

32 位 MCU
ES32VF2264

参 考 手 册

- 产品简介
- 数据手册
- 参考手册

上海东软载波微电子有限公司

2025-03-19

目 录

第 1 章	文档约定	29
1.1	寄存器读写权限的设定	29
第 2 章	系统概述	30
2.1	概述	30
2.2	系统框图	30
2.3	模块功能类别.....	31
2.3.1	32 位 E902 内核模块.....	32
2.3.2	存储器及存储器接口.....	33
2.3.3	系统模块.....	33
2.3.4	时钟管理.....	34
2.3.5	外部接口.....	34
2.3.6	安全管理及运算加速.....	34
2.3.7	定时器	35
2.3.8	通信模块.....	37
2.3.9	模拟模块.....	38
第 3 章	芯片配置指引	39
3.1	概述	39
3.2	E902 内核配置.....	39
3.2.1	E902 内核.....	39
3.2.2	总线.....	39
3.2.3	系统计数器	39
3.2.4	调试器件.....	39
3.3	核内局部中断控制器.....	40
3.3.1	中断优先级	40
3.3.2	中断向量分配	40
3.4	异步唤醒中断和事件.....	42
3.4.1	异步中断唤醒源.....	42
3.4.2	事件唤醒.....	42
3.5	存储器及存储器接口.....	43
3.5.1	系统总线和存储器	43
3.6	系统模块配置.....	45
3.6.1	DMA 控制器配置	45
3.6.2	独立看门狗定时器配置.....	47
3.6.2.1	独立看门狗定时器的时钟	47
3.6.2.2	独立看门狗定时器的低功耗动作模式	47
3.6.3	窗口看门狗定时器配置.....	47
3.6.3.1	窗口看门狗定时器的时钟	47
3.6.3.2	窗口看门狗定时器的低功耗动作模式	47
3.6.4	时钟管理配置	48
3.6.4.1	HOSC 的低功耗动作模式.....	48
3.6.4.2	HRC4M 的低功耗动作模式	48
3.6.4.3	HRC48M 的低功耗动作模式	48

3.6.4.4	PLL 的低功耗动作模式	48
3.6.4.5	LRC 的低功耗动作模式	48
3.7	外部接口配置	48
3.7.1	通用 IO 及端口控制配置	48
3.7.1.1	端口特殊配置说明	48
3.8	定时器配置	49
3.8.1	通用定时器	49
3.8.1.1	通用定时器例化说明	49
3.8.1.2	通用定时器的时钟	49
3.8.1.3	通用定时器的低功耗动作模式	49
3.8.2	基本定时器配置	49
3.8.2.1	基本定时器例化说明	49
3.8.2.2	基本定时器的时钟	49
3.8.2.3	基本定时器的低功耗动作模式	49
3.9	通信配置	50
3.9.1.1	I2C 接口的时钟	50
3.9.1.2	I2C 接口的低功耗动作模式	50
3.9.2	串行外设接口 (SPI) 配置	50
3.9.2.1	串行外设接口 (SPI) 的时钟	50
3.9.2.2	串行外设接口 (SPI) 的低功耗动作模式	50
3.9.3	通用异步收发器 (UART)	51
3.9.3.1	通用异步收发器 (UART) 的时钟	51
3.9.3.2	通用异步收发器 (UART) 的低功耗动作模式	51
3.10	模拟配置	52
3.10.1	ADC 控制配置	52
3.10.1.1	ADC 模块例化	52
3.10.1.2	ADC 转换通道配置	52
3.10.1.3	ADC 电源及参考电压	52
3.10.1.4	ADC 的时钟	52
3.10.1.5	ADC 的低功耗动作模式	53
第 4 章	系统总线和存储器	54
4.1	概述	54
4.2	系统总线	55
4.2.1	S0: DMA 总线	55
4.2.2	S1: S 总线	55
4.2.3	S2: I 总线	55
4.2.4	总线矩阵	55
4.2.5	AHB/APB 总线桥	55
4.3	存储器的组织结构	55
4.3.1	系统存储器映射	55
4.3.2	FLASH 存储器映射	56
4.3.3	SRAM 存储器映射	56
4.3.4	外设存储映射	56
4.3.5	紧耦合总线映射	56

4.4	启动引导	56
第 5 章	存储器系统控制 (MSC)	57
5.1	概述	57
5.2	特性	57
5.3	结构框图	57
5.4	功能描述	58
5.4.1	Flash 保护	58
5.4.1.1	IAP 操作保护 KEY	58
5.4.1.2	Flash 写保护区	58
5.4.1.3	Flash 私有读保护区	58
5.4.1.4	数据 Flash 区	58
5.4.1.5	Flash 全局读保护	59
5.4.2	Flash 页擦除	60
5.4.3	Flash 字编程	60
5.4.4	Flash 编程数据 FIFO	60
5.4.5	IAP 自编程硬件固化模块	60
5.4.5.1	CODE 区单页擦函数	61
5.4.5.2	CODE 区单字编程函数	61
5.4.5.3	CODE 区多字编程	61
5.4.5.4	DATA 区单页擦函数	61
5.4.5.5	DATA 区单字编程函数	61
5.4.5.6	DATA 区多字编程	61
5.5	特殊功能寄存器	63
5.5.1	寄存器列表	63
5.5.2	寄存器描述	64
5.5.2.1	FLASH 程序区关键码寄存器 (MSC_FLASHKEY)	64
5.5.2.2	FLASH 擦除编程地址寄存器 (MSC_FLASHADDR)	64
5.5.2.3	FLASH 编程 FIFO 寄存器 (MSC_FLASHFIFO)	64
5.5.2.4	FLASH 编程数据寄存器 (MSC_FLASHDR)	65
5.5.2.5	FLASH 命令寄存器 (MSC_FLASHCMD)	65
5.5.2.6	FLASH 控制寄存器 (MSC_FLASHCR)	65
5.5.2.7	FLASH 状态寄存器 (MSC_FLASHSR)	67
5.5.2.8	存储器读取等待时间寄存器 (MSC_MEMWAIT)	68
5.5.2.9	FLASH 擦除编程地址反码寄存器 (MSC_FLASHADDINV)	69
第 6 章	系统配置控制器 (SYSCFG)	70
6.1	概述	70
6.2	特性	70
6.3	功能描述	70
6.3.1	系统寄存器写保护	70
6.3.2	启动地址配置	70
6.4	特殊功能寄存器	70
6.4.1	寄存器列表	70
6.4.2	寄存器描述	71
6.4.2.1	系统写保护寄存器 (SYSCFG_PROT)	71

6.4.2.2	启动地址寄存器 (SYSCFG_BTADDR)	71
6.4.2.3	TIM 刹车源配置寄存器 (SYSCFG_TBKCFG)	71
6.4.2.4	PIS 生产端 GPIO 配置寄存器 (SYSCFG_PISIOCFG)	72
第 7 章	电源管理 (PMU) 及低功耗模式.....	75
7.1	概述	75
7.2	特性	75
7.3	结构框图	76
7.4	功能描述	77
7.4.1	芯片电源	77
7.4.1.1	主系统电源域	77
7.4.1.2	独立的模拟模块电源和参考电压	77
7.4.2	电源监视	77
7.4.2.1	上电复位 (POR)	77
7.4.2.2	欠压复位 (BOR)	77
7.4.2.3	低电压检测 (LVD)	78
7.4.3	低功耗模式	79
7.4.3.1	低功耗模式转换	79
7.4.3.2	系统时钟速度	80
7.4.3.3	外设时钟门控	80
7.4.3.4	RUN 模式	80
7.4.3.5	SLEEP 模式	80
7.4.3.6	STOP 模式	80
7.4.3.7	低功耗模式下各模块操作	81
7.5	特殊功能寄存器	82
7.5.1	寄存器列表	82
7.5.2	寄存器描述	83
7.5.2.1	PMU 控制寄存器 (PMU_CR)	83
7.5.2.2	PMU 状态寄存器 (PMU_SR)	85
7.5.2.3	LVD 控制寄存器 (PMU_LVDCR)	86
7.5.2.4	唤醒延时寄存器 (PMU_TWUR)	87
第 8 章	复位管理 (RMU)	88
8.1	概述	88
8.2	特性	88
8.3	结构框图	88
8.4	功能描述	89
8.4.1	硬件复位	89
8.4.1.1	上电复位	89
8.4.1.2	欠压复位	89
8.4.1.3	端口复位	90
8.4.1.4	看门狗复位	90
8.4.1.5	LOCKUP 复位	90
8.4.1.6	读取配置字错误复位	90
8.4.2	软件复位	90
8.4.2.1	芯片复位 (CHIPRST)	90

8.4.2.2	CPU 复位 (CPURST)	90
8.4.2.3	内核复位请求 (SYSRSTREQ)	90
8.4.2.4	外设软件复位.....	90
8.5	特殊功能寄存器	91
8.5.1	寄存器列表	91
8.5.2	寄存器描述	92
8.5.2.1	RMU 控制寄存器 (RMU_CR)	92
8.5.2.2	RMU 复位状态寄存器 (RMU_RSTSR)	92
8.5.2.3	RMU 清复位状态寄存器 (RMU_CRSTSR)	94
8.5.2.4	AHB1 外设复位寄存器 (RMU_AHB1RSTR)	95
8.5.2.5	AHB2 外设复位寄存器 (RMU_AHB2RSTR)	95
8.5.2.6	APB 外设复位寄存器 (RMU_APBSTR)	96
第 9 章	时钟管理 (CMU)	98
9.1	概述	98
9.2	特性	98
9.3	结构框图.....	99
9.4	功能描述	100
9.4.1	外部高速振荡器时钟 (HOSC)	100
9.4.2	内部高速 RC 振荡器时钟 (HRC4M、HRC48M)	101
9.4.3	内部低速 RC 振荡器时钟 (LRC)	101
9.4.4	内部倍频时钟 (PLL)	101
9.4.5	系统时钟选择	101
9.4.6	时钟安全管理	101
9.5	特殊功能寄存器	103
9.5.1	寄存器列表	103
9.5.2	寄存器描述	104
9.5.2.1	CMU 控制状态寄存器 (CMU_CSR)	104
9.5.2.2	CMU 配置寄存器 (CMU_CFGR)	106
9.5.2.3	CMU 时钟使能寄存器 (CMU_CLKENR)	107
9.5.2.4	CMU 时钟状态寄存器 (CMU_CLKSR)	108
9.5.2.5	PLL 配置寄存器 (CMU_PLLCFG)	109
9.5.2.6	HOSC 配置寄存器 (CMU_HOSCCFG)	109
9.5.2.7	HOSC 安全管理控制寄存器 (CMU_HOSMCR)	110
9.5.2.8	PLL 失锁管理控制寄存器 (CMU_PULMCR)	111
9.5.2.9	CMU 时钟输出控制寄存器 (CMU_CLKOCR)	112
9.5.2.10	BUZZ 控制寄存器 (CMU_BUZZCR)	113
9.5.2.11	AHB 外设时钟使能寄存器 (CMU_AHBENR)	114
9.5.2.12	APB 外设时钟使能寄存器 (CMU_APBENR)	114
9.5.2.13	外设时钟低功耗模式使能寄存器 (CMU_LPENR)	116
第 10 章	时钟同步单元 (CSU)	117
10.1	概述	117
10.2	特性	117
10.3	结构图.....	118
10.4	功能描述	119

10.4.1	同步信号输入	119
10.4.2	频率偏差计数器	119
10.4.3	频率偏差评估与自动校准	120
10.4.4	CSU 初始化与配置	121
10.4.5	CSU 中断	122
10.5	特殊功能寄存器	123
10.5.1	寄存器列表	123
10.5.2	寄存器描述	124
10.5.2.1	CSU 控制寄存器 (CSU_CON)	124
10.5.2.2	CSU 配置寄存器 (CSU_CFG)	125
10.5.2.3	CSU 状态寄存器 (CSU_STAT)	127
10.5.2.4	CSU 中断开启寄存器 (CSU_IER)	128
10.5.2.5	CSU 中断关闭寄存器 (CSU_IDR)	129
10.5.2.6	CSU 中断使能状态寄存器 (CSU_IVS)	130
10.5.2.7	CSU 中断标志寄存器 (CSU_RIF)	131
10.5.2.8	CSU 中断屏蔽后状态寄存器 (CSU_IFM)	132
10.5.2.9	CSU 中断清除寄存器 (CSU_ICR)	133
第 11 章	DMA 控制器 (DMA)	134
11.1	概述	134
11.2	特性	134
11.3	请求映射	135
11.4	结构图	135
11.5	功能描述	135
11.5.1	传输事务	135
11.5.2	仲裁率	137
11.5.3	优先级	137
11.5.4	传输模式	138
11.5.4.1	存储器到存储器模式 (Memory-to-memory)	138
11.5.4.2	存储器到外设模式 (Memory-to-peripheral)	138
11.5.4.3	外设到存储器模式 (Peripheral-to-memory)	139
11.5.4.4	外设到外设模式 (Peripheral-to-peripheral)	139
11.5.5	地址递增	140
11.5.6	循环模式 (仅用于存储器和外设之间传输)	140
11.5.7	数据宽度、对齐和字节序	140
11.5.7.1	解决 AHB 外设不支持字节或半字写传输的问题	142
11.5.8	信道配置流程	142
11.5.9	传输完成	143
11.5.10	传输暂停	143
11.5.11	中断事件	143
11.6	特殊功能寄存器	144
11.6.1	寄存器列表	144
11.6.2	寄存器描述	146
11.6.2.1	DMA 中断使能寄存器 (DMA_IER)	146
11.6.2.2	DMA 中断禁止寄存器 (DMA_IDR)	148

11.6.2.3	DMA 中断使能状态寄存器 (DMA_IVS)	150
11.6.2.4	DMA 中断标志寄存器 (DMA_RIF)	152
11.6.2.5	DMA 中断屏蔽后状态寄存器 (DMA_IFM)	154
11.6.2.6	DMA 中断清除寄存器 (DMA_ICR)	156
11.6.2.7	DMA 通道 0 控制寄存器 (DMA_CON0)	158
11.6.2.8	DMA 通道 0 源地址寄存器 (DMA_SAR0)	160
11.6.2.9	DMA 通道 0 目的地址寄存器 (DMA_DAR0)	160
11.6.2.10	DMA 通道 0 数据传输数量寄存器 (DMA_NDT0)	161
11.6.2.11	DMA 通道 1 控制寄存器 (DMA_CON1)	162
11.6.2.12	DMA 通道 1 源地址寄存器 (DMA_SAR1)	164
11.6.2.13	DMA 通道 1 目的地址寄存器 (DMA_DAR1)	164
11.6.2.14	DMA 通道 1 数据传输数量寄存器 (DMA_NDT1)	165
11.6.2.15	DMA 通道 2 控制寄存器 (DMA_CON2)	166
11.6.2.16	DMA 通道 2 源地址寄存器 (DMA_SAR2)	168
11.6.2.17	DMA 通道 2 目的地址寄存器 (DMA_DAR2)	168
11.6.2.18	DMA 通道 2 数据传输数量寄存器 (DMA_NDT2)	169
11.6.2.19	DMA 通道 3 控制寄存器 (DMA_CON3)	170
11.6.2.20	DMA 通道 3 源地址寄存器 (DMA_SAR3)	172
11.6.2.21	DMA 通道 3 目的地址寄存器 (DMA_DAR3)	172
11.6.2.22	DMA 通道 3 数据传输数量寄存器 (DMA_NDT3)	173
11.6.2.23	DMA 通道 4 控制寄存器 (DMA_CON4)	174
11.6.2.24	DMA 通道 4 源地址寄存器 (DMA_SAR4)	176
11.6.2.25	DMA 通道 4 目的地址寄存器 (DMA_DAR4)	176
11.6.2.26	DMA 通道 4 数据传输数量寄存器 (DMA_NDT4)	177
11.6.2.27	DMA 通道 5 控制寄存器 (DMA_CON5)	178
11.6.2.28	DMA 通道 5 源地址寄存器 (DMA_SAR5)	180
11.6.2.29	DMA 通道 5 目的地址寄存器 (DMA_DAR5)	180
11.6.2.30	DMA 通道 5 数据传输数量寄存器 (DMA_NDT5)	181
11.6.2.31	DMA 通道 6 控制寄存器 (DMA_CON6)	182
11.6.2.32	DMA 通道 6 源地址寄存器 (DMA_SAR6)	184
11.6.2.33	DMA 通道 6 目的地址寄存器 (DMA_DAR6)	184
11.6.2.34	DMA 通道 6 数据传输数量寄存器 (DMA_NDT6)	185
11.6.2.35	DMA 通道 x 复用选择寄存器 (DMA_CHx_SELCON) (x=0...6)	186
第 12 章	外设互联 (PIS)	188
12.1	概述	188
12.2	特性	188
12.3	结构框图	189
12.4	功能描述	190
12.4.1	生产端信号	190
12.4.2	消费端信号	190
12.4.3	PIS 通道选择	192
12.4.3.1	同一时钟域互联	193
12.4.3.2	AHB 和 APB 外设之间互联	193

12.4.3.3	生产端信号为异步信号	193
12.4.4	UART 输出调制	194
12.5	特殊功能寄存器	195
12.5.1	寄存器列表	195
12.5.2	寄存器描述	196
12.5.2.1	PIS 通道 x 控制寄存器 (PIS_CHx_CON) (x=0..7)	196
12.5.2.2	PIS 通道端口输出使能寄存器 (PIS_CH_OER)	200
12.5.2.3	PIS 消费端通道控制寄存器 0 (PIS_TAR_CON0)	201
12.5.2.4	PIS 消费端通道控制寄存器 1 (PIS_TAR_CON1)	203
12.5.2.5	UARTx 输出调制控制寄存器 (UARTx_TXMCR) (x=0..4)	205
第 13 章	独立看门狗 (IWDT)	206
13.1	概述	206
13.2	特性	206
13.3	功能描述	206
13.3.1	硬件看门狗	207
13.3.2	软件看门狗	207
13.3.3	调试模式	208
13.3.4	寄存器访问保护	208
13.4	特殊功能寄存器	209
13.4.1	寄存器列表	209
13.4.2	寄存器描述	210
13.4.2.1	IWDT 计数器装载值寄存器 (IWDT_LOAD)	210
13.4.2.2	IWDT 计数器当前值寄存器 (IWDT_VALUE)	210
13.4.2.3	IWDT 控制寄存器 (IWDT_CON)	211
13.4.2.4	IWDT 中断标志清除寄存器 (IWDT_INTCLR)	211
13.4.2.5	IWDT 中断标志寄存器 (IWDT_RIS)	212
13.4.2.6	IWDT 锁定寄存器 (IWDT_LOCK)	212
第 14 章	窗口看门狗 (WWDT)	213
14.1	概述	213
14.2	特性	213
14.3	功能描述	213
14.3.1	窗口看门狗	213
14.3.2	调试模式	215
14.3.3	寄存器访问保护	215
14.4	特殊功能寄存器	216
14.4.1	寄存器列表	216
14.4.2	寄存器描述	217
14.4.2.1	WWDT 计数器装载值寄存器 (WWDT_LOAD)	217
14.4.2.2	WWDT 计数器当前值寄存器 (WWDT_VALUE)	217
14.4.2.3	WWDT 控制寄存器 (WWDT_CON)	218
14.4.2.4	WWDT 中断标志清除寄存器 (WWDT_INTCLR)	219
14.4.2.5	WWDT 中断标志寄存器 (WWDT_RIS)	219
14.4.2.6	WWDT 锁定寄存器 (WWDT_LOCK)	220
第 15 章	通用 IO 及端口控制 (GPIO)	221

15.1	概述	221
15.2	特性	221
15.3	结构框图	222
15.4	功能描述	222
15.4.1	端口控制寄存器	222
15.4.2	端口数据寄存器	223
15.4.3	外部端口中断	223
15.4.4	通用 GPIO 配置	224
15.4.5	外部中断与唤醒	224
15.4.6	外设功能端口复用	225
15.4.7	GPIO 锁定	225
15.4.8	GPIO 输入配置	225
15.4.9	GPIO 输出配置	226
15.4.10	模拟输入配置	226
15.5	特殊功能寄存器	227
15.5.1	寄存器列表	227
15.5.2	寄存器描述	228
15.5.2.1	GPIO 端口输入数据寄存器 (GPIO_DIN)	228
15.5.2.2	GPIO 端口输出数据寄存器 (GPIO_DOUT)	228
15.5.2.3	GPIO 端口置位和复位寄存器 (GPIO_BSRR)	229
15.5.2.4	GPIO 端口翻转寄存器 (GPIO_BIR)	229
15.5.2.5	GPIO 端口模式寄存器 (GPIO_MODE)	230
15.5.2.6	GPIO 端口开漏寄存器 (GPIO_OD)	230
15.5.2.7	GPIO 端口上拉和下拉寄存器 (GPIO_PUPD)	231
15.5.2.8	GPIO 端口输出驱动寄存器 (GPIO_ODRV)	231
15.5.2.9	GPIO 端口滤波寄存器 (GPIO_FLT)	232
15.5.2.10	GPIO 端口类型寄存器 (GPIO_TYPE)	232
15.5.2.11	GPIO 端口复用功能寄存器 0 (GPIO_FUNC0)	233
15.5.2.12	GPIO 端口复用功能寄存器 1 (GPIO_FUNC1)	234
15.5.2.13	GPIO 端口锁定寄存器 (GPIO_LOCK)	235
15.5.2.14	外部中断上升沿触发使能寄存器 (GPIO_EXTIRER)	236
15.5.2.15	外部中断下降沿触发使能寄存器 (GPIO_EXTIFER)	236
15.5.2.16	外部中断使能寄存器 (GPIO_EXTIEN)	237
15.5.2.17	外部中断标志寄存器 (GPIO_EXTIFLAG)	237
15.5.2.18	外部中断标志置位寄存器 (GPIO_EXTISFR)	238
15.5.2.19	外部中断标志清零寄存器 (GPIO_EXTICFR)	238
15.5.2.20	外部中断端口选择寄存器 0 (GPIO_EXTIPSR0)	239
15.5.2.21	外部中断端口选择寄存器 1 (GPIO_EXTIPSR1)	241
15.5.2.22	外部中断滤波控制寄存器 (GPIO_EXTIFLTCR)	243
第 16 章	循环冗余校验 (CRC)	244
16.1	概述	244
16.2	特性	244
16.3	结构框图	244
16.4	功能描述	245

16.4.1	常规操作	245
16.4.2	DMA 请求	245
16.5	特殊功能寄存器	247
16.5.1	寄存器列表	247
16.5.2	寄存器描述	248
16.5.2.1	CRC 控制寄存器 (CRC_CR)	248
16.5.2.2	CRC 写数据寄存器 (CRC_DATA)	249
16.5.2.3	CRC 种子寄存器 (CRC_SEED)	250
16.5.2.4	CRC 校验值寄存器 (CRC_CHECKSUM)	250
第 17 章	高级定时器 (AD16C4T)	251
17.1	概述	251
17.2	特性	251
17.3	结构框图	252
17.4	功能描述	252
17.4.1	预分频器	252
17.4.2	重复计数器	254
17.4.3	时钟源	255
17.4.3.1	内部时钟源 (INT_CLK)	255
17.4.3.2	外部时钟源 1	255
17.4.3.3	外部时钟源 2	256
17.4.3.4	内部触发输入 (ITn)	257
17.4.4	计数模式	257
17.4.4.1	递增计数模式	257
17.4.4.2	递减计数模式	259
17.4.4.3	中心对齐模式	260
17.4.5	捕获/比较通道	261
17.4.6	输入捕获模式	263
17.4.6.1	PWM 输入模式	264
17.4.7	PWM 模式	264
17.4.7.1	PWM 边沿对齐模式	265
17.4.7.2	PWM 中心对齐模式	265
17.4.8	输出比较模式	266
17.4.8.1	外部事件清除比较输出	267
17.4.8.2	强制输出模式	267
17.4.9	单脉冲模式	268
17.4.10	互补输出与死区时间	269
17.4.11	刹车功能	270
17.4.12	编码器接口模式	272
17.4.13	输入异或功能	274
17.4.14	霍尔传感器接口	274
17.4.15	外部触发的同步	274
17.4.15.1	复位模式	274
17.4.15.2	门控模式	275
17.4.15.3	触发模式	276

17.4.15.4	选择外部时钟源 2 的触发模式.....	277
17.4.166	步 PWM 生成.....	278
17.4.17	调试模式.....	278
17.5	特殊功能寄存器.....	279
17.5.1	寄存器列表.....	279
17.5.2	寄存器描述.....	280
17.5.2.1	控制寄存器 1 (AD16C4T_CON1)	280
17.5.2.2	控制寄存器 2 (AD16C4T_CON2)	283
17.5.2.3	从模式控制寄存器 (AD16C4T_SMCON)	285
17.5.2.4	中断使能寄存器 (AD16C4T_IER)	288
17.5.2.5	中断禁止寄存器 (AD16C4T_IDR)	289
17.5.2.6	中断有效状态寄存器 (AD16C4T_IVS)	290
17.5.2.7	原始中断标志寄存器 (AD16C4T_RIF)	291
17.5.2.8	中断标志屏蔽寄存器 (AD16C4T_IFM)	294
17.5.2.9	中断清零寄存器 (AD16C4T_ICR)	296
17.5.2.10	事件生成寄存器 (AD16C4T_SGE)	297
17.5.2.11	通道捕获/比较模式寄存器 1 (AD16C4T_CHMR1)	299
17.5.2.12	通道捕获/比较模式寄存器 2 (AD16C4T_CHMR2)	304
17.5.2.13	捕获/比较使能极性寄存器 (AD16C4T_CCEP)	307
17.5.2.14	计数器 (AD16C4T_COUNT)	309
17.5.2.15	时钟预分频器 (AD16C4T_PRES)	310
17.5.2.16	计数器自动装载寄存器 (AD16C4T_AR)	310
17.5.2.17	重复计数寄存器 (AD16C4T_REPAR)	311
17.5.2.18	通道捕获/比较寄存器 1 (AD16C4T_CCVAL1)	312
17.5.2.19	通道捕获/比较寄存器 2 (AD16C4T_CCVAL2)	312
17.5.2.20	通道捕获/比较寄存器 3 (AD16C4T_CCVAL3)	313
17.5.2.21	通道捕获/比较寄存器 4 (AD16C4T_CCVAL4)	313
17.5.2.22	刹车和死区配置寄存器 (AD16C4T_BDCFG)	314
17.5.2.23	DMA 使能寄存器 (AD16C4T_DMAEN)	317
第 18 章	通用定时器 (GP16C4T0~2)	318
18.1	概述.....	318
18.2	特性.....	318
18.3	结构框图.....	319
18.4	功能描述.....	320
18.4.1	预分频器.....	320
18.4.2	时钟源.....	321
18.4.2.1	内部时钟源 (INT_CLK)	321
18.4.2.2	外部时钟源 1	321
18.4.2.3	外部时钟源 2	322
18.4.2.4	内部触发输入 (ITn)	323
18.4.3	计数器模式.....	324
18.4.3.1	递增计数模式.....	324
18.4.3.2	递减计数模式.....	327
18.4.3.3	中心对齐模式.....	328

18.4.4	捕获/比较通道	329
18.4.5	输入捕获模式	330
18.4.5.1	PWM 输入模式	331
18.4.6	PWM 模式	332
18.4.6.1	PWM 边沿对齐模式	332
18.4.6.2	PWM 中心对齐模式	333
18.4.7	输出比较模式	334
18.4.7.1	外部事件清除比较输出	334
18.4.8	单脉冲模式	335
18.4.9	编码器接口模式	336
18.4.10	输入异或功能	338
18.4.11	定时器和外部触发的同步	338
18.4.11.1	复位模式	338
18.4.11.2	门控模式	339
18.4.11.3	触发模式	339
18.4.11.4	选择外部时钟源 2 的触发模式	340
18.4.12	调试模式	341
18.5	特殊功能寄存器	342
18.5.1	寄存器列表	342
18.5.2	寄存器描述	343
18.5.2.1	控制寄存器 1 (GP16C4T_CON1)	343
18.5.2.2	控制寄存器 2 (GP16C4T_CON2)	346
18.5.2.3	从模式控制寄存器 (GP16C4T_SMCON)	348
18.5.2.4	中断使能寄存器 (GP16C4T_IER)	351
18.5.2.5	中断禁止寄存器 (GP16C4T_IDR)	352
18.5.2.6	中断有效状态寄存器 (GP16C4T_IVS)	353
18.5.2.7	原始中断标志寄存器 (GP16C4T_RIF)	354
18.5.2.8	中断标志屏蔽寄存器 (GP16C4T_IFM)	357
18.5.2.9	中断清零寄存器 (GP16C4T_ICR)	359
18.5.2.10	软件生成事件寄存器 (GP16C4T_SGE)	360
18.5.2.11	捕获/比较模式寄存器 1 (GP16C4T_CHMR1)	361
18.5.2.12	捕获/比较模式寄存器 2 (GP16C4T_CHMR2)	366
18.5.2.13	捕获/比较使能寄存器 (GP16C4T_CCEP)	369
18.5.2.14	计数器寄存器 (GP16C4T_COUNT)	370
18.5.2.15	预分频寄存器 (GP16C4T_PRES)	370
18.5.2.16	自动重载寄存器 (GP16C4T_AR)	371
18.5.2.17	捕获/比较寄存器 1 (GP16C4T_CCVAL1)	371
18.5.2.18	捕获/比较寄存器 2 (GP16C4T_CCVAL2)	372
18.5.2.19	捕获/比较寄存器 3 (GP16C4T_CCVAL3)	372
18.5.2.20	捕获/比较寄存器 4 (GP16C4T_CCVAL4)	372
18.5.2.21	DMA 使能寄存器 (GP16C4T_DMAEN)	373
第 19 章	基本定时器 (BS16T)	374
19.1	概述	374
19.2	特性	374

19.3	结构框图	374
19.4	功能描述	374
19.4.1	预分频器	374
19.4.2	时钟源	375
19.4.3	递增计数模式	375
19.4.4	调试模式	375
19.5	特殊功能寄存器	376
19.5.1	寄存器列表	376
19.5.2	寄存器描述	377
19.5.2.1	控制寄存器 1 (BS16T_CON1)	377
19.5.2.2	中断使能寄存器 (BS16T_IER)	379
19.5.2.3	中断禁止寄存器 (BS16T_IDR)	379
19.5.2.4	中断有效状态寄存器 (BS16T_IVS)	380
19.5.2.5	原始中断标志寄存器 (BS16T_RIF)	381
19.5.2.6	中断标志屏蔽寄存器 (BS16T_IFM)	382
19.5.2.7	中断清零寄存器 (BS16T_ICR)	382
19.5.2.8	事件生成寄存器 (BS16T_SGE)	383
19.5.2.9	计数器 (BS16T_COUNT)	383
19.5.2.10	预分频器 (BS16T_PRES)	384
19.5.2.11	自动重载寄存器 (BS16T_AR)	384
19.5.2.12	DMA 使能寄存器 (BS16T_DMAEN)	385
第 20 章	串行总线 (I2C0~1)	386
20.1	概述	386
20.2	特性	386
20.3	结构图	387
20.4	功能描述	388
20.4.1	I2C 总线协议	388
20.4.1.1	START 和 STOP 条件协议	388
20.4.1.2	应答位	389
20.4.1.3	I2C 寻址从协议	389
20.4.1.4	I2C 发送和接收协议	390
20.4.2	I2C 时钟要求	391
20.4.3	数据传输	392
20.4.4	I2C 从机模式	393
20.4.5	I2C 主机模式	398
20.4.6	I2C_TIMINGR 寄存器的配置的例子	402
20.4.7	SMBus 具体功能	404
20.4.8	SMBus 初始化	405
20.4.9	SMBus: I2C_TIMEOUTR 寄存器配置的例子	407
20.4.10	DMA 请求	408
20.4.11	通信错误类型	408
20.4.12	I2C 中断	409
20.4.13	调试模式	409
20.5	特殊功能寄存器	410

20.5.1	寄存器列表	410
20.5.2	寄存器描述	411
20.5.2.1	I2C 控制寄存器 1 (I2C_CON1)	411
20.5.2.2	I2C 控制寄存器 2 (I2C_CON2)	413
20.5.2.3	I2C 本机地址寄存器 1 (I2C_ADDR1)	416
20.5.2.4	I2C 本机地址寄存器 2 (I2C_ADDR2)	417
20.5.2.5	I2C 时钟寄存器 (I2C_TIMINGR)	418
20.5.2.6	I2C 超时寄存器 (I2C_TIMEOUTR)	419
20.5.2.7	I2C 状态寄存器 (I2C_STAT)	420
20.5.2.8	I2C PEC 寄存器 (I2C_PECR)	421
20.5.2.9	I2C 接收器数据寄存器 (I2C_RXDATA)	422
20.5.2.10	I2C 发送器数据寄存器 (I2C_TXDATA)	422
20.5.2.11	I2C 中断开启寄存器 (I2C_IER)	423
20.5.2.12	I2C 中断关闭寄存器 (I2C_IDR)	425
20.5.2.13	I2C 中断功能有效状态寄存器 (I2C_IVS)	426
20.5.2.14	I2C 原始中断标志寄存器 (I2C_RIF)	428
20.5.2.15	I2C 中断标志屏蔽寄存器 (I2C_IFM)	430
20.5.2.16	I2C 中断清除寄存器 (I2C_ICR)	432
第 21 章	串行外设接口 (SPI0~1) / 音频总线 (I2S0)	434
21.1	概述	434
21.2	特性	434
21.2.1	SPI 的主要特点	434
21.2.2	I2S 的主要特点	435
21.3	SPI 实现	435
21.4	SPI 结构图	436
21.5	SPI 功能描述	437
21.5.1	时钟相位和极性控制	437
21.5.2	数据帧格式	438
21.5.3	从机片选 (NSS) 引脚管理	438
21.5.4	主机与从机的单对单通讯应用	439
21.5.4.1	全双工通信	439
21.5.4.2	半双工通信	440
21.5.4.3	单工通信	441
21.5.5	标准多从机通讯应用	442
21.5.6	多主机通讯应用	443
21.5.7	SPI 从机模式	444
21.5.8	SPI 主机模式	445
21.5.9	数据发送和接收	446
21.5.9.1	接收和发送 FIFO 缓存	446
21.5.9.2	在主机模式下启动通信序列	446
21.5.9.3	在从机模式下启动通信序列	447
21.5.9.4	处理数据发送与接收	447
21.5.10	SPI 关闭流程	455
21.5.11	DMA 请求	456

21. 5. 12CRC 计算	458
21. 5. 13SPI 状态标志	460
21. 5. 13. 1 发送 FIFO 缓存为空 (TXE)	460
21. 5. 13. 2 发送 FIFO 缓存为满 (TXF)	460
21. 5. 13. 3 发送 FIFO 缓存上溢 (TXOV)	460
21. 5. 13. 4 发送 FIFO 缓存下溢 (TXUD)	460
21. 5. 13. 5 发送 FIFO 缓存阈值 (TXTH)	460
21. 5. 13. 6 接收 FIFO 缓存为非空 (RXNE)	460
21. 5. 13. 7 接收 FIFO 缓存为满 (RXF)	460
21. 5. 13. 8 接收 FIFO 缓存上溢 (RXOV)	460
21. 5. 13. 9 接收 FIFO 缓存下溢 (RXUD)	460
21. 5. 13. 10 接收 FIFO 缓存阈值 (RXTH)	461
21. 5. 13. 11 通信忙 (BUSY)	461
21. 5. 14SPI 中断事件	462
21. 5. 14. 1 发送 FIFO 缓存为空 (TXE)	462
21. 5. 14. 2 发送 FIFO 缓存上溢 (TXOV)	462
21. 5. 14. 3 发送 FIFO 缓存下溢 (TXUD)	462
21. 5. 14. 4 发送 FIFO 缓存阈值 (TXTH)	462
21. 5. 14. 5 接收 FIFO 缓存为非空 (RXNE)	462
21. 5. 14. 6 接收 FIFO 缓存为满 (RXF)	463
21. 5. 14. 7 接收 FIFO 缓存上溢 (RXOV)	463
21. 5. 14. 8 接收 FIFO 缓存下溢 (RXUD)	463
21. 5. 14. 9 接收 FIFO 缓存阈值 (RXTH)	463
21. 5. 14. 10 CRC 错误 (CRCERR)	463
21. 5. 14. 11 模式故障 (MODF)	464
21. 5. 14. 12 TI 模式帧格式错误 (FRE)	464
21. 5. 15SPI TI 模式	464
21. 6 I2S 结构图	465
21. 7 I2S 功能描述	466
21. 7. 1 音频协议	466
21. 7. 1. 1 I2S Philips 标准	466
21. 7. 1. 2 MSB 对齐标准	468
21. 7. 1. 3 LSB 对齐标准	470
21. 7. 1. 4 PCM 标准	472
21. 7. 2 时钟产生器	472
21. 7. 3 I2S 主机模式	475
21. 7. 3. 1 设置流程	475
21. 7. 3. 2 发送序列	475
21. 7. 3. 3 接收序列	475
21. 7. 4 I2S 从机模式	476
21. 7. 4. 1 设置流程	476
21. 7. 4. 2 自动侦测同步	476
21. 7. 4. 3 发送序列	476
21. 7. 4. 4 接收序列	477

21.7.5	I2S 状态标志	477
21.7.5.1	发送 FIFO 缓存为空 (TXE)	477
21.7.5.2	发送 FIFO 缓存为满 (TXF)	477
21.7.5.3	发送 FIFO 缓存上溢 (TXOV)	477
21.7.5.4	发送 FIFO 缓存下溢 (TXUD)	477
21.7.5.5	发送 FIFO 缓存阈值 (TXTH)	477
21.7.5.6	接收 FIFO 缓存为非空 (RXNE)	478
21.7.5.7	接收 FIFO 缓存为满 (RXF)	478
21.7.5.8	接收 FIFO 缓存上溢 (RXOV)	478
21.7.5.9	接收 FIFO 缓存下溢 (RXUD)	478
21.7.5.10	接收 FIFO 缓存阈值 (RXTH)	478
21.7.5.11	通信忙 (BUSY)	478
21.7.5.12	声道标志 (CHSIDE)	478
21.7.6	I2S 中断事件	479
21.7.6.1	发送 FIFO 缓存为空 (TXE)	479
21.7.6.2	发送 FIFO 缓存上溢 (TXOV)	479
21.7.6.3	发送 FIFO 缓存下溢 (TXUD)	479
21.7.6.4	发送 FIFO 缓存阈值 (TXTH)	479
21.7.6.5	接收 FIFO 缓存为非空 (RXNE)	479
21.7.6.6	接收 FIFO 缓存为满 (RXF)	480
21.7.6.7	接收 FIFO 缓存上溢 (RXOV)	480
21.7.6.8	接收 FIFO 缓存下溢 (RXUD)	480
21.7.6.9	接收 FIFO 缓存阈值 (RXTH)	480
21.7.6.10	帧格式错误 (FRE)	480
21.8	特殊功能寄存器	482
21.8.1	寄存器列表	482
21.8.2	寄存器描述	483
21.8.2.1	SPI 控制寄存器 1 (SPI_CON1)	483
21.8.2.2	SPI 控制寄存器 2 (SPI_CON2)	486
21.8.2.3	SPI 状态寄存器 (SPI_STAT)	488
21.8.2.4	SPI 数据寄存器 (SPI_DATA)	490
21.8.2.5	SPI CRC 多项式寄存器 (SPI_CRCPOLY)	491
21.8.2.6	SPI RX CRC 寄存器 (SPI_RXCRC)	492
21.8.2.7	SPI TX CRC 寄存器 (SPI_TXCRC)	493
21.8.2.8	SPI I2S 配置寄存器 (SPI_I2SCFG)	494
21.8.2.9	SPI I2S 预分频寄存器 (SPI_I2SPR)	496
21.8.2.10	SPI 中断使能寄存器 (SPI_IER)	497
21.8.2.11	SPI 中断禁用寄存器 (SPI_IDR)	499
21.8.2.12	SPI 中断有效状态寄存器 (SPI_IVS)	501
21.8.2.13	SPI 原始中断标志状态寄存器 (SPI_RIF)	503
21.8.2.14	SPI 中断标志屏蔽状态寄存器 (SPI_IFM)	505
21.8.2.15	SPI 中断清除寄存器 (SPI_ICR)	507
第 22 章	通用异步收发器 (UART0~4)	509
22.1	概述	509

22.2	特性	509
22.3	结构图.....	511
22.4	功能描述.....	512
22.4.1	具体功能配置	513
22.4.2	功能描述.....	513
22.4.3	发送器	515
22.4.4	接收器	516
22.4.4.1	防抖电路.....	516
22.4.4.2	起始位侦测	517
22.4.5	状态寄存器	520
22.4.6	波特率产生器	521
22.4.7	自动波特率侦测.....	522
22.4.8	自动流量控制	524
22.4.8.1	RTS 流量控制.....	525
22.4.8.2	CTS _n 流量控制.....	525
22.4.8.3	RS485 驱动使能 (DE)	525
22.4.9	Modbus 通信	526
22.4.10	校验控制.....	526
22.4.11	多处理器通信	527
22.4.12	LIN 模式 (仅 EUART 支持)	528
22.4.13	单线半双工通讯 (仅 EUART 支持)	530
22.4.14	智能卡模式 (仅 EUART 支持)	531
22.4.15	IrDA SIR 模式 (仅 EUART 支持)	532
22.4.16	使用 DMA 连续通讯.....	534
22.4.17	中断配置.....	535
22.5	特殊功能寄存器	537
22.5.1	寄存器列表	537
22.5.2	寄存器描述	538
22.5.2.1	接收缓冲寄存器 (UART_RXBUF)	538
22.5.2.2	发送缓冲寄存器 (UART_TXBUF)	538
22.5.2.3	波特率寄存器 (UART_BRR)	539
22.5.2.4	格式控制寄存器 (UART_LCON)	540
22.5.2.5	模式控制寄存器 (UART_MCON)	543
22.5.2.6	RS485 控制寄存器 (UART_RS485)	545
22.5.2.7	智能卡控制寄存器 (UART_SCARD)	547
22.5.2.8	LIN 控制寄存器 (UART_LIN)	549
22.5.2.9	接收超时寄存器 (UART_RTOR)	550
22.5.2.10	状态寄存器 (UART_STAT)	551
22.5.2.11	中断开启寄存器 (UART_IER)	553
22.5.2.12	中断关闭寄存器 (UART_IDR)	555
22.5.2.13	中断功能有效状态寄存器 (UART_IVS)	557
22.5.2.14	原始中断状态寄存器 (UART_RIF)	559
22.5.2.15	中断屏蔽后状态寄存器 (UART_IFM)	561
22.5.2.16	中断清除寄存器 (UART_ICR)	563

第 23 章	通用串行总线 (USB)	565
23.1	概述	565
23.2	特性	565
23.3	结构图	566
23.4	功能描述	567
23.4.1	操作模式	567
23.4.2	设备模式	567
23.4.2.1	端点控制器	568
23.4.2.2	设备模式下的 IN 传输	568
23.4.2.3	设备模式下的 OUT 传输	569
23.4.2.4	调度	570
23.4.2.5	其他操作	570
23.4.2.6	设备模式挂起	571
23.4.2.7	起始帧	571
23.4.2.8	USB 复位	571
23.4.2.9	连接和断开	571
23.4.3	主机模式	571
23.4.3.1	端点控制器	572
23.4.3.2	主机模式下的 IN 传输	572
23.4.3.3	主机模式下的 OUT 传输	573
23.4.3.4	事务调度	573
23.4.3.5	干扰	573
23.4.3.6	主机挂起	574
23.4.3.7	USB 复位	574
23.4.3.8	连接/断开	574
23.4.3.9	OTG 模式	574
23.4.3.10	开始会话	574
23.4.3.11	主机协商	575
23.5	特殊功能寄存器	576
23.5.1	寄存器列表	576
23.5.2	寄存器描述	578
23.5.2.1	USB 设备功能地址寄存器 (USB_FADDR)	578
23.5.2.2	USB 电源管理寄存器 (USB_POWER)	579
23.5.2.3	USB DP/DM 控制器寄存器 (USB_DPDMCON)	581
23.5.2.4	USB 软件控制 CID 寄存器 (USB_SWCID)	582
23.5.2.5	USB 软件控制 VBUS 寄存器 (USB_SWVBUS)	583
23.5.2.6	USB 帧号寄存器 1 (USB_FRAME1)	584
23.5.2.7	USB 帧号寄存器 2 (USB_FRAME2)	584
23.5.2.8	USB 端点索引寄存器 (USB_INDEX)	585
23.5.2.9	USB 设备控制寄存器 (USB_DEVCON)	586
23.5.2.10	USB 发送最大数据包大小寄存器 (USB_TXMAXP)	587
23.5.2.11	USB 发送端点中的控制状态寄存器 1 (USB_CSR0L_TXCSRL)	588
23.5.2.12	USB 发送端点中的控制状态寄存器 2 (USB_CSR0H_TXCSRH)	

	593	
23.5.2.13	USB 接收最大数据包大小寄存器 (USB_RXMAXP)	595
23.5.2.14	USB 接收控制状态寄存器 1 (USB_RXCSRL)	596
23.5.2.15	USB 接收控制状态寄存器 2 (USB_RXCSRH)	599
23.5.2.16	USB 接收端点中的接收字节数寄存器 (USB_COUNT0_RX1) 601	
23.5.2.17	USB 接收字节数寄存器 2 (USB_RXCOUNT2)	602
23.5.2.18	USB 发送协议类型寄存器 (USB_TXTYPE)	603
23.5.2.19	USB 端点 NAK 响应超时或发送轮询间隔寄存器 (USB_NAKLIMIT0_TXINTERVAL)	604
23.5.2.20	USB 接收协议类型寄存器 (USB_RXTYPE)	605
23.5.2.21	USB 接收轮询间隔寄存器 (USB_RXINTERVAL)	606
23.5.2.22	USB 发送 FIFO 配置寄存器 1 (USB_TXFIFO1)	607
23.5.2.23	USB 发送 FIFO 配置寄存器 2 (USB_TXFIFO2)	608
23.5.2.24	USB 接收 FIFO 配置寄存器 1 (USB_RXFIFO1)	609
23.5.2.25	USB 接收 FIFO 配置寄存器 2 (USB_RXFIFO2)	610
23.5.2.26	USB 端点 0 FIFO (USB_EP0FIFO)	611
23.5.2.27	USB 端点 1 FIFO (USB_EP1FIFO)	611
23.5.2.28	USB 端点 2 FIFO (USB_EP2FIFO)	612
23.5.2.29	USB 端点 3 FIFO (USB_EP3FIFO)	612
23.5.2.30	USB 端点 4 FIFO (USB_EP4FIFO)	613
23.5.2.31	USB 发送中断使能寄存器 (USB_TXIER)	614
23.5.2.32	USB 接收中断使能寄存器 (USB_RXIER)	615
23.5.2.33	USB 发送中断禁用寄存器 (USB_TXIDR)	616
23.5.2.34	USB 接收中断禁用寄存器 (USB_RXIDR)	617
23.5.2.35	USB 发送中断使能状态寄存器 (USB_TXIVS)	618
23.5.2.36	USB 接收中断使能状态寄存器 (USB_RXIVS)	619
23.5.2.37	USB 发送原始中断事件标志寄存器 (USB_TXRIF)	620
23.5.2.38	USB 接收原始中断事件标志寄存器 (USB_RXRIF)	621
23.5.2.39	USB 发送中断屏蔽标志寄存器 (USB_TXIFM)	622
23.5.2.40	USB 接收中断屏蔽标志寄存器 (USB_RXIFM)	623
23.5.2.41	USB 发送中断清除寄存器 (USB_TXICR)	624
23.5.2.42	USB 接收中断清除寄存器 (USB_RXICR)	625
23.5.2.43	USB 中断使能寄存器 (USB_IER)	626
23.5.2.44	USB 中断禁用寄存器 (USB_IDR)	628
23.5.2.45	USB 中断使能状态寄存器 (USB_IVS)	630
23.5.2.46	USB 原始中断事件标志寄存器 (USB_RIF)	632
23.5.2.47	USB 中断屏蔽标志寄存器 (USB_IFM)	634
23.5.2.48	USB 中断清除寄存器 (USB_ICR)	636
第 24 章	模数转换器 (ADC)	638
24.1	概述	638
24.2	特性	638
24.3	结构框图	639
24.4	功能描述	640

24.4.1	ADC 控制	640
24.4.2	ADC 时钟	640
24.4.3	通道控制	640
24.4.4	单次工作模式	641
24.4.5	连续工作模式	641
24.4.6	时序图	642
24.4.7	模拟看门狗	642
24.4.8	通道扫描	643
24.4.9	插入通道控制	643
24.4.10	不连续采样控制	643
24.4.11	数据对齐	645
24.4.12	可独自设置各通道采样时间	645
24.4.13	外部触发转换和触发极性	645
24.4.14	快速转换模式	646
24.4.15	数据管理	646
24.4.15.1	使用 DMA	646
24.4.15.2	在不使用 DMA 的情况下管理转换序列	646
24.4.15.3	在不使用 DMA 和溢出检测情况下进行转换	646
24.4.16	ADC 中断	647
24.5	特殊功能寄存器	648
24.5.1	寄存器列表	648
24.5.2	寄存器描述	649
24.5.2.1	ADC 状态寄存器 (ADC_STAT)	649
24.5.2.2	ADC 清零寄存器 (ADC_CLR)	650
24.5.2.3	ADC 控制寄存器 0 (ADC_CON0)	651
24.5.2.4	ADC 控制寄存器 1 (ADC_CON1)	653
24.5.2.5	ADC 采样时间寄存器 1 (ADC_SMPT1)	655
24.5.2.6	ADC 采样时间寄存器 2 (ADC_SMPT2)	656
24.5.2.7	ADC 采样时间寄存器 3 (ADC_SMPT3)	657
24.5.2.8	ADC 标准通道数据偏移寄存器 (ADC_NCHOFF)	658
24.5.2.9	ADC 插入通道数据偏移寄存器 1 (ADC_ICHOFF1)	659
24.5.2.10	ADC 插入通道数据偏移寄存器 2 (ADC_ICHOFF2)	659
24.5.2.11	ADC 插入通道数据偏移寄存器 3 (ADC_ICHOFF3)	660
24.5.2.12	ADC 插入通道数据偏移寄存器 4 (ADC_ICHOFF4)	660
24.5.2.13	ADC 标准通道序列寄存器 1 (ADC_NCHS1)	661
24.5.2.14	ADC 标准通道序列寄存器 2 (ADC_NCHS2)	662
24.5.2.15	ADC 标准通道序列寄存器 3 (ADC_NCHS3)	663
24.5.2.16	ADC 标准通道序列寄存器 4 (ADC_NCHS4)	664
24.5.2.17	ADC 插入通道序列寄存器 (ADC_ICHS)	665
24.5.2.18	ADC 通道序列长度寄存器 (ADC_CHSL)	666
24.5.2.19	ADC 看门狗高阈值寄存器 (ADC_WDTH)	667
24.5.2.20	ADC 看门狗低阈值寄存器 (ADC_WDTL)	667
24.5.2.21	ADC 插入通道数据寄存器 1 (ADC_ICHDR1)	667
24.5.2.22	ADC 插入通道数据寄存器 2 (ADC_ICHDR2)	668

	24.5.2.23	ADC 插入通道数据寄存器 3 (ADC_ICHDR3)	668
	24.5.2.24	ADC 插入通道数据寄存器 4 (ADC_ICHDR4)	668
	24.5.2.25	ADC 标准通道数据寄存器 (ADC_NCHDR)	669
	24.5.2.26	ADC 通用控制寄存器 (ADC_CCR)	669
第 25 章	调试控制 (DBGC)		671
25.1	概述		671
25.2	特性		671
25.3	结构框图		671
25.4	功能描述		672
	25.4.1	调试端口	672
	25.4.2	调试冻结	672
	25.4.3	调试复位	672
	25.4.4	MEM-AP 访问端口	673
25.5	特殊功能寄存器		674
	25.5.1	寄存器列表	674
	25.5.2	寄存器描述	674
	25.5.2.1	DBG 器件识别码 (DBG_IDCODE)	674
	25.5.2.2	APB1 外设调试冻结寄存器 (DBG_APB1FZ)	675
	25.5.2.3	APB2 外设调试冻结寄存器 (DBG_APB2FZ)	675
第 26 章	FLASH 信息区		677
26.1	概述		677
26.2	特性		677
26.3	功能描述		678
	26.3.1	FLASH 信息区只读信息	678
	26.3.1.1	芯片唯一码 UID	678
	26.3.1.2	芯片产品识别码 CHIPID	678
	26.3.2	FLASH 信息区配置信息	679
	26.3.2.1	芯片配置字 CFG_WORD	679
	26.3.2.2	写保护区域配置字 CFG_WRP	680
	26.3.2.3	数据区配置字 CFG_DAFLS	680
	26.3.2.4	用户程序校验码 CHKSUM	681
	26.3.2.5	全局读保护配置字 CFG_GBRDP	682
	26.3.2.6	私有读保护配置字 CFG_PCROP	682
第 27 章	修订历史		683

图目录

图 2-1	系统框图.....	30
图 3-1	DMA 多路复用器与 DMA 连接图	45
图 3-2	独立看门狗计数时钟	47
图 3-3	窗口看门狗计数时钟	47
图 4-1	系统总线矩阵	54
图 5-1	存储器控制结构图	57
图 7-1	电源结构框图	76
图 7-2	POR/PDR 示意图.....	77
图 7-3	BOR 示意图	78
图 7-4	LVD 示意图	78
图 7-5	低功耗模式转换图	79
图 8-1	复位结构图.....	88
图 9-1	时钟管理结构图.....	99
图 9-2	HOSC 电路图.....	100
图 9-3	晶体振荡器等效电路示意图	100
图 10-1	CSU 结构图.....	118
图 10-2	CSU 计数器操作	119
图 11-1	直接存储器访问控制器框图.....	135
图 12-1	PIS 结构框图.....	189
图 12-2	高电平调制输出波形图.....	194
图 12-3	低电平调制输出波形图.....	194
图 13-1	独立看门狗时序图	206
图 14-1	窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%)	214
图 14-2	错误的喂狗时序图 (WWDTWIN 设定为 25%)	215
图 15-1	GPIO 结构框图	222
图 15-2	外部中断 GPIO 映像	225
图 16-1	CRC 结构框图.....	244
图 17-1	预分频值计数时序图	253
图 17-2	重复计数器工作模式	254
图 17-3	采用内部时钟计数	255
图 17-4	I1 外部时钟连接	255
图 17-5	外部触发输入模块	256
图 17-6	ITn 外部时钟连接	257
图 17-7	计数器递增计数时序图.....	258
图 17-8	当 ARPEN=0 时计数器时序图	258
图 17-9	当 ARPEN=1 时计数器时序图	259
图 17-10	定时器递减计数时序图.....	260
图 17-11	增减计数器时序图	261
图 17-12	捕获/比较通道	261
图 17-13	捕获/比较通道 1 结构图	262
图 17-14	捕获/比较信道的输出部分	262
图 17-15	PWM 输入模式时序	264
图 17-16	边沿对齐 PWM 波形 (AR=8)	265

图 17-17	中心对齐 PWM 波形 (AR=8)	266
图 17-18	单脉冲模式	268
图 17-19	互补输出含死区时间插入	269
图 17-20	刹车输出行为	271
图 17-21	编码器接口模式下的计数操作	273
图 17-22	I1 滤波后极性反相时编码器接口例子	273
图 17-23	复位模式控制电路	275
图 17-24	门控模式控制电路	276
图 17-25	触发模式控制电路	276
图 17-26	外部时钟源 2+触发模式下的控制电路	277
图 17-27	6 步 PWM 波形示例	278
图 18-1	通用定时器结构框图	319
图 18-2	预分频值计数时序图	320
图 18-3	采用内部时钟计数	321
图 18-4	I1 外部时钟连接	321
图 18-5	外部触发输入模块	322
图 18-6	ITn 外部时钟连接	323
图 18-7	计数器递增计数时序图	324
图 18-8	当 ARPEN=0 时计数器时序图	325
图 18-9	当 ARPEN=1 时计数器时序图	326
图 18-10	计数器递减计数时序图	327
图 18-11	增减计数器时序图	328
图 18-12	捕获/比较通道	329
图 18-13	捕获/比较信道 1 主电路	329
图 18-14	捕获/比较通道的输出阶段	330
图 18-15	PWM 输入模式时序	331
图 18-16	边沿对齐 PWM 波形 (AR=8)	332
图 18-17	中心对齐 PWM 波形 (AR=0x3F)	333
图 18-18	单脉冲模式	335
图 18-19	编码器接口模式下的计数操作	337
图 18-20	滤波后极性反相时编码器接口	337
图 18-21	复位模式控制电路	338
图 18-22	门控模式控制电路	339
图 18-23	触发模式控制电路	340
图 18-24	外部时钟源 2+触发模式下的控制电路	341
图 20-1	I2C 结构图	387
图 20-2	START 和 STOP 信号	388
图 20-3	I2C 总线上的应答位	389
图 20-4	7 位地址格式	390
图 20-5	10 位地址格式	390
图 20-6	主机发送数据	391
图 20-7	主机接收数据	391
图 20-8	I2C 总线上的数据传输	392
图 20-9	从机初始化流程图	395

图 20-10	从机发送的传输序列图.....	396
图 20-11	从机接收的传输序列图.....	397
图 20-12	主机时钟产生	398
图 20-13	SCL 主机时钟同步	399
图 20-14	主机发送的传输序列图.....	400
图 20-15	主机接收的传输序列图.....	401
图 21-1	SPI 电路结构框图	436
图 21-2	SPI 通信时序格式	438
图 21-3	全双工通信.....	439
图 21-4	半双工通信.....	440
图 21-5	单工通信（主机模式下的只发送与从机模式下的只接收）	441
图 21-6	多从机通讯（一个主机和三个从机）	442
图 21-7	多主机通讯应用.....	443
图 21-8	全双工通信时序（直接存取操作模式，连续传输）	448
图 21-9	全双工通信时序（FIFO 缓存操作模式，连续传输）	449
图 21-10	单工通信-只发送模式时序（直接存取操作模式，连续传输）	450
图 21-11	单工通信-只发送模式时序（FIFO 缓存操作模式，连续传输）	451
图 21-12	单工通信-只接收模式时序（直接存取操作模式，连续传输）	452
图 21-13	单工通信-只接收模式时序（FIFO 缓存操作模式，连续传输）	453
图 21-14	发送时（BIDEN = 0 且 RXO = 0）的时序（间断传输）	454
图 21-15	使用 DMA 进行发送.....	456
图 21-16	使用 DMA 进行接收	457
图 21-17	TI 格式的 SPI 通信时序.....	464
图 21-18	I2S 电路结构框图.....	465
图 21-19	I2S Philips 标准波形（16 或 32 位的数据与通道帧，CKPOL = 0）	466
图 21-20	I2S Philips 标准波形（24 位数据帧，CKPOL = 0）	467
图 21-21	发送数据 0x123456.....	467
图 21-22	接收数据 0x123456.....	467
图 21-23	I2S Philips 标准波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）	467
图 21-24	16 位数据帧扩展到 32 位通道帧的示例	468
图 21-25	MSB 对齐协议波形（16 或 32 位的数据与通道帧，CKPOL = 0）	468
图 21-26	MSB 对齐协议波形（24 位数据帧，CKPOL = 0）	468
图 21-27	发送数据 0x123456.....	468
图 21-28	接收数据 0x123456.....	469
图 21-29	MSB 对齐协议波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）	469
图 21-30	16 位数据帧扩展到 32 位通道帧的示例	469
图 21-31	LSB 对齐协议波形（16 或 32 位的数据与通道帧，CKPOL = 0）	470
图 21-32	LSB 对齐协议波形（24 位数据帧，CKPOL = 0）	470
图 21-33	发送数据 0x123456.....	470
图 21-34	接收数据 0x123456.....	470
图 21-35	MSB 对齐协议波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）	471
图 21-36	16 位数据帧扩展到 32 位通道帧的示例	471
图 21-37	PCM 标准波形（16 或 32 位的数据与通道帧）	472
图 21-38	PCM 标准波形（24 位数据帧）	472

图 21-39	PCM 标准波形 (16 位数据帧扩展到 32 位通道帧)	472
图 21-40	音频采样频率定义	473
图 22-1	UART 框图	511
图 22-2	数据宽度设置	514
图 22-3	停止位配置	515
图 22-4	输入防抖动波形	516
图 22-5	防抖动电路模块输出	517
图 22-6	起始位侦测	517
图 22-7	数值采样	518
图 22-8	自动波特率侦测模式 0	523
图 22-9	自动波特率侦测模式 1	523
图 22-10	自动波特率侦测模式 2	524
图 22-11	自动流量控制框图	524
图 22-12	自动 RTSn 控制	525
图 22-13	自动 CTSn 控制	525
图 22-14	驱动开启, AADINV=0	525
图 22-15	使用地址标示侦测模式	527
图 22-16	LIN 模式侦测断路信号 (11 位断路长度-LBDL 位为 1)	529
图 22-17	LIN 模式侦测断路信号与帧错误信号	530
图 22-18	ISO 7816-3 异步协定	531
图 22-19	1.5 位停止位时检测校验错误	532
图 22-20	红外收发框图	533
图 22-21	IrDA 数据调制 (3/16) -正常模式	533
图 23-1	USB 结构图	566
图 24-1	ADC 结构框图	639
图 24-2	ADC 转换时序图	642
图 24-3	右对齐数据示意图	645
图 24-4	左对齐数据示意图	645
图 25-1	CJTAG 调试结构图	671
图 25-2	MEM-AP 地址映射	673

表目录

表 2-1	系统功能模块	32
表 2-2	32 位 E902 内核模块.....	32
表 2-3	存储器及存储接口	33
表 2-4	系统模块.....	33
表 2-5	时钟管理.....	34
表 2-6	外部接口.....	34
表 2-7	系统模块.....	34
表 2-8	定时器	36
表 2-9	通信模块.....	37
表 2-10	模拟模块.....	38
表 3-1	中断向量分配	41
表 3-2	STOP 低功耗模式的中断唤醒源	42
表 3-3	事件唤醒源.....	42
表 3-4	外设存储映射	44
表 3-5	DMA 请求列表.....	46
表 3-6	独立看门狗定时器的低功耗动作模式.....	47
表 3-7	窗口看门狗定时器的低功耗动作模式.....	47
表 3-8	HOSC 的低功耗动作模式.....	48
表 3-9	HRC4M 的低功耗动作模式	48
表 3-10	HRC48M 的低功耗动作模式.....	48
表 3-11	PLL 的低功耗动作模式.....	48
表 3-12	LRC 的低功耗动作模式.....	48
表 3-13	通用定时器的低功耗动作模式.....	49
表 3-14	基本定时器的低功耗动作模式.....	49
表 3-15	I2C 接口的低功耗动作模式.....	50
表 3-16	SPI 接口的低功耗动作模式.....	50
表 3-17	UART 的低功耗动作模式	51
表 3-18	ADC 转换通道配置.....	52
表 3-19	ADC 的低功耗动作模式	53
表 4-1	系统存储器映射.....	56
表 4-2	紧耦合存储器映射	56
表 5-1	写保护区配置字对应表.....	58
表 5-2	私有读保护区配置字对应表	58
表 5-3	数据 Flash 配置字对应表	58
表 5-4	不同全局保护级别下的访问限制表	59
表 7-1	低功耗模式说明.....	80
表 7-2	低功耗模式下各模块操作	81
表 8-1	系统复位与寄存器关系.....	89
表 12-1	生产端信号	190
表 12-2	消费端信号	191
表 12-3	消费端的 PIS 通道分配	192
表 15-1	端口配置表.....	224
表 15-2	端口驱动表.....	224

表 17-1	计数方向与编码器信号的关系.....	272
表 18-1	计数方向与编码器信号的关系.....	336
表 20-1	第一个字节中的位定义.....	390
表 20-2	$F_{I2CCLK} = 8 \text{ MHz}$ 的时序设置示例.....	402
表 20-3	$F_{I2CCLK} = 16 \text{ MHz}$ 的时序设置示例.....	402
表 20-4	$F_{I2CCLK} = 48 \text{ MHz}$ 的时序设置示例.....	403
表 20-5	$F_{I2CCLK} = 72 \text{ MHz}$ 的时序设置示例.....	403
表 20-6	SMBus 超时规格.....	405
表 20-7	各种 I2CCLK 频率的 TIMEOUTA 设置示例 ($T_{TIMEOUT} = 25 \text{ ms}$).....	407
表 20-8	各种 I2CCLK 频率的 TIMEOUTB 设置示例.....	407
表 20-9	各种 I2CCLK 频率的 TIMEOUTA 设置示例 ($T_{IDLE} = 50 \mu\text{s}$).....	407
表 21-1	SPI 特性.....	435
表 21-2	音频频率精度.....	474
表 22-1	UART1~4 具体功能配置.....	513
表 22-2	采样数据的噪音检测数值.....	519
表 22-3	系统时钟为 48MHz, 设置波特率时的误差计算.....	522
表 22-4	帧格式.....	526
表 22-5	中断配置表.....	536
表 23-1	端点特性.....	568
表 24-1	模拟看门狗通道选择.....	642
表 24-2	ADC 中断.....	647
表 25-1	CJTAG 端口描述.....	672

第1章 文档约定

1.1 寄存器读写权限的设定

缩写词	说明	描述
R/W	读/写 (__IO)	软件可以读写这些位
R	只读 (__I)	软件只能读取这些位
W	只写 (__O)	软件只能写入该位，读取该位时将返回复位值
W1	只写 (写 1)	软件只能写入该位，写 1 有效，写 0 无作用。
R/C_W1	读取/清零 (写 1)	软件可以读取该位，也可以通过写入 1 将该位清零。写入“0”对该位的值无影响
R/C_W0	读取/清零 (写 0)	软件可以读取该位，也可以通过写入 0 将该位清零。写入“1”对该位的值无影响
R/C_R	读取/清零 (读取)	软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响
C_W1	清零 (写 1)	通过写入 1 将该位清零。写入“0”对该位的值无影响
S_W1	置位 (写 1)	通过写入 1 将该位置位。写入“0”对该位的值无影响
C_W0	清零 (写 0)	通过写入 0 将该位清零。写入“1”对该位的值无影响
T_W1	触发 (写 1)	通过写入 1 将触发硬件动作。写入“0”对该位的值无影响
Reserved	保留	保留位，必须保持复位值。

第2章 系统概述

2.1 概述

本章节从系统层介绍 ES32VF2264 系列 MCU 所涵盖的功能模块。

2.2 系统框图

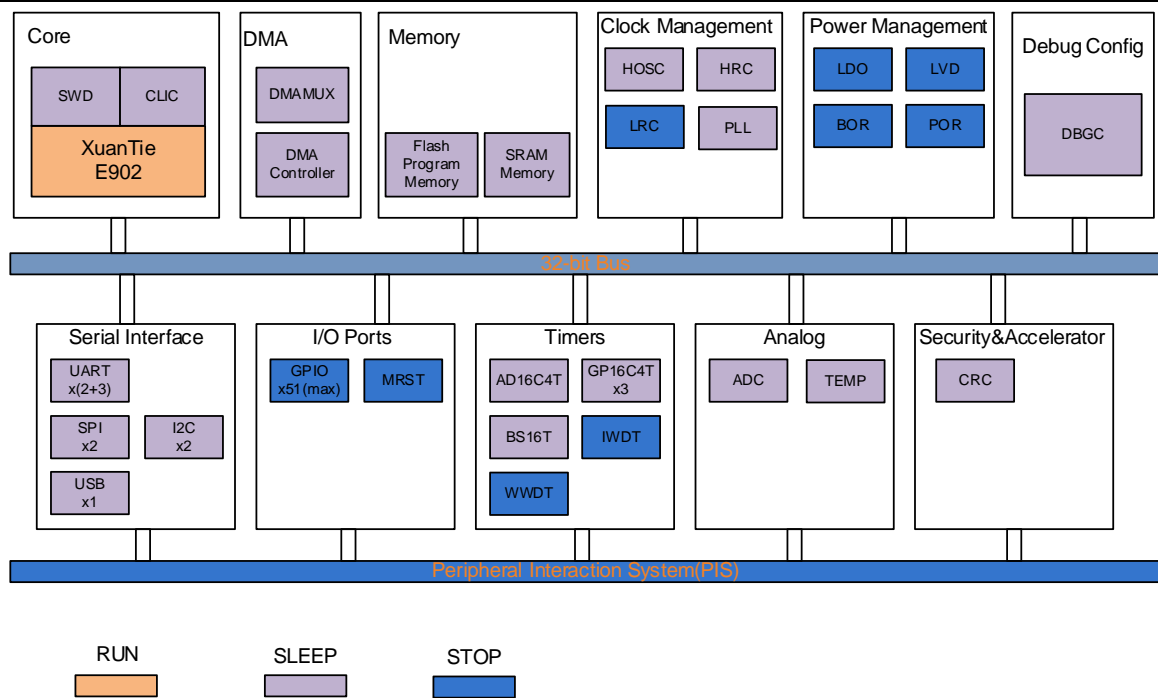


图 2-1 系统框图

2.3 模块功能类别

类别	描述
T-Head 32 位 XuanTie E902 CPU	<ul style="list-style-type: none"> ◆ RISC-V RV32EMC 精简指令结构 ◆ 32 位数据，16/32 位混合编码指令 ◆ 指令集: 47 条 RVE 指令，8 条 RVM 指令，27 条 RVC 指令（16 位指令）和 2 条 RV Privileged（特权）指令 ◆ 2 级顺序执行流水线：指令提取和指令译码、执行与回写。 ◆ 支持 RV 机器模式和可配置的普通用户模式 ◆ 支持硬件乘法器、硬件除法器，通过指令调用 ◆ 单周期指令和数据存储器访问 ◆ 支持 AHB-Lite 总线协议，支持可配置的指令总线 ◆ 支持可配置紧耦合 IP，包括系统计时器，矢量中断控制器等 ◆ Dhystone 性能测试可达 1.55 DMIPS/MHz
存储	<ul style="list-style-type: none"> ◆ 内部存储： <ul style="list-style-type: none"> ◇ FLASH 存储器 ◇ SRAM 存储器 ◆ 系统总线和存储器 ◆ 存储器系统控制
系统管理	<ul style="list-style-type: none"> ◆ 系统配置控制器 ◆ 电源管理及低功耗模式 <ul style="list-style-type: none"> ◇ 可支持多种低功耗模式： <ul style="list-style-type: none"> - SLEEP 模式 - STOP 模式 ◆ 复位控制 ◆ DMA <ul style="list-style-type: none"> ◇ 支持多个 DMA 请求通道，DMAMUX 为每个 DMA 通道提供片上 DMA 请求源选择 ◆ 外设互联（PIS） ◆ 看门狗定时器 <ul style="list-style-type: none"> ◇ 独立看门狗定时器 ◇ 窗口看门狗定时器 ◆ 调试配置控制器（DBGC）
时钟管理	<ul style="list-style-type: none"> ◆ 提供外部和内部多种时钟源选择 <ul style="list-style-type: none"> ◇ HOSC ◇ HRC4M ◇ HRC48M ◇ PLL48/64/72M ◇ LRC ◆ 外部时钟停振检测

类别	描述
	<ul style="list-style-type: none"> ◆ PLL 倍频满足高速应用 ◆ 可灵活选择内核，系统及外设时钟 ◆ 可灵活设置外设时钟门控及时钟分频，以满足低功耗应用需求
外部接口	◆ 通用 IO (GPIO)
安全管理及运算加速	◆ 循环冗余校验模块 (CRC)
定时器	<ul style="list-style-type: none"> ◆ 高级定时器 (AD16C4T) ◆ 通用定时器 (GP16C4T0~2) ◆ 基本定时器 (BS16T)
通信	<ul style="list-style-type: none"> ◆ I2C 总线接口 (I2C) ◆ 串行外设接口 (SPI) ◆ 通用异步收发器 (UART) ◆ 通用串行总线接口 (USB)
模拟	<ul style="list-style-type: none"> ◆ 模数转换 (ADC) ◆ 温度传感器 (TSENSE)

表 2-1 系统功能模块

2.3.1 32 位E902 内核模块

ES32VF2264 微控制器内核模块包含以下功能：

模块	描述
32 位 E902 内核	<ul style="list-style-type: none"> ◆ RISC-V RV32EMC 精简指令结构 ◆ 32 位寄存器，16/32 位混合编码指令 ◆ 32 位硬件乘法器、除法器； ◆ 高效可靠的中断响应； ◆ 支持使用 WFI,WFE 进入低功耗模式
CLIC	<ul style="list-style-type: none"> ◆ 中断使能控制； ◆ 中断优先级设置； ◆ 支持中断嵌套； ◆ 支持 32 个外部中断向量
调试接口	◆ CJTAG 协议调试接口

表 2-2 32 位 E902 内核模块

2.3.2 存储器及存储器接口

ES32VF2264 微控制器包含以下存储器及存储器接口模块:

模块	描述
系统总线 and 存储器	◆ 系统总线架构, 存储器地址空间映射
存储器系统控制	◆ FLASH 存储器的访问控制
FLASH	◆ FLASH 存储器
SRAM	◆ SRAM 存储器

表 2-3 存储器及存储接口

2.3.3 系统模块

ES32VF2264 微控制器包含以下系统模块:

模块	描述
系统配置控制器 (SYSCFG)	◆ 系统的相关配置
电源管理及低功耗模式 (PMU)	◆ 系统电源的管理及低功耗模式控制
复位控制 (RMU)	◆ 系统所有复位的管理
DMA 多路复用 (DMAMUX)	◆ DMA 请求源的多路复用选择器
DMA 控制器 (DMAC)	◆ DMA 控制器可减少 CPU 负荷, 提高系统效率 ◆ 在低功耗场合也可代替 CPU 的部分工作而不必唤醒整个系统, 节省功耗
外设互联 (PIS)	◆ 外设互联系统为外设提供互联接口, 可减少软件负担, 提高了系统响应的及时性, 同时为扩展应用场景提供了便利和灵活性
看门狗定时器 (IWDT, WWDT)	◆ 包含了独立看门狗和窗口看门狗
调试配置控制器 (DBGMC)	◆ 系统调试相关的配置控制 ◆ 产生断点时, 定时器/WDT 是否继续计数可以配置 ◆ 产生断点时, PWM 端口是否为高阻态可以配置

表 2-4 系统模块

2.3.4 时钟管理

ES32VF2264 微控制器包含以下时钟管理模块：

模块	描述
时钟管理 (CMU)	<ul style="list-style-type: none"> ◆ 时钟源配置 ◆ 系统和外设时钟的选择及切换 ◆ 外部时钟停振检测 (时钟安全机制) ◆ 外设时钟门控 ◆ 系统和外设时钟分频 ◆ PLL 倍频

表 2-5 时钟管理

2.3.5 外部接口

ES32VF2264 微控制器包含以下外部接口模块：

模块	描述
通用 IO 及端口控制	<ul style="list-style-type: none"> ◆ 通用 IO 的输入输出功能 ◆ 对 IO 的控制还包括：上、下拉电阻选择，开漏选择，驱动能力选择，端口 CMOS/TTL 输入功能选择，端口模拟滤波器使能控制等 ◆ 通用 IO 端口支持 16 个中断源 (分为 4 组，对应 4 个中断向量号) 和 DMA 请求

表 2-6 外部接口

2.3.6 安全管理及运算加速

ES32VF2264 微控制器包含以下安全管理模块：

模块	描述
循环冗余校验模块 (CRC)	<ul style="list-style-type: none"> ◆ 支持四个常用的多项式： <ul style="list-style-type: none"> ◇ CRC-CCITT ◇ CRC-8 ◇ CRC-16 ◇ CRC-32

表 2-7 系统模块

2.3.7 定时器

ES32VF2264 微控制器包含以下定时器模块：

模块	描述
高级定时器 (AD16C4T)	<ul style="list-style-type: none"> ◆ 16 位递增、递减、递增/递减自动重载计数器 ◆ 16 位可编程预分频器，用于对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间 ◆ 多达 4 个独立通道，可用于： <ul style="list-style-type: none"> ◇ 输入捕获 ◇ 输出比较 ◇ PWM 生成（边沿和中心对齐模式） ◇ 单脉冲模式输出 ◇ 通道 1、2、3 可各自输出 1 组互补 PWM ◆ 3 组带可编程死区的互补输出 (CH1/CH1n、CH2/CH2n、CH3/CH3n) ◆ 使用外部信号控制定时器且可实现多个定时器互连的同步电路 ◆ 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器 ◆ 刹车功能，可将定时器的输出信号置于复位状态或已知状态 ◆ 以下事件将生成中断请求： <ul style="list-style-type: none"> ◇ 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发） ◇ 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数） ◇ 输入捕获 ◇ 输出比较 ◇ 刹车输入 ◆ 支持定位用增量（正交）编码器和霍尔传感器电路 ◆ 外部时钟触发输入或逐周期电流管理 ◆ 该定时器可与其他定时器连接，一起配合使用，以达到同步或事件串联的目的 ◆ 在调试模式下，定时器可被冻结
通用定时器 (GP16C4T)	<ul style="list-style-type: none"> ◆ 16 位递增、递减、递增/递减自动重载计数器 ◆ 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 到 65536 之间

模块	描述
	<ul style="list-style-type: none"> ◆ 多达 4 个独立通道，可用于： <ul style="list-style-type: none"> ◇ 输入捕获 ◇ 输出比较 ◇ PWM 生成（边沿和中心对齐模式） ◇ 单脉冲模式输出 ◆ 使用外部信号控制定时器且可实现多个定时器互联的同步电路 ◆ 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器 ◆ 发生如下事件时生成中断/DMA 请求： <ul style="list-style-type: none"> ◇ 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发） ◇ 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数） ◇ 输入捕获 ◇ 输出比较 ◆ 支持定位用增量（正交）编码器和霍尔传感器电路 ◆ 外部时钟触发输入 ◆ 在调试模式下，定时器可被冻结
基本定时器（BS16T）	<ul style="list-style-type: none"> ◆ 16 位递增自动重载计数器 ◆ 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数 介于 1 到 65536 之间 ◆ 发生如下事件时生成中断/DMA 请求： <ul style="list-style-type: none"> ◇ 更新：计数器上溢 ◆ 在调试模式下，定时器可被冻结

表 2-8 定时器

2.3.8 通信模块

ES32VF2264 微控制器包含以下通信模块：

模块	描述
I2C 总线接口 (I2C)	<ul style="list-style-type: none"> ◆ 支持多主模式和总线仲裁 ◆ 可编程 I2C 地址检测 ◆ 最高通信速率为 400 KHz ◆ 可配置时钟延长 ◆ 兼容 SMBus2.0 协议 ◆ 兼容 PMBus 协议
串行外设接口 (SPI)	<ul style="list-style-type: none"> ◆ 支持半双工/全双工的同步串行通信 ◆ 主模式或从模式操作 ◆ 8 位或 16 位传输帧格式选择
通用异步收发器 (UART)	<ul style="list-style-type: none"> ◆ 支持与外部设备进行全双工数据通信和半双工单线通信 ◆ 提供小数波特率发生器可灵活配置多种波特率 ◆ 智能卡协议 (ISO 7816) ◆ 支持多点通信 ◆ 红外通信协议 ◆ 硬件自动流控制 (CTS/RTS)
通用串行总线 (USB)	<ul style="list-style-type: none"> ◆ 支持 USB2.0 协议 <ul style="list-style-type: none"> ◇ 支持 USB2.0 全速从设备 ◇ 支持 USB2.0 全速 OTG/主机/从设备 ◆ 支持 SRP/HNP ◆ 支持 9 个独立端点 ◆ 动态 512Byte FIFO

表 2-9 通信模块

2.3.9 模拟模块

ES32VF2264 微控制器包含以下模拟模块：

模块	描述
模数转换 (ADC)	12 位逐次逼近型模数转换器 <ul style="list-style-type: none"> ◆ 可配置的转换分辨率 (6/8/10/12 位) ◆ 在标准转换、插入转换结束后以及发生模拟看门狗或溢出事件时产生中断 ◆ 支持单次或连续转换模式 ◆ 用于自动将通道 0 转换为通道 “n” 的扫描模式 ◆ 可配置的数据对齐方式 ◆ 可独立设置各通道采样时间 ◆ 可配置外部触发器选项，可为标准转换和插入转换配置极性 ◆ 支持不连续采样模式 ◆ 可配置的参考源选择 ◆ 可配置的转换时钟分频 ◆ 标准通道转换期间可产生 DMA 请求
温感 (TS)	<ul style="list-style-type: none"> ◆ 温感电压直接连到 ADC 转换通道 17

表 2-10 模拟模块

第3章 芯片配置指引

3.1 概述

本章节主要说明以下内容：

- ◆ 芯片顶层相关模块的连接及信号路径
- ◆ 阅读各模块时可参考的相关信息链接
- ◆ 芯片顶层连接资源的相关配置
- ◆ 模块之间特殊的交互

3.2 E902 内核配置

E902 提供了高性能、低功耗、低成本的平台来满足 MCU 的实现要求，具备出色的计算性能，并能够快速响应中断。

3.2.1 E902 内核

关于 E902 内核技术细节可参考相关技术文档。

3.2.2 总线

E902 内核提供 1 条系统总线和 1 条指令总线，两条总线都支持 AMBA3.0 AHB-Lite 协议。

3.2.3 系统计数器

64 位系统计数器，计数时钟为内核时钟。具体配置使用可参考相关说明。

3.2.4 调试器件

支持标准 CJTAG 协议的调试接口。

3.3 核内局部中断控制器

核内局部中断控制器（以下简称 CLIC），仅用于对中断源进行采样，优先级仲裁和分发。CLIC 仲裁来源包括处理器各个模式下触发的中断。E902 实现的 CLIC 单元基本功能如下：

- ◇ 支持 RISC-V 标准 CLIC SPEC-0.8 版本；
- ◇ 最多支持 32 个外部中断源可配，支持电平中断，脉冲中断，加上兼容 CLINT 的至多 16 个中断（目前仅实现机器模式软件中断、机器模式计时器中断与机器模式外部中断），CLIC 共支持 48 个中断处理；
- ◇ 中断优先级有效位 CLICINTCTLBITS 2~5 可配，最多 32 个级别的中断优先级；
- ◇ 每个中断目标拥有 4 个 memory-mapped 的控制寄存器；
- ◇ 通过写相应中断源的控制寄存器可以配置此中断源的各个属性；
- ◇ 支持了可选的 MSCRATCHCSW，MSCRATCHCSWL 寄存器，供中断处理函数内快速交换栈指针使用。

3.3.1 中断优先级

可参考 E902 技术文档。

3.3.2 中断向量分配

中断向量分配如下表所示：

向量号	优先级	名称	中断标识	说明
3	HIGHEST	调试断点异常	0	
1		取指令访问错误异常	0	
2		非法指令异常	0	
8		用户模式环境调用异常	0	
11		机器模式环境调用异常	0	
6		存储指令非对齐访问异常	0	
4		加载指令非对齐访问异常	0	
7		存储指令访问错误异常	0	
5	LOWEST	加载指令访问错误异常	0	
0-2	可配置	未实现/保留	1	
3	可配置	机器模式软件中断	1	
4-6	可配置	未实现/保留	1	
7	可配置	机器模式计时器中断	1	
8-10	可配置	未实现/保留	1	
11	可配置	机器模式外部中断	1	
12-15	可配置	未实现/保留	1	
16	可配置	WWDT	1	WWDT 全局
17	可配置	IWDT	1	IWDT 全局
18	可配置	LVD	1	LVD 全局

19	可配置	CSU	1	CSU 全局
20	可配置	CMU	1	CMU 全局
21	可配置	EXTI0_3	1	外部端口中断 0~3
22	可配置	EXTI4_7	1	外部端口中断 4~7
23	可配置	EXTI8_11	1	外部端口中断 8~11
24	可配置	EXTI12_15	1	外部端口中断 12~15
25	可配置	DMA	1	DMA 全局
26	可配置	—	1	—
27	可配置	USB	1	USB 全局
28	可配置	ADC	1	ADC 全局
29	可配置	AD16C4T_UP_TRIG_COM	1	AD16C4T0 更新、触发和 COM 中断
30	可配置	AD16C4T_CC	1	AD16C4T0 捕捉和比较中断
31	可配置	BS16T	1	BS16T 全局
32	可配置	—	1	—
33	可配置	GP16C4T0	1	GP16C4T0 全局
34	可配置	GP16C4T1	1	GP16C4T1 全局
35	可配置	GP16C4T2	1	GP16C4T2 全局
36	可配置	—	1	—
37	可配置	AD16C4T	1	AD16C4T 全局
38	可配置	—	1	—
39	可配置	I2C0	1	I2C0 全局
40	可配置	I2C1	1	I2C1 全局
41	可配置	SPI0	1	SPI0 全局
42	可配置	SPI1	1	SPI1 全局
43	可配置	EUART0	1	UART0 全局
44	可配置	EUART1	1	UART1 全局
45	可配置	CUART0	1	UART2 全局
46	可配置	CUART1	1	UART3 全局
47	可配置	CUART2	1	UART4 全局

表 3-1 中断向量分配

3.4 异步唤醒中断和事件

3.4.1 异步中断唤醒源

在芯片 STOP 低功耗模式下的中断唤醒源如下表：

唤醒源	描述
外部端口中断	外部端口输入上升沿或下降沿中断
LVD 中断	LVD 有效边沿或电平产生的中断可唤醒芯片
独立看门狗中断	使用 LRC 计数时，中断可唤醒芯片
窗口看门狗中断	使用 LRC 计数时，中断可唤醒芯片
复位	系统复位（不包含软件复位）

表 3-2 STOP 低功耗模式的中断唤醒源

3.4.2 事件唤醒

ES32VF2264 微控制器支持事件唤醒机制。芯片唤醒后，软件需要清除相应外设的中断标志位和外设在 CLIC 中断通道上的挂起位。芯片 STOP 模式下的事件唤醒源如下表所示：

事件唤醒源	描述
EXTI0	选择 PA0,PB0, PC0, PD0 之一作为唤醒源
EXTI1	选择 PA1,PB1, PC1, PD1 之一作为唤醒源
EXTI2	选择 PA2,PB2, PC2, PD2 之一作为唤醒源
EXTI3	选择 PA3,PB3, PC3 之一作为唤醒源
EXTI4	选择 PA4,PB4, PC4 之一作为唤醒源
EXTI5	选择 PA5,PB5, PC5 之一作为唤醒源
EXTI6	选择 PA6,PB6, PC6 之一作为唤醒源
EXTI7	选择 PA7,PB7, PC7 之一作为唤醒源
EXTI8	选择 PA8,PB8, PC8 之一作为唤醒源
EXTI9	选择 PA9,PB9, PC9 之一作为唤醒源
EXTI10	选择 PA10,PB10, PC10 之一作为唤醒源
EXTI11	选择 PA11,PB11, PC11 之一作为唤醒源
EXTI12	选择 PA12,PB12, PC12 之一作为唤醒源
EXTI13	选择 PA13,PB13, PC13 之一作为唤醒源
EXTI14	选择 PA14,PB14, PC14 之一作为唤醒源
EXTI15	选择 PA15,PB15, PC15 之一作为唤醒源
IWDT	独立看门狗中断事件
WWDT	窗口看门狗中断事件
LVD	LVD 中断事件

表 3-3 事件唤醒源

3.5 存储器及存储器接口

3.5.1 系统总线和存储器

ES32VF2264 系列产品外设存储映射如下表所示:

总线	边界地址	外设
APB	0x4000 0000 - 0x4000 03FF	AD16C4T
	0x4000 0400 - 0x4000 07FF	BS16T
	0x4000 0800 - 0x4000 0BFF	GP16C4T0
	0x4000 0C00 - 0x4000 0FFF	GP16C4T1
	0x4000 1000 - 0x4000 13FF	GP16C4T2
	0x4000 1400 - 0x4000 17FF	—
	0x4000 1800 - 0x4000 1BFF	—
	0x4000 1C00 - 0x4000 1FFF	—
	0x4000 2000 - 0x4000 3FFF	—
	0x4000 4000 - 0x4000 43FF	EUART0
	0x4000 4400 - 0x4000 47FF	EUART1
	0x4000 4800 - 0x4000 4BFF	—
	0x4000 4C00 - 0x4000 4FFF	—
	0x4000 5000 - 0x4000 53FF	CUART0
	0x4000 5400 - 0x4000 57FF	CUART1
	0x4000 5800 - 0x4000 5FFF	CUART2
	0x4000 6000 - 0x4000 63FF	SPI0
	0x4000 6400 - 0x4000 67FF	SPI1
	0x4000 6800 - 0x4000 6BFF	—
	0x4000 6C00 - 0x4000 7FFF	—
	0x4000 8000 - 0x4000 83FF	I2C0
	0x4000 8400 - 0x4000 87FF	I2C1
	0x4000 8800 - 0x4000 8BFF	WWDT
	0x4000 8C00 - 0x4000 8FFF	IWDT
	0x4000 9000 - 0x4000 93FF	DBGC
	0x4000 9400 - 0x4000 97FF	ADC
	0x4000 9800 - 0x4000 AFFF	—
	0x4000 B000 - 0x4000 B3FF	—
	0x4000 B400 - 0x4000 CFFF	—
	0x4000 D000 - 0x4000 D3FF	DMA_MUX
0x4000 D400 - 0x4007 FFFF	—	
AHB	0x4008 0000 - 0x4008 03FF	SYSCFG
	0x4008 0400 - 0x4008 07FF	CMU
	0x4008 0800 - 0x4008 0BFF	RMU
	0x4008 0C00 - 0x4008 0FFF	PMU
	0x4008 1000 - 0x4008 13FF	MSC

总线	边界地址	外设
	0x4008 1400 - 0x4008 17FF	—
	0x4008 1800 - 0x4008 3FFF	—
	0x4008 4000 - 0x4008 4FFF	GPIO
	0x4008 5000 - 0x4008 53FF	CRC
	0x4008 5400 - 0x4008 57FF	DMA
	0x4008 5800 - 0x4008 5BFF	—
	0x4008 5C00 - 0x4008 5FFF	CSU
	0x4008 6000 - 0x4008 63FF	PIS
	0x4008 6400 - 0x4008 67FF	USB
	0x4008 6800 - 0x4008 FFFF	—

表 3-4 外设存储映射

3.6 系统模块配置

3.6.1 DMA控制器配置

DMA 控制器包含 7 个通道，每个 DMA 通道对应一个 DMA 多路复用器，每个多路复用器包含了微控制器所有的 DMA 申请源，由 DMA_CHx_SELCON (x=0~6) 配置选择。多路复用器和 DMA 之间连接图如下：

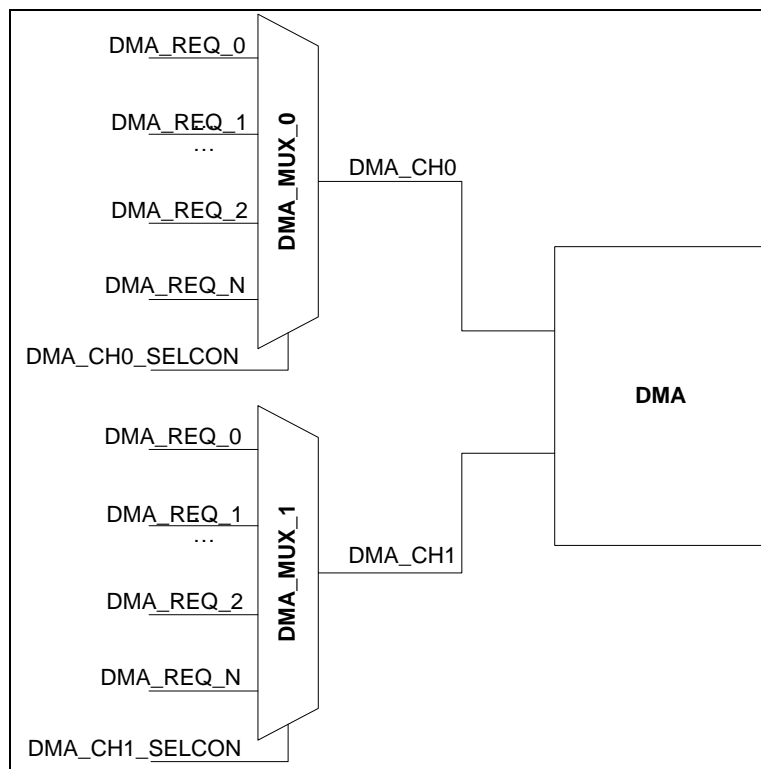


图 3-1 DMA 多路复用器与 DMA 连接图

每个 DMA 多路复用器可选择的 DMA 请求如下表所示：

模块名	DMA 请求源
GPIO	EXTI0~EXTI15
ADC	ADC 转换结束
AD16C4T	AD16C4T_CH1
	AD16C4T_CH2
	AD16C4T_CH3
	AD16C4T_CH4
	AD16C4T_TRIG
	AD16C4T_COM
	AD16C4T_UP
BS16T	BS16T_UP
GP16C4T0	GP16C4T0_CH1
	GP16C4T0_CH2

模块名	DMA 请求源
	GP16C4T0_CH3
	GP16C4T0_CH4
	GP16C4T0_TRIG
	GP16C4T0_UP
GP16C4T1	GP16C4T1_CH1
	GP16C4T1_CH2
	GP16C4T1_CH3
	GP16C4T1_CH4
	GP16C4T1_TRIG
	GP16C4T1_UP
GP16C4T2	GP16C4T2_CH1
	GP16C4T2_CH2
	GP16C4T2_CH3
	GP16C4T2_CH4
	GP16C4T2_TRIG
	GP16C4T2_UP
UART0	UART0_TX
	UART0_RX
UART1	UART1_TX
	UART1_RX
UART2	UART2_TX
	UART2_RX
UART3	UART3_TX
	UART3_RX
UART4	UART4_TX
	UART4_RX
SPI0	SPI0_TX
	SPI0_RX
SPI1	SPI1_TX
	SPI1_RX
I2C0	I2C0_TX
	I2C0_RX
I2C1	I2C1_TX
	I2C1_RX
CRC	CRC
PIS	PIS 通道 0~7

表 3-5 DMA 请求列表

3.6.2 独立看门狗定时器配置

3.6.2.1 独立看门狗定时器的时钟

下图为独立看门狗定时器的计数时钟源选择：

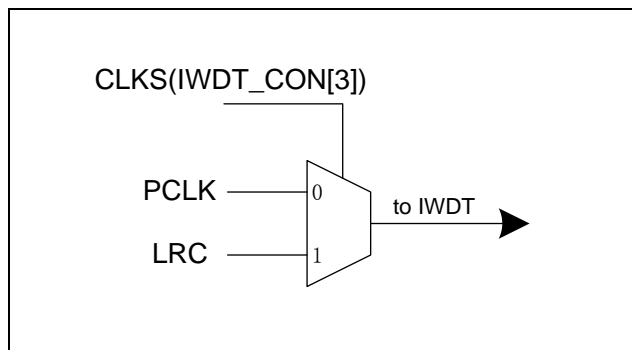


图 3-2 独立看门狗计数时钟

3.6.2.2 独立看门狗定时器的低功耗动作模式

低功耗模式	模块工作模式
STOP	可工作

表 3-6 独立看门狗定时器的低功耗动作模式

3.6.3 窗口看门狗定时器配置

3.6.3.1 窗口看门狗定时器的时钟

下图为窗口看门狗定时器的计数时钟源选择：

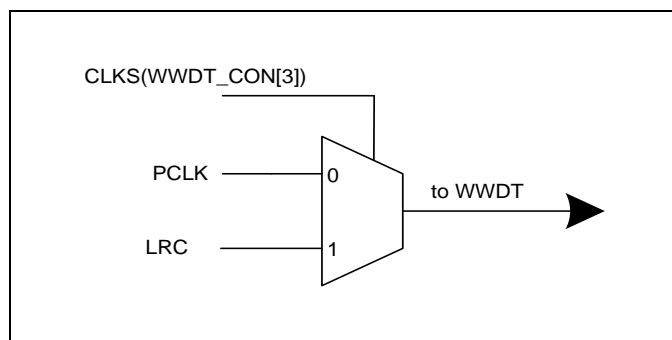


图 3-3 窗口看门狗计数时钟

3.6.3.2 窗口看门狗定时器的低功耗动作模式

低功耗模式	模块工作模式
STOP	可工作

表 3-7 窗口看门狗定时器的低功耗动作模式

3.6.4 时钟管理配置

3.6.4.1 HOSC的低功耗动作模式

低功耗模式	模块工作模式
RUN,SLEEP	可工作
STOP	关闭

表 3-8 HOSC 的低功耗动作模式

3.6.4.2 HRC4M的低功耗动作模式

低功耗模式	模块工作模式
RUN,SLEEP	可工作
STOP	关闭

表 3-9 HRC4M 的低功耗动作模式

3.6.4.3 HRC48M的低功耗动作模式

低功耗模式	模块工作模式
RUN,SLEEP	可工作
STOP	关闭

表 3-10 HRC48M 的低功耗动作模式

3.6.4.4 PLL的低功耗动作模式

低功耗模式	模块工作模式
RUN,SLEEP	可工作
STOP	关闭

表 3-11 PLL 的低功耗动作模式

3.6.4.5 LRC的低功耗动作模式

低功耗模式	模块工作模式
RUN,SLEEP	工作
STOP	工作

表 3-12 LRC 的低功耗动作模式

3.7 外部接口配置

3.7.1 通用IO及端口控制配置

3.7.1.1 端口特殊配置说明

SWDIO 和 SWCLK 默认使用 CMOS 输入，支持 3.3/5V 输入系统。正常量产时若不复用其他功能，请将调试口配置为 FUNC1（GPIO 模式），避免输入悬空。

3.8 定时器配置

3.8.1 通用定时器

3.8.1.1 通用定时器例化说明

ES32VF2264 系列 MCU 中,GP16C4T0,GP16C4T1,GP16C4T2 为 4 通道通用定时器。
AD16C4T 为 4 通道高级定时器。

3.8.1.2 通用定时器的时钟

通用定时器的总线时钟和模块时钟源为 PCLK。

3.8.1.3 通用定时器的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	不工作

表 3-13 通用定时器的低功耗动作模式

3.8.2 基本定时器配置

3.8.2.1 基本定时器例化说明

ES32VF2264 系列 MCU 中,BS16T 为基本定时器。

3.8.2.2 基本定时器的时钟

基本定时器的总线时钟和模块时钟源为 PCLK。

3.8.2.3 基本定时器的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	不工作

表 3-14 基本定时器的低功耗动作模式

3.9 通信配置

3.9.1.1 I2C接口的时钟

I2C 的总线时钟和模块时钟源为 PCLK。

3.9.1.2 I2C接口的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	不工作

表 3-15 I2C 接口的低功耗动作模式

3.9.2 串行外设接口 (SPI) 配置

3.9.2.1 串行外设接口 (SPI) 的时钟

SPI 的总线时钟和模块时钟源为 PCLK。

3.9.2.2 串行外设接口 (SPI) 的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	不工作

表 3-16 SPI 接口的低功耗动作模式

3.9.3 通用异步收发器 (UART)

3.9.3.1 通用异步收发器 (UART) 的时钟

UART 的总线时钟和模块时钟源为 PCLK。

3.9.3.2 通用异步收发器 (UART) 的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	不工作

表 3-17 UART 的低功耗动作模式

3. 10 模拟配置

3. 10. 1 ADC控制配置

3. 10. 1. 1 ADC模块例化

ES32VF2264 包含 1 路 ADC。

3. 10. 1. 2 ADC转换通道配置

ADC 支持 20 个通道选择（其中包含 17 个外部通道），具体分配如下表所示，每个 ADC 通道在管脚上的对应关系，请参考数据手册的管脚功能定义表格。

寄存器 ADC_CON0 的 AWDCH	ADC 通道	信号分配
00000	ADC 通道 0	ADC_IN0
00001	ADC 通道 1	ADC_IN1
00010	ADC 通道 2	ADC_IN2
00011	ADC 通道 3	ADC_IN3
00100	ADC 通道 4	ADC_IN4
00101	ADC 通道 5	ADC_IN5
00110	ADC 通道 6	ADC_IN6
00111	ADC 通道 7	ADC_IN7
01000	ADC 通道 8	ADC_IN8
01001	ADC 通道 9	ADC_IN9
01010	ADC 通道 10	ADC_IN10
01011	ADC 通道 11	ADC_IN11
01100	ADC 通道 12	ADC_IN12
01101	ADC 通道 13	ADC_IN13
01110	ADC 通道 14	ADC_IN14
01111	ADC 通道 15	ADC_IN15
10000	ADC 通道 16	ADC_IN16
10001	ADC 通道 17	温感电压
10010	ADC 通道 18	VDD/4 电压
10011	ADC 通道 19	VREF1.2V
保留	—	—

表 3-18 ADC 转换通道配置

3. 10. 1. 3 ADC电源及参考电压

ADC 电源由 VDD 提供。

ADC 的参考电压可选 VDD 或 VREFP（VREFP 由外部端口输入）。

3. 10. 1. 4 ADC的时钟

ADC 的总线时钟和模块时钟源为 PCLK。

3. 10. 1. 5 ADC的低功耗动作模式

低功耗模式	模块工作模式
SLEEP	可工作
STOP	关闭

表 3-19 ADC 的低功耗动作模式

第4章 系统总线和存储器

4.1 概述

主系统由 32 位多层 AHB 总线矩阵构成，可实现以下部分的互连。

3 条主控总线：

- ◆ E902 内核系统总线 SBus
- ◆ E902 内核指令总线 IBus
- ◆ DMA 存储器总线

4 条被控总线：

- ◆ 内部 Flash 总线
- ◆ 内部 SRAM 总线
- ◆ AHB 外设
- ◆ APB 外设

借助总线矩阵，可以实现主控总线到被控总线的访问，这样即使在多个高速外设同时运行期间，系统也可以实现并发访问和高效运行。

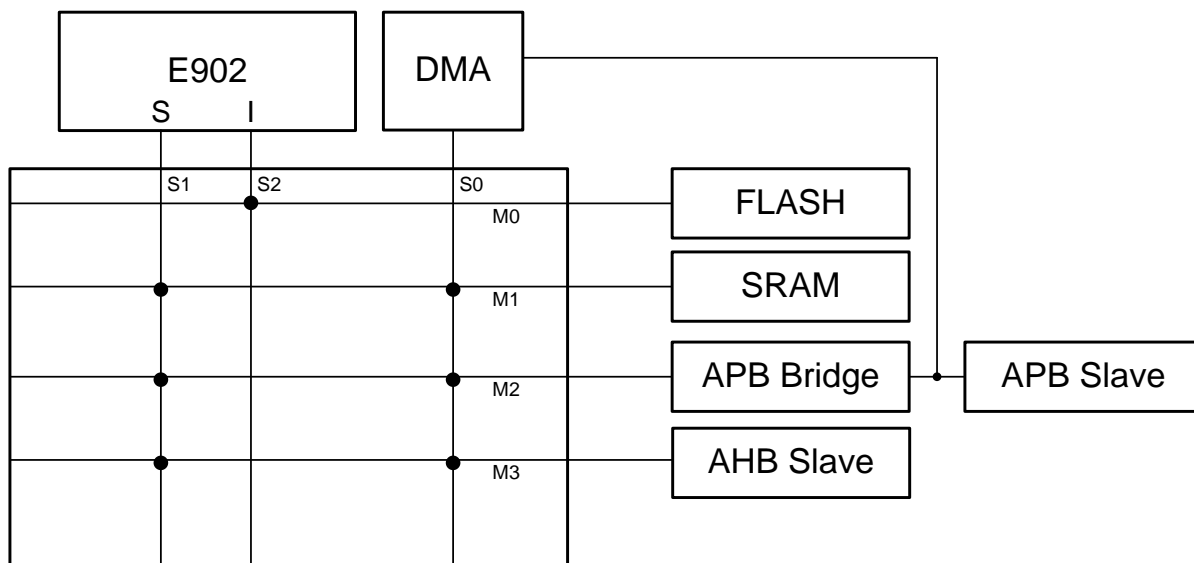


图 4-1 系统总线矩阵

4.2 系统总线

4.2.1 S0: DMA总线

此总线用于将 DMA 存储器总线主接口连接到总线矩阵。DMA 通过此总线访问外设或执行存储器间的数据传输。此总线访问的对象是 AHB 和 APB 外设以及内部 SRAM。

4.2.2 S1: S总线

此总线用于将 E902 内核的系统总线连接到总线矩阵。此总线用于访问位于存储器 SRAM 中的程序和数据，以及系统控制寄存器和其他外设寄存器空间。

4.2.3 S2: I总线

此总线用于将 E902 内核的指令总线连接到总线矩阵。此总线用于访问位于存储器 FLASH 中的程序。

4.2.4 总线矩阵

总线矩阵用于主控总线之间的访问仲裁管理。仲裁采用循环调度算法。

4.2.5 AHB/APB总线桥

借助两个 AHB/APB 总线桥可实现 AHB 总线与两个 APB 总线之间的桥接，从而可灵活配置外设频率。

4.3 存储器的组织结构

程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个顺序的 4 GB 地址空间内。

各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

有关外设寄存器映射的详细信息，请参见相关章节。

可寻址的存储空间分为 8 个主要块，每个块为 512MB。

未分配给片上存储器和外设的所有存储区域均视为“保留区”。请参见产品数据手册中的存储器映射图。

4.3.1 系统存储器映射

系统存储器映射图如下表所示：

地址范围	存储	备注
0x0000_0000~0x0003_FFFF	FLASH 主存储区	256KB
0x0004_0000~0x0004_0400	FLASH 信息区	1KB
0x0005_0000~0x1FFF_FFFF	Reserved	
0x2000_0000~0x2000_7FFF	SRAM	32KB
0x2000_8000~0x3FFF_FFFF	Reserved	
0x4000_0000~0x4007_FFFF	APB 外设	
0x4008_0000~0x4008_FFFF	AHB 外设	

地址范围	存储	备注
0x4009_0000~0x5FFF_FFFF	Reserved	
0x6000_0000~0x9FFF_FFFF	Reserved	
0xA000_0000~0xDFFF_FFFF	Reserved	
0xE000_0000~0xE00F_FFFF	私有外设	
0xE010_0000~0xFFFF_FFFF	Reserved	

表 4-1 系统存储器映射

4.3.2 FLASH存储器映射

FLASH 存储器支持至少 10 万次擦写次数，10 年以上的数据保持时间。支持通过编程器或 IAP 方式进行 FLASH 擦除和编程，无论是否编写相同的数据，在 FLASH 编程前均必须先进行擦除。

Flash 接口管理 CPU 通过系统总线对 Flash 进行访问。该接口可针对 Flash 执行擦除和编程操作，并实施读写保护机制。

FLASH 存储器结构如下：

- ◇ 主存储区（基地址 0x0000_0000）共 256K Bytes，分为 512 页，每页 512 Bytes。
- ◇ 芯片配置字存放在信息区（基地址 0x0004_0000）中，用于配置读写保护、BOR 电压、软件/硬件看门狗、启动地址以及安全保护设置等。

4.3.3 SRAM存储器映射

SRAM 容量为 32K Bytes，地址为 0x2000_0000~0x2000_7FFF，72MHz 主频下可单周期访问。

4.3.4 外设存储映射

ES32VF2264 产品外设存储映射请参考章节“芯片配置指引”的外设存储映射表。

4.3.5 紧耦合总线映射

地址范围	紧耦合外设	备注
0xE000_0000~0xEFFF_FFFF	E902 紧耦合外设	CLIC

表 4-2 紧耦合存储器映射

4.4 启动引导

芯片发生上电或复位后（除内核软复位和 CPU 复位），CPU 从用户配置字指定的地址开始运行程序，在程序运行中可软件配置启动地址寄存器（SYSCFG_BTADDR），再执行内核软复位或 CPU 复位，程序将从 SYSCFG_BTADDR 指定的地址运行。

第5章 存储器系统控制（MSC）

5.1 概述

存储器系统控制（MSC）主要用作控制与管理 Flash 各种操作，包括页擦除，写操作，读操作以及对应的访问权限管理等，并实时反馈各种控制操作中的状态，以便于系统对 Flash 编程进行控制。

存储器系统控制（MSC）支持 Flash 主程序区的全局读保护。

存储器系统控制（MSC）中可以划分独立数据 Flash 区域用于存放用户数据，用户可根据具体应用灵活选择。

5.2 特性

- ◆ 支持对 Flash 的编程和擦除控制
 - ◇ 非私有保护区全擦除
 - ◇ 页擦除
 - ◇ 字编程
- ◆ 支持对存储器等待时间控制
 - ◇ 支持 Flash 读取等待时间可根据系统时钟配置为 0~3 个系统时钟周期
 - ◇ 支持 Sram 读取等待时间可配置为 0~3 个系统时钟周期

5.3 结构框图

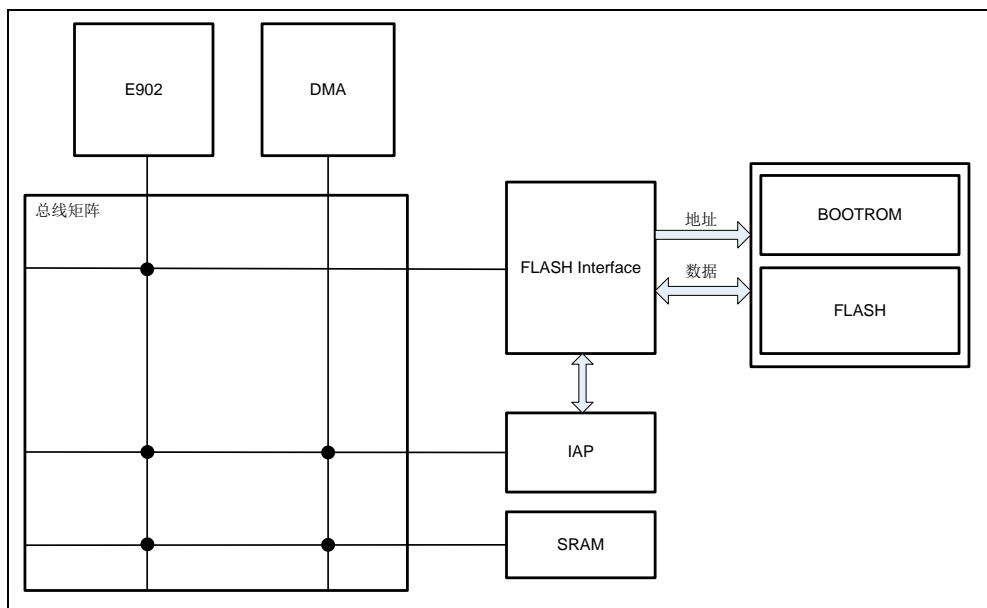


图 5-1 存储器控制结构图

在上电时，系统从 Flash 中加载配置字，配置字加载过程检查成功后，系统从用户配置字指定的启动地址开始运行。DMA 不能直接读取 Flash 中的数据。

5.4 功能描述

5.4.1 Flash保护

5.4.1.1 IAP操作保护KEY

软件通过写 MSC_FLASHKEY，可解除对程序区的保护，处于保护状态时，无法进行擦除和编程的操作。通过检查 MSC_FLASHKEY.STATUS 和 MSC_INFOKEY.STATUS 是否为 0，判断 Flash 是否处于保护状态。

5.4.1.2 Flash写保护区

Flash 存储器可以通过配置字 CFG_WRP0 和 CFG_WRP1 的 START、END 位设置两段写保护区；通过 ENB 位配置两段写保护区使能。

Flash 页擦除和字编程无法对写保护区擦除和写入。

写保护区	使能	起始页号	结束页号
区域 1	CFG_WRP0.ENB	CFG_WRP0.START	CFG_WRP0.END
区域 2	CFG_WRP1.ENB	CFG_WRP1.START	CFG_WRP1.END

表 5-1 写保护区配置字对应表

5.4.1.3 Flash私有读保护区

Flash 存储器可以通过配置字 CFG_PCROP0 和 CFG_PCROP1 的 START、END 位配置两段私有读保护区；通过 ENB 位配置使能。

Flash 对私有读保护区进行任何非法的读取或擦写，均会置位对应的错误标识。

私有读保护区	使能	起始页号	结束页号
区域 1	CFG_PCROP0.ENB	CFG_PCROP0.START	CFG_PCROP0.END
区域 2	CFG_PCROP1.ENB	CFG_PCROP1.START	CFG_PCROP1.END

表 5-2 私有读保护区配置字对应表

5.4.1.4 数据Flash区

Flash 区域可以通过配置字 CFG_DAFLS 划分数据 Flash 区，通过 CFG_DAFLS 的 ENB 位配置数据 Flash 的使能。

Data Flash	使能	起始页号	结束页号
区域 1	CFG_DAFLS.ENB	CFG_DAFLS.START	CFG_DAFLS.END

表 5-3 数据 Flash 配置字对应表

5.4.1.5 Flash全局读保护

Flash 存储器可以进行全局读保护，保护等级分为 Level0，Level2。

当全局保护字为 32 位全 1 时，全局保护级别即为 Level0。

当全局保护字高 16 位为非全 1 且低 16 位也为非全 1 时，全局保护级别即为 Level2。

不同全局加密保护级别下的访问限制如下表：

存储区		全局保护级别	调试 Halt 模式 (调试器读)			用户模式					
			擦	写	读	在 FLASH 中运行			在 SRAM 中运行		
						擦	写	读	擦	写	读
Flash Code 区	非私有读保护区	Level0	NA	NA	是	NA	NA	是	页擦	是	是
		Level2	NA	NA	否	NA	NA	是	页擦	是	否
	私有区保护区	Level0	NA	NA	是	NA	NA	否	否	否	否
		Level2	NA	NA	否	NA	NA	否	否	否	否
	写保护区	Level0	NA	NA	是	NA	NA	是	否	否	是
		Level2	NA	NA	否	NA	NA	是	否	否	否

表 5-4 不同全局保护级别下的访问限制表

- 注 1: 调试 Halt 模式下、在 SRAM 中运行程序时，若全局读保护等级为 Level2 时，不能读取 Flash Code 区数据。
- 注 2: 在 Flash 中运行程序时，禁止对 Flash 本身进行擦写操作，见上表标识 NA。
- 注 3: 用户模式下，在 SRAM 或 IAPROM 中运行程序时，可以擦写 Flash Code 区中的非私有读保护区。
- 注 4: 用户模式下，不支持对 Flash 的非私有读保护区全擦命令。
- 注 5: info 区在所有全局读保护等级下都为只读。
- 注 6: 上表中的 NA 表示无效。

5.4.2 Flash页擦除

页擦除可擦除固定一页空间，其中程序区一页大小可选择为 512Bytes，一次页擦除耗时约 2ms。具体步骤如下：

1. 检查 MSC_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 MSC_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入需擦除页的首地址；
5. 选择页大小；
6. 写入 MSC_FLASHCMD.CMD 命令触发页擦除；
7. 等待 MSC_FLASHSR.BUSY 标志再次变为空闲状态；
8. 判断 MSC_FLASHSR.SERA 标志位是否置起；
9. 设置 Flash 操作请求禁止。

注：数据 Flash 页擦除流程与普通 Flash 页擦除流程一致，仅触发命令不同。

5.4.3 Flash字编程

程序区字编程可一次编程 4 Bytes 空间，一次字编程耗时约 25us。具体步骤如下：

1. 检查 MSC_FLASHSR.BUSY 标志是否处于空闲状态；
2. 通过 MSC_FLASHKEY 解除 Flash 程序区保护状态；
3. 设置 Flash 操作请求使能；
4. 写入需编程地址；
5. 写入需编程数据 MSC_FLASHDR.DATA；
6. 写入 MSC_FLASHCMD.CMD 命令触发字编程；
7. 等待 MSC_FLASHSR.BUSY 标志再次变为空闲状态；
8. 判断 MSC_FLASHSR.PROG 标志位是否置起；
9. 设置 Flash 操作请求禁止。

注：数据 Flash 字编程流程与普通 Flash 字编程流程一致，仅触发命令不同。

5.4.4 Flash编程数据FIFO

FLASH 编程数据 FIFO 可通过 FIFOEN 使能，该 FIFO 为写入 FIFO，读取无效。当数据写入 FIFO 后，可在 MSC_FLASHDR 寄存器中体现。在 FIFO 中写入一次数据时，可触发一次编程。

5.4.5 IAP自编程硬件固化模块

芯片内置 IAP 自编程固化模块，由硬件电路实现，在 IAP 自编程操作程序中可以调用这些

自编程固化模块，以减少 SRAM 中的 IAP 操作代码量。

IAP 自编程硬件固化模块支持页擦，单字编程，双字编程和多字编程，每次调用 IAP 操作函数之前，需要进行解锁操作，分别由如下 IAP 操作函数来实现：

5.4.5.1 CODE区单页擦函数

- ◆ 函数功能：擦除 CODE 区指定的页
- ◆ 入口地址：0x10000004
- ◆ 输入参数：R0-擦除页的首地址，R1-擦除页首地址的反码，R2-固定设为 0
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

5.4.5.2 CODE区单字编程函数

- ◆ 函数功能：向 FLASH CODE 区指定地址写入一个字（32-bits）
- ◆ 入口地址：0x10000008
- ◆ 输入参数：R0-待编程的 FLASH 地址，R1-待编程的 FLASH 地址的反码，R2-待编程数据
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

5.4.5.3 CODE区多字编程

- ◆ 函数功能：向 FLASH CODE 区指定地址写入多个字
- ◆ 入口地址：0x10000000
- ◆ 输入参数：R0-待编程的 FLASH 首地址，R1-待编程的 FLASH 首地址的反码，R2-放在 SRAM 空间的编程数据首地址，R3-编程数据长度（以字节为单位，必须为 4 的整数倍），R4-当编程到页首时是否先进行页擦除（R4≠0 为擦除，R4=0 为不擦除）
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

5.4.5.4 DATA区单页擦函数

- ◆ 函数功能：擦除 DATA 区指定的页
- ◆ 入口地址：0x10000014
- ◆ 输入参数：R0-擦除页的首地址，R1-擦除页首地址的反码，R2-固定设为 0
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

5.4.5.5 DATA区单字编程函数

- ◆ 函数功能：向 FLASH DATA 区指定地址写入一个字（32-bits）
- ◆ 入口地址：0x10000018
- ◆ 输入参数：R0-待编程的 FLASH 地址，R1-待编程的 FLASH 地址的反码，R2-待编程数据
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

5.4.5.6 DATA区多字编程

- ◆ 函数功能：向 FLASH DATA 区指定地址写入多个字
- ◆ 入口地址：0x10000010

- ◆ 输入参数：R0-待编程的 FLASH 首地址，R1-待编程的 FLASH 首地址的反码，R2-放在 SRAM 空间的编程数据首地址，R3-编程数据长度（以字节为单位，必须为 4 的整数倍），R4-当编程到页首时是否先进行页擦除（R4≠0 为擦除，R4=0 为不擦除）
- ◆ 返回值： R0-函数执行状态（R0=1 为成功，R0=0 为失败）

注：在进行页擦时传递的输入参数 R0 必须为 512 的整数倍，否则会导致操作失败。

5.5 特殊功能寄存器

5.5.1 寄存器列表

MSC 寄存器列表		
名称	偏移地址	描述
MSC_FLASHKEY	000 _H	FLASH 程序区操作关键码寄存器
—	004 _H	保留
MSC_FLASHADDR	008 _H	FLASH 擦除编程地址寄存器
MSC_FLASHFIFO	00C _H	FLASH 编程数据写缓存寄存器
MSC_FLASHHDR	010 _H	FLASH 编程数据寄存器
—	014 _H	保留
MSC_FLASHCMD	018 _H	FLASH 操作命令寄存器
MSC_FLASHCR	01C _H	FLASH 控制寄存器
MSC_FLASHSR	020 _H	FLASH 状态寄存器
—	024 _H	保留
MSC_MEMWAIT	028 _H	存储器读取等待时间寄存器
MSC_FLASHADDINV	02C _H	FLASH 擦除编程地址反码寄存器

5.5.2 寄存器描述

5.5.2.1 FLASH程序区关键码寄存器 (MSC_FLASHKEY)

FLASH 程序区关键码寄存器 (MSC_FLASHKEY)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_00000000_00000011 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																													STATUS		

Reserved	Bit 31-2	—	保留
STATUS	Bit 1-0	R	FLASH 程序区状态位 00: 可擦除或编程 其他: 被保护, 不可擦除或编程 IAP复位可将该寄存器复位

注: 对上述该寄存器连续写入 0x8ACE0246 和 0x9BDF1357 可去除保护, 写入其他值或中间插入其他操作将失效。

5.5.2.2 FLASH擦除编程地址寄存器 (MSC_FLASHADDR)

FLASH 擦除编程地址寄存器 (MSC_FLASHADDR)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																	ADDR														

Reserved	Bit 31-18	—	保留, 需固定设置为0
ADDR	Bit 17-0	R/W	FLASH 地址

关于上述寄存器中的 ADDR 位:

注 1: 低 2 位写入无效, 读出始终为 0。

注 2: 页擦除完成后, 地址自动加 0x200。

注 3: 字编程完成后, 地址自动加 4。

5.5.2.3 FLASH编程FIFO寄存器 (MSC_FLASHFIFO)

FLASH 编程 FIFO 寄存器 (MSC_FLASHFIFO)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															

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

FIFO	Bit 31-0	W	FLASH编程FIFO
------	----------	---	--------------------

注 1: 需通过 FIFOEN 使能 FIFO, 写入编程数据, 可适用于 DMA 传输数据, 先写入低位数据, 再写入高位数据。
注 2: 当写入相应个数数据后, 将自动触发字编程。

5.5.2.4 FLASH编程数据寄存器 (MSC_FLASHDR)

FLASH 编程数据寄存器 (MSC_FLASHDR)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

DATA	Bit 31-0	R/W	FLASH编程数据
------	----------	-----	------------------

5.5.2.5 FLASH命令寄存器 (MSC_FLASHCMD)

FLASH 命令寄存器 (MSC_FLASHCMD)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD																															

CMD	Bit 31-0	W	FLASH编程命令 0x00005EA1: 普通Flash页擦除 0x00005DA2: 普通Flash字编程 0x00005BA4: 数据Flash页擦除 0x00005AA5: 数据Flash字编程 0x000050AF: 非私有读保护区全擦除 其他: 保留
-----	----------	---	--

注 1: 若待编程地址上数据非全 F, 无论是否编写相同的数据或代码, 在 FLASH 编程前均必须先对需编程的页进行擦除。

5.5.2.6 FLASH控制寄存器 (MSC_FLASHCR)

FLASH 控制寄存器 (MSC_FLASHCR)																															
----------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								FIFODF	Reserved	FIFOEN	FLASHREQ	Reserved	OTFEN	IAPRST	IAPEN

Reserved	Bit 31-8	—	保留，需固定设置为0
FIFODF	Bit 7	R/W	FIFO 编程数据 Flash 使能 0: 禁止 1: 使能
Reserved	Bit 6	—	保留
FIFOEN	Bit 5	R/W	FIFO 使能 0: 禁止 1: 使能
FLASHREQ	Bit 4	R/W	FLASH 操作请求使能 0: 禁止 1: 使能
Reserved	Bit 3	—	保留
OTFEN	Bit 2	R/W	自编程操作 OTF 模式使能 0: 禁止 1: 使能（仅用作测试，实际应用需设置为 0）
IAPRST	Bit 1	W1	自编程复位 0: 无操作 1: 自编程复位
IAPEN	Bit 0	R/W	自编程使能 0: 禁止 1: 使能

5.5.2.7 FLASH状态寄存器 (MSC_FLASHSR)

FLASH 状态寄存器 (MSC_FLASHSR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						UPCEUL	Reserved	ADDERR	Reserved						UPCEDONE	UPCEBUSY	Reserved						TIMEOUT	PROG	SERA	Reserved	WAE	WPE	BUSY	FLASHACK	

Reserved	Bit 31-26	—	保留
UPCEUL	Bit 25	R	非私有读保护区全擦保护解除位 0: 未解除 1: 已解除 解除保护后, 允许对私有读保护配置区进行擦除
Reserved	Bit 24	—	保留
ADDERR	Bit 23	R	地址反码错误标识位 0: 反码正确 1: 反码错误
Reserved	Bit 22-18	—	保留
UPCEDONE	Bit 17	R	非私有读保护区全擦完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
UPCEBUSY	Bit 16	R	非私有读保护区全擦状态位 0: 空闲 1: 正在进行
Reserved	Bit 15-8	—	保留
TIMEOUT	Bit 7	R	超时错误标志 0: 无错误 1: 发生错误 未在规定时间内完成相应擦除或编程动作时产生错误标志, 可能硬件发生了故障, 需软件触发一次 IAP 复位
PROG	Bit 6	R	字编程完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
SERA	Bit 5	R	页擦除完成标志 0: 未进行或正在进行中 1: 已完成 重新启动新的擦除或编程操作时自动清除
Reserved	Bit 4	—	保留

WAE	Bit 3	R	擦写地址错误标志 0: 无错误 1: 发生错误 可能是在IAP操作在非合法的FLASH地址, 或是在擦除和编程时使用了错误的命令, 需软件触发一次IAP复位
WPE	Bit 2	R	擦写保护错误标志 0: 无错误 1: 发生错误 触发了保护区域的擦除或编程, 操作失败, 需软件触发一次IAP复位
BUSY	Bit 1	R	自编程状态标志 0: 空闲 1: 正在进行
FLASHACK	Bit 0	R	FLASH 操作许可状态 0: 禁止操作 1: 允许操作

5.5.2.8 存储器读取等待时间寄存器 (MSC_MEMWAIT)

存储器读取等待时间寄存器 (MSC_MEMWAIT)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000010 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																SRAM_W		Reserved								FLASH_W					

Reserved	Bit 31-10	—	保留
SRAM_W	Bit 9-8	R/W	SRAM 读取等待时间 00: 无等待 01: 1 个 SYSCLK 10: 2 个 SYSCLK 11: 3 个 SYSCLK 上述配置在任何系统时钟都可用, 通常增加等待时间会降低系统运行效率, 减小运行功耗。
Reserved	Bit 7-2	—	保留
FLASH_W	Bit 1-0	R/W	FLASH 读取等待时间选择 00: 无等待 (系统时钟不超过 24MHz) 01: 1 个 SYSCLK (系统时钟不超过 48MHz) 10: 2 个 SYSCLK (任何系统时钟都可用) 11: 3 个 SYSCLK (任何系统时钟都可用) 通常增加等待时间会降低系统运行效率, 减小运行

			功耗。
--	--	--	-----

5.5.2.9 FLASH擦除编程地址反码寄存器 (MSC_FLASHADDINV)

FLASH 擦除编程地址反码寄存器 (MSC_FLASHADDINV)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ADDRINV																	

Reserved	Bit 31-18	—	保留
ADDRINV	Bit 17-0	R/W	FLASH 地址反码

第6章 系统配置控制器（SYSCFG）

6.1 概述

系统配置模块（SYSCFG）用于芯片的系统级功能配置。

6.2 特性

- ◆ 支持寄存器写保护功能
- ◆ 支持存储器重映射功能
- ◆ 支持中断向量重映射功能

6.3 功能描述

6.3.1 系统寄存器写保护

为避免程序的异常运行对系统级模块的误操作，系统写保护寄存器 SYSCFG_PROT 用于阻止程序对系统级模块其它寄存器的误写入。该寄存器保护范围为除 SYSCFG_PROT 寄存器外的 SYSCFG、PMU、CMU、RMU 模块所有寄存器。

SYSCFG_PROT 寄存器为虚拟寄存器。要对系统级模块其它寄存器进行写操作时，需先对 SYSCFG_PROT 寄存器写 0x55AA6996，之后可对系统级模块其它寄存器进行写操作。对 SYSCFG_PROT 寄存器写入其他值重新进入写保护状态，写保护状态下对系统寄存器进行的写操作将被忽略。

可以通过读取 SYSCFG_PROT 寄存器确认系统级模块是否处于写保护状态，读出值为 0x00000000，表示当前可对系统级模块寄存器进行写操作；读出值为 0x00000001 表示系统级模块处于写保护状态。SYSCFG_PROT 寄存器无其它读出值。

6.3.2 启动地址配置

通过设置启动地址寄存器（SYSCFG_BTADDR）再执行内核软复位或 CPU 复位，程序将从 SYSCFG_BTADDR 指定的地址运行。

6.4 特殊功能寄存器

6.4.1 寄存器列表

SYSCFG 寄存器列表		
名称	偏移地址	描述
SYSCFG_PROT	000 _H	系统写保护寄存器
Reserved	004 _H	保留
SYSCFG_BTADDR	008 _H	启动地址寄存器
SYSCFG_TBKCFG	00C _H	系统刹车源使能寄存器
SYSCFG_PISIOCFG	010 _H	PIS 生产端 GPIO 配置寄存器

6.4.2 寄存器描述

6.4.2.1 系统写保护寄存器 (SYSCFG_PROT)

系统写保护寄存器 (SYSCFG_PROT)																															
偏移地址: 000 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																															PROT

KEY	Bit 31-1	W	保护关键码 0x55AA6996: 去除写保护 其他: 开启写保护
PROT	Bit 0	R/W	保护状态位 0: 无写保护 1: 写保护

6.4.2.2 启动地址寄存器 (SYSCFG_BTADDR)

启动地址寄存器 (SYSCFG_BTADDR)																															
偏移地址: 008 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															BTADDR																

Reserved	Bit 31-18	—	保留
BTADDR	Bit 17-0	R/W	CPU 启动地址值 注: 最低 12 位固定为 0

6.4.2.3 TIM刹车源配置寄存器 (SYSCFG_TBKCFG)

TIM 刹车源配置寄存器 (SYSCFG_TBKCFG)																															
偏移地址: 00C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											CLUBKE	LYDBKE	CCSBKE		

Reserved	Bit 31-3	—	保留
CLUBKE	Bit 2	R/W	CPU 锁死作为 TIM 刹车源使能位 0: 禁止 1: 使能
LVDBKE	Bit 1	R/W	LVD 事件作为 TIM 刹车源使能位 0: 禁止 1: 使能
CCSBKE	Bit 0	R/W	时钟安全事件作为 TIM 刹车源使能位 0: 禁止 1: 使能

6.4.2.4 PIS生产端GPIO配置寄存器 (SYSCFG_PISIOCFG)

PIS 生产端 GPIO 配置寄存器 (SYSCFG_PISIOCFG)																															
偏移地址: 010 _H																															
复位值: 01010101_01010101_01010101_01010101 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IO15SEL	IO14SEL	IO13SEL	IO12SEL	IO11SEL	IO10SEL	IO9SEL	IO8SEL	IO7SEL	IO6SEL	IO5SEL	IO4SEL	IO3SEL	IO2SEL	IO1SEL	IO0SEL																

IO15SEL	Bit 31-30	R/W	PIS 生产端 GPIO15 选择位 00: PA 01: PB 10: PC 11: PD
IO14SEL	Bit 29-28	R/W	PIS 生产端 GPIO14 选择位 00: PA 01: PB 10: PC 11: PD
IO13SEL	Bit 27-26	R/W	PIS 生产端 GPIO13 选择位 00: PA 01: PB 10: PC 11: PD
IO12SEL	Bit 25-24	R/W	PIS 生产端 GPIO12 选择位 00: PA 01: PB 10: PC 11: PD
IO11SEL	Bit 23-22	R/W	PIS 生产端 GPIO11 选择位 00: PA

			01: PB 10: PC 11: PD
IO10SEL	Bit 21-20	R/W	PIS 生产端 GPIO10 选择位 00: PA 01: PB 10: PC 11: PD
IO9SEL	Bit 19-18	R/W	PIS 生产端 GPIO9 选择位 00: PA 01: PB 10: PC 11: PD
IO8SEL	Bit 17-16	R/W	PIS 生产端 GPIO8 选择位 00: PA 01: PB 10: PC 11: PD
IO7SEL	Bit 15-14	R/W	PIS 生产端 GPIO7 选择位 00: PA 01: PB 10: PC 11: PD
IO6SEL	Bit13-12	R/W	PIS 生产端 GPIO6 选择位 00: PA 01: PB 10: PC 11: PD
IO5SEL	Bit11-10	R/W	PIS 生产端 GPIO5 选择位 00: PA 01: PB 10: PC 11: PD
IO4SEL	Bit9-8	R/W	PIS 生产端 GPIO4 选择位 00: PA 01: PB 10: PC 11: PD
IO3SEL	Bit7-6	R/W	PIS 生产端 GPIO3 选择位 00: PA 01: PB 10: PC 11: PD
IO2SEL	Bit 5-4	R/W	PIS 生产端 GPIO2 选择位

			00: PA 01: PB 10: PC 11: PD
IO1SEL	Bit 3-2	R/W	PIS 生产端 GPIO1 选择位 00: PA 01: PB 10: PC 11: PD
IO0SEL	Bit 1-0	R/W	PIS 生产端 GPIO0 选择位 00: PA 01: PB 10: PC 11: PD

第7章 电源管理（PMU）及低功耗模式

7.1 概述

电源管理单元（PMU）管理芯片的电源以及低功耗模式。在每个低功耗模式下，都有其对应的单元模块状态（使能、中止或掉电）。芯片可支持各种功耗模式：RUN，SLEEP，STOP。其中 RUN 为芯片正常运行模式，所有的外设模块均可被使能，SLEEP 和 STOP 为低功耗模式。在 STOP 模式时，CPU 暂停，大部分外设被中止，SRAM 和寄存器中数据保持，唤醒之后外设继续运行，CPU 从暂停处继续运行。

低功耗模式通过软件操作使能，SLEEP，STOP 模式可通过一系列中断或事件唤醒回到 RUN 模式。

7.2 特性

- ◆ 支持多种低功耗模式配置
- ◆ 支持多种唤醒源灵活配置
- ◆ 快速的唤醒时间

7.3 结构框图

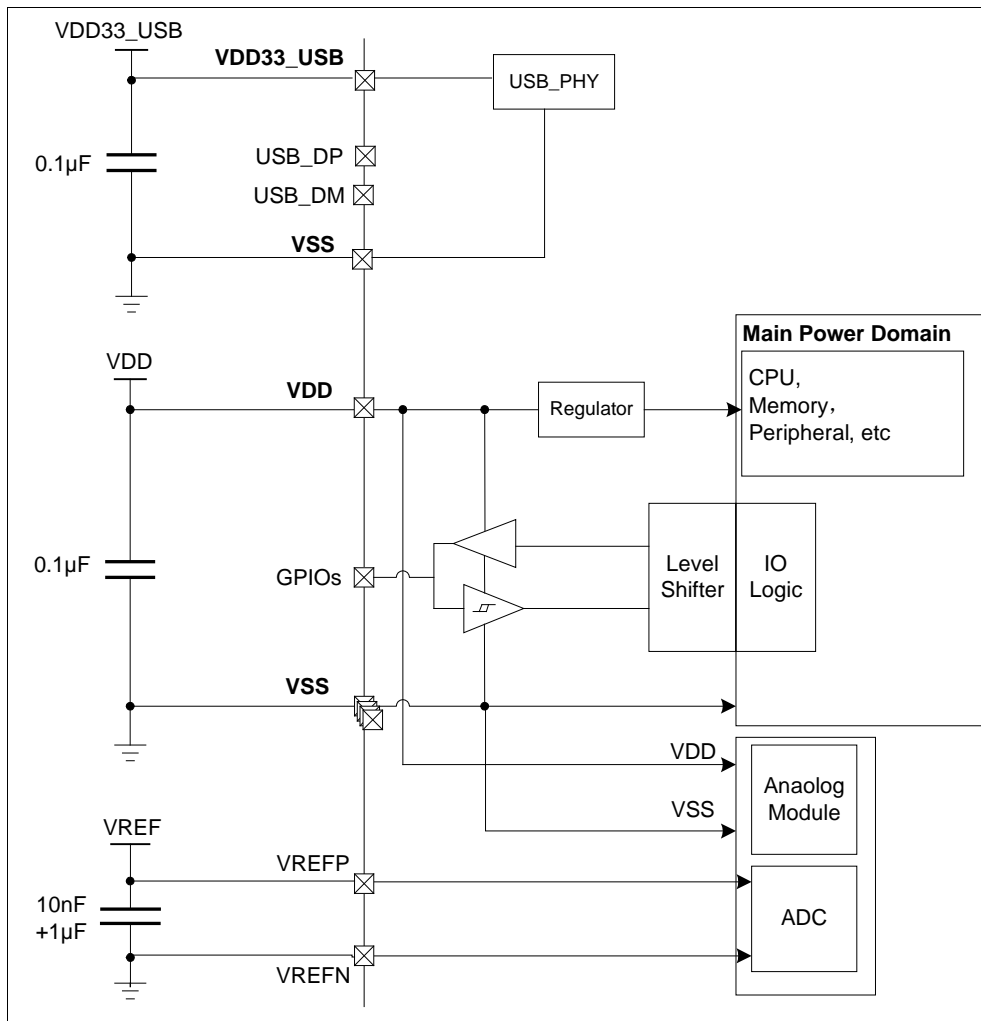


图 7-1 电源结构框图

7.4 功能描述

7.4.1 芯片电源

7.4.1.1 主系统电源域

芯片工作电压 VDD 要求介于 2.2V 到 5.5V 之间。嵌入式 LDO 用于提供内部数字电源。

USB 工作电压 VDD33_USB 给 USB 模块物理层电路和 USB 通信接口 (USB_DM 和 USB_DP) 供电, 当使用芯片 USB 模块进行通信时, VDD33_USB 需根据系统需求, 提供合适的电压, 通常介于 3.0V 到 3.6V 之间; 当不使用 USB 模块时, VDD33_USB 可以与 VDD 短接在一起供电。

7.4.1.2 独立的模拟模块电源和参考电压

ADC 电源由 VDD 提供。

ADC 参考电压可选 VDD 和 VREFP 端口。

为了确保测量低电压时具有更高的精度, 用户可以在 VREFP 上连接单独的 ADC 外部参考电压输入。VREFP 电压应介于 2.2 V 到 VDD 之间。

7.4.2 电源监视

7.4.2.1 上电复位 (POR)

芯片内部集成 POR 产生电路。当 VDD 低于指定阈值 VPOR/VPDR 时, 器件无需外部复位电路便会保持复位状态。

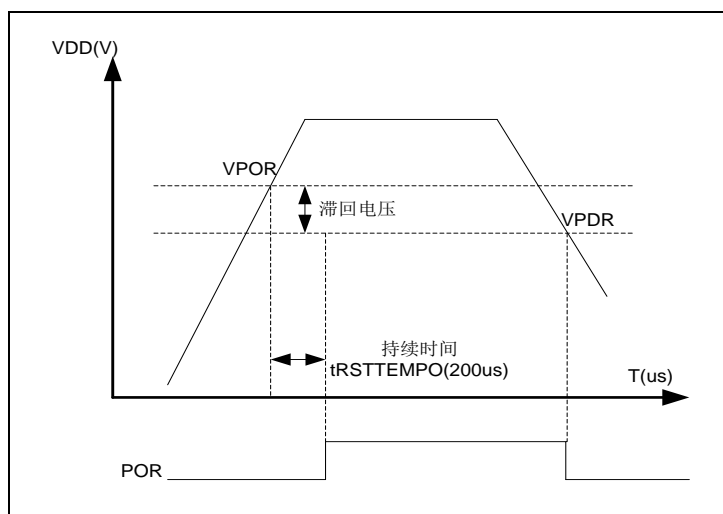


图 7-2 POR/PDR 示意图

7.4.2.2 欠压复位 (BOR)

上电期间, 欠压复位 (BOR) 将使器件保持复位状态, 直到电源电压达到由配置字设定的 VBOR 阈值。芯片支持 7 个 VBOR 阈值可配。

当电源电压 (VDD) 降至所选 VBOR 阈值以下时, 将使器件复位。

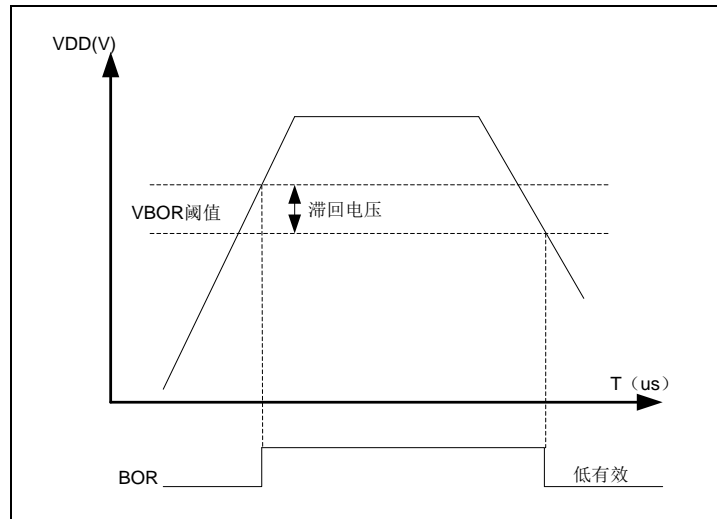


图 7-3 BOR 示意图

7.4.2.3 低电压检测 (LVD)

LVD 可用于监视 VDD 电源，通过设置 LVDEN 使能 LVD，将 VDD 电压和 LVDS 所选择的电压值进行比较，可粗略判断当前电源 VDD 的电压值。

LVD 提供了一个状态标志位 LVDO，用于指示 VDD 是大于还是小于 LVD 阈值。通过使能 LVDIE 可使能 LVD 中断，通过选择 LVDIS 可选择 LVD 中断类型。当 VDD 降至 LVD 阈值以下，以及/或者当 VDD 升至 LVD 阈值以上时，可以产生 LVD 中断，具体取决于 LVDIS 的中断类型配置。该功能的用处之一就是可以在 VDD 发生跌落时，立即进入中断服务程序中执行紧急关闭系统的任务。

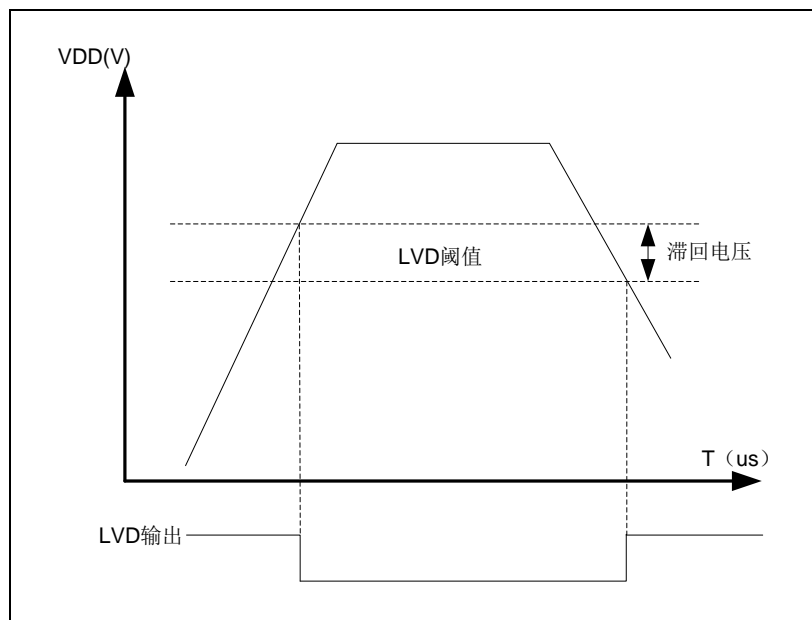


图 7-4 LVD 示意图

7.4.3 低功耗模式

7.4.3.1 低功耗模式转换

默认情况下，系统复位或上电复位后，微控制器进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。系统提供了多个低功耗模式，可在 CPU 不需要运行时（例如等待外部事件时）节省功耗。由用户根据应用选择具体的低功耗模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

芯片支持以下低功耗模式：

- ◇ SLEEP 模式（E902 内核停止，外设保持运行）
- ◇ STOP 模式（DMA 关闭，仅部分低功耗外设（IWDG、WWDT）可工作，HCLK、PCLK 停止，仅 LRC 时钟工作，主电源域稳压器工作在低功耗模式下）

此外，可通过下列方法之一降低运行模式的功耗：

- ◇ 降低系统时钟速度
- ◇ 不使用 APB 和 AHB 外设时，将对应的外设时钟关闭

进入低功耗模式的转换关系如下图所示：

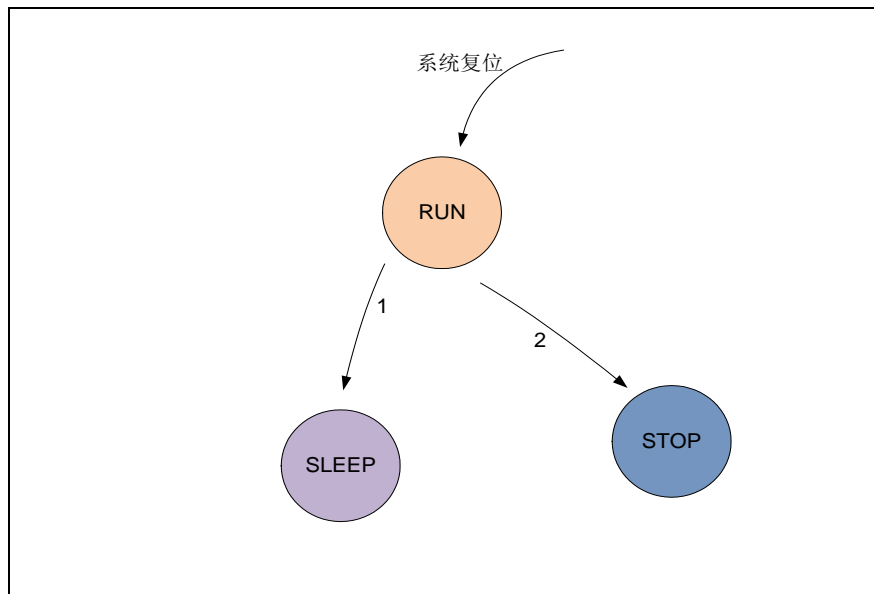


图 7-5 低功耗模式转换图

序号	模式	进入	唤醒	逻辑时钟影响	时钟源影响	稳压器
1	SLEEP	WFI	任意中断	CPU 时钟关闭	无	普通模式
		WFE	唤醒事件			
2	STOP	DEEPSLEEP 位 +WFI/WFE	具体请参照章节“芯片配置指引” STOP 低功耗模式的中断唤醒源	CPU 时钟关闭	HOSC、HRC、PLL 关闭 (LRC 工作)	低功耗模式

表 7-1 低功耗模式说明

7.4.3.2 系统时钟速度

在运行模式下, 可通过对预分频寄存器编程来降低系统时钟 (SYSCLK、HCLK、PCLK) 速度。进入睡眠模式之前, 也可以使用这些预分频器降低外设速度。也可将系统时钟切换至低速时钟源并关闭高速时钟源来降低功耗。

系统时钟速度的有关详细信息, 请参见时钟管理。

7.4.3.3 外设时钟门控

在运行模式下, 可通过设置时钟门控来停止各外设和存储器的总线时钟或模块工作时钟以降低功耗。

要进一步降低低功耗模式的功耗, 可在执行 WFI 或 WFE 指令之前可通过门控关闭外设时钟。

外设时钟门控配置的有关详细信息, 请参见时钟管理。

7.4.3.4 RUN模式

- ◇ 所有高速时钟源可使能
- ◇ 所有外设可使能

7.4.3.5 SLEEP模式

- ◇ 所有高速时钟源可使能
- ◇ CPU 时钟被关断
- ◇ 所有外设可使能

7.4.3.6 STOP模式

- ◇ 高速时钟源默认禁止
- ◇ CPU 时钟关闭
- ◇ 低功耗外设 LVD, IWDG/MWDG 等可工作
- ◇ SRAM 和各寄存器数据保持

7.4.3.7 低功耗模式下各模块操作

下表列举了各模块在低功耗模式下的操作可能性，为低功耗应用提供参考。

功能模块	SLEEP	STOP
CLIC	工作	停止
调试	工作	停止
Flash	工作	可配置是否 STOP
SRAM	数据保持	数据保持
主域稳压器	普通模式	低功耗模式
欠压检测	可配置	可配置
低电压检测	可配置	可配置
DMA 控制器	可工作	停止
外设互联	工作	停止
独立看门狗定时器	可工作	可工作
窗口看门狗定时器	可工作	可工作
HOSC	可配置	可配置
LRC	工作	工作
HRC4M	工作	可配置
HRC48M	工作	可配置
PLL	工作	可配置
内核时钟	停止	停止
系统时钟	工作	停止
GPIO	工作	工作（中断）
CRC	可工作	停止
高级定时器	可工作	停止
通用定时器	可工作	停止
基本定时器	可工作	停止
I2C 接口	可工作	停止
串行外设接口（SPI）	可工作	停止
通用异步收发器（UART）	可工作	停止
USB	可工作	停止
ADC	可工作	停止

表 7-2 低功耗模式下各模块操作

7.5 特殊功能寄存器

7.5.1 寄存器列表

PMU 寄存器列表		
名称	偏移地址	描述
PMU_CR	000 _H	PMU 控制寄存器
PMU_SR	004 _H	PMU 状态寄存器
PMU_LVDCR	008 _H	LVD 控制寄存器
Reserved	00C _H	保留
PMU_TWUR	010 _H	唤醒延时寄存器

7.5.2 寄存器描述

7.5.2.1 PMU控制寄存器 (PMU_CR)

PMU 控制寄存器 (PMU_CR)																																			
偏移地址: 00 _H																																			
复位值: 00011100_00000001_00000000_10000000 _B																																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Reserved			WKEVNSEL				FSTOP		Reserved		BGSTOP		LPSTOP		Reserved		LPVS				Reserved								VROSCEN	NORRTNEN	STPRTNEN	Reserved		CWUF	LPM

Reserved	Bit 31-29	—	保留
WKEVNSEL	Bit 28-24	R/W	WFE 休眠模式事件唤醒选择位 选择对应的中断向量 (表 3-1 中 16~47) 默认值: 1C _H 该寄存器位仅用于测试, 在系统应用时需软件固定设置为 10 _H
FSTOP	Bit 23	R/W	STOP 模式 Flash STOP 使能位 0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
Reserved	Bit 22	—	保留
BGSTOP	Bit 21	R/W	STOP 模式 BG 低功耗使能位 0: 禁止 1: 使能 (设置为使能可以降低 STOP 模式功耗, 使能时会影响芯片 STOP 模式下的 BOR 和 LVD 档位电压, 见表格下方备注)
LPSTOP	Bit 20	R/W	STOP 模式 LDO 低功耗使能位 0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
Reserved	Bit 19-18	—	保留
LPVS	Bit 17-16	R/W	LDO 低功耗模式输出电压选择位 01: 1.5V (默认, 需设置为该电压档位) 其他: 保留 (仅用于内部测试用)
Reserved	Bit 15-8	—	保留
VROSCEN	Bit7	R/W	VR 工作时钟使能位 0: 禁止 (推荐设置为禁止, 以降低功耗) 1: 使能 仅芯片启动时有效, 可在芯片启动后将该位禁止
NORRTNEN	Bit6	R/W	普通模式 SRAM Retention 使能位 0: 禁止 (需固定设置为 0) 1: 使能 (仅用于测试)
STPRTNEN	Bit5	R/W	STOP 模式 SRAM Retention 使能位

			0: 禁止 1: 使能 (推荐设置为使能, 以降低功耗)
Reserved	Bit 4-3	—	保留
CWUF	Bit 2	W1	WUF 标志清除位 0: 无操作 1: 清除WUF标志
LPM	Bit 1-0	R/W	低功耗模式选择位 (需固定设置为 0)

注 1: 寄存器 PMU_CR 的 WKEVNSEL 位仅用于测试, 在系统应用时需软件固定设置为 10_H, 否则可能会误唤醒休眠模式。

注 2: 芯片在 STOP 模式下, 如果 BG 电压模块为低功耗模式 (寄存器 PMU_CR 的 BGSTOP=1), 则 BOR 和 LVD 档位电压相对于芯片工作模式 (RUN 模式或 SLEEP 模式) 下的档位电压均会有约 ±10% 范围的偏差; 如果应用中需要 BOR 和 LVD 档位电压在芯片 STOP 模式与工作模式下保持一致, 则需要禁止 BG 低功耗模式 (设置寄存器 PMU_CR 的 BGSTOP=0), 此时芯片 STOP 模式功耗会增大约 0.5uA。

7.5.2.2 PMU状态寄存器 (PMU_SR)

PMU 状态寄存器 (PMU_SR)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															WUF

Reserved	Bit 31-1	—	保留
WUF	Bit 0	R	唤醒标志位 0: 未发生唤醒事件 1: 有发生唤醒事件 注: 该位通过 PMU_CR 寄存器的 CWUF 位来清零

7.5.2.3 LVD控制寄存器 (PMU_LVDCR)

LVD 控制寄存器 (PMU_LVDCR)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																LVDO	Reserved			LVDFLT	LVDIFS			LVDS				LVDCIF	LVDIF	LVDIE	LV DEN

Reserved	Bit 31-16	—	保留
LVDO	Bit 15	R	LVD 状态标志位 0: 大于阈值 1: 小于阈值
Reserved	Bit 14-12	—	保留
LVDFLT	Bit 11	R/W	LVD 滤波使能位 0: 禁止 1: 使能 注: 使能后 LVDO 稳定时间小于 100us 的变化将被忽略
LVDIFS	Bit 10-8	R/W	LVD 中断标志产生模式选择位 000: LVDO 上升沿产生中断 001: LVDO 下降沿产生中断 010: LVDO 高电平产生中断 011: LVDO 低电平产生中断 1xx: LVDO 变化 (上升或下降沿) 产生中断
LVDS	Bit 7-4	R/W	LVD 触发电压 0000: 2.2V 0001: 2.4V 0010: 2.6V 0011: 2.8V 0100: 3.0V 0101: 3.6V 0110: 4.0V 0111: 4.6V 1xxx: 保留
LVDCIF	Bit 3	W	LVD 中断标志清除位 0: 无操作 1: 清除LVD中断标志
LVDIF	Bit 2	R	LVD 中断标志位 0: LVDO 状态未变化 1: LVDO状态发生变化 注: 该位由LVDCIF位写1清除

LVDIE	Bit 1	R/W	LVD 中断使能位 0: 禁止 1: 使能
LVDEN	Bit 0	R/W	LVD 使能位 0: 禁止 1: 使能

7.5.2.4 唤醒延时寄存器 (PMU_TWUR)

唤醒延时寄存器 (PMU_TWUR)																															
偏移地址: 010 _H																															
复位值: 00000000_00000000_00001111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																TWU															

Reserved	Bit 31-12	—	保留
TWU	Bit 11-0	R/W	睡眠模式唤醒时间选择位 $T_{WAKEUP} = T_{SYSCLK} \times (TWU + 1)$ 注: SLEEP模式的唤醒时间为TWAKEUP; STOP模式的唤醒时间约为TWAKEUP+12TLRC。

注 1: SLEEP 模式下, 唤醒时间选择位 TWU 最小可设置为 0, 唤醒后经过约 7 个系统时钟周期, 芯片 CPU 内核开始运行程序;

