

HC32F155 系列

32 位 ARM® Cortex®-M0+微控制器

参考手册

Rev1.01 2026 年 04 月

声 明

- ★ 小华半导体有限公司（以下简称：XHSC）保留随时更改、更正、增强、修改小华半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- ★ 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- ★ XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- ★ XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- ★ 任何带有®或™标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- ★ 本通知中的信息取代并替换先前版本中的信息。

©2026 小华半导体有限公司 保留所有权利

目 录

声 明.....	2
目 录.....	3
表目录.....	17
图目录.....	20
文档约定.....	24
寄存器相关缩略语.....	24
词汇表.....	24
关于本手册.....	24
简介 (Overview)	25
1 存储器映射 (Memory Mapping)	26
1.1 存储器映射.....	26
1.2 位段空间	29
2 总线架构 (BUS)	30
2.1 概述.....	30
2.2 总线架构	31
2.3 总线功能	31
3 复位控制 (RMU)	32
3.1 概述.....	32
3.2 复位方式和复位标志位	32
3.3 复位时序	34
3.3.1 上电复位	34
3.3.2 NRST 引脚复位	34
3.3.3 低电压检测复位.....	35
3.3.4 专用看门狗复位.....	35
3.3.5 软件复位	36
3.3.6 RAM 奇偶校验复位.....	36
3.3.7 M0+ Lockup 复位	36
3.3.8 外部高速振荡器异常停振复位.....	37
3.3.9 复位方式的判断.....	37
3.4 寄存器描述.....	37
3.4.1 复位标志寄存器 0 (RMU_RSTF0)	38
4 时钟控制器 (CMU)	40
4.1 概述.....	40

4.2	系统框图	41
4.3	时钟源规格.....	42
4.4	工作时钟规格	42
4.5	晶振电路	43
4.5.1	外部高速振荡器	43
4.5.2	外部高速振荡器故障检测	44
4.5.3	外部低速振荡器	46
4.6	内部 RC 时钟	47
4.6.1	HRC 时钟	47
4.6.2	LRC 时钟	47
4.7	时钟切换步骤	48
4.7.1	时钟源切换	48
4.8	时钟输出功能	49
4.9	寄存器描述	50
4.9.1	CMU XTAL 配置寄存器(CMU_XTALCFGR).....	51
4.9.2	CMU XTAL 稳定配置寄存器(CMU_XTALSTBCR).....	51
4.9.3	CMU XTAL 控制寄存器(CMU_XTALCR).....	52
4.9.4	CMU XTAL 振荡故障控制寄存器(CMU_XTALSTDCR)	52
4.9.5	CMU XTAL 振荡故障状态寄存器(CMU_XTALSTDSR).....	53
4.9.6	CMU XTAL32 配置寄存器(CMU_XTAL32CFGR).....	53
4.9.7	CMU XTAL32 滤波寄存器(CMU_XTAL32NFR).....	54
4.9.8	CMU XTAL32 控制寄存器(CMU_XTAL32CR).....	54
4.9.9	CMU HRC 控制寄存器(CMU_HRCCR)	55
4.9.10	CMU HRC 频率配置寄存器(CMU_HRCCFGR).....	55
4.9.11	CMU LRC 控制寄存器(CMU_LRCCR)	56
4.9.12	CMU 时钟源稳定状态寄存器(CMU_OSCSTBSR)	56
4.9.13	CMU 系统时钟源切换寄存器(CMU_CKSWR).....	57
4.9.14	CMU 时钟分频配置寄存器(CMU_SCKDIVR).....	57
4.9.15	CMU ADC 时钟配置寄存器(CMU_PERICKSEL).....	58
4.9.16	CMU 功能时钟控制寄存器(CMU_FCG)	59
4.9.17	CMU MCO1 时钟输出配置寄存器(CMU_MCO1CFGR)	61
5	电源控制 (PWC)	62
5.1	概述	62
5.2	电源分布	62
5.3	低电压检测单元(LVD)说明	63

5.3.1	上电复位/掉电复位动作说明	63
5.3.2	低电压检测(LVD)	63
5.3.3	LVD 中断/复位框图.....	64
5.3.4	输入/输出引脚	64
5.3.5	LVD 中断和复位	65
5.3.6	内部电压采样和检测功能.....	66
5.4	动作模式与低功耗模式	67
5.4.1	运行模式	68
5.4.2	睡眠模式	69
5.4.3	停止模式	69
5.5	降低功耗的方法.....	70
5.5.1	降低系统时钟速度	70
5.5.2	关闭不使用的时钟源.....	70
5.5.3	功能时钟停止	71
5.6	寄存器保护功能.....	71
5.7	寄存器描述.....	72
5.7.1	停止模式控制寄存器(PWC_STPMCR).....	72
5.7.2	电源模式控制寄存器 (PWC_PWRC)	72
5.7.3	电源监控寄存器(PWC_PWRMON).....	73
5.7.4	RAM 控制寄存器(PWC_RAMCR).....	73
5.7.5	功能保护控制寄存器(PWC_FPRC).....	74
5.7.6	LVD ICG 控制寄存器(PWC_LVDICGCR)	75
5.7.7	LVD 控制状态寄存器(PWC_LVDCSR).....	76
5.7.8	调试模式控制寄存器(PWC_DBGCR)	76
6	初始化配置 (ICG)	77
6.1	概述.....	77
6.2	寄存器描述.....	77
6.2.1	初始化配置寄存器 0(ICG0)	78
6.2.2	初始化配置寄存器 1(ICG1)	80
6.2.3	初始化配置寄存器 n(ICGn)n=2~6	81
7	嵌入式 FLASH (EFM)	82
7.1	概述	82
7.2	主要特性	82
7.3	嵌入式 FLASH	82
7.4	读接口.....	83

7.4.1	CPU 时钟和 FLASH 读取时间之间的关系	83
7.5	FLASH 读预取	84
7.6	FLASH 编程和擦除操作	84
7.6.1	单次编程无回读模式	84
7.6.2	单编程回读模式	85
7.6.3	连续编程操作	85
7.6.4	擦除操作	86
7.6.5	总线保持功能	86
7.6.6	FLASH 擦除、编程窗口保护	86
7.6.7	扇区擦除取消功能	87
7.6.8	中断	88
7.7	寄存器描述	89
7.7.1	访问保护寄存器(EFM_FAPRT)	90
7.7.2	FLASH 停止寄存器(EFM_FSTP)	90
7.7.3	读模式寄存器(EFM_FRMC)	91
7.7.4	擦写模式寄存器(EFM_FWMC)	92
7.7.5	状态寄存器(EFM_FSR)	93
7.7.6	状态清除寄存器(EFM_FSCLR)	94
7.7.7	中断许可寄存器(EFM_FITE)	95
7.7.8	FLASH 窗口保护起始地址寄存器(EFM_FPMTSW)	95
7.7.9	FLASH 窗口保护结束地址寄存器(EFM_FPMTEW)	95
7.7.10	UNIQUE ID 寄存器(EFM_UQID0)	96
7.7.11	UNIQUE ID 寄存器(EFM_UQID1)	96
7.7.12	UNIQUE ID 寄存器(EFM_UQID2)	96
7.8	注意事项	97
8	内置 SRAM (SRAM)	98
8.1	概述	98
9	通用 IO (GPIO)	99
9.1	概述	99
9.2	端口功能概要	100
9.3	动作说明	101
9.3.1	通用输入输出 GPIO 功能	101
9.3.2	外围功能	101
9.3.3	外部中断 EIRQ 输入功能	102
9.3.4	模拟功能	102

9.3.5	通用控制	102
9.4	寄存器描述.....	103
9.4.1	通用输入寄存器 (PIDRx)	104
9.4.2	通用输出数据寄存器 (PODRx)	104
9.4.3	通用输出许可寄存器 (POERx)	104
9.4.4	通用输出置位寄存器 (POSRx)	105
9.4.5	通用输出复位寄存器 (PORRx)	105
9.4.6	通用输出翻转寄存器 (POTRx)	105
9.4.7	通用控制寄存器 (PCRxy)	106
9.4.8	特殊控制寄存器 (PSPCR)	108
9.4.9	公共控制寄存器 (PCCR)	108
9.4.10	输入控制寄存器 (PINAER)	109
9.4.11	写保护寄存器 (PWPR)	109
9.4.12	访问位宽	109
9.5	注意事项	109
10	中断控制器 (INTC)	110
10.1	概述	110
10.2	INTC 系统框图	111
10.3	向量表.....	112
10.3.1	中断向量表	112
10.3.2	中断事件请求序号	113
10.4	功能说明	118
10.4.1	不可屏蔽中断	118
10.4.2	EIRQ 管脚中断	118
10.4.3	EKEY 管脚事件.....	118
10.4.4	软件触发事件	119
10.4.5	中断源选择	119
10.4.6	WFE 事件选择.....	120
10.4.7	WFE 唤醒事件管理.....	120
10.4.8	数字滤波器	121
10.4.9	低功耗模式返回	122
10.5	寄存器描述.....	124
10.5.1	功能保护控制寄存器 (INTC_FPRCR)	126
10.5.2	不可屏蔽中断使能寄存器 (INTC_NMIER)	126
10.5.3	不可屏蔽中断标志寄存器 (INTC_NMIFR)	127

10.5.4	不可屏蔽中断标志清除寄存器 (INTC_NMICLR)	127
10.5.5	EIRQ 管脚中断控制寄存器 (INTC_EIRQCRm) (m=0~11)	128
10.5.6	EIRQ 管脚中断标志寄存器 (INTC_EIRQFR)	128
10.5.7	EIRQ 管脚中断标志清除寄存器 (INTC_EIRQCLR)	129
10.5.8	EKEY 管脚事件控制寄存器 (INTC_EKEYCR)	129
10.5.9	软件触发事件控制寄存器 (INTC_STRGCR)	130
10.5.10	中断选择 A 寄存器 (INTC_ISELARm) (m=8~23)	130
10.5.11	中断选择 B 寄存器 (INTC_ISELBRm) (m=24~31)	131
10.5.12	WFE 事件使能寄存器 (INTC_EVTER)	132
10.5.13	唤醒事件使能寄存器 (INTC_WUPEN)	133
10.6	使用注意事项	133
11	DMA 控制器 (DMA)	134
11.1	概述	134
11.2	模块示意图	135
11.3	功能描述	136
11.3.1	使能 DMA 控制器	136
11.3.2	通道选择和通道优先级	136
11.3.3	启动 DMA	136
11.3.4	数据块	136
11.3.5	传输地址控制	136
11.3.6	传输次数	137
11.3.7	中断和事件信号输出	137
11.3.8	连锁传输	137
11.3.9	不连续地址传输	140
11.3.10	传输提前终止	140
11.4	应用举例	141
11.4.1	存储器到存储器的传输	141
11.4.2	存储器到外围电路的传输	142
11.4.3	存储器到存储器的连锁传输	143
11.5	寄存器描述	145
11.5.1	DMA 使能寄存器(DMA_EN)	146
11.5.2	中断状态寄存器 0(DMA_INTSTAT0)	146
11.5.3	中断状态寄存器 0(DMA_INTSTAT1)	147
11.5.4	中断屏蔽寄存器(DMA_INTMASK0)	147
11.5.5	中断屏蔽寄存器(DMA_INTMASK1)	148

11.5.6	中断复位寄存器(DMA_INTCLR0)	148
11.5.7	中断复位寄存器(DMA_INTCLR1)	149
11.5.8	通道使能置位寄存器(DMA_CHEN).....	149
11.5.9	通道状态观测寄存器(DMA_CHSTAT).....	150
11.5.10	链描述符基地址寄存器 1 (DMA_BASE1_LLP).....	150
11.5.11	链描述符基地址寄存器 2 (DMA_BASE2_LLP).....	151
11.5.12	通道使能复位寄存器(DMA_CHENCLR).....	151
11.5.13	传输启动触发源选择寄存器(DMA_TRGSELx) (x=0~1)	152
11.5.14	传输源地址寄存器(DMA_SARx) (x=0~1)	152
11.5.15	传输目标地址寄存器(DMA_DARx) (x=0~1)	153
11.5.16	数据控制寄存器(DMA_CHxCTL0) (x=0~1).....	153
11.5.17	通道控制寄存器(DMA_CHxCTL1) (x=0~1).....	154
12	内部时钟校准器 (CTC)	155
12.1	概述	155
12.2	结构框图	155
12.3	功能说明	156
12.3.1	参考时钟	156
12.3.2	频率校准	159
12.3.3	错误中断	160
12.3.4	编程指南	160
12.4	寄存器描述.....	161
12.4.1	时钟校准控制寄存器 1 (CTC_CR1)	161
12.4.2	时钟校准控制寄存器 2 (CTC_CR2)	163
12.4.3	时钟校准状态寄存器 (CTC_STR)	164
12.4.4	时钟校准计数器 (CTC_CNT)	164
13	模数转换模块 (ADC)	165
13.1	概述	165
13.2	ADC 系统框图	166
13.3	功能说明	167
13.3.1	ADC 时钟.....	167
13.3.2	通道选择	167
13.3.3	触发源选择.....	168
13.3.4	序列 A 单次扫描模式	168
13.3.5	序列 A 连续扫描模式	169
13.3.6	双序列扫描模式.....	170

13.3.7	模拟看门狗功能	172
13.3.8	模拟输入的采样时间和转换时间	173
13.3.9	A/D 数据寄存器自动清除功能	174
13.3.10	中断和事件信号输出	174
13.4	寄存器描述	176
13.4.1	A/D 启动寄存器 ADC_STR	177
13.4.2	A/D 控制寄存器 0 ADC_CR0	177
13.4.3	A/D 控制寄存器 1 ADC_CR1	178
13.4.4	A/D 转换开始触发寄存器 ADC_TRGSR	179
13.4.5	A/D 转换开始片内触发源选择寄存器 ADC_TRGSEL0, ADC_TRGSEL1	180
13.4.6	A/D 通道选择寄存器 A ADC_CHSELRA	181
13.4.7	A/D 通道选择寄存器 B ADC_CHSELRB	182
13.4.8	A/D 扩展通道选择寄存器 ADC_EXCHSELR	182
13.4.9	A/D 采样状态寄存器 ADC_SSTR	183
13.4.10	A/D 中断状态寄存器 ADC_ISR	183
13.4.11	A/D 中断许可寄存器 ADC_ICR	184
13.4.12	A/D 中断状态复位寄存器 ADC_ISCLRR	184
13.4.13	A/D 数据寄存器 ADC_DR	185
13.4.14	模拟看门狗控制寄存器 ADC_AWDCR	186
13.4.15	模拟看门狗状态寄存器 ADC_AWDSR	187
13.4.16	模拟看门狗状态复位寄存器 ADC_AWDSCLR	187
13.4.17	模拟看门狗阈值寄存器 ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、 ADC_AWD1DR1	188
13.4.18	模拟看门狗比较通道选择寄存器 ADC_AWD0CHSR、ADC_AWD1CHSR	189
13.5	使用注意事项	190
13.5.1	数据寄存器读取时注意事项	190
13.5.2	扫描完成中断处理注意事项	190
13.5.3	模块停止与低功耗设定的注意事项	190
13.5.4	A/D 转换模拟通道输入的引脚设定	190
13.5.5	噪声控制	190
14	通用定时器 (TimerB)	191
14.1	概述	191
14.2	基本框图	191
14.3	功能说明	192
14.3.1	基本动作	192

14.3.2	时钟源选择	194
14.3.3	同步启动	194
14.3.4	数字滤波	195
14.3.5	级联计数	195
14.3.6	脉宽测量	196
14.3.7	周期测量	196
14.3.8	PWM 输出	197
14.4	中断及事件说明	198
14.4.1	比较匹配中断及事件	198
14.4.2	周期匹配中断及事件	198
14.5	寄存器描述	199
14.5.1	通用计数值寄存器 (TMRB_CNTER)	200
14.5.2	周期基准值寄存器 (TMRB_PERAR)	200
14.5.3	比较基准值寄存器 (TMRB_CMPAR)	200
14.5.4	控制状态寄存器 L (TMRB_BCSTRL)	201
14.5.5	控制状态寄存器 H (TMRB_BCSTRH)	202
14.5.6	中断控制寄存器 (TMRB_ICONR)	202
14.5.7	事件控制寄存器 (TMRB_ECONR)	203
14.5.8	状态标志寄存器 (TMRB_STFLR)	203
14.5.9	捕获控制寄存器 (TMRB_CCONR)	204
14.5.10	端口控制寄存器 (TMRB_PCONR)	205
14.5.11	硬件触发事件选择寄存器 (TMRB_HCONR)	206
14.5.12	硬件递加事件选择寄存器 (TMRB_HCUPR)	208
14.5.13	硬件递减事件选择寄存器 (TMRB_HCDOR)	209
14.5.14	内部触发事件选择寄存器 (TMRB_TGRSEL)	209
15	通用定时器 (Timer0)	210
15.1	概述	210
15.2	基本框图	210
15.3	功能说明	211
15.3.1	时钟源选择	211
15.3.2	基本计数动作	211
15.3.3	硬件触发动作	212
15.4	中断及事件说明	212
15.4.1	中断输出	212
15.4.2	事件输出	212

15.5	寄存器描述.....	213
15.5.1	计数值寄存器 (TMRO_CNTAR)	213
15.5.2	基准值寄存器 (TMRO_CMPAR)	213
15.5.3	基本控制寄存器 (TMRO_BCONR)	214
15.5.4	触发选择寄存器 (TMRO_TGRSEL)	215
15.5.5	状态标志寄存器 (TMRO_STFLR)	215
15.6	使用注意事项	216
16	实时时钟 (RTC)	217
16.1	概述	217
16.2	基本框图	217
16.3	功能说明	218
16.3.1	上电设定	218
16.3.2	RTC 计数开始设定	218
16.3.3	系统低功耗模式切换	218
16.3.4	读出计数寄存器	218
16.3.5	写入计数寄存器	219
16.3.6	闹钟设定	219
16.3.7	时钟误差补偿	219
16.3.8	1Hz 输出	219
16.4	中断说明	220
16.4.1	闹钟中断	220
16.4.2	定周期中断	221
16.5	寄存器描述	222
16.5.1	控制寄存器 0 (RTC_CR0)	223
16.5.2	控制寄存器 1 (RTC_CR1)	223
16.5.3	控制寄存器 2 (RTC_CR2)	224
16.5.4	控制寄存器 3 (RTC_CR3)	225
16.5.5	秒计数寄存器 (RTC_SEC)	225
16.5.6	分计数寄存器 (RTC_MIN)	226
16.5.7	时计数寄存器 (RTC_HOUR)	226
16.5.8	日计数寄存器 (RTC_DAY)	228
16.5.9	周计数寄存器 (RTC_WEEK)	228
16.5.10	月计数寄存器 (RTC_MON)	229
16.5.11	年计数寄存器 (RTC_YEAR)	229
16.5.12	分闹钟寄存器 (RTC_ALMMIN)	229

16.5.13	时闹钟寄存器 (RTC_ALMHOUR)	230
16.5.14	时闹钟寄存器 (RTC_ALMWEEK)	230
16.5.15	时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)	231
16.6	注意事项	232
17	看门狗计数器 (SWDT)	233
17.1	概述	233
17.2	功能说明	234
17.2.1	启动看门狗	234
17.2.2	硬件启动方式	234
17.2.3	软件启动方式	235
17.2.4	刷新动作	236
17.2.5	标志位	236
17.2.6	中断复位	236
17.2.7	计数下溢	237
17.2.8	刷新错误	238
17.3	寄存器描述	239
17.3.1	控制寄存器 (SWDT_CR)	240
17.3.2	状态寄存器 (SWDT_SR)	241
17.3.3	刷新寄存器 (SWDT_RR)	241
17.4	使用注意事项	241
18	通用同步异步收发器 (USART)	242
18.1	概述	242
18.2	USART 系统框图	243
18.3	管脚说明	243
18.4	功能说明	244
18.4.1	UART	244
18.4.2	多处理器通信	251
18.4.3	UART_LIN	255
18.4.4	时钟同步模式	259
18.4.5	数字滤波功能	264
18.5	寄存器描述	265
18.5.1	状态寄存器 (USART_SR)	266
18.5.2	发送数据寄存器 (USART_TDR)	268
18.5.3	接收数据寄存器 (USART_RDR)	268
18.5.4	波特率寄存器 (USART_BRR)	269

18.5.5	控制寄存器 1 (USART_CR1)	270
18.5.6	控制寄存器 2 (USART_CR2)	272
18.5.7	控制寄存器 3 (USART_CR3)	273
18.5.8	预分频寄存器 (USART_PR)	273
18.6	使用注意事项	274
18.6.1	UART 注意事项	274
18.6.2	时钟同步模式注意事项	274
18.6.3	其他注意事项	274
19	集成电路总线 (I2C)	275
19.1	概述	275
19.2	I2C 系统框图	276
19.2.1	系统框图	276
19.2.2	结构图	277
19.3	动作说明	278
19.3.1	I2C 协议	278
19.3.2	地址匹配	287
19.3.3	SMBus 动作	292
19.3.4	复位	293
19.3.5	中断和事件信号输出	293
19.3.6	可编程数字滤波	294
19.4	应用软件设定 I2C 初始化流程	295
19.5	寄存器描述	295
19.5.1	I2C 控制寄存器 1(I2C_CR1)	297
19.5.2	I2C 控制寄存器 2(I2C_CR2)	299
19.5.3	I2C 控制寄存器 4(I2C_CR4)	300
19.5.4	I2C 从机地址寄存器 0(I2C_SLR0)	301
19.5.5	I2C 从机地址寄存器 1(I2C_SLR1)	302
19.5.6	I2C 状态寄存器(I2C_SR)	303
19.5.7	I2C 状态清零寄存器(I2C_CLR)	307
19.5.8	I2C 数据发送寄存器(I2C_DTR)	308
19.5.9	I2C 数据接收寄存器(I2C_DRR)	308
19.5.10	I2C 数据移位寄存器(I2C_DSR)	308
19.5.11	I2C 时钟控制寄存器(I2C_CCR)	309
19.5.12	I2C 滤波控制寄存器(I2C_FLTR)	310
20	串行外设接口 (SPI)	311

20.1	概述	311
20.2	SPI 系统框图	312
20.3	管脚说明	312
20.4	SPI 动作系统说明	313
20.4.1	主机模式的管脚状态	313
20.4.2	从机模式的管脚状态	313
20.4.3	SPI 系统连接实例	314
20.5	数据通信说明	315
20.5.1	波特率	315
20.5.2	数据格式	316
20.5.3	传送格式	318
20.5.4	通信方式	320
20.6	运行说明	322
20.6.1	运行模式概要	322
20.6.2	SPI 运行模式时的主机动作	323
20.6.3	SPI 运行模式时的从机动作	324
20.6.4	时钟同步运行模式中的主机动作	325
20.6.5	时钟同步运行模式中的从机动作	326
20.6.6	几种 SPI 动作的处理流程	327
20.7	奇偶校验位自诊断	328
20.8	错误检测	329
20.8.1	欠载错误	329
20.8.2	模式错误	330
20.8.3	过载错误	330
20.8.4	奇偶校验错误	330
20.9	SPI 的初始化	331
20.9.1	清除 SPE 位进行初始化	331
20.9.2	系统复位初始化	332
20.10	中断源	333
20.11	可供使用的事件触发源	333
20.12	寄存器描述	334
20.12.1	SPI 数据寄存器 (SPI_DR)	334
20.12.2	SPI 控制寄存器 (SPI_CR1)	335
20.12.3	SPI 通信配置寄存器 1 (SPI_CFG1)	336
20.12.4	SPI 状态寄存器 (SPI_SR)	337

20.12.5	SPI 通信配置寄存器 2 (SPI_CFG2)	338
21	CRC 运算 (CRC)	339
21.1	概述	339
21.2	功能框图	339
21.3	功能说明	340
21.3.1	CRC16 编码模式	340
21.3.2	CRC16 校验模式	340
21.3.3	CRC32 编码模式	341
21.3.4	CRC32 校验模式	341
21.4	寄存器描述	342
21.4.1	控制寄存器 (CRC_CR)	342
21.4.2	结果寄存器 (CRC_RESULT)	342
21.4.3	数据寄存器 (CRC_DAT)	343
22	调试控制器 (DBGMC)	344
22.1	概述	344
22.2	DBGMC 系统框图	344
22.3	调试端口	345
22.4	引脚排列和调试端口引脚	345
22.4.1	调试端口引脚	345
22.4.2	灵活的引脚分配	345
22.5	引脚上的内部上拉	346
22.6	寄存器描述	346
22.6.1	DBG 状态寄存器 (MCUDBGSTAT)	346
22.6.2	外设调试暂停寄存器 (MCUSTPCTL)	347
	版本修订记录	348

表目录

表 1-1 存储器映射	26
表 3-1 复位方式及产生条件	32
表 3-2 复位方式及复位标志	33
表 3-3 RMU 基地址	37
表 3-4 RMU 寄存器列表	37
表 4-1 时钟源的主要特性	42
表 4-2 各个内部时钟的规格	42
表 4-3 CMU 寄存器列表	50
表 5-1 LVD 特性	63
表 5-2 LVD 电压选择	64
表 5-3 低功耗模式的运行条件及各模块在低功耗模式下的状态	67
表 5-4 运行模式说明	68
表 5-5 寄存器保护列表	71
表 5-6 PWC 寄存器列表	72
表 6-1 ICG 基地址	77
表 6-2 ICG 寄存器	77
表 7-1 CPU 时钟频率和 FLASH 读等待周期对照表	84
表 7-2 EFM 基地址	89
表 7-3 EFM 寄存器列表	89
表 8-1 SRAM 空间分配	98
表 9-1 GPIO 基地址	103
表 9-2 GPIO 寄存器列表	103
表 10-1 输入管脚	110
表 10-2 中断向量表	112
表 10-3 中断事件请求序号与选择	113
表 10-4 INTC 基地址	124
表 10-5 INTC 寄存器列表	124
表 11-1 DMA 寄存器列表	145
表 12-1 HRC 目标频率为 32MHz 时的测量误差	157
表 12-2 HRC 目标频率为 24MHz 时的测量误差	158
表 12-3 CTC 基地址	161
表 12-4 CTC 寄存器列表	161
表 13-1 各 ADC 输入信号说明	166

表 13-2	序列 A 和 B 的各种竞争	170
表 13-3	AD 转换时间	174
表 13-4	ADC 寄存器列表	176
表 14-1	TimerB 的基本功能及特性	191
表 14-2	TimerB 端口列表	192
表 14-3	TimerB 基地址	199
表 14-4	TimerB 寄存器列表	199
表 15-1	Timer0 基地址	213
表 15-2	Timer0 寄存器列表	213
表 16-1	RTC 的基本规格	217
表 16-2	RTC 基地址	222
表 16-3	RTC 寄存器列表	222
表 17-1	看门狗计数器的基本特性	233
表 17-2	SWDT 基地址	239
表 17-3	SWDT 寄存器列表	239
表 18-1	USART 管脚说明	243
表 18-2	UART 接收器的容差	249
表 18-3	UART 中断/事件表	251
表 18-4	多处理器模式中断/事件表	255
表 18-5	时钟同步模式中断/事件表	264
表 18-6	USART 基地址	265
表 18-7	USART 寄存器列表	265
表 18-8	波特率计算公式	269
表 19-1	输入/ 输出引脚	277
表 19-2	中断一览表	293
表 19-3	事件信号输出一览表	294
表 19-4	I2C 基地址	295
表 19-5	I2C 寄存器列表	295
表 20-1	SPI 的特性要点	311
表 20-2	管脚说明	312
表 20-3	主机模式时 SPI 管脚状态说明	313
表 20-4	从机模式时 SPI 管脚状态说明	313
表 20-5	部分设定值的位速	315
表 20-6	SPI 模式和寄存器设定关系	322
表 20-7	错误检测对应表	329

表 20-8	SPI 中断源说明	333
表 20-9	SPI 基地址	334
表 20-10	SPI 寄存器列表	334
表 21-1	CRC 基地址	342
表 21-2	CRC 寄存器列表	342
表 22-1	DBGC 基地址	346
表 22-2	DBGC 寄存器列表	346

图目录

图 2-1 总线架构.....	31
图 3-1 上电复位.....	34
图 3-2 NRST 复位时序	34
图 3-3 LVD 复位.....	35
图 3-4 专用看门狗复位	35
图 3-5 软件复位.....	36
图 3-6 RAM 奇偶校验复位	36
图 3-7 M0+ Lockup 复位	37
图 3-8 外部高速振荡异常复位.....	37
图 4-1 时钟系统框图.....	41
图 4-2 外部高速振荡器连接事例	43
图 4-3 外部时钟输入的连接事例图.....	43
图 4-4 外部高速振荡器故障检测例.....	44
图 4-5 系统时钟选择 XTAL，检测到 XTAL 振荡故障例	45
图 4-6 外部低速振荡器连接事例	46
图 4-7 时钟源切换步骤	48
图 5-1 电源构成图	62
图 5-2 上电复位、掉电复位波形	63
图 5-3 LVD 中断/复位框图.....	64
图 5-4 LVD 中断时序图	65
图 5-5 LVD 复位时序图	65
图 5-6 内部电压采样示意图	66
图 7-1 FLASH 地址分配.....	83
图 9-1 端口基本结构示意图	100
图 10-1 中断系统框图.....	111
图 11-1 DMA 结构图	135
图 11-2 连锁传输示意图	139
图 11-3 不连续地址传输示意图.....	140
图 11-4 应用例 1：存储器到存储器传输	142
图 11-5 应用例 2：存储器到外围电路的传输	143
图 12-1 CTC 的基本框图	155
图 12-2 CTC 校准示意图	159
图 13-1 ADC 框图.....	166

图 13-2	内部模拟通道选择.....	167
图 13-3	序列 A 单次扫描模式.....	168
图 13-4	序列 A 连续扫描.....	169
图 13-5	双序列扫描模式 (RSCHSEL=0 序列 A 从被中断通道重新启动)	171
图 13-6	双序列扫描模式 (RSCHSEL=1 序列 A 从第一个通道重新启动)	171
图 13-7	模拟看门狗保护区 (比较模式)	172
图 13-8	A/D 转换时间	173
图 13-9	ADC 中断和事件输出时序	175
图 14-1	TimerB 基本框图.....	191
图 14-2	锯齿波波形 (递加计数)	192
图 14-3	三角波波形	192
图 14-4	比较输出动作.....	193
图 14-5	捕获输入动作.....	194
图 14-6	软件同步动作.....	195
图 14-7	PWM 输入端口的滤波功能.....	195
图 14-8	32 位级联计数动作	196
图 14-9	脉宽测量.....	196
图 14-10	周期测量.....	197
图 14-11	单边对齐 PWM 输出	197
图 14-12	双边对称 PWM 输出	198
图 15-1	Timer0 基本框图.....	210
图 15-2	Timer0 计数时序图	211
图 16-1	RTC 的基本框图	218
图 17-1	硬件启动例	234
图 17-2	软件启动例	235
图 17-3	各种刷新动作时序示例 (动作确认, 刷新要求信号的下降沿等)	236
图 17-4	计数器下溢动作例.....	237
图 17-5	计数器刷新动作例.....	238
图 18-1	USART 系统框图	243
图 18-2	UART 数据格式	245
图 18-3	UART 发送数据图例 1.....	247
图 18-4	UART 发送数据图例 2.....	247
图 18-5	UART 内部同步和采样时序	248
图 18-6	UART 接收数据图例 1.....	250
图 18-7	UART 接收数据图例 2.....	250

图 18-8 多处理器通信图例.....	252
图 18-9 多处理器模式数据格式.....	252
图 18-10 多处理器模式发送数据图例.....	253
图 18-11 多处理器模式接收数据图例 1.....	254
图 18-12 多处理器模式接收数据图例 2.....	255
图 18-13 LIN 总线数据行为.....	256
图 18-14 时钟同步模式数据格式.....	260
图 18-15 时钟同步模式发送数据图例 1.....	261
图 18-16 时钟同步模式发送数据图例 2.....	262
图 18-17 时钟同步模式接收数据图例 1.....	263
图 18-18 时钟同步模式接收数据图例 2.....	263
图 19-1 I2C 系统框图.....	276
图 19-2 I2C 总线的结构例.....	277
图 19-3 I2C 总线的时序图.....	278
图 19-4 I2C 总线的数据格式.....	279
图 19-5 7 位地址格式的主机发送数据时序图（例）.....	280
图 19-6 7 位地址格式的主机接收数据的时序图（例）.....	281
图 19-7 7 位地址格式的从机发送模式时序图（例）.....	282
图 19-8 7 位地址格式从机接收模式时序图（例）.....	283
图 19-9 SCL 同步时序.....	284
图 19-10 从机发送时序图（1）.....	285
图 19-11 从机发送时序图（2）.....	286
图 19-12 从机接收时序图.....	286
图 19-13 快速 ACK/NACK 时序图.....	287
图 19-14 选择 7 位地址格式时的时序.....	288
图 19-15 选择 10 位地址格式时的时序.....	289
图 19-16 广播地址接收时序图.....	290
图 19-17 SMBus 主机地址接收时序图.....	290
图 19-18 SMBus 报警响应地址接收时序图.....	291
图 19-19 SMBus 默认地址接收时序图.....	292
图 19-20 数字滤波电路框图.....	294
图 20-1 系统框图.....	312
图 20-2 主机模式结构.....	314
图 20-3 三线式时钟同步运行.....	314
图 20-4 数据格式.....	316

图 20-5 数据传送格式图 (CPHA=0)	318
图 20-6 数据传送格式 (CPHA=1)	319
图 20-7 全双工同步串行通信	320
图 20-8 只进行发送通信	321
图 20-9 奇偶校验流程	328
图 20-10 过载错误处理	330
图 20-11 奇偶校验错误	331
图 21-1 CRC 应用示意图	339
图 22-1 调试控制系统	344

文档约定

寄存器相关缩略语

本手册寄存器描述中的缩略语。

缩写	中文释义
RW	读写，软件可以读写该位。
R	只读，软件只能读取该位。
W	只写，软件只能写入该位。读值不确定。
RW0	软件可以读写该位，写 0 有效，写 1 无效。
RW1	软件可以读写该位，写 1 有效，写 0 无效。
RC	只读，读操作后自动位清零。
RWA	可读，解锁/安全模式下可写。
Res	保留位，必须保持复位值。

词汇表

本手册中专业术语与缩写词的定义。

专业术语/缩写词	英文全称	中文释义
Byte	-	字节，8 位数据长度。
Half word	-	半字，16 位数据长度。
Word	-	字，32 位数据长度。
Double word	-	双字，64 位数据长度。
AHB	Advanced High-performance Bus	高级高性能总线。
APB	Advanced Peripheral Bus	高速外设总线。
...	...	

关于本手册

本手册主要介绍芯片的功能、操作事项和使用方法。关于芯片各型号产品的外设可用性与数量信息，以及外设接口、电气特性、管脚封装等规格参数，请参阅对应的“数据手册”。

简介 (Overview)

HC32F155 系列是基于 ARM® Cortex®-M0+ 32-bit RISC CPU, 最高工作频率 32MHz 的高性能 MCU。Cortex-M0+内核支持 2 级流水线, 运算能力达到 0.95 Dhrystone MIPS/MHz, 硬件调试电路支持 TOOL0 单线调试和 2-pin 的 SWD 调试。

HC32F155 系列集成了片上存储器, 包括最大 256KB Flash+1KB 的专用数据 Flash, 最大 32KB 的 SRAM。集成了 Flash 访问预取指加速单元 (Prefetch), 提高 Flash 访问效率。支持外设间事件相互触发, 可以显著降低 CPU 的事务处理负荷。

HC32F155 系列集成了丰富的外设功能。包括最大 14 通道的 12bit 1MSPS ADC, 8 个多功能 16bit Timer (TimerB) 支持输入捕捉、输出比较、PWM 输出, 1 个 16bit Timer (Timer0) 支持异步计数, 1 个 I2C 通信接口, 1 个 SPI 通信接口, 6 个 USART 通信接口, 其中 1 个 USART 支持 LIN 通信功能。

HC32F155 系列支持宽电压范围 (2.7-5.5V), 宽温度范围 (-40-85°C) 和低功耗模式。

典型应用

- 家电控制
- 智能硬件

1 存储器映射 (Memory Mapping)

1.1 存储器映射

本 MCU 支持 4GB 的线性地址空间，地址从 0x0000 0000 到 0xFFFF FFFF，其中包含程序和数据。详细存储器映射请参阅下表。

表 1-1 存储器映射

存储器分类*1		开始地址	结束地址	空间大小	模块*2	保护*3	说明
系统	-	0xE010 0000	0xFFFF FFFF	511MB	Reserved	-	自定义空间
	私有外设外部总线	0xE00F F000	0xE00F FFFF	4KB	ROMTABLE	-	-
		0xE004 2400	0xE00F EFFF	755KB	-	-	-
		0xE004 2000	0xE004 23FF	1KB	-	-	-
		0xE004 1000	0xE004 1FFF	4KB	-	-	-
		0xE004 0000	0xE004 0FFF	4KB	-	-	-
	私有外设内部总线	0xE000 F000	0xE003 FFFF	196KB	-	-	-
		0xE000 E000	0xE000 EFFF	4KB	SCS	-	系统控制空间 NVIC 等
		0xE000 3000	0xE000 DFFF	44KB	-	-	-
		0xE000 2000	0xE000 2FFF	4KB	FPB	-	-
		0xE000 1000	0xE000 1FFF	4KB	DWT	-	-
		0xE000 0000	0xE000 0FFF	4KB	-	-	-
外部设备	-	0xA000 0000	0xDFFF FFFF	1024MB	Reserved	-	-
外部 RAM	-	0x6000 0000	0x9FFF FFFF	1024MB	Reserved	-	-
外设	-	0x4400 0000	0x5FFF FFFF	448MB	Reserved	-	-
		0x4200 0000	0x43FF FFFF	32MB	PeriBitBand	-	CPU 以外主机 Reserved
		0x4010 0000	0x41FF FFFF	31MB	Reserved	-	-
		0x4002 0000	0x400F FFFF	896KB	Reserved	-	-
	AHB	0x4001 5800	0x4001 FFFF	42KB	BLANK	-	-
		0x4001 5400	0x4001 57FF	1KB	CRC	-	-
		0x4001 5000	0x4001 53FF	1KB	DBGC	-	-
		0x4001 4800	0x4001 4FFF	1KB	BLANK	-	-
		0x4001 4400	0x4001 47FF	1KB	CMU	-	-
		0x4001 4000	0x4001 43FF	1KB	EMU	-	-
		0x4001 3800	0x4001 3FFF	2KB	GPIO	-	开始 1K 与 FASTIO 共用
		0x4001 3400	0x4001 37FF	1KB	BLANK	-	-
		0x4001 3000	0x4001 33FF	1KB	BLANK	-	-
		0x4001 2800	0x4001 2FFF	2KB	BLANK	-	-
		0x4001 2400	0x4001 27FF	1KB	BLANK	-	-

存储器分类*1		开始地址	结束地址	空间大小	模块*2	保护*3	说明
外设		0x4001 2000	0x4001 23FF	1KB	BLANK	-	-
		0x4001 1000	0x4001 1FFF	4KB	INTC	-	-
		0x4001 0000	0x4001 0FFF	4KB	BLANK	-	-
	APB	0x4000 D800	0x4000 FFFF	10KB	BLANK	-	-
		0x4000 D400	0x4000 D7FF	1KB	RTC	-	-
		0x4000 D000	0x4000 D3FF	1KB	BLANK	-	-
		0x4000 CC00	0x4000 CFFF	1KB	SWDT	-	-
		0x4000 C800	0x4000 CBFF	1KB	BLANK	-	-
		0x4000 BC00	0x4000 C7FF	3KB	BLANK	-	-
		0x4000 B800	0x4000 BBFF	1KB	ADC	-	-
		0x4000 9800	0x4000 B7FF	8KB	BLANK	-	-
		0x4000 9400	0x4000 97FF	1KB	TMRB_8	-	-
		0x4000 9000	0x4000 93FF	1KB	TMRB_7	-	-
		0x4000 8C00	0x4000 8FFF	1KB	TMRB_6	-	-
		0x4000 8800	0x4000 8BFF	1KB	TMRB_5	-	-
		0x4000 8400	0x4000 87FF	1KB	TMRB_4	-	-
		0x4000 8000	0x4000 83FF	1KB	TMRB_3	-	-
		0x4000 7C00	0x4000 7FFF	1KB	TMRB_2	-	-
		0x4000 7800	0x4000 7BFF	1KB	TMRB_1	-	-
		0x4000 7400	0x4000 77FF	1KB	BLANK	-	-
		0x4000 7000	0x4000 73FF	1KB	BLANK	-	-
		0x4000 6C00	0x4000 6FFF	1KB	BLANK	-	-
		0x4000 6800	0x4000 6BFF	1KB	BLANK	-	-
		0x4000 6400	0x4000 67FF	1KB	BLANK	-	-
		0x4000 6000	0x4000 63FF	1KB	BLANK	-	-
		0x4000 5C00	0x4000 5FFF	1KB	BLANK	-	-
		0x4000 5800	0x4000 5BFF	1KB	TMR0	-	-
		0x4000 4C00	0x4000 57FF	3KB	BLANK	-	-
		0x4000 4800	0x4000 4BFF	1KB	I2C	-	-
		0x4000 3C00	0x4000 47FF	3KB	BLANK	-	-
		0x4000 3800	0x4000 3BFF	1KB	SPI	-	-
		0x4000 3000	0x4000 37FF	2KB	BLANK	-	-
		0x4000 2C00	0x4000 2FFF	1KB	USART_6	-	-
		0x4000 2800	0x4000 2BFF	1KB	USART_5	-	-
		0x4000 2400	0x4000 27FF	1KB	USART_4	-	-
		0x4000 2000	0x4000 23FF	1KB	USART_3	-	-
		0x4000 1C00	0x4000 1FFF	1KB	USART_2	-	-

存储器分类*1		开始地址	结束地址	空间大小	模块*2	保护*3	说明
		0x4000 1800	0x4000 1BFF	1KB	USART_1	-	-
		0x4000 1000	0x4000 17FF	2KB	BLANK	-	-
		0x4000 0C00	0x4000 0FFF	1KB	AOS	-	-
		0x4000 0800	0x4000 0BFF	1KB	EFM	-	-
		0x4000 0400	0x4000 07FF	1KB	BLANK	-	-
		0x4000 0000	0x4000 03FF	1KB	CTC	-	-
SRAM	SRAM	0x2400 0000	0x3FFF FFFF	448MB	Reserved	-	-
		0x2200 0000	0x23FF FFFF	32MB	SRAMBitBand	-	CPU 以外主机 Reserved
		0x2010 0000	0x21FF FFFF	31MB	Reserved	-	-
		0x2000 1000	0x200F FFFF	1020KB	Reserved	-	-
		0x2000 0000	0x2000 7FFF	32KB	SRAM	保护	-
CODE	FLASH	0x0110 0000	0x1FFF FFFF	495MB	Reserved	-	-
		0x0100 0C00	0x010F FFFF	1021KB	Reserved	-	-
		0x0100 0800	0x0100 0BFF	1KB	专用数据 Flash	-	-
		0x0100 0000	0x0100 07FF	2KB	Reserved	-	-
		0x0004 0000	0x00FF FFFF	15.75MB	Reserved	-	-
		0x0000 0000	0x0003 FFFF	256KB	Flash	-	-

注：

1. 请参考 *ARM Cortex-M0+* 说明手册存储器系统。
2. *Reserved*：访问总线会引起总线错误；*BLANK*：写访问无效，读访问时读到 0。
3. 保护功能请参考【电源控制（PWC）】章节。

1.2 位段空间

Cortex™-M0+存储器映射包括两个位段区域。这些区域将存储器别名区域中的每个字映射到存储器位段区域中的相应位。在别名区域写入字时，相当于对位段区域的目标位执行读-修改-写操作。

在本 MCU 中，外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 Cortex™-M0+访问，对于其它总线主接口（如 DMA）无效。

在位段区域，每个字由位段别名地址区域 32 个字的 LSB 表示。实际情况是，当访问位段别名地址时，该地址就会被重映射到位段地址。对于读操作，字被读出且选定位的位置被移到读返回数据的 LSB。对于写操作，待写的位数据被移到所需的位置，然后执行读-修改-写操作。

可以进行位段操作的存储器区域有两个：

0x2000 0000~0x200F FFFF（SRAM，1MB）

0x4000 0000~0x400F FFFF（外设，1MB）

举例说明如下：

1. 将地址 0x2000 0000 设置为 0x3355 AACC。
2. 读地址 0x2200 0008。本次读访问被映射为到 0x2000 0000 的读访问，返回值为 1（0x3355 AACC 的 bit[2]）。
3. 将 0x2200 0008 写为 0。本次写访问被重映射为到地址 0x2000 0000 的读-修改-写。数值 0x3355 AACC 被从寄存器中读出来，清除第 2 位后，结果 0x3355 AAC8 被写入地址 0x2000 0000。
4. 现在读取 0x2000 0000，这样会得到返回值 0x3355 AAC8(bit[2]被清除)。
5. 在访问位段别名地址时，只会用到数据的 LSB(bit[0])。另外，对位段别名区域的访问不应该是非对齐的。若非对齐访问在位段区域内执行，结果是不可预测的。

2 总线架构 (BUS)

2.1 概述

主系统由 32 位 AHB 总线矩阵构成，可实现以下主机总线和从机总线的互连：

■ 主机总线

- Cortex-M0+内核 CPU 总线

Cortex-M0+内核总线，CPU 通过该总线获取指令与数据。访问对象是包含内部存储器与外设。

- 系统 DMA 总线

系统 DMA 专用总线，DMA 通过该总线访问存储器和外设。

■ 从机总线

- FLASH 总线
- SRAM 总线
- 外设总线

借助总线矩阵，可以实现主机总线到从机总线高效率的并发访问，总线矩阵采用固定优先级的总线仲裁方式，访问优先级 DMA 访问>CPU 访问，以保证 DMA 数据传送的连续性。

2.2 总线架构

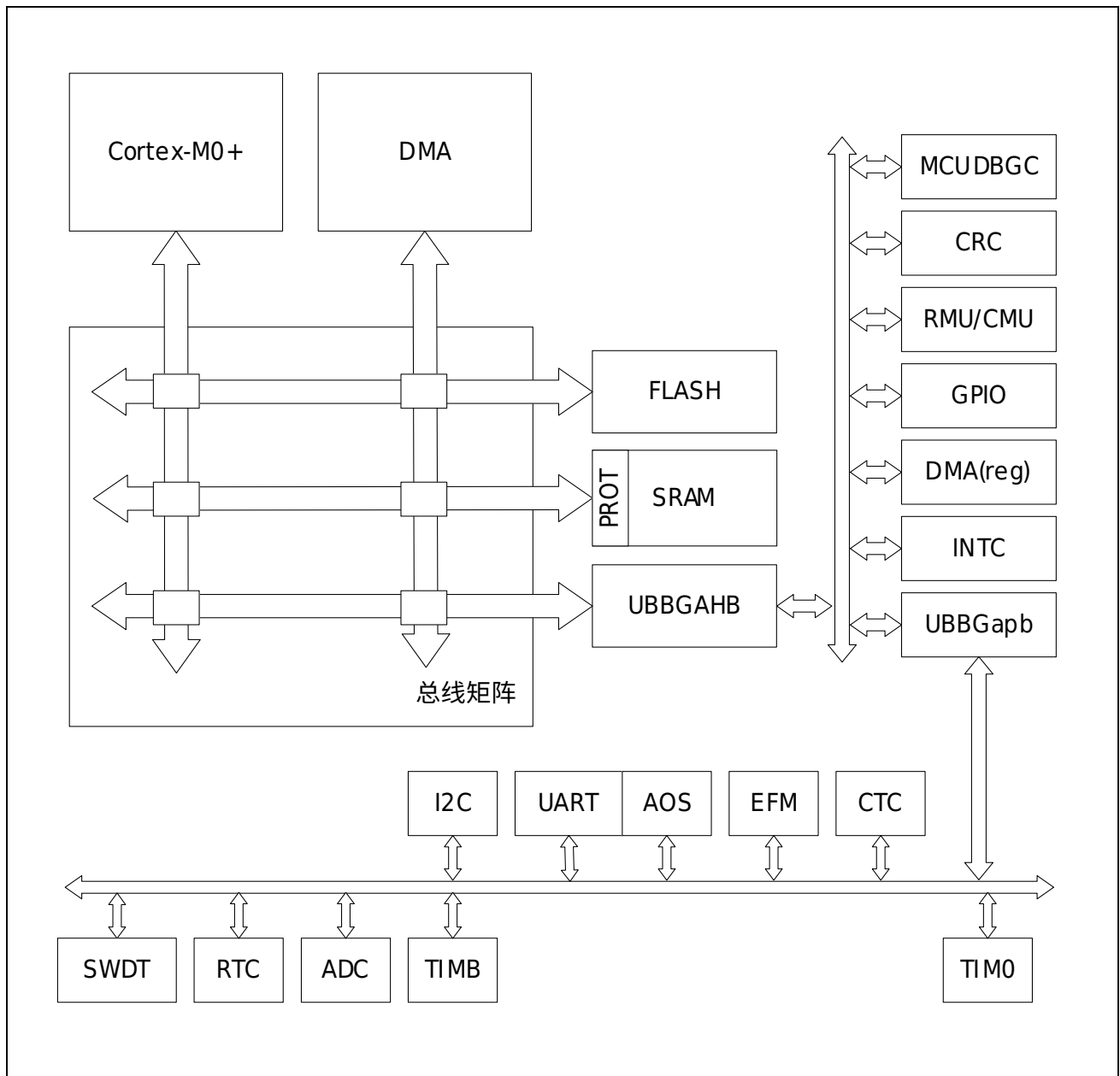


图 2-1 总线架构

注:

- *PROT*: 控制 RAM 访问保护区域, 请参考【RAM 控制寄存器(*PWC_RAMCR*)】位 *RPRTA[1:0]*。

2.3 总线功能

总线负责实现主机对从机的读写访问。通过总线可以实现对 AHB 与 APB 外设的访问。APB 外设通过总线桥与 AHB 连接。

通过总线矩阵, 不同主机总线的访问目标不冲突时, 各访问能够同时进行。例如 CPU 访问 Flash, DMA 访问 SRAM; CPU 访问 APB 外设, DMA 访问 SRAM, 这些访问可以同时进行。

3 复位控制 (RMU)

3.1 概述

芯片配置了 8 种复位方式。

- 上电复位 (POR)
- NRST 引脚复位 (NRST)
- 低电压检测复位 (LVD)
- 专用看门狗复位 (SWDTR)
- 软件复位 (SRST)
- M0+ Lockup 复位
- 外部高速振荡器异常停振复位 (XTALER)
- RAM 奇偶校验复位

3.2 复位方式和复位标志位

复位方式及产生条件如表 3-1 所示。

表 3-1 复位方式及产生条件

复位方式	产生条件
上电复位	V _{DD} 上电
NRST 引脚复位	NRST 引脚输入低电平
低电压检测复位	V _{DD} 电压降至低于 VLVD 电压
专用看门狗复位	看门狗定时器产生刷新错误或者溢出错误
软件复位	设置复位寄存器位 (ARM 寄存器 AIRCR.SYSRESETREQ 位)
RAM 奇偶校验错误复位	RAM 奇偶校验错误产生的复位
M0+ Lockup 复位	当 Cortex-M0+ 遇到严重的异常时, 它会将自己的 PC 指针停在当前地址处, 并锁死自己, 在几个时钟周期延时之后复位整个芯片。
外部高速振荡器异常停振复位	外部高速振荡器异常停振时产生的复位

发生复位时，芯片根据复位方式将对应的复位标志位进行置位，复位标志位如表 3-2 所示。比如发生引脚复位，引脚复位标志位 PINRF 被置 1，在 PINRF 被置位后通过写 CLRf 可以将 PINRF 清零。

表 3-2 复位方式及复位标志

复位标志	复位方式							
	上电复位	引脚复位	低电压检测复位	专用看门狗复位	软件复位	RAM 奇偶校验复位	M0+ Lockup 复位	外部高速振荡器异常停振复位
上电复位标志 (RSTF0.PORF)	✓	—	—	—	—	—	—	—
引脚复位标志 (RSTF0.PINRF)	—	✓	—	—	—	—	—	—
低电压检测复位标志 (RSTF0.LVIRF RSTF0.LVIRF)	—	—	✓	—	—	—	—	—
专用看门狗复位标志 (RSTF0.WDRF)	—	—	—	✓	—	—	—	—
软件复位标志 (RSTF0.SWRF)	—	—	—	—	✓	—	—	—
RAM 奇偶校验错误复位 (RSTF0.RPERF)	—	—	—	—	—	✓	—	—
M0+ Lockup 复位 (RSTF0.CPULKUPRF)	—	—	—	—	—	—	✓	—
外部高速振荡器异常停振复位 (RSTF0.XTALERF)	—	—	—	—	—	—	—	✓

✓：置位 —：不变

3.3 复位时序

3.3.1 上电复位

上电复位是上电复位电路引起的内部复位，时序如图 3-1。在将 NRST 引脚置为高电平的状态下接通电源，就产生上电复位。VDD 电压高于 V_{PORU} 后经过一定的时间 (T_{RSTPOR}) 之后芯片内部复位解除，CPU 开始执行代码。产生上电复位时，上电复位标志 $RMU\_RSTF0.PORF$ 被置位。上电复位的详细说明请参照【上电复位/掉电复位动作说明】。

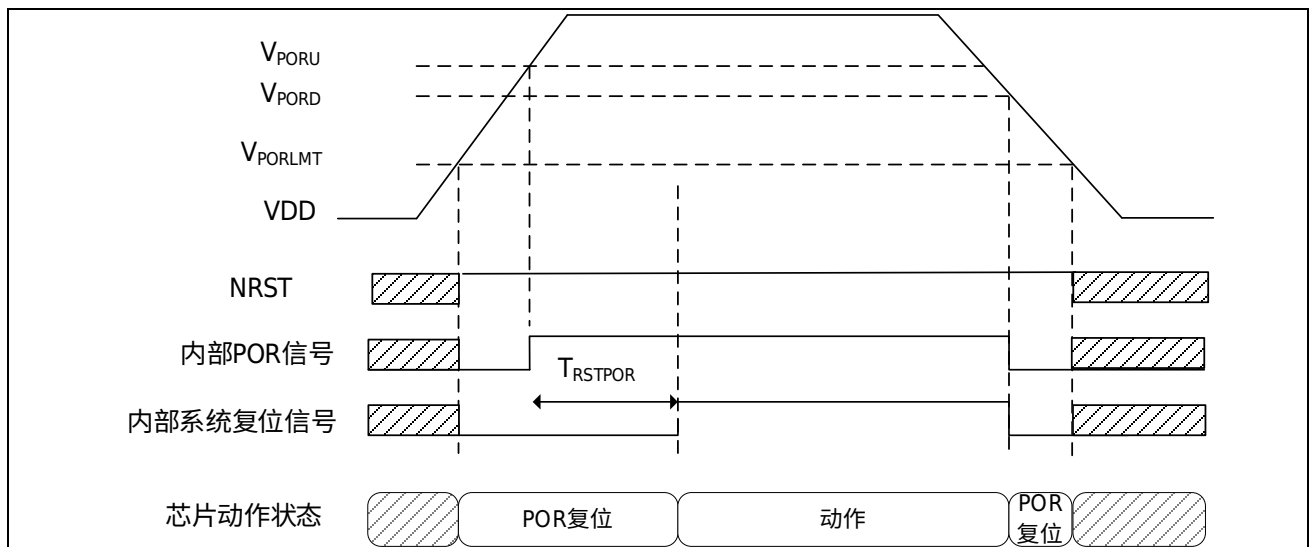


图 3-1 上电复位

3.3.2 NRST 引脚复位

引脚复位是 NRST 引脚被驱动为低电平引起的复位，复位时序如图 3-2。NRST 管脚维持 T_{NRST} 宽度以上的低电平后，经过一定的内部复位时间 (T_{INRST})，解除内部复位。

产生 NRST 引脚复位时，引脚复为标志 $RMU\_RSTF0.PINRF$ 被置位。

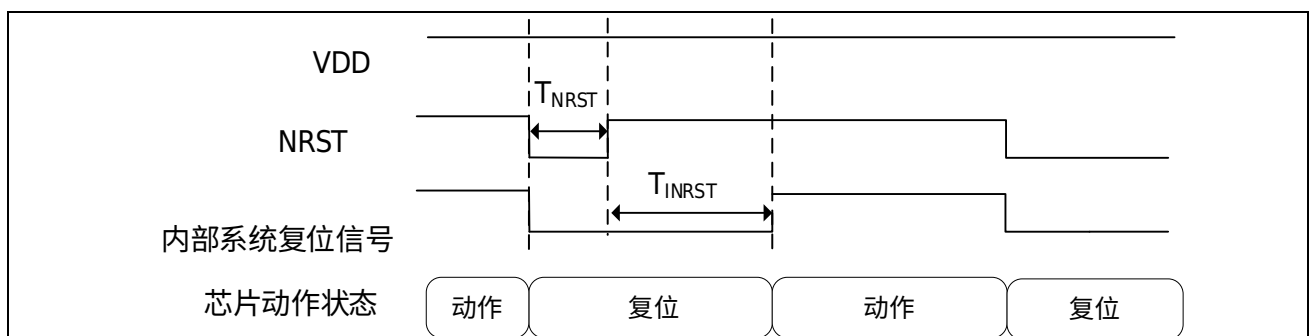


图 3-2 NRST 复位时序

3.3.3 低电压检测复位

低电压检测复位是低电压检测电路引起的内部复位,时序如图 3-3。LVD 设置为复位使能后,如果 V_{DD} 电压低于电压检测电压 V_{LVD} , $RMU\_RSTF0.LVRF$ 被置位。当 V_{DD} 电压高于电压检测电压 V_{LVD} 经过 V_{LVD} 的复位时间 (T_{LVD}) 后解除复位。

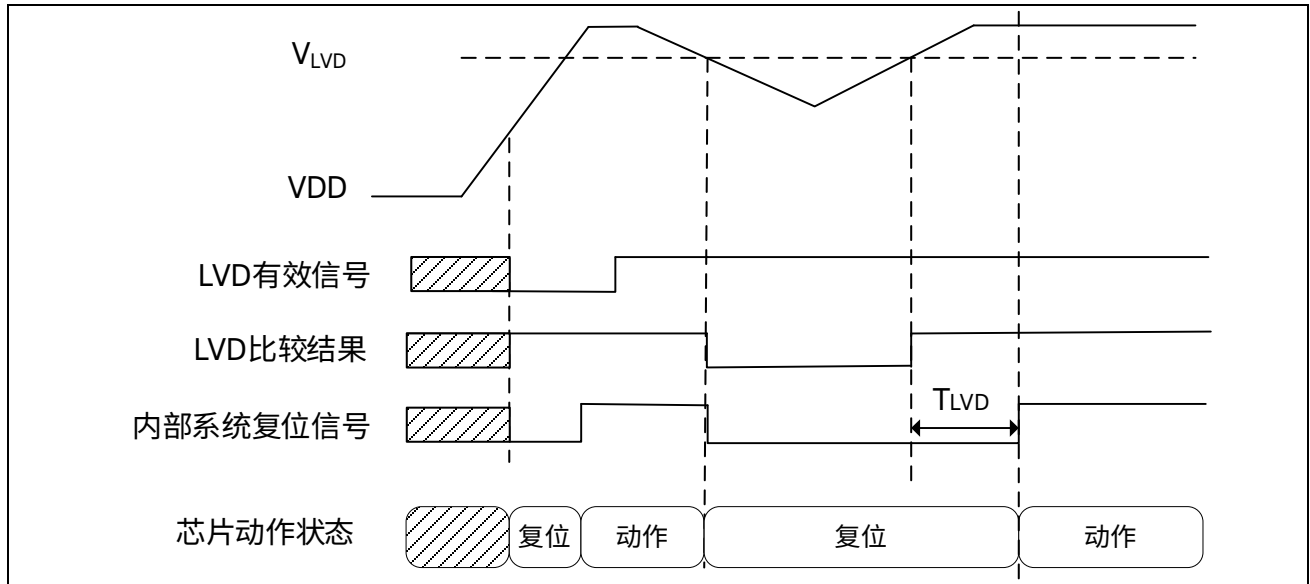


图 3-3 LVD 复位

3.3.4 专用看门狗复位

专用看门狗复位是看门狗定时器引起的内部复位,复位时序如图 3-4。

设定专用看门狗复位有效后,在看门狗定时器产生下溢或者在不允许刷新期间进行写操作时产生专用看门狗复位。专用看门狗复位将 $RMU\_RSTF0.WDRF$ 置位。产生专用看门狗复位后,经过内部复位时间 T_{RIPT} 之后,芯片解除复位。

有关专用看门狗复位的详细内容,请参考【看门狗计数器 (SWDT)】。

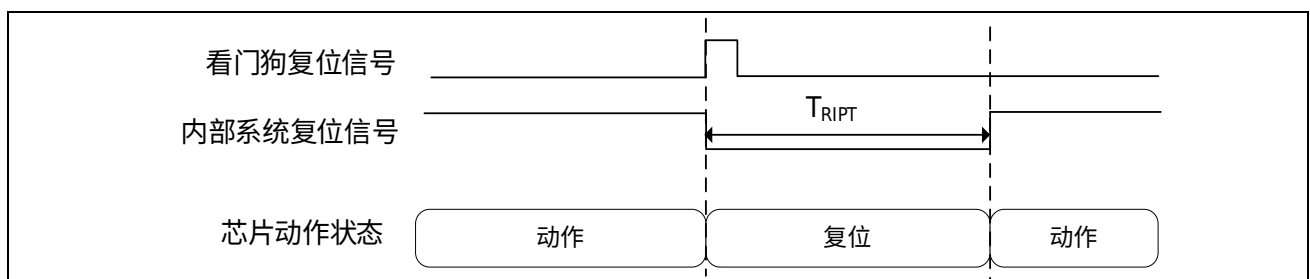


图 3-4 专用看门狗复位

3.3.5 软件复位

通过写 ARM 寄存器 AIRCR 的 SYSRESETREQ 位产生软件复位。产生软件复位时，RMU_RSTF0.SWRF 位被置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

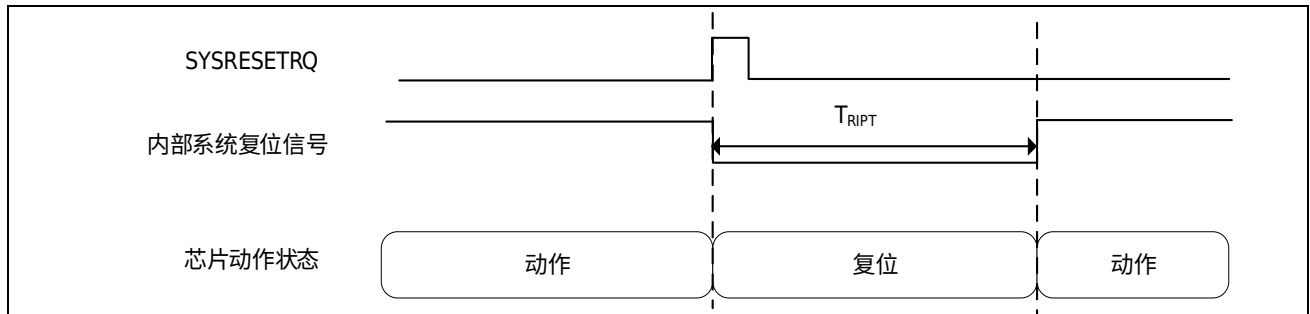


图 3-5 软件复位

3.3.6 RAM 奇偶校验复位

RAM 奇偶校验发生错误时，产生 RAM 奇偶校验复位，时序如图 3-6。RAM 奇偶校验错误将 RMU_RSTF0.RAMPERF 置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

RAM 奇偶校验错误复位的设定，请参照【内置 SRAM (SRAM)】。

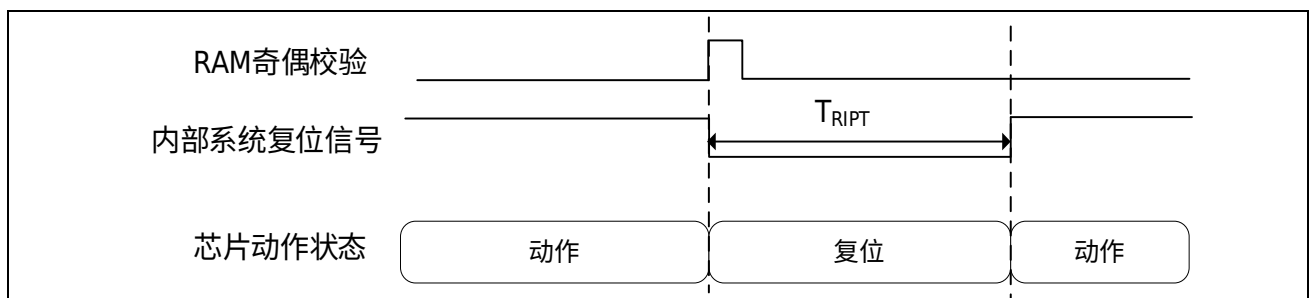


图 3-6 RAM 奇偶校验复位

3.3.7 M0+ Lockup 复位

当 Cortex-M0+遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个芯片，时序如图 3-7。M0+ Lockup 复位将 RSTF0.CPULKUPRF 置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

使用 Lockup 复位，必须将初始化配置寄存器 1 (ICG1) 的 LKUPDIS 位编程为 0。

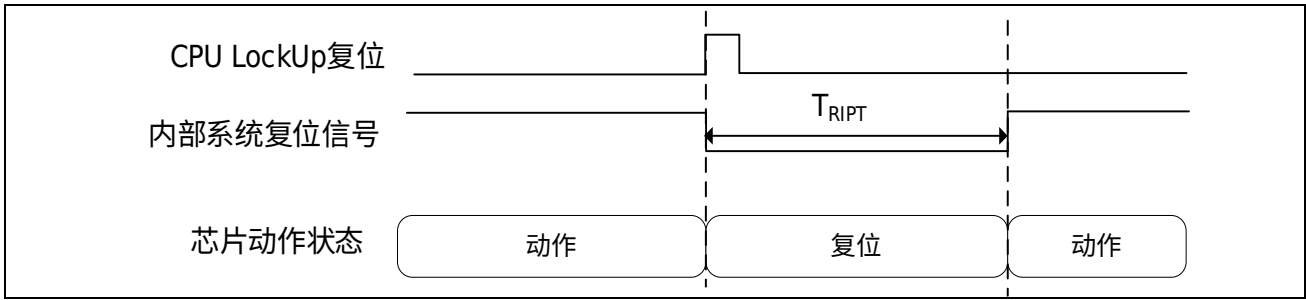


图 3-7 M0+ Lockup 复位

3.3.8 外部高速振荡器异常停振复位

当芯片的震荡停止检测模块有效并且复位使能后，如果发生外部高速振荡器异常停振时，产生外部高速振荡器异常停振复位，RMU_RSTF0.XTALERF 被置位。经过内部复位时间 T_{RIPT} 之后，芯片解除复位。

外部高速振荡器异常停振复位的设定，请参照【外部高速振荡器故障检测】。

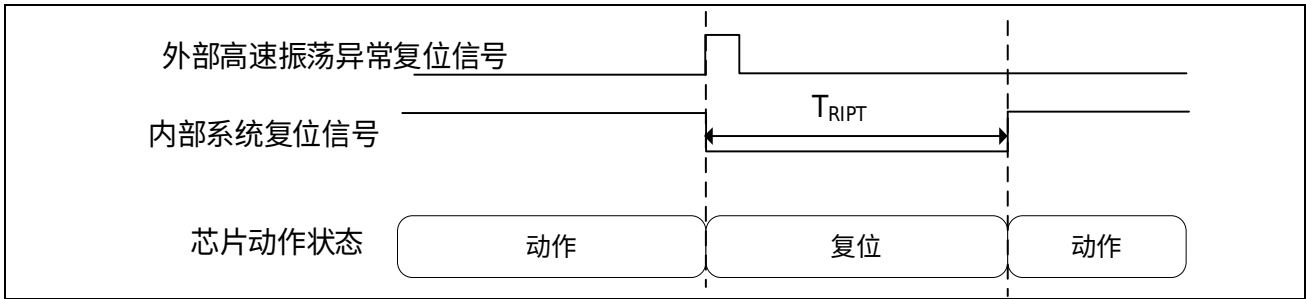


图 3-8 外部高速振荡异常复位

3.3.9 复位方式的判断

根据 RMU_RSTF0 的复位标志可以判断复位方式。当两个或者两个以上复位同时产生时，可能会产生多个复位标志。RMU_RSTF0 中的 MULTIRF 位为 1 时表示发生多个复位。在读 RMU_RSTF0 之后，通过置位 CLRf 位能够将所有的复位标志清 0。

3.4 寄存器描述

表 3-3 RMU 基地址

名称	基地址	描述
RMU	0x4001 4100	RMU 基地址

表 3-4 RMU 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
RMU_RSTF0	复位状态寄存器	0x00	16	根据不同的复位方式复位值不同

3.4.1 复位标志寄存器 0 (RMU_RSTF0)

复位值：0xXXXX (根据复位方式不同，复位值不同)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRf	MULTIRf	XTALERf	CPULKUPRf	Res	RAMPERf	Res	SWRf	Res	Res	WDRf	Res	Res	LVRf	PINRf	PORf

位/位域	标记	位名	功能	读写
b15	CLRf	清除复位标志	软件置 1，用于清除复位标志位。 读出时为 0。置位动作必须在读取 RMU_RSTF0 之后进行。 0：无操作 1：清零复位标志	RW
b14	MULTIRf	2 个以上复位发生标志位	发生两个及两个以上复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生两个及两个以上复位 1：发生两个及两个以上复位时	R
b13	XTALERf	外部高速振荡器异常停振复位标志	发生外部高速振荡器异常停振复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生外部高速振荡器异常停振复位 1：发生外部高速振荡器异常停振复位	R
b12	CPULKUPRf	M0+ Lockup 复位标志	发生 M0+ Lockup 复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生 M0+ Lockup 复位 1：发生 M0+ Lockup 复位	R
b11	Res	保留位	保留，必须保持复位值	R
b10	RAMPERf	RAM 奇偶校验错误复位标志	发生 RAM 奇偶校验错误复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生 RAM 奇偶校验错误复位 1：发生 RAM 奇偶校验错误复位	R
b9	Res	保留位	保留，必须保持复位值	R
b8	SWRf	软件复位标志	发生软件复位复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生软件复位复位 1：发生软件复位复位	R
b7	Res	保留位	保留，必须保持复位值	R
b6	Res	保留位	保留，必须保持复位值	R
b5	WDRf	专用看门狗复位标志	发生专用看门狗复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生专用看门狗复位 1：发生专用看门狗复位	R
b4	Res	保留位	保留，必须保持复位值	R
b3	Res	保留位	保留，必须保持复位值	R
b2	LVRf	低电压检测复位	发生低电压检测复位时，由硬件置位。 通过置位 CLRf 清零。 0：未发生低电压检测复位 1：发生低电压检测复位	R

位/位域	标记	位名	功能	读写
b1	PINRF	NRST 引脚 复位标志	发生引脚复位时，由硬件置位。	R
			通过置位 CLRf 清零。	
			0：未发生 NRST 复位	
			1：发生 NRST 复位	
b0	PORF	上电复位 标志	发生上电复位时，由硬件置位。	R
			通过置位 CLRf 清零。	
			0：未发生上电复位	
			1：发生上电复位	

4 时钟控制器 (CMU)

4.1 概述

时钟控制单元提供了一系列频率的时钟功能，包括：一个外部高速振荡器、一个外部低速振荡器、一个内部高速振荡器、一个内部低速振荡器、时钟预分频器和时钟门控电路。

AHB、APB 和 Cortex-M0+时钟都源自系统时钟，系统时钟的源可选择 4 个时钟源：

- 1) 外部高速振荡器 (XTAL)
- 2) 外部低速振荡器 (XTAL32)
- 3) 内部高速振荡器 (HRC)
- 4) 内部低速振荡器 (LRC)

系统时钟的最大运行时钟频率可以达到 32MHz。

对于每一个时钟源，在未使用时都可以单独打开和关闭，以降低功耗。

4.2 系统框图

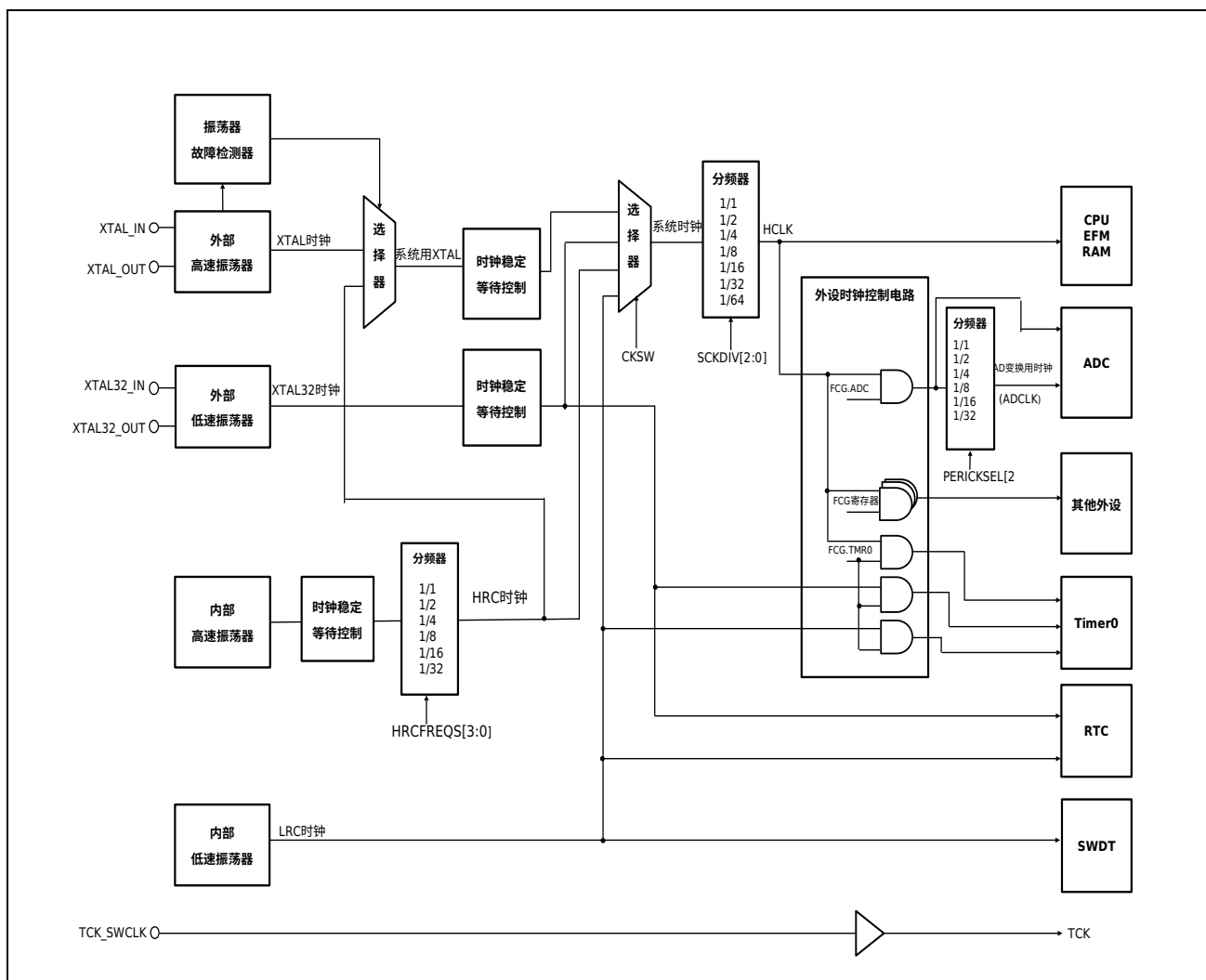


图 4-1 时钟系统框图

4.3 时钟源规格

各时钟源的主要特性如图 4-1 所示。

表 4-1 时钟源的主要特性

时钟源	规格
外部高速振荡器 (XTAL)	晶振的频率范围: 4~20MHz 外部时钟输入: 最高 20MHz 振荡器故障检测功能
外部低速振荡器 (XTAL32)	晶振的频率范围: 32.768KHz
内部高速振荡器 (HRC)	频率: 32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz、1.5MHz。 通过 ICG0.HRCFREQS 位配置。
内部低速振荡器 (LRC)	频率: 32.768KHz 可用作 RTC、SWDT、Timer0 的计数时钟

4.4 工作时钟规格

表 4-2 各个内部时钟的规格

时钟	作用范围	规格
HCLK	CPU、AOS、CRC、EFM(主闪存)、SRAM、GPIO、INTC、USARTn (n=1~6)、SPI、Timer0、TimerBn (n=1~8)、I2C、SWDT (控制逻辑)、ADC (控制逻辑)、CTC	最高频率 32MHz
ADCLK	AD 转换时钟	最高频率 32MHz
STICCLK	CPU 的 SysTickTimer 计数器用时钟, 时钟源为 LRC	频率 32.768KHz
SWDTCLK	SWDT 计数器用时钟	频率 32.768KHz
TCK	SWD 用时钟	最高频率 20MHz

4.5 晶振电路

4.5.1 外部高速振荡器

4.5.1.1 振荡器模式

外部高速振荡器可为系统时钟提供更为精确时钟源。频率范围 4~20MHz。

XTAL 通过 CMU_XTALCR 的 XTALSTP 位打开和关闭。

CMU_OSCSTBSR 的 XTALSTBF 标志位指示外部高速振荡器是否稳定，稳定时间通过寄存器 CMU_XTALSTBCR 配置。CMU_XTALSTBCR 设定的稳定时间一定要大于等于晶振厂商要求的稳定时间。

晶振的电路常数因晶振和安装电路的寄生电容而不同，因此必须和晶振厂商仔细商谈后决定。振荡器的各种特性与用户的电路板设计密接相关，晶振和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。在振荡电路附近不能通过信号线，否则就可能因电感而不能正常振荡。

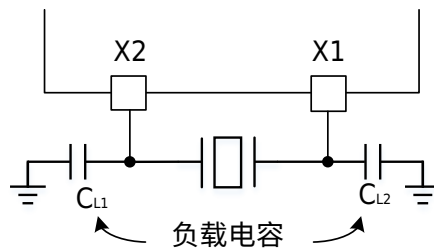


图 4-2 外部高速振荡器连接事例

4.5.1.2 时钟输入模式

时钟输入模式下，必须提供外部时钟源。此模式通过 CMU_XTALCFGR 的 XTALMS 位置 1 和 CMU_XTALCR 的 XTALSTP 位置 0 进行选择。必须使用占空比约为 50% 的外部时钟信号来驱动 X2 引脚。此时 X1 引脚可根据寄存器设定配置成 GPIO。

外部时钟输入的连接例如图 4-3 所示。

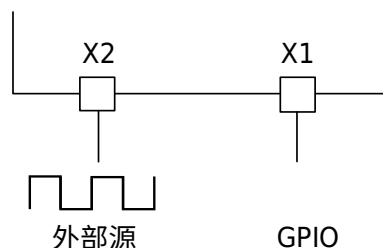


图 4-3 外部时钟输入的连接事例图

4.5.2 外部高速振荡器故障检测

振荡器故障检测是检测外部高速振荡器（XTAL）振荡是否正常。

通过寄存器 CMU_XTALSTDCR 的 XTALSTDE 位打开或关闭。

复位解除后，外部高速振荡器停止振荡，外部高速振荡器故障检测功能无效。要将外部高速振荡器故障检测功能置为有效时，必须使外部高速振荡器振荡，并且在等到外部高速振荡器稳定即 CMU_OSCSTBSR.XTALSTBF 为 1,通过寄存器 CMU_XTALSTDCR 的 XTALSTDE 位打开。

因为振荡器故障检测是检测外部因素导致的振荡器异常振荡，所以要通过软件使外部高速振荡器停止振荡或者转移到停止模式时，将振荡器振荡故障检测功能无效。

如果外部高速振荡器发生故障，动作波形如图 4-4 所示。操作流程参照【检测到 XTAL 故障检测动作】。

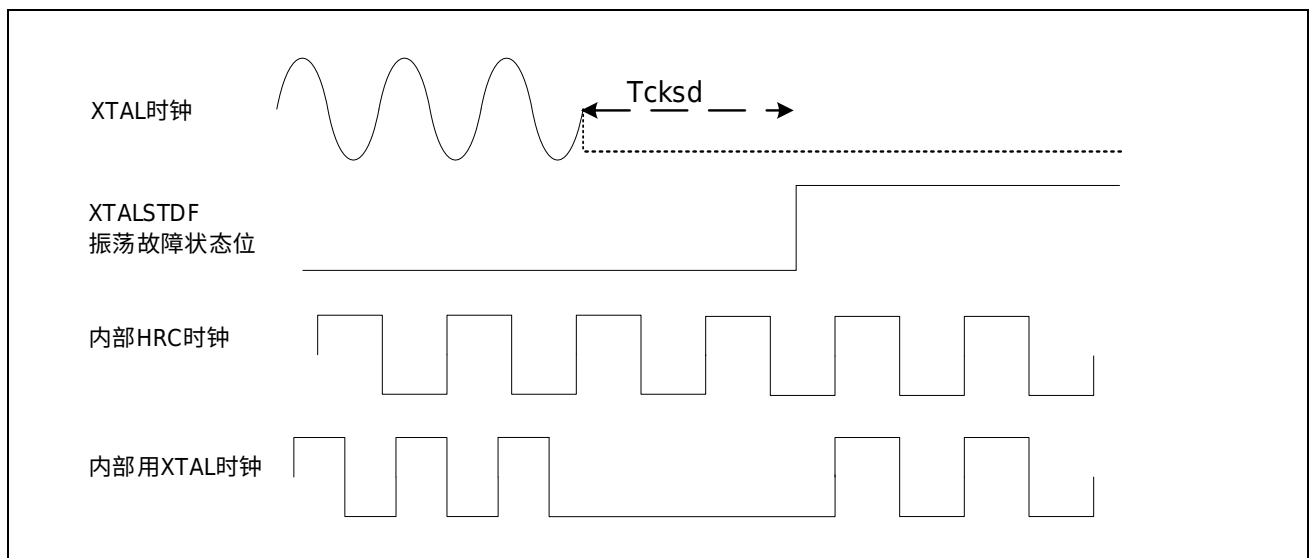


图 4-4 外部高速振荡器故障检测例

4.5.2.1 检测到 XTAL 故障检测动作

当检测到外部高速振荡器振荡故障时，如果系统时钟选择外部高速振荡器作为系统时钟，系统时钟会自动切换到 HRC。

系统时钟选择成 XTAL，检测到 XTAL 故障时，动作例如图 4-5 所示。

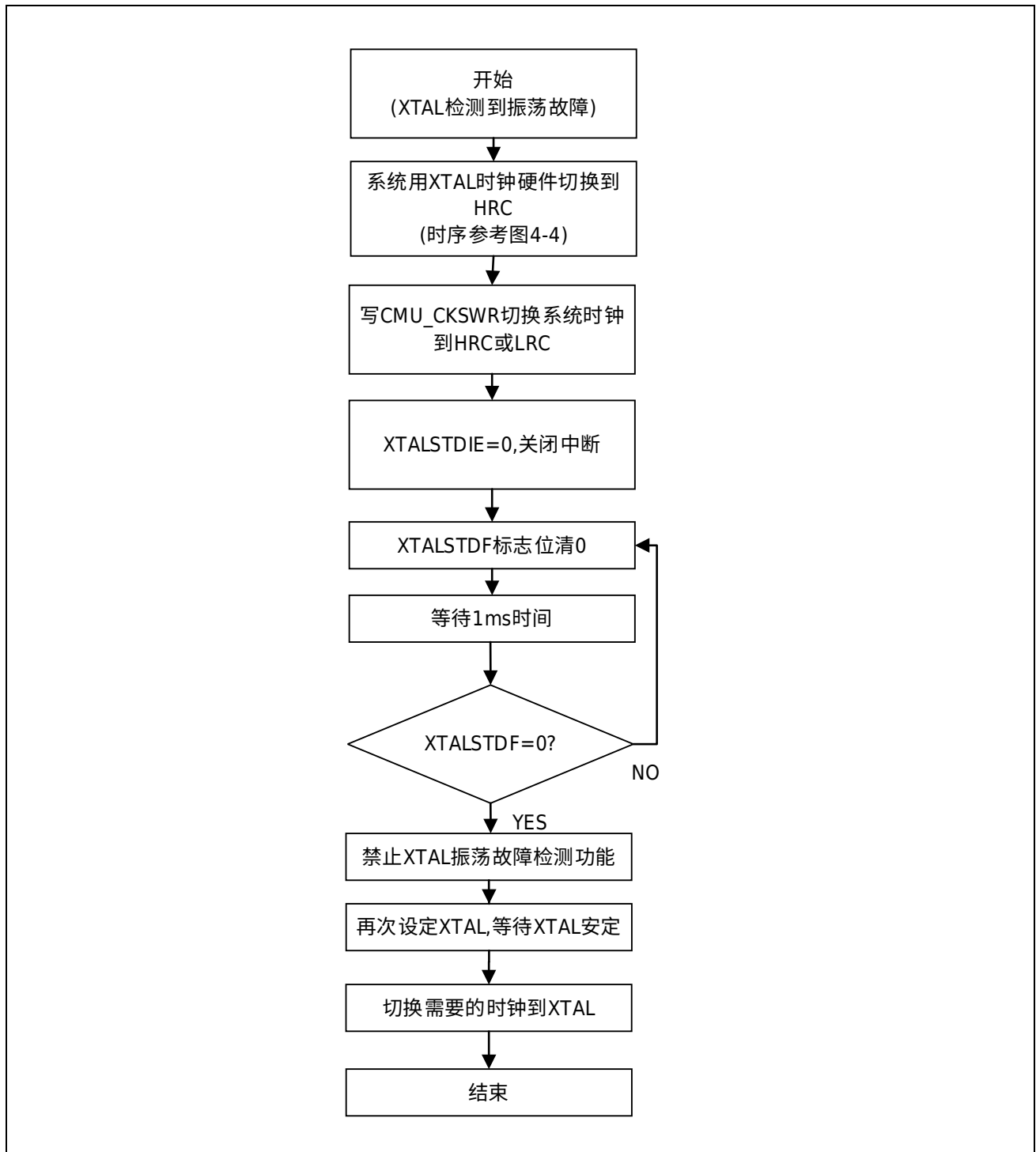


图 4-5 系统时钟选择 XTAL，检测到 XTAL 振荡故障例

4.5.2.2 检测到 XTAL 振荡故障产生中断复位

XTAL 振荡故障中断可配置成可屏蔽中断或非可屏蔽中断，参考【中断控制器（INTC）】章。

XTAL 振荡故障配置成复位时，检测到 XTAL 振荡故障，芯片产生复位，复位动作参考【复位控制（RMU）】章。

4.5.3 外部低速振荡器

32.768KHz 的外部低速振荡器可为系统时钟、实时时钟电路（RTC）提供更为精确时钟源。具有功耗低且精度高的优点。

XTAL32 通过 CMU_XTAL32CR 的 XTAL32STP 位打开和关闭。

晶振的电路常数因晶振和安装电路的寄生电容而不同，因此必须和晶振厂商仔细商谈后决定。振荡器的各种特性与用户的电路板设计密接相关，晶振和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选驱动能力不同做适当调整。在振荡电路附近不能通过信号线，否则就可能因电感而不能正常振荡。

