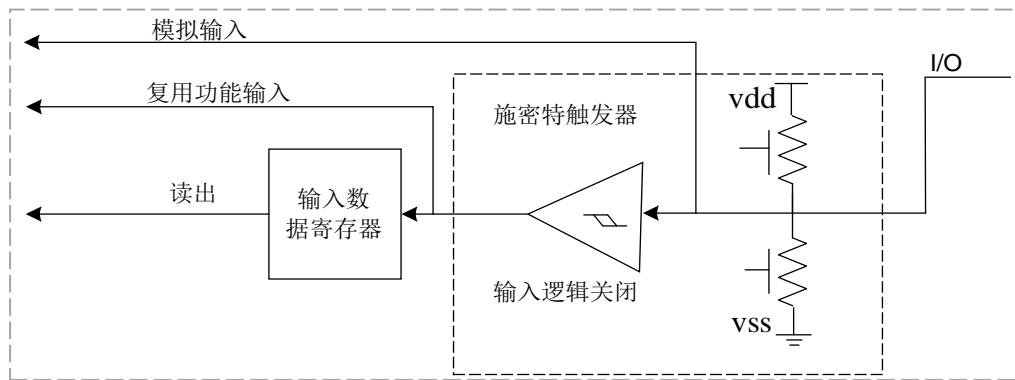


图 10-5 模拟输入

10.3.9 外部时钟复用 GPIO 端口

外部 HSE/LSE 时钟复用 GPIO，当对应的时钟 PIN 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。

10.3.10 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 10-2 SWD 复用功能重映射

复用功能	GPIO 端口
SWDIO	PA13
SWCLK	PA14

10.4 GPIO 寄存器描述

10.4.1 寄存器总览

表 10-3 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_AFRL	端口复用功能低位寄存器	0xFFFFFFFF

Offset	Acronym	Register Name	Reset
0x24	GPIOx_AFRH	端口复用功能高位寄存器	0x000FFFFF

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

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

10.4.3 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		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11		MODE11		CNF10		MODE10		CNF9		MODE9		CNF8		MODE8	
rw		rw		rw		rw		rw		rw		rw		rw	

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

10.4.4 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($y=15\sim0$)

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

10.4.5 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..H) 寄存器可以分别独立的对各个 ODR 位置 1 或清 0。													

10.4.6 GPIOx_BSRR 端口设置/清除寄存器

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
BRy(y=15~0)															
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

10.4.7 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													

10.4.8 GPIOx_LCKR 端口配置锁定寄存器

地址偏移: 0x18

复位值: 0x0000 0000

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

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

10.4.9 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

10.4.10 GPIOx_AFRL 端口复用功能低位寄存器

偏移地址: 0x20

复位值: GPIOx_AFRL (x = A..G): 0xFFFF FFFF

GPIOH_AFRL: 0x0000 00FF

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

10.4.11 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..G): 0xFFFF FFFF; GPIOH_AFRH: 0x0000 0000

GPIOA_AFRH: 0x000F 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	端口 x 的位 y (y = 8..15) 的复用功能选择位, 软件写配置。 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

11 TIM1/8 高级定时器

TIM1、TIM8 具有相同的功能，本章节统一表述为 TIMx，部分示意图以 TIM1 为例展示。

11.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 16 位计数方向可调的自动重装载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。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）。

11.2 功能框图

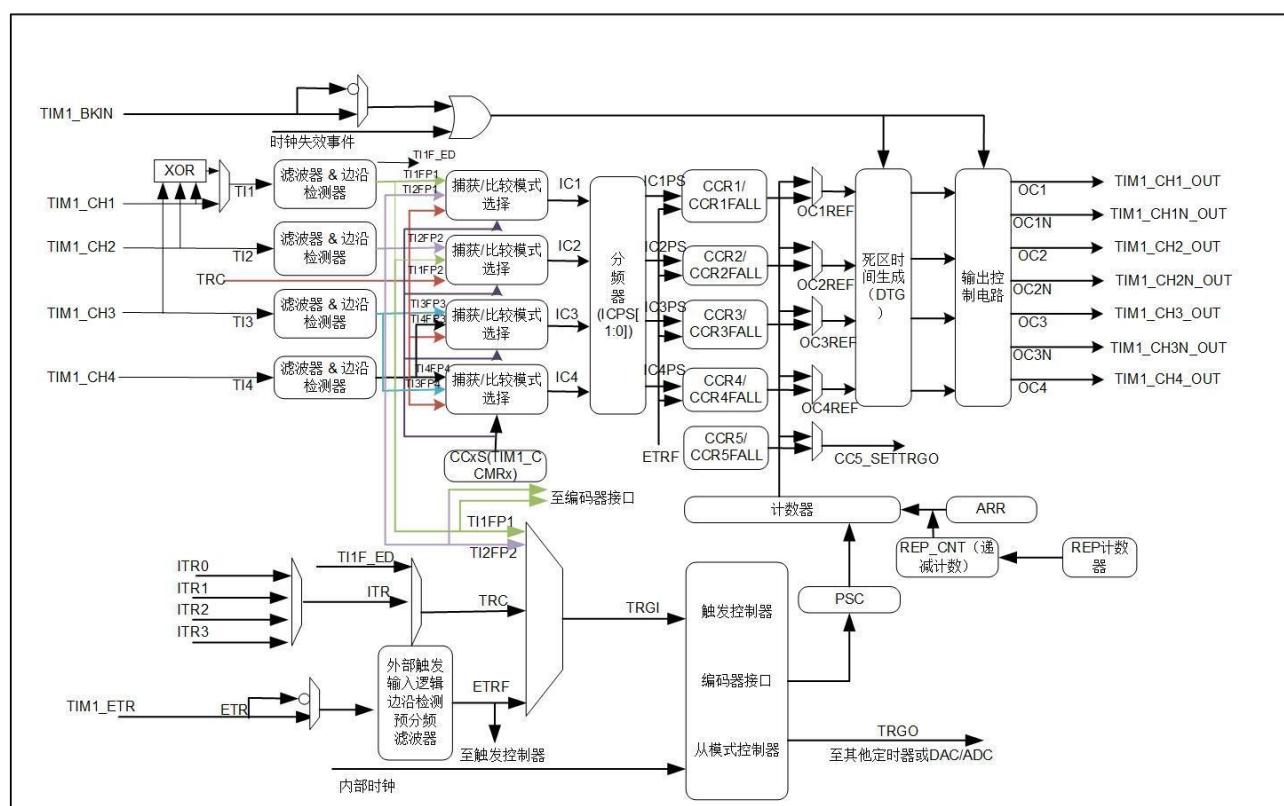


图 11-1 TIMx 结构图

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。

11.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 时钟源可选：内部时钟源，外部时钟输入 (Tlx、ETRx)，内部触发输入(ITRx)
- 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

11.4 功能描述

11.4.1 时钟

11.4.1.1 时钟选择

计数器的时钟源有以下几种：

- 内部时钟 (INT_CK)
- 外部时钟模式 1：外部触发输入 TRGI (包含 Tlx、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：

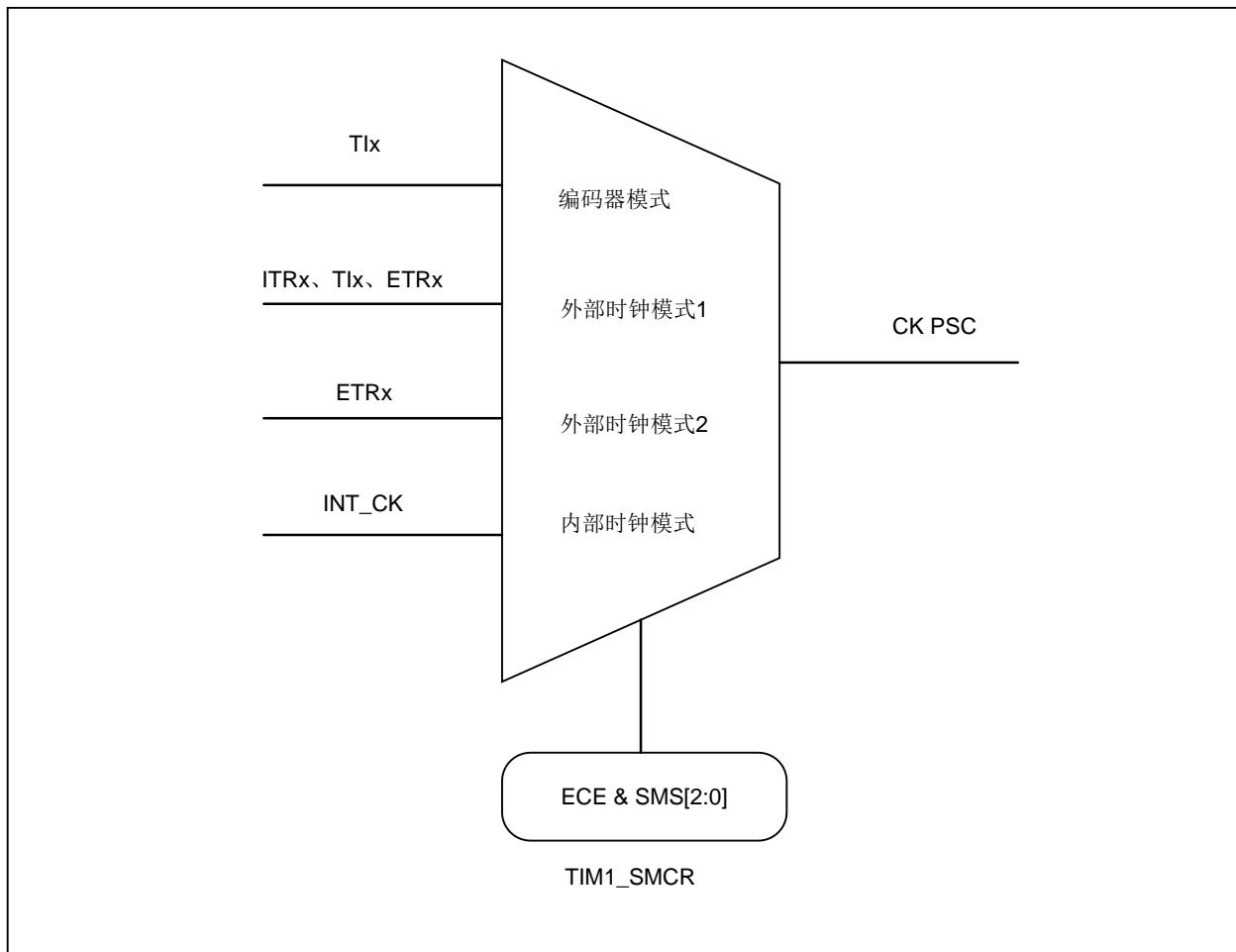


图 11-2 时钟选择

11.4.1.1.1 内部时钟源 (INT_CK)

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

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

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

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

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

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

当 **TI1** 出现有效边沿时, 计数器递增计数一次。**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。

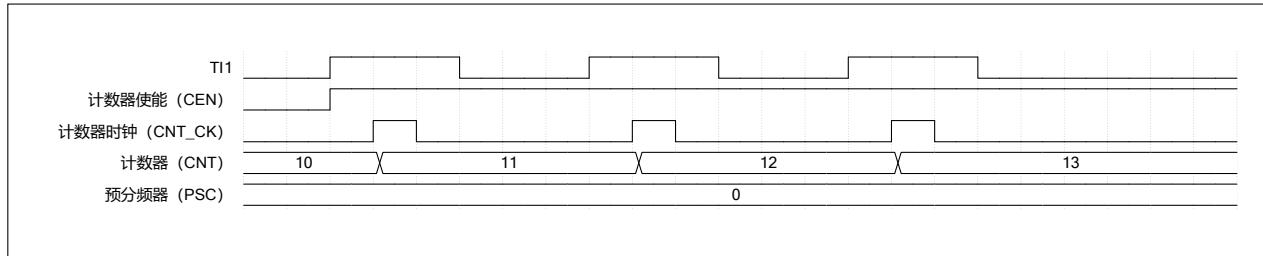


图 11-3 外部时钟模式 1 下的控制电路

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

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

例: **ETR** 的每 4 个下降沿计数一次, 递增计数, 具体配置如下:

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

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

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

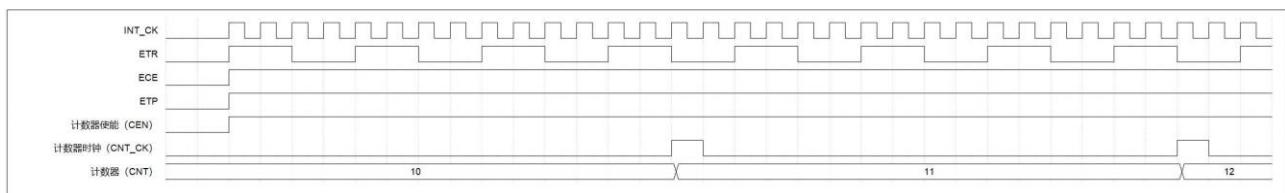


图 11-4 外部时钟模式 2 下的控制电路

11.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

11.4.1.2 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**) 和重复计数器寄存器 (**TIMx_RCR**)。

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

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

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

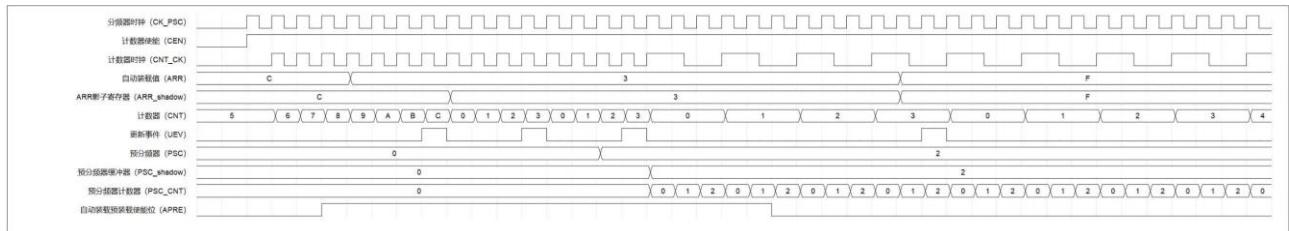


图 11-5 自动预装载

11.4.1.3 计数模式

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

11.4.1.3.1 递增计数模式

配置 **TIMx_CR1** 寄存器 CMS=0, DIR=0，选择递增计数模式。

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

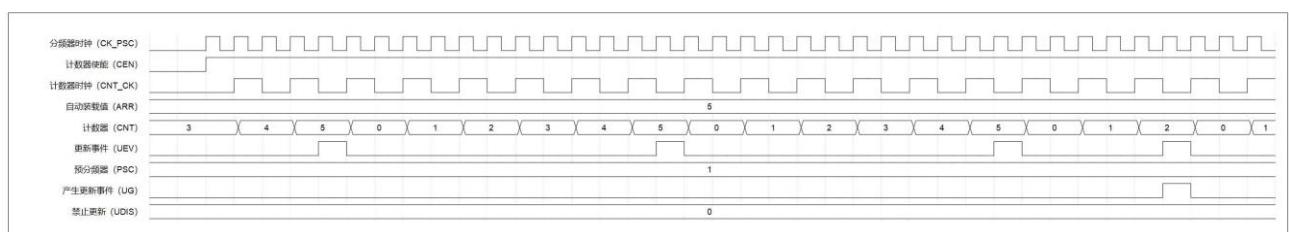


图 11-6 递增计数模式 (UDIS=0)

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

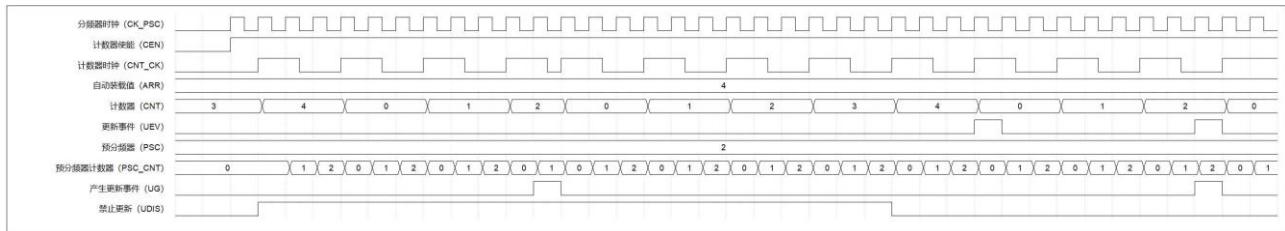


图 11-7 递增计数模式 (UDIS=1 禁止产生更新事件)

注：发生更新事件时

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

11.4.1.3.2 递减计数模式

配置 TIMx_CR1 寄存器的 CMS=0, DIR=1, 选择递减计数模式。

递减计数模式下，计数器从自动预装载值 TIMx_ARR 开始递减计数，计数到 0 时，产生一个下溢事件（更新事件）。当用户启用了重复计数功能后，重复计数器在每次下溢事件时递减计数，只有当重复计数器从设定值递减到 0 时，才会产生更新事件；设置 TIMx_EGR 寄存器的 UG=1，同样可以产生一个更新事件，更新事件后计数器从自动预装载值 TIMx_ARR 开始重新递减计数（TIMx_CR1 寄存器 UDIS=0）。

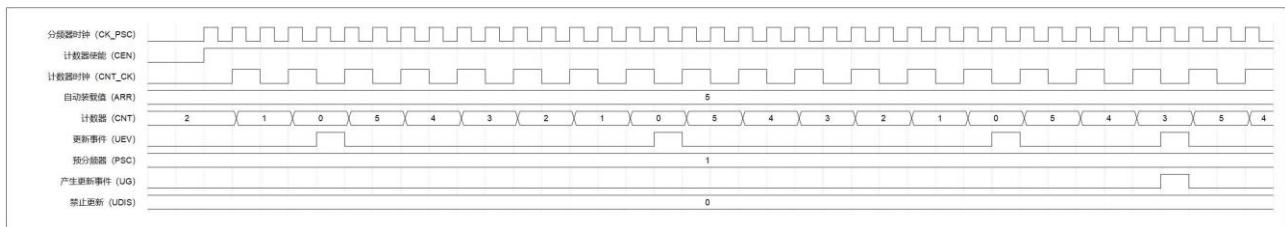


图 11-8 递减计数模式 (UDIS=0)

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

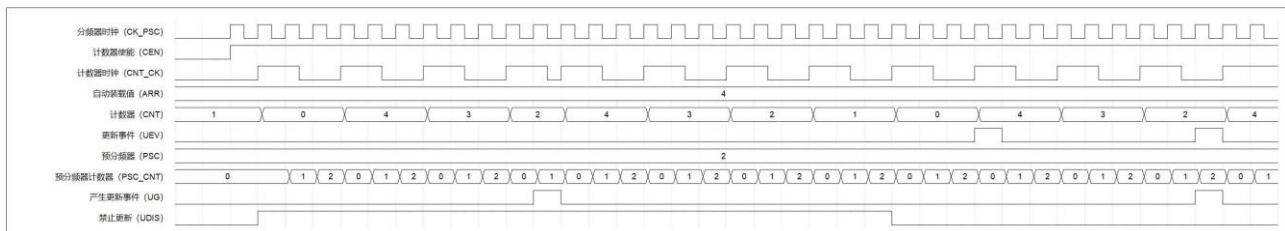


图 11-9 递减计数模式 (UDIS=1 禁止产生更新事件)

11.4.1.3.3 中央计数模式（递增/递减计数模式）

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

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

当用户启用了重复计数功能后，重复计数器在每次上溢事件或下溢事件时递减重复计数器值，只有当重复计数器从设定值递减到 **0** 时，才会产生更新事件；设置 **TIMx_EGR** 寄存器的 **UG=1**，同样可以产生一个更新事件，更新事件后计数器从 **0** 开始重新递增计数（**TIMx_CR1** 寄存器 **UDIS=0**）。

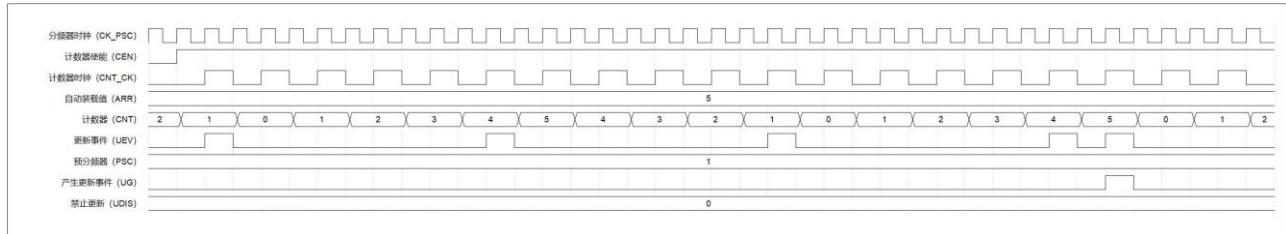


图 11-10 中央计数模式 (UDIS=0)

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

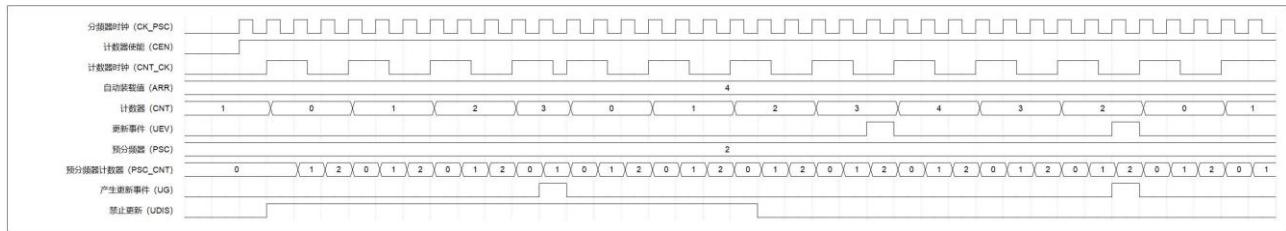


图 11-11 中央计数模式 (UDIS=1 禁止产生更新事件)

11.4.2 重复计数器

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

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

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

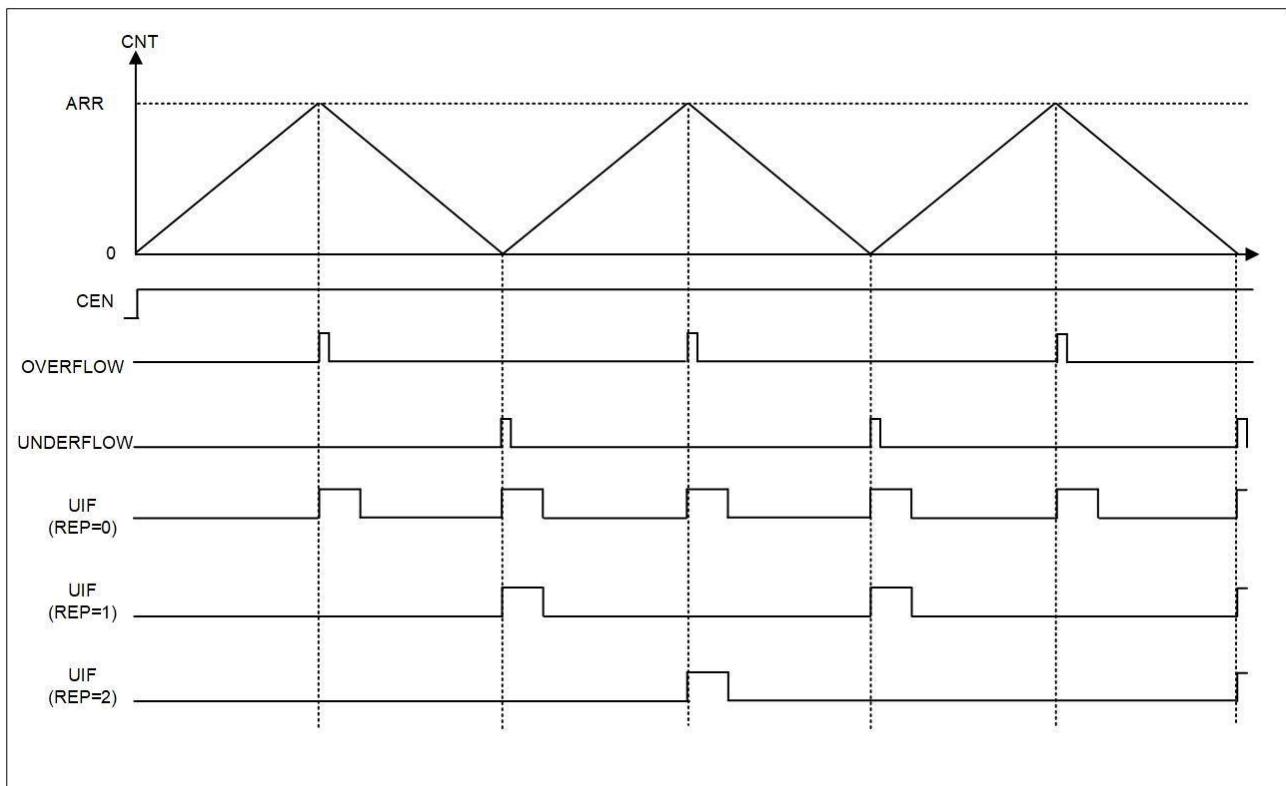


图 11-12 中央对齐模式重复计数时序图

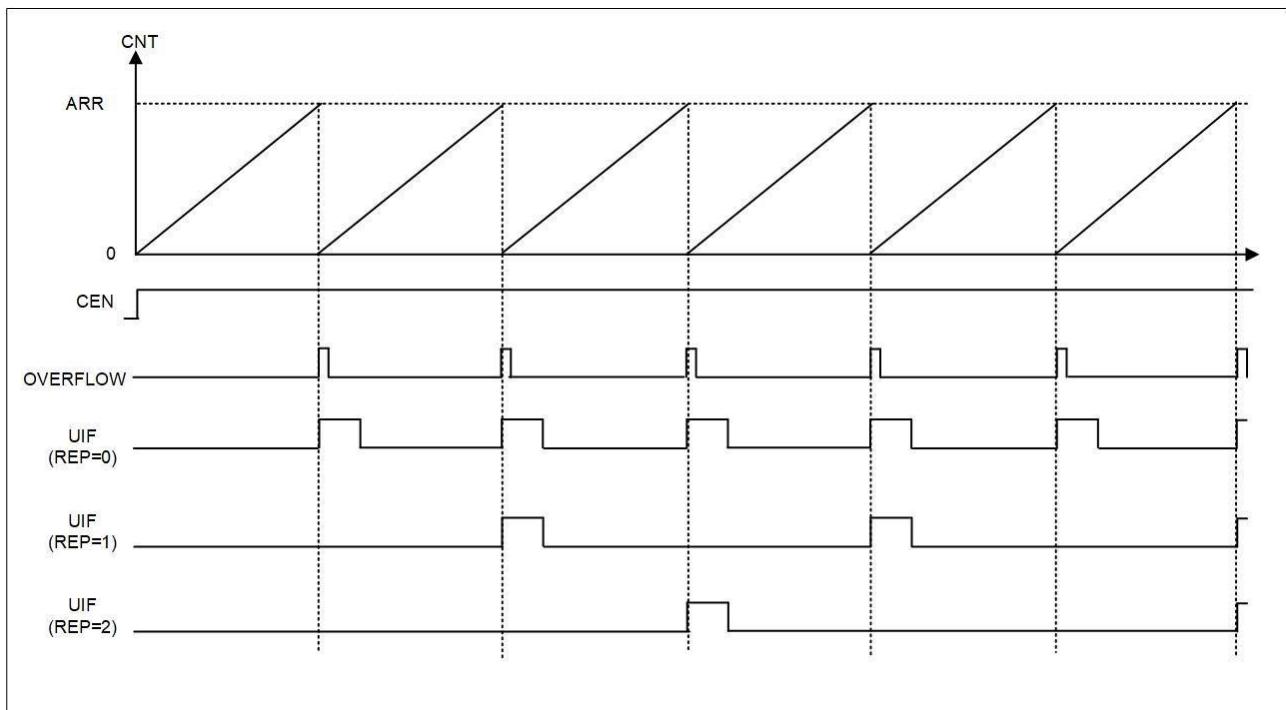


图 11-13 边沿对齐模式递增计数时序图

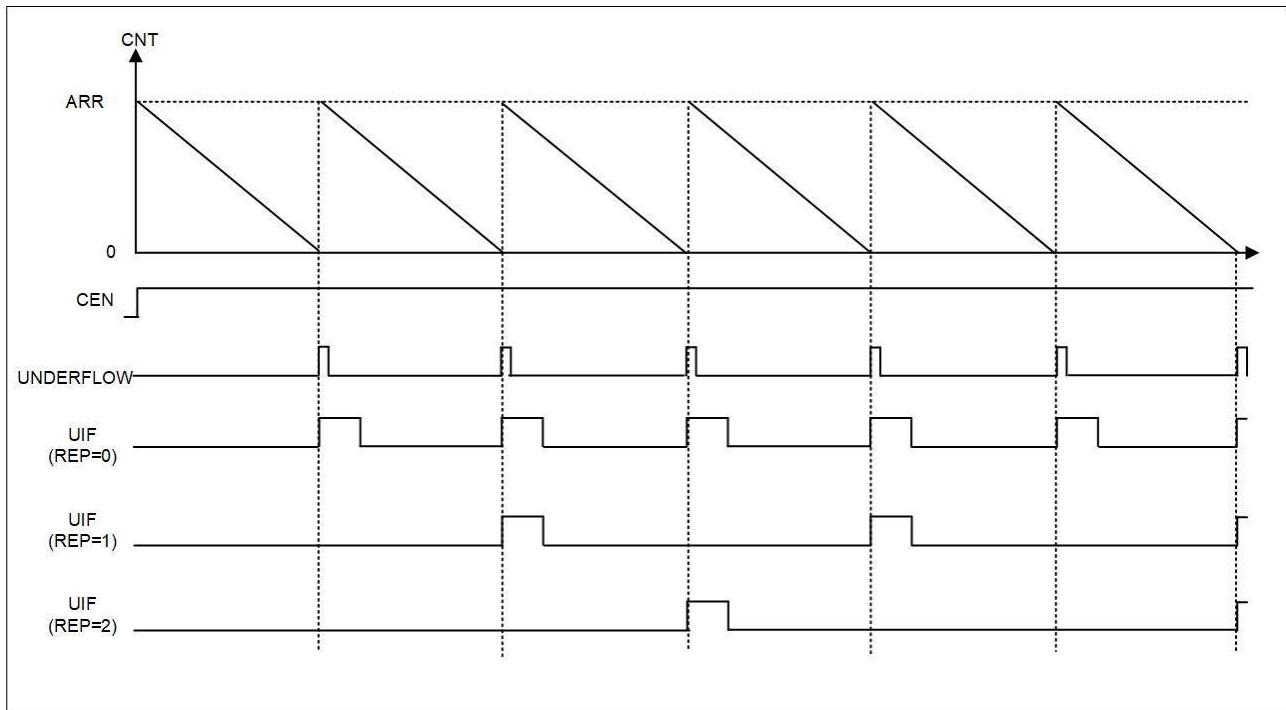


图 11-14 边沿对齐模式递减计数时序图

11.4.3 输入捕获

11.4.3.1 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：

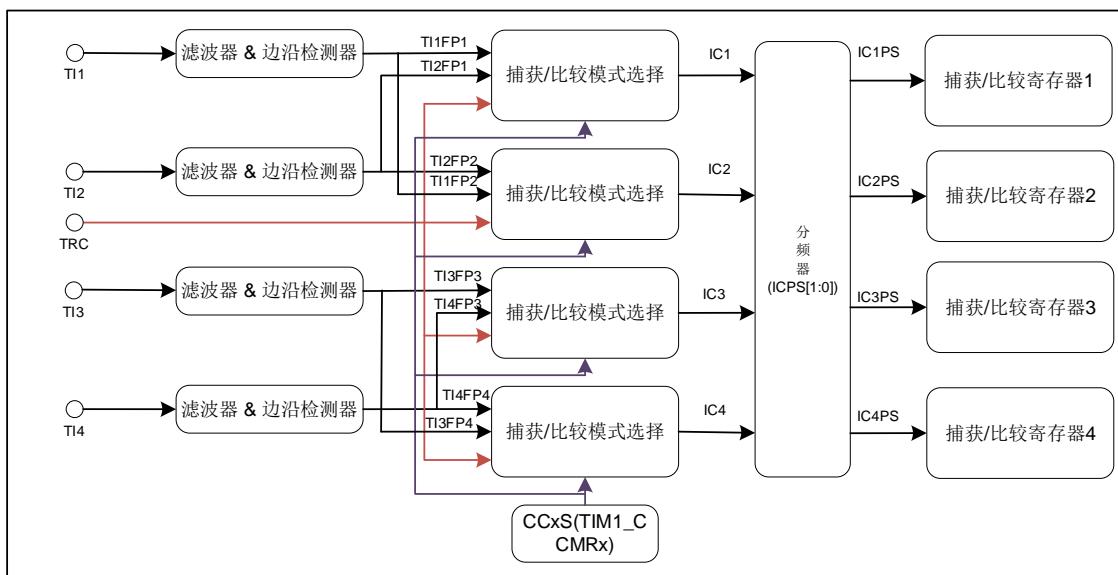


图 11-15 TIMx 输入捕获结构图

通过配置 **TIMx_CCMRx** 寄存器的 **ICxF**，可以设置数字滤波器的滤波宽度（滤波器的采样频率及数字滤波宽度如下表所示），当数字滤波器的输入信号宽度大于滤波宽度时，输入信号有效；数字滤波器对输入

引脚 TIx 的输入信号采样后，产生一个滤波后的信号 TIxF ，然后通过极性可选的边沿检测器，产生一个有效信号 TIxFPx ，这个信号可以作为从模式控制器的触发输入信号，同时该信号经过预分频器产生一个信号 ICxPS ，用于触发输入捕获事件。

表 11-1 数字滤波器宽度与 ICxF 的对应关系表

$\text{IC1F}[3: 0]$	采样频率和滤波宽度	$\text{IC1F}[3: 0]$	采样频率和滤波宽度
0000	无滤波器，以 f_{DTS} 采样	1000	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=6$
0001	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=2$	1001	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=8$
0010	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=4$	1010	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=5$
0011	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=8$	1011	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=6$
0100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/2, N=6$	1100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=8$
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 ICx 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断或 DMA 使能，发生捕获事件时，将产生相应的中断或 DMA 请求。发生捕获事件时，会将状态寄存器(TIMx_SR)中的捕获标志位 CCxIF 置 1，通过配置 $\text{CCxIF}=0$ 或读取 TIMx_CCRx 中的数据，清除 CCxIF 标志位。当 CCxIF 未被清零时，发生输入捕获事件，重复捕获标志位 CCxOF 将会被置 1，通过配置 $\text{CCxOF}=0$ ，可以清除 CCxOF 标志位。

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

1. 配置 TIMx_CCMR1 寄存器的 $\text{CC1S}=01$ ， CC1 通道被配置为输入， IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 $\text{IC1F}[3: 0]$ ，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 $\text{CC1P}=0$ ，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 $\text{IC1PSC}[1:0]$ ，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 $\text{CC1E} = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 TIMx_DIER 寄存器的 $\text{CC1IE}=1$ ，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 $\text{CC1DE}=1$ ，允许捕获/比较通道 1 的 DMA 请求。

注：

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

11.4.3.2 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 IC_x 信号被映射至同一个 TI_x 输入。

- 配置从模式为复位模式，将其中一路 TI_xFP 作为触发输入信号。

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

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

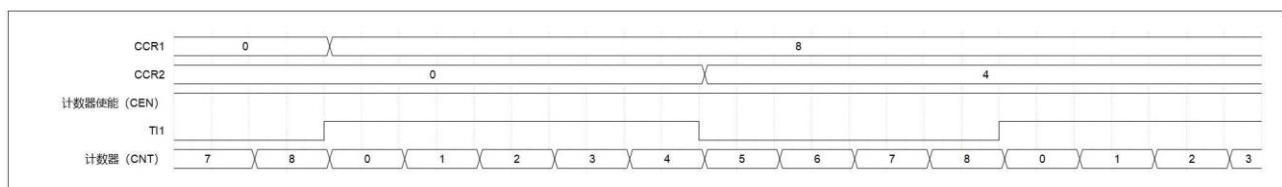


图 11-16 PWM 输入模式时序

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 TIMx_CH1/TIMx_CH2 端口输入信号。

11.4.4 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：

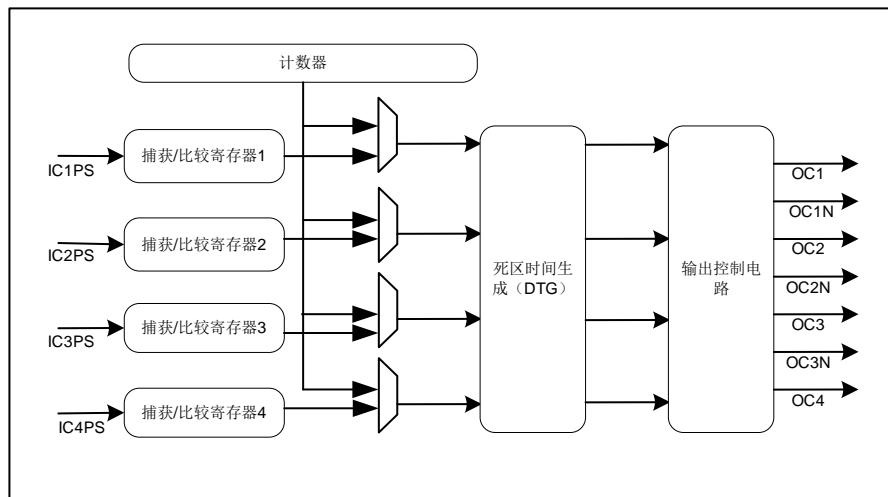


图 11-17 比较输出部分结构图

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。

11.4.4.1 强制输出

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

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

11.4.4.2 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 TIMx_CCMRx 寄存器的 OCxM 位的配置用来输出不同的波形。

例如，当计数器与捕获/比较寄存器的内容匹配时，比较输出模式下的操作如下：

1. 在比较匹配时， OCxM 的值不同，输出通道 x 信号 OCx 的操作不同：

- ◆ $\text{OCxM} = 000$: OCx 信号保持它的电平
- ◆ $\text{OCxM} = 001$: OCx 信号被设置成有效电平
- ◆ $\text{OCxM} = 010$: OCx 信号被设置成无效电平
- ◆ $\text{OCxM} = 011$: OCx 信号进行翻转

2. 匹配时状态寄存器中的标志位置 1 (TIMx_SR 寄存器中的 CCxIF 位)。

3. 当配置了 `TIMx_DIER` 寄存器中的 `CCxIE =1`, 匹配时则产生一个中断。
4. 当配置了 `TIMx_DIER` 寄存器中的 `CCxDE =1`, 匹配时则产生一个 DMA 请求 (仅适用于有内置 DMA 的产品)。

比较输出模式也可以用来输出一个单脉冲 (单脉冲输出模式)。

例如, 通道 1 的比较输出模式的配置步骤如下:

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

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

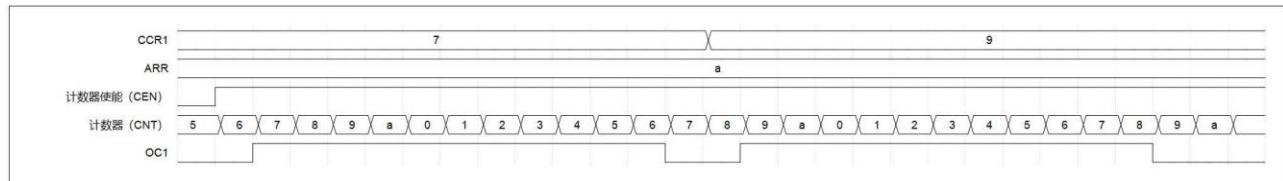


图 11-18 比较输出模式, `OC1` 信号在匹配时翻转

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

11.4.4.3 PWM 输出

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

配置与通道 x 对应的 `TIMx_CCMRx` 寄存器的 `OCxM=110` 或 `OCxM=111`, 选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下, 计数器和 `CCRx` 会一直进行比较, 根据配置和比较结果, 通道 x 输出不同的信号, 因此 `TIMx` 可以产生 4 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 `TIMx_CCRx` 的预装载功能和 `TIMx_ARR` 寄存器的预装载功能。写入 `TIMx_CCRx` 预装载寄存器和

TIMx_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 **TIMx_EGR** 的 **UG=1**，产生更新事件用于初始化所有的寄存器。

配置 **TIMx_CCER** 寄存器的 CCxP 选择 OCx 的有效极性。配置 **TIMx_CCER** 寄存器的 CCxE、CCxNE 位和 **TIMx_BDTR** 寄存器的 MOE、OSSI、OSSR 位控制 OCx 的输出使能。配置 **TIMx_CR1** 寄存器的 CMS 位，可以选择产生边沿对齐或中央对齐的 PWM 信号。

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

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

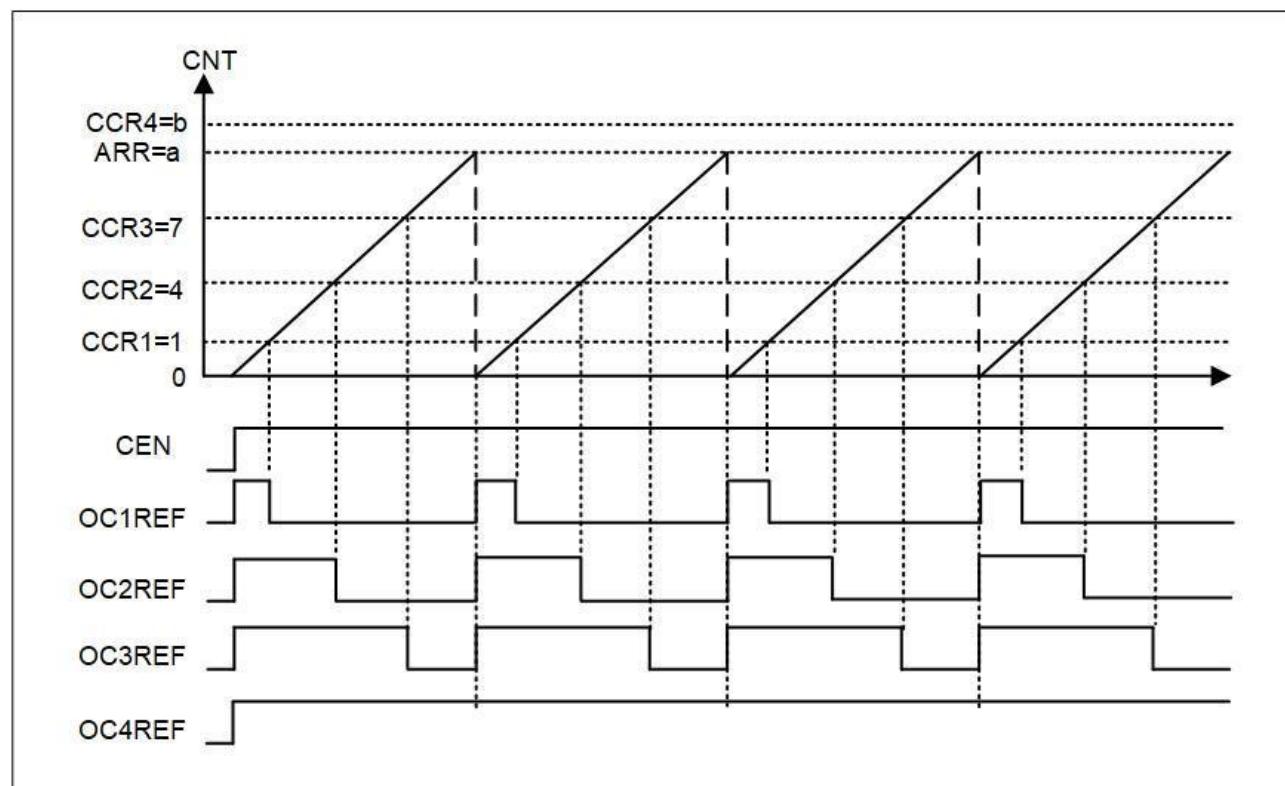


图 11-19 边沿对齐递增计数时 PWM 模式 1 的波形

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

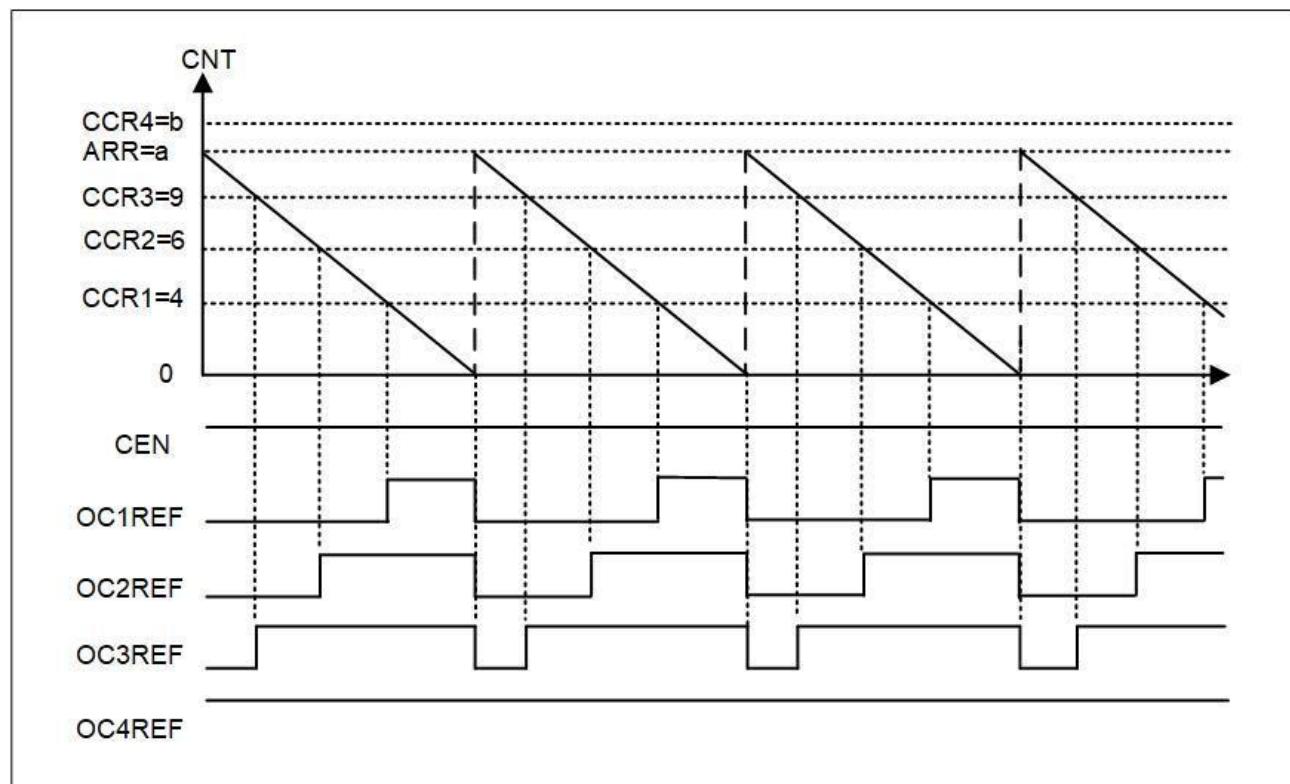


图 11-20 边沿对齐递减计数时 PWM 模式 1 的波形

11.4.4.3.3 PWM 中央对齐模式

首先配置 TIMx 计数器为中央对齐计数模式，配置 TIMx_CCMRx 寄存器的 $\text{CCxS}=00$ ，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 ($\text{CMS}=01$)、在计数器递增计数时被设置 ($\text{CMS}=10$)、或在计数器递增或递减计数时被设置 ($\text{CMS}=11$)。下图为 $\text{CCR1}=4$, $\text{CCR2}=6$, $\text{CCR3}=9$, $\text{CCR4}=b$, $\text{ARR}=a$ 时中央对齐 PWM 模式 1 的波形实例。

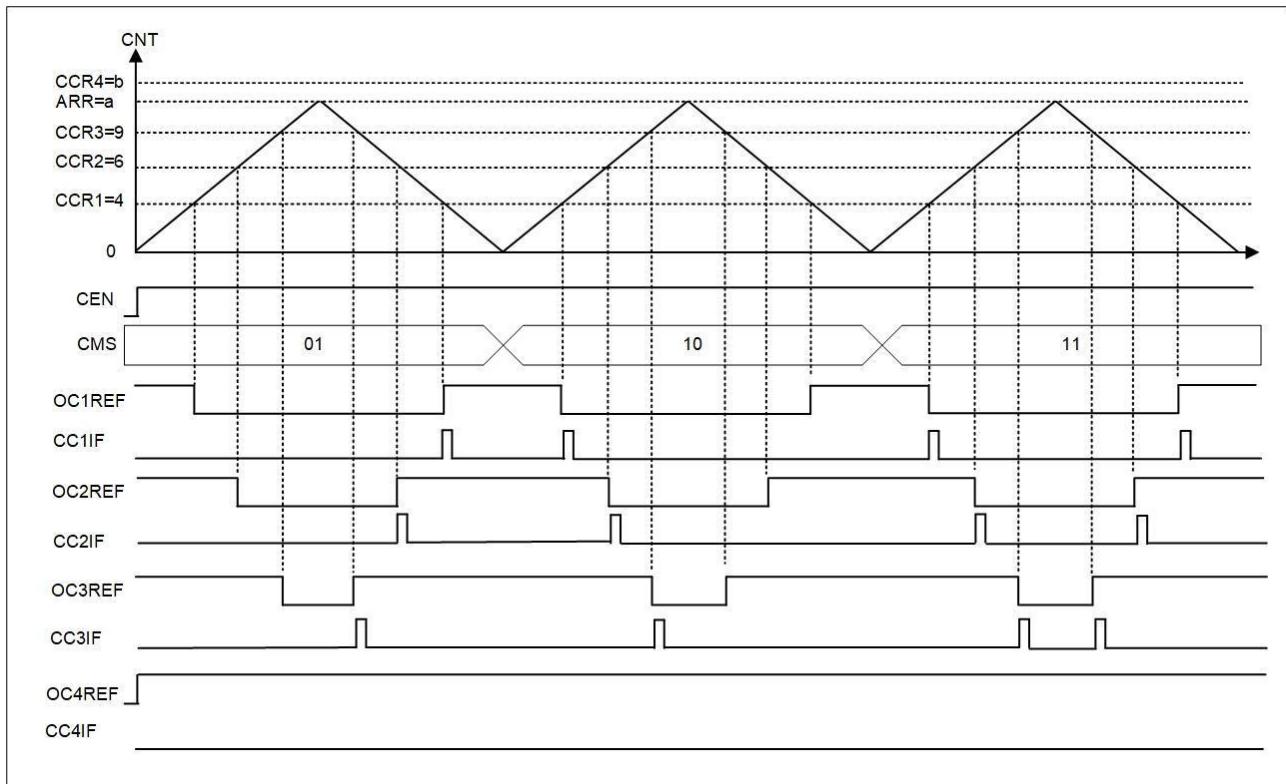


图 11-21 中央对齐 PWM 模式 1 的波形

11.4.4.3.4 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。

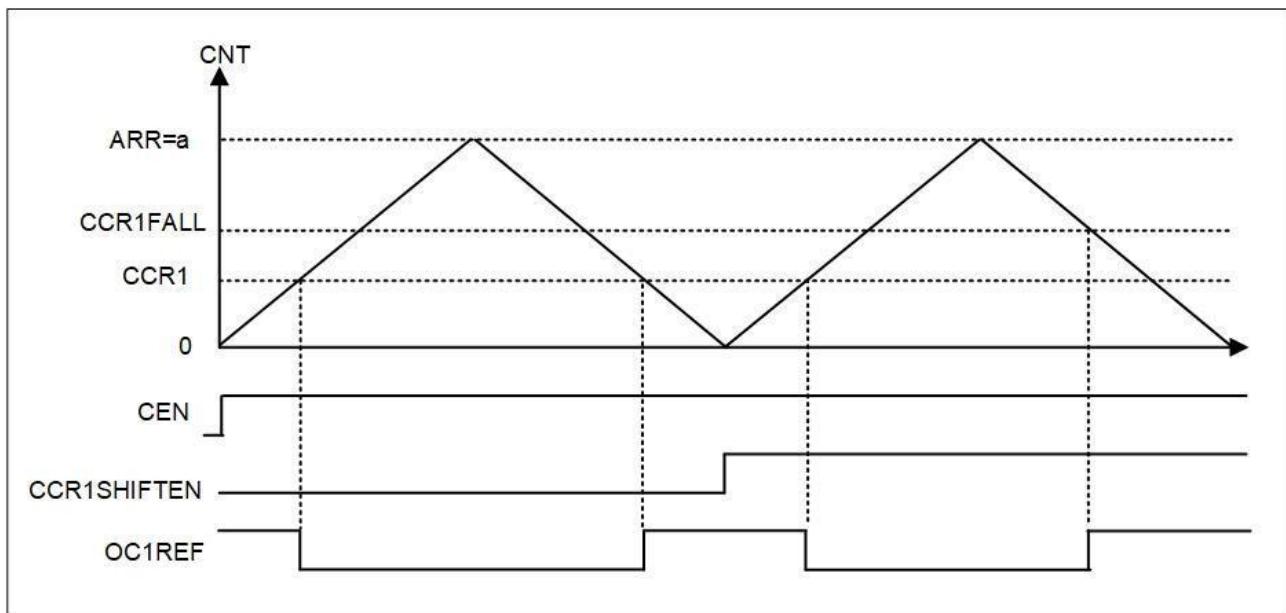


图 11-22 移相功能示意图

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 > TIMx_ARR，计数器会继续递增计数。直接写入 0 或 ARR，会立即更新计数方向，但不会产生更新事件。
- 建议使用中央对齐模式时，在启动计数器之前配置 TIMx_EGR 寄存器的 UG=1，产生一个软件更新，更新所有寄存器，启动计数器后不要修改计数器的值。

11.4.4.3.5 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 TIMx_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 TIMx_DIER 寄存器 COMIE=1 和 COMDE = 1，发生 COM 事件会产生一个 COM 中断和一个 DMA 请求（产生 DMA 请求仅适用于有内置 DMA 的产品）。

下图显示当发生 COM 事件时，不同配置下 OCx 和 OCxN 输出。

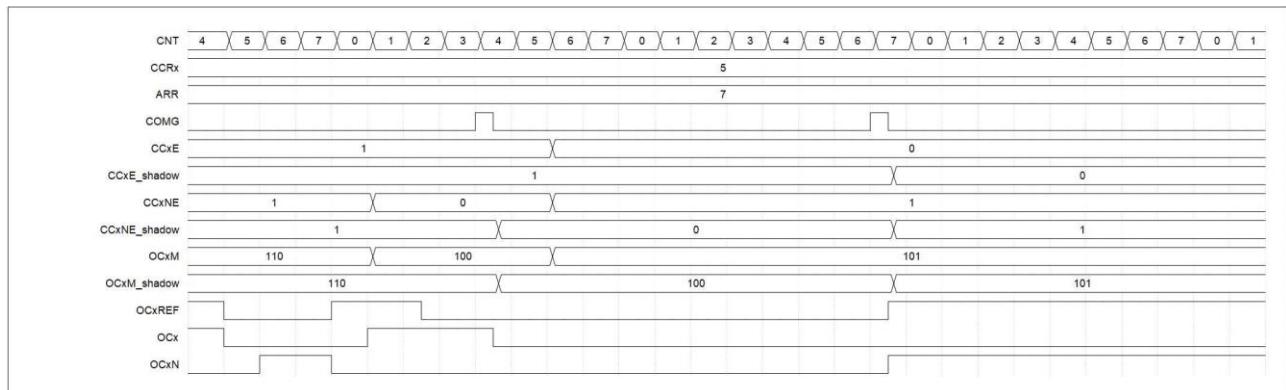


图 11-23 使用 COM 事件产生六步 PWM (OSSR = 1)

11.4.4.4 互补输出和死区插入

OCx 和 OCxN 是一对互补输出通道，TIMx 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。

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 11-2 死区时间计算

DTG[7: 5]	DT
0xx	$DT = DTG[7: 0] \times T_{dtg} (T_{dtg} = T_{DTS})$
10x	$DT = (64 + DTG[5: 0]) \times T_{dtg} (T_{dtg} = 2 \times T_{DTS})$

DTG[7: 5]	DT
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 TIMx_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 TIMx_CCER 寄存器的 $CCxE$ 和 $CCxNE$ 位, TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE 、 $OISx$ 、 $OISxN$ 、 $OSSI$ 和 $OSSR$ 位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。

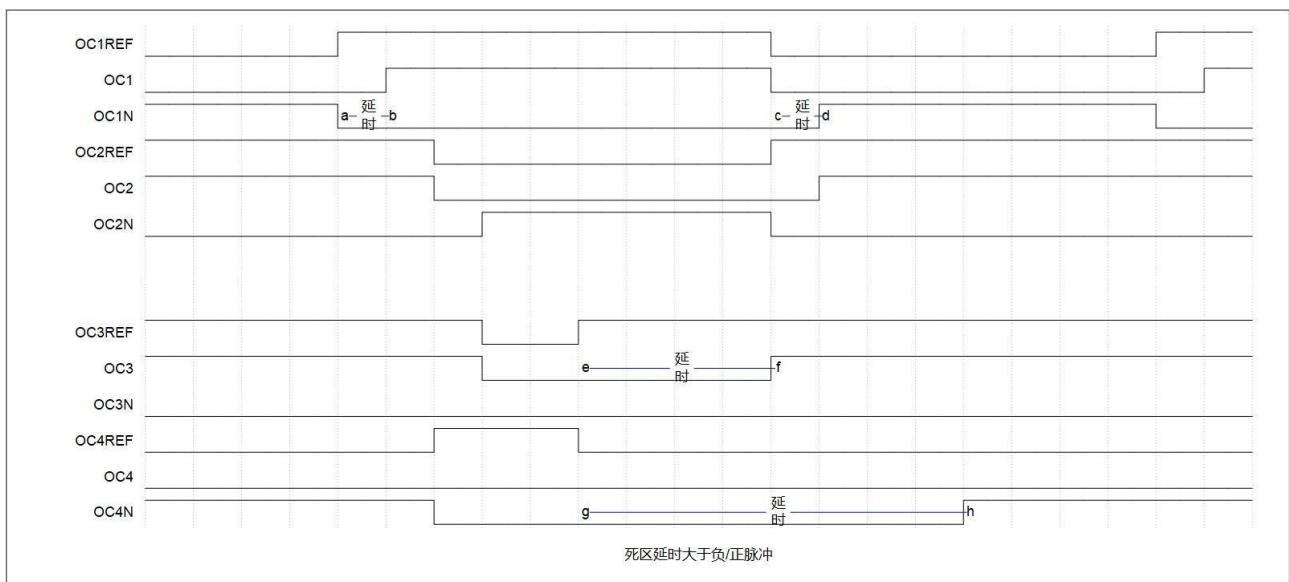


图 11-24 死区插入

11.4.4.5 刹车功能

TIMx 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: TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位, TIMx_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所示。

表 11-3 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 11-4 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 11-5 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 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	

CCxE	CCxNE	OCx	OCxN
		OISx 和 OISxN 都不对应 OCx 和 OCxN 的有效电平	

表 11-6 当 MOE=0, OSSI=1, OSSR=0/1 时:

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

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

系统复位后, MOE=0, 刹车功能禁止, 需要配置 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 OSSI 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 TIMx_CR2 寄存器的 OISx 位决定, OSSI=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 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, OCx/OCxN 极性和被禁止

的状态，OCxM 配置，刹车使能和极性等参数。通过 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：

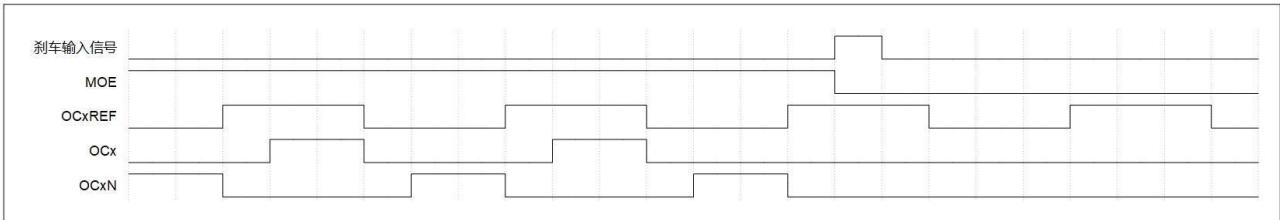


图 11-25 响应刹车的输出 ($OISx=0, OISxN=0$)

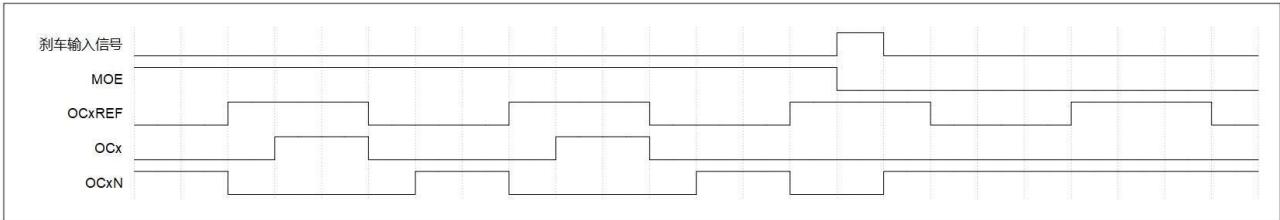


图 11-26 响应刹车的输出 ($OISx=0, OISxN=1$)

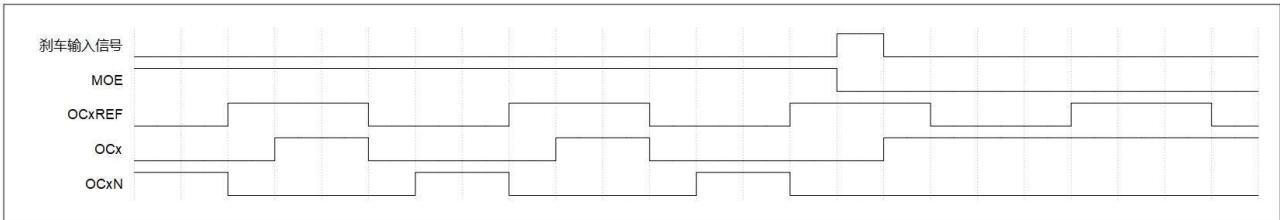


图 11-27 响应刹车的输出 ($OISx=1, OISxN=0$)

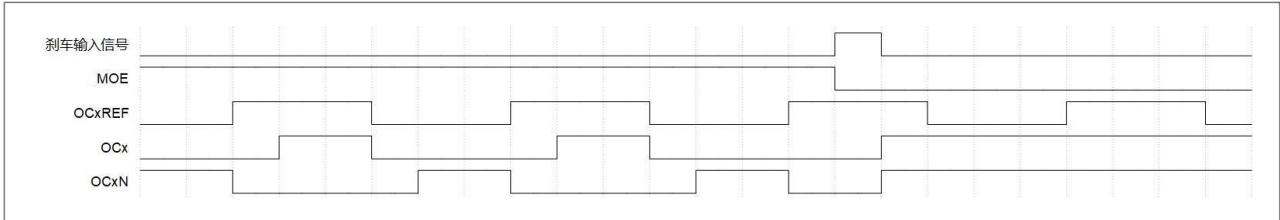


图 11-28 响应刹车的输出 ($OISx=1, OISxN=1$)

11.4.4.6 外部事件清除 OCxREF

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

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

1. 配置 TIMx_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
2. 配置 TIMx_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
3. 配置 TIMx_SMCR 寄存器 ETF[3: 0]和 ETP，配置 ETR 信号的触发极性和滤波宽度。

下图显示了当 ETR 输入变化触发 ETRF 为高时，对应不同 OCxCE 的值，OCxREF 信号的动作 (PWM)

模式)。

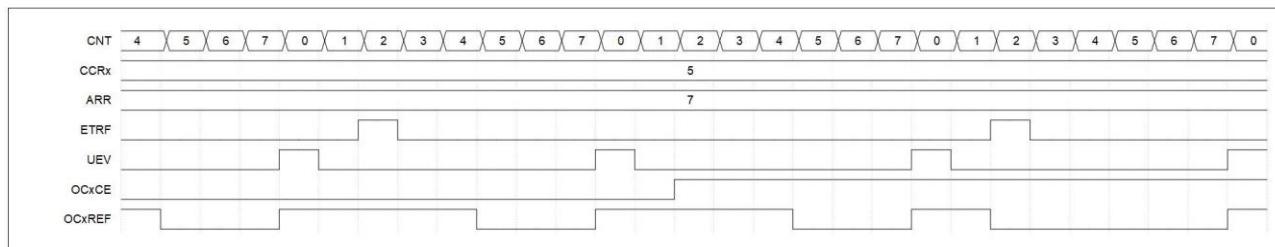


图 11-29 外部事件清除 OCxREF

11.4.4.7 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 **TIMx_CR1** 寄存器的 **OPM=1**，选择单脉冲模式，触发信号有效沿或配置 **CEN=1** 都可以启动计数器，直到下一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 ≤ ARR**。
- 递减计数方式：计数器 **CNT > CCRx**。

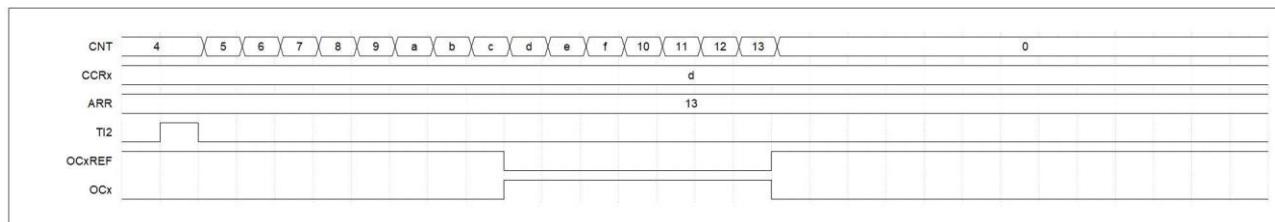


图 11-30 单脉冲模式

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。

假定 **TI2FP2** 作为触发源：

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

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

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

1. 配置 **TIMx_CCMR1** 寄存器 **OC1M = 111**，选择 **PWM** 模式 2。
2. 配置 **TIMx_CCER** 寄存器 **CC1P = 1**，输出低电平有效。
3. 配置 **TIMx_CCMR1** 中 **OC1PE = 1** 和 **TIMx_CR1** 寄存器中 **ARPE=1**，使能预装载寄存器。

4. 配置 `TIMx_CCR1` 寄存器和 `TIMx_ARR` 寄存器。
5. 配置 `TIMx_EGR` 寄存器 `UG=1` 产生一个更新事件。
6. 等待在 `TI2` 上的一个外部触发事件。

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

11.4.4.7.1 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 `TIMx_CCMR` 寄存器的 `OCxFE=1`, 强制 `OCxREF` 直接响应激励而不是依赖计数器和比较值之间的比较结果, 输出波形和比较匹配时的波形一样。这样可以去除比较的时间, 快速输出比较结果。`OCx` 快速输出使能只在 `PWM` 模式下生效。

11.4.5 从模式

11.4.5.1 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 `TIMx_SMCR` 寄存器 `SMS` 位可以选择输入源, 根据输入源的不同, 可以将编码器接口模式分为 3 种模式, `SMS=001`, 编码器接口模式 1; `SMS=010`, 编码器接口模式 2; `SMS=011`, 编码器接口模式 3; 三种模式具体计数操作如下表所示。两个输入 `TI1` 和 `TI2` 被用来作为正交编码器的接口。

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

注: 编码器模式不支持外部时钟模式 2。

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

表 11-7 计数方向与编码器信号的关系

计数模式	相对电平 (<code>TI1FP1</code> 相对于 <code>TI2</code> , <code>TI2FP2</code> 相对于 <code>TI1</code>)	<code>TI1FP1</code> 信号		<code>TI2FP2</code> 信号	
计数模式	相对电平 (<code>TI1FP1</code> 相对于 <code>TI2</code> , <code>TI2FP2</code> 相对于 <code>TI1</code>)	上升	下降	上升	下降

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

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

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

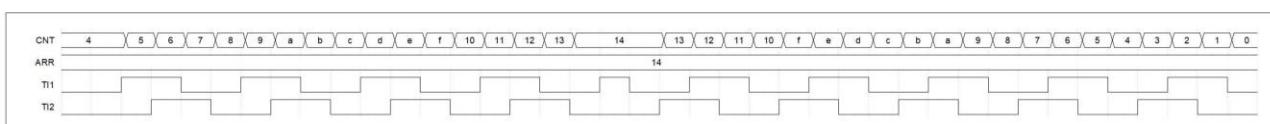


图 11-31 编码器模式下的计数器时序图

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

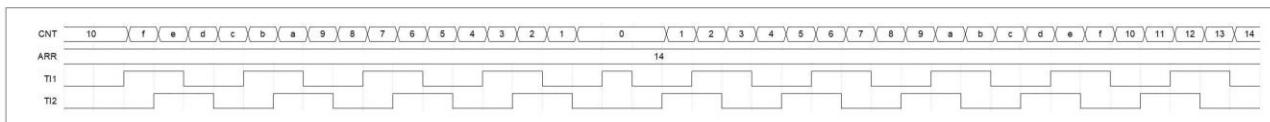


图 11-32 IC1FP1 反相编码器接口模式时序图

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。

11.4.5.2 复位模式

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

例如，`TI2` 输入端的下降沿触发计数器重启：

1. 配置 `TIMx_CCMR1` 寄存器的 `CC2S=01`，`CC2` 通道被配置为输入模式；`IC2` 映射在 `TI2` 上，配置 `TIMx_CCER` 寄存器的 `CC2P=1`，检测下降沿。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS = 100`，从模式选择复位模式；配置 `TIMx_SMCR` 寄存器的 `TS = 110`，选择滤波后的定时器输入 2 (`TI2FP2`) 作为同步计数器的触发输入；配置 `TIMx_SMCR` 寄存器的 `ETF=0000`，不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`，选择计数方向为递增计数；配置 `PSC=0`，不分频；配置 `CEN=1`，使能计数器。

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

下图为复位模式下 `TIMx_ARR = 0x13` 的时序图。

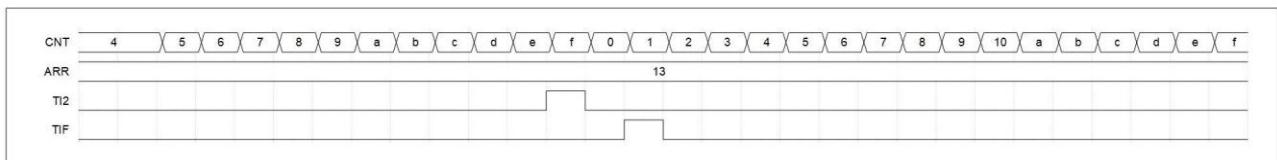


图 11-33 复位模式的控制时序图

11.4.5.3 门控模式

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

例如，计数器只在 `TI1` 为高时计数：

1. 配置 `TIMx_CCMR1` 寄存器的 `CC1S=01`，`CC1` 通道被配置为输入模式，`IC1` 映射在 `TI1` 上；配置 `TIMx_CCER` 寄存器的 `CC1P=0`，检测 `TI1` 上的高电平。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS=101`，从模式选择为门控模式；配置 `TIMx_SMCR` 寄存器的 `TS=101`，选择滤波后的定时器输入 1 (`TI1FP1`) 作为同步计数器的触发输入；配置 `TIMx_SMCR` 寄存器的 `ETF=0000`，不使用数字滤波器。

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

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

下图为门控模式下 `TIMx_ARR=0xf` 的时序图。

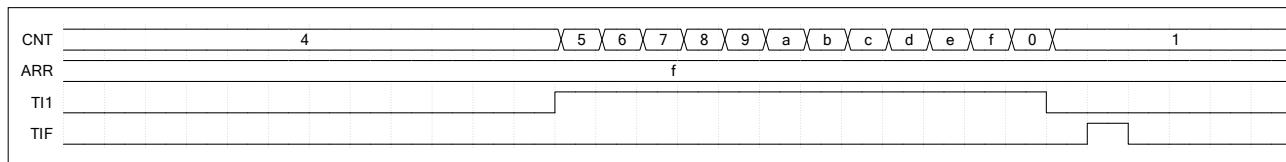


图 11-34 门控模式下的控制时序图

11.4.5.4 触发模式

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

例如, 计数器在 `TI1` 输入的上升沿开始计数:

1. 配置 `TIMx_CCMR1` 寄存器的 `CC1S=01`, `CC1` 通道被配置为输入模式, `IC1` 映射在 `TI1` 上, 配置 `TIMx_CCER` 寄存器的 `CC1P=0`, 检测上升沿。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS = 110`, 从模式选择为触发模式; 配置 `TIMx_SMCR` 寄存器的 `TS=101`, 选择滤波后的定时器输入 1 (`TI1FP1`) 作为计数器的触发输入; 配置 `TIMx_SMCR` 寄存器的 `ETF=0000`, 不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`, 选择计数方向为递增计数; 配置 `PSC=0`, 不分频。

计数器的时钟源由内部时钟提供, 当检测到 `TI1` 的上升沿, 计数器开始计数。

下图为触发模式下 `TIMx_ARR=0xf` 的时序图。

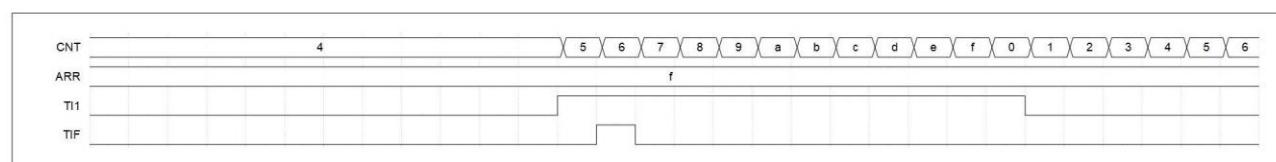


图 11-35 触发器模式下的控制时序图

11.4.5.5 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 `TIMx_SMCR` 寄存器的 `ETF = 0000`, 不使用数字滤波器; 配置 `TIMx_SMCR` 寄存器的 `ETPS = 00`, 关闭预分频; 配置 `TIMx_SMCR` 寄存器的 `ETP = 0`, 检测 `ETR` 的上升沿; 配置 `TIMx_SMCR` 寄存器的 `SMIE=1`, 使能中断。

器的 ECE = 1，使能外部时钟模式 2。

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

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

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

计数器在 TI1 的上升沿开始计数，并将 TIF 置 1。ETR 信号的上升沿和计数器实际计数时钟间的延时取决于 ETR 输入端的同步电路设计。

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

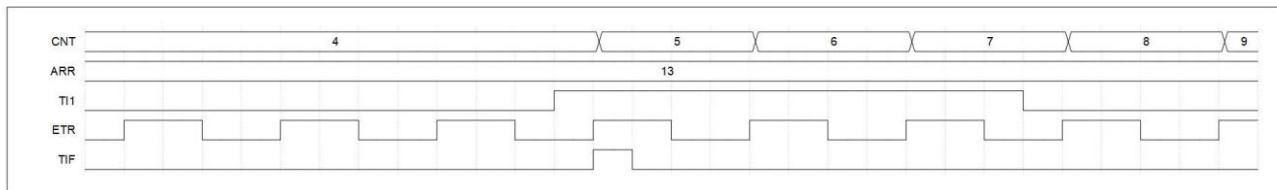


图 11-36 外部时钟模式 2+从模式（触发模式）控制时序图

11.4.6 定时器同步

不同的 TIMx 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系统如下：

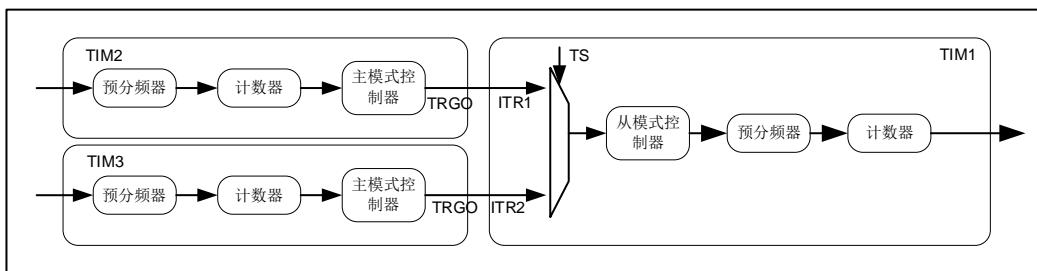


图 11-37 定时器间互联

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。

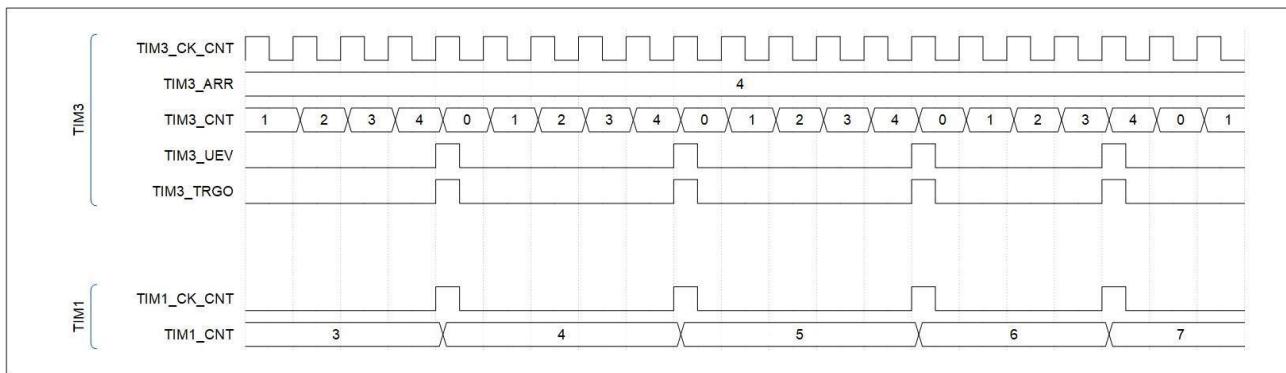


图 11-38 使用 TIM3 作为 TIM1 的预分频器

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，TIMx_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。

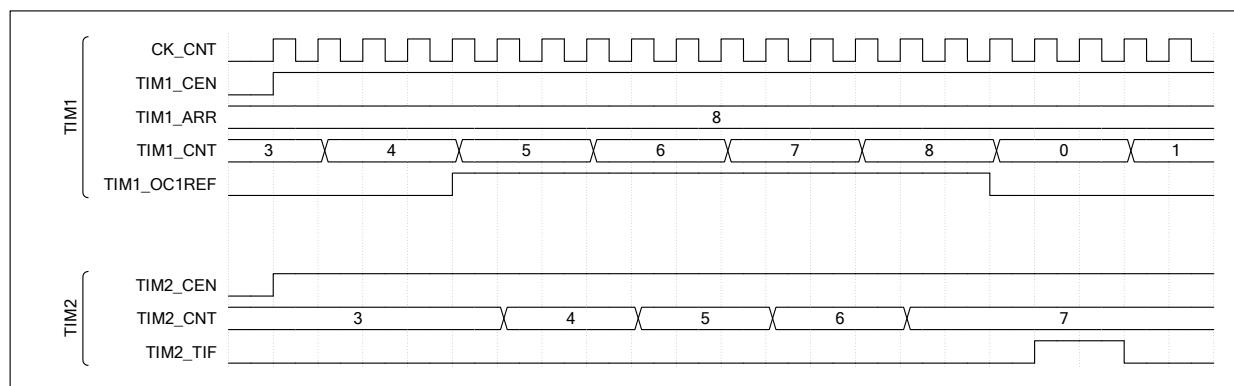


图 11-39 使用 TIM1 使能 TIM2

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。

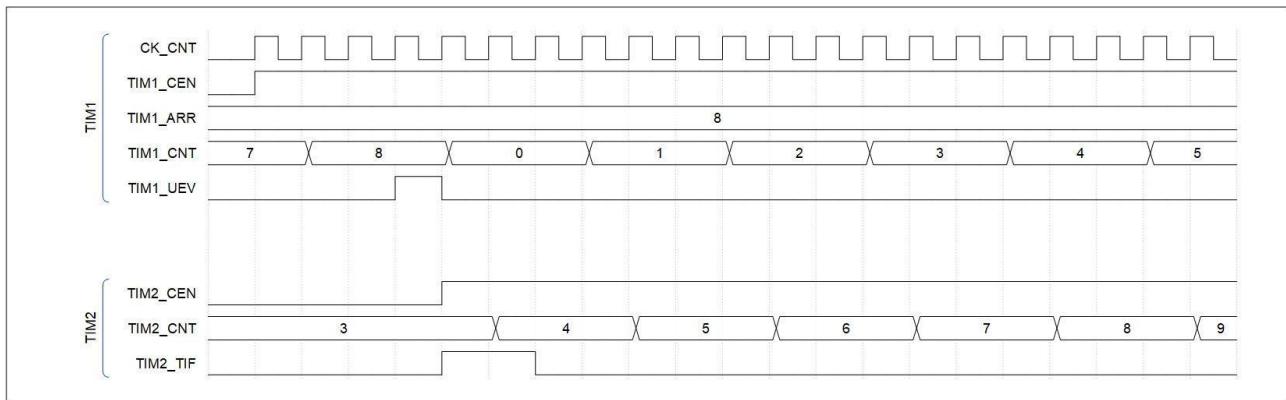


图 11-40 使用 TIM1 的更新事件启动 TIM2

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。

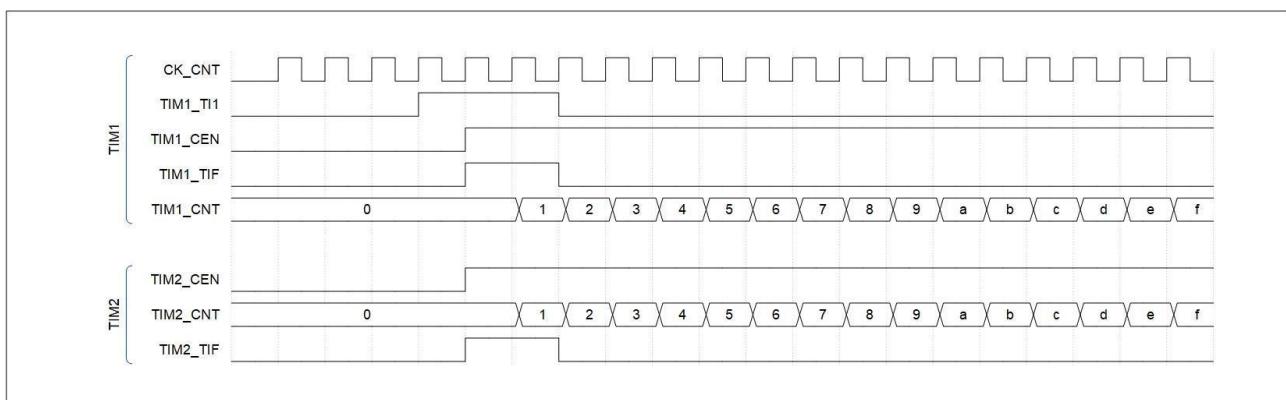


图 11-41 TIM1 的 TI1 同步启动 TIM1 和 TIM2

11.4.7 定时器异或

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

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

1. 配置 `TIMx_CR2` 寄存器的 `TI1S=1`, 配置定时器的三个输入经异或后连接到 `TI1` 输入通道。
2. 配置 `TIMx_CCMR1` 寄存器的 `CC1S=01`, `CC1` 通道被配置为输入, `IC1` 映射在 `TI1` 上。
3. 配置 `TIMx_CCMR1` 寄存器的 `IC1F[3: 0]`, 配置数字滤波器的滤波宽度 (按需配置)。
4. 配置 `TIMx_CCER` 寄存器的 `CC1P=0`, 选择捕获发生在 `TI1` 信号的上升沿。
5. 配置 `TIMx_CCMR1` 寄存器的 `IC1PSC`, 选择预分频系数。
6. 配置 `TIMx_CCER` 寄存器的 `CC1E = 1`, 开启输入/捕获通道 1 的捕获使能。
7. 配置 `TIMx_CR1` 寄存器的 `CEN=1`, 启动计数器。

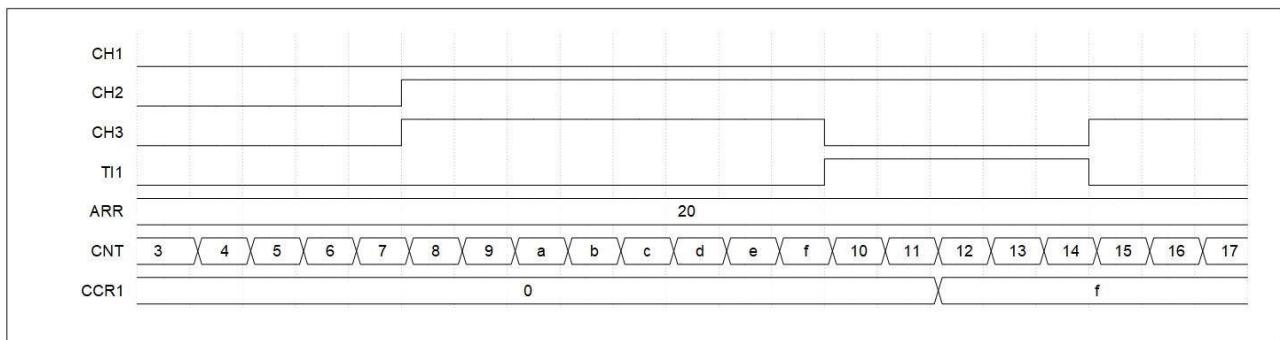


图 11-42 (TI1 异或输入) 输入捕获波形图

11.4.7.1.1 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 `TIMx` 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 `TIMx_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 `TIMx`。

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 `TIMx` 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。

TIMx 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（TIMx_CR2 寄存器中 CCPC = 1），同时触发输入用于控制 COM 事件（TIMx_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：

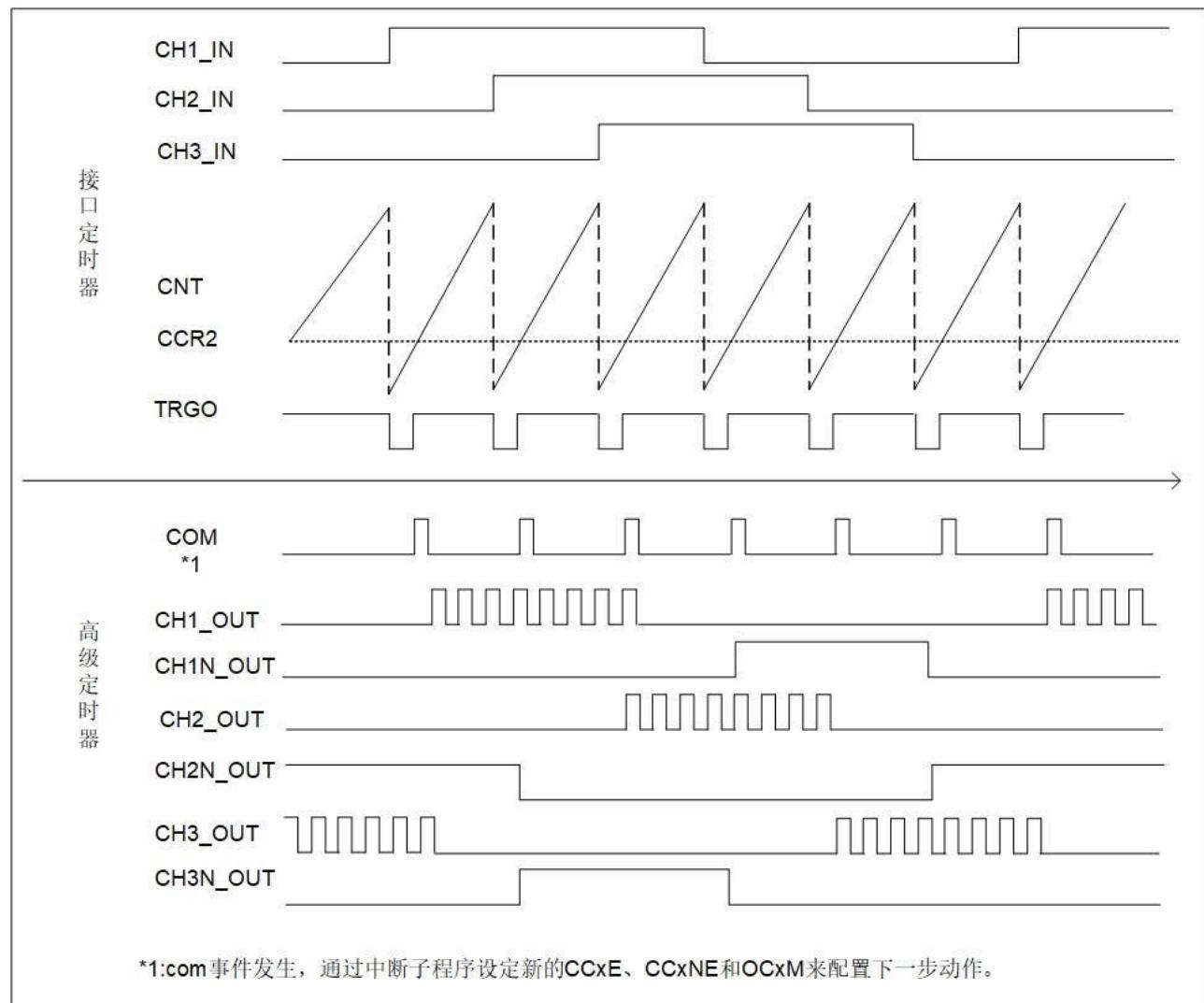


图 11-43 霍尔传感器接口实例

11.4.8 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）

11.4.9 中断

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

表 11-8 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

11.4.10 DMA

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

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

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

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

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

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

11.5 寄存器描述

表 11-9 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x04	TIMx_CR2	控制寄存器 2	0x0000
0x08	TIMx_SMCR	从模式控制寄存器	0x0000
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 0000
0x10	TIMx_SR	状态寄存器	0x0000 0000
0x14	TIMx_EGR	事件产生寄存器	0x0000 0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIMx_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x30	TIMx_RCR	重复计数寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIMx_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIMx_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIMx_CCR4	捕获/比较寄存器 4	0x0000
0x44	TIMx_BDTR	刹车和死区寄存器	0x0000 0000
0x48	TIMx_DCR	DMA 控制寄存器 (仅适用于有内置 DMA 的芯片)	0x0000
0x4C	TIMx_DMAR	连续模式的 DMA 地址 (仅适用于有内置 DMA 的芯片)	0x0000
0x54	TIMx_CCMR3	捕获/比较模式寄存器 3	0x0000
0x58	TIMx_CCR5	捕获/比较寄存器 5	0x0000
0x5C	TIMx_PDER	PWM 移相/DMA repeat 更新请求使能寄存器	0x0000
0x60 ~ 0x70	TIMx_CCRxFALL	PWM 移相递减计数捕获/比较寄存器	0x0000

