



用户手册 User Manual

MM32F5270
MM32F5280

基于 STAR-MC1 处理器(兼容Cortex-M33)的 32 位微控制器

版本：1.02

保留不通知的情况下，更改相关资料的权利

目录

目录	2
附图目录	24
表格目录	30
1 系统和存储器架构	34
1.1 系统架构	34
1.1.1 CBUS 总线	35
1.1.2 SBUS 总线	35
1.1.3 DMA 总线	36
1.1.4 USB_FS 总线	36
1.1.5 ENET 总线	36
1.1.6 TBUS 总线	36
1.1.7 ITCM / DTCM 总线	36
1.1.8 Mulit-AHB 总线矩阵	36
1.1.9 AHB2APB 桥	36
1.2 内核简介	36
2 地址映射	37
2.1 存储器映像和寄存器编址	37
2.2 内置的 SRAM	39
2.3 内置的 TCM SRAM	40
2.4 闪存存储器概述	40
3 芯片特定配置	41
3.1 中断向量表	41
3.2 外部中断映射表	44
3.3 DMA 通道分配	45
3.4 MindSwitch 映射关系	46
3.5 ADC 通道分配	49
3.6 系统模块硬件互联	50
3.7 启动配置	51
3.8 引导程序	51
4 Embedded FLASH 嵌入式闪存	52
4.1 简介	52
4.2 闪存构成与说明	52
4.2.1 闪存构成	52
4.2.2 选项字节说明	53
4.3 闪存操作与流程	55
4.3.1 闪存读操作	55
4.3.2 闪存编程方式与操作流程	55
4.3.3 对闪存块操作限制的解除与使能	58
4.3.4 对选项字节区块操作限制的解除与使能	59
4.3.5 主闪存块擦除	61

4.3.6	主闪存块编程	63
4.3.7	选项字节区块擦除	64
4.3.8	选项字节区块编程	65
4.3.9	闪存保护	66
4.4	寄存器	68
4.4.1	寄存器总览	68
4.4.2	FLASH_ACR 闪存访问控制寄存器	69
4.4.3	FLASH_KEYR FPEC 键寄存器	69
4.4.4	FLASH_OPTKEYR 闪存 OPTKEY 寄存器	70
4.4.5	FLASH_SR 闪存状态寄存器	70
4.4.6	FLASH_CR 闪存控制寄存器	71
4.4.7	FLASH_AR 闪存地址寄存器	72
4.4.8	FLASH_OBR 选项字节寄存器	72
4.4.9	FLASH_WRPRx 写保护寄存器	73
5	PWR 电源控制	74
5.1	供电系统	74
5.1.1	模拟模块供电	75
5.1.2	数字模块供电	75
5.1.3	电池备份模块供电	75
5.1.4	VDD 域	75
5.1.5	备份域	75
5.1.6	CORE 域	75
5.2	电源管理器	76
5.2.1	POR 上电复位和 PDR 掉电复位	76
5.2.2	PVD 可编程电压监测器	76
5.3	功耗控制	77
5.3.1	功耗控制概述	77
5.3.2	运行模式降低系统时钟	80
5.3.3	外设时钟的控制	80
5.3.4	Low Power Run Mode 低功耗运行模式	81
5.3.5	Sleep Mode 睡眠模式	81
5.3.6	Low Power Sleep Mode 低功耗睡眠模式	82
5.3.7	Stop Mode 停机模式	83
5.3.8	DeepStop Mode 深度停机模式	84
5.3.9	Standby Mode 待机模式	86
5.4	调试模式	87
5.5	电源控制寄存器	88
5.5.1	PWR_CR1 电源控制寄存器 1	88
5.5.2	PWR_CSR 电源控制/状态寄存器	89
5.5.3	PWR_CR2 电源控制寄存器 2	90
5.5.4	PWR_CR3 电源控制寄存器 3	90
5.5.5	PWR_CR4 电源控制寄存器 4	90
5.5.6	PWR_CR5 电源控制寄存器 5	91
5.5.7	PWR_CR6 电源控制寄存器 6	92

5.5.8	PWR_SR 电源状态寄存器	93
5.5.9	PWR_SCR 电源状态清除寄存器	93
5.5.10	PWR_CFGR 电源配置寄存器	94
6	RCC 时钟和复位	95
6.1	复位单元	95
6.1.1	简介	95
6.1.2	功能框图	95
6.1.3	主要特征	95
6.1.4	功能描述	96
6.2	时钟单元	97
6.2.1	简介	97
6.2.2	功能框图	98
6.2.3	主要特征	99
6.2.4	功能描述	99
6.3	寄存器	109
6.3.1	寄存器总览	109
6.3.2	RCC_CR 时钟控制寄存器	109
6.3.3	RCC_CFGR 时钟配置寄存器	112
6.3.4	RCC_CIR 时钟中断寄存器	115
6.3.5	RCC_AHBRSTR AHB 外设复位寄存器	117
6.3.6	RCC_APB2RSTR APB2 外设复位寄存器	119
6.3.7	RCC_APB1RSTR APB1 外设复位寄存器	121
6.3.8	RCC_AHBENR AHB 外设时钟使能寄存器	124
6.3.9	RCC_APB2ENR APB2 外设时钟使能寄存器	126
6.3.10	RCC_APB1ENR APB1 外设时钟使能寄存器	128
6.3.11	RCC_BDCR 备份域控制寄存器	131
6.3.12	RCC_CSR 控制状态寄存器	132
6.3.13	RCC_SYSCFG 系统配置寄存器	135
6.3.14	RCC_CFGR2 时钟配置寄存器 2	135
6.3.15	RCC_ICSCR 内部时钟校准寄存器	137
6.3.16	RCC_PLL1CFGR PLL1 配置寄存器	138
6.3.17	RCC_PLL2CFGR PLL2 配置寄存器	139
6.3.18	RCC_ADC1CFGR ADC1 配置寄存器	140
6.3.19	RCC_ADC2CFGR ADC2 配置寄存器	141
6.3.20	RCC_DACCFGR DAC 配置寄存器	141
6.3.21	RCC_TPIUCFGR TPIU 配置寄存器	142
7	CRS 时钟回馈系统	143
7.1	介绍	143
7.2	主要特征	143
7.3	功能描述	144
7.3.1	功能框图	144
7.3.2	信号描述	144
7.3.3	功能概述	144
7.3.4	同步输入	144

7.3.5	频率误差测量	145
7.3.6	频率误差计算及自动校准	149
7.3.7	CRS 初始化及配置	149
7.4	CRS 低功耗模式	150
7.5	中断	150
7.6	寄存器	151
7.6.1	寄存器总览	151
7.6.2	CRS_CR 控制寄存器	151
7.6.3	CRS_CFGR 配置寄存器	152
7.6.4	CRS_ISR 中断状态寄存器	153
7.6.5	CRS_ICR 中断标志清除寄存器	155
8	SYSCFG 系统控制器	156
8.1	SYSCFG 简介	156
8.2	寄存器描述	156
1.2.1	SYSCFG_CFGR1 配置寄存器 1	156
1.2.2	SYSCFG_EXTICR1 外部中断配置寄存器 1	158
1.2.3	SYSCFG_EXTICR2 外部中断配置寄存器 2	158
1.2.4	SYSCFG_EXTICR3 外部中断配置寄存器 3	159
1.2.5	SYSCFG_EXTICR4 外部中断配置寄存器 4	159
1.2.6	SYSCFG_CFGR2 配置寄存器 2	160
1.2.7	SYSCFG_PDETCSR 电源检测配置状态寄存器	161
1.2.8	SYSCFG_VOSDLY 电压调试器输出延时寄存器	162
1.2.9	SYSCFG_DMARMP DMA 重映射寄存器	162
1.2.10	SYSCFG_BUS_PRI 总线优先级配置寄存器	164
9	EXTI 中断和事件	166
9.1	简介	166
9.2	主要特征	166
9.3	功能描述	166
9.3.1	功能框图	166
9.3.2	中断和异常向量	167
9.3.3	唤醒事件管理	167
9.3.4	中断功能描述	167
9.3.5	硬件中断输出	167
9.3.6	硬件事件输出	168
9.3.7	软件中断与事件输出	168
9.3.8	外部中断映射	168
9.4	寄存器	168
9.4.1	寄存器总览	168
9.4.2	EXTI_IMR 中断屏蔽寄存器	168
9.4.3	EXTI_EMR 事件屏蔽寄存器	169
9.4.4	EXTI_RTSTR 上升沿触发选择寄存器	169
9.4.5	EXTI_FTSTR 下降沿触发选择寄存器	170
9.4.6	EXTI_SWIER 软件中断事件寄存器	170
9.4.7	EXTI_PR 软件中断事件挂起寄存器	171

10	GPIO 通用端口	172
10.1	简介	172
10.2	主要特征	172
10.3	功能描述	173
10.3.1	功能框图	173
10.3.2	GPIO 端口配置	173
10.3.3	复用功能	174
10.3.4	GPIO 锁定机制	174
10.3.5	输入配置	175
10.3.6	输出配置	176
10.3.7	复用功能配置	176
10.3.8	模拟输入配置	177
10.3.9	外部时钟复用 GPIO 端口	178
10.3.10	SWD 复用功能重映射	178
10.4	GPIO 寄存器描述	178
10.4.1	寄存器总览	178
10.4.2	GPIOx_CRL 端口配置低寄存器	179
10.4.3	GPIOx_CRH 端口配置高寄存器	179
10.4.4	GPIOx_IDR 端口输入数据寄存器	180
10.4.5	GPIOx_ODR 端口输出数据寄存器	181
10.4.6	GPIOx_BSRR 端口设置/清除寄存器	181
10.4.7	GPIOx_BRR 端口位清除寄存器	182
10.4.8	GPIOx_LCKR 端口配置锁定寄存器	182
10.4.9	GPIOx_DCR 端口输出开漏控制寄存器	183
10.4.10	GPIOx_AFRH 端口复用功能低位寄存器	183
10.4.11	GPIOx_AFRH 端口复用功能高位寄存器	184
11	DMA1/2 直接存储器访问控制器	185
11.1	DMA 简介	185
11.2	DMA 功能框图	185
11.3	DMA 主要特征	185
11.4	中断	186
11.5	DMA	186
11.5.1	DMA 请求映像	186
11.6	功能描述	186
11.6.1	DMA 处理	186
11.6.2	仲裁器	187
11.6.3	DMA 通道	187
11.6.4	可编程的数据传输宽度，对齐方式和数据大小端	188
11.6.5	错误管理	190
11.7	DMA 寄存器描述	190
11.7.1	DMA_ISR DMA 中断状态寄存器	191
11.7.2	DMA_IFCR DMA 中断标志清除寄存器	191
11.7.3	DMA_CCRx DMA 通道 x 配置寄存器 (x=1~8)	192
11.7.4	DMA_CNDTRx DMA 通道 x 传输数量寄存器 (x=1~8)	194

11.7.5	DMA_CPARx DMA 通道 x 外设地址寄存器 (x=1~8)	194
11.7.6	DMA_CMARx DMA 通道 x 存储器地址寄存器 (x=1~8)	195
12	MDS 可编程 IP 互联模块	196
12.1	介绍	196
12.2	主要特性	196
12.3	功能描述	197
12.3.1	功能框图	197
12.3.2	时钟复位	197
12.3.3	功能概述	197
12.3.4	触发源和触发目标	198
12.4	寄存器	199
12.4.1	寄存器总览	199
12.4.2	MDS_TRIGxCR 触发通道 x 控制和状态寄存器	200
12.4.3	MDS_TRIG_CLUxSEL 触发通道 CLUx 逻辑运算输入选择寄存器	201
12.4.4	MDS_TRIG_CLUxCFG 触发通道 CLUx 逻辑运算配置寄存器	201
12.4.5	MDS_SWTRIG 软件触发寄存器	202
12.4.6	MDS_CONNxCR 连接通道 x 控制和状态寄存器	203
12.4.7	MDS_CONN_CLUxSEL 连接通道 CLUx 逻辑运算输入选择寄存器	203
12.4.8	MDS_CONN_CLUxCFG 连接通道 CLUx 逻辑运算配置寄存器	204
13	CRC_POLY 循环冗余校验计算单元 (多项式可配)	206
13.1	介绍	206
13.2	主要特性	206
13.3	功能描述	206
13.3.1	功能框图	206
13.3.2	功能概述	206
13.3.3	使用方法	207
13.4	寄存器	208
13.4.1	寄存器总览	208
13.4.2	CRC_DR CRC 数据寄存器	208
13.4.3	CRC_IDR CRC 独立数据寄存器	208
13.4.4	CRC_CR CRC 控制寄存器	209
13.4.5	CRC_IVR CRC 初值寄存器	210
13.4.6	CRC_PR CRC 多项式寄存器	210
14	CORDIC 坐标旋转算法	211
14.1	介绍	211
14.2	功能描述	211
14.2.1	功能概述	211
14.3	atan 计算的配置实例	211
14.4	sin、cos 计算的配置实例	212
14.5	寄存器	212
14.5.1	寄存器总览	212
14.5.2	CORDIC_DXR CORDIC X 数据寄存器	213
14.5.3	CORDIC_DYR CORDIC Y 数据寄存器	213
14.5.4	CORDIC_DZR CORDIC Z 数据寄存器	213

14.5.5	CORDIC_RXR CORDIC X 结果寄存器	214
14.5.6	CORDIC_RYR CORDIC Y 结果寄存器	214
14.5.7	CORDIC_RZR CORDIC Z 结果寄存器	215
14.5.8	CORDIC_CR CORDIC 控制寄存器	215
14.5.9	CORDIC_SR CORDIC 状态寄存器	216
15	QSPI 四线式 SPI 接口	217
15.1	简介	217
15.2	主要特征	217
15.3	功能描述	218
15.3.1	功能框图	218
15.3.2	信号描述	218
15.3.3	数据格式	218
15.3.4	动作模式	220
15.3.5	总线接口	221
15.3.6	输入数据采样	221
15.3.7	空闲状态控制	222
15.3.8	XIP 模式	222
15.3.9	中断	223
15.3.10	DMA	224
15.4	寄存器	224
15.4.1	寄存器总览	224
15.4.2	QSPI_CR 控制寄存器	225
15.4.3	QSPI_SR 状态寄存器	226
15.4.4	QSPI_DMCR 直接模式控制寄存器	227
15.4.5	QSPI_IMCR 间接模式控制寄存器	229
15.4.6	QSPI_DABR 直接模式交替字节寄存器	231
15.4.7	QSPI_IABR 间接模式交替字节寄存器	231
15.4.8	QSPI_IADR 间接模式地址寄存器	232
15.4.9	QSPI_IDFR 间接模式数据 FIFO 寄存器	232
15.4.10	QSPI_IDLR 间接模式数据长度寄存器	233
15.4.11	QSPI_IWCR 间接模式等待计数寄存器	233
15.4.12	QSPI_IDER 中断 DMA 使能寄存器	234
16	FSMC 存储器接口	236
16.1	设计概述	236
16.1.1	框图	236
16.1.2	接口描述	236
16.1.3	特性	237
16.2	功能描述	237
16.2.1	FSMC 数据位宽	237
16.2.2	FSMC 模式配置	238
16.2.3	NOR Flash/SRAM 模式	239
16.2.4	8080 协议模式	240
16.2.5	6800 协议模式	242
16.3	寄存器描述	243

16.3.1	存储器屏蔽寄存器 (SMSKR0)	244
16.3.2	存储器屏蔽寄存器 (SMSKR1)	245
16.3.3	存储器屏蔽寄存器 (SMSKR2)	246
16.3.4	存储器屏蔽寄存器 (SMSKR3)	247
16.3.5	存储器时序寄存器 (SMTMGR_SET0)	248
16.3.6	存储器时序寄存器 (SMTMGR_SET1)	249
16.3.7	存储器时序寄存器 (SMTMGR_SET2)	249
16.3.8	存储器控制寄存器 (SMCTLR)	250
17	ADC 模拟/数字转换	252
17.1	简介	252
17.2	功能框图	252
17.3	主要特征	253
17.4	中断	253
17.5	DMA	254
17.6	功能描述	254
17.6.1	时钟	254
17.6.2	数据补偿	254
17.6.3	数据对齐	255
17.6.4	可编程分辨率	256
17.6.5	可编程采样时间	256
17.6.6	数据通道寄存器	256
17.6.7	通道选择	256
17.6.8	ADC 开关	257
17.6.9	校准	257
17.6.10	通道转换方式	259
17.6.11	通道转换	259
17.6.12	ADC 过采样	267
17.6.13	ADC 触发信号	270
17.6.14	模拟看门狗	270
17.6.15	内部温度传感器	271
17.6.16	内部电压传感器	271
17.7	寄存器描述	271
17.7.1	ADC_ADDDATA A/D 数据寄存器	272
17.7.2	ADC_ADCFG A/D 配置寄存器	273
17.7.3	ADC_ADCR A/D 控制寄存器	274
17.7.4	ADC_ADCHS A/D 通道选择寄存器	277
17.7.5	ADC_ADCMPR A/D 模拟看门狗比较寄存器	278
17.7.6	ADC_ADSTA A/D 状态寄存器	279
17.7.7	ADC_ADDRn A/D 通道数据寄存器 (n=0~15)	280
17.7.8	ADC_ADSTA_EXT A/D 扩展状态寄存器	280
17.7.9	ADC_CHANY0 A/D 任意通道的通道选择寄存器 0	282
17.7.10	ADC_CHANY1 A/D 任意通道的通道选择寄存器 1	283
17.7.11	ADC_ANY_CFG A/D 任意通道配置寄存器	285
17.7.12	ADC_ANY_CR A/D 任意通道控制寄存器	287

17.7.13	ADC_ADCFG2 ADC 配置寄存器	289
17.7.14	ADC_SMPR1 A/D 采样配置寄存器 1	291
17.7.15	ADC_SMPR2 A/D 采样配置寄存器 2	292
17.7.16	ADC_SMPR3 A/D 采样配置寄存器 3	293
17.7.17	ADC_JOFRn (n=0~3) A/D 注入通道数据补偿寄存器	294
17.7.18	ADC_JSQR A/D 注入通道序列寄存器	295
17.7.19	ADC_JADDATA A/D 注入数据寄存器	295
17.7.20	ADC_JDRn (n=0~3) A/D 注入通道数据寄存器	296
18	DAC 数字/模拟转换器	298
18.1	DAC 简介	298
18.2	主要特征	298
18.3	功能描述	300
18.3.1	DAC 通道的使能	300
18.3.2	使能 DAC 输出缓存	300
18.3.3	DAC 输出电压	300
18.3.4	DAC 触发源的选择	300
18.3.5	DMA 请求	300
18.3.6	DAC 数据格式	301
18.3.7	带三角波生成的 DAC 转换	302
18.3.8	带噪声波生成的 DAC 转换	303
18.3.9	DAC 转换	304
18.3.10	DAC 双通道转换模式	305
18.4	寄存器描述	307
18.4.1	DAC_CR 控制寄存器	307
18.4.2	DAC_SWTRIGR 软件触发寄存器	310
18.4.3	DAC_DHR12R1 通道 1 的 12 位右对齐数据保持寄存器	310
18.4.4	DAC_DHR12L1 通道 1 的 12 位左对齐数据保持寄存器	311
18.4.5	DAC_DHR8R1 通道 1 的 8 位右对齐数据保持寄存器	311
18.4.6	DAC_DHR12R2 通道 2 的 12 位右对齐数据保持寄存器	312
18.4.7	DAC_DHR12L2 通道 2 的 12 位左对齐数据保持寄存器	312
18.4.8	DAC_DHR8R2 通道 2 的 8 位右对齐数据保持寄存器	313
18.4.9	DAC_DHR12RD 双通道的 12 位右对齐数据保持寄存器	313
18.4.10	DAC_DHR12LD 双通道的 12 位左对齐数据保持寄存器	313
18.4.11	DAC_DHR8RD 双通道的 8 位右对齐数据保持寄存器	314
18.4.12	DAC_DOR1 通道 1 数据输出寄存器	314
18.4.13	DAC_DOR2 通道 2 数据输出寄存器	315
19	COMP 比较器	316
19.1	简介	316
19.2	功能框图	316
19.3	主要特征	316
19.4	功能描述	317
19.4.1	比较器时钟和复位	317
19.4.2	比较器开关控制	317
19.4.3	比较器输入和输出	317

19.4.4	比较器通道选择	318
19.4.5	中断和唤醒	318
19.4.6	功耗模式	318
19.4.7	比较器锁定机制	319
19.4.8	迟滞电压	319
19.5	寄存器	319
19.5.1	比较器控制状态寄存器 (COMPx_CSR) (x=1, 2, 3)	320
19.5.2	比较器外部参考电压寄存器 (COMP_CRV)	323
19.5.3	比较器轮询寄存器 (COMPx_POLL) (x=1, 2, 3)	324
20	TIM1/8 高级定时器	326
20.1	简介	326
20.2	功能框图	326
20.3	主要特征	327
20.4	功能描述	327
20.4.1	时钟	327
20.4.2	重复计数器	332
20.4.3	输入捕获	334
20.4.4	比较输出	336
20.4.5	从模式	348
20.4.6	定时器同步	352
20.4.7	定时器异或	355
20.4.8	调试模式	357
20.4.9	中断	357
20.4.10	DMA	357
20.5	寄存器描述	358
20.5.1	TIMx_CR1 控制寄存器 1	359
20.5.2	TIMx_CR2 控制寄存器 2	360
20.5.3	TIMx_SMCR 从模式控制寄存器	362
20.5.4	TIMx_DIER DMA/中断使能寄存器	364
20.5.5	TIMx_SR 状态寄存器	366
20.5.6	TIMx_EGR 事件产生寄存器	368
20.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1	369
20.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2	373
20.5.9	TIMx_CCER 捕获/比较使能寄存器	377
20.5.10	TIMx_CNT 计数器	380
20.5.11	TIMx_PSC 预分频器	380
20.5.12	TIMx_ARR 自动预装载寄存器	380
20.5.13	TIMx_RCR 重复计数寄存器	381
20.5.14	TIMx_CCR1 捕获/比较寄存器 1	381
20.5.15	TIMx_CCR2 捕获/比较寄存器 2	382
20.5.16	TIMx_CCR3 捕获/比较寄存器 3	382
20.5.17	TIMx_CCR4 捕获/比较寄存器 4	382
20.5.18	TIMx_BDTR 刹车和死区寄存器	383
20.5.19	TIMx_DCR DMA 控制寄存器	385

20.5.20	TIMx_DMAR 连续模式 DMA 地址寄存器	386
20.5.21	TIMx_CCMR3 捕获/比较模式寄存器 3	386
20.5.22	TIMx_CCR5 捕获/比较寄存器 5	387
20.5.23	TIMx_PDER PWM 移相/DMA repeat 更新请求使能寄存器	387
20.5.24	TIMx_CCRxFALL PWM 移相递减计数捕获/比较寄存器	388
20.5.25	TIMx_BKINF 刹车输入滤波寄存器	389
21	TIM2/5 通用定时器	392
21.1	简介	392
21.2	功能框图	392
21.3	主要特征	392
21.4	功能描述	393
21.4.1	时钟	393
21.4.2	输入捕获	398
21.4.3	比较输出	400
21.4.4	从模式	407
21.4.5	定时器同步	411
21.4.6	定时器异或	411
21.4.7	调试模式	412
21.4.8	中断	412
21.4.9	DMA	412
21.5	寄存器描述	413
21.5.1	TIMx_CR1 控制寄存器 1	414
21.5.2	TIMx_CR2 控制寄存器 2	415
21.5.3	TIMx_SMCR 从模式控制寄存器	416
21.5.4	TIMx_DIER DMA/中断使能寄存器	419
21.5.5	TIMx_SR 状态寄存器	421
21.5.6	TIMx_EGR 事件产生寄存器	422
21.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1	423
21.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2	426
21.5.9	TIMx_CCER 捕获/比较使能寄存器	430
21.5.10	TIMx_CNT 计数器	432
21.5.11	TIMx_PSC 预分频器	432
21.5.12	TIMx_ARR 自动预装载寄存器	433
21.5.13	TIMx_CCR1 捕获/比较寄存器 1	433
21.5.14	TIMx_CCR2 捕获/比较寄存器 2	434
21.5.15	TIMx_CCR3 捕获/比较寄存器 3	434
21.5.16	TIMx_CCR4 捕获/比较寄存器 4	435
21.5.17	TIMx_DCR DMA 控制寄存器	435
21.5.18	TIMx_DMAR 连续模式 DMA 地址寄存器	436
21.5.19	TIMx_OR 输入选项寄存器	436
22	TIM3/4 通用定时器	438
22.1	简介	438
22.2	功能框图	438
22.3	主要特征	438

22.4	功能描述	439
22.4.1	时钟	439
22.4.2	输入捕获	444
22.4.3	比较输出	446
22.4.4	从模式	453
22.4.5	定时器同步	457
22.4.6	定时器异或	457
22.4.7	调试模式	458
22.4.8	中断	458
22.4.9	DMA	458
22.5	寄存器描述	459
22.5.1	TIMx_CR1 控制寄存器 1	460
22.5.2	TIMx_CR2 控制寄存器 2	461
22.5.3	TIMx_SMCR 从模式控制寄存器	462
22.5.4	TIMx_DIER DMA/中断使能寄存器	465
22.5.5	TIMx_SR 状态寄存器	467
22.5.6	TIMx_EGR 事件产生寄存器	468
22.5.7	TIMx_CCMR1 捕获/比较模式寄存器 1	469
22.5.8	TIMx_CCMR2 捕获/比较模式寄存器 2	472
22.5.9	TIMx_CCER 捕获/比较使能寄存器	476
22.5.10	TIMx_CNT 计数器	478
22.5.11	TIMx_PSC 预分频器	478
22.5.12	TIMx_ARR 自动预装载寄存器	479
22.5.13	TIMx_CCR1 捕获/比较寄存器 1	479
22.5.14	TIMx_CCR2 捕获/比较寄存器 2	480
22.5.15	TIMx_CCR3 捕获/比较寄存器 3	480
22.5.16	TIMx_CCR4 捕获/比较寄存器 4	480
22.5.17	TIMx_DCR DMA 控制寄存器	481
22.5.18	TIMx_DMAR 连续模式 DMA 地址寄存器	481
23	TIM6/7 基本定时器	483
23.1	简介	483
23.2	功能框图	483
23.3	主要特征	483
23.4	功能描述	484
23.4.1	时钟	484
23.4.2	调试模式	485
23.4.3	中断	485
23.4.4	DMA	485
23.5	寄存器描述	485
23.5.1	TIMx_CR1 控制寄存器 1	486
23.5.2	TIMx_DIER DMA/中断使能寄存器	486
23.5.3	TIMx_SR 状态寄存器	487
23.5.4	TIMx_EGR 事件产生寄存器	487
23.5.5	TIMx_CNT 计数器	488

23.5.6	TIMx_PSC 预分频器	488
23.5.7	TIMx_ARR 自动预装载寄存器	488
24	LPTIMER 低功耗定时器	490
24.1	简介	490
24.2	功能框图	490
24.3	主要特征	490
24.4	中断	491
24.5	功能描述	491
24.5.1	计数模式	491
24.5.2	普通定时器	491
24.5.3	Trigger 脉冲触发计数	491
24.5.4	TimeOut 模式	492
24.5.5	PWM 输出	492
24.6	寄存器	492
24.6.1	寄存器概览	492
24.6.2	LPT_CFG 配置寄存器	493
24.6.3	LPT_IE 中断使能寄存器	494
24.6.4	LPT_IF 中断标志寄存器	494
24.6.5	LPT_CTRL 控制寄存器	495
24.6.6	LPT_CNT 计数寄存器	495
24.6.7	LPT_CMP 比较值寄存器	495
24.6.8	LPT_TARGET 目标值寄存器	496
25	IWDG 独立看门狗	497
25.1	简介	497
25.2	功能框图	498
25.3	主要特征	498
25.4	功能描述	499
25.4.1	流程框图	500
25.4.2	独立看门狗超时时间	500
25.4.3	中断	501
25.5	寄存器	501
25.5.1	寄存器总览	501
25.5.2	IWDG_KR 键寄存器	501
25.5.3	IWDG_PR 预分频寄存器	502
25.5.4	IWDG_RLR 重装载寄存器	502
25.5.5	IWDG_SR 状态寄存器	503
25.5.6	IWDG_CR 控制寄存器	504
25.5.7	IWDG_IGEN 中断生成寄存器	504
25.5.8	IWDG_CNT 计数寄存器	505
25.5.9	IWDG_PS 分频计数寄存器	505
26	WWDG 窗口看门狗	507
26.1	简介	507
26.2	功能框图	507
26.3	主要特征	507

26.4	功能描述	508
26.5	窗口看门狗超时时间	509
26.6	寄存器	509
26.6.1	控制寄存器 (WWDG_CR)	510
26.6.2	配置寄存器 (WWDG_CFGR)	510
26.6.3	状态寄存器 (WWDG_SR)	511
27	RTC 实时时钟器	512
27.1	RTC 简介	512
27.2	主要特征	512
27.3	功能描述	513
27.3.1	功能框图	513
27.3.2	功能概述	513
27.3.3	模块复位	514
27.3.4	寄存器读取	514
27.3.5	寄存器配置	514
27.3.6	标志位产生	515
27.3.7	RTC 闹钟描述	516
27.3.8	RTC 外部中断事件输出	516
27.4	寄存器描述	516
27.4.1	寄存器总览	516
27.4.2	控制寄存器高位 (RTC_CRH)	517
27.4.3	控制寄存器低位(RTC_CRL).....	517
27.4.4	预分频装载寄存器高位(RTC_PRLH).....	519
27.4.5	预分频装载寄存器低位(RTC_PRL).....	519
27.4.6	预分频器分频因子寄存器高位(RTC_DIVH).....	519
27.4.7	预分频器分频因子寄存器低位(RTC_DIVL).....	520
27.4.8	计数器寄存器高位(RTC_CNTH)	520
27.4.9	计数器寄存器低位(RTC_CNTL).....	520
27.4.10	闹钟寄存器高位(RTC_ALRH)	521
27.4.11	闹钟寄存器低位(RTC_ALRL).....	521
27.4.12	毫秒闹钟寄存器高位 (RTC_MSRH).....	521
27.4.13	毫秒闹钟寄存器低位 (RTC_MSRL)	522
27.4.14	RTC LSE 配置寄存器 (RTC_LSE_CFG).....	522
28	BKP 备份寄存器	524
28.1	简介	524
28.2	主要特征	524
28.3	功能描述	524
28.3.1	时钟校准	524
28.3.2	侵入检测	524
28.4	寄存器	525
28.4.1	寄存器总览	525
28.4.2	备份数据寄存器 n(BKP_DRn)(n = 1 .. 20).....	525
28.4.3	时钟校准寄存器 (BKP_RTCCR)	525
28.4.4	备份控制寄存器 (BKP_CR).....	526

28.4.5	备份控制状态寄存器 (BKP_CSR).....	527
29	UART 通用异步收发器.....	528
29.1	UART 简介.....	528
29.2	UART 功能框图.....	528
29.3	UART 主要特征.....	528
29.4	UART 功能概述.....	529
29.4.1	UART 特性描述.....	530
29.4.2	分数波特率发生器	530
29.4.3	采样	531
29.4.4	容忍度	531
29.4.5	校验控制	531
29.4.6	发送器	532
29.4.7	接收器	533
29.4.8	自动波特率检测	534
29.4.9	九位数据通信	535
29.4.10	多处理器通信	535
29.4.11	单线半双工通信	537
29.4.12	智能卡	537
29.4.13	红外 IrDA 功能.....	539
29.5	中断	540
29.6	DMA.....	541
29.7	寄存器.....	541
29.7.1	寄存器总览	541
29.7.2	UART_TDR 发送数据寄存器.....	542
29.7.3	UART_RDR 接收数据寄存器.....	542
29.7.4	UART_CSR 当前状态寄存器	542
29.7.5	UART_ISR 中断状态寄存器.....	543
29.7.6	UART_IER 中断使能寄存器	545
29.7.7	UART_ICR 中断清除寄存器.....	546
29.7.8	UART_GCR 全局控制寄存器.....	547
29.7.9	UART_CCR 通用控制寄存器.....	548
29.7.10	UART_BRR 波特率寄存器.....	550
29.7.11	UART_FRA 分数波特率寄存器	550
29.7.12	UART_RXADDR 接收地址寄存器	551
29.7.13	UART_RXMASK 接收掩码寄存器.....	551
29.7.14	UART_SCR SCR 寄存器.....	552
29.7.15	UART_IDLR IDLE 数据长度寄存器.....	552
29.7.16	UART_ABRCR 自动波特率寄存器	553
29.7.17	UART_IRDA 红外功能控制寄存器	553
30	LPUART 低功耗通用异步收发器	555
30.1	简介	555
30.2	功能框图	555
30.3	主要特征	555
30.4	中断	556

30.5	功能描述	557
30.5.1	特性描述	557
30.5.2	接收时序	557
30.5.3	发送时序	558
30.5.4	调制寄存器建议配置	558
30.5.5	奇偶校验控制	559
30.5.6	发送器	559
30.5.7	接收器	560
30.5.8	休眠模式下的数据接收唤醒	561
30.5.9	低功耗模式下的数据自动收发	561
30.6	寄存器描述	562
30.6.1	寄存器概览	562
30.6.2	LPUART_BAUD 波特率寄存器	562
30.6.3	LPUART_MODU 波特率调制控制寄存器	563
30.6.4	LPUART_IFR 中断标志寄存器	563
30.6.5	LPUART_SR 状态寄存器	564
30.6.6	LPUART_CR 控制寄存器	565
30.6.7	LPUART_EN 发送接收使能寄存器	567
30.6.8	LPUART_RXD 接收数据寄存器	567
30.6.9	LPUART_TXD 发送数据寄存器	568
30.6.10	LPUART_COMPARE 数据匹配寄存器	568
30.6.11	LPUART_WKCKE 唤醒寄存器	568
31	SPI_I2S 串行外设接口	570
31.1	SPI_I2S 功能框图	570
31.2	SPI_I2S 简述	570
31.3	SPI 功能描述	571
31.3.1	概述	571
31.3.2	SPI 主要特征	574
31.3.3	SPI 从模式	574
31.3.4	SPI 主模式	575
31.3.5	波特率设置	576
31.3.6	SPI 特殊功能 (TI 模式)	577
31.3.7	中断	579
31.3.8	DMA 传输	579
31.4	I2S 功能描述	580
31.4.1	功能概述	580
31.4.2	I2S 主要特征	580
31.4.3	I2S 总线接口	581
31.4.4	数据格式	581
31.4.5	通信标准	582
31.4.6	从模式	584
31.4.7	主模式	585
31.4.8	时钟预分频器	586
31.4.9	中断	588

31.4.10	DMA 传输	589
31.5	寄存器	589
31.5.1	寄存器总览	589
31.5.2	SPI_I2S_TXREG 发送数据寄存器	590
31.5.3	SPI_I2S_RXREG 接收数据寄存器	590
31.5.4	SPI_I2S_CSTAT 当前状态寄存器	590
31.5.5	SPI_I2S_INTSTAT 中断状态寄存器	591
31.5.6	SPI_I2S_INTEN 中断使能寄存器	593
31.5.7	SPI_I2S_INTCLR 中断清除寄存器	594
31.5.8	SPI_I2S_GCTL 全局控制寄存器	595
31.5.9	SPI_I2S_CCTL 通用控制寄存器	597
31.5.10	SPI_I2S_SPBRG 波特率发生器	599
31.5.11	SPI_I2S_RXDNR 接收数据个数寄存器	599
31.5.12	SPI_I2S_NSSR 从机片选寄存器	600
31.5.13	SPI_I2S_EXTCTL 数据控制寄存器	600
31.5.14	SPI_I2S_I2SCFGR I2S 配置寄存器	601
32	I2C 内部集成电路接口	603
32.1	介绍	603
32.2	主要特征	603
32.3	功能描述	604
32.3.1	功能框图	604
32.3.2	信号描述	604
32.3.3	I2C 协议	605
32.3.4	工作模式	612
32.3.5	SMBus/PMBus	618
32.3.6	中断	620
32.3.7	DMA	621
32.4	寄存器	622
32.4.1	寄存器总览	622
32.4.2	I2C_CR 控制寄存器	623
32.4.3	I2C_TAR 目标地址寄存器	626
32.4.4	I2C_SAR 从地址寄存器	626
32.4.5	I2C_DR 数据命令寄存器	627
32.4.6	I2C_SSHR 标准模式时钟高电平计数寄存器	627
32.4.7	I2C_SSLR 标准模式时钟低电平计数寄存器	628
32.4.8	I2C_FSHR 快速/超快速模式时钟高电平计数寄存器	628
32.4.9	I2C_FSLR 快速/超快速模式时钟低电平计数寄存器	629
32.4.10	I2C_ISR 中断状态寄存器	629
32.4.11	I2C_IMR 中断屏蔽寄存器	630
32.4.12	I2C_RAWISR RAW 中断寄存器	631
32.4.13	I2C_RXTLR 接收阈值	632
32.4.14	I2C_TXTLR 发送阈值	633
32.4.15	I2C_ICR 组合和独立中断清除寄存器	633
32.4.16	I2C_RX_UNDER 清除 RX_UNDER 中断寄存器	633

32.4.17	I2C_RX_OVER 清除 RX_OVER 中断寄存器.....	634
32.4.18	I2C_TX_OVER 清除 TX_OVER 中断寄存器.....	634
32.4.19	I2C_RD_REQ 清除 RD_REQ 中断寄存器.....	635
32.4.20	I2C_TX_ABRT 清除 TX_ABRT 中断寄存器.....	635
32.4.21	I2C_RX_DONE 清除 RX_DONE 中断寄存器.....	636
32.4.22	I2C_ACTIV 清除 ACTIVITY 中断寄存器.....	636
32.4.23	I2C_STOP 清除 STOP_DET 中断寄存器.....	636
32.4.24	I2C_START 清除 START_DET 中断寄存器.....	637
32.4.25	I2C_GC 清除 GEN_CALL 中断寄存器.....	637
32.4.26	I2C_ENR 使能寄存器.....	638
32.4.27	I2C_SR 状态寄存器.....	639
32.4.28	I2C_TXFLR 发送缓冲级别寄存器.....	640
32.4.29	I2C_RXFLR 接收缓冲级别寄存器.....	640
32.4.30	I2C_HOLD SDA 保持时间寄存器.....	641
32.4.31	I2C_TX_ABRT_SRC 传输中止源寄存器.....	641
32.4.32	I2C_SLV_NACK 从接收 NACK 寄存器.....	642
32.4.33	I2C_DMA DMA 控制寄存器.....	643
32.4.34	I2C_SETUP SDA 建立时间寄存器.....	643
32.4.35	I2C_GCR 广播呼叫 ACK 寄存器.....	644
32.4.36	I2C_EN_SR ENABLE 状态寄存器.....	644
32.4.37	I2C_SPKLEN 滤波寄存器.....	645
32.4.38	I2C_SCL_TMO SCL 低电平超时寄存器.....	645
32.4.39	I2C_SDA_TMO SDA 低电平超时寄存器.....	646
32.4.40	I2C_SCL_STUCK 清除 SCL_STUCK 中断寄存器.....	646
32.4.41	I2C_SMB_SEXT SMBus 从器件时钟延长超时寄存器.....	646
32.4.42	I2C_SMB_MEXT SMBus 主器件时钟延长超时寄存器.....	647
32.4.43	I2C_SMB_IDLE SMBus 总线空闲计数寄存器.....	647
32.4.44	I2C_SMB_ISR SMBus 中断状态寄存器.....	648
32.4.45	I2C_SMB_IMR SMBus 中断屏蔽寄存器.....	648
32.4.46	I2C_SMB_RAWISR SMBus RAW 中断寄存器.....	649
32.4.47	I2C_SMB_ICR SMBus 组合和独立中断清除寄存器.....	650
32.4.48	I2C_OPT_SAR 可选从地址寄存器.....	650
32.4.49	I2C_SMB_UDID_LSB SMBus UDID LSB 寄存器.....	651
32.4.50	I2C_SMB_UDID_MSB0 SMBus UDID MSB 寄存器 0.....	651
32.4.51	I2C_SMB_UDID_MSB1 SMBus UDID MSB 寄存器 1.....	652
32.4.52	I2C_SMB_UDID_MSB2 SMBus UDID MSB 寄存器 2.....	652
32.4.53	I2C_SLVMASK 从地址掩码寄存器.....	653
32.4.54	I2C_SLVRCVADDR 从器件接收地址寄存器.....	653
33	FlexCAN 控制器局域网.....	654
33.1	介绍.....	654
33.2	主要特征.....	654
33.3	功能描述.....	655
33.3.1	功能框图.....	655
33.3.2	工作模式.....	656

33.3.3	传输流程	658
33.3.4	仲裁过程	659
33.3.5	接收流程	662
33.3.6	匹配过程	663
33.3.7	转移流程	666
33.3.8	数据一致性	668
33.3.9	Rx FIFO	670
33.3.10	CAN 协议	671
33.3.11	时钟限制	677
33.3.12	复位	677
33.3.13	初始化	678
33.3.14	中断	679
33.4	寄存器/内存映射	679
33.4.1	寄存器/内存映射总览	680
33.4.2	CAN_MCR 模块配置寄存器	681
33.4.3	CAN_CTRL1 控制寄存器 1	684
33.4.4	CAN_TIMER 自由运行计数器寄存器	687
33.4.5	CAN_RXMGMASK Rx 邮箱全局掩码寄存器	687
33.4.6	CAN_RX14MASK Rx 14 掩码寄存器	688
33.4.7	CAN_RX15MASK Rx 15 掩码寄存器	689
33.4.8	CAN_ECR 错误计数器寄存器	689
33.4.9	CAN_ESR1 错误和状态寄存器 1	690
33.4.10	CAN_IMASK1 中断掩码寄存器 1	693
33.4.11	CAN_IFLAG1 中断标志寄存器 1	693
33.4.12	CAN_CTRL2 控制寄存器 2	696
33.4.13	CAN_ESR2 错误和状态 2 寄存器	697
33.4.14	CAN_CRCRC CRC 寄存器	698
33.4.15	CAN_RXFGMASK Rx FIFO 全局掩码寄存器	699
33.4.16	CAN_RXFIR Rx FIFO 信息寄存器	700
33.4.17	CAN_CBT CAN 位时序寄存器	700
33.4.18	CAN_RXIMRn Rx 私有掩码寄存器	702
33.4.19	报文缓冲区结构	702
33.4.20	MB 内存映射	705
33.4.21	Rx FIFO 结构	706
34	USB_FS 全速 USB 控制器	708
34.1	简介	708
34.2	主要特征	708
34.3	功能描述	709
34.3.1	功能框图	709
34.3.2	信号描述	709
34.3.3	USB_FS 典型硬件配置	709
34.3.4	USB_FS 软件编程接口	711
34.3.5	OTG 与 Host 模式操作方式	715
34.3.6	Host 模式操作示例	715

34.3.7	OTG 操作	718
34.3.8	中断	720
34.4	寄存器	721
34.4.1	寄存器总览	721
34.4.2	USB_FS_OTGISTAT OTG 中断状态寄存器	722
34.4.3	USB_FS_OTGICTRL OTG 中断控制寄存器	723
34.4.4	USB_FS_OTGSTAT OTG 状态寄存器	723
34.4.5	USB_FS_OTGCTRL OTG 控制寄存器	724
34.4.6	USB_FS_INTSTAT 中断状态寄存器	725
34.4.7	USB_FS_INTENB 中使能态寄存器	726
34.4.8	USB_FS_ERRSTAT 错误中断状态寄存器	726
34.4.9	USB_FS_ERRENB 错误中断使能寄存器	727
34.4.10	USB_FS_STAT 状态寄存器	728
34.4.11	USB_FS_CTL 控制寄存器	728
34.4.12	USB_FS_ADDR 地址寄存器	729
34.4.13	USB_FS_BDTPAGE1 缓冲区描述符表页寄存器 1	730
34.4.14	USB_FS_FRMNUML 低位帧数寄存器	730
34.4.15	USB_FS_FRMNUMH 高位帧数寄存器	731
34.4.16	USB_FS_TOKEN 令牌寄存器	731
34.4.17	USB_FS_SOFTHLDD SOF 阈值寄存器	731
34.4.18	USB_FS_BDTPAGE2 缓冲区描述符表页寄存器 2	732
34.4.19	USB_FS_BDTPAGE3 缓冲区描述符表页寄存器 3	732
34.4.20	USB_FS_EPCTL0~15 端点控制寄存器 0~15	733
34.4.21	USB_FS_USBCTRL USB 控制寄存器	734
35	ENET 以太网 MAC 控制器	735
35.1	简介	735
35.2	主要特征	735
35.2.1	MAC 子控制器特征	736
35.2.2	DMA 子控制器特征	737
35.2.3	PTP 子控制器特征	737
35.3	功能描述	738
35.3.1	功能框图	738
35.3.2	信号描述	738
35.3.3	连接关系	739
35.3.4	802.3 帧格式	740
35.3.5	MAC 子控制器基本特征	741
35.3.6	MAC 子控制器接收功能	742
35.3.7	MAC 子控制器发送功能	745
35.3.8	MAC 子控制器帧过滤	751
35.3.9	MAC 子控制器回环模式	753
35.3.10	PMT 电源管理	753
35.3.11	MMC 管理计数	756
35.3.12	DMA 子控制器基本特征	757
35.3.13	DMA 子控制器接收功能	760

35.3.14	DMA 子控制器发送功能	762
35.3.15	DMA 子控制器初始化	764
35.3.16	DMA 子控制器描述符	764
35.3.17	PTP 子控制器基本特征	776
35.3.18	PTP 子控制器参考时钟	777
35.3.19	PTP 子控制器报文类型	778
35.3.20	PTP 子控制器接收功能	778
35.3.21	PTP 子控制器发送功能	778
35.3.22	PTP 子控制器时间校准	778
35.3.23	PTP 子控制器初始化	781
35.3.24	PTP 子控制器 PPS 输出	781
35.3.25	中断	782
35.4	寄存器	784
35.4.1	寄存器总览	784
35.4.2	MAC 子控制器寄存器	785
35.4.3	DMA 子控制器寄存器	800
35.4.4	PTP 子控制器寄存器	814
35.4.5	PMT 电源管理寄存器	821
35.4.6	MMC 管理计数寄存器	823
36	DBG 调试支持	830
36.1	介绍	830
36.2	功能描述	830
36.2.1	功能框图	830
36.2.2	SWD 内部上拉与下拉	831
36.2.3	SWJ 调试端口	831
36.3	ID 代码和锁定机制	831
36.3.1	微控制器设备 ID 编码	831
36.3.2	Cortex JEDEC-106 ID 编码	832
36.4	SW 调试端口	832
36.4.1	SW 协议介绍	832
36.4.2	SW 协议序列	832
36.4.3	SW-DP 状态机 (Reset, Idle states, ID code)	833
36.4.4	DP 和 AP 读 / 写访问	833
36.4.5	SW-DP 寄存器	834
36.4.6	SW-AP 寄存器	834
36.5	JTAG 调试端口	834
36.6	MCU 调试模块 (DBGMCU)	835
36.6.1	支持跟踪调试模式	835
36.6.2	支持定时器、看门狗	835
36.7	寄存器	836
36.7.1	寄存器总览	836
36.7.2	DBG_IDCODE ID 编码寄存器	836
36.7.3	DBG_CR 控制寄存器	836
37	Device Electronic Signature 器件电子签名	839

37.1	简介	839
37.2	寄存器	839
37.2.1	UID1 唯一标识码	839
37.2.2	UID2 唯一标识码	840
37.2.3	UID3 唯一标识码	840
38	修订记录	841

附图目录

图 1-1 系统架构框图.....	35
图 4-1 ISP 方式编程流程图.....	57
图 4-2 IAP 编程流程.....	58
图 5-1 电源控制功能框图.....	74
图 5-2 上电复位和掉电复位波形图.....	76
图 5-3 PVD 阈值波形图.....	77
图 6-1 复位功能框图.....	95
图 6-2 时钟树.....	98
图 6-3 高速晶振/陶瓷谐振器.....	99
图 6-4 外部高速输入时钟.....	99
图 6-5 低速晶振/陶瓷谐振器.....	103
图 6-6 外部低速输入时钟.....	104
图 7-1 CRS 功能框图.....	144
图 7-2 CRS 计数器状态图.....	149
图 9-1 EXTI 结构框图.....	167
图 10-1 标准 I/O 端口.....	173
图 10-2 输入浮空/上拉/下拉配置.....	175
图 10-3 输出配置.....	176
图 10-4 复用功能配置.....	177
图 10-5 模拟输入.....	178
图 11-1 DMA 功能框图.....	185
图 12-1 MindSwitch 功能框图.....	197
图 12-2 连接通道组合逻辑操作选项.....	198
图 13-1 CRC_POLY 功能框图.....	206
图 15-1 QSPI 功能框图.....	218
图 15-2 QSPI 数据格式.....	219
图 15-3 QSPI 输入数据采样周期.....	222
图 16-1 FSMC 框图.....	236
图 16-2 NOR FLASH/SRAM 写时序.....	240
图 16-3 NOR FLASH/SRAM 读时序.....	240
图 16-4 8080 写时序.....	241
图 16-5 8080 读时序.....	242
图 16-6 6800 写时序.....	243
图 16-7 6800 读时序.....	243
图 17-1 ADC 系统框图.....	252
图 17-2 数据对齐方式.....	255
图 17-3 校准过程.....	257
图 17-4 单次转换模式时序图.....	260
图 17-5 单周期扫描下使能通道转换时序图（通道方向从低到高）.....	261
图 17-6 单周期扫描下使能通道转换时序图（通道方向从高到低）.....	261
图 17-7 连续扫描模式使能通道转换时序图（通道方向由低到高）.....	262

图 17-8 连续扫描模式使能通道转换时序图（通道方向由高到低）	262
图 17-9 单次转换模式下通道转换时序图	263
图 17-10 单周期扫描下通道转换时序图	264
图 17-11 连续扫描模式下通道转换时序图	265
图 17-12 连续扫描模式下动态更新配置时序图	265
图 17-13 自动注入通道转换单周期扫描通道转换时序图	267
图 17-14 自动注入通道转换连续转换时序图	267
图 17-15 单次过采样模式下通道转换图	268
图 17-16 单周期过采样模式下通道转换图	268
图 17-17 连续过采样模式下通道转换图	269
图 17-18 单周期自动注入过采样模式下通道转换图	269
图 17-19 单周期过采样模式触发复位方式下通道转换图	269
图 17-20 单周期过采样模式触发置位方式下通道转换图	269
图 18-1 DAC 通道功能框图.....	299
图 18-2 DAC 单通道模式的数据保持寄存器的不同数据格式.....	301
图 18-3 DAC 双通道模式的数据保持寄存器的不同数据格式.....	302
图 18-4 DAC 三角波生成.....	302
图 18-5 带三角波生成的 DAC 转换（使能软件触发）	303
图 18-6 DAC LFSR 寄存器算法.....	303
图 18-7 带 LFSR 波形生成的 DAC 转换（使能软件触发）	304
图 18-8 TEN=0 触发关闭时 DAC 转换的时间示意图.....	305
图 19-1 比较器框图.....	316
图 19-2 比较器的迟滞.....	319
图 20-1 TIMx 结构图.....	326
图 20-2 时钟选择	328
图 20-3 外部时钟模式 1 下的控制电路	329
图 20-4 外部时钟模式 2 下的控制电路	329
图 20-5 自动预装载	330
图 20-6 递增计数模式（UDIS=0）	330
图 20-7 递增计数模式（UDIS=1 禁止产生更新事件）	331
图 20-8 递减计数模式（UDIS=0）	331
图 20-9 递减计数模式（UDIS=1 禁止产生更新事件）	331
图 20-10 中央计数模式（UDIS=0）	332
图 20-11 中央计数模式（UDIS=1 禁止产生更新事件）	332
图 20-12 中央对齐模式重复计数时序图	333
图 20-13 边沿对齐模式递增计数时序图	333
图 20-14 边沿对齐模式递减计数时序图	334
图 20-15 TIMx 输入捕获结构图.....	334
图 20-16 PWM 输入模式时序.....	336
图 20-17 比较输出部分结构图	337
图 20-18 比较输出模式，OC1 信号在匹配时翻转.....	338
图 20-19 边沿对齐递增计数时 PWM 模式 1 的波形.....	339
图 20-20 边沿对齐递减计数时 PWM 模式 1 的波形.....	340
图 20-21 中央对齐 PWM 模式 1 的波形.....	341

图 20-22 移相功能示意图	341
图 20-23 使用 COM 事件产生六步 PWM (OSSR = 1)	342
图 20-24 死区插入	343
图 20-25 响应刹车的输出 (OISx=0, OISxN=0)	346
图 20-26 响应刹车的输出 (OISx=0, OISxN=1)	346
图 20-27 响应刹车的输出 (OISx=1, OISxN=0)	346
图 20-28 响应刹车的输出 (OISx=1, OISxN=1)	346
图 20-29 外部事件清除 OCxREF	347
图 20-30 单脉冲模式	347
图 20-31 编码器模式下的计数器时序图	349
图 20-32 IC1FP1 反相编码器接口模式时序图	350
图 20-33 复位模式的控制时序图	350
图 20-34 门控模式下的控制时序图	351
图 20-35 触发器模式下的控制时序图	351
图 20-36 外部时钟模式 2+从模式 (触发模式) 控制时序图	352
图 20-37 定时器间互联	352
图 20-38 使用主定时器作为从定时器的预分频器	353
图 20-39 使用主定时器使能从定时器	353
图 20-40 使用主定时器的更新事件启动从定时器	354
图 20-41 主定时器的 TI1 同步启动主定时器和从定时器	355
图 20-42 (TI1 异或输入) 输入捕获波形图	355
图 20-43 霍尔传感器接口实例	356
图 21-1 TIMx 结构图	392
图 21-2 时钟选择	394
图 21-3 外部时钟模式 1 下的控制电路	395
图 21-4 外部时钟模式 2 下的控制电路	395
图 21-5 自动预装载	396
图 21-6 递增计数模式 (UDIS=0)	396
图 21-7 递增计数模式 (UDIS=1 禁止产生更新事件)	397
图 21-8 递减计数模式 (UDIS=0)	397
图 21-9 递减计数模式 (UDIS=1 禁止产生更新事件)	397
图 21-10 中央计数模式 (UDIS=0)	398
图 21-11 中央计数模式 (UDIS=1 禁止产生更新事件)	398
图 21-12 TIMx 输入捕获结构图	398
图 21-13 PWM 输入模式时序	400
图 21-14 比较输出部分结构图	401
图 21-15 比较输出模式, OC1 信号在匹配时翻转	402
图 21-16 边沿对齐递增计数时 PWM 模式 1 的波形	403
图 21-17 边沿对齐递减计数时 PWM 模式 1 的波形	404
图 21-18 中央对齐 PWM 模式 1 的波形	405
图 21-19 外部事件清除 OCxREF	406
图 21-20 单脉冲模式	406
图 21-21 编码器模式下的计数器时序图	408
图 21-22 IC1FP1 反相编码器接口模式时序图	408

图 21-23 复位模式的控制时序图	409
图 21-24 门控模式下的控制时序图	410
图 21-25 触发器模式下的控制时序图	410
图 21-26 外部时钟模式 2+从模式（触发模式）控制时序图	411
图 21-27 （TI1 异或输入）输入捕获波形图	412
图 22-1 TIMx 结构图	438
图 22-2 时钟选择	440
图 22-3 外部时钟模式 1 下的控制电路	441
图 22-4 外部时钟模式 2 下的控制电路	441
图 22-5 自动预装载	442
图 22-6 递增计数模式（UDIS=0）	442
图 22-7 递增计数模式（UDIS=1 禁止产生更新事件）	443
图 22-8 递减计数模式（UDIS=0）	443
图 22-9 递减计数模式（UDIS=1 禁止产生更新事件）	443
图 22-10 中央计数模式（UDIS=0）	444
图 22-11 中央计数模式（UDIS=1 禁止产生更新事件）	444
图 22-12 TIMx 输入捕获结构图	444
图 22-13 PWM 输入模式时序	446
图 22-14 比较输出部分结构图	447
图 22-15 比较输出模式，OC1 信号在匹配时翻转	448
图 22-16 边沿对齐递增计数时 PWM 模式 1 的波形	449
图 22-17 边沿对齐递减计数时 PWM 模式 1 的波形	450
图 22-18 中央对齐 PWM 模式 1 的波形	451
图 22-19 外部事件清除 OCxREF	452
图 22-20 单脉冲模式	452
图 22-21 编码器模式下的计数器时序图	454
图 22-22 IC1FP1 反相编码器接口模式时序图	454
图 22-23 复位模式的控制时序图	455
图 22-24 门控模式下的控制时序图	456
图 22-25 触发器模式下的控制时序图	456
图 22-26 外部时钟模式 2+从模式（触发模式）控制时序图	457
图 22-27 （TI1 异或输入）输入捕获波形图	458
图 23-1 TIMx 结构图	483
图 23-2 自动预装载	484
图 23-3 递增计数模式（UDIS=0）	484
图 23-4 递增计数模式（UDIS=1 禁止产生更新事件）	485
图 24-1 功能框图	490
图 25-1 功能框图	498
图 25-2 流程框图	500
图 26-1 WWDG 功能框图	507
图 26-2 WWDG 超时时间坐标图	509
图 27-1 RTC 功能框图	513
图 27-2 RTC 秒和闹钟波形图示例，RTC_PRL = 0004，RTC_ALR = 002	515
图 27-3 RTC 溢出波形图示例，RTC_PRL = 0004	515

图 29-1 UART 功能框图	528
图 29-2 UART 时序	530
图 29-3 RX 引脚采样方法	531
图 29-4 发送时状态位变化	533
图 29-5 UART 奇偶校验方框图	538
图 29-6 UART 采样 NACK 信号方框图	539
图 29-7 普通模式下 IrDA 发送和接收图	540
图 30-1 LPUART 功能框图	555
图 30-2 帧时序	557
图 30-3 接收时序	557
图 30-4 发送时序	558
图 31-1 SPI_I2S 功能框图	570
图 31-2 单主和单从应用	571
图 31-3 数据时钟时序图	572
图 31-4 主机接收数据采样点后移调整示意图	573
图 31-5 TI 模式连续传输时序示意图	578
图 31-6 TI 模式非连续传输时序示意图	578
图 31-7 飞利浦标准示意图	582
图 31-8 MSB 对齐标准示意图	583
图 31-9 LSB 对齐标准示意图	583
图 31-10 PCM 标准示意图	584
图 31-11 I2S 从模式下全双工通信（extSD 为输出，SD 为输入）	584
图 31-12 I2S 主模式全双工通信（SD 为输出，extSD 为输入）	585
图 31-13 I2S 主模式半双工通信（SD 为输出或输入）	586
图 31-14 I2S 时钟预分频器示意图	587
图 32-1 I2C 功能框图	604
图 32-2 起始和停止条件	605
图 32-3 7 位地址格式	605
图 32-4 10 位地址格式	606
图 32-5 主发送协议	607
图 32-6 主接收协议	607
图 32-7 带 RESTART（SR）信号的主发送和接收协议	608
图 32-8 起始字节传输	608
图 32-9 I2C_DR 寄存器	609
图 32-10 主发送，TX FIFO 为空	609
图 32-11 主接收，TX FIFO 为空	609
图 32-12 主发送，发生 RESTART	610
图 32-13 主接收，发生 RESTART	610
图 32-14 两个主器件仲裁	610
图 32-15 时钟同步（示意图）	611
图 32-16 时钟同步（时序图）	611
图 32-17 SCL 生成时序	612
图 32-18 流程图（I2C 接口作为从器件）	614
图 32-19 流程图（I2C 接口作为主器件）	616

图 32-20 滤波示例	617
图 32-21 SDA 恢复	618
图 32-22 I2C 中断机制	621
图 33-1 FlexCAN 框图	656
图 33-2 PE 时钟结构	673
图 33-3 位时间内的段（使用 CAN_CTRL1 位时序变量的经典 CAN 格式）	674
图 33-4 匹配和移入时间段	675
图 33-5 仲裁和移出时间段	675
图 33-6 总线关闭结束时的仲裁和移出时间段	676
图 33-7 优化的 Tx 仲裁起点	676
图 33-8 信息寄存器结构示例	702
图 33-9 Rx FIFO 结构	706
图 33-10 ID 过滤表结构	707
图 34-1 USB_FS 功能框图	709
图 34-2 仅主机硬件连接示意图	710
图 34-3 仅从设备硬件连接示意图	710
图 34-4 OTG 硬件连接示意图	711
图 34-5 缓冲区描述表	712
图 34-6 USB 令牌传输示意图	715
图 34-7 OTG A-device 操作方式	718
图 34-8 OTG B-device 操作方式	720
图 35-1 ENET 功能框图	738
图 35-2 MII 接口和 Device PHY 连接图	740
图 35-3 以太网帧格式	741
图 35-4 无错误时接收数据时序图	744
图 35-5 有错误时接收数据时序图	745
图 35-6 假载波指示接收时序图	745
图 35-7 无错误时发送数据时序图	750
图 35-8 有错误时发送数据时序图	751
图 35-9 远程唤醒帧过滤寄存器	754
图 35-10 ENET 描述符结构	758
图 35-11 PTP 时钟同步图	777
图 35-12 系统时间校准	779
图 35-13 ENET 中断控制框图	782
图 35-14 DMA 中断框图	783
图 36-1 调试功能框图	830

表格目录

表 2-1 存储器映像	37
表 3-1 异常向量表	41
表 3-2 中断向量表	41
表 3-3 EXTI 触发源	44
表 3-4 DMA1 通道分配	45
表 3-5 DMA2 通道分配	46
表 3-6 MindSwitch 触发源映射	46
表 3-7 MindSwitch 触发目标映射	48
表 3-8 MindSwitch 连接源映射	48
表 3-9 MindSwitch 连接目标映射	49
表 3-10 ADC 通道分配	49
表 3-11 TIM 之间的互联	50
表 3-12 比较器控制 TIM	50
表 3-13 刹车控制	50
表 3-14 时钟作为 TIM 输入信号	50
表 3-15 启动模式	51
表 4-1 Flash 存储空间	52
表 4-2 信息块	53
表 4-3 选项字节组织结构	53
表 4-4 USER 的位含义	53
表 4-5 Latency 设置关系	55
表 4-6 编程方式	55
表 4-7 保护设置的状态变化	59
表 4-8 Flash 读保护状态	67
表 4-9 Flash 解除读保护状态	67
表 4-10 写保护区域	68
表 4-11 FLASH 寄存器概览	68
表 5-1 低功耗模式列表	79
表 5-2 低功耗运行模式	81
表 5-3 SLEEPNOW 模式	82
表 5-4 SLEEPONEXIT 模式	82
表 5-5 Low Power SLEEPNOW 模式	82
表 5-6 Low Power SLEEPONEXIT 模式	83
表 5-7 停机模式	84
表 5-8 深度停机模式	85
表 5-9 待机模式	86
表 5-10 电源控制寄存器概览	88
表 6-1 RCC 全局中断表	105
表 6-2 MCO 与时钟源对应关系	107
表 6-3 RCC 寄存器概览	109
表 7-1 CRS 中断请求	150

表 7-2 CRS 寄存器概览	151
表 8-1 SYSCFG 寄存器概览	156
表 9-1 EXTI 寄存器总览	168
表 10-1 端口位配置表 (port0 为例)	173
表 10-2 SWD 复用功能重映射	178
表 10-3 GPIO 寄存器概览	178
表 11-1 DMA 中断请求	186
表 11-2 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	189
表 11-3 DMA 寄存器概览	190
表 12-1 MindSwitch 寄存器概览	199
表 13-1 CRC 寄存器概览	208
表 14-1 CORDIC 寄存器概览	212
表 15-1 QSPI 信号描述表	218
表 15-2 QSPI 中断请求	224
表 15-3 QSPI DMA 请求	224
表 15-4 QSPI 寄存器概览	224
表 16-1 接口描述	236
表 16-2 NOR FLASH 控制器	237
表 16-3 BANK 地址划分	237
表 16-4 存储器位宽	237
表 16-5 NOR FLASH 16bit 示例	238
表 16-6 SRAM 16bit 示例	238
表 16-7 NOR FLASH/SRAM 接口	239
表 16-8 NOR FLASH/SRAM 时序参数说明	239
表 16-9 8080 协议接口	240
表 16-10 8080 地址说明	241
表 16-11 8080 时序参数说明	241
表 16-12 6800 协议接口	242
表 16-13 6800 地址说明	242
表 16-14 6800 时序参数说明	243
表 16-15 寄存器映射表	243
表 17-1 寄存器概览	271
表 18-1 DAC 管脚说明	299
表 18-2 DAC 寄存器概览	307
表 19-1 COMP 寄存器概览	319
表 20-1 数字滤波器宽度与 ICxF 的对应关系表	335
表 20-2 死区时间计算	342
表 20-3 当 MOE=1, OSSI=0/1, OSSR=0 时:	344
表 20-4 当 MOE=1, OSSI=0/1, OSSR=1 时:	344
表 20-5 当 MOE=0, OSSI=0, OSSR=0/1 时:	344
表 20-6 当 MOE=0, OSSI=1, OSSR=0/1 时:	345
表 20-7 计数方向与编码器信号的关系	349
表 20-8 中断事件一览表	357
表 20-9 TIMx 寄存器概览	358

表 20-10 ICx 极性/电平选择表	379
表 21-1 数字滤波器宽度与 ICxF 的对应关系表	399
表 21-2 计数方向与编码器信号的关系	407
表 21-3 中断事件一览表	412
表 21-4 TIMx 寄存器概览	413
表 21-5 ICx 极性/电平选择表	432
表 22-1 数字滤波器宽度与 ICxF 的对应关系表	445
表 22-2 计数方向与编码器信号的关系	453
表 22-3 中断事件一览表	458
表 22-4 TIMx 寄存器概览	459
表 22-5 ICx 极性/电平选择表	478
表 23-1 中断事件一览表	485
表 23-2 TIMx 寄存器概览	485
表 24-1 LPTIM 寄存器概览	492
表 25-1 IWDG 超时时间（以 LSI 时钟频率 40kHz 为例）	500
表 25-2 IWDG 寄存器概览	501
表 26-1 超时时间（ $f_{pclk1}=45\text{MHz}$ ）	509
表 26-2 WWDG 寄存器概览	509
表 27-1 RTC 寄存器概览	516
表 28-1 BKP 寄存器概览	525
表 29-1 UART 中断请求	540
表 29-2 UART 寄存器概览	541
表 30-1 LPUART 中断请求	556
表 30-2 调制控制寄存器 MCTL 建议配置参数表	558
表 30-3 调制控制寄存器 MCTL 计算方式	558
表 30-4 LPUART 寄存器概览	562
表 31-1 波特率公式	576
表 31-2 SPI 状态	579
表 31-3 I2S 引脚列表	581
表 31-4 I2S 比特率计算公式	587
表 31-5 音频采样率与 F_{I2SCLK} （APB 时钟频率）计算关系	587
表 31-6 I2S 音频采样率误差表	588
表 31-7 I2S 中断状态	588
表 31-8 SPI_I2S 寄存器概览	589
表 32-1 引脚定义	604
表 32-2 I2C 首字节	606
表 32-3 SMBus 和 I2C	618
表 32-4 中断位的置位和清除	620
表 32-5 I2C 寄存器概览	622
表 32-6 DISSLAVE 和 MASTER 配置	625
表 33-1 本地优先级无效时仲裁值的构成	660
表 33-2 本地优先级使能时仲裁值的构成	660
表 33-3 匹配架构	664
表 33-4 匹配可能性及由此产生的接收结构	665

表 33-5 时间段语法	674
表 33-6 符合 Bosch CAN 2.0B 标准的位时间段设置	674
表 33-7 经典 CAN 格式 CAN 位的最少外设时钟数	677
表 33-8 寄存器访问和复位信息	680
表 33-9 FlexCAN 寄存器/内存映射概览.....	680
表 33-10 MG 位掩码邮箱过滤字段.....	688
表 33-11 FLEXCAN 状态相关位编码.....	693
表 33-12 Rx FIFO 过滤器数量.....	697
表 33-13 Rx FIFO 全局掩码位和 IDF 字段的对应关系.....	699
表 33-14 接收缓冲区代码	703
表 33-15 发送缓冲区代码	703
表 33-16 有效数据字节	705
表 33-17 MB 内存映射.....	705
表 34-1 术语表	708
表 34-2 USB 信号描述表	709
表 34-3 BDT 地址计算	713
表 34-4 BDT 地址计算域说明	713
表 34-5 BD 格式	713
表 34-6 OTG A-device 操作状态描述表.....	719
表 34-7 OTG B-device 操作状态描述表.....	720
表 34-8 USB 寄存器概览	721
表 35-1 ENET 功能引脚说明	739
表 35-2 常规型 RX 描述符	764
表 35-3 增强型 RX 描述符	768
表 35-4 常规型 TX 描述符.....	771
表 35-5 增强型 TX 描述符.....	775
表 35-6 PTP 报文类型	778
表 35-7 ENET 寄存器概览.....	784
表 36-1 SWJ 调试端口管脚	831
表 36-2 ID 编码.....	831
表 36-3 8bit 请求包.....	832
表 36-4 3bit 应答包.....	833
表 36-5 33bit 数据包.....	833
表 36-6 SW-DP 寄存器.....	834
表 36-7 JTAG 调试端口寄存器映射.....	834
表 36-8 JTAG-DP 寄存器	835
表 36-9 DBG 寄存器概览.....	836
表 37-1 存储器容量寄存器概览	839
表 38-1 修订记录	841

1 系统和存储器架构

1.1 系统架构

MM32F5270 和 MM32F5280 是基于安谋科技的 Armv-8 架构的 STAR-MC1 处理器（兼容 Cortex-M33）开发的 32 位微控制器产品，内置 L1 ICache、DCache、指令紧密耦合存储器 ITCM 和数据紧密耦合存储器 DTCM，同时具备高性能和低功耗的特点。

MM32F5270 和 MM32F5280 采用 multi-AHB 总线矩阵结构，实现主控总线到被控总线的访问。Multi-AHB 总线矩阵可连接：

- 6 个总线主控器
 - ◆ CBUS（Code AHB 总线）
 - ◆ SBUS（System AHB 总线）
 - ◆ DMA1
 - ◆ DMA2
 - ◆ OTG FS（USB2.0 OTG FS 内置 DMA）
 - ◆ ENET（Ethernet 内置 DMA）
- 8 个总线从控制器
 - ◆ 片内闪存 eFlash
 - ◆ SRAM1（112KB）
 - ◆ SRAM2（16KB）
 - ◆ TBUS（用于访问 ITCM 和 DTCM）
 - ◆ AHB1 外设
 - ◆ AHB2 外设
 - ◆ FSMC
 - ◆ QSPI

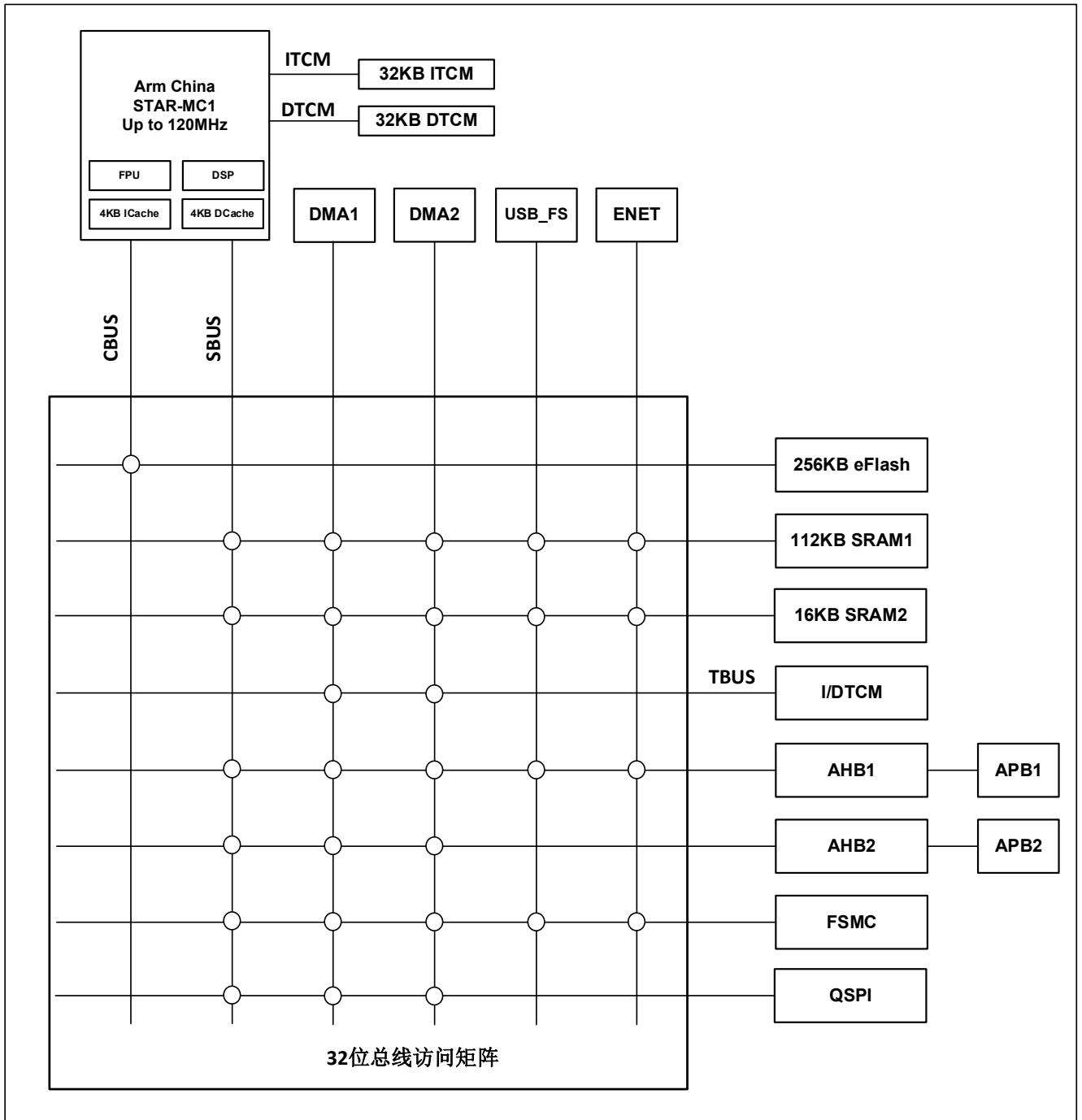


图 1-1 系统架构框图

1.1.1 CBUS 总线

CBUS 总线主控制器用于 STAR-MC1 对内部代码区域的指令预取和数据访问。

1.1.2 SBUS 总线

SBUS 总线用于 STAR-MC1 对 SRAM、外设、外部存储等区域的指令预取和数据访问。

1.1.3 DMA 总线

DMA 总线主控制器用于 DMA 对 SRAM、外部存储区域的数据访问。

1.1.4 USB_FS 总线

OTG FS 总线用于 OTG FS 内置 DMA 对 SRAM 区域的数据访问。

1.1.5 ENET 总线

ENET 总线用于 ENET 内置 DMA 对 SRAM 区域的数据访问。

1.1.6 TBUS 总线

TBUS 总线用于 DMA 访问 ITCM 和 DTCM。

1.1.7 ITCM / DTCM 总线

ITCM / DTCM 总线为 STAR-MC1 私有总线，用于 STAR-MC1 访问指令 TCM 和数据 DTCM。

1.1.8 Multi-AHB 总线矩阵

Multi-AHB 总线可以实现主控总线到被控总线的访问，基于软件可配置优先级的仲裁管理可以实现多个主控总线的并行访问。

1.1.9 AHB2APB 桥

AHB2APB 桥在 AHB 与 APB 总线间提供同步连接。在每次复位之后，所有的外设时钟处于关闭（除了 SRAM 及 Flash 外）状态。在用一下外设前，用户必须打开相应的时钟使能位。另外当对 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。

1.2 内核简介

本产品搭载安谋科技授权的“星辰”STAR-MC1 处理器（兼容 Cortex-M33）。该处理器是基于 Armv8-M Mainline 指令集架构（Instruction-Set Architecture, ISA）的 32 位处理器，内置单精度浮点运算单元（Floating Point Unit, FPU），并支持数字信号处理（Digital Signal Processing, DSP）扩展，提供实时处理和高级中断处理能力，实现性能和电源效率的平衡，适合用于各种类型的实时控制应用。

本产品配置了 4KB 的一级指令缓存（L1 I-Cache）和 4KB 的一级数据缓存（L1 D-Cache），一级缓存与处理器紧密耦合，可显著提高在嵌入式 Flash 或外部存储器中运行代码时的执行效率。

2 地址映射

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

2.1 存储器映像和寄存器编址

存储器映像请参考各外设对应章节的存储器映像图

表 2-1 存储器映像

总线	编址范围	大小	外设
FLASH	0x0000 0000 - 0x0000 7FFF	32 KB	ITCM
	0x0000 8000 - 0x000F FFFF	992 KB	保留
	0x0010 0000 - 0x07FF FFFF	127 MB	保留
	0x0800 0000 - 0x0803 FFFF	256 KB	片内 FLASH 存储器
	0x0804 0000 - 0x081F FFFF	1792 KB	保留
	0x0820 0000 - 0x0FFF FFFF	126 MB	保留
	0x1000 0000 - 0x100D FFFF	896 KB	保留
	0x100E 0000 - 0x1FFD FFFF	255 MB	保留
	0x1FFE 0000 - 0x1FFE 01FF	0.5 KB	保留
	0x1FFE 0200 - 0x1FFE 0FFF	3.5 KB	保留
	0x1FFE 1000 - 0x1FFE 23FF	5 KB	Security memory
	0x1FFE 2400 - 0x1FFF E7FF	113 KB	保留
	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	保留
SRAM	0x2000 0000 - 0x2000 7FFF	32 KB	DTCM
	0x2000 8000 - 0x200F FFFF	992 KB	保留
	0x2010 8000 - 0x2FFF FFFF	255 MB	保留
	0x3000 0000 - 0x3001 BFFF	112 KB	SRAM-1
	0x3001 C000 - 0x3001 FFFF	16 KB	SRAM-2
	0x3002 0000 - 0x300F FFFF	896 KB	保留
	0x3010 0000 - 0x3FFF FFFF	255 MB	保留
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

总线	编址范围	大小	外设
	0x4000 1800 - 0x4000 27FF	4 KB	保留
	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	保留
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2
	0x4000 3C00 - 0x4000 3FFF	1 KB	SPI3
	0x4000 4000 - 0x4000 43FF	1 KB	保留
	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 6BFF	4 KB	保留
	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 BFFF	17 KB	保留
	0x4000 C000 - 0x4000 FFFF	16 KB	FLEXCAN1
APB2	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI
	0x4001 0800 - 0x4001 0BFF	1 KB	LPUART
	0x4001 0C00 - 0x4001 23FF	6 KB	保留
	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 7FFF	15 KB	保留
	0x4001 8000 - 0x4001 BFFF	16 KB	FLEXCAN2
	0x4001 C000 - 0x4001 CFFF	4 KB	保留
	0x4001 D000 - 0x4001 D3FF	1 KB	LPTIM
	0x4001 D400 - 0x4001 FBFF	10 KB	保留
	0x4001 FC00 - 0x4001 FFFF	1 KB	Mindswitch
AHB1	0x4002 0000 - 0x4002 03FF	1 KB	DMA1
	0x4002 0400 - 0x4002 07FF	1 KB	DMA2
	0x4002 0800 - 0x4002 0FFF	2 KB	保留

总线	编址范围	大小	外设
	0x4002 1000 - 0x4002 13FF	1 KB	RCC
	0x4002 1400 - 0x4002 1FFF	3 KB	保留
	0x4002 2000 - 0x4002 23FF	1 KB	Flash memory interface
	0x4002 2400 - 0x4002 2FFF	3 KB	保留
	0x4002 3000 - 0x4002 33FF	1 KB	CRC
	0x4002 3400 - 0x4002 7FFF	19 KB	保留
	0x4002 8000 - 0x4002 9FFF	8 KB	ENET
	0x4002 A000 - 0x4002 A3FF	1 KB	CORDIC
	0x4002 A400 - 0x4003 FFFF	87 KB	保留
	0x4004 0000 - 0x4004 03FF	1 KB	Port A
	0x4004 0400 - 0x4004 07FF	1 KB	Port B
	0x4004 0800 - 0x4004 0BFF	1 KB	Port C
	0x4004 0C00 - 0x4004 0FFF	1 KB	Port D
	0x4004 1000 - 0x4004 13FF	1 KB	Port E
	0x4004 1400 - 0x4004 17FF	1 KB	Port F
	0x4004 1800 - 0x4004 1BFF	1 KB	Port G
	0x4004 1C00 - 0x4004 1FFF	1 KB	Port H
	0x4004 2000 - 0x4004 23FF	1 KB	Port I
	0x4004 2400 - 0x400F FFFF	759 KB	保留
	0x4010 0000 - 0x4FFF FFFF	255 MB	保留
AHB2	0x5000 0000 - 0x5003 FFFF	256 KB	USB_FS
	0x5004 0000 - 0x500F FFFF	768 KB	保留
	0x5010 0000 - 0x5FFF FFFF	255 MB	保留
AHB3	0x6000 0000 - 0x63FF FFFF	64 MB	FSMC Bank1
	0x6400 0000 - 0x67FF FFFF	64 MB	FSMC Bank2
	0x6800 0000 - 0x6BFF FFFF	64 MB	FSMC Bank3
	0x6C00 0000 - 0x6FFF FFFF	64 MB	FSMC Bank4
	0x7000 0000 - 0x8FFF FFFF	512 MB	保留
AHB4	0x9000 0000 - 0x9FFF FFFF	256 MB	QSPI
	0xA000 0000 - 0xA000 0FFF	4 KB	FSMC Register
	0xA000 1000 - 0xA000 13FF	1 KB	QSPI Register
	0xA000 1400 - 0xA00F FFFF	1019 KB	保留
	0xA010 0000 - 0xDFFF FFFF	1023 MB	保留

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 0x3000 0000。

SRAM 可以被 CPU 或者 DMA 用最快的系统时钟且不插入任何等待进行访问。

2.3 内置的 TCM SRAM

内置 32K 字节指令 TCM SRAM 和 32K 字节数据 TCM SRAM。指令 TCM (ITCM) 起始地址为 0x0000 0000，数据 TCM (DTCM) 起始地址为 0x2000 0000。

TCM SRAM 可以被 CPU 或者 DMA 用最快的系统时钟且不插入任何等待进行访问。

2.4 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。

3 芯片特定配置

3.1 中断向量表

在 Handler 模式下，Star-MC1 处理器与内嵌中断向量控制（NVIC）对所有的异常进行优先级区分处理。当异常发生时，系统会将当前处理的工作压栈，执行完中断服务程序后出栈。取向量与当前工作的压栈并行进行的，提高了中断的效率，下表分别列出了异常向量与中断向量。

表 3-1 异常向量表

位置	优先级	优先级类别	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-4	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断/RCC时钟安全系统（CSS）连接到NMI向量	0x0000_0008
	-1	固定	硬件失效（HardFault）	所有类型的失效	0x0000_000C
	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
	-	-	-	保留	0x0000_0020
	-	-	-	保留	0x0000_0024
	-	-	-	保留	0x0000_0028

表 3-2 中断向量表

位置	优先级	优先级类别	名称	说明	地址
	3	可设置	SVCall	通过SWI指令的系统服务调用	0x0000_002C
	4	可设置	调试监（DebugMonitor）	调试监控器	0x0000_0030
	-	可设置	-	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG_IWDG	看门狗中断（IWDG为EXTI21）	0x0000_0040
1	8	可设置	PVD	电源电压检测（PVD）中断（EXTI16）	0x0000_0044
2	9	可设置	BKP_TAMPER	BKP侵入检测中断	0x0000_0048
3	10	可设置	RTC	RTC全局中断	0x0000_004C

位置	优先级	优先级类别	名称	说明	地址
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_CH1	DMA1通道1全局中断	0x0000_006C
12	19	可设置	DMA1_CH2	DMA1通道2全局中断	0x0000_0070
13	20	可设置	DMA1_CH3	DMA1通道3全局中断	0x0000_0074
14	21	可设置	DMA1_CH4	DMA1通道4全局中断	0x0000_0078
15	22	可设置	DMA1_CH5	DMA1通道5全局中断	0x0000_007C
16	23	可设置	DMA1_CH6	DMA1通道6全局中断	0x0000_0080
17	24	可设置	DMA1_CH7	DMA1通道7全局中断	0x0000_0084
18	25	可设置	ADC1/2	ADC1/2全局中断	0x0000_0088
19	26	-	-	保留	0x0000_008C
20	27	可设置	DMA1_CH8	DMA1通道8全局中断	0x0000_0090
21	28	可设置	FlexCAN1	FlexCAN1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
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_ALR	RTC闹钟中断（EXTI17）	0x0000_00E4
42	49	可设置	USB_WKUP	USB唤醒中断（EXTI18）	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

位置	优先级	优先级类别	名称	说明	地址
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	-	-	保留	0x0000_00FC
48	55	-	-	保留	0x0000_0100
49	56	-	-	保留	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_CH1	DMA2通道1全局中断	0x0000_0120
57	64	可设置	DMA2_CH2	DMA2通道2全局中断	0x0000_0124
58	65	可设置	DMA2_CH3	DMA2通道3全局中断	0x0000_0128
59	66	可设置	DMA2_CH4	DMA2通道4全局中断	0x0000_012C
60	67	可设置	DMA2_CH5	DMA2通道5全局中断	0x0000_0130
61	68	可设置	ENET	ENET全局中断	0x0000_0134
62	69	可设置	ENET_WKUP	ENET唤醒中断（EXTI25）	0x0000_0138
63	70	-	-	保留	0x0000_013C
64	71	可设置	COMP	比较器1/2/3全局中断 （EXTI19/20/24）	0x0000_0140
65	72	可设置	FlexCAN2	FLexCAN2全局中断	0x0000_0144
66	73	-	-	保留	0x0000_0148
67	74	可设置	USB_FS	USB_FS全局中断	0x0000_014C
68	75	可设置	DMA2_CH6	DMA2通道6全局中断	0x0000_0150
69	76	可设置	DMA2_CH7	DMA2通道7全局中断	0x0000_0154
70	77	可设置	DMA2_CH8	DMA2通道8全局中断	0x0000_0158
71	78	可设置	UART6	UART6全局中断	0x0000_015C
72	79	-	-	保留	0x0000_0160
73	80	-	-	保留	0x0000_0164
74	81	-	-	保留	0x0000_0168
75	82	-	-	保留	0x0000_016C
76	83	-	-	保留	0x0000_0170
77	84	-	-	保留	0x0000_0174
78	85	-	-	保留	0x0000_0178
79	86	-	-	保留	0x0000_017C
80	87	-	-	保留	0x0000_0180
81	88	-	-	保留	0x0000_0184
82	89	可设置	UART7	UART7全局中断	0x0000_0188
83	90	-	-	保留	0x0000_018C
84	91	-	-	保留	0x0000_0190

位置	优先级	优先级类别	名称	说明	地址
85	92	-	-	保留	0x0000_0194
86	93	-	-	保留	0x0000_0198
87	94	-	-	保留	0x0000_019C
88	95	-	-	保留	0x0000_01A0
89	96	-	-	保留	0x0000_01A4
90	97	-	-	保留	0x0000_01A8
91	98	-	-	保留	0x0000_01AC
92	99	-	-	保留	0x0000_01B0
93	100	-	-	保留	0x0000_01B4
94	101	-	-	保留	0x0000_01B8
95	102	可设置	QSPI	QSPI全局中断	0x0000_01BC
96	103	-	-	保留	0x0000_01C0
97	104	-	-	保留	0x0000_01C4
98	105	-	-	保留	0x0000_01C8
99	106	-	-	保留	0x0000_01CC
100	107	-	-	保留	0x0000_01D0
101	108	-	-	保留	0x0000_01D4
102	109	可设置	LPTIM	LPTIM全局中断（EXTI22）	0x0000_01D8
103	110	-	-	保留	0x0000_01DC
104	111	可设置	LPUART	LPUART全局中断（EXTI23）	0x0000_01E0
105	112	-	-	保留	0x0000_01E4
106	113	-	-	保留	0x0000_01E8
107	114	-	-	保留	0x0000_01EC

3.2 外部中断映射表

GPIO 对应的 16 个外部中断/事件映射关系如下表所示：

表 3-3 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PX0 (X=A,B,C,D,E,F,G,H,I)	SYSCFG_EXTICR1 寄存器中的 EXTI0
EXTI1	PX1 (X=A,B,C,D,E,F,G,H,I)	SYSCFG_EXTICR1 寄存器中的 EXTI1
EXTI2	PX2 (X=A,B,C,D,E,F,G,H)	SYSCFG_EXTICR1 寄存器中的 EXTI2
EXTI3	PX3 (X=A,B,C,D,E,F,G,H)	SYSCFG_EXTICR1 寄存器中的 EXTI3
EXTI4	PX4 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR2 寄存器中的 EXTI4
EXTI5	PX5 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR2 寄存器中的 EXTI5
EXTI6	PX6 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR2 寄存器中的 EXTI6
EXTI7	PX7 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR2 寄存器中的 EXTI7
EXTI8	PX8 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR3 寄存器中的 EXTI8
EXTI9	PX9 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR3 寄存器中的 EXTI9
EXTI10	PX10 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR3 寄存器中的 EXTI10

外部中断线	IO 映射	控制位
EXTI11	PX11 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR3 寄存器中的 EXTI11
EXTI12	PX12 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR4 寄存器中的 EXTI12
EXTI13	PX13 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR4 寄存器中的 EXTI13
EXTI14	PX14 (X=A,B,C,D,E,F,G)	SYSCFG_EXTICR4 寄存器中的 EXTI14
EXTI15	PX15 (X=A,B,C,D,E,F,G)	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 18 连接到 USB_SUSPEND；
- EXTI 线 19 连接到 COMP1 输出；
- EXTI 线 20 连接到 COMP2 输出；
- EXTI 线 21 连接到 IWDG 输出；
- EXTI 线 22 连接到 LPTIM 输出；
- EXTI 线 23 连接到 LPUART 输出；
- EXTI 线 24 连接到 COMP3 输出；
- EXTI 线 25 连接到 ENET 输出；

3.3 DMA 通道分配

MM32F5270 和 MM32F5280 支持两个独立 DMA。

DMA 控制器从外设 TIMx、ADCx、SPIx、QSPI、I2S、I2Cx 和 UARTx 产生的请求，通过逻辑或输入到 DMA 控制器，为了避免冲突，在一个通道中，同时只能有一个外设 DMA 请求有效。参见下图的 DMA 请求映像。

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

表 3-4 DMA1 通道分配

外设	通道1	通道2	通道3	通道4	通道5	通道6	通道7	通道8
ADC	ADC1	ADC2						
DAC								
SPI		SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX			
UART	UART6_RX(0)	UART3_TX	UART3_RX	UART1_TX	UART1_RX	UART2_RX	UART2_TX	UART6_TX(1)
I2C				I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX	
TIM1		TIM1_CC1	TIM1_CC2	TIM1_CC4 TIM1_TRIG(0) TIM1_COM(0)	TIM1_UP	TIM1_CC3	TIM1_TRIG(1)	TIM1_COM(1)
TIM2	TIM2_CC3(0)	TIM2_UP(0)	TIM2_UP(1) TIM2_CC3(1)		TIM2_CC1		TIM2_CC2 TIM2_CC4(0)	TIM2_UP(2) TIM2_CC4(1)

外设	通道1	通道2	通道3	通道4	通道5	通道6	通道7	通道8
TIM3		TIM3_CC3	TIM3_CC4 TIM3_UP			TIM3_CC1 TIM3_TRIG		TIM3_CC2
TIM4	TIM4_CC1			TIM4_CC2	TIM4_CC3		TIM4_UP	TIM4_CC4
TIM5								
TIM6								
TIM7								
TIM8								
FELXCAN								FlexCAN1_RX
QSPI								

表 3-5 DMA2 通道分配

外设	通道1	通道2	通道3	通道4	通道5	通道6	通道7	通道8
ADC								
DAC			DAC_CH1(0)	DAC_CH2(0)			DAC_CH1(1) DAC_CH2(1)	
SPI	SPI3_RX	SPI3_TX						
UART	UART5_RX UART7_RX(0)	UART5_TX UART7_TX(0)	UART4_RX(0)	UART6_TX(0) LPUART_TX	UART4_TX(0)	UART6_RX(1) LPUART_RX	UART7_RX(1) UART4_RX(1)	UART7_TX(1) UART4_TX(1)
I2C								
TIM1								
TIM2								
TIM3								
TIM4								
TIM5	TIM5_CC4 TIM5_TRIG	TIM5_CC3 TIM5_UP(0)		TIM5_CC2	TIM5_CC1	TIM5_UP(1)		
TIM6			TIM6_UP(0)			TIM6_UP(1)		
TIM7				TIM7_UP(0)		TIM7_UP(1)		
TIM8	TIM8_CC3 TIM8_UP(0)	TIM8_CC4 TIM8_TRIG(0) TIM8_COM(0)	TIM8_CC1		TIM8_CC2	TIM8_UP(1)	TIM8_TRIG(1)	TIM8_COM(1)
FELXCAN						FlexCAN2_RX		
QSPI								QSPI

注：当 SYSCFG_DMARMP 寄存器的对应位设定为括号中所示数字时，即为选择该通道的 DMA 功能。

“(0)”表示 SYSCFG_DMARMP 寄存器相关映射位被复位，即默认的 DMA 请求映射在这个通道。

3.4 MindSwitch 映射关系

表 3-6 MindSwitch 触发源映射

序号	触发源	类型
0	低电平	-

序号	触发源	类型
1	高电平	-
2	软件触发	脉冲
3	TIM1 CC1 触发	脉冲
4	TIM1 CC2 触发	脉冲
5	TIM1 CC3 触发	脉冲
6	TIM1 CC4 触发	脉冲
7	TIM1 CC5 触发	脉冲
8	TIM2 CC1 触发	脉冲
9	TIM2 CC2 触发	脉冲
10	TIM2 CC3 触发	脉冲
11	TIM2 CC4 触发	脉冲
12	TIM3 CC1 触发	脉冲
13	TIM3 CC2 触发	脉冲
14	TIM3 CC3 触发	脉冲
15	TIM3 CC4 触发	脉冲
16	TIM4 CC1 触发	脉冲
17	TIM4 CC2 触发	脉冲
18	TIM4 CC3 触发	脉冲
19	TIM4 CC4 触发	脉冲
20	TIM5 CC1 触发	脉冲
21	TIM5 CC2 触发	脉冲
22	TIM5 CC3 触发	脉冲
23	TIM5 CC4 触发	脉冲
24	TIM8 CC1 触发	脉冲
25	TIM8 CC2 触发	脉冲
26	TIM8 CC3 触发	脉冲
27	TIM8 CC4 触发	脉冲
28	TIM8 CC5 触发	脉冲
29	TIM1 触发输出	脉冲
30	TIM2 触发输出	脉冲
31	TIM3 触发输出	脉冲
32	TIM4 触发输出	脉冲
33	TIM5 触发输出	脉冲
34	保留	--
35	保留	--
36	TIM8 触发输出	脉冲
37	外部事件 EXTI0	脉冲
38	外部事件 EXTI1	脉冲
39	外部事件 EXTI2	脉冲
40	外部事件 EXTI3	脉冲
41	外部事件 EXTI4	脉冲
42	外部事件 EXTI5	脉冲

序号	触发源	类型
43	外部事件 EXTI6	脉冲
44	外部事件 EXTI7	脉冲
45	外部事件 EXTI8	脉冲
46	外部事件 EXTI9	脉冲
47	外部事件 EXTI10	脉冲
48	外部事件 EXTI11	脉冲
49	外部事件 EXTI12	脉冲
50	外部事件 EXTI13	脉冲
51	外部事件 EXTI14	脉冲
52	外部事件 EXTI15	脉冲

注：Mindswitch 通道触发源选择寄存器位 MDS_TRIGxCR: SRCSEL 控制触发源选择。

表 3-7 MindSwitch 触发目标映射

序号 x	触发目标	类型
0	ADC1 外部触发	触发
1	ADC1 注入模式外部触发	触发
2	ADC2 外部触发	触发
3	ADC2 注入模式外部触发	触发
6	DAC1 外部触发	触发
7	DAC2 外部触发	触发

注：每个触发目标由 Mindswitch 通道触发源选择寄存器 MDS_TRIGxCR 控制，

x 对应上表中 MM32F5270 和 MM32F5280 所支持的序号。

表 3-8 MindSwitch 连接源映射

序号	连接源	类型
0	低电平	-
1	高电平	-
2	TIM1 CH1 输出	电平
3	TIM1 CH1N 输出	电平
4	TIM1 CH2 输出	电平
5	TIM1 CH2N 输出	电平
6	TIM1 CH3 输出	电平
7	TIM1 CH3N 输出	电平
8	TIM1 CH4 输出	电平
9	TIM1 CH4N 输出	电平
10	TIM2 CH1 输出	电平
11	TIM2 CH2 输出	电平
12	TIM2 CH3 输出	电平
13	TIM2 CH4 输出	电平
14	TIM3 CH4 输出	电平
15	TIM4 CH4 输出	电平

序号	连接源	类型
16	TIM5 CH4 输出	电平
17	TIM8 CH4 输出	电平
18	MDS_IN0	电平
19	MDS_IN1	电平
20	MDS_IN2	电平
21	MDS_IN3	电平

注：Mindswitch 通道连接源选择寄存器位 MDS_CONNxCR: SRCSEL 控制连接源选择。

表 3-9 MindSwitch 连接目标映射

连接序号	连接目标	类型
0	COMP1 blanking	连接
1	COMP2 blanking	连接
2	COMP3 blanking	连接
3	MDS_OUT0	连接
4	MDS_OUT1	连接
5	MDS_OUT2	连接
6	MDS_OUT3	连接

注：每个连接目标由 Mindswitch 通道连接源选择寄存器 MDS_CONNxCR 控制，x 对应上表中 MM32F5270 和 MM32F5280 所支持的序号。

3.5 ADC 通道分配

表 3-10 ADC 通道分配

转换输入通道	ADC1 通道分配	ADC2 通道分配
通道 0	ADC12_IN0	ADC12_IN0
通道 1	ADC12_IN1	ADC12_IN1
通道 2	ADC12_IN2	ADC12_IN2
通道 3	ADC12_IN3	ADC12_IN3
通道 4	ADC12_IN4	ADC12_IN4
通道 5	ADC12_IN5	ADC12_IN5
通道 6	ADC12_IN6	ADC12_IN6
通道 7	ADC12_IN7	ADC12_IN7
通道 8	ADC12_IN8	ADC12_IN8
通道 9	ADC12_IN9	ADC12_IN9
通道 10	ADC12_IN10	ADC12_IN10
通道 11	ADC12_IN11	ADC12_IN11
通道 12	ADC1_IN12	ADC2_IN12
通道 13	ADC1_IN13	ADC2_IN13
通道 14	ADC1_IN14	ADC2_IN14
通道 15	ADC1_IN15	ADC2_IN15

转换输入通道	ADC1 通道分配	ADC2 通道分配
通道 16	ADC1_IN16	ADC2_IN16
通道 17	ADC1_IN17	内置 T-Sensor
通道 18	ADC1_IN18	内置 V-Sensor

3.6 系统模块硬件互联

表 3-11 TIM 之间的互联

从模块	内部触发源 ITR0	内部触发源 ITR1	内部触发源 ITR2	内部触发源 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

注：TIMx 从模式控制寄存器位 TIMx_SMCR:TS 控制内部触发源选择，x = 1, 2, 3, 4, 5, 8。

表 3-12 比较器控制 TIM

比较器输出源	连接触发目标	比较器输出方向控制选择
COMPx_OUT	TIM1 刹车输入	输出方向 2
	TIM8 刹车输入	输出方向 4
	TIM1 Ocrefclear 输入	输出方向 6
	TIM1 输入捕捉 1	输出方向 7
	TIM2 输入捕捉 4	输出方向 8
	TIM2 Ocrefclear 输入	输出方向 9
	TIM3 输入捕捉 1	输出方向 10
	TIM3 Ocrefclear 输入	输出方向 11
	LPTIM 的触发输入	输出方向 15

注：COMPx 控制状态寄存器位 COMPx_CSR:OUT_SEL 控制 COMPx 输出方向选择，x = 1, 2, 3。

表 3-13 刹车控制

源	连接触发目标	说明
IO	TIM1 刹车输入 TIM8 刹车输入	
CSS		
COMP		

表 3-14 时钟作为 TIM 输入信号

源	连接触发目标	说明
LSI	TIM5	
LSE		

3.7 启动配置

用户通过设置 BOOT0 和 BOOT1 引脚值来选择三种启动模式

表 3-15 启动模式

启动模式	启动模式引脚选择		启动地址
	BOOT1	BOOT0	
地址选择 (主闪存存储器/内置 SRAM/QSPI)	x	0	BOOT_ADDR 默认值为主闪存存储器地址 0x0800 0000
系统存储器	0	1	0x1FFF E800
内置 ITCM 存储器	1	1	0x0000 0000

用户通过设置 BOOT0 引脚值和 BOOT1 引脚值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 BOOT1 和 BOOT0 的值来重新确定启动模式。

器件复位后，CPU 先从启动地址开始获取栈顶值，再从启动地址+0x4 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：地址选择，系统存储器和内置 ITCM 存储器。

1. 地址选择启动模式 (BOOT0=0):

MM32F5270 和 MM32F5280 在 Option bytes (0x1FFF_F838~ 0x1FFF_F83E) 中存放一个 32 位地址，用于系统 BOOT。上电复位后该地址会加载到对应的寄存器 BOOT_ADDR[31:0]。BOOT0=0 时，系统选择从 BOOT_ADDR 的地址 BOOT。BOOT_ADDR 地址需要按 2Kbyte 边界对齐。

2. 系统存储器启动模式 (BOOT1= 0 并且 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，系统从 0x1FFF E800 地址启动。

3. 内置 ITCM 存储器启动模式 (BOOT1= 1 并且 BOOT0 = 1):

内置 ITCM 的起始地址是 0x0000 0000，当其被选为启动模式时，系统从 0x0000 0000 地址启动。

3.8 引导程序

出厂后引导程序存放在系统存储器中，可以通过串口 UART1 进行 ISP 编程。MM32F5270 和 MM32F5280 支持通过配置 PA9/PA10 或 PB6/PB7 为 UART1 TX/RX 引脚来进行 ISP 编程引导。

4 Embedded FLASH 嵌入式闪存

4.1 简介

嵌入式闪存支持高达 256K 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），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。

4.2 闪存构成与说明

4.2.1 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 256 页（每页 1K 字节）或 64 个写保护块（每块 4K 字节）划分。
- 主闪存块可按页（每 1K 字节）擦除（Page Erase）。
- 以 4 页（4K 字节）为单位作为 1 个写保护块来设置写保护。
- 整个片内 Flash 由两部分组成：一部分是主存储块，另一部分是信息存储块。
- 主存储块用于存储用户代码和数据，用户代码可以对主存储器进行擦除、编程和读取操作。每个 1K 字节在主存储块中称为一页，可以执行最小单位的擦除；另外以 1 个写保护区为单位（4K 字节，4 页=1 个写保护块）进行写保护分配，如表 4-1 所示。

表 4-1 Flash 存储空间

模块	区块名称	页名称	地址	大小（字节）
主存储块	写保护区 0	页 0	0x0800 0000 - 0x0800 03FF	1K
		页 1	0x0800 0400 - 0x0800 07FF	1K
		页 2	0x0800 0800 - 0x0800 0BFF	1K
		页 3	0x0800 0C00 - 0x0800 0FFF	1K
	...	...	...	...
	写保护区 7	页 28	0x0800 7000 - 0x0800 73FF	1K
		页 29	0x0800 7400 - 0x0800 77FF	1K
		页 30	0x0800 7800 - 0x0800 7BFF	1K
		页 31	0x0800 7C00 - 0x0800 7FFF	1K
	写保护区 8	页 32	0x0800 8000 - 0x0800 83FF	1K
	...	...	...	...
	写保护区 63	页 252	0x0803 F000 - 0x0803 F3FF	1K
		页 253	0x0803 F400 - 0x0803 F7FF	1K
		页 254	0x0803 F800 - 0x0803 FBFF	1K
		页 255	0x0803 FC00 - 0x0803 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。信息存储器可分为系统存储器 ISP 和选项字节二部分，系统存储 ISP 区用于存储出厂的 ISP Bootloader，选项字节（Option byte）部分中的前 12 个字节是主存储器的写和读保护信息，剩余字节可用于存放用户特殊的数据。对于选项字节部分，用户可以通过规定的流程对其擦除、编程和读取。仅 ISP 部分由于用于固化 ISP 升级的代码，不支持用户进行擦除和编程。

表 4-2 信息块

模块	名称	地址	大小（字节）
信息块	系统存储 ISP	0x1FFF E800 - 0x1FFF F7FF	4K
	选项字节	0x1FFF F800 - 0x1FFF F9FF	0.5K

4.2.2 选项字节说明

在选项字节页中，内容主要有写保护使能，硬件看门狗使能等。Flash 控制器可以通过选项字节中值的设置，达到使能主存储器禁止写入功能，以避免非法写入；还可以使能硬件看门狗。相关信息存储在选项字节中，修改选项字节中内容后，需要复位或重新上电后才生效，写入时需按半字高低字节反码方式写入，如 nUser，nData 等。每次系统复位后，选项字节会重新装载选项字节信息块的数据，并做相应的判断与状态改变，这些状态保存在选项字节寄存器（FLASH_OBR 及 FLASH_WRP1R）中。在信息块中每个选择位都有对应的反码位，在加载选择位时反码位用于验证选择位是否正确，如果在加载过程中发现有差别，将产生一个选项字节错误标志（OPTERR），如果开启了对应的中断，将触发中断。

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

表 4-3 选项字节组织结构

地址	[15: 8]	[7: 0]	默认值
0x1FFF F800	nRDP	RDP	0x5AA5
0x1FFF F802	nUSER	USER	0xFFFF
0x1FFF F804	nData0	Data0	0xFFFF
0x1FFF F806	nData1	Data1	0xFFFF
0x1FFF F808	nWRP0	WRP0	0xFFFF
0x1FFF F80A	nWRP1	WRP1	0xFFFF
0x1FFF F80C	nWRP2	WRP2	0xFFFF
0x1FFF F80E	nWRP3	WRP3	0xFFFF
0x1FFF F810	nWRP4	WRP4	0xFFFF
0x1FFF F812	nWRP5	WRP5	0xFFFF
0x1FFF F814	nWRP6	WRP6	0xFFFF
0x1FFF F816	nWRP7	WRP7	0xFFFF
0x1FFF F838	nBoot_addr [7:0]	Boot_addr[7:0]	0xFF00
0x1FFF F83A	nBoot_addr [15:8]	Boot_addr[15:8]	0xFF00
0x1FFF F83C	nBoot_addr [23:16]	Boot_addr[23:16]	0xFF00
0x1FFF F83E	nBoot_addr [31:24]	Boot_addr[31:24]	0xF708

表 4-4 USER 的位含义

	Bit	Field	Type	Default	Description	FLASH_OBR
RDP	7: 0	RDP	rw	0xA5	0xA5	FLASH_OBR.
nRDP	15: 8	nRDP	rw	0x5A	0x5A	Bit1
User Byte	0	WDG_SW	rw	0x01	0: 硬件看门狗 1: 软件看门狗	FLASH_OBR. Bit2
	1	nRST_STOP	rw	0x01	0: 当进入停机（STOP）模式时产生复位 1: 进入停机（STOP）模式时不产生复位	FLASH_OBR. Bit3
	2	nRST_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
	5	Reserved	rw	0x01	保留为 0x01	保留
	6	Reserved	rw	0x01	保留为 0x01	保留
	7	Reserved	rw	0x01	保留为 0x01	保留
DATA0 Byte	0	DATA0.Bit0	rw	0x01	用户自定义	FLASH_OBR. Bit10
	1	DATA0.Bit1	rw	0x01	用户自定义	FLASH_OBR. Bit11
	2	DATA0.Bit2	rw	0x01	用户自定义	FLASH_OBR. Bit12
	3	DATA0.Bit3	rw	0x01	用户自定义	FLASH_OBR. Bit13
	4	DATA0.Bit4	rw	0x01	用户自定义	FLASH_OBR. Bit14
	5	DATA0.Bit5	rw	0x01	用户自定义	FLASH_OBR. Bit15
	6	DATA0.Bit6	rw	0x01	用户自定义	FLASH_OBR. Bit16
	7	DATA0.Bit7	rw	0x01	用户自定义	FLASH_OBR. Bit17
DATA1 Byte	0	DATA1.Bit0	rw	0x01	用户自定义	FLASH_OBR. Bit18
	1	DATA1.Bit1	rw	0x01	用户自定义	FLASH_OBR. Bit19
	2	DATA1.Bit2	rw	0x01	用户自定义	FLASH_OBR. Bit20
	3	DATA1.Bit3	rw	0x01	用户自定义	FLASH_OBR. Bit21
	4	DATA1.Bit4	rw	0x01	用户自定义	FLASH_OBR. Bit22
	5	DATA1.Bit5	rw	0x01	用户自定义	FLASH_OBR. Bit23

	Bit	Field	Type	Default	Description	FLASH_OBR
	6	DATA1.Bit6	rw	0x01	用户自定义	FLASH_OBR. Bit24
	7	DATA1.Bit7	rw	0x01	用户自定义	FLASH_OBR. Bit25

注意：在写保护值中，一个比特位对应四页，即 4096 Bytes。

4.3 闪存操作与流程

4.3.1 闪存读操作

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

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

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

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

表 4-5 Latency 设置关系

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

4.3.2 闪存编程方式与操作流程

嵌入式闪存支持如下三种编程方式。

表 4-6 编程方式

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

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

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

闪存的编程操作由一系列的动作组合而成，主要包括：

- 对 Flash 操作的解锁与保护
- 对 Flash 擦除（页擦除与整片擦除）
- 对 Flash 编程（半字编程）
- 对信息块中各空间（如选项字节）操作的解锁与保护
- 对信息块中各空间（如选项字节）擦除
- 对信息块中各空间（如选项字节）编程（半字编程）

ISP、IAP 方式编程流程

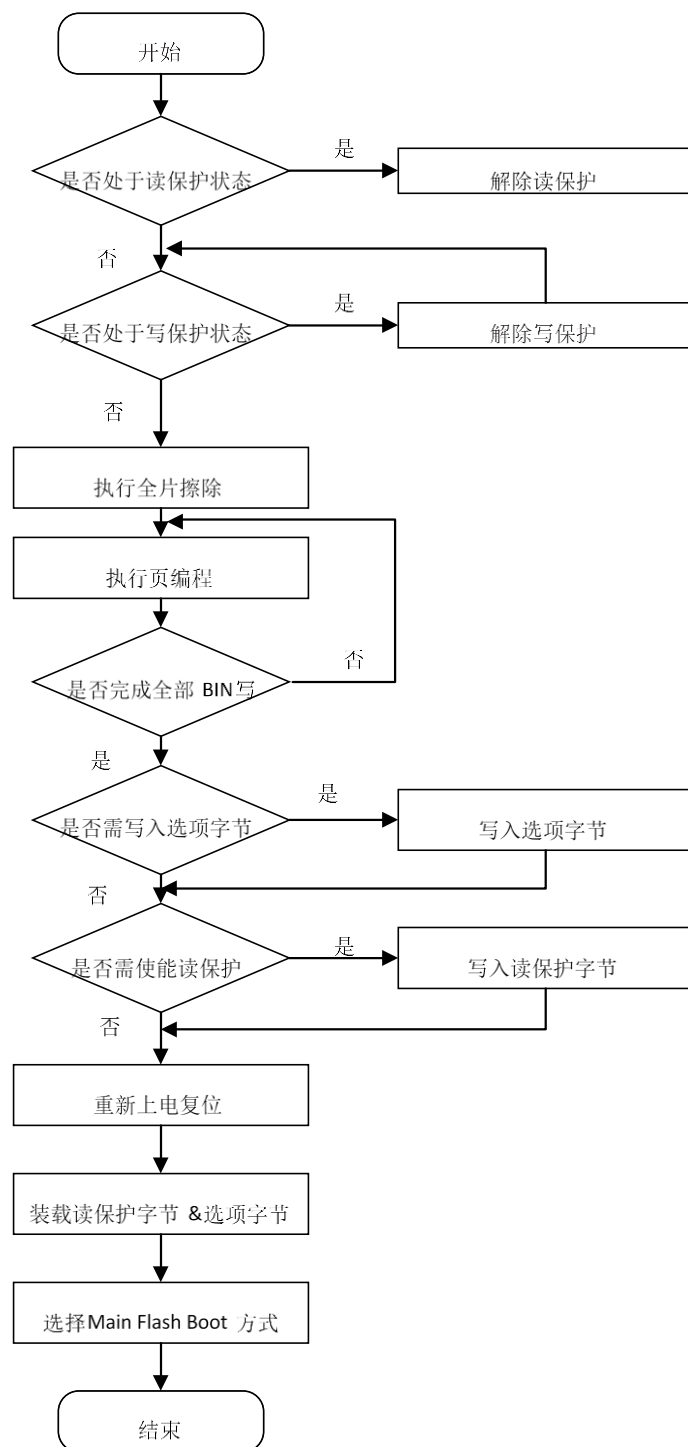


图 4-1 ISP 方式编程流程图

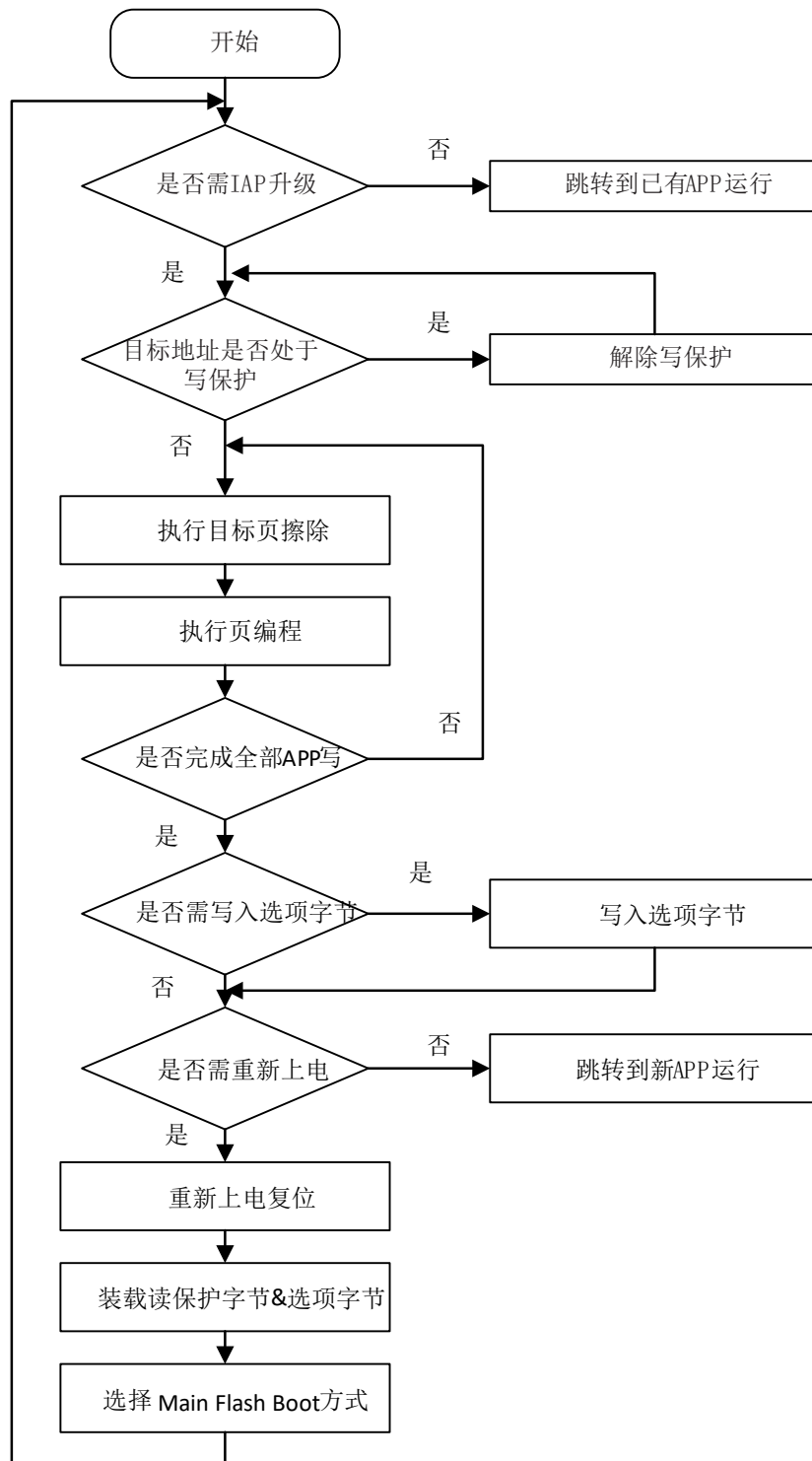


图 4-2 IAP 编程流程

4.3.3 对闪存块操作限制的解除与使能

嵌入式闪存在复位后是处于受保护状态的，可避免意外的页擦除、全片擦除和写值等破坏 Flash 存储空间的操作。复位后，FLASH_CR 寄存器进入锁定状态，FLASH_CR 的 LOCK 位被控制器模块置为 1。只有通过先后向 FLASH_KEYR 寄存器写入 0x45670123 和 0xCDEF89AB 做解锁操作后，FLASH_CR 的 LOCK 位置为 0，才能开启对 FLASH_CR 的访问权限，否则 FLASH_CR 寄存器不允许

被改写。

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

解除保护操作代码：

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

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

使能保护操作代码：

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

4.3.4 对选项字节区块操作限制的解除与使能

闪存控制器在复位后，它的选项字节区块默认是处于写保护的，并且任何时候都是可读的。同样是为了避免对选项字节区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 4-7 保护设置的状态变化

设置与状态	主闪存块	信息块	说明
上电复位 闪存控制器状态为 FLASH_CR.LOCK=1 FLASH_CR.OPTWRE=0	保护	保护	使能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



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

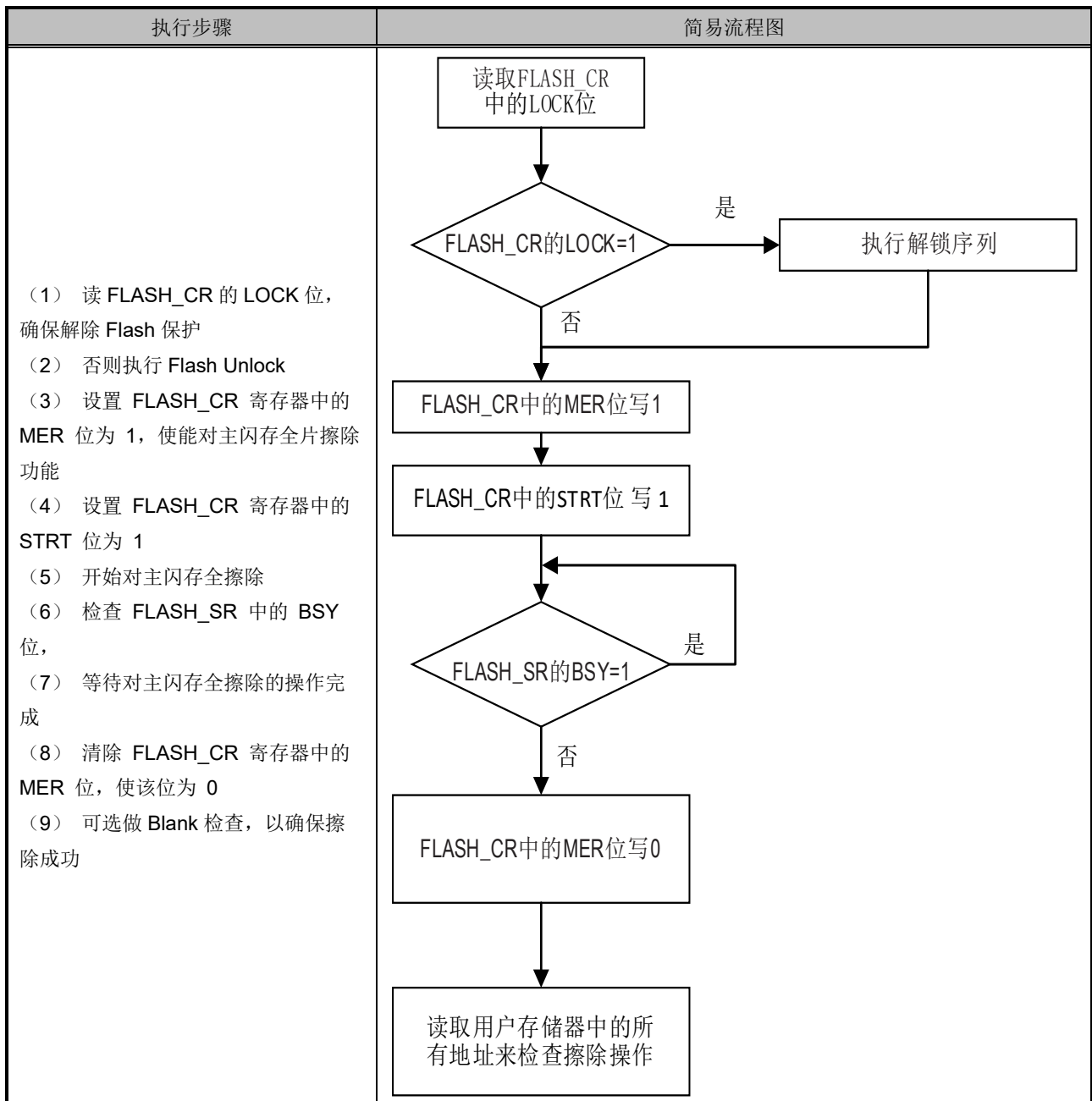
```
#define FLASH_CR_OPTWRE_Pos    (9)  
#define FLASH_CR_OPTWRE      (0x01U << FLASH_CR_OPTWRE_Pos)  
void FLASH_OPT_Lock (void)  
{  
    FLASH->CR &= ~FLASH_CR_OPTWRE;  
}
```

4.3.5 主闪存块擦除

闪存控制器支持整片擦除主闪存块和以页为单位擦除主闪存中的页。

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

整片擦除操作的寄存器设置，具体步骤如下：



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
(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) 把待擦除页基地址写入 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	<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 --> G; 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>

4.3.6 主闪存块编程

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

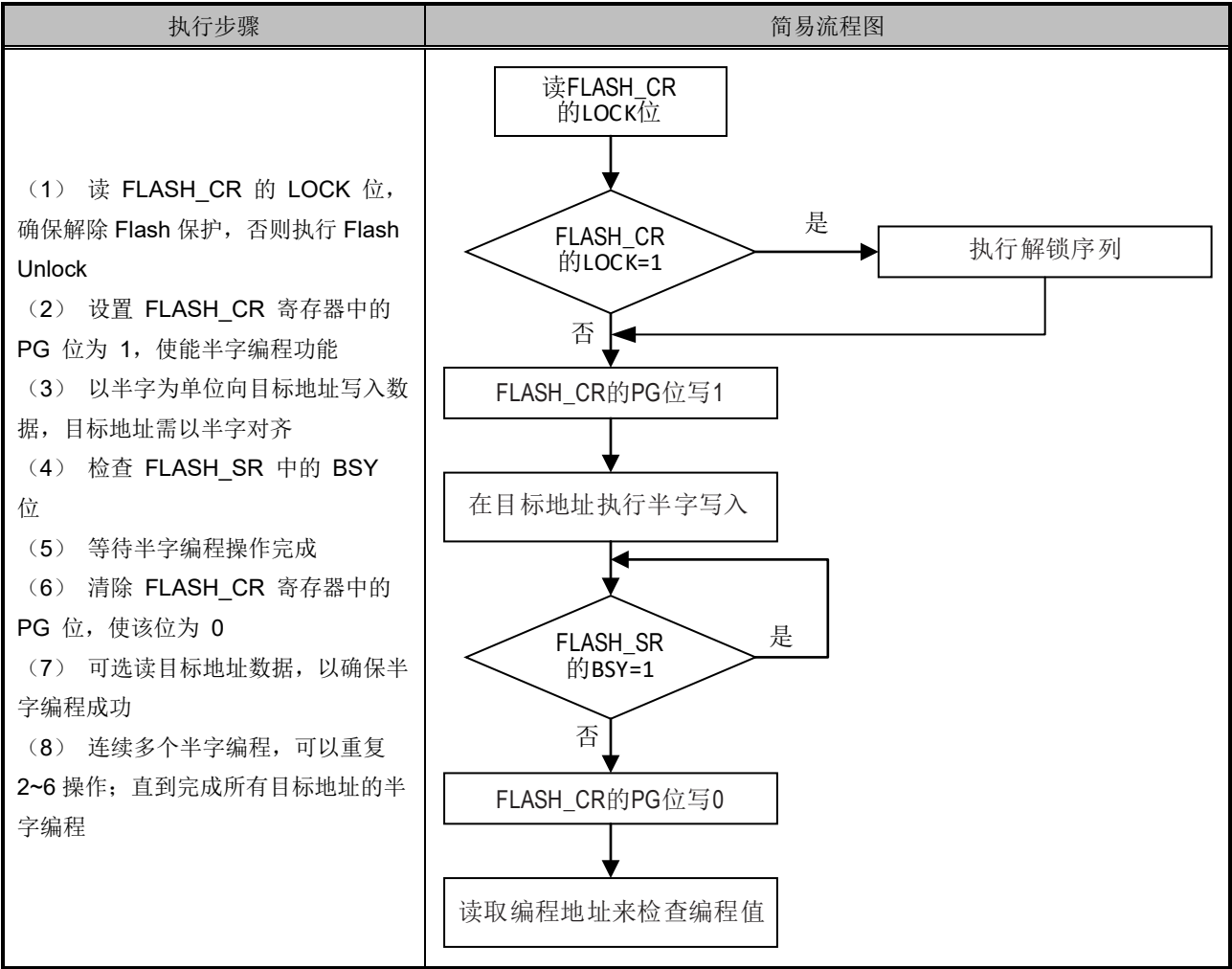
主闪存控制器会预读待编程半字是否为全 1（即是否为 0xFFFF），如果不是，这次编程操作会自动取消，并且在 FLASH_SR 寄存器的 PGERR 位上提示编程错误警告。

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

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

主闪存块编程操作的寄存器设置，具体步骤如下：





4.3.7 选项字节区块擦除

选项字节区块擦除操作的寄存器设置，具体步骤如下：



执行步骤	简易流程图
(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护，否则执行 Flash Unlock (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 (3) 把待擦除区块基地址写入 FLASH_AR 寄存器 (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 (5) 设置 FLASH_CR 寄存器中的 STRT 位为 1 (6) 检查 FLASH_SR 中的 BSY 位 (7) 等待选项字节块擦除操作完成 (8) 清除 FLASH_CR 寄存器中的 OPTER 位，设置值为 0 (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 (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	<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} C --> D D -- 是 --> E[执行解锁OPT序列] D -- 否 --> F[将闪存选项字节块基地址写入FLASH_AR] E --> F F --> G[FLASH_CR中的OPTER位写1] G --> H[FLASH_CR的STRT位写1] H --> I{FLASH_SR的BSY=1} I -- 是 --> I I -- 否 --> J[FLASH_CR中的OPTER位写0] J --> K[通过读取闪存选项存储器中的所有地址来检查闪存选项擦除操作]</pre>

4.3.8 选项字节区块编程

选项字节区块的编程与主闪存块地址的编程不同，因其写入值复位后加载到配置选项，需要更加严格的保护。解除对闪存控制器的访问限制后，还需要对选项字节区块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
(1) 读 FLASH_CR 的 LOCK 位，确保解除 Flash 保护，否则执行 Flash Unlock (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 (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 (4) 写数据（半字）到目标地址，目标地址需以半字对齐 (5) 检查 FLASH_SR 中的 BSY 位 (6) 等待半字编程操作完成 (7) 清除 FLASH_CR 寄存器中的 OPTPG 位，使该位值为 0 (8) 可选读目标地址数据，以确保半字编程成功 (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	<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} C --> D D -- 是 --> E[执行解锁OPT序列] D -- 否 --> F[FLASH_CR的OPTPG位写1] E --> F 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>

4.3.9 闪存保护

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

4.3.9.1 主闪存块读保护

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

使能读保护

按选项字节区块半字编程的操作方式，按顺序写 RDP 半字到对应地址

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



当 RDP 字包含下列数值时，且被重新上电复位后主闪存块被置于保护状态。

表 4-8 Flash 读保护状态

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

当读保护半字被写入相应的值以后：

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

解除读保护

从内置 SRAM 或 ICP 方式解除读保护的过程是：

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

表 4-9 Flash 解除读保护状态

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

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

4.3.9.2 主闪存块写保护

使能写保护

写保护通过设置选项字节区块中的 WRP0~WRP7 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 4-10 写保护区域

地址	[15: 8]	[7: 0]	默认值	注释
0x1FFF F808	nWRP0	WRP0	0xFFFF	
0x1FFF F80A	nWRP1	WRP1	0xFFFF	
0x1FFF F80C	nWRP2	WRP2	0xFFFF	
0x1FFF F80E	nWRP3	WRP3	0xFFFF	
0x1FFF F810	nWRP4	WRP4	0xFFFF	
0x1FFF F812	nWRP5	WRP5	0xFFFF	
0x1FFF F814	nWRP6	WRP6	0xFFFF	
0x1FFF F816	nWRP7	WRP7	0xFFFF	

解除写保护

解除写保护有下述 2 种情形：

1. 情形 1：解除写保护，同时解除读保护：

a. 使用闪存控制寄存器（FLASH_CR）的 OPTER 位擦除整个选项字节区块；写 0x5AA5 半字到对应地址 0x1FFFF800；

b. 对 0x08000000 的主 Flash 全片擦除；

c. 进行系统复位，重装载选项字节（包含新的 WRP 字节），写保护被解除。

使用这种方法，将解除全片主闪存模块的写保护同时擦除全片主闪存块。

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

a. 使用闪存控制寄存器（FLASH_CR）的 OPTER 位擦除整个选项字节区块；

b. 进行系统复位，重装载选项字节（包含新的 WRP 字节），写保护被解除。

使用这种方法，将解除整个主闪存模块的写保护，同时保持读保护有效。

4.4 寄存器

4.4.1 寄存器总览

表 4-11 FLASH 寄存器概览

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

Offset	Acronym	Register Name	Reset
0x20	FLASH_WRPR	写保护寄存器	0xFFFFFFFF

注意：Flash 寄存器只支持以 32 位的方式访问

4.4.2 FLASH_ACR 闪存访问控制寄存器

偏移地址：0x00

复位值：0x0000 0038

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

Bit	Field	Description
31: 6	Res.	保留，必须保持复位值
5	PRFTBS	预取缓冲区状态（Prefetchbufferstatus） 0：预取缓冲区关闭 1：预取缓冲区开启
4	PRFTBE	预取缓冲区使能（Prefetch buffer enable） 0：关闭预取缓冲区 1：启用预取缓冲区 注释 1：只有当 LATENCY 为 0 时，通过此位才能控制预取缓冲区开关； 注释 2：当 RCC_CFGR 中 HPRE[3]位设置为非 0 值时，预取缓冲区硬件自动使能，PRFTBS 位置 1。
3	Res.	保留，必须保持复位值
2: 0	LATENCY	时延（Latency） 这些位表示 SYSCLK（系统时钟）周期与闪存访问时间的比例。 000：零等待状态，当 $0 < \text{SYSCLK} \leq 24\text{MHz}$ 001：一个等待状态，当 $24\text{MHz} < \text{SYSCLK} \leq 48\text{MHz}$ 010：两个等待状态，当 $48\text{MHz} < \text{SYSCLK} \leq 72\text{MHz}$ 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 $96\text{MHz} < \text{SYSCLK} \leq 120\text{MHz}$

4.4.3 FLASH_KEYR FPEC 键寄存器

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	19	18	17	16	15
FKEYR															
w															

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

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

注：所有这些位是只写的，读出时返回 0。

4.4.4 FLASH_OPTKEYR 闪存 OPTKEY 寄存器

偏移地址：0x08

复位值：0x0000 0000

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

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

注：所有这些位是只写的，读出时返回 0。

4.4.5 FLASH_SR 闪存状态寄存器

偏移地址：0x0C

复位值：0x0000 0000

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

Bit	Field	Description
31:6	Res.	保留，必须保持复位值
5	EOP	操作结束 (End of operation) 当闪存操作 (编程 擦除) 完成时，硬件设置这位为“1”，写入“1”可以清除这位状态。
4	WRPRERR	写保护错误 (Write protection error) 试图对写保护的闪存地址编程时，硬件设置这位为“1”，写入“1”可以清除这位状态。

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

4.4.6 FLASH_CR 闪存控制寄存器

偏移地址：0x10

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
						rw		rw	rw	rw	rw		rw	rw	rw

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

Bit	Field	Description
1	PER	页擦除 (Page erase) 选择擦除页。
0	PG	编程 (Programming) 选择编程操作。

4.4.7 FLASH_AR 闪存地址寄存器

偏移地址: 0x14

复位值: 0x0000 0000

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

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

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

4.4.8 FLASH_OBR 选项字节寄存器

偏移地址: 0x1C

复位值: 0x03FF FC1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						Data1								Data0	
						r								r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						Res.					nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR
r											r	r	r	r	r

Bit	Field	Description
31: 26	Res.	保留, 必须保持复位值
25: 18	Data1	Data1
17: 10	Data0	Data0
9: 5	Res.	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

Bit	Field	Description
3	nRST_STOP	进入停机模式时的复位事件 0: 当进入停机（STOP）模式时产生复位 1: 进入停机（STOP）模式时不产生复位
2	WDG_SW	选择看门狗事件 0: 硬件看门狗 1: 软件看门狗
1	RDPRT	读保护（Read protection level status） 当设置为“1”，表示闪存存储器被读保护。 注：该位为只读。
0	OPTERR	选项字节错误（Option byte error） 当该位为“1”时表示选项字节和它的反码不匹配。 注意：该位为只读。

4.4.9 FLASH_WRPx 写保护寄存器

偏移地址：0x20, 0x24

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	写保护（Write protect） 该寄存器包含由 OBL 加载的写保护选项字节。 0: 写保护生效 1: 写保护失效

5 PWR 电源控制

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

5.1 供电系统

芯片由以下四种电源提供供电：

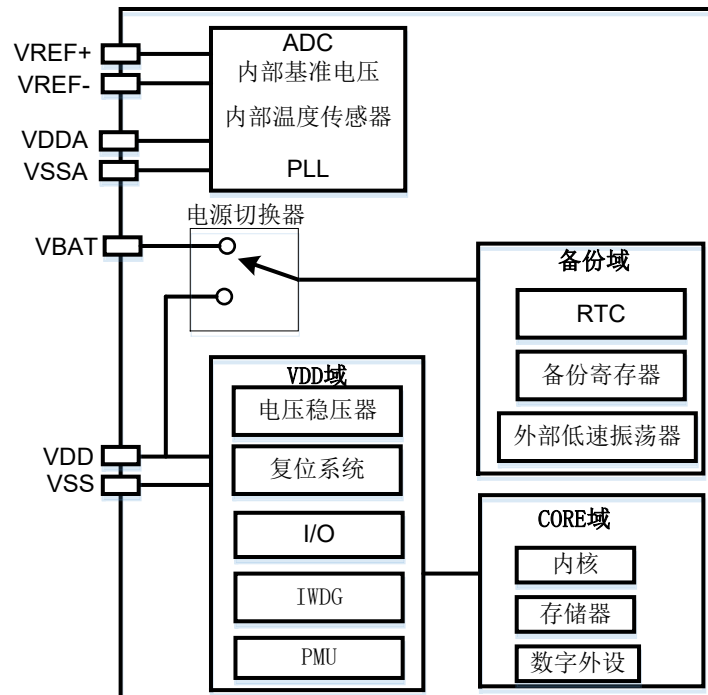


图 5-1 电源控制功能框图

- 由 VDDA 和 VSSA 提供的模拟电源，为芯片模拟模块提供电压，用于 ADC 模块、内部基准电压、内部温度传感器和 PLL 等。
- 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 的电容器，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。

5.1.1 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-

5.1.2 数字模块供电

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

5.1.3 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。

5.1.4 VDD 域

VDD 域主要给 PMU、IWDG 供电，在上电后保持工作状态。

5.1.5 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。

5.1.6 CORE 域

CORE 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 CORE 域，在唤醒后芯片会自动开启。主要有以下几种工作状态：

运行模式（Run Mode）：CORE 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（Low Power Run Mode）：CORE 域以低功耗模式运行，内存、外设都以低功耗工作。

睡眠模式（Sleep Mode）：CORE 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（Low Power Sleep Mode）：CORE 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（Stop Mode）：CORE 域以低功耗模式工作，只保持寄存器和 RAM 的内容。

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

5.2 电源管理器

5.2.1 POR 上电复位和 PDR 掉电复位

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

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

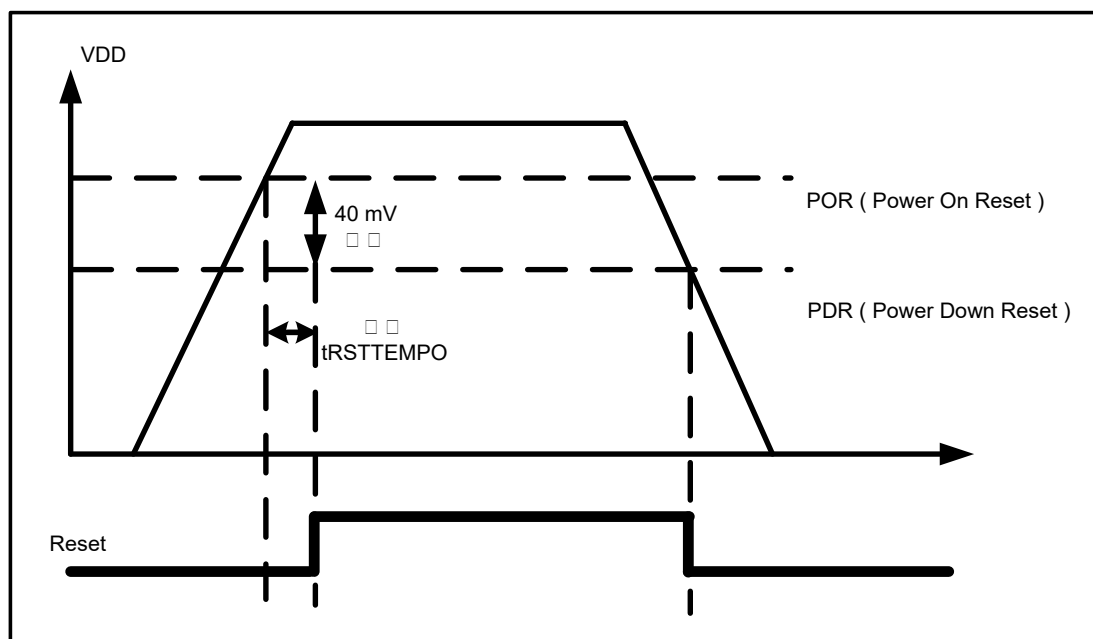


图 5-2 上电复位和掉电复位波形图

5.2.2 PVD 可编程电压监测器

可编程电压监测器 PVD（Programmable Voltage Detector）可以用来监视芯片的供电电压，在供电电压下降到给定的阈值以下时，产生一个中断，软件可以做紧急处理。当供电电压又恢复到给定的阈值以上

时，也会产生一个中断，软件处理供电恢复。供电下降的阈值与供电上升的阈值有一个固定的差值，这就是 PVD 迟滞电压，通过列出的 PVD 阈值数据可以看到这个差别。引入这个差值的目的是为了防止电压在阈值上下小幅抖动，而频繁地产生中断。

用户可以通过软件设置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。

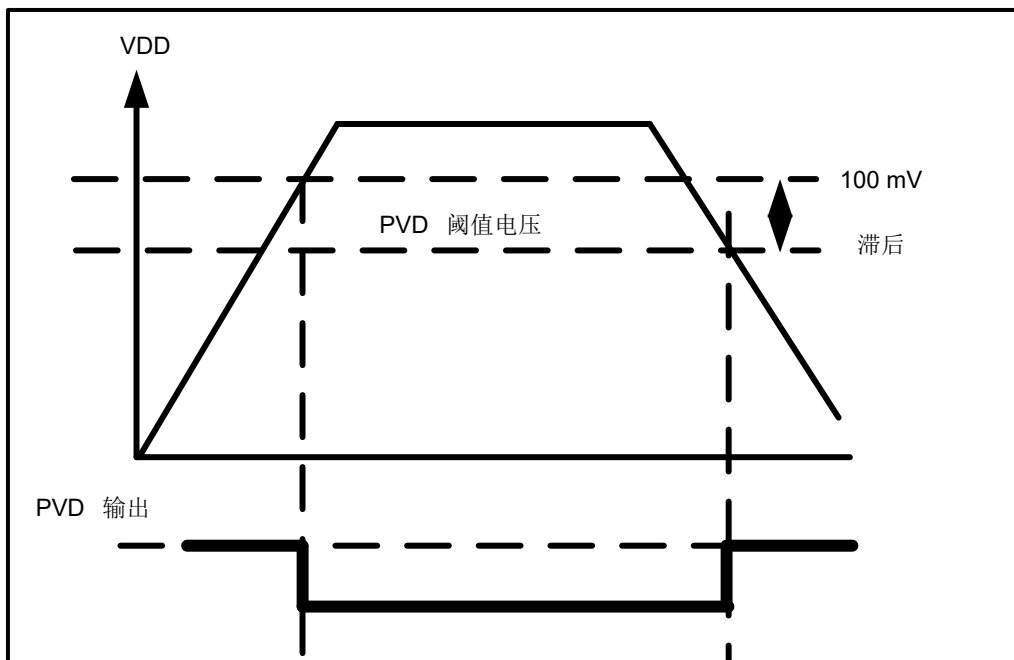


图 5-3 PVD 阈值波形图

5.3 功耗控制

5.3.1 功耗控制概述

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

芯片有六种低功耗模式，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：CORE 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：CORE 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：CORE 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。

5.3.1.1 功耗模式切换关系

各模式之间的切换关系为：

运行模式可以进入低功耗运行模式，并且从低功耗运行模式退出到运行模式。

运行模式可以进入睡眠模式，并且从睡眠模式退出到运行模式。

运行模式可以进入停机模式，并且从停机模式退出到运行模式。

运行模式可以进入深度停机模式，并且从深度停机模式退出到运行模式。

运行模式可以进入待机模式，并且从待机模式退出到运行模式。

低功耗运行模式可以进入低功耗睡眠模式，并且从低功耗睡眠模式退出到低功耗运行模式。

低功耗运行模式可以进入停机模式，并且从停机模式退出到低功耗运行模式。

低功耗运行模式可以进入深度停机模式，并且从深度停机模式退出到低功耗运行模式。

低功耗运行模式可以进入待机模式，并且从待机模式只能退出到运行模式。

模式	进入方式	唤醒方式	对 CORE 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
待机模式 (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 CORE 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
深度待机模式 (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

5.3.2 运行模式降低系统时钟

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

芯片的系统时钟可以灵活配置, 用户可以选择不同的时钟源作为系统时钟; 也可以通过配置不同时钟分频器来降低系统时钟 (SYSCLK、HCLK、PCLK1、PCLK2) 的频率。

进入睡眠模式前, 降低外设的时钟频率可以有效节省睡眠模式下的功耗。

5.3.3 外设时钟的控制

在芯片执行程序过程中, 可以通过关闭外设时钟来降低功耗。

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

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

5.3.4 Low Power Run Mode 低功耗运行模式

5.3.4.1 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。进入该模式前配置 RCC_CFGR.HPRE 或者 RCC_CR.HSIDIV，保证系统频率不超过 2MHz。

5.3.4.2 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 5-2 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	电压稳压器从低功耗模式唤醒的时间

5.3.5 Sleep Mode 睡眠模式

5.3.5.1 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。

5.3.5.2 睡眠模式退出

中断或事件发生后，睡眠模式立即被唤醒。

表 5-3 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 5-4 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

5.3.6 Low Power Sleep Mode 低功耗睡眠模式

5.3.6.1 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。

在低功耗睡眠模式下, 所有的 I/O 引脚都保持在运行模式时的状态。

5.3.6.2 低功耗睡眠模式退出

中断或事件发生后, 低功耗睡眠模式立即被唤醒。

表 5-5 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
退出	如果执行 WFI 进入低功耗睡眠模式: 中断 (参考中断向量表) 如果执行 WFE 进入低功耗睡眠模式: 唤醒事件 (参考唤醒事件管理)
唤醒延时	立即唤醒

表 5-6 Low Power SLEEPONEXIT 模式

Low Power SLEEPON- 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

5.3.7 Stop Mode 停机模式

停机模式下, CPU 进入深度睡眠模式, CORE 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。

5.3.7.1 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。

5.3.7.2 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。

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

表 5-7 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

5.3.8 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，CORE 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。

5.3.8.1 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。

5.3.8.2 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。

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

表 5-8 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 PDDS，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 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

深度待机模式	说明
注意事项	在进入深度待机模式时需将不使用的 GPIO 设置成模拟输入模式

5.3.9 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 CORE 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。

5.3.9.1 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。

5.3.9.2 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 5-9 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

5.3.9.3 滤波功能

配置 PWR_CR4（或 PWR_CR5）寄存器的 FILTSEL0（或 FILTSEL1）位来选择需要过滤的唤醒引脚；FILT CNT0（或 FILT CNT1）选择唤醒源过滤宽度；配置 PWR_CR3.WPn（n=0~5）的极性选择对高电平（或低电平）唤醒源进行滤波；最后配置 FILTE0（或 FILTE1）使能选择位，并且与 PWR_CR3.WPn 的极性匹配，打开唤醒源滤波功能开启对电平的滤波。将电平保持时间大于 $TL SI\_CLK * FILT CNT0$ 的唤醒源滤波通过并唤醒低功耗模式，将电平保持时间小于 $TL SI\_CLK * FILT CNT0$ 以下的唤醒源作为干扰电平滤除掉，以防止误操作唤醒。

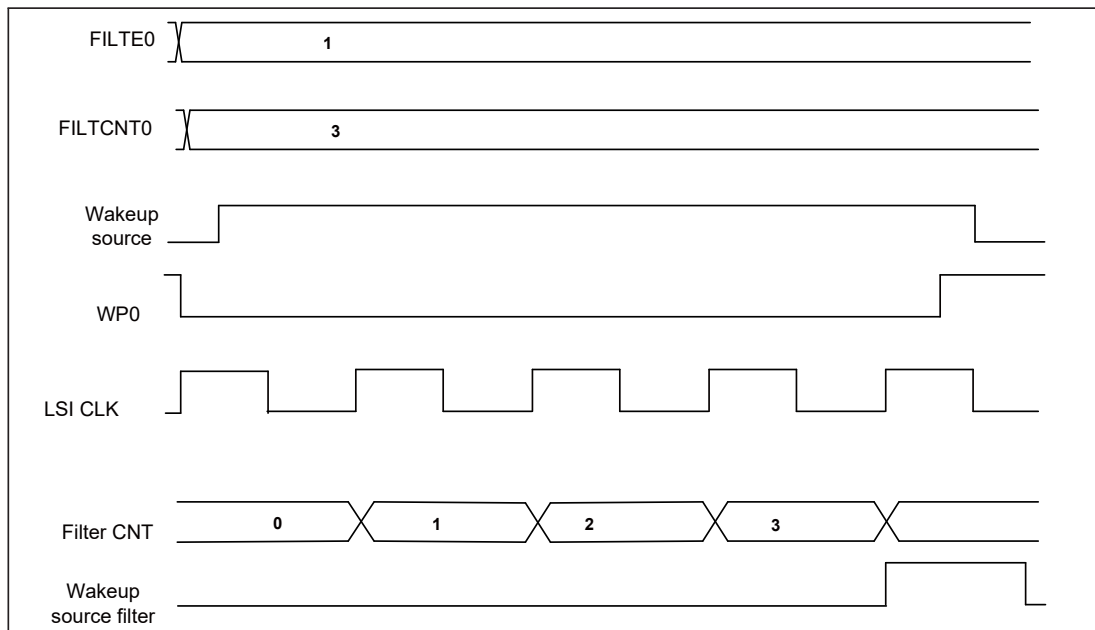


图 5-5 Standby 模式唤醒滤波功能波形图

5.3.9.4 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。

5.4 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。这是由于 CPU 内核失去了时钟，因此需要设置正确的唤醒源。

5.5 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 5-10 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
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

5.5.1 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										w1c		rw	rw

Bit	Field	Description
31:16	Reserved	保留，始终读为 0
15:14	VOS	电压稳压器的电压输出选择 00: 1.55V 01: 1.6V 10: 1.65V 11: 1.7V 注：如果系统时钟需要达到最大频率 120MHZ,需要将 VOS 设置为 2'b11,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
12:4	Reserved	保留，始终读为 0

Bit	Field	Description
3	CSBF	清除待机标志位 (Clear Standby Flag) 始终读为 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
0	LPDS	深睡眠下的低功耗 (Low Power Deepstop) PDDS = 0 时, 与 PDDS 位协同操作 1: 进入停机模式时, 电压稳压器处于低功耗模式。 0: 进入停机模式时, 电压稳压器处于正常功耗模式。 当进入停机模式时, LPDS = 1 时的电流小于 LPDS = 0 时的电流。详见该芯片对应的数据手册。

5.5.2 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	VOSRDY	Res												SBF	Res
	r													r	

Bit	Field	Description
31:15	Reserved	保留, 始终读为 0
14	VOSRDY	电压稳压器的电压输出选择准备完成标志位 1: 准备未完成 0: 准备完成
13:2	Reserved	保留, 始终读为 0
1	SBF	待机标志 (Standby Flag) 该位由硬件设置, 并只能由 POR/PDR (上电/掉电复位) 或设置电源控制寄存器 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
0	Reserved	保留, 始终读为 0

5.5.3 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 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

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5:0	EWUP5~ EWUP0	使能唤醒引脚 0~5 (Enable WakeUp Pin0~5) 1 : WKUP 引脚用于唤醒待机模式 0 : WKUP 引脚上的事件不能唤醒待机模式, 为通用 I/O。

5.5.4 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 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

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: 下降沿 (使能滤波则检测低电平), 唤醒引脚必须先配置为高电平且输入上拉 0: 上升沿 (使能滤波则检测高电平), 唤醒引脚必须先配置为低电平且输入下拉

5.5.5 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 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	
rw											w1c	rw		rw	

Bit	Field	Description
31:16	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	FILTF0	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 对高电平滤波 10：使能滤波器 0 对低电平滤波 其他：保留 注意：与 PWR_CR3 的电平极性配置保持一致
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

5.5.6 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 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
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 对高电平滤波 10：使能滤波器 1 对低电平滤波 其他：保留 与 PWR_CR3 的电平极性配置保持一致
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

5.5.7 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，也不会再在 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
31:3	Reserved	保留，始终读为 0

Bit	Field	Description
2:0	STDBY_FS_WK	STDBY_FS_WK: 快速唤醒 Standby 时间选择位 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 其他: 保留, 禁止使用。 使用待机模式前需要配置该寄存器为非“0”值

5.5.8 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 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

5.5.9 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

5.5.10 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						LSICAL					LSICALSEL				
						rw					w				

Bit	Field	Description
31:10	Reserved	保留，始终读为 0
9:5	LSICAL	内部低速时钟校准位（Internal Low-speed Clock Calibration Bits） 在系统启动时，这些位被自动初始化为出厂校准值，用户可以写入其他校准值，但读出始终为出厂校准值。如果 LSICALSEL=0x1F，写入的值可以重新校正 LSI 频率，否则写入的值不起作用
4:0	LSICALSEL	内部低速时钟校准值选择方式（Internal Low-speed Calibration Select） 初值为 0，当写入不同值时： 写入 1F：选择寄存器 LSICAL 的值，写入 1F 后仍读出为 0。 写入其他：选择出厂校准值

6 RCC 时钟和复位

6.1 复位单元

6.1.1 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。

6.1.2 功能框图

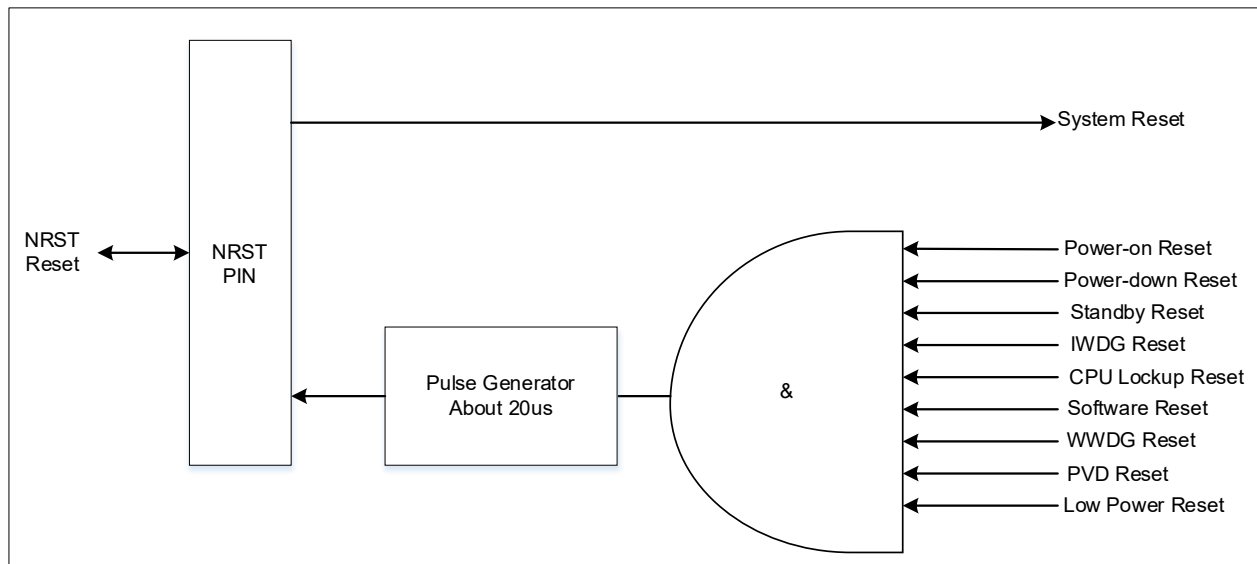


图 6-1 复位功能框图

6.1.3 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）

6.1.4 功能描述

6.1.4.1 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)

6.1.4.2 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

6.1.4.3 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

6.2 时钟单元

6.2.1 简介

六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 1 (PLL1)
- 锁相环 2 (PLL2)
- 外部低速时钟 (LSE)
- 内部低速时钟 (LSI)

6.2.2 功能框图

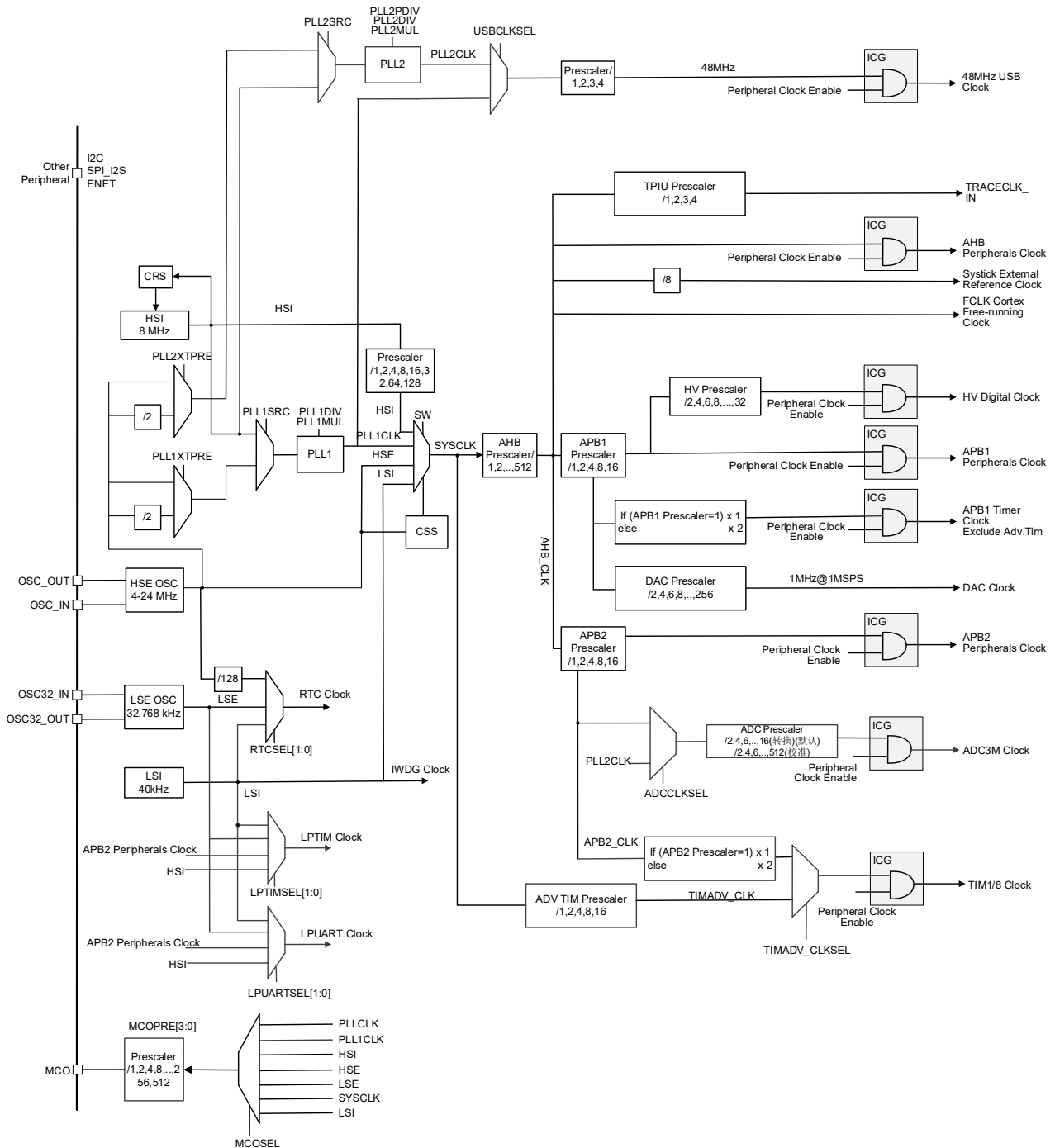


图 6-2 时钟树

6.2.3 主要特征

通过时钟配置寄存器（RCC_CFGR）的预分频控制位来分别配置 AHB、APB1 和 APB2 总线的时钟频率。AHB 和 APB1，APB2 总线时钟的最大频率是 120MHz。

6.2.4 功能描述

6.2.4.1 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。

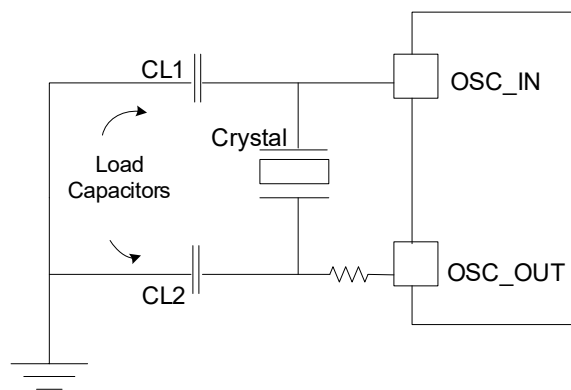


图 6-3 高速晶振/陶瓷谐振器

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。

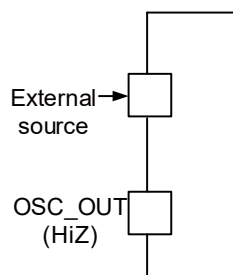


图 6-4 外部高速输入时钟

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。

6.2.4.2 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 PLL1 和 PLL2 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。

6.2.4.3 锁相环 1 (PLL1)

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

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

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

$$\frac{f_{PLL1\_IN}}{(PLL1DIV[2:0]+1)} = \frac{f_{PLL1\_OUT}}{PLL1MUL[6:0]+1}$$

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

注意事项:

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

使能 PLL1 配置步骤: (选择 HSI 为 PLL1 时钟源)

- 配置时钟控制寄存器 (RCC_CR) 中的 HSION 位为 1, 使能 HSI;
- 等待时钟控制寄存器 (RCC_CR) 中的 HSIRDY 位被置位为 1, 表示 HSI 稳定, 可输出有效时钟, 此时才可被选择使用作为系统时钟或外设时钟源;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中 PLL1MUL (倍频系数) 和 PLL1DIV (分频系数) 控制位;
- 配置时钟控制寄存器 (RCC_CR) 中的 PLL1ON 位为 1, 使能 PLL1;
- 等待时钟控制寄存器 (RCC_CR) 中的 PLL1RDY 位被置位为 1, 表示 PLL1 稳定, 可输出有效时钟, 此时才可被选择使用作为系统时钟或外设时钟源。

使能 PLL1 配置步骤: (选择 HSE 为 PLL1 时钟源)

- 配置时钟控制寄存器 (RCC_CR) 中的 HSEON 位为 1, 使能 HSE;
- 等待时钟控制寄存器 (RCC_CR) 中的 HSERDY 位被置位为 1, 表示 HSE 稳定, 可输出有效时钟, 此时才可被选择使用作为系统时钟或外设时钟源;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中的 PLL1SRC 位为 1, 选择 HSE 时钟用作 PLL1 输入时钟源;
- 配置 PLL1 配置寄存器 (RCC_PLL1CFGR) 中的 PLL1XTPRE 位, 选择 HSE 或者 HSE 2

分频时钟为 PLL1 输入时钟源；

- 配置 PLL1 配置寄存器（RCC_PLL1CFGR）中 PLL1MUL（倍频系数）和 PLL1DIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLL1ON 位为 1，使能 PLL1；
- 等待时钟控制寄存器（RCC_CR）中的 PLL1RDY 位被置位为 1，表示 PLL1 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源。

6.2.4.4 锁相环 2（PLL2）

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

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

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

$$\frac{f_{PLL2\_IN}}{(PLL2DIV[2:0]+1)*(PLL2PDIV[2:0]+1)} = \frac{f_{PLL2\_OUT}}{PLL2MUL[7:0]+1}$$

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

注意事项：

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

使能 PLL2 配置步骤：（选择 HSI 为 PLL2 时钟源）

- 配置时钟控制寄存器（RCC_CR）中的 HSION 位为 1，使能 HSI；
- 等待时钟控制寄存器（RCC_CR）中的 HSIRDY 位被置位为 1，表示 HSI 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源；
- 配置 PLL2 配置寄存器（RCC_PLL2CFGR）中 PLL2MUL（倍频系数）、PLL2PDIV（PLL2_IN 分频系数）和 PLL2DIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLL2ON 位为 1，使能 PLL2；
- 等待时钟控制寄存器（RCC_CR）中的 PLL2RDY 位被置位为 1，表示 PLL2 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源。