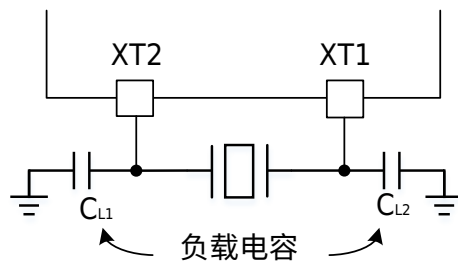


图 4-6 外部低速振荡器连接事例

XTAL32 初次上电的初始化流程如下所示：

1. 通过 CMU_XTAL32CFGR 设定匹配的 XTAL32 驱动能力
2. 通过 CMU_XTAL32NFR 设定滤波功能
3. CMU_XTAL32CR.XTAL32STP 位写 0，XTAL32 振荡
4. 软件等待 XTAL32 稳定，稳定时间参考数据手册中电气特性章节

如果不使用外部低速振荡器，将 CMU_XTAL32CR 的 XTAL32STP 位设 1，关闭外部低速振荡器。

4.6 内部 RC 时钟

4.6.1 HRC 时钟

HRC 时钟信号由内部高速振荡器生成，可直接用作系统时钟。HRC 的频率可由 ICG0. HRCFREQS 配置成 32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz、1.5MHz。也可通过 CMU_HRCCFGR.HRCFREQS[2:0]寄存器软件设定频率。

在复位解除后，CPU 一定以此内部高速振荡时钟开始运行。

HRC 振荡器的优点是成本较低（无需使用外部组件）。此外，其启动速度也要比 XTAL 晶振快，但即使校准后，其精度也不及外部晶振。

频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参照**数据手册中电气特性内部高速（HRC）振荡器**章节。

如果应用受到温度变化影响，则这可能也会影响到 RC 振荡器的精度。

CMU_OSCSTBSR 中的 HRCSTBF 标志指示 HRC 是否稳定。在启动时，硬件将此位置 1 后，HRC 才可以使用。

HRC 可通过 CMU_HRCCR 控制寄存器中的 HRCSTP 位打开或关闭。

4.6.2 LRC 时钟

LRC 时钟信号由内部 32.768KHz 低速振荡器生成，可直接用作系统时钟。LRC 可作为低功耗时钟源在停止模式下保持运行，供 RTC/Timer0/SWDT 使用。

LRC 振荡器的启动速度快，启动后不需要等稳定即可使用。

频率校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，保证精度参照**数据手册中电气特性内部低速（LRC）振荡器**章节。

如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的速度。

LRC 可通过 CMU_LRCCR 控制寄存器中的 LRCSTP 位打开或关闭。

4.7 时钟切换步骤

在系统复位后，默认系统时钟为 HRC。通过设定寄存器 CMU_CKSW 切换时钟源，切换步骤参照时钟源切换。只有在目标时钟源已稳定的状态下，才可以从一个时钟源切换到另一个时钟源。

时钟切换时需要正确配置 Flash 的等待周期，防止系统时钟频率大于 Flash 的最大动作频率。参照【CPU 时钟和 FLASH 读取时间之间的关系】配置。

4.7.1 时钟源切换

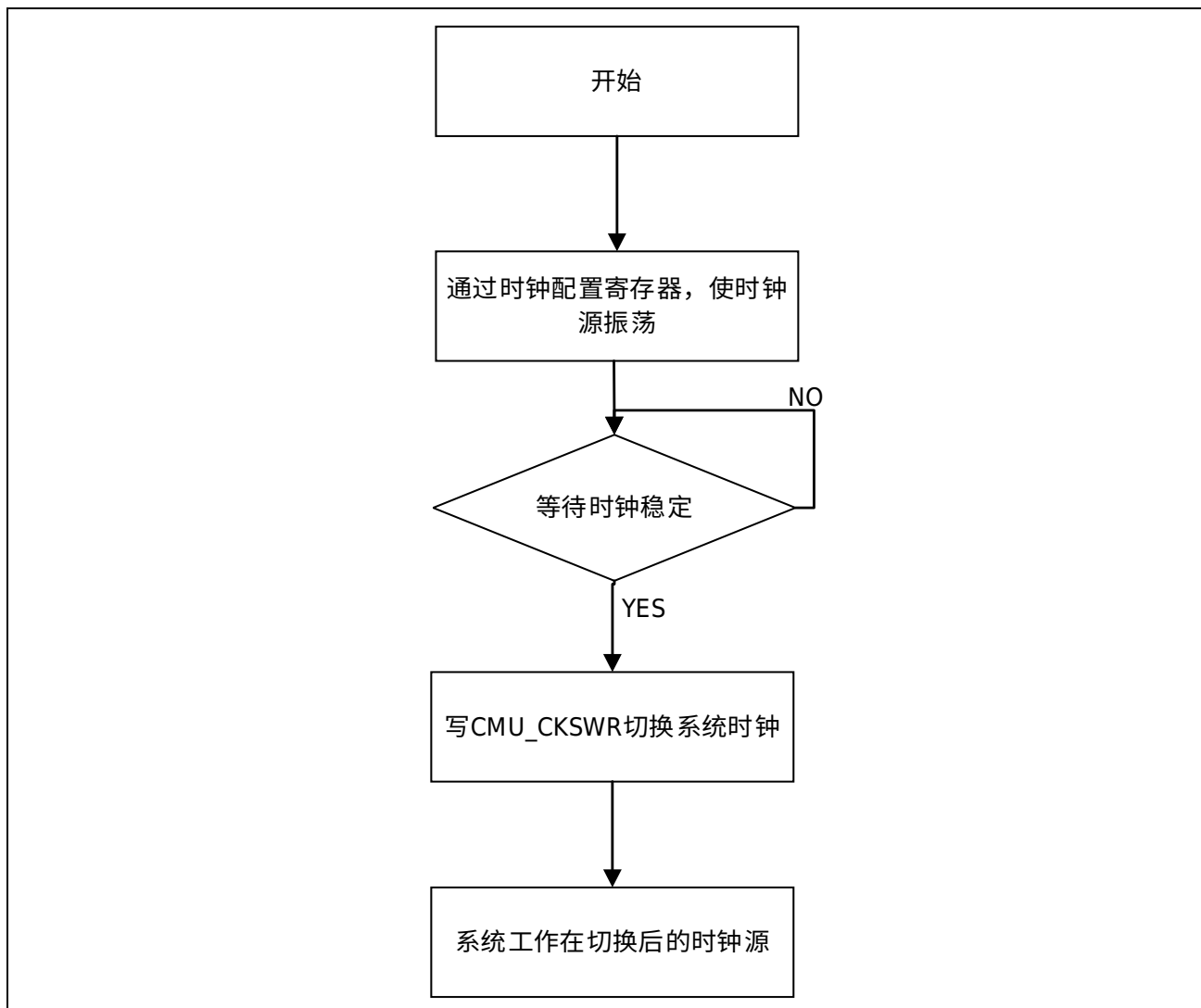


图 4-7 时钟源切换步骤

4.8 时钟输出功能

MCO:

用户可通过可配置的预分配器（1、2、4、8、...、128 分频）向 MCO 引脚输出不同的时钟源：

- HRC 时钟
- LRC 时钟
- XTAL 时钟
- XTAL32 时钟
- 系统时钟

所需的时钟源通过 CMU_MCO1CFGR.MCO1SEL 位选择。

MCO 输出时钟不得超过 16 MHz（最大 I/O 速度）。

4.9 寄存器描述

表 4-3 CMU 寄存器列表

基准地址 1: 0x4001 4400

寄存器	描述	偏移地址	位宽	复位值
CMU_PERICKSEL	CMU ADC 时钟配置寄存器	0x00	8	0x00
CMU_XTALSTDSR	CMU XTAL 振荡故障状态寄存器	0x04	8	0x00
CMU_SCKDIVR	CMU 时钟分频配置寄存器	0x08	8	0x00
CMU_CKSWR	CMU 系统时钟源切换寄存器	0x0C	8	0x00
CMU_XTALCR	CMU XTAL 控制寄存器	0x10	8	0x01
CMU_XTALCFGR	CMU XTAL 配置寄存器	0x14	8	0x80
CMU_XTALSTBCR	CMU XTAL 稳定配置寄存器	0x15	8	0x00
CMU_HRCCR	CMU HRC 控制寄存器	0x18	8	0x00
CMU_OSCSTBSR	CMU 时钟源稳定状态寄存器	0x20	8	0x00
CMU_MCO1CFGR	CMU MCO1 时钟输出配置寄存器	0x24	8	0x00
CMU_XTALSTDCR	CMU XTAL 振荡故障控制寄存器	0x28	8	0x00
CMU_FCG	CMU 功能时钟控制寄存器	0x30	32	0xFFFF FFFF
CMU_XTAL32CR	CMU XTAL32 控制寄存器	0x34	8	0x01
CMU_XTAL32CFGR	CMU XTAL32 配置寄存器	0x38	8	0x00
CMU_XTAL32NFR	CMU XTAL32 滤波寄存器	0x39	8	0x00
CMU_LRCCR	CMU LRC 控制寄存器	0x3C	8	0x00

基准地址 2: 0x4000 0800

寄存器	描述	偏移地址	位宽	复位值
CMU_HRCCFGR	CMU_HRC 频率配置寄存器	0x0282	8	由 ICG0.HRCFREQS[3:0]的值决定

4.9.1 CMU XTAL 配置寄存器(CMU_XTALCFGR)

复位值: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
SUPDRV	XTALMS	XTALDRV[1:0]		Res			

位/位域	标记	位名	功能	读写
b7	SUPDRV	XTAL 超高速驱动允许	0: 禁止超高速驱动 1: 允许超高速驱动 超高速驱动允许时, XTAL 稳定后, 无视此位设定,禁止超高速驱动,降低功耗。	RW
b6	XTALMS	XTAL 模式选择位	0: 振荡器模式 1: 外部时钟输入模式	RW
b5~b4	XTALDRV[1:0]	XTAL 驱动能力选择	00: 高驱动能力(建议 20MHz 晶振) 01: 中驱动能力(建议 10~20MHz 晶振) 10: 小驱动能力(建议 4~10MHz 晶振) 11: 超小驱动能力(建议 4MHz 晶振)	RW
b3~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

4.9.2 CMU XTAL 稳定配置寄存器(CMU_XTALSTBCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res					XTALSTB[2:0]		

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”,写入时写“0”	
b2~b0	XTALSTB[2:0]	XTAL 稳定时间选择	000: $2^8 \times \text{XTAL}$ 周期 001: $2^9 \times \text{XTAL}$ 周期 010: $2^{10} \times \text{XTAL}$ 周期 011: $2^{11} \times \text{XTAL}$ 周期 100: $2^{13} \times \text{XTAL}$ 周期 101: $2^{15} \times \text{XTAL}$ 周期 110: $2^{16} \times \text{XTAL}$ 周期 111: $2^{17} \times \text{XTAL}$ 周期 CMU_XTALCR.XTALSTP 位 1 且 CMU_OSCSTBSR.XTALSTBF 位为 0 的状态下配置此寄存器。 CMU_XTALCFGR.XTALMS 位选择为时钟输入模式时, 无视此等待时间。	RW

4.9.3 CMU XTAL 控制寄存器(CMU_XTALCR)

复位值：0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTALSTP

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	XTALSTP	XTAL 振荡器开启停止位	0: XTAL 振荡器振荡 1: XTAL 振荡器停止	RW

注：

- XTAL 选作系统时钟源时，禁止 XTALSTP 写“1”停止 XTAL 振荡器。

4.9.4 CMU XTAL 振荡故障控制寄存器(CMU_XTALSTDCR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	Res				XTALSTDRIS	XTALSTDRE	XTALSTDIE

位/位域	标记	位名	功能	读写
b7	XTALSTDE	XTAL 振荡故障检测功能允许	0: 禁止 XTAL 振荡故障检测 1: 允许 XTAL 振荡故障检测 注： 振荡器故障检测是检测外部因素导致的振荡器异常振荡，在进入停止模式前，请将振荡器振荡故障检测功能无效。	RW
b6~b3	Res	保留位	读出时为“0”,写入时写“0”	RW
b2	XTALSTDRIS	XTAL 振荡故障复位中断选择	0: XTAL 振荡故障产生中断 1: XTAL 振荡故障产生复位	RW
b1	XTALSTDRE	XTAL 振荡故障复位允许	0: 禁止 XTAL 振荡故障复位 1: 允许 XTAL 振荡故障复位	RW
b0	XTALSTDIE	XTAL 振荡故障中断允许	0: 禁止 XTAL 振荡故障中断 1: 允许 XTAL 振荡故障中断	RW

4.9.5 CMU XTAL 振荡故障状态寄存器(CMU_XTALSTDSR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTALSTDF

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	XTALSTDF	XTAL 振荡故障状态位	0: 未检测到 XTAL 振荡故障 1: 检测到 XTAL 振荡故障 置位条件: XTALSTDE=1 的条件下, XTAL 振荡故障 清零条件: 系统时钟选择 XTAL 以外时钟时, 读 1 写 0。	RW

4.9.6 CMU XTAL32 配置寄存器(CMU_XTAL32CFGR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res				XTAL32IE	Res		XTAL32DRV

位/位域	标记	位名	功能	读写
b7~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
b3	XTAL32IE	XTAL32 时钟输入允许位	0: 禁止通过 P124 输入 XTAL32 时钟 1: 允许通过 P124 输入 XTAL32 时钟	RW
b2~b1	Res	保留位	读出时为 0, 写入时写 0	RW
b0	XTAL32DRV	XTAL32 驱动能力选择	00: 中驱动能力 01: 大驱动能力	RW

4.9.7 CMU XTAL32 滤波寄存器(CMU_XTAL32NFR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						XTAL32NF[1:0]	

位/位域	标记	位名	功能	读写
b7~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1~b0	XTAL32NF[1:0]	XTAL32 振荡器滤波选择	0x: RUN 模式/停止模式, XTAL32 的 3us 滤波有效	RW
			10: RUN 模式 XTAL32 的 3us 滤波有效, 停止模式 XTAL32 的 3us 滤波无效	
			11: RUN 模式/停止模式, XTAL32 的 3us 滤波无效	

4.9.8 CMU XTAL32 控制寄存器(CMU_XTAL32CR)

复位值: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res						XTAL32STP	

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	XTAL32STP	XTAL32 振荡器开启停止位	0: XTAL32 振荡器振荡	RW
			1: XTAL32 振荡器停止	

注:

- XTAL32 选作系统时钟源时, 禁止 XTAL32STP 写“1”停止 XTAL32 振荡器。

4.9.9 CMU HRC 控制寄存器(CMU_HRCCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							HRCSTP

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	HRCSTP	HRC 振荡器开启停止位	0: HRC 振荡器振荡 1: HRC 振荡器停止	RW

注:

- HRC 选作系统时钟源时, 禁止 CMU_HRCCR.HRCSTP 写“1”停止 HRC 时钟。

4.9.10 CMU HRC 频率配置寄存器(CMU_HRCCFGR)

复位值: 由 ICG0.HRCFREQS[3:0]的值决定

b7	b6	b5	b4	b3	b2	b1	b0
Res				HRCFREQS[3:0]			

位/位域	标记	位名	功能	读写
b7~b4	Res	保留位	读出时为“0”,写入时写“0”	RW

内部高速振荡器频率选择, 复位值由 ICG0.HRCFREQS 决定。

HRCFREQS[3]只能通过 ICG0.HRCFREQS[3]设定, 不可以软件改写。

HRCFREQS[2:0]可通过 ICG0.HRCFREQS[2:0]设定。亦可软件改写。该寄存器通过设定 EFM_FAPRT 寄存器解除写保护。

b3~b0	HRCFREQS	HRC 频率选择位	HRCFREQS[2:0]	HRC 频率		RW
				HRCFREQS[3]=0	HRCFREQS[3]=1	
			000	此处设定禁止。	此处设定禁止。	
			001	32MHz	24MHz	
			010	16MHz	12MHz	
			011	8MHz	6MHz	
			100	4MHz	3MHz	
			101~111	2MHz	1.5MHz	

4.9.11 CMU LRC 控制寄存器(CMU_LRCCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							LRCSTP

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	LRCSTP	LRC 振荡器开启停止位	0: LRC 振荡器振荡 1: LRC 振荡器停止	RW

注:

- LRC 选作系统时钟源时, 禁止 LRCSTP 写“1”停止 LRC 时钟。
- RTC 选择 LRC 做时钟源时, LRC 无视此寄存器位, LRC 振荡。RTC 未初始化时, LRC 有振荡的可能。

4.9.12 CMU 时钟源稳定状态寄存器(CMU_OSCSTBSR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			XTAL32STBF	XTALSTBF	Res		HRCSTBF

位/位域	标记	位名	功能	读写
b7~b5	Res	保留位	读出时为“0”,写入时写“0”	R
b4	XTAL32STBF	XTAL32 稳定标志位	0: XTAL32 停止或者未稳定 1: XTAL32 稳定	R
b3	XTALSTBF	XTAL 稳定标志位	0: XTAL 停止或者未稳定 1: XTAL 稳定	R
b2~b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	HRCSTBF	HRC 稳定标志位	0: HRC 停止或者未稳定 1: HRC 稳定	R

4.9.13 CMU 系统时钟源切换寄存器(CMU_CKSWR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						CKSW[1:0]	

位/位域	标记	位名	功能	读写
b7~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	CKSW[1:0]	系统时钟源切换	0 0: 选择 HRC 时钟作为系统时钟 0 1: 选择 XTAL 时钟作为系统时钟 1 0: 选择 LRC 时钟作为系统时钟 1 1: 选择 XTAL32 时钟作为系统时钟 注: 1, 切换的目标时钟源, 需保证在时钟稳定状态。 2, 流程参照【时钟源切换】章节 3, PWC_STPMCR.CKSMRC 位为 1 时, 停止模式唤醒后, 此寄存器初始化, 系统时钟源选择 HRC 时钟。	RW

4.9.14 CMU 时钟分频配置寄存器(CMU_SCKDIVR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						SCKDIV[2:0]	

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”,写入时写“0”	RW
b2~b0	SCKDIV[2:0]	CPU 及 周 边 时 钟 (HCLK) 分频选择	000: 系统时钟 001: 系统时钟的 2 分频 010: 系统时钟的 4 分频 011: 系统时钟的 8 分频 100: 系统时钟的 16 分频 101: 系统时钟的 32 分频 110: 系统时钟的 64 分频 其他禁止设定 注: 停止模式复归时, 根据 PWC_STPMCR.CKSMRC 位设定。CKSMRC 为 1 时, 停止模式唤醒后, 此寄存器初始化。CKSMRC 为 0 时, 停止模式唤醒后, 保持停止模式前分频设定,	RW

4.9.15 CMU ADC 时钟配置寄存器(CMU_PERICKSEL)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res					PERICKSEL[2:0]		

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”,写入时写“0”	RW
B2~b0	PERICKSEL[2:0]	ADC 变换用时钟 (ADCLK) 分频选择	000: HCLK 001: HCLK 的 2 分频 010: HCLK 的 4 分频 011: HCLK 的 8 分频 100: HCLK 的 16 分频 101: HCLK 的 32 分频 除此以外禁止设定。	RW

4.9.16 CMU 功能时钟控制寄存器(CMU_FCG)

复位值: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
USAR T6	USAR T5	SPI	I2C	USAR T4	USAR T3	USAR T2	USAR T1	RTC	Res						TMR0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMRB _8	TMR B_7	TMRB _6	TMRB _5	TMRB _4	TMRB _3	TMRB _2	TMRB _1	CRC	Res	DMA	AOS	Res	CTC	ADC	

位/位域	标记	位名	功能	读写
b31	USART6	USART6 功能控制	0: USART6 功能使能 1: USART6 功能无效	RW
b30	USART5	USART5 功能控制	0: USART5 功能使能 1: USART5 功能无效	RW
b29	SPI	SPI 功能控制	0: SPI 功能使能 1: SPI 功能无效	RW
b28	I2C	I2C 功能控制	0: I2C 功能使能 1: I2C 功能无效	RW
b27	USART4	USART4 功能控制	0: USART4 功能使能 1: USART4 功能无效	RW
b26	USART3	USART3 功能控制	0: USART3 功能使能 1: USART3 功能无效	RW
b25	USART2	USART2 功能控制	0: USART2 功能使能 1: USART2 功能无效	RW
b24	USART1	USART1 功能控制	0: USART1 功能使能 1: USART1 功能无效	RW
b23	RTC	RTC 功能控制	0: RTC 功能使能 1: RTC 功能无效	RW
b22~b17	Res	保留位	读出时为“1”,写入时写“1”	RW
b16	TMR0	TMR0 功能控制	0: TMR0 功能使能 1: TMR0 功能无效	RW
b15	TMRB_8	TMRB_8 功能控制	0: TMRB_8 功能使能 1: TMRB_8 功能无效	RW
b14	TMRB_7	TMRB_7 功能控制	0: TMRB_7 功能使能 1: TMRB_7 功能无效	RW
b13	TMRB_6	TMRB_6 功能控制	0: TMRB_6 功能使能 1: TMRB_6 功能无效	RW
b12	TMRB_5	TMRB_5 功能控制	0: TMRB_5 功能使能 1: TMRB_5 功能无效	RW
b11	TMRB_4	TMRB_4 功能控制	0: TMRB_4 功能使能 1: TMRB_4 功能无效	RW
b10	TMRB_3	TMRB_3 功能控制	0: TMRB_3 功能使能 1: TMRB_3 功能无效	RW
b9	TMRB_2	TMRB_2 功能控制	0: TMRB_2 功能使能 1: TMRB_2 功能无效	RW
b8	TMRB_1	TMRB_1 功能控制	0: TMRB_1 功能使能	RW

位/位域	标记	位名	功能	读写
			1: TMRB_1 功能无效	
b7	CRC	CRC 功能控制	0: CRC 功能使能 1: CRC 功能无效	RW
b6	Res	保留位	读出时为“1”,写入时写“1”	RW
b5	DMA	DMA 功能控制	0: DMA 功能使能 1: DMA 功能无效	RW
b4	AOS	AOS 功能控制	0: AOS 功能使能 1: AOS 功能无效	RW
b3~b2	Res	保留位	读出时为“1”,写入时写“1”	RW
b1	CTC	CTC 功能控制	0: CTC 功能使能 1: CTC 功能无效	RW
b0	ADC	ADC 功能控制	0: ADC 功能使能 1: ADC 功能无效	RW

4.9.17 CMU MCO1 时钟输出配置寄存器(CMU_MCO1CFGR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO1EN	MCO1DIV[2:0]			MCO1SEL[3:0]			

位/位域	标记	位名	功能	读写
b7	MCO1EN	MCO 输出许可	0: 禁止 MCO 输出 1: 允许 MCO 输出	RW
b6~b4	MCO1DIV[2:0]	MCO 分频选择	000: 1 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频	RW
b3~b0	MCO1SEL[3:0]	MCO 时钟源选择	0000: HRC 时钟 0001: LRC 时钟 0010: XTAL 时钟 0011: XTAL32 时钟 0100: 系统时钟 其他禁止设定。	RW

5 电源控制 (PWC)

5.1 概述

电源控制器用来控制芯片的多个电源域在多个运行模式和低功耗模式下的电源供给、切换、检测。芯片的工作电压(V_{DD})为 2.7V 到 5.5V。

低电压检测单元(LVD)提供了上电复位(POR)、掉电复位(PDR)、低电压检测(LVD)等功能,其中 POR、PDR 通过检测 V_{DD} 电压,控制芯片复位动作。LVD 通过检测 V_{DD} 电压或者外部输入比较,根据寄存器设定设置使芯片产生复位或者中断。

5.2 电源分布

图 5-1 是芯片的电源分布图。芯片由 V_{DD} 域、内核电压域、AVDD 电源域构成。

V_{DD} 域通过 V_{DD}/V_{SS} 引脚供电,由电源控制逻辑(PWC)、低电压检测单元(LVD)、电压调节器(LDO)、振荡器电路构成。振荡器电路包括外部高速振荡器(XTAL)、外部低速振荡器(XTAL32)等。

内核电压域由 CPU、数字外设等数字逻辑、RAM、FLASH 等构成,通过 LDO 产生的 REGC 供电。

模拟电源域主要由数模转换器(ADC)、模拟系的输入输出管脚构成,由 V_{DD}/V_{SS} 引脚供电。

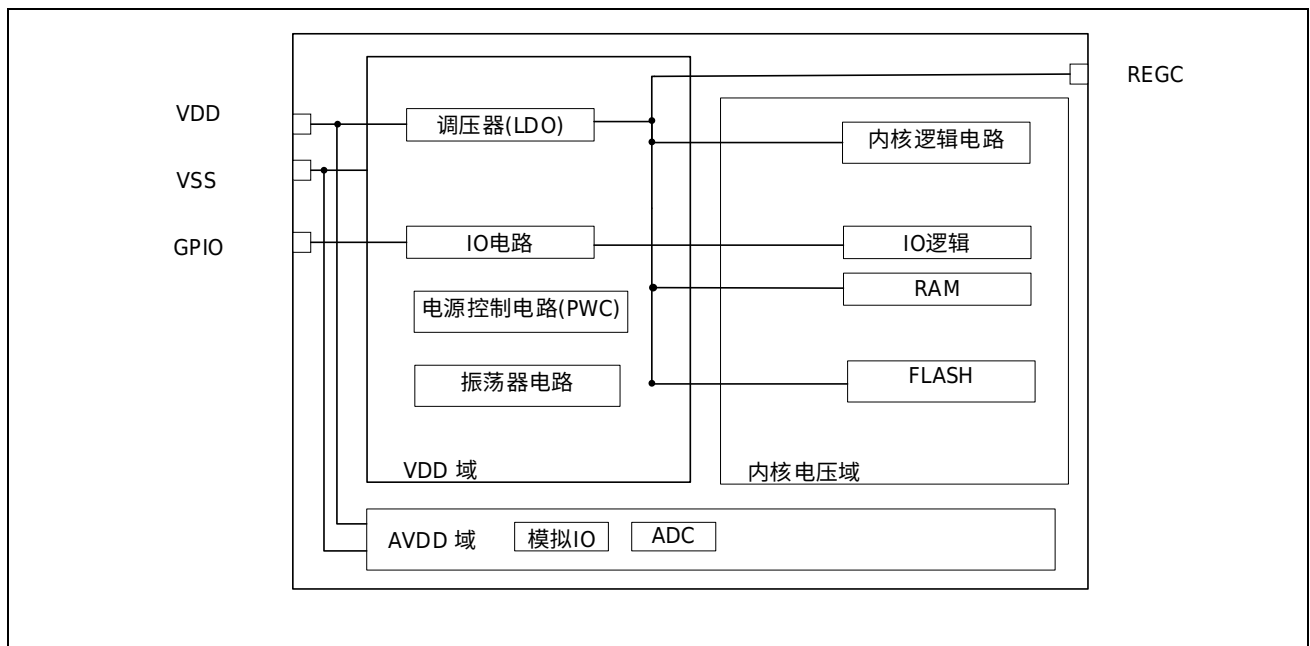


图 5-1 电源构成图

5.3 低电压检测单元(LVD)说明

低电压检测单元(LVD)包括上电复位(POR)、掉电复位(PDR)、低电压检测(LVD)。

5.3.1 上电复位/掉电复位动作说明

芯片集成有上电复位、掉电复位电路。上电复位、掉电复位波形如图 5-2 所示，当 V_{DD} 高于指定阈值 V_{POR} 之后，经过 T_{RSTPOR} 时间之后，芯片解除上电复位状态，CPU 开始执行代码。当 V_{DD} 低于 V_{PDR} 时，芯片保持复位状态。使用上电复位时，复位管脚 NRST 必须为 1。如果复位管脚被下拉，芯片将通过管脚复位的方式复位启动。

V_{POR} 、 V_{PDR} 、 T_{RSTPOR} 等参数的详细信息，请参考数据手册的电气特性。

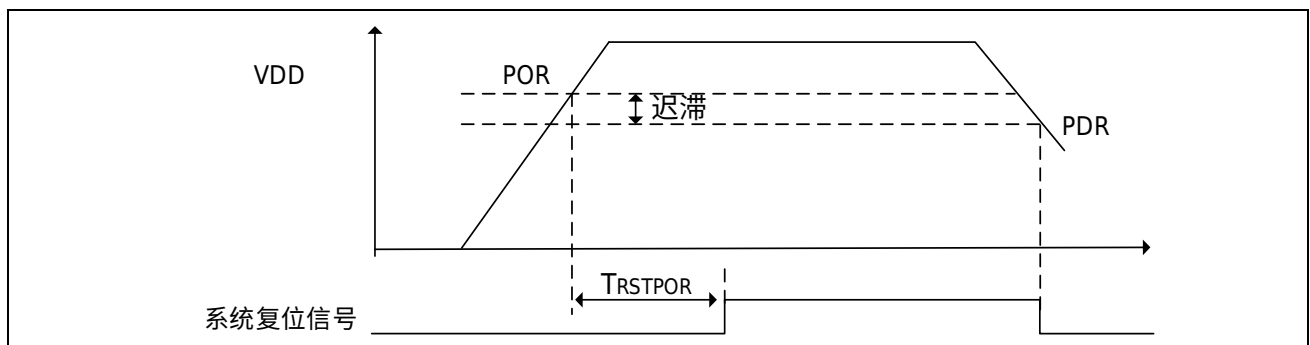


图 5-2 上电复位、掉电复位波形

5.3.2 低电压检测(LVD)

低电压检测(LVD)阈值通过 LVD ICG 控制寄存器(LVDICGCR)的 $LVDLVL$ 、 $LVDDIS$ 位进行配置。LVD 检测电压可以从 12 个阈值中选择。低电压检测电路通过 LVDICGCR 的 $IRDIS$ 和 IRS 位，选择产生复位或者中断功能。

电源电压经过各个检测电路的阈值电压点时，可将该事件编程配置成复位/中断(可屏蔽/不可屏蔽)。低电压检测主要特性如表 5-1 所示。

表 5-1 LVD 特性

项目	LVD
检测对象	V_{DD} 下降/上升过程中是否经过阈值电压点(VLVD) 或者外部比较电压是否经过内部基准电压
检测电压点	LVDICGCR 的 $LVDLVL$
复位	复位: $V_{DD} < VLVD$; 复位解除: $V_{DD} > VLVD$ 经过一定复位处理时间。
中断	LVDICGCR 的 $NMIS$ 位配置成低电压检测的可屏蔽或者非可屏蔽中断 V_{DD} 经过阈值电压点(VLVD)
滤波功能	数字滤波

通过配置 LVDICGCR 的 $LVDDIS$ 和 $LVDLVL$ 位，可以选择 LVD 有效及阈值。

表 5-2 LVD 电压选择

LVDICGCR.LVDDIS	LVDICGCR.LVDLVL	说明
1	XXXX	LVD 无效
0	0000	LVD 有效、选择 LVD 阈值 0(VLVD0)
0	0001	LVD 有效、选择 LVD 阈值 1(VLVD1)
0	0010	LVD 有效、选择 LVD 阈值 2(VLVD2)
0	0011	LVD 有效、选择 LVD 阈值 3(VLVD3)
0	0100	LVD 有效、选择 LVD 阈值 4(VLVD4)
0	0101	LVD 有效、选择 LVD 阈值 5(VLVD5)
0	0110	LVD 有效、选择 LVD 阈值 6(VLVD6)
0	0111	LVD 有效、选择 LVD 阈值 7(VLVD7)
0	1000	LVD 有效、选择 LVD 阈值 8(VLVD8)
0	1001	LVD 有效、选择 LVD 阈值 9(VLVD9)
0	1010	LVD 有效、选择 LVD 阈值 10(VLVD10)
0	1011	LVD 有效、选择 LVD 阈值 11(VLVD11)
0	1100	禁止设定
0	1101	禁止设定
0	1110	外部比较电压输入
0	1111	禁止设定

5.3.3 LVD 中断/复位框图

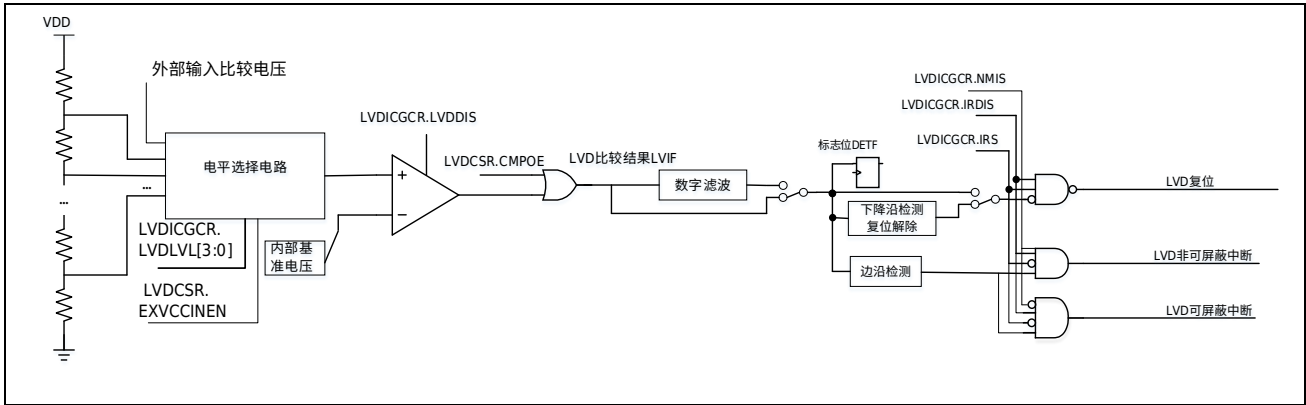


图 5-3 LVD 中断/复位框图

5.3.4 输入/输出引脚

引脚名	输入/输出	功能
LVDINP	输入	外部输入 LVD 比较电压

5.3.5 LVD 中断和复位

在停止模式中使用 LVD 电路时，必须将数字滤波器无效。

下图是 LVD 中断的运行时序图，DETIF 需要清零后才能再次发生中断。

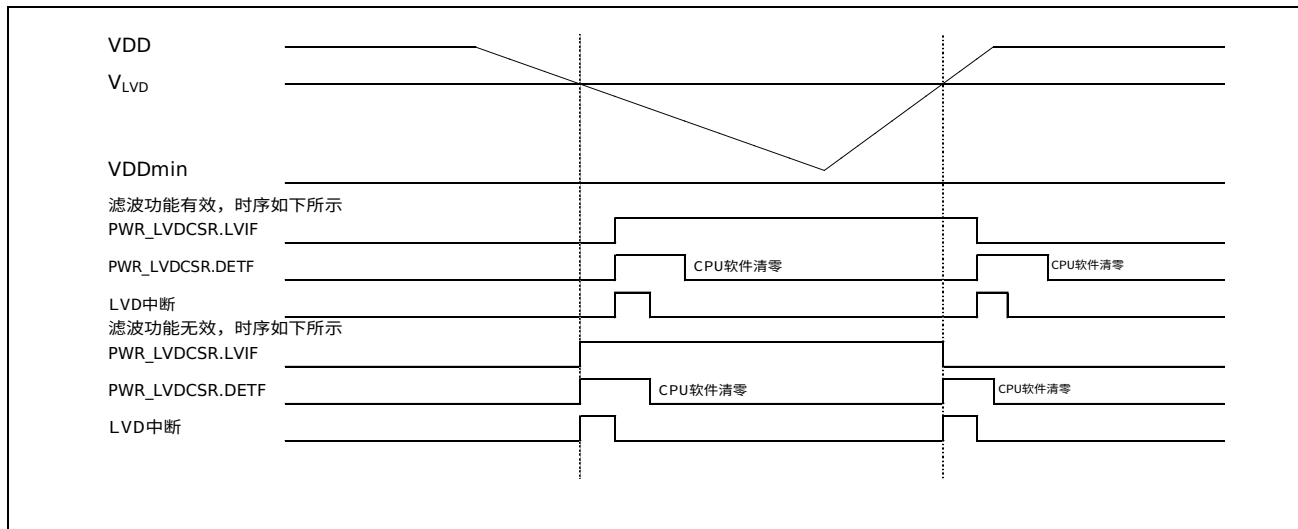


图 5-4 LVD 中断时序图

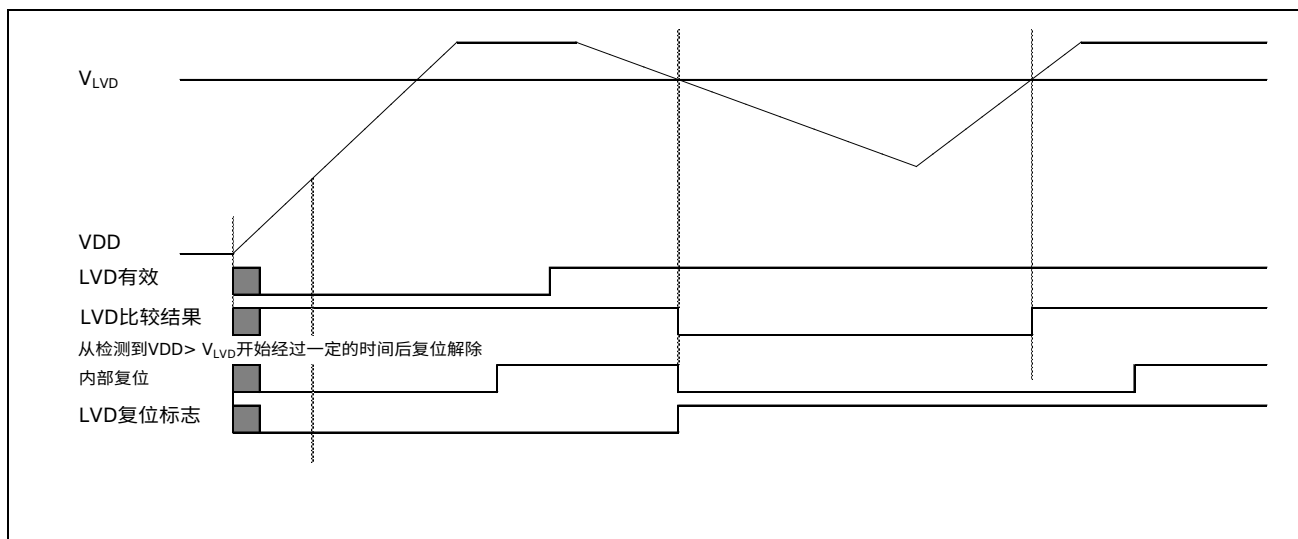


图 5-5 LVD 复位时序图

5.3.6 内部电压采样和检测功能

芯片内部电压采样和检测功能包括基准电压测量，温度传感器电压等两个功能。基准电压测量通路，是使用 ADC 测量基准电压的功能。温度传感器测量通路，是使用 ADC 测量内置温度传感器电压的功能。

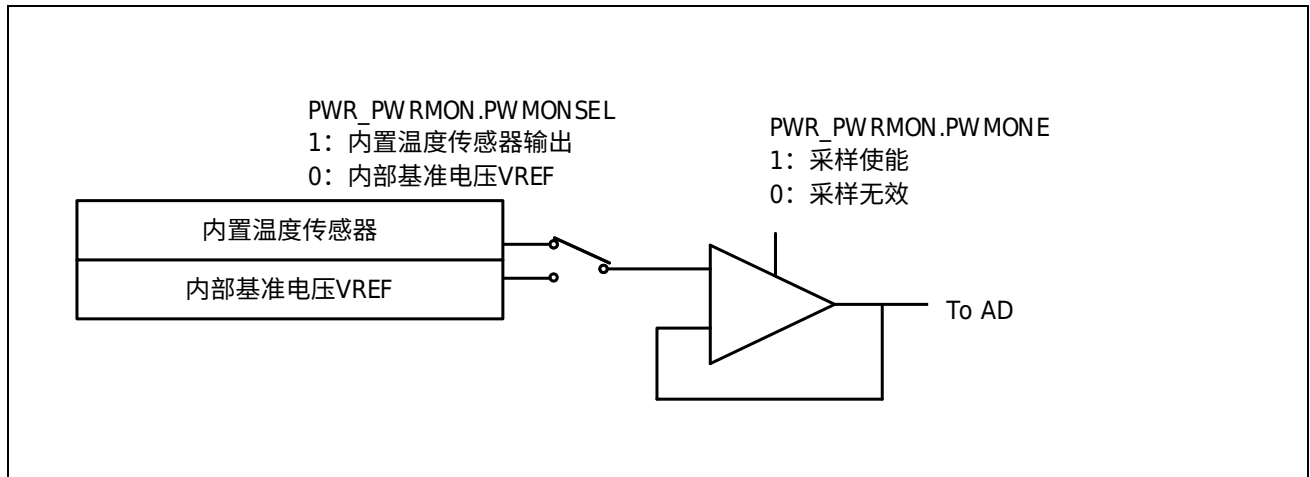


图 5-6 内部电压采样示意图

■ 基准电压测量通路

使用基准电压测量通路时，需要按照下述步骤选中基准电压测量通路。

1. PWC_PWRMON.PWMONSEL=0，选择内部基准电压
2. PWC_PWRMON.PWMONE=1，使能内部电压测量功能
3. ADC_EXCHSEL.R.EXCHSEL=1，使 ADC 选通内部电压测量通道
4. 等待 50uS 后使用 ADC 测量内部基准电压

■ 内置温度传感器测量通路

使用内置温度传感器测量通路时，需要按照下述步骤选中内置温度传感器测量通路。

1. PWC_PWRMON.PWMONSEL=1，选择内置温度传感器输出
2. PWC_PWRMON.PWMONE=1，使能内部电压测量功能
3. ADC_EXCHSEL.R.EXCHSEL=1，使 ADC 选通内部电压测量通道
4. 等待 50uS 后使用 ADC 测量内部基准电压

5.4 动作模式与低功耗模式

系统复位或者上电复位之后，芯片的所有电源域都处于供电状态，芯片进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。

为了节省 CPU 不需要运行时的功耗，系统提供了睡眠模式、停止模式等两种低功耗模式。在睡眠模式中，芯片的 Cortex-M0+内核停止动作，外设保持运行状态；在停止模式中，芯片的外设和 CPU 都停止动作。

用户可以根据应用选择运行模式和低功耗模式，以在低功耗、短启动时间、可唤醒源、和系统执行效率之间寻求最佳平衡。

低功耗模式的运行条件及各模块在低功耗模式下的状态如表 5-3 所示。

表 5-3 低功耗模式的运行条件及各模块在低功耗模式下的状态

项目	睡眠模式	停止模式
进入	PWC_STPMCR.STOP=0 WFI	PWC_STPMCR.STOP=1 WFI
解除	任意中断或者复位	停止模式下可以使用的中断或复位
外部高速振荡器	工作可设	停止
外部低速振荡器	工作可设	工作可设
内部高速振荡器	工作可设	停止
内部低速振荡器	工作可设	工作可设
CPU	停止(保持)	停止(保持)
RAM	工作可设	停止(保持)
Flash	工作可设	停止(保持)
调压器	工作 驱动可以调整	工作 驱动可以调整
上电复位电路	工作	工作
低电压检测模块 LVD	工作可设	工作可设
WDT	工作可设	工作可设
RTC	工作可设	工作可设
Timer0	工作可设	工作可设
其他周边外设模块	工作可设	停止(保持)
AD	工作可设	停止
XT1-XT2	工作可设	作为外部低速振荡器的管脚使用时，保持振荡器动作；设为 GPIO 或者其他外设功能时，请设定保持两个管脚为同样的电平
X1-X2	工作可设	作为外部高速振荡器使用时，振荡器停止震荡，管脚状态保持进入 STOP 模式前的状态；设为 GPIO 或者其他外设功能时，保持 STOP 之前的状态
NRST 管脚	工作	工作

项目	睡眠模式	停止模式
	内置上拉电路有效。芯片外面也可以通过电阻上拉到 V _{DD}	内置上拉电路有效。芯片外面也可以通过电阻上拉到 V _{DD}
P40/TOOL0	工作可设; 作为调试功能时, 内置上拉电路有效	保持; 作为调试功能时, 内置上拉电路有效
P50-P51	工作可设; 作为 SWD 功能时, 内置上拉电路有效	保持; 作为 SWD 功能时, 内置上拉电路有效
其他 GPIO	工作可设	保持

5.4.1 运行模式

芯片可以根据不同的频率范围设定相应的 LDO 驱动, 从而降低芯片功耗, 可以根据表 5-4 设定 PWC_PWRC.PWRDRV[2:0]。

表 5-4 运行模式说明

运行模式	频率范围	寄存器设定
		PWC_PWRC.PWRDRV[2:0]
高速模式	32KHz 以上, 小于等于 32MHz	0b111
超低速模式	32KHz	0b000

频率切换需要遵循如下流程。

1. 高速向低速切换时

- 1) 设置在超低速模式下使用的时钟源, 确保时钟源符合超低速模式下的频率要求
- 2) 关闭在超低速模式下不需要使用的时钟源及模块, 确认 Flash 不处于编程或者擦除状态
- 3) 设置 PWC_PWRC.PWRDRV[2:0]
- 4) 等待 TSWMD1(30uS)
- 5) 将系统时钟切换到低速时钟, 芯片在超低速模式下动作

2. 低速向高速切换时

- 1) 设置 PWC_PWRC.PWRDRV[2:0]
- 2) 等待 TSWMD2(30uS)
- 3) 设置在高速动作时的时钟源, 将系统时钟切换到高速时钟

5.4.2 睡眠模式

在睡眠模式中，CPU 停止运行，它的内部寄存器保持进入睡眠模式之前的状态。看门狗以外的外设和其他系统模块的动作状态不会改变。

通过 ICG 设置为自动启动时，如果初始化配置寄存器 0（ICG0）的 SWDTSLP0FF 位为 1，看门狗在睡眠模式下停止计数；如果 SWDTSLP0FF 位为 0，看门狗在睡眠模式下继续计数。如果看门狗没有通过 ICG0 的 SWDTAUTS 位未设置为自动启动，通过软件启动的方式启动看门狗，则如果 WDT_CR.SLP0FF 位为 1，看门狗在睡眠模式下停止计数；如果 WDT_CR.SLP0FF 位为 0，看门狗在睡眠模式下不停止计数。

■ 进入睡眠模式

在 PWC_STPMCR.STOP=0 时执行 WFI 指令即可进入睡眠模式。

■ 退出睡眠模式

任意中断、复位都可以将芯片从睡眠模式唤醒。通过中断唤醒时，芯片进入中断处理程序；通过复位退出睡眠模式时，芯片进入复位状态。

5.4.3 停止模式

在停止模式中，CPU、大部分外设和时钟源都停止动作。芯片保持 CPU 内部寄存器和 SRAM 数据，外设状态和管脚状态。在停止模式下，由于大部分时钟源停止工作，调压器也降低了驱动能力，因此芯片功耗会显著降低。

STOP 模式下的漏电流在不同的电压温度是不同的，设置的驱动能力必须满足芯片的漏电需要。

通过 ICG 设置为自动启动时，如果 ICG0 的 SWDTSLP0FF 位为 1，专用看门狗在停止模式下停止计数；如果 SWDTSLP0FF 位为 0，专用看门狗在停止模式下继续计数。

执行 WFI 指令进入停止模式之前，需要确保 FLASH 不处于编程或者擦除状态，并且振荡停止监测功能无效，否则芯片会进入睡眠模式而非停止模式。

在停止模式下，ADC 也会发生功耗，除非在进入停止模式前将其禁止。要禁止 ADC，需要 ADC_STR.STRT 位清“0”，在保证 A/D 转换器模拟模块停止稳定时间后在执行 WFI 指令进入停止模式。

STOP 模式唤醒时，通过 STPMCR 寄存器的位 CKSHRC 来选择唤醒后的时钟。CKSHRC 用于控制唤醒后的时钟源，CKSHRC=1 时，唤醒后的系统时钟源选择为 HRC；CKSHRC=0 时，唤醒后的系统时钟维持进入 STOP 之前的时钟源不变。

STOP 模式唤醒时，通过 PWC_STPMCR 的寄存器位 FLNWT 选择是否需要等待 Flash 稳定。FLNWT=0 时，唤醒时需要等待 Flash 稳定；FLNWT=1 时，唤醒时不需要等待 Flash 稳定。FLNWT 必须在程序运行在 RAM 上时才能置位，否则芯片从 STOP 唤醒后的动作不可保证。在程序运行在 RAM 上进入 STOP 模式时，选择 CKSHRC=1，FLNWT=1 将可以在最短的时间内唤醒系统。

在进入 STOP 模式之前将 PWC_STPMCR.HAPORDIS 设置为 1，进入 STOP 模式后，高精度 POR 电路被关闭，能够降低 STOP 模式下的功耗，但是这时不能保证 POR 的监测电压，需要系统确保电源电压稳定。

■ 进入停止模式

在 PWC_STPMCR.STOP=1 时执行 WFI 指令即可进入停止模式。表 5-3 给出了芯片的外设和时钟源在停止模式下的状态。

■ 解除停止模式

停止模式可以通过复位和中断解除。可以用于解除停止模式的复位方式有管脚复位，上电复位，LVD 复位、看门狗复位。可以用于解除停止模式的中断事件如下：

管脚 EIRQ 中断 管脚 EKEY 中断 低电压检测中断 看门狗下溢中断 实时时钟的定周期中断 闹钟中断 Timer0 比较匹配中断

芯片通过中断解除停止模式时，首先启动进入停止模式之前的使用到的时钟源。在所有的时钟源稳定之后，芯片解除停止模式。通过中断解除停止模式时，需要设置 INTC_NMIER 的不可屏蔽中断的许可位或者 INTC_WUPEN 的相应唤醒许可位，详细请参考【中断控制器（INTC）】。

5.5 降低功耗的方法

可以通过下列方法优化运行模式下的功耗。

- 降低系统时钟速度
- 关闭不使用的时钟源
- 设定功能时钟控制寄存器 CMU_FCG 关闭不需要使用的功能

5.5.1 降低系统时钟速度

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(HCLK)的速度。进入睡眠模式前，也可以使用这些预分频器降低外设速度。有关详细信息，请参考【4 时钟控制器（CMU）】。

5.5.2 关闭不使用的时钟源

芯片的系统时钟有 4 个时钟源：

- 外部高速振荡器(XTAL)
- 外部低速振荡器(XTAL32)
- 内部高速振荡器(HRC)
- 内部低速振荡器(LRC)

RTC 可以选择外部低速振荡器或者内部低速振荡器作为时钟源。

对于每一个时钟源，在未使用时都可以单独关闭，降低系统功耗。HRC 配有独立的电源电路，在 HRC 关闭后，清零 PWC_PWRC.VHRCE 位可以切断 HRC 的电源，进一步降低功耗。通过清零 WR_PWRC.VHRCE 位切断 HRC 电源之后，再次启动 HRC 需要置位 WR_PWRC.VHRCE,并等待 30us。

5.5.3 功能时钟停止

芯片的外设模块设有功能时钟停止功能，通过将寄存器对应的位置位，可以将不需要使用的模块停止运行，相应模块的时钟也停止供给，降低功耗。在模块停止状态下，模块内部的寄存器将维持停止之前的状态。时钟控制寄存器 CMU_FCG 的详细说明，参考【4 时钟控制器（CMU）】。

5.6 寄存器保护功能

寄存器保护功能用于将寄存器的写操作无效，以保护寄存器被意外改写。表 5-5 是寄存器保护位和被保护寄存器的列表。

表 5-5 寄存器保护列表

保护寄存器位	被保护寄存器
PWC_FPRC.CKRWE	CMU_PERICKSEL,CMU_XTALSTDSR,CMU_SCKDIVR,CMU_CKSWR CMU_XTALCR,CMU_XTALCFGR,CMU_XTALSTBCR,CMU_HRCCR CMU_OSCSTBSR,CMU_MCO1CFGR,CMU_XTALSTDCR,CMU_FCG CMU_XTAL32CR,CMU_XTAL32CFGR,CMU_XTAL32NFR,CMU_LRCCR
PWC_FPRC.PWRWE	PWC_STPMCR, PWC_PWRC, PWC_PWRMON, PWC_RAMCR,PWC_DBGCR,RMU_RSTF0
PWC_FPRC.LVRWE	PWC_LVDCSR,PWC_LVDICGCR

5.7 寄存器描述

表 5-6 PWC 寄存器列表

基地址 1: 0x4001 4000

寄存器	描述	偏移地址	位宽	复位值
PWC_STPMCR	停止模式控制寄存器	0x00	8	0x00
PWC_PWRC	电源模式控制寄存器	0x04	8	0xFF
PWC_PWRMON	电源监控寄存器	0x08	8	0x00
PWC_RAMCR	RAM 控制寄存器	0x0C	8	0x10
PWC_LVDCSR	LVD 控制状态寄存器	0x10	8	0x00
PWC_FPRC	功能保护控制寄存器	0x20	16	0x0000
PWC_DBGCR	调试模式控制寄存器	0x40	8	0x00

基地址 2: 0x4000 0A84

寄存器	描述	偏移地址	位宽	复位值
PWC_LVDICGCR	LVD ICG 控制寄存器	0x00	16	由 ICG1[15:0] 的值决定

5.7.1 停止模式控制寄存器(PWC_STPMCR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
STOP	Res			HAPORDIS	Res	CKSHRC	FLNWT

位/位域	标记	位名	功能	读写
b7	STOP	STOP 模式控制位	0: CPU WFI 进入 SLEEP 1: CPU WFI 进入 STOP	RW
b6-b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	HAPORDIS	高精度 POR 无效控制	0: 高精度 POR 在 STOP 模式下有效 1: 高精度 POR 在 STOP 模式下无效	RW
b2	Res	保留位	读出时为“0”, 写入时写“0”	RW
b1	CKSHRC	STOP 唤醒后系统时钟 切换控制	0: 不切换系统时钟 1: 切换为 HRC 的 n 分频	RW
b0	FLNWT	Flash 稳定等待控制	0: STOP 唤醒时等待 Flash 稳定 1: STOP 唤醒时不等待 Flash 稳定	RW

5.7.2 电源模式控制寄存器 (PWC_PWRC)

复位值: 0xFF

b7	b6	b5	b4	b3	b2	b1	b0

HRCPWDY	VHRCE	Res	PWDRV[2:0]

位/位域	标记	位名	功能	读写
b7	HRCPWDY	HRC LDO 状态	0: HRC LDO 未准备好 1: HRC LDO 可以使用 HRC 必须在本位为 1 之后等待 30uS 再启动	R
b6	VHRCE	HRC 的 LDO 使能控制	0: HRC 的 LDO 掉电 1: HRC 的 LDO 不掉电	RW
b5-b3	Res	保留位	读出时为“1”,写入时请写“1”	RW
b2-b0	PWDRV[2:0]	电源驱动控制寄存器	000: 仅限系统动作在 32K 时钟时使用 111: 系统频率在 32MHz 以下 其他: 设定禁止	RW

5.7.3 电源监控寄存器(PWC_PWRMON)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						PWMONE	PWMONSEL

位/位域	标记	位名	功能	读写
b7-b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1	PWMONE	电源内部电压监测使能	使用 AD 采样芯片内部电压时, 需要设置本位为 1 0: 无效 1: 有效	RW
b0	PWMONSEL	内部电压选择位	0: 电源内置基准电压 1: 温度传感器电压	RW

5.7.4 RAM 控制寄存器(PWC_RAMCR)

复位值: 0x10

b7	b6	b5	b4	b3	b2	b1	b0
Res		RPEF	RPERDIS	Res		RPRTA[1:0]	

位/位域	标记	位名	功能	读写
b7-b6	Res	保留位	读出时为“0”,写入时写“0”	-

b5	RPEF	RAM 奇偶校验错误标志位	0: 没有产生奇偶校验错误 1: 发生奇偶校验错误 始终有效, 只要产生奇偶校验错误就置位。 只能写 0 清除。	RW
b4	RPERDIS	RAM 奇偶校验复位使能	0: RAM 奇偶校验错误时产生复位 1: RAM 奇偶校验出现错误时不产生复位	RW
b3-b2	Res	保留位	读出时为“0”, 写入时写“0”	RW
b1-b0	RPRTA[1:0]	RAM 访问保护区域	00: 无效, 能写 RAM 01: RAM 起始地址的 128 字节 10: RAM 起始地址的 256 字节 11: RAM 起始地址的 512 字节	RW

5.7.5 功能保护控制寄存器(PWC_FPRC)

复位值: 0x00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PWC_FPRCWE[7:0]								FPRCB7	FPRCB6	FPRCB5	FPRCB4	LVRWE	FPRCB2	PWRWE	CKRWE

位/位域	标记	位名	功能	读写
b15~b8	PWC_FPRCWE[7:0]	PWC_FPRC 寄存器写使能	写入 0xA5h 的同时能够更新 PWC_FPRC 值, 否则对低 8 位写入值无效。读出时为 0x00。	RW
b7	FPRCB7	FPRC 位 7	预留	RW
b6	FPRCB6	FPRC 位 6	预留	-
b5	FPRCB5	FPRC 位 5	预留	RW
b4	FPRCB4	FPRC 位 4	预留	RW
b3	LVRWE	FPRC 位 3	LVD 相关寄存器写使能 0: 写保护 1: 写使能	RW
b2	FPRCB2	FPRC 位 2	预留	-
b1	PWRWE	FPRC 位 1	电源控制相关寄存器写使能 0: 写保护 1: 写使能	RW
b0	CKRWE	FPRC 位 0	时钟控制相关寄存器写使能 0: 写保护 1: 写使能	RW

5.7.6 LVD ICG 控制寄存器(PWC_LVDICGCR)

复位值：0xXXXX，复位值根据初始化配置寄存器 1(ICG1)设定值确定。

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LVDDIS	IRDIS	IRS	NMIS	LVDLVL[3:0]				Res				DFDIS	DFS[1:0]		

位/位域	标记	位名	功能	读写
b15	LVDDIS	低电压检测允许	0: 低电压检测电路有效 1: 低电压检测电路无效	RW
b14	IRDIS	LVD 的中断和复位禁止位	0: 中断复位有效 1: 中断复位无效	RW
b13	IRS	LVD 动作模式选择	0: 中断模式 1: 复位模式	RW
b12	NMIS	LVD 中断类型选择	0: 可屏蔽中断 1: 不可屏蔽中断	RW

选择 LVD 阈值电压 (VLVD0-VLVD11) 或 LVDINP 输入

b11~b8	LVDLVL[3:0]	LVD 阈值电压选择	LVDLVL	选择阈值	下降电压	上升电压	RW
			0000	VLVD0	3.94	4.06	
			0001	VLVD1	3.65	3.75	
			0010	VLVD2	3.06	3.15	
			0011	VLVD3	2.94	3.03	
			0100	VLVD4	2.86	2.94	
			0101	VLVD5	2.74	2.82	
			0110	VLVD6	2.65	2.73	
			0111	VLVD7	2.55	2.63	
			1000	VLVD8	2.44	2.51	
			1001	VLVD9	2.03	2.10	
			1010	VLVD10	1.93	1.99	
			1011	VLVD11	1.83	1.89	
			1100	-	-	-	
			1101	-	-	-	
			1110 ^{*1}	-	-	-	
			1111	-	-	-	

*1: LVDLVL = 1110 时, 并且 LVDCSR 的 EXVCCINEN=1 时选择外部电压 (LVDINP) 输入

下降电压和上升电压为典型值, 详细数据请参考电气特性

b7~b3	Res	保留位	读出时为“0”, 写入时写“0”	RW
b2	DFDIS	LVD 数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效	RW
b1-b0	DFS[1:0]	LVD 数字滤波采样能力 选择	00: 2 个 LRC 周期 01: 4 个 LRC 周期 10: 8 个 LRC 周期 11: 16 个 LRC 周期 注: 只能在 DFDIS 位为“1”时改写该位	RW

5.7.7 LVD 控制状态寄存器(PWC_LVDCSR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
CMPOE	Res		DETF	LVIF	Res		EXVCCINEN

位/位域	标记	位名	功能	读写
b7	CMPOE	LVD 比较结果输出控制	0: LVD 比较结果输出无效 1: LVD 比较结果输出有效 CMPOE 在 PWC_LVDICGCR 的 LVDDIS 位为 1 时被固定为 0。	RW
b6~b5	Res	保留位	读出时为“0”,写入时写“0”	R
b4	DETF	LVD 检测标志位	0: LVD 未检测到 VDD 经过 VLVD 或者 LVDINP 未经过内部基准电压 1: LVD 检测到 VDD 经过 VLVD 或者 LVDINP 经过内部基准电压 注: 当 LVDDIS 位为“0”时, 此标志位有效。LVDDIS 位为“1”时, 标志位被清零。 该位只允许写 0。当该位为 1 时, 写 0 清除该位。	RW
b3	LVIF	低电压检测标志	0: VDD>VLVD 或者 LVD 关闭 1: VDD<VLVD	R
b2-b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	EXVCCINEN	外部输入控制位	0: 外部输入无效 1: 外部输入有效 使用外部输入模式时, PWC_LVDICGCR 的 LVDLVL 必须设置为 0b1110	RW

5.7.8 调试模式控制寄存器(PWC_DBGCR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						DBGWKF	DBGEN

位/位域	标记	位名	功能	读写
b7-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1	DBGWKF	唤醒标志	0: 调试模式下 STOP 唤醒事件未发生 1: 调试模式下 STOP 唤醒事件发生	R
b0	DBGEN	调试使能	0: 调试模式无效 1: 调试模式有效	RW

6 初始化配置 (ICG)

6.1 概述

芯片复位解除后，硬件电路会读取主闪存地址 0x0000 00C0~0x0000 00DB（其中 0x0000 00C8~0x0000 00DB 为预留功能地址，该 20bytes 地址需要用户设定全 1 以确保芯片动作正常）把数据加载到初始化配置寄存器，用户需要编程或擦除扇区 0 来修改初始化配置寄存器。寄存器复位值后初始值由用户 FLASH 地址数据确定。

6.2 寄存器描述

表 6-1 ICG 基地址

名称	基地址	描述
ICG	0x0000 00C0	ICG基地址

表 6-2 ICG 寄存器

寄存器	描述	偏移地址	位宽	复位值
ICG0	初始化配置寄存器 0	0x000	32	不定
ICG1	初始化配置寄存器 1	0x004	32	不定
ICG2	初始化配置寄存器 2	0x008	32	不定
ICG3	初始化配置寄存器 3	0x00C	32	不定
ICG4	初始化配置寄存器 4	0x010	32	不定
ICG5	初始化配置寄存器 5	0x014	32	不定
ICG6	初始化配置寄存器 6	0x018	32	不定

6.2.1 初始化配置寄存器 0(ICG0)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												HRCFREQS[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			SWD TSLP OFF	SWDTWDPT[3:0]				SWDTCK[3:0]				SWDTPERI [1:0]		SWD TITS	SWD TAUT S

位/位域	标记	位名	功能	读写																							
b31~b20	Res	保留位	功能预留位，写入 1，读出 1。	R																							
内部高速振荡器频率选择。																											
b19~b16	HRCFREQS[3:0]	HRC 频率选择位	<table><tr><td rowspan="2">HRCFREQS[2:0]</td><td colspan="2">HRC 频率</td></tr><tr><td>HRCFREQS[3]=0</td><td>HRCFREQS[3]=1</td></tr><tr><td>000</td><td>此处设定禁止。</td><td>此处设定禁止。</td></tr><tr><td>001</td><td>32MHz</td><td>24MHz</td></tr><tr><td>010</td><td>16MHz</td><td>12MHz</td></tr><tr><td>011</td><td>8MHz</td><td>6MHz</td></tr><tr><td>100</td><td>4MHz</td><td>3MHz</td></tr><tr><td>101~111</td><td>2MHz</td><td>1.5MHz</td></tr></table>	HRCFREQS[2:0]	HRC 频率		HRCFREQS[3]=0	HRCFREQS[3]=1	000	此处设定禁止。	此处设定禁止。	001	32MHz	24MHz	010	16MHz	12MHz	011	8MHz	6MHz	100	4MHz	3MHz	101~111	2MHz	1.5MHz	R
			HRCFREQS[2:0]		HRC 频率																						
				HRCFREQS[3]=0	HRCFREQS[3]=1																						
			000	此处设定禁止。	此处设定禁止。																						
			001	32MHz	24MHz																						
			010	16MHz	12MHz																						
			011	8MHz	6MHz																						
			100	4MHz	3MHz																						
101~111	2MHz	1.5MHz																									
b15~b13	Res	保留位	功能预留位	R																							
b12	SWDTSLTPOFF	SWDT 在 Sleep, Stop 模式下计数停止	0: SWDT 在 sleep, stop 模式下计数不停止 1: SWDT 在 sleep, stop 模式下计数停止	R																							
b11~b8	SWDWDPT[3:0]	刷新允许区域 计数值百分比	SWDT 计数值刷新允许区间	R																							
			0000: 0%~100%																								
			0001: 0%~25%																								
			0010: 25%~50%																								
			0011: 0%~50%																								
			0100: 50%~75%																								
			0101: 0%~25%,50%~75%																								
			0110: 25%~75%																								
			0111: 0%~75%																								
			1000: 75%~100%																								
			1001: 0%~25%,75%~100%																								
			1010: 25%~50%,75%~100%																								
			1011: 0%~50%,75%~100%																								
			1100: 50%~100%																								
1101: 0%~25%,50%~100%																											
1110: 25%~100%																											
1111: 0%~100%																											
b7~b4	SWDTCK[3:0]	SWDT 计数时钟	0000: SWDTCLK	R																							
			0100: SWDTCLK/16																								
			0101: SWDTCLK/32																								

位/位域	标记	位名	功能	读写
			0110: SWDTCLK/64 0111: SWDTCLK/128 1000: SWDTCLK/256 1011: SWDTCLK/2048 其它值: 预留	
b3~b2	SWDTPER[1:0]	SWDT 计数溢出周期	00: 256 周期 01: 4096 周期 10: 16384 周期 11: 65536 周期	R
b1	SWDTITS	SWDT 中断选择	0: 中断请求 1: 复位请求	R
b0	SWDTAUTS	SWDT 自动启动	0: 复位后, SWDT 自动启动 (硬件启动) 1: 复位后, SWDT 停止状态	R

6.2.2 初始化配置寄存器 1(ICG1)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															LKUP DIS
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LVDD IS	IRDIS	IRS	NMIS	LVDLVL[3:0]				Res					DFDI S	DFS[1:0]	

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	功能预留位	R
b16	LKUPDIS	LOCKUP 复位禁止位	0: CPU LOCKUP 复位有效 1: CPU LOCKUP 复位无效	R
b15	LVDDIS	电压检测 1 允许	0: 电压检测电路有效 1: 电压检测电路无效	R
b14	IRDIS	LVD 的中断和复位禁止位	0: 中断复位有效 1: 中断复位无效	R
b13	IRS	LVD 动作模式选择	0: 中断模式 1: 复位模式	R
b12	NMIS	LVD 中断类型选择	0: 可屏蔽中断 1: 不可屏蔽中断	R

选择 LVD 阈值电压 (VLVD0-VLVD11) 或 LVDINP 输入

b11~b8	LVDLVL[3:0]	LVD 阈值电压选择	LVDLVL	选择阈值	下降电压	上升电压	R
			0000	VLVD0	3.94	4.06	
			0001	VLVD1	3.65	3.75	
			0010	VLVD2	3.06	3.15	
			0011	VLVD3	2.94	3.03	
			0100	VLVD4	2.86	2.94	
			0101	VLVD5	2.74	2.82	
			0110	VLVD6	2.65	2.73	
			0111	VLVD7	2.55	2.63	
			1000	VLVD8	2.44	2.51	
			1001	VLVD9	2.03	2.10	
			1010	VLVD10	1.93	1.99	
			1011	VLVD11	1.83	1.89	
			1100	-	-	-	
			1101	-	-	-	
			1110 ^{*1}	-	-	-	
			1111	-	-	-	

*1: LVDLVL = 1110 时, 并且 LVDCSR 的 EXVCCINEN=1 时选择外部电压 (LVDINP) 输入

下降电压和上升电压为典型值, 详细数据请参考电气特性

b7~b3	Res	保留位	功能预留位, 写入 1, 读出 1。	R
b2	DFDIS	LVD 数字滤波器屏蔽	0: 数字滤波器有效 1: 数字滤波器无效	R

位/位域	标记	位名	功能	读写
b1-b0	DFS[1:0]	LVD 数字滤波采样能力 选择	00: 2 个 LRC 周期 01: 4 个 LRC 周期 10: 8 个 LRC 周期 11: 16 个 LRC 周期 注: 只能在 DFDIS 位为“1”时改写该位	R

6.2.3 初始化配置寄存器 n(ICGn)n=2~6

复位值: 不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ICGn[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ICGn[15:0]															

位/位域	标记	位名	功能	读写
b31~b0	ICGn[31:0]	-	功能预留位 需要用户设定全 1 以确保芯片动作正常	R

7 嵌入式 FLASH (EFM)

7.1 概述

FLASH 接口通过系统总线对 FLASH 进行访问。该接口可对 FLASH 执行编程，扇区擦除和全擦除操作；通过指令预取加速代码执行。

7.2 主要特性

FLASH 读操作

- FLASH 编程，扇区擦除和全擦除操作
- 总线 4Bytes 预取指
- 支持安全保护*1

*1: 关于 Flash 安全保护的具体规格，请咨询销售窗口

7.3 嵌入式 FLASH

FLASH 具有以下主要特性：

- 容量最高 256KBytes，分为 512 个扇区，每个扇区为 512Bytes
- 1KBytes 的专用数据 Flash 存储器，分为扇区 512 和扇区 513
- 32 位宽数据读取
- 编程单位为 1Byte，擦除单位为 512Bytes

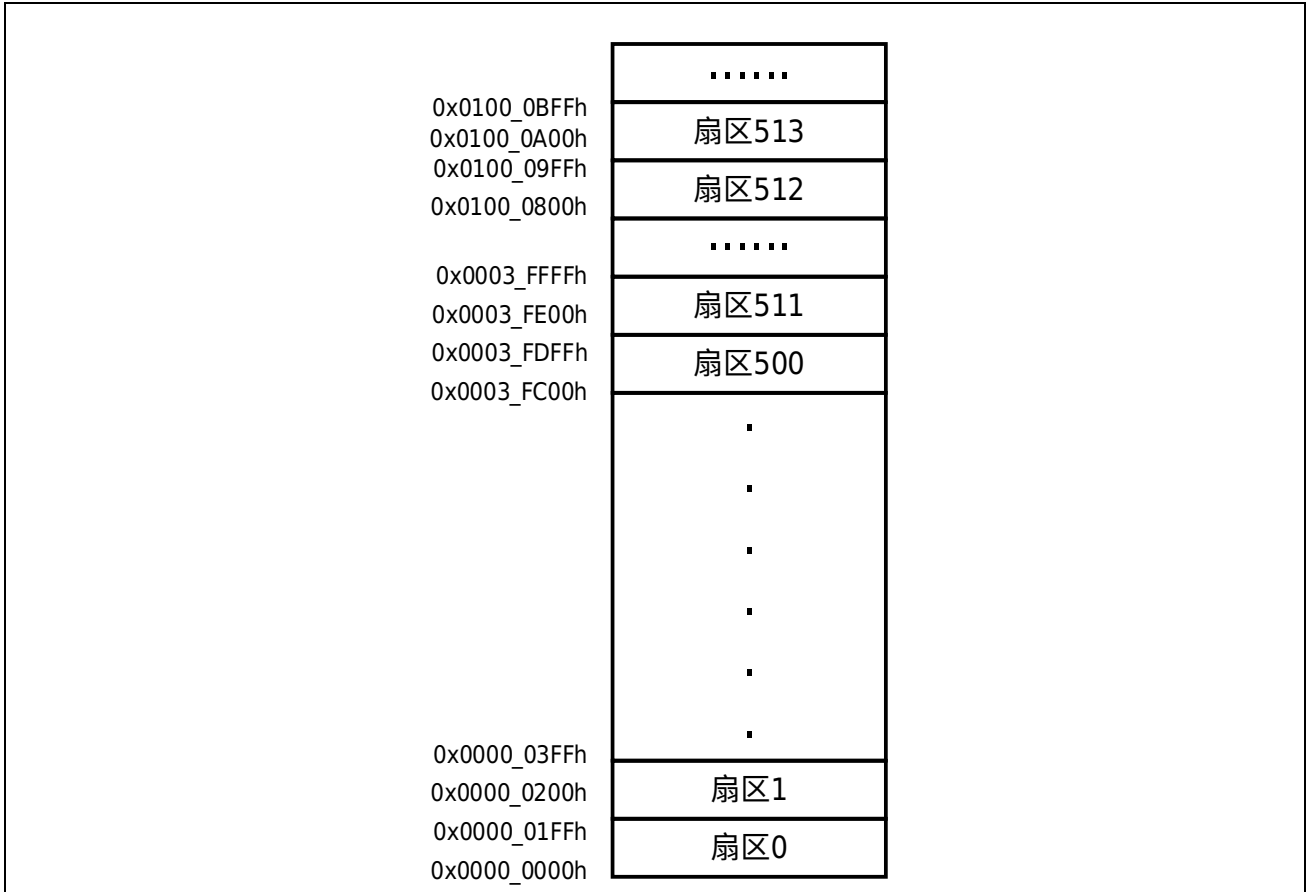


图 7-1 FLASH 地址分配

7.4 读接口

7.4.1 CPU 时钟和 FLASH 读取时间之间的关系

要正确读取 FLASH 数据，用户需要根据 CPU 动作频率在 FLASH 读模式寄存器(EFM_FRMC)中正确设定等待周期数(FLWT)。

系统复位后，CPU 时钟源为 HRC(频率由 ICG0 设定值决定)，FLASH 读等待周期为 2。建议用户按照以下步骤修改 CPU 主频和 FLASH 读等待周期位。等待周期数请参照表 7-1。

CPU 频率提高步骤：

1. 将新的读等待周期设定值写入寄存器 EFM_FRMC。
2. 读取寄存器 EFM_FRMC，检查新的等待周期是否设定成功。
3. 通过设定系统时钟源切换寄存器 CMU_CKSWR(CKSW[2:0]) 或者系统时钟配置寄存器 CMU_SCKDIVR(SCKDIV[2:0])来提升 CPU 时钟频率。
4. 读取寄存器 CMU_CKSWR 或者 CMU_SCKDIVR，检查新的设定是否成功。

CPU 频率降低步骤:

1. 通过设定系统时钟源切换寄存器 CMU_CKSWR(CKSW[2:0]) 或者系统时钟配置寄存器 CMU_SCKDIVR(SCKDIV[2:0])来降低 CPU 时钟频率。
2. 读取寄存器 CMU_CKSWR 或者 CMU_SCKDIVR，检查新的设定是否成功。
3. 将新的读等待周期设定值写入寄存器 EFM_FRMC。
4. 读取寄存器 EFM_FRMC，检查新的等待周期是否设定成功。

表 7-1 CPU 时钟频率和 FLASH 读等待周期对照表

CPU 时钟频率(HCLK)	FRMC 寄存器位 FLWT[1:0]设定
$24\text{MHz} < \text{HCLK} \leq 32\text{MHz}$	FLWT[1:0]=1 或 FLWT[1:0]=2 或 FLWT[1:0]=3 至少插入 1 个等待读周期
$\text{HCLK} \leq 24\text{MHz}$	FLWT[1:0]=0 插入 0 个等待读周期

7.5 FLASH 读预取

为了提升 CPU 性能，FLASH 控制器配置了读预取功能。在 CPU 空闲时，可使用主机总线的预取操作读取 FLASH 下一个 32 位数据。通过 EFM_FRMC 寄存器中的 PREFETE 位置 1，来使能预取功能。CPU 读取指令或数据地址与预取地址一致时，预取数据送给 CPU；若地址不一致，则预取数据作废，由 FLASH 读出数据送给 CPU。

7.6 FLASH 编程和擦除操作

FLASH 编程，扇区擦除，全擦除操作时会使用 HRC，请在这些操作前，打开 HRC，并等待 HRC 振荡稳定后再设定 FLASH 控制寄存器。FLASH 编程单位是 1Byte，对同一地址重复编程不能确保编程的正确性。FLASH 扇区擦除单位为 512Bytes。以下分别介绍，编程和擦除操作的设定步骤。

7.6.1 单次编程无回读模式

单编程无回读模式设定步骤如下：

- 1) 解除 FLASH 接口寄存器写保护 (EFM_FAPRT 先写 0x0123，再写 0x3210)；
- 2) 设定编程，擦写模式许可 (EFM_FWMC.PEMODE=1)；
- 3) 设定单次编程模式 (EFM_FWMC.PEMODE[2:0]=0b001)；
- 4) 对编程地址写入 32/16/8 位数据；
- 5) 等待 FLASH 操作完成且处于空闲状态 (EFM_FSR.OPTEND =1，EFM_FSR.RDY=1)；
- 6) 清除编程结束标志位 (EFM_FSR.OPTEND)；

7) 读取编程地址并判断是否和写入值一致：

一致，表示编程成功；不一致，表示该 Flash 地址已遭破坏，永久废弃。

7.6.2 单编程回读模式

单编程回读模式是指编程结束后自动读取编程地址并和写入数据对比，输出判断一致标志位 EFM_FSR.PGMISMTCH。

单编程回读模式设定步骤如下：

- 1) 解除 FLASH 接口寄存器写保护 (EFM_FAPRT 先写 0x0123，再写 0x3210)；
- 2) 设定编程，擦写模式许可 (EFM_FWMC.PEMODE=1)；
- 3) 设定单次编程回读模式 (EFM_FWMC.PEMODE[2:0]=0b010)；
- 4) 对编程地址写入 32/16/8 位数据；
- 5) 等待 FLASH 操作完成且处于空闲状态 (EFM_FSR.OPTEND =1, EFM_FSR.RDY=1)；
- 6) 判断编程自动读取结果标志位 (EFM_FSR.PGMISMTCH)；
- 7) 清除编程结束标志位 (EFM_FSR.OPTEND)。

7.6.3 连续编程操作

当连续对 FLASH 地址进行编程时，推荐使用连续编程模式。连续编程模式比单编程模式可以节省时间 50% 以上。连续编程模式时，连续编程 FLASH 的写入间隔不能超过 7us。连续编程操作设定步骤如下：

- 1) 解除 FLASH 接口寄存器写保护 (EFM_FAPRT 先写 0x0123，再写 0x3210)；
- 2) 设定编程，擦写模式许可 (EFM_FWMC.PEMODE=1)；
- 3) 设定连续编程模式。(EFM_FWMC.PEMOD[2:0]=0b011)；
- 4) 把步骤 6) 7) 8) 9) 10) 11) 传送到 FLASH 以外区域执行；
- 5) 跳转到步骤 4) 传送的目的地址；
- 6) 对编程地址写 32/16/8 位数据；
- 7) 等待操作结束标志位 (EFM_FSR.OPTEND) 置位；
- 8) 清除操作结束标志位，直至读到标志位 EFM_FSR.OPTEND 为 0；
- 9) 重复 6)，7)，8) 直至所有数据写完；
- 10) 退出连续编程模式 (EFM_FWMC.PEMOD[2:0]=0b000)；
- 11) 等待 FLASH 处于空闲状态 (EFM_FSR.RDY=1)；

12) 读取编程地址并判断是否和写入值一致：

一致，表示编程成功；不一致，表示该 Flash 地址已遭破坏，永久废弃。

注：

- 在 FLASH 连续编程期间，如果发生对 FLASH 的读操作，将会读到不定值。读操作会使读写冲突位 EFM_FSR.COLERR 置位。

7.6.4 擦除操作

EFM 提供了扇区擦除和全擦除两种擦除方式。对 FLASH 进行扇区擦除操作后，该扇区内地址(512Bytes 空间)数据刷新为全 1。对 FLASH 进行全擦除操作后，FLASH 区域 256KBytes 空间 (0x0000 0000~0x0003 FFFF) 数据刷新为全 1，对 1KBytes 的专用数据 Flash 区域地址 (0x0100 0800~0x0100 0BFF) 全擦除无效。扇区擦除和全擦除操作设定步骤如下：

- 1) 解除 FLASH 的寄存器写保护 (EFM_FAPRT 先写 0x0123，再写 0x3210)；
- 2) 设定编程，擦写模式许可 (EFM_FWMC.PEMODE=1)；
- 3) 设定擦除模式 (EFM_FWMC.PEMOD[2:0]=0b100)；
- 4) 全擦除时设定 (EFM_FWMC.PEMOD[2:0]=0b101)；
- 5) 对需要擦除扇区内的任意地址(地址需以 4 对齐)写入 32 位任意值；
- 6) 全擦除时对任意 FLASH 地址(地址需以 4 对齐)写入 32 位任意值；
- 7) 等待 FLASH 操作完成且处于空闲状态 (EFM_FSR.OPTEND =1, EFM_FSR.RDY=1)；
- 8) 清除擦除结束标志位 (EFM_FSR.OPTEND)。

7.6.5 总线保持功能

通过设定寄存器 EFM_FWMC.BUSHLDCTL 位，可设定 FLASH 编程，擦除期间，总线处于保持状态还是释放状态。FLASH 编程，擦除指令在 FLASH 上执行时，必须把该控制位设定为 0；擦除指令在 FLASH 以外空间（比如 RAM）执行时，可根据需要自由设定该控制位。

当设定 BUSHLDCTL 为 1（FLASH 编程，擦写期间，总线释放状态）时，在编程(连续编程除外)，擦写结束前(EFM_FSR.RDY=1)对 FLASH 的读写访问将会被忽视，标志位 EFM_FSR.COLERR 置位。

7.6.6 FLASH 擦除、编程窗口保护

对 FLASH 提供窗口保护功能，只有在允许区域 FLASH 才能被扇区擦除，编程，否则发生擦写错误中断。编程、扇区擦除操作时，硬件电路将预先判断是否在允许区域内，全擦除模式不受窗口保护限制。窗口的起始位置和结束位置由寄存器 EFM_FPMTEW，EFM_FPMTSW 设定。

具体保护功能如下：

- 寄存器 EFM_FPMTEW=寄存器 EFM_FPMTSW
 - 整个 FLASH 区域都可扇区擦除，编程。
- 寄存器 EFM_FPMTEW>寄存器 EFM_FPMTSW
 - FLASH 区域(0x0000 0000~0x0003 FFFF)可扇区擦除，编程区域在两者之间。
 - 1KBytes 的专用数据 Flash 区域(0x0100 0800~0x0100 0BFF)可扇区擦除，编程。
- 寄存器 EFM_FPMTEW<寄存器 EFM_FPMTSW
 - FLASH 区域(0x0000 0000~0x0003 FFFF)不可扇区擦除，编程。
 - 1KBytes 的专用数据 Flash 区域(0x0100 0800~0x0100 0BFF)可扇区擦除，编程。

7.6.7 扇区擦除取消功能

FLASH 扇区擦除期间，可通过扇区擦除取消功能，快速响应中断。扇区擦除取消功能许可后，FLASH 扇区擦除期间，MCU 中断发生，扇区擦除操作提前结束，FLASH 操作结束标志和扇区擦除取消标志置位，FLASH 回归到空闲状态。中断事件处理结束后，需要对擦除取消的扇区再次进行擦除以确保 FLASH 数据刷新为全 1。

扇区擦除取消功能设定步骤：

- 1) 配置 MCU 中断：
 - 可屏蔽中断配置请参考表 10-3
 - 不可屏蔽中断配置请参考【10.4.1 不可屏蔽中断】
- 2) 按照擦除操作步骤，设定 FLASH 扇区擦除模式；
- 3) 设定 FLASH 扇区擦除取消许可（FWMC.ERCALE=1）；
- 4) 设定 FLASH 扇区擦除期间总线保持（FWMC. BUSHLDCTL=0）；
- 5) 对需要擦除扇区内的任意地址（地址需以 4 对齐）写入 32 位任意值；
- 6) 等待 FLASH 操作完成且处于空闲状态（EFM_FSR.OPTEND =1，EFM_FSR.RDY=1）；
- 7) 清除擦除结束标志位（EFM_FSR.OPTEND）；
- 8) 判断 FLASH 是否处于扇区擦除取消状态（FSR.ERCAL=1）；
- 9) 如是，则清除扇区擦除取消标志位。并重复步骤 5) ~8)。

7.6.8 中断

EFM 模块共有 3 个中断，分别是 PE（编程/擦除）错误中断、读写冲突中断和操作结束中断。

当 FLASH 发生 PE 错误中断置位后，不能对 FLASH 进行编程/擦除/全擦除。必须对标志位清零后才能再次对 FLASH 进行编程/擦除/全擦除。

PE 错误中断 EFM_PEERR:

置位：

- 对 FLASH 窗口保护内地址执行编程，扇区擦除操作(PEPRTErr=1)。
- 未设定擦写模式时，执行对 FLASH 写操作(PEWERR=1)。
- 单编程回读模式时，编程地址自读值与写入值不一致(PGMISMTCH=1)。

清零：

寄存器 EFM_FSCLR 对应标志清除位写 1，状态位清零。

FLASH 读写冲突中断 EFM_COLERR:

置位：

- FLASH 连续编程模式发生 FLASH 读操作。
- FLASH 停止模式时发生 FLASH 读写操作。
- FLASH 编程，擦除未结束前，发生 FLASH 读写操作。

清零：

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

操作结束中断 EFM_OPTEND:

置位：

- 编程模式：单个地址编程结束。32/16 位数据写入时，FLASH 会编程 4/2 次，操作结束中断 (EFM_OPTEND) 只会发生 1 次。
- 擦除模式：扇区擦除，全擦除结束。

清零：

寄存器 EFM_FSCLR 对应清除位置 1，状态位清零。

7.7 寄存器描述

表 7-2 EFM 基地址

名称	基地址	描述
EFM	0x4000 0800	EFM基地址

表 7-3 EFM 寄存器列表

寄存器	描述	偏移量	位宽	复位值
EFM_FAPRT	FLASH 访问保护寄存器	0x0000	32	0x0000 0000
EFM_FSTP	FLASH 停止寄存器	0x0004	32	0x0000 0000
EFM_FRMC	FLASH 读模式寄存器	0x0008	32	0x0000 0002
EFM_FWMC	FLASH 擦写模式寄存器	0x000C	32	0x0000 0000
EFM_FSR	FLASH 状态寄存器	0x0010	32	0x0000 0100
EFM_FSCLR	FLASH 状态清除寄存器	0x0014	32	0x0000 0000
EFM_FITE	FLASH 中断许可寄存器	0x0018	32	0x0000 0000
EFM_FPMTSW	FLASH 改写允许区域起始地址寄存器	0x0020	32	0x0000 0000
EFM_FPMTEW	FLASH 改写允许区域结束地址寄存器	0x0024	32	0x0000 0000
FEM_UQID0	FLASHUniqueID 寄存器	0x0050	32	不定
EFM_UQID1	FLASHUniqueID 寄存器	0x0054	32	不定
EFM_UQID2	FLASHUniqueID 寄存器	0x0058	32	不定

7.7.1 访问保护寄存器(EFM_FAPRT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FAPRT[15:0]															

位/位域	标记	位名	功能	读写
b31-b16	Res	保留位	读出时为“0”,写入时写“0”	R
b15-b0	FAPRT[15:0]	EFM 寄存器写保护	访问 EFM 寄存器保护寄存器。 解除方法: 对 FAPRT 先写“16 位数据 0x0123”再写“16 位数据 0x3210”。 在解除保护状态下, 写入任意数据, EFM 寄存器再次进入保护状态。 EFM 寄存器访问保护有效时, 该寄存器读出值为 0x0000 0000。 EFM 寄存器访问保护无效时, 该寄存器读出值为 0x0000 0001。	RW

7.7.2 FLASH 停止寄存器(EFM_FSTP)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															FSTP

位/位域	标记	位名	功能	读写
b31-b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	FSTP	FLASH 停止模式控制	0: FLASH 活动状态 1: FLASH 处于停止模式 当寄存器位由 1 设为 0 后, 程序需要等待 10us 后, 才能进行 FLASH 访问。	RW

7.7.3 读模式寄存器(EFM_FRMC)

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														PREF ETE	Res
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														FLWT[1:0]	

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”,写入时写“0”	R
b17	PREFETE	预取指许可位	0: 关闭预取指功能 1: 预取指功能使能	RW
b16~b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1~b0	FLWT[1:0]	FLASH 读插入的等待周期	00b: 不插入等待周期 01b: 插入 1 个等待周期 10b: 插入 2 个等待周期 11b: 插入 3 个等待周期	RW