11.5.1 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	APRE	CMS	DIR	OPM	URS	UDIS	CEN		
						rw	rw	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, TIx) 所用的时钟之间的分频比例。</p> <p>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</p>
7	APRE	<p>自动重装载预装载使能 (Auto-reload preload enable)</p> <p>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</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 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)</p> <p>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)</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 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	<p>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。</p>
0	CEN	<p>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。</p>

11.5.2 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.	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	<p>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。</p>
8	OIS1	<p>输出空闲状态 1 (OC1 输出) (Output Idle state 1) 0: 当 MOE=0, OC1N=1 时则在死区时间后 OC1 = 0 1: 当 MOE=0, OC1N=1 时则在死区时间后 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。</p>

Bit	Field	Description
7	TI1S	<p>TI1 选择 (TI1 selection)</p> <p>0: TIMx_CH1 管脚连到 TI1 输入 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入</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> <ul style="list-style-type: none"> 000: 复位 TIMx_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001: 使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO), 计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新 更新事件被选为 TRGO。 011: 捕获/比较脉冲 发生一次捕获或一次比较成功时, 触发输出送出一个 TRGO 信号。 100: 比较 OC1REF 信号被用于作为触发输出 (TRGO) 101: 比较 OC2REF 信号被用于作为触发输出 (TRGO) 110: 比较 OC3REF 信号被用于作为触发输出 (TRGO) 111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)
3	CCDS	<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 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。 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 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>

11.5.3 TIMx_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							
----	----	----	----	----	----	----	----

Bit	Field	Description
15	ETP	<p>外部触发极性 (External trigger polarity) 该位选择 ETR 信号的极性。 0: 高电平或上升沿有效 1: 低电平或下降沿有效 注: 仅适用于支持外部触发的产品</p>
14	ECE	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2。 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2, ETRF 信号上的任意有效沿驱动计数器计数 注 1: 仅适用于支持外部触发的产品。 注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。 注 3: TS ≠ 111 时, 复位模式, 门控模式和触发模式可以与外部时钟模式 2 同时使用。 注 4: 同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入是 ETR。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须低于 TIMx 主时钟 PCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8 注: 仅适用于支持外部触发的产品。</p>

Bit	Field	Description
11: 8	ETF	<p>外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。</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> <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) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟，以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步，该功能可以把几个定时器同步到一个单一的外部事件。</p>
6: 4	TS	<p>触发选择 (Trigger selection) 触发输入源选择。 000: 内部触发 0 (ITR0) 001: 内部触发 1 (ITR1) 010: 内部触发 2 (ITR2) 011: 内部触发 3 (ITR3) 100: TI1 的边沿检测器 (TI1F_ED) 101: 滤波后的定时器输入 1 (TI1FP1) 110: 滤波后的定时器输入 2 (TI2FP2) 111: 外部触发输入 (ETR) 更多有关 ITRx 的细节，参见下表。 注：从模式使能后这些位不能修改。</p>
3	OCCS	<p>比较器输出信号清除选择 (Output compare clear selection) 在 PWM 模式下，清除比较器输出 0: 外部触发信号作为清除信号 1: 比较器输出作为清除信号 注：仅适用于支持外部触发或有内置比较器的产品。</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 11-10 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	TIM5	TIM2	TIM3	TIM4
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM1	TIM3	TIM4	TIM8
TIM8	TIM1	TIM2	TIM4	TIM5

11.5.4 TIMx_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
														rw	rw
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
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 18	Reserved	保留，必须保持复位值。
17	CC5DE	<p>允许比较 5 的 DMA 请求 (Compare 5 DMA request enable)</p> <p>0: 禁止比较 5 的 DMA 请求</p> <p>1: 允许比较 5 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 且有通道 5 的产品。</p>
16	CC5IE	<p>允许比较 5 中断 (Compare 5 interrupt enable)</p> <p>0: 禁止比较 5 中断</p> <p>1: 允许比较 5 中断</p>
15	Reserved	保留，必须保持复位值。
14	TDE	<p>允许触发 DMA 请求 (Trigger DMA request enable)</p> <p>0: 禁止触发 DMA 请求</p> <p>1: 允许触发 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>
13	COMDE	<p>允许 COM 的 DMA 请求 (COM DMA request enable)</p> <p>0: 禁止 COM 的 DMA 请求</p> <p>1: 允许 COM 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>
12	CC4DE	<p>允许捕获/比较 4 的 DMA 请求 (Capture/Compare 4 DMA request enable)</p> <p>0: 禁止捕获/比较 4 的 DMA 请求</p> <p>1: 允许捕获/比较 4 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>
11	CC3DE	<p>允许捕获/比较 3 的 DMA 请求 (Capture/Compare 3 DMA request enable)</p> <p>0: 禁止捕获/比较 3 的 DMA 请求</p> <p>1: 允许捕获/比较 3 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>
10	CC2DE	<p>允许捕获/比较 2 的 DMA 请求 (Capture/Compare 2 DMA request enable)</p> <p>0: 禁止捕获/比较 2 的 DMA 请求</p> <p>1: 允许捕获/比较 2 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>
9	CC1DE	<p>允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable)</p> <p>0: 禁止捕获/比较 1 的 DMA 请求</p> <p>1: 允许捕获/比较 1 的 DMA 请求</p> <p>注: 仅适用于有内置 DMA 的产品。</p>

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	BIE	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断 1: 允许刹车中断
6	TIE	允许触发中断 (Trigger interrupt enable) 0: 禁止触发中断 1: 允许触发中断
5	COMIE	允许 COM 中断 (COM interrupt enable) 0: 禁止 COM 中断 1: 允许 COM 中断
4	CC4IE	允许捕获/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

11.5.5 TIMx_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。
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) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1 (在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。</p> <p>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。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1</p>
0	UIF	<p>更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。</p> <p>0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。</p>

11.5.6 TIMx_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														UG	
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	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, TIMx_SR 寄存器的 TIF = 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA, 由硬件自动清 0。
5	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。

11.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址: 0x18

复位值: 0x0000

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

OC2CE	OC2M	OC2PE	OC2FE	CC2S	OC1CE	OC1M	OC1PE	OC1FE	CC1S
IC2F		IC2PSC		CC2S		IC1F		IC1PSC	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CCxS 定义。该寄存器 CCxS 以外其它位的作用在输入模式和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能 (Output compare 2 clear enable) 参考 OC1CE 的描述。
14: 12	OC2M	通道 2 比较输出模式 (Output compare 2 mode) 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能 (Output compare 2 preload enable) 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能 (Output compare 2 fast enable) 参考 OC1FE 的描述。
9: 8	CC2S	通道 2 捕获/比较选择 (Capture/Compare 2 selection) 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00: 通道 2 被配置为输出 01: 通道 2 被配置为输入，IC2 映射在 TI2 上 10: 通道 2 被配置为输入，IC2 映射在 TI1 上 11: 通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)
7	OC1CE	通道 1 比较输出清 0 使能 (Output compare 1 clear enable) 0: OC1REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入高电平时，OC1REF 清零

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode) 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；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平，否则为高电平。</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；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。</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) 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) 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 TIMx_SMCR 寄存器的 TS 位选择)</p>

输入捕获模式：

Bit	Field	Description
15: 12	IC2F	<p>输入捕获 2 滤波器 (Input capture 2 filter) 参考 IC1F 的描述</p>
11: 10	IC2PSC	<p>输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 的描述</p>
9: 8	CC2S	<p>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：</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 TIMx_SMCR 寄存器的 TS 位选择)</p>

Bit	Field	Description
7: 4	IC1F	<p>通道 1 输入捕获滤波器 (Input capture 1 filter) 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) 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。</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) 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 TIMx_SMCR 寄存器的 TS 位选择)</p>

11.5.8 TIMx_CCMR2 捕获/比较模式寄存器 2

偏移地址: 0x1C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	CC4S		OC3CE	OC3M			OC3PE	OC3FE	CC3S	
IC4F			IC4PSC		CC4S		IC3F			IC3PSC			CC3S		
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

比较输出模式:

Bit	Field	Description
15	OC4CE	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 TIMx_SMCR 寄存器的 TS 位选择)
7	OC3CE	通道 3 比较输出清 0 使能 (Output compare 3 clear enable) 0: OC3REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入高电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode) 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。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为高电平</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为低电平</p> <p>011: 匹配时翻转。当 TIMx_CCR3=TIMx_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 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平；在递减计数时，当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平。</p> <p>注 1: 当 LOCK 级别设为 3 (TIMx_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 TIMx_CCR3 寄存器的预装载功能，写入 TIMx_CCR3 寄存器的数值立即生效</p> <p>1: 开启 TIMx_CCR3 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR3 的预装载值在更新事件到来时生效</p> <p>注 1: 当 LOCK 级别设为 3 (TIMx_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 (TIMx_CR1 寄存器的 OPM=1)，无需设定预装载寄存器，其它情况下，需要设定预装载寄存器，否则后续动作不确定。</p>

Bit	Field	Description
2	OC3FE	<p>通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。</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) 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 TIMx_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) 参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>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:</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 TIMx_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) 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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 CC3E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。</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	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</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 TIMx_SMCR 寄存器的 TS 位选择)</p>

11.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	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 (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。

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 (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, ICx 的极性/电平选择如下表所示:

表 11-11 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

11.5.10 TIMx_CNT 计数器

偏移地址: 0x24

复位值: 0x0000

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

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

11.5.11 TIMx_PSC 预分频器

偏移地址: 0x28

复位值: 0x0000

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

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

11.5.12 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。

11.5.13 TIMx_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>

11.5.14 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	<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 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。</p>

11.5.15 TIMx_CCR2 捕获/比较寄存器 2

偏移地址: 0x38

复位值: 0x0000

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

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

11.5.16 TIMx_CCR3 捕获/比较寄存器 3

偏移地址: 0x3C

复位值: 0x0000

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

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

11.5.17 TIMx_CCR4 捕获/比较寄存器 4

偏移地址: 0x40

复位值: 0x0000

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

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

11.5.18 TIMx_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，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
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)
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 (TIMx_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 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。
13	BKP	刹车输入极性 (Break Polarity) 0: 刹车输入低电平有效 1: 刹车输入高电平有效 注: 当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。
12	BKE	刹车功能使能 (Break enable) 0: 禁止刹车输入 1: 开启刹车输入 注 1: 当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1 时，该位不能被修改。 注 2: 刹车输入包括引脚输入、比较器比较结果输入和 CSS 时钟失效三类事件。

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。 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) 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。 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) 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 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_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 TIMx_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 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位) 注: 在系统复位后, LOCK 位只能写一次, 当写入 TIMx_BDTR 寄存器后, LOCK 被写保护。</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。 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>

11.5.19 TIMx_DCR DMA 控制寄存器

偏移地址: 0x48

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 11.4.10 DMA 章节。

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

Res.	DBL	Res.	DBA
	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 基址 (DMA base address) 这些位定义了 DMA 在连续模式下访问 TIMx_DMAR 寄存器的第一个地址。DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移值： 00000: TIMx_CR1 00001: TIMx_CR2 00010: TIMx_SMCR

11.5.20 TIMx_DMAR 连续模式 DMA 地址寄存器

偏移地址: 0x4C

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 11.4.10 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 TIMx_DMAR 寄存器的读写操作会导致对以下地址所在寄存器的存取操作： TIMx_CR1 地址 + DBA + DMA 索引, 其中 TIMx_CR1 地址是 TIMx_CR1 寄存器所在的地址, DBA 是 TIMx_DCR 寄存器中定义的基址, DMA 索引是 DMA 自动控制的偏移量, 它取决于 TIMx_DCR 寄存器中定义的 DBL 值。

11.5.21 TIMx_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 TIMx_CCR5 寄存器的预装载功能, 写入 TIMx_CCR5 寄存器的数值立即生效。 1: 开启 TIMx_CCR5 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR5 的预装载值在更新事件到来时生效。 <small>注 1: 当 LOCK 级别设为 3 (TIMx_BDTR 寄存器中的 LOCK 位) 时, 该位不能被修改。</small>
2: 0	Reserved	保留, 必须保持复位值。

11.5.22 TIMx_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) CC5 通道只能配置为输出: 如果在 TIMx_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 TIMx_CNT 的比较, 由于 CC5 通道为内部通道, 无法输出至引脚, 比较结果用于内部触发事件。

11.5.23 TIMx_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_ SHIFT_ EN	CCR4_ SHIFT_ EN	CCR3_ SHIFT_ EN	CCR2_ SHIFT_ EN	CCR1_ SHIFT_ EN	CCDRE PE
										rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 6		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。

11.5.24 TIMx_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。

12 TIM3/4 通用定时器

TIM3、TIM4 具有相同的功能，本章节统一表述为 TIMx，部分示意图以 TIM3 为例展示。

12.1 简介

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

12.2 功能框图

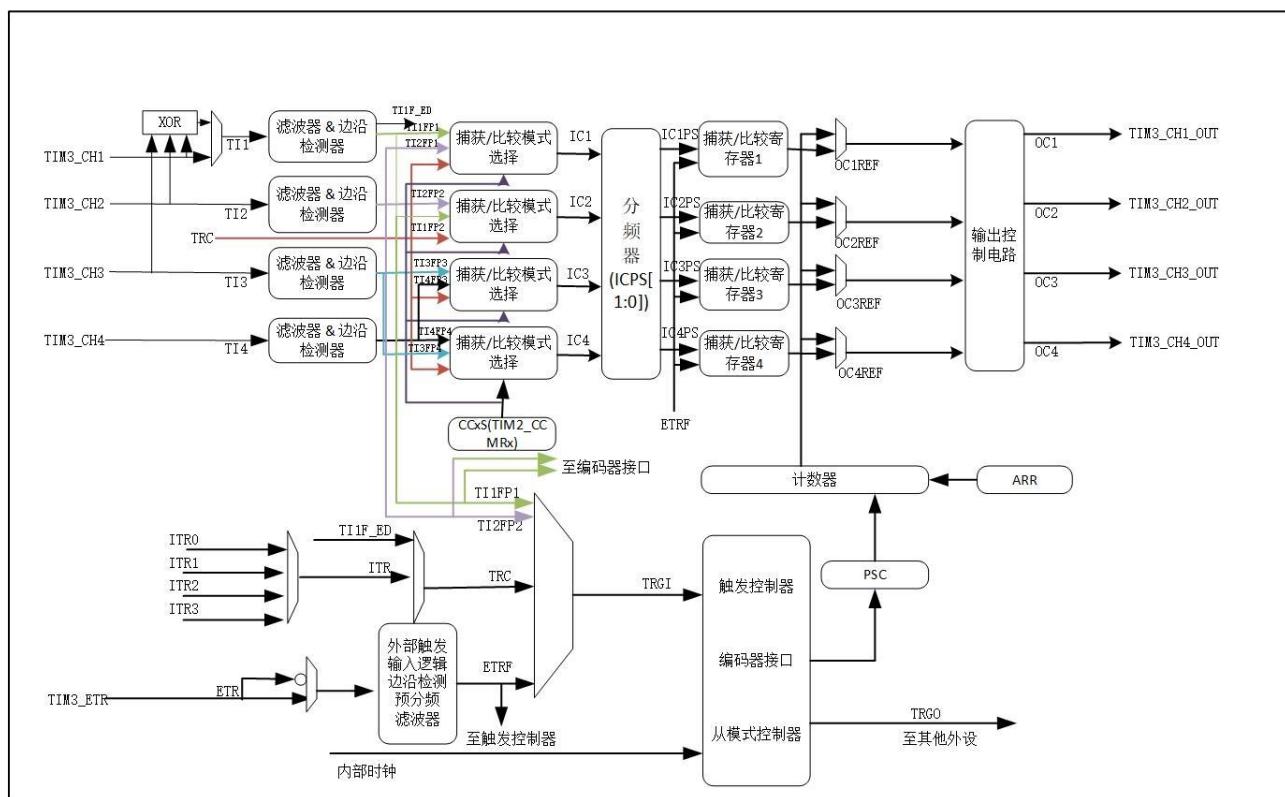


图 12-1 TIMx 结构图

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

12.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 时钟源可选：内部时钟源，外部时钟输入 (TIx、ETRx)，内部触发输入 (ITRx)

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

12.4 功能描述

12.4.1 时钟

12.4.1.1 时钟选择

计数器的时钟源有以下几种：

- 内部时钟 (INT_CK)
- 外部时钟模式 1：外部触发输入 TRGI (包含 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：

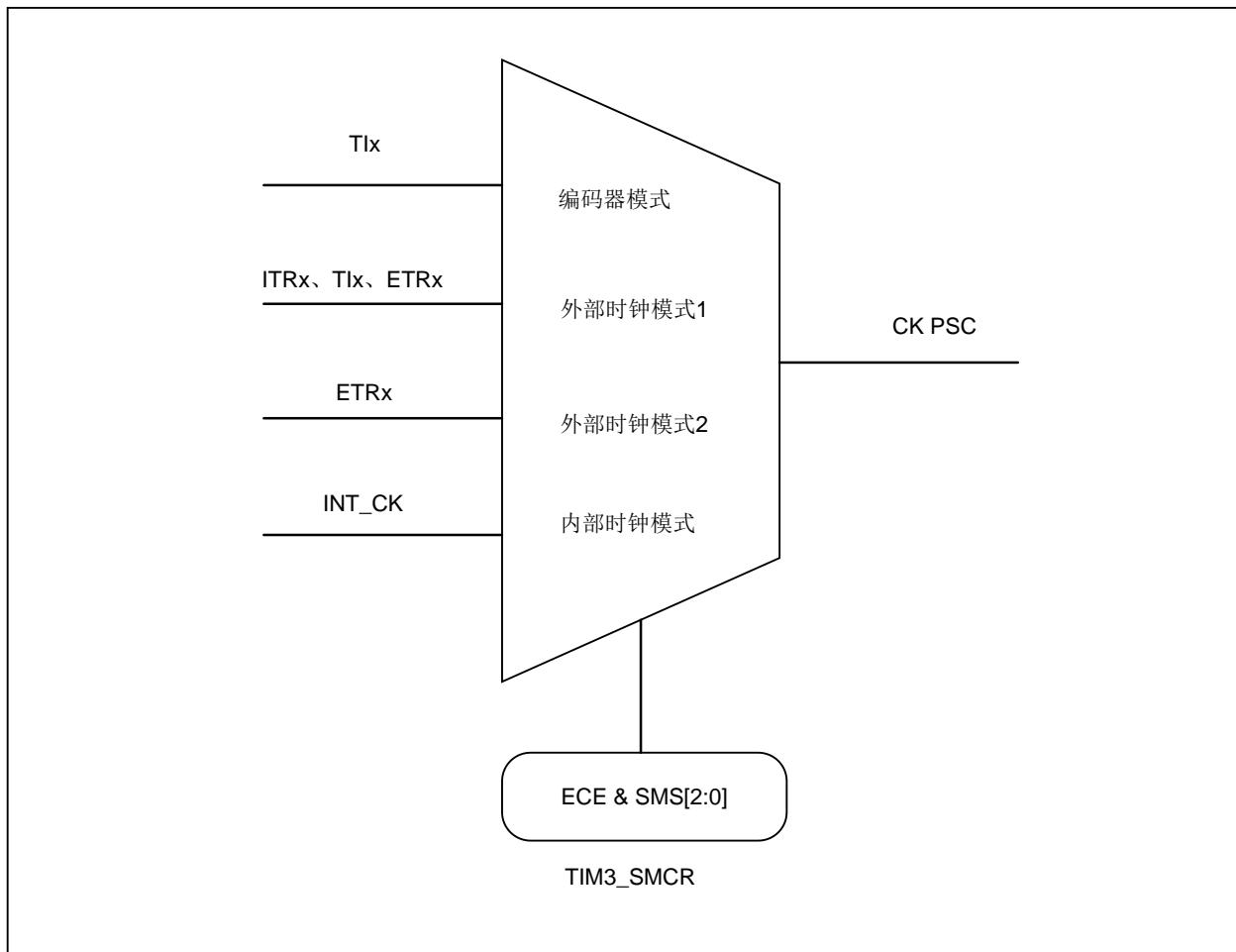


图 12-2 时钟选择

12.4.1.1.1 内部时钟源 (INT_CK)

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

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

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

例：计数器在 `TI1` 输入端的上升沿递增计数，具体配置如下：

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

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

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

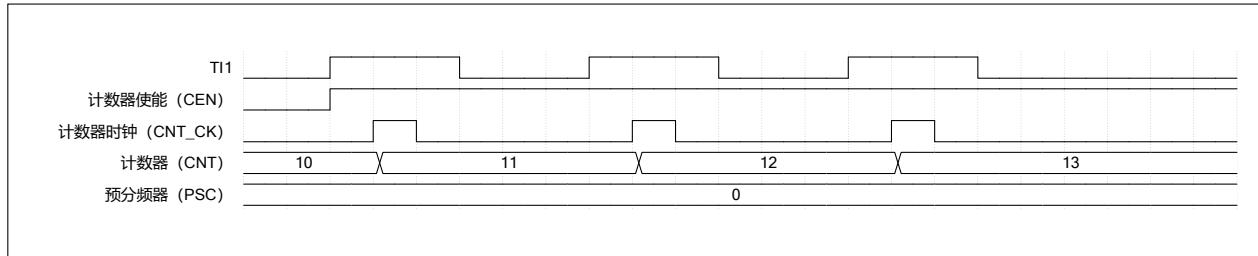


图 12-3 外部时钟模式 1 下的控制电路

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

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

例: `ETR` 的每 4 个下降沿计数一次, 递增计数, 具体配置如下:

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

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

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

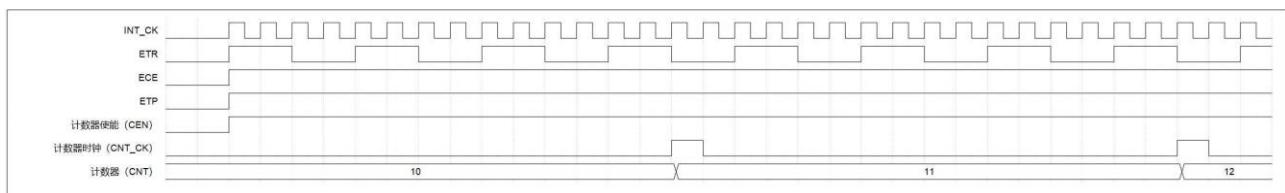


图 12-4 外部时钟模式 2 下的控制电路

12.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

12.4.1.2 时基单元

`TIMx` 的时基单元主要包括: 计数器寄存器 (`TIMx_CNT`)、预分频器寄存器 (`TIMx_PSC`) 和自动预装载寄存器 (`TIMx_ARR`)。

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

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

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

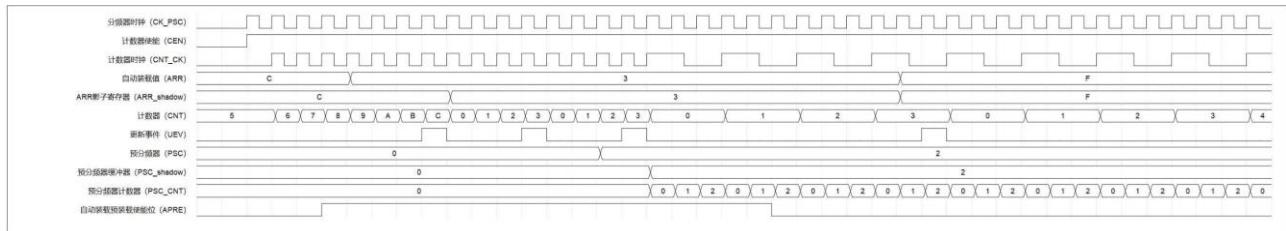


图 12-5 自动预装载

12.4.1.3 计数模式

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

12.4.1.3.1 递增计数模式

配置 **TIMx_CR1** 寄存器 CMS=0, DIR=0，选择递增计数模式。

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

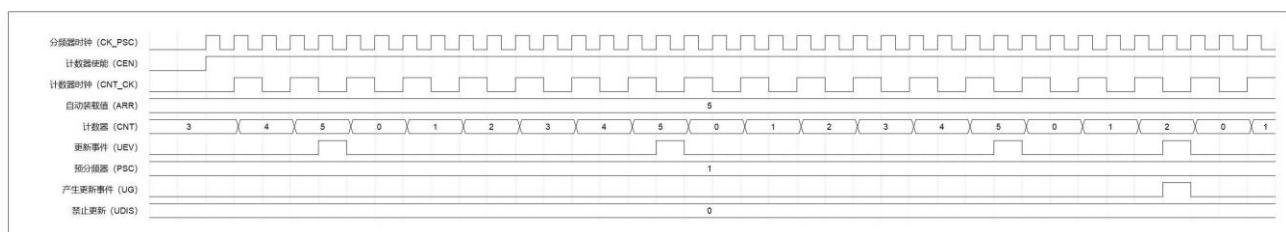


图 12-6 递增计数模式 (UDIS=0)

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

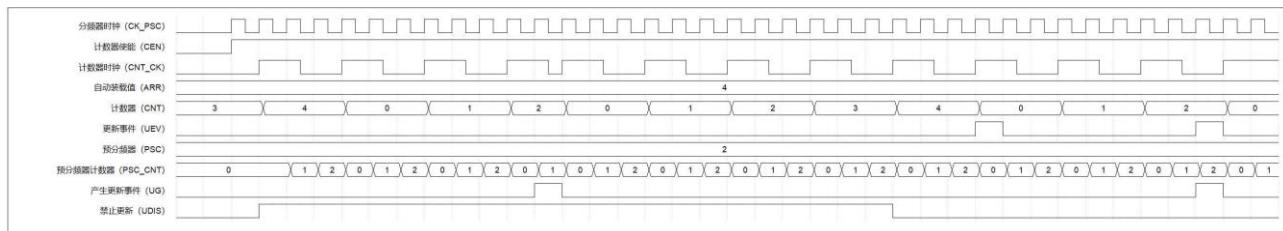


图 12-7 递增计数模式 (UDIS=1 禁止产生更新事件)

注：发生更新事件时

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

12.4.1.3.2 递减计数模式

配置 TIMx CR1 寄存器的 CMS=0, DIR=1, 选择递减计数模式。

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

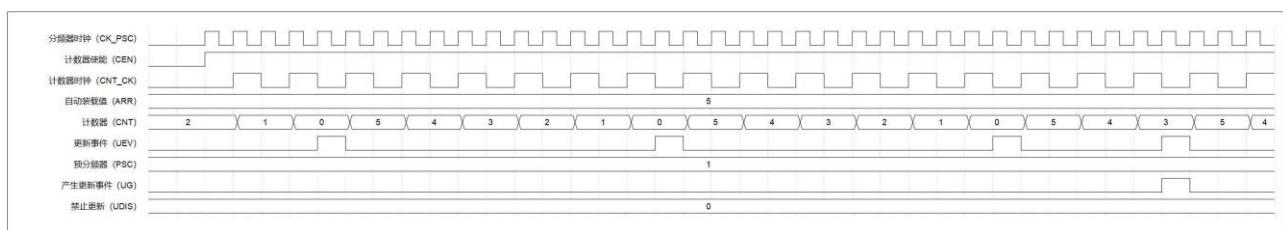


图 12-8 递减计数模式 (UDIS=0)

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

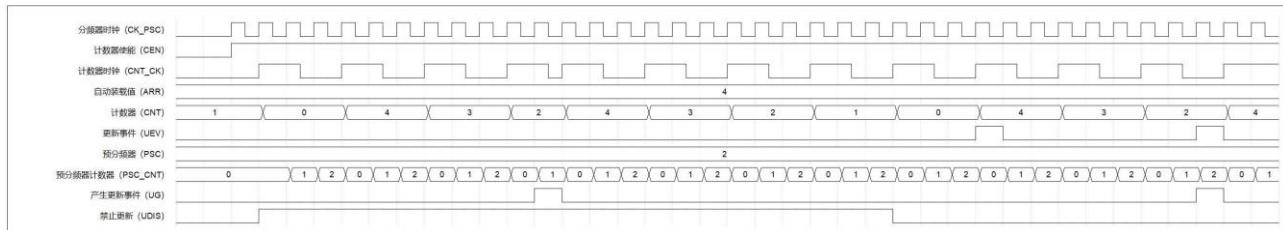


图 12-9 递减计数模式 (UDIS=1 禁止产生更新事件)

12.4.1.3.3 中央计数模式（递增/递减计数模式）

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

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

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

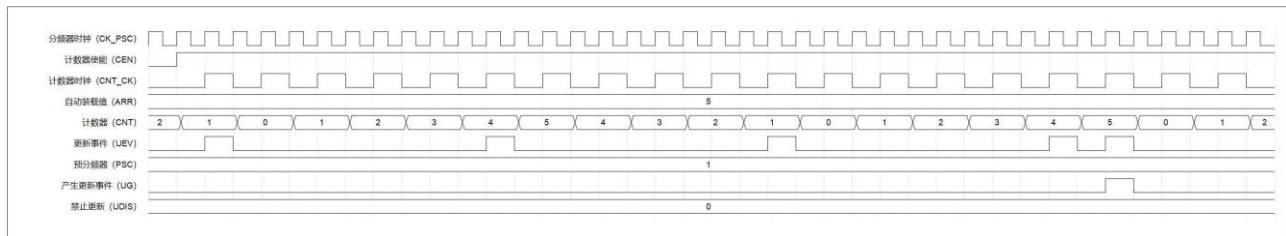


图 12-10 中央计数模式 (UDIS=0)

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

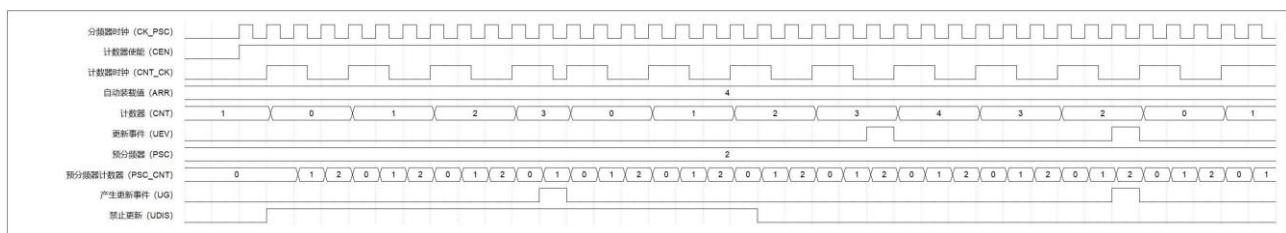


图 12-11 中央计数模式 (UDIS=1 禁止产生更新事件)

12.4.2 输入捕获

12.4.2.1 输入捕获

输入捕获部分包括数字滤波器、多路复用、预分频器等, 其结构如下图所示:

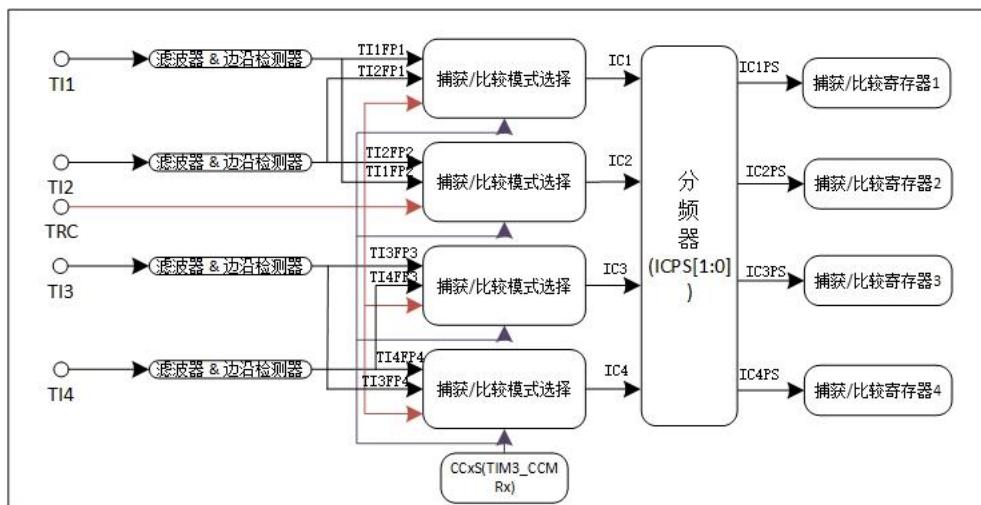


图 12-12 TIMx 输入捕获结构图

通过配置 **TIMx_CCMRx** 寄存器的 **ICxF**, 可以设置数字滤波器的滤波宽度 (滤波器的采样频率及数字滤波宽度如下表所示), 当数字滤波器的输入信号宽度大于滤波宽度时, 输入信号有效; 数字滤波器对输入

引脚 TIx 的输入信号采样后，产生一个滤波后的信号 TIxF ，然后通过极性可选的边沿检测器，产生一个有效信号 TIxFPx ，这个信号可以作为从模式控制器的触发输入信号，同时该信号经过预分频器产生一个信号 ICxPS ，用于触发输入捕获事件。

表 12-1 数字滤波器宽度与 ICxF 的对应关系表

$\text{IC1F}[3: 0]$	采样频率和滤波宽度	$\text{IC1F}[3: 0]$	采样频率和滤波宽度
0000	无滤波器，以 f_{DTS} 采样	1000	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=6$
0001	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=2$	1001	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=8$
0010	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=4$	1010	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=5$
0011	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=8$	1011	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=6$
0100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/2, N=6$	1100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=8$
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 ICx 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断或 DMA 使能，发生捕获事件时，将产生相应的中断或 DMA 请求。发生捕获事件时，会将状态寄存器(TIMx_SR)中的捕获标志位 CCxIF 置 1，通过配置 $\text{CCxIF}=0$ 或读取 TIMx_CCRx 中的数据，清除 CCxIF 标志位。当 CCxIF 未被清零时，发生输入捕获事件，重复捕获标志位 CCxOF 将会被置 1，通过配置 $\text{CCxOF}=0$ ，可以清除 CCxOF 标志位。

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

1. 配置 TIMx_CCMR1 寄存器的 $\text{CC1S}=01$ ， CC1 通道被配置为输入， IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 $\text{IC1F}[3: 0]$ ，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 $\text{CC1P}=0$ ，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 $\text{IC1PSC}[1:0]$ ，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 $\text{CC1E} = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 TIMx_DIER 寄存器的 $\text{CC1IE}=1$ ，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 $\text{CC1DE}=1$ ，允许捕获/比较通道 1 的 DMA 请求。

注：

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

12.4.2.2 PWM 捕获

PWM 输入模式的操作配置与一般输入捕获有以下不同点：

- 两个边沿有效且极性相反的 IC_x 信号被映射至同一个 TI_x 输入。

- 配置从模式为复位模式，将其中一路 TI_xFP 作为触发输入信号。

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

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

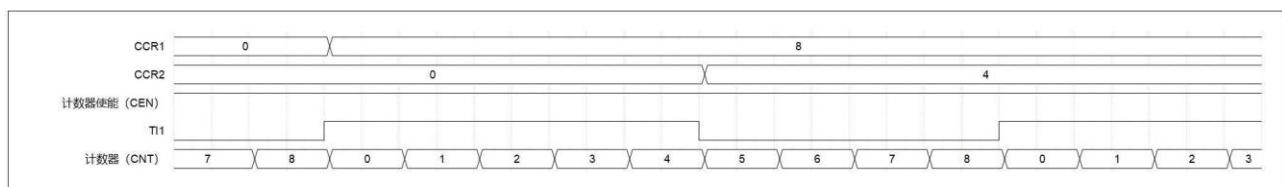


图 12-13 PWM 输入模式时序

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

12.4.3 比较输出

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

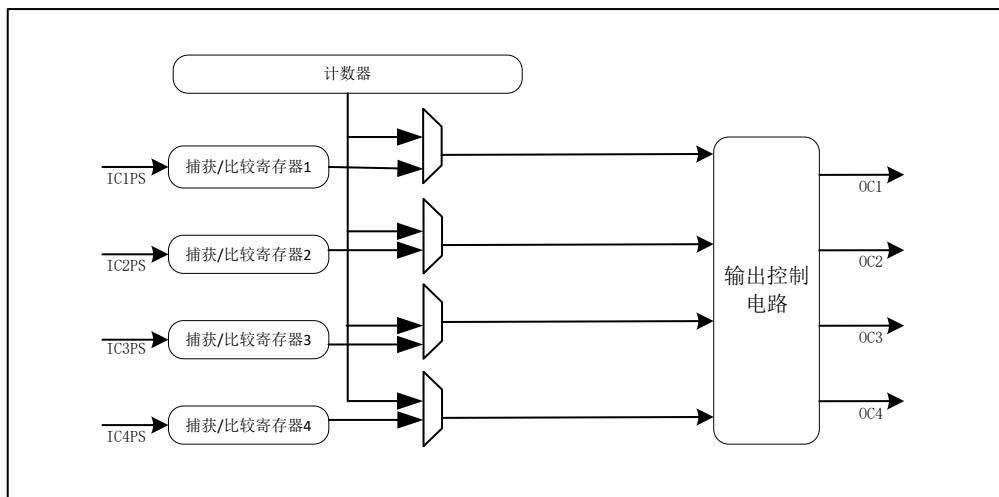


图 12-14 比较输出部分结构图

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

12.4.3.1 强制输出

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

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

12.4.3.2 比较输出

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

例如，当计数器与捕获/比较寄存器的内容匹配时，比较输出模式下的操作如下：

1. 在比较匹配时， OCxM 的值不同，输出通道 x 信号 OCx 的操作不同：

- ◆ $\text{OCxM} = 000$: OCx 信号保持它的电平
- ◆ $\text{OCxM} = 001$: OCx 信号被设置成有效电平
- ◆ $\text{OCxM} = 010$: OCx 信号被设置成无效电平
- ◆ $\text{OCxM} = 011$: OCx 信号进行翻转

2. 匹配时状态寄存器中的标志位置 1 (TIMx_SR 寄存器中的 CCxIF 位)。

3. 当配置了 `TIMx_DIER` 寄存器中的 `CCxIE =1`, 匹配时则产生一个中断。
4. 当配置了 `TIMx_DIER` 寄存器中的 `CCxDE =1`, 匹配时则产生一个 DMA 请求 (仅适用于有内置 DMA 的产品)。

比较输出模式也可以用来输出一个单脉冲 (单脉冲输出模式)。

例如, 通道 1 的比较输出模式的配置步骤如下:

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

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

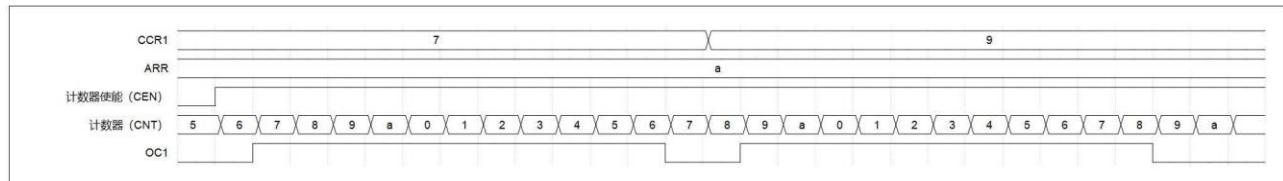


图 12-15 比较输出模式, `OC1` 信号在匹配时翻转

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

12.4.3.3 PWM 输出

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

配置与通道 x 对应的 `TIMx_CCMRx` 寄存器的 `OCxM=110` 或 `OCxM=111`, 选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下, 计数器和 `CCRx` 会一直进行比较, 根据配置和比较结果, 通道 x 输出不同的信号, 因此 `TIMx` 可以产生 4 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 `TIMx_CCRx` 的预装载功能和 `TIMx_ARR` 寄存器的预装载功能。写入 `TIMx_CCRx` 预装载寄存器和

TIMx_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 **TIMx_EGR** 的 **UG=1**，产生更新事件用于初始化所有的寄存器。

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

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

12.4.3.3.1 PWM 边沿对齐模式——递增计数模式

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

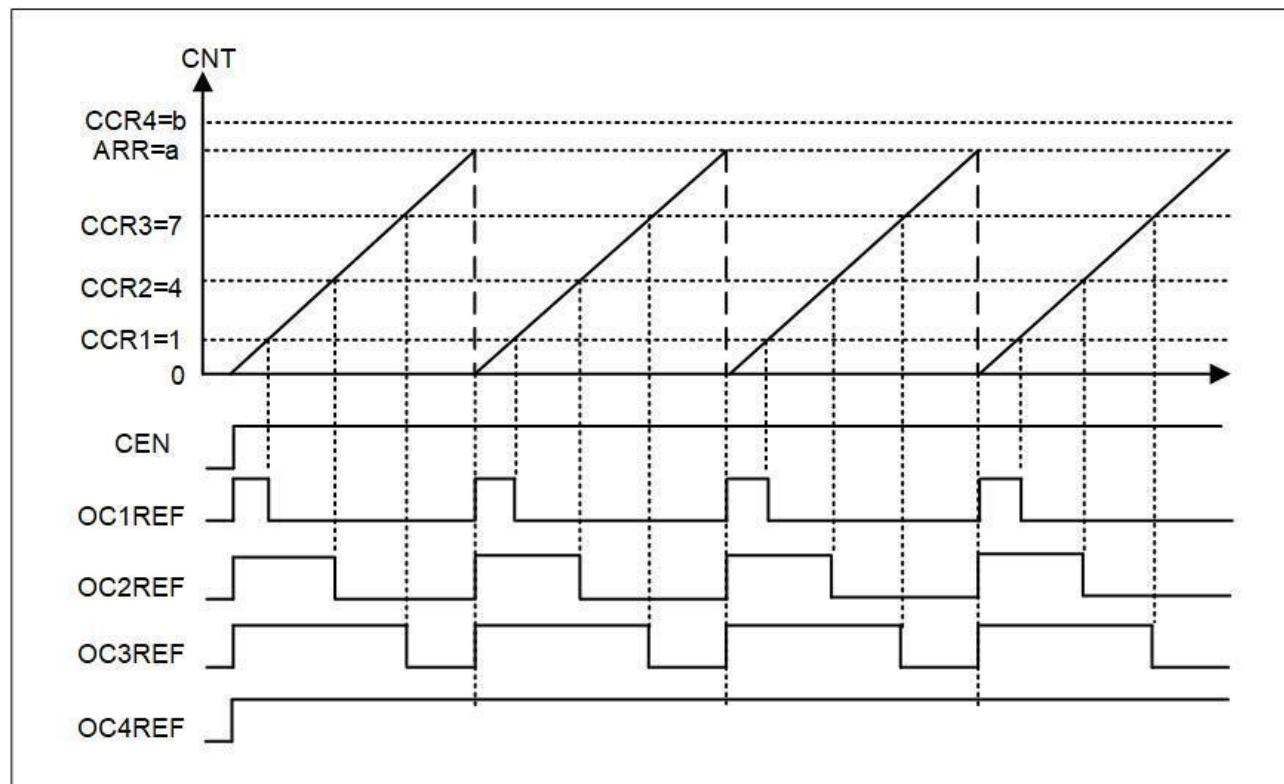


图 12-16 边沿对齐递增计数时 PWM 模式 1 的波形

12.4.3.3.2 PWM 边沿对齐模式——递减计数模式

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

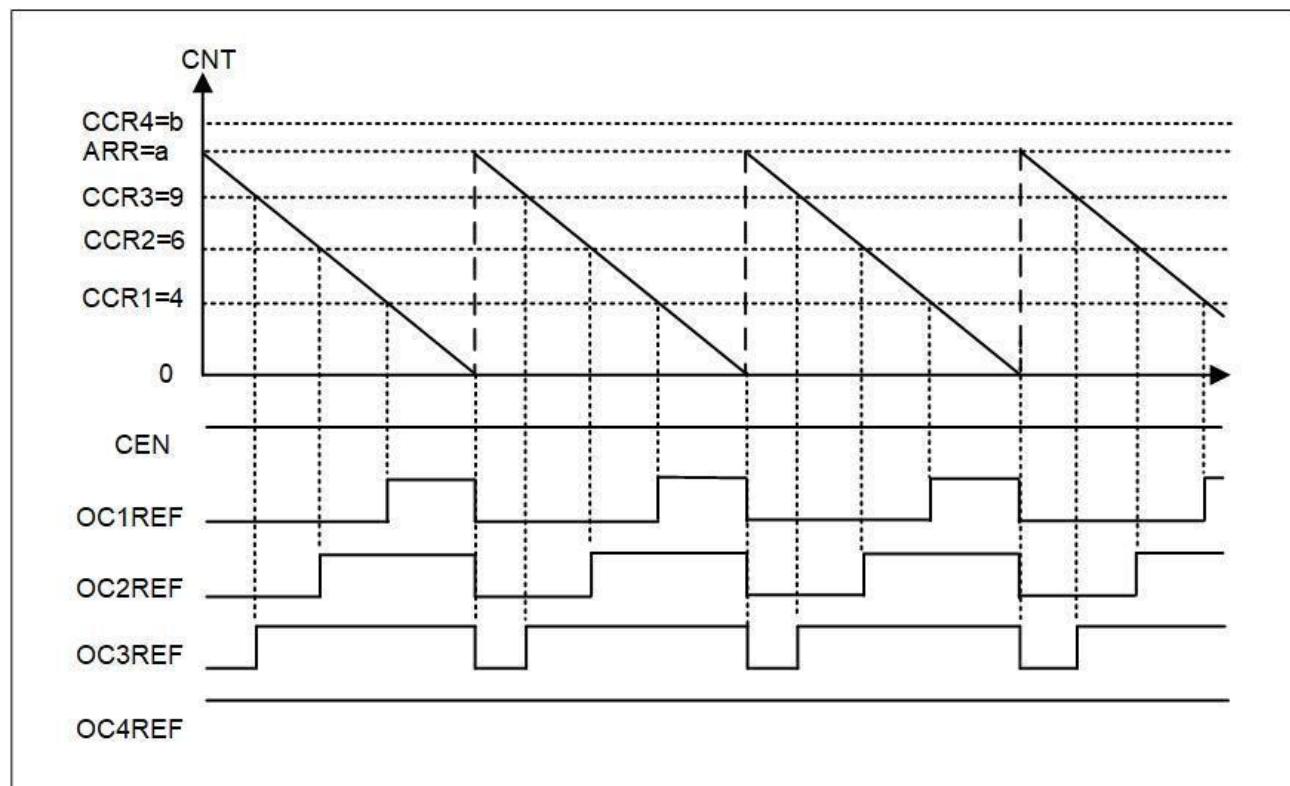


图 12-17 边沿对齐递减计数时 PWM 模式 1 的波形

12.4.3.3.3 PWM 中央对齐模式

首先配置 TIMx 计数器为中央对齐计数模式，配置 TIMx_CCMRx 寄存器的 $\text{CCxS}=00$ ，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 ($\text{CMS}=01$)、在计数器递增计数时被设置 ($\text{CMS}=10$)、或在计数器递增或递减计数时被设置 ($\text{CMS}=11$)。下图为 $\text{CCR1}=4$, $\text{CCR2}=6$, $\text{CCR3}=9$, $\text{CCR4}=b$, $\text{ARR}=a$ 时中央对齐 PWM 模式 1 的波形实例。

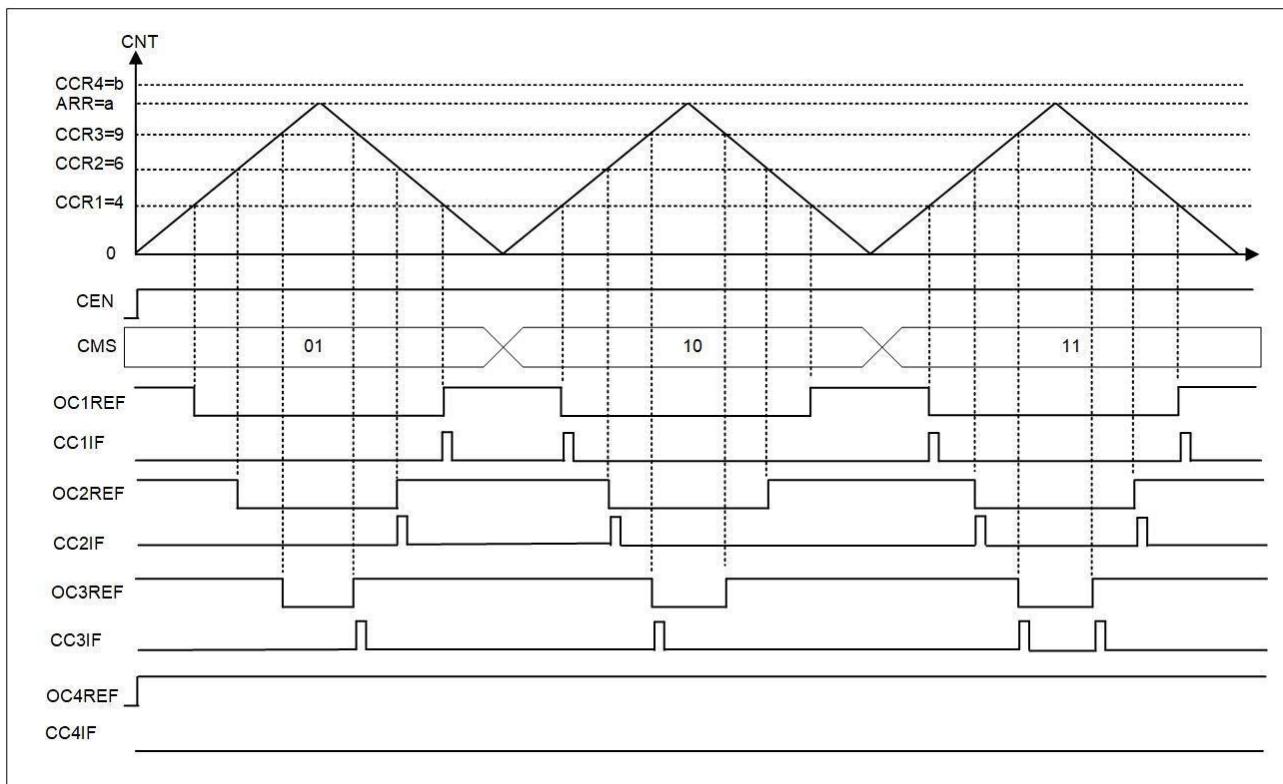


图 12-18 中央对齐 PWM 模式 1 的波形

注：

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

12.4.3.4 外部事件清除 OCxREF

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

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

1. 配置 TIMx_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
2. 配置 TIMx_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
3. 配置 TIMx_SMCR 寄存器 ETF[3: 0]和 ETP，配置 ETR 信号的触发极性和滤波宽度。

下图显示了当 ETR 输入变化触发 ETRF 为高时，对应不同 OCxCE 的值，OCxREF 信号的动作 (PWM 模式)。

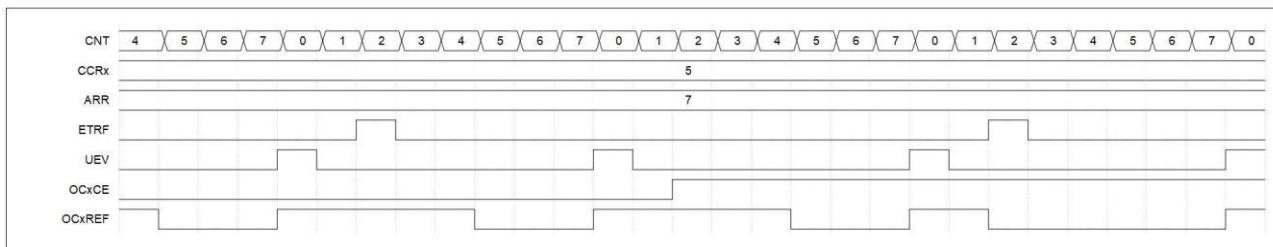


图 12-19 外部事件清除 OCxREF

12.4.3.5 单脉冲输出

单脉冲模式（OPM）下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 **TIMx_CR1** 寄存器的 **OPM=1**，选择单脉冲模式，触发信号有效沿或配置 **CEN=1** 都可以启动计数器，直到下一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 ≤ ARR**。
- 递减计数方式：计数器 **CNT > CCRx**。

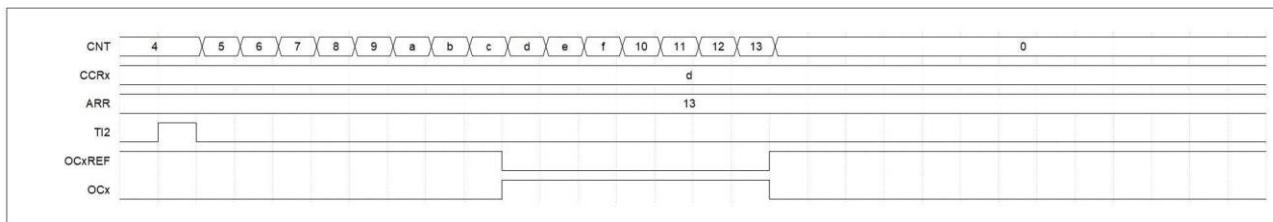


图 12-20 单脉冲模式

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。

假定 **TI2FP2** 作为触发源：

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

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

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

1. 配置 **TIMx_CCMR1** 寄存器 **OC1M = 111**，选择 **PWM** 模式 2。
2. 配置 **TIMx_CCER** 寄存器 **CC1P = 1**，输出低电平有效。
3. 配置 **TIMx_CCMR1** 中 **OC1PE = 1** 和 **TIMx_CR1** 寄存器中 **ARPE=1**，使能预装载寄存器。
4. 配置 **TIMx_CCR1** 寄存器和 **TIMx_ARR** 寄存器。

5. 配置 TIMx_EGR 寄存器 $\text{UG}=1$ 产生一个更新事件。

6. 等待在 TI2 上的一个外部触发事件。

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

12.4.3.5.1 OCx 快速使能

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

12.4.4 从模式

12.4.4.1 编码器接口

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

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

注: 编码器模式不支持外部时钟模式 2。

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

表 12-2 计数方向与编码器信号的关系

计数模式	相对电平 (TI1FP1 相对于 TI2 , TI2FP2 相对于 TI1)	TI1FP1 信号		TI2FP2 信号	
计数模式	相对电平 (TI1FP1 相对于 TI2 , TI2FP2 相对于 TI1)	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	
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模式下的配置和时序图，从图中可以看出计数信号的产生和方向控制。具体配置如下：

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

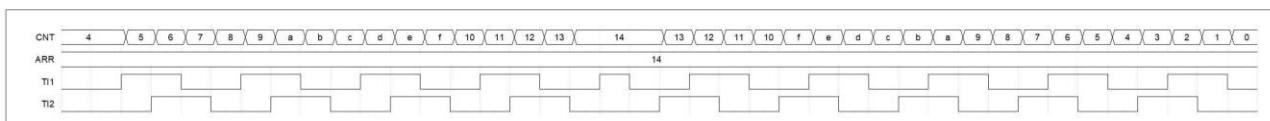


图 12-21 编码器模式下的计数器时序图

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

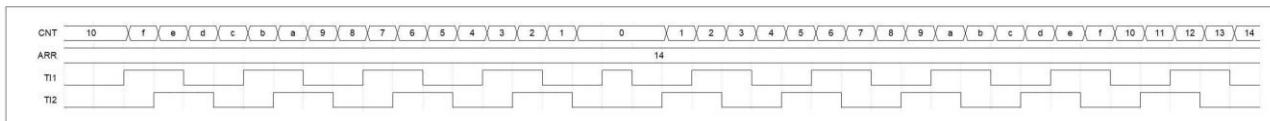


图 12-22 IC1FP1 反相编码器接口模式时序图

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。

12.4.4.2 复位模式

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

例如, `TI2` 输入端的下降沿触发计数器重启:

1. 配置 `TIMx_CCMR1` 寄存器的 `CC2S=01`, `CC2` 通道被配置为输入模式; `IC2` 映射在 `TI2` 上, 配置 `TIMx_CCER` 寄存器的 `CC2P=1`, 检测下降沿。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS = 100`, 从模式选择复位模式; 配置 `TIMx_SMCR` 寄存器的 `TS = 110`, 选择滤波后的定时器输入 2 (`TI2FP2`) 作为同步计数器的触发输入; 配置 `TIMx_SMCR` 寄存器的 `ETF=0000`, 不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`, 选择计数方向为递增计数; 配置 `PSC=0`, 不分频; 配置 `CEN=1`, 使能计数器。

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

下图为复位模式下 `TIMx_ARR = 0x13` 的时序图。

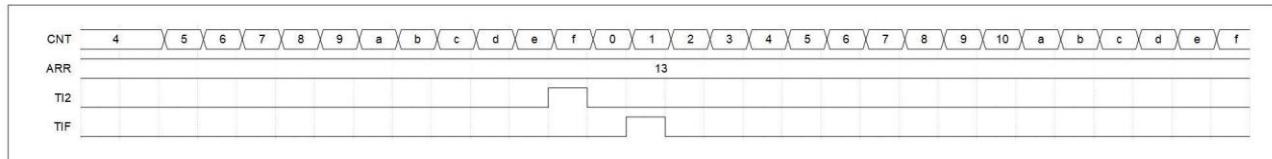


图 12-23 复位模式的控制时序图

12.4.4.3 门控模式

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

例如, 计数器只在 `TI1` 为高时计数:

1. 配置 `TIMx_CCMR1` 寄存器的 `CC1S=01`, `CC1` 通道被配置为输入模式, `IC1` 映射在 `TI1` 上; 配置 `TIMx_CCER` 寄存器的 `CC1P=0`, 检测 `TI1` 上的高电平。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS=101`, 从模式选择为门控模式; 配置 `TIMx_SMCR` 寄存器的 `TS=101`, 选择滤波后的定时器输入 1 (`TI1FP1`) 作为同步计数器的触发输入; 配置 `TIMx_SMCR` 寄存器的 `ETF=0000`, 不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`, 选择计数方向为递增计数; 配置 `PSC=0`, 不分频; 配置 `CEN=1`, 使能计数器。

计数器的时钟源由内部时钟提供, 当检测到 `TI1` 的高电平, 计数器开始计数, 当 `TI1` 为低电平时, 计

数器停止计数。计数器停止会将 TIF 置 1。

下图为门控模式下 TIMx_ARR=0xf 的时序图。

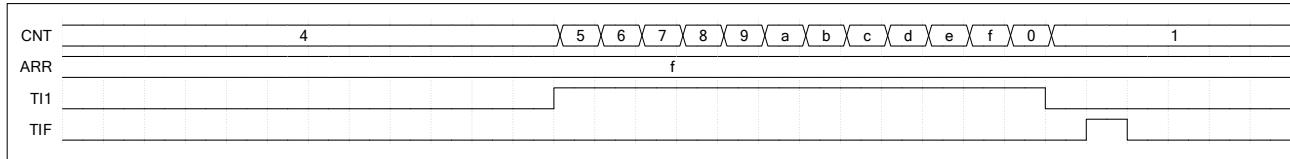


图 12-24 门控模式下的控制时序图

12.4.4.4 触发模式

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

例如，计数器在 TI1 输入的上升沿开始计数：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入模式，IC1 映射在 TI1 上，配置 TIMx_CCER 寄存器的 CC1P=0，检测上升沿。
2. 配置 TIMx_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIMx_SMCR 寄存器的 TS=101，选择滤波后的定时器输入 1 (TI1FP1) 作为计数器的触发输入；配置 TIMx_SMCR 寄存器的 ETF=0000，不使用数字滤波器。
3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。

计数器的时钟源由内部时钟提供，当检测到 TI1 的上升沿，计数器开始计数。

下图为触发模式下 TIMx_ARR=0xf 的时序图。

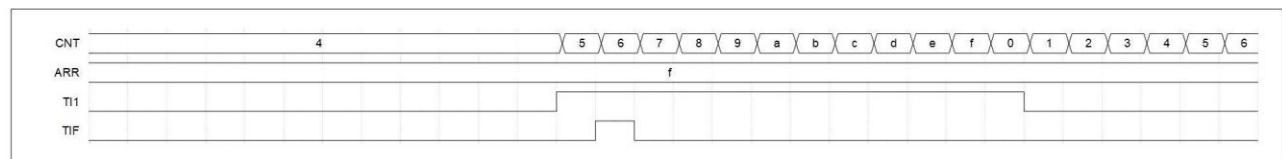


图 12-25 触发器模式下的控制时序图

12.4.4.5 外部时钟模式 2+从模式

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

例如，从模式选择触发模式，计数器在 ETR 的每一个上升沿计数一次：

1. 配置 TIMx_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIMx_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIMx_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIMx_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。
2. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上作为输入捕获源；配置 TIMx_CCER 寄存器的 CC1P=0，选择上升沿有效。

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

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

计数器在 **TI1** 的上升沿开始计数, 并将 **TIF** 置 1。**ETR** 信号的上升沿和计数器实际计数时钟间的延时取决于 **ETR** 输入端的同步电路设计。

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

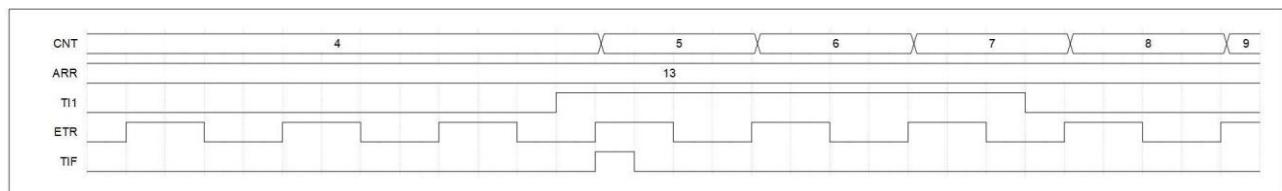


图 12-26 外部时钟模式 2+从模式(触发模式)控制时序图

12.4.5 定时器同步

不同的定时器在内部连接, 可以实现定时器之间的级联或同步。

详细描述请参考 **TIM1** 相关章节。

12.4.6 定时器异或

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

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

1. 配置 **TIMx_CR2** 寄存器的 **TI1S=1**, 配置定时器的三个输入经异或后连接到 **TI1** 输入通道。
2. 配置 **TIMx_CCMR1** 寄存器的 **CC1S=01**, **CC1** 通道被配置为输入, **IC1** 映射在 **TI1** 上。
3. 配置 **TIMx_CCMR1** 寄存器的 **IC1F[3: 0]**, 配置数字滤波器的滤波宽度(按需配置)。
4. 配置 **TIMx_CCER** 寄存器的 **CC1P=0**, 选择捕获发生在 **TI1** 信号的上升沿。
5. 配置 **TIMx_CCMR1** 寄存器的 **IC1PSC**, 选择预分频系数。
6. 配置 **TIMx_CCER** 寄存器的 **CC1E = 1**, 开启输入/捕获通道 1 的捕获使能。
7. 配置 **TIMx_CR1** 寄存器的 **CEN=1**, 启动计数器。

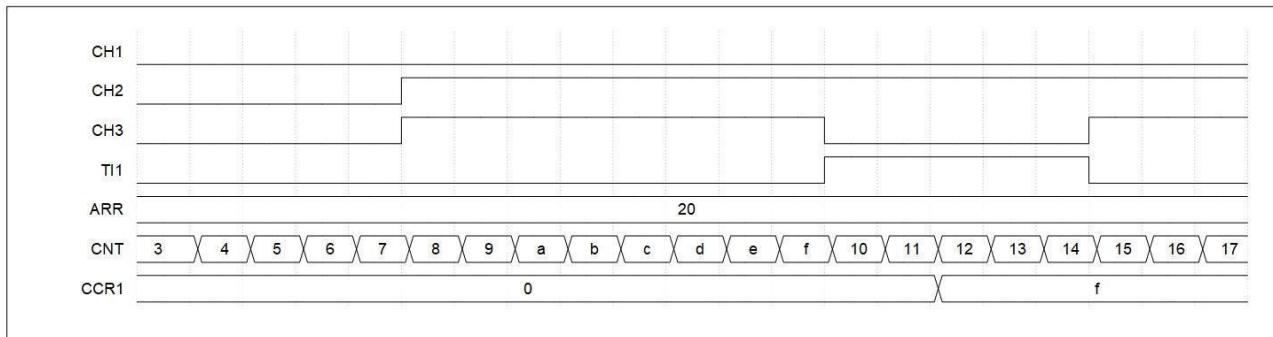


图 12-27 (TI1 异或输入) 输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

12.4.7 调试模式

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

12.4.8 中断

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

表 12-3 中断事件一览表

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

12.4.9 DMA

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

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

TIMx_DCR 寄存器的 DBL 位定义了 DMA 连续传送的长度，即传输寄存器数量。当对 TIMx_DMAR 进行读写操作时，定时器识别 DBL，确定需要传输的寄存器数量。TIMx_DCR 寄存器的 DBA 位定义了

DMA 传输的基地址，定义从 TIMx_CR1 寄存器地址开始的偏移量（00000 为 TIMx_CR1、00001 为 TIMx_CR2 …… 00110 为 TIMx_CCMR1 等）。

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

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

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

12.5 寄存器描述

表 12-4 TIMx 寄存器概览

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

Offset	Acronym	Register Name	Reset
0x4C	TIMx_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000

12.5.1 TIMx_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		APRE		CMS		DIR		OPM		URS		UDIS		CEN	
		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, TIx) 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	APRE	<p>自动重装载预装载使能 (Auto-reload preload enable) 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器</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>

Bit	Field	Description
3	OPM	单脉冲模式 (One pulse mode) 0: 禁止单脉冲模式, 在发生更新事件时, 计数器继续计数 1: 使能单脉冲模式, 在发生下一次更新事件或软件清除 CEN 位时, 计数器停止计数
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC、CCRx) 保持值不变。如果设置了 EGR_UG 位为 1, 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。

12.5.2 TIMx_CR2 控制寄存器 2

偏移地址: 0x04

复位值: 0x0000

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

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	TI1S	TI1 选择 (TI1 selection) 0: TIMx_CH1 管脚连到 TI1 输入 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入

Bit	Field	Description
6: 4	MMS	<p>主模式选择 (Master mode selection) 这些位控制 TRGO 信号的选择，用于选择在主模式下送到从定时器的同步信息：</p> <p>000: 复位 TIMx_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 1: 发生更新事件时，发送 CCx 的 DMA 请求 注：仅适用于有内置 DMA 的产品</p>
2: 0	Reserved	保留，必须保持复位值。

12.5.3 TIMx_SMCR 从模式控制寄存器

偏移地址: 0x08

复位值: 0x0000

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

Bit	Field	Description
14	ECE	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2。 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2, ETRF 信号上的任意有效沿驱动计数器计数</p> <p>注 1: 仅适用于支持外部触发的产品。 注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。 注 3: TS ≠ 111 时, 复位模式, 门控模式和触发模式可以与外部时钟模式 2 同时使用。 注 4: 同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入是 ETR。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须低于 TIMx 主时钟 PCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。</p> <p>00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8</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) 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 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> <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>比较器输出信号清除选择 (Output compare clear selection)</p> <p>在 PWM 模式下, 清除比较器输出</p> <p>0: 外部触发信号作为清除信号</p> <p>1: 比较器输出作为清除信号</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 12-5 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	TIM5	TIM2	TIM3	TIM4
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM1	TIM3	TIM4	TIM8
TIM8	TIM1	TIM2	TIM4	TIM5

12.5.4 TIMx_DIER DMA/中断使能寄存器

偏移地址: 0x0C

复位值: 0x0000

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

Bit	Field	Description
2	CC2IE	允许捕获/比较 2 中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较 2 中断 1: 允许捕获/比较 2 中断
1	CC1IE	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

12.5.5 TIMx_SR 状态寄存器

偏移地址: 0x10

复位值: 0x0000

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

Bit	Field	Description
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 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。

12.5.6 TIMx_EGR 事件产生寄存器

偏移地址: 0x14

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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, TIMx_SR 寄存器的 TIF = 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA, 由硬件自动清 0。
5	Reserved	保留，必须保持复位值。
4	CC4G	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	产生通道 1 捕获/比较事件 (Capture/Compare 1 generation) 该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0。 0: 无动作 1: 通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出: CC1IF 置 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。 若通道 CC1 配置为输入: CC1IF 置 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若 CC1IF 已经为 1, 则设置 CC1OF = 1。
0	UG	产生更新事件 (Update generation) 0: 无动作 1: 初始化计数器, 并产生一个更新事件。由硬件自动清 0, 如果选择了中央对齐或递增计数模式, 计数器被清 0; 如果选择递减计数模式, 计数器将载入自动重载值。预分频计数器将同时被清除。

12.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址: 0x18

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	CC2S		OC1CE	OC1M			OC1PE	OC1FE	CC1S	
IC2F			IC2PSC			CC2S		IC1F				IC1PSC			CC1S
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CC_xS 定义。该寄存器 CC_xS 以外其它位的作用在输入模式和输出模式下不同。OC_{xx} 描述了通道在输出模式下的功能，IC_{xx} 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能（Output compare 2 clear enable） 参考 OC1CE 的描述。
14: 12	OC2M	通道 2 比较输出模式（Output compare 2 mode） 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能（Output compare 2 preload enable） 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能（Output compare 2 fast enable） 参考 OC1FE 的描述。
9: 8	CC2S	通道 2 捕获/比较选择（Capture/Compare 2 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 2 被配置为输出 01：通道 2 被配置为输入，IC2 映射在 TI2 上 10：通道 2 被配置为输入，IC2 映射在 TI1 上 11：通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时（由 TIMx_SMCR 寄存器的 TS 位选择）
7	OC1CE	通道 1 比较输出清 0 使能（Output compare 1 clear enable） 0：OC1REF 不受 ETR 输入的影响 1：当检测到 ETR 输入高电平时，OC1REF 清零

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode) 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。</p> <p>000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。</p> <p>100: 强制为低。强制 OC1REF 为低电平。</p> <p>101: 强制为高。强制 OC1REF 为高电平。</p> <p>110: PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平，否则为高电平。</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；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。</p> <p>注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。</p>
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能，写入 TIMx_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时生效。</p> <p>注：若该位置 1，在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1)，是否设定预装载寄存器无影响；其它情况下，需要设定预装载寄存器，否则后续动作不确定。</p>
2	OC1FE	<p>通道 1 比较输出快速使能 (Output compare 1 fast enable) 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>

Bit	Field	Description
1: 0	CC1S	<p>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：</p> <ul style="list-style-type: none"> 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 TIMx_SMCR 寄存器的 TS 位选择)

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) 参考 IC1F 的描述</p>
11: 10	IC2PSC	<p>输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 的描述</p>
9: 8	CC2S	<p>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：</p> <ul style="list-style-type: none"> 00: 通道 2 被配置为输出 01: 通道 2 被配置为输入，IC2 映射在 TI2 上 10: 通道 2 被配置为输入，IC2 映射在 TI1 上 11: 通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC1F	<p>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。</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) 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。</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) 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 TIMx_SMCR 寄存器的 TS 位选择)</p>

12.5.8 TIMx_CCMR2 捕获/比较模式寄存器 2

偏移地址: 0x1C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	CC4S		OC3CE	OC3M			OC3PE	OC3FE	CC3S	
IC4F			IC4PSC		CC4S		IC3F			IC3PSC			CC3S		
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

比较输出模式:

Bit	Field	Description
15	OC4CE	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 TIMx_SMCR 寄存器的 TS 位选择)
7	OC3CE	通道 3 比较输出清 0 使能 (Output compare 3 clear enable) 0: OC3REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入高电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode) 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。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为高电平</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为低电平</p> <p>011: 匹配时翻转。当 TIMx_CCR3=TIMx_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 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平；在递减计数时，当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平。</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 TIMx_CCR3 寄存器的预装载功能，写入 TIMx_CCR3 寄存器的数值立即生效</p> <p>1: 开启 TIMx_CCR3 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR3 的预装载值在更新事件到来时生效</p> <p>注：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM= 1)，无需设定预装载寄存器，其它情况下，需要设定预装载寄存器，否则后续动作不确定。</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) 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 TIMx_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) 参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>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：</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 TIMx_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) 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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 CC3E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。</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	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</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 TIMx_SMCR 寄存器的 TS 位选择)</p>

12.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20

复位值: 0x0000

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

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	<p>通道 1 输入/捕获输出使能 (Capture/Compare 1 output enable)</p> <p>通道 1 配置为输出时：</p> <p>0: 关闭。OC1 禁止输出</p> <p>1: 开启。OC1 信号输出到对应的输出引脚</p> <p>CC1 通道配置为输入：</p> <p>该位决定了输入捕获功能是否启用。</p> <p>0: 捕获禁止</p> <p>1: 捕获使能</p>

输入模式下，ICx 的极性/电平选择如下表所示：

表 12-6 ICx 极性/电平选择表

CCxP	CCxNP	ICx 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/高电平有效
0	1	保留

12.5.10 TIMx_CNT 计数器

偏移地址：0x24

复位值：0x0000

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

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

12.5.11 TIMx_PSC 预分频器

偏移地址：0x28

复位值：0x0000

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

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

12.5.12 TIMx_ARR 自动预装载寄存器

偏移地址: 0x2C

复位值: 0x0000

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

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

12.5.13 TIMx_CCR1 捕获/比较寄存器 1

偏移地址: 0x34

复位值: 0x0000

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

Bit	Field	Description
15: 0	CCR1	通道 1 捕获/比较的值 (Capture/Compare 1 value) 通道 1 配置为输入: 上一次捕获事件发生时捕获的计数器值存放于 CCR1 (此时 CCR1 寄存器为只读)。 通道 1 配置为输出: 如果在 TIMx_CCMR1 寄存器 (OC1PE 位) 中未选择预装载功能, 写入的数值会立即传输至对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时, 此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较, 并将比较结果反映到 OC1 端口的输出信号上。

12.5.14 TIMx_CCR2 捕获/比较寄存器 2

偏移地址: 0x38

复位值: 0x0000

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

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

12.5.15 TIMx_CCR3 捕获/比较寄存器 3

偏移地址: 0x3C

复位值: 0x0000

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

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

12.5.16 TIMx_CCR4 捕获/比较寄存器 4

偏移地址: 0x40

复位值: 0x0000

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

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

12.5.17 TIMx_DCR DMA 控制寄存器

偏移地址: 0x48

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 12.4.9 DMA 章节。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 基址 (DMA base address) 这些位定义了 DMA 在连续模式下访问 TIMx_DMAR 寄存器的第一个地址。DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移值: 00000: TIMx_CR1 00001: TIMx_CR2 00010: TIMx_SMCR

12.5.18 TIMx_DMAR 连续模式 DMA 地址寄存器

偏移地址: 0x4C

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 12.4.9 DMA 章节。

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

Bit	Field	Description
15: 0	DMAB	DMA 连续传送地址寄存器 (DMA address register for burst accesses) 对 TIMx_DMAR 寄存器的读写操作会导致对以下地址所在寄存器的存取操作： TIMx_CR1 地址 + DBA + DMA 索引，其中 TIMx_CR1 地址是 TIMx_CR1 寄存器所在的地址，DBA 是 TIMx_DCR 寄存器中定义的基地址，DMA 索引是 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL 值。

13 TIM2/5 通用定时器

TIM2、TIM5 具有相同的功能，本章节统一表述为 TIMx，部分示意图以 TIM2 为例展示。

13.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 32 位计数方向可调的自动重装载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。通用定时器具有多种用途，如输入功能（测量输入信号的脉冲宽度、频率，PWM 输入等），输出功能（PWM 输出、单脉冲模式输出等）。

13.2 功能框图

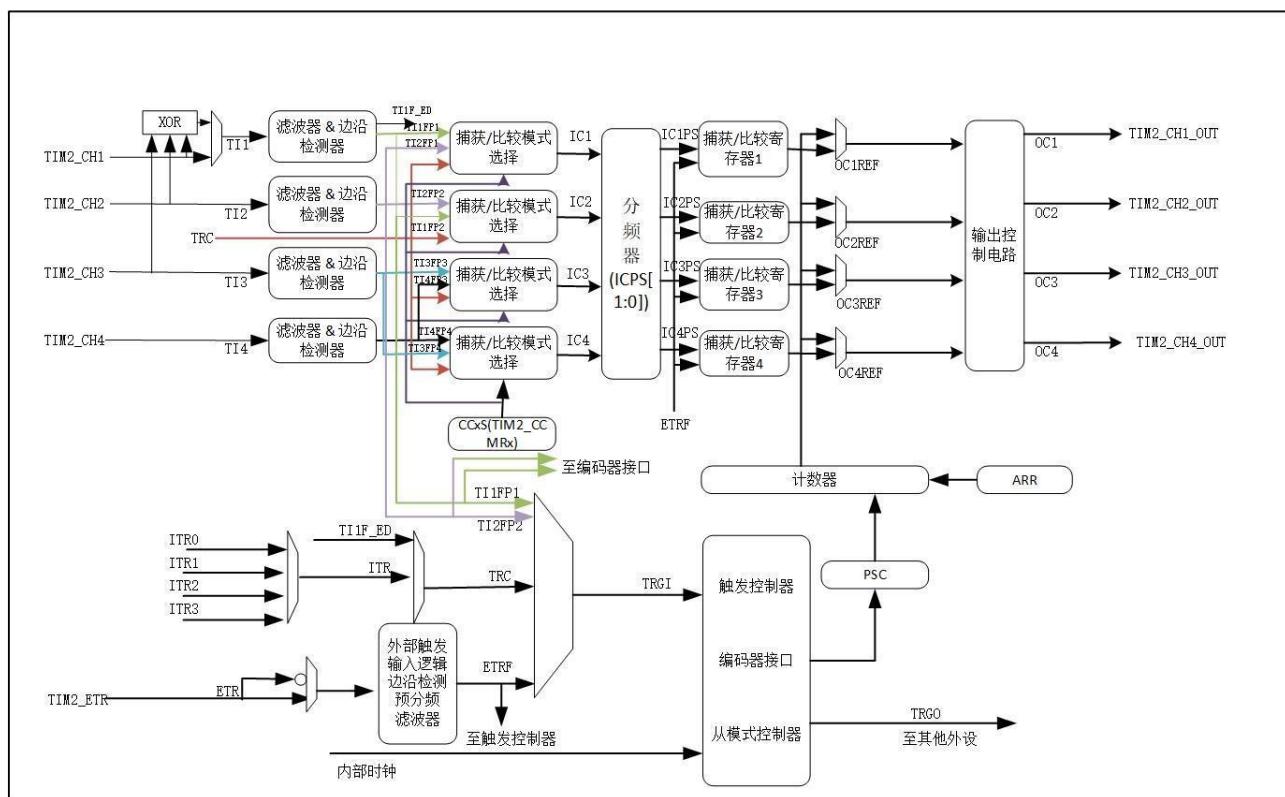


图 13-1 TIMx 结构图

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

13.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 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 位自动重装载计数器（计数方向：递增、递减、递增/递减）
- 输入捕获：输入信号的脉冲宽度、周期的测量
- 触发输入可以作为外部时钟或者逐周期管理
- 支持编码器、霍尔传感器等接口
- 4 个输出通道
- 比较输出（控制输出波形或指示定时器已经计时结束）
- PWM 输出（边沿对齐或中央对齐模式）
- 单脉冲输出
- 产生中断/DMA 请求的事件：更新事件、触发事件、输入捕获、比较输出
- 注：TIM5 不支持外部触发输入 ETR 功能。

13.4 功能描述

13.4.1 时钟

13.4.1.1 时钟选择

计数器的时钟源有以下几种：

- 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：

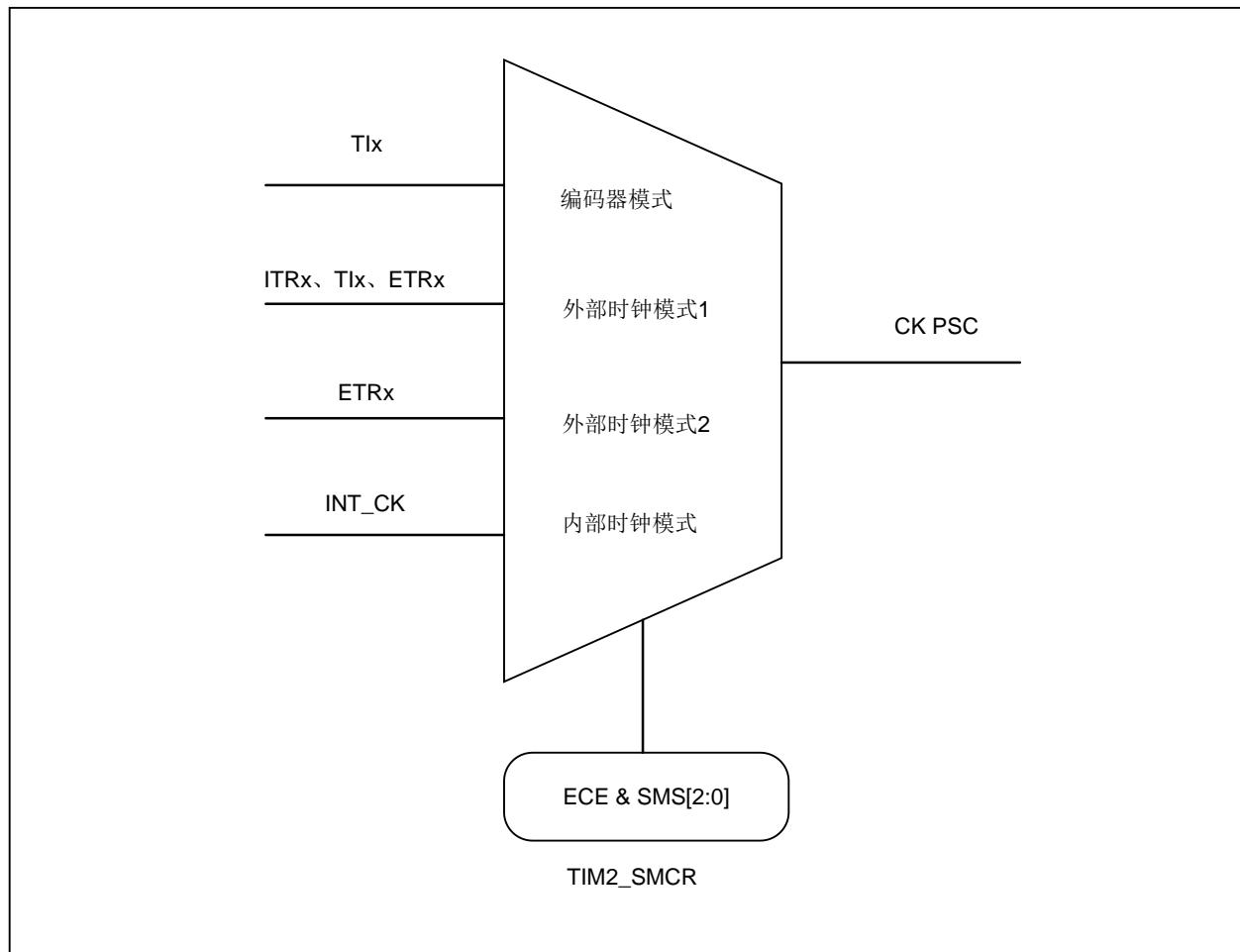


图 13-2 时钟选择

13.4.1.1.1 内部时钟源 (INT_CK)

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

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

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

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

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

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

当 `TI1` 出现有效边沿时, 计数器递增计数一次。`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。

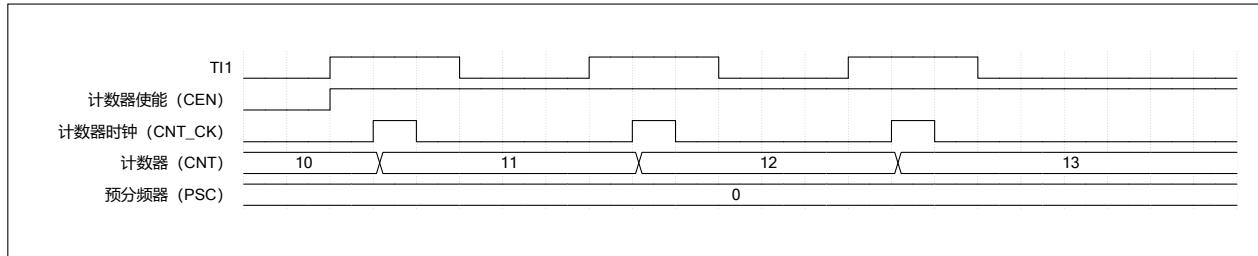


图 13-3 外部时钟模式 1 下的控制电路

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

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

例: `ETR` 的每 4 个下降沿计数一次, 递增计数, 具体配置如下:

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

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

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

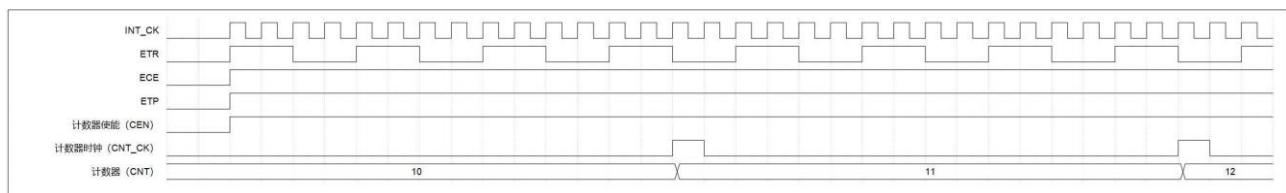


图 13-4 外部时钟模式 2 下的控制电路

13.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

13.4.1.2 时基单元

`TIMx` 的时基单元主要包括: 计数器寄存器 (`TIMx_CNT`)、预分频器寄存器 (`TIMx_PSC`) 和自动预装载寄存器 (`TIMx_ARR`)。

计数单元由一个 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，在下一次更新事件时生效。

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 **TIMx_CR1** 寄存器的 APRE 位选择写入 ARR 寄存器的值立即生效或发生更新事件时载入影子寄存器。

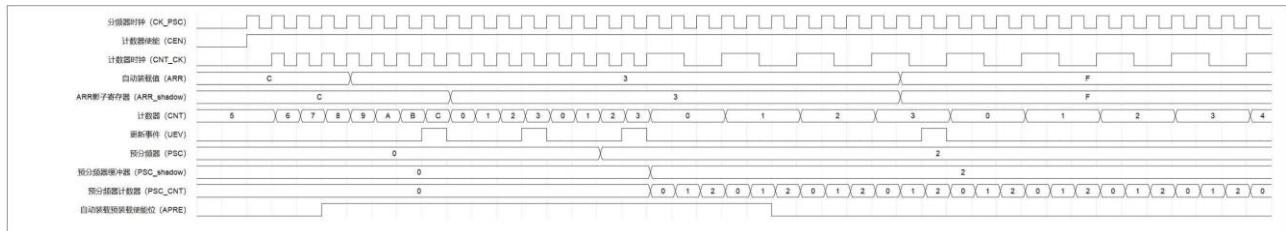


图 13-5 自动预装载

13.4.1.3 计数模式

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

13.4.1.3.1 递增计数模式

配置 **TIMx_CR1** 寄存器 CMS=0, DIR=0，选择递增计数模式。

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

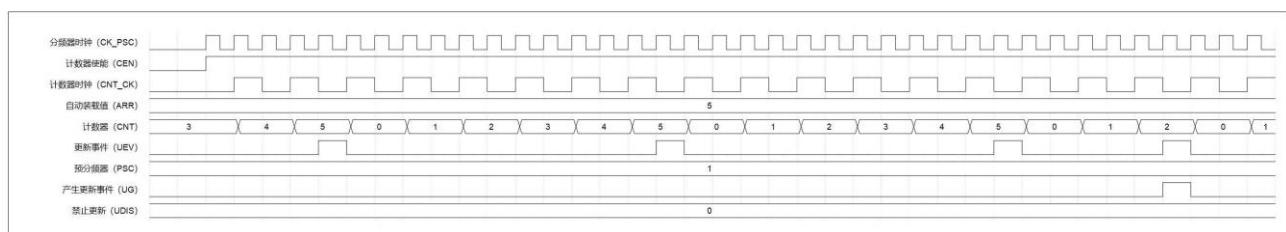


图 13-6 递增计数模式 (UDIS=0)