使能 PLL2 配置步骤：（选择 HSE 为 PLL2 时钟源）

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

6.2.4.5 外部低速时钟（LSE）

外部低速时钟的时钟源有以下两种：

- 外部低速晶振/陶瓷谐振器输入时钟（32.768kHz 晶体振荡器产生）

其为实时时钟单元（RTC）或者其他定时功能提供低功耗且高精度的时钟源。

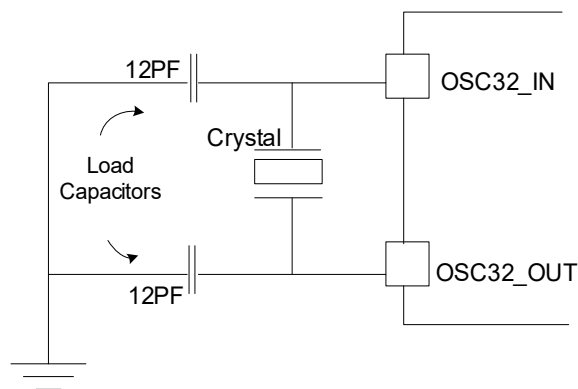


图 6-5 低速晶振/陶瓷谐振器

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。

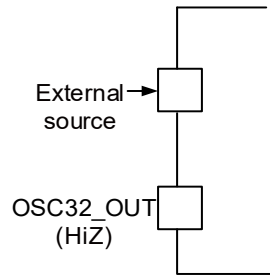


图 6-6 外部低速输入时钟

上图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 APB1 外设时钟使能寄存器（RCC_APB1ENR）的 BKP 位为 1，使能 BKP 时钟；
- 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。

6.2.4.6 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。

6.2.4.7 中断

表 6-1 RCC 全局中断表

中断事件	事件标志位	使能控制位	标志清除位
RCC_PLL1RDY	PLL1RDYF	PLL1RDYIE	PLL1RDYC
RCC_PLL2RDY	PLL2RDYF	PLL2RDYIE	PLL2RDYC
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。

6.2.4.8 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 1（PLL1）：当 HSI（或 HSE）作为 PLL1 时钟源时，HSI（或 HSE）不可禁止
- 内部低速时钟（LSI）

系统时钟配置步骤：

- 使能需要的系统时钟源（HSI，HSE，PLL1，LSI），每个时钟使能方式不同，具体方式请查看（HSI，HSE，PLL1，LSI 章节）；
- 等待被选择的时钟源 RDY 信号被置位为 1，表示系统时钟源稳定（当目标时钟源稳定后，系统时钟才可以切换）；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位来选择系统时钟；
- 通过读取时钟配置寄存器（RCC_CFGR）的 SWS 位，判断当前系统时钟的时钟源。

6.2.4.9 系统时钟频率切换

系统时钟频率从低速到高速切换，或者从高速到低速切换，推荐先切换到中速频率过度，高速低速切换间隔至少 1us。

6.2.4.10 外设复位

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

6.2.4.11 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 PLL1 输入时钟，PLL1 时钟用作系统时钟，则 PLL1 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 PLL1 配置寄存器（RCC_PLL1CFGR）中的 PLL1SRC 位为 1，选择 HSE 时钟用作 PLL1 输入时钟；
- 配置 PLL1 配置寄存器（RCC_PLL1CFGR）中的 PLL1XTPRE 位，选择 HSE 或者 HSE2 分频为 PLL1 输入时钟；
- 配置 PLL1 配置寄存器（RCC_PLL1CFGR）中 PLL1MUL（倍频系数）和 PLL1DIV（分频系数）控制位；
- 配置时钟控制寄存器（RCC_CR）中的 PLL1ON 位为 1，使能 PLL1；
- 等待时钟控制寄存器（RCC_CR）中的 PLL1RDY 位被置位为 1，表示 PLL1 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位为 2'b10 来选择 PLL1 为系统时钟；
- 通过读取时钟配置寄存器（RCC_CFGR）的 SWS 位，判断当前系统时钟的时钟源是否为 PLL1；
- 配置控制状态寄存器（RCC_CSR）的 LSION 位为 1 和 LSIOE 位为 1，使能 LSI；
- 等待控制状态寄存器（RCC_CSR）中的 LSIRDY 位被置位为 1，表示 LSI 稳定，可输出有效时钟；
- 配置时钟控制寄存器（RCC_CR）中的 CSSON 位为 1，时钟安全系统使能。

6.2.4.12 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 7 个时钟信号中的一个作为 MCO 输出时钟：

表 6-2 MCO 与时钟源对应关系

时钟配置寄存器（RCC_CFGR）的 MCO 位	时钟源
000x	没有时钟输出
0010	LSI
0011	LSE
0100	SYSCLK
0101	HSI
0110	HSE
0111	PLL1
1000	PLL2

6.2.4.13 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。

6.2.4.14 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。

注意事项：

- 当高级定时器选择 TIMADV_CLK 时钟时，需确保 TIMADV_CLK=AHB_CLK x2（即 AHB 为 2 分频时，TIMADV_CLK 需设为不分频；AHB 为 4 分频时，TIMADV_CLK 需设为 2 分频等）。如无法满足上述要求时，高级定时器时钟请选择 APB2_CLK。

6.2.4.15 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。

6.3 寄存器

6.3.1 寄存器总览

表 6-3 RCC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	时钟控制寄存器	0x00400221
0x04	RCC_CFGR	时钟配置寄存器	0x00000000
0x08	RCC_CIR	时钟中断寄存器	0x00000000
0x14	RCC_AHBRSTR	AHB 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
0x28	RCC_AHBENR	AHB 外设时钟使能寄存器	0x00036000
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	0x40030000
0x44	RCC_ICSCR	内部时钟源校准寄存器	0x02000000
0x48	RCC_PLL1CFGR	PLL1 配置寄存器	0x00000000
0x4C	RCC_PLL2CFGR	PLL2 配置寄存器	0x00000000
0x60	RCC_ADC1CFGR	ADC1 配置寄存器	0x00000000
0x64	RCC_ADC2CFGR	ADC2 配置寄存器	0x00000000
0x70	RCC_DACCFGR	DAC 配置寄存器	0x00000000
0x7C	RCC_TPIUCFGR	TPIU 配置寄存器	0x00000000

6.3.2 RCC_CR 时钟控制寄存器

偏移地址：0x00

复位值：0x0040 0221

访问：无等待状态，字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		PLL2RDY	PLL2ON	Reserved		PLL1RDY	PLL1ON	HSEIB		HSEAACSEL	Reserved	CSSON	HSEBYP	HSERDY	HSEON
		r	rw			r	rw			rw		rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		HSIDIV			HSEDR		HSEOUTPUTSEL	HSEDEGLITCHSEL	HSEDEGLITCHBYP	HSELPFSEL	HSELPFBYP	Reserved		HSIRDY	HSION
		rw			rw		rw	rw	rw	rw	rw			r	rw

Bit	Field	Description
31: 30	Reserved	保留，必须保持复位值
29	PLL2RDY	PLL2 时钟稳定标志（PLL2 Clock Ready Flag） 由硬件设置 0: PLL2 未稳定 1: PLL2 已稳定
28	PLL2ON	PLL2 使能（PLL2 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 PLL2 1: 使能 PLL2
27: 26	Reserved	保留，必须保持复位值
25	PLL1RDY	PLL1 时钟稳定标志（PLL1 Clock Ready Flag） 由硬件设置 0: PLL1 未稳定 1: PLL1 已稳定
24	PLL1ON	PLL1 使能（PLL1 Enable） 通过软件置“1”或清“0”。 当进入停机或待机模式时，由硬件清除。 如果 PLL1 时钟已经或将要作为系统时钟，则禁止重置此位。 0: 禁止 PLL1 1: 使能 PLL1
23: 22	HSEIB	HSE 启动功耗模式选择（HSE Current Mode Select） 00: 低功耗模式 01: 中功耗模式 10: 中高功耗模式 11: 高功耗模式 注：在 HSE 频率为 4M~12M 时，建议配置为 2'b00 或 2'b01；在 HSE 频率为 12M~24M 时，建议配置为 2'b10 或 2'b11。
21	HSEAACSEL	HSE 振荡幅度选择（HSE Amplitude Auto Calibration Select） 0: 默认 1: 振幅增大模式

Bit	Field	Description
20	Reserved	保留，必须保持复位值
19	CSSON	时钟安全系统使能（Clock Security System Enable） 通过软件置“1”或清“0”。 0：禁止时钟安全监测器 1：如果外部高速振荡器稳定，使能时钟安全监测器
18	HSEBYP	外部高速时钟旁路（External High-speed Clock Bypass） 由软件置“1”来旁路外部晶体振荡器。 只有当 HSE 振荡器被禁止时，才能写入 HSEBYP 位。 0：禁止外部高速晶体振荡器旁路模式 1：使能外部高速晶体振荡器旁路模式
17	HSERDY	外部高速时钟稳定标志（External High-speed Clock Ready Flag） 由硬件设置。 0：外部高速晶体振荡器未稳定 1：外部高速晶体振荡器已稳定
16	HSEON	外部高速时钟使能（External High-speed Clock Enable） 通过软件置“1”或清“0”。 当进入待机或停机模式时，此位由硬件清“0”。当系统时钟已经或将要使用 HSE 作为时钟源时，则禁止重置此位。 0：禁止外部高速晶体振荡器 1：使能外部高速晶体振荡器
15: 14	Reserved	保留，必须保持复位值
13: 11	HSIDIV	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。
10: 9	HSEDR	HSE 驱动能力选择（HSE Driver Ability） 00：低驱动模式 01：中驱动模式 10：中高驱动模式 11：高驱动模式 注：根据晶振 ESR 和电容负载选择合适设置。
8	HSEOUTPUTSEL	输出模式选择 0：直接输出 1：滤波输出

Bit	Field	Description
7	HSEDEGLITCHSEL	去毛刺宽度控制位 0: 典型值 8ns 1: 典型值 5ns
6	HSEDEGLITCHBYP	去毛刺电路的 bypass 控制位 0: 正常采用去毛刺功能 1: bypass 掉去毛刺功能
5	HSELPFSEL	LPF 滤波模式控制 0: LPF 滤波后信号放大输出 1: LPF 滤波后输出
4	HSELPFBYP	LPF 输入信号直连输出的使能信号 0: 经过 LPF 滤波输出 1: LPFIN 直连输出
3: 2	Reserved	保留，必须保持复位值
1	HSIRDY	内部高速时钟稳定标志（Internal High-speed Clock Ready Flag） 由硬件置“1”，表示内部时钟已经稳定。 在 HSION 位被清除后，HSIRDY 在 3 个 AHB 时钟周期后变“0”。 0: 内部高速时钟未稳定 1: 内部高速时钟已稳定
0	HSION	内部高速时钟使能（Internal High-speed Clock Enable） 通过软件置“1”或清“0”。 当退出待机或停机模式或外部振荡器用作系统时钟并且发生故障时，此位由硬件置“1”，来迫使内部振荡器使能。当系统时钟已经或将要使用 HSI 作为时钟源时，则禁止重置此位。 0: 禁止内部高速时钟 1: 使能内部高速时钟

6.3.3 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
Reserved				MCO				USBPRE		Reserved		USBCL KSEL	Reserve d	ADC2C LKSEL	ADC1C LKSEL
				rw				rw				rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserve d	Reserve d	PPRE2			PPRE1			HPRE				SWS		SW	
	rw	rw			rw			rw				r		rw	

Bit	Field	Description
31: 28	Reserved	保留，必须保持复位值
27: 24	MCO	微控制器时钟输出（Micro Controller Clock Output） 由软件置“1”或清“0” 000x：没有时钟输出 0010：LSI 时钟输出 0011：LSE 时钟输出 0100：系统时钟（SYSCLK）输出 0101：HSI 时钟输出 0110：HSE 时钟输出 0111：PLL1 时钟输出 1000：PLL2 时钟输出 注： 该时钟输出在启动和切换 MCO 时钟源时可能会被停止。 建议在使用过程中不要随便切换。 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_FS 时钟。 在配置 RCC_AHBENR 寄存器使能 USB_FS 时钟之前，预先设定好，并且使用期间不要再切换。 00：PLL1/PLL2 时钟直接作为 USB_FS 时钟 01：PLL1/PLL2 时钟 2 分频作为 USB_FS 时钟 10：PLL1/PLL2 时钟 3 分频作为 USB_FS 时钟 11：PLL1/PLL2 时钟 4 分频作为 USB_FS 时钟
21: 20	Reserved	保留，必须保持复位值
19	USBCLKSEL	选择 USBCLK 时钟源 0：选择 PLL1 1：选择 PLL2 注：在使能 RCC_AHBENR 寄存器中 USB_FS 时钟之前，预先设定好；并且使用期间不要再切换。
18	Reserved	保留，必须保持复位值
17	ADC2CLKSEL	选择 ADC2CLK 时钟源 0：选择 PCLK_ADC2 1：选择 PLL2 注：在使能 RCC_APB2ENR 寄存器中 ADC2 时钟之前，预先设定好；并且使用期间不要再切换。
16	ADC1CLKSEL	选择 ADC1CLK 时钟源 0：选择 PCLK_ADC1 1：选择 PLL2 注：在使能 RCC_APB2ENR 寄存器中 ADC1 时钟之前，预先设定好；并且使用期间不要再切换。
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	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: HCLK2 分频 101: HCLK4 分频 110: HCLK8 分频 111: HCLK16 分频
10: 8	PPRE1	PPRE1: APB1 预分频系数 通过软件设置来控制 APB1 时钟（PCLK1）预分频系数。 0xx: HCLK 不分频 100: HCLK2 分频 101: HCLK4 分频 110: HCLK8 分频 111: HCLK16 分频
7: 4	HPRE	AHB 预分频系数 通过软件设置来控制 AHB 时钟的预分频系数。 0xxx: SYSCLK 不分频 1000: SYSCLK 2 分频 1001: SYSCLK 4 分频 1010: SYSCLK 8 分频 1011: SYSCLK 16 分频 1100: SYSCLK 64 分频 1101: SYSCLK 128 分频 1110: SYSCLK 256 分频 1111: SYSCLK 512 分频 注: 当 AHB 时钟的预分频系数大于 1 时, 必须开启预取缓冲器, 详见闪存访问控制寄存器章节。
3: 2	SWS	系统时钟选择状态 (System Clock Switch Status) 00: 选择 HSI 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 PLL1 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟
1: 0	SW	系统时钟选择 (System Clock Switch) 通过软件配置来选择系统时钟源 当从停止或待机模式中返回时, 当直接或间接作为系统时钟的 HSE 出现故障时, 硬件都会强制选择 HSI 作为系统时钟。 00: 选择 HSI 输出用作系统时钟 01: 选择 HSE 输出用作系统时钟 10: 选择 PLL1 输出用作系统时钟 11: 选择 LSI 输出用作系统时钟

6.3.4 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
Reserved								CSSC	Reserved	PLL2RD YC	PLL1RD YC	HSERD YC	HSIRDY C	LSERD YC	LSIRDY C
								w1c		w1c	w1c	w1c	w1c	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PLL2RD	PLL1RD	HSERD	HSIRDY	LSERD	LSIRDY	CSSF	Reserved	PLL2RD	PLL1RD	HSERD	HSIRDY	LSERD	LSIRDY
		YIE	YIE	YIE	IE	YIE	E			YF	YF	YF	F	YF	F
		rW	rW	rW	rW	rW	rW			r	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	Reserved	保留，必须保持复位值
21	PLL2RDYC	清除 PLL2 稳定中断（PLL2 Ready Interrupt Clear） 通过软件置“1”来清除 PLL2 稳定中断标志位 PLL2RDYF。 0：无效 1：清除 PLL2 稳定中断标志位 PLL2RDYF
20	PLL1RDYC	清除 PLL1 稳定中断（PLL1 Ready Interrupt Clear） 通过软件置“1”来清除 PLL1 稳定中断标志位 PLL1RDYF。 0：无效 1：清除 PLL1 稳定中断标志位 PLL1RDYF
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

Bit	Field	Description
16	LSIRDYC	清除 LSI 稳定中断 (LSI Ready Interrupt Clear) 通过软件置“1”来清除 LSI 稳定中断标志位 LSIRDYF。 0: 无效 1: 清除 LSI 稳定中断标志位 LSIRDYF
15: 14	Reserved	保留, 必须保持复位值
13	PLL2RDYIE	PLL2 稳定中断使能 (PLL2 Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止 PLL2 稳定中断。 0: 禁止 PLL2 稳定中断 1: 使能 PLL2 稳定中断
12	PLL1RDYIE	PLL1 稳定中断使能 (PLL1 Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止 PLL1 稳定中断。 0: 禁止 PLL1 稳定中断 1: 使能 PLL1 稳定中断
11	HSERDYIE	HSE 稳定中断使能 (HSE Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止外部振荡器稳定中断。 0: 禁止 HSE 稳定中断 1: 使能 HSE 稳定中断
10	HSIRDYIE	HSI 稳定中断使能 (HSI Ready Interrupt 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
9	LSE RDYIE	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
8	LSIRDYIE	LSI 稳定中断使能 (LSI Ready Interrupt Enable) 通过软件置“1”来使能或清“0”来禁止内部 40KHz 振荡器稳定中断。 0: 禁止 LSI 稳定中断 1: 使能 LSI 稳定中断
7	CSSF	时钟安全系统中断标志 (Clock Security System Interrupt Flag) 在外部振荡器时钟出现故障时, 由硬件置“1”。 通过软件将 CSSC 位置“1”来清除。 0: 无 HSE 时钟失效产生的安全系统中断 1: HSE 时钟失效导致时钟安全系统中断
6	Reserved	保留, 必须保持复位值
5	PLL2RDYF	PLL2 稳定中断标志 (PLL2 Ready Interrupt Flag) 在 PLL2 稳定时, 由硬件置“1”。 通过软件将 PLL2RDYC 位置“1”来清除。 0: 无 PLL2 锁定产生的时钟稳定中断 1: PLL2 锁定导致时钟稳定中断

Bit	Field	Description
4	PLL1RDYF	PLL1 稳定中断标志 (PLL1 Ready Interrupt Flag) 在 PLL1 稳定时, 由硬件置“1”。 通过软件将 PLL1RDYC 位置“1”来清除。 0: 无 PLL1 锁定产生的时钟稳定中断 1: PLL1 锁定导致时钟稳定中断
3	HSERDYF	HSE 稳定中断标志 (HSE Ready Interrupt Flag) 在外部高速时钟稳定时, 由硬件置“1”。 通过软件将 HSERDYC 位置“1”来清除。 0: 无外部振荡器产生的时钟稳定中断 1: 外部振荡器导致时钟稳定中断
2	HSIRDYF	HSI 稳定中断标志 (HSI Ready Interrupt Flag) 在内部高速时钟稳定时, 由硬件置“1”。 通过软件将 HSIRDYC 位置“1”来清除。 0: 无内部 HSI 振荡器产生的时钟稳定中断 1: 内部 HSI 振荡器导致时钟稳定中断
1	LSE RDYF	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 LSE RDYC 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
0	LSIRDYF	LSI 稳定中断标志 (LSI Ready Interrupt Flag) 在内部低速时钟稳定时, 由硬件置“1”。 通过软件将 LSIRDYC 位置“1”来清除。 0: 无内部 40KHz 振荡器产生的时钟稳定中断 1: 内部 40KHz 振荡器导致时钟稳定中断

6.3.5 RCC_AHBRSTR AHB 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
Reserved	QSPI	Reserved	FSMC	Reserved			USB_FS	Reserved	DMA2	DMA1	Reserved	ENET	Reserved		
	rw		rw				rw		rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORP	Reserved		CRC	Reserved			GPIOI	GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
rw			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30	QSPI	QSPI 复位（QSPI Reset） 由软件置“1”或清“0” 0：无效 1：复位
29	Reserved	保留，必须保持复位值
28	FSMC	FSMC 复位（FSMC Reset） 由软件置“1”或清“0” 0：无效 1：复位
27: 25	Reserved	保留，必须保持复位值
24	USB_FS	USB_FS 复位（USB_FS 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	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	Reserved	保留，必须保持复位值
19	ENET	ENET 复位（ENET Reset） 由软件置“1”或清“0” 0：无效 1：复位
18: 16	Reserved	保留，必须保持复位值
15	CORDIC	CORDIC 复位（CORDIC Reset） 由软件置“1”或清“0” 0：无效 1：复位
14: 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: 9	Reserved	保留，必须保持复位值

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

6.3.6 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
LPUART	LPTIM	Reserved	MDS	Reserved			FLEXCAN2	Reserved							
rw	rw		rw				rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP	SYSCFG	Reserved	SPI1	Reserved		ADC2	ADC1	Reserved		UART6	UART1	Reserved		TIM8	TIM1
rw	rw		rw			rw	rw			rw	rw			rw	rw

Bit	Field	Description
31	LPUART	LPUART 复位 (LPUART Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
30	LPTIM	LPTIM 复位 (LPTIM Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
29	Reserved	保留，必须保持复位值
28	MDS	MDS 复位 (MDS Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
27: 25	Reserved	保留，必须保持复位值
24	FLEXCAN2	FLEXCAN2 复位 (FLEXCAN2 Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
23: 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

Bit	Field	Description
11: 10	Reserved	保留，必须保持复位值
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
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

6.3.7 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
Reserved	UART7	DAC	PWR	BKP	DBG	FLEXCAN1	CRS	Reserved	I2C2	I2C1	UART5	UART4	UART3	UART2	Reserved
	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	Reserved		WWDG	Reserved					TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
rw	rw			rw						rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Reserved	保留，必须保持复位值
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
28	PWR	PWR 复位 (Power Interface 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	DBG	DBG 复位 (DBG Reset) 由软件置“1”或清“0” 0: 无效 1: 复位
25	FLEXCAN1	FLEXCAN1 复位 (FLEXCAN1 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

Bit	Field	Description
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
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

Bit	Field	Description
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

6.3.8 RCC_AHBENR AHB 外设时钟使能寄存器

偏移地址: 0x28

复位值: 0x0003 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
Reserved	QSPI	Reserved	FSMC	Reserved			USB_FS	Reserved	DMA2	DMA1	Reserved	ENET	Reserved	DTCM	ITCM
	rw	d	rw				rw	d	rw	rw	d	rw	d	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORP	SRAM	FLASH	CRC	Reserved			GPIOI	GPIOH	GPIOG	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA
rw		rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Reserved	保留, 必须保持复位值
30	QSPI	QSPI 时钟使能 (QSPI Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
29	Reserved	保留, 必须保持复位值
28	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
27: 25	Reserved	保留, 必须保持复位值

Bit	Field	Description
24	USB_FS	USB_FS 时钟使能 (USB_FS 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	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	Reserved	保留, 必须保持复位值
19	ENET	ENET 时钟使能 (ENET Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
18	Reserved	保留, 必须保持复位值
17	DTCM	DTCM 时钟使能 (DTCM Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
16	ITCM	ITCM 时钟使能 (ITCM Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
15	CORDIC	CORDIC 时钟使能 (CORDIC Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
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

Bit	Field	Description
11: 9	Reserved	保留，必须保持复位值
8	GPIOI	GPIOI 时钟使能（GPIOI Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
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

6.3.9 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
LPUART	LPTIM	Reserved	MDS	Reserved		FLEXCAN2		Reserved							
rw	rw	d	rw			rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP	SYSCFG	Reserved	SPI1	Reserved		ADC2	ADC1	Reserved		UART6	UART1	Reserved		TIM8	TIM1
rw	rw	d	rw			rw	rw			rw	rw			rw	rw

Bit	Field	Description
31	LPUART	LPUART 时钟使能 (LPUART Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
30	LPTIM	LPTIM 时钟使能 (LPTIM Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
29	Reserved	保留，必须保持复位值
28	MDS	MDS 时钟使能 (MDS Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
27: 25	Reserved	保留，必须保持复位值
24	FLEXCAN2	FLEXCAN2 时钟使能 (FLEXCAN2 Clock Enable) 由软件置“1”或清“0” 0: 时钟禁止 1: 时钟使能
23: 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

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

6.3.10 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

Reserved	UART7	DAC	PWR/D BG	BKP	Reserved	FLEXCA N1	CRS	Reserved	I2C2	I2C1	UART5	UART4	UART3	UART2	Reserved
	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	Reserved		WWDG	Reserved					TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
rw	rw			rw						rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Reserved	保留，必须保持复位值
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
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	FLEXCAN1	FLEXCAN1 时钟使能（FLEXCAN1 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

Bit	Field	Description
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
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

Bit	Field	Description
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

6.3.11 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。该寄存器在读写操作对应 IP (RTC/LSE/BKP) 以后, 下一次读写操作前要等待对应 IP (RTC/LSE/BKP) 所在总线 (APB1/APB2, 其中 RTC 及 LSE 对应 BKP 所在 APB 总线) 的 5 个周期以上。

注: 备份域控制寄存器中 (RCC_BDCR) 的 LSEON、LSEBYP、RTCSEL 和 RTCEN 位在备份域内。因此, 在复位之后, 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
Reserved							DBP	Reserved							BDRST
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL		Reserved					LSEBYP	LSERD Y	LSEON
rw						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

Bit	Field	Description
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
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） 由硬件设置。 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

6.3.12 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

LPWRR STF	WWDG RSTF	IWDG STF	SFTRST F	PORRS TF	PINRST F	Reserved	RMVF	LOCKU PF	PVDRS TF	Reserved	Reserved				
r	r	r	r	r	r	d	w1c	r	r	r					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							Reserved	LOCKU	PVDRS	LSIOE	Reserved			LSIRDY	LSION
							d	PEN	TEN					r	
							rw	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
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

Bit	Field	Description
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	Reserved	保留，必须保持复位值。
20: 9	Reserved	保留，必须保持复位值
8	Reserved	保留，必须保持复位值。
7	LOCKUPEN	CPU 死锁复位使能（CPU Lockup Reset Enable） 0：禁止 CPU 死锁复位 1：使能 CPU 死锁复位
6	PVDRSTEN	PVD 复位使能（PVD Reset Enable） 0：禁止 PVD 产生复位 1：使能 PVD 产生复位
5	LSIOE	LSI 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

6.3.13 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	HSELPF EN	Reserved				HSE_RFB_SEL		Reserved						Reserved	PROG_ CHECK _EN
	rw													r	r

Bit	Field	Description
31: 15	Reserved	保留，必须保持复位值
14	HSELPFEN	外接晶振低通滤波使能 0：禁止 1：使能 注：在使用 HSE 过程中请不要进行修改
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: 2	Reserved	保留，必须保持复位值
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）

6.3.14 RCC_CFGR2 时钟配置寄存器 2

偏移地址：0x40

复位值：0x4003 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

Reserve d	LPTIMCLKSEL		Reserve d	LPUARTCLKSEL		Reserved		MCO_PRE				APB1_CLK_HV_PRE			
	rw			rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TIMADV_PRE		TIMADV_CKSEL	
												rw		rw	

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30: 29	LPTIMCLKSEL	LPTIM 时钟源选择 00: LSE_CLK 01: LSI_CLK 10: PCLK_LPTIM (默认) 11: HSI_CLK
28	Reserved	保留，必须保持复位值
27: 26	LPUARTCLKSEL	LPUART 时钟源选择 00: LSE_CLK (默认) 01: LSI_CLK 10: PCLK_LPUART 11: HSI_CLK
25: 24	Reserved	保留，必须保持复位值
23: 20	MCO_PRE	MCO 时钟预分频系数 0xxx: MCO 不分频 1000: MCO 2 分频 1001: MCO 4 分频 1010: MCO 8 分频 1011: MCO 16 分频 1100: MCO 64 分频 1101: MCO 128 分频 1110: MCO 256 分频 1111: MCO 512 分频

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: 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

6.3.15 RCC_ICSCR 内部时钟校准寄存器

偏移地址: 0x44

复位值: 0x0200 0000

访问: 无等待状态, 字, 半字和字节访问

只有当访问发生在时钟切换时, 才会插入 1 或 2 个等待周期

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

	rw

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

6.3.16 RCC_PLL1CFGR PLL1 配置寄存器

偏移地址: 0x48

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

Bit	Field	Description
31: 23	Reserved	保留，必须保持复位值
22: 16	PLL1MUL	PLL1 倍频系数 (PLL1 Multiplication Factor)
15: 11	Reserved	保留，必须保持复位值
10: 8	PLL1DIV	PLL1 分频系数 (PLL1 Divide Factor)
7	Reserved	保留，必须保持复位值
6: 4	PLL1_LDS	PLL1 锁定检测器精度选择 (PLL1 Lock Detector Accuracy Select) 000: 关闭精度 001: 高精度 010: 次高精度 011: 中高精度 100: 中精度 101: 中低精度 110: 次低精度 111: 低精度 推荐设置此位为 3'b111。

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

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

6.3.17 RCC_PLL2CFGR PLL2 配置寄存器

偏移地址: 0x4C

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

Bit	Field	Description
31: 27	Reserved	保留, 必须保持复位值
26: 24	PLL2PDIV	PLL2 预分频系数 (PLL2 Pre-divide Factor)
23: 16	PLL2MUL	PLL2 倍频系数 (PLL2 Multiplication Factor)
15: 11	Reserved	保留, 必须保持复位值
10: 8	PLL2DIV	PLL2 分频系数 (PLL2 Divide Factor)
7	Reserved	保留, 必须保持复位值

Bit	Field	Description
6: 4	PLL2_LDS	PLL2 锁定检测器精度选择 (PLL2 Lock Detector Accuracy Select) 000: 无精度 001: 高精度 010: 次高精度 011: 中高精度 100: 中精度 101: 中低精度 110: 次低精度 111: 低精度
3: 2	PLL2_ICTRL	PLL2 Charge Pump 电流控制信号 (PLL2 CP Current Control Signals) 当 PLL2 输入时钟源大于等于 8MHz 时, 推荐设置此位为 2'b11; 当 PLL2 输入时钟源低于 8MHz 时, 推荐设置此位为 2'b01。
1	PLL2XTPRE	HSE 时钟用作 PLL2 输入时的分频选择 通过软件置“1”或清除来分频 HSE 时钟后用作 PLL2 输入时钟。 只有当 PLL2 被禁止时, 才能写入此位。 0: HSE 时钟不分频 1: HSE 时钟 2 分频
0	PLL2SRC	PLL2 输入时钟源选择 (PLL2 Entry Clock Source Selection) 通过软件置“1”或清除来选择 PLL2 输入时钟源。 只有当 PLL2 被禁止时, 才能写入此位。 0: HSI 时钟用作 PLL2 输入时钟 1: HSE 时钟用作 PLL2 输入时钟

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

6.3.18 RCC_ADC1CFGR ADC1 配置寄存器

偏移地址: 0x60

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

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值

Bit	Field	Description
16: 8	PRECAL	ADC1 calibration 时钟分频配置（频率范围 187.5k~1.5M） PCLK2/（PRECAL+2），要求 PRECAL 为偶数，使占空比 50%。
7: 4	Reserved	保留，必须保持复位值
3: 0	PRE	ADC1 预分频（频率范围 15M~48M） 由软件置 1 或清 0 来确定 ADC1 时钟频率。 PCLK2/（PRE+2），要求 PRE 为偶数，使占空比 50%。

注：PCLK2_ADC 的频率要快于 ADC_CLK。如：当 ADC_CLK 为 48M 时，PCLK2_ADC 建议配置 72M 及以上

6.3.19 RCC_ADC2CFGR ADC2 配置寄存器

偏移地址：0x64

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

Bit	Field	Description
31: 17	Reserved	保留，必须保持复位值
16: 8	PRECAL	ADC2 calibration 时钟分频配置（频率范围 187.5k~1.5M） PCLK2/（PRECAL+2），要求 PRECAL 为偶数，使占空比 50%。
7: 4	Reserved	保留，必须保持复位值
3: 0	PRE	ADC2 预分频（频率范围 15M~48M） 由软件置 1 或清 0 来确定 ADC2 时钟频率。 PCLK2/（PRE+2），要求 PRE 为偶数，使占空比 50%。

注：PCLK2_ADC 的频率要快于 ADC_CLK。如：当 ADC_CLK 为 48M 时，PCLK2_ADC 建议配置 72M 及以上

6.3.20 RCC_DACCFGR DAC 配置寄存器

偏移地址：0x70

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

Bit	Field	Description
31: 7	Reserved	保留，必须保持复位值
6: 0	PRE	DAC 预分频 PCLK1/2* (n+1) 分频后作为 DAC 时钟。

6.3.21 RCC_TPIUCFGR TPIU 配置寄存器

偏移地址：0x7C

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

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1: 0	PRE	TPIU 预分频 00: 不分频 01: 2 分频 10: 3 分频 11: 4 分频

7 CRS 时钟回馈系统

7.1 介绍

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。

7.2 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）

7.3 功能描述

7.3.1 功能框图

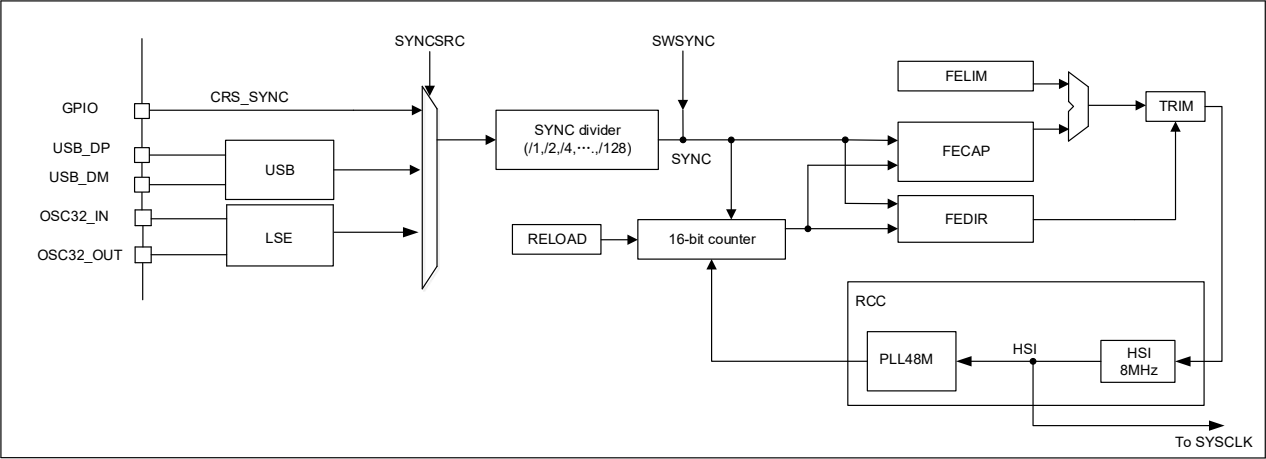


图 7-1 CRS 功能框图

7.3.2 信号描述

信号名	类型	描述
GPIO	输入	来自 PAD 的同步源输入信号
USB_DP/USB_DM	输入	来自 USB 的同步源信号
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

7.3.3 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

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。

7.3.4 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)。

7.3.5 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 RELOAD 值被重新载入该计数器，进行重新计数。在重载后，计数器开始向下重新计数。计数器递减为 0 时，对应 ESYNC 标志位置起。之后计数器自动转换计数方向为向上计数。

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

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

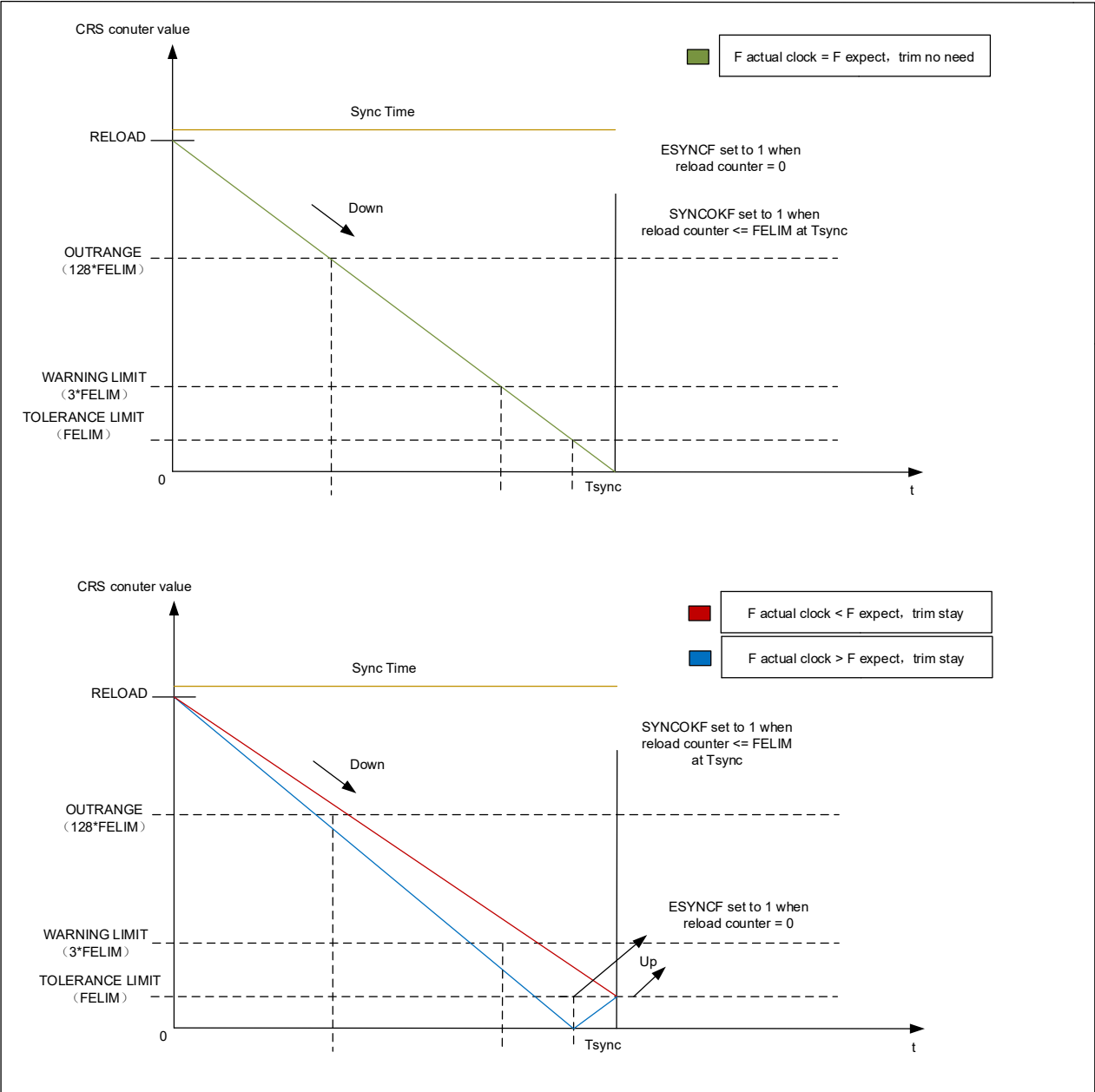
- 停止时 FEDIR=0，即向上计数，实际频率大于期望频率；
- 停止时 FEDIR=1，即向下计数，实际频率小于或等于期望频率。

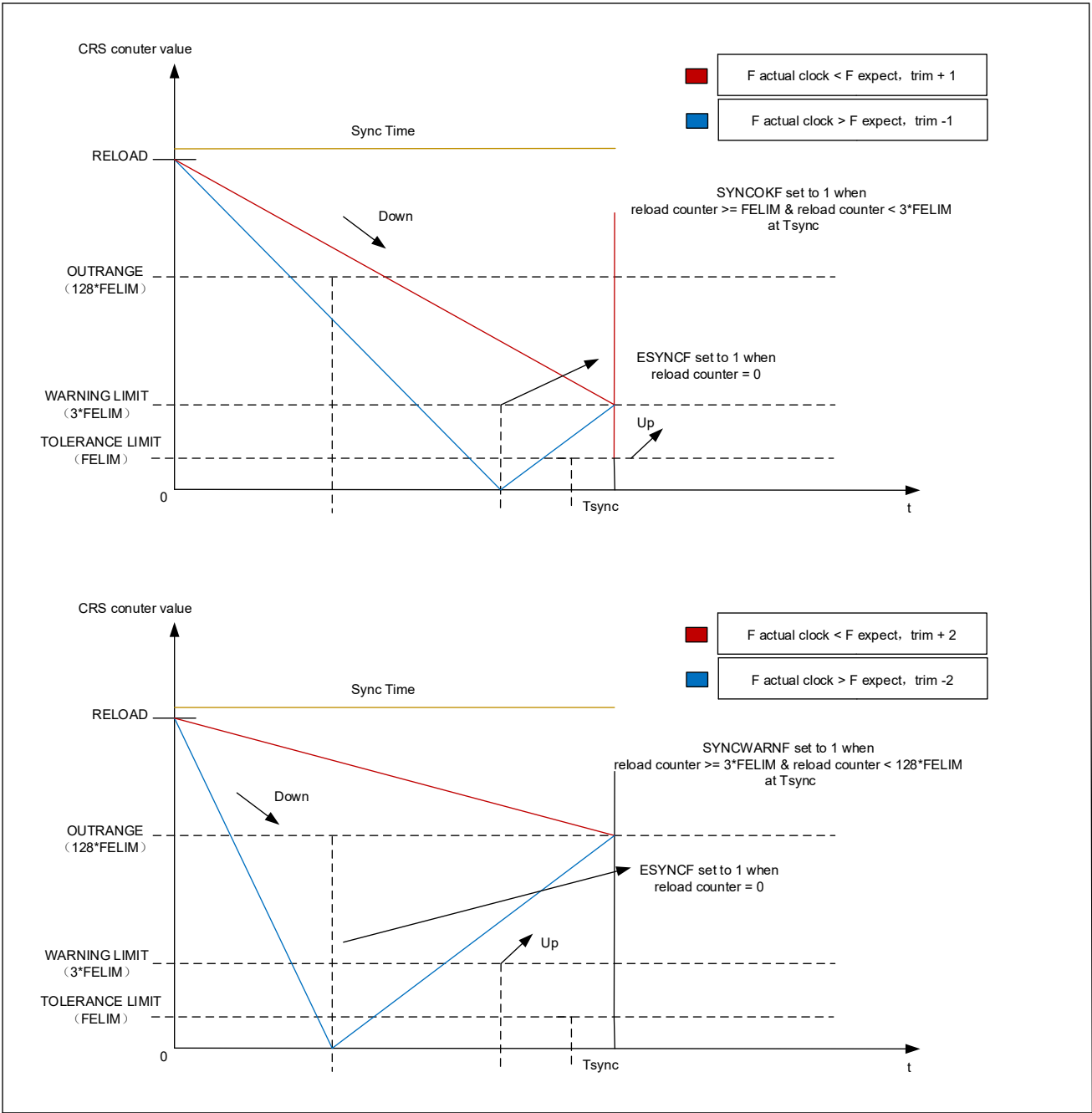
根据计数器因 SYNC 事件停止时的计数值与界限值（界限值可配置，为 CRS_CFGR 寄存器的 FELIM 位）之间的差值，可以得到微调时的步长：

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

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

下图列举了计数器测量时可能发生的各种情况。





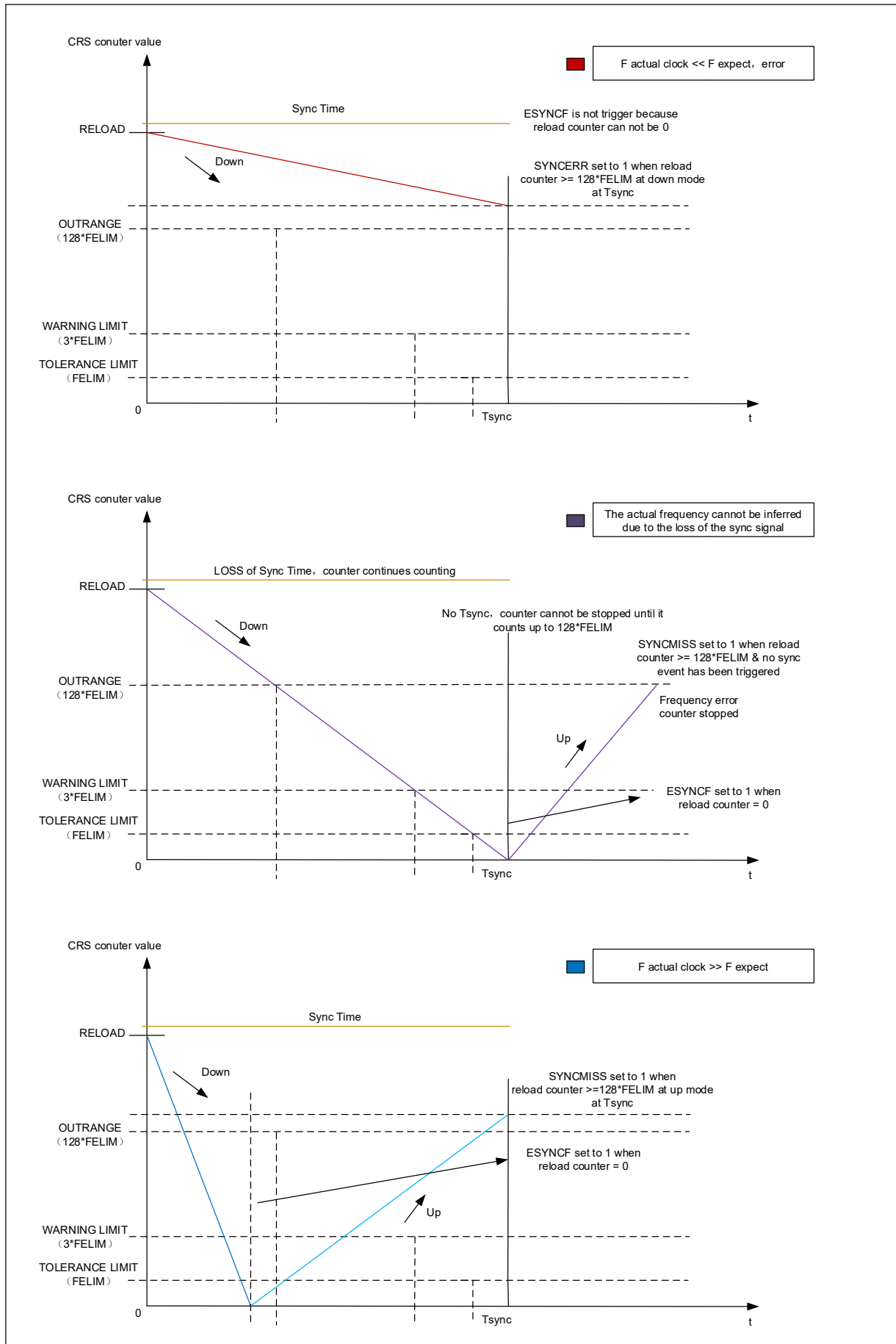


图 7-2 CRS 计数器状态图

7.3.6 频率误差计算及自动校准

通过比较频率误差值与定义的界限区间，可以得到实际频率与期望频率的相对关系：

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

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

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

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

自动校准（AUTOTRIM）模式下，TRIM 位仅由硬件配置且软件只读。

7.3.7 CRS 初始化及配置

7.3.7.1 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）。

7.3.7.2 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。

7.4 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。

7.5 中断

表 7-1 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	SYNCOKIE	SYNCOKC
同步时误差较大	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同步错误或校准错误 (TRIMOVF, SYNCMISS, SYNCERR)	ERRF	ERRIE	ERRC



7.6 寄存器

7.6.1 寄存器总览

表 7-2 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

7.6.2 CRS_CR 控制寄存器

地址偏移: 0x00

复位值: 0x0002_0000

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

Bit	Field	Description
31:18	Reserved	保留, 必须保持复位值
17:8	TRIM[9:0]	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_CRSEL 置 1 时, 写入的值可以重新校正 HSI 频率, 否则写入的值不起作用。 当 AUTOTRIMEN 被设置, 此 TRIM 值由硬件控制, 且为只读。 全 0: 频率最小 全 1: 频率最大
7	SWSYNC	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。 写 0: 无操作 写 1: 软件 SYNC 行为被触发 (生成软件 SYNC 事件)

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	ESYNCE	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

7.6.3 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
SYNCP OL	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。

7.6.4 CRS_ISR 中断状态寄存器

地址偏移: 0x08

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FECAP															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.				TRIMOV	SYNCF	SYNCE	Res.				ESYNC	ERRF	SYNCW	SYNCO
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	频率错误方向（Frequency Error Direction） 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。 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
14:11	Reserved	保留，必须保持复位值
10	TRIMOVF	修正溢出（Trimming Overflow） 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。 0：没有修正溢出错误 1：产生修正溢出错误
9	SYNCMISS	同步信号丢失（SYNC Missed） 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。 0：没有同步信号丢失错误 1：产生同步信号丢失错误
8	SYNCERR	同步信号错误（SYNC Error） 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。 0：没有同步信号错误 1：产生同步信号错误
7:4	Reserved	保留，必须保持复位值
3	ESYNCF	预期同步标志（Expected SYNC Flag） 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。 0：没有预期同步信号 1：产生预期同步信号
2	ERRF	错误标志（Error Flag） 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。 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

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

7.6.5 CRS_ICR 中断标志清除寄存器

地址偏移：0x0C

复位值：0x0000_0000

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

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

8 SYSCFG 系统控制器

8.1 SYSCFG 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

8.2 寄存器描述

表 8-1 SYSCFG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR1	SYSCFG 配置寄存器 1	0x2009000B
0x08	SYSCFG_EXTICR1	外部中断配置寄存器 1	0x00000000
0x0C	SYSCFG_EXTICR2	外部中断配置寄存器 2	0x00000000
0x10	SYSCFG_EXTICR3	外部中断配置寄存器 3	0x00000000
0x14	SYSCFG_EXTICR4	外部中断配置寄存器 4	0x00000000
0x18	SYSCFG_CFGR2	SYSCFG 配置寄存器 2	0x00000000
0x1C	SYSCFG_PDETCSR	电源检测配置状态寄存器	0x00000006
0x20	SYSCFG_VOSDLY	VOSDLY 配置寄存器	0x000001F4
0x24	SYSCFG_DMAMP	DMA 重映射寄存器	0x00000000
0x28	SYSCFG_BUS_PRI	总线优先级配置寄存器	0x00028853

1.2.1 SYSCFG_CFGR1 配置寄存器 1

偏移地址：0x00

复位值：0x2009000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FC_MODESEL		FC_OD ATAEN	Res.						CAN2_S TOP	CAN2_T EST	CAN2_S PV	CAN1_S TOP	CAN1_T EST	CAN1_S PV
	rw	rw	rw							rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												USB_FS _IDOVR	USB_FS _VBUS OVR	USB_FS _IDOW DIS	USB_FS _VBUS OWDIS
												rw	rw	rw	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: 允许作为数据地址使用
27:22	Reserved	保留，必须保持复位值
21	CAN2_STOP	CAN2 调试模式 (FlexCAN2 Debug Stop) 1: CAN2 进入调试模式 0: CAN2 处于正常工作模式
20	CAN2_TEST	CAN2 测试模式 (FlexCAN2 Test Access) 1: 指示当前访问具有测试模式特权。 0: 指示当前访问不具有测试模式特权。
19	CAN2_SPV	CAN2 监察模式 (FlexCAN2 Supervisor Access) 1: 指示当前访问具有监察模式特权 0: 指示当前访问不具有监察模式特权
18	CAN1_STOP	CAN1 调试模式 (FlexCAN1 Debug Stop) 1: CAN1 进入调试模式 0: CAN1 处于正常工作模式
17	CAN1_TEST	CAN1 测试模式 (FlexCAN1 Test Access) 1: 指示当前访问具有测试模式特权。 0: 指示当前访问不具有测试模式特权。
16	CAN1_SPV	CAN1 监察模式 (FlexCAN1 Supervisor Access) 1: 指示当前访问具有监察模式特权 0: 指示当前访问不具有监察模式特权
15:4	Reserved	保留，必须保持复位值
3	USB_FS_IDORV	USB_FS ID 端口改写值 (USB_FS ID Override Value) 当 USB_FS_IDOWDIS = 0 时，该寄存器的值将覆盖 USB_FS_ID 端口的值从而实现 ID 端口值改写功能。
2	USB_FS_VBUSORV	USB_FS VBUS 端口改写值 (USB_FS VBUS Override Value) 当 USB_FS_VBUSOWDIS = 0 时，该寄存器的值将覆盖 USB_FS_VBUS 端口的值从而实现 VBUS 端口值改写功能。
1	USB_FS_IDOWDIS	USB_FS ID 端口值改写禁止 (USB_FS ID Override Disable) 1: 禁止 USB_FS_ID 端口改写，受 USB_FS 控制器使能控制 0: 允许 USB_FS_ID 端口改写，此时释放 USB_FS_ID 端口，可以作为其它目的使用。
0	USB_FS_VBUSOWDIS	USB_FS VBUS 端口值改写禁止 (USB_FS VBUS Override Disable) 1: 禁止 USB_FS_VBUS 端口改写，受 USB_FS 控制器使能控制 0: 允许 USB_FS_VBUS 端口改写，此时释放 USB_FS_VBUS 端口，可以作为其它目的使用。

1.2.2 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	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
15:0	EXTIx	EXTIx 配置（x=0...3）（EXTIx configuration）这些位可用于软件读写。用于选择 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 1000: PI[x] 管脚

1.2.3 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	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

Bit	Field	Description
15:0	EXTIx	EXTIx 配置 (x=4...7) (EXTIx configuration) 这些位可用于软件读写。用于选择 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

1.2.4 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	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
15:0	EXTIx	EXTIx 配置 (x=8...11) (EXTIx configuration) 这些位可用于软件读写。用于选择 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

1.2.5 SYSCFG_EXTICR4 外部中断配置寄存器 4

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

EXTI15				EXTI14				EXTI13				EXTI12			
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
15:0	EXTIx	EXTIx 配置 (x=12...15) (EXTIx configuration) 这些位可用于软件读写。用于选择 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

1.2.6 SYSCFG_CFGR2 配置寄存器 2

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.						QSPI_A	Res.				ENET_S	ENET_	Res.		I2C2_M	I2C1_MO
						UTOIDL					PD	MII_RMI			ODE	
						E_EN					_SEL	I_SEL			_SEL	DE_SEL
						rw					rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.																

Bit	Field	Description
31:26	Reserved	保留，必须保持复位值
25	QSPI_AUTOIDLE_EN	QSPI 无操作时自动 IDLE 控制 1: 使能自动控制 0: 关闭自动控制
24: 22	Reserved	保留，必须保持复位值
21	ENET_SPD_SEL	ENET 接口速度选择 1: 100Mbps 0: 10Mbps
20	ENET_MII_RMII_SEL	ENET PHY 接口选择 1: 选择 RMII 接口 0: 选择 MII 接口 注: 必须在 ENET PHY 处于复位状态下并且在使能 ENET 时钟之前完成此配置。

Bit	Field	Description
19:18	Reserved	保留，必须保持复位值
17	I2C2_MODE_SEL	I2C2 端口模式选择位 0: 开漏模式 1: 推挽模式
16	I2C1_MODE_SEL	I2C1 端口模式选择位 0: 开漏模式 1: 推挽模式
15:0	Reserved	保留，必须保持复位值

1.2.7 SYSCFG_PDETCSR 电源检测配置状态寄存器

偏移地址: 0x1C

复位值: 0x00000006

31	30	29	28	27	26	25	24	23	22	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 _DIV3 _EN	Reserved		Reser ved	Res.	Reser ved	PVDO	PLS				PVDE
				rw	rw	rw	rw		r	r	rw	rw	rw	rw	rw

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11	Reserved	保留，必须保持复位值。
10:9	Reserved	保留，必须保持复位值。
8	Reserved	保留，必须保持复位值。
7	Reserved	保留，必须保持复位值
6	Reserved	保留，必须保持复位值。
5	PVDO	PVD 输出状态 1: PVD 输出高 0: PVD 输出低 注意: PVDE 必须使能。

Bit	Field	Description
4:1	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	PVD 使能 1: PVD 使能 0: PVD 禁止

1.2.8 SYSCFG_VOSDLY 电压调试器输出延时寄存器

偏移地址: 0x20

复位值: 0x000001F4

31	30	29	28	27	26	25	24	23	22	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)

1.2.9 SYSCFG_DMARMP DMA 重映射寄存器

偏移地址: 0x24

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DAC_C	DAC_C	Res.				UART7_	UART7_	UART6_	UART6_	Res.		UART4_	UART4_R
		H2_RM	H1_RM					TX_RM	RX_TM	TX_RM	RX_RM			TX_RM	X_RMP
		P	P					P	P	P	P			P	
		rw	rw					rw	rw	rw	rw			rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM8_COM_RMP	TIM8_UP_RMP	TIM8_TRIG_RMP	TIM7_UP_RMP	Res.	TIM6_UP_RMP	Res.	TIM5_UP_RMP	Res.		TIM2_COM_RMP	TIM2_COM_RMP	TIM2_UP_RMP		TIM1_COM_RMP	TIM1_TRIG_RMP
rw	rw	rw	rw		rw		rw			rw	rw	rw		rw	rw

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29	DAC_CH2_RMP	0: DAC_CH2 的 DMA 功能映射到 DMA2 channel4 1: DAC_CH2 的 DMA 功能重映射到 DMA2 channel7
28	DAC_CH1_RMP	0: DAC_CH1 的 DMA 功能映射到 DMA2 channel3 1: DAC_CH1 的 DMA 功能重映射到 DMA2 channel7
27: 24	Reserved	保留，必须保持复位值
23	UART7_TX_RMP	0: UART7 TX 的 DMA 功能映射到 DMA2 channel2 1: UART7 TX 的 DMA 功能重映射到 DMA2 channel8
22	UART7_RX_RMP	0: UART7 RX 的 DMA 功能映射到 DMA2 channel1 1: UART7 RX 的 DMA 功能重映射到 DMA2 channel7
21	UART6_TX_RMP	0: UART6 TX 的 DMA 功能映射到 DMA2 channel4 1: UART6 TX 的 DMA 功能重映射到 DMA1 channel8
20	UART6_RX_RMP	0: UART6 RX 的 DMA 功能映射到 DMA1 channel1 1: UART6 RX 的 DMA 功能重映射到 DMA2 channel6
19:18	Reserved	保留，必须保持复位值
17	UART4_TX_RMP	0: UART4 TX 的 DMA 功能映射到 DMA2 channel5 1: UART4 TX 的 DMA 功能重映射到 DMA2 channel8
16	UART4_RX_RMP	0: UART4 RX 的 DMA 功能映射到 DMA2 channel3 1: UART4 RX 的 DMA 功能重映射到 DMA2 channel7
15	TIM8_COM_RMP	0: TIM8 COM 的 DMA 功能映射到 DMA2 channel2 1: TIM8 COM 的 DMA 功能重映射到 DMA2 channel8
14	TIM8_UP_RMP	0: TIM8 UP 的 DMA 功能映射到 DMA2 channel1 1: TIM8 UP 的 DMA 功能重映射到 DMA2 channel6
13	TIM8_TRIG_RMP	0: TIM8 COM 的 DMA 功能映射到 DMA2 channel2 1: TIM8 COM 的 DMA 功能重映射到 DMA2 channel7
12	TIM7_UP_RMP	0: TIM7 UP 的 DMA 功能映射到 DMA2 channel4 1: TIM7 UP 的 DMA 功能重映射到 DMA2 channel6
11	Reserved	保留，必须保持复位值
10	TIM6_UP_RMP	0: TIM6 UP 的 DMA 功能映射到 DMA2 channel3 1: TIM6 UP 的 DMA 功能重映射到 DMA2 channel6
9	Reserved	保留，必须保持复位值
8	TIM5_UP_RMP	0: TIM5 UP 的 DMA 功能映射到 DMA2 channel2 1: TIM5 UP 的 DMA 功能重映射到 DMA2 channel6
7:6	Reserved	保留，必须保持复位值

Bit	Field	Description
5	TIM2_CC4_RMP	0: TIM2 CC4 的 DMA 功能映射到 DMA1 channel7 1: TIM2 CC4 的 DMA 功能重映射到 DMA1 channel8
4	TIM2_CC3_RMP	0: TIM2 CC3 的 DMA 功能映射到 DMA1 channel1 1: TIM2 CC3 的 DMA 功能重映射到 DMA1 channel3
3:2	TIM2_UP_RMP	0: TIM2 UP 的 DMA 功能映射到 DMA1 channel2 1: TIM2 UP 的 DMA 功能重映射到 DMA1 channel3 2: TIM2 UP 的 DMA 功能映射到 DMA1 channel8 3: Reserved
1	TIM1_COM_RMP	0: TIM1 COM 的 DMA 功能映射到 DMA1 channel4 1: TIM1 COM 的 DMA 功能重映射到 DMA1 channel8
0	TIM1_TRIG_RMP	0: TIM1 TRIG 的 DMA 功能映射到 DMA1 channel4 1: TIM1 TRIG 的 DMA 功能重映射到 DMA1 channel7

1.2.10 SYSCFG_BUS_PRI 总线优先级配置寄存器

偏移地址: 0x28

复位值: 0x00028853

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														PRI_M6	
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_M6		PRI_M5			PRI_M4			PRI_M3			PRI_M2			PRI_M1	
rw		rw			rw			rw			rw			rw	

Bit	Field	Description
31:8	Reserved	保留，必须保持复位值
17:15	PRI_M6	M6(ENET)优先级 3'b000: 最高优先级 3'b101: 最低优先级
14:12	PRI_M5	M5(USB)优先级 3'b000: 最高优先级 3'b101: 最低优先级
11:9	PRI_M4	M4(DMA2)优先级 3'b000: 最高优先级 3'b101: 最低优先级
8:6	PRI_M3	M3(DMA1)优先级 3'b000: 最高优先级 3'b101: 最低优先级
5:3	PRI_M2	M2(C-BUS)优先级 3'b000: 最高优先级 3'b101: 最低优先级

Bit	Field	Description
2:0	PRI_M1	M1(S-BUS)优先级 3'b000: 最高优先级 3'b101: 最低优先级

注：PRI_M*配置不可以大于 3'b101

9 EXTI 中断和事件

9.1 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 4 位的中断优先级配置位，从而可提供 16 个中断优先级等级。

EXTI 模块包括边沿检测电路，能够产生中断请求或者唤醒事件，边沿检测支持上升沿、下降沿、任意边沿配置。每一个边沿检测电路支持独立的使能与屏蔽。

9.2 主要特征

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

9.3 功能描述

9.3.1 功能框图

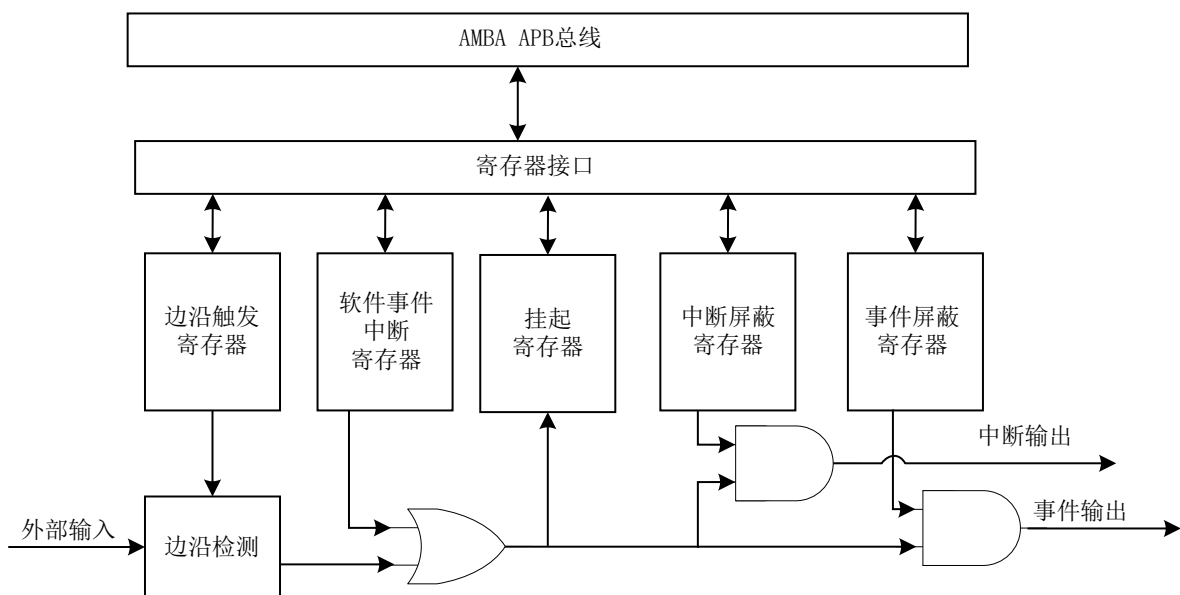


图 9-1 EXTI 结构框图

9.3.2 中断和异常向量

在 Handler 模式下，Star-MC1 处理器与内嵌中断向量控制（NVIC）对所有的异常进行优先级区分处理。当异常发生时，系统会将当前处理的工作压栈，执行完中断服务程序后出栈。取向量与当前工作的压栈并行进行的，提高了中断的效率，具体异常与中断向量的向量编号与地址参考芯片特定配置章节部份。

9.3.3 唤醒事件管理

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

9.3.4 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。

9.3.5 硬件中断输出

配置硬件中断源的具体步骤如下：

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

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

清除 EXTI_PR 寄存器挂起位有以下三种方式：

- EXTI_PR 寄存器的挂起位写 1。
- 如果配置了上升沿触发选择寄存器（EXTI_RTSR），对应位写 0 会清除挂起位。如果配置了下降沿触发选择寄存器（EXTI_FTSR），对应位写 0 会清除挂起位。
- 通过改变 EXTI 线的边沿检测极性清除。

9.3.6 硬件事件输出

配置硬件事件源的具体步骤如下：

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

9.3.7 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）。

9.3.8 外部中断映射

所有的 GPIO 均可用做 EXTI 的触发源用于产生中断或事件请求，通过配置 SYSCFG 章节的 SYSCFG_EXTICRx 寄存器，同时支持内部模块（包括 PVD、RTC、USB、比较器、IWDG）触发。

具体存在的连接参考芯片特定配置章节部分。

9.4 寄存器

9.4.1 寄存器总览

表 9-1 EXTI 寄存器总览

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

9.4.2 EXTI_IMR 中断屏蔽寄存器

偏移地址：0x0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						IMRx(x=25~16)									
						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
31:26	Reserved	保留, 必须保持复位值
25:0	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

9.4.3 EXTI_EMR 事件屏蔽寄存器

偏移地址: 0x04

复位值: 0x0000 0000

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

Bit	Field	Description
31:26	Reserved	保留, 必须保持复位值
25:0	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

9.4.4 EXTI_RTSCR 上升沿触发选择寄存器

偏移地址: 0x08

复位值: 0x0000 0000

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

Bit	Field	Description
31:26	Reserved	保留, 必须保持复位值

Bit	Field	Description
25:0	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

9.4.5 EXTI_FTSR 下降沿触发选择寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

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

Bit	Field	Description
31:26	Reserved	保留, 必须保持复位值
25:0	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

9.4.6 EXTI_SWIER 软件中断事件寄存器

偏移地址: 0x10

复位值: 0x0000 0000

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

Bit	Field	Description
31:26	Reserved	保留, 必须保持复位值
25:0	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

9.4.7 EXTI_PR 软件中断事件挂起寄存器

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						PRx(x=25~16)									
						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
31:26	Reserved	保留，必须保持复位值
25:0	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。

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
- 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

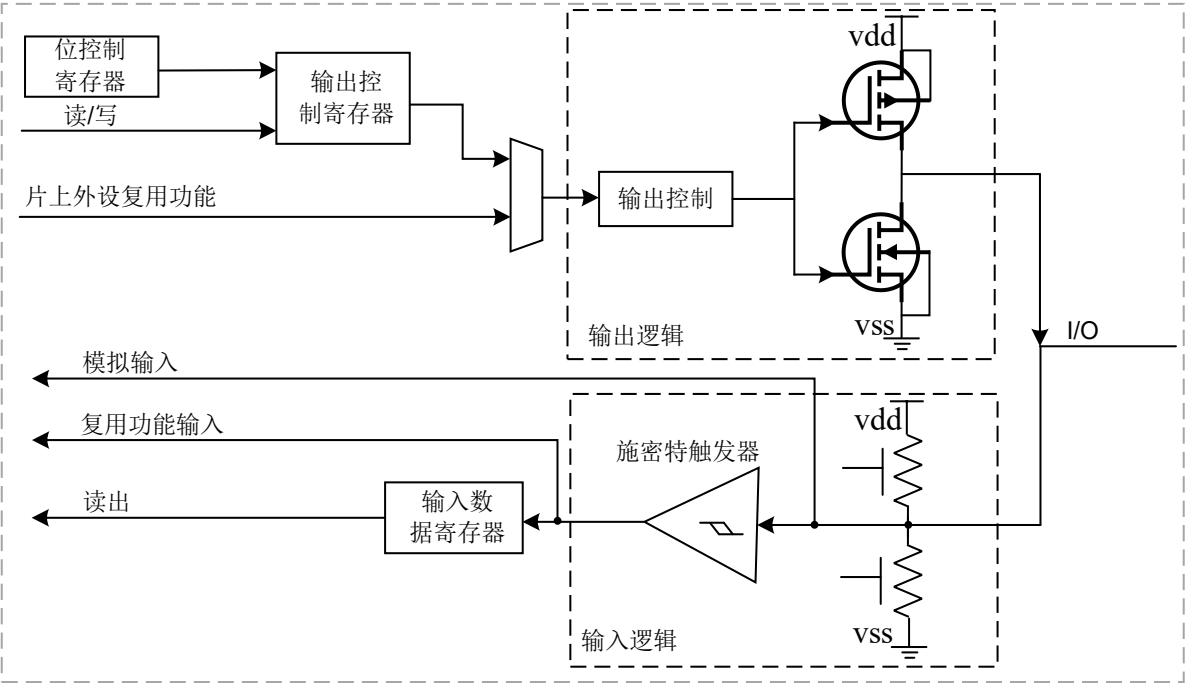


图 10-1 标准 I/O 端口

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		浮空	x	x	0	1		x
复用输入		上拉	x	x	1	0		1
		下拉	x	x	1	0		0
通用输出	推挽	x	x	x	0	0	01 10 11	0 or 1
	开漏	浮空	x	0	0	1		0 or 1
		上拉	1	1	0	1		0 or 1
		下拉	0	1	0	1		0 or 1
复用输出	推挽	x	x	x	1	0		x
	开漏	浮空	x	0	1	1		x
		上拉	1	1	1	1		x
		下拉	0	1	1	1		x

注：x 表示 I/O 在对应的模式下不用关心，ODR0 代表输出数据寄存器第 0 位。

输入输出参考配置如下：

- 通用输入：

用户只需配置 GPIOx_CRL 中的 CNF0 选择输入模式

- 通用输出：

推挽输出：用户配置 MODE0 选择输出速度，配置 CNF0=00；

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

- 复用功能：

配置 AFRLx[3:0]与 AFRHx[3:0]寄存器选择复用功能：

推挽复用输出：用户配置 MODE0 选择输出速度，配置 CNF0=10；

推挽开漏输出：用户配置 MODE0 选择输出速度，配置 CNF0=11。

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

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

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

注：并不是所有芯片都包括 JTAG 和 SWD 调试端口，芯片具体配置可参考芯片数据手册。

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

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，在下次复位之前，不能改变端口对应的配置。锁定键写序列为：

- GPIOx_LCKR[16]='1'+LCKR[15:0]。

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

使能 GPIOA 的 PA[0]端口锁定参考配置如下：

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

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

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

注意事项：

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

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：

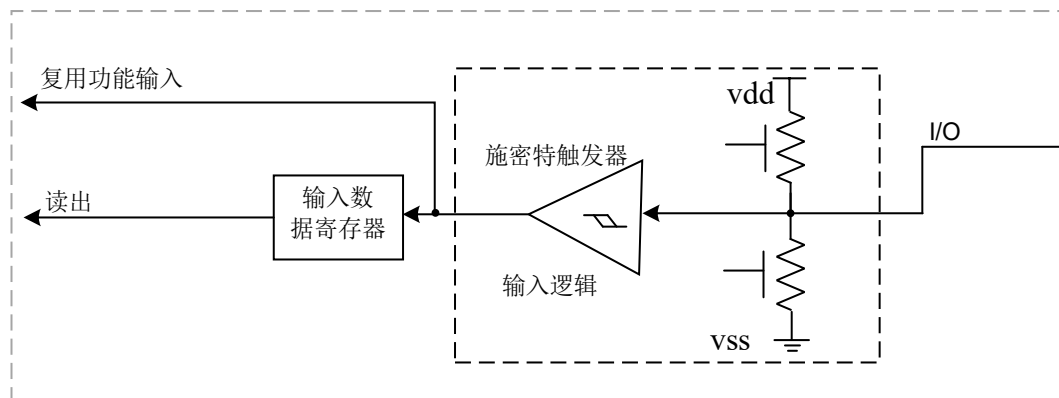


图 10-2 输入浮空/上拉/下拉配置

GPIOA 的 PA[0]端口输入上拉参考配置如下：

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

GPIOA 的 PA[0]端口输入下拉参考配置如下：

- GPIOA->GPIOA_ODR=0x0000。
- GPIOA->GPIOA_CRL=0x00000008。

注意事项：

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

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

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：

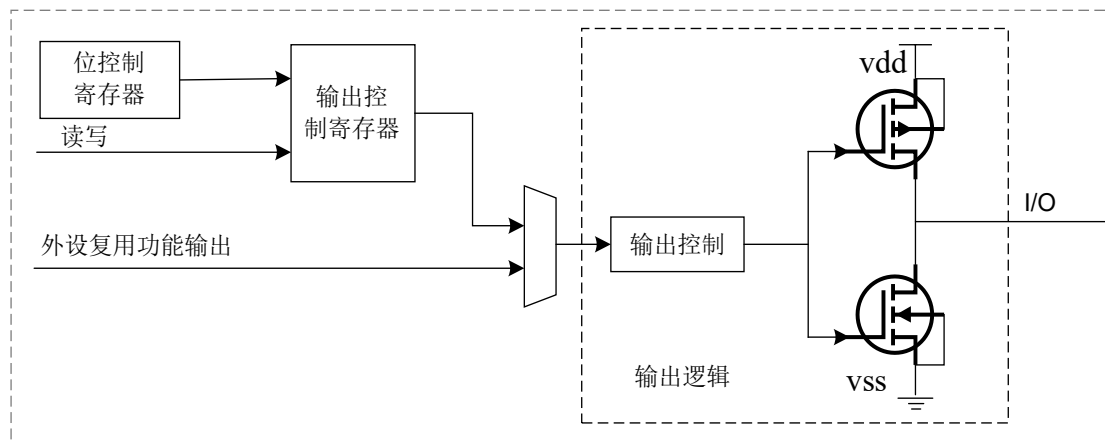


图 10-3 输出配置

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

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

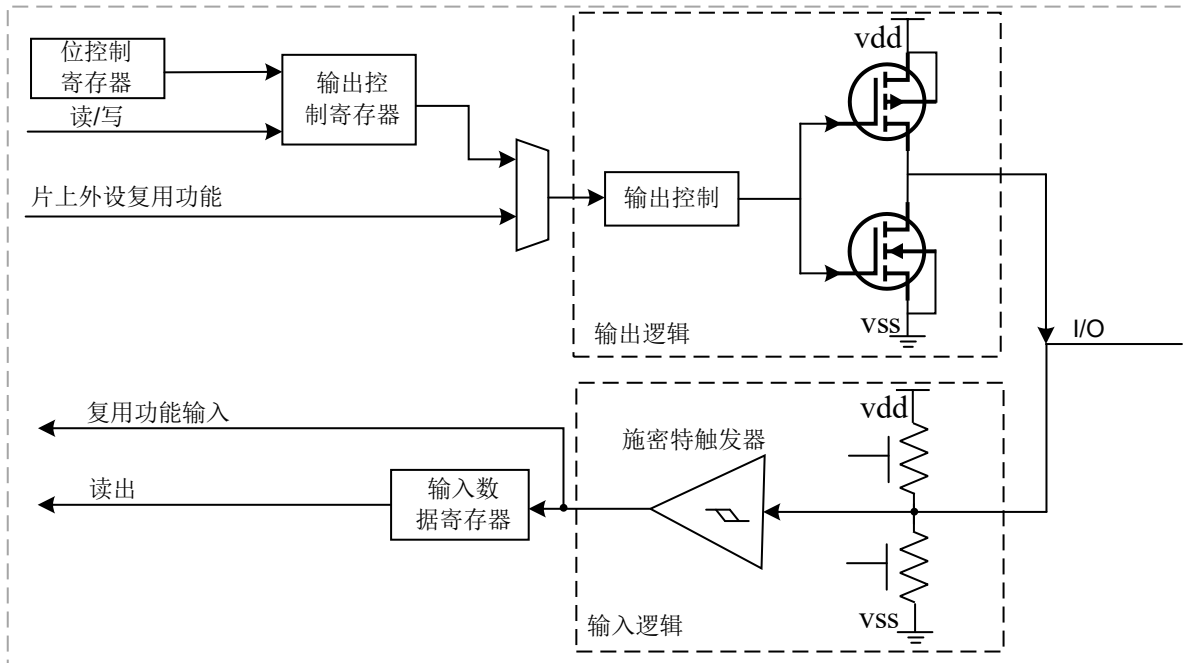


图 10-4 复用功能配置

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

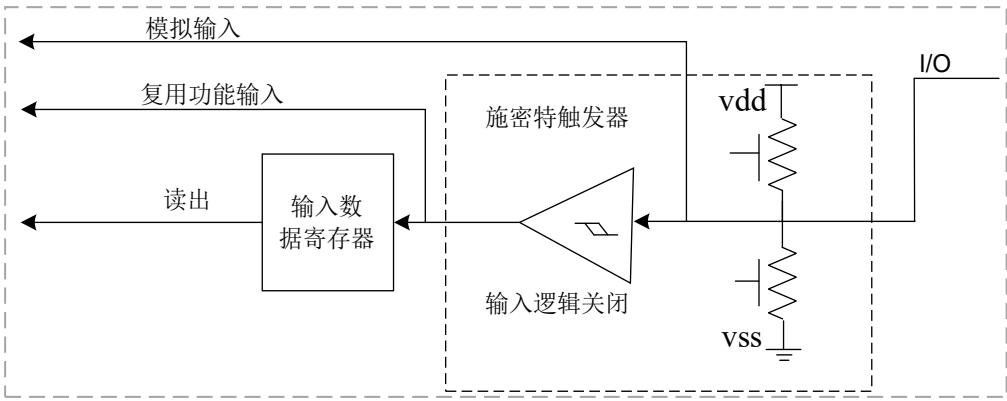


图 10-5 模拟输入

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

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_AFR1	端口复用功能低位寄存器	0x00000000

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

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

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） 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
27:26	CNF6	
23:22	CNF5	
19:18	CNF4	
15:14	CNF3	
11:10	CNF2	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
7:6	CNF1	
3:2	CNF0	
29:28	MODE7	
25:24	MODE6	
21:20	MODE5	端口输入输出配置（MODEy）（y = 0..7） 软件配置相应的 I/O 端口；参考端口位配置表 配置 MODEy 不等于 0 时，不同配置输出速度不同： 00：输入模式；01：低速 10：中速；11：高速
17:16	MODE4	
13:12	MODE3	
9:8	MODE2	
5:4	MODE1	
1:0	MODE0	

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			
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) 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
27:26	CNF14	
23:22	CNF13	
19:18	CNF12	
15:14	CNF11	
11:10	CNF10	
7:6	CNF9	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
3:2	CNF8	
29:28	MODE15	
25:24	MODE14	端口输入输出配置 (MODEy) (y = 15..8) 软件配置相应的 I/O 端口; 参考端口位配置表 配置 MODEy 不等于 0 时, 不同配置输出速度不同: 00: 输入模式; 01: 低速 10: 中速; 11: 高速
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=15~0)															
r															

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

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..F) 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 000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRy (y=15~0)															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSy (y=15~0)															
w															
Bit	Field		Description												
31:16	BRy		端口清除位 y (y=15..0) 写 0 相应的 ODRY 位保持不变 写 1 清除对应的 ODRY 位为 0												
15:0	BSy		端口置位 y (y=15..0) 写 0 相应的 ODRY 位保持不变 写 1 置位对应的 ODRY 位为 1 注：同时写 BSy 位与 BRy 位为 1 时，BSy 的优先级高于 BRy												

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
															LCKK
															rw
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_AFR1 端口复用功能低位寄存器

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
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: 0x0000 0000

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

Bit	Field	Description
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 DMA1/2 直接存储器访问控制器

11.1 DMA 简介

DMA 控制器通过共享系统总线，实现无需 CPU 参与的快速自动数据传输。

DMA1、DMA2 具有相同的功能，DMA1 有 8 个通道，DMA2 有 8 个通道，本章节以 DMA1 表述。

11.2 DMA 功能框图

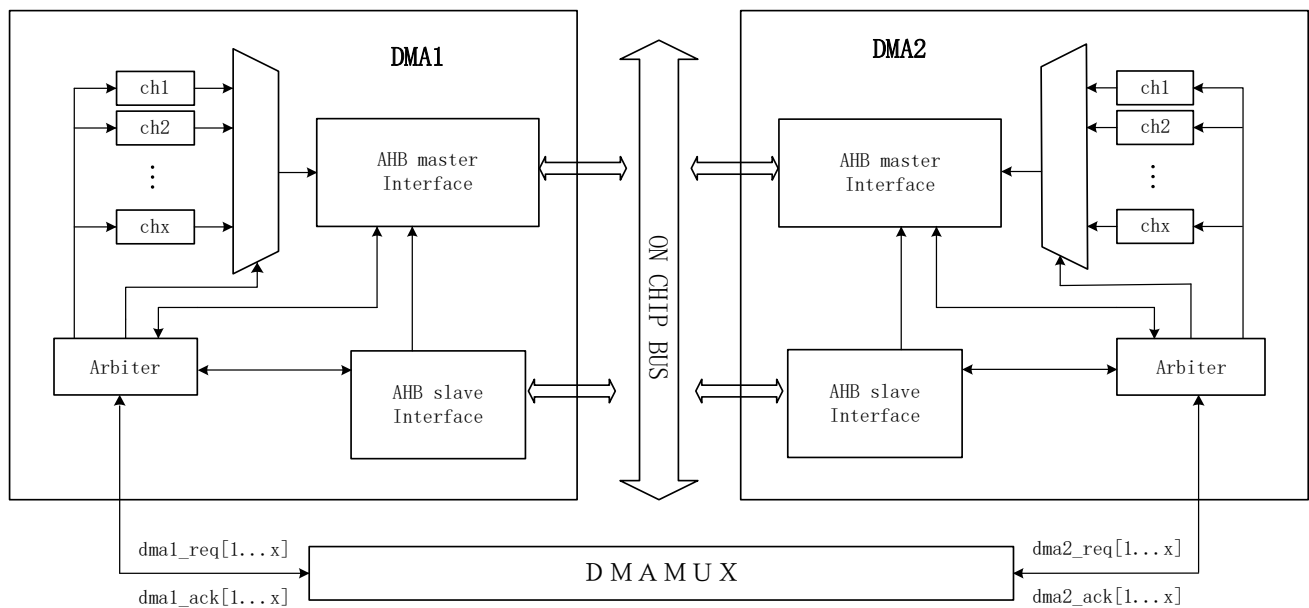


图 11-1 DMA 功能框图

11.3 DMA 主要特征

- 8 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 8 个通道请求之间的处理优先级（共有四级：很高、高、中等和低），若优先级相同，则由硬件自动决定处理顺序（低编号通道请求优先处理）。
- 数据源头与目的地的传输宽度可独立配置为字节、半字、全字。
- 按照数据源的宽度配置进行打包，并在目的地按照目的地的宽度配置进行拆包。要求源和目标地址必须根据各自配置的数据传输宽度对齐。
- 支持循环缓冲器控制。

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

11.4 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 11-1 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

11.5 DMA

11.5.1 DMA 请求映像

从外设产生的多个传输请求，通过 DMAMAP 输入到 DMA 控制器，为了避免冲突，在一个通道中，同时只能有一个外设 DMA 请求有效。具体的 DMA 请求映射参见芯片特定配置文档。

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

11.6 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。

11.6.1 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。

11.6.2 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。

11.6.3 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。

11.6.3.1 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。

11.6.3.2 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。

11.6.3.3 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）。

11.6.3.4 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。

11.6.3.5 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。

11.6.4 可编程的数据传输宽度，对齐方式和数据大小端

当 PSIZE 和 MSIZE 不相同，DMA 模块按照下表进行数据对齐。

源数据比特位对齐写入目标地址，

若目标数据传输宽度大于源数据传输宽度，则目标数据宽度多余位补 0 处理。

若目标数据传输宽度小于源数据传输宽度，则源数据宽度多余部分截断处理。

表 11-2 可配置的数据传输宽度和大小端操作（当 PINC = MINC = 1），传输数目为 4

组合类型	传输宽度		传输操作	
	源端	目标	源（地址 数据）	目标（地址 数据）
源端传输宽度 等于 目标传输宽度 地址步长、数据 宽度一致	8	8	在 0x0 读 B0[7:0] 在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x0 写 B0[7:0] 在 0x1 写 B1[7:0] 在 0x2 写 B2[7:0] 在 0x3 写 B3[7:0]
	16	16	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 B1B0[15:0] 在 0x2 写 B3B2[15:0] 在 0x4 写 B5B4[15:0] 在 0x6 写 B7B6[15:0]
	32	32	在 0x0 读 B3B2B1B0[31:0] 在 0x4 读 B7B6B5B4[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B3B2B1B0[31:0] 在 0x4 写 B7B6B5B4[31:0] 在 0x8 写 BBBAB9B8[31:0] 在 0xC 写 BFBEBDBC[31:0]
源端传输宽度 小于 目标传输宽度 地址步长为传输 宽度 ÷ 8 目标数据多余 bit 位补 0	8	16	在 0x0 读 B0[7:0] 在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x0 写 00B0[15:0] 在 0x2 写 00B1[15:0] 在 0x4 写 00B2[15:0] 在 0x6 写 00B3[15:0]
	8	32	在 0x0 读 B0[7:0] 在 0x1 读 B1[7:0] 在 0x2 读 B2[7:0] 在 0x3 读 B3[7:0]	在 0x0 写 000000B0[31:0] 在 0x4 写 000000B1[31:0] 在 0x8 写 000000B2[31:0] 在 0xC 写 000000B3[31:0]
	16	32	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 0000B1B0[31:0] 在 0x4 写 0000B3B2[31:0] 在 0x8 写 0000B5B4[31:0] 在 0xC 写 0000B7B6[31:0]
源端传输宽度 大于 目标传输宽度 地址步长为传输 宽度 ÷ 8 目标数据不足 bit 位截断	16	8	在 0x0 读 B1B0[15:0] 在 0x2 读 B3B2[15:0] 在 0x4 读 B5B4[15:0] 在 0x6 读 B7B6[15:0]	在 0x0 写 B0[7:0] 在 0x1 写 B2[7:0] 在 0x2 写 B4[7:0] 在 0x3 写 B6[7:0]
	32	8	在 0x0 读 B3B2B1B0[31:0] 在 0x4 读 B7B6B5B4[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B0[7:0] 在 0x1 写 B4[7:0] 在 0x2 写 B8[7:0] 在 0x3 写 BC[7:0]
	32	16	在 0x0 读 B3B2B1B0[31:0] 在 0x4 读 B7B6B5B4[31:0] 在 0x8 读 BBBAB9B8[31:0] 在 0xC 读 BFBEBDBC[31:0]	在 0x0 写 B1B0[15:0] 在 0x2 写 B5B4[15:0] 在 0x4 写 B9B8[15:0] 在 0x6 写 BDBC[15:0]

11.6.4.1 操作一个不支持字节或半字写的 AHB 设备

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

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

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

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

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

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

11.6.5 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。

11.7 DMA 寄存器描述

表 11-3 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+0x14*(x-1)	DMA_CCRx	DMA 通道 x 配置寄存器	0x00000000
0x0C+0x14*(x-1)	DMA_CNDTRx	DMA 通道 x 传输数量寄存器	0x00000000
0x10+0x14*(x-1)	DMA_CPARx	DMA 通道 x 外设地址寄存器	0x00000000
0x14+0x14*(x-1)	DMA_CMARx	DMA 通道 x 存储器地址寄存器	0x00000000

11.7.1 DMA_ISR DMA 中断状态寄存器

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
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
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
31,27,23,19,15 ,11,7,3	TEIFx	通道 x 的传输错误标志 (x=1~8) (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)
30,26,22,18,14 ,10,6,2	HTIFx	通道 x 的半传输标志 (x=1~8) (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)
29,25,21,17,13 ,9,5,1	TCIFx	通道 x 的传输完成标志 (x=1~8) (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)
28,24,20,16,12 ,8,4,0	GIFx	通道 x 的全局中断标志 (x=1~8) (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

11.7.2 DMA_IFCR DMA 中断标志清除寄存器

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31,27,23,19,15 ,11,7,3	CTEIFx	清除通道 x 的传输错误标志 (x=1~8) (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
30,26,22,18,14 ,10,6,2	CHTIFx	清除通道 x 的半传输标志 (x=1~8) (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
29,25,21,17,13 ,9,5,1	CTCIFx	清除通道 x 的传输完成标志 (x=1~8) (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
28,24,20,16,12 ,8,4,0	CGIFx	清除通道 x 的全局中断标志 (x=1~8) (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

11.7.3 DMA_CCRx DMA 通道 x 配置寄存器 (x=1~8)

偏移地址: 0x08+0x14* (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.															BURSTEN
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	rW	rW	rW

Bit	Field	Description
31:17	Reserved	保留, 始终读为 0
16	BURSTEN	突发传输使能 (Burst transfer enable) 该位由软件置'1'或清零。 0:关闭突发传输 1:使能突发传输 Flexcan dma 传输一定要使能突发传输, 其他模块 dma 传输一定要关闭突发传输

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

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

11.7.4 DMA_CNDTRx DMA 通道 x 传输数量寄存器 (x=1~8)

偏移地址: 0x0C+0x14* (通道编号-1)

复位值: 0x0000 0000

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

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

11.7.5 DMA_CPARx DMA 通道 x 外设地址寄存器 (x=1~8)

偏移地址: 0x10+0x14* (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。

11.7.6 DMA_CMARx DMA 通道 x 存储器地址寄存器（x = 1~8）

偏移地址：0x14+0x14*（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。

12 MDS 可编程 IP 互联模块

12.1 介绍

MDS (MindSwitch) 的作用是在不借助软件干预的情况下使几个不同的外设协同工作。它连接了若干个源模块 (比如 TIMx, EXTIx 等) 和目标模块 (比如 ADCx, DACx 等)。通过寄存器配置可以选通任意一个源和目标之间的连接通道, 使目标模块可以成功接收到源信号并自动开始预设的动作。

12.2 主要特性

- 支持触发通道和连接通道
- 可配置触发源和触发目标
- ◆ 触发源和触发目标可自由配置
- ◆ 支持外设硬件触发和软件触发
- ◆ 内部集成四个组合逻辑单元 CLU(Combinational Logic Unit), 提供 or-or 组合逻辑操作。
- 可配置连接源和连接目标
- ◆ 连接源和连接目标可自由配置
- ◆ 内部集成四个组合逻辑单元 CLU(Combinational Logic Unit), 提供多种组合逻辑操作。
- 支持不同时钟频率的触发源和触发目标之间协同工作

12.3 功能描述

12.3.1 功能框图

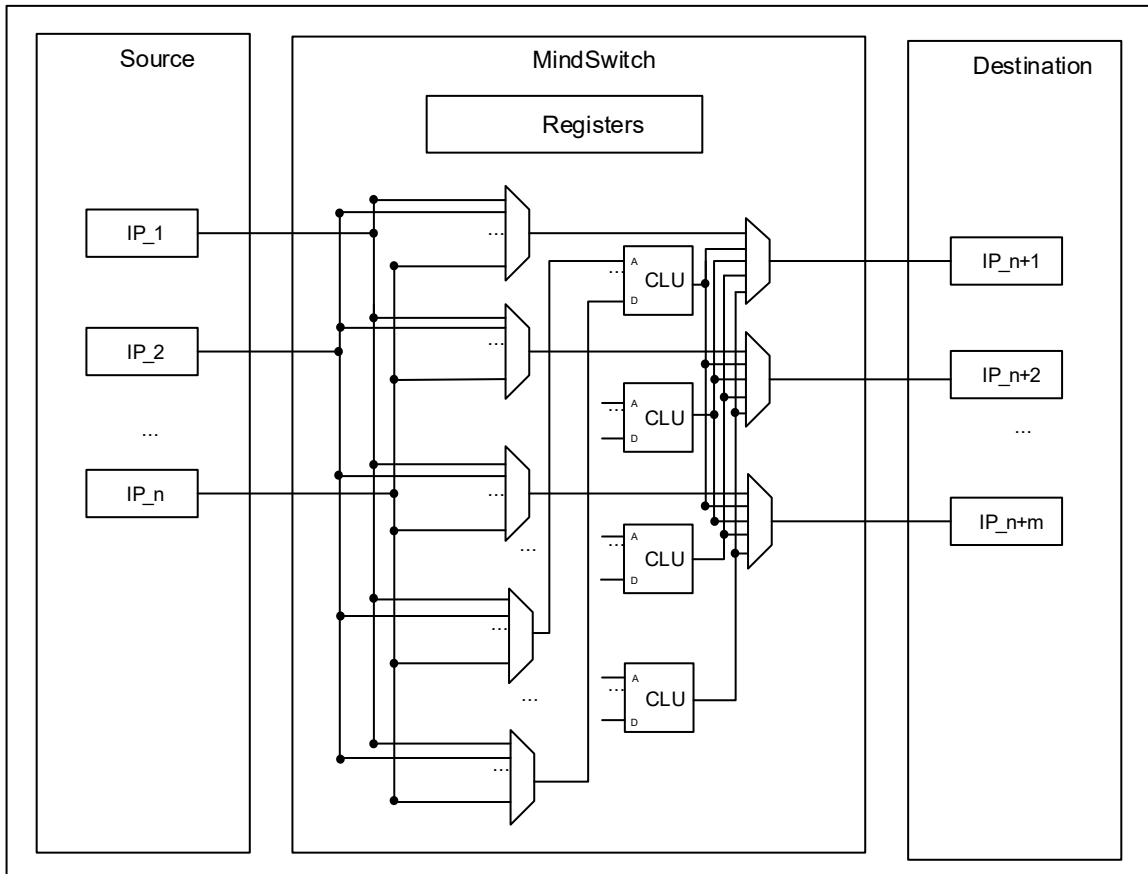


图 12-1 MindSwitch 功能框图

12.3.2 时钟复位

MindSwitch 通过 32-bit APB slave 接口配置寄存器。在配置 MindSwitch 之前，要先使能 RCC 控制器中的时钟使能控制位。

配置 RCC 控制器中的软件复位请求可进行软件复位操作。

12.3.3 功能概述

每个触发目标和连接目标通道有一个通道控制寄存器（MDS_TRIGxCR 和 MDS_CONNxCR），向 CHNSEL 写入触发源对应事件编号即选择对应的事件作为该触发目标或连接目标的源。

触发目标可以选择 Logic 0、Logic 1、软件触发或者其他硬件输出作为触发源。当选择软件触发方式时，通过向 MDS_SWTRIG.SW_TRIG 写 1 来产生一次软件触发。软件触发寄存器为所有触发通道公用，

包括 CLU 输入通道。

连接目标可以选择 Logic0、Logic1 或者其他硬件输出作为源。

配置 CLUEN 为 1 时，触发目标将选择 CLUSEL 对应的 CLU 输出作为触发源。每个 CLU 单元有四个输入，可以通过配置 CLUINx_SEL 分别选择每个 CLU 输入的来源。连接通道 CLU 提供多种组合逻辑操作，可通过配置 MDS_CONN_CLUxCFG.CLMODE 寄存器选择不同的逻辑操作，具体逻辑操作如下图所示。触发通道 CLU 只提供 OR-OR 功能。

当目标不使用 CLU 输出作为源时，需要配置其专有通道的 CLUEN 位为 0，以防止误触发和节约公共 CLU 资源。使用 CLU 模块时，建议先配置 CLUxSEL, CLUxCFG 寄存器，最后使能 CLUEN，以防止配置过程中产生不必要的翻转影响目标的工作。

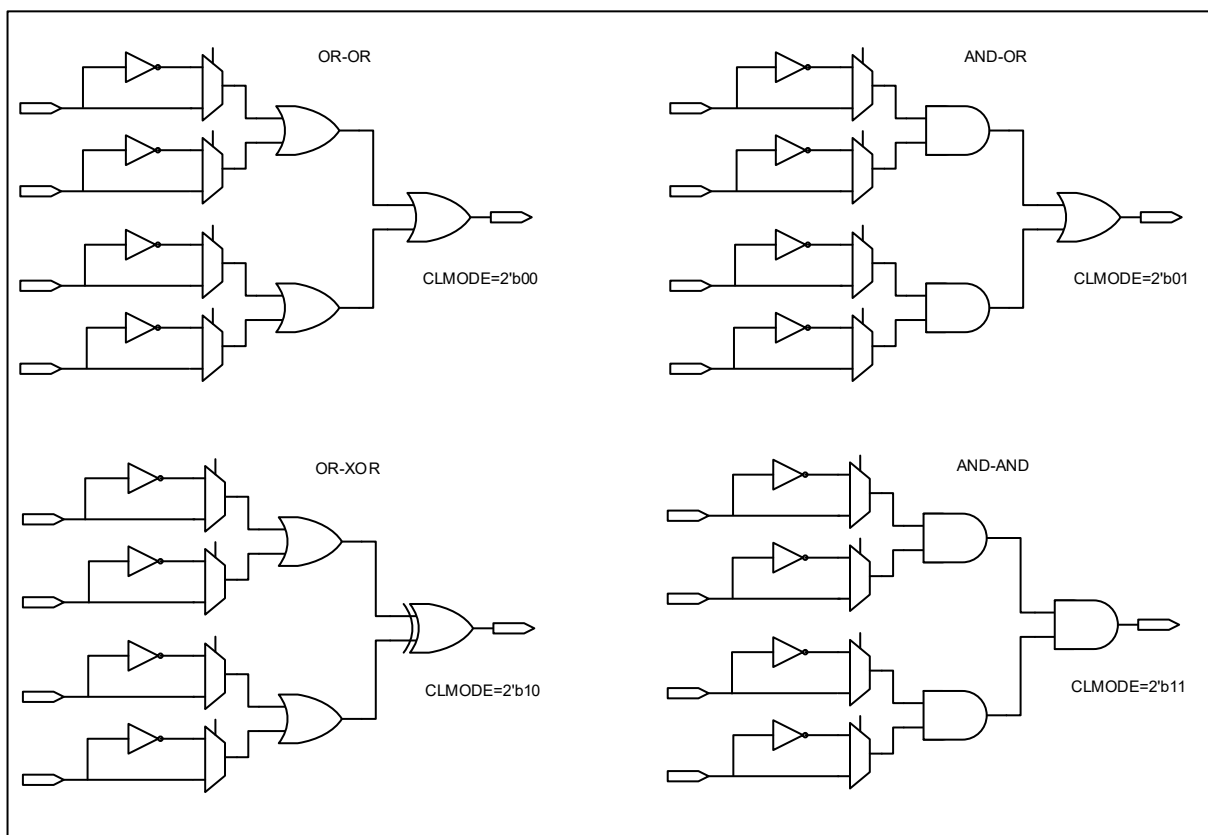


图 12-2 连接通道组合逻辑操作选项

12.3.4 触发源和触发目标

触发目标分为触发（trigger）和连接（connection）两种，MindSwitch 会与触发类型的目标做握手，以确保触发源能被触发目标正确接收，如 ADC/DAC 等的外部触发。对于连接类型的目标，MindSwitch 会将选择过后的触发源与目标直接连接，如比较器输出窗口和 GPIO 等。

12.3.4.1 触发源和触发目标

MindSwitch 支持不同种类的触发源，包括且不限于：

- 低电平
- 高电平
- 软件触发
- TIMx 触发输出
- TIMx 通道捕获和比较事件
- 外部事件 EXTIX

触发目标包括且不限于：

- ADCx 外部触发
- ADCx 注入模式外部触发
- DACx 外部触发

具体映射表详见 MindSwitch 触发源映射表和 MindSwitch 触发目标映射表。

12.3.4.2 连接源和连接目标

MindSwitch 支持不同种类的连接源，包括且不限于：

- 低电平
- 高电平
- TIMx 通道输出
- 外部输入 MDS_INx

连接目标包括且不限于：

- 比较器输出窗口 COMPx_blanking
- GPIO 输出

具体映射表详见 MindSwitch 连接源映射表和连接目标映射表。

12.4 寄存器

12.4.1 寄存器总览

表 12-1 MindSwitch 寄存器概览

Offset	Acronym	Register Name	Reset
0x000, 0x004, 0x008 ...	MDS_TRIGxCR (x=0, 1, 2 ...)	触发通道 x 控制寄存器	0x00000000
0x080, 0x084 0x088, 0x08C	MDS_TRIG_CLUxSEL (x=0, 1, 2, 3)	触发通道 CLUx 逻辑运算输入选择寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0xC0, 0xC4 0xC8, 0xCC	MDS_TRIG_CLUxCFG (x=0, 1, 2, 3)	触发通道 CLUx 逻辑运算配置寄存器	0x00000000
0x0FC	MDS_SWTRIG	软件触发控制寄存器	0x00000000
0x100, 0x104, 0x108 ...	MDS_CONNxCR (x=0, 1, 2...)	连接通道 x 控制寄存器	0x00000000
0x180, 0x184 0x188, 0x18C	MDS_CONN_CLUxSEL (x=0, 1, 2, 3)	连接通道 CLUx 逻辑运算输入选择寄存器	0x00000000
0x1C0, 0x1C4 0x1C8, 0x1CC	MDS_CONN_CLUxCFG (x=0, 1, 2, 3)	连接通道 CLUx 逻辑运算配置寄存器	0x00000000

12.4.2 MDS_TRIGxCR 触发通道 x 控制和状态寄存器

偏移地址：0x00, 0x04, 0x08...

复位值：0x0000 0000

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

Bit	Field	Description
31:26	Reserved	保留，必须保持复位值
25:24	EDGESEL	沿检测选择寄存器（Edge Select Register） 11：上升沿检测 10：上升沿或下降沿检测 01：下降沿检测 00：上升沿检测
23:19	Reserved	保留，必须保持复位值
18:17	CLUSEL	CLU 选择寄存器（CLU Select Register） 11：选择 CLU3 的输出作为触发源 10：选择 CLU2 的输出作为触发源 01：选择 CLU1 的输出作为触发源 00：选择 CLU0 的输出作为触发源 选择 CLU 的输出，该 bit 仅在 CLUEN 置为 1 的时候起作用。
16	CLUEN	CLU 使能寄存器（CLU Enable Register） 1：选择 CLU 输出作为触发源 0：选择通道专用多路选择器输出作为触发源
15:6	Reserved	保留，必须保持复位值

Bit	Field	Description
5:0	TRGSEL	通道触发源选择寄存器（Trigger Source Select Register） 该寄存器仅在 CLUEN 置为 0 的时候起作用，控制通道触发源的选择，具体编号见 MindSwitch 触发源映射列表 当 TRGSEL 配置超出映射列表范围时，该通道将选择第 0 号输入作为触发源

12.4.3 MDS_TRIG_CLUxSEL 触发通道 CLUx 逻辑运算输入选择寄存器

偏移地址：0x80, 0x84, 0x88, 0x8C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		CLUIN3_SEL						Res.		CLUIN2_SEL					
		rw								rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CLUIN1_SEL						Res.		CLUIN0_SEL					
		rw								rw					

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29:24	CLUIN3_SEL	CLU 输入通道 3 选择寄存器（CLU Input3 Select Register） 控制 CLU 输入通道 3 触发源的选择，具体编号见 MindSwitch 触发源映射列表
23:22	Reserved	保留，必须保持复位值
21:16	CLUIN2_SEL	CLU 输入通道 2 选择寄存器（CLU Input2 Select Register） 控制 CLU 输入通道 2 触发源的选择，具体编号见 MindSwitch 触发源映射列表
15:14	Reserved	保留，必须保持复位值
13:8	CLUIN1_SEL	CLU 输入通道 1 选择寄存器（CLU Input1 Select Register） 控制 CLU 输入通道 1 触发源的选择，具体编号见 MindSwitch 触发源映射列表
7:6	Reserved	保留，必须保持复位值
5:0	CLUIN0_SEL	CLU 输入通道 0 选择寄存器（CLU Input0 Select Register） 控制 CLU 输入通道 0 触发源的选择，具体编号见 MindSwitch 触发源映射列表

12.4.4 MDS_TRIG_CLUxCFG 触发通道 CLUx 逻辑运算配置寄存器

偏移地址：0xC0, 0xC4, 0xC8, 0xCC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						CLUIN3_ED		Res.						CLUIN2_ED	
						rw								rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CLUIN1_ED		Res.						CLUIN0_ED	

	rw		rw

Bit	Field	Description
31:26	Reserved	保留，必须保持复位值
25:24	CLUIN3_ED	CLU 输入通道 3 沿检测模式选择（CLUIN3 Edge Detect Mode） 11：上升沿检测 10：上升沿和下降沿检测 01：下降沿检测 00：上升沿检测
23:18	Reserved	保留，必须保持复位值
17:16	CLUIN2_ED	CLU 输入通道 2 沿检测模式选择（CLUIN2 Edge Detect Mode） 11：上升沿检测 10：上升沿和下降沿检测 01：下降沿检测 00：上升沿检测
15:10	Reserved	保留，必须保持复位值
9:8	CLUIN1_ED	CLU 输入通道 1 沿检测模式选择（CLUIN1 Edge Detect Mode） 11：上升沿检测 10：上升沿和下降沿检测 01：下降沿检测 00：上升沿检测
7:2	Reserved	保留，必须保持复位值
1:0	CLUIN0_ED	CLU 输入通道 0 沿检测模式选择（CLUIN0 Edge Detect Mode） 11：上升沿检测 10：上升沿和下降沿检测 01：下降沿检测 00：上升沿检测

12.4.5 MDS_SWTRIG 软件触发寄存器

偏移地址：0xFC

复位值：0x0000 0000

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

Bit	Field	Description
31:1	Reserved	保留，必须保持复位值
0	SW_TRIG	软件触发寄存器（Software trigger register） 1：产生一次软件触发 0：无操作 该位仅在 TRGSEL 选择软件触发时起作用，写 1 产生一次软件触发，写 0 无操作

12.4.6 MDS_CONNxCR 连接通道 x 控制和状态寄存器

偏移地址：0x100, 0x104, 0x108...

复位值：0x0000 0000

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

Bit	Field	Description
31:19	Reserved	保留，必须保持复位值
18:17	CLUSEL	CLU 选择寄存器（CLU Select Register） 11：选择 CLU3 的输出作为触发源 10：选择 CLU2 的输出作为触发源 01：选择 CLU1 的输出作为触发源 00：选择 CLU0 的输出作为触发源 选择 CLU 的输出，该 bit 仅在 CLUEN 置为 1 的时候起作用。
16	CLUEN	CLU 使能寄存器（CLU Enable Register） 1：选择 CLU 输出作为触发源 0：选择通道专用多路选择器输出作为触发源
15:6	Reserved	保留，必须保持复位值
5:0	TRGSEL	通道触发源选择寄存器（Trigger Source Select Register） 该寄存器仅在 CLUEN 置为 0 的时候起作用，控制通道触发源的选择，具体编号见 MindSwitch 连接源映射列表 当 TRGSEL 配置超出映射列表范围时，该通道将选择第 0 号输入作为连接源

12.4.7 MDS_CONN_CLUxSEL 连接通道 CLUx 逻辑运算输入选择寄存器

偏移地址：0x180, 0x184, 0x188, 0x18C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		CLUIN3_SEL						Res.		CLUIN2_SEL					
		rw								rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CLUIN1_SEL						Res.		CLUIN0_SEL					
		rw								rw					

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29:24	CLUIN3_SEL	CLU 输入通道 3 选择寄存器（CLU Input3 Select Register） 控制 CLU 输入通道 3 触发源的选择，具体编号见 MindSwitch 连接源映射列表
23:22	Reserved	保留，必须保持复位值
21:16	CLUIN2_SEL	CLU 输入通道 2 选择寄存器（CLU Input2 Select Register） 控制 CLU 输入通道 2 触发源的选择，具体编号见 MindSwitch 连接源映射列表
15:14	Reserved	保留，必须保持复位值
13:8	CLUIN1_SEL	CLU 输入通道 1 选择寄存器（CLU Input1 Select Register） 控制 CLU 输入通道 1 触发源的选择，具体编号见 MindSwitch 连接源映射列表
7:6	Reserved	保留，必须保持复位值
5:0	CLUIN0_SEL	CLU 输入通道 0 选择寄存器（CLU Input0 Select Register） 控制 CLU 输入通道 0 触发源的选择，具体编号见 MindSwitch 连接源映射列表

12.4.8 MDS_CONN_CLUXCFG 连接通道 CLUX 逻辑运算配置寄存器

偏移地址：0x1C0, 0x1C4, 0x1C8, 0x1CC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				INV3	INV2	INV1	INV0	Res.						CLMODE	
				rw	rw	rw	rw							rw	

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11	INV3	CLU INPUT3 取反（CLU Input3 Invert） 1：将 CLU INPUT3 取反作为组合逻辑运算操作的输入 0：将 CLU INPUT3 作为组合逻辑运算操作的输入
10	INV2	CLU INPUT2 取反（CLU Input2 Invert） 1：将 CLU INPUT2 取反作为组合逻辑运算操作的输入 0：将 CLU INPUT2 作为组合逻辑运算操作的输入

Bit	Field	Description
9	INV1	CLU INPUT1 取反 (CLU Input1 Invert) 1: 将 CLU INPUT1 取反作为组合逻辑运算操作的输入 0: 将 CLU INPUT1 作为组合逻辑运算操作的输入
8	INV0	CLU INPUT0 取反 (CLU Input0 Invert) 1: 将 CLU INPUT0 取反作为组合逻辑运算操作的输入 0: 将 CLU INPUT0 作为组合逻辑运算操作的输入
7:2	Reserved	保留, 必须保持复位值
1:0	CLMODE	CLU 运算模式选择 (CLU Mode Select) 11: $output = in0 \& in1 \& in2 \& in3$ 10: $output = (in0 \mid in1) \wedge (in2 \mid in3)$ 01: $output = (in0 \& in1) \mid (in2 \& in3)$ 00: $output = in0 \mid in1 \mid in2 \mid in3$

13 CRC_POLY 循环冗余校验计算单元（多项式可配）

13.1 介绍

CRC_POLY 计算单元利用可配置的多项式，形成不同的 CRC 算法，计算 8 位、16 位和 32 位数据的 CRC 校验码，用于对数据传输或数据存储的完整性进行验证。

13.2 主要特性

- 支持多项式可配以及 8、16、32 位多项式宽度可选
- 支持所有结果异或值为 0 的 8、16、32 位多项式位宽的算法
- 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
- CRC_POLY 计算时间：32 位宽的数据需要 4 个 HCLK，16 位宽的数据需要 2 个 HCLK，8 位宽的数据需要 1 个 HCLK
- 32 位独立数据寄存器，用于存放临时数据
- 32 位宽初值寄存器，用于存放 CRC_POLY 的初始值
- 支持输入、输出数据反序以及大小端配置

13.3 功能描述

13.3.1 功能框图

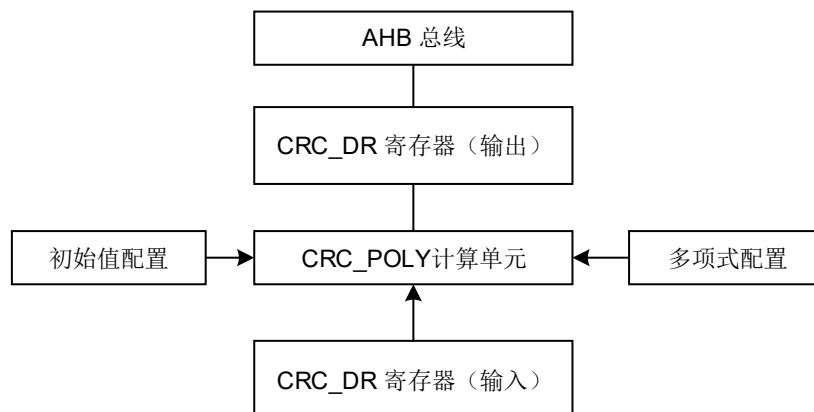


图 13-1 CRC_POLY 功能框图

13.3.2 功能概述

- CRC_POLY 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_POLY 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。
- 配置 CRC 控制寄存器（CRC_CR）寄存器 PWIDTH 位，选择 CRC 多项式位宽；根据多项式的位宽配置 CRC 多项式寄存器（CRC_PR），选择需要使用的 CRC 多项式。
- 可以配置 CRC 控制寄存器（CRC_CR）的 RI、RO 位，设置输入反序，输出反序；配置 BEI 和 BEO，设置输入数据大小端，输出数据大小端。
- 通过置位 CRC 控制寄存器（CRC_CR）中的 RST 位，可以将 CRC 初始值寄存器（CRC_IVR）的值写入 CRC 数据寄存器（CRC_DR）中。
- 提供 32 位 CRC 独立数据寄存器（CRC_IDR），作为缓存寄存器，任何时候都可以对其进行读写操作。

13.3.3 使用方法

13.3.3.1 CRC 计算操作步骤

- 使能 CRC 模块时钟；
- 复位 CRC 模块；
- 配置 CRC_CR 寄存器：
 - ◆ 选择多项式位宽：8 位、16 位或者 32 位；
 - ◆ 选择输入、输出的反序；
 - ◆ 选择输入、输出的大小端。
- 配置 CRC 初值寄存器（CRC_IVR），设置计算初值；
- 通过配置 CRC 控制寄存器（CRC_CR）的 RST 位，可以将 CRC_IVR 寄存器的值写入 CRC_DR 寄存器；
- 配置 CRC 多项式寄存器（CRC_PR），设置 CRC 多项式；
- 将待测数据依次写入 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。

13.4 寄存器

13.4.1 寄存器总览

表 13-1 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_IVR	CRC 初值寄存器	0xFFFFFFFF
0x10	CRC_PR	CRC 多项式寄存器	0x04C11DB7

13.4.2 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) 支持 8、16、32 位宽数据访问； 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_POLY 计算； 读取时，作为输出寄存器，返回 CRC_POLY 计算的结果； 注：位宽设置无需配置寄存器位，需将读写指针设置为 8、16、32 位；当 CRC_CR->RST 位被置位时，该寄存器被设为 CRC_IVR 寄存器的值。

13.4.3 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
IDR															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR															

rw

Bit	Field	Description
31: 0	IDR	32 位通用数据寄存器（General-purpose 32-bit Data Register） 临时存放 4 字节的数据空间。 不受 CRC 控制寄存器（CRC_CR）的 RST 位影响。 注：此寄存器不参与 CRC 计算，用于存放数据。

13.4.4 CRC_CR CRC 控制寄存器

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWIDTH		Reserved							RO	BEO	BEI	Reserved		RI	RST
rw									rw	rw	rw			rw	w

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 14	PWIDTH	多项式位宽（Polynomial Width） 00: 32 位 01: 16 位 1x: 8 位
13: 7	Reserved	保留，必须保持复位值
6	RO	输出数据反序（Reverse Output） 0: 输出数据不反序 1: 输出数据反序
5	BEO	输出大小端选择（Big Endian Output） 0: 小端（Little-endian） 1: 大端（Big-endian）
4	BEI	输入大小端选择（Big Endian Input） 0: 小端（Little-endian） 1: 大端（Big-endian）
3: 2	Reserved	保留，必须保持复位值
1	RI	输入数据反序（Reverse Input） 0: 输入数据不反序 1: 输入数据反序

Bit	Field	Description
0	RST	复位 CRC 计算单元（CRC Reset） 将初始值寄存器的值载入 DR 寄存器。 该位只能写'1'，同时硬件自动实现清'0'。

13.4.5 CRC_IVR CRC 初值寄存器

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

Bit	Field	Description
31: 0	IVR	初始值寄存器（Initial Value Register） 用于设置数据寄存器（CRC_DR）初始值。当多项式为 8 位宽时，只有低 8 位有效；多项式为 16 位宽时，只有低 16 位有效 注：不受 CRC 控制寄存器（CRC_CR）的 RST 位影响。

13.4.6 CRC_PR CRC 多项式寄存器

偏移地址：0x10

复位值：0x04C1 1DB7

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

Bit	Field	Description
31: 0	PR	多项式寄存器（Polynomial Register） 用于配置 CRC 多项式 注：不受 CRC 控制寄存器（CRC_CR）的 RST 位影响。

14 CORDIC 坐标旋转算法

14.1 介绍

本文为 CORDIC IP 设计说明，根据电机应用需求进行细化，设计能直接运算 $\sin$ 、 $\cos$ 、 $\tan$ 函数，间接运算 $\tan$ 功能的 CORDIC IP，迭代深度 22，数据支持 Q24 及 Q15 格式。

14.2 功能描述

14.2.1 功能概述

- 直接运算 $\sin$ 、 $\cos$ 、 $\tan$ 函数
- 圆向量模式，支持满范围 $[-2^{23}, 2^{23}-1]$ 内的初始 X,Y 数据值，进行计算 $\tan$ 和幅值
- 圆旋转模式，支持满范围 $[-2^{23}, 2^{23}-1]$ 内的初始 Z 角度范围，代表范围 $[-\pi, ((2^{23}-1)/2^{23})\pi]$ 内的角度，进行计算 $\sin$ 、 $\cos$ 三角函数值
- 每次运算支持 22 次迭代，迭代角度精度达到 0.00002732 度
- 24 位可访问的数据宽度，支持电机中典型 Q24 及 Q15 应用
 - ◆ 利用 Q 格式进行浮点数据到定点的相互转化。据转化为 Q15，将数据乘以 2^{15} ；Q15 数据转化为浮点数据，将数据除以 2^{15} ，Q24 同理
 - ◆ Q15 表示小数部分有 15 位；Q24 表示小数部分有 24 位
- 接口支持标准的寄存器接口读写时序
- 支持两种启动计算的工作模式，DX 数据写完后直接计算的工作模式、启动位(START)设置后计算的工作模式

14.3 atan 计算的配置实例

以计算 Q24 数据为例，DX= 0x012345，DY=0x067890，求对应 $\tan$ 角度 RZ

配置执行的流程如下：

- 配置 CORDIC_CR 寄存器，写入 0x00000008：选择向量模式、模块使能、写入 CORDIC_DXR 寄存器后立即运算的启动方式
- 配置 CORDIC_DYR 寄存器，将 DY 值写入
- 配置 CORDIC_DZR 寄存器，角度初始值设为“0”，写入 0x0000000
- 查询 CORDIC_SR 寄存器，看是否在计算过程中，BSY 为“0”表示空闲，可以进行计算
- 配置 CORDIC_DXR 寄存器，将 DX 值写入，写入后的下一个时钟周期进入计算状态

- 此时 CORDIC 模块正在进行计算操作
- 查询 CORDIC_SR 寄存器 BSY 位，等待计算完毕
- 读取 CORDIC_RZR 寄存器，此时 CORDIC_RZR 寄存器的数据即为 atan 计算结果

14.4 sin、cos 计算的配置实例

以计算 Q24 数据为例，DZ= 0x012345，求对应 sin、cos 三角函数值

配置执行的流程如下：

- 配置 CORDIC_CR 寄存器，写入 0x0000000E：选择旋转模式、模块使能、在 START 寄存器位置位为“1”时进行计算
- 配置 CORDIC_DXR 寄存器，写入 0x9B74ED (计算 Q15 时 DX 写入 0x4DBA)
- 配置 CORDIC_DZR 寄存器，写入 0x012345
- 配置 CORDIC_DYR 寄存器，写入 0x000000
- 查询 CORDIC_SR 寄存器，看是否在计算过程中，BSY 为“0”表示空闲，可以进行计算
- 配置 CORDIC_CR 寄存器，写入 0x0000000F，将 START 位置“1”，进入计算状态
- 查询 CORDIC_SR 寄存器 BSY 位，等待计算完毕
- 读取 CORDIC_RYR 寄存器，此时 CORDIC_RYR 寄存器的数据即为 sin 计算结果；读取 CORDIC_RXR 寄存器，此时 CORDIC_RXR 寄存器的数据即为 cos 计算结果

14.5 寄存器

14.5.1 寄存器总览

表 14-1 CORDIC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CORDIC_DXR	CORDIC X 数据寄存器	0x00000000
0x04	CORDIC_DYR	CORDIC Y 数据寄存器	0x00000000
0x08	CORDIC_DZR	CORDIC Z 数据寄存器	0x00000000
0x0C	CORDIC_RXR	CORDIC X 结果寄存器	0x00000000
0x10	CORDIC_RYR	CORDIC Y 结果寄存器	0x00000000
0x14	CORDIC_RZR	CORDIC Z 结果寄存器	0x00000000
0x18	CORDIC_CR	CORDIC 控制寄存器	0x00000000
0x1C	CORDIC_SR	CORDIC 状态寄存器	0x00000000

14.5.2 CORDIC_DXR CORDIC X 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.								DX							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DX															
rw															
Bit	Field							Description							
31:24	Reserved							保留，必须保持复位值							
23:0	DX							初始 X 参数数据（满范围 $[-2^{23}, 2^{23}-1]$ 内的初始 X 数据值）							

14.5.3 CORDIC_DYR CORDIC Y 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.								DY							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DY															
rw															
Bit	Field							Description							
31:24	Reserved							保留，必须保持复位值							
23:0	DY							初始 Y 参数数据（满范围 $[-2^{23}, 2^{23}-1]$ 内的初始 Y 数据值）							

14.5.4 CORDIC_DZR CORDIC Z 数据寄存器

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

rw

Bit	Field	Description
31:24	Reserved	保留，必须保持复位值
23:0	DZ	初始 Z 参数数据（满范围 $[-2^{23}, 2^{23}-1]$ 内的初始 Z 角度范围）

14.5.5 CORDIC_RXR CORDIC X 结果寄存器

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

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	RX	X 计算结果（cos 计算结果） 注 1：最高位[24]代表正负，最高位为"1"代表负；最高位为"0"代表正。 注 2：圆向量模式时，可以通过计算 RX/A_n 得到幅值（ A_n 为 1.646760258）。

14.5.6 CORDIC_RYR CORDIC Y 结果寄存器

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

Bit	Field	Description
31:25	Reserved	保留，必须保持复位值
24:0	RY	Y 计算结果（sin 计算结果） 注：最高位[24]代表正负，最高位为"1"代表负；最高位为"0"代表正。

14.5.7 CORDIC_RZR CORDIC Z 结果寄存器

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

Bit	Field	Description
31:24	Reserved	保留，必须保持复位值
23:0	RZ	Z 计算结果（atan 角度计算结果） 注：最高位[23]代表正负，最高位为“1”代表负；最高位为“0”代表正。

14.5.8 CORDIC_CR CORDIC 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.												ENABL	START	WORK	START
												E	MODE	MODE	
												rw	rw	rw	rw

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值
3	ENABLE	模块使能 1：使能 0：禁止
2	STARTMODE	启动模式选择 1：在 START 寄存器位置位为“1”时进行计算 0：执行 CORDIC_DX 寄存器的写操作后启动计算
1	WORKMODE	CORDIC 计算的工作模式选择 1：圆旋转模式 0：圆向量模式

Bit	Field	Description
0	START	在 STARTMODE 设置为"1"时本寄存器位有效，此时，写入"1"且 BSY 位为"0"时为启动一次计算，计算结束后，硬件自动清"0"。

14.5.9 CORDIC_SR CORDIC 状态寄存器

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.														BSY	
														r	

Bit	Field	Description
31:1	Reserved	保留，必须保持复位值
0	BSY	为"1"时表示正在进行计算，计算结束后，硬件自动清"0"。

15 QSPI 四线式 SPI 接口

15.1 简介

四线式 SPI 接口(QSPI)是一个 Memory 控制器,主要用于和带 SPI 兼容接口的外部串行 Memory 设备通信,同时提供了一种直接从外部闪存加载执行程序的机制。该模块支持最常见的串行闪存设备指令,如读取,编程,擦除和其它自定义指令。

15.2 主要特征

- 支持 3 种动作模式: 直接模式 (只读)、间接模式 (可读可写)、非活动模式
- 支持标准 SPI 协议 Mode 0 和 Mode 3 模式
- 支持单线、双线、四线式 Memory 访问
- 支持可编程操作码 (直接模式和间接模式)
- 支持可编程数据格式 (直接模式和间接模式)
- 支持 8 位、16 位、32 位数据访问
- 支持输入数据采样点可调
- 支持 XIP (eXecute In Place) 模式
- 内置 8 级 32 位数据 FIFO (间接模式)
- FIFO 满、FIFO 空、读写访问结束时可发起中断请求
- FIFO 满、FIFO 空、读写访问结束时可发起 DMA 请求

15.3 功能描述

15.3.1 功能框图

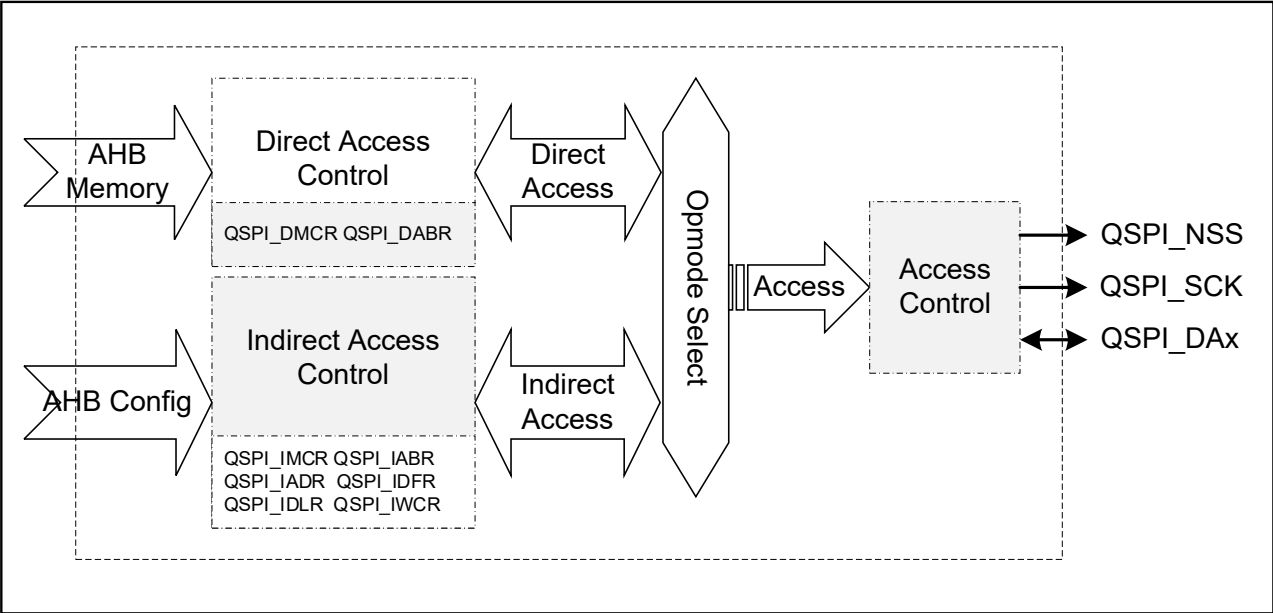


图 15-1 QSPI 功能框图

15.3.2 信号描述

表 15-1 QSPI 信号描述表

信号名	类型	描述
QSPI_SCK	输出	通信时钟输出
QSPI_NSS	输出	片选信号输出（低电平有效）
QSPI_DA0~3	输入输出	通信数据输入输出

15.3.3 数据格式

QSPI 控制器通过命令与外部闪存通信，每条命令包括指令、地址、交替字节、空指令和数据这五个阶段。这些阶段中的任何一个都可配置为跳过，但至少要包含指令、地址、交替字节或数据阶段中的一个。QSPI_NSS 在每条指令开始前下降，在每条指令完成后再次上升。图 15-2 是一个 QSPI 四线式传输的示例。

各命令阶段的数据 Endian 方式遵循标准 SPI 协议，即高位 MSB 先输出，低位 LSB 最后输出。

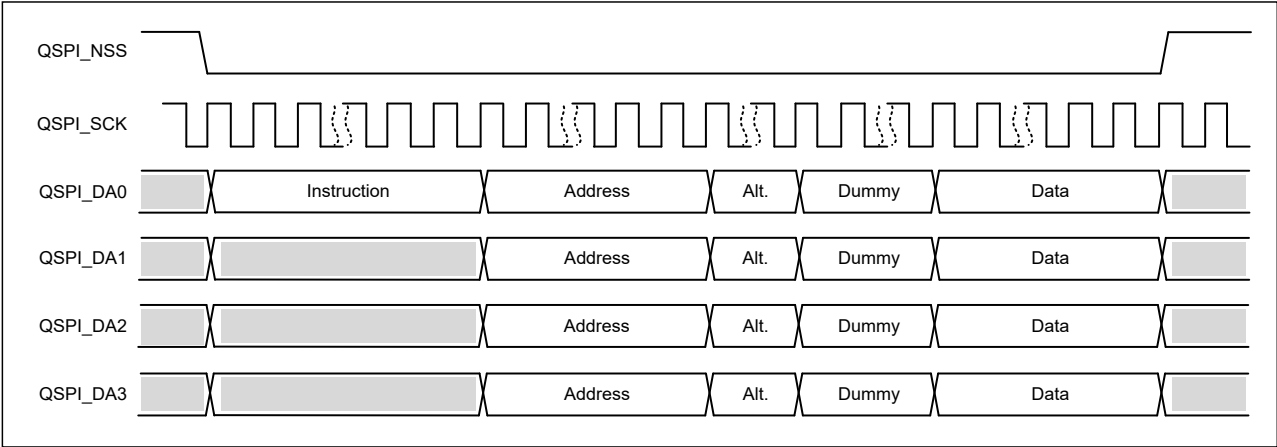


图 15-2 QSPI 数据格式

15.3.3.1 指令阶段（Instruction）

该阶段指明待执行操作的类型，是一个 8 位指令。通过 QSPI_DMCR.INST 位或者 QSPI_IMCR.INST 位设定具体的值。通过 QSPI_DMCR.IMODE 位或者 QSPI_IMCR.IMODE 位指定使用单线、双线、四线传送或者跳过该阶段。

15.3.3.2 地址阶段（Address）

该阶段指明操作地址，它的字节数通过 QSPI_DMCR.ADSIZE 位或者 QSPI_IMCR.ADSIZE 位设定。在间接模式时，通过 QSPI_IADR 寄存器设定具体的值，在直接模式时，由发起的 AHB 访问的地址来决定该阶段的值。通过 QSPI_DMCR.ADMODE 位或者 QSPI_IMCR.ADMODE 位指定使用单线、双线、四线传送或者跳过该阶段。

15.3.3.3 交替字节阶段（Alternate Byte）

该阶段一般用于控制操作模式，它的字节数通过 QSPI_DMCR.ABSIZE 位或者 QSPI_IMCR.ABSIZE 位设定。通过 QSPI_DABR 寄存器或者 QSPI_IABR 寄存器设定具体的值。通过 QSPI_DMCR.ABMODE 位或者 QSPI_IMCR.ABMODE 位指定使用单线、双线、四线传送或者跳过该阶段。

15.3.3.4 空指令阶段（Dummy）

当采用较高的时钟频率时，空指令阶段的设定可以给外部闪存设备留出准备数据阶段的时间。通过 QSPI_DMCR.NUMDC 位或者 QSPI_IMCR.NUMDC 位设定 0~31 个 cycle 的空闲周期。

15.3.3.5 数据阶段 (Data)

该阶段指明操作数据，它的字节数通过 `QSPI_DMCR.DSIZE` 位或者 `QSPI_IMCR.DSIZE` 位设定。在间接模式时，通过 `QSPI_IDFR` 寄存器设定要写出的数据（写模式）或者读取数据（读模式），在直接模式时，由发起的 AHB 读操作读取数据（读模式）。通过 `QSPI_DMCR.DMODE` 位或者 `QSPI_IMCR.DMODE` 位指定使用单线、双线、四线传送或者跳过该阶段。

15.3.4 动作模式

QSPI 控制器支持 3 种操作模式，直接模式、间接模式、非活动模式。在对外部 Memory 有不同访问需求时，3 种模式可满足具体的访问场景。复位之后的初始状态为直接模式。通过设置 `QSPI_CR.OPMODE` 位，可以将 QSPI 控制器配置为其它模式。

直接模式和间接模式使用不同的寄存器组来构造各自的通信。这些设置只有在 `QSPI_CR` 寄存器被激活后配置才生效。用户在使用时，先确认目标模式，配置写入相应的寄存器，然后在写 `QSPI_CR` 寄存器配置模式后，QSPI 将根据 `QSPI_CR.OPMODE` 位的值加载对应模式的参数。

15.3.4.1 直接模式

直接模式下外部闪存的空间映射到 MCU 的地址空间，系统将其视为内部存储器，可进行直接的读取访问。直接读取访问可访问最大为 256MB 的空间。当处于直接模式时，可以随时插入任何其它模式。

直接模式只支持读访问。

15.3.4.2 间接模式

间接模式下所有操作都是通过编程 QSPI 寄存器来执行。它允许软件直接访问内部 TX FIFO 和 RX FIFO。当 `QSPI_IMCR.DMODE` 位不为 0，则 QSPI 将通过数据 FIFO 寄存器 `QSPI_IDFR` 发送或接收一定数量的数据，数据长度由 `QSPI_IDLR` 寄存器配置（最大支持 1MB 数据），数据方向由 `QSPI_IMCR.IDMODE` 位配置。当达到要发送或接收的编程字节数时，状态寄存器 `QSPI_SR.TCF` 位置“1”，表明访问结束。

该模式可以用于访问外部闪存的状态寄存器、保护寄存器以及闪存内容等。对闪存设备的擦除、编程和配置等可通过该模式实现。

间接模式是一种特殊的状态，当间接模式下的访问完成时，硬件将自动返回上一个模式（非活动模式或直接模式，取决于它进入之前的模式）。状态寄存器的 `QSPI_SR.TCF` 位用于指示间接模式下完成了访问操作。

从直接模式切换到间接模式后、在间接模式期间，若 FIFO 容量可满足待处理数据所需空间（间接模式读访问时，RX FIFO 剩余的空 FIFO 个数大于等于剩余的待接收数据，或者间接模式写访问时，TX FIFO 剩余的数据个数大于等于剩余的待发送数据）时，可以发起直接模式读访问请求，此时 AHB Memory

总线接口会保持至此次间接模式结束、然后响应该次直接读访问；否则，直接模式读访问请求被忽略。

间接模式可支持读访问和写访问。

15.3.4.3 非活动模式

QSPI 控制器在和外部设备通信期间，不能随意改变 QSPI_SCK 输出分频（通过 QSPI_CR.SCKDIV 位配置）、QSPI_NSS 的高电平保持时间（通过 QSPI_CR.CSRHT 位配置）、SPI 模式控制（通过 QSPI_CR.SCKMODE 位配置），否则会产生不可预期的动作。因此，提供了非活动模式，用于在通信开始前对这些位进行配置。

非活动模式禁用 QSPI 总线，它提供了一个“干净”状态来设置 QSPI_CR 寄存器的相关位。在非活动模式时，任何对闪存是直接访问都无效。但是可以发出一个间接访问，QSPI 将在完成间接访问传输后返回到非活动模式。

15.3.5 总线接口

QSPI 控制器具有 AHB Memory 总线和 AHB 寄存器配置总线 2 组总线从机接口，AHB Memory 总线接口用于直接模式时主机直接访问外部闪存，AHB 寄存器配置总线接口用于对 QSPI 内部寄存器进行读写访问。2 组总线从机接口均只支持 32 位地址对齐访问操作。

通过 AHB Memory 总线接口对 QSPI 控制器进行访问时，应用程序应禁止以下操作。

- 在非活动模式时进行读访问
- 在状态机从间接模式返回非活动模式时进行读访问
- 在 RX FIFO 剩余的空 FIFO 个数小于剩余的待接收数据（间接模式读操作），或者在 TX FIFO 剩余的数据个数小于剩余的待发送数据（间接模式写操作）时进行读访问
- 任何写访问

通过 AHB 寄存器配置总线接口对 QSPI 控制器进行访问时，应用程序应禁止以下操作。

- 当 FIFO 空时，进行 FIFO 读访问
- 当 FIFO 满时，进行 FIFO 写访问
- 当上一个间接模式访问操作未完成（QSPI_SR.TCF=0）或上一个操作模式更改请求未被授权时（QSPI_SR.OPCRCF=0）时，对 QSPI_CR 寄存器进行写访问

15.3.6 输入数据采样

在某些高速通信的应用场景，时钟、数据通路上物理信号的迟延不同会影响到访问速度。QSPI 控制器可以实现对输入数据采样周期的延迟控制，以达到调整裕量的目的。通过设置 QSPI_DMCR.RXDLY 位或者 QSPI_IMCR.RXDLY 位来控制延迟周期（0~3 个 AHB 总线周期）。图 15-3 所示，是不同数据采样周期的示意图。

注意：用户需根据实际应用中的物理信号延迟信息，对该功能进行配置。

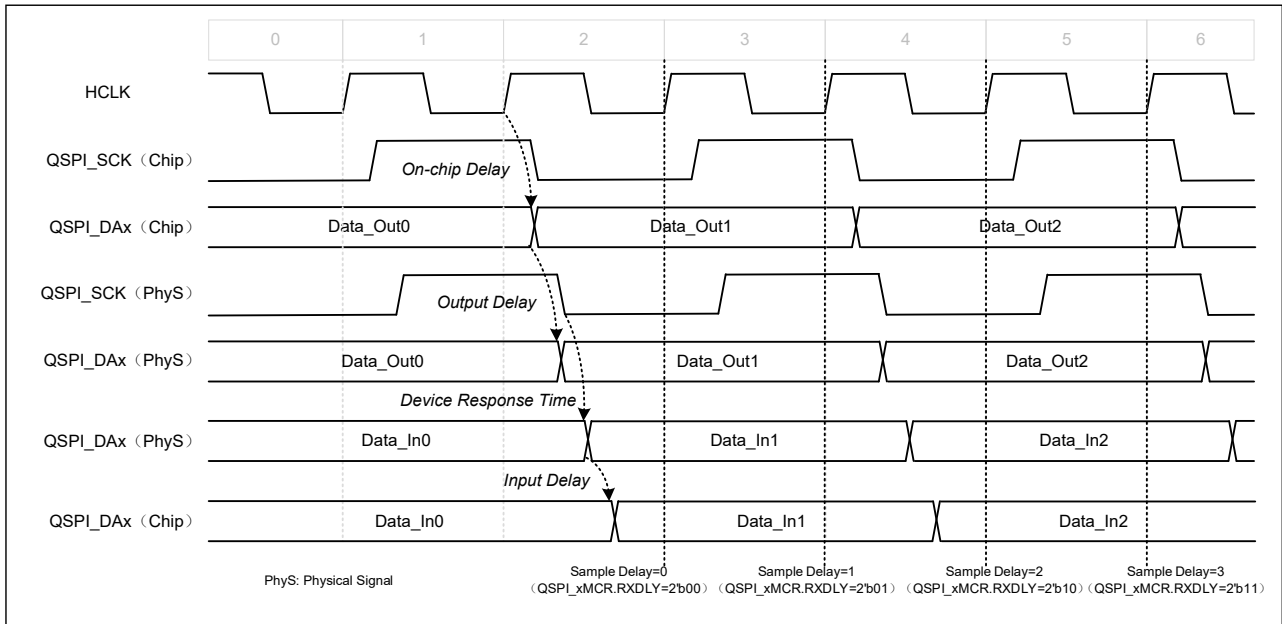


图 15-3 QSPI 输入数据采样周期

15.3.7 空闲状态控制

默认情况下，在选择直接访问模式时，当一次 QSPI 读访问结束，QSPI 控制器不会将 QSPI_NSS 引脚拉高，而是停止 QSPI_SCK 输出，保持当前状态，等待下一个请求。如果下一个请求命中下一个顺序地址，则直接开始 QSPI_SCK 输出进行后续访问；如果下一个请求地址非连续，则 QSPI_NSS 被拉高，经过 QSPI_CR.CSRHT 位设定的时间后，QSPI_NSS 再变为低电平，重新发起访问请求。

QSPI 控制器可配置为自动空闲状态（配置 SYSCFG_CFGR2.QSPI_AUTOIDLE_EN 位），该设定的目的是为了系统在睡眠模式时，QSPI_NSS 引脚被持续拉低处于有效状态。在自动空闲状态设定下，QSPI 控制器将动态监视主机访问。当监测到主机无访问操作时，QSPI 控制器自动进入空闲状态，同时将 QSPI_NSS 拉高；当监测到主机有访问操作时，则重新发起访问请求（根据寄存器设定情况，包含指令阶段、地址阶段等）。

15.3.8 XIP 模式

QSPI 控制器支持一种缩短访问周期、提高访问效率的 XIP 模式。根据 QSPI_CR.XIPMODE 位的设定，QSPI 访问外部设备的动作行为会有不同，具体见下面描述。

- 在直接读模式 & XIP 模式无效（QSPI_CR.XIPMODE=0）时访问外部闪存设备，若地址连续，则后续访问会跳过指令阶段、地址阶段、交替字节阶段和空指令阶段（即使寄存器设定包含这几个阶段），这样可以提高顺序访问的响应速度；若发生地址跳转，则重新发起指令阶段、地址阶段、交替字节阶段和空指令阶段（根据寄存器设定情况）。
- 在直接读模式 & XIP 模式有效（QSPI_CR.XIPMODE=1）时访问外部闪存设备，若地址连

续，则后续访问会跳过指令阶段、地址阶段、交替字节阶段和空指令阶段（即使寄存器设定包含这几个阶段），这样可以提高顺序访问的响应速度；若发生地址跳转，则重新发起地址阶段、交替字节阶段和空指令阶段（根据寄存器设定情况），但不再发起指令阶段（即使寄存器设定包含这个阶段）。

以上两种方式均可实现通过直接模式从外部闪存上读代码并执行，客户需根据外部闪存设备的规格决定是否使能该 XIP 模式。

大多数 SPI 闪存设备都支持这种 XIP 模式。但是，闪存设备制造商没有使用统一的标准方法来进入或者退出该模式。有些闪存设备通过在交替字节阶段使用特殊值来表示下一个连续传输没有指令阶段（XIP 模式）；有些闪存设备通过使用内部配置寄存器来启用 XIP 模式。以下列出 QSPI 控制器支持的进入/退出 XIP 功能方式。

15.3.8.1 通过交替字节进入

将寄存器 QSPI_DABR 设置为特定值，该值将通知 Flash 进入 XIP 模式，并将 QSPI_CR.XIPMODE 位设置为 1，QSPI 控制器和外部闪存将在下一个新传输中以 XIP 模式通信。

15.3.8.2 通过交替字节退出

将寄存器 QSPI_DABR 设置为除特定 XIP 模式之外的任何值，该值将通知 Flash 退出 XIP 模式，并将 QSPI_CR.XIPMODE 位设置为 0，QSPI 控制器和外部闪存将在下一个新传输中以正常模式通信。

15.3.8.3 通过配置寄存器进入

通过寄存器 QSPI_IMCR 和 QSPI_IABR（QSPI_IDFR 等也可以，用于存储要写入外部闪存配置寄存器的值）准备直接读取模式的设置，该设置在 XIP 模式下使用，不必设置 QSPI_CR 寄存器。

设置 QSPI_IMCR.XIPIM 位为 1。间接模式访问完成后，QSPI 控制器和外部闪存将立即在 XIP 模式下工作。寄存器 QSPI_SR.XIPST 位将指示 QSPI 控制器状态。

15.3.8.4 通过配置寄存器退出

通过寄存器 QSPI_IMCR 和 QSPI_IABR（QSPI_IDFR 等也可以，用于存储要写入外部闪存配置寄存器的值）准备直接读取模式的设置，该设置在普通模式下使用，不必设置 QSPI_CR 寄存器。

设置 QSPI_IMCR.XIPIM 位为 0。间接模式访问完成后，QSPI 控制器和外部闪存将立即在普通模式下工作。寄存器 QSPI_SR.XIPST 将指示 QSPI 控制器状态。

15.3.9 中断

QSPI 控制器支持下面 3 种中断源（仅在间接模式时有效）：

- FIFO 空
- FIFO 满
- 读写访问完成

在间接模式下，发生以上 3 种事件中的任何一个，均可产生 QSPI 中断。

表 15-2 QSPI 中断请求

中断事件	状态	中断使能位
FIFO 空	QSPI_SR.EMPTY	QSPI_IDER.EMPTYINTEN
FIFO 满	QSPI_SR.FULL	QSPI_IDER.FULLINTEN
读写访问完成	QSPI_SR.TCF	QSPI_IDER.TCFINTEN

如果设置了对应的中断使能控制位，就可以产生各自对应的中断。

15.3.10 DMA

QSPI 控制器支持下面 3 种 DMA 事件请求源（仅在间接模式时有效）：

- FIFO 空
- FIFO 满
- 读写访问完成

在间接模式下，发生以上 3 种事件中的任何一个，均可产生 QSPI DMA 请求。

表 15-3 QSPI DMA 请求

DMA 事件	状态	DMA 使能位
FIFO 空	QSPI_SR.EMPTY	QSPI_IDER.EMPTYDMAEN
FIFO 满	QSPI_SR.FULL	QSPI_IDER.FULLDMAEN
读写访问完成	QSPI_SR.TCF	QSPI_IDER.TCFDMAEN

如果设置了对应的 DMA 使能控制位，就可以产生各自对应的 DMA 请求。

间接模式下对外部闪存设备进行读写时，可使用相应的 DMA 请求来触发 DMA 主机实现数据搬运。在间接模式写的应用场景下，对外部闪存设备配置流程中，使用数据传输开始前的一个 Command 序列的 FIFO 空事件（含数据段的 Command）或传输完成事件（不含数据段的 Command）来发起对 DMA 传输的首次请求，后续数据搬运仍使用 FIFO 空事件触发 DMA 传输。

15.4 寄存器

15.4.1 寄存器总览

表 15-4 QSPI 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	QSPI_CR	控制寄存器	0x03071B00
0x04	QSPI_SR	状态寄存器	0x08000104

Offset	Acronym	Register Name	Reset
0x08	QSPI_DMCR	直接模式控制寄存器	0x00342503
0x0C	QSPI_IMCR	间接模式控制寄存器	0x00342500
0x10	QSPI_DABR	直接模式交替字节寄存器	0x00000000
0x14	QSPI_IABR	间接模式交替字节寄存器	0x00000000
0x18	QSPI_IADR	间接模式地址寄存器	0x00000000
0x1C	QSPI_IDFR	间接模式数据 FIFO 寄存器	0x00000000
0x20	QSPI_IDLR	间接模式数据长度寄存器	0x00000000
0x24	QSPI_IWCR	间接模式等待计数寄存器	0x00000000
0x28	QSPI_IDER	中断 DMA 使能寄存器	0x00000000

15.4.2 QSPI_CR 控制寄存器

偏移地址: 0x00

复位值: 0x0307 1B00

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

Bit	Field	Description
31: 30	Reserved	保留, 必须保持复位值
29: 24	SCKDIV	输出时钟分频 (QSPI_SCK Output Clock Pre-scalar) 设定 QSPI_SCK 时钟输出频率 (输出时钟占空比为 50%) 000000: AHB_CLK/2 000001: AHB_CLK/4 000010: AHB_CLK/6 000011: AHB_CLK/8 ... 111110: AHB_CLK/126 111111: AHB_CLK/128
23: 19	Reserved	保留, 必须保持复位值
18: 16	CSRHT	高电平时间 (High Level Cycles) 设定两次 QSPI_NSS 有效电平 (Low Level) 之间的高电平保持时间 000: AHB_CLK*2 001: AHB_CLK*3 010: AHB_CLK*4 ... 111: AHB_CLK*9

Bit	Field	Description
15: 6	Reserved	保留，必须保持复位值
5	XIPMODE	XIP 模式使能 (XIP Mode Enable) 表示 QSPI 对 XIP 控制请求 0: 退出 XIP 模式 1: 进入 XIP 模式 该位设定只在直接模式时有效；在间接模式时，从间接模式返回直接模式后，该位状态由 QSPI_IMCR.XIPIM 位的设定值决定，且在 QSPI_SR.TCF 位置“1”时生效
4	SCKMODE	SPI 模式选择 (SPI Mode Select) 0: Mode 0 1: Mode 3 该位仅在非活动模式时设置有效，在其它模式时设置无效
3: 2	Reserved	保留，必须保持复位值
1: 0	OPMODE	操作模式选择 (Operation Mode Select) 00: 直接模式 01: Reserved 10: 间接模式 11: 非活动模式 当处于间接模式时，禁止对该寄存器位进行写操作；当对该位写“00”或从间接模式跳回“00”时，QSPI_DMCR/QSPI_DABR 中的设定值会更新进新的数据传输中；当对该位写“10”时，寄存器 QSPI_IMCR/QSPI_IABR/QSPI_IADR/QSPI_IDLR 中的设定值会更新进新的数据传输中

15.4.3 QSPI_SR 状态寄存器

偏移地址: 0x04

复位值: 0x0800 0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				DEPTH				Res.							
				r											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVEL				Res.		FULL	EMPTY	Res.	XIPST	BUSY	TCF	Res.	OPCRC F	CUROP	
r						r	w1c		r	r	w1c		r	r	

Bit	Field	Description
31: 28	Reserved	保留，必须保持复位值
27: 24	DEPTH	FIFO 深度 (FIFO Depth) 表示可支持 FIFO 级数，只读为“0x08”，表示 8 级 FIFO
23: 16	Reserved	保留，必须保持复位值

Bit	Field	Description
15: 12	LEVEL	待处理 FIFO 级数 (Remain FIFO Level) 间接模式时有效, 表示待处理的 FIFO 数。当 FIFO 空时, 该位为“0”; 当 FIFO 满时, 该位为“8”。 在间接读模式时, 该位表示已接收的数据数 (待读取); 在间接写模式时, 该位表示剩余的数据数 (待发送)
11: 10	Reserved	保留, 必须保持复位值
9	FULL	FIFO 满标志 (FIFO Full Flag) 仅在间接模式时有效, 表示当前 FIFO 已满
8	EMPTY	FIFO 空标志 (FIFO Empty Flag) 仅在间接模式时有效, 表示当前 FIFO 已空。 向该位写“1”可复位 FIFO。在发起复位操作之前, 必须确保期待的传输 已完成或者准备发起一次新的传输
7	Reserved	保留, 必须保持复位值
6	XIPST	XIP 状态 (XIP Status) 表示控制器当前是否处于 XIP 模式 直接模式时, 该位在读访问发生地址跳转时置位; 从间接模式返回直接 模式时, 该位状态由 QSPI_IMCR.XIPIM 位的设定值决定, 且在 QSPI_SR.TCF 位置“1”时生效
5	BUSY	控制器状态 (Controller Status) 0: 控制器处于空闲状态, QSPI_SCK 无翻转 1: 控制器处于动作中, QSPI_SCK 在翻转
4	TCF	传输完成标志 (Transfer Complete Flag) 仅在间接模式时有效, 表示当前传输已完成 向该位写“1”清除该位
3	Reserved	保留, 必须保持复位值
2	OPCRCF	模式切换状态 (Mode Change Status) 表示最近一次的操作模式切换请求已完成 当对 CR 寄存器进行写操作后该位自动清“0”; 当操作模式改变完成时, 该位置“1”
1: 0	CUROP	当前操作模式 (Current Operation Mode) 00: 直接模式 01: Reserved 10: 间接模式 11: 非活动模式

15.4.4 QSPI_DMCR 直接模式控制寄存器

偏移地址: 0x08

复位值: 0x0034 2503

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			NUMDC					RXDLY		DSIZE		DMODE		ABSIZE	

			rw					rw		r		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INST							
rw		rw		rw		rw		rw							

Bit	Field	Description
31: 29	Reserved	保留，必须保持复位值
28: 24	NUMDC	空指令阶段加入的 SCK 周期数 (SCK Cycles at Empty Phase) 00000: 无空指令 Cycles 00001: 加入 1 个 SCK Cycles 的空指令 00010: 加入 2 个 SCK Cycles 的空指令 ... 11111: 加入 31 个 SCK Cycles 的空指令
23: 22	RXDLY	输入数据的采样延迟 (Delay Cycles when Sample the Input Data) 00: 无采样延迟 01: 加入 1 个 AHB_CLK 的采样延迟 10: 加入 2 个 AHB_CLK 的采样延迟 11: 加入 3 个 AHB_CLK 的采样延迟
21: 20	DSIZE	数据阶段位宽 (Data Phase Size) 表示支持的数据位宽数 00: Reserved 01: Reserved 10: Reserved 11: 32bit
19: 18	DMODE	数据阶段模式 (Data Phase Mode) 表示支持的数据线宽数 00: Reserved 01: 单线式 10: 双线式 11: 四线式
17: 16	ABSIZE	交替字节阶段位宽 (Alternate Byte Phase Size) 表示支持的交替字节位宽数 00: 8bit 01: 16bit 10: 24bit 11: 32bit
15: 14	ABMODE	交替字节阶段模式 (Alternate Byte Phase Mode) 表示支持的交替字节线宽数 00: 无交替字节阶段 01: 单线式 10: 双线式 11: 四线式

Bit	Field	Description
13: 12	ADSIZE	地址阶段位宽 (Address Phase Size) 表示支持的地址位宽数 00: Reserved 01: Reserved 10: 24bit 11: 32bit
11: 10	ADMODE	地址阶段模式 (Address Phase Mode) 表示支持的地址线宽数 00: 无地址阶段 01: 单线式 10: 双线式 11: 四线式
9: 8	IMODE	指令阶段模式 (Instruction Phase Mode) 表示支持的指令线宽数 00: 无指令阶段 01: 单线式 10: 双线式 11: 四线式
7: 0	INST	指令代码 (Instruction Code) 表示直接模式时指令阶段的设定值

15.4.5 QSPI_IMCR 间接模式控制寄存器

偏移地址: 0x0C

复位值: 0x0034 2500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	IDMODE	XIPIM	NUMDC					RXDLY		DSIZE		DMODE		ABSIZE	
	rw	rw	rw					rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INST							
rw		rw		rw		rw		rw							

Bit	Field	Description
31	Reserved	保留, 必须保持复位值
30	IDMODE	间接模式访问模式 (Indirect Mode Access) 0: 读访问 1: 写访问 对该位进行写操作, 同时使能 RX FIFO 或 TX FIFO; 当改变该位的设定时, FIFO 也发生一次复位; 当一次间接访问未结束时, 不能改变该位设定值, 若改写该位设定值, FIFO 方向将发生变化, 产生访问错误

Bit	Field	Description
29	XIPIM	从间接模式返回时 XIP 控制 (XIP Control from Indirect Mode) 该位表示从间接模式返回直接模式时, 对 XIP 的控制请求 0: 从间接模式返回直接模式时,QSPI 退出 XIP 模式 1: 从间接模式返回直接模式时,QSPI 进入 XIP 模式
28: 24	NUMDC	空指令阶段加入的 SCK 周期数 (SCK Cycles at Empty Phase) 00000: 无空指令 Cycles 00001: 加入 1 个 SCK Cycles 的空指令 00010: 加入 2 个 SCK Cycles 的空指令 ... 11111: 加入 31 个 SCK Cycles 的空指令
23: 22	RXDLY	输入数据的采样延迟 (Delay Cycles when Sample the Input Data) 00: 无采样延迟 01: 加入 1 个 AHB_CLK 的采样延迟 10: 加入 2 个 AHB_CLK 的采样延迟 11: 加入 3 个 AHB_CLK 的采样延迟
21: 20	DSIZE	数据阶段位宽 (Data Phase Size) 表示支持的数据位宽数 00: 8bit 01: 16bit 10: Reserved 11: 32bit
19: 18	DMODE	数据阶段模式 (Data Phase Mode) 表示支持的数据线宽数 00: 无数据阶段 01: 单线式 10: 双线式 11: 四线式
17: 16	ABSIZE	交替字节阶段位宽 (Alternate Byte Phase Size) 表示支持的交替字节位宽数 00: 8bit 01: 16bit 10: 24bit 11: 32bit
15: 14	ABMODE	交替字节阶段模式 (Alternate Byte Phase Mode) 表示支持的交替字节线宽数 00: 无交替字节阶段 01: 单线式 10: 双线式 11: 四线式

Bit	Field	Description
13: 12	ADSIZE	地址阶段位宽 (Address Phase Size) 表示支持的地址位宽数 00: 8bit 01: 16bit 10: 24bit 11: 32bit
11: 10	ADMODE	地址阶段模式 (Address Phase Mode) 表示支持的地址线宽数 00: 无地址阶段 01: 单线式 10: 双线式 11: 四线式
9: 8	IMODE	指令阶段模式 (Instruction Phase Mode) 表示支持的指令线宽数 00: 无指令阶段 01: 单线式 10: 双线式 11: 四线式
7: 0	INST	指令代码 (Instruction Code) 表示间接模式时指令阶段的设定值

15.4.6 QSPI_DABR 直接模式交替字节寄存器

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

Bit	Field	Description
31: 0	DALT	直接模式交替字节值 (Direct Mode Alternate Byte Value) 表示直接模式时交替字节阶段的设定值

15.4.7 QSPI_IABR 间接模式交替字节寄存器

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
IALT															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IALT															
rw															
Bit		Field				Description									
31: 0		IALT				间接模式交替字节值 (Indirect Mode Alternate Byte Value) 表示间接模式时交替字节阶段的设定值									

15.4.8 QSPI_IADR 间接模式地址寄存器

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
IADDR															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IADDR															
rw															
Bit		Field				Description									
31: 0		IADDR				间接模式地址值 (Indirect Mode Address Value) 表示间接模式时地址阶段的设定值									

15.4.9 QSPI_IDFR 间接模式数据 FIFO 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
IDATA															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA															
rw															

Bit	Field	Description
31: 0	IDATA	间接模式数据值（Indirect Mode Data Value） 表示在间接模式时发送给外部闪存的数据（写操作）或从外部闪存读入的数据（读操作） 该寄存器只能 32 位访问，但是传输数据本身根据 QSPI_IMCR.DSIZE（8 位、16 位、32 位）设定会有不同。例如，在 8 位数据位宽时，3 个字节的数据需要读取 3 次该寄存器；在 16 位数据位宽时，4 个字节的数据需要读取 2 次该寄存器。 应用程序在对该寄存器进行访问之前，要先检查相关的寄存器状态，以实现正确的访问

15.4.10 QSPI_IDLR 间接模式数据长度寄存器

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

Bit	Field	Description
31: 20	Reserved	保留，必须保持复位值
19: 0	IDLEN	间接模式数据长度（Indirect Mode Data Length） 表示间接模式时要传输的数据大小 0: 1 个字节的数据 1: 2 个字节的数据 2: 3 个字节的数据 ... 0xFFFF: 1048576 个（1M）字节的数据 在 QSPI_IMCR.DSIZE 设定为 16bit 位宽时，IDLEN[0] 默认为“1”；在 QSPI_IMCR.DSIZE 设定为 32bit 位宽时，IDLEN[1: 0] 默认为“11”

15.4.11 QSPI_IWCR 间接模式等待计数寄存器

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.		IWCNT													
		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWCNT															
rw															
Bit		Field		Description											
31: 30		Reserved		保留，必须保持复位值											
29: 0		IWCNT		间接模式等待计数器（Indirect Mode Wait Counter） 表示从间接模式返回直接模式时的等待周期（AHB_CLK 周期） 在间接模式完成传输操作之后，若该位设定不为零，则需要等待该位设定的时间周期+3*AHB_CLK 周期之后返回到直接模式；若该位设定为零，则等待 2*AHB_CLK 周期之后返回到直接模式											

15.4.12 QSPI_IDER 中断 DMA 使能寄存器

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
FULLDMAEN	EMPTYDMAEN	Res.						TCFDMAEN	FULLINTEN	EMPTYINTEN	Res.				TCFINTEN
rw	rw							rw	rw	rw					rw

Bit		Field		Description											
31: 16		Reserved		保留，必须保持复位值											
15		FULLDMAEN		FIFO 满 DMA 请求（DMA Request when FIFO Full） 表示 FIFO 满事件发生时 DMA 请求处理 0: 该事件发生时，不发出 DMA 请求 1: 该事件发生时，发出 DMA 请求											
14		EMPTYDMAEN		FIFO 空 DMA 请求（DMA Request when FIFO Empty） 表示 FIFO 空事件发生时 DMA 请求处理 0: 该事件发生时，不发出 DMA 请求 1: 该事件发生时，发出 DMA 请求											
13: 9		Reserved		保留，必须保持复位值											

Bit	Field	Description
8	TCFDMAEN	传输完成 DMA 请求 (DMA Request when Transfer Complete) 表示传输完成事件发生时 DMA 请求处理 0: 该事件发生时, 不发出 DMA 请求 1: 该事件发生时, 发出 DMA 请求
7	FULLINTEN	FIFO 满 INT 请求 (Interrupt Request when FIFO Full) 表示 FIFO 满事件发生时中断请求处理 0: 该事件发生时, 不发出中断请求 1: 该事件发生时, 发出中断请求
6	EMPTYINTEN	FIFO 空 INT 请求 (Interrupt Request when FIFO Empty) 表示 FIFO 空事件发生时中断请求处理 0: 该事件发生时, 不发出中断请求 1: 该事件发生时, 发出中断请求
5: 1	Reserved	保留, 必须保持复位值
0	TCFINTEN	传输完成 INT 请求 (Interrupt Request when Transfer Complete) 表示传输完成事件发生时中断请求处理 0: 该事件发生时, 不发出中断请求 1: 该事件发生时, 发出中断请求

16 FSMC 存储器接口

16.1 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。

16.1.1 框图

下图为 FSMC 的框图：

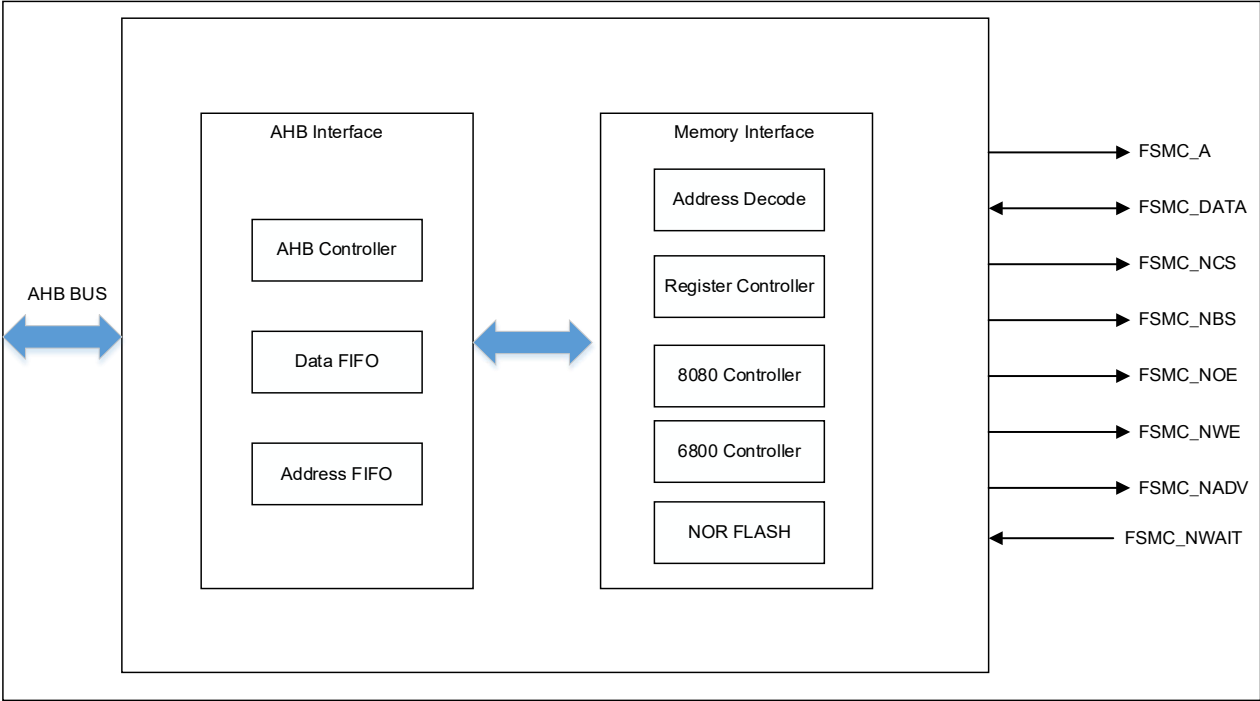


图 16-1 FSMC 框图

16.1.2 接口描述

表 16-1 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
FSMC_NADV	OUTPUT	地址有效信号