7.7.4 擦写模式寄存器(EFM_FWMC)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															ERCALE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							BUSHLCTL	Res	PEMOD[2:0]			Res		PEMODE	

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时为“0”,写入时写“0”	R
b16	ERCALE	FLASH 扇区擦除取消许可位	0: FLASH 扇区擦除取消不许可。 1: FLASH 扇区擦除取消许可。	RW
b15~b9	Res	保留位	读出时为“0”,写入时写“0”	R
b8	BUSHLCTL	FLASH 擦除,编程期间总线控制	0: FLASH 编程擦除期间,总线被占用。 1: FLASH 编程擦除期间,总线释放。	RW
b7	Res	保留位	读出时为“0”,写入时写“0”	R
b6~b4	PEMOD[2:0]	FLASH 擦除,编程模式	000: 只读模式 001: 单编程模式 010: 单编程回读模式 011: 连续编程模式 100: 扇区擦除模式 101: 全擦除模式 110: 只读模式 111: 只读模式 该位只有在 PEMODE=1 时,写入许可。	RW
b3~b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	PEMODE	FLASH 擦除,编程许可模式	0: PEMOD[2:0]改写不许可 1: PEMOD[2:0]改写许可	RW

7.7.5 状态寄存器(EFM_FSR)

复位值: 0x0000 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							RDY	Res	ERCAL	COLERR	OPTEND	PGMISMTCH	Res	PEPRERR	PEWERR

位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”,写入时写“0”	R
b8	RDY	FLASH 忙/空闲状态	0: FLASH 忙状态 1: FLASH 空闲状态	R
b7	Res	保留位	读出时为“0”,写入时写“0”	R
b6	ERCAL	FLASH 扇区擦除取消标志位	0: 未发生 FLASH 扇区擦除取消 1: 发生 FLASH 扇区擦除取消	R
b5	COLERR	FLASH 读写访问错误标志位	0: FLASH 读写访问正常 1: FLASH 读写访问错误 使该位置位的 FLASH 读写访问操作将会被忽视。	R
b4	OPTEND	操作结束标志位	0: 没有对 FLASH 进行擦写或者 FLASH 擦写中 1: 对 FLASH 擦写结束, 置位条件: 编程/擦除/全擦除操作结束	R
b3	PGMISMTCH	单编程回读值不一致标志位	0: 单编程回读值一致 1: 单编程回读值不一致	R
b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1	PEPRERR	对保护地址编程/擦除错误标志位	0: 编程和擦除地址为允许改写区域 1: 对保护窗口内地址进行编程和擦除操作	R
b0	PEWERR	擦写模式错误标志位	0: 在擦写许可模式下擦写 FLASH 1: 在擦写不许可模式下擦写 FLASH	R

7.7.6 状态清除寄存器(EFM_FSCLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									ERCA LCLR	COLE RRCLR	OPTE NDCLR	PGMI SMTCHCLR	Res	PEPR TERRCLR	PEW ERRCLR

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”,写入时写“0”	R
b6	ERCALCLR	清除扇区擦除取消标志位	0: 不发生清除动作 1: 清除扇区擦除取消标志位 读出时, 该位始终为 0。	RW
b5	COLERRCLR	清除读写冲突错误标志位	0: 不发生清除动作 1: 清除读写冲突错误标志位 读出时, 该位始终为 0。	RW
b4	OPTENDCLR	清除操作结束标志	0: 不发生清除动作 1: 清除操作结束标志 读出时, 该位始终为 0。	RW
b3	PGMISMTCHCLR	清除编程回读不一致标志位	0: 不发生清除动作 1: 清除编程回读不一致标志位 读出时, 该位始终为 0。	RW
b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1	PEPRTERRCLR	清除对保护地址编程/擦除错误标志位	0: 不发生清除动作 1: 清除编程/擦除错误标志位 读出时, 该位始终为 0。	RW
b0	PEWERRCLR	擦写模式错误标志位	0: 不发生清除动作 1: 清除模式写错误标志位 读出时, 该位始终为 0。	RW

7.7.7 中断许可寄存器(EFM_FITE)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													COLERRITE	OPTENDITE	PEERRITE

位/位域	标记	位名	功能	读写
b31~b3	Res	保留位	读出时为“0”,写入时写“0”	R
b2	COLERRITE	读写冲突错误中断许可	0: 读写冲突错误中断不许可 1: 读写冲突错误中断许可	RW
b1	OPTENDITE	操作结束中断许可	0: 操作结束中断不许可 1: 操作结束中断许可	RW
b0	PEERRITE	编程/擦除错误中断许可	0: 编程/擦除错误中断不许可 1: 编程/擦除错误中断许可	RW

7.7.8 FLASH 窗口保护起始地址寄存器(EFM_FPMTSW)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														FPMTSW [17: 16]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTSW[15:0]															

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	R
b17-b0	FPMTSW[17: 0]	保护窗口起始地址	FLASH 保护窗口起始地址	RW

7.7.9 FLASH 窗口保护结束地址寄存器(EFM_FPMTEW)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														FPMTEW [17:16]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTEW[15:0]															

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	R
b17-b0	FPMTEW[17: 0]	保护窗口结束地址	FLASH 保护窗口的结束地址	RW

7.7.10 UNIQUE ID 寄存器(EFM_UQID0)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID0[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID0[15:0]															

位/位域	标记	位名	功能	读写
b31-b0	UQID0[31:0]	唯一码	芯片唯一码	R

7.7.11 UNIQUE ID 寄存器(EFM_UQID1)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID1[15:0]															

位/位域	标记	位名	功能	读写
b31-b0	UQID1[31:0]	唯一码	芯片唯一码	R

7.7.12 UNIQUE ID 寄存器(EFM_UQID2)

复位值：不定

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID2[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID2[15:0]															

位/位域	标记	位名	功能	读写
b31-b0	UQID2[31:0]	唯一码	芯片唯一码	R

7.8 注意事项

1. 在 FLASH 进行擦写时，发生复位、擦写操作会强制停止，FLASH 数据将得不到保证。用户需在复位解除后对地址擦除后再次进行操作。
2. 连续编程模式时，FLASH 模拟电路会有高电压状态，长期高压状态会影响 FLASH 特性，一旦连续编程结束请立即退出连续编程模式。禁止在连续编程模式下，MCU 进入低功耗模式（睡眠模式、停止模式）。
3. 设定编程、擦除总线释放(FWMC.BUSHLDCTL=1)时，编程、擦除中如需响应中断，请把中断向量和中断子程序设定到 RAM。
4. 若程序运行在 RAM 上，可通过设定 EFM_FSTP.FSTP=1 来停止 FLASH，以节省功耗。
5. 使用连续编程时，系统时钟必须选择 HRC。
6. Flash 编程、擦除操作结束后，请及时把 EFM_FWMC.PEMOD 修改为只读模式。

8 内置 SRAM (SRAM)

8.1 概述

本产品带有 32KB 系统 SRAM。

SRAM 可按照字节、半字（16 位）或全字（32 位）访问。读写操作以 CPU 速度执行。

SRAM 带有奇偶校验（Even-parity check），每字节数据带有一位校验位。当读取 SRAM 数据发生奇偶校验错误时将发生 SRAM 奇偶校验复位。SRAM 详细定义见表 8-1。

表 8-1 SRAM 空间分配

名称	容量	地址范围	校验方式
SRAM	32KB	0x2000 0000~0x2000 7FFF	Even-parity check

注：

- 读取未经初始化的 SRAM 地址会发生 SRAM 奇偶校验复位。

9 通用 IO (GPIO)

本章中使用的一些略称：

- Px (x=0~7,12,13,14) 表示一组端口，如 P0 表示 P00~P07 这一组的 8 个 I/O 端口
- Pxy (x=0~7,12,13,14,y=0~7,以下同) 表示单个 I/O 端口，如 P10 端口表示 P1 组中第 0 个 I/O, P147 端口表示 P14 组中第 7 个 I/O
- GPIO (General Purpose Input Output) 通用输入输出
- NOD/POD (Nmos/Pmos Open Drain) NMOS/PMOS 开漏输出模式

9.1 概述

主要特性：

- 每组 Port 配有 8 个 I/O Pin，根据实际配置可能不足 8 个
- 支持上拉
- 支持推挽，开漏输出模式
- 支持普通驱动(所有管脚)和高驱动模式
- 支持外部中断的输入
- 支持 I/O pin 外围功能复用，一个 I/O pin 最多可具有 8 个可选的复用功能
- 各个 I/O pin 可独立编程
- 支持 CMOS 和 Schmitt 两种输入方式切换
- 部分寄存器支持 FASTIO 接口，可单周期访问

9.2 端口功能概要

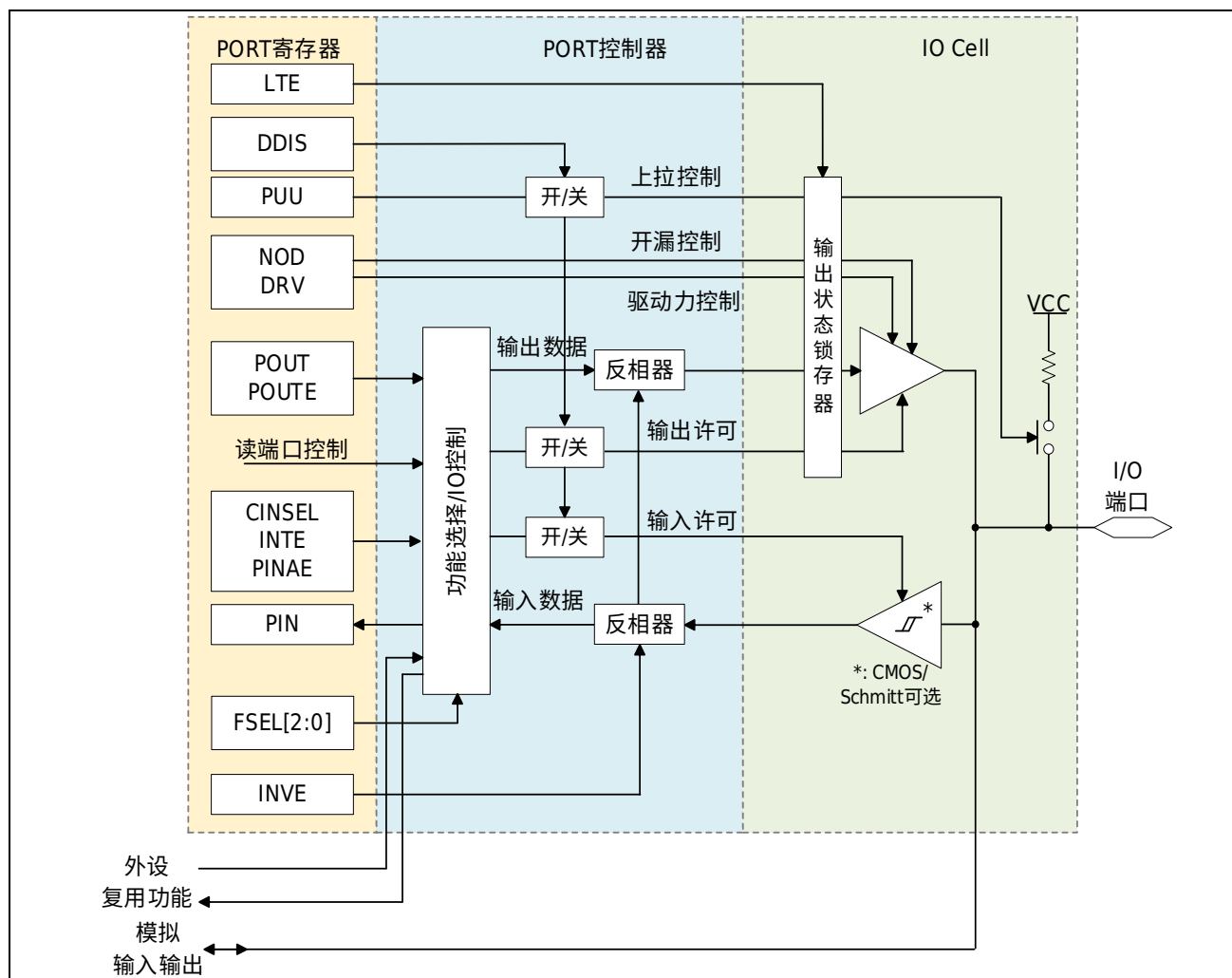


图 9-1 端口基本结构示意图

详细 GPIO 端口数目、驱动能力配置请参考数据手册中引脚配置及功能章节。

9.3 动作说明

9.3.1 通用输入输出 GPIO 功能

通用输入功能 GPI

各 I/O 都具有通用输入 GPI 功能，具有模拟功能的 I/O 在选择模拟功能时，GPI 功能无效。选择模拟功能以外的功能时，GPI 功能始终有效。没有模拟功能的 I/O GPI 功能始终有效。与功能选择寄存器中 PCRxy 的 FSEL[2:0] 设定值无关。通过访问端口输入数据寄存器 PIDRx 可以获取当前端口的状态。也可以通过端口控制寄存器 PCRxy 的 PIN 位查询相应的单 I/O 端口状态，PIDRx.PIN[y] 寄存器位与 PCRxy.PIN 位等价。

默认情况下，为了降低功耗，I/O 的输入 MOS 是被关闭的。只有在读取 PIDRx，PCRxy 寄存器时，才会被打开。根据需要，也可以通过设置寄存器 PINAER.PINAEx 为 1，让 I/O 的输入 MOS 一直处于打开状态。

通用输出功能 GPO

除输入专用的 NRST 端口，其他 I/O 端口都具有通用输出 GPO 功能。通过设置端口功能选择寄存器 PCRxy.FSEL[2:0] 为 0x0 可以有效 GPO 功能。

GPO 功能有效时，可以通过设置通用输出许可寄存器 POERx 来允许或者禁止 I/O 的输出，通用输出数据寄存器 PODRx 来控制的输出值。使用下面的 3 个寄存器也可以控制 I/O 的输出值：输出数据清零寄存器 PORRx，输出数据置位寄存器 POSRx，输出数据翻转寄存器 POTRx。对上述寄存器中相应位写 1 可使对应 I/O 输出 0、1、翻转。写 0 时 I/O 输出状态不改变。

上面的寄存器都是 8 个 PORT 一组一起操作的。为了方便对单个 I/O 的控制，也可通过设置 PCRxy.POUTE 来允许或者禁止 I/O 的输出，PCRxy.POUTE 寄存器位与 POERx.POUTE[y] 等价。可通过设置 PCRxy.POUT 来控制 I/O 的输出值，PCRxy.POUT 寄存器位与 PODRx.POUT[y] 等价。PCRxy 适合用于控制单个 PORT，POERx/PODRx 适合于控制 8 位整组 PORT。

系统复位后，除了 TOOL0 复用端口 P40 以外，其他端口的初始功能均为 GPO，且处于输出禁止状态 (POUTExy=0)。

9.3.2 外围功能

通过功能选择寄存器 PCRxy 的 FSEL[2:0]，每个端口可以配置最多 8 个功能。其中包括 FSEL[2:0]=0x0 对应的通用输出 GPO 功能。各端口具体配置的功能请参考数据手册中引脚功能表。

TOOL0/SWD 调试功能，使用寄存器 PSPCR 选择。PSPCR.SPFE[z] (z=0~1) 为 1 时，对应端口的 PCRxy.FSEL[2:0] 寄存器位无效，即 SPFE 优先级高于 FSEL。PSPCR 寄存器初始值为 0x3，TOOL0/SWD 功能有效。如果要将这些端口设置为 TOOL0/SWD 以外的功能，需要先对对应 SPFE[z] 位写 0。

9.3.3 外部中断 EIRQ 输入功能

部分 I/O 端口具有外部中断输入功能。当 PCRxy.INTE 位设为 1 时，此 I/O 将作为外部中断源 EIRQz 被允许输入。每个 EIRQz 可配置的 I/O 不止一个，使用时每个 EIRQz 请不要同时允许多个 I/O 输入。外部中断输入功能与 PCRxy.FSEL 选择的外围功能（包括 GPIO）可同时有效。

注：

- 没有 EIRQ 功能的 I/O 的 PCRxy 寄存器没有 INTE 位。
- I/O 端口作为外部中断 EIRQ 使用时，需要结合中断控制器 INTC，设置滤波、中断触发沿、中断号等。详细请参考【10 中断控制器（INTC）】。

9.3.4 模拟功能

部分 I/O 端口带有模拟输入输出功能(包括主副振荡器)。在用作模拟功能时，请将寄存器 PCRxy.DDIS 写 1，禁止当前端口的数字功能。

9.3.5 通用控制

1. 上拉电阻

各 I/O 端口带有内部的上拉电阻。可以设置寄存器 PCRxy.PUU 位允许此功能，在 I/O 端口无输入时，内部处于弱上拉状态。

在 I/O 端口处于输出状态时，上拉功能将自动无效。

2. 驱动能力控制

各 I/O 端口都具有普通驱动，高驱动 2 档驱动能力可调，可根据需要设置寄存器 PCRxy.DRV。

本功能只在端口处于输出状态时才有效。

3. 开漏输出模式

设置 PCRxy.NOD 位，可以将 I/O 端口设置成 NMOS 开漏输出模式。当 NOD 有效时，对应端口可正常输出 0，而输出 1 时端口将处于高阻态。

以上所述通用控制功能，如无特别说明，它们与端口具体选择的功能即 FSEL[2:0]的设置无关。

9.4 寄存器描述

表 9-1 GPIO 基地址

名称	基地址	描述
GPIO	0x4001 3800	GPIO基地址

表 9-2 GPIO 寄存器列表

寄存器	描述	偏移地址	位宽	复位值	访问接口
PIDRx	通用输入数据寄存器	0x00+x	8	0xXX	FASTIO
PODRx	通用输出数据寄存器	0x10+x	8	0x00	FASTIO
POERx	通用输出许可寄存器	0x20+x	8	0x00	FASTIO
POSRx	通用输出置位寄存器	0x30+x	8	0x00	FASTIO
PORRx	通用输出复位寄存器	0x40+x	8	0x00	FASTIO
POTRx	通用输出翻转寄存器	0x50+x	8	0x00	FASTIO
PCRxy	通用控制寄存器	0x400+0x10*x+0x2*y	16	0x0X00	AHB
PSPCR	特殊控制寄存器	0x500	16	0x0003	AHB
PCCR	公共控制寄存器	0x504	16	0x4000	AHB
PINAER	输入控制寄存器	0x506	16	0x0000	AHB
PWPR	写保护寄存器	0x508	16	0x0000	AHB

注：

- 地址计算公式中 $x=0\sim7,12,13,14$ $y=0\sim7$

9.4.1 通用输入寄存器 (PIDRx)

复位值: 0xXX

b7	b6	b5	b4	b3	b2	b1	b0
PIN[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	PIN[7:0]	输入状态	0: I/O 端口输入状态为低电平 1: I/O 端口输入状态为高电平	R

本寄存器为只读寄存器, 写无效。在数字功能未被禁止时, 通过读取此寄存器可以获取端口的输入状态, 与功能选择寄存器的 PCRxy.FSEL[2:0]设定值无关。不存在端口对应位的读出值不定。在端口的数字功能禁止状态时, 由于 I/O 输入 MOS 处于关闭状态, 对应 PIN 位读出值为固定值 0x1。

9.4.2 通用输出数据寄存器 (PODRx)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
POUT[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	POUT[7:0]	输出数据	0: 输出低电平 1: 输出高电平	RW

当 I/O 端口被设置成 GPO 功能时, 改写此寄存器可以更改对应端口的输出状态。

9.4.3 通用输出许可寄存器 (POERx)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
POUTE[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	POUTE[7:0]	输出许可	0: 输出禁止 1: 输出许可	RW

当 I/O 端口被设置成 GPO 功能时, 且此寄存器设为 1 时, PODRx 设定值将输出到对应 I/O 端口。此寄存器设为 0 时, 输出关闭, 端口为高阻态。不存在端口对应位请不要写 1。

9.4.4 通用输出置位寄存器 (POSRx)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
POS[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	POS[7:0]	输出高	0: 对应 PODRx.POUT 无变化 1: 对应 PODRx.POUT 置 1	RW

注:

- 此寄存器的读出值始终为 0x00。

9.4.5 通用输出复位寄存器 (PORRx)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
POR[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	POR[7:0]	输出低	0: 对应 PODRx.POUT 无变化 1: 对应 PODRx.POUT 清零	RW

注:

- 此寄存器的读出值始终为 0x00。

9.4.6 通用输出翻转寄存器 (POTRx)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
POT[7:0]							

位/位域	标记	位名	功能	读写
b7~b0	POT[7:0]	输出翻转	0: 对应 PODRx.POUT 无变化 1: 对应 PODRx.POUT 取反	RW

注:

- 此寄存器的读出值始终为 0x00。

9.4.7 通用控制寄存器 (PCRxy)

复位值: 0x0X00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DDIS	FSEL[2:0]			INTE	CINSEL	INVE	PIN	Res	PUU	LTE	DRV	Res	NOD	POUTE	POUT

位/位域	标记	位名	功能	读写
b15	DDIS	数字功能禁止	0: 数字功能有效 1: 数字功能禁止	R/W
b14~b12	FSEL[2:0]	功能选择	各端口的功能配置请参考数据手册中引脚功能表	RW
b11	INTE	外部中断许可	0: 外部中断输入禁止 1: 外部中断输入许可 无外部中断功能的 I/O 此位为保留位, 读出时为 0, 写时请写 0	RW
b10	CINSEL	CMOS/Schmitt 输入选择	0: 选择 Schmitt 输入 1: 选择 CMOS 输入	RW
b9	INVE	反相许可	0: 输入输出数据不反相 1: 输入输出数据反相	RW
b8	PIN	输入状态	0: I/O 端口输入状态为低电平 1: I/O 端口输入状态为高电平 与寄存器 PIDRx 中 PIN[y]功能一致	R
b7	Res	保留位	读出时为 0, 写时请写 0	R
b6	PUU	上拉许可	0: 内部上拉(pullup)电阻无效 1: 内部上拉(pullup)电阻有效	RW
b5	LTE	输出状态锁存	0: 输出锁存无效 1: 输出锁存有效	RW
b4	DRV ^(注)	驱动模式选择	0: 普通驱动力模式 1: 高驱动力模式	RW
b3	Res	保留位	读出时为 0, 写时请写 0	R
b2	NOD	NMOS 开漏	0: 正常 CMOS 输出模式 1: NMOS 开漏输出	RW
b1	POUTE	输出许可	0: 输出禁止 1: 输出许可 与寄存器 POERx 中 POUTE[y]功能一致	RW
b0	POUT	输出数据	0: 输出低电平 1: 输出高电平 与寄存器 PODRx 中 POUT[y]功能一致	RW

各 I/O 端口可以通过 FSEL[2:0]选择配置在该端口上多个功能中的一个。参考数据手册中引脚功能表, FSEL[2:0]设为 b000 表示选择 Func0, 设为 b001 表示选择 Func1, 以此类推。其中 Func0 对应的通用输出功能 GPO。

注:

- P40 端口复位后初始状态为 TOOL0 功能有效, 在配置 FSEL[2:0]选择功能时需要先将寄存器 PSPCR 相应位写 0 无效 TOOL0 功能。

具有模拟功能的 I/O 在选择模拟功能时，对应端口的所有数字功能全部强制无效，包括通用输入输出，外围数字输入输出，上拉/下拉功能，以及外部中断输入功能。

DDIS 设为 1 时，对应端口的所有数字功能全部强制无效，包括通用输入输出，周边的数字输入输出，上拉功能，以及外部中断输入功能。当端口作为模拟输入时，请将 DDIS 位设为 1。

LTE 设为 1 输出锁存有效时，端口当前的输出状态保持，直至 LTE 写成 0。本功能主要在端口功能切换时使用。为避免功能切换时端口输出意想之外的毛刺而导致系统误动作，在功能切换前，先将 LTE 写 1 锁存住端口的输出状态，再改写寄存器选择寄存器切换功能，最后再将 LTE 写 0 解除锁存，端口状态更新为新功能。

INVE 设为 1 时，端口的输入输出数据都会进行反相，包含 GPIO 功能，和其它外围输入输出功能。

CINSEL 用于选择对应 I/O 的输入方式，该 I/O 的所有输入功能将使用 CMOS/Schmitt 输入，包括 GPI 和外围数字功能。

9.4.8 特殊控制寄存器 (PSPCR)

复位值: 0x0003

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														SPFE[1:0]	

位/位域	标记	位名	功能	读写
b15~b2	Res	保留位	读出时为 0，写时请写 0	R
b1	SPFE[1]	特殊功能选择	0: TOOL0 功能无效 1: TOOL0 功能有效	RW
b0	SPFE[0]	特殊功能选择	0: SWCLK 和 SWDIO 功能无效 1: SWCLK 和 SWDIO 功能有效	RW

注:

- SPFE 功能选择位优先级高于 PCRxy.FSEL[2:0]功能选择位。

9.4.9 公共控制寄存器 (PCCR)

复位值: 0x4000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RDWT[1:0]		Res													

位/位域	标记	位名	功能			读写
设置读寄存器 PCRxy 时插入的等待周期数						
			设定值	等待周期	推荐工作频率	RW
b15-b14	RDWT[1:0]	读端口等待	00	无等待	32MHz 以下	
			01（默认值）	1 周期	32MHz	
			10	2 周期	32MHz	
			11	3 周期	32MHz	
b13~b0	Res	保留位	读出时为 0，写时请写 0			R

9.4.10 输入控制寄存器 (PINAER)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PINAER[15:0]															

位/位域	标记	位名	功能	读写
			0: 输入 MOS 常开无效 1: 输入 MOS 常开有效	
			PINAER[0]控制 P00~P06, PINAER[1]控制 P10~P17, PINAER[2]控制 P20~P27, PINAER[3]控制 P30~P31, PINAER[4]控制 P40~P43, PINAER[5]控制 P50~P55 PINAER[6]控制 P60~P63 PINAER[7]控制 P70~P77 PINAER[12]控制 P120~P124 PINAER[13]控制 P130, P137 PINAER[14]控制 P140~P141, P146~P147	
b15~b0	PINAER[15:0]	输入常开		RW

9.4.11 写保护寄存器 (PWPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WP[7:0]								Res							WE

位/位域	标记	位名	功能	读写
b15~b8	WP[7:0]	写保护码	读出时为 0x00 当 b15~b8 写入值为 0xA5 时, b0 值写入 WE 当写 0xA5 以外值时, WE 自动清零	W
b7~b1	Res	保留位	读出时为 0, 写时请写 0	R
b0	WE	写许可	0: PSPCR, PCCR, PINAER, PCRxy 寄存器写禁止 1: PSPCR, PCCR, PINAER, PCRxy 寄存器写许可	RW

9.4.12 访问位宽

上面寄存器中, 通用寄存器包括 PIDRx, PODRx, POERx, POSRx, PORRx, POTRx 支持 8 位、16 位、32 位访问。PSPCR, PCCR, PINAER, PCRxy 支持 16 位、32 位访问。PWPR 仅支持 16 位访问。

9.5 注意事项

请不要将同一功能设置到多个端口上。

请在输出锁存有效时 (LTE=1) 进行端口功能切换, 以避免切换期间端口上输出期待之外毛刺。

10 中断控制器（INTC）

10.1 概述

中断控制器（INTC）的功能有选择系统或外设事件请求作为中断输入到 NVIC，唤醒 WFI；选择系统或外设事件请求作为事件输入到 NVIC，唤醒 WFE；选择系统或外设事件请求作为低功耗模式（休眠模式和停止模式）的唤醒条件；外部管脚 INTP、KR 的中断或事件控制功能。

主要规格：

- 1) NVIC 中断向量：实际使用中断向量数请参考【中断向量表】（不包括 Cortex™-M0+ 的 16 根中断线），每个中断向量可以根据中断选择寄存器选择对应的外设中断事件请求。更多关于异常和 NVIC 编程的说明，请参考《ARM Cortex™-M0+ 技术参考手册》中的第 5 章：嵌套向量中断控制器。
- 2) 可编程优先级：4 个可编程优先级。
- 3) 不可屏蔽中断：独立选择多种系统中断事件请求作为不可屏蔽中断，且各中断事件请求配备独立的使能选择、标志、标志清除寄存器。
- 4) 配备 12 个外部 EIRQ 管脚（INTP）中断。
- 5) 配备 8 个外部 EKEY 管脚（KR）事件。
- 6) 配置多种外设中断事件请求，具体请参考【中断事件请求序号】。
- 7) 中断可唤醒系统休眠模式和停止模式。

表 10-1 输入管脚

管脚名	I/O	说明
INTP0	输入	外部 EIRQ 中断请求管脚 0
INTP1	输入	外部 EIRQ 中断请求管脚 1
INTP2	输入	外部 EIRQ 中断请求管脚 2
INTP3	输入	外部 EIRQ 中断请求管脚 3
INTP4	输入	外部 EIRQ 中断请求管脚 4
INTP5	输入	外部 EIRQ 中断请求管脚 5
INTP6	输入	外部 EIRQ 中断请求管脚 6
INTP7	输入	外部 EIRQ 中断请求管脚 7
INTP8	输入	外部 EIRQ 中断请求管脚 8
INTP9	输入	外部 EIRQ 中断请求管脚 9
INTP10	输入	外部 EIRQ 中断请求管脚 10
INTP11	输入	外部 EIRQ 中断请求管脚 11
KR0	输入	外部 EKEY 事件请求管脚 0
KR1	输入	外部 EKEY 事件请求管脚 1

管脚名	I/O	说明
KR2	输入	外部 EKEY 事件请求管脚 2
KR3	输入	外部 EKEY 事件请求管脚 3
KR4	输入	外部 EKEY 事件请求管脚 4
KR5	输入	外部 EKEY 事件请求管脚 5
KR6	输入	外部 EKEY 事件请求管脚 6
KR7	输入	外部 EKEY 事件请求管脚 7

10.2INTC 系统框图

中断控制器（INTC）的系统框图如下：

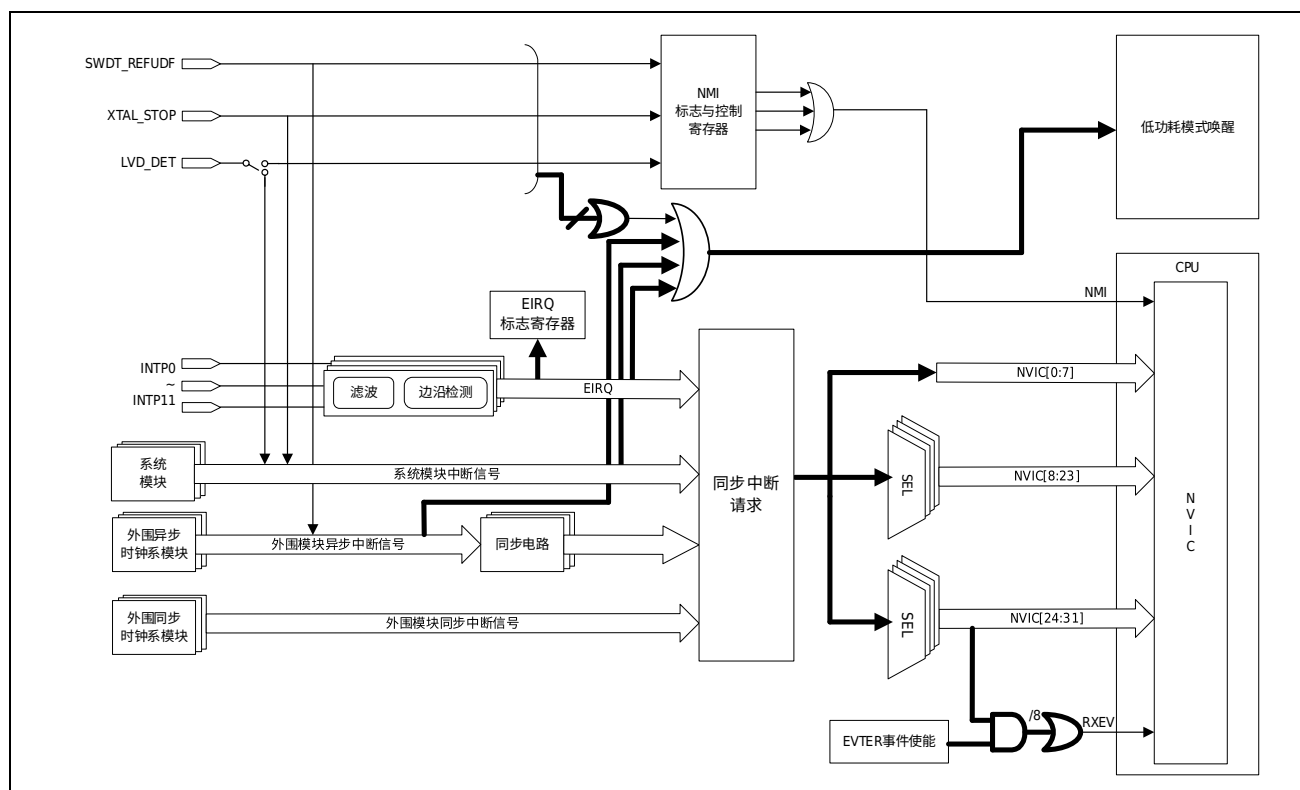


图 10-1 中断系统框图

10.3 向量表

10.3.1 中断向量表

表 10-2 中断向量表

地址	向量 序号	IRQ 序号	中断源	说明
ARM 核中断处理向量				
0x0000 0000	0	-	ARM core	Initial stack pointer
0x0000 0004	1	-	ARM core	Initial Program Counter
0x0000 0008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000 000C	3	-	ARM core	Hard Fault
0x0000 0010	4	-	ARM core	Reserved
0x0000 0014	5	-	ARM core	Reserved
0x0000 0018	6	-	ARM core	Reserved
0x0000 001C	7	-	ARM core	Reserved
0x0000 0020	8	-	ARM core	Reserved
0x0000 0024	9	-	ARM core	Reserved
0x0000 0028	10	-	ARM core	Reserved
0x0000 002C	11	-	ARM core	Supervisor call (SVCall)
0x0000 0030	12	-	ARM core	Reserved
0x0000 0034	13	-	ARM core	Reserved
0x0000 0038	14	-	ARM core	Pendable request for system service (PendableSrvReq)
0x0000 003C	15	-	ARM core	System tick timer (SysTick)
非 ARM 核中断处理向量				
0x0000 0040	16	0	-	外部 EIRQ 中断请求管脚 0 (INTP0) 的中断请求
0x0000 0044	17	1	-	外部 EIRQ 中断请求管脚 1 (INTP1) 的中断请求
0x0000 0048	18	2	-	外部 EIRQ 中断请求管脚 2 (INTP2) 的中断请求
0x0000 004C	19	3	-	外部 EIRQ 中断请求管脚 3 (INTP3) 的中断请求
0x0000 0050	20	4	-	外部 EIRQ 中断请求管脚 4 (INTP4) 的中断请求
0x0000 0054	21	5	-	外部 EIRQ 中断请求管脚 5 (INTP5) 的中断请求
0x0000 0058	22	6	-	外部 EIRQ 中断请求管脚 6 (INTP6) 的中断请求
0x0000 005C	23	7	-	外部 EIRQ 中断请求管脚 7 (INTP7) 的中断请求
0x0000 0060	24	8	INTC_ISELAR8	寄存器 INTC_ISELAR8 选择的中断请求
0x0000 0064	25	9	INTC_ISELAR9	寄存器 INTC_ISELAR9 选择的中断请求
0x0000 0068	26	10	INTC_ISELAR10	寄存器 INTC_ISELAR10 选择的中断请求
0x0000 006C	27	11	INTC_ISELAR11	寄存器 INTC_ISELAR11 选择的中断请求
0x0000 0070	28	12	INTC_ISELAR12	寄存器 INTC_ISELAR12 选择的中断请求
0x0000 0074	29	13	INTC_ISELAR13	寄存器 INTC_ISELAR13 选择的中断请求

地址	向量 序号	IRQ 序号	中断源	说明
0x0000 0078	30	14	INTC_ISELAR14	寄存器 INTC_ISELAR14 选择的中断请求
0x0000 007C	31	15	INTC_ISELAR15	寄存器 INTC_ISELAR15 选择的中断请求
0x0000 0080	32	16	INTC_ISELAR16	寄存器 INTC_ISELAR16 选择的中断请求
0x0000 0084	33	17	INTC_ISELAR17	寄存器 INTC_ISELAR17 选择的中断请求
0x0000 0088	34	18	INTC_ISELAR18	寄存器 INTC_ISELAR18 选择的中断请求
0x0000 008C	35	19	INTC_ISELAR19	寄存器 INTC_ISELAR19 选择的中断请求
0x0000 0090	36	20	INTC_ISELAR20	寄存器 INTC_ISELAR20 选择的中断请求
0x0000 0094	37	21	INTC_ISELAR21	寄存器 INTC_ISELAR21 选择的中断请求
0x0000 0098	38	22	INTC_ISELAR22	寄存器 INTC_ISELAR22 选择的中断请求
0x0000 009C	39	23	INTC_ISELAR23	寄存器 INTC_ISELAR23 选择的中断请求
0x0000 00A0	40	24	INTC_ISELBR24	寄存器 INTC_ISELBR24 选择的中断请求
0x0000 00A4	41	25	INTC_ISELBR25	寄存器 INTC_ISELBR25 选择的中断请求
0x0000 00A8	42	26	INTC_ISELBR26	寄存器 INTC_ISELBR26 选择的中断请求
0x0000 00AC	43	27	INTC_ISELBR27	寄存器 INTC_ISELBR27 选择的中断请求
0x0000 00B0	44	28	INTC_ISELBR28	寄存器 INTC_ISELBR28 选择的中断请求
0x0000 00B4	45	29	INTC_ISELBR29	寄存器 INTC_ISELBR29 选择的中断请求
0x0000 00B8	46	30	INTC_ISELBR30	寄存器 INTC_ISELBR30 选择的中断请求
0x0000 00BC	47	31	INTC_ISELBR31	寄存器 INTC_ISELBR31 选择的中断请求

注：

- $INTC\_ISELAR_m$ ($m=8\sim 23$) 具体选择的中断事件请求编号请参考寄存器描述章节。

10.3.2 中断事件请求序号

中断事件请求由系统或外设产生，中断事件请求被中断控制器选择为中断输入时，中断事件请求被称为中断源；被选择为事件输入时，被称为事件源。外设中断事件请求也可以作为 MCU 低功耗模式返回的条件。

表 10-3 中断事件请求序号与选择

编号	请求 序号	功能 模块	功能 名称	中断 源	内 部 事件 源	分组 编号	组内 编号	对应 NVIC 向量的中断选择寄存器 ^[注 1]		
								NVIC 向量 0~7	NVIC 向量 8~23	NVIC 向量 24~31
0	00h	PORT	PORT_EIRQ0	✓	✓	Group0	0	Vec0	-	Vec24
1	01h	PORT	PORT_EIRQ8	✓	✓		1	-	Vec8~9	
2	02h	DMA	DMA_TC0	✓	✓		2	-	Vec8~9	
3	03h	DMA	DMA BTC0	✓	✓		3	-	Vec8~9	
4	04h	EFM	EFM_PEERR	✓	-		4	-	Vec8~9	
5	05h	EFM	EFM_COLERR	✓	-		5	-	Vec8~9	
6	06h	XTAL	XTAL_STOP	✓	-		6	-	Vec8~9	

编号	请求序号	功能模块	功能名称	中断源	内部事件源	分组编号	组内编号	对应 NVIC 向量的中断选择寄存器 ^[注 1]		
								NVIC 向量 0~7	NVIC 向量 8~23	NVIC 向量 24~31
7	07h	SWDT	SWDT_REFUDF	✓	-		7	-	Vec8~9	
8	08h	TMRB	TMRB_7_OVF	✓	✓		8	-	Vec8~9	
9	09h	TMRB	TMRB_7_UDF	✓	✓		9	-	Vec8~9	
10	0Ah	TMRB	TMRB_8_CMP	✓	✓		A	-	Vec8~9	
11	0Bh	USART	USART_5_EI	✓	-		B	-	Vec8~9	
12	0Ch	USART	USART_1_EI	✓	-		C	-	Vec8~9	
13	0Dh	I2C	I2C_EEI	✓	-		D	-	Vec8~9	
14	0Eh	SPI	SPI_SPEI	✓	-		E	-	Vec8~9	
15	0Fh	USART	USART_4_EI	✓	-		F	-	Vec8~9	
16	10h	PORT	PORT_EIRQ1	✓	✓	Group1	0	Vec1	-	-
17	11h	PORT	PORT_EIRQ9	✓	✓		1	-	Vec10~11	Vec25
18	12h	DMA	DMA_TC1	✓	✓		2	-	Vec10~11	
19	13h	DMA	DMA_BTC1	✓	✓		3	-	Vec10~11	
20	14h	CTC	CTC_ERR	✓	-		4	-	Vec10~11	
21	15h	-	-	-	-		5	-	Vec10~11	
22	16h	-	-	-	-		6	-	Vec10~11	
23	17h	-	-	-	-		7	-	Vec10~11	
24	18h	TMRB	TMRB_8_OVF	✓	✓		8	-	Vec10~11	
25	19h	TMRB	TMRB_8_UDF	✓	✓		9	-	Vec10~11	
26	1Ah	TMRB	TMRB_7_CMP	✓	✓		A	-	Vec10~11	
27	1Bh	USART	USART_6_EI	✓	-		B	-	Vec10~11	
28	1Ch	USART	USART_1_RI	✓	✓		C	-	Vec10~11	
29	1Dh	-	-	-	-		D	-	Vec10~11	
30	1Eh	-	-	-	-		E	-	Vec10~11	
31	1Fh	USART	USART_4_RI	✓	✓		F	-	Vec10~11	
32	20h	PORT	PORT_EIRQ2	✓	✓	Group2	0	Vec2	-	-
33	21h	DMA	DMA_ERR	✓	-		1	-	Vec12~13	Vec26
34	22h	-	-	-	-		2	-	Vec12~13	
35	23h	-	-	-	-		3	-	Vec12~13	
36	24h	-	-	-	-		4	-	Vec12~13	
37	25h	-	-	-	-		5	-	Vec12~13	
38	26h	-	-	-	-		6	-	Vec12~13	
39	27h	-	-	-	-		7	-	Vec12~13	
40	28h	TMRB	TMRB_5_OVF	✓	✓		8	-	Vec12~13	
41	29h	TMRB	TMRB_5_UDF	✓	✓		9	-	Vec12~13	
42	2Ah	TMRB	TMRB_6_CMP	✓	✓		A	-	Vec12~13	

编号	请求序号	功能模块	功能名称	中断源	内部事件源	分组编号	组内编号	对应 NVIC 向量的中断选择寄存器 ^[注 1]		
								NVIC 向量 0~7	NVIC 向量 8~23	NVIC 向量 24~31
43	2Bh	USART	USART_5_RI	✓	✓		B	-	Vec12~13	
44	2Ch	USART	USART_1_TI	✓	✓		C	-	Vec12~13	
45	2Dh	USART	USART_3_EI	✓	-		D	-	Vec12~13	
46	2Eh	-	-	-	-		E	-	Vec12~13	
47	2Fh	-	-	-	-		F	-	Vec12~13	
48	30h	PORT	PORT_EIRQ3	✓	✓	Group3	0	Vec3	-	-
49	31h	KEY	PORT_EKEY	✓	-		1	-	Vec14~15	Vec27
50	32h	TMR0	TMR0_GCMP	✓	✓		2	-	Vec14~15	
51	33h	USART	USART_6_RI	✓	✓		3	-	Vec14~15	
52	34h	-	-	-	-		4	-	Vec14~15	
53	35h	-	-	-	-		5	-	Vec14~15	
54	36h	-	-	-	-		6	-	Vec14~15	
55	37h	-	-	-	-		7	-	Vec14~15	
56	38h	TMRB	TMRB_6_OVF	✓	✓		8	-	Vec14~15	
57	39h	TMRB	TMRB_6_UDF	✓	✓		9	-	Vec14~15	
58	3Ah	TMRB	TMRB_5_CMP	✓	✓		A	-	Vec14~15	
59	3Bh	EVENT	EVENT_STRG	-	✓		B	-	Vec14~15	
60	3Ch	USART	USART_1_TCI	✓	-		C	-	Vec14~15	
61	3Dh	USART	USART_3_RI	✓	✓		D	-	Vec14~15	
62	3Eh	SPI	SPI_SPRI	✓	✓		E	-	Vec14~15	
63	3Fh	-	-	-	-		F	-	Vec14~15	
64	40h	PORT	PORT_EIRQ4	✓	✓	Group4	0	Vec4	-	-
65	41h	-	-	-	-		1	-	Vec16~17	Vec28
66	42h	-	-	-	-		2	-	Vec16~17	
67	43h	-	-	-	-		3	-	Vec16~17	
68	44h	-	-	-	-		4	-	Vec16~17	
69	45h	-	-	-	-		5	-	Vec16~17	
70	46h	-	-	-	-		6	-	Vec16~17	
71	47h	USART	USART_5_TI	✓	✓		7	-	Vec16~17	
72	48h	TMRB	TMRB_3_OVF	✓	✓		8	-	Vec16~17	
73	49h	TMRB	TMRB_3_UDF	✓	✓		9	-	Vec16~17	
74	4Ah	TMRB	TMRB_4_CMP	✓	✓		A	-	Vec16~17	
75	4Bh	ADC	ADC_EOCA	✓	✓		B	-	Vec16~17	
76	4Ch	USART	USART_2_EI	✓	-		C	-	Vec16~17	
77	4Dh	USART	USART_3_TI	✓	✓		D	-	Vec16~17	
78	4Eh	I2C	I2C_TEI	✓	-		E	-	Vec16~17	

编号	请求序号	功能模块	功能名称	中断源	内部事件源	分组编号	组内编号	对应 NVIC 向量的中断选择寄存器 ^[注 1]		
								NVIC 向量 0~7	NVIC 向量 8~23	NVIC 向量 24~31
79	4Fh	SPI	SPI_SPII	✓	-		F	-	Vec16~17	
80	50h	PORT	PORT_EIRQ5	✓	✓	Group5	0	Vec5	-	-
81	51h	-	-	-	-		1	-	Vec18~19	Vec29
82	52h	-	-	-	-		2	-	Vec18~19	
83	53h	-	-	-	-		3	-	Vec18~19	
84	54h	-	-	-	-		4	-	Vec18~19	
85	55h	PORT	PORT_EIRQ10	✓	✓		5	-	Vec18~19	
86	56h	-	-	-	-		6	-	Vec18~19	
87	57h	USART	USART_6_TI	✓	✓		7	-	Vec18~19	
88	58h	TMRB	TMRB_4_OVF	✓	✓		8	-	Vec18~19	
89	59h	TMRB	TMRB_4_UDF	✓	✓		9	-	Vec18~19	
90	5Ah	TMRB	TMRB_3_CMP	✓	✓		A	-	Vec18~19	
91	5Bh	ADC	ADC_EOCB	✓	✓		B	-	Vec18~19	
92	5Ch	USART	USART_2_RI	✓	✓		C	-	Vec18~19	
93	5Dh	USART	USART_3_TCI	✓	-		D	-	Vec18~19	
94	5Eh	I2C	I2C_RXI	✓	✓		E	-	Vec18~19	
95	5Fh	-	-	-	-		F	-	Vec18~19	
96	60h	PORT	PORT_EIRQ6	✓	✓	Group6	0	Vec6	-	-
97	61h	-	-	-	-		1	-	Vec20~21	Vec30
98	62h	-	-	-	-		2	-	Vec20~21	
99	63h	-	-	-	-		3	-	Vec20~21	
100	64h	-	-	-	-		4	-	Vec20~21	
101	65h	PORT	PORT_EIRQ11	✓	✓		5	-	Vec20~21	
102	66h	-	-	-	-		6	-	Vec20~21	
103	67h	USART	USART_5_TCI	✓	-		7	-	Vec20~21	
104	68h	TMRB	TMRB_1_OVF	✓	✓		8	-	Vec20~21	
105	69h	TMRB	TMRB_1_UDF	✓	✓		9	-	Vec20~21	
106	6Ah	TMRB	TMRB_2_CMP	✓	✓		A	-	Vec20~21	
107	6Bh	ADC	ADC_CMP0	✓	✓		B	-	Vec20~21	
108	6Ch	USART	USART_2_TI	✓	✓		C	-	Vec20~21	
109	6Dh	I2C	I2C_TXI	✓	✓		D	-	Vec20~21	
110	6Eh	-	-	-	-		E	-	Vec20~21	
111	6Fh	USART	USART_4_TI	✓	✓		F	-	Vec20~21	
112	70h	PORT	PORT_EIRQ7	✓	✓	Group7	0	Vec7	-	-
113	71h	-	-	-	-		1	-	Vec22~23	Vec31
114	72h	LVD	LVD_DET	✓	-		2	-	Vec22~23	

编号	请求序号	功能模块	功能名称	中断源	内部事件源	分组编号	组内编号	对应 NVIC 向量的中断选择寄存器 ^[注 1]		
								NVIC 向量 0~7	NVIC 向量 8~23	NVIC 向量 24~31
115	73h	USART	USART_6_TCI	✓	-		3	-	Vec22~23	
116	74h	EFM	EFM_OPTEND	✓	✓		4	-	Vec22~23	
117	75h	RTC	RTC_ALM	✓	✓		5	-	Vec22~23	
118	76h	RTC	RTC_PRD	✓	✓		6	-	Vec22~23	
119	77h	RTC	RTC_IT	✓	-		7	-	Vec22~23	
120	78h	TMRB	TMRB_2_OVF	✓	✓		8	-	Vec22~23	
121	79h	TMRB	TMRB_2_UDF	✓	✓		9	-	Vec22~23	
122	7Ah	TMRB	TMRB_1_CMP	✓	✓		A	-	Vec22~23	
123	7Bh	ADC	ADC_CMP1	✓	✓		B	-	Vec22~23	
124	7Ch	USART	USART_2_TCI	✓	-		C	-	Vec22~23	
125	7Dh	SPI	SPI_SPTI	✓	✓		D	-	Vec22~23	
126	7Eh	-	-	-	-		E	-	Vec22~23	
127	7Fh	USART	USART_4_TCI	✓	-		F	-	Vec22~23	

***1：** 中断向量 Vec8~31 中对于中断源的选择，请参考寄存器章节 *INTC_ISELAR8~23*、*INTC_ISELBR24~31* 的相关说明。

10.4 功能说明

10.4.1 不可屏蔽中断

不可屏蔽中断源如下：

- 检测主发振器停止中断
- SWDT 下溢/刷新错误中断
- LVD 低电压检测中断

不可屏蔽中断具有最高的优先级。由于不可屏蔽中断可选择多个中断事件请求，各个中断事件请求状态可以通过查询标志寄存器（INTC_NMIFR）来确定。请在不可屏蔽中断处理退出前确认所有的标志位都为“0”。

使用寄存器设定时，请按照下面流程设定：

使用不可屏蔽中断事件请求时，请配置相应的功能。

对 NMICLR 各寄存器位写“1”，清除 NMIFR 标志寄存器位，防止误动作。

通过设定 NMIER 选择寄存器来使能不可屏蔽中断事件。

注：

- 一旦 NMIER 相应位被设定为“1”后，将不能被更改，除非用 NRST 来复位。

10.4.2 EIRQ 管脚中断

需要使用外部 EIRQ 管脚（INTP）中断请求时，请按照如下流程设定：

- 1) 清除 EIRQCRm.EIRQFEN 位（m=0~11），禁止数字滤波器。
- 2) 设定 EIRQCRm.EIRQTRG 位，选择触发边沿或电平；设定 EIRQCRm.EIRQFCLK 位，选择数字滤波器采样时钟；设定 EIRQCRm.EIRQFEN 位，使能数字滤波器（m=0~11）。

10.4.3 EKEY 管脚事件

外部 EKEY 管脚（KR）在输入低电平时，产生一个输入事件。该事件（在中断事件请求序号表中编号为 49）通过中断选择寄存器（INTC_ISELA/BR）的选通送至 CPU，产生对应的中断请求。

各个外部 EKEY 管脚（KR）均含有一个输入使能控制位，当 EKEY 管脚事件控制寄存器（INTC_EKEYCR）的对应位置 1 时，EKEY 管脚的输入有效。其具体框图如下：

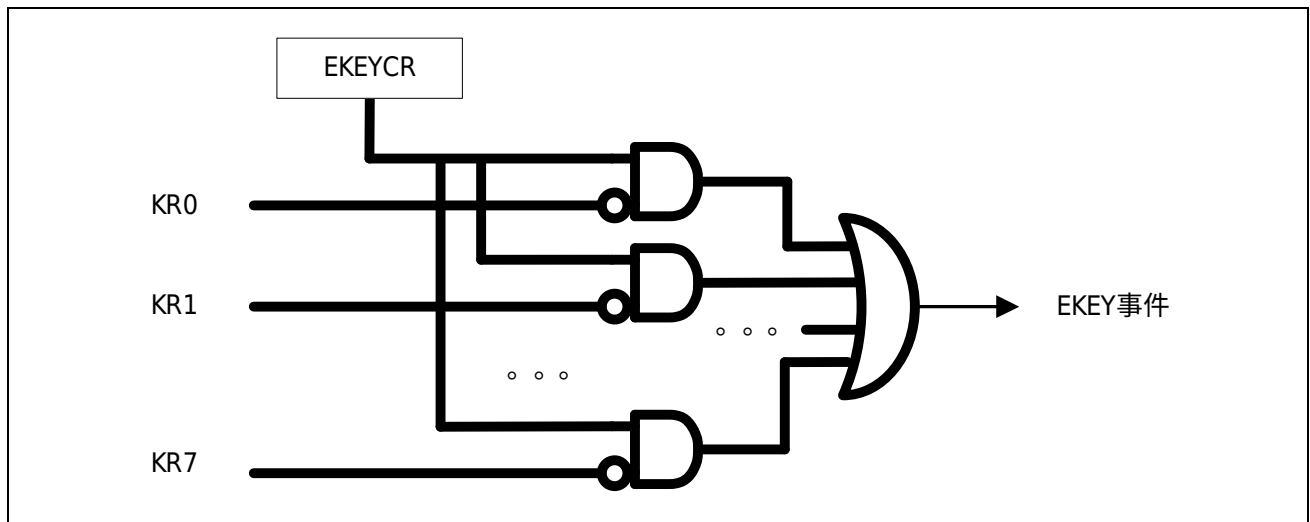


图 10-2 EKEY 事件产生

在使用 EKEY 输入功能时，遵循下面的设定顺序：

- 1) 设定对应管脚的 PCR.PSEL 位，选择该管脚为 EKEY 输入功能
- 2) 设定中断或内部事件源选择相关寄存器，具体参考 ARM 手册 NVIC 相关章节或本手册其它相关模块（Timer、ADC 等）
- 3) 设定 INTC_EKEYCR 的对应使能位

10.4.4 软件触发事件

在中断事件请求编号中编号为 59 的 EVENT_STRG 事件，可通过置位软件触发事件控制寄存器（INTC_STRGCR）的 STRG 位来触发。该事件可用于触发其它的外围模块，如 ADC、Timer 等，具体请参考相关章节。

注：

- 该寄存器不受功能保护寄存器（INTC_FPRCR）的保护控制，且使用软件触发事件功能时，需要先使能外围电路触发功能，使能方法：清零功能时钟控制寄存器位 FCG.AOS。

10.4.5 中断源选择

本中断控制器提供了 3 种中断选择方式，通过灵活的组合来满足各种中断配置需求。

第一种方式

外部 EIRQ 中断请求与中断向量一一对应，占用 Vec0~7 共计 8 个中断向量。

第二种方式

15 选 1 作为中断源，通过中断选择 A 寄存器 INTC_ISELAR8~23 进行选择，占用 Vec8~23 共计 16 个中断向量。

第三种方式

15 个中断事件请求共享 1 个中断向量，各外设都可以申请该中断并通过外设的标志位区分，通过中断选择 B 寄存器 INTC_ISELBR24~31 进行选择，占用 Vec24~31 共计 8 个中断向量。

具体中断向量分配请参考【中断事件请求序号】。

10.4.6 WFE 事件选择

中断选择 B 寄存器 INTC_ISELBR24~31 选择的中断事件请求，可通过事件使能寄存器 INTC_EVTER 被选择为事件输入来唤醒内核（WFE），此时对应的中断向量 Vec24~31 的中断输入变为无效，具体框图如下。

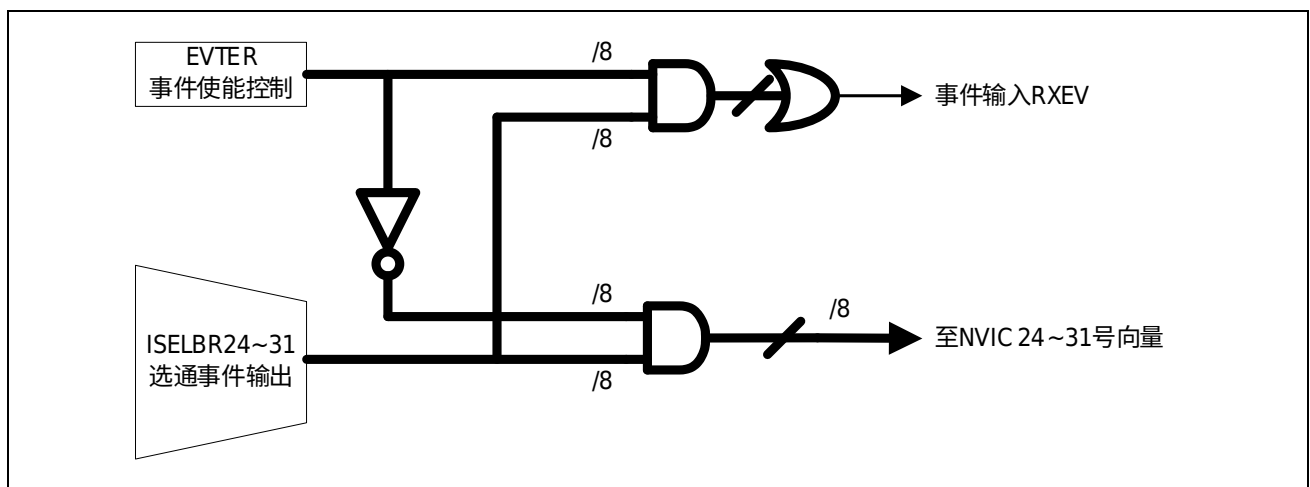


图 10-3 中断事件选择

10.4.7 WFE 唤醒事件管理

本系列产品能够通过事件输入来唤醒内核（WFE）。事件输入唤醒内核可通过以下 2 种方式产生：

- Cortex™-M0+ 系统控制寄存器 SCR.SEVONPEND=1

设定 SCR.SEVONPEND=1，在外设控制寄存器使能一个中断，根据选择的中断向量设定 INTC_ISELA/BR（选择中断向量 0~7 时，不需要设定 INTC_ISELA/BR），但不使能 NVIC 中的对应向量。当 MCU 从 WFE 恢复时，需要清除相应外设的中断标志位和 NVIC 中断标志寄存器。

WFE 停止模式进入与唤醒流程如下（INTP0 为例）：

- 1) 设定停止模式寄存器（参见 PWC 章节寄存器描述）；
- 2) 设定唤醒事件使能寄存器 INTC_WUPEN；
- 3) 设定引脚 INTP0 输入与 EIRQ 管脚中断控制寄存器 INTC_EIRQCR0；
- 4) 设置 SCR 中的 SEVONPEND 位为“1”；

- 5) 执行如下操作确保系统进入停止模式：

_SEV(); 设置内部事件寄存器

_WFE(); 清除事件寄存器

_WFE(); 系统进入停止模式

- 6) 等待所选择的中断事件请求发生，系统将从停止模式唤醒，但不进入中断处理子程序。

■ Cortex™-M0+系统控制寄存器 SCR.SEVONPEND=0

设定 SCR.SEVONPEND=0，在外设控制寄存器使能一个中断，根据选择的中断向量设定 INTC_ISELBRm (m=24~31)，同时将事件使能寄存器 INTC_EVTER 中对应位置位。当 CPU 从 WFE 恢复时，需要清除相应外设的中断标志位，不必清除 NVIC 中断标志寄存器。

WFE 停止模式进入与唤醒流程如下 (INTP9 为例)：

- 1) 设定停止模式寄存器 (参见 PWC 章节寄存器描述)；
- 2) 设定唤醒事件使能寄存器 INTC_WUPEN；
- 3) 设定引脚 INTP9 输入与 EIRQ 管脚中断控制寄存器 INTC_EIRQCR9；
- 4) 选择 INTC_ISELBR25 选择相应的 EIRQ9 中断事件请求序号；
- 5) INTC_EVTER 使能对应的事件请求；
- 6) 执行如下操作确保系统进入停止模式：

_SEV(); 设置内部事件寄存器

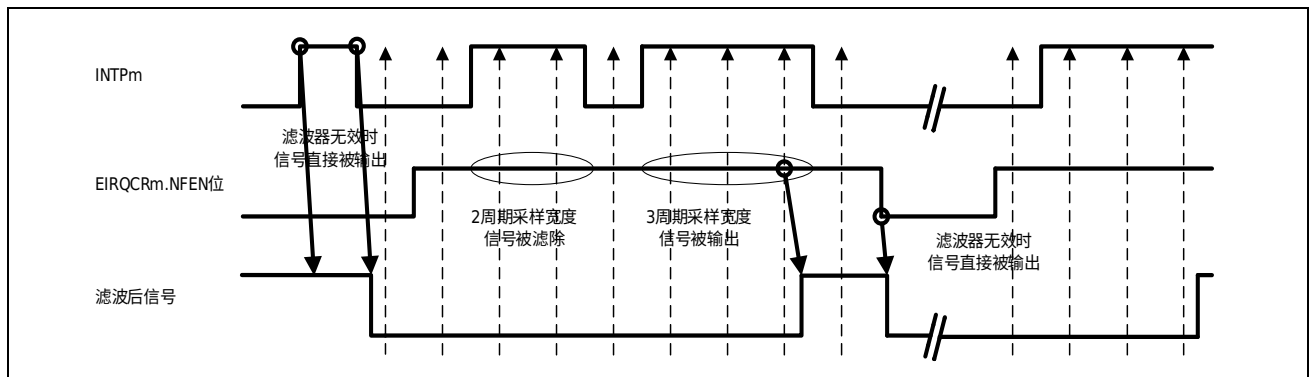
_WFE(); 清除事件寄存器

_WFE(); 系统进入停止模式

- 7) 等待所选择的中断事件请求发生，系统将从停止模式唤醒，但不进入中断处理子程序。

10.4.8 数字滤波器

对于 INTPm (m=0~11) 管脚输入，可选择数字滤波器进行噪声滤波。滤波器的采样时钟为 HCLK，小于 3 个滤波周期的输入信号将被滤除。具体工作时序图如下：



在进入停止模式前，请设定 $INTC\_EIRQCRm.EFEN=0$ ($m=0\sim11$)，禁止使用数字滤波器。从停止模式返回后再使能数字滤波器。设定流程如下：

- 1) 设定系统停止模式寄存器
- 2) 配置停止模式唤醒中断
- 3) 停止数字滤波器
- 4) 执行 WFI，系统进入停止模式

10.4.9 低功耗模式返回

10.4.9.1 休眠模式返回

当选择事件中断源作为休眠模式返回条件时，需要进行如下设定：

- 选择事件为 CPU 中断源
- 使能 NVIC 中的控制寄存器
- 如需要使用不可屏蔽中断，需要设定 $INTC\_NMIER$ 使能寄存器

10.4.9.2 停止模式返回

可以选择不可屏蔽事件中断源或在 $INTC\_WUPEN$ 寄存器中选择的可屏蔽事件源作为停止模式的返回条件。

从停止模式返回需要进行如下设定：

- 选择事件中断源作为停止模式的返回条件
 - A. 对于不可屏蔽中断，通过 $INTC\_NMIER$ 使能寄存器设定
 - B. 对于可屏蔽中断，通过 $INTC\_WUPEN$ 使能寄存器设定
- 选择事件为 CPU 中断源
- 使能 NVIC 中的控制寄存器

对于未被选中的 EIRQ 管脚，由于时钟被关闭，将不会被检测。在 WFI 指令执行前，请确认不可屏蔽中断标志寄存器 INTC_NMIFR 的所有状态位为“0”。

10.5 寄存器描述

表 10-4 INTC 基地址

名称	基地址	描述
INTC	0x4001 1000	INTC基地址

表 10-5 INTC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
INTC_FPRCR	功能保护控制寄存器	0x003C	32	0x0000 0000
INTC_NMIER	不可屏蔽中断使能寄存器	0x0004	32	0x0000 0000
INTC_NMIFR	不可屏蔽中断标志寄存器	0x0008	32	0x0000 0000
INTC_NMICLR	不可屏蔽中断标志清除寄存器	0x000C	32	0x0000 0000
INTC_EIRQCR0	EIRQ 管脚中断控制寄存器 0	0x0040	32	0x0000 0002
INTC_EIRQCR1	EIRQ 管脚中断控制寄存器 1	0x0044	32	0x0000 0002
INTC_EIRQCR2	EIRQ 管脚中断控制寄存器 2	0x0048	32	0x0000 0002
INTC_EIRQCR3	EIRQ 管脚中断控制寄存器 3	0x004C	32	0x0000 0002
INTC_EIRQCR4	EIRQ 管脚中断控制寄存器 4	0x0050	32	0x0000 0002
INTC_EIRQCR5	EIRQ 管脚中断控制寄存器 5	0x0054	32	0x0000 0002
INTC_EIRQCR6	EIRQ 管脚中断控制寄存器 6	0x0058	32	0x0000 0002
INTC_EIRQCR7	EIRQ 管脚中断控制寄存器 7	0x005C	32	0x0000 0002
INTC_EIRQCR8	EIRQ 管脚中断控制寄存器 8	0x0060	32	0x0000 0002
INTC_EIRQCR9	EIRQ 管脚中断控制寄存器 9	0x0064	32	0x0000 0002
INTC_EIRQCR10	EIRQ 管脚中断控制寄存器 10	0x0068	32	0x0000 0002
INTC_EIRQCR11	EIRQ 管脚中断控制寄存器 11	0x006C	32	0x0000 0002
INTC_EIRQFR	EIRQ 管脚中断标志寄存器	0x0074	32	0x0000 0000
INTC_EIRQCLR	EIRQ 管脚中断标志清除寄存器	0x0078	32	0x0000 0000
INTC_EKEYCR	EKEY 管脚事件控制寄存器	0x0024	32	0x0000 0000
INTC_STRGCR	软件触发事件控制寄存器	(0x4000 0C00)	32	0x0000 0000
INTC_ISELAR8	中断选择 A 寄存器 8	0x00A0	32	0x0000 0000
INTC_ISELAR9	中断选择 A 寄存器 9	0x00A4	32	0x0000 0000
INTC_ISELAR10	中断选择 A 寄存器 10	0x00A8	32	0x0000 0000
INTC_ISELAR11	中断选择 A 寄存器 11	0x00AC	32	0x0000 0000
INTC_ISELAR12	中断选择 A 寄存器 12	0x00B0	32	0x0000 0000
INTC_ISELAR13	中断选择 A 寄存器 13	0x00B4	32	0x0000 0000
INTC_ISELAR14	中断选择 A 寄存器 14	0x00B8	32	0x0000 0000
INTC_ISELAR15	中断选择 A 寄存器 15	0x00BC	32	0x0000 0000
INTC_ISELAR16	中断选择 A 寄存器 16	0x00C0	32	0x0000 0000
INTC_ISELAR17	中断选择 A 寄存器 17	0x00C4	32	0x0000 0000
INTC_ISELAR18	中断选择 A 寄存器 18	0x00C8	32	0x0000 0000

寄存器	描述	偏移地址	位宽	复位值
INTC_ISELAR19	中断选择 A 寄存器 19	0x00CC	32	0x0000 0000
INTC_ISELAR20	中断选择 A 寄存器 20	0x00D0	32	0x0000 0000
INTC_ISELAR21	中断选择 A 寄存器 21	0x00D4	32	0x0000 0000
INTC_ISELAR22	中断选择 A 寄存器 22	0x00D8	32	0x0000 0000
INTC_ISELAR23	中断选择 A 寄存器 23	0x00DC	32	0x0000 0000
INTC_ISELBR24	中断选择 B 寄存器 24	0x00E0	32	0x0000 0000
INTC_ISELBR25	中断选择 B 寄存器 25	0x00E4	32	0x0000 0000
INTC_ISELBR26	中断选择 B 寄存器 26	0x00E8	32	0x0000 0000
INTC_ISELBR27	中断选择 B 寄存器 27	0x00EC	32	0x0000 0000
INTC_ISELBR28	中断选择 B 寄存器 28	0x00F0	32	0x0000 0000
INTC_ISELBR29	中断选择 B 寄存器 29	0x00F4	32	0x0000 0000
INTC_ISELBR30	中断选择 B 寄存器 30	0x00F8	32	0x0000 0000
INTC_ISELBR31	中断选择 B 寄存器 31	0x00FC	32	0x0000 0000
INTC_EVTER	WFE 事件使能寄存器	0x0020	32	0x0000 0000
INTC_WUPEN	唤醒事件使能寄存器	0x0070	32	0x0000 0000

10.5.1 功能保护控制寄存器 (INTC_FPRCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								FPRC[7:0]							

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7~b0	FPRC	功能保护位	该位写入 A5 时, INTC 的其它寄存器才可进行写操作, 否则写动作无效。	RW

10.5.2 不可屏蔽中断使能寄存器 (INTC_NMIER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												LVD EN	SWDT EN	XTALSTP EN	Res

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
b3	LVDEN	低电压检测 LVD 中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	RW
b2	SWDTEN	SWDT 下溢/刷新错误中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	RW
b1	XTALSTPEN	主振荡器停止中断选择	0: 禁止中断作为不可屏蔽中断源 1: 选择中断作为不可屏蔽中断源	RW
b0	Res	保留位	读出时为“0”,写入时写“0”	RW

10.5.3 不可屏蔽中断标志寄存器 (INTC_NMIFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												LVDF	SWDTF	XTALSTPF	Res

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”	R
b3	LVDF	低电压检测 LVD 中断标志	0: 没有发生低电压检测 LVD 事件申请 1: 发生低电压检测 LVD 事件申请	R
b2	SWDTF	SWDT 下溢/刷新错误中断标志	0: 没有发生 SWDT 下溢/刷新错误事件申请 1: 发生 SWDT 下溢/刷新错误事件申请	R
b1	XTALSTPF	主振荡器停止中断标志	0: 没有发生主振荡器停止事件申请 1: 发生主振荡器停止事件申请	R
b0	Res	保留位	该位读出为不定	R

10.5.4 不可屏蔽中断标志清除寄存器 (INTC_NMICLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												LVD CL	SWDT CL	XTALSTP CL	Res

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
b3	LVDCL	低电压检测 LVD 中断标志清除	0: 无效 1: 清除低电压检测 LVD 标志	RW
b2	SWDTCL	SWDT 下溢/刷新错误中断标志清除	0: 无效 1: 清除 SWDT 下溢/刷新错误标志	RW
b1	XTALSTPCL	主振荡器停止中断标志清除	0: 无效 1: 清除主振荡器停止标志	RW
b0	Res	保留位	读出时为“0”,写入时写“0”	RW

注:

- 该寄存器的各有效位只可写“1”，读出时为“0”。

10.5.5 EIRQ 管脚中断控制寄存器 (INTC_EIRQCRm) (m=0~11)

复位值: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								EIRQ FEN	Res	EIRQ FCLK[1:0]		Res		EIRQ TRG[1:0]	

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EIRQFEN	EIRQ 数字滤波器使能	0: 禁止数字滤波器功能 1: 允许数字滤波器功能	RW
b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5~b4	EIRQFCLK[1:0]	EIRQ 滤波器采样时钟选择	0 0: HCLK 0 1: HCLK/8 1 0: HCLK/32 1 1: HCLK/64	RW
b3~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1~b0	EIRQTRG[1:0]	EIRQ 触发边沿选择	0 0: 低电平 0 1: 上升沿 1 0: 下降沿 1 1: 双边沿	RW

10.5.6 EIRQ 管脚中断标志寄存器 (INTC_EIRQFR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				EIRQF[11:0]											

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”	R
b11~b0	EIRQF	EIRQ 管脚中断标志位	0: EIRQ 管脚输入事件未发生 1: EIRQ 管脚输入事件发生 注: 各寄存器位与对应编号的 EIRQ 一一对应	R

10.5.7 EIRQ 管脚中断标志清除寄存器 (INTC_EIRQCLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				EIRQCL[11:0]											

位/位域	标记	位名	功能	读写
b31~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
b11~b0	EIRQCL	EIRQ 管脚中断标志位	0: 无效 1: 清除 EIRQ 输入事件状态 注: 各寄存器位与对应编号的 EIRQ 一一对应	RW

注:

- 该寄存器的各有效位只可写“1”,读出时为“0”。

10.5.8 EKEY 管脚事件控制寄存器 (INTC_EKEYCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								EKEY7 EN	EKEY6 EN	EKEY5 EN	EKEY4 EN	EKEY3 EN	EKEY2 EN	EKEY1 EN	EKEY0 EN

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EKEY7EN	EKEY7 输入使能	0: EKEY7 事件输入无效 1: EKEY7 事件输入有效	RW
b6	EKEY6EN	EKEY6 输入使能	0: EKEY6 事件输入无效 1: EKEY6 事件输入有效	RW
b5	EKEY5EN	EKEY5 输入使能	0: EKEY5 事件输入无效 1: EKEY5 事件输入有效	RW
b4	EKEY4EN	EKEY4 输入使能	0: EKEY4 事件输入无效 1: EKEY4 事件输入有效	RW
b3	EKEY3EN	EKEY3 输入使能	0: EKEY3 事件输入无效 1: EKEY3 事件输入有效	RW
b2	EKEY2EN	EKEY2 输入使能	0: EKEY2 事件输入无效 1: EKEY2 事件输入有效	RW
b1	EKEY1EN	EKEY1 输入使能	0: EKEY1 事件输入无效 1: EKEY1 事件输入有效	RW
b0	EKEY0EN	EKEY0 输入使能	0: EKEY0 事件输入无效 1: EKEY0 事件输入有效	RW

10.5.9 软件触发事件控制寄存器 (INTC_STRGCR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														STRG	

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	STRG	软件触发事件位	0: 不产生触发事件 1: 产生触发事件	W

注:

- 软件每置位一次,就产生一次触发事件 (EVENT_STRG)。
- 该寄存器位只能写 1, 不能写 0。

10.5.10 中断选择 A 寄存器 (INTC_ISELARm) (m=8~23)

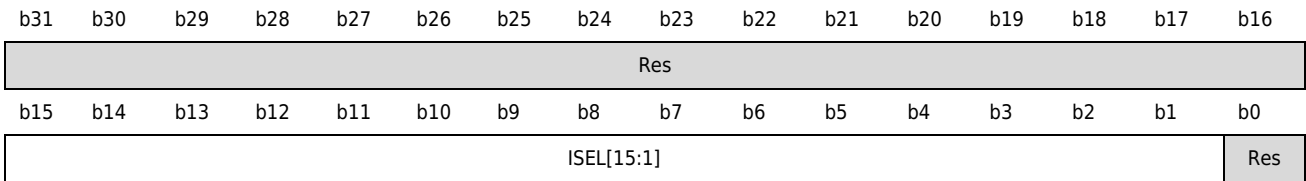
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												ISEL[3:0]			

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”,写入时写“0”	RW
b3~b0	ISEL[3:0]	中断事件请求选择	选择 10.3.2 中断事件请求序号对应的事件。具体对应关系如下: ISELAR8~9 : 选择 Group0 中的事件请求 ISELAR10~11: 选择 Group1 中的事件请求 ISELAR12~13: 选择 Group2 中的事件请求 ISELAR14~15: 选择 Group3 中的事件请求 ISELAR16~17: 选择 Group4 中的事件请求 ISELAR18~19: 选择 Group5 中的事件请求 ISELAR20~21: 选择 Group6 中的事件请求 ISELAR22~23: 选择 Group7 中的事件请求 注 1: 各 Group 中的第一个事件不能被该寄存器位选中, 即该寄存器位写 0 无效。 注 2: 在该位中写入时, 只需写入对应 Group 的组内编号, 即 1~F。	RW

10.5.11 中断选择 B 寄存器 (INTC_ISELBRm) (m=24~31)

复位值: 0x0000 0000



位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15~b1	SEL[15:1]	中断事件请求选择	选择 10.3.2 中断事件请求序号对应的事件。具体对应关系如下： ISELBR24：每一位分别使能 Gropu0 中对应的事件请求 ISELBR25：每一位分别使能 Gropu1 中对应的事件请求 ISELBR26：每一位分别使能 Gropu2 中对应的事件请求 ISELBR27：每一位分别使能 Gropu3 中对应的事件请求 ISELBR28：每一位分别使能 Gropu4 中对应的事件请求 ISELBR29：每一位分别使能 Gropu5 中对应的事件请求 ISELBR30：每一位分别使能 Gropu6 中对应的事件请求 ISELBR31：每一位分别使能 Gropu7 中对应的事件请求	RW
b0	Res	保留位	读出时为“0”,写入时写“0”	RW

10.5.12 WFE 事件使能寄存器 (INTC_EVTER)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								EVT EN7	EVT EN6	EVT EN5	EVT EN4	EVT EN3	EVT EN2	EVT EN1	EVT EN0

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EVTE7	事件选择使能位 7	0: ISELBR31 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR31 寄存器选择的事件作为事件输入送至 NVIC	RW
b6	EVTE6	事件选择使能位 6	0: ISELBR30 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR30 寄存器选择的事件作为事件输入送至 NVIC	RW
b5	EVTE5	事件选择使能位 5	0: ISELBR29 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR29 寄存器选择的事件作为事件输入送至 NVIC	RW
b4	EVTE4	事件选择使能位 4	0: ISELBR28 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR28 寄存器选择的事件作为事件输入送至 NVIC	RW
b3	EVTE3	事件选择使能位 3	0: ISELBR27 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR27 寄存器选择的事件作为事件输入送至 NVIC	RW
b2	EVTE2	事件选择使能位 2	0: ISELBR26 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR26 寄存器选择的事件作为事件输入送至 NVIC	RW
b1	EVTE1	事件选择使能位 1	0: ISELBR25 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR25 寄存器选择的事件作为事件输入送至 NVIC	RW
b0	EVTE0	事件选择使能位 0	0: ISELBR24 寄存器选择的事件作为中断输入送至 NVIC 1: ISELBR24 寄存器选择的事件作为事件输入送至 NVIC	RW

10.5.13 唤醒事件使能寄存器 (INTC_WUPEN)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							RTC PRD WUEN	RTC ALM WUEN	LVD WUEN	Res			TMR0 CMP WUEN	EKEY WUEN	SWDT WUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				EIRQ WUEN[11:0]											

位/位域	标记	位名	功能	读写
b31~b25	Res	保留位	读出时为“0”,写入时写“0”	RW
b24	RTCPRDWUEN	RTC 周期停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b23	RTCALMWUEN	RTC 闹铃停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b22	LVDWUEN	LVD 停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b21~b19	Res	保留位	读出时为“0”,写入时写“0”	RW
b18	TMR0CMPWUEN	TMR0 计数匹配停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b17	EKEYWUEN	EKEY 停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b16	SWDTWUEN	SWDT 停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	RW
b11~b0	EIRQWUEN[11:0]	EIRQ 停止模式唤醒使能	0: 唤醒禁止 1: 唤醒许可	RW

10.6 使用注意事项

关于 ARM 核中断的相关描述请参考 ARM 手册 Cortex-M0+ Technical Reference Manual。

11 DMA 控制器 (DMA)

11.1 概述

DMA 用于在存储器和外围功能模块之间传送数据，能够在 CPU 不参与的情况下实现存储器之间，存储器和外围功能模块之间以及外围功能模块之间的数据交换。

- DMA 总线独立于 CPU 总线，按照 AMBA AHB-Lite 总线协议传输
- 拥有 2 个独立通道，可以独立操作不同的 DMA 传输功能
- 每个通道的启动请求源通过独立的触发源选择寄存器配置
- 每次请求传输一个数据块
- 数据块最小为 1 个数据，最多可以是 256 个数据
- 每个数据可配置为 8bit, 16bit 或 32bit
- 可以配置 1~1023 次传输或无限次传输
- 源地址和目标地址可以独立配置为固定，自增，自减；支持以下功能四选一：源地址循环、源地址指定偏移量的跳转、目标地址循环、目标地址指定偏移量的跳转
- 可产生 3 种中断，块传输完成中断，传输完成中断，传输错误中断。每种中断都可以配置是否屏蔽。其中块传输完成，传输完成可作为事件输出，用作其它具有硬件触发功能外围模块的触发源输入
- 支持连锁传输功能，可实现一次请求传输多个数据块
- 不使用时可设置进入模块停止状态以降低功耗

11.2 模块示意图

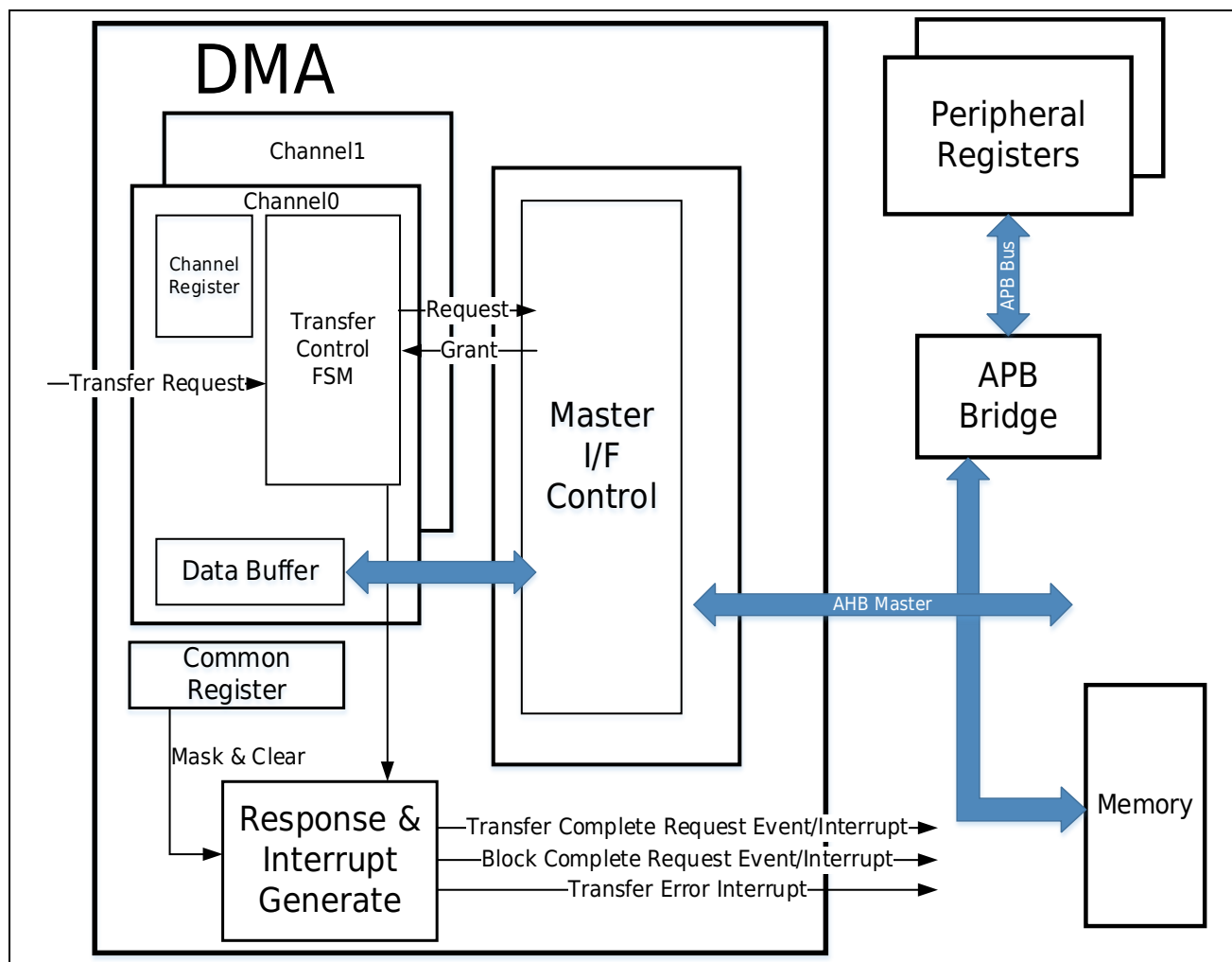


图 11-1 DMA 结构图

11.3 功能描述

11.3.1 使能 DMA 控制器

使用 DMA 时需先写 DMA 使能寄存器控制器，使能方法是写 DMA 使能寄存器 DMA_EN.EN 位。

当不使用 DMA，或者芯片需要进入 STOP 模式时，请将 DMA_EN.EN 设定为 0。EN 写 0 前请先确认寄存器 DMA_CHSTAT.DMAACT 为 0，确保 DMA 已完成所有传输。

11.3.2 通道选择和通道优先级

DMA 中包含 2 个通道，每个通道可以独立配置传输功能。2 个通道优先级顺序为通道 0>通道 1。当有多个通道有启动请求同一个总线接口时将按照优先级顺序执行。但已处于传输中的通道不会被打断，高优先通道需等当前通道传输完成后才会启动。

11.3.3 启动 DMA

DMA 通过外围电路产生的请求启动，这些请求通过触发源选择寄存器 DMA_TRGSELx(x=0~1)进行配置，可以配置通道 0 到通道 1 的启动请求源。当外围电路产生启动请求或软件写寄存器产生启动请求时，并且 DMA 传输使能有效 DMA_EN.EN=1，传输通道处于许可状态 DMA_CHEN.CHEN[x]=1，则立即启动 DMA 通道 x 传输。

使用前，需要先将功能时钟控制寄存器(CMU_FCG)的外围电路触发功能使能位置为有效。

11.3.4 数据块

DMA 每次启动传输的数据量用块(block)表示，块的大小由数据控制寄存器 DMA_CHxCTL0.BLKSIZE 设定，最大可以设置 256 个数据。块的数据宽度由 DMA_CHxCTL0.HSIZE 决定。

11.3.5 传输地址控制

传输的源地址和目标地址可以由寄存器设定为固定、递增、递减、重载或者不连续跳转。其中重载和不连续跳转只能二选一。

固定：源地址和目标地址将在传输过程中固定不变。

递增及递减：源地址和目标地址将在每传输完成 1 个数据后根据 HSIZE 的值向前或向后跳转。例如当 HSIZE 为 8bit 时，地址将每次增加/减少 1，为 16bit 时每次增加/减少 2，为 32bit 时每次增加/减少 4。

重载：寄存器可以配置源地址和目标地址在增加/减少至寄存器配置的重复区域大小时重新返回至最初的地址设定值。重复区域的大小由 DMA_CHxCTL0.HSIZE 的大小和 DMA_CHxCTL1.RPTNSCNT 寄存器设定值决定。

不连续地址传输：传输指定数量的数据后，地址将跳过指定偏移量。

11.3.6 传输次数

DMA 传输的总数据块的个数由数据控制寄存器 DMA_CHxCTL0 的 CNT 位设定。传输次数最多可设置 1023 次。每传输一个数据块寄存器值减 1，当寄存器值减为 0 时代表本次传输完成，自动清除通道传输许可位 DMA_CHEN.CHEN[x]，并产生传输完成中断。如果传输开始时 DMA_DTCTLx.CNT 被设为 0，表示无限次传输，每次启动请求传输一个数据块，但不清除通道传输许可位，也不会产生传输完成中断。

注：

- CHEN[x]自动清 0 后，需要再次启动此通道时，需要重新设置通道配置寄存器后再将 CHEN[x]写 1 使能本通道。否则将按照上次传输结束时的状态开始后续的传输，即源/目标地址为上次传输对应的下个地址，CNT 为 0 的无限次传输。