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

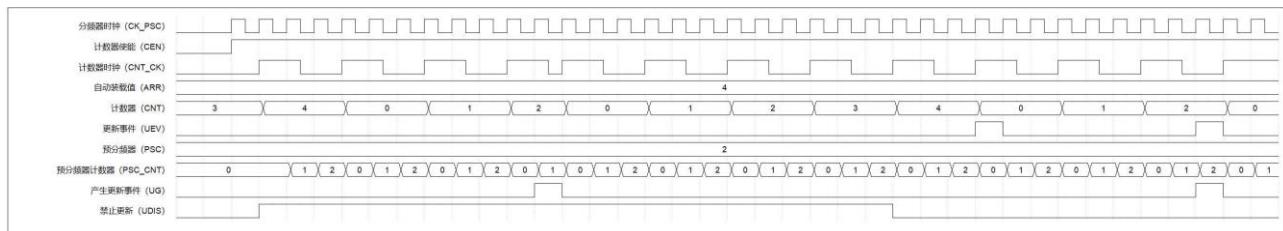


图 13-7 递增计数模式 (UDIS=1 禁止产生更新事件)

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。

13.4.1.3.2 递减计数模式

配置 TIMx CR1 寄存器的 CMS=0, DIR=1, 选择递减计数模式。

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

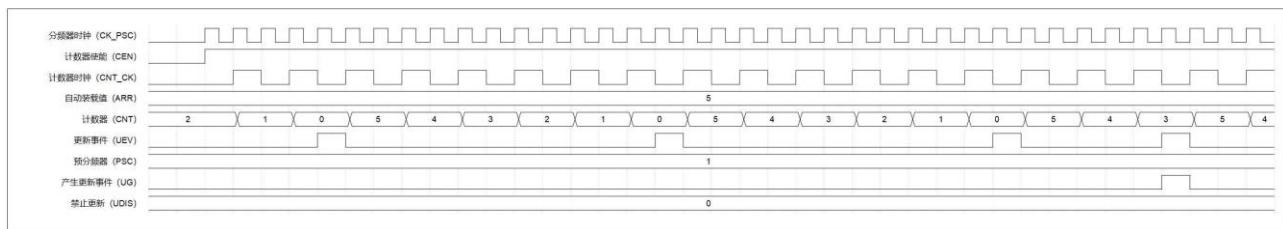


图 13-8 递减计数模式 (UDIS=0)

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

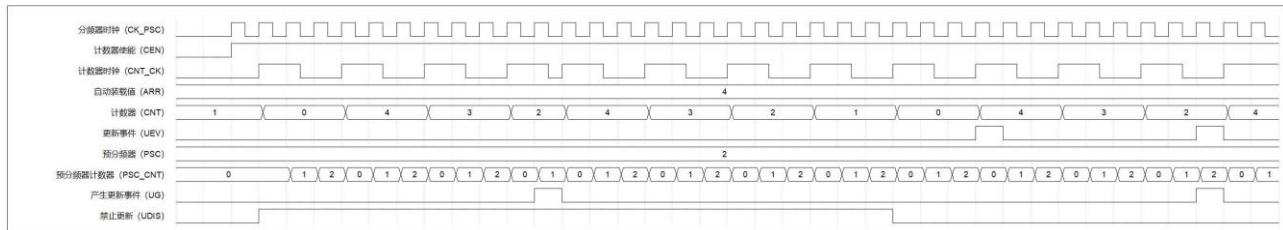


图 13-9 递减计数模式 (UDIS=1 禁止产生更新事件)

13.4.1.3.3 中央计数模式（递增/递减计数模式）

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

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

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

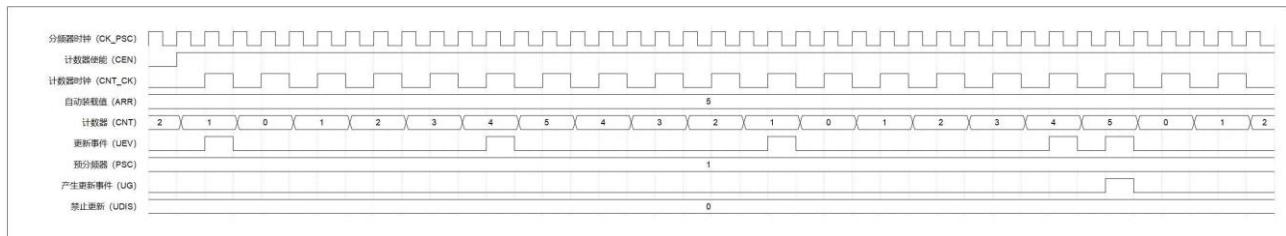


图 13-10 中央计数模式 (UDIS=0)

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

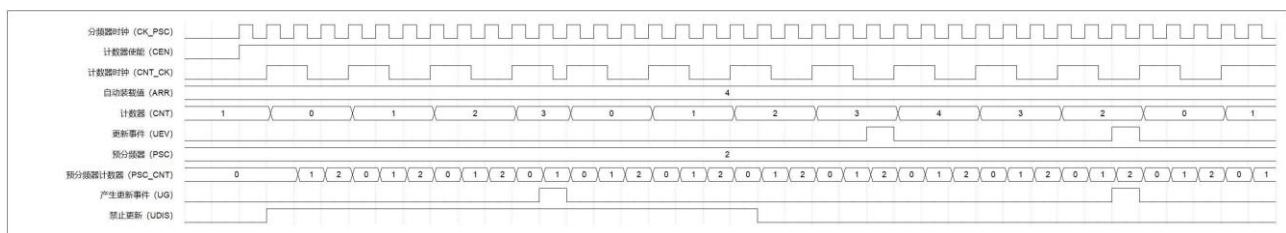


图 13-11 中央计数模式 (UDIS=1 禁止产生更新事件)

13.4.2 输入捕获

13.4.2.1 输入捕获

输入捕获部分包括数字滤波器、多路复用、预分频器等, 其结构如下图所示:

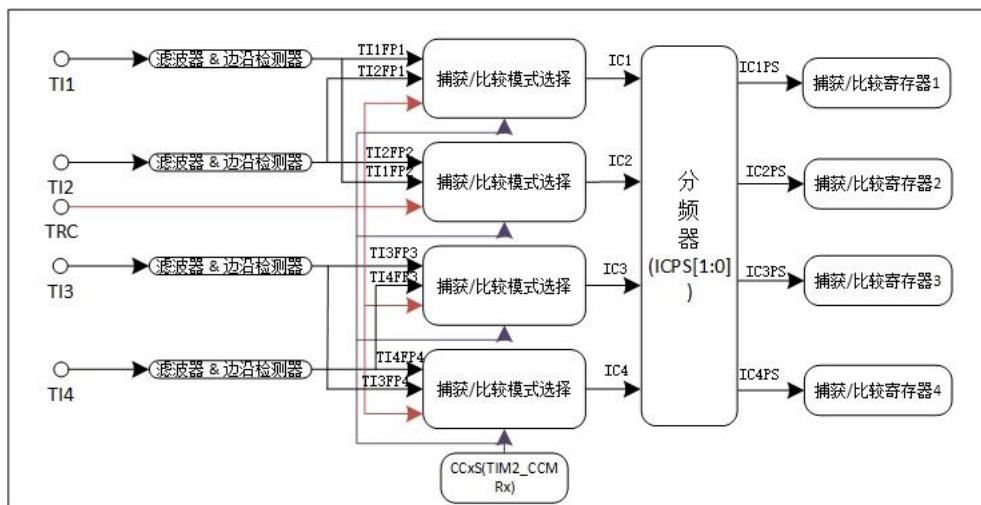


图 13-12 TIMx 输入捕获结构图

通过配置 **TIMx_CCMRx** 寄存器的 **ICxF**, 可以设置数字滤波器的滤波宽度 (滤波器的采样频率及数字滤波宽度如下表所示), 当数字滤波器的输入信号宽度大于滤波宽度时, 输入信号有效; 数字滤波器对输入

引脚 TIx 的输入信号采样后，产生一个滤波后的信号 TIxF ，然后通过极性可选的边沿检测器，产生一个有效信号 TIxFPx ，这个信号可以作为从模式控制器的触发输入信号，同时该信号经过预分频器产生一个信号 ICxPS ，用于触发输入捕获事件。

表 13-1 数字滤波器宽度与 ICxF 的对应关系表

$\text{IC1F}[3: 0]$	采样频率和滤波宽度	$\text{IC1F}[3: 0]$	采样频率和滤波宽度
0000	无滤波器，以 f_{DTS} 采样	1000	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=6$
0001	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=2$	1001	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/8, N=8$
0010	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=4$	1010	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=5$
0011	采样频率 $f_{\text{sampling}}=f_{\text{INT_CK}}, N=8$	1011	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=6$
0100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/2, N=6$	1100	采样频率 $f_{\text{sampling}}=f_{\text{DTS}}/16, N=8$
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 ICx 上的有效边沿后，计数器的当前值被锁存到对应的影子寄存器上，再复制到对应的捕获比较寄存器中。当开启了中断或 DMA 使能，发生捕获事件时，将产生相应的中断或 DMA 请求。发生捕获事件时，会将状态寄存器(TIMx_SR)中的捕获标志位 CCxIF 置 1，通过配置 $\text{CCxIF}=0$ 或读取 TIMx_CCRx 中的数据，清除 CCxIF 标志位。当 CCxIF 未被清零时，发生输入捕获事件，重复捕获标志位 CCxOF 将会被置 1，通过配置 $\text{CCxOF}=0$ ，可以清除 CCxOF 标志位。

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

1. 配置 TIMx_CCMR1 寄存器的 $\text{CC1S}=01$ ， CC1 通道被配置为输入， IC1 映射在 TI1 上。
2. 配置 TIMx_CCMR1 寄存器的 $\text{IC1F}[3: 0]$ ，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 $\text{CC1P}=0$ ，选择捕获发生在 TI1 信号的上升沿。
4. 配置 TIMx_CCMR1 寄存器的 $\text{IC1PSC}[1:0]$ ，选择预分频系数。
5. 配置 TIMx_CCER 寄存器的 $\text{CC1E} = 1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 TIMx_DIER 寄存器的 $\text{CC1IE}=1$ ，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 $\text{CC1DE}=1$ ，允许捕获/比较通道 1 的 DMA 请求。

注：

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

13.4.2.2 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 IC_x 信号被映射至同一个 TI_x 输入。

- 配置从模式为复位模式，将其中一路 TI_xFP 作为触发输入信号。

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

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

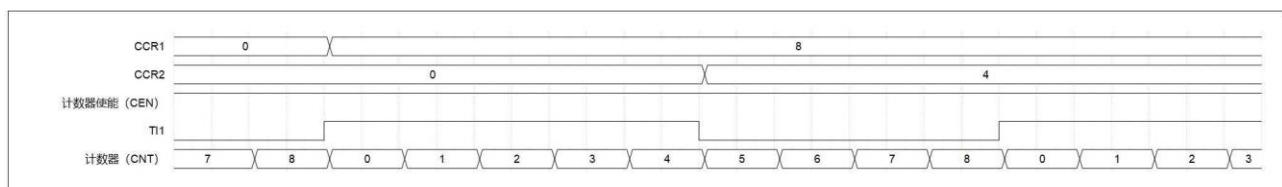


图 13-13 PWM 输入模式时序

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

13.4.3 比较输出

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

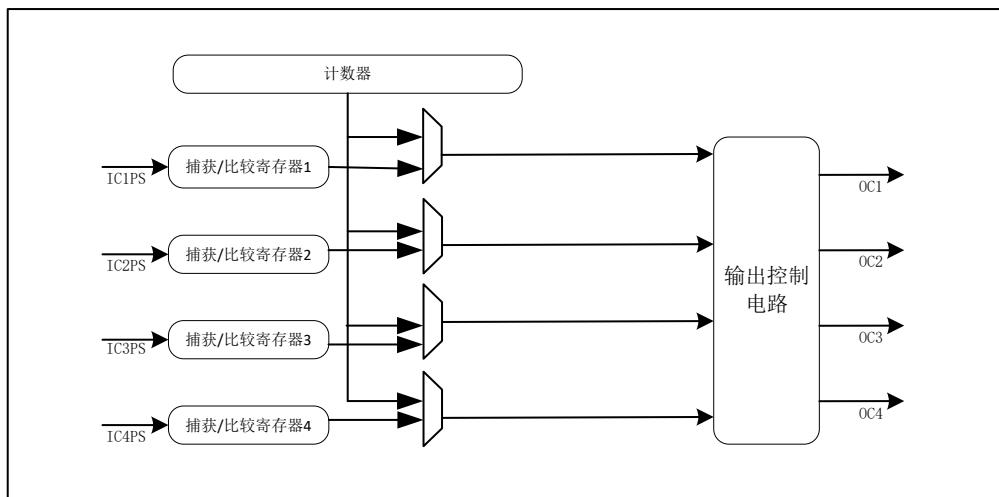


图 13-14 比较输出部分结构图

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。

13.4.3.1 强制输出

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

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

13.4.3.2 比较输出

比较输出模式下，当计数器与捕获比较寄存器值相同时，可以根据 TIMx_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：

- ◆ $\text{OCxM} = 000$: OCx 信号保持它的电平
- ◆ $\text{OCxM} = 001$: OCx 信号被设置成有效电平
- ◆ $\text{OCxM} = 010$: OCx 信号被设置成无效电平
- ◆ $\text{OCxM} = 011$: OCx 信号进行翻转

2. 匹配时状态寄存器中的标志位置 1 (TIMx_SR 寄存器中的 CCxIF 位)。

3. 当配置了 `TIMx_DIER` 寄存器中的 `CCxIE =1`, 匹配时则产生一个中断。
4. 当配置了 `TIMx_DIER` 寄存器中的 `CCxDE =1`, 匹配时则产生一个 DMA 请求 (仅适用于有内置 DMA 的产品)。

比较输出模式也可以用来输出一个单脉冲 (单脉冲输出模式)。

例如, 通道 1 的比较输出模式的配置步骤如下:

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

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

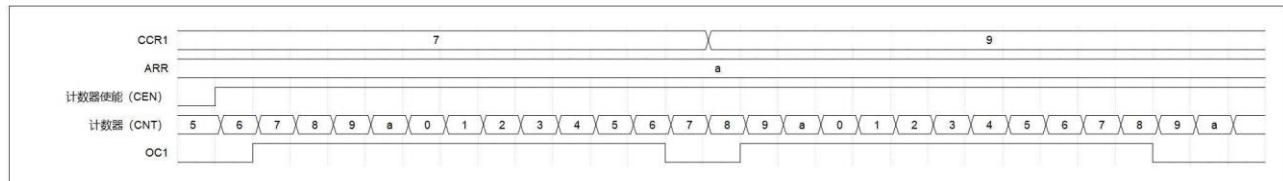


图 13-15 比较输出模式, `OC1` 信号在匹配时翻转

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

13.4.3.3 PWM 输出

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

配置与通道 x 对应的 `TIMx_CCMRx` 寄存器的 `OCxM=110` 或 `OCxM=111`, 选择通道 x 进入 PWM 模式 1 或 PWM 模式 2。PWM 模式下, 计数器和 `CCRx` 会一直进行比较, 根据配置和比较结果, 通道 x 输出不同的信号, 因此 `TIMx` 可以产生 4 个同频率下独立占空比的 PWM 输出信号。PWM 模式下可开启 `TIMx_CCRx` 的预装载功能和 `TIMx_ARR` 寄存器的预装载功能。写入 `TIMx_CCRx` 预装载寄存器和

TIMx_ARR 预装载寄存器的值在发生下个更新事件时，才会生效，载入相应的影子寄存器。PWM 模式下，使能计数器前设置 **TIMx_EGR** 的 **UG=1**，产生更新事件用于初始化所有的寄存器。

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

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

13.4.3.3.1 PWM 边沿对齐模式——递增计数模式

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

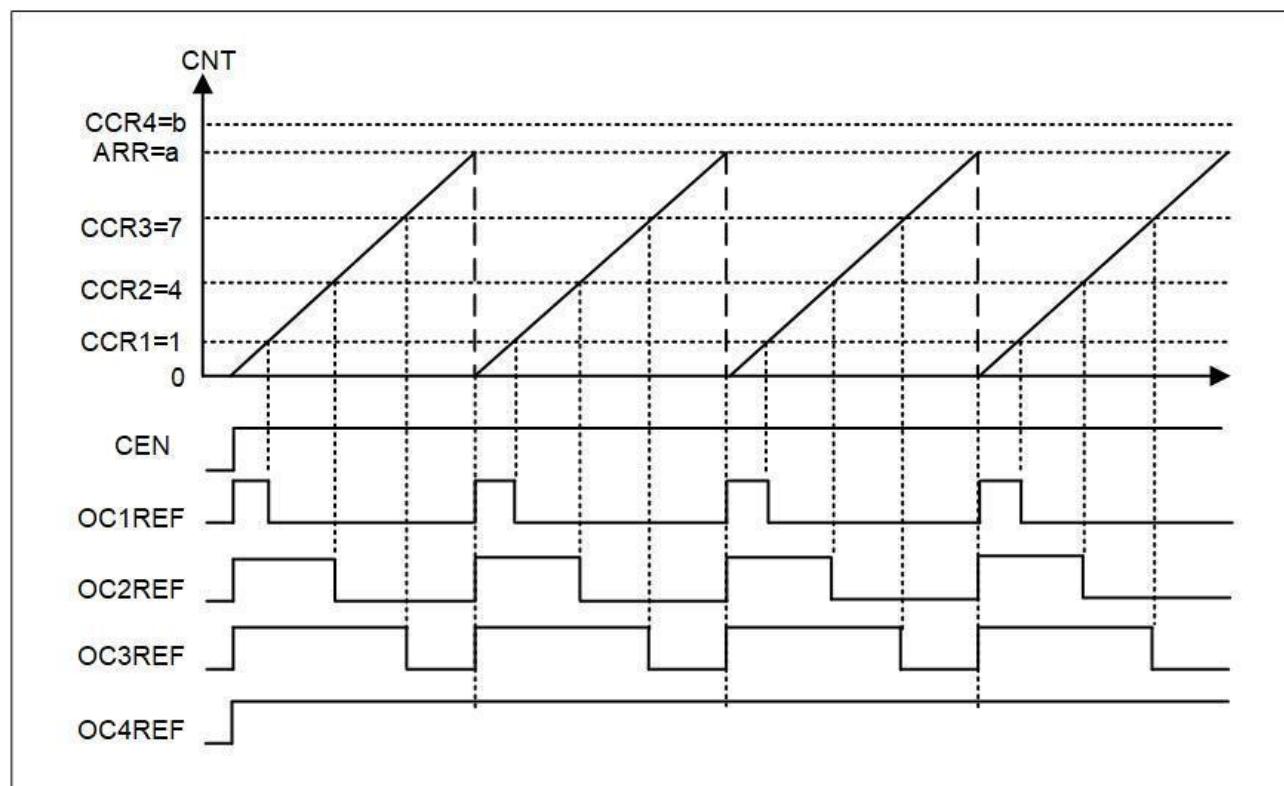


图 13-16 边沿对齐递增计数时 PWM 模式 1 的波形

13.4.3.3.2 PWM 边沿对齐模式——递减计数模式

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

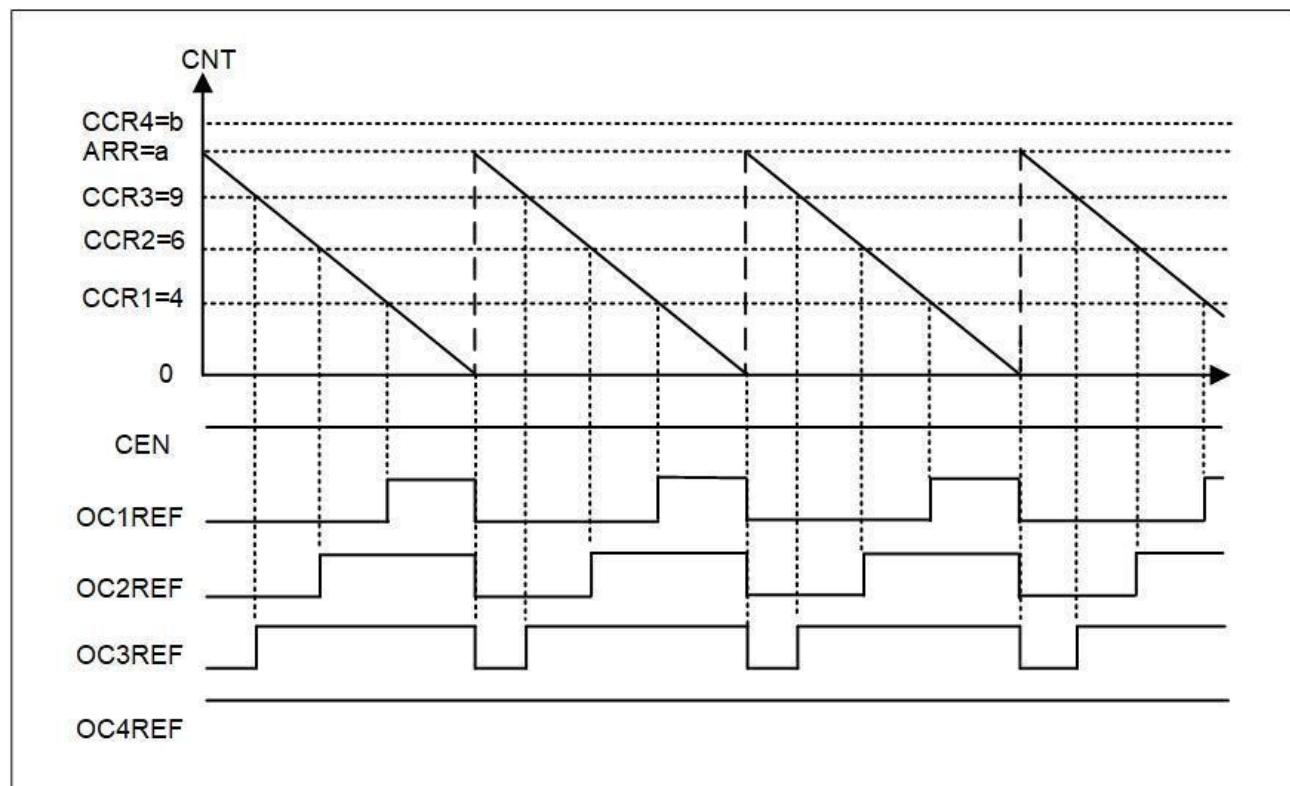


图 13-17 边沿对齐递减计数时 PWM 模式 1 的波形

13.4.3.3.3 PWM 中央对齐模式

首先配置 TIMx 计数器为中央对齐计数模式, 配置 TIMx_CCMRx 寄存器的 $\text{CCxS}=00$, 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 ($\text{CMS}=01$)、在计数器递增计数时被设置 ($\text{CMS}=10$)、或在计数器递增或递减计数时被设置 ($\text{CMS}=11$)。下图为 $\text{CCR1}=4$, $\text{CCR2}=6$, $\text{CCR3}=9$, $\text{CCR4}=b$, $\text{ARR}=a$ 时中央对齐 PWM 模式 1 的波形实例。

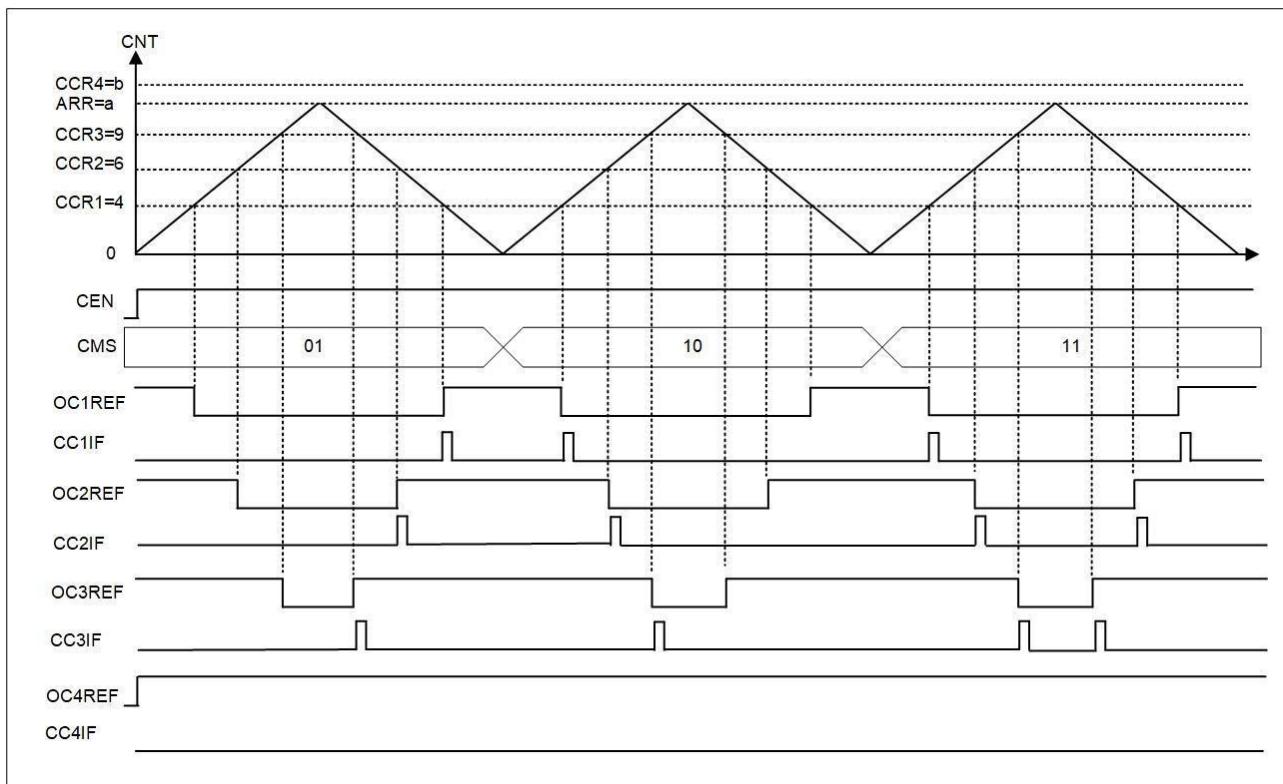


图 13-18 中央对齐 PWM 模式 1 的波形

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 > TIMx_ARR，计数器会继续递增计数。直接写入 0 或 ARR，会立即更新计数方向，但不会产生更新事件。
- 建议使用中央对齐模式时，在启动计数器之前配置 TIMx_EGR 寄存器的 UG=1，产生一个软件更新，更新所有寄存器，启动计数器后不要修改计数器的值。

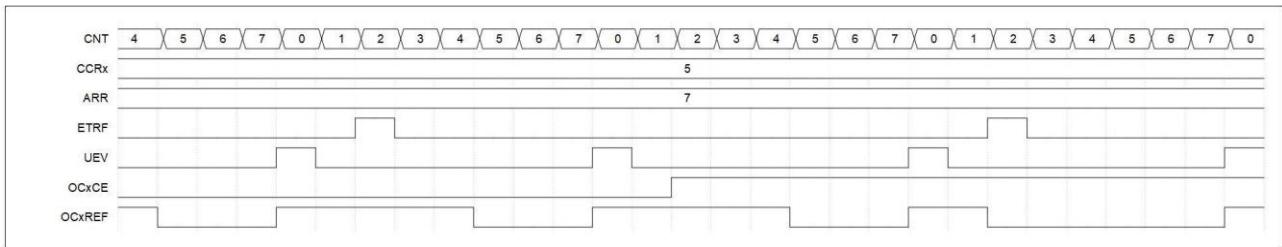
13.4.3.4 外部事件清除 OCxREF

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

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

1. 配置 TIMx_SMCR 寄存器的 ETPS[1: 0]=00，关闭外部触发预分频。
2. 配置 TIMx_SMCR 寄存器 ECE=0，禁用外部时钟模式 2。
3. 配置 TIMx_SMCR 寄存器 ETF[3: 0]和 ETP，配置 ETR 信号的触发极性和滤波宽度。

下图显示了当 ETR 输入变化触发 ETRF 为高时，对应不同 OCxCE 的值，OCxREF 信号的动作 (PWM 模式)。

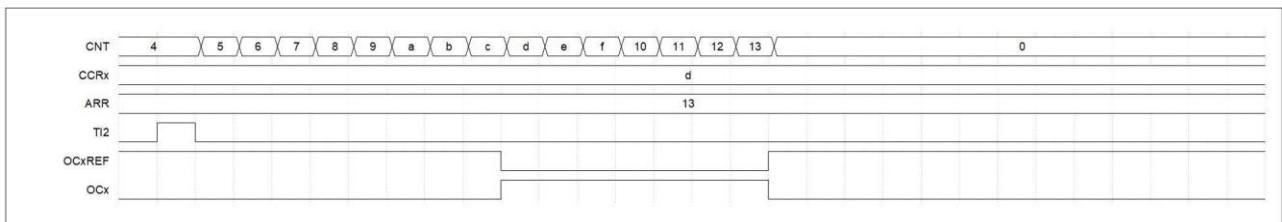


13.4.3.5 单脉冲输出

单脉冲模式 (OPM) 下，计数器响应一个激励，产生一个脉宽可调的脉冲。配置 TIMx_CR1 寄存器的 OPM=1，选择单脉冲模式，触发信号有效沿或配置 CEN=1 都可以启动计数器，直到下一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 ≤ ARR。
- 递减计数方式：计数器 CNT > CCRx。



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。

假定 TI2FP2 作为触发源：

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

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

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

1. 配置 TIMx_CCMR1 寄存器 OC1M = 111，选择 PWM 模式 2。
2. 配置 TIMx_CCER 寄存器 CC1P = 1，输出低电平有效。
3. 配置 TIMx_CCMR1 中 OC1PE = 1 和 TIMx_CR1 寄存器中 ARPE=1，使能预装载寄存器。
4. 配置 TIMx_CCR1 寄存器和 TIMx_ARR 寄存器。

5. 配置 **TIMx_EGR** 寄存器 **UG=1** 产生一个更新事件。

6. 等待在 **TI2** 上的一个外部触发事件。

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

13.4.3.5.1 OCx 快速使能

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

13.4.4 从模式

13.4.4.1 编码器接口

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

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

注: 编码器模式不支持外部时钟模式 2。

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

表 13-2 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	
计数模式	相对电平 (TI1FP1 相对于 TI2 , TI2FP2 相对于 TI1)	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	
编码器接口模式 1 (只在 TI2 计数)	低电平	-	-	递减计数	递增计数
编码器接口模式 2 (只在 TI1 计数)	高电平	递减计数	递增计数	-	-
编码器接口模式 2 (只在 TI1 计数)	低电平	递增计数	递减计数	-	-
编码器接口模式 3 (在 TI1 和 TI2 计数)	高电平	递减计数	递增计数	递增计数	递减计数
编码器接口模式 3 (在 TI1 和 TI2 计数)	低电平	递增计数	递减计数	递减计数	递增计数

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

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

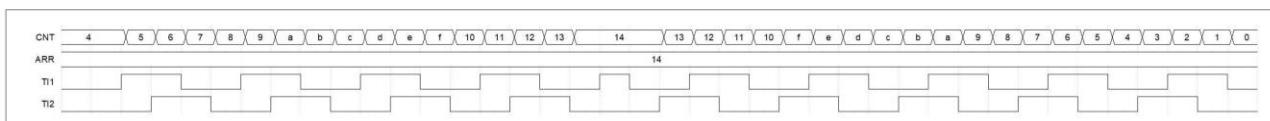


图 13-21 编码器模式下的计数器时序图

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

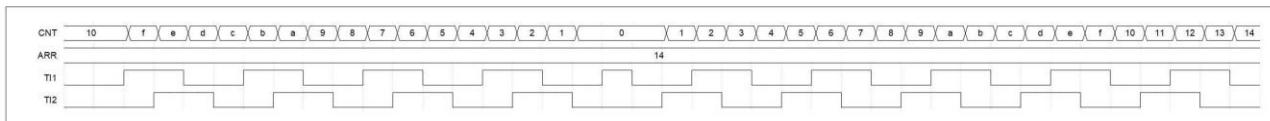


图 13-22 IC1FP1 反相编码器接口模式时序图

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。

13.4.4.2 复位模式

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

例如, `TI2` 输入端的下降沿触发计数器重启:

1. 配置 `TIMx_CCMR1` 寄存器的 `CC2S=01`, `CC2` 通道被配置为输入模式; `IC2` 映射在 `TI2` 上, 配置 `TIMx_CCER` 寄存器的 `CC2P=1`, 检测下降沿。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS = 100`, 从模式选择复位模式; 配置 `TIMx_SMCR` 寄存器的 `TS = 110`, 选择滤波后的定时器输入 2 (`TI2FP2`) 作为同步计数器的触发输入; 配置 `TIMx_SMCR` 寄存器的 `ETF=0000`, 不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`, 选择计数方向为递增计数; 配置 `PSC=0`, 不分频; 配置 `CEN=1`, 使能计数器。

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

下图为复位模式下 `TIMx_ARR = 0x13` 的时序图。

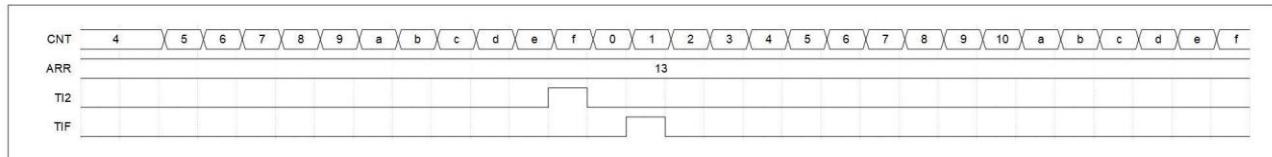


图 13-23 复位模式的控制时序图

13.4.4.3 门控模式

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

例如, 计数器只在 `TI1` 为高时计数:

1. 配置 `TIMx_CCMR1` 寄存器的 `CC1S=01`, `CC1` 通道被配置为输入模式, `IC1` 映射在 `TI1` 上; 配置 `TIMx_CCER` 寄存器的 `CC1P=0`, 检测 `TI1` 上的高电平。
2. 配置 `TIMx_SMCR` 寄存器的 `SMS=101`, 从模式选择为门控模式; 配置 `TIMx_SMCR` 寄存器的 `TS=101`, 选择滤波后的定时器输入 1 (`TI1FP1`) 作为同步计数器的触发输入; 配置 `TIMx_SMCR` 寄存器的 `ETF=0000`, 不使用数字滤波器。
3. 配置 `TIMx_CR1` 寄存器的 `DIR=0`, 选择计数方向为递增计数; 配置 `PSC=0`, 不分频; 配置 `CEN=1`, 使能计数器。

计数器的时钟源由内部时钟提供, 当检测到 `TI1` 的高电平, 计数器开始计数, 当 `TI1` 为低电平时, 计

数器停止计数。计数器停止会将 TIF 置 1。

下图为门控模式下 TIMx_ARR=0xf 的时序图。

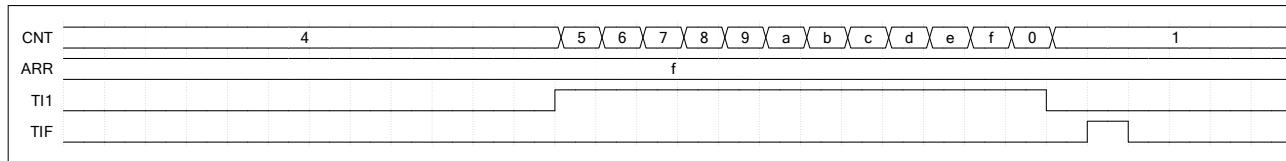


图 13-24 门控模式下的控制时序图

13.4.4.4 触发模式

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

例如，计数器在 TI1 输入的上升沿开始计数：

1. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入模式，IC1 映射在 TI1 上，配置 TIMx_CCER 寄存器的 CC1P=0，检测上升沿。
2. 配置 TIMx_SMCR 寄存器的 SMS = 110，从模式选择为触发模式；配置 TIMx_SMCR 寄存器的 TS=101，选择滤波后的定时器输入 1 (TI1FP1) 作为计数器的触发输入；配置 TIMx_SMCR 寄存器的 ETF=0000，不使用数字滤波器。
3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。

计数器的时钟源由内部时钟提供，当检测到 TI1 的上升沿，计数器开始计数。

下图为触发模式下 TIMx_ARR=0xf 的时序图。

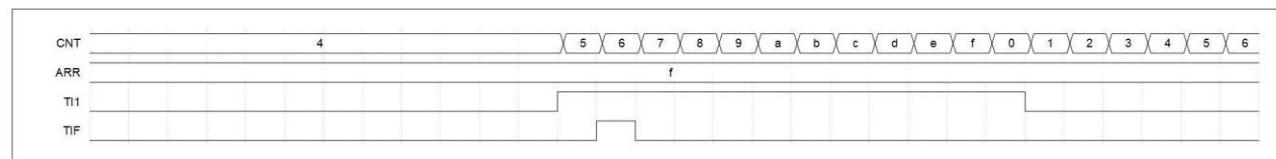


图 13-25 触发器模式下的控制时序图

13.4.4.5 外部时钟模式 2+从模式

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

例如，从模式选择触发模式，计数器在 ETR 的每一个上升沿计数一次：

1. 配置 TIMx_SMCR 寄存器的 ETF = 0000，不使用数字滤波器；配置 TIMx_SMCR 寄存器的 ETPS = 00，关闭预分频；配置 TIMx_SMCR 寄存器的 ETP = 0，检测 ETR 的上升沿；配置 TIMx_SMCR 寄存器的 ECE = 1，使能外部时钟模式 2。
2. 配置 TIMx_CCMR1 寄存器的 CC1S=01，CC1 通道被配置为输入，IC1 映射在 TI1 上作为输入捕获源；配置 TIMx_CCER 寄存器的 CC1P=0，选择上升沿有效。

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

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

计数器在 **TI1** 的上升沿开始计数, 并将 **TIF** 置 1。ETR 信号的上升沿和计数器实际计数时钟间的延时取决于 **ETR** 输入端的同步电路设计。

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

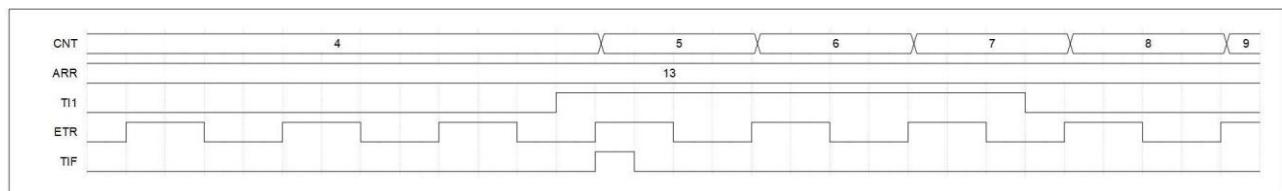


图 13-26 外部时钟模式 2+从模式(触发模式)控制时序图

13.4.5 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。

13.4.6 定时器异或

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

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

1. 配置 **TIMx_CR2** 寄存器的 **TI1S=1**, 配置定时器的三个输入经异或后连接到 **TI1** 输入通道。
2. 配置 **TIMx_CCMR1** 寄存器的 **CC1S=01**, **CC1** 通道被配置为输入, **IC1** 映射在 **TI1** 上。
3. 配置 **TIMx_CCMR1** 寄存器的 **IC1F[3: 0]**, 配置数字滤波器的滤波宽度(按需配置)。
4. 配置 **TIMx_CCER** 寄存器的 **CC1P=0**, 选择捕获发生在 **TI1** 信号的上升沿。
5. 配置 **TIMx_CCMR1** 寄存器的 **IC1PSC**, 选择预分频系数。
6. 配置 **TIMx_CCER** 寄存器的 **CC1E = 1**, 开启输入/捕获通道 1 的捕获使能。
7. 配置 **TIMx_CR1** 寄存器的 **CEN=1**, 启动计数器。

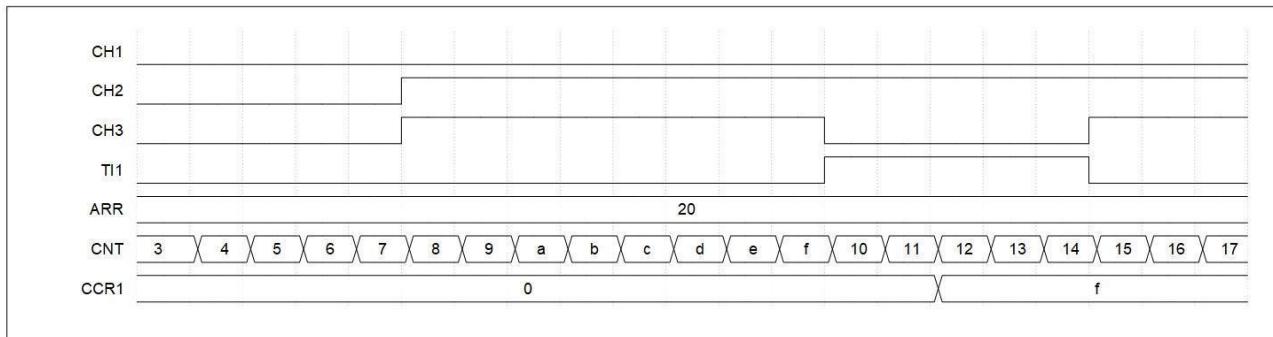


图 13-27 (TI1 异或输入) 输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

13.4.7 调试模式

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

13.4.8 中断

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

表 13-3 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

13.4.9 DMA

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

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

TIMx_DCR 寄存器的 DBL 位定义了 DMA 连续传送的长度，即传输寄存器数量。当对 TIMx_DMAR 进行读写操作时，定时器识别 DBL，确定需要传输的寄存器数量。TIMx_DCR 寄存器的 DBA 位定义了

DMA 传输的基地址，定义从 TIMx_CR1 寄存器地址开始的偏移量（00000 为 TIMx_CR1、00001 为 TIMx_CR2 …… 00110 为 TIMx_CCMR1 等）。

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

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

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

13.5 寄存器描述

表 13-4 TIMx 寄存器概览

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

Offset	Acronym	Register Name	Reset
0x4C	TIMx_DMAR	连续模式的 DMA 地址（仅适用于有内置 DMA 的芯片）	0x0000
0x50	TIMx_OR	输入选项寄存器	0x0000

13.5.1 TIMx_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	APRE	CMS	DIR	OPM	URS	UDIS	CEN		
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频 (clock division) 定义定时器时钟 (INT_CK) 频率与死区时间计数器、数字滤波器 (ETR, TIx) 所用的时钟之间的分频比例。 00: $t_{DTs} = t_{INT_CK}$ 01: $t_{DTs} = 2x t_{INT_CK}$ 10: $t_{DTs} = 4x t_{INT_CK}$ 11: 保留，不要使用这个配置
7	APRE	自动重装载预装载使能 (Auto-reload preload enable) 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 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。

Bit	Field	Description
3	OPM	单脉冲模式 (One pulse mode) 0: 禁止单脉冲模式, 在发生更新事件时, 计数器继续计数 1: 使能单脉冲模式, 在发生下一次更新事件或软件清除 CEN 位时, 计数器停止计数
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC、CCRx) 保持值不变。如果设置了 EGR_UG 位为 1, 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。

13.5.2 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.								TI1S	MMS			CCDS	Res.		
								rw	rw			rw			

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	TI1S	TI1 选择 (TI1 selection) 0: TIMx_CH1 管脚连到 TI1 输入 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入

Bit	Field	Description
6: 4	MMS	<p>主模式选择 (Master mode selection) 这些位控制 TRGO 信号的选择，用于选择在主模式下送到从定时器的同步信息：</p> <p>000: 复位 TIMx_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 1: 发生更新事件时，发送 CCx 的 DMA 请求 注：仅适用于有内置 DMA 的产品</p>
2: 0	Reserved	保留，必须保持复位值。

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

Bit	Field	Description
14	ECE	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2。 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2, ETRF 信号上的任意有效沿驱动计数器计数</p> <p>注 1: 仅适用于支持外部触发的产品。 注 2: 配置 ECE=1 与配置 SMS = 111 和 TS = 111 效果一样。 注 3: TS ≠ 111 时, 复位模式, 门控模式和触发模式可以与外部时钟模式 2 同时使用。 注 4: 同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入是 ETR。</p>
13: 12	ETPS	<p>外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须低于 TIMx 主时钟 PCLK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。</p> <p>00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8</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) 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 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> <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>比较器输出信号清除选择 (Output compare clear selection)</p> <p>在 PWM 模式下, 清除比较器输出</p> <p>0: 外部触发信号作为清除信号</p> <p>1: 比较器输出作为清除信号</p> <p>注: 仅适用于支持外部触发或有内置比较器的产品。</p>

Bit	Field	Description
2: 0	SMS	<p>从模式选择 (Slave mode selection) 当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。</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。 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 13-5 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	TIM5	TIM2	TIM3	TIM4
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM1	TIM3	TIM4	TIM8
TIM8	TIM1	TIM2	TIM4	TIM5

13.5.4 TIMx_DIER DMA/中断使能寄存器

偏移地址: 0x0C

复位值: 0x0000

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

Bit	Field	Description
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

13.5.5 TIMx_SR 状态寄存器

偏移地址: 0x10

复位值: 0x0000

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

Bit	Field	Description
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 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。

13.5.6 TIMx_EGR 事件产生寄存器

偏移地址: 0x14

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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, TIMx_SR 寄存器的 TIF = 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA, 由硬件自动清 0。
5	Reserved	保留，必须保持复位值。
4	CC4G	产生捕获/比较 4 事件 (Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	产生捕获/比较 3 事件 (Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	产生捕获/比较 2 事件 (Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	产生通道 1 捕获/比较事件 (Capture/Compare 1 generation) 该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0。 0: 无动作 1: 通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出: CC1IF 置 1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。 若通道 CC1 配置为输入: 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。

13.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址: 0x18

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	CC2S		OC1CE	OC1M			OC1PE	OC1FE	CC1S	
IC2F			IC2PSC			CC2S		IC1F				IC1PSC			CC1S
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CC_xS 定义。该寄存器 CC_xS 以外其它位的作用在输入模式和输出模式下不同。OC_{xx} 描述了通道在输出模式下的功能，IC_{xx} 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能（Output compare 2 clear enable） 参考 OC1CE 的描述。
14: 12	OC2M	通道 2 比较输出模式（Output compare 2 mode） 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能（Output compare 2 preload enable） 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能（Output compare 2 fast enable） 参考 OC1FE 的描述。
9: 8	CC2S	通道 2 捕获/比较选择（Capture/Compare 2 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00: 通道 2 被配置为输出 01: 通道 2 被配置为输入，IC2 映射在 TI2 上 10: 通道 2 被配置为输入，IC2 映射在 TI1 上 11: 通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时（由 TIMx_SMCR 寄存器的 TS 位选择）
7	OC1CE	通道 1 比较输出清 0 使能（Output compare 1 clear enable） 0: OC1REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入高电平时，OC1REF 清零

Bit	Field	Description
6: 4	OC1M	<p>通道 1 比较输出模式 (Output compare 1 mode) 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。</p> <p>000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为高电平。</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为低电平。</p> <p>011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。</p> <p>100: 强制为低。强制 OC1REF 为低电平。</p> <p>101: 强制为高。强制 OC1REF 为高电平。</p> <p>110: PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平，否则为高电平。</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；在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。</p> <p>注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。</p>
3	OC1PE	<p>通道 1 比较输出预装载使能 (Output compare 1 preload enable) 0: 禁止 TIMx_CCR1 寄存器的预装载功能，写入 TIMx_CCR1 寄存器的数值立即生效。</p> <p>1: 开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时生效。</p> <p>注：若该位置 1，在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1)，是否设定预装载寄存器无影响；其它情况下，需要设定预装载寄存器，否则后续动作不确定。</p>
2	OC1FE	<p>通道 1 比较输出快速使能 (Output compare 1 fast enable) 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>

Bit	Field	Description
1: 0	CC1S	<p>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：</p> <ul style="list-style-type: none"> 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 TIMx_SMCR 寄存器的 TS 位选择)

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) 参考 IC1F 的描述</p>
11: 10	IC2PSC	<p>输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 的描述</p>
9: 8	CC2S	<p>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：</p> <ul style="list-style-type: none"> 00: 通道 2 被配置为输出 01: 通道 2 被配置为输入，IC2 映射在 TI2 上 10: 通道 2 被配置为输入，IC2 映射在 TI1 上 11: 通道 2 被配置为输入，IC2 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC1F	<p>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。</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) 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。</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) 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 TIMx_SMCR 寄存器的 TS 位选择)</p>

13.5.8 TIMx_CCMR2 捕获/比较模式寄存器 2

偏移地址: 0x1C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M		OC4PE	OC4FE	CC4S		OC3CE	OC3M		OC3PE	OC3FE	CC3S			
IC4F		IC4PSC		CC4S		IC3F		IC3PSC		CC3S					
rw	rw		rw	rw	rw		rw	rw		rw	rw	rw			

比较输出模式:

Bit	Field	Description
15	OC4CE	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 TIMx_SMCR 寄存器的 TS 位选择)
7	OC3CE	通道 3 比较输出清 0 使能 (Output compare 3 clear enable) 0: OC3REF 不受 ETR 输入的影响 1: 当检测到 ETR 输入高电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	<p>通道 3 比较输出模式 (Output compare 3 mode) 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。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用</p> <p>001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为高电平</p> <p>010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为低电平</p> <p>011: 匹配时翻转。当 TIMx_CCR3=TIMx_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 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平；在递减计数时，当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平。</p> <p>111: PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平；在递减计数时，当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平。</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 TIMx_CCR3 寄存器的预装载功能，写入 TIMx_CCR3 寄存器的数值立即生效</p> <p>1: 开启 TIMx_CCR3 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR3 的预装载值在更新事件到来时生效</p> <p>注：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM= 1)，无需设定预装载寄存器，其它情况下，需要设定预装载寄存器，否则后续动作不确定。</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) 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> <ul style="list-style-type: none"> 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入，IC3 映射在 TI3 上 10: 通道 3 被配置为输入，IC3 映射在 TI4 上 11: 通道 3 被配置为输入，IC3 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

输入捕获模式：

Bit	Field	Description
15: 12	IC4F	<p>输入捕获 4 滤波器 (Input capture 4 filter) 参考 IC3F 的描述</p>
11: 10	IC4PSC	<p>输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述</p>
9: 8	CC4S	<p>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：</p> <ul style="list-style-type: none"> 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入，IC4 映射在 TI4 上 10: 通道 4 被配置为输入，IC4 映射在 TI3 上 11: 通道 4 被配置为输入，IC4 映射在 TRC 上，此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC3F	<p>通道 3 输入捕获滤波器 (Input capture 3 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。</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	IC3PSC	<p>通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 CC3E=0 (TIMx_CCER 寄存器中) 时, 预分频器复位。</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	CC3S	<p>通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入:</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 TIMx_SMCR 寄存器的 TS 位选择)</p>

13.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20

复位值: 0x0000

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

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 13-6 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

13.5.10 TIMx_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)

13.5.11 TIMx_PSC 预分频器

偏移地址：0x28

复位值：0x0000

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

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

13.5.12 TIMx_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。

13.5.13 TIMx_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	<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 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。</p>

13.5.14 TIMx_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	<p>通道 2 捕获/比较的值 (Capture/Compare 2 value)</p> <p>参考 CCR1 的描述。</p>

13.5.15 TIMx_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。

13.5.16 TIMx_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。

13.5.17 TIMx_DCR DMA 控制寄存器

偏移地址: 0x48

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 13.4.9 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 TIMx_DMAR 寄存器的第一个地址。DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移值： 00000: TIMx_CR1 00001: TIMx_CR2 00010: TIMx_SMCR

13.5.18 TIMx_DMAR 连续模式 DMA 地址寄存器

偏移地址: 0x4C

复位值: 0x0000

注: 本寄存器仅适用于有内置 DMA 的产品, 详见 13.4.9 DMA 章节。

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

Bit	Field	Description
15: 0	DMAB	DMA 连续传送地址寄存器 (DMA address register for burst accesses) 对 TIMx_DMAR 寄存器的读写操作会导致对以下地址所在寄存器的存取操作: TIMx_CR1 地址 + DBA + DMA 索引, 其中 TIMx_CR1 地址是 TIMx_CR1 寄存器所在的地址, DBA 是 TIMx_DCR 寄存器中定义的基地址, DMA 索引是 DMA 自动控制的偏移量, 它取决于 TIMx_DCR 寄存器中定义的 DBL 值。

13.5.19 TIMx_OR 输入选项寄存器

偏移地址: 0x50

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						TI4_RMP		Res.			ETR_RMP				
						rw	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	<p>Timer5 TI4 复用 00: Timer5 CH4 GPIO 输入 01: LSI 时钟输入 10: LSE 时钟输入 11: -</p> <p>Timer2 TI4 复用 00 : CH4 GPIO 输入 其他保留</p>
5:3	Reserved	保留, 必须保持复位值。
2:0	ETR_RMP	<p>Timer2 ETR 复用 00: Timer2 ETR GPIO 输入 其他保留</p> <p>Timer5 无 ETR 功能</p>

14 TIM6/7 基本定时器

TIM6、TIM7 具有相同的功能，本章节统一表述为 TIMx，部分示意图以 TIM6 为例展示。

14.1 简介

TIMx 由一个 16 位可实时编程预分频器和一个 16 位自动重装载计数器组成，可以为用户提供便捷的计数定时功能，计数器时钟由预分频器分频得到。

14.2 功能框图

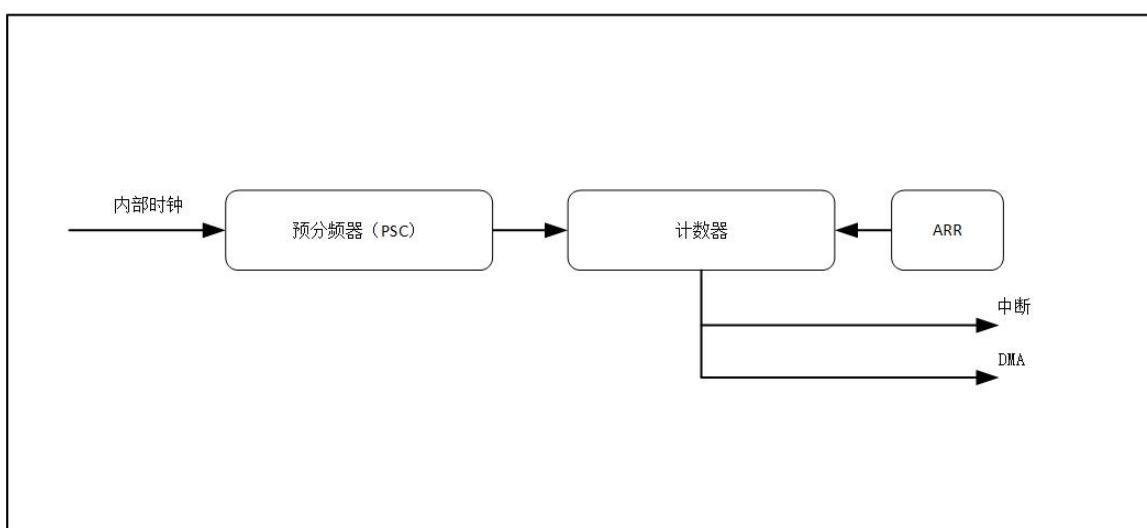


图 14-1 TIMx 结构图

上图为 TIMx 的结构框图。

14.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 16 位自动重装载计数器（计数方向：递增）
- 产生中断/DMA 请求的事件：更新事件

14.4 功能描述

14.4.1 时钟

14.4.1.1 时钟选择

计数器的时钟由内部时钟 (INT_CK) 提供。

14.4.1.2 时基单元

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

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

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

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

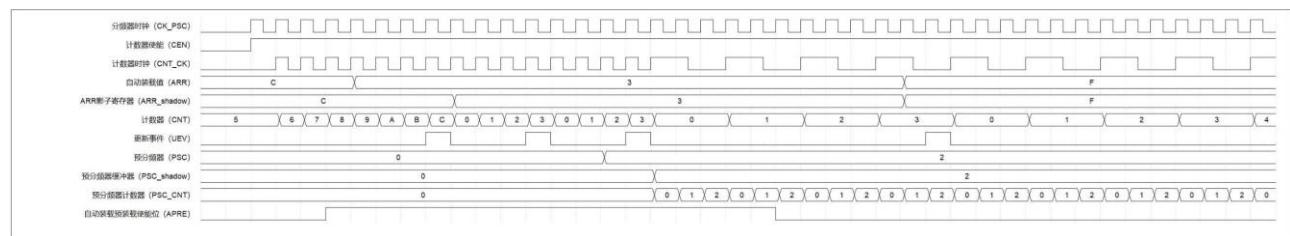


图 14-2 自动预装载

14.4.1.3 计数模式

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

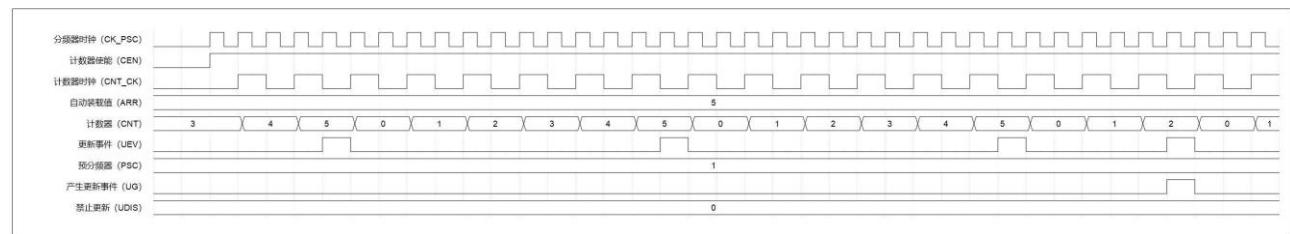


图 14-3 递增计数模式 (UDIS=0)

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。

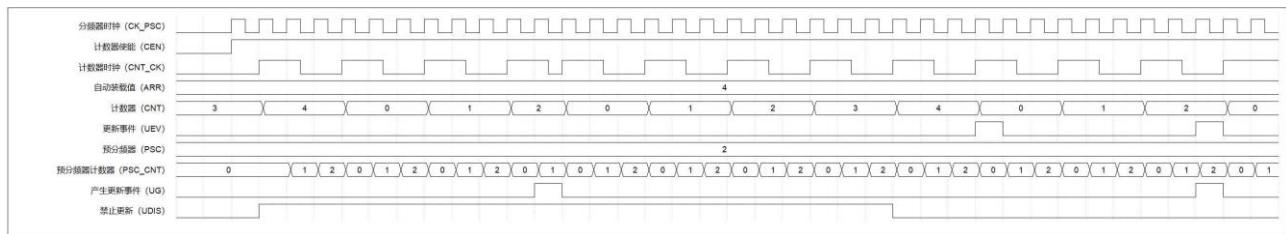


图 14-4 递增计数模式 (UDIS=1 禁止产生更新事件)

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。

14.4.2 调试模式

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

14.4.3 中断

TIMx 仅提供更新中断，当更新中断使能位打开，发生更新事件时，产生更新中断。

表 14-1 中断事件一览表

中断事件	标志位	使能位
更新中断	UIF	UIE

14.4.4 DMA

TIMx 能够在发生更新事件时生成 DMA 请求。

14.5 寄存器描述

表 14-2 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x0C	TIMx_DIER	DMA/中断使能寄存器 (DMA 部分仅适用于有内置 DMA 的芯片)	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000

Offset	Acronym	Register Name	Reset
0x2C	TIMx_ARR	自动装载寄存器	0x0000

14.5.1 TIMx_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.								APRE	Res.			OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	APRE	自动重装载预装载使能 (Auto-reload preload enable) 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 4	Reserved	保留, 必须保持复位值。
3	OPM	单脉冲模式 (One pulse mode) 0: 禁止单脉冲模式, 在发生更新事件时, 计数器继续计数 1: 使能单脉冲模式, 在发生下一次更新事件或软件清除 CEN 位时, 计数器停止计数
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢 - 设置 UG 位 1: 只有计数器上溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC) 保持值不变。如果设置了 EGR_UG 位为 1, 计数器和预分频器被初始化。
0	CEN	计数器使能 (Counter enable) 0: 禁止计数器 1: 使能计数器

14.5.2 TIMx_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.								UDE	Res.								UIE
								rw									rw

Bit	Field	Description
15: 9	Reserved	保留, 必须保持复位值。
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: 1	Reserved	保留, 必须保持复位值。
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

14.5.3 TIMx_SR 状态寄存器

偏移地址: 0x10

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														UIF	
														r_w0c	

Bit	Field	Description
15: 1	Reserved	保留, 必须保持复位值。
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 当计数器产生上溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。

14.5.4 TIMx_EGR 事件产生寄存器

偏移地址: 0x14

复位值: 0x0000

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

w

Bit	Field	Description
15: 1	Reserved	保留，必须保持复位值。
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。由硬件自动清零。

14.5.5 TIMx_CNT 计数器

偏移地址: 0x24

复位值: 0x0000

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

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

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

14.5.7 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。

15 SDIO 安全数字输入输出接口

15.1 简介

SDIO 控制器用于同外部 SDIO/MMC/SD 卡通信，并支持对外部设备的读/写访问。

15.2 主要特征

- 完全兼容 SD 记忆卡规格 1.0
- 完全兼容 SD 存储卡规格 1.1（高速）
- 完全兼容 SD 记忆卡规格 2.0（SDHC）
- 完全兼容 MMC 系统规格 2.0~4.2
- 完全兼容 SDIO 存储卡规格 1.1.0
- 标准的 MMC 模式接口支持
- 可编程时钟速率
- 自动命令/响应 CRC 生成/检查
- 自动数据 CRC 生成/检查
- 可编程超时检测
- 内置 512 Bytes (128 Words) 数据 FIFO，并可通过 CPU 或 DMA 访问
- 支持产生 DMA 请求
- 支持产生中断事件

15.3 功能描述

15.3.1 功能框图

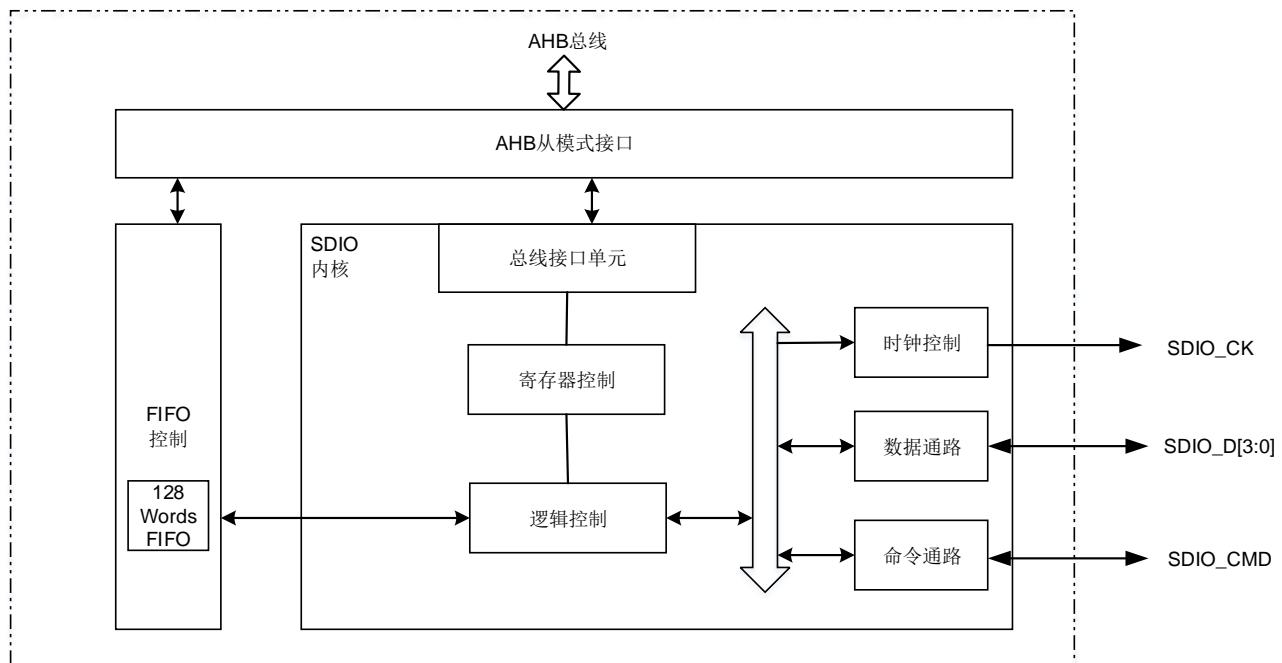


图 15-1 SDIO 功能框图

15.3.2 信号描述

表 15-1 SDIO 信号描述表

信号名	类型	描述
SDIO_CK	输出	输出时钟
SDIO_CMD	输入输出	输出命令/捕获响应
SDIO_D[3: 0]	输入输出	输入输出数据

15.3.3 中断事件

表 15-2 SDIO 中断事件

中断事件	事件标志/清除位	使能控制位
SDIO 数据 1 线路中断	D1MC	D1INTEN
CRC 状态令牌错误中断	CRCEMC	CRCINTEN
命令和响应 Ncr 超时中断	CRNTMC	CRTINTEN
多块超时中断	MBTMC	MBTINTEN
多块完成中断	MBDMC	MBDINTEN

中断事件	事件标志/清除位	使能控制位
CMD CRC 错误中断	CMDEMC	CMDEINTE N
DAT CRC 错误中断	DATEMC	DATEINTEN
DAT 完成中断	DATDMC	DATDINTE N
CMD 完成中断	CMDDMC	CMDDINTE N

注：

- 通过配置 INTEN 寄存器控制是否开放中断请求；
- 通过读取 INTSTA 寄存器判断中断事件是否发生，然后，可向 INTSTA 寄存器对应位写“1”清除该标志位。

15.3.4 DMA

SDIO 模块中，仅有 FIFO 事件可以产生 DMA 请求。

- 在发送数据模式下，当 FIFO 空时，会产生 DMA 请求，需要 DMA 向 FIFO 中搬入即将发送的数据。
- 在接收数据模式下，当 FIFO 满时，会产生 DMA 请求，需要 DMA 从 FIFO 中搬出已经接收的数据。

当需要使用 SDIO FIFO DMA，则需如下配置：

- 启用 DMA 控制器中的 SDIO 对应通道，具体请参考 DMA 章节；
- 通过设置寄存器 BUFCTL[DRM]=0，允许 SDIO 产生 DMA 请求；
- 通过设置寄存器 BUFCTL[DMAHEN]=1，切换为 DMA 访问模式。

15.3.5 FIFO

- 具有一个 512 Bytes (128 Words) 的 FIFO；
- 可通过 BUFCTL 寄存器的 DBE 位来判断 FIFO 是否为空，DBF 位来判断 FIFO 是否为满，DBE 和 DBF 是只读状态标志位；
- 可通过 BUFCTL 寄存器的 DMAHEN 位来选择传输方式为 CPU 或 DMA 访问。
- 可通过 BUFCTL 寄存器的 SBAD 位控制 FIFO 的方向。需要人工根据数据发送或接收模式对应设定 FIFO 的方向。

15.3.6 传输信号时序

SDIO 总线通信可分为命令通信和数据通信：

- **Command:** 命令是一次 SDIO 操作开始的令牌，从主机发送命令到总线上，命令在 SDIO_CMD 线上连续传输。
- **Response:** 响应是从外部设备返回到主机的令牌，作为对先前接收到命令的应答。响应在 SDIO_CMD 线上连续传输。
- **Data:** 数据可以通过 SDIO_D[3: 0]双向传输。
- 时钟：命令通信和数据通信都使用 SDIO_CK 线上的时钟信号同步。

SDIO 以块（block）作为发送或接收数据的基本单元。数据块以 CRC 字段来验证传输成功。SDIO 支持单块和多块数据通信。

在写操作时，多块操作模式效率更高。多块传输在 CMD 线上产生 stop 命令时结束。

在总线上的通信是基于命令和数据传输的。MMC/SD/SDIO 总线上的基本事务是命令/响应事务。这些类型的总线事务直接在命令或响应结构中传输其信息。

从 SDIO 的数据传输在数据块中完成。从 MMC 的数据传输是数据块或数据流。

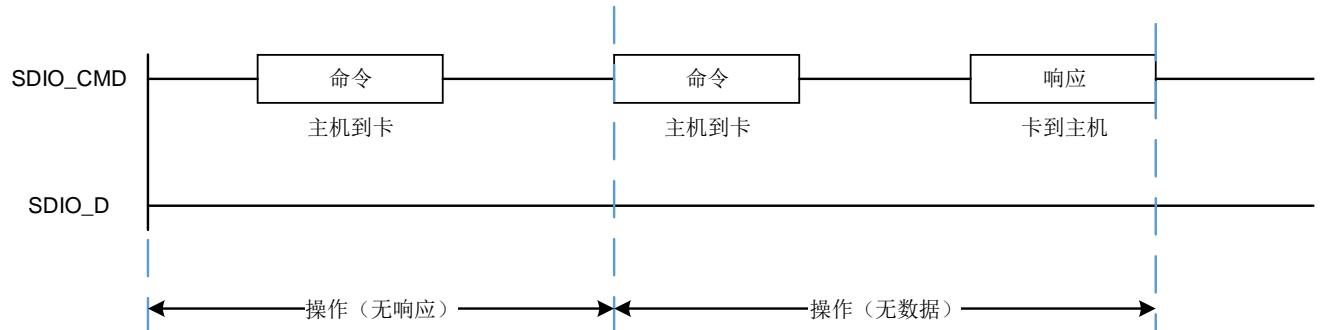


图 15-2 SDIO“无响应”和“无数据”操作

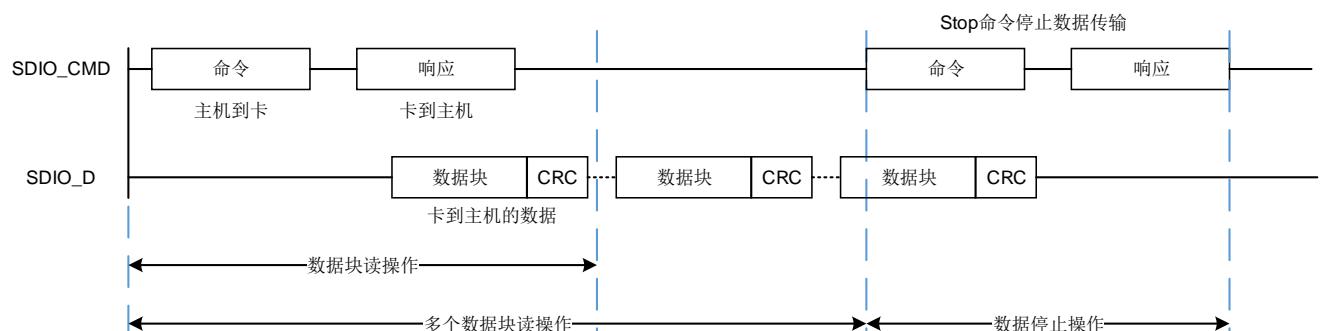


图 15-3 SDIO（多）数据块读操作

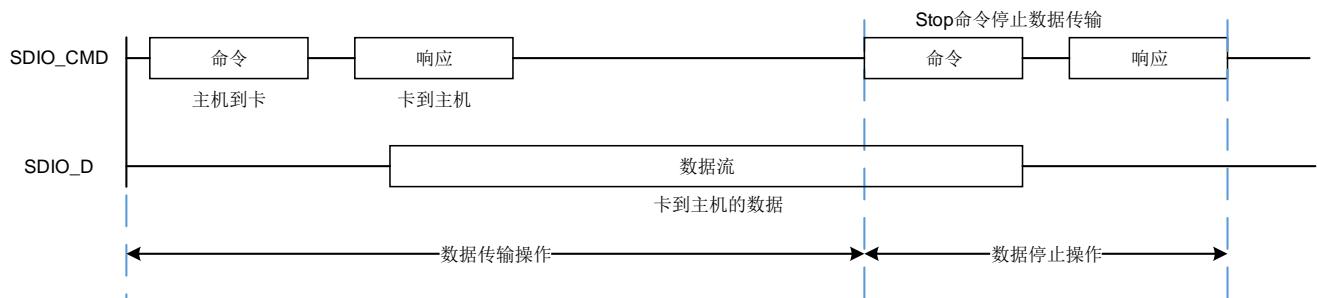


图 15-4 SDIO 连续读操作

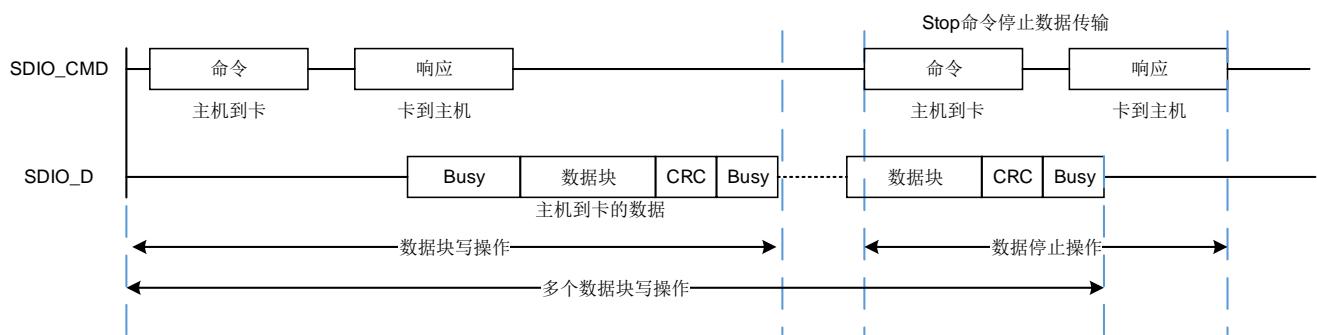


图 15-5 SDIO (多) 数据块写操作

注：当有 Busy 信号时，SDIO D0 被拉低，SDIO 将不会发送任何数据。

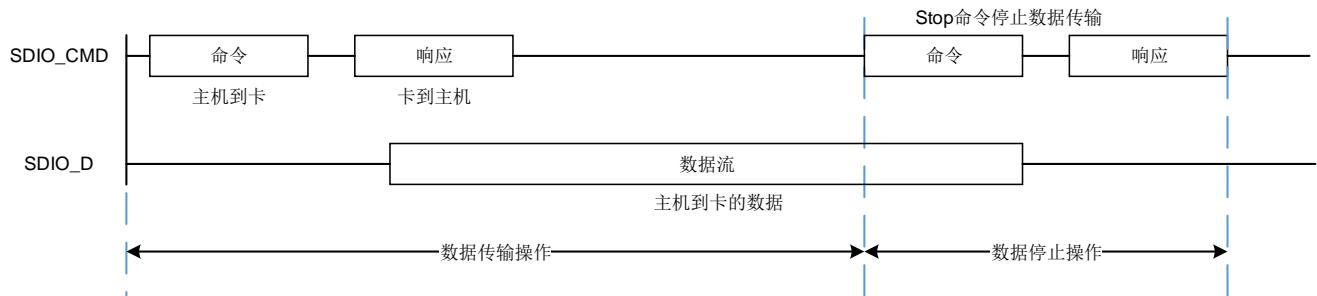


图 15-6 SDIO 连续写操作

15.3.7 传输控制命令

SDIO 通过表 IO 寄存器[7: 0]的详细描述中描述的配置组合，实现通过 SDIO 模块启动传输（命令传输和数据传输）。

当配置 IO[TRANSFDIR]为 0 时，需同时配置 BUFCTL[SBAD]为 1；

当配置 IO[TRANSFDIR]为 1 时，需同时配置 BUFCTL[SBAD]为 0。

表 15-3 IO 寄存器[7: 0]的详细描述

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	操作描述	Transfer
CM DAF	CM DC H	AUT OC LKG	EN RR ESP	PCL KG	CID/ CS DR D	TRA NSF DIR	AUT OD ATT R		
X	X	X	X	X	0	X	0	没有操作	N/A
X	X	1	X	0	Trig	X	0	生成 8 个空时钟	N/A
0	0	0	X	0	Trig	X	0	发送命令	6
0	0	0	0	1	Trig	X	0	接收响应	6
0	0	0	1	1	Trig	X	0	重申响应	17
1	0	0	0	0	Trig	X	0	发送命令+8 个空时钟	N/A
0	1	0	0	0	Trig	X	0	发送命令+接收响应	N/A
1	1	0	0	0	Trig	X	0	发送命令, 接收响应+8 个空时钟	N/A
X	X	X	X	X	0	1	Trig	从卡中读取单个数据+8 个空时钟	BYTECNT
X	X	X	X	X	0	0	Trig	将单个数据到卡写入卡+8 个空时钟	BYTECNT

IO 的 bit[7: 0]定义为 SDIO/MMC 命令, 响应和单块数据操作控制。

注意事项:

- 在上表中, 如果中断未被屏蔽, 除了最后两行, 其它都会产生 CMD 完成中断。
- 在上表中, 如果中断未被屏蔽, 只有最后两行会产生 DAT 完成中断。

表 15-4 IOMBCTL[2: 0]和 IO[7: 6]的详细描述表

7	6	2	1	0	操作说明	传输字节
IO		IOMBCTL				
1	1	Trig	0	0	写入多个块命令+响应+8 个时钟+数据响应+8 个空时钟+数据	BLOCKCNT
1	1	Trig	1	0	读取多个块命令+响应+8 个时钟+数据响应+8 个空时钟+数据	BLOCKCNT
X	X	0	0	Trig	只写入多个块数据	BLOCKCNT
X	X	0	1	Trig	只读取多个块数据	BLOCKCNT

IOMBCTL 的 Bit[2: 0]和 IO 的 Bit[7: 6]是为 SD/MMC/SDIO 多块数据操作控制定义的。

注意事项:

- 在上表中, 如果中断未被屏蔽, 前两个都将产生多块传输完成中断, 每个 block 都会产生 DAT 完成中断和 CMD 完成中断。
- 在上表中, 如果中断未被屏蔽, 当发生超时事件时, 不会产生多块传输完成中断, 但会产生多块传输超时中断。

其他注意事项：

- 如果没有使用中断（中断被屏蔽），当命令/响应/8个空时钟被传输时，可以检查 IO[2]；当数据被传输时，可以检查 IO[0]；当应该传输多个块数据时，可以检查 IOMBCTL[2] / IOMBCTL[0]。
- 如果产生 Ncr 超时或多个块数据超时，传输将被暂停。软件应该重新开始新的控制。

15.4 寄存器

15.4.1 寄存器总览

表 15-5 SDIO 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CTRL	控制寄存器	0x00000045
0x04	IO	IO 信号寄存器	0x00000000
0x08	BYTECNT	数据传输字节计数寄存器	0x00000200
0x0C	TRBLOCKCNT	块计数寄存器	0x00000000
0x10	CRCCTL	CRC 控制寄存器	0x00000000
0x14	CMDCRC	命令 CRC 寄存器	0x00000000
0x18	DATCRC	数据 CRC 寄存器	0x00000000
0x20	PORT	端口控制寄存器	0x0000007F
0x24	INTEN	中断使能寄存器	0x00000000
0x28	INTSTA	中断状态寄存器	0x00000000
0x2C	CARDSEL	卡选择寄存器	0x00000040
0x30	SIG	信号状态寄存器	0x000000FF
0x34	IOMBCTL	多块控制寄存器	0x00000010
0x38	BLOCKCNT	数据块传输计数寄存器	0x00000001
0x3C	TIMEOUTCNT	数据传输超时寄存器	0x00000040
0x40-0x7C	CMDBUFx (0...15)	命令缓冲寄存器	0x00000000
0x80	BUFCTL	数据缓冲控制寄存器	0x00000002
0x100-0x2FF	DATABUF	数据缓冲寄存器	0x00000000

15.4.2 CTRL 控制寄存器

偏移地址：0x00

复位值：0x0000 0045

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					RDWTE N	INTEN	MDEN	DATWT	SELPTS M	CLKSP			OUTM	SELSM	OPMSE L
					rw	rw	rw	rw	rw	rw			rw	rw	rw

Bit	Field	Description
31: 11	Reserved	保留, 必须保持复位值
10	RDWTEN	SDIO 读等待使能信号 (SDIO Read Wait Enable Signal) 0: SDIO 读等待禁止 1: SDIO 读等待使能
9	INTEN	SDIO 中断使能信号 (SDIO Interrupt Enable Signal) 0: SDIO 中断禁止 1: SDIO 中断使能
8	MDEN	SDIO 模式使能 (SDIO Mode Enable) 0: SD/MMC 1: SDIO
7	DATWT	定义 SD/MMC/SDIO 端口 SDIO_D 线的总线宽度 (Define The Bus Width Of SD/MMC/SDIO Port SDIO_D Line) 0: 1 位 1: 4 位
6	SELPTSM	SD/MMC/SDIO 时钟源选择 (Select SD/MMC/SDIO Port Transfer Speed Mode) 0: SD/MMC/SDIO 端口低速传输模式 1: SD/MMC/SDIO 端口高速传输模式 注: 当高速传输模式时, 基准时钟是 HCLK 当低速传输模式时, 基准时钟是 CLK1M 1MHz=Fclk/ ((TSCALE+1) *2)
5: 3	CLKSP	SD/MMC/SDIO 时钟源分频选择位 (SD/MMC/SDIO Port CLK Line Speed Selection) 000: 1/2 基准时钟 001: 1/4 基准时钟 010: 1/6 基准时钟 011: 1/8 基准时钟 100: 1/10 基准时钟 101: 1/12 基准时钟 110: 1/14 基准时钟 111: 1/16 基准时钟 注: 是为 SDIO/MMC 端口速度选择而定义的
2	OUTM	SD/MMC/SDIO 端口 CMD 输出驱动模式选择位 (SD/MMC/SDIO Port CMD Line Output Driver Mode Selection) 0: 推挽输出 1: 开漏输出

Bit	Field	Description
1	SELSM	选择信号模式位 (Select Signal Mode) 0: SD/MMC/SDIO 端口使用 PORT 寄存器指定信号 1: SD/MMC/SDIO 端口 SDIO 模块自动管理传输
0	OPMSEL	SD/MMC/SDIO 端口操作模式选择位 (SD/MMC/SDIO Port Operation Mode Select) 0: SPI 模式 1: SD/MMC/SDIO 模式

15.4.3 IO IO 信号寄存器

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.						CMDAFA	CMDCH	AUTOC LKG	ENRRE SP	PCLKG	CID/CS DRD	RESPC MDSEL	AUTOT R	TRANS FDIR	AUTOD ATTR
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Field	Description
31: 10	Reserved	保留, 必须保持复位值
9	CMDAFA	SDIO CMD12/IO 中止标志 (SDIO CMD12/IO Abort Flag) 0: 标记当前命令不是 CMD12/IO 中止命令 1: 标记当前命令是 CMD12/IO 中止命令
8	CMDCH	SDIO 命令特征位 (SDIO Command Character) 0: 标志当前命令后面既不是数据块也不是响应 1: 标志当前的命令后跟数据块
7	AUTOCLKG	在响应/命令或单个数据块后使能自动转换 8 空时钟 (Enable Auto Generate 8 Null Clock After Response/Command Or Single Block Data) 0: 禁止 1: 使能
6	ENRRESP	在命令后使能自动接收响应 (Enable Auto Receive Response After Command) 0: 禁止 1: 使能

Bit	Field	Description
5	PCLKG	SD/MMC/SDIO 端口 CLK 线 8 个空时钟产生 (SD/MMC/SDIO Port CLK Line 8 Null Clocks Generation) 0: 通过 RESPCMDSEL 位选择接收响应/发送命令 1: 8 个空时钟产生
4	CID/CSDRD	这是为 CID 和 CSD 读取而设计的。发出读取 CID 或 CSD 命令时，SD/MMC/SDIO 卡将响应 CMD 线中的 136 位 CID 或 CSD 数据。将该位置“1”，将得到命令缓冲区[135: 8]中的 CID 或 CSD 数据
3	RESPCMDSEL	PCLKG 位为“0”时的响应/命令选择 (Response/Command Selection When Bit[5] Is“0”) 0: 发送命令 1: 接收回应
2	AUTOTR	设置自动 8 个空时钟/命令/响应传输 (Set Auto 8 Null Clocks/Command/Response Transfer) 0: 禁止自动 8 个空时钟/命令/响应传输 1: 使能自动 8 个空时钟/命令/响应传输 根据 PCLKG 位和 RESPCMDSEL 位，选择生成 8 个空时钟/接收响应/发送命令，当传输完成后，该位将自动清除
1	TRANSFDIR	设置数据传输方向 (Set Data Transfer Direction) 0: 写入数据 1: 读取数据
0	AUTODATTR	设置自动数据传输 (Set Auto Data Transfer) 0: 禁止自动数据传输 1: 使能自动数据传输 数据传输完成后，该位将自动清除。 有关 SD/MMC/SDIO 操作控制的更多信息，请参考 MMC_IO 寄存器[7: 0]的详细描述表

15.4.4 BYTECNT 数据传输字节计数寄存器

偏移地址: 0x08

复位值: 0x0000 0200

31	30	29	28	27	26	25	24	23	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	数据传输字节计数寄存器 (Data Transfer Byte Count Register)

15.4.5 TRBLOCKCNT 块计数寄存器

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

Res.