注 2: STOP 模式下, 当系统时钟为外部振荡器 HOSC 或 PLL 时钟时, 建议将唤醒时间选择位 TWU 设置为最大值 0xFFF, 当系统时钟为内部 HRC4M 或 HRC48M 时钟时, 建议将唤醒时间选择位 TWU 设置为 0x3FF 以上, 当系统时钟为内部 LRC 时钟时, 建议将唤醒时间选择位 TWU 设置为 0xF 以上。

第8章 复位管理 (RMU)

8.1 概述

系统复位可以由下面列出的任一事件触发。这些复位事件标志可以通过读取 RMU_RSTSR 寄存器来判断复位源。

8.2 特性

- ◆ 支持 POR/BOR
- ◆ 支持外部端口复位 MRST
- ◆ 支持看门狗溢出复位
- ◆ 内核锁死 (LOCKUP) 复位
- ◆ 读取配置字错误复位 (CFG_RST)
- ◆ 支持三种软件复位
 - ◇ 复位整个数字域
 - ◇ 复位除启动配置外的整个数字域
 - ◇ 复位内核
- ◆ 支持各外设模块的单独复位

8.3 结构框图

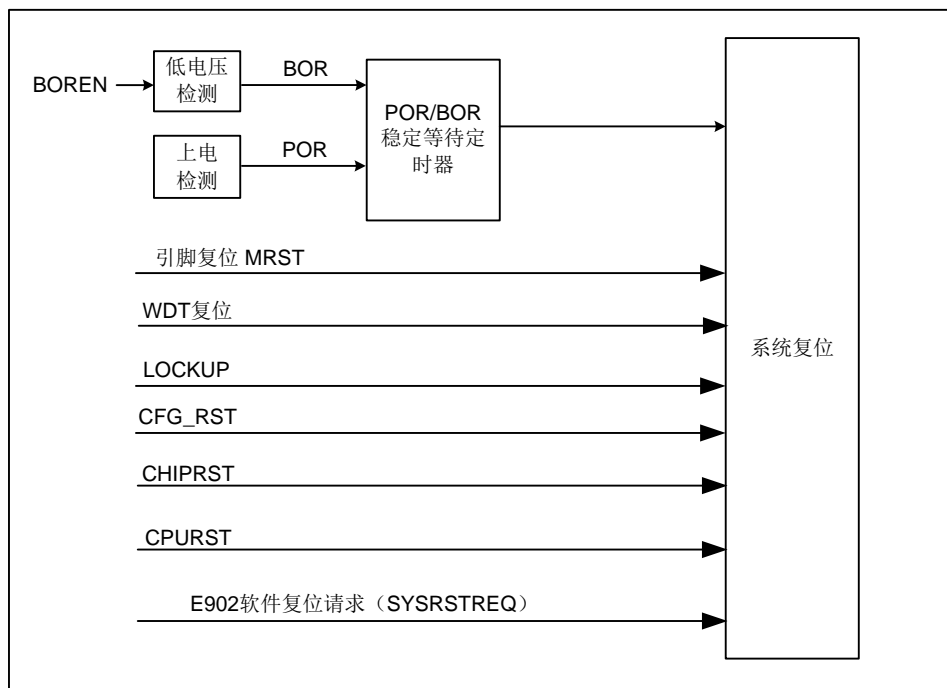


图 8-1 复位结构图

8.4 功能描述

ES32VF2264 微控制器支持 8 种复位源。CPURST 只复位 CPU 内核（不包含调试部分）；其他复位源则复位 CPU 内核和所有外设。各个复位源及寄存器关系如下表所示：

	POR	BOR	MRSTN	WDT	LOCKUP	CHIPRST	SYSRST REQ	CPU RST
RMU 复位状态寄存器 (RMU_RSTSR)	POR=1 WAKEUP=1	BOR=1	NMRST=1	WWDT=1 或 IWDT=1	LOCKUP=1	CHIP=1	MCU=1	CPU=1
LV DEN (LVDCR[0])	0x0	0x0	0x0	0x0	0x0	0x0	0x0	-
SYS_STU (CMU_CSR)	0x1	0x1	0x1	0x1	-	0x1	-	-
CFT_STU (CMU_CSR)	0x0	0x0	0x0	0x0	-	0x0	-	-
系统和外设时钟分频	1 分频	1 分频	1 分频	1 分频	-	1 分频	-	-
HOSC_EN	0x0	0x0	0x0	0x0	0x0	0x0	0x0	-
PLL_EN	0x0	0x0	0x0	0x0	0x0	0x0	0x0	-
HRC4M_EN	0x1	0x1	0x1	0x1	0x1	0x1	0x1	-
HRC48M_EN	0x1	0x1	0x1	0x1	0x1	0x1	0x1	-
LRC_EN	0x1	0x1	0x1	0x1	0x1	0x1	0x1	-
CMU_AHB1ENR	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	-
CMU_APB1ENR	0x0	0x0	0x0	0x0	0x0	0x0	0x0	-
CPU 内核调试模块	复位值	复位值	-	-	-	复位值	-	-
其他外设	复位值	复位值	复位值	复位值	复位值	复位值	复位值	-

表 8-1 系统复位与寄存器关系

注：LOCKUP 复位和 SYSRSTREQ 软件复位，不会复位系统时钟控制电路，当发生 LOCKUP 复位或 SYSRSTREQ 软件复位时，如果外部时钟 HOSC 或倍频时钟 PLL 用作系统时钟，则对应的时钟仍保持工作，不会被关闭。

8.4.1 硬件复位

硬件复位包括上电复位，欠压复位，外部端口复位，LOCKUP 复位，WDT 复位和读取配置字错误复位。

8.4.1.1 上电复位

芯片内部集成 POR 产生电路。当 VDD 低于指定阈值 VPOR/VPDR 时，器件无需外部复位电路便会保持复位状态。

8.4.1.2 欠压复位

上电期间，欠压复位（BOR）将使器件保持复位状态，直到电源电压达到由配置字设定

的 VBOR 阈值。芯片支持 7 个 VBOR 阈值，可通过配置字 CFG_WORD 的 BORVS 位进行选择。

当电源电压（VDD）降至所选 VBOR 阈值以下时，将使器件复位。

8.4.1.3 端口复位

MRST 端口为低电平时，可复位除内核调试模块以外的芯片整体，复位解除后，芯片从 FLASH 启动。

8.4.1.4 看门狗复位

详细描述请参照独立看门狗和窗口看门狗的说明。

可复位除内核调试模块以外的芯片整体，复位解除后，芯片正常从 FLASH 启动。

8.4.1.5 LOCKUP复位

由不可恢复的异常导致的内核锁死，此时将产生复位信号来重新启动内核及系统。详细说明可参考 E902 技术手册。

8.4.1.6 读取配置字错误复位

在配置字加载或者程序运行过程中，由于异常干扰导致配置字的读取和控制出现错误，可能导致严重系统错误，此时将产生复位并重新加载配置字。

8.4.2 软件复位

8.4.2.1 芯片复位（CHIPRST）

芯片复位由寄存器 RMU_AHB2RSTR.CHIPRST 位控制，可复位整体芯片。复位后芯片从 FLASH 空间启动。

8.4.2.2 CPU复位（CPURST）

CPU 复位由寄存器 RMU_AHB2RSTR.CPURST 位控制，可复位内核（不包含调试部分）。

8.4.2.3 内核复位请求（SYSRSTREQ）

MCU 复位从内核产生。由扩展异常状态寄存器（MEXSTATUS）的 SOFT_RST 位控制，将该位置 2'b01 可对系统复位。详细可参考 E902 相关技术手册。

8.4.2.4 外设软件复位

对应每个外设分别分配了一个软件复位。

AHB1 外设复位寄存器（RMU_AHB1RSTR）为 GPIO,CRC ,PIS 等模块提供软件复位。

AHB2 外设复位寄存器（RMU_AHB2RSTR）作为 CPU 复位和芯片复位的控制寄存器。

APB 外设复位寄存器（RMU_APBSTR）为看门狗定时器，模数转换器（ADC），定时器，通信外设等 APB 模块提供软件复位。

8.5 特殊功能寄存器

8.5.1 寄存器列表

RMU 寄存器列表		
名称	偏移地址	描述
RMU_CR	000 _H	RMU 控制寄存器
RMU_RSTSR	010 _H	RMU 复位状态寄存器
RMU_CRSTSR	014 _H	RMU 清复位状态寄存器
RMU_AHB1RSTR	020 _H	AHB1 外设复位寄存器
RMU_AHB2RSTR	024 _H	AHB2 外设复位寄存器
RMU_APBSTR	030 _H	APB 外设复位寄存器

8.5.2 寄存器描述

8.5.2.1 RMU控制寄存器 (RMU_CR)

RMU 控制寄存器 (RMU_CR)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_00000000_00000011 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												BORFLT		Reserved	

Reserved	Bit 31-7	—	保留, 需固定设置为0
BORFLT	Bits 3-1	R/W	BOR 滤波时钟选择 00x: 1 个 LRC 周期 010: 2 个 LRC 周期 011: 3 个 LRC 周期 100: 4 个 LRC 周期 101: 5 个 LRC 周期 110: 6 个 LRC 周期 111: 7 个 LRC 周期
Reserved	Bit 0	R/W	保留, 需固定设置为1, 否则可能会导致芯片复位异常

8.5.2.2 RMU复位状态寄存器 (RMU_RSTSR)

RMU 复位状态寄存器 (RMU_RSTSR)																																					
偏移地址: 10 _H																																					
复位值: 00000000_00000000_00000100_00000011 _B																																					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reserved																CFGERR	Reserved										CFG	CPU	MCU	CHIP	LOCKUP	WWDT	IWDT	NMRST	BOR	WAKEUP	POR

Reserved	Bit 31-17	—	保留
CFGERR	Bit 16	R	配置字错误状态标志位 0: 无配置字错误 1: 产生配置字错误 注: 当程序运行过程中出现配置字错误时, 软件需要触发看门狗复位或芯片全局复位来复位芯片
Reserved	Bit 15-11	—	保留
CFG	Bit 10	R	配置字复位状态标志位 0: 无复位发生或标志位已被清除

			1: 有复位发生 注: 该位仅用于内部测试, 用户可无需关注
CPU	Bit 9	R	软件 CPU 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
MCU	Bit 8	R	软件 MCU 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生 注: 该复位从内核产生, 详细可参考E902相关技术手册。
CHIP	Bit 7	R	软件 CHIP 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
LOCKUP	Bit 6	R	Lockup 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
WWDT	Bit 5	R	WWDT 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
IWDT	Bit 4	R	IWDT 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
NMRST	Bit 3	R	NMRST 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
BOR	Bit 2	R	BOR 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有复位发生
WAKEUP	Bit 1	R	唤醒复位状态标志位 0: 无复位发生或标志位已被清除 1: 有上电复位发生
POR	Bit 0	R	POR 复位状态标志位 0: 无复位发生或标志位已被清除 1: 有上电复位发生

注 1: 标志位 WAKEUP 和 POR 任意一个为 1 时, 均表示芯片发生了上电复位。

注 2: 芯片上电后必须先对标志位 WAKEUP 和 POR 进行清零操作, 否则即使发生 RMU_RSTSR 寄存器 bit2~bit5 和 bit7 对应的复位事件, 可能也无法置起 bit2~bit5 和 bit7 对应的标志位。

注 3: 上述复位标志位可通过向寄存器 RMU_CRSTSR 的对应位写 1 来清除。

8.5.2.3 RMU清复位状态寄存器 (RMU_CRSTSR)

RMU 清复位状态寄存器 (RMU_CRSTSR)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																					CFG	CPU	MCU	CHIP	LOCKUP	WWDT	IWDT	NMRST	BOR	Reserved	POR

Reserved	Bit 31-11	—	保留
CFG	Bit 10	W	配置字复位标志清除 0: 无操作 1: 清除标志
CPU	Bit 9	W	软件 CPU 复位标志清除 0: 无操作 1: 清除标志
MCU	Bit 8	W	软件 MCU 复位标志清除 0: 无操作 1: 清除标志
CHIP	Bit 7	W	软件 CHIP 复位标志清除 0: 无操作 1: 清除标志
LOCKUP	Bit 6	W	LOCKUP 复位标志清除 0: 无操作 1: 清除标志
WWDT	Bit 5	W	WWDT 复位标志清除 0: 无操作 1: 清除标志
IWDT	Bit 4	W	IWDT 复位标志清除 0: 无操作 1: 清除标志
NMRST	Bit 3	W	NMRST 复位标志清除 0: 无操作 1: 清除标志
BOR	Bit 2	W	BOR 复位标志清除 0: 无操作 1: 清除标志
WAKEUP	Bit 1	W	WAKEUP 复位标志清除 0: 无操作 1: 清除标志
POR	Bit 0	W	POR 复位标志清除 0: 无操作

			1: 清除标志
--	--	--	---------

注：芯片上电后必须先对标志位 WAKEUP 和 POR 进行清零操作，否则即使发生 RMU_RSTSR 寄存器 bit2~bit5 和 bit7 对应的复位事件，可能也无法置起 bit2~bit5 和 bit7 对应的标志位。

8.5.2.4 AHB1 外设复位寄存器 (RMU_AHB1RSTR)

AHB1 外设复位寄存器 (RMU_AHB1RSTR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CSURST	USBRST	PISRST	Reserved	DMARST	CRCRST	GPIORST	

Reserved	Bit 31-8	—	保留
CSURST	Bit 7	R/W	CSU 复位 0: 无操作 1: 复位
USBRST	Bit 6	R/W	USB 复位 0: 无操作 1: 复位
PISRST	Bit 5	R/W	PIS 复位 0: 无操作 1: 复位
Reserved	Bit4-3	—	保留
DMARST	Bit 2	R/W	DMA 复位 0: 无操作 1: 复位
CRCRST	Bit 1	R/W	CRC 复位 0: 无操作 1: 复位
GPIORST	Bit 0	R/W	GPIO 复位 0: 无操作 1: 复位

8.5.2.5 AHB2 外设复位寄存器 (RMU_AHB2RSTR)

AHB2 外设复位寄存器 (RMU_AHB2RSTR)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	CPURST	CHIPRST
----------	--------	---------

Reserved	Bit 31-2	—	保留
CPURST	Bit 1	R/W	处理器内核复位 0: 无操作 1: 复位 注: 该复位只复位处理器内核 (不包括 DEBUG 逻辑)
CHIPRST	Bit 0	R/W	芯片全局复位 0: 无操作 1: 复位

8.5.2.6 APB外设复位寄存器 (RMU_APBSTR)

APB 外设复位寄存器 (RMU_APBSTR)																																		
偏移地址: 30 _H																																		
复位值: 00000000_00000000_00000000_00000000 _b																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reserved						ADCRST	DBGCONRST	IWDTRST	WWDTRST	I2C1RST	I2C0RST	Reserved			SPI1RST	SPI0RST	Reserved		CUART2RST	CUART1RST	CUART0RST	Reserved			EUART1RST	EUART0RST	Reserved			GP16C4T2RST	GP16C4T1RST	GP16C4T0RST	BS16T0RST	AD16C4T0RST

Reserved	Bit 31-26	—	保留
ADCRST	Bit25	R/W	ADC 复位 0: 无操作 1: 复位
DBGCONRST	Bit24	R/W	DBGCON 复位 0: 无操作 1: 复位
IWDTRST	Bit23	R/W	IWDT 复位 0: 无操作 1: 复位
WWDTRST	Bit22	R/W	WWDT 复位 0: 无操作 1: 复位
I2C1RST	Bit 21	R/W	I2C1 复位 0: 无操作 1: 复位
I2C0RST	Bit 20	R/W	I2C0 复位

			0: 无操作 1: 复位
Reserved	Bit 19-18	—	保留
SPI1RST	Bit 17	R/W	SPI1 复位 0: 无操作 1: 复位
SPI0RST	Bit 16	R/W	SPI0 复位 0: 无操作 1: 复位
Reserved	Bit 15	—	保留
CUART2RST	Bit 14	R/W	CUART2 复位 (UART4) 0: 无操作 1: 复位
CUART1RST	Bit 13	R/W	CUART1 复位 (UART3) 0: 无操作 1: 复位
CUART0RST	Bit 12	R/W	CUART0 复位 (UART2) 0: 无操作 1: 复位
Reserved	Bit 11-10	—	保留
EUART1RST	Bit 9	R/W	EUART1 复位 (UART1) 0: 无操作 1: 复位
EUART0RST	Bit 8	R/W	EUART0 复位 (UART0) 0: 无操作 1: 复位
Reserved	Bit 7-5	—	保留
GP16C4T2RST	Bit 4	R/W	GP16C4T2 复位 0: 无操作 1: 复位
GP16C4T1RST	Bit 3	R/W	GP16C4T1 复位 0: 无操作 1: 复位
GP16C4T0RST	Bit 2	R/W	GP16C4T0 复位 0: 无操作 1: 复位
BS16T0RST	Bit 1	R/W	BS16T0 复位 0: 无操作 1: 复位
AD16C4T0RST	Bit 0	R/W	AD16C4T0 复位 0: 无操作 1: 复位

第9章 时钟管理（CMU）

9.1 概述

时钟管理单元（CMU）的作用是控制时钟和振荡器。MCU 各外设时钟可独立配置。外设时钟的灵活配置可以有效降低系统功耗。

9.2 特性

- ◆ 支持多种时钟源
 - ◇ 1~12MHz 外部高速晶体振荡（HOSC）
 - ◇ 48/4MHz 两个独立内部高速 RC 振荡器（HRC48M 和 HRC4M）
 - ◇ 32KHz 内部低速 RC 振荡器（LRC）
 - ◇ 48/64/72MHz 内部 PLL 高速时钟
- ◆ 支持低功耗配置
- ◆ 系统时钟、APB 外设时钟可分别预分频
- ◆ 内核和外设均支持独立的时钟门控
- ◆ 支持系统时钟输出可配

9.3 结构框图

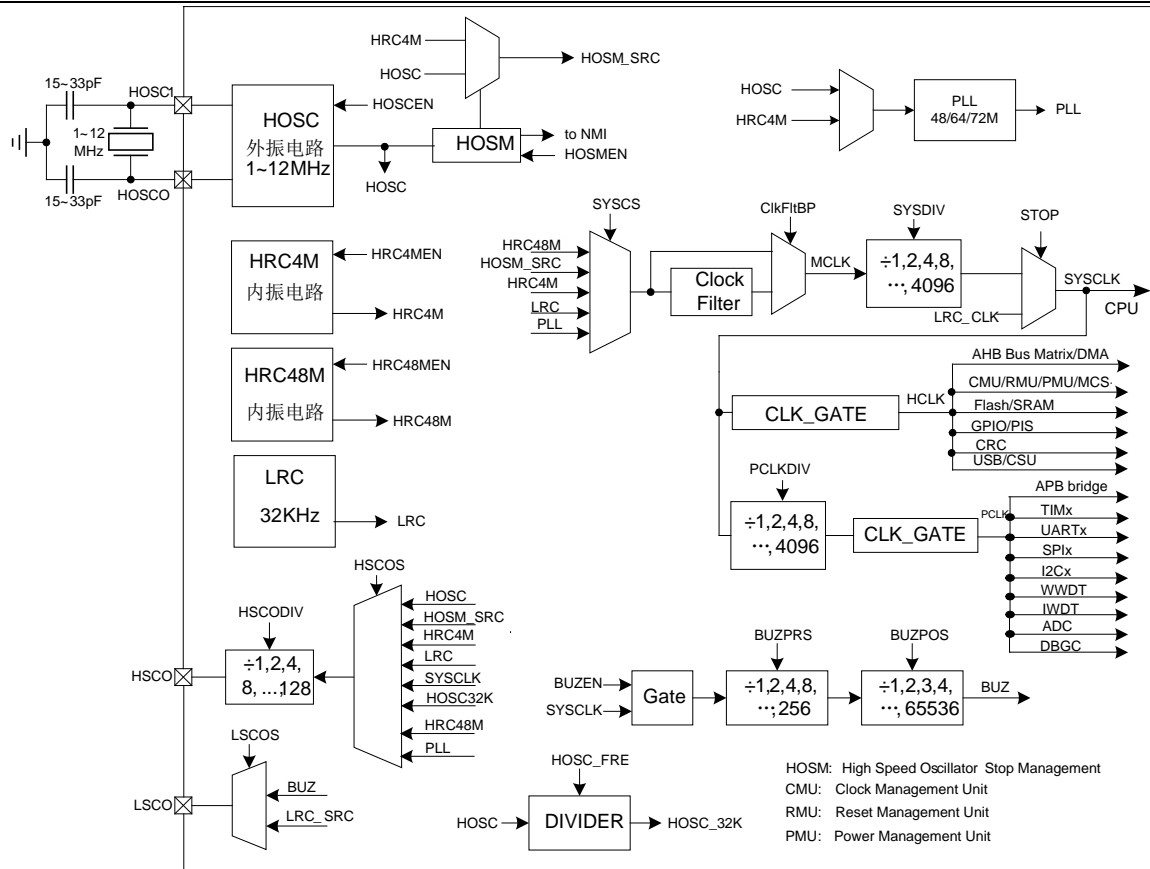


图 9-1 时钟管理结构图

9.4 功能描述

9.4.1 外部高速振荡器时钟 (HOSC)

HOSC 高速振荡电路可驱动 1~12MHz 晶体振荡器和陶瓷振荡器。驱动晶体振荡器时需要匹配 15~33pF 电容，具体电容值需根据外接晶振的参数需求确定，驱动陶瓷振荡器时不需要匹配电容。HOSC 内部自带反馈电阻，驱动外置振荡器不需要外接电阻。

芯片内置外部振荡器时钟输入滤波器，通过设置寄存器 CMU_CLKENR 的 HOSCFBPS 位为 1，可旁路该滤波器，应用中可不旁路该滤波器，以滤除因外界强干扰因素可能产生的时钟毛刺，保障时钟信号稳定。

当不外接晶体振荡器，而是通过外部时钟输入管脚 HOSC_IN 外灌时钟信号时，需设置寄存器 CMU_CLKENR 的 HOSCBPS 位为 1，旁滤 HOSC 振荡器电路，其他情况下保持 HOSCBPS 位为 0 即可。

当使用晶体振荡器时，需外接匹配电容。振荡器电路示意图如下：

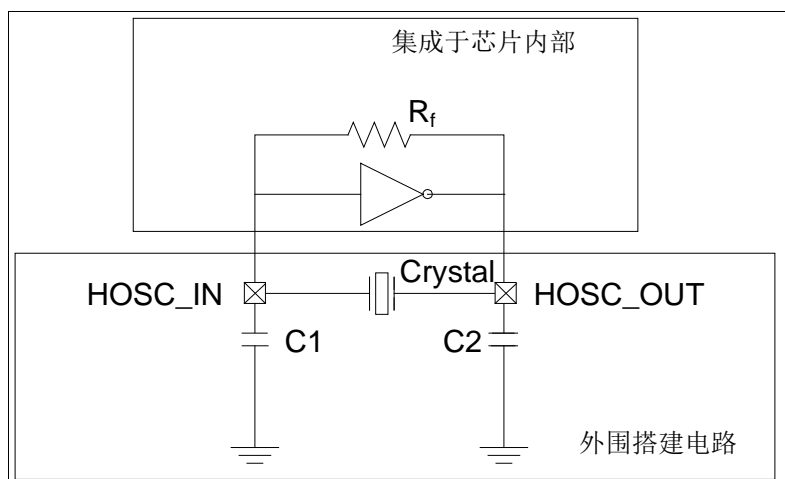


图 9-2 HOSC 电路图

晶体振荡器的等效电路如下图所示：

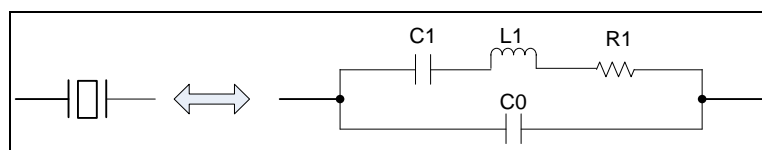


图 9-3 晶体振荡器等效电路示意图

注 1: C1: 动态电容; L1: 动态电感; R1 动态电阻; C0: 静电容; 等效串联电阻 $ESR=R1 \times (1+C0/CL)^2$, CL 为负载电容。

注 2: 晶体振荡器与芯片配合使用时，为达到理想的晶振起振和稳定工作状态，对 1~8MHz 晶振参数选型时，推荐 $ESR \leq 200 \Omega$, $CL \leq 16pF$ (晶振的负载电容参数 $\leq 16pF$)；对 9~12MHz 晶振参数选型时，推荐 $ESR \leq 50 \Omega$, $CL \leq 16pF$ (晶振的负载电容参数 $\leq 16pF$)。

9.4.2 内部高速RC振荡器时钟（HRC4M、HRC48M）

内部两路高速 RC 振荡器 HRC4M 和 HRC48M，分别输出 4MHz、48MHz 两种频率时钟。

HRC48M 振荡器默认支持时钟输入滤波器，用于滤除因外界强干扰因素可能产生的时钟毛刺，通过设置寄存器 CMU_CLKENR 的 HRC48FBPS 位为 1，可旁路该滤波器，不对 HRC48M 时钟信号滤波，在应用中建议不旁路该滤波器。

9.4.3 内部低速RC振荡器时钟（LRC）

内部低速 RC 振荡器 LRC 可输出 32KHz 时钟。LRC 固定为常开，不能关闭。

9.4.4 内部倍频时钟（PLL）

HRC 和 HOSC 时钟源通过内部倍频器 PLL 可将时钟倍频至 48MHz，64MHz 或 72MHz，PLL 时钟源有：

- ◇ HRC4M
- ◇ HOSC4M
- ◇ HOSC8M

通过 PLL 配置寄存器 CMU_PLLCFG，可设置 PLL 时钟源和输出时钟频率。

PLL 支持失锁安全管理机制，可通过 CMU_PULMCR 寄存器的 EN 位使能，当使用 PLL 倍频时钟作为系统时钟时，推荐使能 PLL 失锁安全管理，并设置为 PLL 失锁后将系统时钟切换至 HRC4M，可提升芯片运行时的抗干扰能力。

9.4.5 系统时钟选择

通过 CMU 控制状态寄存器（CMU_CSR）的 SYS_CMD 位进行配置选择和切换时钟。

系统时钟源有：

- ◇ HRC4M
- ◇ HRC48M
- ◇ LRC
- ◇ 带安全管理的 HOSC，当 HOSC 停振时硬件自动切换至 HRC4M
- ◇ 带安全管理的 PLL，可配置 PLL 失锁时硬件自动切换至 HRC4M
- ◇ 系统时钟可通过 CMU_CFGR 寄存器进行分频

系统时钟支持时钟输入滤波器，该滤波器默认为旁路，可通过向寄存器 CMU_CSR 的 CFT_CMD 位写入 0x55 来选择使用该滤波器，当系统时钟大于 48MHz 时，需旁路该系统时钟滤波器，否则可能会造成系统时钟失效。

9.4.6 时钟安全管理

高速振荡器安全管理 HOSM

HOSM 时钟安全系统可以通过软件使能（CMU_HOSMCR 寄存器中的 EN 位为 1）。HOSM 时钟安全监测机制在 HOSC 振荡器启动并稳定后被激活，当软件将 HOSC 振荡器关闭时，安全监测机制也将关闭。

当 HOSC 作为系统时钟时，如果 HOSC 发生故障，系统时钟将自动切换至 HRC4M（需

使能)。时钟故障事件会被送到高级定时器的刹车输入端，且时钟停振中断标志位 (CMU_HOSMCR.STPIF) 被置位，如果时钟停振中断使能 (CMU_HOSMCR.STPIE=1)，则产生时钟安全中断请求，允许软件完成营救操作。若 NMI 中断使能 (CMU_HOSMCR.NMIE=1)，则时钟停振事件将产生 NMI 不可屏蔽中断。当 HOSC 时钟恢复正常后，系统时钟自动切换回 HOSC，且时钟起振中断标志 (CMU_HOSMCR.STRIF) 被置位，如果时钟起振中断使能 (CMU_HOSMCR.STRIE=1)，则产生时钟安全中断请求。

PLL 失锁安全管理 PULM

PULM 失锁安全管理通过将 CMU_PULMCR 寄存器的 EN 位设为 1 使能，一旦使能，PULM 失锁安全监测机制将在 PLL 启动后被激活，若软件将 PLL 关闭，则 PULM 安全监测机制关闭。如果 PLL 发生失锁，硬件对时钟的处理方式可通过 CMU_PULMCR 寄存器的 MODE 位进行选择。

PLL 发生失锁后，PLL 失锁中断标志位 (CMU_PULMCR.ULKIF) 被置起，如果 PLL 失锁中断使能 (CMU_PULMCR.ULKIE=1)，则产生时钟安全中断请求，允许软件完成营救操作。若 NMI 中断使能 (CMU_PULMCR.NMIE=1)，则 PLL 失锁事件将产生 NMI 不可屏蔽中断。当 PLL 锁频时钟恢复正常后，锁定中断标志位 (CMU_PULMCR.LCKIF) 被置位，如果锁定中断使能 (CMU_PULMCR.LCKIE=1)，则产生时钟安全中断请求。

9.5 特殊功能寄存器

9.5.1 寄存器列表

CMU 寄存器列表		
名称	偏移地址	描述
CMU_CSR	000 _H	CMU 控制状态寄存器
CMU_CFGR	004 _H	CMU 配置寄存器
Reserved	008 _H ~00C _H	保留
CMU_CLKENR	010 _H	CMU 时钟使能寄存器
CMU_CLKSR	014 _H	CMU 时钟状态寄存器
CMU_PLLCFG	018 _H	PLL 配置寄存器
CMU_HOSCCFG	01C _H	HOSC 配置寄存器
CMU_HOSMCR	020 _H	HOSC 安全管理控制寄存器
Reserved	024 _H	保留
CMU_PULMCR	028 _H	PLL 失锁管理控制寄存器
Reserved	02C _H	保留
CMU_CLKOCR	030 _H	CMU 时钟输出控制寄存器
CMU_BUZZCR	034 _H	BUZZ 控制寄存器
Reserved	038 _H ~03C _H	保留
CMU_AHBENR	040 _H	AHB 外设时钟使能寄存器
Reserved	044 _H ~04C _H	保留
CMU_APBENR	050 _H	APB 外设时钟使能寄存器
Reserved	054 _H ~05C _H	保留
CMU_LPENR	060 _H	外设时钟低功耗模式使能寄存器

9.5.2 寄存器描述

9.5.2.1 CMU控制状态寄存器 (CMU_CSR)

CMU 控制状态寄存器 (CMU_CSR)																															
偏移地址: 000 _H																															
复位值: 00000001_00000000_00000001_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CFT_RDYN	CFT_STU	CFT_CMD								Reserved			SYS_RDYN	Reserved		SYS_STU			Reserved						SYS_CMD

Reserved	Bit 31-26	—	保留
CFT_RDYN	Bit 25	R	系统时钟滤波器切换状态位 0: 系统时钟滤波器切换完成或未发生切换动作 1: 系统时钟滤波器正在切换 注: 系统时钟滤波器在切换时仅需若干个系统时钟, 软件读取时不保证能读到系统时钟滤波器正在切换的状态
CFT_STU	Bit 24	R	系统时钟滤波器激活状态位 0: 系统时钟滤波器被选择 1: 系统时钟滤波器被旁路
CFT_CMD	Bit 23-16	W	系统时钟滤波切换命令位 0x55: 选择系统时钟滤波器 0xAA: 旁路系统时钟滤波器 其他: 无操作 注1: 系统默认为旁路系统时钟滤波器。 注2: 当系统时钟滤波器正在切换时, 该位写入无效。该位读出始终为0。 注3: 当系统时钟大于48MHz时, 需旁路系统时钟滤波器, 否则可能会造成系统时钟失效。
Reserved	Bit 15-13	—	保留
SYS_RDYN	Bit 12	R	系统时钟切换状态位 0: 系统时钟切换完成或未发生切换动作 1: 系统时钟正在切换 注: 系统时钟在切换时仅需若干个系统时钟, 软件读取时不保证能读到系统时钟正在切换的状态
Reserved	Bit 11	—	保留
SYS_STU	Bit 10-8	R	当前系统时钟状态位 000: 保留 001: 选择HRC48M 010: 选择LRC 011: 选择HOSC 100: 选择PLL

			101: 选择HRC4M 其他: 保留
Reserved	Bit 7-3	—	保留
SYS_CMD	Bit 2-0	W	<p>系统时钟切换命令位</p> <p>000: 无操作</p> <p>001: 选择HRC48M</p> <p>010: 选择LRC</p> <p>011: 选择HOSC</p> <p>100: 选择PLL</p> <p>101: 选择HRC4M</p> <p>其他: 保留 (选择HRC48M)</p> <p>注1: 系统默认为选择HRC48M, 在全温度范围内, HRC4M的频率温漂特性优于HRC48M。</p> <p>注2: 当系统时钟正在切换时, 该位写入无效。该位读出始终为0。</p>

注 1: 因 HRC48M 相对于 HRC4M 的温漂偏差大, 所以对时钟频率精度要求高的非 USB 应用, 推荐使用 HRC4M, 对时钟频率精度要求低的应用或者 USB 应用, 可使用 HRC48M, 在 USB 应用中可通过时钟同步单元 CSU 对 HRC48M 进行实时校准, 得到高精度频率。

注 2: 进行系统时钟切换操作时, 需先使能并等待目标时钟源稳定后 (可判断 CMU_CLKSR 寄存器的对应时钟稳定标志位), 再设置 SYS_CMD 位, 进行系统时钟切换。

注 3: 在软件进行系统时钟切换到新的时钟源时, 需先将系统时钟 SYSCLK 设置为 1 分频 (CMU_CFGR 寄存器的 SYSDIV 位为 0), 然后再执行系统时钟切换命令, 否则可能会导致原时钟源无法被关闭。

9.5.2.2 CMU配置寄存器 (CMU_CFGR)

CMU 配置寄存器 (CMU_CFGR)																															
偏移地址: 004 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PCLKDIV				SYSDIV				Reserved						USBSW	Reserved		I2SSW		

Reserved	Bit 31-20	—	保留
PCLKDIV	Bit 19-16	R/W	PCLK 分频选择位 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 1100: 4096分频 其他: 保留
SYSDIV	Bit 15-12	R/W	SYSDIV 分频选择位 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 1100: 4096分频 其他: 保留
Reserved	Bit11-5	—	保留
USBSW	Bit 4	R/W	选择 USB 时钟 (USBCLK) 来源 0: 选择 HRC48M 作为 USB 时钟 1: 选择 PLLCLK 作为 USB 时钟 (PLL 配置为 48MHz) 如果选择内部 HRC48M 做为 USB 时钟, 建议同时使能 CSU, 通过 SOF 来实时校准 HRC48M。 如果需要高精度的 USB 时钟, 则建议使用 PLLCLK, 并使用外部 HOSC 作为 PLL 时钟源。
Reserved	Bit3-1	—	保留
I2SSW	Bit0	R/W	选择 I2S 时钟 (I2SCLK) 来源 0: 选择 HOSCCLK 作为 I2S 时钟 1: 选择 PLLCLK 作为 I2S 时钟

注: 在软件进行系统时钟切换到新的时钟源时, 需先将系统时钟 SYSCLK 设置为 1 分频 (CMU_CFGR 寄存器的 SYSDIV 位为 0), 然后再执行系统时钟切换命令, 否则可能会导致原时钟源无法被关闭。

9.5.2.3 CMU时钟使能寄存器 (CMU_CLKENR)

CMU 时钟使能寄存器 (CMU_CLKENR)																															
偏移地址: 010 _H																															
复位值: 00000000_00000000_00000000_00000110 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																					HOSCFBPS	HOSCBPS	HOSC32K	USBCKEN	I2SCKEN	Reserved	HRC48FBPS	PLLEN	HRC4MEN	HRC48MEN	HOSCEN

Reserved	Bit 31-11	—	保留
HOSCFBPS	Bit 10	R/W	HOSC 时钟滤波器旁路 (Bypass) 使能位 0: 禁止 1: 使能 (旁路滤波器, 不对时钟信号滤波)
HOSCBPS	Bit 9	R/W	HOSC 振荡器旁路 (Bypass) 使能位 0: 禁止 1: 使能 (当不外接晶振, 而是通过外部时钟输入管脚 HOSC_IN 外灌时钟信号时, 需设置为使能)
HOSC32K	Bit 8	R/W	HOSC32K 软件使能位 0: 禁止 1: 使能
USBCKEN	Bit 7	R/W	USB 48M 软件使能位 0: 禁止 1: 使能 (读写USB模块寄存器时, 需先使能该位)
I2SCKEN	Bit 6	R/W	I2S 软件使能位 0: 禁止 1: 使能
Reserved	Bit 5	—	保留
HRC48FBPS	Bit 4	R/W	HRC48M 时钟滤波器旁路 (Bypass) 使能位 0: 禁止 1: 使能 (旁路滤波器, 不对时钟信号滤波)
PLLEN	Bit 3	R/W	PLL 软件使能位 0: 禁止 1: 使能
HRC4MEN	Bit 2	R/W	HRC4M 软件使能位 0: 禁止 1: 使能
HRC48MEN	Bit 1	R/W	HRC48M 软件使能位 0: 禁止 1: 使能
HOSCEN	Bit 0	R/W	HOSC 软件使能位 0: 禁止

			1: 使能
--	--	--	-------

9.5.2.4 CMU时钟状态寄存器 (CMU_CLKSR)

CMU 时钟状态寄存器 (CMU_CLKSR)																															
偏移地址: 014 _H																															
复位值: 00000000_0000xx00_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							PLLRDY	Reserved							LRCRDY	HRC48MRDY	HRC4MRDY	HOSCRDY	Reserved							HRC48MACT	HRC4MACT	PLLACT	HOSCACT		

Reserved	Bit 31-25	—	保留
PLLRDY	Bit 24	R	PLL 稳定标志位 0: 未稳定或未激活 1: 稳定
Reserved	Bit 23-20	—	保留
LRCRDY	Bit 19	R	LRC 稳定标志位 0: 未稳定或未激活 1: 稳定
HRC48MRDY	Bit 18	R	HRC48M 稳定标志位 0: 未稳定或未激活 1: 稳定
HRC4MRDY	Bit 17	R	HRC4M 稳定标志位 0: 未稳定或未激活 1: 稳定
HOSCRDY	Bit 16	R	HOSC 稳定标志位 0: 未稳定或未激活 1: 稳定
Reserved	Bit 15-4	—	保留
HRC48MACT	Bit 3	R	HRC48M 激活状态位 0: 未激活 1: 激活
HRC4MACT	Bit 2	R	HRC4M 激活状态位 0: 未激活 1: 激活
PLLACT	Bit 1	R	PLL 激活状态位 0: 未激活 1: 激活
HOSCACT	Bit 0	R	HOSC 激活状态位 0: 未激活 1: 激活

9.5.2.5 PLL配置寄存器 (CMU_PLLCFG)

PLL 配置寄存器 (CMU_PLLCFG)																															
偏移地址: 018 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												REFS		CLKOS	

Reserved	Bit 31-4	—	保留
REFS	Bit 3-2	R/W	PLL 参考时钟选择位 00: HRC4M 01: HOSC4M 10: HOSC8M 11: 保留
CLKOS	Bit 1-0	R/W	PLL 时钟输出选择位 00: 72MHz 01: 72MHz 10: 64MHz 11: 48MHz

注: CMU_PLLCFG 必须在 PLL 使能前配置, 不建议在 PLL 使能后再动态修改。

9.5.2.6 HOSC配置寄存器 (CMU_HOSCCFG)

HOSC 配置寄存器 (CMU_HOSCCFG)																															
偏移地址: 01C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												FREQ			

Reserved	Bit 31-3	—	保留
FREQ	Bit 2-0	R/W	HOSC 频率配置位 0x0: 4MHz 0x1: 8MHz 0x2: 12MHz 0x3: 保留 0x4: 保留 0x5: 保留

			其他：4MHz
--	--	--	---------

注：CMU_HOSCCFG 用于将 HOSC 分出 32KHz 给 CSU。

9.5.2.7 HOSC安全管理控制寄存器 (CMU_HOSMCR)

HOSC 安全管理控制寄存器 (CMU_HOSMCR)																															
偏移地址：020 _H																															
复位值：00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												NMIE	STPIF	STRIF	STPIE	STRIE	Reserved						FRQS	Reserved						FLAG	EN

Reserved	Bit 31-21	—	保留
NMIE	Bit 20	R/W	HOSC安全事件不可屏蔽中断使能位 0: 禁止 1: 使能
STPIF	Bit 19	R/W	HOSC停振中断标志位 0: 未发生 HOSC 停振或标志位已被清除 1: HOSC振荡器异常, 发生HOSC停振 注: 该位写1清除, 写0无效
STRIF	Bit 18	R/W	HOSC起振中断标志位 0: 未发生 HOSC 起振或标志位已被清除 1: HOSC停振因素消除后, 发生HOSC起振 注: 该位写1清除, 写0无效
STPIE	Bit 17	R/W	HOSC停振中断使能位 0: 禁止 1: 使能
STRIE	Bit 16	R/W	HOSC起振中断使能位 0: 禁止 1: 使能
Reserved	Bit 15-11	—	保留
FRQS	Bit 10-8	R/W	HOSC 频率配置位 000: 1~2MHz 001: 2~4MHz 010: 4~8MHz 011: 8~12MHz 1xx: 保留 注: 当频率处在两个相邻档位的临界值时, 这两个档位可任意选择
Reserved	Bit 7-2	—	保留
FLAG	Bit 1	R	HOSC 停振状态位

			0: HOSC 正常 1: HOSC停振
EN	Bit 0	R/W	HOSC 安全管理使能位 0: 禁止 1: 使能

注: HOSC 停振标志 STPIF、起振标志 STRIF 和状态 FLAG, 仅在 HOSC 安全管理使能 (CMU_HOSMCR .EN=1) 并且 HOSC 使能 (CMU_CLKENR. HOSCEN=1) 时有效, 在时钟管脚无外接晶振或存在外部干扰的情况下, 上述标志均可能会失效。

9.5.2.8 PLL失锁管理控制寄存器 (CMU_PULMCR)

PLL 失锁管理控制寄存器 (CMU_PULMCR)																															
偏移地址: 028 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											NMIE	ULKIF	LCKIF	ULKIE	LCKIE	Reserved						MODE	Reserved						CLKS	EN	

Reserved	Bit 31-21	—	保留
NMIE	Bit 20	R/W	PLL安全事件不可屏蔽中断使能位 0: 禁止 1: 使能
ULKIF	Bit 19	R/C_W1	PLL失锁中断标志位 0: 未发生 PLL 失锁或标志位已被清除 1: 发生PLL失锁 注: 该位写1清除, 写0无效
LCKIF	Bit 18	R/C_W1	PLL锁定中断标志位 0: 未发生 PLL 锁定或标志位已被清除 1: 发生PLL锁定 注: 该位写1清除, 写0无效
ULKIE	Bit 17	R/W	PLL失锁中断使能位 0: 禁止 1: 使能
LCKIE	Bit 16	R/W	PLL锁定中断使能位 0: 禁止 1: 使能
Reserved	Bit 15-10	—	保留
MODE	Bit 9-8	R/W	PLL失锁管理模式选择位 00: 失锁后无操作 01: 失锁后切换至HRC4M 1x: 失锁后切换至HRC4M, 待重新锁定后自动切

			回PLL 注：如果设置为PLL失锁后切换至HRC4M，则需保持HRC4M为使能（HRC4MEN=1）
Reserved	Bit 7-2	—	保留
CLKS	Bit 1	R	当前选择时钟状态位 0：选择 PLL 1：选择HRC4M
EN	Bit 0	R/W	PLL 失锁管理使能位 0：禁止 1：使能

注：当使用 PLL 倍频时钟作为系统时钟时，推荐使能 PLL 失锁安全管理，并设置为 PLL 失锁后将系统时钟切换至 HRC4M，可提升芯片运行时的抗干扰能力。

9.5.2.9 CMU时钟输出控制寄存器（CMU_CLKOCR）

CMU 时钟输出控制寄存器（CMU_CLKOCR）																																
偏移地址：030 _H																																
复位值：00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved						LSCOS		Reserved								LSCOEN	Reserved	HSCODIV			HSCOS					Reserved						HSCOEN

Reserved	Bit 31-26	—	保留
LSCOS	Bit 25-24	R/W	低速时钟输出源选择位 00：LRC 01：BUZZ 其余：保留
Reserved	Bit 23-17	—	保留
LSCOEN	Bit 16	R/W	低速时钟输出使能位 0：禁止 1：使能
Reserved	Bit 15	—	保留
HSCODIV	Bit 14-12	R/W	高速时钟输出分频选择位 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

HSCOS	Bit 11-8	R/W	高速时钟输出源选择位 000: HOSC 001: HOSM_SRC 010: HRC4M 011: LRC 100: SYSCLK 101: HOSC32K 110: HRC48M 111: PLL 时钟 注: HOSM_SRC 为 HOSC 时钟安全管理模块 HOSM 的输出时钟
Reserved	Bit 7-1	—	保留
HSCOEN	Bit 0	R/W	高速时钟输出使能位 0: 禁止 1: 使能

9.5.2.10 BUZZ控制寄存器 (CMU_BUZZCR)

BUZZ 控制寄存器 (CMU_BUZZCR)																															
偏移地址: 034 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT																Reserved				DIV		Reserved				EN					

DAT	Bit 31-16	R/W	BUZZ频率数据值 $Freq_{BUZZ} = \frac{Freq_{SYSCLK}}{2^{DIV+1} \times (DAT + 1)}$
Reserved	Bit 15-11	—	保留
DIV	Bit 10-8	R/W	BUZZ 时钟分频选择位 000: 2 分频 001: 4 分频 010: 8 分频 011: 16 分频 100: 32 分频 101: 64 分频 110: 128 分频 111: 256 分频
Reserved	Bit 7-1	—	保留
EN	Bit 0	R/W	BUZZ 使能位 0: 禁止 1: 使能

9.5.2.11 AHB外设时钟使能寄存器 (CMU_AHBENR)

AHB 外设时钟使能寄存器 (CMU_AHBENR)																															
偏移地址: 040 _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CSUEN	USBEN	PISEN	Reserved	DMAEN	CRCEN	GPIOEN	

Reserved	Bit 31-8	—	保留
CSUEN	Bit 7	R/W	CSU 时钟使能位 0: 禁止 1: 使能
USBEN	Bit 6	R/W	USB 时钟使能位 0: 禁止 1: 使能
PISEN	Bit 5	R/W	PIS 时钟使能位 0: 禁止 1: 使能
Reserved	Bit 4-3	—	保留
DMAEN	Bit 2	R/W	DMA 时钟使能位 0: 禁止 1: 使能
CRCEN	Bit 1	R/W	CRC 时钟使能位 0: 禁止 1: 使能
GPIOEN	Bit 0	R/W	GPIO 时钟使能位 0: 禁止 1: 使能

9.5.2.12 APB外设时钟使能寄存器 (CMU_APBENR)

APB 外设时钟使能寄存器 (CMU_APBENR)																															
偏移地址: 050 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ADCEN	DBGCCEN	IWDTCEN	WWDTCEN	I2C1EN	I2C0EN	Reserved	SPI1EN	SPI0EN	Reserved	CUART2EN	CUART1EN	CUART0EN	Reserved	EUART1EN	EUART0EN	Reserved	GP16C4T2EN	GP16C4T1EN	GP16C4T0EN	BS16T0EN	AD16C4T0EN				

Reserved	Bit 31-26	—	保留
ADCEN	Bit25	R/W	ADC 时钟使能位 0: 禁止 1: 使能
DBGECEN	Bit24	R/W	DBGEC 时钟使能位 0: 禁止 1: 使能
IWDTEN	Bit23	R/W	IWDT 时钟使能位 0: 禁止 1: 使能
WWDTEN	Bit22	R/W	WWDT 时钟使能位 0: 禁止 1: 使能
I2C1EN	Bit 21	R/W	I2C1 时钟使能位 0: 禁止 1: 使能
I2C0EN	Bit 20	R/W	I2C0 时钟使能位 0: 禁止 1: 使能
Reserved	Bit 19-18	—	保留
SPI1EN	Bit 17	R/W	SPI1 时钟使能位 0: 禁止 1: 使能
SPI0EN	Bit 16	R/W	SPI0 时钟使能位 0: 禁止 1: 使能
Reserved	Bit 15	—	保留
CUART2EN	Bit 14	R/W	CUART2 时钟使能位 (UART4) 0: 禁止 1: 使能
CUART1EN	Bit 13	R/W	CUART1 时钟使能位 (UART3) 0: 禁止 1: 使能
CUART0EN	Bit 12	R/W	CUART0 时钟使能位 (UART2) 0: 禁止 1: 使能
Reserved	Bit11-10	—	保留
EUART1EN	Bit 9	R/W	EUART1 时钟使能位 (UART1) 0: 禁止 1: 使能
EUART0EN	Bit 8	R/W	EUART0 时钟使能位 (UART0) 0: 禁止 1: 使能

Reserved	Bit7-5	—	保留
GP16C4T2EN	Bit 4	R/W	GP16C4T2 时钟使能位 0: 禁止 1: 使能
GP16C4T1EN	Bit 3	R/W	GP16C4T1 时钟使能位 0: 禁止 1: 使能
GP16C4T0EN	Bit 2	R/W	GP16C4T0 时钟使能位 0: 禁止 1: 使能
BS16T0EN	Bit 1	R/W	BS16T0 时钟使能位 0: 禁止 1: 使能
AD16C4T0EN	Bit 0	R/W	AD16C4T0 时钟使能位 0: 禁止 1: 使能

9.5.2.13 外设时钟低功耗模式使能寄存器 (CMU_LPENR)

外设时钟低功耗模式使能寄存器 (CMU_LPENR)																															
偏移地址: 060 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												HOSCEN	HRC48MEN	HRC4MEN	PLLEN

Reserved	Bit 31-4	—	保留
HOSCEN	Bit 3	R/W	STOP 模式 HOSC 时钟使能位 0: 禁止 1: 使能
HRC48MEN	Bit 2	R/W	STOP 模式 HRC48M 时钟使能位 0: 禁止 1: 使能
HRC4MEN	Bit 1	R/W	STOP 模式 HRC4M 时钟使能位 0: 禁止 1: 使能
PLLEN	Bit 0	R/W	STOP 模式 PLL 时钟使能位 0: 禁止 1: 使能

第10章 时钟同步单元（CSU）

10.1 概述

时钟同步单元（CSU）是一个时钟调校数字控制器，用于实时调校片内高速振荡器 HRC48M，CSU 提供一种有效的方法来评估振荡器的输出频率，通过振荡器的输出信号与选择的同步时钟源比较，计算出振荡器频率的快慢。CSU 可以根据评估结果自动校准振荡器的频率。

CSU 非常适合 USB 作为设备模式下的应用，在这种应用情境为 USB 提供精准的时钟，同步时钟源可选择从 USB 总线上的 SOF 封包获取，此封包由 USB 主机端以间隔精准的 1 毫秒发送；亦可选择通过 HOSC 振荡器输出（分频为 32KHz）、外部引脚输入或是由使用者的软件产生。

10.2 特性

- ◆ 支持预分频和极性选择的同步时钟源如下
 - ◇ 外部 CSU_SYNC 引脚（PA8）
 - ◇ HOSC 振荡器分频得到 32KHz 输出
 - ◇ USB 总线上的 SOF 信号
- ◆ 支持用户软件产生同步时钟
- ◆ 振荡器自动校准功能
- ◆ 具有自动记录计数值的 16 位频率偏差计数器
- ◆ 可屏蔽的中断
 - ◇ 理想同步
 - ◇ 校准值错误
 - ◇ 时钟同步异常
 - ◇ 时钟同步错误
 - ◇ 时钟同步警告
 - ◇ 时钟同步匹配

10.3 结构图

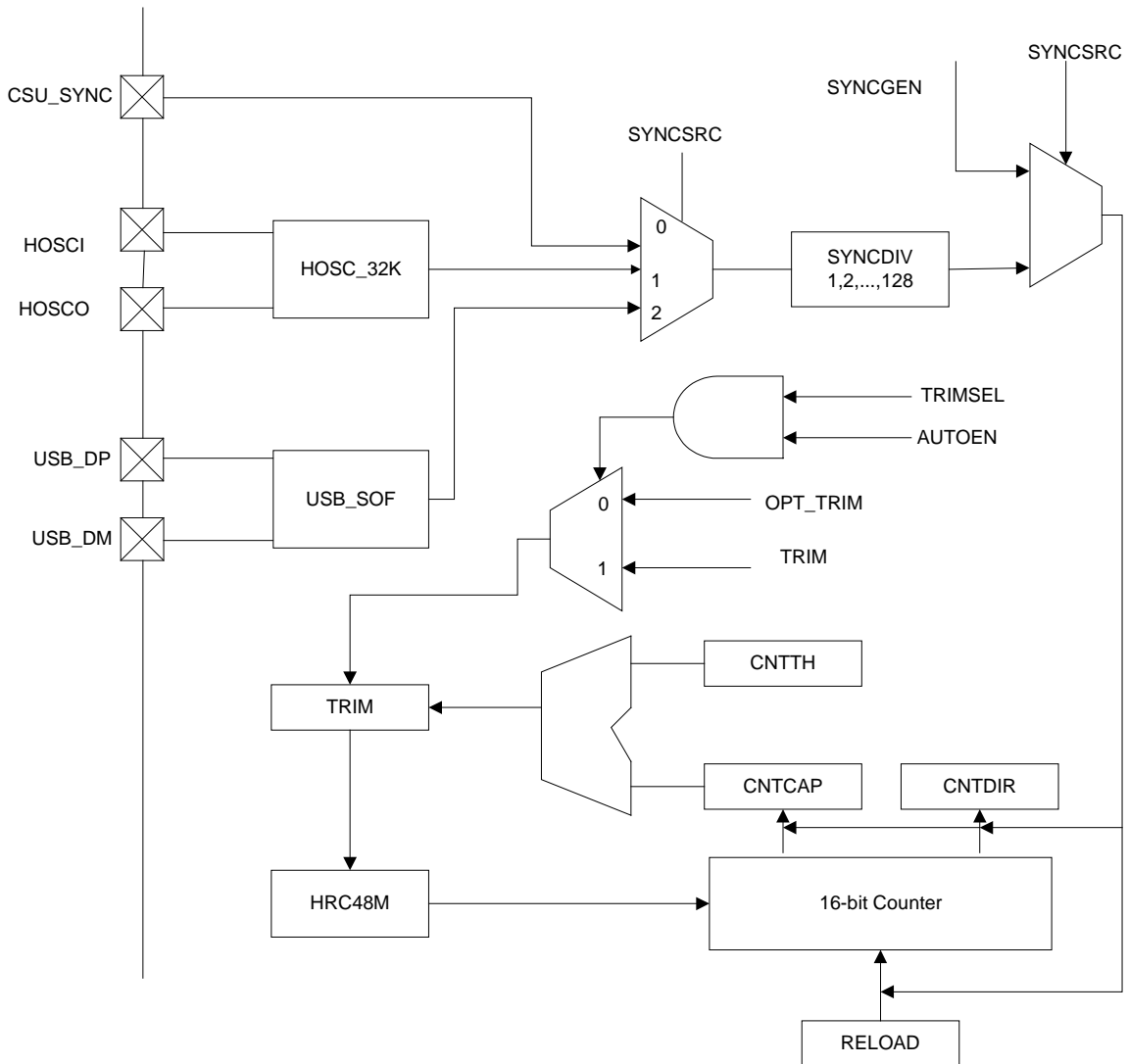


图 10-1 CSU 结构图

10.4 功能描述

10.4.1 同步信号输入

CSU 同步信号源可以通过 CSU_CFG.SYNCSRC 选择，可以选择外部 CSU_SYNC 引脚、HOSC 振荡器分频出 32KHz 时钟、USB 的 SOF 信号或是使用者软件所产生同步时钟。同步信号源若选择 CSU_SYNC 引脚、HOSC 振荡器分频出 32KHz 时钟或 USB 的 SOF 信号时，可以配置 CSU_CFG.POLSEL 选择上升沿触发或是下降沿触发作为计数器的触发事件，也可以配置 CSU_CFG.SYNCDIV 预分频，以得到适合的同步信号频率（建议为 1KHz）。

10.4.2 频率偏差计数器

频率偏差计数器是一个 16 位的递减/递增计数器，计数时钟为内部 HRC48M，当同步信号触发事件发生时，会将 CSU_CFG.RELOAD 重载并开始递减计数，当计数器计数到零时，如果同步信号触发事件再次发生，则会产生一个理想同步中断（FHIT），意味着振荡器的输出频率为精准的目标频率，否则计数器会从零开始递增计数，直到超过极限值（LIMIT）停止计数，并产生时钟同步异常中断（FFAULT），极限值为 CSU_CFG.CNTTH 乘以 128。

每次同步信号触发事件发生时，会自动记录频率偏差计数器的计数方向 CSU_STAT.CNTDIR 与计数值 CSU_STAT.CNTCAP，如果计数值 CNTCAP 小于容许值 CSU_CFG.CNTTH，则表示校准成功，否则需判断计数方向，如果频率偏差计数器为递减计数（CSU_STAT.CNTDIR 为 1），表示振荡器的输出频率低于目标频率，因此需要递增校准值 CSU_CON.TRIM，反之频率偏差计数器为递增计数（CSU_STAT.CNTDIR 为 0），表示振荡器的输出频率高于目标频率，因此需要递减校准值。

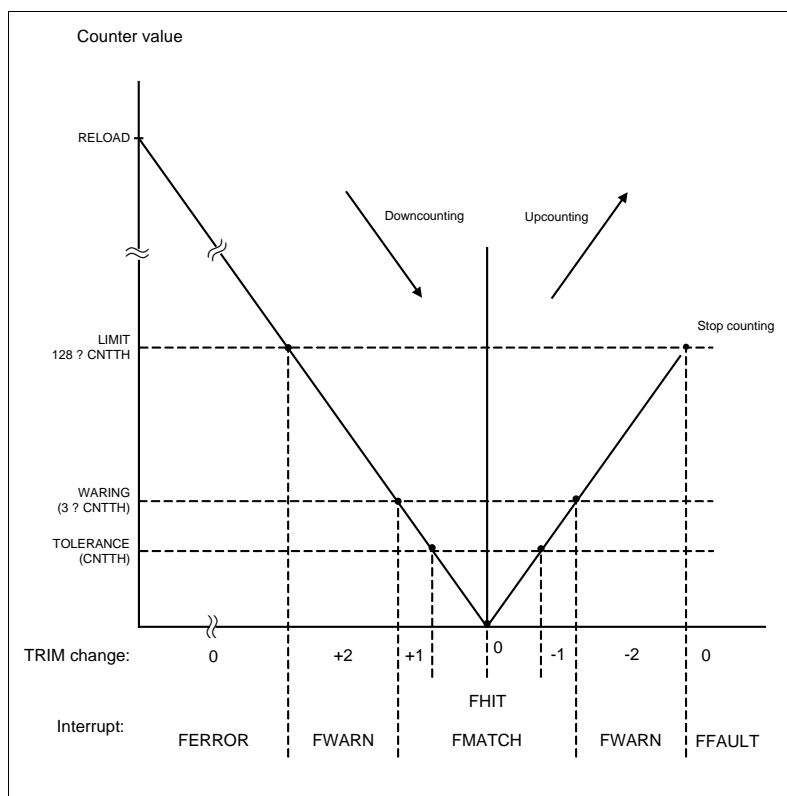


图 10-2 CSU 计数器操作

10.4.3 频率偏差评估与自动校准

频率偏差评估是通过下列三组数值来界定：

- ◆ 容许值 (TOLERANCE)，取决于 CSU 配置寄存器的计数容许值 CSU_CFG.CNTTH
- ◆ 警告值 (WARNING)，定义为 CNTTH 的 3 倍
- ◆ 极限值 (LIMIT)，定义为 CNTTH 的 128 倍

同步信号触发事件时计数器的计数值会自动记录在 CSU_STAT.CNTCAP，将计数值与上述三组数值作比较，以此判断振荡器的输出频率状态并且调整，用户可以使能自动校准功能（配置 CSU_CON.AUTOEN 为 1）来对 HRC48M 时钟频率进行自动校准，CSU 将依据振荡器的输出频率状态自动调整校准值 CSU_CON.TRIM：

- ◆ 当计数值小于容许值，表示振荡器的输出频率相当于目标频率，已校准成功
 - ◇ 时钟同步匹配状态 (FMATCH)
 - ◇ 自动校准模式时，不需要调整校准值
- ◆ 当计数值大于等于容许值且小于警告值，表示振荡器的输出频率接近目标频率，需进行频率细调
 - ◇ 时钟同步匹配状态 (FMATCH)
 - ◇ 自动校准模式时，需要微调校准值，每次只调整一个单位
- ◆ 当计数值大于等于警告值且小于极限值，表示振荡器的输出频率与目标频率有偏差，需进行频率粗调
 - ◇ 时钟同步警告状态 (FWARN)
 - ◇ 自动校准模式时，调整校准值，每次调整两个单位
- ◆ 当计数值大于等于极限值，表示振荡器的输出频率与目标频率差距过大，有可能是输入信号不干净或是没有接收到同步信号，校准失败
 - ◇ 时钟同步错误 (FERROR) / 时钟同步异常 (FFAULT)
 - ◇ 自动校准模式时，不调整校准值

注 1：如果校准值因自动校准调整到上溢出 (Overflow) 或是下溢出 (Underflow)，表示校准值已经达到极限无法继续校准，此时为校准值错误状态 (TRIMERR)。

注 2：当 CSU 在自动校准模式时，校准值寄存器 CSU_CON.TRIM 只能读取，不允许写入。

10.4.4 CSU初始化与配置

校准值选择 (TRIMSEL)

芯片的内部振荡器 HRC48M 时钟频率在出厂前都已经过校准, 并将校准值保存在 Option Byte, 用户可以通过配置 CSU_CFG.TRIMSEL 选择 CSU_STAT.OPT_TRIM 或 CSU_CON.TRIM 作为振荡器的初始校准值。

计数器重载值 (RELOAD)

计数器的重载值是根据目标频率与同步信号源分频后的频率计算的, 将两个频率相除后减一, 让计数器递减计数至零时达到理想的同步状态, 其公式如下:

$$RELOAD = \frac{f_{TARGET}}{f_{SYNC}} - 1$$

RELOAD 寄存器的复位值是以目标频率 48MHz 和同步信号源频率 1KHz(USB 的 SOF 信号) 计算, 即为 $48000-1=47999$ 。

计数容许值 (CNTTH)

计数容许值与内部振荡器 HRC48M 的频率调整曲线有着密切的关系, 在同步信号触发事件发生时, 由于计数器递减至零时为理想的同步状态, 因此理想的容许值应该为校准调整步长(step) 的一半, 其公式如下:

$$CNTTH = \frac{f_{TARGET}}{f_{SYNC}} \times \frac{STEP[\%]}{100\%} \div 2$$

为了达到最佳的校准值调整, 建议将 CNTTH 的结果四舍五入到最接近的整数。

CNTTH 寄存器的复位值是以 $(f_{TARGET}/f_{SYNC}) = 48000$ 且校准步长为理想值 0.199% 来计算的, 对应校准后的频率精度为 +0.0995% ($0.199\% \div 2$), 内部振荡器 HRC48M 的频率校准步长理想值约 0.199% (即校准值 TRIM 每变化 1 个单位, 对应的频率变化率约为 0.199%), 考虑到校准步长的非均匀性, 建议将 HRC48M 校准后的频率精度设置为理想步长值的 2.5 倍以上, 即 +0.5%, 对应的 CNTTH 寄存器值大于等于 $0xF0$ ($48000 \times 0.5\%$), 以免校准失败。

基于校准后的频率精度, 来计算计数容许值的公式如下:

$$CNTTH = \frac{f_{TARGET}}{f_{SYNC}} \times Accu[\%]$$

其中 Accu 为频率精度, 例如期望校准后的 HRC48M 频率精度为 +0.25%, 则 $Accu=0.25\%$ 。

注 1: CSU 并没有针对 RELOAD 和 CNTTH 错误配置的硬件保护, 错误的配置将导致自动校准调整不稳定与频率偏差评估错误。需要正确配置计数器重载值 (根据同步信号源频率 f_{SYNC} 计算), 其值也必须大于极限值 (CNTTH 的 128 倍)。

注 2: 上述 CNTTH 计算示例中, $f_{TARGET}=48MHz$, $f_{SYNC}=1KHz$ 。

10.4.5 CSU中断

时钟同步单元的中断是将所有中断事件合并成一个中断向量，发送至中断控制器，时钟同步单元的中断由六个寄存器控制。

◆ 中断控制

对 CSU 中断开启寄存器 CSU_IER 写 1 来开启中断请求事件。同样地，对 CSU 中断关闭寄存器 CSU_IDR 写 1 来关闭中断请求事件，IER 与 IDR 均是只允许写入 1 的寄存器。IER 与 IDR 设定的结果可通过 CSU 中断使能状态寄存器 CSU_IVS 判断，IVS 是一个只读寄存器。

◆ 中断标志位

CSU 中断标志寄存器 CSU_RIF 是一个只读寄存器，用来读取 CSU 的中断标志。该寄存器显示 CSU 中断真实状态。当有下列事件发生时，CSU 会产生中断

- ◇ 时钟同步匹配
- ◇ 时钟同步警告
- ◇ 时钟同步错误
- ◇ 时钟同步异常
- ◇ 校准值错误
- ◇ 理想同步

◆ 中断屏蔽后状态

通过寄存器 CSU_IVS 可判断中断请求是否被屏蔽，通过 CSU 中断屏蔽后状态寄存器 CSU_IFM 可读取 CSU 的屏蔽后的中断状态，该寄存器显示屏蔽后发生哪些中断，IFM 寄存器为 IVS 与 RIF 寄存器逻辑与运算后的结果。

◆ 中断清除

对 CSU 中断清除寄存器 CSU_ICR 写 1 来清除中断事件，ICR 是只允许写入 1 的寄存器。

10.5 特殊功能寄存器

10.5.1 寄存器列表

CSU 寄存器列表			
名称	偏移地址	类型	描述
CSU_CON	0000 _H	R/W	CSU 控制寄存器
CSU_CFG	0004 _H	R/W	CSU 配置寄存器
CSU_STAT	0008 _H	R	CSU 状态寄存器
CSU_IER	0010 _H	W1	CSU 中断开启寄存器
CSU_IDR	0014 _H	W1	CSU 中断关闭寄存器
CSU_IVS	0018 _H	R	CSU 中断使能状态寄存器
CSU_RIF	001C _H	R	CSU 中断标志寄存器
CSU_IFM	0020 _H	R	CSU 中断屏蔽后状态寄存器
CSU_ICR	0024 _H	C_W1	CSU 中断清除寄存器

10.5.2 寄存器描述

10.5.2.1 CSU控制寄存器 (CSU_CON)

CSU 控制寄存器 (CSU_CON)																															
偏移地址: 0x00																															
复位值: 0x0001 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															TRIM								Reserved			SYNCGEN	AUTOEN	CNTEN			

—	Bit 31-17	—	—
TRIM	Bit 16-8	R/W	<p>校准值</p> <p>HRC48M 的校准值, 可作为自动校准时的初始校准值, 该值每变化 1, 对应的频率约变化 95.54KHz (变化率为 0.199%), 校准值与频率成正比</p> <p>当开启自动校准功能 (AUTOEN = 1) 时, 校准值将自动被更新, 只允许读取, 不允许写入。</p> <p>校准值每变化 1, 对应的频率变化率典型值约 0.199%, 可能会偏大到约 0.9%。</p>
—	Bit 7-3	—	—
SYNCGEN	Bit 2	T_W1	<p>软件同步时钟触发位</p> <p>设定此位以产生一个脉冲 (Pulse) 做为同步时钟, 脉冲产生后, 此位将自动被清零</p> <p>0: 没有作用</p> <p>1: 产生一个脉冲</p>
AUTOEN	Bit 1	R/W	<p>自动校准使能位</p> <p>0: 关闭自动校准</p> <p>1: 开启自动校准</p>
CNTEN	Bit 0	R/W	<p>计数器使能位</p> <p>0: 关闭校准计数器</p> <p>1: 开启校准计数器</p>

10.5.2.2 CSU配置寄存器 (CSU_CFG)

CSU 配置寄存器 (CSU_CFG)																																	
偏移地址: 0x04																																	
复位值: 0x2030 BB7F																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
TRIMSEL		POLSEL		SYNCSRC		Reserved	SYNCDIV			CNTTH								RELOAD															

TRIMSEL	Bit 31	R/W	<p>校准值选择位</p> <p>设定此位选择自动校准时振荡器的校准初始值，HRC48M将载入此校准值</p> <p>0: 使用Option Byte校准值CSU_STAT.OPT_TRIM</p> <p>1: 使用自定义的校准值CSU_CON.TRIM</p> <p>当开启校准计数器 (CNTEN = 1) 时，只允许读取，不允许写入</p>
POLSEL	Bit 30	R/W	<p>同步时钟极性选择位</p> <p>0: 同步时钟上升沿作为触发事件</p> <p>1: 同步时钟下降沿作为触发事件</p> <p>当开启校准计数器 (CNTEN = 1) 时，只允许读取，不允许写入</p>
SYNCSRC	Bit 29-28	R/W	<p>同步时钟来源选择位</p> <p>00: GPIO 引脚 CSU_SYNC</p> <p>01: HOSC 振荡器分频出 32KHz 时钟</p> <p>10: USB 的 SOF 信号</p> <p>11: 软件产生</p> <p>当开启校准计数器 (CNTEN = 1) 时，只允许读取，不允许写入</p>
—	Bit 27	—	—
SYNCDIV	Bit 26-24	R/W	<p>同步时钟源预分频选择位</p> <p>000: 不分频</p> <p>001: 2 分频</p> <p>010: 4 分频</p> <p>011: 8 分频</p> <p>100: 16 分频</p> <p>101: 32 分频</p> <p>110: 64 分频</p>

			<p>111: 128 分频</p> <p>当开启校准计数器 (CNTEN = 1) 时, 只允许读取, 不允许写入</p>
CNTTH	Bit 23-16	R/W	<p>计数容许值</p> <p>设定此寄存器作为计数器的容许值, 此数值将作为计数值范围与时钟同步状态的评估, 该值可根据校准后 HRC48M 时钟频率的预期精度来计算</p> <p>当开启校准计数器 (CNTEN = 1) 时, 只允许读取, 不允许写入</p>
RELOAD	Bit 15-0	R/W	<p>计数器重载值</p> <p>设定此寄存器作为计数器的重载值, 当同步信号触发事件发生时, 频率偏差计数器会载入此数值, 然后开始递减计数, 计数时钟为内部 HRC48M</p> <p>当开启校准计数器 (CNTEN = 1) 时, 只允许读取, 不允许写入</p>

注: 内部振荡器 HRC48M 的频率校准步长理想值约 0.199%, 考虑到校准步长的非均匀性, 建议将 HRC48M 校准后的频率精度设置为理想步长值的 2.5 倍以上, 即+0.5%, 如果同步时钟信号频率为 1KHz, 则对应的 CNTTH 寄存器值建议大于等于 0xF0 (48000 x 0.5%), 以免校准失败。

10.5.2.3 CSU状态寄存器 (CSU_STAT)

CSU 状态寄存器 (CSU_STAT)																															
偏移地址: 0x08																															
复位值: 0x1000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPT_TRIM								Reserved								CNTDIR	CNTCAP														

OPT_TRIM	Bit 31-23	R	Option Byte校准值 此寄存器为HRC 48MHz振荡器Option Byte校准值
—	Bit 22-17	—	—
CNTDIR	Bit 16	R	计数器方向状态位 此位表示频率偏差计数器的计数方向, 当同步信号触发事件发生时将自动被更新 0: 频率偏差计数器为递增计数, 表示振荡器的输出频率高于目标频率 1: 频率偏差计数器为递减计数, 表示振荡器的输出频率低于目标频率
CNTCAP	Bit 15-0	R	计数值 此寄存器记录频率偏差计数器的计数值, 当同步信号触发事件发生时将自动被更新, 计数值越接近零表示振荡器的输出频率越接近目标频率

10.5.2.4 CSU中断开启寄存器 (CSU_IER)

CSU 中断开启寄存器 (CSU_IER)																															
偏移地址: 0x10																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																									FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH	

—	Bit 31-6	—	—
FHIT	Bit 5	W1	理想同步中断使能位 0: 无效 1: 使能, 计数器的计数值等于零发生中断
TRIMERR	Bit 4	W1	校准值错误中断使能位 0: 无效 1: 使能, 自动校准模式时调整校准值达到上溢出或下溢出时发生中断
FFAULT	Bit 3	W1	时钟同步异常中断使能位 0: 无效 1: 使能, 未检测到同步信号触发事件, 且计数器的计数值大于等于极限值时发生中断
FERROR	Bit 2	W1	时钟同步错误中断使能位 0: 无效 1: 使能, 当同步信号触发事件时, 计数器的计数值大于等于极限值发生中断
FWARN	Bit 1	W1	时钟同步警告中断使能位 0: 无效 1: 使能, 当同步信号触发事件时, 计数器的计数值大于等于警告值且小于极限值发生中断
FMATCH	Bit 0	W1	时钟同步匹配中断使能位 0: 无效 1: 使能, 当同步信号触发事件时, 计数器的计数值小于警告值发生中断

注: 寄存器 CSU_IER 的各位写 1 使能中断, 写 0 无效。

10.5.2.5 CSU中断关闭寄存器 (CSU_IDR)

CSU 中断关闭寄存器 (CSU_IDR)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH		

—	Bit 31-6	—	—
FHIT	Bit 5	W1	理想同步中断禁止位 0: 无效 1: 禁止
TRIMERR	Bit 4	W1	校准值错误中断禁止位 0: 无效 1: 禁止
FFAULT	Bit 3	W1	时钟同步异常中断禁止位 0: 无效 1: 禁止
FERROR	Bit 2	W1	时钟同步错误中断禁止位 0: 无效 1: 禁止
FWARN	Bit 1	W1	时钟同步警告中断禁止位 0: 无效 1: 禁止
FMATCH	Bit 0	W1	时钟同步匹配中断禁止位 0: 无效 1: 禁止

注: 寄存器 CSU_IDR 的各位写 1 禁止中断, 写 0 无效。

10.5.2.6 CSU中断使能状态寄存器 (CSU_IVS)

CSU 中断使能状态寄存器 (CSU_IVS)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH		

—	Bit 31-6	—	—
FHIT	Bit 5	R	理想同步中断使能状态位 0: 关闭状态 1: 开启状态
TRIMERR	Bit 4	R	校准值错误中断使能状态位 0: 关闭状态 1: 开启状态
FFAULT	Bit 3	R	时钟同步异常中断使能状态位 0: 关闭状态 1: 开启状态
FERROR	Bit 2	R	时钟同步错误中断使能状态位 0: 关闭状态 1: 开启状态
FWARN	Bit 1	R	时钟同步警告中断使能状态位 0: 关闭状态 1: 开启状态
FMATCH	Bit 0	R	时钟同步匹配中断使能状态位 0: 关闭状态 1: 开启状态

10.5.2.7 CSU中断标志寄存器 (CSU_RIF)

CSU 中断标志寄存器 (CSU_RIF)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH		

—	Bit 31-6	—	—
FHIT	Bit 5	R	理想同步中断标志位 0: 无发生中断 1: 已发生中断
TRIMERR	Bit 4	R	校准值错误中断标志位 0: 无发生中断 1: 已发生中断
FFAULT	Bit 3	R	时钟同步异常中断标志位 0: 无发生中断 1: 已发生中断
FERROR	Bit 2	R	时钟同步错误中断标志位 0: 无发生中断 1: 已发生中断
FWARN	Bit 1	R	时钟同步警告中断标志位 0: 无发生中断 1: 已发生中断
FMATCH	Bit 0	R	时钟同步匹配中断标志位 0: 无发生中断 1: 已发生中断

10.5.2.8 CSU中断屏蔽后状态寄存器 (CSU_IFM)

CSU 中断屏蔽后状态寄存器 (CSU_IFM)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH

—	Bit 31-6	—	—
FHIT	Bit 5	R	理想同步, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断
TRIMERR	Bit 4	R	校准值错误, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断
FFAULT	Bit 3	R	时钟同步异常, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断
FERROR	Bit 2	R	时钟同步错误, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断
FWARN	Bit 1	R	时钟同步警告, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断
FMATCH	Bit 0	R	时钟同步匹配, 屏蔽后中断状态位 0: 无发生中断 1: 已发生中断

10.5.2.9 CSU中断清除寄存器 (CSU_ICR)

CSU 中断清除寄存器 (CSU_ICR)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								FHIT	TRIMERR	FFAULT	FERROR	FWARN	FMATCH		

—	Bit 31-6	—	—
FHIT	Bit 5	C_W1	理想同步中断标志/状态清除位 0: 无效 1: 清除 FHIT 与 FHITIFM
TRIMERR	Bit 4	C_W1	校准值错误中断标志/状态清除位 0: 无效 1: 清除 TRIMERR 与 TRIMERRIFM
FFAULT	Bit 3	C_W1	时钟同步异常中断标志/状态清除位 0: 无效 1: 清除 FFAULT 与 FFAULTIFM
FERROR	Bit 2	C_W1	时钟同步错误中断标志/状态清除位 0: 无效 1: 清除 FERROR 与 FERRORIFM
FWARN	Bit 1	C_W1	时钟同步警告中断标志/状态清除位 0: 无效 1: 清除 FWARN 与 FWARNIFM
FMATCH	Bit 0	C_W1	时钟同步匹配中断标志/状态清除位 0: 无效 1: 清除 FMATCH 与 FMATCHIFM

注: 寄存器 CSU_ICR 的各位写 1 清除中断标志和状态, 写 0 无效。

第11章 DMA控制器（DMA）

11.1 概述

直接存储器访问控制器（DMA）可独立于 CPU 对内部存储器进行操作，利用 DMA 可减轻 CPU 的负担并且节省功耗。DMA 控制器可以通过它的 7 个通道来实现在存储器和外设之间的数据传输（外设到存储器，存储器到外设，存储器到存储器，或者外设到外设，但不能搬移 Flash 中的数据）。

11.2 特性

- ◆ 单 AHB 主控制总线
- ◆ 支持 7 个 DMA 通道。每个通道支持单一方向的传输
- ◆ 支持来源地址和目的地地址的增加模式或固定模式
- ◆ 来源和目的地的传输大小（字节、半字、字）彼此独立，来源和目的地的地址必须根据传输数据的大小进行对齐
- ◆ 硬件通道优先级。DMA 信道 0 具有最高优先级，通道 6 具有最低优先级
- ◆ 软件通道优先级。设置软件信道优先级（每个通道有 4 个级别：最高、高、中、低），当两个 DMA 通道设置为相同的优先级时，它将根据硬件通道优先级被执行
- ◆ 支持循环模式
- ◆ 可编程的传输数量：0 到 $2^{16} - 1$
- ◆ 进行多个或单个 DMA 传输
- ◆ 进行以下不同类型的 DMA 传输
 - ◇ 存储器到存储器
 - ◇ 存储器到外设
 - ◇ 外设到存储器
 - ◇ 外设到外设

注 1: 存储器仅指 SRAM，DMA 不能搬移 Flash 上数据。

注 2: DMA 不同类型传输速率有差异，完成一次单位数据（字节、半字或字）传输所需的时间，存储器到存储器需约 6 个总线时钟（HCLK）周期，存储器到外设（或外设到存储器）需约 18 个总线时钟（HCLK）周期，外设到外设需约 24 个总线时钟（HCLK）周期。

注 3: 外设到外设的 DMA 传输速度较慢，应用时需注意若外设接收太快可能使得其接收 FIFO 溢出，所以外设接收速度不能快于内部 DMA 外设到外设的传输速度。

注 4: 在存储器到外设模式下，在 DMA 通道传输过程中，软件对任意 APB 外设的特殊功能寄存器进行读写操作的指令后，都需额外添加两个 NOP 空操作指令，否则可能会造成 DMA 传输数据错误。

注 5: 在使用 DMA 进行外设之间数据传输时，推荐使用外设到外设的 DMA 传输模式，无需使用存储器进行数据缓冲。

11.3 请求映射

DMA 控制器通过 DMAPMUX 外设连接至 AHB 或 APB 外设的 DMA 请求。

11.4 结构图

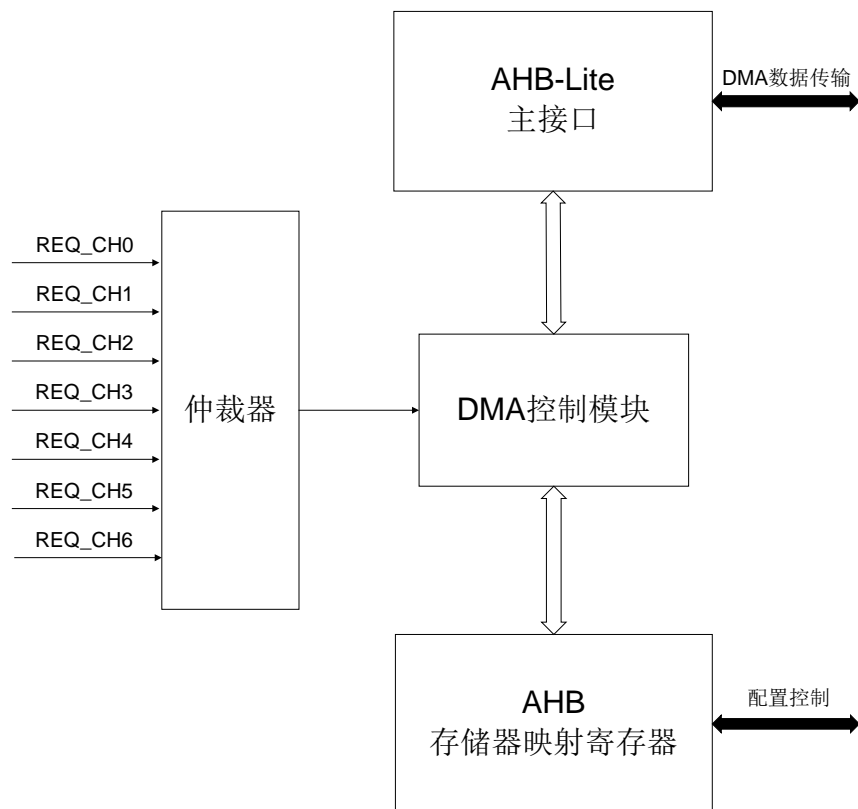


图 11-1 直接存储器访问控制器框图

DMA 控制器通过与其他系统主设备共享 AHB 系统总线来执行直接存储器传输。总线矩阵执行循环调度。当 CPU 与 DMA 访问相同的存储器或外设地址时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。

通过软件配置，DMA 控制器会在 DMA 通道及其接收的相关请求之间进行仲裁。DMA 控制器还会通过单一 AHB 主控制总线调度 DMA 数据传输。

11.5 功能描述

11.5.1 传输事务

配置 DMA 控制器中的通道级别，以便执行数据传输，此类传输由一系列 AHB 总线传输组成。

可以透过外设请求或在存储器到存储器 (M2M=1) 模式时设置 DMA_CONn 寄存器的 CHEN 位为 1 触发数据传输。

当触发事件发生后，会按照以下步骤进行传输：

- ◆ 外设向 DMA 控制器发送 DMA 请求讯号。

- ◆ DMA 控制器按照与此外设请求相关的通道的优先级来处理该请求。
- ◆ 只要 DMA 控制器授权给外设，当传输到最后一次数据时，DMA 控制器就会向外设发送确认讯号。
- ◆ 外设获得 DMA 控制器的确认讯号后，便会立即释放其请求讯号。
- ◆ 一旦外设释放请求讯号后 DMA 控制器就会释放确认讯号。

外设可继续发送请求，再次启动 DMA 传输。

不论外设是传输来源或是传输目的地，都使用请求与应答协议。例如外设到存储器传输，外设向 DMA 控制器发送请求讯号来启动传输。DMA 控制器随后读取外设单个数据，并将该数据写入存储器。

对于给定信道 n，DMA 数据传输由以下的重复序列组成：

- ◆ 单次 DMA 传输 (DMA_CONn.MAX_BURST 为 0)，需要两次总线访问 (读取数据与写入数据) 来执行传输 (通过 DMA AHB 主控制总线实现)：
 - ◇ DMA 控制器获得请求讯号。
 - ◇ 通过 DMA_SARn 寄存器从外设数据寄存器或存储器单元中读取单个数据 (字节、半字或字)。
 - ◇ 通过 DMA_DARn 寄存器向外设数据寄存器或存储器单元中写入单个数据 (字节、半字或字)。
 - ◇ DMA_NDTn 寄存器中 NRDT 位域在传输后递减，该寄存器包含剩余数据传输数量 (“先读后写”的传输次数)。
 - ◇ DMA 控制器发送确认讯号。
 - ◇ 重复该序列，直到 DMA_NDTn 寄存器中 NRDT 位域等于 0。
- ◆ 多次 DMA 传输 (DMA_CONn.MAX_BURST 不为 0)，需要多组两次总线访问 (读取数据与写入数据) 来执行传输 (通过 DMA AHB 主控制总线实现)：
 - ◇ DMA 控制器获得请求讯号。
 - ◇ 通过 DMA_SARn 寄存器从外设数据寄存器或存储器单元中读取单个数据 (字节、半字或字)。
 - ◇ 通过 DMA_DARn 寄存器向外设数据寄存器或存储器单元中写入单个数据 (字节、半字或字)。
 - ◇ DMA_NDTn 寄存器中 NRDT 位域在传输后递减，该寄存器包含剩余数据传输数量 (“先读后写”的传输次数)。
 - ◇ 当传输次数等于 DMA_CONn.MAX_BURST 设定值。
 - ◇ DMA 控制器发送确认讯号。
 - ◇ 重复该序列，直到 DMA_NDTn 寄存器中 NRDT 位域等于 0。

11.5.2 仲裁率

在 DMA 传输过程中，用户可配置控制器何时进行仲裁。该动作可缩短响应高优先通道的延时。

控制器提供 4 个比特位，用来配置在重新进行仲裁前 AHB 总线传输的次数。这些位称为 MAX_BURST 位，输入的数值会提高到 2 的次方 ($2^{\text{MAX_BURST}}$)，进而决定了仲裁率。当 MAX_BURST = 4，则仲裁率为 $2^4 = 16$ ，即每进行 16 次 DMA 先读后写传输就进行一次仲裁。

MAX_BURST	x 次 DMA 传输后进行仲裁
b0000	x = 1
b0001	x = 2
b0010	x = 4
b0011	x = 8
b0100	x = 16
b0101	x = 32
b0110	x = 64
b0111	x = 128
b1000	x = 256
b1001	x = 512
b1010-b1111	x = 1024

注：优先级低的信道中 MAX_BURST 值不要配置太大，这样会导致在重新进行仲裁前，控制器不会响应优先级高的请求。

DMA 通道的传输数量 DMA_NDTn.TNDT 是由用户指定的。当 $\text{DMA_NDTn.NRDT} > 2^{\text{MAX_BURST}}$ 且 DMA_NDTn.NRDT 不是 $2^{\text{MAX_BURST}}$ 的整数倍，控制器会依次先完成 $2^{\text{MAX_BURST}}$ 次传输直到剩余的次数小于 $2^{\text{MAX_BURST}}$ 。

当剩余的次数小于 $2^{\text{MAX_BURST}}$ 时数据传输方式有两种：

- ◆ 在存储器到存储器模式或存储器到外设模式时，剩余的次数会在下一次请求时完成
- ◆ 在外设到存储器模式时，每次外设请求只会执行一次传输，直到传输完成

11.5.3 优先级

DMA 仲裁器管理不同信道间的优先级。

当 DMA 仲裁器授予某个有效通道 n 优先权后，根据 DMA_CONn.MAX_BURST 设定值会发起单次或多次 DMA 先读后写传输。随后仲裁器会再次对有效通道进行仲裁，并选择优先级最高的通道。

优先级分两个阶段进行管理：

- ◆ 软件：可以在 DMA_CONn 寄存器中的 CHPRI 位域配置该信道优先级，分为四个级别：

- ◇ 最高优先级
- ◇ 高优先级
- ◇ 中优先级
- ◇ 低优先级
- ◆ 硬件：如果两个通道请求具有相同的软件优先级，则通道编号低优先于通道编号高。例如信道 2 的优先级高于通道 4。

11.5.4 传输模式

11.5.4.1 存储器到存储器模式 (Memory-to-memory)

DMA 通道在没有外设请求触发的情况下同样可以工作。该模式被称为存储器到存储器模式，由 DMA_CONn 寄存器的 M2M 位设置 1，且 CHEN 位也设置 1 时启动传输。在 DMA_NDTn 寄存器的 NRDT 位域等于 0 后传输停止。

存储器到存储器模式的示例：

配置 DMA 信道 n 为存储器到存储器模式，将数据从 SRAM 地址 0x2000_0000 搬移到 0x2000_1000。

配置流程如下：

- ◆ 设置来源地址：DMA_SARn = 0x2000_0000
- ◆ 设置目的地地址：DMA_DARn = 0x2000_1000
- ◆ 设置总数据传输数量：DMA_NDTn.TNDT = 0x400
- ◆ 使能存储器到存储器模式：DMA_CONn.M2M = 1
- ◆ 使能通道：DMA_CONn.CHEN = 1

注：在循环模式下，不得使用存储器到存储器模式。在存储器到存储器模式 (DMA_CONn.M2M = 1) 模式下，使能通道 (DMA_CONn.CHEN = 1) 前，软件必须将 DMA_CONn 寄存器的 CIRC 位清零。

11.5.4.2 存储器到外设模式 (Memory-to-peripheral)

DMA 通道在 DMA_CONn 寄存器的 DIR 位设置 1 且 M2M 为 0 时，DMA 控制器工作在存储器到外设模式下。

注：在存储器到外设模式下，在 DMA 通道传输过程中，软件对任意 APB 外设的特殊功能寄存器进行读写操作的指令后，都需额外添加两个 NOP 空操作指令，否则可能会造成 DMA 传输数据错误。

存储器到外设模式的示例：

配置 DMA 信道 n 为存储器到外设模式，将数据从 SRAM 地址 0x2000_0000 搬移到 UART 外设的 UART_TXBUF，此外设必须开启发送器 DMA (UART_MCON.TXDMAEN = 1)。

配置流程如下：

- ◆ 设置来源地址：DMA_SARn = 0x2000_0000

- ◆ 设置目的地地址：DMA_DARn = UART_TXBUF
- ◆ 设置总数据传输数量：DMA_NDTn.TNDT = 0x400
- ◆ 禁止存储器到存储器模式：DMA_CONn.M2M = 0
- ◆ 使能存储器到外设模式：DMA_CONn.DIR = 1
- ◆ 使能通道：DMA_CONn.CHEN = 1

11.5.4.3 外设到存储器模式 (Peripheral-to-memory)

DMA 通道在 DMA_CONn 寄存器的 DIR 位设置 0 且 M2M 为 0 时，DMA 控制器工作在外设到存储器模式下。

外设到存储器模式的示例：

配置 DMA 信道 n 为外设到存储器模式，将数据从 UART 外设的 UART_RXBUF 搬移到 SRAM 地址 0x2000_0000，此外设必须开启接收器 DMA (UART_MCON.RXDMAEN = 1)。配置流程如下：

- ◆ 设置来源地址：DMA_SARn = UART_RXBUF
- ◆ 设置目的地地址：DMA_DARn = 0x2000_0000
- ◆ 设置总数据传输数量：DMA_NDTn.TNDT = 0x400
- ◆ 禁止存储器到存储器模式：DMA_CONn.M2M = 0
- ◆ 使能外设到存储器模式：DMA_CONn.DIR = 0
- ◆ 使能通道：DMA_CONn.CHEN = 1

11.5.4.4 外设到外设模式 (Peripheral-to-peripheral)

DMA 控制器可透过软件配置外设的方式与限制，实现外设到外设模式。

外设到外设模式的示例：

配置 DMA 信道 n 为外设到存储器模式，将数据从 UART 外设的 UART_RXBUF 搬移到 UART 外设的 UART_TXBUF，此外设仅可开启接收器 DMA (UART_MCON.RXDMAEN = 1)，并关闭发送器 DMA (UART_MCON.TXDMAEN = 0)。

配置流程如下：

- ◆ 设置来源地址：DMA_SARn = UART_RXBUF
- ◆ 设置目的地地址：DMA_DARn = UART_TXBUF
- ◆ 设置总数据传输数量：DMA_NDTn.TNDT = 0x400
- ◆ 禁止存储器到存储器模式：DMA_CONn.M2M = 0
- ◆ 使能外设到存储器模式：DMA_CONn.DIR = 0
- ◆ 使能通道：DMA_CONn.CHEN = 1

注：

- ◆ 在外设到外设模式下，DMA_CONn 寄存器的 MAX_BURST 位域必须设置为 0。
- ◆ 因为 DMA 每个信道仅有一个请求讯号，所以在此模式时仅能启动每个外设的一个 DMA 功能。

- ◆ 外设到外设的 DMA 传输较慢，应用时注意若外设接收太快可能使得其接收 FIFO 溢出，所以外设接收速度不能快于内部 DMA 外设到外设的传输速度。

11.5.5 地址递增

根据 DMA_CSRn 寄存器中 SINC 和 DINC 的设置，在每次传输后，来源和目的地地址可以选择自动递增或保持不变。

如果使能了递增模式（SINC 或 DINC 置 1），则下次传输的地址是前一次传输的地址加上 1（对应于字节传输）、2（对应于半字传输）或 4（对应于字传输），具体取决于在 DMA_CONn 寄存器中的 SDWSEL[1:0] 或 DDWSEL[1:0] 定义的数据宽度。首次传输的地址可在 DMA_SARx 或 DMA_DARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址。

如果将信道配置为非循环模式，则在最后一次数据传输完成后，将不处理任何 DMA 请求，并将 DMA_CONn 寄存器的 CHEN 清零。

注：如果禁止通道 n，DMA 寄存器不会被复位。DMA_CONn、DMA_SARn 和 DMA_DARn 寄存器仍保留在信道配置阶段设置的初始值。

在循环模式下，最后一次数据传输完成后，当前的内部地址寄存器重新加载 DMA_SARn 和 DMA_DARn 寄存器中的基址值，DMA_NDTn 寄存器的 NRDT 位域重新加载 TNDT 数据数量。

11.5.6 循环模式（仅用于存储器和外设之间传输）

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_CONn 寄存器中的 CIRC 使能该功能。

在存储器到存储器模式下，不能使用循环模式。在循环模式（CIRC = 1）下使能信道前，软件必须将 DMA_CONn 寄存器的 M2M 位清零。

当使能循环模式时，在最后一次数据传输完成后，DMA_NDTn 寄存器的 NRDT 位域将重新加载 TNDT 数据数量，并继续响应 DMA 请求。

可通过两种方式退出循环模式：

- ◆ 当发生块传输完成中断事件后，通过软件将 DMA_CONn 寄存器的 CIRC 位清零，此时 DMA 控制器会持续响应外设请求，直到再次发生块传输完成中断事件后，硬件会将 DMA_CONn 寄存器中的 CHEN 位清零且停止响应外设请求。
- ◆ 软件需要先将外设停止生成 DMA 请求，然后通过软件将 DMA_CONn 寄存器中的 CHEN 位与 CIRC 位清零，停止响应外设的请求并退出循环模式。

11.5.7 数据宽度、对齐和字节序

当 DMA_CONn 寄存器的 SDWSEL[1:0] 和 DDWSEL[1:0] 不相等时，则 DMA 控制器将按照下表所述方式进行数据对齐。

来源端口宽度 (SDWSEL)	目的地端口宽度 (DDWSEL)	传输数量 (TNDT)	来源地址/数据	目的地地址/数据
8	8	4	0x0 / 0xB0	0x0 / 0xB0

			0x1 / 0xB1	0x1 / 0xB1
			0x2 / 0xB2	0x2 / 0xB2
			0x3 / 0xB3	0x3 / 0xB3
8	16	4	0x0 / 0xB0	0x0 / 0x00B0
			0x1 / 0xB1	0x2 / 0x00B1
			0x2 / 0xB2	0x4 / 0x00B2
			0x3 / 0xB3	0x6 / 0x00B3
8	32	4	0x0 / 0xB0	0x0 / 0x000000B0
			0x1 / 0xB1	0x4 / 0x000000B1
			0x2 / 0xB2	0x8 / 0x000000B2
			0x3 / 0xB3	0xC / 0x000000B3
16	8	4	0x0 / 0xB1B0	0x0 / 0xB0
			0x2 / 0xB3B2	0x1 / 0xB2
			0x4 / 0xB5B4	0x2 / 0xB4
			0x6 / 0xB7B6	0x3 / 0xB6
16	16	4	0x0 / 0xB1B0	0x0 / 0xB1B0
			0x2 / 0xB3B2	0x2 / 0xB3B2
			0x4 / 0xB5B4	0x4 / 0xB5B4
			0x6 / 0xB7B6	0x6 / 0xB7B6
16	32	4	0x0 / 0xB1B0	0x0 / 0x0000B1B0
			0x2 / 0xB3B2	0x4 / 0x0000B3B2
			0x4 / 0xB5B4	0x8 / 0x0000B5B4
			0x6 / 0xB7B6	0xC / 0x0000B7B6
32	8	4	0x0 / 0xB3B2B1B0	0x0 / 0xB0
			0x4 / 0xB7B6B5B4	0x1 / 0xB4
			0x8 / 0xBBBAB9B8	0x2 / 0xB8
			0xC / 0xBFBEBC	0x3 / 0xBC
32	16	4	0x0 / 0xB3B2B1B0	0x0 / 0xB1B0
			0x4 / 0xB7B6B5B4	0x2 / 0xB5B4
			0x8 / 0xBBBAB9B8	0x4 / 0xB9B8
			0xC / 0xBFBEBC	0x6 / 0xBDBC
32	32	4	0x0 / 0xB3B2B1B0	0x0 / 0xB3B2B1B0
			0x4 / 0xB7B6B5B4	0x4 / 0xB7B6B5B4

			0x8 / 0xBBBAB9B8	0x8 / 0xBBBAB9B8
			0xC / 0xBFBEBC	0xC / 0xBFBEBC

11.5.7.1 解决 AHB 外设不支持字节或半字写传输的问题

如果 DMA 控制器启动 AHB 字节或半字写传输，则 32 位 AHB 主控数据总线 (HWDATA<31:0>) 的未使用数据在线将会重复所传输的数据。

如果 AHB 从外设不支持字节或半字写传输且不会产生任何错误，则 DMA 控制器会对 HWDATA 的 32 个位执行写操作，如下列两个范例所示：

- ◆ 要写入半字数据为 0xABCD，DMA 控制器会将 HWDATA 总线设为 0xABCDABCD，并采用半字数据大小（在 AHB 主控总线中，将 HSIZE 设为 HalfWord）。
- ◆ 要写入字节数据为 0xAB，DMA 控制器会将 HWDATA 总线设为 0xABABABAB，并采用字节数据大小（在 AHB 主控总线中，将 HSIZE 设为 Byte）。

当目的地地址为 APB 外设，且该外设不考虑 HSIZE 数据（寄存器为字切齐），则任何 AHB 字节或半字传输都将转换为 32 位 APB 传输，如下所述：

- ◆ 将 AHB 字节写传输转换为 APB 字写传输，如向 0x0、0x1、0x2 或 0x3 地址之一写入数据 0xB0，将转换为向 0x0 地址写入 0xB0B0B0B0。
- ◆ 将 AHB 半字写传输转换为 APB 字写传输，如向 0x0 或 0x2 地址写入数据 0xB1B0，将转换为向 0x0 地址写入 0xB1B0B1B0。

11.5.8 信道配置流程

配置 DMA 信道 n 时需按照以下步骤操作：

- ◆ 在 DMA_IER 寄存器中配置下列参数：
 - ◇ 传输完成一半或全部完成时的中断使能
- ◆ 将来源地址设置到 DMA_SARn 寄存器。
使能通道后，在存储器到存储器模式或外设请求发生后，DMA 控制器会通过该地址读取数据。
- ◆ 将目的地地址设置到 DMA_DARn 寄存器。
使能通道后，在存储器到存储器模式或外设请求发生后，DMA 控制器会通过该地址写入数据。
- ◆ 在 DMA_NDTn 寄存器的 TNDT 位域配置传输的总数据数。
当通道使能（软件对 DMA_CONn.CHEN 设置为 1）时，NRDT 会自动加载 TNDT 数值，并每次数据传输后，NRDT 位域都会递减。
- ◆ 在 DMA_CONn 寄存器中配置下列参数：
 - ◇ 通道优先级
 - ◇ 数据传输方向
 - ◇ 循环模式
 - ◇ 外设和存储器递增模式
 - ◇ 外设和存储器数据大小

- ◇ 仲裁率

- ◆ 将 DMA_CONn 寄存器中的 CHEN 位置 1 以使能通道。

信道在使能后可处理来自此通道所连接外设的任意 DMA 请求，或者启动存储器到存储器数据传输。

注：信道配置流程的最后两步可合并为对 DMA_CONn 寄存器进行单次访问来配置和使能信道。

11.5.9 传输完成

- ◆ 信道被配置为非循环模式

在 DMA_NDTn 寄存器的 NRDT 位域递减到零则传输结束，通道被禁止（DMA_CONn 寄存器中的 CHEN 位被硬件清零）并将 DMA_RIF.CHnTC 设置为 1，此时 DMA 控制器不会响应外设请求，除非软件重新编程并重新启用它（通过设置 DMA_CONn 寄存器中的 CHEN 位为 1）。

- ◆ 信道被配置为循环模式

在 DMA_NDTn 寄存器的 NRDT 位域递减到零时，NRDT 位域将重新加载 TNDT 数据数量并将 DMA_RIF.CHnTC 设置为 1，此时 DMA 控制器会继续响应外设的请求。

11.5.10 传输暂停

可以随时暂停 DMA 传输以供稍后重新开始。

分为两种情况：

- ◆ 禁止传输，暂停以后不从停止点重新开始。这种情况下只需将 DMA_CONn 寄存器中的 CHEN 位清零，除此之外不需要任何其他操作。禁止传输可能要花费一些时间（需要首先完成正在进行的传输）。等待 DMA_CONn 寄存器中的 CHEN 位的值为 0，以此确认已经终止传输。DMA_NDTn 寄存器的 NRDT 位域包含数据停止时剩余数据项的数量，这样软件便可以确定数据传输中断前已传输了多少数据项。

- ◆ 暂停并恢复传输，在 DMA_NDTn 寄存器中的 TNDT 位域递减到 0 之前暂停传输。目的是以后通过重新使能通道并重新开始传输。为了在传输停止点重新开始传输，软件必须在通过写入 DMA_CONn 寄存器中的 CHEN 位（需检查确认该位为 0）禁止通道之后，首先读取 DMA_NDTn 寄存器中的 NRDT 位域来了解已经传输的数据项的数量。然后：

- ◇ 必须更新来源或目的地地址以调整地址指针

- ◇ 必须使用要传输的剩余数据项的数目，在暂停时读取 DMA_CONn 寄存器的 NRDT 位域数值，并更新到 DMA_CONn 寄存器的 TNDT 位域

- ◇ 重新使能通道（设置 DMA_CONn 寄存器的 CHEN 为 1），以从停止点重新开始传输

11.5.11 中断事件

对于每个 DMA 通道在发生传输一半或传输完成时都会触发中断事件，产生中断标志。

11.6 特殊功能寄存器

11.6.1 寄存器列表

DMA 寄存器列表			
基地址: 0x4008_5400			
名称	偏移地址	类型	描述
DMA_IER	0000 _H	W1	DMA 中断使能寄存器
DMA_IDR	0004 _H	W1	DMA 中断禁止寄存器
DMA_IVS	0008 _H	R	DMA 中断使能状态寄存器
DMA_RIF	000C _H	R	DMA 中断标志寄存器
DMA_IFM	0010 _H	R	DMA 中断屏蔽后状态寄存器
DMA_ICR	0014 _H	C_W1	DMA 中断清除寄存器
DMA_CON0	0020 _H	R/W	DMA 通道 0 控制寄存器
DMA_SAR0	0024 _H	R/W	DMA 信道 0 来源地址寄存器
DMA_DAR0	0028 _H	R/W	DMA 信道 0 目的地地址寄存器
DMA_NDT0	002C _H	R/W	DMA 信道 0 数据传输数量寄存器
DMA_CON1	0030 _H	R/W	DMA 通道 1 控制寄存器
DMA_SAR1	0034 _H	R/W	DMA 信道 1 来源地址寄存器
DMA_DAR1	0038 _H	R/W	DMA 信道 1 目的地地址寄存器
DMA_NDT1	003C _H	R/W	DMA 信道 1 数据传输数量寄存器
DMA_CON2	0040 _H	R/W	DMA 通道 2 控制寄存器
DMA_SAR2	0044 _H	R/W	DMA 信道 2 来源地址寄存器
DMA_DAR2	0048 _H	R/W	DMA 信道 2 目的地地址寄存器
DMA_NDT2	004C _H	R/W	DMA 信道 2 数据传输数量寄存器
DMA_CON3	0050 _H	R/W	DMA 通道 3 控制寄存器
DMA_SAR3	0054 _H	R/W	DMA 信道 3 来源地址寄存器
DMA_DAR3	0058 _H	R/W	DMA 信道 3 目的地地址寄存器
DMA_NDT3	005C _H	R/W	DMA 信道 3 数据传输数量寄存器
DMA_CON4	0060 _H	R/W	DMA 通道 4 控制寄存器
DMA_SAR4	0064 _H	R/W	DMA 信道 4 来源地址寄存器
DMA_DAR4	0068 _H	R/W	DMA 信道 4 目的地地址寄存器
DMA_NDT4	006C _H	R/W	DMA 信道 4 数据传输数量寄存器
DMA_CON5	0070 _H	R/W	DMA 通道 5 控制寄存器
DMA_SAR5	0074 _H	R/W	DMA 信道 5 来源地址寄存器
DMA_DAR5	0078 _H	R/W	DMA 信道 5 目的地地址寄存器
DMA_NDT5	007C _H	R/W	DMA 信道 5 数据传输数量寄存器
DMA_CON6	0080 _H	R/W	DMA 通道 6 控制寄存器
DMA_SAR6	0084 _H	R/W	DMA 信道 6 来源地址寄存器
DMA_DAR6	0088 _H	R/W	DMA 信道 6 目的地地址寄存器
DMA_NDT6	008C _H	R/W	DMA 信道 6 数据传输数量寄存器

DMA_MUX 寄存器列表			
基地址: 0x4000_D000			
名称	偏移地址	类型	描述
DMA_CH0_SELCON	0000 _H	R/W	DMA 通道 0 复用选择寄存器
DMA_CH1_SELCON	0004 _H	R/W	DMA 通道 1 复用选择寄存器
DMA_CH2_SELCON	0008 _H	R/W	DMA 通道 2 复用选择寄存器
DMA_CH3_SELCON	000C _H	R/W	DMA 通道 3 复用选择寄存器
DMA_CH4_SELCON	0010 _H	R/W	DMA 通道 4 复用选择寄存器
DMA_CH5_SELCON	0014 _H	R/W	DMA 通道 5 复用选择寄存器
DMA_CH6_SELCON	0018 _H	R/W	DMA 通道 6 复用选择寄存器

11.6.2 寄存器描述

11.6.2.1 DMA 中断使能寄存器 (DMA_IER)

DMA 中断使能寄存器 (DMA_IER)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC

—	Bits 31-14	—	—
CH6HT	Bit 13	W1	DMA 通道 6 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 6 传输一半中断
CH6TC	Bit 12	W1	DMA 通道 6 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 6 传输完成中断
CH5HT	Bit 11	W1	DMA 通道 5 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 5 传输一半中断
CH5TC	Bit 10	W1	DMA 通道 5 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 5 传输完成中断
CH4HT	Bit 9	W1	DMA 通道 4 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 4 传输一半中断
CH4TC	Bit 8	W1	DMA 通道 4 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 4 传输完成中断
CH3HT	Bit 7	W1	DMA 通道 3 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 3 传输一半中断
CH3TC	Bit 6	W1	DMA 通道 3 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 3 传输完成中断
CH2HT	Bit 5	W1	DMA 通道 2 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 2 传输一半中断
CH2TC	Bit 4	W1	DMA 通道 2 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 2 传输完成中断

CH1HT	Bit 3	W1	DMA 通道 1 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 1 传输一半中断
CH1TC	Bit 2	W1	DMA 通道 1 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 1 传输完成中断
CH0HT	Bit 1	W1	DMA 通道 0 传输一半中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 0 传输一半中断
CH0TC	Bit 0	W1	DMA 通道 0 传输完成中断使能位 0: 写入 0 无效 1: 使能 DMA 通道 0 传输完成中断

11.6.2.2 DMA 中断禁止寄存器 (DMA_IDR)

DMA 中断禁止寄存器 (DMA_IDR)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC				

—	Bits 31-14	—	—
CH6HT	Bit 13	W1	DMA 通道 6 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 6 传输一半中断
CH6TC	Bit 12	W1	DMA 通道 6 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 6 传输完成中断
CH5HT	Bit 11	W1	DMA 通道 5 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 5 传输一半中断
CH5TC	Bit 10	W1	DMA 通道 5 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 5 传输完成中断
CH4HT	Bit 9	W1	DMA 通道 4 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 4 传输一半中断
CH4TC	Bit 8	W1	DMA 通道 4 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 4 传输完成中断
CH3HT	Bit 7	W1	DMA 通道 3 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 3 传输一半中断
CH3TC	Bit 6	W1	DMA 通道 3 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 3 传输完成中断
CH2HT	Bit 5	W1	DMA 通道 2 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 2 传输一半中断
CH2TC	Bit 4	W1	DMA 通道 2 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 2 传输完成中断
CH1HT	Bit 3	W1	DMA 通道 1 传输一半中断禁止位

			0: 写入 0 无效 1: 禁止 DMA 通道 1 传输一半中断
CH1TC	Bit 2	W1	DMA 通道 1 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 1 传输完成中断
CH0HT	Bit 1	W1	DMA 通道 0 传输一半中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 0 传输一半中断
CH0TC	Bit 0	W1	DMA 通道 0 传输完成中断禁止位 0: 写入 0 无效 1: 禁止 DMA 通道 0 传输完成中断

11.6.2.3 DMA 中断使能状态寄存器 (DMA_IVS)

DMA 中断使能状态寄存器 (DMA_IVS)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC

—	Bits 31-14	—	—
CH6HT	Bit 13	R	DMA 通道 6 传输一半中断使能状态位 0: 禁止 DMA 通道 6 传输一半中断 1: 使能 DMA 通道 6 传输一半中断
CH6TC	Bit 12	R	DMA 通道 6 传输完成中断使能状态位 0: 禁止 DMA 通道 6 传输完成中断 1: 使能 DMA 通道 6 传输完成中断
CH5HT	Bit 11	R	DMA 通道 5 传输一半中断使能状态位 0: 禁止 DMA 通道 5 传输一半中断 1: 使能 DMA 通道 5 传输一半中断
CH5TC	Bit 10	R	DMA 通道 5 传输完成中断使能状态位 0: 禁止 DMA 通道 5 传输完成中断 1: 使能 DMA 通道 5 传输完成中断
CH4HT	Bit 9	R	DMA 通道 4 传输一半中断使能状态位 0: 禁止 DMA 通道 4 传输一半中断 1: 使能 DMA 通道 4 传输一半中断
CH4TC	Bit 8	R	DMA 通道 4 传输完成中断使能状态位 0: 禁止 DMA 通道 4 传输完成中断 1: 使能 DMA 通道 4 传输完成中断
CH3HT	Bit 7	R	DMA 通道 3 传输一半中断使能状态位 0: 禁止 DMA 通道 3 传输一半中断 1: 使能 DMA 通道 3 传输一半中断
CH3TC	Bit 6	R	DMA 通道 3 传输完成中断使能状态位 0: 禁止 DMA 通道 3 传输完成中断 1: 使能 DMA 通道 3 传输完成中断
CH2HT	Bit 5	R	DMA 通道 2 传输一半中断使能状态位 0: 禁止 DMA 通道 2 传输一半中断 1: 使能 DMA 通道 2 传输一半中断
CH2TC	Bit 4	R	DMA 通道 2 传输完成中断使能状态位 0: 禁止 DMA 通道 2 传输完成中断 1: 使能 DMA 通道 2 传输完成中断
CH1HT	Bit 3	R	DMA 通道 1 传输一半中断使能状态位

			0: 禁止 DMA 通道 1 传输一半中断 1: 使能 DMA 通道 1 传输一半中断
CH1TC	Bit 2	R	DMA 通道 1 传输完成中断使能状态位 0: 禁止 DMA 通道 1 传输完成中断 1: 使能 DMA 通道 1 传输完成中断
CH0HT	Bit 1	R	DMA 通道 0 传输一半中断使能状态位 0: 禁止 DMA 通道 0 传输一半中断 1: 使能 DMA 通道 0 传输一半中断
CH0TC	Bit 0	R	DMA 通道 0 传输完成中断使能状态位 0: 禁止 DMA 通道 0 传输完成中断 1: 使能 DMA 通道 0 传输完成中断

注：DMA_IVS 寄存器实时反映系统配置 DMA_IER 与 DMA_IDR 的中断开启状态。

11.6.2.4 DMA 中断标志寄存器 (DMA_RIF)

DMA 中断标志寄存器 (DMA_RIF)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC				

—	Bits 31-14	—	—
CH6HT	Bit 13	R	DMA 通道 6 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH6TC	Bit 12	R	DMA 信道 6 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH5HT	Bit 11	R	DMA 通道 5 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH5TC	Bit 10	R	DMA 信道 5 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH4HT	Bit 9	R	DMA 通道 4 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH4TC	Bit 8	R	DMA 信道 4 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH3HT	Bit 7	R	DMA 通道 3 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH3TC	Bit 6	R	DMA 信道 3 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH2HT	Bit 5	R	DMA 通道 2 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH2TC	Bit 4	R	DMA 信道 2 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH1HT	Bit 3	R	DMA 通道 1 传输一半中断事件标志位

			0: 未发生中断事件 1: 发生中断事件
CH1TC	Bit 2	R	DMA 信道 1 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH0HT	Bit 1	R	DMA 通道 0 传输一半中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CH0TC	Bit 0	R	DMA 信道 0 传输完成中断事件标志位 0: 未发生中断事件 1: 发生中断事件

11.6.2.5 DMA 中断屏蔽后状态寄存器 (DMA_IFM)

DMA 中断屏蔽后状态寄存器 (DMA_IFM)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC

—	Bits 31-14	—	—
CH6HT	Bit 13	R	DMA 通道 6 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH6TC	Bit 12	R	DMA 通道 6 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH5HT	Bit 11	R	DMA 通道 5 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH5TC	Bit 10	R	DMA 通道 5 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH4HT	Bit 9	R	DMA 通道 4 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH4TC	Bit 8	R	DMA 通道 4 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH3HT	Bit 7	R	DMA 通道 3 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH3TC	Bit 6	R	DMA 通道 3 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH2HT	Bit 5	R	DMA 通道 2 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH2TC	Bit 4	R	DMA 通道 2 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH1HT	Bit 3	R	DMA 通道 1 传输一半中断屏蔽后状态位

			0: 未发生中断事件或中断未使能 1: 产生中断
CH1TC	Bit 2	R	DMA 通道 1 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH0HT	Bit 1	R	DMA 通道 0 传输一半中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断
CH0TC	Bit 0	R	DMA 通道 0 传输完成中断屏蔽后状态位 0: 未发生中断事件或中断未使能 1: 产生中断

注：DMA_IFM 寄存器是滤除已关闭中断功能的中断事件，只关注开启中断功能的事件，此寄存器的状态位是将 DMA_RIF 与 DMA_IVS 进行逻辑与运算后的结果。

11.6.2.6 DMA 中断清除寄存器 (DMA_ICR)

DMA 中断清除寄存器 (DMA_ICR)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CH6HT	CH6TC	CH5HT	CH5TC	CH4HT	CH4TC	CH3HT	CH3TC	CH2HT	CH2TC	CH1HT	CH1TC	CH0HT	CH0TC					

—	Bits 31-14	—	—
CH6HT	Bit 13	C_W1	DMA 通道 6 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH6TC	Bit 12	C_W1	DMA 通道 6 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH5HT	Bit 11	C_W1	DMA 通道 5 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH5TC	Bit 10	C_W1	DMA 通道 5 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH4HT	Bit 9	C_W1	DMA 通道 4 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH4TC	Bit 8	C_W1	DMA 通道 4 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH3HT	Bit 7	C_W1	DMA 通道 3 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH3TC	Bit 6	C_W1	DMA 通道 3 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH2HT	Bit 5	C_W1	DMA 通道 2 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH2TC	Bit 4	C_W1	DMA 通道 2 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH1HT	Bit 3	C_W1	DMA 通道 1 传输一半中断标志/状态清除位

			0: 写入 0 无效 1: 清除中断标志/状态
CH1TC	Bit 2	C_W1	DMA 通道 1 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH0HT	Bit 1	C_W1	DMA 通道 0 传输一半中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态
CH0TC	Bit 0	C_W1	DMA 通道 0 传输完成中断标志/状态清除位 0: 写入 0 无效 1: 清除中断标志/状态断

注: DMA_ICR 寄存器设置时, 将清除寄存器 DMA_RIF 与 DMA_IFM 的中断标志状态, 不影响寄存器 DMA_IER、DMA_IDR 与 DMA_IVS。

11.6.2.7 DMA 通道 0 控制寄存器 (DMA_CON0)

DMA 通道 0 控制寄存器 (DMA_CON0)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁</p> <p>b0001: 发生 2 次DMA传输后仲裁</p> <p>b0010: 发生 4 次DMA传输后仲裁</p> <p>b0011: 发生 8 次DMA传输后仲裁</p> <p>b0100: 发生 16 次DMA传输后仲裁</p> <p>b0101: 发生 32 次DMA传输后仲裁</p> <p>b0110: 发生 64 次DMA传输后仲裁</p> <p>b0111: 发生 128 次DMA传输后仲裁</p> <p>b1000: 发生 256 次DMA传输后仲裁</p> <p>b1001: 发生 512 次DMA传输后仲裁</p> <p>b1010-b1111: 发生 1024 次DMA传输后仲裁。</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式</p> <p>1: 使能目的地地址增量模式</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位 0: 禁止来源地址增量模式 1: 使能来源地址增量模式 只能在 CHEN = 0 时对此位执行写操作。</p>
<i>Reserved</i>	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位 b00: 低 b01: 中 b10: 高 b11: 最高 只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位 0: 禁止 1: 使能 只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位 0: 外设到存储器 1: 存储器到外设 只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位 0: 禁止 1: 使能 在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位 0: 禁止 1: 使能</p>

11.6.2.8 DMA 通道 0 源地址寄存器 (DMA_SAR0)

DMA 通道 0 源地址寄存器 (DMA_SAR0)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.9 DMA 通道 0 目的地址寄存器 (DMA_DAR0)

DMA 通道 0 目的地址寄存器 (DMA_DAR0)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11.6.2.10 DMA 通道 0 数据传输数量寄存器 (DMA_NDT0)

DMA 通道 0 数据传输数量寄存器 (DMA_NDT0)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.11 DMA 通道 1 控制寄存器 (DMA_CON1)

DMA 通道 1 控制寄存器 (DMA_CON1)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

位域名称	位域范围	读写属性	描述
Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁 b0001: 发生 2 次DMA传输后仲裁 b0010: 发生 4 次DMA传输后仲裁 b0011: 发生 8 次DMA传输后仲裁 b0100: 发生 16 次DMA传输后仲裁 b0101: 发生 32 次DMA传输后仲裁 b0110: 发生 64 次DMA传输后仲裁 b0111: 发生 128 次DMA传输后仲裁 b1000: 发生 256 次DMA传输后仲裁 b1001: 发生 512 次DMA传输后仲裁 b1010-b1111: 发生 1024 次DMA传输后仲裁。</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式 1: 使能目的地地址增量模式</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留</p>

			只能在 CHEN = 0 时对此位域执行写操作。
SINC	Bit 8	R/W	<p>来源地址增量模式使能位</p> <p>0: 禁止来源地址增量模式</p> <p>1: 使能来源地址增量模式</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
<i>Reserved</i>	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位</p> <p>b00: 低</p> <p>b01: 中</p> <p>b10: 高</p> <p>b11: 最高</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位</p> <p>0: 禁止</p> <p>1: 使能</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位</p> <p>0: 外设到存储器</p> <p>1: 存储器到外设</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位</p> <p>0: 禁止</p> <p>1: 使能</p> <p>在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位</p> <p>0: 禁止</p> <p>1: 使能</p>

11.6.2.12 DMA 通道 1 源地址寄存器 (DMA_SAR1)

DMA 通道 1 源地址寄存器 (DMA_SAR1)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.13 DMA 通道 1 目的地址寄存器 (DMA_DAR1)

DMA 通道 1 目的地址寄存器 (DMA_DAR1)																															
偏移地址: 38 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11.6.2.14 DMA 通道 1 数据传输数量寄存器 (DMA_NDT1)

DMA 通道 1 数据传输数量寄存器 (DMA_NDT1)																															
偏移地址: 3C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.15 DMA 通道 2 控制寄存器 (DMA_CON2)

DMA 通道 2 控制寄存器 (DMA_CON2)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁</p> <p>b0001: 发生 2 次DMA传输后仲裁</p> <p>b0010: 发生 4 次DMA传输后仲裁</p> <p>b0011: 发生 8 次DMA传输后仲裁</p> <p>b0100: 发生 16 次DMA传输后仲裁</p> <p>b0101: 发生 32 次DMA传输后仲裁</p> <p>b0110: 发生 64 次DMA传输后仲裁</p> <p>b0111: 发生 128 次DMA传输后仲裁</p> <p>b1000: 发生 256 次DMA传输后仲裁</p> <p>b1001: 发生 512 次DMA传输后仲裁</p> <p>b1010-b1111: 发生 1024 次DMA传输后仲裁。</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式</p> <p>1: 使能目的地地址增量模式</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位</p> <p>0: 禁止来源地址增量模式 1: 使能来源地址增量模式</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
Reserved	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位</p> <p>b00: 低 b01: 中 b10: 高 b11: 最高</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位</p> <p>0: 禁止 1: 使能</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位</p> <p>0: 外设到存储器 1: 存储器到外设</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位</p> <p>0: 禁止 1: 使能</p> <p>在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位</p> <p>0: 禁止 1: 使能</p>

11.6.2.16 DMA 通道 2 源地址寄存器 (DMA_SAR2)

DMA 通道 2 源地址寄存器 (DMA_SAR2)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.17 DMA 通道 2 目的地址寄存器 (DMA_DAR2)

DMA 通道 2 目的地址寄存器 (DMA_DAR2)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11.6.2.18 DMA 通道 2 数据传输数量寄存器 (DMA_NDT2)

DMA 通道 2 数据传输数量寄存器 (DMA_NDT2)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.19 DMA 通道 3 控制寄存器 (DMA_CON3)

DMA 通道 3 控制寄存器 (DMA_CON3)																															
偏移地址: 50 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁 b0001: 发生 2 次DMA传输后仲裁 b0010: 发生 4 次DMA传输后仲裁 b0011: 发生 8 次DMA传输后仲裁 b0100: 发生 16 次DMA传输后仲裁 b0101: 发生 32 次DMA传输后仲裁 b0110: 发生 64 次DMA传输后仲裁 b0111: 发生 128 次DMA传输后仲裁 b1000: 发生 256 次DMA传输后仲裁 b1001: 发生 512 次DMA传输后仲裁 b1010-b1111: 发生 1024 次DMA传输后仲裁。</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式 1: 使能目的地地址增量模式</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位 0: 禁止来源地址增量模式 1: 使能来源地址增量模式 只能在 CHEN = 0 时对此位执行写操作。</p>
Reserved	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位 b00: 低 b01: 中 b10: 高 b11: 最高 只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位 0: 禁止 1: 使能 只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位 0: 外设到存储器 1: 存储器到外设 只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位 0: 禁止 1: 使能 在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位 0: 禁止 1: 使能</p>

11.6.2.20 DMA 通道 3 源地址寄存器 (DMA_SAR3)

DMA 通道 3 源地址寄存器 (DMA_SAR3)																															
偏移地址: 54 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.21 DMA 通道 3 目的地址寄存器 (DMA_DAR3)

DMA 通道 3 目的地址寄存器 (DMA_DAR3)																															
偏移地址: 58 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11. 6. 2. 22 DMA 通道 3 数据传输数量寄存器 (DMA_NDT3)

DMA 通道 3 数据传输数量寄存器 (DMA_NDT3)																															
偏移地址: 5C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.23 DMA 通道 4 控制寄存器 (DMA_CON4)

DMA 通道 4 控制寄存器 (DMA_CON4)																															
偏移地址: 60 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL<1:0>	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁</p> <p>b0001: 发生 2 次DMA传输后仲裁</p> <p>b0010: 发生 4 次DMA传输后仲裁</p> <p>b0011: 发生 8 次DMA传输后仲裁</p> <p>b0100: 发生 16 次DMA传输后仲裁</p> <p>b0101: 发生 32 次DMA传输后仲裁</p> <p>b0110: 发生 64 次DMA传输后仲裁</p> <p>b0111: 发生 128 次DMA传输后仲裁</p> <p>b1000: 发生 256 次DMA传输后仲裁</p> <p>b1001: 发生 512 次DMA传输后仲裁</p> <p>b1010-b1111: 发生 1024 次DMA传输后仲裁。</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式</p> <p>1: 使能目的地地址增量模式</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节</p> <p>b01: 半字</p> <p>b10: 字</p> <p>b11: 保留</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位 0: 禁止来源地址增量模式 1: 使能来源地址增量模式 只能在 CHEN = 0 时对此位执行写操作。</p>
Reserved	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位 b00: 低 b01: 中 b10: 高 b11: 最高 只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位 0: 禁止 1: 使能 只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位 0: 外设到存储器 1: 存储器到外设 只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位 0: 禁止 1: 使能 在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位 0: 禁止 1: 使能</p>

11.6.2.24 DMA 通道 4 源地址寄存器 (DMA_SAR4)

DMA 通道 4 源地址寄存器 (DMA_SAR4)																															
偏移地址: 64 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.25 DMA 通道 4 目的地址寄存器 (DMA_DAR4)

DMA 通道 4 目的地址寄存器 (DMA_DAR4)																															
偏移地址: 68 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11.6.2.26 DMA 通道 4 数据传输数量寄存器 (DMA_NDT4)

DMA 通道 4 数据传输数量寄存器 (DMA_NDT4)																															
偏移地址: 6C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.27 DMA 通道 5 控制寄存器 (DMA_CON5)

DMA 通道 5 控制寄存器 (DMA_CON5)																															
偏移地址: 70 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁 b0001: 发生 2 次DMA传输后仲裁 b0010: 发生 4 次DMA传输后仲裁 b0011: 发生 8 次DMA传输后仲裁 b0100: 发生 16 次DMA传输后仲裁 b0101: 发生 32 次DMA传输后仲裁 b0110: 发生 64 次DMA传输后仲裁 b0111: 发生 128 次DMA传输后仲裁 b1000: 发生 256 次DMA传输后仲裁 b1001: 发生 512 次DMA传输后仲裁 b1010-b1111: 发生 1024 次DMA传输后仲裁。 只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留 只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式 1: 使能目的地地址增量模式 只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留 只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位</p> <p>0: 禁止来源地址增量模式 1: 使能来源地址增量模式</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
Reserved	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位</p> <p>b00: 低 b01: 中 b10: 高 b11: 最高</p> <p>只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位</p> <p>0: 禁止 1: 使能</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位</p> <p>0: 外设到存储器 1: 存储器到外设</p> <p>只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位</p> <p>0: 禁止 1: 使能</p> <p>在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位</p> <p>0: 禁止 1: 使能</p>

11.6.2.28 DMA 通道 5 源地址寄存器 (DMA_SAR5)

DMA 通道 5 源地址寄存器 (DMA_SAR5)																															
偏移地址: 74 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.29 DMA 通道 5 目的地址寄存器 (DMA_DAR5)

DMA 通道 5 目的地址寄存器 (DMA_DAR5)																															
偏移地址: 78 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11. 6. 2. 30 DMA 信道 5 数据传输数量寄存器 (DMA_NDT5)

DMA 通道 5 数据传输数量寄存器 (DMA_NDT5)																															
偏移地址: 7C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.31 DMA 通道 6 控制寄存器 (DMA_CON6)

DMA 通道 6 控制寄存器 (DMA_CON6)																															
偏移地址: 80 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MAX_BURST				Reserved	DDWSEL	DINC	Reserved	SDWSEL	SINC	Reserved	CHPRI	M2M	DIR	CIRC	CHEN				

Reserved	Bit 31:20	—	保留
MAX_BURST	Bit 19:16	—	<p>控制器重新仲裁前的DMA 传输次数</p> <p>b0000: 发生 1 次DMA传输后仲裁 b0001: 发生 2 次DMA传输后仲裁 b0010: 发生 4 次DMA传输后仲裁 b0011: 发生 8 次DMA传输后仲裁 b0100: 发生 16 次DMA传输后仲裁 b0101: 发生 32 次DMA传输后仲裁 b0110: 发生 64 次DMA传输后仲裁 b0111: 发生 128 次DMA传输后仲裁 b1000: 发生 256 次DMA传输后仲裁 b1001: 发生 512 次DMA传输后仲裁 b1010-b1111: 发生 1024 次DMA传输后仲裁。 只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 15	—	保留
DDWSEL	Bit 14:13	R/W	<p>目的地传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留 只能在 CHEN = 0 时对此位域执行写操作。</p>
DINC	Bit 12	R/W	<p>目的地地址增量模式使能位</p> <p>0: 禁止目的地地址增量模式 1: 使能目的地地址增量模式 只能在 CHEN = 0 时对此位域执行写操作。</p>
Reserved	Bit 11	—	保留
SDWSEL	Bit 10:9	R/W	<p>来源传输数据宽度选择位</p> <p>b00: 字节 b01: 半字 b10: 字 b11: 保留 只能在 CHEN = 0 时对此位域执行写操作。</p>

SINC	Bit 8	R/W	<p>来源地址增量模式使能位 0: 禁止来源地址增量模式 1: 使能来源地址增量模式 只能在 CHEN = 0 时对此位执行写操作。</p>
Reserved	Bit 7:6	—	保留
CHPRI	Bit 5:4	R/W	<p>通道优先级选择位 b00: 低 b01: 中 b10: 高 b11: 最高 只能在 CHEN = 0 时对此位域执行写操作。</p>
M2M	Bit 3	R/W	<p>存储器到存储器模式使能位 0: 禁止 1: 使能 只能在 CHEN = 0 时对此位执行写操作。</p>
DIR	Bit 2	R/W	<p>数据传输方向选择位 0: 外设到存储器 1: 存储器到外设 只能在 CHEN = 0 时对此位执行写操作。</p>
CIRC	Bit 1	R/W	<p>循环模式使能位 0: 禁止 1: 使能 在CHEN = 1 时只能对此位设置为 0。</p>
CHEN	Bit 0	R/W	<p>通道使能位 0: 禁止 1: 使能</p>

11.6.2.32 DMA 通道 6 源地址寄存器 (DMA_SAR6)

DMA 通道 6 源地址寄存器 (DMA_SAR6)																															
偏移地址: 84 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAR																															

SAR	Bit 31:0	R/W	来源数据起始地址寄存器 DMA 的 32 位来源起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	---

11.6.2.33 DMA 通道 6 目的地址寄存器 (DMA_DAR6)

DMA 通道 6 目的地址寄存器 (DMA_DAR6)																															
偏移地址: 88 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAR																															

DAR	Bit 31:0	R/W	目的地数据起始地址 DMA 的 32 位目的地起始地址。 只能在 CHEN = 0 时对此位域执行写操作。
-----	----------	-----	--

11.6.2.34 DMA 通道 6 数据传输数量寄存器 (DMA_NDT6)

DMA 通道 6 数据传输数量寄存器 (DMA_NDT6)																															
偏移地址: 8C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDT																TNDT															

NRDT	Bit 31:16	R	剩余数据传输数量 DMA 的剩余数据传输数量。
TNDT	Bit 15:0	R/W	总数据传输数量 DMA 的总数据传输数量。 只能在 CHEN = 0 时对此位域执行写操作。

11.6.2.35 DMA通道x复用选择寄存器 (DMA_CHx_SELCON) (x=0...6)

DMA 通道 0 复用选择寄存器 (DMA_CH0_SELCON)																															
偏移地址: 4*x+00 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																MSEL				Reserved				MSIGSEL							

Reserved	Bit 31-13	-	保留
MSEL	Bit 12-8	R/W	输入源选择 00000: 无输入 00001: GPIO 00010: ADC 00011: CRC 00100: EUART0 00101: EUART1 00110: CUART0 00111: CUART1 01000: CUART2 01001: SPI0 01010: SPI1 01011: I2C0 01100: I2C1 01101: AD16C4T 01110: GP16C4T0 01111: GP16C4T1 10000: GP16C4T2 10001: PIS 10010: BS16T 其他: 保留
Reserved	Bit 7-4	-	保留
MSIGSEL	Bit 3-0	R/W	MSEL=00000 MSIGSEL 无效 MSEL=00001 0000: EXTIO 0001: EXTI1 0010: EXTI2 0011: EXTI3 0100: EXTI4 0101: EXTI5 0110: EXTI6 0111: EXTI7

			<p>1000: EXTI8 1001: EXTI9 1010: EXTI10 1011: EXTI11 1100: EXTI12 1101: EXTI13 1110: EXTI14 1111: EXTI15</p> <p>MSEL=00010 (SRC=ADC) ADC DMA 申请</p> <p>MSEL=00011 (SRC=CRC) CRC DMA 申请</p> <p>MSEL=00100,00101,00110,00111,01000 (SRC=UART0~4) 0000: 接收缓冲器非空申请 0001: 发送缓冲器空申请</p> <p>MSEL=01001, 01010 (SRC=SPI0~1) 0000: 接收缓冲器非空申请 0001: 发送缓冲器空申请</p> <p>MSEL=01011, 01100 (I2C0~1) 0000: 接收缓冲器非空申请 0001: 发送缓冲器空申请</p> <p>MSEL=01101, 01110, 01111, 10000 (SRC=AD16C4T, GP16C4T0, GP16C4T1, GP16C4T2) 0000: 捕捉比较通道 1 申请 0001: 捕捉比较通道 2 申请 0010: 捕捉比较通道 3 申请 0011: 捕捉比较通道 4 申请 0100: TIMER 触发申请 0101: TIMER 比较匹配申请 0110: TIMER 更新事件申请</p> <p>MSEL=10001 (SRC=PIS) 0000: PIS 通道 0 0001: PIS 通道 1 0010: PIS 通道 2 0011: PIS 通道 3 0100: PIS 通道 4 0101: PIS 通道 5 0110: PIS 通道 6 0111: PIS 通道 7</p> <p>MSEL=10010 (SRC=BS16T) BS16T DMA 申请</p>
--	--	--	---

第12章 外设互联（PIS）

12.1 概述

PIS（Peripheral Interaction System）在微控制器中作为外设互联的桥接口使用，利用 PIS 可实现外设之间的相互触发，控制及自动化工作，提高系统的实时性和快速响应能力，可避免占用过多的 CPU 资源并简化软件工作，提高系统效率。

12.2 特性

- ◆ 最多支持 8 个 PIS 通道选择
- ◆ 支持同步和异步通道选择
- ◆ 支持信号有效边缘选择
- ◆ UART 输出调制可配置
- ◆ 所有 GPIO 可用作 PIS 输入或输出

12.3 结构框图

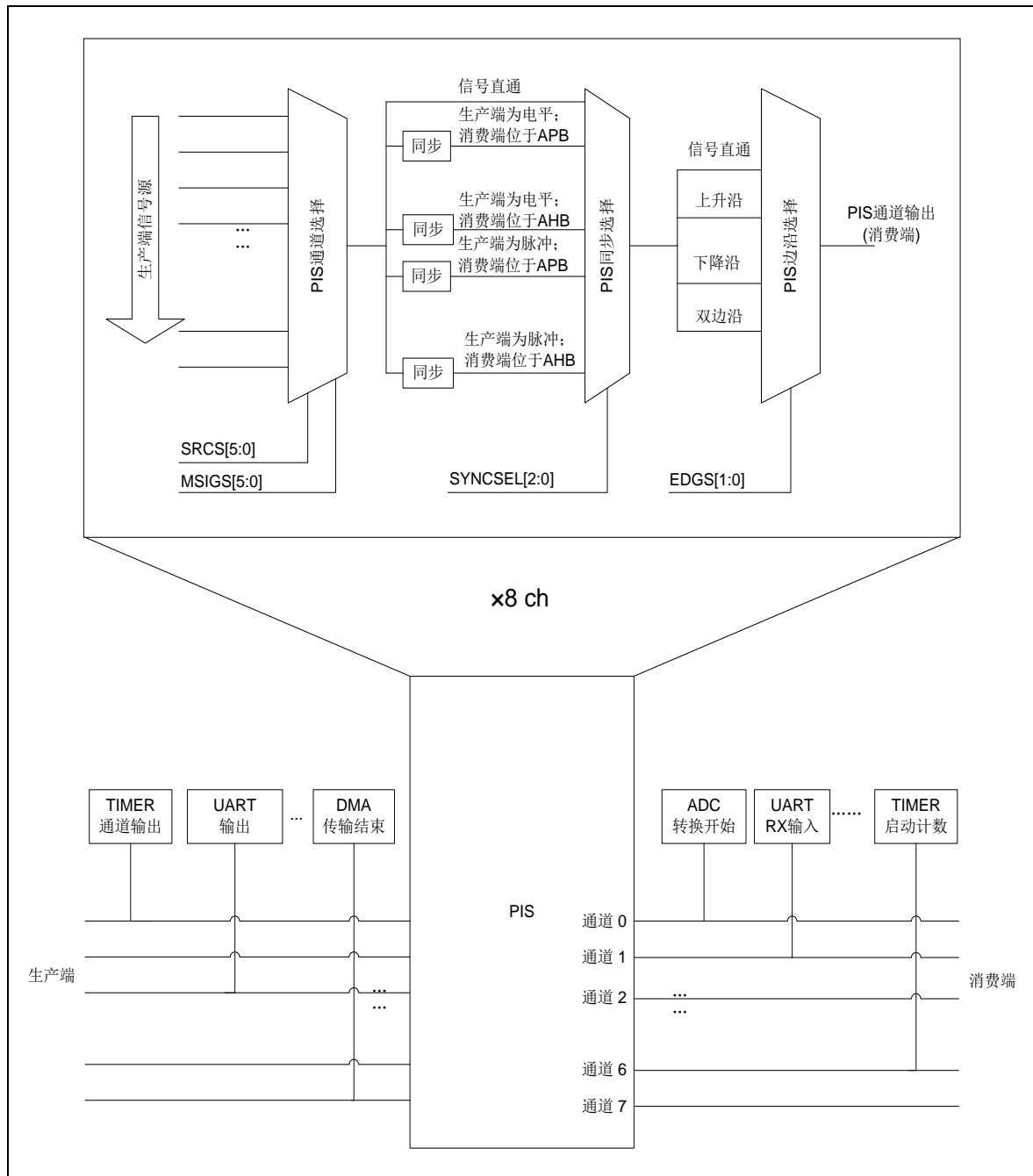


图 12-1 PIS 结构框图

12.4 功能描述

外设互联可支持 8 个通道资源，每个通道均可对生产端信号进行多路复用。针对不同应用可灵活配置。

12.4.1 生产端信号

外设互联的生产端信号如下表所示：

	生产端	输出形式	异步支持	位置 (APB 或 AHB)
GPIO	PA0~15、PB0~15、PC0~15、 PD0~2 输入 (由 SYSCFG_PISIOCFG 配置)	电平	是	AHB 外设
UART	发送空状态中断	脉冲	—	APB 外设
	接收数据中断	脉冲	—	
	RTS 输出	电平	—	
	TX 输出	电平	—	
SPI	接收缓冲器非空	脉冲	—	APB 外设
	发送缓冲器空	脉冲	—	
	片选输出	电平	—	
I2C	接收缓冲器非空	电平	—	APB 外设
	发送缓冲器空	电平	—	
TIMER	更新事件	脉冲	—	APB 外设
	触发事件	脉冲	—	
	输入捕获	脉冲	—	
	输出比较	脉冲	—	
ADC	标准转换组转换结束	脉冲	—	APB 外设
	插入组转换结束	脉冲	—	
LVD	LVD 检测输出	电平	是	—
DMA	DMA 通道完成	脉冲	是	AHB 外设

表 12-1 生产端信号

12.4.2 消费端信号

外设互联的消费端信号如下表所示：

	消费端	输入形式	异步支持	位置 (APB 或 AHB)
UART	RX 输入	电平	—	APB 外设
SPI	RX 输入	电平	—	APB 外设
	CLK 输入	电平	—	
TIMER	启动	脉冲	—	APB 外设
	停止	脉冲	—	
	清零	脉冲	—	

	比较捕捉通道输入	电平或脉冲	—	
	通道输出清除	电平或脉冲	—	
	刹车输入	电平	—	
ADC	启动标准转换组转换	脉冲	—	APB 外设
	启动插入组转换	脉冲	—	
EXT_INT	CPU 外部中断	电平	—	CPU

表 12-2 消费端信号

消费端信号的 PIS 通道分配如下表所示：

	消费端	源通道	备注
UART0	RX 输入	PIS 通道 3	由 PIS_TAR_CON1.UART0_RXD_SEL 设定
UART1	RX 输入	PIS 通道 4	由 PIS_TAR_CON1.UART1_RXD_SEL 设定
UART2	RX 输入	PIS 通道 5	由 PIS_TAR_CON1.UART2_RXD_SEL 设定
UART3	RX 输入	PIS 通道 6	由 PIS_TAR_CON1.UART3_RXD_SEL 设定
UART4	RX 输入	PIS 通道 7	由 PIS_TAR_CON1.UART4_RXD_SEL 设定
SPI0	RX 输入	PIS 通道 0	由 PIS_TAR_CON1.SPI0_RX_SEL 设定
	CLK 输入	PIS 通道 1	由 PIS_TAR_CON1.SPI0_CLK_SEL 设定
SPI1	RX 输入	PIS 通道 2	由 PIS_TAR_CON1.SPI1_RX_SEL 设定
	CLK 输入	PIS 通道 3	由 PIS_TAR_CON1.SPI1_CLK_SEL 设定
TIM0 (AD16T4C0)	ITR0	PIS 通道 4	PCLK 同步
	ITR1	PIS 通道 5	PCLK 同步
	ITR2	PIS 通道 6	PCLK 同步
	ITR3	PIS 通道 7	PCLK 同步
	捕捉通道 1	PIS 通道 1	由 PIS_TAR_CON0.TIM0_CH1IN_SEL 设定
	捕捉通道 2	PIS 通道 2	由 PIS_TAR_CON0.TIM0_CH2IN_SEL 设定
	捕捉通道 3	PIS 通道 3	由 PIS_TAR_CON0.TIM0_CH3IN_SEL 设定
	捕捉通道 4	PIS 通道 4	由 PIS_TAR_CON0.TIM0_CH4IN_SEL 设定
	刹车输入	PIS 通道 0	由 PIS_TAR_CON0.TIM0_BRKIN_SEL 设定 注意只有高级定时器可支持刹车输入。
	通道输出清除源 0	PIS 通道 0	—
	通道输出清除源 1	PIS 通道 1	—
	通道输出清除源 2	PIS 通道 2	—
	通道输出清除源 3	PIS 通道 3	—
TIM1 (GP16T4C0)	ITR0	PIS 通道 4	PCLK 同步
	ITR1	PIS 通道 5	PCLK 同步
	ITR2	PIS 通道 6	PCLK 同步
	ITR3	PIS 通道 7	PCLK 同步
	捕捉通道 1	PIS 通道 1	由 PIS_TAR_CON0.TIM1_CH1IN_SEL 设定
	捕捉通道 2	PIS 通道 2	由 PIS_TAR_CON0.TIM1_CH2IN_SEL 设定
	捕捉通道 3	PIS 通道 3	由 PIS_TAR_CON0.TIM1_CH3IN_SEL 设定
	捕捉通道 4	PIS 通道 4	由 PIS_TAR_CON0.TIM1_CH4IN_SEL 设定
	通道输出清除源 0	PIS 通道 0	PCLK 同步

	消费端	源通道	备注
	通道输出清除源 1	PIS 通道 1	PCLK 同步
	通道输出清除源 2	PIS 通道 2	PCLK 同步
	通道输出清除源 3	PIS 通道 3	PCLK 同步
TIM2 (GP16T4C1)	ITR0	PIS 通道 4	PCLK 同步
	ITR1	PIS 通道 5	PCLK 同步
	ITR2	PIS 通道 6	PCLK 同步
	ITR3	PIS 通道 7	PCLK 同步
	捕捉通道 1	PIS 通道 5	由 PIS_TAR_CON0.TIM2_CH1IN_SEL 设定
	捕捉通道 2	PIS 通道 6	由 PIS_TAR_CON0.TIM2_CH2IN_SEL 设定
	捕捉通道 3	PIS 通道 7	由 PIS_TAR_CON0.TIM2_CH3IN_SEL 设定
	捕捉通道 4	PIS 通道 4	由 PIS_TAR_CON0.TIM2_CH4IN_SEL 设定
	通道输出清除源 0	PIS 通道 0	PCLK 同步
	通道输出清除源 1	PIS 通道 1	PCLK 同步
	通道输出清除源 2	PIS 通道 2	PCLK 同步
	通道输出清除源 3	PIS 通道 3	PCLK 同步
	TIM3 (GP16T4C2)	ITR0	PIS 通道 4
ITR1		PIS 通道 5	PCLK 同步
ITR2		PIS 通道 6	PCLK 同步
ITR3		PIS 通道 7	PCLK 同步
捕捉通道 1		PIS 通道 5	由 PIS_TAR_CON0.TIM3_CH1IN_SEL 设定
捕捉通道 2		PIS 通道 6	由 PIS_TAR_CON0.TIM3_CH2IN_SEL 设定
捕捉通道 3		PIS 通道 7	由 PIS_TAR_CON0.TIM3_CH3IN_SEL 设定
捕捉通道 4		PIS 通道 4	由 PIS_TAR_CON0.TIM3_CH4IN_SEL 设定
通道输出清除源 0		PIS 通道 0	PCLK 同步
通道输出清除源 1		PIS 通道 1	PCLK 同步
通道输出清除源 2		PIS 通道 2	PCLK 同步
通道输出清除源 3		PIS 通道 3	PCLK 同步
ADC		启动标准转换组转换	PIS 通道 4
	启动插入组转换	PIS 通道 5	—
EXT_INT	外部中断请求	PIS 通道 0	由 PIS_TAR_CON1.EXT_INT_SEL 设定
DMA	DMA 传输请求	任意一个 PIS 通道	由 DMA_CHx_SELCON 设定

表 12-3 消费端的 PIS 通道分配

12.4.3 PIS通道选择

PIS 的源端定义为生产端信号，PIS 的输出信号用于消费端。消费端可根据应用需要，来选择合适的生产端信号，并通过选择同步路径以保证正确采样到生产端信号。PIS 的生产端信号选择由 PIS 通道控制寄存器（PIS_CHx_CON，x=0~7）配置。PIS 通道控制寄存器的 PIS_CHx_CON.SRCS 位（x=0~7）用来选择生产端模块，配置 PIS_CHx_CON.MSIGS 位（x=0~7）则可从生产端模块的多路信号中选择一路作为生产端

信号。

以下从几种情况来举例说明 PIS 的配置。

12.4.3.1 同一时钟域互联

以 GP16T4C0 和 ADC 为例，在 GP16T4C0 使用 PCLK 作为计数时钟的情况下，产生与 PCLK 同步的更新事件去触发 ADC 转换动作。可参照如下配置：

1. 通过上表可知 ADC 标准转换组转换的启动信号为 PIS 通道 4。
2. 设定寄存器 PIS_CH4_CON.SRCS 为 011100 选择 GP16T4C0 为生产端模块。
3. 设定寄存器 PIS_CH4_CON.MSIGS 为 0000，选择 GP16T4C0 的同步更新事件（PCLK 同步）作为生产端信号。
4. 设定寄存器 PIS_CH4_CON.SYNCSEL 为 000（信号直通），并将 PIS_CH4_CON.EDGS 设为 00（不输出边沿），此时 TSCKS 可任意设定。
5. 配置 ADC 选择外部触发方式进行标准转换组转换。
6. 配置 GP16T4C0 进行计数，产生更新事件后将触发 ADC 开始 AD 转换。

12.4.3.2 AHB和APB外设之间互联

以 DMA（位于 AHB）和 ADC（位于 APB）为例，ADC 完成转换后去触发 DMA 传输动作。可参照如下配置：

1. 设定寄存器 PIS_CH7_CON.SRCS 为 000110 选择 ADC 为生产端模块。
2. 设定寄存器 PIS_CH7_CON.MSIGS 为 0001，选择 ADC 的标准转换组转换结束事件（PCLK 同步）作为生产端信号。
3. 设定寄存器 PIS_CH7_CON.SYNCSEL 为 110（生产端为脉冲信号，消费端位于 AHB 时钟域），并将 PIS_CH7_CON.EDGS 设为 00（不输出边沿），此时 PIS_CH7_CON.TSCKS 设为 10。
4. 设定寄存器 DMA_CHx_SELCON，配置 DMA 选择 PIS 通道 7 触发传输。
5. 配置 ADC 开始 AD 转换。

12.4.3.3 生产端信号为异步信号

以 GPIO 和 ADC（位于 APB）为例，GPIO 事件脉冲去触发 ADC 转换动作。可参照如下配置：

1. 通过上表 12-3 消费端的 PIS 通道分配可知 ADC 标准转换组转换的启动信号为 PIS 通道 4。
2. 设定寄存器 PIS_CH4_CON.SRCS 为 000001 选择 GPIO 为生产端模块。
3. 设定寄存器 PIS_CH4_CON.MSIGS 为 0000，选择 PA0 事件脉冲作为生产端信号。
4. 设定寄存器 PIS_CH4_CON.SYNCSEL 为 010（生产端为电平信号，消费端位于 APB 时钟域），并将 PIS_CH4_CON.EDGS 设为 01（上升沿），此时 TSCKS 可任

意设定。

5. 配置 ADC 选择外部触发方式进行标准转换组转换。
6. 配置 GPIO 产生事件脉冲后触发 ADC 开始 AD 转换。

12.4.4 UART输出调制

UART 的输出调制功能利用定时器 PWM 波或 BUZ 信号对 UART 的 TX 调制后发送到端口。

调制方式如下图所示：

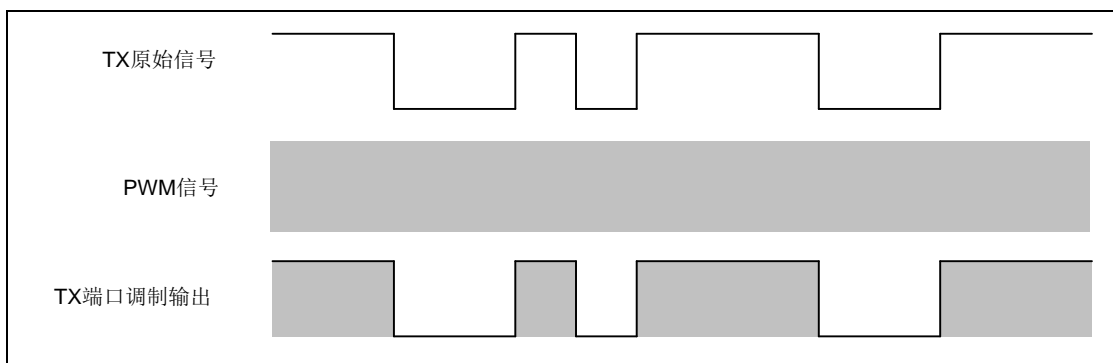


图 12-2 高电平调制输出波形图

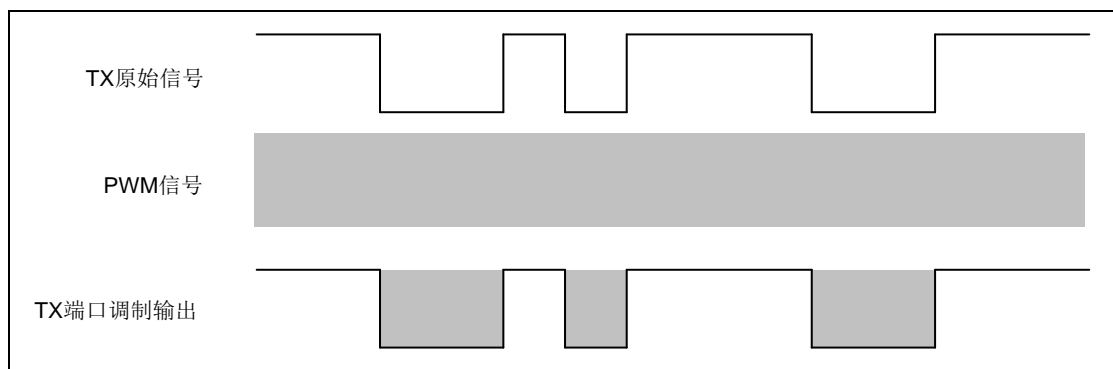


图 12-3 低电平调制输出波形图

以下为 UART 输出调制的参考配置流程（以 AD16C4T 调制 UART0 为例）：

1. 设置寄存器 `UART0_TXMCR.TXMSS` 为 0001，选择 AD16C4T 作为调制源。
2. 设置寄存器 `UART0_TXMCR.TXSIGS` 为 0000，选择 AD16C4T 通道 1 输出作为调制信号。
3. 设置 `UART0_TXMCR.TXMLVLS` 选择调制电平。
4. 配置 AD16C4T 进行计数。
5. 配置 UART 发送数据。

12.5 特殊功能寄存器

12.5.1 寄存器列表

PIS 寄存器列表		
名称	偏移地址	描述
PIS_CH0_CON	0000 _H	PIS 通道 0 控制寄存器
PIS_CH1_CON	0004 _H	PIS 通道 1 控制寄存器
PIS_CH2_CON	0008 _H	PIS 通道 2 控制寄存器
PIS_CH3_CON	000C _H	PIS 通道 3 控制寄存器
PIS_CH4_CON	0010 _H	PIS 通道 4 控制寄存器
PIS_CH5_CON	0014 _H	PIS 通道 5 控制寄存器
PIS_CH6_CON	0018 _H	PIS 通道 6 控制寄存器
PIS_CH7_CON	001C _H	PIS 通道 7 控制寄存器
Reserved	0020 _H ~003C _H	保留
PIS_CH_OER	0040 _H	PIS 通道端口输出使能寄存器
PIS_TAR_CON0	0044 _H	PIS 消费端通道控制寄存器 0
PIS_TAR_CON1	0048 _H	PIS 消费端通道控制寄存器 1
Reserved	004C _H ~005C _H	保留
UART0_TXMCR	0060 _H	UART0 输出调制控制寄存器
UART1_TXMCR	0064 _H	UART1 输出调制控制寄存器
UART2_TXMCR	0068 _H	UART2 输出调制控制寄存器
UART3_TXMCR	006C _H	UART3 输出调制控制寄存器
UART4_TXMCR	0070 _H	UART4 输出调制控制寄存器