11.3.7 中断和事件信号输出

DMA 控制器可以产生以下 3 种中断：

数据块完成中断 DMA_BTCx：完成一个数据块传输后产生中断。

传输完成中断 DMA_TCx：该中断在完成数据控制寄存器 DMA_CHxCTL0 的 CNT 位设置的传输次数后发生。

传输错误中断 DMA_ERR：当启动请求溢出（即通道的上一次请求还未响应时此通道再次触发启动请求）时，或者，传输过程中发生总线错误（如访问了非法地址或受保护的地址）时产生中断，其中，总线错误时会立即终止本次传输。

以上每种中断都可以通过配置 MASK 寄存器来控制是否屏蔽中断。

DMA 控制器可以产生以下 2 种事件信号输出，可用于其它外围电路的触发源：

数据块完成 DMA_BTCx：完成一个数据块传输后。

传输完成 DMA_TCx：完成数据控制寄存器 DMA_CHxCTL0 的 CNT 位设置的传输次数后发生。

注：

- 事件不受 MASK 寄存器控制。

11.3.8 连锁传输

DMA 控制器具有连锁传输功能。连锁传输需要配置以下 4 个寄存器共 4 个 word，称为一个描述符(descriptor)，包含了连锁的各次传输的源地址，目标地址，数据控制信息，地址控制信息，链指针以及传输控制信息。可配置寄存器 DMA_CHxCTL1.LLPSEL 选择描述符存储在链描述符基地址 1 或者链描述符基地址 2。

DMA_SARx

DMA_DARx

DMA_CHxCTL0

DMA_CHxCTL1

其中 DMA_CHxCTL0.LLP 称为链指针(Linked-list Pointer)，它与链描述符基地址寄存器组合指定下一个描述符在存储器中的首地址。使用连锁传输时首先写通道控制寄存器 DMA_CHxCTL0 的 LLPEN 将连锁传输使能，并将第一次传输的描述符信息写入相应寄存器中。再将后面每次传输的描述符按顺序初始化在存储器中。需要结束连锁传输时，将通道控制寄存器 DMA_CHxCTL0 的 LLPEN 置为无效，DMA 控制器将在本次传输完成后结束连锁传输。

当一个描述符的最后一次传输结束时，LLP 指定的下一个描述符将被从存储器中载入通道配置寄存器。等待下一次传输请求输入，开始新描述符的第一次传输。或者根据寄存器 DMA_CHxCTL0.LLPRUN 的设置，在载入新描述符后直接开始第一次传输。

当 DMA_CHxCTL0.LLPRUN=0，在当前描述符的最后一次块传输之后，新的信道描述符从存储器加载到信道寄存器中，BTC 和 TC 中断照常生成，然后等待下一次传输请求以开始传输这个新加载的描述符。

当 DMA_CHxCTL0.LLPRUN=1，新描述符的传输在加载后直接开始，并且对于旧描述符的最后一个块传输既不生成 BTC 也不生成 TC 中断。

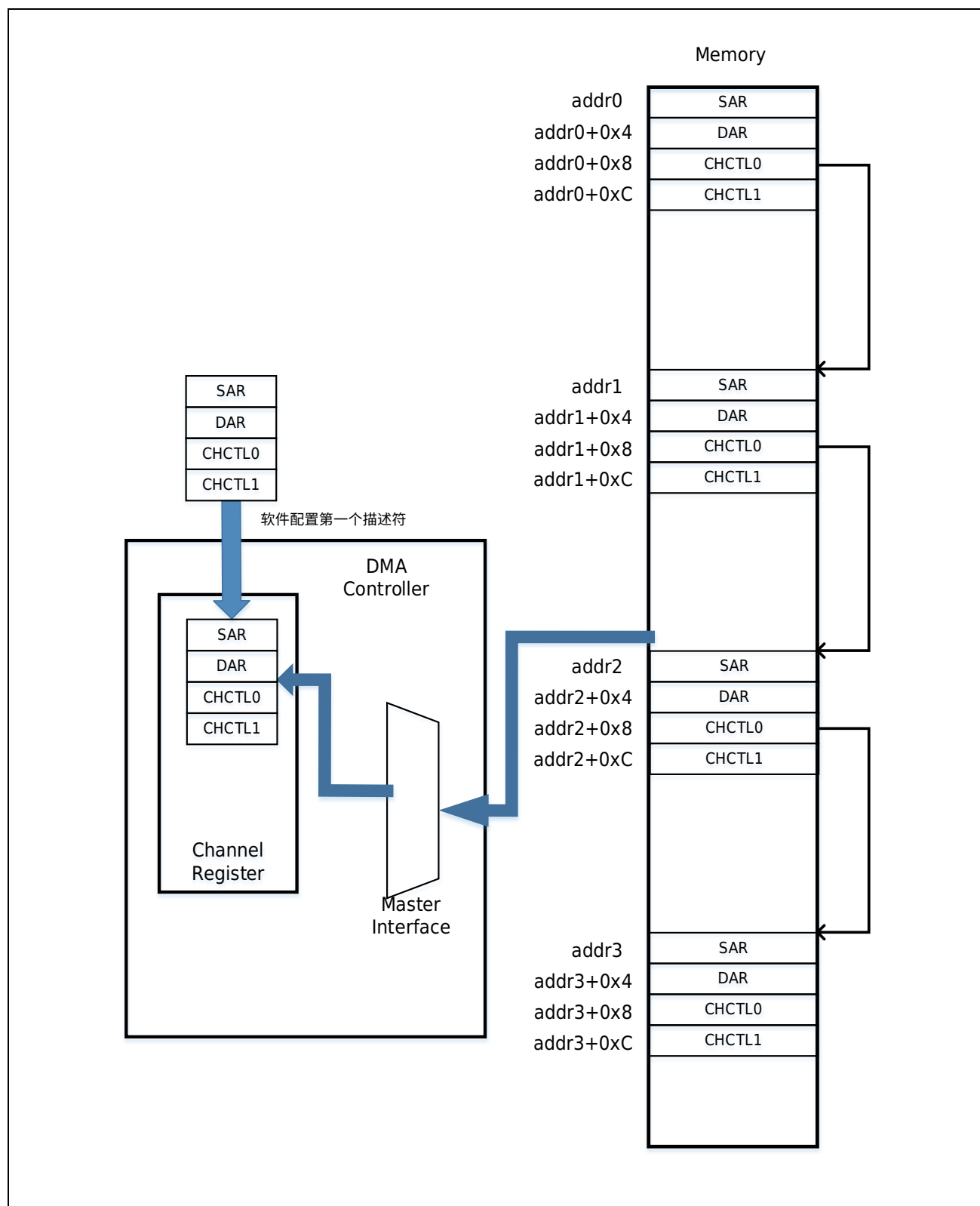


图 11-2 连锁传输示意图

11.3.9 不连续地址传输

使用不连续地址传输能够实现源地址和目标地址在传输完一定量的数据后按照一定的偏移量跳转。跳转的方向按照 DMA_CHxCTL1.SINC 和 DMA_CHxCTL1.DINC 的设定向前或向后跳转。使用时，首先根据需要将通道控制寄存器 DMA_CHxCTL1.RPTNSEN 置 1 并且配置 DMA_CHxCTL1.RPTNSSEL 使不连续地址传输有效。然后配置通道控制寄存器 DMA_CHxCTL1.OFFSET 和 DMA_CHxCTL1.RPTNSCNT。传输过程按照下图方式进行。

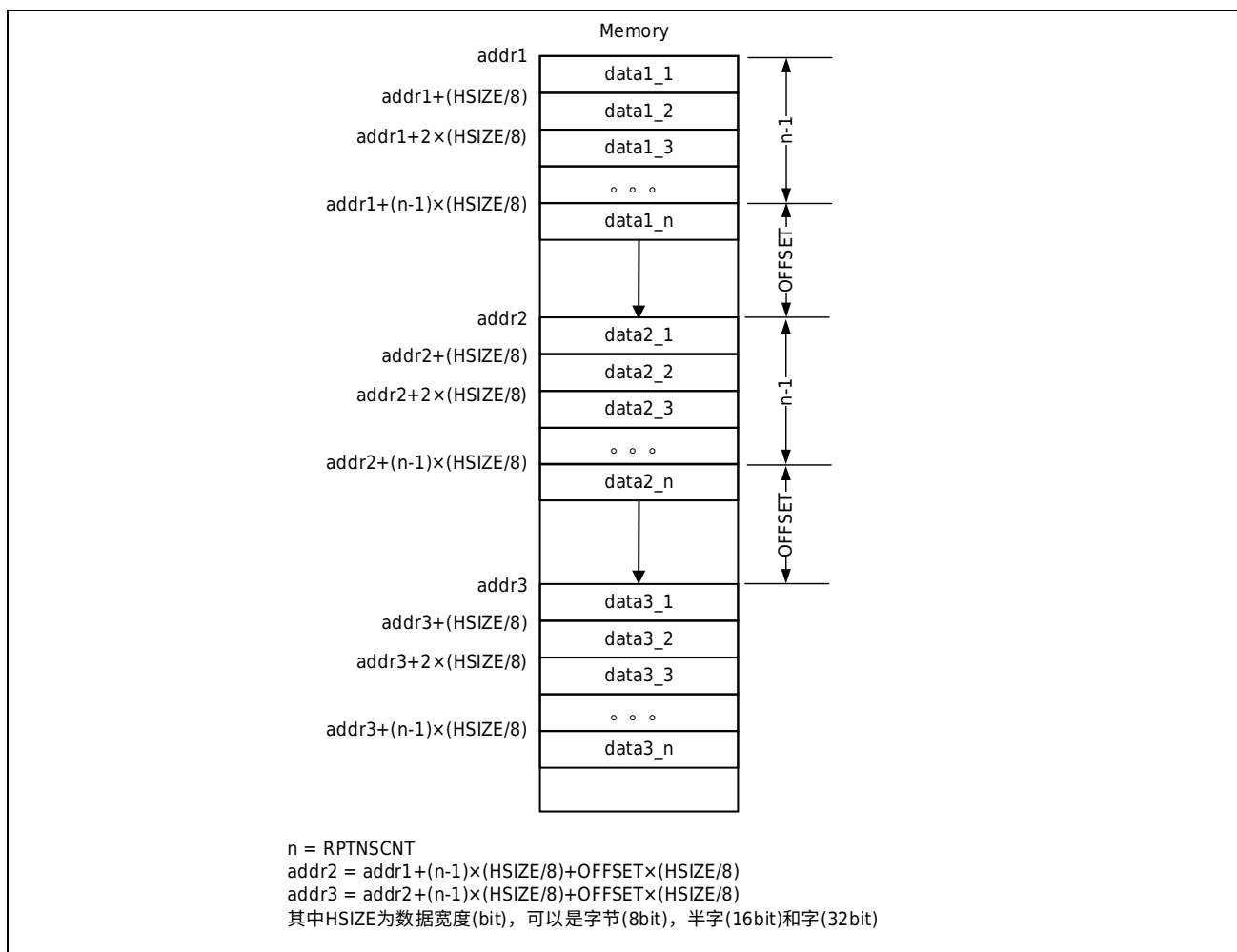


图 11-3 不连续地址传输示意图

11.3.10 传输提前终止

传输过程中通道使能寄存器 DMA_CHEN.CHEN[x]保持有效，非连锁传输时，数据控制寄存器 DMA_CHxCTL0.CNT 设定的传输次数完成后自动置为无效，连锁传输时，最后一次传输的传输次数完成后自动置为无效。传输过程中如果软件写 DMA_CHEN.CHENCLR[x]为 1，则 DMA 将在完成当次数据读写后终止传输，且 DMA_CHEN.CHENx 位清 0。

注意：

- 软件对 CHENCLR_x 位写 1 提前终止传输时，DMA 内部不会保存被终止时的传输状态。在没有重新设置通道配置寄存器（描述符）状态，将 CHEN_x 写 1 再次允许此通道，传输请求输入后，DMA 将重新传输被终止的数据块，而不是断点续传。

11.4 应用举例

11.4.1 存储器到存储器的传输

目标：将 12 个数据从 RAM 地址 0x2000 0000 传送到 0x2000 1000，数据宽度为 32bit。

1) 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道，例如通道 0，配置通道寄存器以实现：
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000
 - 写 DMA_DAR0 配置源地址为 SRAM 区域 0x2000 1000
 - 写 DMA_CH0CTL0 配置数据块的大小为 4，传输次数为 3 次，每次传输完 1 个数据块后产生块传输完成中断，3 次传输完成后产生一个传输完成中断
 - 写 DMA_CH0CTL1.RPTNSCNT 寄存器配置源地址重复区域大小为 6，即传输完成 6 个地址后重新载入初始源地址
 - 配置通道控制寄存器 DMA_CH0CTL0 和 DMA_CH0CTL1 以实现：
 - 源地址和目标地址连锁传输无效
 - 源地址重载有效，和目标地址更新方式为自增
 - 数据宽度为 word(32bit)
 - 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0
 - 配置触发源控制器 DMA0_TRGSEL，选择软件触发作为 DMA 通道 0 的启动请求
 - 写外设事件软件触发寄存器 INTSFTRG.STRG 为 1，发送第一个软件启动请求，DMA 开始传输数据

2) 传输过程

由于传输的数据块大小为 4，当软件写 INTSFTRG.STRG 为 1 后开始第一次传输，当一个数据块传输完成后，传输次数 DMA_CH0CTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTRG.STRG 开始第二次传输。在第二次传输中，由于设置了源地址重复区域大小为 6，因此在传输完 2 个地址后将根据当前传输地址和重复传输计数器 DMA_CH0CTL1.RPTNSCNT 计算得到源地址的初始地址 0x2000 0000 并继续传输剩余的 2 个地址。第二次传输完成后，传输次数 DMA_CH0CTL0.CNT 减 1，并产生一个块传输完成中断，软件可以在中断子程序中继续写 INTSFTRG.STRG 开始第三次传输。第三次传输完成后，传输次数 DMA_CH0CTL0.CNT 减为 0，即本次传输全部完成，DMA 产生一个块传输完成中断和一个传输完成中断，同时通道使能位 DMA_CHEN.CHEN0 将被自动清零。

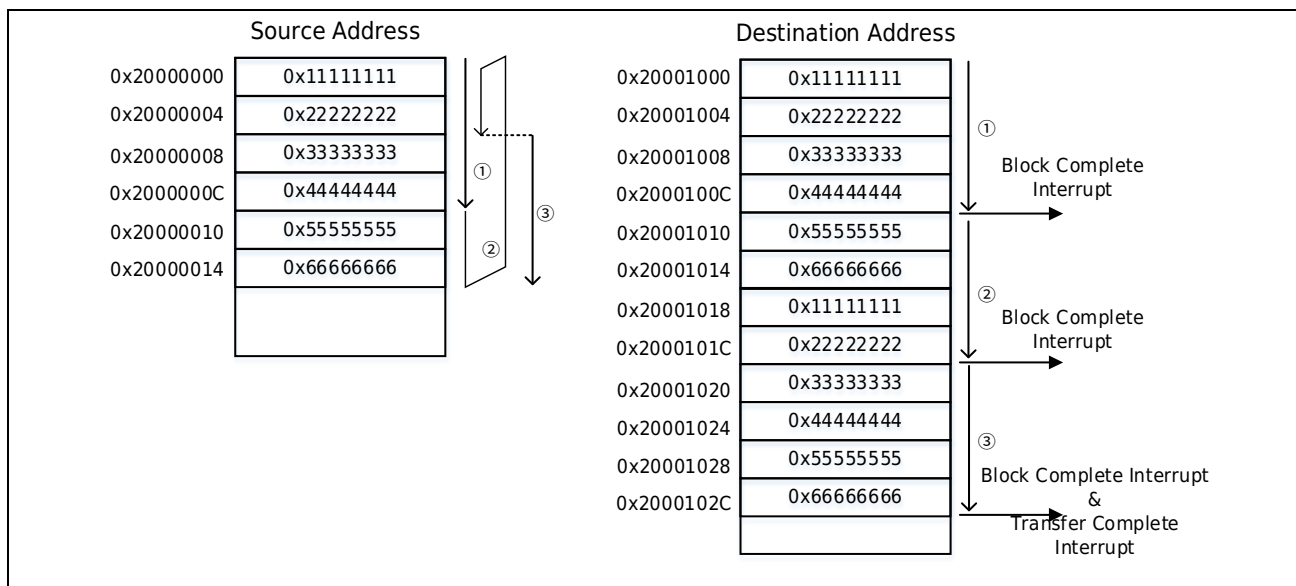


图 11-4 应用例 1：存储器到存储器传输

11.4.2 存储器到外围电路的传输

目标:将 10 个宽度为 half-word 的数据从 RAM 地址 0x2000 0000 传送到通信模块的发送缓冲寄存器中，通信模块每发送完一个数据产生一次传送请求。当最后一个数据发送完成后 DMA 产生一个传输完成中断。

1) 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 配置 DMA_INTMSK1 寄存器，屏蔽块传输完成中断，使能传输完成中断
- 选择一个通道，配置通道寄存器，例如选择通道 0
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000
 - 写 DMA_DAR0 配置源地址为外围电路的寄存器地址 0x4001 D000
 - 写 DMA_CH0CTL0 配置数据块的大小为 1，传输次数为 10 次，每个传输请求传送一次，每次 1 个数据。
 - 配置通道控制寄存器 DMA_CH0CTL0 和 DMA_CH0CTL1 以实现：
 - 源地址和目标地址连锁传输无效
 - 源地址更新方式为自增，目标地址为固定
 - 源地址和目标地址数据宽度为半字(16bit)
 - 配置触发源控制器 DMA0_TRGSEL，选择通信模块的发送寄存器空作为 DMA 通道 0 的启动请求
 - 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0

2) 传输过程

通道使能后，DMA 等待来自通信模块的传输请求。当传输请求产生后，DMA 把 RAM 中的数据传送到通信模块的发送缓冲寄存器中，并等待第二次来自通信模块的传输请求，由于屏蔽了块传输完成中断，DMA 此时不产生中断请求。当 10 个数据全部传送完成后，DMA 产生一个传输完成中断，并且通道使能位 DMA_CHEN.CHEN0 将被自动清零。

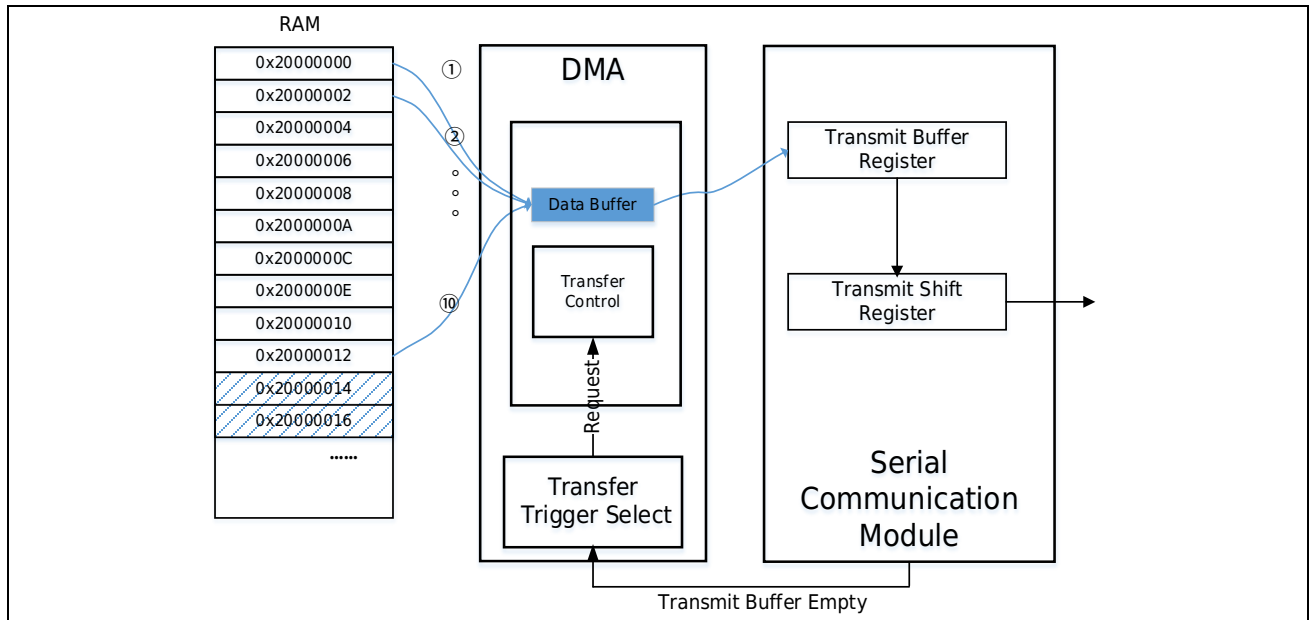


图 11-5 应用例 2：存储器到外围电路的传输

11.4.3 存储器到存储器的连锁传输

1) 寄存器设定

- DMA_EN.EN 写 1 使能 DMA 控制器
- 选择一个通道，配置通道寄存器，例如选择通道 0，配置第一次传输的描述符(descriptor0)
 - 写 DMA_SAR0 配置源地址为 SRAM 区域 0x2000 0000
 - 写 DMA_DAR0 配置目标地址为 SRAM 区域 0x2000 0100
 - 写 DMA_CH0CTL0.BLKSIZE 配置数据块的大小为 10
 - 在链指针寄存器 DMA_CH0CTL0.LLP[13:4]中写入第二个描述符(descriptor1)的地址 0x100
 - 配置通道控制寄存器 DMA_CH0CTL0 和 DMA_CH0CTL1，配置第一个数据块的传输参数以实现：
 - 连锁传输有效
 - 连锁传输模式为直接启动下一次传输
 - 源地址和目标地址更新方式为自增
 - 数据宽度为字(32bit)
- 在 RAM 空间的 0x2000 1000 地址中配置第二次传输的描述符(descriptor1)，包括：
 - 0x2000 1000 中写入 32 位数据 0x2000 0200，此为第二次传输的源地址
 - 0x2000 1004 中写入 32 位数据 0x2000 0300，此为第二次传输的目标地址

- 0x2000 1008 中配置数据块的大小为 20；配置 LLP 为 0x110，此为第三次传输的描述符(descriptor2)所在地址
- 0x2000 100C 中写入第二次传输的控制数据，该控制数据实现：
 - 连锁传输有效
 - 连锁传输模式为直接启动下一次传输
 - 源地址和目标地址更新方式为自增
 - 数据宽度为半字(16bit)
- 在 RAM 空间的 0x2000 1100 地址中配置第三次传输的描述符(descriptor2)，包括：
 - 0x2000 1100 中写入 32 位数据 0x2000 0400，此为第三次传输的源地址
 - 0x2000 1104 中写入 32 位数据 0x2000 0500，此为第三次传输的目标地址
 - 0x2000 1108 配置数据块的大小为 40；配置 LLPEN 写入 0x0，代表该次传输为连锁传输的最后一次传输
 - 0x2000 110C 中写入第三次传输的控制数据实现：
 - 连锁传输无效
 - 源地址和目标地址更新方式为自增
 - 数据宽度为字节(8bit)
- 通道使能位 DMA_CHEN.CHEN0 写 1，使能通道 0
- 配置传输启动触发源选择寄存器 DMA0_TRGSEL，选择软件触发作为 DMA 通道 0 的启动请求
- 写软件触发寄存器 INTSFTTRG.STRG 为 1，发送一个启动请求，DMA 开始传输数据

2) 传输过程

软件启动 DMA 开始传输。第一次传输完成后，由于设置了连锁传输模式为直接启动下一次传输并且中断无效，DMA 将第二次传输的描述符(descriptor1)读入到通道寄存器中。根据描述符配置的参数直接开始第二次传输。第二次传输完成后，将第三次传输的描述符(descriptor2)读入到通道寄存器中。根据描述符配置的参数开始第三次传输。第三次传输完成后，根据配置信息该次为连锁传输的最后一次，且由于中断使能有效，因此 DMA 将产生一个传输完成中断，并清除通道使能位 DMA_CHEN.CHEN0。

11.5 寄存器描述

表 11-1 DMA 寄存器列表

基地址 1: 0x4001 3000

寄存器	描述	偏移地址	位宽	复位值
DMA_EN	DMA 使能寄存器	0x00	32	0x0000 0000
DMA_INTSTAT0	中断状态寄存器 0	0x04	32	0x0000 0000
DMA_INTSTAT1	中断状态寄存器 1	0x08	32	0x0000 0000
DMA_INTMASK0	中断屏蔽寄存器 0	0x0C	32	0x0000 0000
DMA_INTMASK1	中断屏蔽寄存器 1	0x10	32	0x0000 0000
DMA_INTCLR0	中断复位寄存器 0	0x14	32	0x0000 0000
DMA_INTCLR1	中断复位寄存器 1	0x18	32	0x0000 0000
DMA_CHEN	通道使能寄存器	0x1C	32	0x0000 0000
DMA_CHSTAT	传输中通道监视寄存器	0x24	32	0x0000 0000
DMA_BASE1_LL	链描述符基地址寄存器 1	0x30	32	0x2000 0000
DMA_BASE2_LL	链描述符基地址寄存器 2	0x34	32	0x0000 0000
DMA_CHENCLR	通道使能复位寄存器	0x38	32	0x0000 0000
DMA_SARx (*1)	传输源地址寄存器	0x40+0x40*x	32	0x0000 0000
DMA_DARx	传输目标地址寄存器	0x44+0x40*x	32	0x0000 0000
DMA_CHxCTL0	数据控制寄存器	0x48+0x40*x	32	0x0000 0001
DMA_CHxCTL1	通道控制寄存器	0x4C+0x40*x	32	0x0000 1000

基地址 2: 0x4000 0C80

寄存器	描述	偏移地址	位宽	复位值
DMA_TRGSELx	传输启动触发源选择寄存器	0x00 + 0x04*x	32	0x0000 007F

注:

- 地址计算公式中 $x=0\sim1$

11.5.1 DMA 使能寄存器(DMA_EN)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															EN

位/位域	标记	位名	功能	读写
b31-b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	EN	DMA 使能位	0: DMA 无效 1: DMA 使能	RW

11.5.2 中断状态寄存器 0(DMA_INTSTAT0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															REQERR[1:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															TRNERR[1:0]

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	REQERR[1:0]	传输请求溢出错误中断位	0: 该通道没有发生传输请求溢出错误 1: 该通道发生了传输请求溢出错误,即上次请求还处在等待状态时又来了传输请求	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	TRNERR[1:0]	传输错误中断位	0: 该通道没有发生传输错误 1: 该通道发生了传输错误	RW

11.5.3 中断状态寄存器 0(DMA_INTSTAT1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														BTC[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														TC[1:0]	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	BTC[1:0]	块传输完成中断位	该中断在完成一个数据块的传输后发生 0: 该通道没有块传输中断发生 1: 该通道发生块传输中断	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	TC[1:0]	传输完成中断位	该中断在完成传输次数寄存器 DMA_CNTx 设置的传输次数后发生 0: 该通道没有传输完成中断发生 1: 该通道发生传输完成中断	RW

11.5.4 中断屏蔽寄存器(DMA_INTMASK0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														MSKREQERR[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														MSKTRNERR[1:0]	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	MSKREQERR[1:0]	传输请求溢出中断屏蔽	0: 不屏蔽传输请求溢出中断 1: 屏蔽传输请求溢出中断	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	MSKTRNERR[1:0]	传输错误中断屏蔽	0: 不屏蔽传输错误中断 1: 屏蔽传输错误中断	RW

11.5.5 中断屏蔽寄存器(DMA_INTMASK1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														MSKBTC[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														MSKTC[1:0]	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	MSKBTC[1:0]	块传输完成中断屏蔽	0: 不屏蔽块传输完成中断 1: 屏蔽块传输完成中断	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	MSKTC[1:0]	传输完成中断屏蔽	0: 不屏蔽传输完成中断 1: 屏蔽传输完成中断	RW

11.5.6 中断复位寄存器(DMA_INTCLR0)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														CLRREQERR[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CLRTRNERR[1:0]	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	CLRREQERR[1:0]	传输请求溢出错误中断复位	写 0 无任何效果, 写 1 复位传输请求溢出错误中断状态位 读出永远为 0	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	CLRTRNERR[1:0]	传输错误中断复位	写 0 无任何效果, 写 1 复位传输错误中断状态位 读出永远为 0	RW

11.5.7 中断复位寄存器(DMA_INTCLR1)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														CLRBTC[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CLRTC[1:0]	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17-b16	CLRBTC[1:0]	块传输完成中断复位	写 0 无任何效果, 写 1 复位块传输完成中断状态位 读出永远为 0	RW
b15-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	CLRTC[1:0]	传输完成中断复位	写 0 无任何效果, 写 1 复位传输完成中断状态位 读出永远为 0	RW

11.5.8 通道使能置位寄存器(DMA_CHEN)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CHEN[1:0]	

位/位域	标记	位名	功能	读写
b31-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	CHEN[1:0]	通道使能置位	每个位对应一个通道。使用该通道需将该位置 1。传输过程中使能位保持为 1, 当传输完传输次数寄存器 DMA_CHxCTL1.CNT 设置的传输次数后将会自动清零。如果 DMA_CHxCTL1.CNT 设置为 0, 则传输完成后不会自动清零, 即无限次传输。 0: 该通道无效 1: 该通道有效 说明: 本寄存器写 0 无效。若要关闭某通道, 请配置寄存器 DMA_CHENCLR	RW

11.5.9 通道状态观测寄存器(DMA_CHSTAT)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														CHACT[1]	CHACT[0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														ACT	

位/位域	标记	位名	功能	读写
b31-b18	Res	保留位	读出时为“0”,写入时写“0”	RW
b17~b16	CHACT[1:0]	传输动作中通道 监视位	每个位对应一个通道。 0: 该通道空闲中 1: 该通道处于动作中	R
b15-b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	ACT	DMA 动作中监视 位	0: DMA 未处于传输动作中 1: DMA 处于传输动作中	R

11.5.10 链描述符基地址寄存器 1 (DMA_BASE1_LLDP)

复位值: 0x2000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:14]		Res													

位/位域	标记	位名	功能	读写
b31-b14	LLP[31:14]	链描述符的基地址 1	设置链描述符的基地址 1	RW
b13-b0	Res	保留位	读出时为“0”,写入时写“0”	RW

11.5.11 链描述符基地址寄存器 2 (DMA_BASE2_LLP)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:14]		Res													

位/位域	标记	位名	功能	读写
b31-b14	LLP[31:14]	链描述符的基地址 2	设置链描述符的基地址 2	RW
b13-b0	Res	保留位	读出时为“0”,写入时写“0”	RW

11.5.12 通道使能复位寄存器(DMA_CHENCLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CHENCLR[1:0]	

位/位域	标记	位名	功能	读写
b31-b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	CHENCLR[1:0]	通道使能复位	每个位对应一个通道。不使用该通道需将该位置 1。 0: 写无效 1: 对应 CHEN[x]寄存器清 0, 关闭该通道	RW

11.5.13 传输启动触发源选择寄存器(DMA_TRGSELx) (x=0~1)

复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									TRGSEL[6:0]						

位/位域	标记	位名	功能	读写
b31-b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6-b0	TRGSEL[6:0]	触发源选择	选择启动对应通道进行传输的事件编号。 具体编号请参考【中断控制器 (INTC)】。	RW

11.5.14 传输源地址寄存器(DMA_SARx) (x=0~1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SAR[15:0]															

位/位域	标记	位名	功能	读写
设置传输源地址				
注意:				
b31-b0	SAR[31:0]	传输源地址	传输数据宽度为 16bit 时, 即 DMA_CHxctl.HSIZE=01, SAR[0]无效。 传输数据宽度为 32bit 时, 即 DMA_CHxCTL.HSIZE=1x, SAR[1:0] 无效。	RW

11.5.15 传输目标地址寄存器(DMA_DARx) (x=0~1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAR[15:0]															

位/位域	标记	位名	功能	读写
			设置传输目标地址	
			注意:	
b31-b0	DAR[31:0]	传输目标地址	传输数据宽度为 16bit 时, 即 DMA_CHxctl.HSIZE=01, DAR[0]无效。传输数据宽度为 32bit 时, 即 DMA_CHxCTL.HSIZE=1x, DAR[1:0]无效。	RW

11.5.16 数据控制寄存器(DMA_CHxCTL0) (x=0~1)

复位值: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HSIZE[1:0]		LLPRUN	LLPEN	LLP[13:4]										CNT[9:8]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[7:0]								BLKSIZE[7:0]							

位/位域	标记	位名	功能	读写
b31-b30	HSIZE[1:0]	传输数据的宽度	00: 8bit 01: 16bit 1x: 32bit	RW
b29	LLPRUN	连锁传输模式选择	连锁传输有效时, 设置当前传输完成时, 载入链指针指定的新描述符后是否立即开始新描述符对应的传输 0: 不立即传输, 等待下一个传输请求产生后开始传输 1: 新描述符载入后, 立即开始传输	RW
b28	LLPEN	连锁传输使能	0: 连锁传输无效 1: 连锁传输有效	RW
b27-b18	LLP[13:4]	链指针	连锁传输有效时, 设置下一次传输的描述符所在的地址。描述符地址的高 28 位 (即 LLP[31:14]) 由链描述基地址寄存器指定, 中间 10 位 (即 LLP[13:4]) 由本寄存器位指定, 低 4 位 (LLP[3:0]) 固定为 0	RW
b17-b8	CNT[9:0]	传输次数	总的传输次数, 每次请求启动传输一个数据块完成时 CNT 的值减 1, 当寄存器值减到 0 时将发生传输完成中断。如果传输开始时寄存器值为 0, 则每次启动请求传输 1 个数据块, CNT 保持为 0, 并且传输完成后不会产生传输完成中断。	RW
b7-b0	BLKSIZE[7:0]	数据块的大小	设置数据块的大小, 最大可以配置 256 个数据。每个数据的宽度由 DMA_CHxCTL 寄存器的 HSIZE 位决定。传输开始前需要配置寄存器的值, 如果寄存器值为 1 则每次传输 1 个数据, 设为 0 则每次传输 256 个数据。	RW

11.5.17 通道控制寄存器(DMA_CHxCTL1) (x=0~1)

复位值: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DINC[1:0]		SINC[1:0]		LLPSEL	RPTNSSEL[1:0]		RPTNSEN	RPTNSCNT[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OFFSET[15:0]															

位/位域	标记	位名	功能	读写
b31-b30	DINC[1:0]	目标地址的更新方式	00: 固定 01: 递增 1x: 递减	RW
b29-b28	SINC[1:0]	源地址的更新方式	00: 固定 01: 递增 1x: 递减	RW
b27	LLPSEL	链描述符存储地址	0: 链描述符基地址 1 1: 链描述符基地址 2	RW
b26-b25	RPTNSSEL[1:0]	源地址和目标地址重复传输或不连续传输功能选择	00: 源地址重载 01: 目标地址重载 10: 源地址不连续传输 11: 目标地址不连续传输	RW
b24	RPTNSEN	重复传输或地址不连续传输功能使能位	设置是否允许地址重新载入初始值或不连续传输 0: 不允许 1: 允许	RW
b23-b16	RPTNSCNT[7:0]	地址重复或地址跳转的数据量	当不连续地址传输时, 当传输完 CNT 设置的数据量后, 源地址或目标地址将根据 OFFSET 设置的偏移量进行跳转; 当使能地址重载功能时, CNT 表示源地址或目标地址重复区域大小。	RW
b15-b0	OFFSET[15:0]	地址跳转的地址偏移量	当不连续地址传输时, 源地址(或目标地址)跳转的地址偏移量。跳转的方向根据通道控制寄存器 DMA_CHxCTL.SINC (x=0~1)(或 DMA_CHxCTL.DINC(x=0~1))的值向前或向后跳转。当 DMA_CHxCTL.SINC(或 DMA_CHxCTL.DINC)设为地址固定时, 不连续地址传输将无效。跳转地址将根据数据的宽度 (DMA_CHxCTL.HSIZE) (x=0~1)设置的 bit 数和 OFFSET 的值进行计算: 下次传输的源地址(或目标地址)=本次传输的源地址(或目标地址)+OFFSET×(HSIZE(bit)/8) 例如, 当 OFFSET 设为 10, HSIZE 为字(32bit)时, 地址将跳转 10×4=40, HSIZE 为半字(16bit)时, 地址将跳转 10×2=20, HSIZE 为字节(8bit)时, 地址将跳转 10×1=10	RW

12.3 功能说明

12.3.1 参考时钟

CTC 连续校准时有以下三个外部时钟源可作为校准 HRC 频率的参考时钟，使用时通过 CTC_CR1 寄存器中的 REFCKS[1:0]位来选择。

- 外部高速时钟 (XTAL, $\leq 20\text{MHz}$)
- 外部低速时钟 (XTAL32, $\approx 32.768\text{KHz}$)
- 外部基准时钟 (CTCREF, $\geq 1\text{Hz}$)

HRC 的 jitter 和校准过程中的计数误差都会影响校准精度，可以通过设置 CTC_CR1 寄存器中的 REFPSC[2:0]和 HRCPSC[2:0]位分别为参考时钟和 HRC 选择合适的分频来减小误差。误差的计算公式如下所示。

$$TE = \frac{0.099\%}{\sqrt{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div 10^5}} + \frac{4}{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div PSC_{hrc}}$$

注：

- F_{hrc} 表示 HRC 的目标频率，单位 MHz； F_{ref} 表示参考时钟的频率，单位 MHz； PSC_{hrc} 表示 HRC 分频数； PSC_{ref} 表示参考时钟分频数。

CTC 单次校准时只能选择从 CTCREF 输入基准信号，并且通过 CTC_CR1 寄存器中的 REFEDG[1:0]位来选择基准信号的有效边沿。此时 REFPSC[2:0]的设定无效，每次校准只在基准信号输入的两个有效边沿间（如上升沿到上升沿，叫做一个校准周期）进行。单次校准的误差通过以下公式计算。

$$TE = \frac{0.099\%}{\sqrt{F_{hrc} \times T_{trim} \div 10^5}} + \frac{4}{F_{hrc} \times T_{trim} \div PSC_{hrc}}$$

注：

- F_{hrc} 表示 HRC 的目标频率，单位 MHz；
- T_{trim} 表示校准周期，单位 S；
- PSC_{hrc} 表示 HRC 分频数。

表 12-1 HRC 目标频率为 32MHz 时的测量误差

Fhrc		32							
Freq(ref)		0.001							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	0.130%	不可设	不可设	不可设	不可设	不可设	不可设	不可设
	2	0.136%	0.094%	不可设	不可设	不可设	不可设	不可设	不可设
	4	0.149%	0.100%	0.068%	不可设	不可设	不可设	不可设	不可设
	8	0.174%	0.113%	0.074%	0.050%	不可设	不可设	不可设	不可设
	32	0.324%	0.188%	0.112%	0.069%	0.028%	不可设	不可设	不可设
	128	0.924%	0.488%	0.262%	0.144%	0.047%	0.017%	不可设	不可设
	512	不推荐	1.688%	0.862%	0.444%	0.122%	0.036%	0.012%	不可设
	2048	不推荐	不推荐	不推荐	1.644%	0.422%	0.111%	0.030%	0.009%
Freq(ref)		0.032768							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	0.913%	0.603%	0.405%	0.276%	0.132%	不可设	不可设	不可设
	2	1.118%	0.706%	0.457%	0.302%	0.138%	不可设	不可设	不可设
	4	1.528%	0.911%	0.559%	0.353%	0.151%	0.069%	不可设	不可设
	8	不推荐	1.320%	0.764%	0.455%	0.176%	0.075%	不可设	不可设
	32	不推荐	不推荐	1.993%	1.070%	0.330%	0.114%	0.044%	不可设
	128	不推荐	不推荐	不推荐	不推荐	0.944%	0.267%	0.083%	0.028%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	0.882%	0.236%	0.067%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.851%	0.220%
Freq(ref)		0.1							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	1.863%	1.188%	0.775%	0.516%	0.238%	不可设	不可设	不可设
	2	不推荐	1.500%	0.931%	0.594%	0.258%	0.119%	不可设	不可设
	4	不推荐	不推荐	1.244%	0.750%	0.297%	0.129%	不可设	不可设
	8	不推荐	不推荐	1.869%	1.063%	0.375%	0.148%	0.064%	不可设
	32	不推荐	不推荐	不推荐	不推荐	0.844%	0.266%	0.094%	0.037%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	0.734%	0.211%	0.066%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.680%	0.184%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.652%
Freq(ref)		1							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	不推荐	不推荐	不推荐	不推荐	0.887%	0.395%	0.185%	不可设
	2	不推荐	不推荐	不推荐	不推荐	1.082%	0.444%	0.197%	不可设
	4	不推荐	不推荐	不推荐	不推荐	1.473%	0.541%	0.222%	0.099%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	0.737%	0.271%	0.111%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	1.908%	0.564%	0.184%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.735%	0.477%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.649%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
Freq		8							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	不推荐	不推荐	不推荐	不推荐	不推荐	1.369%	0.587%	0.269%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	1.760%	0.684%	0.293%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.880%	0.342%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.270%	0.440%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.026%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐

表 12-2 HRC 目标频率为 24MHz 时的测量误差

Fhrc		24							
Freq(ref)		0.001							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	0.151%	不可设	不可设	不可设	不可设	不可设	不可设	不可设
	2	0.160%	0.109%	不可设	不可设	不可设	不可设	不可设	不可设
	4	0.176%	0.118%	0.080%	不可设	不可设	不可设	不可设	不可设
	8	0.210%	0.134%	0.088%	0.059%	不可设	不可设	不可设	不可设
	32	0.410%	0.234%	0.138%	0.084%	0.034%	不可设	不可设	不可设
	128	1.210%	0.634%	0.338%	0.184%	0.059%	0.021%	不可设	不可设
	512	不推荐	不推荐	1.138%	0.584%	0.159%	0.046%	0.015%	不可设
	2048	不推荐	不推荐	不推荐	不推荐	0.559%	0.146%	0.040%	0.011%
Freq(ref)		0.032768							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	1.091%	0.715%	0.477%	0.323%	0.153%	不可设	不可设	不可设
	2	1.364%	0.851%	0.546%	0.357%	0.162%	不可设	不可设	不可设
	4	1.910%	1.125%	0.682%	0.426%	0.179%	0.081%	不可设	不可设
	8	不推荐	1.671%	0.955%	0.562%	0.213%	0.089%	不可设	不可设
	32	不推荐	不推荐	不推荐	1.381%	0.418%	0.141%	0.053%	不可设
	128	不推荐	不推荐	不推荐	不推荐	1.237%	0.345%	0.104%	0.035%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	1.165%	0.309%	0.086%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.128%	0.291%
Freq(ref)		0.1							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	不推荐	1.427%	0.923%	0.609%	0.279%	0.133%	不可设	不可设
	2	不推荐	1.844%	1.131%	0.714%	0.305%	0.139%	不可设	不可设
	4	不推荐	不推荐	1.548%	0.922%	0.357%	0.152%	0.070%	不可设
	8	不推荐	不推荐	不推荐	1.339%	0.461%	0.178%	0.076%	不可设
	32	不推荐	不推荐	不推荐	不推荐	1.086%	0.335%	0.115%	0.045%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	0.960%	0.271%	0.084%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.896%	0.240%
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.865%
Freq(ref)		1							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	不推荐	不推荐	不推荐	不推荐	1.059%	0.465%	0.216%	不可设
	2	不推荐	不推荐	不推荐	不推荐	1.320%	0.530%	0.232%	0.108%
	4	不推荐	不推荐	不推荐	不推荐	1.840%	0.660%	0.265%	0.116%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	0.920%	0.330%	0.132%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.721%	0.230%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.621%
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
Freq(ref)		8							
PSC(ref)		2	4	8	16	64	256	1024	4096
PSC(hrc)	1	不推荐	不推荐	不推荐	不推荐	不推荐	1.651%	0.695%	0.315%
	2	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	0.825%	0.348%
	4	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.086%	0.413%
	8	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.607%	0.543%
	32	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	1.324%
	128	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	512	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐
	2048	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐	不推荐

12.3.2 频率校准

CTC_CR1 寄存器的 CTCEN 位写为 1 后, 16 位校准计数器在参考时钟的触发下从 CTC_CR2 寄存器中重载 RLDVAL 并开始向下计数, 计数时钟由 HRC 提供。计数开始后, 如果参考时钟触发计数停止, 那么 16 位校准计数器将停止计数。否则 16 位校准计数器将持续计数并在计数值下溢为 0xFFFF 时停止。

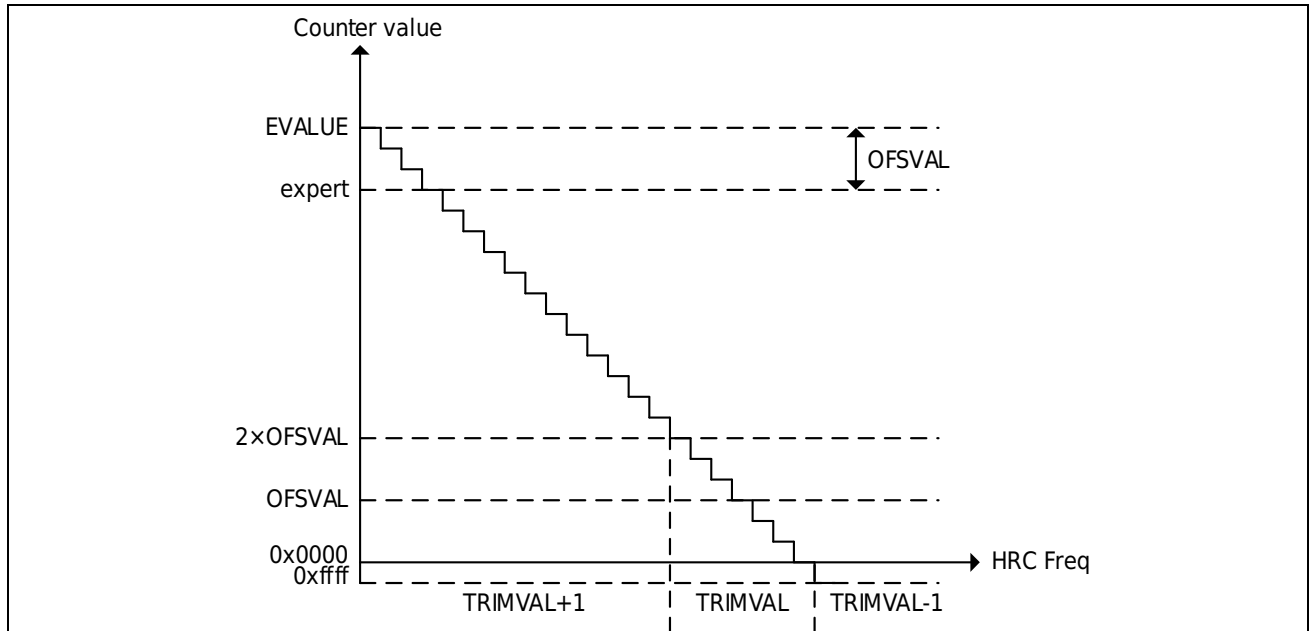


图 12-2 CTC 校准示意图

连续校准时 CTC 根据 16 位校准计数器停止计数后的值与 $2 \times \text{OFSVAL}$ 值的大小关系来决定对 CTC_CR1 寄存器中 TRMVAL[5:0] 的值进行增 1 后者减 1 操作。由于在设定校准计数重载值 RLDVAL 时加上了校准目标偏差值 OFSVAL, 因此如果 16 位校准计数器的值大于 $2 \times \text{OFSVAL}$ 且没有发生下溢, 说明当前频率比期望频率慢, 需要增大 TRMVAL 的值来提高频率。如果 16 位校准计数器的值发生下溢, 则说明当前频率比期望频率快, 需要减小 TRMVAL 的值以降低频率。而如果 16 位校准计数器的值小于 $2 \times \text{OFSVAL}$, 说明当前频率处于期望频率范围内, 就不需要调整 TRMVAL 的值。

- Counter > ($\text{OFSVAL} \times 2$) 时停止计数, TRMVAL 的值自动加 1, CTC_STR 寄存器的 TRIMOK 位更新为 0。
- $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$ 时停止计数, TRMVAL 的值不变, CTC_STR 寄存器的 TRIMOK 位更新为 1。
- Counter = 0xFFFF, CTC_CR1 寄存器的 TRMVAL 的值自动减 1, CTC_STR 寄存器的 TRIMOK 位更新为 0。

连续校准时每次 TRMVAL 调整后会自动开始下一次校准, 只在软件将 CTCEN 清零时或者发生校准错误时才停止校准。单次校准在每次 TRMVAL 调整后会自动停止, 需软件将 CTCEN 写 1 才开始下一次校准。

12.3.3 错误中断

TRMVAL 的中间值是 0x00。如果自动加 1 的结果大于 0x1F，TRMVAL 的值将保持 0x1F 并发生校准上溢错误，这时 CTC_STR 寄存器的 TRMOVF 位自动变为 1。反之，如果自动减 1 的结果小于 0x20，TRMVAL 的值将保持 0x20 并发生校准下溢错误，这时 CTC_STR 寄存器的 TRMUFD 位自动变为 1。发生校准上溢错误或下溢错误时校准会自动停止（CTCEN 自动清零），如果 CTC_CR1 寄存器中的 ERRIE 位为 1，就会发生一次校准错误中断请求。具体设定方法请参考本手册的【10 中断控制器（INTC）】章节。

12.3.4 编程指南

以使用 XTAL32 作为参考时钟将 HRC 校准到 32MHz 且校准目标偏差小于 0.5%为例。这里 XTAL32 设为 8 分频，HRC 设为不分频，经查表 12-1 可知校准误差为 0.405%，小于 0.5%的校准目标偏差，方案可行。

1) 设定寄存器。

CTC_CR2 = 0x1EAC 0027，其中

OFSVAL = $(32 \div 0.032768) \times 8 \times 0.005 \approx 39$ (0x27)；

RLDVAL = $(32 \div 0.032768) \times 8 + 39 \approx 7852$ (0x1EAC)；

2) 设定寄存器。

CTC_CR1 = 0x0000 00E2，其中

TRMVAL=0x00（从中间值开始校准）

HRCPSC=000b（选择 HRC 不分频）

CTCEN=1（启动 CTC）

ERRIE=1（允许错误中断）

REFCKS=10b（选择 XTAL32 作为参考时钟进行连续校准）

REFPSC=010b（选择参考时钟的 8 分频）

3) 一段时间后，确认 CTC_STR.TRIMOK 位的状态。

如果 TRIMOK=1，CTCEN 写 0 停止 CTC；

如果 TRIMOK=0，回到 3)；

4) 确认 CTC_STR.CTCBSY=0，后续操作。

途中如果发生错误中断，则表示由于超出可校准范围发生了校准上溢或者校准下溢，CTCEN 位会自动清零，校准停止。这时请修正 OFSVAL，RLDVAL 以及参考时钟设定后再次尝试校准。

注：

- 频率校准是通过 CTC_CR1 寄存器中的 CTCEN 位来启动或停止的。因此，启动频率校准后，请勿改变任何其他寄存器的设定值。如需修改设置，请先将 CTCEN 位写 0，并确认 CTC_STR 寄存器中的 CTCBSY 位为 0 后再进行修改。

12.4 寄存器描述

表 12-3 CTC 基地址

名称	基地址	描述
CTC	0x4000 0000	CTC基地址

表 12-4 CTC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
CTC_CR1	时钟校准控制寄存器 1	0x00	32	0x0000 0000
CTC_CR2	时钟校准控制寄存器 2	0x04	32	0x0000 0000
CTC_STR	时钟校准状态寄存器	0x08	32	0x0000 0000
CTC_CNT	时钟校准计数器	0x0C	16	0x0000

12.4.1 时钟校准控制寄存器 1 (CTC_CR1)

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										TRMVAL[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		REFEDG[1:0]		Res	HRCPSC[2:0]			CTC EN	ERR IE	REFCKS[1:0]		Res	REFPSC[2:0]		

位/位域	标记	位名	功能	读写
b31~b22	Res	保留位	读出时为“0”,写入时写“0”	R
TRMVAL 为 6 位有符号校准值，中间值是 0。				
0x20: -32				
0x21: -31				
.....				
0x3F: -1				
0x00: 0 (中间值)				
0x01: +1				
.....				
0x1E: +30				
0x1F: +31				
b21~b16	TRMVAL[5:0]	HRC 校准值	CTCEN 位写 1 时 TRMVAL 为只读，由硬件自动修改。如需修改请将 CTCEN 位写 0 并确认 CTC_STR.CTCBSY 位为 0 后再修改。 在任何情况下 HRC 最终的频率校准值均为出厂校准值加上 TRMVAL 的结果。因此如果出厂校准值下的频率已满足要求无需校准，请将 TRMVAL 设为 0x00。	RW

位/位域	标记	位名	功能	读写
b15~b14	Res	保留位	读出时为“0”,写入时写“0”	R
b13~b12	REFEDG[1:0]	参考信号边沿选择	用于 REFCKS=01b 时选择 CTCREF 的有效边沿。 00b: 选择 CTCREF 的上升沿到上升沿进行校准 01b: 选择 CTCREF 的上升沿到下降沿进行校准 10b: 选择 CTCREF 的下降沿到上升沿进行校准 11b: 选择 CTCREF 的下降沿到下降沿进行校准 CTCEN 为 1 时禁止修改 REFEDG 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	R
b10~b8	HRCCKS[1:0]	HRC 分频数选择	000b: 不分频 001b: 2 分频 010b: 4 分频 011b: 8 分频 100b: 32 分频 101b: 128 分频 110b: 512 分频 111b: 2048 分频 CTCEN 为 1 时禁止修改 HRCPSK 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	RW
b7	CTCEN	校准使能	用于启动频率校准。 0: 停止频率校准 1: 启动频率校准 置位条件: 软件置位; 清零条件: (1)软件清零; (2)校准发生上溢或者下溢时自动清零; (3)单次校准结束时自动清零;	RW
b6	ERRIE	错误中断允许	用于选择发生校准上溢或者下溢时是否产生错误中断。 0: 禁止发生错误中断 1: 允许发生错误中断	RW
b5~b4	REFCKS[1:0]	参考时钟选择	用于为频率校准选择参考时钟源。 00b: 选择 CTCREF 作为参考时钟 (连续校准) 01b: 选择 CTCREF 作为参考信号 (单次校准) 10b: 选择 XTAL32 作为参考时钟 (连续校准) 11b: 选择 XTAL 作为参考时钟 (连续校准) CTCEN 为 1 时禁止修改 REFCKS 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	RW
b3	Res	保留位	读出时为“0”,写入时写“0”	R
b2~b0	REFPSC[2:0]	参考时钟分频数选择	REFCKS=00b/10b/11b 时选择参考时钟的分频数。 000b: 2 分频 001b: 4 分频 010b: 8 分频 011b: 16 分频 100b: 64 分频 101b: 256 分频	RW

位/位域	标记	位名	功能	读写
			110b: 1024 分频 111b: 4096 分频 CTCEN 为 1 时禁止修改 REFPSC 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	

12.4.2 时钟校准控制寄存器 2 (CTC_CR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RLDVAL[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								OFSVAL[7:0]							

位/位域	标记	位名	功能	读写
b31~b16	RLDVAL[15:0]	校准计数重载值	RLDVAL 设定的是 16 位校准计数器的重载值。 连续校准时 $RLDVAL = (Fhrc \div Fref) + OFSVAL$ 单次校准时 $RLDVAL = (Fhrc \times Ttrim) + OFSVAL$ Fhrc: HRC 标准频率值, 即 24MHz 或者 32MHz; Fref: 参考时钟频率÷参考时钟预分频数; Ttrim: 单次校准周期; OFSVAL: 校准计数偏差值; CTCEN 为 1 时禁止修改 RLDVAL 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	RW
b15~b8	Res	保留位	读出时为“0”,写入时写“0”	R
b7~b0	OFSVAL[7:0]	校准目标偏差值	OFSVAL 设定的是 HRC 在校准时允许的偏差值。 连续校准时 $OFSVAL = (Fhrc \div Fref) \times TA$ 单次校准时 $OFSVAL = (Fhrc \times Ttrim) \times TA$ Fhrc: HRC 标准频率值, 即 24MHz 或者 32MHz; Fref: 参考时钟频率÷参考时钟预分频数; Ttrim: 单次校准周期; TA: 校准精度, 例如 0.5%; CTCEN 为 1 时禁止修改 OFSVAL 的值。如需修改请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再修改。	RW

12.4.3 时钟校准状态寄存器 (CTC_STR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												CTCB SY	TRMU DF	TRM OVF	TRIM OK

位/位域	标记	位名	功能	读写
b31~b4	Res	保留位	读出时为“0”,写入时写“0”	R
b3	CTCBSY	CTC 状态标志	0: CTC 处于停止状态 1: CTC 处于工作状态	R
b2	TRMU DF	校准下溢标志	校准下溢表示 TRMVAL 已减至 0x20, 但仍未校准到目标范围。校准下溢时自动停止校准, ERRIE 为 1 时将发生错误中断。 0: 未发生校准下溢 1: 发生了校准下溢 TRMU DF 在 CTCEN 写 1 时清零。	R
b1	TRM OVF	校准上溢标志	校准上溢表示 TRMVAL 已增至 0x1F, 但仍未校准到目标范围。校准上溢时自动停止校准, ERRIE 为 1 时将发生错误中断。 0: 未发生校准上溢 1: 发生了校准上溢 TRM OVF 在 CTCEN 写 1 时清零。	R
b0	TRIM OK	连续校准成功标志	0: HRC 频率未校准到目标范围 1: HRC 频率已校准到目标范围 TRIM OK 在 CTCEN 写 1 时清零。	R

12.4.4 时钟校准计数器 (CTC_CNT)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CTCCNT[15:0]															

位/位域	标记	位名	功能	读写
b15~b0	CTCCNT[15:0]	校准计数器	校准计数器, 每次校准开始时重载 RLDVAL 的值并以次为起点向下计数。校准结束后计数器值将保持到下一次 RLDVAL 重载。 请将 CTCEN 写 0 并确认 CTC_STR.CTCBSY 为 0 后再读。	R

13 模数转换模块（ADC）

13.1 概述

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。它最大拥有 14 个模拟输入通道，可以转换来自外部引脚、以及芯片内部的模拟信号。这些模拟输入通道可以任意组合成一个序列，一个序列可以进行单次扫描转换，或连续扫描转换。ADC 模块还搭载模拟看门狗功能，对任意指定通道的转换结果进行监视，检测是否超出用户设定的阈值。

ADC 主要特性

- 高性能
 - 可配置 12 位、10 位和 8 位分辨率
 - A/D 转换时钟 ADCLK 的频率可以选择为周边时钟 HCLK 的 1, 2, 4, 8, 16 或 32 分频
 - 支持 1MSPS 采样率
 - 采样时间可编程
 - 各通道独立数据寄存器
 - 数据寄存器可配置数据对齐方式
 - 模拟看门狗，监视转换结果
 - 不使用时可以将 ADC 模块设定成停止状态
- 模拟输入通道
 - 最大 14 个外部模拟输入通道
 - 1 个内部模拟检测通道，包括内部基准电压和内置温度传感器输出
- 转换开始条件
 - 软件设置转换开始
 - 周边外设同步触发转换开始
 - 外部引脚触发转换开始
- 转换模式
 - 2 个扫描序列 A、B，可任意指定单个或多个通道
 - 序列 A 单次扫描
 - 序列 A 连续扫描
 - 双序列扫描，序列 A、B 独立选择触发源，序列 B 优先级高于 A
- 中断与事件信号输出
 - 序列 A 扫描结束中断和事件 ADC_EOCA
 - 序列 B 扫描结束中断和事件 ADC_EOCB
 - 模拟看门狗 0 比较中断和事件 ADC_CMP0
 - 模拟看门狗 1 比较中断和事件 ADC_CMP1

- 上述的 4 个事件输出都可启动 DMA，仅搭载了 DMA 的产品支持

13.2 ADC 系统框图

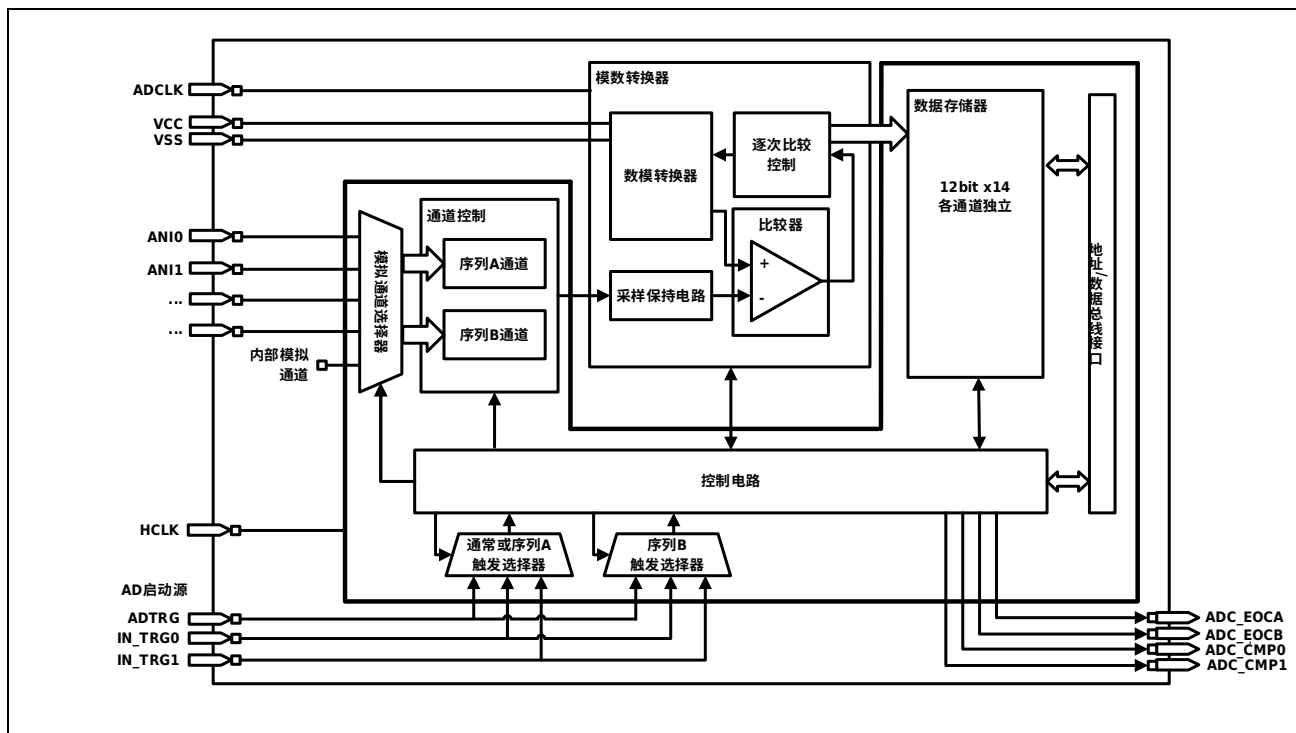


图 13-1 ADC 框图

表 13-1 各 ADC 输入信号说明

引脚名	功能
VDD, VSS	电源，ADC 的基准电压 VREFH/VREFL 与 VDD/VSS 复用
ANI0~ANI13	模拟输入
内部模拟通道	内部基准电压（约 1.575V）或者内置温度传感器输出
ADTRG	外部引脚启动触发输入
IN_TRG0	内部启动触发输入
IN_TRG1	

13.3 功能说明

13.3.1 ADC 时钟

ADC 模块需要使用 2 个时钟：模拟电路时钟 ADCLK，数字接口时钟 HCLK。ADCLK 可以设置为 HCLK 的 1, 2, 4, 8, 16 或 32 分频。具体请参考【4 时钟控制器 (CMU)】章节中 AD 时钟配置寄存器 (CMU_PERICKSEL) 的说明。

注：

- ADCLK 的频率请设置在 1MHz~32MHz 以内。

13.3.2 通道选择

ADC 模块有多个模拟输入通道，可配置为两个序列：序列 A，序列 B 进行转换。序列 A 和 B 配有独立的通道选择寄存器 ADC_CHSELRA，ADC_CHSELRB。寄存器每位代表一个通道，如 bit0 位写 1 表示转换 CH0，写 0 表示不转换 CH0。寄存器位与模拟通道的对应关系请参考寄存器 ADC_CHSELRA、B 的说明。两个序列可独立选择任意 1 个或多个通道进行转换。例如：ADC_CHSELRA 设置为 0x0055，ADC_CHSELRB 设置为 0x0002，则序列 A 的触发条件发生时，将依次转换 CH0，CH2，CH4 和 CH6 这 4 个通道。序列 B 的触发条件发生时，将转换 CH1 这一个通道。

其中，ANI11 与内部模拟通道共用通道 11。也就是说，当需要转换 ANI11，或者内部模拟通道时，需要先设置扩展通道寄存器 ADC_EXCHSELR。选择 ANI11 则 ADC_EXCHSELR.EXCHSEL 位写 0，选择内部模拟通道则 ADC_EXCHSELR.EXCHSEL 位写 1。然后再将通道选择寄存器 ADC_CHSELRA 或 B 的 bit11 位写 1，表示转换通道 11。转换结果都保存在 ADC_DR11 中。

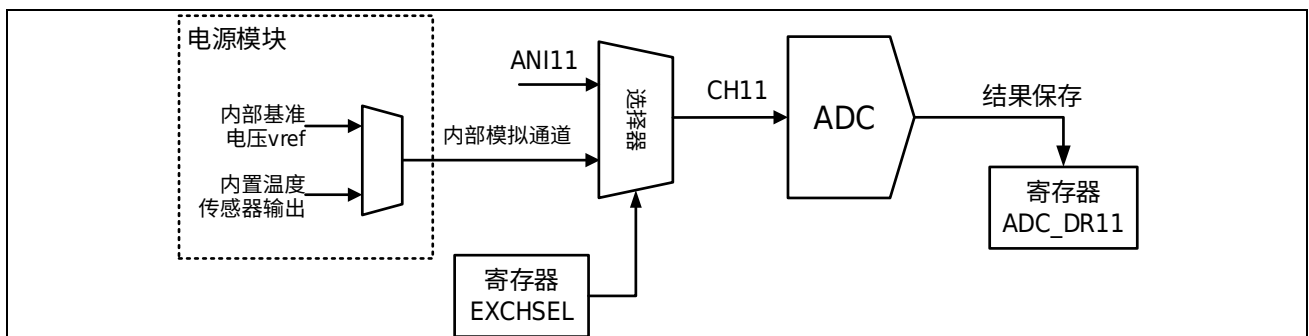


图 13-2 内部模拟通道选择

内部模拟通道的模拟量由电源模块产生，可以是内部基准电压，或者内置温度传感器。由电源控制寄存器 PWC_PWRC 选择，请参考【5 电源控制 (PWC)】章节说明。

注：

- 不要在序列 A 和 B 中选择相同的通道。对于不存在的通道，请不要设置相应的寄存器，保持其复位后的状态。

13.3.3 触发源选择

序列 A, 序列 B 独立选择触发源。可选的触发源包括外部端口 ADTRG, 内部事件 IN_TRG0, IN_TRG1。其中, 端口 ADTRG 下降沿输入有效。IN_TRG0, IN_TRG1 由寄存器 ADC_TRGSEL0, ADC_TRGSEL1 设置, 可以选择芯片内部丰富的事件源。此外, 写寄存器 ADC_STR 可生成软件触发信号, 软件触发只能在 ADC 处于待机状态时使用, 并且只适用于序列 A。

13.3.4 序列 A 单次扫描模式

A/D 控制寄存器 ADC_CR0.MS[1:0] 设为 00b 选择序列 A 单次扫描模式。

该模式下, 当寄存器 ADC_TRGSR 选择的序列 A 启动条件发生, 或者 ADC_STR.STRT 位写 1 软件触发, ADC 启动, 对序列 A 通道选择寄存器 ADC_CHSELRA 中选择的所有通道依次进行采样和转换, 转换结果存入相应的数据寄存器 ADC_DR 中。ADC 转换过程中 ADC_STR.STRT 保持为 1, 当所有通道转换结束后自动清 0, ADC 进入转换待机状态, 等待下次触发条件的发生。

当所有通道转换结束时, 序列 A 转换结束标志位 ADC_ISR.EOCAL 置 1, 并产生序列 A 转换结束事件 ADC_EOCA, 可以用此事件启动 DMA。若 ADC_ICR.EOCAIEN 为 1, 中断许可的状态下, 同时还产生序列 A 转换结束中断请求。

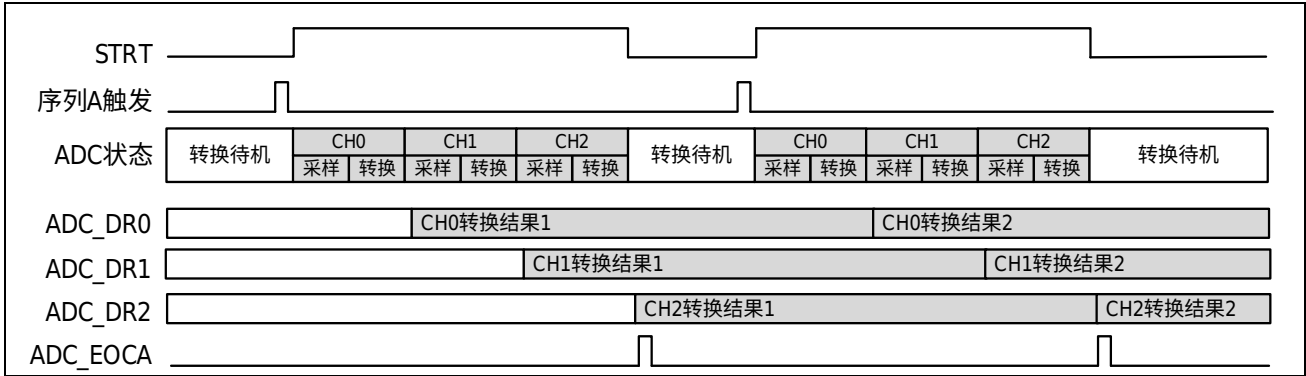


图 13-3 序列 A 单次扫描模式

序列 A 单次扫描模式的软件流程:

1. 确认 ADC_STR.STRT 为 0, ADC 处于转换待机状态。
2. A/D 控制寄存器 ADC_CR0.MS[1:0] 设为 00b 选择序列 A 单次扫描模式。
3. 设置序列 A 通道选择寄存器 ADC_CHSELRA。
4. 设置采样时间寄存器 ADC_SSTR。
5. ADC_STR.STRT 写 1 软件触发序列 A, 或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。
6. 查询序列 A 转换结束标志位 EOCAL。

- 7. 读取各通道数据寄存器 ADC_DR。
- 8. 写 0 清除 EOCAF 标志位，为下次转换准备。

上面的步骤 6~8 的 CPU 查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

13.3.5 序列 A 连续扫描模式

A/D 控制寄存器 ADCR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。

序列 A 连续扫描模式与序列 A 单次扫描模式动作类似，不同点在于，连续模式在所用通道转换结束后不是进入转换待机状态，而是重新开始转换序列 A。STRT 位也不会自动清 0。

当需要停止连续扫描时，对 STRT 位写 0，并读 STRT 确认为 0 以判断 ADC 进入转换待机状态。

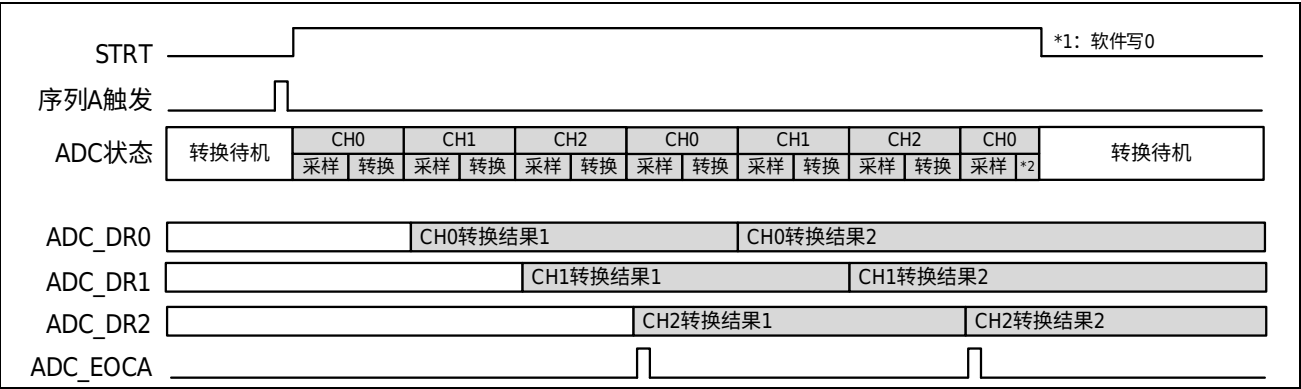


图 13-4 序列 A 连续扫描

序列 A 连续扫描模式的软件流程：

1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态。
2. A/D 控制寄存器 ADC_CR0.MS[1:0]设为 01b 选择序列 A 连续扫描模式。
3. 设置序列 A 通道选择寄存器 ADC_CHSELRA。
4. 设置采样时间寄存器 ADC_SSTR。
5. ADC_STR.STRT 写 1 软件触发序列 A，或者设置寄存器 ADC_TRGSR 选择序列 A 触发条件。
6. 查询序列 A 转换结束标志位 EOCAF。
7. 读取各通道数据寄存器 ADC_DR。
8. 写 0 清楚 EOCAF 标志位，为下次转换准备。
9. 不需要继续转换时，对 STRT 位写 0，并读 STRT 确认为 0 以判断 ADC 进入转换待机状态。

上面的步骤 6~8 的查询方式也可以替换成中断方式，利用 ADC_EOCA 中断来处理转换数据。或者利用 ADC_EOCA 事件启动 DMA 读取数据。

注：

- 由于是连续转换，每次扫描的间隔比较短，特别是只选择 1 个通道转换时。推荐使用 ADC_EOCA 事件启动 DMA 读取数据，避免查询方式下处理不及时导致数据丢失。

13.3.6 双序列扫描模式

A/D 控制寄存器 ADC_CR0.MS[1:0] 设为 10b 或 11b 选择双序列扫描模式，即序列 A 与序列 B 都可由各自选择的触发条件来启动扫描。

当 MS[1:0]=10b 时，序列 A 和 B 等效于两个独立的单次扫描序列。MS[1:0]=11b 序列 A 为连续扫描模式，B 为单次扫描模式。

序列 A 由 ADC_TRGSR.TRGSELA[2:0] 选择触发源，由 ADC_CHSELRA 选择转换的通道。序列 B 由 ADC_TRGSR.TRGSELB[2:0] 选择触发源，由 ADC_CHSELRB 选择转换的通道。

当序列 A 全部通道转换结束时，序列 A 转换结束标志位 ADC_ISR.EOCAL 置 1，并产生序列 A 转换结束事件 ADC_EOCL，若 ADC_ISR.EOCLIF 为 1，中断许可的状态下，同时产生序列 A 转换结束中断请求。当序列 B 全部通道转换结束时，序列 B 转换结束标志位 ADC_ISR.EOCBF 置 1，并产生序列 B 转换结束事件 ADC_EOCB，若 ADC_ISR.EOCBIF 为 1，中断许可的状态下，同时产生序列 B 转换结束中断请求。

双序列扫描模式下，当序列 A 与序列 B 发生竞争时，序列 B 将会被优先处理，即序列 B 优先级高于序列 A。具体情况请参看下表。

表 13-2 序列 A 和 B 的各种竞争

A/D 转换	触发信号发生	处理方式	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
序列 A 转换过程中	序列 A 触发	触发信号无效	
	序列 B 触发	1) 序列 A 的转换被中断，开始序列 B 转换 2) 序列 B 的转换完成后，序列 A 从被中断的通道开始继续转换	1) 序列 A 的转换被中断，开始序列 B 转换 2) 序列 B 的转换完成后，序列 A 从第一个通道开始重新转换
序列 B 转换过程中	序列 A 触发	序列 B 全部通道转换完成后，开始序列 A 转换	
	序列 B 触发	触发信号无效	
AD 空闲中，序列 A、B 同时触发		序列 B 先启动，全部通道转换完成后，开始序列 A 转换	

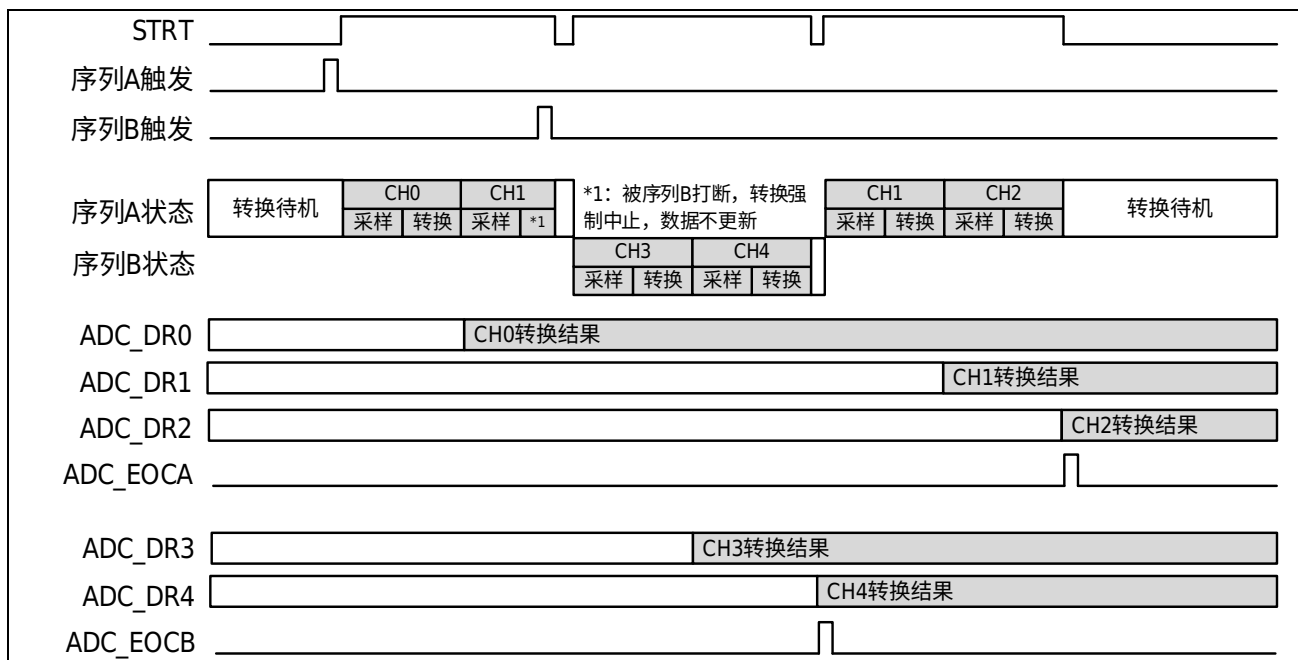


图 13-5 双序列扫描模式 (RSCHSEL=0 序列 A 从被中断通道重新启动)

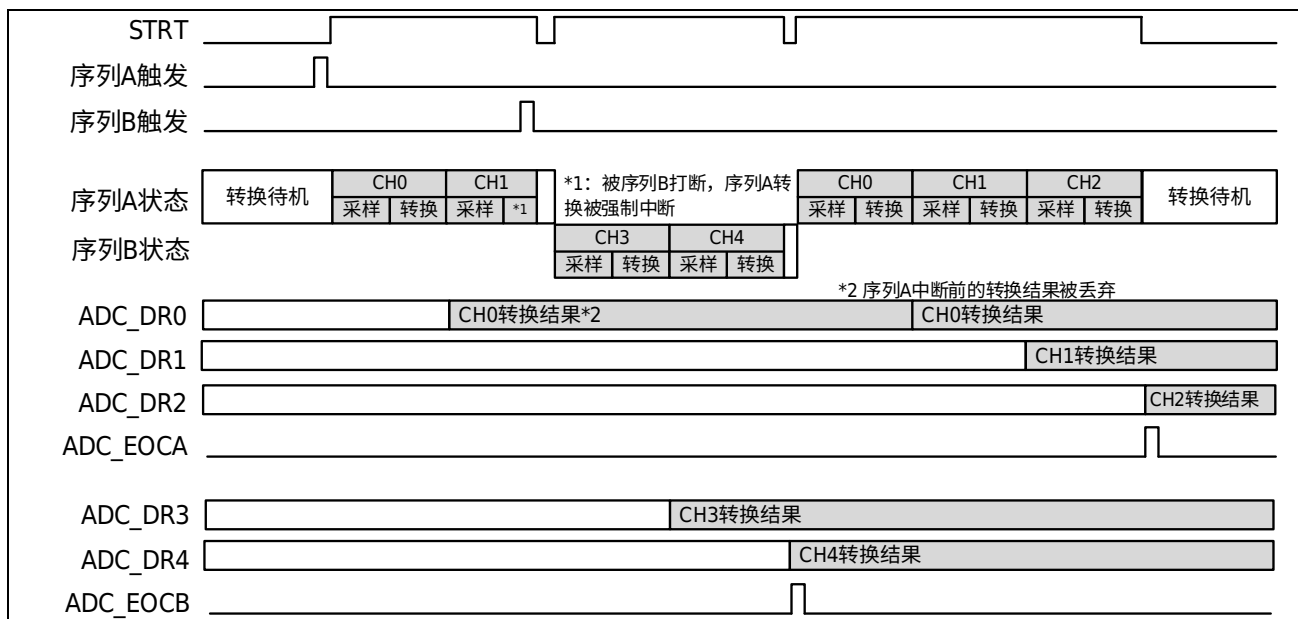


图 13-6 双序列扫描模式 (RSCHSEL=1 序列 A 从第一个通道重新启动)

双序列扫描模式的软件流程：

1. 确认 ADC_STR.STRT 为 0，ADC 处于转换待机状态。
2. A/D 控制寄存器 ADC_CR0.MS[1:0] 设为 10b 或 11b 选择双序列扫描模式。
3. 设置寄存器 ADC_CR1.RSCHSEL 选择序列 A 被打断后启动方式。
4. 设置序列 A 通道选择寄存器 ADC_CHSELRA。

5. 设置序列 B 通道选择寄存器 ADC_CHSELRB。
6. 设置采样时间寄存器 ADC_SSTR。
7. 设置寄存器 ADC_TRGSR 选择序列 A 和 B 触发条件。
8. 通过查询 EOCAF, EOCBF, 或者 ADC_EOCA, ADC_EOCB 中断, 或者启动 DMA 在序列 A 或 B 转换结束后处理转换数据。

注:

- 不要在序列 A 和 B 中选择相同的通道。序列 A 和 B 不要选择相同的触发源。

13.3.7 模拟看门狗功能

模拟看门狗功能是指在通道的 A/D 转换结束时对转换结果进行比较, 本 ADC 支持 2 个比较窗口: 比较窗口 0、比较窗口 1。以比较窗口 0 为例, 如下图所示, 若转换结果在保护区域内, 则生成看门狗比较中断和事件 ADC_CMP0。

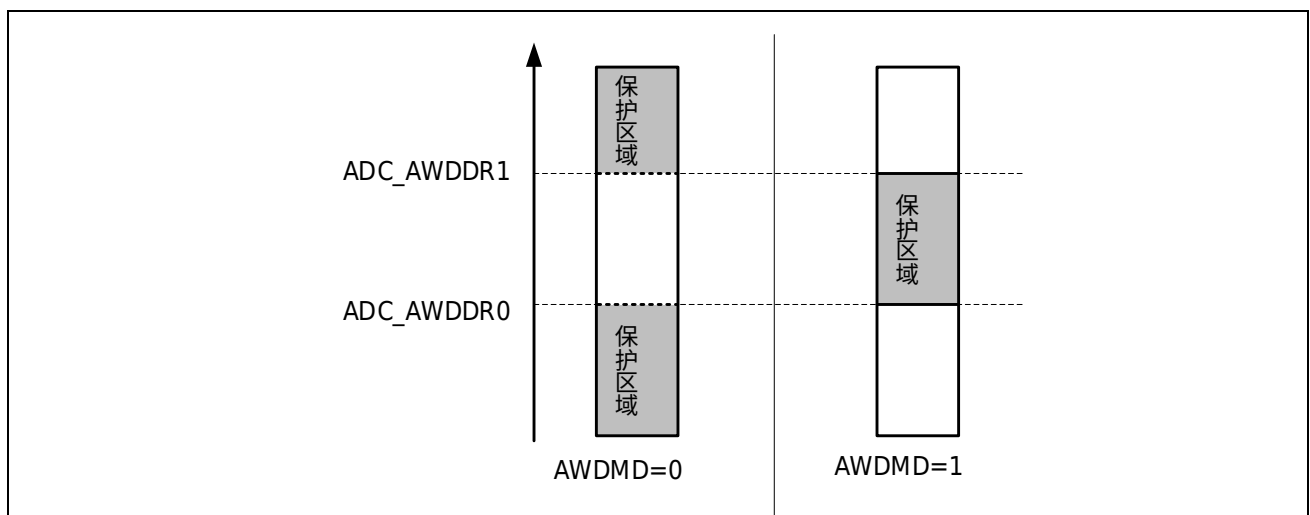


图 13-7 模拟看门狗保护区域 (比较模式)

使用模拟看门狗功能的软件流程, 以比较窗口 0 为例:

1. 设置阈值寄存器 AD_AWD0DR0, ADC_AWD0DR1
2. 设置比较通道寄存器 ADC_AWD0CHSR, 选择需要比较的通道
3. 设置 ADC_AWDCR.AWD0MD 选择保护区域
4. 设置 ADC_AWDCR.AWD0IEN 中断许可位
5. 设置 ADC_AWDCR.AWD0EN 允许窗口 0 比较功能
6. 根据前文, 设置扫描模式, 启动 AD 进行转换

7. 在 ADC_CMP0 中断中, 或 A/D 转换结束后, 查询比较状态寄存器 ADC_AWDSR.AWD0F, 对比较结果进行做相应的处理

比较窗口 1 的使用方法与比较窗口 0 相同。

2 个比较窗口可以组合使用。当窗口组合功能有效时, 在窗口 1 选择的通道转换结束时, 比较中断 ADC_CMP1 输出的不再是窗口 1 单独的比较结果, 而是按照设置, 对窗口 0 和窗口 1 的比较结果进行逻辑或, 逻辑与, 或者逻辑异或后产生的组合结果。

使用窗口组合比较功能的软件流程与窗口单独使用时相似, 需要在设置好窗口 0 和窗口 1 之后, 追加设置 ADC_AWDCR.AWDCM[1:0]寄存器选择组合方式。

13.3.8 模拟输入的采样时间和转换时间

在单次扫描模式中, A/D 转换可选择软件设置, 内部触发 IN_TRG0,1 和外部引脚触发 ADTRG 启动方式。在扫描转换延迟时间 t_D 后, ADC 模块才开始对模拟通道进行采样和转换, 全部转换结束后经过转换结束延迟时间 t_{ED} 后进入待机状态, 一次扫描才最终完成。连续扫描模式与单次扫描相似, 只是在序列的第二次以及之后的启动时没有 t_D 时间。

单个通道的转换时间 $t_{CONV} = t_{SPL} + t_{CMP}$ 。其中 t_{SPL} 表示模拟输入的采样时间, 可以根据输入阻抗设置寄存器 ADC_SSTR 调整采样周期数。 t_{CMP} 表示逐次比较时间, 12 位精度 13 个 ADCLK, 10 位精度 11 个 ADCLK, 8 位精度 9 个 ADCLK。

一次扫描转换的时间 $t_{SCAN} = t_D + \sum t_{CONV} + t_{ED}$ 。其中 $\sum t_{CONV}$ 表示所有扫描通道的转换时间总和。