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

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 0	CNT	当多块传送时, 传输完成的计数器值

15.4.6 CRCCTL CRC 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.								CMD_C RCEN	DAT_C RCEN	ENCHK B	ENRDM B	DAT_CRCS		CMD_C RCE	DAT_C RCE
								rw	rw	rw	rw	rw		r	r

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	CMD_CRCEN	SD/MMC/SDIO 端口 CMD 线路 CRC 电路使能 (SD/MMC/SDIO Port CMD Line CRC Circuit Enable) 0: 禁止 1: 使能
6	DAT_CRCEN	SD/MMC/SDIO 端口 DAT 线路 CRC 电路使能 (SD/MMC/SDIO Port DAT Line CRC Circuit Enable) 0: 禁止 1: 使能

Bit	Field	Description
5	ENCHK	使能自动检查 CRC_STATUS (Enable Auto Check CRC_STATUS) 0: 禁止 1: 使能 如果 CRC_STATUS != 010, CRC 状态错误中断产生, 写数据传输将通过停止命令来终止并清除 IO[0]或 IOMBCTL[2: 0]
4	ENRDMB	在响应之前使能读取多个块数据 (Enable Read Multiple Block Data Before Response) 0: 禁止 1: 使能
3: 2	DAT_CRCS	DAT CRC 选择 (DAT CRC Selection) 00: SD/MMC/SDIO SDIO_D0 线路 CRC 结果 01: SD/MMC/SDIO SDIO_D1 线路 CRC 结果 10: SD/MMC/SDIO SDIO_D2 线路 CRC 结果 11: SD/MMC/SDIO SDIO_D3 线路 CRC 结果 注: 是为 DAT CRC 结果显示设置定义的
1	CMD_CRCE	CMD CRC 错误 (CMD CRC Error) 0: 无 CRC 错误 1: CRC 错误
0	DAT_CRCE	DAT CRC 错误 (DAT CRC Error) 0: 无 CRC 错误 1: CRC 错误

15.4.7 CMDCRC 命令 CRC 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.												CMD_CRCV			
r															

Bit	Field	Description
31: 7	Reserved	保留, 必须保持复位值
6: 0	CMD_CRCV	CMD CRC 寄存器值 (CMD CRC Register Value)

15.4.8 DATCRC 数据 CRC 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
DAT_CRC															
r															

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 0	DAT_CRC	DAT CRC 寄存器值 (DAT CRC Register Value)

15.4.9 PORT 端口控制寄存器

偏移地址: 0x20

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.								PCLKS	PCMDS	PDATS	AUTON TEN	NTCR			
								rw	rw	rw	rw	rw			

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	PCLKS	SD/MMC/SDIO 端口 CLK 线信号 (SD/MMC/SDIO Port CLK Line Signal) 0: 低电平 1: 高电平 当 CTRL[0]为 0 时选择 PORT[7:5]由软件控制。
6	PCMDS	SD/MMC/SDIO 端口 CMD 线信号 (SD/MMC/SDIO Port CMD Line Signal) 0: 低电平 1: 高电平
5	PDATS	SD/MMC/SDIO 端口 DAT 线路信号 (SD/MMC/SDIO Port DAT Line Signal) 0: 低电平 1: 高电平

Bit	Field	Description
4	AUTONTEN	自动 Ncr 定时器输出使能 (Auto Ncr Timeout Enable) 0: 禁止自动检查 Ncr 超时 1: 使能自动检查 Ncr 超时
3: 0	NTCR	Ncr 超时计数寄存器 (Ncr Timeout Count Register) (SD/MMC/SDIO 时钟个数)

15.4.10 INTEN 中断使能寄存器

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.							D1INTE	CRCINT	CRTINT	MBTINT	MBDINT	CMDEI	DATEIN	DATDIN	CMDDI
N	EN	EN	EN	EN	EN	NTEN	TEN	TEN	TEN	NTEN	NTEN	NTEN	NTEN	NTEN	NTEN

Bit	Field	Description
31: 9	Reserved	保留, 必须保持复位值
8	D1INTEN	SDIO 数据 1 线路中断使能 (SDIO DAT1 Line Interrupt Enable) 0: 中断禁止 1: 中断使能
7	CRCINTEN	CRC 状态令牌错误中断使能 (CRC Status Token Error Interrupt Enable) 0: 中断禁止 1: 中断使能
6	CRTINTEN	命令和响应 Ncr 超时中断使能 (Cmd And Resp Ncr Timeout Interrupt Enable) 0: 中断禁止 1: 中断使能
5	MBTINTEN	多块超时中断使能 (Multi Block Timeout Interrupt Enable) 0: 中断禁止 1: 中断使能
4	MBDINTEN	多块完成中断使能 (Multi Block Done Interrupt Enable) 0: 中断禁止 1: 中断使能
3	CMDEINTEN	CMD CRC 错误中断使能 (CMD CRC Error Interrupt Enable) 0: 中断禁止 1: 中断使能
2	DATEINTEN	DAT CRC 错误中断使能 (DAT CRC Error Interrupt Enable) 0: 中断禁止 1: 中断使能

Bit	Field	Description
1	DATDINTEN	DAT 完成中断使能 (DAT Done Interrupt Enable) 0: 中断禁止 1: 中断使能
0	CMDDINTEN	CMD 完成中断使能 (CMD Done Interrupt Enable) 0: 中断禁止 1: 中断使能 注: CRC 状态令牌错误中断, Ncr 超时中断, CMD CRC 错误中断和 DAT CRC 错误中断不影响其他中断的产生

15.4.11 INTSTA 中断状态寄存器

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.							D1MC	CRCEM	CRNTM	MBTMC	MBDMC	CMDEM	DATEM	DATDM	CMDDM
							w1c	w1c	w1c	w1c	w1c	c	c	c	w1c

Bit	Field	Description
31: 9	Reserved	保留, 必须保持复位值
8	D1MC	SDIO 数据 1 线路中断标志/清除 写入时, 清除 SDIO 数据 1 线路中断。在清除中断标志后, 将自动复位为“0” 读取时, SDIO 数据 1 线路中断标志
7	CRCEMC	CRC 状态令牌错误中断标志/清除 写入时, 清除 CRC 状态令牌错误中断标志。在清除中断标志后, 将自动复位为“0” 读取时, CRC 状态令牌错误中断标志 当该位为 1 时, 判断 CRC_STATUS 位
6	CRNTMC	命令和回应 Ncr 超时中断标志/清除 写入时, 清除命令和回应 Ncr 超时中断标志。在清除中断标志后, 将自动复位为“0” 读取时, 命令和回应 Ncr 超时中断标志
5	MBTMC	多块传输超时中断标志/清除 写入时, 清除多块传输超时中断标志。在清除中断标志后, 将自动复位为“0” 读取时, 多块读取和写入超时中断标志

Bit	Field	Description
4	MBDMC	多块传输完成中断标志/清除 写入时, 清除多块传输完成中断标志。在清除中断标志后, 将自动复位为“0” 读取时, 多块传输完成中断标志
3	CMDEMC	CMD CRC 错误中断标志/清除 写入时, 清除 CMD CRC 错误中断标志。在清除中断标志后, 将自动复位为“0” 读取时, CMD CRC 错误中断标志
2	DATEMC	DAT CRC 错误中断标志/清除 写入时, 清除 DAT CRC 错误中断标志。在清除中断标志后, 将自动复位为“0” 读取时, DAT CRC 错误中断标志
1	DATDMC	DAT 完成中断标志/清除 写入时, 清除 DAT 完成中断标志。在清除中断标志后, 将自动复位为“0” 读取时, DAT 完成中断标志
0	CMDDMC	CMD 完成中断标志/清除 写入时, 清除 CMD 完成中断标志。在清除中断标志后, 将自动复位为“0” 读取时, CMD 完成中断标志

15.4.12 CARDSEL 卡选择寄存器

偏移地址: 0x2C

复位值: 0x0000 0040

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								CTREN	ENPCL	TSCALE					
Res.									K						
								rw	rw	rw					

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	CTREN	SD/MMC/SDIO 控制器使能 (SD/MMC/SDIO Controller Enable) 0: 禁止 1: 使能
6	ENPCLK	使能卡的 SD/MMC/SDIO 端口 CLK 时钟 (Enable SD/MMC/SDIO Port CLK Line For Card) 0: 禁止 1: 使能
5: 0	TSCALE	SD/MMC/SDIO 时钟分频系数 (SD/MMC/SDIO Time Scale Base Coefficient) 使用这些位来构建 1Mhz 时钟 $1\text{MHz}=\text{Fhclk}/((\text{TSCALE}+1) * 2)$

15.4.13 SIG 信号状态寄存器

偏移地址: 0x30

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.								PCMDS	CRC_STATUS			PDAT3S	PDAT2S	PDAT1S	PDAT0S
								r	r			r	r	r	r

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7	PCMDS	SD/MMC/SDIO 端口 CMD 线路信号 (SD/MMC/SDIO Port CMD Line Signal)
6: 4	CRC_STATUS	写入数据 CRC 状态令牌时的 CRC 状态 (CRC_STATUS When Write Data CRC Status Token)
3	PDAT3S	SD/MMC/SDIO 端口 SDIO_D3 线路信号 (SD/MMC/SDIO Port SDIO_D3 Line Signal)
2	PDAT2S	SD/MMC/SDIO 端口 SDIO_D2 线路信号 (SD/MMC/SDIO Port SDIO_D2 Line Signal)
1	PDAT1S	SD/MMC/SDIO 端口 SDIO_D1 线路信号 (SD/MMC/SDIO Port SDIO_D1 Line Signal)
0	PDAT0S	SD/MMC/SDIO 端口 SDIO_D0 线路信号 (SD/MMC/SDIO Port SDIO_D0 Line Signal) 当主机读取寄存器时, SD/MMC/SDIO 控制器将在 SD/MMC/SDIO 端口 CLK 线上产生一个时钟脉冲; 当 SD/MMC/SDIO 端口 CLK 线上升时, SD/MMC/SDIO 端口的信号状态将被锁存到寄存器中。

15.4.14 IOMBCTL 多块控制寄存器

偏移地址: 0x34

复位值: 0x0000 0010

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								NTSSEL		BTSSEL		PCLKP	PAUTO	SMBDT	SPMBD

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 6	NTSSEL	<p>SD/MMC/SDIO NAC 超时级别选择 (SD/MMC/SDIO NAC Timeout Scale Selection)</p> <p>00: 1us 01: 100us 10: 10ms 11: 1s</p> <p>注：是用于 SD/MMC/SDIO 端口 NAC 超时刻度选择</p>
5: 4	BTSSEL	<p>SD/MMC/SDIO BUSY 超时级别选择 (SD/MMC/SDIO Busy Timeout Scale Selection)</p> <p>00: 1us 01: 100us 10: 10ms 11: 1s</p> <p>注：是用于 SD/MMC/SDIO 端口 BUSY 超时刻度选择</p>
3	PCLKP	<p>SD/MMC/SDIO 端口时钟极性选择 (SD/MMC/SDIO Port CLK Line Polarity)</p> <p>0: 时钟在空闲状态时为低电平 1: 时钟在空闲状态时为高电平</p>
2	PAUTOTR	<p>设置 SD/MMC/SDIO 端口全自动命令和多块数据传输 (Set SD/MMC/SDIO Port Full Auto CMD And Multiple Block Data Transfer)</p> <p>0: 禁止 1: 使能</p> <p>设置此位为“1”(MMC_IO[7: 6]==11) 将触发 SD/MMC/SDIO 命令，响应 8 个空时钟，多块数据传输。数据传输完成后，该位将自动清“0”。</p>
1	SMBDTD	<p>多块数据传送方向选择位 (Select Multiple Block Data Transfer Direction)</p> <p>0: 写入数据 1: 读取数据</p>
0	SPMBDTR	<p>设置 SD/MMC/SDIO 端口自动多块数据传输位 (Set SD/MMC/SDIO Port Auto Multiple Block Data Transfer)</p> <p>0: 禁止 1: 使能</p> <p>将该位置“1”将触发 SD/MMC/SDIO 多块数据传输。块计数由 BLOCKCNT 寄存器定义。数据传输完成后，该位将自动清“0”。</p> <p>IOMBCTL[2: 0]和 IO[7: 6]为 SD/MMC/SDIO 多块数据操作控制，请参考 IOMBCTL[2: 0]和 IO[7: 6]的详细描述表</p>

15.4.15 BLOCKCNT 数据块传输计数寄存器

偏移地址: 0x38

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
BCNT															
rw															

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 0	BCNT	数据块计数寄存器 (Data Block Number Count Register) 在多块传输模式下, 配置这些位定义将要传输的数据块数量

15.4.16 TIMEOUTCNT 数据传输超时寄存器

偏移地址: 0x3C

复位值: 0x0000 0040

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

Res.

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7: 0	DTCNT	数据传输超时计数寄存器 (Data Transfer Timeout Count Register) Time=Scale*DTCNT 注: Scale 根据 IOMBCTL 寄存器的 NTSSEL 位或者 BTSSEL 位进行定义

15.4.17 CMDBUF_x (x=0...15) 命令缓冲寄存器

偏移地址: 0x40-0x7C

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

Reserved	DAT
	rw

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7: 0	DAT	CMDBUF 字节 x, 映射到命令[(15+8x): 8 (x+1)]位 CMDBUF 字节 0, 映射到命令[15: 8]位 CMDBUF 字节 1, 映射到命令[23: 16]位 CMDBUF 字节 2, 映射到命令[31: 24]位 CMDBUF 字节 3, 映射到命令[39: 32]位 CMDBUF 字节 4, 映射到命令[47: 40]位 CMDBUF 字节 5, 映射到命令[55: 48]位 CMDBUF 字节 6, 映射到命令[63: 56]位 CMDBUF 字节 7, 映射到命令[71: 64]位 CMDBUF 字节 8, 映射到命令[79: 72]位 CMDBUF 字节 9, 映射到命令[87: 80]位 CMDBUF 字节 10, 映射到命令[95: 88]位 CMDBUF 字节 11, 映射到命令[103: 96]位 CMDBUF 字节 12, 映射到命令[111: 104]位 CMDBUF 字节 13, 映射到命令[119: 112]位 CMDBUF 字节 14, 映射到命令[127: 120]位 CMDBUF 字节 15, 映射到命令[135: 128]位

15.4.18 BUFCCTL 数据缓冲控制寄存器

偏移地址: 0x80

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
DBFEN	DRM	Res.	DFIFOS M	SBAD	DMAHE N	DBML						DBE	DBF		
rw	rw		rw	rw	rw	rw						r	r		

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15	DBFEN	数据 Buff 清空使能位 (Data Buff Flush Enable) 0: 没有效果 1: 触发清空数据 Buff 当写入“1”时, 1个时钟后自动清“0”

Bit	Field	Description
14	DRM	DMA 请求屏蔽 (DMA Request Mask) 0: 没有屏蔽 1: 屏蔽 注意: 请在重新配置 DMA 之前, 屏蔽此位, 启用 DMA 后, 取消屏蔽此位, DMA 请求才启动
13	Reserved	保留, 必须保持复位值
12	DFIFOSM	数据 FIFO 状态信号屏蔽位 (Data FIFO Status Signal Mask Bit) 0: 屏蔽数据 FIFO 状态 1: 激活显示数据 FIFO 状态 注: 数据 FIFO 状态信号, 高电平有效。FIFO 满读 SD 卡访问, FIFO 空写 SD 卡访问
11	SBAD	设置 Buff 的访问方向 (Set Buff Access Direction) 0: 读 1: 写
10	DMAHEN	DMA 硬件接口使能 (DMA Hardware Interface Enable) 0: 正常的 CPU 访问 Buffer 数据 1: DMA 硬件接口握手 当使用 DMA 接口时, 当块 (单块传输) 或多块 (多块传输) 传输完成时, 该位将自动复位
9: 2	DBML	数据 Buff 标记 (Data Buff Data Water Mark Level) 只有当 BUF_CTL 寄存器的 DMAHEN 位为 1 时该位才有效
1	DBE	数据 Buff 为空 (Data Buff Empty) 0: 数据 Buff 未空 1: 数据 Buff 为空
0	DBF	数据缓存已满 (Data Buff Full) 0: 数据缓存未满 1: 数据缓存已满

15.4.19 DATABUF 数据缓冲寄存器

偏移地址: 0x100-0x2FF

复位值: 0x0000 0000

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

Bit	Field	Description
31: 0	DB	数据 Buff (Data Buffer)

16 IWDG 独立看门狗

16.1 简介

独立看门狗的设计初衷是为了检测和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。

16.2 功能框图

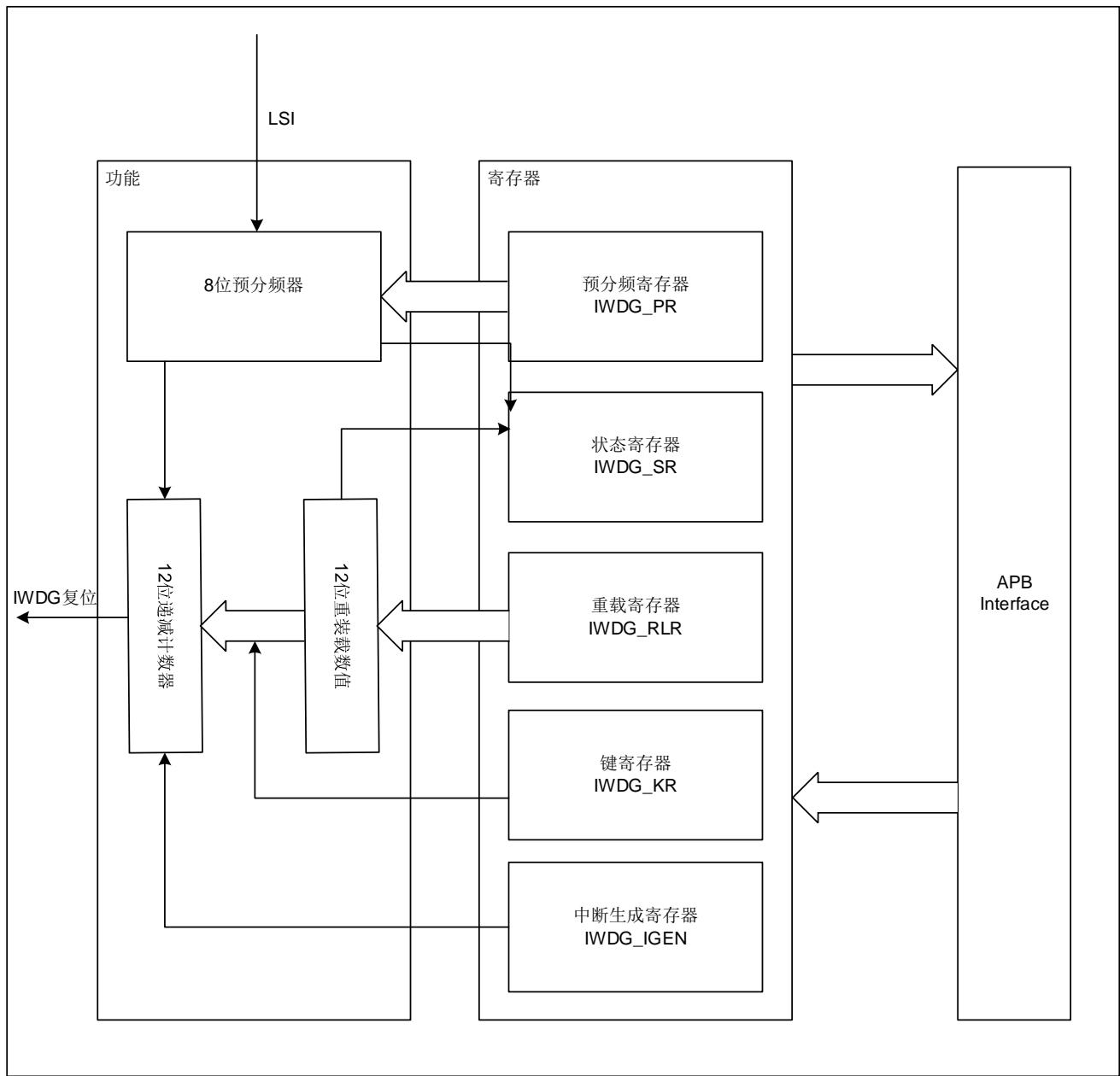


图 16-1 功能框图

16.3 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。

16.4 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。

16.4.1 流程框图

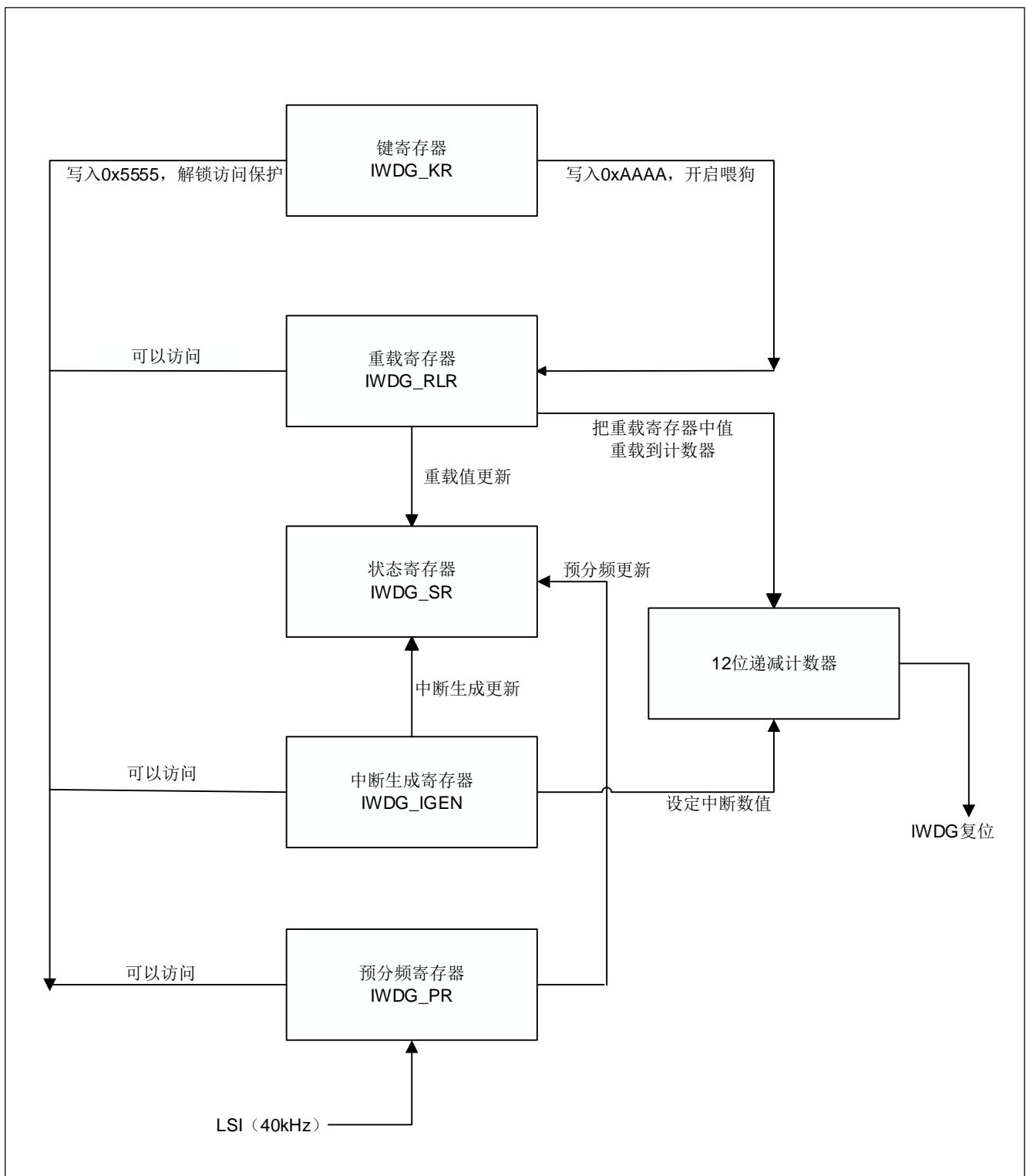


图 16-2 流程框图

16.4.2 独立看门狗超时时间

表 16-1 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:

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

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) \times 看门狗时钟周期。

16.4.3 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。

16.5 寄存器描述

16.5.1 寄存器总览

表 16-2 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	0x00000000
0x1C	IWDG_PS	分频计数寄存器	0x00000001

16.5.2 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	<p>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。</p>

16.5.3 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	<p>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。</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">000: 预分频因子 = 4</td> <td style="width: 50%;">100: 预分频因子 = 64</td> </tr> <tr> <td>001: 预分频因子 = 8</td> <td>101: 预分频因子 = 128</td> </tr> <tr> <td>010: 预分频因子 = 16</td> <td>110: 预分频因子 = 256</td> </tr> <tr> <td>011: 预分频因子 = 32</td> <td>111: 预分频因子 = 256</td> </tr> </table>	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
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									

16.5.4 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。

16.5.5 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 E	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。

Bit	Field	Description
2	IVU	看门狗中断生成值更新 (Watchdog Interrupt Generate value update) 此位由硬件置'1'用来指示中断生成值的更新正在进行中。 当在 VDD 域中的中断生成值更新结束后，此位由硬件清'0'（最多需要 5 个 40KHz 的振荡器周期）中断生成值只有在 IVU 位被清'0'后才可更新。
1	RVU	看门狗计数器重装载值更新 (Watchdog counter reload value update) 重装载值的更新正在进行中，此位置 1。 当重装载更新结束后，此位清'0'（最多需要 5 个 40KHz 的振荡器周期） 重装载值只有在 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装载值、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，表示已经配置好重装载值，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。

16.5.6 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

16.5.7 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											
Res.				rw											

Bit	Field	Description
31:12	Reserved	保留, 必须保持复位值
11: 0	IGEN	<p>IWDG 中断生成值 (Watchdog Interrupt Generate value) 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。</p>

16.5.8 IWDG_CNT 计数寄存器

偏移地址: 0x18

复位值: 0x0000 0000

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

Res.

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

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	IWDG_CNT	IWDG 计数器 counter 的值

16.5.9 IWDG_PS 分频计数寄存器

偏移地址: 0x1C

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.								IWDG_PS							
r															

Bit	Field	Description
31:8	Reserved	保留，必须保持复位值
7: 0	IWDG_PS	IWDG 时钟分频计数器的值

17 WWDG 窗口看门狗

17.1 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。

17.2 功能框图

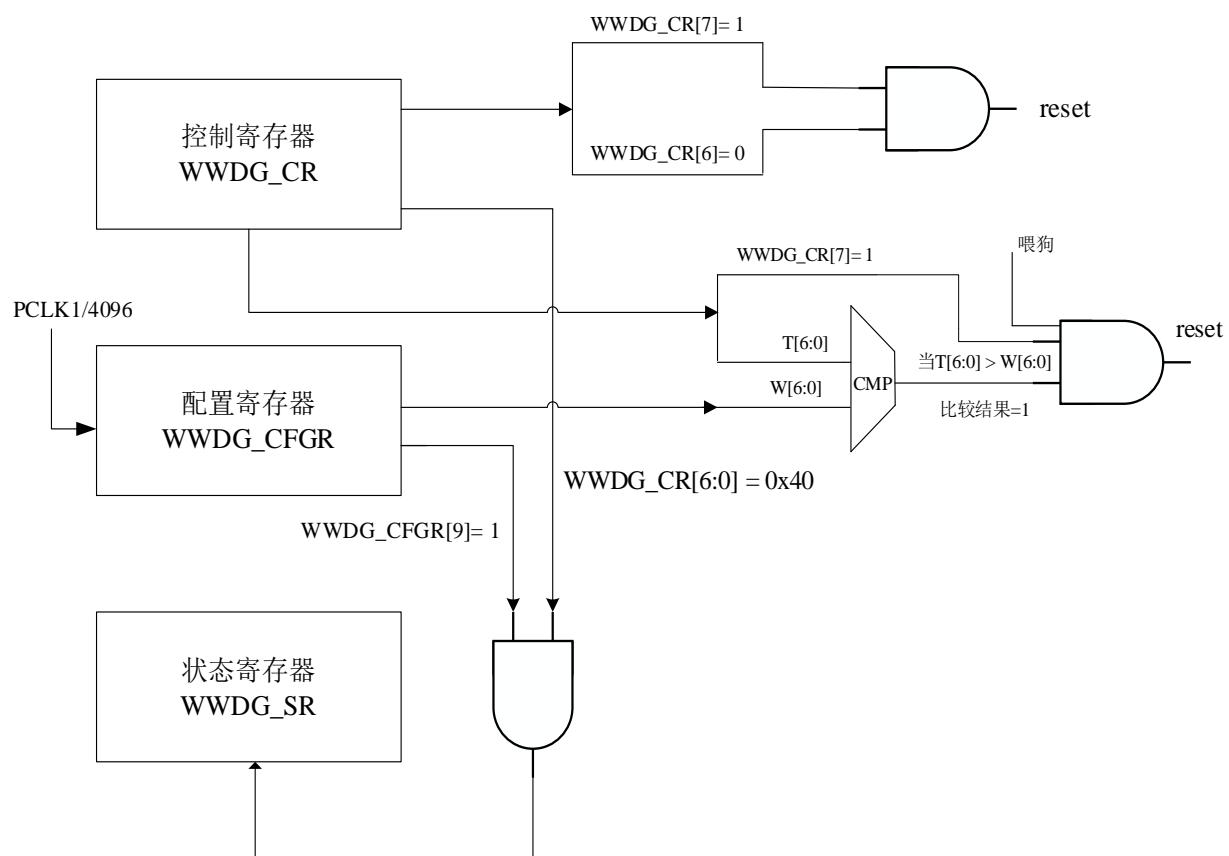


图 17-1 WWDG 功能框图

17.3 主要特征

- 可编程的自由运行的 7 位递减计数器
- 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。

17.4 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 EWI 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。

17.5 窗口看门狗超时时间

T[6:0]CNT递减计数器

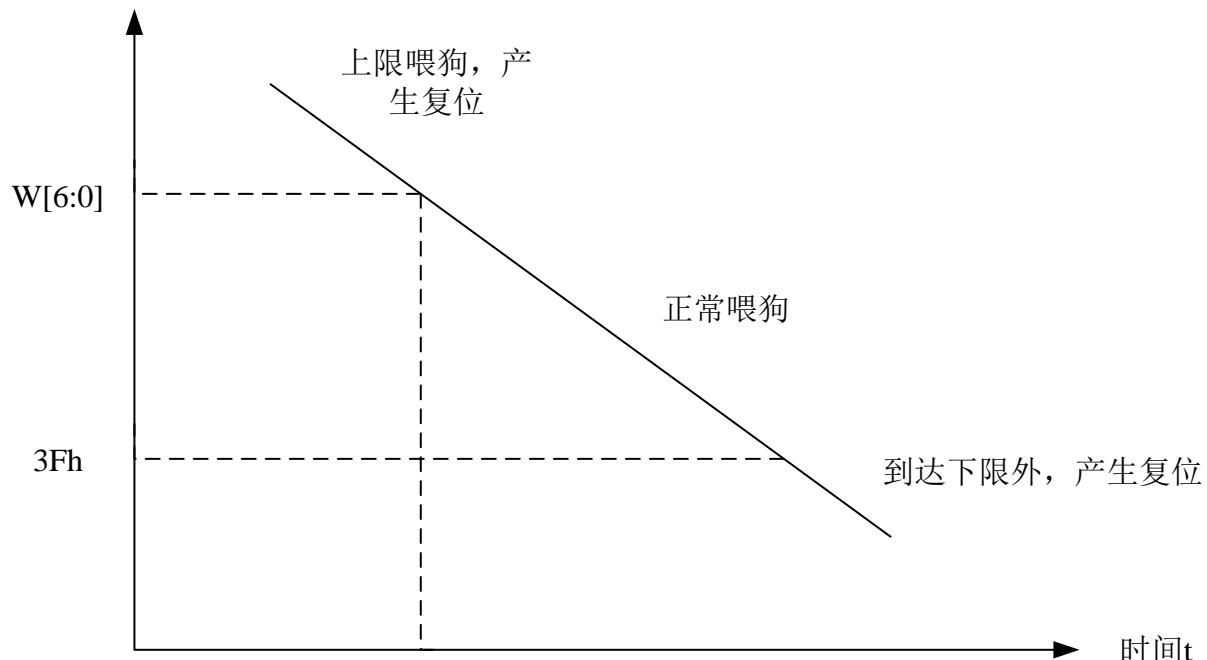


图 17-2 WWDG 超时时间坐标图

表 17-1 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：

$$T = T_{\text{pclk1}} * 4096 * 2^{\text{WDGTB}} * (T[5:0] + 1)$$

WDGTB：分频器的时基

T_{pclk1} ：APB1 的时钟间隔

17.6 寄存器描述

表 17-2 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

17.6.1 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	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值。每 ($4096 \times 2^{\text{WDGTB}}$) 个 PCLK1 周期减 1, 当计数值从 40h 递减为 3Fh 时 (T[6]变成 0), 产生看门狗复位。

17.6.2 配置寄存器 (WWDG_CFR)

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				
rw										rw	rw				

Bit	Field	Description
31:10	Reserved	保留, 必须保持复位值。

Bit	Field	Description
9	EWI	<p>提前唤醒中断使能 (Early Wakeup Interrupt)</p> <p>1: 当计数器值达到 40h, 产生中断。</p> <p>0: 看门狗中断禁止</p> <p>该位只能由硬件在复位后清除。</p>
8: 7	WDGTB	<p>时基 (Timer Base)</p> <p>预分频器的时基可根据如下修改:</p> <p>00: 计时器时钟 (PCLK1 除以 4096) 除以 1</p> <p>01: 计时器时钟 (PCLK1 除以 4096) 除以 2</p> <p>10: 计时器时钟 (PCLK1 除以 4096) 除以 4</p> <p>11: 计时器时钟 (PCLK1 除以 4096) 除以 8</p>
6: 0	W[6:0]	<p>7 位窗口值 (7-Bit Window Value)</p> <p>窗口看门狗的上限窗口值。</p>

17.6.3 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.															

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值。
0	EWIF	<p>EWIF: 提前唤醒中断标志 (Early Wakeup Interrupt Flag)</p> <p>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。</p>

18 RTC 实时时钟器

18.1 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。

18.2 主要特征

- 0 到 2^{20} 的可编程控制预分频系数；
- 32 位的可编程计数器；
- 包含 2 个独立的时钟：RTC 时钟以及用于 APB1 接口的 PCLK1，其中 RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一以上；
- RTC 的时钟源有 3 种：
 - ◆ HSE 时钟的 128 分频；
 - ◆ 外部低速 LSE 振荡器时钟；
 - ◆ 内部低速 LSI 振荡器时钟；
- 3 个可屏蔽中断：
 - ◆ 可编程控制产生的闹钟中断；
 - ◆ 可编程控制产生的秒中断（最长时间为 1 秒）；
 - ◆ 计数溢出中断，内部可编程计数器溢出并返回为 0 的状态；
- 2 个独立的复位系统：
 - ◆ 预分频器、闹钟、毫秒闹钟、计数器和分频器（RTC 核心部分）只能通过上电复位或备份域软件复位；
 - ◆ APB1 接口部分通过系统复位；

18.3 功能描述

18.3.1 功能框图

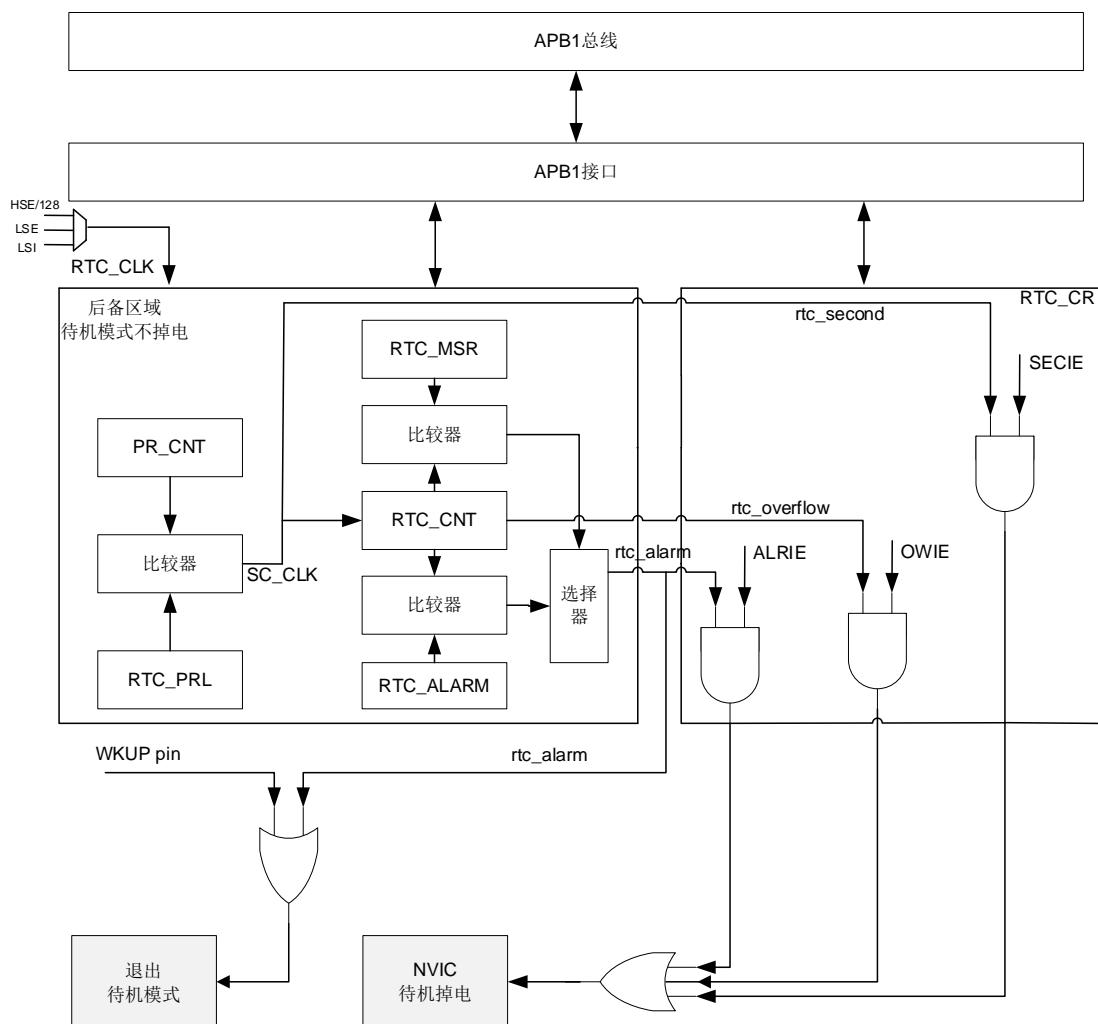


图 18-1 RTC 功能框图

18.3.2 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。

18.3.3 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。

18.3.4 寄存器读取

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

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

可能造成这种情况的几种情形如下：

- 从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。

18.3.5 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；

18.3.6 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。

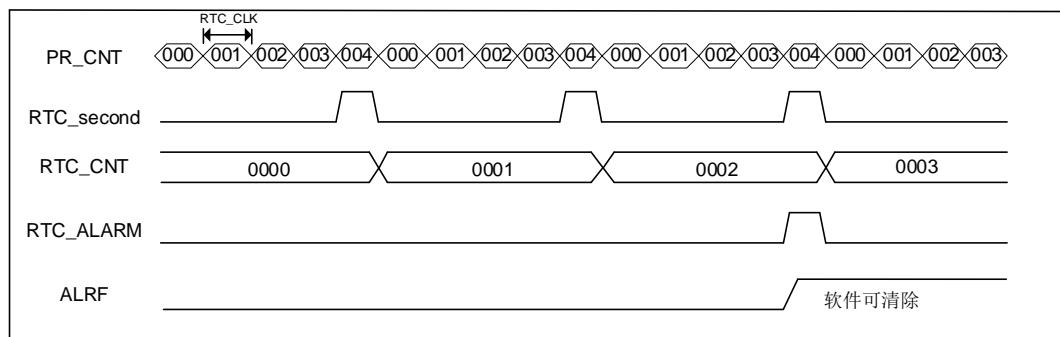


图 18-2 RTC 秒和闹钟波形图示例，RTC_PRL = 0004，RTC_ALR = 002

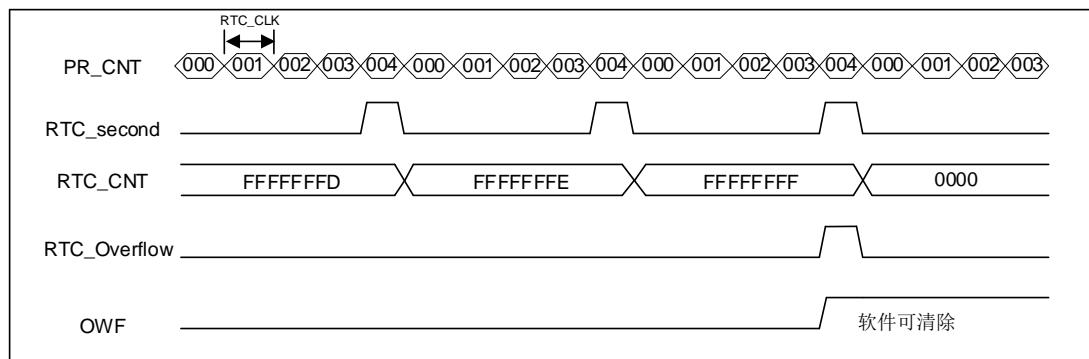


图 18-3 RTC 溢出波形图示例，RTC_PRL = 0004

18.3.7 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。

18.3.8 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。

18.4 寄存器描述

18.4.1 寄存器总览

表 18-1 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_PRLL	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
0x3C	RTC_LSE_CFG	RTC LSE 配置寄存器	0x00000250

18.4.2 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

18.4.3 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
												rw	r	rw	w0c
												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	<p>配置标志（Configuration Flag）</p> <p>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_PRLL/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。</p> <p>0: 退出配置模式 1: 进入配置模式</p>
3	RSF	<p>寄存器同步标志（Registers Synchronized Flag）</p> <p>由软件清零，由硬件置 1。</p> <p>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。</p> <p>0: 寄存器没有同步 1: 寄存器已经同步</p>
2	OWF	<p>溢出标志（Overflow Flag）</p> <p>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。</p> <p>软件清除该位</p> <p>0: 无溢出 1: 32 位可编程计数器溢出</p>
1	ALRF	<p>闹钟标志（Alarm Flag）</p> <p>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。</p> <p>软件清除该位</p> <p>0: 无闹钟产生 1: 有闹钟产生</p>
0	SECF	<p>秒标志（Second Flag）</p> <p>预分频器溢出时，硬件置'1'该位，RTC 计数器加 1。</p> <p>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）</p> <p>通过配置 RTC_CTH 寄存器中的 SECIE 位为 1，可以产生中断。</p> <p>软件清除该位</p> <p>0: 没有秒标志 1: 产生秒标志</p>

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；

18.4.4 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 预分频器重装载值高位 (RTC Prescaler Reload Value High) 计数器的时钟频率计算公式如下: $f_{SC_CLK} = f_{RTC_CLK} / (PRL[19:0] + 1)$ 注: 不推荐配置为 0, 会产生错误的 RTC 中断和标志位。

18.4.5 预分频装载寄存器低位(RTC_PRLL)

偏移地址: 0x0C

复位值: 0x8000

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

Bit	Field	Description
15:0	PRL[15:0]	RTC 预分频器重装载值低位 (RTC Prescaler Reload Value Low)

18.4.6 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]	<p>RTC 时钟分频器分频因子高位 (RTC Clock Divider High)</p> <p>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。</p> <p>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。</p>

18.4.7 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]	<p>RTC 时钟分频器分频因子低位 (RTC Clock Divider Low)</p> <p>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。</p> <p>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。</p>

18.4.8 计数器寄存器高位(RTC_CNTH)

偏移地址: 0x18

复位值: 0x0000

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

Bit	Field	Description
15:0	CNT[31:16]	<p>RTC 计数器高位 (RTC Counter High)</p> <p>通过读 RTC_CNTH 寄存器获取 RTC 计数器当前值的高位部分。先进入配置模式。才能对该寄存器执行写操作。</p> <p>该寄存器受 RTC_CR 的位 RTOFF 写保护。</p> <p>写入该寄存器的值，会直接加载到对应的编程计数器。软件可通过读取该寄存器的值，获得计数器计数值或显示系统时间。</p>

18.4.9 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>

18.4.10 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>

18.4.11 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>

18.4.12 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。

18.4.13 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。

18.4.14 RTC LSE 配置寄存器 (RTC_LSE_CFG)

偏移地址: 0x3C

复位值: 0x0250

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						LSE_IB	LSE_RFB_SEL	LSE_DR	LSE_TEST						
						RW	RW	RW	RW						

Bit	Field	Description
15:10	Reserved	保留，必须保持复位值。
9:8	LSE_IB	LSE 偏置电流调节： 从 00~11 调节逐渐变大，默认配置为 10。
7:6	LSE_RFB_SEL	反馈电阻选择： 00: 12MΩ 01: 10MΩ 10: 6MΩ 11: 3MΩ
5:4	LSE_DR	LSE 驱动能力选择 从 00~11 驱动能力逐渐变大，默认配置为 01。
3:0	LSE_TEST	LSE 测试控制信号 LSE_TEST[3]: 0: 正常模式 1: 输出增强模式 LSE_TEST[2]:保留 LSE_TEST[1:0]: 00: 正常模式 01: 保留 10: 保留 11: 漏电补偿模式

19 SPI_I2S 串行外设接口

19.1 SPI_I2S 功能框图

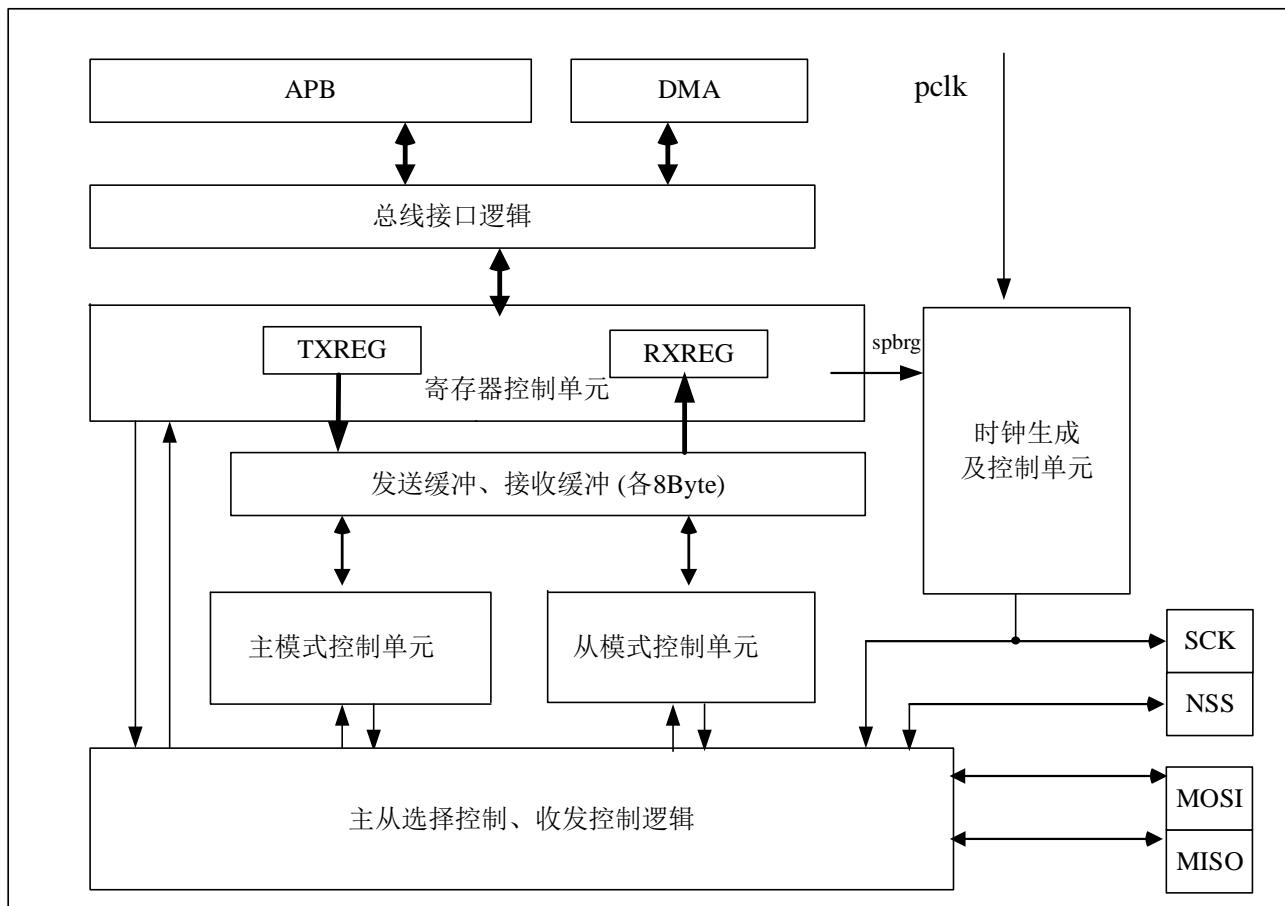


图 19-1 SPI_I2S 功能框图

19.2 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，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，因而避免了因时差引发的失真，所以广泛应用于多媒体系统上各音频设备之间的数据传输。

19.3 SPI 功能描述

19.3.1 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：

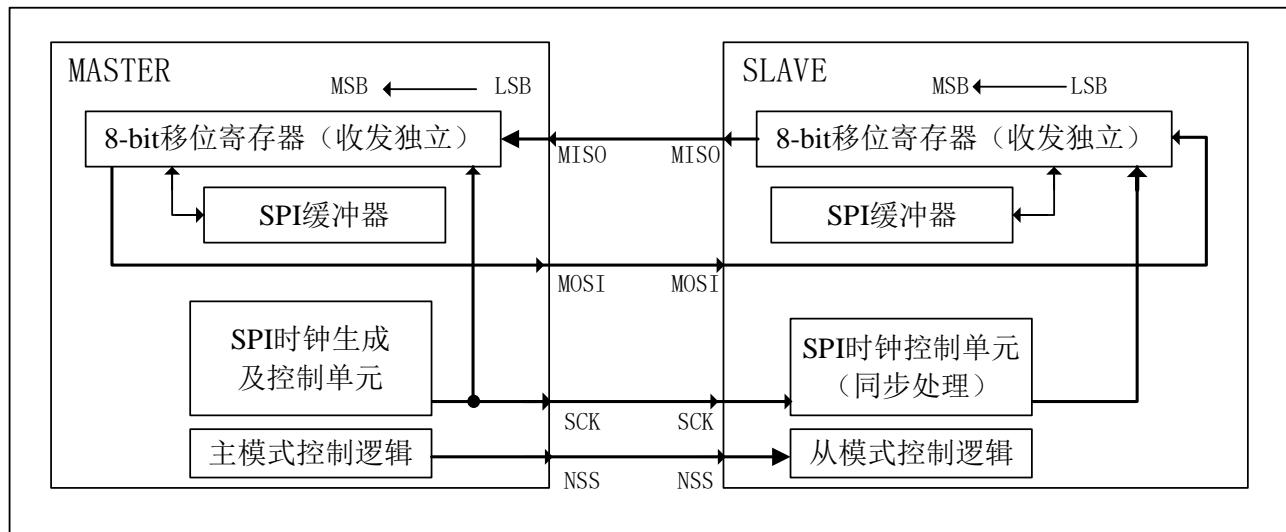


图 19-2 单主和单从应用

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。

19.3.1.1 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。

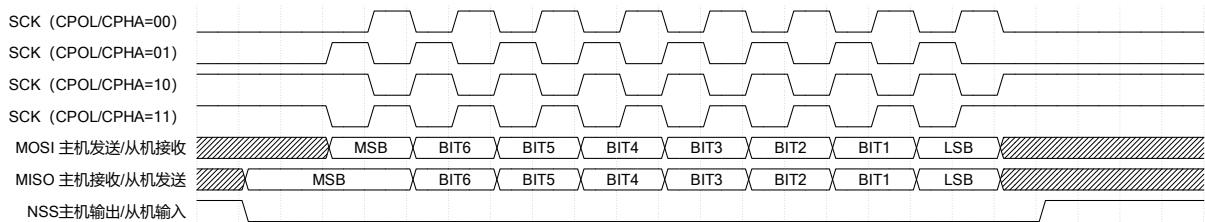


图 19-3 数据时钟时序图

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）。

19.3.1.2 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 ($\geq 10\text{MHz}$)；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 ($< 10\text{MHz}$)。

- 在从模式下，TXEDGE 位为‘1’时，软件一旦配置 TXREG 写入数据，不等待 SCK 时钟输入的边沿而是立即发送到 MISO 引脚线上；TXEDGE 位为‘0’时，从设备总是等到一个有效时钟边沿才发

送数据到 MISO 引脚线上。

- 在主模式下，RXEDGE 位为‘1’时，将后移采样数据的时间点，在传输数据位的尾时钟沿采样数据；RXEDGE 位为‘0’时，在传输数据位的中间点上采样数据。

19.3.1.3 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。

19.3.2 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

19.3.3 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）。

19.3.3.1 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。
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。

19.3.3.2 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。

19.3.3.3 数据接收过程

从设备接收到 MOSI 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也同时会置位 SPI_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。

19.3.4 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。

19.3.4.1 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（当接收到指定个字节数后，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。

19.3.4.2 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）。

19.3.4.3 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）。

19.3.5 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 19-1 波特率公式

模式	公式
SPI 模式	波特率 = Fpclk/X

19.3.6 中断

19.3.6.1 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 19-2 SPI 状态

分类	状态标志	缓冲和信号状态
中断状态	TX_INTF	发送缓冲为空，根据 DW8_32 配置能完成一次发送数据寄存器 TXREG 的写操作
	RX_INTF	根据 DW8_32 设置，至少有一个有效数据的数据，能完成一次接收数据寄存器 RXREG 的读操作
	UNDERRUN_INTF	发送缓冲空且重复发送
	RXOERR_INTF	接收缓冲非空且被覆盖
	RXMATCH_INTF	非空，指定个数的最后 1 个数据传到接收缓冲中（主模式下有效）
	RXFULL_INTF	接收缓冲满，不能再接收新的数据
	TXEPT_INTF	发送缓冲和发送移位寄存器都为空
当前状态	RXAVL_4BYTE	接收缓冲有超过 4 字节有效数据
	TXFULL	发送缓冲满
	TXEPT	发送缓冲和发送移位寄存器都为空
	RXAVL	接收缓冲非空

19.3.7 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，即发送缓冲未满时就发起 DMA 请求；当配置 DW8_32 位为‘1’，发送缓冲有 1 个有效数据的空闲空间时，会发起 DMA 请求。每次请求只进行一次 DMA 传输，且传输数据字节数由 DW8_32 位决定。
- 接收时：当配置 DW8_32 位为‘0’，接收缓冲有大于等于 1 个有效数据（1 字节）时就发起 DMA 请求；当配置 DW8_32 位为‘1’，接收缓冲有 1 个有效数据后才发起 DMA 请求。每次请求只进行一次 DMA 传输。且传输数据字节数由 DW8_32 位决定。

注意：DW8_32=1 时，1 个有效数据的字节数由帧长位数决定（参考 19.3.6.1 状态标志）。

19.4 I2S 功能描述

19.4.1 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)

19.4.2 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）。

19.4.3 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 位数据包帧上发送 16 位有效数据或 24 位有效数据时，软件应注意配置无效数据位为‘0’（例如，针对数据 0x12345678，只发送 16 位有效数据则写 TXREG 为 0x00005678，发送 24 位有效数据则写 0x00345678）。

19.4.4 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 I2SSSTD[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）。

19.4.4.1 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。

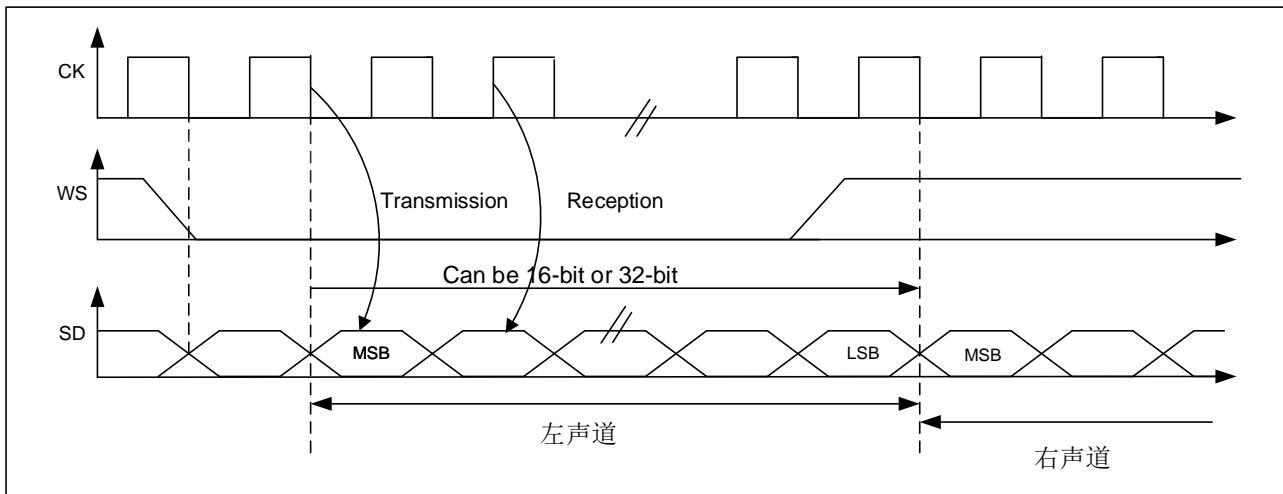


图 19-4 飞利浦标准示意图

19.4.4.2 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。

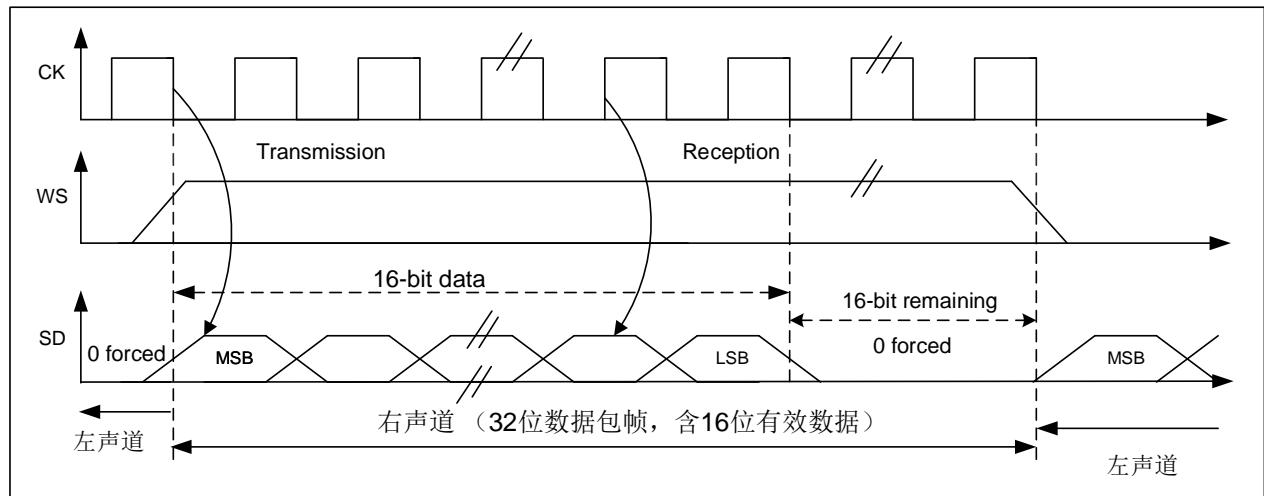


图 19-5 MSB 对齐标准示意图

19.4.4.3 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。

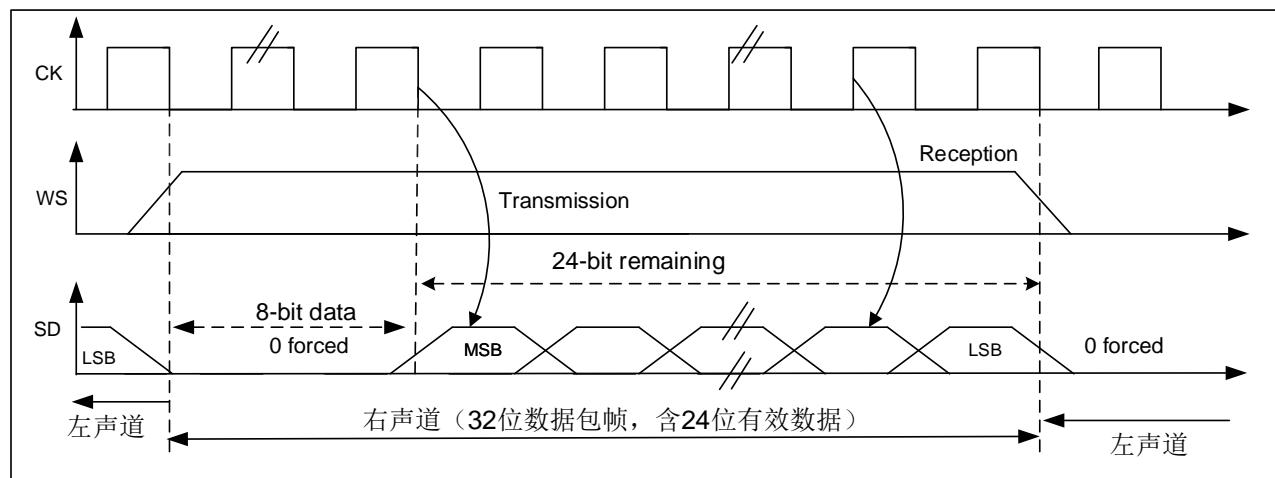


图 19-6 LSB 对齐标准示意图

19.4.4.4 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.PCM_SYNC 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。

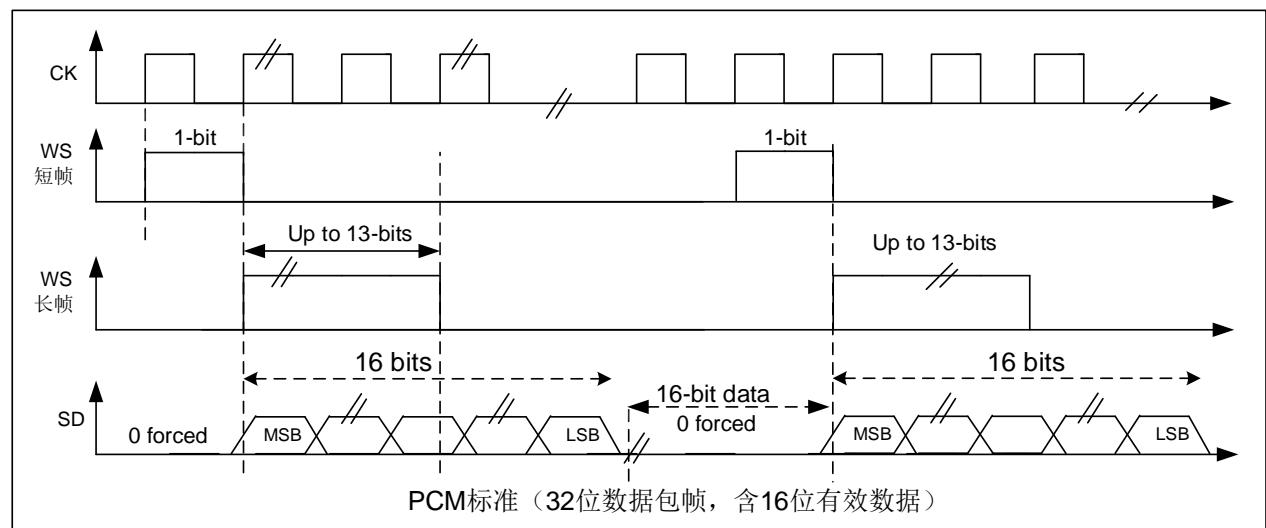


图 19-7 PCM 标准示意图

19.4.5 从模式

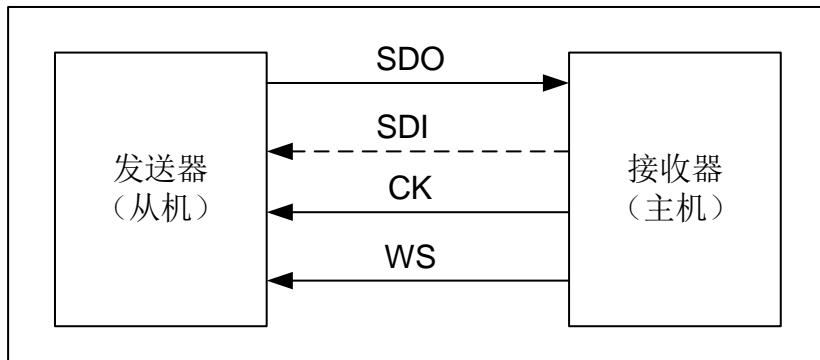


图 19-8 I2S 从模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器)

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;
3. 配置 SPI_I2S_CCTL.CPOL 位, 选择时钟空闲状态时的输入电平;
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 19.4.7 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。

19.4.6 主模式

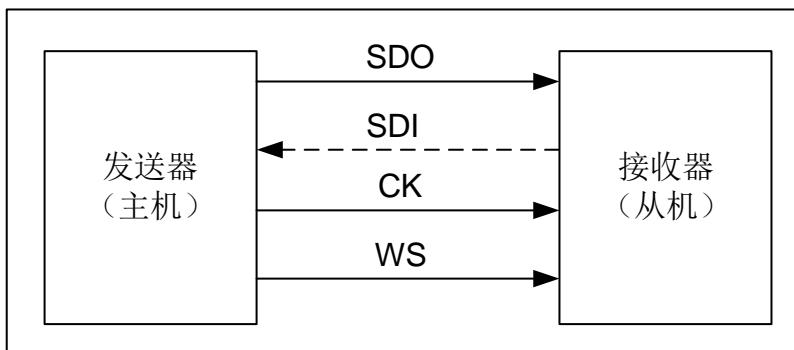


图 19-9 I2S 主模式 (SD 配置成 SDO 为发送器, 配置成 SDI 为接收器)

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 19.4.7 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 I2SSSTD[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）。

19.4.7 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：

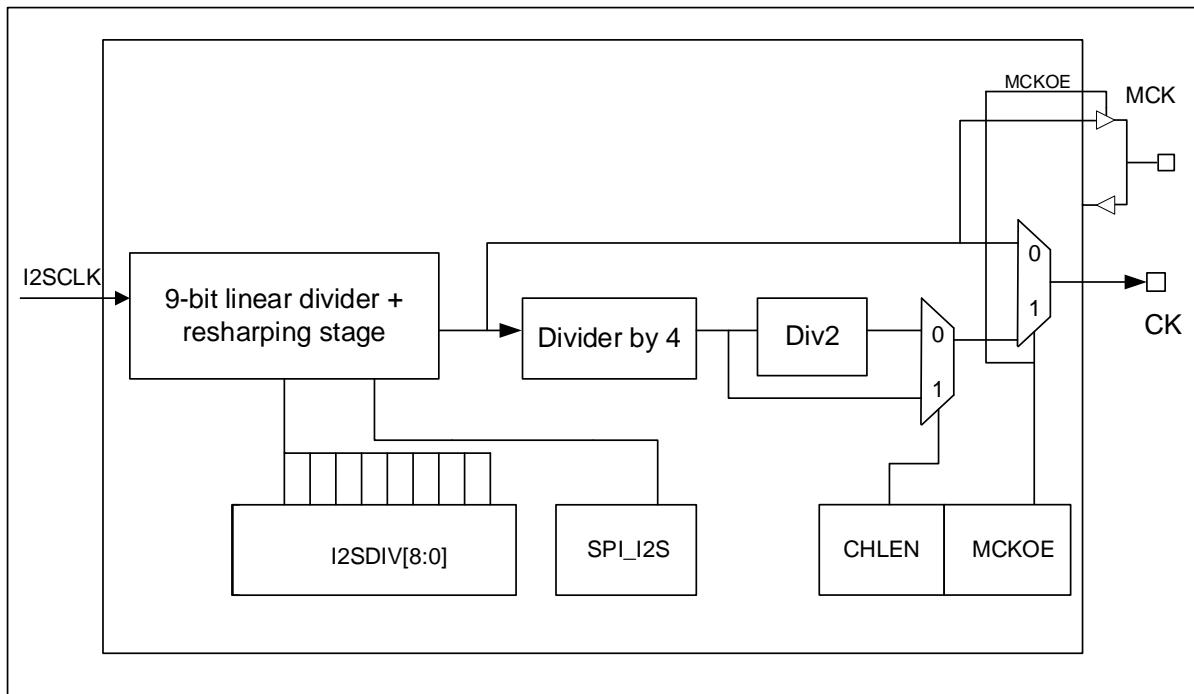


图 19-10 I2S 时钟预分频器示意图

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 19-3 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：

$$F_s = I2S \text{ 比特率} / (\text{通道长度} \times \text{通道数}) = F_{CK} / (\text{通道长度} \times \text{通道数})$$

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 19-4 I2S 音频采样率 F_s 与 F_{I2SCLK} 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 $F_s = 192KHz$, $I2SDIV[8:0] = 1$ ，则要求 APB 时钟 (F_{I2SCLK}) = $1 \times 256 \times 192KHz = 49.152MHz$ 。

针对常用的音频采样率，当按照 $F_{I2SCLK}=48MHz$ 计算时，误差如下：

表 19-5 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

		MCKOE=1					MCKOE=0			
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	
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	

19.4.8 中断

19.4.8.1 状态标志

与 SPI 共用同一个中断向量

表 19-6 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。

19.4.9 DMA 传输

在 I2S 模式, DMA 的工作方式与 SPI 模式完全相同。

19.5 寄存器

19.5.1 寄存器总览

表 19-7 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

19.5.2 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	Description
31: 0	TXREG	发送数据寄存器 (Transmit data register) 有效数据位由 DW8_32 控制: DW8_32=0 时, 只有低 8 位有效 DW8_32=1 时, TXREG[31: 0]都有效

19.5.3 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	Description
31: 0	RXREG	<p>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。</p>

19.5.4 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.															
r															
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 r r r r r r r r															

Bit	Field	Description
31: 14	Reserved	始终读为 0
13	CHSIDE	<p>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。</p>
12	BUSY	<p>忙标志位 表示 I2S 或 SPI 正在传输中</p>
11: 8	RXFADDR	当前接收缓冲中有效字节数
7: 4	TXFADDR	当前发送缓冲中有效字节数
3	RXAVL_4BYTE	<p>接收缓冲中有效数据达到 4 个字节标志位(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)。</p>

Bit	Field	Description
2	TXFULL	发送缓冲满标志位 (Transmitter FIFO full status bit) 0: 发送缓冲未满 1: 发送缓冲满
1	RXAVL	接收有效数据标志位 (Receive available byte data message) 当接收缓冲中接收到一个字节数据时置位该位。 0: 接收缓冲空 1: 接收缓冲非空 注: 该位只读, 由硬件自动置位和清除。
0	TXEPT	发送端空标志位 (Transmitter empty bit) 0: 发送缓冲或发送移位寄存器不为空 1: 发送缓冲和发送移位寄存器都为空 注: 该位只读, 由硬件自动置位和清除。

19.5.5 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	Description
31: 8	Reserved	保留, 始终读为 0
7	FRE_INTF	帧传输错误中断标志位: 表示 I2S 传输协议不匹配 (仅从模式有效)
6	TXEPT_INTF	发送端空中断标志位 (Transmitter empty interrupt flag bit) 硬件自动置位, 写 INTCLR.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。
5	RXFULL_INTF	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

Bit	Field	Description
4	RXMATCH_INTF	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	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	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	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 19.3.6.1 状态标志。
0	TX_INTF	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

19.5.6 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_IE N	TXEPT_IEN	RXFULL_IEN	RXMAT_CH_IEN	RXOER_R_IEN	UNDER_RUN_IE_N	RX_IEN	TX_IEN
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留, 始终读为 0

Bit	Field	Description
7	FRE_IEN	帧传输错误中断使能 (FRE_IEN): 0: 禁止中断 1: 中断使能
6	TXEPT_IEN	发送端空中断使能位 (Transmit empty interrupt enable bit) 0: 禁止中断 1: 中断使能
5	RXFULL_IEN	接收缓冲满中断使能位 (Receive FIFO full interrupt enable bit) 0: 禁止中断 1: 中断使能
4	RXMATCH_IEN	接收指定字节数中断使能位 (Receive data complete interrupt enable bit) 0: 禁止中断 1: 中断使能
3	RXOERR_IEN	接收端溢出错误中断使能位 (Overrun error interrupt enable bit) 0: 禁止中断 1: 中断使能
2	UNDERRUN_IEN	SPI 从机模式下溢中断使能位 (SPI 从机模式) (Transmitterunderrun interrupt enable bit (SPI slave mode only)) 0: 禁止中断 1: 中断使能
1	RX_IEN	接收端数据中断使能位 (Receive FIFO interrupt enable bit) 0: 禁止中断 1: 中断使能
0	TX_IEN	发送缓冲空中断使能位 (Transmit FIFO empty interrupt enable bit) 0: 禁止中断 1: 中断使能

19.5.7 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_IC LR	TXEPT_ ICLR	RXFULL_ ICLR	RXMAT CH_ICL R	RXOER R_ICLR	UNDER RUN_IC LR	RX_ICL R	TX_ICL R
W								W	W	W	W	W	W	W	W

Bit	Field	Description
31: 8	Reserved	保留, 始终读为 0

Bit	Field	Description
7	FRE_ICLR	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	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	接收缓冲满中断清除位 (Receiver buffer full interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
4	RXMATCH_ICLR	接收指定字节数中断清除位 (Receive completed interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
3	RXOERR_ICLR	接收端溢出错误中断清除位 (Overrun error interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
2	UNDERRUN_ICLR	SPI 从机模式下溢中断清除位(SPI 从机模式)(Transmitter underrun interrupt clear bit (SPI slave mode only)) 0: 写 0 无意义 1: 写 1 清除中断
1	RX_ICLR	接收端数据中断清除位 (Receive interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
0	TX_ICLR	发送缓冲空中断清除位 (Transmitter FIFO empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断

19.5.8 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		

Bit	Field	Description
31: 18	Reserved	保留, 始终读为 0

Bit	Field	Description																																																																																																																													
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。</p> <p>信号 SCL、MOSI、NSS、MISO 和 PAD 的映射变换如下：</p> <table> <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)</p> <p>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>																																																																																																																													
10	NSS	<p>硬件或软件控制主模式下的 NSS 输出 (NSS select signal that from software or hardware)</p> <p>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。</p>																																																																																																																													
9	DMAMODE	<p>DMA 方式 (DMA Mode Selection bit)</p> <p>0: 正常模式 1: 开启 DMA 模式</p>																																																																																																																													
8: 5	Reserved	预留，必须保持复位值。																																																																																																																													

Bit	Field	Description
4	RXEN	<p>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。</p>
3	TXEN	<p>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。</p>
2	MODE	<p>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)</p>
1	INTEN	<p>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</p>
0	SPIEN	<p>SPI/I2S 选择位 (SPI/I2S select bit) 0: SPI/I2S 禁止 (复位状态) 1: SPI/I2S 使能</p>

19.5.9 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
6	CPHASEL	<p>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。</p>

Bit	Field	Description
5	TXEDGE	<p>发送数据相位调整位（从模式）(Transmit data edge select)</p> <p>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）。</p> <p>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）</p> <p>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。</p>
4	RXEDGE	<p>接收数据采样时钟沿选择位（主模式）(Receive data edge select)</p> <p>0: 在传输数据位的中间采样数据</p> <p>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）</p> <p>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。</p>
3	SPILEN	<p>SPI 数据宽度位 (SPI character length bit)</p> <p>该位在 DW8_32=0 时有效；DW8_32=1 时，该位需保持为 1。</p> <p>0: 7 位数据</p> <p>1: 8 位数据（缺省）</p> <p>注：I2S 模式下固定为 1。</p>
2	LSBFE	<p>LSBFE: LSB 在前使能位 (LSI first enable bit)</p> <p>0: 数据传输或接收时最高位在前</p> <p>1: 数据传输或接收时最低位在前</p> <p>注：I2S 模式下固定为 0。</p>
1	CPOL	<p>时钟极性标志位 (Clock polarity select bit)</p> <p>0: 时钟在空闲状态为低电平（两次传输之间）</p> <p>1: 时钟在空闲状态为高电平（两次传输之间）</p>
0	CPHA	<p>时钟相位选择位 (Clock phase select bit)</p> <p>0: 第一个数据位采样从第二个时钟边沿开始</p> <p>1: 第一个数据位采样从第一个时钟边沿开始</p> <p>注：I2S 模式下固定为 0。</p>

19.5.10 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	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	SPBRG	SPI 波特率控制寄存器用于产生波特率 (SPI baud rate control register for baud rate) 波特率公式: 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。

19.5.11 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	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	RXDNR	该寄存器用于存储下次接收过程需要接收字节的个数 (The register is used to hold a count of to be received bytes in next receive process) 注: 该寄存器的值仅在 SPI 为主机接收模式下有效; 缺省值是 1。 该寄存器值通过软件来改变, 且不能对该寄存器写值为 0。

19.5.12 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	Description
31: 1	Reserved	保留, 必须保持复位值。

Bit	Field	Description
0	NSS	主模式下片选输出信号。低有效，从模式下该位无效（Chip select output signal in Master mode）。 0: 从器件被选中（允许从器件与主模式建立通信） 1: 从器件未选中

19.5.13 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	Description
31: 5	Reserved	保留, 始终读为 0
4: 0	EXTLEN	控制 SPI 数据帧长度的选择 0 0000: 32 bit 0 0001: 1 bit 0 0010: 2 bit 0 0011: 3 bit 1 1100: 28 bit 1 1101: 29 bit 1 1110: 30 bit 1 1111: 31 bit 注: 仅当 SPI_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)。

19.5.14 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.			PCMSYN	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	<p>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)。</p>
15: 12	Reserved	保留, 始终读为 0
11	MCKOE	<p>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</p>
10	SPI_I2S	<p>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。</p>
9: 7	Reserved	保留, 始终读为 0
6	PCMSYNC	<p>PCM 标准帧同步模式 0: 短帧同步模式 1: 长帧同步模式</p>
5: 4	I2SSTD	<p>I2S 标准 00: PHILIPS 标准 01: MSB 对齐标准 10: LSB 对齐标准 11: PCM 标准</p>
3	Reserved	保留, 始终读为 0
2: 1	DATLEN	<p>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。</p>
0	CHLEN	<p>声道长度 (每个声道包含的数据位数) 0: 16 位宽 1: 32 位宽</p>

20 UART 通用异步收发器

20.1 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。

20.2 功能框图

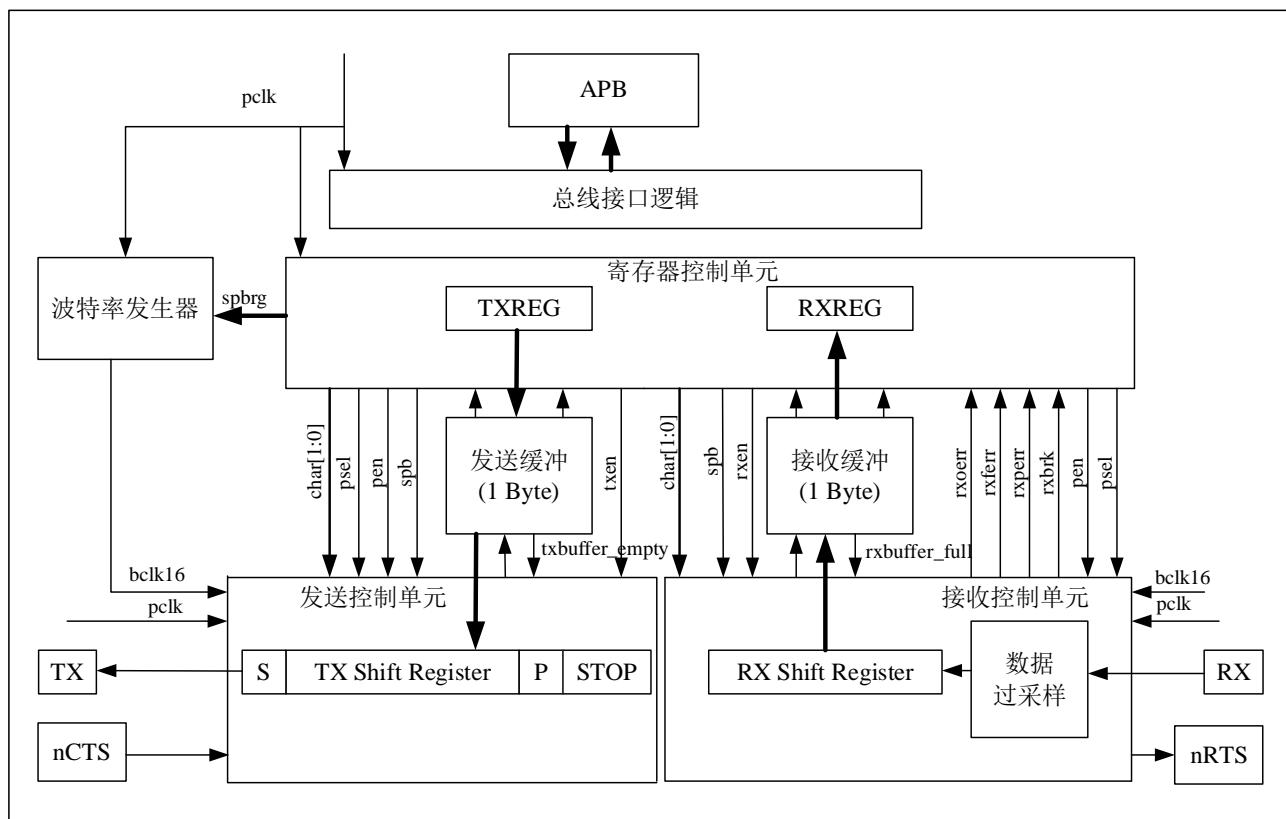


图 20-1 UART 功能框图

20.3 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

20.4 功能描述

在全双工通信的情况下，至少需要分配两个脚给 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。当发送器被禁用时，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。

20.4.1 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。

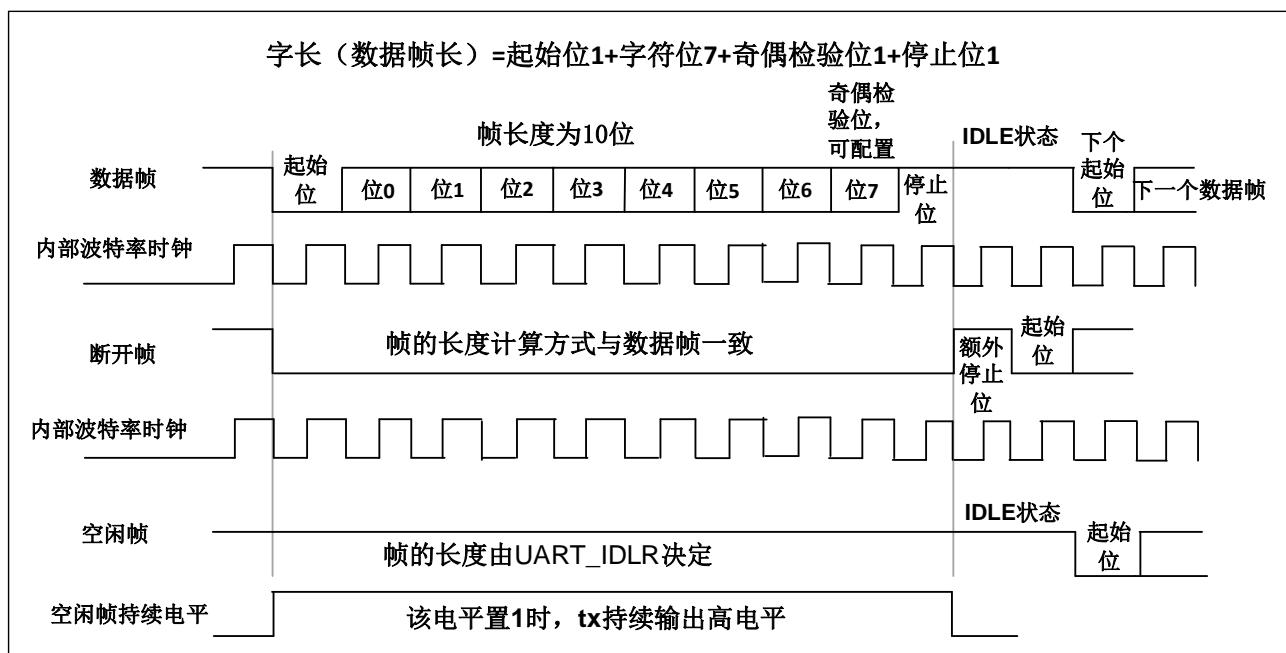


图 20-2 UART 时序

20.4.2 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：

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

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

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

其中 `BRR` 寄存器最小值为 1，`FRA` 的值可配置在 0~15 范围。

20.4.3 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。

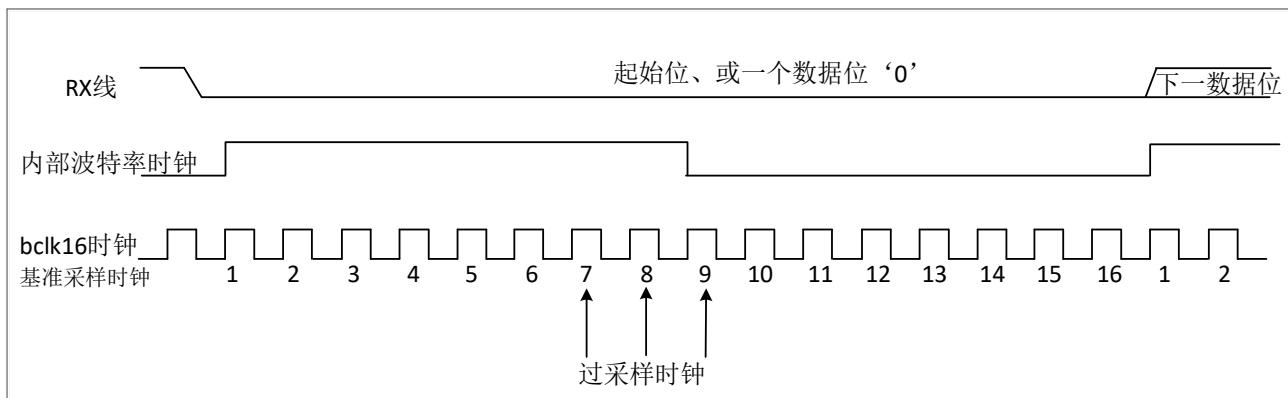


图 20-3 RX 引脚采样方法

20.4.4 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。

20.4.5 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。

20.4.6 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。

20.4.6.1 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。

20.4.6.2 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。

20.4.6.3 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。

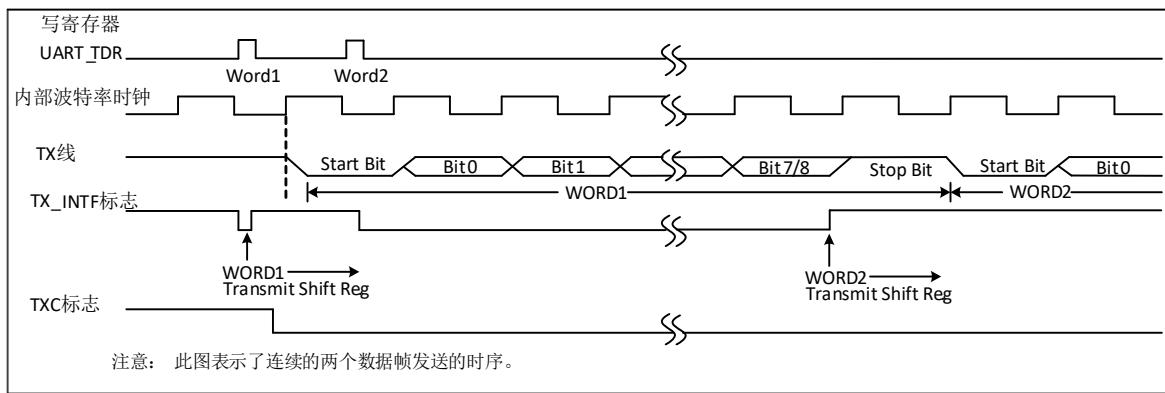


图 20-4 发送时状态位变化

20.4.6.4 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。

20.4.6.5 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。

20.4.7 接收器

20.4.7.1 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。

20.4.7.2 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。

20.4.7.3 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。

20.4.7.4 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。

20.4.7.5 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。

20.4.8 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_ABRCR** 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。