12.5.2 寄存器描述

12.5.2.1 PIS通道x控制寄存器 (PIS_CHx_CON) (x=0..7)

PIS 通道 x 控制寄存器 (PIS_CHx_CON) (x=0..15)																															
偏移地址: 4*x+00 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVEL	PULSE	Reserved			SYNCSEL			Reserved				TSCKS		EDGS		Reserved		SRCS					Reserved			MSIGS					

LEVEL	Bit 31	R/W	软件电平产生位 该位置 1 后产生一次 PIS 触发电平，等待足够的时间后，需软件将其清零。等待时间需确保消费端时钟能够采集到至少一次高电平。
PULSE	Bit 30	R/W	软件脉冲产生触发位 该位置 1 后产生一次 PIS 触发脉冲，产生完成后自动清零。
Reserved	Bit 29-27	—	保留
SYNCSEL	Bit 26-24	R/W	信号同步选择位 000: 信号直通。 在以下场合可设置信号直通： 在一些低功耗应用的场合需采用异步的电平或脉冲； 生产端信号源与消费端处于同一时钟域，无需同步。 001、010: 生产端为电平信号，消费端位于 APB 时钟域 011: 生产端为电平信号，消费端位于 AHB 时钟域 100、101: 生产端为脉冲信号，消费端位于 APB 时钟域 110: 生产端为脉冲信号，消费端位于 AHB 时钟域 111: 预留
Reserved	Bit 23-20	—	保留
TSCKS	Bit 19-18	R/W	触发采样时钟选择位 在 SYNCSEL=000，且触发信号源与触发目标处于同一时钟域时，请根据触发目标时钟域进行设置，比如触发目标为 ADC，请将 TSCKS 设为 01 选择 PCLK。其余情况下，触发采样时钟是确定的，该设定无效。 00、01: PCLK

			10: HCLK 11: 预留
EDGS	Bit 17-16	R/W	边沿选择位 在输入信号为电平时使用，低功耗应用场合下的异步通道请设置为 00。 00: 不输出边沿 01: 上升沿 10: 下降沿 11: 双边沿
Reserved	Bit 15-14	—	保留
SRCS	Bit 13-8	R/W	输入源选择位 000000: 软件触发 000001: GPIO 000010: 系统时钟 000011: 保留 000100: 保留 000101: 保留 000110: ADC0 000111: LVD 001000: UART0 001001: UART1 001010: UART2 001011: UART3 001100: UART4 001101: 保留 001110: SPI0 001111: SPI1 010000: I2C0 010001: I2C1 010010: TIM0 (AD16C4T) 010011: 保留 010100: 保留 010101: 保留 010110: 保留 010111: 保留 011000: 保留 011001: DMA 011010: 保留 011011: TIM4 (BS16T) 011100: TIM5 (GP16C4T0) 011101: TIM6 (GP16C4T1) 011110: TIM7 (GP16C4T2) 011111: 保留 100000: 保留

			其他：保留
Reserved	Bit 7-4	—	保留
MSIGS	Bit 3-0	R/W	<p>输入源信号选择位</p> <p>SRCS=000000 (SRC=软件触发) 0000: LEVEL 电平信号 0001: PULSE 脉冲信号</p> <p>SRCS=000001 (SRC=GPIO, 参考“SYSCFG_PISIOCFG”设置) 0000: Px0 0001: Px1 0010: Px2 </p> <p>1110: Px14 1111: Px15</p> <p>SRCS=000010 (SRC=系统时钟) 0000: HSCO 时钟 0001: LSCO 时钟 0010: BUZ 时钟</p> <p>SRCS=000110 (SRC=ADC0) 0000: 插入组转换结束 0001: 标准转换组转换结束 0010: 模拟看门狗</p> <p>SRCS=000111 (SRC=LVD) 0000: LVD 输出 0001: 保留</p> <p>SRCS=001000, 001001, 001010, 001011, 001100 (SRC=UART0, UART1, UART2, UART3, UART4) 0000: 保留 0001: 保留 0010: 保留 0011: RTS 输出 0100: TX 输出 0101: 发送空状态中断 0110: 接收数据中断</p> <p>SRCS=001110, 001111 (SRC=SPI0, SPI1) 0000: 接收缓冲器非空 0001: 发送缓冲器空 0010: 片选信号输出</p> <p>SRCS=010000, 010001 (I2C0, I2C1) 0000: 接收缓冲器非空电平 0001: 发送缓冲器空电平</p> <p>SRCS=010010 (SRC=AD16C4T) 0000: 更新事件脉冲</p>

			<p>0001: TRGOUT 脉冲</p> <p>0010: 通道 1 捕获/比较脉冲</p> <p>0100: 通道 2 捕获/比较脉冲</p> <p>0110: 通道 3 捕获/比较脉冲</p> <p>1000: 通道 4 捕获/比较脉冲</p> <p>其他: 保留</p> <p>SRCS=011100, 011101, 011110 (SRC=GP16C4T0, GP16C4T1, GP16C4T2)</p> <p>0000: 更新事件脉冲</p> <p>0001: TRGOUT 脉冲</p> <p>0010: 通道 1 捕获/比较脉冲</p> <p>0100: 通道 2 捕获/比较脉冲</p> <p>0110: 通道 3 捕获/比较脉冲</p> <p>1000: 通道 4 捕获/比较脉冲</p> <p>其他: 保留</p> <p>SRCS= 011011 (SRC=BS16T)</p> <p>0000: 更新事件脉冲</p> <p>0001: TRGOUT 脉冲</p> <p>SRCS=011001 (SRC=DMA)</p> <p>0000~0110: DMA 通道 x 传输完成脉冲</p> <p>1111: DMA 所有通道传输完成</p>
--	--	--	--

12.5.2.2 PIS通道端口输出使能寄存器 (PIS_CH_OER)

PIS 通道端口输出使能寄存器 (PIS_CH_OER)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CH7OE	CH6OE	CH5OE	CH4OE	CH3OE	CH2OE	CH1OE	CH0OE

Reserved	Bit 31-8	—	保留
CH7OE	Bit 7	R/W	PIS 通道 7 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH6OE	Bit 6	R/W	PIS 通道 6 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH5OE	Bit 5	R/W	PIS 通道 5 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH4OE	Bit 4	R/W	PIS 通道 4 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH3OE	Bit 3	R/W	PIS 通道 3 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH2OE	Bit 2	R/W	PIS 通道 2 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH1OE	Bit 1	R/W	PIS 通道 1 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能
CH0OE	Bit 0	R/W	PIS 通道 0 输出至端口使能位 0: 输出到端口禁止 1: 输出到端口使能

12.5.2.3 PIS消费端通道控制寄存器 0 (PIS_TAR_CON0)

PIS 消费端通道控制寄存器 0 (PIS_TAR_CON0)																																
偏移地址: 44 _h																																
复位值: 00000000_00000000_00000000_00000000 _b																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved			TIM3_ETRIN_SEL	TIM3_CH4IN_SEL	TIM3_CH3IN_SEL	TIM3_CH2IN_SEL	TIM3_CH1IN_SEL	Reserved			TIM2_ETRIN_SEL	TIM2_CH4IN_SEL	TIM2_CH3IN_SEL	TIM2_CH2IN_SEL	TIM2_CH1IN_SEL	Reserved			TIM1_ETRIN_SEL	TIM1_CH4IN_SEL	TIM1_CH3IN_SEL	TIM1_CH2IN_SEL	TIM1_CH1IN_SEL	Reserved			TIM0_ETRIN_SEL	TIM0_BRKIN_SEL	TIM0_CH4IN_SEL	TIM0_CH3IN_SEL	TIM0_CH2IN_SEL	TIM0_CH1IN_SEL

Reserved	Bit 31-29	—	保留
TIM3_ETRIN_SEL	Bit 28	R/W	GP16C4T2 ETR 输入选择位 0: 从端口输入 1: 将 PIS 通道 6 输出作为输入
TIM3_CH4IN_SEL	Bit 27	R/W	GP16C4T2 输入捕捉通道 4 输入选择位 0: 从端口输入 1: 将 PIS 通道 0 输出作为输入
TIM3_CH3IN_SEL	Bit 26	R/W	GP16C4T2 输入捕捉通道 3 输入选择位 0: 从端口输入 1: 将 PIS 通道 7 输出作为输入
TIM3_CH2IN_SEL	Bit 25	R/W	GP16C4T2 输入捕捉通道 2 输入选择位 0: 从端口输入 1: 将 PIS 通道 6 输出作为输入
TIM3_CH1IN_SEL	Bit 24	R/W	GP16C4T2 输入捕捉通道 1 输入选择位 0: 从端口输入 1: 将 PIS 通道 5 输出作为输入
Reserved	Bit 23-21	—	保留
TIM2_ETRIN_SEL	Bit 20	R/W	GP16C4T1 ETR 输入选择位 0: 从端口输入 1: 将 PIS 通道 6 输出作为输入
TIM2_CH4IN_SEL	Bit 19	R/W	GP16C4T1 输入捕捉通道 4 输入选择位 0: 从端口输入 1: 将 PIS 通道 4 输出作为输入
TIM2_CH3IN_SEL	Bit 18	R/W	GP16C4T1 输入捕捉通道 3 输入选择位 0: 从端口输入 1: 将 PIS 通道 3 输出作为输入
TIM2_CH2IN_SEL	Bit 17	R/W	GP16C4T1 输入捕捉通道 2 输入选择位 0: 从端口输入 1: 将 PIS 通道 2 输出作为输入
TIM2_CH1IN_SEL	Bit 16	R/W	GP16C4T1 输入捕捉通道 1 输入选择位 0: 从端口输入

			1: 将 PIS 通道 1 输出作为输入
Reserved	Bit 15-13	—	保留
TIM1_ETRIN_SEL	Bit 12	R/W	GP16C4T0 ETR 输入选择位 0: 从端口输入 1: 将 PIS 通道 6 输出作为输入
TIM1_CH4IN_SEL	Bit 11	R/W	GP16C4T0 输入捕捉通道 4 输入选择位 0: 从端口输入 1: 将 PIS 通道 4 输出作为输入
TIM1_CH3IN_SEL	Bit 10	R/W	GP16C4T0 输入捕捉通道 3 输入选择位 0: 从端口输入 1: 将 PIS 通道 3 输出作为输入
TIM1_CH2IN_SEL	Bit 9	R/W	GP16C4T0 输入捕捉通道 2 输入选择位 0: 从端口输入 1: 将 PIS 通道 2 输出作为输入
TIM1_CH1IN_SEL	Bit 8	R/W	GP16C4T0 输入捕捉通道 1 输入选择位 0: 从端口输入 1: 将 PIS 通道 1 输出作为输入
Reserved	Bit 7-6	—	保留
TIM0_ETRIN_SEL	Bit 5	R/W	AD16C4T ETR 输入选择位 0: 从端口输入 1: 将 PIS 通道 5 输出作为输入
TIM0_BRKIN_SEL	Bit 4	R/W	AD16C4T 刹车输入选择位 0: 从端口输入 1: 将 PIS 通道 0 输出作为输入
TIM0_CH4IN_SEL	Bit 3	R/W	AD16C4T 输入捕捉通道 4 输入选择位 0: 从端口输入 1: 将 PIS 通道 4 输出作为输入
TIM0_CH3IN_SEL	Bit 2	R/W	AD16C4T 输入捕捉通道 3 输入选择位 0: 从端口输入 1: 将 PIS 通道 3 输出作为输入
TIM0_CH2IN_SEL	Bit 1	R/W	AD16C4T 输入捕捉通道 2 输入选择位 0: 从端口输入 1: 将 PIS 通道 2 输出作为输入
TIM0_CH1IN_SEL	Bit 0	R/W	AD16C4T 输入捕捉通道 1 输入选择位 0: 从端口输入 1: 将 PIS 通道 1 输出作为输入

12.5.2.4 PIS消费端通道控制寄存器 1 (PIS_TAR_CON1)

PIS 消费端通道控制寄存器 1 (PIS_TAR_CON1)																																				
偏移地址: 48 _H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved																EXT_INT_SEL	SPI1_CLK_SEL	SPI1_RX_SEL	SPI0_CLK_SEL	SPI0_RX_SEL	Reserved											UART4_RXD_SEL	UART3_RXD_SEL	UART2_RXD_SEL	UART1_RXD_SEL	UART0_RXD_SEL

Reserved	Bit 31-17	—	保留
EXT_INT_SEL	Bit 16	R/W	CPU EXT INT 输入选择位 0: 无效输入 1: 将 PIS 通道 0 输出作为输入
SPI1_CLK_SEL	Bit 15	R/W	SPI1 CLK 输入选择位 0: 从端口输入 1: 将 PIS 通道 3 输出作为输入
SPI1_RX_SEL	Bit 14	R/W	SPI1 RX 输入选择位 0: 从端口输入 1: 将 PIS 通道 2 输出作为输入
SPI0_CLK_SEL	Bit 13	R/W	SPI0 CLK 输入选择位 0: 从端口输入 1: 将 PIS 通道 1 输出作为输入
SPI0_RX_SEL	Bit 12	R/W	SPI0 RX 输入选择位 0: 从端口输入 1: 将 PIS 通道 0 输出作为输入
Reserved	Bit 11-5	—	保留
UART4_RXD_SEL	Bit 4	R/W	UART4 RXD 输入选择位 0: 从端口 RXD 输入 1: 将 PIS 通道 7 输出作为输入
UART3_RXD_SEL	Bit 3	R/W	UART3 RXD 输入选择位 0: 从端口 RXD 输入 1: 将 PIS 通道 6 输出作为输入
UART2_RXD_SEL	Bit 2	R/W	UART2 RXD 输入选择位 0: 从端口 RXD 输入 1: 将 PIS 通道 5 输出作为输入
UART1_RXD_SEL	Bit 1	R/W	UART1 RXD 输入选择位 0: 从端口 RXD 输入 1: 将 PIS 通道 4 输出作为输入
UART0_RXD_SEL	Bit 0	R/W	UART0 RXD 输入选择位

			0: 从端口 RXD 输入 1: 将 PIS 通道 3 输出作为输入
--	--	--	---------------------------------------

12.5.2.5 UARTx输出调制控制寄存器 (UARTx_TXMCR) (x=0..4)

UARTx 输出调制控制寄存器 (UARTx_TXMCR) (x=0..4)																															
偏移地址: 4*x+60 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							TXMLVLS		TXMSS			TXSIGs			

Reserved	Bit 31-9	—	保留
TXMLVLS	Bit 8	R/W	TX 调制电平选择位 0: 低电平调制 (TX 与所选取的调制信号进行硬件或操作) 1: 高电平调制 (TX 与所选取的调制信号进行硬件与操作)
TXMSS	Bit 7-4	R/W	TX 调制源选择位 0000: 调制禁止 0001: TIM0 (AD16C4T) 0010: 保留 0011: 保留 0100: TIM5 (GP16C4T0) 0101: TIM6 (GP16C4T1) 0110: TIM7 (GP16C4T2) 0111: 保留 1000: BUZ 其余: 无调制
TXSIGs	Bit 3-0	R/W	TX 调制信号选择位 TXMSS=0000 TXSIGs 无效 TXMSS=0001, 0100, 0101, 0110 (调制源=AD16C4T, GP16C4T0, GP16C4T1, GP16C4T2) 0000: TIMER 通道 1 0001: TIMER 通道 2 0010: TIMER 通道 3 0011: TIMER 通道 4 TXMSS=1000 (调制源=BUZ) 调制信号为 BUZ 输出 TXMSS=其余 无效

第13章 独立看门狗 (IWDT)

13.1 概述

独立看门狗 IWDT 可用于检测软件和硬件异常引起的故障，如主时钟停振、用户程序异常无法喂狗等；当计数器达到给定的超时值时，将触发系统复位。

独立看门狗 IWDT，当硬件使能时，时钟强制为独立的 LRC 时钟，且用户无法通过软件关闭 IWDT。

独立看门狗 IWDT 最适合于独立于主程序之外，并且对时间精度要求较低的场景。

13.2 特性

- ◆ 支持硬件使能和关闭看门狗
 - ◇ 芯片配置位 `CFG_WORD.IWDTEN` 配置为 1，则硬件使能 IWDT
 - ◇ 芯片配置位 `CFG_WORD.IWDTEN` 配置为 0，则硬件关闭 IWDT，但可以软件使能 IWDT
 - ◇ 硬件使能后不可通过软件关闭
 - ◇ 硬件使能后 IWDT 时钟强制为 32KHz LRC 时钟
- ◆ IWDT 溢出时间可设定
 - ◇ 写入 `IWDT_LOAD` 寄存器将重新加载看门狗
 - ◇ 溢出时产生 IWDT 复位
- ◆ IWDT 中断可唤醒 STOP 模式

13.3 功能描述

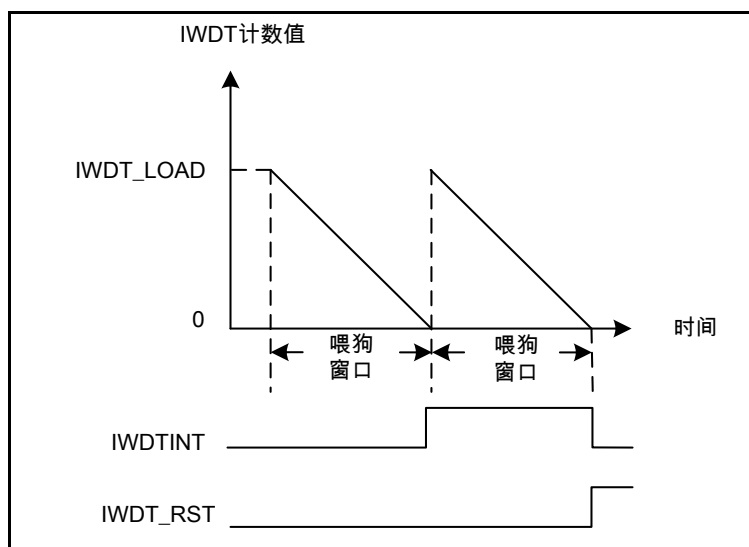


图 13-1 独立看门狗时序图

13.3.1 硬件看门狗

IWDT 看门狗可用于检测软件和硬件异常,用户可通过 `CFG_WORD.IWDTEN` 配置位启用硬件看门狗功能,以提高系统健壮性;硬件看门狗使能后,时钟强制为 32KHz LRC 时钟,即使系统时钟失效, IWDT 仍可正常工作。

当硬件看门狗使能时,系统上电后看门狗立即运行(时钟固定为 32KHz LRC 时钟); IWDT 载入 `IWDT_LOAD` 默认值 `0x0000_4000` (约 0.5s, 对应复位时间约 1s), 并从默认复位值递减计数。

计数器计数到 0 时, IWDT 产生中断标志,并在下一计数时钟到来时,计数器再次载入 `IWDT_LOAD`, 并继续递减计数;当计数器再次计数到 0 时,如果 IWDT 中断标志未被清零, IWDT 将产生复位信号。

`IWDT_INTCLR` 写入值为 `0xFFFFFFFF` 时, IWDT 中断标志位被清零,计数器重载 `IWDT_LOAD` 寄存器值,继续递减计数,写入其他值(非 `0xFFFFFFFF`)时,仅 IWDT 中断标志位被清零,计数器不重载。

喂狗操作

- ◆ 修改 `IWDT_LOAD.LOAD` 值,以确定喂狗间隔;
- ◆ 开启 `IWDG_IRQn` 中断服务,并使能 IWDT 中断 (`IWDT_CON.IE=1`);
- ◆ 在中断服务中检查中断标志位 (`IWDT_RIS.WDTIF`) 是否被置起;
- ◆ 如果 `IWDT_RIS.WDTIF` 标志位置起,则执行喂狗操作;

13.3.2 软件看门狗

当硬件看门狗禁止时,系统上电看门狗不运行,但可通过软件配置 `IWDT_COM.EN=1` 使能看门狗,俗称软件看门狗。

当软件看门狗使能时,计数器载入 `IWDT_LOAD` 值,并开始递减计数;当计数到 0 时, IWDT 产生中断标志,并在下一个计数时钟到来时,计数器再次载入 `IWDT_LOAD`, 并继续递减计数;当计数器再次计数到 0 时,如果 IWDT 中断标志未被清零, IWDT 将产生复位信号。

`IWDT_INTCLR` 写入值为 `0xFFFFFFFF` 时, IWDT 中断标志位被清零,计数器重载 `IWDT_LOAD` 寄存器值,继续递减计数,写入其他值(非 `0xFFFFFFFF`)时,仅 IWDT 中断标志位被清零,计数器不重载。

操作流程

1. 开启 IWDT 中断服务,并使能 IWDT 中断 (`IWDT_CON.IE=1`);
2. 修改 `IWDT_LOAD.LOAD` 值,以确定喂狗间隔;
3. 选择 IWDT 时钟源 `IWDT_CON.CLKS`;
4. 使能软件看门狗 `IWDT_CON.IE=1`;
5. 在中断服务中检查中断标志位 (`IWDT_RIS.WDTIF`) 是否被置起;
6. 如果 `IWDT_RIS.WDTIF` 标志位置起,则执行喂狗操作;

注：软件看门狗可以通过 IWDT_CON.CLKS 选择时钟源，系统提供了 LRC 和 PCLK 两种时钟源供选择；

13.3.3 调试模式

当系统进入调试模式时，通过 DBG_APB2FZ.IWDT_STOP 配置，可以在 CPU 内核停止时，暂停 IWDT 计数，以便查询系统内部状态，查询完成后，恢复程序执行。

13.3.4 寄存器访问保护

IWDT_LOAD、IWDT_CON、IWDT_INTCLR 寄存器具有写保护功能，对其修改时，必须先向 IWDT_LOCK 写入 0x1ACCE551 移除写保护；当 IWDT_LOCK 写入其他任意值时，寄存器重新被保护。

13.4 特殊功能寄存器

13.4.1 寄存器列表

IWDT 寄存器列表		
名称	偏移地址	描述
IWDT_LOAD	0000 _H	IWDT 计数器装载值寄存器
IWDT_VALUE	0004 _H	IWDT 计数器当前值寄存器
IWDT_CON	0008 _H	IWDT 控制寄存器
IWDT_INTCLR	000C _H	IWDT 中断标志清除寄存器
IWDT_RIS	0010 _H	IWDT 中断标志寄存器
IWDT_LOCK	0100 _H	IWDT 锁定寄存器

13.4.2 寄存器描述

13.4.2.1 IWDT计数器装载值寄存器 (IWDT_LOAD)

IWDT 计数器装载值寄存器 (IWDT_LOAD)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_01000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD																															

LOAD<31:0>	Bit 31-0	W	IWDT 计数器重载值 计数范围 0x0000_0001~0xFFFF_FFFF。如果为 0，IWDT 不计 数。
------------	----------	---	--

13.4.2.2 IWDT计数器当前值寄存器 (IWDT_VALUE)

IWDT 计数器当前值寄存器 (IWDT_VALUE)																															
偏移地址: 04 _H																															
复位值: 11111111_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

VALUE	Bit 31-0	R	IWDT 计数器当前值 读取时返回 IWDT 计数器的当前计数值
-------	----------	---	--

13.4.2.3 IWDT控制寄存器 (IWDT_CON)

IWDT 控制寄存器 (IWDT_CON)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																												CLKS	RSTEN	IE	EN

Reserved	Bit 31-4	—	保留
CLKS	Bit 3	R/W	IWDT 计数时钟选择位 (仅在配置位 CFG_WORD.IWDTEN=0 时有效) 0: PCLK 1: LRC 时钟 (约 32KHz)
RSTEN	Bit 2	R/W	IWDT 复位使能位 (仅在配置位 CFG_WORD.IWDTEN=0 时有效) 0: 禁止 1: 使能, IWDT 计数到 0 时, 产生复位信号, 将芯片复位
IE	Bit 1	R/W	IWDT 中断使能位 (仅在配置位 CFG_WORD.IWDTEN=0 时有效) 0: 禁止 1: 使能, IWDT 计数到 0 时, 产生中断标志
EN	Bit 0	R/W	IWDT 模块使能位 (仅在配置位 CFG_WORD.IWDTEN=0 时有效) 0: 禁止 1: 使能

注: IWDT_CON 寄存器中的各个控制位, 仅在配置字 CFG_WORD 的配置位 IWDTEN=0 时才有效。

13.4.2.4 IWDT中断标志清除寄存器 (IWDT_INTCLR)

IWDT 中断标志清除寄存器 (IWDT_INTCLR)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTCLR																															

INTCLR<31:0>	Bit 31-0	W	IWDT 中断标志清 0 位 0xFFFFFFFF: 写入值为 0xFFFFFFFF, IWDT 中断标志位被清零, 计数器重载 IWDT_LOAD 寄存器值, 继续递减计数 其他值: 写入其他值 (非 0xFFFFFFFF), 仅 IWDT 中断标志位被清零, 计数器不重载
--------------	----------	---	---

13.4.2.5 IWDT中断标志寄存器 (IWDT_RIS)

IWDT 中断标志寄存器 (IWDT_RIS)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															WDTIF

Reserved	Bit 31-1	—	保留
WDTIF	Bit 0	R	IWDT 中断标志位 0: 未产生中断 1: IWDT 计数器计数到 0, 产生中断 写寄存器 IWDT_INTCLR, 可清除 IWDT 中断标志位

13.4.2.6 IWDT锁定寄存器 (IWDT_LOCK)

IWDT 锁定寄存器 (IWDT_LOCK)																															
偏移地址: 100 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															LOCK

Reserved	Bit 31-1	—	保留
LOCK	Bit 0	R/W	IWDT 寄存器保护状态位 0: IWDT 寄存器处于未保护状态 1: IWDT 寄存器处于保护状态 对 IWDT_LOCK 寄存器写入 0x1ACCE551, 被保护的寄存器处于未保护状态; 写入其它值, 处于保护状态

第14章 窗口看门狗 (WWDT)

14.1 概述

窗口看门狗 WWDT 通常用来监视, 由外部干扰或不可预见的逻辑条件造成的应用程序背离运行序列而产生的软件故障。

窗口看门狗 WWDT 对于过早或过晚喂狗都将产生 WWDT 复位, 可用于检测软件没有喂狗或在禁止喂狗区内喂狗行为, 防止程序运行至不可控状态。

14.2 特性

窗口看门狗 WWDT 功能概述:

- ◆ 可编程的递减计数器
- ◆ 支持设定喂狗禁止区
 - ◇ 通过 WWDT_CON.WWDTWIN 设置喂狗禁止区
 - ◇ 在喂狗窗口外喂狗产生 WWDT 复位
 - ◇ 在喂狗窗口内产生 WWDT 中断
 - WWDT 中断可用作喂狗请求
- ◆ 安全可靠
 - ◇ 当配置位 CFG_WORD.WWDTEN 为 1 时, 一旦软件使能, 则只能通过复位关断
- ◆ WWDT 溢出长度可设定
 - ◇ 可通过 WWDT_LOAD.LOAD 寄存器设定喂狗窗口溢出长度
 - ◇ 溢出时产生 WWDT 复位
- ◆ WWDT 中断可用作喂狗请求, 可以唤醒 STOP 模式

14.3 功能描述

14.3.1 窗口看门狗

对于窗口看门狗 WWDT, 过早或过晚喂狗都将产生 WWDT 复位, 可用于检测软件的过晚或过早行为, 防止程序跑至不可控状态, 可通过 WWDT 复位消除。如中断异常, 程序不断进入一个带喂狗指令的子程序的情况。

用户可根据程序正常执行的时间设定喂狗窗口, 可检测到程序未按正常次序执行, 跳过某些程序的异常情况。

窗口看门狗时钟源

窗口看门狗 WWDT 对时间窗口有一定的要求, 系统内提供了 LRC 和 PCLK 作为 WWDT 时钟, 可根据自己喂狗精度的需要, 选择合适的时钟源。

用户可通过寄存器 WWDT_CON.CLKS 进行时钟源选择。

工作流程

- ◆ 系统上电后，窗口看门狗不启动，配置 WWDT_LOAD.LOAD，设置计数初值，配置 WWDT_CON.CLKS 选择时钟源，通过配置 WWDT_CON.EN 使能窗口看门狗；
- ◆ 使能后，WWDT 计数器载入 WWDT_LOAD.LOAD 的 1/4，并开始递减计数，当计数到 0 时，窗口计数器加 1；
- ◆ 在下一个计数时钟到来时，计数器再次载入 WWDT_LOAD.LOAD 寄存器值的 1/4，并继续递减计数；
- ◆ WWDTWIN 设置为 25%，则窗口计数器为 1 时，WWDT 产生中断标志；
- ◆ WWDTWIN 设置为 50%，则窗口计数器为 2 时，WWDT 产生中断标志；
- ◆ WWDTWIN 设置为 75%，则窗口计数器为 3 时，WWDT 产生中断标志；
- ◆ WWDT 产生中断后，直至窗口计数器计数到 4 之前(即累计计数等于 WWDT_LOAD)，如果没有在喂狗窗口内进行喂狗，则 WWDT 模块将产生复位信号，如下图所示；
- ◆ 喂狗窗口内，寄存器 WWDT_INTCLR 写入值为 0xFFFFFFFF 时，WWDT 中断标志位被清零，计数器重载 WWDT_LOAD 寄存器值，继续递减计数，写入其他值（非 0xFFFFFFFF）时，仅 WWDT 中断标志位被清零，计数器不重载。

注：若配置字 CFG_WORD 中的 WWDTEN 位配置为 1，则软件使能窗口看门狗之后，不可再通过软件关闭窗口看门狗。

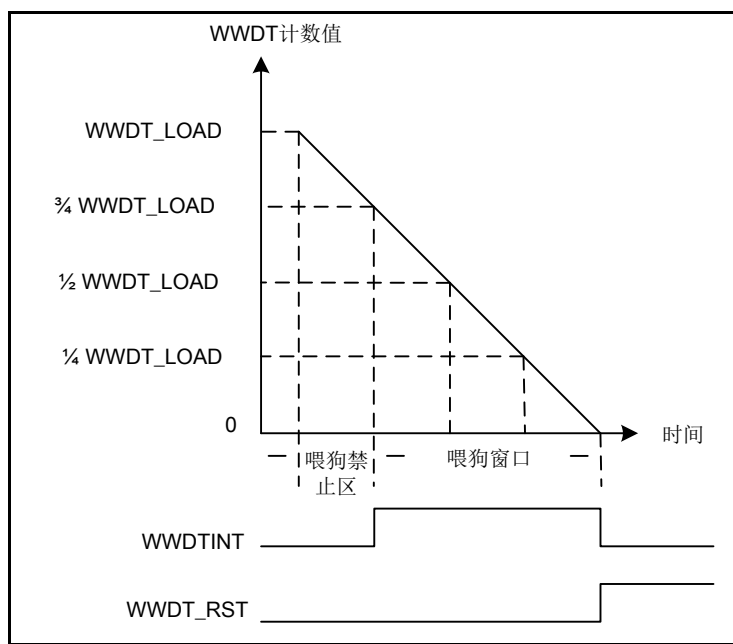


图 14-1 窗口看门狗中断和溢出复位产生时序图 (WWDTWIN 设定为 25%)

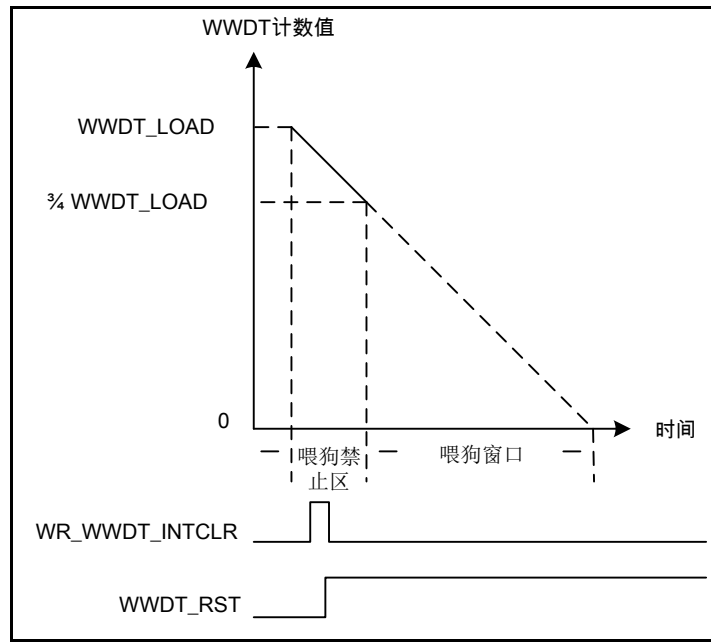


图 14-2 错误的喂狗时序图 (WWDTWIN 设定为 25%)

14.3.2 调试模式

当系统进入调试模式时，通过 `DBG_APB2FZ.WWDT_STOP` 调试暂停选择位配置，可以在 CPU 内核停止时，暂停 WWDT 计数，以便查询系统内部状态，查询完成后，恢复程序执行。

14.3.3 寄存器访问保护

WWDT_LOAD、WWDT_CON、WWDT_INTCLR 寄存器具有写保护功能，对其修改时，必须先向 WWDT_LOCK 写入 0x1ACCE551 移除写保护；当 WWDT_LOCK 写入其他任意值时，寄存器重新被保护。

14. 4 特殊功能寄存器

14. 4. 1 寄存器列表

WWDT 寄存器列表		
名称	偏移地址	描述
WWDT_LOAD	0000 _H	WWDT 计数器装载值寄存器
WWDT_VALUE	0004 _H	WWDT 计数器当前值寄存器
WWDT_CON	0008 _H	WWDT 控制寄存器
WWDT_INTCLR	000C _H	WWDT 中断标志清除寄存器
WWDT_RIS	0010 _H	WWDT 中断标志寄存器
WWDT_LOCK	0100 _H	WWDT 锁定寄存器

14.4.2 寄存器描述

14.4.2.1 WWDT计数器装载值寄存器 (WWDT_LOAD)

WWDT 计数器装载值寄存器 (WWDT_LOAD)																															
偏移地址: 00 _H																															
复位值: 00000000_00000010_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD																															

LOAD	Bit 31-0	W	WWDT 计数器重载值 计数范围 0x0000_0001~0xFFFF_FFFF。如果为 0, WWDT 不计数。
------	----------	---	---

14.4.2.2 WWDT计数器当前值寄存器 (WWDT_VALUE)

WWDT 计数器当前值寄存器 (WWDT_VALUE)																															
偏移地址: 04 _H																															
复位值: 00111111_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

VALUE	Bit 31-0	R	WWDT 计数器当前值 读取时返回 WWDT 计数器的当前计数值, 其中高两位为窗口计数器当前值
-------	----------	---	--

14.4.2.3 WWDT控制寄存器 (WWDT_CON)

WWDT 控制寄存器 (WWDT_CON)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																									WWDTWIN	CLKS	RSTEN	IE	EN		

Reserved	Bit 31-6	—	保留
WWDTWIN	Bit 5-4	R/W	WWDT 禁止喂狗窗口选择位 00: 25%窗口内禁止喂狗, 在禁止窗口内喂狗产生复位 01: 50%窗口内禁止喂狗, 在禁止窗口内喂狗产生复位 10: 75%窗口内禁止喂狗, 在禁止窗口内喂狗产生复位 11: 保留
CLKS	Bit 3	R/W	WWDT 计数时钟选择位 0: PCLK 1: LRC 时钟 (约 32KHz)
RSTEN	Bit 2	R/W	WWDT 复位使能位 0: 禁止 1: 使能, WWDT 计数到 0 时, 产生复位信号, 将芯片复位
IE	Bit 1	R/W	WWDT 中断使能位 0: 禁止 1: 使能, WWDT 计数到 0 时, 产生中断标志
EN	Bit 0	R/W	WWDT 模块使能位 0: 禁止 1: 使能

14.4.2.4 WWDT中断标志清除寄存器 (WWDT_INTCLR)

WWDT 中断标志清除寄存器 (WWDT_INTCLR)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTCLR																															

INTCLR	Bit 31-0	W	WWDT 中断标志清 0 位 0xFFFFFFFF: 写入值为 0xFFFFFFFF, WWDT 中断标志位被清零, 计数器重载 WWDT_LOAD 寄存器值, 继续递减计数 其他值: 写入其他值 (非 0xFFFFFFFF), 仅 WWDT 中断标志位被清零, 计数器不重载。

14.4.2.5 WWDT中断标志寄存器 (WWDT_RIS)

WWDT 中断标志寄存器 (WWDT_RIS)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															WWDTIF

Reserved	Bit 31-1	—	保留
WWDTIF	Bit 0	R	WWDT 中断标志位 0: 未产生中断 1: WWDT 计数器计数到 0, 产生中断 写寄存器 WWDT_INTCLR, 可清除 WWDT 中断标志位

14. 4. 2. 6 WWDT锁定寄存器 (WWDT_LOCK)

WWDT 锁定寄存器 (WWDT_LOCK)																															
偏移地址: 100 _H																															
复位值: 00000000_00000000_00000000_00000001 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															LOCK

Reserved	Bit 31-1	—	保留
LOCK	Bit 0	R/W	WWDT 寄存器保护状态位 0: WWDT 寄存器处于未保护状态 1: WWDT 寄存器处于保护状态 对 WWDT_LOCK 寄存器写入 0x1ACCE551, 被保护的寄存器处于未保护状态; 写入其它值, 处于保护状态

第15章 通用IO及端口控制（GPIO）

15.1 概述

每组通用 GPIO 端口包含 16 个独立的引脚。这些引脚可单独配置为输入或输出，每个引脚输出输入功能中可配置为开漏输出、推挽输出以及浮空输入、带滤波输入模式，配置为输出模式时还可选择每个引脚的驱动强度。

GPIO 引脚可复用为外设功能端口，例如 PWM 输出口、UART 通信口或模拟输入，每个外设均支持复用到多个引脚上。GPIO 端口支持最多 16 个异步外部中断，可被配置到任何一个 IO 引脚上。

15.2 特性

- ◆ 可配置为输入或输出
- ◆ 输出模式可配置
 - ◇ 推挽/开漏
 - ◇ 上拉/下拉
- ◆ 输入模式
 - ◇ 端口浮空
 - ◇ 上拉/下拉
 - ◇ 模拟端口
- ◆ 支持端口输出数据的复位、置位或取反，可按位操作
- ◆ 支持复用为外设功能端口
- ◆ 输出驱动能力可配置：选择普通模式或强驱模式
- ◆ 支持 16 个外部输入中断
- ◆ 支持端口配置写保护功能

15.3 结构框图

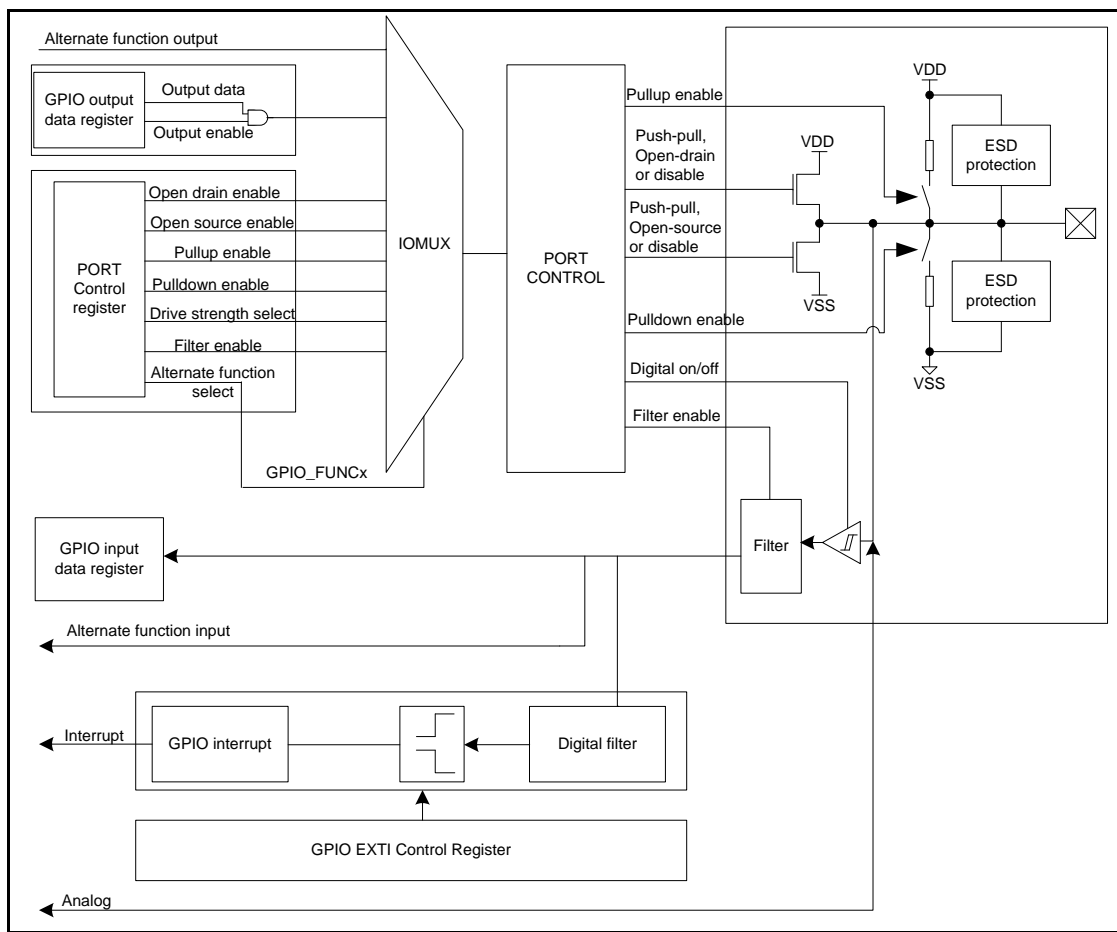


图 15-1 GPIO 结构框图

15.4 功能描述

15.4.1 端口控制寄存器

每组 GPIO 有 16 个相对独立的引脚，每个引脚都可以通过寄存器自由配置。

通过配置 **GPIO_MODE**，可选择相应端口的模式，可配置为输入模式，输出模式和端口关闭模式。选择输出模式时，端口输入同样有效。选择端口关闭模式时，相应端口可被复用为模拟功能。

通过配置 **GPIO_PUPD**，可使能相应端口的上拉或/和下拉电阻。

通过配置 **GPIO_OD**，可将相应端口输出方式配置为推挽或漏极开路两种模式。在配置为漏极开路模式时，可使能相应端口的上拉或下拉，以保证正常输出，也可在端口外部连接上拉或下拉电阻。

通过配置 **GPIO_ODRV**，可选择相应端口的输出驱动能力，以满足不同的负载要求。

通过配置 **GPIO_FLT**，可使能相应端口的输入滤波功能，该滤波器为模拟滤波器，可滤除外部引线上高频信号干扰或毛刺。若输入需要较高的实时性，建议关闭输入滤波功能。

通过配置 GPIO_TYPE，可选择相应端口的输入类型，可选择 TTL 或 SMIT 两种模式。

通过配置 GPIO_FUNC，可选择相应 GPIO 的复用功能，可选择 FUNC_ALT0 ~ FUNC_ALT7。复用功能 GPIO 为输入时必须配置为输入模式，且需外部驱动；复用功能 GPIO 为输出时必须配置为输出模式，推挽或开漏；复用功能 GPIO 为双向模式时，端口必须配置为输出模式，推挽或开漏。

通过配置 GPIO_LOCK，可锁定相应端口的控制寄存器数值。直到下一次 CPU 复位锁定才可被解除。端口数据寄存器不受锁定的控制。

15.4.2 端口数据寄存器

软件可通过读取 GPIO_DIN 才获知端口的电平状态，若相应端口输入滤波被使能，则读到的是端口滤波之后的状态。

通过配置 GPIO_DOUT，可选择端口输出电平值，若端口模式已配置为输出，则该值所对应的电平会在管脚上立即生效。

通过配置 GPIO_BSRR，可按位改写端口输出电平值。对置位寄存器某些位进行写入 1 可置位相应端口，写入 0 的位不会影响相应端口的输出电平。对复位寄存器某些位进行写入 1 可复位相应端口，写入 0 的位不会影响相应端口的输出电平。若同时将某位置位和复位，则置位的优先级更高。

通过配置 GPIO_BIR，可按位翻转端口输出电平值。对翻转寄存器某些位进行写入 1 可将相应端口电平值翻转，写入 0 的位不会影响相应端口的输出电平。

15.4.3 外部端口中断

通过配置 GPIO_EXTIRER，可设置外部中断上升沿触发使能或禁止，其中低 16bit 为上升沿中断触发使能位，高 16bit 为保留位。

通过配置 GPIO_EXTIFER，可设置外部中断下降沿触发使能或禁止，其中低 16bit 为下降沿中断触发使能位，高 16bit 为保留位。

通过配置 GPIO_EXTIEN，可设置外部中断使能或禁止，其中低 16bit 为中断使能设置位，外部中断对应 bit 位配置为 1 表示中断使能，配置为 0 表示中断禁止，高 16bit 为保留位。

通过 GPIO_EXTIFLAG 寄存器，可检测当前有效中断。外部中断对应 bit 位为 1 表示检测到有效中断，为 0 表示未检测到有效中断。该寄存器为只读。

通过配置 GPIO_EXTISFR，可将外部端口中断标志位置位。其中低 16bit 为中断标志位置位，为 1 表示置位中断标志位，为 0 无操作，高 16bit 为保留位。

通过配置 GPIO_EXTICFR，可将外部端口中断标志位清除。其中低 16bit 为中断标志位清除位，为 1 表示清零中断标志位，为 0 无操作，高 16bit 为保留位。对外部端口中断标志位清除时，需先通过配置寄存器 GPIO_EXTIRER（上升沿触发）或 GPIO_EXTIFER（下降沿触发）禁止外部中断触发，来清除中断源，然后再配置寄存器 GPIO_EXTICFR 清除中断标志位，否则有可能导致中断标志位无法被清除。

通过配置 GPIO_EXTIPSR0 和 GPIO_EXTIPSR1 可选择外部中断的 GPIO 端口，每个外部中断可选择 8 个不同的 GPIO 口（PA_n~PH_n）。

通过配置 GPIO_EXTIFLTCR，可设置外部中断滤波参数。GPIO_EXTIFLTCR.FLTEN 位可使能相应外部端口中断滤波功能，配置 GPIO_EXTIFLTCR.FLTSEL 位可设置滤波时间。

15.4.4 通用GPIO配置

每个 GPIO 都可以由软件设置为输出模式或输入模式，也可以复用为外设功能接口。所有 GPIO 端口都有内部上拉或下拉，应用中根据实际需求可以使能或断开。

端口配置表				
配置模式		MODEn<1:0>	PUPDn<1:0>	ODn<1:0>
输出	推挽输出	1x	xx	0x
	开漏输出	1x	xx	1x
输入	浮空输入	01	00	xx
	上拉输入	01	01	xx
	下拉输入	01	10	xx
	模拟输入	01	00	xx

表 15-1 端口配置表

在输出模式中，还可以软件方式设置每个 GPIO 的驱动能力。输入模式中，可以软件方式设置每个 GPIO 的滤波特性。还可以软件选择 GPIO 管脚类型，有 CMOS 和 TTL 两种选择。

ODRVn[1:0]	意义
00	普通电流驱动
01	强电流驱动
10	预留
11	预留

表 15-2 端口驱动表

15.4.5 外部中断与唤醒

每个 GPIO 均可以配置为外部中断源，配置为外部中断的 GPIO 必须设置为输入模式。

每个外部中断通道可以独立的配置输入类型和对应的触发事件（上升沿触发、下降沿触发或双边触发），都可以独立的设置触发或禁止。

EXTI 主要特性如下：

- ◇ 每个外部中断都有独立的触发或禁止设置
- ◇ 每个中断通道有独立的状态标识位
- ◇ 支持多达 16 个外部中断请求
- ◇ 独立设置外部中断滤波特性

外部中断映像如下：

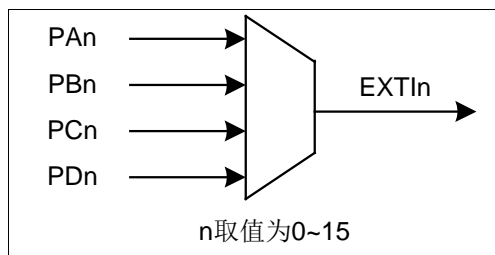


图 15-2 外部中断 GPIO 映像

应用说明:

要产生中断，必须配置好中断端口并使能，然后根据需要通过触发事件寄存器来设置边沿检测，另外根据应用需求可以设置输入滤波。当外部端口发生了预期的边沿时将产生一个中断请求，外部中断标志寄存器中对应的位随之被置 1，此时禁止对应中断触发，并在外部中断标志清零寄存器对应标志位置 1，可以清除中断请求及对应标志位。

外部中断配置步骤如下：

1. 将对应 GPIO 端口配置为输入模式。
2. 配置外部中断端口选择寄存器 0 和外部中断端口选择寄存器 1 选择相应的 GPIO 端口。
3. 配置外部中断上升沿触发使能寄存器设置上升沿触发事件，配置外部中断下降沿触发使能寄存器设置下降沿触发事件。
4. 配置外部中断滤波控制寄存器 GPIO_EXTIFLTR 设置需求对应的滤波特性。
5. 配置外部中断使能寄存器对应的中断使能位，使得外部中断可以有效响应。

15.4.6 外设功能端口复用

使用复用功能时必须先对 GPIO 端口复用功能寄存器 0 或 GPIO 端口复用功能寄存器 1 进行配置，对应的配置参数请查看数据手册管脚功能复用表。

- ◇ 当复用功能为输入时，端口必须配置为输入模式（浮空、上拉或下拉），且输入引脚由外部驱动。
- ◇ 当复用功能为输出时，端口必须配置为输出模式（推挽或开漏）。在配置为开漏模式时，GPIO 可配置为输入功能有效，可以用作双向模式。

如果端口复用功能为输出，则引脚会和片上外设模块的输出信号连接，如果对应外设模块没有被激活，GPIO 的输出信号将不确定。

15.4.7 GPIO 锁定

芯片 GPIO 锁定机制的处理是将端口配置冻结。当一个端口执行了锁定程序后，对应 GPIO 控制寄存器数值将被锁定，在下次复位之前，不能再被更改。

15.4.8 GPIO 输入配置

当 GPIO 端口配置为输入模式时

- ◇ 输出缓冲器被禁止。
- ◇ 根据输入模式配置参数的不同，连接内部上拉、下拉。

- ◇ 输入数据采样到 GPIO 端口输入数据寄存器
- ◇ 访问 GPIO 端口输入数据寄存器即可得到 GPIOn 状态数据

15.4.9 GPIO输出配置

当 GPIO 端口配置为输出模式时：

- ◇ 输出缓冲器被激活
- ◇ 根据配置参数的不同，连接内部上拉、下拉
- ◇ 输入数据采样到 GPIO 端口输入数据寄存器
- ◇ 在开漏模式下，访问输入数据寄存器得到当前 GPIOn 的状态数据。
- ◇ 在推挽模式下，访问输出数据寄存器得到最后一次写的值。

15.4.10 模拟输入配置

当 GPIO 端口配置为模拟输入模式时：

- ◇ 输出缓冲器被禁止
- ◇ 施密特输入触发禁止
- ◇ 内部上拉或下拉禁止
- ◇ 访问输入数据寄存器得到的数据为全 0

15.5 特殊功能寄存器

15.5.1 寄存器列表

GPIO 寄存器列表		
名称	偏移地址	描述
基地址:		
GPIOA_BASE (4008_4000 _H)		
GPIOB_BASE (4008_4040 _H)		
GPIOC_BASE (4008_4080 _H)		
GPIOD_BASE (4008_40C0 _H)		
GPIO_DIN	000 _H	GPIO 端口输入数据寄存器
GPIO_DOUT	004 _H	GPIO 端口输出数据寄存器
GPIO_BSRR	008 _H	GPIO 端口置位和复位寄存器
GPIO_BIR	00C _H	GPIO 端口翻转寄存器
GPIO_MODE	010 _H	GPIO 端口模式寄存器
GPIO_OD	014 _H	GPIO 端口开漏寄存器
GPIO_PUPD	018 _H	GPIO 端口上拉和下拉寄存器
GPIO_ODRV	01C _H	GPIO 端口输出驱动寄存器
GPIO_FLT	020 _H	GPIO 端口滤波寄存器
GPIO_TYPE	024 _H	GPIO 端口类型寄存器
GPIO_FUNC0	028 _H	GPIO 端口复用功能寄存器 0
GPIO_FUNC1	02C _H	GPIO 端口复用功能寄存器 1
GPIO_LOCK	030 _H	GPIO 端口锁定寄存器
基地址: EXTI_BASE (4008_4300 _H)		
GPIO_EXTIRER	000 _H	外部中断上升沿触发使能寄存器
GPIO_EXTIFER	008 _H	外部中断下降沿触发使能寄存器
GPIO_EXTIEN	010 _H	外部中断使能寄存器
GPIO_EXTIFLAG	018 _H	外部中断标志寄存器
GPIO_EXTISFR	020 _H	外部中断标志置位寄存器
GPIO_EXTICFR	028 _H	外部中断标志清零寄存器
GPIO_EXTIPSR0	030 _H	外部中断端口选择寄存器 0
GPIO_EXTIPSR1	034 _H	外部中断端口选择寄存器 1
GPIO_EXTIFLTCR	040 _H	外部中断滤波控制寄存器

15.5.2 寄存器描述

15.5.2.1 GPIO端口输入数据寄存器 (GPIO_DIN)

GPIO 端口输入数据寄存器 (GPIO_DIN)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0

Reserved	Bit 31-16	—	保留
DIN<y>	Bit 15-0	R	GPIO 输入数据

15.5.2.2 GPIO端口输出数据寄存器 (GPIO_DOUT)

GPIO 端口输出数据寄存器 (GPIO_DOUT)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DOUT15	DOUT14	DOUT13	DOUT12	DOUT11	DOUT10	DOUT9	DOUT8	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0

Reserved	Bit 31-16	—	保留
DOUT<y>	Bit 15-0	R/W	GPIO 输出数据

15.5.2.3 GPIO端口置位和复位寄存器 (GPIO_BSRR)

GPIO 端口置位和复位寄存器 (GPIO_BSRR)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR15	BRR14	BRR13	BRR12	BRR11	BRR10	BRR9	BRR8	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	BSR15	BSR14	BSR13	BSR12	BSR11	BSR10	BSR9	BSR8	BSR7	BSR6	BSR5	BSR4	BSR3	BSR2	BSR1	BSR0

BRR<y>	Bit 31-16	W	GPIO 复位控制位 0: 无操作 1: 相应位复位 注: 如果同时对 GPIO 置位和复位, 置位的优先级更高
BSR<y>	Bit 15-0	W	GPIO 置位控制位 0: 无操作 1: 相应位置位

15.5.2.4 GPIO端口翻转寄存器 (GPIO_BIR)

GPIO 端口翻转寄存器 (GPIO_BIR)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																BIR15	BIR14	BIR13	BIR12	BIR11	BIR10	BIR9	BIR8	BIR7	BIR6	BIR5	BIR4	BIR3	BIR2	BIR1	BIR0

Reserved	Bit 31-16	—	保留
BIR<y>	Bit 15-0	W	GPIO 翻转控制位 0: 无操作 1: 相应位翻转

15.5.2.5 GPIO端口模式寄存器 (GPIO_MODE)

GPIO 端口模式寄存器 (GPIO_MODE)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE15																															
	MODE14																														
		MODE13																													
			MODE12																												
				MODE11																											
					MODE10																										
						MODE9																									
							MODE8																								
								MODE7																							
									MODE6																						
										MODE5																					
											MODE4																				
												MODE3																			
													MODE2																		
														MODE1																	
															MODE0																

MODE<y>	Bit 31-0	R/W	GPIO 端口模式控制位 00: 输入、输出同时禁止 01: 输入使能, 输出禁止 1x: 输入、输出同时使能
---------	----------	-----	---

15.5.2.6 GPIO端口开漏寄存器 (GPIO_OD)

GPIO 端口开漏寄存器 (GPIO_OD)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15																															
	OD14																														
		OD13																													
			OD12																												
				OD11																											
					OD10																										
						OD9																									
							OD8																								
								OD7																							
									OD6																						
										OD5																					
											OD4																				
												OD3																			
													OD2																		
														OD1																	
															OD0																

OD<y>	Bit 31-0	R/W	GPIO 开漏控制位 0x: 推挽输出 1x: 开漏输出
-------	----------	-----	---

15.5.2.7 GPIO端口上拉和下拉寄存器 (GPIO_PUPD)

GPIO 端口上拉和下拉寄存器 (GPIO_PUPD)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD15		PUPD14		PUPD13		PUPD12		PUPD11		PUPD10		PUPD9		PUPD8		PUPD7		PUPD6		PUPD5		PUPD4		PUPD3		PUPD2		PUPD1		PUPD0	

PUPD<y>	Bit 31-0	R/W	GPIO 上拉和下拉控制位 00: 无上拉和下拉 01: 上拉 10: 下拉 11: 同时上拉和下拉
---------	----------	-----	---

15.5.2.8 GPIO端口输出驱动寄存器 (GPIO_ODRV)

GPIO 端口输出驱动寄存器 (GPIO_ODRV)																															
偏移地址: 1C _H																															
复位值: 10101010_10101010_10101010_10101010 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODRV15		ODRV14		ODRV13		ODRV12		ODRV11		ODRV10		ODRV9		ODRV8		ODRV7		ODRV6		ODRV5		ODRV4		ODRV3		ODRV2		ODRV1		ODRV0	

ODRV<y>	Bits 31-0	R/W	GPIO 输出驱动控制位 00: 普通电流驱动 01: 强电流驱动 10, 11: 预留
---------	-----------	-----	--

15.5.2.9 GPIO端口滤波寄存器 (GPIO_FLT)

GPIO 端口滤波寄存器 (GPIO_FLT)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																FLT15	FLT14	FLT13	FLT12	FLT11	FLT10	FLT9	FLT8	FLT7	FLT6	FLT5	FLT4	FLT3	FLT2	FLT1	FLT0

Reserved	Bit 31-16	—	保留
FLT<y>	Bit 15-0	R/W	GPIO 滤波控制位 0: 输入滤波禁止 1: 输入滤波使能 注: 该 GPIO 输入滤波器为模拟滤波器, 滤波时间约 20ns

15.5.2.10 GPIO端口类型寄存器 (GPIO_TYPE)

GPIO 端口类型寄存器 (GPIO_TYPE)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																TYPE15	TYPE14	TYPE13	TYPE12	TYPE11	TYPE10	TYPE9	TYPE8	TYPE7	TYPE6	TYPE5	TYPE4	TYPE3	TYPE2	TYPE1	TYPE0

Reserved	Bit 31-16	—	保留
TYPE<y>	Bit 15-0	R/W	GPIO 类型选择位 0: CMOS 1: TTL

15.5.2.11 GPIO端口复用功能寄存器 0 (GPIO_FUNC0)

GPIO 端口复用功能寄存器 0 (GPIO_FUNC0)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSEL_IO7				FSEL_IO6				FSEL_IO5				FSEL_IO4				FSEL_IO3				FSEL_IO2				FSEL_IO1				FSEL_IO0			

FSEL_IO7	Bit 31-28	R/W	GPIO<7>功能复用选择位 请查看管脚功能复用表
FSEL_IO6	Bit 27-24	R/W	GPIO<6>功能复用选择位 请查看管脚功能复用表
FSEL_IO5	Bit 23-20	R/W	GPIO<5>功能复用选择位 请查看管脚功能复用表
FSEL_IO4	Bit 19-16	R/W	GPIO<4>功能复用选择位 请查看管脚功能复用表
FSEL_IO3	Bit 15-12	R/W	GPIO<3>功能复用选择位 请查看管脚功能复用表
FSEL_IO2	Bit 11-8	R/W	GPIO<2>功能复用选择位 请查看管脚功能复用表
FSEL_IO1	Bit 7-4	R/W	GPIO<1>功能复用选择位 请查看管脚功能复用表
FSEL_IO0	Bit 3-0	R/W	GPIO<0>功能复用选择位 请查看管脚功能复用表

15.5.2.12 GPIO端口复用功能寄存器 1 (GPIO_FUNC1)

GPIO 端口复用功能寄存器 1 (GPIO_FUNC1)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSEL_IO15				FSEL_IO14				FSEL_IO13				FSEL_IO12				FSEL_IO11				FSEL_IO10				FSEL_IO9				FSEL_IO8			

FSEL_IO15	Bit 31-28	R/W	GPIO<15>功能复用选择位 请查看管脚功能复用表
FSEL_IO14	Bit 27-24	R/W	GPIO<14>功能复用选择位 请查看管脚功能复用表
FSEL_IO13	Bit 23-20	R/W	GPIO<13>功能复用选择位 请查看管脚功能复用表
FSEL_IO12	Bit 19-16	R/W	GPIO<12>功能复用选择位 请查看管脚功能复用表
FSEL_IO11	Bit 15-12	R/W	GPIO<11>功能复用选择位 请查看管脚功能复用表
FSEL_IO10	Bit 11-8	R/W	GPIO<10>功能复用选择位 请查看管脚功能复用表
FSEL_IO9	Bit 7-4	R/W	GPIO<9>功能复用选择位 请查看管脚功能复用表
FSEL_IO8	Bit 3-0	R/W	GPIO<8>功能复用选择位 请查看管脚功能复用表

15.5.2.13 GPIO端口锁定寄存器 (GPIO_LOCK)

GPIO 端口锁定寄存器 (GPIO_LOCK)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0

KEY	Bit 31-16	W	GPIO 锁定寄存器关键码 注: 检测到写入关键码 0x55AA, 才可对 LOCK<y> 进行写操作, 读该位始终为 0
LOCK<y>	Bit 15-0	R/W	GPIO<y>锁定控制位 0: 相应端口未锁定 1: 相应端口锁定 注: 被锁定相应端口只允许改变输出数据, LOCK<y>一旦被置位, 必须等到下一次 CPU 复位才被清除。

15.5.2.14 外部中断上升沿触发使能寄存器 (GPIO_EXTIRER)

外部中断上升沿触发使能寄存器 (GPIO_EXTIRER)																															
偏移地址: 00 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTIRER15	EXTIRER14	EXTIRER13	EXTIRER12	EXTIRER11	EXTIRER10	EXTIRER9	EXTIRER8	EXTIRER7	EXTIRER6	EXTIRER5	EXTIRER4	EXTIRER3	EXTIRER2	EXTIRER1	EXTIRER0

Reserved	Bit 31-16	—	保留
EXTIRER<y>	Bit 15-0	R/W	EXTI<y>上升沿触发使能位 0: 禁止 1: 使能

15.5.2.15 外部中断下降沿触发使能寄存器 (GPIO_EXTIFER)

外部中断下降沿触发使能寄存器 (GPIO_EXTIFER)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTIFER15	EXTIFER14	EXTIFER13	EXTIFER12	EXTIFER11	EXTIFER10	EXTIFER9	EXTIFER8	EXTIFER7	EXTIFER6	EXTIFER5	EXTIFER4	EXTIFER3	EXTIFER2	EXTIFER1	EXTIFER0

Reserved	Bit 31-16	—	保留
EXTIFER<y>	Bit 15-0	R/W	EXTI<y>下降沿触发使能位 0: 禁止 1: 使能

15.5.2.16 外部中断使能寄存器 (GPIO_EXTIEN)

外部中断使能寄存器 (GPIO_EXTIEN)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTIEN15	EXTIEN14	EXTIEN13	EXTIEN12	EXTIEN11	EXTIEN10	EXTIEN9	EXTIEN8	EXTIEN7	EXTIEN6	EXTIEN5	EXTIEN4	EXTIEN3	EXTIEN2	EXTIEN1	EXTIEN0

Reserved	Bit 31-16	—	保留
EXTIEN<y>	Bit 15-0	R/W	EXTI<y>中断使能位 0: 禁止 1: 使能

15.5.2.17 外部中断标志寄存器 (GPIO_EXTIFLAG)

外部中断标志寄存器 (GPIO_EXTIFLAG)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTIFLAG15	EXTIFLAG14	EXTIFLAG13	EXTIFLAG12	EXTIFLAG11	EXTIFLAG10	EXTIFLAG9	EXTIFLAG8	EXTIFLAG7	EXTIFLAG6	EXTIFLAG5	EXTIFLAG4	EXTIFLAG3	EXTIFLAG2	EXTIFLAG1	EXTIFLAG0

Reserved	Bit 31-16	—	保留
EXTIFLAG<y>	Bit 15-0	R	EXTI<y>中断状态位 0: 未检测到有效中断 1: 检测到有效中断

15.5.2.18 外部中断标志置位寄存器 (GPIO_EXTISFR)

外部中断标志置位寄存器 (GPIO_EXTISFR)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTISFR15	EXTISFR14	EXTISFR13	EXTISFR12	EXTISFR11	EXTISFR10	EXTISFR9	EXTISFR8	EXTISFR7	EXTISFR6	EXTISFR5	EXTISFR4	EXTISFR3	EXTISFR2	EXTISFR1	EXTISFR0

Reserved	Bit 31-16	—	保留
EXTISFR<y>	Bit 15-0	W1	EXTI<y>中断标志位置位 0: 无操作 1: 置位中断标志位 注: 对该位写 1 将中断标志置位, 写 0 无效

15.5.2.19 外部中断标志清零寄存器 (GPIO_EXTICFR)

外部中断标志清零寄存器 (GPIO_EXTICFR)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																EXTICFR15	EXTICFR14	EXTICFR13	EXTICFR12	EXTICFR11	EXTICFR10	EXTICFR9	EXTICFR8	EXTICFR7	EXTICFR6	EXTICFR5	EXTICFR4	EXTICFR3	EXTICFR2	EXTICFR1	EXTICFR0

Reserved	Bit 31-16	—	保留
EXTICFR<y>	Bit 15-0	W1	EXTI<y>中断标志位清零 0: 无操作 1: 清零中断标志位 注: 对该位写 1 将中断标志复位, 写 0 无效

15.5.2.20 外部中断端口选择寄存器 0 (GPIO_EXTIPSR0)

外部中断端口选择寄存器 0 (GPIO_EXTIPSR0)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTIS7				EXTIS6				EXTIS5				EXTIS4				EXTIS3				EXTIS2				EXTIS1				EXTIS0			

EXTIS7	Bit 31-28	R/W	EXTI<7>端口选择位 0000: PA7 0001: PB7 0010: PC7 1111: PIS7 其他: 保留
EXTIS6	Bit 27-24	R/W	EXTI<6>端口选择位 0000: PA6 0001: PB6 0010: PC6 1111: PIS6 其他: 保留
EXTIS5	Bit 23-20	R/W	EXTI<5>端口选择位 0000: PA5 0001: PB5 0010: PC5 1111: PIS5 其他: 保留
EXTIS4	Bit 19-16	R/W	EXTI<4>端口选择位 0000: PA4 0001: PB4 0010: PC4 1111: PIS4 其他: 保留
EXTIS3	Bit 15-12	R/W	EXTI<3>端口选择位 0000: PA3 0001: PB3 0010: PC3 1111: PIS3 其他: 保留
EXTIS2	Bit 11-8	R/W	EXTI<2>端口选择位 0000: PA2 0001: PB2

			0010: PC2 0011: PD2 1111: PIS2 其他: 保留
EXTIS1	Bit 7-4	R/W	EXTI<1>端口选择位 0000: PA1 0001: PB1 0010: PC1 0011: PD1 1111: PIS1 其他: 保留
EXTIS0	Bit 3-0	R/W	EXTI<0>端口选择位 0000: PA0 0001: PB0 0010: PC0 0011: PD0 1111: PIS0 其他: 保留

15.5.2.21 外部中断端口选择寄存器 1 (GPIO_EXTIPSR1)

外部中断端口选择寄存器 1 (GPIO_EXTIPSR1)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTIS15				EXTIS14				EXTIS13				EXTIS12				EXTIS11				EXTIS10				EXTIS9				EXTIS8			

EXTIS15	Bit 31-28	R/W	EXTI<15>端口选择位 0000: PA15 0001: PB15 0010: PC15 1111: PIS7 其他: 保留
EXTIS14	Bit 27-24	R/W	EXTI<14>端口选择位 0000: PA14 0001: PB14 0010: PC14 1111: PIS6 其他: 保留
EXTIS13	Bit 23-20	R/W	EXTI<13>端口选择位 0000: PA13 0001: PB13 0010: PC13 1111: PIS5 其他: 保留
EXTIS12	Bit 19-16	R/W	EXTI<12>端口选择位 0000: PA12 0001: PB12 0010: PC12 1111: PIS4 其他: 保留
EXTIS11	Bit 15-12	R/W	EXTI<11>端口选择位 0000: PA11 0001: PB11 0010: PC11 1111: PIS3 其他: 保留
EXTIS10	Bit 11-8	R/W	EXTI<10>端口选择位 0000: PA10 0001: PB10

			0010: PC10 1111: PIS2 其他: 保留
EXTIS9	Bit 7-4	R/W	EXTI<9>端口选择位 0000: PA9 0001: PB9 0010: PC9 1111: PIS1 其他: 保留
EXTIS8	Bit 3-0	R/W	EXTI<8>端口选择位 0000: PA8 0001: PB8 0010: PC8 1111: PIS0 其他: 保留

15.5.2.22 外部中断滤波控制寄存器 (GPIO_EXTIFLTCR)

外部中断滤波控制寄存器 (GPIO_EXTIFLTCR)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FLTSEL								FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0

Reserved	Bit 31-24	—	保留
FLTSEL	Bit 23-16	R/W	<p>EXTI 端口中断去抖滤波时间选择位</p> <p>滤波时间 = (FLTSEL<7:0> + 1) × 2 个 LRC 时钟周期 (因时钟同步而引入的滤波时间偏差约 1 个 LRC 时钟周期)</p> <p>注: 该数字滤波器可滤除脉宽小于上述滤波时间的非连续抖动脉冲, 对连续抖动的群脉冲, 可能无法完全滤除</p>
FLTEN<y>	Bit 15-0	R/W	<p>EXTI<y> 端口中断去抖滤波使能位</p> <p>0: 禁止</p> <p>1: 使能</p> <p>注: 该位需在中断使能之前配置, 使能之后禁止更改</p>

第16章 循环冗余校验 (CRC)

16.1 概述

循环冗余校验 (CRC) 发生器可以执行带可编程多项式设定的 CRC 计算, 用于对数据传输的完整性和正确性进行校验。

16.2 特性

- ◆ 支持四个常用的多项式: CRC-CCITT, CRC-8, CRC-16 和 CRC-32
 - ◇ CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - ◇ CRC-8: $X^8 + X^2 + X + 1$
 - ◇ CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - ◇ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- ◆ 支持可编程的种子值
- ◆ 支持对输入数据和 CRC 校验值的可编程的反序设定
- ◆ 支持对输入数据和 CRC 校验值的可编程的反码设定
- ◆ 支持 8/16/32 位数据宽度
 - ◇ 8-bit 写模式: 1 个 AHB 时钟周期操作
 - ◇ 16-bit 写模式: 2 个 AHB 时钟周期操作
 - ◇ 32-bit 写模式: 4 个 AHB 时钟周期操作
- ◆ 支持使用 DMA 写数据执行 CRC 操作

16.3 结构框图

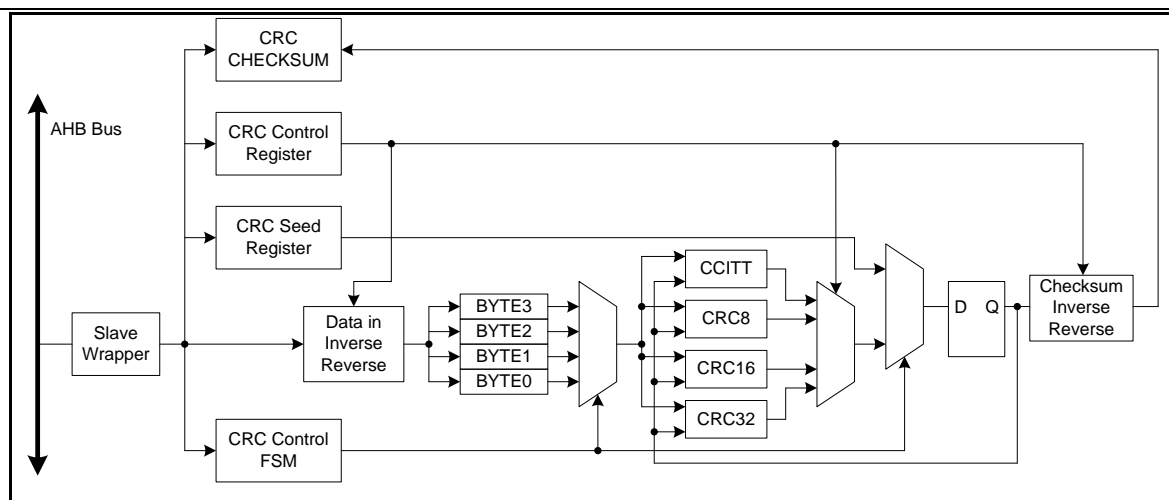


图 16-1 CRC 结构框图

16.4 功能描述

16.4.1 常规操作

CRC 发生器可以执行带可编程多项式设定的 CRC 运算。多项式操作包括 CRC-CCITT, CRC-8, CRC-16 和 CRC-32。用户可以通过设置 MODE 选择 CRC 多项式操作模式。

操作示例:

1. 通过设置 CRC_CR.EN 使能 CRC 发生器
2. CRC 运算初始化设置
 - a. 通过设置 CRC_CR.CHSINV 配置 CRC 校验值反码
 - b. 通过设置 CRC_CR.CHSREV 配置 CRC 校验值位反序
 - c. 通过设置 CRC_CR.DATINV 配置 CRC 写入数据反码
 - d. 通过设置 CRC_CR.DATREV 配置 CRC 写入数据位反序
 - e. 通过设置 CRC_CR.MODE 配置 CRC 校验模式
 - f. 通过设置 CRC_CR.DATLEN 配置 CRC 写入数据长度
3. 通过设置 CRC_CR.RST 执行 CRC 复位, CRC 复位将装载初始种子值到 CRC 运算电路
4. 写数据到 CRC_DATA 寄存器来计算 CRC 校验值
5. 通过读 CRC_CHECKSUM 寄存器来获得 CRC 校验结果

16.4.2 DMA请求

DMA 请求 (在使能后) 仅用于数据传输, 当需要减少 MCU 负荷时可以使用 DMA 功能, 在计算完 CRC 后, DMA 模块将 CRC_CHECKSUM 里面的计算值传输到用户提供的存储区中。

操作示例:

1. 通过设置 CRC_CR.EN 使能 CRC 发生器
2. CRC 运算初始化设置
 - a. 通过设置 CRC_CR.CHSINV 配置 CRC 校验值反码
 - b. 通过设置 CRC_CR.CHSREV 配置 CRC 校验值位反序
 - c. 通过设置 CRC_CR.DATINV 配置 CRC 写入数据反码
 - d. 通过设置 CRC_CR.DATREV 配置 CRC 写入数据位反序
 - e. 通过设置 CRC_CR.MODE 配置 CRC 校验模式
 - f. 通过设置 CRC_CR.DATLEN 配置 CRC 写入数据长度
3. 通过设置 CRC_CR.RST 执行 CRC 复位, CRC 复位将装载初始种子值到 CRC 运算

电路

4. 用户先定义一段特殊类型的存储区，假设此存储区命名为 `DESCR`，把 `DMA_CTRLBASE+x` 的地址赋给此存储区首地址，此存储区数据类型参考 DMA 章节
5. 将 `CRC_DATA` 寄存器的地址填入 `DESCR.DST`，表示每发生一次 CRC 的 DMA 申请事件，数据搬运的目标地址，数据都会从需要计算 CRC 的数据存储区中传到 `CRC_DATA` 寄存器地址下
6. 设置 `DESCR.DSR_INC`，表示目标地址每一次发生 CRC 的 DMA 申请事件后的地址增加量
7. 将用户定义的数据存储区的地址填入 `DESCR.SRC`，表示每发生一次 CRC 的 DMA 申请事件后数据搬运的源地址，数据都会从该存储区中传入 `CRC_DR` 地址下
8. 设置 `DESCR.SRC_INC`，表示源地址每一次发生 CRC 的 DMA 申请事件后的地址增加量
9. 需要传输的总字节数减 1 填入 `DESCR.N_MINUS_1`，则每发生一次 CRC 的 DMA 申请事件后，该值都会递减，注意最大字节数为 1024 个字节
10. 设置 `DMA_CHx_SELCON.MSEL` 选择触发 DMA 的输入源为 CRC
11. 通道的优先级通过设置 `DMA_CHPRSET` 或 `DMA_CHPRCLR` 寄存器来配置
12. 以上步骤都执行完毕后，初始化阶段完成了，需要激活该 DMA 通道，设置 `DMA_CHENSET` 使能用户选择的通道
13. 使能 `CRC_CR.DMAEN` 位。

16.5 特殊功能寄存器

16.5.1 寄存器列表

CRC 寄存器列表		
名称	偏移地址	描述
CRC_CR	0000 _H	CRC 控制寄存器
CRC_DATA	0004 _H	CRC 写数据寄存器
CRC_SEED	0008 _H	CRC 种子寄存器
CRC_CHECKSUM	000C _H	CRC 校验值寄存器

16.5.2 寄存器描述

16.5.2.1 CRC控制寄存器 (CRC_CR)

CRC 控制寄存器 (CRC_CR)																															
偏移地址: 00H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BYTORD	DATLEN	MODE	CHSINV	DATINV	CHSREV	DATREV	Reserved											DMAEN	CWERR	WERR	RST	EN		

Reserved	Bit 31-25	—	保留
BYTORD	Bit 24	R/W	校验值字节顺序选择位 0: 优先校验低字节 1: 优先校验高字节 注: 优先校验低字节即优先校验Bit 7-0; 优先校验高字节, 16-bit模式即优先校验Bit 15-8, 32-bit模式优先校验Bit 31-24; 8-bit模式校验顺序无区别
DATLEN	Bit 23-22	R/W	数据长度选择位 00: 通过写 CRC_DATA 寄存器方式自动判断 01: 数据为 8-bit (CRC_DATA[7:0]有效) 10: 数据为 16-bit (CRC_DATA[15:0]有效) 11: 数据为 32-bit (CRC_DATA[31:0]有效)
MODE	Bit 21-20	R/W	模式选择位 00: CRC-CCITT 多项式模式 01: CRC-8 多项式模式 10: CRC-16 多项式模式 11: CRC-32 多项式模式
CHSINV	Bit 19	R/W	校验值反码使能位 0: 禁止 1: 使能
DATINV	Bit 18	R/W	写数据反码使能位 0: 禁止 1: 使能
CHSREV	Bit 17	R/W	校验值反序使能位 0: 禁止 1: 使能
DATREV	Bit 16	R/W	写数据反序使能位 0: 禁止 1: 使能
Reserved	Bit 15-5	—	保留
DMAEN	Bit 4	R/W	DMA 使能位 0: 禁止

			1: 使能
CWERR	Bit 3	W1	CRC 写数据错误标志清除位 0: 无操作 1: 清除标志位
WERR	Bit 2	R	CRC 写数据错误标志位 0: 无错误 1: 发生写数据错误 注: 写入格式与 DATLEN 所选择不相符时会将标志位置位, 未在最低字节或低半字写数据时也会将标志位置位。
RST	Bit 1	W1	CRC 复位位 0: 无操作 1: 复位 注: 该位复位CRC内部状态机、DMAEN和缓存以及初始化种子值, 但不会复位寄存器值
EN	Bit 0	R/W	CRC 使能位 0: 禁止 1: 使能

16.5.2.2 CRC写数据寄存器 (CRC_DATA)

CRC 写数据寄存器 (CRC_DATA)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

DATA	Bit 31-0	R/W	CRC 写入数据位 用户看可以通过 CPU 或 DMA 直接写数据到该位来执行 CRC 操作。 注 1: 当写数据长度是 8-bit 模式, 该位有效数据为 DATA[7:0], 如果数据长度是 16-bit 模式, 该位有效数据为 DATA[15:0]。如果自动检测数据长度, 则可通过最低字节、低半字或字写入方式任意写入数值。 注 2: 当写入到错误的字节或半字时, 硬件会置位写数据错误标志位 WERR。
------	----------	-----	---

16.5.2.3 CRC种子寄存器 (CRC_SEED)

CRC 种子寄存器 (CRC_SEED)																															
偏移地址: 08 _H																															
复位值: 11111111_11111111_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEED																															

SEED	Bit 31-0	R/W	CRC 种子值 该位表示 CRC 校验种子值
------	----------	-----	----------------------------------

16.5.2.4 CRC校验值寄存器 (CRC_CHECKSUM)

CRC 校验值寄存器 (CRC_CHECKSUM)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHECKSUM																															

CHECKSUM	Bit 31-0	R	CRC 校验值 该位表示 CRC 校验结果
----------	----------	---	---------------------------------

第17章 高级定时器（AD16C4T）

17.1 概述

高级控制定时器（AD16C4T）是一个功能强大、配置灵活的定时器模块，它包含一个 16-bit，定时器，具有定时、计数、脉冲输入信号测量（输入捕获）、产生特定 PWM 波（输出比较）等功能。

17.2 特性

- ◆ 16 位递增，递减，递增/递减自动加载计数器
- ◆ 16 位可编程预分频器，可在定时器运行中对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 带有四个独立通道，每个通道支持以下功能
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出
 - ◇ 单脉冲输出
- ◆ 通道 1~3 支持互补输出，死区时间可配
- ◆ 在给定数目的计数周期之后更新重复计数寄存器
- ◆ 支持刹车功能，刹车后定时器输出状态可控
- ◆ 支持中断/DMA：
 - ◇ 更新事件：计数器上溢/下溢，计数器初始化
 - ◇ 触发事件
 - ◇ 换相事件
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ 刹车输入
- ◆ 支持增量（正交）编码及霍尔电路
- ◆ 通过外设互联（PIS）可支持与片上其他定时器的互联工作

17.3 结构框图

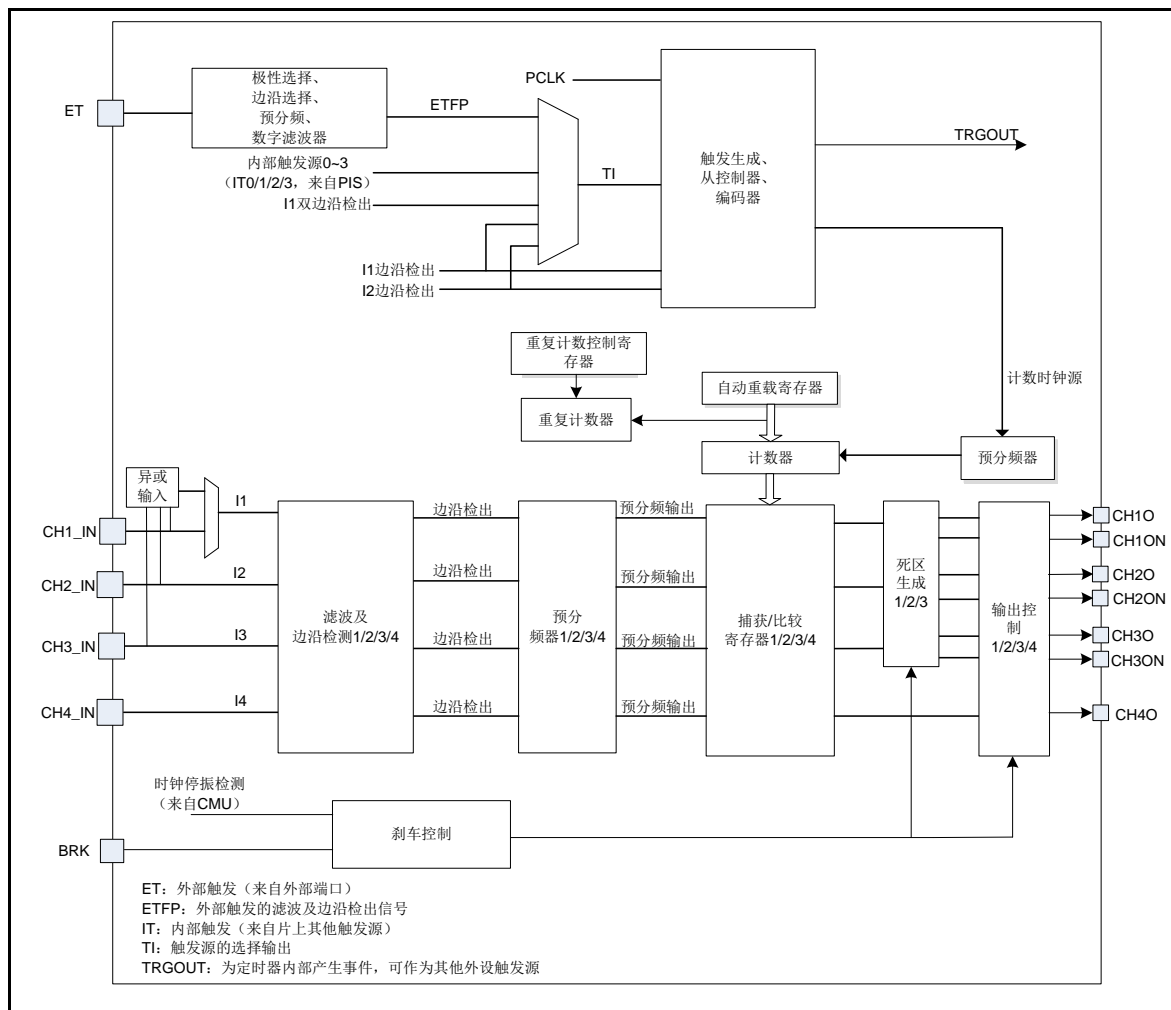


图 17-1 高级定时器电路结构框图

17.4 功能描述

17.4.1 预分频器

定时器包含一个 16-bit 的计数器 (AD16C4T_COUNT)，计数时钟由预分频寄存器 (AD16C4T_PRES) 进行分频。计数周期由自动重载计数器 (AD16C4T_AR) 设定。重复计数寄存器则可指定计数周期数目 (AD16C4T_REPAR)。

自动重载寄存器 (AD16C4T_AR) 是一个可缓存的寄存器。当 AD16C4T_CON1 寄存器的 ARPEN 位复位时，AD16C4T_AR 寄存器重载功能失效，AD16C4T_AR 就是有效寄存器；ARPEN 置位时，AD16C4T_AR 寄存器具有重载功能，产生更新事件 (UEV) 时，加载值 (AD16C4T_AR 寄存器值) 更新到影子寄存器。

当 AD16C4T_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢 (或递减下溢) 时会产生更新事件 (UEV)。同样，软件方式也可产生更新事件。AD16C4T_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 AD16C4T_PRES 寄存器值+1 次分频。由于 AD16C4T_PRES 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

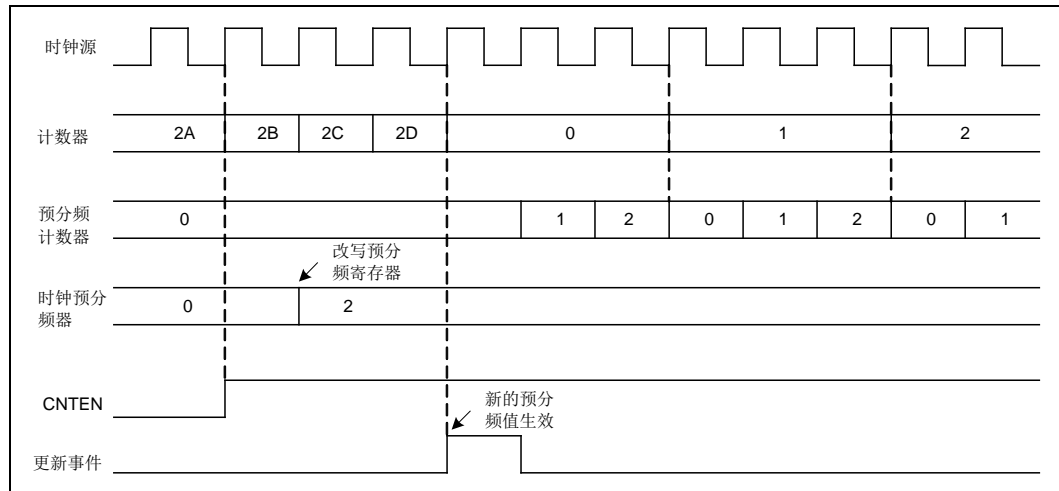


图 17-1 预分频值计数时序图

17.4.2 重复计数器

重复计数器用于控制发生多少次上溢或下溢次后产生更新事件。

重复计数器递减:

- ◇ 递增模式的每次上溢
- ◇ 递减模式的每次下溢
- ◇ 中心对齐模式时，计数器上溢与下溢。中心对齐模式限制了最大重复次数为 128 个 PWM 周期，每个 PWM 周期内可更新两次占空比。

AD16C4T_REPAR 寄存器是一个可缓存寄存器。软件（置位 AD16C4T_SGE 寄存器中的 SGU 位）或硬件从机模式控制方式产生更新事件时，无论重复计数器为何值，AD16C4T_REPAR 寄存器中值会立即更新到重复计数器的影子寄存器中。

中心对齐模式下，AD16C4T_REPAR 中值为奇数时，更新事件是在上溢或下溢时产生，取决于何时写 AD16C4T_REPAR 寄存器及何时开始计数。若在启动计数器前写 AD16C4T_REPAR，则上溢时产生 UEV，反之则在下溢时产生 UEV。

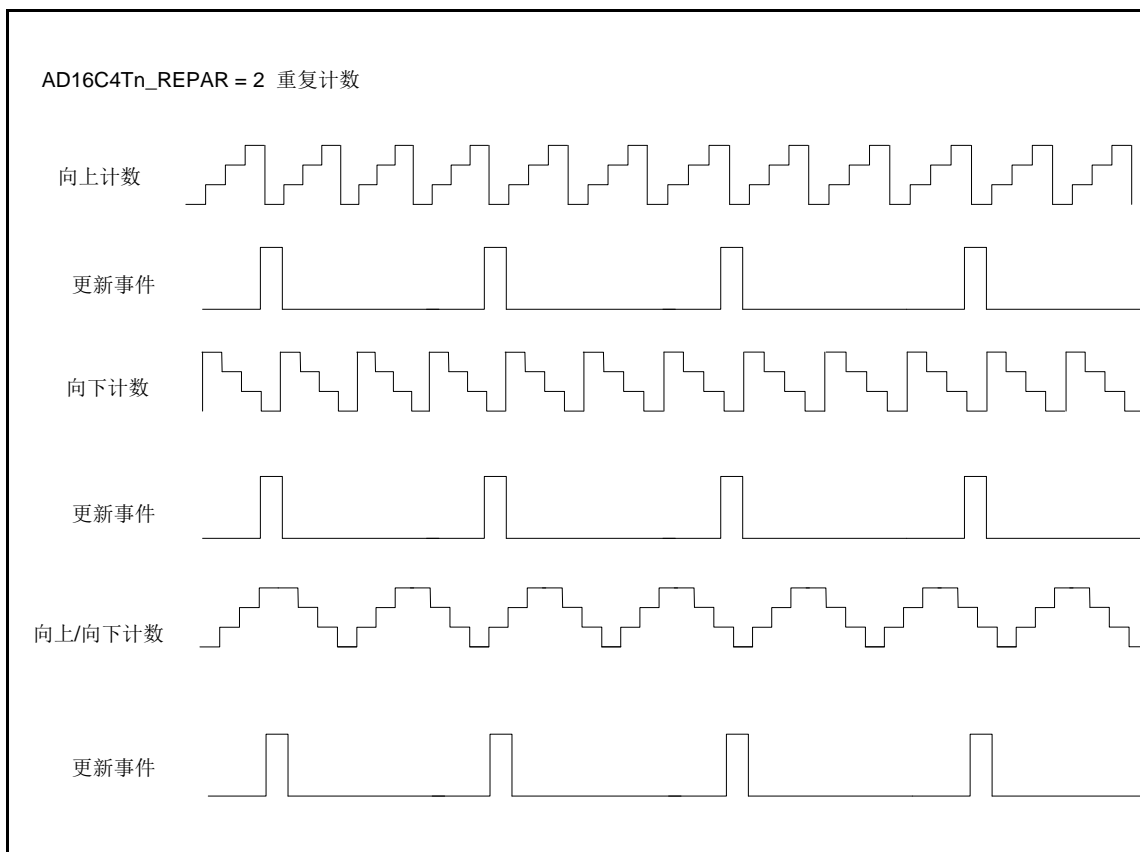


图 17-2 重复计数器工作模式

注意：置位 AD16C4T_SGE 寄存器中的 SGU 位也可以产生更新事件。

17.4.3 时钟源

计数器工作时钟可以选择内部时钟 (INT_CLK)、外部时钟源 1 (I1、I2、I3、I4)、外部时钟源 2 (ET)，内部触发输入 (IT1、IT2、IT3、IT4)

17.4.3.1 内部时钟源 (INT_CLK)

若从模式控制器被关闭 (AD16C4T_SMCON 寄存器的 SMODS= "000"), 则 CNTEN, AD16C4T_CON1.DIRSEL 与 AD16C4T_SGE.SGU 位为实际控制位, 这些位只能软件修改 (SGU 位除外, 仍硬件自动清除)。一旦 CNTEN 位被写为'1', 预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况, 没有分频。

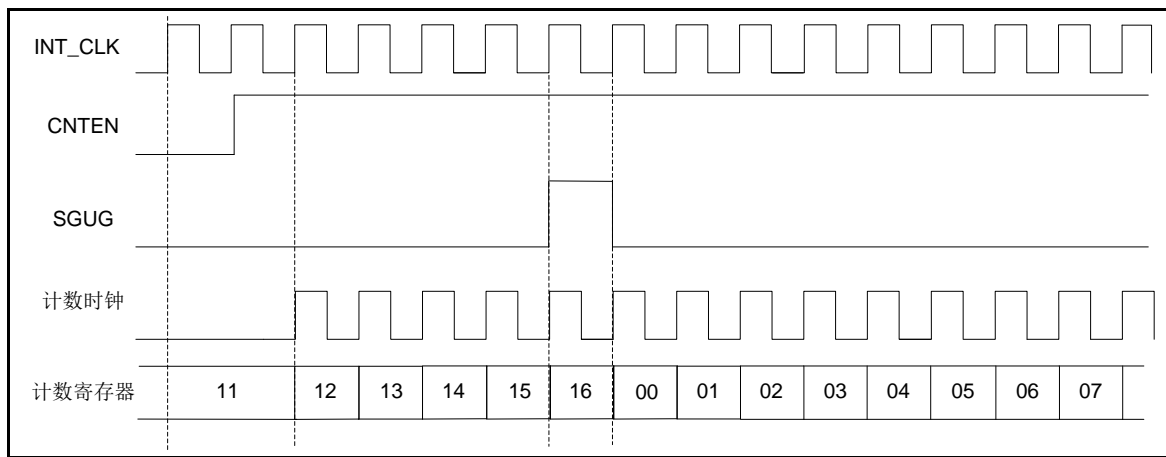


图 17-3 采用内部时钟计数

17.4.3.2 外部时钟源 1

AD16C4T_SMCON 寄存器的 SMODS= "111"时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

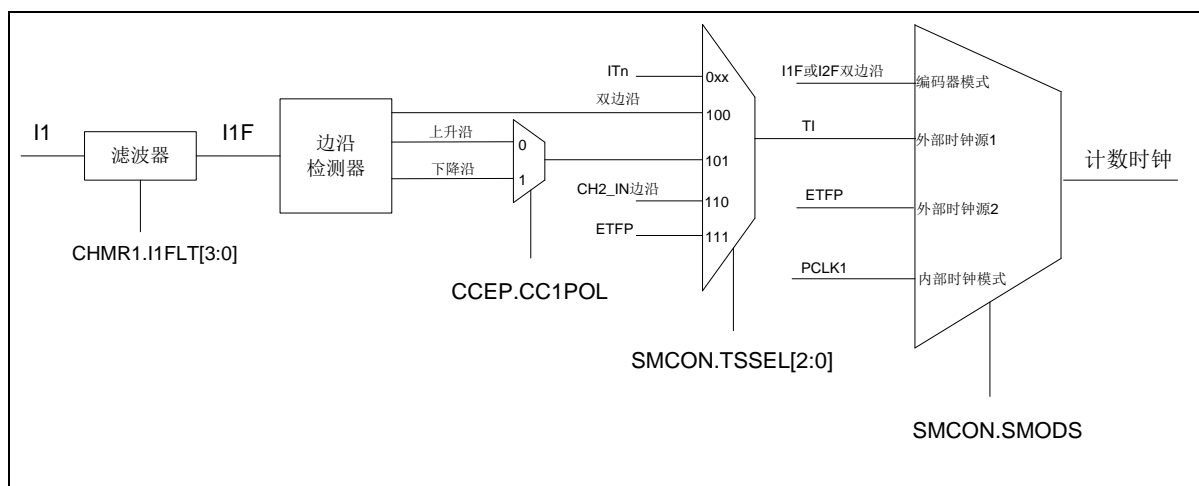


图 17-4 I1 外部时钟连接

配置计数器为外部时钟源 1，步骤如下：

1. AD16C4T_SMCON 寄存器中 SMODS = "111"，配置定时器外部时钟模式 1。
2. 设置 AD16C4T_SMCON 寄存器中的 TSSEL 选择外部时钟源。
3. 如外部时钟源为 I1，可配置 AD16C4T_CHMR1 寄存器 CC1SSEL = "01"，配置通道 1 检测 I1 输入的上升沿；设置 AD16C4T_CCEP 寄存器中 CC1POL = '0'，选择极性位上升沿。
4. 写 AD16C4T_CHMR1 寄存器的 I1FLT[3: 0]位，配置输入滤波器时间（若没有滤波器需求，维持 I1FLT = "0000"）。
5. AD16C4T_CON1 寄存器中 CNTEN = '1'，使能计数器。

当 I1 上出现一次上升沿时，计数器计数一次且 TRGIF 标志位置位。

17.4.3.3 外部时钟源 2

置位 AD16C4T_SMCON 寄存器的 ECM2EN 位选定外部时钟源 2。

计数器可对外部触发输入 ET 进行上升沿或下降沿计数。

下图给出了外部输入输入模块的概况。

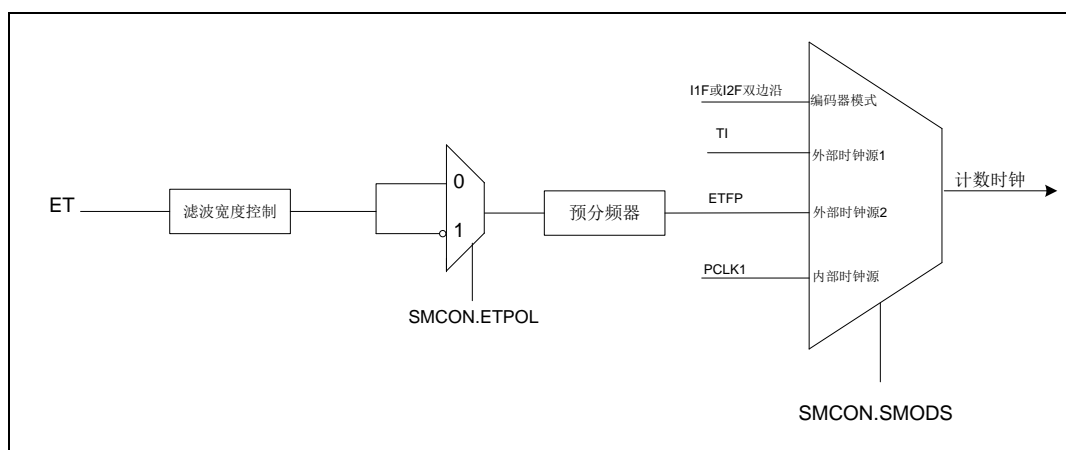


图 17-5 外部触发输入模块

配置计数器为外部时钟源 2，配置过程如下：

1. 设置 AD16C4T_SMCON 寄存器的 ETFLT[3: 0]，配置输入滤波时间。
2. 设置 AD16C4T_SMCON 寄存器中 ETPSEL[1: 0]，设置预分频器。
3. 设置 AD16C4T_SMCON 寄存器中 ETPOL，检测 ET 引脚上升沿或下降沿。
4. 设置 AD16C4T_SMCON 寄存器中 ECM2EN = '1'，使能外部时钟模式 2。
5. 设置 AD16C4T_CON1 寄存器的 CNTEN = '1'，使能计数器。

计数器每两个上升沿计一次数。

17.4.3.4 内部触发输入 (ITn)

当 AD16C4T_SMCON 寄存器的 SMODS= "111", 选定外部时钟模式 1。计数器根据选定的内部输入端的上升或下降沿计数。

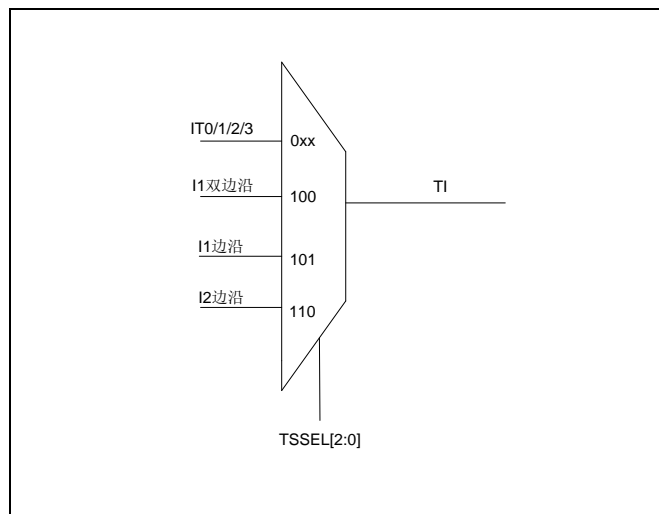


图 17-6 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

1. AD16C4T_SMCON 寄存器中 SMODS = "111", 配置外部时钟模式 1。
2. AD16C4T_SMCON 寄存器的 TSSEL = "0xx", 选定 ITn 作为触发输入源。
3. AD16C4T_CON1 寄存器的 CNTEN = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路。

17.4.4 计数模式

17.4.4.1 递增计数模式

当 AD16C4T_REPAR 寄存器值为 0 时，定时器配置为递增模式，计数器从 0 开始递增，直至 AD16C4T_AR 寄存器值；然后从 0 重新开始计数并产生一个更新事件 (UEV)。当 AD16C4T_REPAR 寄存器不为 0 时，则在 AD16C4T_REPAR+1 次计数后产生更新事件。

当有更新事件 (UEV) 产生时，预装载寄存器会更新到影子寄存器，更新标志位 (AD16C4T_RIF 寄存器中的 UEVTIF 位) 置位 (取决于 UERSEL 位)：

- ◇ 更新 AD16C4T_REPAR 寄存器的值到影子寄存器
- ◇ 更新 AD16C4T_AR 寄存器的值到影子寄存器
- ◇ 更新 AD16C4T_PREP 寄存器的值到影子寄存器

下图为 AD16C4T_AR = 0x16，预分频设为 2 分频时的计数器时序。

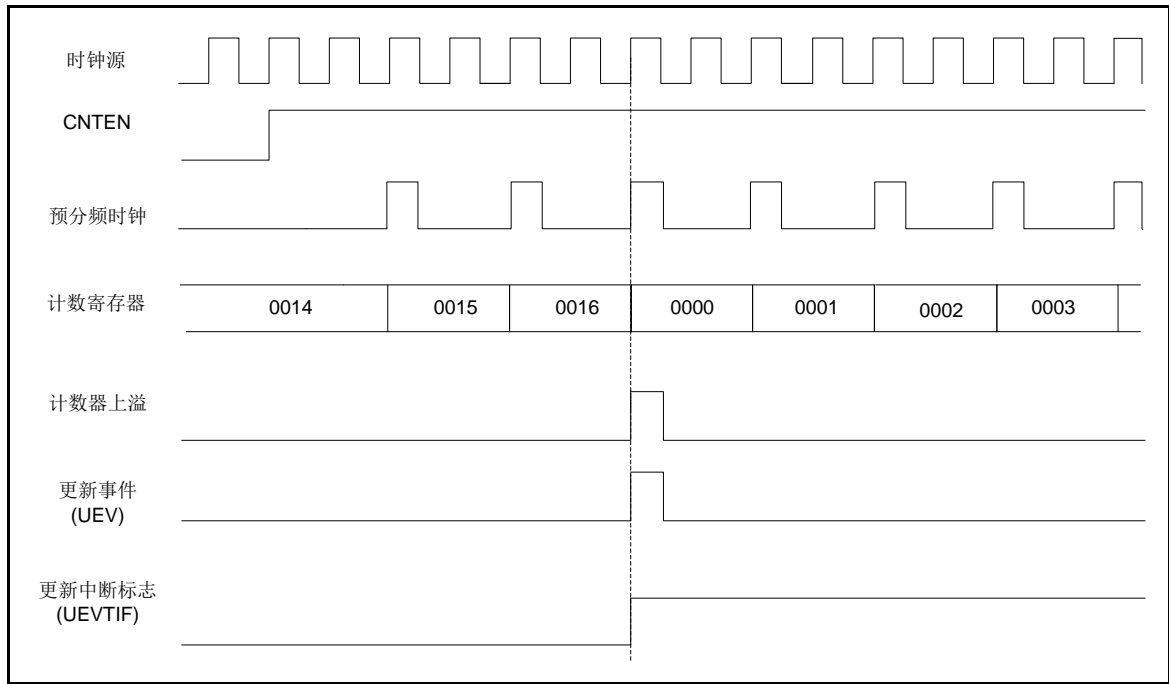


图 17-7 计数器递增计数时序图

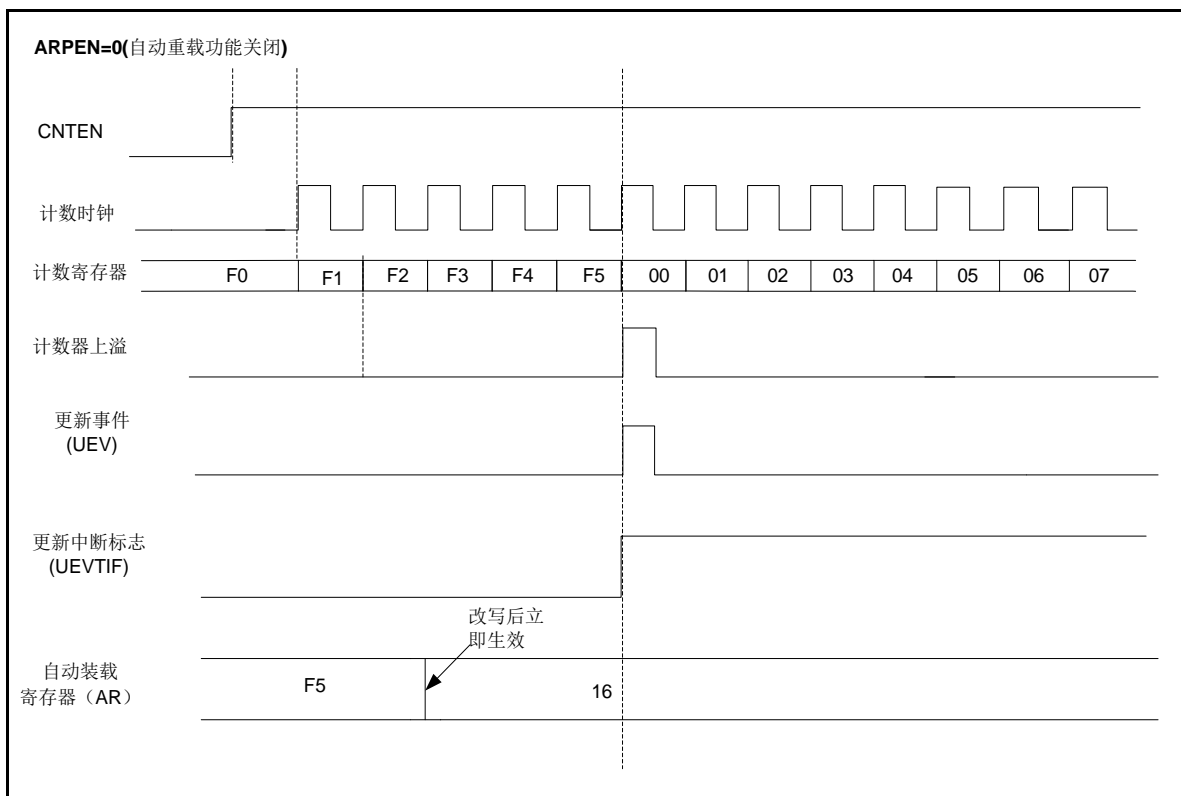


图 17-8 当 ARPEN=0 时计数器时序图

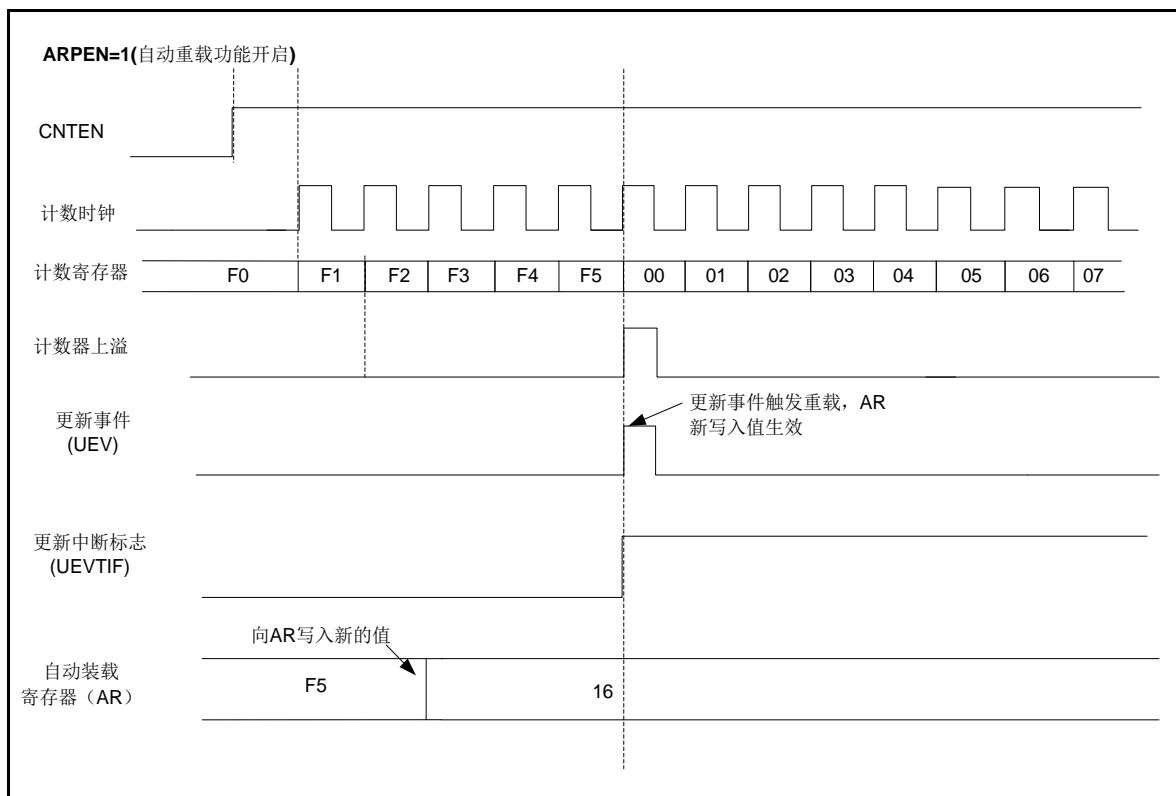


图 17-9 当 ARPEN=1 时计数器时序图

17.4.4.2 递减计数模式

当 AD16C4T_REPAR 寄存器值为 0 时，定时器配置为递减模式，计数器从 AD16C4T_AR 寄存器值开始递减至 0；然后重复递减并产生更新事件 (UEV)。当 AD16C4T_REPAR 寄存器不为 0 时，则在 AD16C4T_REPAR+1 次后产生更新事件。

置位 AD16C4T_SGE 寄存器中的 SGU 位（通过软件或使用从机模式控制器）同样会产生更新事件。

当有更新事件 (UEV) 产生时，预载寄存器值会更新到影子寄存器，更新标志位 (AD16C4T_RIF 寄存器中的 UEVTIF 位) 置位（取决于 UERSEL 位）。

下图为 AD16C4T_AR = 0x27，预分频设为 1 分频时的计数器时序。

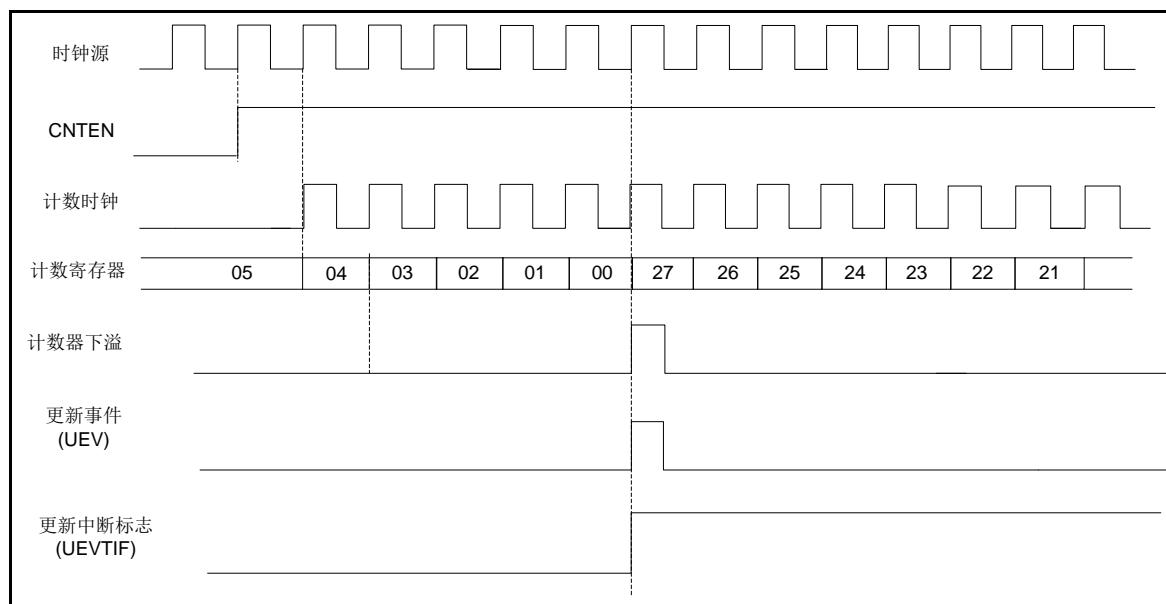


图 17-10 定时器递减计数时序图

17.4.4.3 中心对齐模式

当 AD16C4T_CON1 寄存器的 CMSEL 位的值不等于“00”时,定时器工作在中心对齐模式。定时器配置为中心对齐模式时,计数器先从 0 开始递增至 AD16C4T_AR 寄存器值减 1,并产生更新事件 (UEV);接着计数器从 AD16C4T_AR 寄存器值递减至 1,并产生更新事件。如此循环计数。计数器递减计数 (中心对称模式 1, CMSEL=“01”)、计数器递增计数 (中心对称模式 2, CMSEL=“10”)、计数器递增和递减计数 (中心对称模式 3, CMSEL=“11”),每个通道的输出比较中断标志位都会置位。

在中心对齐模式下,AD16C4T_CON1 寄存器的 DIRSEL 位无法进行写操作,该位由硬件自动更新指示当前计数方向。

计数上溢、下溢或者置位 AD16C4T_EGR 寄存器的 SGU 位 (通过软件或使用从模式控制器)都会产生更新事件。因此,计数器和预分频器都会从 0 开始计数。

软件置位 AD16C4T_CON1 寄存器中的 DISUE 位可关闭更新事件 (UEV) 的产生。更新事件 (UEV) 关闭时,可避免向预载寄存器写新值时更新影子寄存器。DISUE 复位之前都不会产生更新事件。而在正常产生更新事件时,计数器仍然从 0 开始,同样预分频计数也是从 0 开始 (但预分频值没有改变)。此外,若置位 AD16C4T_CON1 寄存器中的 UERSEL 位 (更新请求选择),置位 SGU 位时会产生一次更新事件 (UEV),但 UEVTIF 标志位不会置位 (因此,不会触发中断或 DMA 请求)。这就避免了在捕获事件时,清除计数器值时产生更新和捕获中断。

当有更新事件 (UEV) 产生时,预载寄存器值会更新到影子寄存器,更新标志位 (AD16C4T_RIF 寄存器中的 UEVTIF 位)置位 (取决于 UERSEL 位)。

注:若更新源为计数器上溢,自动重载会在计数器重载前更新。因此,下一周期即为预期值 (计数器载入新值)。

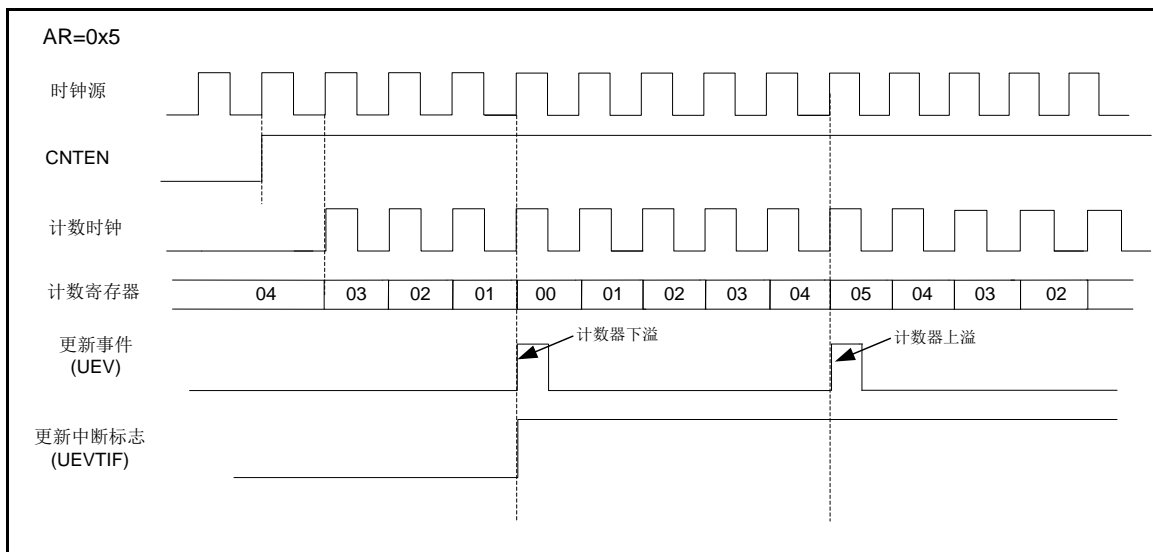


图 17-11 增减计数器时序图

17.4.5 捕获/比较通道

下方 3 张图为捕获/比较通道的概述。

输入电路对 In 输入端的信号进行采样，产生一个经过滤波的信号 InF。之后，一个可极性选择的边沿检测器产生 In 边沿检出信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

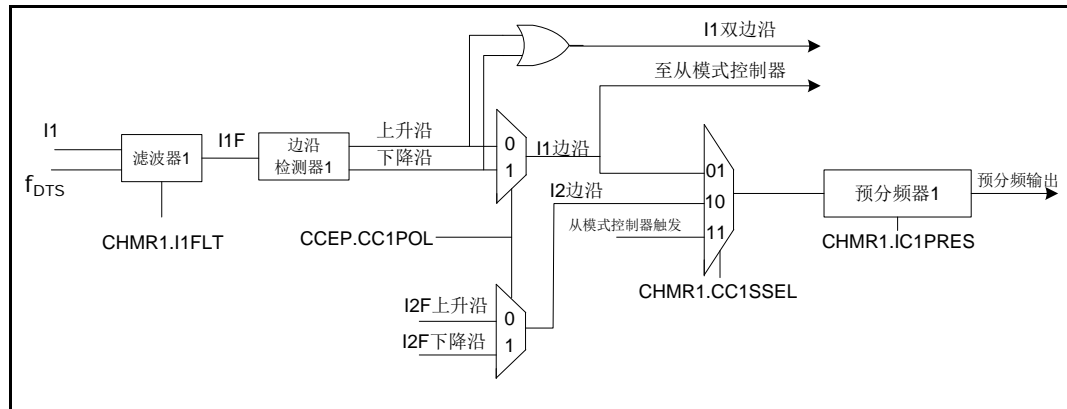


图 17-12 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

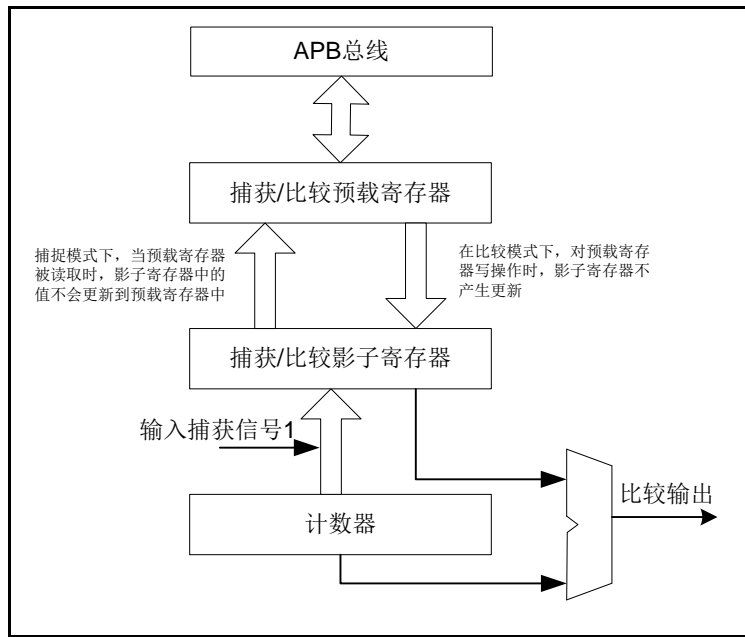


图 17-13 捕获/比较通道 1 结构图

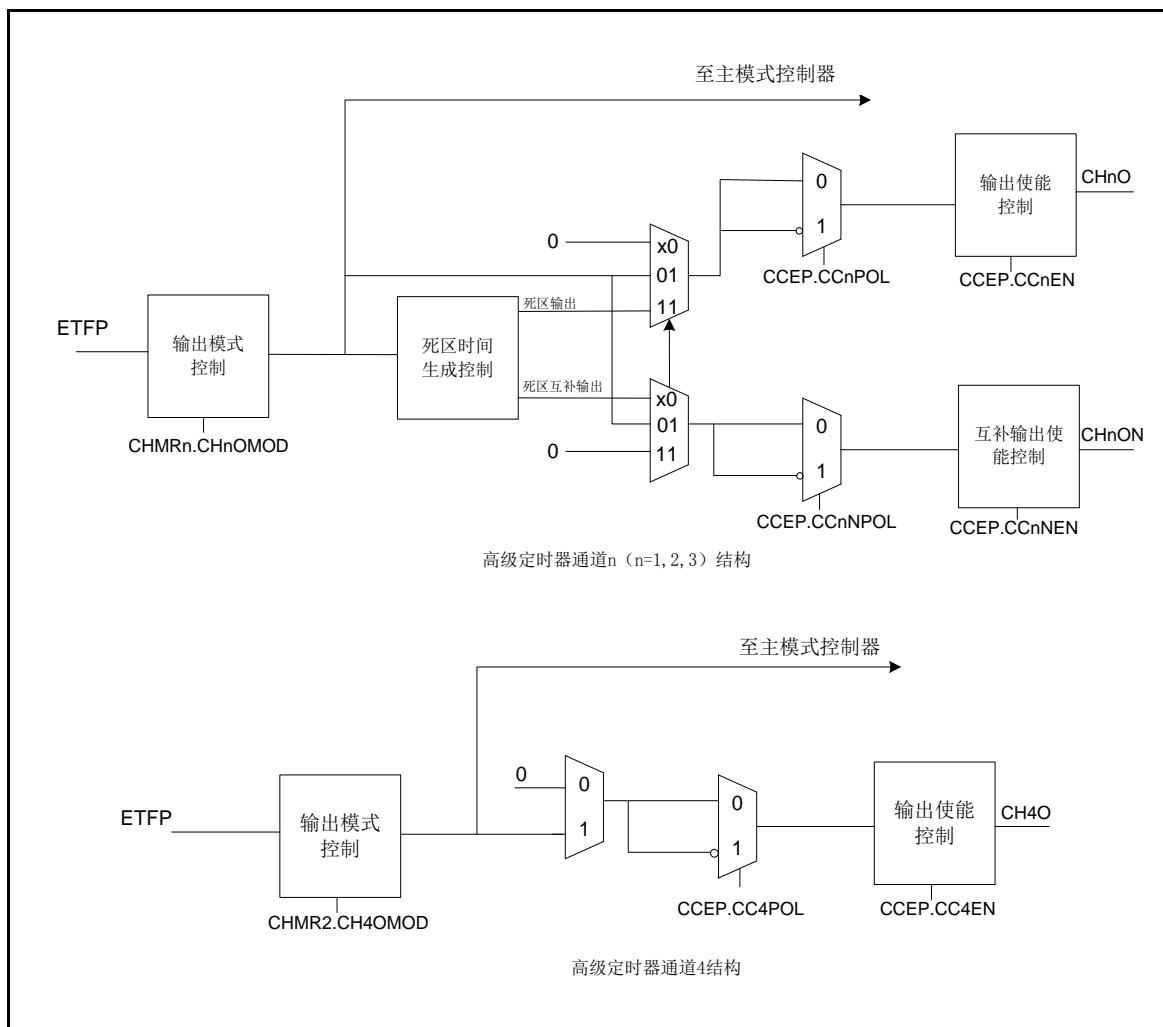


图 17-14 捕获/比较信道的输出部分

17.4.6 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器（AD16C4T_CCVALn）寄存器中。当捕获发生时，相应的 CHnIF 标志位（AD16C4T_RIF）会置位，同时会触发中断或 DMA（如果使能）请求。若发生捕获时，CHnIF 标志位已经置位，则过捕获 CHnOVIF 标志位（AD16C4T_RIF）置位。软件写'0'或读取 AD16C4T_CCVALn 寄存器中的捕获值都可以复位 CHnIF 标志位。对 CHnOVIF 位写'0'可清空该标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

1. 选择有效输入端：AD16C4T_CCVAL1 必须连接到 I1 输入端，因此需将 AD16C4T_CHMR1 寄存器中的 CC1SSEL 位写"01"。只要 CC1SSEL 不为"00"，通道被配置为输入且 AD16C4T_CCVAL1 寄存器为只读。
2. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号内信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
3. 选择 I1 信道的有效边沿变换。AD16C4T_CCEP 寄存器中的 CC1POL 写'0'(上升沿)。
4. 配置输入预分频器。
5. 置位 AD16C4T_CCEP 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
6. 如有需要，置位 AD16C4T_IER 寄存器中的 CC1IT 位，使能中断请求。置位 AD16C4T_DMAEN 寄存器中的 CC1DS 位，使能 DMA 请求。

当发生输入捕获时：

1. 有效边沿产生，AD16C4T_CCVAL1 寄存器获取计数器的值。
2. CH1IF 标志位置位（中断标志）。若至少 2 个连续的捕获发生，但标志位没有及时清除，则 CH1OVIF 也会置位。
3. 中断的产生取决于 CC1IT 位。
4. DMA 请求的产生取决于 CC1DS 位。

为了处理捕获溢出，建议在读取捕获标志位前先读取捕获数据。这可以避免错过读取捕获标志位之后，读数据之前产生的捕获。

注：捕获中断请求可由软件设置 AD16C4T_SGE 寄存器中 SGCCnE 位产生。

17.4.6.1 PWM输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下：

1. 为 AD16C4T_CCVAL1 选择有效的输入：AD16C4T_CHMR1 寄存器中的 CC1SSEL 位写"01"（I1 被选择）。
2. 为 I1 边沿检出选择有效的极性（用于捕获数据到 AD16C4T_CCVAL1 寄存器和计数器清零）：CC1POL 位写'0'（上升沿有效）。
3. 为 AD16C4T_CCVAL2 选择有效输入：AD16C4T_CHMR1 寄存器的 CC2SEL 位写"10"（I1 被选择）。
4. 为 I1 边沿检出选择有效极性（用于捕获数据到 AD16C4T_CCVAL2）：CC2POL 位写'1'。
5. 选择有效的触发输入：AD16C4T_SMCON 寄存器的 TSSEL 位写"101"（I1 边沿检出被选择）。
6. 配置从机模式控制器为复位模式：AD16C4T_SMCON 寄存器的 SMODS 位写"100"。
7. 使能捕获：AD16C4T_CCEP 寄存器的 CC1EN 位和 CC2EN 位写'1'。

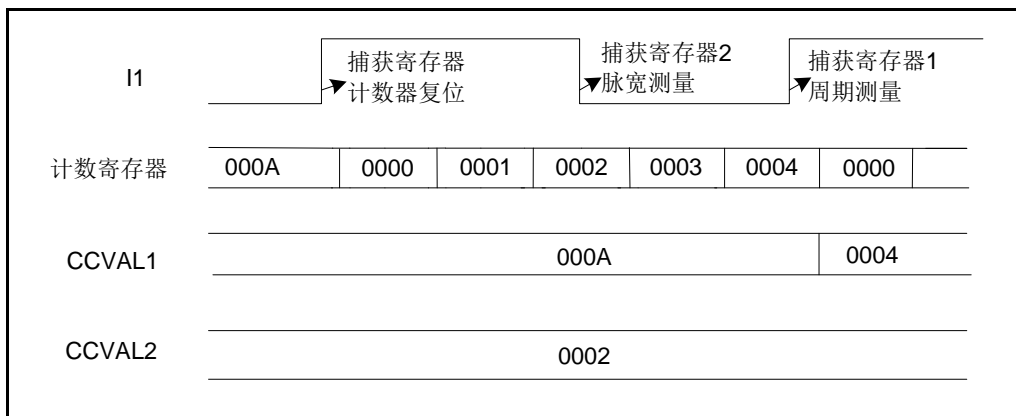


图 17-15 PWM 输入模式时序

17.4.7 PWM模式

脉宽调制模式可以产生一个 AD16C4T_AR 寄存器值确定频率，AD16C4T_CCVALn 寄存器值确定占空比的信号。

每个通道的 PWM 模式是相互独立的（每个 CHnO 输出一个 PWM），AD16C4T_CHMRn 寄存器的 CHnOMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 AD16C4T_CHMRn 寄存器的 CHnOPEN 位来使能相应的预载寄存器，最后还需置位 AD16C4T_CON1 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 AD16C4T_SGE 寄存器的 SGU 位来初始化所有的寄存器。

CHnO 的极性可通过 AD16C4T_CCEP 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHnO 的输出使能由 CCnEN、CCnNE、GOEN、OFFSSI 和 OFFSSR 位

(AD16C4T_CCEP 和 AD16C4T_BDCFG 寄存器) 组合控制。

在 PWM 模式 (1 或 2) 中, AD16C4T_COUNT 和 AD16C4T_CCVALn 寄存器的值会持续的比较, 确定 AD16C4T_CCVALn <= AD16C4T_COUNT 或 AD16C4T_CCVALn >= AD16C4T_COUNT (取决于计数器的计数方向)。

定时器产生 PWM 波形是边沿对齐或中心对齐, 取决于 AD16C4T_CON1 寄存器的 CMSEL 位。

17.4.7.1 PWM 边沿对齐模式

1. 递增计数配置

当 AD16C4T_CON1 寄存器的 DIRSEL 位为低时, 计数器递增计数。

下图给出了 AD16C4T_AR = 8 时的边沿对齐 PWM 波形。

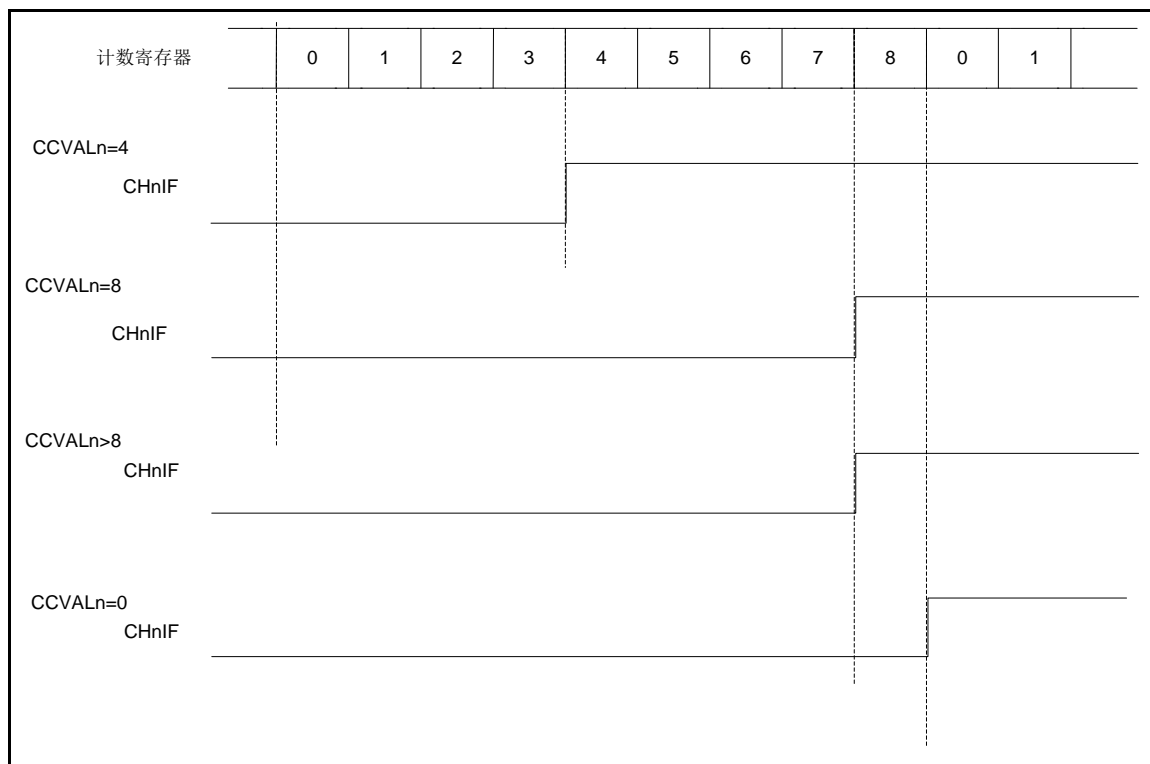


图 17-16 边沿对齐 PWM 波形 (AR=8)

2. 递减计数配置

当 AD16C4T_CON1 寄存器的 DIRSEL 位为高时, 计数器递减计数。

17.4.7.2 PWM 中心对齐模式

当 AD16C4T_CON1 寄存器中的 CMSEL 位不为 "00" 时, 中心对齐模式有效。计数器是递增、递减计数分别置比较标志位或递增递减都置比较标志位, 取决于 CMSEL 位的配置。AD16C4T_CON1 寄存器的方向位 (DIRSEL) 是由硬件更新的, 软件无法修改。

下图为中心对齐方式产生的 PWM 波形的例子:

- ◇ AD16C4T_AR=8
- ◇ PWM 模式 1
 - AD16C4T_CON1 寄存器的 CMSEL="01", 在中心对齐模式 1 下, 计数器向下计数时会置位比较标志位。

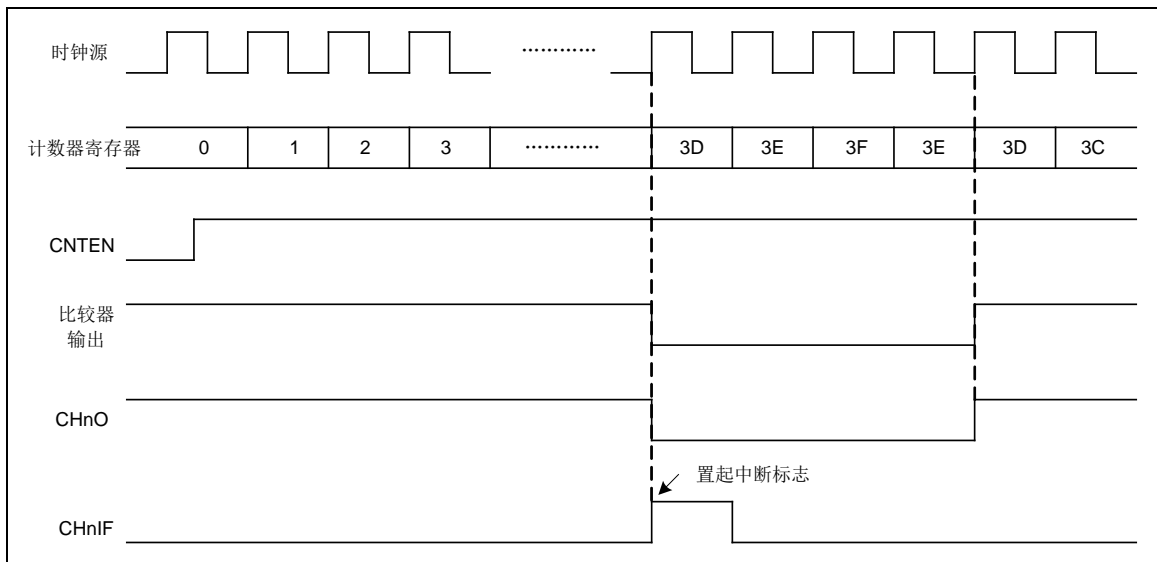


图 17-17 中心对齐 PWM 波形 (AR=8)

中心对齐模式的使用技巧:

- ◇ 当进入中心对齐模式后, 当前递增或递减配置生效。计数器递增或递减计数取决于 AD16C4T_CON1 寄存器的 DIRSEL 位的值。此外, 软件无法对 DIRSEL 和 CMSEL 位同时进行修改。
- ◇ 计数器在中心对齐模式下运行时, 对计数器写操作可能导致不可预知的结果。特别是:
 - 若向计数器写入的值大于自动重载值 ($AD16C4T_COUNT > AD16C4T_AR$), 计数方向不更新。例如, 如果计数器递增计数, 写入值后仍旧递增计数。
 - 若向计数器写 0 或 AD16C4T_AR 中的重载值, 则计数方向更新, 但并没有产生 UEV。
- ◇ 使用中心对齐模式最安全的方式是计数器开始计数前通过软件产生更新事件 (置位 AD16C4T_SGE 寄存器中的 SGU 位) 且在计数器运行过程中不对计数器写值。

17.4.8 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时, 输出比较功能:

- ◇ 输出比较模式 (AD16C4T_CHMRn 寄存器中的 CHnOMOD 位) 和输出极性 (AD16C4T_CCEP 寄存器中的 CCnPOL 位) 的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位 (AD16C4T_RIF 寄存器的 CHnIF 位)。
- ◇ 若相应的中断掩码置位, 则产生中断 (AD16C4T_IER 寄存器的 CCnIT 位)。
- ◇ 若相应的使能位置位 (AD16C4T_DMAEN 寄存器的 CCnDS 位, AD16C4T_CON2 寄存器的 CCDMASEL 位用于 DMA 请求的选择), 则发送 DMA 请求。

AD16C4T_CHMRn 寄存器中 CHnOPEN 位的值可决定 AD16C4T_CCVALn 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 UEV 对 CHnO 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. AD16C4T_AR 与 AD16C4T_CCVALn 寄存器中写入预期值。
3. 若需要产生中断请求，置位 AD16C4T_IER 寄存器中的 CCnIT 位。
4. 选择输出模式，例如：
 - CHnOMOD = "011"，当 CNTV 与 CCRVn 匹配时，CHnO 输出翻转。
 - CHnOPEN = '0'，关闭预载寄存器。
 - CCnPOL = '0'，选择有效极性为高。
 - CCnEN = '1'，使能输出。
5. AD16C4T_CON1 寄存器中的 CNTEN 位置位，使能计数器。

假设预载寄存器没有使能（CHnOPEN = '0'，否则 AD16C4T_CCVALn 影子寄存器只有在下次更新事件发生时才更新）。通过软件方式，AD16C4T_CCVALn 寄存器的值可随时更新控制输出波形。

17.4.8.1 外部事件清除比较输出

ETFP 输入端（AD16C4T_CHMRn 寄存器的 CHnOCLREN 位写'1'）上的高电平，可将给定通道的比较输出信号拉低。在下次更新事件（UEV）发生前，比较输出会一直保持为低。该功能只能应用在输出比较和 PWM 模式中，强制输出模式中不起作用。

ET 信号可以接到电流控制比较器的输出端。该例中，ET 须按如下流程配置：

1. 外部触发预分频器应该关闭：AD16C4T_SMCON 寄存器的 ETPSEL[1:0]位应该写"00"
2. 外部时钟源 2 关闭：AD16C4T_SMCON 寄存器的 ECM2EN 位写'0'
3. 外部触发极性（ETPOL）和外部触发滤波器（ETFLT）可根据用户需要配置

17.4.8.2 强制输出模式

在输出模式中（AD16C4T_CHMRn 寄存器中 CCnSSEL = "00"），软件可强制将每个输出比较信号（CHnO/CHnON）改为有效或无效状态，这种修改独立于输出比较寄存器和计数器的比较结果。

为了将某输出比较信号（CHnO）强制为有效状态，需将相应的 AD16C4T_CHMRn 寄存器中 CHnOMOD 位写"101"。因此，比较输出被强制为高（高时为有效状态）且 CHnO 的值为 CCnPOL 极性位的相反值。

例如：CCnPOL = '0'（CHnO 高电平有效），则 CHnO 被强制为高电平。

对 AD16C4T_CHMRn 寄存器的 CHnOMOD 位写"100"，比较输出可被置低。

无论怎样，AD16C4T_CCVALn 影子寄存器和计数器之间的比较仍然进行，相应的标志位仍可置位。

17.4.9 单脉冲模式

单脉冲模式（SPMEN）下，响应某个触发后，定时器的输出通道在可配置的延迟时间后产生一个脉冲，脉冲长度可配。从模式控制器可控制计数器的启动。脉冲波形可在输出比较模式和 PWM 模式下产生。置位 AD16C4T_CON1 寄存器的 SPMEN 位可选择单脉冲模式。计数器会在下次更新事件 UEV 产生时自动停止。

只有比较值不同于计数器初始值时，单脉冲才可以正确的产生。计数器开始计数前（定时器等待触发），必须如下配置：

- ◇ 递增计数：CNT < CCVALn ≤ AR（特别地，0 < CCVALn）
- ◇ 递减计数：CNT > CCVALn

基于 PWM 模式设置单脉冲输出波形的步骤如下：

- ◇ 设置 AD16C4T_CHMRn 寄存器的 CHnOMOD 位，选择 PWM 模式 1 或 2；
- ◇ 设置 AD16C4T_CCEPn 寄存器的 CCnPOL 位，选择通道端口 CHnO 的输出极性；
- ◇ 设置 AD16C4T_CON1 寄存器的 DIRSEL, CMSEL, SPMEN 位，配置为递增或递减计数，PWM 普通波形模式，单脉冲模式使能；
- ◇ 设置 AD16C4T_CHMR1 寄存器的 CH1OPREN = 1，AD16C4T_CON1 寄存器的 ARPEN = 1，使能比较寄存器和计数重载寄存器的缓冲功能（也可以根据实际情况不使能缓冲）；
- ◇ 设置 AD16C4T_CCVALn 寄存器和 AD16C4T_AR 寄存器，配置单脉冲输出延时和脉宽时间；
- ◇ 设置 AD16C4T_SGE 寄存器的 SGU 位来产生一个更新事件；
- ◇ 设置 AD16C4T_CON1 寄存器的 CNTEN=1 来启动计数器，也可以在触发模式下，通过外部触发输入信号来触发硬件自动设置 CNTEN=1。

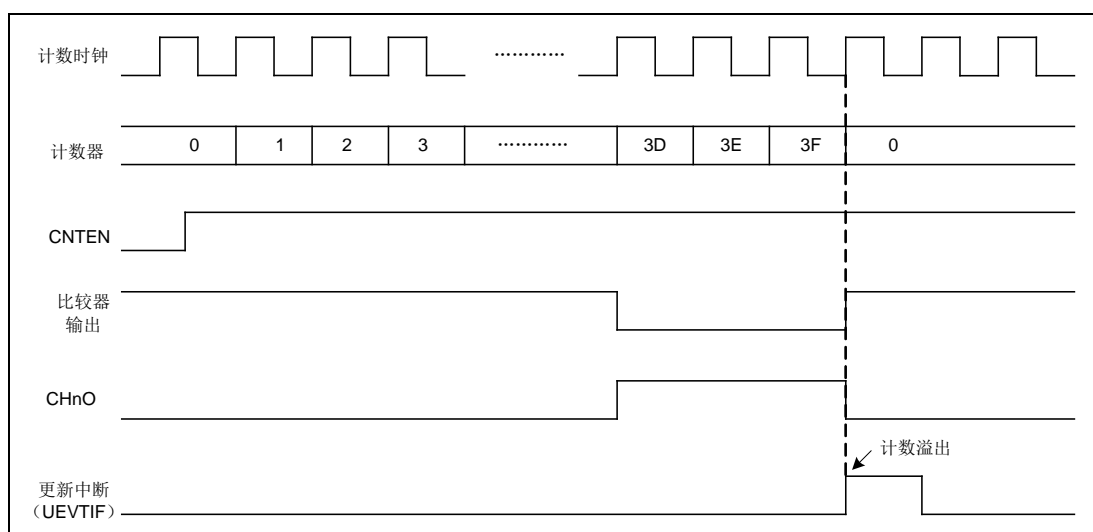


图 17-18 单脉冲模式

17.4.10 互补输出与死区时间

两个互补的通道输出信号，可以用来控制输出的瞬时开关。死区时间可配置。

每个输出可独立选择输出极性（主输出 CHnO 或互补输出 CHnON），该操作可通过写 AD16C4T_CCEP 寄存器的 CCnPOL 和 CCnNPOL 位完成。

互补信号 CHnO 和 CHnON 由几个控制位共同控制，分别是 AD16C4T_CCEP 寄存器中的 CCnEN 和 CCnNE 位，AD16C4T_BDCFG 和 AD16C4T_CON2 寄存器中的 GOEN、OISSn、OISSnN、OFFSSI 及 OFFSSR 位。特别是死区时间使能后的空闲状态的切换（GOEN 变为 0）。

置位 CCnEN 和 CCnNE 位，使能死区时间插入，若有刹车电路，同样需要置位 GOEN 位。AD16C4T_BDCFG 寄存器的 DT[7:0] 可以控制所有通道的死区时间的产生。根据比较输出波形，产生 CHnO 和 CHnON 两路输出。若 CHnO 和 CHnON 有效电平为高：

- ◇ CHnO 的输出信号与参考信号一致。上升沿除外，相对参考信号的上升沿，CHnO 输出会有延迟。
- ◇ CHnON 的输出信号与参考信号相反。上升沿除外，相对参考信号的下降沿，CHnON 输出会有延迟。

若延迟时间大于有效输出的宽度（CHnO 或 CHnON），则相应的脉冲不会产生。

下图给出了死区时间输出信号和比较输出波形之间的关系。假设 CCnPOL = 0, CCnNPOL = 0, GOEN = 1, CCnEN = 1, 和 CCnNE = 1

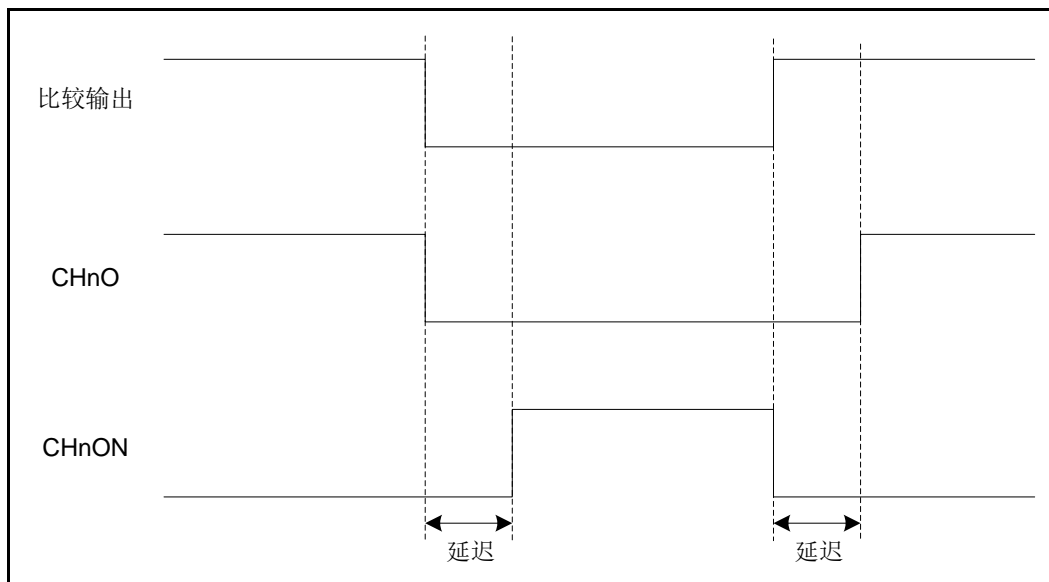


图 17-19 互补输出含死区时间插入

当 PWM 通道配置为互补输出时，如下寄存器控制位都会有缓冲：CHnOMOD、CCnE 和 CCnNE。发生互补通道更新事件时，这些寄存器位才会真正生效，这样就可以预先设置好下一步的配置，并同时对所有互补通道的配置进行更新。互补通道更新事件可以通过设置 AD16C4T_SGE 寄存器的 SGC0M=1 产生，或由触发信号产生（由 AD16C4T_SMCON 寄存器的 TSSEL 位选择触发信号）。

17.4.11 刹车功能

刹车功能模式由以下几个控制位进行设置：AD16C4T_BDCFG 寄存器中的 GOEN、OFFSSI 和 OFFSSR 位，AD16C4T_CON2 寄存器中的 OISSn 和 OISSnN 位，输出使能信号和无效电平都会被修改。

刹车源可以是刹车输入引脚、时钟失败事件以及软件控制 AD16C4T_SGE 寄存器的 SGBRK 位。时钟失败事件由时钟控制器（CMU）中，时钟安全系统（CSS）产生。时钟安全系统（CSS）详细信息可参考时钟安全系统章节。

系统复位后，刹车电路被禁止且 GOEN 位被复位。置位 AD16C4T_BDCFG 寄存器的 BRKEN 位可启用刹车功能，同样寄存器中，BRKEN 位可选择刹车输入信号的极性。BRKEN 和 BRKP 位可同时修改。对 BRKEN 和 BRKP 位写操作后，1 个 APB 时钟周期延时后写入值才会生效。因此，写操作后，需等待 1 个 APB 时钟周期后才能正确读回写入值。

由于 GOEN 的下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（AD16C4T_BDCFG 寄存器中）之间插入了一个同步电路。这也导致了异步和同步信号之间会产生一些延迟。特别是 GOEN 之前为低时对 GOEN 写 1 操作后，要读取正确值，必须先插入一个延时（空指令）。这是因为写入的是异步信号，而读取的是同步信号。

当发生刹车请求时（刹车输入端有刹车电平）：

- ◇ GOEN 位被异步清除，输出端进入无效状态，空闲状态或复位状态（OFFSSI 位选择）。即使 MCU 的振荡器关闭，该功能仍然有效。
- ◇ 一旦 GOEN=0，每个通道输出预先配置的电平。AD16C4T_CON2 寄存器中的 OISSn 位配置该电平。如果 OFFSSI=0，则定时器释放使能输出，否则使能输出一直为高。
- ◇ 当使用互补输出时：
 - 输出端首先被置于复位状态，无效状态（取决于极性）。这个过程异步的，即使定时器没有时钟也有效。
 - 如果定时器时钟仍然存在，则死区时间生成器会重新生效，这样在死区时间后，OISSn 和 OISSnN 位的配置电平可驱动输出。这种情况下，CHnO 和 CHnON 无法驱动输出端都为有效电平。
 - 输出首先被置于复位状态即无效的状态（取决于极性）。这是异步操作，即使定时器没有时钟时，此功能也有效。注，由于对 GOEN 的重新同步，死区时间的周期会比通常情况下长一些（大约 2 个 TIMER 模块时钟周期）。
 - 如果 OFFSSI = 0，则定时器释放使能输出，否则使能输出保持或变高（一旦 CCnEN 和 CCnNE 有一个变高时）。
- ◇ 当刹车状态标志位（AD16C4T_RIF 寄存器中的 BRKIF 位）置位时，若 AD16C4T_IER 寄存器中的 BRKIT 位置位，可触发中断。
- ◇ 当 AD16C4T_BDCFG 寄存器中的 AOEN 位置位时，在下次更新事件（UEV）发生时，GOEN 位会自动置位。例如，该功能可用来整形。否则，GOEN 位会保持为低，直到对其写'1'操作，该特性可用于安全方面的应用，可以将刹车输入端接到一个电源驱动的报警端、热敏传感器或其他安全器件上。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能置位（自动地或者通过软件）GOEN。同时，状态标志 BRKIF 不能被清除。

除刹车输入和输出管理，为保证应用程序的安全，内部刹车电路具有写保护功能。用户可冻结几个配置参数（死区时间，CHnO/CHnON 极性和失能时状态，CHnOMOD 配置，刹车使能和极性）。通过 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位，可从三个保护等级中选择一种保护等级。MCU 复位后，LOCKLVL 位只能写一次。

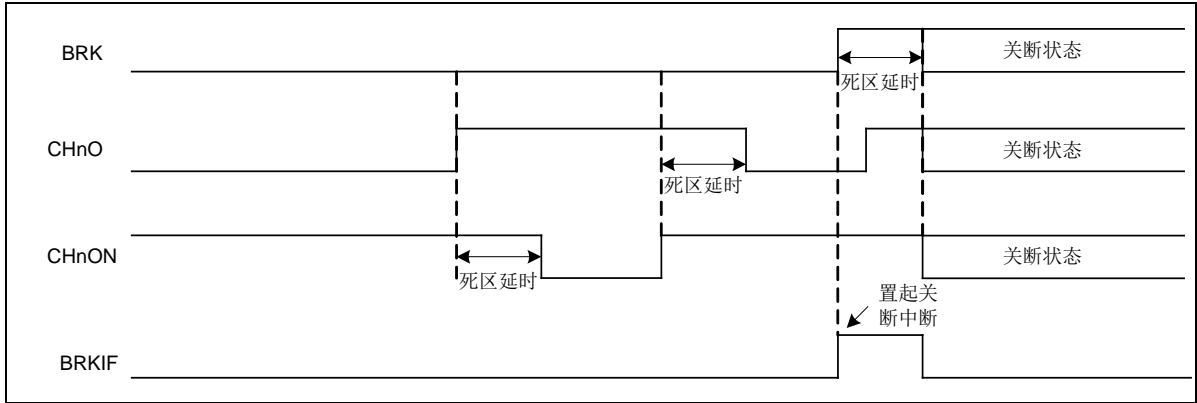


图 17-20 刹车输出行为

17.4.12 编码器接口模式

编码器接口模式的三种配置：若计数器只根据 I2 上的边沿计数，则 AD16C4T_SMCON 寄存器中的 SMODS = "001"；若计数器只根据 I1 上的边沿计数，则 AD16C4T_SMCON 寄存器中的 SMODS = "010"；若计数器同时根据 I1 和 I2 上的边沿计数，则 AD16C4T_SMCON 寄存器中的 SMODS = "011"。