信号名	方向	描述
FSMC_NWAIT	INPUT	FSMC 输入等待信号

16.1.3 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 16-2 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 16-3 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

16.2 功能描述

16.2.1 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 16-4 所示。

表 16-4 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 SMCTLR 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。

16.2.2 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 0，即地址数据线复用。

以下为典型应用场景配置：

表 16-5 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 16-6 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

模式	读/写	AHB 传输宽度	存储宽度	备注
	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。

16.2.3 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 16-7 所示，时序参数说明如表 16-8 所示，NOR Flash/SRAM 写时序如图 16-2 所示，其读时序如图 16-3 所示。

表 16-7 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 16-8 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

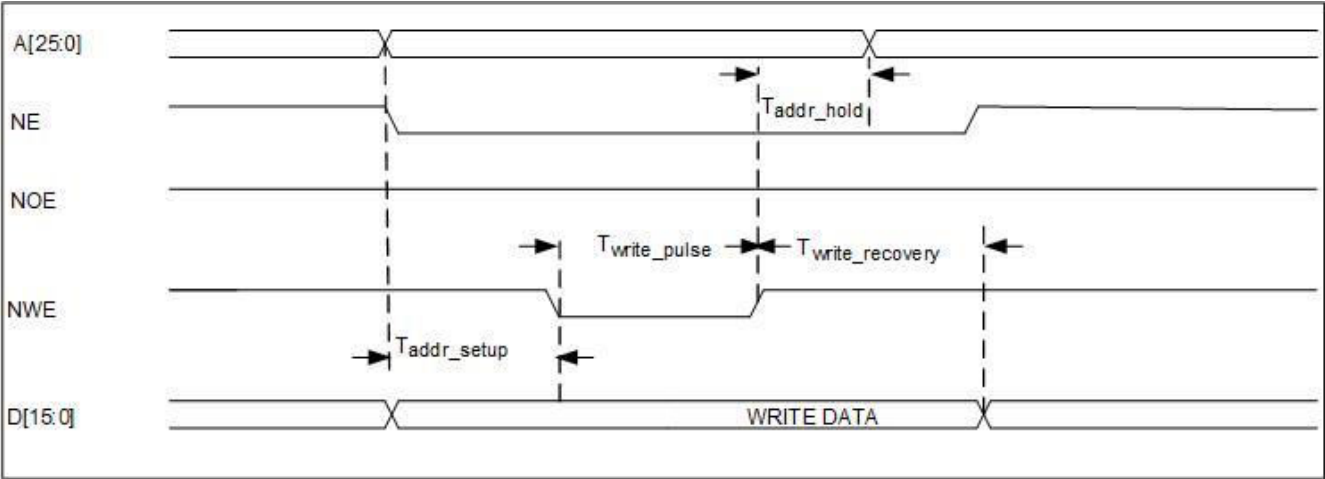


图 16-2 NOR FLASH/SRAM 写时序

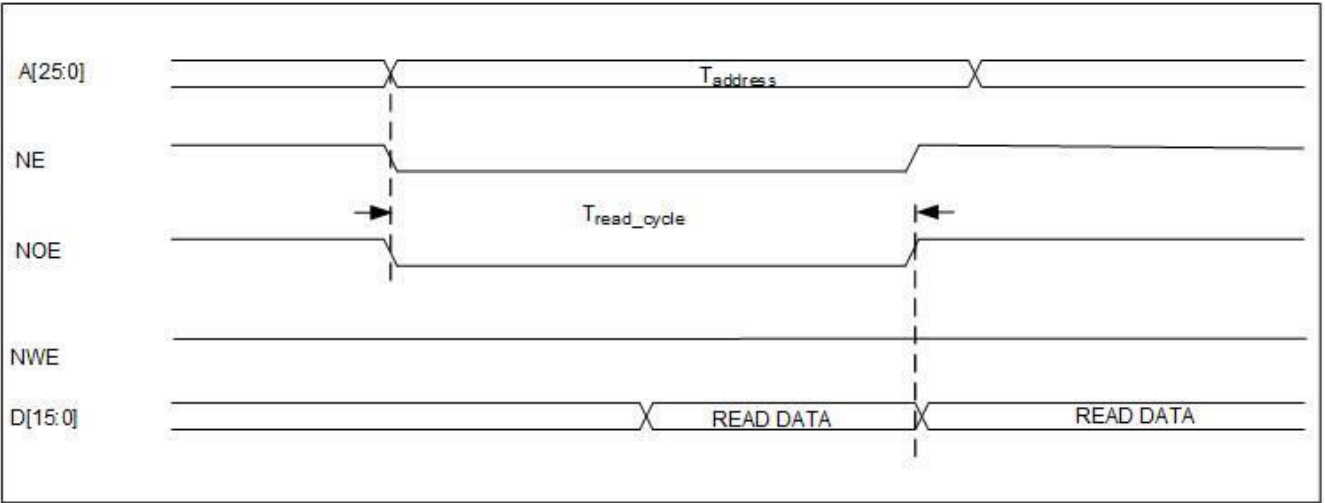


图 16-3 NOR FLASH/SRAM 读时序

16.2.4 8080 协议模式

8080 协议通信只用到了部分接口信号，接口如表 16-9 所示，地址说明如表 16-10 所示，8080 时序参数说明如表 16-11 所示，8080 写时序如图 16-4 所示，其读时序如图 16-5 所示。

表 16-9 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 16-10 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 16-11 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

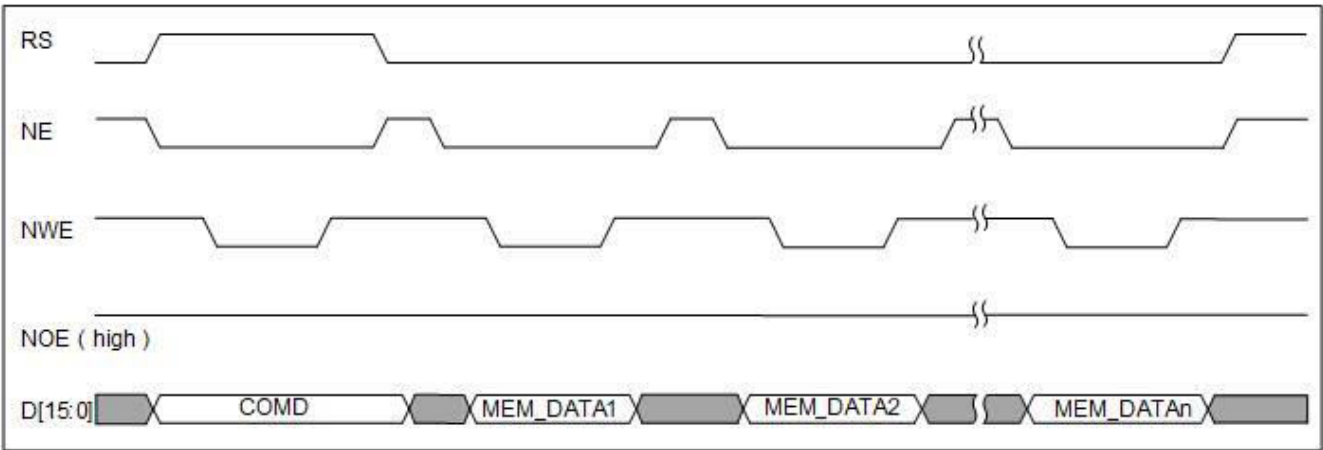


图 16-4 8080 写时序

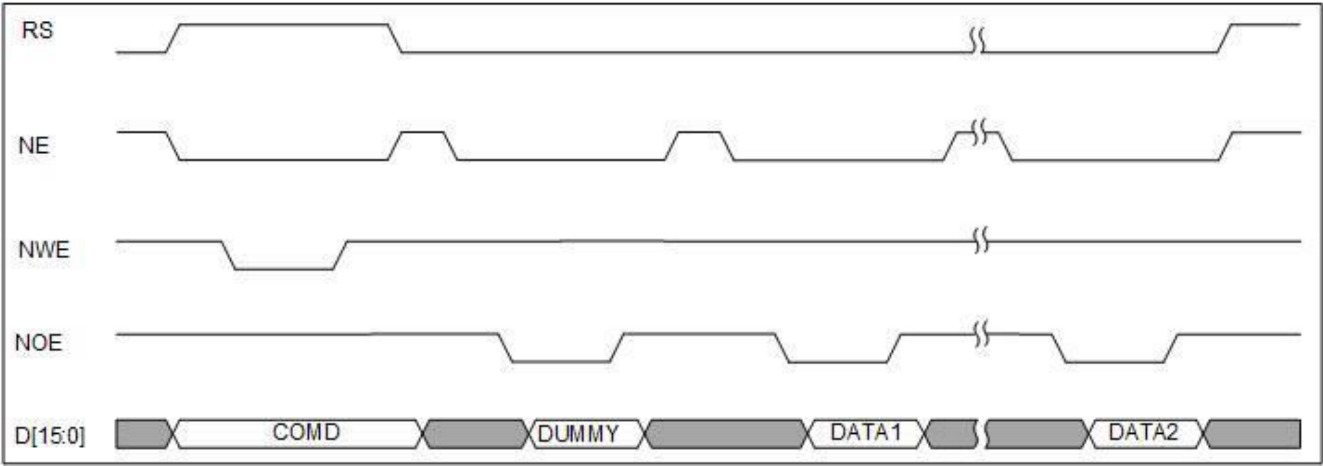


图 16-5 8080 读时序

16.2.5 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 16-12 所示，地址说明如表 16-13 所示，时序参数说明如表 16-14 所示，6800 写时序如图 16-6 所示，其读时序如图 16-7 所示。

表 16-12 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 16-13 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+（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。

6800 写操作时序：

表 16-14 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

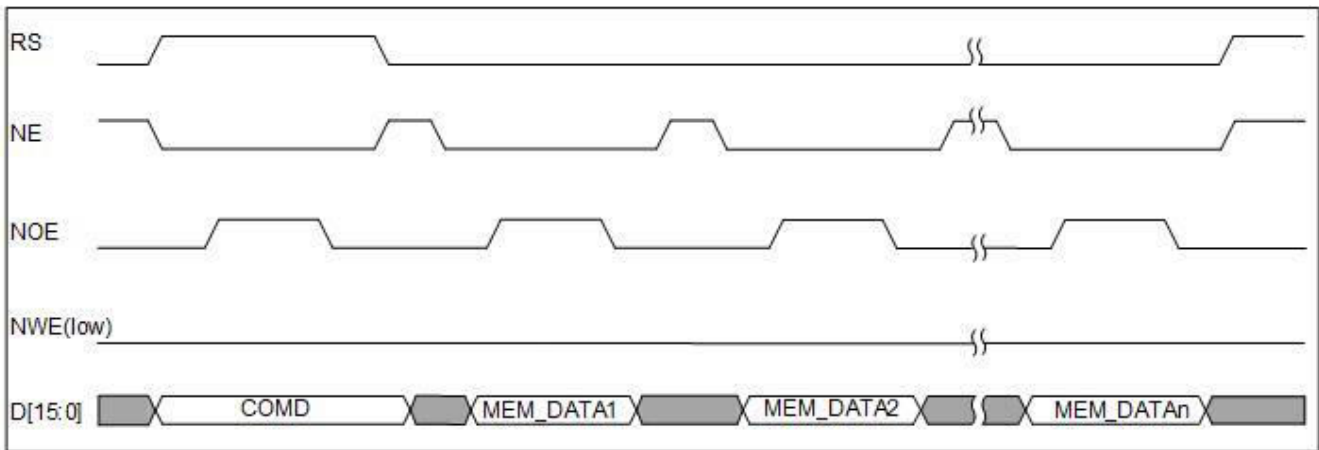


图 16-6 6800 写时序

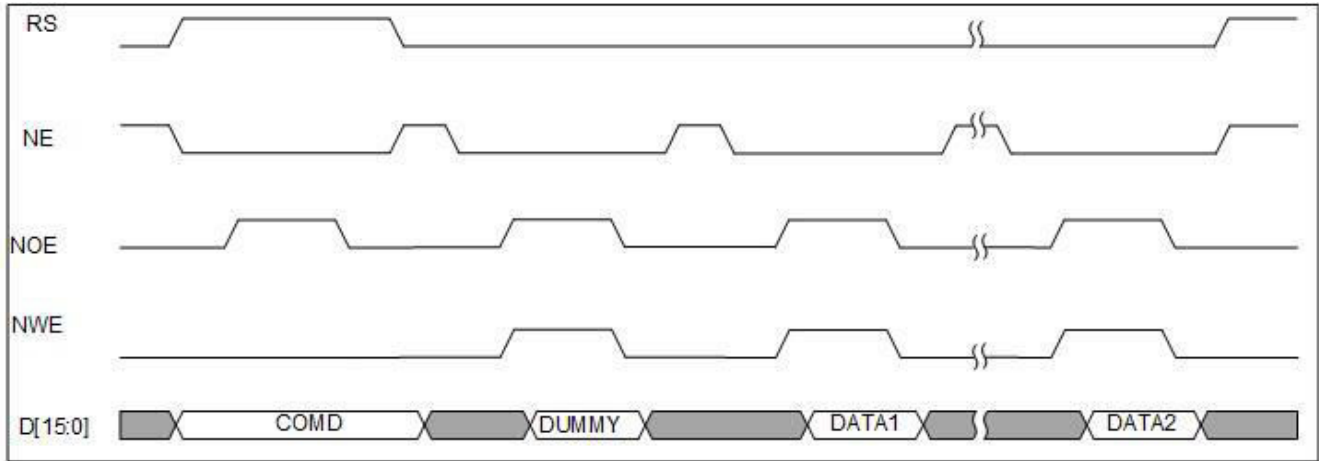


图 16-7 6800 读时序

16.3 寄存器描述

表 16-15 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	SMCTLR	存储器控制寄存器	0x00000000

16.3.1 存储器屏蔽寄存器（SMSKR0）

地址偏移：0x54

复位值：0x0000024B

31	30	29	28	27	26	25	24	23	22	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	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
7:5	MEM_TYPE	MEM_TYPE[2:0]: 外接设备类型 001: SRAM 010: NOR FLASH 其他: 保留

Bit	Field	Description
4:0	MEM_SIZE	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

16.3.2 存储器屏蔽寄存器 (SMSKR1)

地址偏移: 0x58

复位值: 0x0000024B

31	30	29	28	27	26	25	24	23	22	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	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 BANK2 传输的时序参数 001: 选择 SMTMGR_SET1 寄存器来配置 BANK2 传输的时序参数 010: 选择 SMTMGR_SET2 寄存器来配置 BANK2 输的时序参数 其它: 保留
7:5	MEM_TYPE	MEM_TYPE[2:0]: 外接设备类型 001: SRAM 010: NOR FLASH 其他: 保留

Bit	Field	Description
4:0	MEM_SIZE	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

16.3.3 存储器屏蔽寄存器 (SMSKR2)

地址偏移: 0x5C

复位值: 0x0000002B

31	30	29	28	27	26	25	24	23	22	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	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 BANK3 传输的时序参数 001: 选择 SMTMGR_SET1 寄存器来配置 BANK3 传输的时序参数 010: 选择 SMTMGR_SET2 寄存器来配置 BANK3 传输的时序参数 其它: 保留
7:5	MEM_TYPE	MEM_TYPE[2:0]: 外接设备类型 001: SRAM 010: NOR FLASH 其他: 保留

Bit	Field	Description
4:0	MEM_SIZE	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

16.3.4 存储器屏蔽寄存器 (SMSKR3)

地址偏移: 0x60

复位值: 0x0000014B

31	30	29	28	27	26	25	24	23	22	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	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 BANK4 传输的时序参数 001: 选择 SMTMGR_SET1 寄存器来配置 BANK4 传输的时序参数 010: 选择 SMTMGR_SET2 寄存器来配置 BANK4 传输的时序参数 其它: 保留
7:5	MEM_TYPE	MEM_TYPE[2:0]: 外接设备类型 001: SRAM 010: NOR FLASH 其他: 保留

Bit	Field	Description
4:0	MEM_SIZE	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

16.3.5 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

16.3.6 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.									
		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

16.3.7 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

16.3.8 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

17 ADC 模拟/数字转换

17.1 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 有可测量内部或外部信号源，其中 ADC1 有 19 路外部输入通道，ADC2 有 17 路外部输入通道和 2 路内部通道。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。

ADC 的最大输入时钟为 48MHz，它是由 APB2 时钟(PCLK2)分频产生。

17.2 功能框图

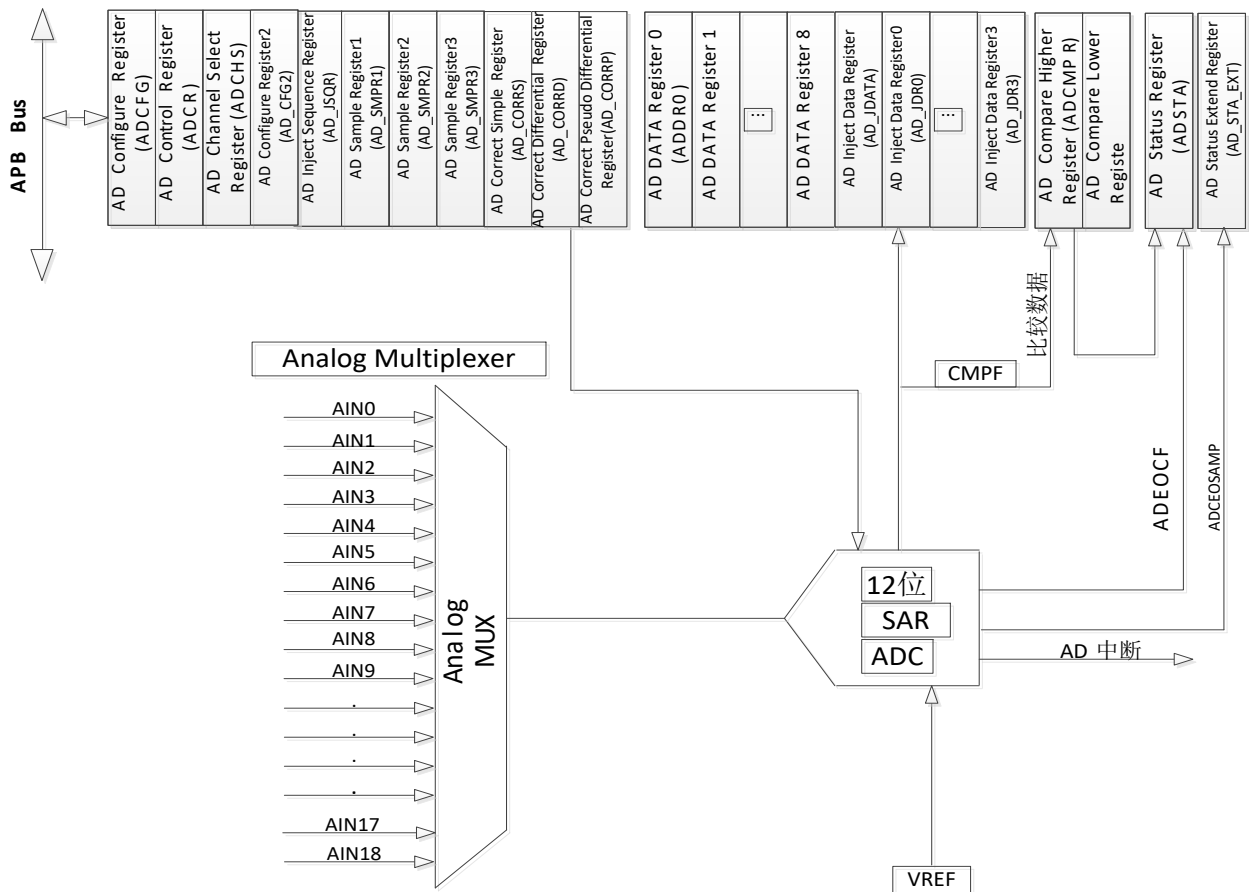


图 17-1 ADC 系统框图

注：T_SENSOR（温度传感器）通道在 ADC2 的 AIN17 通道，V_SENSOR（内部参考电压）通道在 ADC2 的 AIN18 通道。

17.3 主要特征

- 高达 3Msps 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。

- 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转换。

- 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。触发源包括：Timer 和 EXTI
- 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。
- 支持自校准功能，输入时钟 1.5MHz
- 支持单端、差分、伪差分转换
- 支持过采样

17.4 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。

中断	标志位
模拟看门狗比较器中断	AWDIF

中断	标志位
结束校准中断	EOCALIF
通道序列转换结束中断	EOSIF
单通道转换结束中断	EOCIF
采样结束中断	EOSMPIF
注入通道序列转换结束中断	JEOSIF
注入通道转换结束中断	JEOCIF
注入通道采样结束中断	JEOSMPIF

17.5 DMA

A/D 转换结果存储在数据寄存器 ADC_ADDDATA 中，当多个通道连续转换时可以使用 DMA 访问保存数据，避免转换数据丢失。

DMA 使能开启，当通道转换结束后将产生 DMA 请求，将转换数据从 ADC_ADDDATA 寄存器传输到软件指定的目的地址。

每个通道有对应的数据寄存器 ADC_ADDRn，可以访问这些寄存器获取各通道的转换结果。

17.6 功能描述

17.6.1 时钟

ADC 时钟可以通过 RCC_CFG.ADC1CLKSEL 或 RCC_CFG.ADC2CLKSEL 选择 PCLK2 和 PLL2 作为时钟源。在使用 ADC 之前，要先使能 RCC 控制器中的时钟使能控制位。ADC 的自校准过程时钟可通过配置 RCC_ADC1CFGR.PRECAL 或 RCC_ADC2CFGR.PRECAL 为偶数分频；ADC 的转换过程时钟可通过配置 RCC_ADC1CFGR.PRE 或 RCC_ADC2CFGR.PRE 为偶数分频。

17.6.2 数据补偿

只有注入通道数据有数据补偿。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：

分辨率选择	模拟转换完成数据 (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”
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”

17.6.3 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。

注入组通道转换的数据值减去偏移量（ADC_JOFRn 寄存器值），结果可以是负值，SEXT 位是扩展的符号值。

如下图所示：

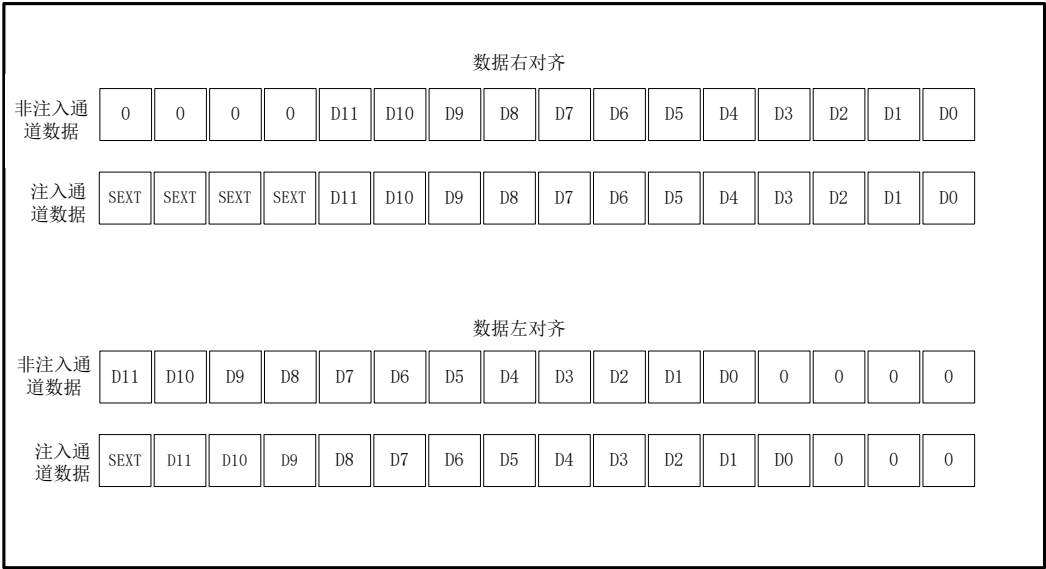


图 17-2 数据对齐方式

17.6.4 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。

17.6.5 可编程采样时间

ADC 转换通道的采样时间可通过 ADC_SMPR1、ADC_SMPR2 和 ADC_SMPR3 寄存器中的 SAMPCTL0~SAMPCTL18 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 RCC_ADC1CFGR.PRE 位来确定，即 $PCLK2/(PRE+2)$ 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、ADC_SMPR2 和 ADC_SMPR3 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 $m \cdot T$ (T 为 ADC 模块的时钟周期)。

转换频率计算如下：

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

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

总转换时间计算如下：

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

例如当 ADC_CLK=48MHz，采样时间为 3.5T， $T_{\text{CONV}} = (3.5+12.5) \cdot T = 16 \cdot T$ ，总转换时间为 0.33μs。

17.6.6 数据通道寄存器

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

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

17.6.7 通道选择

ADC1 有 19 路外部输入通道 0~18，ADC2 有 17 路外部输入通道 0~16、内部温度传感器通道 17、内部 1.2V 参考电压通道 18。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_SEL0~18 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。

17.6.8 ADC 开关

ADC_ADCFG.ADEN 位可控制 A/D 自校准和转换，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。

17.6.9 校准

ADC 的校准可以减小内部电容变化造成的精度误差。ADC 转换结果误差校准的计算基于校准因子。

软件启动自校准过程获得校准因子后，用户通过读取 ADDATA 寄存器保存这些校准因子。在 standby 唤醒或者重新上电后，既可以重新启动自校准过程获取校准因子也可以通过寄存器写入校准因子。

注意：

在转换过程启动前必须获得校准因子

开始自校准前 ADC 必须停止 AD 转换

17.6.9.1 自校准

置位 ADC_ADCFG.ADEN 后配置 ADC_ANY_CR.ADCAL 开启自校准过程，自校准完成后 ADC_ADSTA_EXT.EOCALIF 校准完成标志有效。如果设置 ADC_ANY_CR 寄存器的 EOCALIE，校准完成后产生校准结束中断。

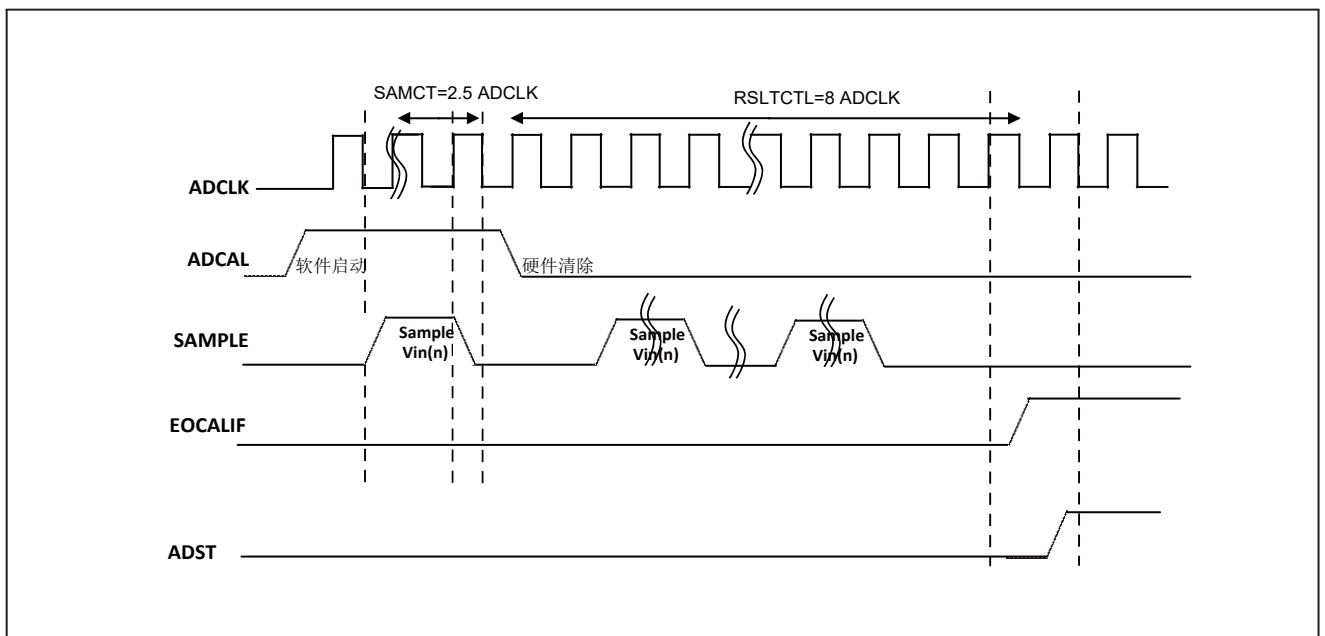


图 17-3 校准过程

17.6.9.2 寄存器写校准

通过 ADC_ADDDATA 寄存器读或写入校准因子，校准因子地址映射如下：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Res		FADDR						FDATA							
Type			rw						rw							

Bit	Field	Type	Reset	Description
15:14	Reserved		0x0	保留，始终读为 0
13 : 9	FADDR	rw	0x00	校准因子地址 1: dos 2: dcpm5 3: dcpm4 4: dcpm3 5: dcpm2 6: dcpm1 7: dcpn4 8: dcpn3 9: dcpn2 10: dcpn1 11: dcpk3 12: dcm5 13: dcm4 14: dcm3 15: dcm2 16: dcm1 17: dcnm0
8 : 0	FDATA	rw	0x00	校准因子 dos~dcnm0 值。

校准因子是由 17 个 9 位数组成，分别是 dos、dcpm5、dcpm4、dcpm3、dcpm2、dcpm1、dcpn4、dcpn3、dcpn2、dcpn1、dcpk3、dcm5、dcm4、dcm3、dcm2、dcm1、dcnm0，各校准因子对应的地址参考表格。

校准因子需要分 17 次读或写入。读校准因子需要将因子的对应的地址写入 ADC_ADDDATA.FADDR 且 ADC_ADDDATA.FDATA 为 0，再读 ADC_ADDDATA 即可获取因子；写校准因子需要将因子的校准值及其对应的地址写入 ADC_ADDDATA 中即可完成一个校准因子写操作，访问 ADC_ADSTA_EXT.FREOC 位查看校准因子读取结束标志位。

例如：

读 dcnm0 校准因子，配置 FDATA=0x000、FADDR=0x11，即写 ADC_ADDDATA 为 0x2200，再读 ADC_ADDDATA 即可获取 dcnm0。

写 dcnm0 校准因子为 0x123, 配置 FDATA=0x123、FADDR=0x11, 即写 ADC_ADDDATA 为 0x2323。

17.6.10 通道转换方式

单端转换是指 0~18 通道与 ADC 内部参考电压 VREF 的差数字量化后输出比较结果。

差分转换是指 VIP（正端）与 VIN（负端）的差被数字量化后的输出比较结果。寄存器 ADC_ADCFG2 的 DC 配置中, 通道 0、2、4、6 为 VIP 端, 通道 1、3、5、7 为 VIN 端。例如: 配置 ADC_ADCFG2.DC 为 1, 那么必须将通道 0 和通道 1 接入需要比较的电压, 这样通道 0（比较器的正端）与通道 1（比较器的负端）进行比较, 通过软件配置开启转换通道 0（或 1）, 转换结束后从 ADC_DR0(或 ADC_DR1)读取通道 0 和通道 1 的比较结果。

伪差分转换是指 VIP（正端）与 VIN（负端）的差被数字量化后的结果, 伪差分转换比差分转换的转换多一个 ADC 时钟周期, 但第一次比较结果被丢弃。寄存器 ADC_ADCFG2 的 DC 和 PSDC 配置中, 通道 0、2、4、6 为 VIP 端, 通道 1、3、5、7 为 VIN 端。例如: 配置 ADC_ADCFG2.DC 和 ADC_ADCFG2.PSDC 为 1, 那么必须将通道 0 和通道 1 接入需要比较的电压, 这样通道 0（比较器的正端）与通道 1（比较器的负端）进行比较, 通过软件配置开启转换通道 0（或 1）, 转换结束后从 ADC_DR0(或 ADC_DR1)读取通道 0 和通道 1 的比较结果。

以上三种转换方式支持普通通道转换、任意通道、注入通道转换。

17.6.11 通道转换

17.6.11.1 普通通道转换

17.6.11.1.1 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 3.5 个 ADC 时钟后清除。
- A/D 转换完成时, SAR 转换结果 SAR_DATA 将存储于数据寄存器 ADC_ADDDATA 和 ADC_ADDRn 中。
- A/D 转换结束时, 状态寄存器 ADC_ADSTA.EOSIF 位置 1, 若此时控制寄存器 ADC_ADCR.EOSIE 位置 1, 将产生 AD 转换结束中断请求。
- A/D 转换期间, ADC_ADCR.ADST 位保持为 1, A/D 通道转换结束以后, ADC_ADCR.ADST 位自动硬件清 0, A/D 转换器进入空闲状态。

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

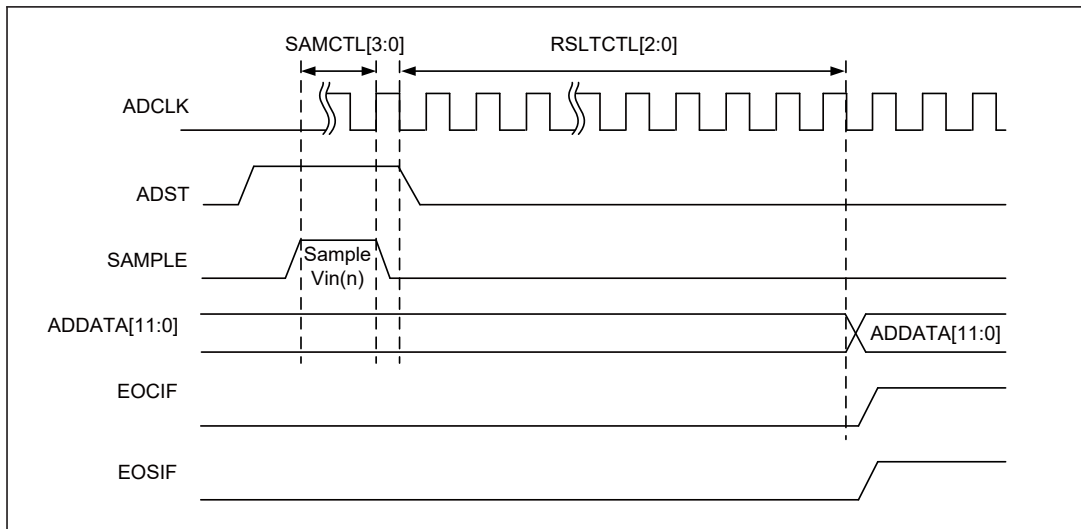


图 17-4 单次转换模式时序图

17.6.11.1.2 单周期扫描模式

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

- 触发或软件置位 `ADC_ADCR.ADST` 可以开启 A/D 转换，触发事件可配置触发延时，方向设置默认从最小序号通道到最大序号通道的 A/D 转换。
- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 `SAR_DATA` 将有序装载到相应通道的数据寄存器 `ADC_ADDDATA` 和 `ADC_ADDRn` 中，当前通道转换结束后 `ADC_ADSTA_EXT.EOCIF` 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 `ADC_ADSTA.EOSIF` 标志被置位，`ADC_ADSTA.EOSIF` 序列转换结束标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。
- 所有 A/D 通道转换结束后，`ADC_ADCR.ADST` 位自动硬件清 0，A/D 转换器会进入空闲状态。

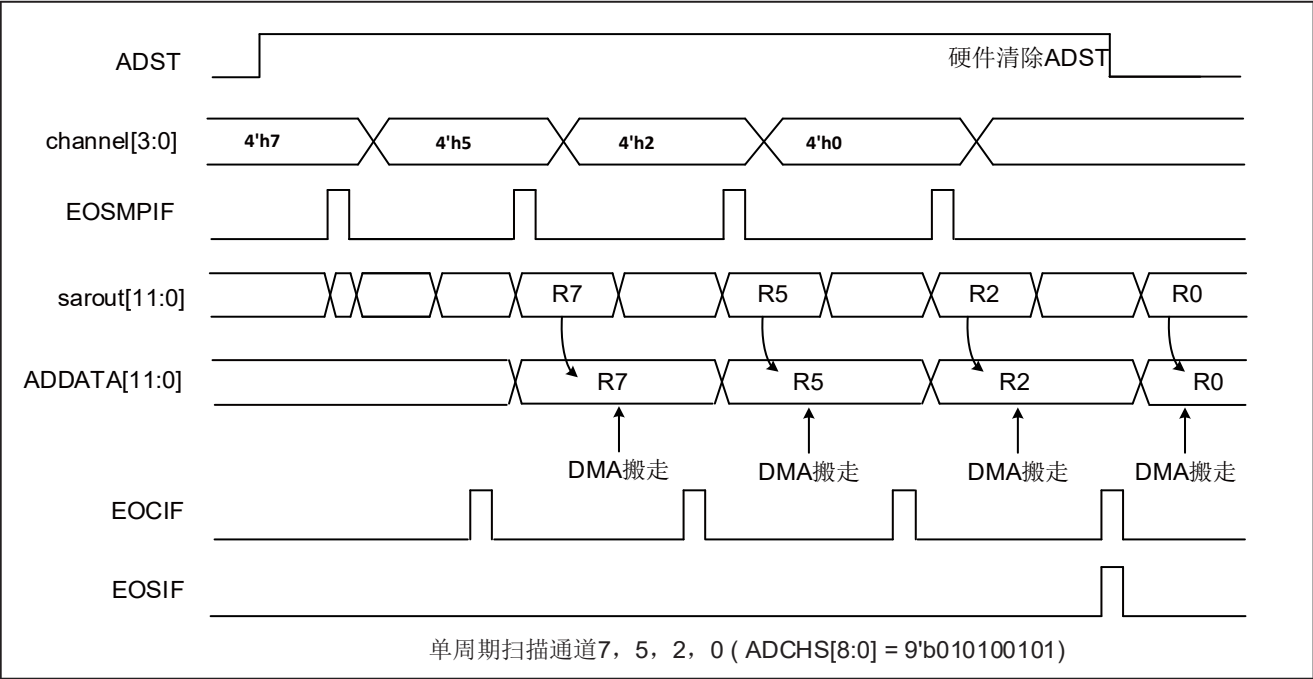


图 17-5 单周期扫描下使能通道转换时序图（通道方向从低到高）

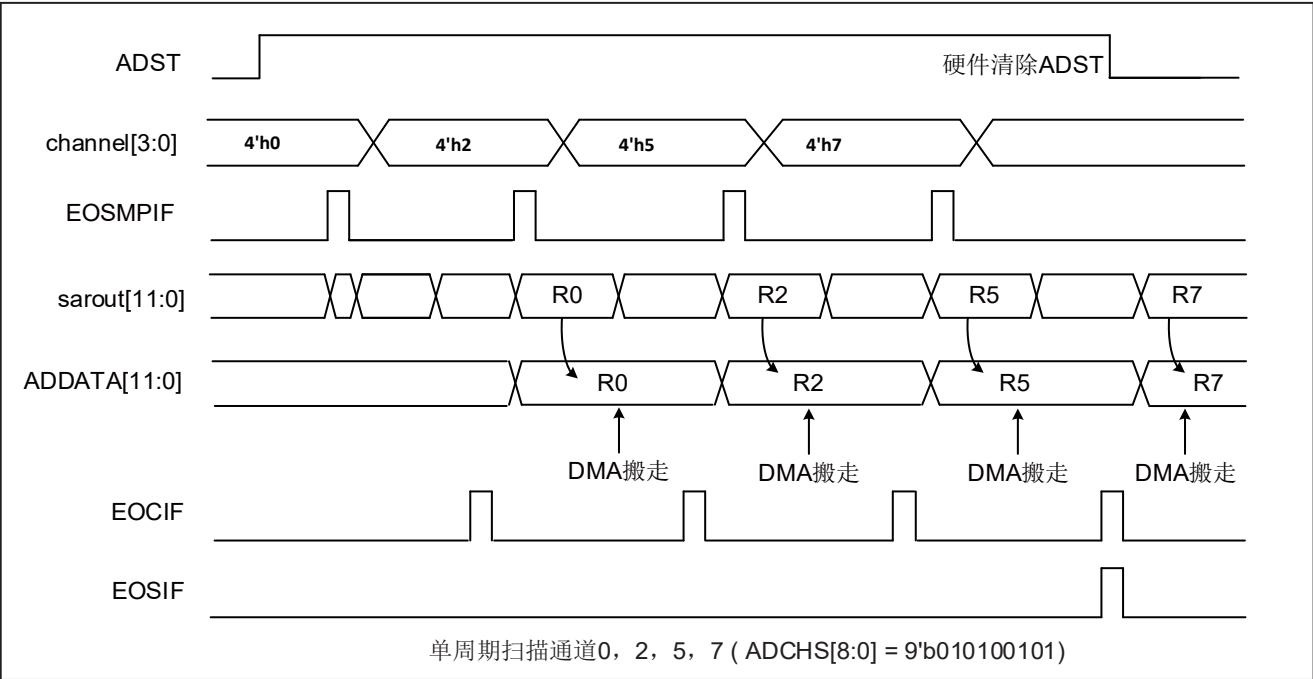


图 17-6 单周期扫描下使能通道转换时序图（通道方向从高到低）

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

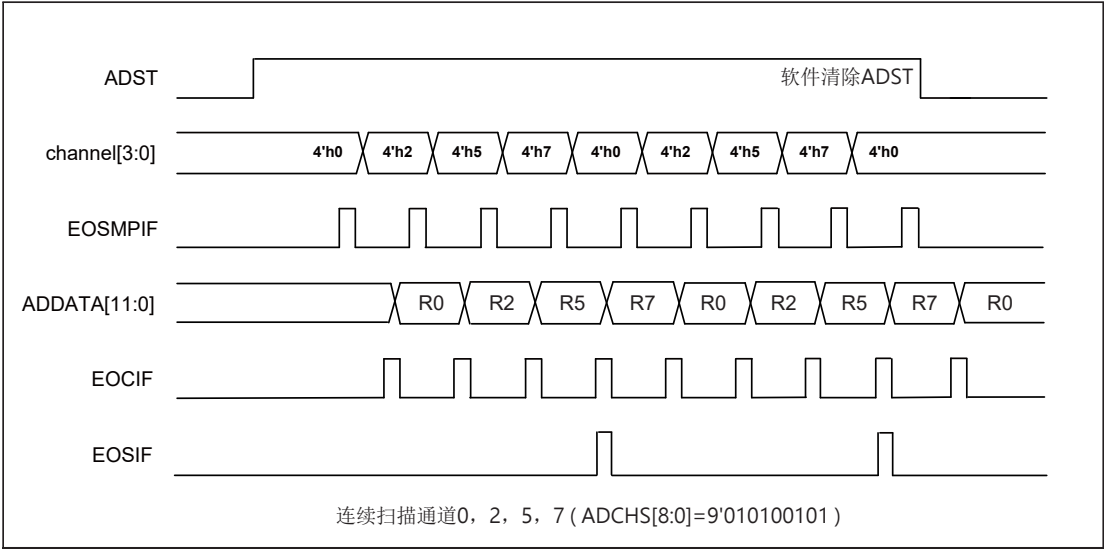


图 17-7 连续扫描模式使能通道转换时序图（通道方向由低到高）

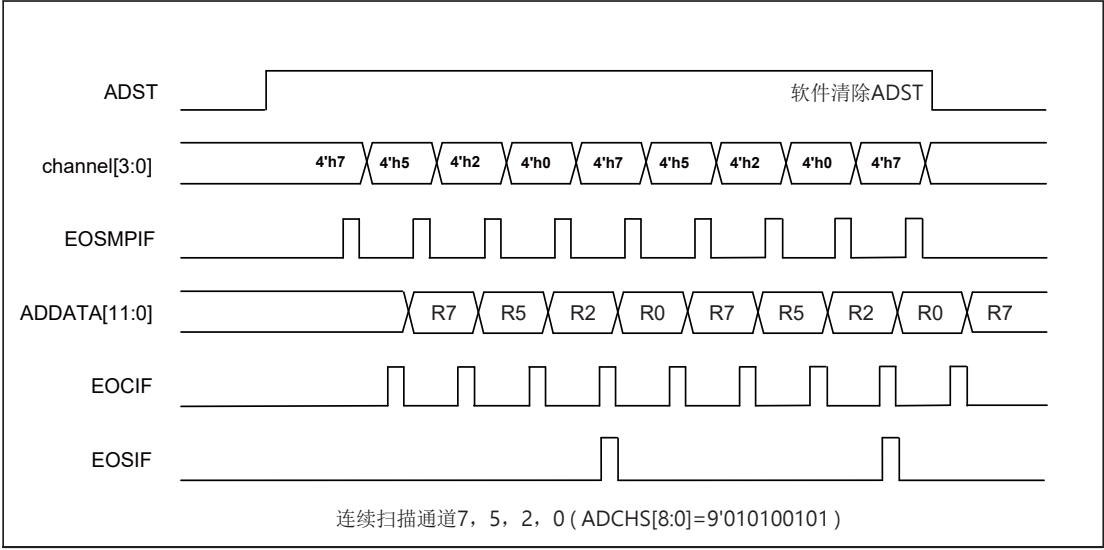


图 17-8 连续扫描模式使能通道转换时序图（通道方向由高到低）

17.6.11.2 任意通道转换

17.6.11.2.1 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 3.5 个 ADC 时钟后清除。
- A/D 转换结束后，A/D 转换结果 SAR_DATA 将装载到 ADC_ADDDATA 和 ADC_ADDR0，通道转换结束后 ADC_ADSTA.EOSIF 标志被置位，如果设置使能了转换结束中断，则产生转换结束中断请求。
- 所有 A/D 通道转换结束后，ADST 位将被硬件自动清除，A/D 转换器会进入空闲状态。

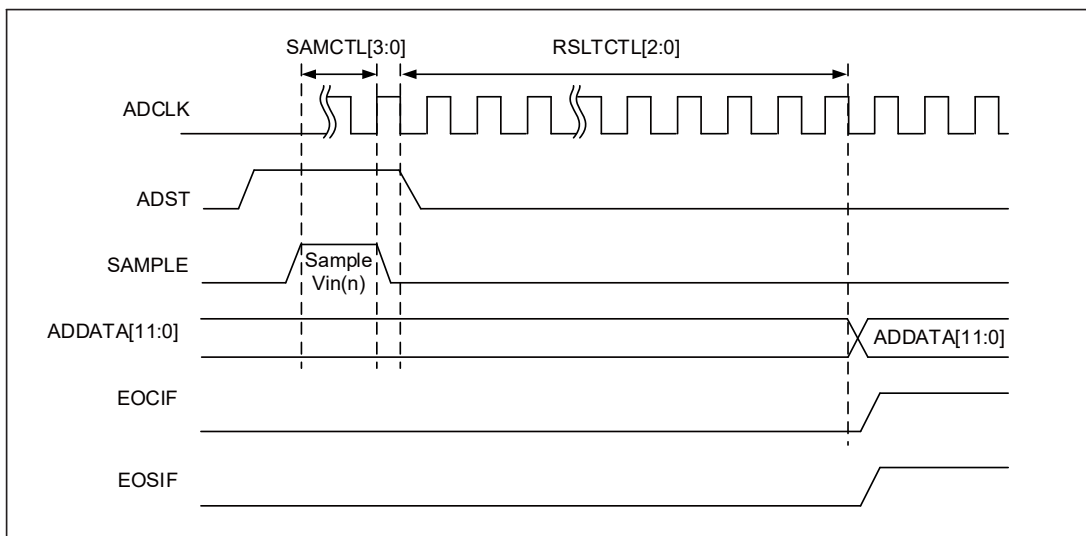


图 17-9 单次转换模式下通道转换时序图

17.6.11.2.2 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_SELO

到 CHANY_SEL18 任意配置，可以完全相同或不同。

- 通道输出按照转换通道配置有序输出。
- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，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。

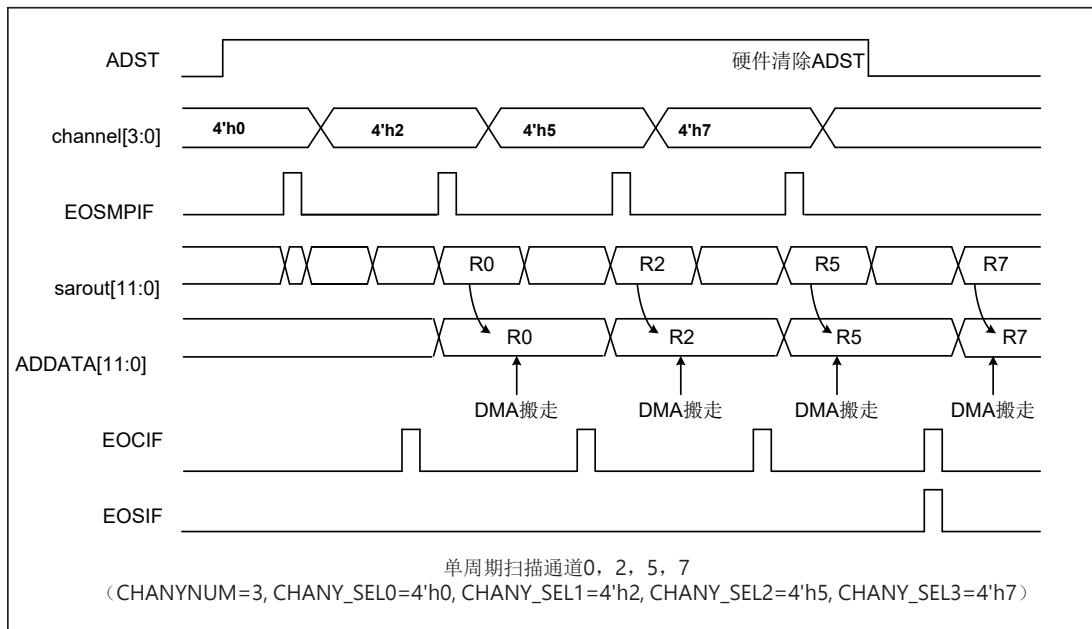


图 17-10 单周期扫描下通道转换时序图

17.6.11.2.3 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_SEL18 任意配置，可以完全相同或不同。
- 通道输出按照转换通道配置有序输出。

- 各通道的 A/D 转换结束后，A/D 转换结果 SAR_DATA 将有序装载到相应通道的数据寄存器 ADC_ADDDATA 和 ADC_ADDRn 中，当前通道转换结束后 ADC_ADSTA_EXT.EOCIF 标志被置位，如果设置转换结束中断，则产生通道转换结束中断请求。序列转换结束后 ADC_ADSTA.EOSIF 标志被置位，如果设置使能了序列转换结束中断，则产生序列转换结束中断请求。
- 只要 ADC_ADCR.ADST 位保持为 1，就可以持续进行 A/D 转换。当 ADC_ADCR.ADST 位被软件清除，当前 A/D 转换结束后停止，A/D 转换器进入空闲状态。
- 若在 A/D 转换过程中，用户更新 ADC_ANY_CFG、ADC_CHANY0、ADC_CHANY1 寄存器后，这些配置不会立即生效，等待所有通道转换结束后生效，即下一个扫描周期开始新的通道转换。

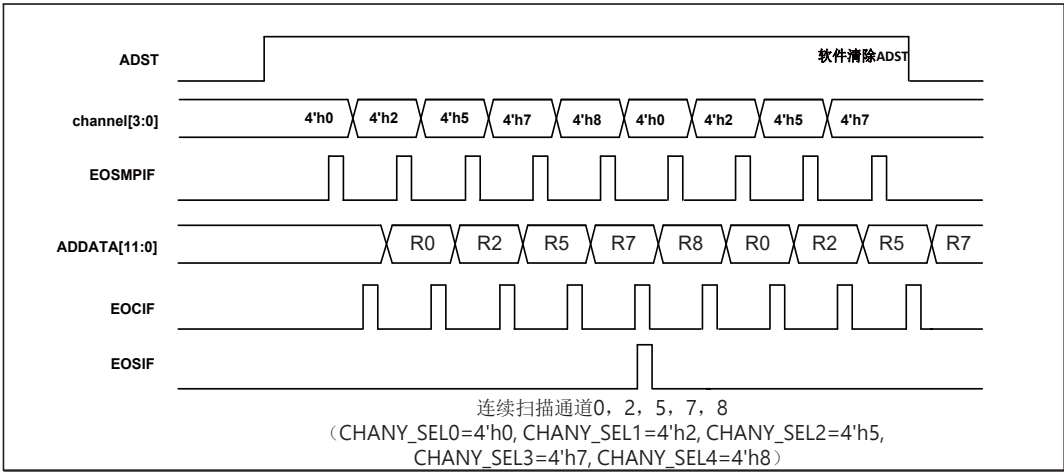


图 17-11 连续扫描模式下通道转换时序图

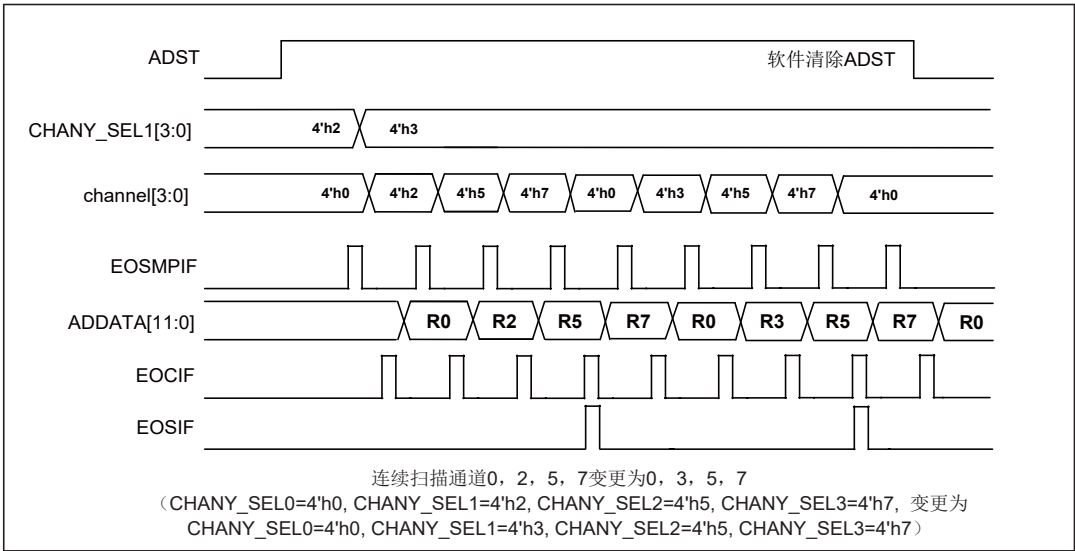


图 17-12 连续扫描模式下动态更新配置时序图

17.6.11.3 注入通道转换

17.6.11.3.1 自动注入通道转换

自动注入，可以扩展转换通道。将任意通道与注入通道配置后进行自动注入，那么可进行转换的通道最多可达 20 个。

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_SEL18 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_ADDDATA 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。

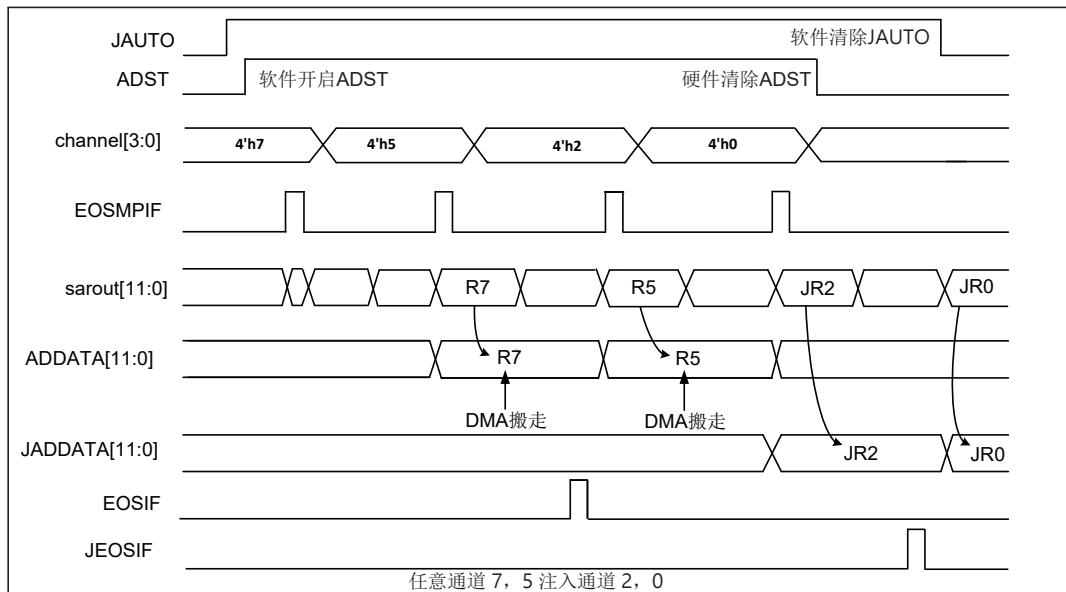


图 17-13 自动注入通道转换单周期扫描通道转换时序图

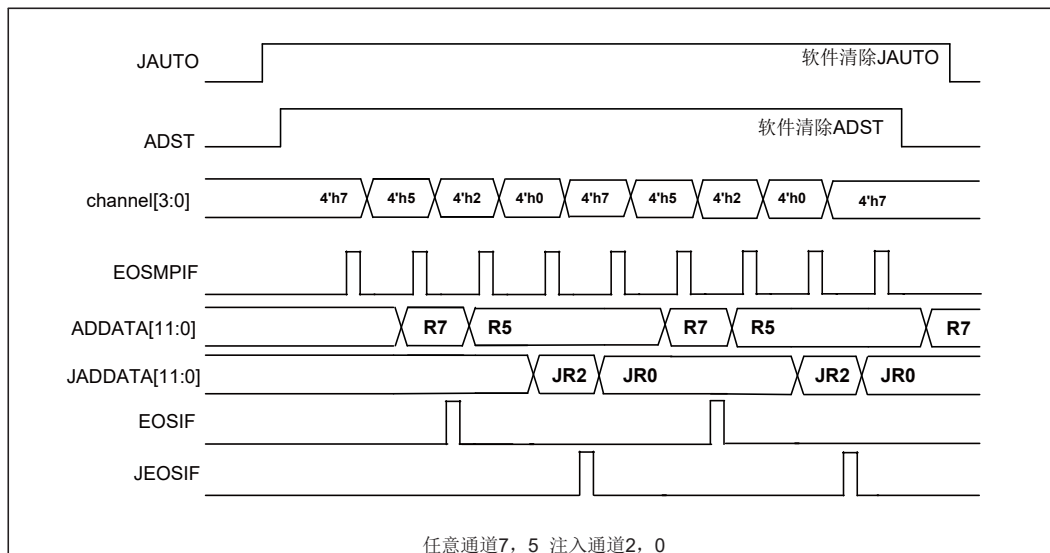


图 17-14 自动注入通道转换连续转换时序图

17.6.12 ADC 过采样

过采样是将多次转换结果取平均，平均结果最高能到 16bit。如下：

$$n=N-1$$

$$\text{Result}=1/M \times \sum \text{conversion}(T_n)$$

$$n=0$$

硬件能实现如下功能：取平均、数据率规约、SNR 改善、基本滤波。

ADC_CFGR2 寄存器的 OVFS[2:0]指定了过采样率 N;OVSS[3:0]指定了数据右移位数 M。

整个过采样过程不会改变配置，过采样期间的采样时间相同，N 次过采样结束后的结果保存在结果数

据寄存器，结果值是最高达 20bit 数据（256*12bit）经过右移后得到的 16bit 关键值。若结果超过 16bit，则将高位丢弃。

过采样支持的转换模式

- 支持任意模式单次、单周期、连续转换；支持自动注入转换
- 支持 ADC 软件、触发启动转换

过采样转换时间为 $N * T_{conv}$ ，状态寄存器中标志位含义如下

- 每一次转换过程的采样结束，EOSAMPIF 标志位置起
- N 次转换过程结束，EOCIF 标志位置起
- 连续的转换结束，EOSIF 标志位置起。（ $N * CHANY\_NUM$ 转换结束）

17.6.12.1 任意通道单次模式过采样过程

配置 ADC_ADCR 和 ADC_CFGR2 寄存器位：配置 ADMODE，置位 ROVSE，配置过采样率 OVSR 和过采样移位 OVSS。

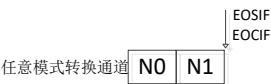


图 17-15 单次过采样模式下通道转换图

17.6.12.2 任意通道单周期模式过采样过程

配置 ADC_ADCR 和 ADC_CFGR2 寄存器位：配置 ADMODE，置位 ROVSE，配置过采样率 OVSR 和过采样移位 OVSS。

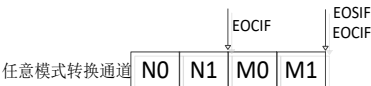


图 17-16 单周期过采样模式下通道转换图

17.6.12.3 任意通道连续模式过采样过程

配置 ADC_ADCR 和 ADC_CFGR2 寄存器位：配置 ADMODE，置位 ROVSE，配置过采样率 OVSR 和过采样移位 OVSS。



图 17-17 连续过采样模式下通道转换图

17.6.12.4 自动注入模式过采样过程

配置 ADC_ANY_CR 寄存器位和 ADC_CFGR2 寄存器位：只支持同时置位 JAUTO、ROVSE 和 JOVSE，配置过采样率 OVSR 和过采样移位 OVSS。

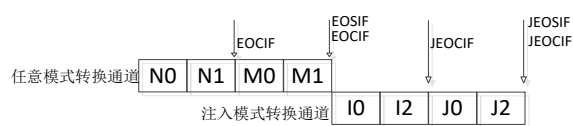


图 17-18 单周期自动注入过采样模式下通道转换图

17.6.12.5 过采样触发转换模式

17.6.12.5.1 过采样触发复位转换

过采样非触发可以配置 ADC_CFGR2 寄存器位：置位 ROVSE，复位 TROVS;配置过采样率 OVSR 和过采样移位 OVSS。

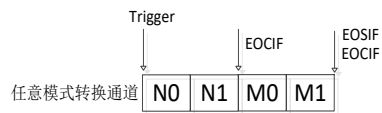


图 17-19 单周期过采样模式触发复位方式下通道转换图

17.6.12.5.2 过采样触发置位转换

过采样触发可以配置 ADC_CFGR2 寄存器位：置位 ROVSE 和 TROVS。配置过采样率 OVSR 和过采样移位 OVSS。

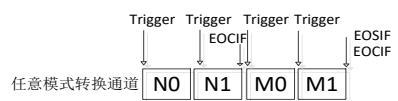


图 17-20 单周期过采样模式触发置位方式下通道转换图

17.6.12.6 过采样转换模式配置

规则过采样 ROVSE	注入过采样 JOVSE	触发规则模式 TROVS	说明
1	0	0	规则模式
1	0	1	触发规则模式
1	1	0	注入和规则模式
1	1	1	不支持

17.6.13 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 Mindswitch 模块配置可以选择任意和注入通道转换外部触发源，具体参考 Mindswitch 模块的描述。

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。

17.6.14 模拟看门狗

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

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

如果控制寄存器 ADC_ADCR.AWDIE 被置位，将产生中断请求。

17.6.15 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：

$$T(^{\circ}\text{C}) = ((\text{Value} * V_{\text{DDA}} - V_{25} * 3300) / (4096 * \text{Avg\_Slope})) + 25$$

V_{DDA} : ADC 当前采样时的 VDDA 电压，单位 mV

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

Value: ADC 的转换结果数据

Avg_Slope: 温度与电压曲线的平均斜率（以 mV/℃ 或 μV/℃ 表示）

Avg_Slope 的典型值请参考数据手册温度传感器章节。

17.6.16 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 Vref，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：

$$3.3/4096 = V_{\text{ref}} / \text{ADC\_ADDR}(\text{VSensor})$$

$$V_{\text{ref}} = (\text{ADC\_ADDR} * 3.3) / 4096$$

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 0x1FFFF7E0，用户可以根据该校准值反向推出当前系统模拟参考电压值，计算公式如下：

$$V_{\text{ref\_calc}} / \text{ADC\_ADDR}(\text{VSensor}) = V_{\text{DDA}} / 4096$$

$$V_{\text{DDA}} = (V_{\text{ref\_calc}} * 4096) / \text{ADC\_ADDR}(\text{VSensor})$$

$V_{\text{ref\_calc}}$: 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。

17.7 寄存器描述

表 17-1 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDDATA	A/D 数据寄存器	0x00000000
0x04	ADC_ADCFG	A/D 配置寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x08	ADC_ADCR	A/D 控制寄存器	0x00000000
0x0C	ADC_ADCHS	A/D 通道选择寄存器	0x00000000
0x10	ADC_ADCMPR	A/D 模拟看门狗比较寄存器	0x00000000
0x14	ADC_ADSTA	A/D 状态寄存器	0x00000000
0x18~0x54	ADC_ADDR 0~15	A/D 通道数据寄存器	0x00000000
0x58	ADC_ADSTA_EXT	A/D 扩展状态寄存器	0x00000000
0x5C	ADC_CHANY0	A/D 任意通道通道选择寄存器 0	0x00000000
0x60	ADC_CHANY1	A/D 任意通道通道选择寄存器 1	0x00000000
0x64	ADC_ANY_CFG	A/D 任意通道配置寄存器	0x00000000
0x68	ADC_ANY_CR	A/D 任意通道控制寄存器	0x00000000
0x6C	ADC_ADCFG2	ADC 配置寄存器	0x003E000
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
0x78	ADC_SMPR3	A/D 采样配置寄存器 3	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

17.7.1 ADC_ADDATA A/D 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.									CHANN ELSEL H	VAILD	OVERR UN	CHANNELSELL			
									r	r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
r															

Bit	Field	Description
31:23	Reserved	保留，必须保持复位值。
22	CHANNELSELH	当前数据所对应的通道高位（Channel selection For High Bits）
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。

Bit	Field	Description
20	OVERRUN	数据覆盖标志位（只读）（Overrun Flag） 1： DATA[11:0] 上一次转换数据被覆盖 0： DATA[11:0] 上一次转换数据未被覆盖 若数据装载至 DATA[11:0] 数据寄存器位之前，若上一次 DATA[11:0]没有被读取，则 OVERRUN 将置位；读 ADC_ADDDATA 寄存器后该位自动清除。
19:16	CHANNELSELL	当前数据所对应的通道低位（Channel selection For Low Bits） CHANNELSEL={CHANNELSELH,CHANNELSELL}: 00000 = 通道 0 的转换数据 00001 = 通道 1 的转换数据 00010 = 通道 2 的转换数据 00011 = 通道 3 的转换数据 00100 = 通道 4 的转换数据 00101 = 通道 5 的转换数据 00110 = 通道 6 的转换数据 00111 = 通道 7 的转换数据 01000 = 通道 8 的转换数据 01001 = 通道 9 的转换数据 01010 = 通道 10 的转换数据 01011 = 通道 11 的转换数据 01100 = 通道 12 的转换数据 01101 =通道 13 的转换数据 01110 =通道 14 的转换数据 01111 =通道 15 的转换数据 10000 = 通道 16 的转换数据 10001 = 通道 17 的转换数据 10010 = 通道 18 的转换数据 其他:无效
15:0	DATA	12 位 A/D 当前通道转换结果（Conversion Data） 软件选择对齐方式。

17.7.2 ADC_ADCFG A/D 配置寄存器

偏移地址：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															JAWDEN
															N
															RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						RSLTCTL			Res.			VSEN	TSEN	AWDEN	ADEN
						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: 10	Reserved	保留，必须保持复位值
9: 7	RSLTCTL	ADC 单端和差分转换数据分辨率（resolution） 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 ADC 伪差分转换数据分辨率： 000:13 位有效 001:12 位有效 010:11 位有效 011:10 位有效 100:9 位有效
6: 4	Reserved	保留，必须保持复位值
3	VSEN	内部参考电压使能（Voltage Sensor Enable） 仅 ADC2 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 ADC2 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

17.7.3 ADC_ADCCR A/D 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.			CMPCH	EOCIE	EOSMPI	TRG_EDGE		Res.		TRGSHIFT			Res.		SCANDI
			H		E						R				
			rw	rw	rw	rw				rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCHL				ALIGN	ADMD		ADST	Res.				DMAEN	TRGEN	AWDIE	EOSIE
rw				rw	rw		rw					rw	rw	rw	rw

Bit	Field	Description
31:29	Reserved	保留，必须保持复位值
28	CMPCHH	模拟看门狗通道选择高位（Compare Channel Selection For Analog Watchdog For High Bits）
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 若 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	触发边沿选择 10:上沿触发 其他:保留 注意：该寄存器位必须配置为“10”
23:22	Reserved	保留，必须保持复位值。
21 : 19	TRGSHIFT	外部触发延时采样（External trigger shift sample） 在触发信号产生后，延时 N 个 PCLK2 的时钟周期再开始采样。 如果是触发扫描模式，其他通道是在上一个采样结束后立即开始。 0: 不延迟 1:4 个周期 2:16 个周期 3:32 个周期 4:64 个周期 5:128 个周期 6:256 个周期 7:512 个周期
18 : 17	Reserved	保留，必须保持复位值。
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	CMPCHL	模拟看门狗通道选择 (Compare Channel Selection For Analog Watchdog For Low Bits) CHANNEL={CHANNELH,CHANNELL} 00000 = 选择比较通道 0 转换结果作为比较对象 00001 = 选择比较通道 1 转换结果作为比较对象 00010 = 选择比较通道 2 转换结果作为比较对象 00011 = 选择比较通道 3 转换结果作为比较对象 00100 = 选择比较通道 4 转换结果作为比较对象 00101 = 选择比较通道 5 转换结果作为比较对象 00110 = 选择比较通道 6 转换结果作为比较对象 00111 = 选择比较通道 7 转换结果作为比较对象 01000 = 选择比较通道 8 转换结果作为比较对象 01001 = 选择比较通道 9 转换结果作为比较对象 01010 = 选择比较通道 10 转换结果作为比较对象 01011 = 选择比较通道 11 转换结果作为比较对象 01100 = 选择比较通道 12 转换结果作为比较对象 01101 = 选择比较通道 13 的转换数据作为比较对象 01110 = 选择比较通道 14 的转换数据作为比较对象 01111 = 选择比较通道 15 的转换数据作为比较对象 10000 = 选择比较通道 16 转换结果作为比较对象 10001 = 选择比较通道 17 转换结果作为比较对象 10010 = 选择比较通道 18 转换结果作为比较对象 11111 = 选择所有扫描通道转换数据作为比较对象
11	ALIGN	数据对齐格式 (Data Alignment) 1: 左对齐 0: 右对齐
10 : 9	ADMD	A/D 转换模式 (ADC Mode) 00: 单次转换 01: 单周期扫描 10: 连续扫描 11: 保留 当改变转换模式时, 软件要先清除 ADST 位。
8	ADST	A/D 转换开始 (ADC Start) 1: 转换开始 0: 转换结束或空闲状态 ADST 清除有下列两种方式: 在单次模式或者单周期模式下, 转换完成后 ADST 将硬件自动清除。 在连续扫描模式下, ADST 由软件或系统复位清除, 否则 A/D 转换将一直进行。
7: 4	Reserved	保留, 必须保持复位值。
3	DMAEN	DMA 使能 (Direct Memory Access Enable) 1: DMA 请求使能 0: DMA 请求禁止

Bit	Field	Description
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。

17.7.4 ADC_ADCHS A/D 通道选择寄存器

偏移地址: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHEN15	CHEN14	CHEN13	CHEN12	CHEN11	CHEN10	CHEN9	CHEN8	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bit	Field	Description
31:16	Reserved	保留, 必须保持复位值
15	CHEN15	模拟输入通道 15 使能 (Analog Input Channel 15 enable) 1: 使能 0: 禁止
14	CHEN14	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) (internal) 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

Bit	Field	Description
10	CHEN10	模拟输入通道 10 使能 (Analog Input Channel 10 Enable) 1: 使能 0: 禁止
9	CHEN9	模拟输入通道 9 使能 (Analog Input Channel 9 Enable) 1: 使能 0: 禁止
8	CHEN8	模拟输入通道 8 使能 (Analog Input Channel 8 Enable) 1: 使能 0: 禁止
7	CHEN7	模拟输入通道 7 使能 (Analog Input Channel 7 Enable) 1: 使能 0: 禁止
6	CHEN6	模拟输入通道 6 使能 (Analog Input Channel 6 Enable) 1: 使能 0: 禁止
5	CHEN5	模拟输入通道 5 使能 (Analog Input Channel 5 Enable) 1: 使能 0: 禁止
4	CHEN4	模拟输入通道 4 使能 (Analog Input Channel 4 Enable) 1: 使能 0: 禁止
3	CHEN3	模拟输入通道 3 使能 (Analog Input Channel 3 Enable) 1: 使能 0: 禁止
2	CHEN2	模拟输入通道 2 使能 (Analog Input Channel 2 Enable) 1: 使能 0: 禁止
1	CHEN1	模拟输入通道 1 使能 (Analog Input Channel 1 Enable) 1: 使能 0: 禁止
0	CHEN0	模拟输入通道 0 使能 (Analog Input Channel 0 Enable) 1: 使能 0: 禁止

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

2) 在 ADC_ADSTA.BUSY 无效时才能操作此寄存器。

17.7.5 ADC_ADCMPR A/D 模拟看门狗比较寄存器

偏移地址：0x10

复位值：0x00000000

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

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

Bit	Field	Description
31:28	Reserved	保留，必须保持复位值
27: 16	CMPHDATA	模拟看门狗比较数值高阈值（Compare High Threshold Data For Analog Watchdog） 看门狗比较阈值的高限
15:12	Reserved	保留，必须保持复位值。
11 : 0	CMPLDATA	模拟看门狗比较数值低阈值（Compare Low Threshold Data For Analog Watchdog） 看门狗比较阈值的低限

17.7.6 ADC_ADSTA A/D 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								CHANNELL				CHANN ELH	BUSY	AWDIF	EOSIF
r								r				r	r	rc_w1	rc_w1

Bit	Field	Description
31:20	OVERRUN	通道 0 ~ 11 的数据覆盖标志位（Overrun Flag）
19:8	VALID	通道 0 ~11 的数据有效标志位（Valid Flag）
7:4	CHANNELL	当前转换通道低位（Current Convert Channel For Low Bits） CHANNEL={CHANNELH,CHANNELL} 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	CHANNELH	当前转换通道高位（Current Convert Channel For High Bits） 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。
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

Bit	Field	Description
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。

17.7.7 ADC_ADDRn A/D 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.													VALID	OVERR	Reserved
													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。

17.7.8 ADC_ADSTA_EXT A/D 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.					FREOCIF	CALBUSY	EOCALIF	Res.		JBUSY	JEOSIF	JEOCIF	JEOSMPIF	EOCIF	EOSMPIF
					rc_w1	r	rc_w1			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
31:27	Reserved	保留，必须保持复位值。
26	FREOCIF	校准因子读写数据寄存器结束标志 1 = 校准因子读写完成 0 = 校准因子读写未完成
25	CALBUSY	CALBUSY: 校准模式忙/空闲 (Busy) 1 = A/D 校准忙碌 0 = A/D 校准空闲
24	EOCALIF	结束校准标志 (End of calibration interrupt flag) 1: 校准完成 0: 校准未完成 在校准完成后硬件置 1，软件清除。
23:22	Reserved	保留，始终读为 0
21	JBUSY	注入通道转换忙/空闲 (Busy/Idle Flag) 1: A/D 转换器注入通道忙碌 0: A/D 转换器注入通道空闲
20	JEOSIF	A/D 注入通道序列转换结束标志位 (End of Injected Sequence Flag) 该位由硬件在通道序列转换结束时置位，由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。
19	JEOCIF	A/D 注入通道转换结束标志位 (End of Injected Conversion Flag) 该位由硬件在通道转换结束时置位，由软件清除。 1: A/D 转换结束 0: A/D 转换未结束 该标志位写“1”清零。
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。

Bit	Field	Description
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: 通道 15 0100: 通道 14 0010: 通道 13 0001: 通道 12
3:0	VALID	通道的有效标志位 (Valid Flag) 1000: 通道 15 0100: 通道 14 0010: 通道 13 0001: 通道 12

17.7.9 ADC_CHANY0 A/D 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_SELL7				CHANY_SELL6				CHANY_SELL5				CHANY_SELL4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SELL3				CHANY_SELL2				CHANY_SELL1				CHANY_SELL0			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SELL7	任意通道转换序列第 7 次转换选择低位 (7th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL7={CHANY_SELH7,CHANY_SELL7}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道

Bit	Field	Description
27:24	CHANY_SELL6	任意通道转换序列第 6 次转换选择低位 (6th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL6={CHANY_SELH6,CHANY_SELL6}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
23:20	CHANY_SELL5	任意通道转换序列第 5 次转换选择低位 (5th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL5={CHANY_SELH5,CHANY_SELL5}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
19:16	CHANY_SELL4	任意通道转换序列第 4 次转换选择低位 (4th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL4={CHANY_SELH4,CHANY_SELL4}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
15:12	CHANY_SELL3	任意通道转换序列第 3 次转换选择低位 (3th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL3={CHANY_SELH3,CHANY_SELL3}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
11:8	CHANY_SELL2	任意通道转换序列第 2 次转换选择低位 (2th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL2={CHANY_SELH2,CHANY_SELL2}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
7:4	CHANY_SELL1	任意通道转换序列第 1 次转换选择低位 (1th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL1={CHANY_SELH1,CHANY_SELL1}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
3:0	CHANY_SELL0	任意通道转换序列第 0 次转换选择低位 (0th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL0={CHANY_SELH0,CHANY_SELL0}: 00000~10011: 可配置为通道 0~通道 18 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。

17.7.10 ADC_CHANY1 A/D 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_SELL15				CHANY_SELL14				CHANY_SELL13				CHANY_SELL12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SELL11				CHANY_SELL10				CHANY_SELL9				CHANY_SELL8			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL15	任意通道转换序列第 15 次转换选择低位 (15th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL15={CHANY_SELH15,CHANY_SELL15}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
27:24	CHANY_SELL14	任意通道转换序列第 14 次转换选择低位 (14th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL14={CHANY_SELH14,CHANY_SELL14}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
23:20	CHANY_SELL13	任意通道转换序列第 13 次转换选择低位 (13th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL13={CHANY_SELH13,CHANY_SELL13}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
19:16	CHANY_SELL12	任意通道转换序列第 12 次转换选择低位 (12th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL12={CHANY_SELH12,CHANY_SELL12}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
15:12	CHANY_SELL11	任意通道转换序列第 11 次转换选择低位 (11th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL11={CHANY_SELH11,CHANY_SELL11}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
11:8	CHANY_SELL10	任意通道转换序列第 10 次转换选择低位 (10th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL10={CHANY_SELH10,CHANY_SELL10}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道

Bit	Field	Description
7:4	CHANY_SELL9	任意通道转换序列第 9 次转换选择低位 (9th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL9={CHANY_SELH9,CHANY_SELL9}: 00000~10011: 可配置为通道 0~通道 18 中的任意一个通道为任意通道转换的输入通道
3:0	CHANY_SELL8	任意通道转换序列第 8 次转换选择低位 (8th Conversion Select for Any Channel sequence For Low Bits) CHANY_SEL8={CHANY_SELH8,CHANY_SELL8}: 00000~10011: 可配置为通道 0~通道 18 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。

17.7.11 ADC_ANY_CFG A/D 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.								CHANY _SELH1 5	CHANY _SELH1 4	CHANY _SELH1 3	CHANY _SELH1 2	CHANY _SELH1 1	CHANY _SELH1 0	CHANY _SELH9	CHANY _SELH8
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY _SELH7	CHANY _SELH6	CHANY _SELH5	CHANY _SELH4	CHANY _SELH3	CHANY _SELH2	CHANY _SELH1	CHANY _SELH0	Res.				CHANY_NUM			
												rw			

Bit	Field	Description
31:24	Reserved	保留，必须保持复位值
23	CHANY_SELH15	任意通道转换序列第 15 次转换选择高位 (15th Conversion Select for Any Channel sequence For High Bits)
22	CHANY_SELH14	任意通道转换序列第 14 次转换选择高位 (14th Conversion Select for Any Channel sequence For High Bits)
21	CHANY_SELH13	任意通道转换序列第 13 次转换选择高位 (13th Conversion Select for Any Channel sequence For High Bits)
20	CHANY_SELH12	任意通道转换序列第 12 次转换选择高位 (12th Conversion Select for Any Channel sequence For High Bits)
19	CHANY_SELH11	任意通道转换序列第 11 次转换选择高位 (11th Conversion Select for Any Channel sequence For High Bits)

Bit	Field	Description
18	CHANY_SELH10	任意通道转换序列第 10 次转换选择高位 (10th Conversion Select for Any Channel sequence For High Bits)
17	CHANY_SELH9	任意通道转换序列第 9 次转换选择高位 (9th Conversion Select for Any Channel sequence For High Bits)
16	CHANY_SELH8	任意通道转换序列第 8 次转换选择高位 (8th Conversion Select for Any Channel sequence For High Bits)
15	CHANY_SELH7	任意通道转换序列第 7 次转换选择高位 (7th Conversion Select for Any Channel sequence For High Bits)
14	CHANY_SELH6	任意通道转换序列第 6 次转换选择高位 (6th Conversion Select for Any Channel sequence For High Bits)
13	CHANY_SELH5	任意通道转换序列第 5 次转换选择高位 (5th Conversion Select for Any Channel sequence For High Bits)
12	CHANY_SELH4	任意通道转换序列第 4 次转换选择高位 (4th Conversion Select for Any Channel sequence For High Bits)
11	CHANY_SELH3	任意通道转换序列第 3 次转换选择高位 (3th Conversion Select for Any Channel sequence For High Bits)
10	CHANY_SELH2	任意通道转换序列第 2 次转换选择高位 (2th Conversion Select for Any Channel sequence For High Bits)
9	CHANY_SELH1	任意通道转换序列第 1 次转换选择高位 (1th Conversion Select for Any Channel sequence For High Bits)
8	CHANY_SELH0	任意通道转换序列第 0 次转换选择高位 (0th Conversion Select for Any Channel sequence For High Bits)
7:4	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): 00000: CHANY_SEL0 通道 00001: CHANY_SEL0~CHANY_SEL1 通道 00010: CHANY_SEL0~CHANY_SEL2 通道 00011: CHANY_SEL0~CHANY_SEL3 通道 00100: CHANY_SEL0~CHANY_SEL4 通道 00101: CHANY_SEL0~CHANY_SEL5 通道 00110: CHANY_SEL0~CHANY_SEL6 通道 00111: CHANY_SEL 0~CHANY_SEL7 通道 01000: CHANY_SEL0~CHANY_SEL8 通道 01001: CHANY_SEL0~CHANY_SEL9 通道 01010: CHANY_SEL0~CHANY_SEL10 通道 01011: CHANY_SEL0~CHANY_SEL11 通道 01100: CHANY_SEL0~CHANY_SEL12 通道 01101: CHANY_SE 0~CHANY_SEL13 通道 01110: CHANY_SEL0~CHANY_SEL14 通道 01111: CHANY_SEL0~CHANY_SEL15 通道 其它: 无效

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_NUM 影子寄存器。在 ADC 工作期间，若更

改 ADC_NUM 的值，只会更新到其影子寄存器中，在 ADC 转换最后一个通道时，影子寄存器的值才会更新至 ADC_NUM，实现通道动态切换。

17.7.12 ADC_ANY_CR A/D 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.										EOCALIE	ADCAL	Res.		JTRG_EDGE	
										rw	rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTRGSHIFT			Res.					JTRGEN	JADST	JAUTO	JEOSIE	JEOCIE	JEOSMPIE	JCEN	CHANY_MDEN
rw			rw					rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description	
31:22	Reserved	保留，必须保持复位值	
21	EOCALIE	自校准结束中断使能（End of calibration interrupt enable） 1：使能自校准结束中断 0：禁止自校准结束中断	
20	ADCAL	A/D 硬件自校准启动（A/D Calibration） 1：开始自校准过程 0：自校准过程完成	
19: 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) 10:上沿触发 其他:保留 注意：该寄存器位必须配置为“10”	

Bit	Field	Description	
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	
12:8	Reserved	保留，必须保持复位值。	
7	JTRGEN	注入通道转换的外部硬件触发源使能（External Trigger Enable For Injected Channels） 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	
6	JADST	注入通道转换的 A/D 转换开始（ADC Start for Injected Channels）：先开启 JCEN 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 JADST 清除有下列两种方式： 在注入通道转换完成后，JADST 将被硬件自动清除； JCEN 为 0 时 JADST 被清除；系统复位时 JADST 被清除。	
5	JAUTO	自动注入转换（Automatic Injected Conversion） 1：开启自动注入转换 0：关闭自动注入转换	
4	JEOSIE	A/D 注入通道序列转换结束中断使能（Interrupt Enable for JEOS） 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 如果 JEOSIE 置位，通道序列 A/D 转换结束后产生中断请求。	
3	JEOCIE	A/D 注入通道转换结束中断使能（Interrupt Enable for JEOC） 1：使能 A/D 转换结束中断 0：禁止 A/D 转换结束中断 如果 JEOCIE 置位，A/D 转换结束后产生中断请求。	

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。

17.7.13 ADC_ADCFG2 ADC 配置寄存器

偏移地址: 0x6C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res						TROVS	OVSS				OVSR			JOVSE	ROVSE
						rw	rw				rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PSDC					Res.	DC					Res.	ADCSR EF	ADCCR	CORREN
	rw						rw						rw	rw	rw

Bit	Field	Description
31:26	Reserved	保留, 始终读为 0
25	TROVS	触发过采样 (Trigger Oversampling Select) 1: 每个通道得每次过采样转换都需要一次触发 0: 所有过采样转换完成只需要一次触发 注: ADC 转换过程空闲可配

Bit	Field	Description
24: 21	OVSS	过采样移位(Oversampling Shift Bits) 0000: 不移位 0001: 移 1 位 0010: 移 2 位 0011: 移 3 位 0100: 移 4 位 0101: 移 5 位 0110: 移 6 位 0111: 移 7 位 1000: 移 8 位 注: ADC 转换过程空闲可配
20: 18	OVSR	过采样率(Oversampling Ratio) 000:2 001:4 010:8 011:16 100:32 101:64 110:128 111:256 注: ADC 转换过程空闲可配
17	JOVSE	注入模式过采样使能(Injected Oversampler Enable) 1: 注入模式过采样使能 0: 注入模式过采样无效 注: ADC 转换过程空闲可配
16	ROVSE	规则任意模式过采样使能(Regular Oversampler Enable) 1: 任意模式过采样使能 0: 任意模式过采样无效 注: ADC 转换过程空闲可配
15	Reserved	保留, 始终读为 0
14:10	PSDC	伪差分配置选择(Pseudo Differential Conversion) 00001: 通道 0 和 1 为一组伪差分通道 00010: 通道 2 和 3 为一组伪差分通道 00100: 通道 4 和 5 为一组伪差分通道 01000: 通道 6 和 7 为一组伪差分通道 其他: 保留 注: 使用伪差分转换, PSDC 与 DC 必须都配置
9	Reserved	保留, 始终读为 0

Bit	Field	Description
8:4	DC	差分配置选择(Differential Conversion) 00001: 通道 0 和 1 为一组差分通道 00010: 通道 2 和 3 为一组差分通道 00100: 通道 4 和 5 为一组差分通道 01000: 通道 6 和 7 为一组差分通道 其他: 保留
3	Reserved	保留, 始终读为 0
2	ADCSREF	单端参考电压选择 (ADC Select Reference Voltage) 1: VREF 选择内部参考电压 0: VREF 选择外部参考电压
1	ADCCR	ADC 低功耗控制(ADC Control) 1: 使能低功耗 0: 禁止低功耗
0	Reserved	保留, 始终读为 0

注: 在任意通道模式, 且单周期 I 连续扫描模式下, 关闭 ADC 时, 必须先禁止 ADC_ADCR 的 ADST 位, 然后判断 ADC_ADSTA 的 BUSY 位是否为 0, 即等到 ADC 转换完成, 再禁止 ADC_ANY_CR 的 CHANY_MDEN 位。

17.7.14 ADC_SMPR1 A/D 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	Type	Reset	Description
31 : 0	SAMP7~ SAMP0	rw	0x00	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。 0000: 3.5 周期 0001: 4.5 周期 0010: 5.5 周期 0011: 6.5 周期 0100: 7.5 周期 0101: 11.5 周期 0110: 13.5 周期 0111: 15.5 周期 1000: 19.5 周期 1001: 29.5 周期 1010: 39.5 周期 1011: 59.5 周期 1100: 79.5 周期 1101: 119.5 周期 1110: 159.5 周期 1111: 240.5 周期 其它: 保留

17.7.15 ADC_SMPR2 A/D 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				Res.				SAMP12			
rw				rw								rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP11				SAMP10				SAMP9				SAMP8			
rw				rw				rw				rw			

Bit	Field	Description
31 : 0	SAMP15~ SAMP8	选择通道 8~15 的采样时间选择 (Channel Sample Time Selection) 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 0000: 3.5 周期 0001: 4.5 周期 0010: 5.5 周期 0011: 6.5 周期 0100: 7.5 周期 0101: 11.5 周期 0110: 13.5 周期 0111: 15.5 周期 1000: 19.5 周期 1001: 29.5 周期 1010: 39.5 周期 1011: 59.5 周期 1100: 79.5 周期 1101: 119.5 周期 1110: 159.5 周期 1111: 240.5 周期 其它: 保留

17.7.16 ADC_SMPR3 A/D 采样配置寄存器 3

偏移地址: 0x78

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.				SAMP18				SAMP17				SAMP16			
				rw				rw				rw			

Bit	Field	Description
31:12	Reserved	保留, 始终读为 0

Bit	Field	Description
11 : 0	SAMP18~ SAMP16	选择通道 16~18 的采样时间选择 (Channel Sample Time Selection) 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 0000: 3.5 周期 0001: 4.5 周期 0010: 5.5 周期 0011: 6.5 周期 0100: 7.5 周期 0101: 11.5 周期 0110: 13.5 周期 0111: 15.5 周期 1000: 19.5 周期 1001: 29.5 周期 1010: 39.5 周期 1011: 59.5 周期 1100: 79.5 周期 1101: 119.5 周期 1110: 159.5 周期 1111: 240.5 周期 其它: 保留

17.7.17 ADC_JOFRn (n=0~3) A/D 注入通道数据补偿寄存器

偏移地址: 0x7C~0x88

复位值: 0x00000000

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

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

17.7.18 ADC_JSQR A/D 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) 00000~10011: 可配置为通道 0~通道 18 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) 00000~10011: 可配置为通道 0~通道 18 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) 00000~10011: 可配置为通道 0~通道 18 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) 00000~10011: 可配置为通道 0~通道 18 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。

17.7.19 ADC_JADDATA A/D 注入数据寄存器

偏移地址：0x90

复位值：0x00000000

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

Res.									JVALID	JOVER RUN	JCHANNELSEL				
									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_JADDDATA 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_JADDDATA 寄存器硬件自动清除。
20:16	JCHANNELSEL	显示当前数据所对应的注入通道（Injected Channel selection） JCHANNELSEH 和 JCHANNELSEL 选择通道： 00000 = 通道 0 的转换数据 00001 = 通道 1 的转换数据 00010 = 通道 2 的转换数据 00011 = 通道 3 的转换数据 ... 10001 = 通道 17 的转换数据 10010 = 通道 18 的转换数据 其他：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。

17.7.20 ADC_JDRn (n=0~3) A/D 注入通道数据寄存器

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（Overflow 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。

18 DAC 数字/模拟转换器

18.1 DAC 简介

数字 / 模拟转换器（ DAC ）包含了两个独立的 12 位数字输入通道和 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，也能配置成与 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。

18.2 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 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
- 既可软件触发转换，也可以由外部信号触发转换

DAC 单个通道的功能框图如下：

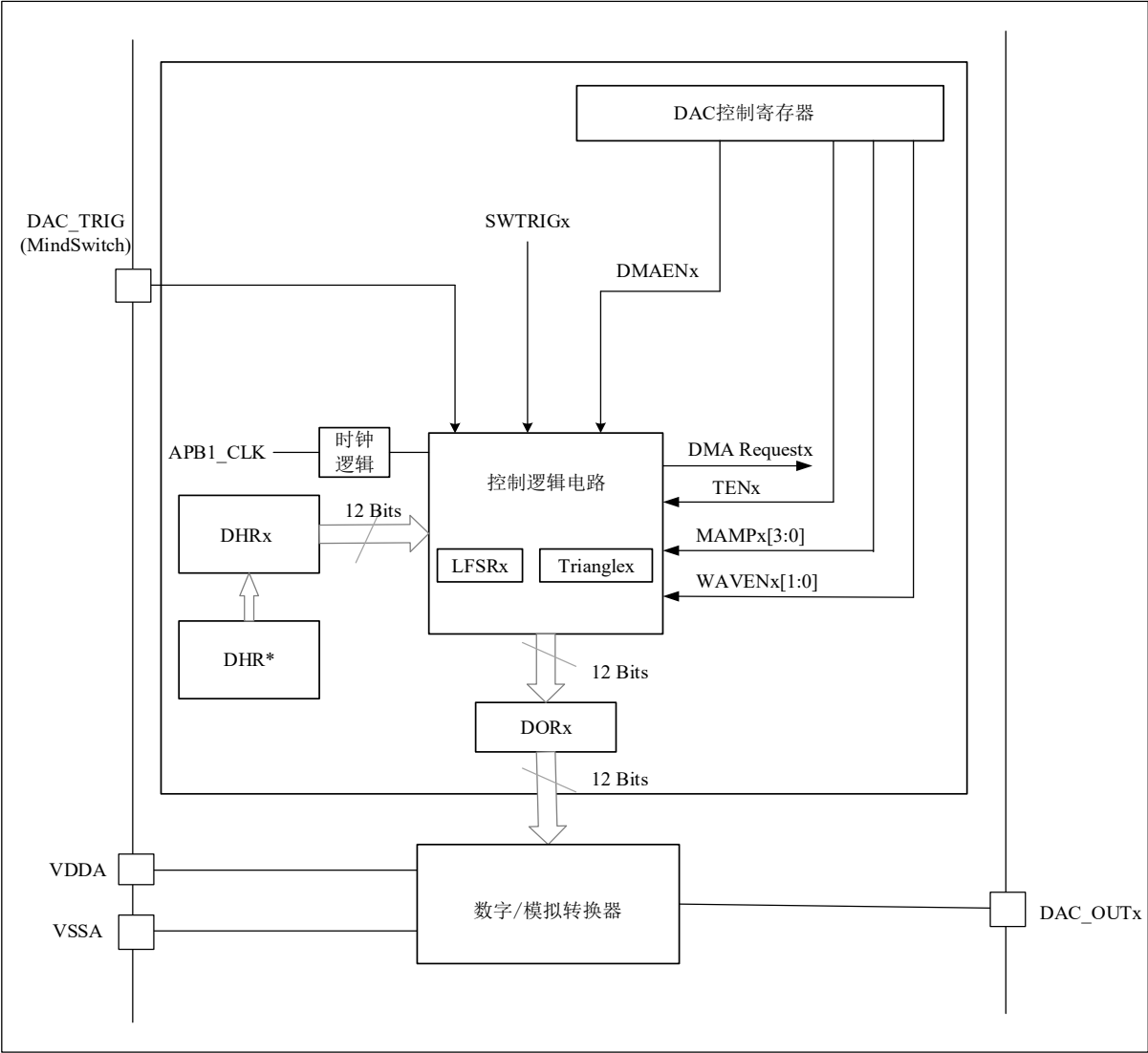


图 18-1 DAC 通道功能框图

相关管脚的描述如下。

表 18-1 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_TRIG	输入	来自 MindSwitch 的外部触发源
DAC_OUTx	模拟输出信号	DAC 通道 x 的模拟输出

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。

18.3 功能描述

18.3.1 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）模拟电路的供电。经过一段启动时间 t_{wakeUp} 后，DAC 通道 x 即被使能。

DAC 通道 x 的数字部分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。

18.3.2 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 通道输出缓存可以通过设置 DAC_CR 寄存器的相应位 BOFFx 来使能或者关闭。

18.3.3 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：

$$\text{DAC\_OUTx} = V_{\text{REF}} \times (\text{DAC\_DORx} / 4096)$$

18.3.4 DAC 触发源的选择

- 外部触发源选择

如果选择外部触发源，根据需求配置触发源（包括来自 timer、中断事件等）。由于 DAC 外部触发源 DAC_TRIG 来自于 MindSwitch，因此，具体配置方法参考 MindSwitch 章节和芯片特定配置章节。

- 软件触发

如果选择软件触发，当 SWTRIG 位置‘1’，转换开始，数据从寄存器 DAC_DHRx 装入寄存器 DAC_DORx 需要一个 APB1 时钟周期。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。

18.3.5 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 传输。

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。

18.3.6 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。

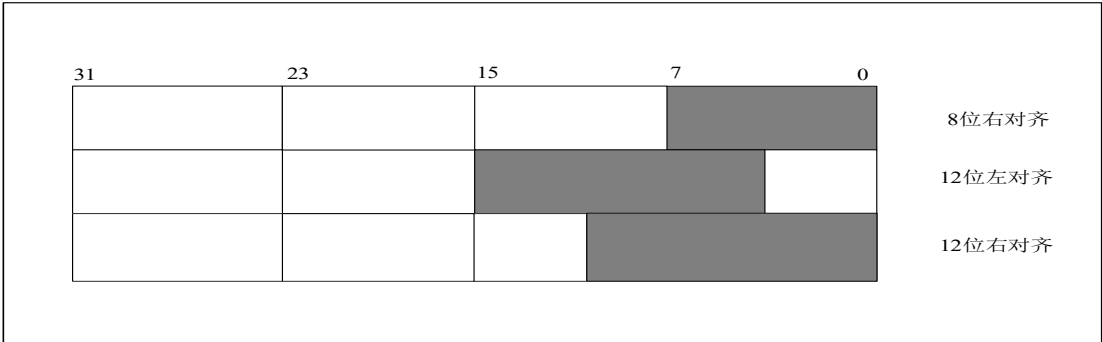


图 18-2 DAC 单通道模式的数据保持寄存器的不同数据格式

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。

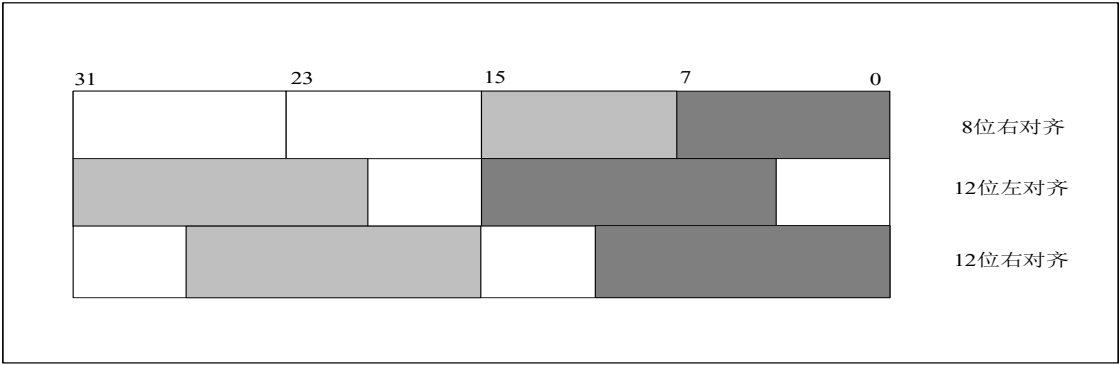


图 18-3 DAC 双通道模式的数据保持寄存器的不同数据格式

18.3.7 带三角波生成的 DAC 转换

通过配置 DAC_CR 寄存器的 WAVEx[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 MAMPx[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 WAVEx[1: 0] 位清零，可以取消三角波的生成。

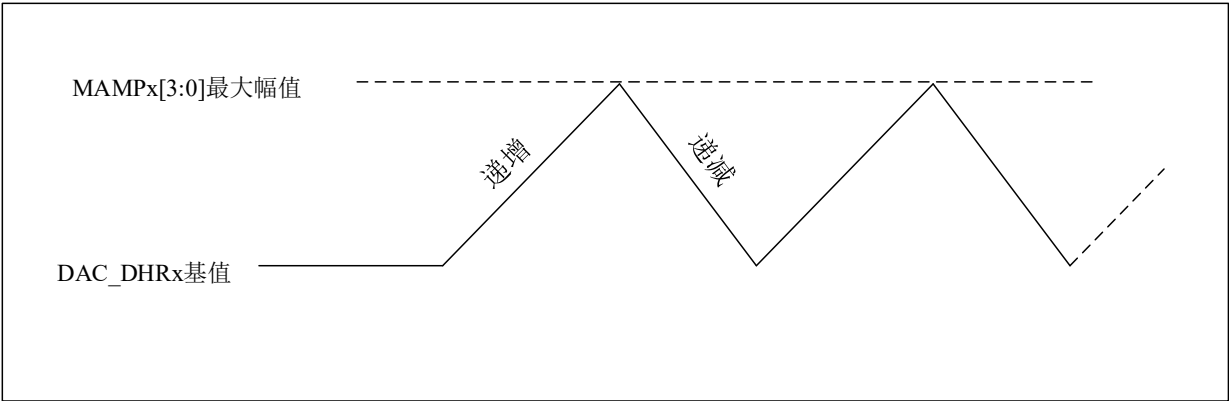


图 18-4 DAC 三角波生成

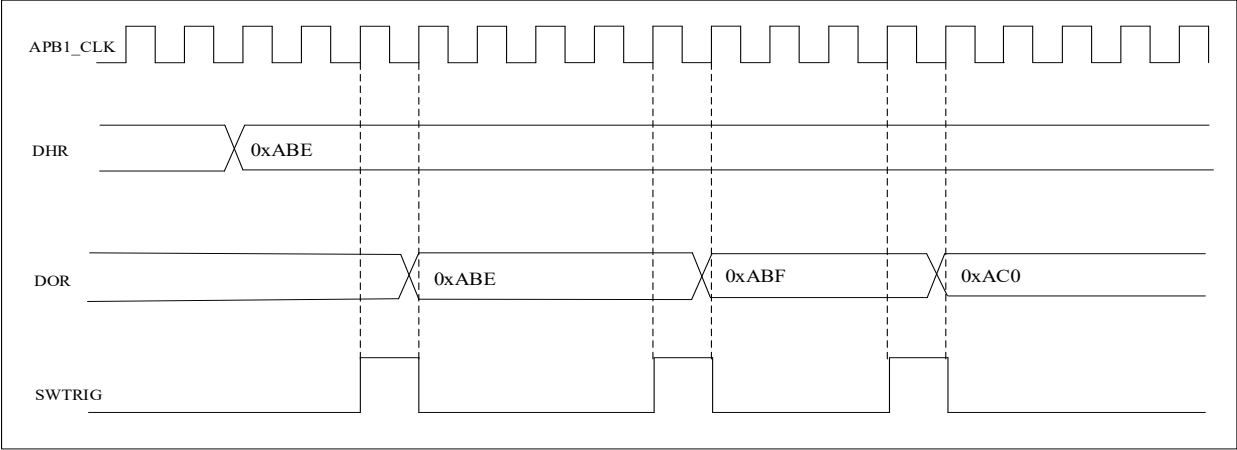


图 18-5 带三角波生成的 DAC 转换（使能软件触发）

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 MAMPx[3: 0] ，否则其值不能修改。

18.3.8 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。

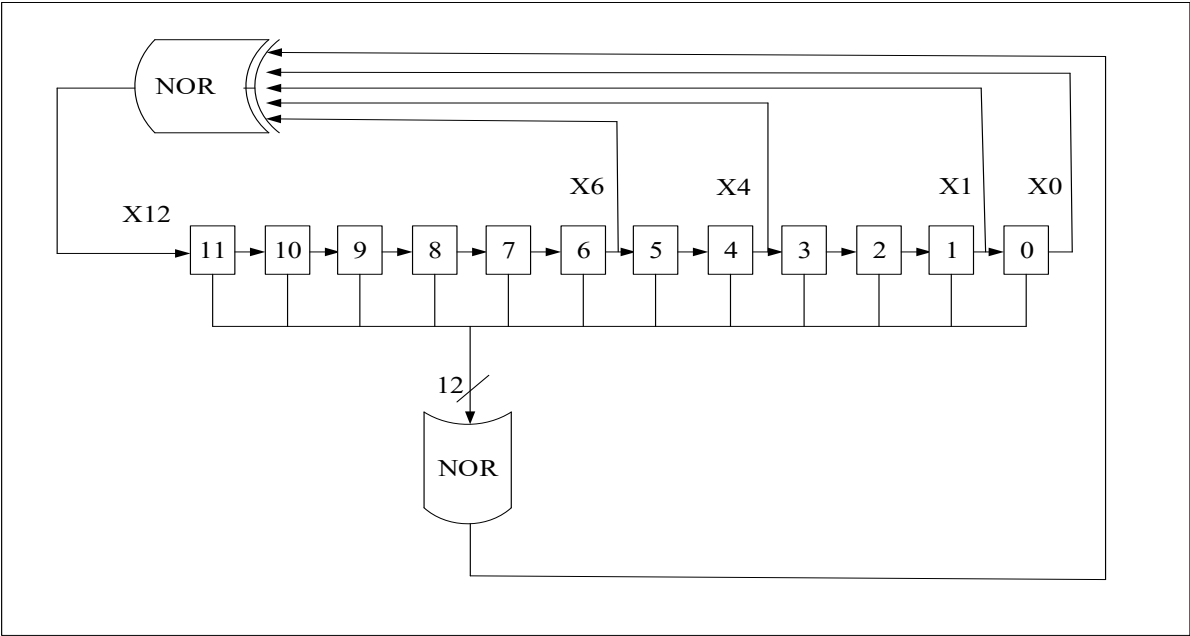


图 18-6 DAC LFSR 寄存器算法

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 LSFR 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。

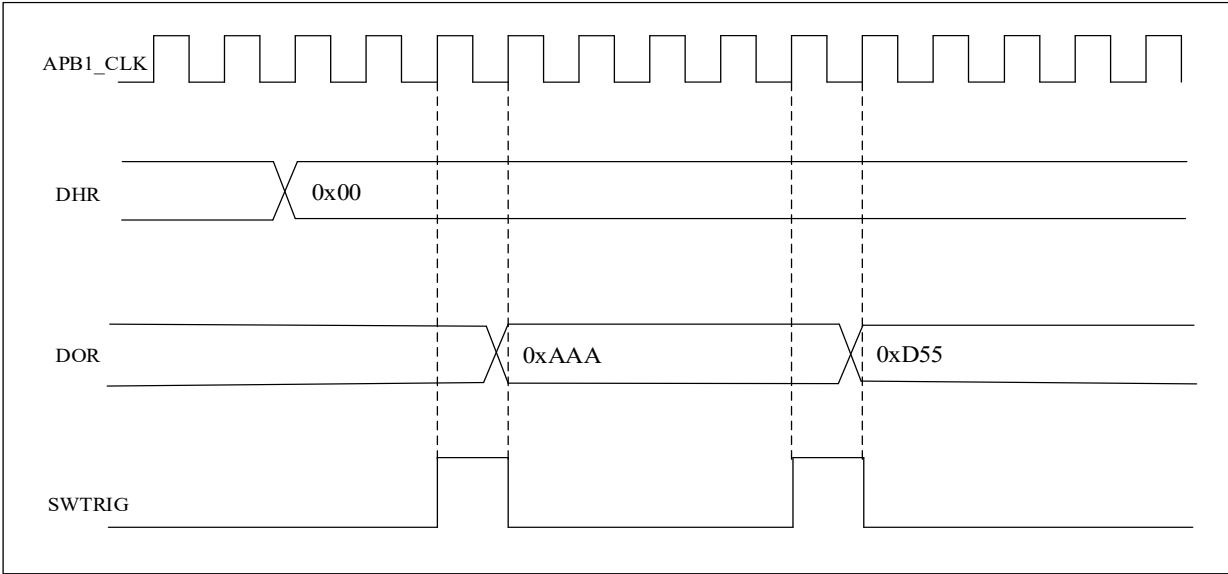


图 18-7 带 LFSR 波形生成的 DAC 转换（使能软件触发）

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。

18.3.9 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 tsettling（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。

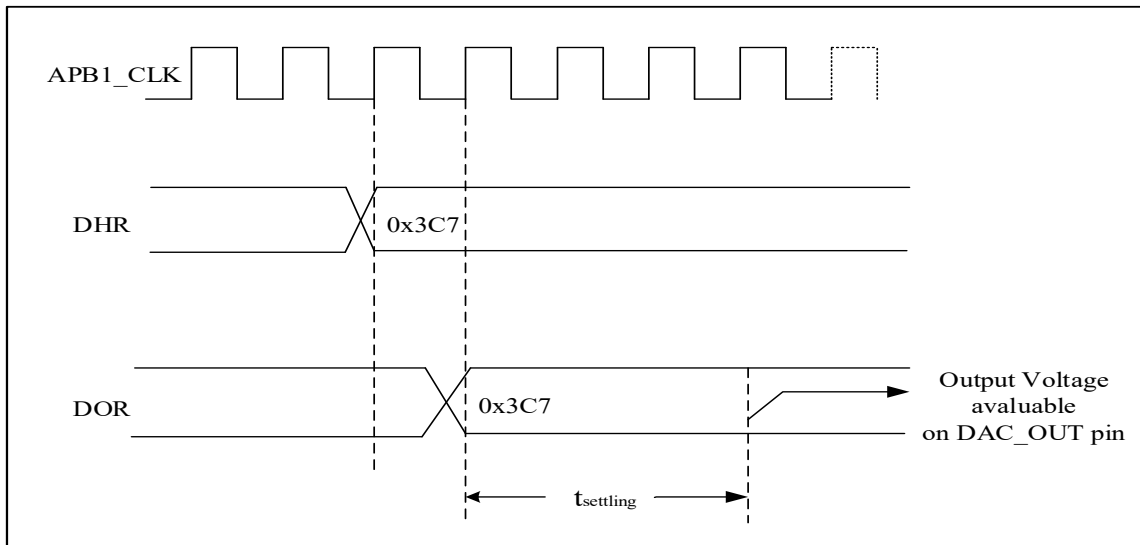


图 18-8 TEN=0 触发关闭时 DAC 转换的时间示意图

18.3.10 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）

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，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，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，WAVEx='01'，MAMPx 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，WAVEx='01'，MAMPx 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）

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，WAVEx='1x'，MAMPx 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，WAVEx='1x'，MAMPx 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，WAVEx='01'，MAMPx 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，WAVEx='01'，MAMPx 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。

18.4 寄存器描述

表 18-2 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	双 DAC 的 12 位右对齐数据保持寄存器	0x00000000
0x24	DAC_DHR12LD	双 DAC 的 12 位左对齐数据保持寄存器	0x00000000
0x28	DAC_DHR8RD	双 DAC 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

18.4.1 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.		OE_EN 2	DMA_E N2	MAMP2				WAVE2		Res.			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.		OE_EN 1	DMA_E N1	MAMP1				WAVE2		Res.			TEN1	BOFF1	EN1
		rw	rw	rw				rw					rw	rw	rw

Bit	Field	Description
31: 30	Reserved	始终读为 0。
29	OE2_EN	DAC 通道 2 输出使能 (DAC channel 2 output enable) 该位由软件设置和清除。 0: 禁止 DAC 通道 2 进行模拟输出转换功能; 1: 使能 DAC 通道 2 进行模拟输出转换功能。

Bit	Field	Description
28	DMA_EN2	DAC 通道 2 DMA 使能 (DAC channel 2 DMA enable) 该位由软件设置和清除。 0: 禁止 DAC 通道 2 DMA 功能; 1: 使能 DAC 通道 2 DMA 功能。
27: 24	MAMP2	DAC 通道 2 屏蔽 / 幅值选择器 (DAC channel 2 mask / amplitude elector) 由软件设置该位, 用来在产生噪声时选择屏蔽位, 产生三角波时选择波形的幅值。 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: 不屏蔽 LSFR 位 [11: 0] / 三角波幅值等于 4095 。
23: 22	WAVE2	DAC 通道 2 噪声 / 三角波生成使能 (DAC channel 2 noise / triangle wave generation enable) 该位由软件设置和清除。 00: 禁止波形生成功能; 01: 生成噪声波形; 1x: 生成三角波形。
21: 19	Reserved	始终读为 0 。
18	TEN2	DAC 通道 2 触发使能 (DAC channel 2 trigger enable) 该位由软件设置和清除, 用来使能 / 禁止 DAC 通道 2 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 ; 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 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 。
17	BOFF2	DAC 通道 2 输出缓存禁止 (DAC channel 2 output buffer disable) 该位由软件设置和清除, 用来使能 / 禁止 DAC 通道 2 的输出缓存。 0: 使能 DAC 通道 2 输出缓存; 1: 禁止 DAC 通道 2 输出缓存。
16	EN2	DAC 通道 2 使能 (DAC channel 2 enable) 该位由软件设置和清除, 用来使能 / 禁止 DAC 通道 2 。 0: 禁止 DAC 通道 2 ; 1: 使能 DAC 通道 2 。

Bit	Field	Description
15: 14	Reserved	始终读为 0。
13	OE_EN1	DAC 通道 1 输出使能 (DAC channel 1 output enable) 该位由软件设置和清除。 0: 禁止 DAC 通道 1 进行模拟输出转换功能; 1: 使能 DAC 通道 1 进行模拟输出转换功能。
12	DMA_EN1	DAC 通道 1 DMA 使能 (DAC channel 1 DMA enable) 该位由软件设置和清除。 0: 禁止 DAC 通道 1 DMA 功能; 1: 使能 DAC 通道 1 DMA 功能。
11: 8	MAMP1	DAC 通道 1 屏蔽 / 幅值选择器 (DAC channel 1 mask / amplitude selector) 由软件设置该位, 用来在产生噪声时选择屏蔽位, 产生三角波时选择波形的幅值。 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: 不屏蔽 LSFR 位 [11: 0] / 三角波幅值等于 4095 ;
7: 6	WAVE1	DAC 通道 1 噪声 / 三角波生成使能 (DAC channel 1 noise / triangle wave generation enable) 该位由软件设置和清除。 00: 禁止波形生成功能; 01: 产生噪声波形; 1x: 产生三角波形。
5: 3	Reserved	始终读为 0。
2	TEN1	DAC 通道 1 触发使能 (DAC channel 1 trigger enable) 该位由软件设置和清除, 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 ; 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 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 。

Bit	Field	Description
1	BOFF1	DAC 通道 1 输出缓存禁止 (DAC channel 1 output buffer disable) 该位由软件设置和清除, 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) 该位由软件设置和清除, 用来使能 / 禁止 DAC 通道 1。 0: 禁止 DAC 通道 1 ; 1: 使能 DAC 通道 1 。

18.4.2 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.														SW_TRI G2 W	SW_TRI G1 W

Bit	Field	Description
31: 2	Reserved	始终读为 0 。
1	SW_TRIG2	DAC 通道 2 软件触发 (DAC channel 2 software trigger) 该位由软件设置和清除, 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) 该位由软件设置和清除, 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)。

18.4.3 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											
				rw											

Bit	Field	Description
31: 12	Reserved	始终读为 0 。
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。

18.4.4 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	始终读为 0 。
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	始终读为 0 。

18.4.5 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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DACC1DHR							
								rw							

Bit	Field	Description
31: 8	Reserved	始终读为 0 。

Bit	Field	Description
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。

18.4.6 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC2DHR											
				rw											

Bit	Field	Description
31: 12	Reserved	始终读为 0 。
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。

18.4.7 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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR												Res.			
rw															

Bit	Field	Description
31: 16	Reserved	始终读为 0 。
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	始终读为 0 。

18.4.8 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	始终读为 0。
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。

18.4.9 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	始终读为 0。
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	始终读为 0。
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。

18.4.10 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 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	始终读为 0 。
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	始终读为 0 。

18.4.11 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	始终读为 0 。
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。

18.4.12 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	始终读为 0。
11: 0	DACC1DOR	DAC 通道 1 输出数据 (DAC channel 1 data output) 表示通道 1 正在进行 DA 转换的数据。

18.4.13 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	始终读为 0。
11: 0	DACC2DOR	DAC 通道 2 输出数据 (DAC channel 2 data output) 表示通道 2 正在进行 DA 转换的数据。

19 COMP 比较器

19.1 简介

芯片内嵌 3 个通用比较器（COMPx, x=1, 2, 3），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。比较器也可以和定时器结合使用，与定时器输出的 PWM 相结合，可以组成周期性的电流控制回路。

19.2 功能框图

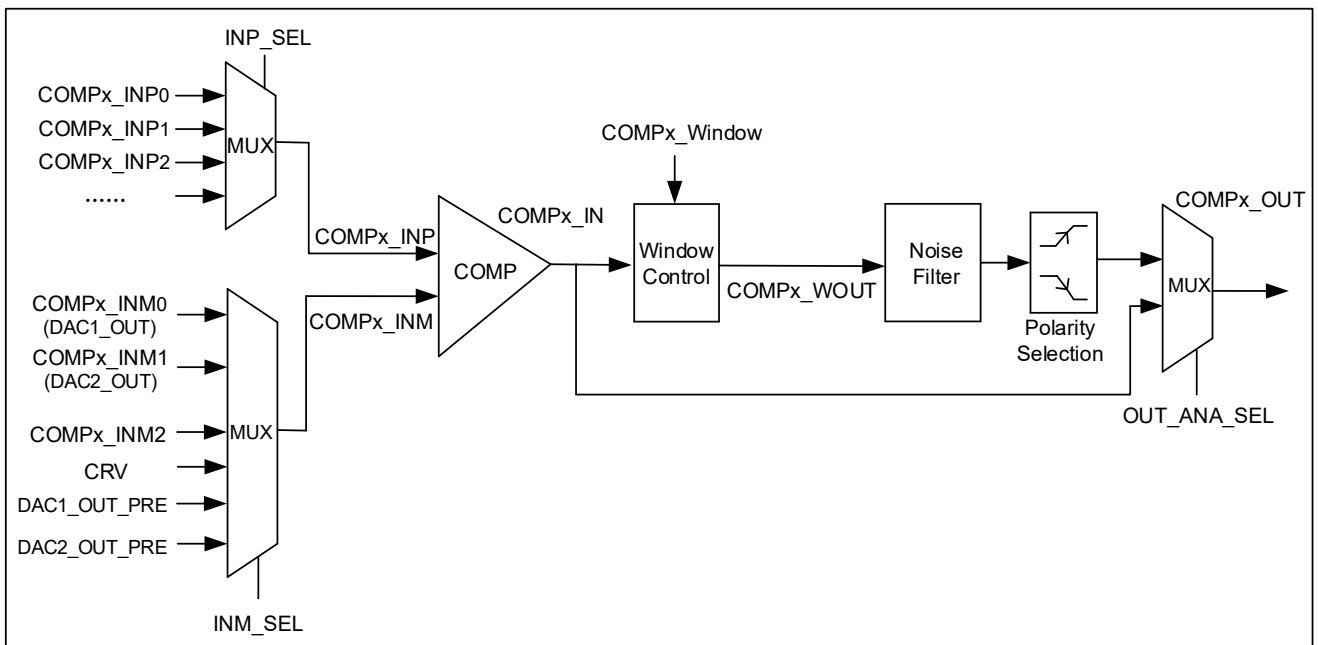


图 19-1 比较器框图

19.3 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；
- 比较器（COMPx, x=1, 2, 3）支持 4 个正相输入和 6 个反相输入，带有轮询功能；

- 比较器（COMP_x, x=1, 2, 3, 4）支持 4 个正相输入和 6 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。

19.4 功能描述

19.4.1 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。

19.4.2 比较器开关控制

在使用比较器之前，需要置位 COMP_x_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。

19.4.3 比较器输入和输出

当 I/O 用作比较器输入时，必须在 GPIO 寄存器中将其设置为模拟输入模式。模拟比较器比较的结果经过 Filter 滤波模块，再经过极性配置后输出，送入其他模块使用；当 OUT_ANA_SEL 为 1 时，模拟比较器的比较结果不经过任何模块，直接输出到 COMP_x_OUT。比较器的输出可选择滤波功能（参考 COMP_x_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 COMP_x_CSR 寄存器的 OUT_SEL 的配置），也可以输出到 I/O。

COMP_x_CSR 寄存器的 WE 位用于开启或关闭窗口控制功能。开启该功能后，当 COMP_x_Window 信号为高时，Window 控制模块周期性（按总线时钟 pclk）地采样模拟比较信号 COMP_x_IN，产生 COMP_x_WOUT 信号，当 COMP_x_Window 信号为低时，COMP_x_WOUT 保持当前值（COMP_x_Window 为高时的最后一次采样）不变；关闭该功能后，模拟比较信号 COMP_x_IN 直接输出到数字滤波模块（Noise Filter）。COMP_x_Window 信号来自 ip2ip 模块的输出，配置方法参考“芯片特定配置”章节。窗口控制功能可以用来过滤比较器输入电压无效时的输出，也可以在某些 PWM 应用中实现过零检测，在电机应用中用于滤除瞬态电流脉冲信号。实际应用中，COMP_x_WOUT 可能会延迟模拟输入最多 1 个总线时钟 pclk，对于一些特点的应用场景，必须仔细考虑窗口的位置，窗口宽度要至少大于 1 个 pclk 时钟周期，以保证信号能被采样。

19.4.4 比较器通道选择

COMP_x (x=1, 2, 3) 有 4 个正相输入和 6 个反相输入通道，正相输入可从 4 个外部引脚之间选取，反相输入可从 3 个外部引脚、CRV 电压分压值。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 COMP_x_CSR (x=1, 2, 3) 寄存器的 INP_SEL 位和 INM_SEL 位，选择所要比较的信号；

配置 COMP_x_CSR 寄存器的 EN 位，比较器开始上电工作；

比较的结果存放于 COMP_x_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 COMP_x_POLL (x=1, 2, 3) 寄存器的 FIXN 位选择跟随 INP 端口变化或者由 COMP_x_CSR (x=1, 2, 3) 的 INM_SEL 位来配置。需要注意的是，当启动轮询功能以后，COMP_x_CSR 的 INP_SEL 位将失去作用，同样的，如果 COMP_x_POLL (x=1, 2, 3) 寄存器的 FIXN 位选择 INM 端口跟随 INP 轮询变化，COMP_x_CSR (x=1, 2, 3) 的 INM_SEL 位也将失去作用。具体流程如下：

配置 COMP_x_POLL (x=1, 2, 3) 寄存器的 PERIOD 位来选择所需要的轮询等待周期；

配置 COMP_x_POLL (x=1, 2, 3) 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 COMP_x_POLL (x=1, 2, 3) 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 COMP_x_POLL (x=1, 2, 3) 寄存器的 POLL_EN 位，启动轮询功能；

配置 COMP_x_CSR (x=1, 2, 3) 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 COMP_x_POLL (x=1, 2, 3) 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。

19.4.5 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。

19.4.6 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。

COMP_x_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。

19.4.7 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）

19.4.8 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）。

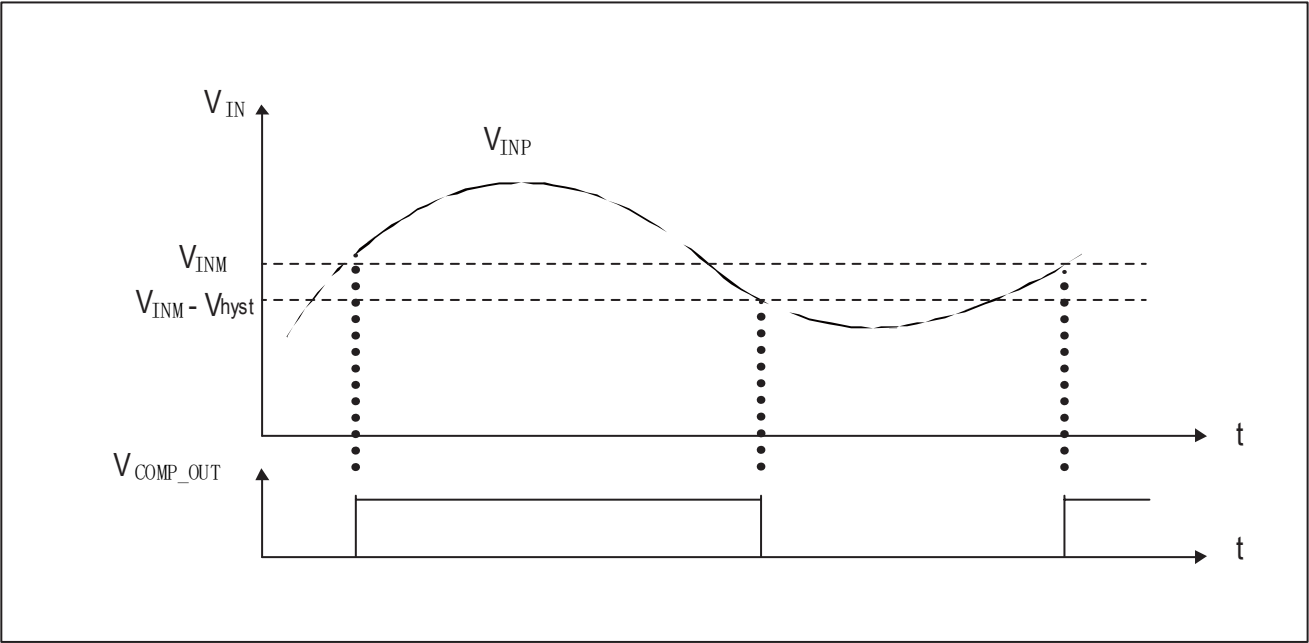


图 19-2 比较器的迟滞

19.5 寄存器

表 19-1 COMP 寄存器概览

Offset	Acronym	Register Name	Reset
0xC, 0x10, 0x30	COMPx_CSR (x=1, 2, 3)	比较器 x (x=1, 2, 3) 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, 0x24	COMPx_POLL (x=1, 2, 3)	比较器 x (x=1, 2, 3) 轮询寄存器	0x00000000

19.5.1 比较器控制状态寄存器 (COMPx_CSR) (x=1, 2, 3)

偏移地址: 0xC, 0x10, 0x30

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT	OUT_A NA_SEL	WE	Res.							OFLT			HYST	
rw	r	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		INM_SEL			MODE		Res.	EN
rw		rw					rw		rw			rw			rw

Bit	Field	Description
31	LOCK	比较器寄存器锁 (Comparator register lock) 该位只能写一次, 由软件置'1', 由系统复位清零。一旦设置, 比较器 x 的所有控制位为只读。 1: COMPx_CSR 只读 0: COMPx_CSR 可读可写
30	OUT	比较器 x 输出 (Comparator x output) 反映比较器 x 输出状态。 1: 高输出 (正相输入高于反相输入) 0: 低输出 (正相输入低于反相输入)
29	OUT_ANA_SEL	比较器 x 输出来源选择 (Comparator x output source selection) 1: 选择模拟输出信号 0: 选择经过同步的模拟输出信号
28	WE	比较器 x 窗口控制输出使能位 (Comparator x window control enable) 1: 使能窗口控制输出 0: 禁止窗口控制输出
27 : 21	Reserved	始终读为 0。

Bit	Field	Description
20 : 18	OFLT	比较器 x 输出滤波周期 (Comparator Output Filter Period) 这些位控制比较器 x 的输出滤波周期, 当比较器输出信号小于滤波周期宽度时, 被认为是无效信号而被过滤掉, 否则认为是有效信号, 其中 n 代表 COMPx_POLL ->PERIOD 配置的轮询等待周期。 111: 128 * n 个时钟周期 110: 64 * n 个时钟周期 101: 32 * n 个时钟周期 100: 16 * n 个时钟周期 011: 8 * n 个时钟周期 010: 4 * n 个时钟周期 001: 2 * n 个时钟周期 000: 无滤波
17 : 16	HYST	比较器 x 迟滞电压 (Comparator x hysteresis) 这些位控制比较器 x 的迟滞电压。 11: 90mV 10: 30mV 01: 15mV 00: 0mV
15	POL	比较器 x 输出极性控制 (Comparator x output polarity) 该位用于选择比较器 x 输出极性。 1: 输出端反相输出 0: 输出端同相输出
14	Reserved	始终读为 0。
13 : 10	OUT_SEL	比较器 x 输出选择 (Comparator x output selection) 这些位用来选择比较器 x 的输出目标。 0010: 定时器 1 输入刹车输入 0100: 定时器 8 输入刹车输入 0110: 定时器 1 Ocrefclear 输入 0111: 定时器 1 输入捕获 1 1000: 定时器 2 输入捕捉 4 1001: 定时器 2 Ocrefclear 输入 1010: 定时器 3 输入捕获 1 1011: 定时器 3 Ocrefclear 输入 1111: LPTIMER 的 trig 输入 其他: 无选择
9	Reserved	始终读为 0。

Bit	Field	Description
8: 7	INP_SEL	比较器 x 正相输入选择 (Comparator x normal phase input selection) 这些位用于选择连接到比较器 x 正相输入端的信号源。 比较器 1: 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; 比较器 2: 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; 比较器 3: 00: COMP3_INP0 (PC1), 即正向输入端通道 0; 01: COMP3_INP1 (PB0), 即正向输入端通道 1; 10: COMP3_INP2 (PE7), 即正向输入端通道 2; 11: COMP3_INP3 (PA0), 即正向输入端通道 3;
6: 4	INM_SEL	比较器 x 反相输入选择 (Comparator x inverting input selection) 这些位用于选择连接到比较器 x 反相输入端的信号源。 比较器 1: 000: COMP1_INM0 (PA4), 即反向输入端通道 0; 001: COMP1_INM1 (PA5), 即反向输入端通道 1; 010: COMP1_INM2 (PA0), 即反向输入端通道 2; 011: COMP1_INM3 (CRV), 即反向输入端通道 3; 100: COMP1_INM4 (DAC1_OUT_PRE), 即反向输入端通道 4; 101: COMP1_INM5 (DAC2_OUT_PRE), 即反向输入端通道 5; 比较器 2: 000: COMP2_INM0 (PA4), 即反向输入端通道 0; 001: COMP2_INM1 (PA5), 即反向输入端通道 1; 010: COMP2_INM2 (PA2), 即反向输入端通道 2; 011: COMP2_INM3 (CRV), 即反向输入端通道 3; 100: COMP2_INM4 (DAC1_OUT_PRE), 即反向输入端通道 4; 101: COMP2_INM5 (DAC2_OUT_PRE), 即反向输入端通道 5; 比较器 3: 000: COMP3_INM0 (PC0), 即反向输入端通道 0; 001: COMP3_INM1 (PB1), 即反向输入端通道 1; 010: COMP3_INM2 (PE8), 即反向输入端通道 2; 011: COMP3_INM3 (CRV), 即反向输入端通道 3; 100: COMP3_INM4 (DAC1_OUT_PRE), 即反向输入端通道 4; 101: COMP3_INM5 (DAC2_OUT_PRE), 即反向输入端通道 5;

Bit	Field	Description
3:2	MODE	比较器 x 模式 (Comparator x mode) 比较器 x 的工作模式控制位, 允许调整速率和功耗。 11: 极低功率/极低速 10: 低功率/低速 01: 中等功率/中速 00: 高功率/高速
1	Reserved	始终读为 0。
0	EN	比较器 x 使能 (Comparator x enable) 该位是比较器开关使能位。 1: 比较器 x 使能 0: 比较器 x 禁止

19.5.2 比较器外部参考电压寄存器 (COMP_CRV)

偏移地址: 0x18

复位值: 0x00000000

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

Bit	Field	Description
31:6	Reserved	保留, 始终读为 0
5	CRV_SRC	比较器参考电压源选择 (Comparator Reference Voltage Source Select) 0: VREFINT (内部电压) 1: VDDA (外部电压) 备注: 当选用内部电压作比较器的参考电压源时, 需要提前配置 ADC_ADCFG 来打开内部电压传感器使能
4	CRV_EN	比较器参考电压使能 (Comparator Reference Voltage Enable) 1: 比较器参考电压 (CRV) 使能 0: 比较器参考电压 (CRV) 禁止

Bit	Field	Description
3:0	CRV_SEL	比较器参考电压选择 (Comparator Reference Voltage Select) 0000: 1/20 VDDA 或 1/20 V _{REFINT} 0001: 2/20 VDDA 或 2/20 V _{REFINT} 0010: 3/20 VDDA 或 3/20 V _{REFINT} 0011: 4/20 VDDA 或 4/20 V _{REFINT} 0100: 5/20 VDDA 或 5/20 V _{REFINT} 0101: 6/20 VDDA 或 6/20 V _{REFINT} 0110: 7/20 VDDA 或 7/20 V _{REFINT} 0111: 8/20 VDDA 或 8/20 V _{REFINT} 1000: 9/20 VDDA 或 9/20 V _{REFINT} 1001: 10/20 VDDA 或 10/20 V _{REFINT} 1010: 11/20 VDDA 或 11/20 V _{REFINT} 1011: 12/20 VDDA 或 12/20 V _{REFINT} 1100: 13/20 VDDA 或 13/20 V _{REFINT} 1101: 14/20 VDDA 或 14/20 V _{REFINT} 1110: 15/20 VDDA 或 15/20 V _{REFINT} 1111: 16/20 VDDA 或 16/20 V _{REFINT}

19.5.3 比较器轮询寄存器 (COMPx_POLL) (x=1, 2, 3)

偏移地址: 0x1C, 0x20, 0x24

复位值: 0x00000000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					POUT			Res.	PERIOD			Res.	FIXN	POLL_C	POLL_E
					r				rw				rw	rw	rw

Bit	Field	Description
31:11	Reserved	保留, 始终读为 0
10:8	POUT	轮询通道输出 (Polling output) 只读, 反映轮询通道输出状态, POUT[0]对应通道 1, POUT[1]对应通道 2, POUT[2]对应通道 3。 1: 高输出 (正相输入高于反相输入) 0: 低输出 (正相输入低于反相输入)
7	Reserved	始终读为 0。

Bit	Field	Description
6: 4	PERIOD	轮询等待周期（Polling wait cycle） 每 n 个 PCLK2 周期切换到下一个轮询通道。 111: 128 个时钟周期 110: 64 个时钟周期 101: 32 个时钟周期 100: 16 个时钟周期 011: 8 个时钟周期 010: 4 个时钟周期 001: 2 个时钟周期 000: 1 个时钟周期
3	Reserved	始终读为 0。
2	FIXN	轮询通道反相输入端固定（Polling inverting input fix） 1: 轮询通道反相输入固定，由 CSR 寄存器 INM_SEL 决定，此时 INP_SEL 无效。 0: 轮询通道反相输入不固定，与 INP 通道同时变化，此时 INP_SEL 和 INM_SEL 无效。
1	POLL_CH	比较器轮询通道（Comparator polling channel） 1: 轮询通道 1/2/3 0: 轮询通道 1/2
0	POLL_EN	比较器轮询模式使能（Comparator polling enable） 1: 比较器轮询模式使能 0: 比较器轮询模式禁止

20 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。

20.1 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）。

20.2 功能框图

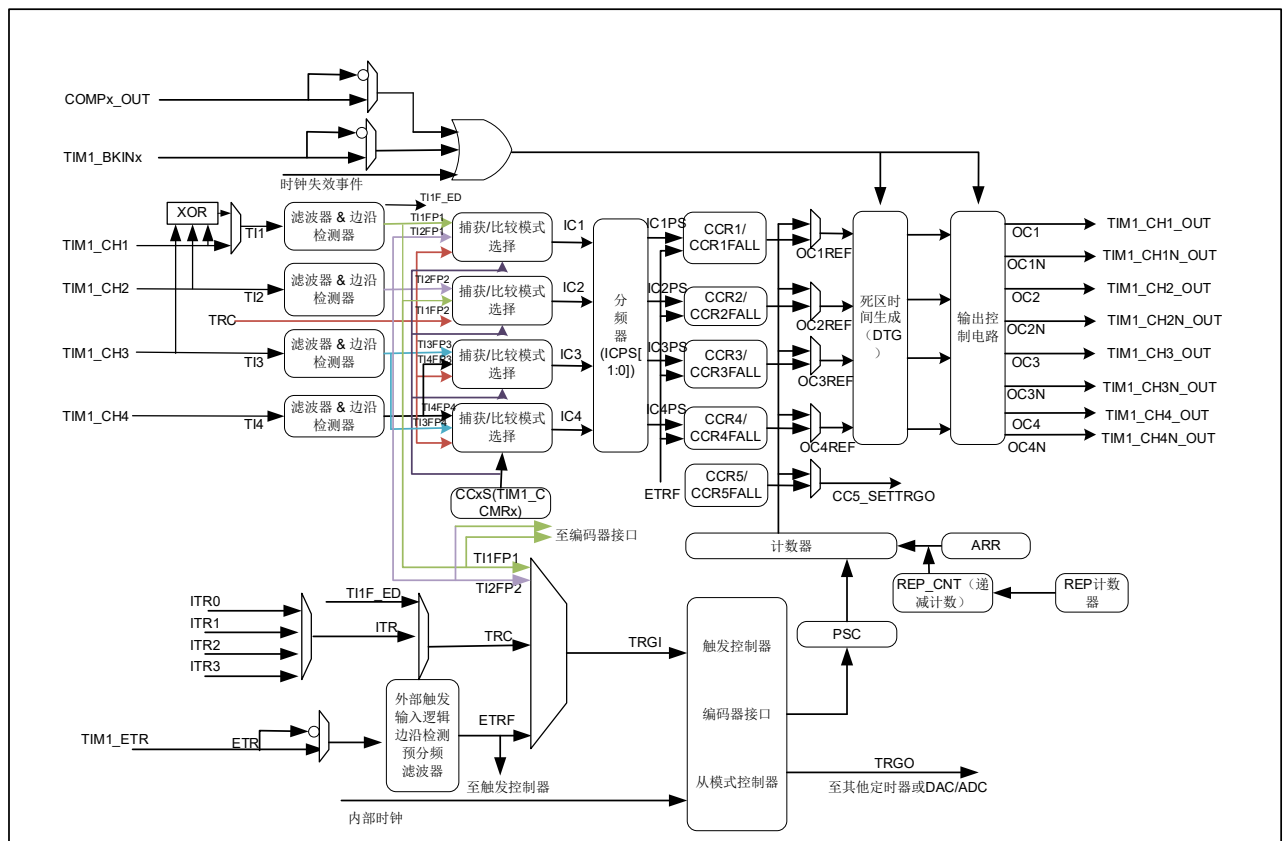


图 20-1 TIMx 结构图

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。

20.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调
- 时钟源可选：内部时钟源，外部时钟输入（Tl_x、ETR_x），内部触发输入（ITR_x）
- 16 位自动重装载计数器（计数方向：递增、递减、递增/递减）
- 8 位可编程重复计数功能，重复计数器可自动重装载（定时器到指定时间后自动更新重复寄存器）
- 外部信号控制定时器并且能够实现定时器间互连的同步电路
- 输入捕获：输入信号的脉冲宽度、周期的测量
- 触发输入可以作为外部时钟或者逐周期管理
- 支持编码器、霍尔传感器等接口
- 4 个包含互补输出功能的输出通道
- 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

20.4 功能描述

20.4.1 时钟

20.4.1.1 时钟选择

计数器的时钟源有以下几种：

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

上述几种时钟选择示意图如下：

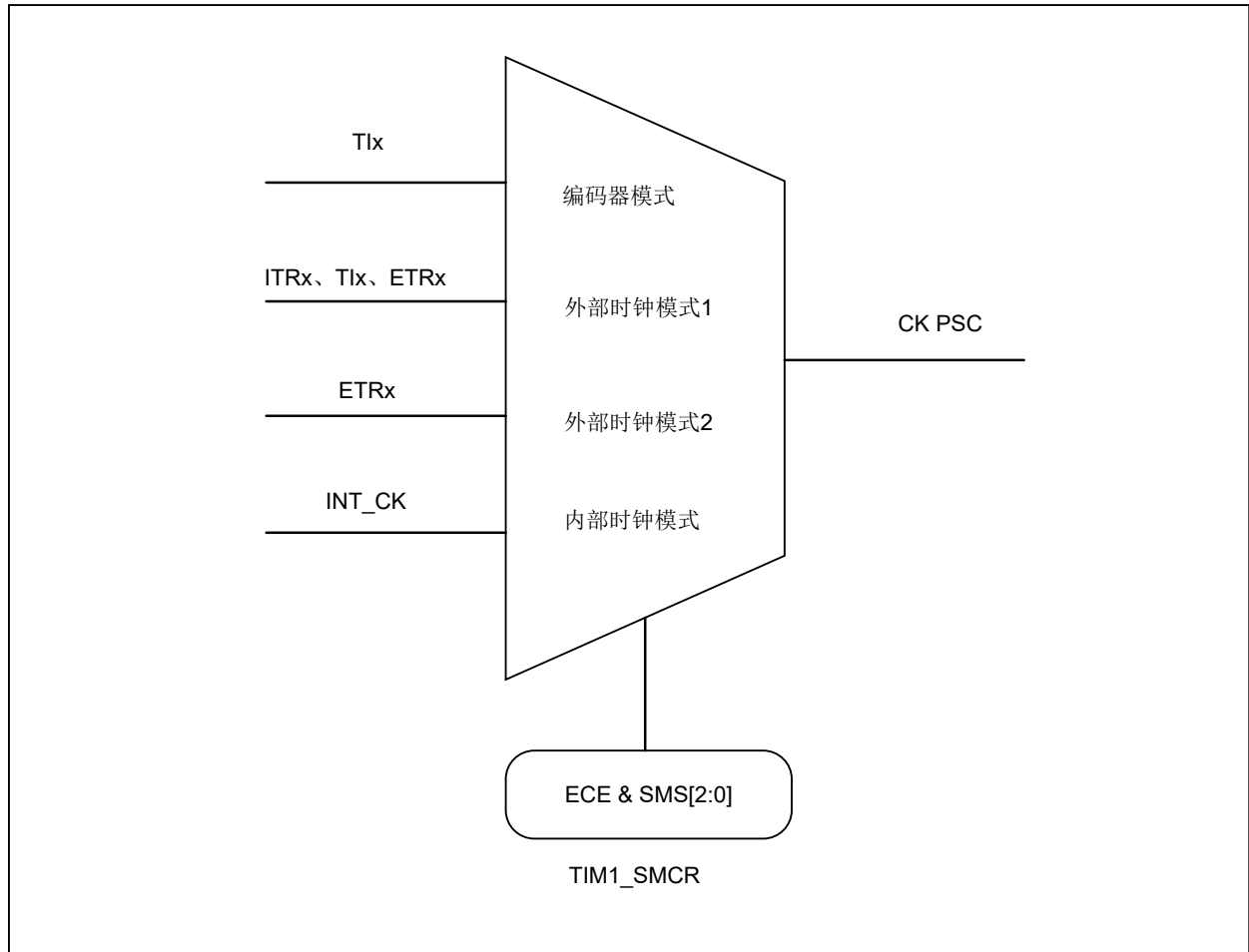


图 20-2 时钟选择

20.4.1.1.1 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。

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

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

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

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

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

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

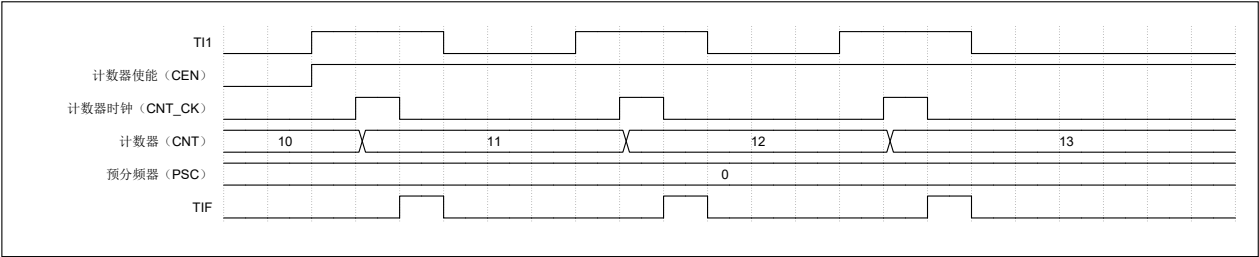


图 20-3 外部时钟模式 1 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）以及重复计数寄存器（TIMx_RCR）为复位值。

20.4.1.1.3 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。

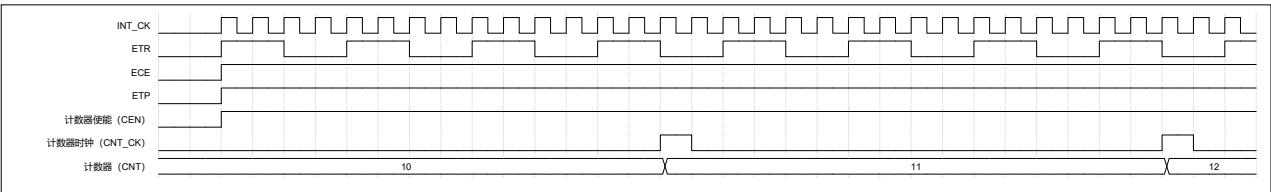


图 20-4 外部时钟模式 2 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）以及重复计数寄存器（TIMx_RCR）为复位值。

20.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

20.4.1.2 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，在下次更新事件时生效。

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

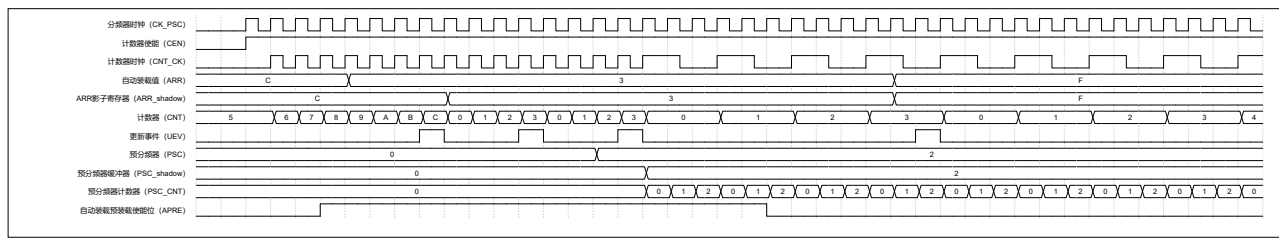


图 20-5 自动预装载

20.4.1.3 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。

20.4.1.3.1 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。

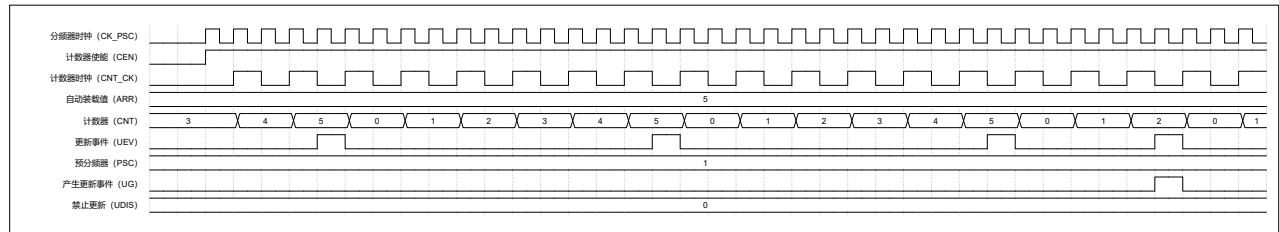


图 20-6 递增计数模式（UDIS=0）

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。

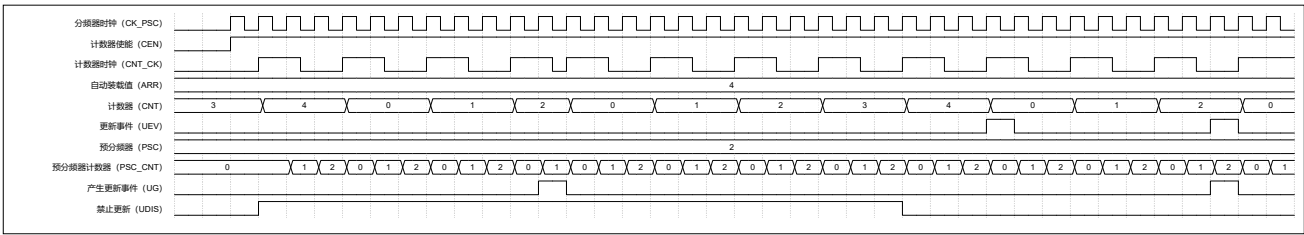


图 20-7 递增计数模式（UDIS=1 禁止产生更新事件）

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。

20.4.1.3.2 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**）。

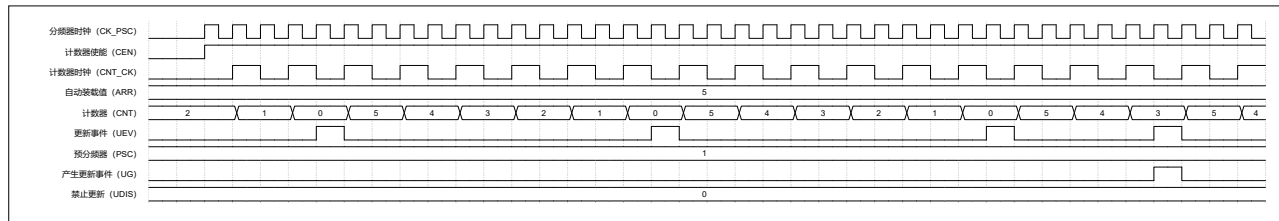


图 20-8 递减计数模式（UDIS=0）

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。

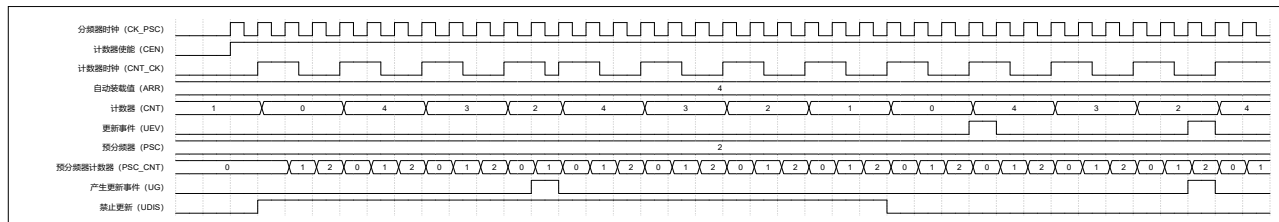


图 20-9 递减计数模式（UDIS=1 禁止产生更新事件）

20.4.1.3.3 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）。

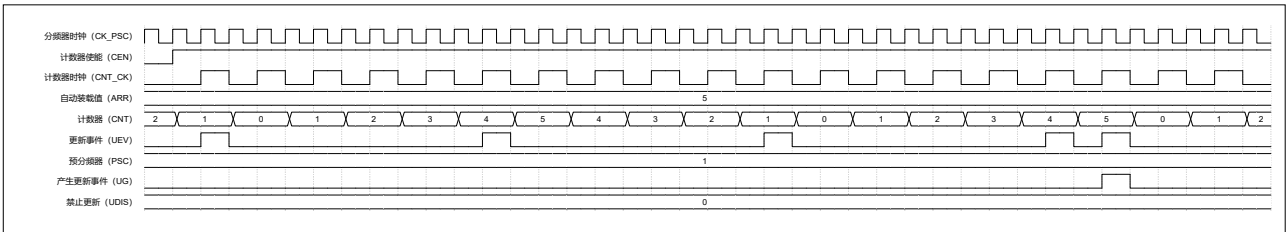


图 20-10 中央计数模式（UDIS=0）

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。

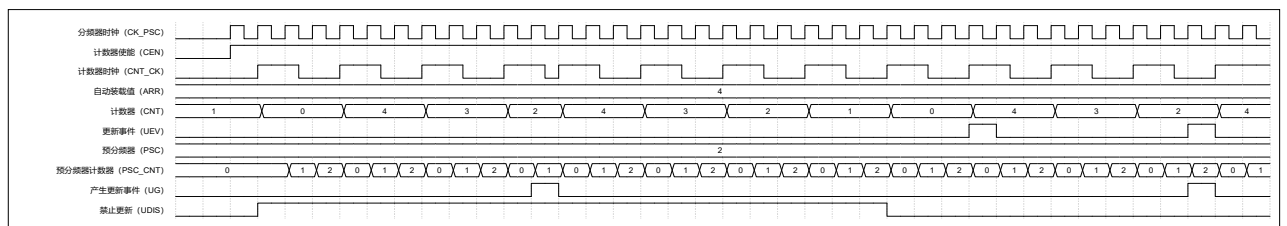


图 20-11 中央计数模式（UDIS=1 禁止产生更新事件）

20.4.2 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。

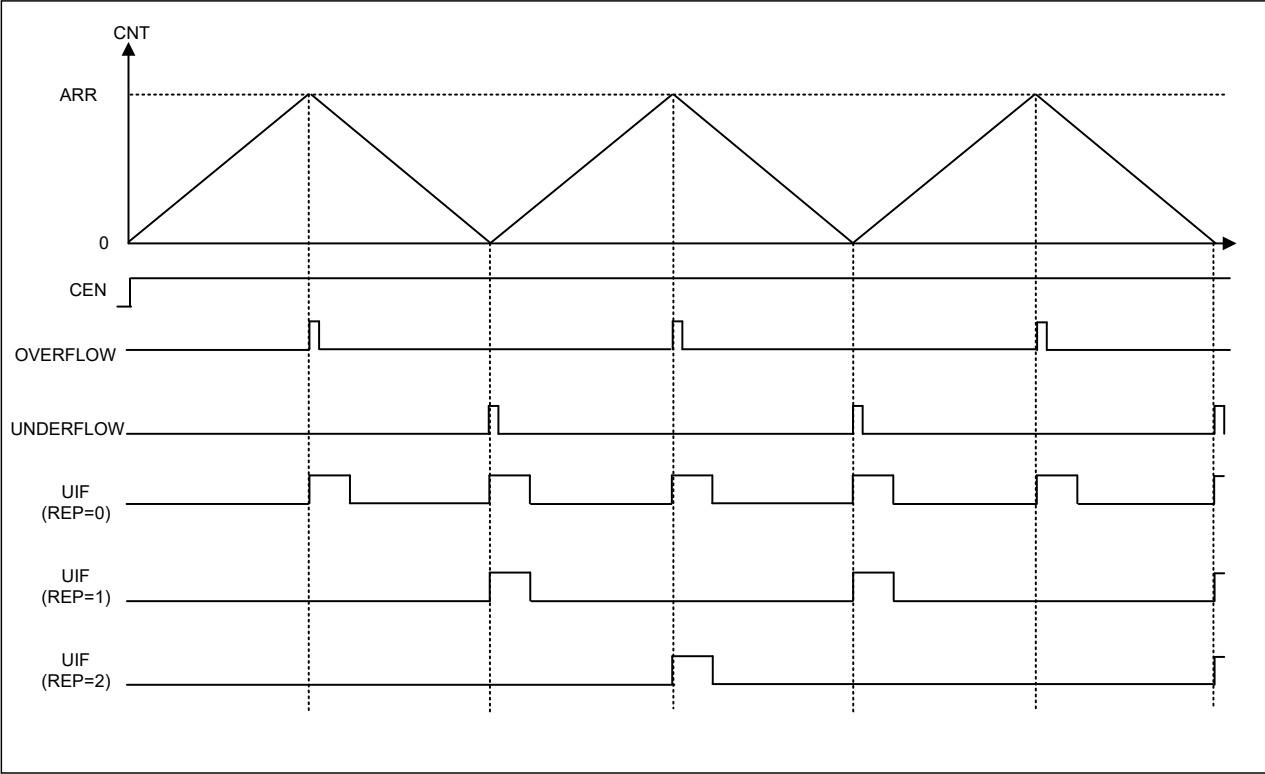


图 20-12 中央对齐模式重复计数时序图

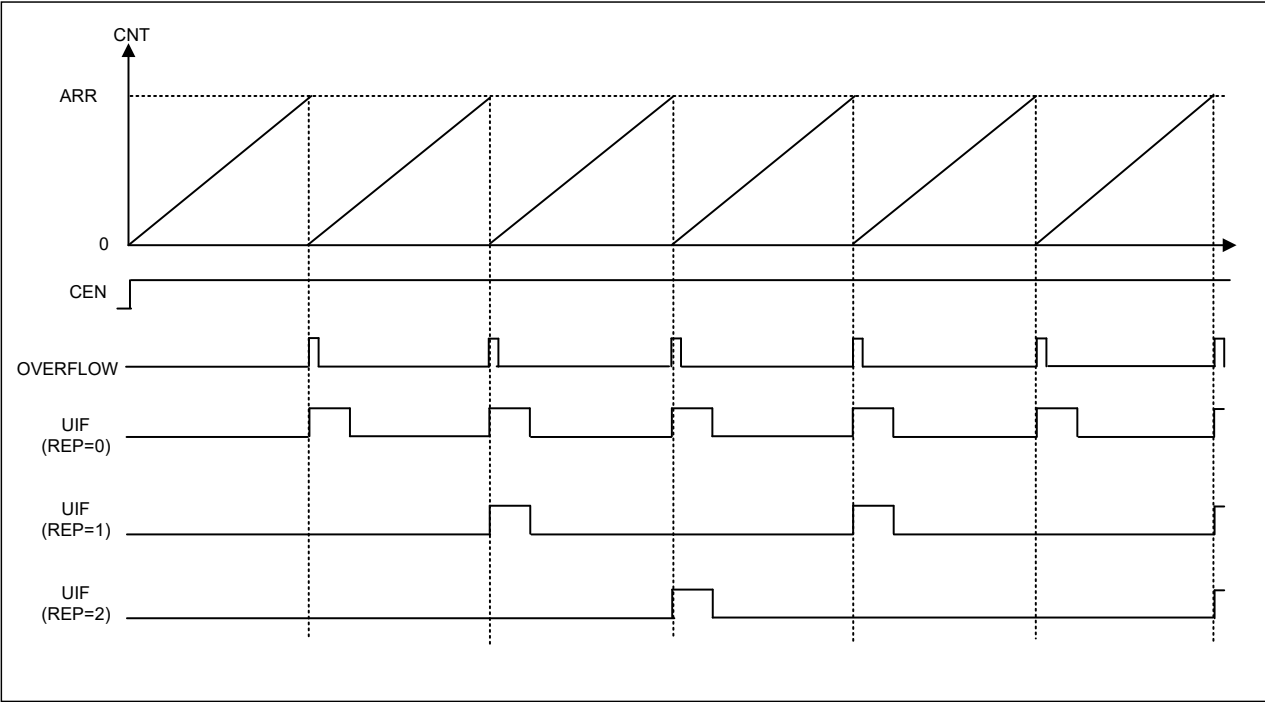


图 20-13 边沿对齐模式递增计数时序图

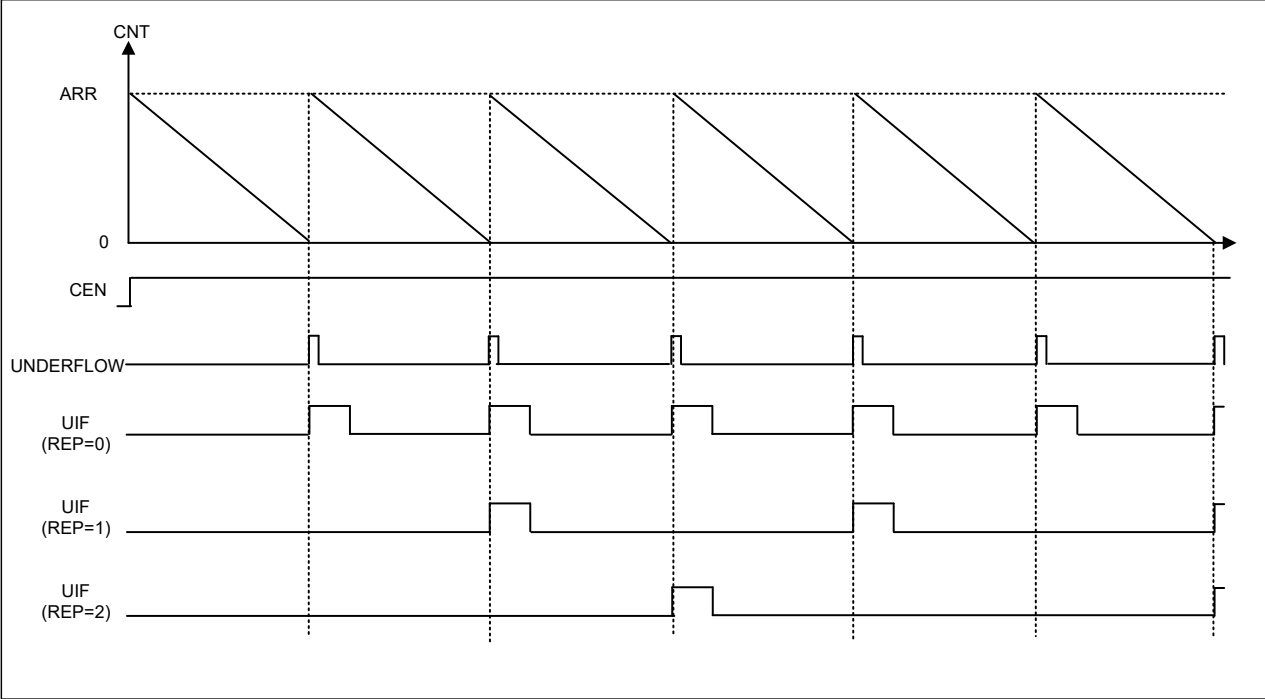


图 20-14 边沿对齐模式递减计数时序图

20.4.3 输入捕获

20.4.3.1 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：

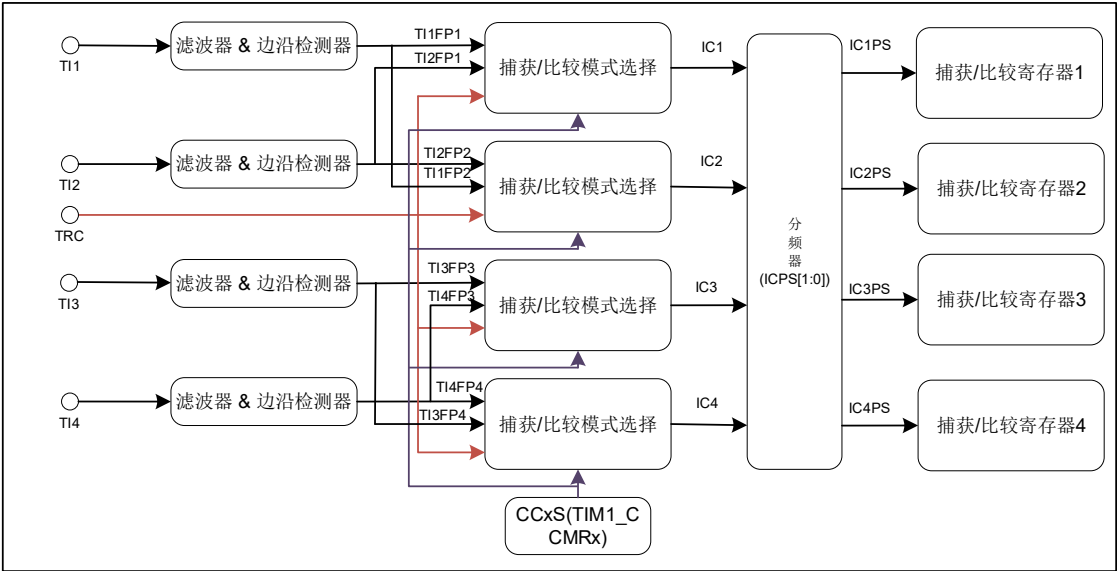


图 20-15 TIMx 输入捕获结构图

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 Tl_x 的输入信号采样后，产生一个滤波后的信号 Tl_xF ，然后通过极性可选的边沿检测器，产生一个有效信号 Tl_xFP_x ，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 IC_xPS ，用于触发输入捕获事件。