20.4.9 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。

20.4.10 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。

20.4.10.1 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。

20.4.10.2 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。

20.4.10.3 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。

20.4.11 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）。

20.4.12 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。

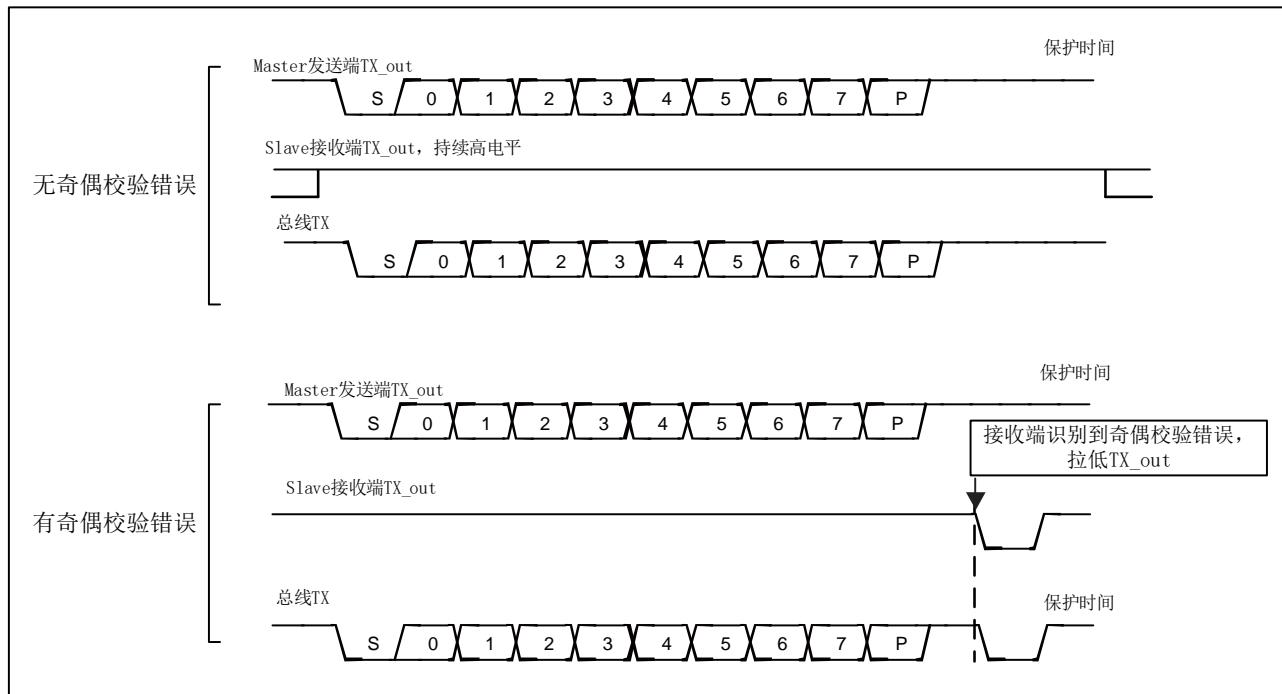


图 20-5 UART 奇偶校验方框图

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）。

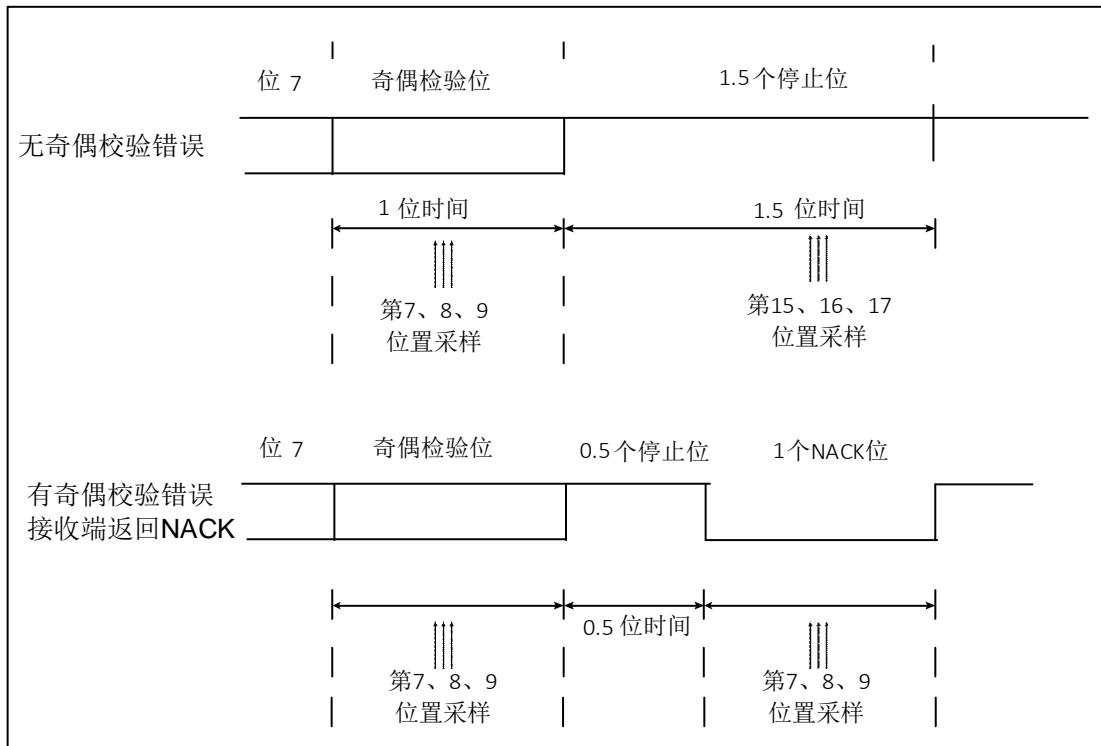


图 20-6 UART 采样 NACK 信号方框图

20.4.13 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 ($1.42 \text{ MHz} < \text{sirlp_clk} < 2.12 \text{ MHz}$)，脉冲的宽度是的 $\text{sirlp_clk} * 3$ ，`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：

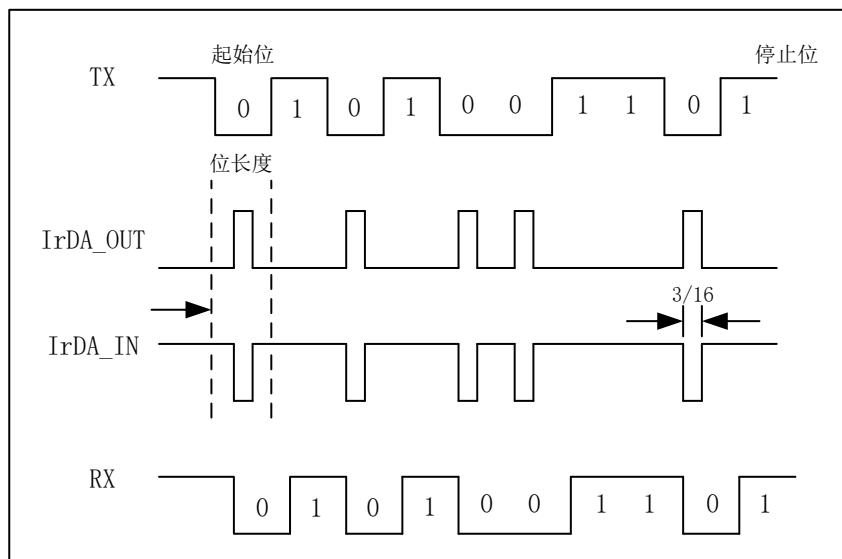


图 20-7 普通模式下 IrDA 发送和接收图

20.5 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 20-1 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	RXPERRLEN
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	ABRENDEIN
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。

20.6 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。

20.7 寄存器

20.7.1 寄存器总览

表 20-2 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

20.7.2 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)

20.7.3 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

20.7.4 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	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	发送结束标识位 (Transmit complete flag bit) 0: 发送不为空 1: 发送缓冲和发送移位寄存器都为空 注: 此标识位在智能卡 (ISO7816) 时与保护时间 (UART_SCR.SCFCNT 配置) 相关。

20.7.5 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				R_INTF	D_INTF	_INTF	NTF	INTF	R_INTF	R_INTF	R_INTF	R_INTF	TF	F	F

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 autoflwen=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

20.7.6 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_I EN	RXBRK _IEN	RXFER R_IEN	RXPER R_IEN	RXOER R_IEN	TXCIE N	RX_IEN	TX_IEN
rw				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

20.7.7 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

20.7.8 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

20.7.9 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	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	<p>停止位选择 (Stop bit selection) 设置发送停止位位数。</p> <p>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</p>
1	PSEL	<p>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</p>
0	PEN	<p>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</p>

20.7.10 UART_BRR 波特率寄存器

偏移地址: 0x20

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
DIV_Mantissa															
rw															

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 0	DIV_Mantissa	<p>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</p>

20.7.11 UART_FRA 分数波特率寄存器

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.	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。

20.7.12 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。

20.7.13 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。

20.7.14 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

20.7.15 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。

20.7.16 UART_ABRCR 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.															

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

20.7.17 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.							
rw								rw							

Bit	Field	Description
31: 16	Reserved	保留, 始终读为 0
15: 8	PSC_REG	<p>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 sirlp_clk:</p> <p>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 < sirlp_clk < 2.12 MHz) 之间。</p>
7: 2	Reserved	保留, 始终读为 0
1	Sirlp	<p>红外低功耗模式 (IrDA low_power)</p> <p>0 : 普通模式 1 : 低功耗模式</p>
0	Siren	<p>IrDA 红外模式使能 (IrDA mode enable)</p> <p>0 : 不使能红外模式 1 : 使能红外模式</p>

21 I2C 内部集成电路接口

21.1 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）。

21.2 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）

21.3 功能描述

21.3.1 功能框图

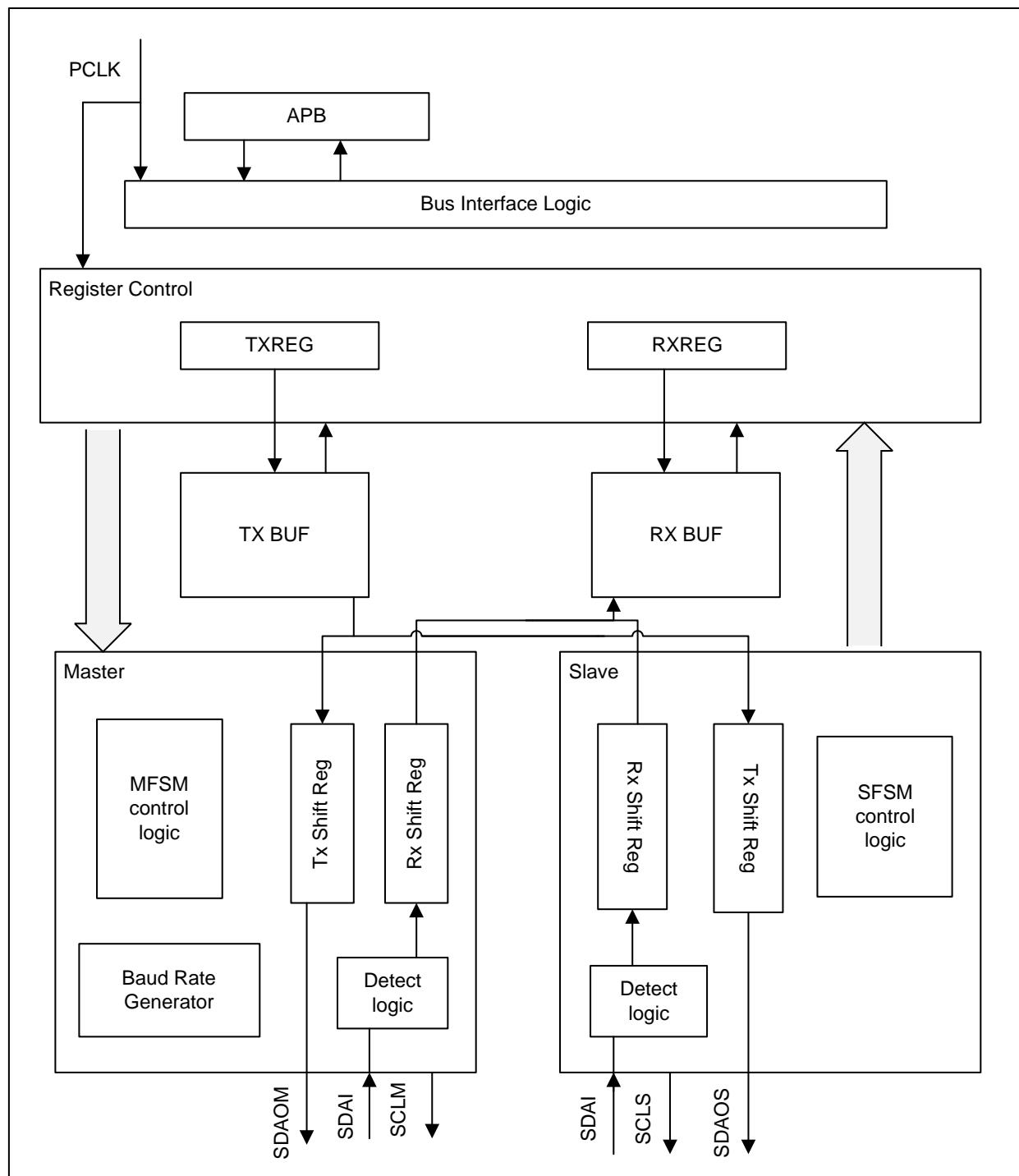


图 21-1 I2C 功能框图

21.3.2 信号描述

表 21-1 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。

21.3.3 I2C 协议

21.3.3.1 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。

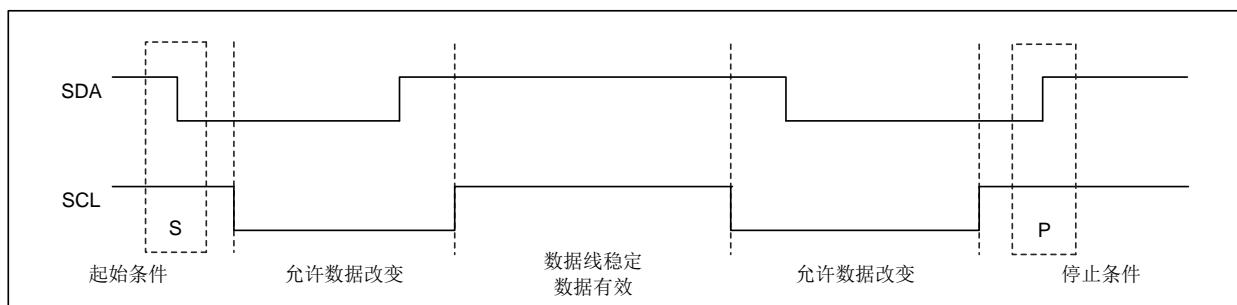


图 21-2 起始和停止条件

21.3.3.2 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。

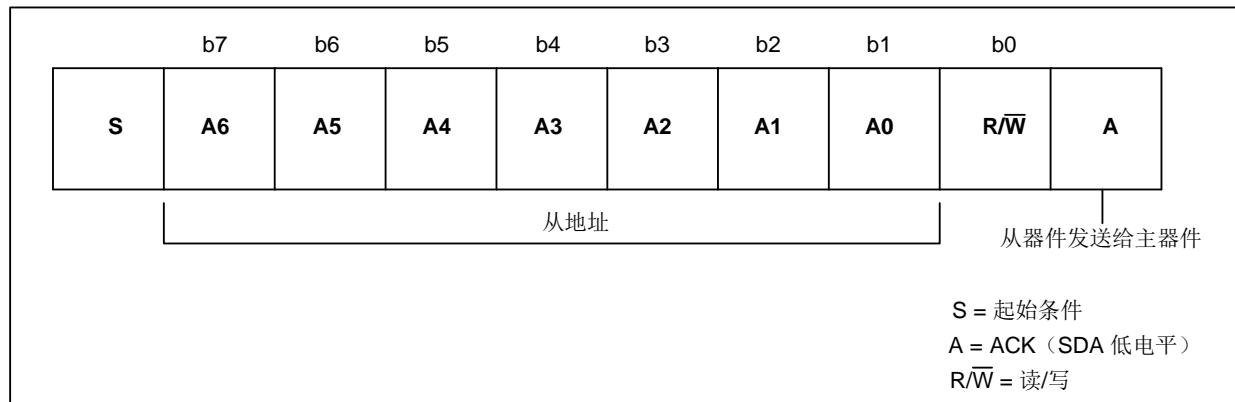


图 21-3 7 位地址格式

- 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：

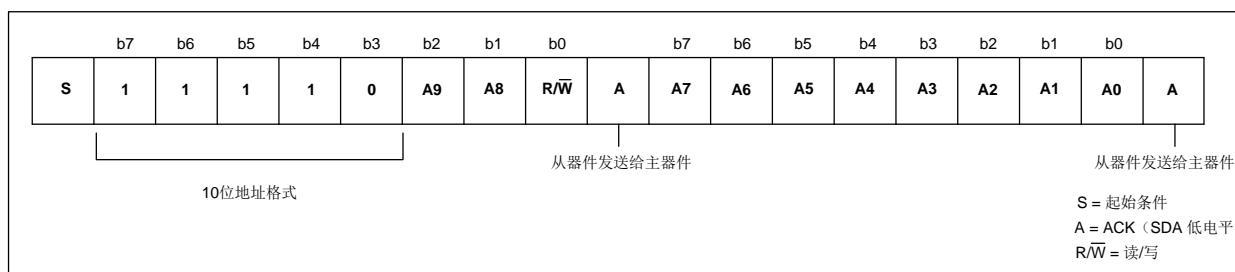


图 21-4 10 位地址格式

下表定义了 I2C 首字节的特殊用途和保留地址。

表 21-2 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

21.3.3.3 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。

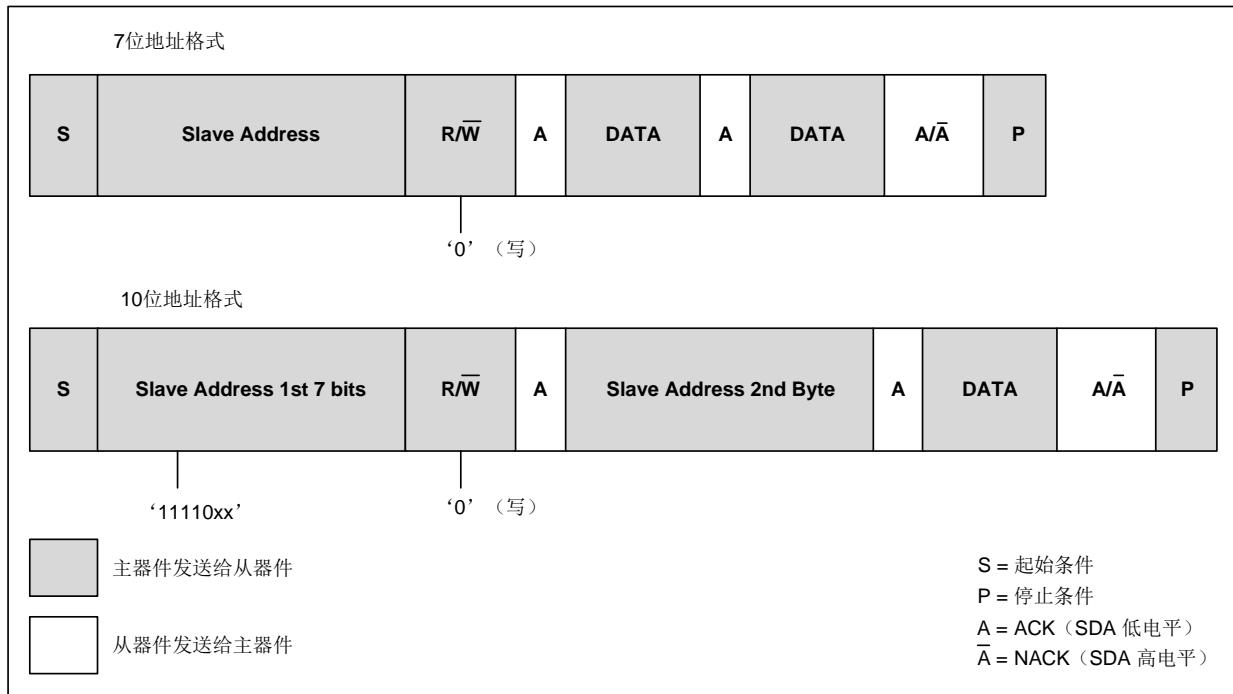


图 21-5 主发送协议

● 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。

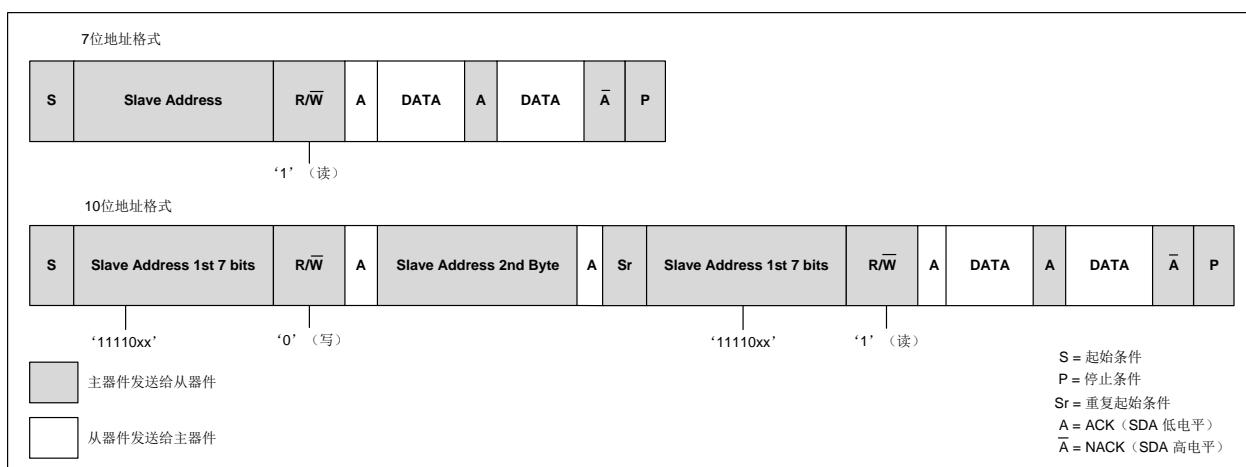


图 21-6 主接收协议

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。

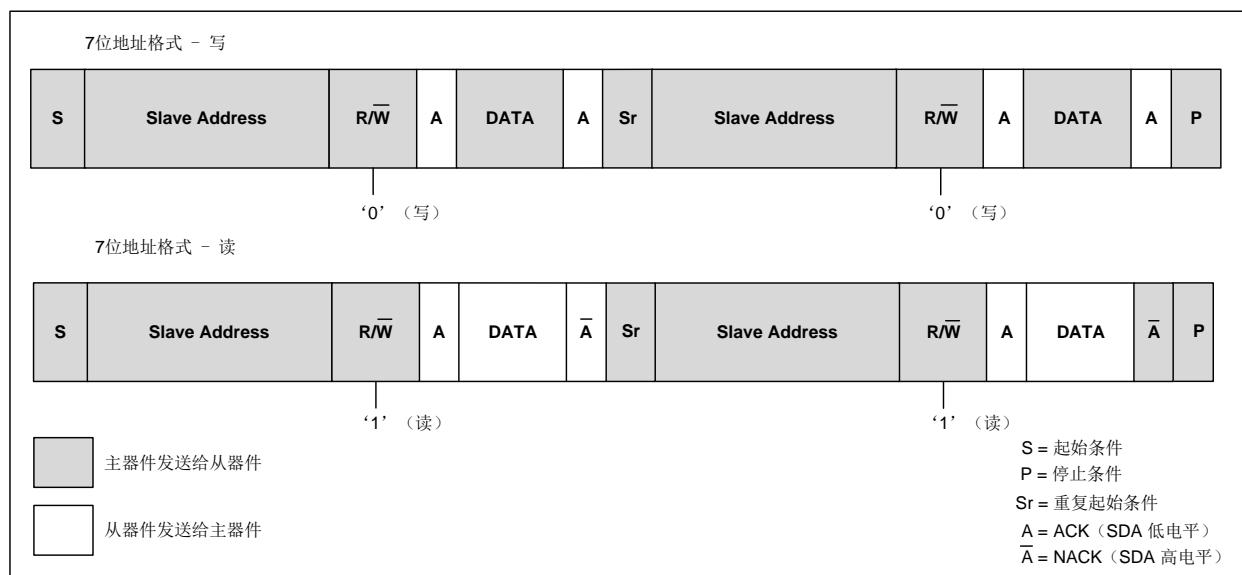


图 21-7 带 RESTART (SR) 信号的主发送和接收协议

● 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。

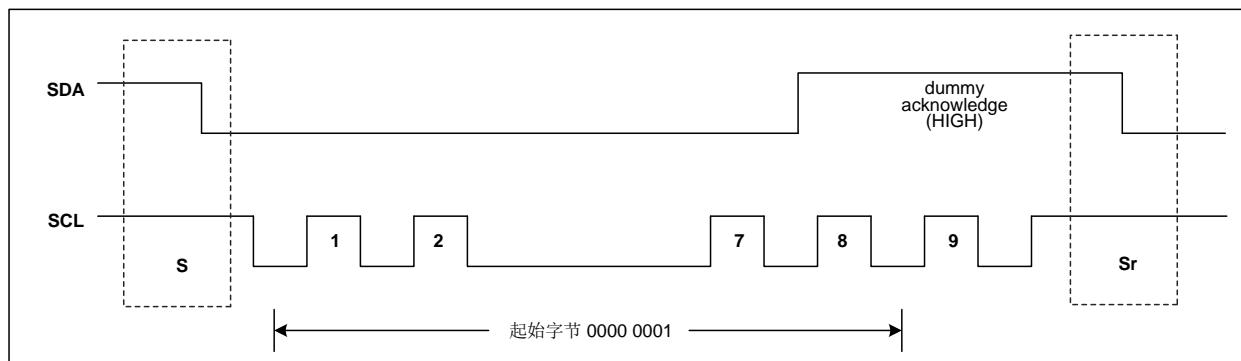


图 21-8 起始字节传输

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。

21.3.3.4 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 21-9 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。

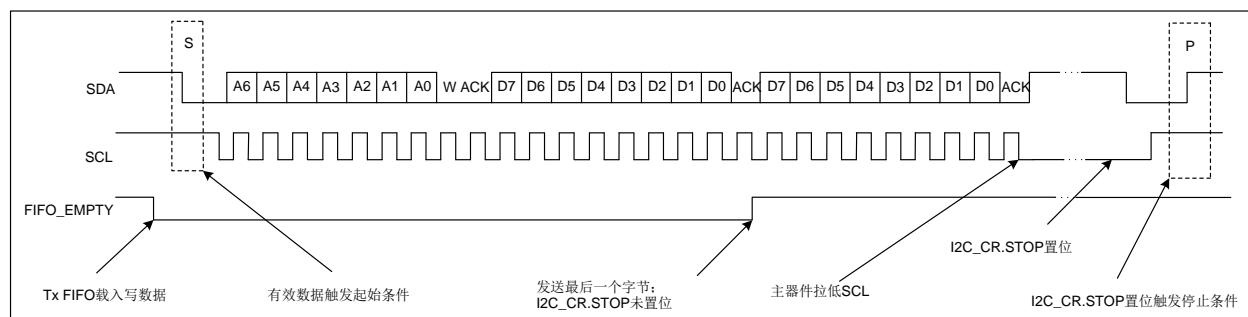


图 21-10 主发送，TX FIFO 为空

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。

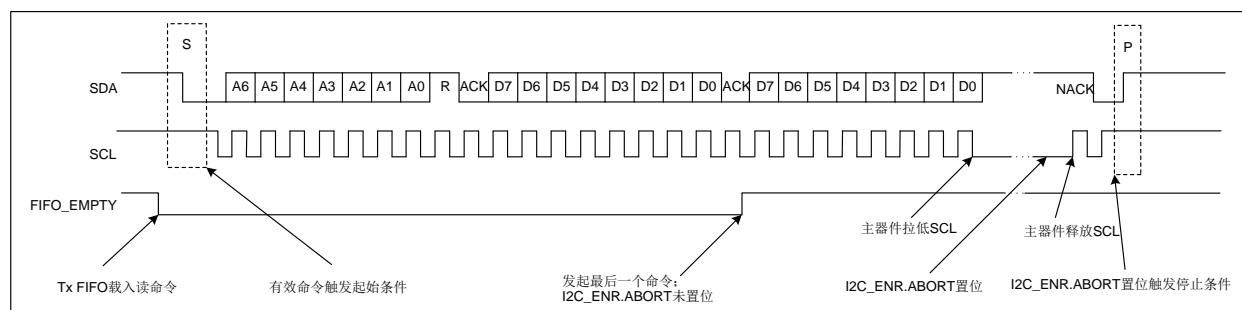


图 21-11 主接收，TX FIFO 为空

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。

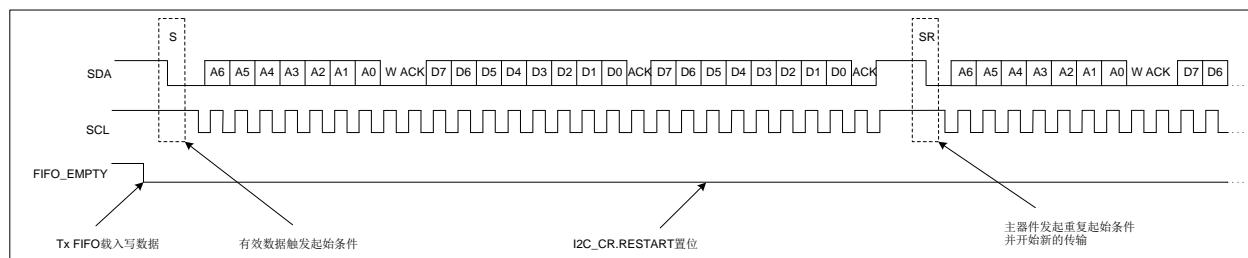


图 21-12 主发送，发生 RESTART

下图描述了 I2C 模块工作在主接收模式下，发生 RESTART 时的时序。

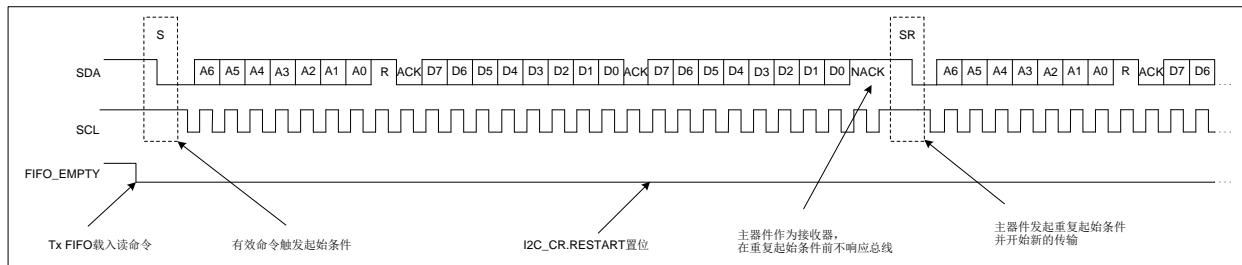


图 21-13 主接收，发生 RESTART

21.3.3.5 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。

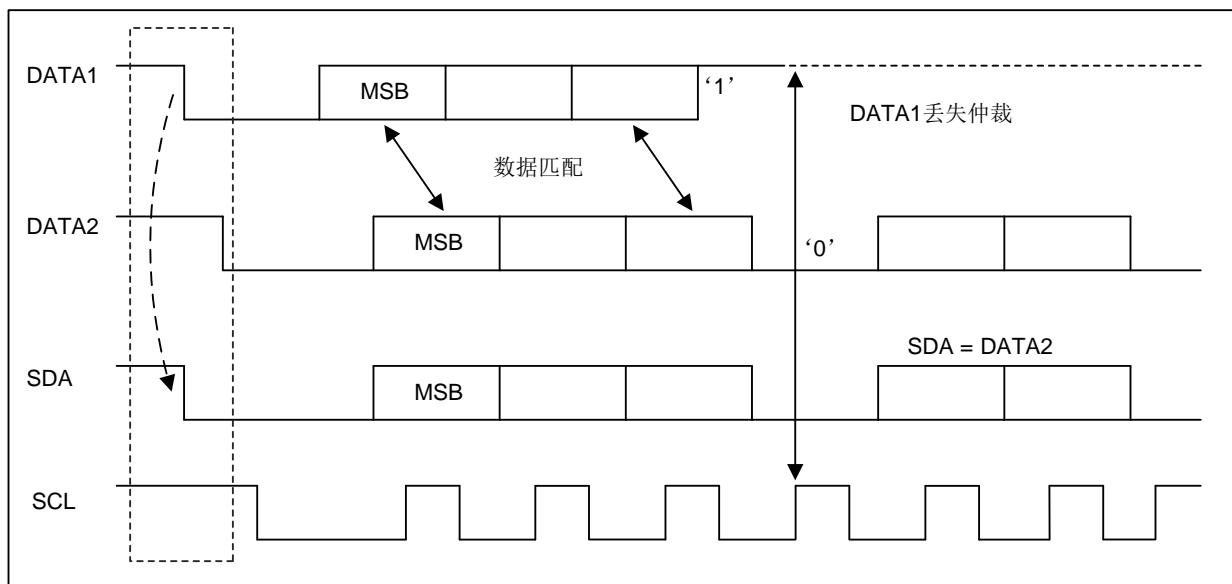


图 21-14 两个主器件仲裁

21.3.3.6 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。

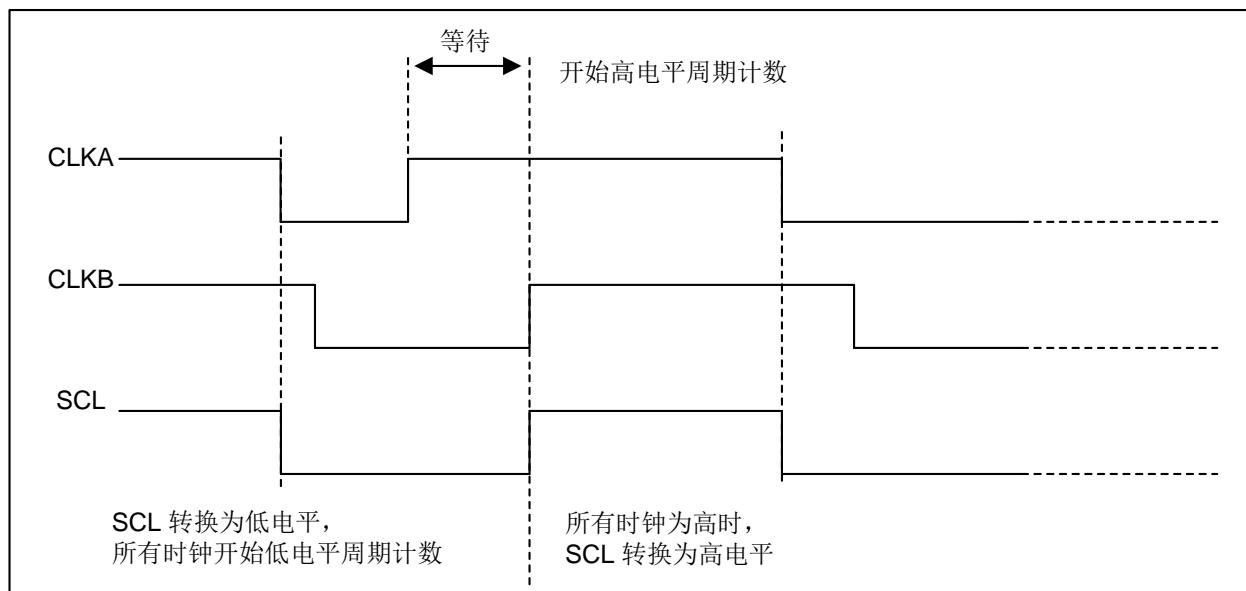


图 21-15 时钟同步（示意图）

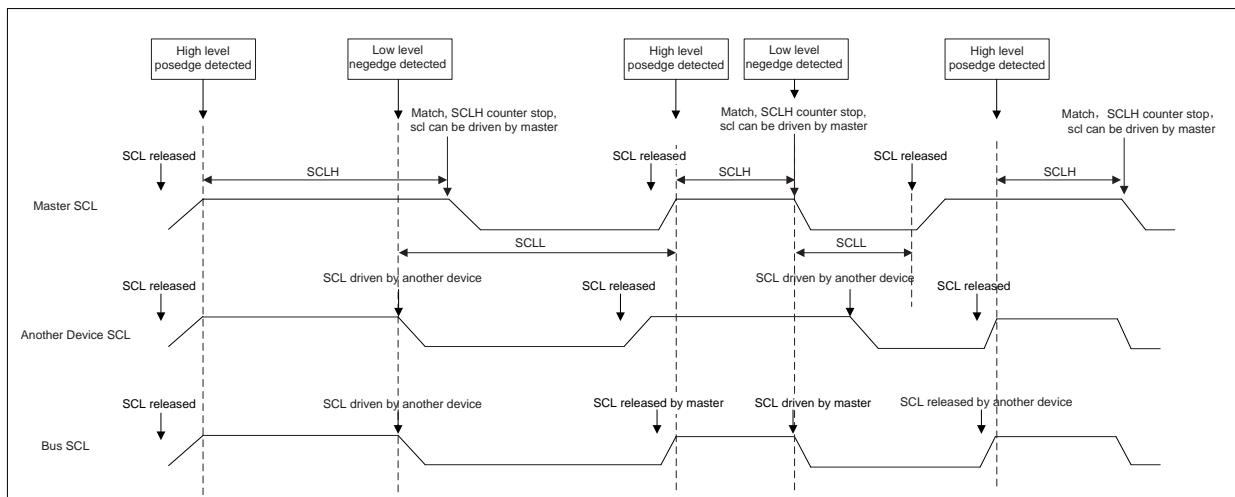


图 21-16 时钟同步（时序图）

21.3.3.7 SCL 配置

I2C 的 SCL 可参考如下配置：

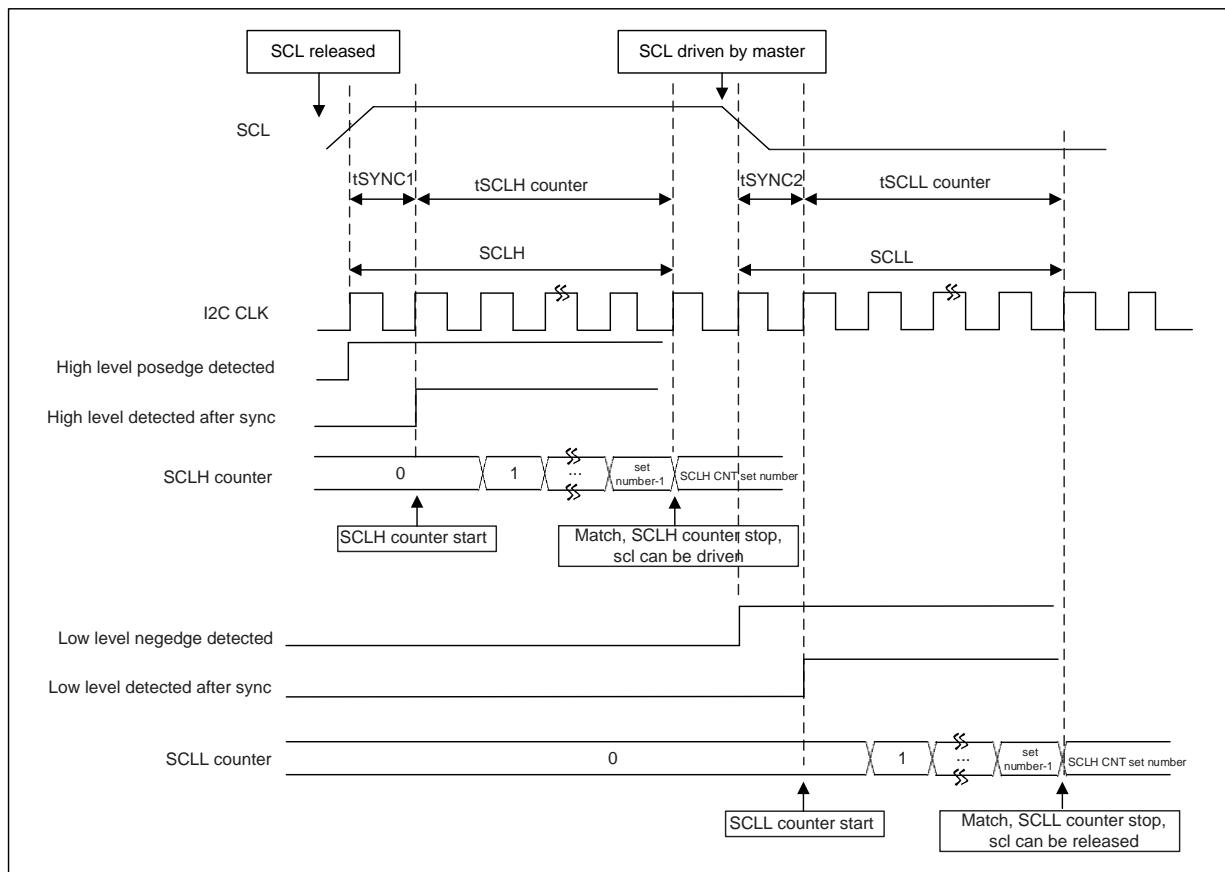


图 21-17 SCL 生成时序

- 标准模式:

$$SCLH = (\text{SSHR} + 12) \times \text{I2C CLK} + t\text{SYNC1}$$

$$SCLL = (\text{SSLR} + 1) \times \text{I2C CLK} + t\text{SYNC2}$$

注: $t\text{SYNC1}$ 为 0~1 个 I2C CLK, $t\text{SYNC2}$ 为 0~1 个 I2C CLK

- 快速模式:

$$SCLH = (\text{FSHR} + 12) \times \text{I2C CLK} + t\text{SYNC1}$$

$$SCLL = (\text{FSLR} + 1) \times \text{I2C CLK} + t\text{SYNC2}$$

注: $t\text{SYNC1}$ 为 2~3 个 I2C CLK, $t\text{SYNC2}$ 为 2~3 个 I2C CLK

21.3.4 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）。

21.3.4.1 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
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：

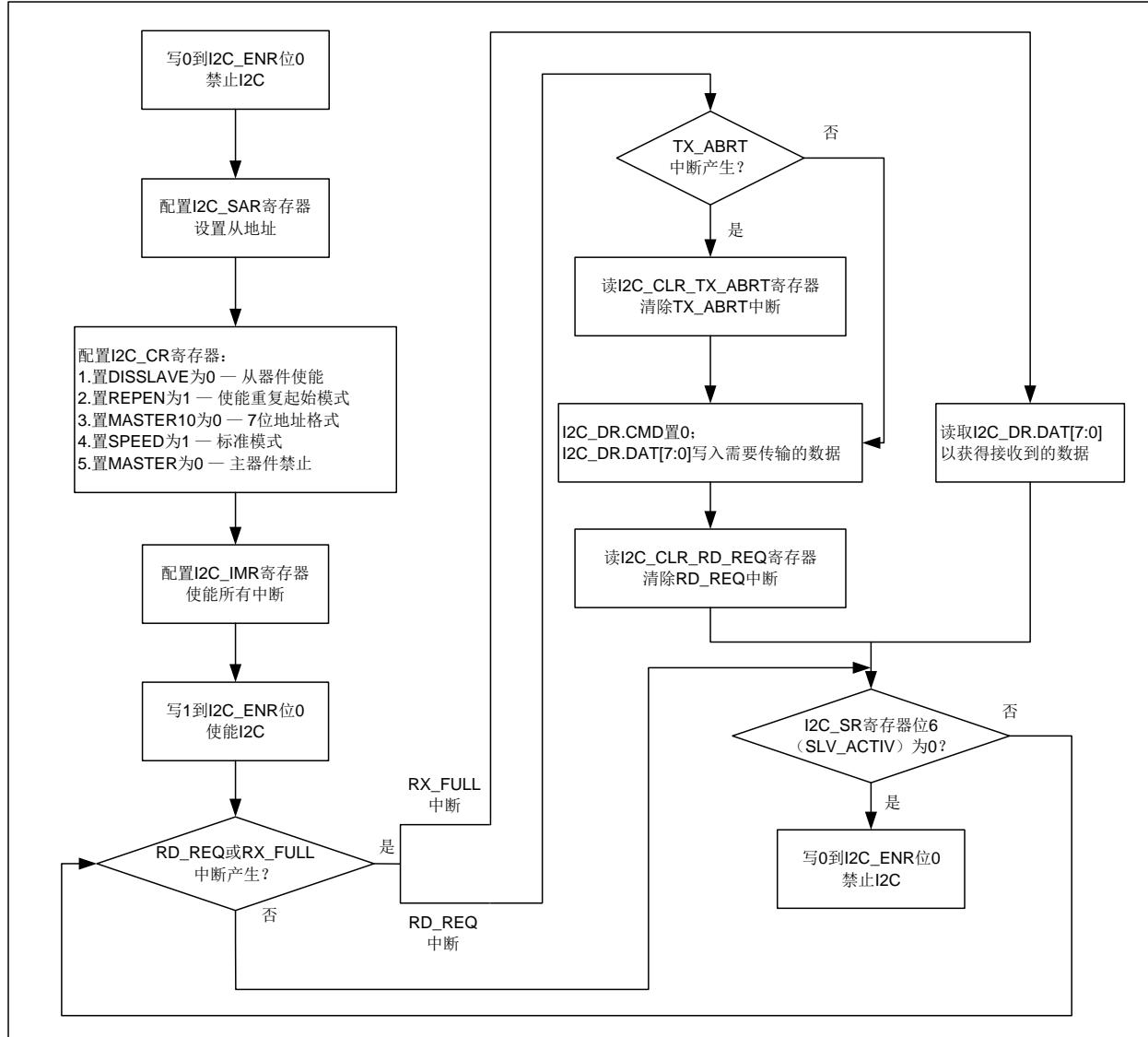


图 21-18 流程图 (I₂C 接口作为从器件)

- 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。

21.3.4.2 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：

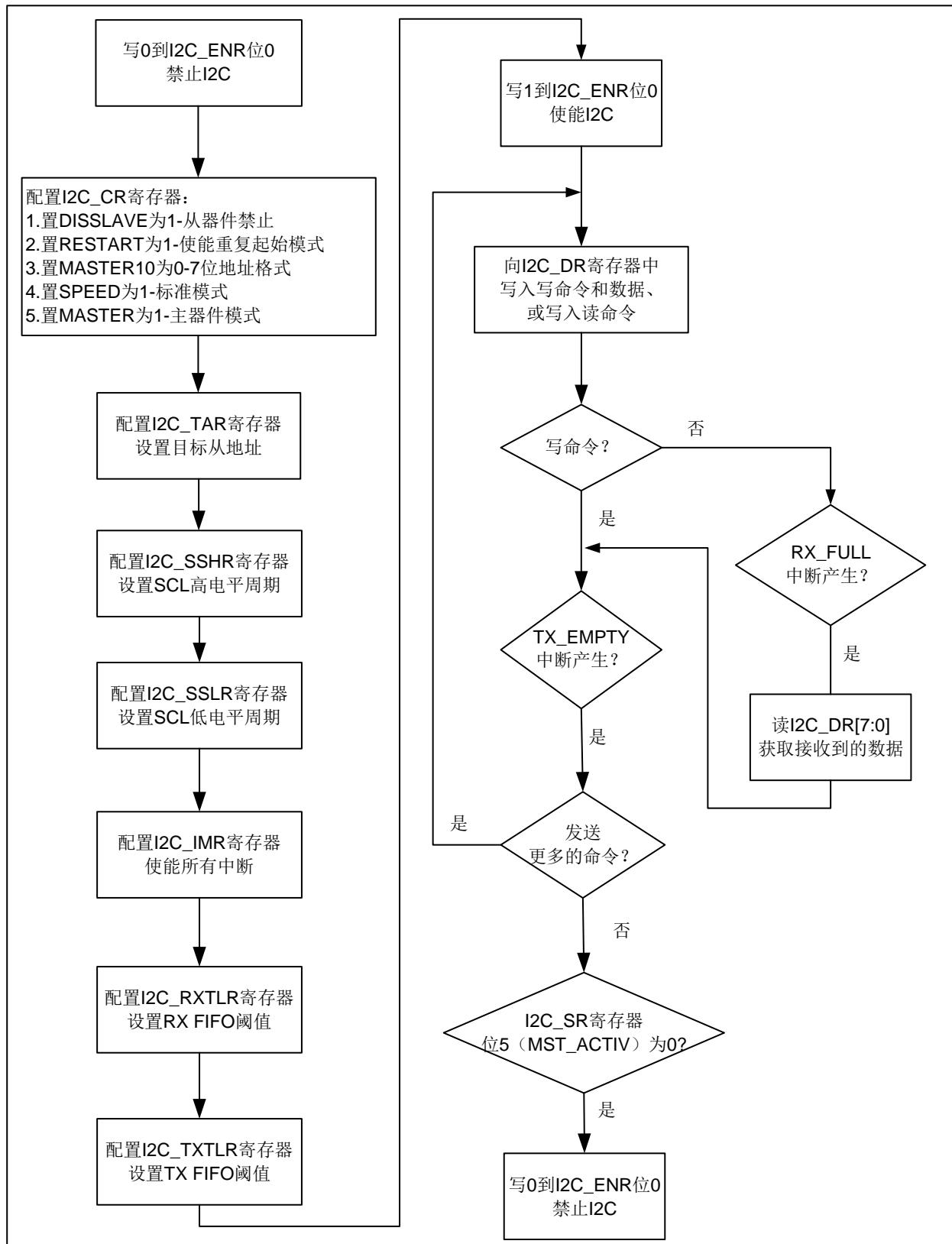


图 21-19 流程图 (I2C 接口作为主器件)

21.3.4.3 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

21.3.5 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 21-3 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	√	✗
START_DET	√	✗
STOP_DET	√	✗
ACTIVITY	√	✗
RX_DONE	√	✗
TX_ABRT	√	✗
RD_REQ	√	✗
TX_EMPTY	✗	√
TX_OVER	√	✗
RX_FULL	✗	√
RX_OVER	√	✗
RX_UNDER	√	✗

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。

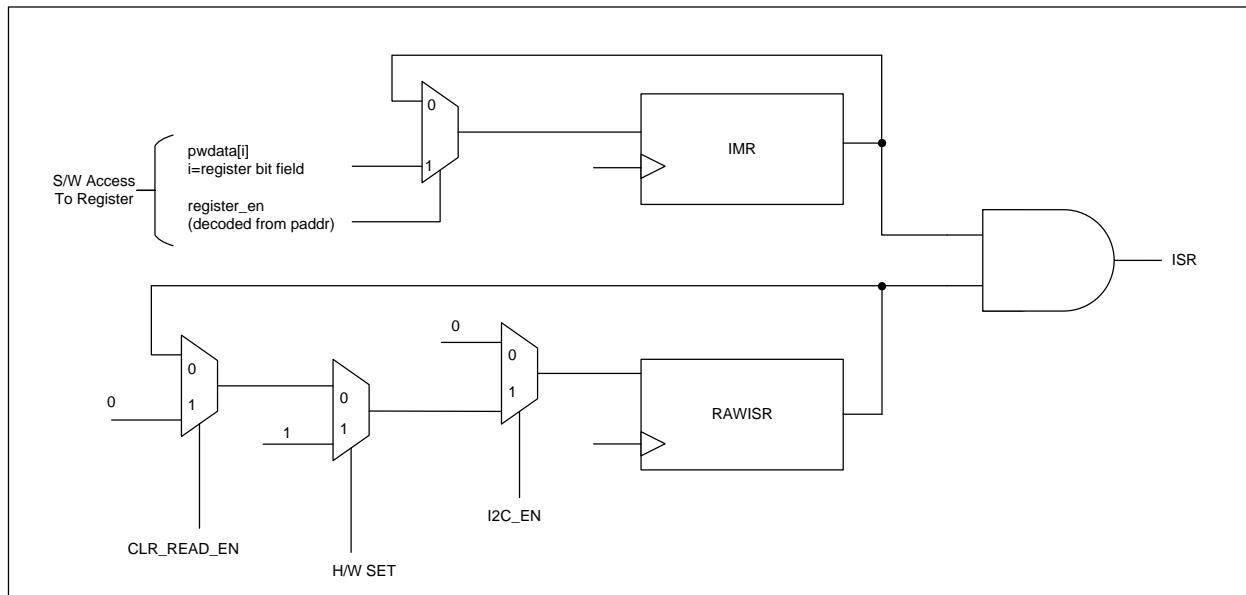


图 21-20 I2C 中断机制

21.3.6 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。

21.3.6.1 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。

21.3.6.2 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。

21.4 寄存器

21.4.1 寄存器总览

表 21-4 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	标准模式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	快速模式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

21.4.2 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.				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: 12	Reserved	保留, 必须保持复位值
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 注: 若需在接收字节后产生 STOP, 请使用 I2C_ENR.ABORT 位实现, 参考 I2C_ENR 寄存器。
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 21-5 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

21.4.3 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.				SPECIA L	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。

21.4.4 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。

21.4.5 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

21.4.6 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。

21.4.7 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。

21.4.8 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。

21.4.9 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。

21.4.10 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	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

21.4.11 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
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

21.4.12 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	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等于阈值时置位。 接收缓冲区数据小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。

21.4.13 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。

21.4.14 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。

21.4.15 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.															
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。

21.4.16 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])

21.4.17 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])

21.4.18 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])

21.4.19 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_REQ
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])

21.4.20 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_ABRT
															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。

21.4.21 I2C_RX_DONE 清除 RX_DONE 中断寄存器

偏移地址: 0x58

复位值: 0x0000 0000

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

21.4.22 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。

21.4.23 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])

21.4.24 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])

21.4.25 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.															
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])

21.4.26 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 E 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) 0: 中止没有发生或者已经结束 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

21.4.27 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: 接收缓冲未满 1: 接收缓冲满
3	RFNE	接收缓冲非空 0: 接收缓冲空 1: 接收缓冲非空
2	TFE	发送缓冲空 0: 发送缓冲非空 1: 发送缓冲空
1	TFNF	发送缓冲未满 0: 发送缓冲满 1: 发送缓冲未满
0	ACTIV	I2C 活动状态 MST_ACTIV 位与 SLV_ACTIV 位相或的结果。

21.4.28 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)

21.4.29 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)

21.4.30 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。

21.4.31 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															

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

21.4.32 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。

21.4.33 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)

21.4.34 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

21.4.35 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

22 CAN 控制器局域网

22.1 简介

CAN (Controller Area Network) 属于总线式串行通信网络。目前 CAN 的应用场景中，网络节点数量庞大，报文传输为多主方式工作，面对大量收到的报文，为了减少 CPU 的负荷，报文接收的实时响应程度需要降低。对于发送报文，支持软件配置优先级。

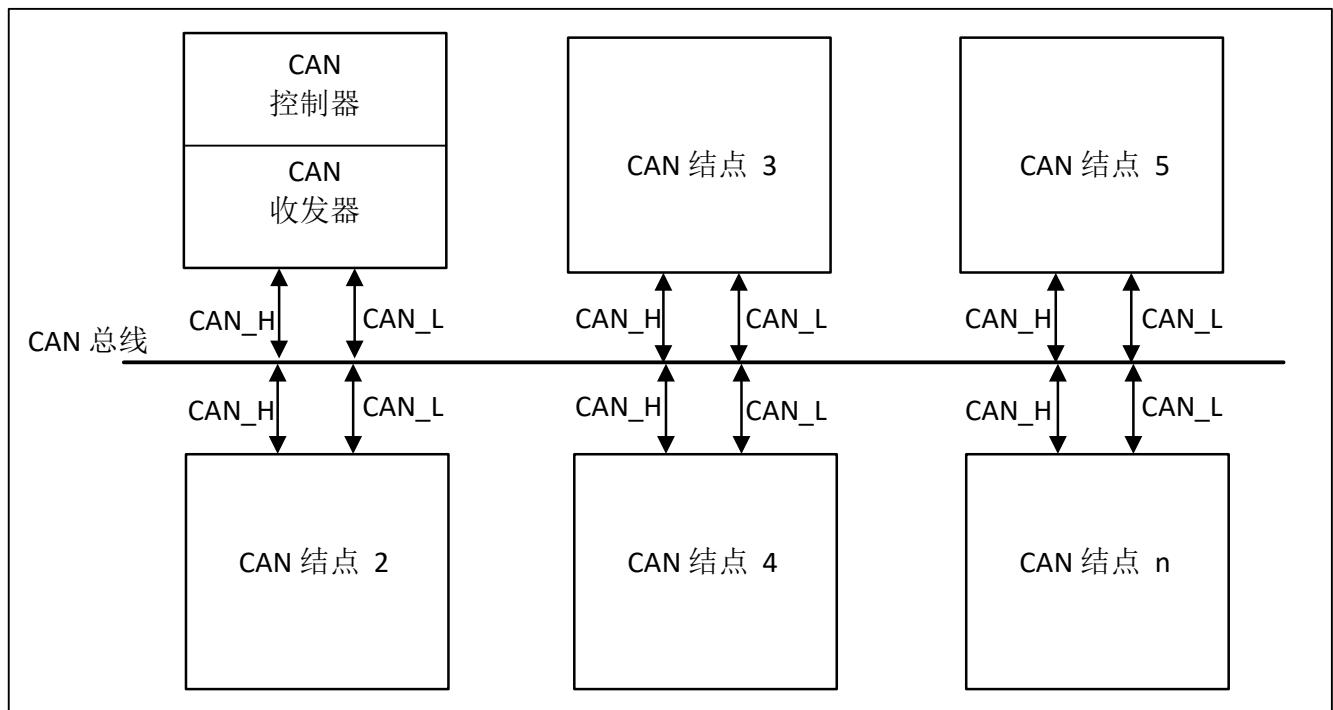


图 22-1 CAN 网络拓扑结构

22.2 主要特征

- 支持 CAN 协议的 2.0 A 和 2.0 B
- 64 Byte FIFO 接收缓冲器
- 同时支持 11 位和 29 位 ID
- 位速率可达 1Mbits/s
- PeliCAN 模式扩展功能
 - ◆ 可读/写访问的错误计数器
 - ◆ 可编程的错误报警限制
 - ◆ 最近一次错误代码寄存器

- ◆ CAN 总线错误中断
- ◆ 保存仲裁丢失的位置（第几位），并可产生仲裁丢失中断
- ◆ 单次发送（无重发）
- ◆ 只听模式（无响应、无主动错误标志）
- ◆ 软件位速率检测
- ◆ 验收滤波器扩展（4 字节代码，4 字节屏蔽）
- ◆ 自身信息接收（自接收请求）

22.3 功能描述

22.3.1 功能框图

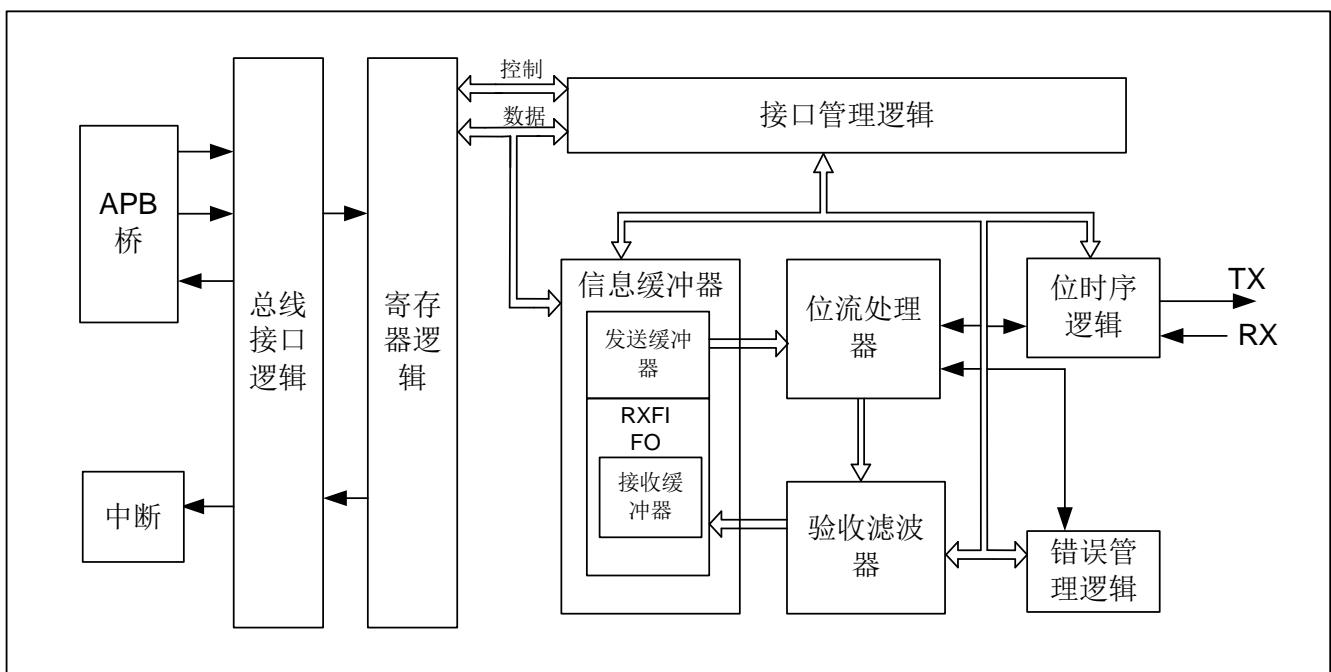


图 22-2 CAN 功能框图

接口管理逻辑(IML)

接口管理逻辑解释来自 CPU 的命令，控制 CAN 寄存器的寻址，向主控制器提供中断信息和状态信息。

发送缓冲器 (TXB)

发送缓冲器是 CPU 和 BSP(位流处理器)之间的接口，能够存储发送到 CAN 网络上的完整信息。缓冲器长 13 个字节，由 CPU 写入、BSP 读出。

接收缓冲器 (RXB, RXFIFO)

接收缓冲器是验收滤波器和 CPU 之间的接口，用来储存从 CAN 总线上接收的信息，接收缓冲器

(RXB, 13 个字节) 作为接收 FIFO (RXFIFO, 长 64 字节) 的一个窗口, 可被 CPU 访问。FIFO 可以实现在 CPU 处理信息的同时 CAN 控制器接收其它信息。

验收滤波器 (ACF)

验收滤波器把接收到的数据和接收识别码的内容进行比较, 以决定是否接收该信息。在纯粹的接收测试中, 所有的信息都保存在 RXFIFO 中。

位流处理器 (BSP)

位流处理器是一个在发送缓冲器、RXFIFO 和 CAN 总线之间控制数据流的程序装置。它还对 CAN 总线进行错误检测、仲裁、位填充和错误处理。

位时序逻辑 (BTL)

位时序逻辑监视 CAN 总线和处理与总线有关的位时序。它在信息开头‘隐性-显性’的总线传输时同步 CAN 总线位流 (硬同步), 接收信息时再次同步下一次传送 (软同步)。BTL 还提供了可编程的时间段来补偿传播延迟时间、相位转换, 以及定义采样点和一位时间内的采样次数。

错位管理逻辑 (EML)

EML 负责管理模块的传输错误。它接收 BSP 的出错报告, 通知 BSP 和 IML 进行错误统计。

22.3.2 信号描述

表 22-1 CAN 信号描述表

信号名	类型	描述
TX	输出	控制器发送信号
RX	输入	控制器接收信号

22.3.3 CAN 工作模式

CAN 控制器有 2 个主要的工作模式:

- BasicCAN 模式
- PeliCAN 模式

系统复位时默认模式是 BasicCAN 模式。

PeliCAN 模式是新的操作模式, 它能处理所有的 CAN 2.0B 规范的帧类型。而且它还提供一些增强功能使得应用于更宽的领域。

寄存器 CAN_CDR.MODE 定义了 CAN 模式。如果 CDR.MODE 是 '0', CAN 控制器工作于 BasicCAN 模式。否则, CAN 控制器工作于 PeliCAN 模式。

在 PeliCAN 模式下, CAN 控制器有一个含很多功能的重组寄存器。PeliCAN 模式支持 CAN 2.0B 协议规定的所有功能 (29 字节的识别码)。下面是 PeliCAN 模式的主要新功能:

- 标准帧和扩展帧的接收和传送
- 接收 FIFO (64 字节)
- 在标准和扩展格式中都有单/双验收滤波器 (含屏蔽和代码寄存器)

- 可读/写访问的错误计数器
- 可编程的错误报警限制
- 最近一次错误代码寄存器
- CAN 总线错误中断
- 保存仲裁丢失的位置（第几位），并可产生仲裁丢失中断
- 单次发送（当出现错误或仲裁丢失时不重发）
- 只听模式（CAN 总线监听，无应答，无错误标志）

22.3.4 BasicCAN 模式

22.3.4.1 复位模式

复位模式即初始化模式。在硬件启动或总线关闭时，复位请求位（CAN_CR.RR）被置为‘1’。如果这些位被软件访问，其值将发生变化，而且会影响内部时钟的下个上升沿。复位请求位变化时内部分频时钟同步，读复位请求位能够反映出这种同步状态。复位模式主要用于 CAN 通讯参数配置，在不同工作模式下内核对 CAN 寄存器的访问权限不同。

复位请求位被设为‘0’后 CAN 控制器将会等待：

- 一个总线空闲信号（11 个隐性位），如果前一次复位请求是硬件复位或 CPU 初始复位。
- 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造成的；必须说明的是，如果复位请求位被置位，一些寄存器的值会被改变。

22.3.4.2 工作模式

在复位模式完成后，软件应该让硬件进入正常模式，以便正常接收和发送报文。复位模式下，当向复位位传送了“1—0”的下降沿时，CAN 控制器便返回工作模式，进行报文的发送和接收。

22.3.4.3 睡眠模式

睡眠模式位设为 1（sleep），CAN 控制器将进入睡眠模式。应在没有总线活动且没有挂起的中断时设定睡眠模式，则会产生唤醒中断。睡眠模式位设为低（唤醒）之后总线进入活动状态或中断被激活。唤醒后，时钟启动且产生一个唤醒中断。由于总线唤醒后，直到检测到 11 个连续的隐性位（总线空闲序列）后才能接收信息。注意在复位模式中是不能设置睡眠模式位的。清除复位模式后，再一次检测到总线空闲时，睡眠模式位的设置才开始有效。

表 22-2 BasicCAN 模式寄存器权限分配表

Offset	段	工作模式		复位模式	
		读	写	读	写
00	控制	控制	控制	控制	控制
04		(FFH)	命令	(FFH)	命令

Offset	段	工作模式		复位模式	
08		状态		状态	
0C		(FFH)		中断	
10		(FFH)		验收代码	验收代码
14		(FFH)		验收屏蔽	验收屏蔽
18		(FFH)		总线定时 0	总线定时 0
1C		(FFH)		总线定时 1	总线定时 1
20		(FFH)			
24		测试	测试	测试	测试
28	发送缓冲器	识别码 (10 ~ 3)	识别码 (10~3)	(0xFF)	
2C		识别码 (2~0)			
RTR 和 DLC	识别码 (2~0)				
RTR 和 DLC	(0xFF)				
30		DATA1	DATA1	(0xFF)	
34		DATA2	DATA2	(0xFF)	
38		DATA3	DATA3	(0xFF)	
3C		DATA4	DATA4	(0xFF)	
40		DATA5	DATA5	(0xFF)	
44		DATA6	DATA6	(0xFF)	
48		DATA7	DATA7	(0xFF)	
4C		DATA8	DATA8	(0xFF)	
50	接收缓冲器	识别码 (10 ~ 3)	识别码 (10 ~ 3)	识别码 (10 ~ 3)	识别码 (10~3)
54		识别码 (2 ~ 0)	识别码 (2 ~ 0)	识别码 (2 ~ 0)	识别码 (2 ~ 0)
		RTR 和 DLC	RTR 和 DLC	RTR 和 DLC	RTR 和 DLC
58		DATA1	DATA1	DATA1	DATA1

注：‘(FFH)’代表读出数据全为‘1’，“空行”代表无写操作权限，其余表示可操作。偏移地址为 0x7C 的“时钟分频器”用于选择 BasicCAN 和 PeliCAN。

22.3.5 PeliCAN 模式

22.3.5.1 复位模式

复位模式即初始化模式。在硬件复位或总线关闭时复位模式位被置为‘1’。如果通过软件访问这一位，值将发生变化且在下一个内部时钟的上升沿有效，读复位请求位能够反映出这种同步状态。复位模式位为‘0’后，CAN 控制器会等待：

- 一个总线空闲信号 (11 个隐性位)，如果上一次复位请求是硬件复位或 CPU 初始复位。
- 128 个总线空闲，如果前一次复位请求是 CAN 控制器在重新进入总线开启模式前初始化总线造

成的。

22.3.5.2 工作模式

在复位模式完成后，软件应该让硬件进入正常模式，以便正常接收和发送报文。复位模式下，当检测到 CAN_MOD 寄存器的 RM 位出现了‘1—0’的下降沿，CAN 控制器便返回工作模式，进行报文的发送和接收。

22.3.5.3 睡眠模式

睡眠模式位设为 ‘1’ (sleep)，CAN 控制器将进入睡眠模式。应在没有总线活动且没有挂起的中断时设定睡眠模式，否则会产生唤醒中断。睡眠模式位设为低（唤醒）之后总线进入活动状态或中断被激活。唤醒后，时钟启动且产生一个唤醒中断。由于总线唤醒后，直到检测到 11 个连续的隐性位（总线空闲序列）后才能接收信息。注意在复位模式中是不能设置睡眠模式位的。清除复位模式后，再一次检测到总线空闲时，睡眠模式位的设置才开始有效。

22.3.5.4 自检测模式

此模式主要用于测试。设置自检测模式位 (CAN_MOD.STM) 为 ‘1’ 进入自检测模式。此模式在没有其他活动的节点的情况下，使用自接收功能可以完成节点的全部功能测试。该模式下即使没有应答，CAN 控制器也会成功发送。

22.3.5.5 只听模式

此模式主要用于测试。设置只听模式位 (CAN_MOD.LOM) 为 ‘1’ 进入只听模式。这种工作模式使 CAN 控制器进入错误被动状态，不能进行信息传输。只听模式下可以使用位速检测功能，另外所有其它功能都能像正常工作模式中一样使用。

在该模式中，CAN 控制器不能在 CAN 总线发送显性位，包含主动错误标志或超载标志，成功接收后的应答信号也不会给出。

注：在进入只听模式之前必须进入复位模式。

表 22-3 PeliCAN 模式寄存器权限分配表

Offset	工作模式		复位模式	
	读	写	读	写
00	模式	模式	模式	模式
04	(00H)	命令	(00H)	命令
08	状态		状态	
0C	中断		中断	
10	中断使能		中断使能	中断使能
14	(00H)		(00H)	
18	总线时序 0		总线时序 0	总线时序 0
1C	总线时序 1		总线时序 1	总线时序 1

Offset	工作模式			复位模式	
20	保留				
24	测试		测试		测试
28	保留				保留
2C	仲裁丢失捕捉				仲裁丢失捕获
30	错误代码捕捉				错误代码捕捉
34	错误报警限制				错误报警限制
38	RX 错误计数器				RX 错误计数器
3C	TX 错误计数器				TX 错误计数器
40	RX 帧信息 SFF	RX 帧错误 EFF	TX 帧错误 SFF	TX 帧错误 EFF	验收代码 0
44	RX 识别码 1	RX 识别码 1	TX 识别码 1	TX 识别码 1	验收代码 1
48	RX 识别码 2	RX 识别码 2	TX 识别码 2	TX 识别码 2	验收代码 2
4C	RX 数据 1	RX 识别码 3	TX 数据 1	TX 识别码 3	验收代码 3
50	RX 数据 2	RX 识别码 4	TX 数据 2	TX 识别码 4	验收屏蔽 0
54	RX 数据 3	RX 数据 1	TX 数据 3	TX 数据 1	验收屏蔽 1
58	RX 数据 4	RX 数据 2	TX 数据 4	TX 数据 2	验收屏蔽 2
5C	RX 数据 5	RX 数据 3	TX 数据 5	TX 数据 3	验收屏蔽 3
60	RX 数据 6	RX 数据 4	TX 数据 6	TX 数据 4	保留
64	RX 数据 7	RX 数据 5	TX 数据 7	TX 数据 5	保留
68	RX 数据 8	RX 数据 6	TX 数据 8	TX 数据 6	保留
6C	(FIFO RAM)	RX 数据 7		TX 数据 7	保留
70	(FIFO RAM)	RX 数据 8		TX 数据 8	保留
74	RX 信息计数器		RX 信息计数器		
78	RX 缓冲器起始地址		RX 缓冲器起始地址	RX 缓冲器起始地址	RX 缓冲器起始地址
7C	时钟分频器	时钟分频器	时钟分频器	时钟分频器	时钟分频器
80	内部 RAM 地址 0 (FIFO)		内部 RAM 地址 0	内部 RAM 地址 0	内部 RAM 地址 0 (FIFO)
84	内部 RAM 地址 1 (FIFO)		内部 RAM 地址 1	内部 RAM 地址 1	内部 RAM 地址 1 (FIFO)
...
17C	内部 RAM 地址 63 (FIFO)		内部 RAM 地址 63	内部 RAM 地址 63	内部 RAM 地址 63 (FIFO)

Offset	工作模式				复位模式	
180	内部 RAM 地址 64 (TX 缓冲器)		内部 RAM 地址 64	内部 RAM 地址 64	180	内部 RAM 地址 64 (TX 缓冲器)
...
1B0	内部 RAM 地址 76 (TX 缓冲器)		内部 RAM 地址 76	内部 RAM 地址 76	1B0	内部 RAM 地址 76 (TX 缓冲器)
1B4	内部 RAM 地址 77 (空闲)		内部 RAM 地址 77	内部 RAM 地址 77	1B4	内部 RAM 地址 77 (空闲)
1B8	内部 RAM 地址 78 (空闲)		内部 RAM 地址 78	内部 RAM 地址 78	1B8	内部 RAM 地址 78 (空闲)
1BC	内部 RAM 地址 79 (空闲)		内部 RAM 地址 79	内部 RAM 地址 79	1BC	内部 RAM 地址 79 (空闲)
1C0	(00H)		(00H)		1C0	(00H)
...
1FC	(00H)		(00H)		1FC	(00H)

22.3.6 发送处理

根据 CAN 协议规范，报文的传输由 CAN 控制器独立完成。微控制器需要先配置标识符，需要发送的数据长度以及数据，然后对命令寄存器的发送请求位置‘1’，来请求发送。当 CAN 控制器正在发送报文时，发送缓冲器被写锁定。所以在写一个新报文到发送缓冲器之前，微控制器必须检查状态寄存器的发送缓冲器状态标志（TBS）。

同时设置命令位 CMR.TR 和 CMR.AT 只会产生一次信息发送，当发送错误或仲裁丢失时是不会重发的（单次发送）。只设置命令位 CMR.TR 数据发送丢失会重传。在自检测模式下，设置命令位 CMR.SRR 和 CMR.TR 会立即产生一次自接收性质的信息发送。

22.3.6.1 中止

一个已经请求发送的报文，可以通过置位命令寄存器位的相应位（CAN_CMR 寄存器中的 AT 位）执行‘中止发送’。

当 CPU 需要当前发送请求等待时，例如：先发送一条比较紧急的信息时，可以使用‘中止发送’。但当前正在处理的传送是不会停止的。可以通过检查传送完成状态位来查看原信息是否发送成功。不过，这应在发送缓冲器状态位置‘1’或产生发送中断后执行。

要注意的是，即使因为发送缓冲器状态位变为‘释放’而使信息被中止，也会产生发送中断。

如果前一条指令中发送请求被置为‘1’，则不能通过清除发送请求为‘0’来取消，而应通过中止发送位为‘1’来取消。

22.3.7 接收管理

接收到的报文由 CAN 控制器独立完成。收到的报文放在接收缓冲器。可以发送给微控制器的报文，由状态寄存器的接收缓冲器状态标志 RBS 和接收中断标志 RI 标出。

22.3.7.1 查询控制接收

微控制器读 CAN 控制器的状态寄存器，检查接收缓冲状态（RBS）查看是否收到一个报文。

当读到 RBS 位为‘1’，表示收到一个或多个报文，微控制器需要从 CAN 中取得报文，然后置位命令寄存器的响应标志位 RRB 发送一个释放接收缓冲器命令。

22.3.7.2 中断控制接收

中断使能位位于 CAN 控制寄存器里（对于 BasicCAN 模式）或位于中断使能寄存器里（对于 PeliCAN 模式）。如果 CAN 控制器已接收一个报文，而且报文已经通过验收滤波器并放在接收 FIFO，那么会产生一个接收中断。进入中断服务程序，微控制器取走报文，然后置位命令寄存器的响应标志位 RRB 发送一个释放接收缓冲器命令。

22.3.7.3 溢出

在接收 FIFO 已满又接收到新的报文的时会导致溢出，同时置位状态寄存器中的数据溢出状态位（如果使能）通知微控制器有数据溢出的情况，CMR.CDO 位置‘1’可清除溢出状态。

CAN_CMR 寄存器的 ERB 位可用于清除 FIFO，当使能 ERB 后，若接收 FIFO 发生溢出，硬件会自动清除接收 FIFO。

22.3.7.4 有效报文

根据 CAN 协议，当报文被正确接收（直到 EOF 域的最后一位都没有错误），且通过了标识符过滤，那么该报文被认为是有效报文。

CAN_CMR 寄存器中的 RRB 位是用于释放接收缓冲器。

22.3.8 标识符过滤

独立的 CAN 控制器装配了一个多功能的验收滤波器，该滤波器允许自动检查标识符和数据字节。滤波器由验收代码寄存器和屏蔽寄存器组成，并根据特定的算法来控制。接收到的数据会和验收代码寄存器中的值进行逐位比较。验收屏蔽寄存器定义对应的位是否参与比较（0 = 参与，1 = 不参与）。

在 CAN 总线上，接收器检测所有的广播报文，只有收到报文的标识符与验收代码寄存器相关的位相同时，报文才会被接收。使用这些有效的过滤方法，可以降低微控制器的处理负载。

22.3.8.1 BasicCAN 模式里的验收滤波器

该滤波器是由两个寄存器，验收代码寄存器（ACR）和验收屏蔽寄存器（AMR）控制。CAN 报文的 11 位标志符的高 8 位和这些寄存器里值相比较。可以定义若干个的标识符被某些节点接收。

例子：验收码寄存器（ACR）包括：

验收码寄存器（ACR）包括：	MSB	LSB																																
验收屏蔽寄存器（AMR）包括：	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	1	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	1	X	X	X	0	1	0	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr> </table>	X	X	X	X	X	X	X	X
0	1	1	1	0	0	1	0																											
0	0	1	1	1	0	0	0																											
0	1	X	X	X	0	1	0																											
X	X	X	X	X	X	X	X																											
带有 11 位标识符信息被接收 (X=无关)	ID.10																																	

图 22-3 CAN 标识符接收示例

在验收屏蔽寄存器里是‘1’的位置上，标识符相应的位可以是任意值。这对于三个最低位也一样。因此在这个例子里可以接收 64 个不同的标识符。在验收屏蔽寄存器里是‘0’的位置上，标识符的相应位必须等于验收代码寄存器相应位的值。

22.3.8.2 PeliCAN 模式里的验收滤波器

在验收滤波器的帮助下，只有当接收信息中的识别位和验收滤波器预定义的值相等时，CAN 控制器才允许将已接收信息存入 RXFIFO。PeliCAN 模式中，验收滤波器由验收代码寄存器（ACRn）和验收屏蔽寄存器（AMRn）定义。要接收的信息的位模式在验收代码寄存器中定义。相应的验收屏蔽寄存器允许定义某些位为‘不影响’（即可为任意值）。有两种不同的过滤模式可在模式寄存器中的 AFM 位选择：

- 1：单滤波器模式
- 0：双滤波器模式

22.3.8.3 单滤波器配置

这种滤波器配置可以定义一个长滤波器（4 字节）。滤波器字节和信息字节之间位的对应关系取决于当前接收帧格式。

标准帧：如果接收的是标准帧格式的信息，在验收滤波中只使用前两个数据字节来存放包括 RTR 位的完整的识别码，后两个字节存放数据字节信息。如果由于置位 RTR 位而导致没有数据字节，或因为设置相应的数据长度代码而没有或只有一个数据字节，信息也会被接收。为了成功接收信息，每个位都必须比较通过。

注：ACR1 和 AMR1 的低四位是不使用的。为了和未来产品兼容这些位可通过设置 AMR1 的位 3 到位 0 为‘1’，而设定为不影响。

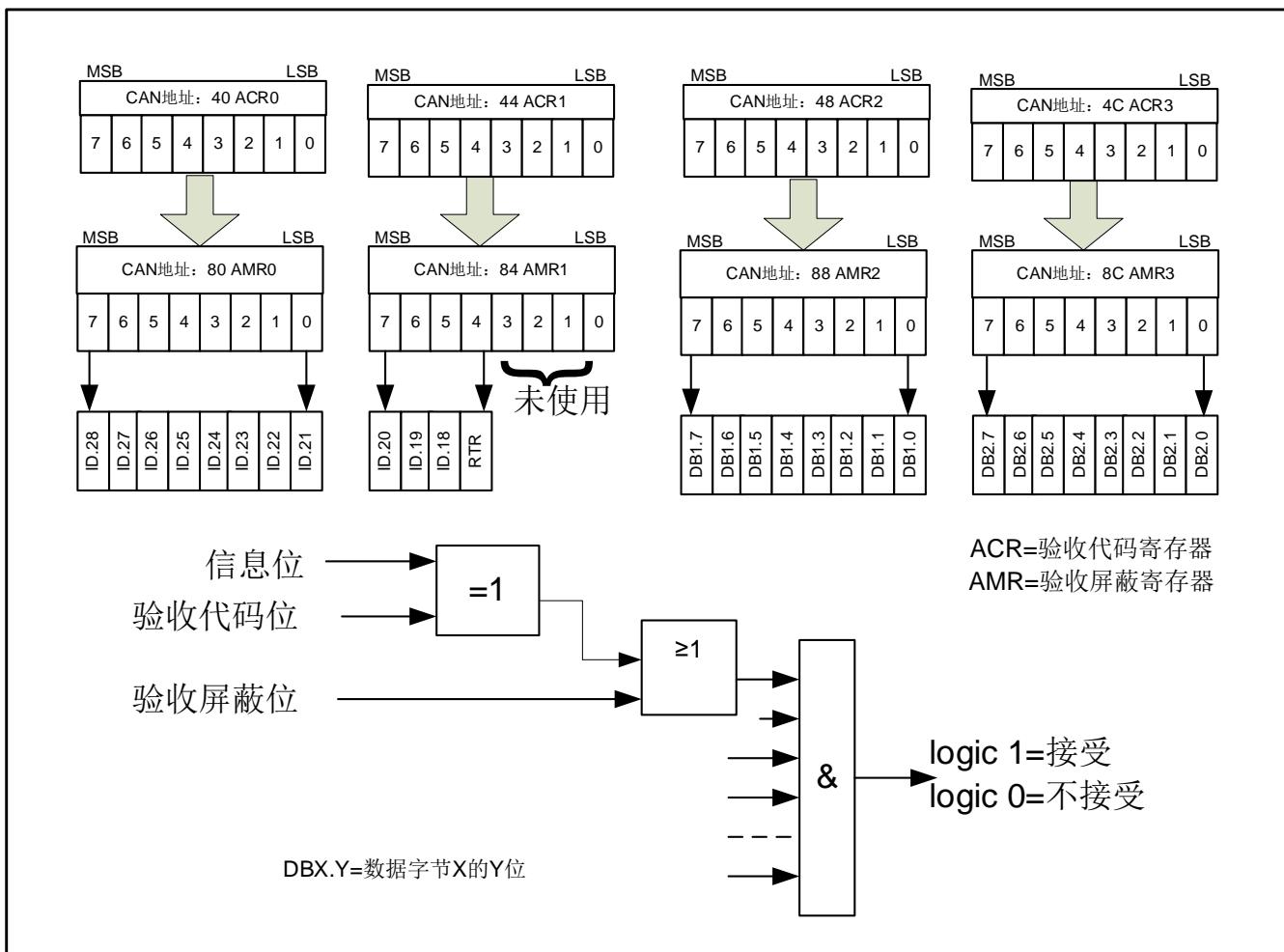


图 22-4 接收标准结构信息时的单个滤波器配置

扩展帧: 如果接收的信息是扩展帧格式, 包括 RTR 位的全部识别码将被接收过滤使用。为了成功接收信息, 每个位都必须比较通过。

注: AMR3 的最低两位和 ACR3 是不用的。这些位应该通过置位 AMR3 的位 1 和位 0 来定为不影响。

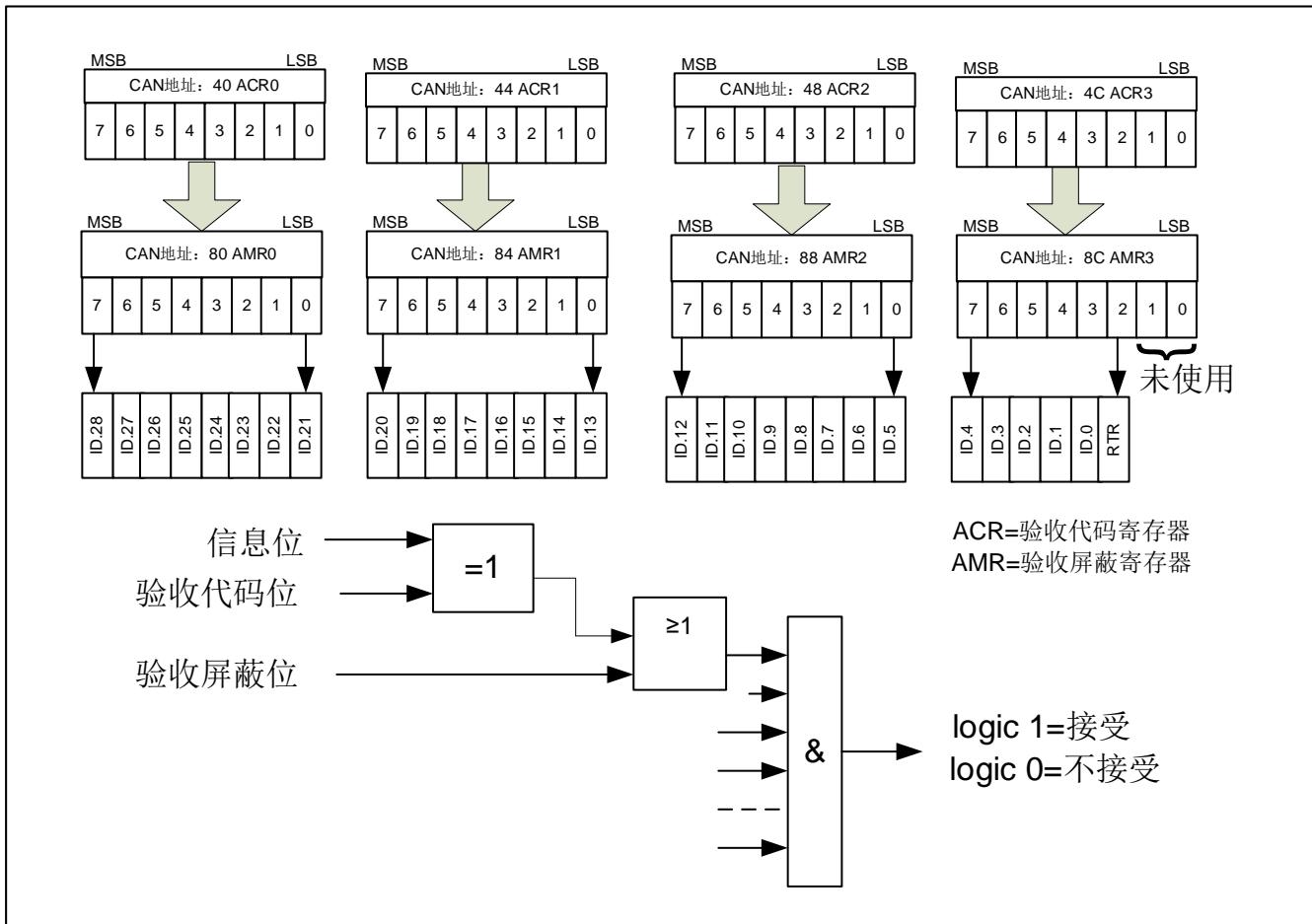


图 22-5 单滤波器配置，接收扩展帧信息

22.3.8.4 双滤波器的配置

这种配置可以定义两个短滤波器。一条接收的信息要和两个滤波器比较来决定是否放入接收缓冲器中。至少有一个滤波器发出接收信号，接收的信息才有效。滤波器字节和信息字节之间位的对应关系取决于当前接收的帧格式。

标准帧：如果接收的是标准帧信息，被定义的两个滤波器是不一样的。第一个滤波器比较包括 RTR 位的整个标准识别码和信息的第一个数据字节。第二个滤波器只比较包括 RTR 位的整个标准识别码。

为了成功接收信息，所有单个位的比较时应至少有一个滤波器。RTR 位置位或数据长度代码是‘0’时表示没有数据字节存在。无论怎样，只要从开始到 RTR 位的部分都被表示接收，信息就可以通过滤波器 1。

如果没有向滤波器请求数据字节过滤，AMR1 和 AMR3 的低四位必须被置为‘1’（不影响）。当使用包括 RTR 位的整个标准识别码时，两个滤波器都同样工作。

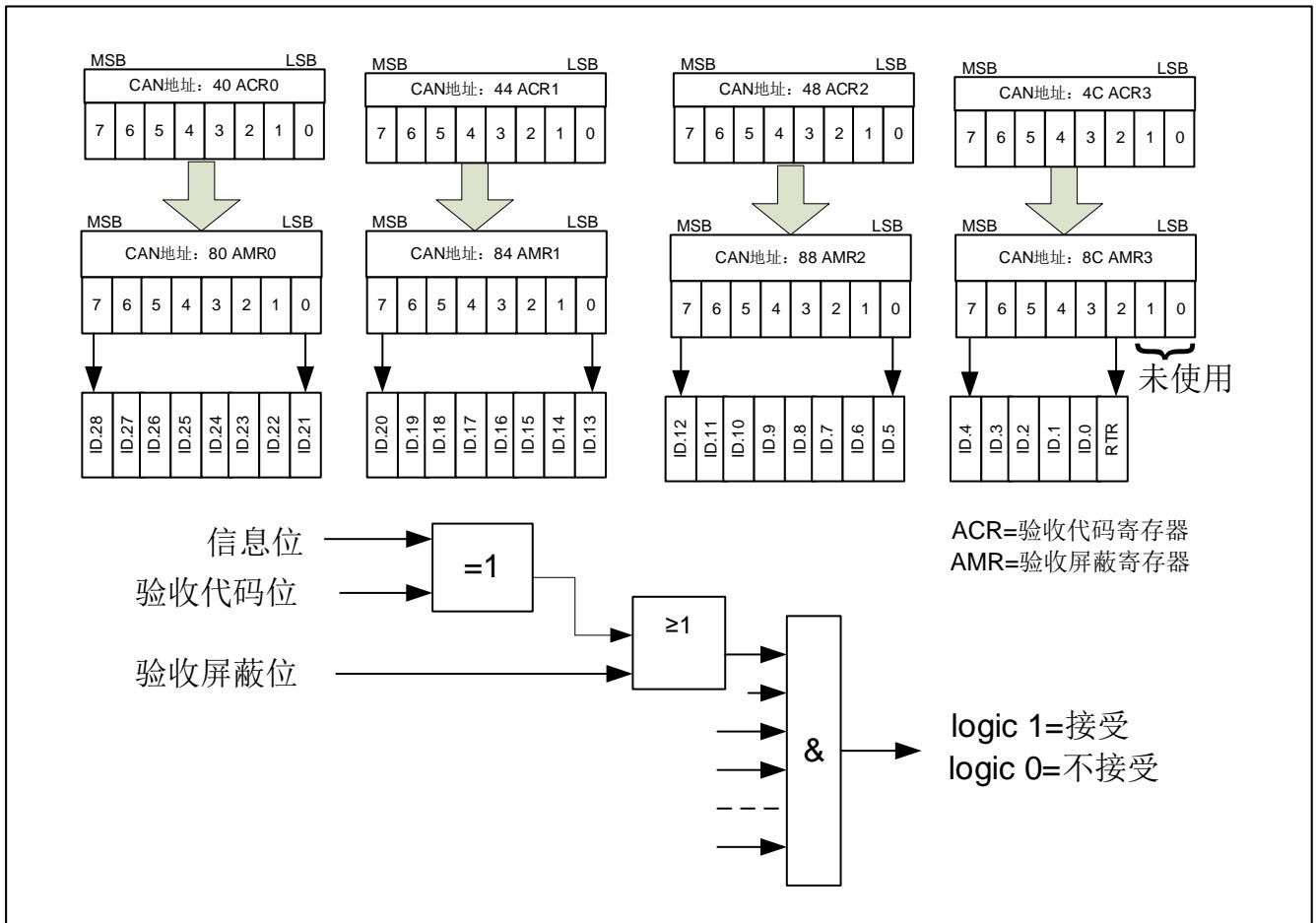


图 22-6 接收标准结构信息时的双个滤波器配置

扩展帧：如果接收到扩展帧信息，定义的两个滤波器是相同的。两个滤波器都只比较扩展帧识别码的前两个字节。

为了能成功接收信息，所有单个位的比较时至少有一个滤波器表示接收。

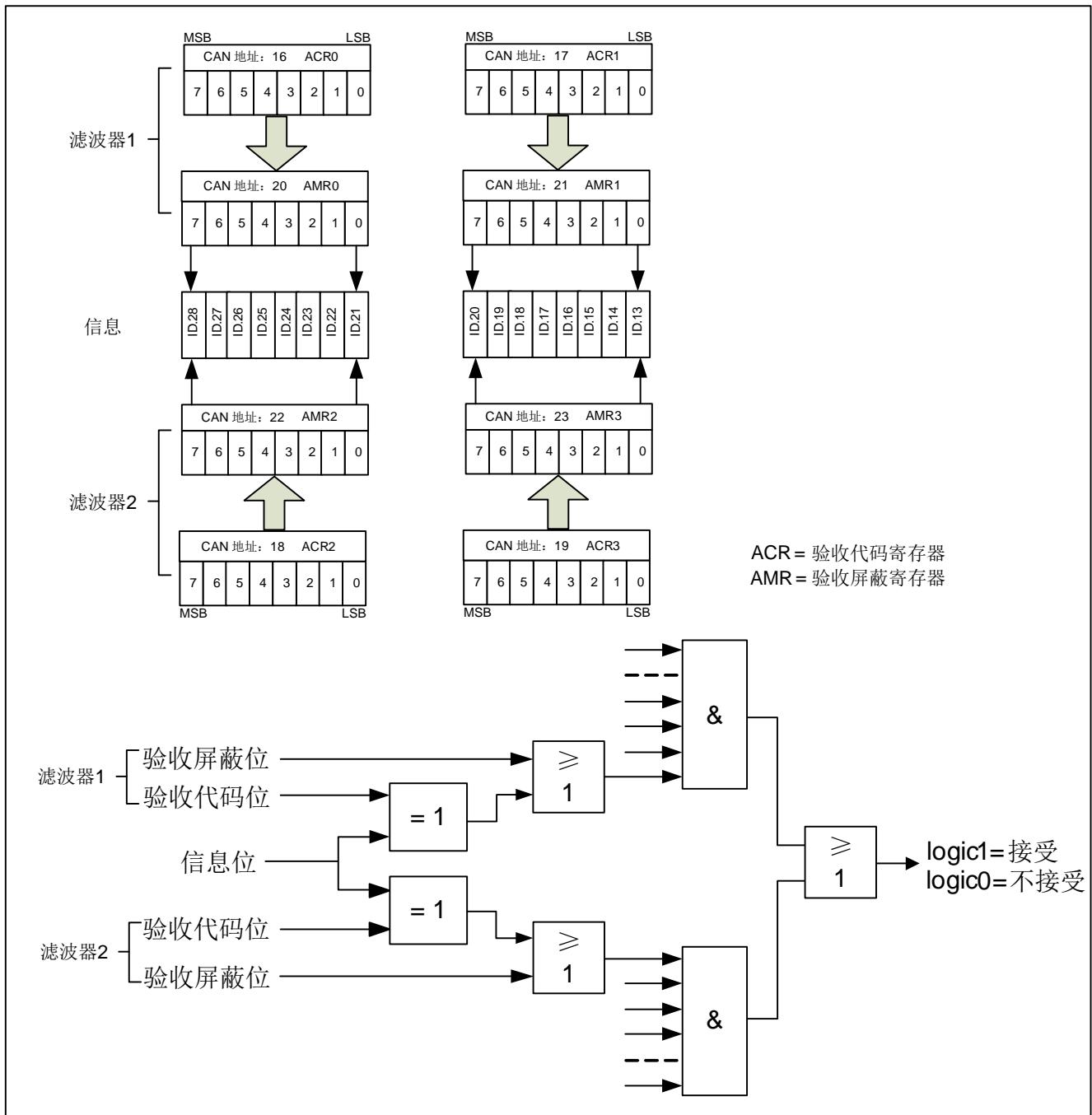


图 22-7 双滤波器配置，接收扩展帧信息

例 1：假设下面 1 个标准帧报文要在 PeliCAN 模式里滤波，可以通过使用一个长滤波器完成（单滤波器模式）。

验收代码寄存器 (ACRn) 和验收屏蔽寄存器 (AMRn) 包括：

n	0	1 (高四位)	2	3
ACRn	01XX X010	XXXX	XXXX XXXX	XXXX XXXX
AMRn	0011 1000	1111	1111 1111	1111 1111
接收的报文 (ID28 ~ ID.18, RTR)	01xx x010 xxxx			

“X”= 不相关“x”= 任意值。只使用了 ACR1 和 AMR1 的高四位。

例 2: 假设下面 2 个有标准帧标识符的报文在标识符不用进一步译码就被接收。数据和远程帧必须被正确接收。数据字节不要求验收滤波。

报文 1: (ID.28) 1011 1100 101 (ID.18)

报文 2: (ID.28) 1111 0100 101 (ID.18)

使用单滤波模式可以接收到四个报文而不仅是要求的两个:

n	0	1 (高四位)	2	3
ACRn	1X11 X100	101X	XXXX XXXX	XXXX XXXX
AMRn	0100 1000	0001	1111 1111	1111 1111
接收的报文 (ID28 ~ ID.18, RTR)	1011 0100 101x 1111 0100 101x (报文 2) 1011 1100 101x (报文 1) 1111 1100 101x			

("X'= 不相关"X'= 任意值'只使用了 ACR1 和 AMR1 的高四位。)

这个结果需要进一步解码才能满足接收两条信息的要求。 使用双滤波器可以得到正确的结果。

n	滤波器 1			滤波器 2	
	0	1	3 低四位	2	3 高四位
ACRn	1011 1100	101X XXXX	... XXXX	1111 0100	101X ...
AMRn	0000 0000	0001 1111	... 1111	0000 0000	0001 ...
接收的信息 (ID28 ~ ID.18, RTR)	1011 1100 101X (报文 1)			1111 0100 101X (报文 2)	

("X'= 不相关, "X'= 任意值)

报文 1 被滤波器 1 接收, 报文 2 被滤波器 2 接收。如果报文至少被两个滤波器中的一个接收, 报文就被存放到接收 FIFO。这种方法可满足于这种要求。

例 3: 在这个例子里, 使用一个长的验收滤波器过滤一组带有扩展标识符的报文。

n	0	1	2	3 (高六位)
ACRn	1011 0100	1011 000X	1100 XXXX	0011 0XXX
AMRn	0000 0000	0001 0001	0000 1111	0000 0111
接收的报文 (ID28 ~ ID.18, RTR)	1011 0100 101x 000x 1100 xxxx 0011 0x			

("X'= 不相关"X'= 任意值'只使用了 ACR1 和 AMR1 的高六位。)

例 4: 有些使用标准帧系统仅用 11 位标识符和头两个数据字节识别报文。如果报文超过 8 个数据字节, 头两个数据字节定义为报文头和使用分段存储协议就会使用像这样的协议。

例如 DeviceNet。对于这种系统类型, CAN 控制器除了 11 位标识符和 RTR 位外, 在单滤波器模式里能过滤两个数据字节, 在双滤波器模式里能过滤一个数据字节 (除了 11 位标识符和 RTR 位)。

下面的例子显示了用双滤波器模式, 在这种系统里有效地过滤报文:

n	滤波器 1			滤波器 2	
	0	1	3 低四位	2	3 高四位
ACRn	1110 1011	0010 1111	... 1001	1111 0100	XXX0 ...
AMRn	0000 0000	0000 0000	... 0000	0000 0000	1110 ...