配置 AD16C4T_CCEP 寄存器中的 CC1POL 和 CC2POL 位的值可选择 I1 和 I2 的极性。如果需要，也可以配置输入滤波器。

CH1_IN 和 CH2_IN 端口作为增量编码器的接口。当计数器使能时，计数器根据 I1 或 I2 上滤波后的有效电平变化时钟计数。I1 和 I2 滤波后的有效信号顺序会产生计数脉冲及方向信号。计数器是递增或递减计数由信号的跳变顺序决定，AD16C4T_CON1 寄存器中的 DIRSEL 计数方向位由自动硬件更新。

编码器接口模式的工作方式类似于一个带有方向选择的外部时钟。计数器在 0 到 AD16C4T_AR 寄存器中的自动重载值之间连续计数。因此，必须在开始计数前配置 AD16C4T_AR 寄存器。同样的，捕获器、预分频器、重复计数器、触发输出的特性正常工作。设定编码模式和选择外部时钟源 2 不兼容，不可以同时选择。

该模式下，计数器会根据增量式编码器的速度和方向自动修改，计数器的值反应的是编码器的位置。计数方向对应着连接传感器的旋转方向。

下表列出了所有的可能组合，假设 I1 和 I2 不同时变换。

有效边沿	有效边沿相对信号的电平 (I1 滤波信号对应 I2,I2 滤波信号对应 I1)	I1 滤波信号		I2 滤波信号	
		上升	下降	上升	下降
仅在 I1 计数	高	下降	上升	不计数	不计数
	低	上升	下降	不计数	不计数
仅在 I2 计数	高	不计数	不计数	上升	下降
	低	不计数	不计数	下降	上升
在 I1 和 I2 上计数	高	下降	上升	上升	下降
	低	上升	下降	下降	上升

表 17-1 计数方向与编码器信号的关系

外部增量编码器可直接与 MCU 连接，无需外部逻辑接口逻辑。而比较器通常用于将编码器的差分输出转换为数字信号，这极大地增加了抗噪声能力。编码器的第三个输出端用于指示机械零点，可以连接到外部中断输入引脚以触发一次计数复位。

下图给出了计数器操作的例子，给出了计数信号了产生和方向控制。同样给出了选择双边沿时，输入抖动如何被补偿。输入抖动可能发生在传感器靠近切换点处。

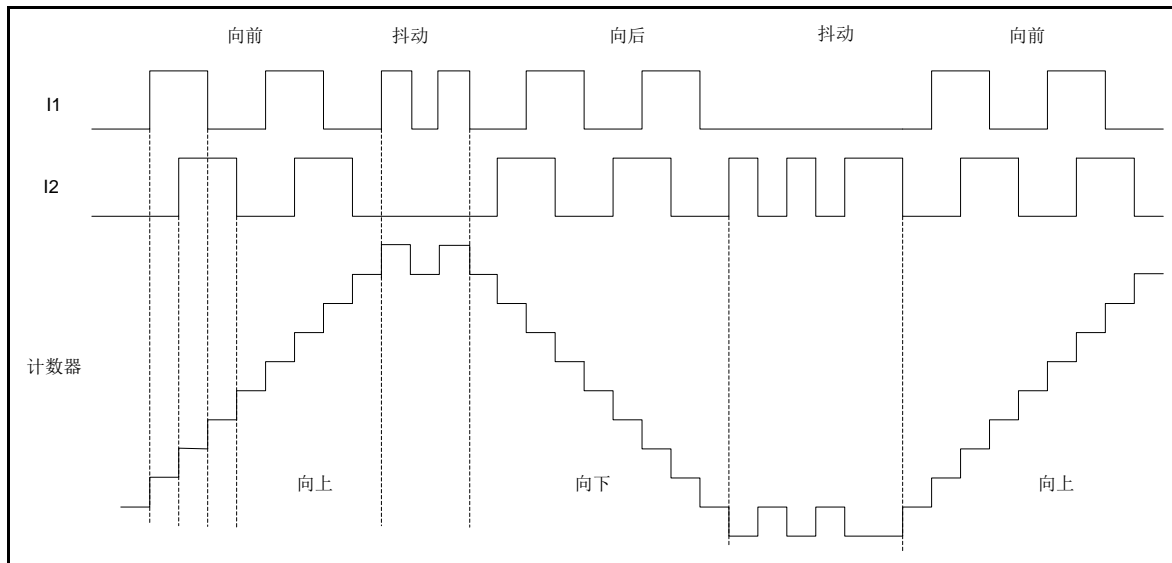


图 17-21 编码器接口模式下的计数操作

下图给出了计数器在 I1 滤波信号极性反相时的计数过程（除了 CC1POL = '1'，其他配置与上面一致）

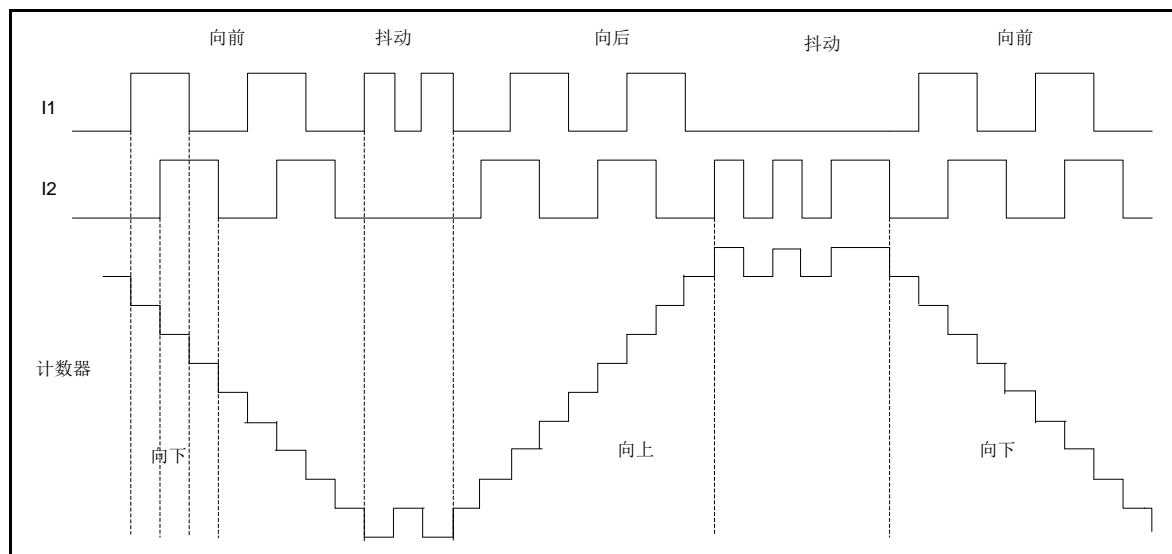


图 17-22 I1 滤波后极性反相时编码器接口例子

当配置为编码器接口模式时，定时器可提供传感器的当前位置信息。配置一个额外定时器为捕获模式，用于测量两个编码器事件的间隔，根据间隔时常获取动态信息（速度、加速度、减速度）。编码器用于指示机械零点的输出就是此用处。根据编码器两个事件间隔，可以周期性的读取计数器的值。如果允许，可以将计数器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且可由其它定时器产生）。条件允许时，可通过实时时钟产生 DMA 请求的方式读取计数器值。

17.4.13 输入异或功能

通过 AD16C4T_CON2 寄存器中 I1FSEL 位, 可将 CH1_IN、CH2_IN 和 CH3_IN 三个输入引脚进行异或 (XOR) 逻辑后, 连接到通道 1 的输入滤波器。

异或逻辑输出可用于定时器的所有输入功能, 如触发或输入捕获。该功能参见下节的霍尔传感器接口。

17.4.14 霍尔传感器接口

高级控制定时器产生 PWM 信号驱动电机, 另外一个定时器作为“接口定时器”。“接口定时器”捕获 3 个引脚输入 (CH1_IN, CH2_IN, CH3_IN), 引脚信号经过一个异或逻辑门后连接到 I1 的输入通道 (置位 AD16C4T_CON2 寄存器中的 I1FSEL 位选择)。

从模式控制器配置为复位模式, 3 个输入后的异或信号作为从输入。因此, 只要 3 个输入信号有一个变化, 计数器重新从 0 开始计数。霍尔输入端的任何变化都可触发创建一个时钟基准。

17.4.15 外部触发的同步

AD16C4Tn 定时器可在多种模式下与外部触发同步: 复位模式、门控模式及触发模式。

17.4.15.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外, 若 AD16C4T_CON1 寄存器的 UERSEL 位为低时会产生一次更新事件 UEV。所有预载寄存器 (AD16C4T_AR, AD16C4T_CCVALn) 都会因更新事件 UEV 而被更新。

在下面例子中, I1 输入端的上升沿让递增计数被清空:

- ◇ 配置通道 1 上检测 I1 上的上升沿。配置输入滤波周期 (本例无需滤波器, 故 I1FLT = "0000")。触发捕获分频器没有使用, 无需配置。CC1SSEL 位只选择输入捕获源, AD16C4T_CHMR1 寄存器中 CC1SSEL = "01"。AD16C4T_CCEP 寄存器中 CC1POL = 0 以确定极性 (只检测上升沿)。
- ◇ 定时器配置位复位模式: AD16C4T_SMCON 寄存器中 SMODS = "100"。选择 I1 作为输入源: AD16C4T_SMCON 寄存器中 TSSEL = "101"。
- ◇ 启动计数器: AD16C4T_CON1 寄存器中 CNTEN = '1'。

计数器依据内部时钟开始计数, 正常计数直到 I1 上出现上升沿。当 I1 上出现上升沿时, 计数器会被清零且从 0 重新开始计数。同时, 标志位置位 (AD16C4T_RIF 寄存器中 TRGIF 位), 如果中断及 DMA 使能 (取决于 AD16C4T_IER 寄存器中的 TRGIT 和 AD16C4T_DMAEN 寄存器的 TDS 位), 会发送中断及 DMA 请求。

下图给出了当自动重载寄存器 AD16C4T_AR = 0x36 时的信号变化。由于 I1 输入的再同步电路, I1 上的上升沿和计数器实际复位之间会存在延时。

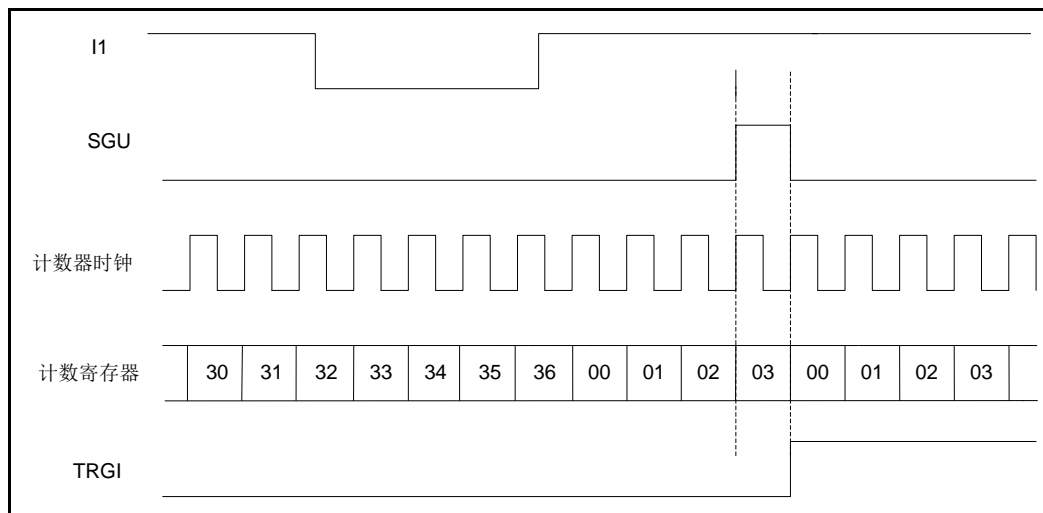


图 17-23 复位模式控制电路

17.4.15.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置通道 1 在 I1 上检测低电平。配置输入滤波周期（本例不需要滤波器，I1FLT = "0000"）。触发捕获分频器没有使用，无需配置。AD16C4T_CHMR1 寄存器中的 CC1SSEL = "01"，选择输入捕获源。AD16C4T_CCEP 寄存器中 CC1POL = '1'，确认极性（只检测低电平）。
- ◇ 配置定时器为门控模式：AD16C4T_SMCON 寄存器中 SMODS = "101"。选择 I1 作为输入源：AD16C4T_SMCON 寄存器中 TSSEL = "101"。
- ◇ 使能计数器：AD16C4T_CON1 寄存器中 CNTEN = '1'（门控模式中，如果 CNTEN = '0'，无论触发输入为何电平，计数器都不会启动）。

只要 I1 为低电平，计数器依据内部时钟开始计数，一旦 I1 为高则停止计数。由于 I1 输入端再同步电路的原因，I1 上出现上升沿和计数器实际停止之间会有一定的延时。

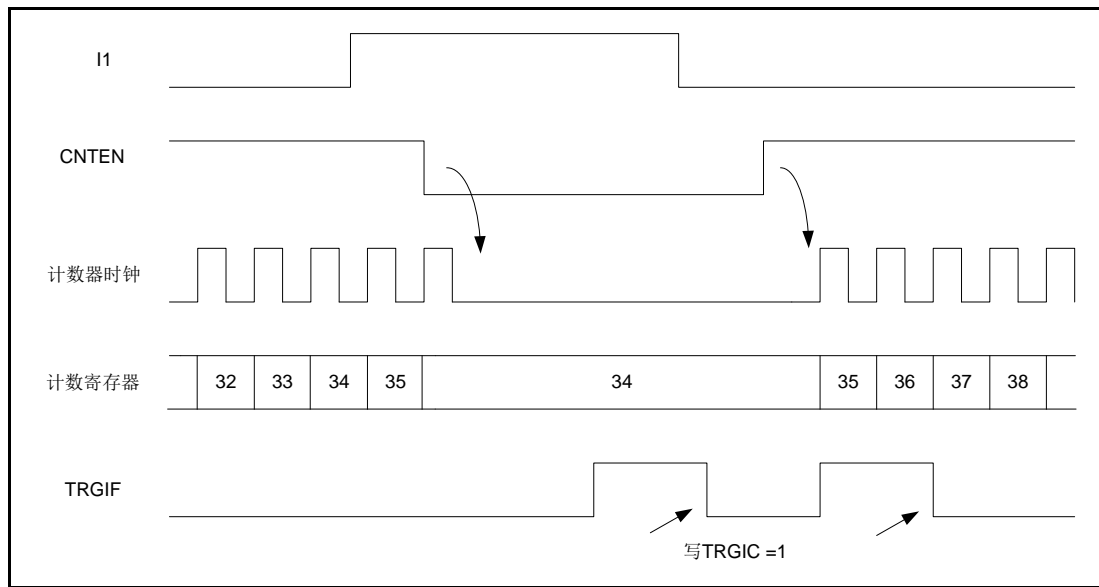


图 17-24 门控模式控制电路

17.4.15.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置通道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。AD16C4T_CHMR1 寄存器中 CC2SEL = "01",用于选择捕获源。AD16C4T_CCEP 寄存器中 CC2POL = '1', 确认极性（只检测低电平）。
- ◇ 配置定时器为触发模式：AD16C4T_SMCON 寄存器中 SMODS = "110"。AD16C4T_SMCON 寄存器中 TSSEL = "110", 用于选择输入源。

I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGIF 标志位。

由于 I2 输入的再同步原因，I2 上出现上升沿和计数器实际停止之间会有一定的延时。

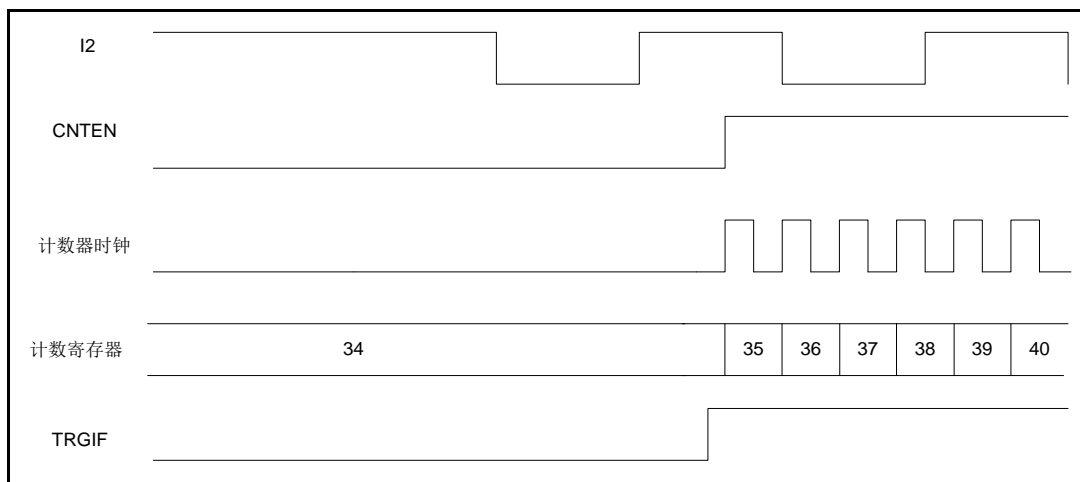


图 17-25 触发模式控制电路

17.4.15.4 选择外部时钟源 2 的触发模式

外部时钟源 2 可和其他模式一起使用（除编码模式）。ET 信号可作为外部时钟输入，另一个输入可选择为触发输入（复位模式、门控模式或触发模式）。不推荐用 AD16C4T_SMCON 寄存器的 TSSEL 位选择 ET 作为 TI。

下面的例子中，一旦 I1 上出现上升沿时，计数器会依据 ET 信号的每个上升沿递增计数。

- ◇ 通过 AD16C4T_SMCON 寄存器，配置外部触发输入电路，过程如下：

ETFLT = "000": 无滤波

ETPSEL = "00": 禁止分频

ETPOL = '0': 检测 ET 的上升沿，ECM2EN = '1'使能外部时钟模式 2

- ◇ 配置通道 1 检测 I 的上升沿，过程如下：

I1FLT = "0000": 无滤波。

触发捕获分频器没有使用，无需配置。

AD16C4T_CHMR1 寄存器中 CC1SSEL = "01"选择输入捕获源，AD16C4T_CCEP 寄存器的 CC1POL = '0'确认极性（只检测上升沿）。

- ◇ 配置定时器为触发模式：AD16C4T_SMCON 寄存器中 SMODS = "110"。

AD16C4T_SMCON 寄存器中 TSSEL = "101"选择 I1 作为输入源。

I1 上出现上升沿时，计数器使能且 TRGIF 标志位置位，然后计数器根据 ET 上的上升沿开始计数。

由于 ETRP 输入再同步电路的原因，ET 信号的上升沿和实际计数器的复位会有延时。

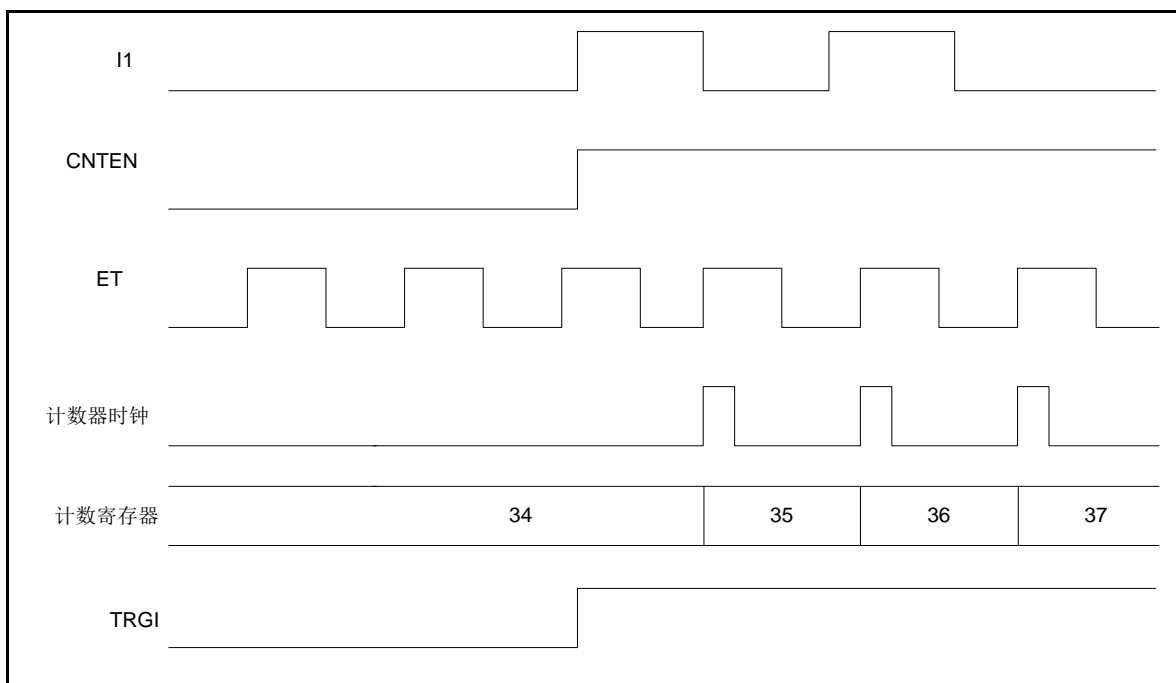


图 17-26 外部时钟源 2+触发模式下的控制电路

17.4.16 6步PWM生成

当在一个通道上需要互补输出时，预装载位有 CHxMOD、CCxEN 和 CCxNEN。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样可以预先设置好下一步骤配置，并在同一个时刻同时修改所有通道的配置。COM 可以通过设置 AD16C4T_SGE 寄存器的 SGC0M 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位 (AD16C4T_RIF 寄存器中的 COMI 位)，这时如果已设置了 AD16C4T_IER 寄存器的 COMI 位，则产生一个中断；如果已设置了 AD16C4T_DMAEN 寄存器的 COMDS 位，则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 CHx 和 CHxN 输出。

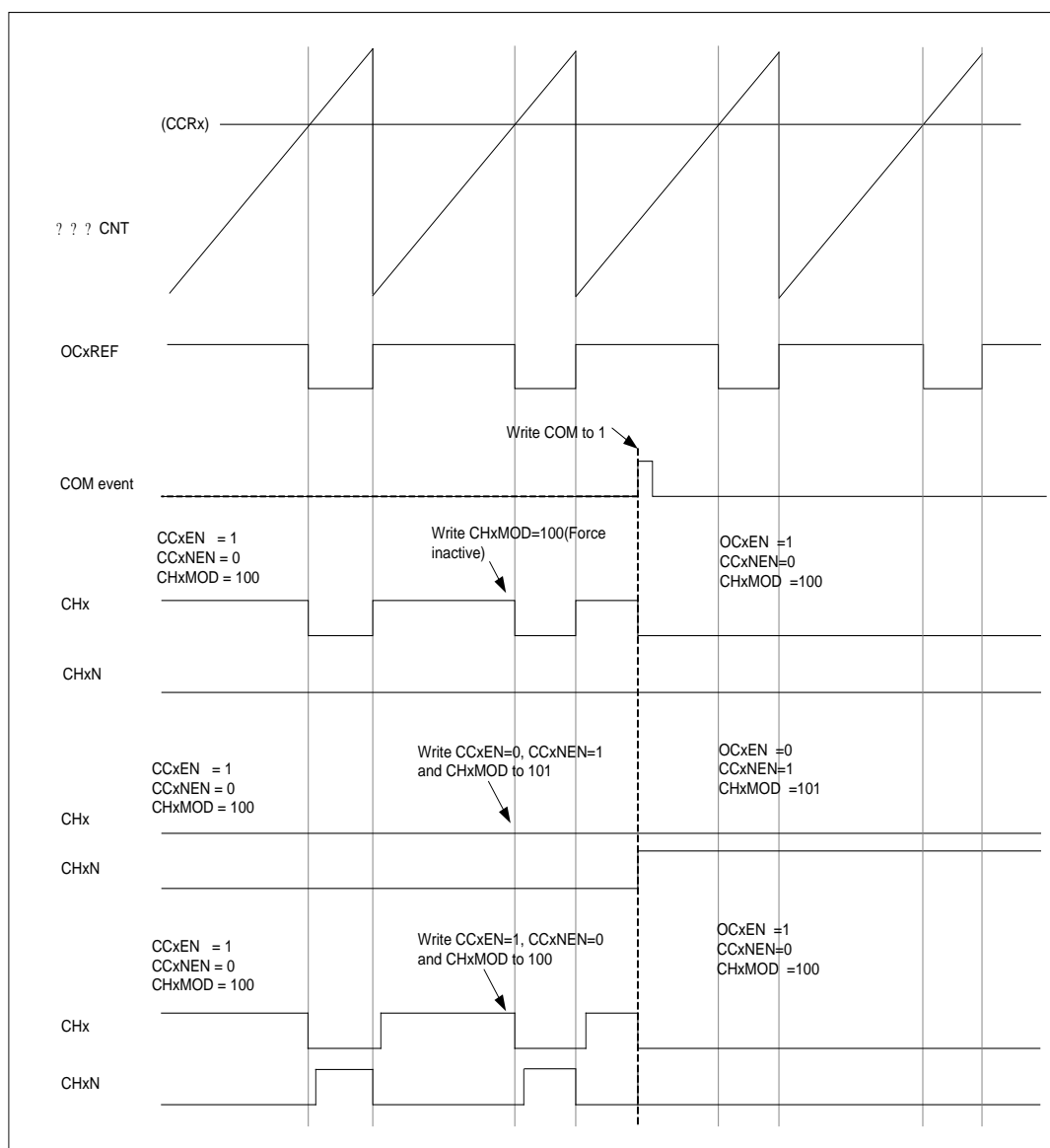


图 17-27 6步 PWM 波形示例

17.4.17 调试模式

当微控制器进入调试模式 (CPU 内核停止)，AD16C4T 计数器可被设置为停止计数。

17.5 特殊功能寄存器

17.5.1 寄存器列表

AD16C4Tn 寄存器列表		
寄存器名称	偏移地址	寄存器描述
AD16C4T_CON1	000 _H	控制寄存器 1
AD16C4T_CON2	004 _H	控制寄存器 2
AD16C4T_SMCON	008 _H	从模式控制寄存器
AD16C4T_IER	00C _H	中断使能寄存器
AD16C4T_IDR	010 _H	中断禁止寄存器
AD16C4T_IVS	014 _H	中断有效状态寄存器
AD16C4T_RIF	018 _H	原始中断标志寄存器
AD16C4T_IFM	01C _H	中断标志屏蔽寄存器
AD16C4T_ICR	020 _H	中断清零寄存器
AD16C4T_SGE	024 _H	事件生成寄存器
AD16C4T_CHMR1	028 _H	通道捕获/比较模式寄存器 1
AD16C4T_CHMR2	02C _H	通道捕获/比较模式寄存器 2
AD16C4T_CCEP	030 _H	捕获/比较使能极性寄存器
AD16C4T_COUNT	034 _H	计数器
AD16C4T_PRES	038 _H	时钟预分频器
AD16C4T_AR	03C _H	计数器自动重载寄存器
AD16C4T_REPAR	040 _H	重复计数寄存器
AD16C4T_CCVAL1	044 _H	通道捕获/比较寄存器 1
AD16C4T_CCVAL2	048 _H	通道捕获/比较寄存器 2
AD16C4T_CCVAL3	04C _H	通道捕获/比较寄存器 3
AD16C4T_CCVAL4	050 _H	通道捕获/比较寄存器 4
AD16C4T_BDCFG	054 _H	刹车和死区配置寄存器
AD16C4T_DMAEN	058 _H	DMA 使能寄存器

17.5.2 寄存器描述

17.5.2.1 控制寄存器 1 (AD16C4T_CON1)

控制寄存器 1 (AD16C4T_CON1)																															
偏移地址: 000 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DBGSEL	Reserved	OCCISS			OCCISP	DFCKSEL		ARPEN	CMSEL		DIRSEL	SPMEN	UERSEL	DISUE	CNTEN

Reserved	Bit 31-16	-	保留, 必须保持为复位值
DBGSEL	Bit 15	R/W	调试时通道输出状态选择: 0: 通道输出为高阻态 1: 通道输出保持
Reserved	Bit 14	-	保留, 必须保持为复位值
OCCISS	Bit 13-11	R/W	通道输出清除内部触发源选择: 000: 无效 001: 清除触发源通道 0 010: 清除触发源通道 1 011: 清除触发源通道 2 100: 清除触发源通道 3 其他: 预留, 不允许编程该值
OCCISP	Bit 10	R/W	通道输出清除内部触发源极性: 0: 低电平有效 (OCCISS 设置为无效时该位设 0) 1: 高电平有效
DFCKSEL	Bit 9-8	R/W	死区发生器和数字滤波器工作时钟频率 Fdfck 选择位 该位为计数器时钟 (INT_CLK) 频率, 显示了死区时间和由死区发生器与数字滤波器 (ETR, In) 所用的采样时钟 (tDTS) 之间的分频关系。 00: tDTS=tINT_CLK 01: tDTS=2*tINT_CLK 10: tDTS=4*tINT_CLK 11: 预留, 不允许编程该值
ARPEN	Bit 7	R/W	计数器自动重载寄存器预载 0: AD16C4T_AR 寄存器未缓冲 1: AD16C4T_AR 寄存器被装入缓冲器
CMSEL	Bit 6-5	R/W	中心对齐模式选择 00: 边沿对齐模式。计数器根据方向为 (DIRSEL) 来向上或向下计数。 01: 中心对齐模式 1。计数器以交替方式向上或向

			<p>下计数。仅当计数器向下计数时，配置为输出的通道（AD16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位才会被设置。</p> <p>10: 中心对齐模式 2。计数器以交替方式向上或向下计数。仅当计数器向上计数时，配置为输出的通道（AD16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位才会被设置。</p> <p>11: 中心对齐模式 3。计数器以交替方式向上或向下计数。当计数器向上或向下计数时，配置为输出的通道（AD16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位均会被设置。</p> <p>注意：当计数器使能时（CNTEN=1），不允许从边沿对齐模式转换到中心对齐模式。</p>
DIRSEL	Bit 4	R/W	<p>计数器方向选择</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p> <p>注意：当计数器配置为中心对齐模式或者编码器模式时，该位只读。</p>
SPMEN	Bit 3	R/W	<p>使能单脉冲模式</p> <p>0: 当发生更新事件时，计数器不停止。</p> <p>1: 当发生下一次更新事件（CNTEN 位清零）时，计数器停止。</p>
URSEL	Bit 2	R/W	<p>选择更新事件请求</p> <p>该位由软件置 1 或清零，来选择 UEV 事件源。</p> <p>0: 如果更新中断或 DMA 请求使能，则下述任一时间都可产生更新中断或 DMA 请求：</p> <ul style="list-style-type: none"> - 计数器上溢/下溢 - 设置 SGU 位 - 从模式控制器产生的更新 <p>1: 如果更新中断或 DMA 请求使能，仅计数器上溢/下溢才能产生更新中断或 DMA 请求中断</p>
DISUE	Bit 1	R/W	<p>禁止更新事件</p> <p>该位由软件置 1 或清零来使能/禁止 UEV 事件的产生。</p> <p>0: UEV 使能。更新事件（UEV）由下列任一事件产生：</p> <ul style="list-style-type: none"> - 计数器上溢/下溢 - 设置 SGU 位 - 从模式控制器产生的更新 <p>缓冲寄存器载入他们的预载值。</p> <p>1: UEV 禁止。不产生更新事件，影子寄存器保持</p>

			他们的值 (ARRV, ICnPRES, CCRVn)。如果从模式控制器接收到硬件复位, 计数器和预分频器将被重新初始化。
CNTEN	Bit 0	R/W	<p>使能计数器</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p> <p>注意: 如果软件设置了 CNTEN 位, 外部时钟, 门控模式和编码器模式才能工作。触发模式可由硬件自动设置 CNTEN 位。</p>

17.5.2.2 控制寄存器 2 (AD16C4T_CON2)

控制寄存器 2 (AD16C4T_CON2)																															
偏移地址: 004 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														OISS4	OISS3N	OISS3	OISS2N	OISS2	OISS1N	OISS1	I1FSEL	TRGOSEL			CCDMASEL	CCUSEL	Reserved	CCPCEN			

Reserved	Bit 31-15	-	保留, 必须保持为复位值
OISS4	Bit 14	R/W	通道 4 输出的空闲状态选择位 参考 OISS1 描述
OISS3N	Bit 13	R/W	通道 3 互补输出的空闲状态选择位 参考 OISS1N 描述
OISS3	Bit 12	R/W	通道 3 输出的空闲状态 3 选择位 参考 OISS1 描述
OISS2N	Bit 11	R/W	通道 2 互补输出的空闲状态选择位 参考 OISS1N 描述
OISS2	Bit 10	R/W	通道 2 输出的空闲状态选择位 参考 OISS1 描述
OISS1N	Bit 9	R/W	通道 1 互补输出的空闲状态选择位 0: 当 GOEN=0, 在一段死区时间后, CH1ON=0 1: 当 GOEN=0, 在一段死区时间后, CH1ON=1 注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1N 不可更改。
OISS1	Bit 8	R/W	通道 1 输出的空闲状态选择位 0: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1O=0 1: 当 GOEN=0, 如果 CH1ON 已实现, 在一段死区时间后, CH1O=1 注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 1,2,或 3 后, OISS1 不可更改。
I1FSEL	Bit 7	R/W	选择 I1 引脚功能 0: AD16C4T_CH1 引脚与 I1 输入连接 1: AD16C4T_CH1, CH2 和 CH3 引脚与 I1 输入 (XOR) 连接。
TRGOSEL	Bit 6-4	R/W	选择主模式 TRGOUT 输出 为同步 (TRGOUT), 该位可选择在主模式下发送至从计数器的信息。 000: 复位-AD16C4T_SGE 寄存器中的 SGU 位

			<p>被采用为触发输出 (TRGOUT)。如果复位由触发输入生成 (从模式控制器配置复位模式), 则相较于实际复位, TRGOUT 上的信号将会延迟。</p> <p>001: 使能-计数器使能新高 CNTEN 被用作触发输出 (TRGOUT)。在从计数器使能的情况下, 该设置用于在同一时间启动数次或者用来控制窗口。计数器使能信号是由 CNTEN 控制位与配置为门控模式的触发输入进行 OR 操作产生的。当计数器使能信号由触发输入控制, TRGOUT 上会产生延迟, 除非被选为主/从模式 (参考 AD16C4T_SMCON 寄存器中的 MSCFG 位的描述)。</p> <p>010: 更新-更新事件被选为触发输出 (TRGOUT)。举例, 主计数器可被用作从计数器的预分频器。</p> <p>011: 比较脉冲-一旦捕获或者比较匹配发生, 当 CH1IF 标志位被置起 (即便已为高电平), 触发输出会发送一个正脉冲。</p> <p>100: 比较- 通道 1 比较输出信号用作触发输出 TRGOUT</p> <p>101: 比较- 通道 2 比较输出信号用作触发输出 TRGOUT</p> <p>110: 比较- 通道 3 比较输出信号用作触发输出 TRGOUT</p> <p>111: 比较- 通道 4 比较输出信号用作触发输出 TRGOUT</p>
CCDMASEL	Bit 3	R/W	<p>使能捕获/比较 DMA</p> <p>0: 当 CCn 事件发生, 会发出 CCn DMA 请求。</p> <p>1: 当发生更新时间, 会发出 CCn DMA 请求。</p>
CCUSEL	Bit 2	R/W	<p>选择捕获比较更新</p> <p>0: 当捕获/比较控制位为预载值 (CCPCEN=1), 则当设置 SGCOM 位时才会被更新。</p> <p>1: 当捕获/比较控制位为预载值 (CCPCEN=1), 则当设置 SGCOM 位或者当 TI 边沿上升时, 均会被更新。</p> <p>注意: 该位只有用作于通道时才有互补输出。</p>
Reserved	Bit 1	-	<p>保留, 必须保持为复位值</p>
CCPCEN	Bit 0	R/W	<p>使能捕获比较预载控制</p> <p>0: CCnEN, CCnNE 和 CHnOMOD 不为预载值。</p> <p>1: CCnEN, CCnNE 和 CHnOMOD 为预载值。当写入预载值后, 仅当发生通信事件 (COM) 时 (SGCOM 位设置或者 TI 上检测到上升沿由 CCUSEL 位决定), 才会被更新)。</p> <p>注意: 该位只有用作于通道时才有互补输出。</p>

17.5.2.3 从模式控制寄存器 (AD16C4T_SMCON)

从模式控制寄存器 (AD16C4T_SMCON)																															
偏移地址: 008 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ETPOL	ECM2EN	ETPSEL	ETFLT				MSCFG	TSSEL			OCCS	SMODS			

Reserved	Bit 31-16	-	保留, 必须保持为复位值
ETPOL	Bit 15	R/W	<p>选择外部触发信号边沿</p> <p>0: 正向 ETR, 高电平有效或上升沿有效。 1: 反正 ETR, 低电平有效或下降沿有效。</p>
ECM2EN	Bit 14	R/W	<p>使能外部时钟模式 2</p> <p>该位使能外部时钟模式 2</p> <p>0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的有效边沿计数。</p> <p>注意:</p> <ol style="list-style-type: none"> 1. 设置 ECM2EN 位与选择外部时钟模式 1 且 TI 与 ETRF 相连接 (SMODS=111 和 TSSEL=111) 具有相同的效果。 2. 可同时使用外部时钟模式 2 与下列从模式: 复位模式, 门控模式和触发模式。在这种情况下, TI 不能与 ETRF 相连接 (TSSEL 不能设置为 111)。 3. 如果外部时钟模式 1 和外部时钟模式 2 同时使能, 外部时钟输入为 ETRF。
ETPSEL	Bit 13-12	R/W	<p>选择外部触发分频</p> <p>外部触发信号 ETRP 频率最多为 AD16C4TnCLK 频率的 1/4。可使能预分频器来减小 ETRP 频率。该位有效用于输入高速外部时钟的情况。</p> <p>00: 预分频器关闭 01: ETRP 频率 2 分频 10: ETRP 频率 4 分频 11: ETRP 频率 8 分频</p>
ETFLT	Bit 11-8	R/W	<p>选择外部触发滤波采样时钟</p> <p>该位定义了 ETRP 信号的采样频率和数字滤波器的滤波长度。</p> <p>数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿。</p> <p>0000: 无滤波器, 采样频率为 fDTS 0001: fSAMPLING = fINT_CLK, N = 2</p>

			<p>0010: fSAMPLING = fINT_CLK, N = 4 0011: fSAMPLING = fINT_CLK, N = 8 0100: fSAMPLING = fDTS / 2, N = 6 0101: fSAMPLING = fDTS / 2, N = 8 0110: fSAMPLING = fDTS / 4, N = 6 0111: fSAMPLING = fDTS / 4, N = 8 1000: fSAMPLING = fDTS / 8, N = 6 1001: fSAMPLING = fDTS / 8, N = 8 1010: fSAMPLING = fDTS / 16, N = 5 1011: fSAMPLING = fDTS / 16, N = 6 1100: fSAMPLING = fDTS / 16, N = 8 1101: fSAMPLING = fDTS / 32, N = 5 1110: fSAMPLING = fDTS / 32, N = 6 1111: fSAMPLING = fDTS / 32, N = 8 注意: 当 ETFLT[3: 0] = 1, 2 or 3 时, 公式中的 fDTS 由 INT_CLK 取代。</p>
MSCFG	Bit 7	R/W	<p>主/从模式配置 0: 无动作 1: 延迟触发输入 (In) 上的事件来允许当前计时器和其从器件之间的同步。该设置有效用于使用单个外部事件来同步多个计时器。</p>
TSSEL	Bit 6-4	R/W	<p>选择从模式触发源 该位用来选择不同的触发输入来同步计数器。 000: 内部触发 0 (IT0) 001: 内部触发 1 (IT1) 010: 内部触发 2 (IT2) 011: 内部触发 3 (IT3) 100: I1 双边沿边沿检出 (I1F_ED) 101: I1 滤波输入 1 110: I2 滤波输入 2 111: 外部触发输入 注意: 为了避免错误边沿检测, 该位在不使用时 (SMODS=000) 才能改变。</p>
OCCS	Bit 3	R/W	<p>输出通道清除源选择 0: OCCISS 选择的内部通道 1: ETFP</p>
SMODS	Bit 2-0	R/W	<p>选择从模式功能 当选择外部信号, 触发信号 TI 的有效边沿与外部输入的极性有关系 (详见输入控制寄存器和控制寄存器描述) 000: 禁止从模式—如果 CNTEN = '1', 则预分频器直接由内部时钟计数。 001 编码器模式 1—计数器向上/向下计数 I2 边沿, 取决于 I1 电平。</p>

		<p>010 : 编码器模式 2 -计数器向上/向下计数 I1 边沿检出边沿, 取决于 I2 边沿检出电平</p> <p>011 : 编码器模式 3 -计数器向上/向下计数 I1 边沿检出和 I2 边沿检出边沿, 取决于另一个输入的电平。</p> <p>100 : 复位模式 -选中的触发输入的上升沿重新初始化计数器, 生成寄存器的更新</p> <p>101 : 门控模式 -当触发输入 TI 为高电平, 计数器时钟使能。一旦触发变为低电平, 计数器停止计数 (并非复位)。计数器的启动和停止均受控制。</p> <p>110 : 触发模式 -计数器在触发信号 TI 的上升沿处启动 (不复位)。仅寄存器的启动受控制。</p> <p>111 : 外部时钟源 1 -计数器在 TI 的上升沿计数</p> <p>注意: 如果 I1 双边沿检出被选为触发输入 (TSSEL='100'), 不能使用门控模式。I1 每一次转换, I1 双边沿检出就会输出 1 个脉冲, 而门控模式则是检查触发信号的电平。</p> <p>注意: 在发生来自自主计时器的接收事件之前, 从计时器的时钟必须先使能, 且在接收来自自主计时器的触发过程中, 从计数器时钟不能即时更改。</p>
--	--	--

17.5.2.4 中断使能寄存器 (AD16C4T_IER)

中断使能寄存器 (AD16C4T_IER)																															
偏移地址: 00C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								BRKIT	TRGIT	COMIT	CC4IT	CC3IT	CC2IT	CC1IT	UIT

Reserved	Bit31-8	-	保留, 必须保持复位值。
BRKIT	Bit7	W	使能刹车中断 0: 无效 1: 使能
TRGIT	Bit6	W	使能触发中断 0: 无效 1: 使能
COMIT	Bit5	W	使能 COM 中断 0: 无效 1: 使能
CC4IT	Bit4	W	使能捕获/比较 4 中断 0: 无效 1: 使能
CC3IT	Bit3	W	使能捕获/比较 3 中断 0: 无效 1: 使能
CC2IT	Bit2	W	使能捕获/比较 2 中断 0: 无效 1: 使能
CC1IT	Bit1	W	使能捕获/比较 1 中断 0: 无效 1: 使能
UIT	Bit0	W	使能更新事件中中断 0: 无效 1: 使能

17.5.2.5 中断禁止寄存器 (AD16C4T_IDR)

中断禁止寄存器 (AD16C4T_IDR)																															
偏移地址: 0010 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							BRKI	TRGI	COMI	CC4I	CC3I	CC2I	CC1I	UI	

Reserved	Bit 31-8	-	保留, 必须保持复位值。
BRKI	Bit 7	W	禁止刹车中断 0: 无效 1: 禁止
TRGI	Bit 6	W	禁止触发中断 0: 无效 1: 禁止
COMI	Bit 5	W	禁止 COM 中断 0: 无效 1: 禁止
CC4I	Bit 4	W	禁止捕获/比较 4 中断 0: 无效 1: 禁止
CC3I	Bit 3	W	禁止捕获/比较 3 中断 0: 无效 1: 禁止
CC2I	Bit 2	W	禁止捕获/比较 2 中断 0: 无效 1: 禁止
CC1I	Bit 1	W	禁止捕获/比较 1 中断 0: 无效 1: 禁止
UI	Bit 0	W	禁止更新中断 0: 无效 1: 禁止

17.5.2.6 中断有效状态寄存器 (AD16C4T_IVS)

中断有效状态寄存器 (AD16C4T_IVS)																															
偏移地址: 0014 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								BKI	TRGI	COMI	CC4I	CC3I	CC2I	CC1I	UEI

Reserved	Bit 31-8	N/A	保留
BKI	Bit 7	R	刹车中断状态 0: 禁止 1: 使能
TRGI	Bit 6	R	触发中断状态 0: 禁止 1: 使能
COMI	Bit 5	R	CM中断状态 0: 禁止 1: 使能
CC4I	Bit 4	R	通道4捕获/比较中断状态 0: 禁止 1: 使能
CC3I	Bit 3	R	通道3捕获/比较中断状态 0: 禁止 1: 使能
CC2I	Bit 2	R	通道2捕获/比较中断状态 0: 禁止 1: 使能
CC1I	Bit 1	R	通道1捕获/比较中断状态 0: 禁止 1: 使能
UEI	Bit 0	R	更新事件中断状态 0: 禁止 1: 使能

17.5.2.7 原始中断标志寄存器 (AD16C4T_RIF)

原始中断标志寄存器 (AD16C4T_RIF)																															
偏移地址: 0018 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CH4OVIF	CH3OVIF	CH2OVIF	CH1OVIF	Reserved	BRKIF	TRGIF	COMIF	CH4IF	CH3IF	CH2IF	CH1IF	UEVTIF							

Reserved	Bit 31-13	N/A	保留
CH4OVIF	Bit 12	R	<p>通道 4 捕获溢出中断标志 仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 AD16C4T_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CH4IF 标志为置起时, 捕获计数器值至 AD16C4T_CCVAL1 寄存器</p>
CH3OVIF	Bit 11	R	<p>通道 3 捕获溢出中断标志 仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 AD16C4T_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CH3IF 标志为置起时, 捕获计数器值至 AD16C4T_CCVAL1 寄存器</p>
CH2OVIF	Bit 10	R	<p>通道 2 捕获溢出中断标志 仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 AD16C4T_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CH2IF 标志为置起时, 捕获计数器值至 AD16C4T_CCVAL1 寄存器</p>
CH1OVIF	Bit 9	R	<p>通道 1 捕获溢出中断标志 仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 AD16C4T_ICR 写 1 来清除原始中断。 0: 未检测到捕获溢出 1: 当 CH1IF 标志为置起时, 捕获计数器值至 AD16C4T_CCVAL1 寄存器</p>
Reserved	Bit 8	R	保留, 必须保持复位值。
BRKIF	Bit 7	R	<p>刹车中断标志 如果刹车中断使能, 一旦刹车输入有效, 该标志</p>

			<p>位由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生刹车事件。</p> <p>1: 在刹车输入上检测到有效电平</p>
TRGIF	Bit 6	R	<p>触发中断标志</p> <p>如果触发中断使能，当从模式控制器在门控模式以外的所有模式下使能，发生触发事件时（TI 上检测到有效边沿），该标志位被硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生触发事件</p> <p>1: 触发中断被挂起</p>
COMIF	Bit 5	R	<p>COM 中断标志</p> <p>如果 COM 中断使能，当发生 COM 事件（当捕获/比较控制位 CCnE, CCnNE, CHnOMOD 更新后），该标志位被硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生 COM 事件</p> <p>1: COM 中断被挂起</p>
CH4IF	Bit 4	R	<p>通道 4 捕获/比较中断标志</p> <p>参考 CH1IF 描述</p>
CH3IF	Bit 3	R	<p>通道 4 捕获/比较中断标志</p> <p>参考 CH1IF 描述</p>
CH2IF	Bit 2	R	<p>通道 2 捕获/比较中断标志</p> <p>参考 CH1IF 描述</p>
CH1IF	Bit 1	R	<p>通道 1 捕获/比较 1 中断标志</p> <p>如果 CC1 通道配置为输出：</p> <p>如果中断使能，除去中心对齐模式的情况（参考 AD16C4T_CON1 寄存器中 CMSEL 的描述），当计数值与比较值匹配，该标志位由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 不匹配。</p> <p>1: AD16C4T_COUNT 计数值与 AD16C4T_CCVAL1 值匹配。当 AD16C4T_CCVAL1 寄存器值大于 AD16C4T_AR 值，发生计数器上溢时（递增模式和递增/递减模式）或下溢时（递减模式），CH1IF 为被置起。</p> <p>如果 CC1 通道配置为输入：</p> <p>发生捕获时，该位由硬件置起。该位可通过软件或者读取 AD16C4T_CCVAL1 寄存器来清零。</p> <p>0: 未发生输入捕获</p> <p>1: 计数值捕获至 AD16C4T_CCVAL1 寄存器（I1 上检测到与选中极性匹配的边沿）</p>
UEVTIF	Bit 0	R	<p>更新事件中断标志</p> <p>如果更新中断使能，当发生更新事件，该标志位</p>

		<p>由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生更新。</p> <p>1: 更新中断被挂起。当寄存器更新时, 该位被硬件置起:</p> <ul style="list-style-type: none"> -当重复计数器值发生上溢或者下溢(若重复计数器=0, 则更新)和当 AD16C4T_CON1 寄存器中 DISUE=0 -当使用 AD16C4T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时, 如果 AD16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0 -当 CNT 由触发事件来重新初始化, 如果 AD16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0
--	--	---

17.5.2.8 中断标志屏蔽寄存器 (AD16C4T_IFM)

中断标志屏蔽寄存器 (AD16C4T_IFM)																															
偏移地址: 001C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							BRKIM	TRGI	COMI	CH4CCI	CH3CCI	CH2CCI	CH1CCI	UEI	

Reserved	Bit 31-8	-	保留
BRKIM	Bit 7	R	<p>刹车中断标志屏蔽</p> <p>如果刹车中断使能，一旦刹车输入有效，该标志位由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生刹车事件。</p> <p>1: 在刹车输入上检测到有效电平</p>
TRGI	Bit 6	R	<p>屏蔽触发中断标志</p> <p>如果触发中断使能，当从模式控制器在门控模式以外的所有模式下使能，发生触发事件时 (TI 上检测到有效边沿)，该标志位被硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生触发事件</p> <p>1: 触发中断被挂起</p>
COMI	Bit 5	R	<p>屏蔽 COM 中断标志</p> <p>如果 COM 中断使能，当发生 COM 事件 (当捕获/比较控制位 CCnEN, CCnNE, CHnOMOD 更新后)，该标志位被硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生 COM 事件</p> <p>1: COM 中断被挂起</p>
CH4CCI	Bit 4	R	<p>屏蔽通道 4 捕获/比较中断标志</p> <p>参考 CH1CCI 描述</p>
CH3CCI	Bit 3	R	<p>屏蔽通道 3 捕获/比较中断标志</p> <p>参考 CH1CCI 描述</p>
CH2CCI	Bit 2	R	<p>屏蔽通道 2 捕获/比较中断标志</p> <p>参考 CH1CCI 描述</p>
CH1CCI	Bit 1	R	<p>屏蔽通道 1 捕获/比较中断标志</p> <p>如果通道 1 配置为输出：</p> <p>如果中断使能，除去中心对齐模式的情况 (参考 AD16C4T_CON1 寄存器中 CMSEL 的描述)，当计数值与比较值匹配，该标志位由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 不匹配。</p> <p>1: AD16C4T_COUNT 计数值与</p>

			<p>AD16C4T_CCVAL1 值匹配。当 AD16C4T_CCVAL1 寄存器值大于 AD16C4T_AR 值，发生计数器上溢时（递增模式和递增/递减模式）或下溢时（递减模式），CH1IF 为被置起。</p> <p>如果通道配置为输入：</p> <p>发生捕获时，该位由硬件置起。该位可通过软件或者读取 AD16C4T_CCVAL1 寄存器来清零。</p> <p>0：未发生输入捕获</p> <p>1：计数值捕获至 AD16C4T_CCVAL1 寄存器（I1 上检测到与选中极性匹配的边沿）</p>
UEI	Bit 0	R	<p>屏蔽更新事件中断标志如果更新中断使能，当发生更新事件，该标志位由硬件置起。对 AD16C4T_ICR 写 1 来清除原始中断。</p> <p>0：未发生更新。</p> <p>1：更新中断被挂起。当寄存器更新时，该位被硬件置起：</p> <ul style="list-style-type: none"> -当重复计数器值发生上溢或者下溢（若重复计数器=0，则更新）和当 AD16C4T_CON1 寄存器中 DISUE=0 -当使用 AD16C4T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时，如果 AD16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0 -当 CNT 由触发事件来重新初始化，如果 AD16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0

17.5.2.9 中断清零寄存器 (AD16C4T_ICR)

中断清零寄存器 (AD16C4T_ICR)																															
偏移地址: 0020 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								BRKIC	TRGIC	COMIC	CH4CCIC	CH3CCIC	CH2CCIC	CH1CCIC	UEIC

Reserved	Bit 31-8	N/A	保留
BRKIC	Bit 7	W1C	清刹车中断 0: 无效 1: 清零 (AD16C4T_RIF)
TRGIC	Bit 6	W1C	清触发中断 0: 无效 1: 清零 (AD16C4T_RIF)
COMIC	Bit 5	W1C	清 COM 中断 0: 无效 1: 清零 (AD16C4T_RIF)
CH4CCIC	Bit 4	W1C	清通道 4 捕获/比较中断 参考 CC1IC 描述
CH3CCIC	Bit 3	W1C	清通道 3 捕获/比较中断 参考 CC1IC 描述
CH2CCIC	Bit 2	W1C	清通道捕获/比较中断 参考 CC1IC 描述
CH1CCIC	Bit 1	W1C	清通道捕获/比较中断 0: 无效 1: 清零 (AD16C4T_RIF)
UEIC	Bit 0	W1C	清更新事件中断 0: 无效 1: 清零 (AD16C4T_RIF)

17.5.2.10 事件生成寄存器 (AD16C4T_SGE)

事件生成寄存器 (AD16C4T_SGE)																															
偏移地址: 0024 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								SGBRK	SGTRG	SGCOM	SGCC4E	SGCC3E	SGCC2E	SGCC1E	SGU

Reserved	Bit 31-8	N/A	保留
SGBRK	Bit 7	W	<p>软件生成刹车事件 该位由软件设置来生成刹车事件，可由硬件自动清零。 0: 无动作 1: 产生刹车事件。GOEN 清零，BRKIF 标志位置起，产生相关中断或 DMA 传输。</p>
SGTRG	Bit 6	W	<p>软件生成触发事件 该位由软件设置来生成触发事件，可由硬件自动清零。 0: 无动作 1: AD16C4T_RIF 寄存器中的 TRGIF 被置起，产生相关中断或 DMA 传输</p>
SGCOM	Bit 5	W	<p>软件生成换相事件捕获 该位由软件设置，由硬件自动清零。 0: 无动作 1: 当 CCPCEN 被置 1, 则可更新 CCnEN, CCnNE 和 CHnOMOD 注意: 该位只有用作于通道时才有互补输出</p>
SGCC4E	Bit 4	W	<p>软件触发通道 4 捕获/比较事件 参考 SGCC1E 描述</p>
SGCC3E	Bit 3	W	<p>软件触发通道 3 捕获/比较事件 参考 SGCC1E 描述</p>
SGCC2E	Bit 2	W	<p>软件触发通道 2 捕获/比较事件 参考 SGCC1E 描述</p>
SGCC1E	Bit 1	W	<p>软件触发通道 1 捕获/比较事件 该位由软件设置来生成事件，可由硬件自动清零。 0: 无动作 1: 通道 1 上产生捕获/比较事件： 如果通道 1 配置为输出： CH1IF 标志位被置起，产生相应中断或 DMA 请求发送。 如果通道 1 配置为输入：</p>

			<p>当前计数值捕获至 AD16C4T_CCVAL1 寄存器。CH1IF 标志位被置起, 产生相应中断或 DMA 请求发送。CH1OVIF 标志位置起 (如果 CH1IF 标志位已为高电平)。</p>
SGU	Bit 0	W	<p>软件触发更新事件 该位由软件设置, 可由硬件自动清零。 0: 无动作 1: 重新初始化计数器, 更新寄存器。注意, 预分频器也会被清零 (但预分频比不会受到影响)。如果使用中心对齐模式或者 DIRSEL=0 (递增), 则计数器将清零; 否则如果 DIRSEL=1 (递减), 则将使用自动重载入值。</p>

17.5.2.11 通道捕获/比较模式寄存器 1 (AD16C4T_CHMR1)

◆ 输出比较模式

通道捕获模式寄存器 1 (AD16C4T_CHMR1)																															
偏移地址: 0028 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CH2OCLREN	CH2OMOD	CH2OPEN	CH2OFEN	CC2SEL	CH1OCLREN	CH1OMOD	CH1OPREN	CH1OHSEN	CC1SSEL						

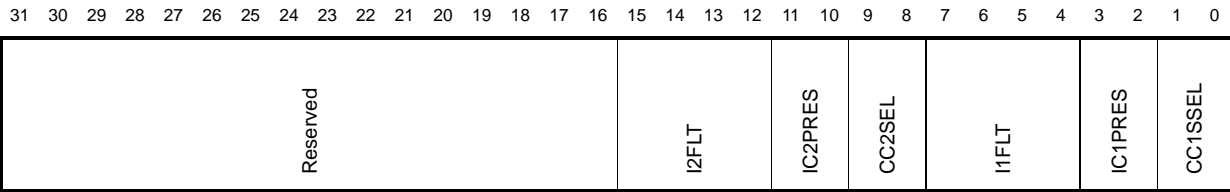
Reserved	Bit 31-16	-	保留
CH2OCLREN	Bit 15	R/W	使能通道 2 输出比较清零 参考 CH1OCLREN 描述
CH2OMOD	Bit 14-12	R/W	通道 2 输出比较模式 参考 CH1OMOD 描述
CH2OPEN	Bit 11	R/W	使能通道 2 输出比较预载入 参考 CH1OPREN 描述
CH2OFEN	Bit 10	R/W	使能通道 2 输出比较高速模式 参考 CH1OHSEN 描述
CC2SEL	Bit 9-8	R/W	选择通道 2 输出比较 该位定义了通道方向以及使用的输入源 00: 通道配置为输出 01: 通道配置为输入, 捕获源为 I2 10: 通道配置为输入, 捕获源为 I1 11: 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出。 仅当内部触发输入通过 TSSEL 位 (AD16C4T_SMCON 寄存器) 选择时, 该模式才能工作。 注意: 当通道为关闭状态时 (AD16C4T_CCEP 中 CC2EN = '0'), CC2SEL 为只写。
CH1OCLREN	Bit 7	R/W	使能通道 1 输出比较清零 0: 通道 1 比较输出不会受到 ETRF 输入影响 1: 当 ETRF 输入上检测到高电平时, 通道 1 比较输出将被清零
CH1OMOD	Bit 6-4	R/W	输出比较 1 模式 该位定义了输出参考信号通道 1 比较输出的行为。 通道 1 比较输出为高有效, CH1O 和 CH1ON 的有效电平由 CC1POL 和 CC1NPOL 位决定。 000: 冻结-输出比较寄存器 AD16C4T_CCVAL1 寄存器和 AD16C4T_COUNT 计数器之间的比较

		<p>对输出无效。</p> <p>001：发生匹配时设置通道 1 为有效电平-当计数器 AD16C4T_COUNT 与捕获/比较寄存器 1 AD16C4T_CCVAL1 发生匹配后，通道 1 比较输出信号强制为高电平。</p> <p>010：发生匹配时设置通道 1 为无效电平。当计数器 AD16C4T_COUNT 与捕获/比较寄存器 1 AD16C4T_CCVAL1 发生匹配后，通道 1 比较输出信号强制为低电平。</p> <p>011：翻转 -当 AD16C4T_COUNT=AD16C4T_CCVAL1，通道 1 比较输出发生翻转。</p> <p>100：强制为无效电平 - 通道 1 比较输出强制为低电平。</p> <p>101：强制为有效电平- 通道 1 比较输出强制为高电平。</p> <p>110：PWM 模式 1 -在递增模式下，当 AD16C4T_COUNT<AD16C4T_CCVAL1，通道 1 为有效电平，否则，通道 1 为无效电平。在递减模式下，当 AD16C4T_COUNT>AD16C4T_CCVAL1，通道 1 为无效电平（通道 1 比较输出='0'），否则通道 1 为有效电平（通道 1 比较输出='1'）。</p> <p>111：PWM 模式 2 -在递增模式下，当 AD16C4T_COUNT<AD16C4T_CCVAL1，通道 1 为无效电平，否则，通道 1 有无效电平。在递减模式下，当 AD16C4T_COUNT>AD16C4T_CCVAL1，通道 1 为有效电平，否则通道 1 为无效电平。</p> <p>注意：</p> <p>1：当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 3，且 CC1SSEL=00（通道为输出模式），该位将不能更改。</p> <p>2：在 PWM 模式 1 和 2 中，仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式，比较输出电平才会更改。</p> <p>3：对于有互补输出的通道，该位设置为预载值。如果 AD16C4T_CON2 寄存器中的 CCPEN 位设置为 1，则只有当 COM 事件发生时，CH1OMOD 有效位才会设置为预载值中新的值。</p>
CH1OPREN	Bit 3	<p>R/W</p> <p>输出比较 1 预载使能</p> <p>0：AD16C4T_CCVAL1 的预载寄存器禁止。AD16C4T_CCVAL1 在任何时候都可写，新写入的值将立刻生效。</p>

			<p>1：AD16C4T_CCVAL1 的预载寄存器使能。读/写操作可访问预载寄存器。每当发生一次更新事件，AD16C4T_CCVAL1 预载入值将会被填入有效寄存器。</p> <p>注意：</p> <p>1：当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 3，且 CC1SSEL=00（通道为输出模式），该位将不能更改。</p> <p>2：仅在单脉冲模式下（AD16C4T_CON1 寄存器中的 SPMEN 设置为 1），PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。</p>
CH1OHSEN	Bit 2	R/W	<p>使能通道 1 输出比较高速</p> <p>该位用来加速在通道 1 输出上的输入触发事件的效应。</p> <p>0：当触发开启，通道 1 运作正常取决于计数器和 CCRV1 的值。当触发输入上发现边沿时，至少需要 5 个时钟周期来激活通道 1 输出。</p> <p>1：触发输入上的有效沿类似于通道 1 输出上的比较匹配。设置 OC 为 1 用来比较电平，采样触发输入和激活通道 1 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式，CH1OHSEN 才会起作用。</p>
CC1SSEL	Bit 1-0	R/W	<p>捕获/比较 1 选择</p> <p>该位定义了通道方向和使用的输入源</p> <p>00：通道配置为输出</p> <p>01：通道配置为输入，捕获源为 I1</p> <p>10：通道配置为输入，捕获源为 I2</p> <p>11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出。</p> <p>只有当内部触发输入是通过 TSSEL 位（AD16C4T_SMCON 寄存器）选择时，该模式才运行。</p> <p>注意：当通道关闭（AD16C4T_CCEP 寄存器中的 CC1EN = '0'），CC1SSEL 为只写。</p>

◆ 输入捕获模式

通道捕获模式寄存器 1 (AD16C4T_CHMR1)
偏移地址: 0028 _H
复位值: 00000000_00000000_00000000_00000000 _B



Reserved	Bit 31-16	-	保留
I2FLT	Bit 15-12	R/W	通道 2 输入捕获滤波器 参考 I1FLT 描述
IC2PRES	Bit 11-10	R/W	通道 2 输入捕获预分频器 参考 IC1PRES 描述
CC2SEL	Bit 9-8	R/W	选择通道 2 输入捕获源 该位定义了通道方向和使用的输入源 00: 通道配置为输出 01: 通道配置为输入, 捕获源为 I2 10: 通道配置为输入, 捕获源为 I1 11: 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出 只有当内部触发输入是通过 TSSEL 位 (AD16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (AD16C4T_CCEP 寄存器中的 CC2EN = '0'), CC2SEL 为只写。
I1FLT	Bit 7-4	R/W	I1 滤波器 该位定义了 I1 输入的采样频率和数字滤波器的长度。 数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿: 0000: 无滤波器, 采样频率为 fDTS 0001: fSAMPLING = fINT_CLK, N = 2 0010: fSAMPLING = fINT_CLK, N = 4 0011: fSAMPLING = fINT_CLK, N = 8 0100: fSAMPLING = fDTS / 2, N = 6 0101: fSAMPLING = fDTS / 2, N = 8 0110: fSAMPLING = fDTS / 4, N = 6 0111: fSAMPLING = fDTS / 4, N = 8 1000: fSAMPLING = fDTS / 8, N = 6 1001: fSAMPLING = fDTS / 8, N = 8 1010: fSAMPLING = fDTS / 16, N = 5

			<p>1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 6$ 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 8$ 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 5$ 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 6$ 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 8$ 注意: 当 I1FLT [3:0] = 1, 2 或 3 时, 公式中的 f_{DTS} 由 INT_CLK 取代</p>
IC1PRES	Bit3-2	R/W	<p>捕获比较器 1 输入预分频器 该位定义了作用在 CC1 输入 (I1) 上的预分频比。当 CC1EN='0' (AD16C4T_CCEP 寄存器), 预分频器将复位。 00: 无预分频器。每当捕获输入上检测到边沿时, 发生捕获动作。 01: 每发生 2 次事件, 执行一次捕获 10: 每发生 4 次事件, 执行一次捕获 11: 每发生 8 次事件, 执行一次捕获</p>
CC1SSEL	Bit 1-0	R/W	<p>捕获比较器 1 输入源选择 该位定义了通道方向和使用的输入源 00: CC1 通道配置为输出 01: CC1 通道配置为输入, 捕获源为 I1 10: CC1 通道配置为输入, 捕获源为 I2 11: CC1 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出。 只有当内部触发输入是通过 TSSEL 位 (AD16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (AD16C4T_CCEP 寄存器中的 CC1EN = '0'), CC1SSEL 为只写。</p>

17.5.2.12 通道捕获/比较模式寄存器 2 (AD16C4T_CHMR2)

◆ 输出比较模式

通道捕获模式寄存器 2 (AD16C4T_CHMR2)																															
偏移地址: 002C _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CH4OCLREN	CH4OMOD	CH4OPEN	CH4OHSEN	CC4SSEL	CH3OCLREN	CH3OMOD	CH3OPEN	CH3OFEN	CC3SSEL						

Reserved	Bit 31-16	-	保留
CH4OCLREN	Bit 15	R/W	通道 4 输出比较清零使能 参考 CH1OCLREN 描述
CH4OMOD	Bit 14-12	R/W	通道 4 输出比较模式 参考 CH1OMOD 描述
CH4OPEN	Bit 11	R/W	通道 4 输出比较预载入使能 参考 CH1OPREN 描述
CH4OHSEN	Bit 10	R/W	通道 4 输出比较高速使能 参考 CH1OHSEN 描述
CC4SSEL	Bit 9-8	R/W	通道 4 输出比较选择 该位定义了通道方向以及使用的输入源 00: 通道配置为输出 01: 通道配置为输入, 捕获源为 I4 10: 通道配置为输入, 捕获源为 I3 11: 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出 仅当内部触发输入通过 TSSEL 位 (AD16C4T_SMCON 寄存器) 选择时, 该模式才能工作。 注意: 当通道为关闭状态时 (AD16C4T_CCEP 中 CC4EN = '0'), CC4SSEL 为只写。
CH3OCLREN	Bit 7	R/W	通道 3 输出比较清零使能 参考 CH1OCLREN 描述
CH3OMOD	Bit 6-4	R/W	通道 3 输出比较模式 参考 CH1OMOD 描述
CH3OPEN	Bit 3	R/W	通道 3 输出比较预载使能 参考 CH1OPREN 描述
CH3OFEN	Bit 2	R/W	通道 3 输出比较高速使能 参考 CH1OHSEN 描述
CC3SSEL	Bit 1-0	R/W	通道 3 捕获/比较选择 该位定义了通道方向和使用的输入源

			<p>00：通道配置为输出</p> <p>01：通道配置为输入，捕获源为 I3</p> <p>10：通道配置为输入，捕获源为 I4</p> <p>11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出。</p> <p>只有当内部触发输入是通过 TSSEL 位（AD16C4T_SMCON 寄存器）选择时，该模式才运行。</p> <p>注意：当通道关闭（AD16C4T_CCEP 寄存器中的 CC3EN = '0'），CC3SSEL 为只写</p>
--	--	--	--

◆ 输入捕获模式

通道捕获模式寄存器 2 (AD16C4T_CHMR2)																															
偏移地址: 002C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																I4FLT	IC4PRES	CC4SSEL	I3FLT	IC3PRES	CC3SSEL										

Reserved	Bit 31-16	-	保留
I4FLT	Bit 15-12	R/W	通道 4 输入捕获滤波器 参考 I1FLT 描述
IC4PRES	Bit 11-10	R/W	通道 4 输入捕获预分频器 参考 IC1PRES 描述
CC4SSEL	Bit 9-8	R/W	选择通道 4 输入捕获源 该位定义了通道和使用的输入的方向。 00：通道配置为 输出 01：通道配置为输入，捕获源为 I4 10：通道配置为输入，捕获源为 I3 11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出 只有当内部触发输入是通过 TSSEL 位（ AD16C4T_SMCON 寄存器）选择时，该模式才运行 注意：当通道关闭（ AD16C4T_CCEP 寄存器中的 CC4EN = '0' ）， CC4SSEL 为只写。
I3FLT	Bit 7-4	R/W	通道 3 输入捕获滤波器 参考 I1FLT 描述
IC3PRES	Bit3-2	R/W	通道 3 输入捕获预分频器 参考 IC1PRES 描述

<p>CC3SSEL</p>	<p>Bit 1-0</p>	<p>R/W</p>	<p>选择通道 3 输入捕获源 该位定义了通道方向和使用的输入源 00: 通道配置为输出 01: 通道配置为输入, 捕获源为 I3 10: 通道配置为输入, 捕获源为 I4 11: 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出 只有当内部触发输入是通过 TSSEL 位 (AD16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (AD16C4T_CCEP 寄存器中的 CC3EN = '0'), CC3SSEL 为只写。</p>
----------------	----------------	------------	---

17.5.2.13 捕获/比较使能极性寄存器 (AD16C4T_CCEP)

通道捕获/比较使能极性寄存器 (AD16C4T_CCEP)																															
偏移地址: 0030 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CC4POL	CC4EN	CC3NPOL	CC3NE	CC3POL	CC3EN	CC2NPOL	CC2NE	CC2POL	CC2EN	CC1NPOL	CC1NE	CC1POL	CC1EN					

Reserved	Bit 31-14	-	保留
CC4POL	Bit 13	R/W	通道 4 捕获/比较输出极性 参考 CC1POL 描述
CC4EN	Bit 12	R/W	使能通道 4 捕获/比较输出 参考 CC1EN 描述
CC3NPOL	Bit 11	R/W	通道 3 捕获/比较互补输出极性 参考 CC1NPOL 描述
CC3NE	Bit 10	R/W	使能通道 3 捕获/比较互补输出 参考 CC1NE 描述
CC3POL	Bit 9	R/W	通道 3 捕获/比较输出极性 参考 CC1POL 描述
CC3EN	Bit 8	R/W	使能通道 3 捕获/比较输出 参考 CC1EN 描述
CC2NPOL	Bit 7	R/W	通道 2 捕获/比较 2 互补输出极性 参考 CC1NPOL 描述
CC2NE	Bit 6	R/W	使能通道 2 捕获/比较互补输出 参考 CC1NE 描述
CC2POL	Bit 5	R/W	通道 2 捕获/比较输出极性 参考 CC1POL 描述
CC2EN	Bit 4	R/W	使能通道 2 捕获/比较输出 参考 CC1EN 描述
CC1NPOL	Bit 3	R/W	通道 1 捕获/比较互补输出极性 通道配置为输出： 0：CH1ON 高有效。 1：CH1ON 低有效。 通道配置为输入： 该位需和 CC1POL 一起使用来定义输入边沿的极性。参考 CC1POL 描述。 注意：对于有互补输出的通道，该位设置为预载值。如果 AD16C4T_CON2 寄存器中的 CCPEN 位设置为 1，则只有当 COM 事件发生时，CC1NPOL 有效位才会设置为预载值中新的值。 注意：当 AD16C4T_BDCFG 寄存器中的

			LOCKLVL 位被设置为锁定级别 2 或 3, 且 CC1SSEL=00(通道为输出模式), 该位将不可写。
CC1NE	Bit 2	R/W	<p>使能通道 1 捕获/比较互补输出</p> <p>0: 关闭 - CH1ON 无效。CH1ON 电平取决于 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 的功能</p> <p>1: 开启 - CH1ON 为对应输出引脚上的输出信号, 由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 决定。</p> <p>注意: 对于有互补输出的通道, 该位设置为预载值。如果 AD16C4T_CON2 寄存器中的 CCPEN 位设置为 1, 则只有当 COM 事件发生时, CC1NE 有效位才会设置为预载值中新的值</p>
CC1POL	Bit 1	R/W	<p>通道 1 捕获/比较输出极性</p> <p>通道配置为输出:</p> <p>0: CH1O 高有效</p> <p>1: CH1O 低有效</p> <p>通道配置为输入:</p> <p>CC1NPOL/CC1POL 为触发和捕获操作选择 I1 边沿检出和 I2 边沿检出的有效极性。</p> <p>00: 正向/上升沿</p> <p>电路对 In 边沿检出的上升沿敏感 (在复位, 外部时钟或触发模式下, 进行捕获或触发), In 边沿检出不反向 (门控模式或编码器模式下, 进行触发)。</p> <p>01: 反向/下降沿</p> <p>电路对 In 边沿检出的下降沿敏感 (在复位, 外部时钟或触发模式下, 进行捕获或触发), In 边沿检出反向 (门控模式或编码器模式下, 进行触发)。</p> <p>10: 预留, 不要使用该配置。</p> <p>11: 正向/上升沿和下降沿</p> <p>电路对 In 边沿检出的上升沿和下降沿均敏感 (在复位, 外部时钟或触发模式下, 进行捕获或触发), In 边沿检出不反向 (门控模式或编码器模式下, 进行触发)。</p> <p>注意: 对于有互补输出的通道, 该位设置为预载值。如果 AD16C4T_CON2 寄存器中的 CCPEN 位设置为 1, 则只有当 COM 事件发生时, CC1POL 有效位才会设置为预载值中新的值。</p> <p>注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位被设置为锁定级别 2 或 3, 且 CC1SSEL=00(通道为输出模式), 该位将不可写。</p>
CC1EN	Bit 0	R/W	<p>使能通道 1 捕获/比较输出</p> <p>通道配置为输出:</p> <p>0: 关闭 - CH1O 无效。CH1ON 电平取决于</p>

			<p>GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 的功能</p> <p>1: 开启 - CH1O 为对应输出引脚上的输出信号, 由 GOEN, OFFSSI, OFFSSR, OISS1, OISS1N 和 CC1EN 决定</p> <p>通道配置为输入:</p> <p>该位决定了计数值是否能捕获到输入捕获/比较寄存器 1 (AD16C4T_CCVAL1)。</p> <p>0: 禁止捕获。</p> <p>1: 使能捕获。</p> <p>注意: 对于有互补输出的通道, 该位设置为预载值。如果 AD16C4T_CON2 寄存器中的 CCPCEN 位设置为 1, 则只有当 COM 事件发生时, CC1EN 有效位才会设置为预载值中新的值。</p>
--	--	--	---

17.5.2.14 计数器 (AD16C4T_COUNT)

计数器寄存器 (AD16C4T_COUNT)																															
偏移地址: 0034 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CNTV															

Reserved	Bit 31-16	-	保留
CNTV	Bit 15-0	R/W	计数值

17.5.2.15 时钟预分频器 (AD16C4T_PRES)

时钟预分频器寄存器 (AD16C4T_PRES)																															
偏移地址: 0038 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PSCV															

Reserved	Bit 31-16	-	保留
PSCV	Bit 15-0	R/W	<p>时钟预分频器值</p> <p>计数器时钟频率 (CK_CNT) = fCK_PSC / (PSCV[15: 0] + 1)。</p> <p>每发生一次更新事件 (包括当计数器由 AD16C4T_SGE 寄存器中的 SGU 位清零或当配置为复位模式时, 通过触发控制器清零), PSCV 包含的值需填入到有效的预分频寄存器内。</p>

17.5.2.16 计数器自动装载寄存器 (AD16C4T_AR)

计数器自动装载寄存器 (AD16C4T_AR)																															
偏移地址: 003C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ARRV															

Reserved	Bit 31-16	-	保留
ARRV	Bit 15-0	R/W	<p>计数器自动装载值</p> <p>ARRV 中的值将被载入实际的自动重载寄存器中。当自动重载值为空, 计数器被屏蔽。</p>

17.5.2.17 重复计数寄存器 (AD16C4T_REPAR)

重复计数寄存器 (AD16C4T_REPAR)																															
偏移地址: 0040 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								REPV							

Reserved	Bit 31-8	-	保留
REPV	Bit 7-0	R/W	<p>重复计数值</p> <p>当预载寄存器使能，该位允许用户设置比较寄存器的更新率（例如：预载到有效寄存器的周期性传输），同样也可以设置更新中断生成率。每次当 REPV_CNT 的相关递减计数器递减至 0，会产生更新事件，会从 REPV 值重新计数。因为只有当发生重复更新事件 U_RC 时，REPV_CNT 才会重新载入 REPV 值，所以只有在发生下一次重复更新事件时，写入 AD16C4T_REPAR 寄存器的值才会生效。</p> <p>即，在 PWM 模式下，(REPV+1) 相当于：</p> <ul style="list-style-type: none"> -在边沿对齐模式下，(REPV+1) 对应的是 PWM 的周期数 -在中心对齐模式下，(REPV+1) 对应的是 1/2 PWM 的周期数

17.5.2.18 通道捕获/比较寄存器 1 (AD16C4T_CCVAL1)

通道捕获/比较寄存器 1 (AD16C4T_CCVAL1)																															
偏移地址: 0044 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCR1															

Reserved	Bit 31-16	-	保留
CCR1	Bit 15-0	R/W	<p>捕获/比较值 1</p> <p>如果通道 CCn 配置为输出： CCR1n 中的值将被载入实际的捕获/比较寄存器中（预载值）。</p> <p>如果在 AD16C4T_CCMRn 寄存器中的预载功能没有选中，CCR1n 中的值将被永久载入；否则，每当发生更新事件，预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与 AD16C4T_COUNT 中的值进行比较，并在 OCn 上输出。</p> <p>如果通道 CCn 配置为输入： CCR1n 为由上一个输入捕获事件 (ICn) 传输的计数值。</p>

17.5.2.19 通道捕获/比较寄存器 2 (AD16C4T_CCVAL2)

通道捕获/比较寄存器 2 (AD16C4T_CCVAL2)																															
偏移地址: 0048 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCR2															

Reserved	Bit 31-16	-	保留
CCR2	Bit15-0	R/W	<p>通道捕获/比较值 2</p> <p>参考 CCR1 描述</p>

17.5.2.20 通道捕获/比较寄存器 3 (AD16C4T_CCVAL3)

捕获/比较寄存器 3 (AD16C4T_CCVAL3)																															
偏移地址: 004C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV3															

Reserved	Bit 31-16	-	保留
CCRV3	Bit 15-0	R/W	捕获/比较值 3 参考 CCRV1 描述

17.5.2.21 通道捕获/比较寄存器 4 (AD16C4T_CCVAL4)

通道捕获/比较寄存器 4 (AD16C4T_CCVAL4)																															
偏移地址: 0050 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV4															

Reserved	Bit 31-16	-	保留
CCRV4	Bit 15-0	R/W	通道捕获/比较值 4 参考 CCRV1 描述

17.5.2.22 刹车和死区配置寄存器 (AD16C4T_BDCFG)

刹车和死区配置寄存器 (AD16C4T_BDCFG)																																						
偏移地址: 0054 _H																																						
复位值: 00000000_00000000_00000000_00000000 _B																																						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Reserved																GOEN	AOEN	BRKP	BRKEN	OFFSSR	OFFSSI	LOCKLVL	DT															

Reserved	Bit 31-16	-	保留
GOEN	Bit 15	RW	<p>通道主要输出使能</p> <p>一旦刹车输入有效, 该位会由硬件异步清零。该位可由软件置 1 或自动置 1, 取决于 AOEN 位。该位仅作用于配置为输出的通道。</p> <p>0: OC 和 OCN 输出禁止或强制为空闲状态。</p> <p>1: 如果 OC 和 OCN 各自的使能位都置 1 (AD16C4T_CCEP 寄存器中的 CCnEN, CCnNE), 则 OC 和 OCN 输出使能。</p>
AOEN	Bit 14	RW	<p>通道自动输出使能</p> <p>0: GOEN 仅可由软件置位</p> <p>1: 在下一个更新事件发生时 (如果刹车输入无效), GOEN 可由软件或自动置位。</p> <p>注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改。</p>
BRKP	Bit 13	RW	<p>选择通道刹车极性</p> <p>0: 刹车输入 BRKP 为低有效</p> <p>1: 刹车输入 BRKP 为高有效</p> <p>注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改</p> <p>注意: 任何对该位的写操作都要延时 1 个 APB 时钟周期后才变为有效。</p>
BRKEN	Bit 12	RW	<p>使能刹车</p> <p>0: 刹车输入 (BRKP 和 CCS 时钟失效事件) 禁止</p> <p>1: 刹车输入 (BRKP 和 CCS 时钟失效事件) 使能</p> <p>注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 则该位不可更改</p> <p>注意: 任何对该位的写操作都要延时 1 个 APB 时</p>

			钟周期后才变为有效。
OFFSSR	Bit 11	R/W	<p>运行模式下的无效状态选择位</p> <p>该位使用于，当 GOEN=1 时，被配置为输出并使用互补输出的通道。如果计时器中没有使用互补输出，则 OFFSSR 不使用。</p> <p>0：无效状态时，OC/OCN 输出禁止（OC/OCN 使能输出信号=0）。</p> <p>1：无效状态时，当 CCnEN=1 或 CCnNE=1 时，便使能 OC/OCN 输出并将其设为无效电平。（OC/OCN 使能输出信号=1）</p> <p>注意：当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2，则该位不可更改。</p>
OFFSSI	Bit 10	R/W	<p>空闲模式下的空闲状态选择位</p> <p>该位使用于，当 GOEN=0 时，被配置为输出通道</p> <p>0：无效状态时，OC/OCN 输出禁止（OC/OCN 使能输出信号=0）。</p> <p>1：无效状态时，当 CCnEN=1 或 CCnNE=1，便将 OC/OCN 输出首先强制为其空闲电平。（OC/OCN 使能输出信号=1）</p> <p>注意：当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 2，则该位不可更改。</p>
LOCKLVL	Bit 9-8	R/W	<p>锁定级别配置</p> <p>针对软件错误，该位提供写保护。</p> <p>00：锁定关闭-不提供写保护</p> <p>01：锁定级别 1 = AD16C4T_BDCFG 寄存器中的 DT，AD16C4T_CON2 寄存器中的 OISSx 和 OISSxN，和 AD16C4T_BDCFG 寄存器中的 BRKEN/BRKP/AOEN 不再可写。</p> <p>10：锁定级别 2 = 锁定级别 1 + CC 极性位（AD16C4T_CCEP 寄存器中的 CCnPOL/CCnNPOL，只要相关通道由 CCnSSEL 配置为输出）以及 OFFSSR 和 OFFSSI 都不再可写。</p> <p>11：锁定级别 3 = 锁定级别 2 + CC 控制位（AD16C4T_CHMRn 寄存器中的 CHnOMOD 和 CHnOPEN，只要相关通道由 CCnSSEL 配置为输出）都不再可写。</p> <p>注意：锁定配置为仅在复位后可写。一旦 AD16C4T_BDCFG 已写，其设置内容在下一个复位前都处于冻结状态。</p>
DT	Bit 7-0	R/W	<p>死区延时设置值</p> <p>该位定义了互补输出之间插入的死区时间。DT 对</p>

		<p>应的就是该时间段。</p> <p>DT[7: 5]=0xx => DT=DT[7:0]x t_{dtg}, 式中 t_{dtg}=t_{DTS}</p> <p>DT[7: 5]=10x => DT= (64+DT[5:0]) x t_{dtg}, 式中 t_{dtg}=2x t_{DTS}</p> <p>DT[7: 5]=110 => DT= (32+DT[4:0]) x t_{dtg}, 式中 t_{dtg}=8x t_{DTS}</p> <p>DT[7: 5]=111 => DT= (32+DT[4: 0]) x t_{dtg}, 式中 t_{dtg}=16x t_{DTS}</p> <p>注意: 当 AD16C4T_BDCFG 寄存器中的 LOCKLVL 位已被设置为锁定级别 1, 2 或 3, 则该位不可更改</p>
--	--	--

17. 5. 2. 23 DMA使能寄存器 (AD16C4T_DMAEN)

DMA 使能寄存器 (AD16C4T_DMAEN)																															
偏移地址: 058 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								TDS	COMDS	CC4DS	CC3DS	CC2DS	CC1DS	UDS	

Reserved	Bit 31-7	-	保留, 必须保持为复位值
TDS	Bit 6	R/W	触发DMA请求使能 0: DMA请求禁止 1: DMA请求使能
COMDS	Bit 5	R/W	COM DMA访问使能 0: DMA请求禁止 1: DMA请求使能
CC4DS	Bit 4	R/W	捕获/比较值 4 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC3DS	Bit 3	R/W	捕获/比较值 3 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC2DS	Bit 2	R/W	捕获/比较值 2 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC1DS	Bit 1	R/W	捕获/比较值 1 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
UDS	Bit 0	R/W	更新 DMA 请求使能 0: DMA 请求禁止 1: DMA 请求使能

第18章 通用定时器（GP16C4T0~2）

18.1 概述

芯片内部共有 3 个 16 位通用定时器（GP16C4T0~2），GP16C4T 包含一个 16 位自动重载计数器，该计数器由可配置的预分频器驱动。

通用定时器（GP16C4T）的用途广泛，可测量信号脉冲长度（输入捕获）或输出脉冲波形（比较输出、PWM）。

18.2 特性

- ◆ 16 位递增，递减，递增/递减自动加载计数器
- ◆ 16 位可编程预分频器，可对计数器工作时钟进行 1 到 65536 间的任意分频
- ◆ 多达四个独立信道
 - ◇ 输入捕获
 - ◇ 输出比较
 - ◇ PWM 输出（边沿与中心对齐模式）
 - ◇ 单脉冲输出模式
- ◆ 同步电路用于外部信号控制定时器及内部互联多个定时器
- ◆ 以下事件中产生中断/DMA：
 - ◇ 更新事件：计数器上溢/下溢，计数器初始化（通过软件或内/外部触发）
 - ◇ 触发事件（计数器起始、停止、初始化或内/外触发计数）
 - ◇ 输入捕获
 - ◇ 输出比较
- ◆ 支持增量（正交）编码及霍尔电路进行定位
- ◆ 触发输入可对外部时钟管理

18.3 结构框图

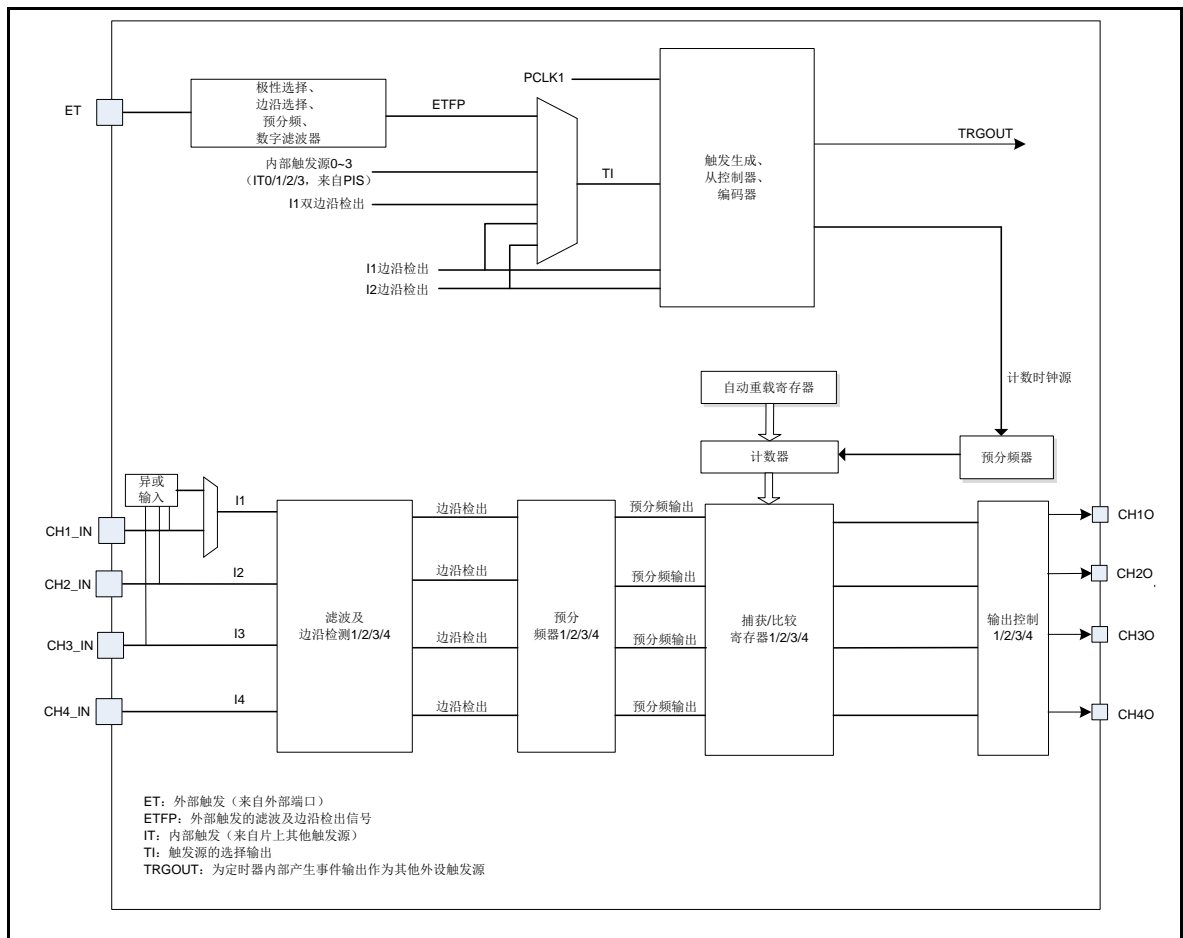


图 18-1 通用定时器结构框图

18.4 功能描述

18.4.1 预分频器

定时器包含一个 16-bit 的计数器 (GP16C4T_COUNT)，计数时钟由预分频寄存器 (GP16C4T_PRES) 进行分频。计数周期由自动重载计数器 (GP16C4T_AR) 设定。

自动重载寄存器 (GP16C4T_AR) 是一个可缓存的寄存器。当 GP16C4T_CON1 寄存器的 ARPEN 位复位时，GP16C4T_AR 寄存器重载功能失效，GP16C4T_AR 就是有效寄存器；ARPEN 置位时，GP16C4T_AR 寄存器具有重载功能，产生更新事件 (UEV) 时，加载值 (GP16C4T_AR 寄存器值) 更新到影子寄存器后才生效。

当 GP16C4T_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢 (或递减下溢) 时会产生更新事件 (UEV)。同样，软件方式也可产生更新事件。GP16C4T_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 GP16C4T_PRES 寄存器值+1 次分频。由于 GP16C4T_PRES 是一个可重载寄存器，因此，定时器工作时可对该寄存器进行修改，修改值在下次更新事件 (UEV) 后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

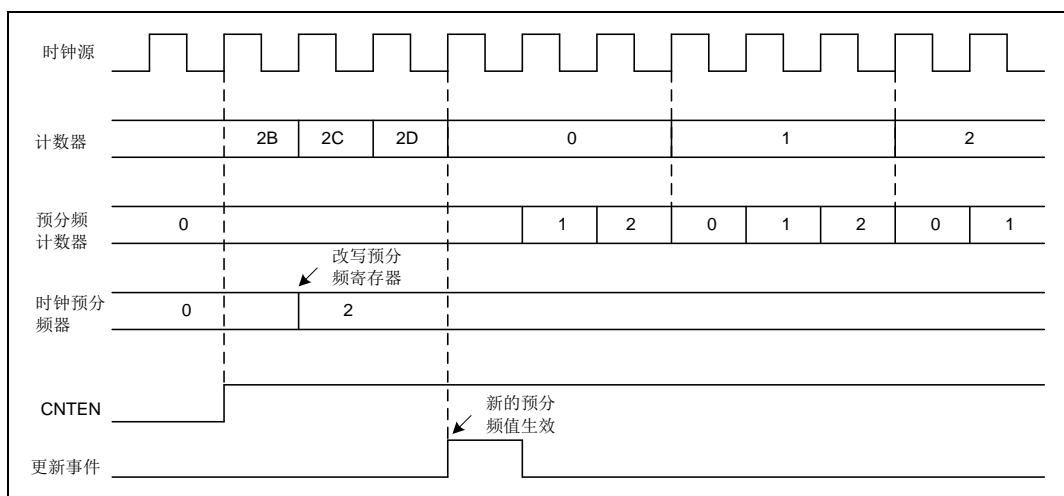


图 18-2 预分频值计数时序图

18.4.2 时钟源

计数器工作时钟可以选择内部时钟 (INT_CLK)、外部时钟源 1 (I1、I2、I3、I4)、外部时钟源 2 (ET)，内部触发输入 (IT1、IT2、IT3、IT4)

18.4.2.1 内部时钟源 (INT_CLK)

若从模式控制器被关闭 (GP16C4T_SMCON 寄存器内, SMODS= "000"), 则 CNTEN, GP16C4T_CON1.DIRSEL 与 GP16C4T_SGE.SGU 位为实际控制位, 这些位只能软件修改 (SGU 位除外, 仍由硬件自动清除)。一旦 CNTEN 位被写为'1', 预分频器就由内部 INT_CLK 提供时钟。

下图给出了通常模式下控制电路和递增计数的情况, 没有分频。

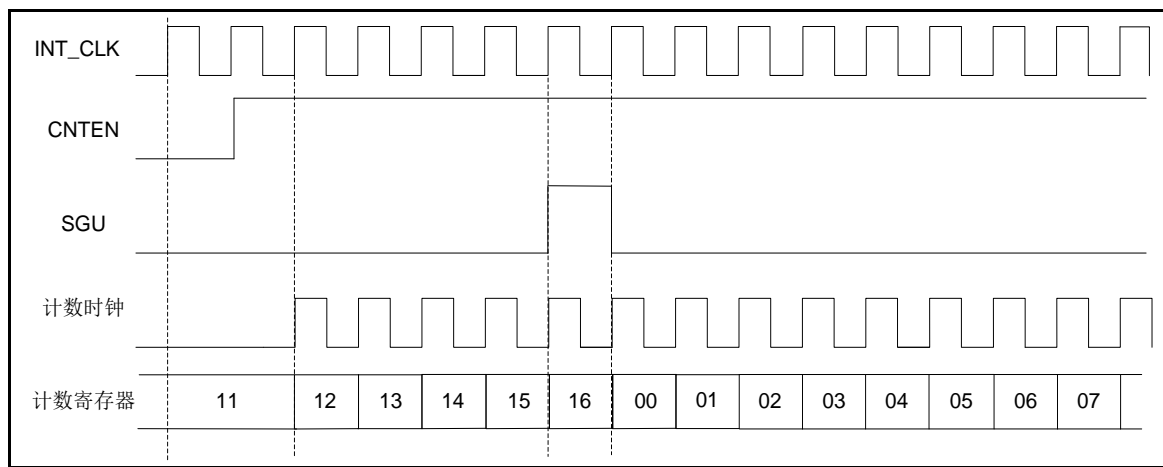


图 18-3 采用内部时钟计数

18.4.2.2 外部时钟源 1

GP16C4T_SMCON 寄存器的 SMODS= "111"时, 可选择外部时钟源 1。计数器可根据选定的上升沿或下降沿计数。

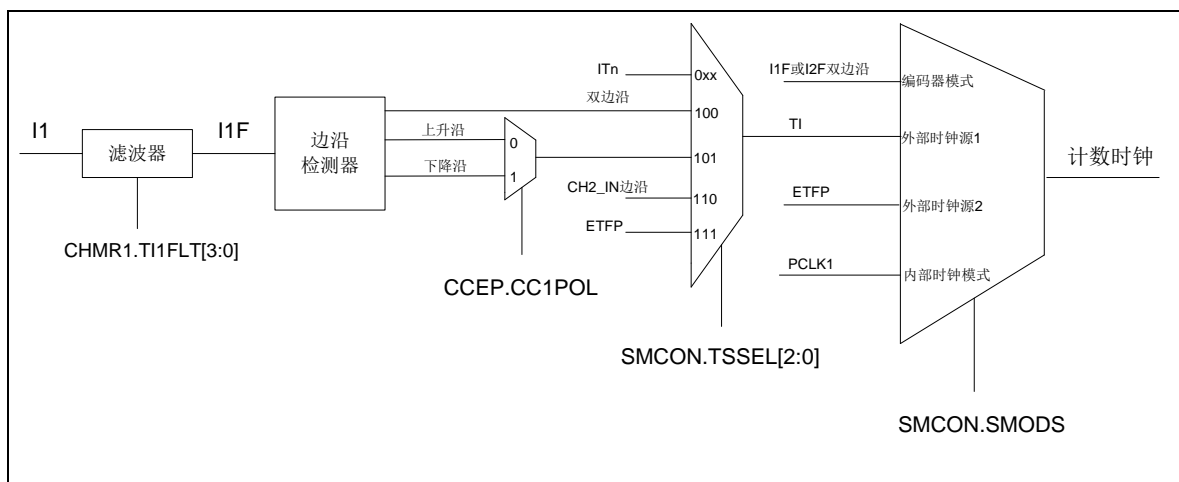


图 18-4 I1 外部时钟连接

配置计数器为外部时钟源 1, 步骤如下:

6. GP16C4T_SMCON 寄存器中 SMODS = "111", 配置定时器外部时钟模式 1。
7. 设置 GP16C4T_SMCON 寄存器中的 TSSEL 选择外部时钟源。
8. 如外部时钟源为 I1, 可配置 GP16C4T_CHMR1 寄存器 CC1SSEL = "01", 配置通道 1 检测 I1 输入的上升沿; 设置 GP16C4T_CCEP 寄存器中 CC1POL = '0', 选择极性为上升沿。
9. 写 GP16C4T_CHMR1 寄存器的 I1FLT[3: 0]位, 配置输入滤波器时间 (若没有滤波器需求, 维持 I1FLT = "0000")。
10. GP16C4T_CON1 寄存器中 CNTEN = '1', 使能计数器。

当 I1 上出现一次上升沿时, 计数器计数一次且 TRGIF 标志位置位。

18.4.2.3 外部时钟源 2

置位 GP16C4T_SMCON 寄存器的 ECM2EN 位选定外部时钟源 2。

计数器可对外部触发输入 ET 进行上升沿或下降沿计数。

下图给出了外部输入输入模块的概况。

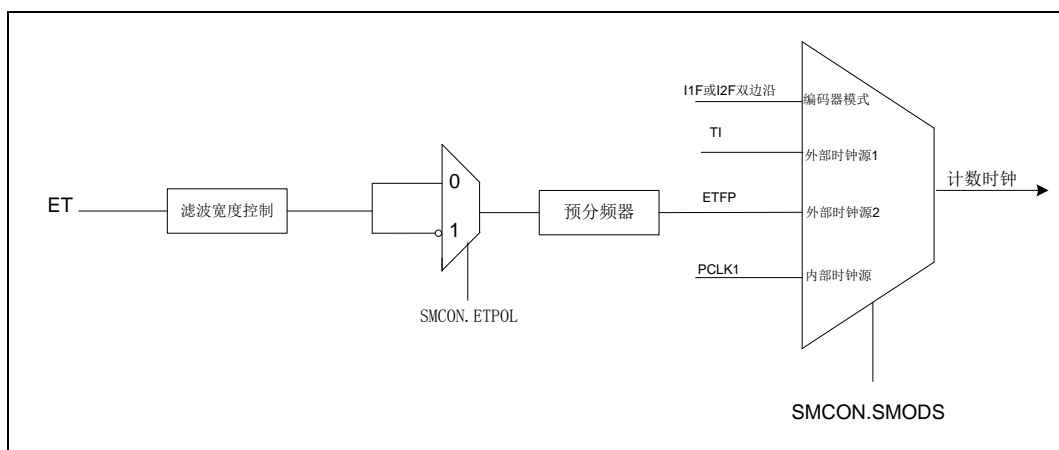


图 18-5 外部触发输入模块

配置计数器为外部时钟源 2, 配置过程如下:

6. 设置 GP16C4T_SMCON 寄存器的 ETFLT<3: 0>, 配置输入滤波时间。
7. 设置 GP16C4T_SMCON 寄存器中 ETPSEL<1: 0>, 设置预分频器。
8. 设置 GP16C4T_SMCON 寄存器中 ETPOL, 检测 ET 引脚上升沿或下降沿。
9. 设置 GP16C4T_SMCON 寄存器中 ECM2EN = '1', 使能外部时钟模式 2。
10. 设置 GP16C4T_CON1 寄存器的 CNTEN = '1', 使能计数器。

计数器每两个上升沿计一次数。

18.4.2.4 内部触发输入 (ITn)

当 GP16C4T_SMCON 寄存器的 SMODS="111", 选定内部触发模式。计数器根据选定的内部输入端的上升或下降沿计数。

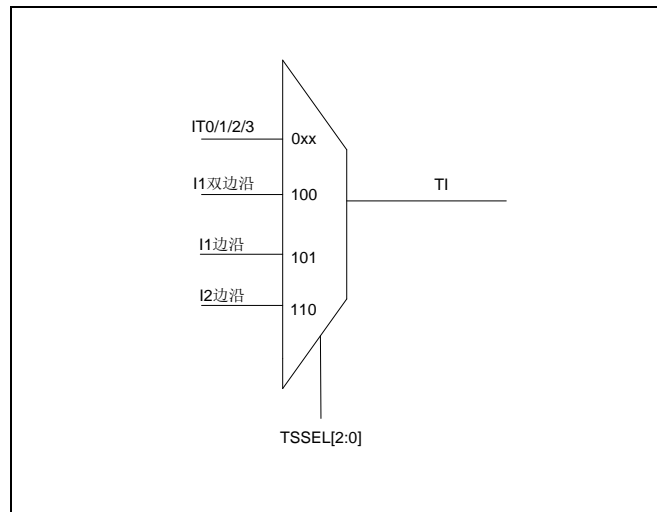


图 18-6 ITn 外部时钟连接

配置计数器在 ITn 输入端的上升沿递增计数，步骤如下：

4. GP16C4T_SMCON 寄存器中 SMODS = "111", 配置外部时钟模式 1。
5. GP16C4T_SMCON 寄存器的 TSSEL = "0xx", 选定 ITn 作为触发输入源。
6. GP16C4T_CON1 寄存器的 CNTEN = '1', 使能计数器。

ITn 产生上升沿时，计数器计数一次。ITn 上升沿与实际时钟间的延时，取决于 ITn 输入的再同步电路，一般为 2~3 个定时器模块时钟周期。

18.4.3 计数器模式

18.4.3.1 递增计数模式

在递增模式下，当 GP16C4T_REPAR 寄存器值为 0 时，计数器从 0 开始递增，直至 GP16C4T_AR 寄存器值；然后从 0 重新开始计数并产生一个更新事件（UEV）。当 GP16C4T_REPAR 寄存器不为 0 时，则在 GP16C4T_REPAR+1 次计数后产生更新事件。

当有更新事件（UEV）产生时，预装载寄存器会更新到影子寄存器，更新标志位（GP16C4T_RIF 寄存器中的 UEVTIF 位）置位（取决于 UERSEL 位）：

- ◇ 更新 GP16C4T_REPAR 寄存器的值到影子寄存器
- ◇ 更新 GP16C4T_AR 寄存器的值到影子寄存器
- ◇ 更新 GP16C4T_PRES 寄存器的值到影子寄存器

下图为 GP16C4T_REPAR=0x0，GP16C4T_AR = 0x16，预分频设为 2 分频时的计数器时序。

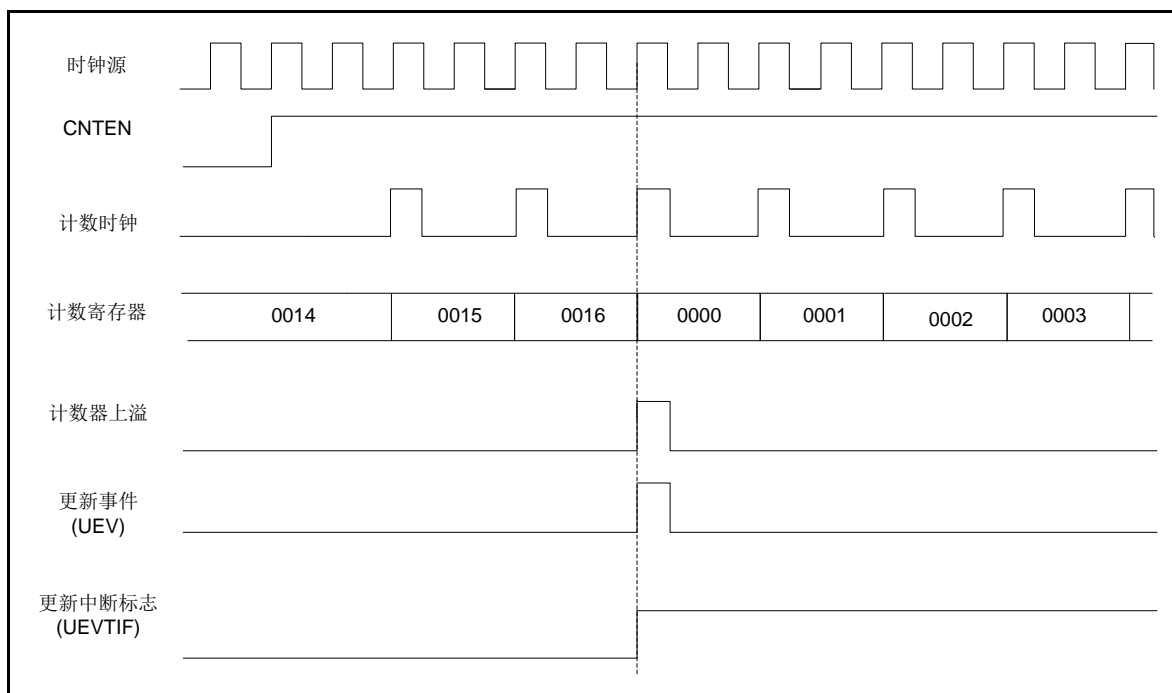


图 18-7 计数器递增计数时序图

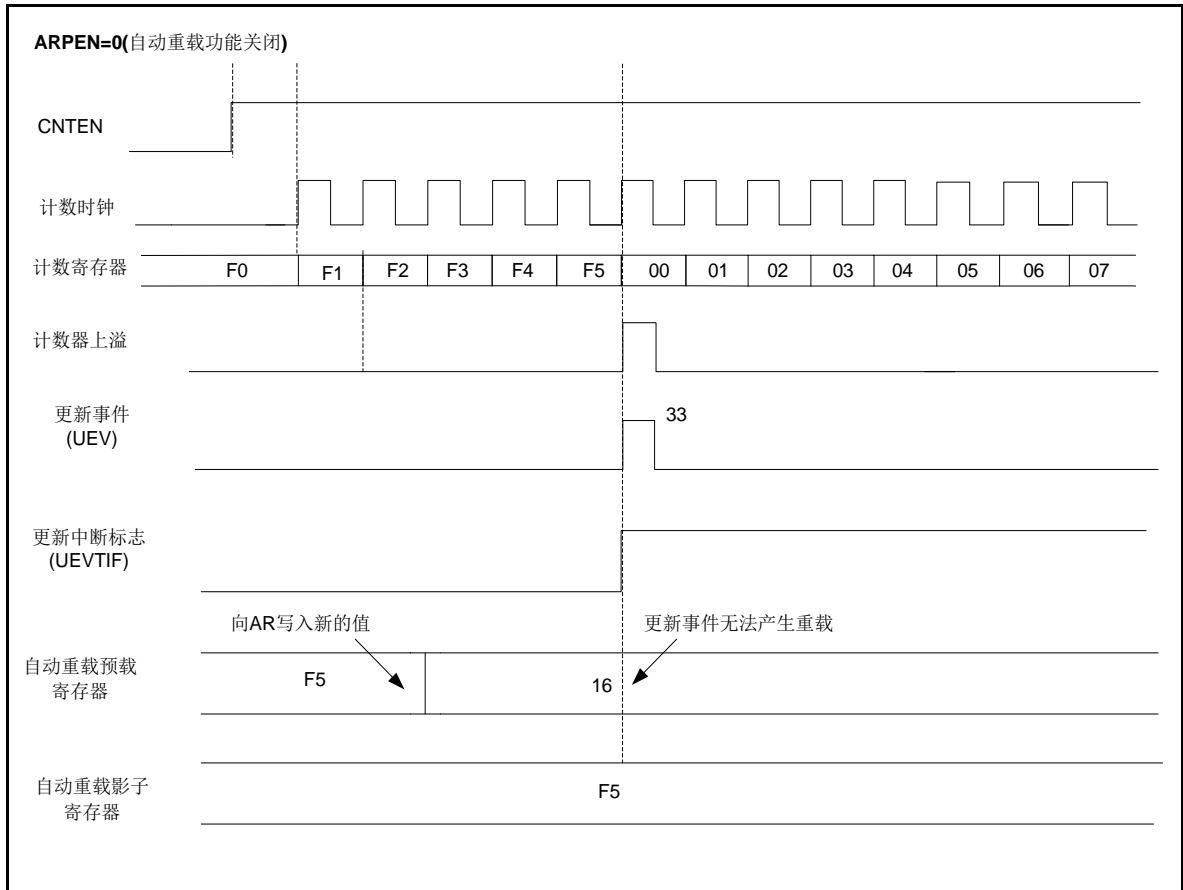


图 18-8 当 ARPEN=0 时计数器时序图

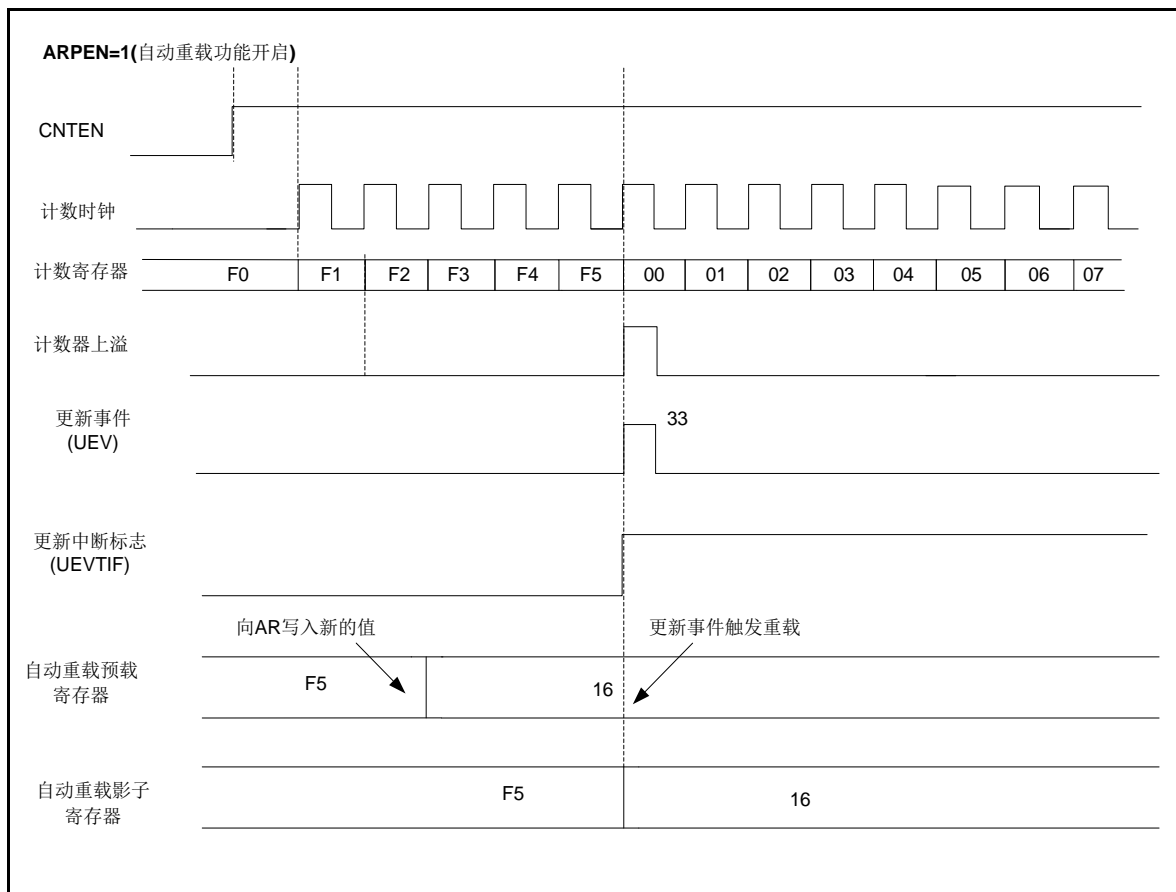


图 18-9 当 ARPEN=1 时计数器时序图

18.4.3.2 递减计数模式

在递减模式下，当 GP16C4T_REPAR 寄存器值为 0 时，计数器从 GP16C4T_AR 寄存器值开始递减至 0；然后重复递减并产生更新事件（UEV）。当 GP16C4T_REPAR 寄存器不为 0 时，则在 GP16C4T_REPAR+1 次后产生更新事件。

置位 GP16C4T_SGE 寄存器中的 SGU 位（通过软件或使用从机模式控制器）同样会产生更新事件。

当有更新事件（UEV）产生时，预载寄存器值会更新到影子寄存器，更新标志位（GP16C4T_RIF 寄存器中的 UEVTIF 位）置位（取决于 UERSEL 位）。

下图为 GP16C4T_REPAR=0x0，GP16C4T_AR = 0x27，预分频设为 1 分频时的计数器时序。

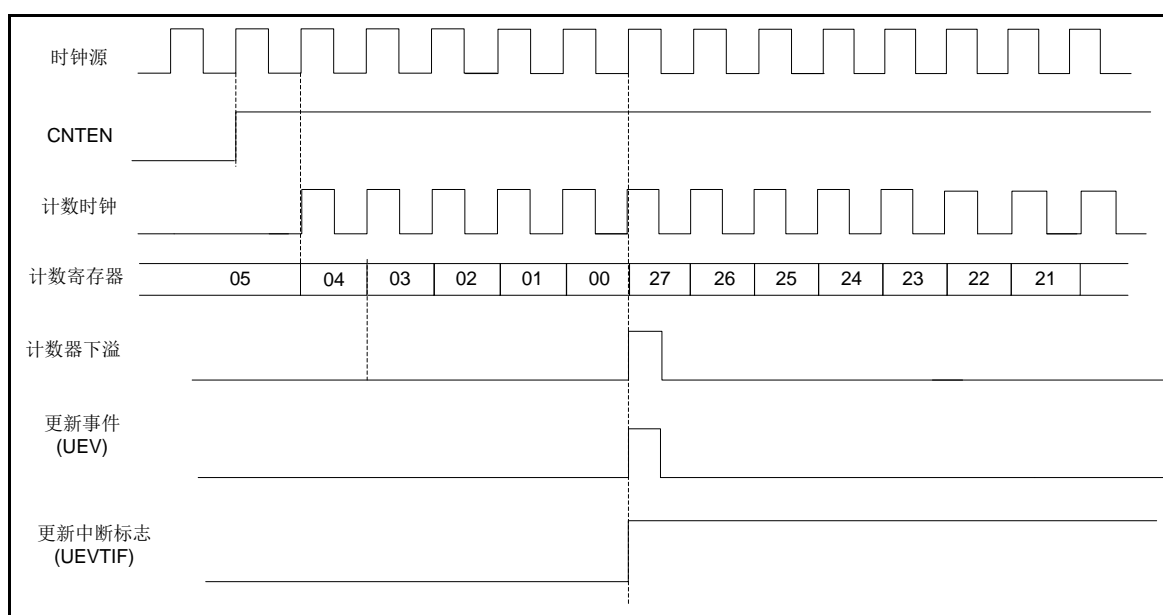


图 18-10 计数器递减计数时序图

18.4.3.3 中心对齐模式

当 GP16C4T_CON1 寄存器的 CMSEL 位的值不等于"00"时,定时器工作在中心对齐模式。定时器配置为中心对齐模式时,计数器先从 0 开始递增至 GP16C4T_AR 寄存器值-1,并产生更新事件 (UEV);在接着计数器从 GP16C4T_AR 寄存器值递减至 1,并产生下溢事件。如此循环计数。计数器递减计数(中心对称模式 1, CMSEL="01")、计数器递增计数(中心对称模式 2, CMSEL="10")、计数器递增和递减计数(中心对称模式 3, CMSEL="11")模式下,当通道配置为输出模式时,其输出比较中断标志位会置位。

在中心对齐模式下, GP16C4T_CON1 寄存器的 DIRSEL 位无法进行写操作,该位由硬件自动更新指示当前计数方向。

计数上溢、下溢或者置位 GP16C4T_EGR 寄存器的 SGU 位(通过软件或使用从模式控制器)都会产生更新事件。因此,计数器和预分频器都会从 0 开始计数。

软件置位 GP16C4T_CON1 寄存器中的 DISUE 位可关闭更新事件 (UEV) 的产生。更新事件 (UEV) 关闭时,可避免向预载寄存器写新值时更新影子寄存器。DISUE 复位之前都不会产生更新事件。而在正常产生更新事件时,计数器仍然从 0 开始,同样预分频计数也是从 0 开始(但预分频值没有改变)。此外,若置位 GP16C4T_CON1 寄存器中的 UERSEL 位(更新请求选择),置位 SGU 位时会产生一次更新事件 (UEV),但 UEVTIF 标志位不会置位(因此,不会触发中断或 DMA 请求)。这就避免了在捕获事件时,清除计数器值时产生更新和捕获中断。

当有更新事件 (UEV) 产生时,预载寄存器值会更新到影子寄存器,更新标志位 (GP16C4T_RIF 寄存器中的 UEVTIF 位)置位(取决于 UERSEL 位)。

注:若更新源为计数上溢,自动重载会在计数器重载前更新。因此,下一周期即为预期值(计数器载入新值)。

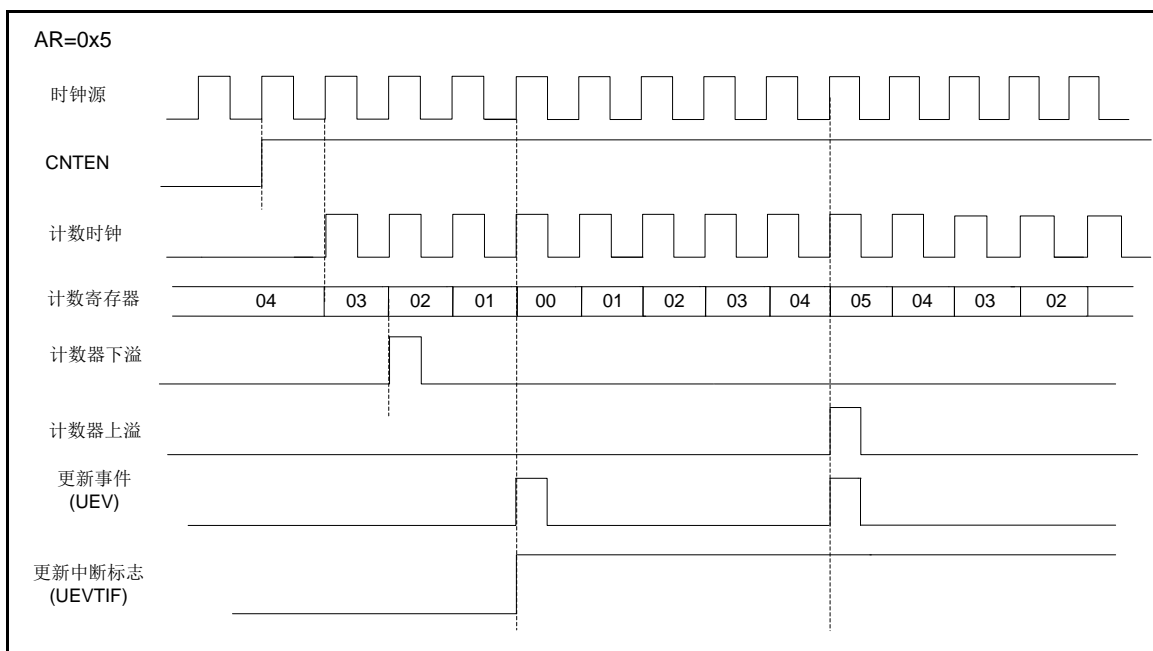


图 18-11 增减计数器时序图

18.4.4 捕获/比较通道

以下各图为捕获/比较通道的概述。

输入电路对 I_n 输入端的信号进行采样，产生一个经过滤波的信号 I_nF 。之后，一个可极性选择的边沿检测器产生 I_n 边沿检出信号，该信号可作为从模式控制器的触发输入或作为捕获控制命令，且信号经过分频后进入捕获寄存器。

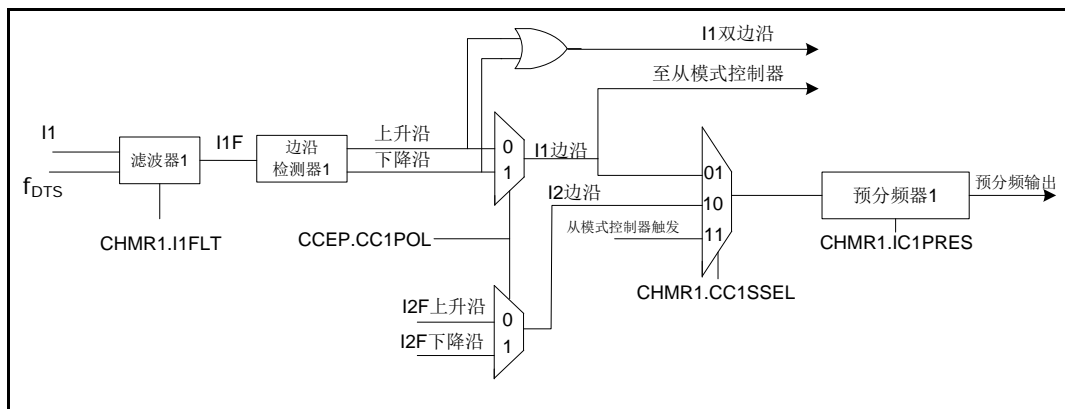


图 18-12 捕获/比较通道

输出部分产生一个中间波形（高有效）作为基准，在输出末端决定最终输出信号的极性。

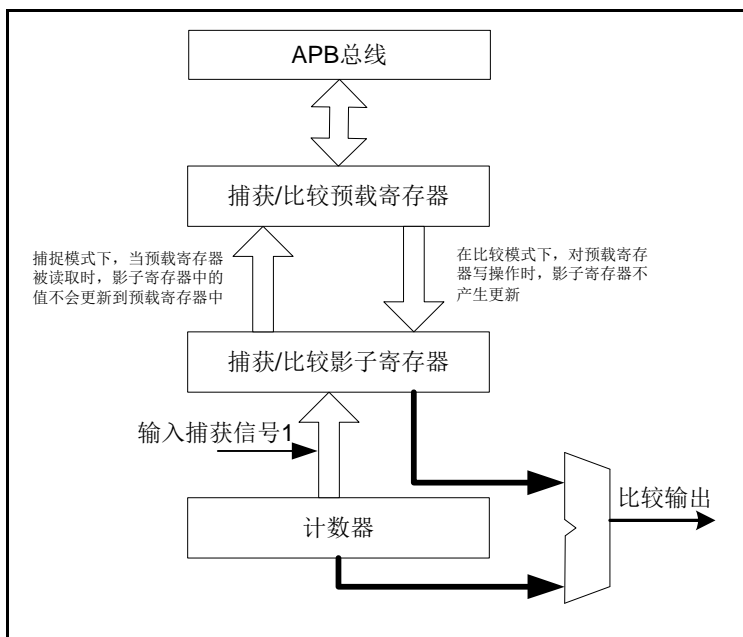


图 18-13 捕获/比较信道 1 主电路

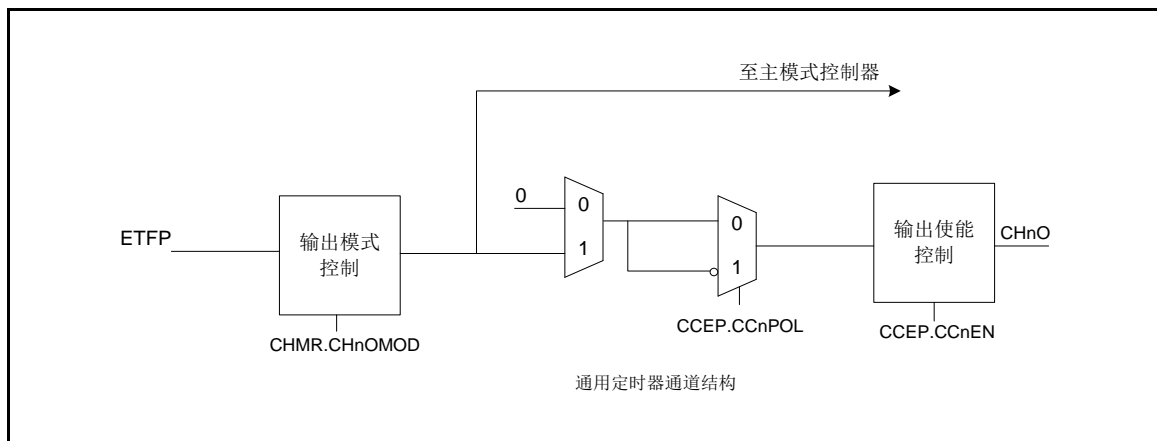


图 18-14 捕获/比较通道的输出阶段

18.4.5 输入捕获模式

在输入捕获模式下，当检测到 In 上相应信号变化时，计数器的值就会被锁存到捕获/比较寄存器（GP16C4T_CCVALn）寄存器中。当捕获发生时，相应的 CHnCCIF 标志位（GP16C4T_RIF）会置位，同时会触发中断或 DMA（如果使能）请求。若发生捕获时，CHnCCIF 标志位已经置位，则过捕获 CHnOVIF 标志位（GP16C4T_RIF）置位。软件写 '0' 或读取 GP16C4T_CCVALn 寄存器中的捕获值都可以复位 CHnCCIF 标志位。对 CHnOVIF 位写 '0' 可清空该标志位。

以下为以 I1 输入上升沿作为捕获输入时的流程：

7. 选择有效输入端：GP16C4T_CCVAL1 必须连接到 I1 输入端，因此需将 GP16C4T_CHMR1 寄存器中的 CC1SSEL 位写 "01"。只要 CC1SSEL 不为 "00"，通道被配置为输入且 GP16C4T_CCVAL1 寄存器为只读。
8. 根据定时器连接的输入信号，配置输入滤波器的持续时间。当输入信号翻转时，前 5 个内部时钟信号是不稳定的，因此必须配置滤波器的时间大于 5 个时钟周期。当 I1 检测到新的电平，连续 8 次采样可确认电平变化有效。
9. 选择 I1 信道的有效边沿变换。GP16C4T_CCEP 寄存器中的 CC1POL 写 '0'（上升沿）。
10. 配置输入预分频器。
11. 置位 GP16C4T_CCEP 寄存器中的 CC1EN 位，使能捕获计数器的值到捕获寄存器。
12. 如有需要，置位 GP16C4T_IER 寄存器中的 CC1IT 位，使能中断请求。置位 GP16C4T_DMAEN 寄存器中的 CC1DMA 位，使能 DMA 请求。

当发生输入捕获时：

5. 有效边沿产生，GP16C4T_CCVAL1 寄存器获取计数器的值。
6. CH1CCIF 标志位置位（中断标志）。若至少 2 个连续的捕获发生，但标志位没有及时清除，则 CH1OVIF 也会置位。
7. 中断的产生取决于 GP16C4T_IVS 寄存器的 CC1IT 位。
8. DMA 请求的产生取决于 CC1DMA。

为了处理捕获溢出，建议在读出捕获溢出标志位之前先读取捕获数据。这可以避免丢失在读出捕获标志位之后与读取数据之前可能重复产生的捕获信息。

注：捕获中断请求可由软件设置 GP16C4T_SGE 寄存器中 SGCCnE 位产生。

18.4.5.1 PWM输入模式

测量 I1 上 PWM 信号的周期和占空比的过程如下：

1. 为 GP16C4T_CCVAL1 选择有效的输入：GP16C4T_CHMR1 寄存器中的 CC1SSEL 位写"01"（I1 被选择）。
2. 为 I1 边沿检出选择有效的极性（用于捕获数据到 GP16C4T_CCVAL1 寄存器和计数器清零）：CC1POL 位写'0'（上升沿有效）。
3. 为 GP16C4T_CCVAL2 选择有效输入：GP16C4T_CHMR1 寄存器的 CC2SSEL 位写"10"（I1 被选择）。
4. 为 I1 边沿检出选择有效极性（用于捕获数据到 GP16C4T_CCVAL2）：CC2POL 位写'1'（下降沿有效）。
5. 选择有效的触发输入：GP16C4T_SMCON 寄存器的 TSSEL 位写"101"（I1 边沿检出被选择）。
6. 配置从机模式控制器为复位模式：GP16C4T_SMCON 寄存器的 SMODS 位写"100"。
7. 使能捕获：GP16C4T_CCEP 寄存器的 CC1EN 位和 CC2EN 位写'1'。

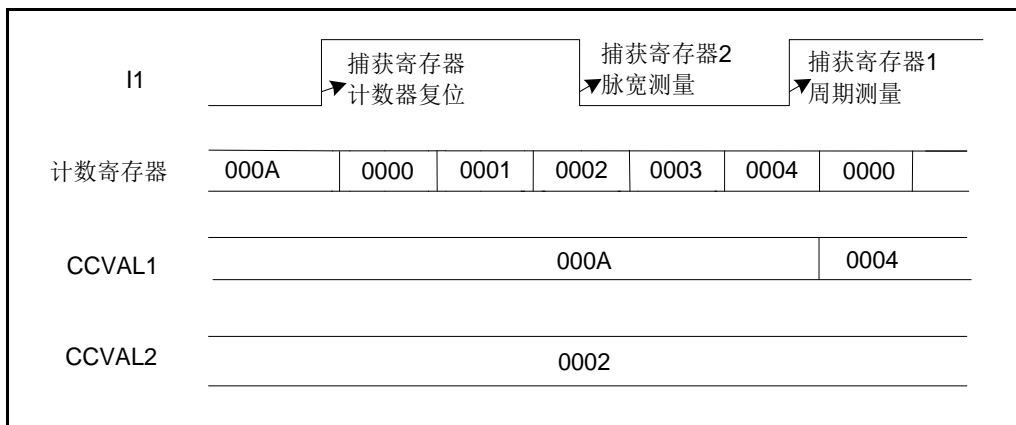


图 18-15 PWM 输入模式时序

18.4.6 PWM模式

脉宽调制模式可以产生一个 GP16C4T_AR 寄存器值确定频率，GP16C4T_CCVALn 寄存器值确定占空比的信号。

每个通道的 PWM 模式是相互独立的（每个 CHnO 输出一个 PWM），GP16C4T_CHMRn 寄存器的 CHnOMOD 位写"110"（PWM 模式 1）或写"111"（PWM 模式 2）。必须通过置位 GP16C4T_CHMRn 寄存器的 CHnOPREN 位来使能相应的预载寄存器，最后还需置位 GP16C4T_CON1 寄存器的 ARPEN 位来使能自动重装预载功能。

只有当更新事件发生时预载寄存器中的值才会传到影子寄存器，因此，在使能计数前，必须通过置位 GP16C4T_SGE 寄存器的 SGU 位来初始化所有的寄存器。

CHnO 的极性可通过 GP16C4T_CCEP 寄存器的 CCnPOL 位配置，有效极性可配置为高或低。CHnO 的输出使能由 CCnEN 位（GP16C4T_CCEP 寄存器）控制。

在 PWM 模式（1 或 2）中，GP16C4T_COUNT 和 GP16C4T_CCVALn 寄存器的值会持续的比较，确定 $GP16C4T_CCVALn \leq GP16C4T_COUNT$ 或 $GP16C4T_CCVALn > GP16C4T_COUNT$ （取决于计数器的计数方向）。

定时器产生 PWM 波形是边沿对齐或中心对齐，取决于 GP16C4T_CON1 寄存器的 CMSEL 位。

18.4.6.1 PWM边沿对齐模式

1. 递增计数配置

当 GP16C4T_CON1 寄存器的 DIRSEL 位为低时，计数器递增计数。

下图给出了 GP16C4T_AR = 8 时的边沿对齐 PWM 波形。

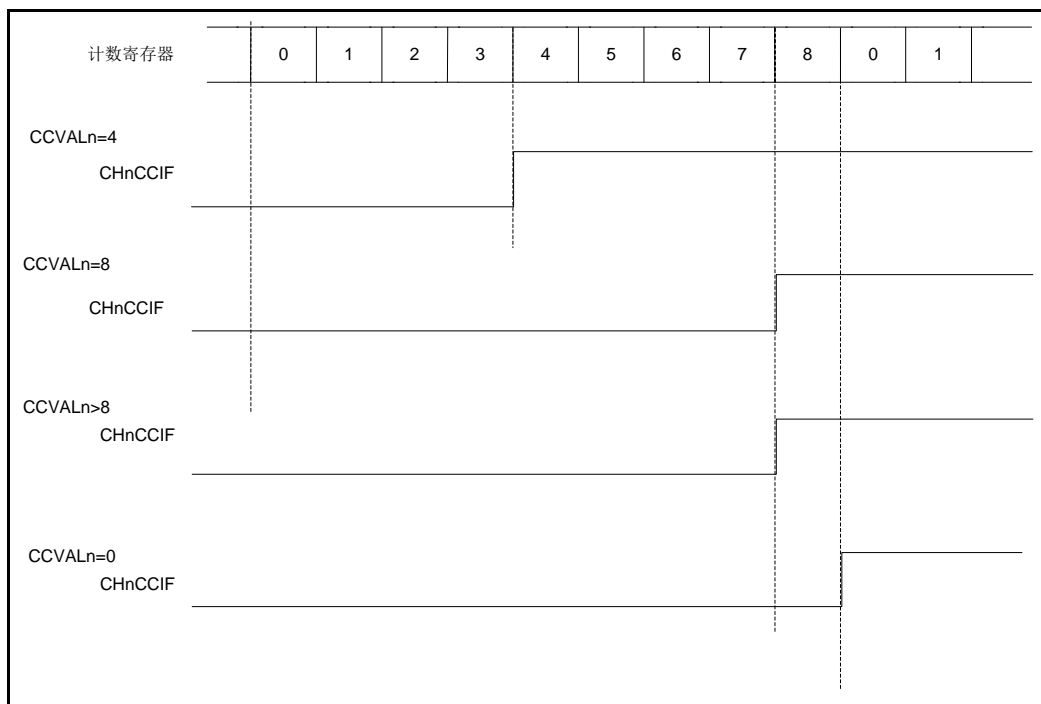


图 18-16 边沿对齐 PWM 波形 (AR=8)

2. 递减计数配置

当 GP16C4T_CON1 寄存器的 DIRSEL 位为高时，计数器递减计数。

18.4.6.2 PWM中心对齐模式

当 GP16C4T_CON1 寄存器中的 CMSEL 位不为"00"时，中心对齐模式有效。计数器是递增、递减计数分别置比较标志位或递增递减都置比较标志位，取决于 CMSEL 位的配置。GP16C4T_CON1 寄存器的方向位 (DIRSEL) 是由硬件更新的，软件无法修改。

下图为中心对齐方式产生的 PWM 波形的例子：

- ◇ GP16C4T_AR=0x3F, GP16C4T_CCVALn=0x3D
- ◇ PWM 模式 1
 - GP16C4T_CON1 寄存器的 CMSEL= "01", 在中心对齐模式 1 下，计数器向下计数时会置位比较标志位。

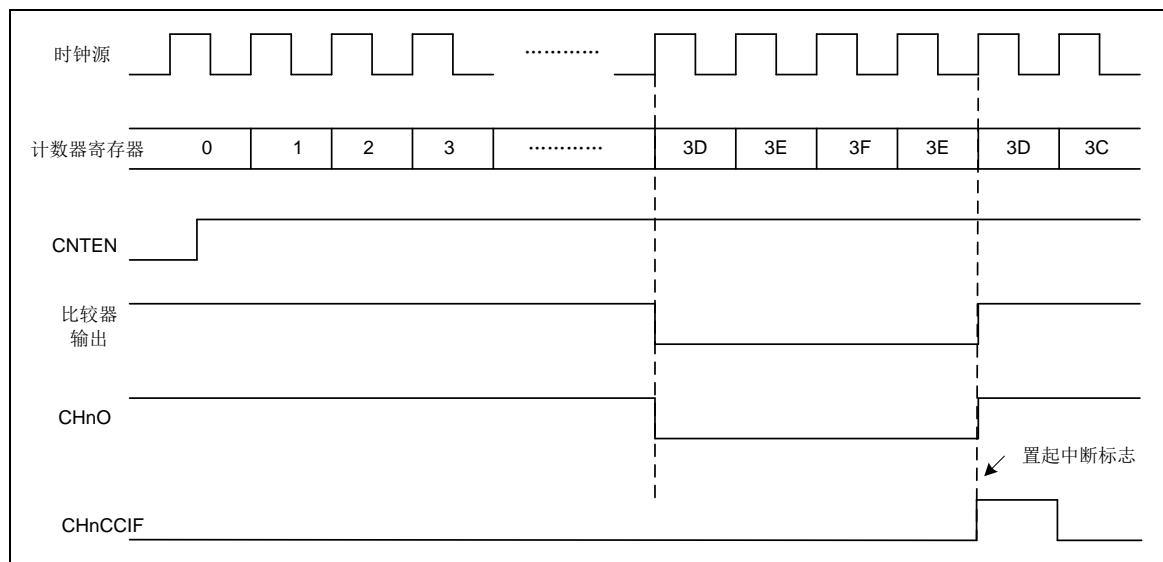


图 18-17 中心对齐 PWM 波形 (AR=0x3F)

中心对齐模式的使用技巧：

- ◇ 当进入中心对齐模式后，当前递增或递减配置生效。计数器递增或递减计数取决于 GP16C4T_CON1 寄存器的 DIRSEL 位的值。
- ◇ 计数器在中心对齐模式下运行时，对计数器写操作可能导致不可预知的结果。特别是：
 - 若向计数器入的值大于自动重载值 (GP16C4T_COUNT>GP16C4T_AR)，计数方向不更新。例如，如果计数器递增计数，写入值后仍旧递增计数。
 - 若向计数器写 0 或 GP16C4T_AR 中的重载值，则计数方向更新，但并没有产生 UEV。
- ◇ 使用中心对齐模式最安全的方式是计数器开始计数前通过软件产生更新事件（置位 GP16C4T_SGE 寄存器中的 SGU 位）且在计数器运行过程中不对计数器写值。

18.4.7 输出比较模式

该功能用于控制输出波形或指示周期时间的结束。

当捕获/比较寄存器和计数器值匹配时，输出比较功能：

- ◇ 输出比较模式（GP16C4T_CHMRn 寄存器中的 CHnOMOD 位）和输出极性（GP16C4T_CCEP 寄存器中的 CCnPOL 位）的配置值输出到对应的引脚上。
- ◇ 中断状态寄存器中的标志位置位（GP16C4T_RIF 寄存器的 CHnCCIF 位）。
- ◇ 若相应的中断掩码置位，则产生中断（GP16C4T_IER 寄存器的 CCnIT 位）。
- ◇ 若相应的使能位置位（GP16C4T_DMAEN 寄存器的 CCnDMA 位，GP16C4T_CON2 寄存器的 CCDMASEL 位用于 DMA 请求的选择），则发送 DMA 请求。

GP16C4T_CHMRn 寄存器中 CHnOPREN 位的值可决定 GP16C4T_CCVALn 寄存器是否带有预装载寄存器。

在输出比较模式中，更新事件 UEV 对 CHnO 的输出没有影响。计时分辨率为计数器的一次计数。输出比较模式同样可以用来输出单个脉冲（单脉冲模式）。

输出比较的配置过程：

1. 选定计数器时钟（内部，外部，预分频）。
2. GP16C4T_AR 与 GP16C4T_CCVALn 寄存器中写入预期值。
3. 若需要产生中断请求，置位 GP16C4T_IER 寄存器中的 CCnIT 位。
4. 选择输出模式，例如：
 - CHnOMOD = "011"，当 CNTV 与 CCRVn 匹配时，CHnO 输出翻转。
 - CHnOPREN = '0'，关闭预载寄存器。
 - CCnPOL = '0'，选择有效极性为高。
 - CCnEN = '1'，使能输出。
5. GP16C4T_CON1 寄存器中的 CNTEN 位置位，使能计数器。

通过配置 GP16C4T_CHMR1 寄存器的 CHnOPREN 位可将 GP16C4T_CCVALn 配置为是否带预装载寄存器。通过软件方式，GP16C4T_CCVALn 寄存器的值可随时更新控制输出波形。

18.4.7.1 外部事件清除比较输出

ETFP 输入端（GP16C4T_CHMRn 寄存器的 CHnOCLREN 位写'1'）上的高电平，可将给定通道的比较输出信号拉低。在下次更新事件（UEV）发生前，比较输出会一直保持为低。该功能只能应用在输出比较和 PWM 模式中，强制输出模式中不起作用。

ET 信号可以接到电流控制比较器的输出端。该例中，ET 须按如下流程配置：

1. 外部触发预分频器应该关闭：GP16C4T_SMCON 寄存器的 ETPSEL[1: 0]位应该写"00"
2. 外部时钟源 2 关闭：GP16C4T_SMCON 寄存器的 ECM2EN 位写'0'
3. 外部触发极性（ETPOL）和外部触发滤波器（ETFLT）可根据用户需要配置

18.4.8 单脉冲模式

单脉冲模式下，响应某个触发后，定时器的输出通道在可配置的延迟时间后产生一个脉冲，脉冲长度可配。从模式控制器可控制计数器的启动。脉冲波形可在输出比较模式和 PWM 模式下产生。置位 GP16C4T_CON1 寄存器的 SPMEN 位可选择单脉冲模式。计数器会在下次更新事件 UEV 产生时自动停止。

只有比较值不同于计数器初始值时，单脉冲才可以正确的产生。计数器开始计数前（定时器等待触发），必须如下配置：

- ◇ 递增计数：CNT < CCVALn ≤ AR（特别地，0 < CCVALn）
- ◇ 递减计数：CNT > CCVALn

基于 PWM 模式设置单脉冲输出波形的步骤如下：

- ◇ 设置 GP16C4T_CHMRn 寄存器的 CHnOMOD 位，选择 PWM 模式 1 或 2；
- ◇ 设置 GP16C4T_CCEP 寄存器的 CCnPOL 位，选择通道端口 CHnO 的输出极性；
- ◇ 设置 GP16C4T_CON1 寄存器的 DIRSEL, CMSEL, SPMEN 位，配置为递增或递减计数，PWM 普通波形模式，单脉冲模式使能；
- ◇ 设置 GP16C4T_CHMR1 寄存器的 CH1OPREN =1，GP16C4T_CON1 寄存器的 ARPEN =1，使能比较寄存器和计数重载寄存器的缓冲功能（也可以根据实际情况不使能缓冲）；
- ◇ 设置 GP16C4T_CCVALn 寄存器和 GP16C4T_AR 寄存器，配置单脉冲输出延时和脉宽时间；
- ◇ 设置 GP16C4T_SGE 寄存器的 SGU=1 来产生一个更新事件；
- ◇ 设置 GP16C4T_CON1 寄存器的 CNTEN=1 来启动计数器，也可以在触发模式下，通过外部触发输入信号来触发硬件自动设置 CNTEN=1。

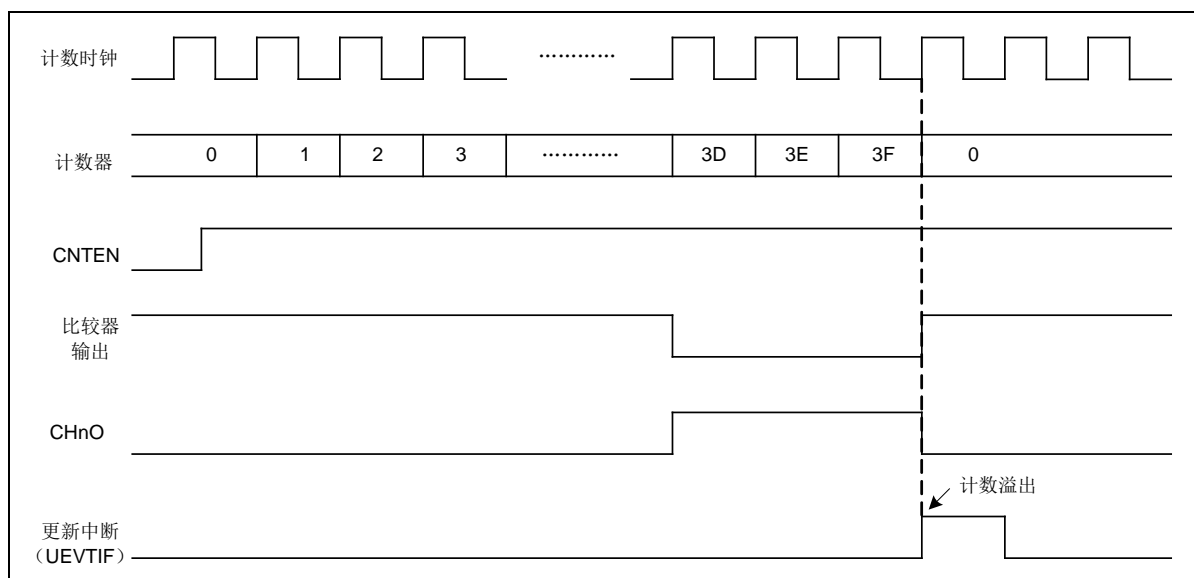


图 18-18 单脉冲模式

18.4.9 编码器接口模式

编码器接口模式的三种配置：若计数器只根据 I2 上的边沿计数，则 GP16C4T_SMCON 寄存器中的 SMODS = "001"；若计数器只根据 I1 上的边沿计数，则 GP16C4T_SMCON 寄存器中的 SMODS = "010"；若计数器同时根据 I1 和 I2 上的边沿计数，则 GP16C4T_SMCON 寄存器中的 SMODS = "011"。

配置 GP16C4T_CCEP 寄存器中的 CC1POL 和 CC2POL 位的值可选择 I1 和 I2 的极性。如果需要，也可以配置输入滤波器。

CH1_IN 和 CH2_IN 端口作为增量编码器的接口。当计数器使能时，计数器根据 I1 或 I2 上滤波后的有效电平变化时钟计数。I1 和 I2 滤波后的有效信号顺序会产生计数脉冲及方向信号。计数器是递增或递减计数由信号的跳变顺序决定，GP16C4T_CON1 寄存器中的 DIRSEL 计数方向位由自动硬件更新。

编码器接口模式的工作方式类似于一个带有方向选择的外部时钟。计数器在 0 到 GP16C4T_AR 寄存器中的自动重载值之间连续计数。因此，必须在开始计数前配置 GP16C4T_AR 寄存器。同样的，捕获器、预分频器、重复计数器、触发输出的特性正常工作。设定编码模式和选择外部时钟源 2 不兼容，不可以同时选择。

该模式下，计数器会根据增量式编码器的速度和方向自动修改，计数器的值反应的是编码器的位置。计数方向对应着连接传感器的旋转方向。

下表列出了所有的可能组合，假设 I1 和 I2 不同时变换。

有效边沿	有效边沿相对信号的电平 (I1 滤波信号对应 I2, I2 滤波信号对应 I1)	I1 滤波信号边沿		I2 滤波信号边沿	
		上升	下降	上升	下降
仅在 I1 计数	高	下降	上升	不计数	不计数
	低	上升	下降	不计数	不计数
仅在 I2 计数	高	不计数	不计数	上升	下降
	低	不计数	不计数	下降	上升
在 I1 和 I2 上计数	高	下降	上升	上升	下降
	低	上升	下降	下降	上升

表 18-1 计数方向与编码器信号的关系

外部增量编码器可直接与 MCU 连接，无需外部逻辑接口逻辑。而比较器通常用于将编码器的差分输出转换为数字信号，这极大地增加了抗噪声能力。编码器的第三个输出端用于指示机械零点，可以连接到外部中断输入引脚以触发一次计数复位。

下图给出了计数器操作的例子，给出了计数信号了产生和方向控制。同样给出了选择双边沿时，输入抖动如何被补偿。输入抖动可能发生在传感器靠近切换点处。

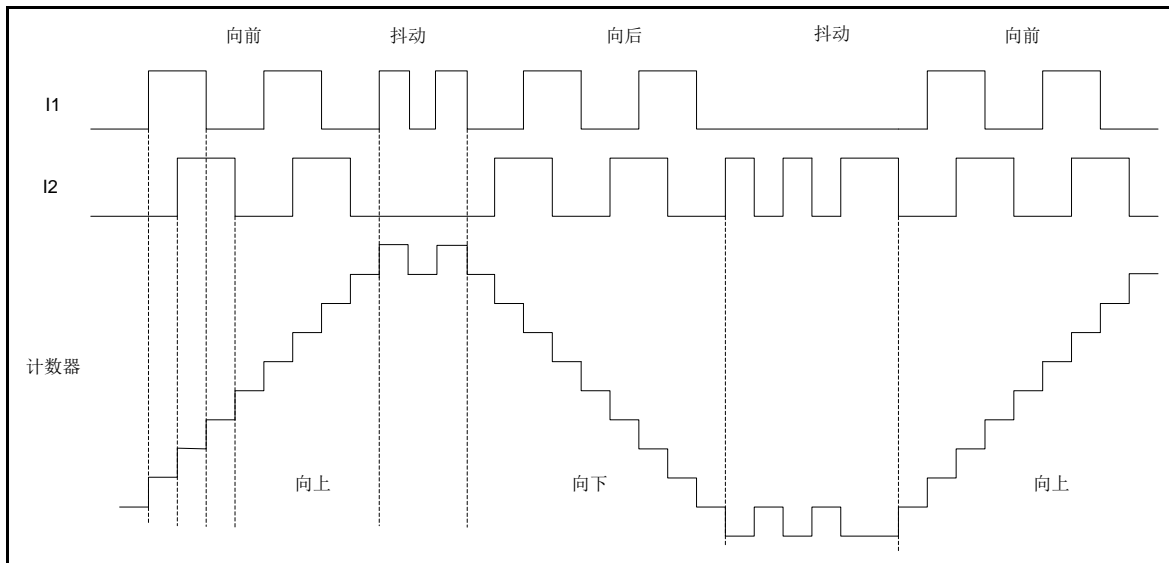


图 18-19 编码器接口模式下的计数操作

下图给出了计数器在 I1 滤波信号极性反相时的计数过程（除了 CC1POL = '1'，其他配置与上面一致）

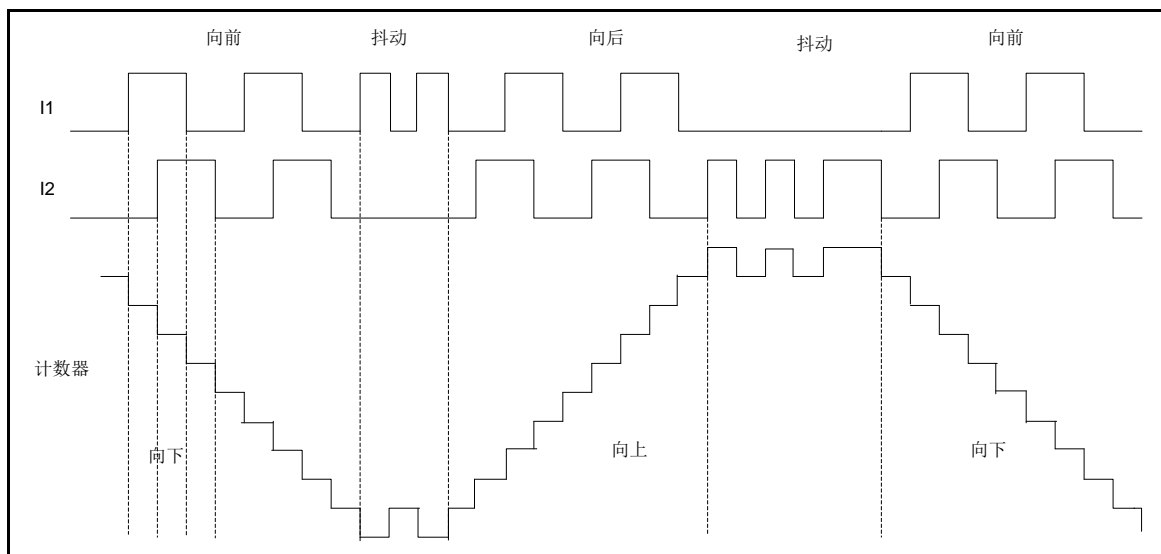


图 18-20 滤波后极性反相时编码器接口

当配置为编码器接口模式时，定时器可提供传感器的当前位置信息。配置一个额外定时器为捕获模式，用于测量两个编码器事件的间隔，根据间隔时长获取动态信息（速度、加速度、减速度）。编码器用于指示机械零点的输出就是此用处。根据编码器两个事件间隔，可以周期性的读取计数器的值。如果允许，可以将计数器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且可由其它定时器产生）。条件允许时，可通过实时时钟产生 DMA 请求的方式读取计数器值。

18.4.10 输入异或功能

通过 GP16C4T_CON2 寄存器中 I1FSEL 位，可将 CH1_IN、CH2_IN 和 CH3_IN 三个输入引脚进行异或（XOR）逻辑后，连接到通道 1 的输入滤波器。

异或逻辑输出可用于定时器的所有输入功能，如触发或输入捕获。

18.4.11 定时器和外部触发的同步

GP16C4T 定时器可在多种模式下与外部触发同步：复位模式、门控模式及触发模式。

18.4.11.1 复位模式

计数器及其预分频器可以在响应触发输入事件时重新初始化。此外，若 GP16C4T_CON1 寄存器的 UERSEL 位为低时会产生一次更新事件 UEV。所有预载寄存器（GP16C4T_AR，GP16C4T_CCVALn）都会因更新事件 UEV 而被更新。