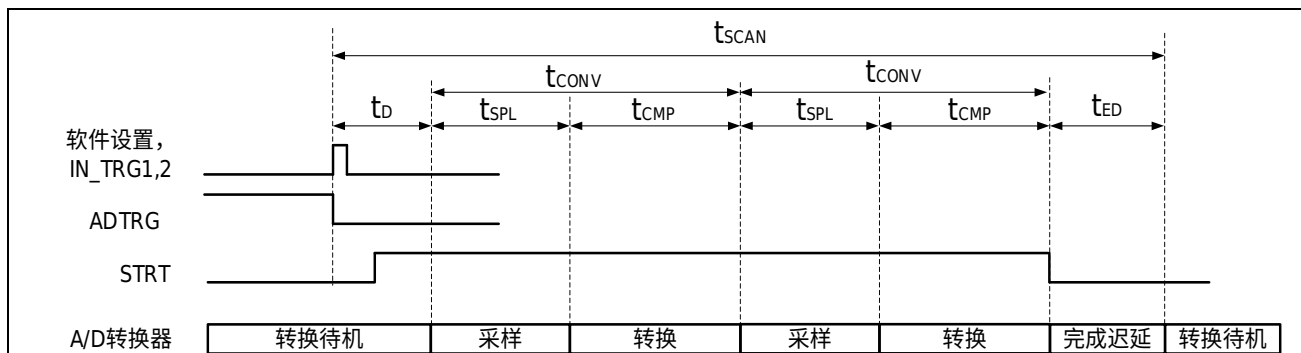


图 13-8 A/D 转换时间

表 13-3 AD 转换时间

标记		说明		条件		
				同步周边触发	外部引脚触发	软件触发
tD		扫 描	ADC 空闲中, 启动转换	4 ADCLK	2 HCLK+ 4 ADCLK	3 ADCLK
		开 始 处 理 时 间	序列 A 转换中被打断, 启动序列 B 转换	5 ADCLK	2 HCLK + 5 ADCLK	-
tCONV	tSPL	采样时间		ADC_SSTR.SST[7:0] x ADCLK		
	tCMP	逐 次 转 换 时 间	12 位分辨率	13 ADCLK		
			10 位分辨率	11 ADCLK		
			8 位分辨率	9 ADCLK		
tED		扫描完成处理时间		3 ADCLK		
tTD		最小连续触发时间间隔		$\Sigma tCONV + 6 ADCLK$		

13.3.9 A/D 数据寄存器自动清除功能

当 ADC_CR0.CLREN 为“1”，A/D 转换数据寄存器 ADC_DR 被 CPU 或者 DMA 读取后将自动被清除为“0x0000”。

使用此功能能够检测到数据寄存器 ADC_DR 是否被更新。以下将举例说明。

- 当 ADC_CR0.CLREN 为“0”，自动清除功能禁止的情况下，A/D 转换结果（0x0222）由于某种原因未被更新到数据寄存器 ADC_DR 中，ADC_DR 寄存器继续保持前次转换值（0x0111）。A/D 转换完成中断处理中将读取未被更新的（0x0111）。为检测 A/D 转换值是否被更新，需要额外将前次转换值存储到 RAM 中，通过对比转换结果来判断。
- 如果 ADC_CR0.CLREN 为“1”，自动清除功能许可的情况下，前次的转换结果（0x0111）被 CPU 或 DMA 读取后，ADC_DR 寄存器将自动被清除为“0x0000”，此后进行 A/D 转换后，如果转换结果未被正确传送到 ADC_DR 寄存器中，ADC_DR 寄存器将保持“0x0000”，这时，如果在中断处理中读出了“0x0000”，将很容易判断 A/D 转换数据是否正确被存储。

13.3.10 中断和事件信号输出

ADC 模块可以产生以下四种事件输出，每个事件发生时，若对应的中断许可寄存器设置为有效时，同时输出中断申请。

1. 序列 A 扫描结束 ADC_EOCA，对应中断许可寄存器 ADC_ICR.EOCAIEN
2. 序列 B 扫描结束 ADC_EOCB，对应中断许可寄存器 ADC_ICR.EOCBIEN
3. 模拟看门狗窗口 0 比较 ADC_CMP0，对应中断许可寄存器 ADC_AWDCR.AWD0IEN
4. 模拟看门狗窗口 1 比较 ADC_CMP1，对应中断许可寄存器 ADC_AWDCR.AWD1IEN

上述四种事件输出，可以启动其他片内周边模块，包括启动 DMA 传送。利用 DMA 传送可以连续读取 A/D 转换结果，不需要软件干预，完全由硬件实现，降低 CPU 的负荷。DMA 的设置请参考 DMA 说明章节。上述四个事件信号输出和中断使能位的控制无关，只要条件发生就会输出。



图 13-9 ADC 中断和事件输出时序

13.4 寄存器描述

表 13-4 ADC 寄存器列表

基地址 1: 0x4000 B800

寄存器	描述	偏移地址	位宽	复位值
ADC_STR	A/D 启动寄存器	0x00	8	0x00
ADC_CR0	A/D 控制寄存器 0	0x02	16	0x0000
ADC_CR1	A/D 控制寄存器 1	0x04	16	0x0000
ADC_TRGSR	A/D 转换开始触发寄存器	0x0A	16	0x0000
ADC_CHSELRA	A/D 通道选择寄存器 A	0x0C	16	0x0000
ADC_CHSELRB	A/D 通道选择寄存器 B	0x10	16	0x0000
ADC_EXCHSELR	A/D 扩展通道选择寄存器	0x18	8	0x00
ADC_SSTR	A/D 采样状态寄存器	0x20	8	0x0B
ADC_ISR	A/D 中断状态寄存器	0x44	8	0x00
ADC_ICR	A/D 中断许可寄存器	0x45	8	0x03
ADC_ISCLRR	A/D 中断状态复位寄存器	0x46	8	0x00
ADC_DRx, x=0~13	A/D 数据寄存器	0x50+2*x	16	0x0000
ADC_AWDCR	模拟看门狗控制寄存器	0xA0	16	0x0000
ADC_AWDSR	模拟看门狗状态寄存器	0xA2	8	0x00
ADC_AWDSCLRR	模拟看门狗状态复位寄存器	0xA3	8	0x00
ADC_AWD0DR0	模拟看门狗窗口 0 阈值寄存器 0	0xA4	16	0x0000
ADC_AWD0DR1	模拟看门狗窗口 0 阈值寄存器 1	0xA6	16	0xFFFF
ADC_AWD0CHSR	模拟看门狗窗口 0 比较通道选择寄存器	0xA8	8	0x00
ADC_AWD1DR0	模拟看门狗窗口 1 阈值寄存器 0	0xAC	16	0x0000
ADC_AWD1DR1	模拟看门狗窗口 1 阈值寄存器 1	0xAE	16	0xFFFF
ADC_AWD1CHSR	模拟看门狗窗口 1 比较通道选择寄存器	0xB0	8	0x00

基地址 2: 0x4000 0C60

寄存器	描述	偏移地址	位宽	复位值
ADC_TRGSEL0	A/D 启动片内触发源 0 选择寄存器	0x00	32	0x0000 007F
ADC_TRGSEL1	A/D 启动片内触发源 1 选择寄存器	0x04	32	0x0000 007F

13.4.1 A/D 启动寄存器 ADC_STR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							STRT

位/位域	标记	位名	功能	读写
b7-b1	Res	保留位	读出时为 0，写入时写 0	RW
0：停止转换 1：开始转换 置“1”条件： (1) 软件设置 (2) 选择的触发条件发生 (3) A/D 转换中 清“0”条件：				
b0	STRT	AD 转换开始	(1) 软件清“0” (2) 转换结束后自动清“0” 注意： STRT 为 0（ADC 空闲中）时写 1 产生软件触发，启动序列 A。 STRT 为 1（ADC 动作中）时写 1 无效。 STRT 为 1 时写 0 表示强制停止 AD 转换。若 ADC_TRGSR 设置了 0x0 以外的值且又不希望 ADC 再启动，请先将 ADC_TRGSR 设置为 0，再对 STRT 写 0。 STRT 为 0 时写 0 无效。	RW

13.4.2 A/D 控制寄存器 0 ADC_CR0

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
DFMT	CLREN	ACCSEL[1:0]		Res		MS[1:0]	

位/位域	标记	位名	功能	读写
b15-b8	Res	保留位	读出时为 0，写入时写 0	RW
b7	DFMT	数据格式	0：转换数据右对齐 1：转换数据左对齐	RW
b6	CLREN	数据寄存器自动清除	0：自动清除禁止 1：自动清除许可 注意：CLREN 位设定后，寄存器 ADC_DRx 将在 CPU、DMA 等读取后被自动清除。自动清除功能主要用于检测数据寄存器是否更新。	RW
b5-b4	ACCSEL[1:0]	分辨率选择	0 0：12 位分辨率 0 1：10 位分辨率 1 0：8 位分辨率	RW

1 1: 禁止设定				
b3-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1-b0	MS[1:0]	模式选择	0 0: 序列 A 单次扫描模式，序列 B 无效	RW
			0 1: 序列 A 连续扫描模式，序列 B 无效	
			1 0: 序列 A 单次扫描模式，序列 B 单次扫描模式	
			1 1: 序列 A 连续扫描模式，序列 B 单次扫描模式	

注：

- 请在 `ADC_STR.STRT` 为“0”时设置本寄存器。

13.4.3 A/D 控制寄存器 1 `ADC_CR1`

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					RSCHSEL	Res	

位/位域	标记	位名	功能	读写
b15-b3	Res	保留位	读出时为 0，写入时写 0	RW
b2	RSCHSEL	序列 A 重启通道选择	0: 被序列 B 中断后，序列 A 重启时从被中断通道开始继续扫描 1: 被序列 B 中断后，序列 A 重启时从被第一个通道开始重新扫描	RW
b1-b0	Res	保留位	读出时为 0，写入时写 0	RW

注：

- 请在 `ADC_STR.STRT` 为“0”时设置本寄存器。

13.4.4 A/D 转换开始触发寄存器 ADC_TRGSR

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	Res					TRGSELB[1]	TRGSELB[0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	Res					TRGSELA[1]	TRGSELA[0]

位/位域	标记	位名	功能	读写
b15	TRGENB	序列 B 触发使能	0: 序列 B 片内或外部引脚触发禁止 1: 序列 B 片内或外部引脚触发许可 注意: 选择外部引脚触发有效。如果 ADTRG 由“High”变为“Low”, 检测到下降沿, 则扫描转换开始, 请保持“Low”1.5*HCLK 周期以上。	RW
b14-b10	Res	保留位	读出时为 0, 写入时写 0	RW
b9-b8	TRGSELB[1:0]	序列 B 触发条件选择	在序列 B 有效模式下(ADC_CR0.MS[1]=1), 作为序列 B 的触发条件 00b: ADTRG 01b: IN_TRG0 10b: IN_TRG1 11b: IN_TRG0 + IN_TRG1 注意: 只在序列 B 有效模式下有效。其他模式设定无效。 两次触发的间隔必须大于或等于扫描周期 t_{SCAN} , 若小于则触发无效。	RW
b7	TRGENA	序列 A 触发使能	0: 序列 A 片内或外部引脚触发禁止 1: 序列 A 片内或外部引脚触发许可 注意: 选择外部引脚触发有效。如果 ADTRG 由“High”变为“Low”, 检测到下降沿, 则扫描转换开始, 请保持“Low”1.5*HCLK 周期以上。	RW
b6-b2	Res	保留位	读出时为 0, 写入时写 0	RW
b1-b0	TRGSELA[1:0]	序列 A 触发条件选择	序列 A 的触发条件。 00b: ADTRG 01b: IN_TRG0 10b: IN_TRG1 11b: IN_TRG0 + IN_TRG1 注意: ADC_STR.STRT 写 1 软件触发, 无视 TRGENA, 或者 TRGSELA[2:0] 的设定, 直接开始 A/D 转换。 两次触发的间隔必须大于或等于扫描周期 t_{SCAN} , 若小于则触发无效。	RW

注:

- 在 ADC_STR.STRT 为“0”时设置本寄存器。

13.4.5 A/D 转换开始片内触发源选择寄存器 ADC_TRGSEL0, ADC_TRGSEL1

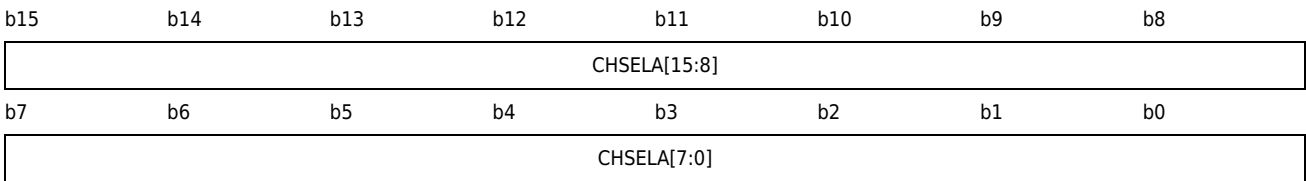
复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res	TRGSEL[6:0]						

位/位域	标记	位名	功能	读写
b31-b7	Res	保留位	读出时为 0，写入时写 0	RW
b6-b0	TRGSEL[6:0]	片内触发源选择	参考中断控制器 (INTC)，写入选定的事件编号。 触发源 IN_TRG0 由寄存器 ADC_TRGSEL0 设置 触发源 IN_TRG1 由寄存器 ADC_TRGSEL1 设置	RW

13.4.6 A/D 通道选择寄存器 A ADC_CHSELRA

复位值：0x0000



位/位域	标记	位名	功能	读写																																
			序列 A 的通道选择，每一位代表一个通道，CHSELA[x]代表通道 CHx，可选择任意组合。 0：未选择对应通道 1：选择对应通道 通道对应关系如下																																	
b15-b0	CHSELA[15:0]	转换通道选择	<table><tr><th>位</th><th>通道</th></tr><tr><td>bit0</td><td>ANI0</td></tr><tr><td>bit1</td><td>ANI1</td></tr><tr><td>bit2</td><td>ANI2</td></tr><tr><td>bit3</td><td>ANI3</td></tr><tr><td>bit4</td><td>ANI4</td></tr><tr><td>bit5</td><td>ANI5</td></tr><tr><td>bit6</td><td>ANI6</td></tr><tr><td>bit7</td><td>ANI7</td></tr><tr><td>bit8</td><td>ANI8</td></tr><tr><td>bit9</td><td>ANI9</td></tr><tr><td>bit10</td><td>ANI10</td></tr><tr><td>bit11</td><td>ANI11/内部模拟通道</td></tr><tr><td>bit12</td><td>ANI12</td></tr><tr><td>bit13</td><td>ANI13</td></tr><tr><td>bit15-bit14</td><td>reserved</td></tr></table>	位	通道	bit0	ANI0	bit1	ANI1	bit2	ANI2	bit3	ANI3	bit4	ANI4	bit5	ANI5	bit6	ANI6	bit7	ANI7	bit8	ANI8	bit9	ANI9	bit10	ANI10	bit11	ANI11/内部模拟通道	bit12	ANI12	bit13	ANI13	bit15-bit14	reserved	
			位	通道																																
			bit0	ANI0																																
			bit1	ANI1																																
			bit2	ANI2																																
			bit3	ANI3																																
			bit4	ANI4																																
			bit5	ANI5																																
			bit6	ANI6																																
			bit7	ANI7																																
			bit8	ANI8																																
			bit9	ANI9																																
			bit10	ANI10																																
			bit11	ANI11/内部模拟通道																																
			bit12	ANI12																																
bit13	ANI13																																			
bit15-bit14	reserved																																			
不存在通道的对应位为 reserved 位，读出时为 0，写入时写 0。																																				
注意：请不要再序列 A 和序列 B 中选择相同的通道。																																				

注：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。

13.4.7 A/D 通道选择寄存器 B ADC_CHSELRB

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSELB[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB[7:0]							

位/位域	标记	位名	功能	读写
b15-b0	CHSELB[15:0]	转换通道选择	序列 B 的通道选择，每一位代表一个通道，CHSELB[x]代表通道 CHx，可选择任意组合。 只有在双序列扫描模式是才有效。 0：未选择对应通道 1：选择对应通道 通道对应关系参考 ADC_CHSELRA 寄存器。 不存在通道的对应位为 reserved 位，读出时为 0，写入时写 0。 注意：请不要再序列 A 和序列 B 中选择相同的通道。	RW

注：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。

13.4.8 A/D 扩展通道选择寄存器 ADC_EXCHSELR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							EXCHSEL

位/位域	标记	位名	功能	读写
b7-b1	Res	保留位	读出时为 0，写入时写 0	RW
b0	EXCHSEL	扩展通道选择	ANI11 与内部模拟通道共同使用通道 11，通过此寄存器进行选择。 0：选择 ANI11 1：选择内部模拟通道，内部模拟源的设置请参考电源控制（PWC）说明	RW

注：

- 请在 ADC_STR.STRT 为“0”时设置本寄存器。

13.4.9 A/D 采样状态寄存器 ADC_SSTR

复位值：0x0B

b7	b6	b5	b4	b3	b2	b1	b0
SST[7:0]							

位/位域	标记	位名	功能	读写
b7-b0	SST[7:0]	采样周期数	设定模拟通道输入的采样时间，可设为 5 到 255 个 ADCLK 周期。 注意：ADCLK 频率为 20MHz 时，一个采样周期为 50ns，初始的转换状态有 11 个采样周期。当外部输入阻抗 R_{AIN} 太大采样时间不足或者 ADCLK 频率低时可以设置寄存器进行调整采样时间。采样时间不要少于 5 周期。 $SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ 其中：R_{AIN} 表示外部输入阻抗 (Ω)，R_{ADC} 表示内部采样开关电阻 (Ω)，C_{ADC} 表示内部采样和保持电容 (F)，N 表示 AD 分辨率 (12/10/8)，f_{ADC} 表示 ADCLK 频率 (Hz)。具体参考电器特性相关说明。	RW

注：

- 请在 $ADC\_STR.STRT$ 为“0”时设置本寄存器。

13.4.10 A/D 中断状态寄存器 ADC_ISR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						EOCBF	EOCAF

位/位域	标记	位名	功能	读写
b7-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1	EOCBF	序列 B 转换完成标志	序列 B 所选通道全部扫描完成后置 1 本寄存器位为只读位	R
b0	EOCAF	序列 A 转换完成标志	序列 A 所选通道全部扫描完成后置 1 本寄存器位为只读位	R

13.4.11 A/D 中断许可寄存器 ADC_ICR

复位值：0x03

b7	b6	b5	b4	b3	b2	b1	b0
Res						EOCBIEN	EOCAIEN

位/位域	标记	位名	功能	读写
b7-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1	EOCBIEN	序列 B 转换完成中断使能	0：序列 B 转换完成中断禁止 1：序列 B 转换完成中断许可	RW
b0	EOCAIEN	序列 A 转换完成中断使能	0：序列 A 转换完成中断禁止 1：序列 A 转换完成中断许可	RW

13.4.12 A/D 中断状态复位寄存器 ADC_ISCLRR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res						CLREOCBF	CLREOCAF

位/位域	标记	位名	功能	读写
b7-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1	CLREOCBF	序列 B 转换完成标志复位	写 0 无任何效果，写 1 复位 EOCBF 状态位， 读出永远为 0	RW
b0	CLREOCAF	序列 A 转换完成标志复位	写 0 无任何效果，写 1 复位 EOCAF 状态位， 读出永远为 0。	RW

13.4.13 A/D 数据寄存器 ADC_DR

ADC_DRx, x=0~13 通道 x 数据寄存器

ADC_DR 寄存器是用于存储各通道 A/D 转换数据的只读寄存器。复位值为 0x0000

根据数据对齐方式和转换分辨率，转换结果数据存储方式有所不同。

数据右对齐-12 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	AD[11:0]											

数据右对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	AD[9:0]									

数据右对齐-8 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0	AD[7:0]							

数据左对齐-12 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[11:0]												0	0	0	0

数据左对齐-10 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[9:0]										0	0	0	0	0	0

数据左对齐-8 位分辨率

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[7:0]								0	0	0	0	0	0	0	0

13.4.14 模拟看门狗控制寄存器 ADC_AWDCR

复位值：0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res						AWDCM[1]	AWDCM[0]
b7	b6	b5	b4	b3	b2	b1	b0
Res	AWD1MD	AWD1IEN	AWD1EN	Res	AWD0MD	AWD0IEN	AWD0EN

位/位域	标记	位名	功能	读写
b15-b10	Res	保留位	读出时为 0，写入时写 0	RW
b9-b8	AWDCM[1:0]	看门狗窗口组合选择	00: 窗口组合无效，ADC_CMP1 输出口 1 独立比较结果 01: 窗口组合有效，ADC_CMP1 输出口 0 与窗口 1 比较结果的逻辑与 10: 窗口组合有效，ADC_CMP1 输出口 0 与窗口 1 比较结果的逻辑与 11: 窗口组合有效，ADC_CMP1 输出口 0 与窗口 1 比较结果的逻辑异或 注意：使用窗口组合功能时需要窗口 0 和窗口 1 都使能，即 AWD0EN 写 1，AWD1EN 写 1。若窗口 0 与窗口 1 选择的比较通道不相同，请保证扫描转换过程中，窗口 1 所选通道在窗口 0 所选通道之后转换。ADC_CMP1 中断或事件在窗口 1 所选通道转换结束时输出。	RW
b7	Res	保留位	读出时为 0，写入时写 0	RW
b6	AWD1MD	看门狗窗口 1 比较模式	窗口 1 保护区域选择 0: 转换结果 < AWD1DR0，或转换结果 > AWD1DR1 1: 转换结果 ≥ AWD1DR0，且转换结果 ≤ AWD1DR1。	RW
b5	AWD1IEN	看门狗窗口 1 中断使能	0: 看门狗窗口 1 比较中断 ADC_CMP1 无效 1: 看门狗窗口 1 比较中断 ADC_CMP1 有效	RW
b4	AWD1EN	看门狗窗口 1 比较功能使能	0: 看门狗窗口 1 比较功能无效 1: 看门狗窗口 1 比较功能有效	RW
b3	Res	保留位	读出时为 0，写入时写 0	RW
b2	AWD0MD	看门狗窗口 0 比较模式	窗口 0 保护区域选择 0: 转换结果 < AWD0DR0，或转换结果 > AWD0DR1 1: 转换结果 ≥ AWD0DR0，且转换结果 ≤ AWD0DR1。	RW
b1	AWD0IEN	看门狗窗口 0 中断使能	0: 看门狗窗口 0 比较中断 ADC_CMP0 无效 1: 看门狗窗口 0 比较中断 ADC_CMP0 有效	RW
b0	AWD0EN	看门狗窗口 0 比较功能使能	0: 看门狗窗口 0 比较功能无效 1: 看门狗窗口 0 比较功能有效	RW

13.4.15 模拟看门狗状态寄存器 ADC_AWDSR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			AWDCMF	Res		AWD1F	AWD0F

位/位域	标记	位名	功能	读写
b7-b5	Res	保留位	读出时为 0，写入时写 0	RW
b4	AWDCMF	看门狗窗口组合比较状态位	看门狗窗口组合比较功能有效（即 AWDCM[1:0]=01b/10b/11b）时，窗口 1 所选通道转换结束时，窗口 0 与窗口 1 比较结果满足组合条件时置 1。对本寄存器位写无效。	R
b3-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1	AWD1F	看门狗窗口 1 比较状态位	窗口 1 所选通道转换结束时，转换结果满足比较条件时置 1。对本寄存器位写无效。	R
b0	AWD0F	看门狗窗口 0 比较状态位	窗口 0 所选通道转换结束时，转换结果满足比较条件时置 1。对本寄存器位写无效。	R

13.4.16 模拟看门狗状态复位寄存器 ADC_AWDSCLRR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			CLRAWDCMF	Res		CLRAWD1F	CLRAWD0F

位/位域	标记	位名	功能	读写
b7-b5	Res	保留位	读出时为 0，写入时写 0	RW
b4	CLRAWDCMF	看门狗窗口组合比较状态位复位	写 0 无任何效果，写 1 复位 AWDCMF 状态位，读出永远为 0	RW
b3-b2	Res	保留位	读出时为 0，写入时写 0	RW
b1	CLRAWD1F	看门狗窗口 1 比较状态复位	写 0 无任何效果，写 1 复位 AWD1F 状态位，读出永远为 0	RW
b0	CLRAWD0F	看门狗窗口 0 比较状态复位	写 0 无任何效果，写 1 复位 AWD0F 状态位，读出永远为 0	RW

13.4.17 模拟看门狗阈值寄存器 ADC_AWD0DR0、ADC_AWD0DR1、ADC_AWD1DR0、ADC_AWD1DR1

复位值：ADC_AWD0DR0=0x0000, ADC_AWD0DR1=0xFFFF

ADC_AWD1DR0=0x0000, ADC_AWD1DR1=0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8
AWDDR[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDDR[7:0]							

位/位域	标记	位名	功能	读写
b15-b0	AWDDR[15:0]	比较数据	比较数据	RW

AWD0DR0 设置窗口 0 低阈值, AWD0DR1 设置窗口 0 高阈值。

AWD1DR0 设置窗口 1 低阈值, AWD1DR1 设置窗口 1 高阈值。

AWD0DR0、AWD0DR1、AWD1DR0、AWD1DR1 根据对齐方式(数据右对齐或左对齐),分辨率(12 位、10 位或者 8 位)会有差异。

- 数据右对齐-12 位分辨率 低 12 位[11:0]可用
- 数据右对齐-10 位分辨率 低 10 位[9:0]可用
- 数据右对齐-8 位分辨率 低 8 位[7:0]可用
- 数据左对齐-12 位分辨率 高 12 位[15:4]可用
- 数据左对齐-10 位分辨率 高 10 位[15:6]可用
- 数据左对齐-8 位分辨率 高 8 位[15:8]可用

13.4.18 模拟看门狗比较通道选择寄存器 ADC_AWD0CHSR、ADC_AWD1CHSR

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			AWDCH[4]	AWDCH[3]	AWDCH[2]	AWDCH[1]	AWDCH[0]

位/位域	标记	位名	功能	读写
b7-b5	Res	保留位	读出时为 0，写入时写 0	RW
b4-b0	AWDCH[4:0]	看门狗比较通道选择	ADC_AWD0CHSR 选择窗口 0 的比较通道，ADC_AWD1CHSR 选择窗口 1 的比较通道。	RW
			0x00: ANI0	
			0x01: ANI1	
			0x02: ANI2	
			0x03: ANI3	
			0x04: ANI4	
			0x05: ANI5	
			0x06: ANI6	
			0x07: ANI7	
			0x08: ANI8	
			0x09: ANI9	
			0x0A: ANI10	
			0x0B: ANI11/内部模拟通道	
			0x0C: ANI12	
			0x0D: ANI13	
			其它：禁止设定	

13.5 使用注意事项

13.5.1 数据寄存器读取时注意事项

A/D 数据寄存器 ADC_DR 请以半字单位访问。请不要以字节单位访问数据寄存器。

13.5.2 扫描完成中断处理注意事项

当对同一通道进行连续两次扫描转换时，从第一次转换完成中断处理到第二次转完成中断处理期间，如果 CPU 没有及时将第一次的转换数据读取，那么第二次的转换数据会将第一次的转换数据覆盖。

13.5.3 模块停止与低功耗设定的注意事项

通过设定寄存器 CMU_FCG，可以设定 ADC 模块停止，降低功耗。ADC 初始状态为停止状态。需要 A/D 模块工作时请先设定 CMU_FCG 寄存器相应位取消停止，并等待 8us 以后再启动 A/D 转换。

在设置模块停止前，请先确认 A/D 处于转换停止中，即 ADC_STR.STRT 位为“0”。

在设置系统进入停止模式（STOP）前，请先将 ADC 设定为模块停止模式。

具体请参考低功耗说明章节。

13.5.4 A/D 转换模拟通道输入的引脚设定

当芯片引脚设定为 A/D 模拟通道输入时，请将对应引脚的功能选择寄存器 PCRxy.DDIS 设置为 0x1，禁用该端口的数字功能。参考【通用 IO（GPIO）】章节。

13.5.5 噪声控制

为防止浪涌等异常电压破坏模拟输入引脚，推荐使用**数据手册中电气特性**章节所示保护电路。

14 通用定时器（TimerB）

14.1 概述

通用定时器 B（TimerB）是一个具有 16 位计数宽度、1 路 PWM 输出的定时器。该定时器支持三角波和锯齿波两种波形模式，可生成各种 PWM 波形（单边对齐 PWM、双边对称 PWM）；支持计数器同步启动；支持单元间级联实现 32 位计数；支持脉宽测量和周期测量。本系列产品搭载 8 个单元 TimerB。

14.2 基本框图

TimerB 基本的功能及特性如表 14-1 所示。

表 14-1 TimerB 的基本功能及特性

波形模式	锯齿波、三角波
基本功能	递加、递减计数方向
	同步启动计数器
	32 位级联计数
	脉宽测量
	周期测量
	1 路 PWM 输出
	比较匹配事件输出
中断类型	比较匹配中断
	周期匹配中断

TimerB 的基本框图如图 14-1 所示。图中“<t>”为单元编号，即“<t>”为 1~8，本章节后文提到“<t>”时均指单元编号，不再赘述。

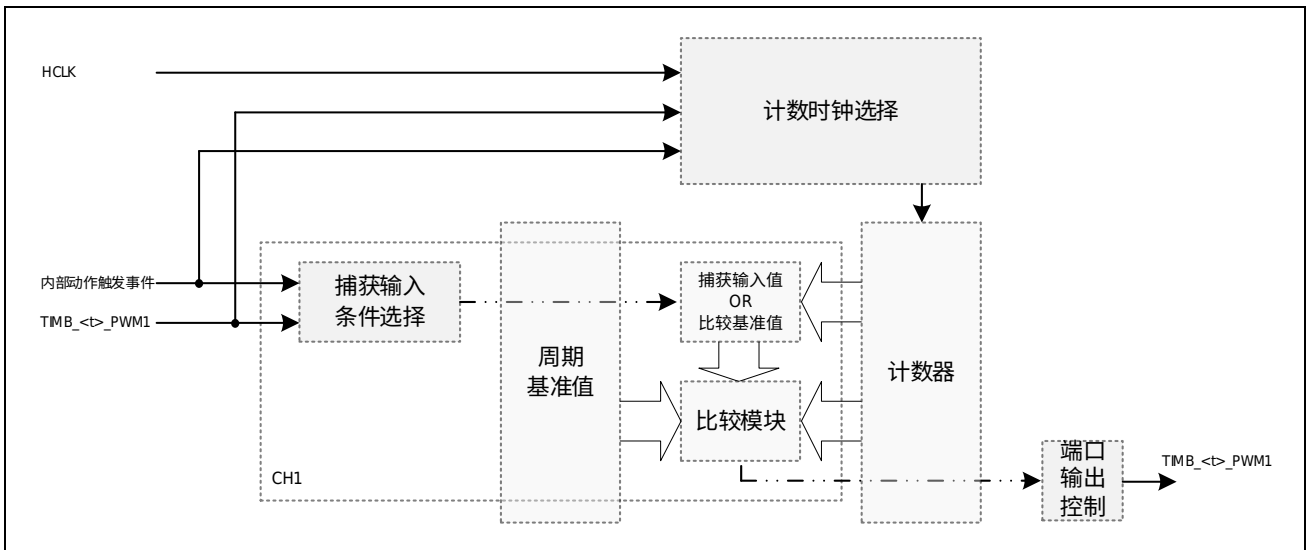


图 14-1 TimerB 基本框图

下表所示为 TimerB 的输入输出端口列表。

表 14-2 TimerB 端口列表

端口名	方向	功能
TMRB_<t>_PWM1	in or out	1) 捕获输入事件端口或 PWM 输出端口 2) 硬件触发启动、停止、清零、计数事件输入端口 3) 单元 4 在与 USART 连动实现 LIN 功能时，端口输入的相关功能（启动、停止、清零、计数、捕获输入）无效，具体参考 USART 章节

14.3 功能说明

14.3.1 基本动作

14.3.1.1 波形模式

TimerB 有 2 种基本计数波形模式，锯齿波模式和三角波模式。两种波形模式的基本波形如图 14-2、图 14-3 所示。

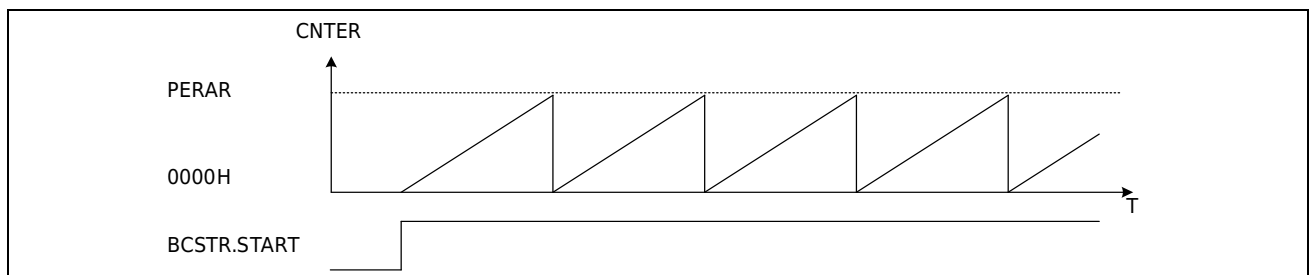


图 14-2 锯齿波波形（递加计数）

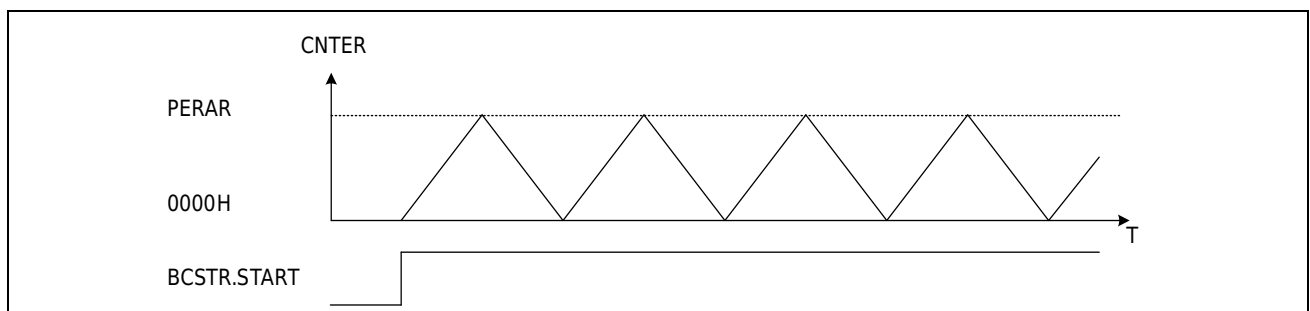


图 14-3 三角波波形

14.3.1.2 比较输出

各个 TimerB 单元内部均含有 1 个通道的比较输出（TMRB_<t>_PWM1），可在计数值与比较基准值比较匹配时输出指定的电平。TMRB_CMPAR 寄存器分别对应了 TMRB_<t>_PWM1 输出端口的计数比较基准值。当定时器的计数值和 TMRB_CMPAR 相等时，TMRB_<t>_PWM1 端口输出指定的电平。

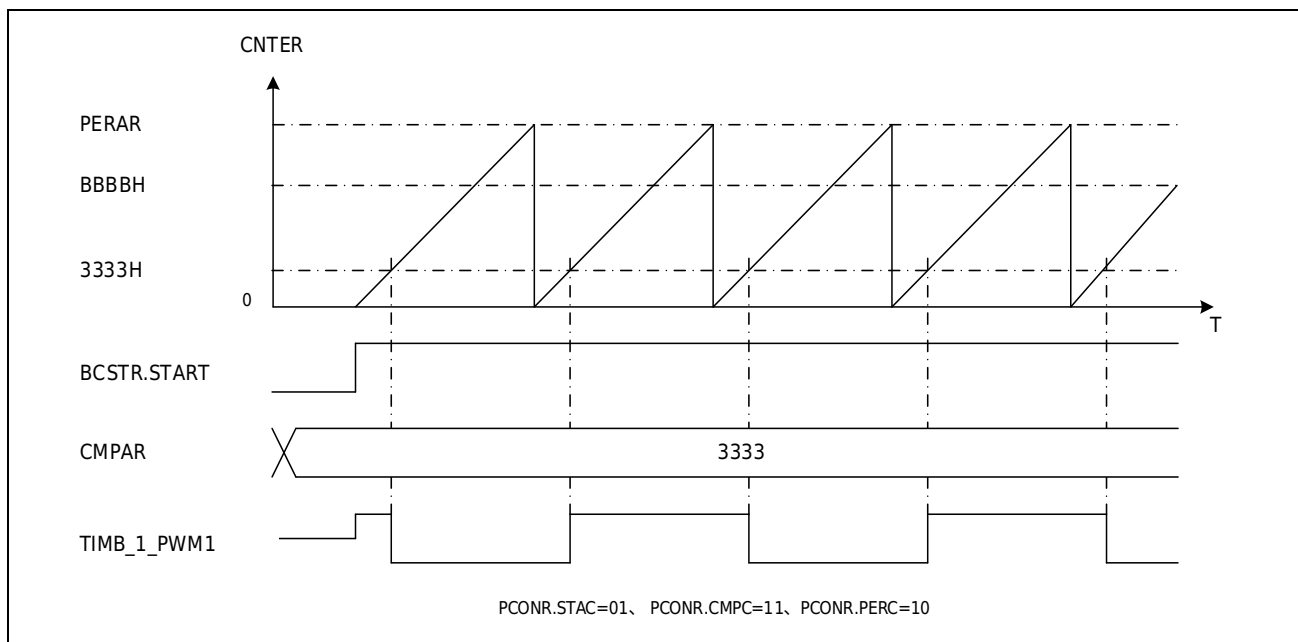


图 14-4 比较输出动作

TMRB_<t>_PWM1 端口的计数开始时的电平、计数停止时的电平、计数比较匹配时的电平、计数周期匹配时的电平，可通过端口控制寄存器（PCONR）的 STAC、STPC、CMPC、PERC、FORC 位设定控制。图 14-4 为单元 1 的比较输出动作例。

14.3.1.3 捕获输入

各个 TimerB 单元的每个 PWM 输出通道都具有捕获输入功能，用于保存捕获到的计数值。设定捕获控制寄存器（CCONR）的 CCONR.CAPMD 位为 1，捕获输入功能变为有效。此时选择对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器（CMPAR）中。

捕获输入条件可以选择内部动作触发事件（通过 TMRB_TGRSEL 寄存器选择）、TMRB_<t>_PWM1 端口输入等，具体的条件选择可通过捕获控制寄存器（CCONR）的 HICP 位来设定。图 14-5 为单元 1 的捕获输入动作例。

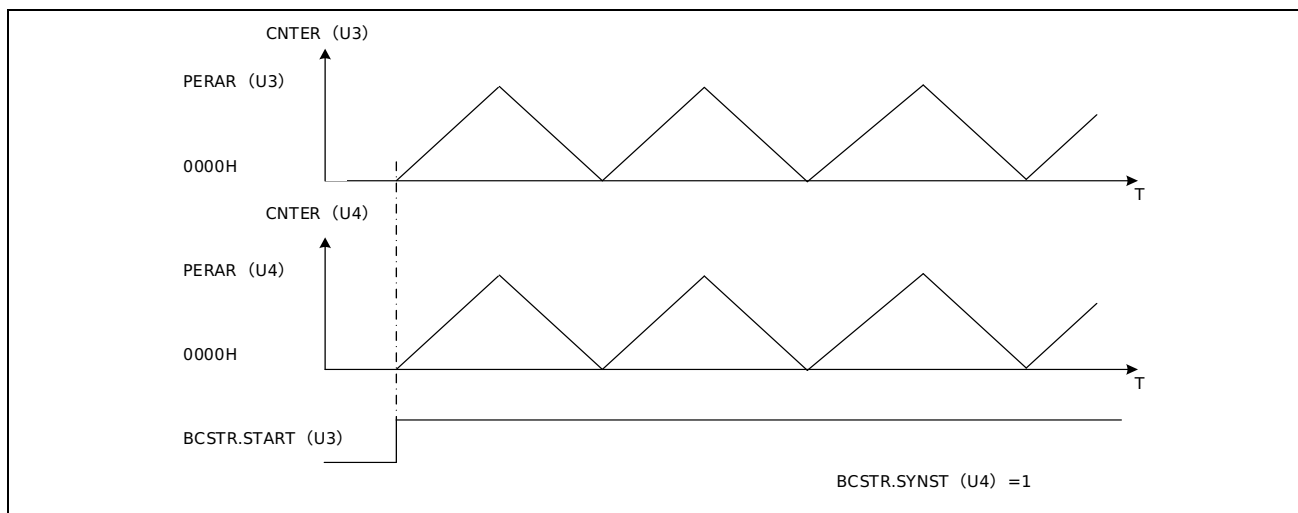


图 14-6 软件同步动作

14.3.4 数字滤波

各个单元的 TMRB_<t>_PWM1 端口输入具有数字滤波功能。该滤波功能的使能和滤波时钟的选择可通过设定捕获控制寄存器 (CCONR) 的对应位来实现。

在滤波采样基准时钟采样到端口上 3 次一致的电平时, 该电平被当作有效电平传送到模块内部; 小于 3 次一致的电平会被当作外部干扰滤掉, 不传送到模块内部。图 14-7 所示为 TMRB_1_PWM1 端口滤波动作例。

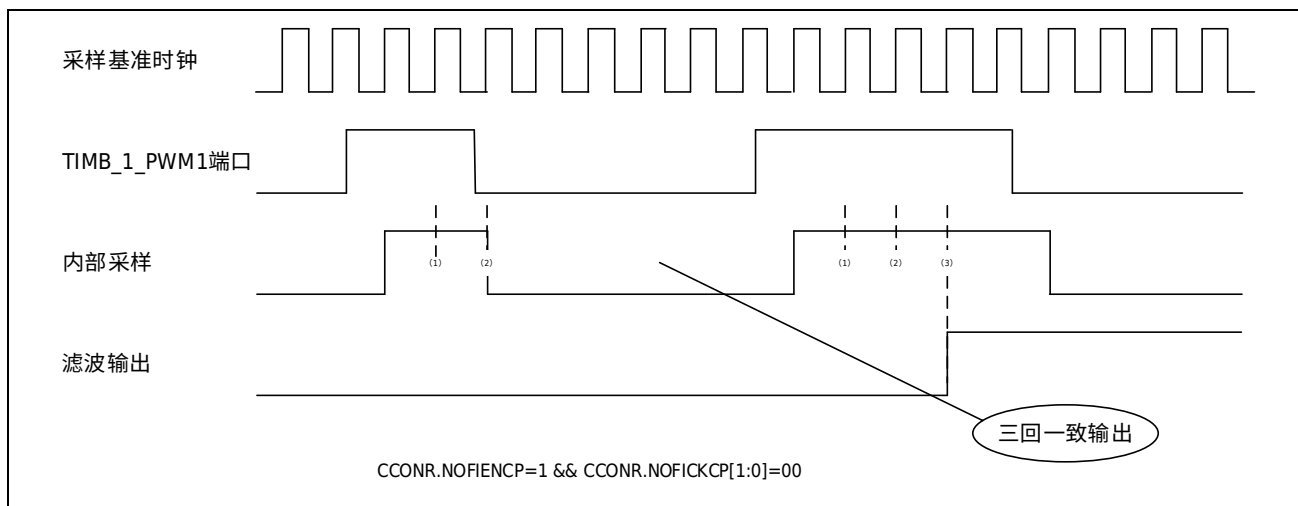


图 14-7 PWM 输入端口的滤波功能

14.3.5 级联计数

在计数时钟源选择章节中, 当时钟源选择 d) 时, 本单元的计数时钟就选择为对称单元的计数溢出 (计数上溢或计数下溢) 事件, 此时两个单元级联合并可实现 32 位计数器。在级联计数中, 对称单元的 CNTER 为低 16 位计数器, 本单元的 CNTER 为高 16 位计数器。

例如，在三角波向上计数模式 (BCSTR.MODE=0、BCSTR.DIR=1) 时，设定单元 1 的计数时钟为 HCLK，设定单元 2 的计数时钟源为单元 1 的计数上溢事件（单元 2 的 TMRB_HCUPR.HCUP11=1），启动单元 1、2 计数（先启动单元 2，再启动单元 1）就实现级联计数。单元 1 的 CNTER 位低 16 位计数器，单元 2 的 CNTER 为高 16 位计数器。如图 14-8 所示是单元 1、2 级联计数的示意图。

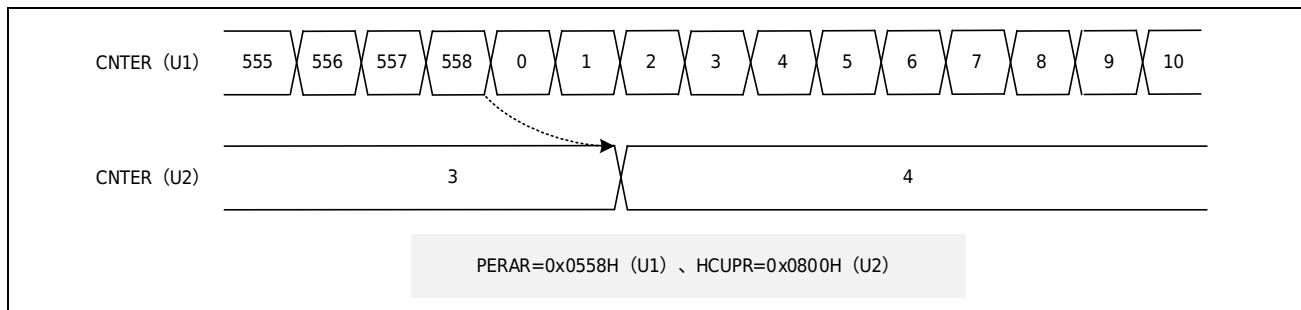


图 14-8 32 位级联计数动作

14.3.6 脉宽测量

将 TimerB 的硬件启动条件设为 TMRB_<t>_PWM1 的上升沿，硬件清零条件、停止条件和捕获输入条件均设为 TMRB_<t>_PWM1 的下降沿，就可以实现连续的脉冲宽度测量。对应动作如图 14-9 所示。

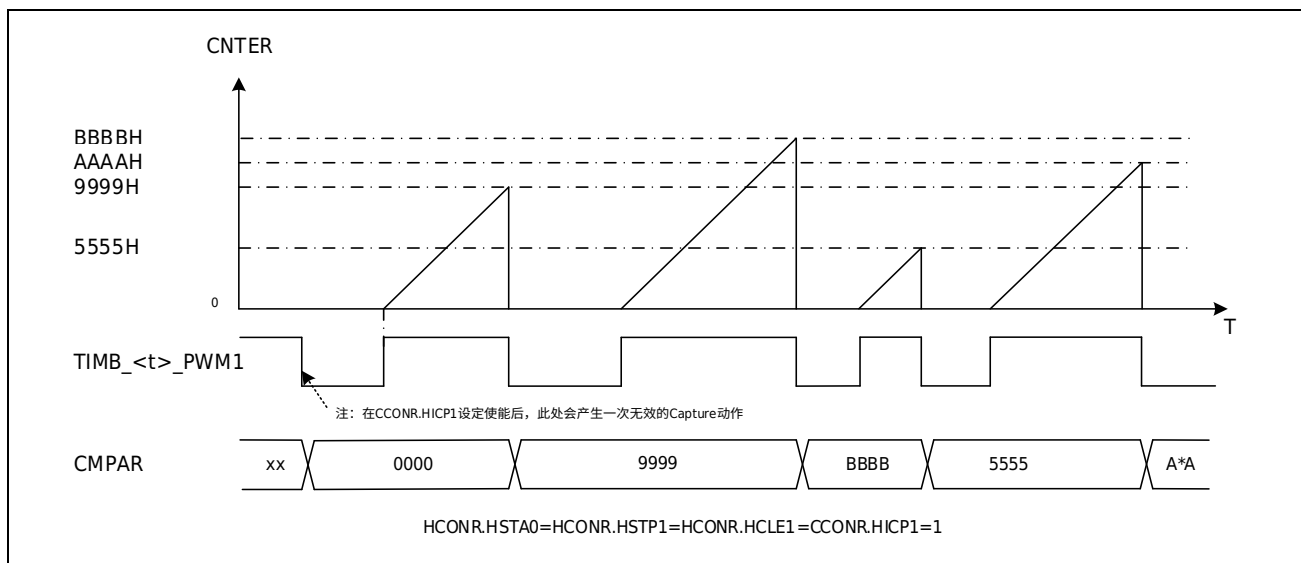


图 14-9 脉宽测量

14.3.7 周期测量

将 TimerB 的硬件启动条件、硬件清零条件和捕获输入条件设为 TMRB_<t>_PWM1 的相同沿（上升沿或下降沿），就可以实现连续的周期宽度测量。对应动作如图 14-10 所示。

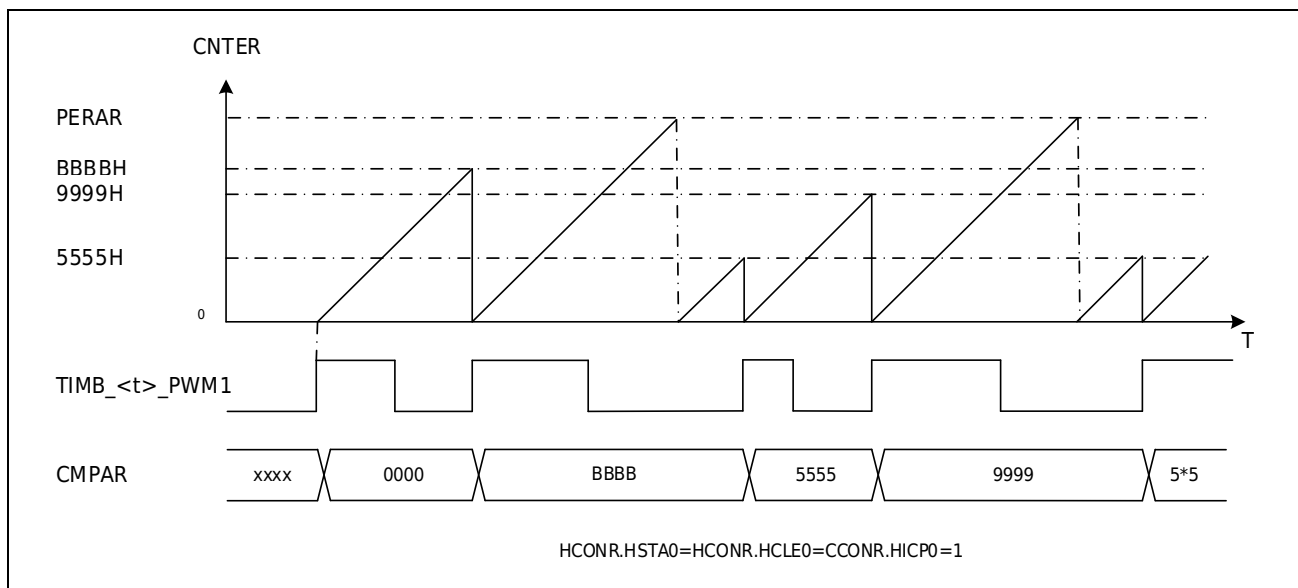


图 14-10 周期测量

14.3.8 PWM 输出

14.3.8.1 单边对齐 PWM 输出

锯齿波计数模式时，输出通道通过各种端口控制设定，可以实现单边对齐 PWM 输出（在计数周期点对齐）。例如，设定在比较基准值比较匹配时翻转（PCONR.CMPC=11）、在周期基准值比较匹配时翻转（PCONR.PERC=11），就可实现在一个周期内产生单边对齐的 PWM 输出。

图 14-11 所示，是单元 1 锯齿波模式下，单边对齐 PWM 输出波形例。

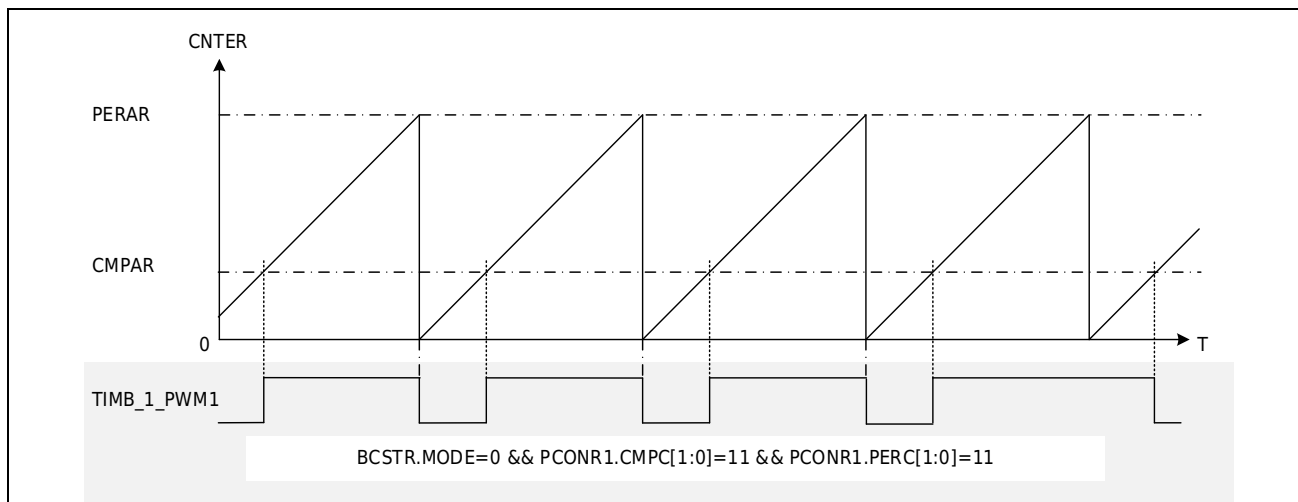


图 14-11 单边对齐 PWM 输出

14.3.8.2 双边对称 PWM 输出

三角波计数模式时，输出通道通过各种端口控制设定，可以实现双边对称 PWM 输出（以计数峰点对称）。例如，设定在比较基准值比较匹配时翻转（PCONR.CMPC=11）、在周期基准值比较匹配时保持（PCONR.PERC=10），就可实现在一个周期内产生双边对称的 PWM 输出。

图 14-12 所示，是三角波模式下的双边对称 PWM 输出波形例。

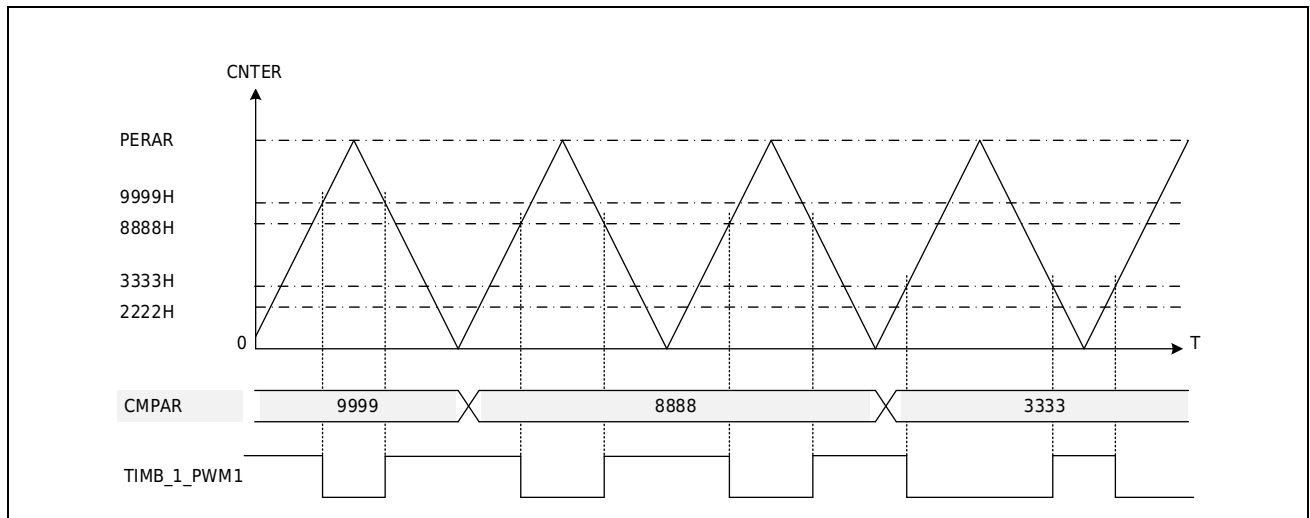


图 14-12 双边对称 PWM 输出

14.4 中断及事件说明

TimerB 含有 3 个中断输出和 3 个事件输出，分别是 1 个比较匹配中断和事件、2 个周期匹配中断和事件。

14.4.1 比较匹配中断及事件

比较基准值寄存器 (CMPAR) 与计数值比较发生比较匹配时，状态标志寄存器 (STFLR) 中的对应位 (STFLR.CMPF) 会被置为 1。此时，若中断控制寄存器 (ICONR) 的对应位 (ICONR.ITEN) 设定为 1，则对应的中断请求 (TMRB_<t>_CMP) 会被触发；若事件控制寄存器 (ECONR) 的对应位 (ECONR.ETEN) 设定为 1，则对应的事件请求 (TMRB_<t>_CMP) 会被触发。

捕获控制寄存器 (CCONR) 选择的捕获输入有效条件产生时，捕获输入动作发生。此时，若中断控制寄存器 (ICONR) 的对应位 (ICONR.ITEN) 设定为 1，则对应的中断请求 (TMRB_<t>_CMP) 被触发；若事件控制寄存器 (ECONR) 的对应位 (ECONR.ETEN) 设定为 1，则对应的事件请求 (TMRB_<t>_CMP) 会被触发。

14.4.2 周期匹配中断及事件

锯齿波模式递加计数至上溢点、锯齿波模式递减计数至下溢点、三角波模式计数至谷点或峰点，控制状态寄存器 (BCSTR) 的 OVFF 或 UDFF 位会被置为 1。此时，若 BCSTR.ITENOVF 或 BCSTR.ITENUDF 位设定为 1 使能中断，则在对应的周期点可触发周期匹配中断 (TMRB_<t>_OVF 和 TMRB_<t>_UDF) 输出至中断模块 (INTC)；周期匹配事件则无对应的使能位控制，在对应的计数周期点就触发周期匹配事件 (TMRB_<t>_OVF 和 TMRB_<t>_UDF) 输出用于选择触发别的模块。

14.5 寄存器描述

表 14-3 TimerB 基地址

名称	基地址	描述
TimerB_1	0x4000 7800	TimerB_1 基地址
TimerB_2	0x4000 7C00	TimerB_2 基地址
TimerB_3	0x4000 8000	TimerB_3 基地址
TimerB_4	0x4000 8400	TimerB_4 基地址
TimerB_5	0x4000 8800	TimerB_5 基地址
TimerB_6	0x4000 8C00	TimerB_6 基地址
TimerB_7	0x4000 9000	TimerB_7 基地址
TimerB_8	0x4000 9400	TimerB_9 基地址

表 14-4 TimerB 寄存器列表

寄存器	描述	偏移量	位宽	复位值
TMRB_CNTER	通用计数值寄存器	0x0000	16	0x0000
TMRB_PERAR	周期基准值寄存器	0x0004	16	0xFFFF
TMRB_CMPAR	比较基准值寄存器	0x0040	16	0xFFFF
TMRB_BCSTR_L	控制状态寄存器 L	0x0080	8	0x02
TMRB_BCSTR_H	控制状态寄存器 H	0x0081	8	0x00
TMRB_ICONR	中断控制寄存器	0x0090	16	0x0000
TMRB_ECONR	事件控制寄存器	0x0094	16	0x0000
TMRB_STFLR	状态标志寄存器	0x009C	16	0x0000
TMRB_CCONR	捕获控制寄存器	0x0100	16	0x0000
TMRB_PCONR	端口控制寄存器	0x0140	16	0x0000
TMRB_HCONR	硬件触发事件选择寄存器	0x0084	16	0x0000
TMRB_HCUPR	硬件递加事件选择寄存器	0x0088	16	0x0000
TMRB_HCDOR	硬件递减事件选择寄存器	0x008C	16	0x0000
TMRB_TGRSEL	内部触发事件选择寄存器	(0x4000 0C50)	32	0x0000 007F

注：

- 内部触发事件选择寄存器（TMRB_TGRSEL）是一个独立的寄存器，为 8 个单元的 TimerB 所共有。

14.5.1 通用计数值寄存器 (TMRB_CNTER)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位/位域	标记	位名	功能	读写
b15~b0	CNT[15:0]	计数值	当前定时器的计数值	RW

14.5.2 周期基准值寄存器 (TMRB_PERAR)

复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															

位/位域	标记	位名	功能	读写
b15~b0	PER[15:0]	计数周期值	设定每轮计数的计数周期值	RW

14.5.3 比较基准值寄存器 (TMRB_CMPAR)

复位值: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															

位/位域	标记	位名	功能	读写
b15~b0	CMP[15:0]	计数比较基准值	设定比较基准值	RW

14.5.4 控制状态寄存器 L (TMRB_BCSTRL)

复位值：0x02

b7	b6	b5	b4	b3	b2	b1	b0
CKDIV[3:0]				SYNST	MODE	DIR	START

位/位域	标记	位名	功能	读写
b7~b4	CKDIV[3:0]	计数时钟选择	0000: HCLK	RW
			0001: HCLK/2	
			0010: HCLK/4	
			0011: HCLK/8	
			0100: HCLK/16	
			0101: HCLK/32	
			0110: HCLK/64	
			0111: HCLK/128	
			1000: HCLK/256	
			1001: HCLK/512	
			1010: HCLK/1024	
b3	SYNST	同步启动使能	0: 本单元为单元 m 时, 与单元 n 的同步启动功能无效 1: 本单元为单元 m 时, 与单元 n 的同步启动功能有效 (当 m=2、4、6、8 时, n=1、3、5、7) 注: 单元 n 的该位设定无效, 读出时为 0	RW
b2	MODE	计数模式	0: 锯齿波模式 1: 三角波模式	RW
b1	DIR	计数方向	0: 计数器向下计数 1: 计数器向上计数 注: 定时器启动计数期间写无效。	RW
b0	START	定时器启动	0: 定时器关闭 1: 定时器启动 注 1: 该位在硬件停止条件有效时, 会自动变为 0 注 2: 本单元 m 的同步启动功能有效时, 单元 n 软件启动后本单元的该位也会被置位 (当 m=2、4、6、8 时, n=1、3、5、7)	RW

14.5.5 控制状态寄存器 H (TMRB_BCSTRH)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
UDFF	OVFF	ITENUDF	ITENOVF	Res			OVSTP

位/位域	标记	位名	功能	读写
b7	UDFF	下溢标志	0: 向下计数时, 未发生计数下溢 1: 向下计数时, 发生计数下溢	RW
b6	OVFF	上溢标志	0: 向上计数时, 未发生计数上溢 1: 向上计数时, 发生计数上溢	RW
b5	ITENUDF	下溢中断使能	0: 计数下溢中断不使能 1: 计数下溢中断使能	RW
b4	ITENOVF	上溢中断使能	0: 计数上溢中断不使能 1: 计数上溢中断使能	RW
b3~b1	Res	保留位	读出时为 0, 写入时写 0	RW
b0	OVSTP	计数溢出停止控制	0: 计数器在计数上溢或下溢后, 继续计数 1: 计数器在计数上溢或下溢后, 停止计数	RW

14.5.6 中断控制寄存器 (TMRB_ICONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															ITEN1

位/位域	标记	位名	功能	读写
b15~b1	Res	保留位	读出时为“0”, 写入时写“0”	RW
b0	ITEN1	计数匹配中断使能 1	0: CMPAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能	RW

14.5.7 事件控制寄存器 (TMRB_ECONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															ET EN1

位/位域	标记	位名	功能	读写
b15~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	ETEN1	计数匹配事件使能 1	0: CMPAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出无效 1: CMPAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该事件输出使能	RW

14.5.8 状态标志寄存器 (TMRB_STFLR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															CMP F1

位/位域	标记	位名	功能	读写
b15-b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	CMPF1	计数匹配标志 1	0: CMPAR 寄存器的值与计数值不相等, 且未发生 TMRB_<t>_PWM1 捕获完成动作 1: CMPAR 寄存器的值与计数值相等, 或发生 TMRB_<t>_PWM1 捕获完成动作	RW

14.5.9 捕获控制寄存器 (TMRB_CCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFI CKCP[1:0]		NOFI ENCP	Res					HICP 2	HICP 1	HICP 0	Res		CAP MD	

位/位域	标记	位名	功能	读写
b15	Res	保留位	读出时为“0”,写入时写“0”	RW
b14~b13	NOFICKCP[1:0]	滤波采样基准时钟 选择 CP	00: HCLK 01: HCLK/4 10: HCLK/16 11: HCLK/64	RW
b12	NOFIENCP	捕获输入端口滤波 CP	0: TMRB_<t>_PWM1 端口输入滤波功能无效 1: TMRB_<t>_PWM1 端口输入滤波功能使能	RW
b11~b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6	HICP2	捕获输入条件使能 2	0: TMRB_TGRSEL 寄存器中指定的事件发生时, 不发生捕获输入动作 1: TMRB_TGRSEL 寄存器中指定的事件发生时, 产生捕获输入动作	RW
b5	HICP1	捕获输入条件使能 1	0: TMRB_<t>_PWM1 端口输入采样到下降沿时, 不发生捕获输入动作 1: TMRB_<t>_PWM1 端口输入采样到下降沿时, 产生捕获输入动作	RW
b4	HICP0	捕获输入条件使能 0	0: TMRB_<t>_PWM1 端口输入采样到上升沿时, 不发生捕获输入动作 1: TMRB_<t>_PWM1 端口输入采样到上升沿时, 产生捕获输入动作	RW
b3~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	CAPMD	功能模式选择	0: 比较输出功能 1: 捕获输入功能	RW

14.5.10 端口控制寄存器 (TMRB_PCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			OUTEN	Res		FORC[1:0]		PERC[1:0]		CMPC[1:0]		STPC[1:0]		STAC[1:0]	

位/位域	标记	位名	功能	读写
b15~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	OUTEN	输出使能	0: PWM 输出功能时的 TMRB_<t>_PWM1 端口输出无效 1: PWM 输出功能时的 TMRB_<t>_PWM1 端口输出有效	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9~b8	FORC[1:0]	强制端口状态设定	0x: 设定无效定 10: 下周期开始, TMRB_<t>_PWM1 端口输出设定为低电平 11: 下周期开始, TMRB_<t>_PWM1 端口输出设定为高电平 注 1: 下周期是指硬件计数模式或锯齿波计数到上溢点或下溢点、三角波计数到谷点 注 2: 该寄存器位可用于实现 PWM 输出占空比 0%或 100%的控制	RW
b7~b6	PERC[1:0]	周期值匹配时端口状态设定	00: 计数值与 PERAR 相等时, TMRB_<t>_PWM1 端口输出设定为低电平 01: 计数值与 PERAR 相等时, TMRB_<t>_PWM1 端口输出设定为高电平 10: 计数值与 PERAR 相等时, TMRB_<t>_PWM1 端口输出保持先前状态 11: 计数值与 PERAR 相等时, TMRB_<t>_PWM1 端口输出设定为反转电平	RW
b5~b4	CMPC[1:0]	比较值匹配时端口状态设定	00: 计数值与 CMPARn 相等时, TMRB_<t>_PWM1 端口输出设定为低电平 01: 计数值与 CMPARn 相等时, TMRB_<t>_PWM1 端口输出设定为高电平 10: 计数值与 CMPARn 相等时, TMRB_<t>_PWM1 端口输出保持先前状态 11: 计数值与 CMPARn 相等时, TMRB_<t>_PWM1 端口输出设定为反转电平	RW
b3~b2	STPC[1:0]	计数停止时端口状态设定	00: 计数停止时, TMRB_<t>_PWM1 端口输出设定为低电平 01: 计数停止时, TMRB_<t>_PWM1 端口输出设定为高电平 10: 计数停止时, TMRB_<t>_PWM1 端口输出保持先前状态 11: 计数停止时, TMRB_<t>_PWM1 端口输出保持先前状态	RW
b1~b0	STAC[1:0]	计数开始时端口状态设定	00: 计数开始时, TMRB_<t>_PWM1 端口输出设定为低电平 01: 计数开始时, TMRB_<t>_PWM1 端口输出设定为高电平 10: 计数开始时, TMRB_<t>_PWM1 端口输出保持先前状态 11: 计数开始时, TMRB_<t>_PWM1 端口输出保持先前状态	RW

14.5.11 硬件触发事件选择寄存器 (TMRB_HCONR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	HCLE 4	HCLE 3	Res	HCLE 2	HCLE 1	HCLE 0	Res	HSTP 2	HSTP 1	HSTP 0	Res	HSTA 2	HSTA 1	HSTA 0	

位/位域	标记	位名	功能	读写
b15~b14	Res	保留位	读出时为“0”,写入时写“0”	RW
b13	HCLE4	硬件清零条件 4	条件: 本单元为单元 m 时, 单元 n 的 TMRB_n_PWM1 端口输入采样到下降沿 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
b12	HCLE3	硬件清零条件 3	条件: 本单元为单元 m 时, 单元 n 的 TMRB_n_PWM1 端口输入采样到上升沿 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10	HCLE2	硬件清零条件 2	条件: TMRB_TGRSEL 寄存器中指定的事件发生 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
b9	HCLE1	硬件清零条件 1	条件: TMRB_<t>_PWM1 端口输入采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
b8	HCLE0	硬件清零条件 0	条件: TMRB_<t>_PWM1 端口输入采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效	RW
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6	HSTP2	硬件停止条件 2	条件: TMRB_TGRSEL 寄存器中指定的事件发生 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	RW
b5	HSTP1	硬件停止条件 1	条件: TMRB_<t>_PWM1 端口输入采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	RW
b4	HSTP0	硬件停止条件 0	条件: TMRB_<t>_PWM1 端口输入采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效	RW
b3	Res	保留位	读出时为“0”,写入时写“0”	RW
b2	HSTA2	硬件启动条件 2	条件: TMRB_TGRSEL 寄存器中指定的事件发生 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效	RW
b1	HSTA1	硬件启动条件 1	条件: 1) 本单元 TMRB_<t>_PWM1 端口输入采样到下降沿 (同步启动功能无效) 2) TMRB_n_PWM1 端口输入采样到下降沿 (同步启动功能有效)	RW

位/位域	标记	位名	功能	读写
			0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效 注: 条件 2) 中, 当本单元为 2、4、6、8 时, n=1、3、5、7; 当本单元为 1、3、5、7 时, 该功能无效	
b0	HSTA0	硬件启动条件 0	条件: 1) 本单元 TMRB_<t>_PWM1 端口输入采样到上升沿 (同步启动功能无效) 2) TMRB_n_PWM1 端口输入采样到上升沿 (同步启动功能有效) 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效 注: 条件 2) 中, 当本单元为 2、4、6、8 时, n=1、3、5、7; 当本单元为 1、3、5、7 时, 该功能无效	RW

14.5.12 硬件递加事件选择寄存器 (TMRB_HCUPR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			HC UP12	HC UP11	HC UP10	HC UP9	HC UP8	Res							

位/位域	标记	位名	功能	读写
b15~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	HCUP12	硬件递加条件 12	条件: 本单元为单元 m 时, 单元 n 发生计数下溢 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	RW
b11	HCUP11	硬件递加条件 11	条件: 本单元为单元 m 时, 单元 n 发生计数上溢 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	RW
b10	HCUP10	硬件递加条件 10	条件: TMRB_TGRSEL 寄存器中指定的事件发生 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	RW
b9	HCUP9	硬件递加条件 9	条件: TMRB_<t>_PWM1 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	RW
b8	HCUP8	硬件递加条件 8	条件: TMRB_<t>_PWM1 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效	RW
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

14.5.13 硬件递减事件选择寄存器 (TMRB_HCDOR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			HC DO12	HC DO11	HC DO10	HC DO9	HC DO8	Res							

位/位域	标记	位名	功能	读写
b15~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	HCDO12	硬件递减条件 12	条件: 本单元为单元 m 时, 单元 n 发生计数下溢 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	RW
b11	HCDO11	硬件递减条件 11	条件: 本单元为单元 m 时, 单元 n 发生计数上溢 (当 m=1、3、5、7 时, n=2、4、6、8; 当 m=2、4、6、8 时, n=1、3、5、7) 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	RW
b10	HCDO10	硬件递减条件 10	条件: TMRB_TGRSEL 寄存器中指定的事件发生 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	RW
b9	HCDO9	硬件递减条件 9	条件: TMRB_<t>_PWM1 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	RW
b8	HCDO8	硬件递减条件 8	条件: TMRB_<t>_PWM1 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效	RW
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

14.5.14 内部触发事件选择寄存器 (TMRB_TGRSEL)

复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									TRGSEL[6:0]						

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”,写入时写“0”	R
b6~b0	TRGSEL	内部触发事件触发源选择	内部触发事件触发源编号写入, 参考 INTC 章节	RW

15 通用定时器 (Timer0)

15.1 概述

通用定时器 0 (Timer0) 是一个可以实现同步计数、异步计数两种方式的基本定时器。该定时器内含 1 个通道, 可以在计数期间产生比较匹配事件。该事件可以触发中断, 也可作为事件输出来控制其它模块等。本系列产品中搭载 1 个单元的 Timer0。

15.2 基本框图

Timer0 的基本框图如图 15-1 所示。

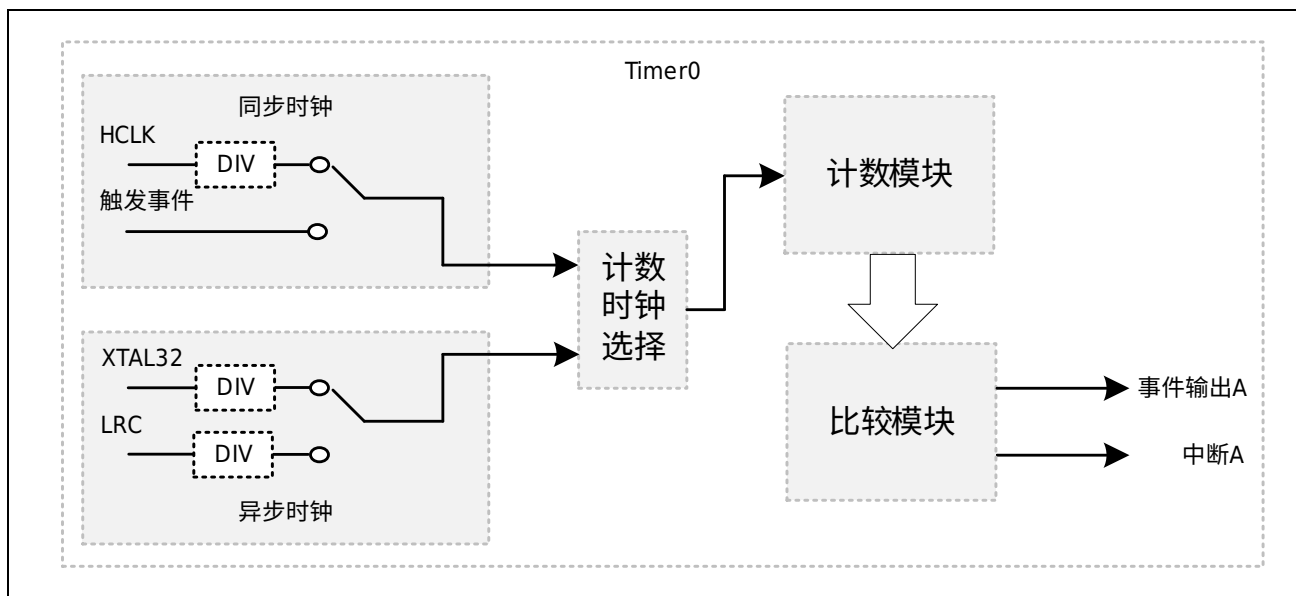


图 15-1 Timer0 基本框图

15.3 功能说明

15.3.1 时钟源选择

Timer0 的计数方式可以选择同步计数方式或异步计数方式。

同步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）有同步时序关系；异步计数方式是指定时器的计数时钟和总线访问时钟（寄存器读写操作时钟）是非同步的时序关系。异步计数方式对寄存器读操作时，定时器等状态可能正在发生变化、读出不可预期的状态。因此，在异步计数方式时，寄存器读操作须在计数停止状态下实现。

15.3.1.1 同步计数时钟源

同步计数方式时（BCONR.SYNSA=0），时钟源可以有以下几种选择（BCONR.SYNCLKA 设定选择）：

- a) HCLK 及 HCLK 的 2、4、8、16、32、64、128、256、512、1024 分频作为同步计数时钟（BCONR.SYNCLKA=0 & BCONR.CKDIVA[3:0]设定）
- b) 内部硬件触发事件输入作为同步计数时钟（BCONR.SYNCLKA=1）

15.3.1.2 异步计数时钟源

异步计数方式时（BCONR.SYNSA=1），时钟源可以有以下几种选择（BCONR.ASYNCLKA 设定选择）：

- a) LRC 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLKA=0 & BCONR.CKDIVA[3:0]设定）
- b) XTAL32 时钟源输入及其 2、4、8、16、32、64、128、256、512、1024 分频作为异步计数时钟（BCONR.ASYNCLKA=1 & BCONR.CKDIVA[3:0]设定）

15.3.2 基本计数动作

Timer0 可设定基准计数值，在计数值和基准值相等时产生计数比较匹配事件。如图 15-2 所示。

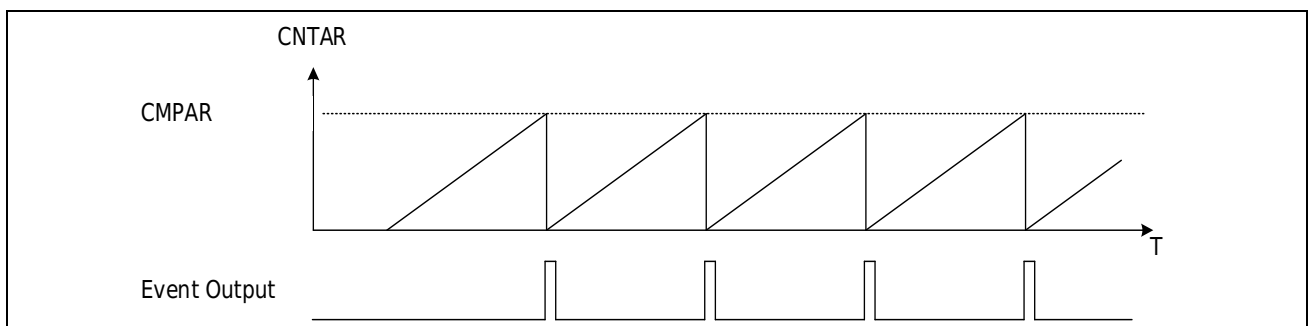


图 15-2 Timer0 计数时序图

15.3.3 硬件触发动作

Timer0 有一个共用的内部硬件触发源，可以通过基本控制寄存器（BCONR）的相关设定，来控制定时器的状态（计数、启动、停止、清零）以及捕获输入动作等。

该硬件触发源的源选择通过向硬件触发选择寄存器（TGRSEL）中输入对应的编号来实现，具体的事件对应关系请参考 INTC 章节。使用内部硬件触发功能时，需要先将功能时钟控制寄存器（FCG）的外围电路触发功能位清零。

15.4 中断及事件说明

15.4.1 中断输出

一个 Timer0 含有 1 个计数比较匹配中断或输入捕获中断。

基准值寄存器（CMPAR）与计数值寄存器（CNTAR）比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMFA 位会被置为 1。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA 位使能中断，则对应的中断请求（TMR0_GCMP）也会被触发。

在内部硬件触发输入作为捕获输入条件时，可以产生相应的捕获输入动作。此时若设定基本控制寄存器（BCONR）的 BCONR.INTENA 位使能中断，则对应的中断请求（TMR0_GCMP）被触发。

在选择异步计数模式时，基准值寄存器（CMPAR）产生的比较匹配中断可用于在低功耗模式时对系统进行唤醒，具体请参考【10 中断控制器（INTC）】章节。

15.4.2 事件输出

在计数过程中发生计数比较匹配时，会分别产生相应的事件请求（TMR0_GCMP）输出信号，可以用于选择触发别的模块。

15.5 寄存器描述

表 15-1 Timer0 基地址

名称	基地址	描述
Timer0	0x4000 5800	Timer0基地址

表 15-2 Timer0 寄存器列表

寄存器	描述	偏移量	位宽	复位值
TMRO_CNTAR	计数值寄存器	0x0000	32	0x0000 0000
TMRO_CMPAR	基准值寄存器	0x0008	32	0x0000 FFFF
TMRO_BCONR	基本控制寄存器	0x0010	32	0x0000 0000
TMRO_TGRSEL	触发选择寄存器	(0x4000 0C20)	32	0x0000 007F
TMRO_STFLR	状态标志寄存器	0x0014	32	0x0000 0000

15.5.1 计数值寄存器 (TMRO_CNTAR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA[15:0]															

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”	R
b15~b0	CNTA[15:0]	计数值 A	当前定时器的计数值 A	RW

15.5.2 基准值寄存器 (TMRO_CMPAR)

复位值: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA[15:0]															

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”	R
b15~b0	CMPA[15:0]	基准值 A	设定计数基准值 A，产生 Compare Match 事件	RW

15.5.3 基本控制寄存器 (TMR0_BCONR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICP A	HCLE A	HSTP A	HSTA A	Res	ASYN CLKA	SYN CLKA	SYN SA	CKDIV A[3:0]				Res	INT ENA	CAP MDA	CST A

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15	HICPA	硬件触发输入捕获 A	条件: 内部硬件触发事件有效 0: 条件匹配时, 捕获输入无效 1: 条件匹配时, 捕获输入有效	RW
b14	HCLEA	硬件触发清零 A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器清零无效 1: 条件匹配时, 定时器清零有效	RW
b13	HSTPA	硬件触发停止 A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器停止无效 1: 条件匹配时, 定时器停止有效	RW
b12	HSTAA	硬件触发启动 A	条件: 内部硬件触发事件有效 0: 条件匹配时, 定时器启动无效 1: 条件匹配时, 定时器启动有效	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	RW
b10	ASYNCLKA	通道 A 异步计数时钟源选择	0: LRC 1: XTAL32	RW
b9	SYNCLKA	通道 A 同步计数时钟源选择	0: HCLK 1: 内部硬件触发事件	RW
b8	SYNSA	通道 A 计数方式选择	0: 同步计数方式 1: 异步计数方式	RW
b7~b4	CKDIVA[3:0]	通道 A 计数时钟分频选择	通道 A 计数时钟分频选择: 0000: 时钟源 0001: 时钟源/2 0010: 时钟源/4 0011: 时钟源/8 0100: 时钟源/16 0101: 时钟源/32 0110: 时钟源/64 0111: 时钟源/128 1000: 时钟源/256 1001: 时钟源/512 1010: 时钟源/1024 请不要设定其它值 注: 被分频的时钟源可以是异步计数时的各种时钟源、同步计数时的 HCLK	RW
b3	Res	保留位	读出时为“0”,写入时写“0”	RW

位/位域	标记	位名	功能	读写
b2	INTENA	计数匹配中断使能 A	0: CMPAR 寄存器与计数值 (CNTAR) 相等时, 或者发生捕获输入事件时, 该中断无效 1: CMPAR 寄存器与计数值 (CNTAR) 相等时, 或者发生捕获输入事件时, 该中断使能	RW
b1	CAPMDA	功能模式选择 A	0: 比较输出功能 1: 捕获输入功能	RW
b0	CSTA	定时器启动	0: 通道 A 定时器关闭 1: 通道 A 定时器启动 注: 该位在硬件触发停止条件有效时, 会自动变为 0	RW

15.5.4 触发选择寄存器 (TMRO_TGRSEL)

复位值: 0x0000 007F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									TRGSEL[6:0]						

位/位域	标记	位名	功能	读写
b31~b7	Res	保留位	读出时为“0”,写入时写“0”	R
b6~b0	TRGSEL	触发源选择	触发源编号写入, 参考 INTC 章节	RW

15.5.5 状态标志寄存器 (TMRO_STFLR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															CMFA

位/位域	标记	位名	功能	读写
b31~b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	CMFA	计数匹配 A	0: CMPAR 寄存器的值与计数值 (CNTAR) 不相等且未发生捕获输入动作 1: CMPAR 寄存器的值与计数值 (CNTAR) 相等或发生捕获输入动作	RW

15.6 使用注意事项

1. 在异步计数动作时，需先设定 BCONR.ASYNCLKA 位选择异步时钟源,再设定 BCONR.SYNSA 位选择异步计数方式，然后再启动 Timer0。
2. 在选择异步计数的情况下，修改计数值(CNTAR)、基准值 (CMPAR)、启动位 (BCONR.CSTA)、状态位 (STFLR.CMFA) 时，Timer0 从接收到写动作后经过 6 个异步计数时钟才将修改值写入对应的寄存器中。
3. 在选择异步计数的情况下，连续对计数值(CNTAR)、基准值 (CMPAR)、启动位 (BCONR.CSTA)、状态位 (STFLR.CMFA) 进行写动作时，需间隔至少 6 个异步计数时钟。
4. 在选择异步计数的情况下，请将 BCONR.SYNCLKA 设定为 0。
5. 对 BCONR.CSTA 写入时，请勿同时改写 BCONR.SYNSA。

16 实时时钟（RTC）

16.1 概述

实时时钟（RTC）是一个以 BCD 码格式保存时间信息的计数器。记录从 00 年到 99 年间的具体日历时间。支持 12/24 小时两种时制，根据月份和年份自动计算日数 28、29（闰年）、30 和 31 日。表 16-1 所示是其基本特性。

表 16-1 RTC 的基本规格

计数时钟源	外部低速振荡器（32.768KHz）
基本功能	BCD 码表示秒、分、时、日、周、月、年时间
	软件启动或停止
	12/24 时制可选、闰年自动识别
	可编程闹钟
	分布式补偿 1Hz 时钟输出
	时钟误差补偿功能
中断	定周期中断
	闹钟中断

16.2 基本框图

RTC 的基本框图如图 16-1 所示。

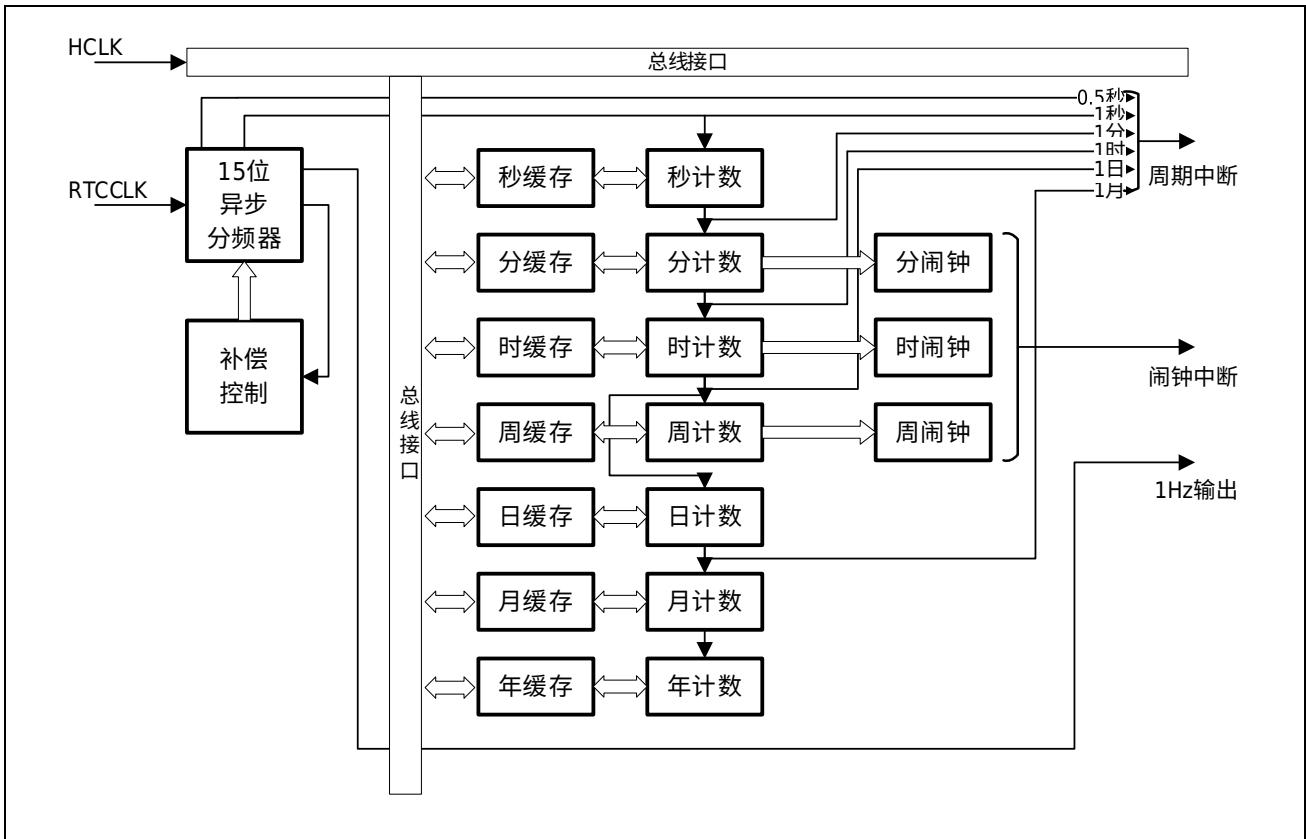


图 16-1 RTC 的基本框图

16.3 功能说明

16.3.1 上电设定

RTC 寄存器通过上电复位和设定控制寄存器 RESET 位复位所有寄存器，在设定控制寄存器，日历初始值、闹钟设置等之后，启动 RTC。RTC 启动之后，其他外部各种复位请求都不能复位 RTC，RTC 会一直处于工作状态。可以设定控制寄存器的 START 位为“0”停止 RTC 工作。RTC 在进行设定时，必须保证时钟源已稳定。