	滤波器 1	滤波器 2	
接收的信息 (ID28 ~ ID.18, RTR)	1110 1011 0010 + 1111 ... 1001 标识符 RTR + 头一个数据字节	1111 0100 标 识符	xxx0 RTR

(“X”= 不相关, “x”= 任意值)

- 滤波器 1 过滤的报文有:
 - ◆ 标识符'11101011001'
 - ◆ RTR ='0', 也就是说是数据帧
 - ◆ 数据字节'11111001'
 - ◆ (这是指例如 DeviceNet: 一个信息的所有段都被过滤)
- 滤波器 2 用来过滤一组 8 个报文, 其中报文有
 - ◆ 标识符'11110100 000'到 11110100111'
 - ◆ RTR ='0', 也即是数据帧

22.3.8.5 滤波器组模式设置

过滤器组共 20 组, 可以通过配置相应的 CAN_FGAx ($x = 0 \sim 2$) 来打开和禁用某些过滤器组, 应用程序中不用的过滤器组, 应该保持在禁用状态;

通过配置 FGAx ($x = 0 \sim 19$) 选择滤波器组单/双滤波模式;

验收代码寄存器和验收屏蔽寄存器 AMR0~19 和 ACR0~19 共 20 组;

每组滤波器功能的使用方式相同, 可单组使用也可多组使用。

22.3.9 报文存储

要在 CAN 总线上发送的数据被载入 CAN 控制器的存储区, 这个存储区叫发送缓冲器。从 CAN 总线上收到的数据也存储在 CAN 控制器的存储区, 这个存储区叫接收缓冲器。这些缓冲器包括 2, 3 或 5 个字节的标识符和帧信息 (取决于模式和帧类型), 而最多可以包含 8 个数据字节。

22.3.9.1 BasicCAN 模式

缓冲器长达 10 个字节

- 2 个标识符字节
- 8 个数据字节

表 22-4 BasicCAN 模式里的 RX 和 TX 缓冲器

相对 CAN 偏移量		寄存器		组成与注释
TX (16 进制)	RX (16 进制)	TX	RX	
28	50	CAN_TXIDR1	CAN_RXIDR1	8 位标识符
2C	54	CAN_TXIDR2	CAN_RXIDR2	3 位标识符, 1 位远程传输请求位, 4 位数据长度代码, 表示数据字节的数量

相对 CAN 偏移量		寄存器		组成与注释
30	58	CAN_TXDR1	CAN_RXDR1	由数据长度代码表示, 最多 8 个数据字节
34	5C	CAN_TXDR2	CAN_RXDR2	由数据长度代码表示, 最多 8 个数据字节
38	60	CAN_TXDR3	CAN_RXDR3	由数据长度代码表示, 最多 8 个数据字节
3C	64	CAN_TXDR4	CAN_RXDR4	由数据长度代码表示, 最多 8 个数据字节
40	68	CAN_TXDR5	CAN_RXDR5	由数据长度代码表示, 最多 8 个数据字节
44	6C	CAN_TXDR6	CAN_RXDR6	由数据长度代码表示, 最多 8 个数据字节
48	70	CAN_TXDR7	CAN_RXDR7	由数据长度代码表示, 最多 8 个数据字节
4C	74	CAN_TXDR8	CAN_RXDR8	由数据长度代码表示, 最多 8 个数据字节

22.3.9.2 PeliCAN 模式

缓冲器长 13 个字节。

- 1 字节帧信息
- 2 个或 4 个标识符字节（标准帧或扩展帧）
- 最多 8 个数据字节

22.3.9.3 发送缓冲器

发送缓冲器的全部列表见下图。请务必分清标准帧格式（SFF）和扩展帧格式（EFF）配置。发送缓冲器允许定义长达 8 个数据字节发送信息。



图 22-8 标准帧和扩展帧格式配置在发送缓冲器的列表

帧信息中的 FF 位决定 CAN 控制器将要发送扩展帧格式还是标准帧格式。

22.3.9.4 接收缓冲器

接收缓冲器的列表与发送缓冲器很相似。接收缓冲器是 RXFIFO 的可访问部分，位于 CAN 地址的 0x40 ~ 0x70。每条信息都分为描述区和数据区。

注：在帧信息字节中的接收字节长度代码代表发送器实际发送的数据长度代码，它有可能大于 8（取决于发送器）。而本接收器 RXFIFO 最大为 8 字节。这一点在读接收缓冲器中的信息时应当考虑。

见下图，RXFIFO 共有 64 个信息字节的空间。一次可以存储多少条信息取决于数据的长度。如果 RXFIFO 中没有足够的空间来存储新的信息，CAN 控制器会产生数据溢出条件，此时信息有效且接收检测为肯定。发生数据溢出情况时，已部分写入 RXFIFO 的信息将被删除。这种情况可以通过状态寄存器和数据溢出中断（中断允许）反应到 CPU。

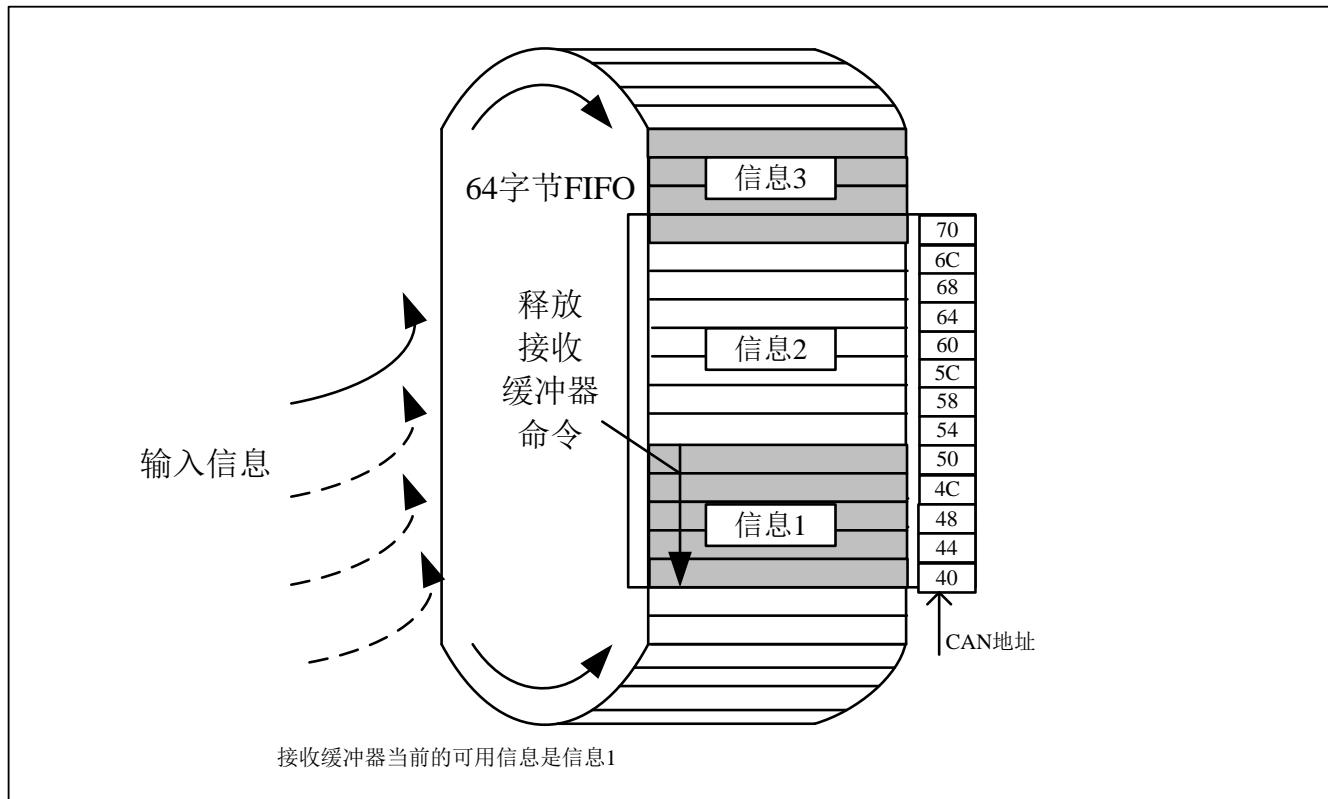


图 22-9 RXFIFO 中的信息存储举例

22.3.10 出错管理

基于错误计数器的值，每个 CAN 控制器能够在三种错误状态之一中工作：错误主动、错误被动或总线关闭。如果错误计数器的值都在 0 ~ 127 之间，CAN 控制器是错误主动状态。此时产生错误主动标志（6 个显性位）。如果一个错误计数器的值在 128 ~ 255 之间，CAN 控制器是错误被动的。此时，在检测到错误前，产生错误被动标志（6 个隐性位）。如果发送错误计数器的值高于 255，则到达总线关闭状态。在这种状态下，自动置位复位请求，CAN 控制器对总线没有影响。总线关闭状态只能在微控制器用命令‘复位请求 = 0’退出。这将启动总线关闭恢复定时器，发送错误计数器计数 128 个总线释放信号。计数结束后，两个错误计数器都是 0，器件再次处于错误主动状态。

22.3.10.1 错误计数器

如上面描述，CAN 的错误状态和发送错误计数器和接收错误计数器的值直接相关。

为了仔细研究错误界定，支持 CAN 控制器的增强的错误分析功能，CAN 控制器提供可读的错误计数器。另外，在复位模式，允许对于两个错误计数器进行写访问。

22.3.10.2 出错中断

有三个中断源来向微处理器发出错的状态。每个中断都能在中断使能寄存器里分别使能。

- 总线错误中断：在 CAN 总线上检测到任何一个错误都会产生中断

- 出错警告中断：如果超过出错警告值，产生出错警告中断。而且它在 CAN 控制器进入总线关闭状态和在此之前再一次进入错误主动状态也会产生这个中断。CAN 控制器的出错警告值在复位模式中可编程。复位后的默认值是 96。
- 错误主动中断：如果错误状态从错误主动变成错误被动或相反，将产生错误主动中断。

22.3.10.3 错误码捕捉

CAN 控制器可以执行在 CAN2.0B 规范定义的所有错误界定。每个 CAN 控制器处理错误的整个过程是完全自动的。但是，为了向用户提供某个错误的详细信息，CAN 控制器提供了错误代码捕捉功能。无论什么时候发生 CAN 总线错误，它都会强制产生相应的总线出错中断。同时，当前位的位置被捕捉入错误代码捕捉寄存器。在主控制器将捕捉的数据读出前，它都会被保存在寄存器中。然后捕捉机制再次激活。寄存器内容可以区分四种错误类型：格式错误、填充错误、位错误和其它错误。如下图表示，寄存器还另外表明在错误是在报文的接收还是发送期间发生。这个寄存器中的五个位表示 CAN 帧内错误的位位置，更多信息参考下面的表和数据表。

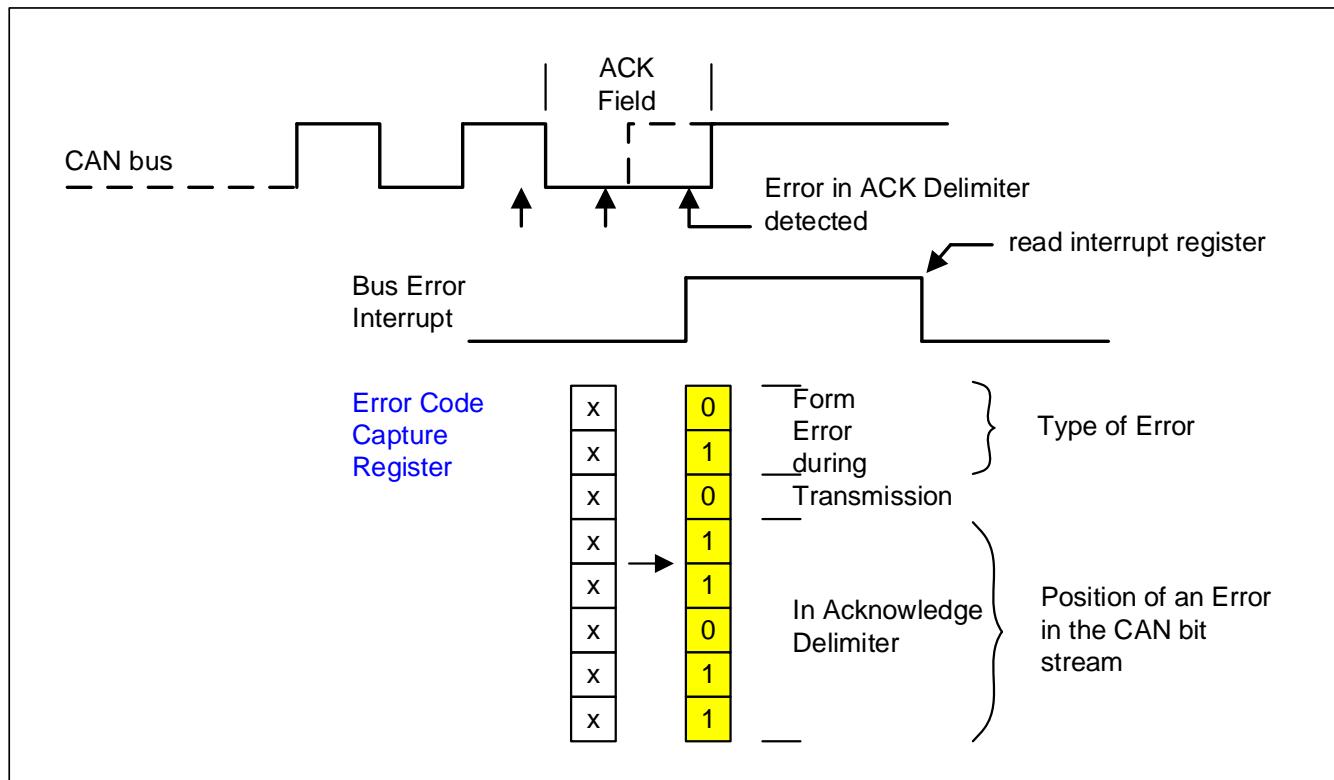


图 22-10 错误码捕捉功能举例

CAN 规范定义了：CAN 总线上的每个位只有特殊类型的错误。下面两张表格显示了 CAN 报文发送和接收期间可能出现的所有错误。左边的部分包括位置和错误的类型，这些由错误码捕捉寄存器捕捉。每张表的右边部分是将错误码转换成上层的错误描述，可以直接从寄存器内容知道其含义。通过使用这些表格，能得到有关错误计数器的变化和在器件发送和接收管脚的错误状态的更多信息。使用这些表时，例如在错误分析软件里，可以详细地分析每个错误状态。关于 CAN 错误类型和位置的信息能用于错误统计和系统维护或在系统优化器件进行纠正。

表 22-5 接收时可能出现的错误

CAN 位流里的错误位置	错误类型	RX 错误计数	错误码捕捉	描述
标识符 SRR、IDE 和 RTR 位 保留位 数据长度码 数据场 CRC 序列	填充	+1	收到 5 个电平相同的连续位	
CRC 定界符	格式填充	+1 +1	RX = 显性 收到超过 5 个电平相同的连续的位	位必须是隐性
应答位	位	+1	RX = 显性, 或检测到 CRC 错误	临界的总线定时或总线长度 CRC 序列不正确
帧结束	格式其他	+1 ±0	RX = 头六位是显性 RX = 最后一位的显性	反应: 发出超载标志, 如果发送器重新发送, 数据可能重复
间隔	其他	±0	RX = 显性	反应: 接收器发出过载标志
主动错误标志	格式其他	+8	TX = 显性, 但 RX = 隐性	不能写显性位
容许的显性位	其他	+8	TX = 显性, 但 RX = 隐性	不能写显性位
错误定界符	格式其他	+8	TX = 显性, 但 RX = 隐性	不能写显性位
过载标志	位	+8	TX = 显性, 但 RX = 隐性	不能写显性位

表 22-6 发送时可能出现的错误

CAN 位流里的错误位置	错误类型	TX 错误计数	错误码捕捉	描述
帧开始	位	+8	TX = 显性, 但 RX = 隐性	不能写显性位
标识符	位 填充	+8 ±0	TX = 显性, 但 RX = 隐性 TX = 显性, 但 RX = 隐性	不能写显性位
SRR 位	位 填充	+8 ±0	TX = 显性, 但 RX = 隐性 TX = 显性, 但 RX = 隐性	不能写显性位
IDE 和 RTR 位	位 填充	+8 ±8	TX = 显性, 但 RX = 隐性 TX = 显性, 但 RX = 隐性	不能写显性位
保留位 数据长度码 数据场 CRC 序列	位	+8	TX = 显性, 但 RX = 隐性	不能写显性位
CRC 定界符	格式	+8	RX = 隐性	位必须是隐性
应答隙	其他	+8 ±0	RX = 隐性 (错误激活) RX = 隐性 (错误认可)	没有应答 没有应答, 节点可能单独在总线上
应答应界符	格式	+8	RX = 显性	临界的总线定时或总线长度

CAN 位流里的错误位置	错误类型	TX 错误计数	错误码捕捉	描述
帧结束	格式其他	+8 ±8	RX = 头六位是显性 RX = 最后一位是显性位	帧已经被一些节点接收，再次发送可能导致接收器里数据重复
间隔	其他	±0	RX = 显性	来自于“旧” CAN 控制器的超载标志
主动错误标志 过载标志	位	+8	TX = 显性，但 RX = 隐性	不能写显性位
允许显性位	格式	+8	RX = 在激活错误标志或过载标志后有超过 7 个显性位	
错误定界符	格式其他	+8 ±0	RX = 头七位是显性位 RX = 定界符的最后一位是显性位	
认可错误标志	其他	+8	RX = 显性（错误认可）	没有收到应答，节点不是单独在总线上。

22.3.10.4 总线关闭恢复

如果发送错误计数器的值高于 255，则达到总线关闭状态。总线状态位被置为‘1’。在这种状态下，自动置位复位请求位，CAN 控制器对总线没有影响。在错误中断允许的情况下，会产生一个错误中断。这种状态会持续到 CPU 清除复位请求位。所有这些完成后，CAN 控制器将会等待协议规定的最长时间（128 个总线空闲信号）。

22.3.11 位时序控制

位时序控制是通过采样的方式监视串行的 CAN 总线，其采样点可以通过同步帧起始位的边沿，及重新同步后面的边沿来调整。

名义上的每位的时间可以分为 3 段：

1. 同步时段 ($t_{SYNCSEG}$)：通常期望位的变化发生在该时间段内。其值固定为 1 个时间单元 ($1 \times t_q$)。
2. 第一时间段 (t_{TSEG1})：定义采样点的位置。它包含 CAN 标准里的 PROP_SEG 和 PHASE_SEG1。其值可以编程为 1 到 16 个时间单元，但也可以被自动延长，以补偿因为网络中不同节点的频率差异所造成的相位的正向偏移。
3. 第二时间段 (t_{TSEG2}) 定义发送点的位置。它代表 CAN 标准里的 PHASE_SEG2。其值可以编程为 1 到 8 个时间单元，但也可以被自动缩短以补偿相位的负向偏移。

CAN 系统时钟的时间单元 t_q 是可编程的，而且决定了相应的位时序。CAN 系统时钟的时间单元由如下公式计算： $t_q = 2 \times t_{PCLK1} \times (BRP + 1)$ 。这里 $t_{PCLK1} = APB1$ 时钟的时间周期。

为了补偿在不同总线控制器的时钟振荡器之间的相位偏移，任何总线控制器必须在当前传送的相关信号边沿重新同步。同步跳转宽度 (SJW) 定义了每一位周期可以被重新同步缩短或延长的时钟周期的最大数目，且 $t_{SJW} = t_q \times (SJW + 1)$ 。SJW 不参与波特率和采样点的计算，其值可以编程为 1 到 4 个时间

单元，表示该总线对波特率范围的容差度。第一时间段 (t_{TSEG1}) 和第二时间段 (t_{TSEG2}) 决定了每一位的时钟数目和采样点的位置，采样点的计算公式为： $\text{Sample} = (1 + t_{TSEG1}) / (1 + t_{TSEG1} + t_{TSEG2})$ ，这里：

$$t_{SYNCSEG} = 1 \times t_q$$

$$t_{TSEG1} = t_q \times (TSEG1 + 1)$$

$$t_{TSEG2} = t_q \times (TSEG2 + 1)$$

有效跳变被定义为，当 CAN 自己没有发送隐性位时，从显性位到隐性位的第 1 次转变。如果在第一时间段 (t_{TSEG1}) 而不是在同步时段 ($t_{SYNCSEG}$) 检测到有效跳变，那么 t_{TSEG1} 的时间就被延长最多 SJW 那么长，从而采样点被延迟了。

相反如果在第二时间段 (t_{TSEG2}) 而不是在 $t_{SYNCSEG}$ 检测到有效跳变，那么 t_{TSEG2} 的时间就被缩短最多 SJW 那么长，从而采样点被提前了。

为了避免软件的编程错误，对位时序控制寄存器 (CAN_BTR0 和 CAN_BTR1) 的设置只能在 CAN 处于初始化状态下进行。CAN 波特率计算公式如下：

$$\text{CAN 波特率} = F_{\text{pclk1}} / (2^* (\text{BRP} + 1) * (TSEG1 + 1 + TSEG2 + 1 + 1))$$

这里 F_{pclk1} 为 CAN 模块所在的 APB1 时钟总线频率，BRP 为 CAN_BTR0[5:0]，TSEG1 和 TSEG2 详见 CAN_BTR0[6:0] 中的描述。

22.3.12 仲裁丢失

仲裁丢失时，会产生相应的仲裁丢失中断（中断允许）。同时，位流处理器的当前位位置被捕获送入仲裁丢失捕捉寄存器。一直到用户通过软件读这个值，寄存器中的内容都不会变，随后，捕捉机制又被激活了。

读中断寄存器时，中断寄存器中相应的中断标志位被清除。直到仲裁丢失捕捉寄存器被读一次之后，新的仲裁丢失中断才有效。

下图为仲裁位解释：

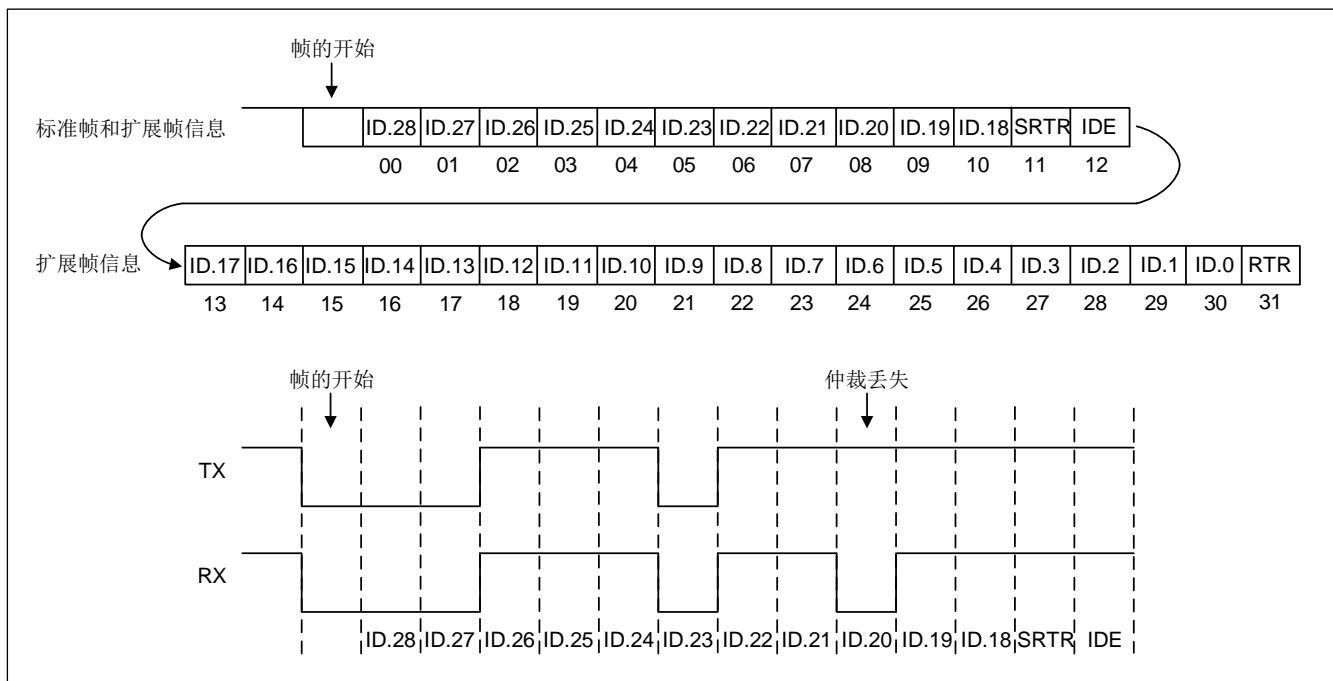


图 22-11 仲裁丢失解释举例

22.3.13 中断

- BasicCAN 模式里有 5 个中断:
 - ◆ 接收中断: 当接收 FIFO 不空且对应的中断使能时产生。标志为 CAN_IR.RI 位。
 - ◆ 发送中断: 发送缓冲器状态从 ‘0’ 变为 ‘1’ (释放) 且对应的中断使能时产生。标志为 CAN_IR.TI 位。
 - ◆ 错误中断: 错误状态位或总线状态位变化且对应的中断使能时产生。标志为 CAN_IR.EI 位。
 - ◆ 数据溢出中断: 数据溢出且对应的中断使能时产生。标志为 CAN_IR.DOI 位。
 - ◆ 唤醒中断: 退出睡眠模式且对应的中断使能时产生, 标志为 CAN_IR.WUI 位。
- PeliCAN 模式里有 8 个中断:
 - ◆ 接收中断: 当接收 FIFO 不空且对应的中断使能时产生。标志为 CAN_IR.RI 位。
 - ◆ 发送中断: 发送缓冲器状态从 ‘0’ 变为 ‘1’ (释放) 且对应的中断使能时产生。标志为 CAN_IR.TI 位。
 - ◆ 错误报警中断: 错误状态位和总线状态位的改变且对应的中断使能时产生。标志为 IR.EI。
 - ◆ 数据溢出中断: 数据溢出且对应的中断使能时产生。标志为 CAN_IR.DOI 位。
 - ◆ 错误被动中断: 当 CAN 控制器到达错误被动状态 (至少一个错误计数器超过协议规定的值 127) 或从错误被动状态又进入错误主动状态且对应的中断使能时产生。标志为 CAN_IR.EPI。
 - ◆ 仲裁丢失中断: 当 CAN 控制器仲裁丢失且对应的中断使能时产生, 此时 CAN 控制自动变为接收器。标志为 CAN_IR.ALI。

- ◆ 总线错误中断：当 CAN 控制器检测到总线上存在错误且对应的中断使能时产生。标志为 `IR.BEI`。
- ◆ 唤醒中断：退出睡眠模式且对应的中断使能时产生，标志为 `CAN_IR.WUI` 位。

22.4 寄存器

22.4.1 寄存器总览

表 22-7 BasicCAN 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CAN_MOD	CAN 模式寄存器	0x0000 0001
0x04	CAN_CMR	CAN 命令寄存器	0x0000 00FF
0x08	CAN_SR	CAN 状态寄存器	0x0000 000C
0x0C	CAN_IR	CAN 中断寄存器	0x0000 00E0
0x10	GROUP0_ACR	CAN 验收代码寄存器组 0	0x0000 00XX
0x14	GROUP0_AMR	CAN 验收屏蔽寄存器组 0	0x0000 00XX
0x18	CAN_BTR0	CAN 位时序寄存器 0	0x0000 0000
0x1C	CAN_BTR1	CAN 位时序寄存器 1	0x0000 0000
0x28	CAN_TXID0	CAN 发送识别码寄存器 0	0x0000 00XX
0x2C	CAN_TXID1	CAN 发送识别码寄存器 1	0x0000 00XX
0x30 + x*0x4	CAN_TXDRx	CAN 发送数据寄存器 x (x=0~7)	0x0000 0000
0x50	CAN_RXID0	CAN 接收识别码寄存器 0	0x0000 0000
0x54	CAN_RXID1	CAN 接收识别码寄存器 1	0x0000 0000
0x58 + x*0x4	CAN_RXDRx	CAN 接收数据寄存器 x (x=0~7)	0x0000 0000
0x7C	CAN_CDR	CAN 时钟分频寄存器	0x0000 0000

表 22-8 PeliCAN 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CAN_MOD	CAN 模式寄存器	0x0000 0001
0x04	CAN_CMR	CAN 命令寄存器	0x0000 0000
0x08	CAN_SR	CAN 状态寄存器	0x0000 000C
0x0C	CAN_IR	CAN 中断寄存器	0x0000 0000
0x10	CAN_IER	CAN 中断使能寄存器	0x00000000
0x18	CAN_BTR0	CAN 位时序寄存器 0	0x00000000
0x1C	CAN_BTR1	CAN 位时序寄存器 1	0x0000 0000
0x2C	CAN_ALC	CAN 仲裁丢失捕捉寄存器	0x0000 0000
0x30	CAN_ECC	CAN 错误代码捕捉	0x0000 0000
0x34	CAN_EWLR	CAN 错误报警限制寄存器	0x0000 0060
0x38	CAN_RXERR	CAN RX 错误计数寄存器	0x0000 0000
0x3C	CAN_TXERR	CAN TX 错误计数寄存器	0x0000 0000
0x40	CAN_SFF	CAN 发送帧信息寄存器	0x0000 00XX
0x44	CAN_TXIDO	CAN 识别码寄存器 0	0x0000 00XX
0x48	CAN_TXID1	CAN 识别码寄存器 1	0x0000 00XX
0x4C + x*0x4	CAN_TXDATAx	CAN 数据寄存器 x (x=0~9)	0x0000 00XX
0x7C	CAN_CDR	CAN 时钟分频寄存器	0x0000 0000

Offset	Acronym	Register Name	Reset
0x80	CAN_AFM0	CAN 滤波模式寄存器 0	0x0000 0000
0x84	CAN_AFM1	CAN 滤波模式寄存器 1	0x0000 0000
0x88	CAN_AFM2	CAN 滤波模式寄存器 2	0x0000 0000
0x8C	CAN_FGA0	CAN 滤波组使能寄存器 0	0x0000 0001
0x90	CAN_FGA1	CAN 滤波组使能寄存器 1	0x0000 0000
0x94	CAN_FGA2	CAN 滤波组使能寄存器 2	0x0000 0000
0x98+ (x-1) *0x20	GROUPx_ACR	CAN 验收代码寄存器组 x (x = 1-19)	0x000000XX
0xA8+ (x-1) *0x20	GROUPx_AMR	CAN 验收屏蔽寄存器组 x (x = 1-19)	0x000000XX

22.4.2 CAN_MOD 模式寄存器

偏移地址: 0x00

复位值: 0x0000 0001

仅 PeliCAN 模式

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

Bit	Field	Description
31: 4	Reserved	保留, 必须保持复位值。
3	AFM	验收滤波器模式 (Acceptance filter mode) 1: 单; 选择单个验收滤波器 (32 位长度) 0: 双; 选择两个验收滤波器 (每个有 16 位有效)
2	STM	自检测模式 (Self test mode) 1: 自检测; 此模式在没有其他活动的节点的情况下, 使用自接收功能可以完成节点的全部功能测试。该模式下即使没有应答, CAN 控制器也会成功发送 0: 正常模式; 成功发送时必须应答信号
1	LOM	只听模式 (Listen only mode) 1: 只听; 此模式下即使成功接收信息, CAN 控制器也不向总线发应答信号; 错误计数器停止在当前值 0: 正常模式
0	RM	复位模式 (Reset mode) 1: 复位; 检测到复位模式位被置位, 中止目前正在接收/发送的信息, 进入复位模式 0: 正常; 复位模式位接收到'1-0'的跳变后, CAN 控制器回到工作模式

22.4.3 CAN_CR 控制寄存器

偏移地址: 0x00

复位值: 0x0000 0021

仅 BasicCAN 模式

控制寄存器的内容是用于改变 CAN 控制器的行为。这些位可以被微控制器设置或置位，微控制器可以对控制寄存器进行读/写操作。

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

Bit	Field	Description
31: 5	Reserved	保留，必须保持复位值。
4	OIE	溢出中断使能 (Overflow interrupt enable) 1: 使能；如果数据溢出位置位，微控制器接收溢出中断信号 0: 禁止；微控制器不从 CAN 控制器接收溢出中断信号
3	EIE	错误中断使能 (Error interrupt enable) 1: 使能；如果出错或总线状态改变，微控制器接收错误中断信号 0: 禁止；微控制器不从 CAN 控制器接收错误中断信号
2	TIE	发送中断使能 (Transmit interrupt enable) 1: 使能；当信息被成功发送或发送缓冲器又能被访问时(例如，中止发送命令后)，微控制器接收 CAN 控制器发出的一个发送中断信号 0: 禁止；微控制器不从 CAN 控制器接收发送中断信号
1	RIE	接收中断使能 (Receive interrupt enable) 1: 使能；信息被无错误接收时，CAN 控制器发出的一个接收中断信号到微控制器 0: 禁止；微控制器不从 CAN 控制器接收中断信号
0	RR	复位请求 (Reset request) 1: CAN 控制器检测到复位请求后，中止当前发送/接收的信息，进入复位模式 0: 复位请求位接收到一个下降沿后，CAN 控制器回到工作模式

22.4.4 CAN_CMRR 命令寄存器

偏移地址: 0x04

复位值: BasicCAN 模式: 0x0000 00FF

PeliCAN 模式: 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
										ERB	SRR	CDO	RRB	AT	TR
										W	W	W	W	W	W

Bit	Field	Description
31: 6	Reserved	保留，必须保持复位值。
5	ERB	<p>清空所有 RXFIFO 数据 (Empty RXFIFO)</p> <p>1 清除：清除 RXFIFO 所有内容</p> <p>0：无动作</p> <p>清空 RXFIFO 数据这个命令位是用来清除数据溢出情况的。但不清除溢出状态标志位。如果数据溢出，且置位此位，接收缓存器中的内容将全部被释放。</p>
4	SRR	<p>PeliCAN 模式：</p> <p>SRR: 自接收请求 (Self reset request)</p> <p>1: 当前信息可被同时发送和接收</p> <p>0: 不支持自接收</p>
3	CDO	<p>清除数据溢出 (Clear data overrun)</p> <p>1: 清除数据溢出状态位</p> <p>0: 无动作</p> <p>清除数据溢出这个命令位是用来清除由数据溢出状态位的。释放接收缓冲器命令的同时是可以发出清除数据溢出命令的。</p>
2	RRB	<p>释放接收缓冲器 (Release receive buffer)</p> <p>1: 释放；接收缓冲器中存放信息的内存空间将被释放</p> <p>0: 无动作</p> <p>读接收缓冲器之后，微控制器可以通过设置该位为'1'来释放 RXFIFO 中当前信息的内存空间。这可能会导致接收缓冲器中的另一条信息立即有效。这样会再产生一次接收中断（使能条件下）。如果没有其它可用信息，就不会再产生接收中断，接收缓冲器状态位被清除。</p>
1	AT	<p>中止传输 (Abort transmission)</p> <p>1: 等待处理的发送请求将取消，正在发送的信息不受影响</p> <p>0: 无动作</p> <p>中止传送位是在 CPU 要求当前传送中止时使用的，例如，传送一条紧急信息。正在进行的传送是不停止的。要查看原始信息是否被成功发送，可以通过传送成功状态位来检测。不过，这必须在发送缓冲器状态位为 1 (释放) 或发送中断产生的条件下才能实现。</p>
0	TR	<p>发送请求 (Transmission request)</p> <p>1: 请求发送消息</p> <p>0: 无动作</p> <p>为'1'后不能通过设置该位为'0'取消发送，但可通过设置 AT 位中止传输。</p>

22.4.5 CAN_SR 状态寄存器

偏移地址: 0x08

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
								BS	ES	TS	RS	TCS	TBS	DOS	RBS
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7	BS	<p>BS: 总线状态 (Bus status)</p> <p>1: 总线关闭; CAN 控制器退出总线活动</p> <p>0: 总线开启; CAN 控制器加入总线活动</p>
6	ES	<p>出错状态 (Error status)</p> <p>1: 出错; 至少出现一个错误计数器满或超过报警限制</p> <p>0: OK; 两个错误计数器都在报警限制以下</p> <p>根据 CAN 2.0B 协议说明, 在接收或发送时检测到错误时会影响错误计数。当至少有一个错误计数器满或超出警告限制 (96) 时, 错误状态位被置位。在中断允许的情况下, 会产生错误中断。</p>
5	TS	<p>发送状态 (Transmit status)</p> <p>1: 发送; CAN 控制器在发送信息中</p> <p>0: 空闲;</p> <p>如果接收状态位和发送状态位都是 '0', 则 CAN 总线是空闲的。</p>
4	RS	<p>接收状态 (Receive status)</p> <p>1: 接收; CAN 控制器正在接收信息</p> <p>0: 空闲; 没有正在接收的信息</p> <p>如果接收状态位和发送状态位都是 '0', 则 CAN 总线是空闲的</p>
3	TCS	<p>发送完成状态 (Transmission complete status)</p> <p>1: 完成; 最近一次发送请求被成功处理</p> <p>0: 未完成; 当前发送请求未处理完毕</p> <p>无论何时发送请求位被置为'1', 发送完成位都会被置为'0' (未完成)。发送完成位的'0'会一直保持到信息被成功发送。</p>
2	TBS	<p>发送缓冲器状态 (Transmit buffer status)</p> <p>1: 释放; CPU 可以向发送缓冲器写信息</p> <p>0: 锁定; CPU 不能访问发送缓冲器; 有信息正在等待发送或正在发送</p> <p>如果 CPU 在发送缓冲器状态位是 '0' (锁定) 时试图写发送缓冲器, 则写入的字节被拒绝接收且会在无任何提示的情况下丢失。</p>
1	DOS	<p>数据溢出状态 (Data overrun status)</p> <p>1: 溢出; 信息丢失, 因为 RXFIFO 中没有足够的空间来存储它</p> <p>0: 当前无数据溢出发生</p> <p>当要被接收的信息成功的通过验收滤波器后 (例如, 仲裁后之初), CAN 控制器需要在 RXFIFO 中用一些空间来存储这条信息的描述符。因此必须有足够的空间来存储接收的每一个数据字节。如果没有足够的空间存储信息, 信息将会丢失且只向</p>

Bit	Field	Description
		CPU 提示数据溢出情况。如果这个接收到的信息除了最后一位之外都无错误，信息有效。
0	RBS	接收缓冲器状态 (Receive buffer status) 1: 满; RXFIFO 中有可用信息 0: 空; 无可用信息在读 RXFIFO 中的信息且用释放接收缓冲器命令来释放内存空间之后，这一位被清除。如果 FIFO 中还有可用信息，此位将在下一位的时限 (t_q) 中被重新设置。

22.4.6 CAN_IR 中断寄存器

偏移地址: 0x0C

复位值: BasicCAN 模式: 0x0000 00E0

PeliCAN 模式: 0x0000 0000

中断寄存器允许中断源的识别。当寄存器的一位或多位被置位时中断就被激活了。中断寄存器对微控制器来说是只读存储器。

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7	BEI	BasicCAN 模式: 保留, 读出值为 1 PeliCAN 模式: BEI: 总线错误中断 (Bus error interrupt) 1: 置位; 当 CAN 控制器检测到总线错误且中断使能寄存器中的 BEIE 被置位时此位被置位 0: 复位
6	ALI	BasicCAN 模式: 保留, 读出值为 1 PeliCAN 模式: ALI: 仲裁丢失中断 (Arbitration lost interrupt) 1: 置位; 当 CAN 控制器丢失仲裁, 变为接收器和中断使能寄存器的 ALIE 为被置位时, 此位被置位 0: 复位
5	EPI	BasicCAN 模式: 保留, 读出值为 1 PeliCAN 模式:

Bit	Field	Description
		EPI: 错误被动中断 (Error passive interrupt) 1: 置位; 当 CAN 控制器到达错误被动状态 (至少一个错误计数器超过协议规定的值 127) 或从错误被动状态又进入错误主动状态以及中断寄存器的 EPIE 位被置位时此位被置‘1’ 0: 复位
4	Reserved	保留, 必须保持复位值。
3	DOI	数据溢出中断 (Data overrun interrupt) 1: 置位; 当数据溢出中断使能位被置为‘1’时, 数据溢出状态位‘0—1’跳变, 此位被置位 0: 复位; 微控制器的任何读访问将清除此位 溢出中断位 (中断允许情况下) 和溢出状态位是同时被置位的。
2	EI	错误中断 (Error interrupt) 1: 置位; 错误中断使能时, 错误状态位或总线状态位的变化会置位此位 0: 复位; 微控制器的任何读访问将清除此位
1	TI	发送中断 (Transmit interrupt) 1: 置位; 发送缓冲器状态从 ‘0’ 变为 ‘1’ (释放) 和发送中断使能时, 置位此位 0: 复位; 微控制器的任何读访问将清除此位
0	RI	接收中断 (Receive interrupt) 1: 置位; 当接收 FIFO 不空和接收中断使能时置位此位 0: 复位; 微控制器的任何读访问将清除此位 接收中断位 (中断允许时) 和接收缓冲器状态位是同时置位的。 必须说明的是接收中断位在读的时候被清除, 即使 FIFO 中还有其它可用信息。当释放接收缓冲器命令执行后, 接收缓冲器中还有其它可用信息, 接收中断 (中断允许时) 会在下一个 t_q 被重置。

22.4.7 CAN_IER 中断使能寄存器

偏移地址: 0x10

复位值: 0x0000 0000

仅存在 PeliCAN 模式

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7	BEIE	总线错误中断使能 (Bus error interrupt enable) 1: 使能; 如果检测到总线错误, 则 CAN 控制器请求相应的中断 0: 禁止

Bit	Field	Description
6	ALIE	仲裁丢失中断使能 (Arbitration lost interrupt enable) 1: 使能; 如果 CAN 控制器已丢失了仲裁, 则请求相应的中断 0: 禁止
5	EPIE	错误被动中断使能 (Error passive interrupt enable) 1: 使能; 若 CAN 控制器的错误状态改变 (从被动到主动或反之), 则请求相应的中断 0: 禁止
4	Reserved	保留, 必须保持复位值。
3	DOIE	数据溢出中断使能 (Data overrun interrupt enable) 1: 使能; 如果数据溢出状态位被置位 (见状态寄存器), CAN 控制器请求相应的中断 0: 禁止
2	EIE	错误报警中断使能 (Error interrupt enable) 1: 使能; 如果错误或总线状态改变 (见状态寄存器), CAN 控制器请求相应的中断 0: 禁止
1	TIE	发送中断使能 (Transmit interrupt enable) 1: 使能; 当信息被成功发送或发送缓冲器又可访问 (例如, 中止发送命令后) 时, CAN 控制器请求相应的中断 0: 禁止
0	RIE	接收中断使能 (Receive interrupt enable) 1: 使能; 当接收缓冲器状态是‘满’时, CAN 控制器请求相应的中断 0: 禁止

22.4.8 GROUP0_ACR 验收代码寄存器组 0

偏移地址: 0x10

复位值: 0x0000 00XX

BasicCAN 模式: GROUP0_ACR

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 0	AC	(Acceptance code) 复位请求位被置高 (当前) 时, 这个寄存器是可以访问 (读/写) 的。如果一条信息通过了验收滤波器的测试而且接收缓冲器有空间, 那么描述符和数据将被分别顺次写入 RXFIFO。

Bit	Field	Description
		<p>当信息被正确的接收完成就会:</p> <p>接收状态位置高 (满)</p> <p>接收中断使能位置高 (使能) 接收中断置高 (产生中断)</p> <p>验收代码位 (AC.7—AC.0) 和信息识别码的高 8 位 (ID.10—ID.3) 相等, 且与验收屏蔽位 (AM.7—AM.0) 的相应位相或为 1。即如果满足以下方程的描述, 则被接收:</p> $((ID.10 - ID.3) = (AC.7 - AC.0)) \mid (AM.7 - AM.0) = 11111111$

PeliCAN 模式: 第 0 组有四个验收代码寄存器分别是 GROUP0_ACR0, GROUP0_ACR1, GROUP0_ACR2, GROUP0_ACR3

GROUP0_ACR0: 偏移地址: 0x40 复位值: 0x0000 00XX

GROUP0_ACR1: 偏移地址: 0x44 复位值: 0x0000 00XX

GROUP0_ACR2: 偏移地址: 0x48 复位值: 0x0000 00XX

GROUP0_ACR3: 偏移地址: 0x4C 复位值: 0x0000 00XX

注: 详细说明见的标识符过滤中 PeliCAN 模式介绍。这是第 0 组验收代码寄存器。其他组见对应的寄存器章节。

22.4.9 GROUP0_AMR 验收屏蔽寄存器组 0

偏移地址: 0x14

复位值: 0x0000 00XX

BasicCAN 模式: GROUP0_AMR

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

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 0	AM	(Acceptance mask) 如果复位请求位置高 (当前) 这个寄存器可以被访问 (读/写)。验收屏蔽寄存器定义验收代码寄存器的相应位对验收滤波器是“相关的”或“无影响的” (即可为任意值)。

PeliCAN 模式: 第 0 组有四个验收屏蔽寄存器分别是 GROUP0_AMR0, GROUP0_AMR1, GROUP0_AMR2, GROUP0_AMR3

GROUP0_AMR0: 偏移地址: 0x50 复位值: 0x0000 00XX

GROUP0_AMR1: 偏移地址: 0x54 复位值: 0x0000 00XX

GROUP0 AMR2: 偏移地址: 0x58 复位值: 0x0000 00XX

GROUP0 AMR3: 偏移地址: 0x5C 复位值: 0x0000 00XX

注：详细说明见的标识符过滤中 peliCAN 模式介绍。这是第 0 组验收屏蔽寄存器，其他组见对应的寄存器章节。

22.4.10 CAN_BTR0 位时序寄存器 0

偏移地址: 0x18

复位值: 0x0000 0000

位时序寄存器 0 定义了波特率预分频值 (BRP) 和同步跳转宽度 (SJW) 的值。复位模式有效时这个寄存器是可以被访问 (读/写) 的。

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 6	SJW	<p>同步跳转宽度 (Synchronization jump width)</p> <p>为了补偿在不同总线控制器的时钟振荡器之间的相位偏移，任何总线控制器必须在当前传送的相关信号边沿重新同步。同步跳转宽度定义了每一位周期可以被重新同步缩短或延长的时钟周期的最大数目。</p>
5: 0	BRP	<p>波特率预分频值 (Baud rate prescaler)</p> <p>CAN 时间单元 t_q 的周期是可编程的而且决定了相应的位时序。</p>

22.4.11 CAN_BTR1 位时序寄存器 1

偏移地址: 0x1C

复位值: 0x0000 0000

位时序寄存器 1 定义了每个位周期的长度、采样点的位置和采样次数。在复位模式中，这个寄存器可以被读/写访问。

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.								SAM	TESG2			TESG1			
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7	SAM	采样 (Sampling) 1: 三倍; 总线采样三次; 建议在低/中速总线 (class A 和 class B) 上使用, 这对过滤总线上的毛刺波是有益的 0: 单倍; 总线采样一次; 建议使用在高速总线上 (SAE class C)
6: 4	TSEG2	第一时间段 (Time segment 1)
3: 0	TSEG1	第二时间段 (Time segment 2)

22.4.12 CAN_TXID0 发送识别码寄存器 0

偏移地址: 0x28

复位值: 0x0000 00XX

注: BasicCAN 模式, 复位模式为 0xFF

发送识别码寄存器 0 定义发送帧的类型与数据长度。仅在工作模式下这个寄存器可以被读/写访问。

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.								ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 0	IDx	CAN 识别码 10 ~ 3 (CAN identifier byte 10 ~3) 注: 在 PeliCAN 模式下, 位 [3: 0] 为 DLC[3: 0]

22.4.13 CAN_TXID1 发送识别码寄存器 1

偏移地址: 0x2C

复位值: 0x0000 00XX

注: 仅存在 BasicCAN 模式, 复位模式为 0xFF

发送识别码寄存器 1 定义发送帧的类型与数据长度。仅在工作模式下这个寄存器可以被读/写访问。

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.								ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 5	IDx	CAN 识别码 2 ~ 0 (CAN identifier byte 2 ~ 0)
4	RTR	帧格式 (Remote transmission request)
3: 0	DLCx	发送数据区长度 0 ~ 8 (Data length code 0 ~ 8)

仅存在 BasicCAN 模式:

偏移地址: 0x30 ~ 0x4C

复位值: 0x0000 0000

发送数据寄存器 CAN_TXDR0 ~ 7。仅在工作模式下这个寄存器可以被读/写访问。接收缓冲与发送缓冲数据格式一致。

22.4.14 CAN_ALC 仲裁丢失捕捉寄存器

偏移地址: 0x2C

复位值: 0x0000 0000

仅存在 PeliCAN 模式

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

Bit	Field	Description
31: 5	Reserved	保留，必须保持复位值。
4: 0	BITNO	值和功能参考下表 (Bit number) 仲裁丢失时，会产生相应的仲裁丢失中断 (中断允许)。同时，位流处理器的当前位置被捕获送到仲裁丢失捕捉寄存器。直到用户通过软件读这个值，寄存器中的内容都不会变。读之后，捕捉机制又被激活。读中断寄存器时，中断寄存器中相应的中断标志位被清除。直到仲裁丢失捕捉寄存器被读一次之后，新的仲裁丢失中断才有效。

位	ALC.4	ALC.3	ALC.2	ALC.1	ALC.0	十进制值	功能
0	0	0	0	0	0	0	仲裁丢失在识别码的 bit1
0	0	0	0	1	1	1	仲裁丢失在识别码的 bit2
0	0	0	1	0	2	2	仲裁丢失在识别码的 bit3
0	0	0	1	1	3	3	仲裁丢失在识别码的 bit4
0	0	1	0	0	4	4	仲裁丢失在识别码的 bit5
0	0	1	0	1	5	5	仲裁丢失在识别码的 bit6

位						十进制值	功能
0	0	1	1	0	6	仲裁丢失在识别码的 bit7	
0	0	1	1	1	7	仲裁丢失在识别码的 bit8	
0	1	0	0	0	8	仲裁丢失在识别码的 bit9	
0	1	0	0	1	9	仲裁丢失在识别码的 bit10	
0	1	0	1	0	10	仲裁丢失在识别码的 bit11	
0	1	0	1	1	11	仲裁丢失在 SRTR 位	
0	1	1	0	0	12	仲裁丢失在 IDE 位	
0	1	1	0	1	13	仲裁丢失在识别码的 bit12	
0	1	1	1	0	14	仲裁丢失在识别码的 bit13	
0	1	1	1	1	15	仲裁丢失在识别码的 bit14;	
1	0	0	0	0	16	仲裁丢失在识别码的 bit15;	
1	0	0	0	1	17	仲裁丢失在识别码的 bit16;	
1	0	0	1	0	18	仲裁丢失在识别码的 bit17;	
1	0	0	1	1	19	仲裁丢失在识别码的 bit18;	
1	0	1	0	0	20	仲裁丢失在识别码的 bit19;	
1	0	1	0	1	21	仲裁丢失在识别码的 bit20;	
1	0	1	1	0	22	仲裁丢失在识别码的 bit21;	
1	0	1	1	1	23	仲裁丢失在识别码的 bit22;	
1	1	0	0	0	24	仲裁丢失在识别码的 bit23;	
1	1	0	0	1	25	仲裁丢失在识别码的 bit24;	
1	1	0	1	0	26	仲裁丢失在识别码的 bit25;	
1	1	0	1	1	27	仲裁丢失在识别码的 bit26;	
1	1	1	0	0	28	仲裁丢失在识别码的 bit27;	
1	1	1	0	1	29	仲裁丢失在识别码的 bit28;	
1	1	1	1	0	30	仲裁丢失在识别码的 bit29;	
1	1	1	1	1	31	仲裁丢失在 RTR 位;	

注：

- 1: 位 11 为标准帧信息的 RTR 位。
- 2: 位 31~位 11 只使用于扩展帧信息。

22.4.15 CAN_ECC 错误代码捕捉寄存器

偏移地址: 0x30

复位值: 0x0000 0000

仅存在 PeliCAN 模式

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

	r	r	r	r	r	r	r	r
--	---	---	---	---	---	---	---	---

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 6	ERRC	错误代码 (Error code) 00: 位错 01: 格式错 10: 填充错 11: 其它错误
5	DIR	方向 (Direction) 1: RX; 接收时发生的错误 0: TX; 发送时发生的错误
4: 0	SEG	段 (Segment) 00010: ID.28—ID.21 00011: 帧开始 00100: SRTR 位 00101: IDE 位 00110: ID.20—ID.18 00111: ID.17—ID.13 01000: CRC 序列 01001: 保留位 0 01010: 数据区 01011: 数据长度代码 01100: RTR 位 01101: 保留位 1 01110: ID.4—ID.0 01111: ID.12—ID.5 10001: 主动错误标志 10010: 中止 10011: 显性位误差 10110: 被动错误标志 10111: 错误界定符 11000: CRC 界定符 11001: 应答通道 11010: 帧结束 11011: 应答界定符 11100: 过载标志 其他: 保留

注：位的设置反映了当前结构段的不同错误事件。

总线发生错误时会产生相应的错误中断（中断允许时）。同时，位流处理器的当前位置被捕捉送入错误代码捕捉寄存器。其内容直到用户通过软件读出时都是不变的。读出后，捕捉机制又被激活了。访问中断寄存器期间，中断寄存器中相应的中断标志位被清除。新的总线中断直到捕捉寄存器被读出一次才可能有

效。

22.4.16 CAN_EWLR 错误报警限制寄存器

偏移地址: 0x34

复位值: 0x0000 0060

仅存在 PeliCAN 模式

错误报警限制在这个寄存器中被定义。默认值（复位值）是 0x0060。复位模式中，此寄存器对 CPU 来说是可读/写的。工作模式中是只读的。

注：只有之前进入复位模式 **EWLR** 才有可能被改变。直到复位模式被再次取消后才有可能发生错误状态的改变（见状态寄存器）和由新的寄存器内容引起的错误报警中断。

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	EWL	可编程的错误限制报警（Programmable error warning limit） 只要一个错误计数器超过错误限制编程值，错误状态位被置位。

22.4.17 CAN_RXERR_RX 错误计数寄存器

偏移地址: 0x38

复位值: 0x0000 0000

仅存在 PeliCAN 模式

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	RXERR	RX 错误计数寄存器（RX error counter register）反应了接收错误计数器的当前值。硬件复位后寄存器被初始化为 0。工作模式中，对 CPU 来说是只读的。只有

Bit	Field	Description
		<p>在复位模式中才可以写访问此寄存器。如果发生总线关闭，RX 错误计数器就被初始化为 0。总线关闭期间，写这个寄存器是无效的。</p> <p>注意：只有之前进入复位模式，才有可能由 CPU 迫使 RX 错误计数器发生改变。直到复位模式被取消后，错误状态的改变（见状态寄存器）、错误报警和由新的寄存器内容引起的错误中断才可能有效。</p>

22.4.18 CAN_TXERR TX 错误计数寄存器

偏移地址: 0x3C

复位值: 0x0000 0000

仅存在 PeliCAN 模式

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	TXERR	<p>TX 错误计数寄存器 (TX error counter register) 反映了发送错误计数器的当前值。工作模式中，这个寄存器对 CPU 是只读的。复位模式中才可以写访问这个寄存器。硬件复位后，寄存器被初始化为 0。如果总线关闭，TX 错误计数器被初始化为 127 来计算总线定义的最长时间 (128 个总线空闲信号)。这段时间里读 TX 错误计数器将反映出总线关闭恢复的状态信息。如果总线关闭是激活的，写访问 TXERR 的 0 □ 254 单元会清除总线关闭标志，复位模式被清除后控制器会等待一个 11 位的连续隐性位 (总线空闲)。</p> <p>向 TXERR 写入 255 会初始化 CPU 驱动的总线关闭事件。只有之前进入复位模式，才有可能发生 CPU 引起的 TX 错误计数器内容的改变。直到复位模式被再次取消，错误或总线状态的改变（见状态寄存器）、错误报警和由新的寄存器内容引起的错误中断才有可能有效。离开复位模式后，就像总线错误引起的一样，给出新的 TX 计数器内容且总线关闭被同样的执行。这意味着重新进入复位模式，TX 错误计数器被初始化到 127，RX 计数器被清 0，所有的相关状态和中断寄存器位被置位。复位模式的清除将会执行协议规定的总线关闭恢复序列（等待 128 个总线空闲信号）。如果在总线关闭恢复 (TXERR > 0) 之前又进入复位模式，总线关闭保持有效且 TXERR 被锁定。</p>

22.4.19 CAN_SFF 发送帧信息寄存器

偏移地址: 0x40

复位值: 0x0000 00XX

仅存在 PeliCAN 模式

发送帧信息寄存器设置发送帧的类型与数据长度。仅在工作模式下这个寄存器可以被读/写访问，写时为发送寄存器，读时为接收寄存器，数据结构相同。

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7	FF	帧格式 (Frame format) 1: EFF; CAN 将发送/接收扩展帧格式 0: SFF; CAN 将发送/接收标准帧格式
6	RTR	远程帧请求 (Remote transmission request) 1: 远程; CAN 将发送/接收远程帧 0: 数据; CAN 将发送/接收数据帧
5: 4	Reserved	保留，必须保持复位值。
3: 0	DLC	数据区长度 (Data length code bit) 发送数据区长度 0 ~ 8 字节。

22.4.20 CAN_TXID0 发送识别码寄存器 0

偏移地址: 0x44

复位值: 0x0000 00XX

仅存在 PeliCAN 模式

发送识别码寄存器 0 设置帧的标识符。仅在工作模式下这个寄存器可以被读/写访问，写时为发送寄存器，读时为接收寄存器，数据结构相同。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	IDx	CAN 标识符 ID28 ~ ID21 (Identifier bit 28~21) 标准帧时，与 ID20 ~ ID18 组成 11 位标识符。

22.4.21 CAN_TXID1 发送识别码寄存器 1

偏移地址: 0x48

复位值: 0x0000 00XX

仅存在 PeliCAN 模式

发送识别码寄存器 1 设置帧的标识符。仅在工作模式下这个寄存器可以被读/写访问，写时为发送寄存器，读时为接收寄存器，数据结构相同。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
rw								rw							

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 5	IDx	CAN 标识符 ID20 ~ ID18 (Identifier bit 20~18)
4: 0	IDx	标准帧无意义 扩展帧 CAN 标识符 ID17 ~ ID13 (Identifier bit 17~13)

22.4.22 CAN_TXDATA0 发送数据寄存器 0

偏移地址: 0x4C

复位值: 0x0000 00XX

仅存在 PeliCAN 模式

发送数据寄存器 0，用于 CAN 数据发送存储。仅在工作模式下这个寄存器可以被读/写访问，写时为发送寄存器，读时为接收寄存器，数据结构相同。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DATA0							
rw								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	DATA0	CAN 发送数据 0 (CAN transmit data 0) 扩展帧 ID12 ~ 5

22.4.23 CAN_TXDATA1 发送数据寄存器 1

偏移地址： 0x50

复位值: 0x0000 00XX

仅存在 PeliCAN 模式

发送数据寄存器 1，用于 CAN 数据发送存储。仅在工作模式下这个寄存器可以被读/写访问，写时为发送寄存器，读时为接收寄存器，数据结构相同。

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	DATA1	CAN 发送数据 1 (CAN transmit data 1) 扩展帧 ID4 ~ 0, 低 3 位无意义

仅存在 PeliCAN 模式:

偏移地址: 0x54 ~ 0x70

复位值: 0x0000 00XX

以上偏移地址均为 CAN 数据寄存器，扩展帧时 CAN 发送数据 0 储存在 DATA2 余下数据依次类推。这些寄存器写时为发送寄存器，读时为接收寄存器，数据结构相同。

22.4.24 CAN CDR 时钟分频寄存器

偏移地址: 0x7C

复位值: 0x0000 0000

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7	MODE	CAN 模式 (CAN mode) 1: CAN 控制器工作于 PeliCAN 模式

Bit	Field	Description
		0: CAN 控制器工作于 BasicCAN 模式 该位只有在复位模式中是可写
6: 0	Reserved	保留, 必须保持复位值。

22.4.25 CAN_AFM0 滤波模式寄存器 0

偏移地址: 0x80

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								AFM7	AFM6	AFM5	AFM4	AFM3	AFM2	AFM1	Res.
rw								rw							

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 1	AFMx	验收滤波器模式 (Acceptance filter mode), x= 1□7 1: 选择单验收滤波器模式 (32 位长度) 0: 选择双验收滤波器模式 (每个为 16 位长度)
0	Reserved	保留, 必须保持复位值。

22.4.26 CAN_AFM1 滤波模式寄存器 1

偏移地址: 0x84

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								AFM15	AFM14	AFM13	AFM12	AFM11	AFM10	AFM9	AFM8
rw								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 0	AFMx	验收滤波器模式 (Acceptance filter mode), x= 8□15 1: 单; 选择单个验收滤波器 (32 位长度) 0: 双; 选择两个验收滤波器 (每个有 16 位有效)

22.4.27 CAN_AFM2 濾波模式寄存器 2

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.															

Bit	Field	Description
31: 4	Reserved	保留, 必须保持复位值。
3: 0	AFMx	验收滤波器模式 (Acceptance filter mode), x= 16~19 1: 单; 选择单个验收滤波器 (32 位长度) 0: 双; 选择两个验收滤波器 (每个有 16 位有效)

22.4.28 CAN_FGA0 滤波组使能寄存器 0

偏移地址: 0x8C

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
FGA7 FGA6 FGA5 FGA4 FGA3 FGA2 FGA1 Res.															

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值。
7: 1	FGAx	Group 滤波 (x=1~7) 有效位 1: 滤波有效 0: 滤波无效
0	Reserved	保留, 必须保持复位值。

22.4.29 CAN_FGA1 滤波组使能寄存器 1

偏移地址: 0x90

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
								FGA15	FGA14	FGA13	FGA12	FGA11	FGA10	FGA9	FGA8
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	FGAx	Group 濾波 ($x=8\sim15$) 有效位 1: 濾波有效 0: 濾波无效

22.4.30 CAN_FGA2 濾波组使能寄存器 2

偏移地址: 0x94

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
												FGA19	FGA18	FGA17	FGA16
												rw	rw	rw	rw

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	FGAx	Group 濾波 ($x=16\sim19$) 有效位 1: 濾波有效 0: 濾波无效

22.4.31 GROUPx_ACR 验收代码寄存器组 x (x = 1 , ..19)

偏移地址: 0x098 + (x - 1) * 0x20

复位值: 0x0000 00XX

BasicCAN 模式: GROUPx_ACR

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	AC	<p>(Acceptance code) 复位请求位被置高（当前）时，这个寄存器是可以访问（读/写）的。如果一条信息通过了验收滤波器的测试而且接收缓冲器有空间，那么描述符和数据将被分别顺次写入 RXFIFO。</p> <p>当信息被正确的接收完成就会：</p> <p>接收状态位置高（满）</p> <p>接收中断使能位置高（使能）接收中断置高（产生中断）</p> <p>验收代码位 (AC.7—AC.0) 和信息识别码的高 8 位 (ID.10—ID.3) 相等，且与验收屏蔽位 (AM.7—AM.0) 的相应位相或为 1。即如果满足下方程的描述，则被接收：</p> $((ID.10 - ID.3) = (AC.7 - AC.0)) \mid (AM.7 - AM.0) = 11111111$

PeliCAN 模式：每一组有四个验收代码寄存器分别是 GROUPx_ACR0, GROUPx_ACR1, GROUPx_ACR2, GROUPx_ACR3

GROUPx_ACR0: 偏移地址: 0x098+ (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_ACR1: 偏移地址: 0x09C + (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_ACR2: 偏移地址: 0x0A0 + (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_ACR3: 偏移地址: 0x0A4 + (x - 1) * 0x20 复位值: 0x0000 00XX

22.4.32 GROUPx_AMR 验收屏蔽寄存器组 x (x = 1 , ..19)

偏移地址: 0x0A8 + (x - 1) * 0x20

复位值: 0x0000 00XX

BasicCAN 模式: GROUPx_AMR

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值。
7: 0	AM	<p>(Acceptance mask) 如果复位请求位置高（当前）这个寄存器可以被访问（读/写）。</p> <p>验收屏蔽寄存器定义验收代码寄存器的相应位对验收滤波器是“相关的”或“无影响的”（即可为任意值）。</p>

PeliCAN 模式：每一组有四个验收屏蔽寄存器分别是 GROUPx_AMR0, GROUPx_AMR1, GROUPx_AMR2, GROUPx_AMR3

GROUPx_AMR0: 偏移地址: 0x0A8 + (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_AMR1: 偏移地址: 0x0AC+ (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_AMR2: 偏移地址: 0x0B0 + (x - 1) * 0x20 复位值: 0x0000 00XX

GROUPx_AMR3: 偏移地址: 0x0B4 + (x - 1) * 0x20 复位值: 0x0000 00XX

23 USB_FS 全速 USB 控制器

23.1 简介

USB_FS 控制器支持配置为 OTG 双角色设备 (dual-role device)，也可仅配置为 USB device 设备或 USB_HOST 嵌入式主机。USB_FS 控制器完全符合《On-The-Go Supplement to the USB 2.0 Specification》。

表 23-1 术语表

英文缩写	说明
FS	全速
LS	低速
USB	通用串行总线
OTG	On-The-Go
PHY	物理层
HNP	主机协商协议
SRP	会话请求协议
BDT	缓冲区描述表
BD	缓冲区描述符
SIE	串行接口引擎

其中，OTG 相关描述，仅在支持 OTG 的产品中适用。

23.2 主要特征

- 全速设备控制器兼容 USB 1.1 和 2.0 规范
- 16 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
- 支持 USB_HOST 嵌入式主机模式
- 支持 dual-role device OTG 双角色设备模式
- 支持通过 ID 线辨认 A-B 设备
- 支持主机协商协议 (HNP) 及会话请求协议 (SRP)
- 在 OTG 应用中，允许主机关闭 VBUS 以节省耗电

23.3 功能描述

23.3.1 功能框图

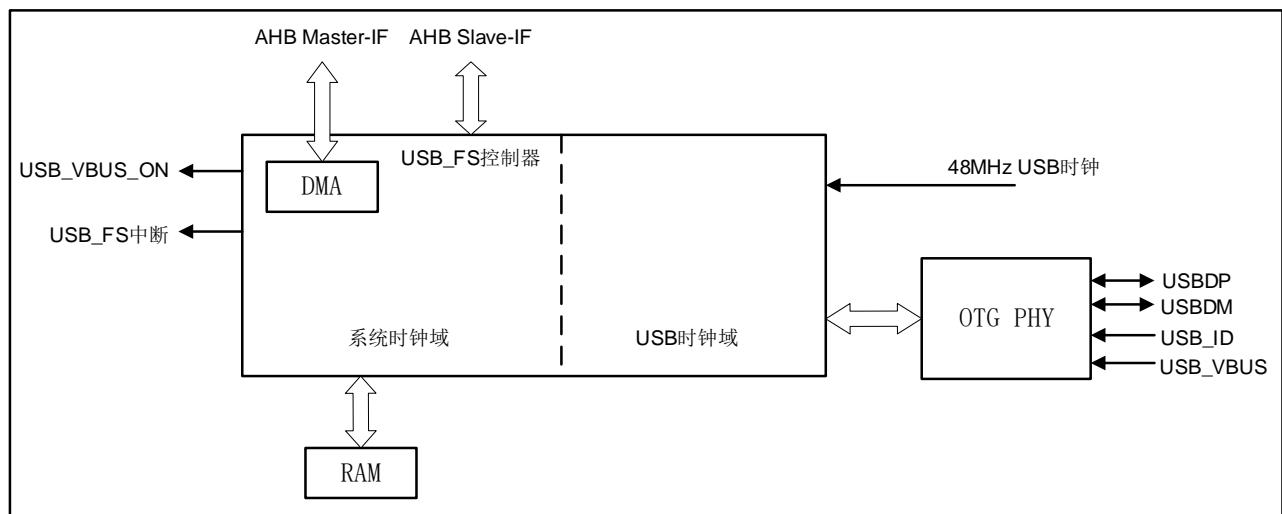


图 23-1 USB_FS 功能框图

23.3.2 信号描述

表 23-2 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
USB_ID	输入	USB A-B 器件识别信号
USB_VBUS	输入	USB VBUS 信号
USB_VBUS_ON	输出	外部电源芯片使能信号

23.3.3 USB_FS 典型硬件配置

下图分别表示仅主机、仅从设备以及 OTG 硬件连接示意图。

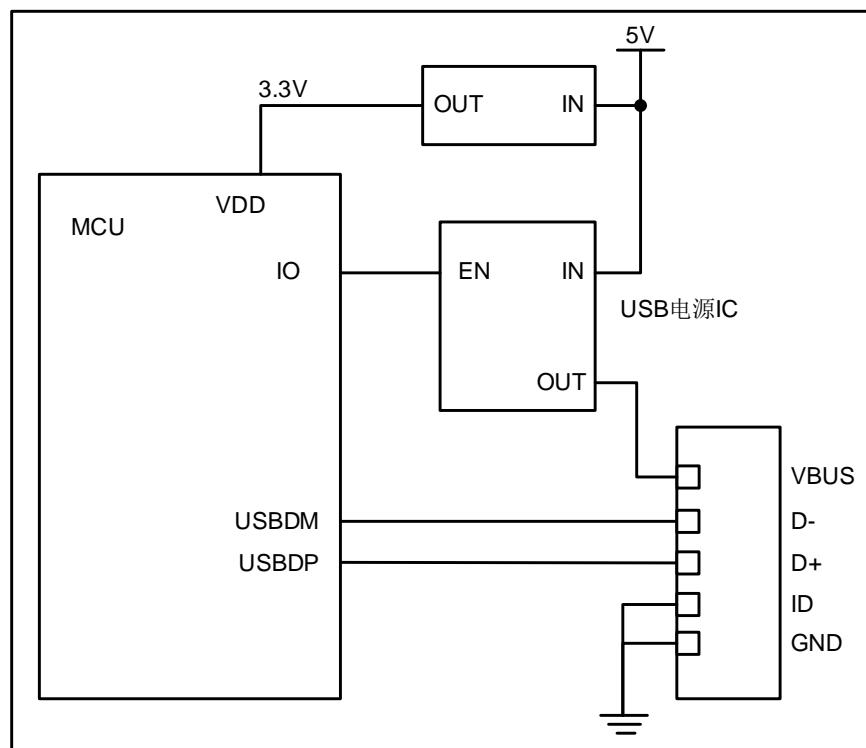


图 23-2 仅主机硬件连接示意图

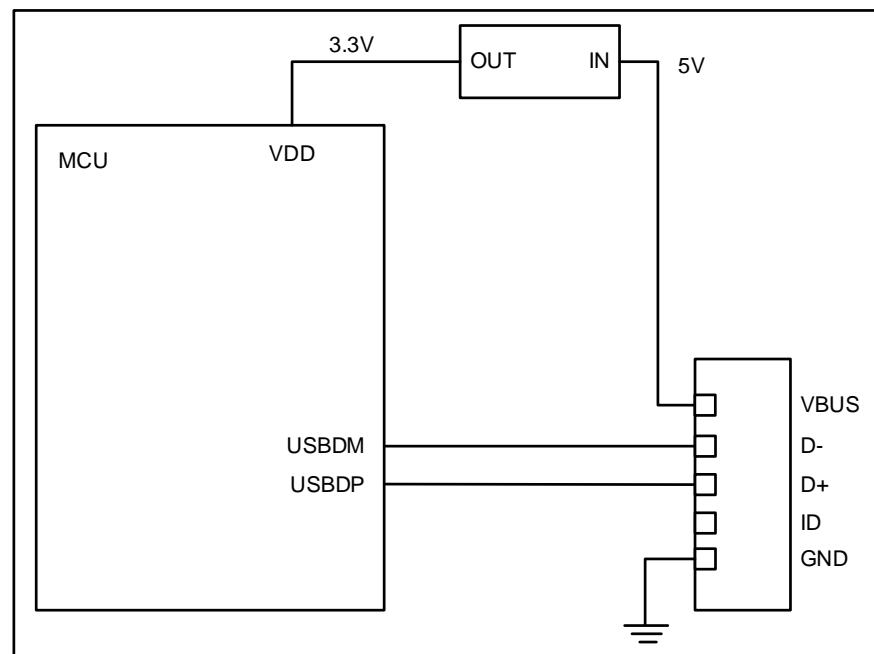


图 23-3 仅从设备硬件连接示意图

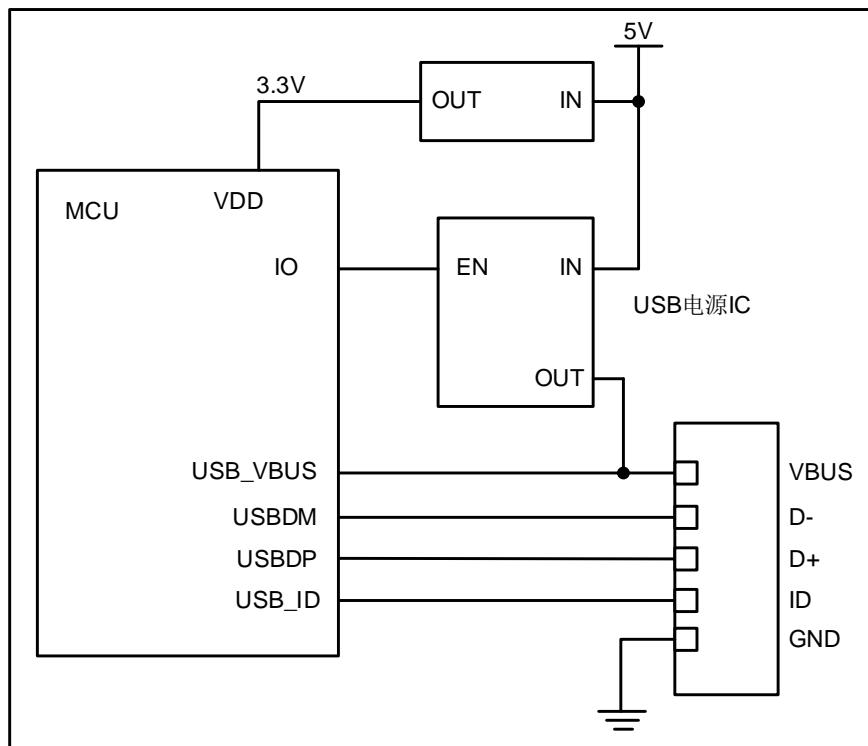


图 23-4 OTG 硬件连接示意图

23.3.4 USB_FS 软件编程接口

23.3.4.1 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**。

23.3.4.2 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。

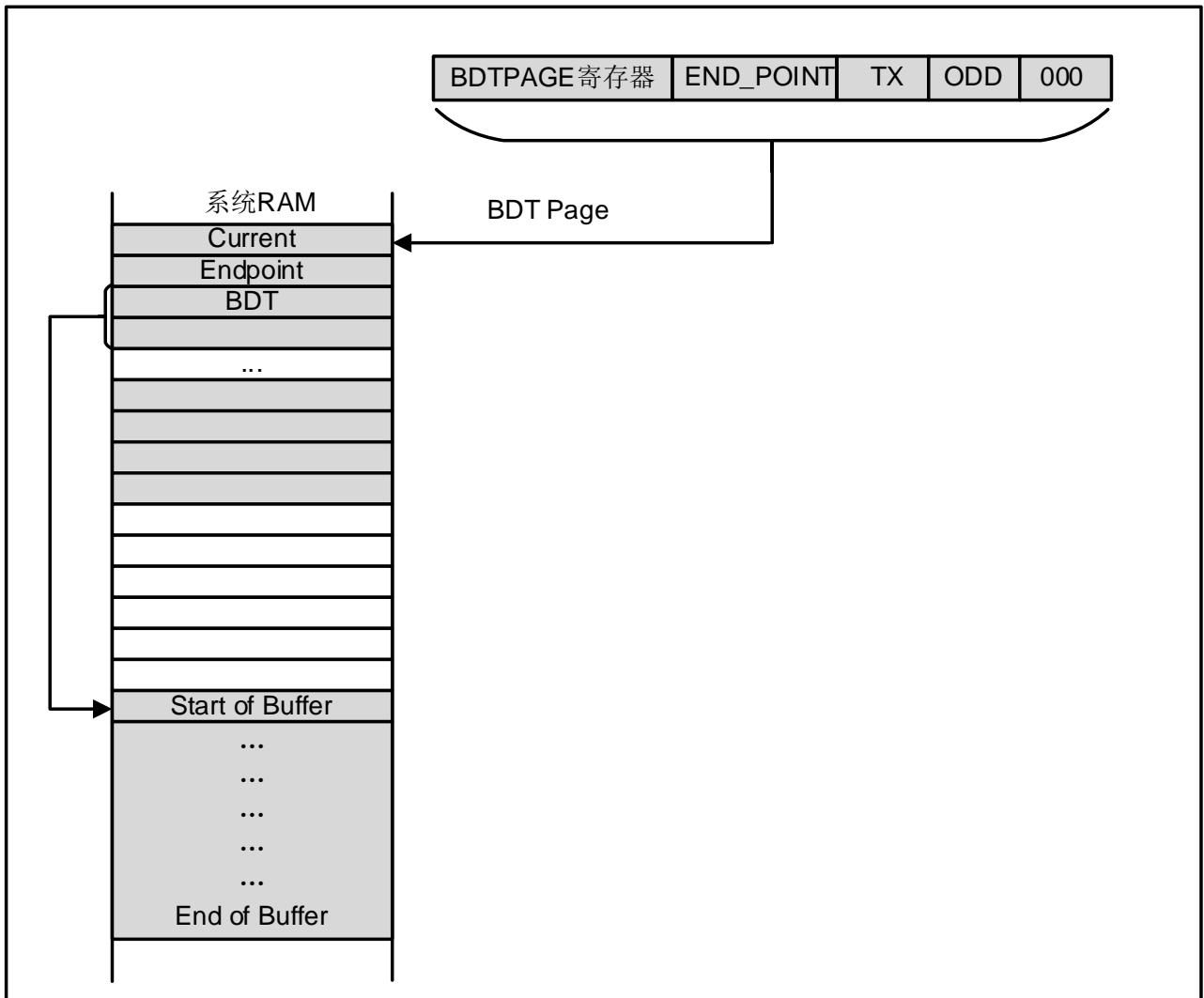


图 23-5 缓冲区描述表

23.3.4.3 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 条目部被索引到 **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 23-3 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 23-4 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 23-5 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。当数据需要从一个单一位置 (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。

23.3.4.4 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。

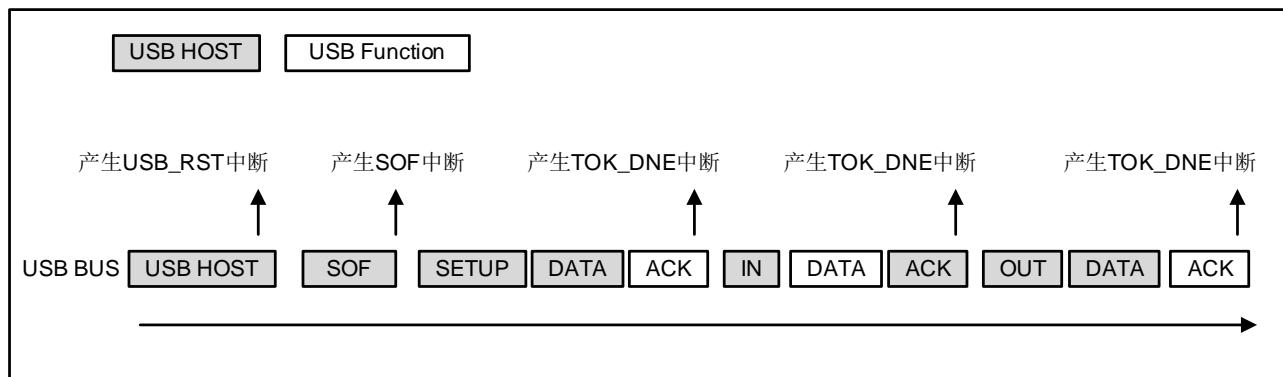


图 23-6 USB 令牌传输示意图

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。主机模式或者设备模式，ERR STAT 寄存器的 DMAERR 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。在设备模式下，BDT 不会被写回，也不会触发 TOK_DNE 中断。在主机模式下，产生 TOK_DNE 中断，并且 BDT 的 OK PID 字段将被设置为‘1111’，以指示 DMA 延迟错误，另外主机模式软件可以决定重试或者跳转到其它流程。