在下面例子中，I1 输入端的上升沿让递增计数被清空：

- ◇ 配置通道 1 上检测 I1 上的上升沿。配置输入滤波周期（本例无需滤波器，故 I1FLT = "0000"）。触发捕获分频器没有使用，无需配置。CC1SSEL 位只选择输入捕获源，GP16C4T_CHMR1 寄存器中 CC1SSEL = "01"。GP16C4T_CCEP 寄存器中 CC1POL = 0 以确定极性（只检测上升沿）。
- ◇ 定时器配置为复位模式：GP16C4T_SMCON 寄存器中 SMODS = "100"。选择 I1 作为输入源：GP16C4T_SMCON 寄存器中 TSSEL = "101"。
- ◇ 启动计数器：GP16C4T_CON1 寄存器中 CNTEN = '1'。

计数器依据内部时钟开始计数，正常计数直到 I1 上出现上升沿。当 I1 上出现上升沿时，计数器会被清零且从 0 重新开始计数。同时，标志位置位（GP16C4T_RIF 寄存器中 TRGIF 位），如果中断及 DMA 使能（取决于 GP16C4T_IER 寄存器中的 TRGIT 和 GP16C4T_DMAEN 的 TRGDMA 位），会发送中断及 DMA 请求。

下图给出了当自动重载寄存器 GP16C4T_AR = 0x36 时的信号变化。由于 I1 输入的再同步电路，I1 上的上升沿和计数器实际复位之间存在延时（包含 2~3 个模块时钟周期的同步延时）。

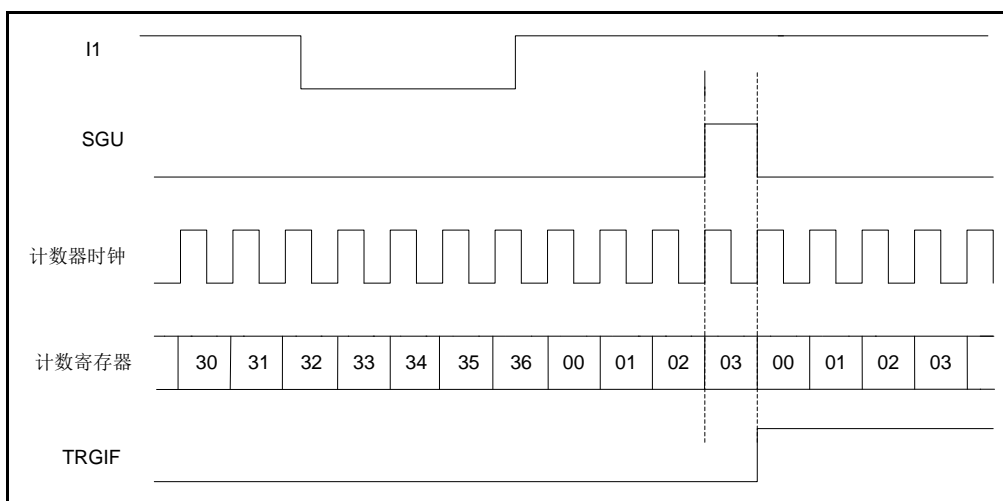


图 18-21 复位模式控制电路

18.4.11.2 门控模式

计数器根据选中的输入电平被使能。

下面的例子中，计数器只在 I1 输入为低电平时才递增计数：

- ◇ 配置通道 1 在 I1 上检测低电平。配置输入滤波周期（本例不需要滤波器，I1FLT = "0000"）。触发捕获分频器没有使用，无需配置。GP16C4T_CHMR1 寄存器中的 CC1SSEL = "01"，选择输入捕获源。GP16C4T_CCEP 寄存器中 CC1POL = '1'，确认极性（只检测低电平）。
- ◇ 配置定时器为门控模式：GP16C4T_SMCON 寄存器中 SMODS = "101"。选择 I1 作为输入源：GP16C4T_SMCON 寄存器中 TSSEL = "101"。
- ◇ 使能计数器：GP16C4T_CON1 寄存器中 CNTEN = '1'（门控模式中，如果 CNTEN = '0'，无论触发输入为何电平，计数器都不会启动）。

只要 I1 为低电平，计数器依据内部时钟开始计数，一旦 I1 为高则停止计数。由于 I1 输入端再同步电路的原因，I1 上出现上升沿和计数器实际停止之间会有一定的延时。

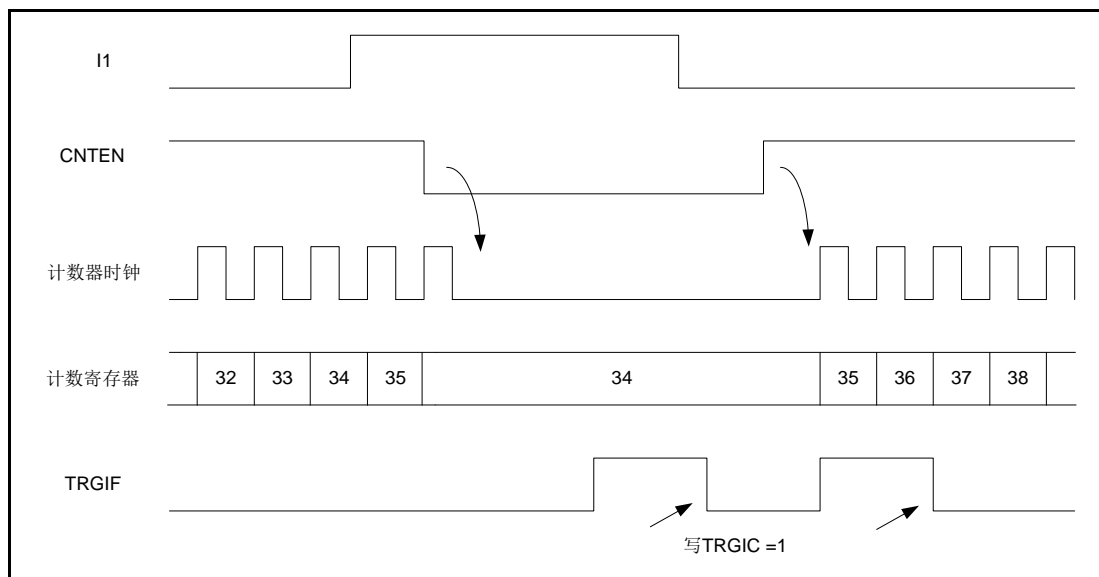


图 18-22 门控模式控制电路

18.4.11.3 触发模式

输入端选中的事件可以使能计数器。

下面的例子中，I2 输入端上的上升沿可以启动递增计数：

- ◇ 配置通道 2 可以检测 I2 上的上升沿。配置滤波时间（本例不需要滤波，I2FLT = "0000"）。触发捕获分频器没有使用，无需配置。GP16C4T_CHMR1 寄存器中 CC2SSEL = "01"，用于选择捕获源。GP16C4T_CCEP 寄存器中 CC2POL = '1'，确认极性（只检测低电平）。
 - ◇ 配置定时器为触发模式：GP16C4T_SMCON 寄存器中 SMODS = "110"。GP16C4T_SMCON 寄存器中 TSSEL = "110"，用于选择输入源。
- I2 上出现上升沿时，计数器开始依据内部时钟计数并置位 TRGIF 标志位。

由于 I2 输入的再同步原因, I2 上出现上升沿和计数器实际停止之间会有一定的延时。

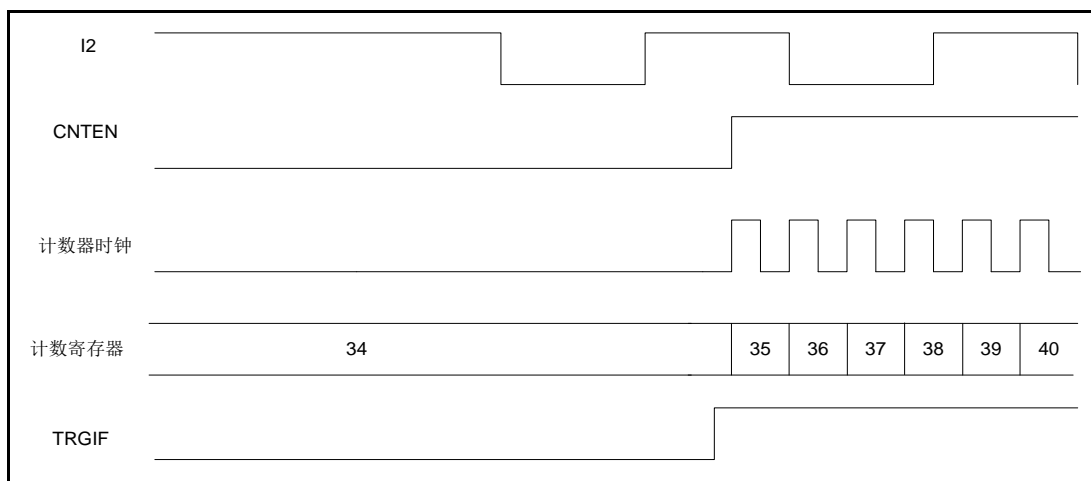


图 18-23 触发模式控制电路

18.4.11.4 选择外部时钟源 2 的触发模式

外部时钟源 2 可和其他模式一起使用 (除编码模式)。ET 信号可作为外部时钟输入, 另一个输入可选择为触发输入 (复位模式、门控模式或触发模式)。不推荐用 GP16C4T_SMCON 寄存器的 TSSEL 位选择 ET 作为 TI。

下面的例子中, 一旦 I1 上出现上升沿时, 计数器会依据 ET 信号的每个上升沿递增计数。

- ◇ 通过 GP16C4T_SMCON 寄存器, 配置外部触发输入电路, 过程如下:

ETFLT = "000": 无滤波

ETPSEL = "00": 禁止分频

ETPOL = '0': 检测 ET 的上升沿, ECM2EN = '1'使能外部时钟模式 2

- ◇ 配置通道 1 检测 I 的上升沿, 过程如下:

I1FLT = "0000": 无滤波。

触发捕获分频器没有使用, 无需配置。

GP16C4T_CHMR1 寄存器中 CC1SSEL = "01"选择输入捕获源, GP16C4T_CCEP 寄存器的 CC1POL = '0'确认极性 (只检测上升沿)。

- ◇ 配置定时器为触发模式: GP16C4T_SMCON 寄存器中 SMODS = "110"。

GP16C4T_SMCON 寄存器中 TSSEL = "101"选择 I1 作为输入源。

I1 上出现上升沿时, 计数器使能且 TRGIF 标志位置位, 然后计数器根据 ET 上的上升沿开始计数。

由于 ETFP 输入再同步电路的原因, ET 信号的上升沿和实际计数器的复位会有延时。

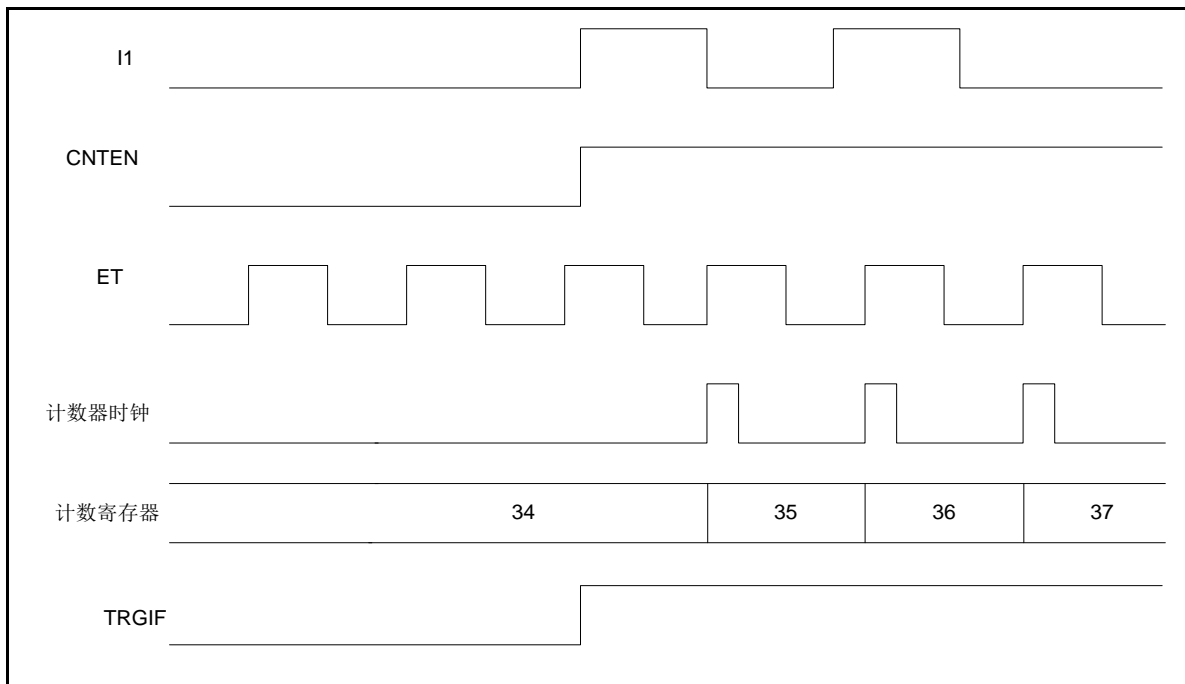


图 18-24 外部时钟源 2+触发模式下的控制电路

18.4.12 调试模式

当微控制器进入调试模式（CPU 内核停止），GP16C4T 计数器可被设置为停止计数。

18.5 特殊功能寄存器

18.5.1 寄存器列表

GP16C4T 寄存器列表		
名称	偏移地址	描述
GP16C4T_CON1	000 _H	控制寄存器 1
GP16C4T_CON2	004 _H	控制寄存器 2
GP16C4T_SMCON	008 _H	从模式控制寄存器
GP16C4T_IER	00C _H	中断使能寄存器
GP16C4T_IDR	010 _H	中断禁止寄存器
GP16C4T_IVS	014 _H	中断有效状态寄存器
GP16C4T_RIF	018 _H	原始中断标志寄存器
GP16C4T_IFM	01C _H	中断标志屏蔽寄存器
GP16C4T_ICR	020 _H	中断清零寄存器
GP16C4T_SGE	024 _H	软件生成事件计算器
GP16C4T_CHMR1	028 _H	捕获/比较模式寄存器 1
GP16C4T_CHMR2	02C _H	捕获/比较模式寄存器 2
GP16C4T_CCEP	030 _H	捕获/比较使能寄存器
GP16C4T_COUNT	034 _H	计数器寄存器
GP16C4T_PRES	038 _H	预分频寄存器
GP16C4T_AR	03C _H	自动重载寄存器
GP16C4T_CCVAL1	044 _H	捕获/比较寄存器 1
GP16C4T_CCVAL2	048 _H	捕获/比较寄存器 2
GP16C4T_CCVAL3	04C _H	捕获/比较寄存器 3
GP16C4T_CCVAL4	050 _H	捕获/比较寄存器 4
GP16C4T_DMAEN	058 _H	DMA 使能寄存器

18.5.2 寄存器描述

18.5.2.1 控制寄存器 1 (GP16C4T_CON1)

控制寄存器 1 (GP16C4T_CON1)																															
偏移地址: 000 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DBGSEL	Reserved	OCCISS			OCCISP	DFCKSEL		ARPEN	CMSEL		DIRSEL	SPMIEN	UERSSEL	DISUE	CNTEN

Reserved	Bit 31-16	-	保留，必须保持为复位值
DBGSEL	Bit 15	R/W	调试时通道输出状态选择 0: 通道输出为高阻态 1: 通道输出保持
Reserved	Bit 14	-	保留，必须保持为复位值
OCCISS	Bit 13-11	R/W	通道输出清除内部触发源选择 000: 无效 001: 清除触发源通道 0 010: 清除触发源通道 1 011: 清除触发源通道 2 100: 清除触发源通道 3 其他: 预留，不允许编程该值
OCCISP	Bit 10	R/W	通道输出清除内部触发源极性选择 0: 低电平有效 (OCCISS 设置为无效时该位设 0) 1: 高电平有效
DFCKSEL	Bit 9-8	R/W	时钟分频 该时钟分频为定时器 (INT_CLK) 频率与死区时间生成器和数字滤波器 (ET, In) 采用的死区时间和采样时钟 (tDTS) 之间的分频比。 00: $t_{DTS}=t_{INT_CLK}$ 01: $t_{DTS}=2*t_{INT_CLK}$ 10: $t_{DTS}=4*t_{INT_CLK}$ 11: 保留
ARPEN	Bit 7	R/W	自动重载预载使能 发生更新事件时，将设定的值载入至缓冲寄存器中 0: GP16C4T_AR 寄存器未缓冲 1: GP16C4T_AR 寄存器被装入缓冲器
CMSEL	Bit 6-5	R/W	中心对齐模式选择 00: 边沿对齐模式。计数器根据方向为 (DIRSEL) 来向上或向下计数。 01: 中心对齐模式 1。计数器以交替方式向上或

			<p>向下计数。仅当计数器向下计数时，配置为输出的通道（GP16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位才会被设置。</p> <p>10: 中心对齐模式 2。计数器以交替方式向上或向下计数。仅当计数器向上计数时，配置为输出的通道（GP16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位才会被设置。</p> <p>11: 中心对齐模式 3。计数器以交替方式向上或向下计数。当计数器向上或向下计数时，配置为输出的通道（GP16C4T_CHMRn 寄存器中 CCnSSEL=00）的输出比较中断标志位均会被设置。</p> <p>注意：当计数器使能时（CNTEN=1），不允许从边沿对齐模式转换到中心对齐模式</p>
DIRSEL	Bit 4	R/W	<p>计数器方向选择</p> <p>0: 计数器向上计数</p> <p>1: 计数器向下计数</p> <p>注意：当计数器配置为中心对齐模式或者编码器模式时，该位只读。</p>
SPMEN	Bit 3	R/W	<p>单脉冲模式</p> <p>0: 当发生更新事件时，计数器不停止。</p> <p>1: 当发生下一次更新事件（CNTEN 位清零）时，计数器停止。</p>
UERSEL	Bit 2	R/W	<p>更新请求源</p> <p>该位由软件置 1 或清零，来选择 UEV 事件源。</p> <p>0: 如果更新中断或 DMA 请求使能，则下述任一事件都可产生更新中断或 DMA 请求：</p> <ul style="list-style-type: none"> - 计数器上溢/下溢 - 设置 SGU 位 - 从模式控制器产生的更新 <p>1: 如果更新中断或 DMA 请求使能，仅计数器上溢/下溢才能产生更新中断或 DMA 请求中断</p>
DISUE	Bit 1	R/W	<p>更新禁止</p> <p>该位由软件置 1 或清零来使能/禁止 UEV 事件的产生。</p> <p>0: UEV 使能。更新事件（UEV）由下列任一事件产生：</p> <ul style="list-style-type: none"> - 计数器上溢/下溢 - 设置 SGU 位 - 从模式控制器产生的更新 <p>缓冲寄存器载入他们的预载值。</p> <p>1: UEV 禁止。不产生更新事件，影子寄存器保持</p>

			他们的值 (ARRV, PSCV, CCRVx) . 如果从模式控制器接收到硬件复位, 计数器和预分频器将被重新初始化。
CNTEN	Bit 0	R/W	<p>计数器使能</p> <p>0: 计数器禁止</p> <p>1: 计数器使能</p> <p>注意: 如果软件设置了 CNTEN 位, 外部时钟, 门控模式和编码器模式才能工作。触发模式可由硬件自动设置 CNTEN 位。</p>

18.5.2.2 控制寄存器 2 (GP16C4T_CON2)

控制寄存器 2 (GP16C4T_CON2)																															
偏移地址: 004 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							I1FSEL	TRGOSEL			CCDMASEL	Reserved			

Reserved	Bit 31-8	-	保留, 必须保持为复位值
I1FSEL	Bit 7	R/W	<p>I1 选择</p> <p>0: GP16C4T_CH1 引脚与 I1 输入连接</p> <p>1: GP16C4T_CH1, CH2 和 CH3 引脚与 I1 输入 (XOR) 连接</p>
TRGOSEL	Bit 6-4	R/W	<p>选择主模式 TRGOUT 输出</p> <p>为同步 (TRGOUT), 该位可选择在主模式下发送至从计数器的信息。</p> <p>000: 复位-GP16C4T_SGE 寄存器中的 SGU 位被采用为触发输出 (TRGOUT)。如果复位由触发输入生成 (从模式控制器配置复位模式), 则相较于实际复位, TRGOUT 上的信号将会延迟。</p> <p>001: 使能-计数器使能信号被用作触发输出 (TRGOUT)。在从计数器使能的情况下, 该设置用于在同一时间启动数次或者用来控制窗口。计数器使能信号是由 CNTEN 控制位与配置为门控模式的触发输入进行 OR 操作产生的。当计数器使能信号由触发输入控制, TRGOUT 上会产生延迟, 除非被选为主/从模式 (参考 GP16C4T_SMCON 寄存器中的 MSCFG 位的描述)。</p> <p>010: 更新-更新事件被选为触发输出 (TRGOUT)。举例, 主计数器可被用作从计数器的预分频器。</p> <p>011: 比较脉冲-一旦捕获或者比较匹配发生, 当 CH1CCIF 标志位被置起 (即便已为高电平), 触发输出会发送一个正脉冲。</p> <p>100: 比较- 通道 1 比较输出信号用作触发输出 TRGOUT</p> <p>101: 比较- 通道 2 比较输出信号用作触发输出 TRGOUT</p> <p>110: 比较- 通道 3 比较输出信号用作触发输出 TRGOUT</p>

			111: 比较- 通道 4 比较输出信号用作触发输出 TRGOUT
CCDMASEL	Bit 3	R/W	捕获/比较 DMA 选择 0: 当 CCn 事件发生, 会发出 CCn DMA 请求。 1: 当发生更新时间, 会发出 CCn DMA 请求。
Reserved	Bit 2-0	-	保留, 必须保持为复位值

18.5.2.3 从模式控制寄存器 (GP16C4T_SMCON)

从模式控制寄存器 (GP16C4T_SMCON)																															
偏移地址: 008 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ETPOL	ECM2EN	ETPSEL		ETFLT				MSCFG	TSSEL			Reserved	SMODS		

Reserved	Bit 31-16	-	保留, 必须保持为复位值
ETPOL	Bit 15	R/W	外部触发极性 0: 正向 ET, 高电平有效或上升沿有效 1: 反正 ET, 低电平有效或下降沿有效
ECM2EN	Bit 14	R/W	使能外部时钟模式 2 该位使能外部时钟模式 2 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2。计数器由 ETFP 信号上的有效边沿计数。 注意: 1. 设置 ECM2EN 位与选择外部时钟模式 1 且 TI 与 ETFP 相连接 (SMODS=111 和 TSSEL=111) 具有相同的效果。 2. 可同时使用外部时钟模式 2 与下列从模式: 复位模式, 门控模式和触发模式。在这种情况下, TI 不能与 ETFP 相连接 (TSSEL 不能设置为 111)。 3. 如果外部时钟模式 1 和外部时钟模式 2 同时使能, 外部时钟输入为 ETFP。
ETPSEL	Bit 13-12	R/W	外部触发预分频器 外部触发信号频率最大为 GP16C4TnCLK 频率的 1/4。可使能预分频器来减小 ETFP 频率。该位有效用于输入高速外部时钟的情况。 00: 预分频器关闭 01: ETFP 频率 2 分频 10: ETFP 频率 4 分频 11: ETFP 频率 8 分频
ETFLT	Bit 11-8	R/W	外部触发滤波器 该位定义了 ETFP 信号的采样频率和数字滤波器的滤波长度。 数字滤波器由一个事件计数器组成, 每 N 个连续事件才视为一个有效边沿。 0000: 无滤波器, 采样频率为 f_{DTS} 0001: $f_{SAMPLING} = f_{INT_CLK}$, $N = 2$

			<p>0010: $f_{\text{SAMPLING}} = f_{\text{INT_CLK}}, N = 4$ 0011: $f_{\text{SAMPLING}} = f_{\text{INT_CLK}}, N = 8$ 0100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2, N = 6$ 0101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 2, N = 8$ 0110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 6$ 0111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 4, N = 8$ 1000: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 6$ 1001: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8, N = 8$ 1010: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 5$ 1011: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 6$ 1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 8$ 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 5$ 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 6$ 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 8$ 注意: 当ETFLT[3:0] = 1, 2或3时, 公式中的f_{DTS}由INT_CLK 取代。</p>
MSCFG	Bit 7	R/W	<p>主/从模式 0: 无动作 1: 延迟触发输入 (In) 上的事件来允许当前计时器和其从器件之间的同步。该设置有效用于使用单个外部事件来同步多个计时器。</p>
TSSEL	Bit 6-4	R/W	<p>触发选择 该位用来选择不同的触发输入来同步计数器。 000: 内部触发 0 (IT0) 001: 内部触发 1 (IT1) 010: 内部触发 2 (IT2) 011: 内部触发 3 (IT3) 100: I1 边沿检测器 (I1F_ED) 101: 滤波计时器输入 1 110: 滤波计时器输入 2 111: 外部触发输入 注意: 为了避免错误边沿检测, 该位在不使用时 (SMODS=000) 才能改变。</p>
Reserved	Bit 3	-	保留, 必须保持为复位值
SMODS	Bit 2-0	R/W	<p>选择从模式功能 当选择外部信号, 触发信号TI的有效边沿与外部输入的极性有关系 (详见输入控制寄存器和控制寄存器描述) 000: 禁止从模式—如果CNTEN = '1', 则预分频器直接由内部时钟计数。 001: 编码器模式1—计数器向上/向下计数I2边沿, 取决于I1电平。 010: 编码器模式2—计数器向上/向下计数I1边沿, 取决于I2电平</p>

		<p>011 :编码器模式3 -计数器向上/向下计数I1边沿检出和I2边沿检出边沿，取决于另一个输入的电平。</p> <p>100 : 复位模式-选中的触发输入的上升沿重新初始化计数器，生成寄存器的更新</p> <p>101 : 门控模式-当触发输入TI为高电平，计数器时钟使能。一旦触发变为低电平，计数器停止计数（并非复位）。计数器的启动和停止均受控制。</p> <p>110 : 触发模式-计数器在触发信号TI的上升沿处启动（不复位）。仅寄存器的启动受控制。</p> <p>111 : 外部时钟模式1-计数器在TI的上升沿计数</p> <p>注意：如果I1双边沿检出被选为触发输入（TSSEL='100'），不能使用门控模式。I1每一次转换，I1双边沿检出就会输出1个脉冲，而门控模式则是检查触发信号的电平。</p> <p>注意：在发生来自自主计时器的接收事件之前，从计时器的时钟必须先使能，且在接收来自自主计时器的触发过程中，从计数器时钟不能即时更改。</p>
--	--	--

18.5.2.4 中断使能寄存器 (GP16C4T_IER)

中断使能寄存器 (GP16C4T_IER)																															
偏移地址: 00C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																			CC4OIT	CC3OIT	CC2OIT	CC1OIT	Reserved	TRGIT	Reserved	CC4IT	CC3IT	CC2IT	CC1IT	UIT	

Reserved	Bit31-13	-	保留, 必须保持复位值。
CC4OIT	Bit12	W	使能捕获/比较 4 捕获溢出中断 0: 无效 1: 使能
CC3OIT	Bit11	W	使能捕获/比较 3 捕获溢出中断 0: 无效 1: 使能
CC2OIT	Bit10	W	使能捕获/比较 2 捕获溢出中断 0: 无效 1: 使能
CC1OIT	Bit9	W	使能捕获/比较 1 捕获溢出中断 0: 无效 1: 使能
Reserved	Bit 8-7	-	保留, 必须保持为复位值
TRGIT	Bit6	W	使能触发中断 0: 无效 1: 使能
Reserved	Bit 5	-	保留, 必须保持为复位值
CC4IT	Bit4	W	使能捕获/比较 4 中断 0: 无效 1: 使能
CC3IT	Bit3	W	使能捕获/比较 3 中断 0: 无效 1: 使能
CC2IT	Bit2	W	使能捕获/比较 2 中断 0: 无效 1: 使能
CC1IT	Bit1	W	使能捕获/比较 1 中断 0: 无效 1: 使能
UIT	Bit0	W	使能更新事件中 0: 无效 1: 使能

18.5.2.5 中断禁止寄存器 (GP16C4T_IDR)

中断禁止寄存器 (GP16C4T_IDR)																															
偏移地址: 0010 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC4OIT	CC3OIT	CC2OIT	CC1OIT	Reserved	TRGIT	Reserved	CC4IT	CC3IT	CC2IT	CC1IT	UIT								

Reserved	Bit 31-13	-	保留, 必须保持复位值。
CC4OIT	Bit12	W	禁止捕获/比较 4 捕获溢出中断 0: 无效 1: 禁止
CC3OIT	Bit11	W	禁止捕获/比较 3 捕获溢出中断 0: 无效 1: 禁止
CC2OIT	Bit10	W	禁止捕获/比较 2 捕获溢出中断 0: 无效 1: 禁止
CC1OIT	Bit9	W	禁止捕获/比较 1 捕获溢出中断 0: 无效 1: 禁止
Reserved	Bit 8-7	-	保留, 必须保持为复位值
TRGIT	Bit 6	W	禁止触发中断 0: 无效 1: 禁止
Reserved	Bit 5	-	保留, 必须保持为复位值
CC4IT	Bit 4	W	禁止捕获/比较 4 中断 0: 无效 1: 禁止
CC3IT	Bit 3	W	禁止捕获/比较 3 中断 0: 无效 1: 禁止
CC2IT	Bit 2	W	禁止捕获/比较 2 中断 0: 无效 1: 禁止
CC1IT	Bit 1	W	禁止捕获/比较 1 中断 0: 无效 1: 禁止
UIT	Bit 0	W	禁止更新中断 0: 无效 1: 禁止

18.5.2.6 中断有效状态寄存器 (GP16C4T_IVS)

中断有效状态寄存器 (GP16C4T_IVS)																															
偏移地址: 0014 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																			CC4OIT	CC3OIT	CC2OIT	CC1OIT	Reserved	TRGIT	Reserved	CC4IT	CC3IT	CC2IT	CC1IT	UIT	

Reserved	Bit 31-13	-	保留
CC4OIT	Bit12	R	捕获/比较 4 捕获溢出中断状态 0: 禁止 1: 使能
CC3OIT	Bit11	R	捕获/比较 3 捕获溢出中断状态 0: 禁止 1: 使能
CC2OIT	Bit10	R	捕获/比较 2 捕获溢出中断状态 0: 禁止 1: 使能
CC1OIT	Bit9	R	捕获/比较 1 捕获溢出中断状态 0: 禁止 1: 使能
Reserved	Bit 8-7	-	保留, 必须保持为复位值
TRGIT	Bit 6	R	触发中断状态 0: 禁止 1: 使能
Reserved	Bit 5	-	保留, 必须保持为复位值
CC4IT	Bit 4	R	通道 4 捕获/比较中断状态 0: 禁止 1: 使能
CC3IT	Bit 3	R	通道3捕获/比较中断状态 0: 禁止 1: 使能
CC2IT	Bit 2	R	通道 2 捕获/比较中断状态 0: 禁止 1: 使能
CC1IT	Bit 1	R	通道2捕获/比较中断状态 0: 禁止 1: 使能
UIT	Bit 0	R	更新事件中断状态 0: 禁止 1: 使能

18.5.2.7 原始中断标志寄存器 (GP16C4T_RIF)

原始中断标志寄存器 (GP16C4T_RIF)																															
偏移地址: 018 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CH4OVIF	CH3OVIF	CH2OVIF	CH1OVIF	Reserved	TRGIF	Reserved	CH4CCIF	CH3CCIF	CH2CCIF	CH1CCIF	UEVTIF				

Reserved	Bit 31-13	-	保留, 必须保持为复位值
CH4OVIF	Bit 12	R	<p>捕获/比较 4 捕获溢出中断标志</p> <p>仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未检测到捕获溢出</p> <p>1: 当 CH4CCIF 标志位置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器</p>
CH3OVIF	Bit 11	R	<p>捕获/比较 3 捕获溢出中断标志</p> <p>仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未检测到捕获溢出</p> <p>1: 当 CH3CCIF 标志位置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器</p>
CH2OVIF	Bit 10	R	<p>捕获/比较 2 捕获溢出中断标志</p> <p>仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未检测到捕获溢出</p> <p>1: 当 CH2CCIF 标志位置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器</p>
CH1OVIF	Bit 9	R	<p>捕获/比较 1 捕获溢出中断标志</p> <p>仅当相应的通道配置为捕获输入状态时, 该标志位才由硬件设置。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未检测到捕获溢出</p> <p>1: 当 CH1CCIF 标志位置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器</p>
Reserved	Bit 8-7	-	保留, 必须保持为复位值
TRGIF	Bit 6	R	<p>触发中断标志</p> <p>如果触发中断使能, 当从模式控制器在门控模式</p>

			<p>以外的所有模式下使能，发生触发事件时（In 上检测到有效边沿），该标志位被硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生触发事件 1: 触发中断被挂起</p>
Reserved	Bit 5	-	保留，必须保持为复位值
CH4CCIF	Bit 4	R	<p>捕获/比较 4 中断标志</p> <p>参考 CH1CCIF 描述</p>
CH3CCIF	Bit 3	R	<p>捕获/比较 3 中断标志</p> <p>参考 CH1CCIF 描述</p>
CH2CCIF	Bit 2	R	<p>捕获/比较 2 中断标志</p> <p>参考 CH1CCIF 描述</p>
CH1CCIF	Bit 1	R	<p>捕获/比较 1 中断标志</p> <p>如果 CC1 通道配置为输出:</p> <p>如果中断使能，除去中心对齐模式的情况（参考 GP16C4T_CON1 寄存器中 CMSEL 的描述），当计数值与比较值匹配，该标志位由硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 不匹配。 1: GP16C4T_COUNT 计数值与 GP16C4T_CCVAL1 值匹配。当 GP16C4T_CCVAL1 寄存器值大于 GP16C4T_AR 值，发生计数器上溢时（递增模式和递增/递减模式）或下溢时（递减模式），CH1CCIF 为被置起</p> <p>如果 CC1 通道配置为输入:</p> <p>发生捕获时，该位由硬件置起。该位可通过软件或者读取 GP16C4T_CCVAL1 寄存器来清零。</p> <p>0: 未发生输入捕获 1: 计数值捕获至 GP16C4T_CCVAL1 寄存器（IC1 上检测到与选中极性匹配的边沿）</p>
UEVTIF	Bit 0	R	<p>更新中断标志</p> <p>如果更新中断使能，当发生更新事件，该标志位由硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生更新。 1: 更新中断被挂起。当寄存器更新时，该位被硬件置起：</p> <ul style="list-style-type: none"> -当重复计数器值发生上溢或者下溢（若重复计数器=0，则更新）和当 GP16C4T_CON1 寄存器中 DISUE=0 -当使用 GP16C4T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时，如果 GP16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0

			-当 CNT 由触发事件来重新初始化，如果 GP16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0
--	--	--	--

18.5.2.8 中断标志屏蔽寄存器 (GP16C4T_IFM)

中断标志屏蔽寄存器 (GP16C4T_IFM)																															
偏移地址: 001C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		CH4OVIM	CH3OVIM	CH2OVIM	CH1OVIM	Reserved	TRGIM	Reserved	CH4CCIM	CH3CCIM	CH2CCIM	CH1CCIM	UEVTIM		

Reserved	Bit 31-13	-	保留
CH4OVIM	Bit 12	R	屏蔽通道 4 捕获/比较捕获溢出中断标志 0: 未检测到捕获溢出 1: 当 CH4CCIF 标志为置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器
CH3OVIM	Bit 11	R	屏蔽通道 3 捕获/比较捕获溢出中断标志 0: 未检测到捕获溢出 1: 当 CH3CCIF 标志为置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器
CH2OVIM	Bit 10	R	屏蔽通道 2 捕获/比较捕获溢出中断标志 0: 未检测到捕获溢出 1: 当 CH2CCIF 标志为置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器
CH1OVIM	Bit 9	R	屏蔽通道 1 捕获/比较捕获溢出中断标志 0: 未检测到捕获溢出 1: 当 CH1CCIF 标志为置起时, 捕获计数器值至 GP16C4T_CCVAL1 寄存器
Reserved	Bit 8-7	-	保留
TRGIM	Bit 6	R	屏蔽触发中断标志 如果触发中断使能, 当从模式控制器在门控模式以外的所有模式下使能, 发生触发事件时 (TI 上检测到有效边沿), 该标志位被硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。 0: 未发生触发事件 1: 触发中断被挂起
Reserved	Bit 5	-	保留
CH4CCIM	Bit 4	R	屏蔽通道 4 捕获/比较中断标志 参考 CH1CCIM 描述
CH3CCIM	Bit 3	R	屏蔽通道 3 捕获/比较中断标志 参考 CH1CCIM 描述
CH2CCIM	Bit 2	R	屏蔽通道 2 捕获/比较中断标志 参考 CH1CCIM 描述
CH1CCIM	Bit 1	R	屏蔽通道 1 捕获/比较中断标志

			<p>如果通道 1 配置为输出： 如果中断使能，除去中心对齐模式的情况（参考 GP16C4T_CON1 寄存器中 CMSEL 的描述），当计数值与比较值匹配，该标志位由硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。 0：不匹配。 1：GP16C4T_COUNT 计数值与 GP16C4T_CCVAL1 值匹配。当 GP16C4T_CCVAL1 寄存器值大于 GP16C4T_AR 值，发生计数器上溢时（递增模式和递增/递减模式）或下溢时（递减模式），CH1CCIF 为被置起。</p> <p>如果通道配置为输入： 发生捕获时，该位由硬件置起。该位可通过软件或者读取 GP16C4T_CCVAL1 寄存器来清零。 0：未发生输入捕获 1：计数值捕获至 GP16C4T_CCVAL1 寄存器（I1 上检测到与选中极性匹配的边沿）</p>
UEVTIM	Bit 0	R	<p>屏蔽更新事件中中断标志 如果更新中断使能，当发生更新事件，该标志位由硬件置起。对 GP16C4T_ICR 写 1 来清除原始中断。 0：未发生更新。 1：更新中断被挂起。当寄存器更新时，该位被硬件置起： -当重复计数器值发生上溢或者下溢（若重复计数器=0，则更新）和当 GP16C4T_CON1 寄存器中 DISUE=0 -当使用 GP16C4T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时，如果 GP16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0 -当 CNT 由触发事件来重新初始化，如果 GP16C4T_CON1 寄存中的 UERSEL=0 和 DISUE=0</p>

18.5.2.9 中断清零寄存器 (GP16C4T_ICR)

中断清零寄存器 (GP16C4T_ICR)																															
偏移地址: 020 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		CH4OVIC	CH3OVIC	CH2OVIC	CH1OVIC	Reserved	TRGIC	Reserved	CH4CCIC	CH3CCIC	CH2CCIC	CH1CCIC	UEVTIC		

Reserved	Bit 31-13	-	保留, 必须保持为复位值
CH4OVIC	Bit 12	R	通道 4 捕获/比较捕获溢出中断标志清除 0: 无效 1: CH4OVIF 清除
CH3OVIC	Bit 11	R	通道 3 捕获/比较捕获溢出中断标志 0: 无效 1: CH3OVIF 清除
CH2OVIC	Bit 10	R	通道 2 捕获/比较捕获溢出中断标志 0: 无效 1: CH2OVIF 清除
CH1OVIC	Bit 9	R	通道 1 捕获/比较捕获溢出中断标志 0: 无效 1: CH1OVIF 清除
Reserved	Bit 8-7	-	保留, 必须保持为复位值
TRGIC	Bit 6	C_W1	触发中断清零 0: 无效 1: 触发中断清零 (GP16C4T_RIF)
Reserved	Bit 5	-	保留, 必须保持为复位值
CH4CCIC	Bit 4	C_W1	捕获/比较 4 中断清零 参考 CH1CCIC 描述
CH3CCIC	Bit 3	C_W1	捕获/比较 3 中断清零 参考 CH1CCIC 描述
CH2CCIC	Bit 2	C_W1	捕获/比较 2 中断清零 参考 CH1CCIC 描述
CH1CCIC	Bit 1	C_W1	捕获/比较 1 中断清零 0: 无效 1: 捕获/比较中断清零 (GP16C4T_RIF)
UEVTIC	Bit 0	C_W1	更新中断清零 0: 无效 1: 更新中断清零 (GP16C4T_RIF)

18.5.2.10 软件生成事件寄存器 (GP16C4T_SGE)

软件生成事件寄存器 (GP16C4T_SGE)																															
偏移地址: 024 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								SGTRG	Reserved	SGCC4E	SGCC3E	SGCC2E	SGCC1E	SGU	

Reserved	Bit 31-7	-	保留, 必须保持为复位值
SGTRG	Bit 6	W	<p>触发生成 该位由软件设置来生成触发事件, 可由硬件自动清零.</p> <p>0: 无动作 1: GP16C4T_RIF 寄存器中的 TRGIF 被置起, 产生相关中断或 DMA 传输</p>
Reserved	Bit 5	-	保留, 必须保持为复位值
SGCC4E	Bit 4	W	<p>捕获/比较 4 生成 参考 SGCC1E 描述</p>
SGCC3E	Bit 3	W	<p>捕获/比较 3 生成 参考 SGCC1E 描述</p>
SGCC2E	Bit 2	W	<p>捕获/比较 2 生成 参考 SGCC1E 描述</p>
SGCC1E	Bit 1	W	<p>捕获/比较 1 生成 该位由软件设置来生成事件, 可由硬件自动清零.</p> <p>0: 无动作 1: 通道 1 上产生捕获/比较事件:</p> <p>如果通道 1 配置为输出: CH1CCIF 标志位被置起, 产生相应中断或 DMA 请求发送</p> <p>如果通道 1 配置为输入: 当前计数值捕获至 GP16C4T_CCVAL1 寄存器. CH1CCIF 标志位被置起, 产生相应中断或 DMA 请求发送。CH1OVIF 标志位置起如果 CH1CCIF 标志位为高电平。</p>
SGU	Bit 0	W	<p>更新生成 该位由软件设置, 可由硬件自动清零.</p> <p>0: 无动作 1: 重新初始化计数器, 更新寄存器。注意, 预分频器也会被清零 (但预分频比不会受到影响)。如果使用中心对齐模式或者 DIRSEL=0 (递增), 则计数器将清零; 否则如果 DIRSEL=1 (递减), 则将使用自动重载入值。</p>

18.5.2.11 捕获/比较模式寄存器 1 (GP16C4T_CHMR1)

◆ 输出比较模式

捕获/比较模式寄存器 1 (GP16C4T_CHMR1)																															
偏移地址: 028 _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CH2OCLREN	CH2OMOD			CH2OPREN	CH2OHSEN	CC2SSEL		CH1OCLREN	CH1OMOD			CH1OPREN	CH1OHSEN	CC1SSEL	

Reserved	Bit 31-16	-	保留，必须保持为复位值
CH2OCLREN	Bit 15	R/W	输出比较 2 清零使能 参考 CH1OCLREN 描述
CH2OMOD	Bit 14-12	R/W	输出比较 2 模式 参考 CH1OMOD 描述
CH2OPREN	Bit 11	R/W	输出比较 2 预载使能 参考 CH1OPREN 描述
CH2OHSEN	Bit 10	R/W	输出比较 2 高速使能 参考 CH1OHSEN 描述
CC2SSEL	Bit 9-8	R/W	输出比较 2 选择 该位定义了通道以及使用的输入的方向（输入/输出） 00：通道配置为输出 01：通道配置为输入，捕获源为 I2 10：通道配置为输入，捕获源为 I1 11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出。 仅当内部触发输入通过 TSEL 位（GP16C4T_SMCON 寄存器）选择时，该模式才能工作。 注意：当通道为关闭状态时（GP16C4T_CCEP 中 CC2EN = '0'），CC2SSEL 为只写。
CH1OCLREN	Bit 7	R/W	输出比较 1 清零使能 0：通道 1 比较输出不会受到 ETFP 输入影响 1：当 ETFP 输入上检测到高电平时，通道 1 比较输出将被清零
CH1OMOD	Bit 6-4	R/W	输出比较 1 模式 该位定义了输出参考信号通道 1 比较输出的行为。 通道 1 比较输出为高有效，CH1O 的有效电平由 CC1POL 位决定。 000：冻结—输出比较寄存器 GP16C4T_CCVAL1

		<p>寄存器和 GP16C4T_COUNT 计数器之间的比较对输出无效。</p> <p>001：发生匹配时设置通道 1 为有效电平-当计数器 GP16C4T_COUNT 与捕获/比较寄存器 1GP16C4T_CCVAL1 发生匹配后，通道 1 比较输出信号强制为高电平。</p> <p>010：发生匹配时设置通道 1 为无效电平。当计数器 GP16C4T_COUNT 与捕获/比较寄存器 1GP16C4T_CCVAL1 发生匹配后，通道 1 比较输出信号强制为低电平。</p> <p>011：翻转 -当 GP16C4T_COUNT=GP16C4T_CCVAL1，通道 1 比较输出发生翻转。</p> <p>100：强制为无效电平 - 通道 1 比较输出强制为低电平。</p> <p>101：强制为有效电平- 通道 1 比较输出强制为高电平。</p> <p>110：PWM 模式 1 -在递增模式下，当 GP16C4T_COUNT<GP16C4T_CCVAL1，通道 1 为有效电平，否则，通道 1 为无效电平。在递减模式下，当 GP16C4T_COUNT>GP16C4T_CCVAL1，通道 1 为无效电平（通道 1 比较输出='0'），否则通道 1 为有效电平（通道 1 比较输出='1'）。</p> <p>111：PWM 模式 2 -在递增模式下，当 GP16C4T_COUNT<GP16C4T_CCVAL1，通道 1 为无效电平，否则，通道 1 为有效电平。在递减模式下，当 GP16C4T_COUNT>GP16C4T_CCVAL1，通道 1 为有效电平，否则通道 1 为无效电平。</p> <p>注意： 在 PWM 模式 1 和 2 中，仅当比较结果更改或当输出比较模式从冻结模式转换成 PWM 模式，比较输出电平才会更改。</p>
CH1OPREN	Bit 3	<p>R/W</p> <p>输出比较 1 预载使能</p> <p>0：GP16C4T_CCVAL1 的预载寄存器禁止。GP16C4T_CCVAL1 在任何时候都可写，新写入的值将立刻生效。</p> <p>1：GP16C4T_CCVAL1 的预载寄存器使能。读/写操作可访问预载寄存器。每当发生一次更新事件，GP16C4T_CCVAL1 预载入值将会被填入有效寄存器。</p> <p>注意：</p>

			<p>仅在单脉冲模式下（GP16C4T_CON1 寄存器中的 SPMEN 设置为 1），PWM 模式可在不经过验证预载寄存器的情况下使用。其他情况下的行为不做保证。</p>
CH1OHSEN	Bit 2	R/W	<p>输出比较 1 高速使能 该位用来加速在 CC 输出上的输入触发事件的效应。 0：当触发开启，通道 1 运作正常取决于计数器和 CCRV1 的值。当触发输入上发现边沿时，至少需要 5 个时钟周期来激活通道 1 输出。 1：触发输入上的有效沿类似于通道 1 输出上的比较匹配。设置 OC 为 1 用来比较电平，采样触发输入和激活通道 1 输出的延时将会减少至 3 个时钟周期。只有当通道配置为 PWM1 或 PWM2 模式，CH1OHSEN 才会起作用。</p>
CC1SSEL	Bit 1-0	R/W	<p>捕获/比较 1 选择 该位定义了通道和使用的输入的方向。 00：通道配置为输出 01：通道配置为输入，捕获源为 I1 10：通道配置为输入，捕获源为 I2 11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出。 只有当内部触发输入是通过 TSSEL 位（GP16C4T_SMCON 寄存器）选择时，该模式才运行。 注意：当通道关闭（GP16C4T_CCEP 寄存器中的 CC1EN = '0'），CC1SSEL 为只写。</p>

◆ 输入捕获模式

捕获/比较模式寄存器 1 (GP16C4T_CHMR1)																															
偏移地址: 028 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																I2FLT			IC2PRES		CC2SSEL		I1FLT			IC1PRES		CC1SSEL			

Reserved	Bit 31-16	-	保留，必须保持为复位值
I2FLT	Bit 15-12	R/W	输入捕获 2 滤波器 参考 I1FLT 描述
IC2PRES	Bit 11-10	R/W	输入捕获 2 预分频器 参考 IC1PRES 描述
CC2SSEL	Bit 9-8	R/W	输入捕获 2 选择 该位定义了通道和使用的输入的方向。 00: 通道配置为 输出 01: 通道配置为输入，捕获源为 I2 10: 通道配置为输入，捕获源为 I1 11: 通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出 只有当内部触发输入是通过 TSSEL 位 (GP16C4T_SMCON 寄存器) 选择时，该模式才运行 注意：当通道关闭 (GP16C4T_CCEP 寄存器中的 CC2EN = '0')，CC2SSEL 为只写。
I1FLT	Bit 7-4	R/W	输入捕获 1 滤波器 该位定义了 I1 输入的采样频率和数字滤波器的长度。 数字滤波器由一个事件计数器组成，每 N 个连续事件才视为一个有效边沿： 0000: 无滤波器，采样频率为 f_{DTS} 0001: $f_{SAMPLING} = f_{INT_CLK}, N = 2$ 0010: $f_{SAMPLING} = f_{INT_CLK}, N = 4$ 0011: $f_{SAMPLING} = f_{INT_CLK}, N = 8$ 0100: $f_{SAMPLING} = f_{DTS} / 2, N = 6$ 0101: $f_{SAMPLING} = f_{DTS} / 2, N = 8$ 0110: $f_{SAMPLING} = f_{DTS} / 4, N = 6$ 0111: $f_{SAMPLING} = f_{DTS} / 4, N = 8$ 1000: $f_{SAMPLING} = f_{DTS} / 8, N = 6$ 1001: $f_{SAMPLING} = f_{DTS} / 8, N = 8$ 1010: $f_{SAMPLING} = f_{DTS} / 16, N = 5$ 1011: $f_{SAMPLING} = f_{DTS} / 16, N = 6$

			<p>1100: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16, N = 8$ 1101: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 5$ 1110: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 6$ 1111: $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32, N = 8$ 注意: 当 ICxF [3: 0] = 1, 2 or 3 时, 公式中的 f_{DTS} 由 INT_CLK 取代。</p>
IC1PRES	Bit 3-2	R/W	<p>输入捕获 1 预分频器 该位定义了作用在 CC1 输入 (I1) 上的预分频比。当 CC1EN='0' (GP16C4T_CCEP 寄存器), 预分频器将复位。 00: 无预分频器。每当捕获输入上检测到边沿时, 发生捕获动作。 01: 每发生 2 次事件, 执行一次捕获 10: 每发生 4 次事件, 执行一次捕获 11: 每发生 8 次事件, 执行一次捕获</p>
CC1SSEL	Bit 1-0	R/W	<p>输入捕获 1 选择 该位定义了通道和使用的输入的方向 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 I1 10: CC1 通道配置为输入, IC1 映射到 I2 11: CC1 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出。只有当内部触发输入是通过 TSSEL 位 (GP16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (GP16C4T_CCEP 寄存器中的 CC1EN = '0'), CC1SSEL 为只写。</p>

18.5.2.12 捕获/比较模式寄存器 2 (GP16C4T_CHMR2)

◆ 输出比较模式

捕获/比较模式寄存器 2 (GP16C4T_CHMR2)																															
偏移地址: 02C _H																															
复位值: 00000000_00000000_00000000_00000000 _b																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CH4OCLREN	CH4OMOD			CH4OPREN	CH4OHSEN	CC4SSEL		CH3OCLREN	CH3OMOD			CH3OPREN	CH3OHSEN	CC3SSEL	

Reserved	Bit 31-16	-	保留，必须保持为复位值
CH4OCLREN	Bit 15	R/W	输出比较 4 清零使能 参考 CH1OCLREN 描述
CH4OMOD	Bit 14-12	R/W	输出比较 4 模式 参考 CH1OMOD 描述
CH4OPREN	Bit 11	R/W	输出比较 4 预载使能 参考 CH1OPREN 描述
CH4OHSEN	Bit 10	R/W	输出比较 4 高速使能 参考 CH1OHSEN 描述
CC4SSEL	Bit 9-8	R/W	输出比较 4 选择 该位定义了通道以及使用的输入的方向（输入/输出）。 00：通道配置为输出 01：通道配置为输入，捕获源为 I4 10：通道配置为输入，捕获源为 I3 11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出 仅当内部触发输入通过 TSEL 位（GP16C4T_SMCON 寄存器）选择时，该模式才能工作。 注意：当通道为关闭状态时（GP16C4T_CCEP 中 CC4EN = '0'），CC4SSEL 为只写。
CH3OCLREN	Bit 7	R/W	输出比较 3 清零使能 参考 CH1OCLREN 描述
CH3OMOD	Bit 6-4	R/W	输出比较 3 模式 参考 CH1OMOD 描述
CH3OPREN	Bit 3	R/W	输出比较 3 预载使能 参考 CH1OPREN 描述
CH3OHSEN	Bit 2	R/W	输出比较 3 高速使能 参考 CH1OHSEN 描述
CC3SSEL	Bit 1-0	R/W	捕获/比较 3 选择

		<p>该位定义了通道和使用的输入的方向。</p> <p>00：通道配置为输出</p> <p>01：通道配置为输入，捕获源为 I3</p> <p>10：通道配置为输入，捕获源为 I4</p> <p>11：通道配置为输入，捕获源为 ITn 或 I1 的双边沿检出。</p> <p>只有当内部触发输入是通过 TSSEL 位（GP16C4T_SMCON 寄存器）选择时，该模式才运行。</p> <p>注意：当通道关闭（GP16C4T_CCEP 寄存器中的 CC3EN = '0'），CC3SSEL 为只写</p>
--	--	---

◆ 输入捕获模式

捕获/比较模式寄存器 2 (GP16C4T_CHMR2)																															
偏移地址: 02C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																I4FLT			IC4PRES		CC4SSEL		IC3FLT			IC3PRES		CC3SSEL			

Reserved	Bit 31-16	-	保留，必须保持为复位值
I4FLT	Bit 15-12	R/W	输入捕获 4 滤波器 参考 I1FLT 描述
IC4PRES	Bit 11-10	R/W	输入捕获 4 预分频器 参考 IC1PRES 描述
CC4SSEL	Bit 9-8	R/W	输入捕获 4 选择 该位定义了通道和使用的输入的方向。 00: CC4 通道配置为输出 01: CC4 通道配置为输入, IC4 映射到 TI4 10: CC4 通道配置为输入, IC4 映射到 TI3 11: CC4 通道配置为输入, IC4 映射到 TRC 只有当内部触发输入是通过 TSSEL 位 (GP16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (GP16C4T_CCEP 寄存器中的 CC4EN = '0'), CC4SSEL 为只写。
IC3FLT	Bit 7-4	R/W	输入捕获 3 滤波器 参考 I1FLT 描述
IC3PRES	Bit 3-2	R/W	输入捕获 3 预分频器 参考 IC1PRES 描述
CC3SSEL	Bit 1-0	R/W	输入捕获 3 选择 该位定义了通道和使用的输入的方向。 00: 通道配置为输出 01: 通道配置为输入, 捕获源为 I3 10: 通道配置为输入, 捕获源为 I4 11: 通道配置为输入, 捕获源为 ITn 或 I1 的双边沿检出 只有当内部触发输入是通过 TSSEL 位 (GP16C4T_SMCON 寄存器) 选择时, 该模式才运行 注意: 当通道关闭 (GP16C4T_CCEP 寄存器中的 CC3EN = '0'), CC3SSEL 为只写。

18.5.2.13 捕获/比较使能寄存器 (GP16C4T_CCEP)

捕获/比较使能寄存器 (GP16C4T_CCEP)																															
偏移地址: 030 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CC4POL	CC4EN	Reserved	CC3POL	CC3EN	Reserved	CC2POL	CC2EN	Reserved	CC1POL	CC1EN								

Reserved	Bit 31-14	-	保留, 必须保持为复位值
CC4POL	Bit 13	R/W	捕获/比较 4 输出极性 参考 CC1POL 描述
CC4EN	Bit 12	R/W	捕获/比较 4 输出使能 参考 CC1EN 描述
Reserved	Bit 11-10	-	保留, 必须保持为复位值
CC3POL	Bit 9	R/W	捕获/比较 3 输出极性 参考 CC1POL 描述
CC3EN	Bit 8	R/W	捕获/比较 3 输出使能 参考 CC1EN 描述
Reserved	Bit 7-6	-	保留, 必须保持为复位值
CC2POL	Bit 5	R/W	捕获/比较 2 输出极性 参考 CC1POL 描述
CC2EN	Bit 4	R/W	捕获/比较 2 输出使能 参考 CC1EN 描述
Reserved	Bit 3-2	-	保留, 必须保持为复位值
CC1POL	Bit 1	R/W	捕获/比较 1 输出极性 通道配置为输出: 0: CH10 高有效 1: CH10 低有效 通道配置为输入: CC1POL 为触发和捕获操作选择 I1 边沿检出和 I2 边沿检出的有效极性。 0: 正向/上升沿 电路对 In 边沿检出的上升沿敏感 (在复位, 外部时钟或触发模式下, 进行捕获或触发), In 边沿检出不反向 (门控模式或编码器模式下, 进行触发)。 1: 反向/下降沿 电路对 In 边沿检出的下降沿敏感 (在复位, 外部时钟或触发模式下, 进行捕获或触发), In 边沿检出反向 (门控模式或编码器模式下, 进行触发)。
CC1EN	Bit 0	R/W	捕获/比较 1 输出使能 通道配置为输出:

			<p>0: 关闭 - CH10 无效。 1: 开启 - CH10 为对应输出引脚上的输出信号, 由 CC1EN 决定 通道配置为输入: 该位决定了计数值是否能捕获到输入捕获/比较寄存器 1 (GP16C4T_CCVAL1)。 0: 禁止捕获。 1: 使能捕获。</p>
--	--	--	---

18.5.2.14 计数器寄存器 (GP16C4T_COUNT)

计数器寄存器 (GP16C4T_COUNT)																															
偏移地址: 034 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CNTV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CNTV	Bit 15-0	R/W	计数值

18.5.2.15 预分频寄存器 (GP16C4T_PRES)

预分频寄存器 (GP16C4T_PRES)																															
偏移地址: 038 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PSCV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
PSCV	Bit 15-0	R/W	<p>预分频器值 计数器时钟频率 (CK_CNT) = $f_{CK_PSC} / (PSCV [15:0] + 1)$ 每发生一次更新事件 (包括当计数器由 GP16C4T_SGE 寄存器中的 SGU 位清零或当配置为复位模式时, 通过触发控制器清零), PSCV 包含的值将被填入到有效的预分频寄存器内。</p>

18.5.2.16 自动重载寄存器 (GP16C4T_AR)

自动重载寄存器 (GP16C4T_AR)																															
偏移地址: 03C _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ARRV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
ARRV	Bit 15-0	R/W	自动重载值 ARRV 中的值将被载入实际的自动重载寄存器中。 当自动重载值为空, 计数器被屏蔽。

18.5.2.17 捕获/比较寄存器 1 (GP16C4T_CCVAL1)

捕获/比较寄存器 1 (GP16C4T_CCVAL1)																															
偏移地址: 044 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV1															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CCRV1	Bit 15-0	R/W	捕获/比较值 1 如果通道 CCn 配置为输出： CCRVn 中的值将被载入实际的捕获/比较寄存器中（预载值）。 如果在 GP16C4T_CHMRn 寄存器中的预载功能没有选中, CCRVn 中的值将被永久载入；否则, 每当发生更新事件, 预载值将会复制到有效的捕获/比较寄存器中。有效捕获/比较寄存器中包含的值将会与 GP16C4T_COUNT 中的值进行比较, 并在 OCn 上输出。 如果通道 CCn 配置为输入： CCRVn 为由上一个输入捕获事件 (ICn) 传输的计数值。

18.5.2.18 捕获/比较寄存器 2 (GP16C4T_CCVAL2)

捕获/比较寄存器 2 (GP16C4T_CCVAL2)																															
偏移地址: 048 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV2															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CCRV2	Bit 15-0	R/W	捕获/比较值 2 参考 CCRV1 描述

18.5.2.19 捕获/比较寄存器 3 (GP16C4T_CCVAL3)

捕获/比较寄存器 3 (GP16C4T_CCVAL3)																															
偏移地址: 04C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV3															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CCRV3	Bit 15-0	R/W	捕获/比较值 3 参考 CCRV1 描述

18.5.2.20 捕获/比较寄存器 4 (GP16C4T_CCVAL4)

捕获/比较寄存器 4 (GP16C4T_CCVAL4)																															
偏移地址: 050 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CCRV4															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CCRV4	Bit 15-0	R/W	捕获/比较值 4 参考 CCRV1 描述

18.5.2.21 DMA使能寄存器 (GP16C4T_DMAEN)

DMA 使能寄存器 (GP16C4T_DMAEN)																															
偏移地址: 058 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								TRGDMA	COMDMA	CC4DMA	CC3DMA	CC2DMA	CC1DMA	UDMA	

Reserved	Bit 31-7	-	保留, 必须保持为复位值
TRGDMA	Bit 6	R/W	触发DMA请求使能 0: DMA请求禁止 1: DMA请求使能
COMDMA	Bit 5	R/W	COM DMA访问使能 0: DMA请求禁止 1: DMA请求使能
CC4DMA	Bit 4	R/W	捕获/比较值 4 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC3DMA	Bit 3	R/W	捕获/比较值 3 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC2DMA	Bit 2	R/W	捕获/比较值 2 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
CC1DMA	Bit 1	R/W	捕获/比较值 1 DMA 访问使能 0: DMA 请求禁止 1: DMA 请求使能
UDMA	Bit 0	R/W	更新 DMA 请求使能 0: DMA 请求禁止 1: DMA 请求使能

第19章 基本定时器（BS16T）

19.1 概述

基本定时器（BS16T）包含一个 16 位自动重载计数器，该计数器由可配置的预分频器驱动。

通过使用定时器的分频器和 APB 时钟控制器的预分频功能，可对脉冲长度和波形周期进行数微妙到几毫秒的调整。

19.2 特性

- ◆ 16 位自动加载递增计数器
- ◆ 16 位可编程预分频器，可对计数器工作时钟进行 1 到 65536 的任意分频(运行中也可以)
- ◆ 计数上溢更新事件产生中断/DMA 请求

19.3 结构框图

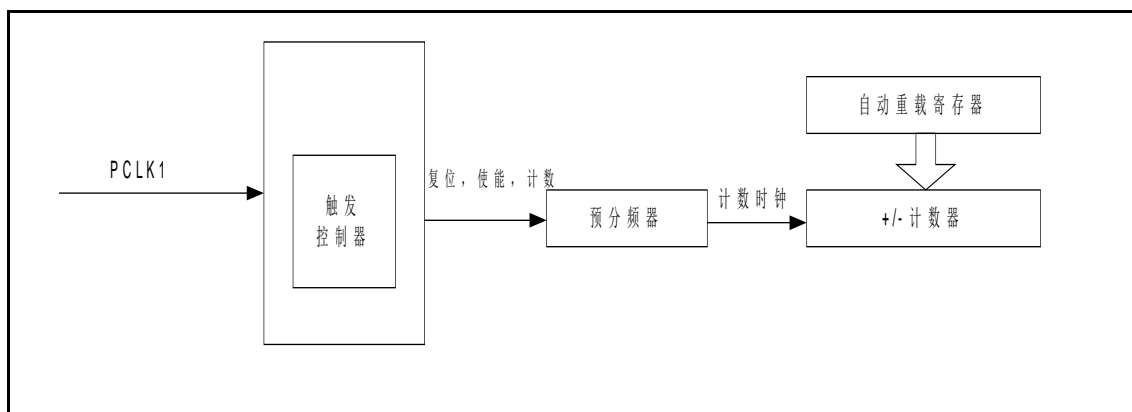


图 19-1 基本定时器电路结构框图

19.4 功能描述

19.4.1 预分频器

定时器包含一个 16-bit 的计数器（BS16T_COUNT），计数时钟由预分频寄存器（BS16T_PRES）进行分频。计数周期由自动重载计数器（BS16T_AR）设定。

自动重载寄存器(BS16T_AR)是一个可缓存的寄存器。当 BS16T_CON1 寄存器的 ARPEN 位复位时，BS16T_AR 寄存器重载功能失效，BS16T_AR 就是有效寄存器；ARPEN 置位时，BS16T_AR 寄存器具有重载功能，产生更新事件（UEV）时，加载值（BS16T_AR 寄存器值）更新到影子寄存器。

当 BS16T_CON1 寄存器中 DISUE 位为 0 时，计数器计数上溢（或递减下溢）时会产生更新事件（UEV）。同样，软件方式也可产生更新事件。BS16T_CON1 寄存器的 CNTEN 置位时，计数器开始计数。

注：计数器在 CNTEN 位置位 1 个时钟周期后开始计数。

预分频器可对定时器工作时钟进行 BS16T_PRES 寄存器值+1 次分频。由于 BS16T_PRES 是一个可重载寄存器，因此，定时器工作时可以对该寄存器进行修改，修改值在下次更新事件（UEV）后有效。

下图给出了定时器运行过程中改变预分频值时计数器的计数情况。

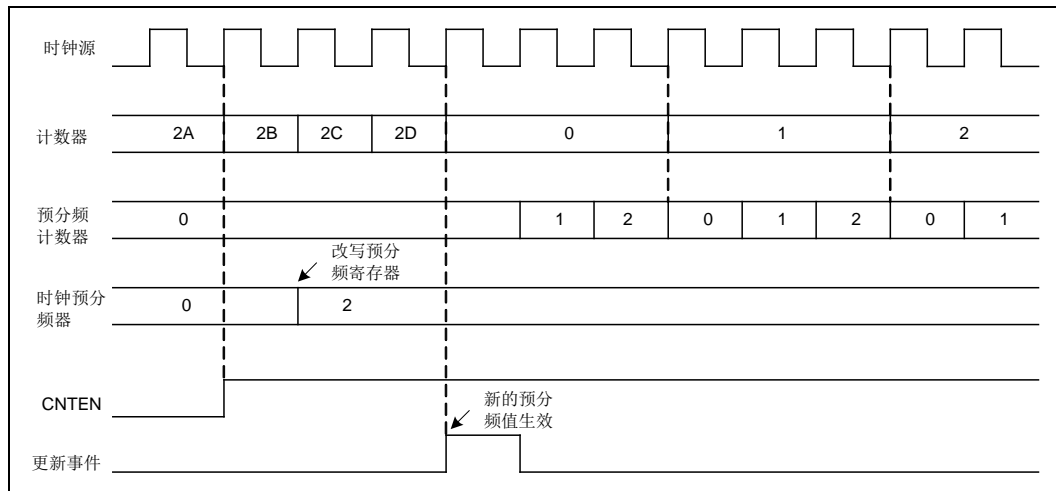


图 19-2 预分频值计数时序图

19.4.2 时钟源

计数时钟由内部时钟源（PCLK1）提供。

CNTEN 位（BS16T_CON1 寄存器）与 SGU 位（BS16T_SGE 寄存器）为实际控制位，这两个位只能软件修改（SGU 位除外，仍硬件自动清除）。一旦 CNTEN 位被写为'1'，预分频器就由内部 PCLK1 提供时钟。

19.4.3 递增计数模式

在递增计数模式中，计数器由 0 开始计数至自动重载值（BS16T_AR 寄存器中的值），然后从 0 开始重新计数并产生一个计数溢出事件。

软件置位 BS16T_CON1 寄存器中的 DISUE 位可关闭更新事件（UEV）的产生。更新事件（UEV）关闭，可避免向预载寄存器写新值的过程中更新影子寄存器。这种情况下，DISUE 位在写'0'之前都不会产生更新事件。正常产生更新事件后，计数器和预载计数器都是从 0 重新开始（但预分频值没有改变）。此外，若置位 BS16T_CON1 寄存器中的 UERSEL 位（更新请求选择），置位 SGU 位时会产生一次更新事件（UEV），但 UEVTIF 标志位不会置位（因此，不会触发中断或 DMA 请求）。

当更新事件发生时，所有寄存器都会被更新且更新标志位（BS16T_RIF 寄存器中的 UEVTIF 位）置位（取决于 UERSEL 位）：

- ◇ 自动重载影子寄存器加载 BS16T_AR 寄存器中的值
- ◇ 预分频器之缓冲器加载 BS16T_PRES 寄存器中的值

19.4.4 调试模式

当微控制器进入调试模式（CPU 内核停止），BS16T 计数器可被设定为停止计数。

19.5 特殊功能寄存器

19.5.1 寄存器列表

BS16T 寄存器列表		
名称	偏移地址	描述
BS16T_CON1	000 _H	控制寄存器 1
Reserved	004 _H	保留
BS16T_IER	00C _H	中断使能寄存器
BS16T_IDR	010 _H	中断禁止寄存器
BS16T_IVS	014 _H	中断有效状态寄存器
BS16T_RIF	018 _H	原始中断标志寄存器
BS16T_IFM	01C _H	中断标志屏蔽寄存器
BS16T_ICR	020 _H	中断清零寄存器
BS16T_SGE	024 _H	事件生成寄存器
BS16T_COUNT	034 _H	计数器
BS16T_PRES	038 _H	预分频器
BS16T_AR	03C _H	自动重载寄存器
BS16T_DMAEN	058 _H	DMA 使能寄存器

19.5.2 寄存器描述

19.5.2.1 控制寄存器 1 (BS16T_CON1)

控制寄存器 1 (BS16T_CON1)																															
偏移地址: 000H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								APREN	Reserved			SPMEN	UERSEL	DISUE	CNTEN

Reserved	Bit 31-8	-	保留, 必须保持为复位值
APREN	Bit7	R/W	自动重载预载使能 0: BS16T_AR 寄存器未缓冲 1: BS16T_AR 寄存器被装入缓冲器
Reserved	Bit 6-4	-	保留, 必须保持为复位值
SPMEN	Bit 3	R/W	单脉冲模式 0: 当发生更新事件时, 计数器不停止。 1: 当发生下一次更新事件 (CEN 位清零) 时, 计数器停止。
UERSEL	Bit 2	R/W	更新请求源 该位由软件置 1 或清零, 来选择 UEV 事件源。 0: 如果更新中断或 DMA 请求使能, 则下述任一事件都可产生更新中断或 DMA 请求: -计数器上溢/下溢 -设置 SGU 位 -从模式控制器产生的更新 1: 如果更新中断或 DMA 请求使能, 仅计数器上溢/下溢才能产生更新中断或 DMA 请求中断
DISUE	Bit1	R/W	更新禁止 该位由软件置 1 或清零来使能/禁止 UEV 事件的发生。 0: UEV 使能. 更新事件 (UEV) 由下列任一事件产生: - 计数器上溢/下溢 -设置 SGU 位 -从模式控制器产生的更新 缓冲寄存器载入他们的预载值。 1: UEV 禁止. 不产生更新事件, 影子寄存器保持他们的值 (AR, PSCV). 如果从从模式控制器接收到硬件复位, 计数器和预分频器将被重新初始化。
CNTEN	Bit0	R/W	CNTEN: 计数器使能

			<p>0: 计数器禁止 1: 计数器使能 注意: 如果软件设置了 CNTEN 位, 外部时钟, 门控模式和编码器模式才能工作。触发模式可由硬件自动设置 CNTEN 位。</p>
--	--	--	--

19.5.2.2 中断使能寄存器 (BS16T_IER)

中断使能寄存器 (BS16T_IEN)																															
偏移地址: 00C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UIT

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UIT	Bit 0	W1	更新中断使能 0: 无效 1: 使能

19.5.2.3 中断禁止寄存器 (BS16T_IDR)

中断禁止寄存器 (BS16T_IDR)																															
偏移地址: 010 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UI

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UI	Bit 0	W1	更新中断禁止 0: 无效 1: 禁止

19.5.2.4 中断有效状态寄存器 (BS16T_IVS)

中断有效状态寄存器 (BS16T_IVS)																															
偏移地址: 014 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UEI

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UEI	Bit 0	R	更新中断有效状态 0: 禁止更新中断 1: 使能更新中断 IER/IDR 写 1 来使能或禁止该位。

19.5.2.5 原始中断标志寄存器 (BS16T_RIF)

原始中断标志 (BS16T_RIF)																																
偏移地址: 018 _H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																															UEVTIF	

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UEVTIF	Bit 0	R	<p>更新中断标志</p> <p>如果更新中断使能, 当发生更新事件, 该标志位由硬件置起。对 BS16T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生更新。</p> <p>1: 更新中断被挂起。当寄存器更新时, 该位被硬件置起:</p> <ul style="list-style-type: none"> -当重复计数器值发生上溢或者下溢(若重复计数器=0, 则更新)和当 BS16T_CON1 寄存器中 DISUE=0 -当使用 BS16T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时, 如果 BS16T_CON1 寄存器中的 UERSEL=0 和 DISUE=0 -当 CNT 由触发事件来重新初始化, 如果 BS16T_CON1 寄存中的 UERSEL=0 和 DISUE=0。

19.5.2.6 中断标志屏蔽寄存器 (BS16T_IFM)

中断标志屏蔽寄存器 (BS16T_IFM)																															
偏移地址: 01C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UEI

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UEI	Bit 0	R	<p>更新中断标志屏蔽</p> <p>如果更新中断使能, 当发生更新事件, 该标志位由硬件置起。对 BS16T_ICR 写 1 来清除原始中断。</p> <p>0: 未发生更新。</p> <p>1: 更新中断被挂起。当寄存器更新时, 该位被硬件置起:</p> <ul style="list-style-type: none"> -当重复计数器值发生上溢或者下溢(若重复计数器=0, 则更新)和当 BS16T_CON1 寄存器中 DISUE=0 -当使用 BS16T_SGE 寄存器中的 SGU 位来由软件重新初始化 CNT 时, 如果 BS16T_CON1 寄存器中的 UERSEL=0 和 DISUE=0 -当 CNT 由触发事件来重新初始化, 如果 BS16T_CON1 寄存中的 UERSEL=0 和 DISUE=0。

19.5.2.7 中断清零寄存器 (BS16T_ICR)

中断清零寄存器 (BS16T_ICR)																															
偏移地址: 020 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UEIC

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UEIC	Bit 0	C_W1	<p>更新中断清零</p> <p>0: 无效</p> <p>1: 更新中断清零 (BS16T_RIF)</p>

19.5.2.8 事件生成寄存器 (BS16T_SGE)

事件生成寄存器 (BS16T_SGE)																															
偏移地址: 024 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SGU

Reserved	Bit 31-1	-	保留, 必须保持为复位值
SGU	Bit 0	W1	更新生成 该位由软件设置, 可由硬件自动清零。 0: 无动作 1: 重新初始化计数器, 更新寄存器。注意, 预分频器也会被清零 (但预分频比不会受到影响)。如果使用中心对齐模式或者 DIR=0 (递增), 则计数器将清零; 否则如果 DIR=1 (递减), 则将使用自动重载入值。

19.5.2.9 计数器 (BS16T_COUNT)

计数器 (BS16T_COUNT)																															
偏移地址: 034 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CNTV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
CNTV	Bit 15-0	R/W	计数值

19.5.2.10 预分频器 (BS16T_PRES)

预分频器 (BS16T_PRES)																															
偏移地址: 038 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PSCV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
PSCV	Bit 15-0	R/W	<p>预分频器值</p> <p>计数器时钟频率 (CK_CNT) = fCK_PSC / (PSCV[15:0] + 1)</p> <p>每发生一次更新事件 (包括当计数器由 BS16T_SGE 寄存器中的 SGU 位清零或当配置为复位模式时, 通过触发控制器清零), PSCV 包含的值将被填入到有效的预分频寄存器内。</p>

19.5.2.11 自动重载寄存器 (BS16T_AR)

自动重载寄存器 (BS16T_AR)																															
偏移地址: 03C _H																															
复位值: 00000000_00000000_11111111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ARRV															

Reserved	Bit 31-16	-	保留, 必须保持为复位值
ARRV	Bit 15-0	R/W	<p>自动重载值</p> <p>AR 中的值将被载入实际的自动重载寄存器中。当自动重载值为空, 计数器被屏蔽。</p>

19. 5. 2. 12 DMA使能寄存器 (BS16T_DMAEN)

DMA 使能寄存器 (BS16T_DMAEN)																															
偏移地址: 058 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															UDEN

Reserved	Bit 31-1	-	保留, 必须保持为复位值
UDEN	Bit 0	R/W	DMA 访问使能 0: DMA 访问禁止 1: DMA 访问使能

第20章 串行总线 (I2C0~1)

20.1 概述

I2C 是两线双向的串行传输总线，提供了一种简单有效的方法来实现设备之间的数据交换。

标准 I2C 是一个多主机总线且包括冲突检测与仲裁，提供了标准模式 (Sm)、快速模式 (Fm) 与极快速模式 (Fm+) 供用户选择，并且也提供 SMBus (系统管理总线) 与 PMBus (电源管理总线)。

芯片内部共有 2 个两线双向串行传输总线 (I2C0~1)。

20.2 特性

- ◆ 可配置为主机或从机
- ◆ 多主机模式
- ◆ 标准模式 (速率最高 100 KHz)
- ◆ 快速模式 (速率最高 400 KHz)
- ◆ 极快速模式 (速率最高 1 MHz)
- ◆ 支持 7 位与 10 位地址模式
- ◆ 支持 2 组 7 位从机地址 (2 个地址，其中一个包括屏蔽功能)
- ◆ 支持所有 7 位地址接听模式
- ◆ 支持广播模式
- ◆ 可配置的建立时间和保持时间
- ◆ 可选择时钟延长
- ◆ 可配置数字滤波器
- ◆ 支持 DMA 传输
- ◆ 支持 SMBus 标准功能
- ◆ 硬件 PEC (封包错误检查) 产生与 ACK 控制
- ◆ 命令与数据应答控制
- ◆ 支持地址解析通讯协议
- ◆ 支持 SMBus 警报
- ◆ 支持侦测超时与空闲功能
- ◆ 支持 PMBus rev 1.1 标准功能

20.3 结构图

关于 I2C 电路模块的结构如下图所示：

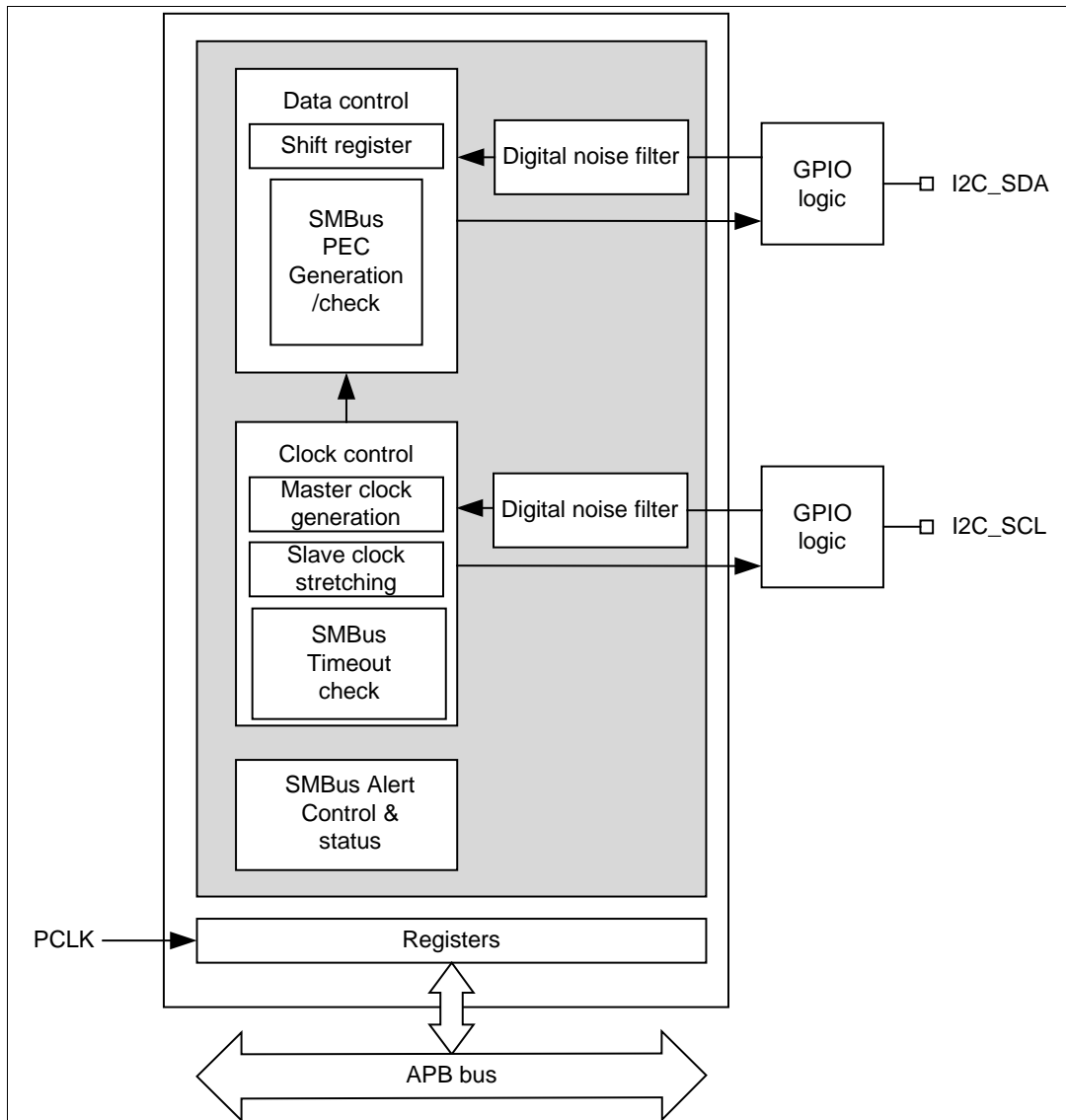


图 20-1 I2C 结构图

20.4 功能描述

I2C 接口支持多个通信检测和数据传输中断,可由软件开启或关闭。该接口通过数据引脚(SDA)和时钟引脚(SCL)连接到 I2C 总线,可以连接标准模式(最高 100 KHz),快速模式(最高 400 KHz)或极快速模式(最高 1 MHz)的 I2C 总线。

该接口也可通过数据引脚(SDA)和时钟引脚(SCL)连接到 SMBus。

如果支持 SMBus 功能: 可以使用 SMBus 警报引脚(SMBA)。

20.4.1 I2C总线协议

I2C 是一种双线双向串行总线,可在设备之间提供简单有效的数据交换方法。标准 I2C 是真正的多主机总线,包括冲突检测和仲裁,如果两个或多个主机同时尝试控制总线,其仲裁可防止数据损坏。数据在主机和从机之间逐字节同步传输到串行数据(SDA)和串行时钟(SCL)总线,每个数据字节长度为 8 位。

20.4.1.1 START和STOP条件协议

I2C 规范将起始条件定义为 SDA 从高电平到低电平的转换,且 SCL 为高电平。起始条件由主机产生,表示总线从空闲状态转换为活动状态。每个数据位有一个 SCL 时钟脉冲,首先发送数据最高位 MSB。每个传送的字节后面都有一个应答位。当 SCL 为高电平时,SDA 的电平变化被解释为通信启动或结束命令(START 或 STOP),因此在数据通信过程中,SDA 仅可以在 SCL 的低电平期间改变,并且必须在 SCL 的高电平期间保持稳定。

当总线空闲时,SCL 和 SDA 都通过总线上的外部上拉电阻拉高。当主机想要在总线上开始传输时,主机发出 START 信号(SDA 从高电平跳变到低电平,且 SCL 为高电平)。当主机想要停止传输时,主机发出 STOP 信号(SDA 从低电平跳变到高电平,且 SCL 为高电平)。START 和 STOP 信号的时序如下图所示。

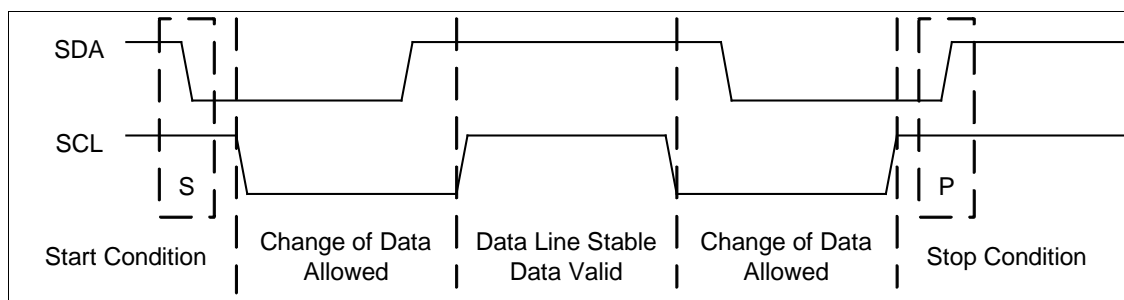


图 20-2 START 和 STOP 信号

20.4.1.2 应答位

数据传输必须带有应答位，应答位相关的时钟脉冲由主机产生。主机在应答时钟脉冲期间释放 SDA（高电平），从机必须在应答时钟脉冲低电平期间下拉 SDA，以便在时钟脉冲的高电平期间保持稳定的低电平。当然，还必须考虑应答位的建立和保持时间。通常，已经被寻址的从机必须在接收到每个字节后产生应答，除非数据以 CBUS 地址开始。

当从机不应答从机地址时（例如，由于它正在执行某些实时功能而无法接收或发送），从机必须将数据线保持为高电平。然后主机可以产生 STOP 条件以中止传输，或者重复 START 条件以开启新传输。

如果从机应答了从机地址，但是在传输过程中的某个时间不能再接收数据字节，则从机需在随后的第一个字节接收时产生非应答来表示，从机使数据线保持高电平，主机产生 STOP 或重复 START 条件。

如果主机想停止接收数据，它必须在最后一个字节接收时产生非应答，来向从机发送数据结束信号，从机必须释放数据线，以允许主机产生 STOP 或重复 START 条件。

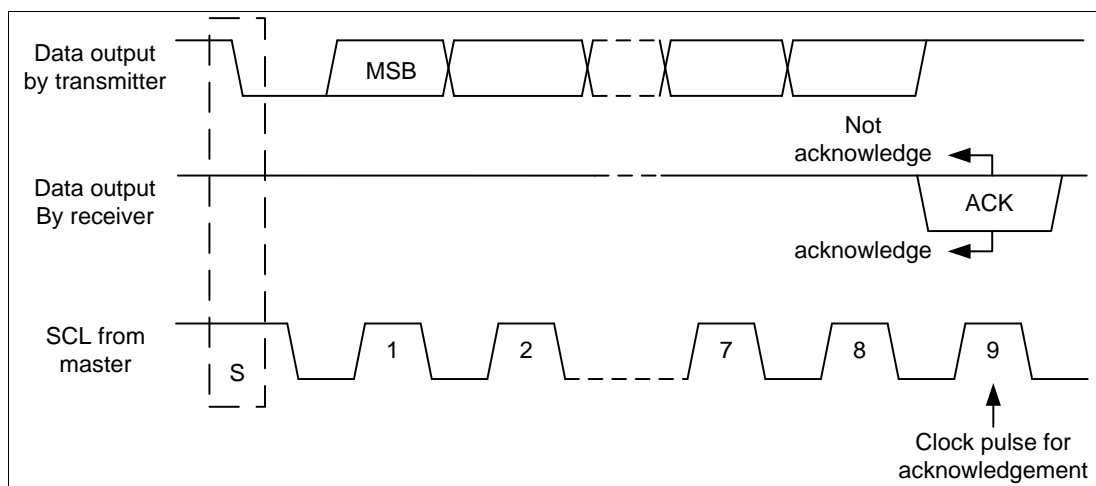


图 20-3 I2C 总线上的应答位

20.4.1.3 I2C寻址从协议

I2C 支持两种地址格式：7 位地址格式和 10 位地址格式。在 7 位地址格式时，第一个字节的前 7 位（位<7:1>）设置从地址，LSB 位（位<0>）是 R/W 位，当 LSB 位设置为 0 时，表示主机写入从机，当 LSB 位设置为 1 时，表示主机由从机读取。数据首先传输最高有效位（MSB）。在 10 位地址格式时，传输两个字节以设置 10 位地址。第一个字节的传输包含以下位定义：前 5 位（位<7:3>）通知从机这是一个 10 位地址传输，接着后两位（位<2:1>）为从机地址位<9:8>，LSB 位（位<0>）是 R/W 位；传输的第二个字节设置从机地址位<7:0>。

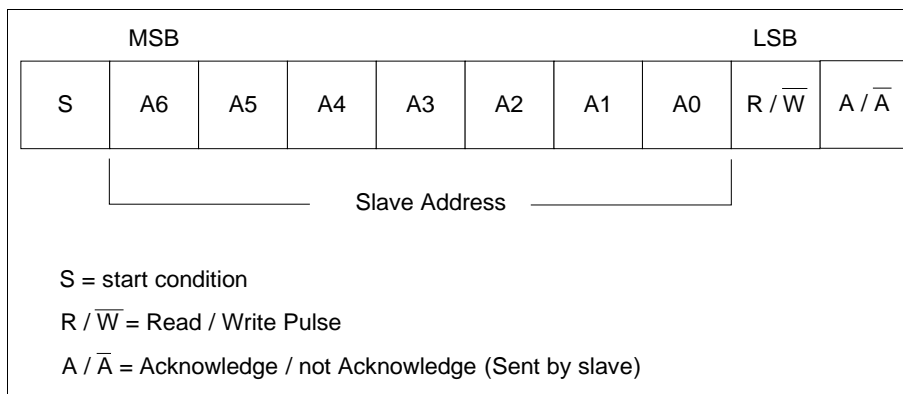


图 20-4 7 位地址格式

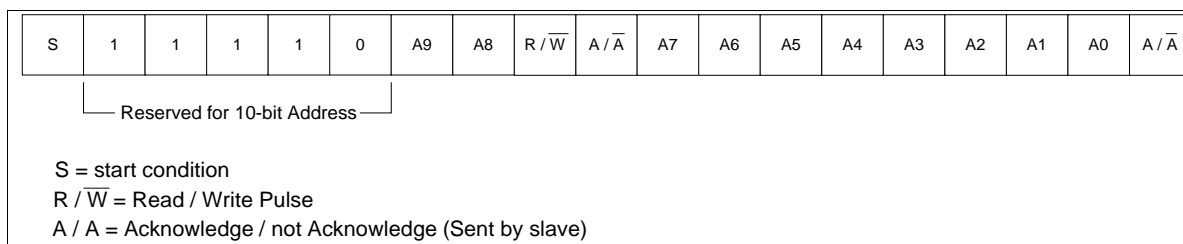


图 20-5 10 位地址格式

从地址	R/W位	描述
0000 000	0	广播地址
0000 000	1	START字节
0000 001	X	CBUS地址
1111 0XX	X	10位从机寻址

表 20-1 第一个字节中的位定义

20.4.1.4 I2C发送和接收协议

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。在主机发送地址和 R/W 位或主机向从机发送一个字节数据后，接收的从机必须响应应答信号，当接收的从机没有响应应答脉冲时，主机通过发出 STOP 条件来中止传输，从机应使 SDA 保持高电平，以便主机可以中止传输。如果主机正在发送数据，则接收的从机在接收到每个数据字节后用应答脉冲响应主机。传输格式如下：

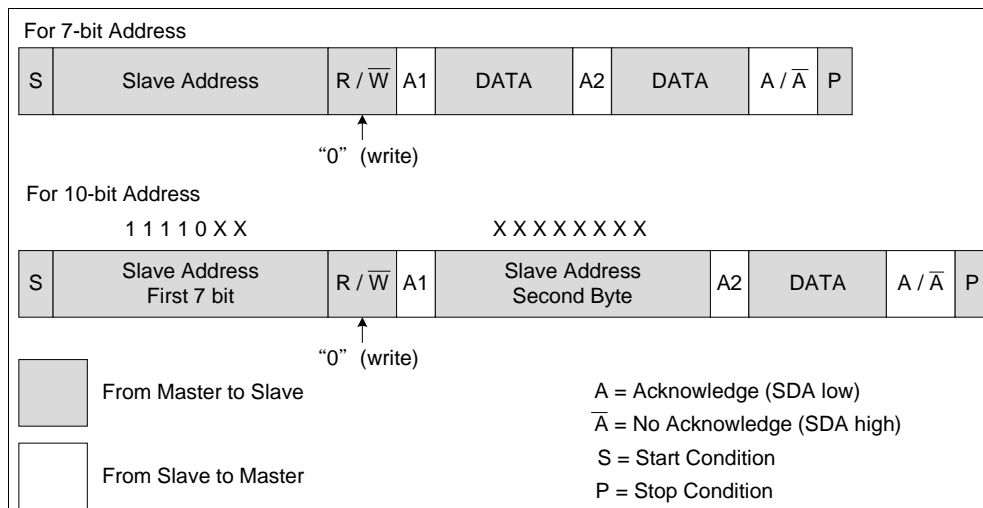


图 20-6 主机发送数据

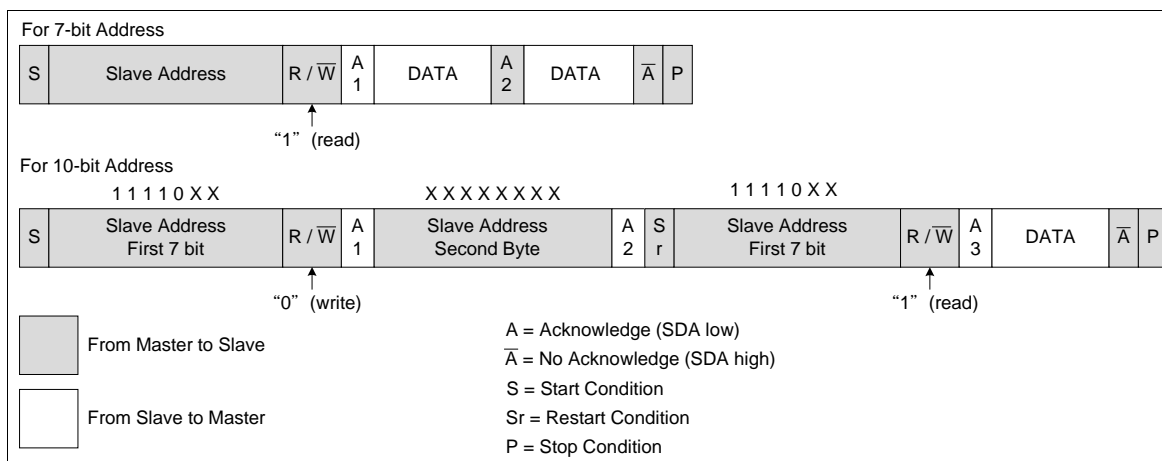


图 20-7 主机接收数据

20.4.2 I2C时钟要求

I2C 模块工作由 I2CCLK 提供时钟。

I2CCLK 周期 T_{I2CCLK} 必须符合以下条件：

$$T_{I2CCLK} < (T_{LOW} - T_{Filter}) / 4 \text{ 与 } T_{I2CCLK} < T_{HIGH}$$

其中：

T_{LOW} ：SCL 低电平时间。

T_{HIGH} ：SCL 高电平时间。

T_{Filter} ：开启数字滤波器时带来的延迟时间。数字滤波器延迟为 $DNF \times T_{I2CCLK}$ 。

PCLK 时钟周期 T_{PCLK} 必须符合以下条件：

$$T_{PCLK} < 4/3 T_{SCL}$$

T_{SCL} ：SCL 周期。

20.4.3 数据传输

SDA 数据线的每个字节必须为 8 位长度。每次传输的字节数不受限制，每个字节后面都必须有一个应答位，首先传输最高位（MSB）。如果从机来不及接收或发送另一个完整的数据字节，则从机可以将 SCL 保持为低电平以强制主机进入等待状态，直到从机准备好接收另一个数据字节并释放 SCL 时，数据传输继续。

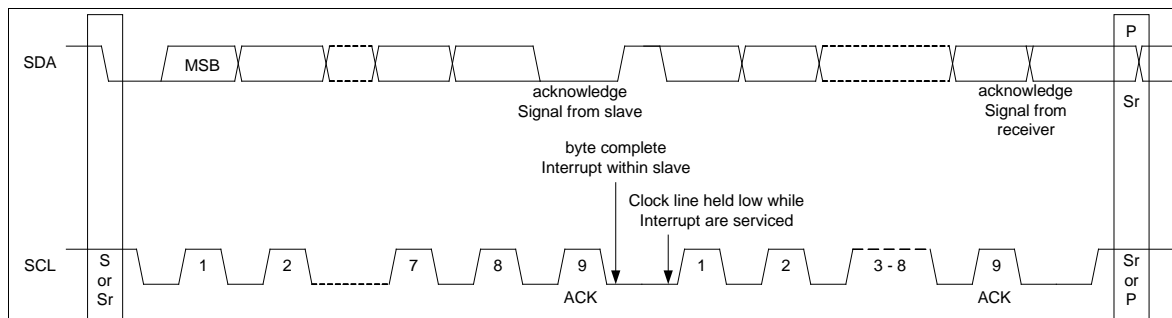


图 20-8 I2C 总线上的数据传输

接收

SDA 输入移位寄存器，在第 8 个 SCL 脉冲之后（接收到完整的数据字节），如果 RXNE=1，表示接收器非空，尚未读取先前接收的数据字节，此时 SCL 会在第 9 个 SCL 脉冲之后延长，直到读取 I2C_RXDATA 寄存器，使得接收器为空。

发送

如果发送器不为空（TXE=0），则第 9 个 SCL 脉冲（应答脉冲）之后将其内容传输到发送移位寄存器中，然后移位寄存器的内容会在 SDA 上发送。如果 TXE=1，表示发送器尚未写入数据，则 SCL 将被拉低（应答脉冲之后），直到通过写入 I2C_TXDATA 寄存器，将数据传输到发送器。

硬件传输管理

I2C 模块包含一个字节计数器，用于管理数据字节传输过程，例如：在主模式下产生 NACK、STOP 和 RESTART，从接收模式下的 ACK 控制，支持 SMBus 功能时的 PEC 产生/检查。

字节计数器通常在主模式下使用，在从机模式下默认关闭，但可以通过设置 I2C_CON1 寄存器中的 SBC 位开启。设置 I2C_CON1 寄存器和 I2C_CON2 寄存器中的 NBYTES<15:0>位可选择要传输的字节数，如果要传输的字节数大于 65535，或者接收时想要控制接收数据字节的应答值，则必须通过设置 I2C_CON2 寄存器中的 RELOAD 位为 1，来选择重载模式。在此模式下，当传送完毕 NBYTES 中所设置的字节数时，会置起 I2C_RIF 寄存器中的 TCR 中断标志位，如果 I2C_IER 寄存器中的 TCR 位为 1，则会产生中断请求。只要置起了 TCR 标志位，SCL 就会被延长。当软件对 NBYTES 写入非零值时，硬件会自动清除 TCR 标志位。

当 NBYTES 计数器重载最后一个字节数时，必须清除 RELOAD 位。

当主模式下 RELOAD 位为 0 时，计数器可用于 2 种模式：

- ◆ **自动结束模式**（I2C_CON2 寄存器中的 AUTOEND 位为 1）。在此模式下，一旦传输完毕 NBYTES<15:0>位所设置的字节数，主机就会自动发送 STOP 条件。

- ◆ **软件结束模式** (I2C_CON2 寄存器中的 AUTOEND 位为 0)。在此模式下,一旦传输完毕 NBYTES<15:0>位所设置的字节数,就会置起 I2C_RIF 寄存器中的 TC 中断标志位,如果 I2C_IER 寄存器中的 TC 位为 1,则会产生中断请求。只要置起了 TC 标志位, SCL 就会被延长。当设置 I2C_CON2 寄存器中的 START 或 STOP 位为 1 时, TC 标志位自动清零。当主机要发送 RESTART 条件时,必须使用此模式。

20.4.4 I2C从机模式

I2C 从机初始化

为了在从机模式下工作,必须至少开启一个从机地址。两个地址寄存器 I2C_ADDR1 和 I2C_ADDR2 可用于设置从机自身地址 OA1 和 OA2。

- ◆ 通过设置 I2C_ADDR1 寄存器中的 OA1MODE 位,可以选择本机地址 1 的地址模式为 7 位或 10 位,并通过 OA1 位设置本机地址 1,然后再设置 OA1EN 位为 1 开启本机地址 1。
- ◆ 如果需要额外的从地址,可以设置第二个从地址 OA2。通过设置 I2C_ADDR2 寄存器中的 OA2MSK<2:0>位,可以屏蔽最多 7 个 OA2 地址低位。对于设置为 1 至 6 的 OA2MSK 位,分别对应于仅使用 OA2<7:2>、OA2<7:3>、OA2<7:4>、OA2<7:5>、OA2<7:6>或 OA2<7>与收到的地址相比较。一旦 OA2MSK 位不等于 0, OA2 的地址比较器就会排除未被应答的 I2C 保留地址 (0000 XXX 和 1111 XXX)。如果 OA2MSK=7,则应答所有接收到的 7 位地址 (保留地址除外)。OA2 只能是 7 位地址。

当 OA2MSK=0,并且 I2C_ADDR1 或 I2C_ADDR2 寄存器中的本机地址开启时,则可应答这些保留地址 (0000 XXX 和 1111 XXX)。通过设置 I2C_ADDR2 寄存器中的 OA2EN 位为 1 开启本机地址 OA2。

- ◆ 通过设置 I2C_CON1 寄存器中的 GCEN 位为 1 开启广播地址。

当其中一个本机地址开启并匹配时,地址匹配中断标志状态被置为 1,如果 I2C_IER 寄存器中的 ADDR 位为 1,则会产生中断请求。

默认情况下,从机使用其时钟延长功能,这表示它在需要时可将 SCL 拉到低电平,以执行软件操作。如果主机不支持时钟延长,则需将 I2C_CON1 寄存器中 NOSTRETCH 位设置为 1。

收到地址匹配中断后,如果开启了多个地址,用户必须读取 I2C_STAT 寄存器中的 ADDCODE<6:0>位,以检查匹配的地址及检查 DIR 位来确认数据传输方向。

从机有时钟延长 (NOSTRETCH=0)

I2C 从机在以下情况下延长 SCL 时钟:

- ◆ 在从机发送过程中,如果当前的数据发送完成,但没有在 I2C_TXDATA 寄存器中写入新数据,从机会拉低 SCL 时钟。当数据写入 I2C_TXDATA 寄存器后,将释放此延长。
- ◆ 在从机接收过程中,如果当前数据接收完成,但尚未读取 I2C_RXDATA 寄存器中的旧数据,从机会拉低 SCL 时钟。当读取 I2C_RXDATA 寄存器后,将释放此延长。
- ◆ 从机字节控制模式下,当发生 TCR=1 时,重载模式 (SBC 位=1 且 RELOAD 位=1),表

示数据字节已被传输。此时通过在 NBYTES<15:0>字段中写入非零值清除 TCR，释放该延长。

- ◆ 从机应答控制模式下，当收到地址或数据时，每个字节的第 8 和第 9 个 SCL 脉冲之间将会延长 SCL。当设置应答位更新时，将会释放该延长并响应应答脉冲。
- ◆ 在 SCL 下降沿检测后，I2C 拉低延长 SCL 的时间为
$$[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times T_{I2CCLK}$$

从机没有时钟延长 (NOSTRETCH=1)

当 I2C_CON1 寄存器中的 NOSTRETCH 位为 1 时，I2C 从机不会延长 SCL。

- ◆ 在从机发送过程中，必须在发送的第一个 SCL 脉冲之前，将数据写入 I2C_TXDATA 寄存器，否则产生发送器下溢错误，I2C_RIF 寄存器中的下溢中断标志位 TXUD 被置为 1，如果 I2C_IER 寄存器中的 TXUD 位为 1，则会产生中断请求。
- ◆ 在从机接收过程中，必须在下一个数据字节的第 9 个 SCL 脉冲（应答脉冲）之前，从 I2C_RXDATA 寄存器中读取旧数据，否则产生接收器溢出，I2C_RIF 寄存器中的溢出中断标志位 RXOV 被置为 1，如果 I2C_IER 寄存器中的 RXOV 位为 1，则会产生中断请求。

从机字节控制模式

为了在从机接收模式下允许字节 ACK 控制，需设置 I2C_CON1 寄存器中的 SBC 位为 1 来开启从机字节控制模式，这需要符合 SMBus 标准。

必须选择重载模式 (RELOAD=1)，以便在从机接收模式中允许字节控制。如果要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每个字节接收后重载到 0x1。当接收到该字节时，TCR 位被设置为 1，在第 9 个 SCL 脉冲之后将 SCL 拉低，用户可以从 I2C_RXDATA 寄存器读取数据，然后通过设置 I2C_CON2 寄存器中的 NACK 位决定下一个收据是否应答。通过将 NBYTES 设置为非零值来释放 SCL 延长：发送应答或不应答。

NBYTES 可以加载大于 0x1 的值，在这种情况下，接收流程在 NBYTES 数据接收期间是连续的。

注 1：当关闭 I2C 时，才可以设置 SBC 位。当 TCR=1 时，可以更改 RELOAD 位。

注 2：从机字节控制模式与 NOSTRETCH 模式不能同时使用，不允许在 NOSTRETCH=1 时开启 SBC 位。



图 20-9 从机初始化流程图

从机发送

在接收到匹配地址时，I2C_RIF 寄存器中的 ADDR 位被置为 1，此时若 TX 已准备好要发送的数据时，从机会通过内部移位寄存器将 TX 中的数据字节发送到 SDA；若 TX 此时为空，从机会延长 SCL 低电平时间，直到 TX 通过 I2C_TXDATA 寄存器写入发送数据为止。

当发送成功，主机会回答应信号，当主机响应 NACK 时，I2C_RIF 寄存器中的 NACK 位被置为 1，此时从机会自动释放 SCL 与 SDA 总线，让主机能够发送后续的 STOP 或 RESTART 命令。

当主机发送 STOP 时，I2C_RIF 寄存器中的 STOP 位被设置为 1，并结束通信，等待下一次收到匹配地址。但如果 TX 内还有尚未传送的数据字节，则可以选择下一次地址匹配时，继续发送数据。

当从机字节控制开启时，可设置 I2C_CON1 寄存器和 I2C_CON2 寄存器中的 NBYTES 位选择需要传送的字节数。

注：当 NOSTRETCH 模式开启时，SCL 时钟无法被延长，则待发送的数据需要提前写入 I2C_TXDATA 寄存器。

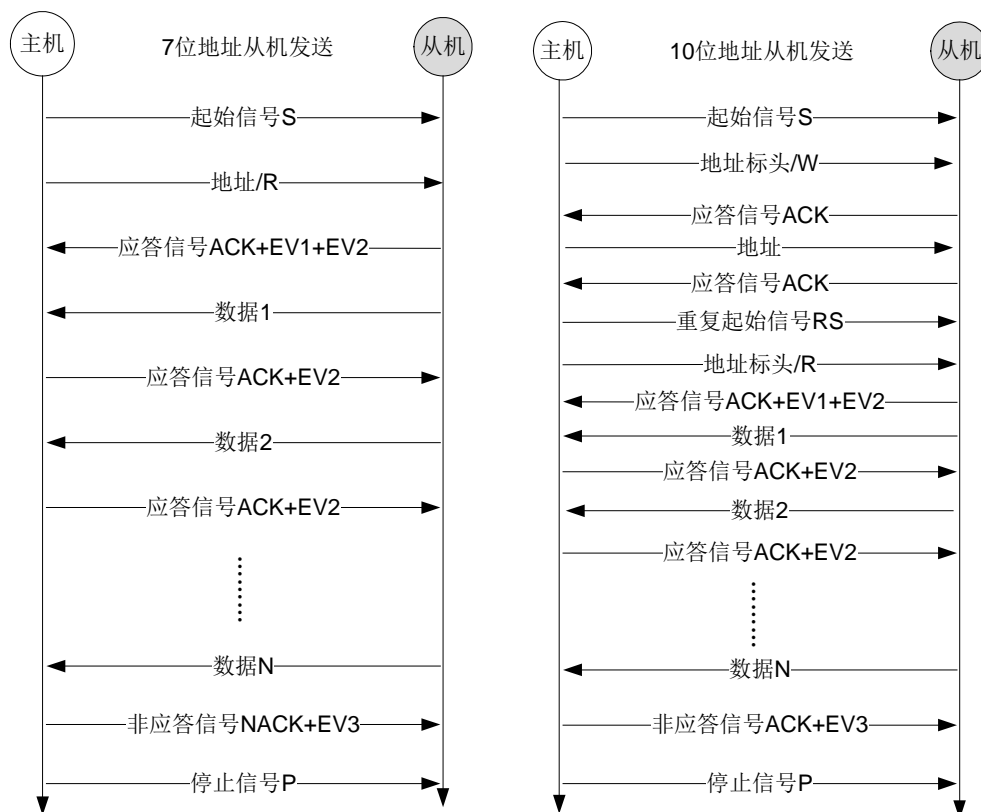


图 20-10 从机发送的传输序列图

注 1: S=起始位, RS=重复起始位, P=停止位, ACK=应答, NACK=非应答
 注 2: EV1=当收到匹配地址时, I2C_RIF 寄存器中的 ADDR 位被设置为 1, 设置 I2C_ICR 寄存器中的 ADDR 位为 1 来清除中断。
 注 3: EV2=判断 I2C_STAT 寄存器中的 TXE 位, 若为空则写入数据至 I2C_TXDATA 寄存器。

注 4: EV3=当收到非应答信号 NACK 时, I2C_RIF 寄存器中的 NACK 位被置为 1, 设置 I2C_ICR 的 NACK 位为 1 来清除中断。此时还需判断 TX 是否还有尚未发送的数据字节, 若有则软件需要额外处理。

注 5: 如果在当前字节传输结束之前尚未写入下一个字节, 导致 TX 为空时, EV2 事件将会延长 SCL 时钟的低电平时间。

从机接收

在接收到匹配地址时, I2C_RIF 寄存器中的 ADDR 位被置为 1, 从机此时会通过内部移位寄存器将 SDA 上的数据字节保存到 RX 中, 无论 RX 是否为空, 从机都会自动发送应答信号, 如果 RX 接收器为非空, 则在应答信号发送后, 会延长 SCL 时钟低电平时间, 等待软件读取 RX 字节后, 才会释放 SCL 时钟。

当主机发送 STOP 时, I2C_RIF 寄存器中的 STOP 位被置为 1, 并结束通信, 等待下一次收到匹配地址。

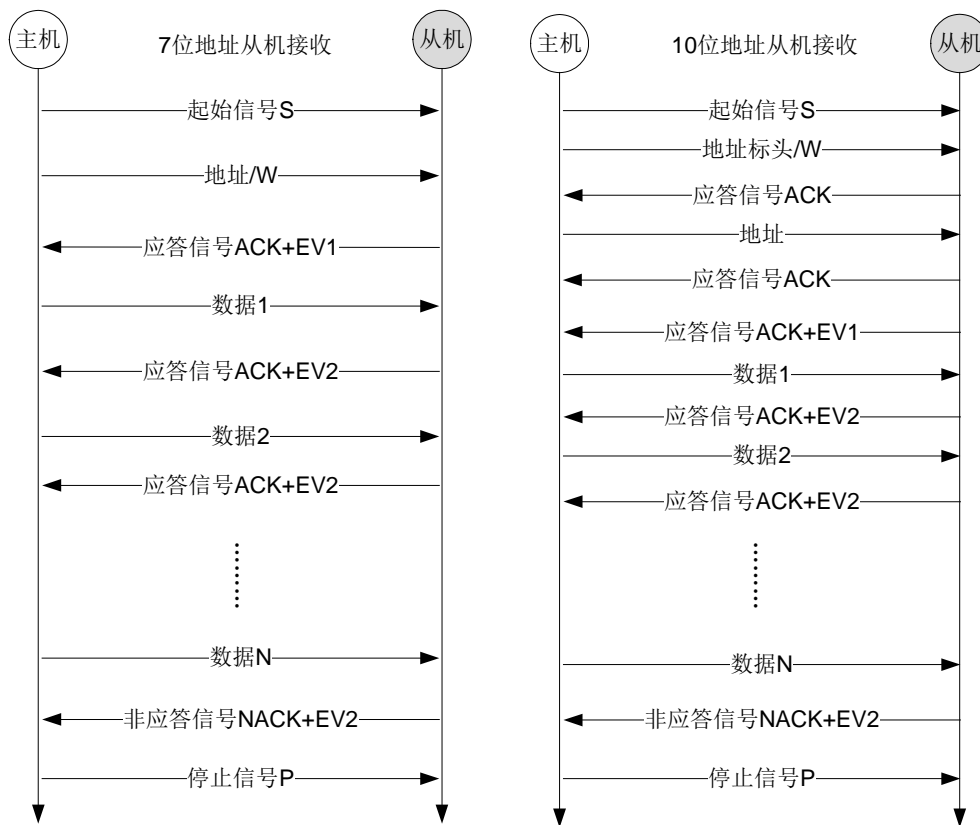


图 20-11 从机接收的传输序列图

注 1: S=起始位, RS=重复起始位, P=停止位, ACK=应答, NACK=非应答

注 2: EV1=当收到匹配地址时, I2C_RIF 寄存器中的 ADDR 位被置为 1, 设置 I2C_ICR 寄存器中的 ADDR 位为 1 来清除中断。

注 3: EV2=判断 I2C_STAT 寄存器中的 RXNE 位, 若非空则读取 I2C_RXDATA 寄存器数据。

注 4: 如果在当前字节接收结束时, RX 接收器为非空, EV2 事件将会延长 SCL 时钟低电平时间。

20.4.5 I2C主机模式

I2C 主机初始化

在开启 I2C 主机之前，必须先通过设置 I2C_TIMINGR 寄存器中的 SCLH 位和 SCLL 位来配置 I2C 主机时钟，实现同步时钟机制以支持多主机环境和从机时钟延长。

SCL 高低电平计数：

- ◆ 从内部检测到 SCL 低电平开始，使用 SCLL 计数器计数低电平时钟。
- ◆ 从内部检测到 SCL 高电平开始，使用 SCLH 计数器计数高电平时钟。

基于 SCL 下降沿，I2C 开始低电平内部检测，在 T_{SYNC1} 延迟后检测到自己的 SCL 低电平，一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器中的 SCLL<7:0>位设置的值，I2C 就会将 SCL 释放为高电平。

基于 SCL 上升沿，I2C 开始高电平内部检测，在 T_{SYNC2} 延迟后检测到自己的 SCL 高电平，一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器中的 SCLH<7:0>位设置的值，I2C 就会将 SCL 拉到低电平。

因此，主机时钟周期为：

$$T_{\text{SCL}} = T_{\text{SYNC1}} + T_{\text{SYNC2}} + \{ [(SCLH+1) + (SCLL+1)] \times (\text{PRESC}+1) \times T_{\text{I2CCLK}} \}$$

T_{SYNC1} 的持续时间取决于以下参数：

- ◆ SCL 下降斜率
- ◆ 开启数字滤波器时，引起的输入延迟： $\text{DNF} \times T_{\text{I2CCLK}}$
- ◆ 由于 SCL 与 I2CCLK 时钟同步导致的延迟：最多 3 个 I2CCLK 周期

T_{SYNC2} 的持续时间取决于以下参数：

- ◆ SCL 上升斜率
- ◆ 开启数字滤波器时，引起的输入延迟： $\text{DNF} \times T_{\text{I2CCLK}}$
- ◆ 由于 SCL 与 I2CCLK 时钟同步导致的延迟：最多 3 个 I2CCLK 周期

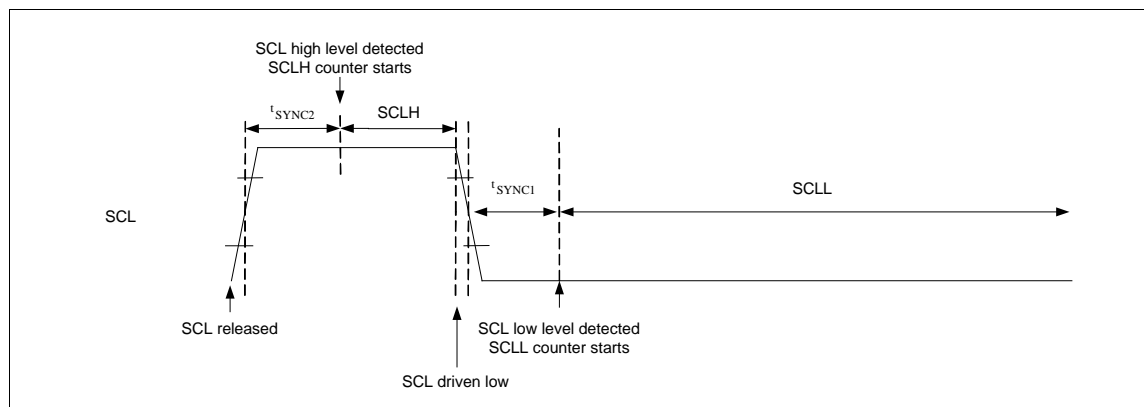


图 20-12 主机时钟产生

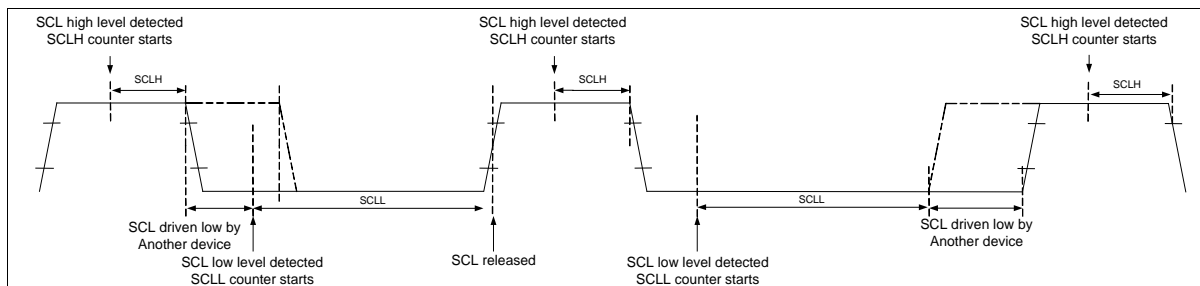


图 20-13 SCL 主机时钟同步

主机通信初始化

在开启通信前，必须先设置 I2C_CON2 寄存器中的寻址从机参数：

- ◆ 寻址模式（7 位或 10 位）：ADD10
- ◆ 要寻址的从机地址：SADD <9: 0>
- ◆ 传输方向：RD_WRN
- ◆ 如果是 10 位地址寻址模式，则需配置 HEAD10R 位，选择是否发送完整的地址序列，或者仅发送地址的前 7 位。
- ◆ 要传输的字节数：NBYTES<15:0>。如果字节数等于或大于 65535，则 NBYTES<15:0> 可以先填入 0xFFFF。

然后再设置 I2C_CON2 寄存器中的 START 位为 1，当 START 位为 1 时，不允许更改所有上述位。一旦检测到总线空闲（BUSY=0）后，主机就会自动发送 START 条件，再发送从机地址。

在仲裁丢失的情况下，主机自动切换回从机模式，如果作为寻址从机，则可以应答自己的地址。

注 1：无论接收到的应答值是什么，当总线上的从机地址发送时，START 位由硬件清除，如果发生仲裁丢失，则 START 位也由硬件清除。在 10 位寻址模式下，当从机地址前 7 位被从机 NACK 时，主机将自动重复起始从机地址传输，直到收到 ACK。如果在 START 位为 1 时 I2C 被寻址为从机（ADDR=1），则 I2C 切换到从机模式，当 ADDR 位为 1 时，START 位清零。

注 2：对重复起始条件应用上述 START 相同的过程，在这种情况下，BUSY=1。

初始化主机 10 位从地址寻址接收模式

- ◆ 如果被寻址的从机地址采用 10 位格式，用户可以通过清零 I2C_CON2 寄存器中的 HEAD10R 位来选择发送完整的从机地址读序列。在这种情况下，主机在 START 位为 1 后自动发送以下完整序列：起始位+从机地址标头/写+从机地址第 2 字节+重复起始+从机地址标头/读；
- ◆ 如果主机寻址 10 位地址从机，将数据发送到该从机，然后从同一从机读取数据，则必须先完成主机发送流程，然后使用从地址标头设置重复起始读。在这种情况下，主机发送以下序列：重复起始+从机地址标头/读。

主机发送

在主机开始发送之前，需先设置 I2C_CON1 和 I2C_CON2 寄存器中的 NBYTES 位，当传输字节大于 65535 时，需要额外设置 RELOAD 位。在此配置下，当 NBYTES 配置的数据字节数传输完成后，I2C_RIF 寄存器中的 TCR 位被置为 1，此时 SCL 时钟会保持低电平，直到 NBYTES 位重新写入新的数值以及对 I2C_TXDATA 寄存器写入发送数据后，才会继续进行传输。

当从机回应 NACK 时，I2C_RIF 寄存器中的 NACK 位被置为 1，并且接下来主机会自动发送停止信号 STOP。

当从机响应 ACK，且 RELOAD=0，NBYTES 所配置的数据字节数都已经传完时，会有以下情况：

- ◆ 若 AUTOEND=1，此时主机会自动发送停止信号 STOP。
- ◆ 若 AUTOEND=0，此时主机会将 SCL 时钟保持低电平，等待软件控制后续操作：
 - ◇ RESTART：设置 I2C_CON2 寄存器中的 START 位为 1，会发送重复起始信号。
 - ◇ STOP：设置 I2C_CON2 寄存器中的 STOP 位为 1，会发送停止信号。

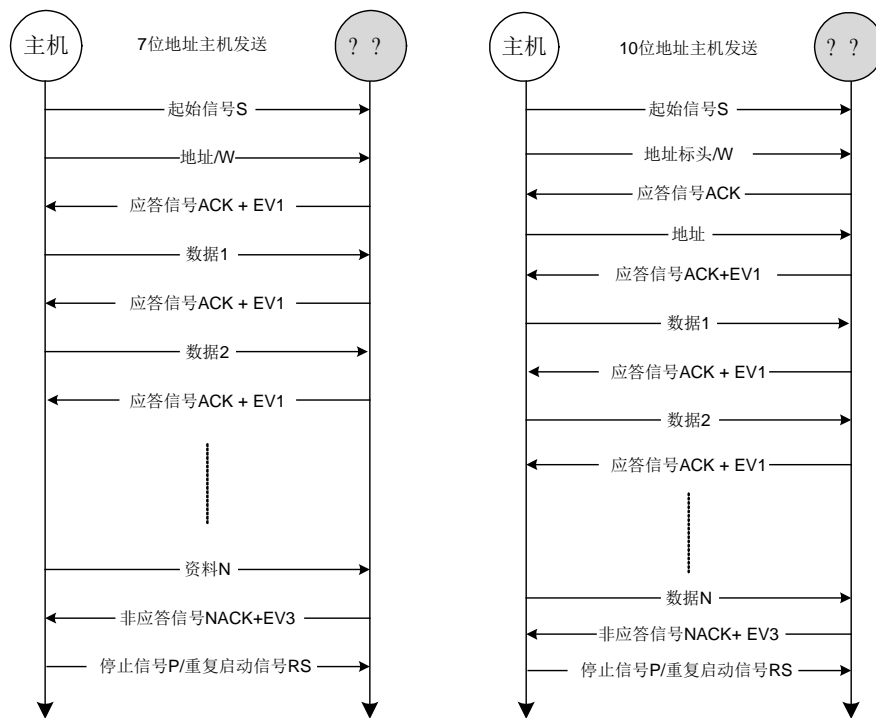


图 20-14 主机发送的传输序列图

注 1：S=起始位，RS=重复起始位，P=停止位，ACK=应答，NACK=非应答

注 2：EV1=传输尚未完成事件，判断 I2C_STAT 寄存器中的 TXE 位，若为空写入数据至 I2C_TXDATA 寄存器。

注 3：EV2=当收到应答信号 ACK 时，若传输已完成，后续操作为停止信号 STOP 或重复起始信号 RESTART。

注 4：EV3=当收到非应答信号 NACK 时，后续操作为停止信号 STOP。

注 5：如果在当前字节发送结束之前尚未写入下一个字节，导致 TX 为空时，EV1 事件将会延长 SCL 时钟低电平时间。

主机接收

在主机开始接收之前，需先设置 I2C_CON1 和 I2C_CON2 寄存器中的 NBYTES 位，当接收字节大于 65535 时，需要额外配置 RELOAD 位。在此配置下，当 NBYTES 配置的数据字节数接收完成后，I2C_RIF 寄存器中的 TCR 位被设置为 1，此时 SCL 时钟会保持低电平，直到 NBYTES 位重新写入新的数值后，才会继续进行接收。

当 RELOAD 位=0 并且 NBYTES 所配置的数据字节数都已经接收完时，会有以下情况：

- ◆ 若 AUTOEND=1，此时会自动发送非应答信号 NACK 与停止信号 STOP。
- ◆ 若 AUTOEND=0，此时会自动发送非应答信号 NACK，并将 SCL 时钟保持低电平，等待软件控制后续操作：
 - ◇ RESTART：对 I2C_CON2 寄存器中的 START 位设置为 1，会发送重复起始信号。
 - ◇ STOP：对 I2C_CON2 寄存器中的 STOP 位设置为 1，会发送停止信号。

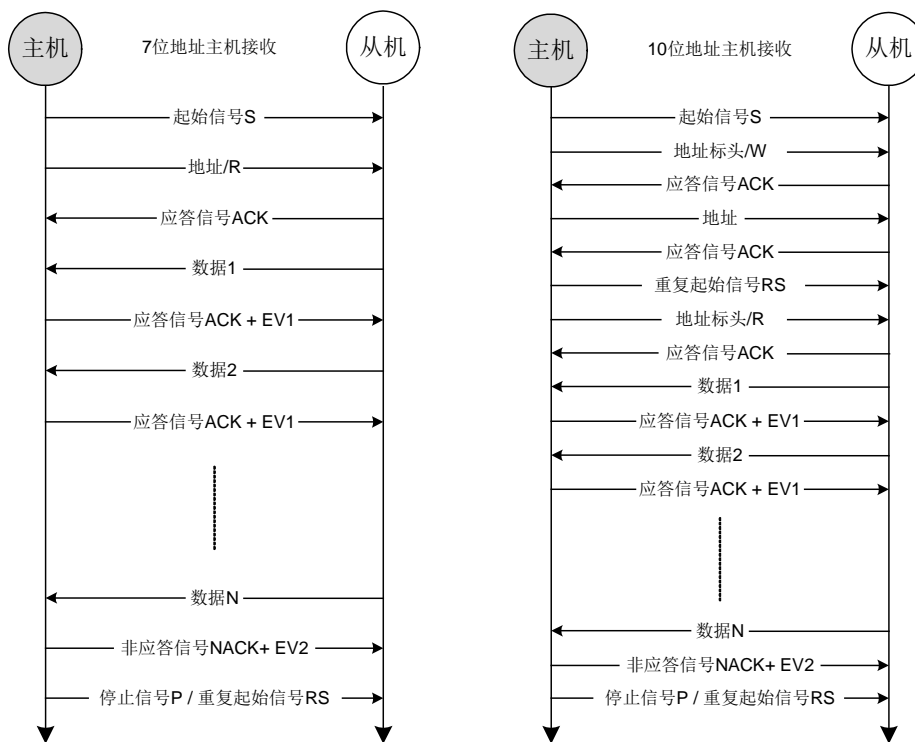


图 20-15 主机接收的传输序列图

注 1：S=起始位，RS=重复起始位，P=停止位，ACK=应答，NACK=非应答

注 2：EV1=传输尚未完成事件，判断 I2C_STAT 寄存器中的 RXNE 位，若非空则读取 I2C_RXDATA 寄存器。

注 3：EV2=传输完成事件，会自动发送非应答信号 NACK，后续根据软件配置来决定发送停止信号 STOP 还是重复起始信号 RESTART。

注 4：如果在当前字节接收结束之前尚未读取之前的数据，导致 RX 非空时，EV1 事件将会延长 SCL 时钟低电平时间。

20.4.6 I2C_TIMINGR寄存器的配置的例子

下表提供了如何对 I2C_TIMINGR 进行设置以获得符合 I2C 规范的时序示例。

参数	标准模式 (Sm)		快速模式 (Fm)	极快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
T _{SCLL}	200x250 ns=50 μs	20x250 ns=5.0 μs	10x125 ns=1250 ns	7x125 ns=875 ns
SCLH	0xC3	0xF	0x3	0x3
T _{SCLH}	196x250 ns=49 μs	16x250 ns=4.0μs	4x125ns=500ns	4x125ns=500ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~2000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x1	0x0
T _{SDADEL}	2x250 ns=500 ns	2x250 ns=500 ns	1x125 ns=125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
T _{SCLDEL}	(0xC7-0x4)x250 ns ⁽⁵⁾	(0x13-0x4)x250 ns ⁽⁵⁾	(0xC7-0x4)x250 ns ⁽⁵⁾	(0xC7-0x4)x250 ns ⁽⁵⁾

表 20-2 F_{I2CCLK} = 8 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟, SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。示例中 T_{SCL} 的值仅供参考。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns, 示例中的值约为 1000 ns。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns, 示例中的值约为 750 ns。
4. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 500 ns, 示例中的值约为 625 ns。
5. T_{SCLDEL} = (SCLL-SCLDEL) x T_{PRESC}。

参数	标准模式 (Sm)		快速模式 (Fm)	极快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
T _{SCLL}	200x250 ns=50 μs	20x250 ns=5.0 μs	10x125 ns=1250 ns	6x62.5 ns=312.5 ns
SCLH	0xC3	0xF	0x3	0x2
T _{SCLH}	196x250 ns=49 μs	16x250 ns=4.0μs	4x125ns=500ns	3x62.5ns=187.5ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~1000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x2	0x0
T _{SDADEL}	2x250 ns=500 ns	2x250 ns=500 ns	2x125 ns=250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
T _{SCLDEL}	(0xC7-0x4)x250 ns ⁽⁵⁾	(0x13-0x4)x250 ns ⁽⁵⁾	(0x9-0x3)x125 ns ⁽⁵⁾	(0x4-0x2)x62.5 ns ⁽⁵⁾

表 20-3 F_{I2CCLK} = 16 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟, SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。示例中 T_{SCL} 的值仅供参考。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns, 示例中的值约为 1000 ns。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns, 示例中的值约为 750 ns。
4. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 250 ns, 示例中的值约为 500 ns。
5. T_{SCLDEL} = (SCLL-SCLDEL) x T_{PRESC}。

参数	标准模式 (Sm)		快速模式 (Fm)	极快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
T _{SCLL}	200x250 ns=50 μs	20x250 ns=5.0 μs	10x125 ns=1250 ns	4x125 ns=500 ns
SCLH	0xC5	0x11	0x5	0x1
T _{SCLH}	198x250 ns=49.5 μs	18x250 ns=4.5μs	6x125ns=750ns	2x125ns=250ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽²⁾	~1000ns ⁽³⁾
SDADEL	0x2	0x2	0x3	0x0
T _{SDADEL}	2x250 ns=500 ns	2x250 ns=500 ns	3x125 ns=375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
T _{SCLDEL}	(0xC7-0x4)x250 ns ⁽⁴⁾	(0x13-0x4)x250 ns ⁽⁴⁾	(0x9-0x3)x125 ns ⁽⁴⁾	(0x3-0x1)x125 ns ⁽⁴⁾

表 20-4 F_{I2CCLK} = 48 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟, SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。示例中 T_{SCL} 的值仅供参考。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 83.3 ns, 示例中的值约为 500 ns。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 83.3 ns, 示例中的值约为 250 ns。
4. T_{SCLDEL} = (SCLL-SCLDEL) x T_{PRESC}。

参数	标准模式 (Sm)		快速模式 (Fm)	极快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xF	0xF	0x8	0x8
SCLL	0xE0	0x18	0x9	0x3
T _{SCLL}	225x222 ns=49.95 μs	25x222 ns=5.55 μs	10x125 ns=1250 ns	4x125 ns=500 ns
SCLH	0xDF	0x12	0x7	0x1
T _{SCLH}	224x222 ns=49.7 μs	19x222 ns=4.2μs	8x125ns=1000ns	2x125ns=250ns
T _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽³⁾	~2500 ns ⁽³⁾	~1000 ns ⁽³⁾
SDADEL	0x2	0x2	0x3	0x0
T _{SDADEL}	2x222 ns=444 ns	2x222 ns=444 ns	3x125 ns=375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
T _{SCLDEL}	(0xE0-0x4)x222 ns ⁽⁴⁾	(0x18-0x4)x222 ns ⁽⁴⁾	(0x9-0x3)x125 ns ⁽⁴⁾	(0x3-0x1)x125 ns ⁽⁴⁾

表 20-5 F_{I2CCLK} = 72 MHz 的时序设置示例

1. 由于 SCL 内部检测延迟, SCL 周期 T_{SCL} 大于 T_{SCLL} + T_{SCLH}。示例中 T_{SCL} 的值仅供参考。
2. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 55.5 ns, 示例中的值约为 350 ns。
3. T_{SYNC1} + T_{SYNC2} 最小值为 4 x T_{I2CCLK} = 55.5 ns, 示例中的值约为 250ns。
4. T_{SCLDEL} = (SCLL-SCLDEL) x T_{PRESC}。

20.4.7 SMBus具体功能

介绍

系统管理总线（SMBus）是一个双线制接口，各设备可通过它在彼此之间或者与系统的其余部分进行通信。它以 I2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。系统可使用 SMBus 与设备进行消息传递，而无需切换各个控制线。

系统管理总线规范涉及三类设备：

- ◆ 从设备用于接收或响应命令。
- ◆ 主设备用于发出命令、生成时钟和中止传输。
- ◆ 主机，专用的主设备，可提供连接系统 CPU 的主接口。主机必须具有主从设备功能，并且必须支持 SMBus 主机通知协议。

系统中只允许存在一个主机。该外设可以配置为主设备或从设备，也可以配置为主机。

总线协议

对于任何给定的设备，有 11 种可能的命令协议，可以使用 11 个协议中的任何一个或全部来进行通信。协议包括快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、进程调用、块读取、块写入和块写入块读取进程调用，这些协议应由用户软件实现。有关这些协议的更多详细信息，请参阅 SMBus 规范 2.0 版。

地址解析通讯协议（ARP）

可以通过为每个从设备动态分配新的唯一地址来解决 SMBus 从地址冲突。为了提供隔离每个设备以便进行地址分配的机制，每个设备必须具有唯一的设备标识符（UDID），该标识符是一个 128 位的数字，由软件实现。

支持地址解析通讯协议(ARP)。通过设置 I2C_CON1 寄存器中的 SMBDEN 位为 1 开启 SMBus 从设备地址（0b1100 001）。ARP 命令由用户通过软件实现，仲裁也在从机模式下执行以支持 ARP。

有关 SMBus 地址解析通讯协议的更多详细信息，请参阅 SMBus 规范 2.0 版。

命令和数据接收应答控制

SMBus 接收器必须能够 NACK 每个接收到的命令或数据。为了在从机模式下允许 ACK 控制，必须设置 I2C_CON1 寄存器中的 SBC 位为 1 来开启从机字节控制模式。

主机通知协议

该外设通过设置 I2C_CON1 寄存器中的 SMBHEN 位为 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址（0b0001 000）。使用此协议时，设备充当主机，主机充当从机。

SMBus 警报

支持 SMBus ALERT 可选信号。仅从设备可以通过 SMBALERT 引脚向主机发送信号选择通信。主机处理中断并同时通过警报响应地址（0b0001 100）访问所有 SMBALERT 设备。只有拉低 SMBALERT 的设备才会应答警报响应地址。

当配置为从设备（SMBHEN=0）时，通过设置 I2C_CON1 寄存器中的 ALERTEN 位为 1，可将 SMBA 引脚拉低，警报响应地址同时开启。

当配置为主机（SMBHEN=1）时，如果在 SMBA 引脚上检测到下降沿且 ALERTEN=1，则在 I2C_RIF 寄存器中的 ALERT 位被置为 1。如果 I2C_IER 寄存器中的 ALERT 位为 1，则会产生中断请求。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也会被视为高电平。

如果不需要 SMBus ALERT 引脚，当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

数据包错误检查

SMBus 规范中引入了一种数据包错误检查机制，以提高可靠性和通信稳健性。通过在每次消息传输结束时附加分组错误代码（PEC）来实现分组错误检查。通过对所有消息字节（包括地址和读/写位）使用 $C_{(x)} = X^8 + X^2 + X + 1$ CRC-8 多项式来计算 PEC。外设嵌入了硬件 PEC 计算器，当接收到的字节与硬件计算的 PEC 不匹配时，允许自动发送非应答。

超时

该外设嵌入了硬件定时器，并符合 SMBus 规范 2.0 版中定义的 3 个超时。

标记	参数	范围		单位
		最小	最大	
T _{TIMEOUT}	检测时钟低电平超时	25	35	ms
T _{LOW : SEXT} ⁽¹⁾	累积时钟低电平延长时间（从设备）	-	25	ms
T _{LOW : MEXT} ⁽²⁾	累积时钟低电平延长时间（主设备）	-	10	ms

表 20-6 SMBus 超时规格

1. T_{LOW : SEXT} 是允许给定从设备在一条消息中，从初始 START 到 STOP 的延长时钟周期的累积时间。另一个从设备或主设备也可能延长时钟，导致组合时钟低电平延长时间大于 T_{LOW : SEXT}，因此该参数是在从设备作为全速主设备的唯一目标的情况下测量的。
2. T_{LOW : MEXT} 是允许主设备在从 START 到 ACK，ACK 到 ACK 或 ACK 到 STOP 定义消息的每个字节内延长其时钟周期的累积时间。从设备或另一个主设备也可能延长时钟，导致组合时钟低电平时间大于给定字节上的 T_{LOW : MEXT}，因此使用全速从设备作为主设备的唯一目标来测量该参数。

总线空闲检测

如果总线检测到时钟和数据信号已经持续高电平并且 T_{IDLE} 大于 T_{HIGH.MAX}，则主设备可以认为总线是空闲的。

该时序参数涵盖了主机已动态添加到总线，并且可能未检测到 SMBCLK 或 SMBDAT 线路上的状态转换的情况。在这种情况下，主设备必须等待足够长的时间以确保当前没有进行传输。

SMBus 支持硬件总线空闲检测。

20.4.8 SMBus初始化

除了 I2C 初始化之外，还必须进行一些其他特定的初始化以执行 SMBus 通信：

特定地址（从机模式）

如果需要，应开启特定的 SMBus 地址。

通过设置 I2C_CON1 寄存器中的 SMBDEN 位为 1 开启 SMBus 设备从机地址(0b1100 001)。

通过设置 I2C_CON1 寄存器中的 SMBHEN 位为 1 开启 SMBus 主机从机地址(0b0001 000)。

通过设置 I2C_CON1 寄存器中的 ALERTEN 位为 1 开启报警响应地址 (0b0001100)。

数据包错误检查

通过设置 I2C_CON1 寄存器中的 PECEN 位为 1 开启 PEC 计算。然后通过硬件字节计数器管理 PEC 传输：I2C_CON1 和 I2C_CON2 寄存器中的 NBYTES<15:0>位。必须先设置 PECEN 位，然后再开启 I2C。

PEC 传输由硬件字节计数器管理，因此在从机模式下连接 SMBus 时必须设置 SBC 位。在设置 PECBYTE 位为 1 且 RELOAD 位清零后，传输 NBYTES-1 个数据字节后传输 PEC，如果设置了 RELOAD 位，则 PECBYTE 无效。

注：开启 I2C 时，不允许更改 PECEN 设置。

超时检测

通过设置 I2C_TIMEOUTR 寄存器中的 TIMEOUTEN 位和 TEXTEN 位为 1 来开启超时检测。定时器必须按以下的方式设置。

◆ T_{TIMEOUT} 检查

为了开启 T_{TIMEOUT} 检查，必须设置 TIMEOUTA<11:0>位为定时器的重载值，以检查 T_{TIMEOUT} 参数。必须将 TIDLE 位设置为 0 才能检测 SCL 低电平超时，然后通过设置 I2C_TIMEOUTR 寄存器中的 TIMEOUTEN 位来开启定时器。如果 SCL 在大于(TIMEOUTA + 1)× 2048 × T_{I2CCCLK} 的时间内被拉低，则 I2C_RIF 寄存器中的 TOUT 位被置为 1。

注：当设置 TIMEOUTEN 位为 1 时，不允许更改设置 TIMEOUTA<11:0>位和 TIDLE 位。

◆ T_{LOW:SEXT} 和 T_{LOW:MEXT} 检查

根据外设是配置为主机还是从机，必须配置 12 位 TIMEOUTB 定时器，以便检查从机的 T_{LOW:SEXT} 和主机的 T_{LOW:MEXT}。由于标准仅指定最大值，因此使用者可以为两者选择相同的值，然后通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来启用定时器。

注：当设置 TEXTEN 位为 1 时，不允许更改设置 TIMEOUTB<11:0>位。

总线空闲检测

为了开启 T_{IDLE} 检查，必须设置 $TIMEOUTA<11:0>$ 位为定时器的重载值，以获得 T_{IDLE} 参数。必须设置 $TIDLE$ 位为 1 开启 SCL 和 SDA 高电平超时功能。

然后通过设置 $I2C_TIMEOUTR$ 寄存器中的 $TIMEOUTEN$ 位为 1 来开启定时器。如果 SCL 和 SDA 都保持高电平的时间大于 $(TIMEOUTA + 1) \times 4 \times T_{I2CCLK}$ ，则在 $I2C_RIF$ 寄存器中的 $TOUT$ 位被置为 1。

注：设置 $TIMEOUTEN$ 时，不允许更改设置 $TIMEOUTA$ 位和 $TIDLE$ 位。

20.4.9 SMBus: I2C_TIMEOUTR 寄存器配置的例子

◆ 将 $T_{TIMEOUT}$ 的最大持续时间配置为 25 ms:

F_{I2CCLK}	$TIMEOUT<11:0>$	$TIDLE$	$TIMEOUTEN$	$T_{TIMEOUT}$
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$
32 MHz	0x186	0	1	$391 \times 2048 \times 31.25 \text{ ns} = 25 \text{ ms}$

表 20-7 各种 I2CCLK 频率的 $TIMEOUTA$ 设置示例 ($T_{TIMEOUT} = 25 \text{ ms}$)

◆ 将 $T_{LOW:SEXT}$ 和 $T_{LOW:MEXT}$ 的最大持续时间配置为 8 ms

F_{I2CCLK}	$TIMEOUT<11:0>$	$TXMEOUTEN$	$T_{LOW:SEXT}$
8 MHz	0x1F	1	$32 \times 2048 \times 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 \times 2048 \times 62.5 \text{ ns} = 8 \text{ ms}$
32 MHz	0x7C	1	$125 \times 2048 \times 31.25 \text{ ns} = 8 \text{ ms}$

表 20-8 各种 I2CCLK 频率的 $TIMEOUTB$ 设置示例

◆ 将 T_{IDLE} 的最大持续时间配置为 50 μs

F_{I2CCLK}	$TIMEOUT<11:0>$	$TIDLE$	$TIMEOUTEN$	T_{IDLE}
8 MHz	0x63	1	1	$100 \times 4 \times 125 \text{ ns} = 50 \mu\text{s}$
16 MHz	0xC7	1	1	$200 \times 4 \times 62.5 \text{ ns} = 50 \mu\text{s}$
32 MHz	0x18F	1	1	$400 \times 4 \times 31.25 \text{ ns} = 50 \mu\text{s}$

表 20-9 各种 I2CCLK 频率的 $TIMEOUTA$ 设置示例 ($T_{IDLE} = 50 \mu\text{s}$)

20.4.10 DMA请求

使用 DMA 发送

通过设置 I2C_CON1 寄存器中的 TXDMAEN 位为 1，可以开启 DMA 进行发送。只要 TXE 位被设置为 1，就会使用 DMA 外配置的 SRAM 区域加载数据到 I2C_TXDATA 寄存器。只有数据可通过 DMA 发送。

使用 DMA 接收

通过设置 I2C_CON1 寄存器中的 RXDMAEN 位为 1，可以开启 DMA 进行接收。只要 RXNE 位被设置为 1，就会将 I2C_RXDATA 寄存器的数据加载到使用 DMA 外配置的 SRAM 区域。只有数据（包括 PEC）可通过 DMA 接收。

20.4.11 通信错误类型

以下是可能导致通信失败的错误情况，当发生通信失败时，用户通过查询状态标志来判断发生了哪种错误类型。

总线错误（BERR）

总线错误是指在总线传输地址或数据期间检测到 START 或 STOP 条件，仅当 I2C 作为主机或寻址从机进行传输时（不在从机模式下的地址阶段），才会设置总线错误标志。

如果在从机模式下检测到错误的 START 或 RESTART，则 I2C 会进入地址识别状态，就像正确的 START 条件一样。

检测到总线错误时，I2C_RIF 寄存器中的 BERR 位被置为 1，如果 I2C_IER 寄存器中的 BERR 位为 1，则会产生中请求。

仲裁丢失（ARLO）

仲裁丢失为当在 SDA 上发送高电平时，但在 SCL 上升沿却采样到低电平。

- ◆ 在主机模式下，在地址阶段，数据阶段和数据应答阶段检测仲裁丢失。在这种情况下，SDA 和 SCL 被释放，START 控制位由硬件清零，主机自动切换到从机模式。
- ◆ 在从机模式下，会在数据阶段和数据应答阶段检测仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 被释放。

当检测到仲裁丢失时，I2C_RIF 寄存器中的 ARLO 位被置为 1，如果 I2C_IER 寄存器中的 ARLO 位为 1，则会产生中断请求。

接收溢出 / 发送下溢错误（RXOV / TXUD）

当 NOSTRETCH = 1 时，在从机模式下检测到溢出或下溢错误：

- ◆ 当接收到新字节且接收器非空时。新接收的字节将会丢失，并且自动发送 NACK 作为对新字节的应答。
- ◆ 在发送时如果待发送的新字节且尚未写入发送器，则会发送 0xFF。

当检测到接收溢出或发送下溢错误时，在 I2C_STAT 寄存器中的 TXUD 位与 RXOV 位被置为 1，如果 I2C_IER 寄存器中的 TXUD 位或 RXOV 位为 1，则会产生中断请求。

20.4.12 I2C中断

I2C 中断由以下寄存器控制。

◆ 中断控制 (IER, IDR, IVS)

通过设置 I2C 中断开启寄存器 I2C_IER 中的各位为 1 来开启中断功能，通过设置 I2C 中断关闭寄存器 I2C_IDR 中的各位为 1 来关闭中断功能。IER 和 IDR 寄存器只能写入，上述寄存器中的结果可由中断功能有效状态寄存器 I2C_IVS 表示，IVS 寄存器只能读取。

◆ 原始中断标志寄存器 (RIF)

I2C 原始中断标志寄存器 I2C_RIF 是一个只能读取的寄存器，用于读取模块的中断状态。该寄存器中的位表示 I2C 中断的真实状态。当出现以下条件时，I2C 可以产生中断：

- ◇ SMBus 警报
- ◇ 发生超时
- ◇ PEC 错误
- ◇ 仲裁丢失
- ◇ 总线错误
- ◇ 发送完成并重载
- ◇ 发送完成
- ◇ 检测 STOP
- ◇ 收到非应答
- ◇ 地址匹配
- ◇ 接收器不为空
- ◇ 发送器为空
- ◇ 发生接收溢出或发送下溢

◆ 中断标志屏蔽寄存器 (IFM)

◇ I2C 中断标志屏蔽寄存器 I2C_IFM 用于读取模块的标志位中断状态，表示是哪个中断。IFM 中的每个位是 IVS 和 RIF 中各个位的逻辑与。

◆ 中断清除寄存器 (ICR)

通过设置 I2C 中断清除寄存器 I2C_ICR 中的各位为 1，可以清除相应的中断。

20.4.13 调试模式

当微控制器进入调试模式 (CPU 内核停止运行)，根据调试控制 (DBG) 章节中 DBG_APB2FZ 寄存器的配置，可选择 SMBus 超时计数器继续正常计数或停止计数。

20.5 特殊功能寄存器

20.5.1 寄存器列表

I2C 寄存器列表		
名称	偏移地址	描述
I2C_CON1	0000 _H	I2C 控制寄存器 1
I2C_CON2	0004 _H	I2C 控制寄存器 2
I2C_ADDR1	0008 _H	I2C 本机地址寄存器 1
I2C_ADDR2	000C _H	I2C 本机地址寄存器 2
I2C_TIMINGR	0010 _H	I2C 时钟寄存器
I2C_TIMEOUTR	0014 _H	I2C 超时寄存器
I2C_STAT	0018 _H	I2C 状态寄存器
I2C_PECR	0020 _H	I2C PEC 寄存器
I2C_RXDATA	0024 _H	I2C 接收器数据寄存器
I2C_TXDATA	0028 _H	I2C 发送器数据寄存器
I2C_IER	002C _H	I2C 中断开启寄存器
I2C_IDR	0030 _H	I2C 中断关闭寄存器
I2C_IVS	0034 _H	I2C 中断功能有效状态寄存器
I2C_RIF	0038 _H	I2C 原始中断标志寄存器
I2C_IFM	003C _H	I2C 中断标志屏蔽寄存器
I2C_ICR	0040 _H	I2C 中断清除寄存器

20.5.2 寄存器描述

20.5.2.1 I2C控制寄存器 1 (I2C_CON1)

I2C 控制寄存器 1 (I2C_CON1)																															
偏移地址: 000H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBYTES				PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	Reserved	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Reserved	DNF			Reserved						PE							

NBYTES	Bit 31-24	R/W	接收/发送的数据字节数 参照I2C_CON2.NBYTES描述
PECEN	Bit 23	R/W	PEC 开启位 0: 关闭 PEC 计算 1: 开启 PEC 计算 注意: 如果不支持 SMBus 功能, 则该位保留并由硬件强制为 0。
ALERTEN	Bit 22	R/W	SMBus 警报开启位 设备模式 (SMBHEN = 0): 0: 释放 SMBA 引脚为高电平且关闭警报响应地址标头 0001100x, 回应 NACK。 1: 将 SMBA 引脚驱动为低电平且开启警报响应地址标头 0001100x, 回应 ACK。 主机模式 (SMBHEN = 1): 0: 不支持 SMBus 警报引脚 (SMBA)。 1: 支持 SMBus 警报引脚 (SMBA)。 注: 当 ALERTEN=0 时, SMBA 引脚可用作标准 GPIO。如果不支持 SMBus 功能, 则该位保留并由硬件强制为 0。
SMBDEN	Bit 21	R/W	SMBus 设备从机地址启用位 0: 关闭设备从机地址 0b1100001x, 回应 NACK。 1: 开启设备从机地址 0b1100001x, 回应 ACK。 注: 如果不支持 SMBus 功能, 则该位保留并由硬件强制为 0。
SMBHEN	Bit 20	R/W	SMBus 主机地址启用位 0: 关闭主机地址 0b0001000x, 回应 NACK。 1: 开启主机地址 0b0001000x, 回应 ACK。 注: 如果不支持 SMBus 功能, 则该位保留并由硬件强制为 0。
GCEN	Bit 19	R/W	广播呼叫开启位 0: 关闭广播呼叫地址 0b00000000, 回应 NACK。 1: 开启广播呼叫地址 0b00000000, 回应 ACK。
—	Bit 18	—	—
NOSTRETCH	Bit 17	R/W	时钟延长关闭位

			<p>该位用于关闭从机模式下的时钟延长。在主机模式下必须保持为0。</p> <p>0: 开启时钟延长 1: 关闭时钟延长</p> <p>注: 只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
SBC	Bit 16	R/W	<p>从机字节控制使能位</p> <p>该位用于在从机模式下开启硬件字节控制。</p> <p>0: 关闭从机字节控制 1: 开启从机字节控制</p>
RXDMAEN	Bit 15	R/W	<p>DMA 接收器开启位</p> <p>0: 关闭 DMA 接收器 1: 开启 DMA 接收器</p>
TXDMAEN	Bit 14	R/W	<p>DMA 发送器开启位</p> <p>0: 关闭 DMA 发送器 1: 开启 DMA 发送器</p>
—	Bit 13-12	—	—
DNF	Bit 11-8	R/W	<p>数字噪声滤波器的滤波宽度</p> <p>该位用于配置 SDA 和 SCL 输入噪声数字滤波器的滤波宽度: DNF<3:0> x T_{I2CCLK}</p> <p>0000: 关闭数字滤波器 0001: 开启数字滤波器, 滤波宽度为 1 T_{I2CCLK} ...</p> <p>1111: 开启数字滤波器, 滤波宽度为 15 T_{I2CCLK}</p> <p>注: 只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
—	Bit 7-1	—	—
PE	Bit 0	R/W	<p>I2C 开启位</p> <p>0: I2C 关闭 1: I2C 开启</p> <p>注: 当 PE=0 时, I2C SCL 和 SDA 被释放, 内部状态机和状态位将恢复为其复位值。清零后, PE 为 0 并必须保持至少 3 个 APB 时钟周期。</p>

20.5.2.2 I2C控制寄存器 2 (I2C_CON2)

I2C 控制寄存器 2 (I2C_CON2)																																	
偏移地址: 004 _H																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved		ACK_UPD	HOLDACK	Reserved	PECBYTE	AUTOEND	RELOAD	NBYTES								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD											

—	Bit 31-30	—	—
ACK_UPD	Bit 29	T_W1	<p>ACK 更新</p> <p>该位由软件设置, 在 PE=0 时或在下一个时钟后由硬件清零。</p> <p>0: 不更新 ACK 状态</p> <p>1: 更新 ACK 状态并释放时钟</p>
HOLDACK	Bit 28	R/W	<p>等待回应 ACK</p> <p>开启时, 第 8 和第 9 个 SCL 脉冲之间会将 SCL 延长, 并等待用户设置 ACK/NACK。在设置 ACK/NACK 之后, 设置 ACK_UPD 以发出 ACK/NACK 并释放 SCL。</p> <p>0: 自动应答 ACK</p> <p>1: 开启手动回应 ACK</p>
—	Bit 27	—	—
PECBYTE	Bit 26	W1	<p>数据包错误检查字节</p> <p>该位由软件设置, 当传输 PEC 时, 或者当接收到 STOP 条件、匹配的地址或当 PE=0 时, 由硬件清零。</p> <p>0: 没有 PEC 传输</p> <p>1: 请求 PEC 发送器/接收器</p> <p>注: 向该位写 0 无效。设置 RELOAD 时, 该位无效。当 SBC=0 时, 该位对从机模式无效。如果不支持 SMBus 功能, 则该位保留并由硬件强制为 0。</p>
AUTOEND	Bit 25	R/W	<p>自动结束模式 (主机模式)</p> <p>该位由软件设置和清除。</p> <p>0: 软件结束模式: 发送器完 NBYTES 所设置的数据字节时设置 TC 标志, 将 SCL 拉低</p> <p>1: 自动结束模式: 发送器完 NBYTES 所设置的数据字节时自动发送 STOP 条件。</p> <p>注: 该位在从机模式或设置 RELOAD 位为 1 时无效。</p>
RELOAD	Bit 24	R/W	<p>NBYTES 重载模式</p> <p>该位由软件设置和清除。</p> <p>0: 在 NBYTES 所设置的数据字节传输完后结束传输 (接下来是 STOP 或 RESTART)。</p>