16.3.2 RTC 计数开始设定

1. 上电后，除 RTC_CR0 以外的所有寄存器复位；也可以设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位所有寄存器；
2. 设定 RTC_CR1.START=0，停止计数；
3. 设定系统时钟寄存器，打开副发振器，再设定 RTC_CR3，选择 RTC 计数时钟源；
4. 设定 RTC_CR1，设定时制、周期、1Hz 时钟输出；
5. 设定秒，分，时，周，日，月，年的日历计数寄存器；
6. 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 RTC_ERRCRL，RTC_ERRCRH；
7. 清除寄存器 RTC_CR1，RTC_TPSR 中的标志寄存器位，并使能中断；
8. 设定 RTC_CR1.START=1，计数开始。

16.3.3 系统低功耗模式切换

在 RTC 计数开始后，系统立即切换为低功耗模式时，请执行下列任何一种确认后再进行模式切换。

1. 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
2. 在 RTC_CR1.START=1 设定后，设定 RTC_CR2.RWREQ=1，查询 RTC_CR2.RWEN=1。设定日历计数寄存器，再设定 RTC_CR2.RWREQ=0，查询 RTC_CR2.RWEN=0，关闭总线时钟。进行模式切换。

16.3.4 读出计数寄存器

1. 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC_CR2.RWREQ=1，停止日历寄存器计数，进入读写模式；
2. 查询直到 RTC_CR2.RWEN=1；

3. 读出秒，分，时，周，日，月，年计数寄存器值；
4. 设定 RTC_CR2.RWREQ=0，计数器计数；
5. 查询直到 RTC_CR2.RWEN=0。

16.3.5 写入计数寄存器

1. 在 RTC_CR1.START=1 设定后，经过 2 个以上的 RTC 计数时钟后，设定 RTC_CR2.RWREQ=1，停止日历寄存器计数，进入读写模式；
2. 查询直到 RTC_CR2.RWEN=1；
3. 写入秒，分，时，周，日，月，年计数寄存器值；
4. 设定 RTC_CR2.RWREQ=0，计数器重新开始计数。注意，须在 1 秒内完成所有写操作；
5. 查询直到 RTC_CR2.RWEN=0。

16.3.6 闹钟设定

1. 设定 RTC_CR2.ALME=0，闹钟禁止；
2. 设定 RTC_CR2.ALMIE=1，闹钟中断许可；
3. 分闹钟 RTC_ALMMIN，时闹钟 RTC_ALMHOUR，周闹钟 RTC_ALMWEEK 设定；
4. 设定 RTC_CR2.ALME=1，闹钟许可；
5. 等待发生中断；
6. 当 RTC_CR2.ALMF=1 时，进入闹钟中断处理。

16.3.7 时钟误差补偿

由于外部副发振晶振在各种温度条件下工作存在误差，在需要得到高精度的计数结果时，需要对误差进行补偿。补偿方法参照【16.5.15 时钟误差补偿寄存器（RTC_ERRCRH、RTC_ERRCRL）】。

16.3.8 1Hz 输出

RTC 可输出 1Hz 时钟，提供两种精度输出方式，第一种，无时钟误差补偿的普通精度 1Hz 输出；第二种，每 32 秒内平均补偿的分布式补偿 1Hz 输出。当时钟误差补偿功能有效 RTC_ERRCRH.COMPEN=1 时选择分布式补偿 1Hz 输出。其中，

普通精度的 1Hz 输出设定如下：

1. 设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位日历计数寄存器；
2. 设定 RTC_CR1.START=0，计数停止；
3. 1Hz 输出引脚设定；
4. RTC_CR1.ONEHZOE=1，时钟输出许可；
5. 设定 RTC_CR1.START=1，计数开始；
6. 等待 2 个计数周期以上；
7. 1Hz 输出开始。

分布式补偿 1Hz 输出设定如下：

1. 设定 RTC_CR0.RESET=0，确认 RESET 位为“0”后，设定 RTC_CR0.RESET=1，复位日历计数寄存器；
2. 设定 RTC_CR1.START=0，计数停止；
3. 1Hz 输出引脚设定；
4. RTC_CR1.ONEHZOE=1，时钟输出许可；
5. 时钟误差补偿寄存器 RTC_ERRCRL.COMP[7:0] 与 RTC_ERRCRH.COMP[8] 补偿数设定；
6. 时钟误差补偿寄存器 RTC_ERRCRH.COMPEN=1，误差补偿有效；
7. 设定 RTC_CR1.START=1，计数开始；
8. 等待 2 个计数周期以上；
9. 1Hz 输出开始。

16.4 中断说明

RTC 支持 2 种中断类型。计时闹钟中断与定周期中断。

16.4.1 闹钟中断

闹钟中断 RTC_ALM，在控制寄存器 2 (RTC_CR2) 的 ALMIE=1 并且控制寄存器 2 (RTC_CR2) 的 ALME=1 时，若当前日历时间与分闹钟寄存器 (RTC_ALMMIN)、时闹钟寄存器 (RTC_ALMHOUR)、周闹钟寄存器 (RTC_ALMWEEK) 相等时，触发闹钟中断。

16.4.2 定周期中断

定周期中断 RTC_PRD，控制寄存器 2（RTC_CR2）的 PRDIE=1 时，选择的周期发生后，触发定周期唤醒中断。

16.5 寄存器描述

表 16-2 RTC 基地址

名称	基地址	描述
RTC	0x4000 D400	RTC 基地址

表 16-3 RTC 寄存器列表

寄存器	描述	偏移量	位宽	复位值
RTC_CR0	控制寄存器 0	0x0000	8	不定
RTC_CR1	控制寄存器 1	0x0004	8	0x00
RTC_CR2	控制寄存器 2	0x0008	8	0x00
RTC_CR3	控制寄存器 3	0x000C	8	0x00
RTC_SEC	秒计数寄存器	0x0010	8	0x00
RTC_MIN	分计数寄存器	0x0014	8	0x00
RTC_HOUR	时计数寄存器	0x0018	8	0x00
RTC_WEEK	周计数寄存器	0x001C	8	0x00
RTC_DAY	日计数寄存器	0x0020	8	0x00
RTC_MON	月计数寄存器	0x0024	8	0x00
RTC_YEAR	年计数寄存器	0x0028	8	0x00
RTC_ALMMIN	分闹钟寄存器	0x002C	8	0x00
RTC_ALMHOUR	时闹钟寄存器	0x0030	8	0x00
RTC_ALM WEEK	周闹钟寄存器	0x0034	8	0x00
RTC_ERRCRH	时钟误差补偿寄存器	0x0038	8	0x00
RTC_ERRCRL	时钟误差补偿寄存器	0x003C	8	0x00

16.5.1 控制寄存器 0 (RTC_CR0)

复位值：不定

b7	b6	b5	b4	b3	b2	b1	b0
Res							RESET

位/位域	标记	位名	功能	读写
b7~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	RESET	RTC 日历计数器复位	写入状态 0: 初始化寄存器无效 1: 初始化寄存器有效 初始化所有的 RTC 寄存器。 读出状态 0: 正常计数状态或 RTC 软件复位结束 1: RTC 处于复位状态 注意: 请在写入“1”前确认该位为“0”, 否则将无法初始化。	RW

16.5.2 控制寄存器 1 (RTC_CR1)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
START	Res	ONEHZOE	Res	AMPM	PRDS[2:0]		

位/位域	标记	位名	功能	读写			
b7	START	RTC 计数开始	0: RTC 计数停止 1: RTC 计数工作	RW			
b6	Res	保留位	读出时为“0”,写入时写“0”	RW			
b5	ONEHZOE	1Hz 输出使能	0: 1Hz 输出禁止 1: 1Hz 输出许可	RW			
b4	Res	保留位	读出时为“0”,写入时写“0”	RW			
b3	AMPM	时制选择	0: 12 小时时制 1: 24 小时时制	RW			
b2~0	PRDS[2:0]	周期中断选择	周期选择设定:		RW		
			PRDS[2]	PRDS[1]		PRDS[0]	周期选择
			0	0		0	不选择
			0	0		1	每 0.5 秒周期
			0	1		0	每 1 秒周期
			0	1		1	每 1 分周期
			1	0		0	每 1 时周期
			1	0		1	每 1 日周期 (每日 00 时 00 分 00 秒)
			1	1		X	每 1 月周期 (每月 1 日 00 时 00 分 00 秒)
注意: 在 START=1 计数过程中写入周期选择时, 为防止误动作请将周期中断许可关闭。并且在写入后应将相关标志位清除。							

16.5.3 控制寄存器 2 (RTC_CR2)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ALME	ALMIE	PRDIE	Res	ALMF	PRDF	RWEN	RWREQ

位/位域	标记	位名	功能	读写
b7	ALME	闹钟使能	0: 闹钟功能禁止 1: 闹钟功能许可 注意: 在 START=1 日历计数过程中并且 ALMIE=1 中断许可的情况下使能 ALME 时, 为防止误动作请将系统中断关闭并将 ALMF 标志位清除。	RW
b6	ALMIE	闹钟中断使能	0: 闹钟中断禁止 1: 闹钟中断许可	RW
b5	PRDIE	周期中断使能	0: 周期中断禁止 1: 周期中断许可	RW
b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	ALMF	闹钟标志	0: 闹钟不匹配 1: 闹钟匹配 注意: 在 ALME=1 时有效, 闹钟匹配时, 32.7689KHz 一个时钟后置“1”。写“0”时清除标志, 写“1”无效。	RW
b2	PRDF	周期标志	0: 周期不发生 1: 周期发生 注意: 设定周期发生后, 该位置“1”。写“0”时清除标志, 写“1”无效。	RW
b1	RWEN	读出/写入允许	0: 读出/写入禁止 1: 读出/写入允许 注意: 日历寄存器读写允许标志。在读出/写入前请确认该位是否为“1”。日历寄存器包括秒, 分, 时, 周, 日, 月, 年计数寄存器。	RW
b0	RWREQ	读出/写入请求	0: 正常计数模式 1: 读出/写入请求 注意: 在读出/写入日历寄存器时请将该位置“1”, 请求读写, 由于计数器在连续计数, 请在 1 秒的时间内完成读出/写入操作并将该位清“0”。	RW

16.5.4 控制寄存器 3 (RTC_CR3)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RCKSEL	Res		LRCEN	Res			

位/位域	标记	位名	功能	读写
b7	RCKSEL	RTC 计数时钟选择	0: 选择副发振器 1: 选择 LRC 注意: RTC 计数时钟选择 LRC 时, 计数偏差参考数据手册 LRC 振荡器特性。但是复位控制章节所示所有复位期间, LRC 频率偏低, 最大偏差-55%。	RW
b6~b5	Res	保留位	读出时为“0”, 写入时写“0”	RW
b4	LRCEN	低速发振器使能	0: 低速发振器停止 1: 低速发振器工作 注意: 低速发振的工作与停止由时钟控制电路和 LRCEN 任何一个设定值决定。低速发振作为 RTC 时钟源时请设定 LRCEN 位使能。	RW
b3~b0	Res	保留位	读出时为“0”, 写入时写“0”	RW

16.5.5 秒计数寄存器 (RTC_SEC)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	SECD[2:0]			SECU[3:0]			

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”, 写入时写“0”	RW
b6~b4	SECD[2:0]	秒十位	秒十位计数值	RW
b3~b0	SECU[3:0]	秒个位	秒个位计数值	RW

表示 0-59 秒, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

16.5.6 分计数寄存器 (RTC_MIN)

复位值：不定

b7	b6	b5	b4	b3	b2	b1	b0
Res	MIND[2:0]			MINU[3:0]			

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6~b4	MIND[2:0]	分十位	分十位计数值	RW
b3~b0	MINU[3:0]	分个位	分个位计数值	RW

表示 0-59 分，采用十进制计数。请写入十进制 0-59 的 BCD 码，写入错误值时，写入值将被忽略。

16.5.7 时计数寄存器 (RTC_HOUR)

复位值：0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	HOURL[1:0]			HOURU[3:0]			

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5~b4	HOURL[1:0]	时十位	时十位计数值	RW
b3~b0	HOURU[3:0]	时个位	时个位计数值	RW

24 小时制时，表示 0-23 小时。12 小时制时，b5=0 表示 AM，则 01~12 表示上午；b5=1 表示 PM，则 21~32 表示下午。

请根据控制位 AMPM 的值，设定正确十进制的 0~23 或者 01~12,21~32 的 BCD 码。写入超出范围的值将被忽略。

具体时间表示参考下表：

24 小时制	AMPM=1	12 小时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
00 时	00H	AM 12 时	12H
01 时	01H	AM 01 时	01H
02 时	02H	AM 02 时	02H
03 时	03H	AM 03 时	03H
04 时	04H	AM 04 时	04H
05 时	05H	AM 05 时	05H
06 时	06H	AM 06 时	06H

24 小时时制	AMPM=1	12 小时时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
07 时	07H	AM 07 时	07H
08 时	08H	AM 08 时	08H
09 时	09H	AM 09 时	09H
10 时	10H	AM 10 时	10H
11 时	11H	AM 11 时	11H
12 时	12H	PM 12 时	32H
13 时	13H	PM 01 时	21H
14 时	14H	PM 02 时	22H
15 时	15H	PM 03 时	23H
16 时	16H	PM 04 时	24H
17 时	17H	PM 05 时	25H
18 时	18H	PM 06 时	26H
19 时	19H	PM 07 时	27H
20 时	20H	PM 08 时	28H
21 时	21H	PM 09 时	29H
22 时	22H	PM 10 时	30H
23 时	23H	PM 11 时	31H

16.5.8 日计数寄存器 (RTC_DAY)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		DAYD[1:0]		DAYU[3:0]			

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5~b4	DAYD[1:0]	日十位	日十位计数值	RW
b3~b0	DAYU[3:0]	日个位	日个位计数值	RW

十进制表示 1~31 日, 自动计算闰年和月份。具体表示如下:

月份	日计数表示
2 月 (普通年)	01~28
2 月 (闰年)	01~29
4、6、9、11 月	01~30
1、3、5、7、8、10、12 月	01~31

16.5.9 周计数寄存器 (RTC_WEEK)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res					WEEK[2:0]		

位/位域	标记	位名	功能	读写
b7~b3	Res	保留位	读出时为“0”,写入时写“0”	RW
b2~b0	WEEK[2:0]	周	周计数值	RW

十进制 0~6 表示周日~周六。请写入正确的十进制 0~6 的 BCD 码, 写入其他值, 将被忽略。周计数值对应关系如下:

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

16.5.10 月计数寄存器 (RTC_MON)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res				MON[4:0]			

位/位域	标记	位名	功能	读写
b7~b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4~b0	MON[4:0]	月	月计数值	RW

十进制 1~12 表示 1~12 月。请写入正确的十进制 1~12 的 BCD 码, 写入其他值, 将被忽略。

16.5.11 年计数寄存器 (RTC_YEAR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
YEAR[3:0]				YEARU[3:0]			

位/位域	标记	位名	功能	读写
b7~b4	YEAR[3:0]	年十位	年十位计数值	RW
b3~b0	YEARU[3:0]	年个位	年个位计数值	RW

十进制 0~99 表示 0~99 年。根据月进位计数。自动计算闰年如: 00、04、08、...、92、96 等。请写入正确的十进制年计数值, 写入错误值将被忽略。

16.5.12 分闹钟寄存器 (RTC_ALMMIN)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMMIND[2:0]			ALMMINU[3:0]			

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6~b4	ALMMIND[2:0]	分闹钟十位	分闹钟十位匹配值	RW
b3~b0	ALMMINU[3:0]	分闹钟个位	分闹钟个位匹配值	RW

请设定十进制 0~59 的 BCD 码。写入其他值, 不会发生闹钟匹配。

16.5.13 时闹钟寄存器 (RTC_ALMHOUR)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		ALMHOURD[1:0]		ALMHOURU[3:0]			

位/位域	标记	位名	功能	读写
b7~b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5~b4	ALMHOURD[1:0]	时闹钟十位	时闹钟十位匹配值	RW
b3~b0	ALMHOURU[3:0]	时闹钟个位	时闹钟个位匹配值	RW

请根据时制设定正确的闹钟匹配值,否则不会发生时闹钟匹配。

16.5.14 时闹钟寄存器 (RTC_ALMWEED)

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMWEED[6:0]						

位/位域	标记	位名	功能	读写
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6~b0	ALMWEED[6:0]	周闹钟	周闹钟匹配值。 b0~b6 分别对应周日~周六,对应为置“1”时,代表每周该日闹钟有效。如, b0=1, b5=1 代表周日和周五闹钟设定有效。	RW

请根据时制设定正确的闹钟匹配值,否则不会发生时闹钟匹配。

16.5.15 时钟误差补偿寄存器 (RTC_ERRCRH、RTC_ERRCRL)

RTC_ERRCRH

复位值: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
COMPEN	Res						COMP[8]

位/位域	标记	位名	功能	读写
b7	COMPEN	补偿使能	0: 时钟误差补偿无效 1: 时钟误差补偿有效	RW
b6~b1	Res	保留位	读出时为“0”,写入时写“0”	RW
b0	COMP[8]	补偿值	与 COMP[7:0]共同设定补偿值	RW

RTC_ERRCRL

复位值: 0x20

b7	b6	b5	b4	b3	b2	b1	b0
COMP[7:0]							

位/位域	标记	位名	功能	读写

通过补偿值设定, 可针对每秒进行 $\pm 0.96\text{ppm}$ 的精度补偿。补偿值为 9 位带小数点的 2 的补码, 后 5 位为小数部分。可补偿范围 $-275.5\text{ppm} \sim +212.9\text{ppm}$ 。最小分辨率 0.96ppm 。具体补偿精度请参考下表:

			补偿值设定								补偿数			
			COMPEN	COMP[8:0]										
b7~b0	COMP[7:0]	补偿值	1	1	0	0	0	0	0	0	0	0	-275.5ppm	
				1	0	0	0	0	0	0	0	1	-274.6ppm	
				-	-	-	-	-	-	-	-	-	-	
				0	0	0	0	0	0	0	0	0	-30.5ppm	
				-	-	-	-	-	-	-	-	-	-	
				0	0	0	0	1	1	1	1	1	-0.96ppm	
				0	0	0	1	0	0	0	0	0	0ppm	
				0	0	0	1	0	0	0	0	1	+0.96ppm	
				-	-	-	-	-	-	-	-	-	-	
				0	0	1	0	0	0	0	0	0	+30.5ppm	
				-	-	-	-	-	-	-	-	-	-	
				-	0	1	1	1	1	1	1	1	0	+212.0ppm
				-	0	1	1	1	1	1	1	1	1	+212.9ppm
				0	X	X	X	X	X	X	X	X	X	无效

RW

补偿计算说明：

当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 0.9999888Hz，则：

$$\text{实际发振频率} = 32768 * 0.9999888 \approx 32767.63$$

$$\begin{aligned}\text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} * 10^6 \\ &= (32767.96 - 32768) / 32768 * 10^6 \\ &= -11.29\text{ppm}\end{aligned}$$

设定值计算：

$$COMP[8:0] = \left(\frac{\text{补偿目标值}[ppm] \times 2^{15}}{10^6} \right)_{\text{取 2 的补码}} + 0001.00000B$$

如果补偿目标值为+20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}COMP[8:0] &= (20.3 * 2^{15}/10^6)_{\text{取 2 的补码}} + 0001.00000B \\ &= (0.6651904)_{\text{取 2 的补码}} + 0001.00000B \\ &= 0000.10101B + 0001.00000B \\ &= 0001.10101B\end{aligned}$$

如果补偿目标值为-20.3ppm，计算相应的寄存器值如下：

$$\begin{aligned}COMP[8:0] &= (-20.3 * 2^{15}/10^6)_{\text{取 2 的补码}} + 0001.00000B \\ &= (-0.6651904)_{\text{取 2 的补码}} + 0001.00000B \\ &= 1111.01011B + 0001.00000B \\ &= 0000.01011B\end{aligned}$$

16.6 注意事项

1. 当 HCLK 时钟与 RTC 计数时钟使用相同时钟源时，HCLK 频率大于等于 RTC 计数时钟频率
2. 当 HCLK 时钟与 RTC 计数时钟使用不同时钟源时，HCLK 频率大于等于 RTC 计数时钟频率的 2 倍

17 看门狗计数器（SWDT）

17.1 概述

看门狗计数器是一种计数时钟源为内部低速 RC（32.768KHz）的专用看门狗计数器（SWDT）。专用看门狗是 16 位递减计数器，用来监测由于外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行而产生的软件故障。看门狗计数器支持窗口功能，在计数开始前可预设窗口区间，计数值位于窗口区间时，可刷新计数器，计数重新开始。基本特性如表 17-1。

表 17-1 看门狗计数器的基本特性

计数时钟	SWDTCLK 的 1/16/32/64/128/256/2048 分频
最长溢出时间	SWDT:1.14hour(max)
计数模式	递减计数
窗口功能	可设定窗口区间，定义刷新动作的允许区间
启动方式	1) 硬件启动 2) 软件启动
停止条件	1) 复位中 2) 下溢，刷新错误发生时 再开始：硬件启动模式下，复位或中断请求输出后自动开始 软件启动模式下，再次设定刷新寄存器
中断/复位条件	1) 计数下溢 2) 刷新错误

17.2 功能说明

17.2.1 启动看门狗

看门狗计数器的启动方式有两种：硬件启动方式和软件启动方式。

硬件启动方式是指启动时从主闪存区域读取看门狗计数器的设定信息（ICG0 寄存器），计数器自动开始计数；软件启动方式是指设定控制寄存器后，写刷新寄存器完成刷新动作，计数器开始计数。

17.2.2 硬件启动方式

ICG0 寄存器的位 0(SWDTAUTS)为 0 时，为硬件启动方式。选择硬件启动方式时，SWDT_CR 寄存器的相关设定信息无效。

硬件启动方式时，在复位期间将 ICG0 寄存器里的 SWDT 相关设定（计数时钟、窗口设定值、计数周期等）载入到 SWDT 的模块中，复位之后、计数器按照设定自动开始计数。图 17-1 为硬件启动方式的动作例。

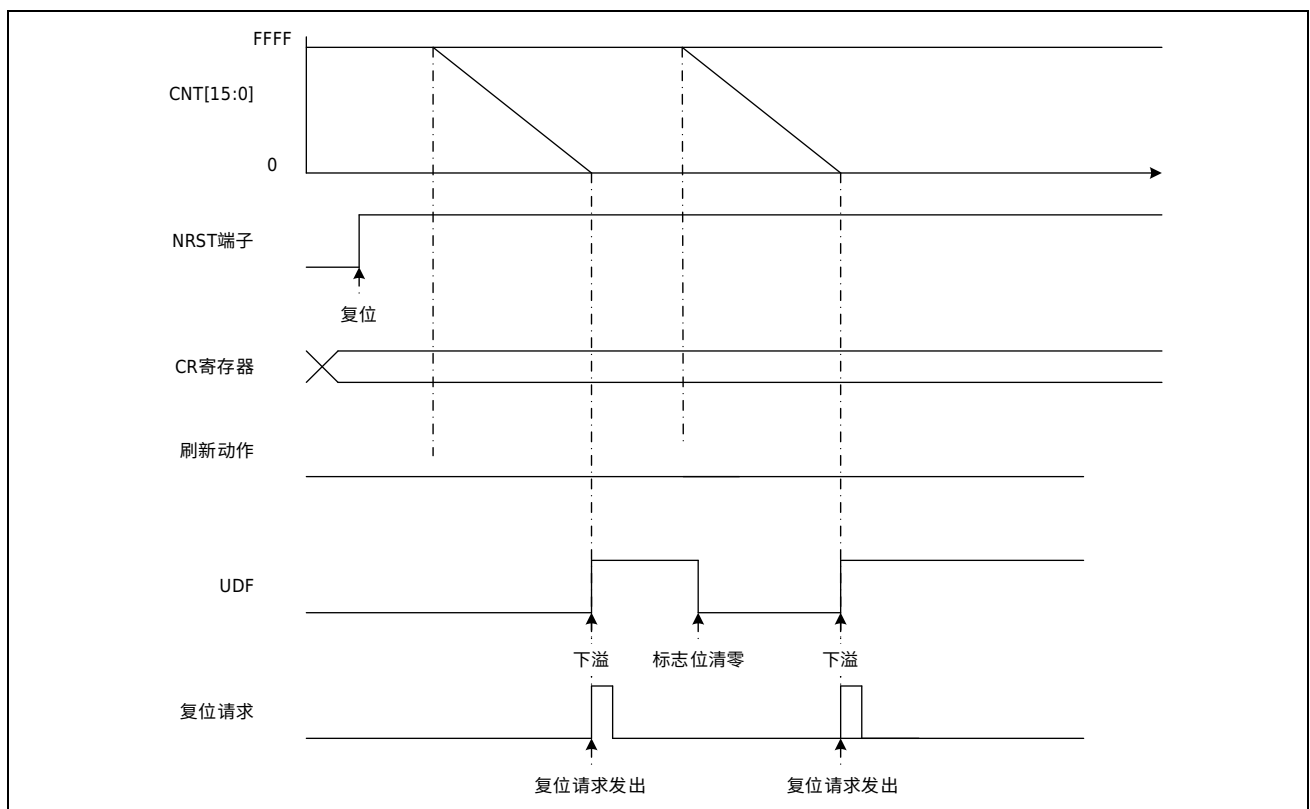


图 17-1 硬件启动例

17.2.3 软件启动方式

ICG0 寄存器的位 0(SWDTAUTS)为 1 时，通过设定刷新寄存器的方式启动 SWDT 为软件启动方式。复位后，设定 SWDT_CR 寄存器中的计数时钟、窗口设定值、计数周期等，然后执行刷新动作，计数器就开始计数。SWDT_CR 设定只能 1 次，再次设定写入值无效。图 17-2 为软件启动方式的动作例。

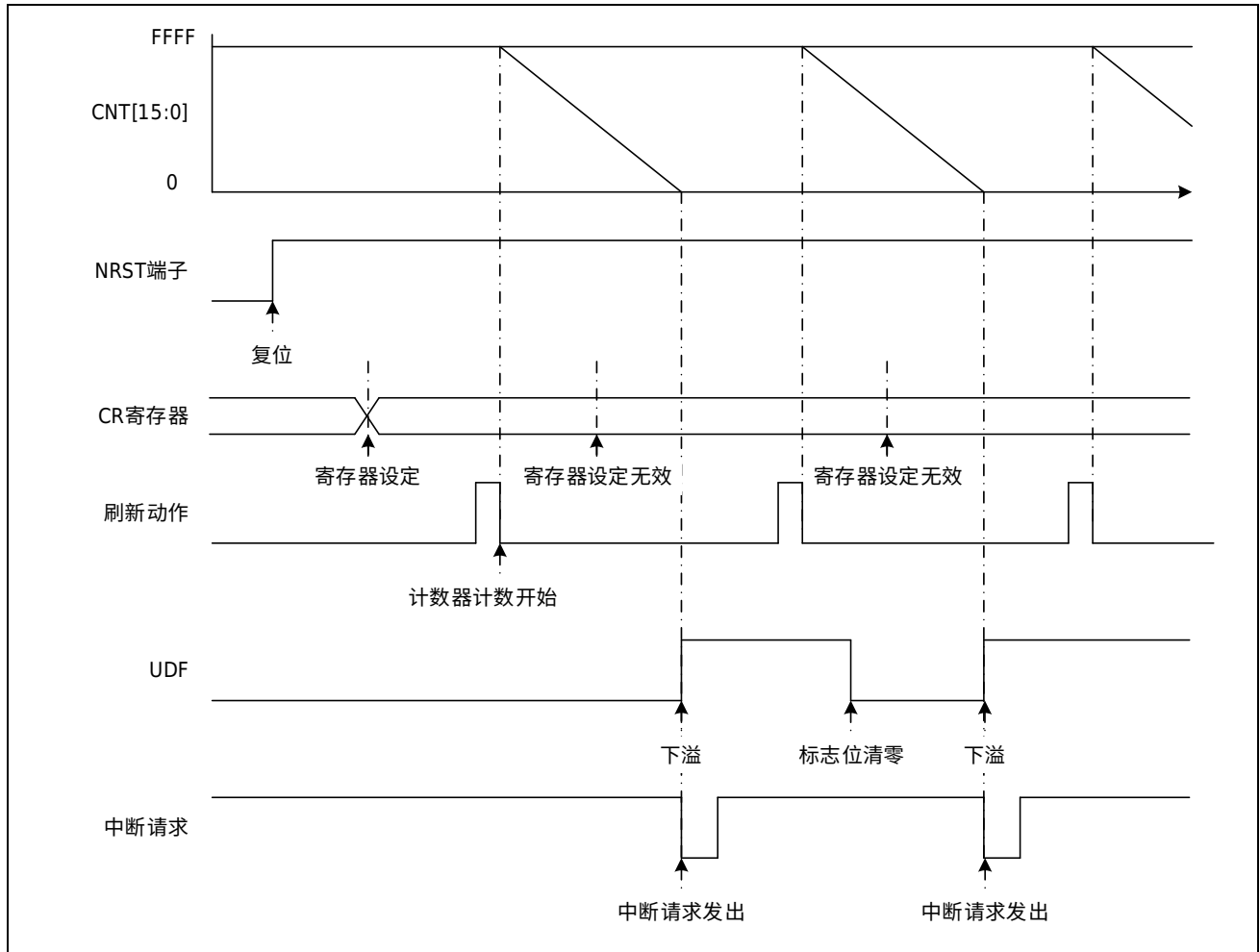


图 17-2 软件启动例

17.2.4 刷新动作

SWDT_RR 寄存器中先写 0x0123、再写 0x3210 完成一次刷新动作，SWDT 的计数器就开始计数(软件启动)或重新开始计数。

SWDT_RR 寄存器在写 0x0123、0x3210 之间，若对发生对其他寄存器访问或读取 SWDT_RR 寄存器等，不影响正常的刷新动作。

如图 17-3 所示动作示例。

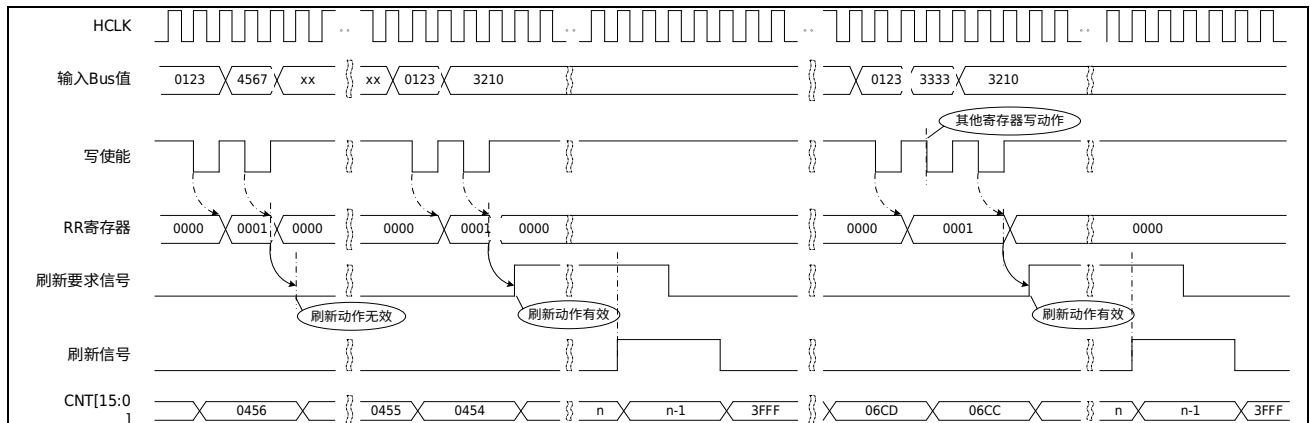


图 17-3 各种刷新动作时序示例（动作确认，刷新要求信号的下降沿等）

刷新动作需要 4 个计数周期完成计数值的更新，所以请在刷新下位窗口和下溢位置的提前 4 个计数值完成刷新寄存器的写入。计数值的确认请读取状态寄存器。

17.2.5 标志位

刷新错误标志位和计数下溢标志位在复位请求和中断请求的情况下都会保持。当复位解除或进入中断后，可以通过查询标志位来确认复位或中断原因。

标志位清零：先读“1”再写“0”。

刷新错误或者计数下溢标志位为置位时，硬件启动模式看门狗计数不停止。对于 SWDT 来说，当对标志位写“0”时，最多需要经过 3 个 SWDTCLK 和 2 个 HCLK 时间后，寄存器位才能被清零。另，在发生刷新错误或者下溢错误的一定时间内，对标志位读“1”清零无效，这段时间为：1 个计数周期+2 个 SWDTCLK。

17.2.6 中断复位

SWDT 在计数器计数下溢或刷新错误时，可以选择产生中断请求或复位请求。硬件启动时通过设定 ICG0 的 SWDTITS 位，软件启动时通过设定 SWDT_CR 寄存器 ITS 位，来决定产生中断请求还是复位请求。

SWDT 的中断复位产生条件有两种。一种是计数器计数产生下溢；一种是在刷新允许区间之外执行刷新动作，产生刷新错误。

17.2.7 计数下溢

如图 17-4 例，在递减计数到零时产生下溢。

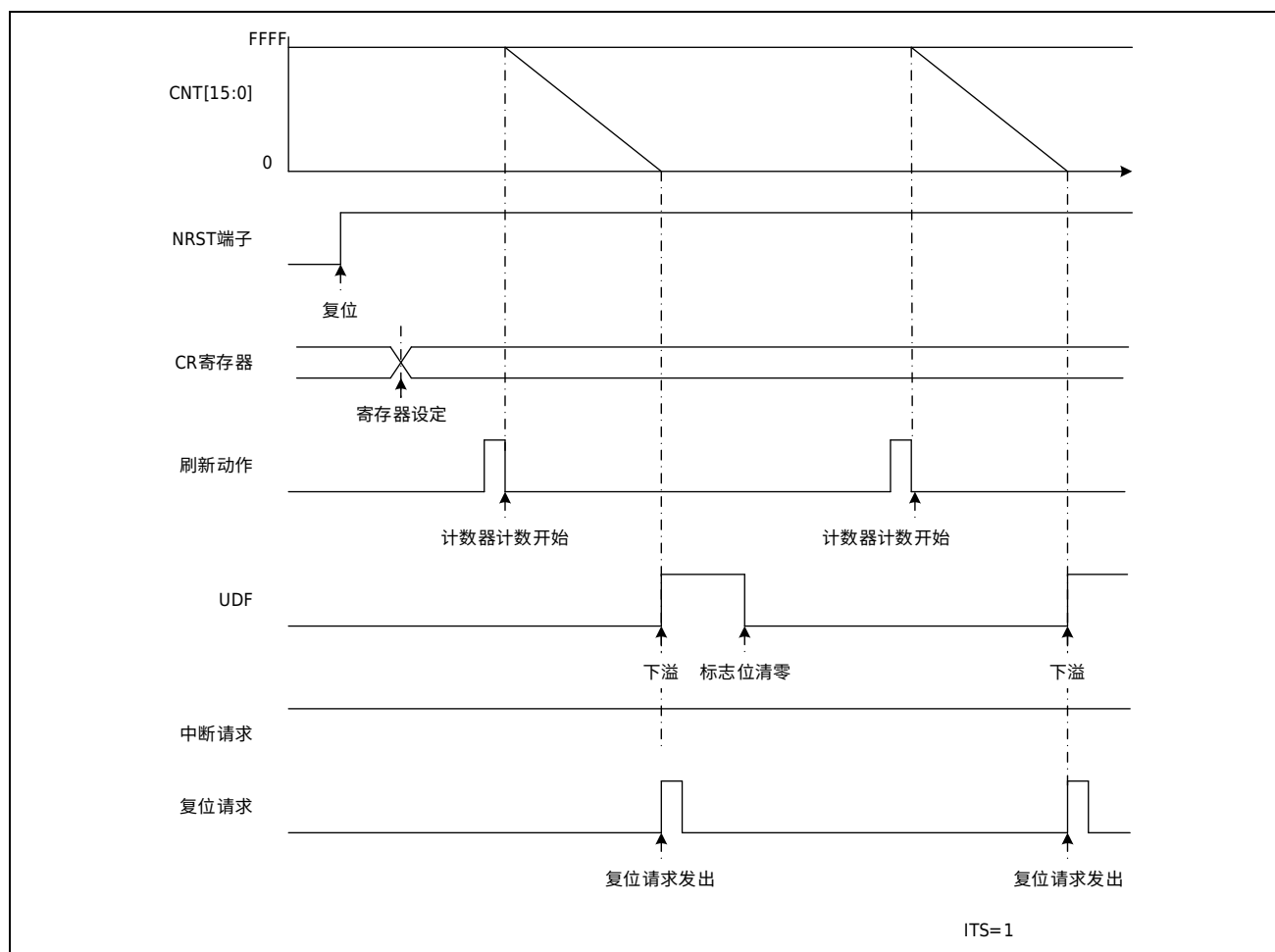


图 17-4 计数器下溢动作例

17.2.8 刷新错误

在设定了窗口区间后，只有在窗口区间内执行刷新动作时计数器才会被刷新、重新开始计数，在窗口区间外执行刷新动作时产生刷新错误。图 17-5 为刷新动作例。

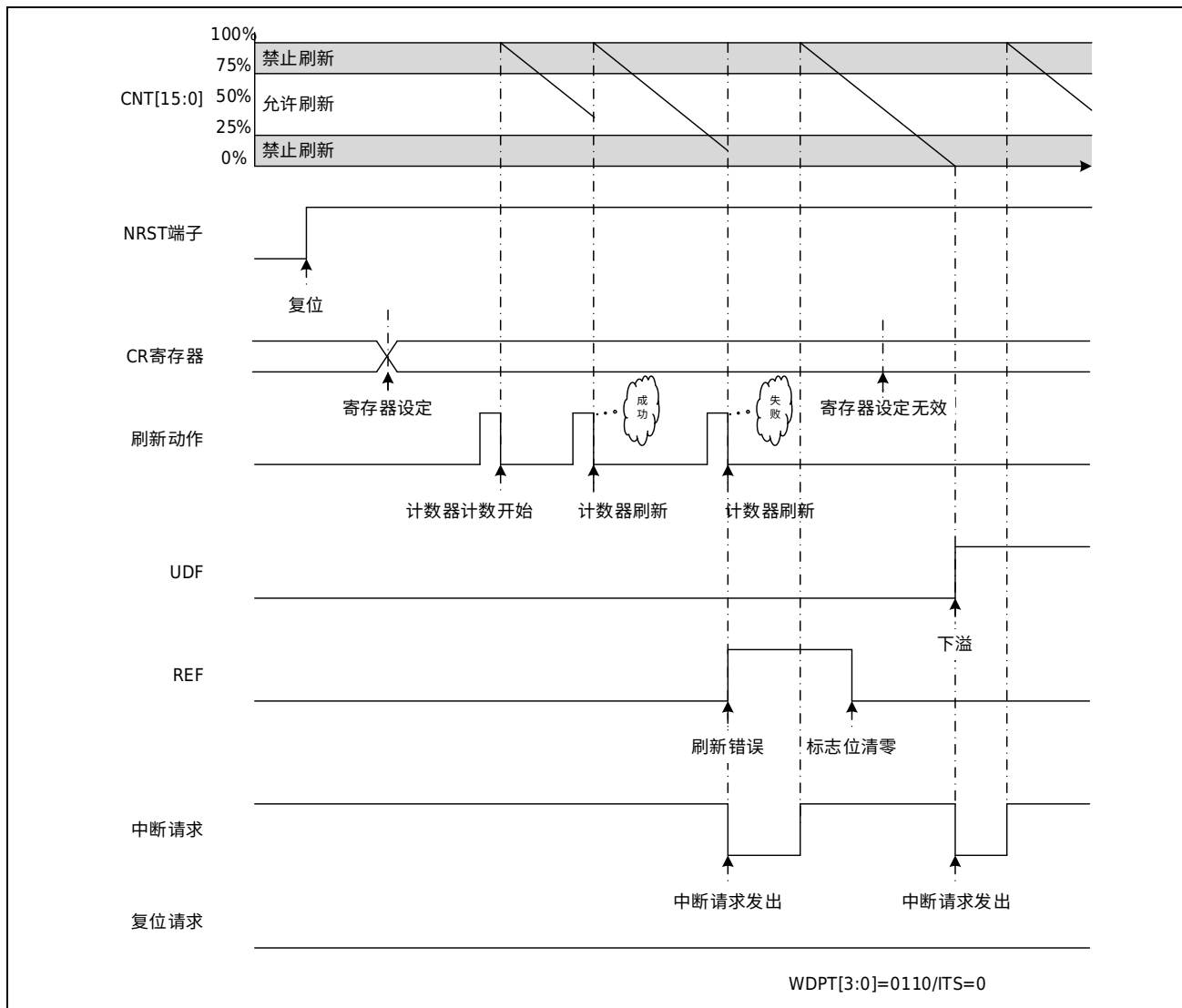


图 17-5 计数器刷新动作例

17.3 寄存器描述

表 17-2 SWDT 基地址

名称	基地址	描述
SWDT	0x4000 CC00	SWDT 基地址

表 17-3 SWDT 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
SWDT_CR	SWDT 控制寄存器	0x00	32	0x8001 0FF3
SWDT_SR	SWDT 状态寄存器	0x04	32	0x0000 0000
SWDT_RR	SWDT 刷新寄存器	0x08	32	0x0000 0000

17.3.1 控制寄存器 (SWDT_CR)

复位值: 0x8001 0FF3

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ITS	Res														SLPOFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDPT[3:0]				CKS[3:0]				Res		PERI[1:0]	

位/位域	标记	位名	功能	读写
b31	ITS	刷新错误/溢出 中断/复位选择	0: 中断请求 1: 复位请求	RW
b30~b17	Res	保留位	读出时为“0”,写入时写“0”	R
b16	SLPOFF	SWDT 在 Sleep/stop 模式 下计数禁止	0: SWDT 在 sleep/stop 模式下计数许可 1: SWDT 在 sleep/stop 模式下计数禁止	RW
b15~b12	Res	保留位	读出时为“0”,写入时写“0”	R
b11~b8	WDPT[3:0]	刷新允许区域计 数值百分比	0000: 0%~100%	RW
			0001: 0%~25%	
			0010: 25%~50%	
			0011: 0%~50%	
			0100: 50%~75%	
			0101: 0%~25%,50%~75%	
			0110: 25%~75%	
			0111: 0%~75%	
			1000: 75%~100%	
			1001: 0%~25%,75%~100%	
			1010: 25%~50%,75%~100%	
			1011: 0%~50%,75%~100%	
b7~b4	CKS[3:0]	计数时钟	1100: 50%~100%	RW
			1101: 0%~25%,50%~100%	
			1110: 25%~100%	
			1111: 0%~100%	
			0000: SWDTCLK	
			0100: SWDTCLK/16	
			0101: SWDTCLK/32	
			0110: SWDTCLK/64	
b3~b2	Res	保留位	0111: SWDTCLK/128	R
			1000: SWDTCLK/256	
			1011: SWDTCLK/2048	
			其余值: 预留功能	
b1~b0	PERI[1:0]	计数周期	00: 256 cycle 01: 4096 cycle 10: 16384 cycle 11: 65536 cycle	RW

17.3.2 状态寄存器 (SWDT_SR)

复位值: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														REF	UDF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

位/位域	标记	位名	功能	读写
b31~b18	Res	保留位	读出时为“0”,写入时写“0”	R
b17	REF	刷新错误标志	0: 没有刷新错误 1: 发生刷新错误 该位读出 1 后, 写入 0, 清零。	RW
b16	UDF	计数下溢标志	0: 没有计数下溢 1: 发生计数下溢 该位读出 1 后, 写入 0, 清零。	RW
b15~b0	CNT[15:0]	计数值	计数器当前计数值	R

17.3.3 刷新寄存器 (SWDT_RR)

复位值: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RF[15:0]															

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	R
b15~b0	RF[15:0]	刷新值	依次写入 0x0123、0x3210 之后, 完成刷新动作 当寄存器写入 0x0123h 后, 读出寄存器为 0x0000 0001;其余情况 读出值都是 0x0000 0000。	RW

17.4 使用注意事项

SWDT 动作时, CPU 时钟 HCLK 的动作频率必须大于或等于计数时钟 (ICG0.SWDTCK[3:0]或 SWDT_CR.CKS[3:0]设定 SWDTCLK 分频后的时钟) 的 4 倍, 即 HCLK 频率 $\geq$ 计数时钟频率 $\times 4$ 。

18 通用同步异步收发器 (USART)

18.1 概述

本产品搭载通用串行收发器模块 (USART) 6 个单元。通用串行收发器模块 (USART) 能够灵活地与外部设备进行数据交换；本 USART 支持通用异步串行通信接口 (UART), 时钟同步通信接口 (CSI)。支持调制解调器操作 (CTS/RTS 操作), 异步 UART 模式支持多处理器操作。其中 UART 单元 1 和 TimerB(Unit4)、INTC(INTP1) 模块配合支持 LIN-bus 功能 (UART_LIN)。

USART 主要特性:

- 支持全双工/半双工异步通信, 全双工时钟同步通信
- 发送器和接收器具有独立使能位
- 通道 1 和 TimerB 配合支持 LIN_bus 功能
- 内置双缓冲器
- LSB/MSB 可选
- 支持调制解调器操作 (CTS/RTS)
- 传输标志: 发送数据寄存器空, 发送数据完成, 接收数据寄存器满, 接收错误标志

UART 主要特性:

- 数据长度可编程: 8 位/9 位
- 校验功能可配置: 奇校验/偶校验/无校验
- 停止位可配置: 1 位/2 位
- 时钟源可选: 内部时钟源(内部波特率生成器生成的时钟)/外部时钟源(USARTn_CK 管脚输入的时钟)
- 接收错误: 校验错误, 帧错误, 上溢错误
- 支持多个处理器间通信
- 内置数字滤波器
- 支持全双工/半双工通信方式

时钟同步模式主要特性:

- 数据长度: 8 位
- 接收错误: 上溢错误
- 时钟源: 内部时钟源(内部波特率生成器生成的时钟)/外部时钟源(USARTn_CK 管脚输入的时钟)

UART_LIN 的主要特性 (配合 TimerB(Unit4)、INTC(INTP1)):

- 数据长度: 8 位
- 支持唤醒信号的检测
- 支持同步间隔段 (BF) 的检测

- 支持同步段测量，波特率计算

18.2 USART 系统框图

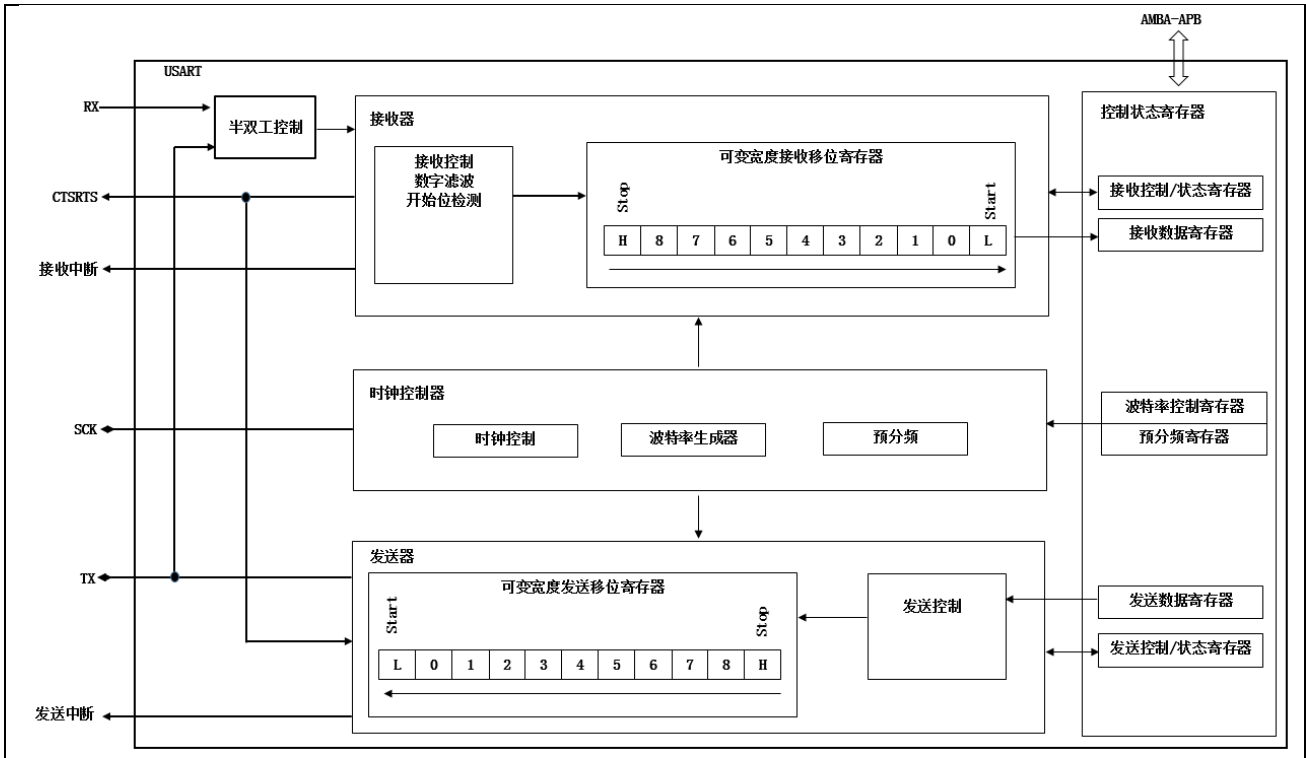


图 18-1 USART 系统框图

18.3 管脚说明

表 18-1 USART 管脚说明

管脚名	方向	功能描述
USARTn_CK	输入输出	时钟
USARTn_TX	输入输出	USART 全双工: 发送数据管脚 USART 半双工: 发送数据管脚/接收数据管脚
USARTn_RX	输入	USART 全双工: 接收数据管脚
USARTn_CTSRTS	输入输出	调制解调器操作管脚 请求发送管脚

n:1~6

18.4 功能说明

本章将对 UART，多处理器通信，LIN，时钟同步模式的功能详细说明。

18.4.1 UART

18.4.1.1 时钟

UART 可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

USARTn_CR2.CLKC[1:0]位设定为 00b 或者 01b 时选择时钟源为内部时钟源即内部波特率生成器生成的时钟。

USARTn_CR2.CLKC[1:0]=00b 时 USARTn_CK 管脚不作为时钟管脚使用，可以作为普通 IO 使用。

USARTn_CR2.CLKC[1:0]=01b 时从 USARTn_CK 管脚输出与通信波特率相同频率的时钟。

内部波特率生成器的时钟源由 USARTn_PR.PSC[1:0]位的设定选择为 HCLK, HCLK/4, HCLK/16, HCLK/64。

外部时钟源

USARTn_CR2.CLKC[1:0]位设定为 10b 或者 11b 时选择时钟源为从 USARTn_CK 管脚输入的外部时钟，输入时钟的频率为波特率的 16 倍(USARTn_CR1.OVER8=0)或者 8 倍(USARTn_CR1.OVER8=1)。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1)}$$

B: 波特率 单位: Mbps

C: USARTn_PR.PSC[1:0]位设定的时钟 (HCLK, HCLK/4, HCLK/16, HCLK/64) 单位: MHz

OVER8: USARTn_CR1.OVER8 设定值

DIV_Integer: USARTn_BRR.DIV_Integer 设定值

最高波特率为 HCLK/8(Mbps)。

外部时钟源时，外部输入 UART 时钟的最高频率要求为 HCLK(MHz)/4，所以时钟源为外部输入时钟时最高波特率为 HCLK/64 (Mbps) (USARTn_CR1.OVER8=0 时) 或者 HCLK/32 (Mbps) (USARTn_CR1.OVER8=1 时)。

需要注意的是，UART 最高通信波特率除了以上描述的基于 HCLK 的计算方法外，还需要参考数据手册电气特性章节规定的最高通信波特率。

18.4.1.2 数据格式

UART 模式时一帧数据是由开始位，数据位，校验位和停止位组成。

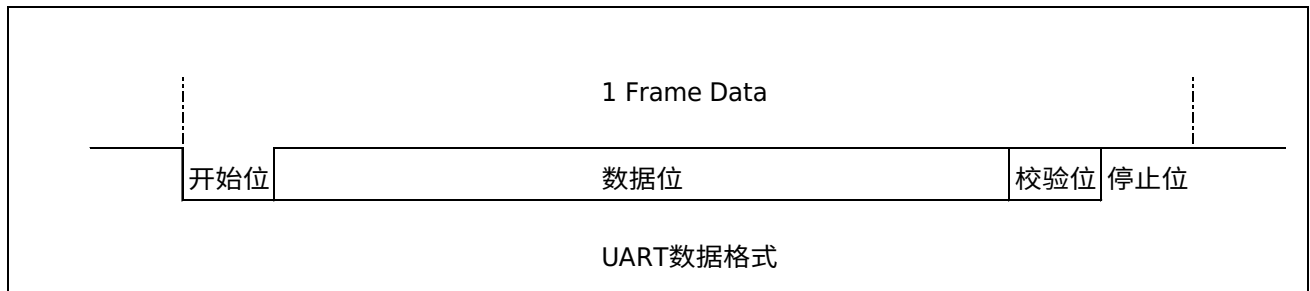


图 18-2 UART 数据格式

开始位

开始位固定有一位的低电平构成。

数据位

数据位可以配置成 8 位或者 9 位。

校验位

校验位可以配置成 1 位偶校验或 1 位奇校验或无校验位。

停止位

停止位固定为高电平，可以配置成 1 位或者 2 位。

18.4.1.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。CTS 功能和 RTS 功能只能二选一，不能同时使用。USARTn_CR3.CTSE=0 时 RTS 功能有效，USARTn_CR3.CTSE=1 时 CTS 功能有效。

CTS 功能

CTS 功能是通过 USARTn_CTSRTS 管脚的输入来控制数据的发送，只有当 USARTn_CTSRTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn_CTSRTS 输入高电平，正在发送的数据不受影响。

RTS 功能

RTS 功能是指通过 USARTn_CTSRTS 管脚输出低电平，请求对方发送数据。

USARTn_CTSRTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn_CR1.RE=1)，且不在接收数据
- USARTn_DR.RDR 寄存器中没有未读取的接收数据
- 无任何接收错误，包括帧错误，校验错误和上溢错误

18.4.1.4 发送器

发送器可发送 8 位或 9 位的数据，具体取决于 USARTn_CR1.M 位的设定值。

发送器使能位 (USARTn_CR1.TE) 置 1，写入发送数据后，发送数据在 TX 管脚上串行输出；相应的时钟脉冲可以选择在 USARTn_CK 管脚输出或者不输出。

发送数据的顺序为：开始位->数据位 (MSB/LSB) ->校验位 (有或者无) ->停止位

发送数据寄存器 TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

发送数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1.TE=1)，如果需要使用发送数据寄存器空中断，则设置 USARTn_CR1.TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空，写通信数据到 USARTn_DR.TDR，数据传输到发送移位寄存器，发送开始 (CTS 功能有效时，USARTn_CTSRTS 输入为低电平时数据传输到发送移位寄存器，发送开始)
8. 如果需要连续发送数据时，重复步骤 7
9. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况，可通过 TI 中断写入最后一个发送数据，并将 USARTn_CR1.TXEIE 写 0，USARTn_CR1.TCIE 写 1，最后一帧数据发送结束后，产生发送完成中断

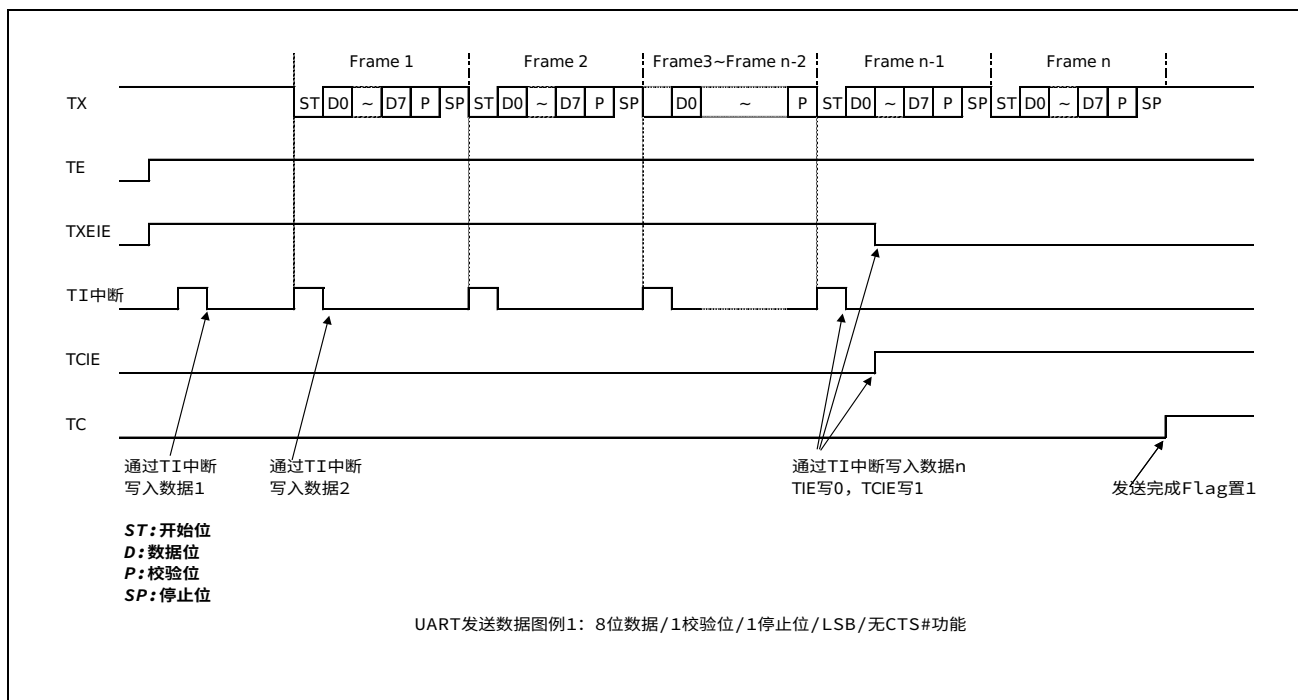


图 18-3 UART 发送数据图例 1

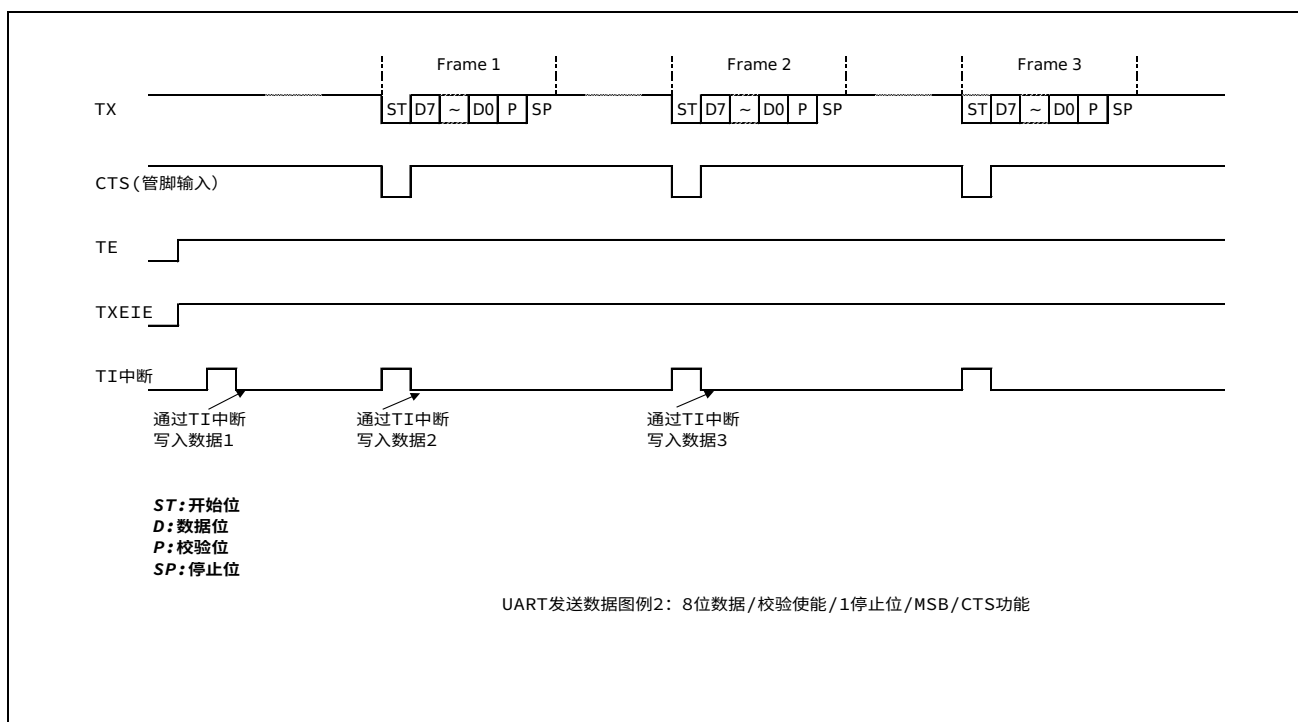


图 18-4 UART 发送数据图例 2

发送器中断

UART 模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1，USARTn_DR.TDR 寄存器的值传送到发送移位寄存器时 TI 中断发生。

TCIE=1，发送数据的最后一位时 USARTn_DR.TDR 寄存器没有更新则 TCI 中断发生。

18.4.1.5 接收器

接收器可接收 8 位或 9 位的数据，具体取决于 USARTn_CR1.M 位的设定值。接收器使能位 (USARTn_CR.RE) 置 1 并检测到开始位后，RX 管脚上数据接收到接收移位寄存器，收满一帧数据，数据从接收移位寄存器传送到接收数据寄存器 USARTn_DR.RDR。

接收数据的顺序为：开始位->数据位(MSB/LSB)->校验位(有或者无)->停止位。

接收数据寄存器 USARTn_DR.RDR 寄存器和内部的接收移位寄存器组成双缓冲器结构，可以连续接收数据。

通过接收数据寄存器满中断读取接收数据时，一次请求只能读取一次数据。

开始位检测

开始位检测可以选择低电平方式或下降沿方式，具体取决于 USARTn_CR1.SBS 位，USARTn_CR1.SBS=0 时为低电平检测，USARTn_CR1.SBS=1 时为下降沿检测。

采样和接收容差

检测到开始条件（低电平或下降沿）后，USART 会基于内部基本时钟对接收数据进行时钟同步，从而开始数据接收。

数据的采样在数据中央，USARTn_CR1.OVER8=0 时在第 8 个内部基本时钟采样，USARTn_MR.OVER8=1 时在第 4 个内部基本时钟采样。

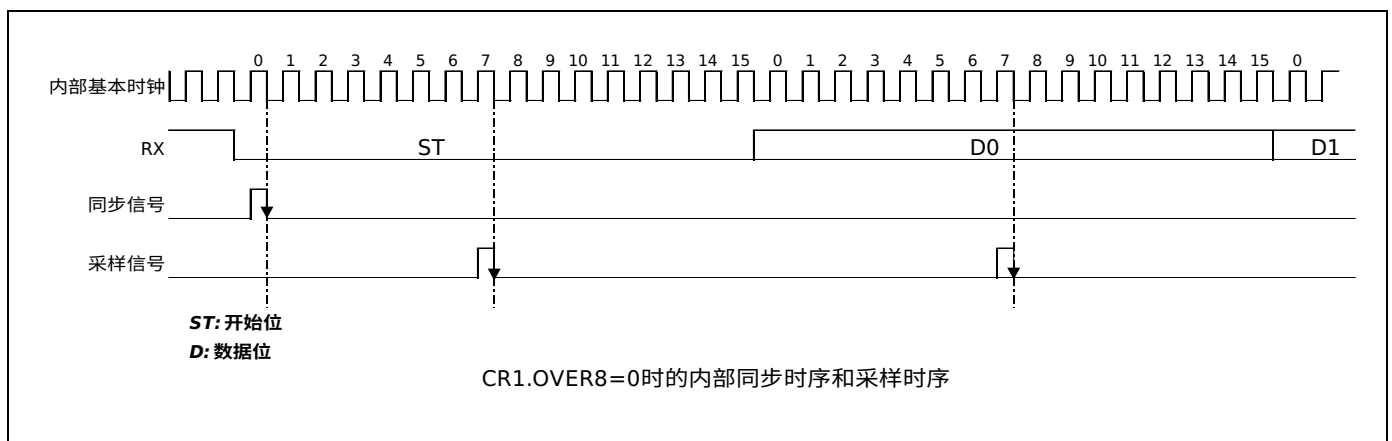


图 18-5 UART 内部同步和采样时序

仅当总时钟系统偏差小于 UART 接收器的容差时，UART 异步接收器才能正常工作。影响总偏差的因素包括：

- 发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- 接收器的波特率量化引起的误差
- 接收器本地振荡器的偏差
- 传输线路引起的偏差

对于正确接收数据，UART 异步接收器所容许的最大偏差值具体取决于以下选项：

- 数据长度 FL。FL 由 USART_CR1 寄存器中 M 位定义的 8 或 9 数据位和 PCE 位定义的校验使能位决定
- 由 USART_CR1 寄存器中 OVER8 位定义的 8 倍或 16 倍过采样

表 18-2 UART 接收器的容差

FL	OVER8 位=0	OVER8 位=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

在特殊情况下，当 RX 管脚接收到数据长度 FL 时间的连续高电平时，表 18-2 中指定的数据可能会略微不同。

接收数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器 (USARTn_CR1. RE=1)，如果需要使用接收中断，则设置 USARTn_CR1. RIE=1
7. 当检测到开始位后，接收器将数据接收到接收移位寄存器，并检查校验位和停止位
 - 1) 校验错误时，接收到的数据传送到 USARTn_DR.RDR 寄存器中并置位 USARTn_SR.PE 标志
 - 2) 停止位不为高电平时，发生帧错误，接收到的数据传送到 USARTn_DR.RDR 寄存器中并置位 USARTn_SR.FE 标志
 - 3) 发生上溢错误时，数据丢失并置位 USARTn_SR.ORE 标志
 - 4) 无错误发生时，接收到的数据传送到 USARTn_DR.RDR 寄存器中，并置位 USARTn_SR.RXNE 标志，读取接收到的数据后重复步骤 7 可连续接收数据

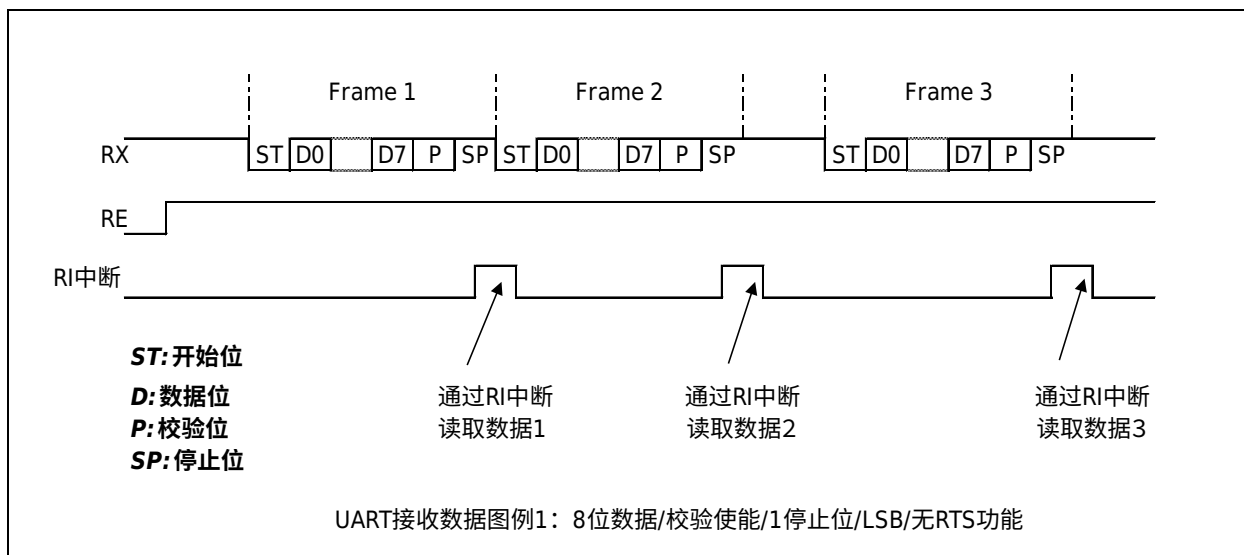


图 18-6 UART 接收数据图例 1

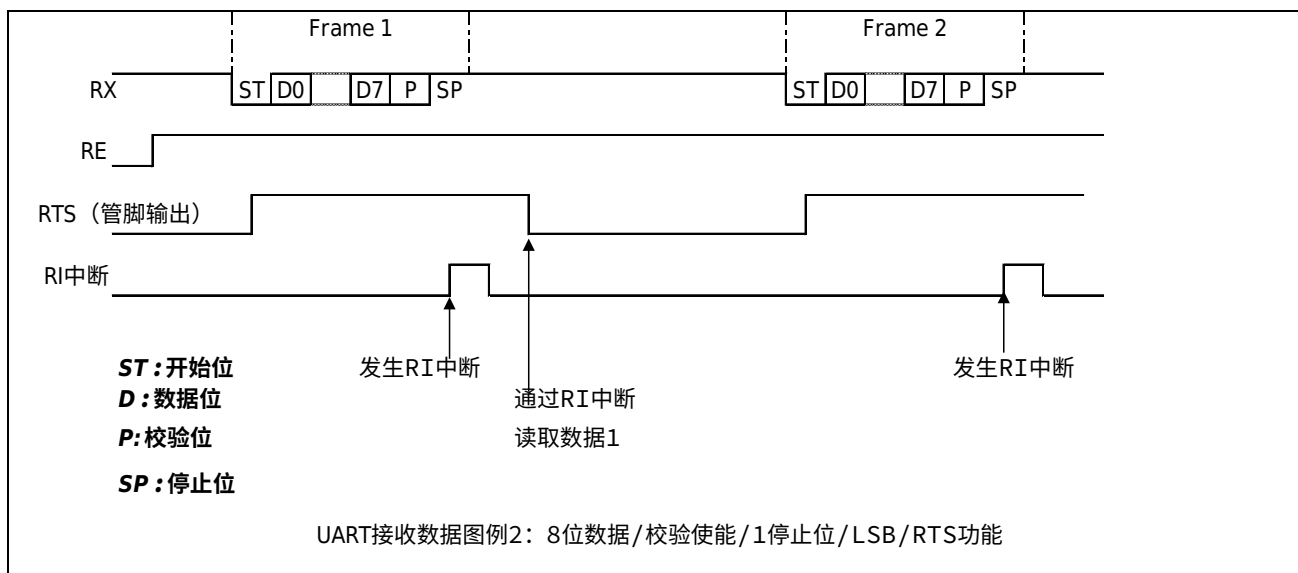


图 18-7 UART 接收数据图例 2

错误处理

接收数据时有三种类型的接收错误，分别为上溢错误（USARTn_SR.ORE），校验错误（USARTn_SR.PE）和帧错误（USARTn_SR.FE）。发生任何一种接收错误都不能再进行数据的接收。可以通过将所有的错误标志清零来重启数据接收，清零方法是写对应清零寄存器。

上溢错误发生的条件是 USARTn_DR.RDR 寄存器值未被读取的情况下又接收到一帧新的数据，所以应该在收到当前帧最后一位之前将接收到的前一帧数据读取。

校验错误发生的条件是发生了奇偶校验错误。

帧错误发生的条件是停止位为低电平，2 个停止位的情况也只检查第一个停止位。

发生上溢错误时接收到的数据丢失，RI 中断不发生。

发生校验错误时接收到的数据传送给 USARTn_DR.RDR，RI 中断不发生。

发生帧错误时接收到的数据传送给 USARTn_DR.RDR，RI 中断不发生。

接收器中断

UART 模式接收器支持两种中断，接收数据寄存器满中断 RI 和接收错误中断 REI。

USARTn_CR.RIE=1，未发生任何接收错误，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

USARTn_CR.RIE=1，接收过程中发生上溢错误，校验错误或者帧错误时 REI 中断发生。

18.4.1.6 UART 半双工通信方式

UART 模式支持单线半双工模式，UART 模式下通过设定 USARTn_CR3.HDSEL=1 启用单线半双工模式。

单线半双工模式时：

- TX 和 RX 线从内部相连接，不再使用 RX 管脚。
- 无数据传输时，TX 管脚处于释放状态。因此使用单线半双工模式时，通过上拉 TX 线来避免无数据传输时的浮空输入。

除此以外，半双工模式与正常 UART 模式通信相似。需要注意的时，发送过程不会被硬件封锁，只要数据在 USARTn_CR1.TE=1 时写入，发送就会进行。因此线路上的冲突必须有软件进行管理。

18.4.1.7 UART 中断和事件

表 18-3 UART 中断/事件表

功能名称	标记	使能位（仅中断）	标志位	可否作为事件源
接收错误中断	REI	RIE	ORE,FE,PE	不可
接收数据寄存器满中断	RI	RIE	RXNE	不可
发送数据寄存器空中断	TI	TXEIE	TXE	不可
发送完成中断	TCI	TCIE	TC	不可

18.4.2 多处理器通信

18.4.2.1 功能概述

多处理器通信模式是指多个处理器间共用通信线的一种通信方式，处理器分为发送站和接收站，每个接收站都有自己固有的 ID。发送站发送数据的类型有接收站 ID 和通信数据两种。通过在数据格式中添加 MPB 位来区分当前发送的是接收站的 ID 还是通信数据。MPB 位为 0 时当前帧为通信数据，MPB 位为 1 时当前帧为接收站的 ID。所有接收站都能接收发送站发送的 ID 并与自己的 ID 比较，如果一致，则接收数据，不一致则进入静默模式（既不接收数据也不置位接收相关标志）直到再次接收到 ID。

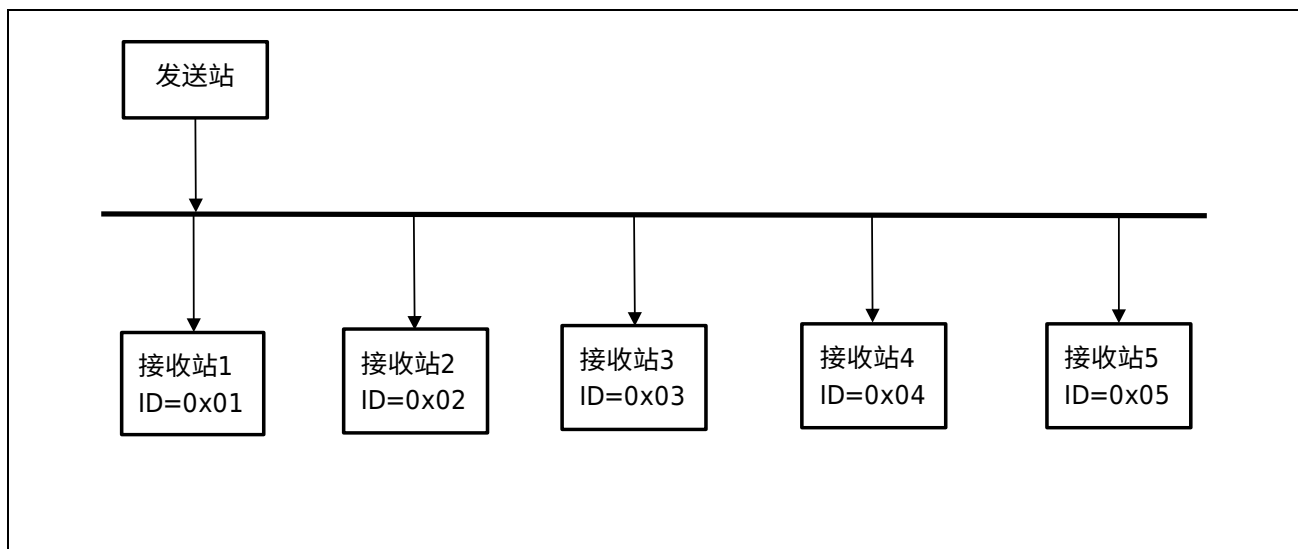


图 18-8 多处理器通信图例

18.4.2.2 数据格式

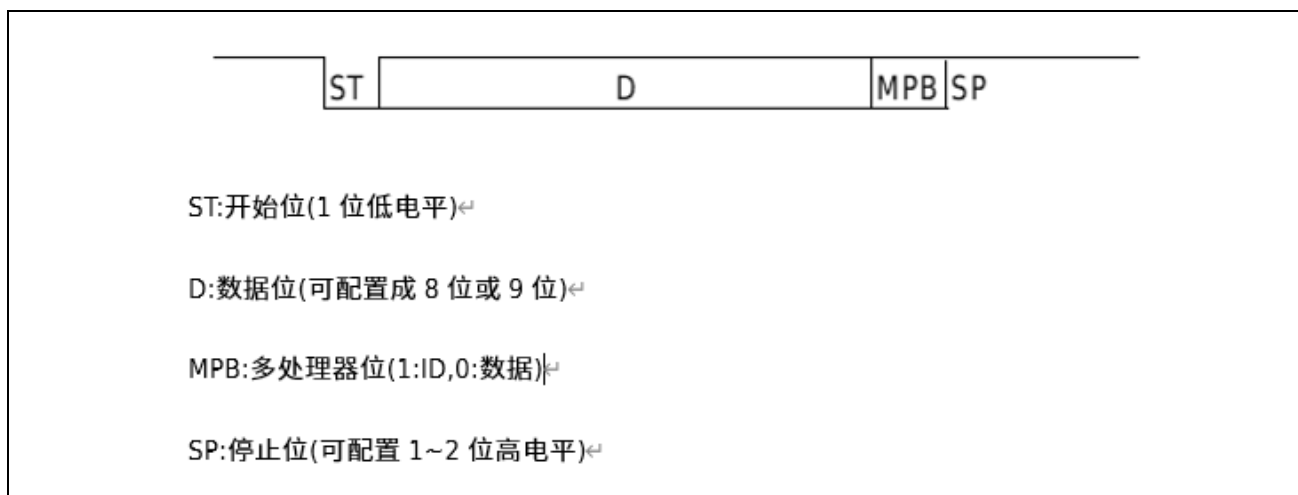


图 18-9 多处理器模式数据格式

18.4.2.3 动作说明

多处理器模式时校验位功能无效，增加了多处理器位功能，其余功能如时钟，中断等与 UART 模式相同。

发送站动作

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器

5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率 (时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1.TE=1), 如果需要使用发送数据寄存器空中断, 则设置 USARTn_CR1.TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空, 设定 USARTn_DR.MPID 位为 1 (发送 ID), 写 ID 值到 USARTn_DR, 发送 ID
 - (CTS 功能有效时, USARTn_CTSRTS 输入为低电平时数据传输到发送移位寄存器, 发送开始)
8. 设定 USARTn_DR.MPID 位为 0 (发送数据), 写数据到 USARTn_DR, 发送数据
 - (CTS 功能有效时, USARTn_CTSRTS 输入低电平时数据传输到发送移位寄存器, 发送开始)
9. 如果需要连续发送数据, 重复步骤 8, 如果需要改变 ID 后再发送数据重复 7 和 8
10. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况, 可通过 TI 中断写入最后一个发送数据, 并将 USARTn_CR1.TXEIE 写 0, USARTn_CR1.TCIE 写 1, 最后一帧数据发送结束后, 产生发送完成中断

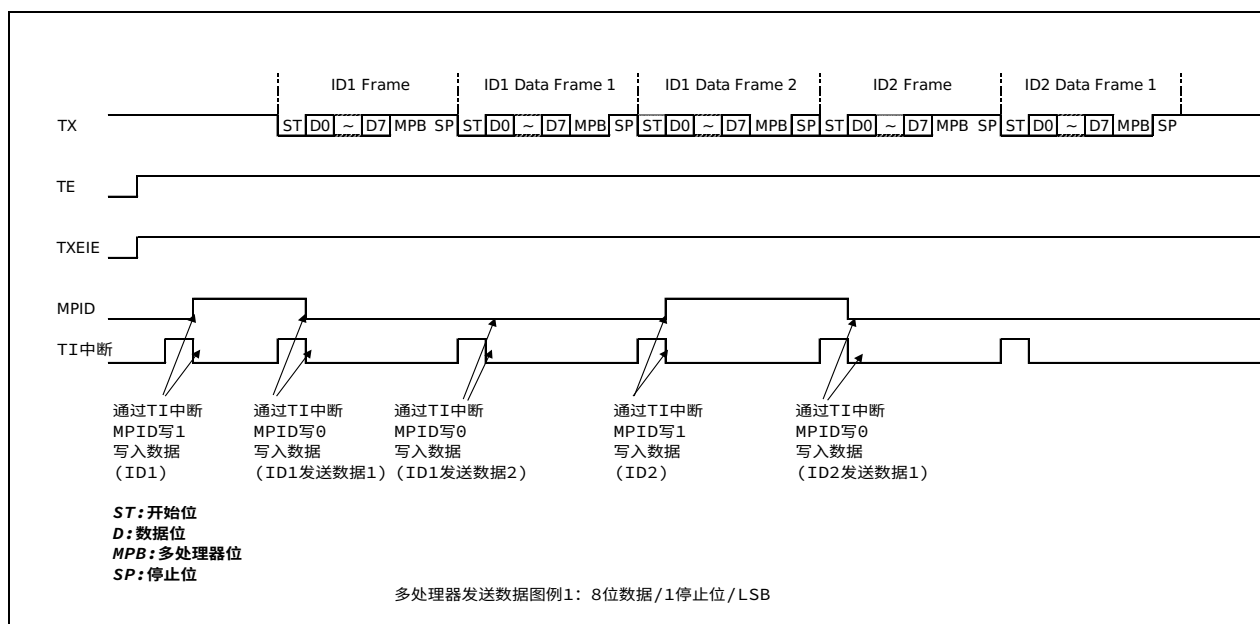


图 18-10 多处理器模式发送数据图例

接收站动作

在多处理器模式时, 接收站必须保证能收到每一个 ID 数据, 并与自身的 ID 比较, 如果一致则接收数据, 不一致则进入静默模式(不接收数据, 也不置位接收相关标志), 直到接收到下一个 ID 数据。通过 USARTn_CR1.SLME 位来实现这个功能。

USARTn_CR1.SLME=0 时正常接收数据。

USARTn_CR1.SLME=1 时，除非接收到 MPB 位为 1（ID）的数据，否则不接收数据，不发生 RI 中断，错误标志 FE,ORE 也不置位。当接收到 MPB 位为 1 的数据时（ID），USARTn_CR1.SLME 位自动清零，正常接收数据和发生中断。

动作步骤：

1. 将 USARTn_CR1 寄存器设定为复位值
 2. 设定所需要使用的管脚
 3. 通过 USARTn_CR1.CLK[1:0]位选择时钟源
 4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
 5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率（时钟源为外部时钟源时不需要设定）
 6. USARTn_CR1.RE=1, USARTn_CR1.SLME=1（等待接收 ID），如果使用接收中断，则设定 USARTn_CR1.RIE=1
 7. 当检测到开始位时，接收器将数据接收到接收移位寄存器，并检查 USARTn_SR.MPB 位
 8. 如果 USARTn_SR.MPB=1, USARTn_CR1.SLME 位自动清零，正常接收数据，软件比较接收的 ID 与自身的 ID
- 1) 如果 ID 一致，则正常接收数据，发生中断，进行错误检测，与 UART 接收数据相同
 - 2) 如果 ID 不一致，软件再次将 USARTn_CR1.SLME 位写 1，重复 8 的动作