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 DMAERR 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。

23.3.5 OTG 与 Host 模式操作方式

USB_FS 支持主机（Host）模式和 OTG 操作（HNP 和 SRP）。

Host 模式旨在用于手持便携式设备，以便于连接简单的 HID 类设备，如打印机和键盘。它不是要执行 PC 主板上的完整 OHCI 或 UHCI 兼容主机控制器的功能。主机模式支持 Control、Bulk、Interrupt 和 Isochronous 四种传输。

设置 CTL 寄存器中的 HOST_MODE_EN 位可启用主机模式。USB_FS 内核只能作为设备或主机模式运行。它不能同时在两种模式下运行。当启用 HOST_MODE 时，只使用端点 0，并且禁用其他端点。

23.3.6 Host 模式操作示例

以下部分说明使用 USB_FS 内核执行 USB 主机功能所需的步骤。

23.3.6.1 启用主机模式并发现连接的设备

1. 使能主机模式 (CTL.HOST_MODE_EN = 1)
2. 下拉电阻使能, 上拉电阻禁用, 并设置 CTL.USB_EN = 1
3. 等待设备连接
- 使能 ATTACH 中断 (INTENB.ATTACH = 1)
- 等待设备连接等待 ATTACH 中断 (INTSTAT.ATTACH)
4. 获取连接设备速度
- 检查控制寄存器中的 JSTATE 和 SE0 位的状态。如果 JSTATE 位为 0, 则连接设备为低速设备。低速设备需设置 ADDR.LS_EN = 1, EPCTL0.HOST_WO_HUB = 1
5. 复位设备
- 设定 CTL.RESET = 1
6. 使能 USB
- 设定 CTL.USB_EN = 1。此时控制器开始发出 SOF, 使连接的设备不会进入挂起状态
7. 枚举以及后续操作

23.3.6.2 与 USB 设备进行 Control 传输

设备连接后, 通过以下步骤与设备进行控制传输。

1. 设置端点控制寄存器用于双向控制传输
- EPCTL0 [4:0] = 0x0d
2. 将需要用到的 Device Framework 相关指令存入内存缓冲区中
3. GET_DESCRIPTOR 相关设定
- 初始化当前 TX EP0 BDT (偶或奇), 用于 GET_DESCRIPTOR 命令。将 BD 设置为 0x00080080 (BC=8, OWN=1)。将 BD 的 ADDR 地址字段设置为 GET_DESCRIPTOR 命令缓冲区的起始地址
4. SET_ADDRESS 相关设定
- 在地址寄存器 (ADDR [6:0]) 中设置 USB 设备地址
5. 执行控制传输的 Setup Stage
- 设置 TOKEN 寄存器为 EP0 的 SETUP 令牌包 (TOKEN = 0xD0)
- SETUP 令牌传输完成后, 设备握手将在 BDT PID 字段中返回
- BDT 写入完成 (INTSTAT.TOK_DNE) 中断将被置位
6. 执行控制传输的 Data Stage (有些令牌不存在 Data Stage)
- 在 RAM 中设置用于要传送的数据缓冲区

- 初始化当前 TX EP0 BDT(偶或奇)传输数据。将 BD 设置为 0x004000C0 (BC=64, OWN=1, DATA0/1 = 1)。将 BD 的 ADDR 地址字段设置为数据缓冲区的起始地址。
 - 设置 TOKEN 寄存器为 EP0 的 IN 令牌或者 OUT 令牌，启动 IN 或者 OUT 传输。如对于 GET_DESCRIPTOR 的 Data Stage，设置 TOKEN 寄存器为 EP0 的 IN 令牌
 - 当传输完成时，BDT 被写入并且完成令牌 (INTSTAT.TOK_DNE) 中断置位
 - 以上步骤完成了一个数据包的传输，如果有多个数据包，遵从 USB 规定的 Data Toggle Synchronization 机制，完成 Data Stage 的操作
7. 执行控制传输的 Status Stage
- 初始化当前 TX EP0 BDT(偶或奇)传送状态数据。将 BD 设置为 0x00000080 (BC=0, OWN=1, DATA0/1 = 0)。将 BD ADDR 地址字段设置为数据缓冲区的起始地址
 - 设置 TOKEN 寄存器为 EP0 的 IN 令牌或者 OUT 令牌，启动 IN 或者 OUT 传输，接收或者发送零长度数据包。如对于 GET_DESCRIPTOR 的 Status Stage，设置 TOKEN 寄存器为 EP0 的 OUT 令牌
 - 当传输完成时，BDT 被写入并且完成令牌 (INTSTAT.TOK_DNE) 中断置位

23.3.6.3 与 USB 设备进行 Bulk OUT 传输

设备连接并完成配置后，通过以下步骤与设备进行 BULK OUT 传输。

1. 设置端点控制寄存器用于双向传输 EP CTL0 = 0x1d
 2. 向 ADDR 寄存器写入设备地址
 3. 设置偶数 TX EP0 BDT 可传送多达 64 个字节 (FS 操作时)
 4. TOKEN 寄存器设置为所需的端点的 OUT 令牌方式
 5. 设置奇数 TX EP0 BDT 可传输多达 64 个字节 (FS 操作时)
 6. TOKEN 寄存器设置为所需的端点的 OUT 令牌方式
- 步骤 3~6 实现了双缓冲结构，提升了吞吐量。
7. 等待 TOK_DNE 中断
- 置位则表示其中一个 BDT 已经被释放给处理器，并且传输已经完成。
 - RETRY_DIS=0 时，设备返回 NAK 时，USB_FS 将继续无限期地在无处理器干预的情况下重试传输
 - RETRY_DIS=1 时，重试传输功能被禁止，则 BDT PID 字段将返回握手 PID
 - 对于 STALL 相应，需要将待处理的数据包从队列中移除，并清除目标设备中的错误条件。
 - 如果发生复位中断，表明设备失去连接状态

重复步骤 3~7 可以实现多个数据的连续传输。

23.3.7 OTG 操作

处理器通过软件来实现主机协商协议（HNP）和会话请求协议（SRP）。

23.3.7.1 OTG 双角色 A-device 操作方式

A-Decive 为将标准 A 或 Mini-A 插头插入其插座的设备。A-device 设备为 VBUS 供电；会话开始时为主机角色，在某些条件下，A-device 设备将放弃主机对 B-device 设备的控制。

双重角色 A-device 将按照以下流程图和状态描述表进行操作。

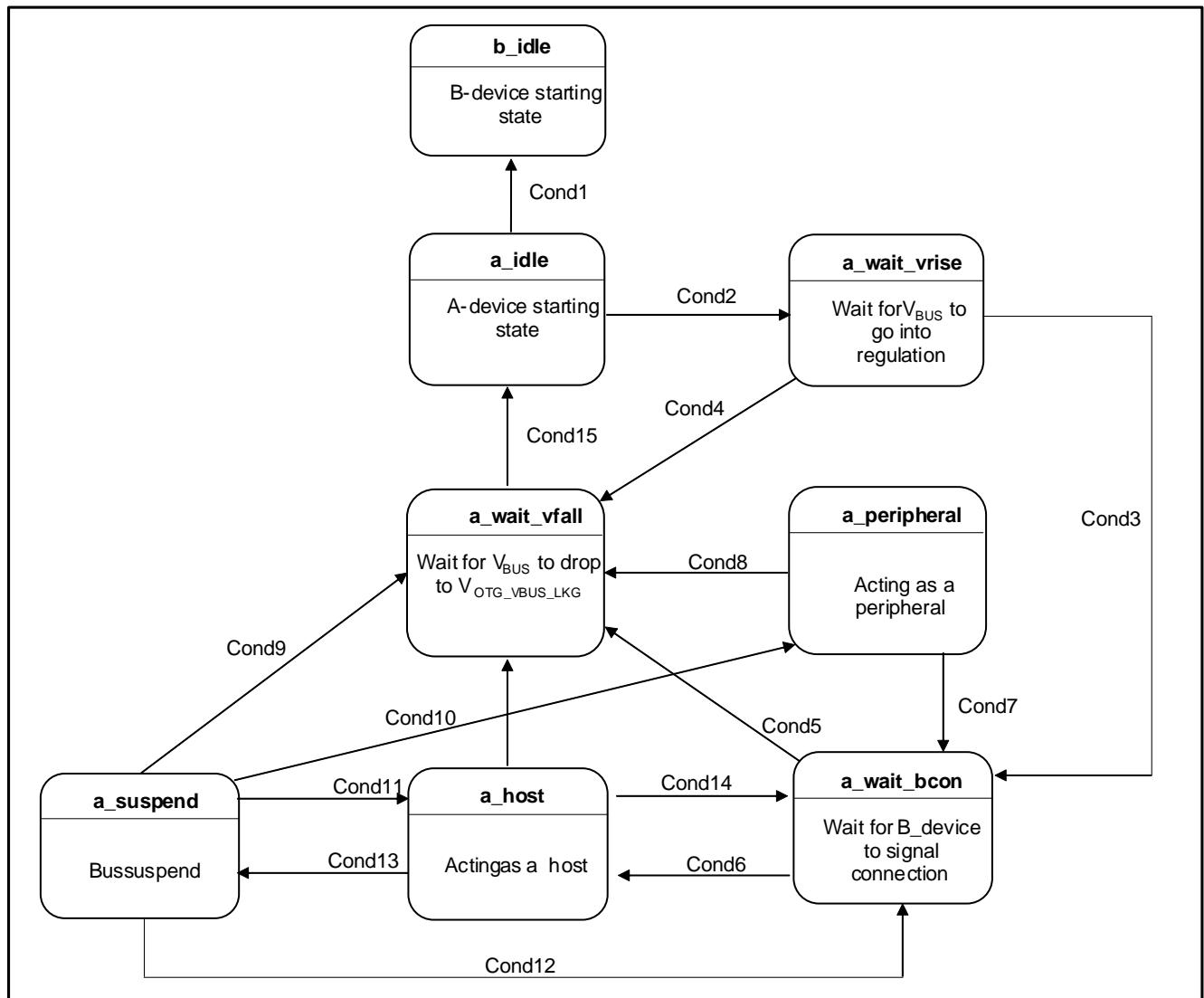


图 23-7 OTG A-device 操作方式

表 23-6 OTG A-device 操作状态描述表

条件	描述
Cond1	ID 线中断。电缆未插入或已连接 B-device。
Cond2	如果 A-device 应用程序想要使用总线，或者 B-device 正在执行一个 SRP，如 A_SESS_VLD 中断或插入或端口状态改变中断响应，检查数据线脉冲 5-10 毫秒。
Cond3	如果 A_VBUS_VLD 中断。
Cond4	如果 ID 中断响应或者如果 100 毫秒之后 A_VBUS_VLD 错误. 电缆已改变或 A-device 设备无法支持 B-device 所需的电流。转到 a_wait_vfall 状态后，关闭 VBUS。
Cond5	200 毫秒后没有连接或 ID 中断。转到 a_wait_vfall 状态后，关闭 VBUS。
Cond6	A_VBUS_VLD 中断且 B-device 连接上。
Cond7	如果总线空闲 3~200 毫秒。
Cond8	如果 ID 中断或者如果 A_VBUS_VLD 中断。转到 a_wait_vfall 状态后，关闭 VBUS。
Cond9	如果 ID 中断，或者如果 150 毫秒 B-device 断开超时（该超时值可能更长），或者如果 A_VBUS_VLD 中断。转到 a_wait_vfall 状态后，关闭 VBUS。
Cond10	如果启用 HNP，并且 B-device 在 150 毫秒内断开，则 B-device 正在成为主机。
Cond11	如果 A-device 想要开启另一个会话。

23.3.7.2 OTG 双角色 B-device 设备操作方式

B-Device 为标准 B 或 Mini-B 插头插入其插座的设备。B-device 是会话开始时的外围设备。如果设备是双重角色，则可以从 A-device 授予主机角色。

双角色 B-Device 将按照以下流程图和状态描述表所示进行操作。

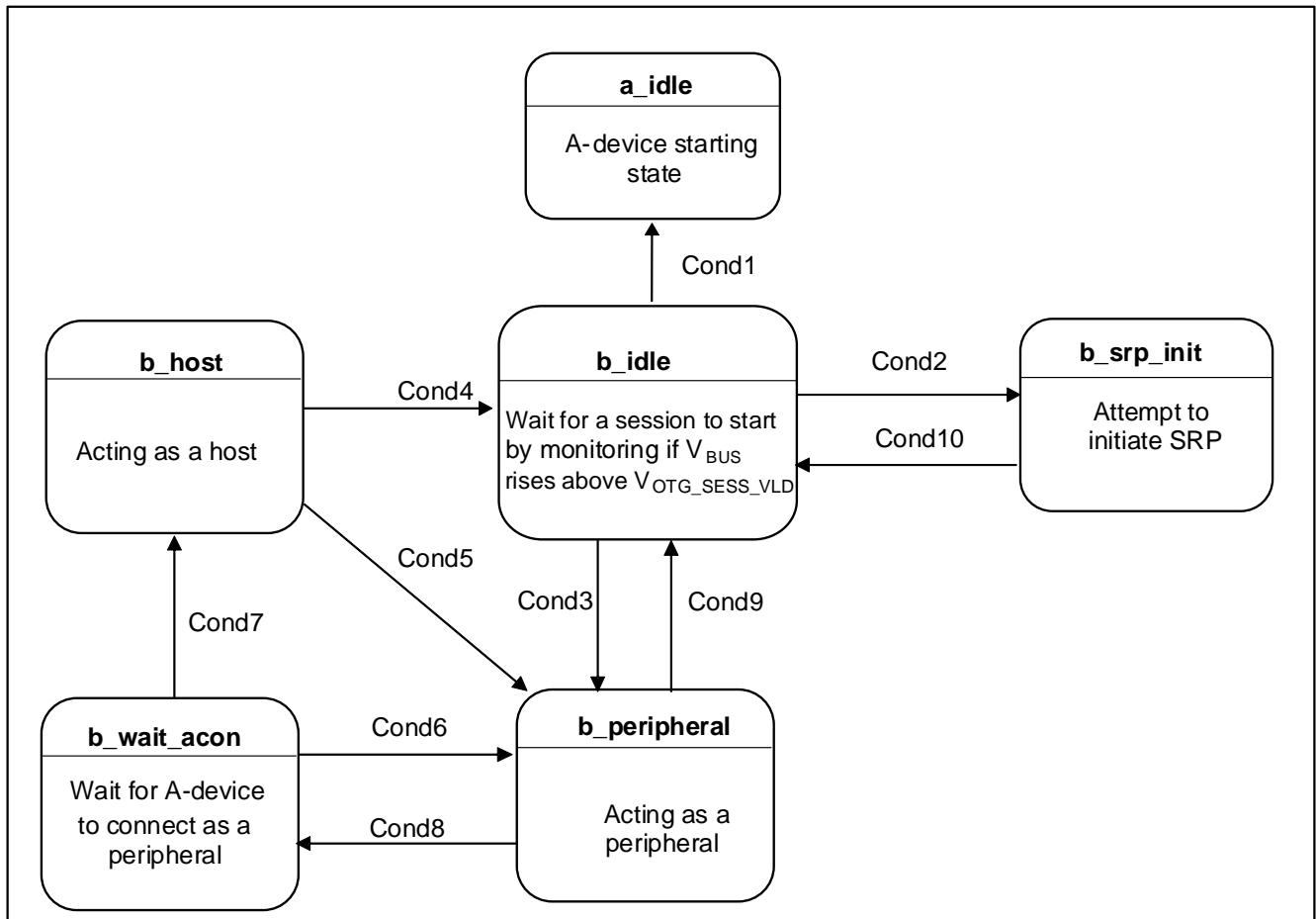


图 23-8 OTG B-device 操作方式

表 23-7 OTG B-device 操作状态描述表

条件	描述
Cond1	如果 ID 中断。“A”型电缆已经插入，设备现在应该作为“A”型设备。
Cond2	如果 B-device 应用程序想要总线控制权且总线处于空闲状态 2 ms 并且 B_SESS_END 位置位，则 B-device 可以执行 SRP。 在 b_srp_init 状态，脉冲 CHRG_VBUS, DP_HIGH 产生 5~10 毫秒的脉冲。
Cond3	如果 B_SESS_VLD 中断。A-device 已打开 VBUS 并开始会话。 在 b_peripheral 状态，打开 DP_HIGH。
Cond4	如果 ID 中断或 B_SESS_VLD 中断。
Cond5	如果 B-device 应用程序已完成或 A-device 断开连接。
Cond6	超过 3.125 毫秒或发生 resume。
Cond7	如果启用 HNP 并且总线挂起，如果 B-device 想要总线，则 B-device 可以成为主机。
Cond8	如果启用 HNP 并且总线挂起，如果 B-device 想要总线，则 B-device 可以成为主机。

23.3.8 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

- ATTACH 中断
- RESUME 中断
- SLEEP 中断
- TOK_DNE 中断
- ERROR 中断
- USB_RST 中断
- OTG 中断
 - ◆ ID 信号变化中断
 - ◆ 1 毫秒定时器中断
 - ◆ 线路状态变化中断
 - ◆ 会话有效中断
 - ◆ B-devcie 会话结束中断
 - ◆ A-device VBUS 变化中断

23.4 寄存器

23.4.1 寄存器总览

表 23-8 USB 寄存器概览

Offset	Acronym	Register Name	Reset
0x10	USB_FS_OTGISTAT	OTG Interrupt Status Register	0x00E8
0x14	USB_FS_OTGICTRL	OTG Interrupt Control Register	0x0000
0x18	USB_FS_OTGSTAT	OTG Status Register	0x00A8
0x1C	USB_FS_OTGCTRL	OTG Control Register	0x0000
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
0xA8	USB_FS_TOKEN	Token Register	0x0000
0xAC	USB_FS_SOFTHL	SOF Threshold Register	0x0000
0xB0	USB_FS_BDTPAGE2	BDT Page Register 2	0x0000

Offset	Acronym	Register Name	Reset
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
0xE0	USB_FS_EPCTL8	Endpoint Control Register 8	0x0000
0xE4	USB_FS_EPCTL9	Endpoint Control Register 9	0x0000
0xE8	USB_FS_EPCTL10	Endpoint Control Register 10	0x0000
0xEC	USB_FS_EPCTL11	Endpoint Control Register 11	0x0000
0xF0	USB_FS_EPCTL12	Endpoint Control Register 12	0x0000
0xF4	USB_FS_EPCTL13	Endpoint Control Register 13	0x0000
0xF8	USB_FS_EPCTL14	Endpoint Control Register 14	0x0000
0xFC	USB_FS_EPCTL15	Endpoint Control Register 15	0x0000

23.4.2 USB_FS_OTGISTAT OTG 中断状态寄存器

偏移地址: 0x10

复位值: 0x0000

OTG 中断状态寄存器记录 ID PIN 和 VBUS PIN 的信号变化。软件读取该寄存器以确定导致中断的事件。仅置位自上次软件读取后已更改的位。写‘1’清除相应的中断标志。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ID_CHG	1_MSEC	LINE_STATE_CHG	Res.	SESS_VLD_CHG	B_SESS_END_CHG	Res.	A_VBUS_VLD_CHG				
				rcw1	rcw1	rcw1			rcw1		rcw1			rcw1	

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	ID_CHG	当检测到 USB ID PIN 信号发生变化时, 该位置位。
6	1_MSEC	该位在 1 毫秒定时器到期时置位。该位保持置位直到被软件清零。中断必须每毫秒服务一次, 以避免丢失 1 毫秒的计数值。
5	LINE_STATE_CHG	当 USB 线路状态 (CTL 寄存器的 SEO 和 JSTATE 位) 稳定, 在 1 毫秒内无变化, 并且线路状态的值与线路状态最后一次稳定时的值不同时, 该中断被设置。当 SEO 和 J, SEO 和 K 以及 J 和 K 之间产生转换, 它将被置位。当 SEO 位为 1 时, JSTATE 的改变不会引起中断。该中断可用于检测复位, 恢复。
4	Reserved	保留, 必须保持复位值。
2	SESS_VLD_CHG	当检测到 VBUS 变化时, 该位置位, 表示会话有效或会话不再有效。

Bit	Field	Description
2	B_SESS_END_CHG	在 B-device 上检测到 VBUS 发生变化时，该位置位。
1	Reserved	保留，必须保持复位值。
0	A_VBUS_VLD_CHG	检测到 VBUS 电压高于“A” VBUS 有效阈值时，该位置‘1’。

23.4.3 USB_FS_OTGICTRL OTG 中断控制寄存器

偏移地址: 0x14

复位值: 0x0000

OTG 中断控制寄存器使能 OTG 中断状态寄存器中定义的相应中断状态位。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ID_EN	1_MSEC_EN	LINE_STATE_EN	Res.	SESS_VLD_EN	B_SESS_END_EN	Res.	Res.	Res.	Res.	Res.	A_VBUS_VLD_EN
				rw	rw	rw		rw	rw		rw		rw		rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	ID_EN	0: 禁止 ID 信号中断。 1: 使能 ID 信号中断。
6	1_MSEC_EN	0: 禁止 1 毫秒定时器中断。 1: 使能 1 毫秒定时器中断。
5	LINE STATE_EN	0: 禁止线路状态变化中断。 1: 使能线路状态变化中断。
4	Reserved	保留，必须保持复位值。
3	SESS_VLD_EN	0: 禁止会话有效中断。 1: 使能会话有效中断。
2	B_SESS_END_EN	0: 禁止 B-device 会话结束中断。 1: 使能 B-device 会话结束中断。
1	Reserved	保留，必须保持复位值。
0	A_VBUS_VLD_EN	0: 禁止“A” VBUS 有效中断。 1: 使能“A” VBUS 有效中断。

23.4.4 USB_FS_OTGSTAT OTG 状态寄存器

偏移地址: 0x18

复位值: 0x0000

OTG 状态寄存器显示 ID PIN 和 VBUS PIN 与比较器的实际输出值

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ID	Res.	LINESTATE_STABLE	Res.	SESS_VLD	B_SESS_END	Res.	Res.	Res.	Res.	Res.	A_VBUS_VLD

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	ID	指示 USB 连接器上的 ID 引脚的当前状态。 0: 表示“A”型电缆已插入 USB 连接器 1: 表示未连接电缆或“B”型电缆已插入 USB 连接器
6	Reserved	保留，必须保持复位值。
5	LINESTATE_STABLE	当线路状态（CTL 寄存器中的 JSTATE 和 SEO）在前 1 毫秒内稳定时，该位置‘1’。该位用在检测连接，断开和恢复信号时提供线路状态的硬件去抖动。当使用该位时，首先读取 CTL 寄存器中的 SEO 和 JSTATE 的状态。然后读出此位。如果该位为 1，那么 CTL 寄存器的 JSTATE 和 SEO 位的值在 1 毫秒内不变，可以被认为是去抖动的。
4	Reserved	保留，必须保持复位值。
3	SESS_VLD	当 VBUS 高于“B”会话的有效阈值时，该位将被置位。
2	B_SESS_END	当 VBUS 电压低于“B”会话结束阈值时，该位将被置位。
1	Reserved	保留，必须保持复位值。
0	A_VBUS_VLD	当 VBUS 电压高于“A”VBUS 有效阈值时，该位将被置位。

23.4.5 USB_FS_OTGCTRL OTG 控制寄存器

偏移地址: 0x01C

复位值: 0x0000

OTG 控制寄存器在 OTG_EN 置位后，控制 VBUS 电压和数据线电阻的上拉及下拉。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
								Res.	DP_HIGH	DM_HIGH	DP_LOW	DM_LOW	VBUS_ON	OTG_EN	VBUS_CHG	VBUS_DSCHG
									rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	DP_HIGH	置位时，启用 D+ 数据线的上拉电阻。
6	DM_HIGH	置位时，启用 D- 数据线的上拉电阻。
5	DP_LOW	置位时，启用 D+ 数据线的下拉电阻。
4	DM_LOW	置位时，启用 D- 数据线的下拉电阻。
3	VBUS_ON	置位时，打开 VBUS 电源信号。
2	OTG_EN	置位时，这个寄存器中的 D+ 和 D- 上拉和下拉控制位有效。 当 OTG_EN = 0 且 USB_EN.HOST=0 时，为设备模式，D+ 上拉被使能。 当 OTG_EN = 0 且 USB_EN.HOST=1 时，为主机模式，D+ 和 D- 的下拉被使能。
1	VBUS_CHG	置位时，将通过电阻为 VBUS 充电。
0	VBUS_DSCHG	置位时，这将通过电阻为 VBUS 放电。

23.4.6 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	ATTACH	RESUME	SLEEP	TOK_DNE	SOF_TOK	ERROR	USB_RST
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7	STALL	STALL 中断用于从机模式和主机模式。 在从机模式下，SIE 发送 STALL 握手包时置位。 在主机模式下，SIE 检测到 STALL 握手响应。
6	ATTACH	如果 USB_FS 检测到 USB 外设的连接，则该位置位。该信号仅在 HOST_MODE_EN =1 时有效。 此中断对于表示外设已连接并且必须进行配置。如果在 USB 数据线 2.5us 没有状态转换并且当前总线状态不是 SE0，则 ATTACH 中断被置位。
5	RESUME	主机模式用于检测设备的 RESUME 信号。 设备模式用于发送 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	设备模式下，如果 USB_FS 收到 SOF 令牌，则此位置位。 在主机模式下，当达到 SOF 阈值时，该位被置位，以便软件可以为下一个 SOF 做准备
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。

23.4.7 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	ATTACH	RESUME	SLEEP	TOK_DNE	SOF_TOK	ERROR	USB_RST
								rw	rw	rw	rw	rw	rw	rw	rw

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	ATTACH	设置该位为 1 使能 ATTACH 中断。
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。

23.4.8 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, 这会引起接收数据溢出。 如果来自主机或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。

Bit	Field	Description
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	这个错误中断有两个功能。 当 USB_FS 工作在设备模式 (HOST_MODE_EN = 0) 时，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。 当 USB_FS 工作在主机模式 (HOST_MODE_EN = 1) 时，该中断将检测帧结束 (EOF) 错误情况。当 USB_FS 正在发送或接收数据并且 SOF 计数器已经达到零时，就会发生这种情况。这个中断在开发 USB 数据包调度软件时很有用，以确保没有 USB 事务跨越下一帧的开始。
0	PID_ERR	PID 自校验错误。

23.4.9 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	Reserved	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。

23.4.10 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。

23.4.11 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	TxSuspendTokenBusy	RESET	HOST_MODE_EN	RESUME	ODD_RST	USB_EN
								r	r	rw	rw	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。

Bit	Field	Description
5	TxdSuspendTokenBusy	当 USB_FS 为从机模式时为 TxdSuspend , 当 USB_FS 处于主机模式时为 TokenBusy 。 TxdSuspend 位通知处理器 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。 TokenBusy 位通知主机处理器 USB_FS 正忙于执行 USB 令牌, 而不应将更多令牌命令写入令牌寄存器。在将任何令牌写入令牌寄存器之前, 软件应检查此位, 以确保令牌命令不会丢失。
4	RESET	设置该位将使 USB_FS 产生 USB 复位信号。这将允许 USB_FS 重置 USB 设备。该控制信号仅在主机模式有效 (即, HOST_MDOE_EN = 1)。软件必须将 RESET 设置为 1 达所需的时间, 然后将其清除为 0 以结束复位信号。具体请 USB 参考协议规范。
3	HOST_MODE_EN	设置该位为 1 将使 USB_FS 在主机模式下工作。
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	设置此位将重置所有的 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。 当主机模式使能时, 清除该位将导致 SIE 停止发送 SOF 令牌。

23.4.12 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** 在外设模式下解码的唯一 **USB** 地址 (**HOST_MODE_EN = 0**)。

在主机模式 (**HOST_MODE_EN = 1**) 下运行时, **USB_FS** 将使用 **TOKEN** 数据包发送此地址。这将使 **USB_FS** 能够寻址任何 **USB** 设备。

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

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	LS_EN	使能此位将告知 USB_FS 写入令牌寄存器的下一个令牌命令必须低速执行。这将使 USB_FS 能够执行低速数据传输所需的要求。
6: 0	ADDR	USB_FS 在设备模式下解码的 USB 地址, 或者在主机模式下传输的 USB 地址。

23.4.13 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。

23.4.14 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。

23.4.15 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。

23.4.16 USB_FS_TOKEN 令牌寄存器

偏移地址: 0xA8

复位值: 0x0000

令牌寄存器用于在主机模式下执行 USB 事务。当主处理器希望对外设执行 USB 事务时，它会将 TOKEN 类型和端点写入该寄存器。一旦写入该寄存器，USB_FS 就会开始指定的 USB 事务。

在执行对令牌寄存器的写操作之前，主机处理器应始终检查控制寄存器中的 TOKEN_BUSY 位，这将确保令牌命令在执行之前不会被覆盖。地址寄存器和端点控制寄存器 0 也在执行令牌命令时使用，因此也必须在令牌寄存器写入之前写入。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 4	TOKEN_PID	设置执行的令牌类型，有效令牌是： 0001: OUT 1001: IN 1101: SETUP
3: 0	TOKEN_ENDPT	设置和设备进行通信端点。

23.4.17 USB_FS_SOFTHLD SOF 阈值寄存器

偏移地址: 0xAC

复位值: 0x0000

SOF 阈值寄存器仅在 HOST_MODE 中使用。

当 HOST_MODE 使能时，内部 14 位 SOF 计数器计算 SOF 帧之间的间隔。

SOF 必须每 1ms 发送一次，因此 SOF 计数器加载值为 12000。当 SOF 计数器达到零时，发送帧起始 (SOF) 令牌。

SOF 阈值寄存器用于在 SOF 停止启动令牌包事务之前编程 USB 字节时间的数量。

通过编程该寄存器保证在 SOF 计时器计数到零时不会主动传输其他数据包。

当 SOF 计数器达到阈值时，在发送 SOF 之前不再发送令牌。

编程到阈值寄存器中的值必须保留足够的时间以确保最坏情况下的事务的完成。

SOF 阈值的典型值为：64 字节数据包为 74；32 字节包为 42；16 字节包为 26；8 字节包为 18。

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

Bit	Field	Description
15: 8	Reserved	保留，必须保持复位值。
7: 0	CNT	设置 SOF 计数阈值。

23.4.18 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。

23.4.19 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。

23.4.20 USB_FS_EPCTL0~15 端点控制寄存器 0~15

偏移地址: 0xC0~0xFC

复位值: 0x0000

在收到 USB_RST 中断后，微处理器应将 ENDPT0 设置为 0x0D。

在主机模式下，ENDPT0 用于确定主机传输的握手，重试和低速特性。对于主机模式控制，批量和中断传输，EP_HSHK 位应设置为 1，对于同步传输，应将其设置为 0。在主机模式下 0x4D 用于控制，批量和中断传输，0x4C 用于同步传输。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HOST_WO_HUB	RETRY_DIS	Res.	EP_CTL_DIS	EP_RX_EN	EP_TX_EN	EP_STALL	EP_HSHK
								RW	RW		RW	RW	RW	RW	RW

Bit	Field	Description																														
15: 8	Reserved	保留，必须保持复位值。																														
7	HOST_WO_HUB	仅用于主机模式，仅存在于端点 0 的控制寄存器中。 0: 通过 HUB 于低速设备通信 1: 可直接于低速设备进行通信																														
6	RETRY_DIS	仅用于主机模式，仅存在于端点 0 的控制寄存器中。 0: 重试 NAK 握手的事务 1: 不重试 NAK 握手的事务																														
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： <table> <tr> <td>EP_CTL_DIS</td><td>EP_RX_EN</td><td>EP_TX_EN</td><td>:</td><td>使能/方向控制 X</td></tr> <tr> <td>X</td><td>0</td><td>0</td><td>:</td><td>端点被禁用</td></tr> <tr> <td>X</td><td>0</td><td>1</td><td>:</td><td>TX</td></tr> <tr> <td>X</td><td>1</td><td>0</td><td>:</td><td>RX</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>:</td><td>控制传输，TX 和 RX</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>:</td><td>TX 和 RX</td></tr> </table>	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
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																														

24 FSMC 存储器接口

24.1 设计概述

FSMC 是一种灵活的存储器接口控制器，通过配置不同模式和时序设置，可以连接不同协议的外部设备。

24.1.1 框图

下图为 FSMC 的框图：

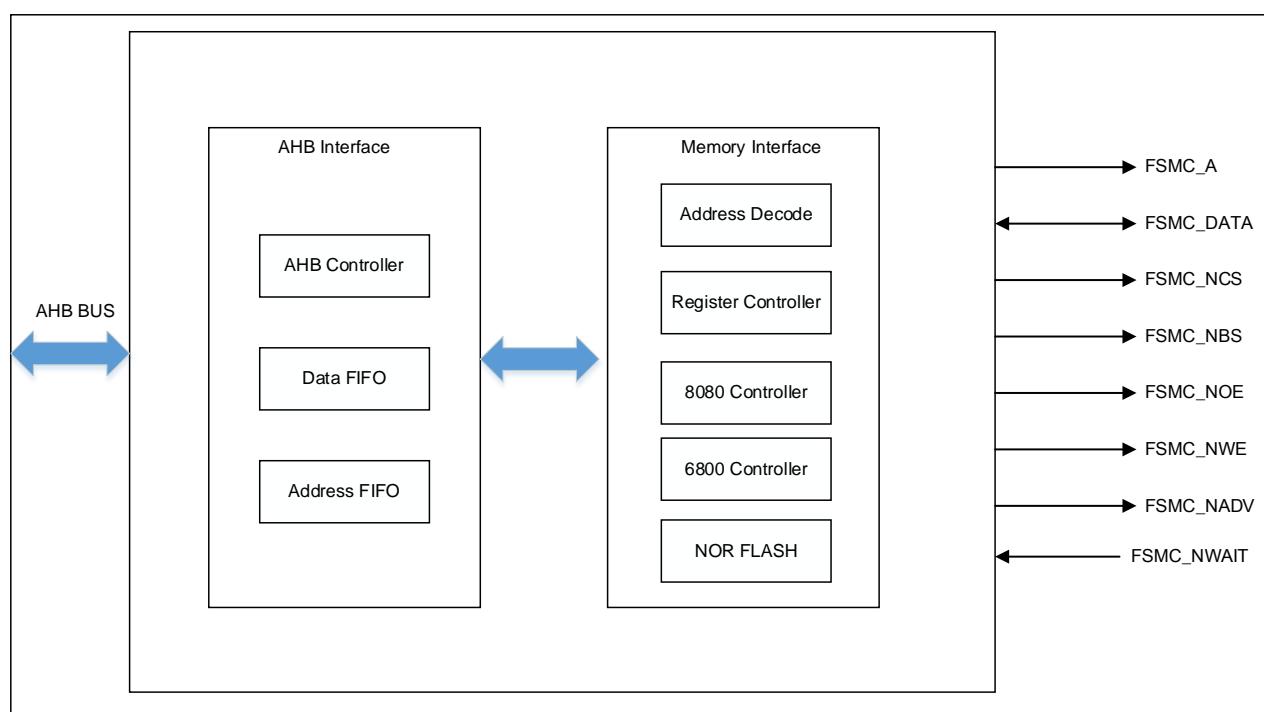


图 24-1 FSMC 框图

24.1.2 接口描述

表 24-1 接口描述

信号名	方向	描述
FSMC_A[25:0]	OUTPUT	地址信号
FSMC_DATA[15:0]	INPUT/OUTPUT	16 位数据（命令）输入输出复用端口
FSMC_NCS[3:0]	OUTPUT	BANK1 至 BANK4 位片选信号
FSMC_NBS[1:0]	OUTPUT	BYTE 选择信号
FSMC_NOE	OUTPUT	读使能信号
FSMC_NWE	OUTPUT	写使能信号

信号名	方向	描述
FSMC_NADV	OUTPUT	地址有效信号
FSMC_NWAIT	INPUT	FSMC 输入等待信号

24.1.3 特性

- 可配置的静态存储器接口包括：SRAM 和 NOR FLASH
- 支持 8080 协议
- 支持 6800 协议
- 8 位,16 位可配置的数据总线
- 分为 4 块 BANK，每块 64MB 空间
- 时序可编程以满足不同的需求：等待周期可编程，总线恢复周期可编程和写/读控制周期可编程
- 可将 32 位的 AHB 访问请求，转换为对外接设备连续的 8 位或 16 位的访问

表 24-2 NOR FLASH 控制器

内部控制器	地址范围	设备类型
NOR FLASH 控制器	0xX000_0000~0xXFFF_FFFF	SRAM NOR Flash

表 24-3 BANK 地址划分

所需存储区	片选信号	地址范围
Bank1	FSMC_NCS[0]	0xX000_0000~0xX3FF_FFFF
Bank2	FSMC_NCS[1]	0xX400_0000~0xX7FF_FFFF
Bank3	FSMC_NCS[2]	0xX800_0000~0xXBFF_FFFF
Bank4	FSMC_NCS[3]	0xXC00_0000~0xXFFF_FFFF

24.2 功能描述

24.2.1 FSMC 数据位宽

AHB 接口为内部 CPU 和 DMA 总线控制访问外部存储器等设备提供了通道，可支持 8 或 16 位接口，当选择的外部存储器的数据通道是 8 位或 16 位时，在 AHB 上的 32 位数据会被分割成连续的 8 位或 16 位数据，分多次完成传输。

寄存器 SMCTLR 的 SM_DATA_WIDTH[2:0]位，定义了外部存储器的数据宽度，用户需根据外部存储器实际数据宽度进行配置，不同配置对应不同的模式，如表 24-4 所示。

表 24-4 存储器位宽

外部存储器宽度 CPU/DMA 支持 AHB 访问宽度	8 位	16 位
8 位	模式一	模式三
16 位	模式二	模式一
32 位	模式二	模式二

FSMC 执行操作规则如下：

模式一： AHB 操作的数据宽度与存储器数据宽度相同，无数据传输一致性的问题。

模式二： AHB 操作的数据宽度大于存储器的数据宽度。存储器的数据宽度，将根据寄存器 SMCTRLR 的 SM_DW0, SM_DW1, SM_DW2 来设置。

例如当 SM_DW0 设置为 0x0 时，表示外部存储器是 16 Bits 数据宽度，此时，AHB 接口将对总线 16 位数据和高 16 位数据进行连续写操作，以适配外部设备的数据宽度；读操作时，总线数据的 16 位是有效数据。

模式三： AHB 操作的数据宽度小于存储器的数据宽度。若存储设备没有高低字节片选，不允许进行写操作，若存储设备有高低字节选择，通过“NBL”信号控制访问对应字节。该模式下，用户可以进行读操作，但有效数据需要用户自行处理。

24.2.2 FSMC 模式配置

FSMC 通过寄存器 SYSCFG_CFGR1 的 MODE_SEL 位来配置不同模式，默认值为 0x01

00：兼容 NOR FLASH、SRAM 接口

01：兼容 8080 协议接口

10：兼容 6800 协议接口

11：保留

除此之外，将 SYSCFG_CFGR1 的 FC_ODATAEN 位配置为 1，即地址数据复用引脚仅作为数据引脚使用。

以下为典型应用场景配置：

表 24-5 NOR FLASH 16bit 示例

模式	读/写	AHB 传输宽度	存储宽度	备注
异步访问	读	8	16	不支持
	写	8	16	不支持
	读	16	16	支持
	写	16	16	支持
	读	32	16	支持，分解成 2 次访问
	写	32	16	支持，分解成 2 次访问

表 24-6 SRAM 16bit 示例

模式	读/写	AHB 传输宽度	存储宽度	备注
异步访问	读	8	16	不支持
	写	8	16	不支持
	读	16	16	支持
	写	16	16	支持
	读	32	16	支持, 分解成 2 次访问
	写	32	16	支持, 分解成 2 次访问

寄存器 SMSKRn[10:8]用来为对应的 BANK 选择时序寄存器 SMTMGR_SETx, 以配置不同的时序, 其中, SMSKR0 寄存器用来配置 BANK1 的相关参数, SMSKR1 寄存器用来配置 BANK2 的相关参数, SMSKR2 寄存器用来配置 BANK3 的相关参数, SMSKR3 寄存器用来配置 BANK4 的相关参数。

24.2.3 NOR Flash/SRAM 模式

控制器外部接口信号在 NOR Flash/SRAM 工作模式的接口如表 24-7 所示, 时序参数说明如表 24-8 所示, NOR Flash/SRAM 写时序如图 24-2 所示, 其读时序如图 24-3 所示。

表 24-7 NOR FLASH/SRAM 接口

信号名	信号方向	使用情况	功能
A[25:0]	输出	使用	地址总线
D[15:0]	输入/输出	使用	双向数据总线
NE[3:0]	输出	使用	四个 BANK 片选
NOE	输出	使用	读使能
NWE	输出	使用	写使能
NADV	输出	未使用	数据/地址复用时的地址锁存
NBL[1:0]	输出	未使用	高低字节选择
NWAIT	输入	使用	外部存储器等待输入信号

表 24-8 NOR FLASH/SRAM 时序参数说明

写操作阶段时序要求	周期(参考 SMTMGR_SETn)	说明
T_addr_setup	T_AS 个 HCLK	地址建立时间, NADV 地址选择有效周期
T_addr_hold	固定为 1 个 HCLK 周期	地址保持时间
T_write_pulse	(T_WP + 1) 个 HCLK	写使能周期, NWE 写有效周期
T_write_recovery	T_WR 个 HCLK	写使能释放周期
T_read_cycle	(T_RC - 1) 个 HCLK 周期	读使能周期, NOE 读有效周期

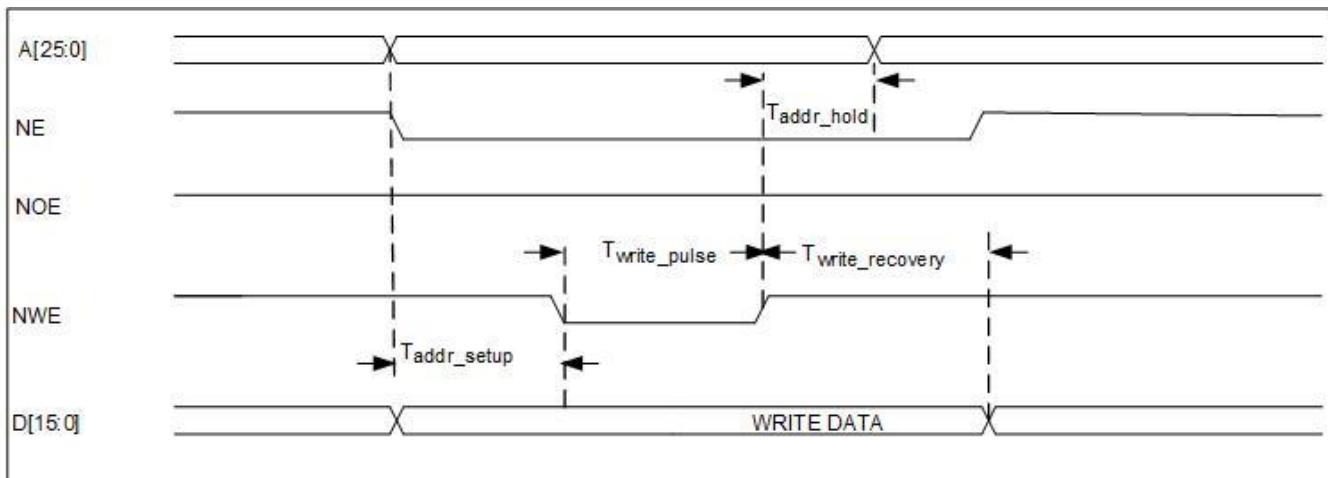


图 24-2 NOR FLASH/SRAM 写时序

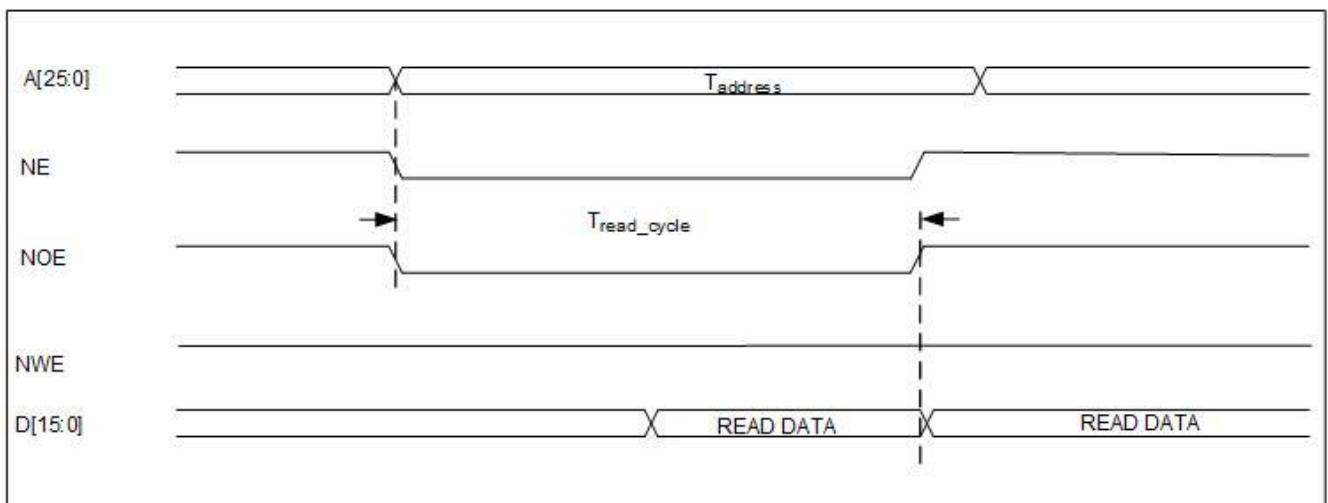


图 24-3 NOR FLASH/SRAM 读时序

24.2.4 8080 协议模式

8080 协议通信只用到了部分接口信号，接口如表 24-9 所示，地址说明如表 24-10 所示，8080 时序参数说明如表 24-11 所示，8080 写时序如图 24-4 所示，其读时序如图 24-5 所示。

表 24-9 8080 协议接口

信号名	信号方向	功能
A[25:0]	输出	只选其中一根地址线作为 RS 信号使用
NE[3:0]	输出	片选信号，低有效
NOE	输出	读使能信号，低有效
NWE	输出	写使能信号，低有效
D[15:0]	输入/输出	16 位数据或命令

在 8080 模式下，FSMC_A[25:16]中的一个 bit 作为 RS 控制信号。下面对地址信号之间的关系进行简单的描述：

表 24-10 8080 地址说明

SM_DWx 的设置	数据位	地址信号
100	8	FSMC_A[25:0]
000	16	FSMC_A[24:0]

例如：

若要写指令，RS 信号需为低电平(根据实际外接设备的时序标准)，

当 SM_DWx 设置为 100 时，向地址(FSMC_BANK1_BASE+(0x0«23))写入命令 CMD，则 FSMC_A[23]作为 RS 控制信号；

当 SM_DWx 设置为 000 时，向地址(FSMC_BANK1_BASE+(0x0«23))写入命令 CMD，则 FSMC_A[22]作为 RS 控制信号。

若要写数据，RS 信号需要为高电平(根据实际外接设备的时序标准)，

当 SM_DWx 设置为 100 时，向地址(FSMC_BANK1_BASE+(0x1«23))写入数据 DATA，则 FSMC_A[23]作为 RS 控制信号；

当 SM_DWx 设置为 000 时，向地址(FSMC_BANK1_BASE+(0x1«23))写入数据 DATA，则 FSMC_A[22]作为 RS 控制信号。

表 24-11 8080 时序参数说明

时序参数	周期(SMTMGR_SETn)	说明
T _{setup}	T_AS 个 HCLK 周期	建立时间
T _{write_pulse}	(T_WP + 1) 个周期	写使能周期，NWE 有效周期
T _{w_rcy}	T_WR 个周期	写使能释放周期
T _{read_cycle}	T_RC 个 HCLK 周期	读使能周期，NOE 有效周期

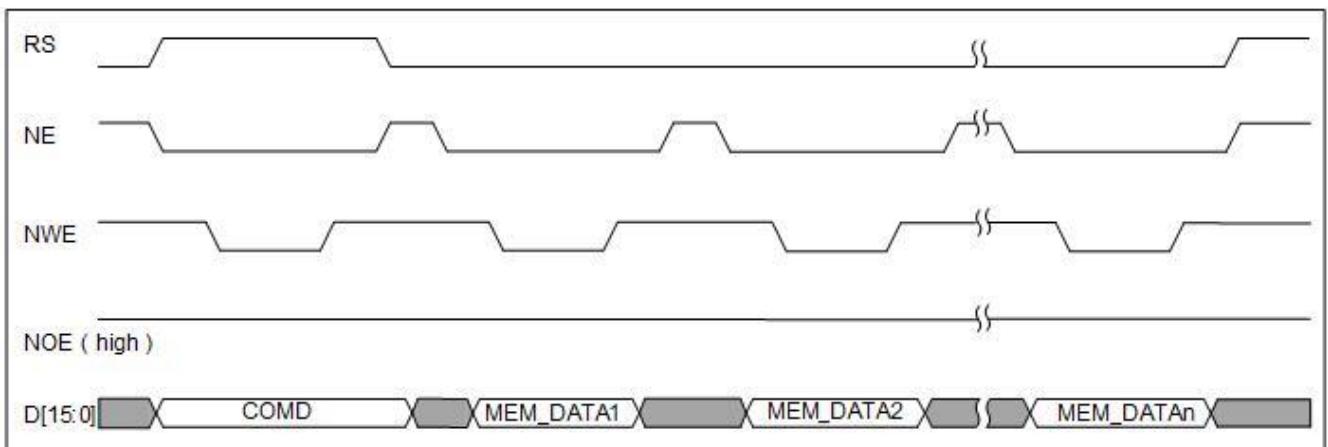


图 24-4 8080 写时序

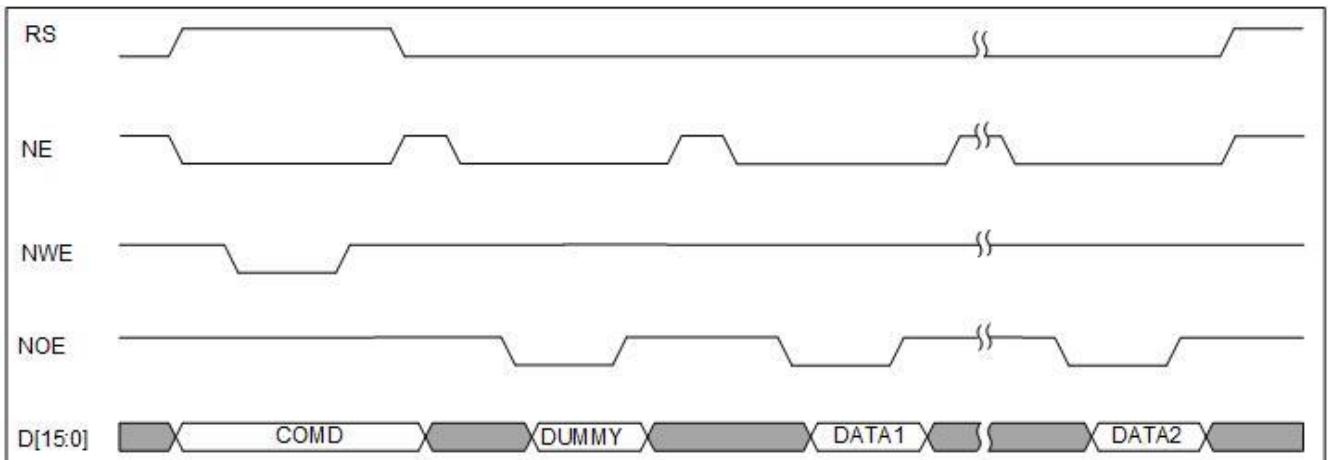


图 24-5 8080 读时序

24.2.5 6800 协议模式

6800 协议通讯只用到了部分接口信号，接口信号说明如表 24-12 所示，地址说明如表 24-13 所示，时序参数说明如表 24-14 所示，6800 写时序如图 24-6 所示，其读时序如图 24-7 所示。

表 24-12 6800 协议接口

信号名	信号方向	功能
A[25:0]	输出	只选其中一根地址线作为 RS 信号使用
NE[3:0]	输出	片选信号，低有效
NOE	输出	读使能信号，低有效
NWE	输出	写使能信号，低有效
D[15:0]	输入/输出	16 位数据/命令

在 6800 模式下，FSMC_A[25:16]中的一个地址线作为控制信号 RS（命令 / 数据选择）。下面对地址信号之间的关系进行简单的描述：FSMC_A[25:16]

表 24-13 6800 地址说明

SM_DWx	数据位	地址信号
100	8	FSMC_A[25:0]
000	16	FSMC_A[24:0]

例如：若要写指令，RS 信号需为低电平(根据实际外接设备的时序标准)，

当 SM_DWx 设置为 100 时，向地址(FSMC_BANK1_BASE+(0x0«23))写入命令 CMD，则 FSMC_A[23] 作为 RS 控制信号；

当 SM_DWx 设置为 000 时，向地址(FSMC_BANK1_BASE+(0x1«23))写入命令 CMD，则 FSMC_A[22] 作为 RS 控制信号。

若要写数据，RS 信号需为高电平(根据实际外接设备的时序标准)，

当 SM_DWx 设置为 100 时，向地址(FSMC_BANK1_BASE+(0x1«23))写入数据 DATA，则 FSMC_A[23] 作为 RS 控制信号；

当SM_DW_x设置为000时,向地址(FSMC_BANK1_BASE+(0x1«23))写入数据DATA,则FSMC_A[22]作为RS控制信号。

6800 写操作时序:

表 24-14 6800 时序参数说明

时序参数	周期(SMTMGR_SETn)	说明
T _{setup}	T_AS 个 HCLK 周期	建立时间
T _{write_pulse}	(T_WP + 1) 个 HCLK 周期	写使能周期, NWE 有效周期
T _{w_rcy}	T_WR 个 HCLK 周期	写使能释放周期
T _{read_cycle}	T_RC 个 HCLK 周期	读使能周期, NOE 有效周期

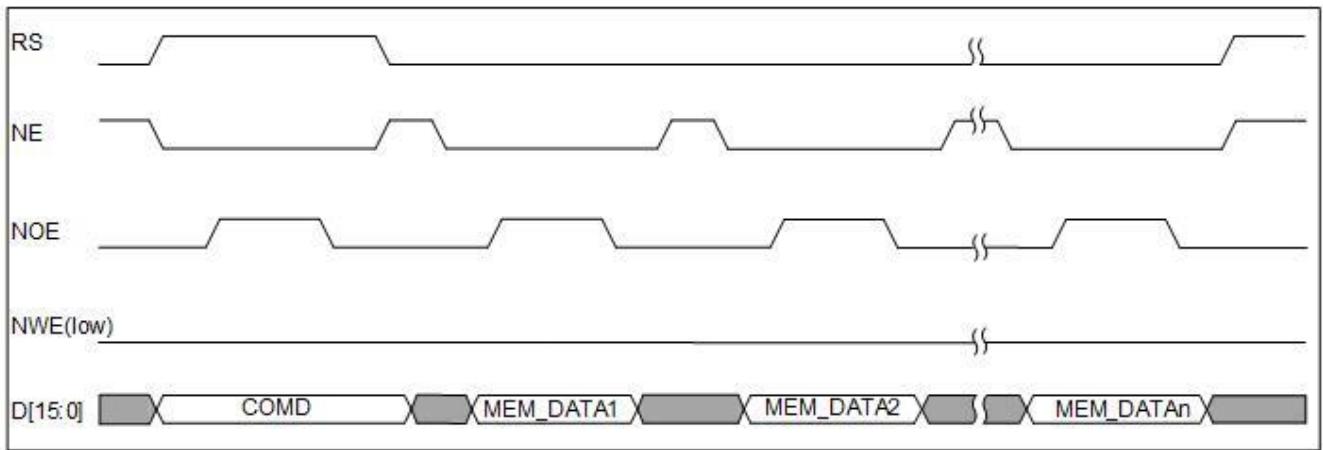


图 24-6 6800 写时序

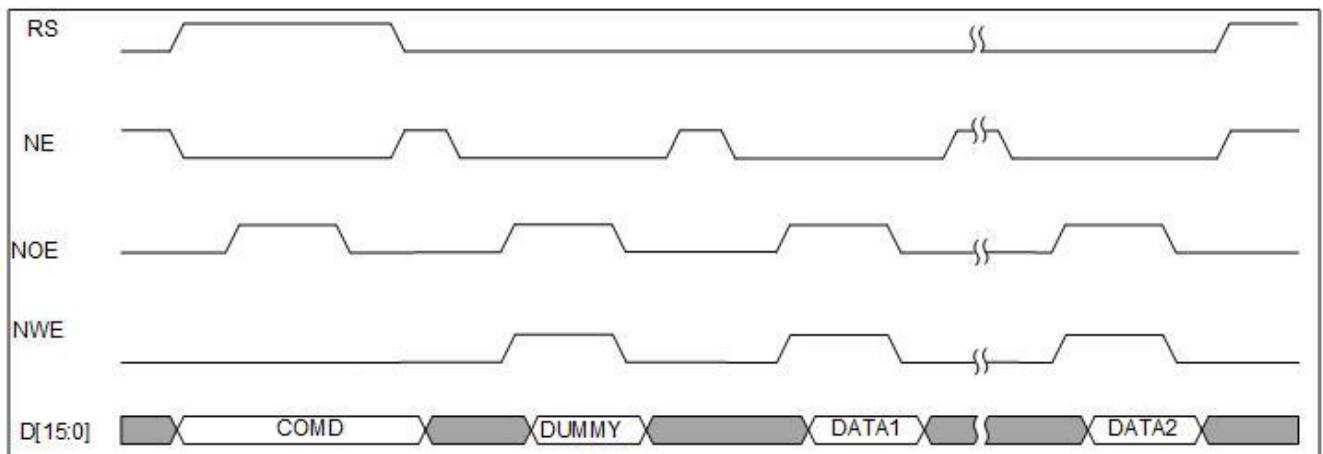


图 24-7 6800 读时序

24.3 寄存器描述

表 24-15 寄存器映射表

Offset	Acronym	Register Name	Reset
0x54	SMSKR0	存储器屏蔽寄存器 0	0x0000024B
0x58	SMSKR1	存储器屏蔽寄存器 1	0x0000024B

Offset	Acronym	Register Name	Reset
0x5C	SMSKR2	存储器屏蔽寄存器 2	0x0000002B
0x60	SMSKR3	存储器屏蔽寄存器 3	0x0000014B
0x94	SMTMGR_SET0	存储器时序寄存器 0	0x04010441
0x98	SMTMGR_SET1	存储器时序寄存器 1	0x007C4F5B
0x9C	SMTMGR_SET2	存储器时序寄存器 2	0x001C194E
0xA4	SMCTRL	存储器控制寄存器	0x00000000

24.3.1 存储器屏蔽寄存器 (SMSKR0)

地址偏移: 0x54

复位值: 0x00000024B

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					REG_SELECT			MEM_TYPE			MEM_SIZE				
					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	REG_SELECT	<p>REG_SELECT[2:0]: 时序参数配置寄存器组选择 000: 选择 SMTMGR_SET0 寄存器来配置 BANK1 传输的时序参数 001: 选择 SMTMGR_SET1 寄存器来配置 BANK1 传输的时序参数 010: 选择 SMTMGR_SET2 寄存器来配置 BANK1 传输的时序参数 其它: 保留</p>
7:5	MEM_TYPE	<p>MEM_TYPE[2:0]: 外接设备类型 001: SRAM 010: NOR FLASH 其他: 保留</p>

Bit	Field	Description
4:0	MEM_SIZE	<p>MEM_SIZE[4:0]: 外接设备容量</p> <p>00000: 未连接任何设备</p> <p>00001: 64KB</p> <p>00010 : 128KB</p> <p>00011: 256KB</p> <p>00100: 512KB</p> <p>00101: 1MB</p> <p>00110: 2MB</p> <p>00111: 4MB</p> <p>01000: 8MB</p> <p>01001: 16MB</p> <p>01010: 32MB</p> <p>01011: 64MB</p> <p>其它: 保留</p>

24.3.2 存储器屏蔽寄存器 (SMSKR1)

地址偏移: 0x58

复位值: 0x00000024B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																																																
Res.																																																															
Res.																																																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5px;"></td><td style="width: 5px;"></td></tr> <tr> <td colspan="5" style="text-align: center;">REG_SELECT</td><td colspan="4" style="text-align: center;">MEM_TYPE</td><td colspan="7" style="text-align: center;">MEM_SIZE</td></tr> <tr> <td style="text-align: center;">rw</td><td style="text-align: center;">rw</td></tr> </table>																																REG_SELECT					MEM_TYPE				MEM_SIZE							rw															
REG_SELECT					MEM_TYPE				MEM_SIZE																																																						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw																																																

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	REG_SELECT	<p>REG_SELECT[2:0]: 时序参数配置寄存器组选择</p> <p>000: 选择 SMTMGR_SET0 寄存器来配置 BANK2 传输的时序参数</p> <p>001: 选择 SMTMGR_SET1 寄存器来配置 BANK2 传输的时序参数</p> <p>010: 选择 SMTMGR_SET2 寄存器来配置 BANK2 传输的时序参数</p> <p>其它: 保留</p>
7:5	MEM_TYPE	<p>MEM_TYPE[2:0]: 外接设备类型</p> <p>001: SRAM</p> <p>010: NOR FLASH</p> <p>其他: 保留</p>

Bit	Field	Description
4:0	MEM_SIZE	<p>MEM_SIZE[4:0]: 外接设备容量</p> <p>00000: 未连接任何设备</p> <p>00001: 64KB</p> <p>00010: 128KB</p> <p>00011: 256KB</p> <p>00100: 512KB</p> <p>00101: 1MB</p> <p>00110: 2MB</p> <p>00111: 4MB</p> <p>01000: 8MB</p> <p>01001: 16MB</p> <p>01010: 32MB</p> <p>01011: 64MB</p> <p>其它: 保留</p>

24.3.3 存储器屏蔽寄存器 (SMSKR2)

地址偏移: 0x5C

复位值: 0x00000002B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																																																
Res.																																																															
Res.																																																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5px;"></td><td style="width: 5px;"></td></tr> <tr> <td colspan="5" style="text-align: center;">REG_SELECT</td><td colspan="4" style="text-align: center;">MEM_TYPE</td><td colspan="7" style="text-align: center;">MEM_SIZE</td></tr> <tr> <td style="text-align: center;">rw</td><td style="text-align: center;">rw</td></tr> </table>																																REG_SELECT					MEM_TYPE				MEM_SIZE							rw															
REG_SELECT					MEM_TYPE				MEM_SIZE																																																						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw																																																

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	REG_SELECT	<p>REG_SELECT[2:0]: 时序参数配置寄存器组选择</p> <p>000: 选择 SMTMGR_SET0 寄存器来配置 BANK3 传输的时序参数</p> <p>001: 选择 SMTMGR_SET1 寄存器来配置 BANK3 传输的时序参数</p> <p>010: 选择 SMTMGR_SET2 寄存器来配置 BANK3 传输的时序参数</p> <p>其它: 保留</p>
7:5	MEM_TYPE	<p>MEM_TYPE[2:0]: 外接设备类型</p> <p>001: SRAM</p> <p>010: NOR FLASH</p> <p>其他: 保留</p>

Bit	Field	Description
4:0	MEM_SIZE	<p>MEM_SIZE[4:0]: 外接设备容量</p> <p>00000: 未连接任何设备</p> <p>00001: 64KB</p> <p>00010: 128KB</p> <p>00011: 256KB</p> <p>00100: 512KB</p> <p>00101: 1MB</p> <p>00110: 2MB</p> <p>00111: 4MB</p> <p>01000: 8MB</p> <p>01001: 16MB</p> <p>01010: 32MB</p> <p>01011: 64MB</p> <p>其它: 保留</p>

24.3.4 存储器屏蔽寄存器 (SMSKR3)

地址偏移: 0x60

复位值: 0x00000014B

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

Res.

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

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	REG_SELECT	<p>REG_SELECT[2:0]: 时序参数配置寄存器组选择</p> <p>000: 选择 SMTMGR_SET0 寄存器来配置 BANK4 传输的时序参数</p> <p>001: 选择 SMTMGR_SET1 寄存器来配置 BANK4 传输的时序参数</p> <p>010: 选择 SMTMGR_SET2 寄存器来配置 BANK4 传输的时序参数</p> <p>其它: 保留</p>
7:5	MEM_TYPE	<p>MEM_TYPE[2:0]: 外接设备类型</p> <p>001: SRAM</p> <p>010: NOR FLASH</p> <p>其他: 保留</p>

Bit	Field	Description
4:0	MEM_SIZE	<p>MEM_SIZE[4:0]: 外接设备容量 00000: 未连接任何设备 00001: 64KB 00010: 128KB 00011: 256KB 00100: 512KB 00101: 1MB 00110: 2MB 00111: 4MB 01000: 8MB 01001: 16MB 01010: 32MB 01011: 64MB 其它: 保留</p>

24.3.5 存储器时序寄存器(SMTMGR_SET0)

地址偏移: 0x94

复位值: 0x04010441

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

Bit	Field	Description
31:30	Reserved	保留, 始终读为 0
29:28	SM_READ_PIPE	读数据延迟周期 (SM_READ_PIPE) 即从 SRAM 读数据输出端到 FSMC 内部读数据端之间的寄存器数。
27	Reserved	保留, 始终读为 0
26	READY_MODE	准备模式 (READY_MODE) 选择是否采用外部设备 FSMC_NWAIT 信号。 0: 不采用外部 FSMC_NWAIT 信号, 而采用内部相关信号; 1: 采用外部设备 FSMC_NWAIT 信号。
25:16	Reserved	保留, 始终读为 0
15:10	T_WP	写操作周期 (Write Pulse Width Time) 取值范围 0~63, 对应写脉冲宽度 1~64 个时钟周期
9:8	T_WR	写操作时地址/数据保持时间 (Write Address/Data Hold time) 取值范围 0~3, 对应写操作时地址的保持时间 0~3 个时钟周期

Bit	Field	Description
7:6	T_AS	写操作时地址的建立时间 (Write Address Setup Time) 取值范围 0~3, 对应写操作时地址的建立时间 0~3 个时钟周期
5:0	T_RC	读操作周期 (Read Cycle Time) 取值范围 1~63, 对应读操作周期 0~62 个时钟周期

24.3.6 存储器时序寄存器(SMTMGR_SET1)

地址偏移: 0x98

复位值: 0x007C4F5B

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

Bit	Field	Description
31:30	Reserved	保留, 始终读为 0
29:28	SM_READ_PIPE	读数据延迟周期 (SM_READ_PIPE) 即从 SRAM 读数据输出端到 FSMC 内部读数据端之间的寄存器数。
27	Reserved	保留, 始终读为 0
26	READY_MODE	准备模式 (READY_MODE) 选择是否采用外部设备 FSMC_NWAIT 信号。 0: 不采用外部 FSMC_NWAIT 信号, 而采用内部相关信号; 1: 采用外部设备 FSMC_NWAIT 信号。
25:16	Reserved	保留, 始终读为 0
15:10	T_WP	写操作周期 (Write Pulse Width Time) 取值范围 0~63, 对应写脉冲宽度 1~64 个时钟周期
9:8	T_WR	写操作时地址/数据保持时间 (Write Address/Data Hold time) 取值范围 0~3, 对应写操作时地址的保持时间 0~3 个时钟周期
7:6	T_AS	写操作时地址的建立时间 (Write Address Setup Time) 取值范围 0~3, 对应写操作时地址的建立时间 0~3 个时钟周期
5:0	T_RC	读操作周期 (Read Cycle Time) 取值范围 1~63, 对应读操作周期 0~62 个时钟周期

24.3.7 存储器时序寄存器(SMTMGR_SET2)

地址偏移: 0x9C

复位值: 0x001C194E

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

Bit	Field	Description
31:30	Reserved	保留, 始终读为 0
29:28	SM_READ_PIPE	读数据延迟周期 (SM_READ_PIPE) 即从 SRAM 读数据输出端到 FSMC 内部读数据端之间的寄存器数。
27	Reserved	保留, 始终读为 0
26	READY_MODE	准备模式 (READY_MODE) 选择是否采用外部设备 FSMC_NWAIT 信号。 0: 不采用外部 FSMC_NWAIT 信号, 而采用内部相关信号; 1: 采用外部设备 FSMC_NWAIT 信号。
25:16	Reserved	保留, 始终读为 0
15:10	T_WP	写操作周期 (Write Pulse Width Time) 取值范围 0~63, 对应写脉冲宽度 1~64 个时钟周期
9:8	T_WR	写操作时地址/数据保持时间 (Write Address/Data Hold time) 取值范围 0~3, 对应写操作时地址的保持时间 0~3 个时钟周期
7:6	T_AS	写操作时地址的建立时间 (Write Address Setup Time) 取值范围 0~3, 对应写操作时地址的建立时间 0~3 个时钟周期
5:0	T_RC	读操作周期 (Read Cycle Time) 取值范围 1~63, 对应读操作周期 0~62 个时钟周期

24.3.8 存储器控制寄存器(SMCTLR)

地址偏移: 0xA4

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
SM_DW2		SM_DW1		SM_DW0		Res.								FLASH_RP	
rw		rw		rw		rw									

Bit	Field	Description
31:16	Reserved	保留, 始终读为 0

Bit	Field	Description
15:13	SM_DW2	存储器数据总线位宽设置 (Static Memory Databus Bit Width Setting) 与寄存器 SMTMGR_SET2 设置的时序相匹配 000: 16 bits 100: 8 bits 其它: 保留
12:10	SM_DW1	存储器数据总线位宽设置 (Static Memory Databus Bit Width Setting) 与寄存器 SMTMGR_SET1 设置的时序相匹配 000: 16 bits 100: 8 bits 其它: 保留
9:7	SM_DW0	存储器数据总线位宽设置 (Static Memory Databus Bit Width Setting) 与寄存器 SMTMGR_SET0 设置的时序相匹配 000: 16 bits 100: 8 bits 其它: 保留
6:1	Reserved	保留
0	FLASH_RP	FLASH 复位低功耗模式选择 (FLASH Reset Low-Power Mode Selection) 0: 强制 FLASH 进入低功耗模式 1: 将 FLASH 退出低功耗模式

25 ADC 模拟/数字转换

25.1 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 ADC1 有 14 路外部输入通道和 2 路内部通道，ADC2 和 ADC3 有 16 路外部输入通道。这些 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，由 APB2 时钟(PCLK2)分频产生。

25.2 功能框图

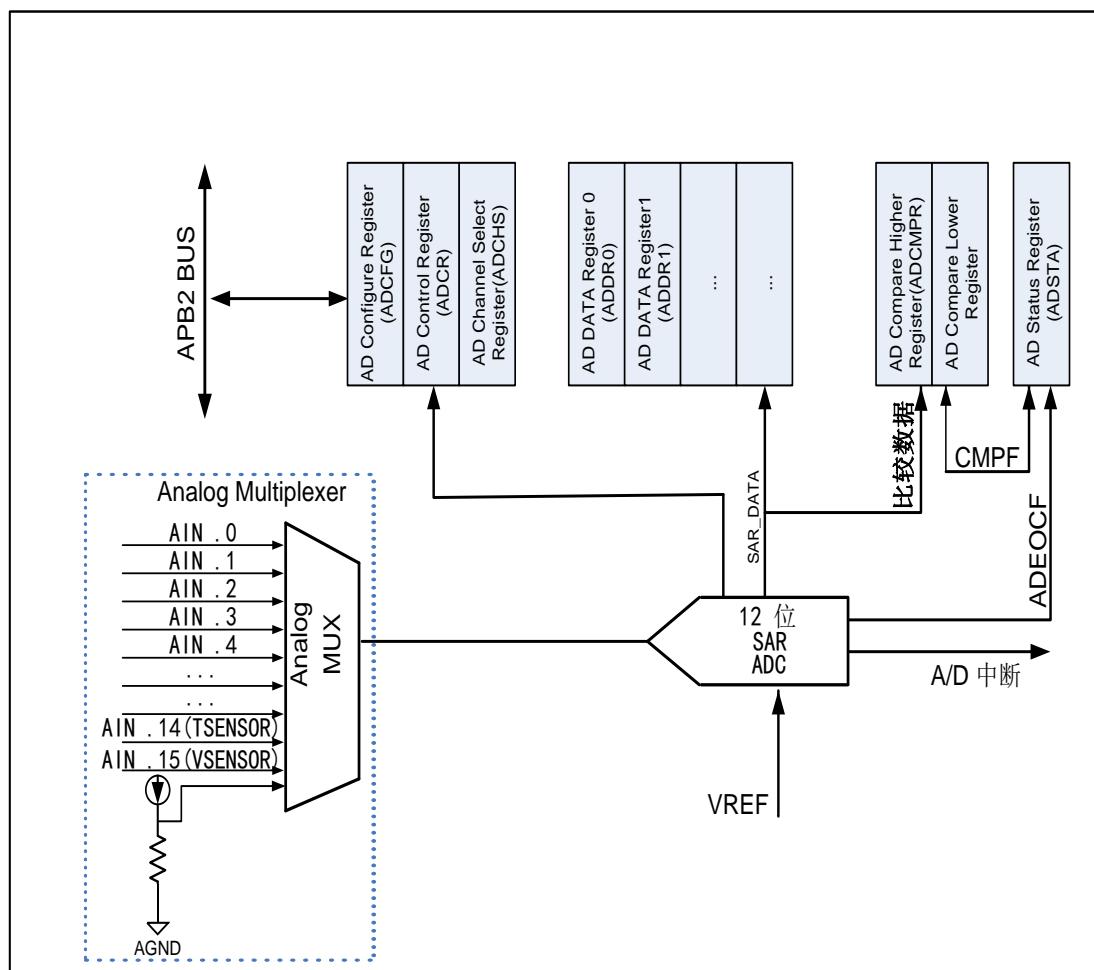


图 25-1 ADC 系统框图

注：T SENSOR（温度传感器）通道在 ADC1 的AIN14 通道，V SENSOR（内部参考电压）通道

在 ADC1 的 AIN15 通道。

25.3 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
- 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

25.4 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 25-1 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

中断	标志位
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

25.5 DMA

A/D 转换结果存储在数据寄存器 ADC_ADDATA 中，当多个通道连续转换时可以使用 DMA 访问保存数据，避免转换数据丢失。

DMA 使能开启，当通道转换结束后将产生 DMA 请求，将转换数据从 ADC_ADDATA 寄存器传输到软件指定的目的地址。

每个通道有对应的数据寄存器 ADC_ADDRn，可以访问这些寄存器获取各通道的转换结果。

25.6 功能描述

25.6.1 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。

25.6.2 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 25-2 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	-
11bit	SAR_DATA[11:1],0	ADC_JOFRn.JOFFESET	11 位有符号数据	用户必须配置 ADC_JOFRn.JOFFSET[0]为 “0”

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

25.6.3 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：

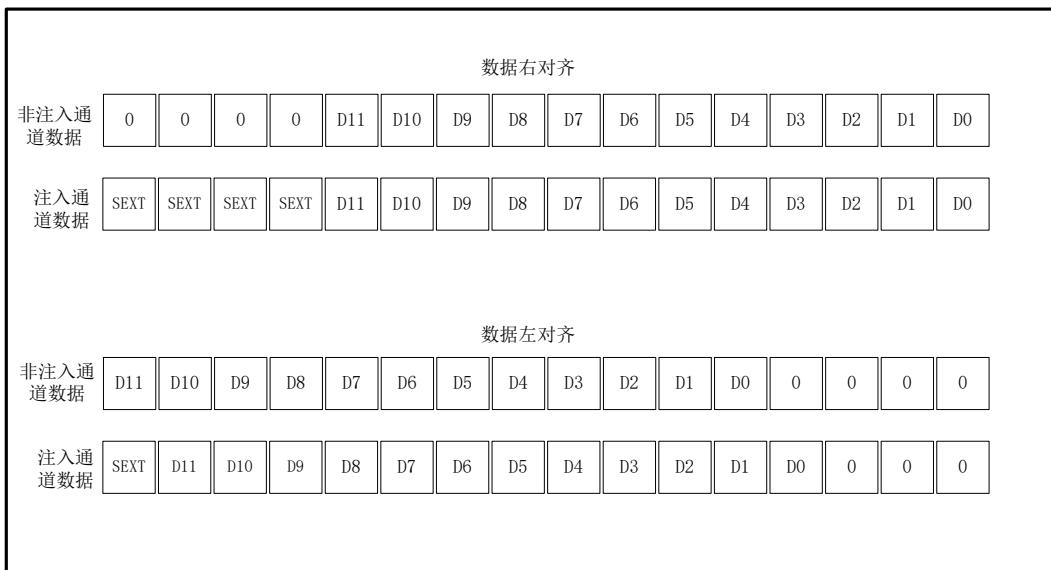


图 25-2 数据对齐方式

25.6.4 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。

25.6.5 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：

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

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

总转换时间计算如下：

T_{CONV} =采样时间+12.5 个转换周期。

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

25.6.6 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_ADDATA 中，ADC_ADDATA.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。

25.6.7 通道选择

ADC1 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，ADC2 和 ADC3 有 16 路外部输入通道 0~15。

不同工作方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。

25.7 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。

25.7.1 普通通道转换

25.7.1.1 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：

- 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_ADDATA 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。

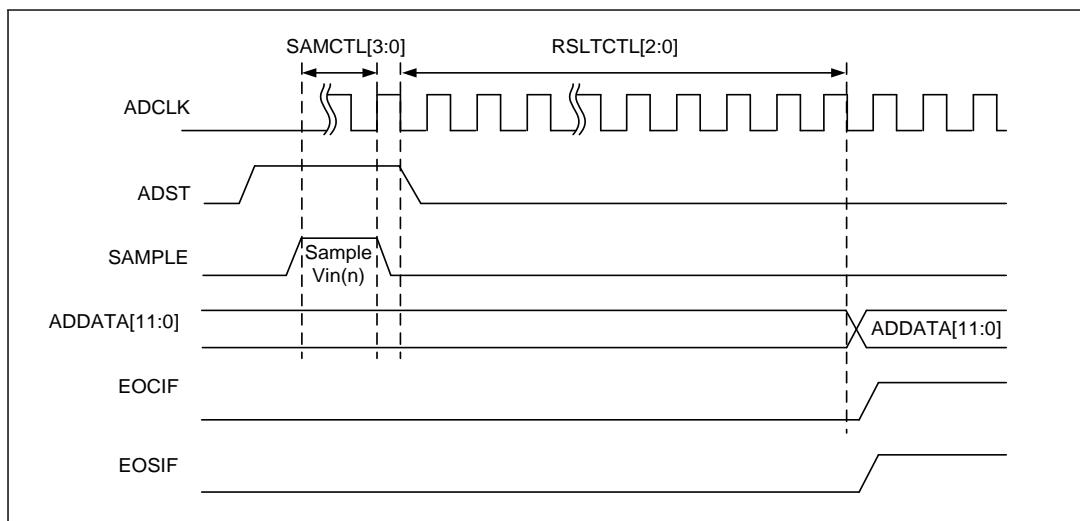


图 25-3 单次转换模式时序图

25.7.1.2 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_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, ADC_ADCR.ADST 位自动硬件清 0, A/D 转换器会进入空闲状态。

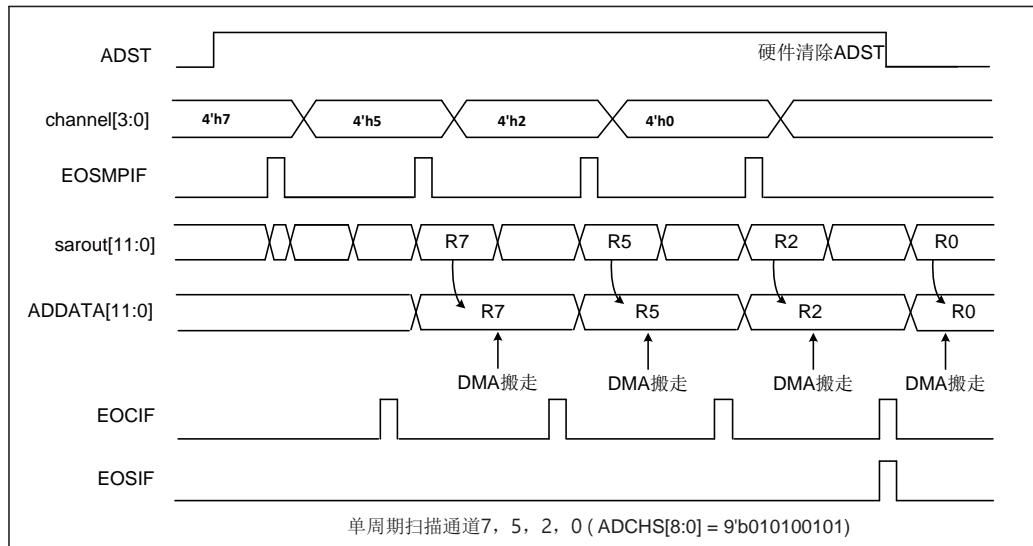


图 25-4 单周期扫描下使能通道转换时序图 (通道方向从高到低)

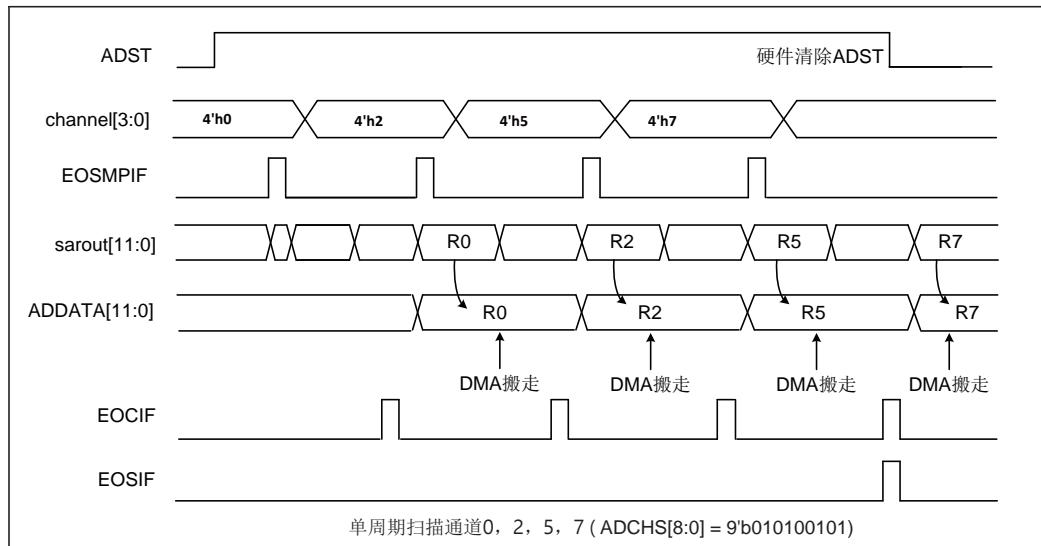


图 25-5 单周期扫描下使能通道转换时序图 (通道方向从低到高)

25.7.1.3 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。

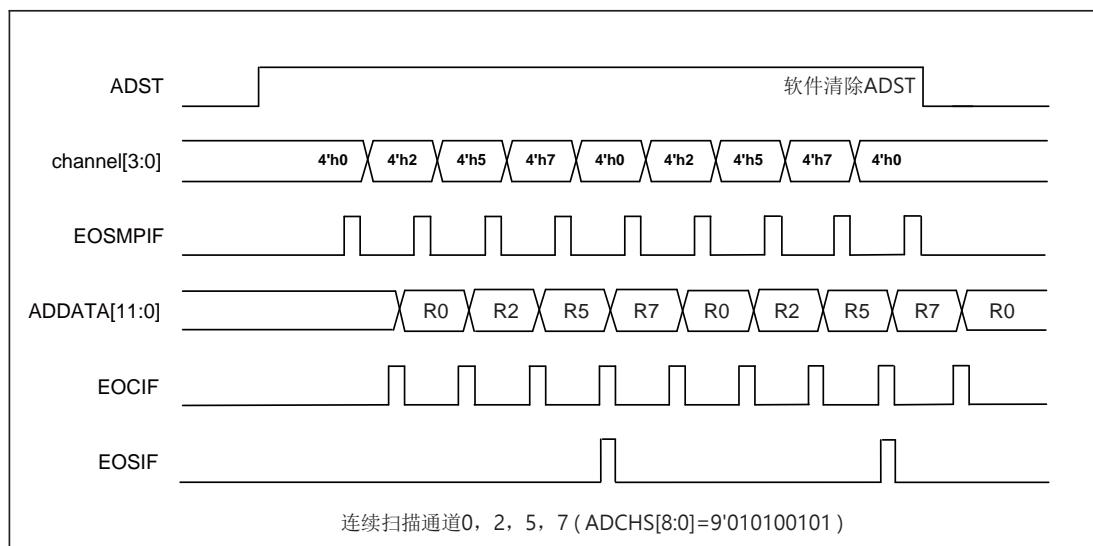


图 25-6 连续扫描模式使能通道转换时序图（通道方向由低到高）

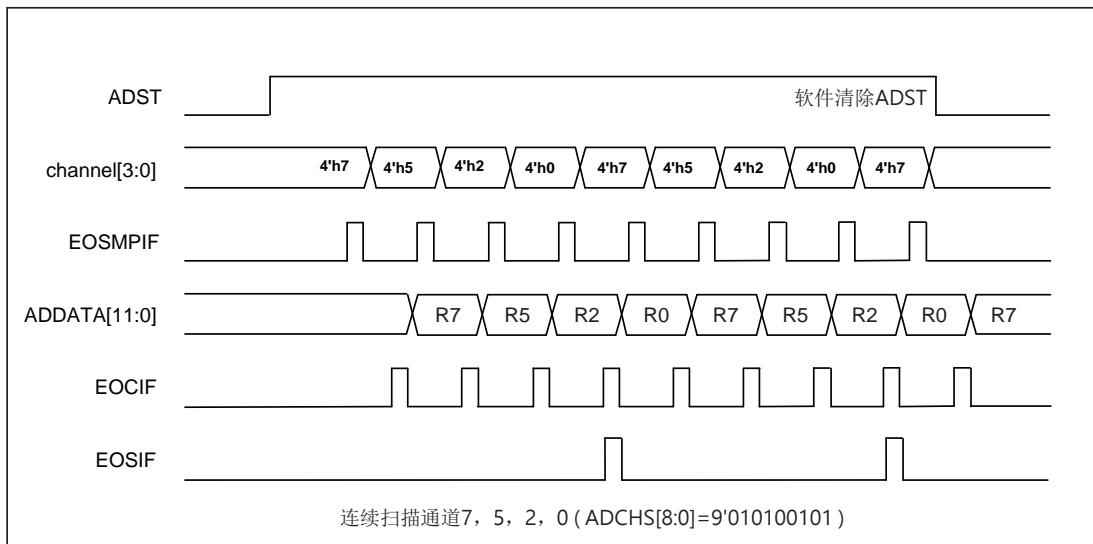


图 25-7 连续扫描模式使能通道转换时序图（通道方向由高到低）

25.7.2 任意通道转换

25.7.2.1 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_ADDATA 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。