表 20-1 数字滤波器宽度与 IC_xF 的对应关系表

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

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

例如，通过采样 $TI1$ 输入信号的有效沿，在 $TI1$ 的上升沿来到时捕获当前计数器的值，锁存到 $TIMx\_CCR1$ 寄存器中，步骤如下：

1. 配置 $TIMx\_CCMR1$ 寄存器的 $CC1S=01$ ， $CC1$ 通道被配置为输入， $IC1$ 映射在 $TI1$ 上。
2. 配置 $TIMx\_CCMR1$ 寄存器的 $IC1F[3: 0]$ ，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 $TIMx\_CCER$ 寄存器的 $CC1P=0$ ，选择捕获发生在 $TI1$ 信号的上升沿。
4. 配置 $TIMx\_CCMR1$ 寄存器的 $IC1PSC[1:0]$ ，选择预分频系数。
5. 配置 $TIMx\_CCER$ 寄存器的 $CC1E=1$ ，开启输入/捕获通道 1 的捕获使能。
6. 配置 $TIMx\_DIER$ 寄存器的 $CC1IE=1$ ，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 DMA，配置 $TIMx\_DIER$ 寄存器的 $CC1DE=1$ ，允许捕获/比较通道 1 的 DMA 请求。

注：

- 当通道配置为输入模式时， $TIMx\_CCR_x$ 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。

20.4.3.2 PWM 捕获

PWM 输入模式的操作配置与一般输入捕获有以下不同点：

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

例：测量 TI1 的 PWM 信号的宽度（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。

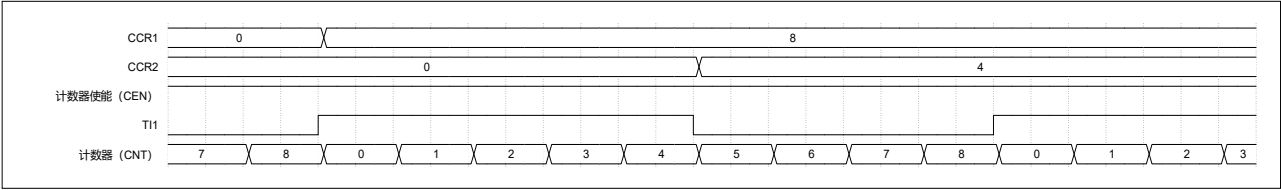


图 20-16 PWM 输入模式时序

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。

20.4.4 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：



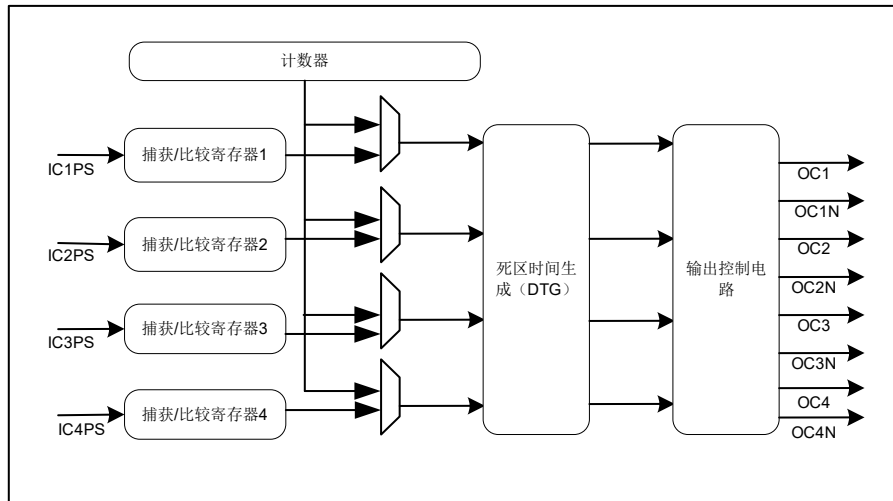


图 20-17 比较输出部分结构图

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。

20.4.4.1 强制输出

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

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

20.4.4.2 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：

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

2. 匹配时状态寄存器中的标志位置 1 (TIMx_SR 寄存器中的 CCxIF 位)。
3. 当配置了 TIMx_DIER 寄存器中的 CCxIE =1, 匹配时则产生一个中断。
4. 当配置了 TIMx_DIER 寄存器中的 CCxDE =1, 匹配时则产生一个 DMA 请求 (仅适用于有内置 DMA 的产品)。

比较输出模式也可以用来输出一个单脉冲 (单脉冲输出模式)。

例如, 通道 1 的比较输出模式的配置步骤如下:

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

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

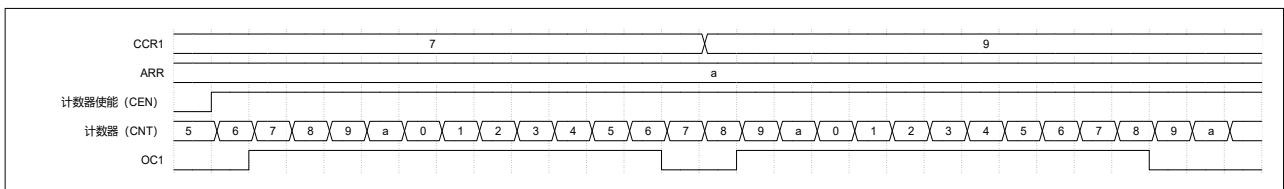


图 20-18 比较输出模式, OC1 信号在匹配时翻转

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。

20.4.4.3 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 CMS 位，可以选择产生边沿对齐或中央对齐的 PWM 信号。

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

20.4.4.3.1 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。

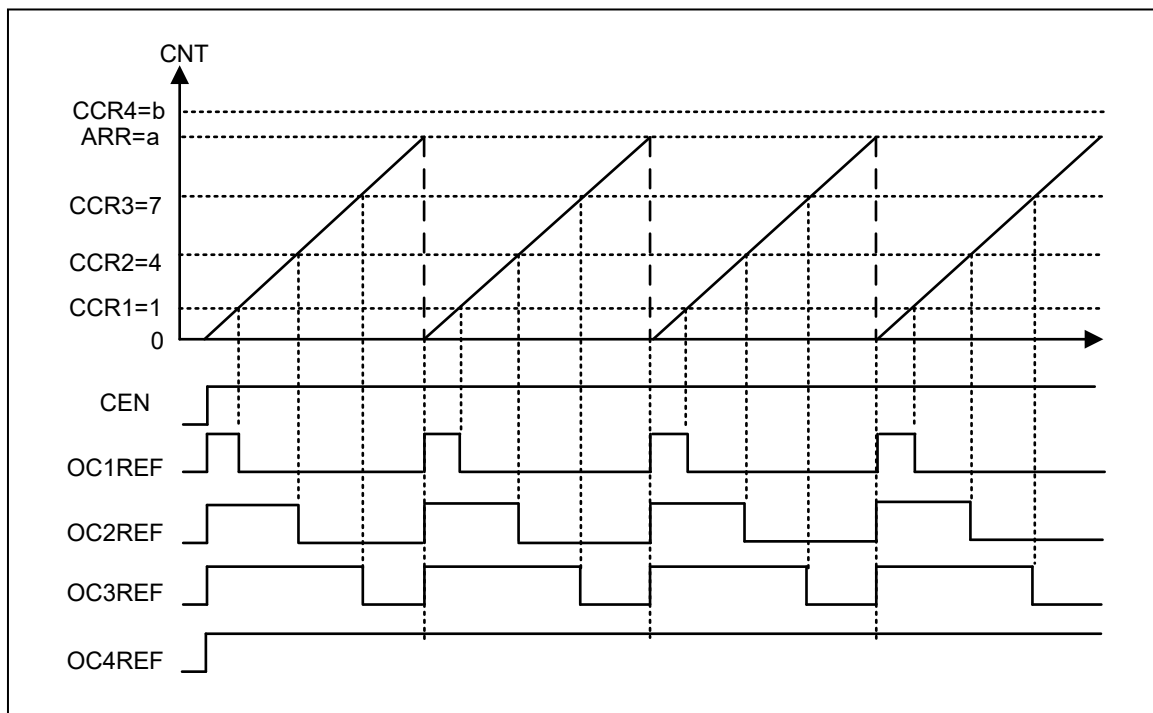


图 20-19 边沿对齐递增计数时 PWM 模式 1 的波形

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

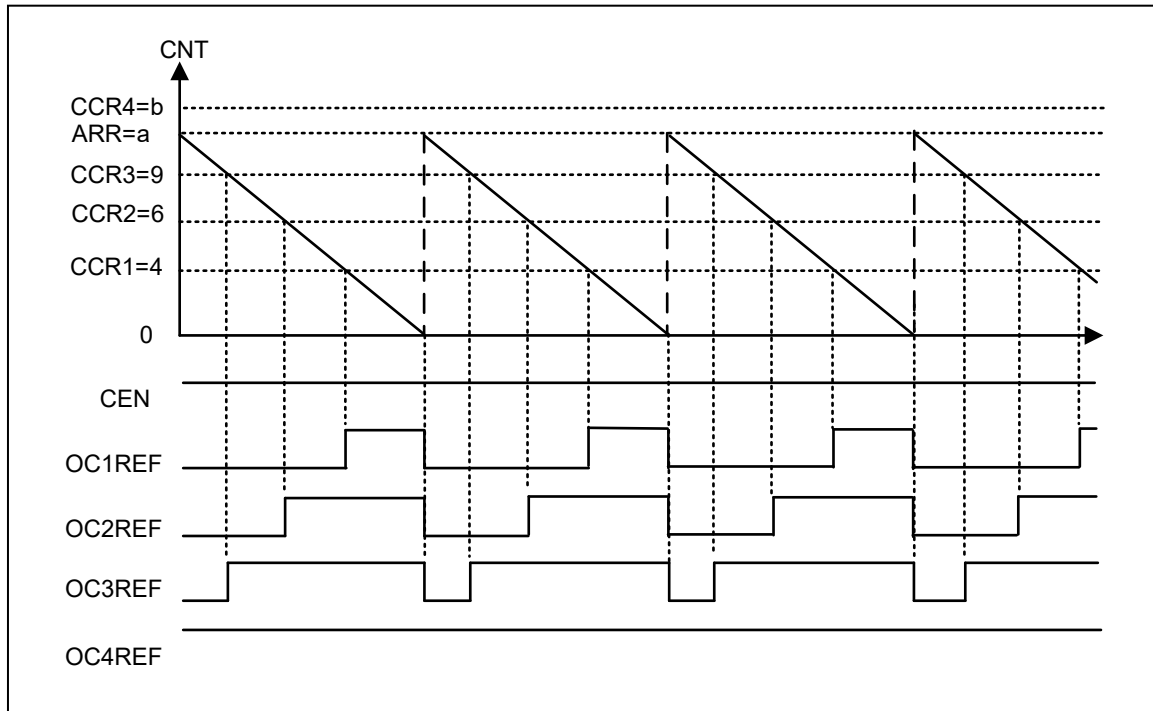


图 20-20 边沿对齐递减计数时 PWM 模式 1 的波形

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

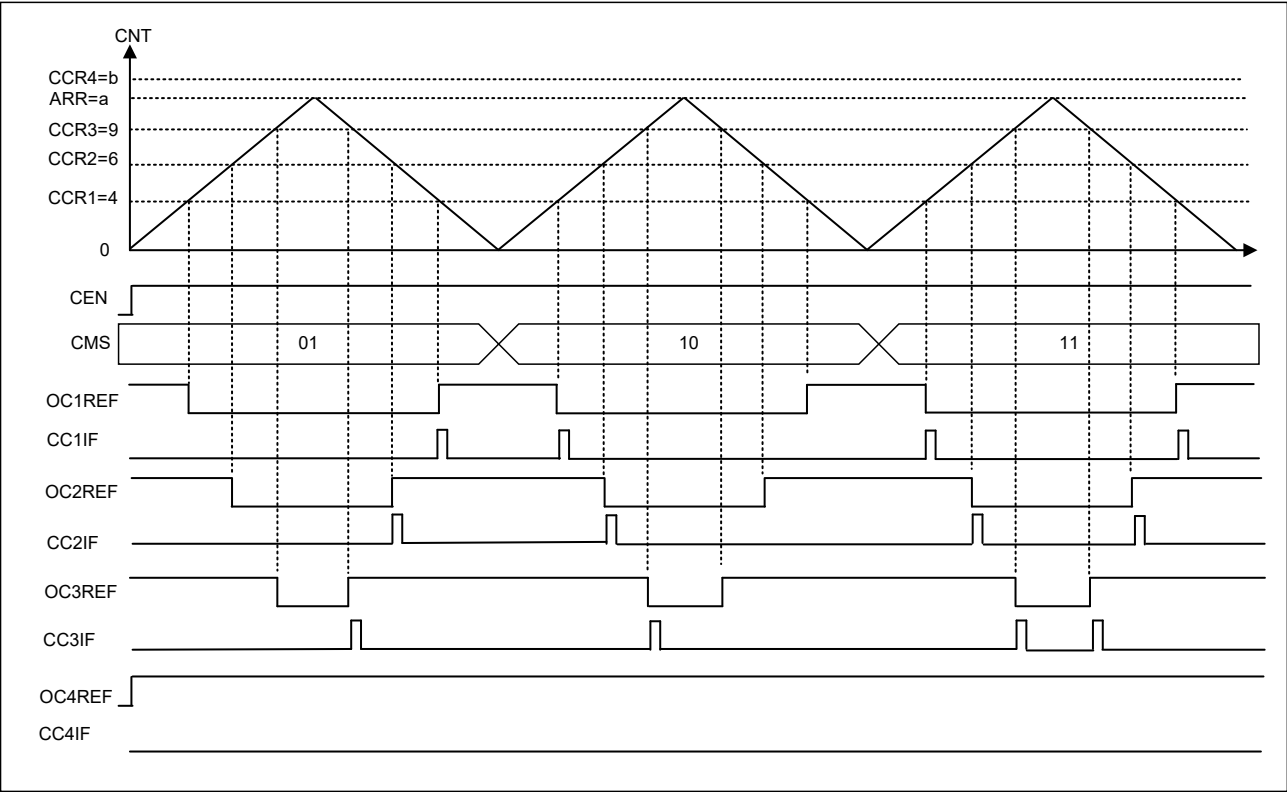


图 20-21 中央对齐 PWM 模式 1 的波形

20.4.4.3.4 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。

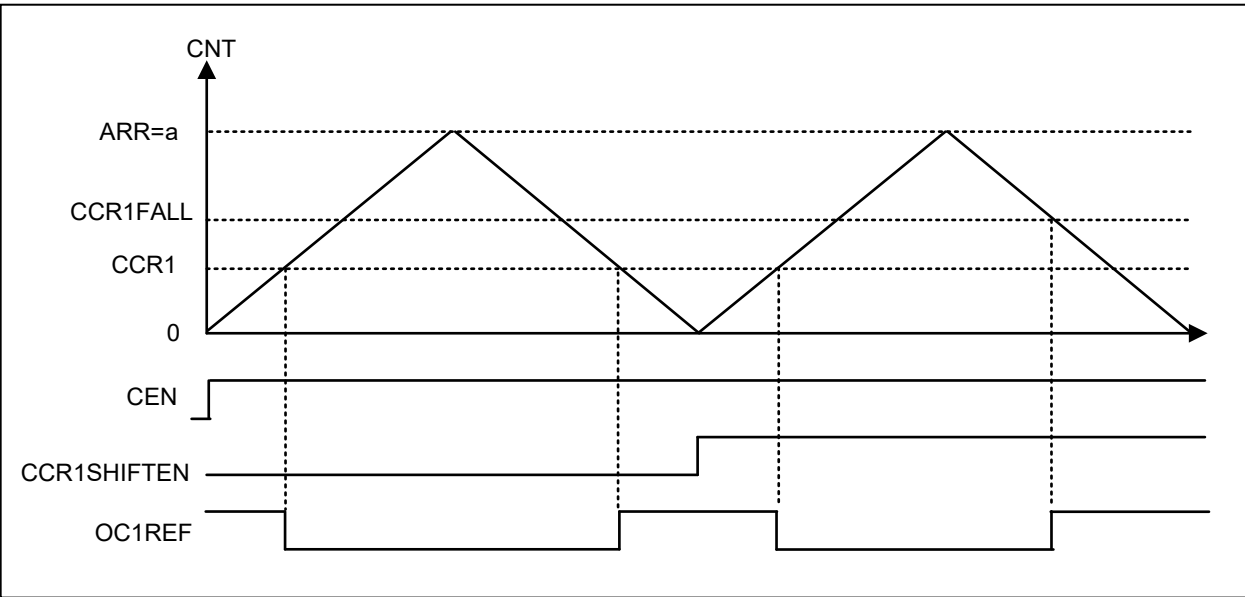


图 20-22 移相功能示意图

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。

20.4.4.3.5 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。

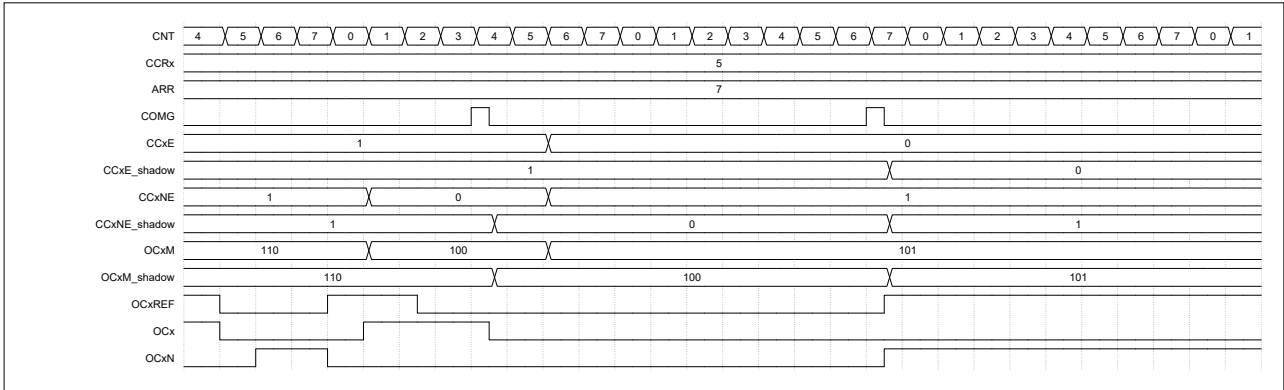


图 20-23 使用 COM 事件产生六步 PWM（OSSR = 1）

20.4.4.4 互补输出和死区插入

OCx 和 OCxN 是一对互补输出通道，TIMx 的所有通道能够输出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 20-2 死区时间计算

DTG[7: 5]	DT
0xx	DT= DTG[7: 0] x T _{dtg} (T _{dtg} =T _{DTS})



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

例如，如果 $T_{DTS}=125ns$ ，可能的死区时间为：

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

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。

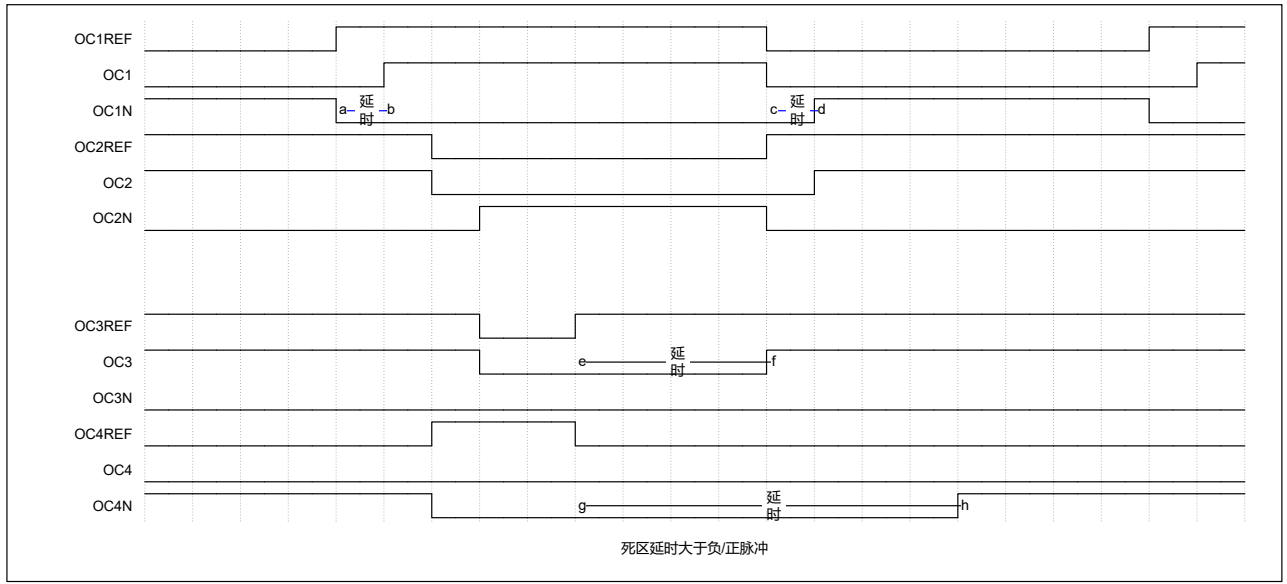


图 20-24 死区插入



20.4.4.5 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格所示。

表 20-3 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 20-4 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 20-5 当 MOE=0, OSSI=0, OSSR=0/1 时:

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

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

表 20-6 当 MOE=0, OSS1=1, OSSR=0/1 时:

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

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

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

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

发生刹车事件时, MOE 被异步清零, 此时根据 OSS1 的配置 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, OSS1=0 时, 定时器关闭输出

使能，否则打开输出使能。当使用互补输出时，输出首先置于复位状态，然后死区重新生成，在死区之后输出电平由 OISx 和 OISxN 决定。

配置 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：

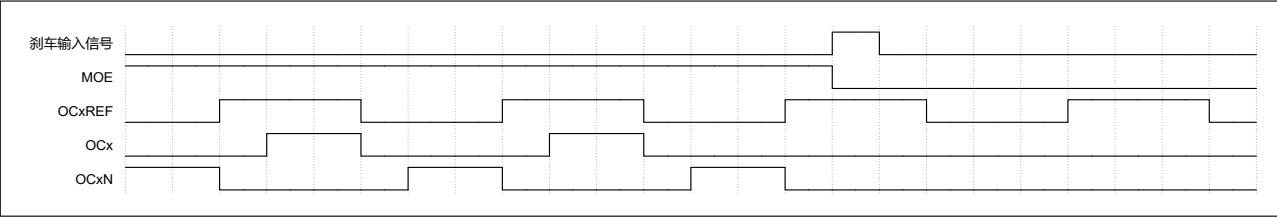


图 20-25 响应刹车的输出（OISx=0，OISxN=0）

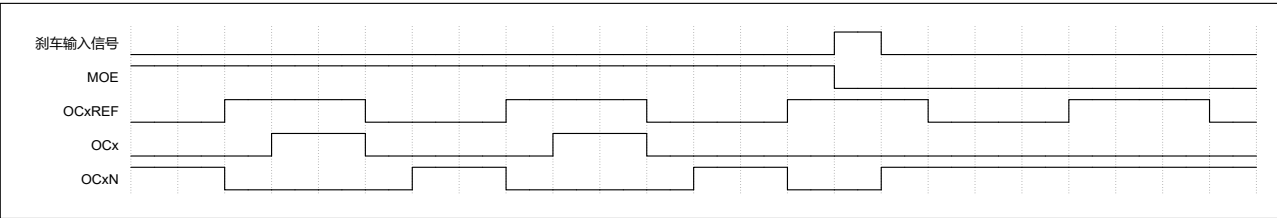


图 20-26 响应刹车的输出（OISx=0，OISxN=1）

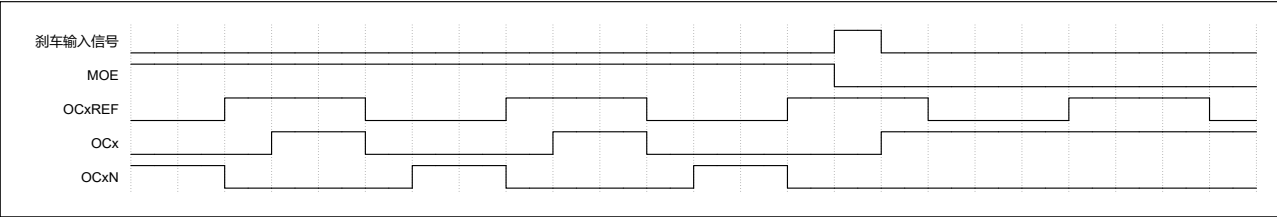


图 20-27 响应刹车的输出（OISx=1，OISxN=0）

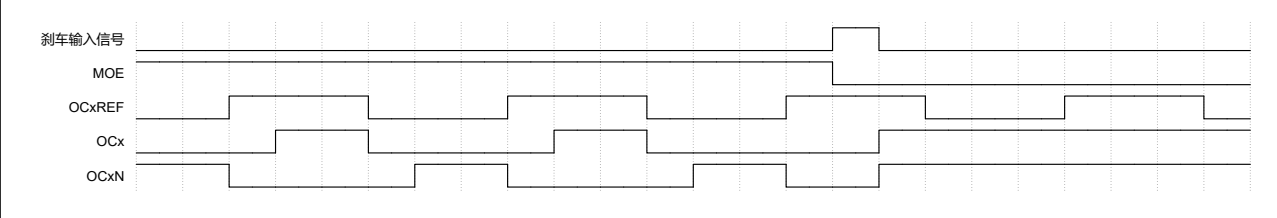


图 20-28 响应刹车的输出（OISx=1，OISxN=1）

20.4.4.6 外部事件清除 OCxREF

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

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

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

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

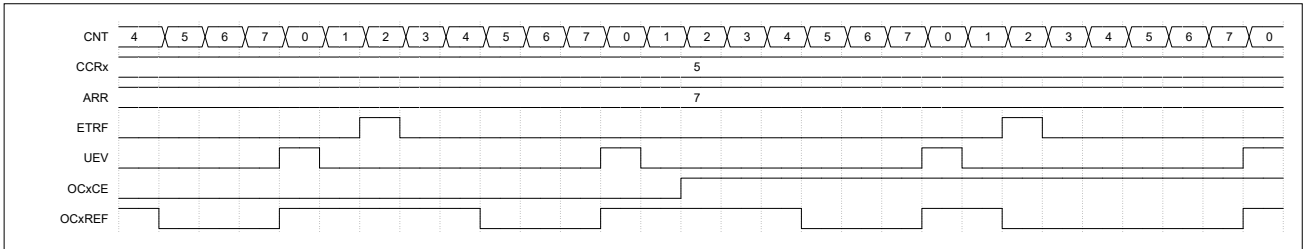


图 20-29 外部事件清除 OCxREF

20.4.4.7 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，直到下个更新事件发生或配置 CEN=0 时，计数器停止计数。

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

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

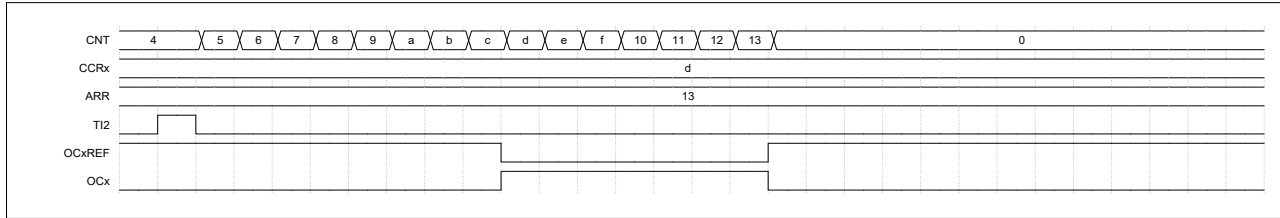


图 20-30 单脉冲模式

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

配置 TI2FP2 作为触发源：

- 1. 配置 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。

20.4.4.7.1 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。

20.4.5 从模式

20.4.5.1 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 20-7 计数方向与编码器信号的关系

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

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

- 1. 配置 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。

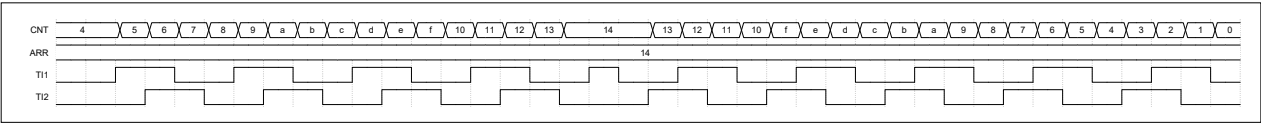


图 20-31 编码器模式下的计数器时序图

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



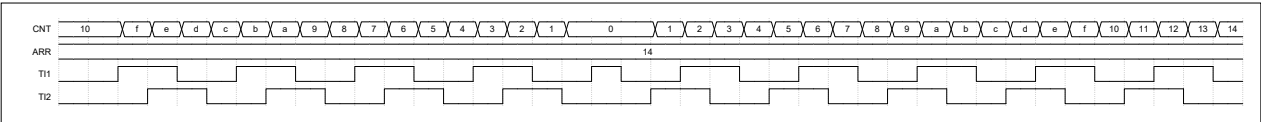


图 20-32 IC1FP1 反相编码器接口模式时序图

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。

20.4.5.2 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。
- 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。

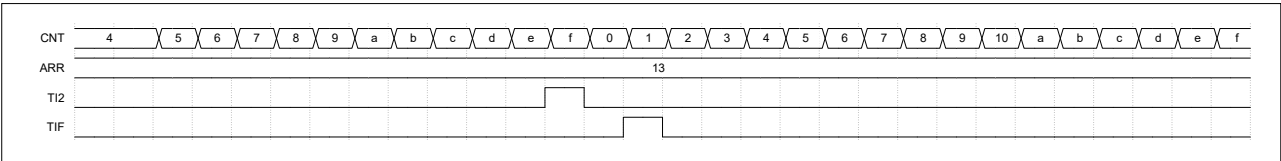


图 20-33 复位模式的控制时序图

20.4.5.3 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。
3. 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

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

下图为门控模式下 TIMx_ARR=0xf 的时序图。

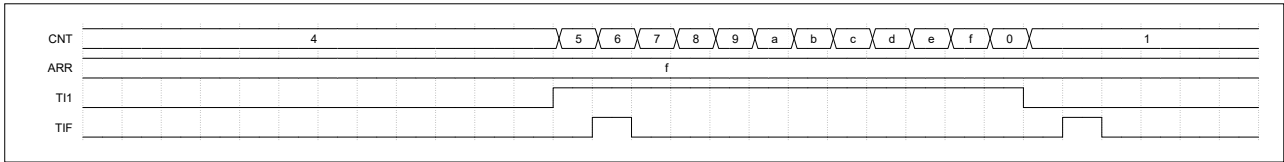


图 20-34 门控模式下的控制时序图

20.4.5.4 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。
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。

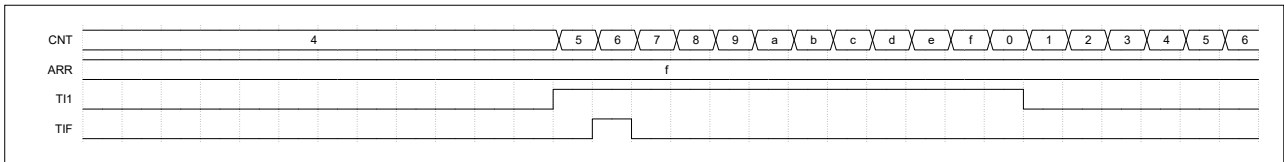


图 20-35 触发器模式下的控制时序图

20.4.5.5 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 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。

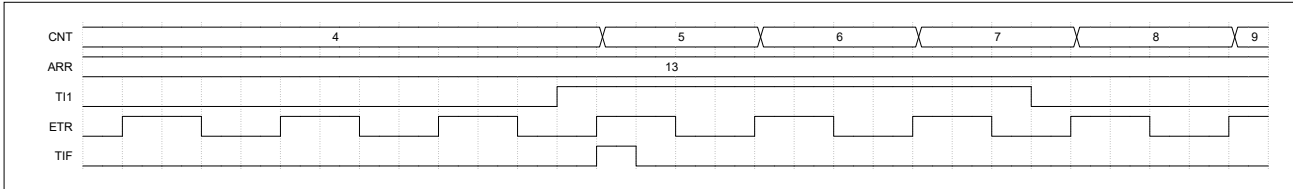


图 20-36 外部时钟模式 2+从模式（触发模式）控制时序图

20.4.6 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示意图如下：

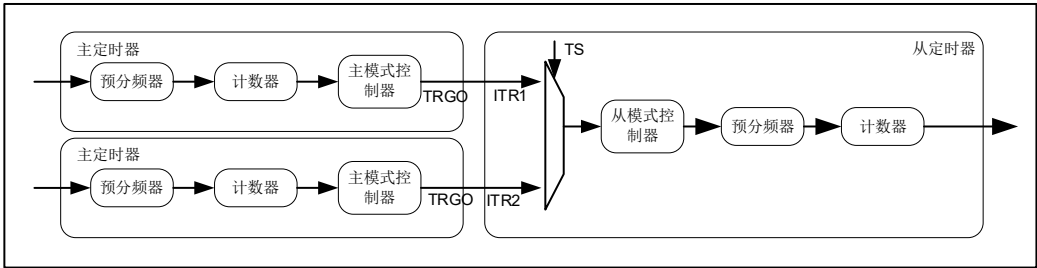


图 20-37 定时器间互联

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：主定时器作为从定时器的预分频器，时序图如下图所示：

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

- 5. 配置主定时器 CR1 寄存器的 CEN=1，启动主定时器。
- 6. 配置从定时器 CR1 寄存器的 CEN=1，启动从定时器。

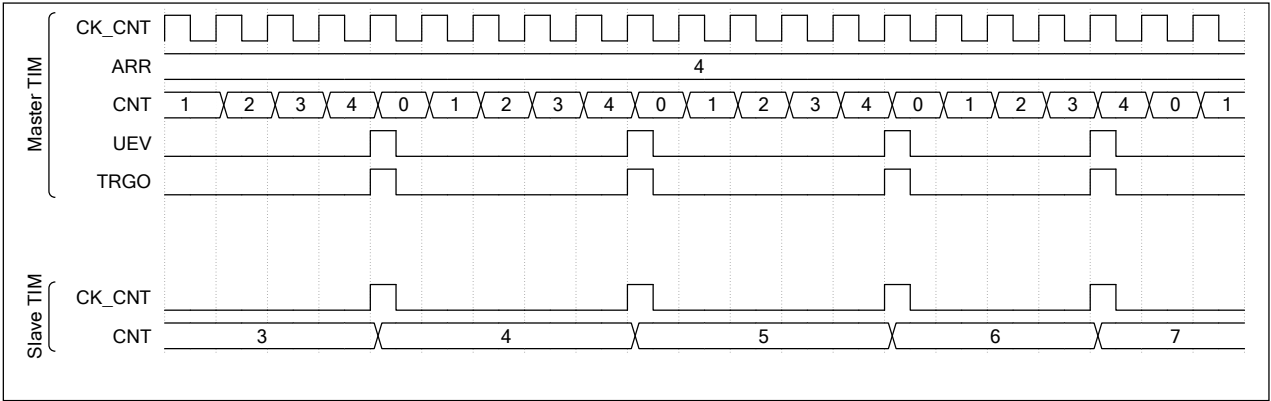


图 20-38 使用主定时器作为从定时器的预分频器

使用一个定时器使能另一个定时器

例：使用主定时器使能从定时器，时序图如下图所示：

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

- 1. 配置主定时器 CR2 寄存器的 MMS=100，选择主定时器的比较输出参考信号（OC1REF）作为触发输出（TRGO）。
- 2. 配置主定时器 CCR1 寄存器、ARR 寄存器、CCMR1 寄存器的 OC1M 位、CC1S 位等相关控制位，配置主定时器输出信号 TRGO 的输出波形。
- 3. 配置从定时器 SMCR 寄存器的 TS，选择主定时器的 OC1REF 作为从定时器的触发输入。
- 4. 配置从定时器 SMCR 寄存器的 SMS=101，选择从定时器为门控模式。
- 5. 配置从定时器 CR1 寄存器的 CEN=1，启动从定时器。
- 6. 配置主定时器 CR1 寄存器的 CEN=1，启动主定时器。

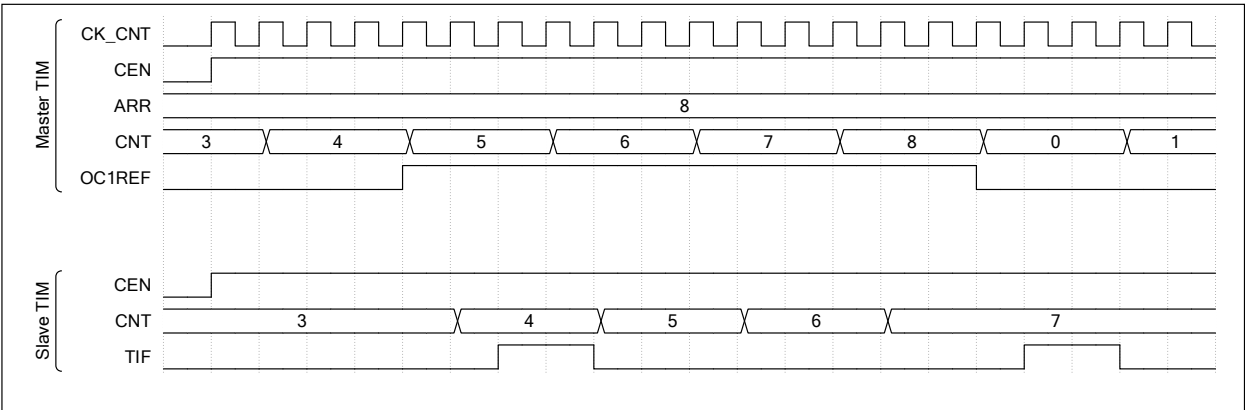


图 20-39 使用主定时器使能从定时器

使用一个定时器启动另一个定时器

例：使用主定时器的更新事件启动从定时器，时序图如下图所示：

当主定时器产生更新事件时，从定时器接收到触发信号，从定时器的 CEN 由硬件自动置 1，从定时器

的计数器开始计数。两个 TIM 的计数时钟为 CK_CNT 的三分频。具体配置如下：

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

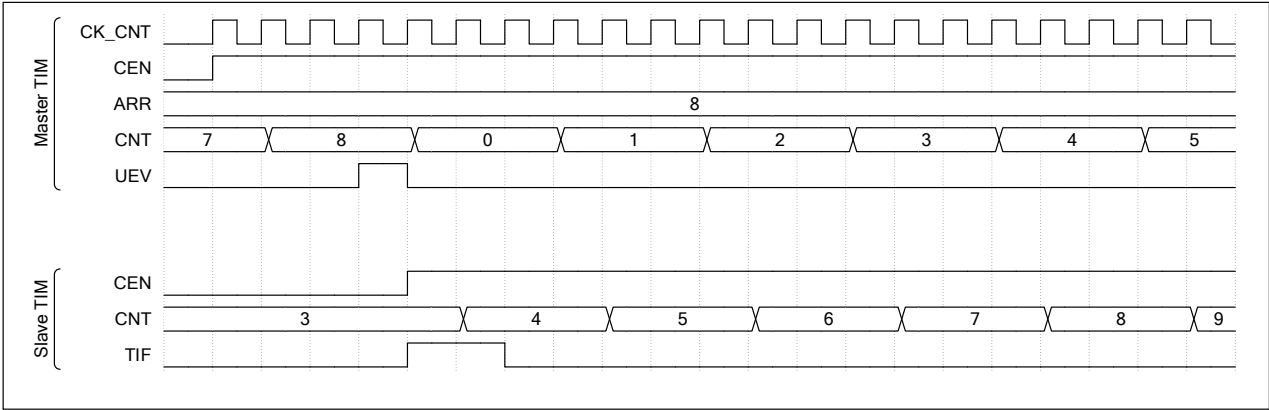


图 20-40 使用主定时器的更新事件启动从定时器

使用一个外部触发同步启动两个定时器

例：使用主定时器的 TI1 上升沿，启动主定时器的同时启动从定时器，时序图如下图所示：

为了确保两个定时器同时开启，主定时器必须在主/从模式下配置。具体配置如下：

- 1. 配置主定时器 CR2 寄存器的 MMS=001，作为主模式时，将主定时器的使能信号 CEN 作为触发输出（TRGO）。
- 2. 配置主定时器 SMCR 寄存器的 TS=100，设置主定时器作为从模式时，将 TI1 作为触发输入。
- 3. 配置主定时器 SMCR 寄存器的 SMS=110，选择主定时器为触发模式。
- 4. 配置从定时器 SMCR 寄存器的 TS，选择主定时器的触发输出作为从定时器的触发输入。
- 5. 配置从定时器 SMCR 寄存器的 SMS=110，选择从定时器为触发模式。

当主定时器的 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。

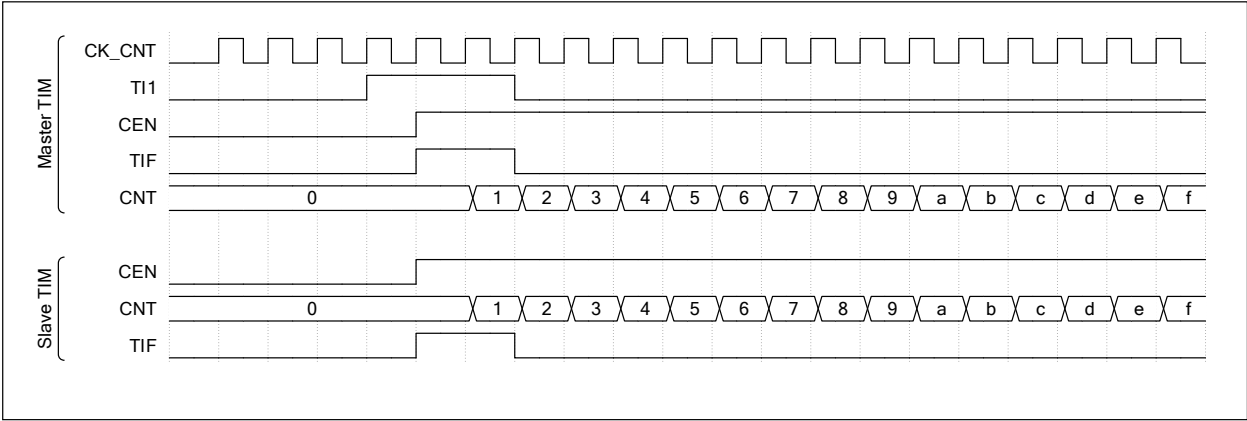


图 20-41 主定时器的 TI1 同步启动主定时器和从定时器

20.4.7 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。

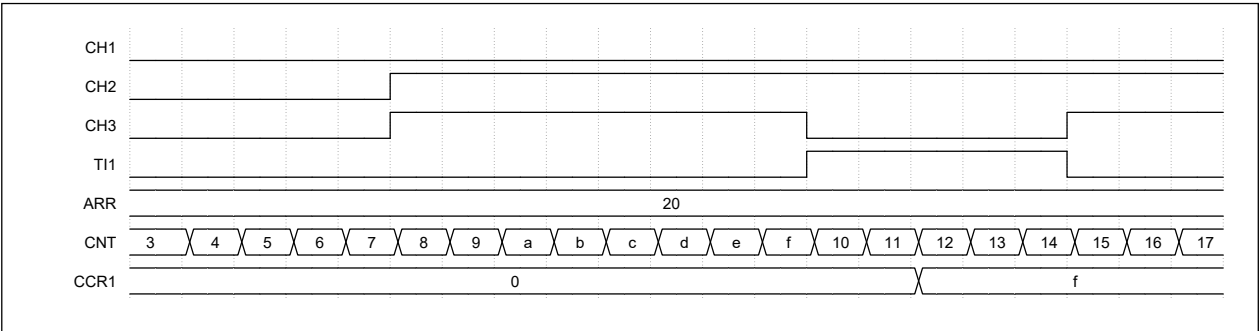


图 20-42 （TI1 异或输入）输入捕获波形图

20.4.7.1.1 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：

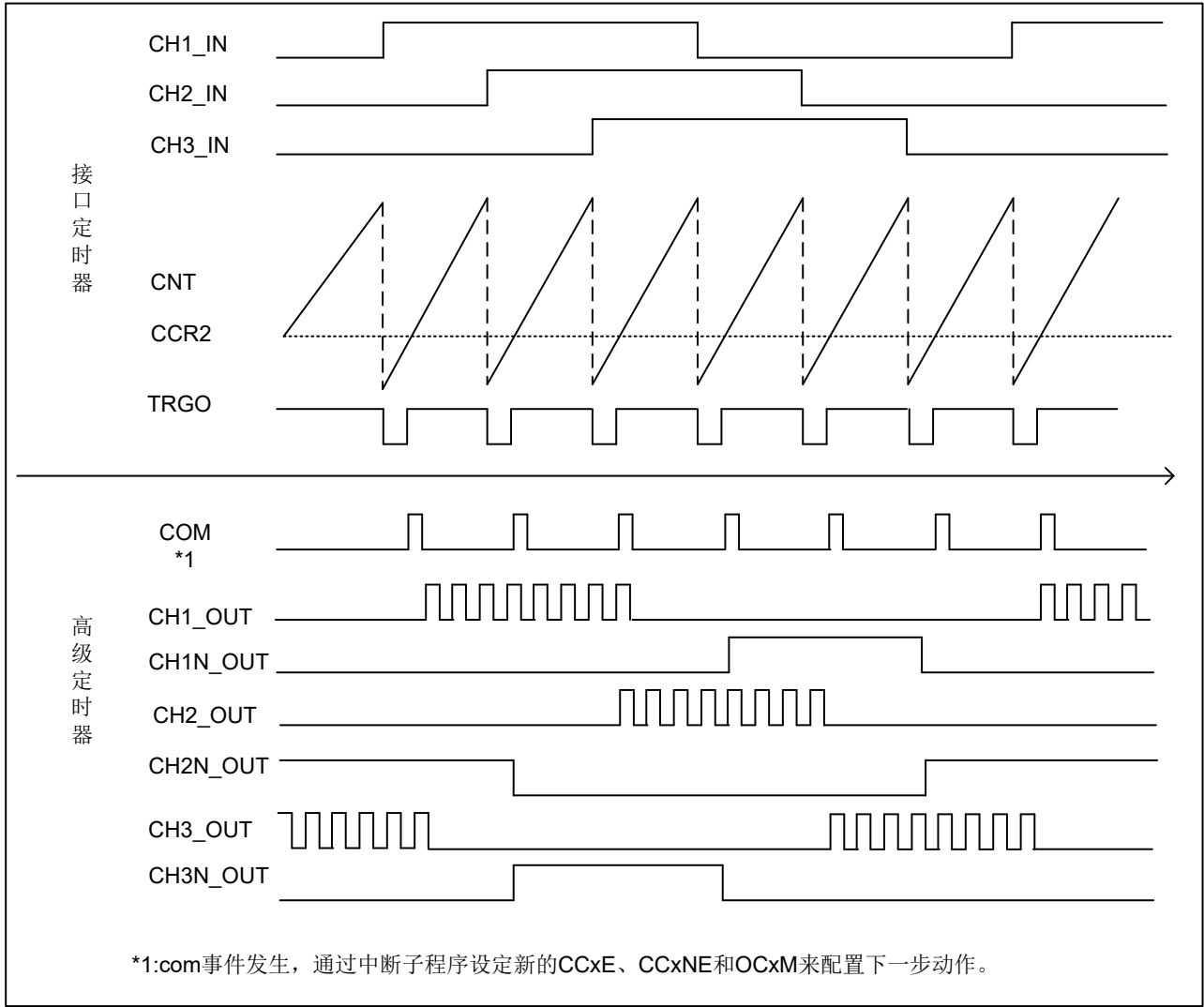


图 20-43 霍尔传感器接口实例

20.4.8 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）

20.4.9 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 20-8 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

20.4.10 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。

20.5 寄存器描述

表 20-9 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

Offset	Acronym	Register Name	Reset
0x74	TIMx_BKINF	刹车输入滤波寄存器	0x0000 0000

20.5.1 TIMx_CR1 控制寄存器 1

偏移地址：0x00

复位值：0x0000

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

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（ETR, Tlx）所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT\_CK}$ 01: $t_{DTS} = 2 \times t_{INT\_CK}$ 10: $t_{DTS} = 4 \times t_{INT\_CK}$ 11: 保留，不要使用这个配置
7	ARPE	自动重载预装载使能（Auto-reload preload enable） 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 5	CMS	中央对齐模式选择（Center-aligned mode selection） 00: 边沿对齐模式。计数方向取决于 DIR 位 01: 中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式，只在计数器递减计数时比较中断标志位被置 1 10: 中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式，只在计数器递增计数时比较中断标志位被置 1 11: 中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式，在计数器递增和递减计数时比较中断标志位均被置 1 注：计数过程中，不允许更改此位。
4	DIR	计数方向（Direction） 0: 计数器递增计数 1: 计数器递减计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	单脉冲模式（One pulse mode） 0: 禁止单脉冲模式，在发生更新事件时，计数器继续计数 1: 使能单脉冲模式，在发生下一次更新事件或软件清除 CEN 位时，计数器停止计数

Bit	Field	Description
2	URS	更新请求源 (Update request source) 软件配置该位, 选择更新事件源。 0: 以下事件可产生一个更新中断或 DMA 请求: - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 只有计数器上溢/下溢才产生一个更新中断或 DMA 请求
1	UDIS	禁止更新 (Update disable) 该位用来允许或禁止更新事件的产生 0: 允许更新事件 (UEV) 1: 禁止更新事件。不产生更新事件, 影子寄存器 (ARR、PSC、CCR _x) 保持值不变。如果设置了 EGR_UG 位为 1, 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。

20.5.2 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
OIS4N	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		rw

Bit	Field	Description
15	OIS4N	输出空闲状态 4 (OC4N 输出)。参见 OIS1N 位。
14	OIS4	输出空闲状态 4 (OC4 输出)。参见 OIS1 位。
13	OIS3N	输出空闲状态 3 (OC3N 输出)。参见 OIS1N 位。
12	OIS3	输出空闲状态 3 (OC3 输出)。参见 OIS1 位。
11	OIS2N	输出空闲状态 2 (OC2N 输出)。参见 OIS1N 位。
10	OIS2	输出空闲状态 2 (OC2 输出)。参见 OIS1 位。
9	OIS1N	输出空闲状态 1 (OC1N 输出) (Output Idle state 1) 0: 当 MOE = 0 时, 死区后 OC1N = 0 1: 当 MOE = 0 时, 死区后 OC1N = 1 注: 已经设置了 LOCK (TIMx_BKR 寄存器) 级别 1、2 或 3 后, 该位不能被修改。

Bit	Field	Description
8	OIS1	输出空闲状态 1 (OC1 输出) (Output Idle state 1) 0: 当 MOE=0, 若 OC1N 有效, 则在死区时间后 OC1 = 0 1: 当 MOE=0, 若 OC1N 有效, 则在死区时间后 OC1 = 1 注: 已经设置了 LOCK (TIMx_BKR 寄存器) 级别 1、2 或 3 后, 该位不能被修改。
7	TI1S	TI1 选择 (TI1 selection) 0: TIMx_CH1 管脚连到 TI1 输入 1: TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入
6: 4	MMS	主模式选择 (Master mode selection) 这些位控制 TRGO 信号的选择, 用于选择在主模式下送到从定时器的同步信息: 000: 复位 TIMx_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001: 使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO), 计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新 更新事件被选为 TRGO。 011: 捕获/比较脉冲 发生一次捕获或一次比较成功时, 触发输出送出一个 TRGO 信号。 100: 比较 OC1REF 信号被用于作为触发输出 (TRGO) 101: 比较 OC2REF 信号被用于作为触发输出 (TRGO) 110: 比较 OC3REF 信号被用于作为触发输出 (TRGO) 111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)
3	CCDS	DMA 请求源选择 (Capture/compare DMA selection) 0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求 1: 发生更新事件时, 发送 CCx 的 DMA 请求 注: 仅适用于有内置 DMA 的产品
2	CCUS	捕获/比较控制更新源选择 (Capture/compare control update selection) 0: CCPC=1 时, 只能配置 COMG=1 更新。 1: CCPC=1 时, 可以通过配置 COMG=1 或检测到 TRGI 上的一个上升沿更新。 注: 此位只在通道为互补输出时有效。
1	Reserved	保留, 必须保持复位值。
0	CCPC	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位预装载禁用 1: CCxE, CCxNE 和 OCxM 位预装载使能 注: 此位只在通道为互补输出时有效。

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

Bit	Field	Description
11: 8	ETF	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$ 注: 仅适用于支持外部触发的产品。
7	MSM	主/从模式 (Master/slave mode) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟, 以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步, 该功能可以把几个定时器同步到一个单一的外部事件。
6: 4	TS	触发选择 (Trigger selection) 触发输入源选择。 000: 内部触发 0 (ITR0) 001: 内部触发 1 (ITR1) 010: 内部触发 2 (ITR2) 011: 内部触发 3 (ITR3) 100: 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。
3	OCCS	比较输出信号 (OCxREF) 清除选择 (Output compare clear selection) 在 PWM 模式下, 清除比较输出 (OCxREF)。 0: 外部触发信号作为清除信号 1: 比较器 (COMP) 输出作为清除信号 注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。

Bit	Field	Description
2: 0	SMS	从模式选择 (Slave mode selection) 当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。 000: 关闭从模式 - 如果 CEN = 1, 则预分频器直接由内部时钟驱动。 001: 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。 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。 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。 100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新事件。 101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器开始计数并且产生一个更新事件。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。 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。 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并且产生一个更新事件。 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。

注: 内部触发连接关系参照芯片特定配置章节 TIM 之间的互联表。

20.5.4 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	允许比较 5 的 DMA 请求 (Compare 5 DMA request enable) 0: 禁止比较 5 的 DMA 请求 1: 允许比较 5 的 DMA 请求 注: 仅适用于有内置 DMA 且有通道 5 的产品。

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

Bit	Field	Description
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

20.5.5 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。

Bit	Field	Description
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, 或门控模式下的任一边沿) 时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断产生
5	COMIF	COM 中断标记 (COM interrupt flag) 当产生 COM 事件 (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。
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1 (在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0 或读取 TIMx_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1

Bit	Field	Description
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 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。

20.5.6 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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

Bit	Field	Description
31: 17	Reserved	保留, 必须保持复位值。
16	CC5G	产生比较 5 事件 (Compare 5 generation) 参考 CC1G 描述。
15: 8	Reserved	保留, 必须保持复位值。
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。

Bit	Field	Description
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：当前的计数器值被捕获至 TIMx_CCR1 寄存器，CC1IF 置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF = 1。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0，如果选择了中央对齐或递增计数模式，计数器被清 0；如果选择递减计数模式，计数器将载入自动重载值。预分频计数器将同时被清除。

20.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址：0x18

复位值：0x0000

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

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CCxS 定义。该寄存器 CCxS

以外其它位的作用在输入模式和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

比较输出模式：

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

Bit	Field	Description
6: 4	OC1M	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。 000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为高电平。 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为低电平。 011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为低。强制 OC1REF 为低电平。 101: 强制为高。强制 OC1REF 为高电平。 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平。 注 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。 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。
3	OC1PE	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。 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。 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。 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。

Bit	Field	Description
2	OC1FE	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。 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
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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	输入捕获 2 滤波器 (Input capture 2 filter) 参考 IC1F 的描述
11: 10	IC2PSC	输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 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)

Bit	Field	Description
7: 4	IC1F	通道 1 输入捕获滤波器 (Input capture 1 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC1PSC	通道 1 输入/捕获预分频器 (Input capture 1 prescaler) 该位定义了 IC1 的预分频系数。当 $CC1E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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)

20.5.8 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 输入有效电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	通道 3 比较输出模式 (Output compare 3 mode) 该位定义了输出参考信号 OC3REF 的动作, 而 OC3REF 决定了 OC3、OC3N 的值。OC3REF 是高电平有效, 而 OC3、OC3N 的有效电平取决于 CC3P、CC3NP 位。 000: 冻结。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为高电平 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为低电平 011: 匹配时翻转。当 TIMx_CCR3=TIMx_CNT 时, 翻转 OC3REF 的电平 100: 强制为低。强制 OC3REF 为低电平 101: 强制为高。强制 OC3REF 为高电平 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平。 注 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。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能 (Output compare 3 preload enable) 0: 禁止 TIMx_CCR3 寄存器的预装载功能, 写入 TIMx_CCR3 寄存器的数值立即生效 1: 开启 TIMx_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR3 的预装载值在更新事件到来时生效 注 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。 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。

Bit	Field	Description
2	OC3FE	通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。 0: 禁止通道 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	输入捕获 4 滤波器 (Input capture 4 filter) 参考 IC3F 的描述
11: 10	IC4PSC	输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述
9: 8	CC4S	通道 4 捕获/比较选择 (Capture/Compare 4 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入, IC4 映射在 TI4 上 10: 通道 4 被配置为输入, IC4 映射在 TI3 上 11: 通道 4 被配置为输入, IC4 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC3F	通道 3 输入捕获滤波器 (Input capture 3 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC3PSC	通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 $CC3E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

20.5.9 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	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
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	CC4NE	通道 4 输入/捕获输出使能 (Capture/Compare 4 output enable) 参考 CC1NE 的描述。
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。

Bit	Field	Description
3	CC1NP	通道 1 输入/捕获互补输出极性 (Capture/Compare 1 complementary output polarity) 通道 1 配置为输出时, 此位定义了输出信号极性: 0: OC1N 高电平有效 1: OC1N 低电平有效 通道 1 配置为输入时, CC1P/CC1NP 配合使用定义了输入信号极性和电平, 详细参考 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。
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 20-10 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

20.5.10 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)											

20.5.11 TIMx_PSC 预分频器

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC															
rw															
Bit		Field		Description											
15: 0		PSC		预分频器的值 (Prescaler value) 计数器的时钟频率 (ck_cnt) = $f_{ck\_psc} / (PSC + 1)$ 当发生更新事件时，PSC 的值装入当前预分频寄存器。											

20.5.12 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。											

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

20.5.14 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	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。

20.5.15 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。

20.5.16 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。

20.5.17 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。									

20.5.18 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，DOE、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) 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。									
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									

Bit	Field	Description
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, 刹车使能前应先配置 TIMx_BKINF 寄存器中 BKIN_SEL 位, 选择刹车源。
11	OSSR	运行模式下“关闭状态”选择 (Off-state selection for Run mode) 该位仅适用于当 MOE = 1 且通道为互补输出。 0: 当定时器不工作时, 禁止 OC/OCN 输出 1: 当定时器不工作时, 如果 CCxE = 1 或 CCxNE = 1, 首先开启 OC/OCN 并输出无效电平, 然后置位 OC/OCN 输出使能信号。 注: 当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 2 时, 该位不能被修改。
10	OSSI	空闲模式下“关闭状态”选择 (Off-state selection for Idle mode) 该位仅适用于当 MOE = 0 且通道设为输出时。 0: 当定时器不工作时, 禁止 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。

Bit	Field	Description
9: 8	LOCK	锁定设置 (Lock configuration) 该位定义了寄存器的写保护功能。 00: 写保护功能关闭, 寄存器无写保护 01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 DOE、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。
7: 0	DTG	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。 注: 当 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3 时, 不能修改这些位。

20.5.19 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 20.4.10 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。

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

20.5.20 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 20.4.10 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。

20.5.21 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。 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。
2: 0	Reserved	保留，必须保持复位值。

20.5.22 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。

20.5.23 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_	CCR4_	CCR3_	CCR2_	CCR1_	CCDRE
										SHIFT_	SHIFT_	SHIFT_	SHIFT_	SHIFT_	PE
										EN	EN	EN	EN	EN	
										rw	rw	rw	rw	rw	rw

Bit	Field	Description
15: 6		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。

20.5.24 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。

20.5.25 TIMx_BKINF 刹车输入滤波寄存器

偏移地址: 0x74

复位值: 0x0000 0000

TIM1:

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

Bit	Field	Description
31: 16	Reserved	保留, 必须保持复位值。
15: 13	COMPBKIN_SEL	COMP 刹车输入选择 (COMP break input sel) COMPBKIN_SEL[2]: 0: COMP3_OUT 刹车输入无效 1: COMP3_OUT 刹车输入有效 COMPBKIN_SEL[1]: 0: COMP2_OUT 刹车输入无效 1: COMP2_OUT 刹车输入有效 COMPBKIN_SEL[0]: 0: COMP1_OUT 刹车输入无效 1: COMP1_OUT 刹车输入有效
12: 10	Reserved	保留, 必须保持复位值。
9: 6	IOBKIN_SEL	IO 刹车输入选择 (IO break input sel) IOBKIN_SEL[3]: 0: TIM1_BKIN4 刹车输入无效 1: TIM1_BKIN4 刹车输入有效 IOBKIN_SEL[2]: 0: TIM1_BKIN3 刹车输入无效 1: TIM1_BKIN3 刹车输入有效 IOBKIN_SEL[1]: 0: TIM1_BKIN2 刹车输入无效 1: TIM1_BKIN2 刹车输入有效 IOBKIN_SEL[0]: 0: TIM1_BKIN1 刹车输入无效 1: TIM1_BKIN1 刹车输入有效

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

TIM8:

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

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

Bit	Field	Description
14: 13	COMPBKIN_SEL	COMP 刹车输入选择 (COMP break input sel) COMPBKIN_SEL[1]: 0: COMP2_OUT 刹车输入无效 1: COMP2_OUT 刹车输入有效 COMPBKIN_SEL[0]: 0: COMP1_OUT 刹车输入无效 1: COMP1_OUT 刹车输入有效
12: 8	Reserved	保留, 必须保持复位值。
7: 6	IOBKIN_SEL	IO 刹车输入选择 (IO break input sel) IOBKIN_SEL[1]: 0: TIM8_BKIN2 刹车输入无效 1: TIM8_BKIN2 刹车输入有效 IOBKIN_SEL[0]: 0: TIM8_BKIN1 刹车输入无效 1: TIM8_BKIN1 刹车输入有效
5	CSSBKIN_SEL	CSS 刹车输入选择 (CSS break input sel) CSSBKIN_SEL: 0: CSS 刹车输入无效 1: CSS 刹车输入有效
4: 1	BKINF	BKIN 数字滤波采样频率 (break input filter) 0000: 2 周期 0001: 4 周期 0010: 8 周期 0011: 16 周期 0100: 32 周期 0101: 64 周期 0110: 128 周期 0111: 256 周期 1000: 384 周期 1001: 512 周期 1010: 640 周期 1011: 768 周期 1100: 896 周期 1101: 1024 周期 1110: 1152 周期 1111: 1280 周期 注: 更改档位时应先关闭刹车滤波使能位。
0	BKINF	BKIN 数字滤波使能 (break input filter enable) 1: 使能 BKIN 管脚数字滤波 0: 禁止 BKIN 管脚数字滤波 注: 滤波使能信号在刹车条件配置之后打开。 滤波功能用于过滤电平。

21 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。

21.1 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）。

21.2 功能框图

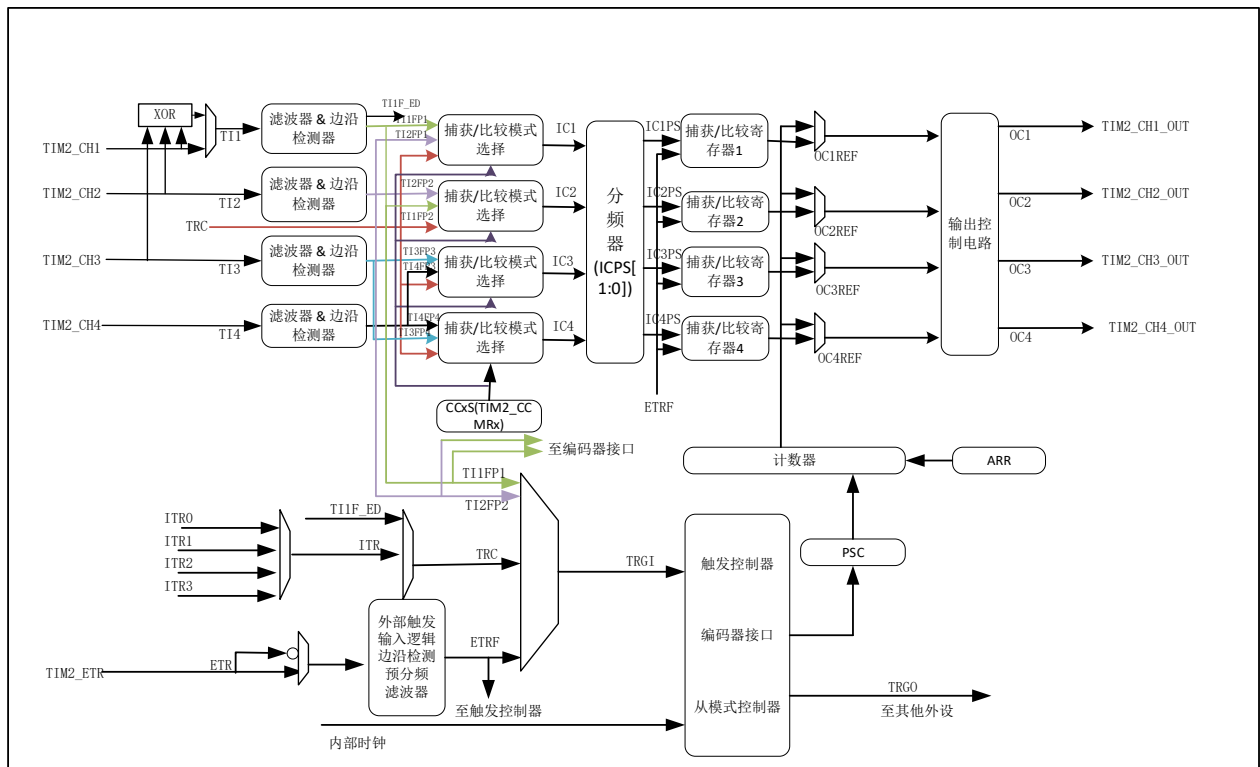


图 21-1 TIMx 结构图

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

21.3 主要特征

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

21.4 功能描述

21.4.1 时钟

21.4.1.1 时钟选择

计数器的时钟源有以下几种：

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

上述几种时钟选择示意图如下：

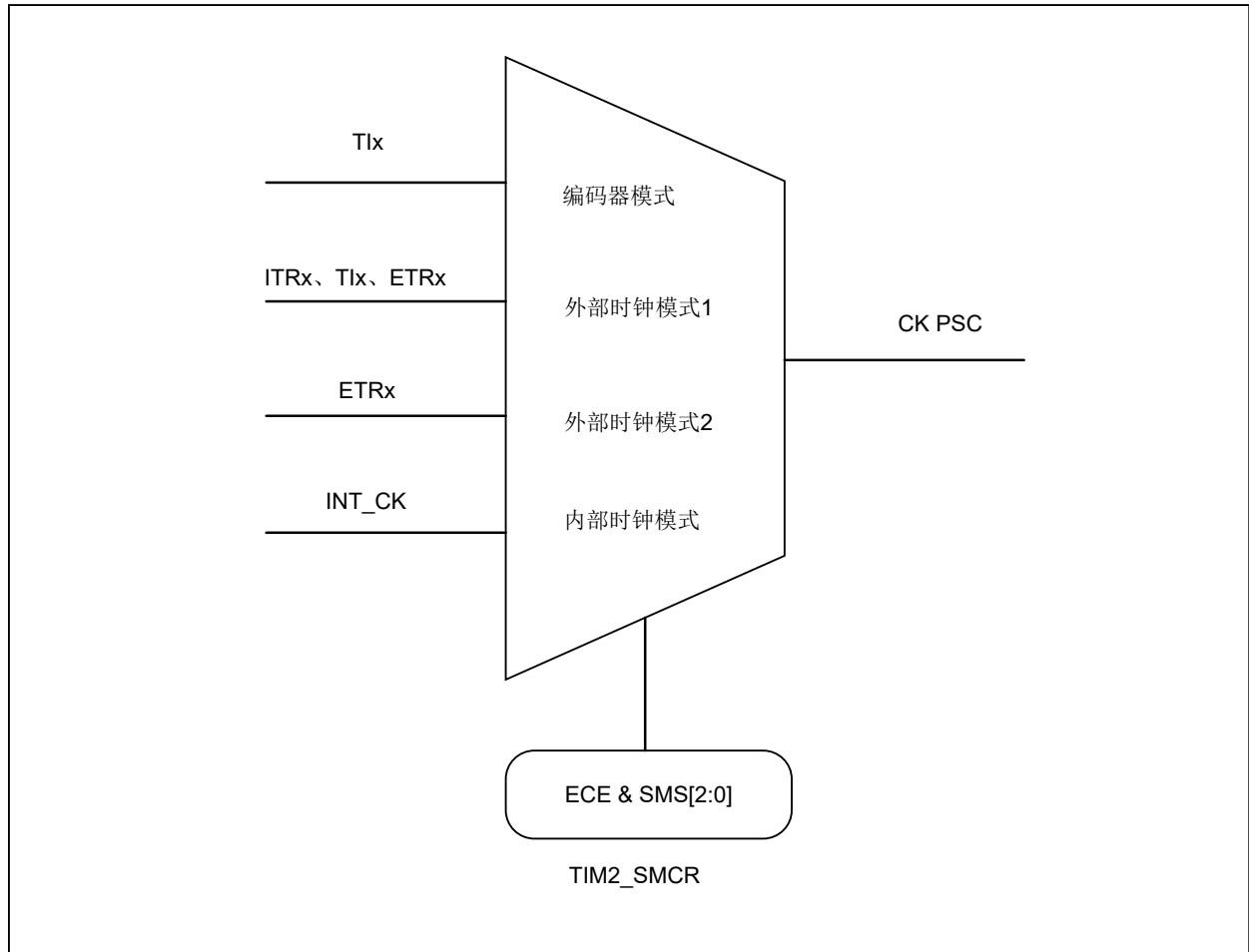


图 21-2 时钟选择

21.4.1.1.1 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。

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

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

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

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

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

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

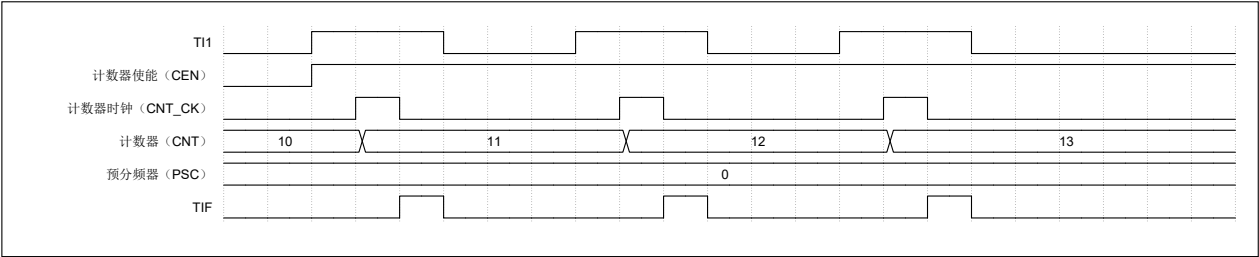


图 21-3 外部时钟模式 1 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）为复位值。

21.4.1.1.3 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。

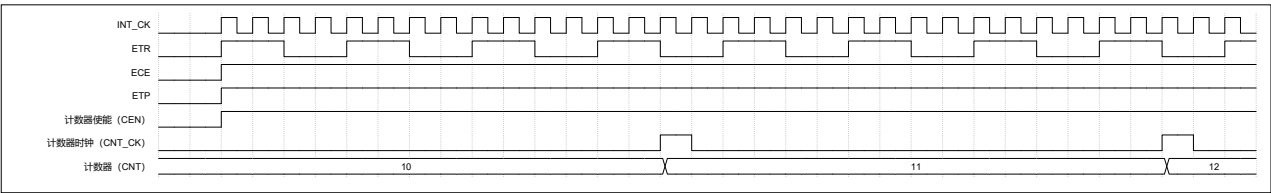


图 21-4 外部时钟模式 2 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）为复位值。

21.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

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）。

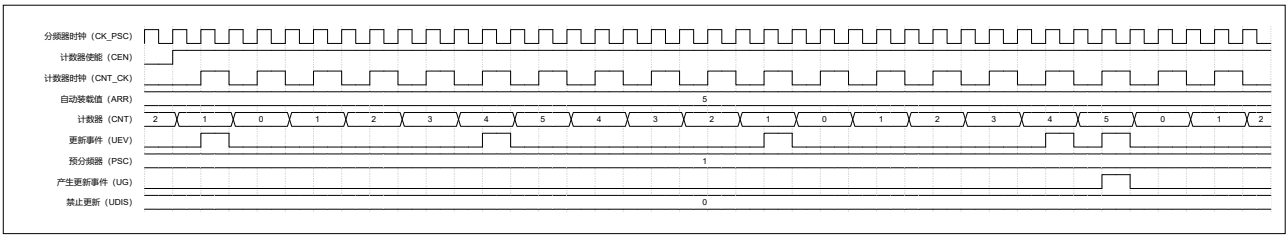


图 21-10 中央计数模式（UDIS=0）

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。

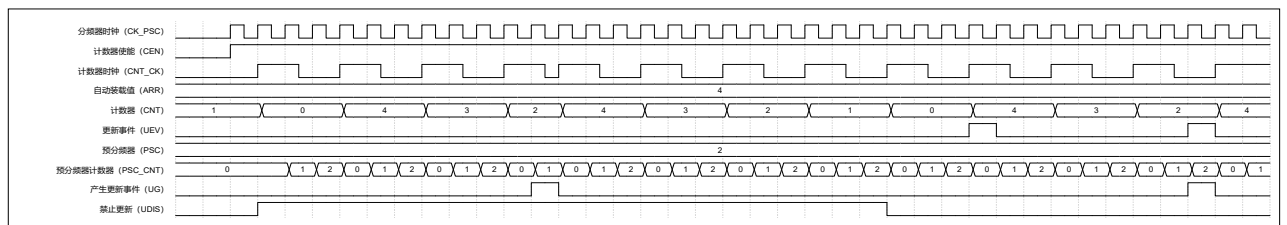


图 21-11 中央计数模式（UDIS=1 禁止产生更新事件）

21.4.2 输入捕获

21.4.2.1 输入捕获

输入捕获部分包括数字滤波器、多路复用、预分频器等，其结构如下图所示：

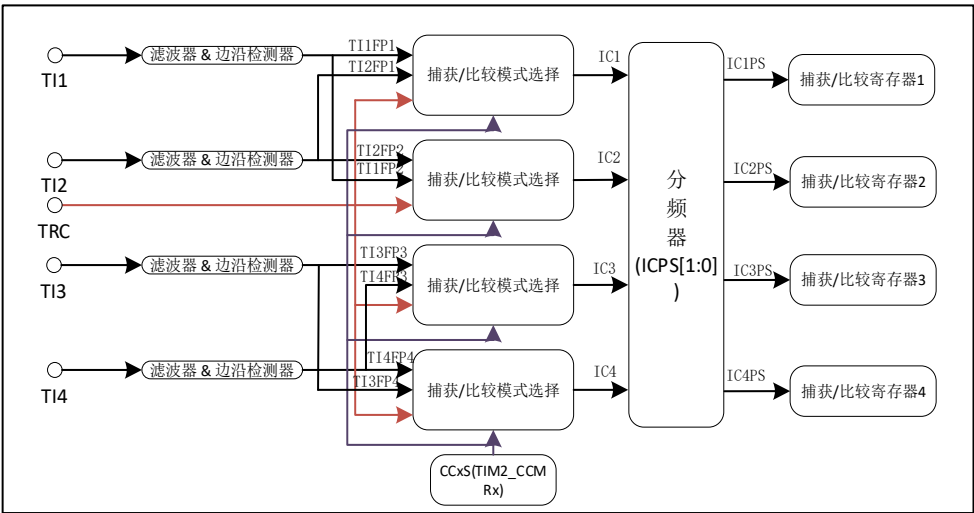


图 21-12 TIMx 输入捕获结构图

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 21-1 数字滤波器宽度与 **ICxF** 的对应关系表

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

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

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

1. 配置 **TIMx_CCMR1** 寄存器的 **CC1S=01**，**CC1** 通道被配置为输入，**IC1** 映射在 **TI1** 上。
2. 配置 **TIMx_CCMR1** 寄存器的 **IC1F[3: 0]**，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 **TIMx_CCER** 寄存器的 **CC1P=0**，选择捕获发生在 **TI1** 信号的上升沿。
4. 配置 **TIMx_CCMR1** 寄存器的 **IC1PSC[1:0]**，选择预分频系数。
5. 配置 **TIMx_CCER** 寄存器的 **CC1E = 1**，开启输入/捕获通道 1 的捕获使能。
6. 配置 **TIMx_DIER** 寄存器的 **CC1IE=1**，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 **DMA**，配置 **TIMx_DIER** 寄存器的 **CC1DE=1**，允许捕获/比较通道 1 的 **DMA** 请求。

注：

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

21.4.2.2 PWM 捕获

PWM 输入模式的操作配置与一般输入捕获有以下不同点：

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

例：测量 TI1 的 PWM 信号的宽度（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。

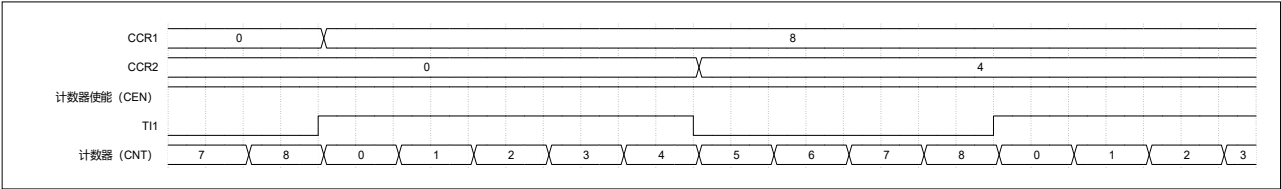


图 21-13 PWM 输入模式时序

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。

21.4.3 比较输出

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

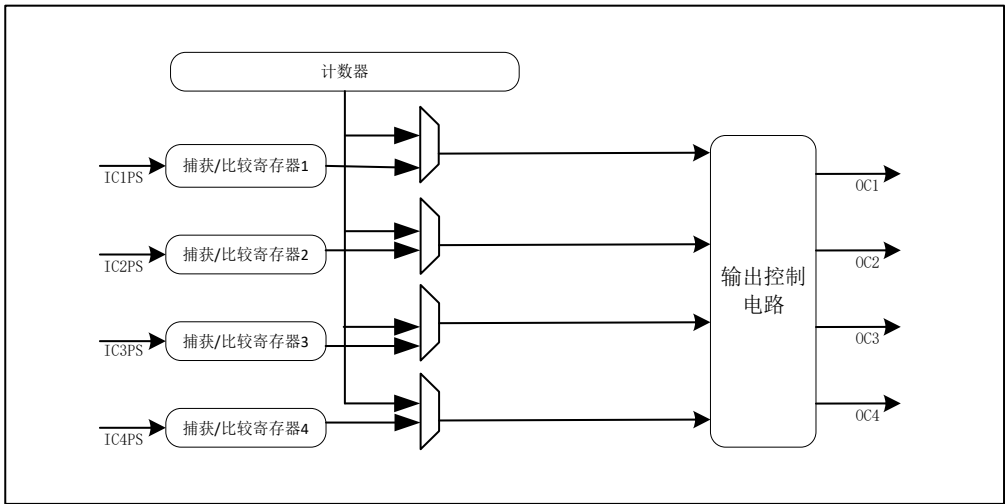


图 21-14 比较输出部分结构图

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。

21.4.3.1 强制输出

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

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

21.4.3.2 比较输出

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

例如，当计数器与捕获/比较寄存器的内容匹配时，比较输出模式下的操作如下：

- 1. 在比较匹配时，OCxM 的值不同，输出通道 x 信号 OCx 的操作不同：
 - ◆ OCxM = 000: OCx 信号保持它的电平
 - ◆ OCxM = 001: OCx 信号被设置成有效电平
 - ◆ OCxM = 010: OCx 信号被设置成无效电平
 - ◆ OCxM = 011: OCx 信号进行翻转
- 2. 匹配时状态寄存器中的标志位置 1（TIMx_SR 寄存器中的 CCxIF 位）。

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

比较输出模式也可以用来输出一个单脉冲（单脉冲输出模式）。

例如，通道 1 的比较输出模式的配置步骤如下：

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

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

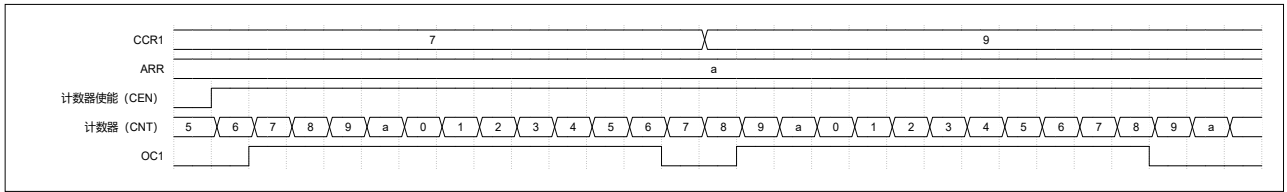


图 21-15 比较输出模式，OC1 信号在匹配时翻转

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。

21.4.3.3 PWM 输出

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

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



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

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

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

21.4.3.3.1 PWM 边沿对齐模式——递增计数模式

在递增计数模式配置的基础上，配置 TIMx_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。

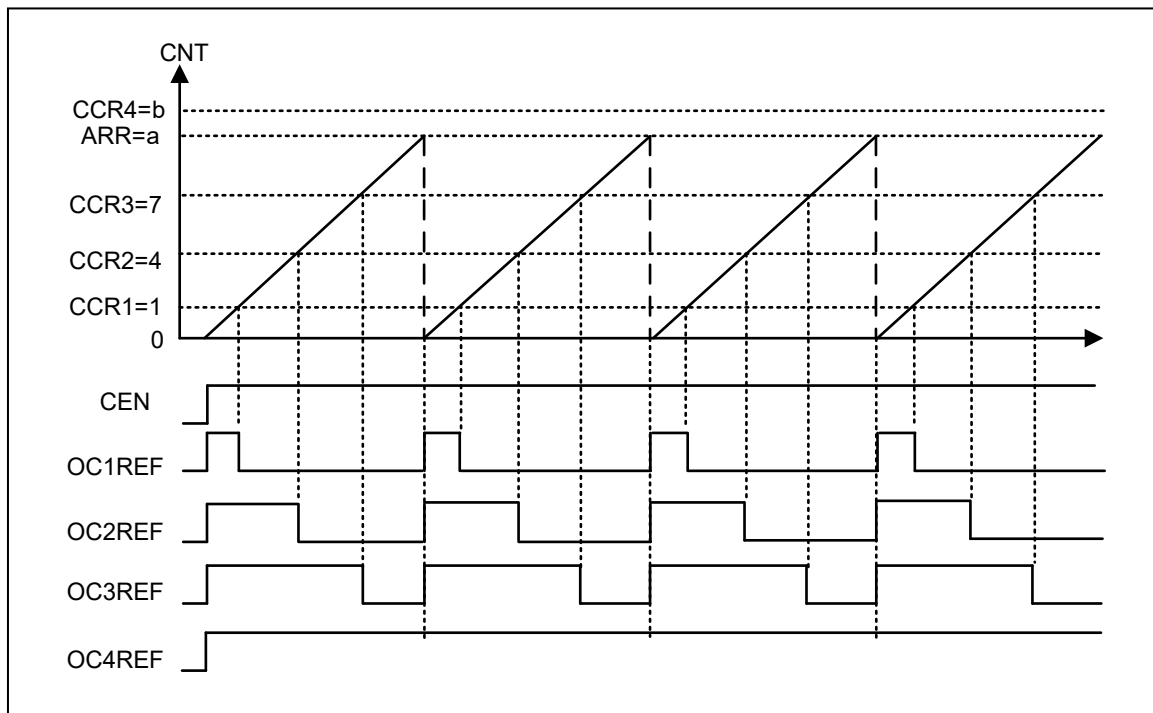


图 21-16 边沿对齐递增计数时 PWM 模式 1 的波形

21.4.3.3.2 PWM 边沿对齐模式——递减计数模式

在递减计数模式配置的基础上，配置 TIMx_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，

选择 PWM 模式 1，当 $TIMx\_CNT > TIMx\_CCRx$ 时通道 x ($OCxREF$) 为无效电平，否则有效电平。下图为 $CCR1=4$, $CCR2=6$, $CCR3=9$, $CCR4=b$, $ARR=a$ 时边沿对齐递减计数时 PWM 模式 1 的波形实例。

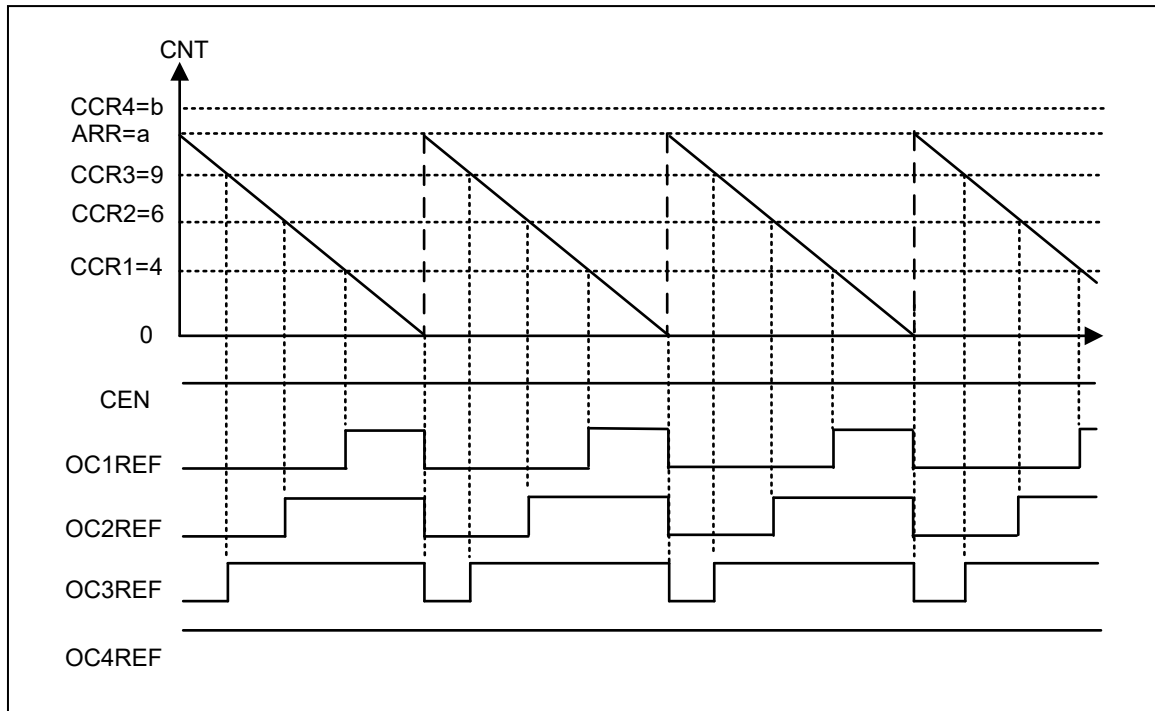


图 21-17 边沿对齐递减计数时 PWM 模式 1 的波形

21.4.3.3.3 PWM 中央对齐模式

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

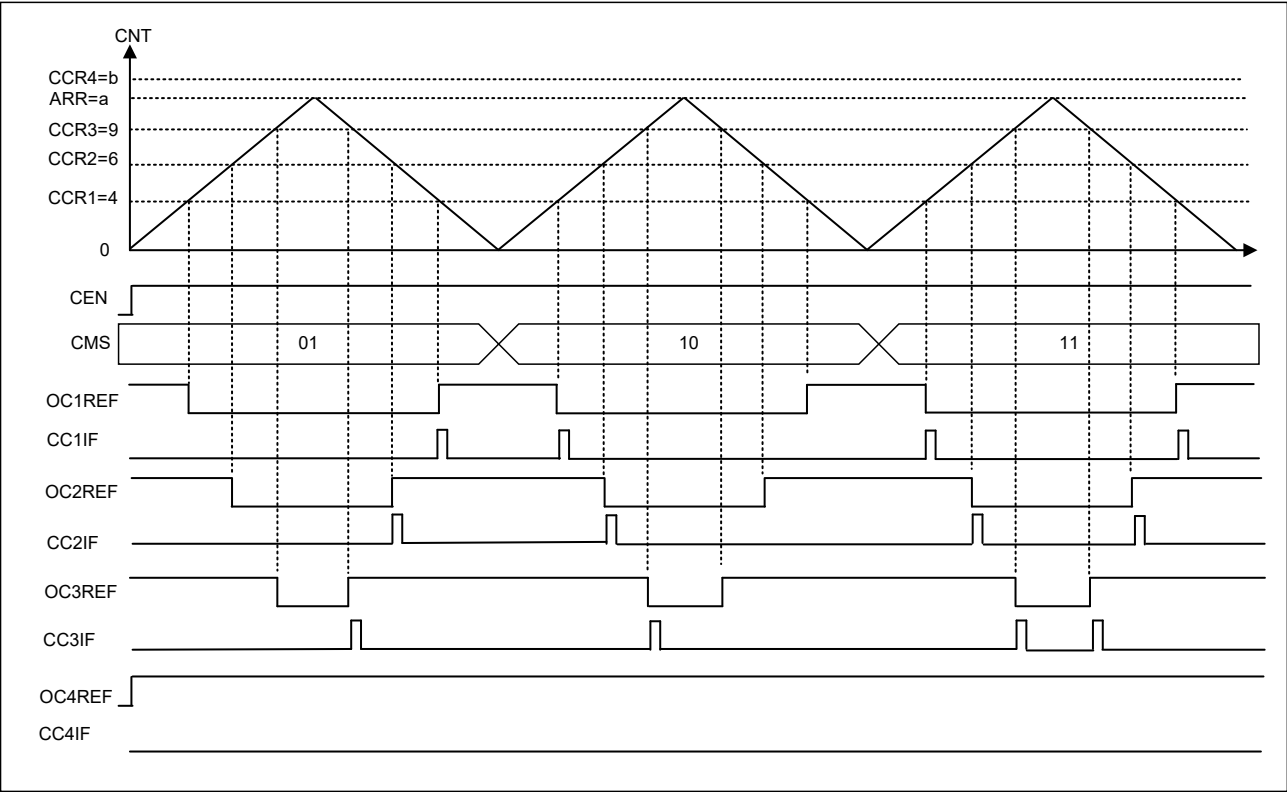


图 21-18 中央对齐 PWM 模式 1 的波形

注：

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

21.4.3.4 外部事件清除 OCxREF

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

例，OCxREF 信号连到一个外部输入时，ETR 配置如下：

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

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

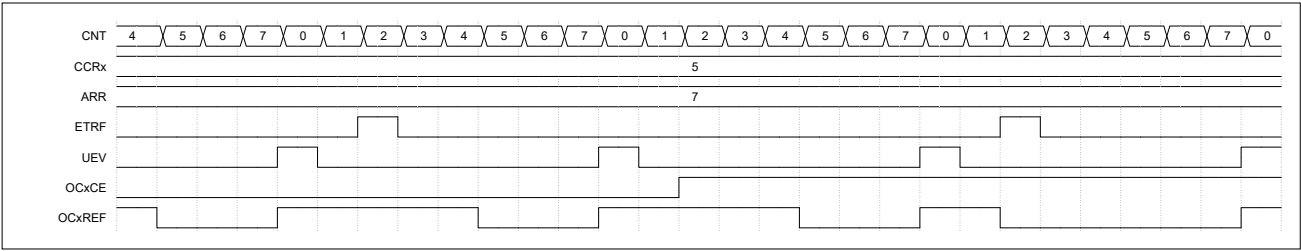


图 21-19 外部事件清除 OCxREF

21.4.3.5 单脉冲输出

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

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

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

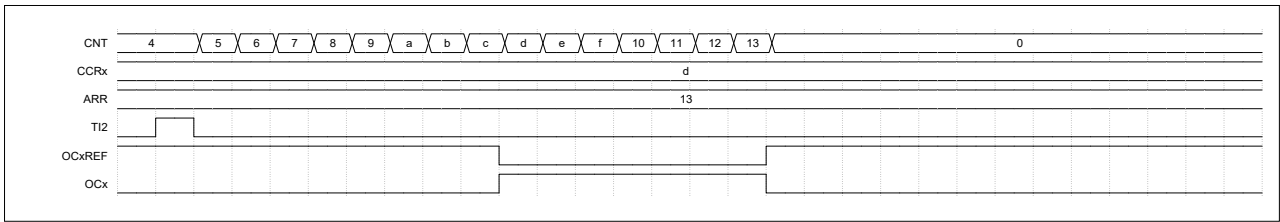


图 21-20 单脉冲模式

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

配置 TI2FP2 作为触发源：

1. 配置 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。

21.4.3.5.1 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。

21.4.4 从模式

21.4.4.1 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 21-2 计数方向与编码器信号的关系

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



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

下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。

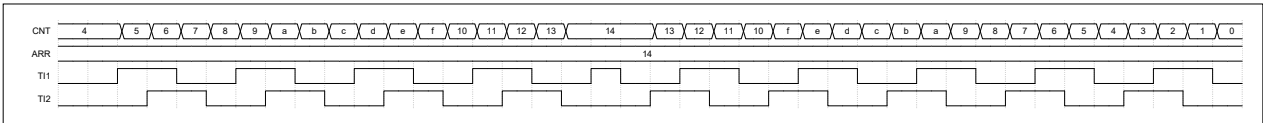


图 21-21 编码器模式下的计数器时序图

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

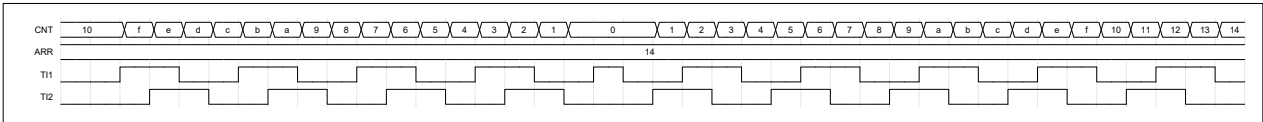


图 21-22 IC1FP1 反相编码器接口模式时序图

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。

21.4.4.2 复位模式

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

例如，TI2 输入端的下降沿触发计数器重启：

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

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

下图为复位模式下 TIMx_ARR = 0x13 的时序图。

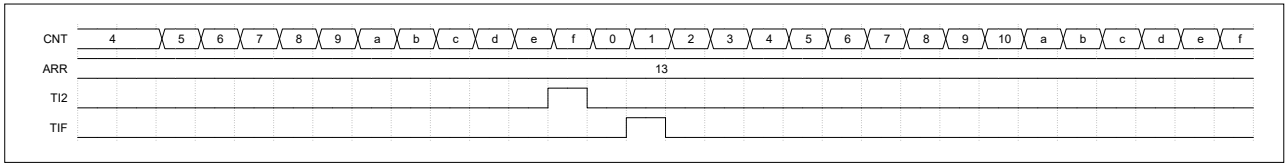


图 21-23 复位模式的控制时序图

21.4.4.3 门控模式

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

例如，计数器只在 TI1 为高时计数：

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

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

下图为门控模式下 TIMx_ARR=0xf 的时序图。



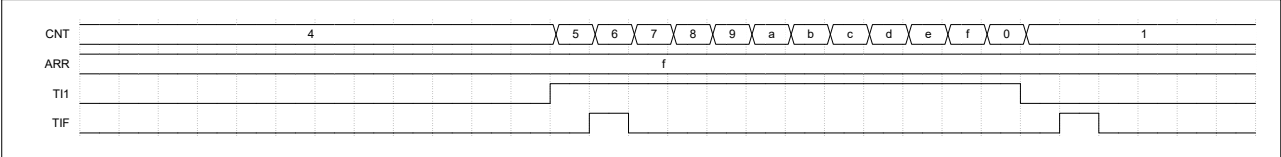


图 21-24 门控模式下的控制时序图

21.4.4.4 触发模式

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

例如，计数器在 TI1 输入的上升沿开始计数：

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

计数器的时钟源由内部时钟提供，当检测到 TI1 的上升沿，计数器开始计数。

下图为触发模式下 TIMx_ARR=0xf 的时序图。

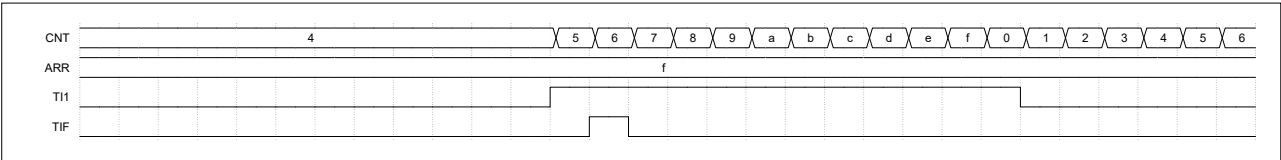


图 21-25 触发器模式下的控制时序图

21.4.4.5 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。

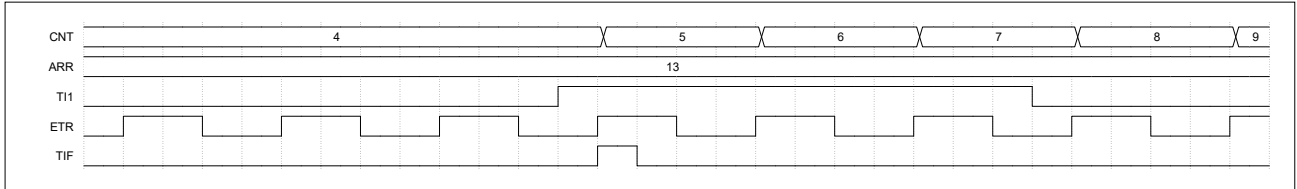


图 21-26 外部时钟模式 2+从模式（触发模式）控制时序图

21.4.5 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。

21.4.6 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。

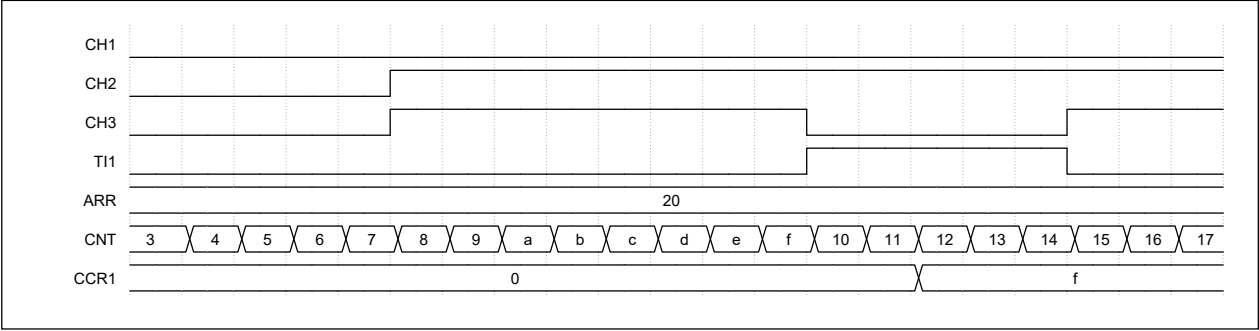


图 21-27 （TI1 异或输入）输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

21.4.7 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）

21.4.8 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 21-3 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

21.4.9 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。

21.5 寄存器描述

表 21-4 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

21.5.1 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		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（ETR, Tlx）所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT\_CK}$ 01: $t_{DTS} = 2 \times t_{INT\_CK}$ 10: $t_{DTS} = 4 \times t_{INT\_CK}$ 11: 保留，不要使用这个配置
7	ARPE	自动重载预装载使能（Auto-reload preload enable） 0: 关闭 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、CCR _x ）保持值不变。如果设置了 EGR_UG 位为 1，计数器和预分频器被初始化，如果从模式控制器接收到硬件复位，计数器将被初始化。
0	CEN	计数器使能（Counter enable） 0：禁止计数器 1：使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

21.5.2 TIMx_CR2 控制寄存器 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								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	主模式选择 (Master mode selection) 这些位控制 TRGO 信号的选择, 用于选择在主模式下送到从定时器的同步信息: 000: 复位 TIMx_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001: 使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO), 计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新 更新事件被选为 TRGO。 011: 捕获/比较脉冲 发生一次捕获或一次比较成功时, 触发输出送出一个 TRGO 信号。 100: 比较 OC1REF 信号被用于作为触发输出 (TRGO) 101: 比较 OC2REF 信号被用于作为触发输出 (TRGO) 110: 比较 OC3REF 信号被用于作为触发输出 (TRGO) 111: 比较 OC4REF 信号被用于作为触发输出 (TRGO)
3	CCDS	DMA 请求源选择 (Capture/compare DMA selection) 0: 当 CCx 发生捕获/比较事件时, 发送 CCx 的 DMA 请求 1: 发生更新事件时, 发送 CCx 的 DMA 请求 注: 仅适用于有内置 DMA 的产品
2: 0	Reserved	保留, 必须保持复位值。

21.5.3 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	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

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

Bit	Field	Description
7	MSM	主/从 模式 (Master/slave mode) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟, 以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步, 该功能可以把几个定时器同步到一个单一的外部事件。
6: 4	TS	触发选择 (Trigger selection) 触发输入源选择。 000: 内部触发 0 (ITR0) 001: 内部触发 1 (ITR1) 010: 内部触发 2 (ITR2) 011: 内部触发 3 (ITR3) 100: 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。
3	OCCS	比较输出信号 (OCxREF) 清除选择 (Output compare clear selection) 在 PWM 模式下, 清除比较输出 (OCxREF)。 0: 外部触发信号作为清除信号 1: 比较器 (COMP) 输出作为清除信号 注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。

Bit	Field	Description
2: 0	SMS	从模式选择 (Slave mode selection) 当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。 000: 关闭从模式 - 如果 CEN = 1, 则预分频器直接由内部时钟驱动。 001: 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。 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。 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。 100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新事件。 101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器开始计数并且产生一个更新事件。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。 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。 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并且产生一个更新事件。 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。

注: 内部触发连接关系参照芯片特定配置章节 TIM 之间的互联表。

21.5.4 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。

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

Bit	Field	Description
0	UIE	允许更新事件中断 (Update interrupt enable) 0: 禁止更新事件中断 1: 允许更新事件中断

21.5.5 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。
6	TIF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在 TRGI 输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断产生
5	Reserved	保留, 必须保持复位值。
4	CC4IF	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。

Bit	Field	Description
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1 (在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0 或读取 TIMx_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。

21.5.6 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。

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

21.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址：0x18

复位值：0x0000

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

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

比较输出模式：

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能 (Output compare 2 clear enable) 参考 OC1CE 的描述。

Bit	Field	Description
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 输入有效电平时, OC1REF 清零
6: 4	OC1M	通道 1 比较输出模式 (Output compare 1 mode) 该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。 000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为高电平。 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为低电平。 011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为低。强制 OC1REF 为低电平。 101: 强制为高。强制 OC1REF 为高电平。 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平。 注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。

Bit	Field	Description
3	OC1PE	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。 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。 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。
2	OC1FE	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。 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
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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	输入捕获 2 滤波器 (Input capture 2 filter) 参考 IC1F 的描述
11: 10	IC2PSC	输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 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)

Bit	Field	Description
7: 4	IC1F	通道 1 输入捕获滤波器 (Input capture 1 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC1PSC	通道 1 输入/捕获预分频器 (Input capture 1 prescaler) 该位定义了 IC1 的预分频系数。当 $CC1E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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)

21.5.8 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 输入有效电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	通道 3 比较输出模式 (Output compare 3 mode) 该位定义了输出参考信号 OC3REF 的动作, 而 OC3REF 决定了 OC3 的值。OC3REF 是高电平有效, 而 OC3 的有效电平取决于 CC3P 位。 000: 冻结。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为高电平 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为低电平 011: 匹配时翻转。当 TIMx_CCR3=TIMx_CNT 时, 翻转 OC3REF 的电平 100: 强制为低。强制 OC3REF 为低电平 101: 强制为高。强制 OC3REF 为高电平 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平。 注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能 (Output compare 3 preload enable) 0: 禁止 TIMx_CCR3 寄存器的预装载功能, 写入 TIMx_CCR3 寄存器的数值立即生效 1: 开启 TIMx_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR3 的预装载值在更新事件到来时生效 注: 仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM= 1), 无需设定预装载寄存器, 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。
2	OC3FE	通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。 0: 禁止通道 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能

Bit	Field	Description
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	输入捕获 4 滤波器 (Input capture 4 filter) 参考 IC3F 的描述
11: 10	IC4PSC	输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述
9: 8	CC4S	通道 4 捕获/比较选择 (Capture/Compare 4 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入, IC4 映射在 TI4 上 10: 通道 4 被配置为输入, IC4 映射在 TI3 上 11: 通道 4 被配置为输入, IC4 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC3F	通道 3 输入捕获滤波器 (Input capture 3 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC3PSC	通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 $CC3E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

21.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20

复位值: 0x0000

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

Bit	Field	Description
15	CC4NP	通道 4 输入捕获极性 (Capture 4 polarity) 参考 CC1NP 的描述。
14	Reserved	保留, 必须保持复位值。
13	CC4P	通道 4 输入/捕获输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	通道 4 输入/捕获输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	通道 3 输入捕获极性 (Capture 3 polarity) 参考 CC1NP 的描述。
10	Reserved	保留, 必须保持复位值。
9	CC3P	通道 3 输入/捕获输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	通道 3 输入/捕获输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	通道 2 输入捕获极性 (Capture 2 polarity) 参考 CC1NP 的描述。
6	Reserved	保留, 必须保持复位值。
5	CC2P	通道 2 输入捕获输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	通道 2 输入/捕获输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
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 21-5 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

21.5.10 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)							

21.5.11 TIMx_PSC 预分频器

偏移地址：0x28

复位值：0x0000

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

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

21.5.12 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。

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

21.5.14 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	通道 2 捕获/比较的值 (Capture/Compare 2 value) 参考 CCR1 的描述。

21.5.15 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。

21.5.16 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。

21.5.17 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 21.4.9 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

21.5.18 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 21.4.9 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。

21.5.19 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	

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

Bit	Field	Description
7:6	TI4_RMP	Timer5 TI4 复用 00: CH4 GPIO 输入 01: LSI 时钟输入 10: LSE 时钟输入 11: 保留 Timer2 TI4 复用 00 : GPIO 输入 其他保留
5:2	Reserved	保留，必须保持复位值。
1:0	ETR_RMP	Timer2 ETR 复用 00: ETR GPIO 输入 其他保留 Timer5 无 ETR 功能

22 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。

22.1 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）。

22.2 功能框图

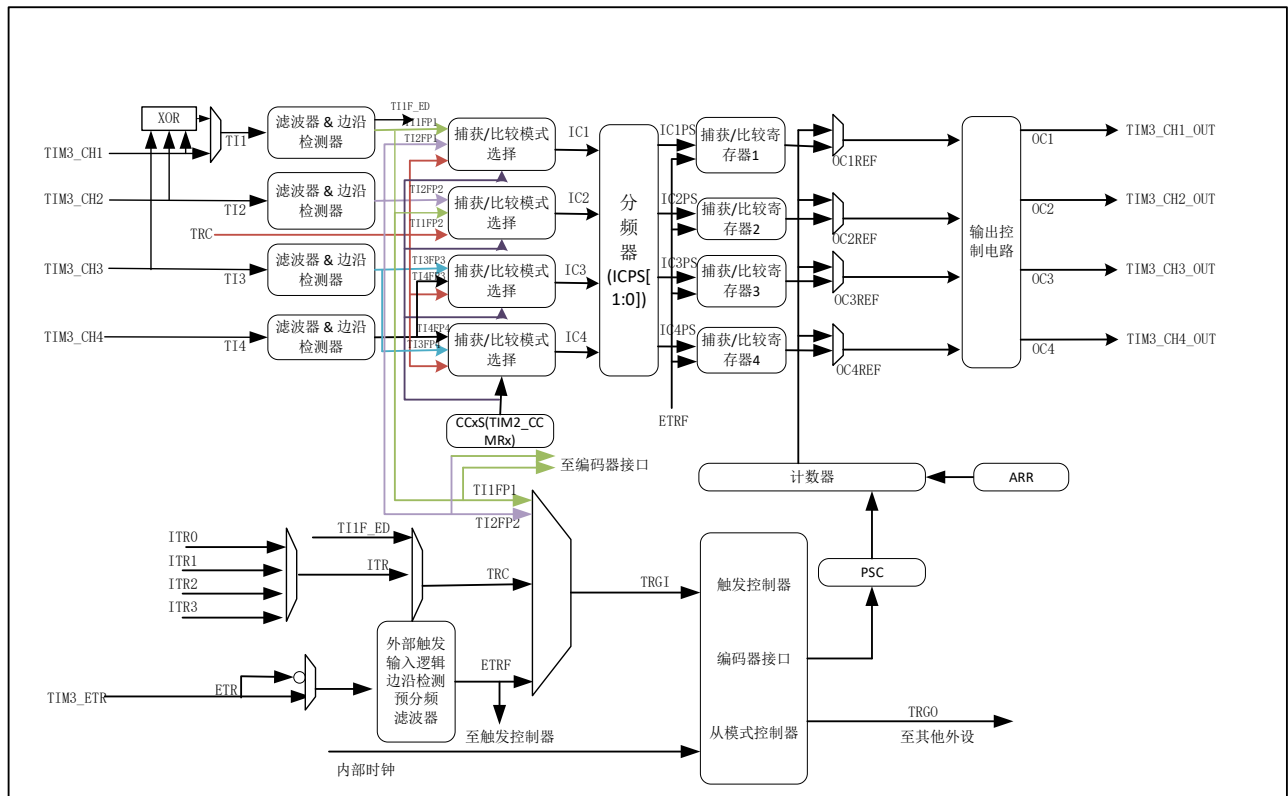


图 22-1 TIMx 结构图

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。

22.3 主要特征

- 16 位可实时编程预分频器，分频系数：1–65536 可调

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

22.4 功能描述

22.4.1 时钟

22.4.1.1 时钟选择

计数器的时钟源有以下几种：

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

上述几种时钟选择示意图如下：

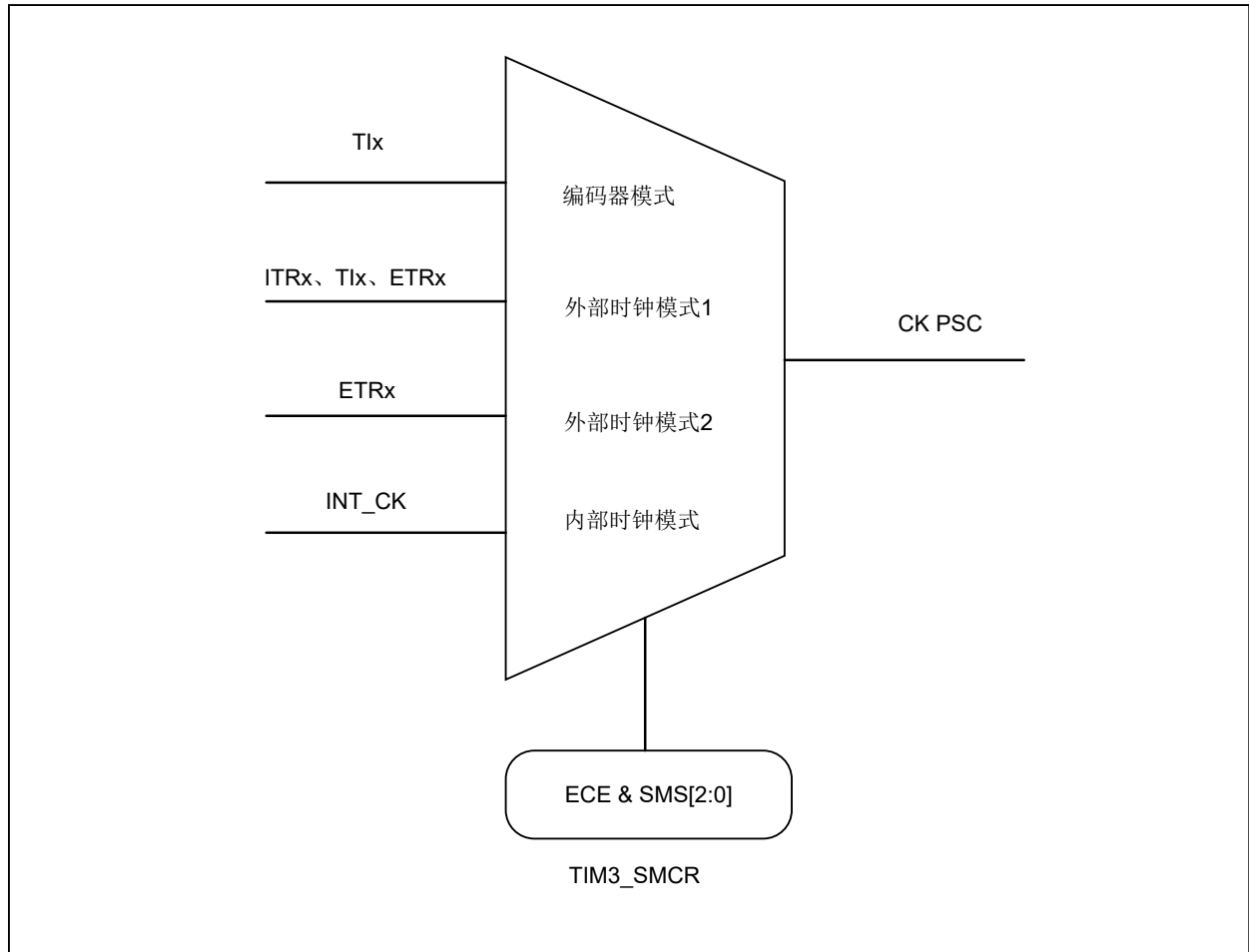


图 22-2 时钟选择

22.4.1.1.1 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。

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

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

例：计数器在 TI1 输入端的上升沿递增计数，具体配置如下：

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

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

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

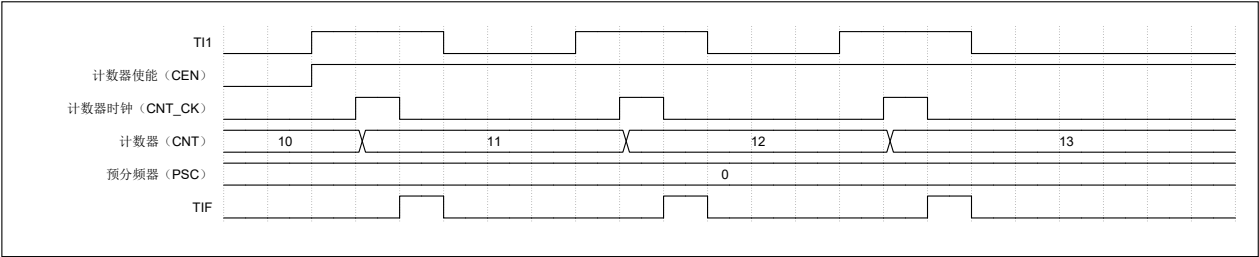


图 22-3 外部时钟模式 1 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）为复位值。

22.4.1.1.3 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。

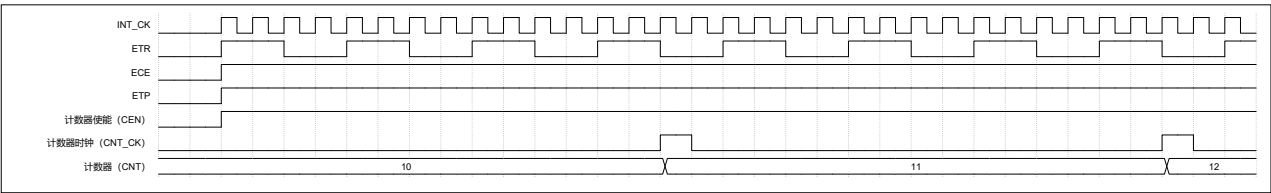


图 22-4 外部时钟模式 2 下的控制电路

注：使用外部时钟模式时，需要保持预分频器（TIMx_PSC）为复位值。

22.4.1.1.4 编码器模式

具体参考从模式-编码器接口章节。

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）。

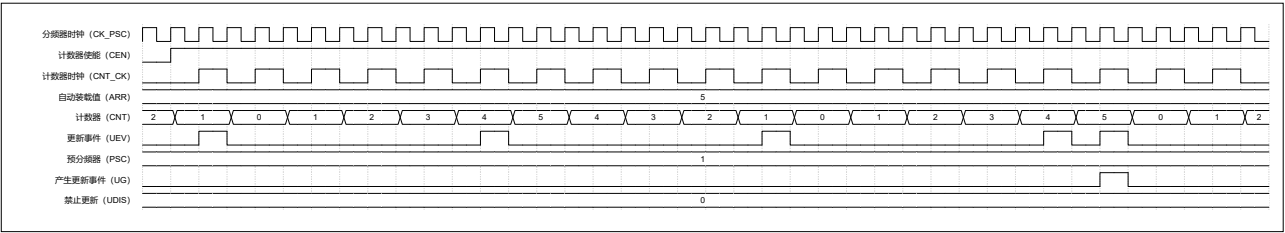


图 22-10 中央计数模式（UDIS=0）

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。

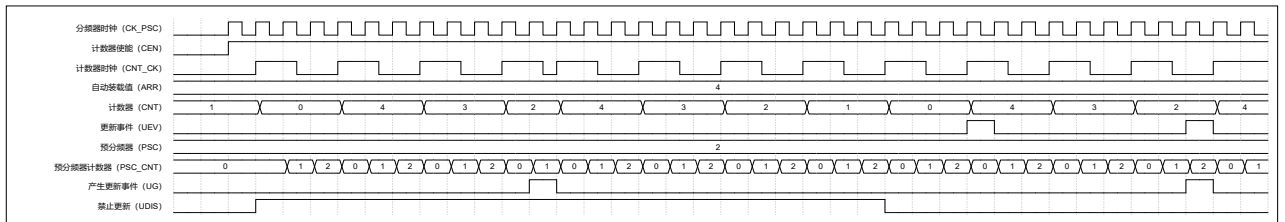


图 22-11 中央计数模式（UDIS=1 禁止产生更新事件）

22.4.2 输入捕获

22.4.2.1 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：

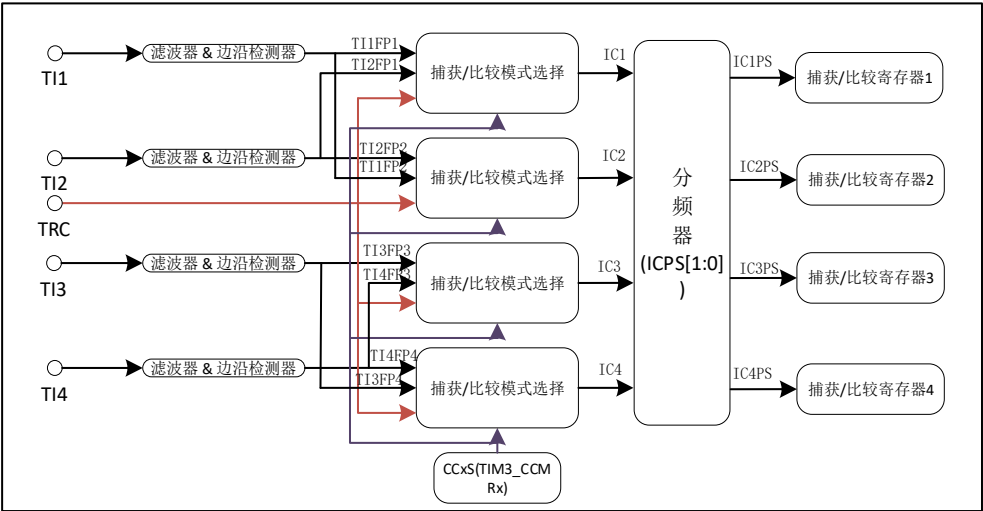


图 22-12 TIMx 输入捕获结构图

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 22-1 数字滤波器宽度与 **ICxF** 的对应关系表

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

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

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

1. 配置 **TIMx_CCMR1** 寄存器的 **CC1S=01**，**CC1** 通道被配置为输入，**IC1** 映射在 **TI1** 上。
2. 配置 **TIMx_CCMR1** 寄存器的 **IC1F[3: 0]**，配置数字滤波器的滤波宽度（按需配置）。
3. 配置 **TIMx_CCER** 寄存器的 **CC1P=0**，选择捕获发生在 **TI1** 信号的上升沿。
4. 配置 **TIMx_CCMR1** 寄存器的 **IC1PSC[1:0]**，选择预分频系数。
5. 配置 **TIMx_CCER** 寄存器的 **CC1E = 1**，开启输入/捕获通道 1 的捕获使能。
6. 配置 **TIMx_DIER** 寄存器的 **CC1IE=1**，使能通道 1 的捕获/比较通道 1 中断请求；如果芯片有内置 **DMA**，配置 **TIMx_DIER** 寄存器的 **CC1DE=1**，允许捕获/比较通道 1 的 **DMA** 请求。

注：

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

22.4.2.2 PWM 捕获

PWM 输入模式的操作配置与一般输入捕获有以下不同点：

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

例：测量 TI1 的 PWM 信号的宽度（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。

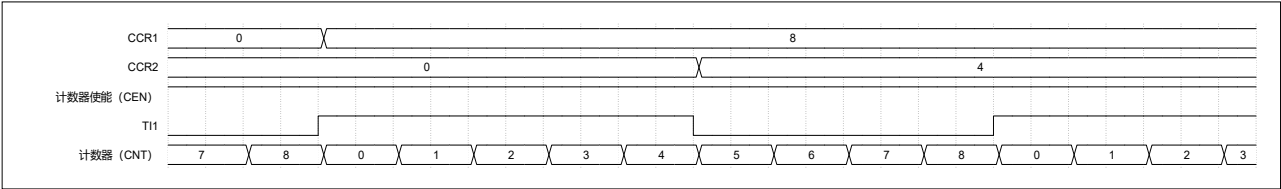


图 22-13 PWM 输入模式时序

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。

22.4.3 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：

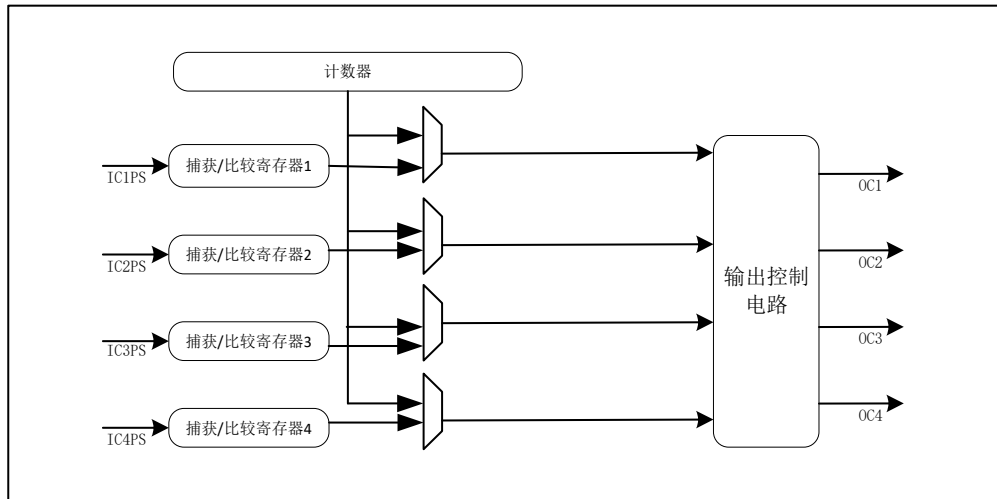


图 22-14 比较输出部分结构图

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。

22.4.3.1 强制输出

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

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

22.4.3.2 比较输出

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

例如，当计数器与捕获/比较寄存器的内容匹配时，比较输出模式下的操作如下：

1. 在比较匹配时，`OCxM` 的值不同，输出通道 `x` 信号 `OCx` 的操作不同：
 - ◆ `OCxM = 000`：`OCx` 信号保持它的电平
 - ◆ `OCxM = 001`：`OCx` 信号被设置成有效电平
 - ◆ `OCxM = 010`：`OCx` 信号被设置成无效电平
 - ◆ `OCxM = 011`：`OCx` 信号进行翻转
2. 匹配时状态寄存器中的标志位置 1（`TIMx_SR` 寄存器中的 `CCxIF` 位）。

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

比较输出模式也可以用来输出一个单脉冲（单脉冲输出模式）。

例如，通道 1 的比较输出模式的配置步骤如下：

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

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

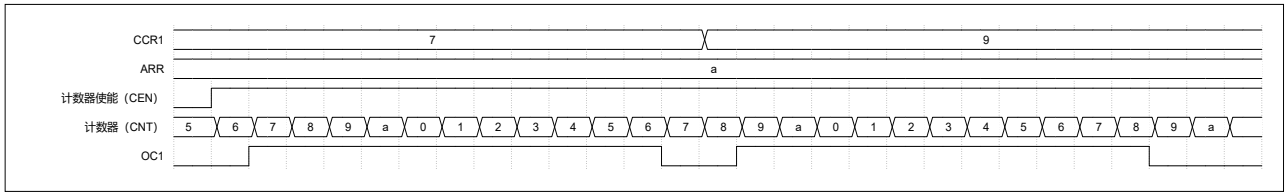


图 22-15 比较输出模式，OC1 信号在匹配时翻转

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。

22.4.3.3 PWM 输出

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

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



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

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

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

22.4.3.3.1 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。

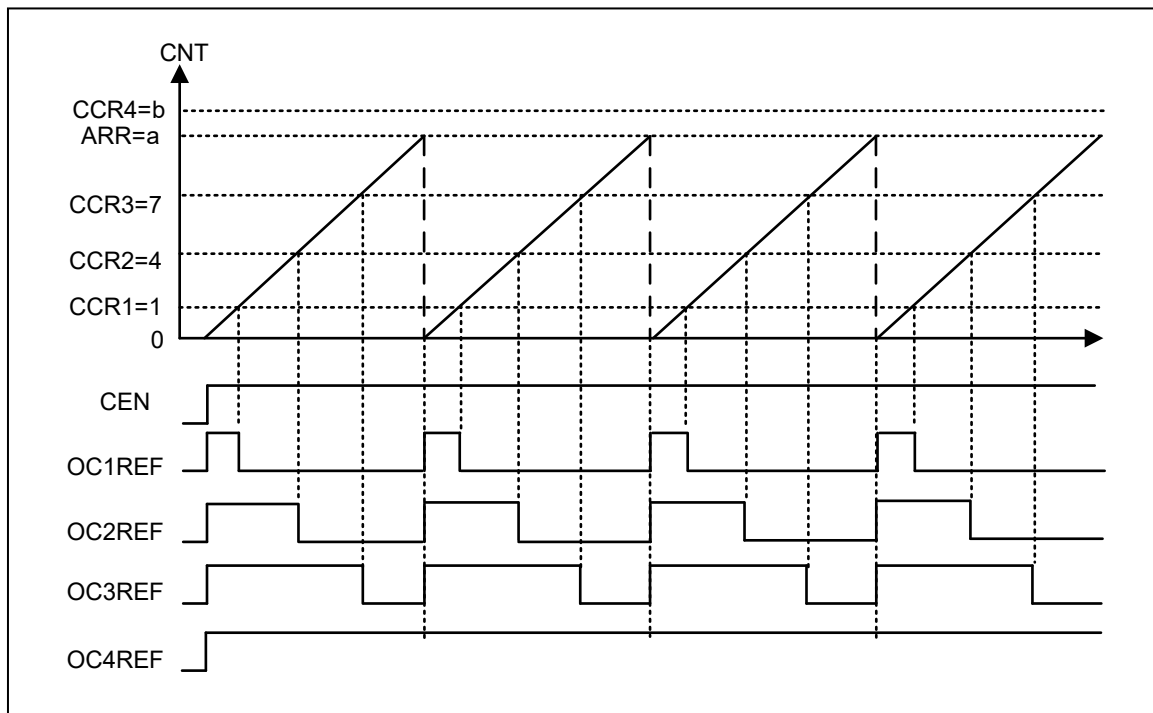


图 22-16 边沿对齐递增计数时 PWM 模式 1 的波形

22.4.3.3.2 PWM 边沿对齐模式——递减计数模式

在递减计数模式配置的基础上，配置 TIMx_CCMRx 寄存器的 CCxS=00，选择输出模式，OCxM=110，

选择 PWM 模式 1，当 $TIMx\_CNT > TIMx\_CCRx$ 时通道 x ($OCxREF$) 为无效电平，否则有效电平。下图为 $CCR1=4$, $CCR2=6$, $CCR3=9$, $CCR4=b$, $ARR=a$ 时边沿对齐递减计数时 PWM 模式 1 的波形实例。

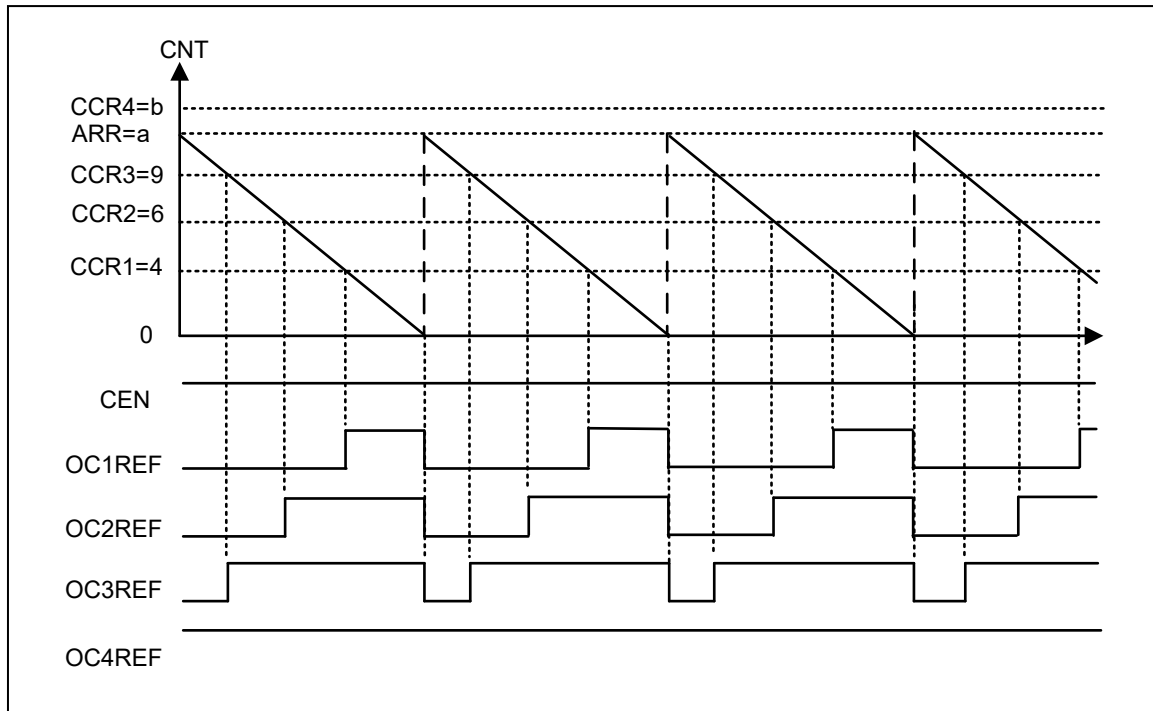


图 22-17 边沿对齐递减计数时 PWM 模式 1 的波形

22.4.3.3.3 PWM 中央对齐模式

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

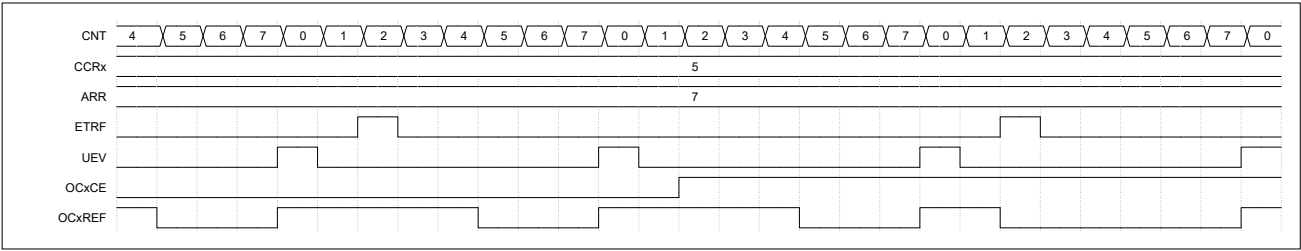


图 22-19 外部事件清除 OCxREF

22.4.3.5 单脉冲输出

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

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

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

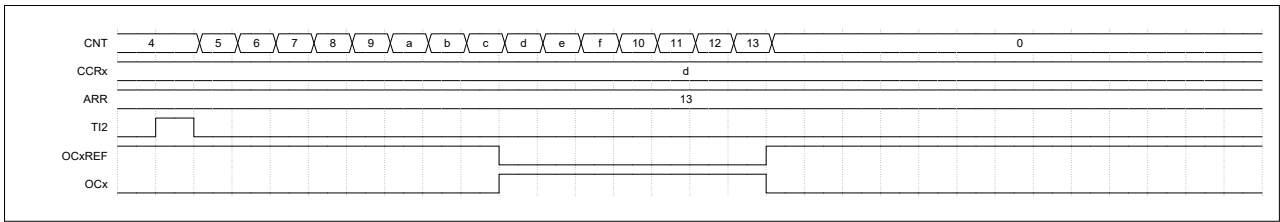


图 22-20 单脉冲模式

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

配置 TI2FP2 作为触发源：

1. 配置 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。

22.4.3.5.1 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。

22.4.4 从模式

22.4.4.1 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 22-2 计数方向与编码器信号的关系

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

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

下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。

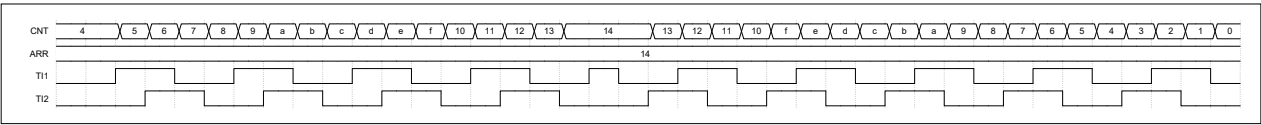


图 22-21 编码器模式下的计数器时序图

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

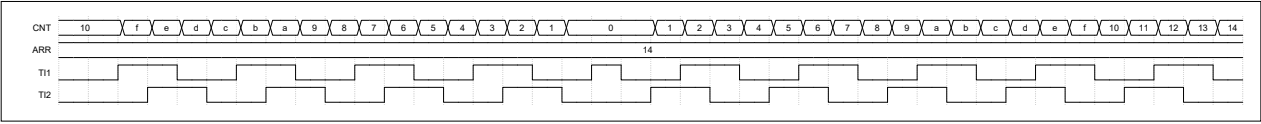


图 22-22 IC1FP1 反相编码器接口模式时序图

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。

22.4.4.2 复位模式

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

例如，TI2 输入端的下降沿触发计数器重启：

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

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

下图为复位模式下 TIMx_ARR = 0x13 的时序图。

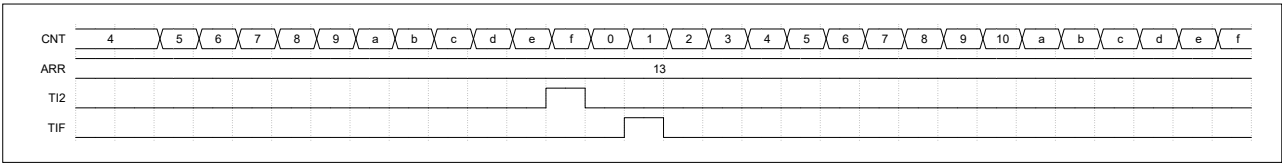


图 22-23 复位模式的控制时序图

22.4.4.3 门控模式

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

例如，计数器只在 TI1 为高时计数：

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

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

下图为门控模式下 TIMx_ARR=0xf 的时序图。



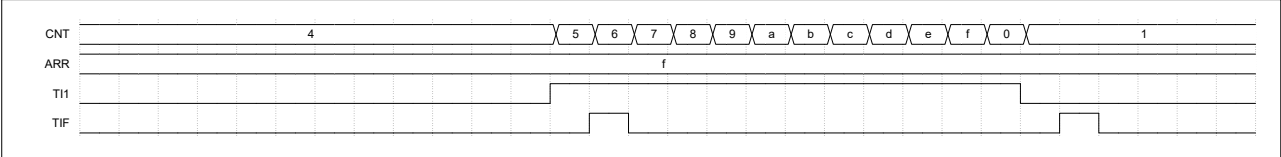


图 22-24 门控模式下的控制时序图

22.4.4.4 触发模式

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

例如，计数器在 TI1 输入的上升沿开始计数：

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

计数器的时钟源由内部时钟提供，当检测到 TI1 的上升沿，计数器开始计数。

下图为触发模式下 TIMx_ARR=0xf 的时序图。

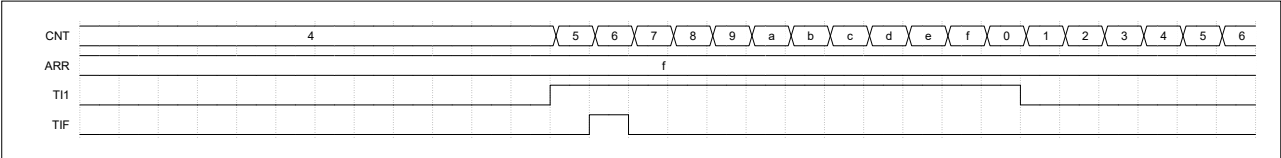


图 22-25 触发器模式下的控制时序图

22.4.4.5 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。

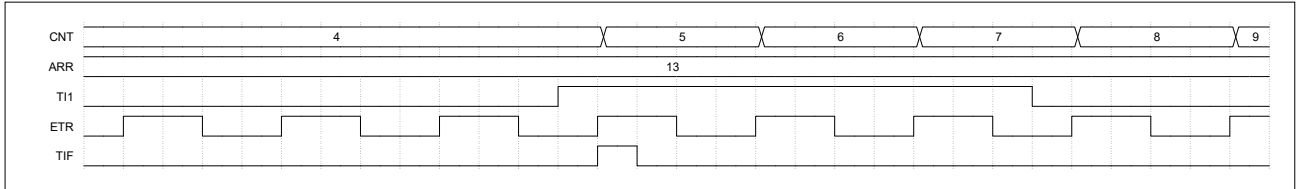


图 22-26 外部时钟模式 2+从模式（触发模式）控制时序图

22.4.5 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。

22.4.6 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。

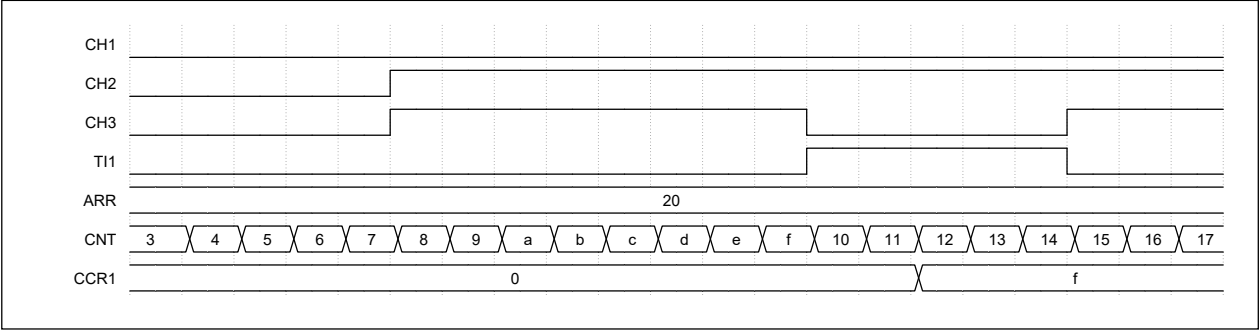


图 22-27 （TI1 异或输入）输入捕获波形图

霍尔接口电路

详细描述请参考 TIM1 相关章节。

22.4.7 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）

22.4.8 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 22-3 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

22.4.9 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。

22.5 寄存器描述

表 22-4 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

22.5.1 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		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15: 10	Reserved	保留，必须保持复位值。
9: 8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（ETR, TIX）所用的时钟之间的分频比例。 00: $t_{DTS} = t_{INT\_CK}$ 01: $t_{DTS} = 2 \times t_{INT\_CK}$ 10: $t_{DTS} = 4 \times t_{INT\_CK}$ 11: 保留，不要使用这个配置
7	ARPE	自动重载预装载使能（Auto-reload preload enable） 0: 关闭 TIMx_ARR 寄存器的影子寄存器 1: 使能 TIMx_ARR 寄存器的影子寄存器
6: 5	CMS	中央对齐模式选择（Center-aligned mode selection） 00: 边沿对齐模式。计数方向取决于 DIR 位 01: 中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式，只在计数器递减计数时比较中断标志位被置 1 10: 中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式，只在计数器递增计数时比较中断标志位被置 1 11: 中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式，在计数器递增和递减计数时比较中断标志位均被置 1 注：计数过程中，不允许更改此位。
4	DIR	计数方向（Direction） 0: 计数器递增计数 1: 计数器递减计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。

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、CCR _x ）保持值不变。如果设置了 EGR_UG 位为 1，计数器和预分频器被初始化，如果从模式控制器接收到硬件复位，计数器将被初始化。
0	CEN	计数器使能（Counter enable） 0：禁止计数器 1：使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

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

22.5.3 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	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

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

Bit	Field	Description
7	MSM	主/从 模式 (Master/slave mode) 0: 无作用 1: 触发输入 (TRGI) 事件被延迟, 以实现当前定时器 (通过 TRGO) 与它的从定时器间的完美同步, 该功能可以把几个定时器同步到一个单一的外部事件。
6: 4	TS	触发选择 (Trigger selection) 触发输入源选择。 000: 内部触发 0 (ITR0) 001: 内部触发 1 (ITR1) 010: 内部触发 2 (ITR2) 011: 内部触发 3 (ITR3) 100: 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。
3	OCCS	比较输出信号 (OCxREF) 清除选择 (Output compare clear selection) 在 PWM 模式下, 清除比较输出 (OCxREF)。 0: 外部触发信号作为清除信号 1: 比较器 (COMP) 输出作为清除信号 注: 仅适用于支持外部触发或有内置比较器 (COMP) 的产品。

Bit	Field	Description
2: 0	SMS	从模式选择 (Slave mode selection) 当选择了外部信号作为触发源, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关。 000: 关闭从模式 - 如果 CEN = 1, 则预分频器直接由内部时钟驱动。 001: 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。 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。 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。 100: 复位模式 - 选中的触发输入 (TRGI) 的上升沿重新初始化计数器, 并且产生一个更新事件。 101: 门控模式 - 当触发输入 (TRGI) 为高时, 计数器开始计数并且产生一个更新事件。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。 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。 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并且产生一个更新事件。 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。

注: 内部触发连接关系参照芯片特定配置章节 TIM 之间的互联表。

22.5.4 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。

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

Bit	Field	Description
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

22.5.5 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。
6	TIF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在 TRGI 输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置 1。它由软件清 0。 0: 无触发器事件产生 1: 触发器中断产生
5	Reserved	保留, 必须保持复位值。
4	CC4IF	捕获/比较 4 中断标记 (Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	捕获/比较 3 中断标记 (Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	捕获/比较 2 中断标记 (Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。

Bit	Field	Description
1	CC1IF	捕获/比较 1 中断标记 (Capture/Compare 1 interrupt flag) 通道 1 为输出模式: 当计数器值与比较值匹配时该位由硬件置 1 (在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位)。它由软件清 0。 0: 无匹配发生 1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式: 当发生捕获事件时该位由硬件置 1, 由软件清 0 或读取 TIMx_CCR1 的值清 0。 0: 无输入捕获产生 1: 计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新中断发生 1: 发生更新中断 当寄存器被更新时该位由硬件置 1: - 若 TIMx_CR1 寄存器的 UDIS=0, 且 REP_CNT=0, 当计数器产生上溢/下溢事件时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 从模式控制器产生更新事件时。

22.5.6 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。

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

22.5.7 TIMx_CCMR1 捕获/比较模式寄存器 1

偏移地址: 0x18

复位值: 0x0000

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

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

比较输出模式:

Bit	Field	Description
15	OC2CE	通道 2 比较输出清零使能 (Output compare 2 clear enable) 参考 OC1CE 的描述。

Bit	Field	Description
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 输入有效电平时, OC1REF 清零
6: 4	OC1M	通道 1 比较输出模式 (Output compare 1 mode) 该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。 000: 冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为高电平。 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时, 强制 OC1REF 为低电平。 011: 匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。 100: 强制为低。强制 OC1REF 为低电平。 101: 强制为高。强制 OC1REF 为高电平。 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平, 否则为低电平。 注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。

Bit	Field	Description
3	OC1PE	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。 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。 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。
2	OC1FE	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。 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
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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	输入捕获 2 滤波器 (Input capture 2 filter) 参考 IC1F 的描述
11: 10	IC2PSC	输入/捕获 2 预分频器 (Input capture 2 prescaler) 参考 IC1PSC 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)

Bit	Field	Description
7: 4	IC1F	通道 1 输入捕获滤波器 (Input capture 1 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC1PSC	通道 1 输入/捕获预分频器 (Input capture 1 prescaler) 该位定义了 IC1 的预分频系数。当 $CC1E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC1S	通道 1 捕获/比较选择 (Capture/Compare 1 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 1 被配置为输出 01: 通道 1 被配置为输入, 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)

22.5.8 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 输入有效电平时，OC3REF 清零

Bit	Field	Description
6: 4	OC3M	通道 3 比较输出模式 (Output compare 3 mode) 该位定义了输出参考信号 OC3REF 的动作, 而 OC3REF 决定了 OC3 的值。OC3REF 是高电平有效, 而 OC3 的有效电平取决于 CC3P 位。 000: 冻结。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用 001: 匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为高电平 010: 匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时, 强制 OC3REF 为低电平 011: 匹配时翻转。当 TIMx_CCR3=TIMx_CNT 时, 翻转 OC3REF 的电平 100: 强制为低。强制 OC3REF 为低电平 101: 强制为高。强制 OC3REF 为高电平 110: PWM 模式 1。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平; 在递减计数时, 当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平。 111: PWM 模式 2。在递增计数时, 当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平, 否则为高电平; 在递减计数时, 当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平, 否则为低电平。 注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时, OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能 (Output compare 3 preload enable) 0: 禁止 TIMx_CCR3 寄存器的预装载功能, 写入 TIMx_CCR3 寄存器的数值立即生效 1: 开启 TIMx_CCR3 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR3 的预装载值在更新事件到来时生效 注: 仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM= 1), 无需设定预装载寄存器, 其它情况下, 需要设定预装载寄存器, 否则后续动作不确定。
2	OC3FE	通道 3 比较输出快速使能 (Output compare 3 fast enable) 该位为 1 时, 若通道配置为 PWM 模式, 会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配, 此时 OC 被设置为比较电平, 与比较结果无关。 0: 禁止通道 3 比较输出快速使能 1: 开启通道 3 比较输出快速使能

Bit	Field	Description
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

输入捕获模式:

Bit	Field	Description
15: 12	IC4F	输入捕获 4 滤波器 (Input capture 4 filter) 参考 IC3F 的描述
11: 10	IC4PSC	输入/捕获 4 预分频器 (Input capture 4 prescaler) 参考 IC3PSC 的描述
9: 8	CC4S	通道 4 捕获/比较选择 (Capture/Compare 4 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 4 被配置为输出 01: 通道 4 被配置为输入, IC4 映射在 TI4 上 10: 通道 4 被配置为输入, IC4 映射在 TI3 上 11: 通道 4 被配置为输入, IC4 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

Bit	Field	Description
7: 4	IC3F	通道 3 输入捕获滤波器 (Input capture 3 filter) 数字滤波器由一个事件计数器组成, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=2$ 0010: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=4$ 0011: 采样频率 $f_{sampling}=f_{INT\_CK}$, $N=8$ 0100: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{sampling}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{sampling}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{sampling}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{sampling}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{sampling}=f_{DTS}/32$, $N=8$
3: 2	IC3PSC	通道 3 输入/捕获预分频器 (Input capture 3 prescaler) 该位定义了 IC3 的预分频系数。当 $CC3E=0$ (TIMx_CCER 寄存器中) 时, 预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1: 0	CC3S	通道 3 捕获/比较选择 (Capture/Compare 3 selection) 该位定义通道的方向和输入信号的选择, 只有在通道关闭时这些位才可写入: 00: 通道 3 被配置为输出 01: 通道 3 被配置为输入, IC3 映射在 TI3 上 10: 通道 3 被配置为输入, IC3 映射在 TI4 上 11: 通道 3 被配置为输入, IC3 映射在 TRC 上, 此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)

22.5.9 TIMx_CCER 捕获/比较使能寄存器

偏移地址: 0x20

复位值: 0x0000

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

Bit	Field	Description
15	CC4NP	通道 4 输入捕获极性 (Capture 4 polarity) 参考 CC1NP 的描述。
14	Reserved	保留, 必须保持复位值。
13	CC4P	通道 4 输入/捕获输出极性 (Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	通道 4 输入/捕获输出使能 (Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	通道 3 输入捕获极性 (Capture 3 polarity) 参考 CC1NP 的描述。
10	Reserved	保留, 必须保持复位值。
9	CC3P	通道 3 输入/捕获输出极性 (Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	通道 3 输入/捕获输出使能 (Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	通道 2 输入捕获极性 (Capture 2 polarity) 参考 CC1NP 的描述。
6	Reserved	保留, 必须保持复位值。
5	CC2P	通道 2 输入捕获输出极性 (Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	通道 2 输入/捕获输出使能 (Capture/Compare 2 output enable) 参考 CC1E 的描述。
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 22-5 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

22.5.10 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)

22.5.11 TIMx_PSC 预分频器

偏移地址：0x28

复位值：0x0000

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

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

22.5.12 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。

22.5.13 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。

22.5.14 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。								

22.5.15 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。								

22.5.16 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。								

22.5.17 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 22.4.9 DMA 章节。

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

Bit	Field	Description
15: 13	Reserved	保留，必须保持复位值。
12: 8	DBL	DMA 连续传送长度（DMA burst length） 这些位定义了 DMA 在连续模式下的访问寄存器的数量 00000: 1 次传输 00001: 2 次传输 00010: 3 次传输 10001: 18 次传输
7: 5	Reserved	保留，必须保持复位值。
4: 0	DBA	DMA 基地址（DMA base address） 这些位定义了 DMA 在连续模式下访问 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

22.5.18 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 22.4.9 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: $TIMx\_CR1 \text{ 地址} + DBA + \text{DMA 索引}$, 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。

23 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。

23.1 简介

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

23.2 功能框图

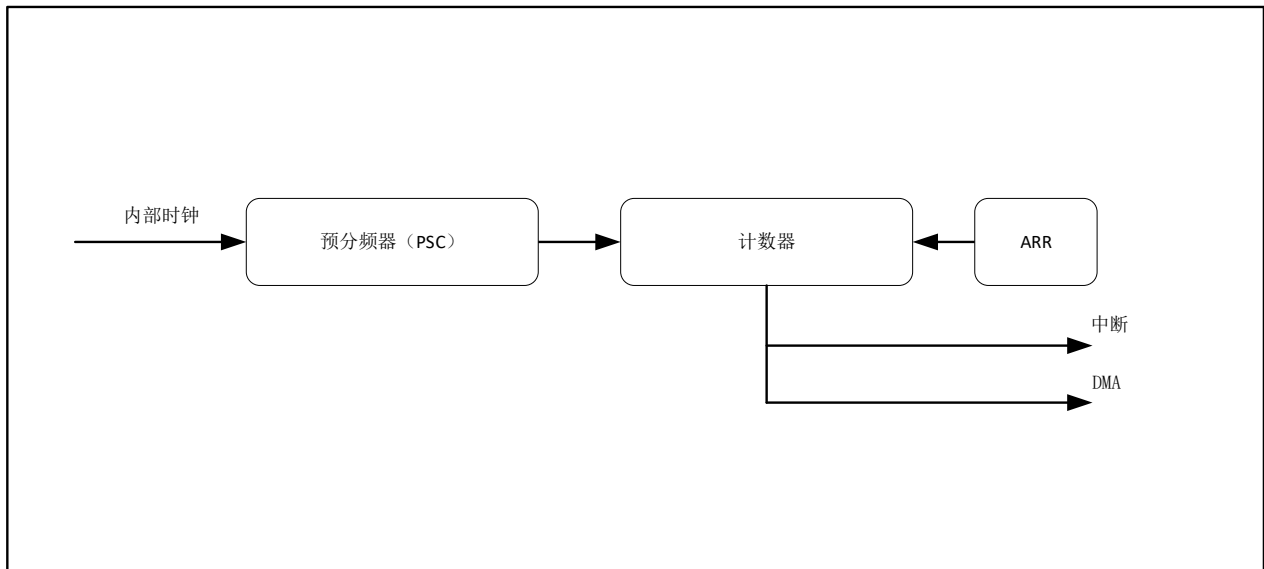


图 23-1 TIMx 结构图

上图为 TIMx 的结构框图。

23.3 主要特征

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

23.4 功能描述

23.4.1 时钟

23.4.1.1 时钟选择

计数器的时钟由内部时钟（INT_CK）提供。

23.4.1.2 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，在下次更新事件时生效。

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

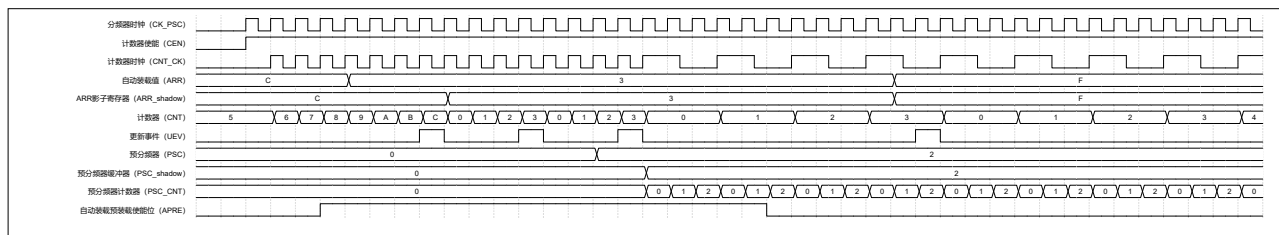


图 23-2 自动预装载

23.4.1.3 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。

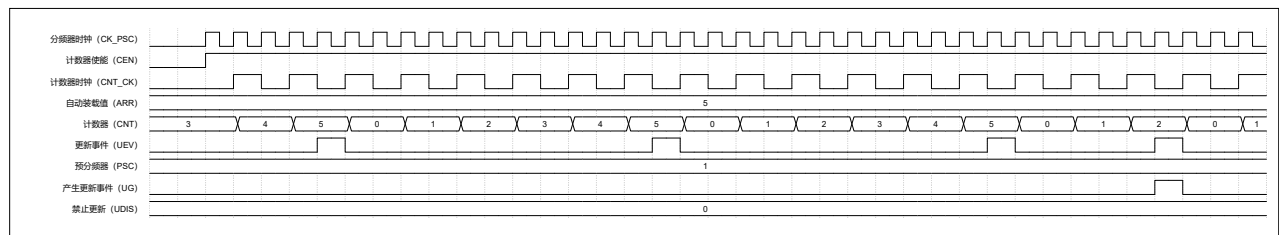


图 23-3 递增计数模式（UDIS=0）

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。

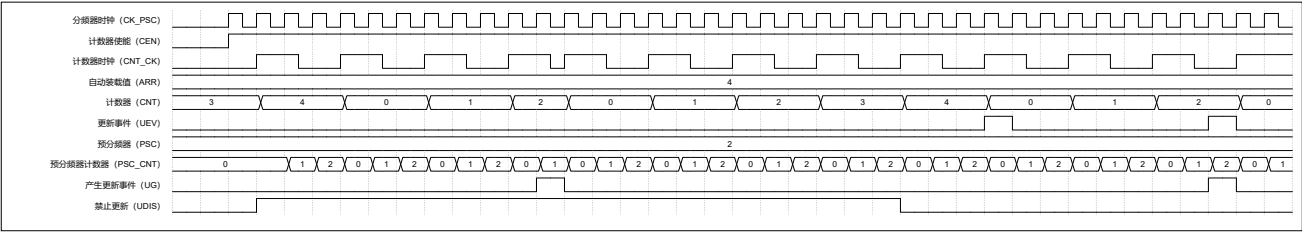


图 23-4 递增计数模式（UDIS=1 禁止产生更新事件）

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。

23.4.2 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）

23.4.3 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 23-1 中断事件一览表

中断事件	标志位	使能位
更新中断	UIF	UIE

23.4.4 DMA

TIMx 能够在发生更新事件时生成 DMA 请求。

23.5 寄存器描述

表 23-2 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

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

23.5.2 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

23.5.3 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。

23.5.4 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

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。

23.5.5 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）

23.5.6 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。

23.5.7 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。

24 LPTIMER 低功耗定时器

24.1 简介

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

24.2 功能框图

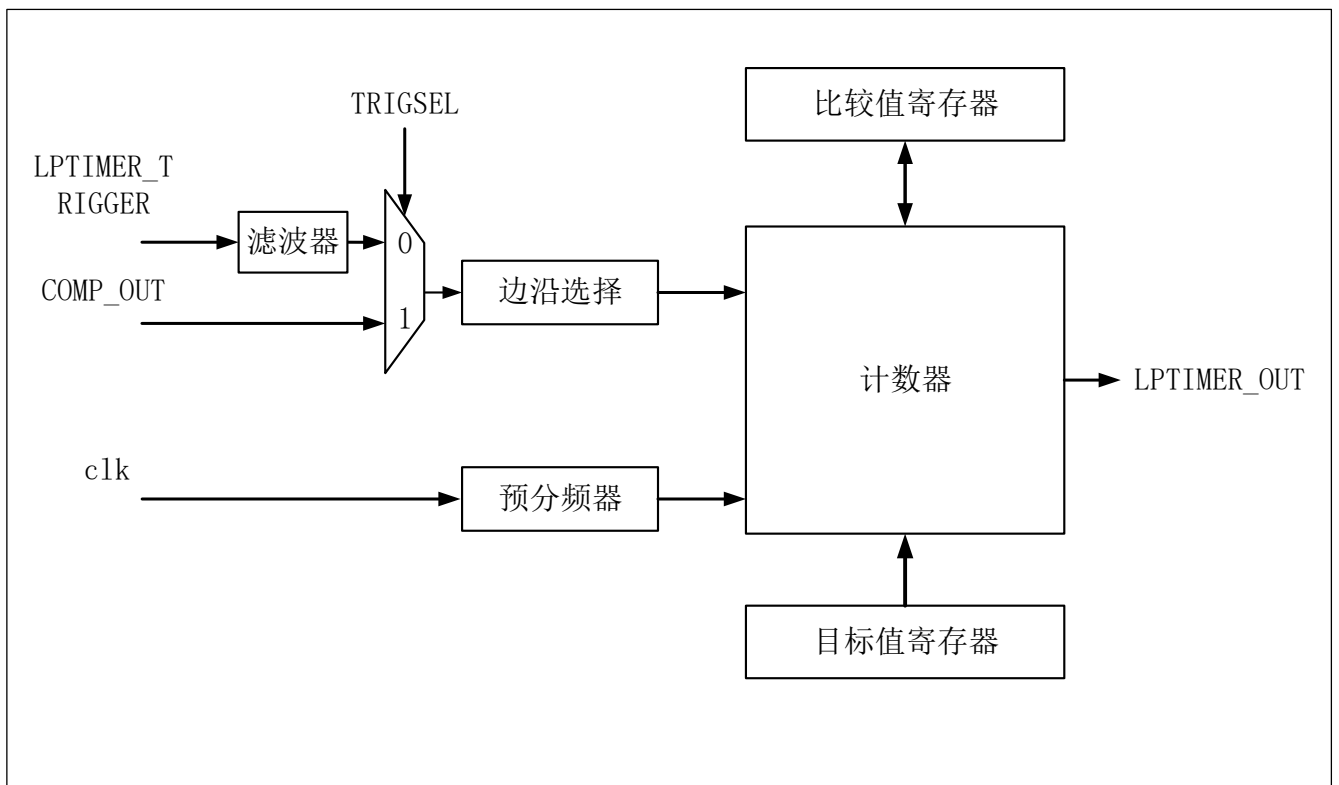


图 24-1 功能框图

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

24.3 主要特征

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

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

24.4 中断

LPTIM 的中断包括：外部触发中断，比较匹配中断，计数器溢出中断，当相应的中断使能位打开，发生相应的事件时，产生相应的中断。

24.5 功能描述

24.5.1 计数模式

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

24.5.2 普通定时器

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

24.5.3 Trigger 脉冲触发计数

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

24.5.4 TimeOut 模式

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

注：

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

24.5.5 PWM 输出

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

24.6 寄存器

24.6.1 寄存器概览

表 24-1 LPTIM 寄存器概览

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

24.6.2 LPT_CFG 配置寄存器

偏移地址：0x0

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLTEN	Res.				DIVSEL			TRIGCFG		TRIGSEL	POLARITY	PWM	TMODE		MODE
rw					rw			rw		rw	rw	rw	rw		rw

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

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

注: FLTEN, DIVSEL, TRIGCFG 应在 LPTEN 为 0 时配置。

24.6.3 LPT_IE 中断使能寄存器

偏移地址: 0x04

复位值: 0x0000

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

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

24.6.4 LPT_IF 中断标志寄存器

偏移地址: 0x08

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													CMPIF	TRIGIF	OVIF
													w1c	w1c	w1c

Bit	Field	Description
15:3	Reserved	保留, 始终读为 0
2	CMPIF	比较匹配中断标志位, 写 1 清零 0: 无中断产生 1: 比较匹配中断产生

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

24.6.5 LPT_CTRL 控制寄存器

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.															LPTEN
															rw

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

24.6.6 LPT_CNT 计数寄存器

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
CNT[15:0]															
r															

Bit	Field	Description
15:0	CNT	LPTIM 计数器值

24.6.7 LPT_CMP 比较值寄存器

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
COMPARE_REG[15:0]															
rw															