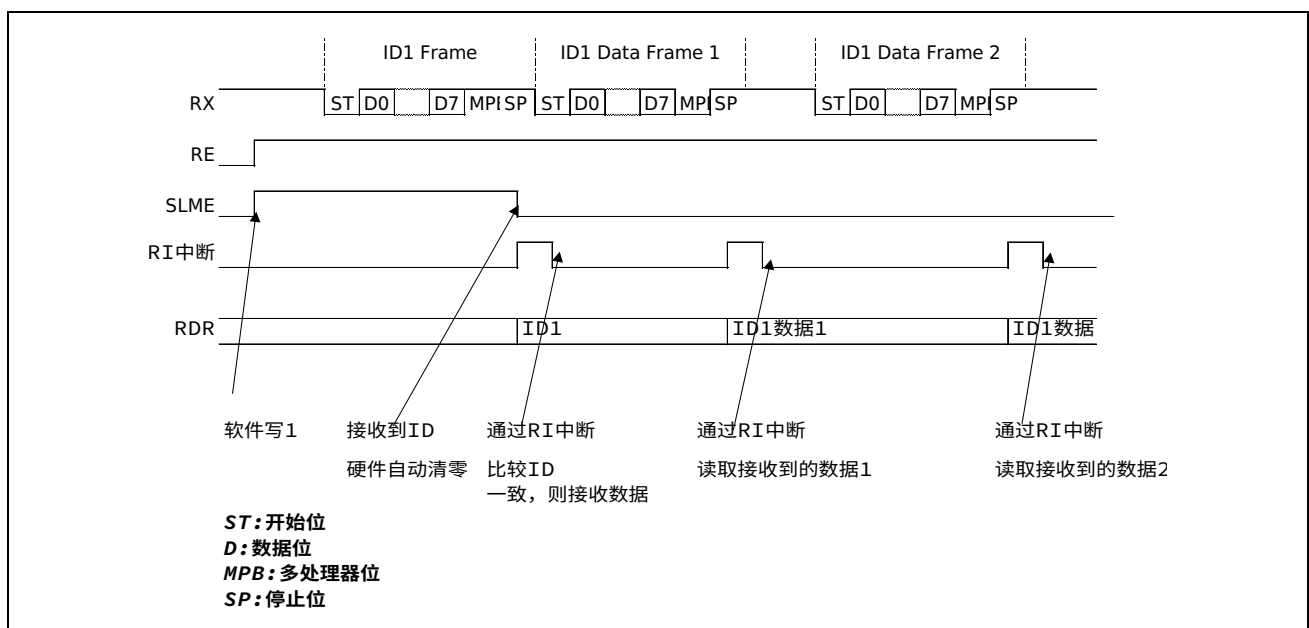


图 18-11 多处理器模式接收数据图例 1

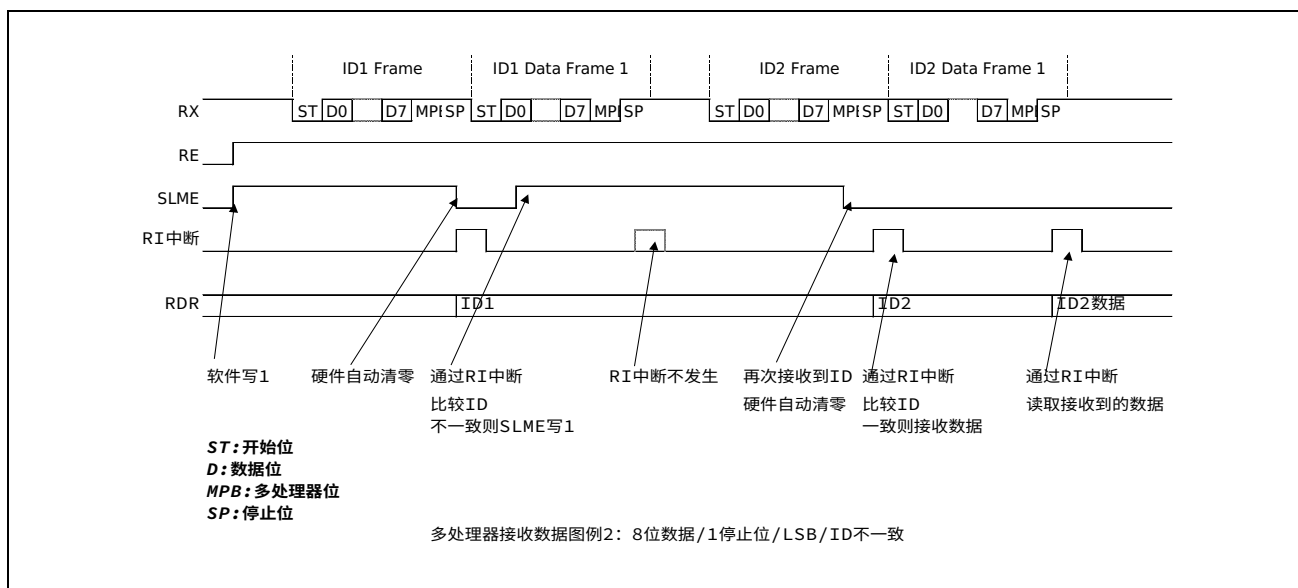


图 18-12 多处理器模式接收数据图例 2

18.4.2.4 中断和事件

多处理器模式除了无校验错误外，中断处理与 UART 模式相同。

表 18-4 多处理器模式中断/事件表

功能名称	标记	使能位（仅中断）	标志	可否作为事件源
接收错误中断	REI	RIE	ORE,FE	不可
接收数据寄存器满中断	RI	RIE	RXNE	不可
发送数据寄存器空中断	TI	TXEIE	TXE	不可
发送完成中断	TCI	TCIE	TC	不可

18.4.3 UART_LIN

18.4.3.1 功能概述

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。

本产品中采用 UART(Unit1)+TimerB(Unit4)+EIRQ(INTP1)的方式实现 LIN_bus 功能，其中 UART 用于数据传输，波特率设定等功能，TimerB 模块用于实现间隔场检测，同步场的测量以及波特率计算，而 EIRQ 用于唤醒信号的检测。

TimerB 模块的间隔场检测，同步场的测量功能无需设定 TimerB 模块对应的管脚，仅需要设定 UART 对应的管脚。

18.4.3.2 LIN 数据格式

LIN 的一帧数据是由开始位+8 位数据+1 停止位组成，以 LSB 方式发送和接收数据。LIN 总线上的数据行为如下图所示。

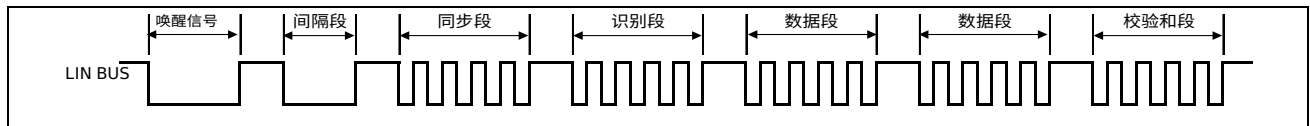


图 18-13 LIN 总线数据行为

18.4.3.3 LIN 发送

唤醒信号帧的发送

设置合适的波特率，通过发送数据 0x80 对应信号帧的发送。

间隔段发送

同步间隔段表示一帧数据的开始，至少持续 13 位（以主机节点的速率为准）的显性电平。本产品中采用发送数据 0x00 的方式以实现 13 位显性电平的间隔时间。

$$(\text{间隔段的波特率}) = \frac{9 \times B}{13}$$

B：主机节点的波特率 单位：bps

同步段发送

通过发送数据 0x55 实现同步段数据的发送

发送数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值

（以下步骤 6~9 为唤醒信号帧的发送，如无必要则跳过）

6. 设定 USARTn_BRR 寄存器，设定唤醒信号帧的波特率
7. 设定 USARTn_CR1.TE=1
8. 写通信数据 0x80 到 USARTn_DR.TDR，数据传输到发送移位寄存器，发送开始

(CTS 功能有效时, USARTn_CTSRTS 输入为低电平时数据传输到发送移位寄存器, 发送开始)

9. 等待发送完成

(以下步骤 10~13 为同步间隔段的发送, 需要改变波特率)

10. 设定 USARTn_CR1.TE=0

11. 设定 USARTn_BRR 寄存器, 设定同步间隔段的波特率, 波特率计算请参考【间隔段发送】中的描述

12. 写通信数据 0x00 到 USARTn_DR.TDR, 数据传输到发送移位寄存器, 发送开始

(CTS 功能有效时, USARTn_CTSRTS 输入为低电平时数据传输到发送移位寄存器, 发送开始)

13. 等待发送完成

(以下步骤 14~15 为同步场的发送, 需要重新设定波特率, 其它过程与 UART 发送过程相同)

14. 设定 USARTn_CR1.TE=0

15. 重新设定通信波特率

16. 发送同步段

17. 发送识别段

18. 发送数据

19. 发送校验和 (校验和由软件实现)

18.4.3.4 LIN 接收

唤醒信号帧的检测

通过 EIRQ(INTP1)检测 RX 下降沿产生中断来实现唤醒信号的检测。

同步间隔段的检测

通过 TimerB (Unit4) 的脉宽测量功能来实现同步段的检测。

检测到同步间隔段(RX)的下降沿, TimerB 开始测量低电平宽度, 并且在同步间隔段(RX)的上升沿进行捕捉。根据捕捉到的值判断是否为同步段信号。判定基准为低电平宽度是否大于等于 11 位通信数据周期宽度。

(TIMEB 详细设定请参考 TIMEB 脉宽测量章节)

同步段的检测与波特率测量

通过 TimerB (Unit4) 的周期测量功能来实现同步段的检测和测量。连续测量 RX 的 5 个下降沿间的周期，将测量的 4 个周期值累加，累加值记为 A，则通信波特率=TimerB 计数时钟频率÷ (A÷8)。

(TIMEB 详细设定请参考 TIMEB 周期测量章节)

接收数据设定步骤

1. 将 USARTn_CR1 寄存器设定为复位值
2. 设定 UART 所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1(RE=1,RIE=0), USARTn_CR2, USARTn_CR3 寄存器，并使能数字滤波

(以下步骤 5~6 为唤醒信号帧的检测，如无必要则跳过)

5. 设定中断模块，EIRQ (INTP1) 设定为下降沿检测，等待唤醒信号
6. 检测到唤醒信号，设定 TimerB 为脉宽测量功能，等待同步段间隔场

(以下步骤 7 为同步段间隔场检测)

7. TimerB 检测到的 RX 低电平宽度大于等于 11 位数据宽度即为检测到同步间隔场，则进入同步场测量。否则继续等待同步间隔场

(以下步骤 8~11 为同步段周期测量与通信波特率计算)

8. 检测到同步间隔场后，设定 TimerB (Unit4) 为周期测量。连续测量 RX 5 个下降沿的周期宽度
9. 将步骤 8 测量到 4 个值累加后除以 8 即为需要设定的波特率宽度
10. 设定 USARTn_CR1.RE=0, USARTn_CR1.TE=0, 清除通信错误标志 (USARTn_SR.ORE/USARTn_SR.PE / USARTn_SR.FE), 读取 USARTn_DR, 并忽略接收的数据
11. 设定 USARTn_BRR 寄存器

(以下步骤 12~15 为接收识别段，数据以及校验和，过程与 UART 接收数据过程相同)

12. 使能接收器 (USARTn_CR1. RE=1)，如果需要使用接收中断，则设置 USARTn_CR1. RIE=1
13. 接收识别段
14. 接收数据
15. 接收校验和

18.4.4 时钟同步模式

18.4.4.1 时钟

时钟同步模式可以选择内部波特率生成器生成的时钟(内部时钟源)或 USARTn_CK 管脚输入的时钟(外部时钟源)作为通信的时钟源。

内部时钟源

同步时钟从 USARTn_CK 管脚输出，一帧数据输出 8 个时钟脉冲，既不发送数据也不接收数据时，时钟输出固定为高电平。

外部时钟源

外部时钟源即从 USARTn_CK 管脚输入时钟作为通信时钟。

最高波特率

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$B = \frac{C}{4 \times (\text{DIV\_Integer} + 1)}$$

B: 波特率 单位: Mbps

C: USARTn_PR.PSC[1:0]位设定的时钟 (HCLK,HCLK/4,HCLK/16,HCLK/64) 单位: MHz

DIV_Integer:USARTn_BRR.DIV_Integer 设定值

内部时钟源时，当 C 为 HCLK, DIV_Integer=1 时，最高波特率为 HCLK/8(Mbps)。注意同步模式时 DIV_Integer 禁止设置为 0。

外部时钟源时，外部输入时钟的最大频率要求为 HCLK(MHz)/6，所以最高波特率为 HCLK/6(Mbps)。

需要注意的是，同步模式最高通信波特率除了以上描述的基于 HCLK 的计算方法外，还需要参考**数据手册电气特性**章节规定的最高通信波特率。

18.4.4.2 数据格式

时钟同步模式一帧数据固定有 8 位组成，一帧数据的发送和接收需要 8 个同步时钟脉冲。发送数据时数据在同步时钟的下降沿送出，接收数据时数据在同步时钟的上升沿被采样。

同步时钟在没有数据传输时固定为高电平，最后一位发送完后，通信线保持最后一位的值。

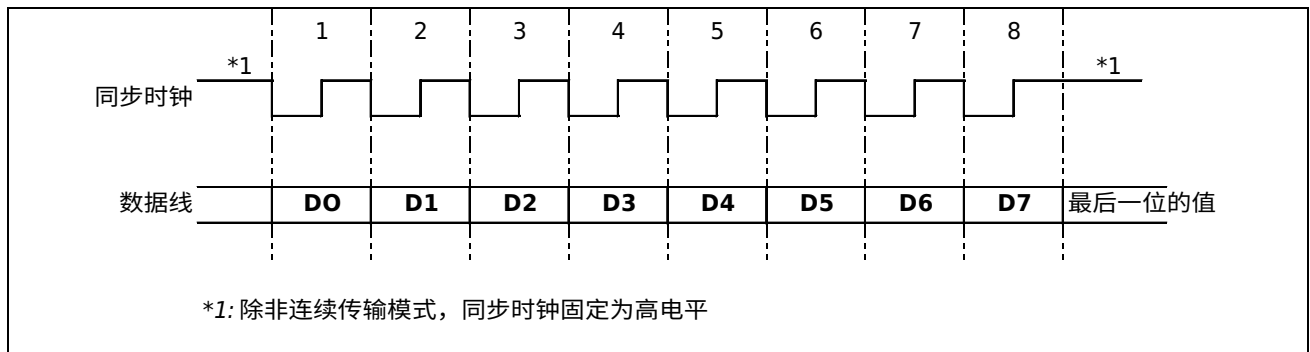


图 18-14 时钟同步模式数据格式

18.4.4.3 调制解调器操作

调制解调器操作包括 CTS 功能和 RTS 功能。CTS 功能和 RTS 功能只能二选一，不能同时使用。USARTn_CR3.CTSE=0 时 RTS 功能有效，USARTn_CR3.CTSE=1 时 CTS 功能有效。

CTS 功能

CTS 功能是通过 USARTn_CTSRTS 管脚的输入来控制数据的发送，只有当 USARTn_CTSRTS 管脚输入低电平时才可以发送数据，发送数据过程中如果 USARTn_CTSRTS 输入高电平，正在发送的数据不受影响。

RTS 功能

RTS 功能是指通过 USARTn_CTSRTS 管脚输出低电平，请求对方发送数据。

USARTn_CTSRTS 管脚输出低电平需要满足以下全部条件：

- 接收使能(USARTn_CR1.RE=1)，且不在接收数据
- USARTn_DR.RDR 寄存器中没有未读取的数据(USARTn_CR1.RE=1 时)
- USARTn_DR.TDR 更新完成(USARTn_CR1.TE=1 时)
- 无任何接收错误

如果不能同时满足以上全部条件，USARTn_CTSRTS 则输出高电平。

18.4.4.4 发送器

发送器使能位 (USARTn_CR1.TE) 置 1 时，发送移位寄存器中的数据在 USARTn_TX 管脚串行输出，相应的时钟脉冲在 USARTn_CK 管脚输出。

发送数据寄存器 USARTn_DR.TDR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。

通过发送数据寄存器空中断写入发送数据时，为保证发送的正确性，一次请求只能写入一次数据。

发送数据设定步骤

1. 将 USARTn_CR1, USARTn_SR1 寄存器设定为复位值

2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLK[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值, USARTn_BRR 寄存器设定通信波特率 (时钟源为外部时钟源时不需要设定)
6. 使能发送器 (USARTn_CR1.TE=1), 如果需要使用发送数据寄存器空中断, 则设置 USARTn_CR1.TXEIE=1 (TE 和 TXEIE 位同时写入 1)
7. 等待发送数据寄存器空, 写通信数据到 USARTn_DR.TDR, 数据传输到发送移位寄存器, 发送开始 (CTS 功能有效时, USARTn_CTSRTS 输入为低电平时数据传输到发送移位寄存器, 发送开始)
8. 如果需要连续发送数据时, 重复步骤 7
9. 通过确认 USARTn_SR.TC 位确认发送是否完成。连续发送数据并使用发送中断的情况, 可通过 TI 中断写入最后一个发送数据, 并将 USARTn_CR1.TXEIE 写 0, USARTn_CR1.TCIE 写 1, 最后一个数据发送结束后, 产生发送完成中断

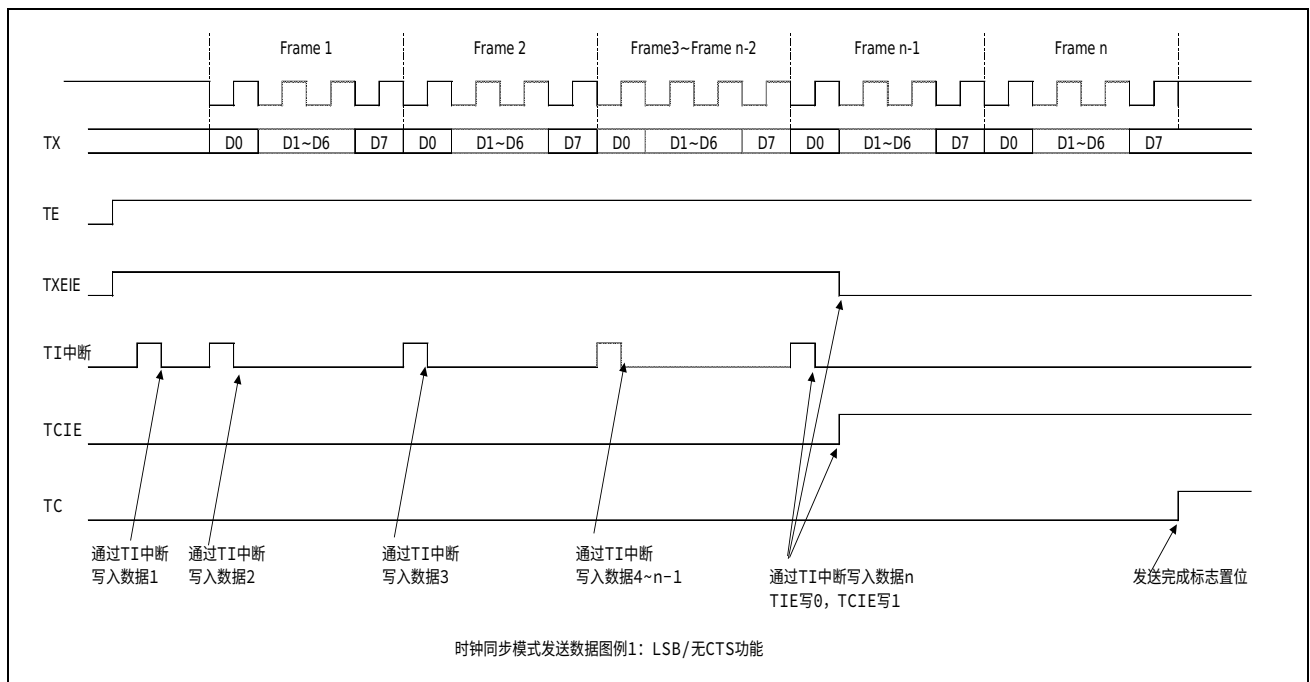


图 18-15 时钟同步模式发送数据图例 1

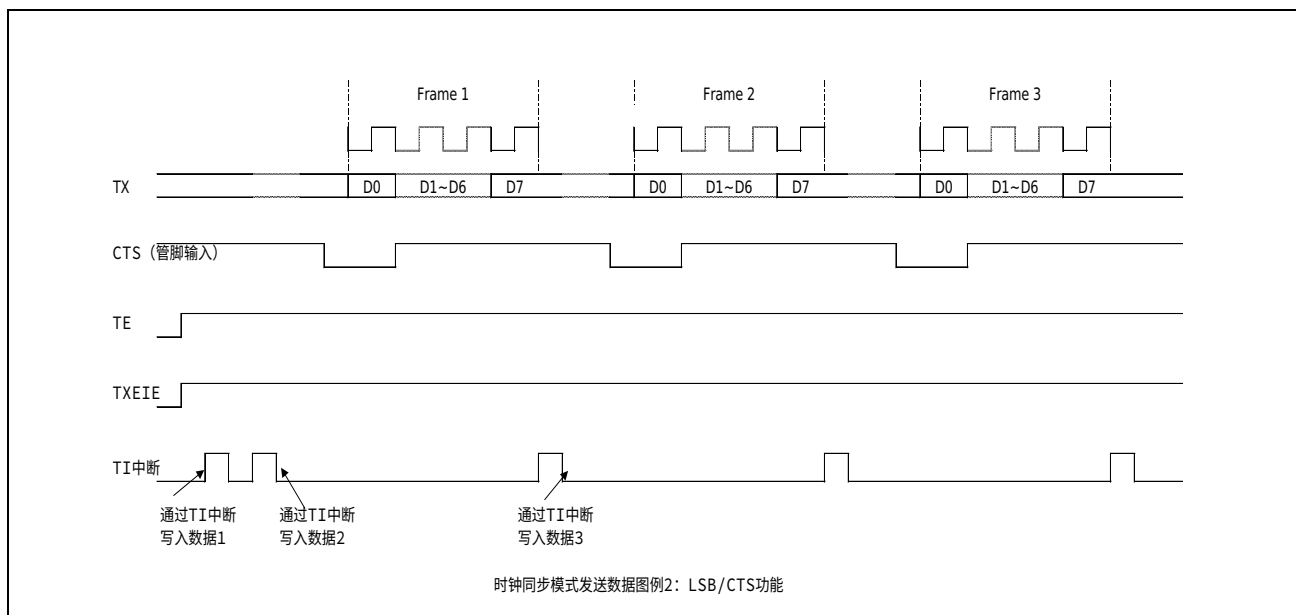


图 18-16 时钟同步模式发送数据图例 2

发送器中断

时钟同步模式发送器支持两种中断，发送数据寄存器空中断 TI 和发送完成中断 TCI。

TXEIE=1，USARTn_DR.TDR 寄存器的值传送到发送移位寄存器时 TI 中断发生。

TCIE=1，发送数据的最后一位时 USARTn_DR.TDR 寄存器没有更新则 TCI 中断发生。

18.4.4.5 接收器

接收数据设定步骤

1. 将 USARTn_CR1, USARTn_SR 寄存器设定为复位值
2. 设定所需要使用的管脚
3. 通过 USARTn_CR2.CLKC[1:0]位选择时钟源
4. 设定 USARTn_CR1, USARTn_CR2, USARTn_CR3 寄存器
5. 设定 USARTn_PR 选择预分频值，USARTn_BRR 寄存器设定通信波特率(时钟源为外部时钟源时不需要设定)
6. 使能接收器 (USARTn_CR1. RE=1)，如果需要使用接收中断，则设置 USARTn_CR1. RIE=1
(使用 RTS 功能时，RE=1 后 USARTn_CTSRTS 输出低电平)
7. 同步于输入同步时钟或者内部生成的同步时钟开始接收数据，接收数据到接收移位寄存器
 - 1) 发生上溢错误时，数据丢失并置位 USARTn_SR.ORE 标志

- 2) 无错误发生时, 接收到的数据传送到 USARTn_DR.RDR 寄存器中, 置位 USARTn_SR.RXNE 标志, 将当前接收到的数据在接收到下一帧数据最后一位前读取后重复步骤 7 可实现连续接收数据功能。

(使用 RTS 功能时, 数据读取后 USARTn_CTSRTS 输出低电平)

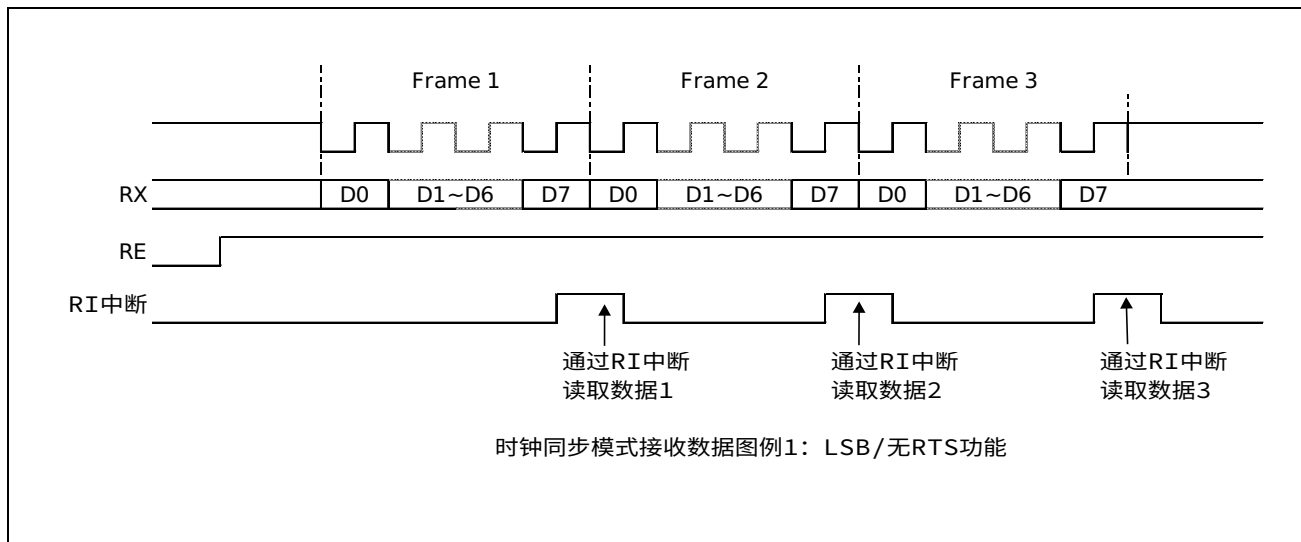


图 18-17 时钟同步模式接收数据图例 1

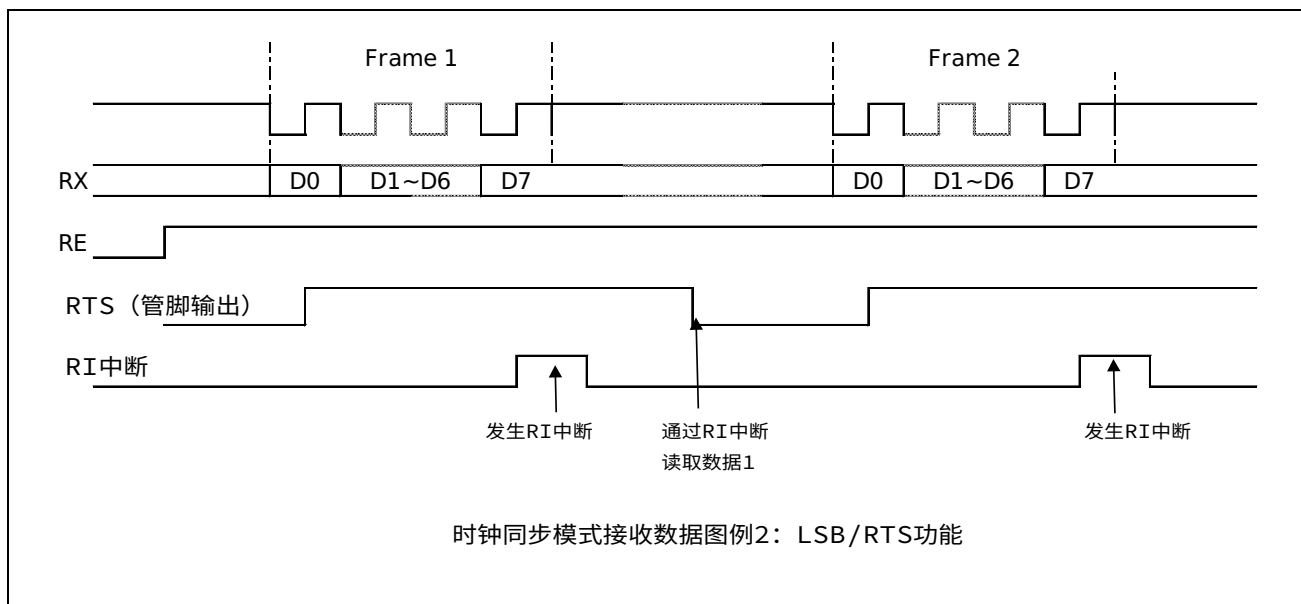


图 18-18 时钟同步模式接收数据图例 2

错误处理

时钟同步模式接收数据时接收错误为上溢错误 (USARTn_SR.ORE)。发生接收错误不能再进行数据的接收和发送。可以通过将错误标志清零来重启数据传输, 清除方法是写对应的清零寄存器。

上溢错误发生的条件是 USARTn_DR.RDR 值未被读取的情况下又收到新的数据, 所以应该在收到当前帧最后一位之前将前一帧收到的数据读取。发生上溢错误时接收到的数据丢失, RI 中断不发生。

接收器中断

时钟同步模式接收器支持两种中断，接收数据寄存器满中断 RI 和接收错误中断 REI。

RIE=1，数据从接收移位寄存器传送到接收数据寄存器时 RI 中断发生。

RIE=1，接收数据发生错误（上溢错误）时 REI 中断发生。

18.4.4.6 同时发送接收数据

USART 时钟同步模式支持全双工动作，同时发送接收数据。同时发送接收数据时需要一个命令把 RE,TE,RIE,TXEIE 写 1，其它设定流程与发送器和接收器相同。

18.4.4.7 时钟同步模式中断和事件

表 18-5 时钟同步模式中断/事件表

中断名称	标记	使能位(仅中断)	标志位	可否作事件源
接收错误中断	REI	RIE	ORE	不可
接收数据寄存器满中断	RI	RIE	RXNE	不可
发送数据寄存器空中断	TI	TXEIE	TXE	不可
发送完成中断	TCI	TCIE	TC	不可

18.4.5 数字滤波功能

USARTn_CR1.NFE=1 时，内置数字滤波器功能有效。数字滤波器仅在 UART 模式时有效，可以除去接收数据线 RX 上的噪音。

内置数字滤波器可以滤除小于一位数据的 3/16(USARTn_CR1.OVER8=0) 宽度或者 3/8 宽度 (USARTn_CR1.OVER8=1)的噪音。

如果数字滤波器的时钟停止后再开始时，数字滤波器从时钟停止时保持的状态继续工作。

USARTn_CR.TE=0 且 USARTn_CR.RE=0，将数字滤波器内部的 Flip-Flop 状态复位为 1。

18.5 寄存器描述

表 18-6 USART 基地址

名称	基地址	描述
USART1	0x4000 1800	USART1 基地址
USART2	0x4000 1C00	USART2 基地址
USART3	0x4000 2000	USART3 基地址
USART4	0x4000 2400	USART4 基地址
USART5	0x4000 2800	USART5 基地址
USART6	0x4000 2C00	USART6 基地址

表 18-7 USART 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
USART_SR	状态寄存器	0x00	32	0x0000 00C0
USART_TDR	发送数据寄存器	0x04	16	0x01FF
USART_RDR	接收数据寄存器	0x06	16	0x0000
USART_BRR	波特率寄存器	0x08	32	0x0000 FF00
USART_CR1	控制寄存器 1	0x0C	32	0x8000 0000
USART_CR2	控制寄存器 2	0x10	32	0x0000 0000
USART_CR3	控制寄存器 3	0x14	32	0x0000 0000
USART_PR	预分频寄存器	0x18	32	0x0000 0000

18.5.1 状态寄存器 (USART_SR)

复位值: 0x0000 00C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								TXE	TC	RXNE	Res	ORE	Res	FE	PE

位/位域	标记	位名	功能	读写
b31~b17	Res	保留位	读出时为“0”,写入时写“0”	R
b16	MPB	多处理器位	多处理器位标志 0: 当前接收数据为通信数据 1: 当前接收数据为 ID 注意: MPB 位只在多处理器模式时有效	R
b15~b8	Res	保留位	读出时为“0”,写入时写“0”	R
b7	TXE	发送数据寄存器空	发送数据寄存器空标志 0: 数据未传输到移位寄存器, 发送数据寄存器非空 1: 数据传输到移位寄存器, 发送数据寄存器空 注意: TXE 位由硬件置 1 和清 0, 数据未传输到移位寄存器时硬件将 TXE 清 0, 数据传输到移位寄存器时硬件将 TXE 置 1	R
b6	TC	发送完成标志	发送完成标志位 0: 发送数据中 1: 发送数据完成 TC 置位条件 - TE=0 发送禁止时 - 送出一帧数据的最后一位时, 发送数据寄存器的值没有被更新 TC 清零条件 - TE=1 时, 向发送数据寄存器写入发送数据 注意: TE 位由 0 变化为 1 时 TC 为保持为 1	R

位/位域	标记	位名	功能	读写
b5	RXNE	接收数据寄存器不为空	接收数据寄存器不为空标志 0: 未接收到数据 1: 准备好读取接收到的数据 注意: RXNE 位由硬件置 1 和清 0, 准备好读取接收到的数据时硬件将 RXNE 置 1, 读取接收数据后硬件将 RXNE 清 0	R
b4	Res	保留位	读出时为“0”,写入时写“0”	R
b3	ORE	接收上溢错误	接收上溢错误标志位 0: 无接收上溢错误 1: 发生接收上溢错误 ORE 置位条件 • 接收数据寄存器未被读取的情况下, 又接收到一帧新的数据 ORE 清零条件 • 清零寄存器 CR1.CORE 位写入 1 注意: RE=0 并不能复位 ORE 位 ORE=1 之前接收到的数据会被保持, ORE=1 时接收到的数据会丢掉 ORE=1 后不能继续接收数据, 时钟同步模式下也不能发送数据	R
b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1	FE	接收帧错误	接收帧错误标志位 0: 无接收帧错误 1: 发生接收帧错误 FE 置位条件 • 接收的数据帧的停止位为低电平, 两个停止位的情况只检查第一个停止位 FE 清零条件 • 清零寄存器 CR1.CFE 位写入 1 注意: UART 模式时: RE=0 并不能复位 PE 位 FE=1 时收到的数据会保留但是 RI 中断不会发生, FE=1 后不能继续接收数据	R
b0	PE	接收数据校验	接收数据校验错误标志 0: 无接收数据校验错误 1: 发生接收数据校验错误 PE 置位条件 • 接收数据发生奇偶校验错误时 PE 清零条件 • 清零寄存器 CR1.CPE 位写入 1 注意: RE=0 并不能复位 PE 位 PE=1 时收到的数据会保留但是 RI 中断不会发生, PE=1 后不能继续接收数据	R

18.5.2 发送数据寄存器 (USART_TDR)

复位值: 0x01FF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						MPID	TDR[8:0]								

位/位域	标记	位名	功能	读写
b15~b10	Res	保留位	读出时为“0”,写入时写“0”	R
b9	MPID	多处理器模式 ID 位	多处理器模式时, 发送通信数据或者发送 ID 的选择位 0: 发送数据 1: 发送 ID 注意: MPID 位只在多处理器模式时有效,其他模式须设定为复位值	RW
b8~b0	TDR[8:0]	发送数据寄存器	发送数据寄存器 注意: 最高位 TDR[8]仅在 UART 模式而且数据长度设定为 9 位时有效	RW

18.5.3 接收数据寄存器 (USART_RDR)

复位值: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						RDR[8:0]									

位/位域	标记	位名	功能	读写
b15~b9	Res	保留位	读出时为“0”,写入时写“0”	R
b8~b0	RDR[8:0]	接收数据寄存器	接收数据寄存器 注意: 最高位 RDR[8]仅在 UART 模式而且数据长度设定为 9 位时有效	R

18.5.4 波特率寄存器 (USART_BRR)

复位值: 0x0000 FF00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_Integer [7:0]								Res							

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	R
b15~b8	DIV_Integer[7:0]	整数分频寄存器	整数分频寄存器 注意: DIV_Integer[7:0]只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	R

表 18-8 波特率计算公式

模式	波特率计算公式	误差 E(%)计算公式
UART 模式 多处理器模式	$B = \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1)}$	$E(\%) = \left\{ \frac{C}{8 \times (2 - OVER8) \times (DIV\_Integer + 1) \times B} - 1 \right\} \times 100$
时钟同步模式	$B = \frac{C}{4 \times (DIV\_Integer + 1)}$	-

B: 波特率 单位: Mbps

C: PR.PSC[1:0]位设定的时钟 单位: MHz

18.5.5 控制寄存器 1 (USART_CR1)

复位值: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	Res	ML	Res	MS	Res	CORE	Res	CFE	CPE					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER8	Res	M	Res	PCE	PS	Res	TXEIE	TCIE	RIE	SLME	TE	RE	Res		

位/位域	标记	位名	功能	读写
b31	SBS	UART 模式接收数据开始位检测方式设定位	UART 模式接收数据时, 开始位检测方式设定位 0: 开始位检测方式为 RX 管脚低电平 1: 开始位检测方式为 RX 管脚下降沿 注意: 非 UART 模式时 SBS 位须保持复位值 SBS 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b30	NFE	数字滤波使能位	数字滤波使能位 0: 禁止数字滤波功能 1: 使能数据滤波功能 注意: 非 UART 模式时 NFE 位必须保持复位值 NFE 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b29	Res	保留位	读出时为“0”,写入时写“0”	R
b28	ML	MSB/LSB 选择位	UART 模式/时钟同步模式时, MSB/LSB 方式选择位 0: LSB 方式 1: MSB 方式 注意: ML 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b27~b25	Res	保留位	读出时为“0”,写入时写“0”	R
b24	MS	通信模式选择位	通信模式选择位 0: UART 模式 1: 时钟同步模式 注意: MS 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b23~b20	Res	保留位	读出时为“0”,写入时写“0”	R
b19	CORE	ORE 标志清零位	ORE 标志清零位 0: 不清零 ORE 标志 1: 清零 ORE 标志 注意: CORE 位写 1 清除 ORE 标志, 读取时返回 0	RW
b18	Res	保留位	读出时为“0”,写入时写“0”	R
b17	CFE	FE 标志清零位	FE 标志清零位 0: 不清零 FE 标志 1: 清零 FE 标志 注意: CFE 位写 1 清除 FE 标志, 读取时返回 0	RW
b16	CPE	PE 标志清零位	PE 标志清零位 0: 不清零 PE 标志 1: 清零 PE 标志 注意: CPE 位写 1 清除 FE 标志, 读取时返回 0	RW
b15	OVER8	UART 过采样模式	UART 过采样模式设定, 即一位数据传输期间的基本时钟数 0: 16 位	RW

位/位域	标记	位名	功能	读写
			1: 8 位 注意: 非 UART 模式时 OVER8 位必须保持复位值 OVER8 位只能在 TE=0&RE=0(发送/接收禁止)时设定	
b14~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	M	数据长度设定位	UART 模式时, 发送/接收数据长度设定位 0: 8 位 1: 9 位 注意: 非 UART 模式时 M 位必须保持复位值 M 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b11	Res	保留位	读出时为“0”,写入时写“0”	R
b10	PCE	校验使能位	UART 模式时, 奇偶校验使能位 0: 无校验 1: 校验 注意: 时钟同步模式时 PCE 位必须保持复位值 PCE 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b9	PS	校验位	UART 模式时, 奇偶校验选择位 0: 偶校验 1: 奇校验 注意: PS 位只能在 TE=0&RE=0(发送/接收禁止)时设定, PS 位只在 PCE=1 时有有效	RW
b8	Res	保留位	读出时为“0”,写入时写“0”	R
b7	TXEIE	发送数据寄存器空中断使能位	发送数据寄存器空中断使能位 0: TI 中断请求无效, TI 中断不发生 1: TI 中断请求有效, TI 中断发生 注意: TXEIE 位和 TE 位应同时写入 1	RW
b6	TCIE	发送完成中断使能位	发送完成中求使能位 0: TCI 中断请求无效, TCI 中断不发生 1: TCI 中断请求使能, TCI 中断发生	RW
b5	RIE	接收中断使能位	接收中断使能位 0: 接收中断请求无效, RI 和 REI 中断不发生 1: 接收中断请求有效, RI 和 REI 中断发生	RW
b4	SLME	静默模式使能位	多处理器操作时, 静默模式使能位 0: 正常模式 1: 静默模式 SLME=1 时, MPB 位为 0 的通信数据不会从接收移位寄存器读取到接收数据寄存器, 同时错误标志 ORE 和 FE 位也不置位。接收到 MPB 为 1 的 ID 数据时, SLME 自动清零, 开始正常的数据接收动作。 注意: SLME 位只在 UART 多处理器模式时有效, 其他模式时此位必须保持复位值。	RW
b3	TE	发送器使能位	发送器使能位 0: 发送器禁止 1: 发送器使能 注意: 时钟同步模式时 TE 位只能在 TE=0&RE=0(发送/接收禁止)时写 1。	RW
b2	RE	接收器使能位	接收器使能位 0: 接收器禁止 1: 接收器使能 注意: 时钟同步模式时 RE 位只能在 TE=0&RE=0(发送/接收禁止)时写 1	RW

位/位域	标记	位名	功能	读写
b1~b0	Res	保留位	读出时为“0”,写入时写“0”	R

18.5.6 控制寄存器 2 (USART_CR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	LINE N	STOP	CLKC[1:0]	Res										MPE	

位/位域	标记	位名	功能	读写
b31~b15	Res	保留位	读出时为“0”,写入时写“0”	R
b14	LINEN	LIN 功能使能	UART 模式时, LIN 功能使能位 0: 禁止 LIN 功能 1: 使能 LIN 功能 注意: LINEN 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b13	STOP	停止位设定位	UART 模式时, 停止位长度设定位 0: 1 停止位 1: 2 停止位 注意: 非 UART 模式时 STOP 位必须保持复位值 STOP 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b12~b11	CLKC[1:0]	时钟控制位	UART 模式 00b: 时钟源为内部波特率生成器生成的时钟, 时钟不输出到 USARTn_CK 管脚, USARTn_CK 管脚可以当作普通 IO 使用 01b: 时钟源为内部波特率生成器生成的时钟, 时钟输出到 USARTn_CK 管脚, 输出时钟频率和波特率相同 10b or 11b: 时钟源为外部输入时钟, 输入时钟的频率为波特率的 16 倍(OVER8=0) 或者 8 倍(OVER8=1) 时钟同步模式 00b or 01b: 时钟源为内部波特率生成器生成的时钟, 输出到 USARTn_CK 管脚 10b or 11b: 时钟源为外部输入时钟, 输入时钟的频率和波特率相同 注意: CLKC[1:0]位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b10~b1	Res	保留位	读出时为“0”,写入时写“0”	R
b0	MPE	多处理器功能使能位	UART 模式时, 多处理器功能使能位 0: 禁止 1: 使能 注意: 非 UART 模式时 MPE 位必须保持复位值 MP 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW

18.5.7 控制寄存器 3 (USART_CR3)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						CTSE	Res					HDSE L	Res		

位/位域	标记	位名	功能	读写
b31~b10	Res	保留位	读出时为“0”,写入时写“0”	R
b9	CTSE	CTS 功能使能位	CTS 功能使能位 0: RTS 功能 1: CTS 功能 注意: CTSE 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b8~b4	Res	保留位	读出时为“0”,写入时写“0”	R
b3	HDSEL	UART 单线半双工模式使能位	UART 单线半双工模式使能位 0: UART 全双工模式 1: UART 半双工模式 注意: HDSEL 位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW
b2~b0	Res	保留位	读出时为“0”,写入时写“0”	R

18.5.8 预分频寄存器 (USART_PR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														PSC[1:0]	

位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1~b0	PSC[1:0]	预分频器值	内部时钟源时, 预分频器分频值选择位 00: HCLK 01: HCLK/4 10: HCLK/16 11: HCLK/64 注意: PSC[1:0]位只能在 TE=0&RE=0(发送/接收禁止)时设定	RW

18.6 使用注意事项

18.6.1 UART 注意事项

发送器

UART 模式发送器发送禁止时(USARTn_CR1.TE=0)，则 TX 管脚可以当作普通 IO 使用，可以设定输出值和方向。如果输出 0，则会使接收方产生帧错误，从而中断数据传输。如果输出 1，则使接收方检测不到开始位从而无法开始数据传输。

接收器

UART 模式产生帧错误时，可以软件检测后续 RX 线是否为低电平，从而判断发送方是否想中断传输。

如果接收数据开始位检测方式为低电平检测，则在清除错误标志后继续接收全为低电平数据，接收错误会再次发生。

18.6.2 时钟同步模式注意事项

- 使用外部输入时钟发送数据时，USARTn_DR.TDR 的更新需要在时钟输入之前完成，写入数据后，至少需要等待一位数据时间再输入时钟。
- 连续发送数据时，下一帧数据需要在当前帧最后一位发送前完成更新。

18.6.3 其他注意事项

为了防止发送禁止时 TX 通信线 Hi-Z 状态，可以采用以下方法：

- 通信线上拉
- 发送数据结束时，USARTn_CR1.TE=0 之前，将 TX 管脚设为普通 IO 输出
- 发送数据开始前，USARTn_CR1.TE=1 之后，将 IO 设为 TX 功能

19 集成电路总线（I2C）

19.1 概述

I2C（集成电路总线）用作微控制器和 I2C 串行总线之间的接口。提供多主模式功能，可以控制所有 I2C 总线的协议、仲裁。支持标准模式、快速模式。还支持 SMBus 总线。

I2C 主要特性：

- I2C 总线方式、SMBUS 总线方式可选。主机模式、从机模式可选。自动确保与传送速率相对应的各种准备时间、保持时间和总线空闲时间。
- 标准模式最大 100Kbps，快速模式最大 400Kbps。
- 自动生成开始条件、重新开始条件和停止条件，并能检测到总线的开始条件，重新开始条件和停止条件。
- 可以设定 2 个从机模式地址。可同时设定 7 位地址格式和 10 位地址格式。能检测到广播呼叫地址，SMBus 主机地址，SMBus 设备默认地址，SMBus 报警地址。
- 发送时可以自动判定应答位。接收时可以自动发送应答位。
- 握手功能。
- 仲裁功能。
- SCL 输入和 SDA 输入内置数字滤波器，滤波能力可编程。
- 通信错误，接收数据满，发送数据空，一帧发送结束，地址匹配一致中断。

19.2 I2C 系统框图

19.2.1 系统框图

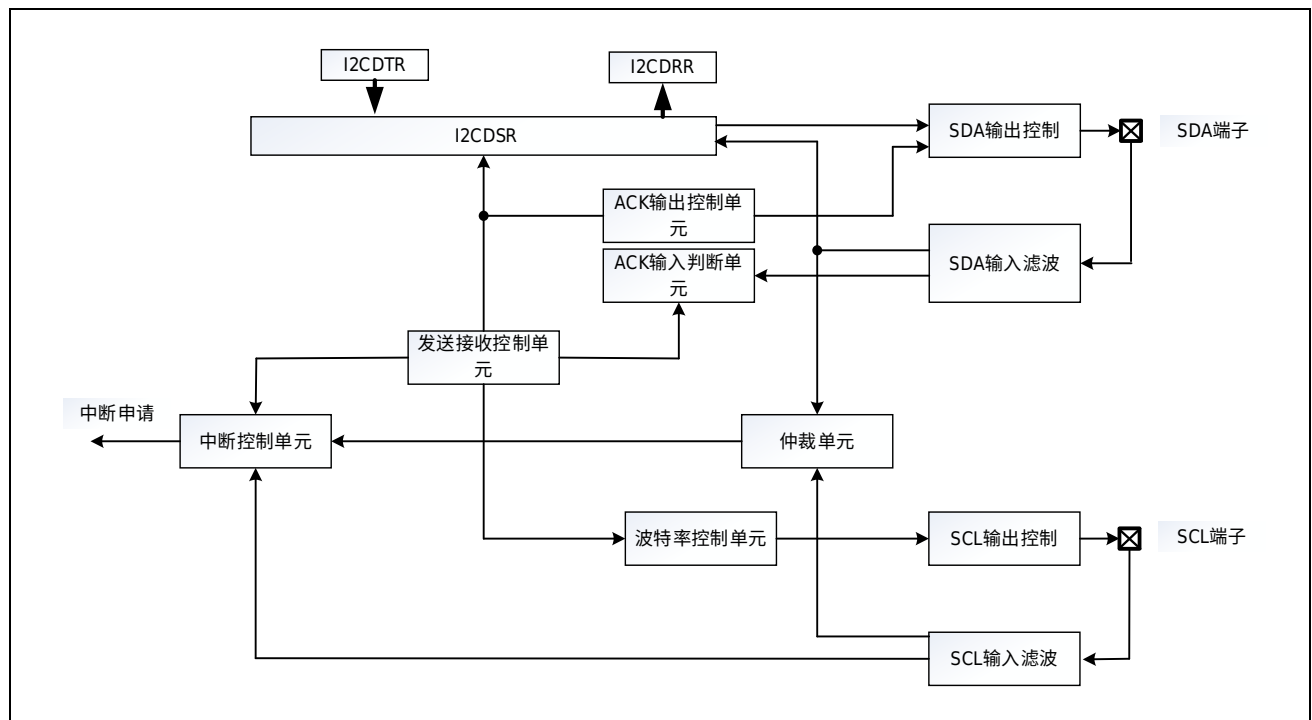


图 19-1 I2C 系统框图

19.2.2 结构图

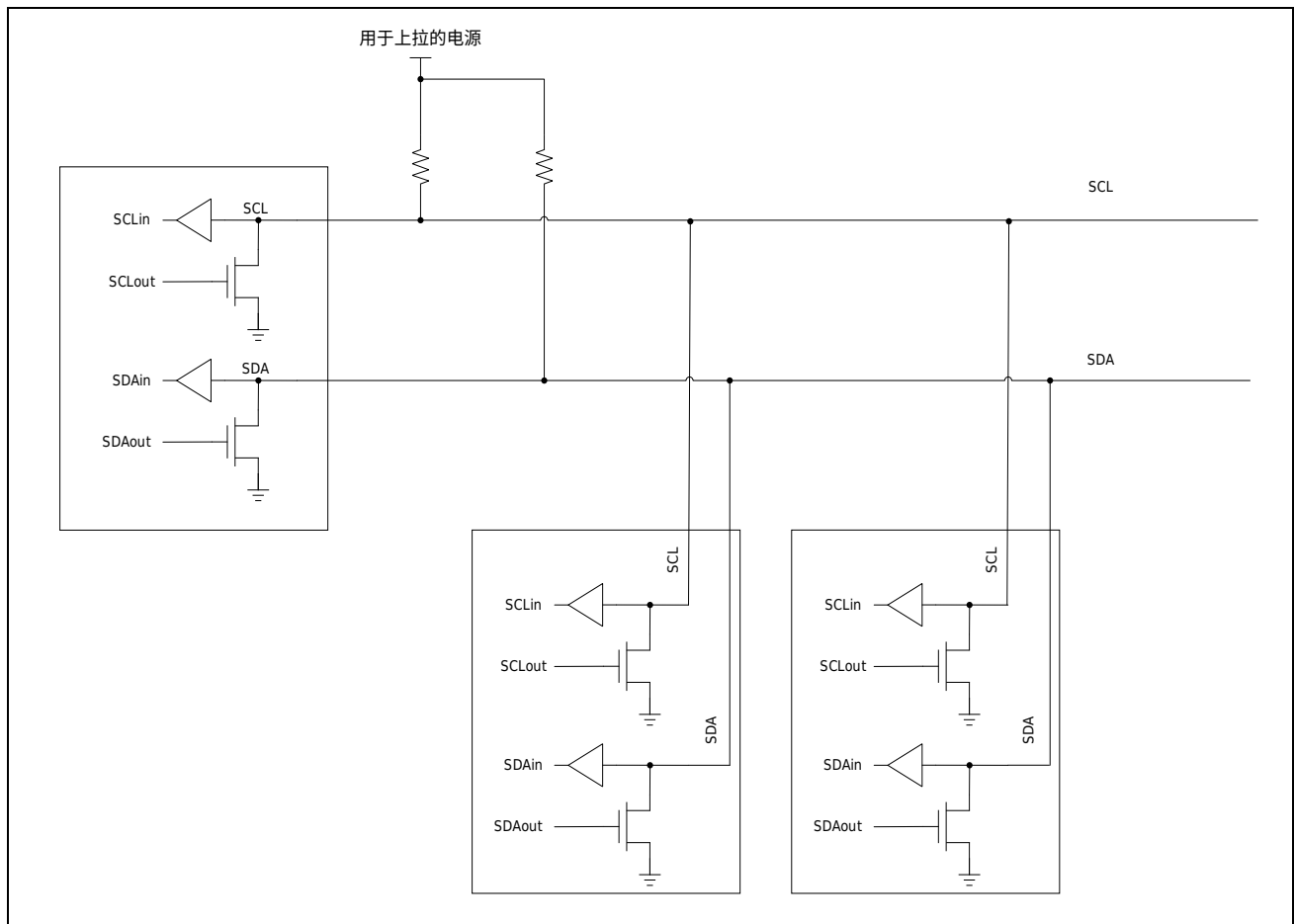


图 19-2 I2C 总线的结构例

表 19-1 输入/ 输出引脚

引脚名	输入/输出	功能
SCL	输入/输出	串行时钟的输入/输出引脚
SDA	输入/输出	串行数据的输入/输出引脚

当选择 I2C 总线时，SCL/SDA 输入电平为 Schmitt 电平（兼容 CMOS 电平）。当选择 SMBus 时，请通过设定【通用 IO (GPIO)】章节中的 PCRxy 寄存器 (PCRxy.CINSEL) 将 SCL/SDA 输入电平设定为 CMOS 电平。

19.3 动作说明

本节提供了 I2C 模块功能的描述。

19.3.1 I2C 协议

I2C 总线由一根时钟线（SCL），一根数据线（SDA）构成。所有的连接器件必须是漏极开路输出。SCL，SDA 线外接上拉电阻。电阻阻值取决于系统应用。

通常情况下，一个完整的通信过程包括下列 4 部分：

1. 开始条件
2. 地址传送
3. 数据传送
4. 停止条件

下图是 I2C 总线的时序图。

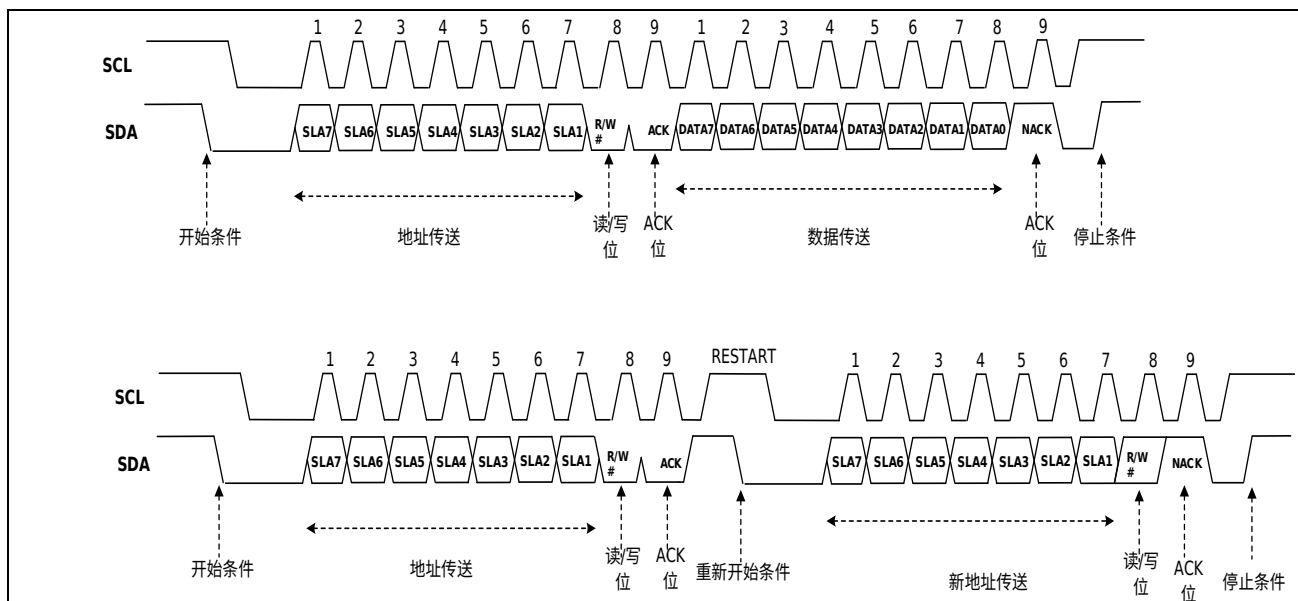


图 19-3 I2C 总线的时序图

19.3.1.1 开始条件

当总线上的主机都不驱动总线，总线进入空闲状态。SCL 和 SDA 都为高电平。总线空闲状态下总线上设备都可以通过发送开始条件启动通信。

在 I2C_SR.BUSY 标志为“0”（总线空闲）的状态下如果将 START 位置“1”，就发行开始条件。如果检测到开始条件，就自动将 I2C_SR.BUSY 标志和 I2C_SR.STARTF 标志置“1”，并且自动将 START 位置“0”。此时，如果在 START 位为“1”的状态下发送的 SDA 信号和 SDA 线的信号状态相同，并且检测到开始条件，就视为通过 START 位正确地发行了开始条件，在将 I2C_SR.MSL 位和 I2C_SR.TRA 位自动置“1”后

变为主控发送模式。另外，I2C_SR.EMPTIF 因 TRA 位为“1”而自动变为“1”。接下来把从机地址写入 I2C_DTR 寄存器，发送地址。

19.3.1.2 地址传送

开始条件或者重新开始条件后面的帧是地址帧，用于指定主机通信的对象地址。在发送停止条件之前，指定的从机一直有效。

地址帧的高 7 位为从机地址。地址帧第 8 位来决定数据帧传送的方向。

1) 7 位寻址模式见下图[7 位地址格式]

主机发送模式，主机发送地址帧第 8 位为 0

主机接收模式，主机发送地址帧第 8 位为 1

2) 10 位寻址模式见下图[10 位地址格式]

主机发送模式，主机第一帧发送头序列（11110XX0，其中 XX 表示 10 位地址的高两位），然后第二帧发送低八位从机地址。

主机接收模式，主机第一帧发送头序列（11110XX0，其中 XX 表示 10 位地址的高两位），然后第二帧发送低八位从机地址。接下来会发送一个重新开始条件，然后再发送一帧头序列（11110XX1，其中 XX 表示 10 位地址的高两位）。

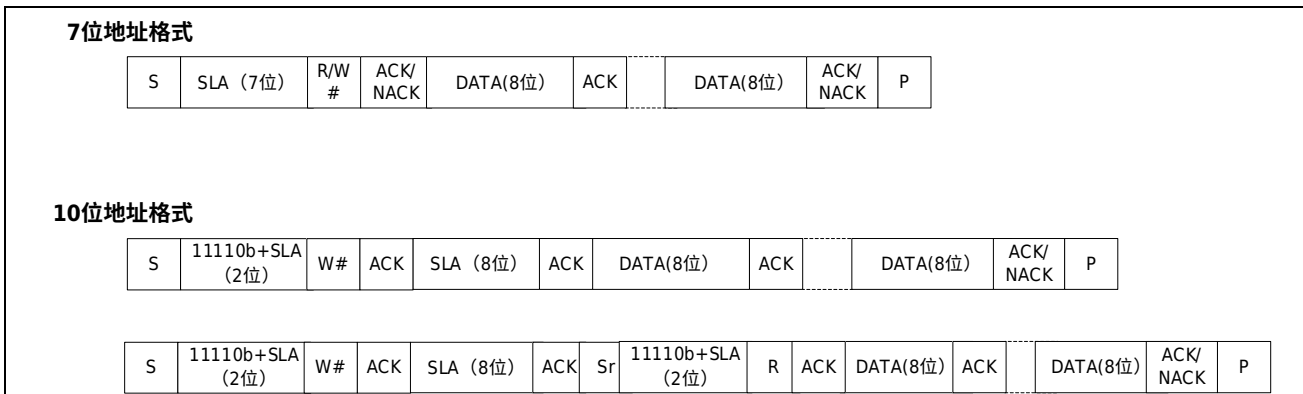


图 19-4 I2C 总线的数据格式

S : 表示开始条件。

SLA : 表示从机地址。

RW# : 表示发送和接收的方向。当 RW#为“1”时，将数据从从机发送到主机；当 RW#为“0”时，将数据从主机发送到从机。

Sr : 表示重新开始条件。

DATA : 表示发送和接收的数据。

P : 表示停止条件。

19.3.1.3 数据传送

地址匹配一致后，总线上的主机根据 RW 定义的方向一帧一帧的传送数据。

所有的地址帧后传送的数据都视为数据帧。即使是 10 位地址格式的低 8 位地址也视为数据帧。

数据帧的长度是 8 位。SCL 的低电平 SDA 变化，SCL 的高电平 SDA 保持，每个时钟周期发送一位数据。数据帧后的第 9 个时钟是应答位，是接收方向发送方传送的握手信号。

如果总线上从机接收数据，在第 9 个时钟周期不响应主机，从机必须发送 NACK。如果总线上主机接收数据，第 9 个周期发送 NACK，从机接收到 NACK，从机停止发送数据。

无论主机还是从机发送了 NACK，数据传送终止。主机可以做下列任一动作：

- 1) 发送停止条件释放总线
- 2) 发送重新开始条件开始一个新的通信

主机发送数据

在主机发送模式中，主机输出 SCL 时钟和发送数据，从机接收数据并返回应答。主机发送数据运行时序例如下图所示。

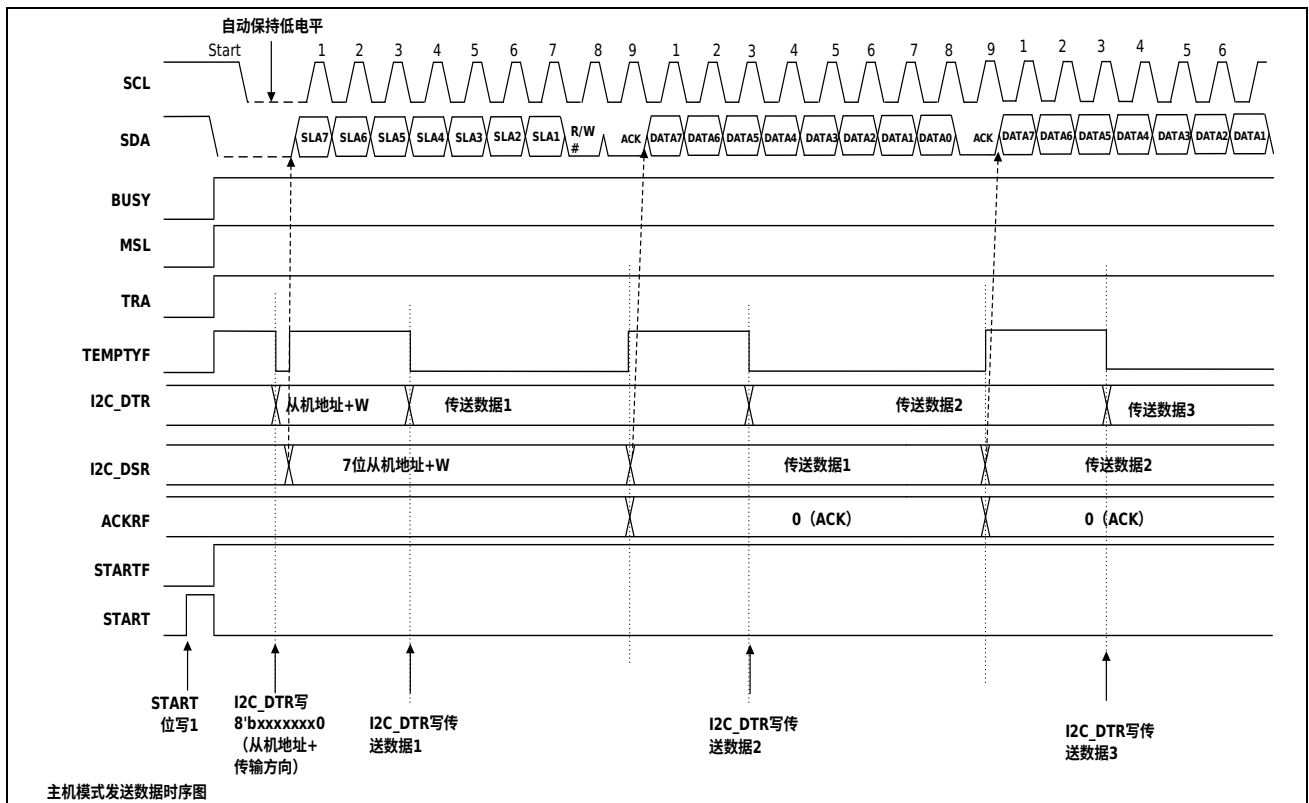


图 19-5 7 位地址格式的主机发送数据时序图（例）

主机接收数据

在主机接收模式中，主机输出 SCL 时钟，接收从机数据并返回应答。主机接收数据的运行时序例如下图所示。

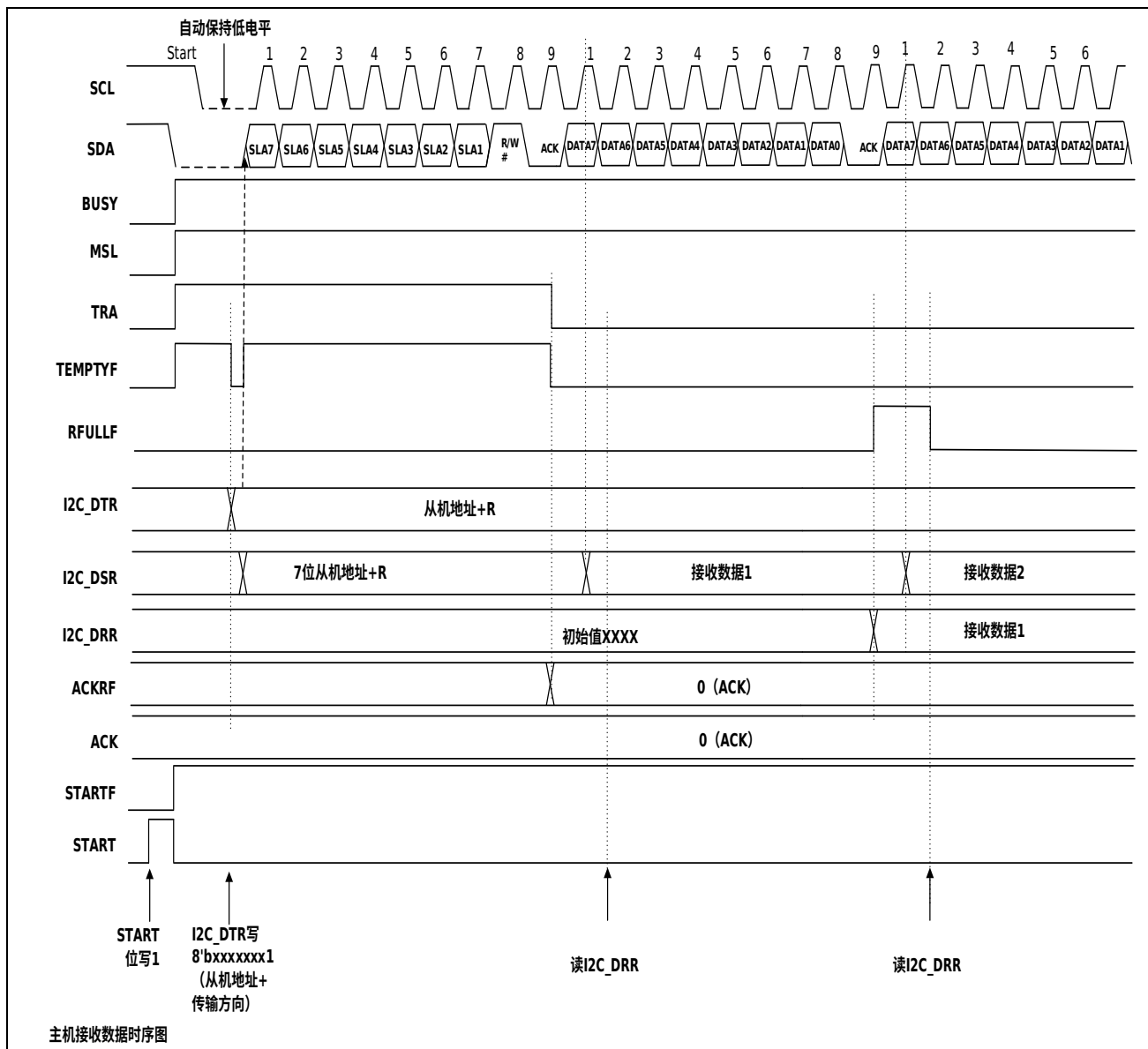


图 19-6 7 位地址格式的主机接收数据的时序图（例）

从机发送数据

在从机发送模式中，接收来自主机的 SCL 时钟，本产品为从机发送数据，并且接收主机返回应答。从机发送数据的运行时序例如下图所示。

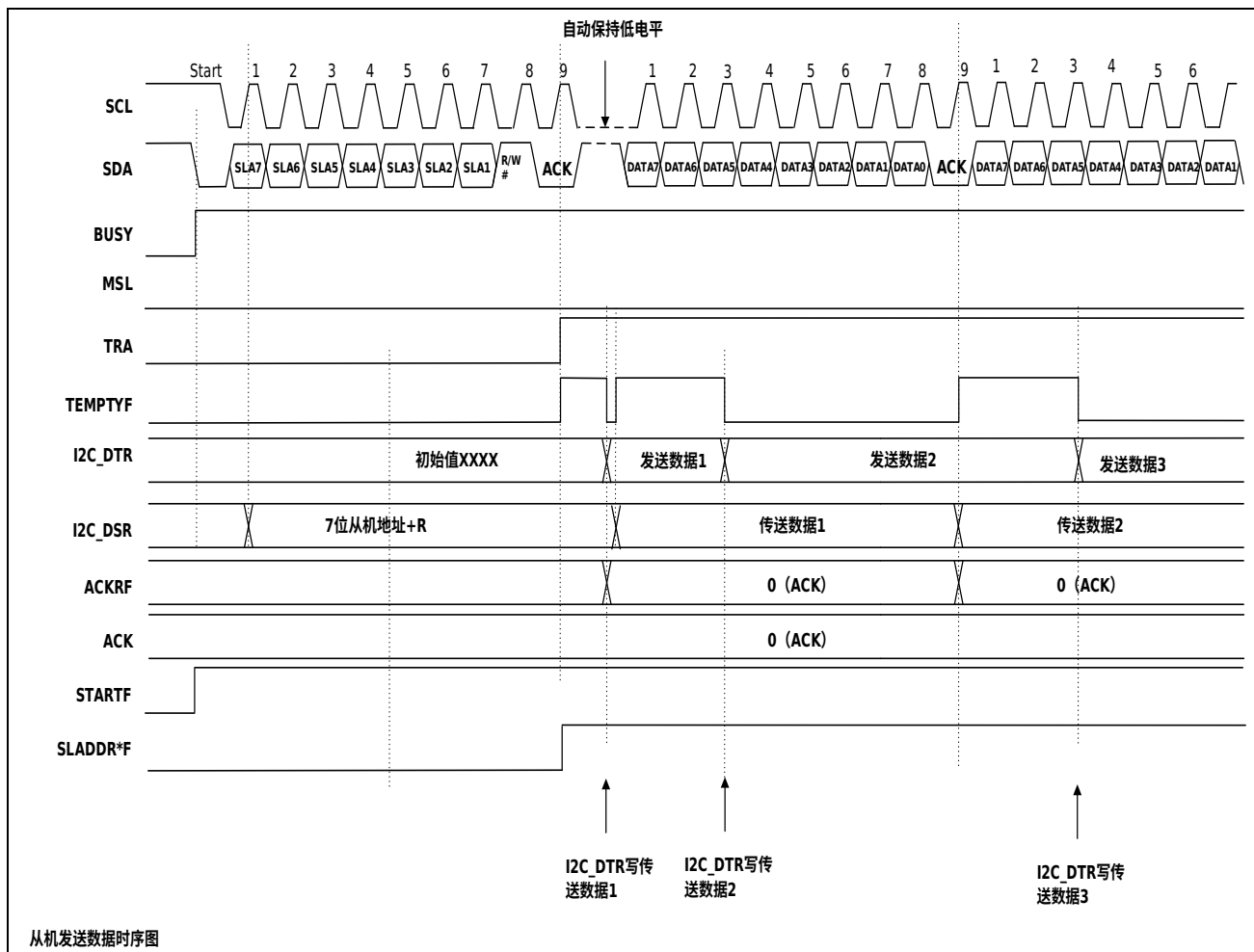


图 19-7 7 位地址格式的从机发送模式时序图（例）

从机接收数据

在从机接收模式中，接收来自主机的 SCL 时钟和数据，接收完数据后返回应答。从机接收数据的运行时序例如下图所示。

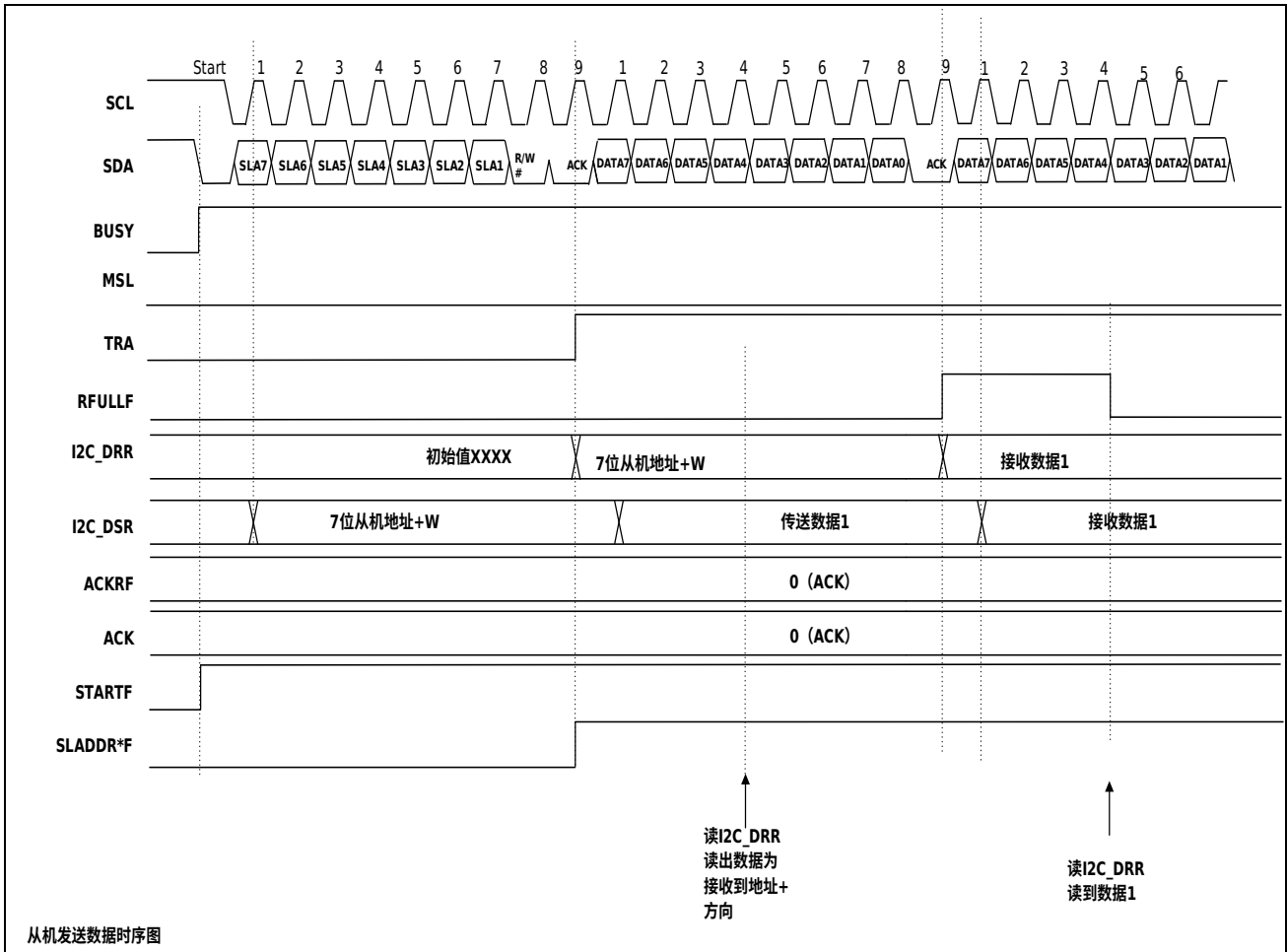


图 19-8 7 位地址格式从机接收模式时序图（例）

19.3.1.4 停止条件

通过 I2C_CR1.STOP 位发行停止条件。

在 I2C_SR.BUSY 标志为“1”（总线忙）并且 I2C_SR.MSL 位为“1”（主机模式）的状态下将 STOP 位置“1”，就发行停止条件。

19.3.1.5 重新开始条件

通过 I2C_CR1.RESTART 位生成重新开始条件。

在 I2C_SR.BUSY 标志为“1”（总线忙）并且 I2C_SR.MSL 位为“1”（主机模式）的状态下 RESTART 位置“1”，就生成重新开始条件。

通过重新开始条件，主机可以在不释放 BUS 权的情况下，切换发送/接收模式。也可以在不释放 BUS 权的情况下和另一个从机建立通信。

19.3.1.6 SCL 时钟同步

在多主机模式中使用 I2C 总线时，有可能因和其他主机的竞争而使 SCL 时钟发生冲突。如果 SCL 时钟发生冲突，主机就需要与 SCL 时钟同步，而且需要逐位进行 SCL 时钟的同步。当检测到 SCL 线的上升沿并且在对 I2C_CCR.SHIGHW 寄存器设定的高电平进行计数的过程中，如果因其他主机的 SCL 时钟输出而使 SCL 线下降，就在检测到 SCL 线的下降沿时中止高电平宽度的递增计数，并且在 SCL 线被驱动为低电平的同时开始对 I2C_CCR.SLOWW 设定的低电平宽度进行递增计数，在结束低电平宽度的计数时结束 SCL 线的低电平驱动，并释放 SCL 线。此时，如果其他主机的 SCL 时钟的低电平宽度大于 SLOWW 设定的低电平宽度，就延长 SCL 时钟的低电平宽度。当其他主机结束低电平输出时，释放 SCL 线并且 SCL 时钟上升。因此，在发生 SCL 时钟输出冲突时，SCL 时钟的高电平宽度与短时钟同步，低电平宽度与长时钟同步。

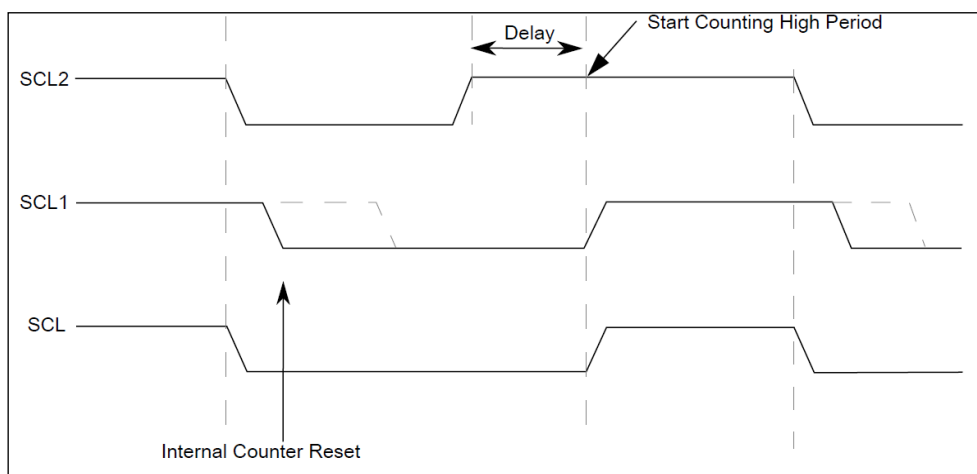


图 19-9 SCL 同步时序

19.3.1.7 仲裁

I2C 总线是一个真正的多主机总线，允许多个主机连接。

如果两个或两个以上的主机试图同时控制总线，SCL 时钟同步过程决定了总线时钟。总线时钟的低周期取决于最长的低电平时钟，高周期取决于最短的高电平时钟。高电平采集到的数据决定了仲裁结果。发送的 SDA 输出为高电平输出（SDA 引脚为高阻抗状态）而检测到 SDA 线为低电平时，就产生仲裁失败。I2C_SR.ARLOF 位会硬件置“1”。如果发生主机仲裁失败，就立即转移到从机接收模式。此时，如果包括广播地址在内的从机地址匹配，就继续从机模式的运行。

19.3.1.8 握手

数据传送过程中通过 SCL 时钟同步机制实现握手。从机在传送完一帧数据后（包含 ACK 位），将 SCL 时钟线维持在低电平。在这种情况下，SCL 时钟的低电平让主机进入等待状态，直到从机释放 SCL 线。

从机发送模式

- 1) 在发送模式中 (I2C_SR.TRA 位=1)，如果移位寄存器 (I2C_DSR 寄存器) 为空状态并且未写发送数据 (I2C_DTR 寄存器)，就在第 9 时钟和下次传送的第 1 个时钟的低电平区间自动保持 SCL 线的低电平，动作时序如下图所示。

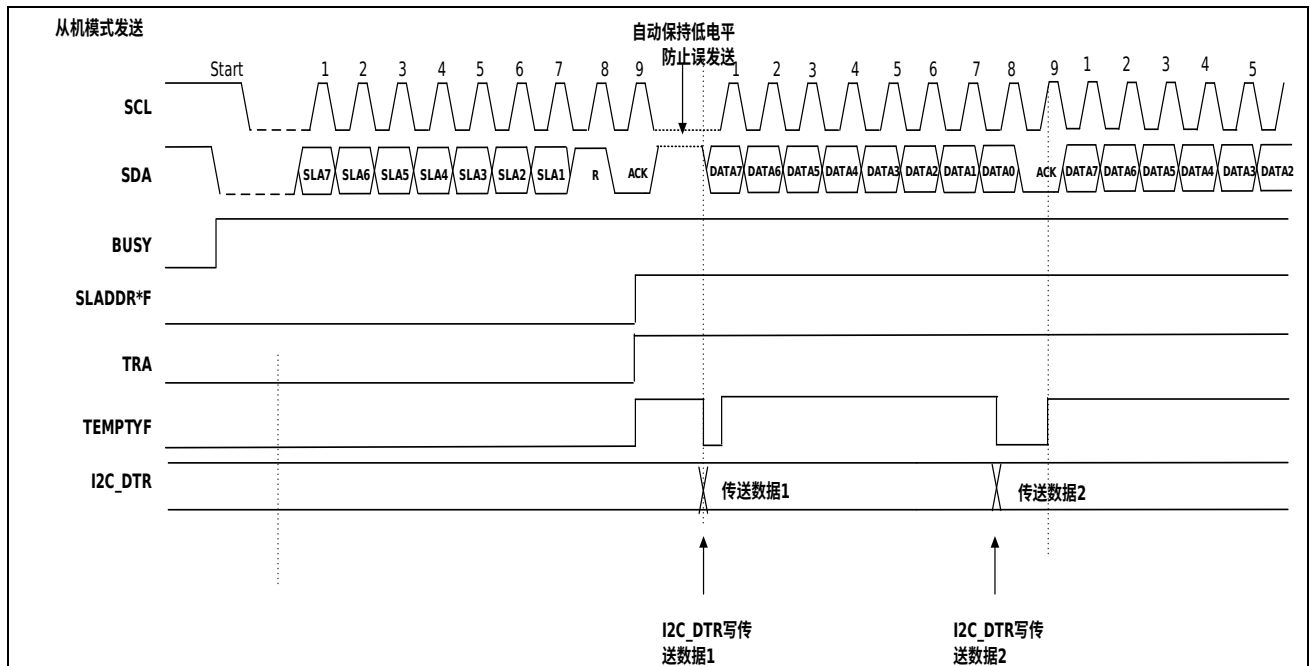


图 19-10 从机发送时序图 (1)

- 2) 在 I2C_SR.NACKF 标志变为“1”或者将最后的发送数据写到 I2C_DTR 寄存器后,在 I2C_SR.TEMPTF 标志为“1”的状态下等到 I2C_SR.TENDF 标志变为“1”。当 IC_SR.NACKF 标志或者 TENDF 标志为“1”时,在第 9 个时钟下降后将 SCL 线保持为低电平。此时必须通过读 I2C_DRR 寄存器来结束通信,从而释放 SCL 线。

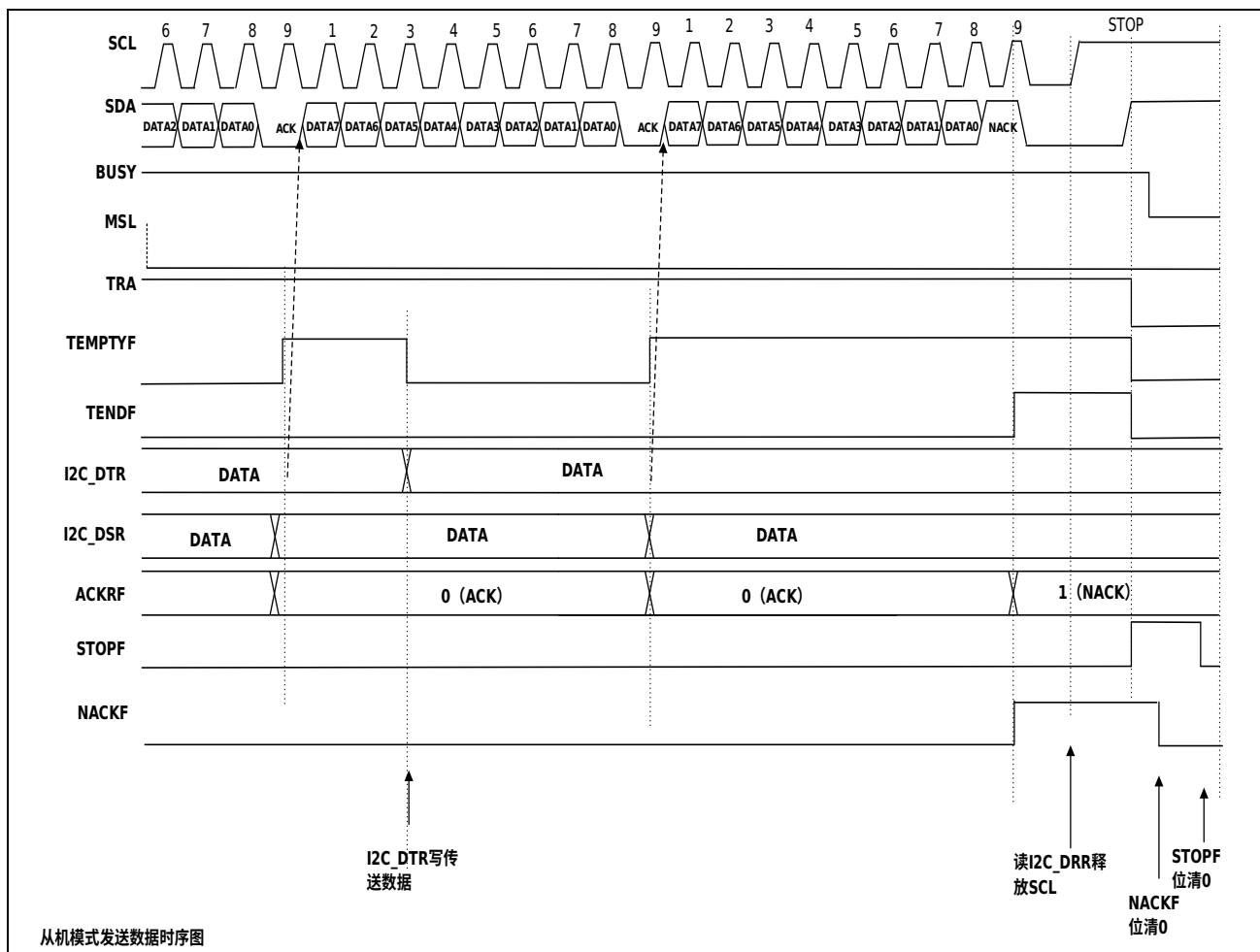


图 19-11 从机发送时序图 (2)

从机接收模式

如果在接收模式中 (I2C_SR.TRA 位=0) 并且接收数据满 (I2C_SR.RFULLF 标志=1) 的状态下, 因至少推迟 1 个传送帧读接收数据 (I2C_DRR 寄存器) 等而发生响应处理延迟, 就在开始下一个数据接收前, 在第 8 个 SCL 和第 9 个 SCL 时钟之间自动保持 SCL 线的低电平, 动作时序如下图所示。