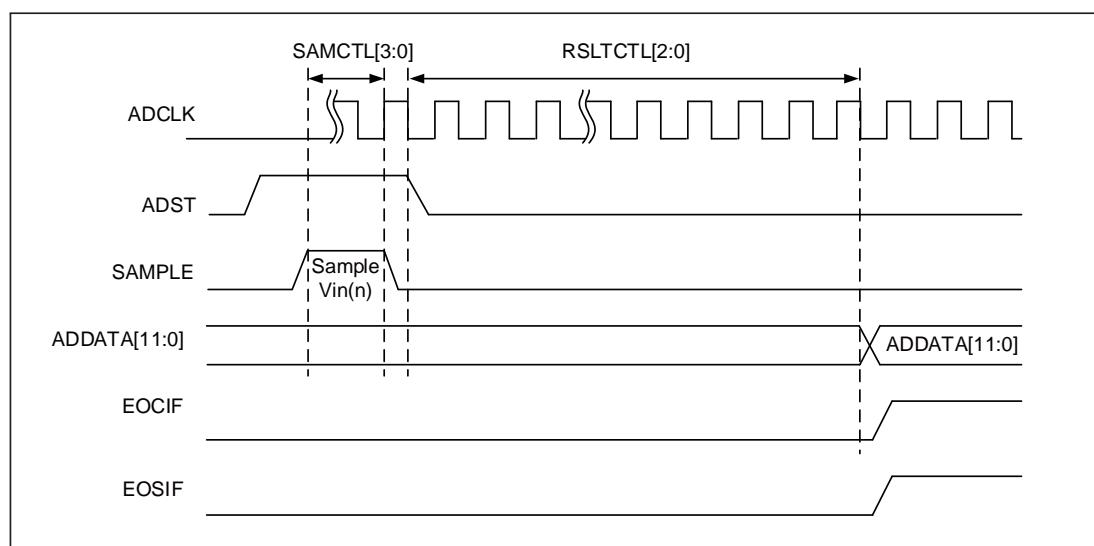


图 25-8 单次转换模式下通道转换时序图

25.7.2.2 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。

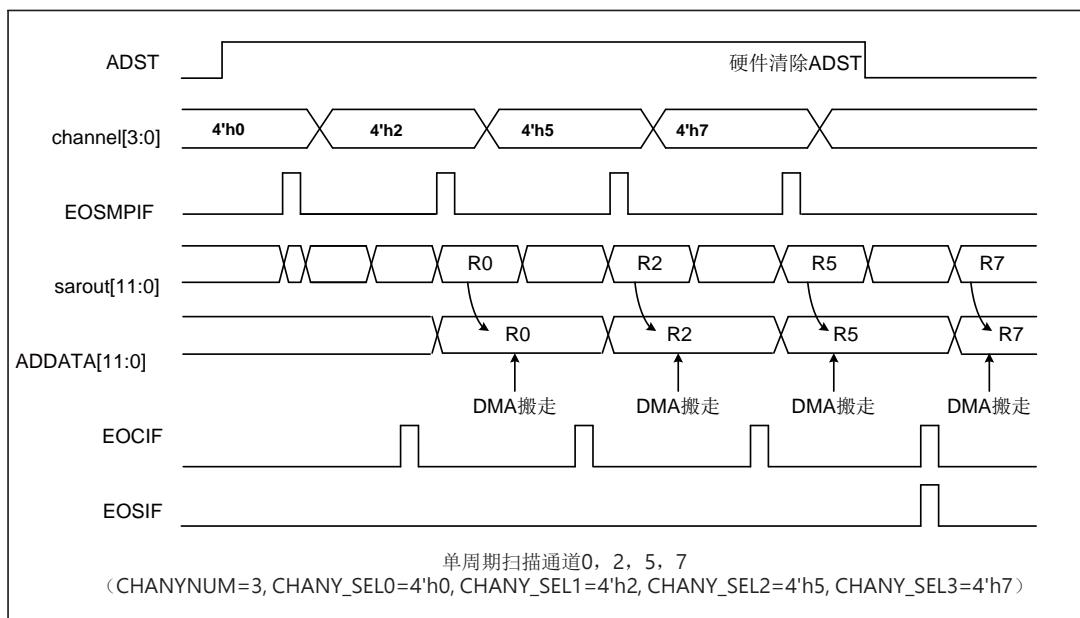


图 25-9 单周期扫描下通道转换时序图

25.7.2.3 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_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，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。

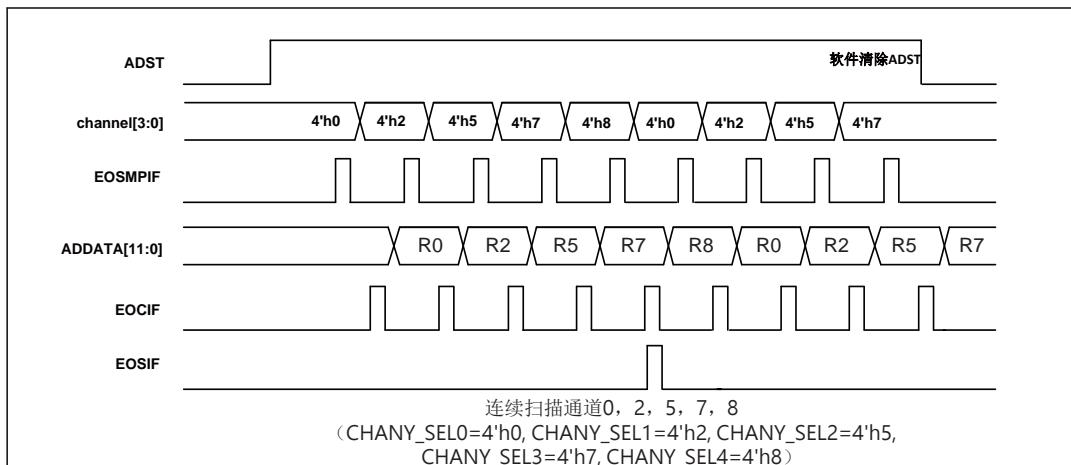


图 25-10 连续扫描模式下通道转换时序图

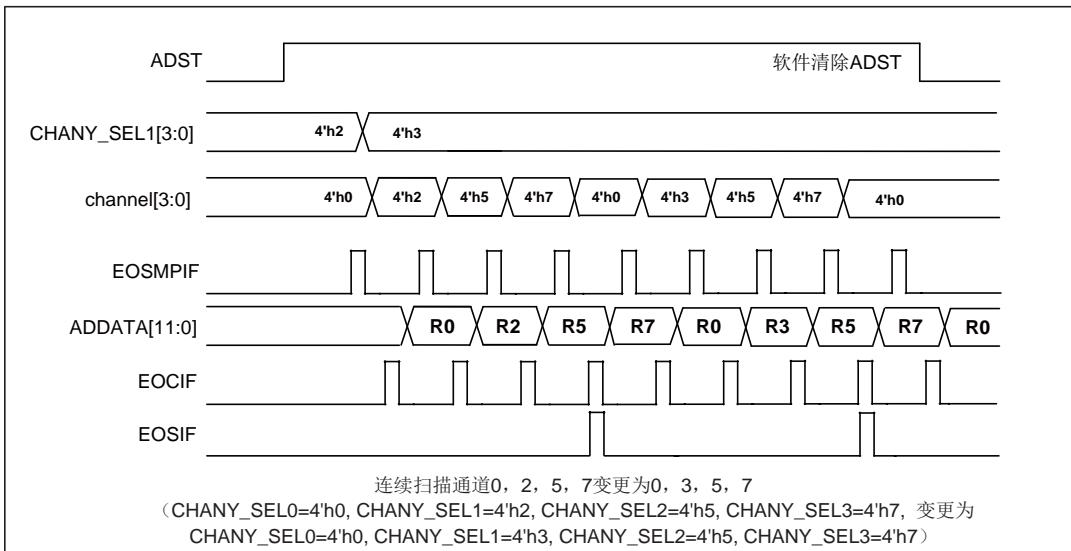


图 25-11 连续扫描模式下动态更新配置时序图

25.7.3 注入通道转换

25.7.3.1 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。

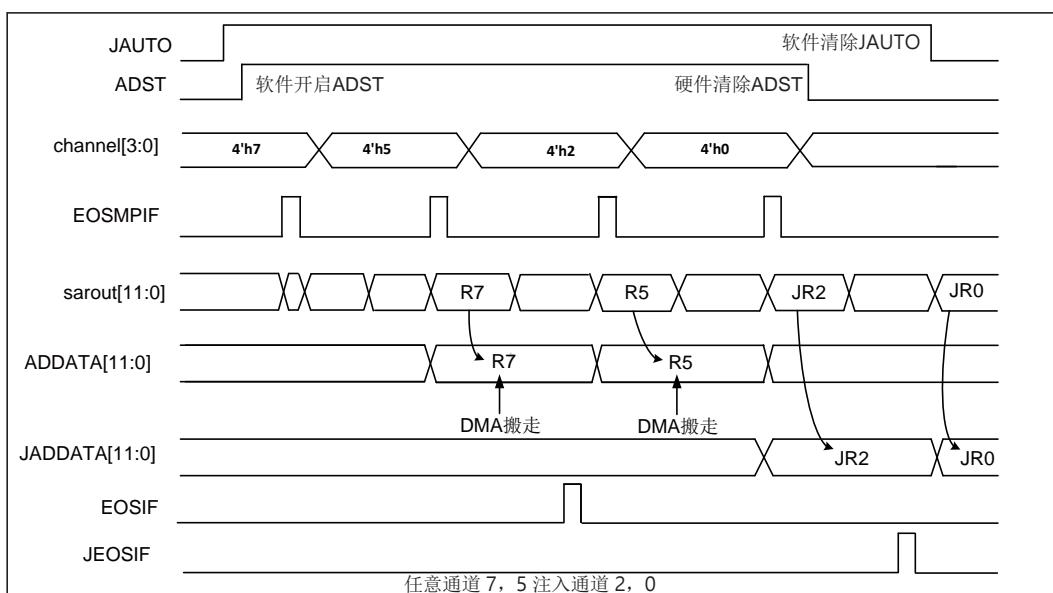


图 25-12 自动注入通道转换单周期扫描通道转换时序图

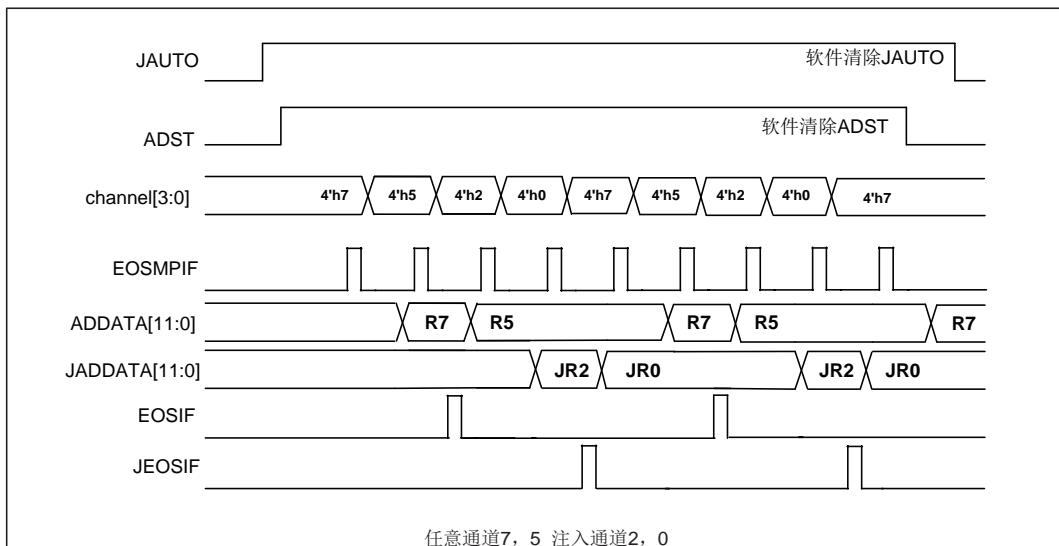


图 25-13 自动注入通道转换连续转换时序图

25.7.4 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。

25.7.5 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.CMPLDATA 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.CMPLDATA 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。

25.7.6 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 0x1FFFF7FC 的高 16 位。

V_{SENSE} : 温度传感器当前的输出电压

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

Avg_Slope: 温度与 V_{SENSE} 曲线的平均斜率（以 mV/°C 或 μV/°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。

25.7.7 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 VDD 为 3.3V, V_SENSOR 通道转换值为 ADC_ADDR(VSensor)计算公式如下：

$$3.3/4096 = V_{ref} / \text{ADC_ADDR(VSensor)}$$

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

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 0x1FFFF7FC 的低 16 位，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：

$$V_{ref_calc} / \text{ADC_ADDR(VSensor)} = V_{DDA} / 4096$$

$$V_{DDA} = (V_{ref_calc} * 4096) / \text{ADC_ADDR(VSensor)}$$

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。

25.8 寄存器

25.8.1 寄存器总览

表 25-3 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDATA	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_ADESTA	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_ADESTA_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
0xB0~0xBC	ADC_JDR0~3	A/D 注入通道数据寄存器	0x00000000

25.8.2 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.												VAILD	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。											

Bit	Field	Description
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。
19:16	CHANNELSEL	该 4 位显示当前数据寄存器中的数据所对应的通道 (Channel Selection) 0000 : 表示当前数据寄存器所属通道为通道 0 0001 : 表示当前数据寄存器所属通道为通道 1 0010 : 表示当前数据寄存器所属通道为通道 2 0011 : 表示当前数据寄存器所属通道为通道 3 0100 : 表示当前数据寄存器所属通道为通道 4 0101 : 表示当前数据寄存器所属通道为通道 5 0110 : 表示当前数据寄存器所属通道为通道 6 0111 : 表示当前数据寄存器所属通道为通道 7 1000 : 表示当前数据寄存器所属通道为通道 8 1001 : 表示当前数据寄存器所属通道为通道 9 1010 : 表示当前数据寄存器所属通道为通道 10 1011 : 表示当前数据寄存器所属通道为通道 11 1100 : 表示当前数据寄存器所属通道为通道 12 1101 : 表示当前数据寄存器所属通道为通道 13 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) 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)
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。

25.8.3 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	JAWDE
																N
																RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	ADCPR EL	Res.				RSLTCTL			ADCPREH			VSEN	TSEN	AWDEN	ADEN	

	rw			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 注入通道转换结果看门狗使能 (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
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= (ADCPRE+2)
3	VSEN	内部参考电压使能 (Voltage Sensor Enable) 仅 ADC1 有效 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) 仅 ADC1 有效 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

25.8.4 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		
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
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 0001 : 选择通道 1 转换结果作为比较对象 0010 : 选择通道 2 转换结果作为比较对象 0011 : 选择通道 3 转换结果作为比较对象 0100 : 选择通道 4 转换结果作为比较对象 0101 : 选择通道 5 转换结果作为比较对象 0110 : 选择通道 6 转换结果作为比较对象 0111 : 选择通道 7 转换结果作为比较对象 1000 : 选择通道 8 转换结果作为比较对象 1001 : 选择通道 9 转换结果作为比较对象 1010 : 选择通道 10 转换结果作为比较对象 1011 : 选择通道 11 转换结果作为比较对象 1100 : 选择通道 12 转换结果作为比较对象 1101 : 选择通道 13 转换结果作为比较对象 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) 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 0: 右对齐</p>
10 : 9	ADMD	<p>A/D 转换模式 (ADC Mode)</p> <p>00: 单次转换 01: 单周期扫描 10: 连续扫描 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 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} ADC1、ADC2、ADC3 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: TIM8_CC4 01010: TIM8_CC4 和 TIM8_CC5 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。

25.8.5 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
CHENV S	CHENT S	CHEN1 3	CHEN1 2	CHEN11	CHEN1 0	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	ADC1: 内部参考电压使能 (Voltage Sensor Enable) ADC2/3: 模拟输入通道 15 使能 (Analog Input Channel 15 enable) 1: 使能 0: 禁止
14	CHEN14	ADC1: 温度传感器使能 (Temperature Sensor Enable) ADC2/3: 模拟输入通道 14 使能 (Analog Input Channel 14 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

Bit	Field	Description
4	CHEN4	模拟输入通道 4 使能 (Analog Input Channel 4 Enable) 1: 使能 0: 禁止
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	CHENO	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 或 0xF。
- 2) 在 ADC_ADSTA.BUSY 无效时才能操作此寄存器。

25.8.6 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											
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

25.8.7 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			
r												Res.	BUSY	AWDIF	EOSIF
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。

25.8.8 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	Res.	
r												UN	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA												r			

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值。
21	VALID	有效标志位（只读）(Valid Flag) 1: DATA[11:0]位数据有效 0: DATA[11:0]位数据无效 对应通道转换完成后硬件置位，读 ADC_ADDRn 寄存器自动清除。
20	OVERRUN	数据覆盖标志位（只读）(Overrun Flag) 1: DATA [11:0]数据被覆盖 0: DATA [11:0]数据最近一次转换结果 若在装载至 DATA[11:0] 数据寄存器之前，上一次 DATA[11:0]数据没有被读取，OVERRUN 将置位。读 ADC_ADDRn 寄存器硬件自动清除。
19 : 16	Reserved	保留，必须保持复位值。
15 : 0	DATA	通道 n 的 12 位 A/D 转换结果 (Covert Data) 软件选择对齐方式。

25.8.9 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。

Bit	Field	Description
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。
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: ADC2/3 通道 15 ADC1: 内部参考电压 (V_SENSOR) 0100: ADC2/3 通道 14 ADC1: 温度传感器 (T_SENSOR) 0010: 通道 13 0001: 通道 12
3:0	VALID	通道的有效标志位 (Valid Flag) 1000: ADC2/3 通道 15 ADC1: 内部参考电压 (V_SENSOR) 0100: ADC2/3 通道 14 ADC1: 温度传感器 (T_SENSOR) 0010: 通道 13 0001: 通道 12

25.8.10 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。

25.8.11 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

Bit	Field	Description
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。

25.8.12 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	
Res.														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_SEL0~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_SEL0~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。

25.8.13 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
12:11	JTRGSEL	<p>ADC1/ADC2 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_TRGO 00001: TIM1_CC4 00010: TIM1_CC4 和 TIM1_CC5 00011: TIM2_CC1 00100: TIM3_CC4 00101: TIM8_CC4 00110: TIM8_CC4 和 TIM8_CC5 00111: EXTI 12</p> <p>其它: 无效</p> <p>选择外部触发源 ADC3</p> <p>00000: TIM1_TRGO 00001: TIM1_CC4 00010: TIM1_CC4 和 TIM1_CC5 00011: TIM4_CC3 00100: TIM5_CC4 00101: TIM8_CC4 00110: TIM8_CC4 和 TIM8_CC5 00111: EXTI 12</p> <p>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:</p> <p>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 JEOCIF 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。

25.8.14 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) 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>

25.8.15 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	<p>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。</p> <p>0000: 2.5 周期 0001: 8.5 周期 1000: 3.5 周期 0100: 42.5 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 1001: 4.5 周期 1010: 5.5 周期 1011: 6.5 周期 1100: 7.5 周期</p> <p>其它: 保留</p>

25.8.16 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	<p>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)</p> <p>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.</p> <p>注: 该位在 JADST=0 时允许进行写操作</p>

25.8.17 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。

25.8.18 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.										JVAILD	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。

25.8.19 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。

26 DAC 数字/模拟转换器

26.1 简介

数字 / 模拟转换器（DAC）包含了两个独立的 12 位数据输入通道和 2 个模拟电压输出通道。每个通道都有单独的转换器，既可以工作在单通道模式，也可以工作在双通道模式。在双通道模式下，可以同步地更新 2 个通道的输出数据，2 个通道的数据转换既能各自分别进行，也可以同时进行。

数字 / 模拟转换数据有 8 位或者 12 位模式，能配置成与 DMA 控制器协同工作。DAC 工作在 12 位数据模式时，数据可以配置成左对齐格式或者右对齐格式。

为了保证 DAC 能够正常工作，DAC 的输入时钟频率不能超过 1MHz。

26.2 主要特征

- 具有 2 个 DAC 转换通道，每个通道各自对应 1 个转换器
- 具有单通道模式和双通道模式
- 双通道数据转换可同时或者分别进行
- 每个通道都有 DMA 功能
- 转换数据可以 8 位或者 12 位输出
- 在 12 位数据模式下，数据可以配置成左对齐或者右对齐格式
- 具有数据同步更新功能
- 具有噪声波形生成功能
- 具有三角波形生成功能
- 既可由软件触发转换，也可由外部信号触发转换

DAC 单个通道的功能框图如下：

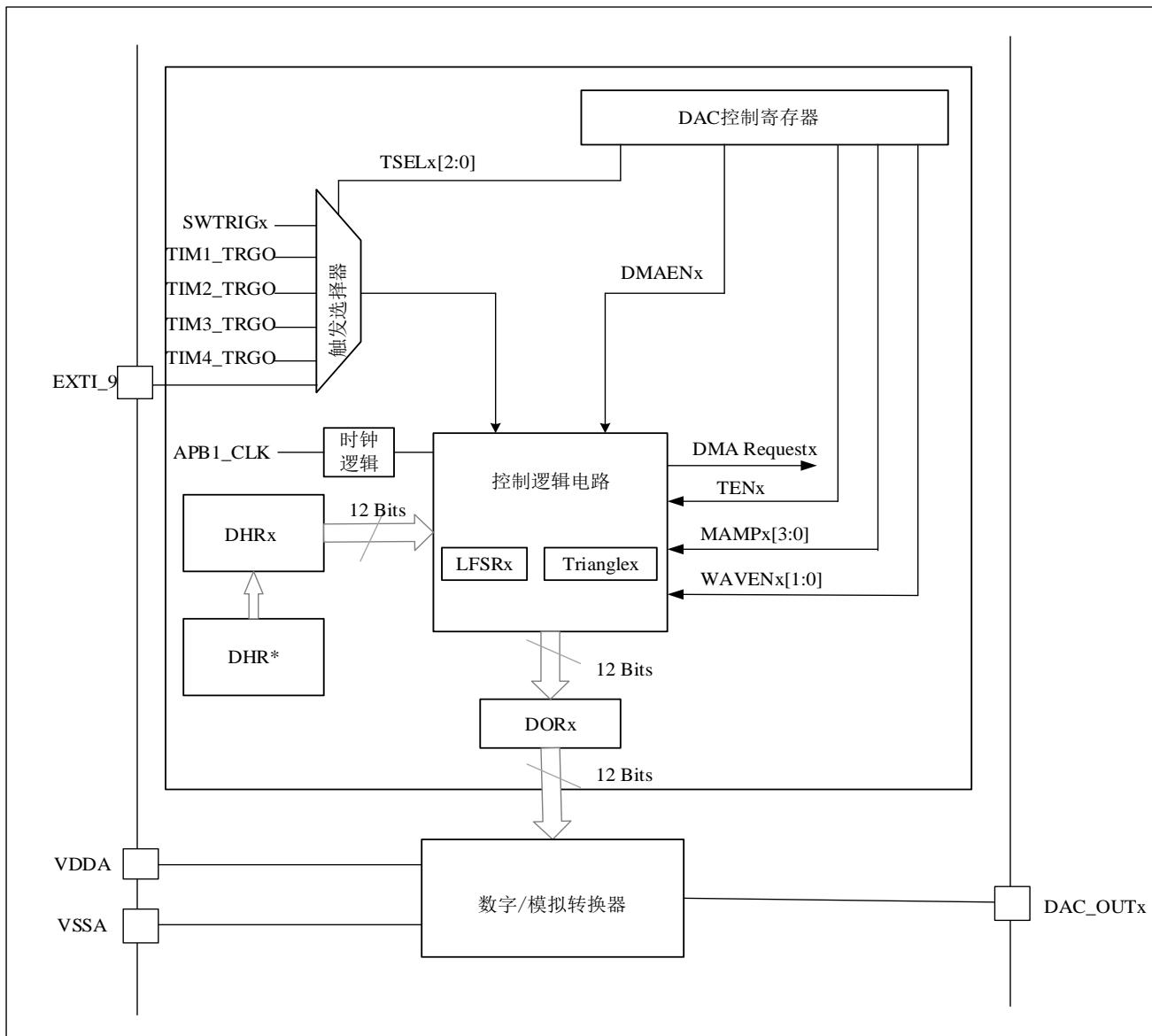


图 26-1 DAC 单通道功能框图

相关管脚的描述如下：

表 26-1 DAC 管脚说明

名称	型号类型	注释
VDDA	输入, 模拟电源	模拟电源
VSSA	输入, 模拟电源地	模拟电源的地线
DAC_OUTx	模拟输出信号	DAC 通道 x 的模拟输出 ($x = 1, 2$)

注：

- 当 DAC 通道使能时，相应的 GPIO 管脚会自动连接到 DAC 的模拟输出 DAC_OUTx。
- 为了防止额外的功耗和避免寄生的干扰，应当在使能之前设置管脚为模拟输入。

26.3 功能描述

26.3.1 DAC 通道的使能

置位 **DAC_CR** 控制寄存器的 **ENx** 位，即可开始对 **DAC** 通道 **x** (**x=1, 2** 下同) 模拟电路进行供电。经过一段启动时间 **t_{wakeup}** 后，**DAC** 通道 **x** 即被使能。

DAC 通道 **x** 的数字电路的正常工作不受该位控制，**ENx** 位只是使能通道 **x** 的模拟电路。

26.3.2 使能 **DAC** 输出缓存

DAC 集成了 2 个输出缓存，可以用来减少输出阻抗，实现无需外部运放即可对外部负载进行直接驱动。每个 **DAC** 通道的输出缓存可以通过设置 **DAC_CR** 寄存器的 **BOFFx** 位来使能或者关闭。

26.3.3 **DAC** 输出电压

数字输入经过 **DAC** 转换成模拟电压输出 **DAC_OUTx**，其范围为 0 到 **V_{REF}**，满足下面的公式：

$$\text{DAC_OUTx} = \text{V}_{\text{REF}} \times (\text{DAC_DORx} / 4096)$$

26.3.4 **DAC** 触发源的选择

置位 **DAC_CR** 寄存器的 **TENx** 位，则 **DAC** 转换可以由某外部事件触发（定时计数器，外部中断线）。共有 5 个外部事件触发和 1 个软件触发，通过配置寄存器 **DAC_CR** 的 **TSELx[2: 0]** 位，选择其中 1 个事件进行触发转换。

表 26-2 外部触发

触发源	类型	TSEL[2: 0]
定时器 1 TRGO 事件	来自片上定时器的内部信号	000
定时器 3 TRGO 事件		001
无效		010
无效		011
定时器 2 TRGO 事件		100
定时器 4 TRGO 事件		101
EXTI 线 9	外部管脚	110
SWTRIG (软件触发)	软件控制位	111

如果选择硬件触发（**TSEL[2: 0]** 设置不为 111），每次 **DAC** 接口检测到来自选中定时器 **TRGO** 输出，或者外部中断线 9 的上升沿，数据保持寄存器 **DAC_DHRx** 中的数据将会被传送到数据输出寄存器 **DAC_DORx** 中。在 3 个 APB1 时钟周期后，寄存器 **DAC_DORx** 被更新为最新值。

如果选择软件触发（TSEL[2: 0] 设置为 111），当 SWTRIG 位置‘1’，转换开始。

当寄存器 DAC_DORx 从寄存器 DAC_DHRx 取得数据后，SWTRIG 位由硬件自动清 0。

注：

- TSELx[2: 0] 位在 ENx 位被置位后不能改变。
- 如果选中软件触发，则数据从寄存器 DAC_DHRx 载入寄存器 DAC_DORx 只需要 1 个 APB1 时钟周期。

26.3.5 DMA 请求

任一 DAC 通道都具有 DMA 功能，可分别用于各自通道的 DMA 请求。当寄存器 DAC_DHRx 的数据被传到寄存器 DAC_DORx 时，如果 DMA_ENx 位置位，则产生 1 个 DMA 请求。

在双 DAC 通道模式下，可以置位 2 个通道的 DMA_ENx 位，产生 2 个 DMA 请求。也可以选择置位 1 个 DMA_ENx 位，只进行 1 个 DMA 请求。

注：

- DAC 的 DMA 请求不会累计，因此如果第 2 个外部触发发生在响应第 1 个外部触发之前，则第 2 个 DMA 请求无效，也不会报错。

26.3.6 DAC 数据格式

数据写入数据保持寄存器的格式根据单双通道模式各有不同，分别说明如下：

DAC 单通道模式下数据格式分以下 3 种情况：

- 8 位数据右对齐：需将数据写入寄存器 DAC_DHR8Rx 的 [7: 0] 位（实际是存入数据保持寄存器 DAC_DHRx[11: 4] 位）
- 12 位数据左对齐：需将数据写入寄存器 DAC_DHR12Lx 的 [15: 4] 位（实际是存入数据保持寄存器 DAC_DHRx[11: 0] 位）
- 12 位数据右对齐：需将数据写入寄存器 DAC_DHR12Rx 的 [11: 0] 位（实际是存入数据保持寄存器 DAC_DHRx[11: 0] 位）

经过相应的移位操作，写入 DAC_DHR8Rx, DAC_DHR12Lx, DAC_DHR12Rx 寄存器的数据会被转存到内部数据保持寄存器 DHRx 中。DHRx 的内容可配置成三种方式传送给 DORx 数据输出寄存器。一是被自动地传送到 DORx 寄存器，二是通过软件触发被传送到 DORx 寄存器，三是由外部事件触发被传送到 DORx 寄存器。

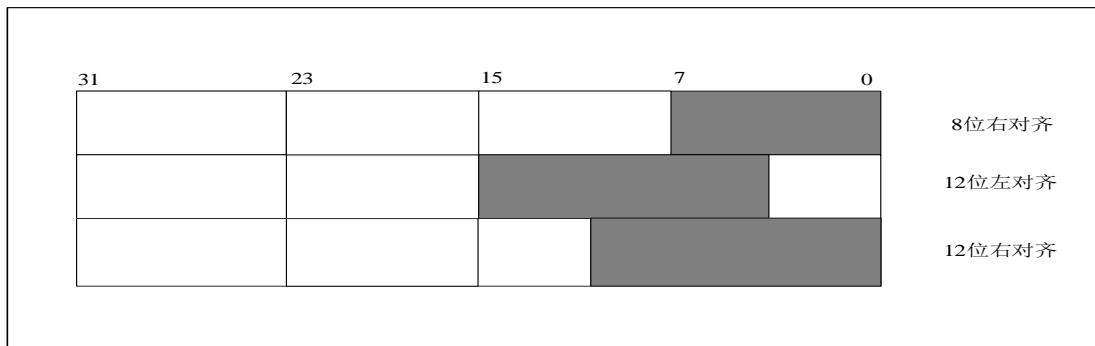


图 26-2 DAC 单通道模式的数据保持寄存器的不同数据格式

DAC 双通道模式下数据格式分以下 3 种情况：

- 8 位数据右对齐：将 DAC 通道 1 数据写入寄存器 DAC_DHR8RD 的 [7: 0] 位（实际是存入数据保持寄存器 DAC_DHR1[11: 4] 位），将 DAC 通道 2 数据写入寄存器 DAC_DHR8RD 的 [15: 8] 位（实际是存入数据保持寄存器 DAC_DHR2[11: 4] 位）
- 12 位数据左对齐：将 DAC 通道 1 数据写入寄存器 DAC_DHR12LD 的 [15: 4] 位（实际是存入数据保持寄存器 DAC_DHR1[11: 0] 位），将 DAC 通道 2 数据写入寄存器 DAC_DHR12LD 的 [31: 20] 位（实际是存入数据保持寄存器 DAC_DHR2[11: 0] 位）
- 12 位数据右对齐：将 DAC 通道 1 数据写入寄存器 DAC_DHR12RD 的 [11: 0] 位（实际是存入数据保持寄存器 DAC_DHR1[11: 0] 位），将 DAC 通道 2 数据写入寄存器 DAC_DHR12RD 的 [27: 16] 位（实际是存入数据保持寄存器 DAC_DHR2[11: 0] 位）

经过相应的移位操作，写入 DAC_DHR8RD, DAC_DHR12LD, DAC_DHR12RD 寄存器的数据会被转存到内部数据保持寄存器 DHRx 中。随后，DHR1 和 DHR2 的内容可配置成三种方式传送给 DORx 数据输出寄存器。一是被自动地传送到 DORx 寄存器，二是通过软件触发被传送到 DORx 寄存器，三是由外部事件触发被传送到 DORx 寄存器。

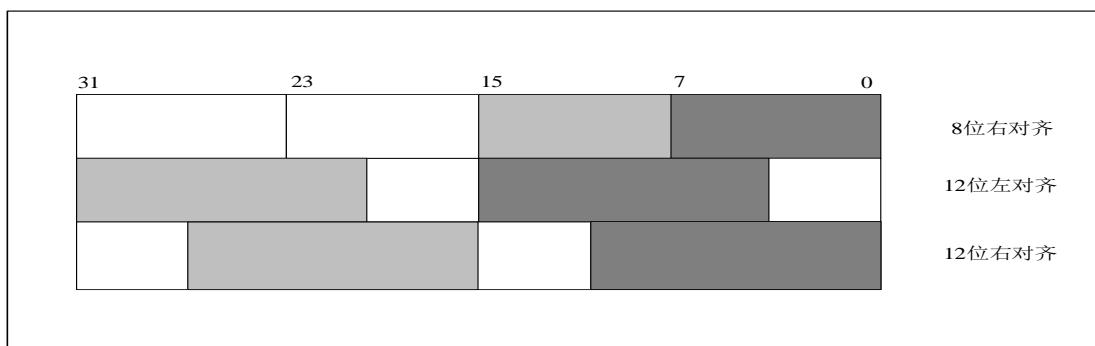


图 26-3 DAC 双通道模式的数据保持寄存器的不同数据格式

26.3.7 带三角波生成的 DAC 转换

通过配置 DAC_CR 寄存器的 WAVE_x[1: 0] 位为‘10’，可以选中 DAC 的三角波生成功能，使得在直流信号或者缓慢变化的输出信号上加上 1 个小幅度的三角波。三角波的幅度可以通过设置 DAC_CR 寄存器的 MAMP_x[3: 0] 位来选择。内部的三角波计数器在每次触发事件之后，再经过 3 个 APB1 时钟周

期会累加 1。将计数器的值与寄存器 **DAC_DHRx** 的值相加后丢弃掉溢出位写入寄存器 **DAC_DORx**。在三角波计数器的值小于 **MAMP[3: 0]** 位定义的最大幅度时，三角波计数器才会累加。当累加到设置的最大幅度时，计数器开始累减至 0，然后再开始累加，如此循环往复。将 **DAC_CR** 寄存器的 **WAVE[1: 0]** 位清零，可以取消三角波的生成。

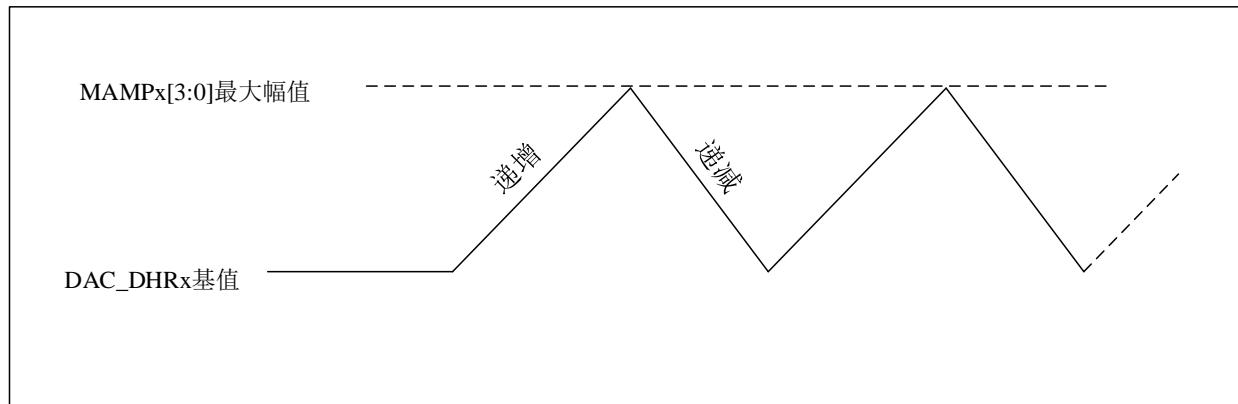


图 26-4 DAC 三角波生成

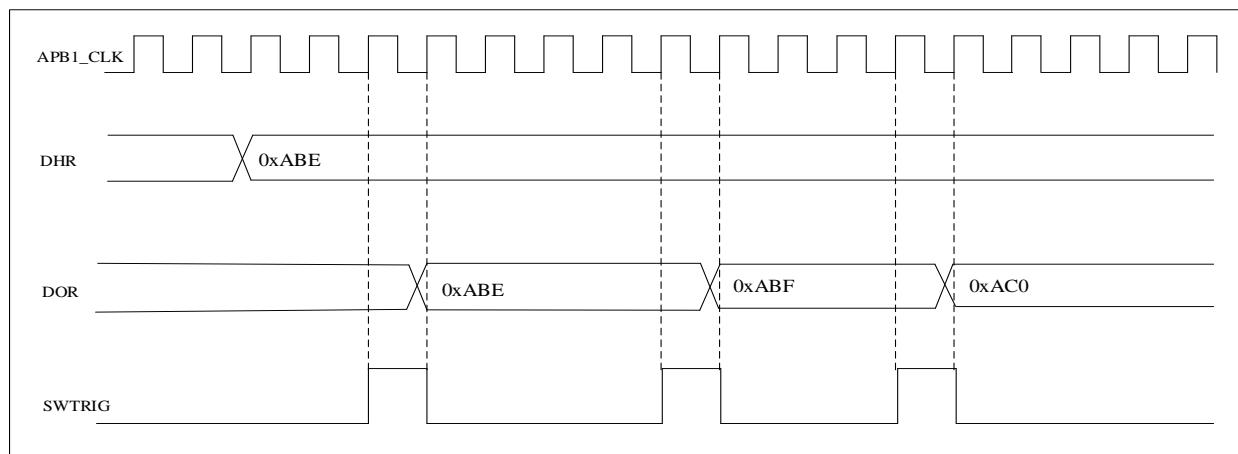


图 26-5 带三角波生成的 DAC 转换（使能软件触发）

注：

- 必须使能 DAC 触发，才能产生三角波，即置位 **DAC_CR** 寄存器的 **TENx** 位。
- 必须在使能 DAC 之前设置 **MAMP[3: 0]**，否则其值不能修改。

26.3.8 带噪声波生成的 DAC 转换

可以通过设置 **DAC_CR** 寄存器 **WAVE[1: 0]** 位为‘01’，来选中 DAC 噪声波形生成功能。利用线性反馈移位寄存器 LFSR（Linear Feedback Shift Register）来产生幅度变化的伪噪声。根据下图的移位寄存器算法，在每次触发事件后，再经过 3 个 APB1 时钟周期，更新该寄存器的值。寄存器 LFSR 的复位值为 0xAAA。

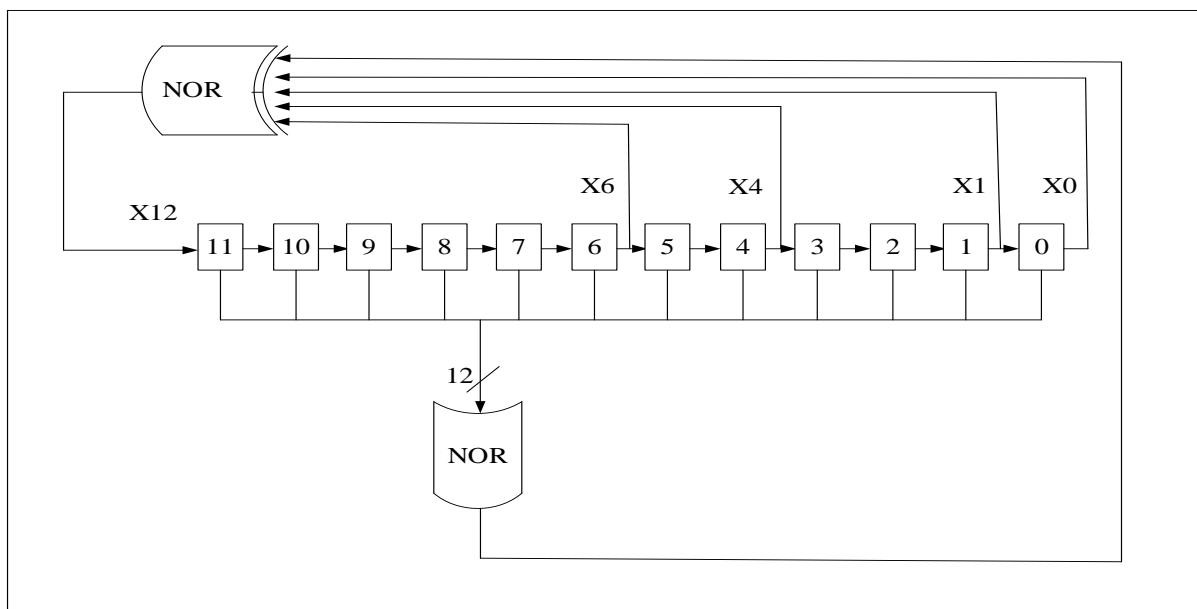


图 26-6 DAC LFSR 寄存器算法

设置寄存器 `DAC_CR` 的 `MAMPx[3: 0]` 位，可以屏蔽部分或者全部 LFSR 的数据。经屏蔽的 LFSR 值与 `DAC_DHRx` 的数值相加，去掉溢出位后，被写入寄存器 `DAC_DORx`。如果寄存器 `LFSR` 值为 `0x000`，则会重新注入‘1’给移位寄存器第 0 位以保证继续产生噪声波（防锁定机制）。将 `WAVEx[1: 0]` 位清零，可以取消 LFSR 波形的生成。

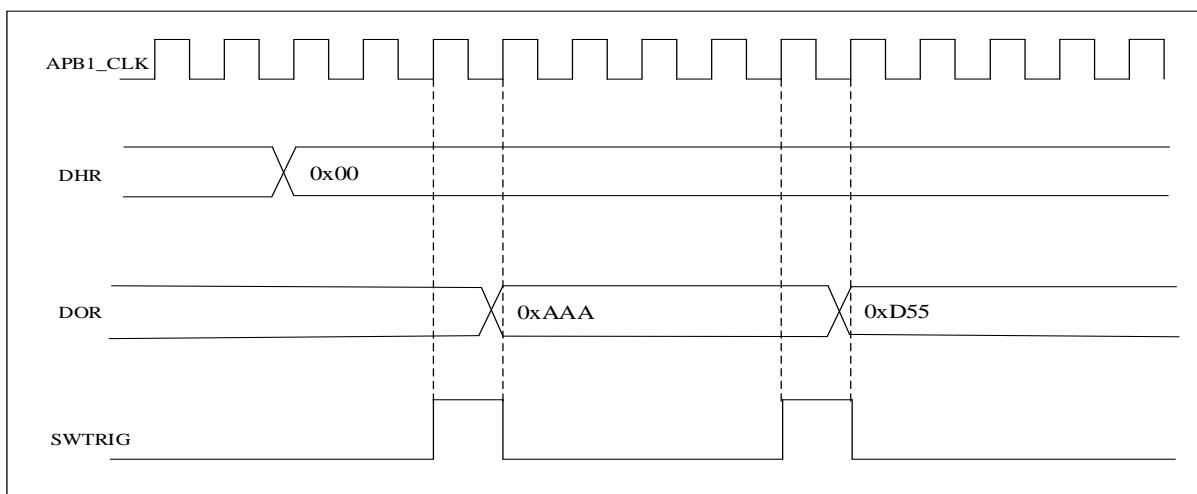


图 26-7 带 LFSR 波形生成的 DAC 转换（使能软件触发）

注：

- 必须使能 DAC 触发才能产生噪声，即置位 `DAC_CR` 寄存器的 `TENx` 位。

26.3.9 DAC 转换

DAC 通道 x 输出的数据不可以直接写入数据输出寄存器 `DAC_DORx`，而必须通过内部的数据保持寄存器 `DAC_DHRx` 载入（实际上是将数据先写入寄存器 `DAC_DHR8Rx`，`DAC_DHR12Lx`，`DAC_DHR12Rx`，`DAC_DHR8RD`，`DAC_DHR12LD` 或者 `DAC_DHR12RD` 后，再载入到数据输出

寄存器 `DAC_DORx` 中)。

当数据从寄存器 `DAC_DHRx` 载入寄存器 `DAC_DORx` 后, 经过一段时间 $t_{settling}$ (模拟 DAC 稳定时间), 输出即有效, 这段时间的长短根据电源电压和模拟输出负载的不同而有所不同。

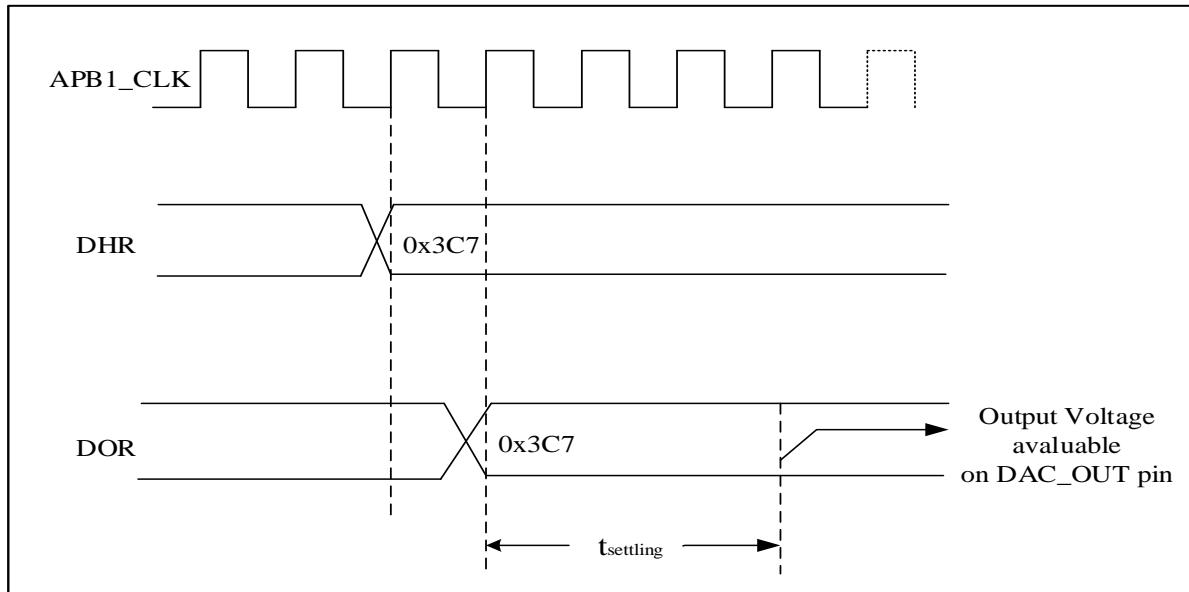


图 26-8 $TEN=0$ 触发关闭时 DAC 转换的时间示意图

26.3.10 DAC 双通道转换模式

DAC 集成了 3 个供双 DAC 模式使用的寄存器: `DHR8RD`、`DHR12RD` 和 `DHR12LD`, 能够更高效地利用总线带宽, 在需要 2 个 DAC 通道同时工作的情况下, 只需要访问 1 个寄存器即可完成同时驱动 2 个 DAC 通道的任务。

对于双 DAC 通道转换和这些专用寄存器, 共有 11 种不同配置组合的转换模式可供选择使用 (每种模式配置完成后均需将双 DAC 通道转换数据载入所需的 DHR 寄存器 (`DHR12RD`, `DHR12LD` 或者 `DHR8RD`) 中)。

- 统一触发的不带波形生成的转换 (`TENx` 置位, `TSELx` 为相同值)

当触发事件发生, 寄存器 `DHR1` 和 `DHR2` 的值在 3 个 APB1 时钟周期后分别传入寄存器 `DAC_DOR1` 和 `DAC_DOR2`。

- 统一触发的带相同三角波生成的转换 (`TENx` 置位, `TSELx` 为相同值, `WAVEx='1x'`, `MAMPx` 为相同值)

当触发事件发生, 寄存器 `DHR1` 的值加上相同最大幅值的三角波计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 `DAC_DOR1`, 然后更新通道 1 三角波计数器。同时, 寄存器 `DHR2` 的值加上相同最大幅值的三角波计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 `DAC_DOR2`, 然后更新通道 2 三角波计数器。

- 统一触发的带不同三角波生成的转换 (`TENx` 置位, `TSELx` 为相同值, `WAVEx='1x'`, `MAMPx`

为不同值)

当触发事件发生，寄存器 DHR1 的值加上 MAMP1[3: 0] 所设最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1，然后更新通道 1 三角波计数器。同时，寄存器 DHR2 的值加上 MAMP2[3: 0] 所设最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2，然后更新通道 2 三角波计数器。

- 统一触发的带相同 LFSR 生成的转换 (TENx 置位, TSELx 为相同值, WAVE_x=‘01’, MAMP_x 为相同值)

当触发事件发生，寄存器 DHR1 的值加上带相同屏蔽的 LFSR1 计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1，然后更新 LFSR1 计数器。同时，寄存器 DHR2 的值加上带相同屏蔽的 LFSR2 计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2，然后更新 LFSR2 计数器。

- 统一触发的带不同 LFSR 生成的转换 (TENx 置位, TSELx 为相同值, WAVE_x=‘01’, MAMP_x 为不同值)

当触发事件发生，寄存器 DHR1 的值加上带 MAMP1[3: 0] 所设屏蔽的 LFSR1 计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1，然后更新 LFSR1 计数器。同时，寄存器 DHR2 的值加上带 MAMP1[3: 0] 所设屏蔽的 LFSR2 计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2，然后更新 LFSR2 计数器。

- 统一软件启动的转换

该配置下，1 个 APB1 时钟周期后，寄存器 DHR1 和 DHR2 的值即被分别传入寄存器 DAC_DOR1 和 DAC_DOR2。

- 分别触发的不带波形生成的转换 (TENx 置位, TSELx 为不同值以选择不同触发源)

当 DAC 通道 1 触发事件发生，寄存器 DHR1 的值在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1。当 DAC 通道 2 触发事件发生，寄存器 DHR2 的值在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2。

- 分别触发的带相同三角波生成的转换 (TENx 置位, TSELx 为不同值, WAVE_x=‘1x’, MAMP_x 为相同值)

当 DAC 通道 1 触发事件发生，寄存器 DHR1 的值加上相同最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1，然后更新 DAC 通道 1 三角波计数器。当 DAC 通道 2 触发事件发生，寄存器 DHR2 的值加上相同最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2，然后更新 DAC 通道 2 三角波计数器。

- 分别触发的带不同三角波生成的转换 (TENx 置位, TSELx 为不同值, WAVE_x=‘1x’, MAMP_x 为不同值)

当 DAC 通道 1 触发事件发生，寄存器 DHR1 的值加上 MAMP1[3: 0] 所设最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1，然后更新 DAC 通道 1 三角波计数器。当 DAC 通道 2 触发事件发生，寄存器 DHR2 的值加上 MAMP2[3: 0] 所设最大幅值的三角波计数器值，其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2，然后更新 DAC 通道 2 三角波计数器。

- 分别触发的带相同 LFSR 生成的转换 (TENx 置位, TSELx 为不同值, WAVE_x=‘01’, MAMP_x 为相同的 LFSR 屏蔽值)

当 DAC 通道 1 触发事件发生, 寄存器 DHR1 的值加上带相同屏蔽的 LFSR1 计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1 , 然后更新 LFSR1 计数器。当 DAC 通道 2 触发事件发生, 寄存器 DHR2 的值加上带相同屏蔽的 LFSR2 计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2 , 然后更新 LFSR2 计数器。

- 分别触发的带不同 LFSR 生成的转换 (TENx 置位, TSELx 为不同值, WAVE_x=‘01’, MAMP_x 为不同值)

当 DAC 通道 1 触发事件发生, 寄存器 DHR1 的值加上带 MAMP1[3: 0] 所设屏蔽的 LFSR1 计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1, 然后更新 LFSR1 计数器。当 DAC 通道 2 触发事件发生, 寄存器 DHR2 的值加上带 MAMP2[3: 0] 所设屏蔽的 LFSR2 计数器值, 其和在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR2, 然后更新 LFSR2 计数器。

这些转换模式在只使用 1 个 DAC 通道的情况下仍然可用。

26.4 寄存器描述

表 26-3 DAC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	DAC_CR	DAC 控制寄存器	0x00000000
0x04	DAC_SWTRIGR	DAC 软件触发寄存器	0x00000000
0x08	DAC_DHR12R1	DAC 通道 1 的 12 位右对齐数据保持寄存器	0x00000000
0x0C	DAC_DHR12L1	DAC 通道 1 的 12 位左对齐数据保持寄存器	0x00000000
0x10	DAC_DHR8R1	DAC 通道 1 的 8 位右对齐数据保持寄存器	0x00000000
0x14	DAC_DHR12R2	DAC 通道 2 的 12 位右对齐数据保持寄存器	0x00000000
0x18	DAC_DHR12L2	DAC 通道 2 的 12 位左对齐数据保持寄存器	0x00000000
0x1C	DAC_DHR8R2	DAC 通道 2 的 8 位右对齐数据保持寄存器	0x00000000
0x20	DAC_DHR12RD	DAC 双通道的 12 位右对齐数据保持寄存器	0x00000000
0x24	DAC_DHR12LD	DAC 双通道的 12 位左对齐数据保持寄存器	0x00000000
0x28	DAC_DHR8RD	DAC 双通道的 8 位右对齐数据保持寄存器	0x00000000
0x2C	DAC_DOR1	DAC 通道 1 数据输出寄存器	0x00000000
0x30	DAC_DOR2	DAC 通道 2 数据输出寄存器	0x00000000

26.4.1 DAC_CR 控制寄存器

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.		DMA_E N2	MAMP2				WAVE2		TSEL2			TEN2	BOFF2	EN2	

			rw	rw				rw		rw			rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DMA_E_N1	MAMP1				WAVE1		TSEL1			TEN1	BOFF1	EN1
			rw	rw				rw		rw			rw	rw	rw

Bit	Field	Description
31: 29	Reserved	保留, 必须保持复位值
28	DMA_EN2	<p>DAC 通道 2 DMA 使能 (DAC Channel 2 DMA Enable)</p> <p>该位由软件置位和清除</p> <p>0: 禁止 DAC 通道 2 的 DMA 功能</p> <p>1: 使能 DAC 通道 2 的 DMA 功能</p>
27: 24	MAMP2	<p>DAC 通道 2 屏蔽 / 幅值选择器 (DAC Channel 2 Mask / Amplitude Selector)</p> <p>该位由软件置位, 用来在产生噪声时选择屏蔽位, 或在产生三角波时选择波形的幅值</p> <p>0000: 不屏蔽 LSFR 位 0 / 三角波幅值等于 1</p> <p>0001: 不屏蔽 LSFR 位 [1: 0] / 三角波幅值等于 3</p> <p>0010: 不屏蔽 LSFR 位 [2: 0] / 三角波幅值等于 7</p> <p>0011: 不屏蔽 LSFR 位 [3: 0] / 三角波幅值等于 15</p> <p>0100: 不屏蔽 LSFR 位 [4: 0] / 三角波幅值等于 31</p> <p>0101: 不屏蔽 LSFR 位 [5: 0] / 三角波幅值等于 63</p> <p>0110: 不屏蔽 LSFR 位 [6: 0] / 三角波幅值等于 127</p> <p>0111: 不屏蔽 LSFR 位 [7: 0] / 三角波幅值等于 255</p> <p>1000: 不屏蔽 LSFR 位 [8: 0] / 三角波幅值等于 511</p> <p>1001: 不屏蔽 LSFR 位 [9: 0] / 三角波幅值等于 1023</p> <p>1010: 不屏蔽 LSFR 位 [10: 0] / 三角波幅值等于 2047</p> <p>1011 ~ 1111: 不屏蔽 LSFR 位 [11: 0] / 三角波幅值等于 4095</p>
23: 22	WAVE2	<p>DAC 通道 2 噪声 / 三角波生成使能 (DAC Channel 2 Noise / Triangle Wave Generation Enable)</p> <p>该位由软件置位和清除</p> <p>00: 禁止波形生成功能</p> <p>01: 生成噪声波形</p> <p>1x: 生成三角波形</p>

Bit	Field	Description
21: 19	TSEL2	<p>DAC 通道 2 触发选择 (DAC Channel 2 Trigger Selection)</p> <p>该位用于选择 DAC 通道 2 的外部触发源</p> <p>000: TIM1 TRGO 事件 001: TIM3 TRGO 事件 010: 无效 011: 无效 100: TIM2 TRGO 事件 101: TIM4 TRGO 事件 110: 外部中断线 9 111: 软件触发</p> <p>注: 该位只能在 TEN2 = 1 (DAC 通道 2 触发使能) 时设置</p>
18	TEN2	<p>DAC 通道 2 触发使能 (DAC Channel 2 Trigger Enable)</p> <p>该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 2 的触发</p> <p>0: 不需要硬件触发或软件触发, 写入寄存器 DAC_DHRx 的数据 在 1 个 APB1 时钟周期后传入寄存器 DAC_DORx (自动传输模式)</p> <p>1: 使能 DAC 通道 2 触发(硬件触发或软件触发), 写入寄存器 DAC_DHRx 的数据, 在 触发后 3 个 APB1 时钟周期后传入寄存器 DAC_DORx</p> <p>注: 如果选择软件触发, 写入寄存器 DAC_DHRx 的数据只要 1 个 APB1 时钟周期就可以传入寄存器 DAC_DORx</p>
17	BOFF2	<p>DAC 通道 2 输出缓存禁止 (DAC Channel 2 Output Buffer Disable)</p> <p>该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 2 的输出缓存</p> <p>0: 使能 DAC 通道 2 输出缓存 1: 禁止 DAC 通道 2 输出缓存</p>
16	EN2	<p>DAC 通道 2 使能 (DAC Channel 2 Enable)</p> <p>该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 2</p> <p>0: 禁止 DAC 通道 2 1: 使能 DAC 通道 2</p>
15: 13	Reserved	保留, 必须保持复位值
12	DMA_EN1	<p>DAC 通道 1 DMA 使能 (DAC Channel 1 DMA Enable)</p> <p>该位由软件置位和清除</p> <p>0: 禁止 DAC 通道 1 DMA 功能 1: 使能 DAC 通道 1 DMA 功能</p>

Bit	Field	Description
11: 8	MAMP1	<p>DAC 通道 1 屏蔽 / 幅值选择器 (DAC Channel 1 Mask / Amplitude Selector) 由软件置位该位, 用来在产生噪声时选择屏蔽位, 或在产生三角波时选择波形的幅值</p> <p>0000: 不屏蔽 LSFR 位 0 / 三角波幅值等于 1 0001: 不屏蔽 LSFR 位 [1: 0] / 三角波幅值等于 3 0010: 不屏蔽 LSFR 位 [2: 0] / 三角波幅值等于 7 0011: 不屏蔽 LSFR 位 [3: 0] / 三角波幅值等于 15 0100: 不屏蔽 LSFR 位 [4: 0] / 三角波幅值等于 31 0101: 不屏蔽 LSFR 位 [5: 0] / 三角波幅值等于 63 0110: 不屏蔽 LSFR 位 [6: 0] / 三角波幅值等于 127 0111: 不屏蔽 LSFR 位 [7: 0] / 三角波幅值等于 255 1000: 不屏蔽 LSFR 位 [8: 0] / 三角波幅值等于 511 1001: 不屏蔽 LSFR 位 [9: 0] / 三角波幅值等于 1023 1010: 不屏蔽 LSFR 位 [10: 0] / 三角波幅值等于 2047 1011 ~ 1111: 不屏蔽 LSFR 位 [11: 0] / 三角波幅值等于 4095</p>
7: 6	WAVE1	<p>DAC 通道 1 , 噪声 / 三角波生成使能 (DAC Channel 1 Noise Triangle Wave Generation Enable) 该位由软件置位和清除 00: 禁止波形生成功能 01: 产生噪声波形 1x: 产生三角波形</p>
5: 3	TSEL1	<p>DAC 通道 1 触发选择 (DAC Channel 1 Trigger Selection) 该位用于选择 DAC 通道 1 的外部触发源 000: TIM1 TRGO 事件 001: TIM3 TRGO 事件 010: 无效 011: 无效 100: TIM2 TRGO 事件 101: TIM4 TRGO 事件 110: 外部中断线 9 111: 软件触发 注: 该位只能在 TEN1 = 1 (DAC 通道 1 触发使能) 时设置</p>
2	TEN1	<p>DAC 通道 1 触发使能 (DAC Channel 1 Trigger Enable) 该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 1 的触发 0: 不需要硬件触发或软件触发, 写入寄存器 DAC_DHRx 的数据 在 1 个 APB1 时钟周期后传入寄存器 DAC_DORx (自动传输模式) 1: 使能 DAC 通道 1 触发 (硬件触发或软件触发), 写入寄存器 DAC_DHRx 的数据, 在 触发后 3 个 APB1 时钟周期后传入寄存器 DAC_DORx 注: 如果选择软件触发, 写入寄存器 DAC_DHRx 的数据只要 1 个 APB1 时钟周期就可以传入寄存器 DAC_DORx</p>

Bit	Field	Description
1	BOFF1	DAC 通道 1 输出缓存禁止 (DAC Channel 1 Output Buffer Disable) 该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 1 的输出 0: 使能 DAC 通道 1 输出缓存 1: 禁止 DAC 通道 1 输出缓存
0	EN1	DAC 通道 1 使能 (DAC Channel 1 Enable) 该位由软件置位和清除, 用来使能 / 禁止 DAC 通道 1 0: 禁止 DAC 通道 1 1: 使能 DAC 通道 1

26.4.2 DAC_SWTRIGR 软件触发寄存器

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.	DAC_PRE								Res.						SW_TRI
	W								Res.						G2
	W								W						G1
	W								W						W

Bit	Field	Description
31: 15	Reserved	保留, 必须保持复位值
14: 8	DAC_PRE	DAC 预分频 (DAC Prescaler) PCLK1 经 $2^{(n+1)}$ 分频后作为 DAC 时钟。n 为 DAC_PRE 的值
7: 2	Reserved	保留, 必须保持复位值
1	SW_TRIG2	DAC 通道 2 软件触发 (DAC Channel 2 Software Trigger) 该位由软件置位和清除, 用来使能 / 禁止软件触发 0: 禁止 DAC 通道 2 软件触发 1: 使能 DAC 通道 2 软件触发 注: 当寄存器 DAC_DHR2 的数据传入寄存器 DAC_DOR2 时, 该位由硬件置'0' (1 个 APB1 时钟周期后)
0	SW_TRIG1	DAC 通道 1 软件触发 (DAC Channel 1 Software Trigger) 该位由软件置位和清除, 用来使能 / 禁止软件触发 0: 禁止 DAC 通道 1 软件触发 1: 使能 DAC 通道 1 软件触发 注: 当寄存器 DAC_DHR1 的数据传入寄存器 DAC_DOR1 时, 该位由硬件置'0' (1 个 APB1 时钟周期后)

26.4.3 DAC_DHR12R1 通道 1 的 12 位右对齐数据保持寄存器

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.				DACC1DHR											
Res.				rw											

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	DACC1DHR	DAC 通道 1 的 12 位右对齐数据 (DAC Channel 1 12-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 12 位数据

26.4.4 DAC_DHR12L1 通道 1 的 12 位左对齐数据保持寄存器

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
DACC1DHR												Res.			
rw															

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 4	DACC1DHR	DAC 通道 1 的 12 位左对齐数据 (DAC Channel 1 12-bit Left Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 12 位数据
3: 0	Reserved	保留, 必须保持复位值

26.4.5 DAC_DHR8R1 通道 1 的 8 位右对齐数据保持寄存器

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														DACC1DHR	
rw														rw	

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7: 0	DACC1DHR	DAC 通道 1 的 8 位右对齐数据 (DAC Channel 1 8-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 8 位数据

26.4.6 DAC_DHR12R2 通道 2 的 12 位右对齐数据保持寄存器

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.														DACC2DHR	
rw														rw	

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	DACC2DHR	DAC 通道 2 的 12 位右对齐数据 (DAC Channel 2 12-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 12 位数据

26.4.7 DAC_DHR12L2 通道 2 的 12 位左对齐数据保持寄存器

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.														DACC2DHR	
rw														Res.	

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 4	DACC2DHR	DAC 通道 2 的 12 位左对齐数据 (DAC Channel 2 12-bit Left Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 12 位数据
3: 0	Reserved	保留, 必须保持复位值

26.4.8 DAC_DHR8R2 通道 2 的 8 位右对齐数据保持寄存器

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.								DACC2DHR							
rw															

Bit	Field	Description
31: 8	Reserved	保留, 必须保持复位值
7: 0	DACC2DHR	DAC 通道 2 的 8 位右对齐数据 (DAC Channel 2 8-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 8 位数据

26.4.9 DAC_DHR12LD 双通道的 12 位左对齐数据保持寄存器

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
DACC2DHR														Res.	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR														Res.	
rw															

Bit	Field	Description
31: 20	DACC2DHR	DAC 通道 2 的 12 位左对齐数据 (DAC Channel 2 12-bit Left Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 12 位数据
19: 16	Reserved	保留, 必须保持复位值
15: 4	DACC1DHR	DAC 通道 1 的 12 位左对齐数据 (DAC Channel 1 12-bit Left Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 12 位数据
3: 0	Reserved	保留, 必须保持复位值

26.4.10 DAC_DHR12RD 双通道的 12 位右对齐数据保持寄存器

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.				DACC2DHR											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC1DHR											
rw															

Bit	Field	Description
31: 28	Reserved	保留, 必须保持复位值
27: 16	DACC2DHR	DAC 通道 2 的 12 位右对齐数据 (DAC Channel 2 12-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 12 位数据
15: 12	Reserved	保留, 必须保持复位值
11: 0	DACC1DHR	DAC 通道 1 的 12 位右对齐数据 (DAC Channel 1 12-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 12 位数据

26.4.11 DAC_DHR8RD 双通道的 8 位右对齐数据保持寄存器

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
DACC2DHR								DACC1DHR							
rw								rw							

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值
15: 8	DACC2DHR	DAC 通道 2 的 8 位右对齐数据 (DAC Channel 2 8-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 2 的 8 位数据
7: 0	DACC1DHR	DAC 通道 1 的 8 位右对齐数据 (DAC Channel 1 8-bit Right Aligned Data) 该位由软件写入, 表示 DAC 通道 1 的 8 位数据

26.4.12 DAC_DOR1 通道 1 数据输出寄存器

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.				DACC1DOR											
r															

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	DACC1DOR	DAC 通道 1 输出数据 (DAC Channel 1 Data Output) 该位由软件写入, 表示 DAC 通道 1 的输出数据

26.4.13 DAC_DOR2 通道 2 数据输出寄存器

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.				DACC2DOR											
r															

Bit	Field	Description
31: 12	Reserved	保留, 必须保持复位值
11: 0	DACC2DOR	DAC 通道 2 输出数据 (DAC Channel 2 Data Output) 该位由软件写入, 表示 DAC 通道 2 的输出数据

27 COMP 比较器

27.1 简介

芯片内嵌 2 个通用比较器 (COMP1, 2), 每个可独立使用。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。

27.2 功能框图

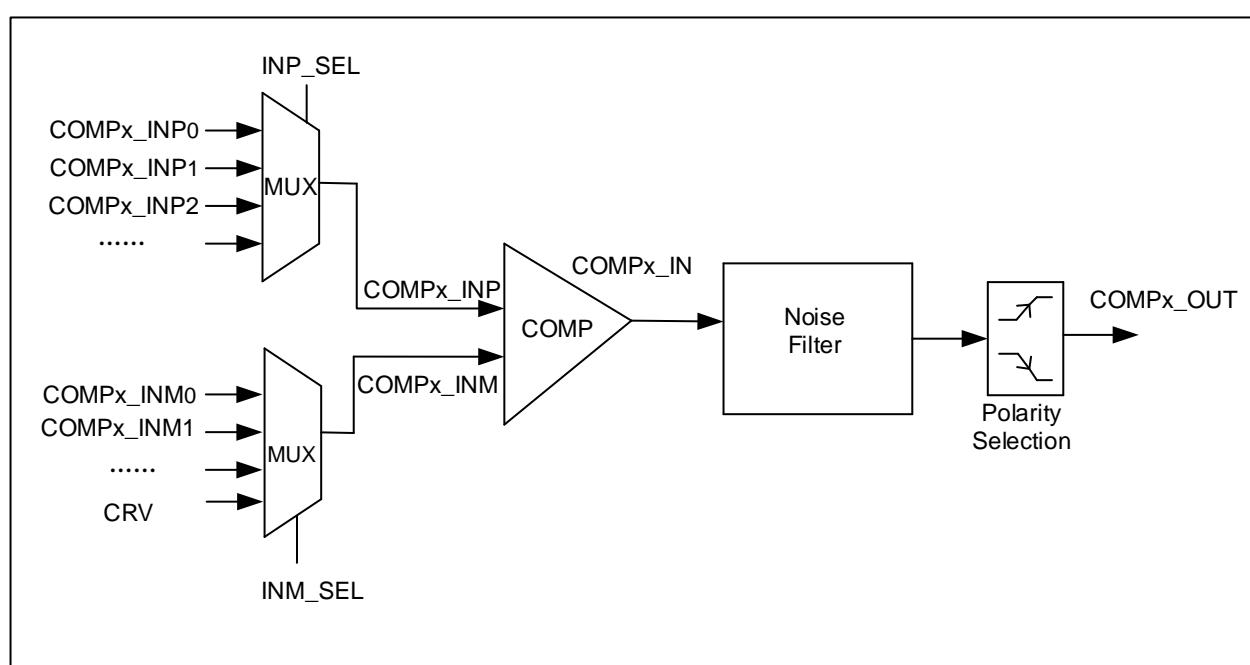


图 27-1 比较器框图

27.3 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 4 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。

27.4 功能描述

27.4.1 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。

27.4.2 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。

27.4.3 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。

27.4.4 比较器通道选择

每个 COMP 有四个正相输入和四个反相输入通道，正相输入可从四个外部引脚之间选取，反相输入可从三个外部引脚或者 CRV 电压分压值选取。CRV 的电压可选择 VDDA 或者内部 1.2V 基准电压(V_{REFINT})的分压。

COMP 可以在普通工作模式下通过软件选择输入通道，也可以在轮询工作模式下通过硬件轮询的方式分时监测多个通道的比较结果，从逻辑上类似于多个比较器同时工作。

在普通工作模式下，比较器比较所选择的 INP 和 INM 端口上的信号，具体流程如下：

配置 COMPx_CSR 寄存器的 INP_SEL 位和 INM_SEL 位，选择所要比较的信号；

配置 COMPx_CSR 寄存器的 EN 位，比较器开始上电工作；

比较的结果存放于 COMPx_CSR 寄存器的 OUT 位。

另外，当 COMP 的 INM_SEL 选择 CRV 时，需要配置 COMP_CRV 寄存器的 CRV_SEL 位，然后将 CRV_EN 置位（在上面步骤 2 之前）。

在轮询工作模式下，COMP 的 INP 端口上的信号将会周期性的轮询变化，而 INM 端口的信号可以配

置 COMPx_POLL 寄存器的 FIXN 位选择跟随 INP 端口变化或者由 COMPx_CSR 的 INM_SEL 位来配置。需要注意的是，当启动轮询功能以后，COMPx_CSR 的 INP_SEL 位将失去作用，同样的，如果 COMPx_POLL 寄存器的 FIXN 位选择 INM 端口跟随 INP 轮询变化，COMPx_CSR 的 INM_SEL 位也将失去作用。具体流程如下：

- 配置 COMPx_POLL 寄存器的 PERIOD 位来选择所需要的轮询等待周期；
- 配置 COMPx_POLL 寄存器的 FIXN 位来决定 INM 端口的信号是否跟随 INP 端口轮询变化；
- 配置 COMPx_POLL 寄存器的 POLL_CH 位决定所需要轮询的通道是 1/2/3 或者 1/2；
- 配置 COMPx_POLL 寄存器的 POLL_EN 位，启动轮询功能；
- 配置 COMPx_CSR 寄存器的 EN 位，比较器开始上电工作；
- 轮询比较的结果存放于 COMPx_POLL 寄存器的 POUT 位，其中 POUT[2]、POUT[1]、POUT[0] 位分别存放轮询通道 3/2/1 的比较结果。

27.4.5 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。

27.4.6 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。

27.4.7 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）

27.4.8 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)。

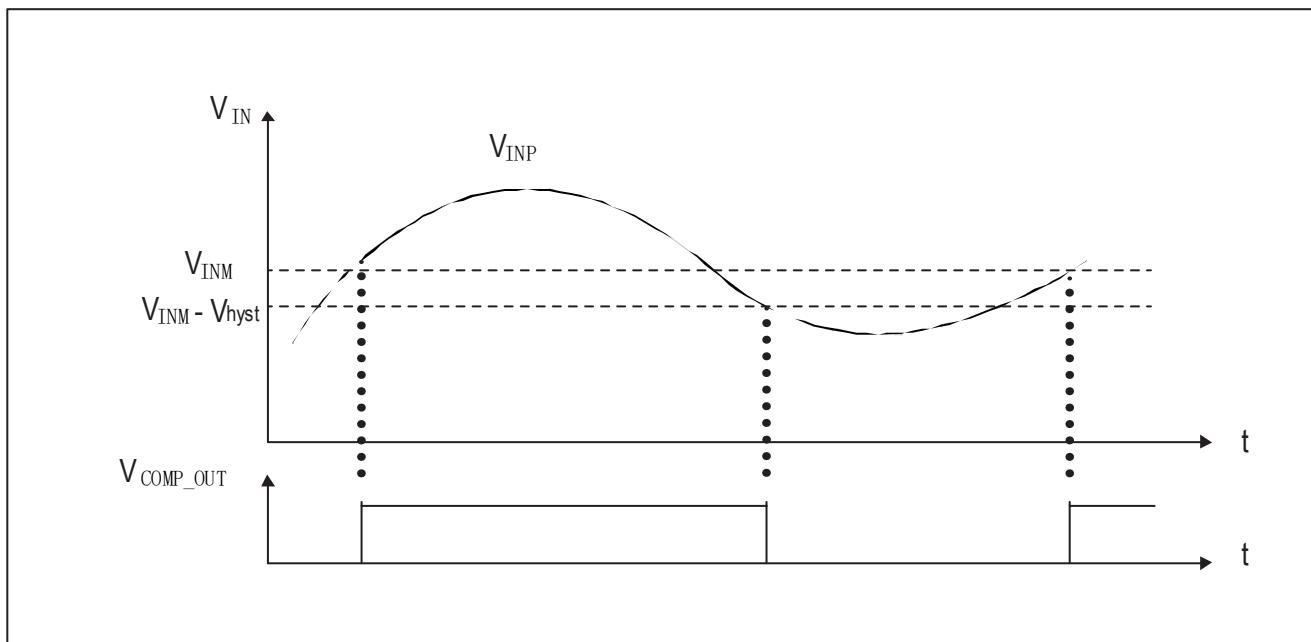


图 27-2 比较器的迟滞

27.5 比较器寄存器描述

表 27-1 COMP 寄存器概览

Offset	Acronym	Register Name	Reset
0x0C, 0x10	COMPx_CSR ($x=1, 2$)	比较器 x ($x=1, 2$) 控制和状态寄存器	0x00000000
0x18	COMP_CRV	比较器外部参考电压寄存器	0x00000000
0x1C, 0x20	COMPx_POLL ($x=1, 2$)	比较器 x ($x=1, 2$) 轮询寄存器	0x00000000

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

偏移地址: 0x0C, 0x10

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LOCK	OUT	Res.										OFLT			HYST	
rw	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		Res.	INM_SEL		MODE			Res.	EN
rw	rw	rw				rw	rw		rw	rw		rw			rw	rw

Bit	Field	Description
31	LOCK	<p>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。</p> <p>1: COMPx_CSR 只读 0: COMPx_CSR 可读可写</p>
30	OUT	<p>比较器 x 输出 (Comparator x Output) 反映比较器 x 输出状态。</p> <p>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)</p>
29 : 21	Reserved	始终读为 0。
20 : 18	OFLT	<p>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, 否则认为是有效信号。</p> <p>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: 无滤波</p>
17 : 16	HYST	<p>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。</p> <p>11: 90mV 10: 30mV 01: 15mV 00: 0mV</p>
15	POL	<p>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。</p> <p>1: 输出端反相输出 0: 输出端同相输出</p>
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) 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 0100: 定时器 8 刹车输入 0110: 定时器 1 Ocrefclear 输入 0111: 定时器 1 输入捕捉 1 1000: 定时器 2 输入捕捉 4 1001: 定时器 2 OCrefclear 输入 1010: 定时器 3 输入捕捉 1 1011: 定时器 3 OCrefclear 输入 其他: 无选择</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) 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 (PA0), 即正向输入端通道 0 01: COMP1_INP1 (PA1), 即正向输入端通道 1 10: COMP1_INP2 (PA2), 即正向输入端通道 2 11: COMP1_INP3 (PA3), 即正向输入端通道 3</p> <p>比较器 2:</p> <p>00: COMP2_INP0 (PA0), 即正向输入端通道 0 01: COMP2_INP1 (PA1), 即正向输入端通道 1 10: COMP2_INP2 (PA2), 即正向输入端通道 2 11: COMP2_INP3 (PA3), 即正向输入端通道 3</p>
6	Reserved	始终读为 0。
5: 4	INM_SEL	<p>比较器 x 反相输入选择 (Comparator x Inverting Input Selection) 这些位用于选择连接到比较器 x 反相输入端的信号源。</p> <p>比较器 1:</p> <p>00: COMP1_INM0 (PA4), 即反向输入端通道 0 01: COMP1_INM1 (PA5), 即反向输入端通道 1 10: COMP1_INM2 (PA0), 即反向输入端通道 2 11: COMP1_INM3 (CRV), 即反向输入端通道 3</p> <p>比较器 2:</p> <p>00: COMP2_INM0 (PA4), 即反向输入端通道 0 01: COMP2_INM1 (PA5), 即反向输入端通道 1 10: COMP2_INM2 (PA2), 即反向输入端通道 2 11: COMP2_INM3 (CRV), 即反向输入端通道 3</p>
3: 2	MODE	<p>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。</p> <p>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</p>

Bit	Field	Description
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

27.5.2 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_E N	CRV_SEL			
										rw	rw	rw			

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5	CRV_SRC	比较器参考电压源选择 (Comparator Reference Voltage Source Select) 0: VREFINT (内部电压) 1: VDDA (外部电压)
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}

27.5.3 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	POLLC_H	POLLE_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	<p>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。</p> <p>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</p>
3	Reserved	始终读为 0。
2	FIXN	<p>轮询通道反相输入端固定 (Polling Inverting Input Fix)</p> <p>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。</p>
1	POLL_CH	<p>比较器轮询通道 (Comparator Polling Channel)</p> <p>1: 轮询通道 1/2/3 0: 轮询通道 1/2</p>
0	POLL_EN	<p>比较器轮询模式使能 (Comparator Polling Enable)</p> <p>1: 比较器轮询模式使能 0: 比较器轮询模式禁止</p>

28 CRC 循环冗余校验计算单元

28.1 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。

28.2 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

28.3 功能描述

28.3.1 功能框图

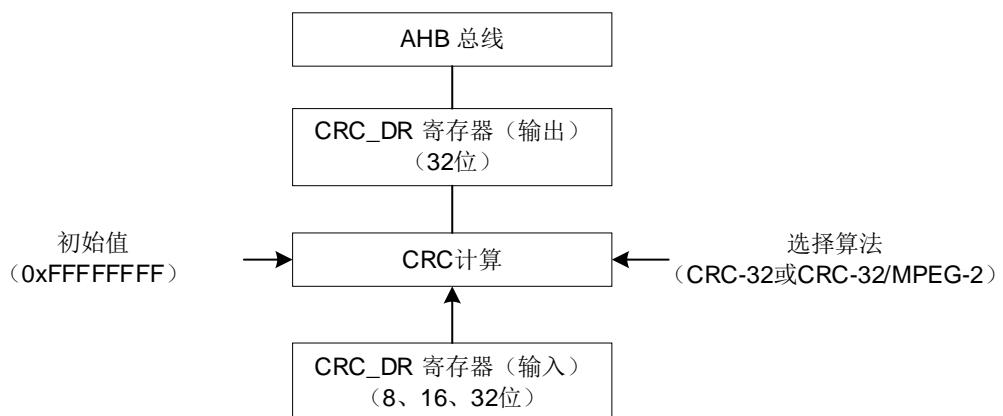


图 28-1 CRC 功能框图

28.3.2 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。

28.3.3 使用方法

28.3.3.1 CRC 计算操作步骤

- 使能 CRC 模块时钟（配置 RCC_AHBENR 的 CRC 位为 1）；
- 复位 CRC 模块（配置 RCC_AHBRSTR 的 CRC 位为 1）；
- 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。

28.3.3.2 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) 中读取出第一串数据的 CRC 运算结果。

28.4 寄存器

28.4.1 寄存器总览

表 28-1 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

28.4.2 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

28.4.3 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。

28.4.4 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	
								rw		rw		rw		rw	

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”

28.4.5 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。

29 DBG 调试支持

29.1 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。

29.2 功能描述

29.2.1 功能框图

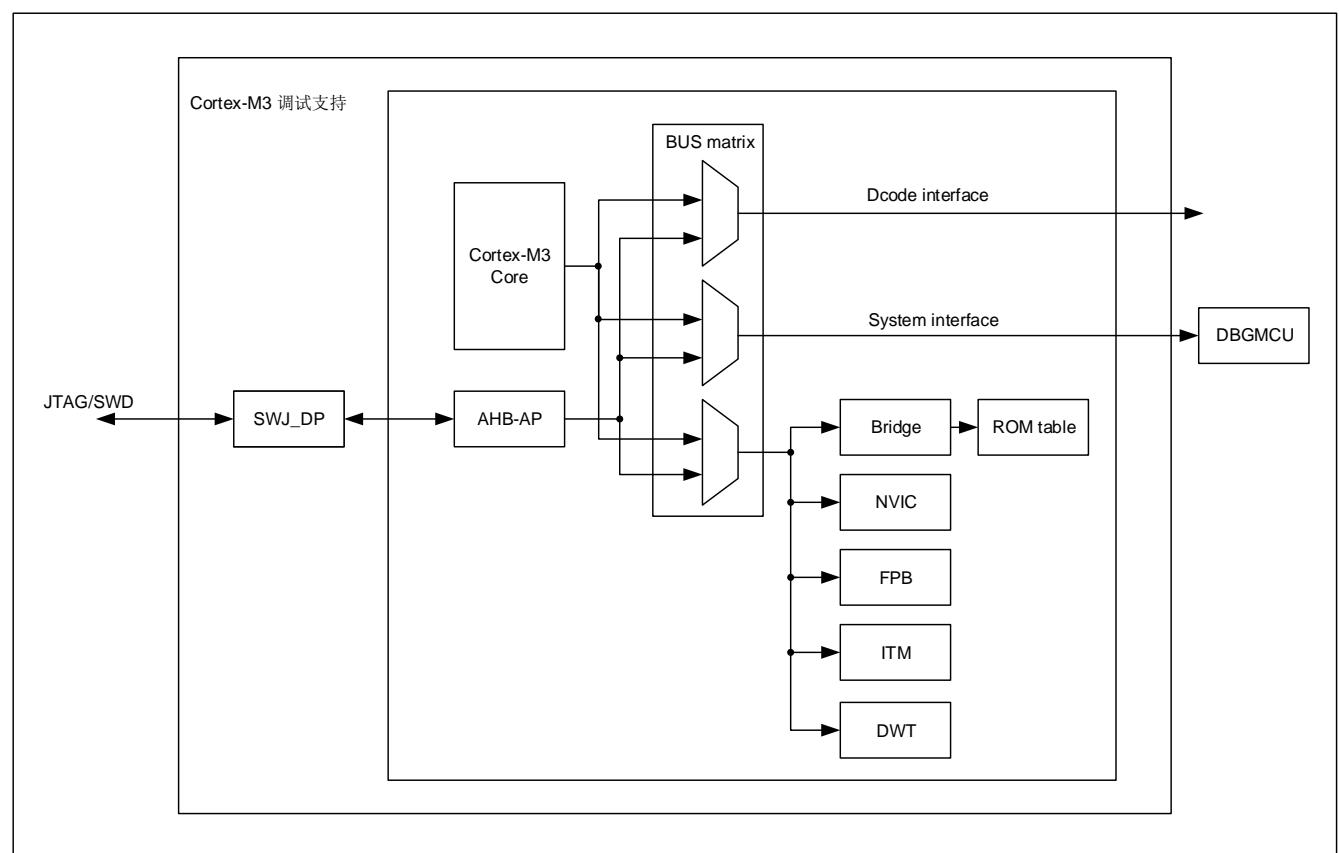


图 29-1 调试功能框图

Cortex®-M3 内核含有调试单元，该单元由以下部分组成：

SWJ_DP: SW/JTAG 调试端口

BPU: 断点调试单元

DWT: 数据观察点和跟踪

29.2.2 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。

29.2.3 SWJ 调试端口

该芯片的 5 个普通 I/O 口可用作 SWJ-DP 接口引脚，不同封装都支持 SWJ 调试端口。

表 29-1 SWJ 调试端口管脚

SWJ-DP 端口引脚名称	SW 调试接口		JTAG 调试接口		引脚分配
	类型	调试功能	类型	调试功能	
JTMS/SWDIO	输入/输出	串行数据输入/输出	输入	JTAG 模式选择	PA13
JTCK/SWCLK	输入	串行时钟	输入	JTAG 时钟	PA14
JTDI	-	-	输入	JTAG 数据输入	PA15
JTDO	-	-	输出	JTAG 数据输出	PB3
NJTRST	-	-	输入	JTAG nRESET	PB4

29.3 ID 代码和锁定机制

在芯片内部有多个 ID 编码，如下表：

表 29-2 ID 编码

ID 名	芯片
DEV_ID	0xCC9AA0E7
CPU TAP SW ID	0x2BA01477
CPU TAP JTAG ID	0x4BA00477

29.3.1 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。

29.3.2 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。

29.4 SWD 调试端口

29.4.1 SWD 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 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），这段区间内，任何设备不能驱动此信号线。

29.4.2 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 29-3 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 29-4 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 29-5 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。

29.4.3 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;

29.4.4 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;

29.4.5 SW-DP 寄存器

当 APnDP=0 时, 可以访问以下这些寄存器。

表 29-6 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 0x2BA0 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。

29.4.6 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

29.5 JTAG 调试端口

标准的 JTAG 状态机是通过一个 4 比特位的指令寄存器 (IR) 和 5 个数据寄存器实现的。详见 ARM Cortex-M3 r1p1 Technical Reference Manual

表 29-7 JTAG 调试端口寄存器映射

IR	数据寄存器	描述
1111	BYPASS	-
1110	IDCODE	标识代码寄存器 固定为 0x4BA00477 (用于识别 JTAG-DP)。
1010	DPACC	调试接口寄存器 初始化调试端口, 并允许访问调试接口寄存器
1011	APACC	存取接口寄存器 初始化存取接口, 并允许访问存取接口寄存器

IR	数据寄存器	描述
1000	ABORT	中止寄存器 第 0 位是中止位，写 1 产生一个 DAP 中止操作

表 29-8 JTAG-DP 寄存器

地址	寄存器	描述
0x0	-	保留，读作 0，写操作忽略
0x4	CTRL/STAT	控制/状态寄存器 提供对 DP 的控制以及有关 DP 的状态信息
0x8	AP SELECT	AP 选择寄存器 选择当前的存取接口（AP）以及该 AP 中有效的 4 字存取接口组（APBANK）
0xC	RDBUFF	读缓冲 读操作结果始终为 0，写操作无效。提供没有任何副作用的 DP 读操作，意味着调试器能够在操作序列的末尾插入一个读缓冲的 DP 读操作，来返回最后一次传输的读结果以及 ACK。

29.6 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

29.6.1 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。

29.6.2 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

29.7 寄存器

29.7.1 寄存器总览

表 29-9 DBG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID 编码寄存器	0xCC9AA0E7
0x04	DBG_CR	DBG 控制寄存器	0x00000000

29.7.2 DBG_IDCODE ID 编码寄存器

偏移地址: 0x40007080 (只支持 32 位访问, 只读)

复位值: 0xCC9AA0E7

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

29.7.3 DBG_CR 控制寄存器

偏移地址: 0x40007084 (只支持 32 位访问)

复位值: 0x0000 0000 (POR 复位, 不被系统复位所复位)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
											DBG_TI	DBG_TI	DBG_TI	DBG_TI	
					M7_ST	M6_ST	M5_ST	M8_ST							
					OP	OP	OP	OP							
											rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Res.	DBG_TI	DBG_TI	DBG_TI	DBG_TI	DBG_W	DBG_I	Res.	DBG_S	DBG_S	DBG_S	DBG_S
	M4_ST OP	M3_ST OP	M2_ST OP	M1_ST OP	WDG_S TOP	WDG_S TOP		TOP_F OR_LD O	TANDB Y		
	rw	rw	rw	rw	rw	rw		rw	rw		

Bit	Field	Description
31:21	Reserved	保留，必须保持复位值
20	DBG_TIM7_STOP	当内核进入调试状态时计数器 7 停止工作 (TIM7 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
19	DBG_TIM6_STOP	当内核进入调试状态时计数器 6 停止工作 (TIM6 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
18	DBG_TIM5_STOP	当内核进入调试状态时计数器 5 停止工作 (TIM5 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_TIM8_STOP	当内核进入调试状态时计数器 8 停止工作 (TIM8 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
16:14	Reserved	保留，必须保持复位值
13	DBG_TIM4_STOP	当内核进入调试状态时计数器 4 停止工作 (TIM4 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
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

Bit	Field	Description
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。

30 Device Electronic Signature 器件电子签名

30.1 器件电子签名简介

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

30.2 寄存器描述

基地址：0x1FFF F7E0

表 30-1 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

30.2.1 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)

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

U_ID (15: 0)

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)

30.2.2 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)															
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)												

30.2.3 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)															
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)												

31 修订记录

表 31-1 修订记录表

修订时间	修订版本	修订内容
2022/03/11	V1.1	内容和版式优化
2021/12/7	V1.0	初版发布