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

注：COMPARE_REG 应在 LPTEN 为 0 时配置。

24.6.8 LPT_TARGET 目标值寄存器

偏移地址：0x18

复位值：0x0000

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

TARGET_REG[15:0]

RW

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

注：TARGET_REG 应在 LPTEN 为 0 时配置。

25 IWDG 独立看门狗

25.1 简介

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

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

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

25.2 功能框图

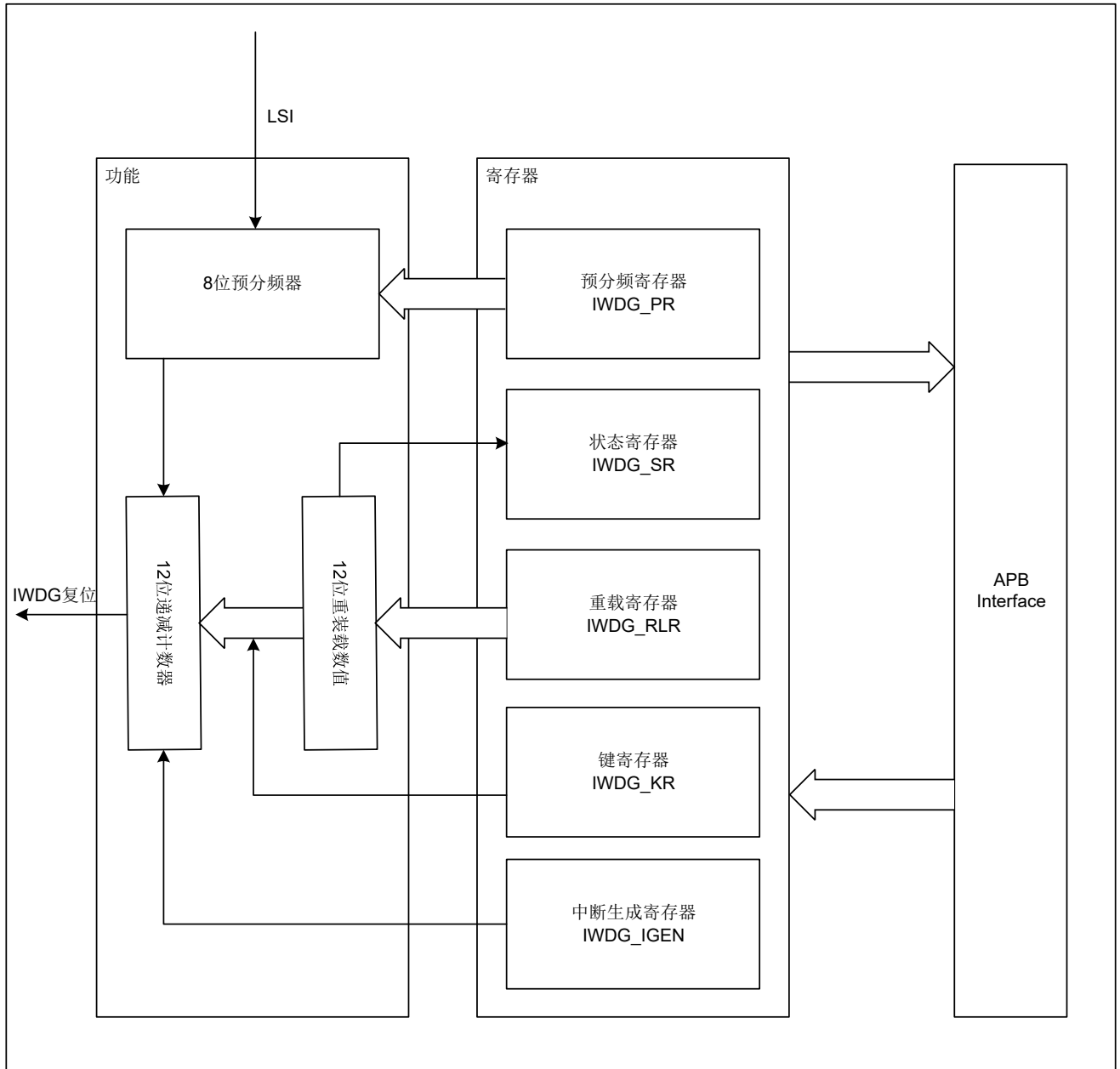


图 25-1 功能框图

25.3 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。
- 工作时钟由 **LSI** 提供，可在低功耗模式下继续进行工作，参考 **PWR** 章节；

- 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。

25.4 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 0xFFFF 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。
- 独立看门狗可以在低功耗模式下正常计数，它的复位能够使系统退出低功耗模式。
- 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。

25.4.1 流程框图

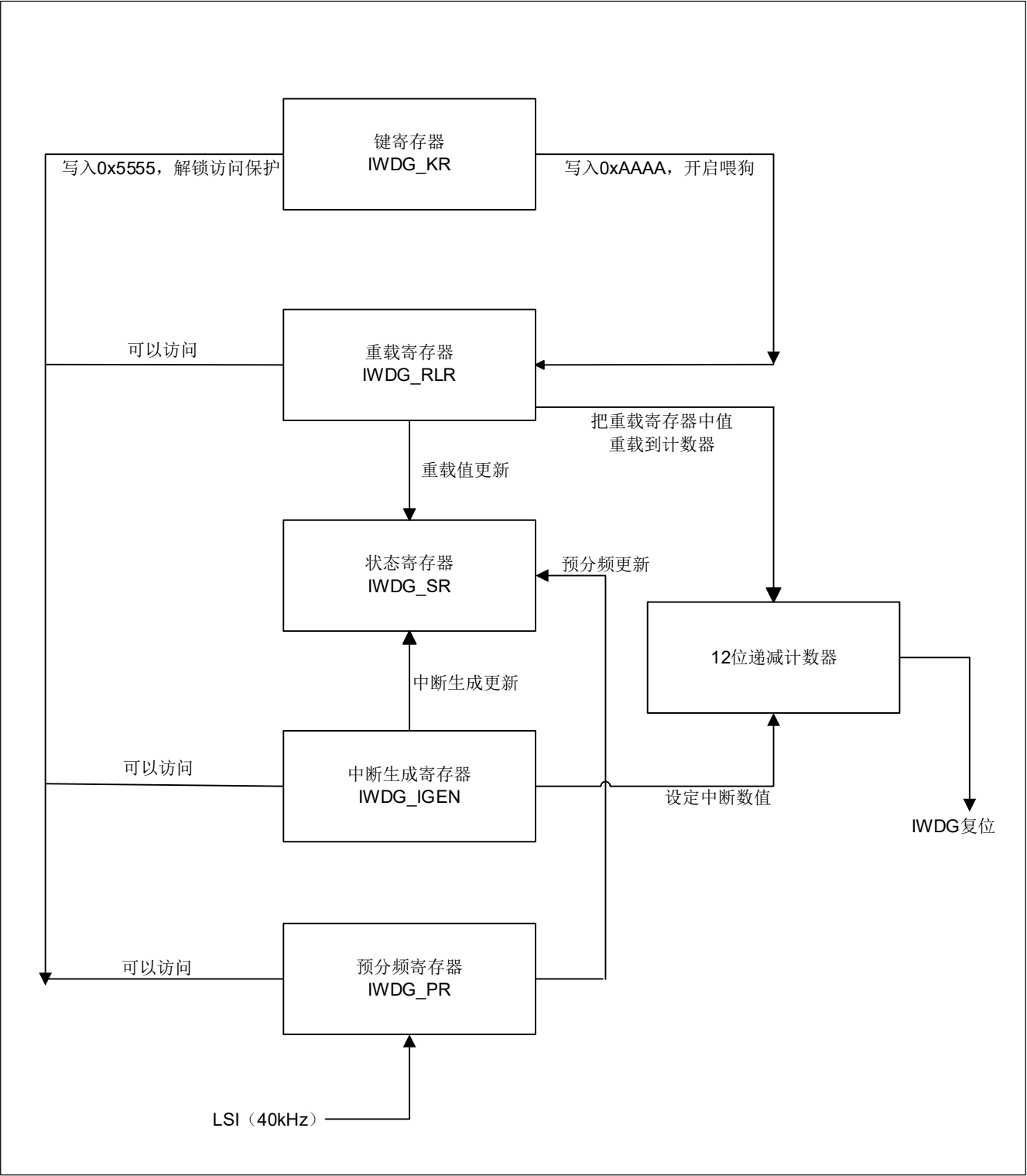


图 25-2 流程框图

25.4.2 独立看门狗超时时间

表 25-1 IWDG 超时时间（以 LSI 时钟频率 40kHz 为例）

预分频系数	PR[2:0] 位	最短时间 (ms) RL[11:0]=0x000	最长时间 (ms)
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 或 7)	6.4	26214.4

超出（溢出）时间计算：

$$T_{out} = ((4 * 2^{PR}) * RL) / 40$$

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) * 看门狗时钟周期。

25.4.3 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。

25.5 寄存器

25.5.1 寄存器总览

表 25-2 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

25.5.2 IWDG_KR 键寄存器

偏移地址：0x00

复位值：0x0000 0000

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

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

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	KEY	键值（只写寄存器）（Key value） 软件每隔一段时间，写进 0xAAAA 进行喂狗操作，否则当计数器递减到 0x0000 时，会产生一个复位信号，使系统复位。 软件写入 0x5555 表示解除保护，可以访问其他配置寄存器（IWDG_PR、IWDG_RLR、IWDG_CR（bit0），IWDG_IGEN） 软件写入 0xCCCC，开启看门狗。

25.5.3 IWDG_PR 预分频寄存器

偏移地址：0x04

复位值：0x0000 0000（系统复位）

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		
													rw		

Bit	Field	Description								
31:3	Reserved	始终读为 0								
2: 0	PR	预分频因子（Prescaler divider） 通过设置这些位来选择 LSI 时钟的预分频因子 要改变预分频因子，需要先解除保护（向 IWDG_KE 中写入 0x5555）才能写入，当预分频因子被更新完成后，PUV 寄存器位会变为 0，此时读出数据才是有效的。 <table><tr><td>000: 预分频因子 = 4</td><td>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									

25.5.4 IWDG_RLR 重装载寄存器

偏移地址：0x08

复位值：0x0000 0FFF（系统复位）

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											
				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。

25.5.5 IWDG_SR 状态寄存器

偏移地址：0x0C

复位值：0x0000 0000（RCC 复位和系统复位）

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
												r	r	r	r

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值
3	UPDATE	看门狗重装载值更新标志 当 IWDG_KR 寄存器中写入 0xAAAA 时，update 置位，当看门狗计数器被更新，重装载值写入到计数器中，update 自动清零。
2	IVU	看门狗中断生成值更新（Watchdog Interrupt Generate value update） 此位由硬件置‘1’用来指示中断生成值的更新正在进行中。 当在 VDD 域中的中断生成值更新结束后，此位由硬件清‘0’（最多需要 5 个 LSI 的振荡器周期）中断生成值只有在 IVU 位被清‘0’后才可更新。

Bit	Field	Description
1	RVU	看门狗计数器重装载值更新 (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 LSI 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 LSI 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。

25.5.6 IWDG_CR 控制寄存器

偏移地址: 0x10

复位值: 0x0000 0000 (RCC 复位和系统复位)

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.														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

25.5.7 IWDG_IGEN 中断生成寄存器

偏移地址: 0x14

复位值: 0x0000 0FFF (系统复位)

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.				IGEN											
				rw											

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11: 0	IGEN	IWDG 中断生成值（Watchdog Interrupt Generate value） 用于定义看门狗中断生成值，每当计数器值递减等于该值时，会产生中断。 要改变该位数值，需要先解除保护。 当改变数值被更新完成后 IWDG_SR 寄存器中的 IVU 位为清 0 此时，读出数据才是有效的。

25.5.8 IWDG_CNT 计数寄存器

偏移地址：0x18

复位值：0x0000 0000（RCC 复位和系统复位）

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.				IWDG_CNT											
				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

25.5.9 IWDG_PS 分频计数寄存器

偏移地址：0x1C

复位值：0x0000 0001（RCC 复位和系统复位）

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.								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

26 WWDG 窗口看门狗

26.1 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。

26.2 功能框图

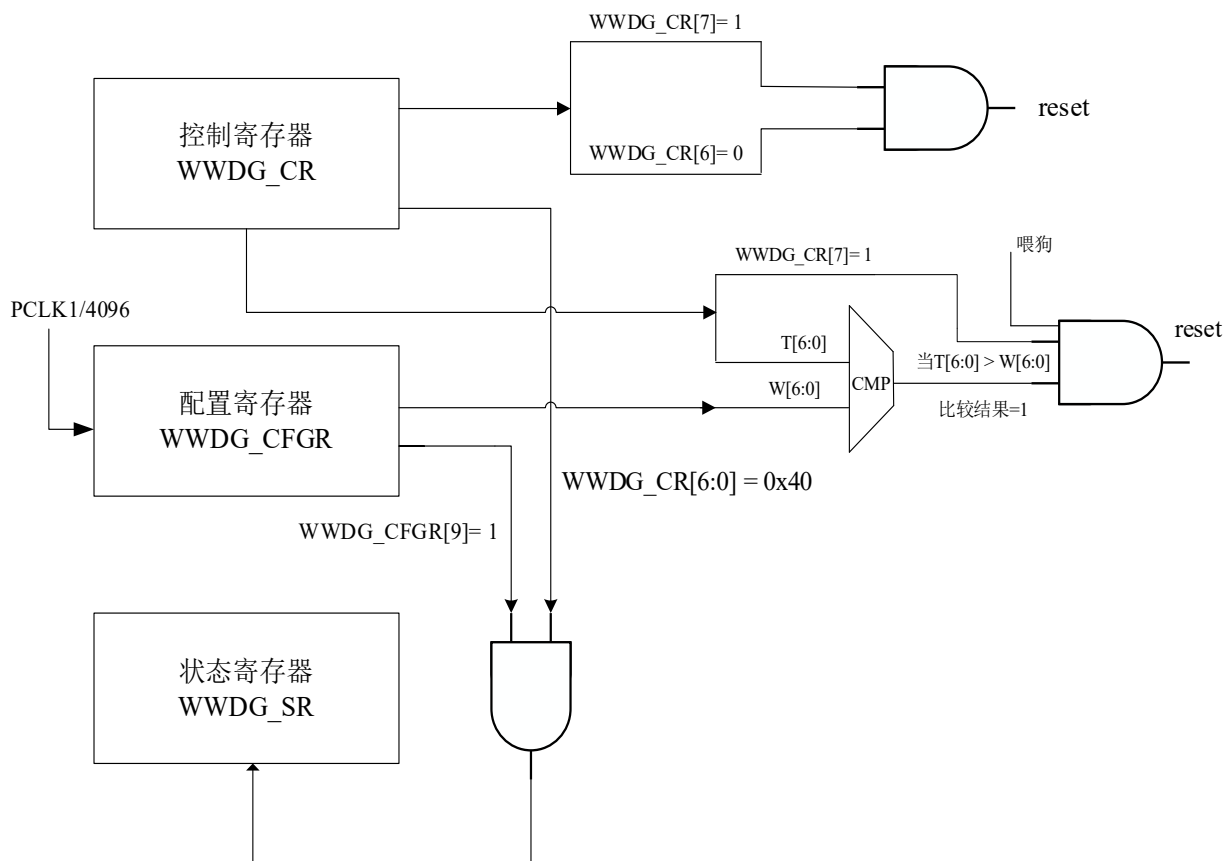


图 26-1 WWDG 功能框图

26.3 主要特征

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

- ### ● 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_CFR**）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。

26.4 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。

26.5 窗口看门狗超时时间

T[6:0]CNT递减计数器

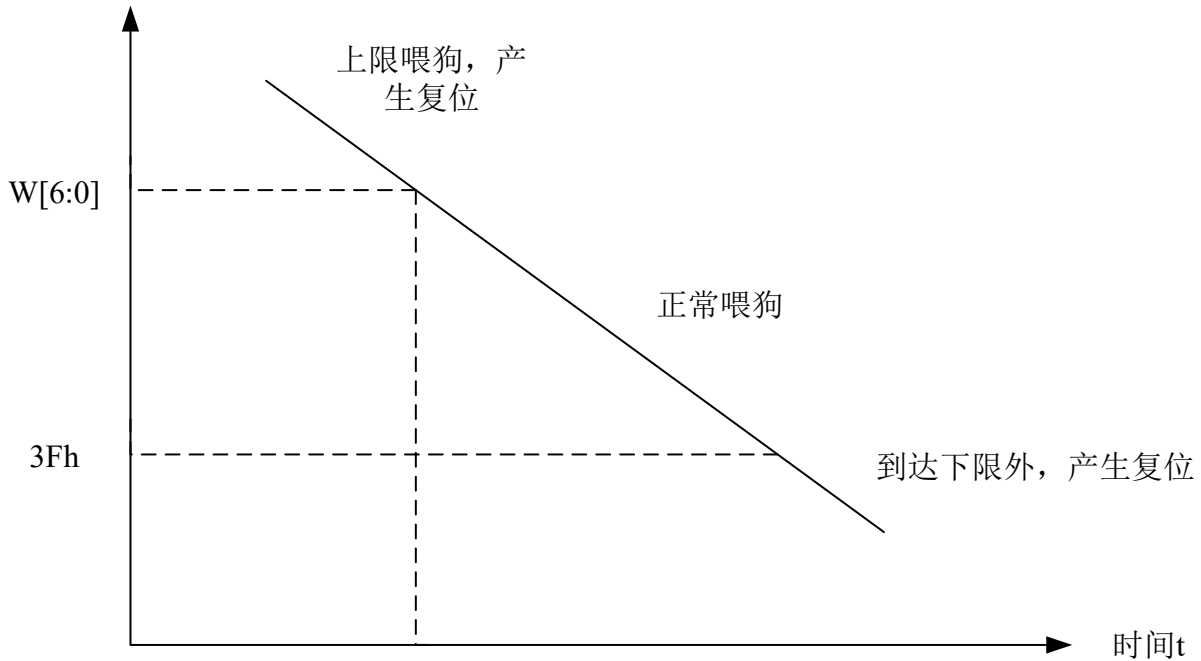


图 26-2 WWDG 超时时间坐标图

表 26-1 超时时间 ($f_{\text{pclk1}}=45\text{MHz}$)

时基 (WDGTB)	最短时间	最长时间
0	90.93us	5.83ms
1	181.86us	11.65ms
2	363.72us	23.3ms
3	727.45us	46.6ms

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

26.6 寄存器

表 26-2 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

26.6.1 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WDGA	T[6:0]						
								rw	rw	rw	rw	rw	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器值。每 (4096x2 ^{WDGTB}) 个 PCLK1 周期减 1, 当计数器值从 40h 递减为 3Fh 时 (T[6]变成 0), 产生看门狗复位。

26.6.2 配置寄存器 (WWDG_CFGR)

偏移地址: 0x04

复位值: 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EWI	WDGTB	W[6:0]						
							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	EWI	提前唤醒中断使能 (Early Wakeup Interrupt) 1: 当计数器值达到 40h, 产生中断。 0: 看门狗中断禁止 该位只能由硬件在复位后清除。

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

26.6.3 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWIF
															w0c

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值。
0	EWIF	EWIF: 提前唤醒中断标志 (Early Wakeup Interrupt Flag) 当计数器值达到 40h, 此位由硬件置 1; 通过软件写 0 来清除, 写 1 无效。

27 RTC 实时时钟器

27.1 RTC 简介

RTC 模块内部包含一组连续计数的计数器。它作为一个独立的定时器，在相应软件配置下，可提供时钟功能。修改计数器的值可以重新设置系统当前的时间。

RTC 模块处于备份域中，即在当芯片被系统复位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。

27.2 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；

27.3 功能描述

27.3.1 功能框图

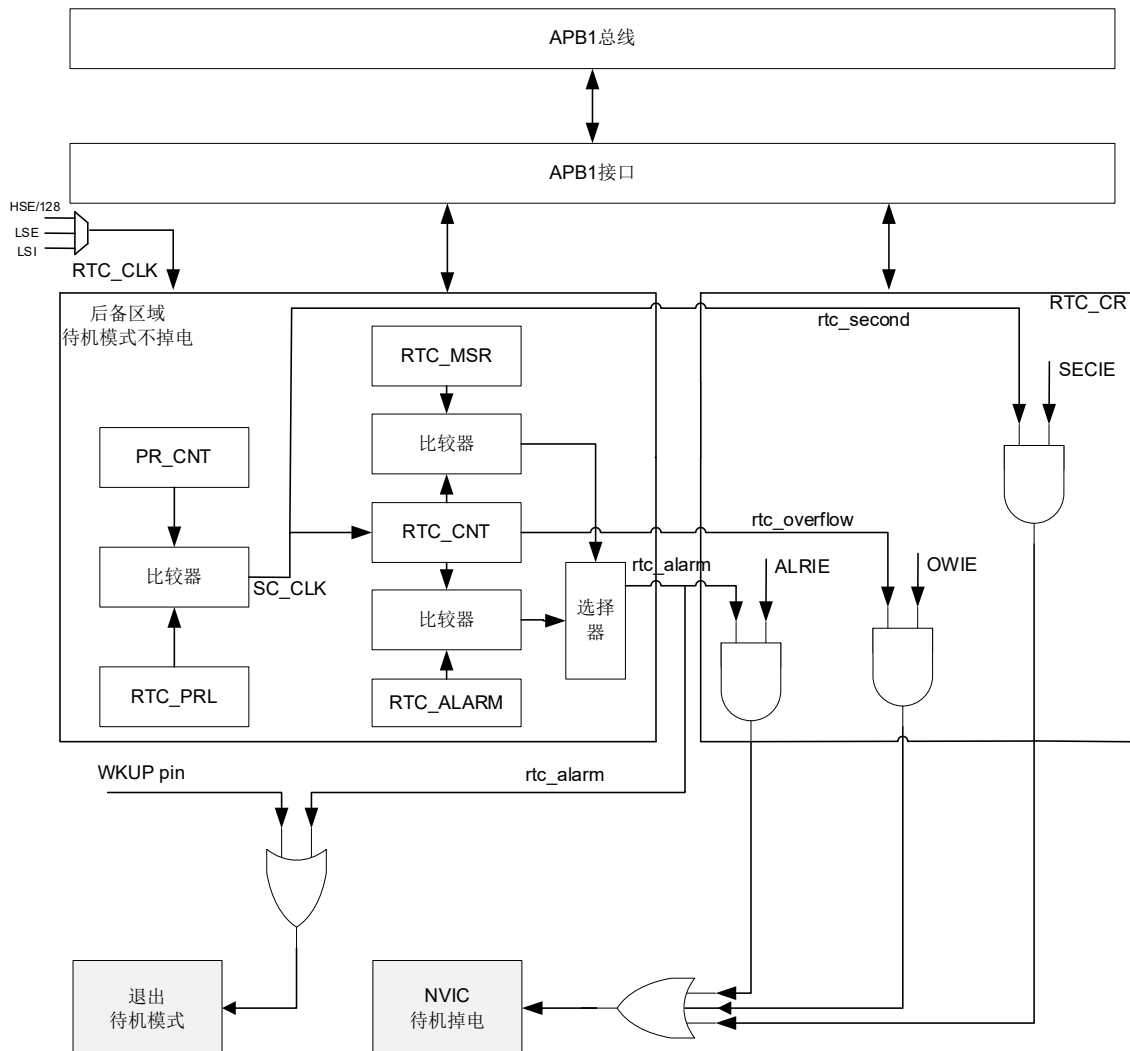


图 27-1 RTC 功能框图

27.3.2 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。

27.3.3 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。

27.3.4 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_CR1 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。

27.3.5 寄存器配置

配置 RTC_CR1 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_CR1 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；

27.3.6 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。

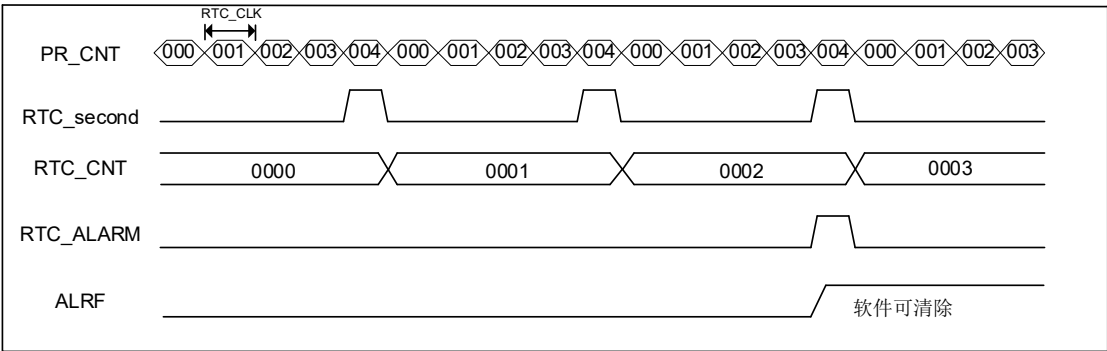


图 27-2 RTC 秒和闹钟波形图示例，RTC_PRL = 0004，RTC_ALR = 002

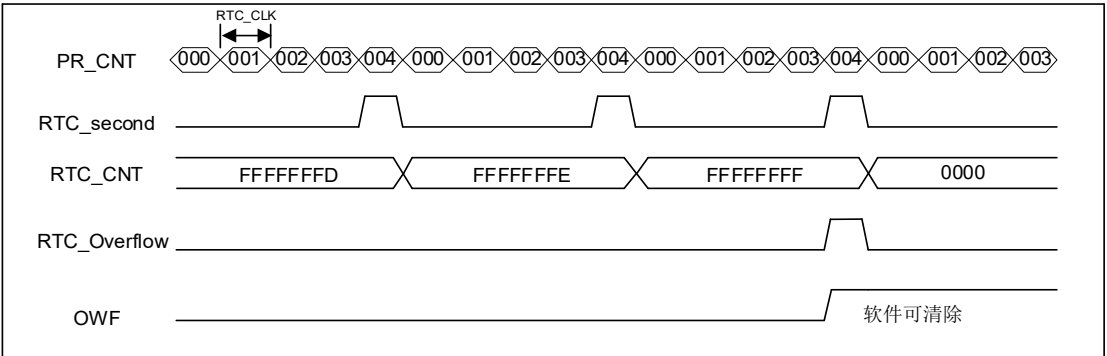


图 27-3 RTC 溢出波形图示例，RTC_PRL = 0004

27.3.7 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。

27.3.8 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。

27.4 寄存器描述

27.4.1 寄存器总览

表 27-1 RTC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RTC_CRH	RTC 控制寄存器高位	0x0000
0x04	RTC_CRL	RTC 控制寄存器低位	0x0020
0x08	RTC_PRLH	RTC 预分频装载寄存器高位	0x0000
0x0C	RTC_PRLH	RTC 预分频装载寄存器低位	0x8000
0x10	RTC_DIVH	RTC 预分频器分频因子寄存器高位	0x0000
0x14	RTC_DIVL	RTC 预分频器分频因子寄存器低位	0x0000
0x18	RTC_CNTH	RTC 计数器寄存器高位	0x0000
0x1C	RTC_CNTL	RTC 计数器寄存器低位	0x0000
0x20	RTC_ALRH	RTC 闹钟寄存器高位	0xFFFF
0x24	RTC_ALRL	RTC 闹钟寄存器低位	0xFFFF
0x28	RTC_MSRH	RTC 毫秒寄存器高位	0x0000
0x2C	RTC_MSRL	RTC 毫秒寄存器低位	0x0000
0x3C	RTC_LSE_CFG	RTC LSE 配置寄存器	0x0250

27.4.2 控制寄存器高位 (RTC_CRH)

偏移地址: 0x0

复位值: 0x0000

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

Bit	Field	Description
15:3	Reserved	保留, 必须保持复位值
2	OWIE	溢出中断使能位 (Overflow Interrupt Enable) 0: 溢出中断禁止 1: 溢出中断使能
1	ALRIE	闹钟中断使能位 (Alarm Interrupt Enable) 0: 闹钟中断禁止 1: 闹钟中断使能
0	SECIE	秒中断使能位 (Second Interrupt Enable) 0: 秒中断禁止 1: 秒中断使能

27.4.3 控制寄存器低位(RTC_CRL)

偏移地址: 0x04

复位值: 0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ALPEN	RTOFF	CNF	RSF	OWF	ALRF	SECF
									rw	r	rw	w0c	w0c	w0c	w0c

Bit	Field	Description
15:7	Reserved	保留, 必须保持复位值
6	ALPEN	RTC 闹钟循环使能 (RTC Alarm Loop Enable) 0: 单次发生闹钟事件 1: 循环发生闹钟事件
5	RTOFF	RTC 操作状态 (RTC Operation OFF) 指示对寄存器最后一次写操作是否完成。若此位为'0', 则表示无法对任何的 RTC 寄存器进行读写操作。此位只读。 0: 对 RTC 寄存器的写操作仍在进行 1: 完成对 RTC 寄存器的写操作

Bit	Field	Description
4	CNF	配置标志（Configuration Flag） 只能通过软件置'1'进入配置模式，之后可以向 RTC_CNTL/H、RTC_ALRL/H、RTC_MSRL/H 或 RTC_PRL/H 寄存器写入数据。由软件清'0'后，才能进入下一次对寄存器的写操作过程。 0：退出配置模式 1：进入配置模式
3	RSF	寄存器同步标志（Registers Synchronized Flag） 由软件清零，由硬件置 1。 用户必须等待这位被硬件置'1'，才能进行读操作。确保同步后的 RTC_CNT、RTC_ALR、RTC_PRL、RTC_DIV 以及 RTC_MSR 寄存器的值正确。 0：寄存器没有同步 1：寄存器已经同步
2	OWF	溢出标志（Overflow Flag） 当可编程计数器溢出时，硬件置'1'，通过配置 RTC_CRH 寄存器中的 OWIE 位为 1，可以产生中断。 软件清除该位 0：无溢出 1：32 位可编程计数器溢出
1	ALRF	闹钟标志（Alarm Flag） 可编程计数器等于 RTC_ALR 寄存器设定值时，硬件置'1'，通过配置 RTC_CRH 寄存器中的 ALRIE 位为 1，可以产生中断。 软件清除该位 0：无闹钟产生 1：有闹钟产生
0	SECF	秒标志（Second Flag） 预分频器溢出时，硬件置'1'该位，RTC 计数器加 1。 该标志为可编程的 RTC 计数器提供计数时钟（常用周期为 1 秒） 通过配置 RTC_CTH 寄存器中的 SECIE 位为 1，可以产生中断。 软件清除该位 0：没有秒标志 1：产生秒标志

注：

- 当状态位 RTOFF=0 时，表示上一次写操作没有完成，此时不能对 RTC_CR 寄存器进行写操作；
- 复位时禁止所有中断，无任何挂起的中断请求，可以写 RTC 寄存器；
- 硬件置位 OWF、ALRF、SECF 和 RSF 等标志位，软件清零；
- 若关闭 APB1 时钟，OWF、ALRF、SECF 和 RSF 位值保持不变；
- 标志位置位后将保持挂起状态，直到中断响应且软件配置 RTC_CR 清除该标志位；

27.4.4 预分频装载寄存器高位(RTC_PRLH)

偏移地址: 0x08

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRL[19:16]			
												w			

Bit	Field	Description
15:4	Reserved	保留, 必须保持复位值
3:0	PRL[19:16]	RTC 预分频器重载值高位 (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。

27.4.5 预分频装载寄存器低位(RTC_PRL)

偏移地址: 0x0C

复位值: 0x8000

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

Bit	Field	Description
15:0	PRL[15:0]	RTC 预分频器重载值低位 (RTC Prescaler Reload Value Low)

27.4.6 预分频器分频因子寄存器高位(RTC_DIVH)

偏移地址: 0x10

复位值: 0x0000

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

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

Bit	Field	Description
3:0	DIV[19:16]	RTC 时钟分频器分频因子高位 (RTC Clock Divider High) 只读寄存器，在 RTC_PRL 或 RTC_CNT 寄存器被软件更新后，由硬件重新装载。 在每个 SC_CLK 时钟周期里，RTC 预分频器的值会被 RTC_PRL 寄存器的值重置。通过读取 RTC_DIV 寄存器可以得到预分频计数器的当前值或当前精确的时间。

27.4.7 预分频器分频因子寄存器低位(RTC_DIVL)

偏移地址：0x14

复位值：0x0000

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

Bit	Field	Description
15:0	DIV[15:0]	RTC 时钟分频器分频因子低位 (RTC Clock Divider Low) 只读寄存器，在 RTC_PRL 或 RTC_CNT 寄存器被软件更新后，由硬件清零。 在每个 SC_CLK 时钟周期里，RTC 预分频器的值会被 RTC_PRL 寄存器的值重置。通过读取 RTC_DIV 寄存器可以得到预分频计数器的当前值或当前精确的时间。

27.4.8 计数器寄存器高位(RTC_CNTH)

偏移地址：0x18

复位值：0x0000

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

Bit	Field	Description
15:0	CNT[31:16]	RTC 计数器高位 (RTC Counter High) 通过读 RTC_CNTH 寄存器获取 RTC 计数器当前值的高位部分。先进入配置模式。才能对该寄存器执行写操作。 该寄存器受 RTC_CR 的位 RTOFF 写保护。 写入该寄存器的值，会直接加载到对应的编程计数器。软件可通过读取该寄存器的值，获得计数器计数值或显示系统时间。

27.4.9 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]	RTC 计数器低位 (RTC Counter Low) 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。 该寄存器受 RTC_CR 的位 RTOFF 写保护。 写入该寄存器的值, 会直接加载到对应的编程计数器。软件可通过读取该寄存器的值, 获得计数器计数值或显示系统时间。

27.4.10 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]	RTC 闹钟值高位 (RTC Alarm High) 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。

27.4.11 闹钟寄存器低位(RTC_ALRL)

偏移地址: 0x24

复位值: 0xFFFF

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

Bit	Field	Description
15:0	ALR[15:0]	RTC 闹钟值低位 (RTC Alarm Low) 软件写入的闹钟时间的低位部分。先进入配置模式, 才能对该寄存器执行写操作。

27.4.12 毫秒闹钟寄存器高位 (RTC_MSRLH)

偏移地址: 0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MSR[19:16]			
												rw			

Bit	Field	Description
15:4	Reserved	保留，必须保持复位值
3:0	MSR[19:16]	RTC 闹钟值高位（RTC Msec High） 软件写入的毫秒闹钟时间的高位部分 注：进入配置模式，才能对该寄存器执行写操作。

27.4.13 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。

27.4.14 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

Reserved	LSE_AA C	LSE_IB	LSE_RFB_SEL	LSE_DR	Reserved	LSE_OUTENH	LSE_TC[1:0]
	rw	rw	rw	rw		rw	rw

Bit	Field	Description
15:11	Reserved	保留，必须保持复位值。
10	LSE_AAC	LSE 振荡幅度选择位 0: 正常模式 1: 振幅增大模式
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	Reserved	保留，必须保持复位值。
2	LSE_OUTENH	LSE 输出增强模式 0: 正常模式 1: 输出增强模式
1:0	LSE_TC[1:0]	LSE 测试补偿模式 00: 正常模式 01: 漏电弱补偿模式 10: 漏电常规补偿模式 11: 漏电增强补偿模式

28 BKP 备份寄存器

28.1 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。

28.2 主要特征

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

28.3 功能描述

28.3.1 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*10⁶。

若校准值为 1，则 RTC 校准时，每 2²⁰ 个时钟周期扣除 1 个时钟脉冲，这相当于 0.954ppm(1/2²⁰*10⁶ = 0.954)。而 BKP_RTCCR 寄存器校准值 CAL 最大为 127，所以最大可以减慢 121ppm(0.954ppm*127 = 121)。

28.3.2 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。

28.4 寄存器

28.4.1 寄存器总览

表 28-1 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

28.4.2 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）。

28.4.3 时钟校准寄存器 (BKP_RTCCR)

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ASOS	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 2^{20} 个时钟脉冲）。可以用来对 RTC 进行校准，以 $1/2^{20} * 10^6$ ppm 的比例减慢时钟。 注：RTC 时钟可以被减慢 0~121ppm。

28.4.4 备份控制寄存器 (BKP_CR)

偏移地址：0x44

复位值：0x0000

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

28.4.5 备份控制状态寄存器 (BKP_CSR)

偏移地址：0x48

复位值：0x0000

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

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

29 UART 通用异步收发器

29.1 UART 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。

29.2 UART 功能框图

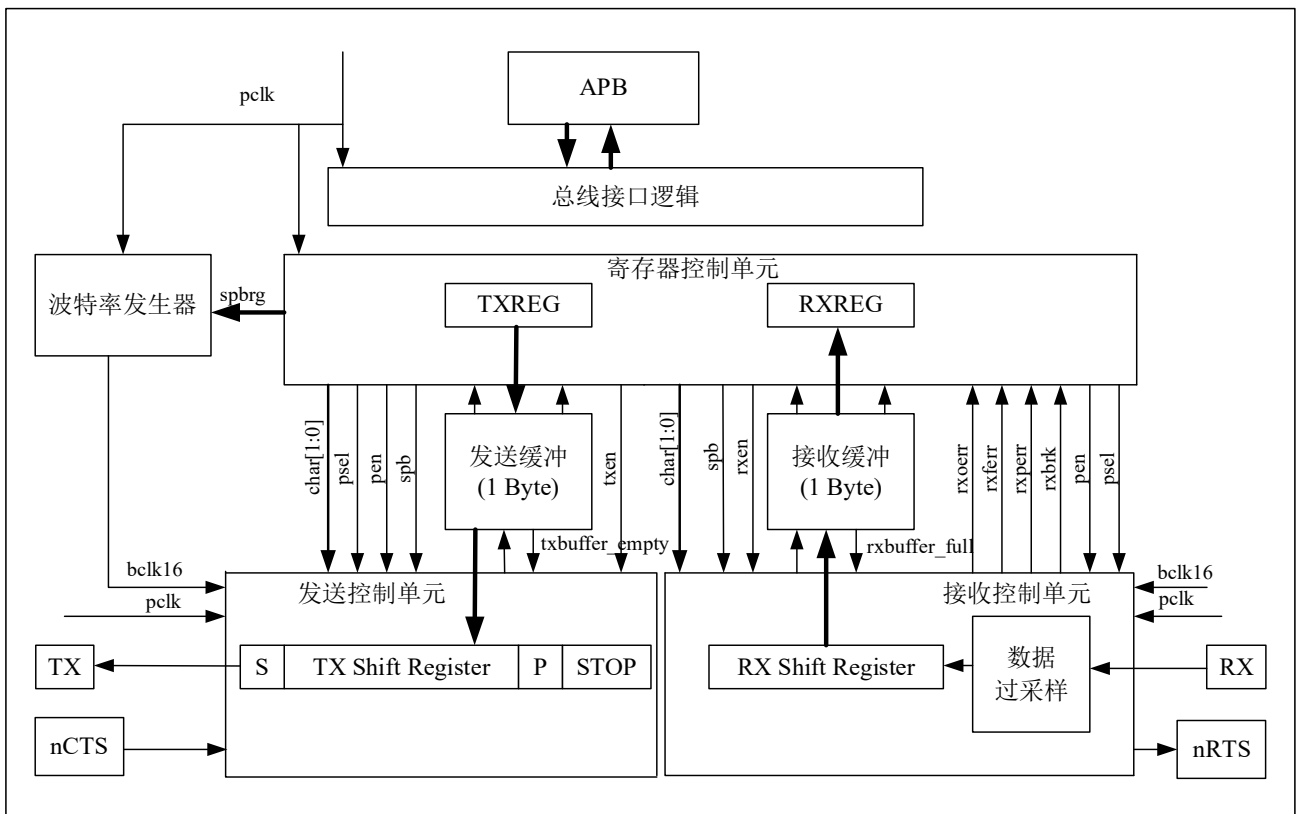


图 29-1 UART 功能框图

29.3 UART 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

29.4 UART 功能概述

在全双工通信的情况下，至少需要分配两个引脚给 **UART**：接收数据输入(**RX**)和发送数据输出(**TX**)。

RX：外部串行数据通过该引脚，传送给 **UART** 接收器。对于传输过程中产生的噪音，可以使用过采样的技术将其与数据区分并剔除，得到原本的数据。

TX：**UART** 发送器内部产生的串行数据通过该引脚发送输出。当发送器被使能，并且无数据发送时，**TX** 引脚输出高电平。当发送器被除能时，**TX** 引脚恢复到它的 I/O 端口配置状态。

空闲状态为总线在开始发送或者开始接收前的初始状态。

起始位为一位，用‘0’表示。

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

停止位用‘1’表示一帧的结束，位数可配置为 0.5、1、1.5、2 位。

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

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

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

在硬件流控模式中还需要下列引脚：

nCTS 清除发送：当其为高电平时，表明当前接收端不能进行数据接收，发送端应停止之后的发送。

nRTS 请求发送：当其为低电平时，表明当前接收端可以接收数据。

29.4.1 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。

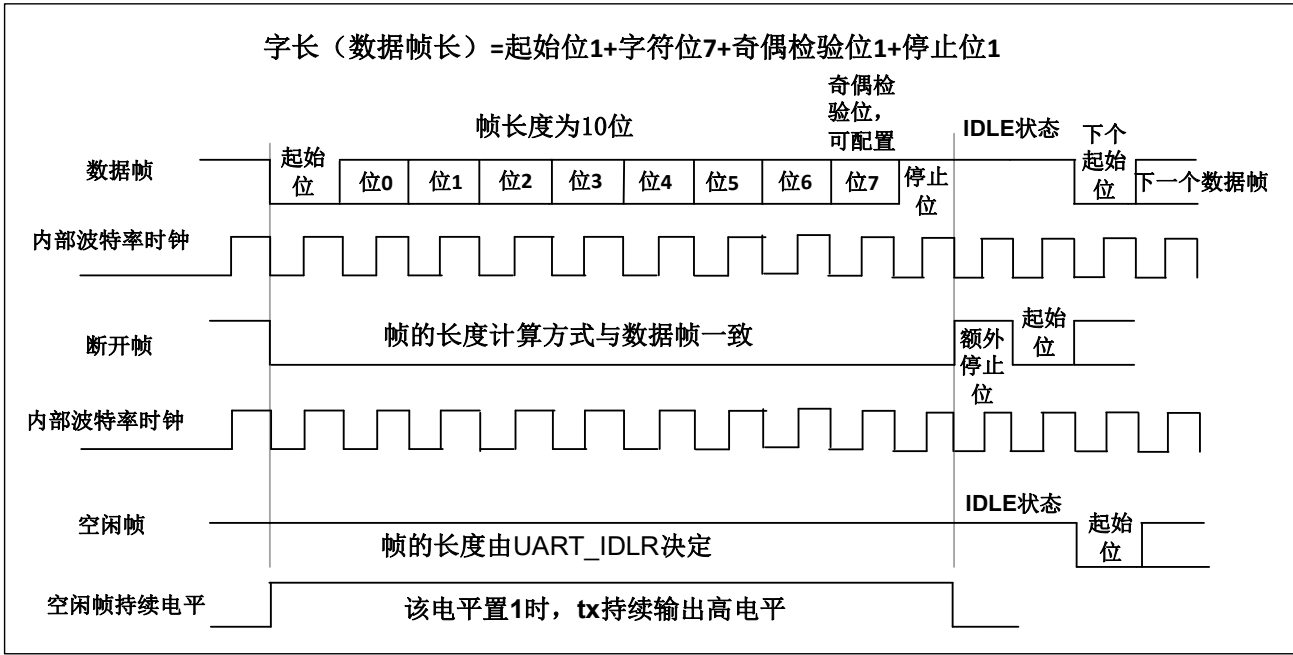


图 29-2 UART 时序

29.4.2 分数波特率发生器

设置 BRR 和 FRA 寄存器，可设置相应波特率，参考如下公式：

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

其中 BRR 寄存器最小值为 1，FRA 的值可配置在 0~15 范围。

29.4.3 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。

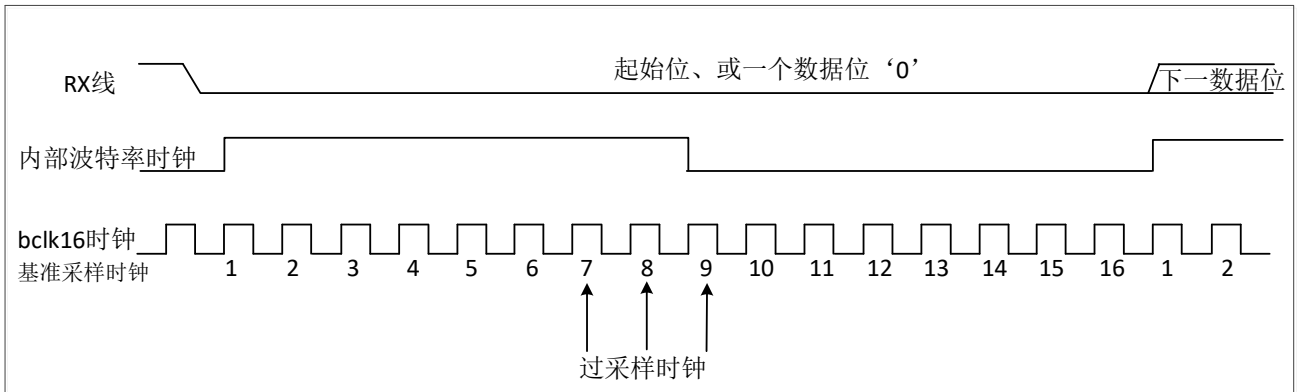


图 29-3 RX 引脚采样方法

29.4.4 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。

29.4.5 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。

29.4.6 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。

29.4.6.1 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。

29.4.6.2 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。

29.4.6.3 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。

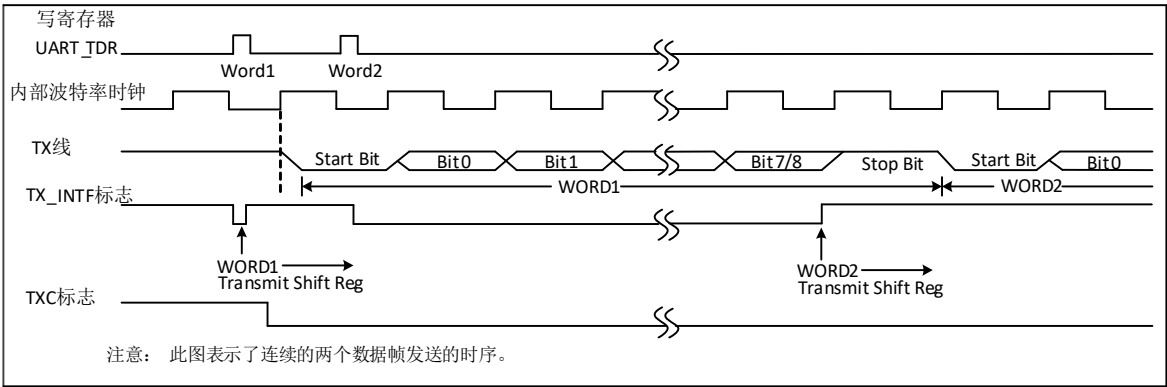


图 29-4 发送时状态位变化

29.4.6.4 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。

29.4.6.5 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。

29.4.7 接收器

29.4.7.1 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。

29.4.7.2 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。

29.4.7.3 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。

29.4.7.4 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。

29.4.7.5 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。

29.4.8 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。

29.4.9 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。

29.4.10 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。

29.4.10.1 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。

29.4.10.2 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。

29.4.10.3 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。

29.4.11 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）。

29.4.12 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。

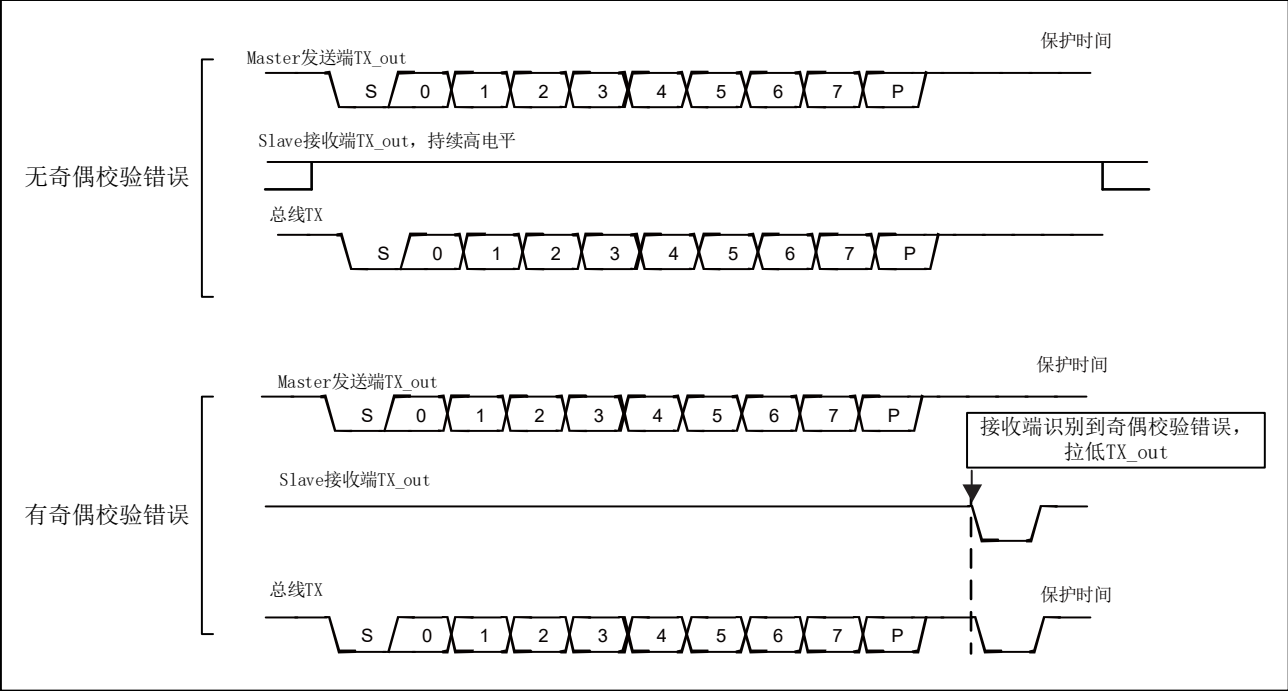


图 29-5 UART 奇偶校验方框图

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：

- 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。

- 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）。

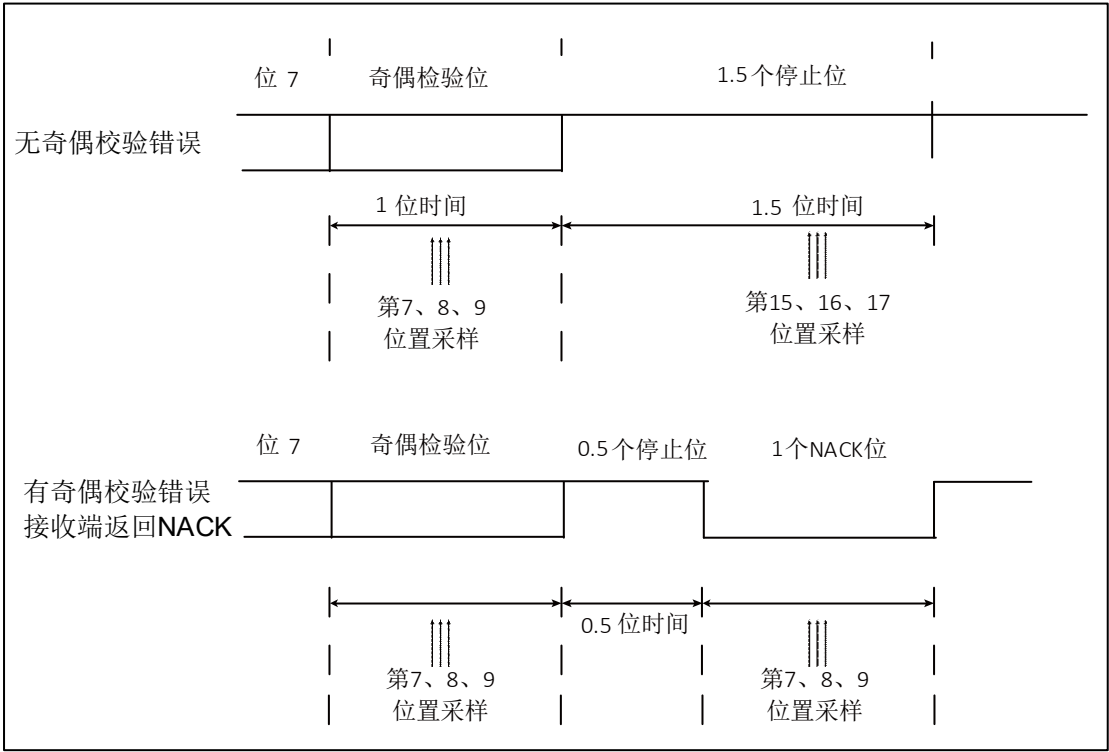


图 29-6 UART 采样 NACK 信号方框图

29.4.13 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 MHz < sirip_clk < 2.12 MHz)，脉冲的宽度是 sirip_clk*3，sirip_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 Sirip=1。

下图示意了数据格式转换的样式：

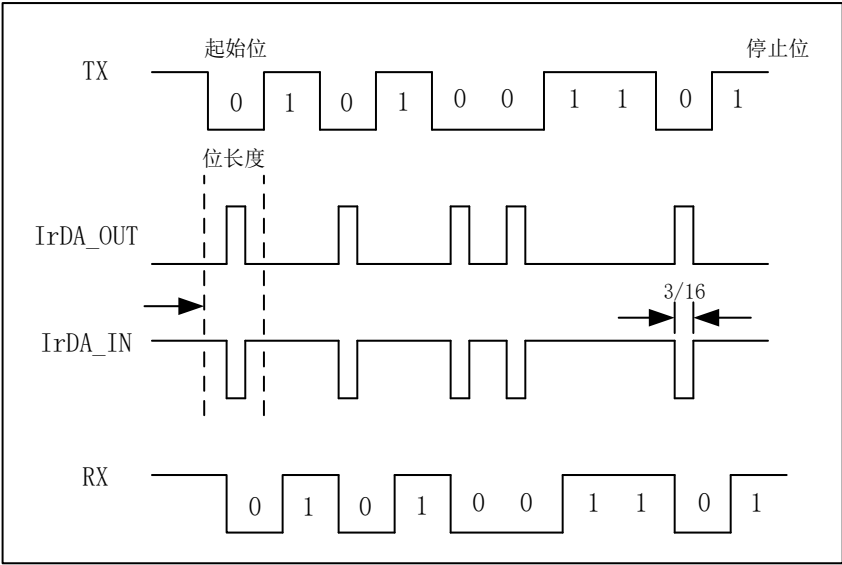


图 29-7 普通模式下 IrDA 发送和接收图

29.5 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 29-1 UART 中断请求

中断事件	中断状态	使能位
发送缓冲空	TX_INTF	TXIEN
接收到有效数据	RX_INTF	RXIEN
发送移位寄存器完成	TXC_INTF	TXC_EN
接收溢出错误	RXOERR_INTF	RXOERREN
奇偶校验错误	RXPERR_INTF	RXPERREN
帧错误	RXFERR_INTF	RXFERREN

中断事件	中断状态	使能位
接收断开帧	RXBRK_INTF	RXBRKEN
发送断开帧	TXBRK_INTF	TXBRK_EN
接收同步帧	RXB8_INTF	RXB8_EN
接收空闲帧	RXIDLE_INTF	RXIDLEN
自动波特率结束	ABREND_INTF	ABRENDIEN
自动波特率错误	ABRERR_INTF	ABRERRIEN

如果设置了对应的中断使能控制位，这些设置就可以产生各自对应的中断。

29.6 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。

29.7 寄存器

29.7.1 寄存器总览

表 29-2 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

Offset	Acronym	Register Name	Reset
0x2C	UART_RXMASK	UART 接收掩码寄存器	0x000000FF
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

29.7.2 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							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）

29.7.3 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							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

29.7.4 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TXEPT	TXFULL	RXAVL	TXC
												r	r	r	r

Bit	Field	Description
31: 4	Reserved	保留，始终读为 0
3	TXEPT	发送缓冲空标识位（Transmit buffer empty flag bit） 0：发送缓冲不为空 1：发送缓冲为空
2	TXFULL	发送缓冲满标识位（Transmit buffer full flag bit） 0：发送缓冲不满 1：发送缓冲满
1	RXAVL	接收有效字节数据标识位（Receive valid data flag bit） 当接收缓冲接收了一个完整字节的数据时置位该位。 0：接收缓冲为空 1：接收缓冲接收了一个完整有效的字节数据
0	TXC	发送完成标识位（Transmit complete flag bit） 0：移位寄存器发送未完成 1：移位寄存器数据发送完成 注：此标识位在智能卡（ISO7816）时与保护时间（UART_SCR.SCFCNT 配置）相关。

29.7.5 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ABRER	ABREN	RXIDLE	RXB8_I	TXBRK_	RXBRK	RXFER	RXPER	RXOER	TXC_IN	RX_INT	TX_INT
				R_INTF	D_INTF	_INTF	NTF	INTF	_INTF	R_INTF	R_INTF	R_INTF	TF	F	F
				r	r	r	r	r	r	r	r	r	r	r	r

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
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

Bit	Field	Description
9	RXIDLE_INTF	UART 接收空闲帧中断标志位 (Receive frame idle interrupt flag bit) 在停止位后 RX 引脚在一段时间内接收到若干 (UART_IDLR 数量) 高电平。 0: 没有检测到空闲帧 1: 检测到空闲帧
8	RXB8_INTF	UART 同步帧中断标志位 在 9 位通讯模式下, 当接收到数据的第九位与寄存器 CCR.B8POL 相同时, RXB8_INT 位置。该位可以作为中断请求信号 0: 没有接收到同步帧 1: 接收到同步帧
7	TXBRK_INTF	UART 断开帧发送完成中断标志位 1: 移位寄存器断开帧数据发送完成 0: 移位寄存器空或正在移位发送备注: 不能连续发送断开帧。
6	RXBRK_INTF	UART 接收断开帧中断标志位 (Receive frame break interrupt flag bit) 在异常停止位后 RX 引脚在一段时间内接收到 10 个或大于 10 位的低电平。 0: 没有检测到断开帧 1: 检测到断开帧
5	RXFERR_INTF	帧错误中断标志位 (Frame error interrupt flag bit) 帧错误发生在当检测到异常停止位。 0: 没有检测到帧错误 1: 检测到帧错误
4	RXPERR_INTF	奇偶校验错误中断标志位 (Parity error interrupt flag bit) 0: 没有检测到奇偶校验错误 1: 检测到奇偶校验错误
3	RXOERR_INTF	接收溢出错误中断标志位 (Receive overflow error interrupt flag bit) 仅当 autoflowen=0 时会置位。 0: 没有溢出错误 1: 接收溢出错误
2	TXC_INTF	UART 发送移位寄存器完成中断标志位 0: 移位寄存器非空或正在移位发送 1: 移位寄存器数据发送完成 注: 此标志位在智能卡 (ISO7816) 时与保护时间 (UART_SCR.SCFCNT 配置) 相关。
1	RX_INTF	接收有效数据中断标志位 (Receive valid data interrupt flag bit) 当接收缓冲接收了一个完整字节的数据时置位该位。 0: 接收缓冲为空 1: 接收缓冲有效字节数据
0	TX_INTF	发送缓冲空中断标志位 (Transmit buffer empty interrupt flag bit) 0: 发送缓冲不为空 1: 发送缓冲空

29.7.6 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ABRER	ABREN	RXIDLE	RXB8_I	TXBRK_	RXBRK	RXFER	RXPER	RXOER	TXC_IE	RX_IEN	TX_IEN
				R_IEN	D_IEN	_IEN	EN	IEN	_IEN	R_IEN	R_IEN	R_IEN	N		
				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

29.7.7 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				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
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

Bit	Field	Description
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

29.7.8 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					TXTOG	RXTOG	SWAP	SELB8	Res.		TXEN	RXEN	AUTOFL	DMAMODE	UARTEN
					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
9	RXTOG	接收取反 0: 无效 1: 接收取反

Bit	Field	Description
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

29.7.9 UART_CCR 通用空制寄存器

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LIN	WAKE	RWU	B8EN	B8TOG	B8POL	B8TXD	B8RXD	SPB1	CHAR		BRK	SPB0	PSEL	PEN
	rw	rw	rw	rw	rw	rw	rw	r	rw	rw		rw	rw	rw	rw

Bit	Field	Description
31: 15	Reserved	保留, 始终读为 0

Bit	Field	Description
14	LIN	UART LIN 协议收发断开帧 (UART LIN enable bit) 0: LIN 协议无效 1: LIN 协议有效
13	WAKE	唤醒方法。 该位决定把 UART 唤醒的方法。 0: 空闲总线唤醒 1: 地址标记唤醒
12	RWU	接收静默。 该位用来决定是否把 UART 置于静默模式。该位可以由软件设置或清除。当唤醒序列到来时, 硬件也会自动将其清零。 0: 接收器处于正常工作模式 1: 接收器处于静默模式 在设置地址标记唤醒时, 如果接收 buffer 非空则不能软件修改。
11	B8EN	UART 同步帧第九位使能控制位。 该位使能后校验使能位 PEN 不起作用。 0: 禁止同步帧第九位发送 1: 使能同步帧第九位发送
10	B8TOG	UART 同步帧发送第九位自动翻转控制位。 0: 禁止第九位自动翻转 1: 使能第九位自动翻转 注: 在 B8TXD 和 B8POL 的值相同时, 在配置完寄存器后传输的第二个数据开始翻转, 第一个数据默认为地址位。
9	B8POL	UART 同步帧第九位极性控制位。 0: 同步帧第九位低电平有效 1: 同步帧第九位高电平有效
8	B8TXD	UART 同步帧发送数据第九位。 0: 发送同步帧第九位为低电平 1: 发送同步帧第九位为高电平
7	B8RXD	UART 同步帧接收数据第九位。 只读。 0: 接收同步帧第九位为低电平 1: 接收同步帧第九位为高电平
6	SPB1	停止位选择位, 与 SPB0 结合设置停止位位数。
5: 4	CHAR	UART 数据宽度位 (UART width bit) 00: 5 位; 01: 6 位 10: 7 位; 11: 8 位
3	BRK	UART 发送断开帧 (UART transmit frame break) 0: 不输出断开帧 1: 串行强制输出逻辑'0' (断开帧)

Bit	Field	Description
2	SPB0	停止位选择（Stop bit selection） 设置发送停止位位数。 SPB1, SPB0 00: 1 个停止位 SPB1, SPB0 01: 2 个停止位 SPB1, SPB0 10: 0.5 个停止位 SPB1, SPB0 11: 1.5 个停止位
1	PSEL	校验选择位（Parity selection bit） 当校验使能后，该位用于选择是采用偶校验还是奇校验。 0: 奇校验 1: 偶校验
0	PEN	校验使能位（Parity enable bit） 0: 禁止奇偶校验 1: 发送接收使能奇偶校验

29.7.10 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
Reserved															
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	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

29.7.11 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												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。

29.7.12 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								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。

29.7.13 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RXMASK							
								rw							

Bit	Field	Description
31: 8	Reserved	保留，始终读为 0
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。

29.7.14 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
Reserved															
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

29.7.15 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
Reserved															
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。

29.7.16 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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											Latter_ede	Former_ede	Abr_bitcnt		Abren
											rw	rw	rw		rw

Bit	Field	Description
31: 5	Reserved	保留，始终读为 0
4	Latter_edge	自动波特率后一个边沿选择。 0: 下降沿 1: 上升沿
3	Former_edge	自动波特率前一个边沿选择。 0: 下降沿 1: 上升沿
2: 1	Abr_bitcnt	自动波特率检测长度。 检测前一个边沿和后一个边沿之间的位长。 00: 1 位 01: 2 位 10: 4 位 11: 8 位
0	Abren	自动波特率使能。 只能在 UART 空闲时使能自动波特率，使能后检测接收信号的边沿，完成自动波特率检测后，硬件自动设置 UART_BRR 和 UART_FRA 寄存器。 0: 自动波特率禁止 1: 自动波特率使能

29.7.17 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

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

Bit	Field	Description
31: 16	Reserved	保留，始终读为 0
15: 8	PSC_REG	预分频寄存器（Prescaler value） 在红外低功耗模式下，对 UART 源时钟（pclk）分频获得低功耗模式下频率 sirIp_clk: 0000 0000 : 保留，不能写入该值 0000 0001 : 对源时钟 1 分频 0000 0010 : 对源时钟 2 分频 0000 0011 : 对源时钟 3 分频 注：保证分频后的时钟频率在（1.42 MHz < sirIp_clk < 2.12 MHz）之间。
7: 2	Reserved	保留，始终读为 0
1	SirIp	红外低功耗模式（IrDA low_power） 0 : 普通模式 1 : 低功耗模式
0	Siren	IrDA 红外模式使能（IrDA mode enable） 0 : 不使能红外模式 1 : 使能红外模式

30 LPUART 低功耗通用异步收发器

30.1 简介

LPUART 为低功耗通用异步收发器，相比 UART，其功耗极低，支持在低功耗模式运行以及唤醒芯片，支持唤醒的低功耗模式参考 PMU 模块。

30.2 功能框图

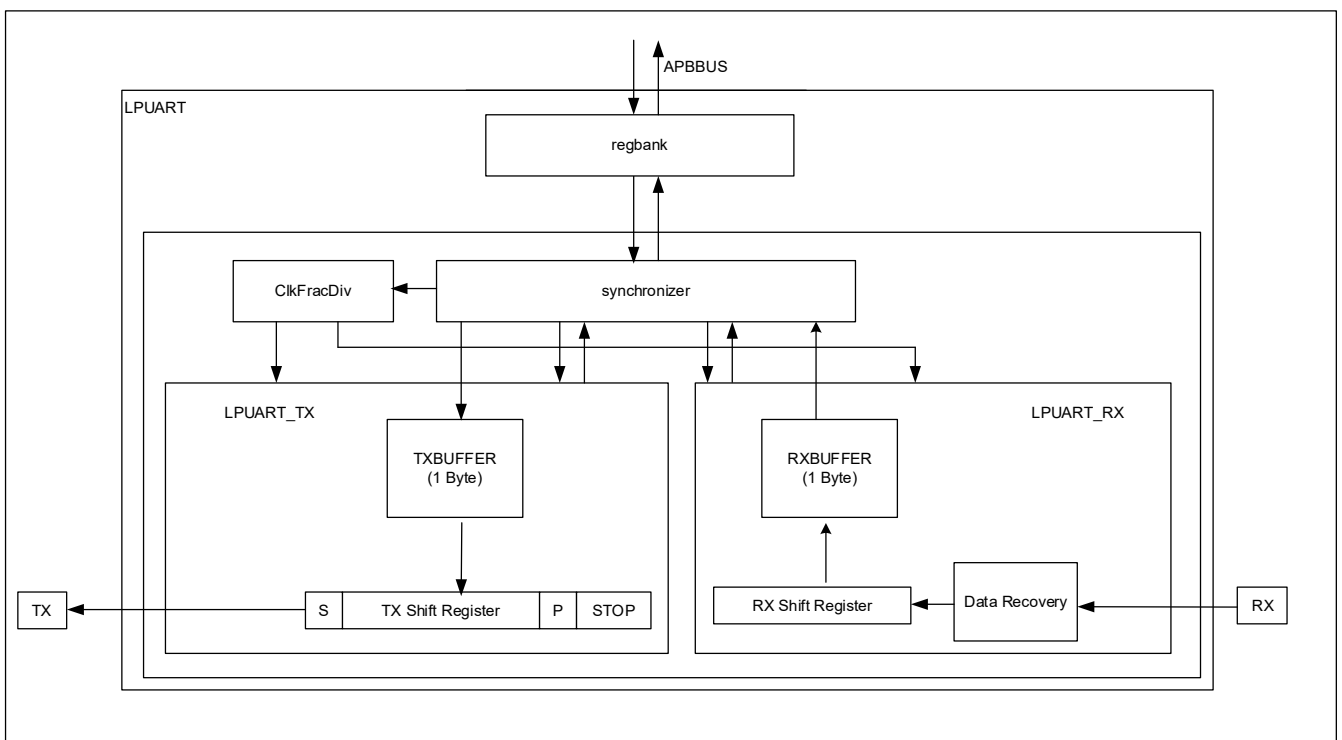


图 30-1 LPUART 功能框图

30.3 主要特征

- 支持 UART 帧格式的异步数据收发
- 全双工异步操作
- 支持输入任意频率时钟源，配置参考 RCC 模块 RCC_CFGR2 寄存器描述。
- 支持可编程的波特率数据传输，计算方法参考本章节表 3。
- 内置独立的 1 字节发送和 1 字节接收缓冲
- 发送和接收数据低位在前
- 一个起始位开始，后面接数据位，输出的数据长度可为 7 或 8 位，最后为停止位。另外可选

择是否有加奇偶校验位，奇偶校验位在数据位之后停止位之前。

- 支持硬件奇数或者偶数校验产生和侦测
- 支持信号接收和发送取反
- 支持下面中断源：
- 接收缓冲溢出
- 帧错误
- 奇偶校验错误
- 接收器检测到起始位
- 接收器检测到下降沿
- 接收器完整接收 1byte 数据
- 接收器完整接收数据且与预设数据匹配
- 发送器数据完成发送
- 发送器缓冲空
- 支持低功耗模式以下唤醒源：
- 接收器检测到下降沿唤醒
- 接收器检测到起始位唤醒
- 接收器 1 字节接收完成唤醒
- 接收器 1 字节数据接收并匹配唤醒

30.4 中断

表 30-1 LPUART 中断请求

中断事件	中断状态	使能
RX 检测到 START 位中断	LPUART_SR.START	LPUART_CR.RXIE & LPUART_CR.RXEV=00
RX 接收完成 1byte 数据中断	LPUART_IFR.RXIF	LPUART_CR.RXIE & LPUART_CR.RXEV=01
RX 接收数据匹配成功中断	LPUART_SR.MATCH	LPUART_CR.RXIE & LPUART_CR.RXEV=10
RX 检测到下降沿中断	LPUART_IFR.RXNEGIF	LPUART_CR.RXIE & LPUART_CR.RXEV=11
发送 buffer 空	LPUART_IFR.TXIF	LPUART_CR.TXIE
发送一帧数据完成	LPUART_IFR.TC_IF	LPUART_CR.TCIE
检验位错误中断	LPUART_SR.PERR	LPUART_CR.ERRIE
帧格式错误中断	LPUART_SR.FERR	LPUART_CR.ERRIE
接收 buffer 溢出中断	LPUART_SR.RXOV	LPUART_CR.ERRIE

如果设置了对应的中断使能控制位，这些设置就可以产生各自对应的中断。

30.5 功能描述

LPUART 为全双工通信，需要分配两个脚给 LPUART：接收数据输入（RX）和发送数据输出（TX）。

LPUART 数据帧符合 UART 协议，空闲帧与断开帧未定义。

空闲状态为总线在开始发送或者开始接收前的初始状态。

起始位为一位，用 0 表示。

一个数据帧(7 或 8 位)，发送和接收顺序从最低位到最高位。

停止位用 1 表示一帧的结束，位数可配置为 1、2 个。

可配置有无奇偶检验位，可配置奇校验或者偶校验。

通过使能 LPUART_CR 寄存器的 RXPOL/ TXPOL 位，可以将接收/发送端的信号取反。

30.5.1 特性描述

数据长度可以编程为 7 或 8 位，需要配置 LPUART_CR 寄存器中的 DL 位。发送器会在发送起始位时拉低 TX 引脚，在发送停止位时拉高 TX 引脚。

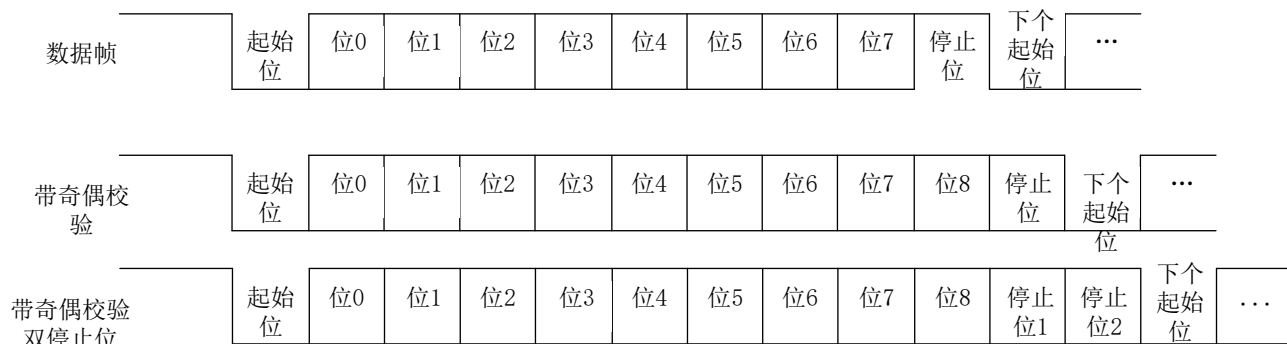


图 30-2 帧时序

30.5.2 接收时序

由于 LPUART 工作时钟不是波特率的整数倍，采用固定分频系数的话会引入累积误差，在接收时采用 3、4 分频交替进行接收，确保在每个 bit 的中间位置采样，每个 bit 采样一次。每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：

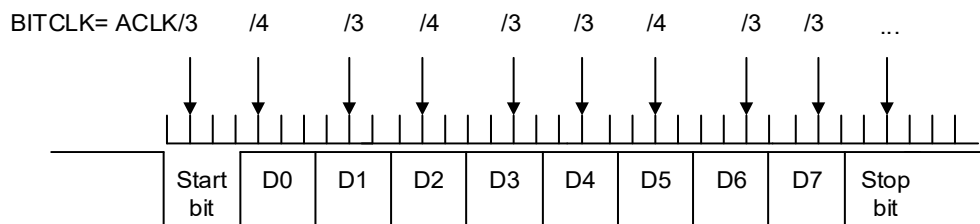


图 30-3 接收时序



30.5.3 发送时序

类似于 LPUART 接收，LPUART 工作时钟不是波特率的整数倍，采用固定分频系数同样会引入累积误差，在发送时也采用了 3、4 分频交替进行发送，每个 bit 采用 3 分频还是 4 分频，则由 MCTL 寄存器控制。例如：

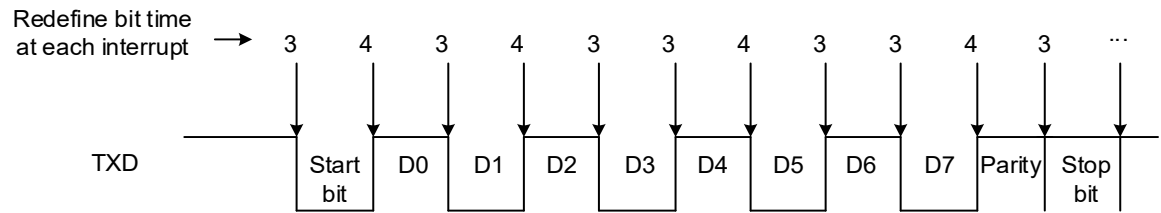


图 30-4 发送时序

30.5.4 调制寄存器建议配置

当 LPUART 工作时钟为准确的 32768Hz 时，软件可配置 BREN 为 0，然后需要根据通信波特率的不同合理配置调制控制寄存器 MCTL，建议的配置参数表如下：

表 30-2 调制控制寄存器 MCTL 建议配置参数表

Baud	MCTL											
	Bit0 (start)	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

当 LPUART 工作时钟不是 32768Hz 时，软件可配置 BREN 为 1，并根据工作时钟计算 BR 和 LPUART_MODU 的值，以时钟源 40Khz 和波特率 9600 为例计算方式如下：

表 30-3 调制控制寄存器 MCTL 计算方式

i	通信帧 bit	累计周期 S	单 bit 采样周期 N	BR	MCTL
1	START bit0	4	4	3	0
2	bit1	8	4		0
3	bit2	13	5		1
4	bit3	17	4		0
5	bit4	21	4		0
6	bit5	25	4		0
7	bit6	29	4		0
8	bit7	33	4		0
9	bit8	38	5		1
10	bit9	42	4		0



i	通信帧 bit	累计周期 S	单 bit 采样周期 N	BR	MCTL
11	bit10	46	4		0
12	bit11	50	4		0

$$S = \text{ROUND}(F_{\text{uclk}}/F_{\text{bps}}*i, 0);$$

$$N = S_i - S_{i-1};$$

$$BR = \text{MIN}(N) - 1;$$

$$\text{MCTL} = N - BR - 1;$$

30.5.5 奇偶校验控制

奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）可以通过设置 LPUART_CR 寄存器上的 PAREN 位而激活。如果奇偶校验出错，无效数据仍然会从移位寄存器传送到 LPUART_RXD 寄存器，LPUART_SR 的 PERR 位硬件自动置 1。LPUART_SR 的 PERR 位软件写“1”可清零该状态标志位。

偶校验：校验位加上数据总共的 1 的个数为偶数。

奇校验：校验位加上数据总共的 1 的个数为奇数。

待发送数据的 1 的个数是不可改变的，通过配置校验位为 1 或者 0，满足奇偶校验的要求。

例如：数据 = 1000_0110，有 3 个‘1’

偶校验：(LPUART_CR 中的 PTYP = 0)，校验位将是‘1’。

奇校验：(LPUART_CR 中的 PTYP = 1)，校验位将是‘0’。

在带奇偶校验的传输中，奇偶校验位的发送将跟在数据发送完最高有效位的后面。

30.5.6 发送器

发送 buffer 深度位 1 个字节。

DL 位的配置决定发送器发送数据的位数（7~8 位）。发送数据需要先置起发送使能位（TXEN），之后数据随着发送移位寄存器串行输出到 TX 引脚上。

30.5.6.1 字符发送

在 LPUART 发送期间，数据从 LPUART_TXD 寄存器写入，经过一字节缓冲器缓冲，最后通过发送移位寄存器，以最低字节到最高字节的顺序，串行在 TX 引脚上输出。

一位低电平的起始位会先于字符发送，而在字符发送完成的最后会发送高电平的停止位。停止位的位数可通过寄存器配置。

当前数据传输未完成前不能清零 TXEN 位，否则此时波特率发生器会停止产生时钟，发送器由于没有时钟驱动导致发送数据后部分丢失，数据不完整。

30.5.6.2 可配置的停止位

通过 SL 位进行编程，可以控制发送器每帧输出停止位的位数位 1 或者 2。

30.5.6.3 配置步骤

- 配置 LPUART_BAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUART_CR 寄存器，选择帧格式、极性、中断参数等
- 配置 LPUART_EN 寄存器打开发送使能
- 将待发送数据写入 LPUART_TXD 寄存器
- 等待中断事件
- 当发送完成时，LPUART_SR 的 TXE 标志位会被硬件置起，表示数据已传入移位寄存器，发送 buffer 为空。此时可往 LPUART_TXD 写入下一个数据。软件向发送 buffer 写数据时 TXE 标志位自动清零。

30.5.7 接收器

接收 buffer 深度位 1 个字节。

30.5.7.1 字符接收

对于 LPUART 的接收器，RX 串行输入数据，接收顺序从数据最低有效位到最高有效位。在接收器中，内部总线和接收移位寄存器之间插入了 LPUART_RXD 寄存器包含的缓冲器。

30.5.7.2 配置步骤

- 配置 LPUART_BAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUART_CR 寄存器，选择帧格式、极性、中断参数等
- 配置 LPUART_EN 寄存器打开接收使能和中断使能
- 等待相应中断事件

当接收一个完整帧时，LPUART_SR 的 RXF 标志位置起，表示已完整接收数据，此时软件可读取 LPUART_RXD 读出接收到的数据。软件读 LPUART_RXD 寄存器时，RXF 标志位自动清零。为了防止溢出错误，在下一字符接收结束前，RXF 位必须被清零。

在接收数据时，不应该软件清零 RXEN 位。如果在接收时清零 RXEN 位，还在 RX 引脚上的数据将不再被接收。

30.5.7.3 溢出错误

如果在 LPUART_RXD 前一个字符没有被读出前又接收到一个字符，则发生溢出错误。当溢出错误产生时：

- LPUART_SR 的 RXOV 位被置位。
- 新的数据将会刷新 LPUART_RXD 寄存器，上一个未读数据将会丢失。
- 若想产生中断，需配 LPUART_CR 的 ERRIE 置位。

30.5.7.4 帧错误

当停止位没有在预期的时间上接收和识别出来时检测到帧错误。当帧错误被检测到时：

- LPUART_SR 的 FERR 位被硬件置起。
- 新的数据将会刷新 LPUART_RXD 寄存器。
- 若想产生中断，需配 LPUART_CR 的 ERRIE 置位。

30.5.8 休眠模式下的数据接收唤醒

LPUART 支持在休眠模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

- 配置 LPUART_BAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUART_CR 寄存器，选择帧格式、极性，通过 LPUART_CR.RXEV 选择唤醒事件为 START 位、一帧接收完成、一帧数据匹配或 RXD 下降沿检测
- 配置 LPUART_EN 寄存器打开接收使能
- 软件进入休眠模式

30.5.9 低功耗模式下的数据自动收发

软件可以实现低功耗模式下一定数据量的 LPUART 自动收发，而无需 CPU 干预，低功耗模式参考 PMU 模块。

- 配置 LPUART_BAUD 寄存器决定波特率
- 根据波特率选择合适的调制参数，配置 MCTL 寄存器
- 配置 LPUART_CR 寄存器，选择帧格式、极性、中断参数等
- 将系统主时钟选为 LSCLK
- 软件进入 LPRUN
- 配置 LPUART_EN 寄存器打开发送接收使能

- 如 CPU 无额外工作，可以主动进入 WFI/WFE，等待中断唤醒

30.6 寄存器描述

30.6.1 寄存器概览

表 30-4 LPUART 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	LPUART_BAUD	波特率寄存器	0x00000000
0x04	LPUART_MODU	波特率调制控制寄存器	0x00000000
0x08	LPUART_IFR	中断标志寄存器	0x00000002
0x0C	LPUART_SR	状态寄存器	0x00000030
0x10	LPUART_CR	控制寄存器	0x00000000
0x14	LPUART_EN	发送接收使能寄存器	0x00000000
0x18	LPUART_RXD	接收数据寄存器	0x00000000
0x1C	LPUART_TXD	发送数据寄存器	0x00000000
0x20	LPUART_COMPARE	数据匹配寄存器	0x00000000
0x24	LPUART_WKCKE	唤醒寄存器	0x00000000

30.6.2 LPUART_BAUD 波特率寄存器

偏移地址：0x00

复位值：0x0000 0000

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

Bit	Field	Description
31:16	BR	波特率分频 时钟源到波特率的分频系数 BR+1 BR 不可配置为 0 和 1。
15:9	Reserved	保留，始终读为 0
8	BREN	波特率分频使能 0：波特率有 BAUD 寄存器控制，此时只支持 32.768KHz 的时钟作为时钟源 1：波特率有 BR 寄存器控制，可以支持 32.768KHz 以及其他频率的时钟作为时钟源

Bit	Field	Description
7:3	Reserved	保留，始终读为 0
2: 0	BAUD	波特率控制，仅在 BREN=0 时有效 000b : 9600bps 001b : 4800bps 010b : 2400bps 011b : 1200bps 100b : 600bps 101b / 110b / 111b :300bps 注意：当 BREN = 1 时，需要配置 BAUD ! = 0

30.6.3 LPUART_MODU 波特率调制控制寄存器

偏移地址：0x04

复位值：0x0000 0000

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

Bit	Field	Description
31:12	Reserved	保留，始终读为 0
11:0	MCTL	LPUART 每个 bit 的调制控制信号

30.6.4 LPUART_IFR 中断标志寄存器

偏移地址：0x08

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TC_IF	RXNEGI F	TXIF	RXIF
												r/w1c	r/w1c	r/w1c	r/w1c

Bit	Field	Description
31:4	Reserved	保留，始终读为 0
3	TC_IF	发送完成中断标志 1：一帧数据发送完后中断产生 0：无中断产生 该位写 1 清零
2	RXNEGIF	RXD 下降沿中断标志 1：RXD 下降沿中断产生 0：无中断产生 该位写 1 清零
1	TXIF	发送 buffer 空中断标志 1：发送 buffer 空后中断产生 0：无中断产生 该位写 1 清零
0	RXIF	接收完成中断标志 1：接收完一帧数据后中断产生 0：无中断产生 该位写 1 清零 注意：需要与 LPUART_WKCKE 寄存器配合使用

30.6.5 LPUART_SR 状态寄存器

偏移地址：0x0C

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								START	PERR	TC	TXE	RXF	MATCH	FERR	RXOV
								r/w1c	r/w1c	r	r	r	r/w1c	r/w1c	r/w1c

Bit	Field	Description
31:8	Reserved	保留，始终读为 0
7	START	起始位检测标志 1：检测到起始位 0：未检测到起始位 该位写 1 清零
6	PERR	校验位错误 1：检测到校验位错误 0：未检测到校验位错误 该位写 1 清零

Bit	Field	Description
5	TC	发送空闲状态标志 1: 当发送过程处于等待状态下且发送 buffer 为空 0: 数据发送过程正在进行或发送 buffer 非空 该位在发送 buffer 非空或数据发送开始时硬件自动清零
4	TXE	发送 buffer 空标志 1: 发送 buffer 为空 0: 发送 buffer 非空 该位硬件置位, 在软件向 LPUART_TXD 写数据时自动清零
3	RXF	接收缓冲满 1: 接收 buffer 为满 0: 接收 buffer 非满 该位在软件读 LPUART_RXD 寄存器时自动清零
2	MATCH	数据匹配标志 1: 接收缓冲区内的数据与数据匹配寄存器相同 0: 接收缓冲区内的数据与数据匹配寄存器不同 该位写 1 清零
1	FERR	帧格式错误 1: 检测到帧格式错误 0: 未检测到帧格式错误 该位写 1 清零
0	RXOV	接收缓冲溢出 1: 接收缓冲发生溢出错误 0: 接收缓冲未发生溢出错误 该位写 1 清零

30.6.6 LPUART_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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TXPOL	RXPOL	PAREN	PTYP	SL	DL	RXEV		ERRIE	TCIE	TXIE	NEDET	RXIE
			rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW

Bit	Field	Description
31:13	Reserved	保留，始终读为 0
12	TXPOL	数据发送极性 0: 非取反 1: 取反
11	RXPOL	接收极性 0: 非取反 1: 取反
10	PAREN	校验位使能 0: 数据帧无奇偶校验位 1: 数据帧有奇偶校验位
9	PTYP	校验位类型 0: 偶校验 1: 奇校验
8	SL	停止位长度 0: 1bit 1: 2bits
7	DL	数据长度 0: 8bits 1: 7bits
6:5	RXEV	接收中断事件配置 用于控制何种事件下向 CPU 提供接收中断 00: START 位检测唤醒 01: 1byte 数据接收完成 10: 接收数据匹配成功 11: 下降沿检测唤醒
4	ERRIE	错误中断使能 0: 禁止接收错误中断 1: 允许接收错误中断
3	TCIE	发送完成中断使能 0: 禁止发送完成中断 1: 允许发送完成中断
2	TXIE	发送 buffer 空中断使能 0: 禁止发送 buffer 空中断 1: 允许发送 buffer 空中断
1	NEDET	下降沿采样使能位 0: 不使用 32k 时钟下降沿检测 start bit 1: 使用 32k 时钟下降沿检测 start bit
0	RXIE	接收中断使能 0: 禁止接收中断 1: 允许接收中断

30.6.7 LPUART_EN 发送接收使能寄存器

偏移地址：0x14

复位值：0x0000 0000

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

Bit	Field	Description
31:2	Reserved	保留，始终读为 0
1	RXEN	接收使能 0：关闭 LPUART 接收 1：打开 LPUART 接收 软件写 1 使能后，要反复读取此寄存器，直到读到 1 为止才能进行后面的操作（因为有同步操作）
0	TXEN	发送使能 0：关闭 LPUART 发送 1：打开 LPUART 发送 软件写 1 使能后，要反复读取此寄存器，直到读到 1 为止才能进行后面的操作（因为有同步操作）

30.6.8 LPUART_RXD 接收数据寄存器

偏移地址：0x18

复位值：0x0000 0000

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

Bit	Field	Description
31:8	Reserved	保留，始终读为 0
7:0	DATA	接收数据寄存器 该寄存器只读

30.6.9 LPUART_TXD 发送数据寄存器

偏移地址：0x1C

复位值：0x0000 0000

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

Bit	Field	Description
31:8	Reserved	保留，始终读为 0
7:0	DATA	发送数据寄存器 该寄存器只写，读返回 0

30.6.10 LPUART_COMPARE 数据匹配寄存器

偏移地址：0x20

复位值：0x0000 0000

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

Bit	Field	Description
31:8	Reserved	保留，始终读为 0
7:0	COMPARE	比较数据 如果配置 LPUART_CR 的 RXEV=10，当接收缓冲区内的数据与 COMPARE 相同时，触发接收完成中断

30.6.11 LPUART_WKCKE 唤醒寄存器

偏移地址：0x24

复位值: 0x0000 0000

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

Bit	Field	Description
31:1	Reserved	保留，始终读为 0
0	LPUART_WKCKE	1 字节唤醒休眠模式控制位 当 1 字节接收完成或 1 字节数据匹配时，可将芯片从休眠模式唤醒； 1: 如果接收到的 1 字节数据有帧错误或奇偶校验错误时，不能唤醒，且对应标志位 RXIF 与 MATCH 不置起 0: 不管接收到的 1 字节数据是否存在错误，都可唤醒

31 SPI_I2S 串行外设接口

31.1 SPI_I2S 功能框图

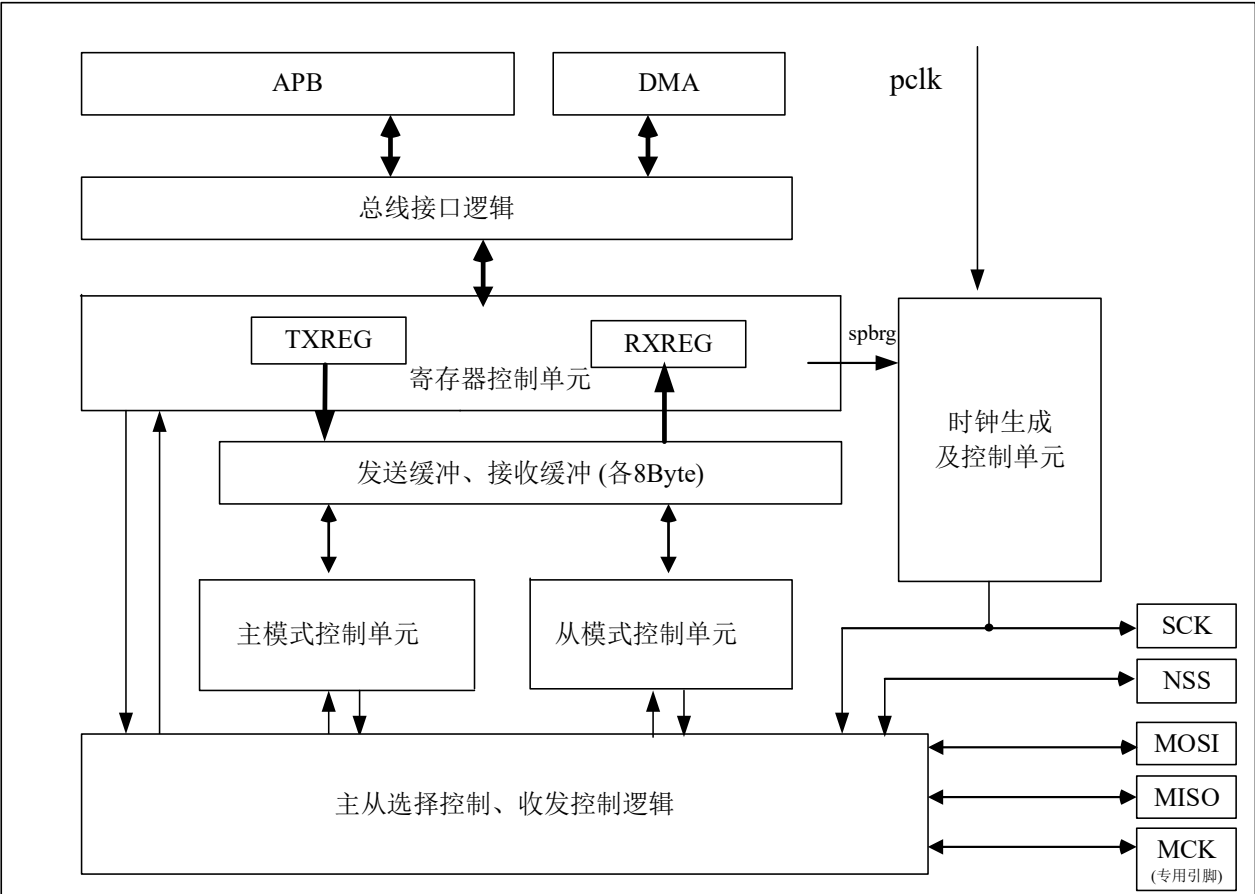


图 31-1 SPI_I2S 功能框图

31.2 SPI_I2S 简述

SPI (Serial Peripheral Interface) 接口广泛用于不同设备之间的板级通讯，如扩展串行 Flash，ADC 等。许多 IC 制造商生产的器件都支持 SPI 接口。

SPI 允许 MCU 与外部设备以全双工、同步、串行方式通信。应用软件可以通过查询状态或 SPI 中断来通信。

I2S (Inter—IC Sound) 总线，也称为集成电路内置音频总线，是飞利浦公司针对数字音频设备之间的音频数据传输而制定的一种总线标准。它使用独立的导线分别传输时钟与数据信号，因而避免了因时差诱发的失真，广泛应用于多媒体系统上各音频设备之间的数据传输。

31.3 SPI 功能描述

31.3.1 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：

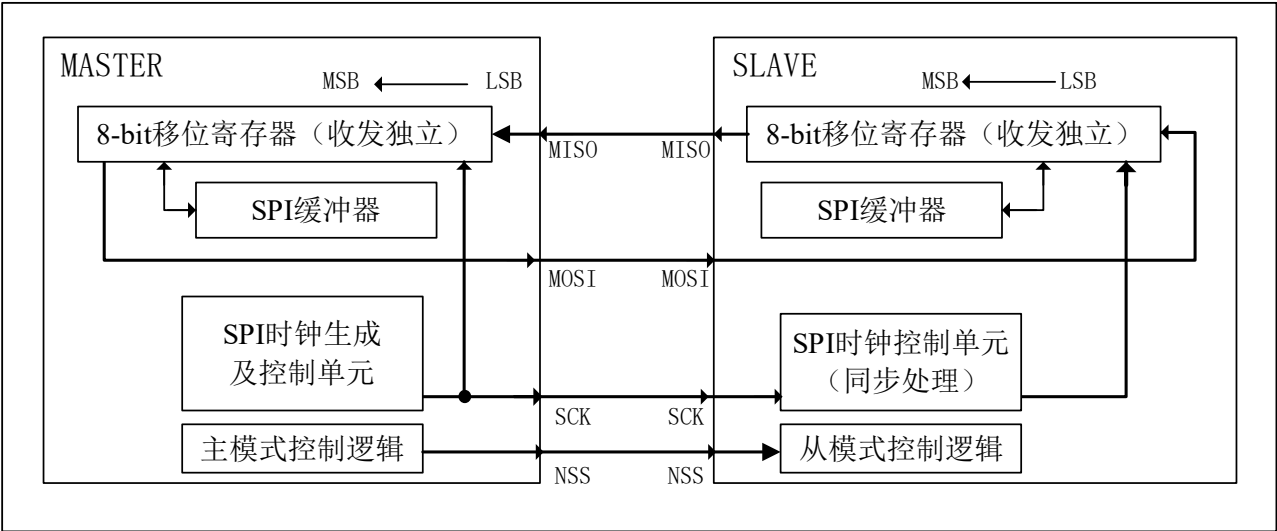


图 31-2 单主和单从应用

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。



31.3.1.1 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。

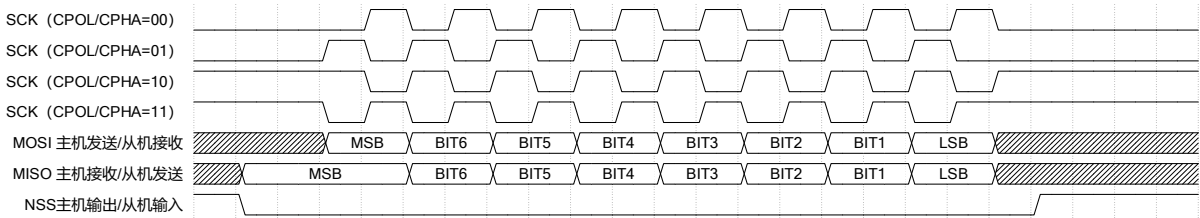


图 31-3 数据时钟时序图

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）。

31.3.1.2 高速传输

针对高速传输时板级延时敏感的情况，可配置 SPI_I2S_CCTL 寄存器中 TXEDGE/RXEDGE 位对数据发送/接收采样的时间点进行调整。高速传输，针对主设备的波特率发生器配置在 SPBRG<=4 且输出 SCK 时钟较快（>=10MHz）；低速传输，针对主设备波特率发生器配置在 SPBRG>4 且输出 SCK 时钟较慢（<10MHz）。

- 在从模式下，TXEDGE 位为‘1’时，软件一旦配置 TXREG 写入数据，不等待 SCK 时钟输入



的边沿而是立即发送到 MISO 引脚线上; TXEDGE 位为‘0’时, 从设备总是等到一个有效时钟边沿才发送数据到 MISO 引脚线上。

- 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。另外, 考虑到主机的时钟输出 SCK 经由 PCB 板布线到从机发送数据 MISO 至主机自身采样的最差延迟路径, 增加了通用控制寄存器 SPI_I2S_CCTL.MRDECHG[2:0]功能位来调整主机接收数据的采样时钟沿。
- ◆ 当 MRDECHG[2:0]配置为 3'b000 时, RXEDGE 位配置有效;
- ◆ 当 MRDECHG[2:0]不为 3'b000 时, RXEDGE 位配置无效; 此时, 主模式的数据采样点按照 MRDECHG[2:0]的配置, 以内部 APB 时钟为基准后移 1 到 4 个时钟周期。
- 下图示意了主机接收数据时, 采样点后移调整的时序 (mclk 为 APB 时钟, 以配置 SPI_I2S_SPBRG=2 时为例。此时 SCK 输出周期也为 APB 时钟的 2 倍)。

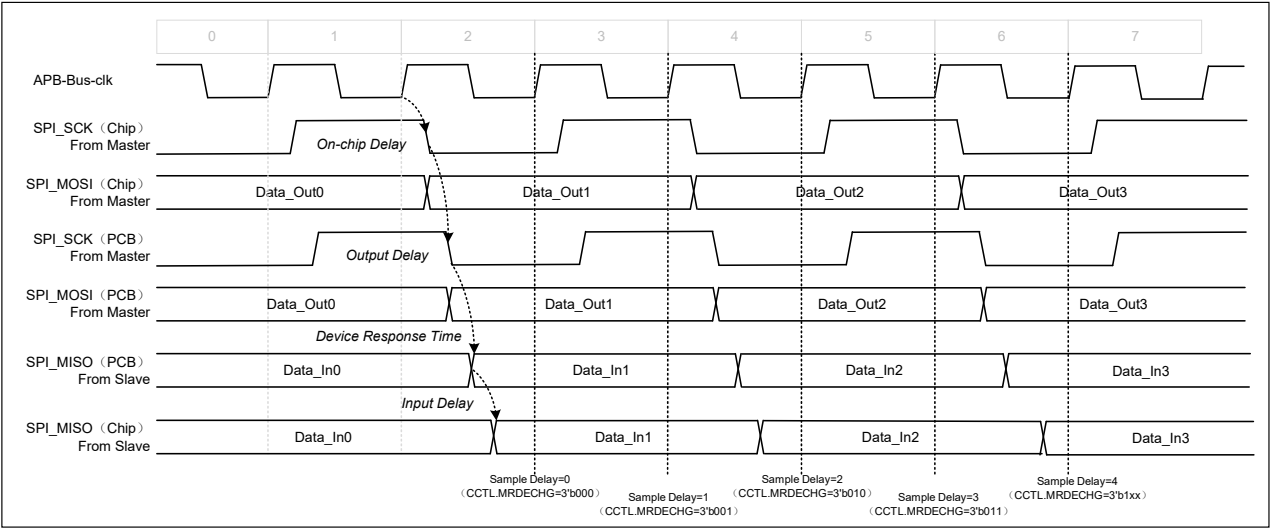


图 31-4 主机接收数据采样点后移调整示意图

31.3.1.3 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。



31.3.2 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）
- 支持 TI 模式传输（连续传输模式及非连续传输模式）
- 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

31.3.3 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）。

31.3.3.1 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. 根据传输速度的需求，高速模式下配置寄存器 `SPI_I2S_CCTL.TXEDGE` 位为‘1’。

6. 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。

31.3.3.2 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。

31.3.3.3 数据接收过程

从设备接收到 `MOSI` 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也会同时置位 `SPI_I2S_INTSTAT.RX_INTF` 标志。之后软件通过读 `SPI_I2S_RXREG`，就能从接收缓冲中获取该数据。
- 软件配置 `SPI_I2S_INTEN.RX_IEN` 位为‘1’来打开中断使能，利用 CPU 中断获取接收数据。

31.3.4 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。

31.3.4.1 配置步骤

1. 配置 `SPI_I2S_SPBREG` 寄存器，定义串行时钟波特率。
2. 配置 `SPI_I2S_CCTL` 寄存器中 `CPOL`、`CPHA/CPHASEL` 位，确定时序模式。
3. 配置 `SPI_I2S_CCTL.SPILEN` 来定义 8 或 7 位数据帧格式；配置 `SPI_I2S_GCTL.DW8_32` 为‘1’，可配置 `SPI_I2S_EXTCTL` 寄存器来定义成任意帧格式（`SPILEN` 需固定为‘1’）。
4. 配置 `SPI_I2S_CCTL.LSBFE` 来确定数据收发的顺序（`LSB` 或 `MSB` 位优先）。

5. 如果只接收而不发送数据，可配置 SPI_I2S_RXDNR 寄存器来定义需要接收的字节数（当接收到指定个数字节后，SCK 时钟输出会结束并保持在 CPOL 位配置的状态上）。

6. 配置寄存器 SPI_I2S_GCTL 中的 MODE 位为‘1’（主模式）、SPIEN 位为‘1’（SPI 功能使能），并配置 SPI 工作所需的 GPIO 功能引脚。

7. 根据传输速度的需求，高速模式下配置寄存器 SPI_I2S_CCTL.RXEDGE 位为‘1’或 MRDECHG[2:0] 位，来调整主机接收数据的采样时钟沿。

8. 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。

31.3.4.2 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）。

31.3.4.3 数据接收过程

主设备接收到 MISO 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也会同时置位 SPI_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）。

31.3.5 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 31-1 波特率公式

模式	公式
SPI 模式	波特率 = F_{pclk}/X

31.3.6 SPI 特殊功能（TI 模式）

SPI 接口支持 TI 协议下的数据通信。通过配置 SPI_CTL.TISEL 位为‘1’，使能 TI 模式。

TI 协议的时序中，当启动数据传输，首先从设备使能信号 NSS 置高并维持一个串口时钟（SCK）周期后，在串口时钟 SCK 的上升沿发送数据，并在其下降沿采样接收数据。

TI 模式将 NSS 作为一种特殊的帧头标志信号，因而 SPI_I2S_GCTL.NSS 位会被硬件固定为 1'b1，即数据传输时由硬件自动控制 NSS 输出。

由于时钟极性和相位都必须符合 TI 协议的要求，因而 SPI_I2S_CTL.CPHASEL 位会被硬件固定为 1'b1、CPOL/CPHA 位都会被硬件固定为 1'b0。

在 TI 模式下，数据帧初始化设置为 DW8_32=0 且 SPILEN=1，即 8bit 数据帧长格式；如果更改 SPILEN=0 则数据帧格式调整为 7bit 长度；也可通过配置 SPI_I2S_GCTL.DW8_32 位为‘1’来调整数据帧格式（限定在 4~32bit 范围内），注意此时需配置数据控制寄存器 SPI_I2S_EXTCTL（禁止设定 1~3bit 数据帧长）。

在 TI 协议的从机模式下，从机需要监视 NSS 信号；如果检测到错误的 NSS 信号（传输协议不匹配），将会置位 SPI_INISTAT.FRE_INTF 中断标志位。例如，NSS 信号在一帧数据的中间位发生了翻转。因此，在 TI 模式下软件可以通过帧传输错误（FRE_INTF）标志位来监控从机的状态。

需注意，FRE_INTF 中断标志位在 SPI 功能（非 TI 模式）时无效。

31.3.6.1 连续传输

SPI 在 TI 协议下，当主机写寄存器 TXREG 的速度较快，发送缓冲总不为空即可实现数据的连续传输。

连续传输情况下，在前一帧的 LSB 位置处片选 NSS 信号就会拉高并维持一个串口时钟 SCK 周期，紧接着就是后一帧数据 MSB 的开始。因而，只在连续传输的最开始时需要一个额外的时钟周期。

下面为 TI 模式下连续传输的时序图例（数据帧长度设定 8bit 为例）。

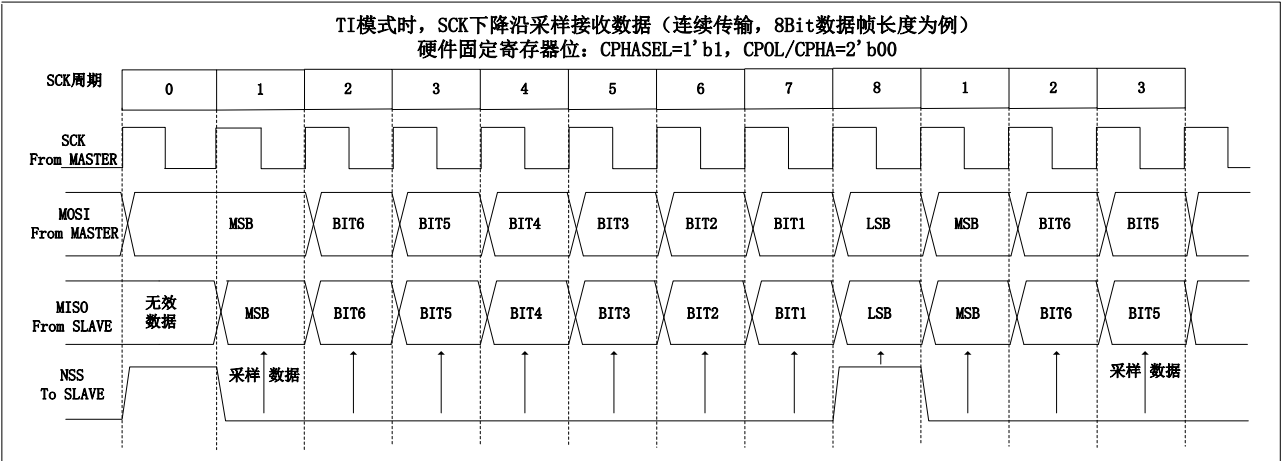


图 31-5 TI 模式连续传输时序示意图

从上图中可见，在通信刚开始帧头标志 **NSS** 的高电平周期，主模式下会提前把 **MSB** 位数据发送到引脚上；而从模式则先从发送缓冲加载好数据放置在移位寄存器中，等待 **NSS** 由高变低（即 **SCK** 的第二个上升沿到来时）立即送出 **MSB** 位数据到引脚上。

31.3.6.2 非连续传输

相比较连续传输而言，非连续传输情况下，在每一帧数据传输前都需要一个额外的时钟周期（用于片选 **NSS** 输出高电平，作为特殊的数据帧头的标志信号）。

当主机写 **TXREG** 的速度较慢，发送缓冲总出现空状态时，即工作在非连续传输。

下面为 **TI** 模式下非连续传输的时序图例（数据帧长度设定 **8bit** 为例）。

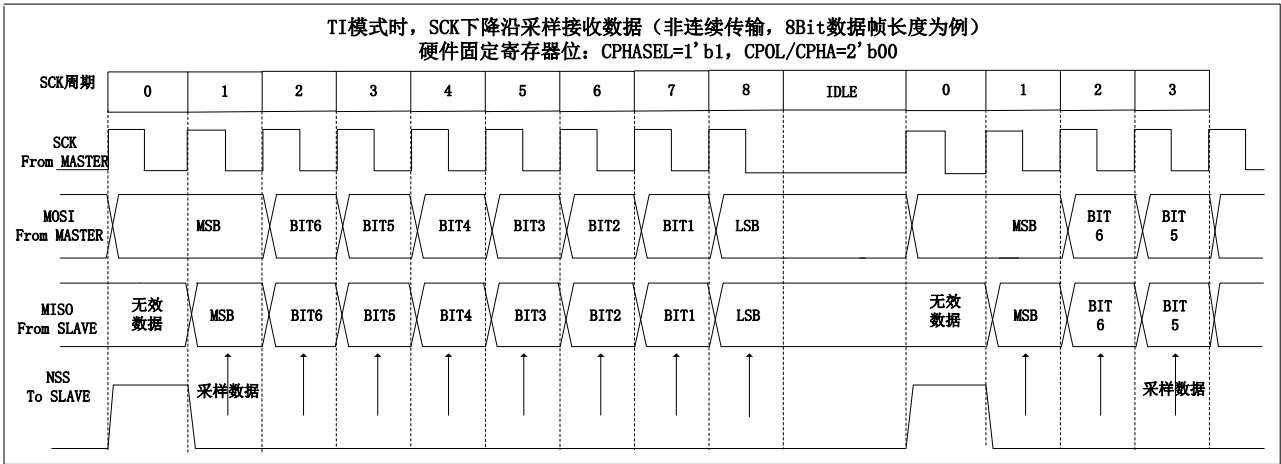


图 31-6 TI 模式非连续传输时序示意图

非连续传输情况下，**SCK** 周期为 **IDLE** 的时长由传输数据时发送缓冲空状态而决定，并不固定。在每个数据帧 **LSB** 位输出时，**NSS** 信号保持低电平；且 **LSB** 位的输出会保持到下一帧数据的 **MSB** 位输出前。

每帧数据的 **MSB** 位输出，与连续传输在通信刚开始时的情况一致。

31.3.7 中断

31.3.7.1 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 8 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 31-2 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
	FRE_INTF	帧传输错误(仅用于 SPI-TI 模式下的从机,监视 NSS 输入信号)
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,至少还能接收一个字节

31.3.8 DMA 传输

SPI 可以利用 DMA 来搬运数据,包含读取接收数据和写入发送数据。DMA 请求及应答机制,能监控收发缓冲的空满状态,并提高对 SPI 收发数据寄存器 RXREG、TXREG 的读写速率,从而加快 SPI 通信速度。

配置 SPI_I2S_GCTL.DMAMODE 位为‘1’,来实现 SPI 模块与 DMA 之间的信号交互。当发送缓冲有空闲空间,即请求 DMA 写入 TXREG;当接收缓冲有可读的有效数据,即请求 DMA 读取 RXREG。

- 发送时：当配置 DW8_32 位为‘0’，发送缓冲有大于等于 1 个空闲数据（1 字节）的空间，即发送缓冲未滿时就发起 DMA 请求；当配置 DW8_32 位为‘1’，发送缓冲有 1 个有效数据的空间空闲时，会发起 DMA 请求。每次请求只进行一次 DMA 传输，且传输数据字节数由 DW8_32 位决定。
- 接收时：当配置 DW8_32 位为‘0’，接收缓冲有大于等于 1 个有效数据（1 字节）时就发起 DMA 请求；当配置 DW8_32 位为‘1’，接收缓冲有 1 个有效数据后才发起 DMA 请求。每次请求只进行一次 DMA 传输。且传输数据字节数由 DW8_32 位决定。
- 注意：DW8_32=1 时，1 个有效数据的字节数由帧长位数决定（可参考 31.3.7.1 状态标志）。

31.4 I2S 功能描述

31.4.1 功能概述

I2S 与 SPI 区别在于，I2S 主模式的控制逻辑使用独立的时钟分频处理单元，可参考章节 31.4.8 时钟预分频器。

I2S 支持半双工通信，也支持全双工模式（通过配置 SPI_I2S_I2SCFGR.HDSEL 位来决定）。

在半双工模式下，可使用 MCK 引脚输出驱动时钟，用于驱动外部音频组件（通过配置 SPI_I2S_I2SCFGR.MCKSEL 位，选择驱动时钟映射在 MISO 引脚，或专用的 MCK 功能引脚）。

在全双工模式下，为了能同时进行数据的收发，MCK 驱动时钟只能从专用的 MCK 功能引脚输出。

31.4.2 I2S 主要特征

- 半双工通信（仅发送器或接收器）
- 支持全双工通信（SD，extSD 的数据方向根据主/从模式配置决定）
- 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）

31.4.3 I2S 总线接口

I2S 引脚功能如下：

表 31-3 I2S 引脚列表

信号名	类型	描述
I2Sn_CK	主模式输出，从模式输入	串口时钟（映射在 SCK 引脚）
I2Sn_WS	主模式输出，从模式输入	从设备选择（映射在 NSS 引脚）
I2Sn_MCK *注 1	主模式输出，从模式输入	可选的驱动时钟（由 MCKOE 配置输出许可；半双工时由 MCKSEL 配置可选择映射在 MISO 引脚或专用 MCK 引脚）
I2Sn_SD	全双工：主模式输出，从模式输入 半双工：主/从模式下的输入或输出	串行数据接口（映射在 MOSI 引脚上）
I2Sn_extSD	全双工：主模式输入，从模式输出 半双工：无此功能引脚	全双工串行数据接口（映射在 MISO 引脚）

注 1：专用 MCK 引脚不兼具任何 SPI 功能，作为 I2S 全双工时 MCK 输出的必选引脚。

31.4.4 数据格式

三线总线处理音频数据，必须经过分时复用两个声道：右声道和左声道。因为只有一个 32 位寄存器用于传输或接收，所以软件应依次配置寄存器 TXREG 为各声道的数据，或依次读取寄存器 RXREG 为各声道的数据。按照 I2S 协议，总是先发送左声道，然后发送右声道。

注意：在 PCM 协议下 SPI_I2S_CSTAT.CHSIDE 状态标志无效。

数据采用以下格式发送：

- 16 位数据打包在 16 位数据包帧
- 16 位有效数据打包在 32 位数据包帧（软件配置无效位为‘0’）
- 24 位有效数据打包在 32 位数据包帧（软件配置无效位为‘0’）
- 32 位数据打包在 32 位数据包帧

当使用 32 位帧上发送 16 位数据时，前 16 位（MSB）是有效的位，16 位 LSB 强制为 0，无需任何软件操作，通过硬件实现。其他格式相似。

31.4.5 通信标准

I2S 接口支持四种音频标准，通过配置寄存器 SPI_I2S_I2SCFGR 中的 I2SSTD[1:0]、PCMSYNC 位进行切换；数据格式则通过配置 DATLEN[1:0]、CHLEN 来进行选择。

对于所有通信标准及数据格式，总是先发送最高位（MSB 优先）。

31.4.5.1 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。

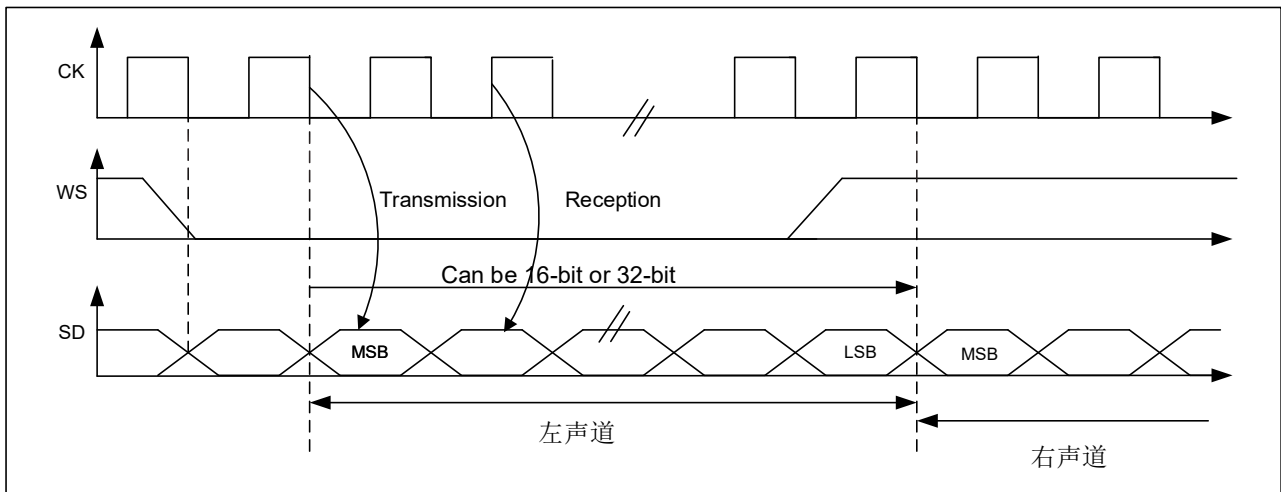


图 31-7 飞利浦标准示意图

31.4.5.2 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。

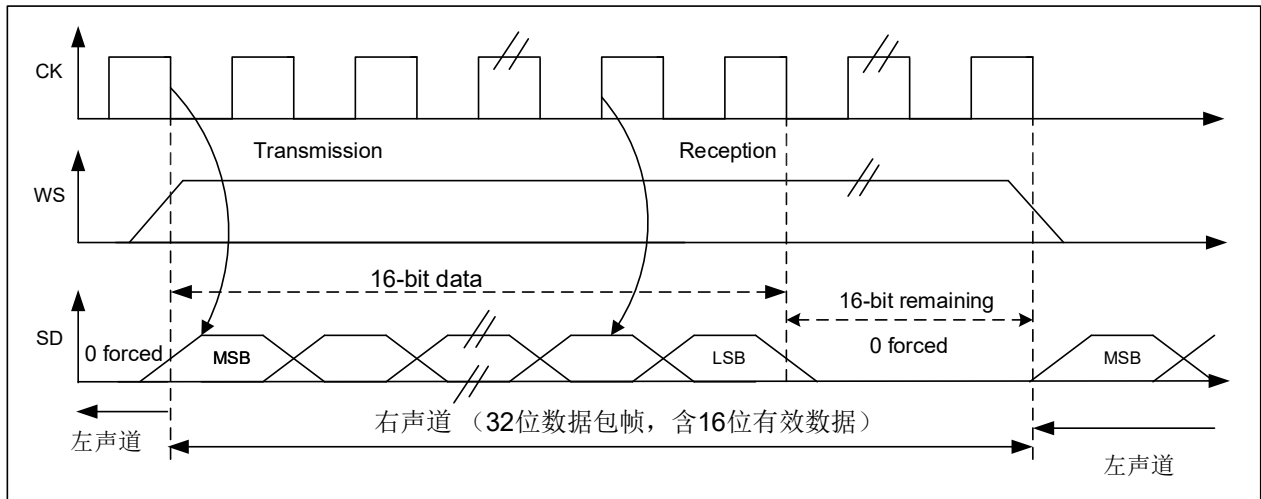


图 31-8 MSB 对齐标准示意图

31.4.5.3 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。

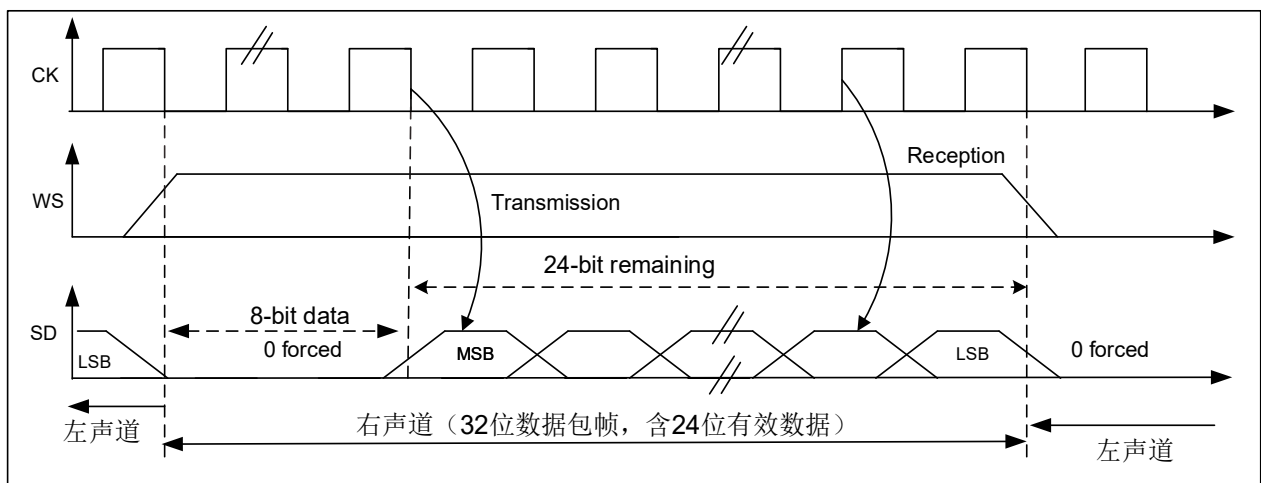


图 31-9 LSB 对齐标准示意图

31.4.5.4 PCM 标准

对于 PCM 标准，WS 引脚不作为声道信息使用（CHSIDE 标志位无效）。

PCM 标准有两个模式，分别为短帧模式和长帧模式；通过配置 SPI_I2S_I2SCFGR.PCMSYNC 位进行切换。在 PCM 模式下，输出信号（WS，SD）在 CK 时钟的上升沿锁存，输入信号（WS，SD）在 CK 时钟的下降沿采样。在主模式下应配置 CK 时钟和 WS 引脚为输出。

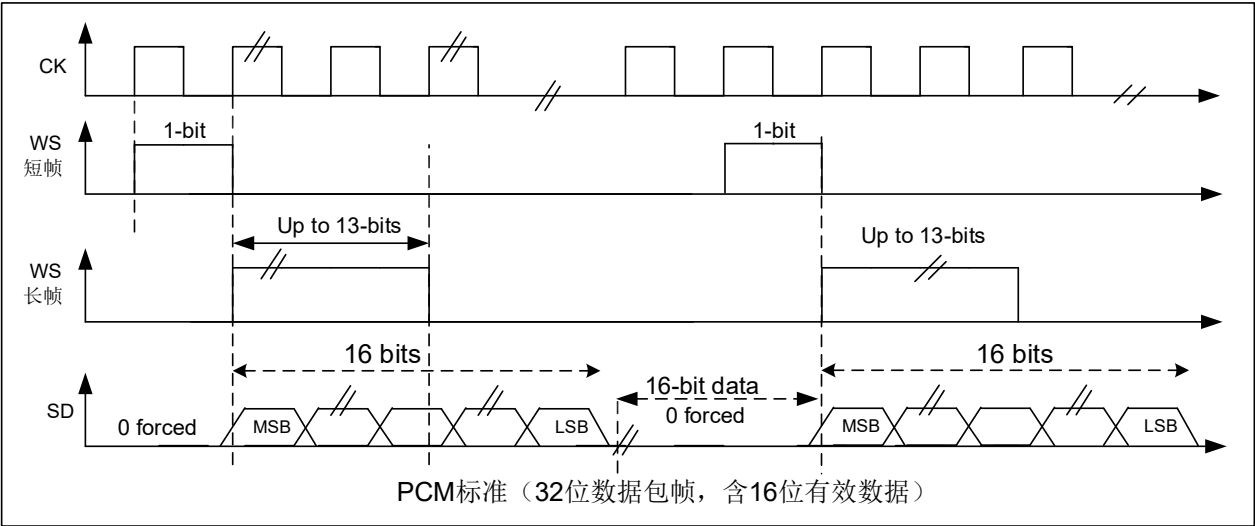


图 31-10 PCM 标准示意图

31.4.6 从模式

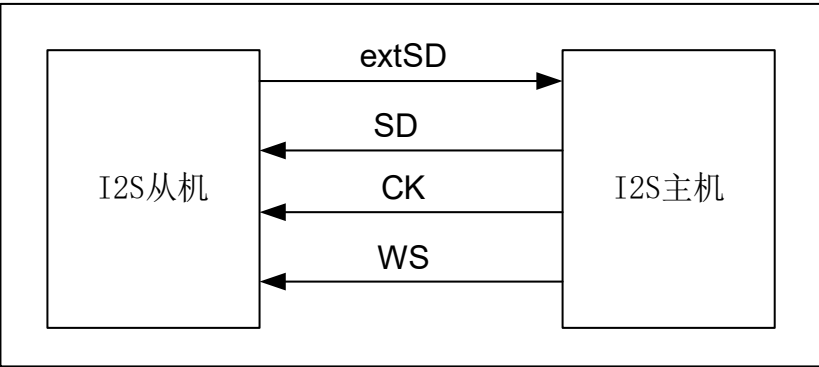


图 31-11 I2S 从模式下全双工通信（extSD 为输出，SD 为输入）

I2S 功能的从模式下，全双工工作时，SD 引脚作为从机接收接口，extSD 作为从机发送接口，SCK 和 WS 作为从机的输入信号；而当选择半双工工作时，SD 接口可作为从机的发送接口或接收接口。

从机内部收发状态、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 SPI_I2S_I2SCFGR.HDSEL，选择 I2S 工作在全双工或半双工。
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，计算方法请参考章节 31.4.8 时钟预分频器；
6. 配置 SPI_I2S_I2SCFGR.SPI_I2S 位为‘1’，使能 I2S 传输功能；
7. 配置寄存器 SPI_I2S_I2SCFGR 中的 I2SSTD[1:0]、PCMSYNC 位，选择 I2S 传输时使用的通信

标准；

- 8. 配置 SPI_I2S_GCTL.DMAMODE 位为‘1’，以启用 DMA 传输；
- 9. 开启半双工传输时，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’）。开启全双工传输时，配置 TXEN 和 RXEN 位都为‘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。

31.4.7 主模式

I2S 功能的主模式下，选择全双工工作时，SD 引脚作为主机发送接口，extSD 作为主机接收接口，SCK 和 WS 作为主机的输出信号；同时 MCK 向外部提供可选的驱动时钟（配置 SPI_I2S_I2SCFGR.MCKOE 位为‘1’使能 MCK 输出）。连接示意图如下：

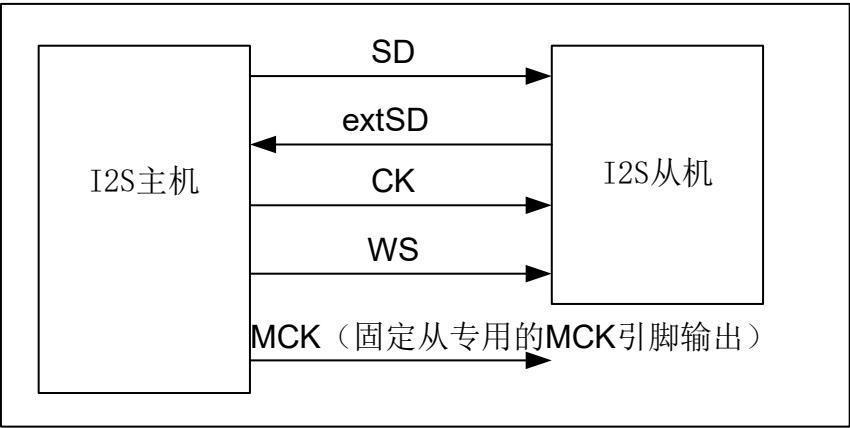


图 31-12 I2S 主模式全双工通信（SD 为输出，extSD 为输入）

I2S 功能的主模式下，选择半双工工作时，SD 引脚作为主机的发送接口或接收接口。此时 MCK 驱动时钟通过配置寄存器 SPI_I2S_I2SCFGR.MCKOE 位来选择是否输出；通过配置 SPI_I2S_I2SCFGR.MCKSEL 位来选择从 MCK 专用引脚或 extSD 引脚输出。连接示意图如下：

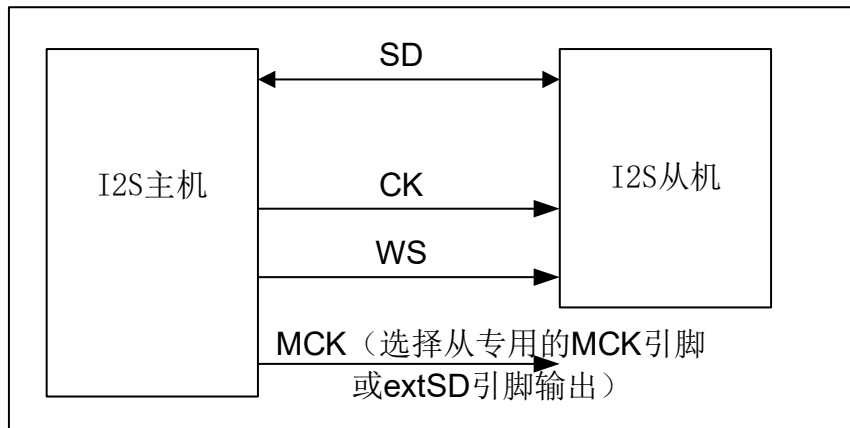


图 31-13 I2S 主模式半双工通信（SD 为输出或输入）

主模式的收发状态、实现方法类同 SPI 主模式，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.HDSEL，选择工作于全双工或半双工。
5. 配置 SPI_I2S_I2SCFGR.MCKOE 位，选择是否向外部器件提供 MCK 驱动时钟；如果选择半双工动作，则还需考虑选择 MCK 的输出引脚（配置 SPI_I2S_I2SCFGR.MCKSEL 位）。
6. 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，计算方法请参考章节 31.4.8 时钟预分频器；
7. 配置 SPI_I2S_I2SCFGR.SPI_I2S 位为‘1’，使能 I2S 传输功能；
8. 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；
9. 配置 SPI_I2S_GCTL.DMAMODE 位为‘1’，以启用 DMA 传输；
10. 开启半双工传输许可时，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’）。开启全双工传输时，配置 TXEN 和 RXEN 位都为‘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）。

31.4.8 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：

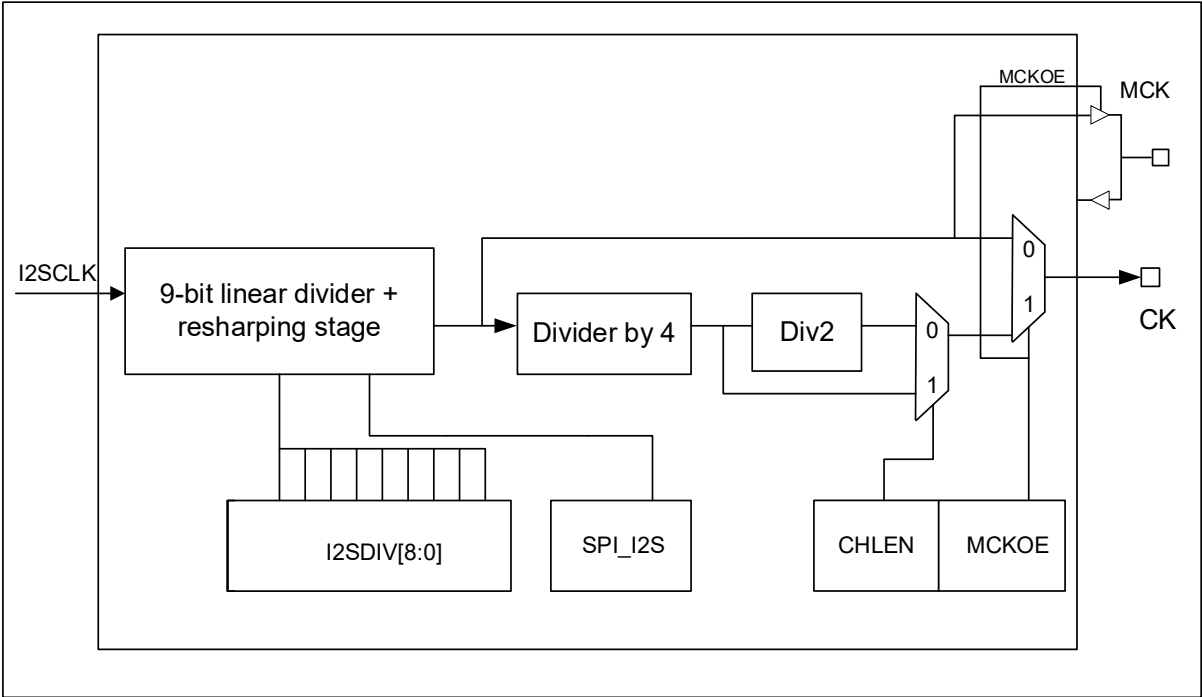


图 31-14 I2S 时钟预分频器示意图

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 31-4 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（Fs）和 I2S 比特率的关系由如下的公式定义：

$F_s = \text{I2S 比特率} / (\text{通道长度} \times \text{通道数}) = F_{CK} / (\text{通道长度} \times \text{通道数})$

注：通道长度，即数据包帧长度，可配置为 16 位或 32 位；通道数为左右声道，值固定为 2。

综上所述，根据 I2SDIV[8:0]、MCKOE 和 CHLEN 位的配置情况，得到音频采样率与 F_{I2SCLK}（APB 时钟频率）的关系如下：

表 31-5 音频采样率与 F_{I2SCLK}（APB 时钟频率）计算关系

MCKOE	CHLEN	音频采样率 F_s 与 F_{I2SCLK} 计算关系	注意事项
0	0	$F_{I2SCLK} = I2SDIV[8:0] \times 32 \times F_s$	$I2SDIV[8:0]$ 应配置在 2~511 范围内
0	1	$F_{I2SCLK} = I2SDIV[8:0] \times 64 \times F_s$	$I2SDIV[8:0]$ 应配置在 2~511 范围内
1	0	$F_{I2SCLK} = I2SDIV[8:0] \times 256 \times F_s$	$I2SDIV[8:0]$ 应配置在 1~511 范围内
1	1	$F_{I2SCLK} = I2SDIV[8:0] \times 256 \times F_s$	$I2SDIV[8:0]$ 应配置在 1~511 范围内

当 MCKOE 位为‘1’，芯片会输出 MCK 驱动时钟，且满足 $F_{MCLK} = 256 \times F_s$ ；此时假设 $F_s = 192\text{KHz}$ ， $I2SDIV[8:0] = 1$ ，则要求 APB 时钟 (F_{I2SCLK}) = $1 \times 256 \times 192\text{KHz} = 49.152\text{MHz}$ 。

针对常用的音频采样率，当按照 $F_{I2SCLK}=48\text{MHz}$ 计算时，误差如下：

表 31-6 I2S 音频采样率误差表

F_s	声道 位宽	MCKOE=1				MCKOE=0			
		I2SDIV 理想值	I2SDIV 配置值	真实 F_s	误差 (%)	I2SDIV 理想值	I2SDIV 配置值	真实 F_s	误差 (%)
192000	32	0.976563	1	187500	-2.34375	3.90625	4	187500	-2.34375
192000	16	0.976563	1	187500	-2.34375	7.8125	8	187500	-2.34375
96000	32	1.953125	2	93750	-2.34375	7.8125	8	93750	-2.34375
96000	16	1.953125	2	93750	-2.34375	15.625	16	93750	-2.34375
48000	32	3.90625	4	46875	-2.34375	15.625	16	46875	-2.34375
48000	16	3.90625	4	46875	-2.34375	31.25	31	48387.1	0.806452
44100	32	4.251701	4	46875	6.292517	17.0068	17	44117.65	0.040016
44100	16	4.251701	4	46875	6.292517	34.01361	34	44117.65	0.040016
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

31.4.9 中断

31.4.9.1 状态标志

I2S 与 SPI 共用同一个中断向量。

表 31-7 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

中断事件	中断寄存器标志
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。

31.4.10 DMA 传输

在 I2S 模式, DMA 的工作方式与 SPI 模式完全相同。

31.5 寄存器

31.5.1 寄存器总览

表 31-8 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

31.5.2 SPI_I2S_TXREG 发送数据寄存器

偏移地址：0x00

复位值：0x0000 0000

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

Bit	Field	Type	Reset	Description
31: 0	TXREG	rw	0x0000 0000	发送数据寄存器 (Transmit data register) 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，TXREG[31: 0]都有效

31.5.3 SPI_I2S_RXREG 接收数据寄存器

偏移地址：0x04

复位值：0x0000 0000

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

Bit	Field	Type	Reset	Description
31: 0	RXREG	r	0x0000 0000	接收数据寄存器 (Receive data register) 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，RXREG[31: 0]都有效 注：该寄存器可读不可写。

31.5.4 SPI_I2S_CSTAT 当前状态寄存器

偏移地址：0x08

复位值：0x0000 2001

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CHSIDE	BUSY	RXFADDR				TXFADDR				RXAVL_4BYTE	TXFULL	RXAVL	TXEPT
		r	r	r				r				r	r	r	r

Bit	Field	Type	Reset	Description
31: 14	Reserved			始终读为 0
13	CHSIDE	r	0x01	声道标志位 0: 表示正在传输的声道为左声道 1: 表示正在传输的声道为右声道 注: SPI 模式下不可用; 且 I2S 的 PCM 标准模式下无意义。
12	BUSY	r	0x00	忙标志位 表示 I2S 或 SPI 正在传输中
11: 8	RXFADDR	r	0x00	当前接收缓冲中有效字节个数
7: 4	TXFADDR	r	0x00	当前发送缓冲中有效字节个数
3	RXAVL_4BYTE	r	0x00	接收缓冲中有效数据达到 4 个字节标志 (Receive available 4 byte data message) 0: 接收缓冲中数据小于 4 个字节 1: 接收缓冲中有超过 4 个字节 注: 工作在 I2S 模式下时, 该位在接收到一个声道数据后置位 (如 CHLEN=0, 在接收到 16bit 数据后置位)。
2	TXFULL	r	0x00	发送缓冲满标志位 (Transmitter FIFO full status bit) 0: 发送缓冲未滿 1: 发送缓冲满
1	RXAVL	r	0x00	接收有效数据标志位 (Receive available byte data message) 当接收缓冲中接收到一个字节数据时置位该位。 0: 接收缓冲空 1: 接收缓冲非空 注: 该位只读, 由硬件自动置位和清除。
0	TXEPT	r	0x01	发送端空标志位 (Transmitter empty bit) 0: 发送缓冲或发送移位寄存器不为空 1: 发送缓冲和发送移位寄存器都为空 注: 该位只读, 由硬件自动置位和清除。

31.5.5 SPI_I2S_INTSTAT 中断状态寄存器

偏移地址: 0x0C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								FRE_IN TF	TXEPT_ INTF	RXFULL_ INTF	RXMAT CH_INT F	RXOER R_INTF	UNDER RUN_IN TF	RX_INT F	TX_INT F
								r	r	r	r	r	r	r	r

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留，始终读为 0
7	FRE_INTF	r	0x00	帧传输错误中断标志位： 表示 I2S/TI 模式下传输协议不匹配（仅从模式有效）
6	TXEPT_INTF	r	0x00	发送端空中断标志位（Transmitter empty interrupt flag bit） 硬件自动置位，写 INTCLR.TXEPT_ICLR 位为‘1’清除。 0：发送缓冲或发送移位寄存器不为空 1：发送缓冲和发送移位寄存器都为空 注：该位是中断状态信号，TXEPT 是状态信号。
5	RXFULL_INTF	r	0x00	接收缓冲满中断标志位（RX FIFO full interrupt flag bit） 硬件自动置位，写 INTCLR.RXFULL_ICLR 位为‘1’清除。 0：RX 缓冲未满 1：RX 缓冲满
4	RXMATCH_INTF	r	0x00	接收到指定字节数中断标志位（Receive data match the RXDNR number, the receive process will be completed and generate the interrupt） 硬件自动置位，写 INTCLR.RXMATCH_ICLR 位为‘1’清除。 0：未完成 RXDNR 寄存器指定的字节数 1：接收了 RXDNR 寄存器指定的字节数
3	RXOERR_INTF	r	0x00	接收端溢出错误中断标志位（Receive overrun error interrupt flag bit） 硬件自动置位，写 INTCLR.RXOERR_ICLR 位为‘1’清除。 0：没有溢出错误 1：溢出错误
2	UNDERRUN_INTF	r	0x00	SPI 从机模式下溢标志位（SPI underrun interrupt flag bit） 硬件自动置位，写 INTCLR.UNDERRUN_ICLR 位为‘1’清除。 0：没有下溢错误 1：下溢错误
1	RX_INTF	r	0x00	接收缓冲数据有效中断标志位（Receive data available interrupt flag bit） 硬件自动置位，写 INTCLR.RX_ICLR 位为‘1’清除。当接收缓冲接收了一个完整有效数据时置位。 0：接收缓冲空 1：接收缓冲接收到一个完整有效数据 注：有效数据包含的字节数，参考 31.3.7.1 状态标志。

Bit	Field	Type	Reset	Description
0	TX_INTF	r	0x00	发送缓冲空中断标志位（缓冲为空，可写 TXREG）（Transmit FIFO available interrupt flag bit） 硬件自动置位，发送缓冲不为空时自动清零。 0：发送缓冲不为空 1：发送缓冲为空

31.5.6 SPI_I2S_INTEN 中断使能寄存器

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								FRE_IEN	TXEPT_IEN	RXFULL_IEN	RXMAT_CH_IEN	RXOERR_IEN	UNDER_RUN_IEN	RX_IEN	TX_IEN
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留，始终读为 0
7	FRE_IEN	rw	0x00	帧传输错误中断使能（FRE_IEN）： 0：中断禁止 1：中断使能
6	TXEPT_IEN	rw	0x00	发送端空中断使能位（Transmit empty interrupt enable bit） 0：中断禁止 1：中断使能
5	RXFULL_IEN	rw	0x00	接收缓冲满中断使能位（Receive FIFO full interrupt enable bit） 0：中断禁止 1：中断使能
4	RXMATCH_IEN	rw	0x00	接收指定字节数中断使能位（Receive data complete interrupt enable bit） 0：中断禁止 1：中断使能
3	RXOERR_IEN	rw	0x00	接收端溢出错误中断使能位（Overrun error interrupt enable bit） 0：中断禁止 1：中断使能

Bit	Field	Type	Reset	Description
2	UNDERRUN_IEN	rw	0x00	SPI 从机模式下溢中断使能位（SPI 从机模式） （Transmitterunderrun interrupt enable bit（SPI slave mode only）） 0：中断禁止 1：中断使能
1	RX_IEN	rw	0x00	接收端数据中断使能位（Receive FIFO interrupt enable bit） 0：中断禁止 1：中断使能
0	TX_IEN	rw	0x00	发送缓冲空中断使能位（Transmit FIFO empty interrupt enable bit） 0：中断禁止 1：中断使能

31.5.7 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

Bit	Field	Type	Reset	Description
31: 8	Reserved			保留，始终读为 0
7	FRE_ICLR	w	0x00	帧传输错误中断清除 0：写 0 无意义 1：写 1 清除中断
6	TXEPT_ICLR	w	0x00	发送端空中断清除位（Transmitter empty interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
5	RXFULL_ICLR	w	0x00	接收缓冲满中断清除位（Receiver buffer full interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
4	RXMATCH_ICLR	w	0x00	接收指定字节数中断清除位（Receive completed interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断

Bit	Field	Type	Reset	Description
3	RXOERR_ICLR	w	0x00	接收端溢出错误中断清除位 (Overrun error interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
2	UNDERRUN_ICLR	w	0x00	SPI 从机模式下溢中断清除位 (SPI 从机模式) (Transmitter underrun interrupt clear bit (SPI slave mode only)) 0: 写 0 无意义 1: 写 1 清除中断
1	RX_ICLR	w	0x00	接收端数据中断清除位 (Receive interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断
0	TX_ICLR	w	0x00	发送缓冲空中断清除位 (Transmitter FIFO empty interrupt clear bit) 0: 写 0 无意义 1: 写 1 清除中断

31.5.8 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	DMAMODE	Res.					RXEN	TXEN	MODE	SPIEN
rw				rw	rw	rw						rw	rw	rw	rw

Bit	Field	Type	Reset	Description
31: 18	Reserved			保留, 始终读为 0

Bit	Field	Type	Reset	Description
17: 13	PAD_SEL	rw	0x00	PAD_SEL SCL MOSI NSS MISO
				0 PAD0 PAD1 PAD2 PAD3
				1 PAD0 PAD1 PAD3 PAD2
				2 PAD0 PAD2 PAD1 PAD3
				3 PAD0 PAD2 PAD3 PAD1
				4 PAD0 PAD3 PAD1 PAD2
				5 PAD0 PAD3 PAD2 PAD1
				6 PAD1 PAD0 PAD2 PAD3
				7 PAD1 PAD0 PAD3 PAD2
				8 PAD1 PAD2 PAD0 PAD3
				9 PAD1 PAD2 PAD3 PAD0
				10 PAD1 PAD3 PAD0 PAD2
				11 PAD1 PAD3 PAD2 PAD0
				12 PAD2 PAD0 PAD1 PAD3
				13 PAD2 PAD0 PAD3 PAD1
				14 PAD2 PAD1 PAD0 PAD3
				15 PAD2 PAD1 PAD3 PAD0
				16 PAD2 PAD3 PAD0 PAD1
				17 PAD2 PAD3 PAD1 PAD0
				18 PAD3 PAD0 PAD1 PAD2
				19 PAD3 PAD0 PAD2 PAD1
				20 PAD3 PAD1 PAD0 PAD2
				21 PAD3 PAD1 PAD2 PAD0
				22 PAD3 PAD2 PAD0 PAD1
				23 PAD3 PAD2 PAD1 PAD0
PAD0, PAD1, PAD2, PAD3 分别对应引脚分布中的 SCK、MOSI、NSS、MISO 引脚。信号 SCK、MOSI、NSS、MISO 和 PAD 的映射变换如下：				
12	Reserved			预留，必须保持复位值。
11	DW8_32	rw	0x00	发送和接收数据寄存器有效数据选择（Valid byte or double-word data select signal） 0：只有低 8 位有效 1：32 位数据都有效 注：I2S 模式下固定为 1。

Bit	Field	Type	Reset	Description
10	NSS	rw	0x00	硬件或软件控制主模式下的 NSS 输出（NSS select signal that from software or hardware） 0：由 NSSR 寄存器值控制 1：进行数据传输时硬件自动控制 注：I2S/TI 模式下固定为 1。
9	DMAMODE	rw	0x00	DMA 方式（DMA Mode selection bit） 0：正常模式 1：开启 DMA 模式
8: 5	Reserved			预留，必须保持复位值。
4	RXEN	rw	0x00	接收使能位（Receive enable bit） 0：接收禁止。同时可以清空 RX 缓冲 1：接收使能 注：当 SPI 只工作在主机接收模式时，TXEN 必须设置为 0。
3	TXEN	rw	0x00	发送使能位（Transmit enable bit） 0：发送禁止。同时可以清空 TX 缓冲 1：发送使能 注：当在主机模式下发送和接收同时发生。
2	MODE	rw	0x01	主机模式位（Master mode bit） 0：从机模式（串行时钟来自外部主机） 1：主机模式（由内部 BRG 产生串行时钟）
1	INTEN	rw	0x00	SPI/I2S 中断使能位（SPI/I2S interrupt enable bit） 1：使能 SPI/I2S 中断 0：禁止 SPI/I2S 中断
0	SPIEN	rw	0x00	SPI/I2S 选择位（SPI/I2S select bit） 0：SPI/I2S 禁止（复位状态） 1：SPI/I2S 使能

31.5.9 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.					MRDECHG			TISEL	CPH	TX	RX	SPI LEN	LSB FE	CPOL	CPHA
					rw			rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Type	Reset	Description
31: 11	Reserved			保留，始终读为 0

Bit	Field	Type	Reset	Description
10:8	MRDECHG	rw	0x00	Master 接收采样时钟沿调整功能位 (Receive data edge change) 3'b000: 若 RXEDGE=0, 采样点不做调整; 若 RXEDGE=1, 采样点后移 SPBRG/2 个 mclk 周期。 3'b001: 采样点后移 1 个 mclk 周期 3'b010: 采样点后移 2 个 mclk 周期 3'b011: 采样点后移 3 个 mclk 周期 3'b1xx: 采样点后移 4 个 mclk 周期 注: mclk 为 SPI 模块使用的 APB 时钟, SPBRG 代表波特率控制寄存器的设定值; 当 MRDECHG 设定值不为 3'b000 时, RXEDGE 的设定即失效。在高速通信需求的场合, 可根据板级布线延迟, 调试系统以决定该配置, 使 SPI 主模式接收时能抵消传输延迟从而提升传输速度。
7	TISEL	rw	0x00	TI 模式使能选择 (TI mode select) 0: TI 模式无效 1: 使能 TI 模式, SPI 将按照 TI 模式进行数据通信 注: i2s 模式下此位固定为 0 (I2S 设定比 TI 模式设定优先级更高)
6	CPHASEL	rw	0x00	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/TI 模式下固定为 1。
5	TXEDGE	rw	0x00	发送数据相位调整位 (从模式) (Transmit data edge select) 0: 发送数据在一个有效时钟边沿后发送到数据总线 可用于低速模式时 (从模式输入时钟较慢)。 1: 发送数据立即发送到数据总线 可用于高速传输时 (从模式输入时钟较快, 超过 10MHz) 注: 建议该位配置为 1, 以免通信速率较快时不满足 AC 特性, 导致从模式下发送数据不符合主机采样时序要求。
4	RXEDGE	rw	0x00	接收数据采样时钟沿选择位 (主模式) (Receive data edge select) 0: 在传输数据位的中间采样数据 1: 在传输数据位的尾时钟沿采样数据 (用于高速传输, 后移采样时间点以应对板级布线及从模式发送数据的延迟) 注: 建议该位配置为 1, 以免通信速率较快时不满足 AC 特性, 导致主模式下接收数据不正确。
3	SPILEN	rw	0x01	SPI 数据宽度位 (SPI character length bit) 该位在 DW8_32=0 时有效; DW8_32=1 时, 该位需保持为 1。 0: 7 位数据 1: 8 位数据 (缺省) 注: I2S 模式下固定为 1。

Bit	Field	Type	Reset	Description
2	LSBFE	rw	0x00	LSBFE: LSB 在前使能位 (LSI first enable bit) 0: 数据传输或接收时最高位在前 1: 数据传输或接收时最低位在前 注: I2S/TI 模式下固定为 0。
1	CPOL	rw	0x00	时钟极性标志位 (Clock polarity select bit) 0: 时钟在空闲状态为低电平 (两次传输之间) 1: 时钟在空闲状态为高电平 (两次传输之间) 注: TI 模式下固定为 0。
0	CPHA	rw	0x00	时钟相位选择位 (Clock phase select bit) 0: 第一个数据位采样从第二个时钟边沿开始 1: 第一个数据位采样从第一个时钟边沿开始 注: I2S/TI 模式下固定为 0。

31.5.10 SPI_I2S_SPBRG 波特率发生器

偏移地址: 0x20

复位值: 0x0000 0002

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

Bit	Field	Type	Reset	Description
31: 16	Reserved			保留, 始终读为 0
15: 0	SPBRG	rw	0x0002	SPI 波特率控制寄存器用于产生波特率 (SPI baud rate control register for baud rate) 波特率公式: 波特率 = $F_{pclk} / SPBRG$ (F_{pclk} 是 APB 时钟频率) 注: 不能对该寄存器写值为 0、1。

31.5.11 SPI_I2S_RXDNR 接收数据个数寄存器

偏移地址: 0x24

复位值: 0x0000 0001

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

RXDNR

rw

Bit	Field	Type	Reset	Description
31: 16	Reserved			保留，始终读为 0
15: 0	RXDNR	rw	0x0001	该寄存器用于存储下次接收过程需要接收字节的个数（The register is used to hold a count of to be received bytes in next receive process） 注：该寄存器的值仅在 SPI 为主机接收模式下有效；缺省值是 1。该寄存器值通过软件来改变，且不能对该寄存器写值为 0。

31.5.12 SPI_I2S_NSSR 从机片选寄存器

偏移地址：0x28

复位值：0x0000 00FF

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

Bit	Field	Type	Reset	Description
31: 1	Reserved			保留，必须保持复位值。
0	NSS	rw	0x1	主模式下片选输出信号。低有效，从模式下该位无效（Chip select output signal in Master mode）。 0：从器件被选中（允许从器件与主模式建立通信） 1：从器件未选中

31.5.13 SPI_I2S_EXTCTL 数据控制寄存器

偏移地址：0x2C

复位值：0x0000 0008

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

Bit	Field	Type	Reset	Description
31: 5	Reserved			保留，始终读为 0
4: 0	EXTLEN	rw	0x08	控制 SPI 数据帧长度的选择 0 0000: 32 bit 0 0001: 1 bit 0 0010: 2 bit 0 0011: 3 bit 1 1100: 28 bit 1 1101: 29 bit 1 1110: 30 bit 1 1111: 31 bit 注：仅当 SPI_I2S_GCTL.DW8_32=1 时有效；当 DW8_32=0 时， 必须保持初始值 5'h8)。I2S 模式下此配置为无效： I2S 模式下，当 CHLEN=1, EXTLEN 值固定为 5'b00000(32 位)； I2S 模式下，当 CHLEN=0, EXTLEN 值固定为 5'b10000(16 位)。

31.5.14 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.		HDSEL	MCKSE	MCKOE	SPI_I2S	Res.			PCMSY	I2SSTD		Res.	DATLEN		CHLEN
			L						NC						
		rw	rw	rw	rw				rw	rw			rw		rw

Bit	Field	Type	Reset	Description
31: 25	Reserved			保留，始终读为 0
24: 16	I2SDIV	rw	0x1	I2S 预分频器的分频系数 注：主模式下才有效，且不可配置该位值为 0；当主模式下不 输出 MCK 驱动时钟，即 MCKOE=0 时，不可配置该位值为 1 （不支持 1 分频）。
15: 14	Reserved			保留，始终读为 0
13	HDSEL	rw	0x0	I2S 全双工、半双工选择 0: 全双工模式 1: 半双工模式

Bit	Field	Type	Reset	Description
12	MCKSEL	rw	0x0	MCK 输出信号选择 0: 从 MISO 引脚输出（与原半双工兼容） 1: 从 MCK 专用引脚输出（仅半双工时设定有效） 注：该位在半双工时才有效；全双工时该位设定无意义（MCK 必须从专用引脚输出）。
11	MCKOE	rw	0x0	主时钟输出使能 0: 主模式驱动时钟 MCK 输出禁止 1: 主模式驱动 MCK 输出使能
10	SPI_I2S	rw	0x0	模块功能选择 0: 启用 SPI 功能 1: 启用 I2S 功能 注：当配置该位为 1，即 I2S 模式下，以下寄存器功能位会被硬件固定，无需软件配置。 GCTL 寄存器中的 DW8_32、NSS 位； CCTL 寄存器中的 CPHASEL、SPILEN、LSBFE、CPHA 位； EXTCTL 寄存器中的 EXTLEN 位。
9: 7	Reserved			保留，始终读为 0
6	PCMSYNC	rw	0x0	PCM 标准帧同步模式 0: 短帧同步模式 1: 长帧同步模式
5: 4	I2SSTD	rw	0x0	I2S 标准 00: PHILIPS 标准 01: MSB 对齐标准 10: LSB 对齐标准 11: PCM 标准
3	Reserved			保留，始终读为 0
2: 1	DATLEN	rw	0x0	数据长度 00: 16 位宽 01: 24 位宽 10: 32 位宽 11: 禁止设置该值 注：当 CHLEN = 0 时，DATLEN 值由硬件固定为 2'b00。
0	CHLEN	rw	0x0	声道长度（每个声道包含的数据位数） 0: 16 位宽 1: 32 位宽

32 I2C 内部集成电路接口

32.1 介绍

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 有三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）、超快速模式（数据传输速率最大为 1Mbps）。

I2C 还支持 SMBus 和 PMBus 协议，用于系统管理和电源管理。

32.2 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）、超快速模式（最大 1Mbps）
- 分别有 4 字节的发送和接收缓冲
- 尖峰滤波时间可编程
- SDA 保持时间可编程
- 兼容 SMBus 2.0 和 PMBus 1.2
- 支持 DMA 操作
- 支持中断和查询操作
- 支持多个从地址（详见 I2C_SLVMASK 寄存器描述）

32.3 功能描述

32.3.1 功能框图

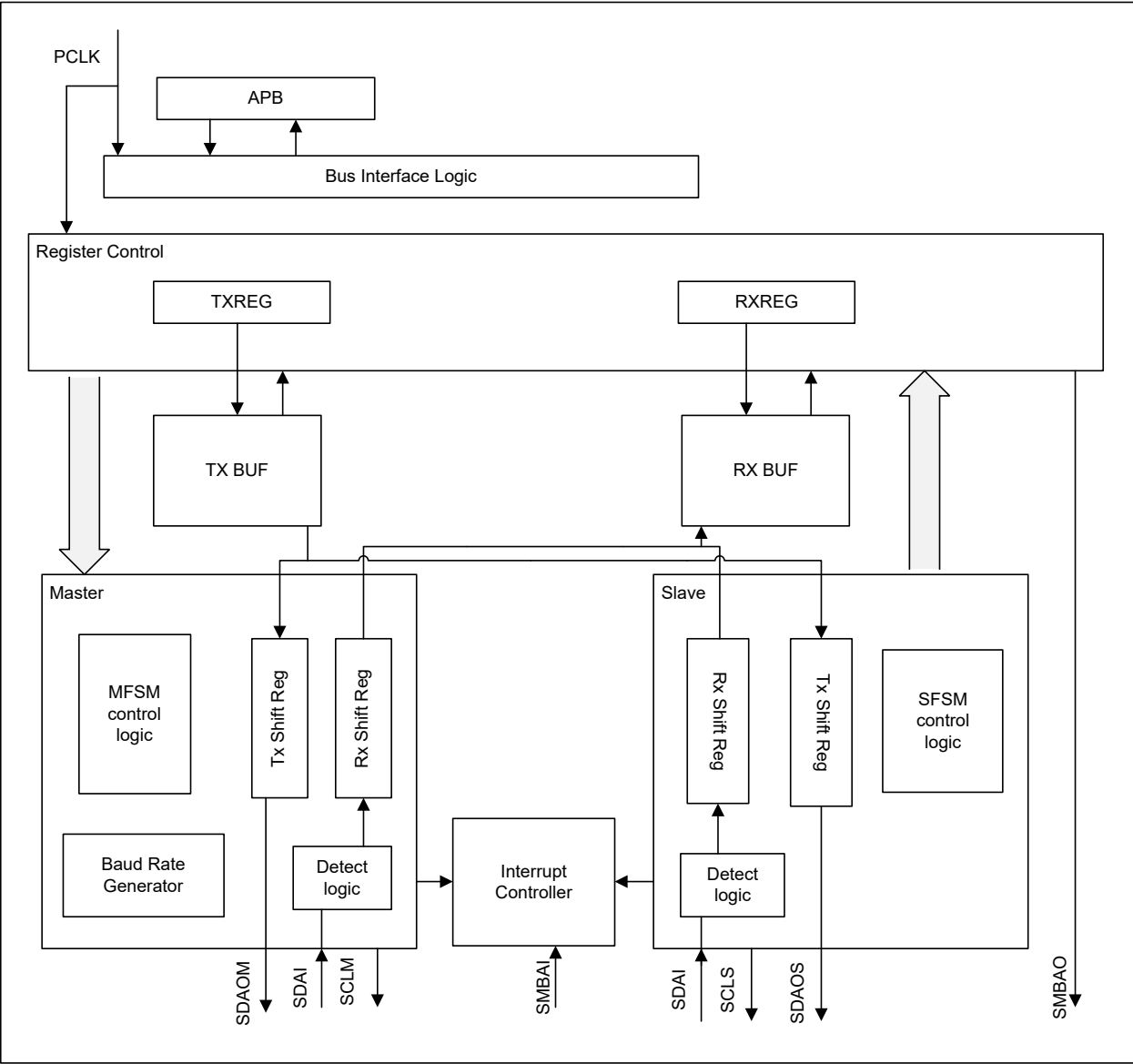


图 32-1 I2C 功能框图

32.3.2 信号描述

表 32-1 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
SMBA	I/O	SMBus 报警

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。

32.3.3 I2C 协议

32.3.3.1 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。

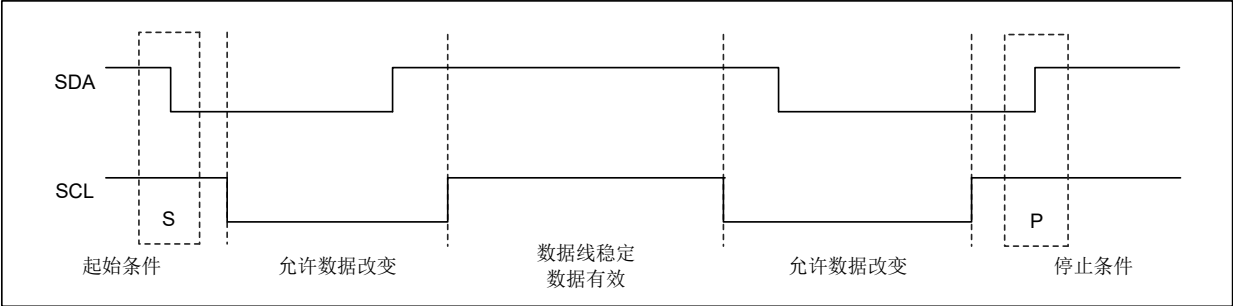


图 32-2 起始和停止条件

32.3.3.2 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。

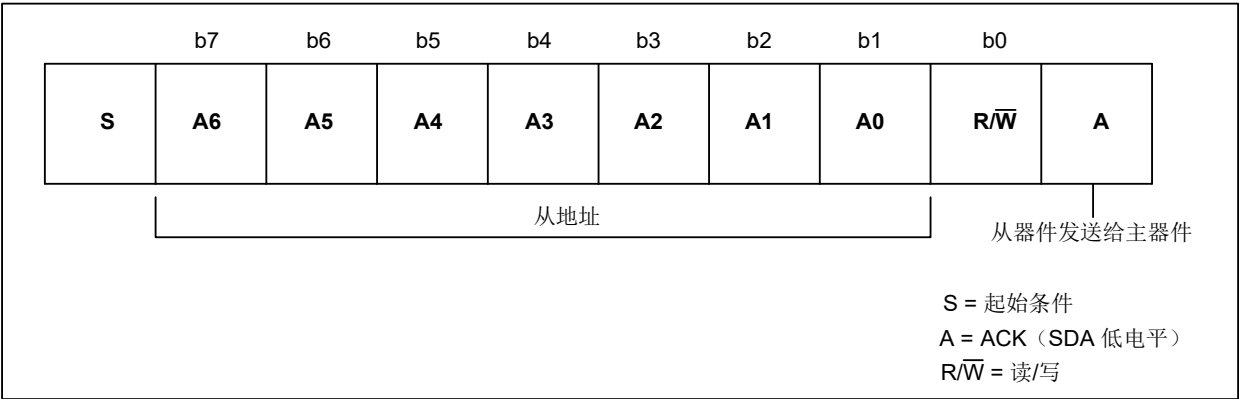


图 32-3 7 位地址格式

● 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：

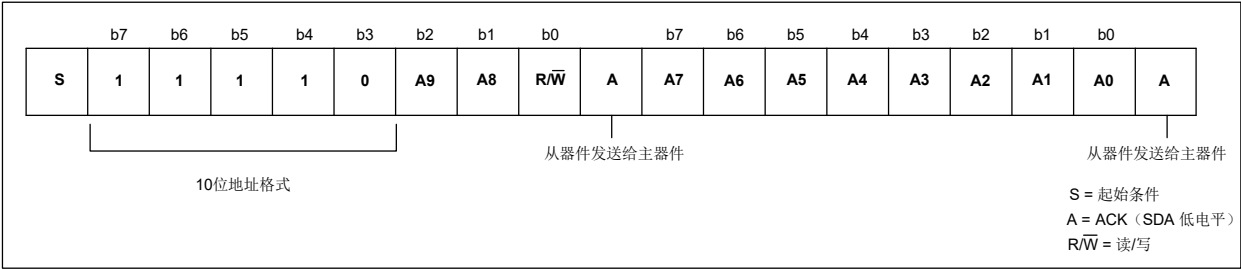


图 32-4 10 位地址格式

下表定义了 I2C 首字节的特殊用途和保留地址。

表 32-2 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
0001 000	x	SMBus 主机地址
0001 100	x	SMBus 报警响应地址
1100 001	x	SMBus 器件默认地址