			1: 在 NBYTES 所设置的数据字节传输完后将继续传输 (NBYTES 将重载), TCR 标志置 1, 将 SCL 拉低。
NBYTES	Bit 23-16	R/W	接收/发送的数据字节数 此位与 I2C_CON1.NBYTES 组合成 16 位, NBYTES={ I2C_CON1.NBYTES[7:0],I2C_CON2.NBYTES[7:0]}, 用于设置要发送/接收的字节数。 从机模式下, 当 SBC=0 时, 该位无效。 注: 不允许在设置 START 位时更改该位。
NACK	Bit 15	W1	NACK 产生 (从机模式) 该位由软件设置, 在发送器 NACK 时由硬件清零, 或在接收到 STOP 条件或地址匹配时, 或当 PE=0 时由硬件清零。 0: 无操作 1: 在下一个接收字节发送 NACK 注: 向该位写 0 无效。 该位仅用于从机模式: 在主机接收模式下, 无论 NACK 位值如何, 在 STOP 或 RESTART 条件之前的最后一个字节自动产生 NACK。当从机接收 NOSTRETCH 模式发生溢出时, 无论 NACK 位值如何, 都会自动产生 NACK。当开启硬件 PEC 检查 (PECBYTE=1) 时, PEC 确认值不依赖于 NACK 值。
STOP	Bit 14	W1	STOP 产生 (主机模式) 该位由软件设置, 当检测到停止条件时, 或当 PE=0 时由硬件清零, 该位仅用于主机模式。 0: 无操作 1: 当前字节传输完后产生 STOP 注: 向该位写 0 无效。
START	Bit 13	W1	START 产生 该位由软件设置, 并在起始位后跟随地址序列发送器、仲裁丢失、检测超时错误或 PE=0 时由硬件清零。 0: 无操作 1: 产生 RESTART/START 如果 I2C 处于主机模式且 AUTOEND=0, 则在 NBYTES 所设置的数据字节传输结束后, 当 RELOAD=0 时, 将该位设置为 1 会产生重复起始条件 RESTART。 当总线空闲时, 将该位设置为 1 或产生 START 条件。 注: 向该位写 0 无效。 即使总线忙, 也可以设置 START 位。在 10 位地址模式下, 如果在地址的第一部分接收到 NACK, 则 START 位不会被硬件清零, 主机将重新发送地址序列, 除非 START 位被软件清零
HEAD10R	Bit 12	R/W	10 位地址序列发送方式选择 (主机接收模式) 0: 主机发送完整的 10 位从机地址序列: 起始位 + 地址标头/写 + 10 位地址的低 8 位 + 重新起始位 + 地址

			<p>标头/读</p> <p>1: 主机仅发送 10 位地址的前 7 位, 然后发送读取方向</p> <p>注: 不允许在设置 START 位时更改该位。</p>
ADD10	Bit 11	R/W	<p>地址模式选择 (主机模式)</p> <p>0: 主机使用 7 位地址模式</p> <p>1: 主机使用 10 位地址模式</p> <p>注: 不允许在设置 START 位时更改该位。</p>
RD_WRN	Bit 10	R/W	<p>传输方向 (主机模式)</p> <p>0: 主机发送</p> <p>1: 主机接收</p> <p>注: 不允许在设置 START 位时更改该位。</p>
SADD	Bit 9-0	R/W	<p>从机地址位 (主机模式)</p> <p>在 7 位地址模式下, bit<7:1>为待寻址的从机地址;</p> <p>在 10 位地址模式下, bit<9:0>为待寻址的从机地址。</p> <p>注: 不允许在设置 START 位时更改该位。</p>

20.5.2.3 I2C本机地址寄存器 1 (I2C_ADDR1)

I2C 本机地址寄存器 1 (I2C_ADDR1)																															
偏移地址: 008 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															OA1EN	Reserved					OA1MODE	OA1									

—	Bit 31-16	—	—
OA1EN	Bit 15	R/W	本机地址 1 开启位 0: 关闭本机地址 1。接收到从机地址 OA1 时响应 NACK。 1: 开启本机地址 1。接收到从机地址 OA1 时响应 ACK。
—	Bit 14-11	—	—
OA1MODE	Bit 10	R/W	本机地址 1 的地址模式选择位 0: 本机地址 1 使用 7 位地址模式 1: 本机地址 1 使用 10 位地址模式 注: 当 OA1EN=0 时才能写入该位。
OA1	Bit 9:0	R/W	本机地址 1 7 位地址模式下, bit <7:1>为本机地址 1; 10 位地址模式下, bit <9:0>为本机地址 1。 注: 当 OA1EN=0 时才能写入该位。

20.5.2.4 I2C本机地址寄存器 2 (I2C_ADDR2)

I2C 本机地址寄存器 2 (I2C_ADDR2)																															
偏移地址: 00C _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															OA2EN	Reserved			OA2MSK	OA2							Reserved				

—	Bit 31-16	—	—
OA2EN	Bit 15	R/W	本机地址 2 开启位 0: 关闭本机地址 2。接收到从机地址 OA2 时响应 NACK。 1: 开启本机地址 2。接收到从机地址 OA2 时响应 ACK。
—	Bit 14-11	—	—
OA2MSK	Bit 10-8	R/W	本机地址 2 屏蔽位 000: 没有屏蔽 001: OA2<1>被屏蔽, 仅比较 OA2<7:2>。 010: OA2<2:1>被屏蔽, 仅比较 OA2<7:3>。 011: OA2<3:1>被屏蔽, 仅比较 OA2<7:4>。 100: OA2<4:1>被屏蔽, 仅比较 OA2<7:5>。 101: OA2<5:1>被屏蔽, 仅比较 OA2<7:6>。 110: OA2<6:1>被屏蔽, 仅比较 OA2<7>。 111: OA2<7:1>被屏蔽, 不进行比较, 并且应答所有接收到的 7 位地址 (保留地址除外)。 注: 当 OA2EN=0 时才能写入该位。 一旦 OA2MSK 不等于 0, 则 I2C 即使比较匹配, 也不会应答保留地址 (0b0000xxx 和 0b1111xxx)。
OA2	Bit 7-1	R/W	本机地址 2 7 位地址模式: 7 位地址 注: 当 OA2EN=0 时才能写入该位。
—	Bit 0	—	—

20.5.2.5 I2C时钟寄存器 (I2C_TIMINGR)

I2C 时钟寄存器 (I2C_TIMINGR)																															
偏移地址: 010 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRESC				Reserved				SCLDEL				SDADEL				SCLH								SCLL							

PRESC	Bit 31-28	R/W	<p>时钟预分频器</p> <p>该位用于预分频 I2CCLK，以产生用于数据建立和保持的计数时钟以及 SCL 高电平和低电平的计数时钟，预分频后的时钟周期 T_{PRESC} 为：</p> $T_{PRESC} = (PRESC + 1) \times T_{I2CCLK}$ <p>只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
—	Bit 27-24	—	—
SCLDEL	Bit 23-20	R/W	<p>数据建立时间</p> <p>该位用于在 SDA 边沿和 SCL 上升沿之间产生延迟 T_{SCLDEL}。在主机和从机模式下，NOSTRETCH=0 时，SCL 线在 T_{SCLDEL} 期间拉低。</p> $T_{SCLDEL} = (SCLL - SCLDEL) \times T_{PRESC}$ <p>注：T_{SCLDEL} 用于产生 $T_{SU:DAT}$ 时序。</p> <p>只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
SDADEL	Bit 19-16	R/W	<p>数据保持时间</p> <p>该位用于在 SCL 下降沿和 SDA 边沿之间产生延迟 T_{SDADEL}。在主机和从机模式下，NOSTRETCH=0 时，SCL 线在 T_{SDADEL} 期间拉低。</p> $T_{SDADEL} = SDADEL \times T_{PRESC}$ <p>注：SDADEL 用于产生 $T_{HD:DAT}$ 时序。</p> <p>只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
SCLH	Bit 15-8	R/W	<p>SCL 高电平时间 (主机模式)</p> <p>该位用于在主机模式下产生 SCL 高电平脉宽。</p> $T_{SCLH} = (SCLH + 1) \times T_{PRESC}$ <p>注：SCLH 还用于产生 $T_{SU:STO}$ 和 $T_{HD:STA}$ 时序。</p> <p>只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>
SCLL	Bit 7-0	R/W	<p>SCL 低电平时间 (主机模式)</p> <p>该位用于在主机模式下产生 SCL 低电平脉宽。</p> $T_{SCLL} = (SCLL + 1) \times T_{PRESC}$ <p>注：SCLL 还用于产生 T_{BUF} 和 $T_{SU:STA}$ 时序。</p> <p>只有在关闭 I2C (PE=0) 时才能对该位进行设置。</p>

20.5.2.6 I2C 超时寄存器 (I2C_TIMEOUTR)

I2C 超时寄存器 (I2C_TIMEOUTR)																															
偏移地址: 014 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEXTEN		Reserved		TIMEOUTB												TIMEOUTEN		Reserved		TIDLE		TIMEOUTA									

TEXTEN	Bit 31	R/W	<p>累积时钟延长超时检测开启位</p> <p>0: 关闭累积时钟延长超时检测</p> <p>1: 开启累积时钟延长超时检测</p> <p>当 I2C 接口累积 SCL 延长超过 $T_{LOW:EXT}$ 时, 会检测到超时错误 (TOUT=1)。</p>
—	Bit 30-28	—	—
TIMEOUTB	Bit 27-16	R/W	<p>总线超时 B</p> <p>该位用于配置累积时钟延长超时:</p> <p>在主机模式下, 检测主机累积时钟低延长时间。</p> <p>在从机模式下, 检测从机累积时钟低延长时间。</p> <p>$T_{LOW:EXT} = (TIMEOUTB + 1) \times 2048 \times T_{I2CCCLK}$</p> <p>注: 当 TEXTEN=0 时才能写入该位。</p>
TIMEOUTEN	Bit 15	R/W	<p>时钟超时检测开启位</p> <p>0: 关闭 SCL 超时检测</p> <p>1: 开启 SCL 超时检测: 当 SCL 为低电平的时间超过 $T_{TIMEOUT}$ ($TIDLE=0$) 或高电平的时间超过 T_{IDLE} ($TIDLE=1$) 时, 检测到超时错误 (TOUT=1)。</p>
—	Bit 14-13	—	—
TIDLE	Bit 12	R/W	<p>空闲时钟超时检测选择位</p> <p>0: TIMEOUTA 用于检测 SCL 低超时</p> <p>1: TIMEOUTA 用于检测 SCL 和 SDA 高超时 (总线空闲状态)</p> <p>注: 当 TIMEOUTEN=0 时才能写入该位。</p>
TIMEOUTA	Bit 11-0	R/W	<p>总线超时 A</p> <p>该位用于配置:</p> <ul style="list-style-type: none"> - 当 TIDLE=0 时, SCL 低超时条件 $T_{TIMEOUT}$ <p>$T_{TIMEOUT} = (TIMEOUTA + 1) \times 2048 \times T_{I2CCCLK}$</p> <ul style="list-style-type: none"> - 当 TIDLE=1 时, 总线空闲状态 (SCL 和 SDA 均为高电平) <p>$T_{IDLE} = (TIMEOUTA + 1) \times 4 \times T_{I2CCCLK}$</p> <p>注: 当 TIMEOUTEN=0 时才能写入该位。</p>

20.5.2.7 I2C状态寄存器 (I2C_STAT)

I2C 状态寄存器 (I2C_STAT)																																	
偏移地址: 018 _H																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved								ADDRESS								DIR	BUSY	Reserved				TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE

—	Bit 31-24	—	—
ADDRESS	Bit 23-17	R	地址匹配代码 (从机模式) 当发生地址匹配事件 (ADDR = 1) 时, 该位为接收到的地址。在 10 位地址模式下, ADDRESS 为地址标头 (包括地址的 2 个 MSB 位)。
DIR	Bit 16	R	传输方向位 (从机模式) 发生地址匹配事件 (ADDR = 1) 时更新此标志。 0: 写入传输, 从机进入接收模式。 1: 读取传输, 从机进入发送模式。
BUSY	Bit 15	R	总线忙标志位 0: 总线空闲 1: 总线正在进行通信, 检测到 START 条件时由硬件设置为 1。当检测到停止条件或 PE=0 时, 由硬件清零
—	Bit 14-12	—	—
TCR	Bit 11	R	传输完成并重载标志位 当 RELOAD=1 且已将 NBYTES 所设置的数据字节传输完成时, 此标志由硬件设置为 1。当 NBYTES 位写入非零值时, 由硬件清零。 注: 当 PE=0 时, 该位由硬件清零。该标志仅用于主机模式, 或者用于从模式且设置 SBC 位为 1 时。
TC	Bit 10	R	传输完成标志位 (主机模式) 当 RELOAD=0、AUTOEND=0 且已将 NBYTES 所设置的数据字节传输完成时, 该标志由硬件设置为 1。当设置 START 位或 STOP 位为 1 时, 由硬件清零。 注: 当 PE=0 时, 该位由硬件清零。
—	Bit 9	—	—
RXUD	Bit 8	R	接收器下溢标志位 读取 I2C_STAT 后, 由硬件清零。 0: 接收器没有下溢 1: 接收器下溢

RXOV	Bit 7	R	接收器溢出标志位 读取 I2C_STAT 后, 由硬件清零。 0: 接收器没有溢出 1: 接收器溢出
—	Bit 6	—	—
RXNE	Bit 5	R	接收器非空标志位 0: 接收器空 1: 接收器非空
—	Bit 4	—	—
TXUD	Bit 3	R	发送器下溢标志位 读取 I2C_STAT 后, 由硬件清零。 0: 发送器没有下溢 1: 发送器下溢
TXOV	Bit 2	R	发送器溢出标志位 读取 I2C_STAT 后, 由硬件清零。 0: 发送器没有溢出 1: 发送器溢出
—	Bit 1	—	—
TXE	Bit 0	R	发送器空标志位 0: 发送器没有空 1: 发送器空

20.5.2.8 I2C PEC寄存器 (I2C_PECR)

I2C PEC 寄存器 (I2C_PECR)																															
偏移地址: 020 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								PEC							

—	Bit 31-8	—	—
PEC	Bit 7-0	R	数据包错误检查寄存器 当 PECEN = 1 时, 该位包含内部 PEC。 当 PECEN = 0 时, PEC 由硬件清零。

20.5.2.9 I2C接收器数据寄存器 (I2C_RXDATA)

I2C 接收器数据寄存器 (I2C_RXDATA)																															
偏移地址: 024 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RXDATA							

—	Bits 31-8	—	—
RXDATA	Bits 7-0	R	8 位接收器数据 从 I2C 总线接收器的数据字节。

20.5.2.10 I2C发送器数据寄存器 (I2C_TXDATA)

I2C 发送器数据寄存器 (I2C_TXDATA)																															
偏移地址: 028 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								TXDATA							

—	Bit 31-8	—	—
TXDATA	Bit 7-0	W	8 位发送器数据 要传输到 I2C 总线的数据字节。

20.5.2.11 I2C中断开启寄存器 (I2C_IER)

I2C 中断开启寄存器 (I2C_IER)																																									
偏移地址: 02C _H																																									
复位值: 0x0000 0000																																									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Reserved											ALERT	TOUT	PECE	ARLO	BERR	Reserved	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE										

—	Bit 31-21	—	—
ALERT	Bit 20	W1	<p>开启 SMBus 报警中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到 SMBus 报警事件时发生中断</p>
TOUT	Bit 19	W1	<p>开启超时中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到超时事件时发生中断</p>
PECE	Bit 18	W1	<p>开启 PEC 错误中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到 PEC 错误事件时发生中断</p>
ARLO	Bit 17	W1	<p>开启仲裁丢失中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到仲裁丢失事件时发生中断</p>
BERR	Bit 16	W1	<p>开启总线错误中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到总线错误事件时发生中断</p>
—	Bit 15	—	—
STOP	Bit 14	W1	<p>开启检测停止位中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到停止位事件时发生中断</p>
NACK	Bit 13	W1	<p>开启接收 NACK 中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到接收器 NACK 事件时发生中断</p>
ADDR	Bit 12	W1	<p>开启地址匹配中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到地址匹配事件时发生中断</p>
TCR	Bit 11	W1	<p>开启传输完成并重载中断功能</p> <p>此位置 1 时, 开启中断功能, 硬件侦测到传输完成</p>

			并重载事件时发生中断
TC	Bit 10	W1	开启传输完成中断功能 此位置 1 时，开启中断功能，硬件侦测到传输完成事件时发生中断
—	Bit 9	—	—
RXUD	Bit 8	W1	开启接收器下溢中断功能 此位置 1 时，开启中断功能，硬件侦测到接收器下溢事件时发生中断
RXOV	Bit 7	W1	开启接收器溢出中断功能 此位置 1 时，开启中断功能，硬件侦测到接收器溢出事件时发生中断
—	Bit 6	—	—
RXNE	Bit 5	W1	开启接收器非空中断功能 此位置 1 时，开启中断功能，硬件侦测到接收器非空事件时发生中断
—	Bit 4	—	—
TXUD	Bit 3	W1	开启发送器下溢中断功能 此位置 1 时，开启中断功能，硬件侦测到发送器下溢事件时发生中断
TXOV	Bit 2	W1	开启发送器溢出中断功能 此位置 1 时，开启中断功能，硬件侦测到发送器溢出事件时发生中断
—	Bit 1	—	—
TXE	Bit 0	W1	开启发送器空中断功能 此位置 1 时，开启中断功能，硬件侦测到发送器空事件时发生中断

20.5.2.12 I2C中断关闭寄存器 (I2C_IDR)

I2C 中断关闭寄存器 (I2C_IDR)																															
偏移地址: 030 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ALERT	TOUT	PECE	ARLO	BERR	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE	

—	Bit 31-21	—	—
ALERT	Bit 20	W1	关闭 SMBus 报警中断功能 此位置 1 时, 关闭 SMBus 报警中断功能
TOUT	Bit 19	W1	关闭超时中断功能 此位置 1 时, 关闭超时中断功能
PECE	Bit 18	W1	关闭 PEC 错误中断功能 此位置 1 时, 关闭 PEC 错误中断功能
ARLO	Bit 17	W1	关闭仲裁丢失中断功能 此位置 1 时, 关闭仲裁丢失中断功能
BERR	Bit 16	W1	关闭总线错误中断功能 此位置 1 时, 关闭总线错误中断功能
—	Bit 15	—	—
STOP	Bit 14	W1	关闭检测停止中断功能 此位置 1 时, 关闭检测停止位中断功能
NACK	Bit 13	W1	关闭接收器 NACK 中断功能 此位置 1 时, 关闭接收器 NACK 中断功能
ADDR	Bit 12	W1	关闭地址匹配中断功能 此位置 1 时, 关闭地址匹配中断功能
TCR	Bit 11	W1	关闭传输完成并重载中断功能 此位置 1 时, 关闭传输完成并重载中断功能
TC	Bit 10	W1	关闭传输完成中断功能 此位置 1 时, 关闭传输完成中断功能
—	Bit 9	—	—
RXUD	Bit 8	W1	关闭接收器下溢中断功能 此位置 1 时, 关闭接收器下溢中断功能
RXOV	Bit 7	W1	关闭接收器溢出中断功能 此位置 1 时, 关闭接收器溢出中断功能
—	Bit 6	—	—
RXNE	Bit 5	W1	关闭接收器非空中断功能 此位置 1 时, 关闭接收器非空中断功能
—	Bit 4	—	—

TXUD	Bit 3	W1	关闭发送器下溢中断功能 此位置 1 时，关闭发送器下溢中断功能
TXOV	Bit 2	W1	关闭发送器溢出中断功能 此位置 1 时，关闭发送器溢出中断功能
—	Bit 1	—	—
TXE	Bit 0	W1	关闭发送器空中断功能 此位置 1 时，关闭发送器空中断功能

20.5.2.13 I2C中断功能有效状态寄存器 (I2C_IVS)

I2C 中断功能有效状态寄存器 (I2C_IVS)																															
偏移地址: 034 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ALERT	TOUT	PECE	ARLO	BERR	Reserved	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE

—	Bit 31-21	—	—
ALERT	Bit 20	R	SMBus 报警中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
TOUT	Bit 19	R	超时中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
PECE	Bit 18	R	PEC 错误中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
ARLO	Bit 17	R	仲裁丢失中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
BERR	Bit 16	R	总线错误中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 15	—	—
STOP	Bit 14	R	检测停止位中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
NACK	Bit 13	R	接收器 NACK 中断功能状态 0: 中断功能处于关闭状态

			1: 中断功能处于开启状态
ADDR	Bit 12	R	地址匹配中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
TCR	Bit 11	R	传输完成并重载中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
TC	Bit 10	R	传输完成中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 9	—	—
RXUD	Bit 8	R	接收器下溢中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
RXOV	Bit 7	R	接收器溢出中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 6	—	—
RXNE	Bit 5	R	接收器非空中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 4	—	—
TXUD	Bit 3	R	发送器下溢中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
TXOV	Bit 2	R	发送器溢出中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 1	—	—
TXE	Bit 0	R	发送器空中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态

注：I2C_IVS 寄存器，是实时反映系统配置 I2C_IER 与 I2C_IDR 的中断开启状态。

20.5.2.14 I2C原始中断标志寄存器 (I2C_RIF)

I2C 原始中断标志寄存器 (I2C_RIF)																															
偏移地址: 038 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ALERT	TOUT	PECE	ARLO	BERR	Reserved	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE

—	Bits 31-21	—	—
ALERT	Bit 20	R	SMBus 报警, 原始中断状态 0: 无发生中断 1: 已发生中断
TOUT	Bit 19	R	超时, 原始中断状态 0: 无发生中断 1: 已发生中断
PECE	Bit 18	R	PEC 错误, 原始中断状态 0: 无发生中断 1: 已发生中断
ARLO	Bit 17	R	仲裁丢失, 原始中断状态 0: 无发生中断 1: 已发生中断
BERR	Bit 16	R	总线错误, 原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 15	—	—
STOP	Bit 14	R	检测停止位, 原始中断状态 0: 无发生中断 1: 已发生中断
NACK	Bit 13	R	接收器 NACK, 原始中断状态 0: 无发生中断 1: 已发生中断
ADDR	Bit 12	R	地址匹配, 原始中断状态 0: 无发生中断 1: 已发生中断
TCR	Bit 11	R	传输完成并重载, 原始中断状态 0: 无发生中断 1: 已发生中断
TC	Bit 10	R	传输完成, 原始中断状态 0: 无发生中断 1: 已发生中断

—	Bit 9	—	—
RXUD	Bit 8	R	接收器下溢，原始中断状态 0: 无发生中断 1: 已发生中断
RXOV	Bit 7	R	接收器溢出，原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 6	—	—
RXNE	Bit 5	R	接收器非空，原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 4	—	—
TXUD	Bit 3	R	发送器下溢，原始中断状态 0: 无发生中断 1: 已发生中断
TXOV	Bit 2	R	发送器溢出，原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 1	—	—
TXE	Bit 0	R	发送器空，原始中断状态 0: 无发生中断 1: 已发生中断

20.5.2.15 I2C中断标志屏蔽寄存器 (I2C_IFM)

I2C 中断标志屏蔽寄存器 (I2C_IFM)																															
偏移地址: 03C _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ALERT	TOUT	PECE	ARLO	BERR	Reserved	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE

—	Bits 31-21	—	—
ALERT	Bit 20	R	SMBus 报警, 标志位中断状态 0: 无发生中断 1: 已发生中断
TOUT	Bit 19	R	超时, 标志位中断状态 0: 无发生中断 1: 已发生中断
PECE	Bit 18	R	PEC 错误, 标志位中断状态 0: 无发生中断 1: 已发生中断
ARLO	Bit 17	R	仲裁丢失, 标志位中断状态 0: 无发生中断 1: 已发生中断
BERR	Bit 16	R	总线错误, 标志位中断状态 0: 无发生中断 1: 已发生中断
—	Bit 15	—	—
STOP	Bit 14	R	检测停止位, 标志位中断状态 0: 无发生中断 1: 已发生中断
NACK	Bit 13	R	接收器 NACK, 标志位中断状态 0: 无发生中断 1: 已发生中断
ADDR	Bit 12	R	地址匹配, 标志位中断状态 0: 无发生中断 1: 已发生中断
TCR	Bit 11	R	传输完成并重载, 标志位中断状态 0: 无发生中断

			1: 已发生中断
TC	Bit 10	R	传输完成, 标志位中断状态 0: 无发生中断 1: 已发生中断
—	Bit 9	—	—
RXUD	Bit 8	R	接收器下溢, 标志位中断状态 0: 无发生中断 1: 已发生中断
RXOV	Bit 7	R	接收器溢出, 标志位中断状态 0: 无发生中断 1: 已发生中断
—	Bit 6	—	—
RXNE	Bit 5	R	接收器非空, 标志位中断状态 0: 无发生中断 1: 已发生中断
—	Bit 4	—	—
TXUD	Bit 3	R	发送器下溢, 标志位中断状态 0: 无发生中断 1: 已发生中断
TXOV	Bit 2	R	发送器溢出, 标志位中断状态 0: 无发生中断 1: 已发生中断
—	Bit 1	—	—
TXE	Bit 0	R	发送器空, 标志位中断状态 0: 无发生中断 1: 已发生中断

注: I2C_IFM 寄存器, 是滤除已关闭中断功能的中断事件, 只关注开启中断功能的事件。此寄存器状态是将 I2C_RIF 与 I2C_IVS 进行硬件逻辑与运算。

20.5.2.16 I2C中断清除寄存器 (I2C_ICR)

I2C 中断清除寄存器 (I2C_ICR)																															
偏移地址: 040 _H																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											ALERT	TOUT	PECE	ARLO	BERR	Reserved	STOP	NACK	ADDR	TCR	TC	Reserved	RXUD	RXOV	Reserved	RXNE	Reserved	TXUD	TXOV	Reserved	TXE

—	Bits 31-21	—	—
ALERT	Bit 20	C_W1	SMBus 报警中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
TOUT	Bit 19	C_W1	超时中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
PECE	Bit 18	C_W1	PEC 错误中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
ARLO	Bit 17	C_W1	仲裁丢失中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
BERR	Bit 16	C_W1	总线错误中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
—	Bit 15	—	—
STOP	Bit 14	C_W1	检测停止位中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
NACK	Bit 13	C_W1	接收器 NACK 中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
ADDR	Bit 12	C_W1	地址匹配中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
TCR	Bit 11	C_W1	传输完成并重载中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
TC	Bit 10	C_W1	传输完成中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
—	Bit 9	—	—
RXUD	Bit 8	C_W1	接收器下溢中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
RXOV	Bit 7	C_W1	接收器溢出中断清除 此位置 1 时, 清除中断状态 (I2C_RIF 与 I2C_IFM)
—	Bit 6	—	—

RXNE	Bit 5	C_W1	接收器非空中断清除 此位置 1 时，清除中断状态 (I2C_RIF 与 I2C_IFM) 注：当 I2C_STAT 寄存器的 RXNE 位为 1 时，对该位写 1 无效。
—	Bit 4	—	—
TXUD	Bit 3	C_W1	发送器下溢中断清除 此位置 1 时，清除中断状态 (I2C_RIF 与 I2C_IFM)
TXOV	Bit 2	C_W1	发送器溢出中断清除 此位置 1 时，清除中断状态 (I2C_RIF 与 I2C_IFM)
—	Bit 1	—	—
TXE	Bit 0	C_W1	发送器空中断清除 此位置 1 时，清除中断状态 (I2C_RIF 与 I2C_IFM) 注：当 I2C_STAT 寄存器的 TXE 位为 1 时，对该位写 1 无效。

注：I2C_ICR 寄存器设置时，将清除 I2C_RIF 与 I2C_IFM 中断标志状态；此设置不影响中断 I2C_IER、I2C_IDR 与 I2C_IVS 寄存器，只清除标志状态 I2C_RIF 与 I2C_IFM。此寄存器通过硬件清除中断。

第21章 串行外设接口（SPI0~1）/音频总线（I2S0）

21.1 概述

SPI 接口提供两个主要功能，支持 SPI 协议或 I2S 音频协议。默认情况下，选择的是 SPI 功能。可通过软件将接口从 SPI 切换到 I2S。

串行外设接口（SPI）可与外部 SPI 设备进行半双工或全双工的同步串行通信。该接口可配置为主机模式或从机模式。在配置为主机模式时，可以为外部 SPI 从设备提供通信时钟（SCK）。该接口还能够多主机模式下工作。

I2S 协议也是同步串行通信接口。它可在全双工模式（使用 4 个引脚）或半双工模式（使用 3 个引脚）下作为从机或主机工作。当 I2S 配置为通信主机模式时，该接口可以向外部 I2S 从设备提供主时钟（MCLK）。它可以满足四种不同的音频标准，包括 I2S Philips 标准、MSB 和 LSB 对齐标准以及 PCM 标准。

芯片内部共有 2 个串行外设接口（SPI0~1），其中 SPI0 还支持 I2S 协议（I2S0）。

21.2 特性

21.2.1 SPI的主要特点

- ◆ 支持主机或从机模式操作
- ◆ 基于三线的全双工同步传输
- ◆ 基于双线的半双工同步传输，其中一条线作为双向数据线
- ◆ 基于双线的单工同步传输，其中一条线作为单向数据线
- ◆ 8 位或 16 位传输帧格式选择
- ◆ 支持多主机模式功能
- ◆ 主机模式支持波特率可配置
- ◆ 从机模式波特率最高可达 $f_{PCLK}/2$
- ◆ 对于主机模式和从机模式都可通过硬件或软件进行 NSS 控制
- ◆ 时钟极性和相位可编程
- ◆ 数据顺序可编程，可最先移位 MSB 或 LSB
- ◆ 具有发送或接收的 FIFO 缓存状态标志
- ◆ 具有显示 SPI 总线忙状态标志
- ◆ 支持 SPI 摩托罗拉协议
- ◆ 支持 SPI TI 协议
- ◆ 支持用于确保可靠通信的硬件 CRC 功能：

- 在发送模式下可将 CRC 值作为最后一个字节数据发送
- 根据收到的最后一个字节自动进行 CRC 错误校验
- ◆ 提供独立 16 级深度的发送和接收 FIFO 数据缓存
- ◆ 提供 12 种中断事件可触发中断
- ◆ 支持 DMA 传输：发送和接收请求

21.2.2 I2S的主要特点

- ◆ 全双工通信
- ◆ 半双工通信（仅作为发送器或接收器）
- ◆ 支持主机或从机模式操作
- ◆ 8 位可编程线性预分频器，可达到精确的音频采样频率（从 8kHz 到 96kHz）
- ◆ 数据格式可以是 16 位、24 位或 32 位
- ◆ 数据长度由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32 位数据帧）
- ◆ 可编程时钟极性
- ◆ 提供 9 种中断事件可触发中断
- ◆ 支持的 I2S 协议：
 - I2S Philips 标准
 - MSB 对齐标准（左对齐）
 - LSB 对齐标准（右对齐）
 - PCM 标准
- ◆ 数据方向始终为 MSB 位在前
- ◆ 支持 DMA 传输（16 位宽）：发送和接收请求
- ◆ 可输出主时钟以驱动外部音频组件。比特率固定为 $256 \times F_s$ （其中 F_s 为音频采样频率）
- ◆ 提供独立 16 级深度的发送和接收 FIFO 数据缓存

21.3 SPI实现

本手册介绍了 SPI0 和 SPI1 中实现的全部功能。

SPI 模块/特性	SPI0	SPI1
Rx 和 Tx 的 FIFO 缓存大小	16 Bytes	4 Bytes
I2S 功能	有	无

表 21-1 SPI 特性

21.4 SPI结构图

SPI 允许 MCU 与外部设备之间进行同步串行通信。应用软件可以通过轮询状态标志或使用专用 SPI 中断来管理通信。SPI 的主要模块及其相互关系如下面的框图所示。

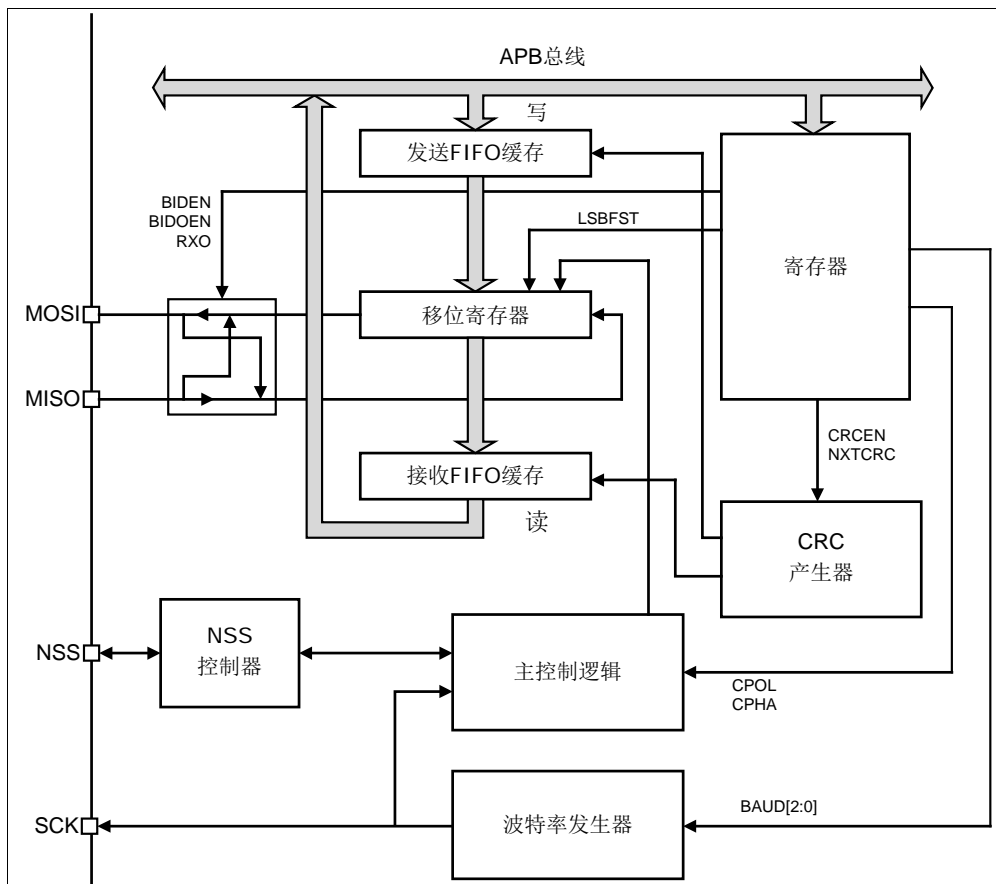


图 21-1 SPI 电路结构框图

通常，SPI 使用四个 I/O 引脚来与外部器件连接：

- ◆ **MISO**：主机输入/从机输出数据引脚。此引脚可用于在从机模式下发送数据和在主机模式下接收数据。
- ◆ **MOSI**：主机输出/从机输入数据引脚。此引脚可用于在主机模式下发送数据和在从机模式下接收数据。
- ◆ **SCK**：用于 SPI 主机的串行时钟输出以及 SPI 从机的串行时钟输入。
- ◆ **NSS**：从机选择引脚。此引脚用作“片选”，可让 SPI 主机与从机进行单独通信，从而避免数据线上的竞争。从机的 NSS 输入可由主机的 IO 端口驱动。

在主机模式下（SPI_CON1.MSTREN=1），当 SPI_CON2 寄存器中 NSSOE 位置 1 时，NSS 引脚配置为输出，传输时硬件驱动 NSS 引脚为低电平，当 SPI_CON2 寄存器中 NSSOE 位置 0 时，NSS 引脚配置为输入，如果 NSS 被拉至低电平将产生冲突，SPI_CON1.MSTREN 位将自动清零，SPI 将进入模式错误状态（更多信息请参见“模式故障 (MODF)”章节）。

21.5 SPI功能描述

在 SPI 通信期间，接收和发送操作同时执行。串行时钟（SCK）同步数据线上信息的移位和采样。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够一起通信，主机和从机必须遵循相同的通信格式。

21.5.1 时钟相位和极性控制

通过配置 SPI_CON1.CPOL 和 SPI_CON1.CPHA 位，可以选择四种可能的时序关系。SPI_CON1.CPOL（时钟极性）位控制空闲时时钟线的电平状态，此位对主机和从机都有作用。如果复位 SPI_CON1.CPOL 位，SCK 引脚在空闲状态时处于低电平。如果将 SPI_CON1.CPOL 位置 1，SCK 引脚在空闲状态时处于高电平。

如果将 SPI_CON1.CPHA 位置 1，则 SCK 引脚上的第二个边沿（如果 SPI_CON1.CPOL 位配置为 0，则为下降沿；如果 SPI_CON1.CPOL 位配置为 1，则为上升沿）对 MSB 采样。即在第二个时钟边沿锁存数据。如果复位 CPHA 位，则 SCK 引脚上的第一个边沿（如果 SPI_CON1.CPOL 位配置为 0，则为上升沿；如果 SPI_CON1.CPOL 位配置为 1，则为下降沿）对 MSB 采样。即在第一个时钟边沿锁存数据。

可通过组合 SPI_CON1.CPOL 和 SPI_CON1.CPHA 位来选择数据捕获的时钟边沿。

下图显示了在 SPI_CON1.CPHA 和 SPI_CON1.CPOL 位的四种组合下的 SPI 传输。可以将该图解释为主机或从机时序图，其中 SCK 引脚、MISO 引脚、MOSI 引脚直接连接在主机和从机之间。

注 1：在切换 SPI_CON1.CPOL 或 SPI_CON1.CPHA 位之前，必须通过复位 SPI_CON1.SPIEN 位来关闭 SPI。必须以同一时序模式对主机和从机进行编程。

注 2：主机和从机的时序需要配置成相同，通信才能正常。

注 3：SCK 的空闲状态必须与 SPI_CON1 寄存器中选择的极性相对应（如果 SPI_CON1.CPOL=1，则上拉 SCK；如果 SPI_CON1.CPOL=0，则下拉 SCK）。

注 4：通过 SPI_CON1.FLEN 位选择数据帧长度（8 或 16 位），该格式决定了发送与接收过程中的数据长度。

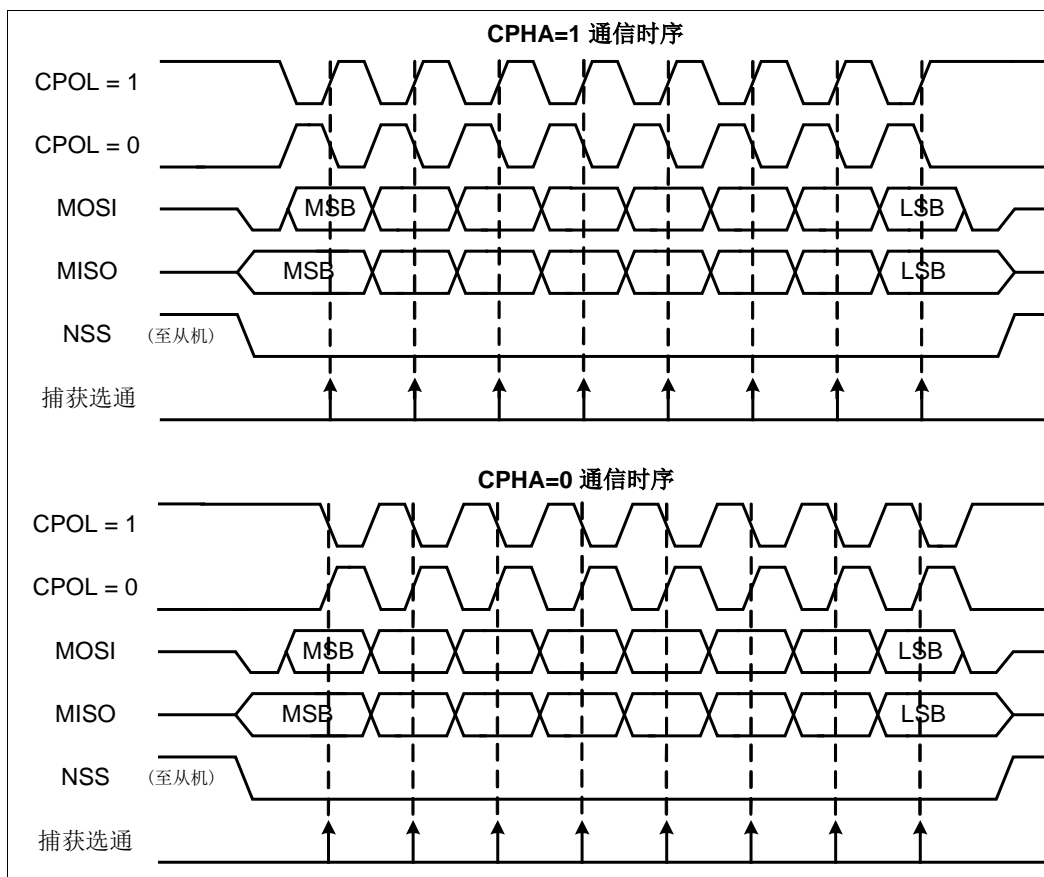


图 21-2 SPI 通信时序格式

注：上图中显示的是当 SPI_CON1.LSBFST 处于复位状态时的时序图。

21.5.2 数据帧格式

SPI 移位寄存器可以设置为移出 MSB 优先或 LSB 优先，具体取决于 SPI_CON1.LSBFST 位的值。

每个数据帧的长度均为 8 位或 16 位，具体取决于 SPI_CON1.FLEN 位的配置。所选的数据帧长度适用于发送和接收。

21.5.3 从机片选 (NSS) 引脚管理

可以使用 SPI_CON1.SSEN 位设置硬件或软件控制从机片选。

- ◆ 软件控制 NSS (SPI_CON1.SSEN = 1)，从机片选是由内部 SPI_CON1.SSOUT 位的值决定。
- ◆ 硬件管理 NSS (SPI_CON1.SSEN = 0)，根据 NSS 输出配置 (SPI_CON2.NSSOE 位)，硬件管理 NSS 有两种模式。
 - ◇ NSS 输出使能 (SPI_CON1.SSEN = 0, SPI_CON2.NSSOE = 1) 仅当 SPI 设备在主机模式下工作时才使用此配置。当主机开始传输数据时，NSS 信号驱动为低电平，并保持到数据传输结束为止。
 - ◇ NSS 输出禁止 (SPI_CON1.SSEN = 0, SPI_CON2.NSSOE = 0) 对于在主机模式

下工作的设备，此配置允许许多主机模式功能。对于设置为从机模式的设备，NSS 引脚用作传统 NSS 输入：在 NSS 为低电平时片选该从机，在 NSS 为高电平时取消对它的片选。

21.5.4 主机与从机的单对单通讯应用

SPI 允许 MCU 使用不同的配置进行通信，具体取决于所针对的设备和应用要求。当使用软件 NSS 管理时通过 2 或 3 线进行通信，使用硬件 NSS 管理时通过 3 或 4 线进行通信。通信始终由主机发起。

21.5.4.1 全双工通信

默认情况下，SPI 配置为全双工通信。在此配置中，MOSI 引脚连接在一起，MISO 引脚连接在一起。通过这种方式，主机和从机之间以串行方式传输数据（最高有效位在前）。

通信始终由主机发起。当主机通过 MOSI 引脚向从机发送数据时，从机同时通过 MISO 引脚发出准备好的数据。这是一个数据输出和数据输入都由同一时钟进行同步的全双工通信过程，时钟信号由主机的 SCK 引脚发出提供给从机。

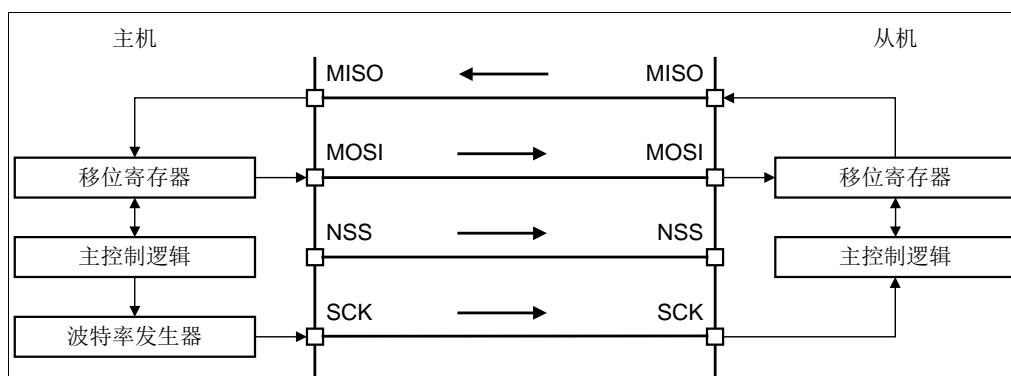


图 21-3 全双工通信

21.5.4.2 半双工通信

SPI 可将 SPI_CON1.BIDEN 位置 1 来使能此模式。在此模式下, SCK 作为时钟信号输出引脚, MOSI (主机模式下) 或 MISO (从机模式下) 作为数据通信引脚。通过 SPI_CON1.BIDOEN 位来选择传输方向 (输入或输出)。当该位置 1 时数据线为输出, 该位置 0 时为输入。

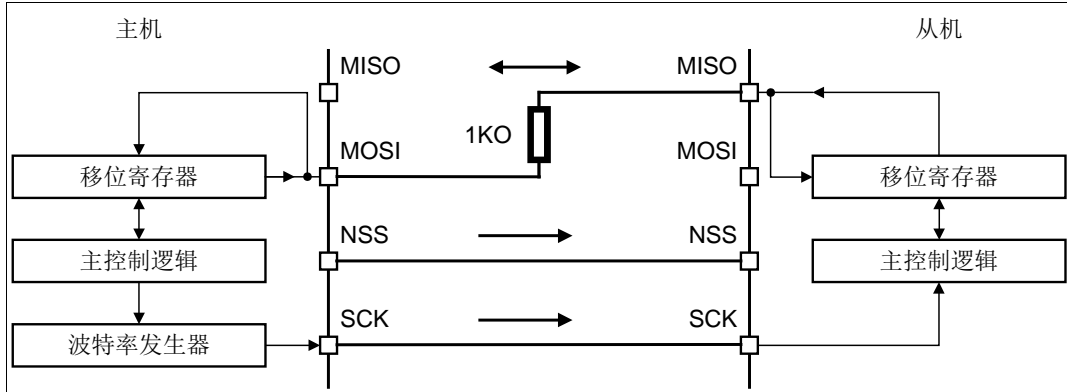


图 21-4 半双工通信

注 1: NSS 引脚可用于在主机和从机之间提供硬件控制流。如果不使用 NSS 引脚的话, 必须在主机和从机的处理程序中增加控制流。有关更多详细信息, 请参见“从机片选 (NSS) 引脚管理”章节。

注 2: 在此配置中, 主机的 MISO 引脚和从机的 MOSI 引脚可用作 GPIO。

注 3: 当在双向模式下工作的两个节点之间的通信方向更改不同步, 或同时在公共线上临时提供相反的输出电平进行斗争时, 建议在 MISO 和 MOSI 引脚之间插入一个串联电阻, 以保护输出并在这种情况下限制它们之间的电流。

21.5.4.3 单工通信

SPI 可以在单工模式下通信，方法是使用 SPI_CON1.RXO 位将 SPI 设置为只发送或只接收。在这种模式下，只有一条线路用于主机和从机之间的数据传输。

- ◆ 只发送模式类似于全双工模式（SPI_CON1.BIDEN=0、SPI_CON1.RXO=0）：在发送引脚（主机模式下的 MOSI 或从机模式下的 MISO）上发送数据。接收引脚（主机模式下的 MISO 或从机模式下的 MOSI）可用作通用 IO。在此模式下接收的数据是无意义的，应用程序只需要忽略接收 FIFO 缓存。
- ◆ 只接收模式下，应用程序可将 SPI_CON1.RXO 位置 1 来关闭 SPI 输出功能。在这种情况下，发送 IO 引脚（主机模式下的 MOSI 或从机模式下的 MISO）可用于其它用途。

当 SPI 设置为只接收模式时：

- ◆ 一旦在主机模式下使能 SPI 后，主机会立即从 SCK 引脚发送时钟，即意味着通信开启，当 SPI_CON1.SPIEN 位清 0 后，SPI 模块关闭，通信也立即停止。此模式下无需读取 BUSY 标志，因为开始通信后此标志一直为 1。
- ◆ 在从机模式下，只要 NSS 引脚被拉低（或在 NSS 软件模式下将 SPI_CON1.SSOUT 位清零），意味着从机被选中，同时一直有来自主机的 SCK 输入，SPI 就会继续接收。

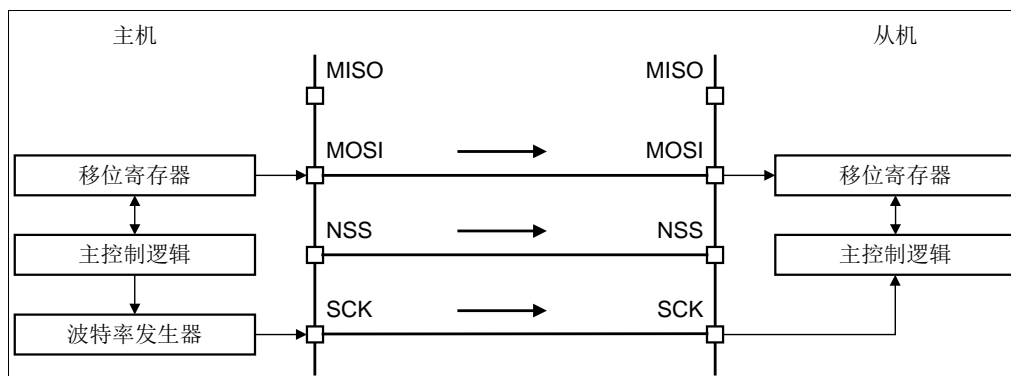


图 21-5 单工通信（主机模式下的只发送与从机模式下的只接收）

注：SPI 单工模式字节接收，SCK 时钟仅支持最快为 PCLK 的 4 分频；SPI 单工模式半字接收或单工发送支持最快为 PCLK 的 2 分频。

21.5.5 标准多从机通讯应用

在具有两个或更多独立从机的配置中，主机使用 GPIO 引脚来管理每个从机的芯片选择线，请见下图。主机必须通过拉低连接到从机 NSS 输入的 GPIO 来单独选择一个从机，实现主机和被选择从机的专用通信。

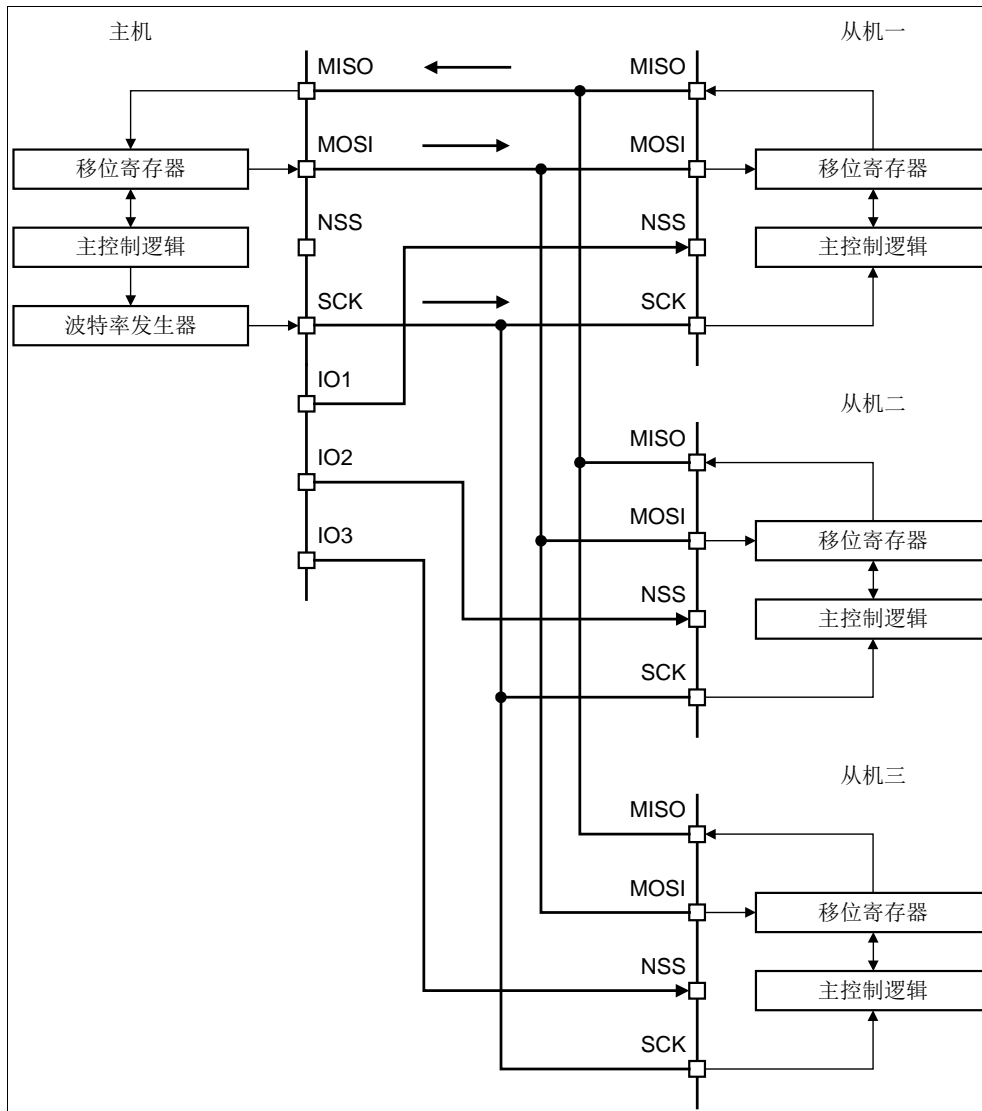


图 21-6 多从机通讯（一个主机和三个从机）

21.5.6 多主机通讯应用

多主机模式仅支持两个 SPI 节点，因为一次只有一个节点可以将其输出应用于公共数据线上。可以使用内置功能来检测主控总线的两个节点之间的潜在冲突。对于此检测必须将 NSS 引脚配置为硬件输入模式。

当节点处于非活动状态时，默认情况下两个节点都处于从机模式。一旦一个节点想要超越总线上的控制，它就会切换到主机模式并通过专用的 GPIO 引脚对另一个节点的从机选择输入应用活动电平。会话完成后释放活动的从机选择信号，节点控制总线临时返回被动从机模式，等待下一个会话启动。

如果可能两个节点同时提出了它们的主控请求，则会出现总线冲突事件（参见“模式故障 (MODF)”章节）。使用者可以应用一些简单的仲裁过程（例如通过在两个节点上定义不同的超时机制来推迟下一次请求）。

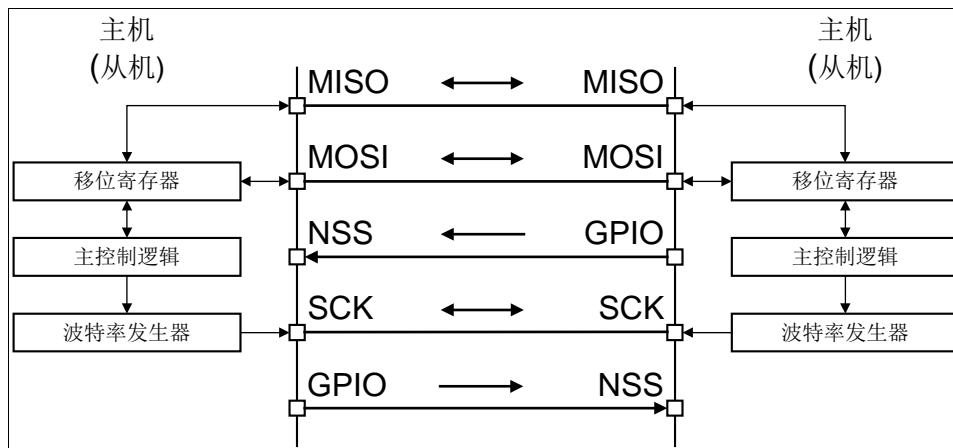


图 21-7 多主机通讯应用

21.5.7 SPI从机模式

SPI 作为从机时，时钟信号由主机提供，所以建议先使能从机，然后主机再发送时钟，否则数据传输可能不正常。建议按以下步骤配置 SPI 为从机：

步骤

1. 先配置时钟，将 SPI_CON1.CPOL 和 SPI_CON1.CPHA 位配置好，以定义数据传输和时钟之间的关系，注意从机和主机的这两位配置需保持一致。
2. 通过 SPI_CON1.FLEN 位设置数据帧的长度，可选择 8 位或 16 位。
3. 帧格式（MSB 在前或 LSB 在前取决于 SPI_CON1.LSBFST 位的值）必须与主机的帧格式相同。
4. 在硬件模式下（请参见“从机片选（NSS）引脚管理”章节），NSS 引脚在整个字节发送序列期间都会保持低电平。在 NSS 软件模式下将 SPI_CON1.SSEN 位置 1，将 SPI_CON1.SSOUT 位清零。
5. 将 SPI_CON1.MSTREN 位清零，并将 SPI_CON1.SPIEN 位置 1。
6. 在从机模式下，MOSI 引脚作为数据输入，MISO 引脚作为数据输出。

发送序列

数据字节在写周期内被并行加载到发送 FIFO 缓存中。

当从机收到时钟信号并在 MOSI 引脚上收到数据的最高有效位时，发送序列开始。其余位（8 位数据帧长度中的 7 个位，16 位数据帧长度中的 15 个位）将加载到移位寄存器中。

SPI_STAT.TXE 标志在发送 FIFO 缓存的最后一个数据加载到移位寄存器时置 1，并且在 SPI_IER.TXE 位置 1 时将生成中断请求。

接收序列

对于接收器在数据传输完成时，移位寄存器中的数据将传输到接收 FIFO 缓存，并且将 SPI_STAT.RXNE 位置 1。

在最后一个采样时钟边沿后，将 SPI_STAT.RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收 FIFO 缓存中，并且在 SPI_IER.RXNE 位置 1 时将生成中断请求。当读取 SPI_DATA 寄存器时，SPI 外设将返回此缓冲值。

21.5.8 SPI主机模式

时钟信号由主机从 SCK 引脚发出传输给从机。

步骤

1. 先配置时钟，设置 SPI_CON1.BAUD 位域，定义时钟波特率。
2. 配置 SPI_CON1.CPOL 和 SPI_CON1.CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）。
3. 通过 SPI_CON1.FLEN 位设置数据帧的长度，可选择 8 位或 16 位。
4. 通过 SPI_CON1.LSBFST 位设置定义帧格式，可选择 MSB 在前或 LSB 在前。
5. 当 NSS 引脚配置成输入时，在 NSS 硬件模式下，NSS 引脚在整个发送序列期间必须连接到高电平信号；在 NSS 软件模式下，需要将 SPI_CON1.SSEN 和 SPI_CON1.SSOUT 位都置 1。如果 NSS 引脚配置成输出，只需要将 SPI_CON2.NSSOE 位置 1 即可。
6. SPI_CON1.MSTREN 位置 1 使 SPI 工作在主机模式下，然后 SPI_CON1.SPIEN 位置 1 使能 SPI 模块（当 NSS 引脚配置成输入且在 NSS 硬件模式时，NSS 引脚必须维持高电平信号，这两个位才保持置 1）。
7. 在此配置中，MOSI 引脚作为数据输出，MISO 引脚作为数据输入。

发送序列

在 SPI_DATA 寄存器写入数据字节时，发送序列开始。在第一个位传输期间，数据并行加载到移位寄存器中，然后以串行方式移出到 MOSI 引脚，至于是 MSB 在前还是 LSB 在前则取决于 SPI_CON1.LSBFST 位。TXE 标志在发送 FIFO 缓存的最后一个数据加载到移位寄存器时置 1，并且在 SPI_IER.TXE 位置 1 时将生成中断请求。

接收序列

对于接收器，在数据传输最后一个采样时钟边沿时，将 SPI_STAT.RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收 FIFO 缓存中，并且在 SPI_IER.RXNE 位置 1 时将生成中断请求。当读取 SPI_DATA 寄存器时，可读取到接收 FIFO 缓存中的数据。

如果在发送开始后将要发送的下一个数据置于发送 FIFO 缓存，则可保持连续的数据发送流。请注意，仅当 SPI_STAT.TXF 位为 0 时，才可以对发送 FIFO 缓存执行写操作。

注：如果与之通信的从机需要在每个字节传输之间拉低片选信号，必须将该主机的 NSS 配置成 GPIO，或使用其他 GPIO，通过软件控制从机的片选。

21.5.9 数据发送和接收

21.5.9.1 接收和发送FIFO缓存

所有 SPI 数据传输都通过嵌入式 16 级深度的 FIFO 缓存。使 SPI 能够连续传输工作。发送和接收都有自己的 FIFO 缓存。

对 SPI_DATA 寄存器的读访问将返回存储在接收 FIFO 缓存中但尚未读取的最旧的值。对 SPI_DATA 的写访问将已写数据存储在发送队列末尾的发送 FIFO 缓存中。SPI_STAT 寄存器中 RXFLV 和 TXFLV 位域指示两个 FIFO 缓存的有效数据个数。

对 SPI_DATA 寄存器的读访问必须由 RXTH 事件管理。当数据存储在接收 FIFO 缓存中并且达到阈值（由 SPI_CON2 寄存器中 RXFTH 位域定义）时，触发此事件。当 RXTH 被清除时，表示接收 FIFO 缓存中的有效数据个数小于阈值。以类似的方式，要发送的数据帧的写访问由 TXTH 事件管理。当发送 FIFO 缓存有效数据个数小于或等于阈值（由 SPI_CON2 寄存器中 TXFTH 位域定义）时将触发此事件。

21.5.9.2 在主机模式下启动通信序列

- ◆ 在全双工通信（SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0）
 - ◇ 将数据写入到 SPI_DATA 寄存器（发送 FIFO 缓存）后，启动通信序列。
 - ◇ 随后在第一个位的发送期间，将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - ◇ 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DATA 寄存器（接收 FIFO 缓存）中。
- ◆ 在单工通信-只接收模式（SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1）
 - ◇ 只要 SPI_CON1.SPIEN = 1，通信序列就立即开始。
 - ◇ MISO 引脚上接收的数据会先以串行方式移入 8 位移位寄存器，接着再从移位寄存器并行加载到 SPI_DATA 寄存器（接收 FIFO 缓存）中。
- ◆ 在半双工通信-发送模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1）
 - ◇ 将数据写入到 SPI_DATA 寄存器（发送 FIFO 缓存）时，通信序列启动。
 - ◇ 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
 - ◇ 不接收任何数据。
- ◆ 在半双工通信-接收模式（SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=0）
 - ◇ 只要 SPI_CON1.SPIEN=1 且 SPI_CON1.BIDOEN=0，通信序列就立即开始。
 - ◇ 在 MOSI 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DATA 寄存器（接收 FIFO 缓存）中。
 - ◇ 不会有数据以串行方式移出到 MOSI 引脚。

21.5.9.3 在从机模式下启动通信序列

- ◆ 在全双工模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0)
 - ◇ 当从机收到时钟信号并在 MOSI 引脚上收到数据的第一个位时, 通信序列开始。其余 7 个位将加载到移位寄存器中。
 - ◇ 同时, 在第一个位的发送期间, 将数据从发送缓冲区并行加载到 8 位移位寄存器中, 然后以串行方式将其移出到 MISO 引脚。在 SPI 主机启动传输前, 软件必须已把要从机发送的数据写入发送 FIFO 缓存。
- ◆ 在单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1)
 - ◇ 当从机收到时钟信号并在 MOSI 引脚上收到数据的第一个位时, 通信序列开始。其余 7 个位将加载到移位寄存器中。
 - ◇ 不会有数据以串行方式移出到 MISO 引脚。
- ◆ 在半双工通信-发送模式 (SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1)
 - ◇ 当从机收到时钟信号, 并且 MISO 引脚上发出发送 FIFO 缓存中的第一位数据时, 通信序列开始。
 - ◇ 随后在第一个位的发送期间, 将数据从发送 FIFO 缓存并行加载到 8 位移位寄存器中, 然后以串行方式将其移出到 MISO 引脚。在 SPI 主机启动传输前, 软件必须已把要从机发送的数据写入发送 FIFO 缓存。
 - ◇ 不接收任何数据。
- ◆ 在半双工通信-接收模式 (SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=0)
 - ◇ 当从机收到时钟信号并在 MOSI 引脚上收到数据的第一个位时, 通信序列开始。
 - ◇ 在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器, 然后并行加载到 SPI_DATA 寄存器 (接收 FIFO 缓存) 中。
 - ◇ 不会有数据以串行方式移出到 MISO 引脚。

21.5.9.4 处理数据发送与接收

全双工通信 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=0), 发送和接收数据的处理过程。

直接存取操作模式:

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI, 将第一个要发送的数据项写入 SPI_DATA 寄存器 (此操作会将 SPI_STAT.TXE 位清零)。
2. 等待 SPI_STAT.TXE=1, 然后写入要发送的第二个数据项。然后等待 SPI_STAT.RXNE=1, 读取 SPI_DATA 以获取第一个接收到的数据 (此操作会将 SPI_STAT.RXNE 位清零)。对每个要发送和接收的数据项重复此操作, 直到发送并接收完最后的数据。
3. 检查 SPI_STAT.TXE=1, 然后等待至 SPI_STAT.BUSY=0, 再关闭 SPI。
4. 此外, 还可以使用 TXE 或 RXNE 中断事件对应的各个中断子程序来实现该过程。

FIFO 缓存操作模式:

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI。

2. 配置 SPI_CON2.TXFTH 与 SPI_CON2.RXFTH。
3. 当 SPI_STAT.TXTH=1, 将要发送的数据写入 SPI_DATA 寄存器 (写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值), 当 SPI_STAT.RXTH=1, 读取 SPI_DATA 寄存器以获取接收到的数据 (读取的数据个数必须为 SPI_CON2.RXFTH 设定的阈值), 重复此操作直到写入最后要发送的数据。
4. 等待至 SPI_STAT.BUSY=0, 读取 SPI_DATA 寄存器以获取接收到的数据直到 SPI_STAT.RXFLV 位域为 0, 再关闭 SPI。
5. 此外, 还可以使用 TXTH 或 RXTH 中断事件对应的各个中断子程序来实现该过程。

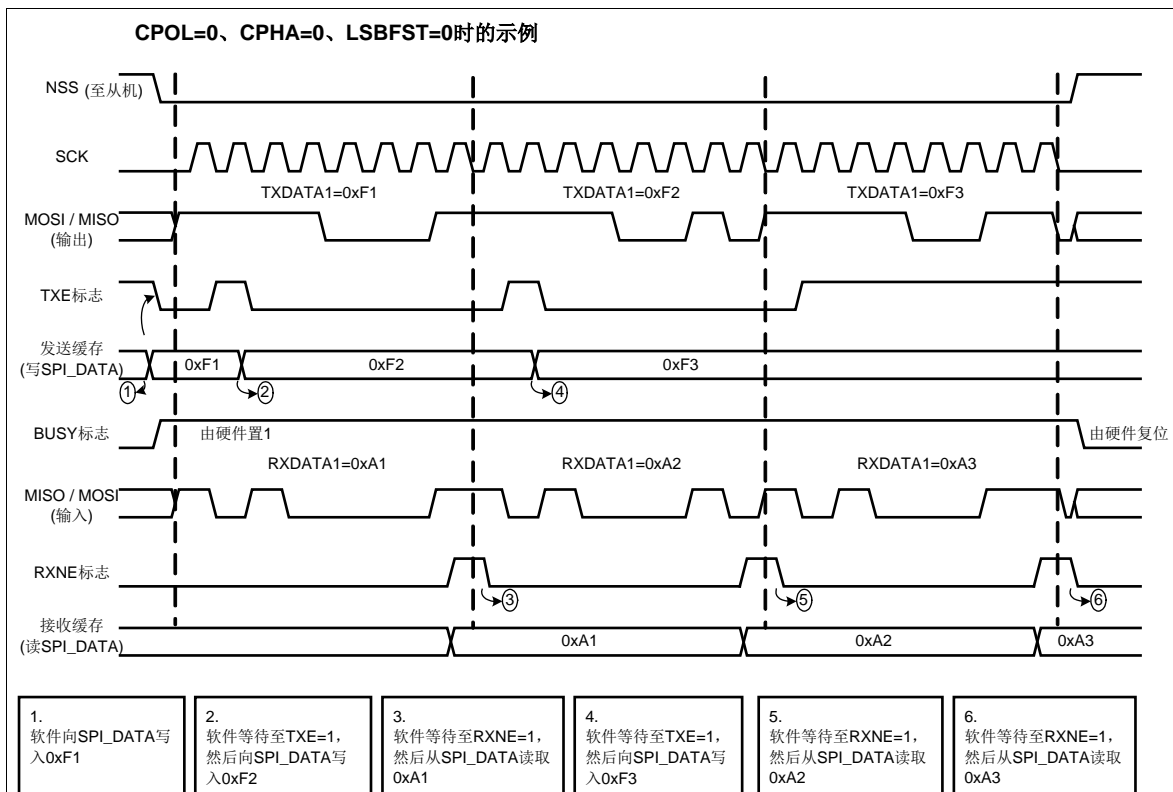


图 21-8 全双工通信时序 (直接存取操作模式, 连续传输)

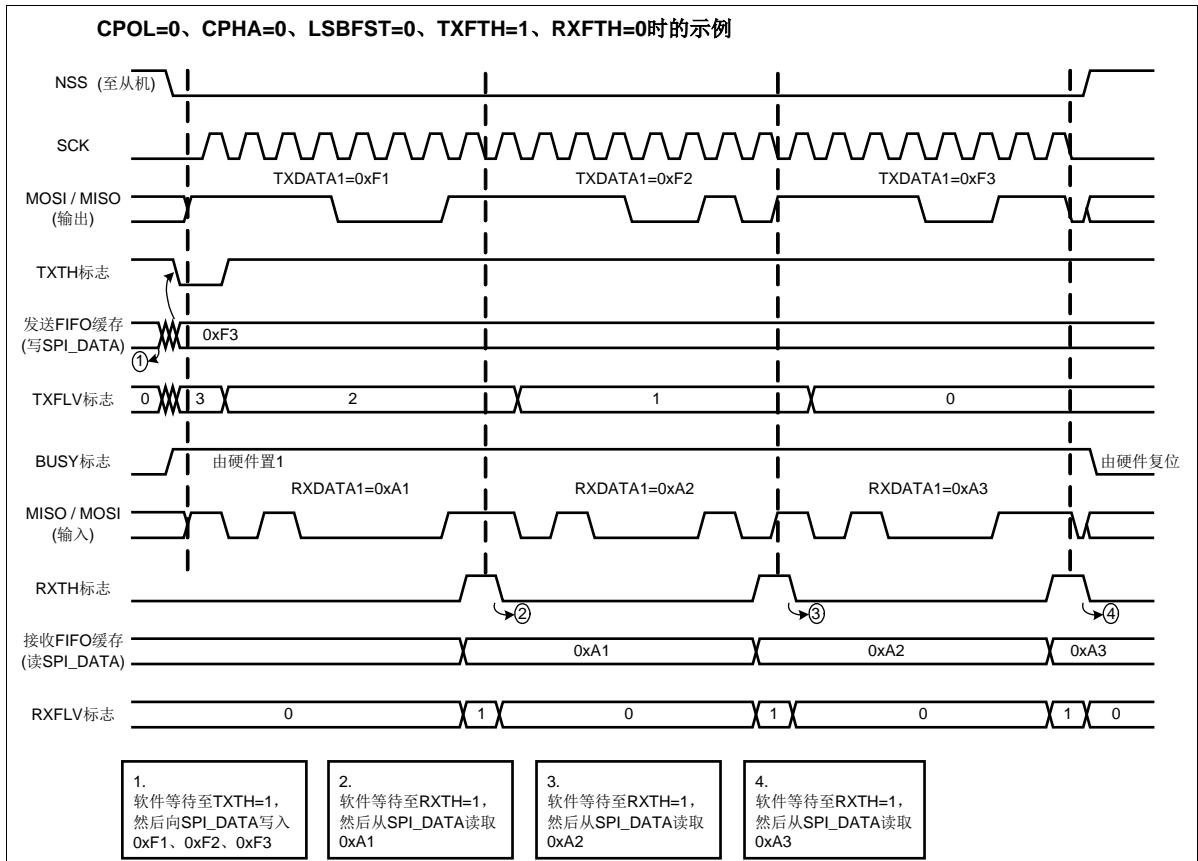


图 21-9 全双工通信时序 (FIFO 缓存操作模式, 连续传输)

单工通信-只发送模式 (SPI_CON1.BIDEN=0、SPI_CON1.RXO=0)，发送数据的处理过程。

直接存取操作模式：

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI。
2. 等待 SPI_STAT.TXE=1，然后写入要发送的数据。对每个要发送的数据项重复此步骤。
3. 将最后一个数据写入 SPI_DATA 寄存器后，等待至 SPI_STAT.TXE=1，然后等待至 SPI_STAT.BUSY=0 再关闭 SPI，这表示最后的数据发送完成。
4. 此外，还可以使用在 TXE 中断事件对应的中断子程序来实现该过程。

FIFO 缓存操作模式：

1. 通过将 SPI_CON1.SPIEN 位置 1 来使能 SPI。
2. 配置 SPI_CON2.TXFTH。
3. 当 SPI_STAT.TXTH=1，将要发送的数据写入 SPI_DATA 寄存器（写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值），重复此操作直到写入最后要发送的数据。
4. 等待至 SPI_STAT.BUSY=0 再关闭 SPI。
5. 此外，还可以使用 TXTH 中断事件对应的中断子程序来实现该过程。

注 1：在不连续通信期间，在对 SPI_DATA 寄存器执行写操作与 SPI_STAT.BUSY 位置 1 之间有延迟。因此在只发送模式下，写入最后的数据后，必须先等待 SPI_STAT.TXE 位置 1，然后等待 SPI_STAT.BUSY 位清零。

注 2：在只发送模式下，发送 17 个数据项后，SPI_STAT.RXOV 标志将置 1，因为始终不会读取接收的数据。

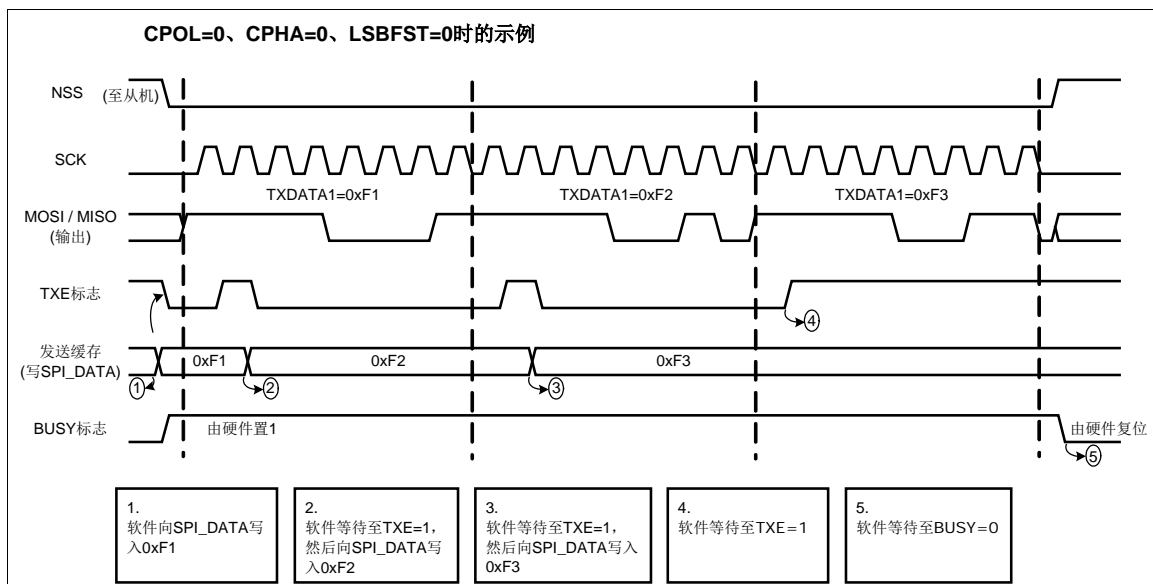


图 21-10 单工通信-只发送模式时序（直接存取操作模式，连续传输）

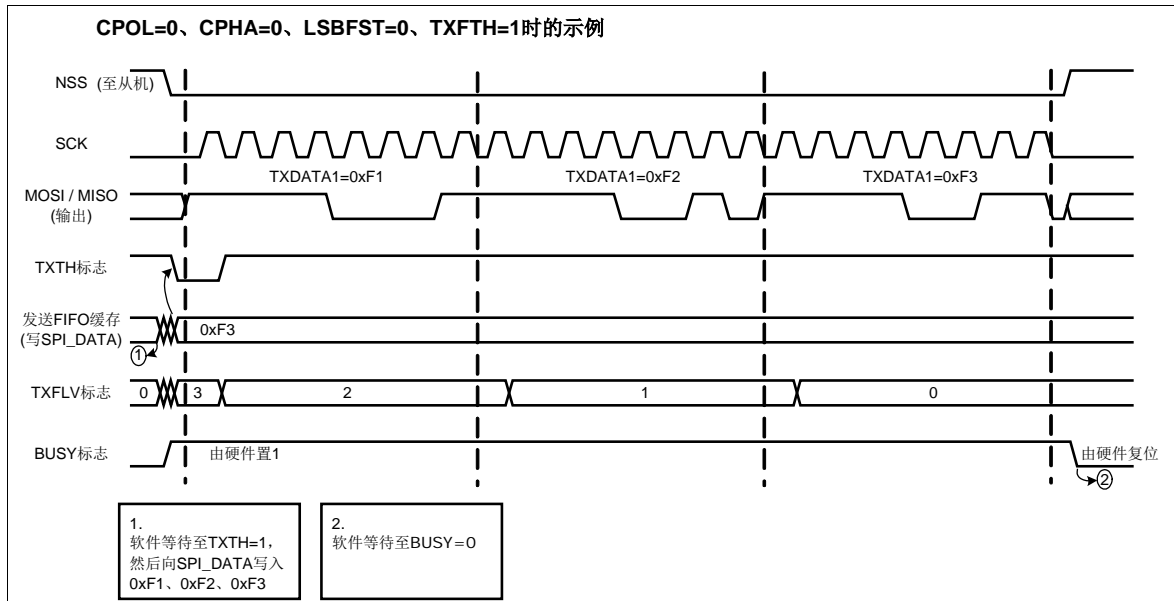


图 21-11 单工通信-只发送模式时序 (FIFO 缓存操作模式, 连续传输)

半双工通信-发送模式 (SPI_CON1.BIDEN=1 且 SPI_CON1.BIDOEN=1)，发送数据的处理过程。

此模式与单工通信-只发送模式数据的处理过程相似，但是在 SPI 模块使能前，必须将 SPI_CON1.BIDEN 位和 SPI_CON1.BIDOEN 位置 1。

单工通信-只接收模式 (SPI_CON1.BIDEN=0 且 SPI_CON1.RXO=1)，接收数据的处理过程。
直接存取操作模式：

1. 将 SPI_CON1.RXO 位置 1。
2. 通过将 SPI_CON1.SPIEN 位置 1 使能 SPI。
3. 等待 SPI_STAT.RXNE=1，然后读取 SPI_DATA 寄存器以获取接收的数据（此操作会将 SPI_STAT.RXNE 位清零）。对每个要接收的数据项重复此操作。
4. 此外，还可以使用 RXNE 中断事件对应的中断子程序来实现该过程。

FIFO 缓存操作模式：

1. 将 SPI_CON1.RXO 位置 1。
2. 配置 SPI_CON2.RXFTH。
3. 通过将 SPI_CON1.SPIEN 位置 1 使能 SPI。
4. 当 SPI_STAT.RXTH=1，读取 SPI_DATA 寄存器以获取接收到的数据（读取的数据个数必须为 SPI_CON2.RXFTH 设定的阈值），重复此操作直到获取最后要接收的数据。
5. 此外，还可以使用 RXTH 中断事件对应的中断子程序来实现该过程。

注 1. 在主机模式下，一旦 SPI 使能后 SCK 会立即发送时钟，从机接收到时钟后会发送数据，直到主机关闭 SPI 功能，结束通信。

注 2. 在从机模式下，当 NSS 被拉低并且接收到 SCK 时钟后开始接收数据。

注 3. 如果需要在最后一次传输后关闭 SPI，请参见 28.5.10 SPI 关闭流程

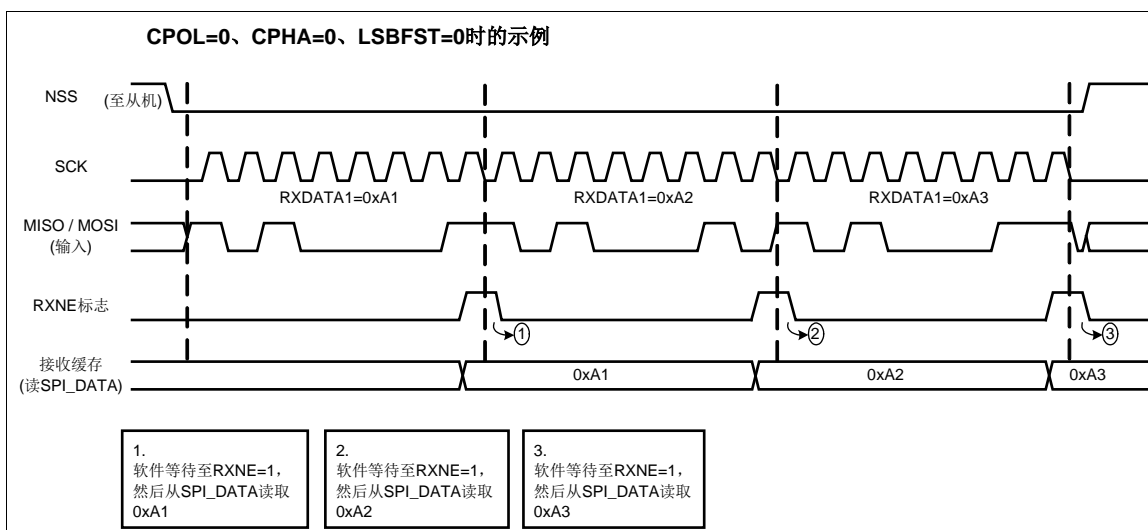


图 21-12 单工通信-只接收模式时序（直接存取操作模式，连续传输）

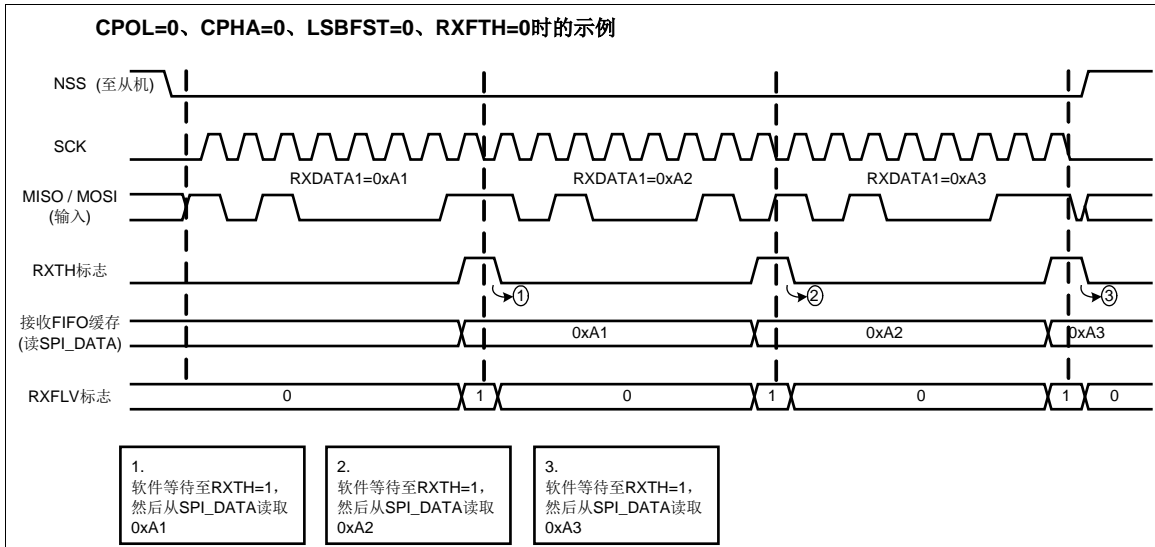


图 21-13 单工通信-只接收模式时序 (FIFO 缓存操作模式, 连续传输)

半双工通信-接收模式 (SPI_CON1.BIDEN=1 和 SPI_CON1.BIDOEN=0), 接收数据的处理过程。

此模式与单工通信-只接收模式数据的处理过程相似, 但是在 SPI 模块使能之前, 需要将 SPI_CON1.BIDEN 位置 1, 并将 SPI_CON1.BIDOEN 与 SPI_CON1.RXO 位清 0。

连续传输和间断传输

在主机模式下发送数据时, 如果软件处理速度足够快, 可以在检测到 SPI_STAT.TXE=1 (或发生 TXE 中断事件), 并且当前数据传输未结束, 立即将下一次的数据写入 SPI_DATA 寄存器, 则能实现连续的通信。或者配置 SPI_CON2.TXFTH, 检测到 SPI_STAT.TXTH=1 (或发生 TXTH 中断事件), 将要发送的数据写入 SPI_DATA 寄存器 (写入的数据个数必须大于 SPI_CON2.TXFTH 设定的阈值), 实现连续的通信。观察到的现象是 SPI_STAT.BUSY 位一直为 1 不被清除, 并且每个数据的 SPI 时钟保持连续。

相反, 如果软件速度不够快, 则可能导致通信中断。在这种情况下, 各数据传输之间会清零 SPI_STAT.BUSY 位。

在主机或从机模式下的单工通信-只接收模式 (SPI_CON1.RXO=1), 通信始终是连续的, 且 SPI_STAT.BUSY 位始终为 1。

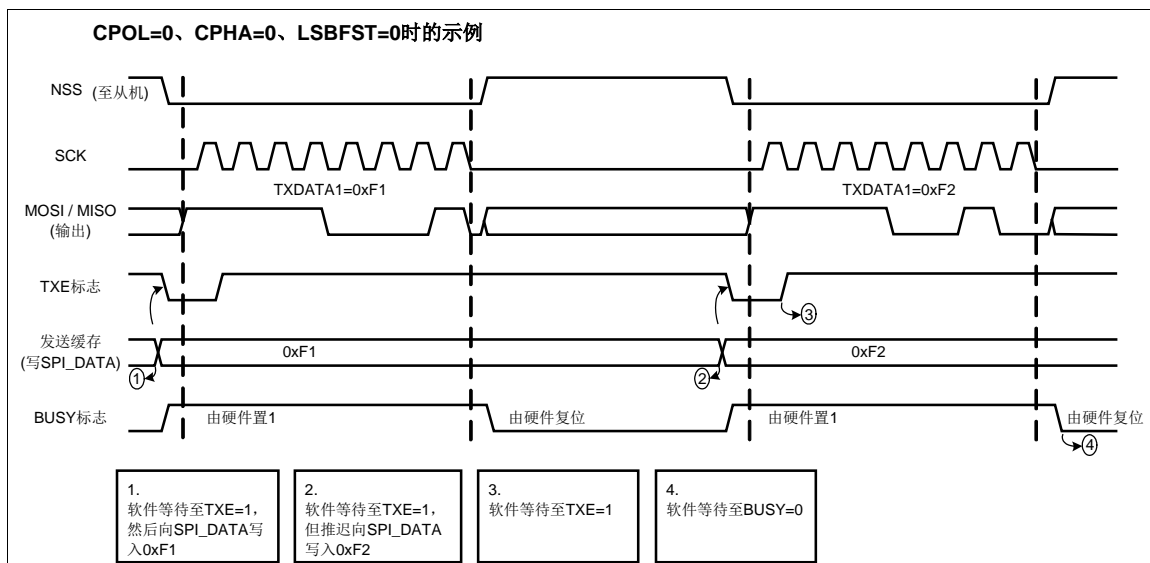


图 21-14 发送时 (BIDEN =0 且 RXO=0) 的时序 (间断传输)

21.5.10 SPI关闭流程

传输终止时，通过清除 SPI_CON1.SPIEN 位来关闭 SPI 模块。

建议在关闭 SPI 时按以下步骤操作：

主机或从机的全双工通信模式 (BIDEN=0、RXO=0)

1. 等待 SPI_STAT.RXNE=1 或 SPI_STAT.RXFLV!=0 以接收最后的数据。
2. 等待 SPI_STAT.TXE=1，并且 SPI_STAT.BUSY=0。
3. 设置 SPI_CON1.SPIEN=0 以关闭 SPI，最后进入停止模式（或关闭外设时钟）。

主机或从机的单工通信-只发送模式 (BIDEN=0、RXO=0) 或半双工通信-发送模式 (BIDEN=1、BIDOEN=1)

最后的数据写入 SPI_DATA 寄存器后：

1. 等待 SPI_STAT.TXE=1。
2. 然后等待 SPI_STAT.BUSY=0。
3. 设置 SPI_CON1.SPIEN=0 以关闭 SPI，最后进入停止模式（或关闭外设时钟）。

主机的单工通信-只接收模式 (MSTREN=1、BIDEN=0、RXO=1) 或半双工通信-接收模式 (MSTREN=1、BIDEN=1、BIDOEN=0)

避免多余的 SPI 数据传输，必须以特殊方式管理这种情况：

1. 等待倒数第二个数据（第 n-1 个）对应的 RXNE 标志位置 1。
2. 在单工通信模式下将 SPI_CON1 寄存器的 RXO 位清零。在半双工通信模式下则将 SPI_CON1 寄存器的 BIDOEN 置 1。
3. 再等待最后的 SPI_STAT.RXNE=1 或 SPI_STAT.RXFLV!=0，才能关闭 SPI (SPIEN=0) 然后进入停止模式（或关闭外设时钟）。

从机的单工通信-只接收从模式 (MSTREN=0、BIDEN=0、RXO=1) 或半双工通信-接收模式 (MSTREN=0、BIDEN=1、BIDOEN=0)

1. 可以随时关闭 SPI（写入 SPI_CON1.SPIEN=0）。当前传输将舍弃并立即关闭 SPI。
2. 如果要进入停止模式，则必须首先等待至 SPI_STAT.BUSY = 0，才能关闭 SPI (SPIEN=0)，然后才能进入停止模式（或关闭外设时钟）。

21.5.11 DMA请求

为了更方便的实现高速通信，SPI 提供了 DMA 功能。DMA 请求条件是根据 SPI_CON2 寄存器中的 TXFTH 与 RXFTH 位域配置，当使能 SPI_CON2 寄存器中相应的 DMA 使能位时，将请求 DMA 访问。发送 FIFO 缓存和接收 FIFO 缓存会发出各自的 DMA 请求：

- ◆ 在发送过程中，当 SPI_STAT.TXTH 位置 1 时会发出 DMA 请求。DMA 随后对 SPI_DATA 寄存器执行写操作（此操作会将 SPI_STAT.TXTH 位清零）。
- ◆ 在接收过程中，当 SPI_STAT.RXTH 位置 1 时会发出 DMA 请求。DMA 随后对 SPI_DATA 寄存器执行读操作（此操作会将 SPI_STAT.RXTH 位清零）。

当 SPI 仅用于只发送数据时，可以只使能 SPI TX DMA 通道。在这种情况下，SPI_STAT.RXOV 位会置 1，因为未读取接收的数据。

当 SPI 仅用于接收数据时，可以只使能 SPI RX DMA 通道。

在发送模式下，DMA 完成了所有要发送数据的传输后，DMA_RIF 寄存器会产生相对应通道的传输完成标志，使用者可以对 BUSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须等待 SPI_STAT.BUSY=0。

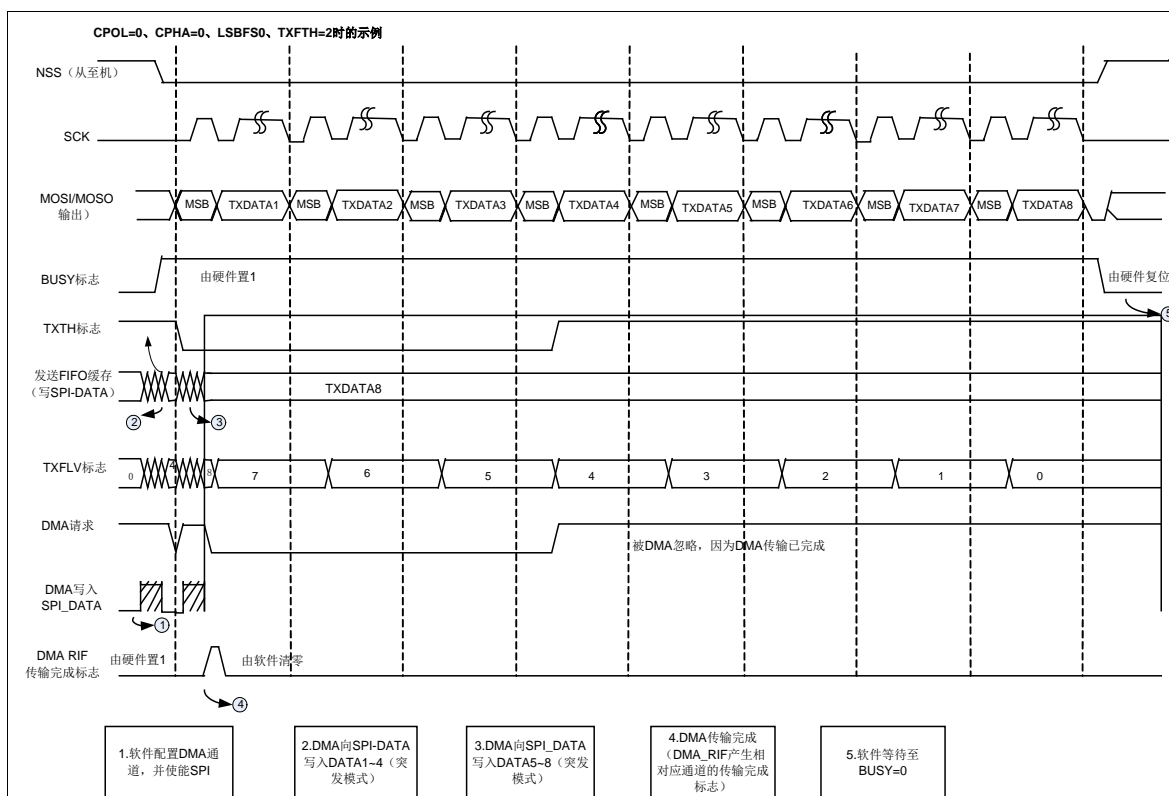


图 21-15 使用 DMA 进行发送

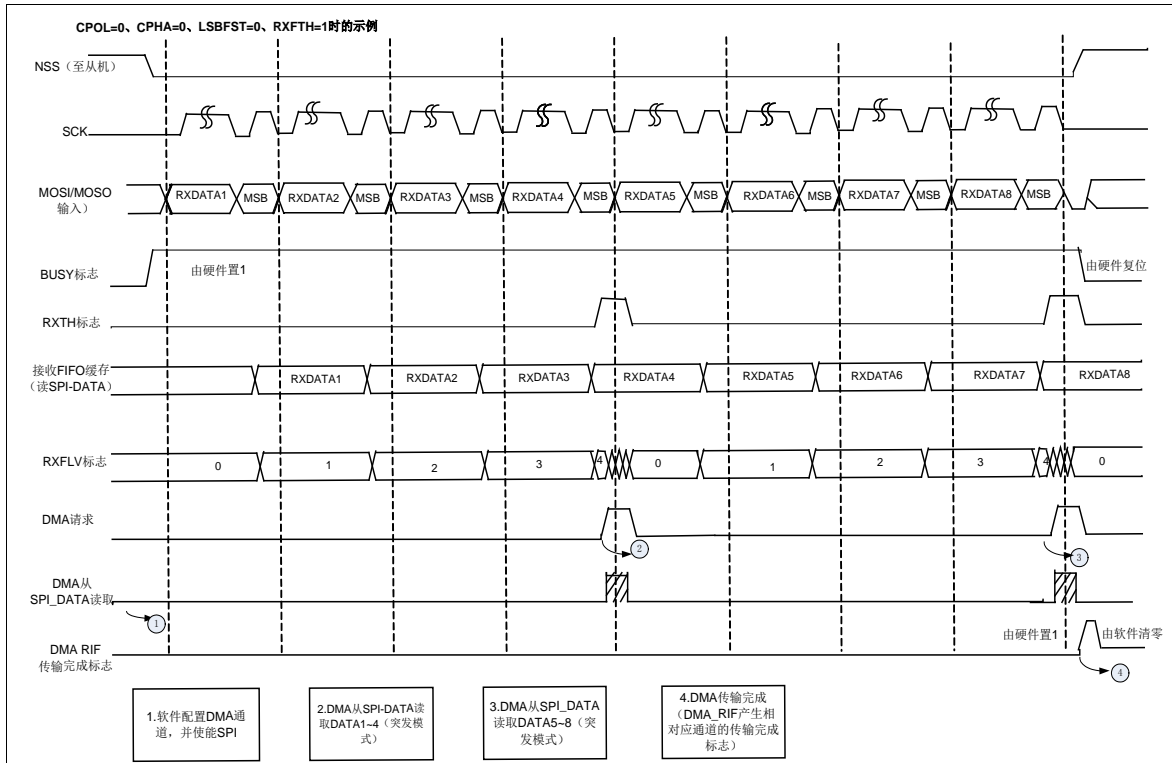


图 21-16 使用 DMA 进行接收

21.5.12 CRC计算

为确保通信的可靠性，SPI 模块实现了硬件 CRC 功能。

针对发送或接收的数据帧宽度有 8 位和 16 位的选择，硬件 CRC 计算也提供了两种计算标准，分别为 8 位数据的 CRC8 和 16 位数据的 CRC16。CRC 是使用 SPI_CRCPOLY 寄存器中设置的多项式串行计算的。

将 SPI_CON1.CRCEN 位置 1 来使能 CRC 计算功能，此操作会复位 CRC 寄存器(SPI_RXCRC 和 SPI_TXCRC)。在全双工或只发送模式下，如果传输由软件（CPU 模式）管理，在连续传输的情况下，可以在最后一个数据写入前的任意时间点将 SPI_CON1.NXTCRC 位置 1，当最后一次数据传输结束时，将发送 SPI_TXCRC 寄存器内的值。在间断传输的情况下，必须在最后传输的数据写入 SPI_DATA 后，立即对 SPI_CON1.NXTCRC 位执行写操作，当最后一次数据传输结束时，将发送 SPI_TXCRC 寄存器内的值。如果传输由 DMA 管理，则在使能发送 FIFO 缓存 DMA 前，将 SPI_CON1.NXTCRC 位置 1，当最后一次数据传输结束时，将发送 SPI_TXCRC 寄存器内的值。

在只接收模式下，如果传输由软件（CPU 模式）管理，在连续传输的情况下，可以在接收到最后一个数据前将 SPI_CON1.NXTCRC 位置 1，在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。在间断传输的情况下，则在接收到倒数第二个数据后，必须对 SPI_CON1.NXTCRC 位执行写操作，在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。如果传输由 DMA 管理，则在使能接收 FIFO 缓存 DMA 前，将 SPI_CON1.NXTCRC 位置 1，在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。

如果传输过程中出现数据损坏，则在数据和 CRC 传输结束时，SPI_RIF.CRCERR 位将置 1。

如果发送 FIFO 缓存中存在数据，则只有在发送完数据字节后才会发送 CRC 值。在 CRC 发送期间，CRC 计算器处于关闭状态且寄存器值保持不变。

可通过以下步骤使用 CRC 进行 SPI 通信：

1. 对 SPI_CON1.BAUD、SPI_CON1.CPOL、SPI_CON1.CPHA、SPI_CON1.LSBFST、SPI_CON1.SSEN、SPI_CON1.SSOUT 和 SPI_CON1.MSTREN 值进行设置。
2. 向 SPI_CRCPOLY 寄存器中写入计算 CRC 的多项式。
3. 通过将 SPI_CON1.CRCEN 位置 1 来使能 CRC 计算。此操作还会将 SPI_RXCRC 和 SPI_TXCRC 寄存器清零。
4. 通过将 SPI_CON1.SPIEN 位置 1 使能 SPI。
5. 启动并保持通信，直到只剩下一个字节或半字未发送或接收。
 - ◇ 在全双工或只发送模式下，如果传输由软件管理，在连续传输的情况下，可以在最后一个数据写入前的任意时间点将 SPI_CON1.NXTCRC 位置 1，以表示在发送完最后一个字节后将发送 CRC。在间断传输的情况下，必须在最后传输的数据写入 SPI_DATA 后，立即对 SPI_CON1.NXTCRC 位执行写操作，以表示在发送完最后一个字节后将发送 CRC。

- ◇ 在只接收模式下，在连续传输的情况下，可以在接收到最后一个数据前将 SPI_CON1.NXTCRC 位置 1，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在间断传输的情况下，则在接收到倒数第二个数据后，必须对 SPI_CON1.NXTCRC 位执行写操作，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在 CRC 传输期间，CRC 计算将冻结。
- 6. 传输完最后一个字节或半字后，SPI 进入 CRC 传输和校验阶段。在全双工模式或只接收模式下，将接收的 CRC 与 SPI_RXCRC 值进行比较。如果两个值不匹配，则 SPI_RIF.CRCERR 位将置 1，并且在 SPI_IER.CRCERR 位置 1 时会产生中断请求。

当 SPI 处于从机模式时，注意只能在时钟稳定（时钟处于空闲电平）时使能 CRC 计算。否则，可能导致 CRC 计算错误。

在 SPI 通信时钟频率较高的情况下，发送 CRC 时务必小心。在 CRC 传输阶段 CPU 应尽可能保持空闲，因此禁止在 CRC 发送阶段调用函数，以避免最后的数据和 CRC 接收出错。实际上在发送或接收最后的数据之前必须对 SPI_CON1.NXTCRC 位执行写操作。

SPI 通信时钟频率较高时，建议使用 DMA 模式来避免由于 CPU 访问影响 SPI 带宽而导致 SPI 速度性能下降。

如果将 SPI 配置为从机，并且使用 NSS 硬件模式，则需要在数据阶段和 CRC 阶段之间将 NSS 引脚保持为低电平。

在对从机片选的切换期间内，应在主机和从机两端同时将 CRC 值清零，以重新同步主机和从机双方的 CRC 计算。

要将 CRC 值清零，请按以下步骤操作：

1. 将 SPI_CON1.CRCEN 位清零。
2. 将 SPI_CON1.CRCEN 位置 1。

21.5.13 SPI状态标志

21.5.13.1 发送FIFO缓存为空 (TXE)

此标志置 1 时,表示发送 FIFO 缓存为空,此时可以将待发送的数据加载到发送 FIFO 缓存中。对 SPI_DATA 寄存器执行写操作时,会将 TXE 标志清零。

21.5.13.2 发送FIFO缓存为满 (TXF)

此标志置 1 时,表示发送 FIFO 缓存为满,此时无法将待发送的数据加载到发送 FIFO 缓存中。当从发送 FIFO 缓存加载一个数据到移位寄存器时,会将 TXF 标志清零。

21.5.13.3 发送FIFO缓存上溢 (TXOV)

当发送 FIFO 缓存已满时,对 SPI_DATA 寄存器执行写操作。在这种情况下,新写入的数据不会加载到发送 FIFO 缓存中,并将此标志置 1。对 SPI_STAT 寄存器执行读访问时,将 TXOV 标志清零。

21.5.13.4 发送FIFO缓存下溢 (TXUD)

在从机模式下当发送 FIFO 缓存为空时,但主机提出数据请求。在这种情况下,不会有数据从发送 FIFO 缓存加载到移位寄存器中,并将此标志置 1。对 SPI_STAT 寄存器执行读访问时,将 TXUD 标志清零。

21.5.13.5 发送FIFO缓存阈值 (TXTH)

此标志置 1 时,表示发送 FIFO 缓存中的有效数据个数少于或者等于 SPI_CON2.TXFTH 设置的值,此时可以将待发送的数据加载到发送 FIFO 缓存中。当加载到 FIFO 缓存中的有效数据个数大于 SPI_CON2.TXFTH 设置的值时,会将 TXTH 标志清零。

21.5.13.6 接收FIFO缓存为非空 (RXNE)

此标志置 1 时,表示接收 FIFO 缓存中存在有效的已接收数据。此时可读取 SPI_DATA 寄存器,当读取后接收 FIFO 缓存中没有有效数据时,此标志位会被清零。

21.5.13.7 接收FIFO缓存为满 (RXF)

此标志置 1 时,表示接收 FIFO 缓存为满,此时无法将接收的数据加载到接收 FIFO 缓存中。对 SPI_DATA 寄存器执行读访问时,将 RXF 标志清零。

21.5.13.8 接收FIFO缓存上溢 (RXOV)

当接收 FIFO 缓存已满时,没对 SPI_DATA 寄存器执行读访问。在这种情况下,主机发送的下一个数据帧不会加载到接收 FIFO 缓存中,同时将此标志置 1。对 SPI_STAT 寄存器执行读访问时,会将 RXOV 标志清零。

21.5.13.9 接收FIFO缓存下溢 (RXUD)

当接收 FIFO 缓存为空时,但对 SPI_DATA 寄存器执行读访问。在这种情况下,读访问不会从接收 FIFO 缓存中读到有效的数据,并将此标志置 1。对 SPI_STAT 寄存器执行读访问时,会

将 RXUD 标志清零。

21.5.13.10 接收FIFO缓存阈值 (RXTH)

此标志置 1 时，表示接收 FIFO 缓存中的有效数据个数大于或者等于 SPI_CON2.RXFTH 设置的值，此时对 SPI_DATA 寄存器执行读访问读取接收 FIFO 缓存中的数据。当读取到 FIFO 缓存中的有效数据个数少于 SPI_CON2.RXFTH 设置的值时，会将 RXTH 标志清零。

21.5.13.11 通信忙 (BUSY)

BUSY 标志用于指示 SPI 通信的状态。此标志由硬件置 1 和清零。

SPI_STAT.BUSY 位置 1 时，表示 SPI 正在通信中。在通信结束前，可检测 SPI_STAT.BUSY 位是否为 0，如果为 0 则表示通信已结束，此时关闭 SPI 模块停止通信。

BUSY 标志还可用于避免在多主机模式系统中发生写冲突。

在以下情况硬件将清零该标志：

- ◆ 传输完成时（主机模式下的连续通信除外）
- ◆ 关闭 SPI 时
- ◆ 发生模式错误时（SPI_RIF.MODF=1）

当通信不连续时，BUSY 标志在各通信之间处于低电平。

当通信连续时，BUSY 标志在所有传输期间均保持高电平。

注：请勿使用 BUSY 标志处理每次数据发送或接收，最好改用 TXTH 标志和 RXTH 标志。

21.5.14 SPI中断事件

21.5.14.1 发送FIFO缓存为空 (TXE)

下列两种情况将产生 TXE 的中断事件

- ◆ SPI_RIF 寄存器的 TXE 位默认值为 0，当发送 FIFO 缓存为空 (SPI_STAT.TXE=1)，且对 SPI_IER 寄存器中的 TXE 位置 1 时。在这种情况下，SPI_RIF 寄存器的 TXE 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 TXE 位置 1，会将 SPI_RIF 寄存器的 TXE 位清零并清除中断。
- ◆ 当发送 FIFO 缓存中的最后一个有效数据加载到移位寄存器时。在这种情况下，SPI_RIF 寄存器的 TXE 位会被设置为 1，如果 SPI_IER 寄存器中的 TXE 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXE 位置 1，会将 SPI_RIF 寄存器的 TXE 位清零并清除中断。

21.5.14.2 发送FIFO缓存上溢 (TXOV)

当发送 FIFO 缓存已满 (SPI_STAT.TXF=1) 时，对 SPI_DATA 寄存器执行写操作。在这种情况下，SPI_RIF 寄存器的 TXOV 位会被设置为 1，如果 SPI_IER 寄存器中的 TXOV 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXOV 位置 1，会将 SPI_RIF 寄存器的 TXOV 位清零并清除中断。

21.5.14.3 发送FIFO缓存下溢 (TXUD)

在从机模式下当发送 FIFO 缓存为空时，但主机提出数据请求。在这种情况下，SPI_RIF 寄存器的 TXUD 位会被设置为 1，如果 SPI_IER 寄存器中的 TXUD 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXUD 位置 1，会将 SPI_RIF 寄存器的 TXUD 位清零并清除中断。

21.5.14.4 发送FIFO缓存阈值 (TXTH)

下列两种情况将产生 TXTH 的中断事件

- ◆ SPI_RIF 寄存器的 TXTH 位默认值为 0，当 SPI_STAT.TXTH=1 时，且对 SPI_IER 寄存器中的 TXTH 位置 1。在这种情况下，SPI_RIF 寄存器的 TXTH 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 TXTH 位置 1，会将 SPI_RIF 寄存器的 TXTH 位清零并清除中断。
- ◆ 当发送 FIFO 缓存中的数据加载到移位寄存器时，使 FIFO 缓存中的有效数据个数等于 SPI_CON2.TXFTH 设置的值。在这种情况下，SPI_RIF 寄存器的 TXTH 位会被设置为 1，如果 SPI_IER 寄存器中的 TXTH 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXTH 位置 1，会将 SPI_RIF 寄存器的 TXTH 位清零并清除中断。

21.5.14.5 接收FIFO缓存为非空 (RXNE)

下列两种情况将产生 RXNE 的中断事件

- ◆ 当 SPI_STAT.RXNE=1 时，且对 SPI_IER 寄存器中的 RXNE 位置 1。在这种情况下，SPI_RIF 寄存器的 RXNE 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXNE 位置 1，会将 SPI_RIF 寄存器的 RXNE 位清零并清除中断。
- ◆ 当接收的数据加载到 FIFO 缓存中，使接收 FIFO 缓存为非空的时候。在这种情况下，

SPI_RIF 寄存器的 RXNE 位会被设置为 1，如果 SPI_IER 寄存器中的 RXNE 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXNE 位置 1，会将 SPI_RIF 寄存器的 RXNE 位清零并清除中断。

21.5.14.6 接收FIFO缓存为满 (RXF)

下列两种情况将产生 RXF 的中断事件

- ◆ 当 SPI_STAT.RXF=1 时，且对 SPI_IER 寄存器中的 RXF 位置 1。在这种情况下，SPI_RIF 寄存器的 RXF 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXF 位置 1，会将 SPI_RIF 寄存器的 RXF 位清零并清除中断。
- ◆ 当接收的数据加载到 FIFO 缓存中，使接收 FIFO 缓存为满的时候。在这种情况下，SPI_RIF 寄存器的 RXF 位会被设置为 1，如果 SPI_IER 寄存器中的 RXF 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXF 位置 1，会将 SPI_RIF 寄存器的 RXF 位清零并清除中断。

21.5.14.7 接收FIFO缓存上溢 (RXOV)

当接收 FIFO 缓存已满时，下一个接收的数据帧不会加载到接收 FIFO 缓存中。在这种情况下，SPI_RIF 寄存器的 RXOV 位会被设置为 1，如果 SPI_IER 寄存器中的 RXOV 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXOV 位置 1，会将 SPI_RIF 寄存器的 RXOV 位清零并清除中断。

21.5.14.8 接收FIFO缓存下溢 (RXUD)

当接收 FIFO 缓存为空时，但对 SPI_DATA 寄存器执行读访问。在这种情况下，SPI_RIF 寄存器的 RXUD 位会被设置为 1，如果 SPI_IER 寄存器中的 RXUD 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXUD 位置 1，会将 SPI_RIF 寄存器的 RXUD 位清零并清除中断。

21.5.14.9 接收FIFO缓存阈值 (RXTH)

下列两种情况将产生 RXTH 的中断事件

- ◆ 当 SPI_STAT.RXTH=1 时，且对 SPI_IER 寄存器中的 RXTH 位置 1。在这种情况下，SPI_RIF 寄存器的 RXTH 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXTH 位置 1，会将 SPI_RIF 寄存器的 RXTH 位清零并清除中断。
- ◆ 当接收的数据帧加载到接收 FIFO 缓存时，使 FIFO 缓存中的有效数据个数等于 SPI_CON2.RXFTH 设置的值。在这种情况下，SPI_RIF 寄存器的 RXTH 位会被设置为 1，如果 SPI_IER 寄存器中的 RXTH 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXTH 位置 1，会将 SPI_RIF 寄存器的 RXTH 位清零并清除中断。

21.5.14.10 CRC错误 (CRCERR)

当 SPI_CON1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPI_RXCRC 的值不匹配，SPI_RIF 寄存器中的 CRCERR 位将置 1。如果 SPI_IER 寄存器中的 CRCERR 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 CRCERR 位置 1，会将 SPI_RIF 寄存器的 CRCERR 位清零并清除中断。

21.5.14.11 模式故障 (MODF)

当主机的 NSS 引脚拉低 (NSS 硬件模式下) 或 SPI_CON1.SSOUT 位为 0 (NSS 软件模式下) 时, 会发生模式错误, 自动将 SPI_RIF.MODF 位置 1。模式错误会在以下几方面影响 SPI 外设:

- ◆ 如果 SPI_IER 寄存器中的 MODF 位置 1 则产生中断。
- ◆ SPI_CON1.SPIEN 位清零, 这将关闭所有输出, 并关闭 SPI 接口。
- ◆ SPI_CON1.MSTREN 位清零, 从而强制 SPI 进入从机模式。

通过对 SPI_ICR 寄存器中的 MODF 位置 1, 会将 SPI_RIF 寄存器的 MODF 位清零并清除中断。

为避免包含多个 MCU 的系统中发生多从机模式冲突, 必须在 SPI_RIF.MODF 位清零前将 NSS 引脚拉高。在 SPI_RIF.MODF 位清零后可以将 SPI_CON1.SPIEN 和 SPI_CON1.MSTREN 位恢复到原始状态。

硬件不允许在 SPI_RIF.MODF 位为 1 时将 SPI_CON1.SPIEN 和 SPI_CON1.MSTREN 位置 1。

在多主机模式配置中, 可在 SPI_RIF.MODF 位为 1 时处于从机模式。在这种情况下, SPI_RIF.MODF 位指示系统控制可能存在多主机模式冲突。可使用中断程序从此状态完全恢复, 方法是执行复位或返回到默认状态。

21.5.14.12 TI 模式帧格式错误 (FRE)

如果 SPI 在从机模式下工作, 并配置为符合 TI 模式协议, 则在持续通信期间出现 NSS 脉冲时, 将检测到 TI 模式帧格式错误。出现此错误时, SPI_RIF 寄存器中的 FRE 位将置 1。发生错误时不会关闭 SPI, 但会忽略 NSS 脉冲, 并且 SPI 会等待至下一个 NSS 脉冲, 然后再开始新的传输。由于错误检测可能导致丢失两个数据字节, 因此数据可能会损坏。

如果 SPI_IER 寄存器中的 FRE 位置 1, 则检测到帧格式错误时将产生中断。在这种情况下, 由于无法保证数据的连续性, 应关闭 SPI 并在重新使能 SPI 后, 由主机重新发起通信。

通过对 SPI_ICR 寄存器中的 FRE 位置 1, 会将 SPI_RIF 寄存器的 FRE 位清零并清除中断。

21.5.15 SPI TI 模式

SPI 接口与 TI 协议兼容。SPI_CON2 寄存器的 FRF 位可用于配置 SPI 以符合此协议。

无论 SPI_CON1 寄存器中设置的值如何, 都必须强制时钟极性和相位符合 TI 协议要求。NSS 管理也特定于 TI 协议, 在这种情况下无法通过 SPI_CON1 和 SPI_CON2 寄存器 (SSEN、SSOUT、NSSOE) 配置 NSS 管理。

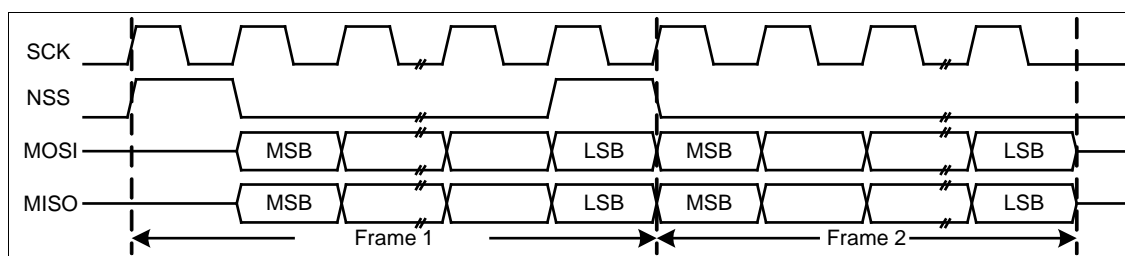


图 21-17 TI 格式的 SPI 通信时序

21.6 I2S结构图

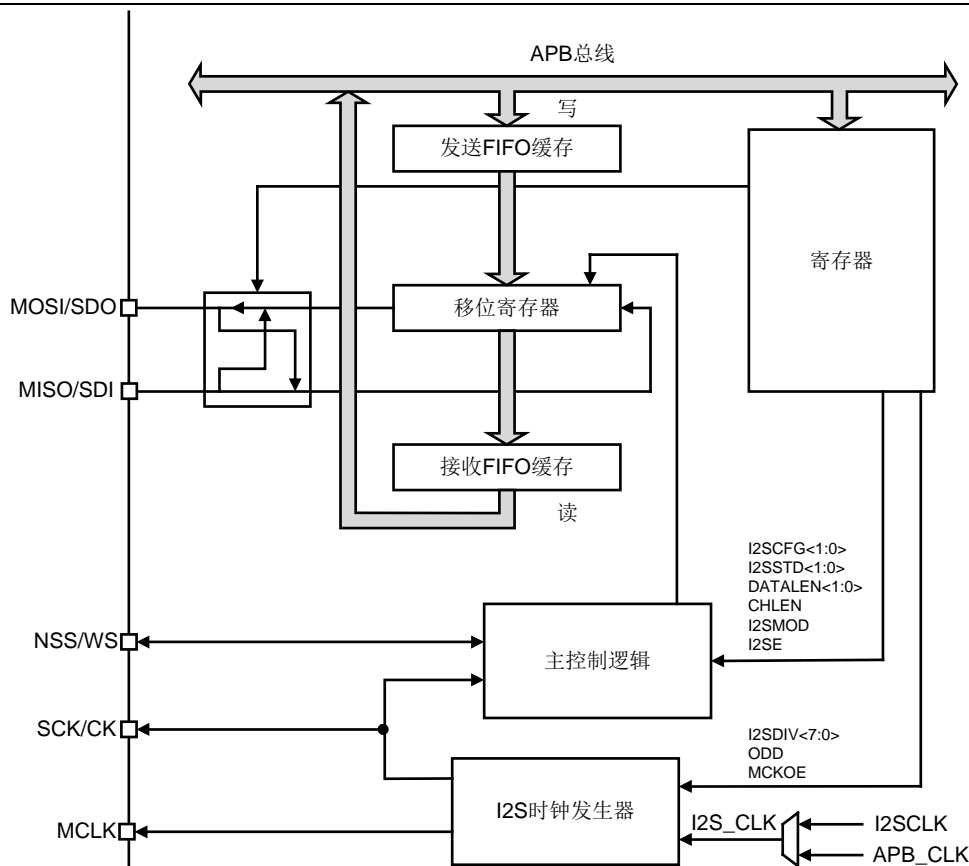


图 21-18 I2S 电路结构框图

当 I2S 功能使能时（通过设置 SPI_I2SCFG.I2SMOD 位），SPI 可用作音频 I2S 接口。该接口主要使用与 SPI 相同的引脚、标志和中断。

I2S 与 SPI 共享四个引脚：

- ◆ SDO: 串行数据输出（映射在 MOSI 引脚上），此引脚在主机模式和从机模式下发送数据。
- ◆ SDI: 串行数据输入（映射在 MISO 引脚上），此引脚在主机模式和从机模式下接收数据。
- ◆ WS: 声道选择（映射在 NSS 引脚上），此引脚在主机模式下输出数据控制信号，在从机模式下输入数据控制信号。
- ◆ CK: 串行时钟（映射在 SCK 引脚上），此引脚在主机模式下作为串行时钟输出，在从机模式下作为串行时钟输入。

当某些外部音频设备需要主时钟输出时，可以使用额外的引脚：

- ◆ MCLK: 当 I2S 配置为主机模式时（以及当 SPI_I2SPR.MCKOE 位置 1 时），使用主时钟（单独映射）输出此附加时钟，输出预先配置的频率等于 $256 \times F_S$ 的附加时钟，其中 F_S 是音频采样频率。

当 I2S 设置为主机模式时，它使用自己的时钟发生器产生通信时钟。在 I2S 模式下有两个额外的寄存器，一个是 SPI_I2SPR 寄存器用于配置时钟发生器，另一个是 SPI_I2SCFG 寄存器配置 I2S 的基本设定（音频标准、从机或主机模式、数据长度、通道长度和时钟极性）。

在 I2S 模式下不使用 SPI_CON1 寄存器和所有 CRC 寄存器，以及不使用 SPI_CON2 寄存器中的 NSSOE、NSSP、FRF 位和所有中断寄存器中的 MODF、CRCERR 位。

I2S 的数据传输与 SPI 使用相同的 SPI_DATA 寄存器进行数据传输。

21.7 I2S功能描述

21.7.1 音频协议

四线总线在处理时分为复用的两个通道（右声道和左声道）上的音频数据。由于只有一个 16 位寄存器进行数据发送或接收，因此当软件向 SPI_DATA 寄存器写入数据或者读取数据，其数据顺序为先左声道然后右声道。SPI_STAT 寄存器中的 CHSIDE 位仅用于观察当前通道状态为左声道或右声道（CHSIDE 对 PCM 协议没有意义）。

提供四种数据和通道长度组合：

- ◆ 16 位通道帧中有一个 16 位数据帧
- ◆ 32 位通道帧中有一个 16 位数据帧
- ◆ 32 位通道帧中有一个 24 位数据帧
- ◆ 32 位通道帧中有一个 32 位数据帧

当在一个 32 位通道帧上使用一个 16 位数据帧时，只有 16 位是有效位，因此仅需要一次读或写操作。另外的 16 位会被硬件强制为 0，无需任何软件操作或 DMA 请求。

在一个 32 位通道帧上使用一个 24 位或 32 位数据帧时，在软件操作情况下则需要对 SPI_DATA 寄存器进行两次 CPU 读或写操作，当使用 DMA 功能时则会产生两次 DMA 读或写操作。对于 24 位数据帧，硬件会将 8 位非有效位强制为 0。

对于所有数据格式和通信标准，始终首先发送最高有效位（MSB 优先）。

I2S 接口支持四种音频标准，可使用 SPI_I2SCFG 寄存器中的 I2SSTD 和 PCMSYNC 位对其进行配置。

21.7.1.1 I2S Philips标准

对于该标准使用 WS 信号来指示正在传输数据的是哪个通道。该信号从当前通道数据的第一个位（MSB）之前的一个 CK 时钟周期开始有效。

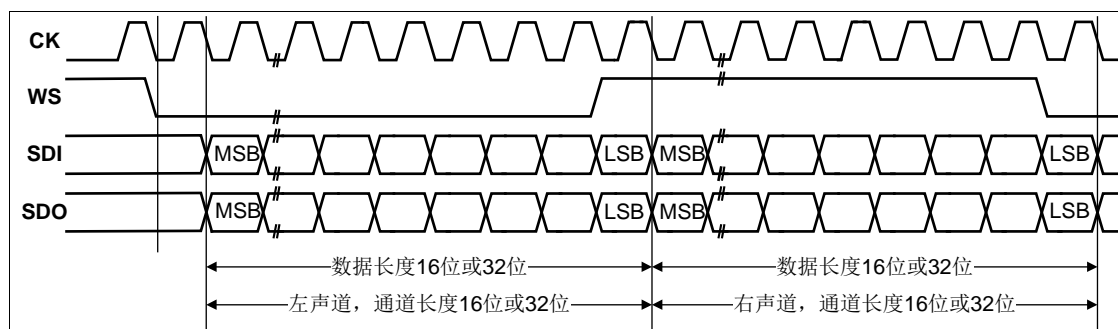


图 21-19 I2S Philips 标准波形（16 或 32 位的数据与通道帧，CKPOL = 0）

发送方在时钟信号（CK）的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

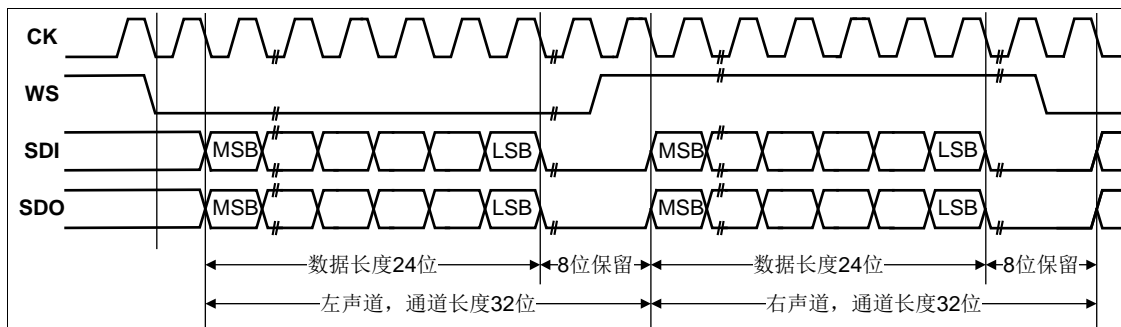


图 21-20 I2S Philips 标准波形（24 位数据帧，CKPOL = 0）

在发送模式下：

如果要发送的数据为 0x123456（24 位），则软件或 DMA 需要对 SPI_DATA 寄存器进行两次写操作。

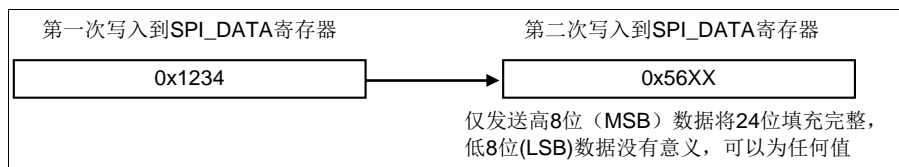


图 21-21 发送数据 0x123456

在接收模式下：

如果接收的数据为 0x123456（24 位），则软件或 DMA 需要对 SPI_DATA 寄存器进行两次读操作。

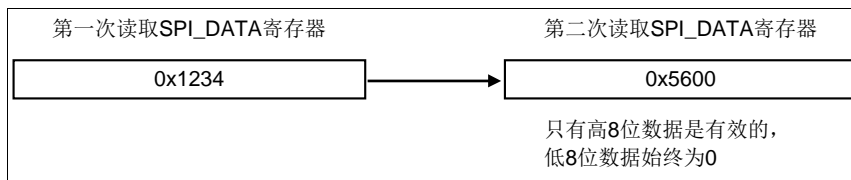


图 21-22 接收数据 0x123456

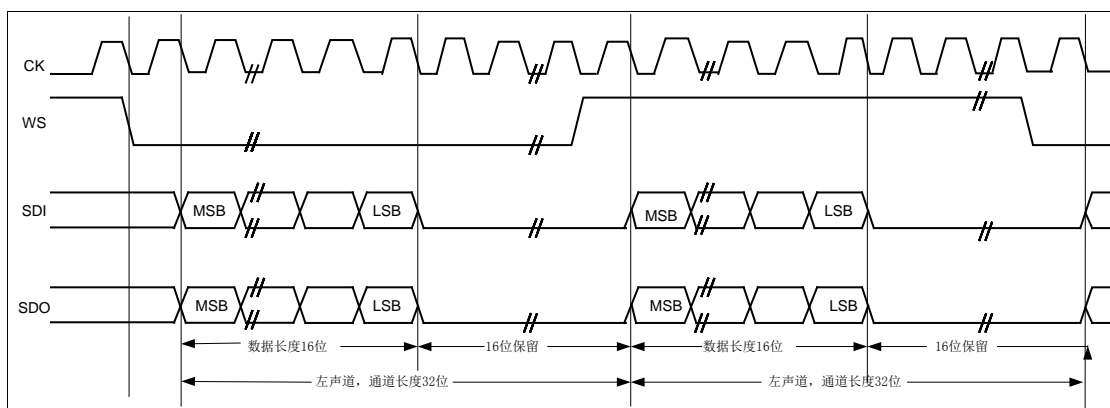


图 21-23 I2S Philips 标准波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）

当在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧时，只需要访问一次 SPI_DATA 寄存器。其余 16 位由硬件强制为 0x0000，以便将数据扩展为 32 位格式。

如果要传输的数据或接收的数据是 0x4567 (0x4567 0000 扩展到 32 位), 则需要执行下图所示的操作。

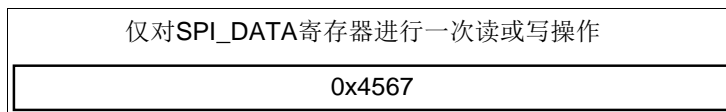


图 21-24 16 位数据帧扩展到 32 位通道帧的示例

21.7.1.2 MSB对齐标准

对于该标准 WS 信号与第一个数据位 (MSB) 同时生成。

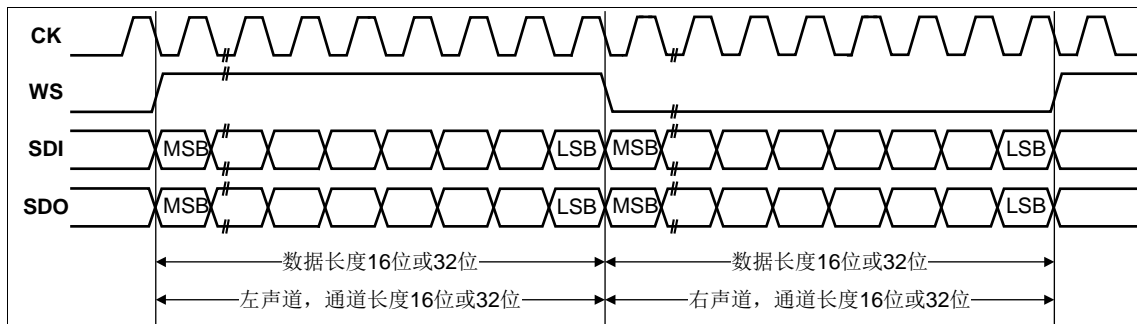


图 21-25 MSB 对齐协议波形 (16 或 32 位的数据与通道帧, CKPOL = 0)

发送方在时钟信号 (CK) 的下降沿改变数据, 接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

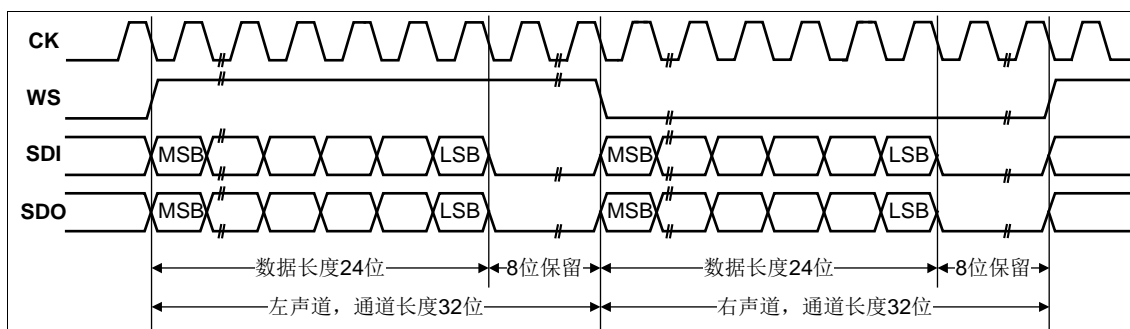


图 21-26 MSB 对齐协议波形 (24 位数据帧, CKPOL = 0)

在发送模式下:

如果要发送的数据为 0x123456 (24 位), 则软件或 DMA 需要对 SPI_DATA 寄存器进行两次写操作。

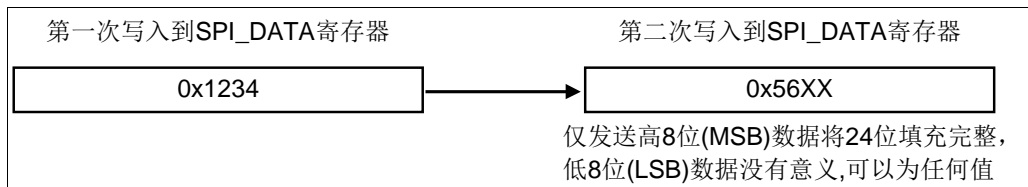


图 21-27 发送数据 0x123456

在接收模式下:

如果接收的数据为 0x123456 (24 位), 则软件或 DMA 需要对 SPI_DATA 寄存器进行两次读操作。

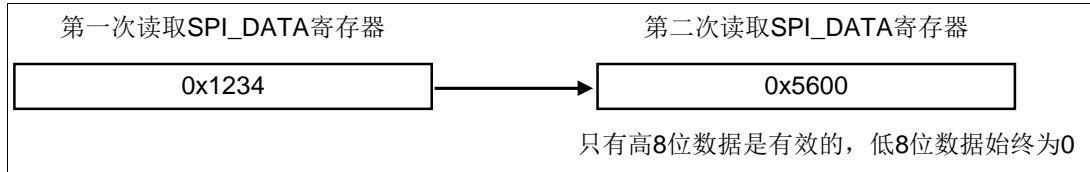


图 21-28 接收数据 0x123456

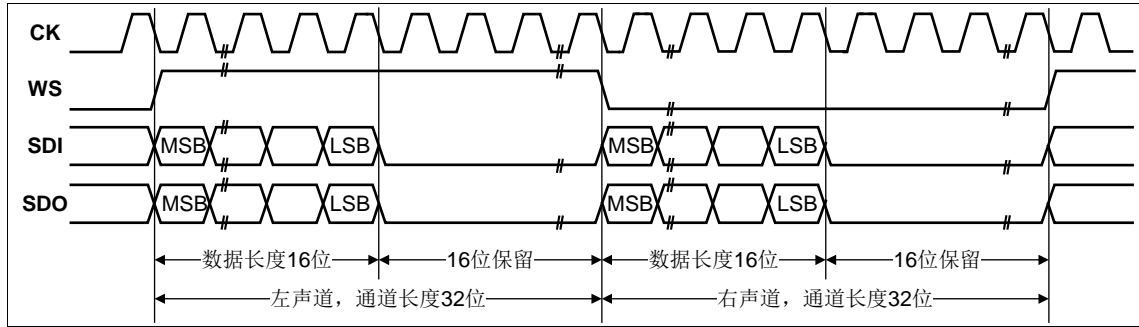


图 21-29 MSB 对齐协议波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）

当在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧时，只需要访问一次 SPI_DATA 寄存器。其余 16 位由硬件强制为 0x0000，以便将数据扩展为 32 位格式。

如果要传输的数据或接收的数据是 0x4567（0x4567 0000 扩展到 32 位），则需要执行下图所示的操作。

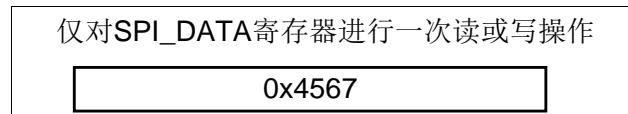


图 21-30 16 位数据帧扩展到 32 位通道帧的示例

21.7.1.3 LSB对齐标准

该标准类似于 MSB 对齐标准（16 位和 32 位通道帧格式没有区别）。

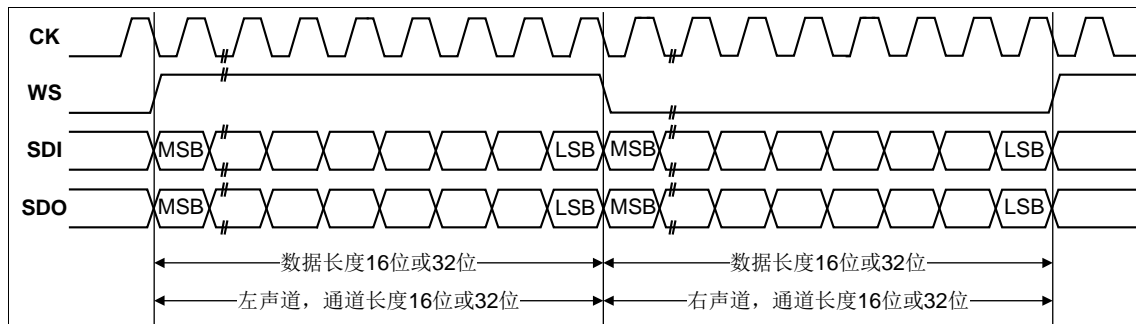


图 21-31 LSB 对齐协议波形（16 或 32 位的数据与通道帧，CKPOL = 0）

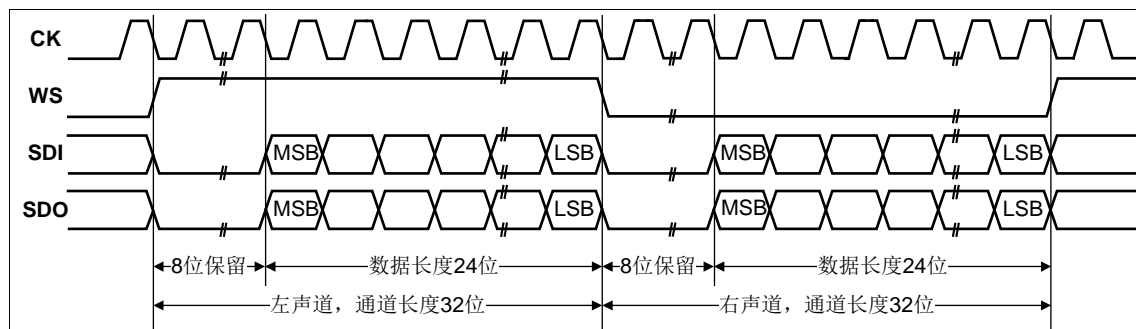


图 21-32 LSB 对齐协议波形（24 位数据帧，CKPOL = 0）

在发送模式下：

如果要发送的数据为 0x123456（24 位），则软件或 DMA 需要对 SPI_DATA 寄存器进行两次写操作。

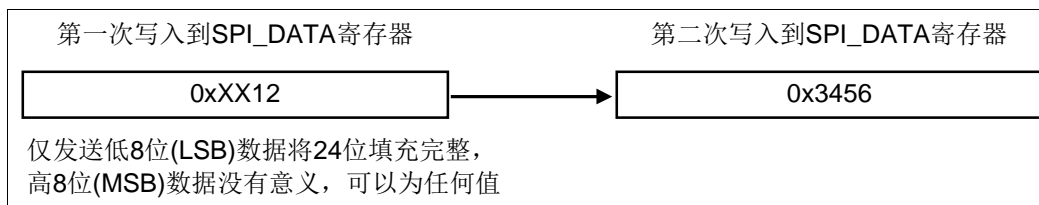


图 21-33 发送数据 0x123456

在接收模式下：

如果接收的数据为 0x123456（24 位），则软件或 DMA 需要对 SPI_DATA 寄存器进行两次读操作。

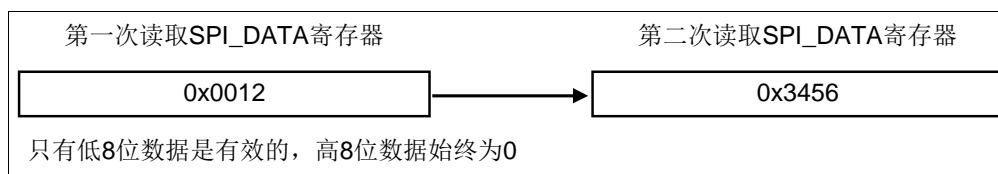


图 21-34 接收数据 0x123456

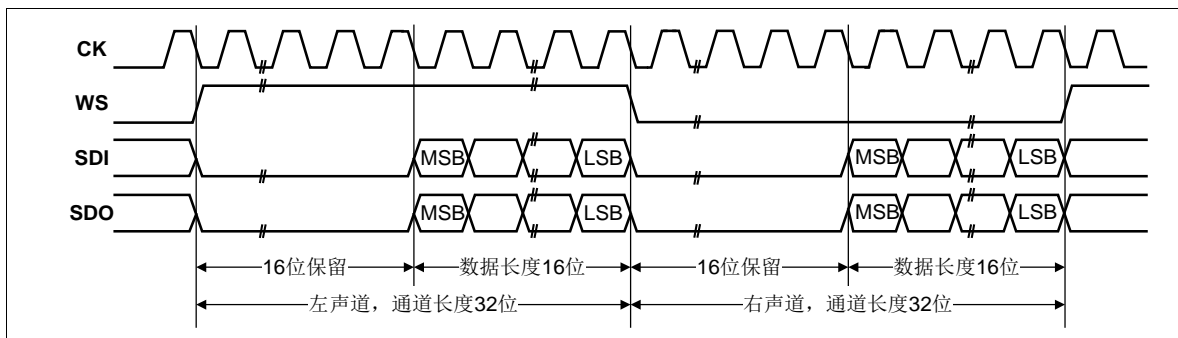


图 21-35 MSB 对齐协议波形（16 位数据帧扩展到 32 位通道帧，CKPOL = 0）

当在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧时，只需要访问一次 SPI_DATA 寄存器。其余 16 位由硬件强制为 0x0000，以便将数据扩展为 32 位格式。

如果要传输的数据或接收的数据是 0x4567（0x0000 4567 扩展到 32 位），则需要执行下图所示的操作。

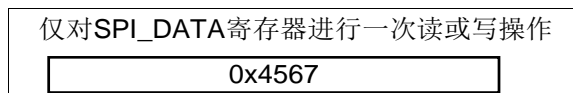


图 21-36 16 位数据帧扩展到 32 位通道帧的示例

21.7.1.4 PCM标准

对于 PCM 标准，无需使用通道信息（CHSIDE）。使用 SPI_I2SCFG 中的 PCMSYNC 位来配置两种 PCM 模式（短帧和长帧）。

对于长帧同步，在主机模式下固定将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号只有一个周期。

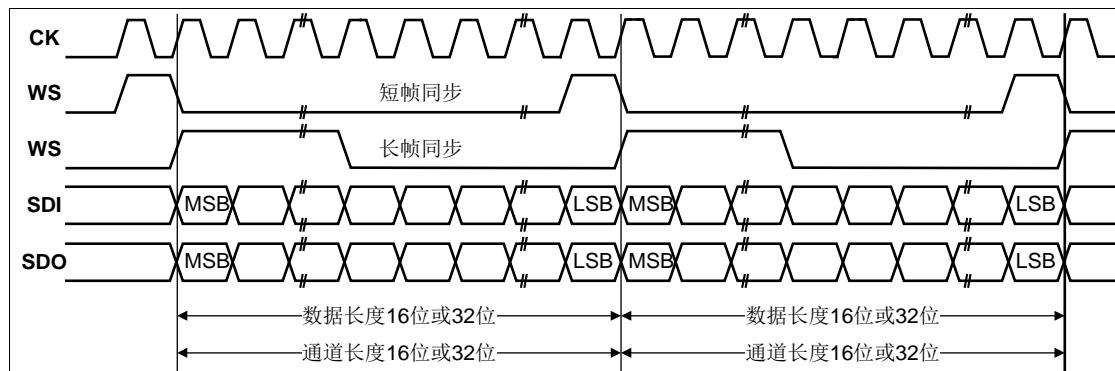


图 21-37 PCM 标准波形（16 或 32 位的数据与通道帧）

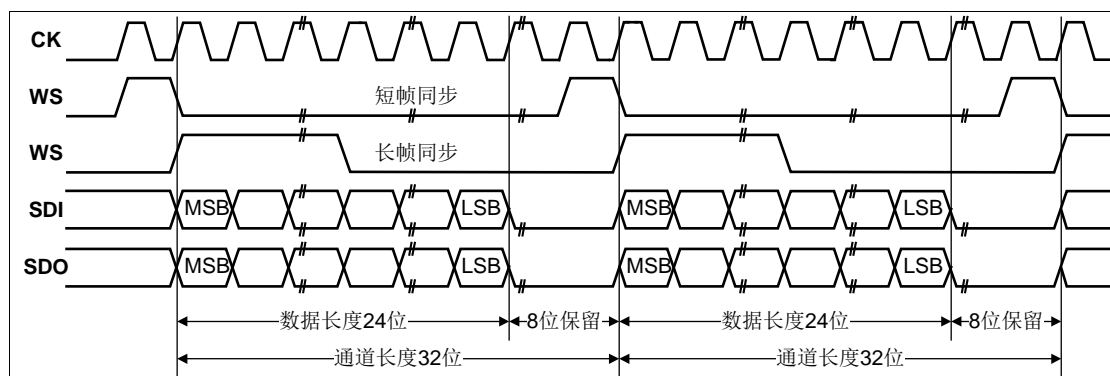


图 21-38 PCM 标准波形（24 位数据帧）

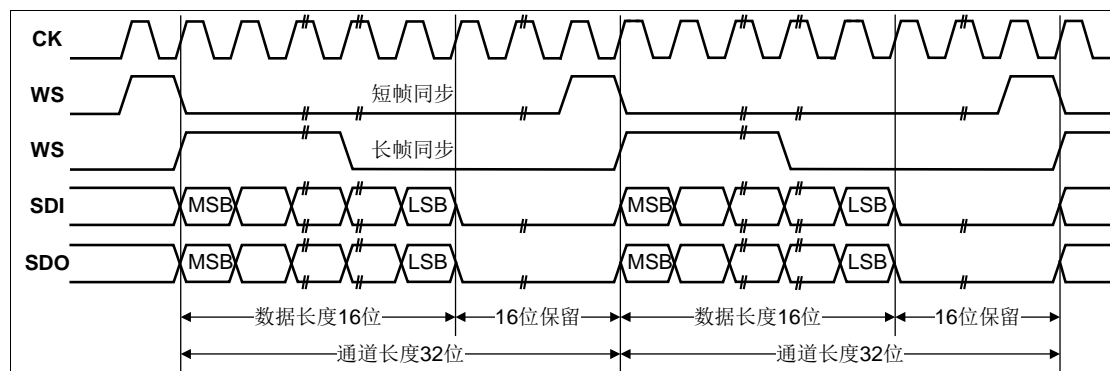


图 21-39 PCM 标准波形（16 位数据帧扩展到 32 位通道帧）

注：对于两种模式（主机和从机）以及两种同步（短帧和长帧），即使在从机模式下，也需要指定两个连续数据（以及两个同步信号）之间的位数（SPI_I2SCFG.DATLEN 和 SPI_I2SCFG.CHLEN 位）。

21.7.2 时钟产生器

I2S 比特率决定 I2S 数据线上的数据流和 I2S 时钟信号频率。

I2S 比特率=每个信道长度×信道数×采样音频

对于信道长度为 16 位的双通道音频（左右声道），I2S 比特率计算如下：

$$\text{I2S 比特率} = 16 \times 2 \times F_s$$

如果信道长度为 32 位的双通道音频（左右声道），则 I2S 比特率= $32 \times 2 \times F_s$ 。

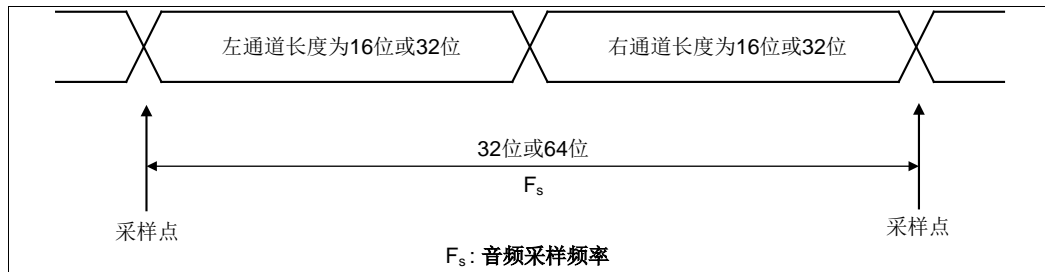


图 21-40 音频采样频率定义

I2S_CLK 时钟源可通过 SPI_I2SPR 寄存器中的 EXTCKEN 位选择 APB_CLK 或是 I2SCLK。要实现高品质的音频性能，建议选择 I2SCLK 作为 I2S_CLK 时钟源。音频采样频率可以是 96kHz、48kHz、44.1kHz、32kHz、22.05kHz、16kHz、11.025kHz 或 8kHz（或该范围内的任何其他值）为了达到所需频率，需要根据以下公式对线性分频器进行编程：

在 I2S Philips、MSB 与 LSB 模式时信道数（CH）为 2。

在 PCM 模式时信道数（CH）为 1。

当产生主时钟时（SPI_I2SPR 寄存器中的 MCKOE 位置 1）：

- ◆ 当通道长度为 16 位时， $F_s = \text{I2S_CLK} / [(16 * 2) * ((\text{CH} * \text{I2SDIV}) + \text{ODD}) * 8]$
- ◆ 当通道长度为 32 位时， $F_s = \text{I2S_CLK} / [(32 * 2) * ((\text{CH} * \text{I2SDIV}) + \text{ODD}) * 4]$

禁用主时钟时（SPI_I2SPR 寄存器中的 MCKOE 位清零）：

- ◆ 当通道长度为 16 位时， $F_s = \text{I2S_CLK} / [(16 * 2) * ((\text{CH} * \text{I2SDIV}) + \text{ODD})]$
- ◆ 当通道长度为 32 位时， $F_s = \text{I2S_CLK} / [(32 * 2) * ((\text{CH} * \text{I2SDIV}) + \text{ODD})]$

下表提供了当 I2S_CLK 时钟源选择 APB_CLK 时，针对不同时钟配置的示例精度值。

APB_CLK (MHz)	Data length	I2SDIV	I2SODD	MCLK	Target fs (Hz)	Real fs (Hz)	Error
48	16	8	0	No	96000	93750	2.3738%
48	32	4	0	No	96000	93750	2.3438%
48	16	15	1	No	48000	48387.0968	0.8065%
48	32	8	0	No	48000	46875	2.3438%
48	16	17	0	No	44100	44117.6471	0.0400%
48	32	8	1	No	44100	44117.6471	0.0400%
48	16	23	1	No	32000	31914.8936	0.2660%
48	32	11	1	No	32000	32608.6957	1.9022%
48	16	34	0	No	22050	22058.8235	0.0400%
48	32	17	0	No	22050	22058.8235	0.0400%
48	16	47	0	No	16000	15957.4468	0.2660%
48	32	23	1	No	16000	15957.4468	0.2660%
48	16	68	0	No	11025	11029.4118	0.0400%
48	32	34	0	No	11025	11029.4118	0.0400%
48	16	94	0	No	8000	7978.7234	0.2660%
48	32	47	0	No	8000	7978.7234	0.2660%
48	16	1	0	Yes	96000	93750	2.3438%
48	32	1	0	Yes	96000	93750	2.3438%
48	16	2	0	Yes	48000	46875	2.3438%
48	32	2	0	Yes	48000	46875	2.3438%
48	16	2	0	Yes	44100	46875	6.2625%
48	32	2	0	Yes	44100	46875	6.2925%
48	16	3	0	Yes	32000	31250	2.3438%
48	32	3	0	Yes	32000	31250	2.3438%
48	16	4	1	Yes	22050	20833.3333	5.5178%
48	32	4	1	Yes	22050	20833.3333	5.5178%
48	16	6	0	Yes	16000	15625	2.3438%
48	32	6	0	Yes	16000	15625	2.3438%
48	16	8	1	Yes	11025	11029.4118	0.0400%
48	32	8	1	Yes	11025	11029.4118	0.0400%
48	16	11	1	Yes	8000	8152.1739	1.9022%
48	32	11	1	Yes	8000	8152.1739	1.9022%

表 21-2 音频频率精度

21.7.3 I2S主机模式

I2S 配置为主机模式时，会从引脚 CK 输出串行时钟和引脚 WS 输出字选择信号，可以通过设置 SPI_I2SPR 寄存器的 MCKOE 位来选择输出或者不输出主时钟（MCLK）。主机模式可配置为单工的发送或接收模式，或者是全双工同时收发模式。

21.7.3.1 设置流程

1. 设置 SPI_I2SPR 寄存器的 I2SDIV 位域与 ODD 位，以定义串行时钟波特率，从而达到适当的音频采样频率。
2. 如果需要将主时钟 MCLK 提供给外部 DAC 或 ADC 音频组件，则将 SPI_I2SPR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD 值应根据 MCLK 输出的状态进行计算，更多详细信息参见“时钟产生器”章节的描述）。
3. 配置 SPI_I2SCFG 寄存器的 CKPOL 位以定义时钟在空闲时的电平状态，将 I2SMOD 位置 1 以激活 I2S 功能，I2S 标准是由 I2SSTD 和 PCMSYNC 位来选择，通过 DATLEN 位域选择数据长度，通过配置 CHLEN 位来选择通道长度。此外，通过 SPI_I2SCFG 寄存器的 I2SCFG 位域选择 I2S 主机模式的配置（单工的发送或接收模式，或者是全双工同时收发模式）。
4. 通过配置 SPI_CON2 寄存器 RXDMA 位或 TXDMA 位来启动 DMA 功能。
5. 通过配置 SPI_IER 寄存器选择所需的中断事件来触发中断。

配置 WS 和 CK 为输出模式。如果 SPI_I2SPR 寄存器的 MCKOE 位置 1，MCLK 也是输出模式，最后设置 SPI_I2SCFG 寄存器的 I2SE 位。

21.7.3.2 发送序列

当半字写入发送 FIFO 缓存时，发送序列随即开始。

写入发送 FIFO 缓存的第一个数据始终对应于左声道数据。当左声道的数据写入发送 FIFO 缓存后必须紧跟着写入对应于右声道数据。CHSIDE 标志指示当前发送的声道。

一个完整帧表示先进行左声道数据发送再进行右声道数据发送。不存在仅发送左声道的部分帧。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 SDO 引脚（MSB 在前）。每次数据从发送 FIFO 缓存加载到移位寄存器，导致缓存中有效数据的个数为零时，TXE 标志将置 1，并且在 SPI_IER.TXE 位置 1 时将生成中断。有关各种 I2S 标准模式中的写操作的更多详细信息，请参见“音频协议”章节的描述。

为了确保连续的音频数据传输，必须在当前传输结束之前将对 SPI_DATA 寄存器写入下一个要传输的数据。若要禁用 I2S，必须等待 TXE=1 且 BUSY=0 时，通过清除 SPI_I2SCFG 寄存器的 I2SE 位来关闭 I2S。

21.7.3.3 接收序列

当接收 FIFO 缓存满时，RXF 标志将置 1，如果 SPI_IER.RXF 位置 1 时则产生中断。根据数据和信道长度配置，右声道或左声道接收到音频数据可通过一次或两次接收操作进入接收 FIFO