32.3.3.3 发送和接收协议

主器件可以发起数据传输，作为主发送器或主接收器，向总线发送数据或从总线接收数据。从器件响应主器件的请求，充当从发送器或从接收器。

● 主发送和从接收

所有数据都以字节格式传输，对每次传输的字节数没有限制。主器件发送完地址和 R/W 位或者一个字节的的数据到从器件后，从接收器必须产生一个响应信号（ACK）。从接收器不能产生响应信号（ACK）时，主器件将会产生一个停止条件中止传输。从器件不能响应时，必须释放 SDA 为高电平，以便主器件产生停止条件。

当主发送器如下图所示传输数据时，从接收器在接收到的每个字节后产生一个 ACK 来响应主发送器。

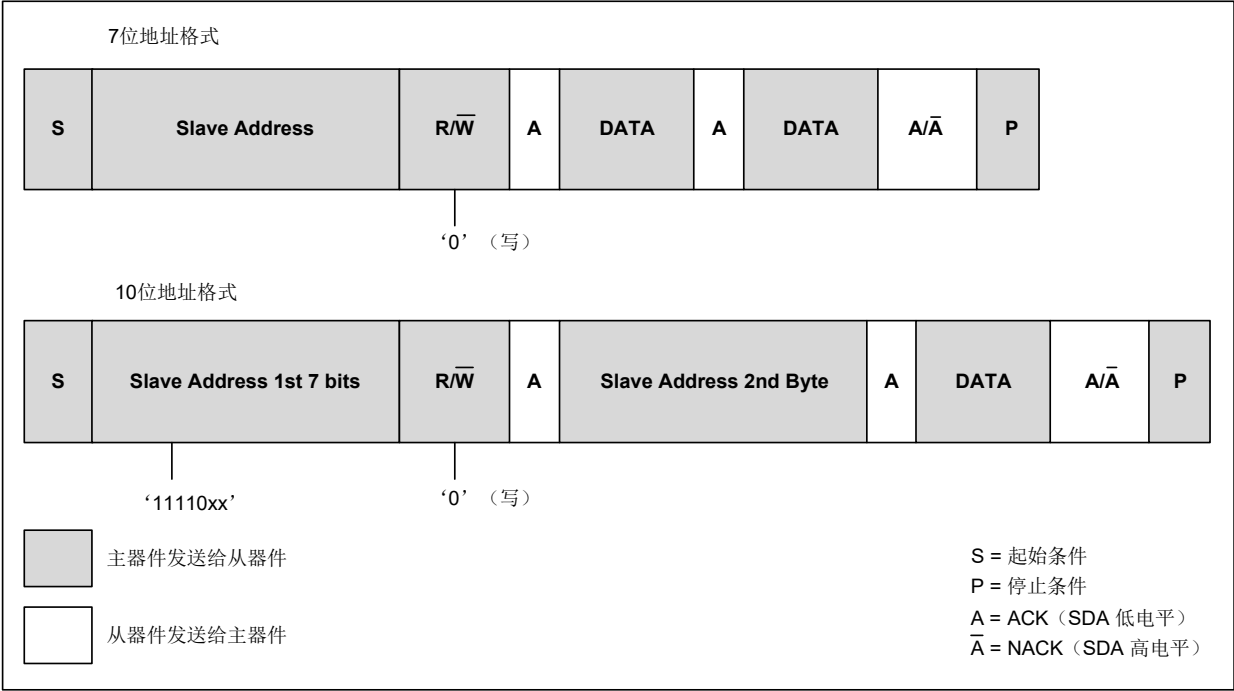


图 32-5 主发送协议

● 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。

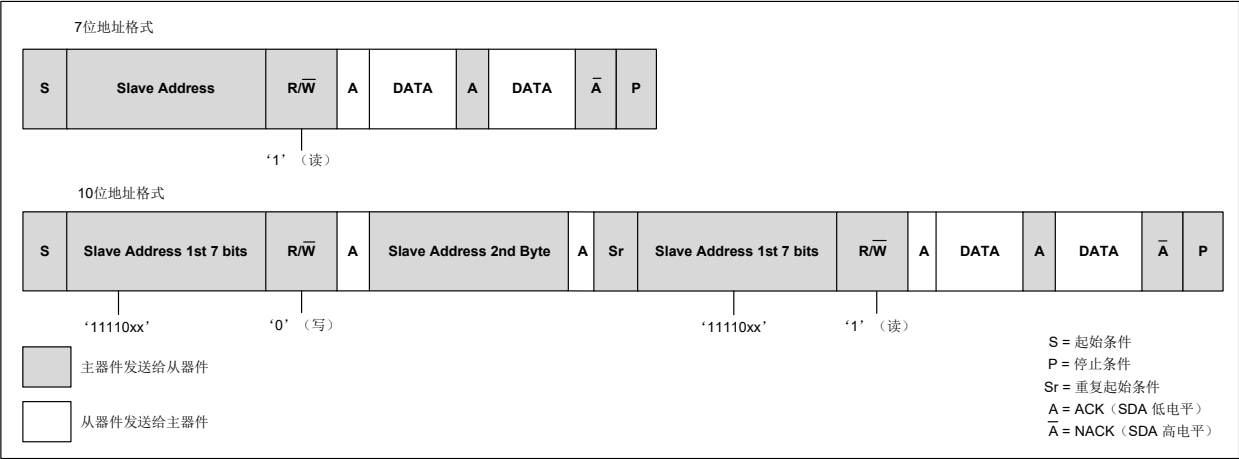


图 32-6 主接收协议

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。

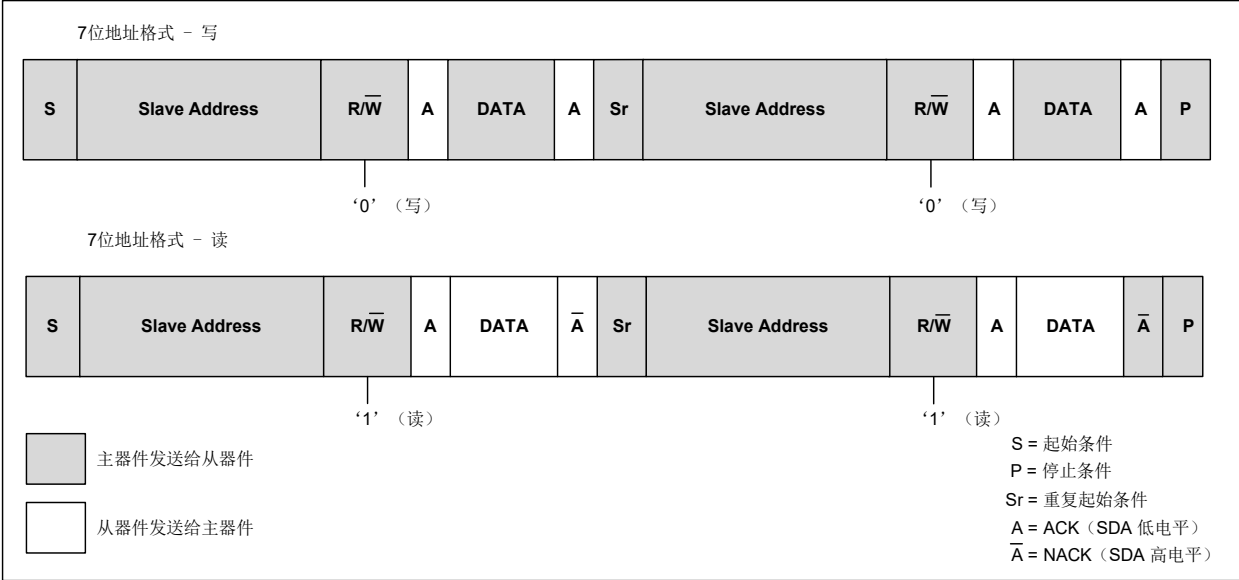


图 32-7 带 RESTART (SR) 信号的主发送和接收协议

● 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。

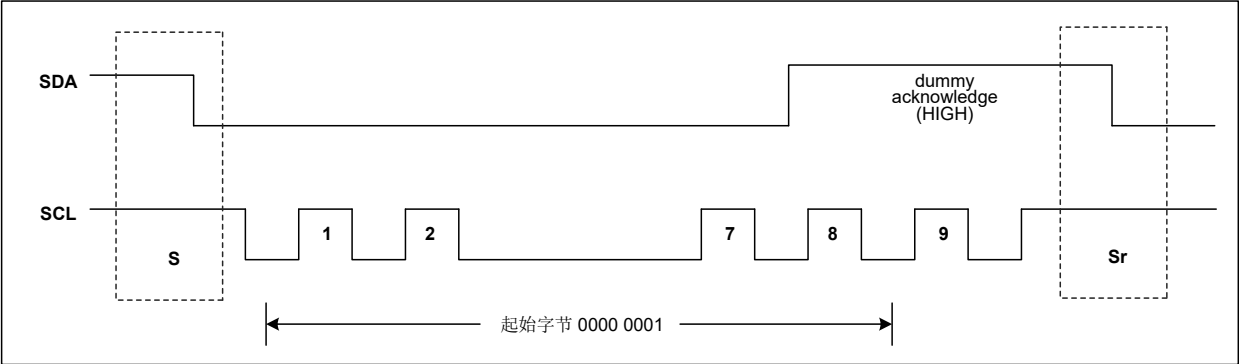


图 32-8 起始字节传输

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。

32.3.3.4 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。

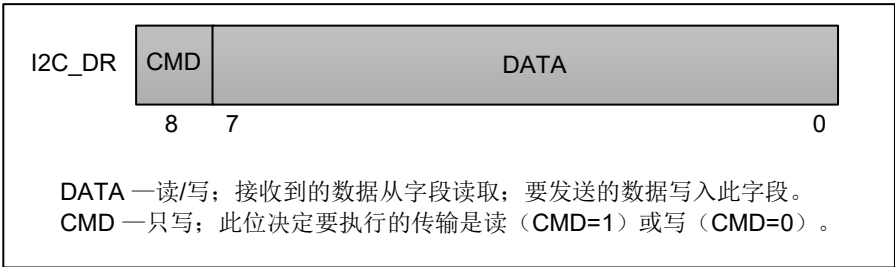


图 32-9 I2C_DR 寄存器

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。

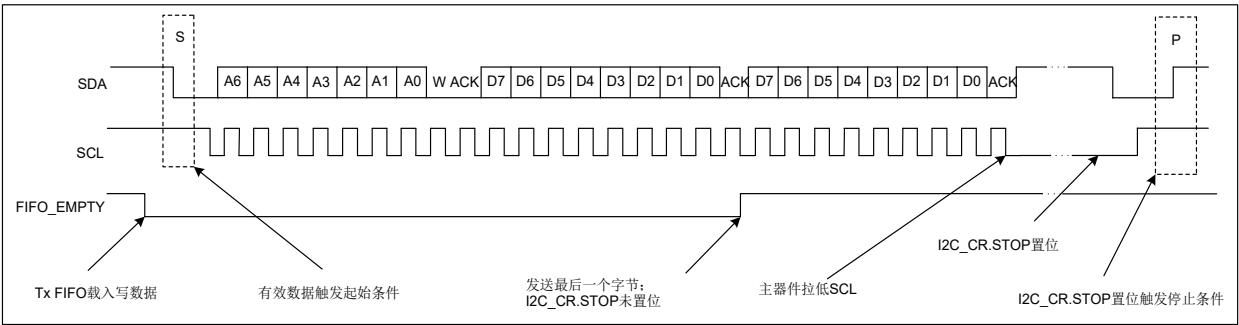


图 32-10 主发送，TX FIFO 为空

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。

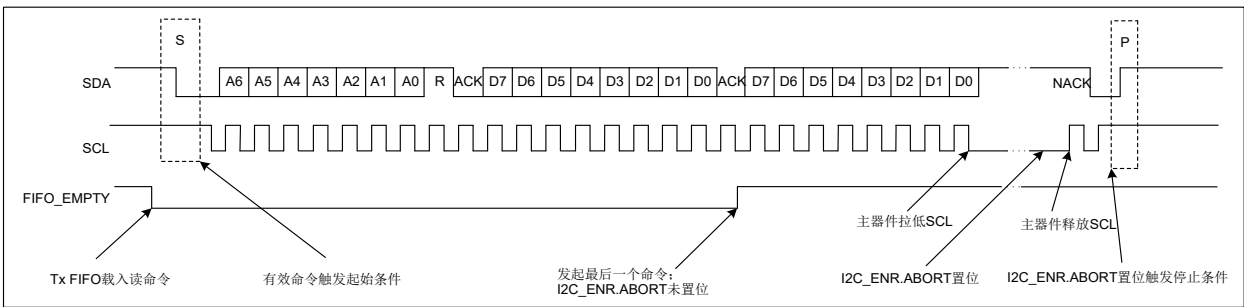


图 32-11 主接收，TX FIFO 为空

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。

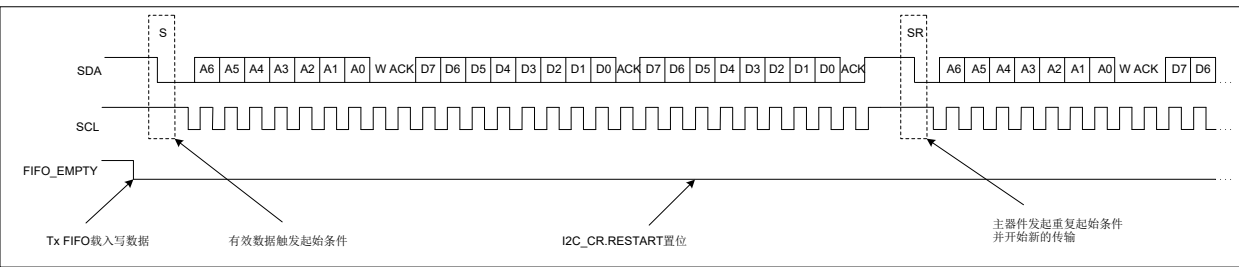


图 32-12 主发送，发生 RESTART

下图描述了 I2C 模块工作在主接收模式下，发生 RESTART 时的时序。

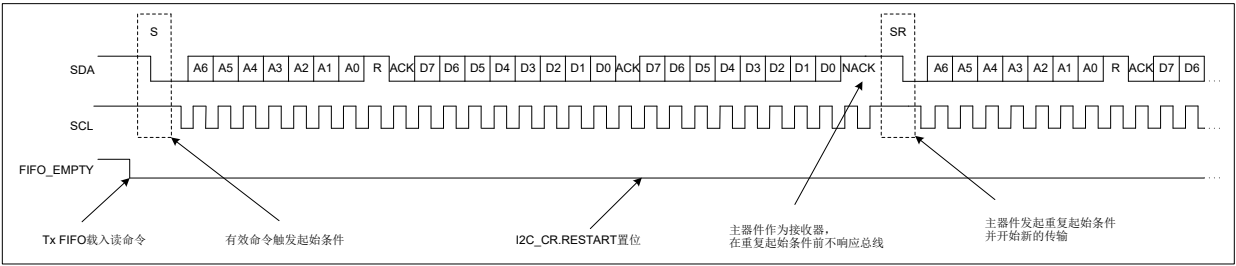


图 32-13 主接收，发生 RESTART

32.3.3.5 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。

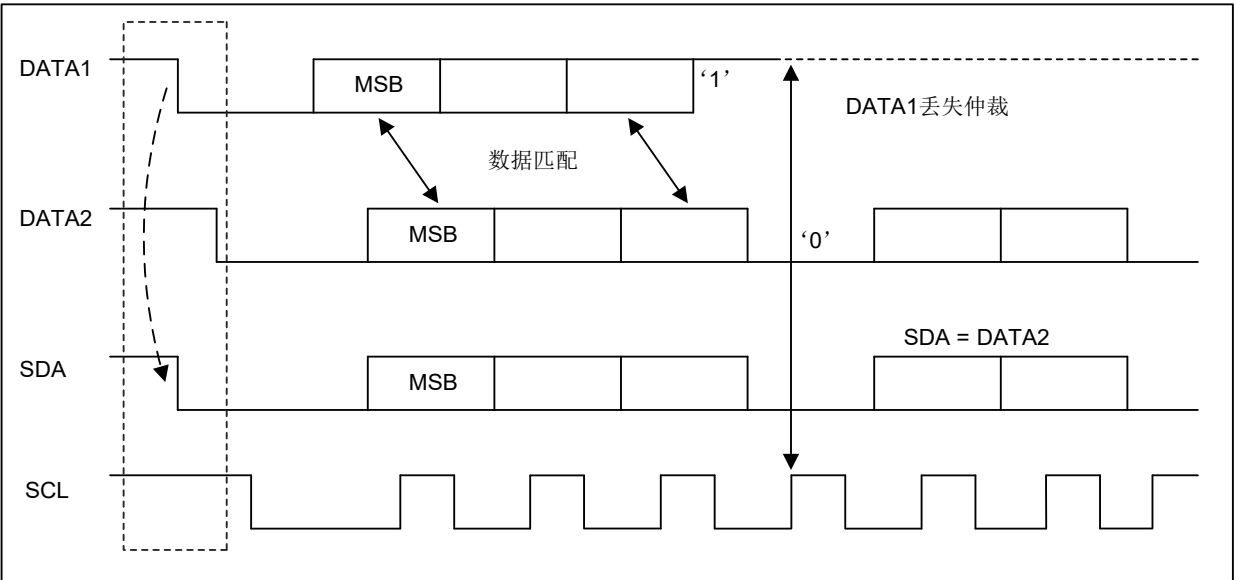


图 32-14 两个主器件仲裁

32.3.3.6 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。

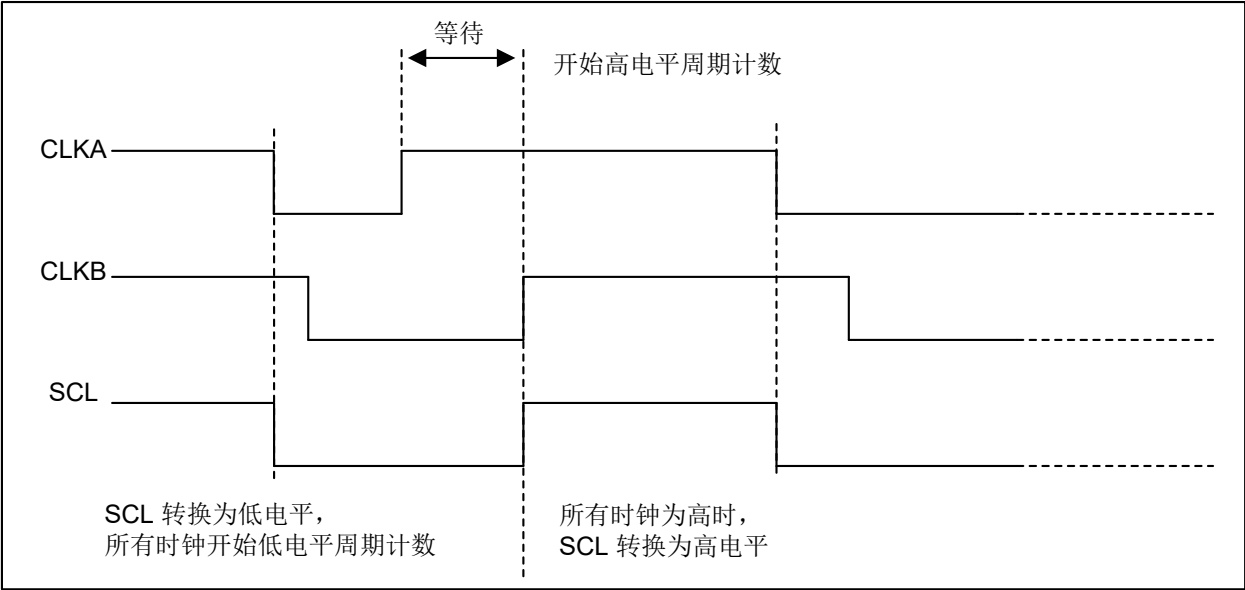


图 32-15 时钟同步（示意图）

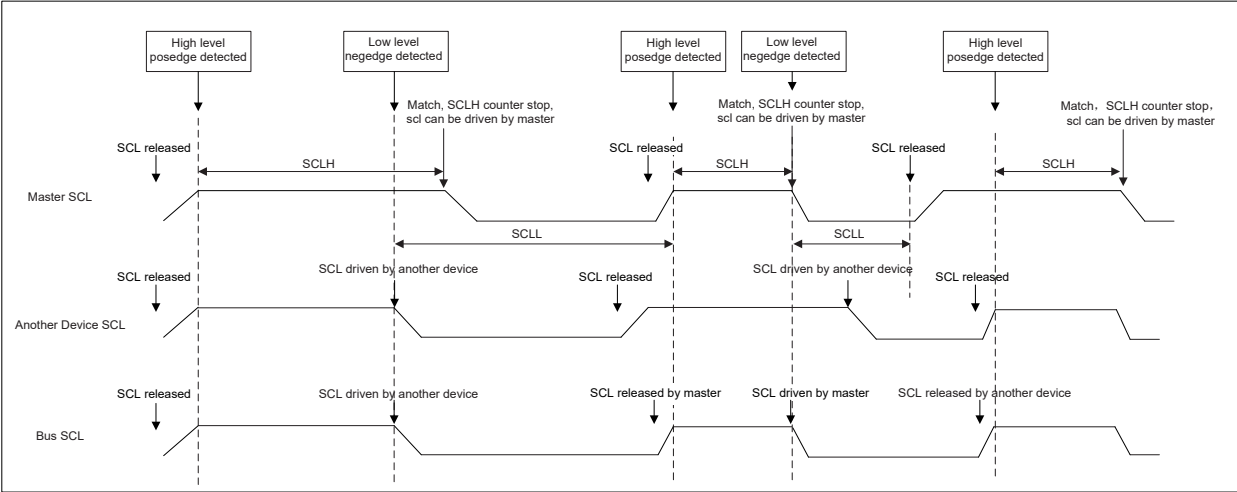


图 32-16 时钟同步（时序图）

32.3.3.7 SCL 配置

I2C 的 SCL 可参考如下配置：

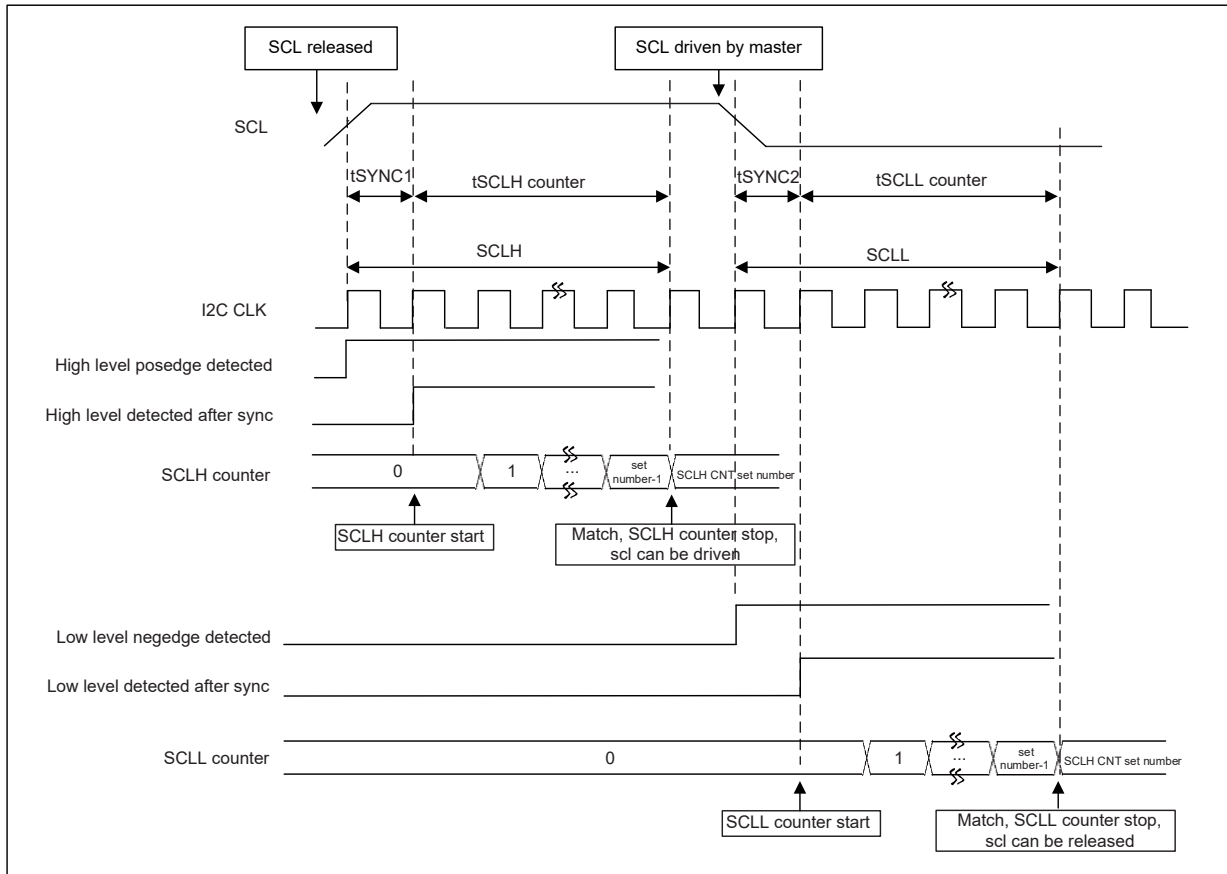


图 32-17 SCL 生成时序

- 标准模式：

$$SCLH = (SSHR + SPKLEN + 3) \times I2C\ CLK + t_{SYNC1}$$

$$SCLL = SSLR \times I2C\ CLK + t_{SYNC2}$$

注： t_{SYNC1} 为 0~1 个 I2C CLK， t_{SYNC2} 为 0~1 个 I2C CLK

- 快速模式、超快速模式：

$$SCLH = (FSHR + SPKLEN + 3) \times I2C\ CLK + t_{SYNC1}$$

$$SCLL = FSLR \times I2C\ CLK + t_{SYNC2}$$

注： t_{SYNC1} 为 2~3 个 I2C CLK， t_{SYNC2} 为 2~3 个 I2C CLK

32.3.4 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）。

32.3.4.1 从模式

- 初始化配置

- 1) 写 0 到 I2C_ENR.ENABLE（位 0）以禁止 I2C
- 2) 通过配置 I2C_SAR 寄存器指定从地址（该地址为 I2C 接口所响应的地址）
- 3) 通过配置 I2C_CR.SLAVE10（位 3）指定地址格式（7 位或 10 位地址）；写 0 到 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）
- 4) 置位 I2C_ENR.ENABLE（位 0），使能 I2C 接口模块

- 从发送的单字节操作

I2C 接口被其他 I2C 主器件寻址并请求数据时，I2C 接口工作在从发送模式，工作流程如下：

- 1) 其他 I2C 主器件发起 I2C 传输，发送地址与 I2C_SAR 寄存器的从地址匹配
- 2) I2C 接口响应发送的地址，识别传输方向以表明其工作在从发送模式
- 3) I2C 接口产生 RD_REQ 中断（I2C_RAWISR 位 5），并将 SCL 线拉低；总线一直处于等待状态直到软件响应。若 RD_REQ 中断被屏蔽（I2C_IMR 位 5 为 0），建议 CPU 定期查询 I2C_RAWISR 寄存器：

- a. 读到 I2C_RAWISR.RD_REQ（位 5）为 1 等效于产生了 RD_REQ 中断
- b. 软件必须采取行动以满足 I2C 传输的要求
- c. 时间间隔通常在 10 个 SCL 时钟周期左右（例如，400kbps 的时间间隔是 25us）

- 4) 如果在接收到读请求前 TX FIFO 中仍有数据，I2C 接口会产生 TX_ABRT 中断（I2C_RAWISR 位 6），以清空 TX FIFO 中的数据（I2C_CR.SLV_TX_ABRT_DIS 为 0 时）

因此，在试图写入 TX FIFO 前，软件有必要通过读取 I2C_TX_ABRT 寄存器来清除 TX_ABRT 中断。若 TX_ABRT 中断被屏蔽（I2C_IMR 位 6 为 0），建议 CPU 定期查询 I2C_RAWISR 寄存器。读到 I2C_RAWISR.TX_ABRT（位 6）为 1 等效于产生了 TX_ABRT 中断。

- 5) 软件写数据到 I2C_DR 寄存器（其中位 8 设置为 0）
- 6) 软件必须先清除 I2C_RAWISR.RD_REQ（位 5）和 I2C_RAWISR.TX_ABRT（位 6）中断
- 7) I2C 接口释放 SCL，并发送数据字节
- 8) 主器件发送重复起始条件控制总线，或发送停止条件释放总线

- 从接收的单字节操作

其他主器件寻址 I2C 接口并发送数据时，I2C 接口工作在从接收模式，工作流程如下：

- 1) 其他 I2C 主器件发起 I2C 传输，发送地址与 I2C_SAR 寄存器的从地址匹配
- 2) I2C 接口响应发送的地址，识别传输方向以表明其工作在从接收模式

3) I2C 接口接收主器件发送的数据并将其存储在接收缓冲中

4) I2C 接口产生 RX_FULL 中断 (I2C_RAWISR 位 2)

若 RX_FULL 中断被屏蔽 (I2C_IMR 位 2 为 0)，建议 CPU 定期查询 I2C_SR 寄存器。读到 I2C_SR.RFNE (位 3) 为 1 等效于产生了 RX_FULL 中断

5) 软件通过读 I2C_DR 寄存器 (位 7:0) 来获得接收到的数据

6) 主器件发送重复起始条件控制总线，或发送停止条件释放总线

● 程序流程图

下图为 I2C 接口作为从器件时的程序示例：

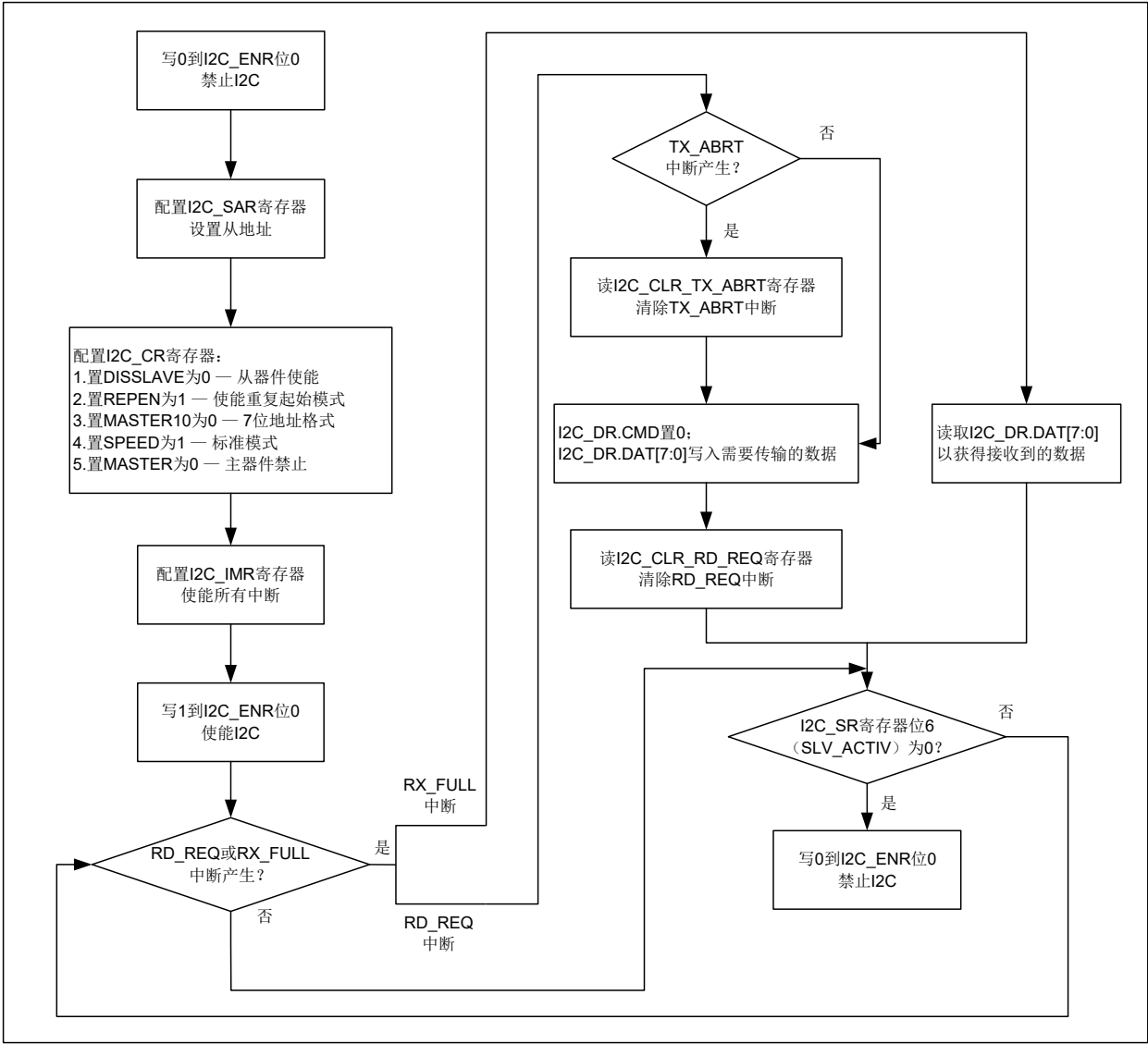


图 32-18 流程图 (I2C 接口作为从器件)

● 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。

32.3.4.2 主模式

- 初始化配置

- 1) 写 0 到 I2C_ENR.ENABLE (位 0) 以禁止 I2C
- 2) 通过配置 I2C_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.DISSLAVER (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.ENABLE (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：

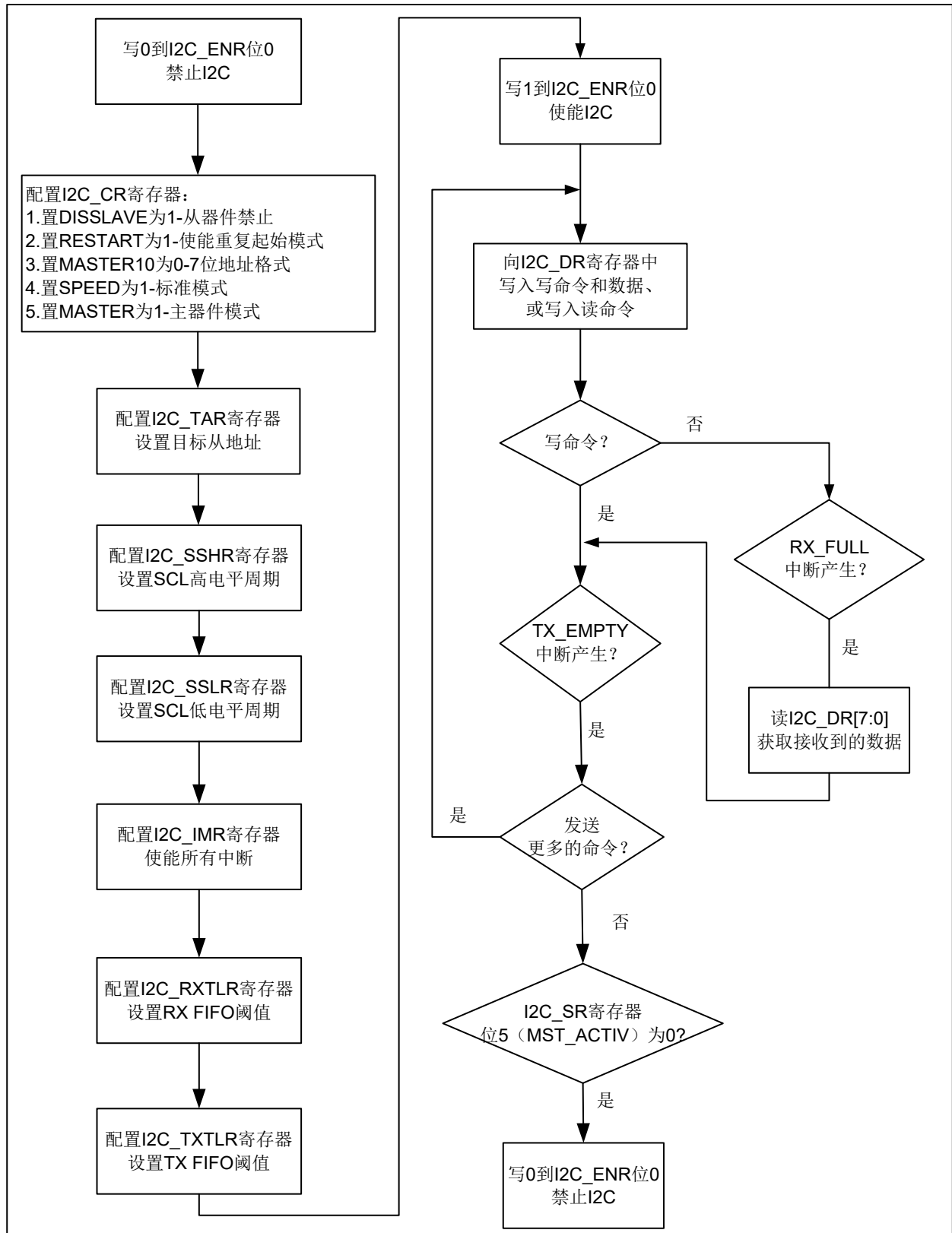


图 32-19 流程图（I2C 接口作为主器件）

32.3.4.3 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
- 5) 读取 I2C_TX_ABRT_SRC 寄存器以获得中断源的信息

32.3.4.4 尖峰滤波

I2C 支持可编程的尖峰滤波器。在内部采样之前检查输入信号 (SCL 和 SDA) 是否在一定数量的 APB 时钟周期内保持稳定。APB 时钟周期数可通过寄存器 I2C_SPKLEN 进行配置。

下图描述了滤波宽度为 5 个 APB 时钟周期 (I2C_SPKLEN.SPKLEN = 5) 时的滤波过程。

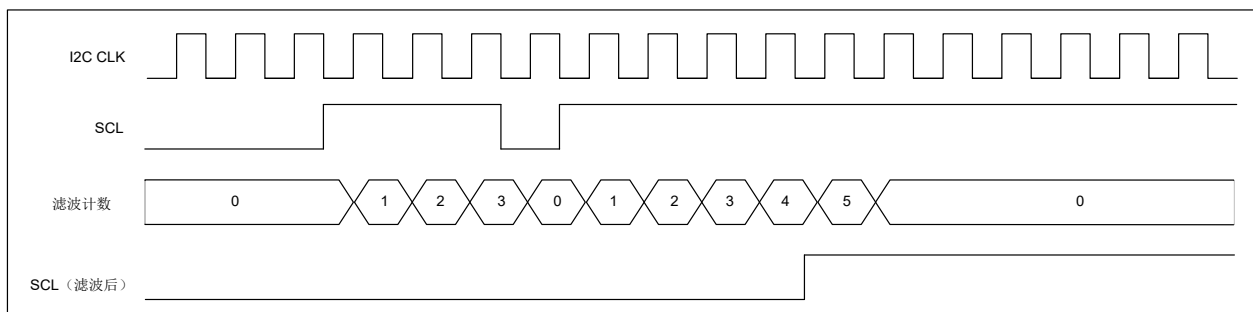


图 32-20 滤波示例

32.3.4.5 超快速模式

I2C 支持高达 1Mbps 的超快速模式。超快速模式下, 数据传输前需进行如下操作:

- 1) 设置 APB 时钟频率为 12MHz 以上
- 2) 配置 I2C_CR.SPEED (位 2: 1), 选择超快速模式
- 3) 配置 I2C_FSHR、I2C_FSLR 寄存器, 设定 SCL 的高低电平周期
- 4) 配置 I2C_FS_SPKLEN 寄存器, 设定滤波宽度为 50ns
- 5) 配置 I2C_SDA_SETUP 以满足最小建立时间 (tSU; DAT)

32.3.4.6 SDA 恢复

I2C 支持 SDA 恢复机制。当 SDA 线被长时间拉低时，主器件通过以下操作进行恢复：

1) 主器件发送时钟脉冲（最多 9 个）以恢复总线。时钟脉冲的数量随着从器件发送的剩余比特数变化

2) 主器件通过在 SDA 线上检测 1 来判断 SDA 是否恢复

a) 若 SDA 未恢复，主器件将继续发送时钟（最多 9 个）

b) 若 SDA 在 9 个时钟内恢复，则主器件发起 STOP 来释放总线

c) 若 SDA 在 9 个时钟内未恢复，则需要硬件复位

下图为 SDA 在第 6 个时钟恢复的时序图。

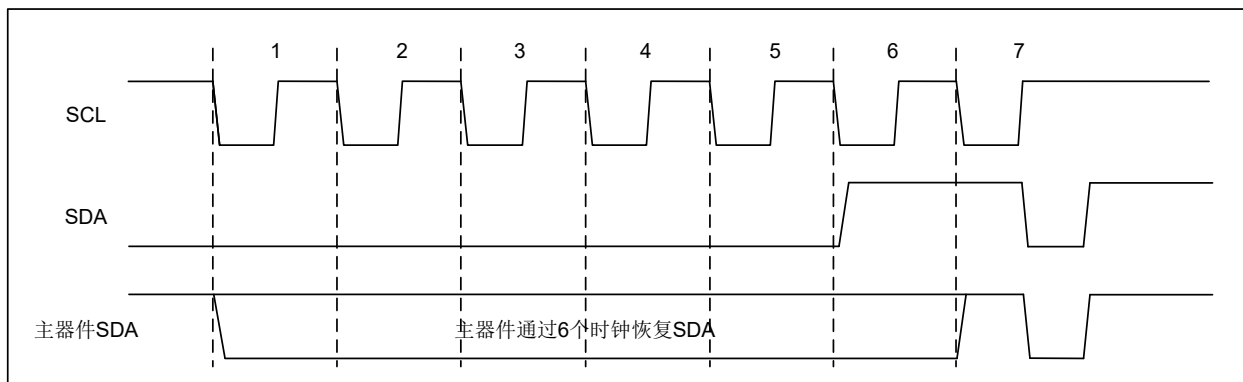


图 32-21 SDA 恢复

32.3.5 SMBus/PMBus

系统可使用 SMBus 与器件进行通信。SMBus 规范涉及三类器件：

- 从器件：接收或响应命令。
- 主器件：发出命令、生成时钟和中止传输。
- 主机（Host）：专用的主器件，为系统 CPU 提供主接口。主机必须具有主-从器件功能，且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

详细信息请参见 SMBus 规范版本 2.0（<http://smbus.org/specs/>）。

下表介绍了 SMBus 和 I2C 的区别。

表 32-3 SMBus 和 I2C

项目	SMBus	I2C
速度	10~100Kbps	~1Mbps（无最小速度）
超时	时钟低电平超时：35ms	未定义超时
逻辑电平	固定值	与 VDD 有关
协议	面向命令的协议	未定义命令

32.3.5.1 超时

1) SMBCLK 低电平超时

SMBus 能够检测器件无限期地保持时钟为低电平的异常状态。

SMBus 模式下启用总线清除功能，用户可以通过 I2C_SCL_TMO 寄存器设定 tTIMEOUT, MIN 来检测 SMBCLK 低电平超时。检测到 SCL_STUCK_LOW 中断后，从器件将重置其通信接口并释放总线。主器件通过 I2C_ENR.ABORT（位 1）产生传输中断，完成当前传输并在总线上生成 STOP 条件。

2) SMBDAT 低电平超时

SMBDAT 长时间保持低电平会阻碍主器件产生 STOP 条件。

SMBus 模式下启用总线清除功能，用户可以通过 I2C_SDA_TMO 寄存器设定 tTIMEOUT, MAX 来检测 SMBDAT 低电平超时。检测到超时时，软件可以置位 I2C_ENR.SMB_CLK_RST（位 16），使 SCL 保持为低至少 tTIMEOUT, MAX，以重置总线上所有器件的 SMBus 接口。

3) 从器件超时

SMBus 定义了 tLOW: SEXT 作为从器件的累积时钟低电平延长时间。测量区间：

- 从初始 START 到 STOP

从器件通过 I2C_SMB_SEXT 寄存器检测超时，并产生中断；主器件可通过 I2C_ENR.ABORT（位

1) 中断超时的通信。

4) 主器件超时

SMBus 定义了 tLOW: MEXT 作为主器件的累积时钟低电平延长时间。测量区间：

- START 到 ACK
- ACK 到 ACK
- ACK 到 STOP

主器件通过 I2C_SMB_MEXT 寄存器检测超时，并产生中断。

32.3.5.2 总线协议

支持 11 类命令协议：快速命令、发送字节、接收字节、写字节、写字、读字节、读字、进程调用、块读、块写、块读进程调用。

SMBus 通过在总线协议的结尾附加 PEC 字节来引入包错误检查机制。

32.3.5.3 地址解析协议（ARP）

SMBus 从地址冲突可以通过为每个从器件动态分配一个新的唯一地址来解决。该功能允许器件热插拔到系统中。

SMBus 为系统中的每个器件引入了唯一的 128 位器件 ID（UDID），以便进行地址分配。ARP 主

器件可以发出通用或定向的地址解析协议（ARP）命令来为系统中的从器件分配动态地址。

32.3.5.4 额外的从地址

SMBus 包含一个额外的从地址寄存器 I2C_OPT_SAR，可写入任何有效的从地址。

32.3.5.5 SMBus 报警

简易器件可使用报警信号（SMBA）请求主机的注意。当检测到报警信号时，主机必须发送一个报警响应地址，只有拉低报警信号的器件响应该地址。

I2C 作为从器件工作时，置位 I2C_ENR.SMB_ALT_CTRL（位 18）后，等待主机发送报警响应地址。收到报警响应地址后，从器件发送地址（I2C_SAR）给主机，传输成功后清除 SMB_ALT_CTRL 和报警信号；若作为主机工作，用户需要通过发送带有报警响应地址的读字节命令来服务报警中断。报警信号的当前状态可通过读寄存器 I2C_SR.SMB_ALT（位 20）获得。

32.3.6 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 32-4 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
SCL_STUCK_LOW	x	√
MST_ON_HOLD	x	√
GEN_CALL	√	x
START_DET	√	x
STOP_DET	√	x
ACTIVITY	√	x
RX_DONE	√	x
TX_ABRT	√	x
RD_REQ	√	x
TX_EMPTY	x	√
TX_OVER	√	x
RX_FULL	x	√
RX_OVER	√	x
RX_UNDER	√	x
SMB_ALERT	√	x
SLV_RX_PEC_NACK	√	x
ARP_ASSGN_ADDR	√	x
ARP_GET_UDID	√	x
ARP_RST	√	x

中断位	硬件置位/软件清除	硬件置位和清除
ARP_PREPARE	√	x
HOST_NOTIFY_MST	√	x
QUICK	√	x
MST_TIMEOUT	√	x
SLV_TIMEOUT	√	x

下图描述了中断寄存器中，中断位被硬件置位和软件清除的操作。

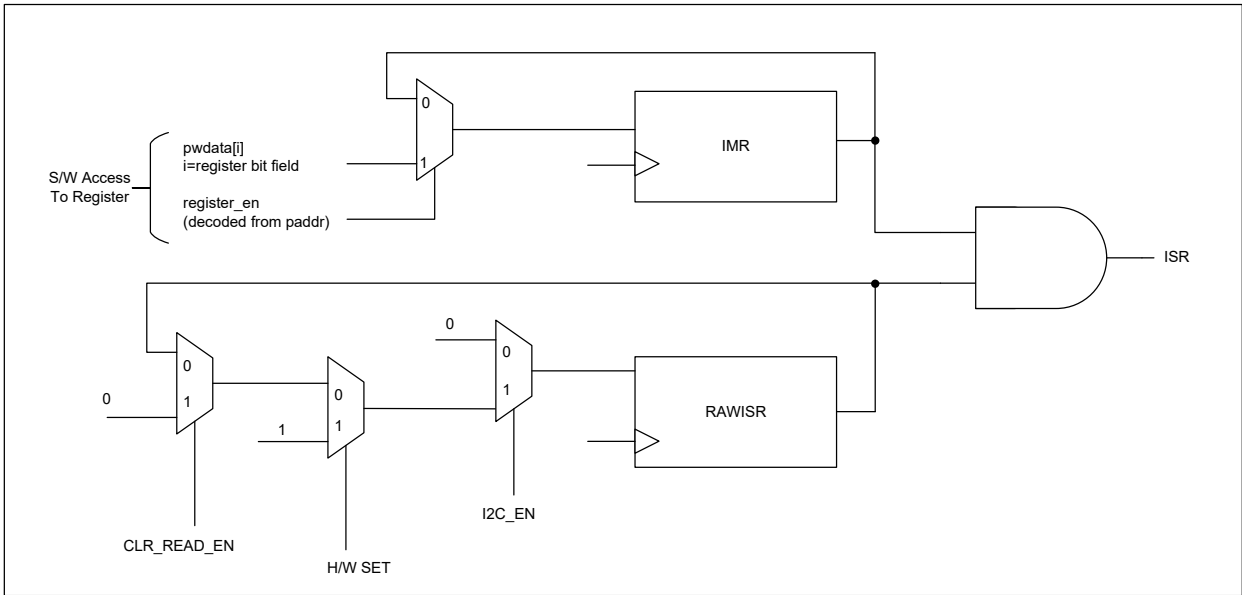


图 32-22 I2C 中断机制

32.3.7 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。

32.3.7.1 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。

32.3.7.2 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。

32.4 寄存器

32.4.1 寄存器总览

表 32-5 I2C 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	I2C_CR	控制寄存器	0x00000065
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/超快速模式时钟高电平计数寄存器	0x0000003C
0x20	I2C_FSLR	快速/超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	0x000048FF
0x34	I2C_RAWISR	RAW 中断寄存器	0x00000000
0x38	I2C_RXTLR	接收阈值寄存器	0x00000000
0x3C	I2C_TXTLR	发送阈值寄存器	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_ABORT	清除 TX_ABORT 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
0x80	I2C_TX_ABORT_SRC	传输中止源寄存器	0x00000000
0x84	I2C_SLV_NACK	从接收 NACK 寄存器	0x00000000
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
0x9C	I2C_EN_SR	ENABLE 状态寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0xA0	I2C_SPKLEN	滤波寄存器	0x00000005
0xAC	I2C_SCL_TMO	SCL 低电平超时寄存器	0xFFFFFFFF
0xB0	I2C_SDA_TMO	SDA 低电平超时寄存器	0xFFFFFFFF
0xB4	I2C_SCL_STUCK	清除 SCL_STUCK 中断寄存器	0x00000000
0xBC	I2C_SMB_SEXT	SMBus 从器件时钟延长超时寄存器	0xFFFFFFFF
0xC0	I2C_SMB_MEXT	SMBus 主器件时钟延长超时寄存器	0xFFFFFFFF
0xC4	I2C_SMB_IDLE	SMBus 总线空闲计数寄存器	0x0000FFFF
0xC8	I2C_SMB_ISR	SMBus 中断状态寄存器	0x00000000
0xCC	I2C_SMB_IMR	SMBus 中断屏蔽寄存器	0x000007FF
0xD0	I2C_SMB_RAWISR	SMBus RAW 中断寄存器	0x00000000
0xD4	I2C_SMB_ICR	SMBus 组合和独立中断清除寄存器	0x00000000
0xD8	I2C_OPT_SAR	可选从地址寄存器	0x00000000
0xDC	I2C_SMB_UDID_LSB	SMBus UDID LSB 寄存器	0xFFFFFFFF
0xE0	I2C_SMB_UDID_MSB0	SMBus UDID MSB 寄存器 0	0x00000000
0xE4	I2C_SMB_UDID_MSB1	SMBus UDID MSB 寄存器 1	0x00000000
0xE8	I2C_SMB_UDID_MSB2	SMBus UDID MSB 寄存器 2	0x00000000
0xEC	I2C_SLVMASK	从地址掩码寄存器	0x000003FF
0xF0	I2C_SLVRCVADDR	从器件接收地址寄存器	0x00000000

32.4.2 I2C_CR 控制寄存器

偏移地址：0x00

复位值：0x0000 0065

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									RX_FUL L_HLD	BUS_CL R	STOP_ DET_M ST_ACT	SMB_P SA_EN	SMB_A RP_EN	SMB_S LV_QC_ EN	OPT_S AR_EN
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			PAD_SE L	SLV_TX _ ABRT_ DIS	RESTA RT	STOP	EMPINT	STOPIN T	DISSLA VE	REPEN	MASTE R10	SLAVE1 0	SPEED		MASTE R
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31: 23	Reserved	保留，必须保持复位值
22	RX_FULL_HLD	0: RX FIFO 已满（达到最大深度）时不保持总线 1: RX FIFO 已满时保持总线

Bit	Field	Description
21	BUS_CLR	0: 总线清除 (Bus Clear) 禁止 1: 总线清除使能 注: 从模式该位无效
20	STOP_DET_MST_ACT	主模式时, 0: 无论主器件是否活动都产生 STOP 中断 1: 仅在主器件活动时产生 STOP 中断
19	SMB_PSA_EN	0: 非持久从器件 (no-persistent slave); 发生 General/Directed Reset 命令时从器件清除地址有效标志位 1: 持久从器件 (persistent slave); 地址有效标志位始终为 1 注: 该位仅支持从模式。
18	SMB_ARP_EN	0: SMBus 模式下地址解析逻辑禁止 1: SMBus 模式下解码并响应地址解析协议 (ARP) 命令 注: 该位仅支持从模式。
17	SMB_SLV_QC_EN	0: SMBus 模式下从器件接收除快速命令以外的所有命令 1: SMBus 模式下从器件只接收快速命令 注: 该位仅支持从模式。
16	OPT_SAR_EN	0: 禁用 OPT_SAR 寄存器 1: 启用 OPT_SAR 寄存器; 置 1 前需向 OPT_SAR 写入有效的从地址
15: 13	Reserved	保留, 必须保持复位值
12	PAD_SEL	PAD MUX 交换 SCL、SDA 对应 PAD 0: PAD0 $\leftrightarrow$ SCL; PAD1 $\leftrightarrow$ SDA 1: PAD0 $\leftrightarrow$ SDA; PAD1 $\leftrightarrow$ SCL
11	SLV_TX_ABRT_DIS	I2C 作为从发送器时 0: 接收到 RD_REQ 信号后清除 TX FIFO 1: 接收到 RD_REQ 信号后禁止清除 TX FIFO
10	RESTART	发送或接收字节前, 是否产生 RESTART 0: 如果 REPEN 为 1, 则仅在前一个命令改变传输方向时才产生 RESTART; 如果 REPEN 为 0, 则先产生 STOP 再产生 START 1: 如果 REPEN 为 1, 则在数据接收或发送 (根据 CMD 的值) 前产生 RESTART, 无论前一个命令是否改变数据的传输方向; 如果 REPEN 为 0, 则先产生 STOP 再产生 START 注: I2C_ENR.ENABLE 为 1 时该位可写。
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: I2C_ENR.ENABLE 为 1 时该位可写。

Bit	Field	Description
8	EMPINT	该位控制 TX_EMPTY 中断的产生, 细节请参考 I2C_RAWISR 寄存器
7	STOPINT	从模式下, 是否产生 STOP 中断 0: 无论地址是否匹配, 都产生 STOP 中断 1: 仅在地址匹配时产生 STOP 中断 注: 广播地址寻址时, 如果该位置位, 从器件不产生 STOP 中断。STOP 中断仅在发送地址与从地址 (I2C_SAR) 匹配时产生。
6	DISSLAVE	该位控制是否禁用 I2C 从器件 0: 从器件使能 1: 从器件禁止
5	REPEN	I2C 作为主器件时是否发送 RESTART 条件 0: 禁止 1: 使能 RESTART 条件可以替换为先发送停止条件再发送起始条件。 当 RESTART 被禁止时, I2C 接口作为主器件不能执行以下功能: 发送起始字节 组合格式模式下改变传输方向 10 位地址格式的读操作 执行上述操作会置位 I2C_RAWISR.TX_ABRT (位 6)。
4	MASTER10	I2C 作为主器件时的地址格式 0: 7 位地址格式 1: 10 位地址格式 注: I2C_ENR.ENABLE 为 0; 或 I2C_ENR.ENABLE 为 1 且 I2C_CR.MASTER 为 1 时该位可写。
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) 或超快速模式 (最大 1Mbps)
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

注: I2C_ENR.ENABLE 为 0 时该寄存器 (除 STOP、RESTART、MASTER10 位以外) 可写。

I2C_CR.DISSLAVE (位 6) 和 I2C_CR.MASTER (位 0) 的配置组合如下表所示:

表 32-6 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

32.4.3 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.															SMB_QC
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				SPECIAL	GC	ADDR									
				rw	rw	rw									

Bit	Field	Description
31: 17	Reserved	保留，必须保持复位值
16	SMB_QC	位 11 置位时，该位指示 I2C 执行的是否为快速命令 0：非快速命令 1：快速命令
15: 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。

注：I2C_ENR.ENABLE 为 0；或 I2C_ENR.ENABLE 为 1 且 I2C_CR.MASTER 为 1 时该寄存器可写。

32.4.4 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。

32.4.5 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.				FIRST_DATA	Res.		CMD	DAT							
				r			w	rw							

Bit	Field	Description
31: 12	Reserved	保留，必须保持复位值
11	FIRST_DATA	主接收或从接收模式下，地址传输后接收到第一个数据字节时置位。
10: 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

32.4.6 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 1 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。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.7 I2C_SSLR 标准模式时钟低电平计数寄存器

偏移地址：0x18

复位值：0x0000 01D6

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

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	CNT	标准模式下 SCL 时钟低电平周期，最小值为 6。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.8 I2C_FSHR 快速/超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/超快速模式下 SCL 时钟高电平周期，最小值为 1。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.9 I2C_FSLR 快速/超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/超快速模式下 SCL 时钟低电平周期，最小值为 6。												

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.10 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.	R_SCL_STUCK_AT_LOW	R_MST_ON_HOLD	Res.	R_GC	R_START	R_STOP	R_ACTIV	R_RX_DONE	R_TX_ABORT	R_RD_REQ	R_TX_EMPTY	R_TX_OVER	R_RX_FULL	R_RX_UNDER	
	r	r		r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 15	Reserved	保留，必须保持复位值
14	R_SCL_STUCK_AT_LOW	具体描述可以参考 I2C_RAWISR 寄存器
13	R_MST_ON_HOLD	具体描述可以参考 I2C_RAWISR 寄存器
12	Reserved	保留，必须保持复位值
11	R_GC	具体描述可以参考 I2C_RAWISR 寄存器
10	R_START	具体描述可以参考 I2C_RAWISR 寄存器
9	R_STOP	具体描述可以参考 I2C_RAWISR 寄存器
8	R_ACTIV	具体描述可以参考 I2C_RAWISR 寄存器
7	R_RX_DONE	具体描述可以参考 I2C_RAWISR 寄存器

Bit	Field	Description
6	R_TX_ABRT	具体描述可以参考 I2C_RAWISR 寄存器
5	R_RD_REQ	具体描述可以参考 I2C_RAWISR 寄存器
4	R_TX_EMPTY	具体描述可以参考 I2C_RAWISR 寄存器
3	R_TX_OVER	具体描述可以参考 I2C_RAWISR 寄存器
2	R_RX_FULL	具体描述可以参考 I2C_RAWISR 寄存器
1	R_RX_OVER	具体描述可以参考 I2C_RAWISR 寄存器
0	R_RX_UNDER	具体描述可以参考 I2C_RAWISR 寄存器

32.4.11 I2C_IMR 中断屏蔽寄存器

偏移地址：0x30

复位值：0x0000 48FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	M_SCL_STUCK	M_MST_ON_HOLD	Res.	M_GC	M_START	M_STOP	M_ACTIV	M_RX_DONE	M_TX_ABRT	M_RD_REQ	M_TX_EMPTY	M_TX_OVER	M_RX_FULL	M_RX_OVER	M_RX_UNDER
	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31: 15	Reserved	保留，必须保持复位值
14	M_SCL_STUCK	屏蔽与 I2C_ISR 寄存器对应的中断状态位
13	M_MST_ON_HOLD	屏蔽与 I2C_ISR 寄存器对应的中断状态位
12	Reserved	保留，必须保持复位值
11	M_GC	屏蔽与 I2C_ISR 寄存器对应的中断状态位
10	M_START	屏蔽与 I2C_ISR 寄存器对应的中断状态位
9	M_STOP	屏蔽与 I2C_ISR 寄存器对应的中断状态位
8	M_ACTIV	屏蔽与 I2C_ISR 寄存器对应的中断状态位
7	M_RX_DONE	屏蔽与 I2C_ISR 寄存器对应的中断状态位
6	M_TX_ABRT	屏蔽与 I2C_ISR 寄存器对应的中断状态位
5	M_RD_REQ	屏蔽与 I2C_ISR 寄存器对应的中断状态位
4	M_TX_EMPTY	屏蔽与 I2C_ISR 寄存器对应的中断状态位
3	M_TX_OVER	屏蔽与 I2C_ISR 寄存器对应的中断状态位
2	M_RX_FULL	屏蔽与 I2C_ISR 寄存器对应的中断状态位
1	M_RX_OVER	屏蔽与 I2C_ISR 寄存器对应的中断状态位
0	M_RX_UNDER	屏蔽与 I2C_ISR 寄存器对应的中断状态位

32.4.12 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.	SCL_STUCK	MST_ON_HOLD	Res.	GC	START	STOP	ACTIV	RX_DONE	TX_ABORT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
	r	r		r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 15	Reserved	保留, 必须保持复位值
14	SCL_STUCK	SCL 低电平超时检测 当 SCL 线的低电平时间超过 I2C_SCL_TMO 的设定值时置位。
13	MST_ON_HOLD	主器件保持总线且 TX FIFO 为空时置位。
12	Reserved	保留, 必须保持复位值
11	GC	广播呼叫 (General call) 接收到广播呼叫地址并响应时置位。 禁止 I2C 接口或当 CPU 读 I2C_GC.GC (位 0) 时清零。 I2C 将接收到的数据存储在接收缓冲中。
10	START	起始条件检测 (START condition detection) 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 (STOP condition detection) 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: - 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。

Bit	Field	Description
7	RX_DONE	从发送结束 (Transmit done) 当 I2C 作为从发送器时, 如果主器件没有响应发送的字节, 则该位置位。 该情况发生在传输的最后一个字节则表示传输结束。
6	TX_ABRT	发送中止 (Transmit abort) 当 I2C 接口作为发送器, 不能发送完缓冲中的数据时置位。 注意: 发送中止会将 I2C 接口的接收和发送缓冲清空。发送缓冲会处于刷新状态直到读 I2C_TX_ABRT 寄存器。一旦该读操作执行后, 发送缓冲就可以接收 APB 接口上的新数据。
5	RD_REQ	读请求 (Read request) 当 I2C 作为从器件, 其他主器件试图从 I2C 接口读取数据时置位。 I2C 接口会使总线保持在等待状态 (SCL = 0) 直到中断被处理。这表明 I2C 接口被其他主器件寻址成功且被要求发送数据。处理器必须响应该中断然后写数据到 I2C_DR 寄存器中。该位在处理器读 I2C_RD_REQ 寄存器后清零。
4	TX_EMPTY	发送缓冲空 (Transmit buffer empty) 该位状态取决于 I2C_CR.EMPINT (位 8) 的状态: EMPINT = 0, 发送缓冲区数据个数小于等于阈值时置位; EMPINT = 1, 发送缓冲区数据个数小于等于阈值且前一个发送地址/数据的内部移位寄存器发送结束时置位。 发送缓冲区数据个数大于阈值时由硬件自动清零。
3	TX_OVER	发送缓冲过载 (Transmit buffer over) 发送缓冲满时处理器写入新数据导致溢出时置位。
2	RX_FULL	接收缓冲满 (Receive buffer full) 接收缓冲区数据大于阈值时置位。 接收缓冲区数据小于等于阈值时由硬件清零。
1	RX_OVER	接收缓冲过载 (Receive buffer over) 接收缓冲满时收到新数据时置位。此时 I2C 接口会响应, 但新的数据会丢失。 注: RX_FULL_HLD = 1 时无法发生 RX_OVER 中断。
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。

32.4.13 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: 2	Reserved	保留，必须保持复位值
1: 0	TL	接收 FIFO 阈值（Receive FIFO threshold level） 控制 RX_FULL 中断触发。

32.4.14 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: 2	Reserved	保留，必须保持复位值
1: 0	TL	发送 FIFO 阈值（Transmit FIFO threshold level） 控制 TX_EMPTY 中断触发。

32.4.15 I2C_ICR 组合和独立中断清除寄存器

偏移地址：0x40

复位值：0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	ICR	读该寄存器将会清除所有组合中断、独立中断。 该位不清除硬件可自动清除的中断，仅清除软件可清除中断。

32.4.16 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])

32.4.17 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])

32.4.18 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]）

32.4.19 I2C_RD_REQ 清除 RD_REQ 中断寄存器

偏移地址：0x50

复位值：0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RD_REQ	读该寄存器清零 RD_REQ 中断（I2C_RAWISR[5]）

32.4.20 I2C_TX_ABRT 清除 TX_ABRT 中断寄存器

偏移地址：0x54

复位值：0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	TX_ABRT	读该寄存器清零 TX_ABRT 中断（I2C_RAWISR[6]） 同时将 TX FIFO 从刷新/复位状态中释放，以便接收写入的数据。

32.4.21 I2C_RX_DONE 清除 RX_DONE 中断寄存器

偏移地址：0x58

复位值：0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	RX_DONE	读该寄存器清零 RX_DONE 中断（I2C_RAWISR[7]）

32.4.22 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。

32.4.23 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])

32.4.24 I2C_START 清除 START_DET 中断寄存器

偏移地址: 0x64

复位值: 0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值
0	START	读该寄存器清零 START 中断 (I2C_RAWISR[10])

32.4.25 I2C_GC 清除 GEN_CALL 中断寄存器

偏移地址: 0x68

复位值: 0x0000 0000

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

Bit	Field	Description
31: 1	Reserved	保留, 必须保持复位值
0	GC	读该寄存器清零 GC 中断 (I2C_RAWISR[11])

32.4.26 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.													SMB_A LT_EN	Res.	SMB_C LK_RST
													rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												SDA_R CV_EN	TX_CM D_BLO CK	ABORT	ENABL E
												rw	rw	rw	rw

Bit	Field	Description
31: 19	Reserved	保留，必须保持复位值
18	SMB_ALT_EN	SMBus 报警使能 0: SMBus 报警信号禁止 1: SMBus 报警信号使能 检测到来自主机的报警响应地址确认后，该位自动清除。
17	Reserved	保留，必须保持复位值
16	SMB_CLK_RST	SMBus 主机 (Host) 模式下，置位该位可发起 SMBus 主时钟复位。 SMBCLK 保持 I2C_SCL_TMO.SCL_TIMEOUT 个周期的低电平以复位从器件。 注：只能在主器件空闲时启用。
15: 4	Reserved	保留，必须保持复位值
3	SDA_RCV_EN	SDA 因 TX_ABORT 中断而停留在低电平时，该位可用于启用 SDA 恢复机制（发送最多 9 个 SCL 时钟和停止条件来释放 SDA 线），之后该位自动清除。
2	TX_CMD_BLOCK	主模式下， 0: TX FIFO 中第一个数据有效时自动开始传输 1: 阻止数据在 I2C 总线上的传输，即使 TX FIFO 有数据为阻塞主器件命令的执行，需在 TX FIFO 为空且主器件处于 IDLE 状态时置位；该位未置 0 前，TX FIFO 中的任何命令都不会被执行。
1	ABORT	I2C 传输中止 (I2C transfer abort) 0: 中止没有发生或者已经结束 1: 中止操作正在进行 I2C 模块作为主器件时，软件可以置位该位以中止 I2C 传输。一旦置位不能立即清除。置位后 I2C 模块控制逻辑会在完成当前传输后产生 STOP 条件并清空发送缓冲，中止操作之后产生 TX_ABORT 中断。该 ABORT 位会在中止操作结束后自动清零。

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

32.4.27 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.											SMB_A LT	Res.	SMB_S LV_AR	SMB_S LV_AV	SMB_Q C
											r		r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				SDA_N OT_RE COVER ED	SLV_HO LD_RX_ FULL	SLV_HO LD_TX_ EMPTY	MST_H OLD_R X_FULL	MST_H OLD_TX _EMPT Y	SLV_AC TIV	MST_A CTIV	RFF	RFNE	TFE	TFNF	ACTIV
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 21	Reserved	保留，必须保持复位值
20	SMB_ALT	SMBus 报警状态位 0: 未检测到 SMBus 报警 1: 检测到 SMBus 报警
19	Reserved	保留，必须保持复位值
18	SMB_SLV_AR	SMBus 从地址解析状态位（SMBus Slave address Resolved） 0: 从地址未解析 1: 从地址已解析
17	SMB_SLV_AV	SMBus 从地址有效状态位（SMBus Slave address Valid） 0: 从地址无效 1: 从地址有效
16	SMB_QC	SMBus 快速命令状态位 0: 未接收到快速命令 1: 接收到快速命令 该位读后自动清零。
15: 12	Reserved	保留，必须保持复位值
11	SDA_NOT_RECOVERED	恢复机制启用后，SDA 未恢复时置位
10	SLV_HOLD_RX_FULL	从器件由于接收过载而处于保持状态时置位
9	SLV_HOLD_TX_EMPTY	从器件由于收到读请求时 TX FIFO 为空而处于保持状态时置位 直到 TX FIFO 有数据为止，从器件始终处于保持状态。
8	MST_HOLD_RX_FULL	主器件由于接收过载而处于保持状态时置位

Bit	Field	Description
7	MST_HOLD_TX_EMPTY	主器件由于 TX FIFO 为空而处于保持状态时置位
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：接收缓冲未滿 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	发送缓冲未滿 0：发送缓冲满 1：发送缓冲未滿
0	ACTIV	I2C 活动状态 MST_ACTIV 位与 SLV_ACTIV 位相或的结果。

32.4.28 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: 3	Reserved	保留，必须保持复位值
2: 0	CNT	发送缓冲中有效数据的个数（0~4）

32.4.29 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: 3	Reserved	保留，必须保持复位值
2: 0	CNT	接收缓冲中有效数据的个数（0~4）

32.4.30 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。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.31 I2C_TX_ABRT_SRC 传输中止源寄存器

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
TX_FLUSH_CNT									Res.					SDA_ LOW	USER_ ABRT
r														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SLVRD_INTX	SLV_ARBLOST	SLVFLUSH_TXFIFO	LOST	MST_DIS	10B_RD_NORSTRT	SBYTE_NORSTRT	Res.	SBYTE_ACKDET	Res.	GC_READ	GC_NOACK	TXDATA_NOACK	10ADDR2_NOACK	10ADDR1_NOACK	7ADDR_NOACK
r	r	r	r	r	r	r		r		r	r	r	r	r	r

Bit	Field	Description
31: 23	TX_FLUSH_CNT	因传输中断被刷新掉的 TX FIFO 中的数据/命令个数
22: 18	Reserved	保留, 必须保持复位值
17	SDA_LOW	主器件检测到 SDA 低电平超时则该位置位
16	USER_ABRT	主器件检测到传输中止 (I2C_ENR.ABORT 为 1) 时置位
15	SLVRD_INTX	从发送模式, I2C 被请求数据时, 向 I2C_DR.CMD (位 8) 写 1 (读命令), 则该位置位
14	SLV_ARBLOST	从器件发送数据时丢失总线, 则该位与位 12 同时置位
13	SLVFLUSH_TXFIFO	1: 从器件接收到读请求时 TX FIFO 中仍有数据, 从器件通过产生 TX_ABRT 中断清除 TX FIFO 中的旧数据。
12	LOST	主器件仲裁失败, 或位 14 置 1 时从器件仲裁失败, 则该位置位
11	MST_DIS	主模式禁用时操作主器件, 则该位置位
10	10B_RD_NORSTRT	RESTART 未使能 (I2C_CR.REPEN 为 0) 时, 主器件在 10 位寻址模式下发送读请求, 则该位置位
9	SBYTE_NORSTRT	RESTART 未使能 (I2C_CR.REPEN 为 0) 时, 发送起始字节, 则该位置位
8	Reserved	保留, 必须保持复位值
7	SBYTE_ACKDET	主器件发送起始字节且被响应 (错误行为) 时置位
6	Reserved	保留, 必须保持复位值
5	GC_READ	主器件发出广播呼叫后, 接着发送读请求时置位
4	GC_NOACK	主器件发出广播呼叫后, 无从器件响应时置位
3	TXDATA_NOACK	主器件收到地址响应后, 无从器件响应发送的数据时置位
2	10ADDR2_NOACK	主器件工作在 10 位寻址模式, 无从器件响应第二个地址字节时置位
1	10ADDR1_NOACK	主器件工作在 10 位寻址模式, 无从器件响应第一个地址字节时置位
0	7ADDR_NOACK	主器件工作在 7 位寻址模式, 无从器件响应地址时置位

32.4.32 I2C_SLV_NACK 从接收 NACK 寄存器

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.															NACK
															rw

Bit	Field	Description
31: 1	Reserved	保留，必须保持复位值
0	NACK	0: 正常产生 ACK/NACK 1: 从器件接收数据后产生 NACK

注：I2C_ENR.ENABLE 为 0 且 I2C_SR.SLV_ACTIV 为 0 时该寄存器可写。

32.4.33 I2C_DMA DMA 控制寄存器

偏移地址：0x88

复位值：0x0000 0000

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

Bit	Field	Description
31: 2	Reserved	保留，必须保持复位值
1	TXEN	0: DMA 发送禁止 1: DMA 发送使能
0	RXEN	0: DMA 接收禁止 1: DMA 接收使能

32.4.34 I2C_SETUP SDA 建立时间寄存器

偏移地址：0x94

复位值：0x0000 0064

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	CNT	SDA 建立时间，最小值为 2。 如果所需延迟时间为 1000ns，当 APB 时钟频率为 10MHz 时，建议将该寄存器设为 11。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.35 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)

32.4.36 I2C_EN_SR ENABLE 状态寄存器

偏移地址：0x9C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	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_RX_DATA_LOST	SLV_RX_DATA_LOST	SLV_DI S_WHIL E_BUS Y	IC_EN
													r	r	r

Bit	Field	Description
31: 3	Reserved	保留，必须保持复位值
2	SLV_RX_DATA_LOST	该位表示由于将 I2C_ENR.ENABLE 从 1 设为 0 而导致从接收动作（至少接收到一个数据字节）被中断。（Slave Received Data Lost） 对该位的读操作需在 IC_EN 读 0 后进行。

Bit	Field	Description
1	SLV_DIS_WHILE_BUSY	该位指示是否由于将 I2C_ENR.ENABLE 从 1 设为 0 而终止了进行中的从模式动作。(Slave Disabled While Busy) 以下情况该位置 1： 从发送模式下 I2C 正在接收地址时，写 0 到 I2C_ENR.ENABLE 从接收模式下 I2C 正在接收数据或地址时，写 0 到 I2C_ENR.ENABLE 对该位的读操作需在 IC_EN 读 0 后进行。
0	IC_EN	0: I2C 处于非活动状态 1: I2C 处于活动状态

32.4.37 I2C_SPKLEN 滤波寄存器

偏移地址：0xA0

复位值：0x0000 0005

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

Bit	Field	Description
31: 8	Reserved	保留，必须保持复位值
7: 0	SPKLEN	配置尖峰滤波（Spike Suppression）时间，单位为 APB 时钟周期 最小值为 1。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.38 I2C_SCL_TMO SCL 低电平超时寄存器

偏移地址：0xAC

复位值：0xFFFF FFFF

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

Bit	Field	Description
31: 0	SCL_TIMEOUT	配置 SCL 低电平超时的时长, 单位为 APB 时钟周期

32.4.39 I2C_SDA_TMO SDA 低电平超时寄存器

偏移地址: 0xB0

复位值: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDA_TIMEOUT															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDA_TIMEOUT															
rw															
Bit	Field		Description												
31: 0	SDA_TIMEOUT		配置 SDA 低电平超时的时长，单位为 APB 时钟周期 通过使能 I2C_ENR.SDA_RCV_EN（位 3）可在超时后启动 SDA 恢复机制。												

32.4.40 I2C_SCL_STUCK 清除 SCL_STUCK 中断寄存器

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.															SCL_STUCK
															r
Bit	Field				Description										
31: 1	Reserved				保留，必须保持复位值										
0	SCL_STUCK				读该寄存器清零 SCL_STUCK 中断（RAWISR[14]）										

32.4.41 I2C_SMB_SEXT SMBus 从器件时钟延长超时寄存器

偏移地址: 0xBC

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMB_SEXT															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB_SEXT															
rw															
Bit	Field		Description												
31: 0	SMB_SEXT		配置从器件时钟延长超时 (tLOW:SEXT)，单位为 APB 时钟周期 从器件时钟延长：主模式时，一次通信过程中 (START 到 STOP)，由从器件引起的时钟延长												

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.42 I2C_SMB_MEXT SMBus 主器件时钟延长超时寄存器

偏移地址：0xC0

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMB_MEXT															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB_MEXT															
rw															
Bit	Field		Description												
31: 0	SMB_MEXT		配置主器件传输一个字节的时钟延长超时 (tLOW:MEXT)， 单位为 APB 时钟周期												

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.43 I2C_SMB_IDLE SMBus 总线空闲计数寄存器

偏移地址：0xC4

复位值：0x0000 FFFF

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

Bit	Field	Description
31: 16	Reserved	保留，必须保持复位值
15: 0	SMB_IDLE_CNT	配置所需的总线空闲时间，单位为 APB 时钟周期 当一个主器件被动态添加到总线，且可能没有检测到 SCL 或 SDA 线上的状态转换时使用。在这种情况下，主器件必须等待，以确保当前没有传输正在进行。

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.44 I2C_SMB_ISR SMBus 中断状态寄存器

偏移地址：0xC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	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_SMB _ALT	Res.	R_PEC_ NACK	R_ARP_ ASSGN	R_ARP_ UDID	R_ARP_ RST	R_ARP_ PRE	R_ARP_ NOTIFY	R_ARP_ QUICK	R_MST_ _TMO	R_SLV_ TMO
					r		r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 11	Reserved	保留，必须保持复位值
10	R_SMB_ALT	具体描述可以参考 I2C_SMB_RAWISR 寄存器
9	Reserved	保留，必须保持复位值
8	R_PEC_NACK	具体描述可以参考 I2C_SMB_RAWISR 寄存器
7	R_ARP_ASSGN	具体描述可以参考 I2C_SMB_RAWISR 寄存器
6	R_ARP_UDID	具体描述可以参考 I2C_SMB_RAWISR 寄存器
5	R_ARP_RST	具体描述可以参考 I2C_SMB_RAWISR 寄存器
4	R_ARP_PRE	具体描述可以参考 I2C_SMB_RAWISR 寄存器
3	R_ARP_NOTIFY	具体描述可以参考 I2C_SMB_RAWISR 寄存器
2	R_ARP_QUICK	具体描述可以参考 I2C_SMB_RAWISR 寄存器
1	R_MST_TMO	具体描述可以参考 I2C_SMB_RAWISR 寄存器
0	R_SLV_TMO	具体描述可以参考 I2C_SMB_RAWISR 寄存器

32.4.45 I2C_SMB_IMR SMBus 中断屏蔽寄存器

偏移地址：0xCC

复位值：0x0000 07FF

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					M_SMB _ALT	Res.	M_PEC _NACK	M_ARP _ASSG N	M_ARP _UDID	M_ARP _RST	M_ARP _PRE	M_ARP _NOTIF Y	M_ARP _QUICK	M_MST _TMO	M_SLV_ TMO
					rw		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	M_SMB_ALT	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
9	Reserved	保留，必须保持复位值
8	M_PEC_NACK	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
7	M_ARP_ASSGN	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
6	M_ARP_UDID	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
5	M_ARP_RST	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
4	M_ARP_PRE	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
3	M_ARP_NOTIFY	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
2	M_ARP_QUICK	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
1	M_MST_TMO	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位
0	M_SLV_TMO	屏蔽与 I2C_SMB_ISR 寄存器对应的中断状态位

32.4.46 I2C_SMB_RAWISR SMBus RAW 中断寄存器

偏移地址：0xD0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					SMB_A LT	Res.	PEC_N ACK	ARP_A SSGN	ARP_U DID	ARP_R ST	ARP_P RE	ARP_N OTIFY	ARP_Q UICK	MST_T MO	SLV_TM O
					r		r	r	r	r	r	r	r	r	r

Bit	Field	Description
31: 11	Reserved	保留，必须保持复位值
10	SMB_ALT	检测到从器件报警（SMBA 为低电平）时置位
9	Reserved	保留，必须保持复位值
8	PEC_NACK	从器件不响应 ARP 命令的 PEC 字节时置位
7	ARP_ASSGN	接收到 Assign Address ARP 命令时置位
6	ARP_UDID	接收到 Get UDID ARP 命令时置位

Bit	Field	Description
5	ARP_RST	接收到 Reset ARP 命令时置位
4	ARP_PRE	接收到 Prepare to ARP 命令时置位
3	ARP_NOTIFY	接收到 Host Notify 命令时置位
2	ARP_QUICK	接收到 Quick 命令时置位
1	MST_TMO	主器件传输一个字节（START 到 ACK、ACK 到 ACK、ACK 到 STOP）的时钟延长超时则该位置位
0	SLV_TMO	从器件通信一次（START 到 STOP）的时钟延长超时则该位置位

32.4.47 I2C_SMB_ICR SMBus 组合和独立中断清除寄存器

偏移地址：0xD4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CLR_S MB_ALT	Res.	CLR_PE C_NACK	CLR_A RP_ASSGN	CLR_A RP_UDID	CLR_A RP_RST	CLR_A RP_PR	CLR_A RP_NO	CLR_A RP_QUICK	CLR_M ST_TM O	CLR_SL V_TMO
					w		w	w	w	w	w	w	w	w	w

Bit	Field	Description
31: 11	Reserved	保留，必须保持复位值
10	CLR_SMB_ALT	写该位清零与 I2C_SMB_RAWISR 对应的中断
9	Reserved	保留，必须保持复位值
8	CLR_PEC_NACK	写该位清零与 I2C_SMB_RAWISR 对应的中断
7	CLR_ARP_ASSGN	写该位清零与 I2C_SMB_RAWISR 对应的中断
6	CLR_ARP_UDID	写该位清零与 I2C_SMB_RAWISR 对应的中断
5	CLR_ARP_RST	写该位清零与 I2C_SMB_RAWISR 对应的中断
4	CLR_ARP_PRE	写该位清零与 I2C_SMB_RAWISR 对应的中断
3	CLR_ARP_NOTIFY	写该位清零与 I2C_SMB_RAWISR 对应的中断
2	CLR_ARP_QUICK	写该位清零与 I2C_SMB_RAWISR 对应的中断
1	CLR_MST_TMO	写该位清零与 I2C_SMB_RAWISR 对应的中断
0	CLR_SLV_TMO	写该位清零与 I2C_SMB_RAWISR 对应的中断

32.4.48 I2C_OPT_SAR 可选从地址寄存器

偏移地址：0xD8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	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: 7	Reserved	保留，必须保持复位值
6: 0	ADDR	I2C 接口工作在 SMBus 从模式时的从地址

32.4.49 I2C_SMB_UDID_LSB SMBus UDID LSB 寄存器

偏移地址：0xDC

复位值：0xFFFF FFFF

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

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

Bit	Field	Description
31: 0	SMB_UDID_LSB	配置 SMBus UDID[31:0]

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.50 I2C_SMB_UDID_MSB0 SMBus UDID MSB 寄存器 0

偏移地址：0xE0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMB_UDID_MSB0															

rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB_UDID_MSB0															
rw															

Bit	Field	Description
31: 0	SMB_UDID_MSB0	配置 SMBus UDID[63:32]

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.51 I2C_SMB_UDID_MSB1 SMBus UDID MSB 寄存器 1

偏移地址：0xE4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMB_UDID_MSB1															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB_UDID_MSB1															
rw															
Bit	Field					Description									
31: 0	SMB_UDID_MSB1					配置 SMBus UDID[95:64]									

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.52 I2C_SMB_UDID_MSB2 SMBus UDID MSB 寄存器 2

偏移地址：0xE8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMB_UDID_MSB2															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB_UDID_MSB2															
rw															
Bit	Field					Description									
31: 0	SMB_UDID_MSB2					配置 SMBus UDID[127:96]									

注：I2C_ENR.ENABLE 为 0 时该寄存器可写。

32.4.53 I2C_SLVMASK 从地址掩码寄存器

偏移地址: 0xEC

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

32.4.54 I2C_SLVRCVADDR 从器件接收地址寄存器

偏移地址: 0xF0

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	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

33 FlexCAN 控制器局域网

33.1 介绍

FlexCAN 模块是一个通信控制器，遵循 ISO 11898-1 标准和 CAN 2.0B 协议规范。CAN 协议主要被设计用作车载串行总线，在以下方面满足规范要求：

- 实时处理
- 车辆在电磁干扰环境下的可靠操作
- 成本效益
- 带宽要求

该模块支持标准和扩展帧，具有非常灵活的用于传输和接收的邮箱系统。邮箱系统由 16 个报文缓冲区（MB）组成。报文缓冲区用于存储配置、控制数据、时间戳、报文 ID 以及数据（参见章节“报文缓冲区结构”）。和 MB 相对应的内存空间可以通过配置支持 FIFO 接收机制，该机制能够结合 ID 过滤表（高达 40 个扩展 ID 或 80 个标准 ID 或 160 个 8 位 ID）检测接收帧，并为高达 16 个 ID 表元素提供了私有掩码寄存器。

FlexCAN 支持 FIFO 和邮箱同时接收。对于邮箱接收，通过匹配算法可以将接收到的帧只存储到具有相同 ID 的 MB 中。掩码机制能够将 MB 中设置的 ID 与一系列接收帧的 ID 进行匹配。传输时，仲裁算法基于报文的 ID（提供本地优先级可选）或 MB 的编号来决定待传输 MB 的优先级。

33.2 主要特征

- 完全支持 CAN 2.0B 协议：
 - a) 标准帧
 - b) 扩展帧
 - c) 0~8 字节数据长度
 - d) 可编程比特率，最高 1Mbps
 - e) 基于内容的寻址方式
 - 符合 ISO 11898-1 标准
 - 每个邮箱都可配置为接收或发送，均支持标准和扩展报文
 - 每个邮箱都有私有接收掩码寄存器
 - 全功能的 Rx FIFO，存储容量多达 6 帧，自动进行内部指针处理且支持 DMA
 - 传输中止功能
 - 支持只听模式

- 可编程的回环模式，支持自测试
- 可编程的传输优先级机制：最低 ID、最小邮箱编号或最高优先级
- 基于 16 位自由运行计时器的时间戳机制
- 全局网络时间（由特定的报文进行同步）
- 中断掩码
- 高优先级仲裁机制，延迟时间短
- 远程帧可以由软件处理或自动处理
- Tx 邮箱状态（最低优先级缓冲区或空缓冲区）
- ID 接收过滤命中指示（IDHIT）寄存器
- 发送报文的 CRC 状态
- Rx FIFO 全局掩码寄存器
- 匹配过程中邮箱或 Rx FIFO 优先级可选
- 高效的 Rx FIFO ID 过滤功能，高达 40 个 ID 过滤表元素

33.3 功能描述

33.3.1 功能框图

FlexCAN 的功能框图如下，包括用于存储报文缓冲区（MB）、接收全局掩码寄存器、接收私有掩码寄存器、接收（Rx）FIFO 过滤器以及接收 FIFO 标识符过滤器的内存。

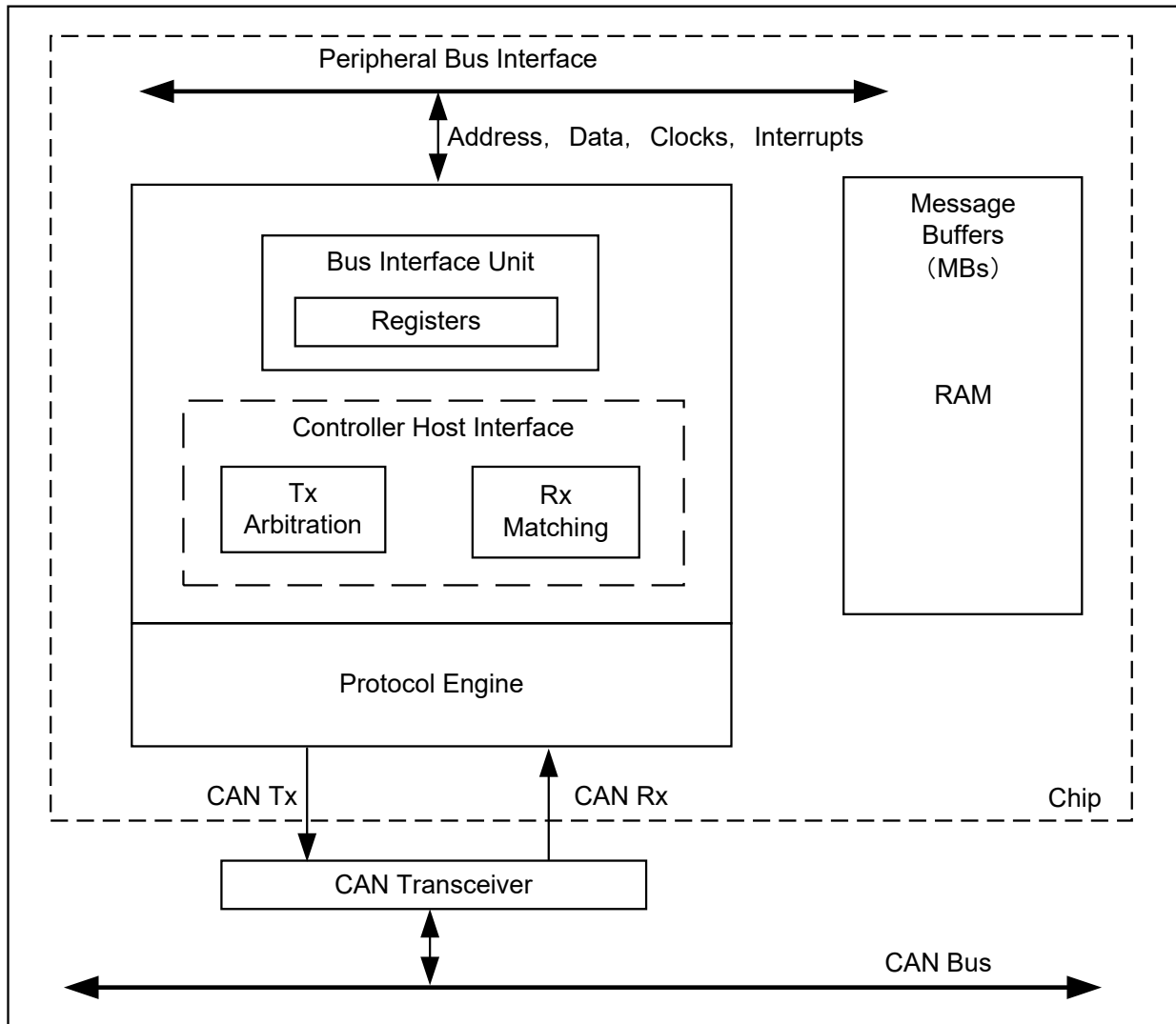


图 33-1 FlexCAN 框图

协议引擎（PE）子模块管理 CAN 总线上的串行通信：

- 请求存取 RAM 接收和传输帧
- 验收接收到的报文
- 进行错误处理

控制器主机接口（CHI）子模块负责选择接收和传输的报文缓冲区，以及对报文的仲裁和 ID 匹配算法。

总线接口单元（BIU）子模块控制内部接口总线的访问，建立与 CPU 和其他模块的连接。时钟、地址和数据总线、中断输出、DMA 都通过 BIU 进行访问。

33.3.2 工作模式

FlexCAN 具有以下 4 种功能模式：

- 正常模式

- 冻结模式
- 回环模式
- 只听模式

33.3.2.1 正常模式（用户或管理员模式）

FlexCAN 收发帧，正常处理错误，CAN 协议的功能全部使能。用户模式和管理员模式在访问一些受限的控制寄存器时有所不同，参见表格“寄存器访问和复位信息”。

33.3.2.2 冻结模式

使能冻结模式（置位 CAN_MCR.FRZ）后，可通过以下方式之一请求进入冻结模式：

- 非低功耗模式下，置位 CAN_MCR.HALT
- 非低功耗模式下，芯片进入调试模式

FlexCAN 硬件置位 CAN_MCR.FRZACK 以响应上述请求。当请求和响应的条件都满足时，CPU 认为模块处于冻结模式。该模式下收发帧停止，与 CAN 总线的同步丢失。

请求冻结模式时，FlexCAN 执行以下操作：

- 等待模块处于间歇（Intermission）、被动错误、总线关闭或空闲状态
- 等待所有内部活动（仲裁、匹配、移入和移出等）完成，待处理的移入不会阻止模块进入冻结模式
- 忽略 Rx 引脚的输入，驱动 Tx 引脚为隐性
- 停止预分频器，由此中止所有 CAN 协议活动
- 允许对错误计数寄存器执行写操作（该寄存器在其他模式下为只读）
- 置位 CAN_MCR.NOTRDY 和 CAN_MCR.FRZACK

请求冻结模式后，用户在执行任何其他操作前，必须等待 CAN_MCR.FRZACK 被置位。否则，FlexCAN 的动作无法预测。冻结模式下，所有内存映射的寄存器都可以被访问（CAN_CTRL1.CLKSRC 除外，该位为只读）。

退出冻结模式的方式有以下两种：

- CPU 清零 CAN_MCR.FRZ
- 芯片退出调试模式和/或 CAN_MCR.HALT 被清零

当 PE 识别到冻结模式请求无效后，CAN_MCR.FRZACK 被硬件清零。退出冻结模式时，FlexCAN 尝试通过等待 11 个连续隐性位与 CAN 总线重新同步。

33.3.2.3 回环模式

CAN_CTRL1.LPB 被置位时，FlexCAN 将进入回环模式。该模式下，Rx 引脚的输入被忽略，Tx 引

脚的输出处于隐性状态（逻辑 1）。发送器的输出在内部反馈给接收器的输入，可用于模块自测试。发送报文时，FlexCAN 将自身发送的报文视为从远程节点接收到的报文。为保证能正确接收到自身发送的报文，FlexCAN 会忽略 CAN 帧应答场 ACK 槽（ACK SLOT）发送的位。发送和接收中断均会产生。

33.3.2.4 只听模式

CTRL1.LOM 被置位时，FlexCAN 将进入只听模式。该模式下，FlexCAN 工作在 CAN 错误被动模式，所有的错误计数器都被冻结。发送被禁止，只能接收被其他 CAN 节点应答的报文。如果 FlexCAN 检测到一个未被应答的报文，则标记为 BIT0 错误（不改变接收错误计数值）。

FlexCAN 具有以下低功耗模式：

- 模块禁止模式

33.3.2.5 模块禁止模式

一种低功耗模式，通常用于临时停止 FlexCAN 模块，不耗电。CPU 置位 CAN_MCR.MDIS 以请求模块禁止，FlexCAN 置位 CAN_MCR.LPMACK 响应请求。当请求和响应的条件都满足时，CPU 认为模块处于禁止模式。

如果在冻结模式下禁止 FlexCAN，则模块请求停止 PE 和 CHI 子模块的时钟，并置位 CAN_MCR.LPMACK 和 CAN_MCR.FRZACK。

如果在传输或接收期间禁止 FlexCAN，则模块执行以下操作：

- 等待模块处于总线关闭或空闲状态，或等待 Intermission 的第三位为隐性
- 等待所有内部活动（仲裁、匹配、移入和移出等）完成，不考虑待处理的移入
- 忽略 Rx 引脚的输入，驱动 Tx 引脚为隐性
- 关闭 PE 和 CHI 子模块的时钟
- 置位 CAN_MCR.NOTRDY 和 CAN_MCR.LPMACK

总线接口单元继续运行，允许 CPU 访问内存映射寄存器（除 RXMGMASK、RX14MASK、RX15MASK 和 RXFGMASK）。模块禁止模式下，不能访问 RAM 内的寄存器 CAN_RXFIR、MBs、CAN_RXIMRn 和预留字。

清零 CAN_MCR.MDIS 可以退出模块禁止模式。FlexCAN 请求恢复时钟，并在 PE 子模块识别到 CPU 取消禁止模式后清零 CAN_MCR.LPMACK。

33.3.3 传输流程

传输 CAN 帧时，CPU 必须为传输准备报文缓冲区，步骤如下：

- 1) 检查相应的中断标志位是否被置位，若被置位则将其清零
- 2) 如果 MB 为激活状态（传输待处理），则向控制和状态（C/S）字的代码（CODE）字段写入中止

代码（0b1001）以请求中止传输。通过轮询 CAN_IFLAG1 寄存器，等待相应的 IFLAG 置位；或使能相应的 IMASK 位，通过中断请求来实现。接着，回读 CODE 字段以检查传输是否被中止（参见章节“传输中止机制”），并清零相应的中断标志

注：如果一个报文缓冲区既能参与匹配又能参与仲裁过程，则称该报文缓冲区处于激活状态（ACTIVE）。CODE 字段为 0b0000 的接收 MB 处于非激活状态（参见表格“接收缓冲区代码”）；代码为 0b1000 或 0b1001 的发送 MB 也处于非激活状态（参见表格“发送缓冲区代码”）。

3) 写入 ID，包括本地优先级（若 CAN_MCR.LPRIOEN 被置位）

4) 写入数据字节

5) 根据需要配置 C/S：ID 类型（IDE）、远程传输请求（RTR）、数据长度（DLC），写入代码 0xC 以激活 MB 进行 CAN 帧传输

激活的 MB 将参与仲裁过程，并最终依照优先级进行传输。用于传输的 MB 中存储的 DLC 值大于相应的 MB 负载大小时，FlexCAN 使用 0xCC 添加必要的字节数，以完成预期的 DLC。

传输成功结束时：

- 自由运行计时器的值被写入时间戳字段
- C/S 的 CODE 字段被更新
- CAN_CRC 寄存器被更新
- CAN_IFLAG1 的状态标志置位；若相应的中断使能，则产生中断

传输后的新 CODE 字段取决于用于激活 MB 的代码（参见表格“发送缓冲区代码”）。

中止功能使能（CAN_MCR.AEN 置位）时，邮箱（被配置为发送缓冲区）的中断标志被置位后，CPU 无法更新邮箱。因此，开始为新传输或接收准备该 MB 前，CPU 必须清零相应的 CAN_IFLAG1 位。

33.3.4 仲裁过程

仲裁过程按照邮箱编号由低到高进行扫描，搜索用于下一次传输的邮箱。该邮箱存放了下一次要传输的报文，被称为仲裁获胜邮箱。下列事件会触发仲裁过程：

- CAN 帧的 CRC 字段，起始点取决于 CAN_CTRL2.TASD 字段的值
- CAN 帧的错误界定符（Error Delimiter）字段期间
- CAN 帧的过载界定符（Overload Delimiter）字段期间
- 获胜邮箱非激活，且 CAN 总线尚未到达 Intermission 字段首位
- CPU 写获胜邮箱的 C/S，且 CAN 总线尚未到达 Intermission 字段的首位
- CHI 处于空闲状态，且 CPU 写任意 MB 的 C/S
- FlexCAN 退出总线关闭状态
- 离开冻结模式或低功耗模式

如果在 CAN 总线到达 Intermission 字段的首位前，仲裁过程没能评估所有邮箱，则临时获胜邮箱无效，且 FlexCAN 不会参与下一次的 CAN 总线竞争。仲裁过程在扫描结束时，会根据 CAN_CTRL1.LBUF

和 CAN_MCR.LPRIOEN 的配置，从激活的 Tx 邮箱中选择获胜邮箱。

33.3.4.1 最小邮箱编号优先

CAN_CTRL1.LBUF 被置位时，CAN_MCR.LPRIOEN 位无效，搜索到的第一个（最小编号）激活的 Tx 邮箱为仲裁获胜邮箱。

33.3.4.2 最高优先级邮箱优先

CAN_CTRL1.LBUF 为 0 时，仲裁过程根据最高优先级搜索激活的 Tx 邮箱。当多个外部节点同时竞争总线时，该邮箱的帧赢得 CAN 总线仲裁的几率更高。

用于此次仲裁的位序列被称为邮箱仲裁值。在所有 Tx 邮箱中，优先级最高的 Tx 邮箱仲裁值最小。如果两个或多个邮箱具有相同的仲裁值，则编号最小的邮箱为仲裁获胜邮箱。

仲裁值的构成取决于 CAN_MCR.LPRIOEN 的配置。

- 本地优先级禁止

如果 CAN_MCR.LPRIOEN 为 0，则本地优先级无效，根据 CAN 帧的位传输顺序确立仲裁值。

表 33-1 本地优先级无效时仲裁值的构成

格式	邮箱仲裁值（32 位）				
标准（IDE = 0）	标准 ID （11 位）	RTR （1 位）	IDE （1 位）	— （18 位）	— （1 位）
扩展（IDE = 1）	扩展 ID [28:18] （11 位）	SRR （1 位）	IDE （1 位）	扩展 ID [17:0] （18 位）	RTR （1 位）

- 本地优先级使能

如果 CAN_MCR.LPRIOEN 被置位，则本地优先级使能，邮箱的 PRIO 字段纳入仲裁值的最左侧。

表 33-2 本地优先级使能时仲裁值的构成

格式	邮箱仲裁值（35 位）					
标准（IDE = 0）	PRIO （3 位）	标准 ID （11 位）	RTR （1 位）	IDE （1 位）	— （18 位）	— （1 位）
扩展（IDE = 1）	PRIO （3 位）	扩展 ID [28: 18] （11 位）	SRR （1 位）	IDE （1 位）	扩展 ID [17: 0] （18 位）	RTR （1 位）

PRIO 字段为仲裁值的最高位，因此 PRIO 值低的邮箱优先级更高，与剩余仲裁值无关。

注：PRIO 字段不属于 CAN 总线帧，仅影响内部仲裁过程。

33.3.4.3 仲裁完成

找出仲裁获胜邮箱后，其内容会被复制到一个名为 Tx 串行报文缓冲区（Tx SMB）的隐藏辅助 MB 中，该 MB 与普通 MB 结构相同，但用户无法访问。此复制操作称为移出（Move Out）。移出完成后，无法写相应 MB 的 C/S（CAN_MCR.AEN 被置位时）。以下事件可使写操作恢复：

- MB 被传输后，且相应的 IFLAG 位被 CPU 清零
- FlexCAN 进入冻结模式或总线关闭
- FlexCAN 丢失总线仲裁或出现传输错误

当 CAN 总线上出现第一个机会窗口时，Tx SMB 内的报文依据 CAN 协议规则进行传输。

33.3.4.4 仲裁启动和停止条件

仲裁在下列情况下被触发：

- Rx 帧和 Tx 帧期间，从 CAN CRC 字段至帧结束。配置 CAN_CTRL2.TASD 可优化仲裁起点
- CAN 总线关闭状态，从 TX_ERR_CNT 的值为 124 到 128。配置 CAN_CTRL2.TASD 可优化仲裁起点
- 总线空闲状态，CPU 写 C/S。第一个 C/S 写操作启动仲裁过程，同一个仲裁期间的第二个 C/S 写操作会重启流程。如果执行了其他 C/S 写操作，则 Tx 仲裁进程被挂起。如果在仲裁过程结束后没有获胜者，则 Tx 仲裁机启动一个新的仲裁过程。如果存在待处理的仲裁，且总线空闲状态启动，则仲裁过程被触发。此时，总线空闲状态下的第一个和第二个 C/S 写操作不会重启仲裁过程。在等待总线空闲（Wait For Bus Idle）状态并且下一个状态为空闲时，可能没有足够的时间完成仲裁。此情况下，扫描不会中断，而是在总线空闲状态下完成。仲裁期间，C/S 写操作不会导致仲裁重启
- 仲裁获胜邮箱在有效仲裁期间失活
- 退出冻结模式（等待总线空闲状态的首位）。如果在等待总线空闲状态下进行重同步，则仲裁过程会重启

仲裁在下列情况下停止：

- 所有邮箱都被扫描过
- 最低缓冲区（lowest buffer）功能使能时，发现一个激活的 Tx 邮箱
- 仲裁获胜邮箱在任何仲裁期间失活或中止
- 没有足够的时间完成 Tx 仲裁过程。此情况下，仲裁过程被挂起
- 总线错误或过载
- 总线空闲状态下请求低功耗或冻结模式

仲裁在下列情况下被挂起：

- 无法及时完成仲裁过程
- 仲裁期间执行 C/S 写操作，如果执行写操作的 MB 编号小于 Tx 仲裁指针
- 没有进行中的 Tx 仲裁过程时，执行任何 C/S 写操作
- Rx 匹配刚刚更新了 Rx 代码至 Tx 代码
- 进入总线关闭状态

仲裁期间写 C/S 会产生下列影响：

- 如果仲裁获胜邮箱执行 C/S 写操作，则立即重启新流程
- 如果在编号比 Tx 仲裁指针大的 MB 中执行 C/S 写操作，则进行中的仲裁过程将正常扫描该 MB

33.3.5 接收流程

为确保邮箱正常接收帧，CPU 必须执行下列步骤以准备接收邮箱：

1) 如果邮箱 (Tx 或 Rx) 为激活状态，需要使邮箱失活 (参见章节“邮箱失活”)，最好选用安全的失活方式 (参见章节“传输中止机制”)

2) 写 ID

3) 向 C/S 的代码字段写入空代码 (0b0100) 以激活邮箱

MB 激活后，能够接收与过滤器匹配的帧。接收成功结束时，通过以下移入流程 (参见章节“移入”) 更新邮箱：

1) 接收到的数据字段 (最多 8 字节) 被存储

2) 接收到的 ID 段被存储

3) 帧标识符字段第二位时自由运行计时器的值，被写入邮箱的时间戳字段

4) 接收到的 SRR、IDE、RTR 和 DLC 字段被存储

5) C/S 的代码字段被更新 (参见表格“接收缓冲区代码”和“发送缓冲区代码”)

6) 中断标志寄存器的状态位被置位。如果相应的中断掩码使能，则生成中断

建议 CPU 按照以下方式处理 (读取) 邮箱中接收到的帧：

1) 读取邮箱的 C/S

2) 检查 BUSY 位是否为 0，表明邮箱被上锁。重复步骤 1) 直到该位清零。参见章节“邮箱上锁机制”

3) 读取邮箱内容。邮箱上锁时，移入流程不会更改其内容 (参见章节“移入”)

4) 确认 CAN_IFLAG1 寄存器的相应标志位

5) 读取自由运行计时器以解锁邮箱

CPU 应该通过检查 CAN_IFLAG1 寄存器中，特定邮箱的状态标志位来轮询帧接收，而不是轮询该邮箱的代码字段。轮询代码字段无效，因为一旦接收到帧，CPU 处理邮箱 (解锁邮箱后读取 C/S)，代码字段不会转变为 EMPTY，而是仍然为 FULL (如表格“接收缓冲区代码”所示)。如果 CPU 试图在未提前安全地使邮箱失活的情况下，在读取邮箱后通过强制向 C/S 写 EMPTY 来解决这一问题，则与相应邮箱过滤器相匹配的新接收帧可能丢失。因此，禁止通过直接读取邮箱的 C/S 进行轮询，而应该读取 CAN_IFLAG1 寄存器。

接收帧的标识符始终存储在匹配的邮箱中。因掩码出现的匹配，相应邮箱中 ID 段的内容可能会发生改变。

当 CAN_MCR.SRXDIS 被置位时，FlexCAN 不会存储自身传输的任何 MB 内的帧，即使模块包含一个匹配的 Rx 邮箱，且不会产生中断标志或中断信号；当 CAN_MCR.SRXDIS 为 0 时，如果有匹配的 Rx 邮箱，FlexCAN 可以接收到自身发送的帧。

为了能够通过 Rx FIFO 接收 CAN 帧，CPU 必须在冻结模式下使能并配置 Rx FIFO（参见章节“Rx FIFO”）。接收到 Rx FIFO 有可用帧的中断时（CAN_IFLAG1.BUF5I），CPU 应通过下列程序处理接收到的帧：

- 1) 读取 C/S（可选，仅在 IDE 和 RTR 位使用掩码时才是必须的）
- 2) 读取 ID 段（可选，仅在使用掩码时才是必须的）
- 3) 读取数据字段
- 4) 读取 CAN_RXFIR 寄存器（可选）

5) 通过向 CAN_IFLAG1.BUF5I 写 1 来清零 Rx FIFO 有可用帧的中断（必须，释放 MB 并允许 CPU 读取下一个 Rx FIFO 条目）

当 CAN_MCR.DMA 被置位时，Rx FIFO 接收到帧，CAN_IFLAG1.BUF5I 不生成 CPU 中断，而是生成 DMA 请求（参见章节“DMA”）。Rx FIFO 区域的 CAN_IMASK1 未被使用。

DMA 通过下列程序处理接收到的帧：

- 1) 读取 C/S（读取 0x80 地址，可选）
- 2) 读取 ID 段（读取 0x84 地址，可选）
- 3) 读取所有数据字节（从 0x88 地址开始读取，可选）
- 4) 读取最后一个数据字节（读取 0x8C 地址，必须）

33.3.6 匹配过程

匹配过程扫描 MB 内存，搜索与从 CAN 总线上接收到的帧具有相同 ID 的 Rx MB。如果 Rx FIFO 使能，则可在邮箱和 FIFO 过滤器中选择扫描优先级。匹配从较低编号的 MB 向较高编号的 MB 进行。如果在第一种结构内未发现匹配，则继续扫描下一个结构。如果 FIFO 已满，则匹配算法在 FIFO 区域以外寻找匹配的 MB。

正在被接收的帧将存储在名为 Rx 串行报文缓冲区（Rx SMB）的隐藏辅助 MB 内，该 MB 与普通 MB 结构相同。

匹配过程的开始点取决于以下条件：

- 如果接收到的是远程帧，开始点为帧的 CRC 字段
- 如果接收到的是 DLC 字段为 0 的数据帧，开始点为帧的 CRC 字段
- 如果接收到的是 DLC 字段不为 0 的数据帧，开始点为帧的数据字段

如果在 FIFO 或邮箱中发现匹配的 ID，Rx SMB 的内容会通过移入流程传输至 FIFO 或匹配的邮箱。如果检测到任何 CAN 协议错误，则在接收结束时没有匹配结果传输至 FIFO 或匹配的邮箱。

匹配过程扫描 Rx FIFO（使能时）和激活的 Rx 邮箱（代码字段为 EMPTY、FULL、OVERRUN 或 RANSWER）内的所有匹配元素，以寻找一次和 Rx SMB 内匹配元素的成功比较。与成功比较的匹配元

素相关联的接收结构（Rx FIFO 或邮箱）称为匹配结构。扫描结束时，从匹配结构中选择匹配获胜者，具体取决于下表所列条件。

表 33-3 匹配架构

结构	SMB. RTR	CTRL2. RRS	CTRL2. EACEN	MB.IDE	MB.RTR	MB.ID ⁽¹⁾	MB.CODE
邮箱	0	—	0	cmp ⁽²⁾	cmp_msk ⁽³⁾	cmp_msk ⁽⁴⁾	EMPTY、FULL、OVERRUN
邮箱	0	—	1	cmp_msk	cmp_msk	cmp_msk	EMPTY、FULL、OVERRUN
邮箱	1	0	—	cmp	no_cmp	cmp	RANSWER
邮箱	1	1	0	cmp	no_cmp	cmp_msk	EMPTY、FULL、OVERRUN
邮箱	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY、FULL、OVERRUN
Rx FIFO ⁽⁵⁾	—	—	—	cmp_msk	cmp_msk	cmp_msk	—

1) 对于邮箱结构，SMB.IDE 被置位时 ID 为 29 位（标准 ID + 扩展 ID）；SMB.IDE 为 0 时，ID 为 11 位（标准 ID）。对于 Rx FIFO 结构，ID 取决于 CAN_MCR.IDAM

2) cmp: 比较 Rx SMB 和 MB 的内容，忽略掩码

3) no_cmp: 不比较 Rx SMB 和 MB 的内容

4) cmp_msk: 比较 Rx SMB 和 MB 的内容，考虑掩码

5) 当 CAN_MCR.IDAM 为格式 C 时，不考虑 SMB.IDE 和 SMB.RTR

满足以下任一条件时，接收结构为自由接收：

- 邮箱代码字段为 EMPTY
- 邮箱代码字段为 FULL 或 OVERRUN，且该邮箱已被处理过（CPU 读取 C/S 并解锁，参见章节“邮箱上锁机制”）
- 邮箱代码字段为 FULL 或 OVERRUN，且该邮箱已失活（参见章节“邮箱失活”）
- Rx FIFO 未滿

邮箱和 Rx FIFO 的扫描顺序为，从最小编号的匹配元素到最大编号的匹配元素。

匹配获胜邮箱受 CAN_MCR.IRMQ 位影响。如果该位为 0，则匹配获胜邮箱为第一个匹配的邮箱，无论是否为自由接收邮箱；如果该位被置位，则根据以下优先级来选择匹配获胜邮箱：

- 1) 第一个自由接收的匹配邮箱
- 2) 最后一个非自由接收的匹配邮箱

可通过 CAN_CTRL2.MRP 位选择扫描优先级：邮箱优先或 Rx FIFO 优先。选择 Rx FIFO 优先时：

- 如果 Rx FIFO 是匹配结构且自由接收，则 Rx FIFO 为匹配获胜者，不再扫描邮箱
- 如果 Rx FIFO 不是匹配结构或不是自由接收，则在上述邮箱中搜索匹配获胜者

选择邮箱优先时：

- 如果找到自由接收的匹配邮箱，则该邮箱为匹配获胜者，不再扫描 Rx FIFO
- 如果未找到匹配邮箱，则在 Rx FIFO 中搜索匹配获胜者

如果在上述两种条件均未能满足的情况下，找到了一个非自由接收的匹配邮箱，则匹配获胜者由 CAN_MCR.IRMQ 位决定：

- CAN_MCR.IRMQ 为 0 时，匹配获胜者为第一个匹配的邮箱
- CAN_MCR.IRMQ 被置位时，如果 Rx FIFO 为自由接收的匹配结构，则匹配获胜者是 Rx FIFO；否则，匹配获胜者是最后一个非自由接收的匹配邮箱

下表总结了所有匹配可能：

表 33-4 匹配可能性及由此产生的接收结构

RFEN	IRMQ	MRP	MB 匹配	FIFO 匹配	接收结构	描述
无 FIFO，只有 MB，匹配始终为 MB 优先：						
0	0	X ⁽¹⁾	无 ⁽²⁾	— ⁽³⁾	无	因无匹配而丢失帧
0	0	X	自由 ⁽⁴⁾	—	First MB	
0	1	X	无	—	无	因无匹配而丢失帧
0	1	X	自由	—	First MB	
0	1	X	非自由	—	Last MB	溢出
FIFO 使能，FIFO 中无匹配，如同 FIFO 不存在：						
1	0	X	无	无 ⁽⁵⁾	无	因无匹配而丢失帧
1	0	X	自由	无	First MB	
1	1	X	无	无	无	因无匹配而丢失帧
1	1	X	自由	无	First MB	
1	1	X	非自由	无	Last MB	溢出
FIFO 使能，队列禁止：						
1	0	0	X	非满 ⁽⁶⁾	FIFO	
1	0	0	无	满 ⁽⁷⁾	无	因 FIFO 满而丢失帧（FIFO 溢出）
1	0	0	自由	满	First MB	
1	0	0	非自由	满	First MB	
1	0	1	无	非满	FIFO	
1	0	1	无	满	无	因 FIFO 满而丢失帧（FIFO 溢出）
1	0	1	自由	X	First MB	
1	0	1	非自由	X	First MB	溢出
FIFO 使能，队列使能：						
1	1	0	X	非满	FIFO	
1	1	0	无	满	无	因 FIFO 满而丢失帧（FIFO 溢出）
1	1	0	自由	满	First MB	
1	1	0	非自由	满	Last MB	溢出
1	1	1	无	非满	FIFO	
1	1	1	自由	X	First MB	
1	1	1	非自由	非满	FIFO	
1	1	1	非自由	满	Last MB	溢出

1) X: “don't care”条件

2) MB 匹配“无”表示帧没有匹配任何 MB（自由接收或非自由接收）

- 3) —: 禁止条件
- 4) MB 匹配“自由”表示帧至少匹配一个自由接收 MB, 无论是否匹配非自由接收 MB
- 5) FIFO 匹配“无”表示帧没有匹配任何 Rx FIFO 过滤器, 如同 Rx FIFO 不存在 (CAN_CTRL2.RFEN = 0)
- 6) FIFO 匹配“未满足”表示帧已匹配 FIFO 过滤器, 且有空余接收空间
- 7) FIFO 匹配“满足”表示帧已匹配 FIFO 过滤器, 但 FIFO 没有空余接收空间, 无法存储

如果在匹配过程中发生了不安全的邮箱失活操作 (参见章节“邮箱失活”), 且失活的邮箱为临时匹配获胜邮箱, 则临时匹配获胜邮箱无效。匹配元素扫描不会停止或重启, 而是正常继续。后果是, 当前的匹配过程继续运行, 邮箱失活前的匹配元素比较仿佛不存在, 因此可能会丢失一个报文。

例如: 假设 Rx FIFO 被禁用, IRMQ 被置位, 2 号和 5 号 MB 具有相同的 ID, FlexCAN 开始接收具有该 ID 的报文。当第一个报文到达时, 首先匹配到 2 号 MB, 该 MB 的代码为 EMPTY, 因此报文存储于此。当第二条报文到达时, 再次匹配到 2 号 MB, 由于该 MB 为非自由接收, 匹配算法继续寻找, 找到 5 号 MB 后将报文存储于此。如果又有一条具有相同 ID 的报文到达, 匹配算法发现没有自由接收的匹配 MB, 因而决定覆盖最后一个匹配 MB, 即 5 号 MB。覆盖的同时, 设置 MB 代码字段为 OVERRUN。

为多个 MB 编程相同的 ID, 能够实现队列接收 (除全功能 FIFO 外), 以便 CPU 有更多时间处理 MB。CPU 通过检查 MB 的时间戳字段可以确定报文到达的顺序。

FlexCAN 的每个 MB 都支持私有掩码 (CAN_RXIMRn 寄存器), 使用掩码可以匹配一系列的 ID。在匹配算法阶段, 如果掩码被置位, 则比较相应的 ID 位; 如果掩码位为 0, 则不比较相应的 ID 位。私有掩码寄存器位于 RAM 中, 因此复位后不会被初始化, 并且只能在 FlexCAN 处于冻结模式时被编程。

FlexCAN 还支持一个仅有四个掩码寄存器 (CAN_RXFGMASK、CAN_RXMGMASK、CAN_RX14MASK 和 CAN_RX15MASK) 的替代掩码方案, 该方案在 CAN_MCR.IRMQ 为 0 时启用。

33.3.7 转移流程

转移流程有两种类型: 移入和移出。

33.3.7.1 移入

移入流程将 Rx SMB 接收到的报文复制到匹配的 Rx 邮箱或 FIFO 中。如果移入的目标是 Rx FIFO, 则报文的属性也会被复制至 CAN_RXFIR FIFO 内。每个 Rx SMB 均有独立的移入流程, 但在给定的时间内只执行一个移入操作。只有当 Rx SMB 中的报文存在对应的匹配获胜者 (参见章节“匹配过程”), 且以下条件都满足时, 才开始移入:

- CAN 总线已经到达或让位于:
 - a) 靠近携带 Rx SMB 信息帧的 Intermission 字段的第二位, 或者
 - b) 靠近携带 Rx SMB 信息帧的过载帧的第一位
- 没有正在进行的匹配过程

- 目标邮箱未被 CPU 锁定
- 其他 Rx SMB 没有正在进行的移入流程。如果同时启动一个以上的移入流程，则两个都可以执行，并用最新的替代最旧的

本章节使用“待处理的移入”表示尚未满足上述所有条件的、将要开始的移入流程

如果满足以下任一条件，则移入流程被取消，且 Rx SMB 能够接收其他报文：

- CAN 总线到达携带报文的帧附近的 Intermission 字段首位，且相应的匹配过程完成后，目标邮箱失活
- 先前有一个待处理的移入，并且该移入具有相同的目标邮箱
- Rx SMB 正在接收 FlexCAN 传输的帧，且自接收禁止（CAN_MCR.SRXDIS 被置位）
- 检测到任何 CAN 协议错误

注：如果 FlexCAN 进入冻结或低功耗模式，待处理的移入流程不会被取消，而是处于等待状态。等待从冻结或低功耗模式退出并被解锁。如果 MB 在冻结模式下被解锁，则移入流程即刻执行。

FlexCAN 按照以下步骤执行移入流程：

- 1) 如果报文的目标是 Rx FIFO，则将 IDHIT 移入 RXFIR FIFO
- 2) 根据为 Rx 存储元素选定的有效负载数，从 Rx SMB 读取所有数据字
- 3) 根据为 Rx 存储元素选定的有效负载数，向 Rx 邮箱写入所有数据字。如果存储元素的数据数小于报文 DLC 字段的有效负载数，则有效负载被截断，不符合目标大小的高阶字节丢失
- 4) 从 Rx SMB 读取 C/S 和 ID
- 5) 将 C/S 和 ID 写入 Rx 邮箱，并更新代码字段

移入流程并不是自动的，该流程会因目标邮箱失活（参见章节“邮箱失活”）而被立即取消。此情况可能导致邮箱被部分更新，因而出现不一致。例外情况是，如果移入目标为 Rx FIFO，则流程不能被取消。当移入流程正在进行时，会置位目标报文缓冲区的 BUSY 位（代码字段的最低有效位），以警告 CPU 报文缓冲区的内容暂时不一致。

33.3.7.2 移出

移出流程是指，传输报文可用时，将报文从 Tx 邮箱复制到 Tx SMB 的过程（参见章节“仲裁过程”）。移出在以下情况发生：

- Intermission 字段的首位
- 总线关闭期间，TX 错误计数器的值在 124 到 128 之间
- 总线空闲状态期间
- 等待总线空闲状态期间

移出流程并不是自动的。总线空闲状态以外，只有 CPU 具有并发访问内存的优先级。在总线空闲状态，对于并发的内存访问，移出流程优先级最低。

33.3.8 数据一致性

为了保持数据的一致性以及 FlexCAN 的正确运行，CPU 必须遵守“传输流程”和“接收流程”所描述的规则。

33.3.8.1 传输中止机制

中止机制提供了一种安全的方式，以请求中止待处理的传输。如果传输被中止，或帧不能被中止而是被传输了，提供反馈机制通知 CPU。

中止传输必须满足两个主要条件：

- CAN_MCR.AEN 必须被置位
- CPU 的第一个动作必须是向 C/S 的代码字段写入中止代码（0b1001）

为传输配置的激活的 MB 在更新前必须先中止。如果向当前正在传输的邮箱，或已载入 Tx SMB 中用于传输的邮箱写中止代码，则写操作无效，传输不受干扰。但中止请求会被捕获并保持挂起，直到满足以下条件之一：

- FlexCAN 丢失仲裁
- 传输期间出现错误
- FlexCAN 进入冻结模式
- FlexCAN 进入总线关闭状态
- 有一个过载帧

如果上述条件均未满足，则正确传输 MB，并置位相应的中断标志（CAN_IFLAG1）；生成中断（如果使能）。当中断标志被置位时，中止请求自动清零；如果满足了上述条件之一，则帧不会被传输，中止代码被写入代码字段，相应的中断标志（CAN_IFLAG1）置位；生成中断（可选）。

如果 CPU 在内部传输开始前写入中止代码，则 MB 被更新，中断标志被置位。通过这种方式，CPU 只需读取中止代码即可确保 MB 安全失活。尽管 CAN_MCR.AEN 被置位且 CPU 写入了中止代码，但此时 MB 处于失活状态而且不会中止，因为传输尚未开始。中止邮箱的条件是中止请求被捕获且保持挂起状态，直到满足先前所述的条件之一。

中止流程总结如下：

- CPU 检查相应的 IFLAG 并清零
- CPU 向 C/S 的代码字段写入 0b1001
- CPU 等待相应的 IFLAG，以了解帧是被传输还是被中止
- CPU 读代码字段，以检查帧是被传输（代码 = 0b1000）或是被中止（代码 = 0b1001）
- 清除相应的 IFLAG，以允许 MB 被重新配置

33.3.8.2 邮箱失活

提供失活机制是为了保护邮箱不被 FlexCAN 内部流程所更新，从而允许 CPU 在更新后信赖邮箱数据的一致性，即使是在正常模式下。

只有当 MCR.AEN 位被清零时才能使能传输邮箱失活机制。

失活的邮箱在重新激活前，不会参与仲裁或匹配过程。参见章节“传输流程”和“接收流程”，以了解更多关于失活和重新激活邮箱的描述。

为了使邮箱失活，CPU 必须更新代码字段为 INACTIVE (0b0000 或 0b1000)。

因为 CPU 无法和 FlexCAN 内部流程同步更新代码字段，使邮箱失活可能会导致以下后果：

- 与失活的 Rx 邮箱过滤相匹配的总线帧，可能在未通知的情况下丢失，即使有其他具有相同过滤的邮箱
- 失活的 Tx 邮箱内包含报文的帧可能会在未设置相应 IFLAG 的情况下传输

为了确保安全失活并避免 Tx 邮箱产生上述后果，CPU 必须采用传输中止机制（参见章节“传输中止机制”）。

失活机制自动解锁邮箱（参见章节“邮箱上锁机制”）。

注：作为 Rx FIFO 一部分的报文缓冲区不能被失活。FlexCAN 的 Rx FIFO 区域没有写保护。当 RFEN 被置位时，CPU 必须保持 Rx FIFO 区域的数据一致性。

33.3.8.3 邮箱上锁机制

除邮箱失活外，FlexCAN 还有另外一种适用于接收流程的数据一致性机制。当 CPU 从代码为 FULL 或 OVERRUN 的 Rx MB 读取 C/S 时，FlexCAN 假定 CPU 想以自动的方式读取整个 MB，因此为该 MB 设置内部上锁标志。当 CPU 读取自由运行计时器（全局解锁操作），或读取其他 MB 的 C/S（无论代码如何）时，锁定解除。CPU 写 C/S 也能解锁 MB，但不推荐用于正常解锁，因为这种方式会取消待处理的转移，可能丢失一个接收到的报文。MB 上锁可以防止在 CPU 读取某个 MB 时将新帧写入到该 MB 中。

注：锁定机制仅适用于 Rx MB，该 MB 不属于 Rx FIFO 的一部分，且相应的代码不为 INACTIVE (0b0000) 或 EMPTY (0b0100)。Tx MB 不能被锁定。

例如，假定 Rx FIFO 被禁用，2 号和 5 号 MB 被编程为相同的 ID，且 FlexCAN 已接收到报文并将其存入两个 MB 中。假设 CPU 决定读取 5 号 MB，与此同时具有相同 ID 的其他报文到达。当 CPU 读取 5 号 MB 的 C/S 时，该 MB 被锁定。新报文到达且匹配算法发现没有自由接收的 MB，因此决定覆盖 5 号 MB。但该 MB 被上锁，新报文无法写入。新报文将继续留在 Rx SMB，等待 MB 被解锁，然后才能被写入 MB。如果又有一条具有相同 ID 的报文到达时，MB 没有及时解锁，则新的报文会覆盖掉 Rx SMB 中的报文，且 MB 的代码字段或错误和状态寄存器中，不会有报文丢失的提示。

当报文从 Rx SMB 移入 MB 时，代码字段的 BUSY 位被置位。如果 CPU 读取 C/S 并发现 BUSY 位被置位，则应该延缓访问 MB 直到 BUSY 位被清零。

注：如果 BUSY 位被置位或者 MB 为空，则读取 C/S 不会锁定 MB。

失活优先于上锁。如果 CPU 失活一个锁定的 Rx MB，则其锁定状态被忽略且 MB 在本轮匹配中被标记为无效。Rx SMB 中的任何未被处理的报文不会传输至 MB。当 CPU 读取自由运行计时器或其他 MB 的 C/S 时，MB 被解锁。锁定和解锁机制在正常模式和冻结模式下具有相同的功能。

正常或冻结模式下解锁会导致移入未处理的报文。如果解锁在低功耗模式下进行，则移入过程会被延后，且只能在模块回到正常或冻结模式时发生。

33.3.9 Rx FIFO

CAN_MCR.RFEN 被置位时 Rx FIFO 使能。Rx FIFO 有 6 条报文的深度。Rx FIFO 结构占有的内存区域（包括报文缓冲区和 Rx FIFO 引擎），参见章节“Rx FIFO 结构”。通过在 Rx FIFO 的输出接口重复地读取报文缓冲区结构，CPU 可以按照报文被接收的顺序进行读取。

当 Rx FIFO 中至少有一个帧可被读取时，CAN_IFLAG1.BUF5I（Rx FIFO 中有可用帧）被置位。使能相应的掩码位可生成中断。一旦接收到中断，CPU 能够读取报文（以报文缓冲区的方式访问 Rx FIFO 的输出）和 CAN_RXFIR 寄存器，之后清除中断。如果 Rx FIFO 中有更多的报文，则清除中断的行为，会将 FIFO 输出更新为下一条报文，并将 CAN_RXFIR 更新为该报文的属性，再次向 CPU 发出中断。否则，标志位会被清除。Rx FIFO 的输出仅在 CAN_IFLAG1.BUF5I 被置位时有效。

接收到新报文后，Rx FIFO 中未读取的报文数从 4 增加至 5 时，CAN_IFLAG1.BUF6I（Rx FIFO 警告）被置位，表明 Rx FIFO 将满。该标志位会一直保持置位，直到 CPU 将其清零。

因 Rx FIFO 满而导致新来的报文丢失时，CAN_IFLAG1.BUF7I（Rx FIFO 溢出）被置位。当 Rx FIFO 满且新报文被邮箱捕获时，CAN_IFLAG1.BUF7I 不会被置位。BUF7I 会一直保持置位，直到 CPU 将其清零。

清除这三个标志位中的任何一个都不会影响其他两个的状态。

如果 IFLAG 位被置位且相应的掩码位也被置位，则生成中断。

FlexCAN 提供了一个高效的过滤机制，只为目标应用接收帧，减少了中断处理的负荷。过滤标准通过编程最多 40 个 32 位寄存器来制定，根据 CAN_CTRL2.RFFN 的设定，可被配置为下列格式之一：

- 格式 A：40 IDAFs（包含 IDE 和 RTR 的标准或扩展 ID）
- 格式 B：80 IDAFs（包含 IDE 和 RTR 的标准 ID 和扩展 ID 的 14 位）
- 格式 C：160 IDAFs（标准或扩展 ID 的 8 位）

注：选定的格式适用于过滤表的所有条目。表格里的格式不能混用。参见章节“Rx FIFO 结构”。

Rx FIFO 中的每个可用帧均有一个相应的 IDHIT（ID 接收过滤命中指示），可从 C/S 的 IDHIT 字段中读取。CPU 还可通过访问 CAN_RXFIR 寄存器获得该信息。CAN_RXFIR.IDHIT 字段在 CAN_IFLAG1.BUF5I 标志被置位时有效，与 Rx FIFO 输出的报文相关联。CAN_RXFIR 寄存器必须在标志被清零前读取，以确保信息指示 Rx FIFO 里正确的帧。

根据 CAN_CTRL2.RFFN 的设置，过滤表中高达 16 个元素分别受到私有掩码寄存器（CAN_RXIMRn）的影响，方便定义很高效的过滤条件。如果 CAN_MCR.IRMQ 为 0，则 Rx FIFO 过

滤表受 CAN_RXFGMASK 影响。

33.3.9.1 DMA

通过置位 CAN_MCR.RFEN 和 CAN_MCR.DMA 可以使能 Rx FIFO 的 DMA 功能。

DMA 控制器可以通过读取 Rx FIFO 输出接口 0x80 ~ 0x8C 地址范围的报文缓冲区结构来读取接收到的报文。

当 CAN_MCR.DMA 被置位时，CPU 禁止访问 Rx FIFO 输出接口地址范围。CAN_MCR.DMA 置位前，CPU 必须处理 Rx FIFO 区域被置位的 IFLAG。否则，这些 IFLAG 可能会显示 Rx FIFO 中有待处理的数据，并错误地生成 DMA 请求。CAN_MCR.DMA 被清零前，CPU 必须执行 Rx FIFO 清除操作。

当 Rx FIFO 中至少有一个帧可被读取时，CAN_IFLAG1.BUF5I（Rx FIFO 中有可用帧）被置位，同时生成 DMA 请求。一旦接收到请求，DMA 控制器能够读取报文（以报文缓冲区的方式访问 Rx FIFO 的输出）。DMA 读取流程必须以读地址 0x8C 结束，该操作会清零 CAN_IFLAG1.BUF5I，并将 Rx FIFO 输出更新为下一条报文（如果 Rx FIFO 不为空），将 CAN_RXFIR 更新为该报文的属性。如果 Rx FIFO 内还存储了更多的报文，则 CAN_IFLAG1.BUF5I 被重新置位并发出新的 DMA 请求。否则，CAN_IFLAG1.BUF5I 仍为清零状态。

注：DMA 完成 Rx FIFO 读取后，不能读取 CAN_RXFIR 寄存器的内容。IDHIT 信息还可从地址 0x080 的 C/S 获取（参见章节“Rx FIFO 结构”）。

DMA 功能使能时，CAN_IFLAG1.BUF6I 和 CAN_IFLAG1.BUF7I 不可用。

当 FlexCAN 与 DMA 协作时，CPU 不会收到任何 Rx FIFO 中断，且不会清零相关的 IFLAG。此外，相关的 IMASK 并不用于屏蔽 DMA 请求的生成。

33.3.9.2 Rx FIFO 清除操作

当 CAN_MCR.RFEN 被置位时，Rx FIFO 清除操作用于清空 Rx FIFO 的内容。CAN_MCR.RFEN 被置位后，当 CPU 向 CAN_IFLAG1.BUF0I 写 1 时，Rx FIFO 被清除。该操作只能在冻结模式下进行，且不会清零 Rx FIFO 相关的 IFLAG。因此，CPU 在执行 Rx FIFO 清除操作前，必须处理所有 Rx FIFO 相关的 IFLAG。当 Rx FIFO 与 DMA 协作时，Rx FIFO 清除操作会清零 CAN_IFLAG1.BUF5I 标志，DMA 请求被取消。

注：除非 CAN_MCR.DMA 被置位，否则 Rx FIFO 清除操作不会清零 IFLAG。当 CAN_MCR.DMA 被置位时，只有 CAN_IFLAG1.BUF5I 标志会被清零。

33.3.10 CAN 协议

本章节描述了 CAN 协议相关的功能。

33.3.10.1 远程帧

CPU 可以通过将邮箱传输时的 RTR 位设置为‘1’来编程邮箱为远程帧。远程帧成功传输后，邮箱变为具有相同 ID 的接收报文缓冲区。

FlexCAN 将结果应答帧接收到和传输远程帧相同的报文缓冲区中。由于应答帧的 RTR = 0，而远程帧的 RTR = 1，在比较传入的应答帧时，对该报文缓冲区的过滤不能考虑 RTR 位。要排除 RTR 位，请确保以下位中的一个或两个都为 0：

- 各掩码寄存器的 RTR 位，可以是私有掩码寄存器 RXIMRn[31] 或全局掩码寄存器 RXMGMASK[31]，取决于 CAN_MCR.IRMQ。CAN_MCR.IRMQ = 1 时，RXIMRn[31] 禁用报文缓冲区 n 的 RTR 比较；CAN_MCR.IRMQ = 0 时，RXMGMASK[31]禁用所有报文缓冲区的 RTR 比较
- CAN_CTRL2.EACEN = 0 可全局地禁止比较传入帧的 RTR 位，忽略报文缓冲区的 RTR 掩码位

当 FlexCAN 接收到远程帧时，根据远程请求存储位（CAN_CTRL2.RRS）和 Rx FIFO 使能位（CAN_MCR.RFEN）的配置，可采取以下三种方式进行处理：

- 如果 RRS 为 0，则帧的 ID 与代码字段为 0b1010 的传输报文缓冲区的 ID 相比较。如果有匹配的 ID，则该邮箱被传输。注：如果匹配邮箱的 RTR 位被置位，则 FlexCAN 传输远程帧进行响应。接收到的远程帧不会存入接收缓冲区，而是仅用于触发应答帧的传输。掩码寄存器不用于远程帧的匹配，接收到的帧的所有 ID 位（除了 RTR）都应匹配。如果接收到远程帧并且匹配了一个邮箱，则报文缓冲区立即进入内部仲裁过程，但会被认为是一个正常的没有高优先级的 Tx 邮箱。该帧的数据长度与启动传输的远程帧的 DLC 字段无关。
- 如果 RRS 被置位，则帧的 ID 与代码字段为 0b0100、0b0010 或 0b0110 的接收邮箱的 ID 相比较。如果有匹配的 ID，则该邮箱以与数据帧相同的方式存储远程帧。不会自动产生远程响应帧；掩码寄存器被用于匹配过程。
- 如果 RFEN 被置位，则 FlexCAN 不会自动响应与 Rx FIFO 过滤条件匹配的远程帧。如果远程帧匹配目标 ID 中的任何一个，则会被存储于 Rx FIFO 并提交给 CPU。

注：对于过滤格式 A 和 B，可以选择是否接收远程帧；对于格式 C，远程帧总是被接收（如果 ID 匹配）。远程帧被认为是普通帧，当接收成功且 Rx FIFO 已满时，会产生 Rx FIFO 溢出。

33.3.10.2 过载帧

FlexCAN 在 CAN 总线上检测到以下条件时会传输过载帧：

- 在 Intermission 的第一/第二位检测到一个显性位
- 在 Rx 帧结束字段的第 7 位（最后一位）检测到一个显性位
- 在错误帧界定符或过载帧界定符的第 8 位（最后一位）检测到一个显性位

33.3.10.3 时间戳

自由运行计时器的值为 CAN 总线上 ID 段起始时的采样值，在移入结束时存入报文缓冲区的时间戳字段，为网络行为提供时间参考。

自由运行计时器通过 FlexCAN 的位时钟计时，位时钟定义了 CAN 总线上的波特率。报文传输/接收时，每传输或接收一位，计时器就加 1。当总线上没有报文时，使用先前编程的波特率计数。

模块禁止模式和冻结模式时自由运行计时器不会增加。计时器可以在接收到特定帧时复位，使网络时间同步。

33.3.10.4 协议时序

下图为产生 PE 时钟的电路结构。时钟源选择位 (CAN_CTRL1.CLKSRC) 定义了内部时钟为异步时钟或同步时钟。其中，同步时钟为外设时钟；异步时钟暂不支持。为保证可靠运行，应在模块禁止模式时 (CAN_MCR.MDIS 置位) 选择时钟源为同步时钟。

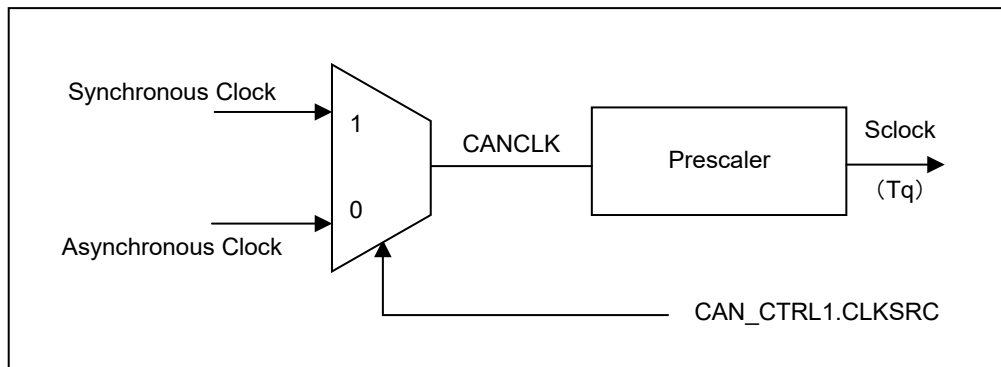


图 33-2 PE 时钟结构

FlexCAN 支持多种方式来设置 CAN 协议所要求的位时序参数。控制寄存器 1 (CAN_CTRL1) 有各种用于控制位时序参数的字段：PRES DIV、PROPSEG、PSEG1、PSEG2 和 RJW。CAN 位时序寄存器 (CAN_CBT) 扩展了 CAN_CTRL1 中 CAN 位时序变量的范围。

PRES DIV 字段（及其扩展范围 EPRES DIV）定义了串行时钟 (Sclock) 的预分频（见下列方程）。串行时钟的周期定义了用于构成 CAN 波形的最小 Time Quantum (Tq)。Tq 为 CAN 引擎所能处理的最小时间单元。

$$Tq = \frac{(PRES DIV + 1)}{f_{CANCLK}}$$

比特率定义了接收或传输 CAN 报文的速率，公式如下：

$$CAN \text{ Bit Time} = (\text{Number of Time Quanta in 1 bit time}) * Tq$$

$$\text{Bit Rate} = \frac{1}{CAN \text{ Bit Time}}$$

位时间可以细分为三个部分：

- 同步段 (SYNC_SEG)：1Tq 的固定长度；信号边沿出现在该段内
- 时间段 1：包括 CAN 标准的传播段和相位段 1。该段可通过设置 CAN_CTRL1 寄存器的

PROPSEG 和 PSEG1 字段来编程，其总和 (+2) 为 2~16Tq。当 CAN_CBT.BTF 被置位时，FlexCAN 使用来自 CAN_CBT 寄存器的 EPROPSEG 和 EPSEG1 字段，其总和 (+2) 为 2~96Tq。

- 时间段 2：CAN 标准的相位段 2。该段可通过设置 CAN_CTRL1 寄存器的 PSEG2 字段来编程，其值(+1)为 2~8Tq。当 CAN_CBT.BTF 被置位时，FlexCAN 使用来自 CAN_CBT 寄存器的 EPSEG2 字段，其值(+1)为 2~32Tq。时间段 2 不能小于信息处理时间(IPT)，IPT 在 FlexCAN 中为 2Tq。

注：上述时间段定义的位时间必须不小于 5Tq。对于位时间计算，使用 2Tq 的信息处理时间 (IPT)，该值在 FlexCAN 中实现。

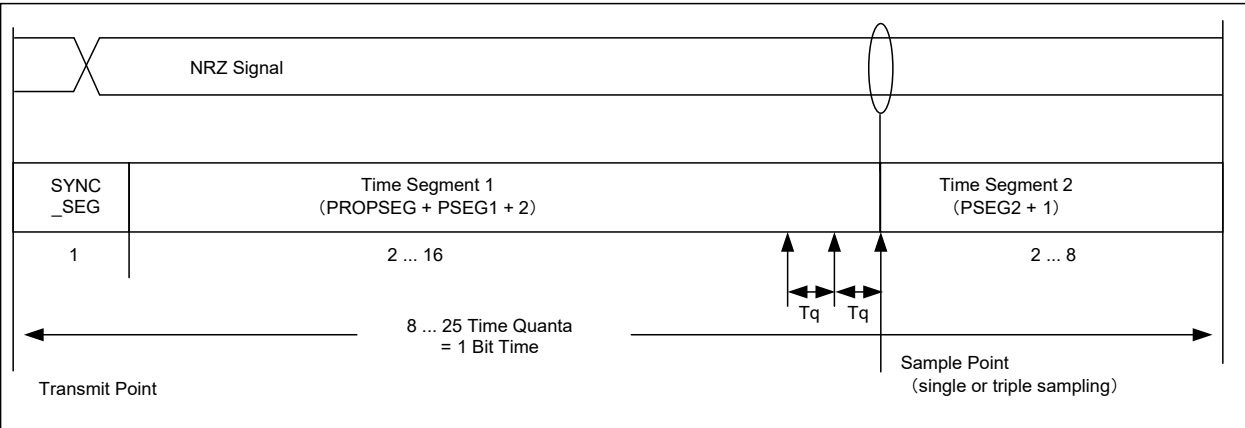


图 33-3 位时间内的段（使用 CAN_CTRL1 位时序变量的经典 CAN 格式）

表 33-5 时间段语法

语法	描述
SYNC_SEG	系统期望在此期间总线上发生转换
TSEG1	对应于 PROPSEG 和 PSEG1 的总和
TSEG2	对应于 PSEG2 的值
传输点	传输模式下的节点在该点传输新值到 CAN 总线上
采样点	节点在该点采样总线。如果选择每位采样三次，则该点位于第三次采样的位置

下表列举了一些经典 CAN 格式（Bosch CAN 2.0B）报文的时间段设置示例。

表 33-6 符合 Bosch CAN 2.0B 标准的位时间段设置

时间段 1	时间段 2	重同步补偿宽度
5 ~ 10	2	1 ~ 2
4 ~ 11	3	1 ~ 3
5 ~ 12	4	1 ~ 4
6 ~ 13	5	1 ~ 4
7 ~ 14	6	1 ~ 4
8 ~ 15	7	1 ~ 4
9 ~ 16	8	1 ~ 4

注：用户必须确保位时间的设置符合 CAN 协议标准（ISO 11898-1）。

当采用 CAN 位作为持续时间的衡量标准时（例如，评估报文中的 CAN 位事件），一个 CAN 位的



外设时钟个数（NumClkBit）为：

$$\text{NUMClkBit} = \frac{f_{\text{SYS}}}{f_{\text{CANCLK}}} \times (\text{PRES DIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

其中：

- NumClkBit 为一个 CAN 位的外设时钟个数
- f_{CANCLK} 为 PE 时钟，单位 Hz
- f_{SYS} 为系统（CHI）时钟频率，单位 Hz
- PSEG1 为 CAN_CTRL1.PSEG1 的值
- PSEG2 为 CAN_CTRL1.PSEG2 的值
- PROPSEG 为 CAN_CTRL1.PROPSEG 的值
- PRES DIV 为 CAN_CTRL1.PRES DIV 的值

上述公式也适用于 CAN 位时序寄存器（CAN_CBT）所述的 CAN 位时间变量。

33.3.10.5 仲裁和匹配时序

在正常接收和传输期间，匹配、仲裁、移入和移出流程都在 CAN 帧的特定时间段被执行，如下图所示。

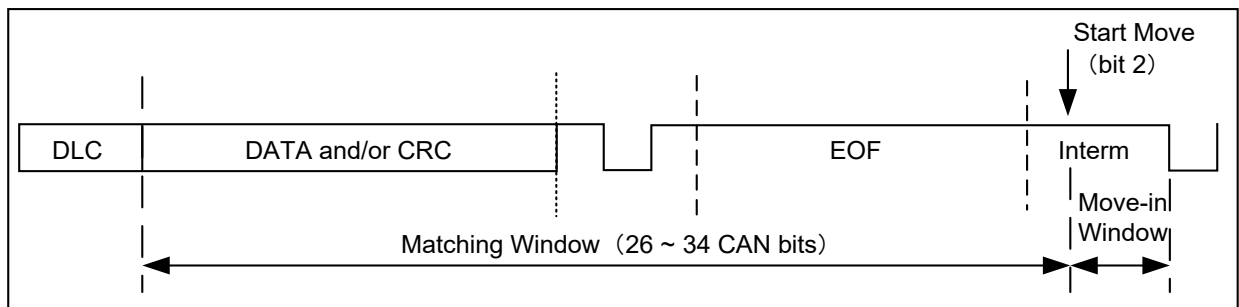


图 33-4 匹配和移入时间段

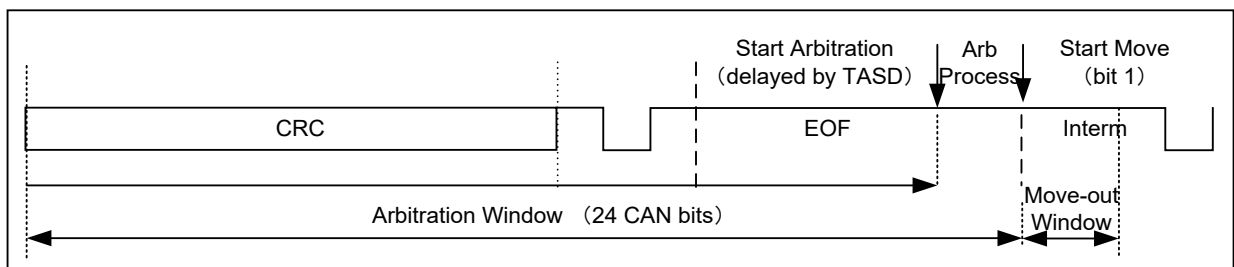


图 33-5 仲裁和移出时间段

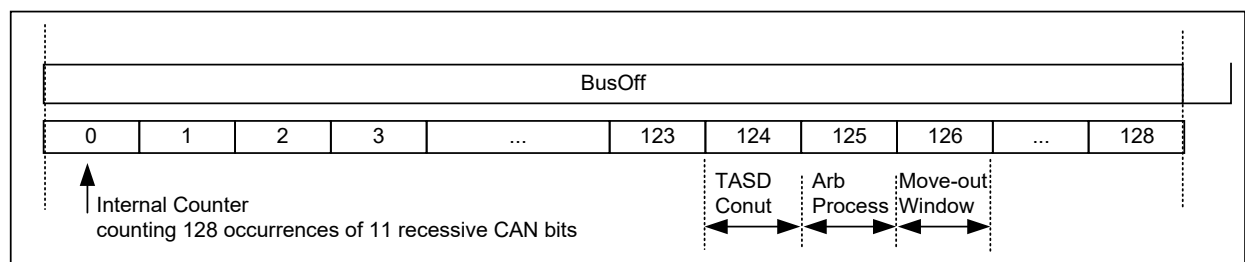


图 33-6 总线关闭结束时的仲裁和移出时间段

注：上图中，匹配和仲裁时序未考虑 CPU 或 FlexCAN 内部其他子模块同时访问内存所引起的延迟。

33.3.10.6 Tx 仲裁启动延迟

Tx 仲裁启动延迟 (CAN_CTRL2.TASD) 位字段，指示了 FlexCAN 从当前帧的 CRC 字段首位开始，延迟 Tx 仲裁过程起点所用的 CAN 位个数。CAN_CTRL2.TASD 只能在冻结模式下写入，其他模式下被硬件锁定。

CPU 在内部仲裁过程结束后重新配置传输 MB 的能力，影响了传输性能。如果仲裁在 Intermission 字段的首位前结束得过早，则 CPU 有机会重新配置部分 Tx MB，获胜 MB 不再是传输的最佳候选者。

TASD 可用于通过定义仲裁起点来优化传输性能，基于以下要素：

- 决定 CAN 位速率的 CAN 位时序变量
- 匹配和仲裁过程所采用的 MB 数量

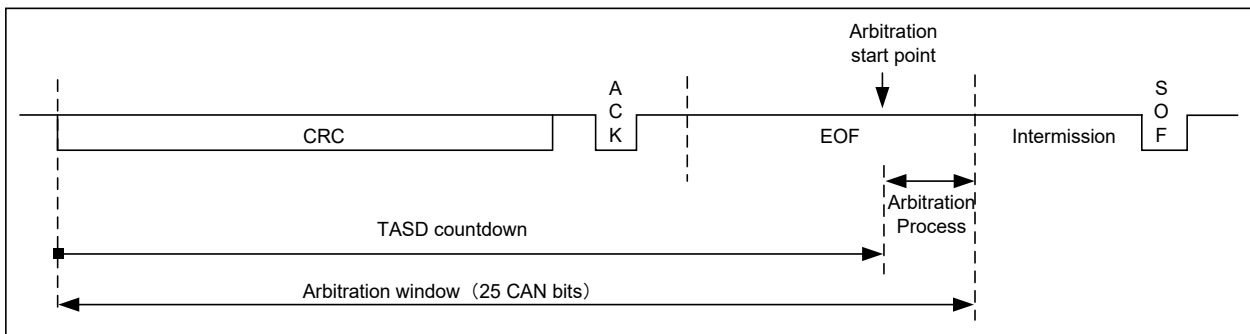


图 33-7 优化的 Tx 仲裁起点

从 CAN 位的角度看，仲裁过程的持续时间与可用的 MB 数量和 CAN 位速率成正比，与外设时钟频率成反比。

最优的仲裁时序，是在 CAN 帧 Intermission 字段首位前扫描最后一个 MB。例如，如果只有少量的 MB，且 CAN 波特率低，则仲裁可以更靠近帧的末端，增加起点的延迟，反之亦然。如果 TASD 被设置为 0，则仲裁起点不延迟，且更多的时间会被预留给仲裁。此外，如果 TASD 接近 24，则 CPU 可以稍后配置 Tx MB，并减少仲裁预留时间。如果为仲裁预留的时间太少，则 FlexCAN 可能无法及时找到获胜邮箱，并在最佳时机发送以赢得总线仲裁。

优化 TASD 的值计算如下：

$$TASD = 25 - \frac{2 \times (MAXMB + 1) + 4}{CPCB}$$

其中：

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRES DIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB = CPCB_N$$

- MAXMB 为 CAN_MCR.MAXMB 的值
- BITRATE_N 是由 CAN 位时间变量计算出的 CAN 位速率，单位 bps
- CPCB 是每个 CAN 位的外设时钟数量
- f_{CANCLK} 为 PE 时钟，单位 Hz
- f_{SYS} 为外设时钟，单位 Hz
- EPSEG1 为 CAN_CBT.EPSEG1 的值（也可使用 CAN_CTRL1.PSEG1）
- EPSEG2 为 CAN_CBT.EPSEG2 的值（也可使用 CAN_CTRL1.PSEG2）
- EPROPSEG 为 CAN_CBT.EPROPSEG 的值（也可使用 CAN_CTRL1.PROPSEG）
- EPRES DIV 为 CAN_CBT.EPRES DIV 的值（也可使用 CAN_CTRL1.PRES DIV）

更多细节参见章节“协议时序”。

33.3.11 时钟限制

进行匹配和仲裁时，FlexCAN 需要在一个 CAN 帧的时间段内，扫描整个 MB 内存。该时间段由多个 CAN 位组成，为了获得充足的时间，CAN 位需要满足最少外设时钟数的限制。

表 33-7 经典 CAN 格式 CAN 位的最少外设时钟数

邮箱数量	CAN_MCR.RFEN	CAN 位的最少外设时钟数
16	0	16
16	1	16

CAN 比特率取决于 CAN 位的 Tq 数，可通过 CAN_CTRL1 或 CAN_CBT 寄存器进行调整。Tq 的定义参见章节“协议时序”。每个 CAN 位的最少 Tq 数为 8。

33.3.12 复位

FlexCAN 支持三种复位方式：

- 芯片硬件复位：异步复位所有内存映射的寄存器
- FlexCAN 软件复位（CAN_MCR.SOFTTRST）：同步复位部分内存映射的寄存器，参见表格“寄存器访问和复位信息”
- 芯片软件复位：效果同 CAN_MCR.SOFTTRST

软件复位是同步的，当软件复位待处理时，CAN_MCR.SOFTTRST 保持置位。因此，轮询该位可得知

复位何时完成。低功耗模式时钟关闭时不能应用软件复位，复位前应退出低功耗模式并恢复时钟。

当 FlexCAN 处于模块禁止模式时，应置位 CAN_CTRL1.CLKSRC。置位该位并使能 FlexCAN 时（CAN_MCR.MDIS 位清零），FlexCAN 自动进入冻结模式。在冻结模式下，FlexCAN 未与 CAN 总线同步，CAN_MCR.HALT 和 CAN_MCR.FRZ 被置位，内部状态机停止，且 CAN_MCR.FRZACK 和 CAN_MCR.NOTRDY 位被置位。Tx 管脚为隐性，FlexCAN 不会启动任何 CAN 帧的传输或接收。注：MB 和 Rx 私有掩码寄存器不受复位影响，因此不会自动被初始化。

33.3.13 初始化

任何配置的更改/初始化，都要求 FlexCAN 必须进入冻结模式（参见章节“冻结模式”）。以下为 FlexCAN 的通用初始化步骤：

1) 初始化模块配置寄存器（CAN_MCR）：

- 置位 IRMQ，使能各 MB 的私有过滤和接收队列功能
- 置位 WRNEN，使能警告中断
- 根据需要，置位 SRXDIS 以停止帧的自接收
- 置位 RFEN，使能 Rx FIFO
- 如果 Rx FIFO 已使能且需使用 DMA，则置位 DMA 位
- 置位 AEN 位，使能中止机制
- 置位 LPRIOEN 位，使能本地优先级功能

2) 初始化控制寄存器 1（CAN_CTRL1）、CAN 位时序寄存器（CAN_CBT）（可选）

- 确定位时序参数：PROPSEG、PSEG1、PSEG2 和 RJW（或 EPROPSEG、EPSEG1、EPSEG2 和 ERJW）
- 编程 PRES DIV（或 EPRES DIV）字段以配置位速率
- 确定内部仲裁模式（LBUF 位）

3) 初始化报文缓冲区：

- 所有 MB 的 C/S 必须被初始化
- 如果 Rx FIFO 使能，则 ID 过滤表必须被初始化
- 各 MB 的其他条目应按要求进行初始化

4) 初始化 Rx 私有掩码寄存器（CAN_RXIMRn）

5) 置位所需中断掩码位：CAN_CTRL1/CAN_CTRL2（总线关闭和错误中断）、CAN_IMASKn（所有 MB 中断）

6) 清除 CAN_MCR.HALT 位

最后一个步骤后，FlexCAN 尝试同步至 CAN 总线。

33.3.14 中断

FlexCAN 支持很多中断源：由 MB 引起的中断、总线关闭、总线关闭完成、错误、Tx 警告和 Rx 警告。

如果设置了相应的 IMASK 位，每个 MB 都可以作为中断来源。每个缓冲区在 CAN_IFLAG1 寄存器中都有指定的标志位，当对应的缓冲区成功完成传输时置位；当 CPU 写 1 至该位时清零（除非在同一时间产生了另一个中断）。

注：必须保证 CPU 仅清零引起当前中断的位。因此，位操作指令（BSET）不能用于清除中断标志，这些指令可能会导致进入当前中断服务程序后设置的中断标志被意外清除。

Rx FIFO 使能（CAN_MCR.RFEN = 1）且 DMA 禁止（CAN_MCR.DMA = 0）时，对应于 MB0 ~ 7 的中断含义不同：

- CAN_IFLAG1 寄存器位 7 为“Rx FIFO 溢出”标志
- 位 6 为“Rx FIFO 警告”标志
- 位 5 为“Rx FIFO 中有可用帧”标志
- 位 4 ~ 0 未使用

参加章节“中断标志寄存器 1（CAN_IFLAG1）”以获得更多信息。

如果 Rx FIFO 和 DMA 都被使能（CAN_MCR.RFEN = 1 且 CAN_MCR.DMA = 1），FlexCAN 不会生成任何 Rx FIFO 中断。CAN_IFLAG1 寄存器位 5 仍表示“Rx FIFO 中有可用帧”，并生成 DMA 请求。位 7、6 和 4 ~ 0 未使用。

对于组合中断（多个 MB 中断源的“或”），任何一个相关的 MB（或 FIFO）产生中断时，就会生成组合中断。此时，CPU 必须通过读取 CAN_IFLAG1 寄存器来确定造成中断的 MB（或 FIFO）是哪一个。

总线关闭、总线关闭完成、错误、Tx 警告和 Rx 警告中断，可以从 CAN_ESR1 寄存器读取。总线关闭、错误、Tx 警告、Rx 警告中断的掩码位位于 CAN_CTRL1 寄存器。

33.4 寄存器/内存映射

CPU 访问 FlexCAN 寄存器需遵循下列规则：

- 无限制地读写管理员权限的寄存器（下表中，管理员模式下访问类型为 S/U 的寄存器；或访问类型为 S 的寄存器）会导致访问错误
- 读写预留地址空间会导致访问错误
- 对当前所有位都为只读的地址执行写操作会导致访问错误。如果至少有一个位不是只读，则不会发生访问错误。某地址或其部分位的写权限会随操作模式或临时状态而改变，细节请参考寄存器和位描述
- 如果 CAN_MCR.MAXMB 的值小于可用的 MB 数，则可以将未使用的内存空间用作通用 RAM 空间

寄存器的访问类型可以是管理员（S）或无限制（U），大部分的寄存器都可通过编程 CAN_MCR.SUPV

配置为管理员或无限制访问。

表 33-8 寄存器访问和复位信息

寄存器	访问类型	硬件复位	软件复位
CAN_MCR 模块配置寄存器	S	是	是
CAN_CTRL1 控制寄存器 1	S/U	是	否
CAN_TIMER 自由运行计时器	S/U	是	是
CAN_RXMGMASK Rx 邮箱全局掩码寄存器	S/U	否	否
CAN_RX14MASK Rx 14 掩码寄存器	S/U	否	否
CAN_RX15MASK Rx 15 掩码寄存器	S/U	否	否
CAN_ECR 错误计数寄存器	S/U	是	是
CAN_ESR1 错误和状态寄存器 1	S/U	是	是
CAN_IMASK1 中断掩码寄存器 1	S/U	是	是
CAN_IFLAG1 中断标志寄存器 1	S/U	是	是
CAN_CTRL2 控制寄存器 2	S/U	是	否
CAN_ESR2 错误和状态寄存器 2	S/U	是	是
CAN_CRCRC CRC 寄存器	S/U	是	是
CAN_RXFGMASK Rx FIFO 全局掩码寄存器	S/U	否	否
CAN_RXFIR Rx FIFO 信息寄存器	S/U	否	否
CAN_CBT CAN 位时序寄存器	S/U	是	否
报文缓冲区	S/U	否	否
CAN_RXIMRn Rx 私有掩码寄存器	S/U	否	否

33.4.1 寄存器/内存映射总览

FlexCAN 可使用报文缓冲区和 Rx FIFO 结构存储 CAN 报文，用于传输与接收。RAM 空间从偏移地址 0x80 开始。

下表为 FlexCAN 的内存映射，包括 16 个 128 位的报文缓冲区（MB），占据了偏移地址 0x80 至 0x17F。

表 33-9 FlexCAN 寄存器/内存映射概览

偏移地址	寄存器	访问权限	复位值
00h	CAN_MCR 模块配置寄存器	R/W	D890_000Fh
04h	CAN_CTRL1 控制寄存器 1	R/W	0000_0000h
08h	CAN_TIMER 自由运行计时器	R/W	0000_0000h
10h	CAN_RXMGMASK Rx 邮箱全局掩码寄存器	R/W	自定义
14h	CAN_RX14MASK Rx 14 掩码寄存器	R/W	自定义
18h	CAN_RX15MASK Rx 15 掩码寄存器	R/W	自定义
1Ch	CAN_ECR 错误计数寄存器	R/W	0000_0000h
20h	CAN_ESR1 错误和状态寄存器 1	R/W	0000_0000h
28h	CAN_IMASK1 中断掩码寄存器 1	R/W	0000_0000h
30h	CAN_IFLAG1 中断标志寄存器 1	R/W	0000_0000h
34h	CAN_CTRL2 控制寄存器 2	R/W	00B0_0000h
38h	CAN_ESR2 错误和状态寄存器 2	R/W	0000_0000h

偏移地址	寄存器	访问权限	复位值
44h	CAN_CRCCR CRC 寄存器	R	0000_0000h
48h	CAN_RXFGMASK Rx FIFO 全局掩码寄存器	R/W	自定义
4Ch	CAN_RXFIR Rx FIFO 信息寄存器	R	自定义
50h	CAN_CBT CAN 位时序寄存器	R/W	0000_0000h
880h ~ 8BCh	CAN_RXIMRn Rx 私有掩码寄存器 (n = 0 ~ 15)	R/W	自定义

33.4.2 CAN_MCR 模块配置寄存器

偏移地址：0x00

复位值：0xD890 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MDIS	FRZ	RFEN	HALT	NOTRDY	Res.	SOFTRESET	FRZACK	SUPV	Res.	WRNEN	LPMACK	Res.		SRXDIS	IRMQ
rw	rw	rw	rw	r		rw	r	rw		rw	r			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA	Res.	LPRIOE N	AEN	Res.		IDAM		Res.	MAXMB						
rw		rw	rw			rw			rw						

字段	名称	描述
31	MDIS	1: 禁止 FlexCAN, 停止 PE 和 CHI 子模块的时钟 0: 使能 FlexCAN 该位不受软件复位影响。
30	FRZ	1: 使能冻结模式 0: 禁止冻结模式 当 CAN_MCR.HALT 置位或芯片请求调试模式时, FRZ 用于指定 FlexCAN 的行为。FRZ 被置位时, FlexCAN 将会进入冻结模式; 该位置 0 将使 FlexCAN 退出冻结模式。
29	RFEN	1: 使能 Rx FIFO 0: 禁止 Rx FIFO 置位 RFEN 时, MB0~5 不能用于正常的接收和传输, 因为相应的内存空间(0x80 ~ 0xDC) 被 FIFO 引擎使用。额外的 MB (最多 10 个, 取决于 CAN_CTRL2.RFFN 的设置) 用于 Rx FIFO ID 过滤表元素。 RFEN 只能在冻结模式写入。
28	HALT	1: 如果 FRZ 被置位, 则进入冻结模式 0: 没有冻结模式请求 该位置位会使 FlexCAN 进入冻结模式。CPU 在初始化报文缓冲区和控制寄存器 (CAN_CTRL1 和 CAN_CTRL2) 后应将该位清零。FlexCAN 在 HALT 清零前不会进行接收或传输。模块处于低功耗模式时, 不能进入冻结模式。

字段	名称	描述
27	NOTRDY	1: FlexCAN 处于禁止或冻结模式 0: FlexCAN 处于正常、只听或回环模式 该位只读，在 FlexCAN 退出禁止或冻结模式后清零，不受软件复位影响。
26	Reserved	保留，必须保持复位值
25	SOFRST	软件复位： 1: 复位受软件复位影响的寄存器 0: 无复位请求 该位置位时，FlexCAN 复位内部状态机和部分内存映射寄存器。 该位可直接由 CPU 写 CAN_MCR 置位。 如果复位没有完成，SOFRST 将保持置位，完成复位后自动清零。因此软件可以轮询该位以判断软件复位是否完成。 在低功耗模式下，如果时钟被关闭，则软件复位将不起作用。FlexCAN 应该首先退出低功耗模式，然后软件复位才会起作用。 该位不受软件复位影响。
24	FRZACK	冻结模式应答： 1: FlexCAN 处于冻结模式，预分频器停止 0: FlexCAN 未处于冻结模式，预分频器运行 该位只读。直到当前传输或者接收过程完成，冻结模式请求才会被执行。因此软件可通过轮询该位以了解 FlexCAN 是否已确切地进入冻结模式。如果冻结模式请求被取消，则一旦 FlexCAN 模块预分频器又重新运行，该位被清零。如果 FlexCAN 在低功耗模式时请求冻结模式，则只有当模块退出低功耗模式时 FRZACK 位才会被置位。参见章节“冻结模式”。 该位不受软件复位影响。
23	SUPV	1: FlexCAN 处于管理员模式，受影响的寄存器只允许管理员访问 0: FlexCAN 处于用户模式，受影响的寄存器允许管理员和无限制访问 受 SUPV 影响的寄存器在表格“寄存器访问和复位信息”的访问类型列被标记为 S/U。 SUPV 复位值为 1，因此受影响的寄存器只能从管理员访问开始。该位只能在冻结模式下写入，其他模式下被硬件锁定。
22	Reserved	保留，必须保持复位值
21	WRNEN	警告中断使能： 1: 当相应的错误计数器从小于 96 上升至大于等于 96 时，TWRNINT 和 RWRNINT 位被置位 0: TWRNINT 和 RWRNINT 位为零，与错误计数器的值无关 WRNEN 被置位时，错误和状态寄存器 1 (CAN_ESR1) 产生 TWRNINT 和 RWRNINT 标志。如果 WRNEN 被清零，则 TWRNINT 和 RWRNINT 标志将一直为零，与错误计数器的值无关，且不会生成警告中断。 WRNEN 只能在冻结模式下写入，其他模式下被硬件锁定。

字段	名称	描述
20	LPMACK	低功耗模式应答： 1: FlexCAN 模块处于低功耗模式 0: FlexCAN 模块未处于低功耗模式 该位只读。直到所有当前传输或接收流程已完成，FlexCAN 才能进入低功耗模式。因此，CPU 可通过轮询该位以了解 FlexCAN 是否已确切地进入低功耗模式。 LPMACK 不受软件复位影响。
19: 18	Reserved	保留，必须保持复位值
17	SRXDIS	1: 禁止自接收 0: 使能自接收 该位控制 FlexCAN 是否接收自身传输的帧。如果 SRXDIS 被置位，由 FlexCAN 传输的帧不会存储在任 MB 中，无论 MB 的 ID 是否与帧匹配，且接收帧时不会生成中断标志或中断信号。 SRXDIS 只能在冻结模式下写入，其他模式下被硬件锁定。
16	IRMQ	1: 使能私有 Rx 掩码和队列功能 0: 禁止私有 Rx 掩码和队列功能 IRMQ 控制 Rx 匹配过程是基于私有掩码和队列功能或基于 CAN_RXMGMASK、CAN_RX14MASK、CAN_RX15MASK 和 CAN_RXFGMASK 构成的掩码机制。 IRMQ 只能在冻结模式下写入，其他模式下被硬件锁定。
15	DMA	1: 使能 Rx FIFO DMA 功能 0: 禁止 Rx FIFO DMA 功能 DMA 功能只能在 Rx FIFO 下使用，因此，CAN_MCR.RFEN 必须被置位。置位了 DMA 和 RFEN 时，CAN_IFLAG1.BUF5I 会生成 DMA 请求，而不是生成 Rx FIFO 中断。 DMA 位只能在冻结模式下写入，其他模式下被硬件锁定。
14	Reserved	保留，必须保持复位值
13	LPRIEN	1: 使能本地优先级 0: 禁止本地优先级 本地优先级用于在仲裁期间扩展 ID，仲裁过程基于完整的 32 位字完成。但实际传输的 ID 仍为标准帧的 11 位、扩展帧的 29 位 ID。 LPRIEN 只能在冻结模式下写入，其他模式下被硬件锁定。
12	AEN	1: 使能中止 0: 禁止中止 Tx 中止机制确保安全地中止未完成的传输，因此在没有通知的情况下 CAN 总线上不会有帧传输，参见章节“传输中止机制”。 AEN 只能在冻结模式下写入，其他模式下被硬件锁定。 注：当 AEN 被置位时，中止机制只能用于更新传输邮箱，将中止代码写入 Rx 邮箱会导致不可预测的结果。
11: 10	Reserved	保留，必须保持复位值

字段	名称	描述
9: 8	IDAM	Rx FIFO ID 过滤表元素格式： 00: 格式 A — 每个 ID 过滤表元素有一个完整的 ID（标准和扩展） 01: 格式 B — 每个 ID 过滤表元素都有两个完整的标准 ID 或两个不完整的 14 位扩展 ID 10: 格式 C — 每个 ID 过滤表元素有四个不完整的 8 位标准 ID 11: 格式 D — 拒绝所有帧 所有表元素同时被 IDAM 配置（采用相同格式），参见章节“Rx FIFO 结构”。 IDAM 只能在冻结模式下写入，其他模式下被硬件锁定。
7	Reserved	保留，必须保持复位值
6: 0	MAXMB	最后一个将参与匹配和仲裁过程的 MB 编号 复位值为 0x0F，即配置了 16 个报文缓冲区。 MAXMB 只能在冻结模式下写入，其他模式下被硬件锁定。 注：MAXMB 必须小于等于可用 MB 的数量。此外，MAXMB 的值必须考虑 Rx FIFO 占用的 MB 以及由 CAN_CTRL2.RFFN 定义的 ID 过滤表空间。

33.4.3 CAN_CTRL1 控制寄存器 1

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESDIV								RJW		PSEG1			PSEG2		
rw								rw		rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOFFM SK	ERRMS K	CLKSR C	LPB	TWRNM SK	RWRN MSK	Res.		SMP	BOFFR EC	TSYN	LBUF	LOM	PROPSEG		
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

字段	名称	描述
31: 24	PRESDIV	预分频器分频系数 — 定义 PE 时钟频率与串行时钟（Sclock）频率之间的比率 $\text{Sclock 频率} = \text{PE 时钟频率} / (\text{PRESDIV} + 1)$ Sclock 周期定义了 CAN 协议的 Tq。对于复位值，Sclock 频率等于 PE 时钟频率。PRESDIV 的最大值为 0xFF，因此 Sclock 的最小时钟频率为 PE 时钟频率除以 256，参见章节“协议时序”。 PRESDIV 只能在冻结模式下写入，其他模式下被硬件锁定。
23: 22	RJW	重同步补偿 — 定义了一次重同步补偿的最大 Tq 数 重同步补偿宽度 = RJW + 1 RJW 有效值为 0~3，只能在冻结模式下写入，其他模式下被硬件锁定。
21: 19	PSEG1	相位段 1 — 定义位时间（bit time）中相位段 1 的长度 $\text{相位段 1} = (\text{PSEG1} + 1) \times \text{Tq}$ PSEG1 有效值为 0~7，只能在冻结模式下写入，其他模式下被硬件锁定。

字段	名称	描述
18: 16	PSEG2	相位段 2 — 定义位时间中相位段 2 的长度 相位段 2 = (PSEG2+1) × Tq PSEG2 有效值为 0 ~ 7, 只能在冻结模式下写入, 其他模式下被硬件锁定。
15	BOFFMSK	总线关闭中断 (CAN_ESR1.BoffInt) 掩码: 1: 总线关闭中断使能 0: 总线关闭中断禁止
14	ERRMSK	错误中断 (CAN_ESR1.ErrInt) 掩码: 1: 错误中断使能 0: 错误中断禁止
13	CLKSRC	FlexCAN 通信时钟源选择: 1: 同步时钟 0: 异步时钟 只能在模块禁止模式下写入, 其他模式下被硬件锁定。 注: 通信时该位必须置 1。
12	LPB	1: 回环模式使能 0: 回环模式禁止 在回环模式下, FlexCAN 进行内部循环用于自检。发送器的输出从内部反馈到接收器的输入。Rx CAN 输入引脚被忽略, Tx CAN 输出处于隐性状态 (逻辑 1)。发送报文时, FlexCAN 和正常传输时一样, 将自身传输的报文看作从远程节点接收到的报文。 在回环模式下, FlexCAN 忽略 CAN 帧应答字段的 ACK, 生成内部应答位以确保能正确接收自身发送的报文, 并产生发送和接收中断。 LPB 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: 回环模式下, CAN_MCR.SRXDIS 不能被置位, 否则会妨碍自接收。
11	TWRNMSK	Tx 警告中断 (CAN_ESR1.TwrnInt) 掩码: 1: 警告中断使能 0: 警告中断禁止 当 CAN_MCR.WRNEN 为 0 时, 该位读数为 0。 只有 CAN_MCR.WRNEN 被置位时, 才能写入该位。
10	RWRNMSK	Rx 警告中断 (CAN_ESR1.RwrnInt) 掩码: 1: 警告中断使能 0: 警告中断禁止 当 CAN_MCR.WRNEN 为 0 时, 该位读数为 0。 只有 CAN_MCR.WRNEN 被置位时, 才能写入该位。
9: 8	Reserved	保留, 必须保持复位值
7	SMP	Rx 输入的 CAN 位采样模式: 0: 仅采样一次来决定 CAN 位的值 1: 采样三次来决定 CAN 位的值。一次常规采样 (采样点) 和 2 次之前的采样, 使用多数决规则。 SMP 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: 为确保正确运行, 置位 SMP 时, 必须保证 CAN_CTRL1.PSEG1 (或 CAN_CBT.EPSEG1) 最小 2Tq。

字段	名称	描述
6	BOFFREC	FlexCAN 如何从总线关闭状态中恢复: 0: 自动从总线关闭状态恢复使能 1: 自动从总线关闭状态恢复禁止 如果 BOFFREC 为 0, 则根据 CAN 2.0B 自动从总线关闭状态中恢复; 如果 BOFFREC 被置位, 则禁止自动恢复, FlexCAN 保持在总线关闭状态, 直到 BOFFREC 被软件清零。如果清零发生在 CAN 总线上检测到 128 次连续的 11 个隐性位之前, 则总线关闭恢复, 如同 BOFFREC 从未被置位一样; 如果清零发生在 128 次 11 个隐性位之后, 则 FlexCAN 通过等待 11 个隐性位来重新同步总线。 该位清零后, 在总线关闭期间可再次被重新置位, 但只有在 FlexCAN 下一次进入总线关闭状态时才会生效。如果 BOFFREC 在 FlexCAN 进入总线关闭时被清零, 总线关闭期间置位 BOFFREC 不会影响当前的总线关闭恢复。
5	TSYN	计时器同步 — MB0 每接收到一次报文就复位自由运行计时器的机制: 1: 计时器同步使能 0: 计时器同步禁止 计时器同步功能为同步多个 FlexCAN 站点提供特定的 SYNC 报文(全局网络时间)。如果 CAN_MCR.RFEN 被置位 (Rx FIFO 使能), 则第一个可用的邮箱(通过 CAN_CTRL2.RFFN 设置)取代 MB0, 被用于计时器同步。 TSYN 只能在冻结模式下写入, 其他模式下被硬件锁定。
4	LBUF	报文缓冲区传输的顺序: 1: 最小编号 MB 优先传输 0: 最高优先级 MB 优先传输 当 LBUF 被置位时, CAN_MCR.LPRIOEN 位不影响优先级仲裁。 LBUF 只能在冻结模式下写入, 其他模式下被硬件锁定。
3	LOM	配置 FlexCAN 在只听模式下运行: 1: FlexCAN 在只听模式下运行 0: 只听模式已停用 在只听模式下, 传输停止, CAN_ECR 描述的所有错误计数器都被冻结, 且 FlexCAN 在 CAN 错误被动模式下运行。只有被其他 CAN 节点应答了的报文才可以被接收。如果 FlexCAN 检测到一个尚未被应答的报文, 则标记为一个 BIT0 错误, 而不改变接收错误计数器 (CAN_ECR.RXERRCNT), 如同试图应答报文一样。 只听模式由指示被动错误的 CAN_ESR1.FLTCONF 字段的 状态 应答, 请求和应答之间可能有一些延迟。 LOM 只能在冻结模式下写入, 其他模式下被硬件锁定。
2: 0	PROPSEG	传播时间段 — 定义位时间中传播段的长度 传播时间 = (PROPSEG + 1) × Tq PROPSEG 有效值为 0~7, 只能在冻结模式下写入, 其他模式下被硬件锁定。

注 1: **CAN** 位时序变量 (**PRESDIV**、**PROPSEG**、**PSEG1**、**PSEG2** 和 **RJW**) 还可以在 **CAN_CBT** 寄存器中配置, 该寄存器扩展了所有相关变量的范围。如果 **CAN_CBT.BTF** 被置位, 则 **CAN_CTRL1** 中上述位时序变量字段变为只读。

注 2: **CAN_CTRL1** 和 **CAN_CBT** 的 **CAN** 位时序变量存储在同一个寄存器内。

33.4.4 CAN_TIMER 自由运行计数器寄存器

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER															
rw															
字段	名称		描述												
31: 16	Reserved		保留，必须保持复位值												
15: 0	TIMER		计时器值：包含自由运行计数器的值												

CAN_TIMER 寄存器代表一个 16 位自由运行的计数器，可通过 CPU 进行读写。该计数器复位值为 0x0，计数到 0xFFFF 后返回，重新开始计数。

计时器通过 CAN 位时钟（位时钟决定 CAN 总线波特率）来计数。报文传输/接收期间，每接收或传输一次，计时器加一；总线上没有报文时，计时器使用先前编程的波特率来计数；在冻结、禁止模式下计时器不增加。

当任何帧的 ID 段的第二位在 CAN 总线上时，计时器的值被捕获。捕获的值在报文成功被接收或传输后，被写入报文缓冲区的时间戳字段。

如果 CAN_CTRL1.TSYN 被置位，则每当第一个可用邮箱接收到报文时（根据 CAN_CTRL2.RFFN 的设置）计时器将会被复位。

CPU 可以在任何时候写该寄存器，但如果写操作和第一个邮箱的接收导致计时器复位同时发生，那么写入的值将被丢弃。

33.4.5 CAN_RXMGMASK Rx 邮箱全局掩码寄存器

CAN_RXMGMASK 寄存器位于 RAM 空间，用于掩码所有 Rx MB 的过滤字段，不包括具有私有掩码寄存器的 MB14 ~ 15。

CAN_MCR.IRMQ 为 0 时，CAN_RXMGMASK 有效；CAN_MCR.IRMQ 被置位时，CAN_RXMGMASK 无效。

偏移地址：0x10

复位值：x

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

rw

字段	名称	描述
31: 0	MG	Rx 邮箱全局掩码位: 0: 过滤器相应位“don't care” 1: 过滤器相应位被检查 只能在冻结模式下写入, 其他模式下被硬件锁定。 注: MG 与邮箱 ID 的对齐并不完美, 因为最高两个 MG 位会影响 RTR 和 IDE 字段(位于邮箱的 C/S)。 下表详细介绍了 MG 位是如何掩码各邮箱过滤字段的。

表 33-10 MG 位掩码邮箱过滤字段

SMB.RTR ⁽¹⁾	CAN_CTRL2. RRS	CAN_CTRL2. EACEN	邮箱过滤字段			
			MB.RTR	MB.IDE	MB.ID	预留
0	—	0	注 ⁽²⁾	注 ⁽³⁾	MG[28: 0]	MG[31: 29]
0	—	1	MG[31]	MG[30]	MG[28: 0]	MG[29]
1	0	—	—	—	—	MG[31: 0]
1	1	0	—	—	MG[28: 0]	MG[31: 29]
1	1	1	MG[31]	MG[30]	MG[28: 0]	MG[29]

- 1) 接收帧的 RTR 位, 被存放在 Rx SMB 中
- 2) 如果 CAN_CTRL2.EACEN 为 0, 则邮箱的 RTR 位永远不会和接收帧的 RTR 位比较
- 3) 如果 CAN_CTRL2.EACEN 为 1, 则邮箱的 IDE 位总是会和接收帧的 IDE 位比较

33.4.6 CAN_RX14MASK Rx 14 掩码寄存器

CAN_RX14MASK 寄存器位于 RAM 空间, 用于掩码 MB14 的过滤字段。

CAN_MCR.IRMQ 被置位时, CAN_RX14MASK 无效。

偏移地址: 0x14

复位值: x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX14M[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX14M[15:0]															
rw															
字段	名称		描述												
31: 0	RX14M		Rx MB14 掩码位: 0: 过滤器相应位“don't care”												

字段	名称	描述
		1: 过滤器相应位被检查 每位用作 MB14 相应过滤字段的掩码，与 CAN_RXMGMASK 的设置方式相同。 只能在冻结模式下写入，其他模式下被硬件锁定。

33.4.7 CAN_RX15MASK Rx 15 掩码寄存器

CAN_RX15MASK 寄存器位于 RAM 空间，用于掩码 MB15 的过滤字段。

CAN_MCR.IRMQ 被置位时，CAN_RX15MASK 无效。

偏移地址：0x18

复位值：x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX15M[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX15M[15:0]															
rw															

字段	名称	描述
31: 0	RX15M	Rx MB15 掩码位: 0: 过滤器相应位“无关” 1: 过滤器相应位被检查 每位用作 MB15 相应过滤字段的掩码，与 CAN_RXMGMASK 的设置方式相同。 只能在冻结模式下写入，其他模式下被硬件锁定。

33.4.8 CAN_ECR 错误计数器寄存器

偏移地址：0x1C

复位值：0x0000 0000

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

字段	名称	描述
31: 16	Reserved	保留，必须保持复位值
15: 8	RXERRCNT	接收错误计数器 — 计算在接收到的报文中检测到的所有错误

字段	名称	描述
		非冻结模式下，RXERRCNT 计数器为只读
7: 0	TXERRCNT	传输错误计数器 — 计算在传输的报文中检测到的所有错误 非冻结模式下，TXERRCNT 计数器为只读

故障限制状态（CAN_ESR1.FLTCONF）基于 TXERRCNT 和 RXERRCNT 计数器更新。TXERRCNT 和 RXERRCNT 计数器的自加和自减规则见 CAN 协议。

FlexCAN 总线状态转换的基本原则：

- 如果 TXERRCNT 或 RXERRCNT 增加至 ≥ 128 ，则 CAN_ESR1.FLTCONF 字段将会更新，以反映错误被动状态
- 如果 FlexCAN 为错误被动状态，TXERRCNT 或 RXERRCNT 自减至 ≤ 127 ，且另一个计数器已满足该条件，则 CAN_ESR1.FLTCONF 字段将会更新，以反映错误主动状态
- 如果 TXERRCNT 增加至 > 255 ，则 CAN_ESR1.FLTCONF 字段将会更新，以反映总线关闭状态，且可能会产生中断。随后 TXERRCNT 复位为 0
- 如果 FlexCAN 为总线关闭状态，TXERRCNT 与另一个内部计数器级联计数，计数到总线上出现 128 次 11 个连续隐性位时，TXERRCNT 复位为 0。计数的方式为，内部计数器计数 11 位后重新计数；同时 TXERRCNT 增加。当 TXERRCNT 增加至 128 时，CAN_ESR1.FLTCONF 字段将会更新为错误主动状态，且两个错误计数器都复位为 0。在少于 11 个连续隐性位之后出现显性位时，内部计数器复位为 0，且不影响 TXERRCNT 的值
- 如果系统启动期间，只有一个节点在运行，由于应答错误（CAN_ESR1.ACKERR），TXERRCNT 在试图发送的每一个报文中增加。转换为错误被动状态后，TXERRCNT 不再因应答错误而增加。因此，设备不会进入总线关闭状态
- RXERRCNT 增加至 127 以上后不会再增加，即使接收器检测到更多的错误。在下次成功接收报文时，将计数器的值设置在 119 至 127 之间，以恢复错误主动状态

33.4.9 CAN_ESR1 错误和状态寄存器 1

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										ERROVR	Res.	BOFFD ONEINT	SYNCH	TWRNINT	RWRNINT
										rW		rW	r	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMER	STFER	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	Res.
r	r	r	r	r	r	r	r	r	r	r		r	rW	rW	

字段	名称	描述
31: 22	Reserved	保留, 必须保持复位值
21	ERROVR	错误溢出 — 表示在已有错误标志被置位的情况下又发生了错误: 1: 溢出 0: 未溢出 写 1 清零
20	Reserved	保留, 必须保持复位值
19	BOFFDONEINT	总线关闭完成中断: 1: FlexCAN 已完成总线关闭流程 当 Tx 错误计数器 (TXERRCNT) 在 CAN 总线上计数到 128 次 11 个连续的隐性位, 并准备离开总线关闭状态时, BOFFDONEINT 被置位。如果相应的掩码位 (CAN_CTRL2.BOFFDONEMSK) 被置位, 则 CPU 生成中断。 写 1 清零, 写 0 无效。
18	SYNCH	CAN 同步状态: 1: FlexCAN 已同步至 CAN 总线, 并能够参与相应的通信流程 0: FlexCAN 未同步至 CAN 总线 该位只读, 由 FlexCAN 置位和清零。
17	TWRNINT	Tx 警告中断标志: 1: Tx 错误计数器从小于 96 增至超过或等于 96 如果 CAN_MCR.WRNEN 被置位, 当 TXWRN 标志由 0 变 1 时, TWRNINT 被置位, 表示 Tx 错误计数器达到 96。如果相应的掩码位 (CAN_CTRL1.TWRNMSK) 被置位, 则产生中断; 当 CAN_MCR.WRNEN 被清零时, TWRNINT 标志被掩码。CPU 必须在 CAN_MCR.WRNEN 被清零前清除 TWRNINT 标志。否则, 当 CAN_MCR.WRNEN 再次被置位时, TWRNINT 将被置位。 TWRNINT 不会在总线关闭状态置位, 且不会在冻结模式下更新。 写 1 清零, 写 0 无效。
16	RWRNINT	Rx 警告中断标志: 1: Rx 错误计数器从小于 96 增至超过或等于 96 如果 CAN_MCR.WRNEN 被置位, 当 RXWRN 标志由 0 变 1 时, RWRNINT 被置位, 表示 Rx 错误计数器达到 96。如果相应的掩码位 (CAN_CTRL1.RWRNMSK) 被置位, 则产生中断。 当 CAN_MCR.WRNEN 被清零时, RWRNINT 标志被掩码。CPU 必须在 CAN_MCR.WRNEN 被清零前清除 RWRNINT 标志。否则, 当 CAN_MCR.WRNEN 再次被置位时, RWRNINT 将被置位。 RWRNINT 不会在总线关闭状态置位, 且不会在冻结模式下更新。 写 1 清零, 写 0 无效。
15	BIT1ERR	Bit1 错误 — 表示报文的传输位和接收位出现不一致: 1: 至少一个隐性发送位被接收为显性 注: 在仲裁字段或 ACK SLOT, 或节点发送显性的被动错误标志时, BIT1ERR 不会被发送器设置。
14	BIT0ERR	Bit0 错误 — 表示报文的传输位和接收位出现不一致: 1: 至少一个显性发送位被接收为隐性
13	ACKERR	应答错误 — 表示发送节点检测到应答错误 (即在 ACK SLOT 未检测到显性位):

字段	名称	描述
		1: 自上次读取该寄存器后发生了 ACK 错误
12	CRCERR	CRC 校验错误 — 表示接收节点检测到 CRC 错误(即计算的 CRC 值不同于接收到的 CRC): 1: 自上次读取该寄存器后发生了 CRC 错误
11	FRMERR	格式错误 — 表示接收节点检测到格式错误(即固定格式的位字段包括至少一个非法位): 1: 自上次读取该寄存器后发生了格式错误
10	STFERR	填充错误 — 表示接收节点检测到填充错误: 1: 自上次读取该寄存器后发生了填充错误
9	TXWRN	TX 错误警告 — 表示报文传输期间发生重复的错误: 1: TXERRCNT 超过或等于 96 TXWRN 只受 CAN_ECR.TXERRCNT 的值影响, 冻结模式下不会更新。
8	RXWRN	Rx 错误警告 — 表示报文接收期间发生重复的错误: 1: RXERRCNT 超过或等于 96 RXWRN 只受 CAN_ECR.RXERRCNT 的值影响, 冻结模式下不会更新。
7	IDLE	1: CAN 总线为 IDLE 状态 参见表格“FLEXCAN 状态相关位编码”。
6	TX	1: FlexCAN 正在传输报文 0: FlexCAN 未传输报文 参见表格“FLEXCAN 状态相关位编码”。
5: 4	FLTCONF	FlexCAN 的错误状态: 00: 主动错误 01: 被动错误 1x: 总线关闭 如果 CAN_CTRL1.LOM 被置位, 经过延迟后(延迟时间取决于 CAN 位时序), FLTCONF 字段会标志为“被动错误”。同样的延迟会影响 FLTCONF 将更新反映到 CPU 的 CAN_ECR 寄存器, 可能需要一个 CAN 位时间来使两者保持一致。 FLTCONF 受软件复位影响, 但如果 CAN_CTRL1.LOM 被置位, 则 FLTCONF 的复位值仅持续一个 CAN 位, 之后 FLTCONF 指示被动错误。
3	RX	1: FlexCAN 正在接收报文 0: FlexCAN 未接收报文 参见表格“FLEXCAN 状态相关位编码”。
2	BOFFINT	总线关闭中断: 1: FlexCAN 进入总线关闭状态 当 FlexCAN 进入总线关闭状态时, BOFFINT 被置位。如果相应的掩码位(CAN_CTRL1.BOFFMSK)被置位, 则产生中断。 写 1 清零, 写 0 无效。
1	ERRINT	错误中断: 1: 表明 CAN_ESR1 中至少有一个错误标志(BIT1ERR、BIT0ERR、ACKERR、CRCERR、FRMERR 或 STFERR)被置位 如果相应的掩码位(CAN_CTRL1.ERRMSK)被置位, 则产生中断。 写 1 清零, 写 0 无效。
0	Reserved	保留, 必须保持复位值

在单个 CAN 帧中检测到的错误可通过一个或多个错误标志报告。此外，错误报告可以累计，以防 CPU 读取 CAN_ESR1 之前的后续帧中还有错误事件。

建议 CPU 在处理相应位的中断请求时采用以下流程：

- 读取 CAN_ESR1 以确定所有错误条件和状态位。此操作会清除自上次读取访问以来置位的相应位
- 写 1 以清除触发中断请求的中断位
- 若 ERR_OVR 被置位，则写 1 以清除该位

从清除所有错误标记开始，第一个错误事件置位 ERRINT（如果相应的掩码位被置位）。如果在 CPU 处理中断请求前，后续帧发生其他错误事件，则 ERR_OVR 被置位以指示已累计来自不同帧的错误。

表 33-11 FLEXCAN 状态相关位编码

SYNCH	IDLE	TX	RX	FlexCAN 状态
0	0	0	0	未同步至 CAN 总线
1	1	x	x	空闲
1	0	1	0	传输
1	0	0	1	接收

33.4.10 CAN_IMASK1 中断掩码寄存器 1

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUF15TO0M															
rw															

字段	名称	描述
31: 16	Reserved	保留，必须保持复位值
15: 0	BUF15TO0M	MBi 掩码位 — 使能或禁止 MB15 至 MB0 的相应中断： 1：相应的 MB 中断使能 0：相应的 MB 中断禁止 注：如果相应的 CAN_IFLAG1 位被置位，则置位或清零 CAN_IMASK1 可以产生或屏蔽中断请求。

33.4.11 CAN_IFLAG1 中断标志寄存器 1

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUF15TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
rw								rw	rw	rw	rw				rw

字段	名称	描述
31: 16	Reserved	保留，必须保持复位值
15: 8	BUF15TO8I	MBi 中断标志位 — 每个位对应 MB15 至 MB8 1: 相应的缓冲区已成功传输或接收 0: 相应的缓存区未成功传输或接收
7	BUF7I	MB7 中断或 Rx FIFO 溢出: 当 CAN_MCR.RFEN 为 0 时, 1: MB7 完成传输/接收 0: MB7 未完成传输/接收 当 CAN_MCR.RFEN 为 1 (Rx FIFO 使能) 时, 1: Rx FIFO 溢出, 表示信息丢失, 因为 Rx FIFO 已满 0: Rx FIFO 未溢出 Rx FIFO 已满且报文被邮箱捕获时, BUF7I 不会置位。 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF7I。
6	BUF6I	MB6 中断或 Rx FIFO 警告: 当 CAN_MCR.RFEN 为 0 时, 1: MB6 完成传输/接收 0: MB6 未完成传输/接收 当 CAN_MCR.RFEN 为 1 (Rx FIFO 使能) 时, 1: Rx FIFO 几乎为满, 表示未读报文数因接收到新报文而从 4 增加至 5 0: Rx FIFO 未几乎为满 如果未读报文数超过 4 时 BUF6I 被清零, 则 BUF6I 不会再次被置位, 直到 Rx FIFO 中未读报文数降至 4 或更少。 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF6I。
5	BUF5I	MB5 中断或 Rx FIFO 中有可用的帧: 当 CAN_MCR.RFEN 为 0 时, 1: MB5 完成传输/接收 0: MB5 未完成传输/接收 当 CAN_MCR.RFEN 为 1 (Rx FIFO 使能) 时, 1: Rx FIFO 中有可用的帧, 表示 Rx FIFO 中至少有一个帧可读 0: Rx FIFO 中无可用的帧 当 CAN_MCR.RFEN 和 CAN_MCR.DMA 被置位时, BUF5I 用于生成 DMA 请求, 且 CPU 不能通过写 1 清除该位。 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF5I。
4: 1	BUF4TO1I	MB4 ~ 1 中断或预留: 当 CAN_MCR.RFEN 为 0 时,

字段	名称	描述
		1: 相应的缓存区成功完成传输或接收 0: 相应的缓存区未成功完成传输或接收 当 CAN_MCR.RFEN 为 1 (Rx FIFO 使能) 时, BUF4TO1I 为预留位 注: 当 CAN_MCR.RFEN 被 CPU 写操作改变时, FlexCAN 会清除 BUF4TO1I。
0	BUF0I	MB0 中断或清除 Rx FIFO 当 CAN_MCR.RFEN 为 0 时, 1: MB0 成功完成传输或接收 0: MB0 未成功完成传输或接收 当 CAN_MCR.RFEN 为 1 (Rx FIFO 使能) 时, BUF0I 触发 FIFO 清除操作, 该操作清空 Rx FIFO 的内容。执行该操作前, CPU 必须处理所有 Rx FIFO 相关的 IFLAG。当 CAN_MCR.DMA 被置位 (DMA 使能) 时, 该操作还会清除 BUF5I 标志并中止 DMA 请求。FIFO 清除操作发生在 CPU 写 1 到 BUF0I 时, 仅在冻结模式下发生, 其他情况下该位被硬件锁定。

每个缓冲区都有一个对应的中断标志位, 每次成功传输或接收, 都会置位相应的 CAN_IFLAG1 位。如果相应的 CAN_IMASK1 位被置位, 则会产生中断。中断标志必须通过写 1 清零, 写 0 无效。

更新 CAN_MCR.MAXMB 字段前, CPU 必须处理 MB 编号大于 CAN_MCR.MAXMB 更新值的 CAN_IFLAG1 位, 否则标志位会保持置位, 且与可用的 MB 数不一致。

Rx FIFO DMA 使能后的例外情况如下所述:

33.4.11.1 Rx FIFO

CAN_MCR.RFEN = 1 且 CAN_MCR.DMA = 0 时:

- BUF7I、BUF6I 和 BUF5I 表示 FIFO 的运行条件
- BUF0I 用于清空 FIFO
- BUF4I 至 BUF1I 预留

置位 CAN_MCR.RFEN 前, CPU 必须处理 Rx FIFO 区域被置位的 IFLAG (参见章节“Rx FIFO 结构”)。否则, 这些 IFLAG 将错误地显示相关 MB (现在用于 FIFO) 有内容需要处理。

当 CAN_MCR.RFEN 被清零时, FIFO 标志必须被清零。当选定的 CAN_CTRL2.RFFN 值使 Rx FIFO 过滤器超出 MB7 时, 同样需要注意。例如当 RFFN 为 0x3 时, MB0 ~ 13 被 Rx FIFO 过滤器占据, 相应的 IFLAG 位必须被清零。

33.4.11.2 Rx FIFO DMA

CAN_MCR.RFEN = 1 且 CAN_MCR.DMA = 1 (使能 Rx FIFO DMA 功能) 时:

- BUF7I、BUF6I 和 BUF4I ~ BUF1I 未使用
- BUF5I 表示 Rx FIFO 的运行条件



- BUF5I 不生成 CPU 中断，而是生成 DMA 请求
- BUF0I 用于清空 FIFO

当 CAN_MCR.DMA 被置位时，不考虑 Rx FIFO 区域的 IMASK1 位。此外，当 DMA 使能时，CPU 不能清零 BUF5I。置位 CAN_MCR.DMA 前，CPU 必须处理 Rx FIFO 区域被置位的 IFLAG。当 CAN_MCR.DMA 被清零时，Rx FIFO 必须为空。

33.4.12 CAN_CTRL2 控制寄存器 2

CAN_CTRL2 寄存器为冻结模式下的内存写访问、扩展 Rx FIFO 过滤器数量、调整 FlexCAN 内部流程（如匹配和仲裁）提供控制位。

偏移地址：0x34

复位值：0x00B0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	BOFFD ONEMSK K	Res.		RFFN				TASD				MRP		RRS	EACEN
	rw			rw				rw				rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

字段	名称	描述
31	Reserved	保留，必须保持复位值
30	BOFFDONEMSK	总线关闭完成中断（CAN_ESR1.BOFFDONEINT）掩码： 1：总线关闭完成中断使能 0：总线关闭完成中断禁止
29: 28	Reserved	保留，必须保持复位值
27: 24	RFFN	Rx FIFO 过滤器数量，参见表格“Rx FIFO 过滤器数量” RFFN 的值不能使 Rx FIFO 和 Rx FIFO ID 过滤器占有的 MB 数超过由 CAN_MCR.MAXMB 定义的邮箱数量。 每组八个过滤器占用相当于两个 MB 的内存空间，因此过滤器越多可用的邮箱越少。 考虑到 Rx FIFO 占用了原本为 MB0 ~ 5 预留的内存空间，RFFN 的值应不大于可用内存字的数量： $(\text{SETUP\_MB} - 6) \times 4$ 其中：SETUP_MB 为最大邮箱数 16 和 CAN_MCR.MAXMB 中的较小值。 剩余可用邮箱数量： $(\text{SETUP\_MB} - 8) - (\text{RFFN} \times 2)$ 如果 RFFN 设定的 Rx FIFO 过滤器数量超过 SETUP_MB，则超出部分无效。 注：最后一个剩余可用邮箱的编号为最大邮箱数 16-1 和 CAN_MCR.MAXMB 之间的最小值。 如果 Rx 私有掩码寄存器未使能，则所有 Rx FIFO 过滤器受 Rx FIFO 全局掩码影响。 RFFN 只能在冻结模式下写入，其他模式下被硬件锁定。

字段	名称	描述
23: 19	TASD	Tx 仲裁启动延迟 — 定义从 CAN 总线的 CRC 字段首位开始, Tx 仲裁过程起始点延迟的 CAN 位数, 参见章节“Tx 仲裁启动延迟”。 TASD 只能在冻结模式下写入, 其他模式下被硬件锁定。
18	MRP	邮箱接收优先级: 1: 匹配从邮箱开始, 若没有发生匹配, 则在 Rx FIFO 中继续匹配 0: 匹配从 Rx FIFO 开始, 在邮箱继续 MRP 只能在冻结模式下写入, 其他模式下被硬件锁定。
17	RRS	远程请求存储: 1: 远程帧被存储 0: 产生远程响应帧 如果 RRS 被置位, 则将远程帧提交至匹配过程, 并以和数据帧相同的方式存储在相应的报文缓冲区中, 不会自动生成远程响应帧。 如果 RRS 被清零, 则将远程帧提交到匹配过程, 且如果一个代码为 0b1010 的报文缓冲区具有相同的 ID, 则自动生成远程响应帧。 RRS 只能在冻结模式下写入, 其他模式下被硬件锁定。
16	EACEN	Rx 邮箱全帧仲裁字段比较: 1: 匹配过程比较 Rx 邮箱过滤器的 IDE、RTR 位和接收帧的相应位, 掩码位适用 0: 匹配过程比较 Rx 邮箱过滤器的 IDE 位, 但不比较 RTR, 与掩码位无关 EACEN 不影响 Rx FIFO 的匹配。 EACEN 只能在冻结模式下写入, 其他模式下被硬件锁定。
15: 0	Reserved	保留, 必须保持复位值

表 33-12 Rx FIFO 过滤器数量

RFFN[3: 0]	Rx FIFO 过滤元素数量	Rx FIFO 和 ID 过滤表占用的报文缓冲区	剩余可用邮箱	受 Rx 私有掩码影响的 Rx FIFO ID 过滤表元素	受 Rx FIFO 全局掩码影响的 Rx FIFO ID 过滤表元素
0x0	8	MB0 ~ 7	MB8 ~ 15	元素 0 ~ 7	无
0x1	16	MB0 ~ 9	MB10 ~ 15	元素 0 ~ 9	元素 10 ~ 15
0x2	24	MB0 ~ 11	MB12 ~ 15	元素 0 ~ 11	元素 12 ~ 23
0x3	32	MB0 ~ 13	MB14 ~ 15	元素 0 ~ 13	元素 14 ~ 31
0x4	40	MB0 ~ 15	无	元素 0 ~ 15	元素 16 ~ 39

33.4.13 CAN_ESR2 错误和状态 2 寄存器

偏移地址: 0x38

复位值: 0x0000 0000

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

	r	r	
字段	名称	描述	
31: 23	Reserved	保留，必须保持复位值	
22: 16	LPTM	最低优先级的 Tx 邮箱 如果 CAN_ESR2.VPS 被置位，则 LPTM 指示编号最小的非活动邮箱（见 CAN_ESR2.IMB 位描述）。如果没有非活动邮箱，则所示邮箱取决于 CAN_CTRL1.LBUF 的值： CAN_CTRL1.LBUF 为 0 时，指示具有最大仲裁值的邮箱（参见章节“最高优先级有效优先”）； CAN_CTRL1.LBUF 被置位时，指示编号最大的活动 Tx 邮箱 如果 Tx 邮箱正在被传输，则 LPTM 不考虑该邮箱。如果 CAN_ESR2.IMB 未被置位，且帧被成功传输，则 LPTM 更新为其邮箱号。	
15	Reserved	保留，必须保持复位值	
14	VPS	CAN_ESR2.IMB 和 CAN_ESR2.LPTM 的内容当前是否有效： 1: IMB 和 LPTM 的内容有效 0: IMB 和 LPTM 的内容无效 VPS 在每次完成 Tx 仲裁过程时被置位，除非 CPU 向已被扫描的邮箱写入 C/S（在 Tx 仲裁期间，位于 Tx 仲裁点后）。如果没有非活动邮箱且只有一个 Tx 邮箱被传输，则 VPS 不会被置位。 VPS 在每次启动 Tx 仲裁进程或向任何邮箱写入 C/S 时被清零。 注：CPU 向被中止机制锁定的 MB 写入 C/S 不影响 VPS。当 CAN_MCR.AEN 被置位时，向正在被传输的 MB 的 C/S 写入中止代码（中止待处理），或试图写 CAN_IFLAG1 被置位的 Tx MB，写操作都会被锁定。	
13	IMB	非活动邮箱 如果 CAN_ESR2.VPS 被置位，IMB 指示是否有任何非活动邮箱（代码字段为 0b1000 或 0b0000）： 1: 如果 CAN_ESR2.VPS 被置位，则至少有一个非活动邮箱。CAN_ESR2.LPTM 为第一个邮箱的编号 0: 如果 CAN_ESR2.VPS 被置位，则 CAN_ESR2.LPTM 不是非活动邮箱 IMB 在下列情况下被置位： 仲裁期间，如果 CAN_ESR2.LPTM 被找到且非活动 如果 CAN_ESR2.MB 未被置位，且帧被成功传输 IMB 在仲裁开始时被清零（参见章节“仲裁过程”）。 注：如果 MB 被成功传输且 CAN_ESR2.IMB=0（无非活动邮箱），则 CAN_ESR2.VPS 和 CAN_ESR2.IMB 被置位，该 MB 的编号被写入 CAN_ESR2.LPTM。	
12: 0	Reserved	保留，必须保持复位值	

33.4.14 CAN_CRCCR CRC 寄存器

Tx 中断标志被置位的同时, CAN_CRCCR 被更新。

偏移地址: 0x44

复位值: 0x0000 0000

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

字段	名称	描述
31: 23	Reserved	保留，必须保持复位值
22: 16	MBCRC	CRC 邮箱 — 表示与 CAN_CRCCR.TXCRC 字段对应的邮箱编号
15	Reserved	保留，必须保持复位值
14: 0	TXCRC	传输的 CRC 值 — 表示最后一次传输的报文的 CRC 值

33.4.15 CAN_RXFGMASK Rx FIFO 全局掩码寄存器

CAN_RXFGMASK 位于 RAM 空间。Rx FIFO 使能时，用于掩码 Rx FIFO ID 过滤表元素（依照 CAN_CTRL2.RFFN 的设定，没有相应 CAN_RXIMR 的元素）。

偏移地址：0x48

复位值：x

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

字段	名称	描述
31: 0	FGM	Rx FIFO 全局掩码位 — 掩码 ID 过滤表元素的相应位： 1：过滤表的相应位被检查 0：过滤表的相应位“don't care”，细节见下表 只能在冻结模式下写入，其他模式下被硬件锁定。

表 33-13 Rx FIFO 全局掩码位和 IDF 字段的对应关系

Rx FIFO ID 过滤表元素格式 (CAN_MCR.IDAM)	ID 接收过滤器字段					
	RTR	IDE	RXIDA	RXIDB ⁽¹⁾	RXIDC ⁽²⁾	预留
A	FGM[31]	FGM[30]	FGM[29: 1]	—	—	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	—	FGM[29: 16] , FGM[13: 0]	—	—

Rx FIFO ID 过滤表元 素格式 (CAN_MCR.IDAM)	ID 接收过滤器字段					
	RTR	IDE	RXIDA	RXIDB ⁽¹⁾	RXIDC ⁽²⁾	预留
C	—	—	—	—	FGM[31: 24], FGM[23: 16], FGM[15: 8], FGM[7: 0]	—

1) 如果选用格式 B, 则接收帧 ID 只有高 14 位与 Rx FIFO 过滤器相比较

2) 如果选用格式 C, 则接收帧 ID 只有高 8 位与 Rx FIFO 过滤器相比较

33.4.16 CAN_RXFIR Rx FIFO 信息寄存器

偏移地址: 0x4C

复位值: x

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

字段	名称	描述
31: 9	Reserved	保留, 必须保持复位值
8: 0	IDHIT	ID 接收过滤命中指示 — 指示被 Rx FIFO 中接收到的报文命中的 ID 接收过滤器 (IDAF) 如果多个过滤器与接收到的报文 ID 相匹配, 则指示匹配过程中第一个匹配的 IDAF (最小编号)。 IDHIT 仅在 CAN_IFLAG1.BUF5I 被置位时有效。

CPU 通过 CAN_RXFIR 访问位于 RAM 的 RXFIR FIFO 的输出。当一个新报文被移入 Rx FIFO 时, FlexCAN 会写该寄存器; 当 Rx FIFO 的输出被下一条报文更新时, 该寄存器的输出也被更新。CAN_RXFIR 读取说明参见章节“Rx FIFO”。

33.4.17 CAN_CBT CAN 位时序寄存器

CAN_CBT 寄存器是 CAN_CTRL1 中 CAN 位时序变量的扩展。

偏移地址: 0x50

复位值: 0x0000 0000

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

EPROPSEG	EPSEG1	EPSEG2
rw	rw	rw

字段	名称	描述
31	BTF	位时序格式 — 使用扩展的 CAN 位时序字段 EPRES DIV、EPROPSEG、EPSEG1、EPSEG2 和 ERJW，取代 CAN_CTRL1 定义的 CAN 位时序变量： 1：使能扩展的位时序定义 0：禁止扩展的位时序定义 BTF 只能在冻结模式写入，其他模式下被硬件锁定。
30: 21	EPRES DIV	扩展的预分频器分频系数 当 CAN_CBT.BTF 被置位时，EPRES DIV 定义了 PE 时钟频率和串行时钟 (Sclock) 频率之间的比率。EPRES DIV 扩展了 CAN_CTRL1.PRES DIV 的数值范围。 $Sclock \text{ 频率} = PE \text{ 时钟} / (EPRES DIV + 1)$ Sclock 周期定义了 CAN 协议的 Tq。对于复位值，Sclock 频率等于 PE 时钟频率（参见章节“协议时序”）。 当 CAN_CBT.BTF 为 0 时，EPRES DIV 无效。 EPRES DIV 只能在冻结模式下写入，其他模式下被硬件锁定。
20: 16	ERJW	扩展的重同步补偿 当 CAN_CBT.BTF 被置位时，ERJW 定义一次重同步补偿的最大 Tq。ERJW 扩展了 CAN_CTRL1.RJW 的数值范围。 重同步补偿宽度 = ERJW + 1, Tq = Sclock 周期 当 CAN_CBT.BTF 被为 0 时，ERJW 无效。 ERJW 只能在冻结模式下写入，其他模式下被硬件锁定。
15: 10	EPROPSEG	扩展传播段 当 CAN_CBT.BTF 被置位时，EPROPSEG 定义传播段的位时间长度。EPROPSEG 扩展了 CAN_CTRL1.PROPSEG 的数值范围。 传播时间 = (EPROPSEG + 1) × Tq, Tq = Sclock 周期 当 CAN_CB.BTF 为 0 时，EPROPSEG 无效。 EPROPSEG 只能在冻结模式下写入，其他模式下被硬件锁定。
9: 5	EPSEG1	扩展相位段 1 当 CAN_CBT.BTF 被置位时，EPSEG1 定义位时间中相位段 1 的长度。EPSEG1 扩展了 CAN_CTRL1.EPSEG1 的数值范围。 相位段 1 = (EPSEG1 + 1) × Tq, Tq = Sclock 周期 当 CAN_CBT.BTF 为 0 时，EPSEG1 无效。 EPSEG1 只能在冻结模式下写入，其他模式下被硬件锁定。
4: 0	EPSEG2	扩展相位段 2 当 CAN_CBT.BTF 被置位时，EPSEG2 定义位时间中相位段 2 的长度。EPSEG2 扩展了 CAN_CTRL1.PSEG2 的数值范围。 相位段 2 = (EPSEG2 + 1) × Tq, Tq = Sclock 周期 当 CAN_CBT.BTF 为 0 时，EPSEG2 无效。 EPSEG2 只能在冻结模式下写入，其他模式下被硬件锁定。

注：CAN_CTRL1 和 CAN_CBT 的 CAN 位变量存储在相同的寄存器内。

33.4.18 CAN_RXIMRn Rx 私有掩码寄存器

CAN_RXIMRn 位于 RAM 中，必须在任何接收之前进行初始化。用于存储 Rx MB 和 Rx FIFO 的 ID 过滤接收掩码。

偏移地址：0x880 + 4 * n

复位值：x

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

字段	名称	描述
31: 0	MI	私有掩码位： 1: 过滤器的相应位被检查 0: 过滤器的相应位“don't care” Rx FIFO 禁止（CAN_MCR.RFEN = 0）时，一一对应地为每个可用的 Rx 邮箱提供私有掩码，见 RXMGMASK 寄存器描述； Rx FIFO 使能（CAN_MCR.RFEN = 1）时，一一对应地为每个 Rx FIFO ID 过滤表元素提供私有掩码，见 RXFGMASK 寄存器描述。具体取决于 CAN_CTRL2.RFFN 的设置（参见章节“Rx FIFO”）。 只能在冻结模式下由 CPU 访问，不受复位影响。

33.4.19 报文缓冲区结构

下图为 FlexCAN 所使用的报文缓冲区结构，包括 CAN 2.0B 的两种帧格式：扩展帧（29 位 ID）和标准帧（11 位 ID）。每个报文缓冲区（MB）由 16 字节组成，其中包括 8 字节的数据。邮箱使用 0x80 ~ 0x17F 的内存区域。

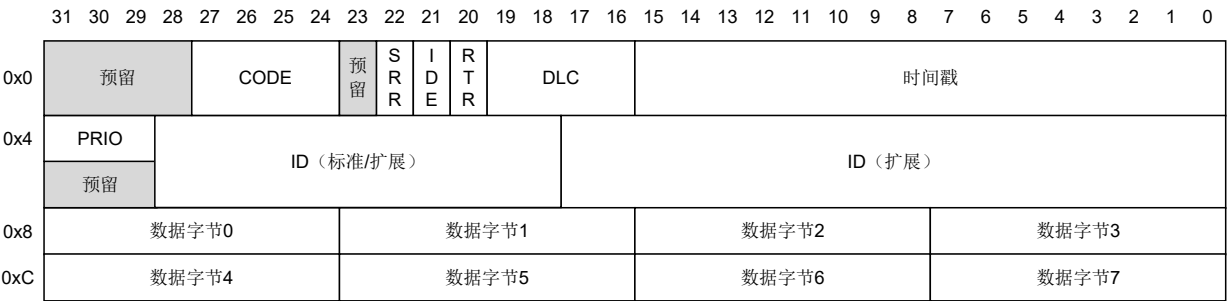


图 33-8 信息寄存器结构示例

- CODE — 报文缓冲区代码

CODE 字段可以被 CPU 和 FlexCAN 读写，用作报文缓冲区匹配和仲裁过程的一部分。编码如下表所示。



表 33-14 接收缓冲区代码

CODE	接收到新帧前的 Rx code	SRV ⁽¹⁾	成功接收后的 Rx code ⁽²⁾	RRS ⁽³⁾	注解
0b0000: INACTIVE MB 未激活	INACTIVE	—	—	—	MB 不参与匹配过程
0b0100: EMPTY MB 激活且为空	EMPTY	—	FULL	—	成功接收到帧时（在移入过程之后），CODE 自动更新为 FULL
0b0010: FULL MB 满	FULL	Yes	FULL	—	读取 C/S 并解锁 MB（SRV）不会使代码返回 EMPTY。如果在 MB 被处理后，有新帧被移入 MB，则代码仍保持 FULL。细节参见章节“匹配过程”。
		No	OVERRUN	—	如果 MB 为 FULL，新帧在 CPU 使用该 MB 前被移入，则 CODE 会自动更新为 OVERRUN。细节参见章节“匹配过程”。
0b0110: OVERRUN MB 被覆盖到满的缓冲区中	OVERRUN	Yes	FULL	—	如果 CODE 为 OVERRUN 且 CPU 已处理了 MB，则当一个新帧移动到 MB 时，代码返回 FULL。
		No	OVERRUN	—	如果 CODE 为 OVERRUN，且必须移入另一个新帧，则 MB 将再次被覆盖，代码将保持 OVERRUN。细节参见章节“匹配过程”。
0b1010: RANSWER ⁽⁴⁾ 帧被配置为识别远程帧并回复响应帧	RANSWER	—	TANSWER (1110)	0	远程应答被配置用于识别接收到的远程帧。之后，MB 被配置为发送响应帧，代码自动变更为 TANSWER（0b1110）。细节参见章节“匹配过程”。如果 CAN_CTRL2.RRS 被清零，则在收到相同 ID 的远程帧时发送响应帧。
		—	—	1	匹配与仲裁阶段，忽略此代码。细节参见章节“匹配过程”。
CODE[0] = 1: BUSY FLexCAN 正在更新 MB，CPU 禁止访问	BUSY ⁽⁵⁾	—	FULL	—	表示正在更新 MB。自动清零且不影响下一个 CODE。
		—	OVERRUN	—	

1) SRV: 已处理的 MB。MB 被读取，并通过读 CAN_TIMER 或其他 MB 被解锁

2) 帧被移入 MB 则认为该帧被成功接收。参见章节“移入”

3) 远程请求存储位，细节参见寄存器 CAN_CTRL2

4) CODE 0b1010 不被认作 Tx，具有该 CODE 的 MB 不应被中止

5) 对于 Tx MB，读取时应忽略 BUSY 位，除非 CAN_MCR.AEN 被置位。如果该位被置位，则相应的 MB 不参与匹配过程

表 33-15 发送缓冲区代码

CODE	发送帧前的 Tx code	MB RTR	成功传输后的 Tx code	注解
0b1000: INACTIVE MB 未激活	INACTIVE	—	—	MB 不参与匹配过程
0b1001: ABORT MB 被丢弃	ABORT	—	—	MB 不参与匹配过程
0b1100: DATA MB 为 Tx 数据帧 (MB RTR 必须为 0)	DATA	0	INACTIVE	无条件地发送数据帧一次。传输后, MB 自动返回 INACTIVE 状态
0b1100: REMOTE MB 为远程帧 (MB RTR 必须为 1)	REMOTE	1	EMPTY	无条件地发送远程帧一次。传输后, MB 自动转换为具有相同 ID 的接收空 MB
0b1110: TANSWER MB 为远程帧的响应帧	TANSWER	—	RANSWER	中间代码, 匹配远程帧后由 CHI 自动写入 MB。远程响应帧将无条件地传输一次, 之后代码将自动返回到 RANSWER (0b1010)。 CPU 写该 Code 可以产生相同的效果。远程响应帧可以是数据帧, 也可以是另一个远程帧, 取决于 RTR 位的值。参见章节“匹配过程”和“仲裁过程”。

● SRR — 替代远程请求

1: 扩展帧格式传输时, 必须使用隐性位

0: 扩展帧格式传输时, 显性位无效

固定的隐性位, 只用于扩展帧格式。传输时 (发送缓冲区) 该位必须设置为 1, 且将会和从 CAN 总线上接收到的值一起存储于接收缓冲区。该位可以被接收为隐性或显性, 如果 FlexCAN 以显性位接收, 则认为仲裁丢失。

● IDE — ID 扩展位

1: 扩展帧

0: 标准帧

● RTR — 远程传输请求

1: 如果是发送 MB, 则表示当前 MB 可能有一个远程帧待发送; 如果是接收 MB, 则接收到的远程帧将会被存储;

0: 表示当前的 MB 中有一个数据帧待传输。在接收 MB 中, 可能会被用于匹配过程。

如果 FlexCAN 传输 1 (隐性), 接收到 0 (显性), 则认为仲裁丢失。如果 RTR 传输 0 (显性), 接收到 1 (隐性), 则认为是位错误。如果接收到的值与发送值相同, 则被认为是一次成功的位传输。

● DLC — 数据字节长度

该 4 位字段为发送/接收数据的长度 (以字节为单位), 位于偏移地址为 0x8 到 0xF 的 MB 空间。接收阶段, 该字段由 FlexCAN 写入, 从接收帧的 DLC (Data Length Code) 字段复制而得; 传输阶段, 该字段由 CPU 写入, 且与要传输的帧的 DLC 字段相对应。当 RTR = 1 时, 被传输的帧为远程帧, 不

包含数据字段（DLC 字段的设置无效，参见表格“有效数据字节”）。

- TIME STAMP — 自由运行计时器时间戳

该 16 位字段为自由运行计时器的复制，当标识符字段开头出现在 CAN 总线上时进行捕获。

- PRIO — 本地优先级

该 3 位字段只有当 MCR.LPRIO_EN 被置位时才有效，且只针对传输邮箱。用于附加到 ID 来定义传输优先级，不会被传输。参见章节“仲裁过程”。

- ID — 帧标识符

标准帧格式，只有高 11 位（28~18）用于识别接收或发送帧，忽略低 18 位。扩展帧格式，所有位都用于识别传输或接收帧。

- DATA BYTE0 ~ 7 — 数据字段

数据帧最多可以使用 8 个字节，取决于为 MB 选择的有效负载大小。从总线上接收到的帧以该帧被接收时的格式进行存放。只有 n 小于 DLC 时，DATA BYTE（n）才有效。

表 33-16 有效数据字节

DLC	有效数据字节
0	无
1	数据字节 0
2	数据字节 0 ~ 1
3	数据字节 0 ~ 2
4	数据字节 0 ~ 3
5	数据字节 0 ~ 4
6	数据字节 0 ~ 5
7	数据字节 0 ~ 6
8	数据字节 0 ~ 7

33.4.20 MB 内存映射

FlexCAN 内存缓冲区的内存映射如下表所示。

表 33-17 MB 内存映射

MB	偏移地址 (8 字节负载)
MB0	0x0080
MB1	0x0090
MB2	0x00A0
MB3	0x00B0
MB4	0x00C0
MB5	0x00D0
MB6	0x00E0
MB7	0x00F0
MB8	0x0100

MB	偏移地址 (8 字节负载)
MB9	0x0110
MB10	0x0120
MB11	0x0130
MB12	0x0140
MB13	0x0150
MB14	0x0160
MB15	0x0170

33.4.21 Rx FIFO 结构

当 CAN_MCR.RFEN 被置位时，Rx FIFO 引擎使用从 0x80 至 0xDC 的内存空间（通常被 MB 0 ~ 5 占用）。

0x80 ~ 0x8C 包含 Rx FIFO 的输出，必须作为报文消息缓冲区由 CPU 读取。Rx FIFO 输出包含已被接收但未被读取的最旧的报文；0x90 ~ 0xDC 为 Rx FIFO 引擎内部使用预留。

一个额外的内存空间，从 0xE0 开始，可扩展至 0x17C（通常被 MB6 ~ 15 占用），取决于 CAN_CTRL2.RFFN 的设置，包括 ID 过滤表（可配置 8~40 个表元素），该表指定了 Rx FIFO 接收帧的过滤条件。

复位时，可变的 ID 过滤表内存空间默认为 0xE0，仅可扩展至 0xFC，对应于 RFFN = 0 时的 MB 6 ~ 7。

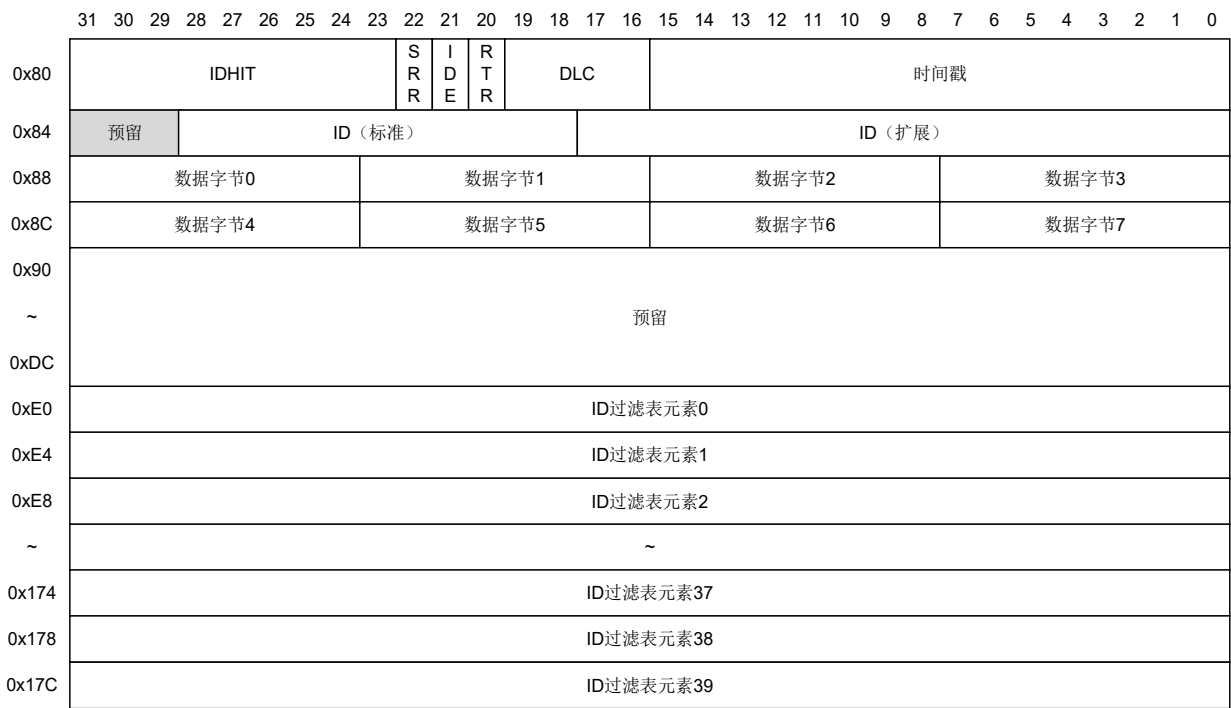


图 33-9 Rx FIFO 结构

各 ID 过滤表元素占有整个 32 位字，且可由 1、2 或 4 个标识符接收过滤器（IDAF）组成，取决



于 CAN_MCR.IDAM 的设置。下图显示了三个不同格式的 ID 表元素，所有表元素必须具有相同的格式。
参见章节“Rx FIFO”。

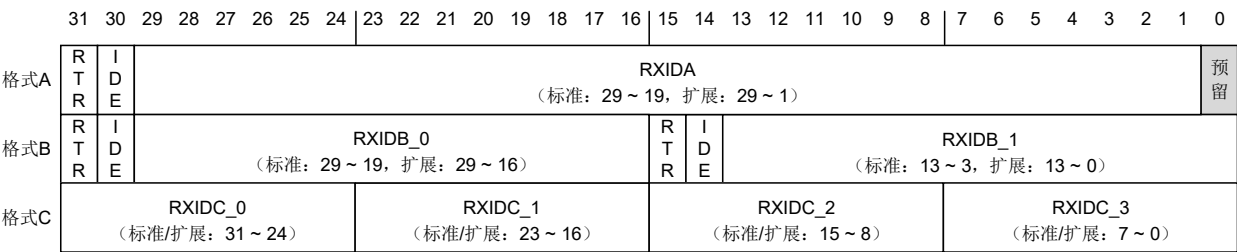


图 33-10 ID 过滤表结构

- RTR — 远程帧，指定如果远程帧匹配目标 ID，是否被接收到 Rx FIFO 中：

- 1: 接收远程帧，拒绝数据帧
- 0: 拒绝远程帧，接收数据帧

- IDE — 扩展帧，指定 Rx FIFO 是否接收匹配目标 ID 的扩展帧或标准帧：

- 1: 接收扩展帧，拒绝标准帧
- 0: 拒绝扩展帧，接收标准帧

- RXIDA — Rx 帧标识符（格式 A）

指定作为 Rx FIFO 接收条件的 ID。标准帧格式，只有高 11 位（29 ~ 19）用作帧标识符；扩展帧格式，所有位都被使用。

- RXIDB_0、RXIDB_1 — Rx 帧标识符（格式 B）

指定作为 Rx FIFO 接收条件的 ID。标准帧格式，只有高 11 位（29 ~ 19 和 13 ~ 3）用作帧标识符；扩展帧格式，所有 14 位都与接收到的 ID 的高 14 位比较。

- RXIDC_0、RXIDC_1、RXIDC_2、RXIDC_3 — Rx 帧标识符（格式 C）

指定作为 Rx FIFO 接收条件的 ID。标准帧和扩展帧格式下，所有的 8 位都与接收到的 ID 的高 8 位相比较。

- IDHIT — 标识符接收过滤器命中指示

9 位的 IDHIT 字段表示 Rx FIFO 输出的接收报文命中的标识符接收过滤器。参见章节“Rx FIFO”。



34 USB_FS 全速 USB 控制器

34.1 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 34-1 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。

34.2 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

34.3 功能描述

34.3.1 功能框图

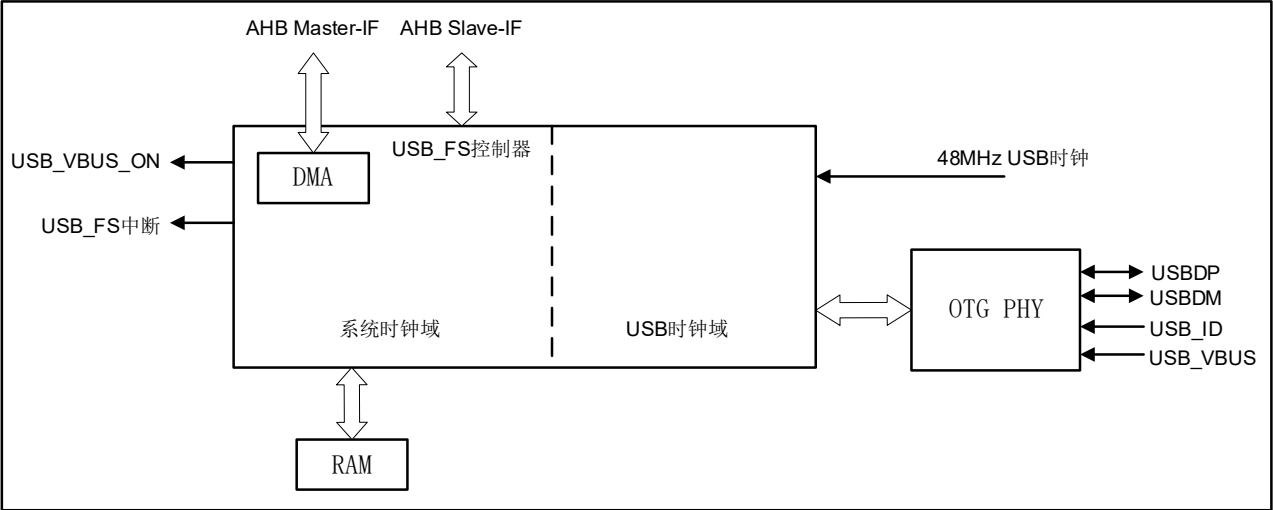


图 34-1 USB_FS 功能框图

34.3.2 信号描述

表 34-2 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

34.3.3 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。

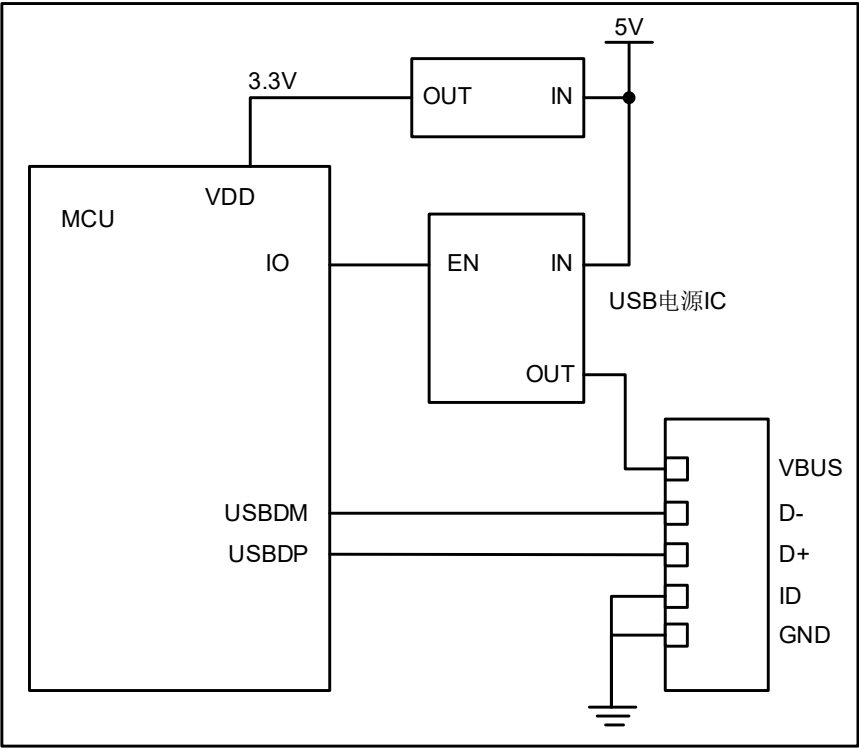


图 34-2 仅主机硬件连接示意图

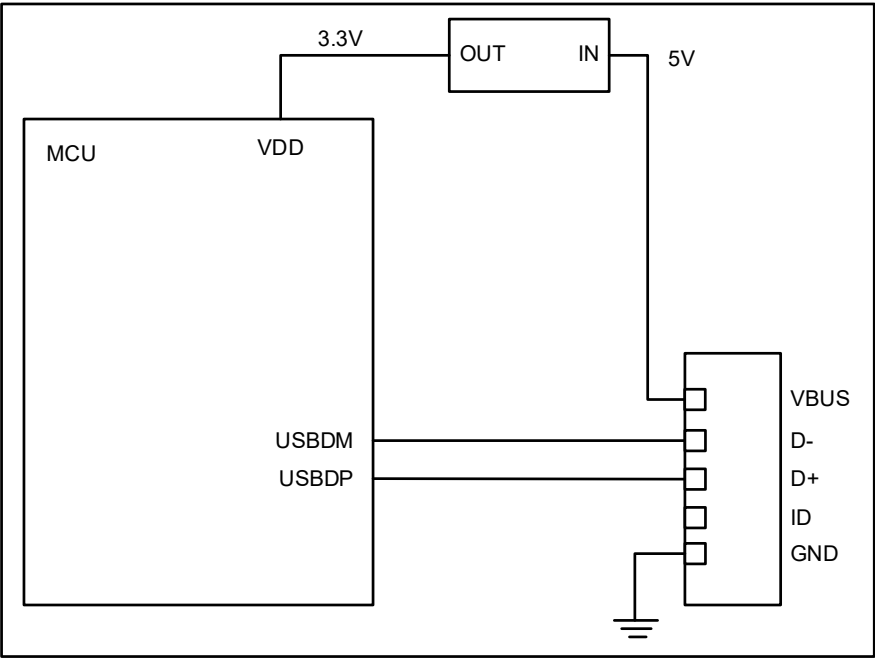


图 34-3 仅从设备硬件连接示意图

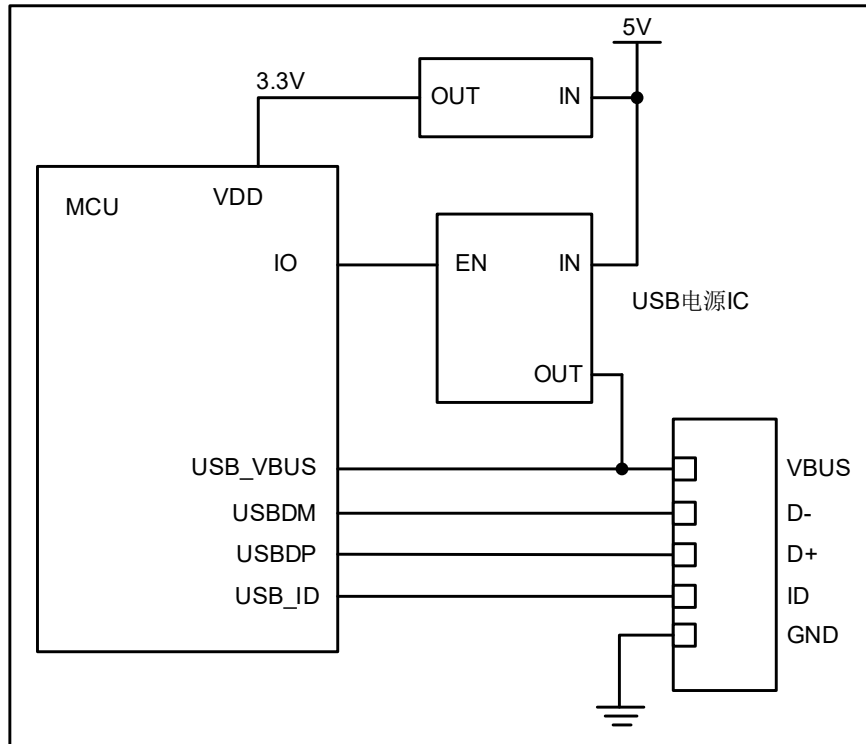


图 34-4 OTG 硬件连接示意图

34.3.4 USB_FS 软件编程接口

34.3.4.1 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。

34.3.4.2 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。

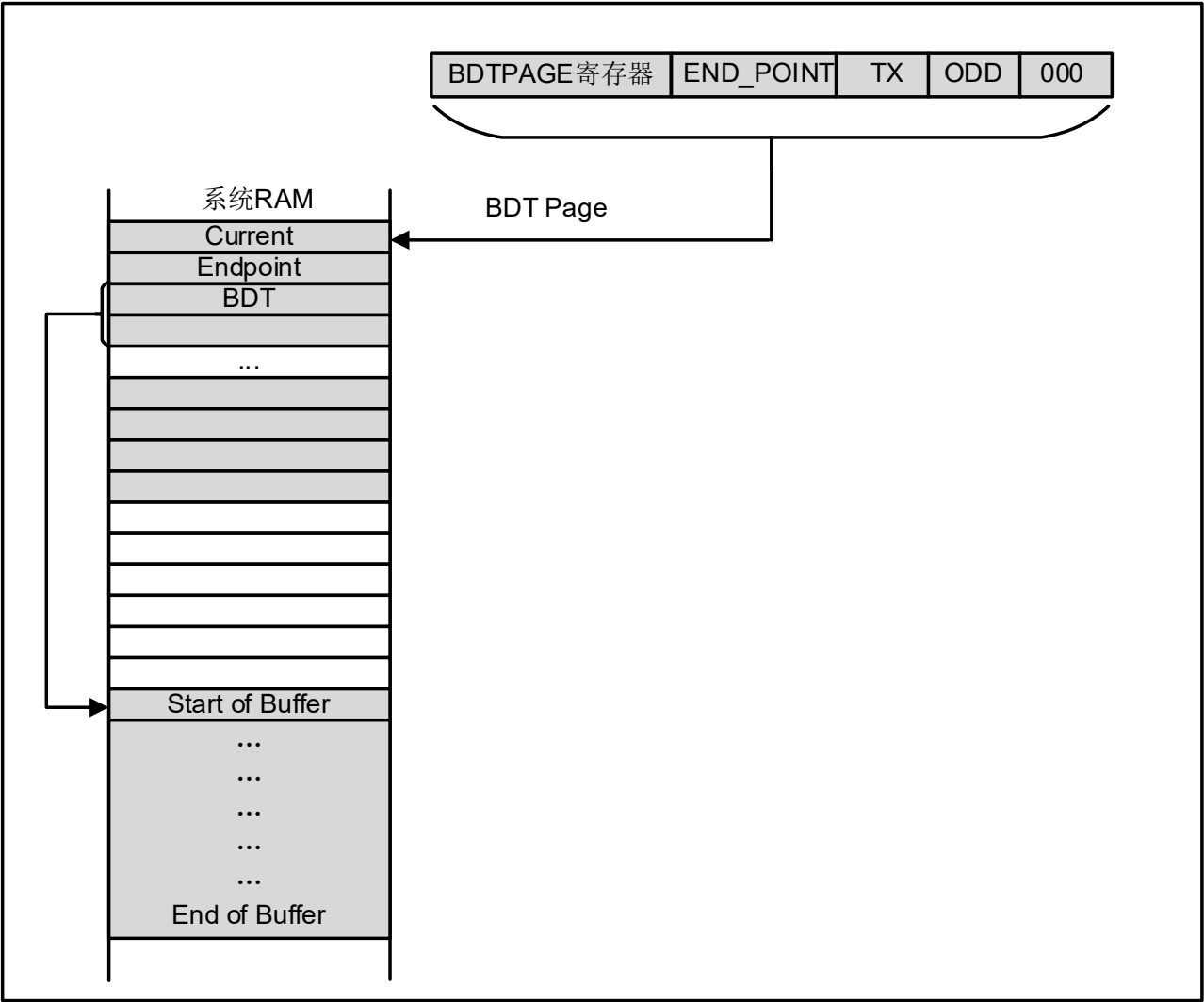


图 34-5 缓冲区描述表

34.3.4.3 缓冲区描述表寻址机制

要通过 USB_FS 或微处理器访问端点数据，必须理解缓冲区描述符表的寻址机制。如前所述，每个 USB 端点方向需要 16 个字节。BDT 页面寄存器指向 BDT 的起始位置。BDT 起始地址位于系统内存的 512 字节的边界上。所有启用的 TX 和 RX 端点的 BD 条目都被索引到 BDT 中，以便通过 USB_FS 或微处理器轻松访问。

当 USB_FS 在启用的端点上接收到 USB 令牌时，它将使用其集成的内置 DMA 控制器来询问 BDT。USB_FS 必须读取相应的端点 BD 条目并确定它是否拥有系统存储器中的 BD 和相应的缓冲区的访问权限。

为计算 BDT 的入口地址，BDTPAGE 寄存器与当前端点以及 TX 和 ODD 字段连接在一起形成一个 32 位地址。该地址机制如下表所示：



表 34-3 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 34-4 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 34-5 BD 格式

31: 26	25: 16	15:8	7	6	5	4	3	2	1	0
Res.	BC	Res.	OWN	DATA0/1	KEEP/ TOK_PID[3]	NINC/ TOK_PID[2]	DTS/ TOK_PID[1]	BDT_STALL/ TOK_PID[0]	0	0
Buffer Address (32-Bits)										

Field	说明
BC	字节计数位。USB_FS SIE 将在完成指定字节的 RX 传输后更改此字段。
OWN	如果 OWN = 1, USB_FS 可以独占访问 BD。如果 OWN = 0, 则微处理器可以独占访问 BD。这个 OWN 位决定了谁当前拥有缓冲区。 SIE 在完成接收一个令牌时一般将 '0' 写入该位, 除非 KEEP = 1。 当 OWN = 0 时, USB_FS 忽略 BD 中的所有其他字段, 微处理器可以访问整个 BD。BD 的这个字节应该始终是微处理器在初始化 BD 时更新的最后一个字节。 一旦 BD 由 USB_FS 拥有, 微处理器不应该以任何方式改变它。
DATA0/1	发送或接收 DATA0 或 DATA1 包。它不由 USB_FS 改变。
KEEP	如果 KEEP 等于 1, 一旦 OWN 位被置'1', BD 将永远由 USB_FS 持有。KEEP 必须等于 '0' 才能允许 USB_FS 在令牌处理完成后释放 BD。ISO 端点正在填充一个 FIFO 时需设置此位。微处理器不会被告知令牌已被处理, 数据被简单地传送或接收到 FIFO。当 KEEP = 1 时, NINC 位通常也被置'1', 以防止地址递增。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 3 会被 USB_FS 写回到 BD 中。
NINC	NINC 位禁用 DMA 地址递增。这将强制 DMA 从相同的地址读取或写入数据。当数据需要从单一位置 (如 FIFO) 读取或写入单个地址时, 这对端点很有用。通常情况下, 该位与 KEEP 位一同设置, 对于连接到 FIFO 的 ISO 端点。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 2 会被 USB_FS 写回 BD。
DTS	设置该位将使 USB_FS 执行数据切换同步机制。当该位为 0 时, 不执行数据切换同步。如果 KEEP = 1, 则该位不会被 USB_FS 改变, 否则当前令牌 PID 的位 1 会被 USB_FS 写回 BD。
BDT_STALL	如果 SIE 使用此地址的 BDT 接收令牌, 则设置此位将导致 USB_FS 发出 STALL 握手。当设置 BDT_STALL 位时, SIE 不改变 BDT (OWN 保持, BDT 的其余部分保持不变)。
TOK_PID[3: 0]	当传输完成时, 当前令牌 PID 由 USB_FS 写回 BD。写回的值是来自 USB 规范的令牌 PID。具体请参考 USB 规范。
ADDR [31: 0]	地址位表示系统存储器中的 32 位缓冲区地址。由用户设置。

34.3.4.4 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。

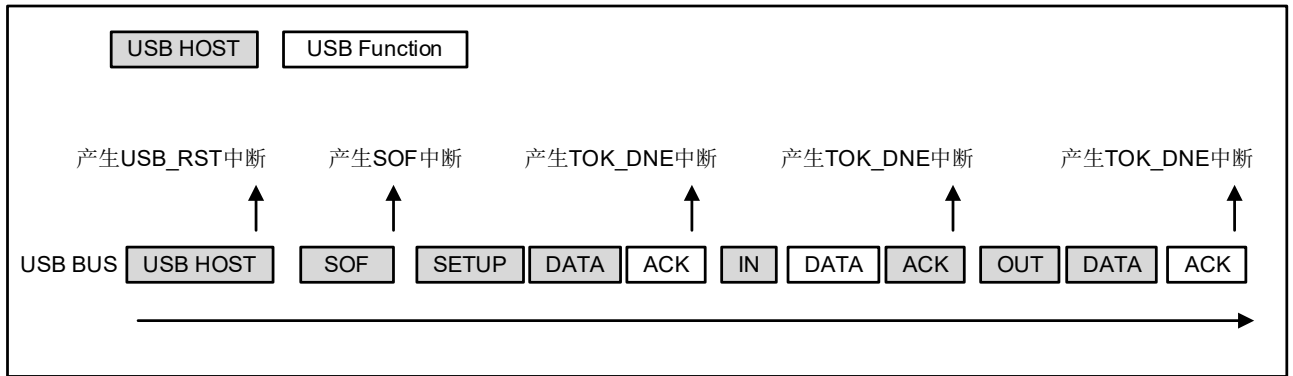


图 34-6 USB 令牌传输示意图

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 中断也将会被触发。回写到 BDT 的数据包长度将是 MaxPacket size 值，用于表示实际写入内存的数据长度。软件可以取消传输，禁止断点等操作。

34.3.5 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。

34.3.6 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。

34.3.6.1 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

34.3.6.2 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 Divce 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

34.3.6.3 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。

34.3.7 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）。

34.3.7.1 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。

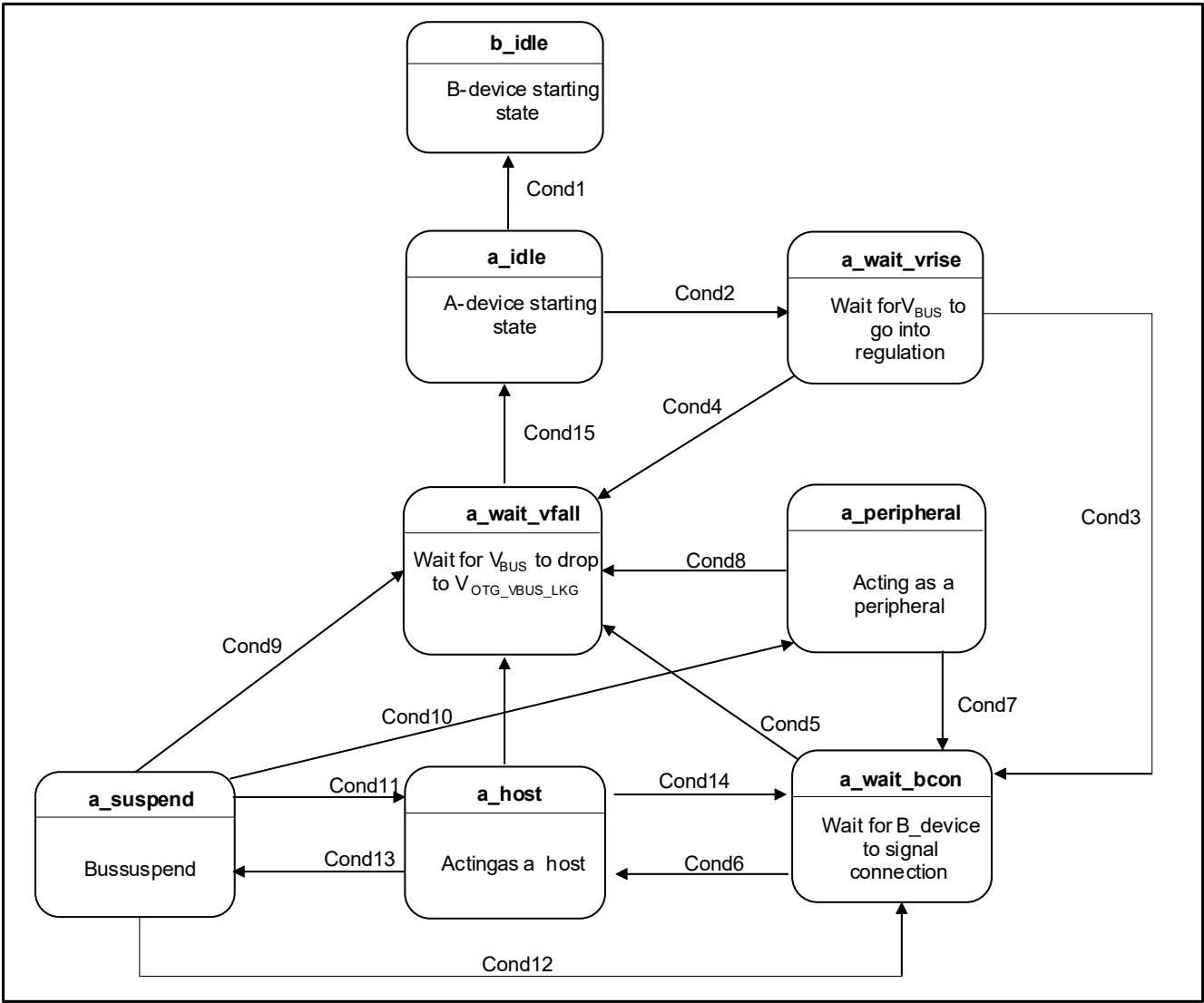


图 34-7 OTG A-device 操作方式

表 34-6 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。

34.3.7.2 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 是会话开始时的外国设备。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。

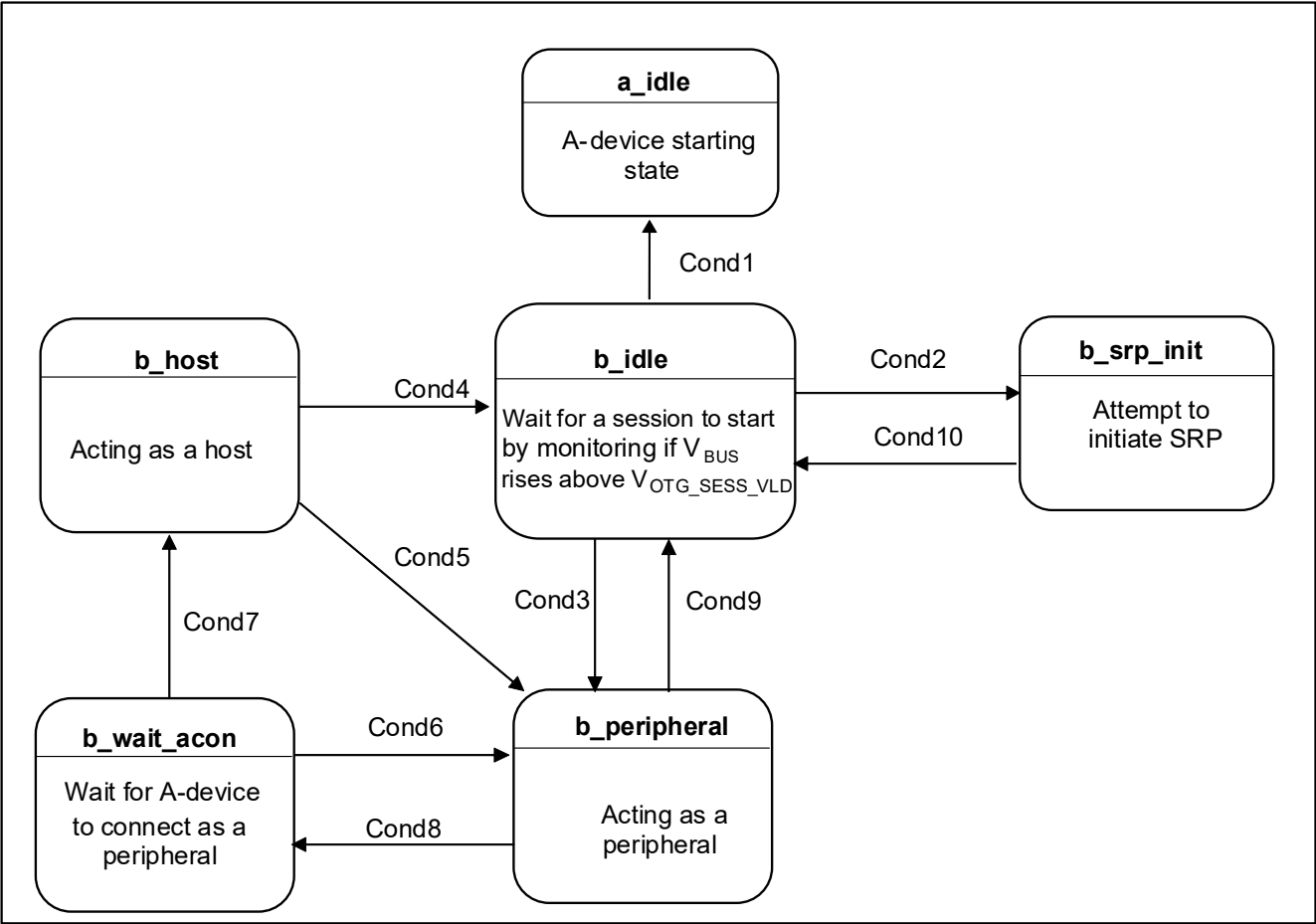


图 34-8 OTG B-device 操作方式

表 34-7 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。

34.3.8 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-device 会话结束中断
- ◆ A-device VBUS 变化中断

34.4 寄存器

34.4.1 寄存器总览

表 34-8 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_SOFTHLD	SOF Threshold Register	0x0000

Offset	Acronym	Register Name	Reset
0xB0	USB_FS_BDTPAGE2	BDT Page Register 2	0x0000
0xB4	USB_FS_BDTPAGE3	BDT Page Register 3	0x0000
0xC0	USB_FS_EPCTL0	Endpoint Control Register 0	0x0000
0xC4	USB_FS_EPCTL1	Endpoint Control Register 1	0x0000
0xC8	USB_FS_EPCTL2	Endpoint Control Register 2	0x0000
0xCC	USB_FS_EPCTL3	Endpoint Control Register	0x0000
0xD0	USB_FS_EPCTL4	Endpoint Control Register 4	0x0000
0xD4	USB_FS_EPCTL5	Endpoint Control Register 5	0x0000
0xD8	USB_FS_EPCTL6	Endpoint Control Register 6	0x0000
0xDC	USB_FS_EPCTL7	Endpoint Control Register 7	0x0000
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
0x100	USB_FS_USBCTRL	USB Control register	0x0000

34.4.2 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 SE0 和 JSTATE 位）稳定，在 1 毫秒内无变化，并且线路状态的与线路状态最后一次稳定时的值不同时，该中断被设置。当 SE0 和 J，SE0 和 K 以及 J 和 K 之间产生转换，它将被置位。当 SE0 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。

Bit	Field	Description
4	Reserved	保留，必须保持复位值。
2	SESS_VLD_CHG	当检测到 VBUS 发生变化时，该位置位，表示会话有效或会话不再有效。
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’。

34.4.3 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.	A_VBUS_VLD_EN
								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。

34.4.4 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	Reserved.	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 SE0）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 SE0 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 SE0 位的值在 1 毫秒内不变，可以被认为去抖动的。
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。

34.4.5 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- 数据线上的下拉电阻。
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。

Bit	Field	Description
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。

34.4.6 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。

34.4.7 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。

34.4.8 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。

Bit	Field	Description
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。
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。

34.4.9 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。

Bit	Field	Description
0	PID_ERR	设置该位为 1 使能 PID_ERR 中断。

34.4.10 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。

34.4.11 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	TxdSuspendTokenBusy	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。
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_MODE_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。

34.4.12 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。

34.4.13 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。

34.4.14 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。

34.4.15 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。

34.4.16 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。

34.4.17 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。

34.4.18 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。

34.4.19 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。

34.4.20 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
Res.								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																														

Bit	Field	Description
0	EP_HSHK	0: 该端点不返回握手信号 1: 该端点返回握手信号

34.4.21 USB_FS_USBCTRL USB 控制寄存器

偏移地址: 0x100

复位值: 0x0000

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

Bit	Field	Description
15: 8	Reserved	保留, 必须保持复位值。
7	SUSPE	片内 USB PHY 挂起状态使能位, 用于 USB 控制器处于挂起状态时, 设置此位可以使片内 PHY 处于挂起状态, 进一步节省功耗。清除该位使片内 USB PHY 处于正常工作状态。SUSPE=1 是, 允许 Resume 或 Remote Wakeup 操作唤醒系统 STOP 模式。 0: 片内 USB PHY 处于正常工作状态 1: 片内 USB PHY 处于挂起状态, 允许 Resume 或 Remote Wakeup 操作唤醒系统 STOP 模式
6: 0	Reserved	保留, 必须保持复位值。

35 ENET 以太网 MAC 控制器

35.1 简介

以太网是一种计算机局域网技术。IEEE 组织的 IEEE802.3 标准制定了以太网的技术标准，它规定了包括物理层的连线、电气信号和介质访问层协议的相关内容。以太网 MAC 控制器（ENET）用于在以太网网络中发送和接收数据，它有多种应用领域，如交换机、网络接口卡等。

本产品中以太网 MAC 控制器与外部物理层（PHY）之间，可以通过媒体独立接口（MII）工业标准接口相连。

35.2 主要特征

NENT 章节中提到的主要缩略词的含义解释如下：

- SMI: Station Management Interface 站点管理接口
- MII: Media Independent Interface 媒体独立接口
- PR: Preamble 前导码
- SFD: Start of Frame Data 帧开始数据
- DA: Destination Address 目标地址
- SA: Source Address 源地址
- LEN: Length 长度；指以太网帧的 Length/Type 部分
- CRC: Cyclic Redundancy Check 循环冗余校验码
- IFG: Interframe Gap 帧间距；本章节还指相关的寄存器位
- FCS: Frame Check Sequence 帧检测序列；指以太网帧的 CRC 部分
- SOF: Start of Frame 帧开始；一般指以太网帧的 PR+SFD 部分
- EOF: End of Frame 帧结尾；一般指以太网帧的 FCS 部分
- VLAN: Virtual Local Area Network 虚拟局域网
- VLAN TAG: VLAN Tag VLAN 标签
- COE: Checksum Offload Engine 校验和减荷引擎
- PMT: Power Management 电源管理
- MMC: Management Counter 管理计数器
- PTP: Precision Time Protocol 精密时间协议
- PPS: Pulse Per Second 秒脉冲

其它未列出的缩略词含义，请参考以太网标准协议或手册其它章节。

35.2.1 MAC 子控制器特征

- 支持外部 PHY 接口实现 10/100Mbps 数据传输速率
- 为应用程序提供单独的发送、接收和控制接口
- 使用 SMI 接口配置和管理最多 32 个 PHY 设备
- 支持检测远程唤醒帧和 AMD Magic Packet™ 帧
- 支持内部回送的 Loopback 模式
- 支持半双工操作的 CSMA/DA 协议
- 支持半双工操作的背压流量控制
- 支持适用于全双工操作的 IEEE802.3x 流量控制
- 全双工操作时可以将接收的暂停控制帧转发到应用程序
- 全双工操作中如果流量控制输入信号消失，将自动发送零时间片暂停帧
- 接收以太网帧时，自动去除数据帧（长度字段小于 0x600）的 PAD 和 FCS 字段
- 接收以太网帧时，自动去除类型帧的 FCS 字段
- 接收以太网帧时，自动进行接收帧的 CRC 计算
- 发送以太网帧时，对 SA 进行插入、替换
- 发送以太网帧时，对 CRC 进行插入、替换
- 发送以太网帧时，对 VLAN 帧标识符进行插入、替换、删除
- 发送以太网帧时，对小于 60 字节（DA+SA+LEN+DATA）的帧自动生成 PAD 填充
- 发送以太网帧时，处理冲突帧的自动重新发送
- 发送以太网帧时，支持可编程帧间隔（40-96 位时间，以 8 为步长）
- 支持 6 个 48 位完全（DA）地址过滤器，可对每个字节进行掩码操作
- 支持 5 个 48 位完全（SA）地址过滤器，可对每个字节进行掩码操作
- 支持 64 位 Hash 过滤器，适用于单播和多播目标地址过滤
- 可传送所有多播地址帧
- 可传送所有帧，无需为网络监视进行过滤
- 传送所有传入数据包时（每次过滤时）均附有一份状态报告
- 支持一组 VLAN 标识符过滤
- 支持 12/16 位 VLAN 标识符选择
- 支持 VLAN 标识符 Hash 滤波方式
- 支持四组远程唤醒帧过滤器，控制四组掩码操作
- 在接收帧时，可选择丢弃无 TCP/UDP 字段的帧
- 对接收到的 IPv4 报文的 Header Checksum 字段进行校验

- 对接收到的 TCP 报文、UDP 报文、ICMP 报文的 Checksum 字段进行校验
- 对发送的 IPv4 报文进行 Checksum 计算，将计算结果插入到 Header Checksum 字段
- 对发送的 TCP 报文、UDP 报文、ICMP 报文进行 Checksum 计算，将计算结果插入到 Checksum 字段
- 支持 MMC 计数器进行网络统计
- 具有一个可编程阈值的 2KB 专用发送 Memory (TxFIFO) 和一个可编程阈值的 2KB 专用接收 Memory (RxFIFO)
- RxFIFO 和 TxFIFO 均支持存储转发模式
- 软件控制刷新 TxFIFO
- RxFIFO 在存储转发模式下，可选择过滤所有的错误帧，不将这些错误帧转发给应用
- RxFIFO 可根据填充（阈值可配置）级别自动生成要发送给 MAC 子控制器的暂停控制帧或背压信号
- RxFIFO 中丢失或损坏的帧可进行数据统计
- RxFIFO 进行多帧存储时，通过在 EOF 传输后向 RxFIFO 插入接收状态矢量，从而使得 RxFIFO 无需存储这些帧的接收状态

35.2.2 DMA 子控制器特征

- 支持软件设定 AHB Master 接口的 AHB 突发类型（固定或不确定突发）
- AHB Slave 接口中支持所有 AHB 突发类型
- AHB Master 接口可选择地址对齐突发传输
- 支持 RxDMA 和 TxDMA 引擎单独编程突发长度，以充分利用总线
- 支持对数据缓冲区进行字节对齐寻址
- 支持环式和链式两种描述符链接方式
- 每个描述符可传输高达 8KB 的数据
- TxDMA 和 RxDMA 均支持存储转发模式
- 报告正常工作和传输错误时的综合状态
- 可编程中断选项，根据不同的应用场景选择不同的中断
- 按帧控制发送/接收完成中断
- 接收引擎和发送引擎间支持循环调度仲裁或固定优先级仲裁

35.2.3 PTP 子控制器特征

- 支持接收帧和发送帧进行时间戳快照，插入时间戳
- 支持对系统时间进行粗校准和细校准

- 支持 2 组目标时间设定，系统时间大于目标时间时会触发中断
- 支持 PPS 输出，可以作为事件输出来控制其它模块的动作

35.3 功能描述

35.3.1 功能框图

下图 35-1 所示，是 ENET 的整体功能框图。

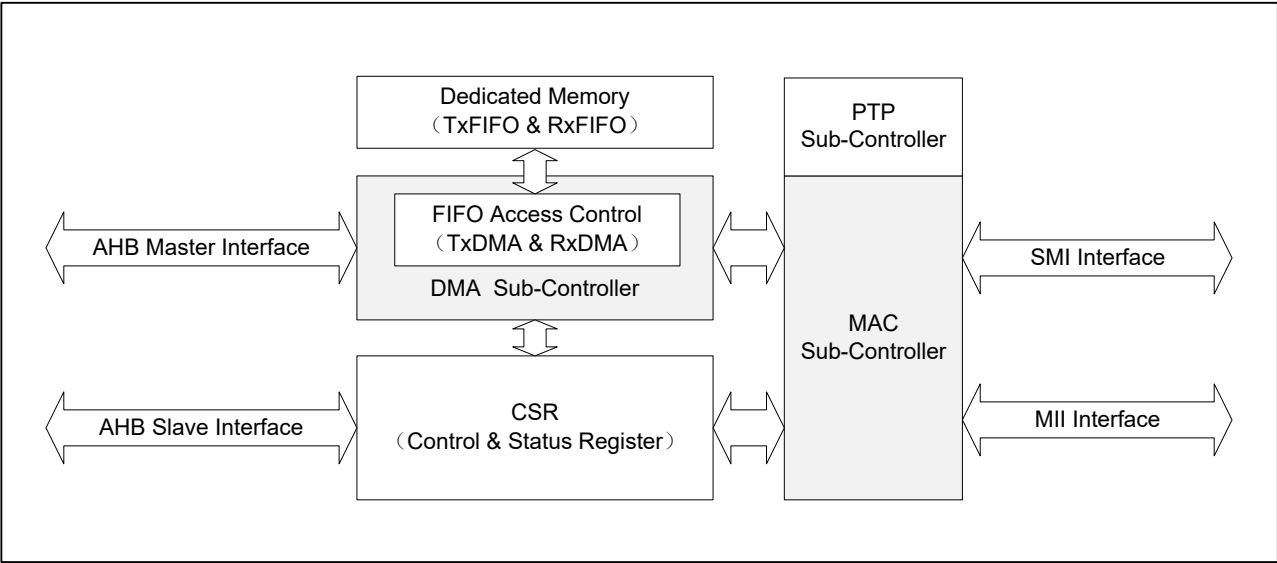


图 35-1 ENET 功能框图

ENET 包括 AHB Master 接口、AHB Slave 接口、DMA 子控制器、MAC 子控制器、PTP 子控制器、接口控制逻辑等。DMA 子控制器通过 AHB 接口将 MAC 子控制器和系统存储器相连。PTP 子控制器用于在接收和发送帧过程中，生成和管理时间戳。AHB Master 接口用于数据传输，AHB Slave 接口用于 ENET 的基本寄存器访问配置等。

在发送数据时，将数据从系统存储器由 DMA 子控制器内的 TxDMA 引擎送至专用 Memory (TxFIFO)，经 MAC 子控制器后通过 MII 接口送至外部 PHY。

在接收数据时，经过 MII 接口接收进来的数据，经 MAC 子控制器后送至专用 Memory (RxTxFIFO)，DMA 子控制器的 RxDMA 引擎再将接收数据传送到系统存储器。

35.3.2 信号描述

下表 35-1 显示了 MAC 信号和 SMI/MII 接口的映射。每个功能接口信号映射到不同的 I/O 引脚，通过复用功能模式进行配置。

表 35-1 ENET 功能引脚说明

SMI/MII 接口	类型	描述
ENET_MDC	输出	周期性时钟，提供最大频率 2.5MHz 传输数据时的参考时序。MDC 的最短高电平和最短低电平必须均为 160ns。MDC 的最小周期必须为 400 ns。在空闲状态下，SMI 将 ENET_MDC 驱动为低电平
ENET_MDIO	输入输出	数据输入/输出比特流，用于通过 ENET_MDC 时钟信号与 PHY 设备进行数据信息传输
ENET_MII_RX_CLK	输入	RX 数据传输时的参考时钟。MII 接口时，速率 10Mbps 时为 2.5MHz、速率为 100Mbps 时为 25MHz
ENET_MII_RX_DV	输入	接收数据有效，表示 PHY 当前正对 MII 接口发送数据。为正确地接收帧，该信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 字段出现的时间
ENET_MII_RX_ER	输入	接收错误，该信号必须保持一个或多个发送时钟周期，从而向 MAC 子层指示在帧的某处检测到错误
ENET_MII_RXD0	输入	接收数据，由外部 PHY 同步驱动 在 ENET_MII_RX_DV 信号有效时，才为有效信号
ENET_MII_RXD1	输入	
ENET_MII_RXD2	输入	
ENET_MII_RXD3	输入	
ENET_MII_TX_CLK	输入	TX 数据传输时的参考时钟。速率为 10Mbps 时为 2.5MHz；速率为 100Mbps 时为 25MHz
ENET_MII_TX_EN	输出	发送使能，表示 MAC 当前正通过 MII 接口发送数据
ENET_MII_TXD0	输出	发送数据，由 MAC 子层同步驱动 在 ENET_MII_TX_EN 信号有效时，才为有效信号；在 ENET_MII_TX_EN 信号无效时，发送数据不会对 PHY 产生任何影响
ENET_MII_TXD1	输出	
ENET_MII_TXD2	输出	
ENET_MII_TXD3	输出	
ENET_MII_CRS	输入	载波侦听，当发送或接收介质处于非空闲状态时，由 PHY 使能该信号；发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 ENET_MII_CRS 信号在冲突条件下保持有效状态。该信号无需与发送和接收时钟保持同步。在全双工模式下，该信号没意义
ENET_MII_COL	输入	冲突检测，检测到介质上存在冲突后，PHY 必须立即使能冲突检测信号，并且只要存在冲突条件，冲突检测信号必须保持有效状态。该信号无需与发送和接收时钟保持同步。在全双工模式下，该信号没意义
ENET_PPS_OUT	输出	PTP 功能的 PPS 输出

35.3.3 连接关系

MII 接口和 PHY 芯片的连接关系如图 35-2 所示。

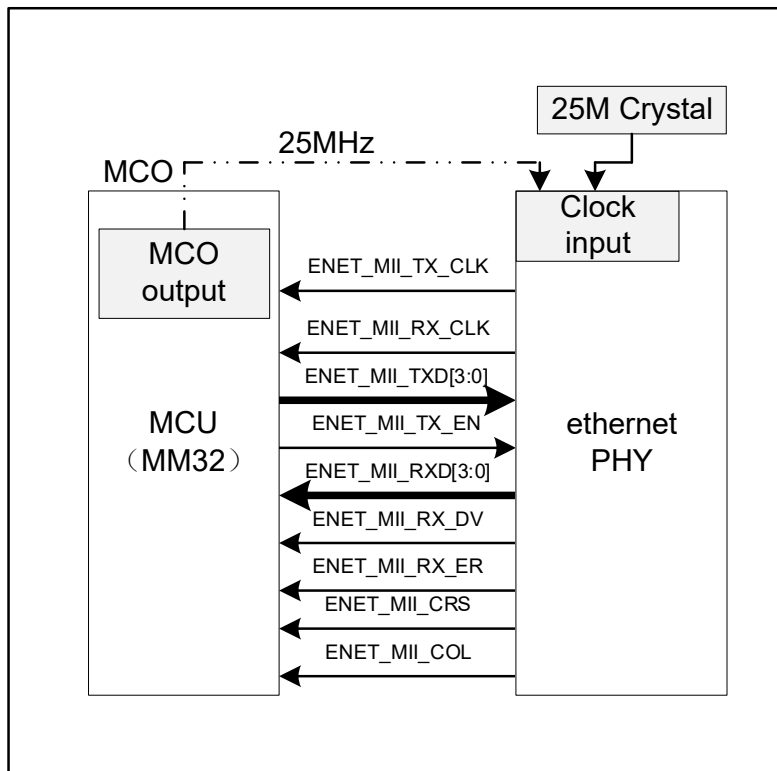


图 35-2 MII 接口和 Device PHY 连接图

要生成 ENET_MII_TX_CLK 和 ENET_MII_RX_CLK 时钟信号，必须向外部 PHY 提供 25MHz 时钟，如图所示。除了使用外部 25 MHz 石英晶体提供该时钟，还可以通过本产品的 MCO 引脚输出该时钟。

35.3.4 802.3 帧格式

以太网 MAC 层的数据通信可使用两种帧格式：

- 基本 MAC 帧格式
- 带 VLAN 标识符 MAC 帧格式（扩展了基本 MAC 帧格式）

帧格式具体分段内容如下图 35-3 所示。

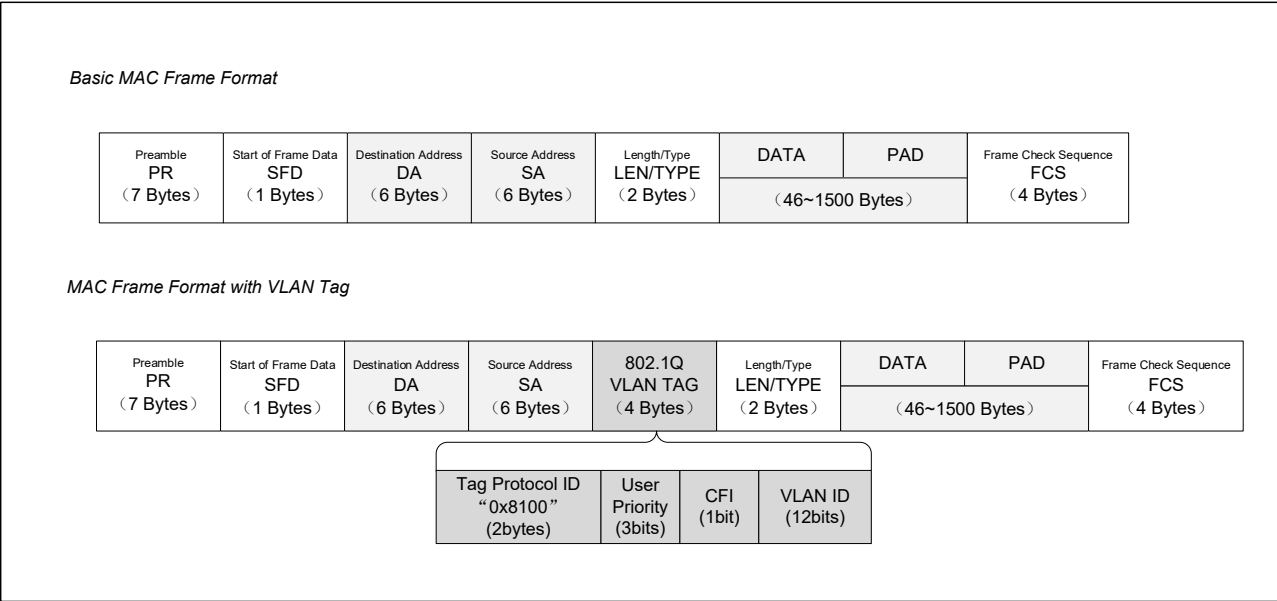


图 35-3 以太网帧格式

除了帧校验序列 FCS，以太网 MAC 控制器发送每个字节时都按照低位先出的次序进行传输。

CRC 计算包括帧数据的所有字节除去前导码和帧首界定码域。以太网帧 CRC32 生成多项式为 0x04C11DB7，且此多项式用于以太网模块中所有的 32 位 CRC 计算，如下式所示：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

通过以下条件之一定义一个帧为无效 MAC 帧：

- 帧长度与长度/类型字段指定的预期值不一致。如果长度/类型字段包含类型值，则认为帧长度与此字段一致（没有无效帧）
- 帧长度不是字节的整数倍
- 根据传入帧计算出的 CRC 值与包含的 FCS 不匹配

35.3.5 MAC 子控制器基本特征

MAC 子控制器执行以下与数据链路控制步骤相关的功能：

- 数据封装（发送和接收）
- ◆ 组帧（帧边界定界、帧同步）
- ◆ 寻址（源地址和目标地址）
- ◆ 错误检测
- 介质访问管理
- ◆ 介质分配（冲突避免）
- ◆ 竞争解决（冲突处理）
- MAC 子控制器主要有两个工作模式：
- 半双工模式：站点使用 CSMA/CD 算法争用物理介质



- 全双工模式：满足以下条件时，无需解决竞争问题（CSMA/CD 算法不是必需的）便可同时发送和接收数据：
 - ◆ 物理介质能够支持同步发送和接收
 - ◆ 正好有 2 个站点与 LAN 相连
 - ◆ 两个站均配置为全双工工作模式

35.3.6 MAC 子控制器接收功能

MAC 子控制器接收的帧将送入 RxFIFO，在接收达到一定数量条件时，就会指示 RxDMA 进行数据传送，RxDMA 可向 AHB 接口发起预配置的突发传输。

以太网 MAC 控制器搭载 2Kbyte 的 RxFIFO。RxDMA 将 RxFIFO 中的数据输送到系统存储器有两种操作模式：阈值模式和存储转发模式。

● 阈值模式

在阈值模式下，当 RxFIFO 接收到的字节数大于配置的接收阈值时（使用 DMA 工作模式寄存器的 RTC 位（ENET_DMAMDR.RTC）配置）或完整的数据包时，数据将发起传送，其可用性将通知给 RxDMA。RxDMA 向 AHB 接口发起传输后，数据传输将从 RxFIFO 持续进行，直到传输完成整个数据包。完成 EOF 帧的传输后，接收状态字将返回并发送到 RxDMA。在该模式下，某些错误帧不会被丢弃，因为在帧结束时接收到错误状态，而此时已从 RxFIFO 将数据传送到系统存储器。

● 存储转发模式

在存储转发模式下，当帧完全写入 RxFIFO 后才可读出帧。在此模式下，将丢弃所有错误帧（如果配置为执行此操作），这样只会读出有效帧并将其转发到应用。接收数据后状态立即可用，因此只要 RxFIFO 未满，就可以向 RxFIFO 中存储帧。

当 MAC 在 MII 上检测到 SFD 时，将启动接收操作。MAC 子控制器将去除前导码和 SFD，然后再继续处理帧。检查地址字段以进行过滤，FCS 字段用于验证帧的 CRC。如果帧未通过地址过滤器，则丢弃该帧。

35.3.6.1 接收协议

MAC 接收数据时，首先去除接收的帧的前导码和 SFD。检测到 SFD 后，开始向 RxFIFO 传送以太网帧数据，从 SFD 后面的第一个字节（目标地址）开始发送。

如果接收的帧长度/类型字段小于 0x600 并且为 MAC 编程了自动去除 CRC/PAD 选项，则 MAC 将向 RxFIFO 发送帧数据（数据量不超过长度/类型字段中指定的数量），然后开始丢弃字节（包括 FCS 字段）。如果长度/类型字段大于或等于 0x600，则不管编程的自动 CRC 去除选项的值如何，MAC 都会向 RxFIFO 发送所有接收到的以太网帧数据。

默认情况下，使能 MAC 看门狗定时器，即超过 2K 个字节（DA+SA+LEN+DATA+PAD+FCS）的帧会被切断。可通过对 MAC 控制寄存器的 WTD 位（ENET_MACCR.WTD）编程来禁止此功能。但是，即使禁止看门狗定时器，仍将切断大于 16KB 的帧并给出看门狗超时状态。

35.3.6.2 错误处理

如果在从 MAC 接收 EOF 数据之前 RxFIFO 已满，则将声明上溢并丢弃整个帧，同时 DMA 帧丢失统计寄存器中的 OVFC 位（ENET_DMAFLCR.OVFC）所指示的计数器将递增。由于上溢，状态将指示这是一个部分帧。如果使能 DMA 工作模式寄存器的 FEF 位或 FUF 位（ENET_DMAMDR.FEF 或 ENET_DMAMDR.FUF），RxFIFO 可过滤错误帧和过小帧。

在阈值模式下，从 RxFIFO 读取帧 SOF 时，可获取当前帧的基本信息，通过与接收描述符的基本设定比对，则可判定该帧是否为错误帧，并丢弃整个错误帧；在存储转发模式下，则可在数据传送之前过滤并丢弃所有错误帧。

35.3.6.3 CRC&PAD 处理

MAC 子控制器将计算接收的帧（包括目标地址字段到 FCS 字段）的 32 位 CRC，并检查接收帧中的任何 CRC 错误。不管是否自动去除 CRC/PAD（通过设定 MAC 控制寄存器的 APCS 位（ENET_MACCR.APCS）来控制），MAC 都将接收整个帧来计算所接收帧的 CRC 校验，并将校验的结果通过接收描述符 RDES0 的 CRE 位反馈给应用。

35.3.6.4 状态字

以太网帧接收结束时，MAC 输出接收状态通过 RxDMA 送到接收描述符 RDES0 中以供应用参考查询。

35.3.6.5 暂停帧

在帧发送过程中，MAC 将检测接收暂停帧并暂停帧发送，暂停时间为接收的暂停帧内指定的延迟（仅限全双工模式）。可通过 MAC 流控寄存器中的 FRE 位（ENET_MACFCR.FRE）使能或禁止暂停帧检测功能。使能接收流量控制后，将开始监视接收帧的目标地址是否与控制帧的多播地址（0x0180C2000001）匹配。如果检测到匹配（接收的帧的目标地址与保留的控制帧的目标地址匹配），MAC 将根据 MAC 地址过滤寄存器中的 PCF 位（ENET_MACAFR.PCF）来决定是否将接收的控制帧传输到应用。

MAC 还将对接收到的控制帧的类型、操作码和暂停定时器字段进行解码。如果状态的字节计数指示 64 个字节，并且不存在任何 CRC 错误，则 MAC 发送器将暂停任何数据帧的发送，暂停时间为解码的暂停时间值乘以时隙（对于 10/100Mbps 模式，均为 64 字节时间）。同时，如果检测到另一个零暂停时间值的暂停帧，MAC 将复位暂停时间并管理新的暂停请求。如果接收到的控制帧与类型字段（0x8808）、操作码（0x00001）以及字节长度（64 字节）均不匹配，或存在 CRC 错误，则 MAC 不会生成暂停。

对于具有多播目标地址的暂停帧，MAC 将根据地址匹配来过滤帧；对于具有单播目标地址的暂停帧，MAC 将根据 DA 字段是否与 MAC 地址寄存器 0（ENET_MACAH/LR0）的过滤地址匹配以及 MAC 流控寄存器中的 UPF 位（ENET_MACFCR.UPF）是否置 1（检测具有单播目标地址的暂停帧）来进行过滤。

35.3.6.6 接收 COE

在网络互联过程中，基本的通信协议（例如 TCP 和 UDP）都实施校验和，并配备校验和字段，这有助于确定通过网络发送的数据的完整性。由于以太网最广泛的用途是通过 IP 数据报文封装 TCP 和 UDP，因此以太网 MAC 子控制器具有校验和减荷功能（Checksum Offload Engine，简称 COE 引擎），该功能包含接收 COE 引擎和发送 COE 引擎，接收 COE 引擎支持对接收的以太网帧中的 IPv4 和 IPv6 帧进行检测和处理，以确保数据完整性。

通过将 MAC 控制寄存器中的 IPCO 位（ENET_MACCR.IPCO）置 1 来使能接收 COE 引擎。MAC 通过检查所接收的以太网帧的类型字段中是否存在值 0x0800 或 0x86DD，来识别 IPv4 或 IPv6 帧。此识别方法也适用于带 VLAN 标识符的帧。

接收 COE 引擎可计算 IPv4 报文头校验和，并检查其是否与接收的 IPv4 报文头校验和匹配。如果指示的有效负载类型（以太网类型字段）与 IP 报文头版本之间有任何不匹配，或接收的帧的字节数小于 IPv4 报文头的长度字段中指示的数量（IPv4 或 IPv6 报文头中的可用字节数少于 20），IP 报文头错误位将置 1。

接收 COE 引擎还可以识别接收的 IP 数据报文（IPv4 或 IPv6）中的 TCP、UDP 或 ICMP 有效负载，并正确计算此类有效负载的校验和字段，包括用于校验和计算的 TCP/UDP/ICMPv6 伪报文头字节，并检查接收的校验和字段是否与计算的匹配。

COE 引擎将绕过分段 IP 数据报文的有效负载、带安全功能的 IP 数据报文、IPv6 路由报文头以及除 TCP、UDP 或 ICMP 以外的有效负载。

35.3.6.7 接收时序

下图 35-4、图 35-5、图 35-6 所示，是基本的接收时序图。

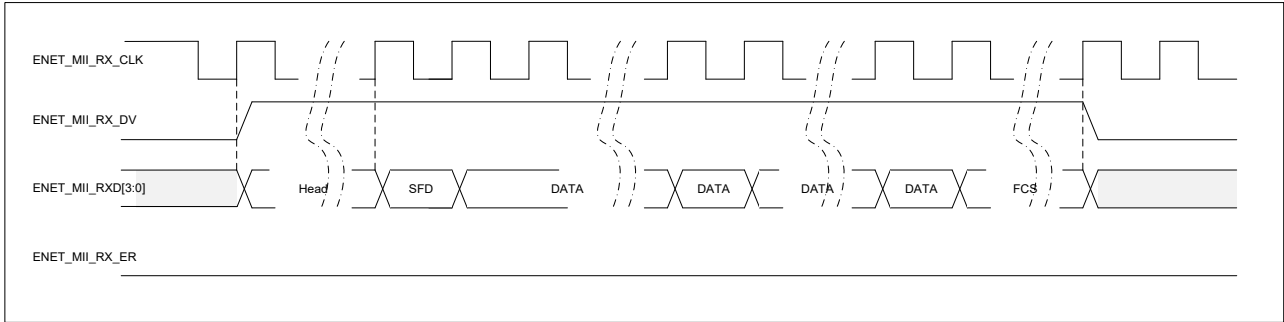


图 35-4 无错误时接收数据时序图

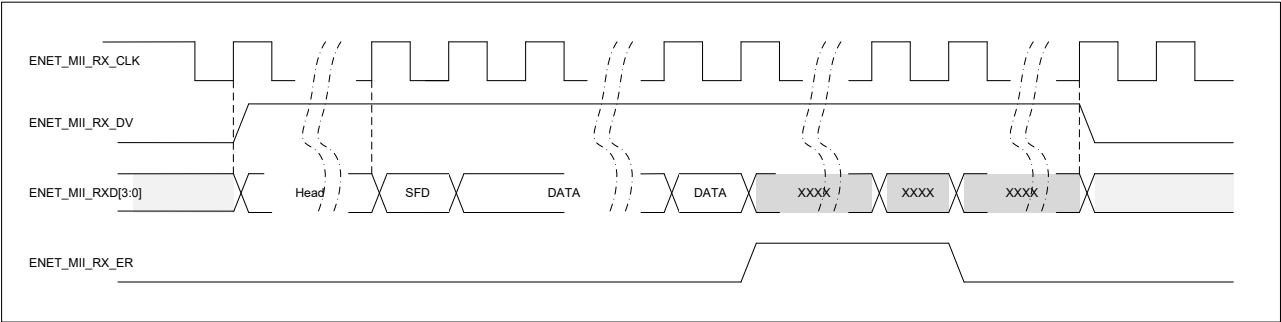


图 35-5 有错误时接收数据时序图

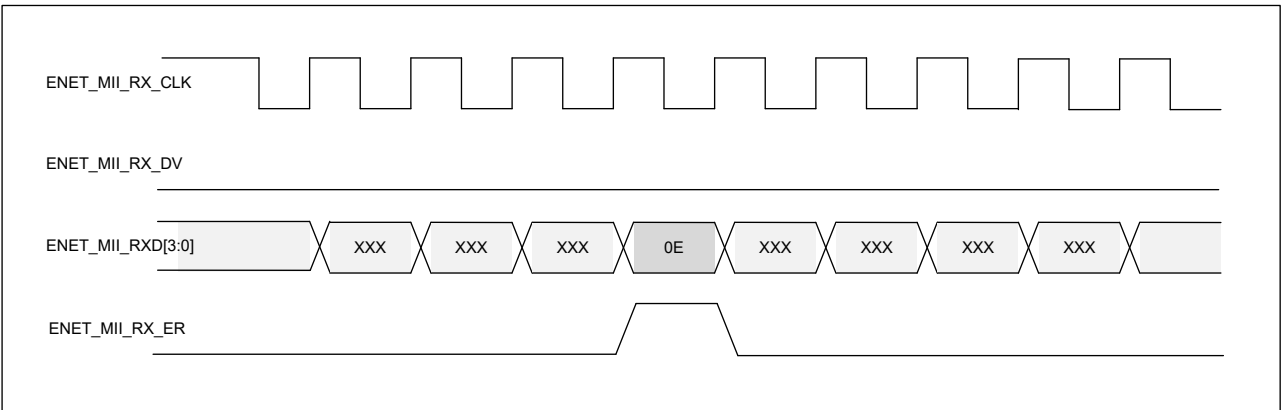


图 35-6 假载波指示接收时序图

35.3.7 MAC 子控制器发送功能

TxDMA 控制发送路径的所有事务。从系统存储器读取的以太网帧由 TxDMA 送入 TxFIFO，然后传输至 MAC 子控制器。帧传输结束时，从 MAC 子控制器获取发送状态并传回 TxDMA。

以太网 MAC 控制器搭载 2KB 的 TxFIFO。TxFIFO 填充级别将指示给 DMA 子控制器，以便 TxDMA 可通过 AHB 接口在所需的系统存储器启动数据传送。来自 AHB Master 接口的数据将送入 TxFIFO。

检测到 SOF 时，MAC 接收数据并开始向 MII 发送。在应用程序启动发送后，向 MII 发送帧数据所需的时间是可变的，具体取决于 IFG 延迟、发送前导码和 SFD 的时间以及半双工模式的任意回退延迟等延迟因素。EOF 传输到 MAC 子控制器后，MAC 子控制器将完成正常的发送，然后将发送的状态返回给 TxDMA。如果在发送过程中发生常规冲突（在半双工模式下），MAC 子控制器将使发送状态有效，然后接受并丢弃所有后续数据，直至收到下一个 SOF。检测到来自 MAC 的重试请求（在状态中）时，应从 SOF 重新发送同一帧。

如果发送期间未连续提供数据，MAC 将发出下溢状态。在帧的正常传输期间，如果 MAC 在未获得前一帧的 EOF 的情况下接收到 SOF，则将忽略该 SOF 并将新的帧视为前一帧的延续。

向 MAC 子控制器输送数据有两种操作模式：阈值模式和存储转发模式。

- 阈值模式

在阈值模式下，只要 TxFIFO 中的字节数超过配置的阈值（或在超过阈值前写入帧结束），数据就准

备好并转发到 MAC 子控制器。该阈值可使用 DMA 工作模式寄存器的 TTC 位(ENET_DMAMDR.TTC)配置。

- 存储转发模式

在存储转发模式下,仅当在 TxFIFO 中存储完整的帧后,才会向 MAC 内核输送帧。如果 TxFIFO 的大小小于要发送的以太网帧,则在 TxFIFO 接近填满时向 MAC 内核输送帧。

应用可通过将 DMA 工作模式寄存器的 FTF 位(ENET_DMAMDR.FTF)置 1 来清空 TxFIFO 的所有内容。此位自动硬件清零,并将 FIFO 指针初始化为默认状态。如果向 MAC 内核传输帧时将该位置 1,则传输将停止,此时 TxFIFO 被视为空,MAC 发送器将出现下溢事件并且相应的状态字将转发给 TxDMA。

35.3.7.1 发送协议

MAC 子控制器控制以太网帧的发送操作。它执行下列功能以满足 IEEE802.3/802.3z 规范。包括:

- 生成前导码和 SFD
- 在半双工模式下生成 Jam 阻塞信号
- 控制 Jabber 超时
- 控制半双工模式下的流量(背压)
- 生成发送帧状态

当请求发送新的帧时,MAC 将发送前导码和 SFD,紧接着发送数据。前导码定义为“10101010”样式的 7 个字节,SFD 定义为“10101011”样式的 1 个字节。

冲突窗口定义为 1 个时隙(对于 10/100 Mbps 以太网,为 512 个位时间)。阻塞信号生成仅适用于半双工模式,不适用于全双工模式。在 MII 模式下,如果在开始传输帧到 CRC 字段结束之间的任何时间发生冲突,MAC 将在 MII 上发送 0x55555555 的 32 位 Jam 阻塞信号,通知所有其它站已发生冲突。如果在前导码发送阶段发生冲突,MAC 将完成前导码和 SFD 的发送,然后发送 Jam 阻塞信号。

MAC 子控制器内使用一个 Jabber 定时器,用于在传输的字节超过 2048 字节时切断以太网帧的发送。

在半双工模式下,MAC 使用延迟机制进行流量控制(背压)。当上层应用请求停止接收帧时,如果已使能发送流量控制,则只要 MAC 检测到接收帧,就会发送一个 32 字节的 Jam 信号。这会导致冲突并使远程站回退。应用程序通过将 MAC 流控寄存器中的 FCBPBA 位(ENET_MACFCR.FCBPBA)置 1 来请求流量控制。如果应用请求发送帧,则即使激活背压功能,也将按调度计划发送。如果背压功能长时间保持激活(发生的连续冲突事件超过 16 个),则远程站将由于冲突过多而中止发送。

35.3.7.2 调度算法

MAC 子控制器负责调度 MII 上的帧发送。通过 MAC 控制寄存器的 IFG 位(ENET_MACCCR.IFG)设定或截断二进制指数回退算法来保持两个发送帧之间的帧间隔。在满足 IFG 和回退条件后,MAC 使能

发送。

IFG 设定可确保两个发送帧之间的空闲期段，即配置的帧间隔。如果要发送的帧在配置的 IFG 时间之前到达，则 MII 会等待来自 MAC 的使能信号，然后再开始发送。只要 MII 的载波信号进入无效状态，MAC 就会启动 IFG 计数器。在 IFG 计数值与设定值相等时，MAC 将以全双工模式使能发送。

在半双工模式下，当 IFG 配置为 96 个位时间时，MAC 将遵循 IEEE802.3 规范指定的顺从规则。如果在 IFG 间隔的前三分之二时间内（对于所有 IFG 值都为 64 位时间）检测到载波，MAC 将复位其 IFG 计数器。如果在 IFG 间隔的后三分之一时间内检测到载波，MAC 将继续执行 IFG 计数并在 IFG 间隔结束后使能发送器。

在半双工模式下，实施截断二进制指数回退算法，具体通过 MAC 控制寄存器的 BL 位（ENET_MACCR.BL）设定控制。

35.3.7.3 冲突重发

在半双工模式下，向 MAC 子控制器传输帧时，可能在 MAC 接口上发生冲突事件。MAC 子控制器会在接收到帧结束状态之前就指示重试，然后将使能重新发送并再次将帧从 TxFIFO 中送出。

当超过 96 个字节送到 MAC 子控制器后，TxFIFO 控制器将释放该空间，使 DMA 可送入更多数据。这意味着超过阈值后或 MAC 子控制器指示延迟冲突事件时，无法重新发送。

35.3.7.4 TxFIFO 刷新

MAC 子控制器可通过 DMA 工作模式寄存器中的 FTF 位（ENET_DMAMDR.FTF）来清空 TxFIFO。清空操作是立即操作，即使 TxFIFO 正在向 MAC 内核传输帧，TxFIFO 和相应的指针也会清零到初始状态。这将导致 MAC 发送器中生成下溢事件，并且帧发送将中止，此时帧的状态将同时标记下溢和清空事件（TDES0.UDE 和 TDES0.FFF）。清空操作期间，没有数据从 TxDMA 传输到 TxFIFO，TxDMA 根据清空的帧数（包括局部帧），将相应数量的传输发送状态字传输到应用。完全清空的帧的帧清空状态位（TDES0.FFF）将置 1。当 TxDMA 接收所有已清空的帧的状态字后，清空操作完成，DMA 工作模式寄存器中的 FTF 位（ENET_DMAMDR.FTF）随后将清零。此时，将接收来自 TxDMA 的新帧。清空操作完成后，将丢弃所有提交的数据，除非数据以 SOF 标记开头。

35.3.7.5 CRC&PAD 处理

当从应用接收的字节数少于 60（DA+SA+LEN+DATA）时，可通过设定发送描述符 TDES0 的 DPAD 位复位，向发送帧附加零（PAD），使数据长度正好为 46 字节，以满足 IEEE802.3 的最小数据字段要求，也可选择不附加任何填充值。

MAC 子控制器还会计算帧检查序列（FCS）字段的循环冗余校验（CRC），根据发送描述符 TDES0 的 DCRC 位和 CRCR 位的设定可选择将校验值插入或替换到正在发送的数据中，具体请参考常规型 Tx 描述符。

如果将 MAC 编程为不将 CRC 值附加到以太网帧的末尾，则不发送计算出的 CRC。但当 TDES0 的 DPAD 位为 0 时，小于 60 字节 (DA+SA+LEN+DATA) 的帧会在数据后附加填充 PAD，CRC 结果也将自动附加在填充帧的末尾。

35.3.7.6 源地址 SA 处理

从应用过来的数据如果不含源地址字段，MAC 可对该字段进行插入；如果含有源地址字段，可选择用 MAC 过滤地址寄存器 0 (ENET_MACAH/LR0) 或 MAC 过滤地址寄存器 1 (ENET_MACAH/LR1) 中设定的值对该字段进行替换。具体的对 SA 字段处理方式可以通过 MAC 控制寄存器的 SAIRC 位 (ENET_MACCR.SAIRC) 或描述符 TDES1 的 SAIRC 位进行控制。

注：在通过描述符方式使能该功能时，必须使能增强型 Tx 描述符。

35.3.7.7 VLAN TAG 处理

MAC 还可以对从应用来的 VLAN 帧的标识符字段进行插入、替换、删除处理。具体的处理方式可以通过 VLAN 标签发送寄存器 (ENET_MACVLTTR) 的 VLANC 位或描述符 TDES0 的 VLANC 位进行控制。

注：在通过描述符方式使能该功能时，必须使能增强型 Tx 描述符。

35.3.7.8 状态字

在向 MAC 子控制器传输以太网帧结束时，以及子控制器完成帧的发送后，发送状态会通过发送描述符提供给应用。发送状态的详细说明参见 TDES0。如果使能 IEEE1588 时间戳功能，将返回特定帧的 64 位时间戳以及发送状态。

35.3.7.9 暂停帧

在全双工模式下，当 MAC 流控寄存器的 FTE 位 (ENET_MACFCR.FTE) 置 1 时，MAC 子控制器将生成暂停帧并根据需要发送暂停帧。暂停帧与计算出的 CRC 附加在一起一并发送。可以通过两种方式启动暂停帧的生成。

通过将 MAC 流控寄存器中的 FCBBPA 位 (ENET_MACFCR.FCBBPA) 置 1 来请求流量控制，MAC 子控制器将生成并发送单个暂停帧。生成的帧中的暂停时间值通过 MAC 流控寄存器的 PSET 位 (ENET_MACFCR.PSET) 设定。要在先前发送的暂停帧中指定的时间之前延长或结束暂停时间，用户必须在应用程序中以适当的值编程 MAC 流控寄存器的 PSET 位 (ENET_MACFCR.PSET)，来设定暂停时间值，然后请求另一次暂停帧发送。

RxFIFO 填满时并且应用程序已请求流量控制，则 MAC 将生成并发送暂停帧。如果在此暂停时间结束前，RxFIFO 在可配置的时隙数 (通过 MAC 流控寄存器的 PLT 位 (ENET_MACFCR.PLT) 设定) 期间保持填满状态，将发送第二个暂停帧。只要 RxFIFO 保持填满状态，该过程将一直重复下去。如果在暂

停时间结束之前不再满足此条件，MAC 子控制器将发送一个暂停时间为零的暂停帧，向远程端口表明 RxFIFO 已准备好接收新数据帧。

35.3.7.10 发送 COE

在网络互联过程中，基本的通信协议（例如 TCP 和 UDP）都实施校验和，并配备校验和字段，这有助于确定通过网络发送的数据的完整性。由于以太网最广泛的用途是通过 IP 数据报文封装 TCP 和 UDP，因此以太网 MAC 子控制器具有校验和减荷功能（Checksum Offload Engine，简称 COE 引擎），该功能包含接收 COE 引擎和发送 COE 引擎，发送 COE 引擎支持发送路径中的校验和计算、插入以及接收路径中的校验和计算、检测。

COE 发送引擎通过完整的帧来计算 TCP、UDP 或 ICMP 的校验和，然后将其插入报文头的校验和字段。由于此要求，仅当 TxFIFO 配置为存储转发模式（通过 DMA 工作模式寄存器的 TSF 位（ENET_DMAMDR.TSF）置 1 来设定）时，才使能此功能。如果内核配置为阈值模式，则将绕过 COE 引擎。

必须确保 TxFIFO 足够深，使其能存储一个完整的帧，以便将该帧传输到 MAC 子控制器发送器。如果 FIFO 的深度小于输入以太网帧的大小，则将绕过 TCP/UDP/ICMP 校验和插入功能，仅支持 IPv4 报文头校验和插入功能，即使在存储转发模式下也是如此。

发送校验和减荷功能可通过设定发送描述符 TDSE0 的 CIC 位来实现每个帧的校验和控制。

在 IPv4 数据报文中，报文头字段的完整性由 16 位头校验和字段（IPv4 数据报文的第 11 字节和第 12 字节）指示。当以太网帧的类型字段值为 0x0800 且 IP 数据报文的版本字段值为 0x4 时，COE 引擎将检测到 IPv4 数据报文。计算期间，将忽略输入帧的校验和字段并将其替换为计算出的值。IPv6 报文头没有校验和字段，因此，COE 不修改 IPv6 报文头字段。IPv4 报文头校验和计算的结果由发送状态中的 IP 报文头错误状态位（TDES0.IPHE）指示。只要以太网类型字段的值和 IP 报文头版本字段的值不一致，或当以太网帧没有足够的数据（如 IP 报文头长度字段所指示）时，此状态位将置 1。

因此，总结下来，在以下情况下会发生 IP 报文头错误：

- IPv4 数据报文
 - ◆ 以太网类型为 0x0800，但 IP 报文头版本字段不等于 0x4
 - ◆ IPv4 报文头长度字段指示小于 0x5（20 字节）的值
 - ◆ 总的帧长度小于 IPv4 报文头长度字段给定的值
- IPv6 数据报文
 - ◆ 以太网类型为 0x86DD，但 IP 报文头版本字段不等于 0x6
 - ◆ 帧在 IPv6 报文头（40 字节）之前结束，或已完全接收到扩展报文头（在相应的报文头长度字段中给定）

如果以太网类型字段指示为 IPv4 有效负载，即使 COE 引擎测到 IP 报文头错误，也会插入 IPv4 报文头校验和。

TCP/UDP/ICMP 校验和功能会对 IPv4 或 IPv6 报文头（包括扩展报文头）进行处理，并确定封装的有效负载是 TCP、UDP 还是 ICMP。

在下面两种情况下，COE 引擎会无效：

- 非 TCP、非 UDP 或非 ICMP/ICMPv6 的有效负载
- 分段的 IP 帧（IPv4 或 IPv6）、带安全功能（例如，验证报文头或封装的安全有效负载）的 IP 帧和带路由报文头的 IPv6 帧

在 COE 引擎有效的情况下，会计算 TCP、UDP 或 ICMP 有效负载的校验和，然后将其插入报文头中相应的字段。它可工作在以下两种方式：

- TCP、UDP 或 ICMPv6 伪报文头未包含在校验和计算中，COE 引擎用计算出的校验和替换原校验和字段的伪报文头，作为最终的校验和字段
- TCP、UDP 或 ICMPv6 伪报文头包含在校验和计算中，COE 引擎将伪报文头一起参与校验和计算，并使用最终计算出的值插入到校验和字段

注：对于 ICMP-over-IPv4 数据包，由于没有为其定义伪报文头，因此在这两种模式下 ICMP 数据包中的校验和字段都必须始终为 0x0000。如果不等于 0x0000，可能向数据包插入不正确的校验和。

此操作的结果由发送描述符中的有效负载错误状态位（TDES0.IPCE）指示。当检测到下列情况之一时，IPCE 位置 1：

- 在存储转发模式下，帧已转发到 MAC 发送器，但帧结束未写入 FIFO
- 在接收到 IP 报文头中的有效负载长度字段指示的字节数前，数据包已结束

注：当数据包的长度大于指示的有效负载长度时，将字节作为填充字节忽略，并且不报告错误。检测到第一种类型的错误时，不修改 TCP、UDP 或 ICMP 报文头。对于第二种错误类型，计算的校验和仍将插入相应的报文头字段。

35.3.7.11 发送时序

下图 35-7、图 35-8 所示，是基本的发送时序图。

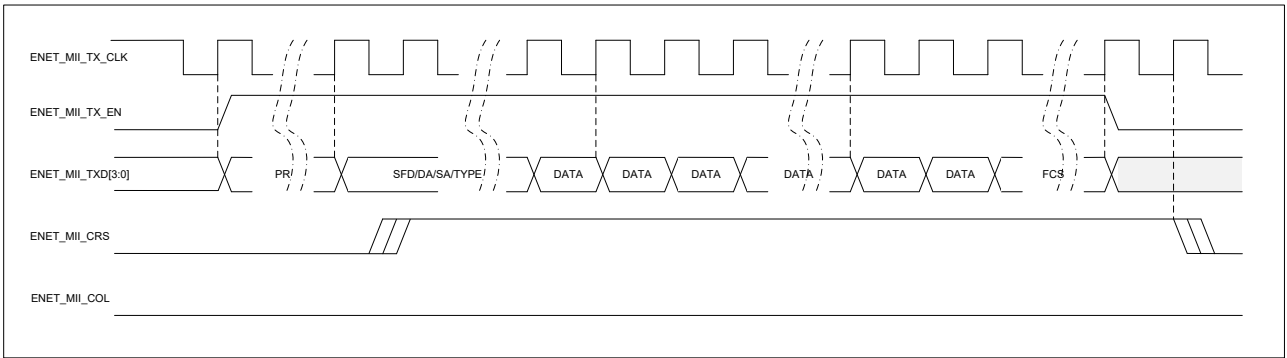


图 35-7 无错误时发送数据时序图

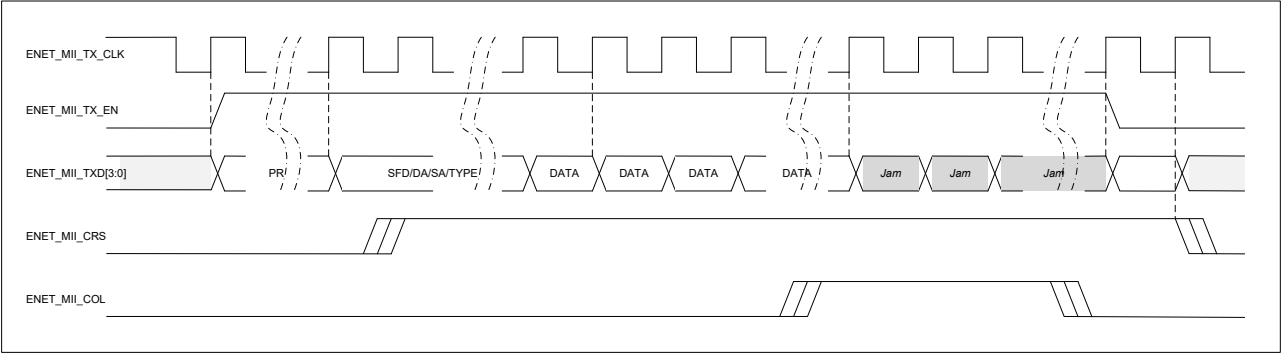


图 35-8 有错误时发送数据时序图

35.3.8 MAC 子控制器帧过滤

MAC 可以对接收到的各种帧进行过滤以选择应用需要的帧，帧过滤方式包括以下内容：

- MAC 帧的源地址 SA 过滤，包括完全过滤方式
- MAC 帧的目标地址 DA 过滤，包括完全过滤和 Hash 过滤两种方式
- VLAN 帧的标识符 Tag 过滤，包括完全过滤和 Hash 过滤两种方式

35.3.8.1 SA 完全过滤

MAC 子控制器可根据接收的帧的源地址字段来执行完全过滤。默认情况下，MAC 将 SA 字段与 MAC 地址寄存器中编程的值进行比较。可通过将 MAC 过滤地址寄存器中的 SELE 位（ENET_MACAHRx.SELE）置 1，来将 MAC 过滤地址寄存器 1~5 配置为 SA 字段进行比较。如果地址过滤控制寄存器中的 SAF 位（ENET_MACAFR.SAF）置 1，则 MAC 子控制器将丢弃未通过 SA 过滤的帧。否则，SA 过滤的结果将通过接收状态字中的状态位给出（参见接收描述符字 RDES0）。

注：SAF 位置 1 时，对 SA 过滤和 DA 过滤的结果进行与运算，以决定是否需要转发帧。这意味着任何一个过滤未通过都将丢弃帧。两个过滤必须都通过，才能将帧转发到应用。

MAC 子控制器在 MAC 过滤地址寄存器 1~5 中地址值的各个字节与接收的相应 SA 字节进行比较时，可以将寄存器中相应的屏蔽字节控制位置 1 来屏蔽该字节，从而实现 SA 的组地址过滤。

对于源地址 SA 过滤，可在最终输出时选择反转过滤匹配结果，即在 SA 地址比较匹配时认定为过滤失败。该功能由地址过滤控制寄存器的 SAIF 位（ENET_MACAFR.SAIF）控制。

35.3.8.2 DA 完全过滤

- DA（目标地址）过滤-单播

MAC 支持多达 6 个用于单播过滤的 MAC 目标地址。如果选择完全过滤（通过地址过滤控制寄存器中的 HU 位（ENET_MACAFR.HU）设为 0 来控制），MAC 将接收的单播地址的所有 48 位目标地址与编程的 MAC 过滤地址进行比较来确定是否匹配。默认情况下，始终使能 MAC 过滤地址寄存器 0，其它 MAC 过滤地址寄存器 1~5 则通过单独的使能位进行选择。



MAC 子控制器在 MAC 过滤地址寄存器 1~5 中地址值的各个字节与接收的相应 DA 字节进行比较，可以将寄存器中相应的屏蔽字节控制位置 1 来屏蔽该字节，从而实现 DA 的组地址过滤。

- DA（目标地址）过滤-多播

通过将地址过滤控制寄存器中的 PM 位（ENET_MACAFR.PM）置 1，将 MAC 子控制器编程为通过所有多播帧。如果 PM 位复位，MAC 子控制器将根据地址过滤控制寄存器中的 HM 位（ENET_MACAFR.HM）设定，执行对多播地址的过滤。在完全过滤模式下，将多播地址与编程的 MAC 过滤地址寄存器 1~5 进行比较。该过滤方式也支持组地址过滤。

- DA（目标地址）过滤-广播

在默认模式下，MAC 子控制器不过滤任何广播帧。但是，将地址过滤控制寄存器中的 BCF 位（ENET_MACAFR.BCF）置 1 来将 MAC 编程为拒绝所有广播帧，则会丢弃任何广播帧。

对于目标地址 DA 过滤（单播、多播、广播），也可在最终输出结果时选择反转过滤匹配结果，即在 DA 地址比较匹配时认定为过滤失败。该功能由地址过滤控制寄存器的 DAIF 位（ENET_MACAFR.DAIF）控制，同样适用于 Hash 过滤结果。

35.3.8.3 DA Hash 过滤

对于单播目标地址和多播目标地址的过滤也支持 Hash 过滤方式。在 Hash 过滤模式（地址过滤控制寄存器中的 HU/HM 位（ENET_MACAFR.HU/HM）置 1）下，MAC 将使用 64 位 Hash 表执行对单播地址或多播地址的不完全过滤，即 Hash 过滤。对于 Hash 过滤，MAC 将使用接收的目标地址的 6 个高 CRC 位来索引 Hash 表的内容。值为 000000 时，选取所选寄存器中的位 0；值为 111111 时，选取散列表寄存器中的位 63。如果相应位（由 6 位 CRC 指示）已置 1，将认为单播帧或多播帧已通过 Hash 过滤，否则认为帧未能通过 Hash 过滤。

注：CRC 是使用下列多项式编码的 32 位值：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

35.3.8.4 VLAN TAG 完全过滤

当接收帧为 VLAN 帧时，MAC 可通过设定地址过滤控制寄存器的 VLTF 位（ENET_MACAFR.VLTF）来使能对 VLAN 标识符字段的过滤。过滤用比较值设置在 VLAN TAG 接收寄存器的 VLFLT 位（ENET_MACVLTRR.VLFLT）中，同时还可通过设定 VLAN TAG 接收寄存器的 VTSEL 位（ENET_MACVLTRR.VTSEL）来选择对全部 VLAN 标识符字段进行比较过滤还是仅比较低 12 位标识符。

对于 VLAN 标识符字段的过滤，也可在最终输出时选择反转过滤匹配结果，即在标识符字段比较匹配时认定为过滤失败。该功能由 VLAN TAG 接收寄存器的 VTIM 位（ENET_MACVLTRR.VTIM）控制，同样适用于 Hash 过滤结果。

35.3.8.5 VLAN TAG Hash 过滤

对于 VLAN 标识符的过滤也可以选择 Hash 过滤方式。在 Hash 过滤模式有效时（通过 VLAN TAG 接收寄存器的 VTHM 位（ENET_MACVLTRR.VTHM）置 1 来控制），MAC 将使用 16 位 Hash 表执行对 VLAN 标识符的不完全过滤，即 Hash 过滤。对于 Hash 过滤，MAC 将使用 VLAN 标识符的 CRC 位来索引 Hash 表的内容。值为 1 时，标明通过 VLAN 标识符过滤；值为 0 时，标明未通过 VLAN 标识符过滤。

注：CRC 是使用下列多项式编码的 32 位值：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

35.3.9 MAC 子控制器回环模式

MAC 子控制器支持对发送到其接收器的帧进行回送。默认情况下，禁止 MAC 回送功能，可通过配置 MAC 控制寄存器中的 LM 位（ENET_MACCR.LM）使能该功能。

35.3.10 PMT 电源管理

ENET 控制器具有电源管理（PMT）机制。可通过远程唤醒帧使能位以及魔术数据包使能位使能 PMT 模块。应用程序通过配置 PMT 控制寄存器的 WKEN 位或 MPEN 位（ENET_PMTCR.WKEN、ENET_PMTCR.MPEN）来选择允许接收的对应包。在使能低功耗模式（通过设定 PMT 控制寄存器的 PWDN 位（ENET_PMTCR.PWDN）来控制）时，MAC 将丢弃所有接收到的帧并且不会将这些帧转发给应用。仅当使能相应检测位且接收到对应的远程唤醒帧或魔术数据包时，MAC 控制器才会退出低功耗模式。

35.3.10.1 远程唤醒帧过滤

ENET 控制器有 8 个唤醒帧过滤寄存器。要对每个寄存器执行写操作，需要逐个值加载远程唤醒帧过滤寄存器（ENET_PMTRWFR）。连续加载远程唤醒帧过滤寄存器八次，便可对远程唤醒帧过滤寄存器加载所需的值。读操作与写操作相同，要读取八个值，必须读唤醒帧过滤寄存器八次后，才能到达最后一个寄存器。每次读/写操作都会将唤醒帧过滤寄存器指针指向下一个过滤寄存器。通过读取远程唤醒帧过滤寄存器的 RTWKPT 位（ENET_PMTCR.RTWKPT）可以知道当前的读写操作在哪个过滤寄存器上，同时可通过对该寄存器的 RTWKFR 位（ENET_PMTCR.RTWKFR）置 1 复位内部的指针，复位之后可对该远程唤醒帧过滤寄存器重新进行操作。下图 35-9 所示是 ENET 内部的 8 个远程唤醒帧过滤寄存器。

ENET_PMTRWFR 0	FLT0 Byte Mask							
ENET_PMTRWFR 1	FLT1 Byte Mask							
ENET_PMTRWFR 2	FLT2 Byte Mask							
ENET_PMTRWFR 3	FLT3 Byte Mask							
ENET_PMTRWFR 4	-	FLT3 Command	-	FLT2 Command	-	FLT1 Command	-	FLT0 Command
ENET_PMTRWFR 5	FLT3 Offset		FLT2 Offset		FLT1 Offset		FLT0 Offset	
ENET_PMTRWFR 6	FLT1 CRC16				FLT0 CRC16			
ENET_PMTRWFR 7	FLT3 CRC16				FLT2 CRC16			

图 35-9 远程唤醒帧过滤寄存器

各部分的功能说明如下：

● 字节掩码（Byte Mask）

该寄存器定义过滤器 i 检测帧的哪些字节来确定帧是否为唤醒帧（i=0~3）。MSB（第 31 位）必须为零，位 j[30:0] 为字节掩码。如果将字节掩码的位 j（字节数）置 1，则传入帧的过滤器 i 偏移 +j 由 CRC 模块处理；否则将忽略过滤器 i 偏移 +j。

● 命令（Command）

该 4 位命令控制过滤器 i 操作（i=0~3）。bit3 为地址类型选择，置位时只检测多播帧，反之只检测单播帧；bit2 和 bit1 为保留位；bit0 为过滤器 i 的使能位，置位时使能过滤器 i，反之禁止过滤器 i。

● 偏移（Offset）

该寄存器定义过滤器 i 要检测的帧的偏移（i=0~3）。该 8 位模式偏移是要检测的过滤器 i 第一个字节的偏移。允许的最小值为 12，表示帧的第 13 个字节（偏移值 0 表示帧的第一个字节）。

● CRC16（CRC）

该寄存器包含预先写入的 CRC16 值（i=0~3），用于与帧数据（过滤器 i Offset 且 Byte Mask 对应之后）计算的 CRC16 值进行比较。

CRC16 的生成多项式为： $G(x) = x^{16} + x^{15} + x^2 + 1$ 。

35.3.10.2 远程唤醒帧检测

当 MAC 处于低功耗模式并已将 PMT 控制寄存器中的远程唤醒 WKEN 位（ENET_PMTCR.WKEN）



置 1 时，MAC 可在接收到远程唤醒帧后恢复正常工作。PMT 支持四个可编程过滤器，这些过滤器可由应用配置为不同的接收帧模式。如果传入帧通过过滤器的地址过滤，并且过滤器 CRC-16 与传入的检测模式相匹配，则可接收唤醒帧。只需要检查唤醒帧是否存在长度错误、FCS 错误、帧尾错误、MII 错误、冲突，确保其不是矮帧，即使唤醒帧长度超过 512 字节，但只要帧的 CRC 值有效，帧便有效。

PMT 控制寄存器中的唤醒帧检测状态 WKFR 位 (ENET_PMTCR.WKFR) 会指示收到远程唤醒帧。此外，还会生成 PMT 中断（如果已使能）来指示已接收到远程唤醒帧。

35.3.10.3 魔术数据包检测

魔术数据包是基于 AMD 公司的魔术数据包技术对网络上处于睡眠模式下的器件上电。MAC 接收称为魔术数据包的特定信息包，此信息包的目标地址是网络上的节点。MAC 控制器只对目标地址为本器件或多播地址的魔术数据包进行检查，以确定这些数据包是否满足唤醒要求。对通过地址过滤（单播或多播）的魔术数据包进行检测，确定其是否符合远程唤醒数据格式，即 6 个字节的数据所有位为全 1 的数据包加上重复出现 16 次的 MAC 地址。

应用通过向 PMT 控制寄存器中的 MPEN 位 (ENET_PMTCR.MPEN) 写入 1 来使能魔术数据包唤醒。PMT 模块持续监测目标地址为对应于指定魔术数据包模式的节点的每个帧。检查每个接收帧的目标地址和源地址字段之后是否为 0xFFFFFFFF 形式的同步数据流。之后，PMT 模块检查帧是否存在重复 16 次且没有任何断开或中断的 MAC 地址。如果重复 16 次的地址中存在中断，则会再次扫描传入帧中是否存在 0xFFFFFFFF 形式。这 16 次重复可位于帧中的任何位置，但必须在同步数据流后面（0xFFFFFFFF）。只要检测到重复 16 次的 MAC 地址，器件就可以接收多播帧。例如，如果节点 MAC 地址为 0x001122334455，则 MAC 扫描的数据序列为：

```
Destination Address Source Address ... FFFF FFFF FFFF

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455

... CRC
```

认为接收到了一个魔术唤醒包。PMT 控制寄存器 (ENET_PMTCR) 中的魔术数据包检测会针对接收到的魔术数据包进行更新。此外，还会生成 PMT 中断（如果已使能）来指示已接收到魔术数据包。

35.3.10.4 低功耗模式控制

在低功耗模式期间应保持 MAC 接收器使能状态，原因是此时要对魔术数据包/远程唤醒帧进行相关的接收检测，通过将 MAC 控制寄存器中的 RE 位 (ENET_MACCR.RE) 置 1 来使能 MAC 接收器。



在低功耗模式期间应保持 MAC 发送器关闭状态。通过将 MAC 控制寄存器中的 TE 位（ENET_MACCR.TE）清零来关闭 MAC 发送器。

在低功耗模式期间应禁止以太网 DMA，原因是此时无需将魔术数据包/远程唤醒帧传送到系统存储器。通过将工作模式寄存器中的 STT 位和 STR 位（ENET_DMAMDR.STT/ENET_DMAMDR.STR）清零来关闭 TxDMA 和 RxDMA。

推荐的低功耗进入和唤醒顺序如下：

- step1.禁止 TxDMA，并等待所有之前的帧发送完成
- step2.通过将 MAC 控制寄存器中的 RE 位和 TE 位（ENET_MACCR.RE/TE）清零来禁止 MAC 发送器和 MAC 接收器
- step3.等待 RxDMA 清空 RxFIFO 中的所有帧
- step4.禁止 RxDMA
- step5.通过将 PMT 控制寄存器中的 WKEN/MPEN 位（ENET_PMTCR.WKEN/MPEN）置 1 来使能远程唤醒帧/魔术数据包检测
- step6.通过将 PMT 控制寄存器中的 PWDN 位（ENET_PMTCR.PWDN）置 1 来使能 MAC 低功耗模式
- step7.通过将 MAC 控制寄存器中的 RE 位（ENET_MACCR.RE）置 1 来使能 MAC 接收器
- step8.等待接收魔术数据包和远程唤醒帧
- step9.接收到有效唤醒帧时，以太网控制器退出低功耗模式
- step10.读取 PMT 控制寄存器（ENET_PMTCR）以清零 PMT 事件标志，使能 MAC 发送器、TxDMA 和 RxDMA

35.3.11 MMC 管理计数

MAC 管理计数器（MMC）搭载一组寄存器来收集有关已接收帧和已发送帧的统计信息。其中包括一个用于控制各寄存器行为的 MMC 控制寄存器（ENET_MMCCR）、两个包含当前接收和发送统计状态的状态寄存器（ENET_MMCRSR、ENET_MMCTSR）、两个包含接收和发送中断控制的中断寄存器（ENET_MMCRIR、ENET_MMCTIR），以及各种帧类型的统计寄存器。这些统计寄存器均可从应用中访问，寄存器章节中详细描述了这些统计寄存器的用途和意义，请参考寄存器相关章节。

MAC 管理计数器（MMC）针对通过地址过滤的接收帧更新相应的统计寄存器，已丢弃帧的统计信息不会进行更新，除非丢弃的帧为小于 6 字节的矮帧（DA 字节不能完全接收）。

如果帧成功发送，则将发送的帧视为“好帧”。即，如果帧发送过程没有因为以下错误而中止，则认为发送的帧是“好帧”：

- Jabber 超时
- 无载波/载波丢失

- 延迟冲突
- 帧下溢
- 过度延迟
- 过度冲突

如果帧接收成功，则将接收的帧视为“好帧”。即，如果接收帧不存在以下错误，则认为接收帧是“好帧”：

- CRC 错误
- 矮帧（短于 64 字节）
- 对齐错误（仅限 10/100Mb/s）
- 长度错误（仅限非类型帧）
- 超出范围（仅限非类型帧，超过最大大小）
- ENET_MII_RXER 输入错误

35.3.12 DMA 子控制器基本特征

ENET 的 DMA 子控制器具有独立的发送引擎（TxDMA）和接收引擎（RxDMA）。发送引擎将数据从系统存储器传送到 TxFIFO，而接收引擎将数据从 Rx FIFO 传送到系统存储器。DMA 可以在 CPU 完全干预的情况下，通过描述符有效地将数据在 FIFO 和系统存储器之间传送。该 DMA 控制器可经过应用编程，在完成帧发送和接收操作时以及其它正常/错误条件下产生 CPU 中断。ENET 的 DMA 和 MCU 内核之间通过两种数据结构进行通信：

- 控制和状态寄存器
- 描述符列表和数据缓冲区

DMA 子控制器既可将 MAC 接收到的数据帧传送系统的数据缓冲区，也可以将系统数据缓冲区中的数据发送给 MAC，描述符中包含指向这些数据缓冲区的指针。DMA 子控制器共有两个描述符列表：一个用于接收的 Rx 描述符列表，一个用于发送的 Tx 描述符列表。Rx 描述符列表和 Tx 描述符列表的基地址分别通过描述符地址寄存器（ENET_DMARXDSAR、ENET_DMATXDSAR）设定。该描述符列表是一种前向链表（无论是隐式还是显式），最后一个描述符会指回第一个描述符以构成环形结构。描述符列表位于系统的物理存储空间。

每个描述符最多可指向两个缓冲区，这样便能使用两个物理寻址的缓冲区替代存储器中两个连续的缓冲区。通过配置 Rx 描述符 RDES1 的 RCH 位和 Tx 描述符 TDES0 的 TCH 位来完成描述符的显式链接。