缓存。通过读 SPI_DATA 寄存器来获取接收的音频数据。有关各种 I2S 标准模式中的读操作的更多详细信息，请参见“音频协议”章节的描述。

如果在接收 FIFO 缓存满时，且尚未读取先前接收的数据的同时接收到新的数据，则生成溢出并设置 RXOV 标志。如果 SPI_IER 寄存器中的 RXOV 位置 1，则会产生中断以指示错误。

21.7.4 I2S从机模式

I2S 配置为从机模式时，会从引脚 CK 输入串行时钟和引脚 WS 输入字选择信号，在此模式下不输出主时钟（MCLK），因此用户无需配置时钟。从机模式可配置为单工的发送或接收模式，或者是全双工同时收发模式。

21.7.4.1 设置流程

1. 将 SPI_I2SCFG 寄存器的 I2SMOD 位置 1 选择 I2S 模式，I2S 标准是由 I2SSTD 和 PCMSYNC 位来选择，通过 DATLEN 位域选择数据长度，通过配置 CHLEN 位来选择通道长度。此外，通过 SPI_I2SCFG 寄存器的 I2SCFG 位域选择 I2S 从机模式的配置（单工的发送或接收模式，或者是全双工同时收发模式）。
2. 通过配置 SPI_CON2 寄存器 RXDMA 位或 TXDMA 位来启动 DMA 功能。
3. 通过配置 SPI_IER 寄存器选择所需的 interrupt 事件来触发中断。
4. 设置 SPI_I2SCFG 寄存器的 I2SE 位。

21.7.4.2 自动侦测同步

当处于从机模式下，用户设置 SPI_I2SCFG 寄存器的 I2SE 后，硬件会先侦测到两个完整的数据控制信号（WS）。对于 I2S Philips 标准、MSB 对齐标准或 LSB 对齐标准，一个完整的数据控制信号表示包含左右声道。在 PCM 模式下，一个完整的数据控制信号仅包含单声道。当检测完后才会进行发送或接收序列。

21.7.4.3 发送序列

首先配置从机并使能，对要传输的数据写入发送 FIFO 缓存后，外部连接的主机才能开始通信。当外部主机发送时钟并且 WS 信号请求数据传输时，发送序列开始。

对于 I2S，MSB 对齐和 LSB 对齐模式，写入发送 FIFO 缓存的第一个数据始终对应于左声道数据。当左声道的数据写入发送 FIFO 缓存后必须紧跟着写入对应于右声道的数据。通信开始时，数据从发送 FIFO 缓存加载到移位寄存器。

CHSIDE 标志指示当前发送的声道。与主机传输模式相比，在从机模式下 CHSIDE 对应来自外部主机设备的 WS 信号。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 SDO 引脚（MSB 在前）。每次数据从发送 FIFO 缓存加载到移位寄存器，导致缓存中有效数据的个数为零时，TXE 标志将置 1，并且在 SPI_IER.TXE 位置 1 时将生成中断。

注意，在尝试写入发送 FIFO 缓存之前，应检查 TXF 标志为 0。

有关各种 I2S 标准模式中的写操作的更多详细信息，请参见“音频协议”章节的描述。

为了确保连续的音频数据传输，必须在当前传输结束之前将 SPI_DATA 寄存器写入下一个要传输的数据。如果在下一次数据通信的第一个时钟沿之前没有将数据写入 SPI_DATA 寄存器，则会产生下溢标志并产生中断。软件可以获知所传输的数据是错误的。当发生下溢中断时，若要重新开始传输，此时需要先将 I2S 禁用后，重新启动使其重新侦测并对齐左声道输出数据。

若要禁用 I2S，必须等待 TXE=1 时，通过清除 SPI_I2SCFG 寄存器的 I2SE 位来关闭 I2S。

21.7.4.4 接收序列

无论数据长度或通道长度为多少，音频数据都由 16 位数据包进行接收。根据数据和信道长度配置，右声道或左声道接收到音频数据可通过一次或两次接收操作进入接收 FIFO 缓存。通过读 SPI_DATA 寄存器来获取接收的音频数据。

有关各种 I2S 标准模式中的读操作的更多详细信息，请参见“音频协议”章节的描述。

如果在接收 FIFO 缓存满时，且尚未读取先前接收的数据，同时接收到新的数据，则发生接收溢出并设置 RXOV 标志。如果 SPI_IER 寄存器中的 RXOV 位置 1，则会产生中断以指示错误。

注：外部主机应具有通过以 16 位或 32 位音频通道发送与接收数据帧的能力。

21.7.5 I2S状态标志

21.7.5.1 发送FIFO缓存为空 (TXE)

此标志置 1 时，表示发送 FIFO 缓存为空，此时可以将待发送的数据加载到发送 FIFO 缓存中。对 SPI_DATA 寄存器执行写操作时，会将 TXE 标志清零。

21.7.5.2 发送FIFO缓存为满 (TXF)

此标志置 1 时，表示发送 FIFO 缓存为满，此时无法将待发送的数据加载到发送 FIFO 缓存中。当从发送 FIFO 缓存加载一个数据到移位寄存器时，会将 TXF 标志清零。

21.7.5.3 发送FIFO缓存上溢 (TXOV)

当发送 FIFO 缓存已满时，使用者对 SPI_DATA 寄存器执行写操作。在这种情况下，新写入的数据不会加载到发送 FIFO 缓存中，并将此标志置 1。对 SPI_STAT 寄存器执行读访问时，将 TXOV 标志清零。

21.7.5.4 发送FIFO缓存下溢 (TXUD)

在从机模式下当发送 FIFO 缓存为空时，但主机提出数据请求。在这种情况下，不会有数据从发送 FIFO 缓存加载到移位寄存器中，并将此标志置 1。对 SPI_STAT 寄存器执行读访问时，将 TXUD 标志清零。

21.7.5.5 发送FIFO缓存阈值 (TXTH)

此标志置 1 时，表示发送 FIFO 缓存中的有效数据个数少于或者等于 SPI_CON2.TXFTH 设置的值，此时可以将待发送的数据加载到发送 FIFO 缓存中。当加载到 FIFO 缓存中的有效数据个数大于 SPI_CON2.TXFTH 设置的值时，会将 TXTH 标志清零。

21.7.5.6 接收FIFO缓存为非空 (RXNE)

此标志置 1 时,表示接收 FIFO 缓存中存在有效的已接收数据。此时可读取 SPI_DATA 寄存器,当读取后接收 FIFO 缓存中没有有效数据时,此标志位会被清零。

21.7.5.7 接收FIFO缓存为满 (RXF)

此标志置 1 时,表示接收 FIFO 缓存为满,此时无法将接收的数据加载到接收 FIFO 缓存中。对 SPI_DATA 寄存器执行读访问时,将 RXF 标志清零。

21.7.5.8 接收FIFO缓存上溢 (RXOV)

当接收 FIFO 缓存已满时,没对 SPI_DATA 寄存器执行读访问。在这种情况下,主机发送的下一个数据帧不会加载到接收 FIFO 缓存中,同时将此标志置 1。对 SPI_STAT 寄存器执行读访问时,会将 RXOV 标志清零。

21.7.5.9 接收FIFO缓存下溢 (RXUD)

当接收 FIFO 缓存为空时,但使用者对 SPI_DATA 寄存器执行读访问。在这种情况下,读访问不会从接收 FIFO 缓存中读到有效的数据,并将此标志置 1。对 SPI_STAT 寄存器执行读访问时,会将 RXUD 标志清零。

21.7.5.10 接收FIFO缓存阈值 (RXTH)

此标志置 1 时,表示接收 FIFO 缓存中的有效数据个数大于或者等于 SPI_CON2.RXFTH 设置的值,此时对 SPI_DATA 寄存器执行读访问读取接收 FIFO 缓存中的数据。当读取到 FIFO 缓存中的有效数据个数少于 SPI_CON2.RXFTH 设置的值时,会将 RXTH 标志清零。

21.7.5.11 通信忙 (BUSY)

BUSY 标志由硬件置位和清除。当 BUSY 标志置 1 时,表示 I2S 上正在进行数据传输 (I2S 总线忙)。BUSY 标志可以用于检测传输的结束,从而防止最后一次传输损坏。

在以下情况硬件将清零该标志:

- ◆ 关闭 I2S 时
- ◆ 传输完成时

当通信连续时, BUSY 标志在所有传输期间均保持高电平。

注: 请勿使用 BUSY 标志处理每次数据发送或接收,最好改用 TXTH 标志和 RXTH 标志。

21.7.5.12 声道标志 (CHSIDE)

此标志是提供使用者判断此时总线上正在传输的声道为左声道还右声道,只有在 BUSY 标志被设置时有效。该标志在 PCM 标准中没有意义 (短帧和长帧模式)。

21.7.6 I2S中断事件

21.7.6.1 发送FIFO缓存为空 (TXE)

下列两种情况将产生 TXE 的中断事件

- ◆ SPI_RIF 寄存器的 TXE 位默认值为 0，当发送 FIFO 缓存为空 (SPI_STAT.TXE=1)，且对 SPI_IER 寄存器中的 TXE 位置 1 时。在这种情况下，SPI_RIF 寄存器的 TXE 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 TXE 位置 1，会将 SPI_RIF 寄存器的 TXE 位清零并清除中断。
- ◆ 当发送 FIFO 缓存中的最后一个有效数据加载到移位寄存器时。在这种情况下，SPI_RIF 寄存器的 TXE 位会被设置为 1，如果 SPI_IER 寄存器中的 TXE 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXE 位置 1，会将 SPI_RIF 寄存器的 TXE 位清零并清除中断。

21.7.6.2 发送FIFO缓存上溢 (TXOV)

当发送 FIFO 缓存已满 (SPI_STAT.TXF=1) 时，对 SPI_DATA 寄存器执行写操作。在这种情况下，SPI_RIF 寄存器的 TXOV 位会被设置为 1，如果 SPI_IER 寄存器中的 TXOV 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXOV 位置 1，会将 SPI_RIF 寄存器的 TXOV 位清零并清除中断。

21.7.6.3 发送FIFO缓存下溢 (TXUD)

在从机模式下当发送 FIFO 缓存为空时，但主机提出数据请求。在这种情况下，SPI_RIF 寄存器的 TXUD 位会被设置为 1，如果 SPI_IER 寄存器中的 TXUD 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXUD 位置 1，会将 SPI_RIF 寄存器的 TXUD 位清零并清除中断。

21.7.6.4 发送FIFO缓存阈值 (TXTH)

下列两种情况将产生 TXTH 的中断事件

- ◆ SPI_RIF 寄存器的 TXTH 位默认值为 0，当 SPI_STAT.TXTH=1 时，且对 SPI_IER 寄存器中的 TXTH 位置 1。在这种情况下，SPI_RIF 寄存器的 TXTH 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 TXTH 位置 1，会将 SPI_RIF 寄存器的 TXTH 位清零并清除中断。
- ◆ 当发送 FIFO 缓存中的数据加载到移位寄存器时，使 FIFO 缓存中的有效数据个数等于 SPI_CON2.TXFTH 设置的值。在这种情况下，SPI_RIF 寄存器的 TXTH 位会被设置为 1，如果 SPI_IER 寄存器中的 TXTH 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 TXTH 位置 1，会将 SPI_RIF 寄存器的 TXTH 位清零并清除中断。

21.7.6.5 接收FIFO缓存为非空 (RXNE)

下列两种情况将产生 RXNE 的中断事件

- ◆ 当 SPI_STAT.RXNE=1 时，且对 SPI_IER 寄存器中的 RXNE 位置 1。在这种情况下，SPI_RIF 寄存器的 RXNE 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXNE 位置 1，会将 SPI_RIF 寄存器的 RXNE 位清零并清除中断。
- ◆ 当接收的数据加载到 FIFO 缓存中，使接收 FIFO 缓存为非空的时候。在这种情况下，

SPI_RIF 寄存器的 RXNE 位会被设置为 1，如果 SPI_IER 寄存器中的 RXNE 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXNE 位置 1，会将 SPI_RIF 寄存器的 RXNE 位清零并清除中断。

21.7.6.6 接收FIFO缓存为满 (RXF)

下列两种情况将产生 RXF 的中断事件

- ◆ 当 SPI_STAT.RXF=1 时，且对 SPI_IER 寄存器中的 RXF 位置 1。在这种情况下，SPI_RIF 寄存器的 RXF 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXF 位置 1，会将 SPI_RIF 寄存器的 RXF 位清零并清除中断。
- ◆ 当接收的数据加载到 FIFO 缓存中，使接收 FIFO 缓存为满的时候。在这种情况下，SPI_RIF 寄存器的 RXF 位会被设置为 1，如果 SPI_IER 寄存器中的 RXF 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXF 位置 1，会将 SPI_RIF 寄存器的 RXF 位清零并清除中断。

21.7.6.7 接收FIFO缓存上溢 (RXOV)

当接收 FIFO 缓存已满时，下一个接收的数据帧不会加载到接收 FIFO 缓存中。在这种情况下，SPI_RIF 寄存器的 RXOV 位会被设置为 1，如果 SPI_IER 寄存器中的 RXOV 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXOV 位置 1，会将 SPI_RIF 寄存器的 RXOV 位清零并清除中断。

21.7.6.8 接收FIFO缓存下溢 (RXUD)

当接收 FIFO 缓存为空时，但对 SPI_DATA 寄存器执行读访问。在这种情况下，SPI_RIF 寄存器的 RXUD 位会被设置为 1，如果 SPI_IER 寄存器中的 RXUD 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXUD 位置 1，会将 SPI_RIF 寄存器的 RXUD 位清零并清除中断。

21.7.6.9 接收FIFO缓存阈值 (RXTH)

下列两种情况将产生 RXTH 的中断事件

- ◆ 当 SPI_STAT.RXTH=1 时，且对 SPI_IER 寄存器中的 RXTH 位置 1。在这种情况下，SPI_RIF 寄存器的 RXTH 位会被设置为 1，并产生中断。通过对 SPI_ICR 寄存器中的 RXTH 位置 1，会将 SPI_RIF 寄存器的 RXTH 位清零并清除中断。
- ◆ 当接收的数据帧加载到接收 FIFO 缓存时，使 FIFO 缓存中的有效数据个数等于 SPI_CON2.RXFTH 设置的值。在这种情况下，SPI_RIF 寄存器的 RXTH 位会被设置为 1，如果 SPI_IER 寄存器中的 RXTH 位置 1 则产生中断。通过对 SPI_ICR 寄存器中的 RXTH 位置 1，会将 SPI_RIF 寄存器的 RXTH 位清零并清除中断。

21.7.6.10 帧格式错误 (FRE)

仅当 I2S 配置为从机模式时，才能通过硬件设置该标志。如果外部主机没有按照从机期望的那样改变 WS 信号，则此标志将置 1。如果同步丢失，则需要执行以下步骤以从此状态恢复并使外部主机与 I2S 从机重新同步：

1. 禁用 I2S。
2. 设置 SPI_I2SCFG 寄存器的 I2SE 位为 1。当使能 I2S 后硬件会自动侦测同步，请参见“自动侦测同步”章节的描述。

主机和从机之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步线上存在噪音干扰造成。如果 SPI_IER 寄存器中的 FRE 位置 1，则可以产生帧格式错误中断。通过对 SPI_ICR 寄存器中的 FRE 位置 1，会将 SPI_RIF 寄存器的 FRE 位清零并清除中断。

21.8 特殊功能寄存器

21.8.1 寄存器列表

SPI 寄存器列表			
名称	偏移地址	类型	描述
SPI_CON1	0000 _H	R/W	SPI 控制寄存器 1
SPI_CON2	0004 _H	R/W	SPI 控制寄存器 2
SPI_STAT	0008 _H	R	SPI 状态寄存器
SPI_DATA	000C _H	R/W	SPI 数据寄存器
SPI_CRCPOLY	0010 _H	R/W	SPI CRC 多项式寄存器
SPI_RXCRC	0014 _H	R	SPI RX CRC 寄存器
SPI_TXCRC	0018 _H	R	SPI TX CRC 寄存器
SPI_I2SCFG	001C _H	R/W	SPI I2S 配置寄存器
SPI_I2SPR	0020 _H	R/W	SPI I2S 预分频寄存器
SPI_IER	0024 _H	W1	SPI 中断启用寄存器
SPI_IDR	0028 _H	W1	SPI 中断禁用寄存器
SPI_IVS	002C _H	R	SPI 中断有效状态寄存器
SPI_RIF	0030 _H	R	SPI 原始中断标志状态寄存器
SPI_IFM	0034 _H	R	SPI 中断标志屏蔽状态寄存器
SPI_ICR	0038 _H	C_W1	SPI 中断清除寄存器

21.8.2 寄存器描述

21.8.2.1 SPI控制寄存器 1 (SPI_CON1)

SPI 控制寄存器 1 (SPI_CON1)																															
偏移地址: 0x000																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																BIDEN	BIDOEN	CRCEN	NXTCRC	FLEN	RXO	SSEN	SSOUT	LSBFST	SPIEN	BAUD			MSTREN	CPOL	CPHA

—	Bits 31-16	—	—
BIDEN	Bit 15	R/W	<p>双向通信使能位</p> <p>该位使能使用单线双向数据线实现半双工通信。当选择半双工通信模式时，保持 RXO 位清零。</p> <p>0: 选择全双工通信数据模式 1: 选择半双工通信数据模式</p> <p>注：该位不用于 I2S 模式。</p>
BIDOEN	Bit 14	R/W	<p>双向通信输出使能位</p> <p>此位结合 BIDEN 位，用于选择半双工通信模式下的传输方向</p> <p>0: 禁止输出（只接收模式） 1: 使能输出（只发送模式）</p> <p>注： 1.在主机模式下，使用 MOSI 引脚，在从机模式，使用 MISO 引脚。 2.该位不用于 I2S 模式。</p>
CRCEN	Bit 13	R/W	<p>CRC 硬件计算使能位</p> <p>0: 禁止 CRC 计算 1: 使能 CRC 计算</p> <p>注： 1.为确保正确操作，只应在禁止 SPI (SPIEN = 0) 时对此位执行写操作。 2.该位不用于 I2S 模式。</p>
NXTCRC	Bit 12	R/W	<p>下一次传输 CRC 使能位</p> <p>0: 数据传输结束时不传输 CRC 1: 数据传输结束时传输 CRC</p> <p>注：</p>

			<p>1.SPI 配置为全双工或只发送模式时，在写入最后一个数据前将该位置 1。当 SPI 配置为只接收模式时，必须在读取最后一个数据数据前将该位置 1。</p> <p>2.该位不用于 I2S 模式。</p>
FLEN	Bit 11	R/W	<p>数据帧长度选择位</p> <p>0: 发送/接收选择 8 位数据帧长度</p> <p>1: 发送/接收选择 16 位数据帧长度</p> <p>注:</p> <p>1.为确保正确操作，只应在禁止 SPI (SPIEN = 0) 时对此位执行写操作。</p> <p>2.该位不用于 I2S 模式。</p>
R XO	Bit 10	R/W	<p>只接收使能位</p> <p>此位结合 BIDEN 位，用于选择双线单向模式下的传输方向。此位也适用于多从机模式系统，在此类系统中，关闭未被访问的从机输出，以防止被访问的从机输出被其他从机干扰。</p> <p>0: 全双工（发送和接收）</p> <p>1: 关闭输出（只接收模式）</p> <p>注：该位不用于 I2S 模式。</p>
SSEN	Bit 9	R/W	<p>软件控制从机使能位</p> <p>当 SSEN 位置 1 时，NSS 引脚输入将被 SSOUT 位的值替换。</p> <p>0: 禁止软件控制从机</p> <p>1: 使能软件控制从机</p> <p>注：该位不用于 I2S 模式和 SPI TI 模式。</p>
SSOUT	Bit 8	R/W	<p>软件控制片选引脚输出位</p> <p>0: NSS 引脚输出为 0</p> <p>1: NSS 引脚输出为 1</p> <p>仅当 SSEN 位置 1 时该位才有效。此位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 IO 值。</p> <p>注：该位不用于 I2S 模式和 SPI TI 模式。</p>
LSBFST	Bit 7	R/W	<p>先发最低有效位选择位</p> <p>0: 使用 MSB 发送与接收数据</p> <p>1: 使用 LSB 发送与接收数据</p> <p>注:</p> <p>1.正在通信时不应更改此位。</p> <p>2.该位不用于 I2S 模式和 SPI TI 模式。</p>
SPIEN	Bit 6	R/W	<p>SPI 模块使能位</p>

			<p>0: 关闭 SPI 外设 1: 使能 SPI 外设 注: 该位不用于 I2S 模式。</p>
BAUD	Bit 5-3	R/W	<p>波特率选择位 000: fPCLK/2 001: fPCLK/4 010: fPCLK/8 011: fPCLK/16 100: fPCLK/32 101: fPCLK/64 110: fPCLK/128 111: fPCLK/256 注: 1.正在通信时不应更改此位。 2.该位不用于 I2S 模式。</p>
MSTREN	Bit 2	R/W	<p>主机模式使能位 0: 从机配置 1: 主机配置 注: 1.正在通信时不应更改此位。 2.该位不用于 I2S 模式。</p>
CPOL	Bit 1	R/W	<p>时钟的极性控制位 0: 在空闲的状态下, SCK 引脚保持低电平输出 1: 在空闲的状态下, SCK 引脚保持高电平输出 注: 1.正在通信时不应更改此位。 2.该位不用于 I2S 模式和 SPI TI 模式。</p>
CPHA	Bit 0	R/W	<p>时钟的相位控制位 0: 从第一个时钟边沿开始采样数据 1: 从第二个时钟边沿开始采样数据 注: 1.正在通信时不应更改此位。 2.该位不用于 I2S 模式和 SPI TI 模式。</p>

21.8.2.2 SPI控制寄存器 2 (SPI_CON2)

SPI 控制寄存器 2 (SPI_CON2)																															
偏移地址: 0x004																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RXFTH	TXFTH	Reserved						FRF	NSSP	NSSOE	TXDMA	RXDMA			

—	Bits 31-16	—	—
RXFTH	Bit 15-14	R/W	<p>接收 FIFO 阈值</p> <p>用于选择接收 FIFO 缓存的阈值。</p> <p>00: 接收 FIFO 中有 1 个数据</p> <p>01: 接收 FIFO 中数据达到其深度的 1/4</p> <p>10: 接收 FIFO 中数据达到其深度的 1/2</p> <p>11: 接收 FIFO 中数据再增加 2 个达到其深度</p> <p>注: 如果 RXDMA 使能, 该位也提供接收 FIFO 缓存的 DMA 请求条件。</p>
TXFTH	Bit 13-12	R/W	<p>发送 FIFO 阈值</p> <p>用于选择发送 FIFO 缓存的阈值。</p> <p>00: 发送 FIFO 内无数据</p> <p>01: 发送 FIFO 中数据少于或等于 2 个</p> <p>10: 发送 FIFO 中数据少于或等于其深度的 1/4</p> <p>11: 发送 FIFO 中数据少于或等于其深度的 1/2</p> <p>注: 如果 TXDMA 使能, 该位也提供发送 FIFO 缓存的 DMA 请求条件。</p>
—	Bit 11-5	—	—
FRF	Bit 4	R/W	<p>帧格式选择位</p> <p>0: SPI 摩托罗拉模式</p> <p>1: SPI TI 模式</p> <p>注:</p> <p>1. 只有在禁止 SPI (SPIEN = 0) 时才能写入该位。</p> <p>2. 该位不用于 I2S 模式。</p>
NSSP	Bit 3	R/W	<p>NSS 脉冲管理位</p> <p>该位仅用于主机模式。它允许 SPI 在进行连续传输时在两个连续数据之间产生 NSS 脉冲。在单次数据传输的情况下, 它会在传输后强制 NSS 引脚为高电</p>

			<p>平。如果 CPHA = '1' 或 FRF = '1' 则没有意义。</p> <p>0: 没有 NSS 脉冲</p> <p>1: 产生 NSS 脉冲</p> <p>注:</p> <p>1. 只有在禁止 SPI (SPIEN = 0) 时才能写入该位。</p> <p>2. 该位不用于 I2S 模式和 SPI TI 模式。</p>
NSSOE	Bit 2	R/W	<p>NSS 引脚输出使能位</p> <p>0: 在主机模式下禁止 NSS 输出, 可在多主机模式配置下工作</p> <p>1: 在主机模式下使能 NSS 输出, 不能在多主机模式环境下工作</p> <p>注: 该位不用于 I2S 模式和 SPI TI 模式。</p>
TXDMA	Bit 1	R/W	<p>发送 FIFO 缓存 DMA 使能位</p> <p>该位置 1 时, 只要 TXTH 标志置 1, 就会产生 DMA 请求。</p> <p>0: 关闭发送 FIFO 缓存的 DMA</p> <p>1: 使能发送 FIFO 缓存的 DMA</p>
RXDMA	Bit 0	R/W	<p>接收 FIFO 缓存 DMA 使能位</p> <p>该位置 1 时, 只要 RXTH 标志置 1, 就会产生 DMA 请求。</p> <p>0: 关闭接收 FIFO 缓存的 DMA</p> <p>1: 使能接收 FIFO 缓存的 DMA</p>

21.8.2.3 SPI状态寄存器 (SPI_STAT)

SPI 状态寄存器 (SPI_STAT)																															
偏移地址: 0x008																															
复位值: 0x0000 0011																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			RXFLV				Reserved			TXFLV				BUSY	CHSIDE	Reserved	RXTH	RXUD	RXOV	RXF	RXNE	Reserved				TXTH	TXUD	TXOV	TXF	TXE	

—	Bits 31-29	—	—
RXFLV	Bit 28-24	R	接收 FIFO 缓存数据个数 该位域表明接收 FIFO 缓存的有效数据个数。
—	Bit 23-21	—	—
TXFLV	Bit 20-16	R	发送 FIFO 缓存数据个数 该位域表明发送 FIFO 缓存的有效数据个数。
BUSY	Bit 15	R	忙标志位 0: SPI (或 I2S) 不繁忙 1: SPI (或 I2S) 忙于通信 此标志由硬件置 1 和清零。
CHSIDE	Bit 14	R	通道信息声道标志位 0: 当前正在发送或接收左声道 1: 当前正在发送或接收右声道 注: 该位不用于 SPI 模式, 在 PCM 模式下没有意义。
—	Bit 13	—	—
RXTH	Bit 12	R	接收FIFO缓存个数超出阈值 0: 接收FIFO缓存的有效数据个数少于RXFTH 设置的值 1: 接收FIFO缓存的有效数据个数大于或等于RXFTH设置的值
RXUD	Bit 11	R/C_R	接收FIFO缓存下溢 0: 接收FIFO缓存未发生下溢 1: 接收FIFO 缓存发生下溢 注: 读取STAT寄存器清除此位。
RXOV	Bit 10	R/C_R	接收FIFO缓存上溢 0: 接收FIFO缓存未发生上溢 1: 接收FIFO缓存发生上溢

			注：读取STAT寄存器清除此位。
RXF	Bit 9	R	接收FIFO缓存满 0: 接收FIFO缓存未 1: 接收FIFO缓存已
RXNE	Bit 8	R	接收FIFO缓存非空 0: 接收FIFO缓存为 1: 接收FIFO缓存为
—	Bit 7-5	—	—
TXTH	Bit 4	R	发送FIFO缓存个数低于阈值 0: 发送FIFO缓存的有效数据个数大于TXFTH 设置的值 1: 发送FIFO缓存的有效数据个数少于或等于TXFTH设置的值
TXUD	Bit 3	R/C_R	发送FIFO缓存下溢 0: 发送FIFO缓存未发生下溢 1: 发送FIFO 缓存发生下溢 注：读取STAT寄存器清除此位。
TXOV	Bit 2	R/C_R	发送FIFO缓存上溢 0: 发送FIFO缓存未发生上溢 1: 发送FIFO缓存发生上溢 注：读取STAT寄存器清除此位。
TXF	Bit 1	R	发送FIFO缓存满 0: 发送FIFO缓存未 1: 发送FIFO缓存已
TXE	Bit 0	R	发送FIFO缓存空 0: 发送FIFO缓存非 1: 发送FIFO缓存为

21.8.2.4 SPI数据寄存器 (SPI_DATA)

SPI 数据寄存器 (SPI_DATA)																															
偏移地址: 0x00C																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DATA															

—	Bits 31-16	—	—
DATA	Bits 15-0	R/W	<p>数据寄存器</p> <p>已接收或者要发送的数据。</p> <p>数据寄存器分为2个FIFO缓存，一个用于写入（发送FIFO缓存），一个用于读取（接收FIFO缓存）。</p> <p>对数据寄存器执行写操作时，数据将写入发送FIFO缓存，从数据寄存器执行读取时，将返回接收FIFO缓存中的值。</p> <p>注：数据始终是右对齐的。写入寄存器时忽略未使用的位，读取寄存器时未使用的位数据为0。</p>

21.8.2.5 SPI CRC多项式寄存器 (SPI_CRCPOLY)

SPI CRC 多项式寄存器 (SPI_CRCPOLY)																															
偏移地址: 0x010																															
复位值: 0x0000 0007																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CRCPOLY															

—	Bits 31-16	—	—
CRCPOLY	Bits 15-0	R/W	<p>CRCPOLY</p> <p>此寄存器包含用于CRC计算的多项式。CRC多项式 (0007h) 是此寄存器的复位值。可根据需要配置另一个多项式。</p> <p>注:</p> <ol style="list-style-type: none"> 1. 多项式值应仅为奇数, 不支持偶数。 2. 该位域不用于I2S模式。

21.8.2.6 SPI RX CRC寄存器 (SPI_RXCRC)

SPI RX CRC 寄存器 (SPI_RXCRC)																															
偏移地址: 0x014																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RXCRC															

—	Bits 31-16	—	—
RXCRC	Bits 15-0	R	<p>接收 CRC 值</p> <p>使能 CRC 计算后, RXCRC<15:0>位将包含后续接收字节在计算后所得到的 CRC 值。当 SPI_CON1 寄存器中的 CRCEN 位写入 1 时, 此寄存器复位。CRC 通过 SPI_CRCPOLY 寄存器中设置的多项式连续计算。数据帧长度设置为 8 位数据 (SPI_CON1 寄存器的 FLEN 位清零) 时, 仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。选择 16 位数据帧长度 (SPI_CON1 寄存器的 FLEN 位置 1) 时, 考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。</p> <p>注:</p> <ol style="list-style-type: none"> 1.当 BUSY 标志置 1 时, 读取此寄存器可能返回一个不正确的值。 2.该位域不用于 I2S 模式。

21.8.2.7 SPI TX CRC寄存器 (SPI_TXCRC)

SPI TX CRC 寄存器 (SPI_TXCRC)																															
偏移地址: 0x018																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																TXCRC															

—	Bits 31-16	—	—
TXCRC	Bits 15-0	R	<p>发送 CRC 值</p> <p>使能 CRC 计算后, TXCRC<15:0> 位将包含后续发送字节在计算后所得到的 CRC 值。当 SPI_CON1 寄存器中的 CRCEN 位写入 1 时, 此寄存器复位。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。数据帧长度设置为 8 位数据 (SPI_CON1 寄存器的 FLEN 位清零) 时, 仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。选择 16 位数据帧长度 (SPI_CON1 寄存器的 FLEN 位置 1) 时, 考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。</p> <p>注:</p> <ol style="list-style-type: none"> 1.当 BUSY 标志置 1 时, 读取此寄存器可能返回一个不正确的值。 2.该位域不用于 I2S 模式。

21.8.2.8 SPI I2S配置寄存器 (SPI_I2SCFG)

SPI I2S 配置寄存器 (SPI_I2SCFG)																															
偏移地址: 0x01C																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												I2SMOD	I2SE	I2SCFG			PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN	CHLEN								

—	Bits 31-13	—	—
I2SMOD	Bit 12	R/W	I2S 模式选择位 0: 选择 SPI 模式 1: 选择 I2S 模式 注: 应在 SPI 或 I2S 禁止时配置此位。
I2SE	Bit 11	R/W	I2S 模块使能位 0: 关闭 I2S 外设 1: 使能 I2S 外设 注: 该位不用于 SPI 模式。
I2SCFG	Bit 10-8	R/W	I2S 配置模式选择位 000: 从机 - 全双工 (发送和接收) 001: 从机 - 发送 010: 从机 - 接收 100: 主机 - 全双工 (发送和接收) 101: 主机 - 发送 110: 主机 - 接收 1. 为确保正确操作, 只应在禁止 I2S (I2SE = 0) 时对此位域执行写操作。 2. 该位域不用于 SPI 模式。
PCMSYNC	Bit 7	R/W	PCM 帧同步选择位 0: 短帧同步 1: 长帧同步 注: 仅当 I2SSTD = 11 (使用 PCM 标准) 时, 该位才有意义, 不用于 SPI 模式。
—	Bit 6	—	—
I2SSTD	Bit 5-4	R/W	I2S 标准选择位 00: I2S Philips 标准 01: MSB 对齐标准 (左对齐)

			<p>10: LSB 对齐标准 (右对齐)</p> <p>11: PCM 标准</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置这些位域, 不用于 SPI 模式。</p>
CKPOL	Bit 3	R/W	<p>空闲状态的时钟电平选择位</p> <p>0: I2S 时钟在空闲状态时钟为低电平</p> <p>1: I2S 时钟在空闲状态时钟为高电平</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位, 不用于 SPI 模式。</p>
DATLEN	Bit 2-1	R/W	<p>传输的数据长度选择位</p> <p>00: 16 位数据长度</p> <p>01: 24 位数据长度</p> <p>10: 32 位数据长度</p> <p>11: 不允许</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置这些位域, 不用于 SPI 模式。</p>
CHLEN	Bit 0	R/W	<p>信道长度 (每个音频通道的位数) 选择位</p> <p>0: 16 位宽</p> <p>1: 32 位宽</p> <p>只有在 DATLEN 为 00 时, 此位的写操作才有意义, 否则无论填入何值, 通道长度始终由硬件固定为 32 位。</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位, 不用于 SPI 模式。</p>

21.8.2.9 SPI I2S预分频寄存器 (SPI_I2SPR)

SPI I2S 预分频寄存器 (SPI_I2SPR)																															
偏移地址: 0x020																															
复位值: 0x0000 0002																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EXTCKEN		MCKOE	ODD	I2SDIV																	

—	Bits 31-11	—	—
EXTCKEN	Bit 10	R/W	<p>I2S_CLK 时钟选择位</p> <p>0: 选择 APB 时钟</p> <p>1: 选择 I2SCLK 时钟</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位, 不用于 SPI 模式。</p>
MCKOE	Bit 9	R/W	<p>主时钟输出使能位</p> <p>0: 禁用主时钟输出</p> <p>1: 使能主时钟输出</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位。仅在 I2S 处于主机模式时使用, 不用于 SPI 模式。</p>
ODD	Bit 8	R/W	<p>预分频器的奇数因子</p> <p>0: 实际分频器值= I2SDIV * 2</p> <p>1: 实际分频器值= (I2SDIV * 2) + 1</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位。仅在 I2S 处于主机模式时使用, 不用于 SPI 模式。</p>
I2SDIV	Bit 7-0	R/W	<p>I2S 线性预分频器</p> <p>0x00: 禁用预分频</p> <p>0x01~0xFF: 预分频值</p> <p>注: 为了正确操作, 应在禁用 I2S 时配置该位域。仅在 I2S 处于主机模式时使用, 不用于 SPI 模式。</p>

21.8.2.10 SPI中断使能寄存器 (SPI_IER)

SPI 中断启用寄存器 (SPI_IER)																																	
偏移地址: 0x024																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved													FRE	MODF	CRCERR	Reserved					RXTH	RXUD	RXOV	RXF	RXNE	Reserved			TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	W1	帧格式错误中断使能位 0: 写入 0 无效 1: 使能帧格式错误中断
MODF	Bit 17	W1	模式故障中断使能位 0: 写入 0 无效 1: 使能模式故障中断
CRCERR	Bit 16	W1	CRC 错误中断使能位 0: 写入 0 无效 1: 使能 CRC 错误中断
—	Bit 15-13	—	—
RXTH	Bit 12	W1	接收 FIFO 缓存超过阈值中断使能位 0: 写入 0 无效 1: 使能接收 FIFO 缓存超过阈值中断
RXUD	Bit 11	W1	接收 FIFO 缓存下溢中断使能位 0: 写入 0 无效 1: 使能接收 FIFO 缓存下溢中断
RXOV	Bit 10	W1	接收 FIFO 缓存上溢中断使能位 0: 写入 0 无效 1: 使能接收 FIFO 缓存上溢中断
RXF	Bit 9	W1	接收 FIFO 缓存满中断使能位 0: 写入 0 无效 1: 使能接收 FIFO 缓存满中断
RXNE	Bit 8	W1	接收 FIFO 缓存非空中断使能位 0: 写入 0 无效 1: 使能接收 FIFO 缓存非空中断
—	Bit 7-5	—	—
TXTH	Bit 4	W1	发送 FIFO 缓存低于阈值中断使能位

			0: 写入 0 无效 1: 使能发送 FIFO 缓存低于阈值中断
TXUD	Bit 3	W1	发送 FIFO 缓存下溢中断使能位 0: 写入 0 无效 1: 使能发送 FIFO 缓存下溢中断
TXOV	Bit 2	W1	发送 FIFO 缓存上溢中断使能位 0: 写入 0 无效 1: 使能发送 FIFO 缓存上溢中断
—	Bit 1	—	—
TXE	Bit 0	W1	发送 FIFO 缓存空中断使能位 0: 写入 0 无效 1: 使能发送 FIFO 缓存空中断

21.8.2.11 SPI中断禁用寄存器 (SPI_IDR)

SPI 中断禁用寄存器 (SPI_IDR)																																	
偏移地址: 0x028																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved													FRE	MODF	CRCERR	Reserved					RXTH	RXUD	RXOV	RXF	RXNE	Reserved			TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	W1	帧格式错误中断禁止位 0: 写入0无效 1: 禁止帧格式错误中断
MODF	Bit 17	W1	模式故障中断禁止位 0: 写入 0 无效 1: 禁止模式故障中断
CRCERR	Bit 16	W1	CRC 错误中断禁止位 0: 写入 0 无效 1: 禁止 CRC 错误中断
—	Bit 15-13	—	—
RXTH	Bit 12	W1	接收 FIFO 缓存超过阈值中断禁止位 0: 写入 0 无效 1: 禁止接收 FIFO 缓存超过阈值中断
RXUD	Bit 11	W1	接收 FIFO 缓存下溢中断禁止位 0: 写入 0 无效 1: 禁止接收 FIFO 缓存下溢中断
RXOV	Bit 10	W1	接收 FIFO 缓存上溢中断禁止位 0: 写入 0 无效 1: 禁止接收 FIFO 缓存上溢中断
RXF	Bit 9	W1	接收 FIFO 缓存满中断禁止位 0: 写入 0 无效 1: 禁止接收 FIFO 缓存满中断
RXNE	Bit 8	W1	接收FIFO缓存非空中断禁止位 0: 写入0无效 1: 禁止接收FIFO缓存非空中断
—	Bit 7-5	—	—
TXTH	Bit 4	W1	发送 FIFO 缓存低于阈值中断禁止位

			0: 写入 0 无效 1: 禁止发送 FIFO 缓存低于阈值中断
TXUD	Bit 3	W1	发送FIFO缓存下溢中断禁止位 0: 写入0无效 1: 禁止发送FIFO缓存下溢中断
TXOV	Bit 2	W1	发送 FIFO 缓存上溢中断禁止位 0: 写入 0 无效 1: 禁止发送 FIFO 缓存上溢中断
—	Bit 1	—	—
TXE	Bit 0	W1	发送 FIFO 缓存空中断禁止位 0: 写入 0 无效 1: 禁止发送 FIFO 缓存空中断

21.8.2.12 SPI中断有效状态寄存器 (SPI_IVS)

SPI 中断有效状态寄存器 (SPI_IVS)																																		
偏移地址: 0x02C																																		
复位值: 0x0000 0000																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reserved														FRE	MODF	CRCERR	Reserved				RXTH	RXUD	RXOV	RXF	RXNE	Reserved				TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	R	帧格式错误中断使能状态 0: 禁止帧格式错误中断 1: 使能帧格式错误中断
MODF	Bit 17	R	模式故障中断使能状态 0: 禁止模式故障中断 1: 使能模式故障中断
CRCERR	Bit 16	R	CRC错误中断使能状态 0: 禁止CRC错误中断 1: 使能CRC错误中断
—	Bit 15-13	—	—
RXTH	Bit 12	R	接收 FIFO 缓存超过阈值中断使能状态 0: 禁止接收 FIFO 缓存超过阈值中断 1: 使能接收 FIFO 缓存超过阈值中断
RXUD	Bit 11	R	接收 FIFO 缓存下溢中断使能状态 0: 禁止接收 FIFO 缓存下溢中断 1: 使能接收 FIFO 缓存下溢中断
RXOV	Bit 10	R	接收 FIFO 缓存上溢中断使能状态 0: 禁止接收 FIFO 缓存上溢中断 1: 使能接收 FIFO 缓存上溢中断
RXF	Bit 9	R	接收 FIFO 缓存满中断使能状态 0: 禁止接收 FIFO 缓存满中断 1: 使能接收 FIFO 缓存满中断
RXNE	Bit 8	R	接收 FIFO 缓存非空中断使能状态 0: 禁止接收 FIFO 缓存非空中断 1: 使能接收 FIFO 缓存非空中断
—	Bit 7-5	—	—
TXTH	Bit 4	R	发送 FIFO 缓存低于阈值中断使能状态

			0: 禁止发送 FIFO 缓存低于阈值中断 1: 使能发送 FIFO 缓存低于阈值中断
TXUD	Bit 3	R	发送 FIFO 缓存下溢中断使能状态 0: 禁止发送 FIFO 缓存下溢中断 1: 使能发送 FIFO 缓存下溢中断
TXOV	Bit 2	R	发送FIFO缓存上溢中断使能状态 0: 禁止发送FIFO缓存上溢中断 1: 使能发送FIFO缓存上溢中断
—	Bit 1	—	—
TXE	Bit 0	R	发送 FIFO 缓存空中断使能状态 0: 禁止发送 FIFO 缓存空中断 1: 使能发送 FIFO 缓存空中断

21.8.2.13 SPI原始中断标志状态寄存器 (SPI_RIF)

SPI 原始中断标志状态寄存器 (SPI_RIF)																																	
偏移地址: 0x030																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved													FRE	MODF	CRCERR	Reserved					RXTH	RXUD	RXOV	RXF	RXNE	Reserved			TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	R	帧格式错误中断事件标志位 0: 未发生中断事件 1: 发生中断事件
MODF	Bit 17	R	模式故障中断事件标志位 0: 未发生中断事件 1: 发生中断事件
CRCERR	Bit 16	R	CRC错误中断事件标志位 0: 未发生中断事件 1: 发生中断事件
—	Bit 15-13	—	—
RXTH	Bit 12	R	接收 FIFO 缓存超过阈值中断事件标志位 0: 未发生中断事件 1: 发生中断事件
RXUD	Bit 11	R	接收 FIFO 缓存下溢中断事件标志位 0: 未发生中断事件 1: 发生中断事件
RXOV	Bit 10	R	接收 FIFO 缓存上溢中断事件标志位 0: 未发生中断事件 1: 发生中断事件
RXF	Bit 9	R	接收 FIFO 缓存满中断事件标志位 0: 未发生中断事件 1: 发生中断事件
RXNE	Bit 8	R	接收 FIFO 缓存非空中断事件标志位 0: 未发生中断事件 1: 发生中断事件
—	Bit 7-5	—	—
TXTH	Bit 4	R	发送 FIFO 缓存低于阈值中断事件标志位

			0: 未发生中断事件 1: 发生中断事件
TXUD	Bit 3	R	发送 FIFO 缓存下溢中断事件标志位 0: 未发生中断事件 1: 发生中断事件
TXOV	Bit 2	R	发送FIFO缓存上溢中断事件标志位 0: 未发生中断事件 1: 发生中断事件
—	Bit 1	—	—
TXE	Bit 0	R	发送 FIFO 缓存空中断事件标志位 0: 未发生中断事件 1: 发生中断事件

21.8.2.14 SPI中断标志屏蔽状态寄存器 (SPI_IFM)

SPI 中断标志屏蔽状态寄存器 (SPI_IFM)																																	
偏移地址: 0x034																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved													FRE	MODF	CRCERR	Reserved					RXTH	RXUD	RXOV	RXF	RXNE	Reserved			TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	R	帧格式错误中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
MODF	Bit 17	R	模式故障中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
CRCERR	Bit 16	R	CRC错误中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 15-13	—	—
RXTH	Bit 12	R	接收 FIFO 缓存超过阈值中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
RXUD	Bit 11	R	接收 FIFO 缓存下溢中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
RXOV	Bit 10	R	接收 FIFO 缓存上溢中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
RXF	Bit 9	R	接收 FIFO 缓存满中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
RXNE	Bit 8	R	接收 FIFO 缓存非空中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 7-5	—	—
TXTH	Bit 4	R	发送 FIFO 缓存低于阈值中断屏蔽标志位

			0: 未发生中断事件或中断未使能 1: 产生中断
TXUD	Bit 3	R	发送 FIFO 缓存下溢中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
TXOV	Bit 2	R	发送FIFO缓存上溢中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 1	—	—
TXE	Bit 0	R	发送 FIFO 缓存空中断屏蔽标志位 0: 未发生中断事件或中断未使能 1: 产生中断

21.8.2.15 SPI中断清除寄存器 (SPI_ICR)

SPI 中断清除寄存器 (SPI_ICR)																																	
偏移地址: 0x038																																	
复位值: 0x0000 0000																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved													FRE	MODF	CRCERR	Reserved					RXTH	RXUD	RXOV	RXF	RXNE	Reserved			TXTH	TXUD	TXOV	Reserved	TXE

—	Bits 31-19	—	—
FRE	Bit 18	C_W1	帧格式错误中断清除 0: 写入0无效 1: 清除中断事件与中断
MODF	Bit 17	C_W1	模式故障中断清除 0: 写入 0 无效 1: 清除中断事件与中断
CRCERR	Bit 16	C_W1	CRC 错误中断清除 0: 写入0无效 1: 清除中断事件与中断
—	Bit 15-13	—	—
RXTH	Bit 12	C_W1	接收 FIFO 缓存超过阈值中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RXUD	Bit 11	C_W1	接收 FIFO 缓存下溢中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RXOV	Bit 10	C_W1	接收 FIFO 缓存上溢中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RXF	Bit 9	C_W1	接收 FIFO 缓存满中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RXNE	Bit 8	C_W1	接收 FIFO 缓存非空中断清除 0: 写入 0 无效 1: 清除中断事件与中断
—	Bit 7-5	—	—
TXTH	Bit 4	C_W1	发送 FIFO 缓存低于阈值中断清除

			0: 写入 0 无效 1: 清除中断事件与中断
TXUD	Bit 3	C_W1	发送 FIFO 缓存下溢中断清除 0: 写入 0 无效 1: 清除中断事件与中断
TXOV	Bit 2	C_W1	发送 FIFO 缓存上溢中断清除 0: 写入 0 无效 1: 清除中断事件与中断
—	Bit 1	—	—
TXE	Bit 0	C_W1	发送 FIFO 缓存空中断清除 0: 写入 0 无效 1: 清除中断事件与中断

第22章 通用异步收发器 (UART0~4)

22.1 概述

通用异步收发器 (UART) 提供了一个灵活的方式, 使 MCU 可以与外部设备通过工业标准 NRZ 的形式实现全双工异步串行数据通信。UART 可以使用小数波特率产生器, 提供了超宽的波特率设置范围。

UART 支持异步通信模式和半双工单线通信, 也支持 LIN (本地互连网络)、智能卡协议、IrDA (红外数据协会) SIR ENDEC 规范和 modem 流控操作 (CTS/RTS), 同时还支持多机通信方式。

支持使用 DMA 实现多缓冲区设置, 从而能够支持高速数据通信。

芯片内部共有 5 个通用异步收发器 (UART0~4), 其中 UART0~1 为增强型 EUART (分别为 EUART0~1), UART2~4 为普通型 CUART (分别为 CUART0~2)。EUART 所支持的增强功能包括: 智能卡模式、单线半双工模式、IrDA SIR 模式和 LIN 模式。

22.2 特性

- ◆ 全双工异步通信
- ◆ 兼容 16C550 标准
 - 可设置的通信波特率
- ◆ 支持自动波特率检测
- ◆ 十五个中断源
- ◆ 支持 DMA 使用
 - 利用 DMA 功能将收/发字节缓冲到保留的 SRAM 空间
- ◆ 内置小数波特率发生器, 覆盖范围广
 - 在时钟频率为 48 MHz 时, 可设置收发波特率高达 3 MBps, 最低可达 732.4 Bps
 - 在时钟频率为 4 MHz 时, 可设置收发波特率高达 250 KBps, 最低可达 61 Bps
- ◆ 支持硬件自动流量控制功能 (CTS、RTS), 可设置 RTS 控制触发点
 - Modem 硬件自动控制
 - RS485 发送使能控制
- ◆ 支持 CTS 唤醒功能
- ◆ 支持 IrDA SIR 模式 (仅 EUART)
 - 支持 3/16 位宽功能
- ◆ 支持 RS-485

- 支持 9-位模式
- 多处理器通信
- ◆ 可设置的串行接口特性
 - 可设置数据位个数，即 5、6、7、8、9 位，9 位用于 RS485 模式
 - 校验位，奇、偶、无校验
 - 停止位长度可设置：1，2 位，在智能卡模式中支持 0.5，1.5 位
 - 支持设置高位（MSB）在前或低位（LSB）在前
- ◆ 单线半双工通讯（仅 EUART）
- ◆ 可配置交换 TX/RX 引脚
- ◆ LIN 主机的断路信号发送功能和 LIN 从机的断路信号检测功能（仅 EUART）
 - UART 设置为 LIN 模式时，有 13 位的断路信号产生器与断路信号检测功能
- ◆ 智能卡模式（仅 EUART）
 - 支持 ISO/IEC7816-3 标准定义的 T=0 和 T=1 智能卡异步协议
 - 智能卡使用的 1.5 停止位长度
- ◆ 支持 ModBus 通讯
 - CR/LF 字符识别
 - 超时检测功能
- ◆ 噪声侦测

22.3 结构图

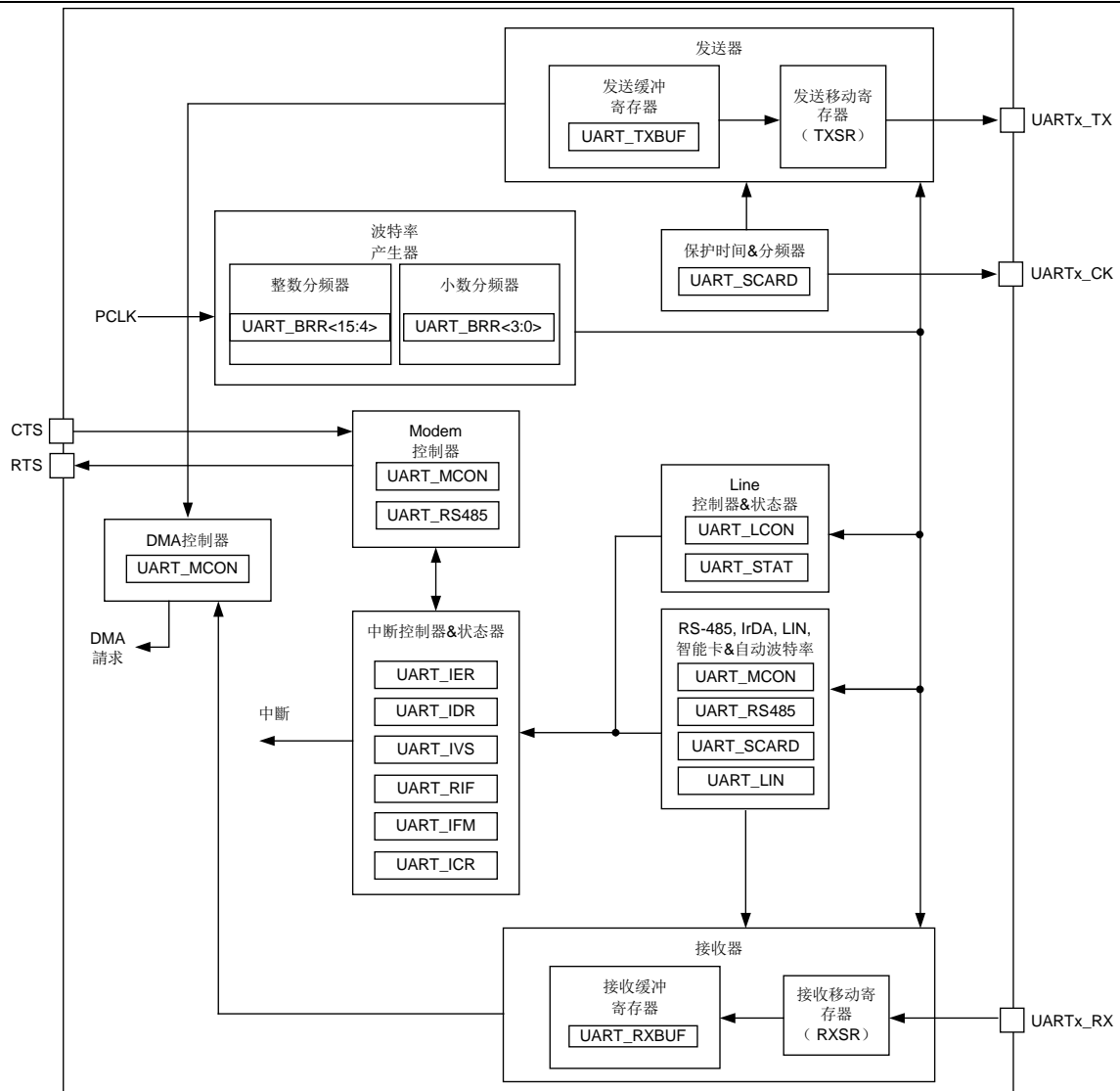


图 22-1 UART 框图

22.4 功能描述

任何 UART 双向通讯要求最少有两个引脚：接收数据输入（RX）和发送数据输出（TX）

- ◆ RX：接收数据输入，是串行数据的输入串口。使用过采样方式完成数据恢复，以区别输入数据和噪声。
- ◆ TX：数据发送输出。当发送器被关闭，引脚回到其 I/O 串口配置状态。当发送器开启，但不发送数据时，TX 脚为高电平输出。在单线半双工通信和智能卡模式中，这个串口既用于发送数据也用于接收数据。

通过这些引脚，串行数据用数据帧的形式发送和接收：

- ◆ 在发送和接收之前为空闲状态
- ◆ 起始位
- ◆ 数据可通过设置 UART_LCON 寄存器的 MSB 位，选择高位还是低位在前
- ◆ 1 或 2 个停止位表明帧的结束（0.5 或 1.5 个停止位用于智能卡模式）
- ◆ 采用小数波特率产生器，整数 12 位与小数 4 位
- ◆ 一个状态寄存器（UART_STAT）
- ◆ 独立的接收和发送数据寄存器（UART_RXBUF, UART_TXBUF）
- ◆ 一个波特率寄存器（UART_BRR）-12 位整数和 4 位小数。
- ◆ 一个智能卡寄存器（UART_SCARD）用于智能卡模式。
- ◆ 一个接收时间寄存器（UART_RTOR）侦测输入信号时间并产生中断

下面的引脚在智能卡模式中会用到：

CK：时钟输出。智能卡模式中，CK 引脚会向智能卡提供时钟。

下列引脚用于支持硬件流控制模式：

CTS：低发送，当高电平时作为发送阻止信号。

RTS：请求发送，表明 UART 已经准备好接收数据（低的时候）。

下列引脚用于 RS485 驱动开启控制：

DE：驱动开启将开启外部收发器的发送模式。

注：DE 和 RTS 共享同一个外部引脚。

22.4.1 具体功能配置

UART 特性	UART0/1 (EUART0/1)	UART2/3/4 (CUART0/1/2)
Modem 的硬件控制	v	v
使用 DMA 实现连续通信	v	v
多机通信模式	v	v
智能卡模式	v	-
单线半双工模式	v	-
IrDA SIR 模块	v	-
LIN 模式	v	-
超时检测功能	v	v
Modbus 通信	v	v
自动波特率检测模式	v	v
RS485 的驱动开启信号	v	v
UART 数据宽度	5、6、7、8、9 Bits	

表 22-1 UART1~4 具体功能配置

注：v 表示支持，支持 RS485 9bit 模式

22.4.2 功能描述

配置 UART_LCON 寄存器中的 DLS 位可选择 5、6、7、8 位字长。

默认设置中，发送和接收的起始位都是低电平，而停止位都是高电平。

TX/RX 引脚电平逻辑可以通过 UART_LCON 寄存器的 TXINV 与 RXINV 位设置为反向。

- ◆ 8 位字符宽度：DLS[1:0]=00
- ◆ 7 位字符宽度：DLS[1:0]=01
- ◆ 6 位字符宽度：DLS[1:0]=10
- ◆ 5 位字符宽度：DLS[1:0]=11

注：第 9 位使用于 RS485 多机通信模式

空闲符号被视为完全由'1' 组成的完整的数据帧，后面跟着包含了数据的下一帧的开始位（'1' 的位数也包括了停止位的位数）。

断路符号被视为在一个帧周期内全部收到'0'（包括停止位期间，也是'0'）。在断路帧结束时，发送器会再插入 2 个停止位。

发送和接收由一个共享的波特率产生器驱动，当发送器和接收器的开启位分别设置为 1 时，分别为其产生时钟。

下面是每个模块的详细说明。

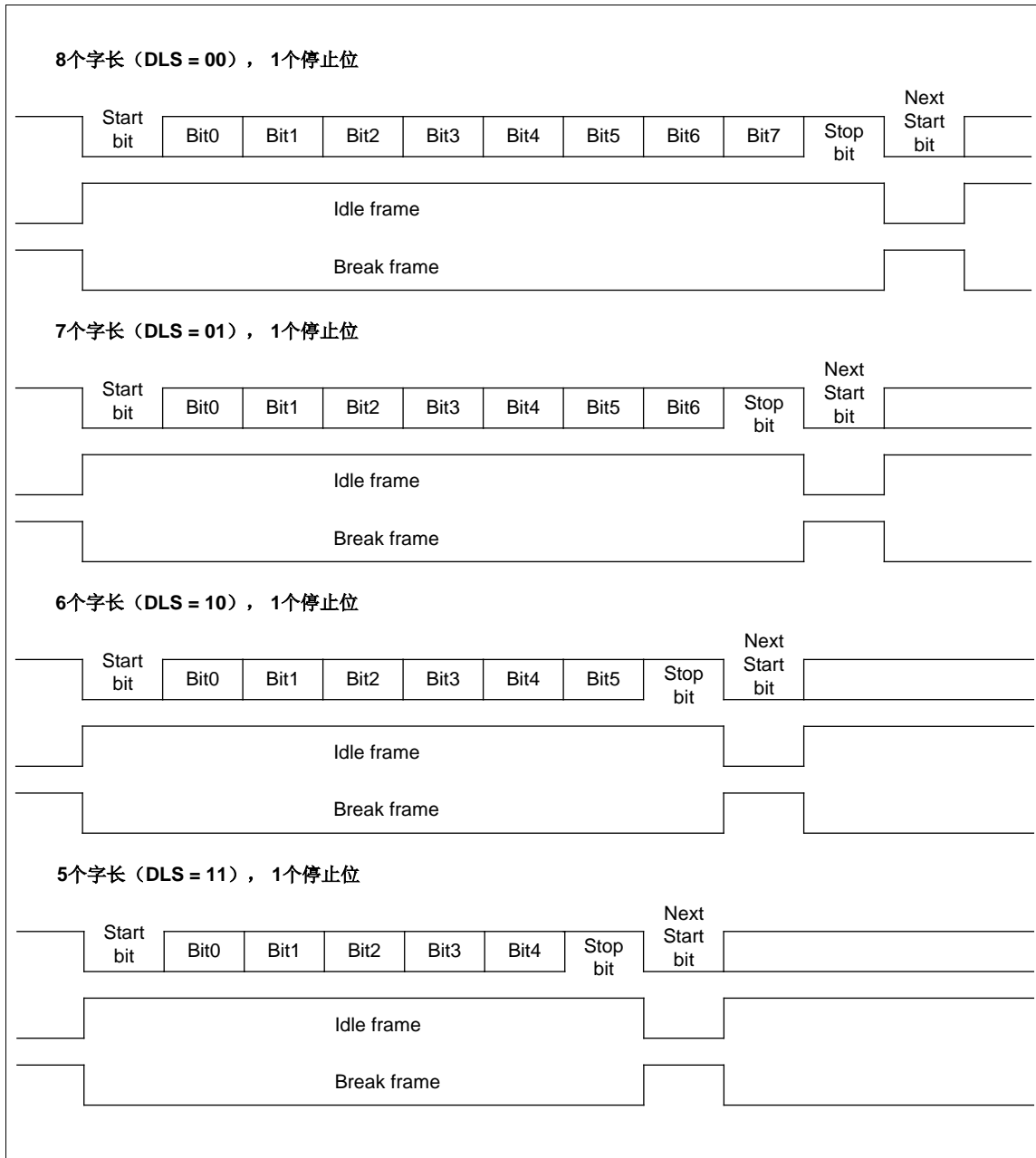


图 22-2 数据宽度设置

22.4.3 发送器

发送器根据 UART_LCON 寄存器的 DLS 位选择发送 5、6、7、8 位的数据，当写入 UART_TXBUF 寄存器，数据将存入移位寄存器中并将数据在 TX 引脚上输出。

在 UART 发送期间，在 TX 引脚上首先发送数据为最低有效位，UART_TXBUF 寄存器充当了一个内部总线和发送移位寄存器之间的缓冲器 (TXSR)。

在发送每个数据字节之前有一个低电平的起始位，在数据字节发送结束后发送停止位，设置 UART_LCON 寄存器的 STOP 位选择停止位数。

UART 支持多种停止位的选择: 0.5, 1, 1.5 和 2 个停止位(智能卡模式支持 0.5 与 1.5 停止位)。

- ◆ 0.5 个停止位：在智能卡模式中发送和接收数据时使用。
- ◆ 1 个停止位：停止位的位数默认值。
- ◆ 1.5 个停止位：在智能卡模式中发送和接收数据时使用。
- ◆ 2 个停止位：可用于普通 UART 模式、以及调制解调器模式。

注 1: 在写入 UART_TXBUF 寄存器数据前必须先等待 UART_STAT 寄存器中的 TFEMPTY 位为 1。

注 2: 开启 TX 开关后，数据才能在 TX 引脚上输出。

空闲帧包括了停止位。

断路帧可通过 UART_MCON 寄存器的 BKREQ 位产生 10 位低电平（当 DLS=00 时），9 位低电平（当 DLS=01 时），8 位低电平（当 DLS=10 时）或者 7 位低电平（当 DLS=11 时），后面跟 2 个停止位。

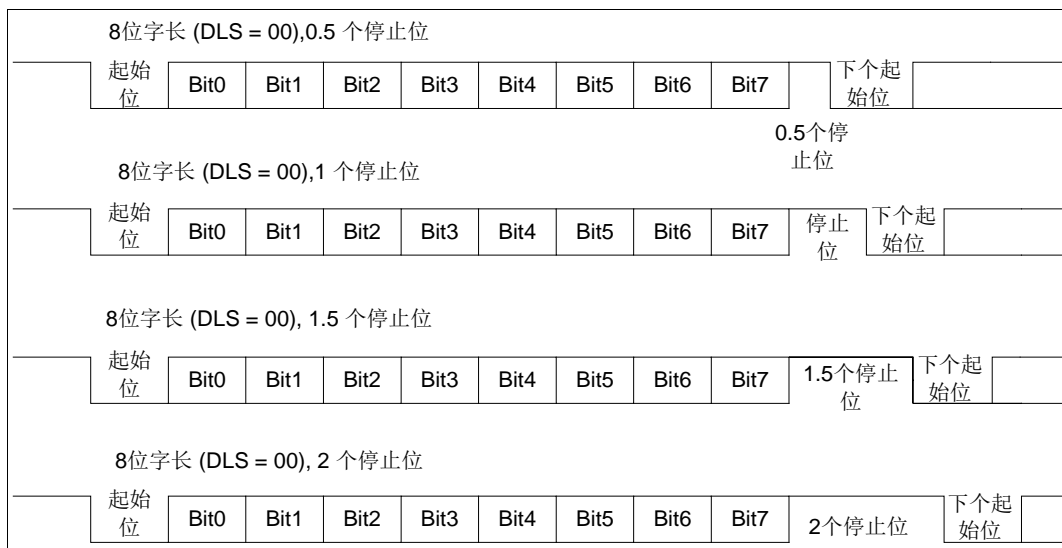


图 22-3 停止位配置

UART 发送配置步骤:

1. 设置 UART_LCON 寄存器中的 DLS 位配置字长。
2. 设置 UART_LCON 寄存器中的 STOP 位配置停止位的位数。
3. 设置 UART_LCON 寄存器中的 PE 与 PS 位配置校验控制开关与极性。
4. 设置 UART_BRR 寄存器选择波特率。
5. 如果采用多缓冲器通信, 设置 UART_MCON 寄存器中的 TXDMAEN 位为 1。按多缓冲器通信中的描述设置 DMA 寄存器。
6. 设置 UART_LCON 寄存器中的 TXEN 位, 开启发送器。
7. 把要发送的数据写进 UART_TXBUF 寄存器 (此动作将清除 UART_STAT 寄存器中的 TFEMPTY 位)。
8. 在 UART_TXBUF 寄存器中写入数据时, 要等待 UART_STAT 寄存器中的 TFEMPTY 位为 1。当关闭 UART 时, 需要先确认传输结束 (UART_STAT 寄存器中的 TSBUSY 位为 0), 避免破坏最后一次传输。

注: 当 UART_LCON 寄存器的 TXEN 与 RXEN 位为 1 时, 无法写入 UART_LCON 与 UART_BRR 寄存器。

22.4.4 接收器

22.4.4.1 防抖电路

在 UART_RX 引脚上配置了一个防抖动电路, 设置 UART_LCON 寄存器中的 DBCEN 位可开启防抖功能, 输入信号须维持至少 8 个时钟周期的高或低电平, 才能使得信号反应至 UART 中, 反之则被忽略, 如下叙述。

在下图中, SYNC0 是指输入信号由系统时钟一次采样; SYNC1 是指 SYNC0 被系统时钟一次采样; SYNC2 表示 SYNC1 由系统时钟一次采样。SYNC0、SYNC1 和 SYNC2 可以表示成 $x[n] \times Ts$, $x[n+1] \times Ts$ 和 $x[n+2] \times Ts$ 。Ts 是系统时钟周期, n 是采样时间。如果关闭防抖动模块, 时间就可以表示为 $x[n+1] \times Ts$ 。

如果开启防抖动模块, SYNC2 信号将进入去抖电路, 那么它将被采样与计数 (可设置样本频率和计数时间值)。时间可表示为 $[(SPT+1) \times (FILTCNT+1)] \times Ts$ 。SPT 是采样频率值, FILTCNT 是计数次数。当 SYNC1 和 SYNC2 不相等时, 计数值将被清零。如果计数值溢满 FILTCNT 寄存器, 防抖动模块将输出信号。

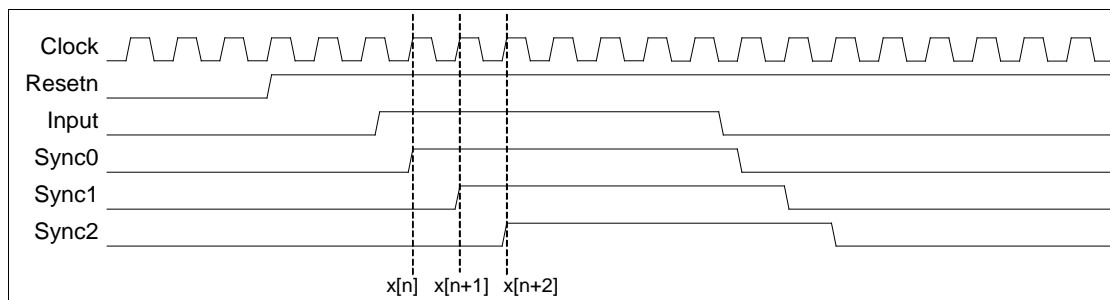


图 22-4 输入防抖动波形

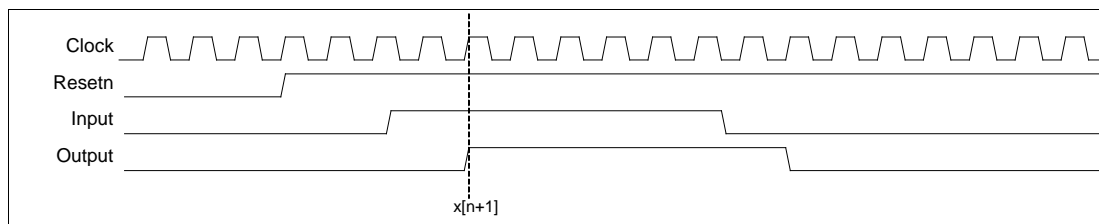


图 22-5 防抖动电路模块输出

22.4.4.2 起始位侦测

接收器根据 UART_LCON 寄存器中 DLS 位的状态接收 5、6、7、8 位的数据字。

起始位侦测在 UART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。

该序列为：1 1 1 0 X 0 X 0 X 0 0 0 X X X X X X X

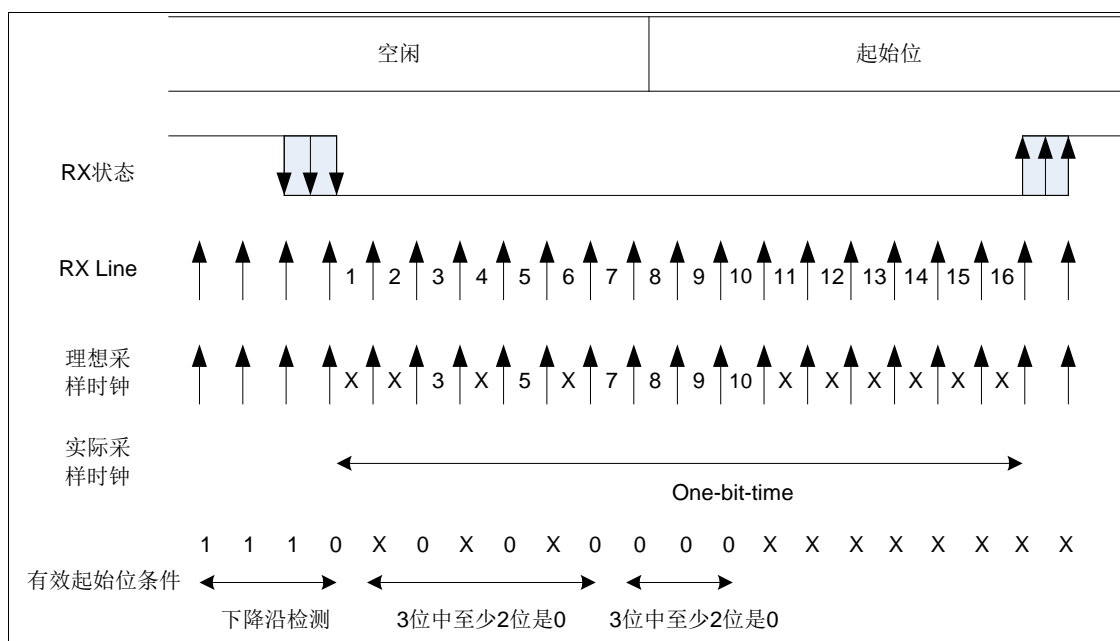


图 22-6 起始位侦测

注 1：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态（不设置标志位）开始等待下降沿。

注 2：如果 3 个采样点都为 '0'（在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为 '0'），则确认收到起始位。

注 3：如果两次 3 个采样点有 2 个是 '0'（第 3、5、7 位的采样点和第 8、9、10 位的采样点），那么起始位仍然是有效的。如果不能满足这个条件，则中止起始位的侦测过程，接收器会回到空闲状态。

注 4：如果两次 3 个采样点有 2 个是 '1'（第 3、5、7 位的采样点和第 8、9、10 位的采样点），那么起始位是无效的，将退出起始位侦测并回到空闲状态，并会设置噪声标志位。

UART 接收配置步骤:

1. 设置 UART_LCON 寄存器中的 DLS 位配置字长。
2. 设置 UART_LCON 寄存器中的 STOP 位配置停止位的位数。
3. 设置 UART_LCON 寄存器中的 PE 与 PS 位配置校验控制开关与极性。
4. 设置 UART_BRR 寄存器选择配置的波特率。
5. 如果采用多缓冲器通信, 设置 UART_MCON 寄存器中的 RXDMAEN 位为 1。根据多缓冲器通信中的描述设置 DMA 寄存器。
6. 设置 UART_LCON 寄存器中的 RXEN 位, 这将开启接收器, 使它开始寻找起始位。

当一个字节被接收到时, UART_STAT 寄存器的 RFNEMPTY 位被设置为 1。它表明移位寄存器的内容被转移到 RX 中。换句话说, 数据已经被接收并且可以被读出 (包括与之有关的错误标志)。

在数据接收期间如果检测到帧错误, 噪音或溢出错误, 错误标志将被设置为 1。同时 RXBERR 位也会和 RFNEMPTY 位一起被设置为 1。在多缓冲器通信时, RFNEMPTY 在接收 1 个字节后被设置为 1, 并由 DMA 对数据寄存器的读取而清除。

RFNEMPTY 位必须在下一字节接收结束前被清零, 以避免溢出错误。

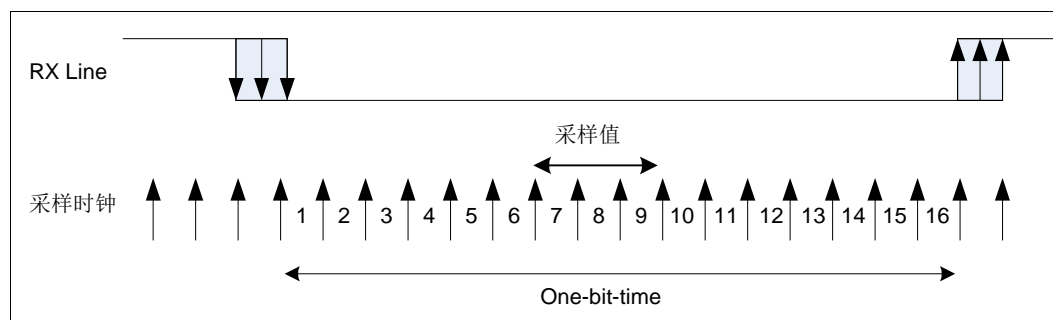


图 22-7 数值采样

帧错误

当以下情况发生时检测到帧错误:

由于没有同步成功或大量噪音的原因, 停止位没有在预期的时间接收和检测出来。

当帧错误被检测到时:

1. FERR 位被硬件置 1
2. 此时读取的 UART_RXBUF 寄存器数据可能有错。
3. 寄存器 UART_STAT 中的 FERR 位显示当前读取的 UART_RXBUF 寄存器是否为帧错误。
4. 寄存器 UART_RIF 中的 RXBERR 位则会在接收的过程中被设置为 1, 若 UART_IER 中的 RXBERR 位为 1 则会产生中断。

奇偶位错误

当以下情况发生时检测到奇偶性错误:

由于没有同步成功或大量噪音的原因, 奇偶位没有在预期的时间接收和检测出来。

当奇偶位错误被检测到时：

1. PERR 位被硬件设置为 1
2. 此时读取的 UART_RXBUF 寄存器数据可能有错。
3. 寄存器 UART_STAT 中的 PERR 位显示当前读取的 UART_RXBUF 寄存器是否为奇偶位错误。
4. 寄存器 UART_RIF 中的 RXBERR 位则会在接收的过程中被设置为 1，若 UART_IER 中的 RXBERR 位为 1 则会产生中断。

断路错误

当以下情况发生时检测到断路错误：

由于没有同步成功或大量噪音的原因，数据与停止位为 0 且没有在预期的时间接收和检测出来。

当断路错误被检测到时：

1. BKERR 位被硬件设置为 1
2. 此时读取的 UART_RXBUF 寄存器数据可能有错。
3. 寄存器 UART_STAT 中的 BKERR 位显示当前读取的 UART_RXBUF 寄存器是否为断路错误。
4. 这个错误并不会产生中断。

溢出错误

当以下情况发生时检测到溢出错误：

RX 还没有被读取，而又接收到一个数据字节，则发生溢出错误。

当溢出错误被检测到时：

1. RFOERR 位被硬件设置为 1
2. UART_RXBUF 内容将不会丢失，读取 UART_RXBUF 寄存器仍能得到先前的数据。
3. 移位缓存器中以前的内容将被覆盖，随后接收的数据将丢失。
4. 寄存器 UART_RIF 中的 RFOERR 位则会在接收的过程中被设置为 1，若 UART_IER 中的 RFOERR 位为 1 则会产生中断。

采样值	采样位数值
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

表 22-2 采样数据的噪音检测数值

22.4.5 状态寄存器

在 UART 中配置了 11 种 UART 状态，描述如下

- ◆ **PERR** (Parity error):
当所接收的数据没有正确的校验位，产生奇偶位错误。
- ◆ **FERR** (Frame error):
当接收到的停止位为 0 时，产生帧错误。
- ◆ **BKERR** (Break error):
当接收到的数据与停止位为 0 时，产生断路错误。
- ◆ **CTSSTA** (CTS status error):
清除发送，此位为显示 CTS 引脚上的输入状态。当 CTSSTA 位为 0，表示调制解调器和数据设备已准备好与 UART 进行数据交换。
- ◆ **RSBUSY** (RX shifter register busy):
当此位为 1 表示接收器正在接收数据，为 0 则为完成接收。
- ◆ **RFNEMPTY** (RXnot empty):
当此位为 1 表示 RX 已接收 1 个数据。
- ◆ **RFOERR** (RX overrun):
当此位为 1 表示 RX 已满，且又再接收 1 个数据字节，此时 RX 数据不会丢失，新接收的数据字节则会被丢失。
- ◆ **RFUERR** (RX underrun):
当此位为 1 表示 RX 内无任何数据，且又被读取。
- ◆ **TSBUSY** (TX shifter register busy):
当此位为 1 表示发送器正在传送数据，为 0 则为传送完成，当写入第一个数据至 UART_TXBUF 寄存器就会使得 TSBUSY 位为 1。
- ◆ **TFEMPTY** (TX empty):
当此位为 1 表示 TX 内无任何数据，为 0 则为准备发送 1 个以上的数据字节。
- ◆ **TFOERR** (TX overrun):
当此位为 1 表示 TX 已满，且又在写入 1 个字节，此时 TX 内的数据不会丢失，新写入的数据字节则会被丢失。

22.4.6 波特率产生器

设置 UART_BRR 寄存器可配置接收器和发送器的波特率。

- ◆ UARTDIV=UART.BRR.
- ◆ TX/RX baud = PCLK / UARTDIV

注 1: 当 UART_LCON 寄存中的 RXEN 与 TXEN 为 1 时, UART_BRR 寄存器无法被写入。

注 2: 当 $BRR[15:4] \geq 0$ 时, UART 无法运行。

根据 UART_BRR 寄存器值可计算出 UART 波特率

- ◆ 在 4 MHz 时, 为了得到 115200 波特
 - $UARTDIV = 4000000/115200 = 34.7$
 - $BRR[15:0] = UARTDIV = 35 = 0x23$ (四舍五入)
- ◆ 在 8 MHz 时, 为了得到 9600 波特
 - $UARTDIV = 8000000/9600 = 833.33$
 - $BRR[15:0] = UARTDIV = 833 = 0x341$ (四舍五入)
- ◆ 在 16 MHz 时, 为了得到 1200 波特
 - $UARTDIV = 16000000/1200 = 13333.33$
 - $BRR[15:0] = UARTDIV = 13333.33 = 0x3415$ (四舍五入)
- ◆ 在 24 MHz 时, 为了得到 460800 波特
 - $UARTDIV = 24000000/460800 = 52.08$
 - $BRR[15:0] = UARTDIV = 52 = 0x34$ (四舍五入)
- ◆ 在 48 MHz 时, 为了得到 115200 波特
 - $UARTDIV = 48000000/115200 = 416.66$
 - $BRR[15:0] = UARTDIV = 417 = 0x1A1$ (四舍五入)
- ◆ 在 48 MHz 时, 为了得到 921600 波特
 - $UARTDIV = 48000000/921600 = 52.08$
 - $BRR[15:0] = UARTDIV = 52 = 0x34$ (四舍五入)

预期波特率	实际波特率	BRR[15:0]	%误差
734Bps	733KBps	0xFFCC	0
1.2KBps	1.2KBps	0x9C40	0
2.4KBps	2.4KBps	0x4E20	0
4.8KBps	4.8KBps	0x2710	0
9.6KBps	9.6KBps	0x1388	0
19.2KBps	19.2KBps	0x9C4	0
38.4KBps	38.4KBps	0x4E2	0
57.6KBps	57.62KBps	0x341	0.03
115.2KBps	115.11KBp	0x1A1	0.08
230.4KBps	230.77KBp	0xD0	0.16
460.8KBps	461.54KBp	0x68	0.16
921.6KBps	923.07KBp	0x34	0.16
1.5Mbps	1.5Mbps	0x20	0
2Mbps	2Mbps	0x18	0
3Mbps	3Mbps	0x10	0

表 22-3 系统时钟为 48MHz，设置波特率时的误差计算

22.4.7 自动波特率侦测

UART 可以根据接收到的一个字节来检测和自动设置 UART_BRR 寄存器的值。自动波特率检测用于以下两种情况下：

- ◆ 通信速度不可知的情况下
- ◆ 使用低精度时钟源时，需要在不测量时钟偏差的条件下调整波特率的时候

时钟源的频率必须和预期的波特率保持相对的比例（过采样率为 16，并且波特率处于 $f_{PCLK}/65535$ 和 $f_{PCLK}/16$ 之间）。

在开启自动波特率检测时，必须先确认数据字节的内容。根据不同的数据字节内容选择模式，能够通过 UART_MCON 寄存器中的 ABRMOD 位进行选择。具体是：

- ◆ 模式 0 (0x00)：波特率在 UART 的 RX 引脚的两个连续的下降沿时间测量（起始位的下降沿和最低数据位的下降沿 (LSB)）
- ◆ 模式 1 (0x01)：波特率在 UART 的 RX 引脚下降沿和后续上升沿时间（起始位的长度）测量。
- ◆ 模式 2 (0x10)：波特率在 UART 的 RX 引脚下降沿和后续上升沿时间（起始位的长度加上 bit<0>的长度）测量（e.g. 0xFE）。

设置 UART_MCON 寄存器中的 ABREN 位为 1，开启自动波特率检测功能。UART 在 RX 上等待第一个数据字节过来。当自动波特率操作结束后，UART_RIF 寄存器中的 ABEND 位会被硬件自动设置为 1，若 UART_IER 寄存器中的 ABEND 位为 1 则会产生中断。

如果线路噪声严重，不能保证得到的波特率是准确的。这时 BRR 值可能是错的或者 ABEND

位会被设置为 1。在通信速度超出自动波特率检测范围（位长度不在 0x10 到 0xFFFF 个时钟周期之间）时也会发生这种情况。

在此模式中配置了两个中断，分别为侦测超时与侦测结束中断。

在软件未清除 UART_MCON 寄存器中的 ABREN 的情况下，UART 计数器会持续计数直到 0xFFFF 后，硬件会自动将 ABREN 清除，并将 UART_RIF 寄存器中的 ABTO 位设置为 1，如果开启中断，则会产生 ABTO 中断。

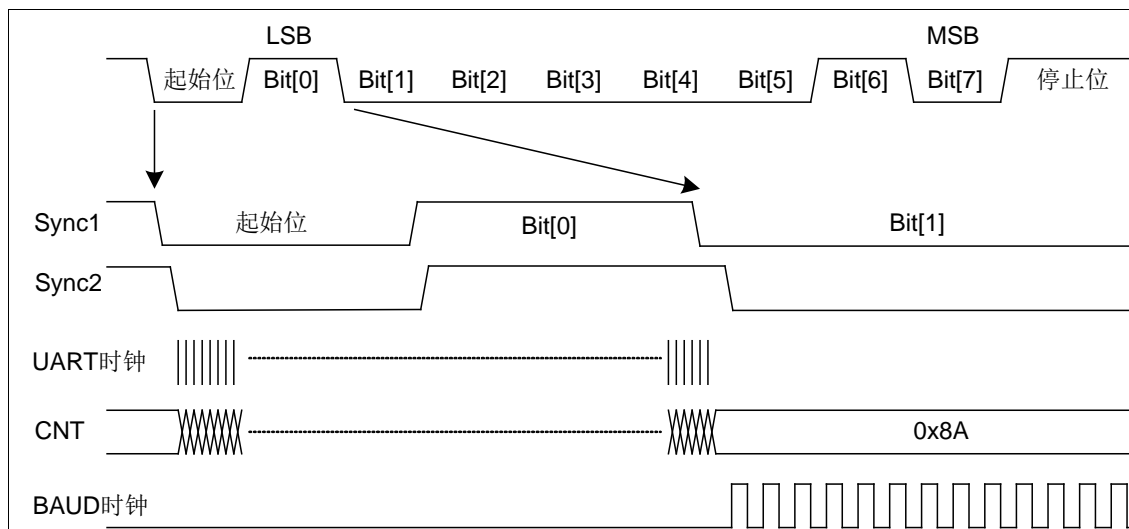


图 22-8 自动波特率侦测模式 0

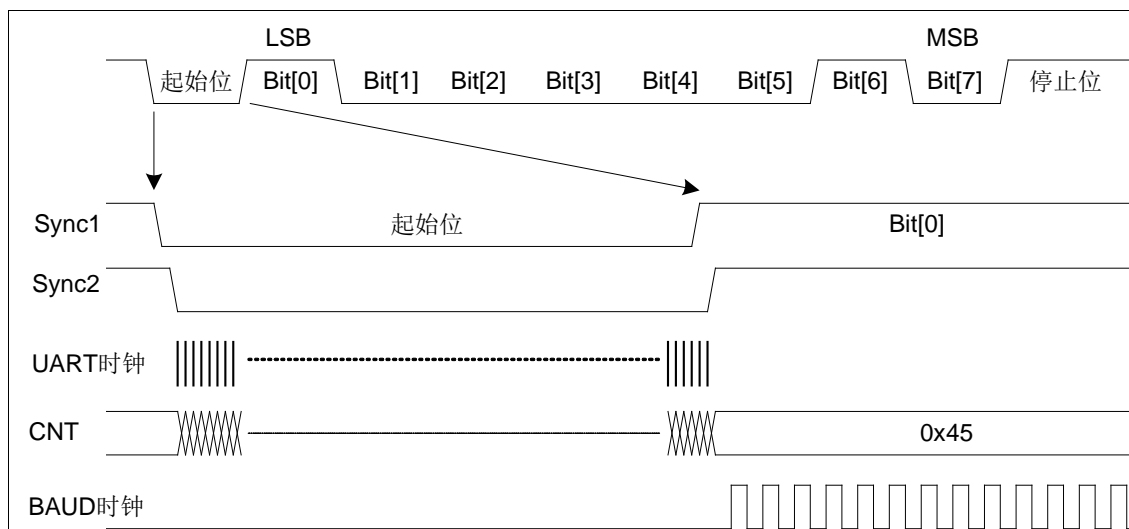


图 22-9 自动波特率侦测模式 1

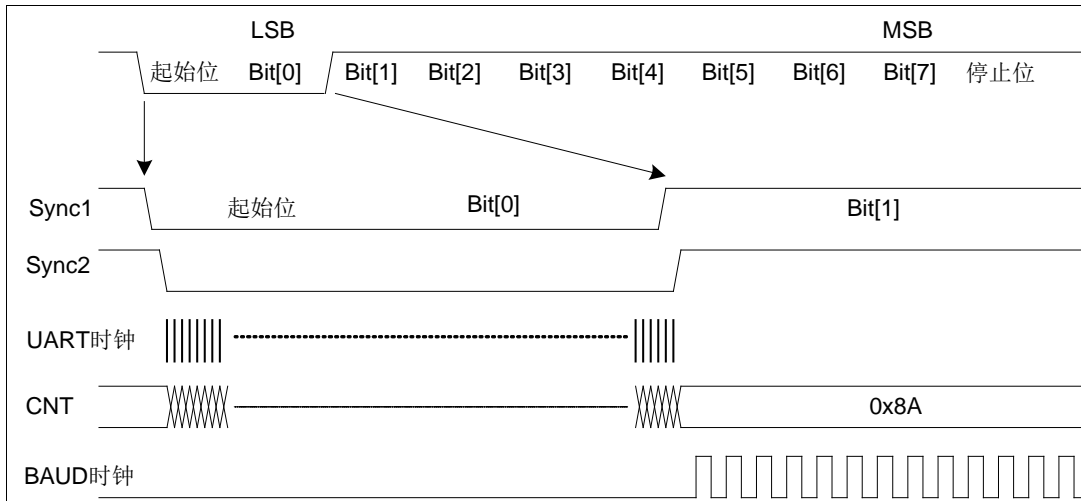


图 22-10 自动波特率侦测模式 2

22.4.8 自动流量控制

如果使能自动流控制，接收 FIFO 和发送 FIFO 会通过 UARTx_RTS 和 UARTx_CTS 引脚去控制 UART 的接收 (RX) 和发送 (TX)。

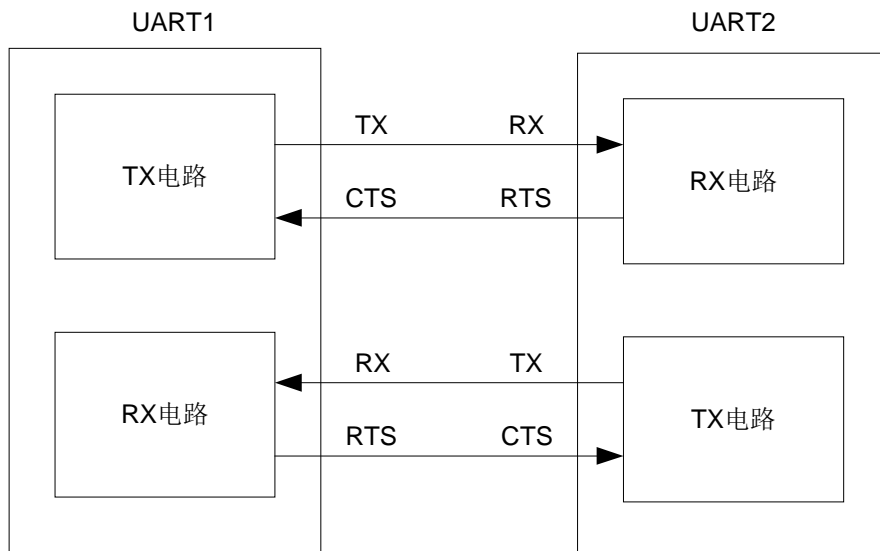


图 22-11 自动流量控制框图

22.4.8.1 RTS流量控制

当开启自动 RTS 控制时 (AFCEN=1)，当 UART 接收器准备好接收新数据，便会将 RTS 转为有效状态 (输出低电平)。当接收器已满时，会将 RTS 转为无效状态 (输出高电平)，表示发送过程会在当前帧结束后停止。

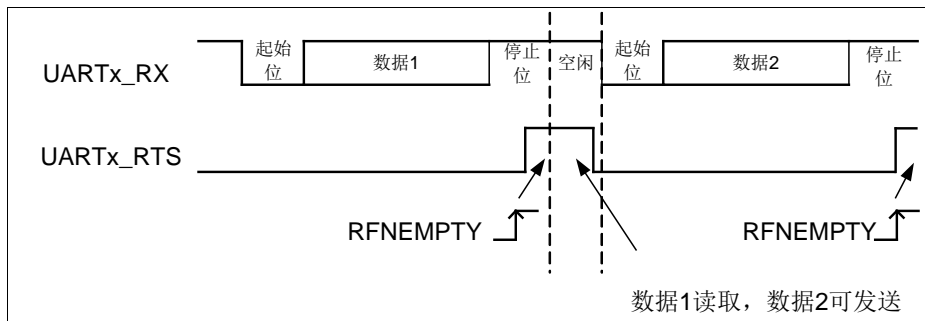


图 22-12 自动 RTSn 控制

22.4.8.2 CTSn流量控制

当开启自动 CTS 控制时 (AFCEN=1)，发送器会在发送下一帧前检查 CTS。如果 CTS 为有效状态 (输入低电平)，则会发送下一个数据 (假设数据已准备好发送，即 TFEMPTY=0)；否则不会进行发送。如果在发送过程中 CTS 转为无效状态 (输入高电平)，则当前发送完成之后停止发送器。只要 CTS 发生变化，UART_STAT 寄存器的 CTSSTA 位便会由硬件自动设置为 1 或 0。这表示接收器是否已准备好进行通信。如果 UART_IER 寄存器中的 DCTS 位为 1 则会产生中断。

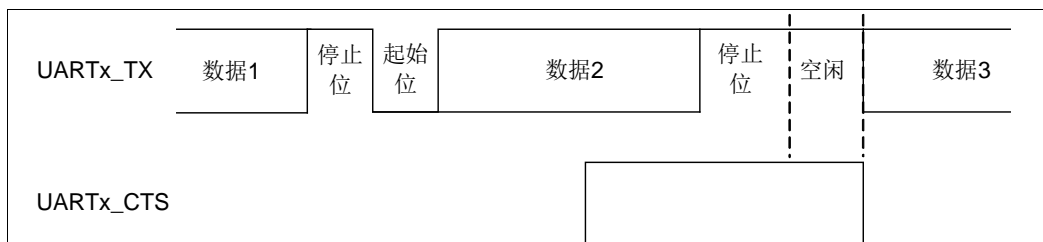


图 22-13 自动 CTSn 控制

22.4.8.3 RS485 驱动使能 (DE)

当开启 RS485 驱动开启功能 (UART_RS485 寄存器的 AADEN=1 或 AADNEN=1)，允许用户通过 DE (驱动开启) 信号来开启外部收发器的控制端。延迟时间是在发送最后一个数据字节的停止位和释放 DE 信号之间插入延迟，这个时间通过 UART_RS485 寄存器中的 DLY 位设置。而 DE 信号的极性则可以通过 UART_RS485 寄存器中的 AADINV 位中设置并进行选择极性。

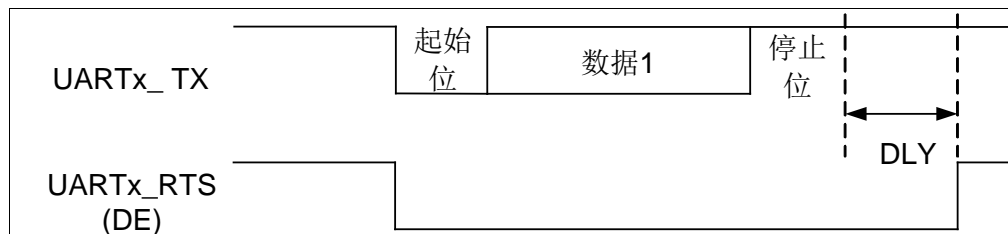


图 22-14 驱动开启, AADINV=0

22.4.9 Modbus通信

UART 提供对 Modbus/RTU 和 Modbus/ASCII 协议实现的基本支持。Modbus/RTU 是一个半双工的块式传输协议。协议的控制部分（地址检测，块完整性控制和命令解析）必须由软件来完成。UART 提供对块尾的基本支持检测，无需软件的经常性介入。

◆ **Modbus/RTU**

这个模式中，一个块结束为一个超过 2 个字节长度的空闲状态。通过设置一个超时长度来实现该功能。

超时功能和相应的中断必须通过 UART_RTOR 寄存器中的 RTOEN 位和 UART_IER 寄存器中的 RXTO 位来开启。UART_RTOR 寄存器中的 RTO 位要填入一个与超时长度相当的数字（例如 2 个字节长度为 22 个位长）。当接收线路保持空闲阶段达到这个长度时，在最后一个停止位被收到之后，会产生一个中断，表示当前的块接收已经完毕。

◆ **Modbus/ASCII**

在这个模式下，一个块结束通过特定 (CR/LF) 字节侦测，通过字节匹配功能实现该机制。将 LF 的 ASCII 码写到 ADD<7:0>区域，设置 UART_IER 寄存器中的 ADDRM 位为 1 开启功能，那么软件就会在收到 LF 字节后或者能够在 DMA 缓冲区中找到 CR/LF 字节时得到通知。

22.4.10 校验控制

设置 UART_LCON 寄存器中的 PE 位为 1 开启校验控制（发送时生成一个校验位，接收时进行校验位检查）。根据 DLS 位配置的帧长度，下表中列出可能的 UART 帧格式。

DLS<1:0>	PE	UART 帧
00	0	起始位+8 位数据+停止位
00	1	起始位+8 位数据+校验位+停止位
01	0	起始位+7 位数据+停止位
01	1	起始位+7 位数据+校验位+停止位
10	0	起始位+6 位数据+停止位
10	1	起始位+6 位数据+校验位+停止位
11	0	起始位+5 位数据+停止位
11	1	起始位+5 位数据+校验位+停止位

表 22-4 帧格式

◆ **奇校验**

校验位为一帧中的 8、7、6 或 5 个 LSB 位以及校验位'1'的个数为奇数。

例如：数据=00110101，有 4 个'1'，如果选择奇校验（在 UART_LCON 寄存器中的 PS 位为 0），校验位将是'1'。

◆ **偶校验**

校验位为一帧中的 8、7、6 或 5 个 LSB 位以及校验位'1'的个数为偶数。

例如：数据=00110101，有 4 个'1'，如果选择偶校验（在 UART_LCON 寄存器中的 PS

位为 1)，校验位将是'0'。

◆ 接收时的校验检查

如果校验检查失败，UART_STAT 寄存器中的 PERR 位会被设置为 1，如果 UART_IER 寄存器中的 RXBERR 位为 1 则会产生中断。

◆ 发送时的校验生成

设置 UART_LCON 寄存器的 PE 位为 1 时，写进数据寄存器的数据的 MSB 位由校验位替换后发送出去（选择奇校验奇数个'1'（PS=0）或选择偶校验偶数个'1'（PS=1））。

22.4.11 多处理器通信

设置 DLS 位为 8 位字长（第 9bit 为判断地址或数据）。

设置 UART_RS485 寄存器的 AADEN 位为 1 以进入模式。

设置 UART_RS485 寄存器的 ADDR 位配置匹配地址。

可以将多个 UART 连接成一个网络来实现多机通信。例如某个 UART 设备可以是主 UART，它的 TX 输出和其他 UART 从设备的 RX 输入相连接；UART 从设备各自的 TX 输出在逻辑上通过与运算连在一起，并且和主设备的 RX 输入相连接。

在多处理器配置中，通过特定地址的将接收器开启，来接收随后的数据，这样就可以减少由未被寻址接收器的参与带来的多余的 UART 服务开销。

未被寻址的设备可开启静默功能进入静默模式。设置 UART_RS485 寄存器的 AADEN 位为 1 开启静默模式功能。

在这个模式中，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 UART_RS485 寄存器的 ADDR 位。

如果接收到的字节与它设置的地址不匹配时，UART 进入静默模式。当 UART 进到静默模式后，接收字节时既不会影响 UART_RIF 寄存器的 RFNEMPTY 位，也不会产生中断或发出 DMA 请求。

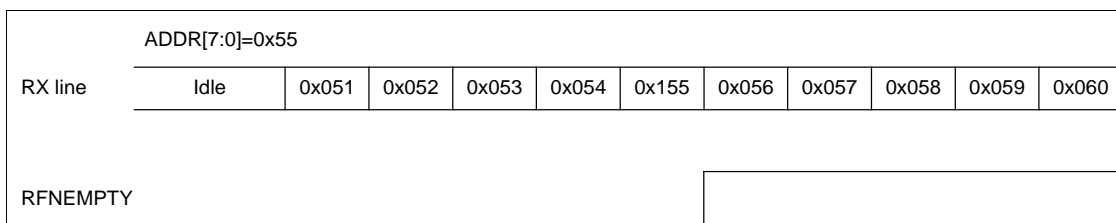


图 22-15 使用地址标示侦测模式

22. 4. 12 LIN模式（仅EUART支持）

◆ LIN 发送

和普通的 UART 发送相同，但有下列区别：

设置 DLS 位为 8 位字长。

设置 UART_LIN 寄存器的 LINEN 位为 1 以进入 LIN 模式。设置 UART_LIN 寄存器的 LINBKRQ 位为 1 将发送 13 位'0' 作为断路符号。然后发两位'1'，以检测下一个开始位。

◆ LIN 接收

当 LIN 模式开启时(UART_LIN 寄存器的 LINEN 位为 1)，检测断路符号电路自动被开启。该检测独立于 UART 接收器。不管是在空闲状态还是在发送数据期间，断路符号只要一出现就能被检测到。

一旦接收器被开启(LCON 寄存器的 RXEN 位为 1)，电路就开始检测 RX 上的起始信号。检测起始位的方法和检测断路符号或数据是一样的。当起始位被检测到后，电路检测接下来的每个位，在每个位的第 8, 9, 10 个过采样时钟点上进行采样，就像采样数据一样。如果 10 个（当 UART_LIN 寄存器中的 LINBDL=0）或 11 个（当 UART_MCON 寄存器中的 LINBDL=1）连续位都是'0'，并且又跟着一个分隔符，UART_RIF 寄存器的 LINBK 位就会被设置为 1。如果 UART_IER 寄存器的 LINBK 位为 1 则会产生中断。在确认断路符号前，要检查分隔符，因为它表示 RX 已经回到高电平。如果在第 10 或 11 个采样点之前采样到了'1'，检测电路取消当前检测并重新寻找起始位。如果 LIN 模式被关闭，接收器则继续正常 UART 工作，不再考虑检测断路符号。如果 LIN 模式被开启（LINEN=1），只要一发生帧错误（例如：停止位检测到'0'，这种情况出现在断路符号被接收到的时候），接收器就会立即停止，直到电路检测断路符号后接收到一个'1'（这种情况发生于断路符号没有完整的发出来），或一个分隔符（这种情况发生于已经检测到一个完整的断路符号）。

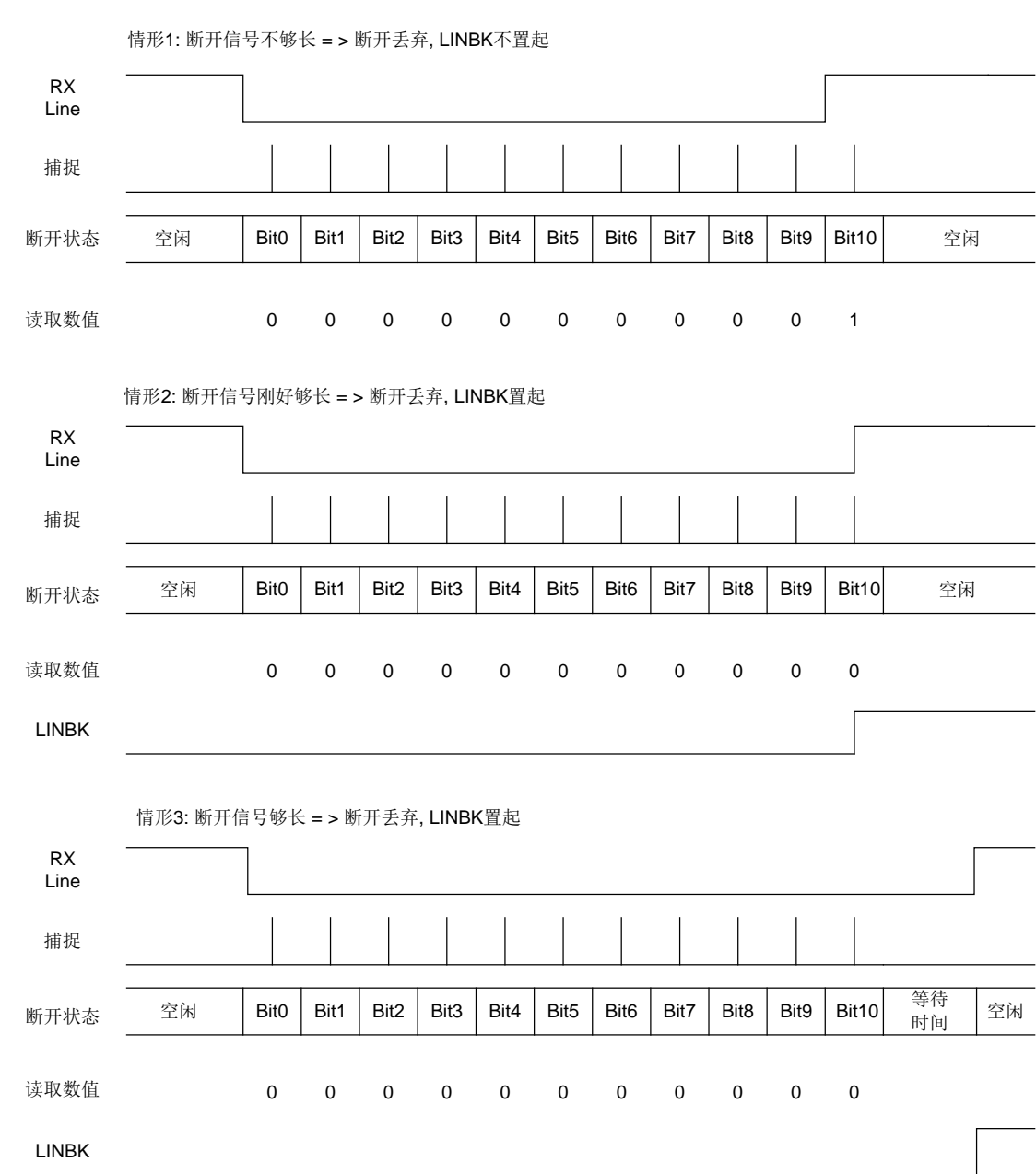


图 22-16 LIN 模式侦测断路信号 (11 位断路长度-LBDL 位为 1)

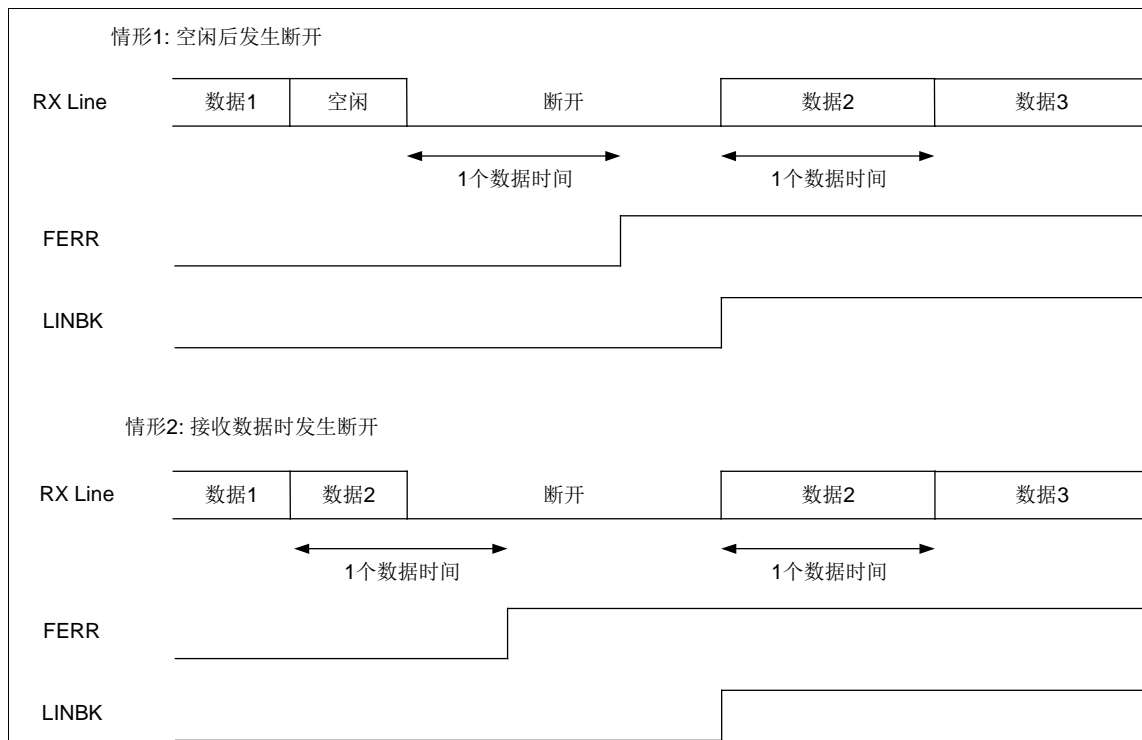


图 22-17 LIN 模式侦测断路信号与帧错误信号

22.4.13 单线半双工通讯（仅EUART支持）

UART 可以配置成遵循单线半双工协议。在单线半双工模式中，TX 和 RX 引脚在芯片内部是连在一起的。使用控制位（UART_MCON 寄存器中的 HDEN 位）选择半双工或全双工通信。

◆ 当 HDEN 为 1 时：

- TX 和 RX 引脚在芯片内部是连在一起的。
- RX 不再被使用。
- 当没有数据传输时，TX 引脚为释放状态。因此，在空闲状态或接收状态时为一个标准 I/O 串口。这就表示该 I/O 串口在不被 UART 驱动时，必须配置成悬空输入（或开漏的高输出）。

除此以外，通信与正常 UART 模式类似。由软件来管理在线的冲突（例如通过使用一个中央仲裁器）。特别的是，发送从不会被硬件所阻碍。当 LCON 寄存器的 TXEN 位为 1，只要数据一写到数据寄存器中，发送就会开始。

22.4.14 智能卡模式（仅EUART支持）

设置 8 位数据位加校验位：即 LCON 寄存器中 DLS=00，PE=1。

设置 0.5/1.5 个停止位：即 LCON 寄存器的 STOP=0/1。

设置 UART_SCARD 寄存器的 SCEN 位为 1 以进入智能卡模式。

在 T=0（字节）模式中，当校验错误时，在数据字节发送完毕后将接收器拉为低电平产生保护时间。

下图所示为数据在线有校验错误和没有校验错误时的情形。

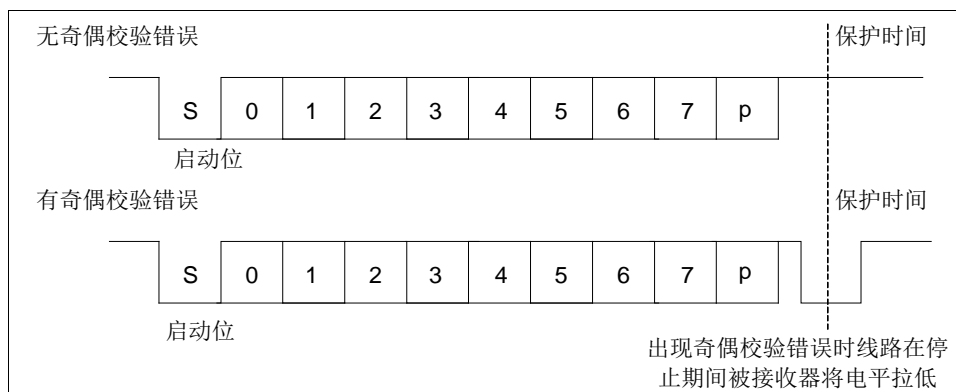


图 22-18 ISO 7816-3 异步协定

当连接到智能卡时，UART 的 TX 引脚和智能卡通过同一根双向数据线进行通信。所以 TX 引脚必须配置成开漏状态。智能卡是一个单线半双工通信协议：

- ◆ 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个时钟边沿开始移位。在智能卡模式中，此发送过程还会产生一个 1/2 波特时钟周期的延迟。
- ◆ 如果在接收一个使用 0.5 或 1.5 个停止位设置的帧期间检测到奇偶位错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 UART 的数据尚未正确接收。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。软件根据协议重新发送数据。如果设置 NACK 位为 1，则接收器会发送 NACK 信号；否则不会发送 NACK 信号（T=1 模式中使用）。
- ◆ 通过对保护时间寄存器进行设置，可以延迟 UART_STAT 寄存器的 TBC 的设置时间。正常工作时，当发送移位寄存器为空时，会对 TBC 位设置为 1。在智能卡模式中，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TBC 位被强制设置为 0。当保护时间计数器达到设置值时，TBC 位被设置为 1。
- ◆ 对 TBC 位的设置不受智能卡模式的影响。
- ◆ 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收器不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。

- ◆ 在接收端，如果检测到奇偶位错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注：在智能卡模式中带有帧错误的 0x00 数据将被视为数据，而非中断。

下图详细介绍了 UART 如何对 NACK 信号采样。在本例中，UART 正在发送数据并配置了 1.5 个停止位。UART 的接收部分已开启，以检查数据的完整性和 NACK 信号。

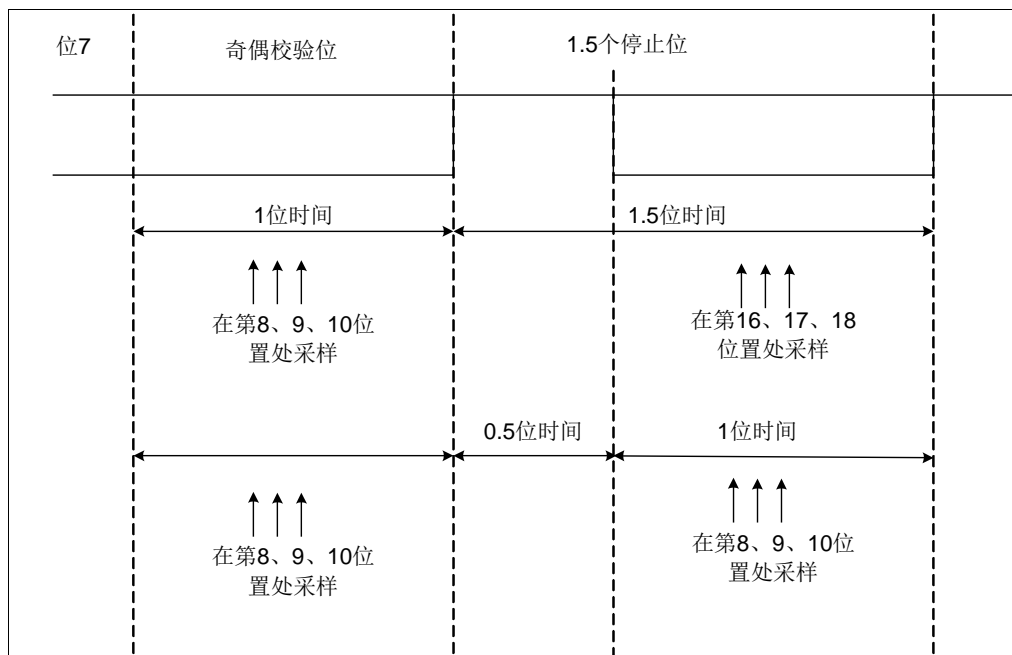


图 22-19 1.5 位停止位时检测校验错误

UART 可以通过 CK 引脚向智能卡提供时钟。智能卡模式中，CK 时钟和通信没有关系，只是通过一个 5 位的预分频器从内部外设时钟源得到时钟信号。这个分频系数通过 UART_SCARD 寄存器的 PSC 位配置。CK 频率可以设置在 $f_{CK}/2$ 到 $f_{CK}/64$ 之间， f_{CK} 指外设输入时钟。

22. 4. 15 IrDA SIR 模式（仅 EUART 支持）

设置 UART_MCON 寄存器的 IREN 为 1 以进入 IrDA 模式。

IrDA SIR 物理层规定使用反相归零（RZI）调制方案，它以红外光脉冲表示逻辑 0。

SIR 发送编码器用于调制 UART 发出的非归零（NRZ）编码。输出脉冲会发送到外部输出驱动器和红外线 LED。UART 支持的 SIR 编码比特率最高为 115.2Kbps。在正常模式中，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收译码器用于解调由红外探测器发出的归零编码，并将接收到的 NRZ 串行编码输出到 UART。在空闲状态下，译码器输入为高电平（标记状态）。发送编码器输出的极性与译码器输入相反。当译码器输入为低电平时，会检测到起始位。

IrDA 是一个半双工通信协议。如果发送器发送时，例如 UART 正在向 IrDA 编码器发送数据，则 IrDA 译码器会忽略 IrDA 接收在线的所有数据；如果接收器接收时，例如 UART 正在接收来自 RX 引脚上的数据，则 IrDA 不会对 UART 发送到 IrDA 的 TX 数据进行编码。在接收数据时，

应避免同时进行发送，因为这样做可能会破坏要发送的数据。

- ◆ SIR 发送逻辑把 0 作为高脉冲发送，把 1 作为低电平发送。脉冲的宽度规定为所选位周期的 3/16。
- ◆ SIR 译码器用于将兼容 IrDA 的接收信号转换为 UART 的编码。
- ◆ SIR 接收逻辑把高电平状态作为 1，把低脉冲作为 0。
- ◆ 发送编码器输出的极性与译码器输入相反。
- ◆ 在 IrDA 模式中，LCON 寄存器中的 STOP 位必须配置成 1 个停止位。

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10ms 的时间间隔（IrDA 是一个半双工协议）。

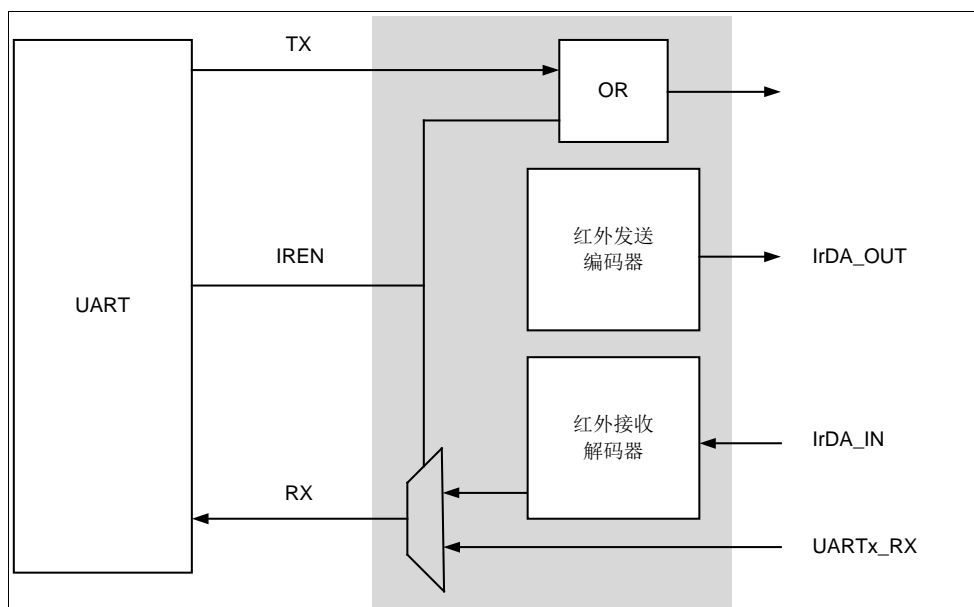


图 22-20 红外收发框图

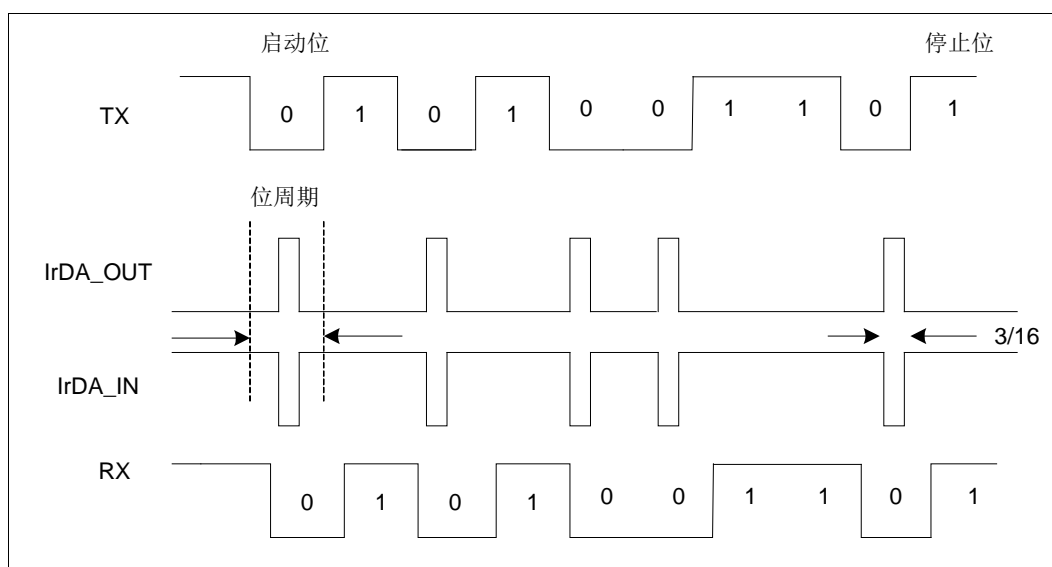


图 22-21 IrDA 数据调制 (3/16) - 正常模式

22.4.16 使用DMA连续通讯

设置 UART_MCON 的 RXDMAEN 位为 1 开启 RX DMA 或 TXDMAEN 位为 1 开启 TX DMA。

UART 可以利用 DMA 连续通信。RX 缓冲器和 TX 缓冲器分别产生 DMA 请求。

利用 DMA 发送

使用 DMA 进行发送, 可以通过设置 UART_MCON 寄存器中的 TXDMAEN 位开启。当 TFEMPTY 位被设置为 1, 可使用 DMA 外设 (请参见相应的 DMA 控制器部分) 将数据从配置的 SRAM 地址加载到 UART_TXBUF 寄存器中。要映像一个 DMA 通道以进行 UART 发送, 请按以下步骤操作:

- ◆ 通过 DMA 控制寄存器设置 UART_TXBUF 寄存器地址配置成 DMA 传输的目的地址。在每个 TFEMPTY 事件后, 数据将被传送到这个地址。
- ◆ 通过 DMA 控制寄存器将 SRAM 地址配置成 DMA 传输的来源地址。在每个 TFEMPTY 事件后, 将从此内存区读出数据并传送到 UART_TXBUF 寄存器中。
- ◆ 通过 DMA 控制寄存器配置要传输的字节数。
- ◆ 通过 DMA 寄存器配置信道优先级。
- ◆ 根据应用程序的要求, 配置在传输完成一半还是全部完成时产生 DMA 中断。
- ◆ 设置 UART_ICR 寄存器的 TFEMPTY 位为 1 以清除 UART_RIF 寄存器的 TFEMPTY 位。
- ◆ 设置 DMA 寄存器开启该通道。

在发送模式中, 当 DMA 传输完所有要发送的数据时, DMA 控制器设置 UART_IFM 寄存器的 TFEMPTY 位; 检查 UART_RIF 寄存器的 TFEMPTY 位可以确认 UART 通信是否结束。这样可以在关闭 UART 或进入停机模式之前, 避免破坏最后一次传输的数据。软件必须等待 TSBUSY 被设置为 0。TSBUSY 位在全部数据发送期间会是 1, 并且在最后一帧数据发送完成之后会由硬件设置为 0。

利用 DMA 接收

使用 DMA 进行接收, 可以通过设置 UART_MCON 寄存器中的 RXDMAEN 位为 1 开启。当接收数据字节时, 可使用 DMA 外设 (请参见相应的 DMA 控制器部分) 将数据从 UART_RXBUF 寄存器读取出来加载到配置的 SRAM 地址。要映像一个 DMA 通道以进行 UART 接收, 请按以下步骤操作:

- ◆ 通过 DMA 控制寄存器设置 UART_RXBUF 寄存器地址配置成 DMA 传输的来源地址。在每个 RFNEMPTY 事件后, 读取数据将从这个地址加载。
- ◆ 通过 DMA 控制寄存器将 SRAM 地址配置成 DMA 传输的目标地址。在每个 RFNEMPTY 事件后, 读取数据将从 UART_RXBUF 寄存器加载这个目标地址。
- ◆ 通过 DMA 控制寄存器配置要传输的字节数。
- ◆ 通过 DMA 寄存器配置信道优先级。
- ◆ 根据应用程序的要求, 配置在传输完成一半还是全部完成时产生 DMA 中断。

- ◆ 设置 UART_ICR 寄存器的 RFNEMPTY 位为 1 以清除 UART_RIF 寄存器的 RFNEMPTY 位。
- ◆ 设置 DMA 寄存器开启该通道。

多缓冲器通信中的错误标志和中断产生

在多缓冲器通信的情况下，通信期间如果发生任何错误，会在当前字节传输后将错误标志设置为 1。如果中断开启位被设置为 1 则会产生中断。在单个数据字节接收的情况下，UART_IFM 寄存器中的 RXBERR 位和 RFOERR 位一起被设置为 1，表示帧错误和溢出错误，有分别的错误标志中断开启位，如果被设置为 1，则会在当前字节传输结束后，产生中断。

22.4.17 中断配置

- ◆ **UART_IER 中断开启寄存器**
此位设置 1 时，表示开启中断功能，并且同时反映在 UART_IVS 寄存器。此寄存器只能写入，并且只允许写入 1，无法写 0 取消开启中断设置。
- ◆ **UART_IDR 中断关闭寄存器**
此位设置 1 时，表示关闭中断功能，并且同时反映在 UART_IVS 寄存器。此寄存器只能写入，并且只允许写入 1，无法写 0 取消关闭中断设置。
- ◆ **UART_IVS 中断功能有效状态寄存器**
反映 UART_IER 与 UART_IDR 寄存器所设置的结果。0：中断关闭；1：中断开启
- ◆ **UART_RIF 原始中断状态寄存器**
反映所有发生中断事件的状态，无论 UART_IVS 是否有开启中断，皆会反映在此寄存器中，主要用于监控无屏蔽的中断位，是否有错误事件发生。
- ◆ **UART_IFM 中断屏蔽后状态寄存器**
记录中断开启位所发生的中断事件。0：无中断事件；1：发生中断事件
- ◆ **UART_ICR 中断清除寄存器**
此位设置 1 时，清除中断标志 UART_RIF 与 UART_IFM，此寄存器通过写入 1 将该位清零，并且只允许写入 1 清除，无法写 0 清除。

在 UART 中，配置了 15 种中断，如下表所示。

中断事件	中断标志
接收器字节格式错误	RXBERR
自动波特率侦测结束	ABEND
自动波特率侦测超时	ABTO
CTS 引脚电平改变	DCTS
接收超时	RXTO
地址匹配	ADDRM
LIN 断开侦测	LINBK
块结束	EOB
噪声位侦测	NOISE

接收器非空	RFNEMPTY
接收器溢出	RFOERR
接收器下溢	RFUERR
发送器发送完成	TBC
发送器空	TFEMPTY
发送器溢出	TFOVER

表 22-5 中断配置表

22.5 特殊功能寄存器

22.5.1 寄存器列表

UART 寄存器列表		
名称	偏移地址	描述
UART_RXBUF	0000 _H	接收缓冲寄存器
UART_TXBUF	0004 _H	发送缓冲寄存器
UART_BRR	0008 _H	波特率寄存器
UART_LCON	000C _H	格式控制寄存器
UART_MCON	0010 _H	模式控制寄存器
UART_RS485	0014 _H	RS485 控制寄存器
UART_SCARD	0018 _H	智能卡控制寄存器
UART_LIN	001C _H	LIN 控制寄存器
UART_RTOR	0020 _H	接收超时寄存器
UART_STAT	0028 _H	状态寄存器
UART_IER	002C _H	中断开启寄存器
UART_IDR	0030 _H	中断关闭寄存器
UART_IVS	0034 _H	中断功能有效状态寄存器
UART_RIF	0038 _H	原始中断状态寄存器
UART_IFM	003C _H	中断屏蔽后状态寄存器
UART_ICR	0040 _H	中断清除寄存器

22.5.2 寄存器描述

22.5.2.1 接收缓冲寄存器 (UART_RXBUF)

接收缓冲寄存器 (UART_RXBUF)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RXBUF							

—	Bit 31-9	—	—
RXBUF	Bit 8-0	R	<p>接收缓冲寄存器</p> <p>包含接收到的数据字节。</p> <p>RXBUF寄存器提供接收移位寄存器和内部总线间的并行接口。</p> <p>注: 位8用于RS485地址模式</p>

22.5.2.2 发送缓冲寄存器 (UART_TXBUF)

发送缓冲寄存器 (UART_TXBUF)																															
偏移地址: 04 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								TXBUF							

—	Bit 31-9	—	—
TXBUF	Bit 8-0	R/W	<p>发送缓冲寄存器</p> <p>用于写入要发送的数据字节。</p> <p>TXBUF寄存器提供发送移位寄存器与内部总线间的并行接口。</p> <p>注: 位8用于RS485地址模式</p>

22.5.2.3 波特率寄存器 (UART_BRR)

波特率寄存器 (UART_BRR)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																BRR															

—	Bit 31-16	—	—
BRR	Bit 15-0	R/W	<p>波特率寄存器</p> <p>整数部分 $BRR\langle 15:4 \rangle = DIVISOR\langle 11:0 \rangle$</p> <p>小数部分 $BRR\langle 3:0 \rangle = FRACTION\langle 3:0 \rangle$</p> <p>此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>注：使用自动波特率功能时则可自动写入</p>

22.5.2.4 格式控制寄存器 (UART_LCON)

格式控制寄存器 (UART_LCON)																																				
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved																TXEN	RXEN	DBCEN	Reserved	BREAK	SWAP	TXINV	RXINV	DATAINV	MSB	PS	PE	STOP	DLS							

—	Bit 31-16	—	—
TXEN	Bit 15	R/W	发送器开启位 开启发送器，此位由软件设置1和清除。 0: 发送器关闭 1: 发送器开启
RXEN	Bit 14	R/W	接收器开启位 开启接收器，此位由软件设置1和清除。 0: 接收器关闭 1: 接收器开启
DBCEN	Bit 13	R/W	防抖动开启位 开启防抖动功能，此位由软件设置为1和清除。 0: 防抖动关闭 1: 防抖动开启，RX的有效高低电平都须维持至少8个时钟周期
—	Bit 12-11	—	—
BREAK	Bit 10	R/W	断路开启位 开启断路功能，此位由软件设置为1和清除。 此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。 0: 断路关闭 1: 断路开启，会使得TX输出为0
SWAP	Bit 9	R/W	交换TX/RX引脚开启位 开启交换功能，此位由软件设置为1和清除。 此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。 0: 交换关闭，TX和RX引脚按照标准引脚配置使用 1: 交换开启，TX和RX引脚功能交换使用，此功能用与其他UART接口进行交叉互联时

TXINV	Bit 8	R/W	<p>TX引脚电平反向使能位</p> <p>TX引脚反向功能，此位由软件设置为1和清除。此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: TX引脚信号工作于标准逻辑电平(VDD=1/idle, Gnd=0/mark)</p> <p>1: TX引脚信号反向(VDD=0/mark, Gnd=1/idle)。此功能可用于TX上带有外部反向器时</p>
RXINV	Bit 7	R/W	<p>RX引脚电平反向使能位</p> <p>RX引脚反向功能，此位由软件设置为1和清除。此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: RX引脚信号工作于标准逻辑电平(VDD=1/idle, Gnd=0/mark)</p> <p>1: RX引脚信号反向(VDD=0/mark, Gnd=1/idle)。此功能可用于RX在线带有外部反向器时</p>
DATAINV	Bit 6	R/W	<p>数据反向开启位</p> <p>数据反向功能，此位由软件设置为1和清除。此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: 缓冲寄存器中的逻辑数据在接收的时候，采用正/直接逻辑。(1=H, 0=L)</p> <p>1: 缓冲寄存器中的逻辑数据在接收的时候，采用负/反向逻辑。(1=L, 0=H)</p>
MSB	Bit 5	R/W	<p>高位在前开启位</p> <p>高位在前功能，此位由软件设置为1和清除。此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: 数据在发送和接收的时候，采用起始位后面跟着第0位(LSB)的顺序</p> <p>1: 数据在发送和接收的时候，采用起始位后面跟着的最高位(MSB)的顺序</p>
PS	Bit 4	R/W	<p>校验位奇偶选择</p> <p>当开启校验功能时，选择校验位为奇校验或偶校验，此位由软件设置为1和清除。此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: 奇校验</p>

			1: 偶校验
PE	Bit 3	R/W	<p>校验开启位</p> <p>开启校验功能，计算好的校验位被插入到最高位，并检测接收数据的校验位（接收与发送功能），此位由软件设置为1和清除。</p> <p>此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>0: 校验位关闭 1: 校验位开启</p>
STOP	Bit 2	R/W	<p>停止位选择</p> <p>此位由软件设置为1和清除。</p> <p>此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>普通模式： 0: 1个停止位 1: 2个停止位（在5字长模式为1.5个停止位）</p> <p>智能卡模式： 0: 0.5个停止位 1: 1.5个停止位</p>
DLS	Bit 1-0	R/W	<p>数据字长选择位</p> <p>此位由软件设置为1和清除。</p> <p>此位在LCON寄存器中的RXEN位与TXEN位为0时才可以写入。</p> <p>00: 8字长 01: 7字长 10: 6字长 11: 5字长</p>

22.5.2.5 模式控制寄存器 (UART_MCON)

模式控制寄存器 (UART_MCON)																															
偏移地址: 0x10																															
复位值: 0x0000 0000																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															TXFLOAT	TXDMAEN	RXDMAEN	Reserved	ABRREPT	ABRMOD	ABREN	Reserved	BKREQ	HIDEN	IREN	AFCEN	RTSSET	LPBKEN			

—	Bit 31-17	—	—
TXFLOAT	Bit 16	R/W	<p>发送器等待发送状态选择 此位由软件设置为1和清除。 0: 发送器未发送时, TX引脚输出高 1: 发送器未发送时, TX引脚开漏</p>
TXDMAEN	Bit 15	R/W	<p>发送器DMA开启 此位由软件设置为1和清除。 0: 发送器DMA通信关闭 1: 发送器DMA通信开启</p>
RXDMAEN	Bit 14	R/W	<p>接收器DMA开启 此位由软件设置为1和清除。 0: 接收器DMA通信关闭 1: 接收器DMA通信开启</p>
—	Bit 13-12	—	—
ABRREPT	Bit 11	R/W	<p>重复侦测自动波特率 在开启侦测自动波特率时, 侦测波特率超时时并不会清除自动波特率开关, 并在下一个下降沿时重复侦测自动波特率, 此位由软件设置为1和清除。 0: 重复侦测自动波特率关闭 1: 重复侦测自动波特率开启</p>
ABRMOD	Bit 10-9	R/W	<p>自动波特率模式选择 此位由软件设置为1和清除。 00: 模式0, 侦测第一个下降沿到第二个下降沿时间 (侦测2位) 01: 模式1, 侦测第一个下降沿到第一个上升沿时间 (侦测1位) 10: 模式2, 侦测第一个下降沿到第一个上升沿时间 (侦测2位)</p>

			11: 保留
ABREN	Bit 8	R/W	自动波特率开启 此位在开启并完成侦测自动波特率后会 自动清除, 也可由软件设置为1和清除。 0: 自动波特率关闭 1: 自动波特率开启
—	Bit 7-6	—	—
BKREQ	Bit 5	W	断开请求 此位在写入后的下一个时钟会自动清除。 0: 断路请求关闭 1: 断路请求开启, 根据设定的N位长 (8、7、6 或5) 产生N个低脉冲信号
HDEN	Bit 4	R/W	单线半双工开启 此位由软件设置为1和清除。 此位在LCON寄存器中的RXEN位与TXEN位为0 时才可以写入。 0: 单线半双工关闭 1: 单线半双工开启
IREN	Bit 3	R/W	IrDA红外线模式开启 此位由软件设置为1和清除。 0: IrDA红外线模式关闭 1: IrDA红外线模式开启
AFCEN	Bit 2	R/W	自动流量控制开启 此位由软件设置为1和清除。 0: 自动流量控制关闭 1: 自动流量控制开启
RTSSET	Bit 1	R/W	RTS设置控制 此位由软件设置为1和清除。 0: 自动流量控制关闭时, RTS引脚输出高 1: 自动流量控制关闭时, RTS引脚输出低
LPBKEN	Bit 0	R/W	回送模式使能 此模式是用于UART测试检测模式, 在UART普通 模式下运行, TX引脚输出为高电平, 串行的数据 在内部回送至RX。在此模式下, 所有中断都是正 常运行的, 此位由软件设置为1和清除。 0: 回送模式关闭 1: 回送模式开启

22.5.2.6 RS485 控制寄存器 (UART_RS485)

RS485 控制寄存器 (UART_RS485)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DLY								ADDR								Reserved				AADINV	AADACEN	AADNEN	AADEN

—	Bit 31-24	—	—
DLY	Bit 23-16	R/W	<p>延迟数值</p> <p>此位由软件设置和清除。</p> <p>用于设置延迟RTS的输出时间,是由一个8位计数器计数,时钟源为1/16 Baud 时钟。</p>
ADDR	Bit 15-8	R/W	<p>地址匹配数值</p> <p>此位由软件设置和清除。</p> <p>用于多机通信时地址标记的检测。</p> <p>接收器在RS485自动检测模式时,当接收数据的最高位为1且匹配ADDR,数据才允许接收,否则舍弃此数据。</p>
—	Bit 7-4	—	—
AADINV	Bit 3	R/W	<p>驱动开启反向</p> <p>在自动流量控制模式时,设置驱动开启引脚(RTS/DE)的输出电平,此位由软件设置和清除。</p> <p>0: 当开始发送数据时,驱动开启引脚输出0,发送完成且TX内无数据时,驱动开启引脚输出1</p> <p>1: 当开始发送数据时,驱动开启引脚输出1,发送完成且TX内无数据时,驱动开启引脚输出0</p>
AADACEN	Bit 2	R/W	<p>自动流量控制模式开启</p> <p>此位由软件设置为1和清除。</p> <p>0: 自动流量控制模式关闭</p> <p>1: 自动流量控制模式开启</p>
AADNEN	Bit 1	R/W	<p>普通模式开启</p> <p>此位由软件设置为1和清除。</p> <p>0: 普通模式关闭</p> <p>1: 普通模式开启,接收地址位为第8位 (UART_RXBUF)</p>

AADEN	Bit 0	R/W	<p>自动地址侦测模式开启</p> <p>在普通模式时，设置此位无效，此位由软件设置为1和清除。</p> <p>0: 自动地址侦测模式关闭</p> <p>1: 自动地址侦测模式开启，当接收数据的地址为1且匹配ADDR时，才会接收数据</p>
-------	-------	-----	---

22.5.2.7 智能卡控制寄存器 (UART_SCARD)

智能卡控制寄存器 (UART_SCARD)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLEN								GT								PSC								Reserved		SCCNT		SCLKEN	SCNACK	SCEN	

BLEN	Bit 31-24	R/W	<p>块长度</p> <p>设置了智能卡模式T=1的接收时的块长度，此位由软件设置为1和清除。</p> <p>例如： BLEN = 0 → 0个信号字节 BLEN = 1 → 1个信号字节 BLEN = 255 → 255个信号字节</p> <p>这个功能也可以在其他模式中使用，当LCON寄存器的RXEN位清除时，块长度计数器会重新计数</p>
GT	Bit 23-16	R/W	<p>保护时间</p> <p>设置保护时间长度，是使用波特时钟为单位。</p> <p>在智能卡模式中使用，完成标志（RIF寄存器TBC位）在保护时间过后设置为1，此位由软件设置为1和清除</p>
PSC	Bit 15-8	R/W	<p>分频器数值</p> <p>此位由软件设置为1和清除。</p> <p>在红外低功耗和正常模式下： PSC[7:0]: IrDA正常和低功耗模式波特率对UART时钟源进行分频以获得低功耗模式下的频率： 00000000: 保留 00000001: 1分频 00000010: 2分频</p> <p>智能卡模式： PSC[4:0]: 输出时钟分频数值 用于设定UART时钟的分频数，得到智能卡输出时钟，由五个有效位组成，乘以2得到的数值作为分频</p>

			00000: 保留 00001: 2分频 00010: 4分频 00011: 6分频
—	Bits 7-6	—	—
SCCNT	Bits 5-3	R/W	智能卡重试计数器 设置智能卡模式中接收和发送的重试次数。此位由软件设置为1和清除。 在发送模式下，在产生帧错误前重试发送的次数 在接收模式下，在接收到NACK后重试接收的次数 0x0: 重试功能关闭，在发送与接收模式下不进行自动重试 0x1~0x7: 在产生错误前自动重试的次数
SCLKEN	Bit 2	R/W	智能卡时钟开启 此位由软件设置为1和清除。 0: CK引脚关闭 1: CK引脚开启
SCNACK	Bit 1	R/W	智能卡NACK发送开启 此位由软件设置为1和清除。 0: 出现校验错误时关闭发送NACK信号 1: 出现校验错误时开启发送NACK信号
SCEN	Bit 0	R/W	智能卡模式开启 此位由软件设置为1和清除。 0: 智能卡模式关闭 1: 智能卡模式开启

22.5.2.8 LIN控制寄存器 (UART_LIN)

LIN 控制寄存器 (UART_LIN)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										LINBKREQ	LINBDL	LINEN			

—	Bit 31-3	—	—
LINBKREQ	Bit 2	W1	<p>LIN模式断路请求开启位 在LIN模式下，发送器将发送13位'0' 作为断路符号后，发送2位1用于对下一个开始位的检测。此位由软件设置为1并在下一个时钟后自动清除。</p> <p>0: LIN模式断路请求关闭 1: LIN模式断路请求开启</p>
LINBDL	Bit 1	R/W	<p>LIN模式断路字长选择位 此位由软件设置为1和清除。</p> <p>0: 10位断路字节侦测 1: 11位断路字节侦测</p>
LINEN	Bit 0	R/W	<p>LIN模式开启位 此位由软件设置为1和清除。</p> <p>0: LIN模式关闭 1: LIN模式开启</p>

22.5.2.9 接收超时寄存器 (UART_RTOR)

接收超时寄存器 (UART_RTOR)																																
偏移地址: 0x20																																
复位值: 0x0000 00FF																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved							RTOEN	RTO																								

—	Bit 31-25	—	—
RTOEN	Bit 24	R/W	<p>接收器超时开启位</p> <p>此位由软件设置1和清除。</p> <p>0: 接收器超时关闭</p> <p>1: 接收器超时开启</p>
RTO	Bit 23-0	R/W	<p>接收器超时数值</p> <p>设置接收超时时间，使用波特率时钟的字长为单 位。</p> <p>在标准模式下，接收最后一个字节后，在超时时间 内未检测到新的起始位，将 RIF 寄存器的 RXTO 位设置为 1，此位由软件设置和清除。</p> <p>在智能卡模式下，这个数值是用来实现 CWT 和 BWT。</p> <p>注：UART3/4 只支持 8 位</p>

22.5.2.10 状态寄存器 (UART_STAT)

状态寄存器 (UART_STAT)																																	
偏移地址: 0x28																																	
复位值: 0x0001 0008																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved														TFOERR	Reserved	TFEMPTY	Reserved	TSBUSY	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	RSBUSY	Reserved					CTSSTA	BKERR	FERR	PERR

—	Bit 31-19	—	—
TFOERR	Bit 18	R	<p>发送器溢出错误</p> <p>当发送器内已有数据时, 有新数据再次写入TX中时, 此位由硬件设置为1并舍弃新数据, 在发送数据或读取UART_STAT寄存器后清除</p> <p>0: 发送器溢出错误未产生</p> <p>1: 发送器溢出错误产生</p>
—	Bit 17	—	—
TFEMPTY	Bit 16	R	<p>发送器空</p> <p>当发送器内无任何数据时, 此位由硬件设置为1, 在TX写入数据时清除。</p> <p>0: 发送器有数据</p> <p>1: 发送器无数据</p>
—	Bit 15	—	—
TSBUSY	Bit 14	R	<p>发送器移位寄存器忙碌</p> <p>当写入数据由硬件设置为1, 在发送最后一个数据完成后清除。</p> <p>0: 发送器内无数据等待传送</p> <p>1: 发送器内有数据等待传送且未发送完最后一个数据</p>
RFUERR	Bit 13	R	<p>接收器下溢错误</p> <p>当接收器无数据时, 又再次读取接收器时, 由硬件设置为1, 在接收数据或读取UART_STAT寄存器后清除</p> <p>0: 接收器下溢错误未产生</p> <p>1: 接收器下溢错误产生</p>
RFOERR	Bit 12	R	<p>接收器溢位错误</p> <p>当接收器内已有数据时, 有新数据再次接收时, 此位由硬件设置为1并舍弃新数据, 在读取数据或读取</p>

			UART_STAT寄存器后清除 0: 接收器溢出错误未产生 1: 接收器溢出错误产生
—	Bit 11	—	—
RFNEMPTY	Bit 10	R	接收器非空 当接收器内有1笔数据时，此位由硬件设置为1，接收数据时清除。 0: 接收器无数据 1: 接收器有数据
—	Bit 9	—	—
RSBUSY	Bit 8	R	接收移位寄存器忙碌 当接收数据时，由硬件设置为1，在完成接收数据后清除 0: 接收器未接收数据 1: 接收器正在接收数据
—	Bit 7-4	—	—
CTSSTA	Bit 3	R	CTS状态 此位显示CTS输入引脚状态，由硬件设置为1和清除。 0: CTS输入引脚为0 1: CTS输入引脚为1
BKERR	Bit 2	R	断路错误 当接收数据与停止位皆为0时，由硬件设置为1。此位为显示当前读取接收器数值状态。 0: 断路错误未产生 1: 断路错误产生
FERR	Bit 1	R	帧错误 当接收数据的停止位为0时，由硬件设置为1。此位为显示当前读取接收器数值。 0: 帧错误未产生 1: 帧错误产生
PERR	Bit 0	R	校验错误 当接收数据的校验位接收错误时，由硬件设置为1。此位为显示当前读取接收器数值。 0: 校验错误未产生 1: 校验错误产生

22.5.2.11 中断开启寄存器 (UART_IER)

中断开启寄存器 (UART_IER)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOERR	Reserved	TFEMPTY	Reserved	TBC	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOERR	Bit 18	W1	开启发送器溢出中断功能 0: 写0无效 1: 使能
—	Bit 17	—	—
TFEMPTY	Bit 16	W1	开启发送器空中断功能 0: 写0无效 1: 使能
—	Bit 15	—	—
TBC	Bit 14	W1	开启发送数据完成中断功能 0: 写0无效 1: 使能
RFUERR	Bit 13	W1	开启接收器下溢中断功能 0: 写0无效 1: 使能
RFOERR	Bit 12	W1	开启接收器溢出中断功能 0: 写0无效 1: 使能
—	Bit 11	—	—
RFNEMPTY	Bit 10	W1	开启接收器非空中断功能 0: 写0无效 1: 使能
—	Bit 9	—	—
NOISE	Bit 8	W1	开启侦测噪声位中断功能 0: 写0无效 1: 使能
EOB	Bit 7	W1	开启块结束中断功能 0: 写0无效

			1: 使能
LINBK	Bit 6	W1	开启侦测LIN断路中断功能 0: 写0无效 1: 使能
ADDRM	Bit 5	W1	开启地址匹配中断功能 此位设置时, 开启中断功能, 硬件侦测地址匹配事件时发生中断
RXTO	Bit 4	W1	开启接收超时中断功能 0: 写0无效 1: 使能
DCTS	Bit 3	W1	开启CTS引脚电平中断功能 0: 写0无效 1: 使能
ABTO	Bit 2	W1	开启侦测自动波特率超时中断功能 0: 写0无效 1: 使能
ABEND	Bit 1	W1	开启侦测自动波特率结束中断功能 0: 写0无效 1: 使能
RXBERR	Bit 0	W1	开启接收器数据字节格式错误中断功能 0: 写0无效 1: 使能

22.5.2.12 中断关闭寄存器 (UART_IDR)

中断关闭寄存器 (UART_IDR)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOVER	Reserved	TFEMPTY	Reserved	TBC	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOVER	Bit 18	W1	关闭发送器溢出中断功能 0: 写0无效 1: 关闭
—	Bit 17	—	—
TFEMPTY	Bit 16	W1	关闭发送器空中断功能 0: 写0无效 1: 关闭
—	Bit 15	—	—
TBC	Bit 14	W1	关闭发送数据完成中断功能 0: 写0无效 1: 关闭
RFUERR	Bit 13	W1	关闭接收器下溢中断功能 0: 写0无效 1: 关闭
RFOERR	Bit 12	W1	关闭接收器溢出中断功能 0: 写0无效 1: 关闭
—	Bit 11	—	—
RFNEMPTY	Bit 10	W1	关闭接收器非空中断功能 0: 写0无效 1: 关闭
—	Bit 9	—	—
NOISE	Bit 8	W1	关闭侦测噪声位中断功能 0: 写0无效 1: 关闭
EOB	Bit 7	W1	关闭块结束中断功能 0: 写0无效

			1: 关闭
LINBK	Bit 6	W1	关闭侦测LIN断路中断功能 0: 写0无效 1: 关闭
ADDRM	Bit 5	W1	关闭地址匹配中断功能 0: 写0无效 1: 关闭
RXTO	Bit 4	W1	关闭接收超时中断功能 0: 写0无效 1: 关闭
DCTS	Bit 3	W1	关闭CTS引脚电平中断功能 0: 写0无效 1: 关闭
ABTO	Bit 2	W1	关闭侦测自动波特率超时中断功能 0: 写0无效 1: 关闭
ABEND	Bit 1	W1	关闭侦测自动波特率结束中断功能 0: 写0无效 1: 关闭
RXBERR	Bit 0	W1	关闭接收器字节格式错误中断功能 0: 写0无效 1: 关闭

22.5.2.13 中断功能有效状态寄存器 (UART_IVS)

中断功能有效状态寄存器 (UART_IVS)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOVER	Reserved	TFEMPTY	Reserved	TBC	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOVER	Bit 18	R	发送器溢出中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 17	—	—
TFEMPTY	Bit 16	R	发送器空中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 15	—	—
TBC	Bit 14	R	发送数据完成中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
RFUERR	Bit 13	R	接收器下溢中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
RFOERR	Bit 12	R	接收器溢出中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 11	—	—
RFNEMPTY	Bit 10	R	接收器非空中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
—	Bit 9	—	—
NOISE	Bit 8	R	侦测噪声位中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
EOB	Bit 7	R	块结束中断功能状态 0: 中断功能处于关闭状态

			1: 中断功能处于开启状态
LINBK	Bit 6	R	侦测LIN断路中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
ADDRM	Bit 5	R	地址匹配中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
RXTO	Bit 4	R	接收超时中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
DCTS	Bit 3	R	CTS引脚电平中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
ABTO	Bit 2	R	侦测自动波特率超时中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
ABEND	Bit 1	R	侦测自动波特率结束中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态
RXBERR	Bit 0	R	接收器字节格式错误中断功能状态 0: 中断功能处于关闭状态 1: 中断功能处于开启状态

注: UART_IVS 寄存器, 是实时反映系统配置 UART_IER 与 UART_IDR 的中断开启状态。

22.5.2.14 原始中断状态寄存器 (UART_RIF)

原始中断状态寄存器 (UART_RIF)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOVER	Reserved	TFEMPTY	Reserved	TBC	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOVER	Bit 18	R	发送器溢出, 原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 17	—	—
TFEMPTY	Bit 16	R	发送器空, 原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 15	—	—
TBC	Bit 14	R	发送数据完成, 原始中断状态 0: 无发生中断 1: 已发生中断
RFUERR	Bit 13	R	接收器下溢, 原始中断状态 0: 无发生中断 1: 已发生中断
RFOERR	Bit 12	R	接收器溢出, 原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 11	—	—
RFNEMPTY	Bit 10	R	接收器非空, 原始中断状态 0: 无发生中断 1: 已发生中断
—	Bit 9	—	—
NOISE	Bit 8	R	侦测噪声位, 原始中断状态 0: 无发生中断 1: 已发生中断
EOB	Bit 7	R	块结束, 原始中断状态 0: 无发生中断

			1: 已发生中断
LINBK	Bit 6	R	侦测LIN断路, 原始中断状态 0: 无发生中断 1: 已发生中断
ADDRM	Bit 5	R	地址匹配, 原始中断状态 0: 无发生中断 1: 已发生中断
RXTO	Bit 4	R	接收超时, 原始中断状态 0: 无发生中断 1: 已发生中断
DCTS	Bit 3	R	CTS引脚电平, 原始中断状态 0: 无发生中断 1: 已发生中断
ABTO	Bit 2	R	侦测自动波特率超时, 原始中断状态 0: 无发生中断 1: 已发生中断
ABEND	Bit 1	R	侦测自动波特率, 原始中断状态 0: 无发生中断 1: 已发生中断
RXBERR	Bit 0	R	接收器字节格式错误, 原始中断状态 0: 无发生中断 1: 已发生中断

22.5.2.15 中断屏蔽后状态寄存器 (UART_IFM)

中断屏蔽后状态寄存器 (UART_IFM)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOVER	Reserved	TFEMPTY	Reserved	TBC	RFUERR	RFOERR	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOVER	Bit 18	R	发送器溢出, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
—	Bit 17	—	—
TFEMPTY	Bit 16	R	发送器空, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
—	Bit 15	—	—
TBC	Bit 14	R	发送数据完成, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
RFUERR	Bit 13	R	接收器下溢, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
RFOERR	Bit 12	R	接收器溢出, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
—	Bit 11	—	—
RFNEMPTY	Bit 10	R	接收器非空, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
—	Bit 9	—	—
NOISE	Bit 8	R	侦测噪声位, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
EOB	Bit 7	R	块结束, 屏蔽后中断状态 0: 无发生中断

			1: 已发生中断
LINBK	Bit 6	R	侦测LIN断路, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
ADDRM	Bit 5	R	地址匹配, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
RXTO	Bit 4	R	接收超时, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
DCTS	Bit 3	R	CTS引脚电平, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
ABTO	Bit 2	R	侦测自动波特率超时, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
ABEND	Bit 1	R	侦测自动波特率, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断
RXBERR	Bit 0	R	接收器字节格式错误, 屏蔽后中断状态 0: 无发生中断 1: 已发生中断

注: UART_IFM 寄存器, 是滤除已关闭中断功能的中断事件, 只关注开启中断功能的事件。此寄存器状态是将 UART_RIF 与 UART_IVS 进行硬件逻辑与运算。

22.5.2.16 中断清除寄存器 (UART_ICR)

中断清除寄存器 (UART_ICR)																															
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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TFOVER	Reserved	TFEMPTY	Reserved	TBC	RFUERC_W1	RFOERC_W1	Reserved	RFNEMPTY	Reserved	NOISE	EOB	LINBK	ADDRM	RXTO	DCTS	ABTO	ABEND	RXBERR