数据缓冲区同样位于系统的物理存储空间，通常由整个帧或部分帧组成，但不会超过单个帧。数据缓冲区中仅包含数据，数据通信的状态保存在描述符中。检测到帧结束时，DMA 子控制器会跳到下一个帧缓冲区，数据链接可实现跨越多个数据缓冲区存储帧，可以使能或禁止数据链接。描述符有两种结构方式：环形结构与链接结构，其实现方式如图 35-10 所示。

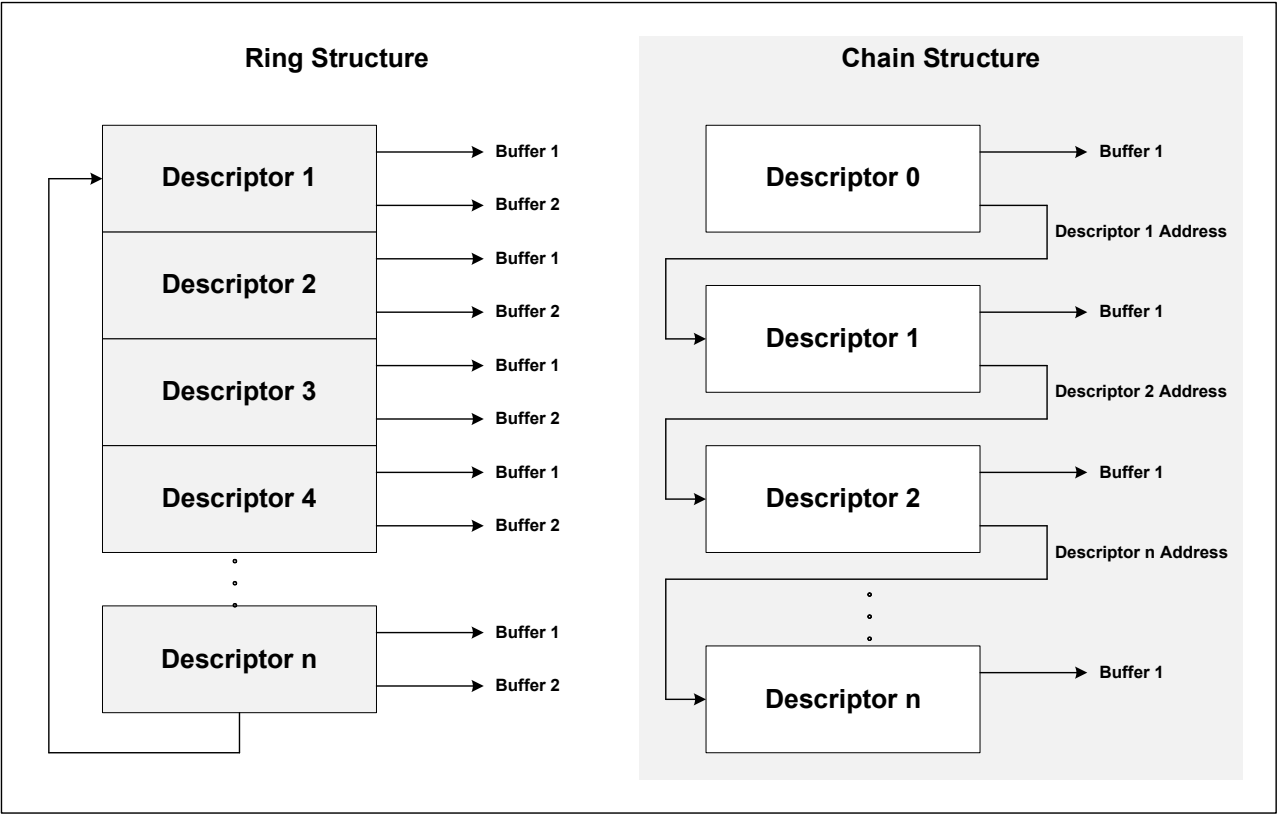


图 35-10 ENET 描述符结构

35.3.12.1 突发访问

DMA 子控制器会尝试在 AHB Master 接口上执行固定长度的突发传送（通过 DMA 总线控制寄存器中的 FBST 位（ENET_DMABSR.FBST）进行相应配置）。突发的最大长度由 DMA 总线控制寄存器中的 PBL 位（ENET_DMABSR.PBL）指示和限制。对于要读取的 16 个字节，接收和发送描述符始终采用可能的最大突发大小进行访问。

仅当 TxFIFO 中的空间足以容纳配置的突发或帧结束之前的字节数时（当帧短于配置的突发长度时），TxDMA 才会启动数据传送。TxDMA 会向 AHB Master 接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4（4 拍增量突发）、INCR8（8 拍增量突发）、INCR16（16 拍增量突发）和 SINGLE（单一传输）的最佳搭配来传送数据。否则（非固定长度突发）会使用 INCR（未定义长度）与 SINGLE 传送数据。

仅当 RxFIFO 中有足够的数据用于配置的突发时，或者在 RxFIFO 中检测到帧结束时（当帧短于配置的突发长度时），RxDMA 才会启动数据传送。RxDMA 会向 AHB Master 接口指示起始地址和所需的传送次数。当 AHB 接口配置为固定长度突发时，将使用 INCR4、INCR8、INCR16 和 SINGLE 的最佳搭配来传送数据，如果在 AHB 接口上的固定突发结束前已到达帧结束，将执行空传送以完成固定长度的突发传送。否则（非固定长度突发）会使用 INCR（未定义长度）和 SINGLE 传送数据。

当 AHB 接口配置为地址对齐的节拍（通过 DMA 总线控制寄存器中的 AAL 位（ENET_DMABSR.AAL）配置）时，两个 DMA 引擎会确保 AHB 启动的第一次突发传送小于或者等于已配置 PBL 位（ENET_DMABSR.PBL）的大小。这样，后续的所有节拍都会从与已配置 PBL 位

(ENET_DMABSR.PBL) 对齐的地址开始。由于 AHB 接口不支持多于 INCR16 的传送，DMA 只能对齐节拍最大为 16 (PBL>16) 的地址。

35.3.12.2 对齐访问

发送和接收数据缓冲区在起始地址对齐方面没有任何限制。缓冲区的起始地址可与四个字节中的任意一个对齐。但是，DMA 子控制器始终在地址与总线宽度对齐时启动传输，并且在不需要的字节通道上传输空数据。这通常发生在以太网帧的开始或结束传送期间。

- 缓冲区读操作示例：

如果发送缓冲区地址为 0x00000FF2，并且需要传送 15 个字节，则 TxDMA 将从地址 0x00000FF0 读取 5 个全字，但在将数据传送到 Tx FIFO 时，丢弃或忽略额外的字节（前两个字节）。同样地，还将忽略最后一次传送的最后 3 个字节。TxDMA 始终确保向 Tx FIFO 传送的是全 32 位数据，除非是帧结束。

- 缓冲区写操作示例：

如果接收缓冲区地址为 0x00000FF2，并且需要传送接收到的帧的 16 个字节，则 RxDMA 将从地址 0x00000FF0 开始写入 5 个全 32 位数据。但是，在第一次传送的前 2 个字节与第三次传送的最后 2 个字节将包含空数据。

35.3.12.3 缓冲器计算

DMA 子控制器不会更新发送和接收描述符中的缓冲区大小字段，只更新描述符的状态字段。应用必须用驱动程序计算缓冲区的大小。

TxDMA 会向 MAC 内核传送准确的字节数（由 TDES1 中的缓冲区大小字段指示）。如果将描述符标记为第一个描述符 (TDES0.TFS=1)，则 DMA 会将缓冲区的第一次传送标记为帧起始。如果将描述符标记为最后一个描述符 (TDES0.TLS=1)，则 DMA 会将数据缓冲区的最后一次传送标记为帧结束。

RxDMA 持续将数据传送至缓冲区，直到缓冲区已满或接收到帧结束为止。当 Rx 描述符的 RFS 位置 1 时，如果未将描述符标记为最后一个描述符 (RDES0.RLS)，则该描述符对应的缓冲区会填满，有效数据量由 RDES1 中的缓冲区大小指示；如果将描述符标记为最后一个描述符，则缓冲区不会填满，有效数据量将通过缓冲区大小字段减去数据缓冲指针偏移 (RDES0.FL) 得到的结果表示，当数据缓冲区指针与数据总线宽度对齐时，偏移为零。

注：即使当接收缓冲区的起始地址与系统数据总线宽度未对齐时，系统也应分配一个大小与系统总线宽度对齐的接收缓冲区。例如，如果系统分配一个起始地址为 0x1000、大小为 1024 字节 (1KB) 的接收缓冲区，则软件可将接收描述符中的缓冲区起始地址编程为具有 0x1002 偏移。RxDMA 将帧写入该缓冲区，其中前两个单元 (0x1000 和 0x1001) 中为空数据。实际帧从单元 0x1002 开始写入。因此，尽管已将缓冲区大小编程为 1024 字节，但由于存在起始偏移地址，因此该缓冲区的实际有用空间为 1022 字节。

35.3.12.4 DMA 仲裁

DMA 子控制器内的仲裁器会在发送和接收通道对 AHB Master 接口进行的访问之间进行仲裁。可以使用两类仲裁：循环优先级和固定优先级。如果选择循环优先级仲裁（通过 DMA 总线控制寄存器的 DMAA 位（ENET_DMABSR.DMAA）来控制），仲裁器会在发送和接收 DMA 同时请求访问时，按照 DMA 总线控制寄存器的 FTPR 位（ENET_DMABSR.FTPR）设置的优先比率分配数据总线；如果选择固定优先级仲裁（通过 DMA 总线控制寄存器的 DMAA 位（ENET_DMABSR.DMAA）来控制），仲裁器会设定接收 DMA 总是比发送 DMA 对总线拥有更高的访问优先级线。

35.3.12.5 错误响应

对于由 DMA 子控制器发起的任何数据传送，如果从机给出错误响应，则相应 DMA 将停止所有操作并更新 DMA 状态寄存器中的总线错误 EBUS 位（ENET_DMASR.EBUS）和致命总线错误 FBS 位（ENET_DMASR.FBS）。此时，该 DMA 子控制器只能在软复位或硬复位外设以及重新初始化 DMA 之后才能恢复操作。

35.3.13 DMA 子控制器接收功能

35.3.13.1 DMA 操作

RxDMA 引擎的操作顺序如下：

- step1.用户设置接收描述符（RDES0-RDES3）并将 Rx 描述符的 OWN 位置 1
- step2.工作模式寄存器的 STR 位（ENET_DMAMDR.STR）置 1 后，RxDMA 进入运行状态。在运行状态下，RxDMA 轮询接收描述符列表，尝试获取空闲描述符。如果获得的描述符不空闲（由 CPU 所拥有），则 RxDMA 进入挂起状态并跳转到第 9 步
- step3.RxDMA 将所获取描述符中的接收数据缓冲区地址解码
- step4.处理传入帧并将其放入所获取描述符的数据缓冲区
- step5.当缓冲区已满或帧传输完成时，RxDMA 引擎将获取下一个描述符
- step6.如果当前的帧传输完成，RxDMA 将继续执行第 7 步。如果 RxDMA 未拥有下一个接收描述符且帧传输尚未完成（尚未传输 EOF 字段），则 RxDMA 会将 RDES0 的 DPE 位置 1（除非禁止刷新）。RxDMA 关闭当前描述符并通过将最后一个描述符位（RDES0.RLS）清零来将其标记为中间描述符（如果禁止刷新，则将其标记为最后一个描述符），随后继续执行第 8 步；如果 RxDMA 已拥有下一个描述符，但当前的帧传输尚未完成，则 RxDMA 将关闭当前描述符作为中间值并返回第 4 步
- step7.如果已使能 IEEE1588 时间戳功能，DMA 会将时间戳（如果可用）写入当前描述符的 RDES6 和 RDES7。RxDMA 随后获取所接收帧的状态并将该状态字写入当前描述符的 RDES0，同时 OWN 位清零且最后一个描述符位置 1

- step8.RxDMA 引擎检查最新描述符的 OWN 位。如果 CPU 拥有该描述符 (RDES0.OWN =0)，则 DMA 状态寄存器的接收缓冲区不可用 RUS 位 (ENET_DMASR.RUS) 置 1 且 RxDMA 引擎进入挂起状态 (第 9 步)；如果 RxDMA 拥有该描述符，引擎将返回第 4 步并等待下一个帧。
- step9.在 RxDMA 引擎进入挂起状态前，将从 RxFIFO 中刷新部分帧 (使用 DMA 工作模式寄存器的 DFRF 位 (ENET_DMAMDR.DFRF) 控制刷新)
- step10.当收到接收轮询要求命令或者可以从 RxFIFO 获得下一个帧的起点时，RxDMA 将退出挂起状态。引擎继续执行第 2 步并重新获取下一个描述符

35.3.13.2 帧获取处理

RxDMA 引擎始终会尝试为即将传入的帧获取一个额外的描述符。只要满足以下任一条件，即尝试获取描述符：

- RxDMA 进入运行状态后，DMA 工作模式寄存器的接收启动 STR 位 (ENET_DMAMDR.STR) 立即置 1
- 在当前传输的帧结束前，当前描述符的数据缓冲区已满
- MAC 子控制器已完成数据帧接收，但当前的接收描述符尚未关闭
- 接收过程因描述符由 CPU 所拥有 (RDES0.OWN=0) 而挂起，并且接收到新帧
- 已发出接收轮询要求命令

当帧通过地址过滤且其大小大于或等于为 RxFIFO 所设置的可配置阈值字节数时，或者在存储转发模式下将整个帧写入 RxFIFO 时，RxDMA 才会将接收的帧传输到系统存储器；当帧未通过地址过滤时，将丢弃该帧 (除非地址过滤控制寄存器的接收所有 RALL 位 (ENET_MACAFR.RALL) 置 1)。不足 64 字节的帧由于会发生冲突或过早结束，因而可从 RxFIFO 中清除。接收 64 (可配置阈值) 个字节后，RxDMA 开始将帧数据传输到当前描述符所指定的接收缓冲区中。DMAAHB 接口准备好接收数据传输后，将第一个描述符位 (RDES0.RFS) 置 1，以分隔该帧。数据缓冲区已满或者帧的末段已传输到接收缓冲区时，OWN (RDES0) 位将复位为 0，此时释放描述符。如果帧包含在单个描述符中，则最后一个描述符位 (RDES0.RLS) 和第一个描述符位 (RDES0.RFS) 均置 1。

RxDMA 获取下一个帧的描述符时，将最后一个描述符位 (RDES0.RLS) 置 1，并释放前一个帧描述符中的状态位，然后将 DMA 状态寄存器的接收完成状态 RCS 位 (ENET_DMASR.RCS) 置 1。该过程将重复执行，直至 RxDMA 遇到被标记为由 CPU 所有的描述符。这种情况时，接收过程将 DMA 状态寄存器中接收缓冲区不可用 RUS 位 (ENET_DMASR.RUS) 置 1，随后进入挂起状态，但其在接收列表中的位置仍然保留。

35.3.13.3 过程挂起

如果在接收过程处于挂起状态时有新的接收帧到达，则 RxDMA 将重新获取系统存储器中的当前描述符。如果该描述符现在由 RxDMA 所拥有，接收过程将重新进入运行状态并开始接收帧；如果该描述符

仍由主机所拥有，则默认情况下，RxDMA 将丢弃 RxFIFO 顶部的当前帧并将丢失帧计数器递增。如果 RxFIFO 中存储了多个帧，将重复执行该过程。

若将 DMA 工作模式寄存器的 DFRF 位（ENET_DMAMDR.DFRF）置 1 后，可避免丢弃或刷新 RxFIFO 顶部的帧。在这种情况下，接收过程将 DMA 状态寄存器的接收缓冲区不可用 RUS 位（ENET_DMASR.RUS）置 1 并返回到挂起状态。

35.3.14 DMA 子控制器发送功能

35.3.14.1 DMA 操作

在连续帧传送过程中，TxDMA 根据 DMA 操作模式寄存器的 OSF 位（ENET_DMAMDR.OSF）的设定不同，对应的第二帧处理时序不同。当 OSF 位置 1 时，发送过程完成第一个帧的传送后，会立即轮询第二个帧的发送描述符列表，如果第二个帧有效，发送过程会在写入第一个帧的状态信息前发送第二帧。这种发送过程无需关闭第一个帧的状态描述符便可同时获取两个帧的方式称之为 OSF 模式，反之为非 OSF 模式。

在 ENET_DMAMDR.OSF=0 下的操作顺序：

- step1.在数据缓冲区设置好相应的以太网帧数据后，用户配置发送描述符（TDES0-TDES3）并将 OWN 位置 1
- step2.DMA 工作模式寄存器的 STT 位（ENET_DMAMDR.STT）置 1 后，TxDMA 立即进入运行状态
- step3.在运行状态下，TxDMA 为需要传送的帧轮询发送描述符列表。轮询启动后，TxDMA 会以连续的描述符环形顺序或链接顺序持续进行。如果 TxDMA 检测到标记为 CPU 所有的描述符（TDES0.OWN=0），或者发生错误条件，会挂起传送并将发送缓冲区不可用位（DMA 工作模式寄存器的 TUS 位（ENET_DMASR.TUS）与正常中断汇总状态位（DMA 工作模式寄存器的 AIS 位（ENET_DMASR.AIS）置 1。发送引擎继续执行第 9 步
- step4.如果获得的描述符标记为 TxDMA 所有（TDES0.OWN=1），则 TxDMA 将根据取得的描述符对发送数据缓冲区地址进行解码
- step5.TxDMA 从系统存储器中获取发送数据并传送这些数据
- step6.如果以太网帧保存在多个描述符的数据缓冲区中，则 TxDMA 将关闭中间描述符并获取下一个描述符。重复执行第 3、4 和 5 步，直到完成以太网帧数据的传送
- step7.完成帧传送后，状态信息将写入该发送描述符的各状态位（TDES0）。此步骤中会清零 OWN 位，此时该描述符变为由 CPU 所有
- step8.帧发送完成后，在其最后一个描述符位（TDES0.TLS）置 1 时，DMA 状态寄存器的 TCS 位（ENET_DMASR.TCS）将置 1。随后 TxDMA 引擎返回第 3 步
- step9.在挂起状态下，TxDMA 会在接收到发送轮询要求时尝试重新获取描述符（返回第 3 步），并将下溢中断状态位清零

在 ENET_DMAMDR.OSF=1 下的操作顺序：

- step1.TxDMA 的操作过程和 OSF=0 模式下 TxDMA 的第 1~6 步相同
- step2.无需关闭前一帧的最后一个描述符，TxDMA 便可获取下一个描述符
- step3.如果 TxDMA 拥有所需描述符，便会对该描述符中的发送缓冲区地址进行解码。如果 TxDMA 未拥有描述符，则会进入挂起模式并跳到第 7 步
- step4.TxDMA 从系统存储器中取得发送帧并发送该帧，直到帧数据发送结束，如果该帧拆分到多个描述符中，则会同时关闭中间描述符
- step5.TxDMA 等待前一个帧的发送状态。当状态可用时，TxDMA 将状态写入相应 TDES0，进而关闭描述符
- step6.TxDMA 获取下一个描述符，然后继续执行第 3 步（状态正常时）。如果上一个发送状态显示下溢错误，则 TxDMA 会进入挂起模式（第 7 步）
- step7.在挂起模式下，如果 TxDMA 接收到挂起状态，会将状态写入相应的 TDES0。之后，将相关中断置 1 并返回挂起模式
- step8.只有在接收到发送轮询要求（通过写入 DMA 发送轮询寄存器（ENET_DMATXPDR））后，TxDMA 才会退出挂起模式并进入运行状态（转到第 1 步或第 2 步，具体取决于挂起状态）

35.3.14.2 帧获取处理

系统数据缓冲区须包含完整的以太网帧，不包括前导码、PAD 和 FCS 字段，DA 字段、SA 字段和类型/长度字段须包含在有效数据中。如果发送描述符指示 MAC 子控制器必须禁止插入 CRC 或填充字节，则缓冲区必须具有包括 CRC 字节的完整以太网帧（不包括前导码）。帧可以采用数据链接形式，也可以跨越多个缓冲区。帧必须由第一个描述符位（TDES0.TFS）和最后一个描述符位（TDES0.TLS）界定。传送启动时，必须将 TDES0 的 TFS 位置 1，之后帧数据便从存储器缓冲区传送到 TxFIFO。同时，如果当前帧的 TDES0 的 TLS 位为 0，发送过程将尝试获取下一个描述符。TDES0 的 TLS 位为 0，则指示中间缓冲区，TDES0 的 TLS 位为 1，则指示帧的最后一个缓冲区。当帧的最后一个缓冲区完成传送后，TxDMA 会将最终状态信息写回描述符的（TDES0）相关状态位。此时，如果 TDES0 的 TEIE 位置 1，则 DMA 状态寄存器的发送完成 TCS 位（ENET_DMASR.TCS）将置 1，并获取下一个描述符，然后重复执行该过程。

根据 DMA 工作模式寄存器的 TSF 位（ENET_DMAMDR.TSF）的设定，实际的帧传送过程在 TxFIFO 达到可编程的发送阈值时（根据 DMA 工作模式寄存器的 TTC 位（ENET_DMAMDR.TTC）设定），或者 TxFIFO 中包含完整的帧时启动传送过程。TxDMA 完成帧的传送后会释放描述符（清零 TDES0 的 OWN 位）。

35.3.14.3 过程挂起

可通过以下任意条件暂停发送轮询：

- TxDMA 检测到 CPU 所有的描述符 (TDES0.OWN=0)，并且 DMA 状态寄存器的发送缓冲区不可用 TUS 位 (ENET_DMASR.TUS) 置 1。如要恢复，驱动程序必须将描述符的所有权交给 TxDMA，然后发出轮询查询要求命令
- 检测到由下溢导致的传送错误时，将中止帧的传送。相应的发送描述符 (TDES0) 位将置 1

如果 TxDMA 由于第一个条件而进入挂起状态，则 DMA 状态寄存器的正常中断状态 NIS 位与发送缓冲区不可用 TUS 位 (ENET_DMASR.NIS/ENET_DMASR.TUS) 均置 1；如果发生第二个条件，DMA 状态寄存器的异常中断状态 AIS 位与发送下溢 UNS 位 (ENET_DMASR.AIS/ENET_DMASR.UNS) 都将置 1，并且信息将写入发送描述符，从而导致挂起。两种情况下，发送列表中的位置都会保留。保留的位置是 TxDMA 关闭的最后一个描述符后面的描述符位置。纠正挂起原因后，驱动程序必须明确发出发送轮询要求命令。

35.3.15 DMA 子控制器初始化

DMA 的初始化步骤如下：

- step1.对 DMA 总线控制寄存器 (ENET_DMABSR) 执行写操作以设置总线访问参数
- step2.对 DMA 中断寄存器 (ENET_DMAIR) 执行写操作以屏蔽不必要的中断源
- step3.创建描述符列表，对 DMA 接收轮询寄存器 (ENET_DMARXDSAR) 和 DMA 发送轮询寄存器 (ENET_DMATXDSAR) 执行写操作，为 DMA 提供各列表的起始地址
- step4.对 MAC 寄存器执行写操作设定所需的过滤选项
- step5.对 MAC 控制寄存器 (ENET_MACCR) 执行写操作以配置和使能发送与接收工作模式
- step6.对 DMA 工作模式寄存器 (ENET_DMAMDR) 执行写操作，将 DMA 工作模式寄存器的 STT 位 (ENET_DMAMDR.STT) 和 STR 位 (ENET_DMAMDR.STR) 置 1 以启动发送和接收
- step7.发送与接收引擎进入运行状态，并尝试从相应描述符列表中获取描述符，这两个引擎随后开始处理接收和发送操作。发送和接收处理过程彼此独立，可单独进行启动或停止

35.3.16 DMA 子控制器描述符

35.3.16.1 常规型 RX 描述符

RxDMA 的常规型接收描述符有 4 个，由四个 32 位字组成，如表 35-2 所示，分别定义为 RDES0、RDES1、RDES2 和 RDES3。

表 35-2 常规型 RX 描述符

编号	位描述					
RDES0	OWN[31]	状态位[30:0]				
RDES1	控制位	Res.	字节计数缓冲区 2	控制位	Res.	字节计数缓冲区 1



编号	位描述					
	[31]		[28:16]	[15:14]		[12:0]
RDES2	缓冲地址 1[31:0]					
RDES3	缓冲地址 2[31:0]					

下面分别对 RDES0~RDES3 的具体位做说明。

● RDES0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	DAF	FL[13:0]													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERSM	DPE	SAF	LEE	OVE	VLANT	RFS	RLS	IPCE	LCOE	FRAT	WTE	REE	DBE	CRE	DAS

Bit	Field	Description
31	OWN	描述符所有权 (Description Owner) 0: 该描述符为 CPU 所有 1: 该描述符为 DMA 所有 DMA 在帧接收完成或此描述符的关联缓冲区已满时将该位清零
30	DAF	目标地址过滤失败 (DA Filter Fail) 该位指示帧未能通过 MAC 内核中的 DA 过滤
29:16	FL	帧长 (Frame Length) 该位指示传输到主机存储器中的接收帧字节长度 (含 CRC)。只有在最后一个描述符位 (RDES0.RLS) 置 1, 并且描述符错误位 (RDES0.DPE) 和上溢错误位 (RDES0.OVE) 复位时, 该字段才有效。当最后一个描述符位和错误汇总位均未置 1 时, 该字段指示为当前帧传输的累计字节数
15	ERSM	接收错误汇总 (Error Receive Summary) 该描述符的 B[14]、B[11]、B[7]、B[6]、B[4]、B[3]、B[1] 位及 REDS4 描述符的 B[4]、B[3] 位中有一个被置位, 该位也随之被置位
14	DPE	描述符错误 (Description Error) 该位指示某个帧因超过当前描述符缓冲区的大小而被截断以及 DMA 未拥有下一个描述符 注: 该位只有在最后一个描述符 (RDES0.RLS) 置 1 时才有效
13	SAF	源地址过滤失败 (SA Filter Fail) 该位指示帧的 SA 字段未能通过 MAC 内核中的 SA 过滤
12	LEE	长度字段错误 (Length Error) 该位指示接收帧的实际长度与长度/类型字段的值不符 注 1: 该位仅在帧类型位 (RDES0.FRAT) 复位后有效 注 2: 在 CRC 错误位 (RDES0.CRE) 有效时, 该位无效
11	OVE	上溢错误 (Overflow Error) 该位指示接收帧因缓冲区上溢而损坏
10	VLANT	VLAN 标识符帧 (VLAN Tag) 该位指示描述符所指向的帧是由 MAC 内核标记的 VLAN 帧

Bit	Field	Description
9	RFS	接收首描述符 (Receive First Description) 该位指示此描述符包含帧的第一个缓冲区。如果第一个缓冲区的大小为 0, 则第二个缓冲区将包含帧的帧头; 如果第二个缓冲区的大小为 0, 则下一个描述符将包含帧的帧头
8	RLS	接收尾描述符 (Receive Last Description) 该位指示此描述符指向的缓冲区为帧的最后一个缓冲区
7	IPCE	IP COE 引擎错误 (IP Checksum Error) 当 Checksum Offload 功能有效时: 该位指示 IPv4 或 IPv6 报文头中存在错误。导致此错误的原因可能是: 1) 以太网类型字段与 IP 报文头版本字段值不一致 2) IPv4 中报文头的校验和不匹配 3) 以太网帧缺少所需的 IP 报文头字节数 当时间戳功能有效时: 该位指示有一个时间戳快照被记录在了 RDES6 和 RDES7 中, 该位只在最后一个描述符 (RDES0.RLS) 置 1 时有效 以上两个均无效时: 该位指示 MAC 接收到一个巨帧
6	LCOE	延迟冲突错误 (Late Collision Error) 该位指示延迟冲突错误, 在以半双工模式接收帧时发生了延迟冲突
5	FRAT	帧类型 (Frame Type) 0: MAC 接收到以太网帧 1: MAC 接收到 PTP 帧 注: 当接收到一个矮帧时, 该位无效
4	WTE	看门狗计数器错误 (Watchdog Count Error) 该位指示接收看门狗计时器在接收当前帧时超时, 且当前帧在看门狗超时后被截断
3	REE	接收错误 (Receive Error) 该位指示在帧接收期间, PHY 发出 RX_DV 信号时, 发出了 RX_ER 信号
2	DBE	Dribble 位错误 (Dribble Bit Error) 该位指示接收的帧具有非整数倍数的字节 (奇数半字节) 注: 该位仅在 MII 模式下有效
1	CRE	CRC 错误 (CRC Error) 该位指示接收的帧发生循环冗余校验 (CRC) 错误 注: 该位只在最后一个描述符 (RDES0.RLS) 置 1 有效
0	DAS	目标地址过滤成功 (DA Filter Success) 在 COE 功能和 PTP 功能无效时: 0: 接收帧通过了 MAC 地址寄存器 0 的 DA 地址过滤 1: 接收帧通过了 MAC 地址寄存器 1~5 的 DA 地址过滤 注: 当该描述符的 DAF 位置位时, 该位无效。 在 COE 功能或 PTP 功能有效时: 该位指示 RDES4 描述符有效 注: 该位只在最后一个描述符 (RDES0.RLS) 置 1 时有效

● RDES1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REID	Res.		RBS2[12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RER	RCH	Res.	RBS1[12:0]												

Bit	Field	Description
31	REID	接收完成且禁止中断（Receive End & Interrupt Disable） 该位置 1 时，表明接收帧在由该描述符所指示的缓冲区中结束，并禁止引发主机的中断，同时也阻止 DMA 状态寄存器的 RCS 位（ENET_DMASR.RCS）置 1 注：该位仅在最后一个描述符位（RDES0.RLS）置 1 的情况下有效
28:16	RBS2	接收缓冲区大小 2（Receive Buffer Size 2） 该位以字节为单位指示第二个数据缓冲区的大小 注：该位在该描述符的 RCH 位置 1 时无效
15	RER	接收描述符结束（Receive Description End of Ring） 该位指示描述符列表已到达其最后一个描述符，DMA 会返回描述符列表的首地址，形成一个描述符环
14	RCH	接收描述符链（Receive Description Chain） 该位置 1 时，描述符中的第二个地址是下一个描述符地址，而非第二个缓冲区地址，此时该描述符的 RBS2 位为无效 注：RER 位优先级高于 RCH 位
12:0	RBS1	接收缓冲区大小 1（Receive Buffer Size 1） 该位以字节为单位指示第一个数据缓冲区的大小。如果该字段为 0，DMA 将忽略该缓冲区并使用缓冲区 2 或下一个描述符，具体取决于该描述符的 RCH 的设置

● RDES2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBA1[15:0]															

Bit	Field	Description
31:0	RBA1	接收缓冲区地址 1（Receive Buffer Address 1） 该位向 DMA 指示数据在存储器中的位置，当软件为 DMA 提供此描述符（RDES0.OWN=1）时，这些位将指示缓冲区 1 的物理地址

● RDES3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBA2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBA2[15:0]															

Bit	Field	Description
31:0	RBA2	接收缓冲区地址 2 (Receive Buffer Address 2) 该位向 DMA 指示数据在存储器中的位置，当软件为 DMA 提供此描述符 (RDES0.OWN=1) 并且使用描述符环结构时，这些位将指示缓冲区 2 的物理地址；如果 RDES1.RCH 位置 1，则该地址包含下一个描述符所在物理寄存器的指针 注：只有在 RDES1.RCH 位置 1 时，缓冲区地址指针才必须与总线宽度相符

35.3.16.2 增强型 RX 描述符

增强型 Rx 描述符结构由 8 个 32 位字组成，如表 35-3 所示，分别定义为 RDES0、RDES1、RDES2、RDES3、RDES4、RDES5、RDES6 和 RDES7。其中 RDES0~RDES3 的功能与常规型相同。

表 35-3 增强型 RX 描述符

编号	位描述					
RDES0	OWN[31]	状态位[30:0]				
RDES1	控制位 [31]	Res.	字节计数缓冲区 2 [28:16]	控制位 [15:14]	Res.	字节计数缓冲区 1 [12:0]
RDES2	缓冲地址 1[31:0]					
RDES3	缓冲地址 2[31:0]					
RDES4	扩展状态位[31:0]					
RDES5	Reserved					
RDES6	时间戳低位时间[31:0]					
RDES7	时间戳高位时间[31:0]					

下面分别对 RDES0~RDES7 的具体位做说明。

- RDES0

同常规型描述符 RDES0 的功能。

- RDES1

同常规型描述符 RDES1 的功能。

- RDES2

同常规型描述符 RDES2 的功能。

- RDES3

同常规型描述符 RDES3 的功能。

- RDES4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TSPD	PTPV	PTPFT	MTP[3:0]				IPV6FR	IPV4FR	IPCB	IPPE	IPHE	IPPT[2:0]		

Bit	Field	Description
14	TSPD	时间戳丢弃 (Timestamp Dropped) 该位指示由于 RxFIFO 溢出, 虽进行了时间戳快照但是丢弃了时间戳
13	PTPV	PTP 版本 (PTP Version) 该位指示接收到的 PTP 帧的版本类型 0: IEEE1588v1 版本的 PTP 帧 1: IEEE1588v2 版本的 PTP 帧
12	PTPFT	PTP 帧类型 (PTP Frame Type) 该位指示接收到的 PTP 帧类型 0: Over UDP-IPv4 或 Over UDP-IPv6 类型的 PTP 帧 1: Over Ethernet 类型的 PTP 帧 注: 具体为 Over UDP-IPv4 还是 Over UDP-IPv6 可通过该描述符的 IPv6FR、IPv4FR 位获取
11:8	MTP	报文类型 (Message Type) 该位指示接收到的报文类型 0000: 未接收到任何 PTP 消息 0001: SYNC (所有时钟类型) 0010: Follow_Up (所有时钟类型) 0011: Delay_Req (所有时钟类型) 0100: Delay_Resp (所有时钟类型) 0101: Pdelay_Req (在点对点透明时钟中) 0110: Pdelay_Resp (在点对点透明时钟中) 0111: Pdelay_Resp_Follow_Up (在点对点透明时钟中) 1000: Announce 1001: Management 1010: Signaling 1111: 默认报文类型 其它: 保留
7	IPv6FR	IPv6 报文 (IPv6 Packet Received) 该位指示接收到 IPv6 报文
6	IPv4FR	IPv4 报文 (IPv4 Packet Received) 该位指示接收到 IPv4 报文
5	IPCB	绕过 COE 引擎 (IP COE Bypass) 该位表示绕过 COE 引擎
4	IPPE	有效负载错误 (IP Payload Error) 该位指示 IP 有效负载错误 以下情况时, 该位置 1: 1) 当内核计算的 16 位 IP 有效负载校验和 (即 TCP、UDP 或 ICMP 校验和) 与接收帧中对应的校验和字段不匹配 2) 当 TCP、UDP 或 ICMP 段长度与 IP 报文头字段中的有效负载长度值不匹配
3	IPHE	IP 报文头错误 (IP Header Error) 该位指示由内核计算的 16 位 IPv4 报文头校验和与接收的校验和字节不匹配, 或 IP 报文头版本字段与以太网类型值不一致

Bit	Field	Description
2:0	IPPT	IP 负载类型 (IP Payload Type) 该位表示 IP 数据报中封装的有效负载类型, 当 IP 报文头出错或存在分段的 IP 时, 这些位为 “000” 000: 未知, 或未处理 IP 有效负载 001: UDP 010: TCP 011: ICMP 1xx: 保留

- RDES5

Reserved。

- RDES6

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

Bit	Field	Description
31:0	RTSL	接收时间戳低位 (Receice Timestamp Low) DMA 将相应接收帧所捕获的时间戳低 32 位更新进该字段 注: 只有在描述符中最后一个描述符位 (RDES0.RLS) 置 1 时, 该字段才包含时间戳

- RDES7

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

Bit	Field	Description
31:0	RTSH	接收时间戳高位 (Receice Timestamp High) DMA 将相应接收帧所捕获的时间戳高 32 位更新进该字段 注: 只有在描述符中最后一个描述符位 (RDES0.RLS) 置 1 时, 该字段才包含时间戳

35.3.16.3 常规型 TX 描述符

TxDMA 的常规型接收描述符有 4 个, 由四个 32 位字组成, 如表 35-4 所示, 分别定义为 TDES0、TDES1、TDES2 和 TDES3。

表 35-4 常规型 TX 描述符

编号	位描述					
TDES0	OWN [31]	控制位 [30:26]	TTSE [25]	控制位 [24:18]	TTSS [17]	状态位 [16:0]
TDES1	控制位 [31:29]	字节计数缓冲区 2 [28:16]				Res. [12:0]
TDES2	缓冲地址 1[31:0]					
TDES3	缓冲地址 2[31:0]					

下面分别对 TDES0~TDES3 的具体位做说明。

● TDES0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	TEIE	TLS	TFS	DCRC	DPAD	TTSE	CRCR	CIC[1:0]		TER	TCH	VLANC[1:0]		TTSS	IPHE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSM	JTE	FFF	IPCE	LCAE	NCE	LCE	ECE	VLAN F	COC[3:0]			EDE	UDE	DEE	

Bit	Field	Description
31	OWN	描述符所有权 (Description Owner) 0: 该描述符为 CPU 所有 1: 该描述符为 DMA 所有 DMA 在完成帧发送时或在描述符分配的缓冲区全部读取完成后将该位清零。 属于同一个帧的所有后续描述符置 1 后, 也应该将帧的第一个描述符的 OWN 位置 1
30	TEIE	发送完成且使能中断 (Transfer End & Interrupt Enable) 该位置 1 时, 当前帧发送完毕后, DMA 状态寄存器的 TCS 位 (ENET_DMASR.TCS) 置 1 注: 该位仅在最后一个描述符位 (TDES0.TLS) 置 1 的情况下有效
29	TLS	发送尾描述符 (Transfer Last Description) 该位指示此描述符指向的缓冲区为帧的最后一个缓冲区
28	TFS	发送首描述符 (Transfer First Description) 该位指示此描述符包含帧的第一个缓冲区。如果第一个缓冲区的大小为 0, 则第二个缓冲区将包含帧的帧头; 如果第二个缓冲区的大小为 0, 则下一个描述符将包含帧的帧头
27	DCRC	无效 CRC (Disable CRC) 该位置 1 时, MAC 不会将循环冗余校验附加到所发送帧的末尾 注: 该位仅在第一个描述符位 (TDES0.TFS) 置 1 的情况下有效
26	DPAD	无效 PAD (Disable PAD) 0: MAC 会自动为不足 64 字节的帧添加补位项和 CRC, 不管该描述符的 DCRC 位的设定 1: MAC 不会自动为不足 64 字节的帧添加补位项 注: 该位仅在第一个描述符位 (TDES0.TFS) 置 1 的情况下有效

Bit	Field	Description
25	TTSE	发送时间戳使能 (Transfer Timestamp Enable) 该位置 1 时, 将针对描述符所描述的发送帧激活 IEEE1588 硬件时间戳功能 注: 该位仅在第一个描述符位 (TDES0.TFS) 置 1 的情况下有效
24	CRCR	CRC 替换 (CRC Replace) 该位置 1 时, MAC 将计算好的 CRC 值替换即将发送的发送帧的 FCS 字段 注: 只有当该描述符的 TDES0.DCRC 位置 1 时, 该位有效
23:22	CIC	Checksum 插入控制 (Checksum Insert Control) 这些位控制校验和的计算与插入, 如下所示: 00: 禁止插入校验和 01: 仅使能 IP 报文头校验和的计算与插入 10: 使能 IP 报文头校验和以及 TCP/UCP/ICMP 校验和的计算与插入, 但不会在硬件中计算伪报文头校验和 11: 使能 IP 报文头校验和以及 TCP/UDP/ICMP 校验和的计算与插入, 并在硬件中计算伪报文头校验和 注: 该位仅在最后一个描述符位 (TDES0.TLS) 置 1 的情况下有效
21	TER	发送描述符结束 (Transfer Description End of Ring) 该位指示描述符列表已到达其最后一个描述符, DMA 会返回描述符列表的首地址, 形成一个描述符环
20	TCH	发送描述符链 (Transfer Description Chain) 该位置 1 时, 描述符中的第二个地址是下一个描述符地址, 而非第二个缓冲区地址, 此时该描述符的 TBS2 位为无效 注: TER 位优先级高于 TCH 位
19:18	VLANC	VLAN 控制 (VLAN Tag Control) 该位指示对 VLAN 帧的 Tag 字段处理 00: 对 VLAN 帧不做处理 01: 在发送前将 VLAN 帧的标记和类型字段删除 10: 将 VLAN 标记发送寄存器 (ENET_MACVLTR) 中的 VLANV 标记值插入到发送帧中 11: 用 VLAN 标记发送控制寄存器 (ENET_MACVLTR) 中的 VLANV 标记值替换原发送帧中的 VLAN 标识符
17	TTSS	发送时间戳状态 (Transfer Timestamp Status) 该位指示已为当前发送帧捕获到时间戳, 置 1 时 TDES6 和 TDES7 将保存发送帧捕获到的时间戳值 注: 该位仅在最后一个描述符位 (TDES0.TLS) 置 1 的情况下有效
16	IPHE	IP 报文头错误 (IP Header Error) 该位置 1 时指示 MAC 发送器在 IP 数据报文头中检测到错误 注: 当 COE 引擎检测出一个 IP Header 错误时, 它仍然会将计算好的 Checksum 值插入到 IPv4 Header Checksum 字段
15	ETSM	发送错误汇总 (Error Transfer Summary) 该描述符的 B[16]、B[14]、B[13]、B[12]、B[11]、B[10]、B[9]、B[8]、B[2]、B[1] 位中有一个被置位, 该位也随之被置位

Bit	Field	Description
14	JTE	Jabber 超时错误 (Jabber Timeout Error) 该位置 1 时表示 MAC 发送器经历了 Jabber 超时 注：只有在 MAC 控制寄存器的 JAB 位 (ENET_MACCR.JAB) 未置 1 时，该位才会置 1
13	FFF	帧刷新完成 (Frame Flush Finish) 该位指示 DMA 已依照 CPU 发出的软件刷新命令刷新帧
12	IPCE	IP COE 引擎错误 (IP Checksum Error) 该位指示 COE 引擎在 TCP、UDP 或 ICMP 报文有效负载中检测到错误，且不会更新原帧中的 Checksum 字段
11	LCAE	丢失载波错误 (Lost Carrier Error) 该位指示帧发送期间丢失载波。即，帧发送期间有一个或多个发送时钟周期的 ENET_MII_CRS 信号无效 注：该位仅对在 MAC 处于半双工模式时实现无冲突发送的帧有效
10	NCE	无载波错误 (No Carrier Error) 该位指示在发送期间未由 PHY 触发载波侦听信号
9	LCE	延迟冲突错误 (Late Collision Error) 该位指示帧发送过程因冲突窗口 (MII 模式下为 64 个字节时间，含前导码) 后出现冲突而中止 注：如果该描述符的 UDE 错误位置 1，则该位无效
8	ECE	过度冲突错误 (Excessive Collision Error) 该位指示尝试发送当前帧时因出现 16 个连续冲突而中止发送 注：如果 MAC 控制寄存器中的 RETY (禁止重试) 位 (ENET_MACCR.RETY) 置 1，则此位会在出现首个冲突后置 1 并且帧发送过程将中止
7	VLANF	VLAN 标识符帧 (VLAN Tag Frame) 该位指示所发送的帧为 VLAN 帧
6:3	COC	冲突个数 (Count of Collision) 该位指示发送帧之前出现的冲突个数 注：过度冲突位 (TDES0.ECE) 置 1 时，该计数无效
2	EDE	过度延时错误 (Excessive Delay Error) 该位指示因出现超过 24288 个位时间的过度延迟而中止发送 注：该位在 MAC 控制寄存器中的延迟检查 DLYC 位 (ENET_MACCR.DLYC) 置 1 时有效
1	UDE	下溢错误 (Underflow Error) 该位指示 MAC 因发送缓冲区的数据未及时到达而中止了帧发送，产生下溢错误，指明 DMA 在帧发送期间遇到发送缓冲区为空的情况
0	DEE	延时错误 (Delay Error) 该位指示 MAC 在发送前因存在载波而延迟 注：该位仅在半双工模式下有效

● TDES1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAIRC[2:0]			TBS2[12:0]												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			TBS1[12:0]												

Bit	Field	Description
31:29	SAIRC	SA 插入替换控制 (SA Insert Replace Control) 该位指示对发送帧的 SA 字段进行如下处理： 001：将 MAC 地址寄存器 0 中的地址值作为 SA 地址插入到发送帧中 101：将 MAC 地址寄存器 1 中的地址值作为 SA 地址插入到发送帧中 010：将 MAC 地址寄存器 0 中的地址值作为 SA 地址替换发送帧中的 SA 字段 110：将 MAC 地址寄存器 1 中的地址值作为 SA 地址替换发送帧中的 SA 字段 其它值：无操作
28:16	TBS2	发送缓冲区大小 2 (Transfer Buffer Size 2) 该位以字节为单位指示第二个数据缓冲区的大小。 注：该位在该描述符的 TCH 位置 1 时无效
12:0	TBS1	发送缓冲区大小 1 (Transfer Buffer Size 1) 该位以字节为单位指示第一个数据缓冲区的大小。如果该字段为 0，DMA 将忽略该缓冲区并使用缓冲区 2 或下一个描述符，具体取决于该描述符的 TCH 的设置

● TDES2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBA1[15:0]															

Bit	Field	Description
31:0	TBA1	发送缓冲地址 1 (Transfer Buffer Address 1) 该位向 DMA 指示数据在存储器中的位置，当软件为 DMA 提供此描述符 (TDES0.OWN=1) 时，这些位将指示缓冲区 1 的物理地址

● TDES3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBA2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBA2[15:0]															

Bit	Field	Description
31:0	TBA2	发送缓冲地址 2 (Transfer Buffer Address 2) 该位向 DMA 指示数据在存储器中的位置，当软件为 DMA 提供此描述符 (TDES0.OWN=1) 并且使用描述符环结构时，这些位将指示缓冲区 2 的物理地址；如果 TDES0.TCH 位置 1，则该地址包含下一个描述符所在物理寄存器的指针 注：只有在 TDES0.TCH 位置 1 时，缓冲区地址指针才必须与总线宽度相符

35.3.16.4 增强型 TX 描述符

增强型 Tx 描述符结构由 8 个 32 位字组成，如表 35-5 所示，分别定义为 TDES0、TDES1、TDES2、TDES3、TDES4、TDES5、TDES6 和 TDES7。其中 TDES0~TDES3 的功能与常规型相同。

表 35-5 增强型 TX 描述符

编号	位描述					
TDES0	OWN [31]	控制位 [30:26]	TTSE [25]	控制位 [24:18]	TTSS [17]	状态位 [16:0]
TDES1	控制位 [31:29]	字节计数缓冲区 2 [28:16]				Res. [12:0]
TDES2	缓冲地址 1[31:0]					
TDES3	缓冲地址 2[31:0]					
TDES4	Reserved					
TDES5	Reserved					
TDES6	时间戳低位时间[31:0]					
TDES7	时间戳高位时间[31:0]					

下面分别对 TDES0~TDES7 的具体位做说明。

● TDES0

同常规型描述符 TDES0 的功能。

● TDES1

同常规型描述符 TDES1 的功能。

● TDES2

同常规型描述符 TDES2 的功能。

● TDES3

同常规型描述符 TDES3 的功能。

● TDES4

Reserved。

● TDES5

Reserved。

● TDES6



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

Bit	Field	Description
31:0	TTSL	发送时间戳低位 (Transfer Timestamp Low) DMA 将相应接收帧所捕获的时间戳低 32 位更新进该字段 注: 只有在描述符中最后一个描述符位 (TDES0.TLS) 置 1 且发送时间戳状态位 (TDES0.TTSS) 置 1 时, 该字段才包含时间戳

● TDES7

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

Bit	Field	Description
31:0	TTSH	发送时间戳高位 (Transfer Timestamp High) DMA 将相应接收帧所捕获的时间戳高 32 位更新进该字段 注: 只有在描述符中最后一个描述符位 (TDES0.TLS) 置 1 且发送时间戳状态位 (TDES0.TTSS) 置 1 时, 该字段才包含时间戳

35.3.17 PTP 子控制器基本特征

IEEE1588 标准定义了一种协议, 此协议支持使用网络通信、局域计算和分布式对象等技术实现测量和控制系统中的精密时钟同步。该协议适用于利用支持多播消息传送的局域网 (包括但不限于以太网) 进行通信的系统。该协议用于对非均匀系统进行同步, 这类系统包含固有精度、分辨率和稳定性都不断变化的时钟。该协议支持亚秒范围的系统级同步精度, 并且需要极少的网络和本地时钟计算资源。这种基于消息的协议称为精密时间协议 (PTP), 它通过 UDP/IP 传送。系统或网络归类为主节点和从节点, 用于分配时序/时钟信息。该协议通过交换 PTP 报文来同步从节点与主节点, 如图 35-11 所示。

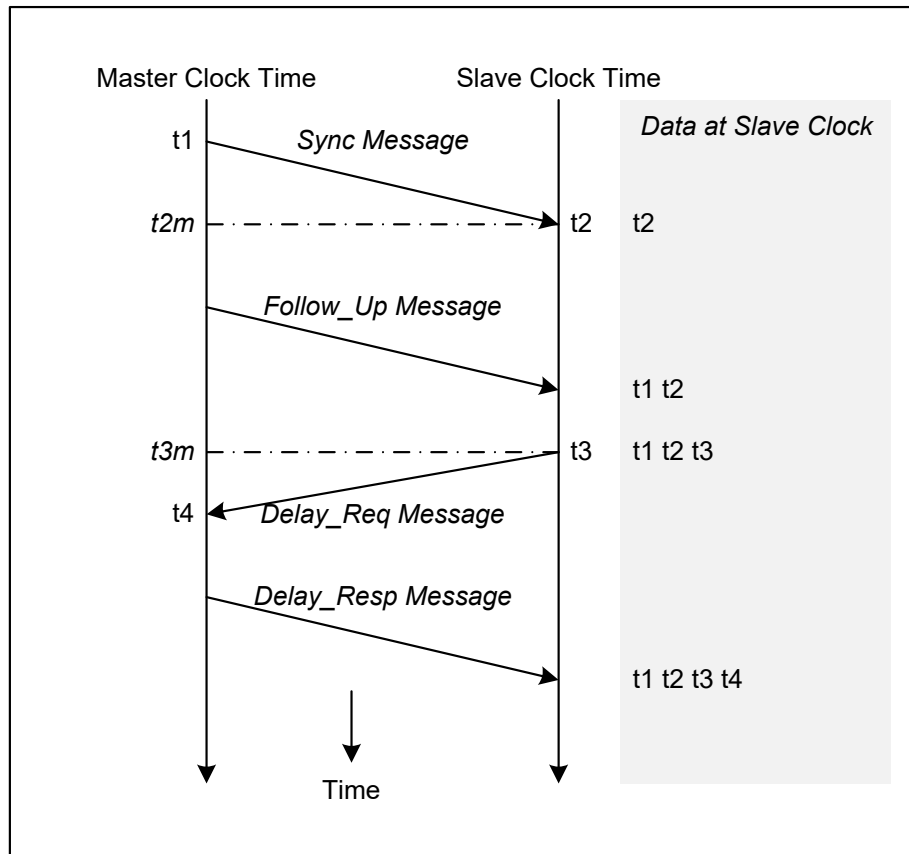


图 35-11 PTP 时钟同步图

- step1.主节点向其所有节点广播 PTP 同步报文。同步报文包含主节点的参考时间信息。报文离开主节点系统的时间为 t_1
- step2.从节点接收同步报文并利用其参考时序捕获准确时间 t_2
- step3.主节点随后会向从节点发送一个包含 t_1 信息的 Follow_up 报文以备后用
- step4.从节点向主节点发送一个 Delay_Req 报文，标出该帧离开从节点的准确时间 t_3
- step5.主节点接收该报文，捕获报文进入其系统的准确时间 t_4
- step6.主节点利用 Delay_Resp 报文将 t_4 信息发送给从节点
- step7.从节点使用 t_1 、 t_2 、 t_3 和 t_4 这四个值来同步其本地参考时序与主节点的参考时序

虽然大多数协议都在 UDP 层之上通过软件实现，但需要硬件完成对特定 PTP 包进入或离开以太网端口时的准确时间进行捕获。

35.3.18 PTP 子控制器参考时钟

根据 IEEE1588 规范的定义，要获取时间快照，内核需要一个 64 位格式的参考时间（分成两个 32 位通道，高 32 位表示时间的秒数，低 32 位表示时间的亚秒数）。

PTP 参考时钟用于在内部生成参考时间（亦称作系统时间）以及捕获时间戳。该参考时钟的频率必须大于等于时间戳计数器的分辨率。主节点与各从节点之间的同步精度目标约为 100ns。精度取决于 PTP 参考时钟输入周期、振荡器的特性（漂移）以及同步过程的频率。

由于从 Tx 发送时钟域和 Rx 接收时钟域到 PTP 参考时钟域的同步,时间戳锁存值的不确定度为 1 个参考时钟周期。

35.3.19 PTP 子控制器报文类型

时间戳功能使能后,可通过 PTP 控制寄存器的 PTPTYPS 位(ENET_PTPCR.PTPTYPS)来控制对哪一个或哪一类报文进行时间戳快照。具体的对应关系如下表 35-6 所示。

表 35-6 PTP 报文类型

PTPTYPS[3:0]	PTP 使能的报文类型
00X0	SYNC、Follow_Up、Delay_Req、Delay_Resp
0001	SYNC
0011	Delay_Req
01X0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up
0101	SYNC、Pdelay_Req、Pdelay_Resp
0111	Delay_Req、Pdelay_Req、Pdelay_Resp
10XX	SYNC、Delay_Req
11XX	Pdelay_Req、Pdelay_Resp

35.3.20 PTP 子控制器接收功能

使能 IEEE1588 时间戳功能时,以太网 MAC 将捕获接口上接收到的所有帧的时间戳。MAC 会在完成帧接收过程时提供时间戳。

捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时间戳会随帧的接收状态一同发送回相应的接收描述符内。64 位时间戳信息会写回 RDES6 和 RDES7 字段,其中 RDES6 会保存时间戳的 32 个低有效位,RDES7 会保存时间戳的 32 个高有效位。

35.3.21 PTP 子控制器发送功能

使能 IEEE1588 时间戳功能时,在发送接口上输出帧的 SFD 时,将捕获时间戳。对于需要捕获时间戳的帧,可对每个发送帧进行标记以指示是否有必要捕获该帧的时间戳。

无需处理发送帧即可识别 PTP 帧,可通过发送描述符中的控制位执行帧控制。

捕获到的时间戳返回给应用的方式与提供帧状态的方式相同。时间戳会随帧的发送状态一同发送回相应的发送描述符内,从而自动将时间戳与特定 PTP 帧相连。64 位时间戳信息会写回 TDES6 和 TDES7 字段,其中 TDES6 会保存时间戳的 32 个低有效位,TDES7 会保存时间戳的 32 个高有效位。

35.3.22 PTP 子控制器时间校准

使用 PTP 参考时钟更新 64 位 PTP 时间。该 PTP 时间可用作时钟源,以获取传输接口上发送或

接收的以太网帧的快照（时间戳）。可使用粗校准方式或细校准方式对系统时间定时器进行初始化或校准。

使用粗校准方式时，初始值或偏移值会写入时间戳更新寄存器（ENET_PTPBSR/ENET_PTPUNSR）。对于初始化，会将该寄存器中的值（初始值）写入系统时间计数器；对于系统时间校准，会将该寄存器中的值（偏移值）加到系统时间中或从系统时间中减去。

使用细校准方式时，从参考时钟频率相对于主时钟（如 IEEE1588 中定义）的偏移会在一段时间内进行校准，而不像粗校准方式中那样，在单个时钟周期内进行校准。校准时间越长，越有助于保持线性时间，并且不会导致各 PTP 同步消息间隔之间的参考时间发生剧烈变化（或者大型抖动）。在此方法中，会使用一个累加器对基本加数寄存器（ENET_PTPBSAR）中的内容求和，如图 35-12 所示。累加器生成的算数进位将用作使系统时间计数器递增的脉冲。累加器和基本加数寄存器（ENET_PTPBSAR）均为 32 位寄存器。

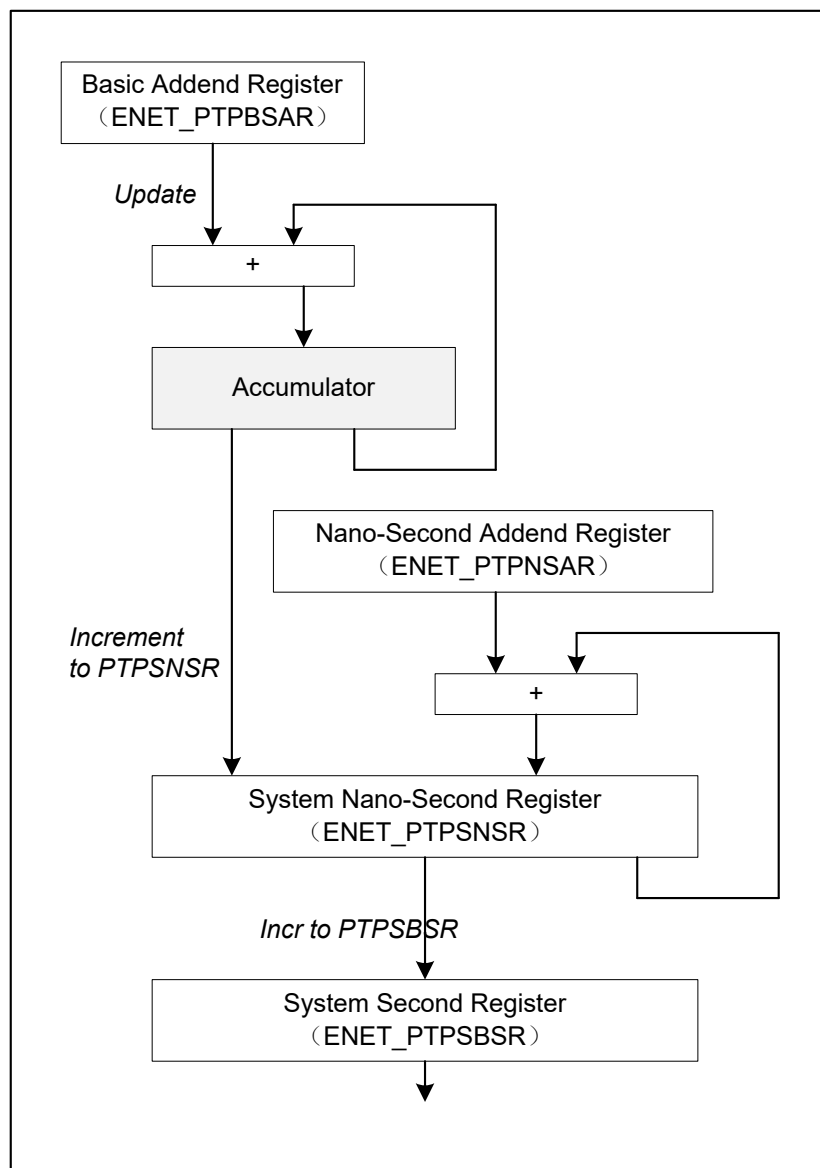


图 35-12 系统时间校准

系统时间更新逻辑需要使用 50MHz 的时钟频率，以达到 20ns 的精度。如果参考时钟频率为 66MHz，则通过计算可知参考时钟频率与所需时钟频率的比率（分频比）为 $66\text{MHz}/50\text{MHz}=1.32$ ，基本

加数寄存器 (ENET_PTPBSAR) 中设置的默认加数值为 $2^{32}/1.32$ ，相当于 0xC1F07C1F；如果参考时钟向下偏移降至 65MHz，则分频比为 $65\text{MHz}/50\text{MHz}=1.3$ ，基本加数寄存器 (ENET_PTPBSAR) 中要设置的值为 $2^{32}/1.3$ ，相当于 0xC4EC4EC4；如果时钟向上偏移升至 67MHz，则必须将基本加数寄存器 (ENET_PTPBSAR) 设置为 0xBF0B7672。当时钟偏移为零时，应编程的默认加法值为 0xC1F07C1F ($2^{32}/1.32$)。

用于亚秒加数寄存器 (ENET_PTPNSAR) 递增的值设定为十进制数 0d43。这可使系统时间的精度达到 20ns（换句话说，增量步长时间为 20ns）。

软件必须根据同步报文对频率偏移进行计算，并相应更新基本加数寄存器 (ENET_PTPBSAR)。首先，使用基本加数寄存器 (ENET_PTPBSAR) 中的 FreqCompensationValue0 设置从时钟。该值的计算公式如下：

$$\text{FreqCompensationValue0} = 2^{32} / \text{FreqDivisionRatio}$$

如果起初假定 MasterToSlaveDelay 对于连续的同步消息是相同的，则必须应用下述算法。数个同步周期之后，频率会锁定。从时钟随后可确定 MasterToSlaveDelay 的精确值，并使用新值重新与主时钟同步。

该算法如下：

- step1. 在 MasterSyncTime (n) 时刻，主时钟向从时钟发送同步报文。从时钟在其本地时钟为 SlaveClockTime (n) 时接收到该报文，并用如下公式计算 MasterClockTime (n)：

$$\text{MasterClockTime (n)} = \text{MasterSyncTime (n)} + \text{MasterToSlaveDelay (n)}$$

- step2. 当前同步周期的主时钟计数 MasterClockCount (n) 的计算公式如下：

$$\text{MasterClockCount (n)} = \text{MasterClockTime (n)} - \text{MasterClockTime (n-1)}$$

（假定 MasterToSlaveDelay 对于同步周期 n 和 n-1 是相同的）

- step3. 当前同步周期的从时钟计数 SlaveClockCount (n) 的计算公式如下：

$$\text{SlaveClockCount (n)} = \text{SlaveClockTime (n)} - \text{SlaveClockTime (n-1)}$$

- step4. 当前同步周期的主从时钟计数差值 SlaveClockCount (n) 的计算公式如下：

$$\text{ClockDiffCount (n)} = \text{MasterClockCount (n)} - \text{SlaveClockCount (n)}$$

- step5. 从时钟的分频系数 FreqScaleFactor (n) 的计算公式如下：

$$\text{FreqScaleFactor (n)} = (\text{MasterClockCount (n)} + \text{ClockDiffCount (n)}) / \text{SlaveClockCount (n)}$$

- step6. 加数寄存器的频率补偿值 FreqCompensationValue (n) 的计算公式如下：

$$\text{FreqCompensationValue (n)} = \text{FreqScaleFactor (n)} \times \text{FreqCompensationValue (n-1)}$$

理论上，该算法可在一个同步周期内实现锁定，但是由于网络传播延迟和工作条件会不断变化，因此可能需要多个周期。该算法可进行自校准，如果出于某些原因导致最初通过主时钟设置的从时钟不正确，则该算法会花费更多同步周期将其校准。

35.3.23 PTP 子控制器初始化

通过将 PTP 控制寄存器中的 PTPEN 位 (ENET_PTPCR.PTPEN) 置 1 来使能时间戳功能。但之后必须对时间戳计数器进行初始化才能启动时间戳操作。设定顺序如下：

- step1. 通过将 MAC 中断屏蔽寄存器中的 PTPIM 位 (ENET_MACIMR.PTPIM) 置 1 以屏蔽时间戳触发中断
- step2. 编程 PTP 控制寄存器的 PTPEN 位 (ENET_PTPCR.PTPEN) 以使能时间戳
- step3. 根据 PTP 时钟频率设定亚秒加数寄存器 (ENET_PTPNSAR)
- step4. 设定基本加数寄存器 (ENET_PTPBSAR) 并将 PTP 控制寄存器的 PTPADUP 位 (ENET_PTPCR.PTPADUP) 置 1
- step5. 轮询确认 PTP 控制寄存器的 PTPADUP 位 (ENET_PTPCR.PTPADUP) 清零
- step6. 选择精密校准方式时，将 PTP 控制寄存器的 PTPCS 位 (ENET_PTPCR.PTPCS) 置 1
- step7. 用适当的时间值编程时间戳更新秒寄存器 (ENET_PTPUBSR) 和时间戳更新亚秒寄存器 (ENET_PTPUNSR)
- step8. 将 PTP 控制寄存器 PTPINI 位 (ENET_PTPCR.PTPINI) 置 1，进行时间戳初始化
- step9. 用更新寄存器中写入的值初始化时间戳计数器后，时间戳计数器便开始运行
- step10. 使能 MAC 接收器和发送器以使时间戳功能正常运行

35.3.24 PTP 子控制器 PPS 输出

PTP 子控制器可以将生成的内部时钟通过 PPS(PPS0)的方式输出到 ENET_PPS_OUT 端口上(连续输出模式)，用于网络系统的所有节点的时钟同步，同时有两个 PPS (PPS0 和 PPS1) 的输出(单次输出模式)，可用于触发外部中断。

在 PPS 连续输出模式时，系统时间发生秒进位(亚秒时间溢出)后，PPS0 通道可生成一系列特定的脉冲或时钟。具体输出脉冲或输出时钟频率的设定由 PPS 控制寄存器的 PPSOUT0 位 (ENET_PTPPSR.PPSOUT0) 控制。只有 PPS0 通道支持连续输出模式。

在 PPS 单次输出模式时，系统时间若到达目标秒/亚秒寄存器 0/1 (ENET_PTPTBSR0/1、ENET_PTPTNSR0/1) 设定的时间后，在 PPS0 或 PPS1 通道上可对应产生一个脉冲。该功能在设定 PPS 控制寄存器的 TTSEL0/1 位 (ENET_PTPPSR.TTSEL0/1) 后使能、在设定 PPS 控制寄存器的 PPSOUT0/1 位 (ENET_PTPPSR.PPSOUT0/1) 被激活。在脉冲产生之后，若想再次产生脉冲输出，需重新设定目标秒/亚秒寄存器 0/1 (ENET_PTPTBSR0/1、ENET_PTPTNSR0/1) 和 PPS 控制寄存器的 PPSOUT0/1 位 (ENET_PTPPSR.PPSOUT0/1)。

35.3.25 中断

下图 35-13 是 ENET 中的中断控制框图。

MAC 控制器将 DMA 的正常事件(ENET_DMASR.NIS)、DMA 的异常事件(ENET_DMASR.AIS)、PMT 事件 (ENET_MACISR.PMTIS)、MMC 事件 (ENET_MACISR.MMC**IS)、PTP 事件 (ENET_MACISR.PTPIS) 等通过中断的使能或屏蔽控制汇总成一个中断事件输出至中断模块，所以系统在中断响应时，需要通过读取相应的状态信息来确定具体的中断事件。

同时，ENET 的 PMT 模块还有一个非屏蔽的中断输出。在 PMT 事件产生时，不管 MAC 中断屏蔽寄存器的 PMTIM 位 (ENET_MACIMR.PMTIM) 是否置 1，都会产生 PMT 非屏蔽中断。

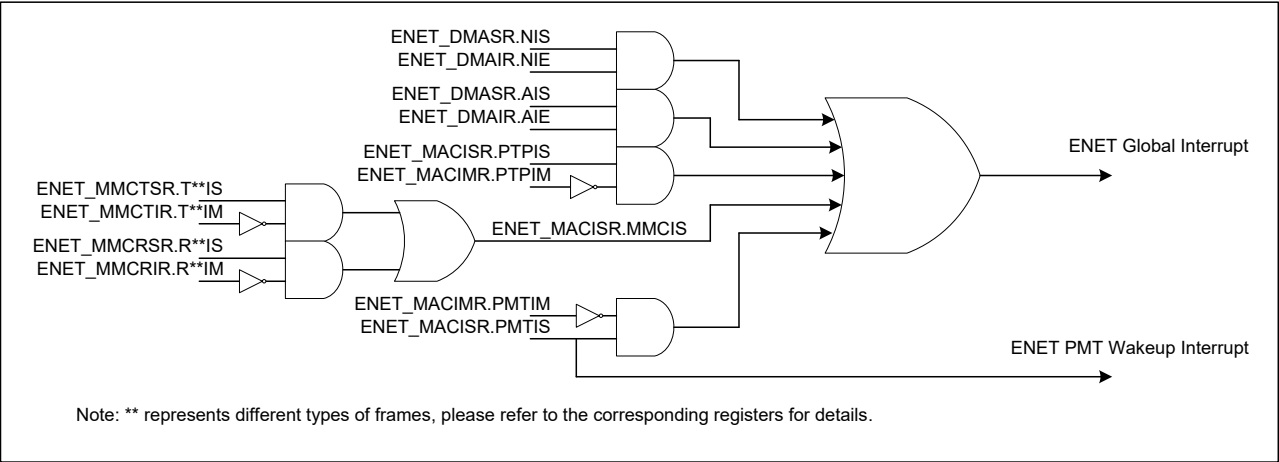


图 35-13 ENET 中断控制框图

35.3.25.1 DMA 子控制器中断

DMA 在发送数据和接收数据过程中产生的各种正常事件会触发普通中断、各种异常事件会触发异常中断。如图 35-14 所示。

各具体事件的中断控制可通过 DMA 中断寄存器 (ENET_DMAIR) 的各个位控制。参见 DMA 中断寄存器章节。

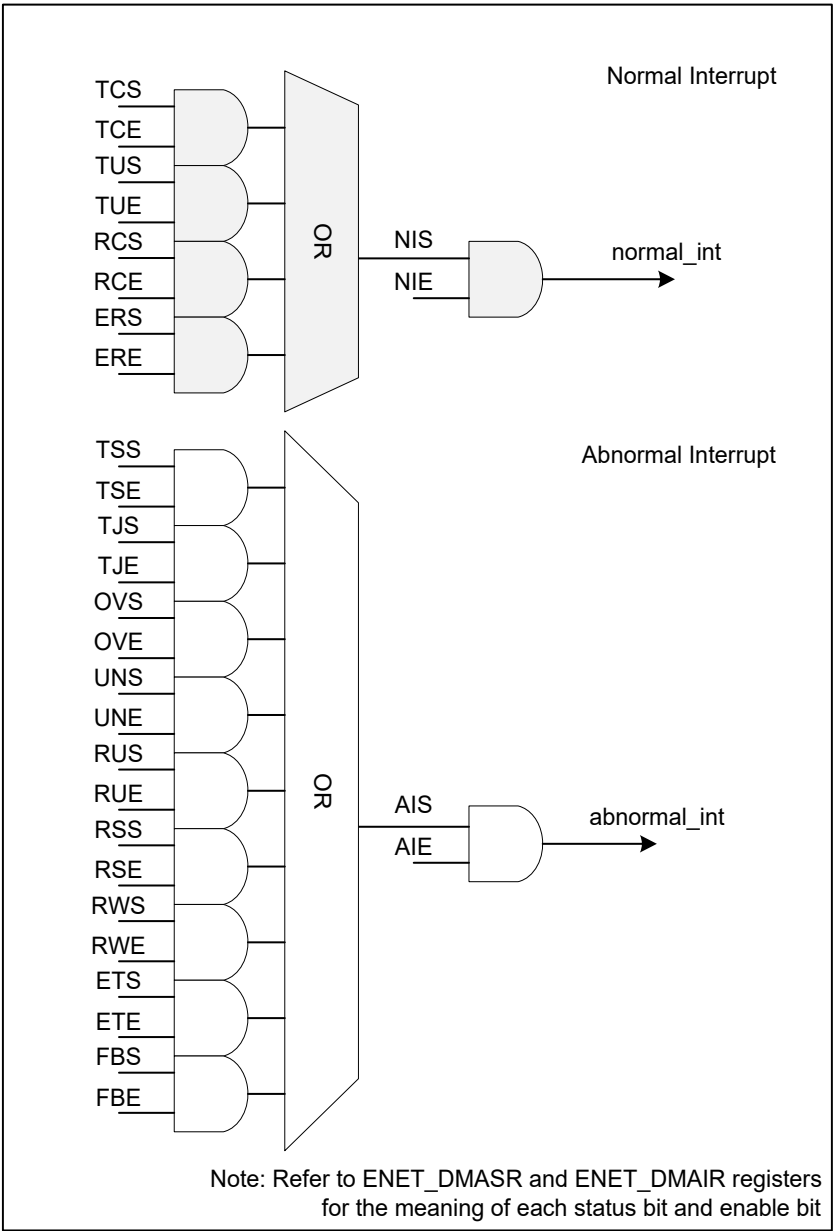


图 35-14 DMA 中断框图

35.3.25.2 PTP 子控制器中断

在 PTP 功能有效时，系统时间到达时间戳目标时间寄存器 0~1 中的设定值时可触发生成 PTP 中断。该中断由 PPS 控制寄存器的 TTSEL0/1 位（ENET_PTPPSR.TTSEL0/1）和 MAC 中断屏蔽寄存器的 PTPIM 位（ENET_MACIMR.PTPIM）控制。

35.3.25.3 PMT 电源管理中断

MAC 在接收到远程唤醒帧和魔术数据包时可触发生成 PMT 中断（可屏蔽中断和非屏蔽中断）。其中可屏蔽中断的生成由 MAC 中断屏蔽寄存器的 PMTIM 位（ENET_MACIMR.PMTIM）控制。

35.3.25.4 MMC 管理计数中断

在 MMC 中各项发送帧统计数和接收帧统计数溢出时可触发生成 MMC 中断。该中断由 MMC 的接收中断寄存器 (ENET_MMCRIR) 和发送中断寄存器 (ENET_MMCTIR) 的各个位控制。参见寄存器章节。

35.4 寄存器

35.4.1 寄存器总览

表 35-7 ENET 寄存器概览

Offset	Acronym	Register Name	Reset
0x0000	ENET_MACCR	MAC 控制寄存器	0x00008000
0x0004	ENET_MACAFR	MAC 地址过滤控制寄存器	0x00000000
0x0008	ENET_MACHTHR	MAC HASH 表高位寄存器	0x00000000
0x000C	ENET_MACHTLR	MAC HASH 表低位寄存器	0x00000000
0x0010	ENET_MACSMIAR	MAC SMI 地址寄存器	0x00000000
0x0014	ENET_MACSMIDR	MAC SMI 数据寄存器	0x00000000
0x0018	ENET_MACFCR	MAC 流控寄存器	0x00000000
0x001C	ENET_MACVLTRR	MAC VLAN TAG 接收寄存器	0x00000000
0x0024	ENET_MACSR	MAC 状态寄存器	0x00000000
0x0028	ENET_PMTRWFR	MAC 远程唤醒帧过滤寄存器	0x00000000
0x002C	ENET_PMTCR	MAC PMT 控制寄存器	0x00000000
0x0038	ENET_MACISR	MAC 中断状态寄存器	0x00000000
0x003C	ENET_MACIMR	MAC 中断屏蔽寄存器	0x00000000
0x0040	ENET_MACAHR0	MAC 过滤地址高位寄存器 0	0x8000FFFF
0x0044	ENET_MACALR0	MAC 过滤地址低位寄存器 0	0xFFFFFFFF
0x0048	ENET_MACAHR1	MAC 过滤地址高位寄存器 1	0x0000FFFF
0x004C	ENET_MACALR1	MAC 过滤地址低位寄存器 1	0xFFFFFFFF
0x0050	ENET_MACAHR2	MAC 过滤地址高位寄存器 2	0x0000FFFF
0x0054	ENET_MACALR2	MAC 过滤地址低位寄存器 2	0xFFFFFFFF
0x0058	ENET_MACAHR3	MAC 过滤地址高位寄存器 3	0x0000FFFF
0x005C	ENET_MACALR3	MAC 过滤地址低位寄存器 3	0xFFFFFFFF
0x0060	ENET_MACAHR4	MAC 过滤地址高位寄存器 4	0x0000FFFF
0x0064	ENET_MACALR4	MAC 过滤地址低位寄存器 4	0xFFFFFFFF
0x0068	ENET_MACAHR5	MAC 过滤地址高位寄存器 5	0x0000FFFF
0x006C	ENET_MACALR5	MAC 过滤地址低位寄存器 5	0xFFFFFFFF
0x0100	ENET_MMCCR	MMC 控制寄存器	0x00000000
0x0104	ENET_MMCRSR	MMC 接收状态寄存器	0x00000000
0x0108	ENET_MMCTSR	MMC 发送状态寄存器	0x00000000
0x010C	ENET_MMCRIR	MMC 接收中断寄存器	0x00000000
0x0110	ENET_MMCTIR	MMC 发送中断寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x014C	ENET_MMCTGSCR	MMC 单个冲突错误后发送好帧统计寄存器	0x00000000
0x0150	ENET_MMCTGMCR	MMC 多个冲突错误后发送好帧统计寄存器	0x00000000
0x0168	ENET_MMCTGR	MMC 发送好帧通统计寄存器	0x00000000
0x0194	ENET_MMCRRCRER	MMC CRC 错误接收帧统计寄存器	0x00000000
0x0198	ENET_MMCRALIER	MMC 对齐错误接收帧统计寄存器	0x00000000
0x01C4	ENET_MMCRUGR	MMC 接收单播好帧统计寄存器	0x00000000
0x0584	ENET_MACVLTR	MAC VLAN TAG 发送寄存器	0x00000000
0x0588	ENET_MACVLHTR	MAC VLAN HASH 表寄存器	0x00000000
0x0700	ENET_PTPCR	PTP 控制寄存器	0x00002000
0x0704	ENET_PTPNSAR	PTP 亚秒加数寄存器	0x00000000
0x0708	ENET_PTPSBSR	PTP 系统秒寄存器	0x00000000
0x070C	ENET_PTPSNSR	PTP 系统亚秒寄存器	0x00000000
0x0710	ENET_PTPUBSR	PTP 更新秒寄存器	0x00000000
0x0714	ENET_PTPUNSR	PTP 更新亚秒寄存器	0x00000000
0x0718	ENET_PTPBSAR	PTP 基本加数寄存器	0x00000000
0x071C	ENET_PTPTBSR0	PTP 目标秒寄存器 0	0x00000000
0x0720	ENET_PTPTNSR0	PTP 目标亚秒寄存器 0	0x00000000
0x0728	ENET_PTPSR	PTP 状态寄存器	0x00000000
0x072C	ENET_PTPPPSR	PTP PPS 控制寄存器	0x00000000
0x0780	ENET_PTPTBSR1	PTP 目标秒寄存器 1	0x00000000
0x0784	ENET_PTPTNSR1	PTP 目标亚秒寄存器 1	0x00000000
0x1000	ENET_DMABSR	DMA 总线控制寄存器	0x00020101
0x1004	ENET_DMATXPDR	DMA 发送轮询寄存器	0x00000000
0x1008	ENET_DMARXPDR	DMA 接收轮询寄存器	0x00000000
0x100C	ENET_DMARXDSAR	DMA 接收描述符地址寄存器	0x00000000
0x1010	ENET_DMATXDSAR	DMA 发送描述符地址寄存器	0x00000000
0x1014	ENET_DMASR	DMA 状态寄存器	0x00000000
0x1018	ENET_DMAMDR	DMA 工作模式寄存器	0x00000000
0x101C	ENET_DMAIR	DMA 中断寄存器	0x00000000
0x1020	ENET_DMAFLCR	DMA 帧丢失统计寄存器	0x00000000
0x1024	ENET_DMAWDTR	DMA 看门狗定时寄存器	0x00000000
0x1048	ENET_DMACURTXDSAR	DMA 当前发送描述符地址寄存器	0x00000000
0x104C	ENET_DMACURRXDSAR	DMA 当前接收描述符地址寄存器	0x00000000
0x1050	ENET_DMACURTXBFAR	DMA 当前发送缓冲区地址寄存器	0x00000000
0x1054	ENET_DMACURRXBFAR	DMA 当前接收缓冲区地址寄存器	0x00000000

35.4.2 MAC 子控制器寄存器

35.4.2.1 ENET_MACCR MAC 控制寄存器

偏移地址：0x0000

复位值：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SAIRC			Res.		CST	Res.	WTD	JAB	Res.		IFG			MCRS
	rw					rw		rw	rw			rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		RDIS	LM	DM	IPCO	RETY	Res.	APCS	BL		DLYC	TE	RE	Res.	
		rw	rw	rw	rw	rw		rw	rw		rw	rw	rw		

Bit	Field	Description
31	Reserved	保留，必须保持复位值
31:28	SAIRC	SA 插入替换控制（SA Insert Replace Control） 010：将 MAC 过滤地址寄存器 0 中的地址值作为 SA 地址插入到发送帧中 110：将 MAC 过滤地址寄存器 1 中的地址值作为 SA 地址插入到发送帧中 011：将 MAC 过滤地址寄存器 0 中的地址值作为 SA 地址替换发送帧中的 SA 字段 111：将 MAC 过滤地址寄存器 1 中的地址值作为 SA 地址替换发送帧中的 SA 字段 x0x：发送帧中 SA 字段地址的处理由 Tx 描述符 TDES1 的 SAIRC 位控制
27:26	Reserved	保留，必须保持复位值
25	CST	CRC 丢弃（CRC Strip） 0：该功能无效 1：在将帧转发到应用之前，去除并丢弃所有 TYPE（类型字段大于 0x0600）帧的最后 4 个字节（FCS 字段）
24	Reserved	保留，必须保持复位值
23	WTD	看门狗无效（Watchdog Disable） 0：MAC 允许接收的帧不超过 2048 字节，并会截断超过此限制接收的任何字节 1：MAC 禁止接收器上的看门狗定时器，并可接收多达 16384 字节的帧
22	JAB	Jabber 定时器（Jabber Count） 0：发送期间应用发送超过 2048 字节的数据，MAC 会截断发送器 1：MAC 禁止发送器上的 Jabber 定时器，并可发送多达 16384 字节的帧
21:20	Reserved	保留，必须保持复位值
19:17	IFG	帧间间隙（Interframe Gap） 发送期间帧间的最小间隙 000：96 位时间 001：88 位时间 010：80 位时间 111：40 位时间 注：在半双工模式下，最小 IFG 仅能配置为 64 位时间（IFG=100），不考虑更低的值

Bit	Field	Description
16	MCRS	屏蔽载波侦听 (Mask Carrier Sense) 0: 半双工模式下帧发送期间, MAC 发送器会生成载波侦听错误, 甚至中止发送 1: 半双工模式下帧发送期间, 不会因在此发送期间载波丢失或无载波而生成错误 注: 该位只在半双工模式时有效, 全双工模式时无效
15:14	Reserved	保留, 必须保持复位值
13	RDIS	接收无效 (Receive Disable) 0: 在半双工模式下, MAC 接收来自 PHY 的所有包 1: 在半双工模式下, MAC 禁止接收帧 注: 该位只在半双工模式时有效, 全双工模式时无效
12	LM	回环模式 (Loopback Mode) 0: Loopback 模式无效 1: Loopback 模式有效
11	DM	双工模式 (Duplex Mode) 0: 半双工模式 1: 全双工模式
10	IPCO	COE 引擎 (IP Checksum Offload) 0: IP Checksum Offload 功能无效 1: IP Checksum Offload 功能有效
9	RETY	重发尝试 (Retry Transfer) 0: 在 MII 模式下发生冲突时, MAC 会尝试根据该寄存器的 BL 位的设置进行重试 1: 在 MII 模式下发生冲突时, MAC 会忽略当前帧发送并以发送帧状态下过度冲突错误报告帧中止, 即 MAC 仅尝试一次发送 注: 该位只在半双工模式时有效, 全双工模式时无效
8	Reserved	保留, 必须保持复位值
7	APCS	自动 CRC/PAD 丢弃 (Automatic CRC/PAD Strip) 0: MAC 不管接收到的帧的长度, 传送所有接收帧到应用 1: MAC 在长度字段值小于 1536 字节时自动去除接收帧上的 PAD/FCS 字段, 在长度字段大于或等于 1536 字节时不去除 PAD/FCS 字段
6:5	BL	后退限制 (Back-off Limit) 后退限制决定发生冲突后重试期间 MAC 在重新安排一次发送之前等待的随机整数 (r) 个时间片延迟 (512 位时间) 00: $k = \min(n, 10)$ 01: $k = \min(n, 8)$ 10: $k = \min(n, 4)$ 11: $k = \min(n, 1)$ 其中 n = 重新发送尝试的次数。随机整数 r 的取值范围为: $0 \leq r < 2^k$ 。 注: 该位只在半双工模式时有效, 全双工模式时无效

Bit	Field	Description
4	DLYC	延迟检查 (Delay Check) 0: 禁止延迟检查功能 1: 使能延迟检查功能。当发送状态机在延迟超过 24288 位时间时, MAC 将指示帧中止状态, 同时在发送帧状态中将过度延迟错误位置 1 注 1: 延迟的产生: 当发送器准备好发送但因载波侦听信号而被阻止时延迟计数就开始。延迟时间是非累积性的, 如果发送器延迟 10000 位时间, 则重新执行一遍发送、冲突、后退, 然后在完成后退后复位延迟计数器为 0 并重新开始计数 注 2: 该位只在半双工模式时有效, 全双工模式时无效
3	TE	发送使能 (Transfer Enable) 0: 发送无效 1: 发送使能
2	RE	接收使能 (Receive Enable) 0: 接收无效 1: 接收使能
1:0	Reserved	保留, 必须保持复位值

35.4.2.2 ENET_MACAFR MAC 地址过滤控制寄存器

偏移地址: 0x0004

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RALL	Res.									DNTU	Res.				VLTF
rw										rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					HPF	SAF	SAIF	PCF		BCF	MCF	DAIF	HM	HU	PM
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	RALL	接收全部 (Receive All) 0: MAC 接收器将通过地址过滤的接收帧传送给应用 1: MAC 接收器将所有的接收帧传送到应用, 不管它们是否已通过地址过滤
30:22	Reserved	保留, 必须保持复位值
21	DNTU	丢弃无 TCP/UDP 帧 (Dropped no-TCP/UDP) 0: MAC 接收器将所有接收的 IP 报传送到应用, 不管它们是否有 TCP/UDP 字段 1: MAC 接收器丢弃所有无 TCP 或 UDP 字段的 IP 报
20:17	Reserved	保留, 必须保持复位值

Bit	Field	Description
16	VLTF	VLAN 标识符过滤 (VLAN TAG Filter) 0: MAC 接收器将所有接收的 VLAN 帧传送到应用, 不管它们是否已通过 VLAN 标识符过滤 1: MAC 接收器丢弃与 VLAN 标识符设定的过滤值比较不匹配的 VLAN 帧
15:11	Reserved	保留, 必须保持复位值
10	HPF	Hash/完全过滤 (Hash/Perfect Filter) 0: 若该寄存器的 HU 位或 HM 位置 1, 则地址过滤器只传送与 Hash 过滤器匹配的帧 1: 若该寄存器的 HU 位或 HM 位置 1, 则地址过滤器传送与地址过滤器或 Hash 过滤器匹配的帧
9	SAF	源地址过滤 (Source Address Filter) 0: MAC 接收器将所有接收的帧传送到应用, 不管它们是否已通过源地址过滤 1: MAC 接收器丢弃与源地址设定的过滤值比较不匹配的帧
8	SAIF	源地址反向过滤 (Source Address Inverse Filter) 0: MAC 接收器接收到的帧 SA 字段与设定的过滤值不匹配时, 认定为过滤失败 1: MAC 接收器接收到的帧 SA 字段与设定的过滤值匹配时, 认定为过滤失败
7:6	PCF	暂停控制帧过滤 (Pause Control Frame Filter) 对控制帧 (包括单播暂停帧和多播暂停帧) 的转发做如下操作: 00: MAC 阻止所有控制帧到达应用 01: MAC 将除了暂停控制帧以外的所有控制帧转发到应用, 即使这些帧的地址过滤失败 10: MAC 将所有控制帧转发到应用, 即使这些帧的地址过滤失败 11: MAC 转发通过地址过滤的控制帧 关于暂停控制帧的几点说明: 注 1: 在全双工模式时, 暂停控制帧的处理由流控寄存器的 FRE 位 (ENET_MACFCR.FRE) 决定 注 2: 在流控寄存器的 UPF 位 (ENET_MACFCR.UPF) 置 1 时, 接收到的帧的目标地址和 MAC 过滤地址寄存器 0 (ENET_MACAH/LR0) 设定的值匹配时, 该帧被认为是单播暂停帧 注 3: 接收到的帧的 TYPE 段值为 0x8808 或 OPCODE 段值为 0x0001 时, 该帧被认为是暂停控制帧
5	BCF	广播帧过滤 (Broadcast Frame Filter) 0: 地址过滤器不对接收到的广播帧进行过滤 1: 地址过滤器对所有传入的广播帧进行过滤 注: 该位置 1 时, 其优先级高于其它过滤设定
4	MCF	多播帧过滤 (Multicast Frame Filter) 0: 对接收到的带多播目标地址 (目标地址字段的第一位是 1) 的帧进行地址过滤, 过滤方式由该寄存器的 HM 位决定 1: 对接收到的带多播目标地址 (目标地址字段的第一位是 1) 的帧不进行地址过滤

Bit	Field	Description
3	DAIF	目标地址反向过滤（Destination Address Inverse Filter） 0: MAC 接收器接收到的单播帧或多播帧的目标地址与设定的过滤值不匹配时，认定为过滤失败 1: MAC 接收器接收到的单播帧或多播帧的目标地址与设定的过滤值匹配时，认定为过滤失败
2	HM	Hash 多播（Hash Multicast） 0: MAC 对多播帧执行完全目标地址过滤，即将 DA 字段与设定的过滤值进行比较 1: MAC 对多播帧执行 Hash 目标地址过滤
1	HU	Hash 单播（Hash Unicast） 0: MAC 对单播帧执行完全目标地址过滤，即将 DA 字段与设定的过滤值进行比较 1: MAC 对单播帧执行 Hash 目标地址过滤
0	PM	直通模式（Pass Mode） 0: 地址过滤器进行正常的地址过滤 1: 地址过滤器不管目标或源地址，都传送所有传入的帧 注：该位与本寄存器 RALL 位的区别是，该位置 1 时不会将地址过滤的状态结果反应进接收描述符；RALL 位置 1 时会将地址过滤的状态结果反应进接收描述符

35.4.2.3 ENET_MACHTHR MAC HASH 表高位寄存器

偏移地址：0x0008

复位值：0x0000 0000

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

Bit	Field	Description
31:0	MACHTABH	MAC Hash 地址过滤 Hash 表高位（MAC Hash Address Filter Hash Table High） Hash Table 的高 32 位值

35.4.2.4 ENET_MACHTLR MAC HASH 表低位寄存器

偏移地址：0x000C

复位值：0x0000 0000

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

Bit	Field	Description
31:0	MACHTABL	MAC Hash 地址过滤 Hash 表低位 (MAC Hash Address Filter Hash Table Low) Hash Table 的低 32 位值

35.4.2.5 ENET_MACSMIAR SMI 地址寄存器

偏移地址: 0x0010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMIADD					SMIREG					SMICLK				SMIWR	SMIBY
rw					rw					rw				rw	w1c

Bit	Field	Description															
31:16	Reserved	保留, 必须保持复位值															
15:11	SMIADD	SMI 地址 (SMI Address) 指示要访问 32 个可能的 PHY 器件中的哪一个															
10:6	SMIREG	SMI 寄存器 (SMI Reg) 指示在所选 PHY 器件中要选择寄存器															
5:2	SMICLK	SMI 时钟 (SMI Clock) 基于系统频率决定 MDC 时钟频率。 <table border="1"> <thead> <tr> <th>选项</th><th>系统频率</th><th>MDC 时钟</th></tr> </thead> <tbody> <tr> <td>0000</td><td>60-100MHz</td><td>系统频率/42</td></tr> <tr> <td>0001</td><td>100-120MHz</td><td>系统频率/62</td></tr> <tr> <td>0010</td><td>20-35MHz</td><td>系统频率/16</td></tr> <tr> <td>0011</td><td>35-60MHz</td><td>系统频率/26</td></tr> </tbody> </table> 其它值设定禁止	选项	系统频率	MDC 时钟	0000	60-100MHz	系统频率/42	0001	100-120MHz	系统频率/62	0010	20-35MHz	系统频率/16	0011	35-60MHz	系统频率/26
选项	系统频率	MDC 时钟															
0000	60-100MHz	系统频率/42															
0001	100-120MHz	系统频率/62															
0010	20-35MHz	系统频率/16															
0011	35-60MHz	系统频率/26															
1	SMIWR	SMI 读写 (SMI Write & Read) 0: SMI 读操作 1: SMI 写操作															

Bit	Field	Description
0	SMIBY	SMI 忙碌状态 (SMI Busy) 向 SMI 地址寄存器 (ENET_MACSMIAR) 和 SMI 数据寄存器 (ENET_MACSMIDR) 写入前, 此位应读取逻辑 0; 向 SMI 地址寄存器 (ENET_MACSMIAR) 写入过程中, 此位也必须复位为 0; 在 PHY 寄存器访问过程中, 此位由应用程序设为 1, 指示读或写访问正在进行中; 在对 PHY 进行写操作过程中, SMI 数据寄存器 (ENET_MACSMIDR) 应始终有效, 直到 MAC 将此位清零; 在对 PHY 进行读操作过程中, SMI 数据寄存器 (ENET_MACSMIDR) 始终无效, 直到 MAC 将此位清零; 此位清零后, 才可以向 SMI 数据寄存器 (ENET_MACSMIDR) 写入值

35.4.2.6 ENET_MACSMIDR SMI 数据寄存器

偏移地址: 0x0014

复位值: 0x0000 0000

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

Bit	Field	Description
31:16	Reserved	保留, 必须保持复位值
15:0	SMIDATA	SMI 数据 (SMI Data) 站点管理读操作之后从 PHY 中读取的 16 位数据值, 或站点管理写操作之前要写入 PHY 的 16 位数据值

35.4.2.7 ENET_MACFCR MAC 流控寄存器

偏移地址: 0x0018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSET															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								DZQP	Res.	PLT		UPF	FRE	FTE	FCBBP A
								rw		rw		rw	rw	rw	rw

Bit	Field	Description
31:16	PSET	暂停时间设定 (Pause Time Setting) 该位表示发送控制帧中暂停时间字段要使用的值
15:8	Reserved	保留, 必须保持复位值
7	DZQP	禁止生成零时间片暂停控制帧 (Disable Zero-Quanta Pause Frame) 0: 当流控制信号有效时, 正常生成零时间片暂停控制帧 1: 当流控制信号有效时, 禁止生成零时间片暂停控制帧
6	Reserved	保留, 必须保持复位值
5:4	PLT	暂停低阈值 (Pause Low Threshold) 配置暂停定时器的阈值, 达到该值时会自动重新传输 PAUSE 帧 00: 暂停时间减去 4 个时隙 01: 暂停时间减去 28 个时隙 10: 暂停时间减去 144 个时隙 11: 暂停时间减去 256 个时隙 注 1: 时隙定义为 MII 接口每发送 512 位 (64 字节) 所需的时间 注 2: 该阈值应始终小于暂停时间。例如, 如果暂停时间 PSET=100H (256 个时隙), 而 PLT=01, 则在第一个 PAUSE 帧发送完成后, 第二个 PAUSE 帧会在第 228 (256-28) 个时隙启动时自动发送
3	UPF	单播暂停帧 (Unicast Pause Frame) 0: MAC 仅检测具有 802.3x 标准中指定的唯一多播地址的暂停帧 1: MAC 除了检测具有唯一多播地址的暂停帧外, 还会检测具有 MAC 过滤地址寄存器 0 (ENET_MACAH/LR0) 所指定的站单播地址的暂停帧
2	FRE	流控接收使能 (Flow Control Receive Enable) 0: 禁止暂停帧的解码功能 1: MAC 对接收到的暂停帧进行解码, 并禁止其在指定的暂停时间内发送
1	FTE	流控发送使能 (Flow Control Transfer Enable) 全双工模式时: 0: 禁止 MAC 传送任何暂停帧 1: 使能流控制操作, MAC 发送暂停帧 半双工模式时: 0: 禁止背压功能 1: 使能背压操作
0	FCBBPA	流控忙碌/背压有效 (Flow Control Busy/Back Pressure Active) 向流控制寄存器写入数据前此位应读为 0 全双工模式时: 此位置 1 时 MAC 子控制器启动暂停控制帧。在控制帧发送过程中, 此位保持置 1 以指示帧发送正在进行中。当暂停控制帧发送完成后, MAC 会将此位复位为 0。此位清零后, 才可以对流控制寄存器执行写操作 在半双工模式时: 此位置 1 时, MAC 子控制器将置位背压功能。在背压操作期间, 当 MAC 接收到新帧时, 发送器会开始发送一个导致冲突的 Jam 信号

35.4.2.8 ENET_MACVLTRR VLAN TAG 接收寄存器

偏移地址: 0x001C

复位值: 0x0000 0000

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

Bit	Field	Description
31:20	Reserved	保留, 必须保持复位值
19	VTHM	VLAN 标识符 Hash 过滤模式 (VLAN TAG Hash Mode) 0: VLAN 标识符 Hash 过滤功能无效 1: VLAN 标识符 Hash 过滤功能使能
18	Reserved	保留, 必须保持复位值
17	VTIM	VLAN 标识符反转模式 (VLAN TAG Inverse Mode) 0: VLAN 帧的标识符字段与该寄存器 VLFLT 位设定的过滤值不匹配时, 认定为过滤失败 1: VLAN 帧的标识符字段与该寄存器 VLFLT 位设定的过滤值匹配时, 认定为过滤失败
16	VTSEL	VLAN 标识符过滤选择 (VLAN TAG Filter Select) 0: VLAN 标识符字段的全 16 位与 VLFLT 设定值比较 1: VLAN 标识符字段的低 12 位与 VLFLT 设定值比较 注: 该位的设定也适用于 Hash 过滤方式
15:0	VLFLT	VLAN 标识符过滤 Tag 值 (VLAN TAG Filter Tag Value) 此字段包含用于过滤 VLAN 帧的标识符字段的比较值, 该位与接收的 VLAN 帧的第 15 和第 16 字节进行比较

35.4.2.9 ENET_MACSR MAC 状态寄存器

偏移地址: 0x0024

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						TXFF	TXFNE	Res.	TXFWA	TXFRS		MTP	MTS		MTE
						r	r		r	r		r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						RXFL		Res.	RXFRS		RXFWA	Res.	MRS		MRE
						r			r		r		r		r

Bit	Field	Description
31:26	Reserved	保留，必须保持复位值
25	TXFF	TxFIFO 满 (Tx FIFO Full) 0: TxFIFO 未满 1: TxFIFO 已满，无法再接收发送的帧
24	TXFNE	TxFIFO 空 (Tx FIFO Empty) 0: TxFIFO 空 1: TxFIFO 非空，还有未发送的帧
23	Reserved	保留，必须保持复位值
22	TXFWA	TxFIFO 写有效 (Tx FIFO Write Active) 0: TxFIFO 写无效 1: TxFIFO 写控制器有效且正在将数据传输到 TxFIFO
21:20	TXFRS	TxFIFO 读状态 (Tx FIFO Read Status) 00: 空闲态 01: 读状态 (数据从 TxFIFO 往 MAC 发送器传送) 10: 等待来自 MAC 发送器的 Tx Status 11: 写入收到的 Tx Status 或刷新 TxFIFO
19	MTP	MAC 发送器暂停 (MAC Transmitter in Pause) 0: MAC 发送器处于发送数据状态 1: MAC 发送器处于暂停状态 (全双工模式时)，不安排任何帧发送
18:17	MTS	MAC 发送状态 (MAC Transfer Status) 00: 空闲态 01: 等待前一个帧的状态或 IFG 回退阶段结束 10: 生成并发送暂停控制帧 (在全双工模式下) 11: 传输要发送的输入帧
16	MTE	MAC 内部 MII 发送引擎状态 (MAC MII Transfer Engine Status) 0: MAC 内部 MII 发送引擎处于空闲态 1: MAC 内部 MII 发送引擎正在主动发送数据而未处于空闲状态
15:10	Reserved	保留，必须保持复位值
9:8	RXFL	RxFIFO 填充级别 (Rx FIFO Fill Level) 00: RxFIFO 为空 01: RxFIFO 填充级别低于流控制，取消激活阈值 10: RxFIFO 填充级别高于流控制，激活阈值 11: RxFIFO 已满
7	Reserved	保留，必须保持复位值
6:5	RXFRS	RxFIFO 读状态 (Rx FIFO Read Status) 00: 空闲态 01: 读取帧数据 10: 读取帧状态或时间戳 11: 刷新帧数据和状态
4	RXFWA	RxFIFO 写有效 (Rx FIFO Write Active) 0: RxFIFO 写无效 1: RxFIFO 写有效，并正在将收到的帧传输到 RxFIFO
3	Reserved	保留，必须保持复位值

Bit	Field	Description
2:1	MRS	MAC 接收状态 (MAC Receive Status) 00: 空闲 01: MAC 接收器中的 FIFO 写控制器在工作 10: MAC 接收器中的 FIFO 读控制器在工作 11: MAC 接收器中的 FIFO 读和写控制器在工作
0	MRE	MAC 内部 MII 接收引擎状态 (MAC MII Receive Engine Status) 0: MAC 内部 MII 接收引擎处于空闲态 1: MAC 内部 MII 接收引擎正在主动接收数据而未处于空闲状态

35.4.2.10 ENET_MACISR MAC 中断状态寄存器

偏移地址: 0x0038

复位值: 0x0000 0000

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

Bit	Field	Description
31:10	Reserved	保留, 必须保持复位值
9	PTPIS	PTP 中断状态 (PTP Interrupt Status) 当系统时间值等于或大于目标时间寄存器中指定的值时, 或者系统时间溢出时, 此位置 1 注: 该位在读取 PTP 寄存器的 TSOVF 位 (ENET_PTPSR.TSOVF) 时清零
8:7	Reserved	保留, 必须保持复位值
6	MMCTXIS	MMC 发送中断状态 (MMC Transfer Interrupt Status) MMC 发送中断寄存器 (ENET_MMCTIR) 中有中断生成时, 此位置 1 注: 该位在 MMC 发送中断寄存器 (ENET_MMCTIR) 中所述所有中断清零时清零
5	MMCRXIS	MMC 接收中断状态 (MMC Receive Interrupt Status) MMC 接收中断寄存器 (ENET_MMCRIR) 中有中断生成时, 此位置 1 注: 该位在 MMC 接收中断寄存器 (ENET_MMCRIR) 中所述所有中断清零时清零
4	MMCIS	MMC 中断状态 (MMC Interrupt Status) 该寄存器的 MMCTXIS 或 MMCRXIS 位有高电平, 此位就变为高电平 注: 该位在该寄存器的 MMCTXIS 和 MMCRXIS 均变为 0 时清零

Bit	Field	Description
3	PMTIS	PMT 中断状态 (PMT Interrupt Status) 在低功耗模式下接收到魔术数据包或远程唤醒帧, 此位就变为高电平 注: 该位在 PMT 控制寄存器的 WKFR 和 MPFR 位 (ENET_PMTCR.WKFR/ENET_PMTCR.MPFR) 均变为 0 时清零
2:0	Reserved	保留, 必须保持复位值

35.4.2.11 ENET_MACIMR MAC 中断屏蔽寄存器

偏移地址: 0x003C

复位值: 0x0000 0000

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

Bit	Field	Description
31:10	Reserved	保留, 必须保持复位值
9	PTPIM	PTP 中断屏蔽 (PTP Interrupt Mask) 0: PTP 时间戳中断不屏蔽 1: PTP 时间戳中断屏蔽
8:4	Reserved	保留, 必须保持复位值
3	PMTIM	PMT 中断屏蔽 (PMT Interrupt Mask) 0: PMT 中断不屏蔽 1: PMT 中断屏蔽
2:0	Reserved	保留, 必须保持复位值

35.4.2.12 ENET_MACAHRx MAC 过滤地址高位寄存器 (x=0~5)

偏移地址: 0x0040/0x0048/0x0050/0x0058/0x0060/0x0068

复位值: 0x8000 FFFF (HR0)、0x0000 FFFF (HR1~5)

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

Bit	Field	Description
31	ADDE	地址过滤使能 (Address Filter Enable) 0: 忽略 MAC 地址 x (x=0~5) 地址过滤 1: 使用 MAC 地址 x (x=0~5) 进行地址过滤 注: 当 x=0 时, 该位固定为 1, 为只读位
30	SELE	过滤目标选择 (SA/DA Filter Select Enable) 0: 用 MAC 地址 x (x=1~5) 对接收帧的 DA 进行过滤 1: 用 MAC 地址 x (x=1~5) 对接收帧的 SA 进行过滤 注: 该位只在 x=1~5 时可设定, x=0 时固定为 0
29:24	MBYTEC	屏蔽字节控制 (Mask Byte Control) 地址过滤时屏蔽对应的字节 b29: 屏蔽 MACADDH[15:8] 位 b28: 屏蔽 MACADDH[7:0] 位 b27: 屏蔽 MACADDL[31:24] 位 b26: 屏蔽 MACADDL[23:16] 位 b25: 屏蔽 MACADDL[15:8] 位 b24: 屏蔽 MACADDL[7:0] 位 注: 该位只在 x=1~5 时可设定, x=0 时固定为 0
23:16	Reserved	保留, 必须保持复位值
15:0	MACADDH	MAC 地址过滤地址值高位 (MAC Address Filter Address Value High) 当 x=0 时, MAC 使用此字段过滤所接收的帧以及将 MAC 地址插入到发送流控制 (暂停) 帧中, 当 x=1~5 时, MAC 使用此字段过滤所接收的帧

35.4.2.13 ENET_MACALRx MAC 过滤地址低位寄存器 (x=0~5)

偏移地址: 0x0044/0x004C/0x0054/0x005C/0x0064/0x006C

复位值: 0xFFFF FFFF

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

Bit	Field	Description
31:0	MACADDL	MAC 地址过滤地址值低位 (MAC Address Filter Address Value Low) 当 x=0 时, MAC 使用此字段过滤所接收的帧以及将 MAC 地址插入到发送流控制 (暂停) 帧中, 当 x=1~5 时, MAC 使用此字段过滤所接收的帧

35.4.2.14 ENET_MACVLTTR VLAN TAG 发送寄存器

偏移地址: 0x0584

复位值: 0x0000 0000

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

Bit	Field	Description
31:19	Reserved	保留, 必须保持复位值
18	VLANS	VLAN 选择 (VLAN Select) 0: 在发送 VLAN 帧时, 对标记字段的处理由 Tx 描述符的 TDES0 的 VLANC 位设定决定 1: 在发送 VLAN 帧时, 对标记字段的处理由该寄存器的 VLANC 位设定决定
17:16	VLANC	VLAN 控制 (VLAN Control) 在发送 VLAN 帧时, 对标记字段的处理: 00: 不做操作, 直接发送 01: 将 VLAN 帧的 TYPE 字段和标记字段删除 10: 在插入 TYPE 字段之后把该寄存器的 VLANV 位的值插入作为标记字段, 不管原帧中是否含有标记字段 11: 用该寄存器的 VLANV 位的值替换原帧中的标记字段
15:0	VLANV	VLAN 值 (VLAN Value) 用户定义 VLAN 标记字段值, 用于在发送帧时插入或替换标记字段

35.4.2.15 ENET_MACVLHTR VLAN HASH 表寄存器

偏移地址: 0x0588

复位值: 0x0000 0000

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

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

Bit	Field	Description
15:0	VLHTAB	VLAN Hash 表值 (VLAN Hash Table Value) VLAN Hash Table 值

35.4.3 DMA 子控制器寄存器

35.4.3.1 ENET_DMABSR 总线控制寄存器

偏移地址: 0x1000

复位值: 0x0002 0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				TXPR	MBST	AAL	MPBL	SPBL	RPBL						FBST
				rw	rw	rw	rw	rw	rw						rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTPR		TPBL						DSEN	DSL				DMAA	SR	
Rw		rw						rw	rw				rw	rw	

Bit	Field	Description
31:28	Reserved	保留, 必须保持复位值
27	TXPR	发送优先级 (Transfer Priority) 0: RxDMA 事务优先级高于 TxDMA 事务 1: TxDMA 事务优先级高于 RxDMA 事务
26	MBST	混合突发 (Mixed Burst) 该位置 1 且 FBST 位为 0 时, AHB Master 口将以 INCR 启动所有长度大于 16 的突发传输, 以 SINGLE 和 INCRx 启动长度等于或小于 16 的突发传输 (x=4/8/16)
25	AAL	地址对齐 (Address Align) 该位置 1 时: 1) 若 FBST=1 时, AHB 接口会生成与起始地址 LS 位对齐的所有突发 2) 若 FBST=0 时, 则第一个突发 (访问数据缓冲器的起始地址) 不对齐, 但后续的突发与地址对齐
24	MPBL	多倍 PBL 模式 (Multiple PBL Mode) 该位置 1 时, 会将编程的 PBL 值 (RPBL 位和 TPBL 位) 乘以 8 倍, 因此, DMA 根据 PBL 值以最大 8、16、32、64、128 和 256 个节拍传输数据
23	SPBL	独立 PBL 模式 (Separate PBL Mode) 0: 该寄存器的 TPBL 设定用于 TxDMA 事务和 RxDMA 事务 1: 该寄存器的 TPBL 设定只用于 TxDMA 事务, RPBL 设定用于 RxDMA 事务
22:17	RPBL	接收可编程突发长度 (Receive Programmable Burst Length) 和该寄存器的 TPBL 位的意义相同, 仅用于 RxDMA 事务 注: 该位只在该寄存器的 SPBL 位有效时才有用

Bit	Field	Description
16	FBST	固定突发 (Fixed Burst) 0: DMA Master 接口只使用 SINGLE 和 INCR 访问类型 1: DMA Master 接口使用 SINGLE 和 INCR4、INCR8、INCR16 访问类型
15:14	FTPR	固定传输优先级比 (Fixed Transfer Priority Ratio) Rx 和 Tx 对 BUS 的抢占优先级比 00: 1: 1 11: 2: 1 10: 3: 1 11: 4: 1 注 1: 该位只在该寄存器的 DMAA 位为 0 时有效 注 2: Tx:Rx 或 Rx:Tx 的选择由该寄存器的 TXPR 位决定
13:8	TPBL	发送可编程突发长度 (Transfer Programmable Burst Length) 该位指示要在一个 TxDMA 事务或 RxDMA 事务中传输的最大节拍数, 这是在单个块读/写操作中使用的最大值。DMA 每次在主机总线上开始突发传输时, 始终尝试按 TPBL 中指定的方式进行突发。允许使用值 1、2、4、8、16 和 32 对 TPBL 进行编程, 任何其它值都会产生未定义的行为 注: TPBL 值有以下限制: 1) 可能的最大节拍数 (TPBL) 受 TxFIFO 和 RxFIFO 大小的限制 2) 请不要编程超出范围的 TPBL 值
7	DSEN	增强型描述符使能 (Advanced Descriptor Enable) 0: 常规型描述符 1: 增强型描述符 注: 在 PTP 功能有效 (通过置位 PTP 控制寄存器的 PTPEN 位 (ENET_PTPCR.PTPEN) 来使能) 或校验和减荷功能有效 (通过置位 MAC 控制寄存器的 IPCO 位 (ENET_MACCCR.IPCO) 来使能) 时, 必须使用增强型描述符
6:2	DSL	描述符跳转长度 (Descriptor Skip Length) 该位指定两个不以链式结构连接的描述符之间跳过的 Word 数, 地址从当前描述符结束处开始跳到下一个描述符起始处。当 DSL 值等于零时, 在环形模式下, DMA 会将描述符表视为连续的
1	DMAA	DMA 仲裁 (DMA Arbiter) 0: 循环优先级, 仲裁方式由该寄存器的 FTPR 位决定 1: 固定优先级, 优先方式由该寄存器的 TXPR 位决定
0	SR	软件复位 (Software Reset) 该位置 1 时, DMA 控制器复位所有的 MAC 子系统的内部寄存器和逻辑。在所有内核时钟域完成复位操作后, 该位自动清零。在重新配置 MAC 寄存器时, 确保该位为 0。

35.4.3.2 ENET_DMATXPDR 发送轮询寄存器

偏移地址: 0x1004

复位值: 0x0000 0000

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

Bit	Field	Description
31:0	TXPD	发送轮询（Transfer Poll Demand） 向该位写入任何值时，DMA 都会读取 DMA 发送描述符地址寄存器（ENET_DMATXDSAR）指向的当前描述符，如果该描述符不可用（由主机所有），则 TxDMA 会返回到挂起状态，并将 DMA 状态寄存器的 TUS 位（ENET_DMASR.TUS）进行置位；如果该描述符可用，则发送会继续进行。 使用该寄存器来指示 TxDMA 轮询发送描述符列表，检查当前描述符是否为 DMA 所有。如果 TxDMA 处于挂起模式，则发出发送轮询要求命令将其唤醒；如果发送帧中出现下溢错误或 TxDMA 所拥有的描述符不可用，则 TxDMA 会进入挂起模式 用户可以随时发出此命令，当该命令开始重新获取主机存储器的当前描述符后，TxDMA 即会对其进行复位 注：该位在读出时始终读出零

35.4.3.3 ENET_DMARXPDR 接收轮询寄存器

偏移地址：0x1008

复位值：0x0000 0000

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

Bit	Field	Description
31:0	RXPD	接收轮询（Receive Poll Demand） 向该位写入任何值时，DMA 都会读取 DMA 接收描述符地址寄存器（ENET_DMARXDSAR）指向的当前描述符，如果该描述符不可用（由主机所有），则 RxDMA 会返回到挂起状态，并将 DMA 状态寄存器的 RUS 位（ENET_DMASR.RUS）进行置位；如果该描述符可用，则 RxDMA 将返回到活动状态 使用该寄存器来指示 RxDMA 轮询接收描述符列表，检查新描述符，用于将 RxDMA 从挂起状态唤醒，仅当 RxDMA 所拥有的描述符不可用时，它才会进入挂起状态 注：该位在读出时始终读出零

35.4.3.4 ENET_DMARXDSAR 接收描述符地址寄存器

偏移地址：0x100C

复位值：0x0000 0000

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

Bit	Field	Description
31:0	RXDSA	接收描述符地址（Receive Descriptor Address） 该位表示接收描述符列表中的首个描述符的地址 注：最低 2 位在读出时始终读出零

35.4.3.5 ENET_DMATXDSAR 发送描述符地址寄存器

偏移地址：0x1010

复位值：0x0000 0000

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

Bit	Field	Description
31:0	TXDSA	发送描述符地址（Transfer Descriptor Address） 该位表示发送描述符列表中的首个描述符的地址 注：最低 2 位在读出时始终读出零

35.4.3.6 ENET_DMASR DMA 状态寄存器

偏移地址：0x1014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		PTPS	PMTS	MMCS	Res.	EBUS			TPS			RPS			NIS
		r	r	r		r			r			r			w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	ERS	FBS	Res.		ETS	RWS	RSS	RUS	RCS	UNS	OVS	TJS	TUS	TSS	TCS
w1c	w1c	w1c			w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29	PTPS	PTP 中断状态（PTP Interrupt Status） 0：PTP 模块无时间戳事件产生 1：PTP 模块有时间戳事件发生 注：置位该位的事件状态清零时该位自动清零
28	PMTS	PMT 中断状态（PMT Interrupt Status） 0：PMT 模块无唤醒事件发生 1：PMT 模块有唤醒事件发生 注：置位该位的事件状态清零时该位自动清零
27	MMCS	MMC 中断状态（MMC Interrupt Status） 0：MMC 模块无帧统计事件发生 1：MMC 模块有帧统计事件发生 注：置位该位的事件状态清零时该位自动清零
26	Reserved	保留，必须保持复位值
25:23	EBUS	错误总线状态（Error Bus State） 该位指示导致总线错误（AHB 接口上的错误响应）的错误类型，不会产生中断。 000：RxDMA 写数据时发生错误 011：TxDMA 读数据时发生错误 100：RxDMA 写描述符时发生错误 101：TxDMA 写描述符时发生错误 110：RxDMA 读描述符时发生错误 111：TxDMA 读描述符时发生错误 注：该位仅在该寄存器的 FBS 位置 1 时有效

Bit	Field	Description
22:20	TPS	发送过程状态（Transfer Process State） 该位指示 TxDMA FSM 的状态，不会产生中断 000: 停止，发出复位或停止发送命令 001: 运行中，正在获取发送传输描述符 010: 运行中，正在等待状态 011: 运行中，正在读取主机存储器缓冲区中的数据并将其加入发送缓冲区（TxFIFO）队列 100: 时间戳写入 101: Reserved 110: 挂起，发送描述符不可用或发送缓冲区下溢 111: 运行中，正在关闭发送描述符
19:17	RPS	接收过程状态（Receive Process State） 该位指示 RxDMA FSM 的状态，不会产生中断 000: 停止，发出复位或停止接收命令 001: 运行中，正在获取接收传输描述符 010: Reserved 011: 运行中，正在等待接收数据包 100: 挂起，接收描述符不可用 101: 运行中，正在关闭接收描述符 110: 时间戳写入 111: 运行中，将接收数据包的数据从接收缓冲区传输到主机存储器
16	NIS	普通中断状态（Normal Interrupt Status） 下列任何一个条件满足，该位就被置 1： 1) TCS=1 && ENET_DMAIR.TCE=1 2) TUS=1 && ENET_DMAIR.TUE=1 3) RCS=1 && ENET_DMAIR.RCE=1 4) ERS=1 && ENET_DMAIR.ERE=1 注 1: 只有未屏蔽的中断使能位会影响该普通中断状态汇总位 注 2: 每当导致该位置 1 的对应位被清零时，该位也必须清零（通过向此位写入 1）

Bit	Field	Description
15	AIS	异常中断状态（Abnormal Interrupt Status） 下列任何一个条件满足，该位就被置 1： 1) TSS=1 && ENET_DMAIR.TSE=1 2) TJS=1 && ENET_DMAIR.TJE=1 3) OVS=1 && ENET_DMAIR.OVE=1 4) UNS=1 && ENET_DMAIR.UNE=1 5) RUS=1 && ENET_DMAIR.RUE=1 6) RSS=1 && ENET_DMAIR.RSE=1 7) RWS=1 && ENET_DMAIR.RWE=1 8) ETS=1 && ENET_DMAIR.ETE=1 9) FBS=1 && ENET_DMAIR.FBE=1 注 1：只有未屏蔽的中断使能位会影响该异常中断状态汇总位 注 2：每当导致该位置 1 的对应位被清零时，该位也必须清零（通过向此位写入 1）
14	ERS	提前接收状态（Early Receive Status） 该位指示 DMA 已填满数据包的首个数据缓冲区 注：可通过向该位写入 1 来清零该位，也可在该寄存器的 RCS 为置位后自动清零
13	FBS	致命总线错误状态（Fatal Bus Error Status） 该位指示发生了总线错误，具体错误类型参见该寄存器的 EBUS 位，当该位置 1 后，对应的 DMA 引擎会禁止其所有的总线访问
12:11	Reserved	保留，必须保持复位值
10	ETS	提前发送状态（Early Transfer Status） 该位指示要发送的帧已完全传输到 TxFIFO
9	RWS	接收看门狗溢出状态（Receive Watchdog Timeout Status） 当接收到的帧的长度大于 2048 个字节时，该位置 1
8	RSS	接收停止状态（Receive Stop Status） 当接收过程进入停止状态时，该位置 1
7	RUS	接收缓冲区不可用状态（RxFIFO Unavailable Status） 该位指示接收列表中的下一个描述符由主机所拥有，DMA 无法获取，接收过程进入挂起状态 要恢复处理接收描述符，主机应更改描述符的拥有关系，然后发出接收轮询要求命令。如果未发出接收轮询要求命令，则当接收到下一个识别的输入帧时，接收过程会恢复。仅当上一个接收描述符由 DMA 所拥有时，该位置 1
6	RCS	接收完成状态（Receive Complete Status） 该位指示帧接收已完成，特定的帧状态信息已发布在描述符中，接收保持运行状态
5	UNS	下溢状态（Underflow Status） 该位指示在帧发送期间，发送缓冲区发生下溢，发送会进入挂起状态，且 Tx 描述符中下溢错误标志位置 1
4	OVS	上溢状态（Overflow Status） 该位指示在帧接收期间，接收缓冲区发生上溢，如果部分帧已传输到应用，则 Rx 描述符中上溢错误位置 1

Bit	Field	Description
3	TJS	发送 Jabber 超时状态 (Transfer Jabber Timeout Status) 该位指示发送 Jabber 定时器已过期, 这意味着发送器过度有效, 发送过程会中止并将其置于停止状态, 且描述符中发送 Jabber 超时标志位置 1
2	TUS	发送缓冲区不可用状态 (TxFIFO Unavailable Status) 该位指示发送列表中的下一个描述符由主机所拥有, DMA 无法获取, 发送会进入挂起状态 要恢复处理发送描述符, 主机应更改描述符的拥有关系, 然后发出发送轮询要求命令
1	TSS	发送停止状态 (Transfer Stop Status) 当发送过程进入停止状态时, 该位置 1
0	TCS	发送完成状态 (Transfer Complete Status) 该位指示帧发送已完成, 特定的帧状态信息已发布在描述符中

35.4.3.7 ENET_DMAMDR 工作模式寄存器

偏移地址: 0x1018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					DTCOE	RSF	DFRF	Res.		TSF	FTF	Res.			TTC
					rw	rw	rw			rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTC		STT	Res.					FEF	FUF	DGF	RTC		OSF	STR	Res.
rw		rw						rw	rw	rw	rw		rw	rw	

Bit	Field	Description
31:27	Reserved	保留, 必须保持复位值
26	DTCOE	丢弃 TCP/IP COE 错误帧 (Drop of TCP/IP Checksum Error Frame) 0: 如果该寄存器的 FEF 位置 1, 则丢弃所有错误帧 1: 如果接收帧中仅存在由接收校验和减荷引擎检测出来的错误, 则内核不会丢弃该帧 注: 这类帧在 MAC 接收到的以太网帧中没有任何错误 (包括 CRC 错误), 而仅在封装的有效负载中有错误
25	RSF	接收存储转发 (Receive Store & Forward) 0: RxFIFO 中的帧达到该寄存器的 RTC 位指定的值时, 才进行 RxFIFO 的读动作 1: RxFIFO 写入完整帧后, 就执行读动作, 同时忽略 RTC 位
24	DFRF	禁止刷新接收帧 (Disable Flush of Receive Frame) 该位置 1 时, RxDMA 不会因为接收描述符/缓冲区不可用而刷新任何帧, 参见“过程挂起”章节
23:22	Reserved	保留, 必须保持复位值

Bit	Field	Description
21	TSF	发送存储转发（Transfer Store & Forward） 0: TxFIFO 中的帧达到该寄存器的 TTC 位指定的值时，才进行 TxFIFO 的读动作，即 MAC 发送才启动 1: TxFIFO 中有一个完整帧，则 MAC 发送会启动，同时忽略 TTC 位 注：该位只有在已停止传输时才能更改
20	FTF	刷新发送 FIFO（Flush TxFIFO） 该位置 1 时，TxFIFO 复位为默认值，此时 TxFIFO 中的所有数据均会丢失/刷新。刷新操作结束时该位自动清零 注：此位清零之前不得对工作模式寄存器执行写操作
19:17	Reserved	保留，必须保持复位值
16:14	TTC	发送阈值控制（Transfer Threshold Control） 该位用于控制 TxFIFO 的阈值级别。当 TxFIFO 中的帧大小大于阈值时启动 DMA 发送。此外，长度小于阈值的完整帧会自动传输 000: 64 001: 128 010: 192 011: 256 100: 40 101: 32 110: 24 111: 16 注：该位只有在 TSF 位为 0 时才使用
13	STT	启动传输（Start Transfer） 0: 发送过程在完成发送当前帧的任务之后进入停止状态，并保存发送列表中的下一个描述符位置，该描述符位置在重启发送后会成为当前位置 1: 启动发送过程，DMA 检查发送列表中的当前位置，即 DMA 发送描述符地址寄存器（ENET_DMATXDSAR）设定的地址，或者上一次停止发送时的保留位置尝试获取描述符，来查找待发送的帧 如果当前描述符不属于 TxDMA，则发送过程会进入挂起状态；如果该命令在设置 DMA 发送描述符地址寄存器（ENET_DMATXDSAR）寄存器之前发出，则 TxDMA 行为无法预知 注：启动发送命令只有在传输已停止时才有效；停止发送命令只有在当前帧发送过程结束或发送过程处于挂起状态时才有效
12:8	Reserved	保留，必须保持复位值
7	FEF	传送错误帧（Forward Error Frame） 0: RxFIFO 会丢弃带错误状态（CRC 错误、冲突错误、巨帧、看门狗超时、上溢）的帧 如果某个帧的起始字节指针已传输到读控制器端（阈值模式下），则不会丢弃该帧；如果总线上未传输帧的起始字节，则 RxFIFO 会丢弃此错误帧 1: 除矮帧错误帧之外的所有帧都会转发

Bit	Field	Description
6	FUF	传送小字节帧（Forward Undersize Frame） 0: RxFIFO 丢弃所有不足 64 字节的帧，除非因接收阈值下限更低（例如 RTC=01）而导致此类帧已传输 1: RxFIFO 转发包含 PAD 字节和 CRC 的过小帧（无错误但长度不足 64 字节的帧）
5	DGF	丢弃巨帧（Drop Giant Frame） 0: RxFIFO 不丢弃巨帧 1: RxFIFO 丢弃巨帧
4:3	RTC	接收阈值控制（Receive Threshold Control） 该位用于控制 RxFIFO 的阈值级别。当 RxFIFO 中的帧大小大于阈值时启动 DMA 传输（请求）。此外，长度小于阈值的完成帧会自动传输 00: 64 01: 32 10: 96 11: 128 注：该位只有在 RSF 位为 0 时才使用
2	OSF	处理第二帧（Operate on Second Frame） 该位置 1 时会命令 DMA 处理第二个发送数据帧，即使尚未获得首个帧的状态。
1	STR	启动接收（Start Receive） 0: RxDMA 会在传输当前帧之后停止操作，并保存接收列表中的下一个描述符位置，该描述符位置在接收过程重启后会成为当前位置 1: 启动接收过程，DMA 尝试从接收列表中的当前位置，即 DMA 接收描述符地址寄存器（ENET_DMARXDSAR）设定的地址，或者上一次停止接收时的保留位置尝试获取描述符，并处理传入帧 如果 RxDMA 未占据任何描述符，则接收过程会进入挂起状态；如果该命令在设置 DMA 接收描述符地址寄存器（ENET_DMARXDSAR）寄存器之前发出，则 RxDMA 行为无法预知 注：启动接收命令只有在已停止接收时才有效；停止接收命令只有在当前接收过程处于运行（等待接收数据包）或挂起状态时才有效
0	Reserved	保留，必须保持复位值

35.4.3.8 ENET_DMAIR DMA 中断寄存器

偏移地址：0x101C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															NIE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIE	ERE	FBE	Res.		ETE	RWE	RSE	RUE	RCE	UNE	OVE	TJE	TUE	TSE	TCE

rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:17	Reserved	保留，必须保持复位值
16	NIE	普通中断使能（Normal Interrupt Enable） 0：普通事件中断无效 1：普通事件中断有效 注：普通事件是指 DMA 状态寄存器（ENET_DMASR）的 B0、B2、B6、B14 位所指示的事件
15	AIE	异常中断使能（Abnormal Interrupt Enable） 0：异常事件中断无效 1：异常事件中断有效 注：异常事件是指 DMA 状态寄存器（ENET_DMASR）的 B1、B3、B4、B5、B7、B8、B9、B10、B13 位所指示的事件
14	ERE	提前接收中断使能（Early Receive Interrupt Enable） 0：提前接收中断无效 1：提前接收中断使能 注：该位在 NIE 位置 1 后有效
13	FBE	致命总线错误中断使能（Fatal Bus Error Interrupt Enable） 0：致命总线错误中断无效 1：致命总线错误中断有效 注：该位在 AIE 位置 1 后有效
12:11	Reserved	保留，必须保持复位值
10	ETE	提前发送中断使能（Early Transfer Interrupt Enable） 0：提前发送中断无效 1：提前发送中断有效 注：该位在 AIE 位置 1 后有效
9	RWE	接收看门狗溢出中断使能（Receive Watchdog Timeout Interrupt Enable） 0：接收看门狗溢出中断无效 1：接收看门狗溢出中断有效 注：该位在 AIE 位置 1 后有效
8	RSE	接收停止中断使能（Receive Stop Interrupt Enable） 0：接收停止中断无效 1：接收停止中断有效 注：该位在 AIE 位置 1 后有效
7	RUE	接收缓冲区不可用中断使能（RxFIFO Unavailable Interrupt Enable） 0：接收缓冲区不可用中断无效 1：接收缓冲区不可用中断有效 注：该位在 AIE 位置 1 后有效
6	RCE	接收完成中断使能（Receive Complete Interrupt Enable） 0：接收完成中断无效 1：接收完成中断有效 注：该位在 NIE 位置 1 后有效

Bit	Field	Description
5	UNE	下溢中断使能（Underflow Interrupt Enable） 0：发送下溢中断无效 1：发送下溢中断有效 注：该位在 AIE 位置 1 后有效
4	OVE	上溢中断使能（Overflow Interrupt Enable） 0：接收上溢中断无效 1：接收上溢中断有效 注：该位在 AIE 位置 1 后有效
3	TJE	发送 Jabber 超时中断使能（Transfer Jabber Timeout Interrupt Enable） 0：发送 Jabber 超时中断无效 1：发送 Jabber 超时中断有效 注：该位在 AIE 位置 1 后有效
2	TUE	发送缓冲区不可用中断使能（TxFIFO Unavailable Interrupt Enable） 0：发送缓冲区不可用中断无效 1：发送缓冲区不可用中断有效 注：该位在 NIE 位置 1 后有效
1	TSE	发送停止中断使能（Transfer Stop Interrupt Enable） 0：发送停止中断无效 1：发送停止中断有效 注：该位在 AIE 位置 1 后有效
0	TCE	发送完成中断使能（Transfer Complete Interrupt Enable） 0：发送完成中断无效 1：发送完成中断有效 注：该位在 NIE 位置 1 后有效

35.4.3.9 ENET_DMAFLCR 帧丢失统计寄存器

偏移地址：0x1020

复位值：0x0000 0000

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

Bit	Field	Description
31:29	Reserved	保留，必须保持复位值
28	OVFF	上溢统计计数器溢出标志（OVF Count Overflow Flag） 该寄存器的 OVFC 位统计丢失帧计数器上溢时，该位置 1 注：该位在读出该寄存器 OVFC 位后自动清零

Bit	Field	Description
27:17	OVFC	上溢统计计数器 (OVF Count) 该位指示因 RxFIFO 上溢情况以及帧长度过短 (不足 64 字节的好帧) 而丢失的帧数量
16	BUAF	缓冲不可用统计计数器溢出标志 (BUA Count Overflow Flag) 该寄存器的 BUAC 位统计丢失帧计数器上溢时, 该位置 1 注: 该位在读出该寄存器 BUAC 位后自动清零
15:0	BUAC	缓冲不可用统计计数器 (Buffer Unavailable Count) 该位指示因缓冲区不可用 (无可用的接收描述符) 而丢失的帧数量

35.4.3.10 ENET_DMAWDTR 看门狗定时寄存器

偏移地址: 0x1024

复位值: 0x0000 0000

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

Bit	Field	Description
31:8	Reserved	保留, 必须保持复位值
31:0	RIWT	接收中断看门狗计数器 (Receive Interrupt Watchdog Timer Counter) 该位指示系统时钟周期数乘以 256 之后的时间值, 即看门狗定时器的设定时间值 看门狗定时器会在 RxDMA 结束帧传输之后由编程设定的值触发, 此时, DMA 状态寄存器的 RCS 位 (ENET_DMASR.RCS) 因相对应的描述符中 RDES1 的 REID 位置 1 而未被置 1, 直到看门狗定时器计时结束时, RCS 位才置 1 且看门狗定时器停止。当 DMA 状态寄存器的 RCS 位 (ENET_DMASR.RCS) 设为高电平时, 看门狗定时器复位

35.4.3.11 ENET_DMACURTXDSAR 当前发送描述符地址寄存器

偏移地址: 0x1048

复位值: 0x0000 0000

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

r

Bit	Field	Description
31:0	CURTXDSA	当前发送描述符地址（Current Transfer Descriptor Address） 该位指向 DMA 所读取的当前发送描述符的起始地址

35.4.3.12 ENET_DMACURRXDSAR 当前接收描述符地址寄存器

偏移地址：0x104C

复位值：0x0000 0000

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

Bit	Field	Description
31:0	CURRXDSA	当前接收描述符地址（Current Receive Descriptor Address） 该位指向 DMA 所读取的当前接收描述符的起始地址

35.4.3.13 ENET_DMACURTXBFAR 当前发送缓冲区地址寄存器

偏移地址：0x1050

复位值：0x0000 0000

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

Bit	Field	Description
31:0	CURTXBFA	当前发送缓冲区地址（Current Transfer Buffer Address） 该位指向 DMA 所读取的当前发送缓冲区地址

35.4.3.14 ENET_DMACURRXBFAR 当前接收缓冲区地址寄存器

偏移地址：0x1054

复位值：0x0000 0000

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

Bit	Field	Description
31:0	CURRXBFA	当前接收缓冲区地址（Current Receive Buffer Address） 该位指向 DMA 所读取的当前接收缓冲区地址

35.4.4 PTP 子控制器寄存器

35.4.4.1 ENET_PTPCR PTP 控制寄存器

偏移地址：0x0700

复位值：0x0000 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													PTPDA F	PTPTYPS	
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTPTYPS		PTPIPV 4	PTPIPV 6	PTPET H	PTPVE R	PTPNS S	PTPEAL L	Res.		PTPAD UP	PTPINT	PTPUP	PTPINI	PTPCS	PTPEN
rw		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	w

Bit	Field	Description
31:19	Reserved	保留，必须保持复位值
18	PTPDAF	PTP 目标地址过滤（PTP DA Filter） PTP 帧通过以太网发送时，若该位置 1，则对该 PTP 帧进行目标地址过滤
17:14	PTPTYPS	PTP 报文类型（PTP Message Type） 该位选择要拍摄时间戳快照的报文类型
13	PTPIPV4	IPv4 封装 PTP 帧（PTP Over UDP-IPv4） 0：忽略 Over UDP-IPv4 封装形式的 PTP 帧 1：接收 Over UDP-IPv4 封装形式的 PTP 帧
12	PTPIPV6	IPv6 封装 PTP 帧（PTP Over UDP-IPv6） 0：忽略 Over UDP-IPv6 封装形式的 PTP 帧 1：接收 Over UDP-IPv6 封装形式的 PTP 帧

Bit	Field	Description
11	PTPETH	以太网封装 PTP 帧 (PTP Over Ethernet) 0: 忽略 Over Ethernet 封装形式的 PTP 帧 1: 接收 Over Ethernet 封装形式的 PTP 帧
10	PTPVER	PTP 版本 (PTP Version) 0: 用 IEEE1588v1 版本格式对接收到的 PTP 帧处理 1: 用 IEEE1588v2 版本格式对接收到的 PTP 帧处理
9	PTPNSS	PTP 亚秒计数器选择 (PTP Nanosecond Select) 0: 亚秒加数寄存器的值达到 0x7FFFFFFFH, 时间戳秒计数器增加 1 秒, 亚秒加数寄存器归零重新计数 1: 亚秒加数寄存器的值达到 0x3B9AC9FFH, 时间戳秒寄存器增加 1 秒, 亚秒加数寄存器归零重新计数 注 1: 0x 3B9AC9FF=0d 999999999 注 2: 亚秒增量必须根据 PTP 参考时钟频率和此位的值正确进行编程
8	PTPEALL	使能所有时间戳 (PTP Timestamp Enable All) 该位置 1 时, 对接收的所有帧都拍摄时间戳快照
7:6	Reserved	保留, 必须保持复位值
5	PTPADUP	加数更新 (PTP Addend Updata) 该位置 1 时, PTP 基本加数寄存器 (ENET_PTPBSAR) 的值被更新到 PTP 进行精密校准, 更新结束时此位会硬件自动清零 注: 该寄存器位必须在读为 0 后, 才可以将其置 1
4	PTPINT	PTP 中断使能 (PTP Interrupt Enable) 如果系统时间大于在目标时间寄存器 (ENET_PTPTBSR/ENET_PTPTNSR) 中写入的值, 将产生时戳中断 注: 产生时间戳触发中断时, 此位会硬件自动清零
3	PTPUP	秒/压秒时间更新 (PTP Second/Nanosecond Updata) 该位置 1 时, PTP 更新秒寄存器 (ENET_PTPUBSR) 和 PTP 更新亚秒寄存器 (ENET_PTPUNSR) 中指定的值将对系统时间进行更新 (加入/减去)。更新结束时此位会硬件自动清零 注: 该寄存器位必须在自己和 PTPINI 两位均读为 0 后, 才可以将其置 1
2	PTPINI	秒/亚秒时间初始化 (PTP Second/Nanosecond Initial) 该位置 1 时, PTP 更新秒寄存器 (ENET_PTPUBSR) 和 PTP 更新亚秒寄存器 (ENET_PTPUNSR) 中指定的值将对系统时间进行初始化 (覆盖)。初始化结束时此位会硬件自动清零 注: 该寄存器位必须在读为 0 后, 才可以将其置 1
1	PTPCS	校准方式选择 (PTP Calibration Select) 0: 粗校准方式 1: 细校准方式
0	PTPEN	PTP 使能 (PTP Enable) 0: 时间戳功能无效, MAC 不会为发送和接收帧添加时间戳 1: 时间戳功能有效, 针对发送和接收帧添加时间戳 注: 该位一旦置 1, 需要对系统时间进行初始化

35.4.4.2 ENET_PTPNSAR 亚秒加数寄存器

偏移地址：0x0704

复位值：0x0000 0000

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

Bit	Field	Description
31:8	Reserved	保留，必须保持复位值
7:0	NSADD	亚秒计数器（Nanosecond Addend） 该寄存器包含系统亚秒寄存器递增时使用的 8 位值 在粗校准模式下，每个时钟周期都将该寄存器的值添加到系统时间；在细校准模式下，只在 32 位累加器溢出时将该寄存器的值添加到系统时间

35.4.4.3 ENET_PTPSBSR 系统秒寄存器

偏移地址：0x0708

复位值：0x0000 0000

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

Bit	Field	Description
31:0	SBSEC	系统秒时间（System Base Second） 该位表示系统时间的秒时间

35.4.4.4 ENET_PTPSNSR 系统亚秒寄存器

偏移地址：0x070C

复位值：0x0000 0000

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

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

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30:0	SNSEC	系统亚秒时间（System Nanosecond） 该位表示系统时间的亚秒时间

35.4.4.5 ENET_PTPUBSR 更新秒寄存器

偏移地址：0x0710

复位值：0x0000 0000

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

Bit	Field	Description
31:0	UBSEC	更新秒时间（Updata Base Second） 该位表示要更新进系统时间的秒时间

35.4.4.6 ENET_PTPUNSR 更新亚秒寄存器

偏移地址：0x0714

复位值：0x0000 0000

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

Bit	Field	Description
31	UPNSEC	亚秒时间更新选择（Updata Nanosecond Select） 0：从系统时间寄存器的值中减去更新寄存器的值 1：在系统时间寄存器的值中加上更新寄存器的值

Bit	Field	Description
30:0	UNSEC	更新亚秒时间（Update Nanosecond） 该位表示要更新进系统时间的亚秒时间

35.4.4.7 ENET_PTPBSAR 基本加数寄存器

偏移地址：0x0718

复位值：0x0000 0000

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

Bit	Field	Description
31:0	BSADD	基本加数器（Base Addend） 该寄存器由软件来重新线性调节时钟频率，使其与主时钟频率匹配。在每个时钟周期将该寄存器的值添加到 32 位累加器中 注：该寄存器只在细校准模式下有效

35.4.4.8 ENET_PTPTBSRx 目标秒寄存器（x=0~1）

偏移地址：0x071C/0x0780

复位值：0x0000 0000

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

Bit	Field	Description
31:0	TBSEC	目标秒时间（Target Second） 系统时间超过或等于该寄存器和目标亚秒寄存器的设定值时，发生中断事件或 PPS 事件

35.4.4.9 ENET_PTPTNSRx 目标亚秒寄存器 (x=0~1)

偏移地址: 0x0720/0x0784

复位值: 0x0000 0000

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

Bit	Field	Description
31	Reserved	保留, 必须保持复位值
30:0	TNSEC	目标亚秒时间 (Target Nanosecond) 系统时间超过或等于该寄存器和目标秒寄存器的设定值时, 发生中断或 PPS 事件

35.4.4.10 ENET_PTPSR PTP 状态寄存器

偏移地址: 0x0728

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										TSER1	TSTA1	TSER0	Res.	TSTA0	TSOVF
										rc	rc	rc		rc	rc

Bit	Field	Description
31:6	Reserved	保留, 必须保持复位值
5	TSER1	目标秒时间错误 1 (Target Second Error 1) 0: 要写入目标时间寄存器 1 的时间正确 1: 要写入目标时间寄存器 1 的时间已经过去, 该写入值错误
4	TSTA1	目标秒时间到达 1 (Target Second Time Arrived 1) 0: 系统时间未达到目标时间寄存器 1 指示的时间 1: 系统时间达到目标时间寄存器 1 指示的时间
3	TSER0	目标秒时间错误 0 (Target Second Error 0) 0: 要写入目标时间寄存器 0 的时间正确 1: 要写入目标时间寄存器 0 的时间已经过去, 该写入值错误
2	Reserved	保留, 必须保持复位值

Bit	Field	Description
1	TSTA0	目标秒时间到达 0 (Target Second Time Arrived 0) 0: 系统时间未达到目标时间寄存器 0 指示的时间 1: 系统时间达到目标时间寄存器 0 指示的时间
0	TSOVF	时间戳溢出 (Timestamp Overflow) 0: 系统时间未溢出 1: 系统时间溢出

35.4.4.11 ENET_PTTPPSR PPS 控制寄存器

偏移地址: 0x072C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TTSEL1		Res.		PPSOUT1			Res.	TTSEL0		PPSOM D	PPSOUT0			
	rw				rw				rw		rw	rw			

Bit	Field	Description
31:15	Reserved	保留, 必须保持复位值
14:13	TTSEL1	目标时间事件输出选择 1 (Target Time Event Select 1) 00: 目标秒/亚秒寄存器 1 (ENET_PTPTBSR1/ENET_PTPTNSR1) 仅用于中断输出事件 01: 无意义, 请不要设定 10: 目标秒/亚秒寄存器 1 (ENET_PTPTBSR1/ENET_PTPTNSR1) 用于中断输出事件和 PPS1 单次输出事件 11: 目标秒/亚秒寄存器 1 (ENET_PTPTBSR1/ENET_PTPTNSR1) 仅用于 PPS1 单次输出事件
12:11	Reserved	保留, 必须保持复位值
10:8	PPSOUT1	PPS 输出控制 1 (PPS Output Control 1) 001: 当系统时间等于目标秒/亚秒寄存器 1 时, 产生一次脉冲, 之后不再产生脉冲 其它值设定禁止 注: 该位读出时为 0
7	Reserved	保留, 必须保持复位值

Bit	Field	Description
6:5	TTSEL0	目标时间事件输出选择 0 (Target Time Event Select 0) 00: 目标秒/亚寄存器 0 (ENET_PTPTBSR0/ENET_PTPTNSR0) 仅用于中断输出事件 01: 无意义, 请不要设定 10: 目标秒/亚寄存器 0 (ENET_PTPTBSR0/ENET_PTPTNSR0) 用于中断输出事件和 PPS0 单次输出事件 11: 目标秒/亚寄存器 0 (ENET_PTPTBSR0/ENET_PTPTNSR0) 仅用于 PPS0 单次输出事件
4	PPSOMD	PPS 输出模式 (PPS Output Mode) 0: 连续输出模式 1: 单次输出模式 注: PPS0 支持上述两种模式, PPS1 仅支持单次输出模式
3:0	PPSOUT0	PPS 输出控制 0 (PPS Output Control 0) 连续输出模式时: 当系统时间发生秒进位 (亚秒时间溢出) 时, PPS0 根据下面的设定产生不同形式的输出: 0000: 连续输出间隔为 1Hz 的脉冲 0001: 连续输出频率为 2Hz 的时钟 0010: 连续输出频率为 4Hz 的时钟 0011: 连续输出频率为 8Hz 的时钟 0100: 连续输出频率为 16Hz 的时钟 1111: 连续输出频率为 32768Hz 的时钟 单次输出模式时: 0001: 当系统时间等于目标秒/亚秒寄存器 0 时, 产生一次脉冲, 之后不再产生脉冲 其它值设定禁止 注: 1) 连续输出模式时, PTP 控制寄存器的 PTPNSS 位 (ENET_PTPCR.PTPNSS) 须设定为 0 2) 单次输出模式时, 该位读出时为 0

35.4.5 PMT 电源管理寄存器

35.4.5.1 ENET_PMTRWFR 远程唤醒帧过滤寄存器

偏移地址: 0x0028

复位值: 0x0000 0000

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

RTWKFLT

rw

Bit	Field	Description
31:0	RTWKFLT	远程唤醒帧过滤值（Remote-Wakeup Filter Value） 应用将通过此地址对远程唤醒帧过滤寄存器进行写/读操作。详细信息请参见“远程唤醒帧过滤”章节

35.4.5.2 ENET_PMTCR PMT 控制寄存器

偏移地址：0x002C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTWKF R	Res.				RTWKPT				Res.						
rw					r										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					RTWKT R	GLUB	Res.		WKFR	MPFR	Res.		WKEN	MPEN	PWDN
					rw	rw			rc	rc			rw	rw	rw

Bit	Field	Description
31	RTWKFR	远程唤醒帧过滤器复位（Remote-Wakeup Filter Reset） 0：远程唤醒帧过滤寄存器指针不复位 1：远程唤醒帧过滤寄存器指针复位 注：复位完成后该位自动清零
31:27	Reserved	保留，必须保持复位值
26:24	RTWKPT	远程唤醒帧过滤器指针（Remote-Wakeup Filter Point） 该位表明远程唤醒帧过滤寄存器指针在第几个寄存器上（该值为 0~7）
23:11	Reserved	保留，必须保持复位值
10	RTWKTR	远程唤醒帧收发使能（Remote-Wakeup Frame Transfer & Receive Enable） 该位置 1 后，MAC 不再接收除魔术数据包和远程唤醒帧以外的包，并将接收到的远程唤醒帧转发给应用 注 1：该位在接收到魔术数据包或远程唤醒帧之后会自动变为 0；也可以在接收到魔术数据包或远程唤醒帧之前软件复位 注 2：当该寄存器的 GLUB 位或 WKEN 位或 MPEN 位为 1、同时 PWDN 位为 0 时，该位必须设定为 1 注 3：在 PWDN 位为 1 时，该位设定无效
9	GLUB	全局单播（Global Unicast） 0：将 MAC DA 地址过滤的任意单播帧视为无效 1：将 MAC DA 地址过滤的任意单播帧视为唤醒帧
8:7	Reserved	保留，必须保持复位值

Bit	Field	Description
6	WKFR	远程唤醒帧状态 (Remote-Wakeup Frame Receive Status) 0: 未接收到远程唤醒帧 1: 接收到远程唤醒帧 注: 该位在读过之后自动清零
5	MPFR	魔术数据包状态 (Magic-Packet Frame Receive Status) 0: 未接收到魔术数据包 1: 接收到魔术数据包 注: 该位在读过之后自动清零
4:3	Reserved	保留, 必须保持复位值
2	WKEN	远程唤醒帧接收使能 (Remote-Wakeup Frame Receive Enable) 0: 不接收远程唤醒帧 1: 接收远程唤醒帧, 触发 PMT 事件
1	MPEN	魔术数据包接收使能 (Magic-Packet Frame Receive Enable) 0: 不接收魔术数据包 1: 接收魔术数据包, 触发 PMT 事件
0	PWDN	掉电模式 (Power Down Mode) 0: 活动 (Active) 模式 1: 低功耗 (Power Down) 模式 在低功耗模式时, 所有接收到的帧都丢弃, 只有接收到魔术数据包或远程唤醒帧时, 此位会自动清零, 同时退出 Power Down 模式; 该位清零后收到的帧会转发给应用 注: 只有该寄存器的 WKEN 位或 MPEN 位置 1 时, 才可以将此位置 1

35.4.6 MMC 管理计数寄存器

35.4.6.1 ENET_MMCCR MMC 控制寄存器

偏移地址: 0x0100

复位值: 0x0000 0000

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

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

Bit	Field	Description
5	CPSEL	计数器预设选择 (Counter Preset Select) 0: 将所有 MMC 统计寄存器的值预设“预设半值” (0x7FF0H) 1: 将所有 MMC 统计寄存器的值预设“预设全值” (0xFFFF0H)
4	CPSET	计数器预设设定 (Counter Preset Setting) 该位置 1 后, 根据该寄存器的 CPSEL 位的设定将所有的 MMC 统计寄存器预设全值或半值 注: 该位在置位后 1 个时钟后硬件自动清零
3	CFIX	计数器锁定 (Counter Fixed) 该位置 1 后, 所有 MMC 统计计数器的值被冻结, 不再因为接收或发送帧而变化
2	CRRST	计数器读后复位 (Counter Reset after Read) 该位置 1 后, 所有 MMC 统计计数器在被读取之后复位为零 注: 不管该寄存器的 CFIX 位是否有效, 该位都可有效
1	COSR	计数器溢出后停止计数 (Counter Overflow Stop Rollover) 0: 计数器计到 0xFFFFH 后, 归零, 重新计数 1: 计数器计到 0xFFFFH 后, 停止, 不再计数
0	CRST	计数器复位 (Counter Reset) 该位置 1 后, 所有 MMC 统计计数器的值复位 注: 该位在置位后 1 个时钟后硬件自动清零

35.4.6.2 ENET_MMCSR MMC 接收状态寄存器

偏移地址: 0x0104

复位值: 0x0000 0000

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

Bit	Field	Description
31:18	Reserved	保留, 必须保持复位值
17	RUGIS	接收单播好帧中断状态 (Receive Unicast Good Frame Interrupt Status) 0: 接收到的单播好帧数未到半值或全值 1: 接收到的单播好帧数达到半值或全值 注: 该位在读出对应 MMC 统计寄存器后自动清零
16:7	Reserved	保留, 必须保持复位值

Bit	Field	Description
6	RALIEIS	接收对齐错误帧中断状态（Receive Align Error Frame Interrupt Status） 0：接收到的对齐错误帧数未到半值或全值 1：接收到的对齐错误帧数达到半值或全值 注：该位在读出对应 MMC 统计寄存器后自动清零
5	RRCREIS	接收 CRC 错误帧中断状态（Receive CRC Error Frame Interrupt Status） 0：接收到的 CRC 错误帧数未到半值或全值 1：接收到的 CRC 错误帧数达到半值或全值 注：该位在读出对应 MMC 统计寄存器后自动清零
4:0	Reserved	保留，必须保持复位值

35.4.6.3 ENET_MMCTSR MMC 发送状态寄存器

偏移地址：0x0108

复位值：0x0000 0000

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

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值
21	TAGIS	发送所有好帧中断状态（Transfer All Good Frame Interrupt Status） 0：发送的好帧数未到半值或全值 1：发送的好帧数达到半值或全值 注：该位在读出对应 MMC 统计寄存器后自动清零
20:16	Reserved	保留，必须保持复位值
15	TGMCIS	多个冲突后发送好帧中断状态（Transfer Good Frame after Multi-Collision Interrupt Status） 0：多个冲突后发送的好帧数未到半值或全值 1：多个冲突后发送的好帧数达到半值或全值 注：该位在读出对应 MMC 统计寄存器后自动清零
14	TGSCIS	单个冲突后发送好帧中断状态（Transfer Good Frame after Single-Collision Interrupt Status） 0：单个冲突后发送的好帧数未到半值或全值 1：单个冲突后发送的好帧数达到半值或全值 注：该位在读出对应 MMC 统计寄存器后自动清零
13:0	Reserved	保留，必须保持复位值

35.4.6.4 ENET_MMCRIR MMC 接收中断寄存器

偏移地址: 0x010C

复位值: 0x0000 0000

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

Bit	Field	Description
31:18	Reserved	保留, 必须保持复位值
17	RUGIM	接收单播好帧中断屏蔽 (Receive Unicast Good Frame Interrupt Mask) 0: 接收到的单播好帧数达到半值或全值时, 单播好帧 MMC 统计中断不屏蔽 1: 接收到的单播好帧数达到半值或全值时, 单播好帧 MMC 统计中断屏蔽
16:7	Reserved	保留, 必须保持复位值
6	RALIEIM	接收对齐错误帧中断屏蔽 (Receive Align Error Frame Interrupt Mask) 0: 接收到的对齐错误帧数达到半值或全值时, 对齐错误帧 MMC 统计中断不屏蔽 1: 接收到的对齐错误帧数达到半值或全值时, 对齐错误帧 MMC 统计中断屏蔽
5	RCRCEIM	接收 CRC 错误帧中断屏蔽 (Receive Align Error Frame Interrupt Mask) 0: 接收到的 CRC 错误帧数达到半值或全值时, CRC 错误帧 MMC 统计中断不屏蔽 1: 接收到的 CRC 错误帧数达到半值或全值时, CRC 错误帧 MMC 统计中断屏蔽
4:0	Reserved	保留, 必须保持复位值

35.4.6.5 ENET_MMCTIR MMC 发送中断寄存器

偏移地址: 0x0110

复位值: 0x0000 0000

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

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值
21	TAGIM	发送所有好帧中断屏蔽（Transfer All Good Frame Interrupt Mask） 0：发送的好帧数达到半值或全值时，发送好帧 MMC 统计中断不屏蔽 1：发送的好帧数达到半值或全值时，发送好帧 MMC 统计中断屏蔽
20:16	Reserved	保留，必须保持复位值
15	TGMCIM	多个冲突后发送好帧中断屏蔽（Transfer Good Frame after Multi-Collision Interrupt Mask） 0：多个冲突后发送的好帧数达到半值或全值时，多个冲突后发送好帧 MMC 统计中断不屏蔽 1：多个冲突后发送的好帧数达到半值或全值时，多个冲突后发送好帧 MMC 统计中断屏蔽
14	TGSCIM	单个冲突后发送好帧中断屏蔽（Transfer Good Frame after Single-Collision Interrupt Mask） 0：单个冲突后发送的好帧数达到半值或全值时，单个冲突后发送好帧 MMC 统计中断不屏蔽 1：单个冲突后发送的好帧数达到半值或全值时，单个冲突后发送好帧 MMC 统计中断屏蔽
13:0	Reserved	保留，必须保持复位值

35.4.6.6 ENET_MMCTGSCR 单个冲突错误后发送好帧统计寄存器

偏移地址：0x014C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TGSC															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGSC															
r															
Bit	Field		Description												
31:0	TGSC		单个冲突后发送好帧数量（Transfer Good Frame after Single-Collision） 该位表示在半双工模式下于单个冲突后成功发送的帧的数量												

35.4.6.7 ENET_MMCTGMCR 多个冲突错误后发送好帧统计寄存器

偏移地址：0x0150

复位值：0x0000 0000

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

TGMC															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGMC															
r															
Bit	Field		Description												
31:0	TGMC		多个冲突后发送好帧数量（Transfer Good Frame after Multi-Collision） 该位表示在半双工模式下于多个冲突后成功发送的帧的数量												

35.4.6.8 ENET_MMCTGR 发送好帧统计寄存器

偏移地址：0x0168

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMCTG															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMCTG															
r															
Bit	Field		Description												
31:0	MMCTG		发送好帧数量（MMC Transfer Good Frame） 该位表示发送的好帧的数量												

35.4.6.9 ENET_MMCRRCR CER CRC 错误接收帧统计寄存器

偏移地址：0x0194

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCRCE															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCRCE															
r															
Bit	Field		Description												
31:0	RCRCE		接收对齐错误帧数量（MMC Receive CRC Error Frame） 该位表示接收到的 CRC 错误帧的数量												

35.4.6.10 ENET_MMCRALIER 对齐错误接收帧统计寄存器

偏移地址：0x0198

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RALIE															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RALIE															
r															
Bit	Field		Description												
31:0	RALIE		接收对齐错误帧数量（MMC Receive Align Error Frame） 该位表示接收到的对齐错误帧的数量												

35.4.6.11 ENET_MMCRUGR 接收单播好帧统计寄存器

偏移地址：0x01C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMCRUG															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMCRUG															
r															
Bit	Field		Description												
31:0	MMCRUG		接收单播好帧数量（MMC Receive Unicast Good Frame） 该位表示接收到的单播好帧的数量												

TPIU: 追踪端口的接口单元

36.2.2 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。

36.2.3 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 36-1 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

36.3 ID 代码和锁定机制

在芯片内部有多个 ID 编码，如下表：

表 36-2 ID 编码

ID 名	芯片
DEV_ID	0x4D4D0800
CPU TAP SW ID	0x1BE12AEB
CPU TAP JTAG ID	0x0BE11AEB

36.3.1 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。

36.3.2 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 0xE00FE000_0xE00FEFFF 的 4KB ROM 表中。

36.4 SW 调试端口

36.4.1 SW 协议介绍

此同步串行协议使用 2 个引脚：主机到目标的时钟信号（SWCLK）与双向数据信号（SWDIO）。

SWDIO 作为双向数据线，需连接上拉电阻（ARM 建议值 100K）。SWDIO 引脚内嵌了上拉电阻，无需额外的外接电阻。

数据从低位开始传输，允许读写寄存器组 DPACC 与 APACC。

根据协议，当 SWDIO 改变方向时，同时需要插入一个转换时间（默认一个 Bit 时间，具体可以通过 SWCLK 调整），这段间内，任何设备不能驱动此信号线。

36.4.2 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 36-3 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 36-4 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 36-5 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。

36.4.3 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;

36.4.4 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。如果下一次的访问不是对 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；

36.4.5 SW-DP 寄存器

当 APnDP=0 时，可以访问以下这些寄存器。

表 36-6 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 0x1BE1 2AEB（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。

36.4.6 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

36.5 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 36-7 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 0x0BA04AEB（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

IR	数据寄存器	描述
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
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 36-8 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。

36.6 MCU 调试模块 (DBGMCU)

MCU 调试模块提供以下调试器协助功能：

- 支持追踪模式
- 断点时定时器与看门狗的时钟控制

36.6.1 支持跟踪调试模式

支持非侵入式调试，通过配置 TRACE_MODE 位，可以选择相应的跟踪调试模式。当使用追踪调试模式时，必须先置位 TRACE_IOE，分配对应的跟踪调试引脚。

36.6.2 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

36.7 寄存器

36.7.1 寄存器总览

表 36-9 DBG 寄存器概览

Offset	Acronym	Register Name	Reset
0x80	DBG_IDCODE	DBG ID 编码寄存器	0x4D4D0800
0x84	DBG_CR	DBG 控制寄存器	0x00000000

36.7.2 DBG_IDCODE ID 编码寄存器

偏移地址：0x80 （只支持 32 位访问，只读）

复位值：0x4D4D0800

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

36.7.3 DBG_CR 控制寄存器

偏移地址：0x84 （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
Reserved					debug_l ptim_sto	debug_ti m20_sto	debug_ti m22_sto	debug_ti m21_sto	debug_ti m17_sto	debug_ti m16_sto	debug_ti m7_stop	debug_ti m6_stop	debug_ti m5_stop	debug_ti m8_stop	Reserve d
					p	p	p	p	p	p					
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	debug_ti m4_stop	debug_ti m3_stop	debug_ti m2_stop	debug_ti m1_stop	debug_ wwdg_st op	debug_i wdg_sto p	trance_mode	trance_i oe	Reserved	debug_s top_for_l do	Reserved
	rw	rw	rw	rw	rw	rw	rw	rw		rw	

Bit	Field	Description
31:27	Reserved	保留，必须保持复位值
26	debug_lptim_stop	当内核进入调试状态时低功耗计数器停止工作 (LPTIM counter stopped when core is halted) 0: 选中低功耗计数器仍然正常工作 1: 选中低功耗计数器停止工作
25:21	debug_timx_stop	当内核进入调试状态时计数器停止工作 (x = 16,17,21,22,20) (TIMx 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
20:17	debug_timx_stop	当内核进入调试状态时计数器停止工作 (x = 8,5,6,7) (TIMx 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
13:10	debug_timx_stop	当内核进入调试状态时计数器停止工作 x = 1,2,3,4 (TIMx counter stopped when core is halted) 0: 选中定时器的计数器仍然正常工作 1: 选中定时器的计数器停止工作
9	debug_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	debug_iwdg_stop	当内核进入调试状态时看门狗停止工作 (Debug independent watchdog stopped when core is halted) 0: 看门狗计数器仍然正常工作 1: 看门狗计数器停止工作
7:6	trace_mode	TRACE_MODE 选择 当 TRACE_IOEN=1 时: TRACE_MODE=00: 跟踪引脚使用异步模式; TRACE_MODE=01: 跟踪引脚使用同步模式, 并且数据长度为 1; TRACE_MODE=10: 跟踪引脚使用同步模式, 并且数据长度为 2; TRACE_MODE=11: 跟踪引脚使用同步模式, 并且数据长度为 4。
5	trace_ioen	TRACE_IOEN: 跟踪引脚分配控制使能 0: 不分配跟踪引脚(默认状态)。 1: 分配跟踪引脚, 开启 trace_clk 使能
4	Reserved	保留，必须保持复位值

Bit	Field	Description
3	debug_stop_for_ldo	调试停机模式（Debug Stop mode） 0: 进入 stop 模式时，关掉 LDO 1: 不能进入 stop 模式，不会关掉 LDO 和 PLL 供电，cpu 进入 deepsleep，HCLK 关闭
2: 0	Reserved	保留，必须保持复位值

37 Device Electronic Signature 器件电子签名

37.1 简介

器件电子签名是存放在闪存存储器系统存储区域内用来唯一识别一颗微控制器的身份标识码(96 bits)。在任何情况下,用户都不可以修改器件电子签名。

可以通过软件读取器件电子签名,用来实现以下功能:

- 作为密码使用,在编程闪存时,通过器件电子签名与软件加密算法结合使用,提高代码在闪存中的安全性
- 作为序列号,用作终端应用中的序列号
- 激活安全引导流程

37.2 寄存器

基地址: 0x1FFF F7E0

表 37-1 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

37.2.1 UID1 唯一标识码

偏移地址: 0x00

复位值: 其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (31: 16)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (15: 0)															
r															
Bit	Field		Description												
31: 0	U_ID (31: 0)		U_ID: 唯一身份标志 31: 0 位 (31: 0 unique ID bits)												

37.2.2 UID2 唯一标识码

偏移地址：0x04

复位值：其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (63: 48)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (47: 32)															
r															
Bit	Field		Description												
31: 0	U_ID (63: 32)		U_ID: 唯一身份标志 63: 32 位 (63: 32 unique ID bits)												

37.2.3 UID3 唯一标识码

偏移地址：0x08

复位值：其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID (95: 80)															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID (79: 64)															
r															
Bit	Field		Description												
31: 0	U_ID (95: 64)		U_ID: 唯一身份标志 95 : 64 位 (95: 64 unique ID bits)												

38 修订记录

表 38-1 修订记录

修订时间	修订版本	修订内容
2024/11/18	V1.02	1.优化章节顺序排布 2.修复原 CRS、CORDIC 章节格式问题
2024/10/28	V1.01	芯片特定配置章节：补充 DMA MAP 配置 SYSCFG 相关使用说明
2024/08/28	V1.0	文档进版
2023/01/12	V0.9	更新 ENET、EXTI、SYSCFG、FlexCAN 和 FSMC 章节内容
2022/10/20	V0.52	更新 LPUART、RCC、CORDIC、ENET、Timer、DMA 章节内容
2022/07/28	V0.51	1.更新 LPUART、CRC_POLY、FlexCAN、Timer、QSPI 等章节内容 2.添加了 MM32F5280 产品
2022/04/24	V0.5	初版发布