—	Bit 31-19	—	—
TFOVER	Bit 18	C_W1	清除发送器溢出中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
—	Bit 17	—	—
TFEMPTY	Bit 16	C_W1	清除发送器空中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
—	Bit 15	—	—
TBC	Bit 14	C_W1	清除发送数据完成中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
RFUERC_W1	Bit 13	C_W1	清除接收器下溢中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
RFOERC_W1	Bit 12	C_W1	清除接收器溢出中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
—	Bit 11	—	—
RFNEMPTY	Bit 10	C_W1	清除接收器非空中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
—	Bit 9	—	—
NOISE	Bit 8	C_W1	清除侦测噪声位中断状态 此位设置时, 清除中断状态 (UART_RIF与 UART_IFM)
EOB	Bit 7	C_W1	清除块结束中断状态 此位设置时, 清除中断状态 (UART_RIF与

			UART_IFM)
LINBK	Bit 6	C_W1	清除侦测LIN断路中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
ADDRM	Bit 5	C_W1	清除地址匹配中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
RXTO	Bit 4	C_W1	清除接收超时中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
DCTS	Bit 3	C_W1	清除CTS引脚电平中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
ABTO	Bit 2	C_W1	清除侦测自动波特率超时中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
ABEND	Bit 1	C_W1	清除侦测自动波特率结束中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）
RXBERR	Bit 0	C_W1	清除接收器字节格式错误中断状态 此位设置时，清除中断状态（UART_RIF与UART_IFM）

注：UART_ICR 寄存器设置时，将清除 UART_RIF 与 UART_IFM 中断标志状态；此设置不影响中断 UART_IER、UART_IDR 与 UART_IVS 寄存器，只清除标志状态 UART_RIF 与 UART_IFM。

第23章 通用串行总线（USB）

23.1 概述

该USB控制器作为全速USB设备的控制器，兼容USB2.0协议规范中全速数据传输（12Mbps）和嵌入式设备（On-The-GO）标准。支持点对点通信时工作于主机或设备两种模式。

支持会话请求协议SRP（Session Request Protocol）和主机协商协议HNP（Host Negotiation Protocol）通信协议；支持4种数据传输类型：控制传输/同步传输/中断传输/批量传输；支持DMA对端点FIFO的访问。

23.2 特性

- ◆ USB设备控制器支持全速（12 Mbps）和低速（1.5 Mbps）的数据传输模式
- ◆ 注意：在主机模式，支持所有通信速度。然而在设备模式下，只有全速通信支持
- ◆ 支持点对点通信时工作于主机或设备两种模式
- ◆ 兼容USB2.0协议规范中全数据传输（12 Mbps）和嵌入式设备（On-The-GO）标准
- ◆ 支持在OTG模式下与一个全速/低速设备通信
- ◆ 支持会话请求协议SRP（Session Request Protocol）和主机协商协议HNP（Host Negotiation Protocol）
- ◆ 支持4种数据传输类型：控制传输/同步传输/中断传输/批量传输
- ◆ 支持挂起状态和恢复功能
- ◆ USB设备模式下支持软连接和断开功能
- ◆ 内建动态配置512B SRAM端点FIFO
- ◆ 支持9个端点，端点大小最大支持512字节数据：
 - ◇ EP0IN/OUT：仅支持控制传输
 - ◇ EP1IN~EP4IN（Rx Endpoints）：支持同步传输、中断传输、批量传输
 - ◇ EP1OUT~EP4OUT（Tx Endpoints）：支持同步传输、中断传输、批量传输
- ◆ 支持使用DMA对端点FIFO进行访问

23.3 结构图

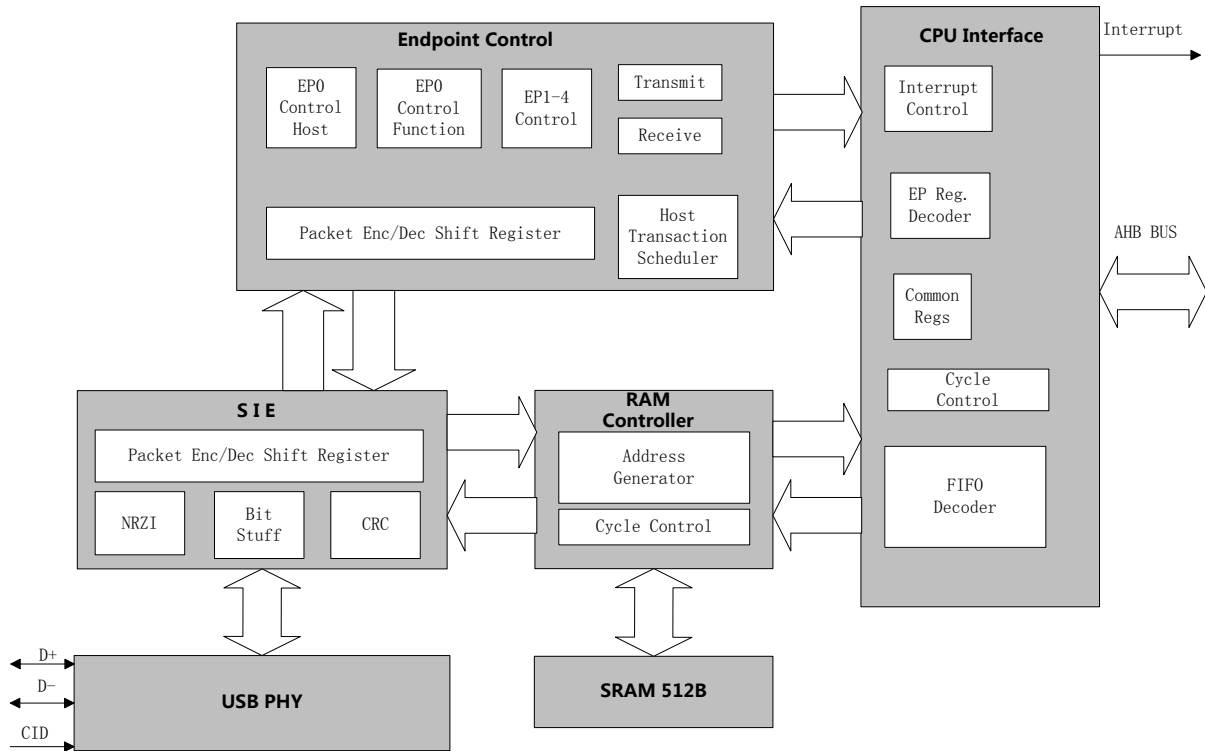


图 23-1 USB 结构图

23.4 功能描述

23.4.1 操作模式

USB 控制器有两种主要的工作模式：USB 设备模式和 USB 主机模式。

当 USB 控制器工作于设备模式，USB 控制器完成 USB 数据包的发送接收以及数据的编码、解码、校验等工作。对于 IN 事务的处理是通过发送端点（EP0OUT/EP1OUT~EP4OUT）完成，OUT 事务的处理是通过接收端点（EP0IN/EP1IN~EP4IN）完成。可支持控制传输、同步传输、中断传输、批量传输四种数据传输方式。

当 USB 控制器工作于主机模式，与另一个 USB 设备进行点对点通信时，USB 控制器可支持控制传输、同步传输、中断传输、批量传输。对于 IN 事务的处理是通过接收端点（EP0IN/EP1IN~EP4IN）完成，OUT 事务的处理是通过发送端点（EP0OUT/EP1OUT~EP4OUT）完成。USB 控制器完成 USB 数据包的发送接收以及数据的编码、解码、校验等工作，同时对于同步端点和中断端点，USB 控制器会自动根据端点的配置在 USB 帧/小帧中安排事务传输。

USB 控制器初始连接是工作于主机模式还是工作于设备模式，是由 CID 引脚电平决定。CID 引脚为低电平时，表示工作于主机模式（A 类设备），CID 引脚为高电平时，表示工作于设备模式（B 类设备）。当 USB 控制器为 B 类设备工作于设备模式时，检测到 USB 总线处于空闲状态时，软件可以配置 USB_DEVCON 寄存器的控制位 HOSTREQ 请求工作于主机模式。

注：在 USB 模块复位后（包括芯片系统复位和 USB 外设模块复位 USBRST），USB 模块硬件电路会进行初始化操作，软件需要等待至少 20us 后，再进行 USB 模块寄存器的读写操作，以及启动 USB 模块工作。

23.4.2 设备模式

在设备模式下，IN 事务由端点传输接口控制，并使用发送端点完成。OUT 事务使用端点接收接口控制，并使用接收端点完成。要注意端点 FIFO 大小要适合端点最大数据包的大小。

批量传输

批量传输端点的 FIFO 必须支持最大数据包的大小（最多 64 个字节）或最大值的两倍封包大小（如果使用双缓冲）。

中断传输

中断传输端点的 FIFO 必须支持最大数据包的大小（最多 64 个字节）或最大值的两倍封包大小（如果使用双缓冲）。

同步传输

同步传输端点的 FIFO 比较灵活，最大可支持 511 字节。

控制传输

通常，USB 设备应使用 USB 控制器端点 0 作为专用控制端点。

23.4.2.1 端点控制器

USB 控制器支持 9 个端点：EP0IN/OUT、EP1IN~EP4IN、EP1OUT~EP4OUT。EP0 端点仅支持控制传输，其他端点同时支持同步传输、中断传输、批量传输。各个端点的 FIFO 大小是可以动态配置的。端点特性如下表所示。

端点 0 是专用的控制端点，用于在枚举过程中所有的对端点 0 的控制传输事务，或者是对端点 0 的任意其它的控制请求。

配对的 IN 端点和 OUT 端点可以配置成不同的类型。比如 OUT 端点配置成批量端点，而该端点对应的 IN 端点可以配置成中断端点。

端点号	端点类型	双缓冲
EP0 (IN/OUT)	控制	无
EP1 (IN/OUT)	同步、中断、批量	有
EP2 (IN/OUT)	同步、中断、批量	有
EP3 (IN/OUT)	同步、中断、批量	有
EP4 (IN/OUT)	同步、中断、批量	有

表 23-1 端点特性

注 1: Tx Endpoints 对应端点 EP* OUT, Rx Endpoints 对应端点 EP* IN, 都是相对于主机而言。
 注 2: USB 控制器工作于设备模式时, IN 事务处理通过 Tx Endpoints, OUT 事务处理通过 Rx Endpoints。
 注 3: USB 控制器工作于主机模式时, IN 事务处理通过 Rx Endpoints, OUT 事务处理通过 Tx Endpoints。

23.4.2.2 设备模式下的IN传输

IN 传输的数据通过 OUT 端点的 FIFO 处理发送。

单缓冲

如果发送端点的 FIFO 大小小于该端点最大数据包长（由 USB_TXMAXP 寄存器设定，或由 USB_TXFIFO1 和 USB_TXFIFO2 寄存器设定）的两倍，只有一个数据包可以被缓冲到 FIFO 中。当每个数据包加载到发送 FIFO 中时，USB_CSR0L_TXCSRL 寄存器的 TXRDY 位必须置 1。

如果 USB_CSR0H_TXCSRH 寄存器的 AUTOSET 位为 1，当最大大小的数据包被加载到 FIFO 中时，TXRDY 位会被自动置 1。对于包长小于最大包长的数据包，TXRDY 位必须手动置 1。

当 TXRDY 位被置 1 时，数据包已准备好发送，且 FIFONE 也置位。当数据包成功发送后，TXRDY 和 FIFONE 位会被自动清零，并产生相应的发送端点中断。此时下一个数据包可以被加载到 FIFO 中。

双缓冲

如果发送端点的 FIFO 大小至少是此端点的最大包长（由 USB_TXMAXP 寄存器设定，或由 USB_TXFIFO1 和 USB_TXFIFO2 寄存器设定）的两倍时，这时允许双缓冲，即两个数据包可

以同时缓存在 FIFO 中。当每个数据包加载到发送 FIFO 中时，USB_CSR0L_TXCSRL 寄存器的 TXRDY 位必须置 1。

如果 USB_CSR0H_TXCSRH 寄存器的 AUTOSET 位为 1，当最大包长的数据包被加载到 FIFO 中时，TXRDY 位会被自动置 1。对于小于最大包长的数据包或高带宽同步传输/中断传输的场合，TXRDY 位必须手动置 1。

当 TXRDY 位被置 1 时，数据包已准备好发送，且 FIFONE 也置位。当第一个数据包加载后，TXRDY 位会被清零，并产生相应的发送端点中断。此时第二个数据包可以被加载到 FIFO 中，TXRDY 位再次被置 1。这时，二个数据包都已准备好被发送。当二个数据包都已成功发送后，TXRDY 位会被清零，并产生相应的发送端点中断。此时下一个数据包可以被加载到 FIFO 中。

需要注意的是，如果要使用双包缓存发送必须设置 USB_TXFIFO2 寄存器的 DPB 位为 1。

23.4.2.3 设备模式下的OUT 传输

在设备模式下 OUT 事务传输通过 USB 控制器的接收 FIFO 处理。当双缓冲使能时，两个数据包都可以缓存到 FIFO 中。当双缓冲不使能时，只有一个数据包可以缓存到 FIFO 中。

单缓冲

如果接收端点的 FIFO 大小小于该端点最大包长的两倍时，只能使用单缓冲，只有一个数据包可以被缓冲到 FIFO 中。

当一个数据包接收并缓存到 FIFO 中时，USB_RXCSRL 寄存器的 RXRDY 位和 FULL 位都会被置 1，可以从 FIFO 中读取接收到的数据。当接收的数据从 FIFO 中读取后，软件必须将 RXRDY 位清零，以便允许接收更多数据包。此操作还会产生 ACK 信号给到主机控制器。

当 USB_RXCSRH 寄存器的 AUTOCLR 位被置 1 时，最大包长的数据包被从 FIFO 中取出后，RXRDY 位和 FULL 位会被自动清零。如果数据包不是最大包长，RXRDY 位必须手动清零。

双缓冲

如果接收端点的 FIFO 大小至少是此端点最大包大小的两倍，这时允许双缓冲，则两个数据包可以缓存在 FIFO 中。

当第一个数据包被接收并缓存在 FIFO 中后，USB_RXCSRL 寄存器的 RXRDY 位被置 1，相应的接收端点中断会产生，可以从 FIFO 中读取接收到的数据。需要注意的是，当第一个数据包被接收到后，USB_RXCSRL 寄存器的 FULL 位不会被置 1。只有当第二个数据包被接收到后，USB_RXCSRL 寄存器的 FULL 位才会被置 1。当所有数据包都从 FIFO 中取出后，软件必须将 RXRDY 位清零，以便允许接收更多数据包。

如果 USB_RXCSRH 寄存器的 AUTOCLR 位为 1，当最大包长的数据包被从 FIFO 中读出后，RXRDY 位会被自动清零。对于小于最大包长的数据包，RXRDY 位必须手动清零。

需要注意的是，如果要使用双包缓存接收，必须设置 USB_RXFIFO2 寄存器的 DPB 位为 1。

23.4.2.4 调度

设备控制器不能控制传输事务的调度，因为调度由主机控制器决定。USB 控制器可以随时建立传输事务。设备控制器等待来自主机控制器的请求，并在事务完成时产生中断。如果主机控制器发出请求，而设备控制器没有准备好，设备控制器向所有请求发送 NACK 响应，直到它准备就绪。

23.4.2.5 其他操作

USB 控制器自动响应 USB 总线的某些状态或来自主机控制器的操作。例如当 USB 控制器自动停止控制传输或收到意外的零长度 OUT 数据包。

暂停控制传输

USB 控制器检测到下列这些情况会自动发出暂停 (STALL) 控制传输信号，情况如下：

- ◆ 在控制传输的输出数据过程中，主机发送的数据比建立过程中设备请求的数据多。当设备接收到最后一个输出数据包，并置位 USB_CSR0L_TXCSRL 寄存器的 DATAEND 位后，主机发送了输出令牌包（应该为输入令牌包）。
- ◆ 在控制传输的输入数据过程中，主机请求的数据比建立过程中设备请求的数据多。当控制器收到最后一个数据包时，将 USB_CSR0L_TXCSRL 寄存器的 TXRDY 位清零、USB_CSR0L_TXCSRL 寄存器的 DATAEND 位置 1 来指示当前是最后一个数据包，等待主机发出 ACK 信号。主机发送了输入令牌包（应该为输出令牌包）。
- ◆ 主机发送大于 USB_RXMAXP 寄存器所配置的数据长度。
- ◆ 主机在输出阶段发送多于一个的零长度数据包。

零长度数据包

一个零长度的输出数据包用于指示控制传输的结束。在正常情况下，这个零长度数据包应该只在设备请求的全部数据已经传输完成后才会被接收。但是如果主机在设备请求的全部数据发送完成之前就发送零长度的 OUT 数据包，就表示此次数据传输提前结束。在这种情况下，USB 控制器自动清空 FIFO 中的 IN 令牌阶段收到的全部数据，并将 USB_CSR0L_TXCSRL 寄存器的 DATAEND 位置 1。

设备地址设置

当主机尝试枚举 USB 设备时，主机会请求设备将其地址从零改到其他值。通过将主机请求的值写入 USB_FADDR 寄存器来实现地址的更改。但是在写入时应该小心，避免在事务完成之前更改地址。只能在 SET_ADDRESS 命令完成后才能去设置 USB_FADDR 寄存器。在 SET_ADDRESS 命令下，设备在主机输入请求时发送零长度数据包来响应主机，表明 SET_ADDRESS 命令已完成。设备一旦响应输入请求，USB_FADDR 寄存器需要立刻改写为新的值，以避免丢失发送到新地址的任何新命令。

23.4.2.6 设备模式挂起

当 USB 总线空闲超过 3ms，USB 控制器自动进入挂起模式，这时会立即产生挂起中断（挂起中断已使能）。当处于挂起模式时，USB 的物理层也会进入挂起模式。当检测到恢复信号时，USB 控制器退出挂起模式。这时会产生恢复中断。通过设置 USB_POWER 寄存器的 RESUME 位，也可以强制 USB 控制器退出挂起模式。当该位置 1 时，USB 控制器退出挂起模式，并将恢复信号驱动到总线上。RESUME 位必须在 10ms（最多 15ms）之后清零，以结束恢复信号。

23.4.2.7 起始帧

当 USB 控制器工作在设备模式下时，它会接收到一个来至主机的起始帧数据包（SOF），起始帧数据包每隔 1ms 发送一次。当接收到起始帧数据包时，数据包中包含的 11 位帧编号被写入 USB_FRAME1 寄存器与 USB_FRAME2 寄存器，并且会产生一个 SOF 中断。一旦 USB 控制器已经开始接收起始帧数据包，它每 1ms 就需要成功接收到一次。如果在 1.00358ms 内没有接收到起始帧数据包，则认为该数据包已丢失，并且 USB_FRAME1 寄存器与 USB_FRAME2 寄存器不会被更新。当 SOF 包重新成功接收时，USB 控制器继续工作，并重新同步这个脉冲。

23.4.2.8 USB 复位

当 USB 控制器工作在设备模式下时，在 USB 总线检测到满足复位条件时，USB 控制器自动执行以下操作：

- ◆ 清零 USB_FADDR 寄存器
- ◆ 清零 USB_INDEX 寄存器
- ◆ 清空全部端点的 FIFO
- ◆ 清零全部控制状态寄存器
- ◆ 使能全部端点中断
- ◆ 产生复位中断

23.4.2.9 连接和断开

USB 控制器与 USB 总线的连接由软件处理。通过将 USB_DPDMCON 寄存器中的 PHYPWREN 位置 0，可以将 USB 物理层在正常模式和非驱动模式之间切换。USB 控制器的默认状态是非驱动模式。

当 USB 物理层处在正常模式下时，同时 USBDP 和 USBDM 差分线是使能的，这时和其他设备是连接的。当 USB 物理层处在非驱动模式下时，USBDP 和 USBDM 差分线为高阻态，这时和其他设备是断开连结的。

23.4.3 主机模式

当 USB 控制器在主机模式下运行时，它可以与另一个 USB 设备进行点对点通信，并且同时支持全速、低速传输。主机模式支持控制、批量、同步和中断传输。

当处于主机模式时，输入事务由端点的接收接口控制。所有输入事务使用接收端点寄存器，并且所有输出事务使用端点的发送端点寄存器。端点的 FIFO 应考虑端点最大数据包的大小。

批量传输

批量传输端点的 FIFO 必须支持最大数据包的大小（最多 64 个字节）或最大值的两倍封包大小（如果使用双缓冲）。

中断传输

中断传输端点的 FIFO 必须支持最大数据包的大小（最多 64 个字节）或最大值的两倍封包大小（如果使用双缓冲）。

同步传输

同步传输端点的 FIFO 比较灵活，最大可支持 1023 字节。

控制传输

控制传输可以指定单独的控制端点与设备通信。在大多数情况下，USB 控制器应使用专用控制端点 0 作为专用的控制端点与设备的端点 0 进行通信。

23.4.3.1 端点控制器

端点控制器用于控制 USB 端点接口，通过这个接口可与连接的设备进行通信。端点由一个专用控制输入与输出端点、6 个输出端点和 6 个输入端点组成。

专用的控制接口只能用于与设备的端点 0 之间的控制传输。他们用于设备枚举或其他使用设备端点的控制功能。控制端点的输入和输出事务共享 USB 控制器 FIFO 内存的前 64 字节。其余输入和输出接口可配置为与控制端点、批量端点、中断端点或同步端点通信。

这些 USB 接口可同时调度，用于与任何设备的任何端点的 6 个独立的输出事务和 6 个独立的输入事务。输入和输出控制有成对的寄存器。通过配置后它们可以与不同类型的端点以及不同设备的不同端点进行通讯。例如，第一对端点可分开控制，输出部分与设备的批量输出端点 1 通信，同时输入部分与设备的中断输出端点 2 通信。

在访问设备之前，必须设置 USB_FADDR 寄存器。

23.4.3.2 主机模式下的IN 传输

输入事务的处理，采用与设备模式处理输出事务类似的方式，但传输事务必须通过设置 USB_CSR0L_TXCSRL 寄存器中的 REQPKT 位开始，向事务调度表明此端点存在一个正在运行的传输。此时事务调度向设备发送一个输入令牌包。

当接收到数据包且存到相应接收 FIFO 中时，USB_CSR0L_TXCSRL 寄存器中的 RXRDY 位置 1，同时产生相应的接收端点中断信号，指示有一个数据包需要从 FIFO 中读出。

当数据包被读出时，必须将 RXRDY 位清零。USB_RXCSRH 寄存器中的 AUTOCLR 位可用于当最大包长的数据包从 FIFO 中读出时，将 RXRDY 位自动清零。

如果设备用 NACK 响应批量或中断传输的输入令牌，USB 主机控制器将重试，直到达到设置的 NACK 限制次数。如果目标设备用 STALL 响应，USB 主机将不重试传输，而将 USB_RXCSRL 寄存器中的 STALLED 位置位来产生中断。如果目标设备在规定的时间内不响应输入令牌包，

或者包存在 CRC 或位填充错误，USB 主机将重试传输。如果三次重试，目标设备仍无响应，USB 控制器将 REQPKT 位清零，将 USB_RXCSRL 寄存器中的 ERROR 位置 1 来产生中断。

23.4.3.3 主机模式下的OUT 传输

当数据包装载到发送 FIFO 中时，USB_CSR0L_TXCSRL 寄存器中的 TXRDY 位必须置 1。如果将 USB_CSR0H_TXCSRH 寄存器中的 AUTOSSET 位置 1，当最大包长的数据包装载到 FIFO 中时 TXRDY 位自动置 1。

如果目标设备用 NACK 响应输出令牌包，USB 主机控制器将重试，直到达到设置的 NACK 限制次数。如果目标设备用 STALL 响应输出令牌包，USB 主机将不重试传输，而通过将 USB_CSR0L_TXCSRL 寄存器中的 STALLED 位置 1 来中断主处理器。如果目标设备在需要的时间内不响应输出令牌包，或者存在 CRC 或位填充错误，USB 主机将重试传输。如果三次重试，目标设备仍无响应，USB 控制器将清空 FIFO，并将 USB_CSR0L_TXCSRL 寄存器的 ERROR 位置 1。

23.4.3.4 事务调度

事务调度由 USB 主机控制器自动处理。主机控制器会根据端点事务类型配置端点通讯调度。中断传输可以是每 1 帧进行一次，也可以每 255 帧进行一次，可以在 1 帧到 255 帧之间以 1 帧增量调度。批量端点不处理调度参数，但在设备的端点不响应时，允许 NAK 超时。同步端点可以在每帧到每 216 帧之间调度。

USB 控制器维持帧计数。如果目标设备为全速设备，控制器在每帧开始时自动发送 SOF 包，同时帧计数加 1。如果目标设备为低速设备，将在总线上发送 K 状态来保持总线活动，防止低速设备进入挂起模式。

在 SOF 包发送后，USB 主机控制器应检查所有配置的端点，寻找激活的传输事务。REQPKT 位置 1 的接收端点或 TXRDY 位置 1 的发送端点，被视为存在激活的传输事务。

如果传输建立在一帧的第一个调度周期，而且端点的间隔计数器减到 0，则同步传输和中断传输开始。所以每个端点的中断传输和同步传输每 x 帧才发生一次，其中 x 是通过 USB 端点 x 主机发送间隔（USB_NAKLIMIT0_TXINTERVAL）或 USB 端点 x 主机接收间隔（USB_RXINTERVAL）寄存器设置的间隔。

如果在帧中下一个 SOF 包之前有足够的时间完成传输，则激活的批量传输立即开始。如果传输需要重发时（例如，收到 NACK 或设备未响应），需要在调度器先检查完其他所有端点是否有其它激活的传输之后，传输才能重传。这保证了一个发送大量 NACK 响应的端点不阻塞总线上的其他传输正常进行。控制器同样允许使用者设置的目标设备端点发送 NACK 的超时限制。

23.4.3.5 干扰

只有总线至少空闲一个最小间隔包的时间，USB 主机控制器才会开始传输。USB 控制器不会发起事务传输，除非它能在结束帧前完成。如果在结束帧时 USB 总线上仍有活动，USB 主机将判定连接的目标设备发生故障，同时 USB 控制器挂起所有传输事务，并产生干扰（Babble）

中断信号。

23.4.3.6 主机挂起

如果 USB_POWER 寄存器中的 SUSPEND 位置 1, USB 主机控制器完成当前的传输事务, 然后停止事务调度和帧计数。此时, 不再启动事务传输, 不再产生 SOF 包。

要离开挂起模式, 可以将 RESUME 位置 1 并将 SUSPEND 位清零。当 RESUME 位置 1 时, USB 主机将在总线上产生恢复信号。但 20ms 之后, 必须将 RESUME 位清零, 此时, 帧计数和事务调度开始。主机支持远程恢复检测。

23.4.3.7 USB 复位

如果 USB_POWER 寄存器中的 RESET 位置 1, USB 主机控制器将在总线上产生 USB 复位信号。RESET 位需要保持置位至少 20ms, 以确保目标设备的正确复位。软件清除此位后, USB 主机控制器开始帧计数和事务调度。

23.4.3.8 连接/断开

通过将 USB_DEVCON 寄存器中的 SESSION 位置 1 来启动会话。当检测到设备时, 将产生连接中断信号。连接的设备的速度, 通过读 USB_DEVCON 寄存器来确定。如果 FSDEV 位置 1, 连接的设备为全速设备; 如果 LSDEV 位置 1, 连接的设备为低速设备。USB 控制器必须对设备发出一个复位信号, 此时 USB 主机开始设备沟通。如果会话过程中设备断开连接, 将产生断开中断。

23.4.3.9 OTG模式

OTG 控制器通过物理层采样 CID 输入来决定哪个是 A 设备哪个是 B 设备。CID 信号拉低时, 检测到输入 A 设备 (表示 OTG 控制器作为 A 设备); CID 信号为高时, 检测到输入 B 设备 (表示 OTG 控制器作为 B 设备)。注意当在 OTG A 和 OTG B 之间切换时, 控制器保留所有的寄存器内容。

23.4.3.10 开始会话

当 USB OTG 控制器准备开始会话时, USB_DEVCON 寄存器中的 SESSION 位必须置 1。此时 OTG 控制器使能 CID 引脚检测。当检测到 A 类型连接时, CID 输入为低; 当检测到 B 类型连接时, CID 输入为高。同时设置 USB_DEVCON 寄存器中的 CID 位, 来表明 USB OTG 控制器用作 A 设备还是 B 设备。

如果 USB OTG 控制器是 A 设备, 则它进入主机模式 (A 设备总是默认为主机)。此时, OTG 控制器等待外设接入。当检测到外设接入, 则产生一个连接中断信号, USB_DEVCON 寄存器中的 FSDEV 或 LSDEV 位置 1 (取决于接入的是全速设备还是低速设备)。这时, USB 控制器向接入的设备发送一个复位信号。可以通过将 USB_DEVCON 寄存器中的 SESSION 位清零来结束会话。如果发生干扰时 OTG 控制器将自动结束会话。

如果 OTG 控制器用作 B 设备, 它使用 USB OTG 规范中定义的会话请求协议来请求会话。会话结束时, SESSION 位可通过 OTG 控制器或应用软件清零。

23.4.3.11 主机协商

如果 USB 控制器是 A 设备，CID 信号为低，当会话发起时它将自动进入主机模式。如果 USB 控制器是 B 设备，CID 信号为高，当会话发起时它将自动进入设备状态。但是也可以通过软件方式将 USB_DEVCON 寄存器中的 HOSTREQ 位置 1 使 USB 控制器从设备模式变为主机模式。此位可以在通过将 USB_DEVCON 寄存器中的 SESSION 位置 1 发起会话请求的同时置位，也可以在发起请求之后任意时候置位。当 USB 控制器下次进入挂起模式时，如果 HOSTREQ 位保持置 1，控制器进入主机模式，并开始主机协商，引发物理层断开 D+ 线路上的上拉电阻，触发之前的 A 设备切换到设备模式，并连接设备模式的上拉电阻。当 USB 控制器检测到此情况，将产生连接中断信号，并将 USB_POWER 寄存器中的 RESET 位置 1 使之前的 A 设备复位。USB 控制器自动开始复位序列，确保复位在之前的 A 设备连接上拉电阻后的 1ms 内开始。主处理器应等待至少 20ms，然后将 RESET 位清零，开始枚举之前的 A 设备。

当 USB OTG 控制器 B 设备使用完总线，它将 USB_POWER 寄存器中的 SUSPEND 位置 1 进入挂起模式。A 设备检测此情况，则结束会话或恢复到主机模式。如果 A 设备是 OTG 控制器，将产生一个连接断开的中断信号。

23.5 特殊功能寄存器

23.5.1 寄存器列表

USB 寄存器列表		
名称	偏移地址	描述
USB_FADDR	0000 _H	USB 设备功能地址寄存器
USB_POWER	0001 _H	USB 电源管理寄存器
USB_DPDMDCON	0002 _H	USB DP/DM 控制器寄存器
USB_SWCID	0003 _H	USB 软件控制 CID 寄存器
USB_SWVBUS	0004 _H	USB 软件控制 VBUS 寄存器
USB_FRAME1	000C _H	USB 帧号寄存器 1
USB_FRAME2	000D _H	USB 帧号寄存器 2
USB_INDEX	000E _H	USB 端点索引寄存器
USB_DEVCON	000F _H	USB 设备控制寄存器
USB_TXMAXP	0010 _H	USB 发送最大数据包大小寄存器
USB_CSR0L_TXCSRSL	0011 _H	USB 发送端点中的控制状态寄存器 1
USB_CSR0H_TXCSRH	0012 _H	USB 发送端点中的控制状态寄存器 2
USB_RXMAXP	0013 _H	USB 接收最大数据包大小寄存器
USB_RXCSRSL	0014 _H	USB 接收控制状态寄存器 1
USB_RXCSRH	0015 _H	USB 接收控制状态寄存器 2
USB_COUNT0_RX1	0016 _H	USB 接收端点中的接收字节数寄存器
USB_RXCOUNT2	0017 _H	USB 接收字节数寄存器 2
USB_TXTYPE	0018 _H	USB 发送协议类型寄存器
USB_NAKLIMIT0 _TXINTERVAL	0019 _H	USB 端点 NAK 响应超时或发送轮询间隔寄存器
USB_RXTYPE	001A _H	USB 接收协议类型寄存器
USB_RXINTERVAL	001B _H	USB 接收轮询间隔寄存器
USB_TXFIFO1	001C _H	USB 发送 FIFO 配置寄存器 1
USB_TXFIFO2	001D _H	USB 发送 FIFO 配置寄存器 2
USB_RXFIFO1	001E _H	USB 接收 FIFO 配置寄存器 1
USB_RXFIFO2	001F _H	USB 接收 FIFO 配置寄存器 2
USB_EP0FIFO	0020 _H	USB 端点 0 FIFO
USB_EP1FIFO	0024 _H	USB 端点 1 FIFO
USB_EP2FIFO	0028 _H	USB 端点 2 FIFO
USB_EP3FIFO	002C _H	USB 端点 3 FIFO
USB_EP4FIFO	0030 _H	USB 端点 4 FIFO
USB_TXIER	0080 _H	USB 发送中断使能寄存器

USB 寄存器列表		
名称	偏移地址	描述
USB_RXIER	0082 _H	USB 接收中断使能寄存器
USB_TXIDR	0084 _H	USB 发送中断禁用寄存器
USB_RXIDR	0086 _H	USB 接收中断禁用寄存器
USB_TXIVS	0088 _H	USB 发送中断使能状态寄存器
USB_RXIVS	008A _H	USB 接收中断使能状态寄存器
USB_TXRIF	008C _H	USB 发送原始中断事件标志寄存器
USB_RXRIF	008E _H	USB 接收原始中断事件标志寄存器
USB_TXIFM	0090 _H	USB 发送中断屏蔽标志寄存器
USB_RXIFM	0092 _H	USB 接收中断屏蔽标志寄存器
USB_TXICR	0094 _H	USB 发送中断清除寄存器
USB_RXICR	0096 _H	USB 接收中断清除寄存器
USB_IER	00A0 _H	USB 中断使能寄存器
USB_IDR	00A4 _H	USB 中断禁用寄存器
USB_IVS	00A8 _H	USB 中断使能状态寄存器
USB_RIF	00AC _H	USB 原始中断事件标志寄存器
USB_IFM	00B0 _H	USB 中断屏蔽标志寄存器
USB_ICR	00B4 _H	USB 中断清除寄存器

23.5.2 寄存器描述

23.5.2.1 USB设备功能地址寄存器 (USB_FADDR)

USB 设备功能地址寄存器 (USB_FADDR)							
偏移地址: 0x0000							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	ADDR						

—	Bit 7	—	—
ADDR	Bit 6-0	R/W	USB设备功能地址设置位 设备地址。

注 1: USB_FADDR 寄存器是一个 8 位寄存器，应该用事务的外设部分的 7 位地址来写。

注 2: 当 USB 在设备模式 (USB_DEVCON.HOST=0) 中使用时，这个寄存器应该使用通过 SET_ADDRESS 命令接收的地址来写入，然后 SET_ADDRESS 命令将用于解码后续令牌分组中的功能地址。

注 3: 当 USB 在主机模式下使用时 (USB_DEVCON.HOST=1)，这个寄存器应该被设置为在设备枚举期间，在 SET_ADDRESS 命令中发送的值作为外围设备的地址。

23. 5. 2. 2 USB电源管理寄存器 (USB_POWER)

主机模式

USB 电源管理寄存器 (USB_POWER)							
偏移地址: 0x001							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved				RESET	RESUME	SUSPEND	Reserved

—	Bit 7-4	—	—
RESET	Bit 3	R/W	复位信号 0: 结束总线上的 RESET 信号 1: 使能总线上的 RESET 信号
RESUME	Bit 2	R/W	恢复信号 0: 结束总线上的 RESUME 信号 1: 当设备处于挂起模式时, 使能 RESUME 信号 该位必须在被设置后维持 20ms, 由软件清除。
SUSPEND	Bit 1	R/W	挂起模式使能位 0: 正常工作模式 1: 启用 SUSPEND 模式 当 CPU 读取中断寄存器或将 RESUME 位置 1 或离开主机模式时, 该位将被清除。
—	Bit 0	—	—

设备模式

USB 设备功能地址寄存器 (USB_POWER)							
偏移地址: 0x001							
复位值: 0x00							
7	6	5	4	3	2	1	0
ISOUDT	Reserved			RESET	RESUME	SUSPEND	SUSPENDEN

ISOUDT	Bit 7	R/W	<p>同步端点更新控制位</p> <p>0: 没有效果</p> <p>1: 在发送数据包之前设置了 TXRDY 之后, USB 等待一个 SOF 令牌。如果在 SOF 令牌之前接收到 IN 令牌, 则将发送零长度数据包</p> <p>此位仅适用于同步传输。</p>
—	Bits 6-4	—	—
RESET	Bit 3	R	<p>复位信号</p> <p>0: 结束总线上的 RESET 信号</p> <p>1: 使能总线上的 RESET 信号</p>
RESUME	Bit 2	R/W	<p>恢复信号</p> <p>0: 结束总线上的 RESUME 信号</p> <p>1: 当设备处于挂起模式时, 使能 RESUME 信令。在设备模式时, 置 1 之后应该在 10ms(最大 15ms) 之后清零此控制位。</p>
SUSPEND	Bit 1	R	<p>挂起模式标志位</p> <p>0: 当软件读取中断寄存器或将上面的 RESUME 位置 1 时, 该位被清除</p> <p>1: USB 控制器处于挂起模式</p>
SUSPENDEN	Bit 0	R/W	<p>挂起模式使能位</p> <p>0: 当在总线上接收到挂起信号时禁止进入挂起模式</p> <p>1: 当在总线上接收到挂起信号时允许进入挂起模式</p>

注: 这个 8 位寄存器用于控制暂停和恢复信令, 用于同步端点的 IN 分组定时, 以及用于点对点通信, 指示连接的设备类型。

23. 5. 2. 3 USB DP/DM控制器寄存器 (USB_DPDMCON)

USB DP/DM 控制器寄存器 (USB_DPDMCON)							
偏移地址: 0x0002							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			DPPUD		DMPUD		PHYPWREN

—	Bits 7-5	—	—
DPPUD	Bits 4-3	R/W	USB_DP 信号引脚内部上拉/下拉选择位 00: 无上拉/下拉 01: 上拉 10: 下拉 11: 保留
DMPUD	Bits 2-1	R/W	USB_DM 信号引脚内部上拉/下拉选择位 00: 无上拉/下拉 01: 上拉 10: 下拉 11: 保留
PHYPWREN	Bit 0	R/W	USB PHY 电源使能位 0: 关闭内部 USB PHY 的电源 1: 开启内部 USB PHY 的电源

23.5.2.4 USB软件控制CID寄存器 (USB_SWCID)

USB 软件控制 CID 寄存器 (USB_SWCID)							
偏移地址: 0x0003							
复位值: 0x03							
7	6	5	4	3	2	1	0
Reserved						HOST	CIDCTRL

—	Bits 7-2	—	—
HOST	Bit 1	R/W	主机模式选择位 0: USB 控制器作为主机。 1: USB 控制器作为一个设备
CIDCTRL	Bit 0	R/W	连接 ID (CID) 信号控制位 0: 硬件控制 1: 软件控制

23. 5. 2. 5 USB软件控制VBUS寄存器 (USB_SWVBUS)

USB 软件控制 VBUS 寄存器 (USB_SWVBUS)							
偏移地址: 0x0004							
复位值: 0x0F							
7	6	5	4	3	2	1	0
Reserved				VALTH	SESVALTH	SESENDTH	SIGCTRL

—	Bits 7-4	—	—
VALTH	Bit 3	R/W	VBUS 有效阈值 0: 低于 VBUS 有效阈值 1: 高于 VBUS 有效阈值
SESVALTH	Bit 2	R/W	会话有效阈值 0: 低于会话有效阈值 1: 高于会话有效阈值
SESENDTH	Bit 1	R/W	会话结束阈值 0: 低于在会话结束阈值 1: 高于会话结束阈值
SIGCTRL	Bit 0	R/W	VBUS 信号控制位 0: 硬件控制 1: 软件控制

23.5.2.6 USB帧号寄存器 1 (USB_FRAME1)

USB 帧号寄存器 1 (USB_FRAME1)							
偏移地址: 0x000C							
复位值: 0x00							
7	6	5	4	3	2	1	0
LOWFRAME							

LOWFRAME	Bits 7-0	R	低8位的帧号
----------	----------	---	--------

23.5.2.7 USB帧号寄存器 2 (USB_FRAME2)

USB 帧号寄存器 2 (USB_FRAME2)							
偏移地址: 0x000D							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved					UPFRAME		

—	Bits 7-3	—	—
UPFRAME	Bits 2-0	R	高3位的帧号

23.5.2.8 USB 端点索引寄存器 (USB_INDEX)

USB 端点索引寄存器 (USB_INDEX)							
偏移地址: 0x000E							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved				EPTIDX			

—	Bits 7-4	—	—
EPTIDX	Bits 3-0	R/W	<p>端点号</p> <p>该位字段配置在读取或写入 USB 控制器的索引寄存器之一时，所访问的端点。0x0 的值对应于端点 0，0x4 的值对应于端点 4。</p>

23. 5. 2. 9 USB设备控制寄存器 (USB_DEVCON)

USB 设备控制寄存器 (USB_DEVCON)							
偏移地址: 0x000F							
复位值: 0x00							
7	6	5	4	3	2	1	0
CID	FSDEV	LSDEV	Reserved		HOST	HOSTREQ	SESSION

CID	Bit 7	R	CID输入信号的状态 0: A-type 1: B-type
FSDEV	Bit 6	R	全速设备检测标志位 0: 在端口上没有检测到全速设备 1: 在端口上检测到全速设备
LSDEV	Bit 5	R	低速设备检测标志位 0: 在端口上未检测到低速设备 1: 在端口上检测到低速设备
—	Bits 4-3	—	—
HOST	Bit 2	R	主机模式选择位 0: USB 控制器作为一个设备 1: USB 控制器作为主机
HOSTREQ	Bit 1	R/W	主机请求 0: 无效果 1: 在进入暂停模式时启动主机协商。
SESSION	Bit 0	R/W	会话开始或结束控制位 0: 当软件清除时, 该位结束会话。 1: 当由软件设置时, 该位启动会话。

23. 5. 2. 10 USB发送最大数据包大小寄存器 (USB_TXMAXP)

USB 发送最大数据包大小寄存器 (USB_TXMAXP)							
偏移地址: 0x0010							
复位值: 0x00							
7	6	5	4	3	2	1	0
MAXSIZE							

MAXSIZE	Bits 7-0	R/W	发送最大数据包大小 允许的最大数据包大小 每个配置的发送端点 (除了端点0) 都有一个 USB_TXMAXP 寄存器

23.5.2.11 USB发送端点中的控制状态寄存器 1 (USB_CSR0L_TXCSRL)

端点 0 中的主机模式

USB 发送端点中的控制状态寄存器 1 (USB_CSR0L_TXCSRL)							
偏移地址: 0x0011							
复位值: 0x00							
7	6	5	4	3	2	1	0
NAKTO	STATUSPKT	REQPKT	ERROR	SETUPPKT	STALLED	TXRDY	RXRDY

NAKTO	Bit 7	R/W	<p>NACK 超时标志位</p> <p>0: 没有超时</p> <p>1: 端点 0 收到 NACK 的次数大于 USB_NAKLIMIT 寄存器设置的次数</p> <p>注: 当此标志位置 1 时, 应通过软件写 0 清零此标志位, 使端点 0 继续进行事务处理。</p>
STATUSPKT	Bit 6	R/W	<p>Status stage 事务处理控制位</p> <p>0: 无交易</p> <p>1: 启动 STATUS 阶段事务注</p> <p>软件在设置 TXRDY 或 REQPKT 位的同时设置此位, 以执行状态阶段事务。设置该位可确保将数据切换设置为 1, 以便将 DATA1 数据包用于状态阶段事务。</p>
REQPKT	Bit 5	R/W	<p>IN 事务处理请求控制位</p> <p>0: 无 IN 事务处理请求</p> <p>1: 请求 IN 事务处理</p> <p>当 RXRDY 位置 1 时, 该位清零。</p>
ERROR	Bit 4	R/C_W0	<p>错误标志位</p> <p>0: 无错误</p> <p>1: 已经进行了三次尝试来执行事务, 而外围设备没有响应</p> <p>软件写 0 清零此标志位, 写 1 无效。</p>
SETUPPKT	Bit 3	R/W	<p>SETUP 令牌包发送控制位</p> <p>0: 发送 OUT 令牌</p> <p>1: 为事务发送 SETUP 令牌而不是 OUT 令牌</p> <p>注: 软件在设置 TXRDY 位的同时设置此位。</p>
STALLED	Bit 2	R/W	<p>接收 STALL 信号</p> <p>0: 尚未收到 STALL 信号</p> <p>1: 已收到 STALL 信号</p> <p>注: 软件必须清除该位。</p>
TXRDY	Bit 1	R/S_W1	<p>发送数据包 Ready 标志位</p>

			<p>0: 端点 0 数据包加载未完成</p> <p>1: 端点 0 数据包加载完成 (软件写 1 有效, 写 0 无效)</p> <p>注: 当软件完成端点 0 数据包加载时, 软件需要将此控制位置 1。当完成数据包发送时, 此控制位自动清零, 同时端点 0 产生中断 (中断使能时)。</p>
RXRDY	Bit 0	R/C_W0	<p>数据包接收 Ready 标志位</p> <p>0: 数据包接收未完成</p> <p>1: 数据包接收完成</p> <p>注: 当此控制位置 1 时, 如果中断使能, 端点 0 会产生中断。软件写 0 清零此控制位 (写 1 无效)。</p>

注: 寄存器是一个 8 位寄存器, 它为端点 0 提供控制和状态位。寄存器的解释取决于 USB 是否作为主机。

端点 0 中的设备模式

USB EP0 或 EPTX 中的控制状态寄存器 1 (USB_CSR0L_TXCSR1L)							
偏移地址: 0x0011							
复位值: 0x00							
7	6	5	4	3	2	1	0
SETENDC	RXRDYC	STALL	SETEND	DATAEND	STALLED	TXRDY	RXRDY

SETENDC	Bit 7	C_W1	<p>SETEND 清除</p> <p>向该位写入 1 将清除 SETEND 位。</p>
RXRDYC	Bit 6	C_W1	<p>RXRDY 清除</p> <p>向该位写入 1 将清除 RXRDY 位。</p>
STALL	Bit 5	W1	<p>发送 STALL 控制位</p> <p>向该位写入 1 将中止当前事务处理, 并且发送 STALL 信号。传输 STALL 信号后, 该位将自动清除。</p>
SETEND	Bit 4	R	<p>Setup 事务处理完成标志位</p> <p>在 DATAEND 控制位置 1 之前完成控制事务处理, 此控制位置 1, 同时会产生相应的中断和 FIFO Flush 操作。</p> <p>0: 控制事务处理未完成</p> <p>1: 控制事务处理完成</p> <p>注: 通过将 1 写入 SETENDC 位来清除该位。</p>
DATAEND	Bit 3	W1	<p>数据完成控制位</p> <p>当产生如下情况时, 软件需要将此位置 1:</p> <p>1. 完成最后一包数据加载置高 TXRDY;</p>

			<p>2. 完成最后一包数据读取清零 RXRDY; 3. 发送长度为 0 的数据包置高 TXRDY。 注：此控制位自动清零。</p>
STALLED	Bit 2	R/C_W0	<p>发送 STALL 信号 0: STALL 信号尚未发送 1: STALL 信号已发送 注：软件必须向该位写入 0 清除该位。</p>
TXRDY	Bit 1	R/S_W1	<p>发送数据包 Ready 标志位 0: 端点 0 数据包加载未完成 1: 端点 0 数据包加载完成（软件写 1 有效，写 0 无效） 注：当软件完成端点 0 数据包加载时，软件需要将此控制位置 1。当完成数据包发送时，此控制位自动清零，同时端点 0 产生中断（中断使能时）。</p>
RXRDY	Bit 0	R	<p>数据包接收 Ready 标志位 0: 数据包接收未完成 1: 数据包接收完成 注：当此控制位置 1 时，如果中断使能，端点 0 会产生中断。通过将 1 写入 RXRDYC 位来清除该位。</p>

注：寄存器是一个 8 位寄存器，它为端点 0 提供控制和状态位。寄存器的解释取决于 USB 是否作为一个设备。

端点 1~4 中的主机模式

USB EP0 或 EPTX 中的控制状态寄存器 1 (USB_CSR0L_TXCSR1L)							
偏移地址: 0x0011							
复位值: 0x00							
7	6	5	4	3	2	1	0
NAKTO	CLRDT	STALLED	1	FLUSH	ERROR	FIFONE	TXRDY

NAKTO	Bit 7	R/C_W0	<p>NACK 超时标志位 仅对批量端点有效。 0: 没有超时 1: 传输端点接收 NACK 的次数超过了 USB_NAKLIMIT0_TXINTERVAL 寄存器设置的次数 注：当此标志位置 1 时，应通过软件写 0 清零此标志位，使端点继续传输。</p>
CLRDT	Bit 6	S_W1	<p>清除数据转换 将 1 写入此位以重置端点 Data Toggle 重置为 0。</p>

STALLED	Bit 5	R/W	接收 STALL 信号 当该位置 1 时，FIFO 将被完全清空，并且将 TXRDY 位清除。 0: 尚未收到 STALL 信号 1: 已收到 STALL 信号 注：软件必须清除该位。
—	Bit 4	—	—
FLUSH	Bit 3	R/W	清空 FIFO 向该位写入 1，以刷新要从发送端点 FIFO 发送的下一个数据包。同时 FIFO 指针被复位，TXRDY 位被清除。 注：除非设置了 TXRDY，否则 FLUSH 无效。如果 FIFO 是双缓冲的，则可能需要将 FLUSH 设置两次以完全清除 FIFO。
ERROR	Bit 2	R/W	错误标志位 0: 无错误 1: 已经发送了 3 个数据包且没有收到握手数据包 注：软件必须清除该位。仅在端点以批量或中断模式运行时有效。
FIFONE	Bit 1	R/W	FIFO 不空标志位 0: FIFO 是空的 1: 至少一个数据包在传输 FIFO 中
TXRDY	Bit 0	R/W	发送数据包准备就绪标志位 当数据包已被发送时，该位自动清除，并产生中断（中断使能时）。 0: 没有发送数据包准备就绪 1: 软件在加载数据包到发送 FIFO 之后设置此位

注：寄存器是一个 8 位寄存器，通过当前选择的 TX 端点传输提供控制和状态位。每个配置的 TX 端点都有一个寄存器（不包括端点 0）。寄存器的解释取决于 USB 是否作为主机。

端点 1~4 中的设备模式

USB EP0 或 EPTX 中的控制状态寄存器 1 (USB_CSR0L_TXCSR1L)							
偏移地址: 0x0011							
复位值: 0x00							
7	6	5	4	3	2	1	0
—	CLRDT	STALLED	STALL	FLUSH	UNDRUN	FIFONE	TXRDY

—	Bit 7	—	—
---	-------	---	---

CLRDT	Bit 6	S_W1	清除数据转换 将 1 写入此位以重置端点 Data Toggle 重置为 0。
STALLED	Bit 5	R/W	发送 STALL 信号 当该位置 1 时, FIFO 将被完全清空, 并且将 TXRDY 位清除。 0: STALL 信号尚未发送 1: STALL 信号已发送 注: 软件必须向该位写入 0 清除该位。
STALL	Bit 4	R/W	发送 STALL 控制位 0: 没有效果 1: 向 IN 令牌发出 STALL 信号 软件清除该位以终止 STALL 条件。 注: 该位在同步传输中不起作用。
FLUSH	Bit 3	R/W	清空 FIFO 向该位写入 1, 以刷新要从发送端点 FIFO 发送的下一个数据包。同时 FIFO 指针被复位, TXRDY 位被清除。 注: 除非设置了 TXRDY, 否则 FLUSH 无效。如果 FIFO 是双缓冲的, 则可能需要将 FLUSH 设置两次以完全清除 FIFO。
UNDRUN	Bit 2	R/W	欠载标志位 0: 没有欠载 1: 未设置 TXRDY 时已接收到 IN 令牌 注: 软件必须清除该位。
FIFONE	Bit 1	R/W	FIFO 不空标志位 0: FIFO 是空的 1: 至少一个数据包在传输 FIFO 中
TXRDY	Bit 0	R/W	发送数据包准备就绪标志位 当数据包已被发送时, 该位自动清除, 并产生中断 (中断使能时)。 0: 没有发送数据包准备就绪 1: 软件在加载数据包到发送 FIFO 之后设置此位

注: 寄存器是一个 8 位寄存器, 通过当前选择的 TX 端点提供传输控制和状态位。每个配置的 TX 端点都有一个寄存器 (不包括端点 0)。寄存器的解释取决于 USB 是否作为一个设备。

23.5.2.12 USB发送端点中的控制状态寄存器 2 (USB_CSR0H_TXCSRH)

端点 0

USB 发送端点中的控制状态寄存器 2 (USB_CSR0H_TXCSRH)							
偏移地址: 0x0012							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved							FLUSH

—	Bit 7-1	—	—
FLUSH	Bit 0	R/W	FIFO Flush控制位 执行刷新后, 该位将自动清除。 0: 无效 1: 从端点0 FIFO刷新要发送与读取的下一个数据包。FIFO指针复位, TXRDY与RXRDY位清零 注: 除非设置了TXRDY或RXRDY, 否则FlushFIFO无效。

注: 寄存器包括单个自清除位, 其可用于刷新端点 0 FIFO。

端点 1~4

USB EP0 或 EPTX 中的主控制状态寄存器 2 (USB_CSR0H_TXCSRH)							
偏移地址: 0x0012							
复位值: 0x00							
7	6	5	4	3	2	1	0
AUTOSET	ISO	MODE	Reserved	FDT	Reserved		

AUTOSET	Bit 7	R/W	自动置位 0: 必须手动将 TXRDY 位置 1 1: 当最大数据包大小的数据 (USB_TXMAXP 寄存器中的值) 加载到发送 FIFO 中时, 使 TXRDY 位自动设置。如果装入的数据包小于最大数据包大小, 则必须手动设置 TXRDY 位。
ISO	Bit 6	R/W	同步传输使能位

			<p>0: 允许传输端点进行批量传输或中断传输 1: 启用传输端点进行同步传输 注: 该位仅在设备模式下有效。在主机模式下, 它始终返回零。</p>
MODE	Bit 5	R/W	<p>模式选择位 0: 设置端点方向为接收 1: 设置端点方向为发送 注: 此位只在发送和接收传输使用相同端点 FIFO 时起作用。</p>
—	Bit 4	—	—
FDT	Bit 3	R/W	<p>强制切换 Data Toggle 0: 无效果 1: 无论是否接收到 ACK, 都强制切换端点的 Data Toggle 位并从 FIFO 中清除数据包。 中断传输端点可使用此位, 该端点用于传达同步端点的速率反馈。</p>
—	Bits 2-0	—	—

注: 寄存器是一个 8 位寄存器, 通过当前选择的发送端点提供用于传输的进一步控制位。每个配置的发送端点都有一个寄存器 (不包括端点 0)。

23.5.2.13 USB接收最大数据包大小寄存器 (USB_RXMAXP)

USB 接收最大数据包大小寄存器 (USB_RXMAXP)							
偏移地址: 0x0013							
复位值: 0x00							
7	6	5	4	3	2	1	0
MAXSIZE							

MAXSIZE	Bits 7-0	R/W	接收端点单次可传输的最大数据的长度 允许的最大数据包大小 每个配置接收端点 (除了端点0) 都有一个 USB_RXMAXP寄存器

注 1: 寄存器是一个 8 位寄存器, 用于保存通过当前选择的 Rx 端点进行的事务的最大数据包大小-以 8 字节为单位, 但值 128 会将最大数据包大小设置为 1023 (在全速事务中传输同步数据包的最大大小) 而不是 1024。在设置此值时, 应注意 USB 规范对全速操作中的批量, 中断和同步事务的数据包大小设置的约束。

注 2: 每个配置接收端点都有一个寄存器 (除了端点 0)。

注 3: 写入到该寄存器的值所表示的数据总量不能超过 RX FIFO 大小, 或者使用动态 FIFO 大小, 在 USB_RXFIFO2 寄存器中指定的最大分组大小。如果写入此寄存器的值小于或等于 RX FIFO 大小的一半, 则可以缓冲两个数据包 (除非选择了动态 FIFO 大小)。

23.5.2.14 USB接收控制状态寄存器 1 (USB_RXCSRL)

主机模式

USB 接收控制状态寄存器 1 (USB_RXCSRL)							
偏移地址: 0x0014							
复位值: 0x00							
7	6	5	4	3	2	1	0
CLRDT	STALLED	REQPKT	FLUSH	DATAERR/ NAKTO	ERROR	FULL	RXRDY

CLRDT	Bit 7	R/W	清除数据转换 将 1 写入此位以重置端点 Data Toggle 重置为 0。
STALLED	Bit 6	R/W	接收 STALL 信号 0: 尚未收到 STALL 信号 1: 已收到 STALL 信号, 此时会产生中断 注: 软件必须清除该位。
REQPKT	Bit 5	R/W	IN 事务处理请求控制位 0: 无 IN 事务处理请求 1: 请求 IN 事务处理 当 RXRDY 位置 1 时, 该位清零。
FLUSH	Bit 4	R/W	清空 FIFO 向该位写入 1, 以刷新要从端点 Rx FIFO 读取的下一个数据包。同时 FIFO 指针被复位, RXRDY 位被清除。 注: 除非设置了 RXRDY, 否则 FLUSH 无效。另请注意, 如果 FIFO 是双缓冲的, 则可能需要将 FLUSH 设置两次以完全清除 FIFO。
DATAERR/NAKTO	Bit 3	R/W	数据错误/ NAK 超时 0: 正常运行 1: 在同步模式下操作时, 如果数据包具有 CRC 或位填充错误, 则当 RXRDY 设置为 1 时, 同时此位置 1; 当 RXRDY 清除时, 同时清除该位。在批量模式下, 当接收到 NAK 响应后, 如果 Rx 端点暂停的时间超过 USB_RXINTERVAL 寄存器设置为 NAK 限制的时间, 则该位置 1。 注: 软件必须清除该位以允许端点继续。
ERROR	Bit 2	R/W	错误标志位 0: 没有错误 1: 已经进行了三次尝试来接收数据包, 并且没有接收到数据包 软件必须清除该位。

			注：仅当接收端点在批量或中断模式下运行时，此位才有效。在同步模式下，它始终返回零。
FULL	Bit 1	R	FIFO 满标志位 0: 接收 FIFO 未 1: 满，无法再将更多数据包加载到接收 FIFO 中
RXRDY	Bit 0	R/W	数据包接收 Ready 标志位 0: 数据包接收未完成 1: 数据包接收完成 注：从 Rx FIFO 卸载数据包后，软件应清除该位。当该位置 1 时，产生一个中断。

注：寄存器是一个 8 位寄存器，通过当前选择的接收端点提供控制和状态位进行传输。每个配置的接收端点都有一个寄存器（不包括端点 0）。寄存器的解释取决于 USB 是否作为主机。

设备模式

USB 接收控制状态寄存器 1 (USB_RXCSRL)							
偏移地址: 0x0014							
复位值: 0x00							
7	6	5	4	3	2	1	0
CLRDT	STALLED	STALL	FLUSH	DATAERR	OVERRUN	FULL	RXRDY

CLRDT	Bit 7	R/W	清除数据转换 将 1 写入此位以重置端点 Data Toggle 重置为 0。
STALLED	Bit 6	R/W	发送 STALL 信号 0: STALL 信号尚未发送 1: STALL 信号已发送 注：软件必须清除该位。
STALL	Bit 5	C_W1	发送 STALL 控制位 向该位写入 1 以发出 STALL 信号。软件清除该位以终止停止条件。 注：当端点用于同步传输时，此位无效。
FLUSH	Bit 4	R/W	清空 FIFO 向该位写入 1，以刷新要从端点 Rx FIFO 读取的下一个数据包。同时 FIFO 指针被复位，RXRDY 位被清除。
DATAERR	Bit 3	R/W	数据错误 0: 正常运行 1: 在同步模式下操作时，如果数据包具有 CRC 或

			<p>位填充错误, 则当 RXRDY 设置为 1 时, 同时此位置 1; 当 RXRDY 清除时, 同时清除该位</p> <p>注: 仅当端点在同步模式下运行时, 此位才有效。在批量模式下, 它始终返回零。</p>
OVERRUN	Bit 2	R/W	<p>溢出标志位</p> <p>0: 无溢出错误</p> <p>1: 发生溢出, 表示无法将 OUT 数据包加载到接收 FIFO 中</p> <p>注: 仅当端点在同步模式下运行时, 此位才有效。在批量模式下, 它始终返回零。</p>
FULL	Bit 1	R	<p>FIFO 满标志位</p> <p>0: 接收 FIFO 未满</p> <p>1: 满, 无法再将更多数据包加载到接收 FIFO 中</p>
RXRDY	Bit 0	R/W	<p>数据包接收 Ready 标志位</p> <p>0: 数据包接收未完成</p> <p>1: 数据包接收完成</p> <p>注: 从 Rx FIFO 卸载数据包后, 软件应清除该位。当该位置 1 时, 产生一个中断。</p>

注: 寄存器是一个 8 位寄存器, 通过当前选择的接收端点提供控制和状态位进行传输。每个配置接收端点都有一个寄存器 (不包括端点 0)。寄存器的解释取决于 USB 是否作为一个设备。

23.5.2.15 USB接收控制状态寄存器 2 (USB_RXCSRH)

主机模式

USB 接收控制状态寄存器 2 (USB_RXCSRH)							
偏移地址: 0x0015							
复位值: 0x00							
7	6	5	4	3	2	1	0
AUTOCLR	AUTOREQ	Reserved					

AUTOCLR	Bit 7	R/W	自动清除 0: 无效果 1: 当从接收 FIFO 接收了一个 UBS_RXMAXP 寄存器字节包时, 可以自动清除 RXRDY 位。当接收小于最大分组大小的数据包时, 必须手动清除 RXRDY。
AUTOREQ	Bit 6	R/W	自动请求 0: 没有效果 1: 当 RXRDY 位被清除时, 可以自动设置 REQPKT 位
—	Bits 5-0	—	—

注: 寄存器是一个 8 位寄存器, 通过当前选择的接收端点提供用于传输的其他控制位。每个配置的接收端点都有一个寄存器 (不包括端点 0)。寄存器的解释取决于 USB 是否作为主机。

设备模式

USB 接收控制状态寄存器 2 (USB_RXCSRH)							
偏移地址: 0x0015							
复位值: 0x00							
7	6	5	4	3	2	1	0
AUTOCLR	ISO	Reserved					

AUTOCLR	Bit 7	R/W	自动清除 0: 无效果 1: 当从接收 FIFO 接收了一个 UBS_RXMAXP 寄存器字节包时, 可以自动清除 RXRDY 位。当接收小于最大分组大小的数据包时, 必须手动清除 RXRDY。
ISO	Bit 6	R/W	同步传输 0: 允许传输端点进行批量传输或中断传输 1: 启用传输端点进行同步传输
—	Bits 5-0	—	—

注: 寄存器是一个 8 位寄存器, 通过当前选择的接收端点提供用于传输的其他控制位。每个配置接收端点都有一个寄存器 (不包括端点 0)。寄存器的解释取决于 USB 是否作为设备。

23.5.2.16 USB接收端点中的接收字节数寄存器 (USB_COUNT0_RX1)

端点 0

USB 接收端点中接收字节数寄存器 (USB_COUNT0_RX1)							
偏移地址: 0x0016							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	COUNT						

—	Bit 7	—	—
COUNT	Bits 6-0	R	FIFO 数据字节数 该字段是一个只读值，指示端点 0 FIFO 中接收到的数据字节数

注：寄存器是一个 7 位只读寄存器，它指示端点 0 FIFO 中接收到的数据字节数。当设置 USB_CSR0L_TXCSRL.RXRDY 时，返回的值有效。

端点 1~4

USB EP0 或 EPRX 中接收字节数的寄存器 (USB_COUNT0_RX1)							
偏移地址: 0x0016							
复位值: 0x00							
7	6	5	4	3	2	1	0
PKTLOW							

PKTLOW	Bits 7-0	R	接收数据字节数的低 8 位 指示接收数据字节数的低 8 位。
--------	----------	---	--

注：寄存器是 8 位只读寄存器，它保存与当前选择的接收端点相关联的 FIFO 中的分组中接收数据字节数的较低 8 位。当设置 USB_RXCSRL.RXRDY 时，返回的值有效。

23.5.2.17 USB接收字节数寄存器 2 (USB_RXCOUNT2)

USB 接收字节数寄存器 2 (USB_RXCOUNT2)							
偏移地址: 0x0017							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved				PKTHIGH			

—	Bits 7-3	—	—
PKTHIGH	Bits 2-0	R	接收数据字节数的高 3 位 指示接收数据字节数的高 3 位。

注：寄存器是 3 位只读寄存器，它保存与当前选择的接收端点相关联的 FIFO 中的分组中接收数据字节数的高 3 位。当设置 USB_RXCSRL.RXRDY 时，返回的值有效。

23.5.2.18 USB发送协议类型寄存器 (USB_TXTYPE)

USB 发送协议类型寄存器 (USB_TXTYPE)							
偏移地址: 0x0018							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved		PROTOCOL		TEPN			

—	Bits 7-6	—	—
PROTOCOL	Bit 5-4	R/W	协议 软件必须设置选择发送端点所需的协议: 00: 控制 01: 同步 10: 批量 11: 中断
TEPN	Bit 3-0	R/W	目标端点号 软件必须将此值配置为在设备枚举期间返回到 USB 控制器的发送端点描述符中包含的端点号。

注: 寄存器是一个 6 位的寄存器, 应该用端点编号来写, 该端点编号要由位于较低 4 位的端点作为目标, 事务协议用于位于较高 2 位的当前选择的发送端点。每个配置的发送端点都有一个寄存器 (除了端点 0)。

23.5.2.19 USB 端点NAK响应超时或发送轮询间隔寄存器 (USB_NAKLIMIT0_TXINTERVAL)

端点 0

USB 端点 NAK 响应超时或发送轮询间隔寄存器 (USB_NAKLIMIT0_TXINTERVAL)							
偏移地址: 0x0019							
复位值: 0x00							
7	6	5	4	3	2	1	0
NAKLMT							

NAKLMT	Bits 7-0	R/W	NAK 极限 此字段指定接收 NAK 响应后的帧的数目。 注: 0 或 1 的值禁用 NAK 超时功能。
--------	----------	-----	---

注 1: 寄存器是一个 8 位寄存器, 用于设置接收 NAK 响应后端点 0 应该超时的帧数。(其他端点的等效设置可以通过 USB_TXINTERVAL 和 USB_RXINTERVAL 寄存器进行)。

注 2: 所选择的帧的数目可以在 2 和 255 之间。如果主机从目标接收到 NAK 响应, 其帧数超过该寄存器中的极限所表示的数目, 则端点将停止。

注 3: 寄存器的解释取决于 USB 是否作为主机。

端点 1~4

USB 端点 NAK 响应超时或发送轮询间隔寄存器 (USB_NAKLIMIT0_TXINTERVAL)							
偏移地址: 19 _H							
复位值: 00000000 _B							
7	6	5	4	3	2	1	0
TXPOLL/ NAKLMT <7:0>							

TXPOLL/NAKLMT	Bits 7-0	R/W	发送轮询/NAK 限制 中断/同步传输的轮询间隔; 批量传输的 NAK 限制。 注: USB 规范允许同步端点的轮询间隔大于 255ms。如果需要, 这些必须以软件实现。
---------------	----------	-----	--

注 1: 寄存器是一个 8 位寄存器, 对于中断和同时传输, 它定义当前选择的发送端点的轮询间隔。对于批量端点, 此寄存器设置接收 NAK 响应时端点应该超时的帧数。每个配置的发送端点都有一个寄存器 (除了端点 0)。

注 2: 寄存器的解释取决于 USB 是否作为主机。

23.5.2.20 USB接收协议类型寄存器 (USB_RXTYPE)

USB 接收协议类型寄存器 (USB_RXTYPE)							
偏移地址: 0x001A							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved		PROTOCOL		TEPN			

—	Bits 7-6	—	—
PROTOCOL	Bits 5-4	R/W	协议 软件必须设置选择接收端点所需的协议： 00: 控制 01: 同步 10: 批量 11: 中断
TEPN	Bits 3-0	R/W	目标端点号 软件必须将此值配置为在设备枚举期间返回到 USB 控制器的接收端点描述符中包含的端点号。

注 1: 寄存器是一个 6 位的寄存器, 它应该用端点编号来写, 该端点编号要由低 4 位的端点作为目标, 事务协议用于高 2 位的当前选择的接收端点。每个配置的接收端点都有一个寄存器 (除了端点 0)。

注 2: 寄存器的解释取决于 USB 是否作为主机。

23.5.2.21 USB接收轮询间隔寄存器 (USB_RXINTERVAL)

USB 接收轮询间隔寄存器 (USB_RXINTERVAL)							
偏移地址: 0x001B							
复位值: 0x00							
7	6	5	4	3	2	1	0
RX POLL / NAK LMT							

RXPOLL/NAKLMT	Bits 7-0	R/W	接收轮询/NAK极限
			中断/同步传输的轮询间隔; 批量传输的NAK限制。 注: USB规范允许同步端点的轮询间隔大于255ms。如果需要, 这些必须以软件实现。

注 1: 寄存器是一个 8 位寄存器, 对于中断和同时传输, 它为当前选择的接收端点定义轮询间隔。对于批量端点, 此寄存器设置接收 NAK 响应时端点应该超时的帧数。每个配置的接收端点都有一个寄存器 (除了端点 0)。

注 2: 寄存器的解释取决于 USB 是否作为主机。

23.5.2.22 USB发送FIFO配置寄存器 1 (USB_TXFIFO1)

USB 发送 FIFO 配置寄存器 1 (USB_TXFIFO1)							
偏移地址: 0x001C							
复位值: 0x00							
7	6	5	4	3	2	1	0
ADDRL							

ADDRL	Bits 7-0	R/W	发送起始地址 (低8位) 端点FIFO的起始地址 (低8位) ADDRH+ADDL起始地址 0x000 0x0000 0x001 0x0008 0x002 0x0010 0xFFF 0x7FF8

注: 寄存器是一个 8 位寄存器, 共同控制起始地址和选定的发送端点 FIFO 的大小。

23.5.2.23 USB发送FIFO配置寄存器 2 (USB_TXFIFO2)

USB 发送 FIFO 配置寄存器 2 (USB_TXFIFO2)							
偏移地址: 0x001D							
复位值: 0x00							
7	6	5	4	3	2	1	0
MAXPKTSIZE			DPB	ADDRH			

MAXPKTSIZE	Bits 7-5	R/W	最大数据包大小 (字节) 0x0: 8 0x1: 16 0x2: 32 0x3: 64 0x4: 128 0x5: 256 0x6: 512 0x7: 1024 如果 DPB=0, FIFO 也将是这个大小; 如果 DPB=1, FIFO 将是这个大小的两倍。
DPB	Bit 4	R/W	双数据包缓冲支持使能位 0: 仅支持单数据包缓冲 1: 支持双数据包缓冲
ADDRH	Bits 3-0	R/W	发送起始地址 (高 3 位) 端点 FIFO 的起始地址 (高 3 位)

注: 寄存器是一个 8 位寄存器, 共同控制起始地址和选定的发送端点 FIFO 的大小。

23.5.2.24 USB接收FIFO配置寄存器 1 (USB_RXFIFO1)

USB 接收 FIFO 配置寄存器 1 (USB_RXFIFO1)							
偏移地址: 0x001E							
复位值: 0x00							
7	6	5	4	3	2	1	0
ADDRL							

ADDRL	Bits 7-0	R/W	接收起始地址 (低8位) 端点FIFO的起始地址 (低8位) ADDRH+ADDL起始地址 0x000 0x0000 0x001 0x0008 0x002 0x0010 0xFFFF 0x7FF8

注: 寄存器是一个 8 位寄存器, 共同控制起始地址和选定的接收端点 FIFO 的大小。

23.5.2.25 USB接收FIFO配置寄存器 2 (USB_RXFIFO2)

USB 接收 FIFO 配置寄存器 2 (USB_RXFIFO2)							
偏移地址: 0x001F							
复位值: 0x00							
7	6	5	4	3	2	1	0
MAXPKTSIZE			DPB	ADDRH			

MAXPKTSIZE	Bits 7-5	R/W	最大数据包大小 (字节) 0x0: 8 0x1: 16 0x2: 32 0x3: 64 0x4: 128 0x5: 256 0x6: 512 0x7: 1024 如果 DPB=0, FIFO 也将是这个大小; 如果 DPB=1, FIFO 将是这个大小的两倍。
DPB	Bit 4	R/W	双数据包缓冲支持使能位 0: 仅支持单数据包缓冲 1: 支持双数据包缓冲
ADDRH	Bits 3-0	R/W	接收起始地址 (高 3 位) 端点 FIFO 的起始地址 (高 3 位)

注: 寄存器是一个 8 位寄存器, 共同控制起始地址和选定的接收端点 FIFO 的大小。

23.5.2.26 USB端点 0 FIFO (USB_EP0FIFO)

USB 端点 0 FIFO (USB_EP0FIFO)							
偏移地址: 0x0020							
复位值: 0x00							
7	6	5	4	3	2	1	0
FIFO0							

FIFO0	Bits 7-0	R/W	<p>端点0 FIFO</p> <p>对此寄存器进行写操作，将向发送FIFO中写入数据，对此寄存器进行读操作，将从接收FIFO中读出数据。</p>
-------	----------	-----	--

23.5.2.27 USB端点 1 FIFO (USB_EP1FIFO)

USB 端点 1 FIFO (USB_EP1FIFO)							
偏移地址: 0x0024							
复位值: 0x00							
7	6	5	4	3	2	1	0
FIFO1							

FIFO1	Bits 7-0	R/W	<p>端点 1 FIFO</p> <p>对此寄存器进行写操作，将向发送 FIFO 中写入数据，对此寄存器进行读操作，将从接收 FIFO 中读出数据。</p>
-------	----------	-----	---

23.5.2.28 USB端点 2 FIFO (USB_EP2FIFO)

USB 端点 2 FIFO (USB_EP1FIFO)							
偏移地址: 0x0028							
复位值: 0x00							
7	6	5	4	3	2	1	0
FIFO2							

FIFO2	Bits 7-0	R/W	端点 2 FIFO 对此寄存器进行写操作，将向发送 FIFO 中写入数据，对此寄存器进行读操作，将从接收 FIFO 中读出数据。
-------	----------	-----	--

23.5.2.29 USB端点 3 FIFO (USB_EP3FIFO)

USB 端点 3 FIFO (USB_EP3FIFO)							
偏移地址: 0x002C							
复位值: 0x00							
7	6	5	4	3	2	1	0
FIFO3							

FIFO3	Bits 7-0	R/W	端点3 FIFO 对此寄存器进行写操作，将向发送FIFO中写入数据，对此寄存器进行读操作，将从接收FIFO中读出数据。
-------	----------	-----	---

23. 5. 2. 30 USB端点 4 FIFO (USB_EP4FIFO)

USB 端点 4 FIFO (USB_EP4FIFO)							
偏移地址: 0x0030							
复位值: 0x00							
7	6	5	4	3	2	1	0
FIFO4							

FIFO4	Bits 7-0	R/W	<p>端点 4 FIFO</p> <p>对此寄存器进行写操作，将向发送 FIFO 中写入数据，对此寄存器进行读操作，将从接收 FIFO 中读出数据。</p>
-------	----------	-----	---

23.5.2.31 USB发送中断使能寄存器 (USB_TXIER)

USB 发送中断使能寄存器 (USB_TXIER)							
偏移地址: 0x0080							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	W1	发送端点 4 中断使能 0: 写入 0 无效 1: 使能发送端点 4 中断
EP3	Bit 3	W1	发送端点 3 中断使能 0: 写入 0 无效 1: 使能发送端点 3 中断
EP2	Bit 2	W1	发送端点 2 中断使能 0: 写入 0 无效 1: 使能发送端点 2 中断
EP1	Bit 1	W1	发送端点 1 中断使能 0: 写入 0 无效 1: 使能发送端点 1 中断
EP0	Bit 0	W1	发送端点 0 中断使能 0: 写入 0 无效 1: 使能发送端点 0 中断

23. 5. 2. 32 USB接收中断使能寄存器 (USB_RXIER)

USB 接收中断使能寄存器 (USB_RXIER)							
偏移地址: 0x0082							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	W1	接收端点 4 中断使能 0: 写入 0 无效 1: 使能接收端点 4 中断
EP3	Bit 3	W1	接收端点 3 中断使能 0: 写入 0 无效 1: 使能接收端点 3 中断
EP2	Bit 2	W1	接收端点 2 中断使能 0: 写入 0 无效 1: 使能接收端点 2 中断
EP1	Bit 1	W1	接收端点 1 中断使能 0: 写入 0 无效 1: 使能接收端点 1 中断
—	Bit 0	—	—

23. 5. 2. 33 USB发送中断禁用寄存器 (USB_TXIDR)

USB 发送中断禁用寄存器 (USB_TXIDR)							
偏移地址: 0x0084							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	W1	发送端点 4 中断禁止 0: 写入 0 无效 1: 禁止发送端点 4 中断
EP3	Bit 3	W1	发送端点 3 中断禁止 0: 写入 0 无效 1: 禁止发送端点 3 中断
EP2	Bit 2	W1	发送端点 2 中断禁止 0: 写入 0 无效 1: 禁止发送端点 2 中断
EP1	Bit 1	W1	发送端点 1 中断禁止 0: 写入 0 无效 1: 禁止发送端点 1 中断
EP0	Bit 0	W1	发送端点 0 中断禁止 0: 写入 0 无效 1: 禁止发送端点 0 中断

23.5.2.34 USB接收中断禁用寄存器 (USB_RXIDR)

USB 接收中断禁用寄存器 (USB_RXIDR)							
偏移地址: 0x0086							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	W1	接收端点 4 中断禁止 0: 写入 0 无效 1: 禁止接收端点 4 中断
EP3	Bit 3	W1	接收端点 3 中断禁止 0: 写入 0 无效 1: 禁止接收端点 3 中断
EP2	Bit 2	W1	接收端点 2 中断禁止 0: 写入 0 无效 1: 禁止接收端点 2 中断
EP1	Bit 1	W1	接收端点 1 中断禁止 0: 写入 0 无效 1: 禁止接收端点 1 中断
—	Bit 0	—	—

23. 5. 2. 35 USB发送中断使能状态寄存器 (USB_TXIVS)

USB 发送中断使能状态寄存器 (USB_TXIVS)							
偏移地址: 0x0088							
复位值: 0x7F							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	R	发送端点 4 中断使能状态 0: 禁止发送端点 4 中断 1: 使能发送端点 4 中断
EP3	Bit 3	R	发送端点 3 中断使能状态 0: 禁止发送端点 3 中断 1: 使能发送端点 3 中断
EP2	Bit 2	R	发送端点 2 中断使能状态 0: 禁止发送端点 2 中断 1: 使能发送端点 2 中断
EP1	Bit 1	R	发送端点 1 中断使能状态 0: 禁止发送端点 1 中断 1: 使能发送端点 1 中断
EP0	Bit 0	R	发送端点 0 中断使能状态 0: 禁止发送端点 0 中断 1: 使能发送端点 0 中断

23. 5. 2. 36 USB接收中断使能状态寄存器 (USB_RXIVS)

USB 接收中断使能状态寄存器 (USB_RXIVS)							
偏移地址: 0x008A							
复位值: 0x7E							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	R	接收端点 4 中断使能状态 0: 禁止接收端点 4 中断 1: 使能接收端点 4 中断
EP3	Bit 3	R	接收端点 3 中断使能状态 0: 禁止接收端点 3 中断 1: 使能接收端点 3 中断
EP2	Bit 2	R	接收端点 2 中断使能状态 0: 禁止接收端点 2 中断 1: 使能接收端点 2 中断
EP1	Bit 1	R	接收端点 1 中断使能状态 0: 禁止接收端点 1 中断 1: 使能接收端点 1 中断
—	Bit 0	—	—

23.5.2.37 USB发送原始中断事件标志寄存器 (USB_TXRIF)

USB 发送原始中断事件标志寄存器 (USB_TXRIF)							
偏移地址: 0x008C							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	R	发送端点 4 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP3	Bit 3	R	发送端点 3 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP2	Bit 2	R	发送端点 2 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP1	Bit 1	R	发送端点 1 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP0	Bit 0	R	发送端点 0 中断事件标志 0: 未发生中断事件 1: 发生中断事件

23. 5. 2. 38 USB接收原始中断事件标志寄存器 (USB_RXRIF)

USB 接收原始中断事件标志寄存器 (USB_RXRIF)							
偏移地址: 0x008E							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	R	接收端点 4 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP3	Bit 3	R	接收端点 3 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP2	Bit 2	R	接收端点 2 中断事件标志 0: 未发生中断事件 1: 发生中断事件
EP1	Bit 1	R	接收端点 1 中断事件标志 0: 未发生中断事件 1: 发生中断事件
—	Bit 0	—	—

23. 5. 2. 39 USB发送中断屏蔽标志寄存器 (USB_TXIFM)

USB 发送中断屏蔽标志寄存器 (USB_TXIFM)							
偏移地址: 0x0090							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	R	发送端点 4 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP3	Bit 3	R	发送端点 3 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP2	Bit 2	R	发送端点 2 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP1	Bit 1	R	发送端点 1 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP0	Bit 0	R	发送端点 0 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断

23.5.2.40 USB接收中断屏蔽标志寄存器 (USB_RXIFM)

USB 接收中断屏蔽标志寄存器 (USB_RXIFM)							
偏移地址: 0x0092							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	R	接收端点 4 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP3	Bit 3	R	接收端点 3 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP2	Bit 2	R	接收端点 2 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
EP1	Bit 1	R	接收端点 1 中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 0	—	—

23.5.2.41 USB发送中断清除寄存器 (USB_TXICR)

USB 发送中断清除寄存器 (USB_TXICR)							
偏移地址: 0x0094							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	EP0

—	Bit 7-5	—	—
EP4	Bit 4	C_W1	发送端点 4 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP3	Bit 3	C_W1	发送端点 3 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP2	Bit 2	C_W1	发送端点 2 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP1	Bit 1	C_W1	发送端点 1 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP0	Bit 0	C_W1	发送端点 0 中断清除 0: 写入 0 无效 1: 清除中断事件与中断

23. 5. 2. 42 USB接收中断清除寄存器 (USB_RXICR)

USB 接收中断清除寄存器 (USB_RXICR)							
偏移地址: 0x0096							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved			EP4	EP3	EP2	EP1	Reserved

—	Bit 7-5	—	—
EP4	Bit 4	C_W1	接收端点 4 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP3	Bit 3	C_W1	接收端点 3 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP2	Bit 2	C_W1	接收端点 2 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
EP1	Bit 1	C_W1	接收端点 1 中断清除 0: 写入 0 无效 1: 清除中断事件与中断
—	Bit 0	—	—

23. 5. 2. 43 USB中断使能寄存器 (USB_IER)

主机模式

USB 中断使能寄存器 (USB_IER)							
偏移地址: 0x00A0							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	W1	会话请求中断使能 0: 写入 0 无效 1: 使能会话请求中断
DISCON	Bit 5	W1	断开中断使能 0: 写入 0 无效 1: 使能断开中断
CON	Bit 4	W1	连接中断使能 0: 写入 0 无效 1: 使能连接中断
SOF	Bit 3	W1	帧中断使能 0: 写入 0 无效 1: 使能帧中断
BAB	Bit 2	W1	干扰中断使能 0: 写入 0 无效 1: 使能干扰中断
RES	Bit 1	W1	恢复中断使能 0: 写入 0 无效 1: 使能恢复中断
—	Bit 0	—	—

设备模式

USB 中断使能寄存器 (USB_IER)							
偏移地址: 0x00A0							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	W1	会话请求中断使能 0: 写入 0 无效 1: 使能会话请求中断
DISCON	Bit 5	W1	断开中断使能 0: 写入 0 无效 1: 使能断开中断
—	Bit 4	—	—
SOF	Bit 3	W1	帧中断使能 0: 写入 0 无效 1: 使能帧中断
REST	Bit 2	W1	复位中断使能 0: 写入 0 无效 1: 使能复位中断
RES	Bit 1	W1	恢复中断使能 0: 写入 0 无效 1: 使能恢复中断
SUSPD	Bit 0	W1	挂起中断使能 0: 写入 0 无效 1: 使能挂起中断

23.5.2.44 USB中断禁用寄存器 (USB_IDR)

主机模式

USB 中断禁用寄存器 (USB_IDR)							
偏移地址: 0x00A4							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	W1	会话请求中断禁止 0: 写入 0 无效 1: 禁止会话请求中断
DISCON	Bit 5	W1	断开中断禁止 0: 写入 0 无效 1: 禁止断开中断
CON	Bit 4	W1	连接中断禁止 0: 写入 0 无效 1: 禁止连接中断
SOF	Bit 3	W1	帧中断禁止 0: 写入 0 无效 1: 禁止帧中断
BAB	Bit 2	W1	干扰中断禁止 0: 写入 0 无效 1: 禁止干扰中断
RES	Bit 1	W1	恢复中断禁止 0: 写入 0 无效 1: 禁止恢复中断
—	Bit 0	—	—

设备模式

USB 中断禁用寄存器 (USB_IDR)							
偏移地址: 0x00A4							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	W1	会话请求中断禁止 0: 写入 0 无效 1: 禁止会话请求中断
DISCON	Bit 5	W1	断开中断禁止 0: 写入 0 无效 1: 禁止断开中断
—	Bit 4	—	—
SOF	Bit 3	W1	帧中断禁止 0: 写入 0 无效 1: 禁止帧中断
REST	Bit 2	W1	复位中断禁止 0: 写入 0 无效 1: 禁止复位中断
RES	Bit 1	W1	恢复中断禁止 0: 写入 0 无效 1: 禁止恢复中断
SUSPD	Bit 0	W1	挂起中断禁止 0: 写入 0 无效 1: 禁止挂起中断

23. 5. 2. 45 USB中断使能状态寄存器 (USB_IVS)

主机模式

USB 中断使能状态寄存器 (USB_IVS)							
偏移地址: 0x00A8							
复位值: 0x06							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断使能状态 0: 禁止会话请求中断 1: 使能会话请求中断
DISCON	Bit 5	R	断开中断使能状态 0: 禁止断开中断 1: 使能断开中断
CON	Bit 4	R	连接中断使能状态 0: 禁止连接中断 1: 使能连接中断
SOF	Bit 3	R	帧中断使能状态 0: 禁止帧中断 1: 使能帧中断
BAB	Bit 2	R	干扰中断使能状态 0: 禁止干扰中断 1: 使能干扰中断
RES	Bit 1	R	恢复中断使能状态 0: 禁止恢复中断 1: 使能恢复中断
—	Bit 0	—	—

设备模式

USB 中断使能状态寄存器 (USB_IVS)							
偏移地址: 0x00A8							
复位值: 0x06							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断使能状态 0: 禁止会话请求中断 1: 使能会话请求中断
DISCON	Bit 5	R	断开中断使能状态 0: 禁止断开中断 1: 使能断开中断
—	Bit 4	—	—
SOF	Bit 3	R	帧中断使能状态 0: 禁止帧中断 1: 使能帧中断
REST	Bit 2	R	复位中断使能状态 0: 禁止复位中断 1: 使能复位中断
RES	Bit 1	R	恢复中断使能状态 0: 禁止恢复中断 1: 使能恢复中断
SUSPD	Bit 0	R	挂起中断使能状态 0: 禁止挂起中断 1: 使能挂起中断

23. 5. 2. 46 USB原始中断事件标志寄存器 (USB_RIF)

主机模式

USB 原始中断事件标志寄存器 (USB_RIF)							
偏移地址: 0x00AC							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断事件标志 0: 未发生中断事件 1: 发生中断事件
DISCON	Bit 5	R	断开中断事件标志 0: 未发生中断事件 1: 发生中断事件
CON	Bit 4	R	连接中断事件标志 0: 未发生中断事件 1: 发生中断事件
SOF	Bit 3	R	帧中断事件标志 0: 未发生中断事件 1: 发生中断事件
BAB	Bit 2	R	干扰中断事件标志 0: 未发生中断事件 1: 发生中断事件
RES	Bit 1	R	恢复中断事件标志 0: 未发生中断事件 1: 发生中断事件
—	Bit 0	—	—

设备模式

USB 原始中断事件标志寄存器 (USB_RIF)							
偏移地址: 0x00AC							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断事件标志 0: 未发生中断事件 1: 发生中断事件
DISCON	Bit 5	R	断开中断事件标志 0: 未发生中断事件 1: 发生中断事件
—	Bit 4	—	—
SOF	Bit 3	R	帧中断事件标志 0: 未发生中断事件 1: 发生中断事件
REST	Bit 2	R	复位中断事件标志 0: 未发生中断事件 1: 发生中断事件
RES	Bit 1	R	恢复中断事件标志 0: 未发生中断事件 1: 发生中断事件
SUSPD	Bit 0	R	挂起中断事件标志 0: 未发生中断事件 1: 发生中断事件

23. 5. 2. 47 USB中断屏蔽标志寄存器 (USB_IFM)

主机模式

USB 中断屏蔽标志寄存器 (USB_IFM)							
偏移地址: 0x00B0							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
DISCON	Bit 5	R	断开中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
CON	Bit 4	R	连接中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
SOF	Bit 3	R	帧中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
BAB	Bit 2	R	干扰中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
RES	Bit 1	R	恢复中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 0	—	—

设备模式

USB 中断屏蔽标志寄存器 (USB_IFM)							
偏移地址: 0x00B0							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	R	会话请求中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
DISCON	Bit 5	R	断开中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
—	Bit 4	—	—
SOF	Bit 3	R	帧中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
REST	Bit 2	R	复位中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
RES	Bit 1	R	恢复中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断
SUSPD	Bit 0	R	挂起中断屏蔽标志 0: 未发生中断事件或中断未使能 1: 产生中断

23. 5. 2. 48 USB中断清除寄存器 (USB_ICR)

主机模式

USB 中断清除寄存器 (USB_ICR)							
偏移地址: 0x00B4							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	CON	SOF	BAB	RES	Reserved

—	Bit 7	—	—
SESREQ	Bit 6	C_W1	会话请求中断清除 0: 写入 0 无效 1: 清除中断事件与中断
DISCON	Bit 5	C_W1	断开中断清除 0: 写入 0 无效 1: 清除中断事件与中断
CON	Bit 4	C_W1	连接中断清除 0: 写入 0 无效 1: 清除中断事件与中断
SOF	Bit 3	C_W1	帧中断清除 0: 写入 0 无效 1: 清除中断事件与中断
BAB	Bit 2	C_W1	干扰中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RES	Bit 1	C_W1	恢复中断清除 0: 写入 0 无效 1: 清除中断事件与中断
—	Bit 0	—	—

设备模式

USB 中断清除寄存器 (USB_ICR)							
偏移地址: 0x00B4							
复位值: 0x00							
7	6	5	4	3	2	1	0
Reserved	SESREQ	DISCON	Reserved	SOF	REST	RES	SUSPD

—	Bit 7	—	—
SESREQ	Bit 6	C_W1	会话请求中断清除 0: 写入 0 无效 1: 清除中断事件与中断
DISCON	Bit 5	C_W1	断开中断清除 0: 写入 0 无效 1: 清除中断事件与中断
—	Bit 4	—	—
SOF	Bit 3	C_W1	帧中断清除 0: 写入 0 无效 1: 清除中断事件与中断
REST	Bit 2	C_W1	复位中断清除 0: 写入 0 无效 1: 清除中断事件与中断
RES	Bit 1	C_W1	恢复中断清除 0: 写入 0 无效 1: 清除中断事件与中断
SUSPD	Bit 0	C_W1	挂起中断清除 0: 写入 0 无效 1: 清除中断事件与中断

第24章 模数转换器（ADC）

24.1 概述

该 ADC 为逐次逼近型模数转换器，采样分辨率为 12 位，最多可以测量 17 个外部信号、1 个内部温感信号、1 个内部参考电压（VREF1.2V）和一个 1/4 VDD 电压。通道的转换可选择单次、连续、扫描或不连续等采样模式，其采样结果存储在 16 位数据寄存器，数据存储格式可以选择左对齐或右对齐存储。

ADC 模块具有模拟看门狗特性，允许应用程序检测输入电压是否超过了用户设定的阈值上限或下限。

24.2 特性

- ◆ 可配置的转换分辨率（6/8/10/12 位）
- ◆ 支持单次或连续工作模式
- ◆ 在标准转换、插入转换结束后以及发生模拟看门狗或溢出事件时产生中断
- ◆ 用于自动将通道 0 转换为通道“n”的扫描模式
- ◆ 可配置的数据对齐方式
- ◆ 可独立设置各通道采样时间
- ◆ 可配置外部触发器选项，可为标准转换和插入转换配置极性
- ◆ 支持不连续采样模式
- ◆ 可配置的参考源选择
- ◆ 可配置的转换时钟分频
- ◆ 支持标准数据转换的 DMA 请求标准

24.3 结构框图

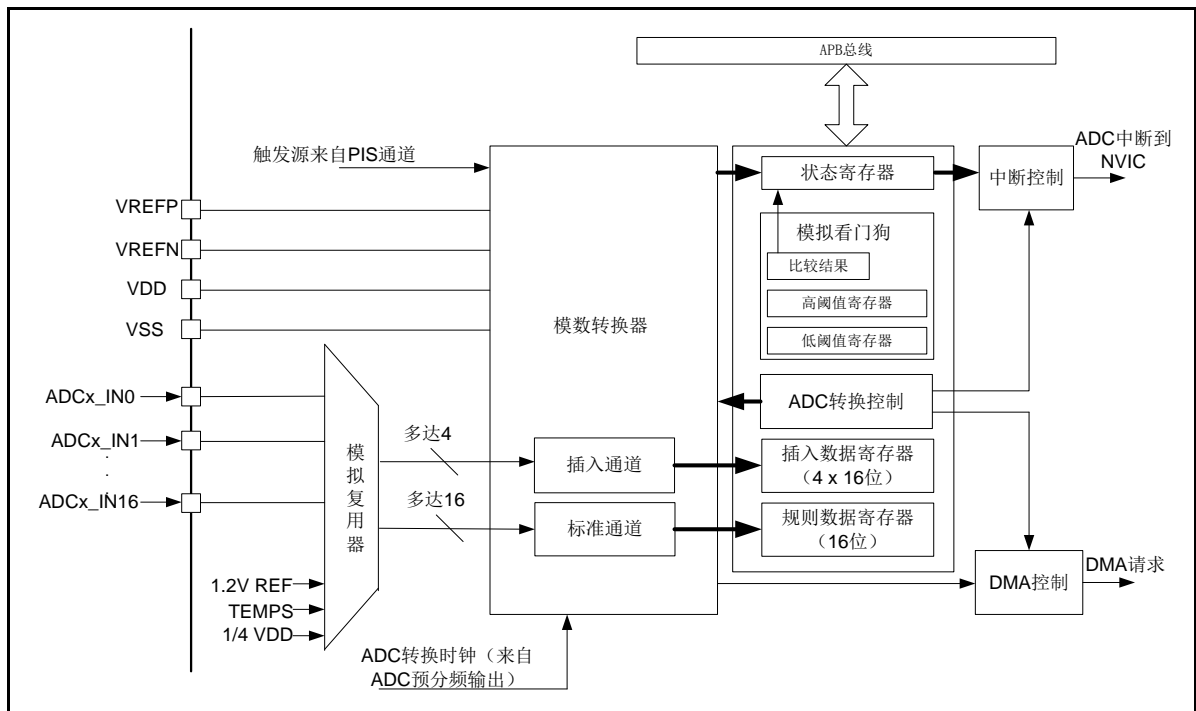


图 24-1 ADC 结构框图

24.4 功能描述

24.4.1 ADC控制

通过将 ADC_CON1 寄存器中的 ADCEN 位置 1 来使能 ADC。

通过将 ADC_CON1 寄存器中的 NCHTRG 或 ICHTRG 位置 1 来启动 AD 转换。

通过将 ADC_CON1 寄存器中的 ADCEN 位清零来关闭 ADC。

ADC 在使能后需要一定的建立时间，才能正常工作，所以在 ADCEN 使能后的第一次转换结果可能不准确，在应用中需要丢弃该次转换结果。

24.4.2 ADC时钟

ADC 内部工作需要两路时钟：

- ◇ 用于模拟电路的时钟：ADCCLK

此时钟来自于经可编程预分频器分频的 APB 时钟，该分频器可将 APB 时钟产生 2~256（若使能 1/3 预分频，分频系数再乘以 3）分频时钟供 ADC 模拟电路使用。

- ◇ 用于数字接口的时钟（用于寄存器读/写访问）

此时钟为 APB 时钟。

24.4.3 通道控制

外部的 16 条复用通道可分为两组：标准转换和插入转换，每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：AIN3、AIN8、AIN2、AIN2、AIN0、AIN2、AIN2、AIN15。

- ◇ 一个标准转换组最多由 16 个转换构成。必须在 ADC_NCHSx 寄存器中选择转换序列的标准通道及其顺序。标准转换组中的转换总数必须写入 ADC_CHSL 寄存器中的 NSL 位。
- ◇ 一个插入转换组最多由 4 个转换构成。必须在 ADC_ICHS 寄存器中选择转换序列的插入通道及其顺序。插入转换组中的转换总数必须写入 ADC_CHSL 寄存器中的 ISL 位。

如果在转换期间修改 ADC_NCHSx 或 ADC_ICHS 寄存器，将复位当前转换并向 ADC 发送一个新的启动脉冲，以转换新选择的组。

温感、1/4VDD、VREF 内部通道：

- ◇ 温感电压 VTS 连接到 ADC_IN17。
- ◇ 1/4 VDD 电压信号内部连接到通道 ADC_IN18。
- ◇ 内部参考电压 VREF1.2V 连接到 ADC_IN19。

24.4.4 单次工作模式

单次工作模式是指 ADC 执行一次转换，其工作流程如下：

- ◇ 将 ADC_CON1.CM 位清 0，然后通过后续方式来启动此模式：
- ◇ 将 ADC_CON1 寄存器中的 NCHTRG 位置 1（仅适用于标准通道）
- ◇ 将 ADC_CON1 寄存器中的 ICHTRG 位置 1（仅适用于插入通道）
- ◇ 外部触发（适用于标准通道或插入通道）

完成所选通道的转换之后：

- ◇ 使用标准通道：
 - 转换数据存储在 16 位 ADC_NCHDR 寄存器中
 - 标准转换结束标志 ADC_STAT.NCHE 置 1
 - 若标准转换完成中断使能位 ADC_CON0.NCHEIE 置 1，将产生中断
 - ◇ 使用插入通道：
 - 转换数据存储在 16 位 ADC_ICHDR 寄存器中
 - 插入转换结束标志 ADC_STAT.ICHE 置 1
 - 若插入转换完成中断使能位 ADC_CON0.ICHEIE 置 1，将产生中断
- 然后，ADC 停止。

24.4.5 连续工作模式

连续工作模式是指 ADC 执行一次转换任务后，立即执行下一个转换任务，其工作流程如下：

- ◇ 将 ADC_CON1.CM 位置 1
- ◇ 通过外部触发或 ADC_CON1.NCHTRG 置 1，来启动此模式（仅适用于标准通道）

每次转换之后：

- ◇ 如果转换了标准通道组：
 - 上次转换的数据存储在 16 位 ADC_NCHDR 寄存器中
 - 标准转换结束标志 ADC_STAT.NCHE 置 1
 - 若标准转换完成中断使能位 ADC_CON0.NCHEIE 置 1，将产生中断

注 1：连续转换模式不适用于插入组通道。

注 2：连续模式下唯一的例外情况是，在使能 ADC_CON0.IAUTO 时，插入通道组在标准通道之后自动转换。

24.4.6 时序图

在使能 ADC 后，需要一段稳定时间 t_{STAB} ，然后再启动 ADC 转换，并经过若干个时钟周期后 CHE 标志置 1，如下图所示。

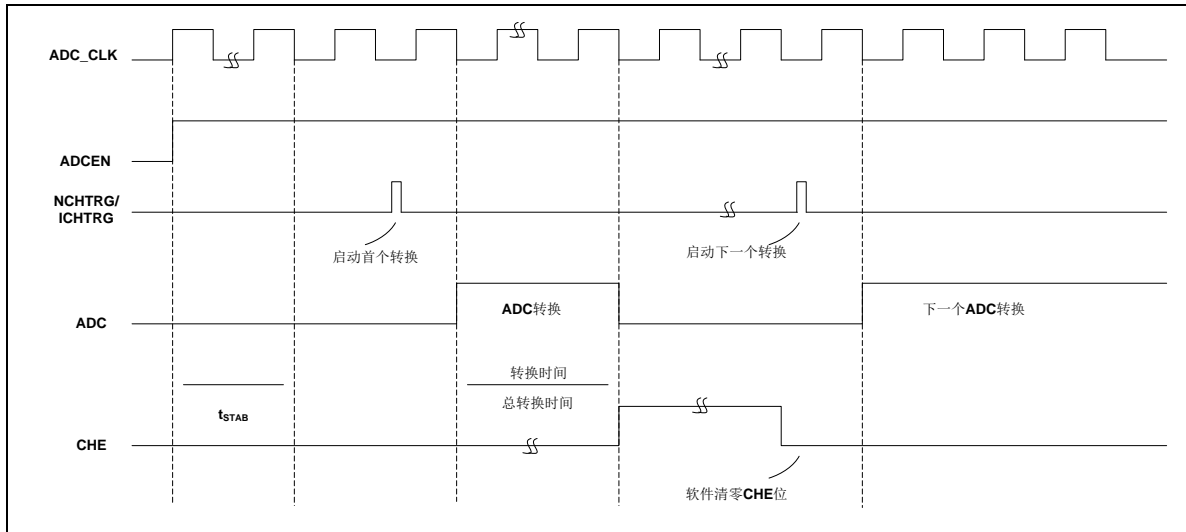


图 24-2 ADC 转换时序图

24.4.7 模拟看门狗

如果 ADC 转换的模拟电压低于阈值下限（寄存器 ADC_WDTL）或高于阈值上限（寄存器 ADC_HTR），则 AWD 模拟看门狗状态位 ADC_STAT.AWDF 位置 1。使能 ADC_CON0.AWDIE 位可以打开模拟看门狗中断。

上下限值寄存器是低 12 位有效的寄存器，而 ADC 结果寄存器可以左对齐和右对齐，但看门狗是在对齐之前将模拟电压与阈值上限和下限进行比较。

下表介绍了应如何配置 ADC_CON0 寄存器才能在一个或多个通道上使能模拟看门狗。

模拟看门狗保护通道	AWDSGL	NCHWDEN	ICHWDTEN
无	X	0	0
所有插入通道	0	0	1
所有标准通道	0	1	0
所有标准通道和插入通道	0	1	1
单个插入通道（AWDCH 选择）	1	0	1
单个标准通道（AWDCH 选择）	1	1	0
单个标准/插入通道（AWDCH 选择）	1	1	1

表 24-1 模拟看门狗通道选择

使用模拟看门狗流程

1. 设置 ADC_CON0.AWDSGL 位来选择看门狗作用于单个通道或一组通道，清 0 选择一组通道，置 1 选择单个通道。选择单个通道通过 AWDCH 选择哪个通道。
2. 置位 ADC_CON0.NCHWDEN 或 ADC_CON0.ICHWDTEN 选择使能标准组或插入组通道。

24.4.8 通道扫描

扫描模式是指连续依次转换标准组所有通道或者插入组所有通道。通过将 ADC_CON0.SCANEN 位置 1 来选择扫描模式。ADC 会扫描在 ADC_NCHSx 寄存器（对于标准通道）或 ADC_ICHS 寄存器（对于插入通道）中选择的所有通道，为组中的每个通道依次执行一次转换。如果将 CM 位置 1，并且是标准组扫描，ADC 会循环转换标准组所有通道。

如果将 ADC_CON1.DMA 位置 1，则在每次通道转换结束标志置位后通过 DMA 控制器将转换自标准通道组的数据寄存器（ADC_NCHDR）传输到 SRAM。

在以下情况下，ADC_STAT 寄存器中的 NCHE 位置 1：

- ◇ 如果 CON1.NCHESEL 位清零，在每个标准组序列转换结束时
- ◇ 如果 CON1.NCHESEL 位置 1，在每个标准通道转换结束时

从插入通道转换的数据始终存储在 ADC_ICHDRx 寄存器中。

24.4.9 插入通道控制

触发插入

要使用触发插入，必须将 ADC_CON0.IAUTO 位清零。使用触发插入时，必须确保触发事件之间的间隔长于插入序列。

在插入转换期间出现标准通道转换事件，插入转换不会被中断，标准转换在插入转换结束时执行。

在标准组转换期间触发插入：

1. 通过外部触发或将 ADC_CON1.NCHTRG 位置 1 来启动标准通道组转换。
2. 如果在标准通道组转换期间出现外部插入触发或者 ADC_CON1.ICHTRG 位置 1，则当前的转换会打断，并且插入通道序列会切换为单次扫描模式。
3. 然后，标准通道组的标准转换会从上次中断的标准转换处恢复。

在插入转换期间触发插入

自动插入

如果将 IAUTO 位置 1，则插入组中的通道会在标准组通道之后自动转换。这可用于转换最多由 20 个 AD 转换构成的序列，这些转换在 ADC_NCHSx 和 ADC_ICHS 寄存器中编程。

在此模式下，必须禁止插入通道上的外部触发。

如果 CM 位和 IAUTO 位均已置 1，则在转换标准通道之后会继续转换插入通道。

24.4.10 不连续采样控制

标准组

可将 ADC_CON0.NCHDCEN 位置 1 来使能此模式。该模式可用于转换含有 n ($n \leq 8$) 个转换的短序列，该短序列是在 ADC_NCHSx 寄存器中选择的转换序列的一部分。可通过写入 ADC_CON0.ETRGN 位来指定 n 的值。

出现外部触发时，将启动在 ADC_NCHSx 寄存器中选择的 n 个转换，直到序列中的所有转换均完成为止。通过 ADC_CHSL.NSL 位定义总序列长度。

示例：

n = 4, ADC_CHSL.NSL=9, 要转换的通道 = 0、2、3、5、6、7、8、10、11

第 1 次触发：转换序列 0、2、3、5

第 2 次触发：转换序列 6、7、8、10

第 3 次触发：转换序列 11 并生成 NCHE 事件

第 4 次触发：转换序列 0、2、3、5

在不连续采样模式下转换标准组时，最后一次触发剩余的通道不足 ADC_CON0.ETRGN 指定的通道数时，不会接着从头转换，转换到序列最后一个通道就会停止，生成 NCHE 事件。转换完所有序列通道后，下一个触发信号将启动序列的第一个通道。在上述示例中，第 4 次触发重新转换了序列的 0、2、3、5 通道。

插入组

可将 ADC_CON0 寄存器中的 ICHDCEN 位置 1 来使能此模式。在出现外部触发事件之后，可使用该模式逐通道 (n=1) 转换在 ADC_ICHS 寄存器中选择的序列。

出现外部触发时，将启动在 ADC_ICHS 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_CHSL 寄存器中的 ISL 位定义总序列长度。

示例：

n = 1, ADC_CHSL.ISL=3, 要转换的通道 = 1、3、4

第 1 次触发：转换通道 1

第 2 次触发：转换通道 3

第 3 次触发：转换通道 4 并生成 CHE 和 ICHE 事件

第 4 次触发：通道 1

转换完所有插入通道后，下一个触发信号将启动第一个插入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个插入通道。

不能同时使用自动插入和不连续采样模式。

不得同时为标准组和插入组设置不连续采样模式。只能选择其一进行不连续采样模式。

24.4.11 数据对齐

ADC_CON1.ALIGN 位用于选择转换后存储的数据的对齐方式,可选择左对齐和右对齐两种方式,如下图所示。

插入通道组的转换数据将加上 ADC_ICHOFFx 寄存器中写入的用户自定义偏移量,因此结果可以是一个负值,图中 EXS 位表示扩展的符号值。

对于标准组中的通道,不会减去任何偏移量,因此只有 12 位有效。

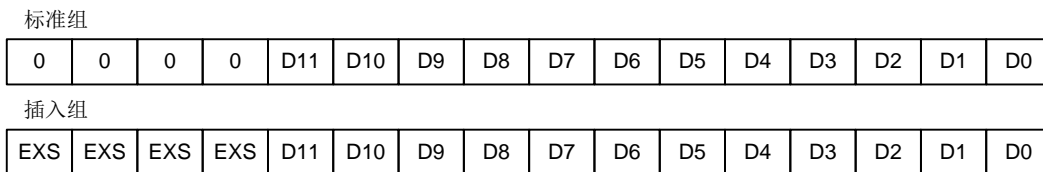


图 24-3 右对齐数据示意图

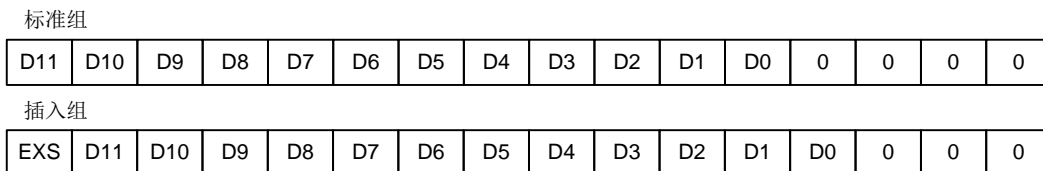


图 24-4 左对齐数据示意图

24.4.12 可独自设置各通道采样时间

ADC 会在数个 ADCCLK 周期内对输入电压进行采样,可使用 ADC_SMPT1、ADC_SMPT2 和 ADC_SMPT3 寄存器中的 CHTx 位修改采样周期数。每个通道均可以使用不同的采样时间进行采样。

总转换时间的计算公式如下:

$$T_{conv} = \text{采样时间} + 14 \text{ 个周期}$$

示例:

ADCCLK = 36MHz 且采样时间 = 4 个周期时:

$$T_{conv} = 4 + 14 = 18 \text{ 个周期} = 0.5\mu\text{S} \text{ (APBCLK 为 72MHz)}$$

24.4.13 外部触发转换和触发极性

可以通过配置 PIS 通道选择相应信号触发 ADC 转换, PIS 的配置请查看 PIS 相应章节描述。当 PIS 某一通道配置成 ADC 外部触发时,若相应的触发源事件发生,将自动触发标准序列或插入序列转换。

24.4.14 快速转换模式

可通过降低 ADC 分辨率来执行快速转换。RSEL 位用于选择数据寄存器中可用的位数。每种分辨率的最小转换时间如下：

- ◇ 12 位：4 + 14 = 18 个 ADCCLK 周期
- ◇ 10 位：4 + 12 = 16 个 ADCCLK 周期
- ◇ 8 位：4 + 10 = 14 个 ADCCLK 周期
- ◇ 6 位：4 + 8 = 12 个 ADCCLK 周期

24.4.15 数据管理

24.4.15.1 使用DMA

标准组只有一个数据寄存器 (ADC_NCHDR) 用于存储 AD 转换结果值，所以对于多个标准通道的转换，使用 DMA 可以快速存储数据，避免在上一次转换结果的值还未读出时新的 ADC 结果值又写入 ADC_CHDR 寄存器，造成数据丢失。

每完成标准通道组中的一个通道转换后，都会生成一个 DMA 请求。这样便可将转换的数据从 ADC_NCHDR 寄存器传输到软件指定的目标内存位置。

24.4.15.2 在不使用DMA的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，必须将 ADC_CON1 寄存器中的 NCHESEL 位置 1，才能使 NCHE 状态位在每次转换结束时置 1，而不仅是在序列结束时置 1。当 NCHESEL = 1 时，会自动使能溢出检测。因此，每当转换结束时，NCHE 都会置 1，并且可以读取 ADC_NCHDR 寄存器。如果数据丢失（溢出），则会将 ADC_STAT 寄存器中的 OVR 位置 1 并生成一个中断（如果 ADC_CON0.OVRIE 位已置 1）。

要在 NCHESEL 位置 1 时将 ADC 从 OVR 状态中恢复，请按以下步骤操作：

1. 将 ADC_STAT 寄存器中的 OVR 位清零
2. 触发 ADC 以开始转换

24.4.15.3 在不使用DMA和溢出检测情况下进行转换

当 ADC 存在转换一个或多个通道时不需要每次读取数据的情况时，例如使用模拟看门狗，可将 ADC_CON1.OVRDIS 置 1，并且仅在序列结束（NCHESEL = 0）时才将 NCHE 位置 1。这样溢出检测被禁止。

24. 4. 16 ADC中断

当模拟看门狗状态位和溢出状态位分别置 1 时，标准组和插入组在转换结束时可能会产生中断。可以使用单独的中断使能位以实现灵活性。

中断事件	事件标志位	使能控制位
结束标准组的转换	NCHE	NCHEIE
结束插入组的转换	ICHE	ICHEIE
发生模拟看门狗事件	AWDF	AWDIE
溢出	OVR	OVRIE

表 24-2 ADC 中断

注：ADC 中断源为脉冲信号，即触发 CLIC 的 ADC 中断向量后，ADC 中断源信号由硬件自动清零，在应用时需设置 ADC 中断向量对应的中断属性寄存器 CLICINTATTR 中的 TRIG=2'b01，将 ADC 中断触发方式设置为上升沿触发中断。

24.5 特殊功能寄存器

24.5.1 寄存器列表

ADC 寄存器列表			
名称	偏移地址	类型	描述
ADC_STAT	000 _H	R	ADC 状态寄存器
ADC_CLR	004 _H	W	ADC 清零寄存器
ADC_CON0	008 _H	R/W	ADC 控制寄存器 0
ADC_CON1	00C _H	R/W	ADC 控制寄存器 1
ADC_SMPT1	010 _H	R/W	ADC 采样时间寄存器 1
ADC_SMPT2	014 _H	R/W	ADC 采样时间寄存器 2
ADC_SMPT3	018 _H	R/W	ADC 采样时间寄存器 3
ADC_NCHOFF	01C _H	R/W	ADC 标准通道数据偏移寄存器
ADC_ICHOFF1	020 _H	R/W	ADC 插入通道数据偏移寄存器 1
ADC_ICHOFF2	024 _H	R/W	ADC 插入通道数据偏移寄存器 2
ADC_ICHOFF3	028 _H	R/W	ADC 插入通道数据偏移寄存器 3
ADC_ICHOFF4	02C _H	R/W	ADC 插入通道数据偏移寄存器 4
ADC_NCHS1	030 _H	R/W	ADC 标准通道序列寄存器 1
ADC_NCHS2	034 _H	R/W	ADC 标准通道序列寄存器 2
ADC_NCHS3	038 _H	R/W	ADC 标准通道序列寄存器 3
ADC_NCHS4	03C _H	R/W	ADC 标准通道序列寄存器 3
ADC_ICHS	040 _H	R/W	ADC 插入通道序列寄存器
ADC_CHSL	044 _H	R/W	ADC 通道序列长度寄存器
ADC_WDTH	048 _H	R/W	ADC 看门狗高阈值寄存器
ADC_WDTL	04C _H	R/W	ADC 看门狗低阈值寄存器
ADC_ICHDR1	050 _H	R	ADC 插入通道数据寄存器 1
ADC_ICHDR2	054 _H	R	ADC 插入通道数据寄存器 2
ADC_ICHDR3	058 _H	R	ADC 插入通道数据寄存器 3
ADC_ICHDR4	05C _H	R	ADC 插入通道数据寄存器 4
ADC_NCHDR	060 _H	R	ADC 标准通道数据寄存器
ADC_CCR	064 _H	R/W	ADC 通用控制寄存器

24.5.2 寄存器描述

24.5.2.1 ADC状态寄存器 (ADC_STAT)

ADC 状态寄存器 (ADC_STAT)																																
偏移地址: 00H																																
复位值: 00000000_00000000_00000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																					ICHS		NCHS		Reserved				OVR	ICHE	NCHE	AWDF

Reserved	Bit 31-10	—	保留
ICHS	Bit 9	R	插入通道转换开始标志位 0: 未开始插入转换或标志位已被清除 1: 已开始插入转换 注: 该位由硬件置1, 通过操作ADC_CLR清零
NCHS	Bit 8	R	标准通道转换开始标志位 0: 未开始标准转换或标志位已被清除 1: 已开始标准转换 注: 该位由硬件置1, 通过操作ADC_CLR清零
Reserved	Bit 7-4	—	保留
OVR	Bit 3	R	转换溢出标志位 0: 未发生溢出或标志位已被清除 1: 发生溢出 注 1: 溢出检测仅在 DMA=1 或 NCHESEL=1 时使能 注 2: 该位由硬件置 1, 通过操作 ADC_CLR 清零
ICHE	Bit 2	R	插入通道转换结束标志位 0: 所有插入转换未完成或标志位已被清除 1: 所有插入转换已完成 注: 该位由硬件置 1, 通过操作 ADC_CLR 清零
NCHE	Bit 1	R	标准通道转换结束标志位 NCHESEL = 0时 0: 标准转换序列未完成或标志位已被清除 1: 标准转换序列已完成 NCHESEL = 1 时 0: 单次标准转换未完成或标志位已被清除 1: 单次标准转换已完成 注: 该位由硬件置1, 通过操作ADC_CLR清零
AWDF	Bit 0	R	模拟看门狗标志位 0: 未发生看门狗事件或标志位已被清除 1: 已发生看门狗事件 注: 该位由硬件置1, 通过操作ADC_CLR清零

24.5.2.2 ADC清零寄存器 (ADC_CLR)

ADC 清零寄存器 (ADC_CLR)																																				
偏移地址: 04 _H																																				
复位值: 00000000_00000000_00000000_00000000 _B																																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved																						ICHS		NCHS		Reserved							OVR	ICHE	NCHE	AWDF

Reserved	Bit 31-10	—	保留
ICHS	Bit 9	W1	插入通道开始标志位清零 0: 无操作 1: 插入转换开始标志位清零
NCHS	Bit 8	W1	标准通道开始标志位清零 0: 无操作 1: 标准转换开始标志位清零
Reserved	Bit 7-4	—	保留
OVR	Bit 3	W1	转换溢出标志位清零 0: 无操作 1: 转换溢出标志位清零
ICHE	Bit 2	W1	插入转换结束标志位清零 0: 无操作 1: 插入转换结束标志位清零
NCHE	Bit 1	W1	标准转换结束标志位清零 0: 无操作 1: 标准转换结束标志位清零
AWDF	Bit 0	W1	模拟看门狗标志位清零 0: 无操作 1: 模拟看门狗标志位清零

24.5.2.3 ADC控制寄存器0 (ADC_CON0)

ADC 控制寄存器 0 (ADC_CON0)																															
偏移地址: 08 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OVRIE	RSEL		NCHWDEN	ICHWDTEN	Reserved				CNTW			ETRGN			ICHDCEN	NCHDCEN	IAUTO	AWDSGL	SCANEN	ICHEIE	AWDIE	NCHEIE	AWDCH				

Reserved	Bit 31-27	—	保留
OVRIE	Bit 26	R/W	溢出中断使能位 0: 禁止 1: 使能
RSEL	Bit 25-24	R/W	ADC 转换分辨率选择位 00: 6 位, 转换结果数据位为 D11~D6 01: 8 位, 转换结果数据位为 D11~D4 10: 10 位, 转换结果数据位为 D11~D2 11: 12 位, 转换结果数据位为 D11~D0
NCHWDEN	Bit 23	R/W	标准通道看门狗使能位 0: 禁止 1: 使能
ICHWDTEN	Bit 22	R/W	插入通道看门狗使能位 0: 禁止 1: 使能
Reserved	Bit 21-19	—	保留
CNTW	Bit 18-16	R/W	通道切换等待时间 000: 0 个 ADC 时钟 001: 1 个 ADC 时钟 ... 111: 7 个 ADC 时钟
ETRGN	Bit 15-13	R/W	外部触发不连续转换通道数 000: 1 个通道 001: 2 个通道 111: 8 个通道
ICHDCEN	Bit 12	R/W	插入通道不连续转换使能位 0: 禁止 1: 使能
NCHDCEN	Bit 11	R/W	标准通道不连续转换使能位 0: 禁止 1: 使能
IAUTO	Bit 10	R/W	插入组自动转换使能位

			0: 禁止 1: 使能
AWDSGL	Bit 9	R/W	扫描模式单一通道模拟看门狗使能位 0: 禁止 1: 使能
SCANEN	Bit 8	R/W	扫描模式使能位 0: 禁止 1: 使能
ICHEIE	Bit 7	R/W	插入通道转换完成中断使能位 0: 禁止 1: 使能
AWDIE	Bit 6	R/W	模拟看门狗中断使能位 0: 禁止 1: 使能
NCHEIE	Bit 5	R/W	标准通道转换完成中断使能位 0: 禁止 1: 使能
AWDCH	Bit 4-0	R/W	模拟看门狗通道选择位 00000: ADC 输入通道 0 00001: ADC 输入通道 1 10011: ADC 输入通道 19 其他: 保留

注: ADC 中断源为脉冲信号, 即触发 CLIC 的 ADC 中断向量后, ADC 中断源信号由硬件自动清零, 在应用时需设置 ADC 中断向量对应的中断属性寄存器 CLICINTATTR 中的 TRIG=2'b01, 将 ADC 中断触发方式设置为上升沿触发中断。

24.5.2.4 ADC控制寄存器 1 (ADC_CON1)

ADC 控制寄存器 1 (ADC_CON1)																															
偏移地址: 0C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	NCHTRG	NETS		Reserved				ICHTRG	IETS		Reserved						ALIGN	NCHESEL	Reserved	DMA	Reserved				CM	ADCEN					

Reserved	Bit 31	—	保留
NCHTRG	Bit 30	W1	标准通道触发位 0: 无操作 1: 触发开始标准通道转换
NETS	Bit 29-28	R/W	标准转换外部触发极性选择位 00: 外部触发禁止 01: 上升沿触发 10: 下降沿触发 11: 上升沿和下降沿触发
Reserved	Bit 27-23	—	保留
ICHTRG	Bit 22	W1	插入通道触发位 0: 无操作 1: 触发开始插入通道转换
IETS	Bit 21-20	R/W	插入转换外部触发极性选择位 00: 外部触发禁止 01: 上升沿触发 10: 下降沿触发 11: 上升沿和下降沿触发
Reserved	Bit 19-12	—	保留
ALIGN	Bit 11	R/W	数据对齐方式位 0: 右对齐 1: 左对齐
NCHESEL	Bit 10	R/W	标准转换结束标志选择位 0: 标准转换序列结束时将 STAT.NCHE 位置 1 1: 单次标准转换结束时将 STAT.NCHE 位置 1
Reserved	Bit 9	—	保留
DMA	Bit 8	R/W	DMA 访问使能位 0: 禁止 1: 使能
Reserved	Bit 7-2	—	保留
CM	Bit 1	R/W	转换模式 0: 单次转换 1: 连续转换

ADCEN	Bit 0	R/W	ADC 使能位 0: 禁止 1: 使能
-------	-------	-----	----------------------------------

注：因 ADC 在使能后需要一定的建立时间，才能正常工作，所以在 ADCEN 使能后的第一次转换结果可能不准确，在应用中需要丢弃该次转换结果。

24.5.2.5 ADC采样时间寄存器 1 (ADC_SMPT1)

ADC 采样时间寄存器 1 (ADC_SMPT1)																															
偏移地址: 10 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHT7				CHT6				CHT5				CHT4				CHT3				CHT2				CHT1				CHT0			

CHT<y>	Bit 31-0	R/W	通道 y 采样时间选择位 (y=0..7) 0000: 4 个 Tadclk 周期 0001: 6 个 Tadclk 周期 0010: 10 个 Tadclk 周期 0011: 18 个 Tadclk 周期 其他: 预留
--------	----------	-----	--

24.5.2.6 ADC采样时间寄存器 2 (ADC_SMPT2)

ADC 采样时间寄存器 2 (ADC_SMPT2)																															
偏移地址: 14 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHT15				CHT14				CHT13				CHT12				CHT11				CHT10				CHT9				CHT8			

CHT<y>	Bit 31-0	R/W	通道 y 采样时间选择位 (y=8..15) 0000: 4 个 Tadclk 周期 0001: 6 个 Tadclk 周期 0010: 10 个 Tadclk 周期 0011: 18 个 Tadclk 周期 其他: 预留
--------	----------	-----	---

24.5.2.7 ADC采样时间寄存器 3 (ADC_SMPT3)

ADC 采样时间寄存器 3 (ADC_SMPT3)																															
偏移地址: 18 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																CHT19		CHT18		CHT17		CHT16									

CHT<y>	Bits 31-0	R/W	通道 y 采样时间选择位 (y=16..19) 0000: 4 个 Tadclk 周期 0001: 6 个 Tadclk 周期 0010: 10 个 Tadclk 周期 0011: 18 个 Tadclk 周期 其他: 预留
--------	-----------	-----	--

24. 5. 2. 8 ADC标准通道数据偏移寄存器 (ADC_NCHOFF)

ADC 标准通道数据偏移寄存器 (ADC_NCHOFF)																															
偏移地址: 1C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																NOFF															

Reserved	Bit 31-12	—	保留
NOFF	Bit 11-0	R/W	标准通道数据偏移量 ADC_NCHDR 中的数据为原始转换数据加上偏移量

24.5.2.9 ADC插入通道数据偏移寄存器 1 (ADC_ICHOFF1)

ADC 插入通道数据偏移寄存器 1 (ADC_ICHOFF1)																															
偏移地址: 20 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																IOFF1															

Reserved	Bit 31-12	—	保留
IOFF1	Bit 11-0	R/W	插入通道 1 数据偏移量 ADC_ICHDR1 中的数据为原始转换数据加上偏移量

24.5.2.10 ADC插入通道数据偏移寄存器 2 (ADC_ICHOFF2)

ADC 插入通道数据偏移寄存器 2 (ADC_ICHOFF2)																															
偏移地址: 24 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																IOFF2															

Reserved	Bit 31-12	—	保留
IOFF2	Bit 11-0	R/W	插入通道 2 数据偏移量 ADC_ICHDR2 中的数据为原始转换数据加上偏移量

24.5.2.11 ADC插入通道数据偏移寄存器 3 (ADC_ICHOFF3)

ADC 插入通道数据偏移寄存器 3 (ADC_ICHOFF3)																															
偏移地址: 28 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																IOFF3															

Reserved	Bit 31-12	—	保留
IOFF3	Bit 11-0	R/W	插入通道 3 数据偏移量 ADC_ICHDR3 中的数据为原始转换数据加上偏移量

24.5.2.12 ADC插入通道数据偏移寄存器 4 (ADC_ICHOFF4)

ADC 插入通道数据偏移寄存器 4 (ADC_ICHOFF4)																															
偏移地址: 2C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																IOFF4															

Reserved	Bit 31-12	—	保留
IOFF4	Bit 11-0	R/W	插入通道 4 数据偏移量 ADC_ICHDR4 中的数据为原始转换数据加上偏移量

24.5.2.13 ADC标准通道序列寄存器 1 (ADC_NCHS1)

ADC 标准通道序列寄存器 1 (ADC_NCHS1)																															
偏移地址: 30 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		NS4				Reserved		NS3				Reserved		NS2				Reserved		NS1											

Reserved	Bit 31-29	—	保留
NS4	Bit 28-24	R/W	标准序列第 4 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 23-21	—	保留
NS3	Bit 20-16	R/W	标准序列第 3 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 15-13	—	保留
NS2	Bit 12-8	R/W	标准序列第 2 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 7-5	—	保留
NS1	Bit 4-0	R/W	标准序列第 1 次转换通道编号 00000~10011: 通道 0~19 其他: 预留

24.5.2.14 ADC标准通道序列寄存器 2 (ADC_NCHS2)

ADC 标准通道序列寄存器 2 (ADC_NCHS2)																															
偏移地址: 34 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NS8				Reserved				NS7				Reserved				NS6				Reserved				NS5			

Reserved	Bit 31-29	—	保留
NS8	Bit 28-24	R/W	标准序列第 8 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 23-21	—	保留
NS7	Bit 20-16	R/W	标准序列第 7 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 15-13	—	保留
NS6	Bit 12-8	R/W	标准序列第 6 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 7-5	—	保留
NS5	Bit 4-0	R/W	标准序列第 5 次转换通道编号 00000~10011: 通道 0~19 其他: 预留

24.5.2.15 ADC标准通道序列寄存器 3 (ADC_NCHS3)

ADC 标准通道序列寄存器 3 (ADC_NCHS3)																															
偏移地址: 38 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			NS12				Reserved			NS11				Reserved			NS10				Reserved			NS9							

Reserved	Bit 31-29	—	保留
NS12	Bit 28-24	R/W	标准序列第 12 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 23-21	—	保留
NS11	Bit 20-16	R/W	标准序列第 11 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 15-13	—	保留
NS10	Bit 12-8	R/W	标准序列第 10 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 7-5	—	保留
NS9	Bit 4-0	R/W	标准序列第 9 次转换通道编号 00000~10011: 通道 0~19 其他: 预留

24.5.2.16 ADC标准通道序列寄存器 4 (ADC_NCHS4)

ADC 标准通道序列寄存器 4 (ADC_NCHS4)																															
偏移地址: 3C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NS16				Reserved				NS15				Reserved				NS14				Reserved				NS13			

Reserved	Bit 31-29	—	保留
NS16	Bit 28-24	R/W	标准序列第 16 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 23-21	—	保留
NS15	Bit 20-16	R/W	标准序列第 15 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 15-13	—	保留
NS14	Bit 12-8	R/W	标准序列第 14 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 7-5	—	保留
NS13	Bit 4-0	R/W	标准序列第 13 次转换通道编号 00000~10011: 通道 0~19 其他: 预留

24.5.2.17 ADC插入通道序列寄存器 (ADC_ICHS)

ADC 插入通道序列寄存器 (ADC_ICHS)																															
偏移地址: 40 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				IS4				Reserved				IS3				Reserved				IS2				Reserved				IS1			

Reserved	Bit 31-29	—	保留
IS4	Bit 28-24	R/W	插入序列第 4 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 23-21	—	保留
IS3	Bit 20-16	R/W	插入序列第 3 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 15-13	—	保留
IS2	Bit 12-8	R/W	插入序列第 2 次转换通道编号 00000~10011: 通道 0~19 其他: 预留
Reserved	Bit 7-5	—	保留
IS1	Bit 4-0	R/W	插入序列第 1 次转换通道编号 00000~10011: 通道 0~19 其他: 预留

24. 5. 2. 18 ADC通道序列长度寄存器 (ADC_CHSL)

ADC 通道序列长度寄存器 (ADC_CHSL)																															
偏移地址: 44 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																					ISL		Reserved				NSL				

Reserved	Bit 31-10	—	保留
ISL	Bit 9-8	R/W	插入通道序列长度 00: 1 次转换 01: 2 次转换 10: 3 次转换 11: 4 次转换
Reserved	Bit 7-4	—	保留
NSL	Bit 3-0	R/W	标准通道列长度 0000: 1 次转换 0001: 2 次转换 1111: 16 次转换

24.5.2.19 ADC看门狗高阈值寄存器 (ADC_WDTH)

ADC 看门狗高阈值寄存器 (ADC_WDTH)																															
偏移地址: 48 _H																															
复位值: 00000000_00000000_00001111_11111111 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																HT															

Reserved	Bit 31-12	—	保留
HT	Bit 11-0	R/W	模拟看门狗高阈值 当原始数据大于高阈值时产生看门狗事件

24.5.2.20 ADC看门狗低阈值寄存器 (ADC_WDTL)

ADC 看门狗低阈值寄存器 (ADC_WDTL)																															
偏移地址: 4C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																LT															

Reserved	Bit 31-12	—	保留
LT	Bit 11-0	R/W	模拟看门狗低阈值 当原始数据小于低阈值时产生看门狗事件

24.5.2.21 ADC插入通道数据寄存器 1 (ADC_ICHDR1)

ADC 插入通道数据寄存器 1 (ADC_ICHDR1)																															
偏移地址: 50 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																VAL															

Reserved	Bit 31-16	—	保留
VAL	Bit 15-0	R	插入通道 1 转换数据 该数据为对齐之后的数据

24.5.2.22 ADC插入通道数据寄存器 2 (ADC_ICHDR2)

ADC 插入通道数据寄存器 2 (ADC_ICHDR2)																															
偏移地址: 54 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																VAL															

Reserved	Bit 31-16	—	保留
VAL	Bit 15-0	R	插入通道 2 转换数据 该数据为对齐之后的数据

24.5.2.23 ADC插入通道数据寄存器 3 (ADC_ICHDR3)

ADC 插入通道数据寄存器 3 (ADC_ICHDR3)																															
偏移地址: 58 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																VAL															

Reserved	Bit 31-16	—	保留
VAL	Bit 15-0	R	插入通道 3 转换数据 该数据为对齐之后的数据

24.5.2.24 ADC插入通道数据寄存器 4 (ADC_ICHDR4)

ADC 插入通道数据寄存器 4 (ADC_ICHDR4)																															
偏移地址: 5C _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																VAL															

Reserved	Bit 31-16	—	保留
VAL	Bit 15-0	R	插入通道 4 转换数据 该数据为对齐之后的数据

24.5.2.25 ADC标准通道数据寄存器 (ADC_NCHDR)

ADC 标准通道数据寄存器 (ADC_NCHDR)																															
偏移地址: 60 _H																															
复位值: 00000000_00000000_00000000_00000000 _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																VAL															

Reserved	Bit 31-16	—	保留
VAL	Bit 15-0	R	标准通道转换数据 该数据为对齐之后的数据

24.5.2.26 ADC通用控制寄存器 (ADC_CCR)

ADC 通用控制寄存器 (ADC_CCR)																																
偏移地址: 64 _H																																
复位值: 00000000_00000000_11000000_00000000 _B																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	CALZERO	TRMEN	Reserved													TSEN	Reserved	VRNSEL	VRPSEL	Reserved	PWRMODESEL	Reserved			IREFEN	Reserved	VREFEN	Reserved			PREDIV	POSDIV

Reserved	Bit 31-30	—	保留
CALZERO	Bit 29	R/W	ADC 调零使能位 (仅测试用) 0: 禁止 (应用时需固定设置为 0) 1: 使能
TRMEN	Bit 28	R/W	ADC 数据修调使能位 (仅测试用) 0: 禁止 (应用时需固定设置为 0) 1: 使能
Reserved	Bit 27-21	—	保留
TSEN	Bit 20	—	温感使能位 0: 禁止 1: 使能
Reserved	Bit 19	—	保留
VRNSEL	Bit 18	R/W	负向参考电压选择位 0: VREFN 管脚 (PC0) 1: VSS
VRPSEL	Bit 17	R/W	正向参考电压选择位

			0: VDD 1: VREFP 管脚 (PC1)
Reserved	Bit 16	—	保留
PWRMODSEL	Bit 15	R/W	ADC 工作模式选择 0: 高速模式 (需软件固定设置为 0) 1: 低速模式 (仅用于测试)
Reserved	Bit 14-12	—	保留
IREFEN	Bit 11	R/W	偏置电流使能位 0: 禁止 1: 使能 (ADC 工作时, 必须设置为 1)
Reserved	Bit10-9	—	保留
VREFEN	Bit 8	R/W	内部基准电压 1.2V 使能位 0: 禁止 1: 使能
Reserved	Bit7-4	—	保留
PREDIV	Bit 3	R/W	ADC 时钟预分频 1/3 使能位(基于 PCLK 预分频) 0: 禁止 1: 使能
POSDIV	Bit 2-0	R/W	ADC 时钟后分频选择位(基于 1/3 预分频后时钟) 000: 2 分频 001: 4 分频 111: 256 分频

注 1: ADC 正常工作时, 需使能 IREFEN=1, 否则会导致 ADC 工作异常; 在 STOP 模式下 ADC 不工作, 需设置 IREFEN=0, 否则会增大芯片功耗;

注 2: ADC 工作时, 需设置 PWRMODSEL=0, 选择为高速模式, 提高转换结果精度。

第25章 调试控制 (DBGC)

25.1 概述

ES32VF2264 系列 MCU 使用的内核是 XuanTie E902 (RISC-V 架构)，该内核包含用于高级调试功能的硬件。利用这些调试功能，可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核在停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，可以恢复内核和系统，并恢复程序执行。

当调试器与 MCU 相连并进行调试时，将使用内核的硬件调试模块。

ES32VF2264 系列 MCU 提供 CJTAG 调试接口，支持 IEEE1149.7 标准。

25.2 特性

- ◆ 支持 SW-DP：调试端口电路，实现 DAP 电路和外部调试主机的通讯
- ◆ 支持 MEM-AP：访问端口电路，实现 DAP 电路与被调试单元的通讯
- ◆ 支持断点 (Breakpoint)：4 个断点
- ◆ 支持数据观测和追踪 (DWT)：2 个数据观测点

25.3 结构框图

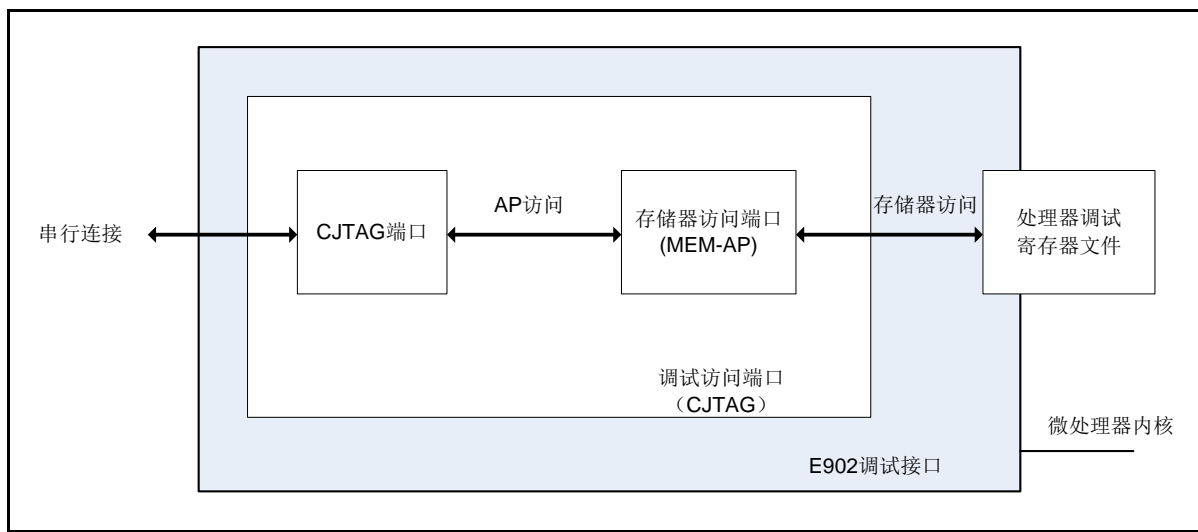


图 25-1 CJTAG 调试结构图

25.4 功能描述

25.4.1 调试端口

下表为 CJTAG 调试用端口，芯片上电后默认作为调试功能使用。

端口功能	输入/输出	说明
SWCLK	输入	调试时钟。该端口在调试模式下为调试电路提供通信时钟。
SWDIO	输入/输出	调试数据输入/输出端口。用于与外部调试主机的数据交互。

表 25-1 CJTAG 端口描述

注：若 CJTAG 调试口没有复用其他功能，用户产品量产时应将对应的两个调试口配置为 GPIO 模式（FUNC1），并设置输出 0，避免应用时输入悬空。

25.4.2 调试冻结

程序开发过程中，会经历反复调试，通过调试工具对运行程序进行暂停，实际上在暂停内核的时候，仍有部分硬件外设在工作（如：定时器、看门狗等），影响调试效果；DBG_C 模块可以实现内核和外设同时暂停，实现高级调试功能。

操作示例：

1. 按照需求正确初始化使用的外设，如定时器 BS16T
2. 为保证程序在调试过程中暂停时 BS16T 计数同时暂停，需设置 `DBG_APB1FZ.BS16T0_STOP = 1`
3. 如果未设置 `DBG_APB1FZ.BS16T0_STOP = 1`，程序在调试过程中暂停时，实际 BS16T 仍然在计数。

25.4.3 调试复位

内核调试电路和调试控制寄存器只可被上电复位、欠压复位及软件复位中的芯片全局复位（`RMU_AHB2RSTR.CHIPRST`）所复位。

25.4.4 MEM-AP访问端口

MEM-AP 端口，用于访问被调试单元的存储器映射区域。

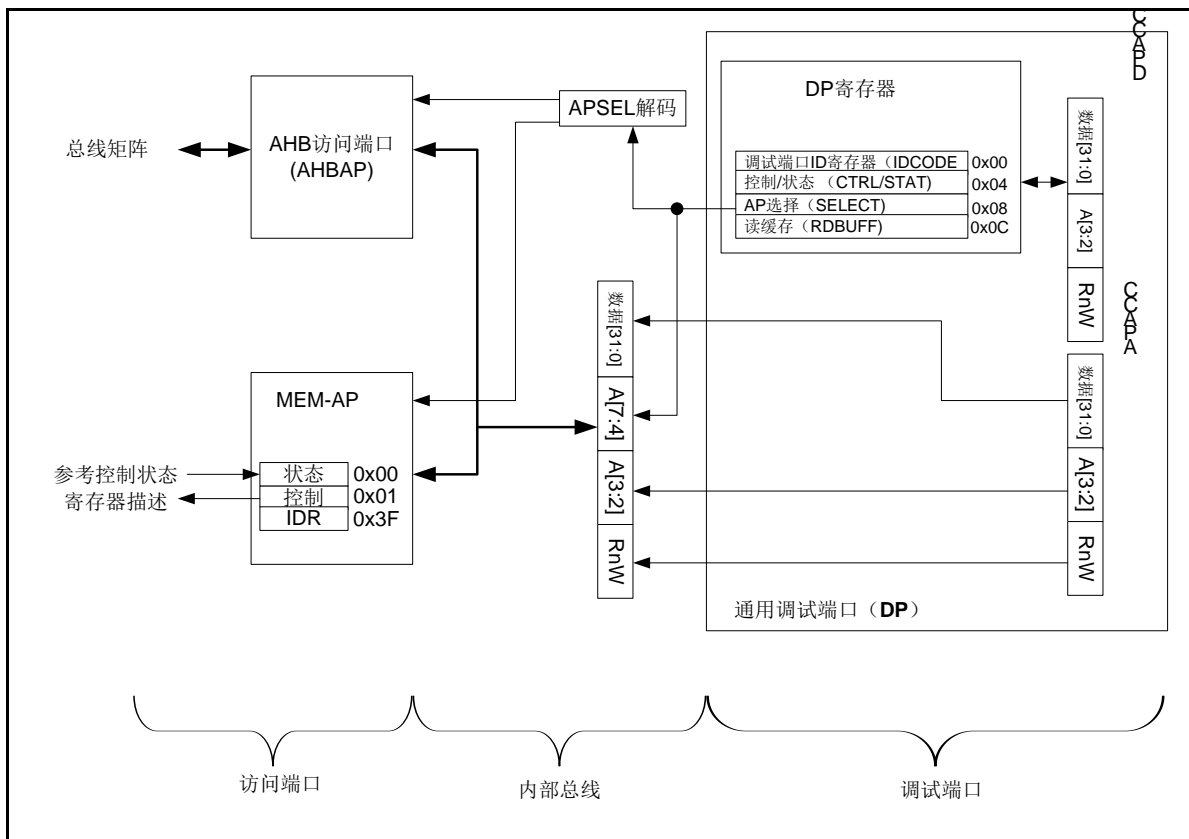


图 25-2 MEM-AP 地址映射

25.5 特殊功能寄存器

25.5.1 寄存器列表

DBGC 寄存器列表		
名称	偏移地址	描述
DBG_IDCODE	000 _H	DBG 器件识别码
Reserved	004 _H	保留
DBG_APB1FZ	008 _H	APB1 外设调试冻结寄存器
DBG_APB2FZ	00C _H	APB2 外设调试冻结寄存器

25.5.2 寄存器描述

25.5.2.1 DBG器件识别码 (DBG_IDCODE)

DBG 器件识别码寄存器 (DBG_IDCODE)																															
偏移地址: 000 _H																															
上电复位值: xxxxxxxx_xxxxxxxx_xxxxxxxx_xxxxxxxx _B																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV_ID																CORE_ID				DEV_ID											

REV_ID	Bit 31-16	R	版本识别码 0x1000: 版本 A
CORE_ID	Bit 15-12	R	内核识别码 0x0: E902
DEV_ID	Bit 11-0	R	器件识别码 0x032: MCU 识别码

25.5.2.2 APB1 外设调试冻结寄存器 (DBG_APB1FZ)

APB1 外设调试冻结寄存器 (DBG_APB1FZ)																																	
偏移地址: 008 _H																																	
上电复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved																							GP16C4Tx_STOP		Reserved					BS16T0_STOP		AD16C4T0_STOP	

Reserved	Bit 31-9	—	保留
GP16C4Tx_STOP	Bit 8	R/W	GP16C4T0/1/2 调试暂停选择位 0: 内核停止时, 仍正常计数 1: 内核停止时, 暂停计数
Reserved	Bit 7-2	—	保留
BS16T0_STOP	Bit 1	R/W	BS16T0 调试暂停选择位 0: 内核停止时, 仍正常计数 1: 内核停止时, 暂停计数
AD16C4T0_STOP	Bit 0	R/W	AD16C4T0 调试暂停选择位 0: 内核停止时, 仍正常计数 1: 内核停止时, 暂停计数

注: 该寄存器仅支持按字写入。

25.5.2.3 APB2 外设调试冻结寄存器 (DBG_APB2FZ)

APB2 外设调试冻结寄存器 (DBG_APB2FZ)																																	
偏移地址: 00C _H																																	
上电复位值: 00000000_00000000_00000000_00000000 _B																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved																							WWDT_STOP	IWDT_STOP	Reserved					I2C1_STOP		I2C0_STOP	

Reserved	Bit 31-10	—	保留
WWDT_STOP	Bit 9	R/W	WWDT 调试暂停选择位 0: 内核停止时, 仍正常计数 1: 内核停止时, 暂停计数
IWDT_STOP	Bit 8	R/W	IWDT 调试暂停选择位 0: 内核停止时, 仍正常计数

			1: 内核停止时, 暂停计数
Reserved	Bit 7-2	—	保留
I2C1_STOP	Bit 1	R/W	I2C1 调试暂停选择位 0: 内核停止时, SMBus 超时计数器仍正常计数 1: 内核停止时, SMBus 超时计数器暂停计数
I2C0_STOP	Bit 0	R/W	I2C0 调试暂停选择位 0: 内核停止时, SMBus 超时计数器仍正常计数 1: 内核停止时, SMBus 超时计数器暂停计数

注: 该寄存器仅支持按字写入。

第26章 FLASH信息区

26.1 概述

FLASH 信息区用来存储芯片的只读信息和配置信息。

用户在程序中只能对只读信息和配置信息进行读操作。可以使用芯片烧录工具对配置信息进行修改，但无法改变只读信息。

26.2 特性

- ◆ FLASH 信息区存储的只读信息包括：
 - ◇ 产品识别码
 - ◇ 芯片唯一码
- ◆ FLASH 信息区存储的配置信息包括：
 - ◇ 芯片配置字
 - ◇ 写保护区域配置字
 - ◇ 数据区配置字
 - ◇ 全局读保护配置字
 - ◇ 用户程序校验码

26.3 功能描述

26.3.1 FLASH信息区只读信息

FLASH 信息区的基地址为 0x0004_0000，可以字读取。

26.3.1.1 芯片唯一码UID

芯片唯一码 UID 为 96 位，每一颗芯片都是唯一的编码，可以用做：

- ◇ 终端产品序列号
- ◇ 通过特定的加密算法生成安全密钥

寄存器名称	芯片唯一码 0 (UID0)	
地址偏移	09E0 _H	
UID0	Bit 31-0	芯片唯一码 0

寄存器名称	芯片唯一码 1 (UID1)	
地址偏移	09E8 _H	
UID1	Bit 31-0	芯片唯一码 1

寄存器名称	芯片唯一码 2 (UID2)	
地址偏移	09F0 _H	
UID2	Bit 31-0	芯片唯一码 2

26.3.1.2 芯片产品识别码CHIPID

CHIPID 用来区分芯片产品型号，为 32 位编码。

寄存器名称	芯片产品识别码 (CHIPID)	
地址偏移	0BF8 _H	
CHIPID	Bit 31-0	CHIPID

26.3.2 FLASH信息区配置信息

FLASH 信息区的配置信息在芯片程序运行前生效。

FLASH 信息区可以字读取。只能使用芯片烧录工具对配置信息进行修改。

芯片配置字默认值仅表示为烧录器界面的缺省设置，配置字地址单元在出厂时可能为非空，在 FLASH 编程之前必须先擦除，才能正确写入所设置的配置字。

26.3.2.1 芯片配置字CFG_WORD

芯片的部分特性需要通过芯片配置字配置实现，这些配置在程序运行前生效。

寄存器名称	芯片配置字 (CFG_WORD)	
地址偏移	0000 _H	
低 16Bits 复位值	1001_1000_0000_0000 _B (9800 _H)	
—	Bits 63-32	保留未用
—	Bits 31-16	Bits 15-0 取反值 (不满足取反时 Bit15-0 强制为 9F80 _H)
—	Bit 15	保留 (需固定为 1)
IWDTEN	Bit 14	IWDT 使能位 0: 由软件使能 1: 硬件强制使能 注: 硬件强制使能后, 软件无法关闭; 中断强制使能, 软件无法关闭; 复位强制使能, 软件无法关闭; 时钟源固定为 LRC, 软件无法切换。
WWDTEN	Bit 13	WWDT 使能位 0: 软件使能后可关闭 1: 软件使能后无法关闭
BORVS	Bits 12-10	BOR 电压点选择位 000: 4.6V 001: 4.0V 010: 3.6V 011: 3.1V 100: 2.8V 101: 2.5V 110: 2.1V 111: 保留
—	Bit 9-6	保留 (需固定为 0000)
BOOTADDR	Bit 5-0	芯片复位后 Flash 启动地址 (4KB 对齐)

注 1: 复位值是指芯片烧录器界面的配置字缺省设置, 芯片出厂时的配置字 CFG_WORD 可能与该复位值不同, 在使用前必须通过编程器设置并写入正确的配置字。

注 2: 用户系统必须按照某种方式配置芯片配置字 CFG_WORD, 不能使之为空 (即 0xFFFF)。

26.3.2.2 写保护区域配置字CFG_WRP

芯片支持 2 个保护区域，分别通过 CFG_WRP0 和 CFG_WRP1 来配置。设置为写保护的区域，用户程序将不能通过 IAP 对其进行擦写。

寄存器名称	写保护区域 x 配置字 (CFG_WRPx) (x=0..1)	
地址偏移	0020 _H , 0030 _H	
低 16 位复位值	0000_0000_0000_0001 _B (0001 _H)	
—	Bits 63-32	保留未用
—	Bits 31-16	Bits 15-0 取反值 (不满足取反时 Bit15-0 强制为默认值)
END	Bits 15-9	保护结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 0x7F: Flash Page 511 注: 保护结束页数必须配置为大于或等于起始页数, 否则保护配置失效
—	Bits 8	保留未用 (固定为 0)
START	Bits 7-1	保护起始页配置位 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 0x7F: Flash Page 508
ENB	Bit 0	保护使能位 0: 使能 1: 禁止 (默认)

26.3.2.3 数据区配置字CFG_DAFLS

芯片支持 1 个数据区域，通过 CFG_DAFLS 来配置。通过其配置可以将 FLASH 空间分为程序区和数据区。程序区和数据区的 IAP 擦写命令不同。

寄存器名称	数据 Flash 配置字 (CFG_DAFLS)	
地址偏移	0040 _H	
低 16 位复位值	0000_0000_0000_0001 _B (0001 _H)	
—	Bits 63-32	保留未用
—	Bits 31-16	Bits 15-0 取反值 (不满足取反时 Bit15-0 强制为默认值)
END	Bits 15-9	数据 Flash 结束页配置位 0x0: Flash Page 3 (默认) 0x1: Flash Page 7 0x2: Flash Page 11 0x7F: Flash Page 511

		注：数据 Flash 结束页数必须配置为大于或等于起始页数，否则数据 Flash 配置失效
—	Bits 8	保留未用（固定为 0）
START	Bits 7-1	数据 Flash 起始页配置位 0x0: Flash Page 0（默认） 0x1: Flash Page 4 0x2: Flash Page 8 0x7F: Flash Page 508
ENB	Bit 0	数据 Flash 使能位 0: 使能 1: 禁止（默认）

26.3.2.4 用户程序校验码CHKSUM

烧录工具将用户程序校验码写入此区域。

寄存器名称	用户程序校验码（CHKSUM）	
地址偏移	01C0 _H	
—	Bits 63-32	保留未用
CHKSUM	Bits 31-0	用户程序校验码

寄存器名称	用户程序校验码反码（CHKSUMN）	
地址偏移	01C8 _H	
—	Bits 63-32	保留未用
CHKSUMN	Bits 31-0	用户程序校验码反码

26.3.2.5 全局读保护配置字CFG_GBRDP

全局读保护分为 Level0~2 三个等级，Level0 为不保护，Level2 的详细说明请参考“存储器系统控制（MSC）”中“FLASH 保护”章节的描述。

寄存器名称	全局读保护配置字（CFG_GBRDP）	
地址偏移	0050 _H	
复位值	0101_0101_1010_1010_0101_0101_1010_1010 _B (55AA_55AA _H)	
—	Bits 63-32	保留未用
GBRDP	Bits 31-0	全部读保护配置位 0xFFFF_FFFF: 读保护等级 Level 0 0xYYYY_XXXX: 读保护等级为 Level 2 (Y、X 不为 F) (默认)

26.3.2.6 私有读保护配置字CFG_PCROP

寄存器名称	私有代码读出保护区域 x 配置字（CFG_PCROP _x ）（x=0、1）	
地址偏移	0200 _H , 0210 _H	
低 16 位复位值	1111_1110_0000_0000 _B (FE00 _H)	
—	Bits 63-32	保留未用
—	Bits 31-16	Bits 15-0 取反值（不满足取反时 Bit15-0 强制为默认值）
END	Bits 15-9	保护结束页配置位 0x0: Flash Page 3 0x1: Flash Page 7 0x2: Flash Page 11 0x7F: Flash Page 511 (默认) 注：保护结束页数必须配置为大于或等于起始页数，否则整个 Flash 程序区域都将处于非保护状态
—	Bit 8	保留未用（固定为 0）
START	Bits 7-1	保护起始页配置位 0x0: Flash Page 0 (默认) 0x1: Flash Page 4 0x2: Flash Page 8 0x7F: Flash Page 508
ENB	Bit 0	保护使能位 0: 使能 (默认) 1: 禁止

第27章 修订历史

版本	修订日期	修订内容
V1.0	2023-12-7	初版发布
V1.1	2024-05-21	更新外部振荡器时钟的频率范围，修改为 1~12MHz。
V1.2	2024-07-29	1.完善DMA不同传输模式的通信速率描述； 2.完善 SPI 单工模式的通信速率描述。
V1.3	2025-03-19	完善存储器到外设的DMA传输模式描述。