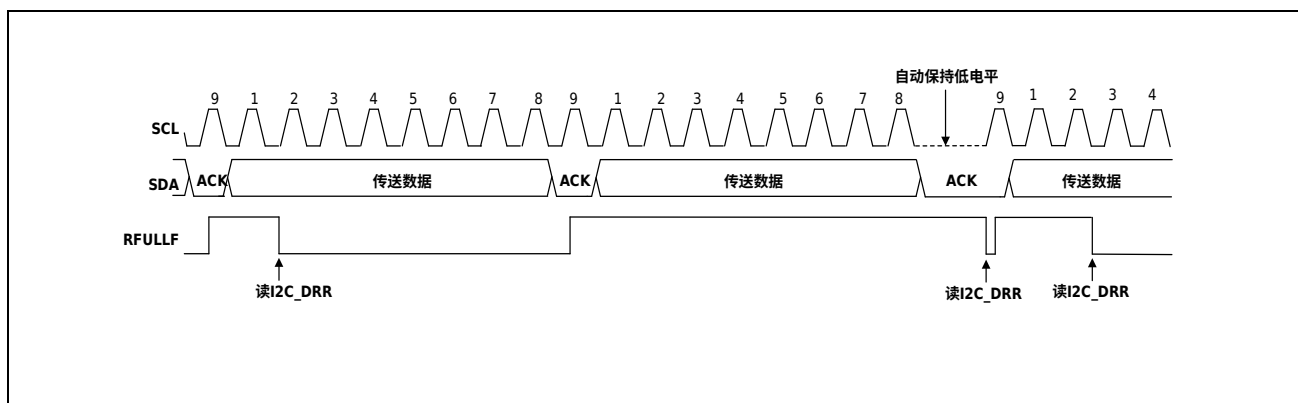


图 19-12 从机接收时序图

快速 ACK/NACK

SMBUS 通信中，利用系统内置的 CRC 运算器，计算 SMBUS 的数据包错误码（PEC）或者检查接收的数据。在检查 PEC 码的过程中，在最后字节根据是否匹配发送 ACK 或者 NACK。这就必须在接收的最后字节的 SCL 的第 8 个时钟的下降沿将 SCL 保持低电平。以此来满足软件处理时间。软件根据计算结果，写 I2C_CR1.ACK 位来解除 SCL 低电平。快速 ACK/NACK 通过 I2C_CR1.FACKEN 位控制，动作时序如下图所示。

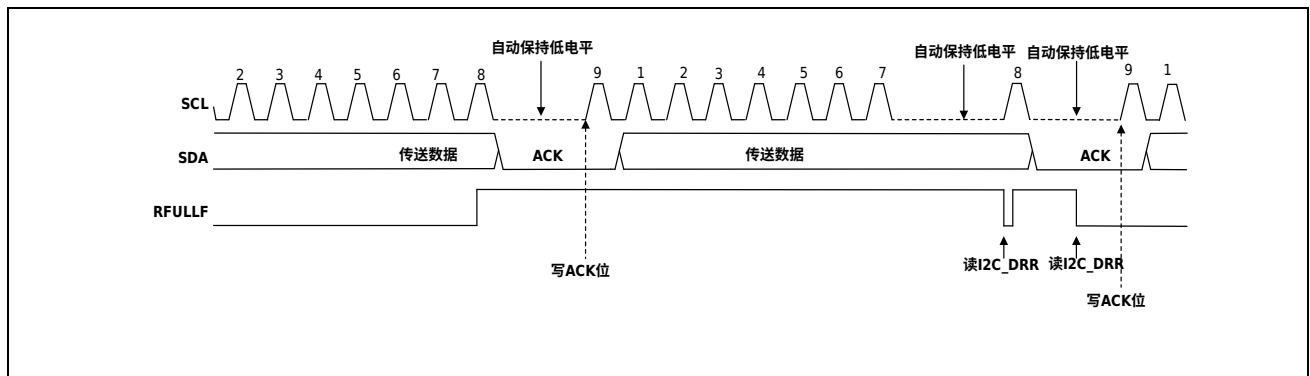


图 19-13 快速 ACK/NACK 时序图

19.3.2 地址匹配

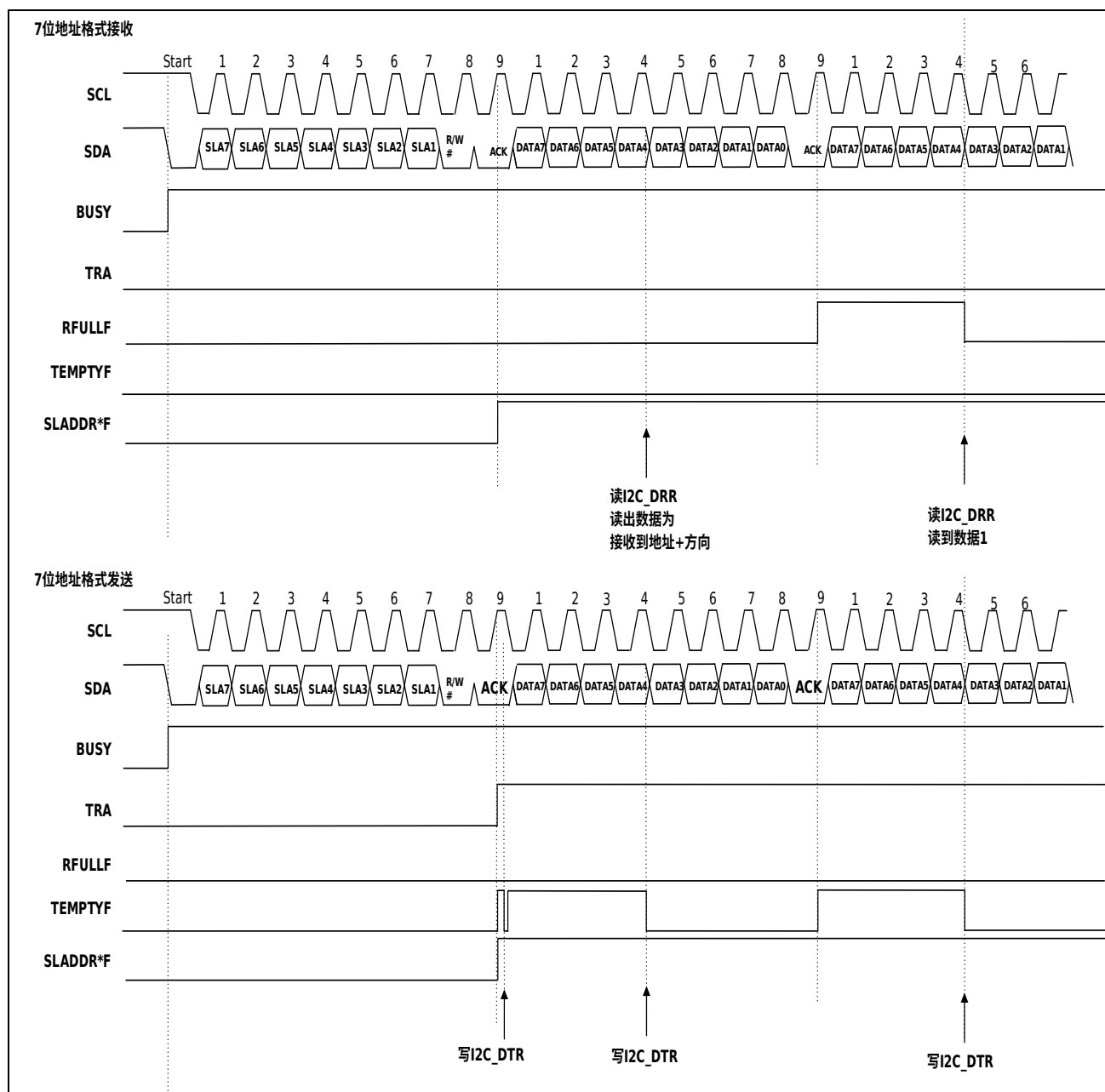
作为从机，可以设定除广播地址和主机通知地址以外的 2 种地址，从机地址能设定 7 位地址或者 10 位地址格式。

19.3.2.1 从机地址匹配

本 I2C 总线能设定 2 种从机地址，有分别对应的从机地址检测功能。当 SLADDR1EN、SLADDR0EN 为“1”时，能检测到 I2C_SLR1、I2C_SLR0 寄存器设定的从属地址。

如果设定的从机地址匹配一致，就在 SCL 时钟的第 9 个时钟的下降沿将对应的 SLADDR1F、SLADDR0F 置“1”，然后根据后续的数据位将 I2C_SR.RFULLF 标志或者 I2C_SR.EMPTIF 标志置“1”。由此，就能产生接收数据满中断或者发送数据空中断，并且能通过确认 I2C_SR.SLADDR1F、SLADDR0F 标志判断指定了哪个从机地址。

I2C_SR.SLADDR1F、SLADDR0F 标志变为“1”的时序分别如下图所示。



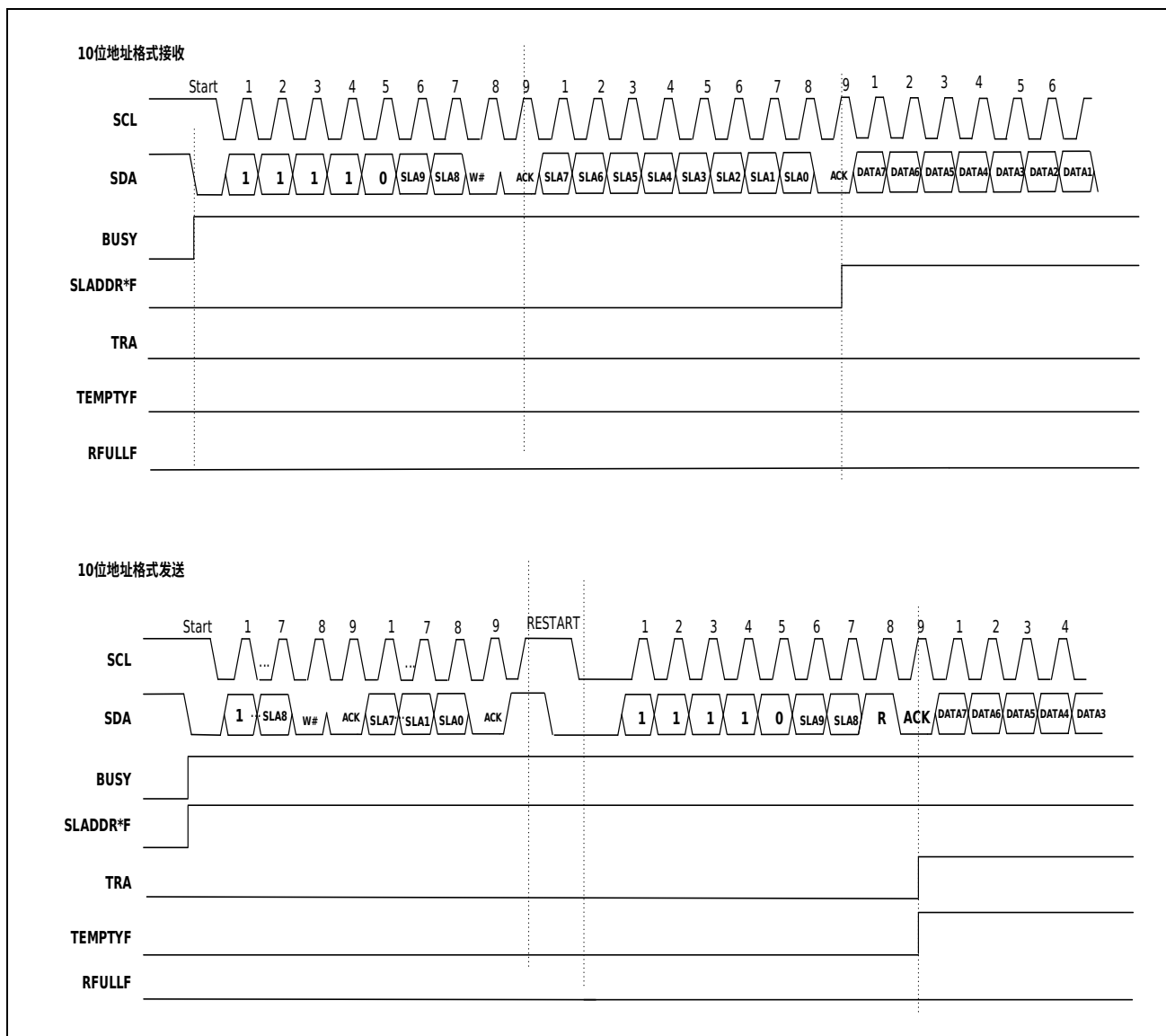


图 19-15 选择 10 位地址格式时的时序

19.3.2.2 广播地址匹配

当 I2C_CR1.GCEN 位为“1” 时，能检测广播地址（0000 000b+0[W]）。

但是开始条件或者重新开始条件后的地址为 0000 000b+1[R]（开始字节），就将此地址视为 All“0” 的从机地址而不视为广播地址。

如果匹配到广播地址，就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.GENCALLF 标志置“1”。

广播地址匹配一致后的运行和普通的从机接收运行相同。

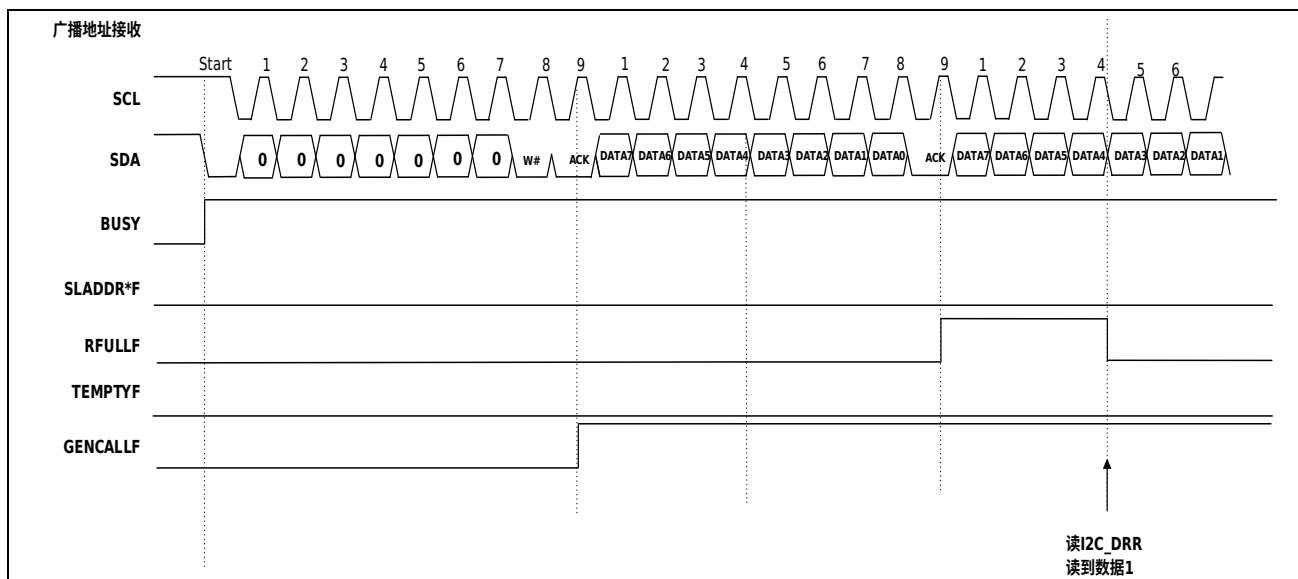


图 19-16 广播地址接收时序图

19.3.2.3 SMBus 主机地址匹配

本产品具有 SMBus 运行时的主机地址检测功能。如果在 I2C_CR1.SMBUS 位为“1”时将 I2C_CR1.SMBHOSTEN 位置“1”，就能在从机接收模式 (I2C_CR1.MSL 位与 TRA 位为“00b”) 中检测主机地址 (0001 000b)。

如果检测到 SMBUS 主机地址,就在 SCL 时钟的第 9 个时钟的下降沿将 I2C_SR.SMBHOSTF 标志置“1”。

即使接在 SMBUS 主机地址 (0001 000b) 后面的位是 Rd 位 (RW# 位接收到“1”),也能检测 SMBUS 主机地址。SMBUS 主机地址检测后的运行和普通的从机模式运行相同。

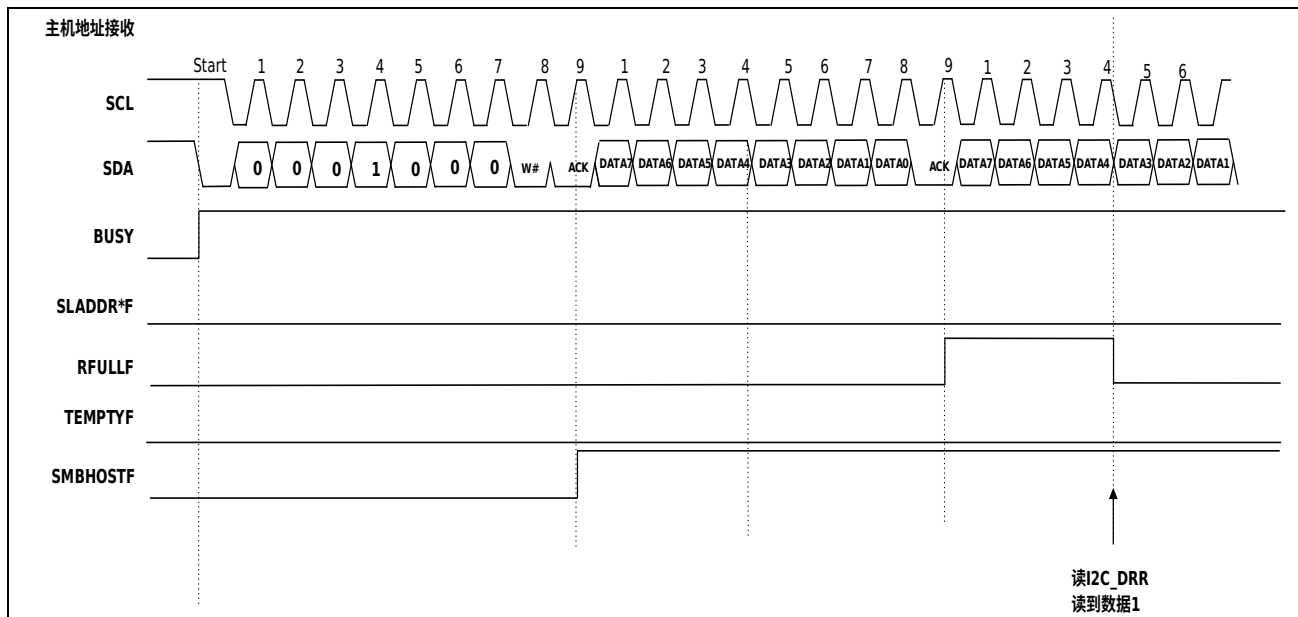


图 19-17 SMBus 主机地址接收时序图

在主机进行超时测量时，使用开始条件检测中断、停止条件检测中断以及发送结束中断或者接收数据满中断，通过芯片定时器测量各区间的时间。此超时的测量时间必须在 SMBus 规格的时钟低电平的累积时间[主机]TLOW:MEXT: 10ms (max) 以内，开始条件到停止条件的全部 TLOW:MEXT 的累加结果必须在 TLOW:SEXT: 25ms (max) 以内。

如果定时器测量的时间超过 SMBus 规格的时钟低电平的累积时间[主控设备]TLOW:MEXT: 10ms (max)，或者各测量时间的累加结果超过 SMBus 规格的时钟低电平检测的超时 TTIMEOUT: 25ms (min)，主机就需要中止处理。在主机发送时，必须立即中止发送（写 I2C_DTR 寄存器）。通过发行停止条件中止主机的处理。

19.3.3.2 数据包错误码 (PEC)

通信中，利用 CPU 运算 CRC，发送 SMBus 的数据包错误码 (PEC) 或者检查接收数据。

19.3.4 复位

具有对通信模块进行复位的功能。有 2 种复位，1 种是对包括 I2C_SR.BUSY 标志在内的全部寄存器进行初始化的复位，另一种是在保持各种设定值的状态下解除从属地址匹配状态以及对内部计数器进行初始化等的内部复位。

复位后，必须将 I2C_CR1.SWRST 位置“0”。

因为无论进行哪种复位都要解除 SCL 引脚/SDA 引脚的输出状态而变为高阻抗状态，所以也能用于解除总线意外停机状态。

从属模式中的复位会引起与主控设备的不同步，因此尽量避免使用。必须注意：在复位 (I2C_CR1.PE 位和 I2C_CR1.SWRST 位为“01b”) 过程中不能监视开始条件等的总线状态。

19.3.5 中断和事件信号输出

I2C 具有 4 种中断和用于触发启动其他外围电路的事件输出供用户选择。包括：通信错误的发生（仲裁失败检测、NACK 检测、开始条件检测、停止条件检测）、接收结束、发送数据空、发送结束。

中断一览表如下表所示。

表 19-2 中断一览表

名称	中断源	中断标志	中断条件
I2C_EEI	通信错误/通信事件	ARLOF	ARLOF=1& ARLOIE=1
		SLADDR0F	SLADDR0F=1& SLADDR0IE=1
		SLADDR1F	SLADDR1F=1& SLADDR1IE=1
		SMBALRTF	SMBALRTF =1& SMBALRTIE=1
		SMBHOSTF	SMBHOSTF =1& SMBHOSTIE=1
		SMBDEFAULTF	SMBDEFAULTF =1& SMBDEFAULTIE=1

名称	中断源	中断标志	中断条件
		GENCALLF	GENCALLF =1& GENCALLIE=1
		NACKF	NACKF=1&NACKIE=1
		STARTF	STARTF=1&STARTIE=1
		STOPF	STOPF=1&STOPIE=1
I2C_RXI	接收数据满	RFULLF	RFULLF=1&RFULLIE=1
I2C_TXI	发送数据空	EMPTYF	EMPTYF=1&EMPTYIE=1
I2C_TEI	发送结束	TENDF	TENDF=1&TENDIE=1

事件信号输出一览表如下表所示。

表 19-3 事件信号输出一览表

名称	事件源	事件条件
I2C_EEE	通信错误/通信时间	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		STARTF=1
		STOPF=1
I2C_RXE	接收数据满	RFULLF=1
I2C_TXE	发送数据空	EMPTYF=1
I2C_TEE	发送结束	TENDF=1

19.3.6 可编程数字滤波

SCL 引脚和 SDA 引脚的状态经由数字滤波器进入内部。数字滤波器电路的框图下图所示。

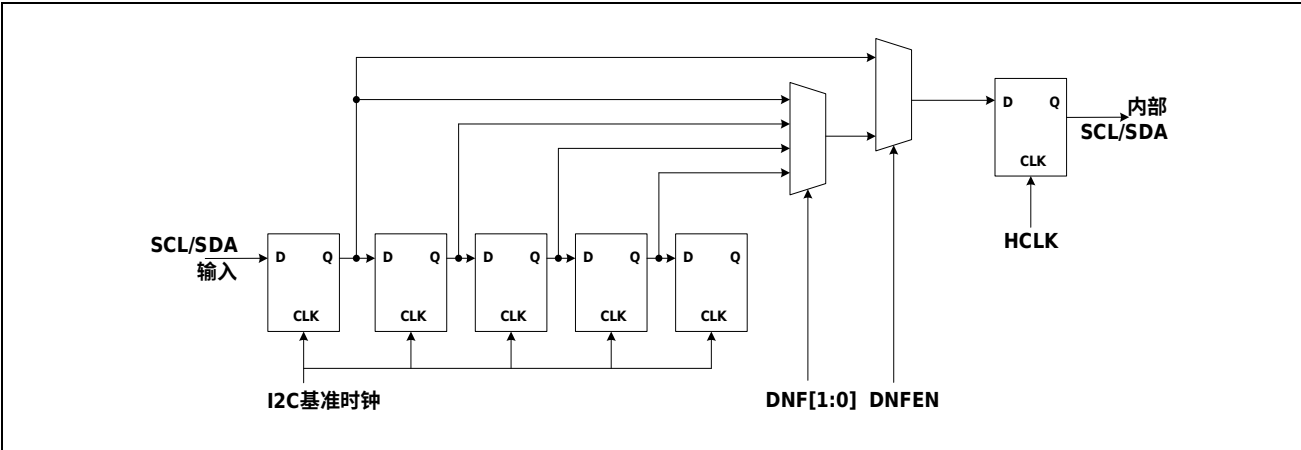


图 19-20 数字滤波电路框图

内部数字滤波器电路由 4 段串联的触发电路和匹配检测电路构成。

通过 I2C_FLTR.DNF 位选择数字滤波器的有效段数，根据选择的有效段数，噪声消除能力为 1~4 个 I2C 周期。

在 I2C 内部时钟的下降沿对 SCL 引脚的输入信号（或者 SDA 引脚的输入信号）进行采样，如果 I2C_FLTR.DNF 位设定的有效段数的触发电路输出全部匹配，就将该电平作为内部信号进行传输，否则就保持原来的值。

19.4 应用软件设定 I2C 初始化流程

在开始发送或者接收数据时，必须按下图所示的步骤进行初始化。

1. PE 位设定为 0。
2. SWRST 设定为 1，通信复位
3. PE 位设定为 1，内部状态复位
4. 设定从机地址格式和地址
5. 设定波特率
6. 根据需要设定控制寄存器功能及中断
7. SWRST 位设定为 0，解除内部状态复位。
8. 初始化结束。可发送接收数据。

19.5 寄存器描述

表 19-4 I2C 基地址

名称	基地址	描述
I2C	0x4000 4800	I2C 基地址

表 19-5 I2C 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
I2C_CR1	I2C 控制寄存器 1	0x00	32	0x0000 0040
I2C_CR2	I2C 控制寄存器 2	0x04	32	0x0000 0000
I2C_CR4	I2C 控制寄存器 4	0x0C	32	0x0000 0707
I2C_SLR0	I2C 从机地址寄存器 0	0x10	32	0x0000 1000
I2C_SLR1	I2C 从机地址寄存器 1	0x14	32	0x0000 0000
I2C_SR	I2C 状态寄存器	0x1C	32	0x0000 0000
I2C_CLR	I2C 状态清零寄存器	0x20	32	0x0000 0000

I2C_DTR	I2C 数据发送寄存器	0x24	8	0xFF
I2C_DRR	I2C 数据接收寄存器	0x28	8	0x00
I2C_CCR	I2C 波特率控制寄存器	0x2C	32	0x0000 1F1F
I2C_FLTR	I2C 滤波控制寄存器	0x30	32	0x0000 0010

19.5.1 I2C 控制寄存器 1(I2C_CR1)

复位值: 0x0000 0040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRST	Res				ACK	STOP	START	RESTART	GCEN	ACKEN	SMBHOSTEN	SMBDEFAULTEN	SMBALRTEN	SMBUS	PE

位	标记	位名	功能	读写									
b31-16	Res	保留位	读出时为“0”,写入时写“0”	RW									
b15	SWRST	软件复位	0：解除复位 1：软件复位 本位与 PE 位组合，选择内部状态复位或者通信复位 <table><tr><td>SWRST</td><td>PE</td><td>复位内容</td></tr><tr><td>1</td><td>0</td><td>通信复位：I2C 内部的全部寄存器及内部状态复位。</td></tr><tr><td>1</td><td>1</td><td>内部状态复位：I2C_SR，I2C_DSR 寄存器和内部状态机进行复位</td></tr></table>	SWRST	PE	复位内容	1	0	通信复位：I2C 内部的全部寄存器及内部状态复位。	1	1	内部状态复位：I2C_SR，I2C_DSR 寄存器和内部状态机进行复位	RW
SWRST	PE	复位内容											
1	0	通信复位：I2C 内部的全部寄存器及内部状态复位。											
1	1	内部状态复位：I2C_SR，I2C_DSR 寄存器和内部状态机进行复位											
b14-b11	Res	保留位	读出时为“0”,写入时写“0”	RW									
b10	ACK	发送应答	0：应答位发送“0” （发送 ACK） 1：应答位发送“1” （发送 NACK）	RW									
b9	STOP	停止条件生成位	0：不生成停止条件 1：生成停止条件 此位可软件置 1 和清 0。 硬件清 0 条件： 检测到停止条件 仲裁失败 检测到开始条件 通信复位	RW									
b8	START	起始条件生成位	0：不生成起始条件 1：生成起始条件 此位可软件置 1 和清 0。 硬件清 0 条件： 检测到开始条件 仲裁失败时 通信复位	RW									
b7	RESTART	重复起始条件生成位	0：不生成重复起始条件 1：生成重复起始条件 此位可软件置 1 和清 0。 硬件清 0 条件： 1) 检测到开始条件 2) 仲裁失败时 3) 通信复位	RW									
b6	GCEN	广播呼叫时能	0：广播地址检测无效	RW									

位	标记	位名	功能	读写
1: 广播地址检测有效				
b5	FACKEN	RFULLF 标志位置位时 间点选择	0: 在 SCL 时钟的第 9 个时钟上升时, 此位为“1”。(在第 8 个时 钟的下降沿, SCL 线不保持低电平) 1: 在 SCL 时钟的第 8 个时钟上升时, 此位为“1”。(在第 8 个时 钟的下降沿, SCL 线保持低电平) 通过写 ACK 位来解除保持的低电平	RW
b4	SMBHOSTEN	允许匹配 SMBUS 主机 地址位	0: 禁止匹配 SMBUS 主机地址 1: 允许匹配 SMBUS 主机地址	RW
b3	SMBDEFAULTEN	允许匹配 SMBUS 默认 地址位	0: 禁止匹配 SMBUS 默认地址 1: 允许匹配 SMBUS 默认地址	RW
b2	SMBALRTEN	允许匹配 SMBUS 报警 响应地址位	0: 禁止 SMBUS 报警响应地址 1: 允许 SMBUS 报警响应地址	RW
b1	SMBUS	SMBUS/I2C 总线模式 选择位	0: I2C 总线模式 1: SMBUS 总线模式	RW
b0	PE	I2C 功能使能	0: I2C 功能禁止 1: I2C 功能允许 本位与 SWRST 位组合, 选择内部状态复位或者通信复位	RW

19.5.2 I2C 控制寄存器 2(I2C_CR2)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBALRTIE	SMBHOSTIE	SMBDEFAULTIE	GENCALLIE	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			NACKIE	Res		ARLOIE	Res	EMPTYIE	RFULLIE	Res	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARTIE

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	RW
b23	SMBALRTIE	SMBUS 报警响应地址匹配一致中断允许	0: SMBUS 报警响应地址匹配一致中断禁止 1: SMBUS 报警响应地址匹配一致中断允许	RW
b22	SMBHOSTIE	SMBUS 主机地址匹配一致中断允许	0: SMBUS 主机地址匹配一致中断禁止 1: SMBUS 主机地址匹配一致中断允许	RW
b21	SMBDEFAULTIE	SMBUS 默认地址匹配一致中断允许	0: SMBUS 默认地址匹配一致中断禁止 1: SMBUS 默认地址匹配一致中断允许	RW
b20	GENCALLIE	广播呼叫地址匹配一致中断允许	0: 广播呼叫地址匹配一致中断禁止 1: 广播呼叫地址匹配一致中断允许	RW
b19~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	NACKIE	NACK 中断允许	0: 接收到 NACK 中断禁止 1: 接收到 NACK 中断允许	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9	ARLOIE	仲裁失败中断允许	0: 仲裁失败中断禁止 1: 仲裁失败中断允许	RW
b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	EMPTYIE	发送数据空中断允许位	0: 发送数据空中断禁止 1: 发送数据空中断允许	RW
b6	RFULLIE	接收数据满中断允许位	0: 接收数据满中断禁止 1: 接收数据满中断允许	RW
b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4	STOPIE	停止条件中断允许	0: 总线检测到停止条件中断禁止 1: 总线检测到停止条件中断允许	RW
b3	TENDIE	发送一帧数据结束中断允许位	0: 发送一帧数据结束中断禁止 1: 发送一帧数据结束中断允许	RW
b2	SLADDR1IE	从机地址 1 匹配一致中断允许	0: 从机地址 1 匹配一致中断禁止 1: 从机地址 1 匹配一致中断允许	RW
b1	SLADDR0IE	从机地址 0 匹配一致中断允许	0: 从机地址 0 匹配一致中断禁止 1: 从机地址 0 匹配一致中断允许	RW
b0	STARTIE	开始条件/重新开始条件中断允许	0: 总线检测到开始条件中断禁止 1: 总线检测到开始条件中断允许	RW

19.5.3 I2C 控制寄存器 4(I2C_CR4)

复位值：0x0000 0707

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					BUS WAIT	Res		Res				Res			

位/位域	标记	位名	功能	读写
b31~b11	Res	保留位	读出时为 0，写入时写 0	RW
b10	BUSWAIT	总线等待位	0：I2C_DRR接收满、I2C_DSR为空时，在第9个时钟和下次传送的第1 个时钟之间不保持低电平，继续接收下一个数据。 1：I2C_DRR 接收满、I2C_DSR 为空时，在第 9 个时钟和下次传送的第 1 个时钟之间保持低电平，通过读 I2C_DRR 寄存器来解除保持的低电平。	RW
b9~b8	Res	保留位	读出时为 1，写入时写 1	RW
b7~b3	Res	保留位	读出时为 0，写入时写 0	RW
b2~b0	Res	保留位	读出时为 1，写入时写 1	RW

19.5.4 I2C 从机地址寄存器 0(I2C_SLR0)

复位值: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD0	Res		SLADDR0EN	Res		SLADDR0[9:0]									

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15	ADDRMOD0	7 位/10 位地址格式选择位	0: 选择 7 位地址格式 1: 选择 10 位地址格式	RW
b14~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	SLADDR0EN	从机地址 0 有效位	0: 从机地址寄存器 0 设定值无效 1: 从机地址寄存器 0 设定值有效	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9~b8	SLADDR0[9:8]	10 位从机地址的高位	设定从机地址。 当 ADDRMOD0 位为“0”时, 此位设定无效。 当 ADDRMOD0 位为“1”时, 此位作为 10 位从机地址的高两位。	RW
b7~b0	SLADDR0[7:0]	7 位地址/10 位地址的低位	设定从机地址。 当 ADDRMOD0 位为“0”时, SLADDR0[7:1]为 7 位从机地址。 SLADDR0[0]位无效。 当 ADDRMOD0 位为“1”时, SLADDR0[7:0]为 10 位从机地址的低 8 位地址。	RW

19.5.5 I2C 从机地址寄存器 1(I2C_SLR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDR MOD1	Res		SLADDR1EN	Res		SLADDR1[9:0]									

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15	ADDRMOD1	7 位/10 位地址格式选择位	0: 选择 7 位地址格式 1: 选择 10 位地址格式	RW
b14~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	SLADDR1EN	从机地址 1 有效位	0: 从机地址寄存器 1 设定值无效 1: 从机地址寄存器 1 设定值有效	RW
b11~b10	Res	保留位	读出时为“0”,写入时写“0”	RW
b9~b8	SLADDR1[9:8]	10 位从机地址的高位	设定从机地址。 当 ADDRMOD1 位为“0”时, 此位设定无效。 当 ADDRMOD1 位为“1”时, 此位作为 10 位从机地址的高两位。	RW
b7~b0	SLADDR1[7:0]	7 位地址/10 位地址的低位	设定从机地址。 当 ADDRMOD1 位为“0”时, SLADDR1[7:1]为 7 位从机地址。 SLADDR1[0]位无效。 当 ADDRMOD1 位为“1”时, SLADDR1[7:0]为 10 位从机地址的低 8 位地址。	RW

19.5.6 I2C 状态寄存器(I2C_SR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBALRTF	SMBHOSTF	SMBDEFAULTF	GENCALLF	Res	TRA	BUSY	MSL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	-	Res	NACKF	Res	ACKRF	ARLOF	Res	TEMPTYF	RFULF	Res	STOPF	TENDF	SLADDR1F	SLADDR0F	STARTRTF

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	读出时为“0”,写入时写“0”	R
b23	SMBALRTF	SMBUS 报警响应地址匹配一致标志位	0: 未匹配到 SMBUS 报警响应地址 1: 检测到主机地址 置“1”条件: 接收到的地址和 0001 100b 匹配一致 清“0”条件: SMBALRTFCLR 写“1” 检测到停止条件 通信复位	R
b22	SMBHOSTF	SMBUS 主机地址匹配一致标志位	0: 未匹配到 SMBUS 主机地址 1: 匹配到 SMBUS 主机地址 地址匹配一致条件如下: 置“1”条件: 接收到的地址和 0001 000b 匹配一致 清“1”条件: SMBHOSTFCLR 写“1” 检测到停止条件 通信复位	R
b21	SMBDEFAULTF	SMBUS 默认地址匹配一致标志位	0: 未匹配到 SMBUS 默认地址 1: 匹配到 SMBUS 默认地址 置“1”条件: 接收到的地址和 1100 001b 匹配一致 清“0”条件: SMBDEFAULTFCLR 写“1” 检测到停止条件 通信复位	R
b20	GENCALLF	广播呼叫地址匹配一致标志	0: 未匹配到广播呼叫地址 1: 匹配到广播呼叫地址 置“1”条件: 当接收的从机地址与广播呼叫地址 (All“0”) 匹配一致时 清“0”条件: GENCALLFCLR 写“1” 检测到停止条件 通信复位	R
b19	Res	保留位	读出时为“0”,写入时写“0”	R

位/位域	标记	位名	功能	读写															
			此位表示选择发送数据还是接收数据。 0：接收数据 1：发送数据 此位可软件置 1 和清 0。 硬件置“1”条件 检测到开始条件																
b18	TRA	发送接收选择位	主机模式下，发送的 RW 位为 0 从机模式下，地址匹配且接收的 RW 位为 1 硬件清“0”条件 检测到停止条件 主机模式下，发送的 RW 位为 1 从机模式下，地址匹配且接收的 RW 位为 0 通信复位	RW															
b17	BUSY	总线忙标志位	0：空闲状态，总线上无通信 1：占有状态，总线正在通信 置“1”条件： 检测到总线上开始条件 清“0”条件： 检测到总线停止条件 通信复位	R															
b16	MSL	主从机选择位	此位表示主机还是从机。 0：从机模式 1：主机模式 通过和 TRA 位的组合，表示 I2C 的运行模式。 <table><tr><td>MSL</td><td>TRA</td><td>I2C 运行模式</td></tr><tr><td>0</td><td>0</td><td>从机接收模式</td></tr><tr><td>0</td><td>1</td><td>从机发送模式</td></tr><tr><td>1</td><td>0</td><td>主机接收模式</td></tr><tr><td>1</td><td>1</td><td>主机发送模式</td></tr></table> 此位可软件置 1 和清 0。 硬件置“1”条件 START 位为 1 的状态下，检测到开始条件 硬件清“0”条件 1) 检测到停止条件 2) 仲裁失败 3) 通信复位	MSL	TRA	I2C 运行模式	0	0	从机接收模式	0	1	从机发送模式	1	0	主机接收模式	1	1	主机发送模式	RW
MSL	TRA	I2C 运行模式																	
0	0	从机接收模式																	
0	1	从机发送模式																	
1	0	主机接收模式																	
1	1	主机发送模式																	
b15-b13	Res	保留位	读出时为“0”,写入时写“0”	R															
b12	NACKF	NACK 标志位	0：未接收到 NACK 1：接收到 NACK 置“1”条件： 发送模式下，接收到 NACK 清“0”条件： NACKFCLR 写“1” 通信复位	R															
b11	Res	保留位	读出时为“0”,写入时写“0”	R															

位/位域	标记	位名	功能	读写
b10	ACKRF	接收应答位	0: 接收到应答位为“0” (接收 ACK) 1: 接收到应答位为“1” (接收 NACK) 置“1”条件: 发送模式下, 接收到 NACK 清“0”条件: 发送模式下, 接收到 ACK 通信复位	R
b9	ARLOF	仲裁失败标志位	0: 未发生仲裁失败 1: 仲裁失败 置“1”条件: 仲裁失败 清“0”条件: ARLOFCLR 写“1” 通信复位	R
b8	Res	保留位	读出时为“0”,写入时写“0”	R
b7	EMPTYF	发送数据空标志位	0: I2C_DTR 寄存器满 1: I2C_DTR 寄存器空 置“1”条件: I2C_DTR 数据传送到 I2C_DSR TRA 位置 1 清“0”条件: 写 I2C_DTR TRA 位清 0 通信复位	R
b6	RFULLF	接收数据满标志位	0: I2C_DRR 寄存器空 1: I2C_DRR 寄存器满 置“1”条件: 接收的数据从 I2C_DSR 传送到 I2C_DRR 清“0”条件: 读 I2C_DRR RFULLFCLR 写“1” 通信复位	R
b5	Res	保留位	读出时为“0”,写入时写“0”	R
b4	STOPF	停止条件标志位	0: 总线未检测到停止条件 1: 总线检测到停止条件 置“1”条件: 检测到停止条件 清“0”条件: STOPFCLR 写“1” 通信复位	R
b3	TENDF	发送数据结束标志位	0: I2C_DSR 寄存器发送中 1: I2C_DSR 寄存器发送结束 置“1”条件: EMPTYF=1 的条件下, SCL 的第 9 个上升沿此位置“1” 清“0”条件: 检测到停止条件	R

位/位域	标记	位名	功能	读写
			写 I2C_DTR TENDFCLR 写“1” 通信复位	
b2	SLADDR1F	从机地址寄存器 1 匹配一致标志	0: 未检测到从机地址寄存器 1 一致地址 1: 检测到从机地址寄存器 1 一致地址 置“1”条件: 当 I2C_SLR1.ADDRMODE1 位为“0”时, 接收到的从机地址和 I2C_SLR1. SLADDR1[7:1]匹配时。 当 I2C_SLR1.ADDRMODE1 位为“1”时, 接收 10 位从机地址的第一个字节地址与 11110b+I2C_SLR1. SLADDR1[9:8]匹配一致并且第二个字节地址与 I2C_SLR1. SLADDR1[7:0]匹配一致。 清“0”条件: 检测到停止条件 SLADDR1FCLR 写“1” 通信复位	R
b1	SLADDR0F	从机地址寄存器 0 匹配一致标志	0: 未检测到从机地址寄存器 0 一致地址 1: 检测到从机地址寄存器 0 一致地址 置“1”条件: 当 I2C_SLR0.ADDRMODE0 位为“0”时, 接收到的从机地址和 I2C_SLR0. SLADDR0[7:1]匹配时。 当 I2C_SLR0.ADDRMODE0 位为“1”时, 接收 10 位从机地址的第一个字节地址与 11110b+I2C_SLR0. SLADDR0[9:8]匹配一致并且第二个字节地址与 I2C_SLR0. SLADDR0[7:0]匹配一致。 清“0”条件: 检测到停止条件 SLADDR0FCLR 写“1” 通信复位	R
b0	STARTF	开始条件/重新开始条件标志位	0: 总线未检测到开始条件 1: 总线检测到开始条件 置“1”条件 1) 检测到开始条件 清“0”条件 1) 检测到停止条件 2) STARTFCLR 写“1” 3) 通信复位	R

19.5.7 I2C 状态清零寄存器(I2C_CLR)

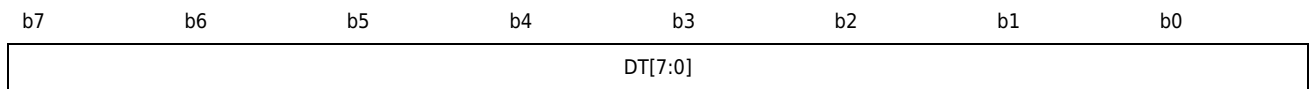
复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBAL RTFCL R	SMBH OSTFC LR	SMBDE FAULT FCLR	GENC ALLFC LR	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			NAC KFCL R	Res		ARL OFC LR	Res	TEMPT YFCLR	RFULL FCLR	Res	STOP FCLR	TEND FCLR	SLAD DR1F CLR	SLADD R0FCL R	STA RTF CLR

位/位域	标记	位名	功能	读写
b31~b24	Res	保留位	写入时写“0”	W
b23	SMBALRTFCLR	SMBUS 报警响应地址 匹配一致标志清零位	写“1”清除 SMBALRTF 标志位	W
b22	SMBHOSTFCLR	SMBUS 主机地址匹配 一致标志清零位	写“1”清除 SMBHOSTF 标志位	W
b21	SMBDEFAULTFCLR	SMBUS 默认地址匹配 一致标志清零位	写“1”清除 SMBDEFAULTF 标志位	W
b20	GENCALLFCLR	广播呼叫地址匹配一 致标志	写“1”清除 GENCALLF 标志位	W
b19~b13	Res	保留位	写入时写“0”	W
b12	NACKFCLR	NACK 标志位	写“1”清除 NACKF 标志位	W
b11~b10	Res	保留位	写入时写“0”	W
b9	ARLOFCLR	仲裁失败标志位	写“1”清除 ARLOF 标志位	W
b8	Res	保留位	写入时写“0”	W
b7	EMPTYFCLR	发送数据空标志位	写“1”清除 EMPTYF 标志位	W
b6	RFULLFCLR	接收数据满标志位	写“1”清除 RFULLF 标志位	W
b5	Res	保留位	写入时写“0”	W
b4	STOPFCLR	停止条件标志位	写“1”清除 STOPF 标志位	W
b3	TENDFCLR	发送数据结束标志位	写“1”清除 TENDF 标志位	W
b2	SLADDR1FCLR	从机地址寄存器 1 匹 配一致标志清零位	写“1”清除 SLADDR1F 标志位	W
b1	SLADDR0FCLR	从机地址寄存器 1 匹 配一致标志清零位	写“1”清除 SLADDR0F 标志位	W
b0	STARTFCLR	开始条件/重新开始条 件标志清零位	写“1”清除 STARTF 标志位	W

19.5.8 I2C 数据发送寄存器(I2C_DTR)

复位值：0xFF



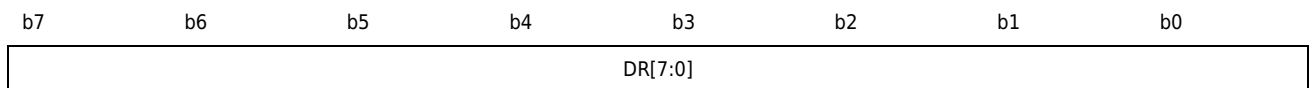
如果 I2C_DSR 寄存器为空，就将写在 I2C_DTR 寄存器的发送数据传送到 I2C_DSR 寄存器，发送模式时开始发送数据到 SDA 上。

I2C_DSR 寄存器和 I2C_DTR 寄存器是双缓冲结构，在 I2C_DSR 寄存器数据发送过程中，如果预先写 I2C_DTR 寄存器的数据，就能进行连续发送数据。

I2C_DTR 寄存器可读可写。请在发送数据空中断要求发生时，仅写一次 I2C_DTR 寄存器。

19.5.9 I2C 数据接收寄存器(I2C_DRR)

复位值：0x00



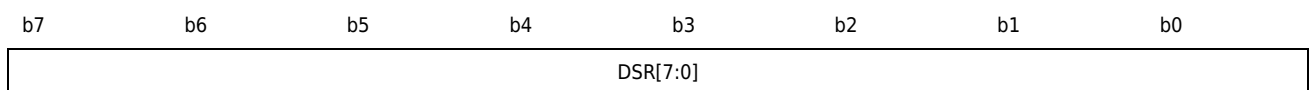
如果接收到 1 帧数据，就能将接收数据从移位寄存器（I2C_DSR）转存到 I2C_DRR 寄存器，进而可以进入到下一个数据接收状态。

I2C_DSR 寄存器和 I2C_DRR 寄存器是双缓冲结构，在 I2C_DSR 寄存器数据接收过程中，如果读取了 I2C_DRR 寄存器的数据，就能进行连续接收数据。

禁止对 I2C_DRR 寄存器写。请在接收数据满中断要求发生时，仅读一次 I2C_DRR 寄存器。

在 I2C_SR.RFULLF 标志位为“1”的状态下，如果不读取 I2C_DRR 寄存器的数据，而立即接收下一个数据，SCL 时钟就在下一次 RFULLF 标志位变“1”的前一个 SCL 时钟自动保持为低电平。

19.5.10 I2C 数据移位寄存器(I2C_DSR)



I2C_DSR 寄存器用于发送和接收数据的移位寄存器。I2C_DSR 寄存器不可读也不可写。

在数据发送时，将发送数据从 I2C_DTR 寄存器传送到 I2C_DSR 寄存器，从 SDA 引脚发送数据。在数据接收时，一旦 1 帧数据接收结束，就将数据从 I2C_DSR 寄存器传送到 I2C_DRR 寄存器。

19.5.11 I2C 时钟控制寄存器(I2C_CCR)

复位值: 0x0000 1F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									Res			CKDIV[2:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			SHIGHW[4:0]					Res			SLOWW[4:0]				

位	标记	位名	功能	读写
b31~b23	Res	保留位	读出时为“0”,写入时写“0”	RW
b22~b19	Res	保留位	读出时为“1”,写入时写“1”	RW
b18~b16	CKDIV[2:0]	I2C 基准时钟频率设定位	0 0 0: I2C 基准时钟频率=HCLK/1 0 0 1: I2C 基准时钟频率=HCLK/2 0 1 0: I2C 基准时钟频率=HCLK/4 0 1 1: I2C 基准时钟频率=HCLK/8 1 0 0: I2C 基准时钟频率=HCLK/16 1 0 1: I2C 基准时钟频率=HCLK/32 1 1 0: I2C 基准时钟频率=HCLK/64 1 1 1: I2C 基准时钟频率=HCLK/128	RW
b15~b13	Res	保留位	读出时为“1”,写入时写“1”	RW
b12~b8	SHIGHW[4:0]	设定 SCL 高电平宽度位	设定 SCL 时钟的高电平宽度	RW
b7~b5	Res	保留位	读出时为“1”,写入时写“1”	RW
b4~b0	SLOWW[4:0]	设定 SCL 低电平宽度位	设定 SCL 时钟的低电平宽度	RW

SHIGHW 位 (设定 SCL 高电平宽度位)

在主机模式下, SHIGHW 是用于设定 SCL 时钟的高电平宽度。在从机模式下, 设定无效。

SLOWW 位 (设定 SCL 低电平宽度位)

SLOWW 是用于设定 SCL 时钟的低电平宽度。在从机模式下, 设定值要大于数据准备时间。数据准备时间 (tSU:DAT) 250ns (~100kbps: 标准模式) 100ns (~400kbps: 快速模式)

波特率:

DNFE=0,CKSDIV=000

波特率=1/{[(SHIGHW+3)+(SLOWW+3)]/ΦI2C+SCL 上升时间+SCL 下降时间}

SDNFE=1,CKSDIV=000

波特率=1/{[(SHIGHW+3+滤波能力)+(SLOWW+3+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=0,CKSDIV!=000

波特率=1/{[(SHIGHW+2)+(SLOWW+2)]/ΦI2C+SCL 上升时间+SCL 下降时间}

DNFE=1,CKSDIV!=000

波特率=1/{[(SHIGHW+2+滤波能力)+(SLOWW+2+滤波能力)]/ΦI2C+SCL 上升时间+SCL 下降时间}

注:

- SCL 线的上升时间[tr] 和下降时间[tf] 取决于总线的总电容[Cb] 和上拉电阻[Rp], 详细内容请参考 NXP 公司的 I2C 总线规格书。

19.5.12 I2C 滤波控制寄存器(I2C_FLTR)

复位值: 0x0000 0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											DNFE N	Res		DNF[1:0]	

位/位域	标记	位名	功能	读写
b31~b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4	DNFEN	数字滤波功能允许位	0: 数字滤波功能禁止 1: 数字滤波功能允许	RW
b3~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1-b0	DNF[1:0]	数字滤波器滤波能力选择	00: 滤波能力 1 个 I2C 基准时钟周期 01: 滤波能力 2 个 I2C 基准时钟周期 10: 滤波能力 3 个 I2C 基准时钟周期 11: 滤波能力 4 个 I2C 基准时钟周期	RW

20 串行外设接口（SPI）

20.1 概述

本产品搭载 1 个通道的串行外设接口 SPI，支持高速全双工串行同步传输，方便地与外围设备进行数据交换。用户可根据需要进行三线/四线，主机/从机及波特率范围的设置。

SPI 主要特性：

表 20-1 SPI 的特性要点

要点	描述
通道数	1 通道
串行通信功能	支持 4 线式 SPI 模式和 3 线式时钟同步运行模式 支持全双工和只传送两种通信方式 可调整通信时钟 SCK 的极性和相位
数据格式	可选择数据移位顺序:MSB 开始/LSB 开始 可选择数据宽度: 8/16 位 单次最多可传送或接收 1 帧宽度为 16 位的数据
波特率	主机模式下可通过内置专用波特率发生器对波特率进行调整，波特率范围为 HCLK 的 2 分频~256 分频 从机模式下允许的最大波特率为 HCLK 的 6 分频
错误监测	模式故障错误监测 数据过载错误监测 数据欠载错误监测 奇偶校验错误监测
片选信号控制	每个通道配置一根片选信号线 极性可调
主机模式下的传输控制	通过将数据写入数据寄存器启动传输
中断/AOS 源	接收数据区域已满 发送数据区域已空 SPI 错误（模式/过载/欠载/奇偶校验） SPI 闲置 传输完成
低功耗控制	可设置模块停止
其他功能	SPI 初始化功能

注：

- 在主接收模式使用通信自动挂起功能时，由于通信时钟停止，将不会发生过载错误。详情请参考【20.8.3 过载错误】。

20.2 SPI 系统框图

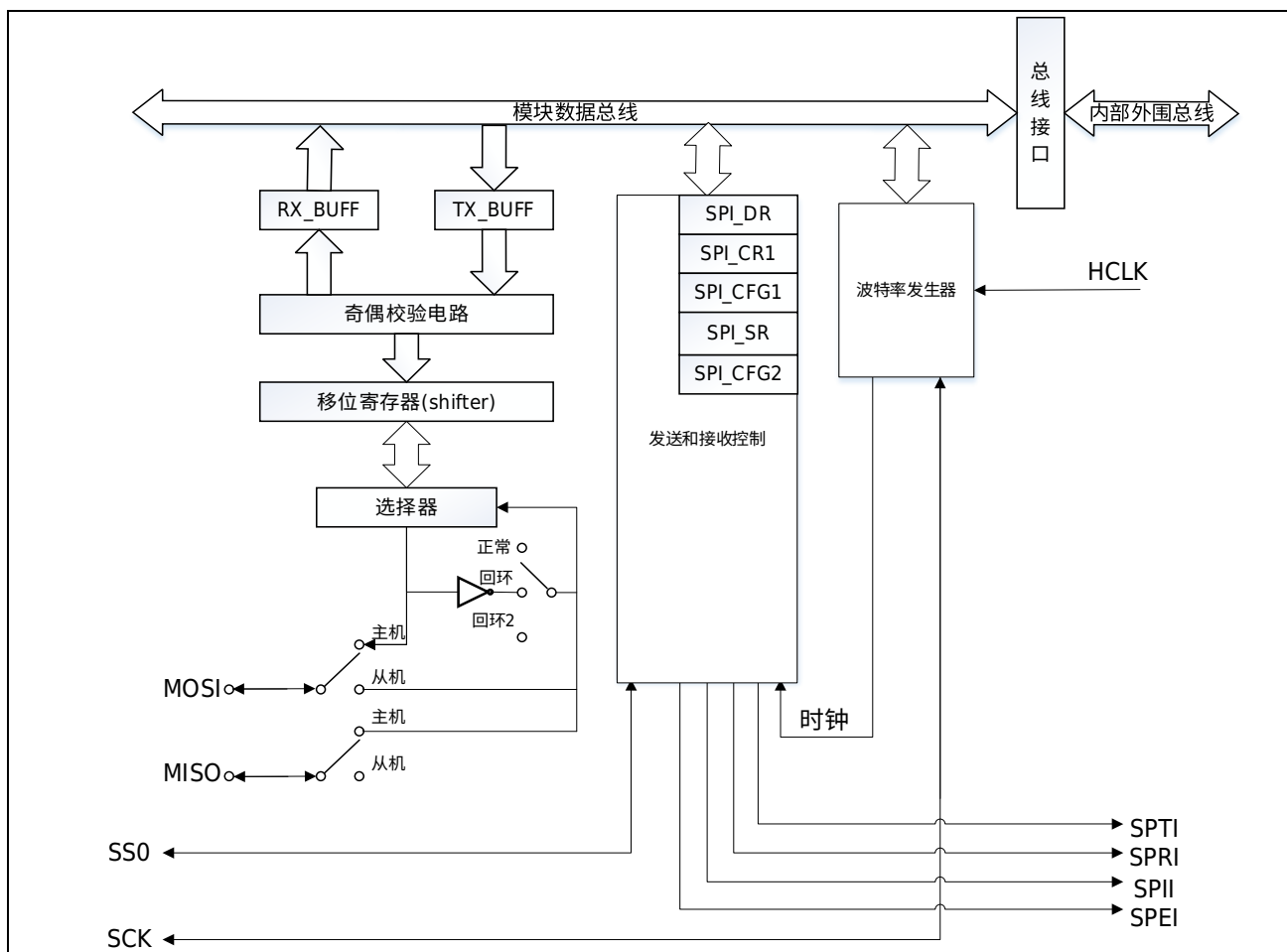


图 20-1 系统框图

20.3 管脚说明

表 20-2 管脚说明

管脚名	端口方向	功能
SCK	输入/输出	通信时钟管脚
MOSI	输入/输出	主机数据传输管脚
MISO	输入/输出	从机数据传输管脚
SS0	输入/输出	从机选择输入/输出管脚

20.4 SPI 动作系统说明

20.4.1 主机模式的管脚状态

SPI 工作在主机模式下时，各个管脚的状态如下表 20-3 所示。

表 20-3 主机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态(PFS.ODS=0)	管脚状态(PFS.ODS=1)
SPI 动作 (SPIMDS=0)	主机模式 (MSTR=1 、 MODFE=0)	SCK	CMOS 输出	OD 输出
		SS0	CMOS 输出	OD 输出
		MOSI	CMOS 输出	OD 输出
		MISO	输入	输入
时钟同步运行 (SPIMDS=1)	主机模式 (MSTR=1)	SCK	CMOS 输出	OD 输出
		SS0 (不使用)	Hi-Z (可作为通用 I/O)	Hi-Z (可作为通用 I/O)
		MOSI	CMOS 输出	OD 输出
		MISO	输入	输入

注：

- 当 SS0 输入有效电平时，SPI 让出总线控制权，管脚状态为 Hi-Z。

20.4.2 从机模式的管脚状态

SPI 工作在从机模式下时，各个管脚的状态如下表 20-4 所示。

表 20-4 从机模式时 SPI 管脚状态说明

模式		管脚名	管脚状态(PFS.ODS=0)	管脚状态(PFS.ODS=1)
SPI 动作 (SPIMDS=0)	从机模式 (MSTR=0 、 MODFE=0)	SCK	输入	输入
		SS0	输入	输入
		MOSI	输入	输入
		MISO (注 2)	CMOS 输出/Hi-Z	OD 输出/Hi-Z
时钟同步运行 (SPIMDS=1)	从机模式 (MSTR=0)	SCK	输入	输入
		SS0 (不使用)	Hi-Z (可作为通用 I/O)	Hi-Z (可作为通用 I/O)
		MOSI	输入	输入
		MISO	CMOS 输出	OD 输出

20.4.3 SPI 系统连接实例

主机模式

在主机-多从机模式的 SPI 系统结构中，主机驱动 SCK、MOSI 和 SS0。在 SPI 从机设备 0~3 中，当某一从机的 SS0 输入为有效电平时，该从机设备驱动 MISO。

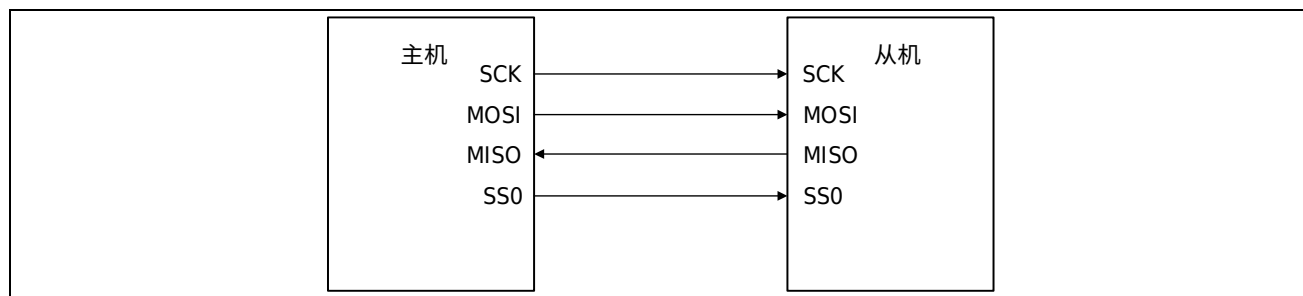


图 20-2 主机模式结构

时钟同步运行

在用作时钟同步运行动作的 SPI 系统结构中，主机设备驱动 SCK 和 MOSI，从机设备驱动 MISO。SS 管脚不作使用。

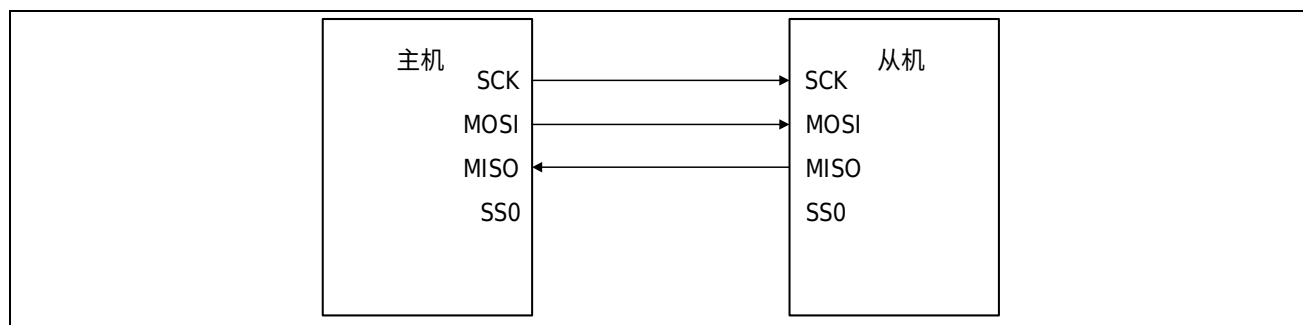


图 20-3 三线式时钟同步运行

20.5数据通信说明

20.5.1 波特率

主机模式下，SPI 时钟由内部波特率发生器提供；从机模式下，时钟为 SCK 脚输入。

波特率取决于 SPI_CFG2.MBR[2:0] 位的设置。其计算方法如下面公式所示，公式中 N 为 MBR[2:0] 位的设定值，范围是 0~7。

$$\text{波特率} = \frac{fpck}{2^{N+1}}$$

表 20-5 部分设定值的位速

MBR[2:0]位的设定值	分频比	波特率		
		HCLK=5MHz	HCLK=10MHz	HCLK=20MHz
0	2	2.50Mbps	5.00Mbps	10.0Mbps
1	4	1.25Mbps	2.50Mbps	5.00Mbps
2	8	625kbps	1.25Mbps	2.50Mbps
3	16	313kbps	625kbps	1.25Mbps
4	32	156kbps	313kbps	625kbps
5	64	78kbps	156kbps	313kbps
6	128	39kbps	78kbps	156kbps
7	256	20kbps	39kbps	78kbps

20.5.2 数据格式

SPI 的数据格式取决于 SPI 命令寄存器 SPI_CFG2 和 SPI 控制寄存器 SPI_CR1 中的奇偶校验许可位 PAE 的设定值。SPI 将数据寄存器 SPI_DR 中从 LSB 位开始 8/16 位(数据长度由寄存器 SPI_CFG2 中的 DSIZE 位设置)的数据作为传送对象进行处理,与 MSB/LSB 移位顺序无关。

SPI_CFG2.DSIZE 决定数据的位宽,位宽范围为 8/16 位, SPI_CR1.PAE 决定数据最末位, PAE 为 1 时最末位作为奇偶校验位,为 0 时则是数据本身最低位。如图 20-4 所示。

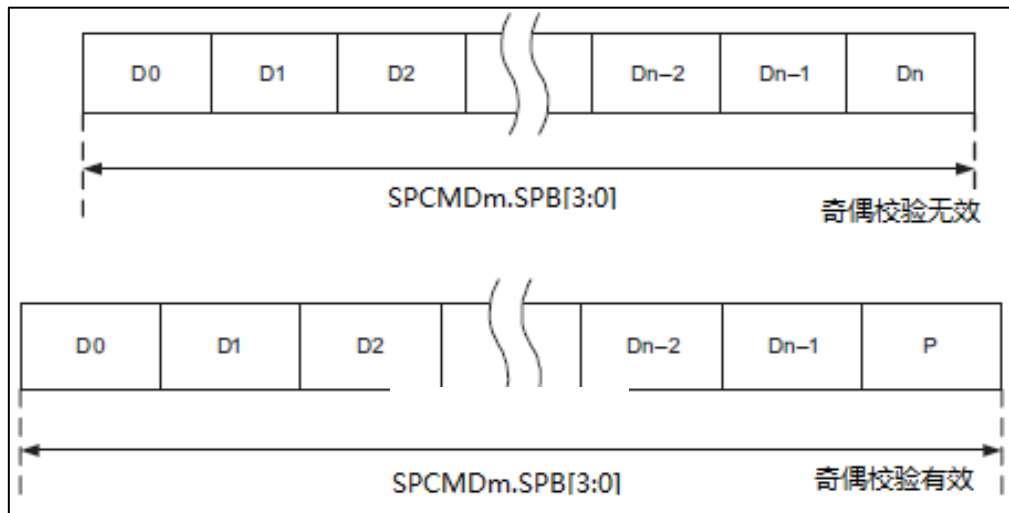


图 20-4 数据格式

SPI 数据发送时,传送数据先进入发送缓冲器 (TX_BUFF),再将 TX_BUFF 的数据复制到移位寄存器 (shifter),shifter 依次发出数据;SPI 数据接收时,数据从 shifter 依次移入,移入完成后再将 shifter 的数据复制到接收缓冲器 (RX_BUFF)。

数据传输时,根据移位顺序控制位 SPI_CFG2.LSBF 和奇偶校验控制位 SPI_CR1.PAE 的设置分为 4 种情况:

1) MSB 先传, 奇偶校验无效

发送时,数据 d15~d0 按照顺序从 TX_BUFF 复制到 shifter,按照 d15~d0 的顺序从 shifter 的最高位移出;

接收时,数据 d15~d0 从 shifter 的最低位移入,等到数据全部移入后再将数据复制到 RX_BUFF。

2) LSB 先传, 奇偶校验无效

发送时,数据 d15~d0 按照 d0~d15 的顺序从 TX_BUFF 复制到 shifter,按照 d0~d15 的顺序从 shifter 的最高位移出;

接收时,数据 d0~d15 从 shifter 的最低位移入,等到数据全部移入后,按照 d15~d0 的顺序从 shifter 复制到 RX_BUFF。

3) MSB 先传，奇偶校验有效时

发送时，先根据 d15~d1 的值计算出奇偶校验位 P 的值，然后用 P 代替 d0，按照 d15~d1, P 的顺序复制到 shifter，按照 d15~P 的顺序从 shifter 的最高位移出；

接收时，数据 d15~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。最后再将数据复制到 RX_BUFF。

4) LSB 先传，奇偶校验有效时

发送时，先根据 d14~d0 的值计算出奇偶校验位 P 的值，然后用 P 代替 d15，按照 d0~P 的顺序从 TX_BUFF 复制到 shifter，按照 d0~P 的顺序从 shifter 的最高位移出；

接收时，数据 d0~P 从 shifter 的最低位移入，在数据复制到 shifter 时，进行奇偶校验。数据 d0~P 在复制时重新排列，按照 P~d0 的顺序复制到 RX_BUFF。

20.5.3 传送格式

1) CPHA=0 的情况

当 SPI_CFG2.CPHA 位为“0”时，SPI 在 SCK 的奇数边沿进行数据采样，偶数边沿进行数据更新。图 20-5 是 CPHA=0 时 SPI 的传送时序图。当 SSi 信号的输入电平变为有效电平时，MOSI/MISO 开始更新传输数据。在 SSi 信号变为有效后的第一个 SCK 信号边沿进行第一次数据采样，在此之后，每过一个 SCK 周期对数据进行一次采样。每次采样后的 1/2 个 SCK 周期时 MOSI/MISO 信号上数据进行更新。CPOL 位的设定值不影响 SCK 信号的运行时序，而只影响信号的极性。

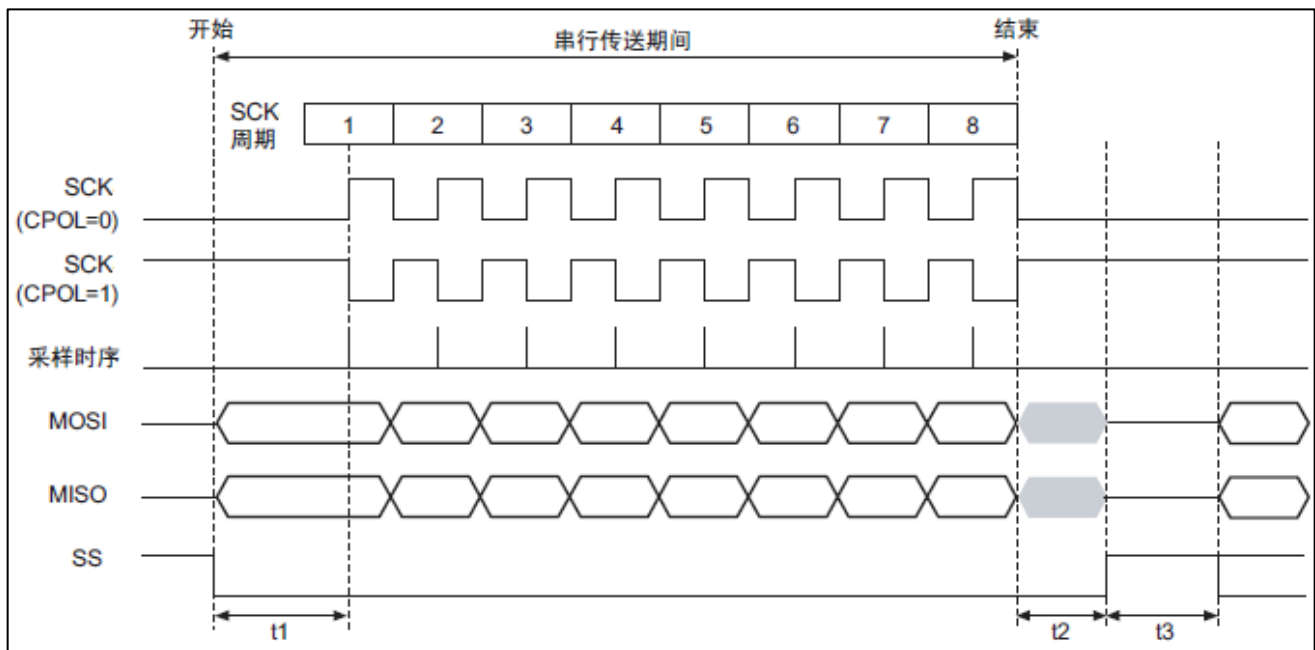


图 20-5 数据传送格式图 (CPHA=0)

2) CPHA=1 的情况

当 SPI_CFG2.CPHA 位为“1”时，SPI 在 SCK 的奇数边沿进行数据更新，偶数边沿进行数据采样。图 20-6 是 CPHA=1 时 SPI 的传送时序图。MOSI/MISO 在 SSi 信号变为有效电平后的第一个 SCK 信号边沿时开始传输数据的更新。在此之后，每过一个 SCK 周期对数据进行一次更新。每次更新后的 1/2 个 SCK 周期时对数据进行采样。SPI_CFG2.CPOL 位的设定值不影响 SCK 信号的运行时序而只影响信号的极性。

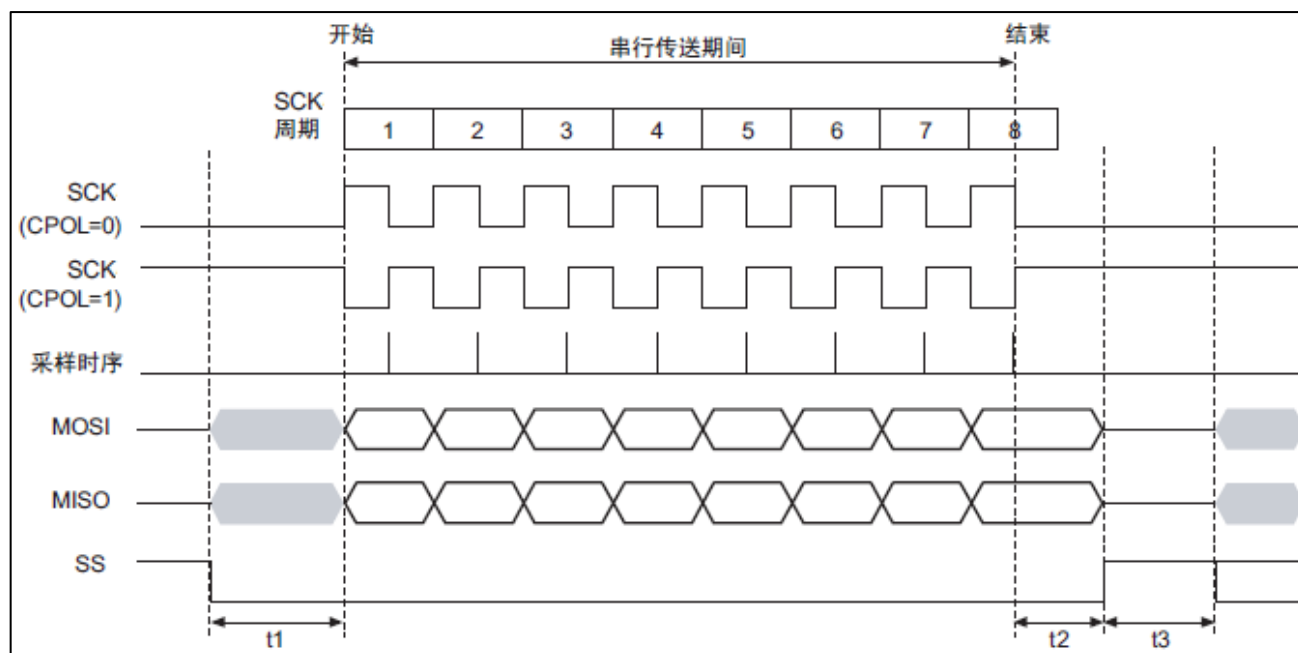


图 20-6 数据传送格式 (CPHA=1)

20.5.4 通信方式

本 SPI 有全双工同步串行通信和只发送串行通信两种通信方式，可通过 SPI 控制寄存器（SPI_CR1）的 TXMDS 位进行选择。

1) 全双工同步串行通信方式

当 SPI_CR1.TXMDS 位为“0”时，SPI 运行在全双工同步串行通信方式。如图 20-7 所示，SPI_CFG2.CPHA 位为“1”并且 SPI_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

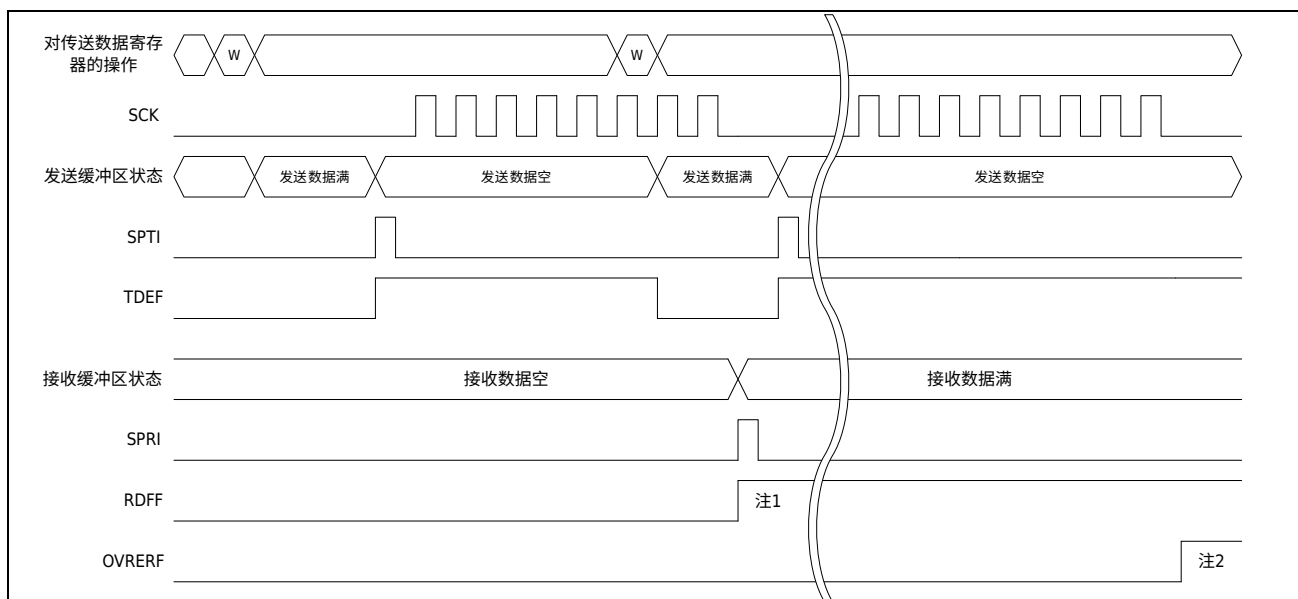


图 20-7 全双工同步串行通信

注：

1. 当本次串行传输结束时，如果接收数据缓冲寄存器为空，SPI 将会把接收到的数据从移位寄存器复制到接收数据缓冲寄存器中，接收数据缓冲寄存器满的标志位被置成 1 (RDFF)，并产生一个接收数据满的中断请求 (SPRI)。
2. 当本次串行传输结束时，如果接收数据缓冲寄存器中还保持着上次收到的数据而没有被系统读取，SPI 会将数据过载标志位置成 1，本次数据接收无效，接收移位寄存器中的数据将被丢弃。

2) 只进行发送通信方式

当 SPI_CR1.TXMD5 位为“1” 时，SPI 运行在只发送通信方式。如图 20-8 所示，SPI_CFG2.CPHA 位为“1” 并且 SPI_CFG2.CPOL 位为“0”，SPI 进行 8 位串行传送。

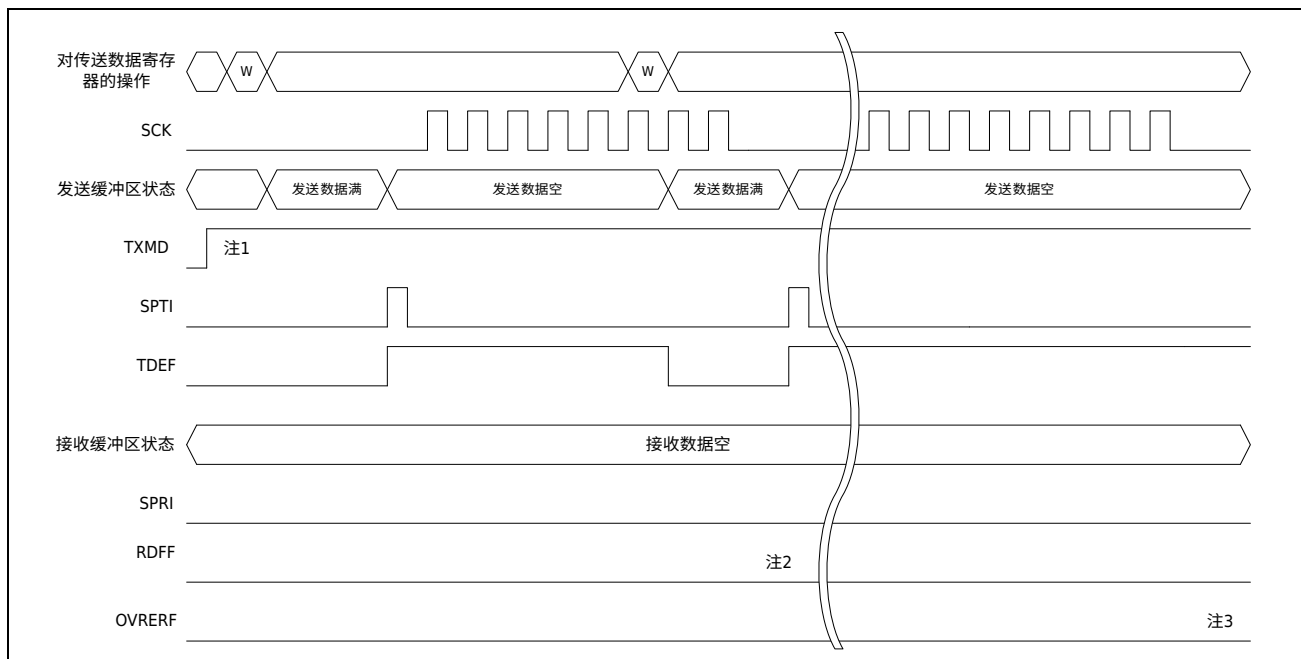


图 20-8 只进行发送通信

注：

1. 在设置进入只发送通信方式之前，请确保接收缓冲寄存器中没有未读取的数据（即 *RDFF* 为 0）且未出现数据过载错误（即 *OVRERF* 为 0）。
2. 在只发送通信方式下，当本次串行传输结束时，即使接收数据缓冲寄存器为空，也不会有数据被接受，*RDFF* 将始终保持 0 的状态。
3. 在只发送通信方式下，由于接收数据缓冲寄存器始终为空，所以不会发生数据过载错误，*OVRERF* 标志位始终保持 0 的状态。

20.6 运行说明

20.6.1 运行模式概要

本 SPI 支持 4 线式 SPI 模式和 3 线式时钟同步运行模式。每个运行模式下都可作为主机或者从机进行串行通信。设置 SPI 控制寄存器 SPI_CR1 中的 MSTR 和 SPIMDS 位设定 SPI 的模式。SPI 的模式和 SPI_CR1 寄存器设定的关系以及各模式概要如表 20-6 所示。

表 20-6 SPI 模式和寄存器设定关系

模式	主机 (SPI 运行)	从机 (SPI 运行)	主机 (时钟同步运行)	从机 (时钟同步运行)
MSTR 位的设定	1	0	1	0
SPIMDS 位的设定	0	0	1	1
SCK 信号	输出	输入	输出	输入
MOSI 信号	输出	输入	输出	输入
MISO 信号	输入	输出/Hi-Z	输入	输出
SS0 信号	输出	输入	Hi-Z (不使用)	Hi-Z (不使用)
SS 极性变更功能	有	有	-	-
最大传输速率	~HCLK/2	~HCLK/6	~HCLK/2	~HCLK/6
时钟源	内部波特率发生器	SCK 输入	内部波特率发生器	SCK 输入
时钟极性	2 种	2 种	2 种	2 种
时钟相位	2 种	2 种	2 种	1 种 (CHPA=1)
开始传送位	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
传送数据长度	8/16 位	8/16 位	8/16 位	8/16 位
传送启动方法	通过发送缓冲器空中断请求, 写发送缓冲器	SS 输入有效或 SCK 时钟边沿	通过发送缓冲器空中断请求, 写发送缓冲器	SCK 振荡
发送缓冲器空检测	有	有	有	有
接收缓冲器满检测	有 (注 1)	有 (注 1)	有 (注 1)	有 (注 1)
过载错误检测	有 (注 1)	有 (注 1)	有 (注 1)	有 (注 1)
奇偶校验错误检测	有 (注 1、注 2)	有 (注 1、注 2)	有 (注 1、注 2)	有 (注 1、注 2)
欠载错误检测	无	有	无	有

注:

1. 当 SPI_CR1.TXMDS 位为“1”时, 不进行接收缓冲器满的检测、过载错误的检测和奇偶校验错误的检测。
2. 当 SPI_CR1.PAE 位为“0”时, 不进行奇偶校验错误的检测。

20.6.2 SPI 运行模式时的主机动作

1) SPI 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX_BUFF) 为空的状态下 (状态寄存器 SPI_SR 中的 TDEF 标志位为 1)，将数据写到 SPI 数据寄存器 (SPI_DR) 后，SPI 会将 SPI_DR 数据更新至 TX_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI_CFG2.CPHA 位无关。在接收缓冲器 (RX_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX_BUFF 中，可通过数据寄存器 SPI_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI_CFG2.DSIZE 位的设定值，SS 输出引脚的极性取决于 SPI_CFG1 寄存器的设定值。有关 SPI 传送格式的详细内容，请参照 20.5.3 传送格式。

2) SPI 主机模式的初始化

- ① 设置通信配置寄存器 1 (SPI_CFG1)，包括有波特率的设定等。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有 SS 电平设定，数据移位顺序设定，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

20.6.3 SPI 运行模式时的从机动作

1) SPI 作为从机时的动作说明

当 SPI_CFG2.CPHA 位为 0 时，如果 SPI 检测到 SS0 输入信号变为有效电平，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为 0 时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。因此，在 CPHA 位为“1”时，将 SS0 信号处于有效电平状态下的首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束，结束时间与 SPI_CFG2.CPHA 位无关。在 RX_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接受数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

在串行传送期间，如果 SPI 检测到 SS0 输入信号无效，就发生模式故障错误。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE 位的设定值，SS0 输入信号的极性取决于 SPI_CFG1.SS0PV 位的设定值。有关 SPI 传送格式的详细内容，请参照【传送格式】。

注：

- 当 SPI_CFG2.CPHA 位为“0”时，将 SS0 输入信号电平从无效变为有效视为开始串行传送的触发信号。如果在单从机模式的结构中 SS0 输入信号被固定为有效状态，此时 SPI 将无法开始串行传送。因此，在 SS0 输入信号被固定为有效状态的结构中，要使从机模式的 SPI 正常进行发送和接收，必须将 CPHA 位设置为“1”。如果需要将 CPHA 位设成“0”，则不能固定 SS0 输入信号。

2) SPI 从机模式的初始化

- ① 设定通信配置寄存器 1 (SPI_CFG1)。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。

- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

20.6.4 时钟同步运行模式中的主机动作

当 SPI 控制寄存器 SPI_CR1 中的 SPIMDS 位为 1 时，SPI 处于时钟同步运行模式。在该模式动作时，SPI 只使用 SCK、MOSI 和 MISO 这 3 个管脚进行通信，SS0 管脚被释放可用于普通 I/O 功能。

尽管时钟同步运行模式时不使用 SS0 管脚，但模块内部的运行和 SPI 运行模式是相同的。但由于没有了 SS0 的输入，所以检测不到模式故障错误。

1) SPI 作为主机时的动作说明

在 SPI 数据发送缓冲寄存器 (TX_BUFF) 为空的状态下 (状态寄存器 SPI_SR 中的 TDEF 标志位为 1)，将数据写到 SPI 数据寄存器 (SPI_DR) 后，SPI 会将 SPI_DR 数据更新至 TX_BUFF 中。此时若移位寄存器 (shifter) 为空，SPI 将 TX_BUFF 的数据复制到移位寄存器开始串行传送。

当发送数据复制到 shifter 后，SPI 将会 shifter 状态改为满状态；当串行传送结束，改为空状态。shifter 的状态无法进行读取。

当 SPI 发送完最后采样时序所需的 SCK 边沿后，本次串行传送结束，结束时间与 SPI_CFG2.CPHA 位无关。在接收缓冲器 (RX_BUFF) 为空的情况下，串行传送结束后 SPI 将 shifter 中的数据复制到 RX_BUFF 中，可通过数据寄存器 SPI_DR 进行读取。

最后的采样时序取决于传送数据的位长，主控模式的 SPI 数据长度取决于 SPI_CFG2.DSIZE 位的设定值，SS 输出引脚的极性取决于 SPI_CFG1 寄存器的设定值。有关 SPI 传送格式的详细内容，请参照 20.5.3 传送格式。

2) 时钟同步运行模式时主机的初始化设置

- ① 设置通信配置寄存器 1 (SPI_CFG1)，包括有波特率的设定等。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有数据移位顺序设定，数据格式及时钟极性相位的设定等。

- ③ 如需要使用中断，请设置系统的中断寄存器。
- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

20.6.5 时钟同步运行模式中的从机动作

1) SPI 作为从机时的动作说明

当 SPI_CFG2.CPHA 位为 0 时，需要 SPI 检测到 SS0 输入信号变为有效电平作为开始串行通信的触发信号。由于时钟同步运行模式下不使用 SS0 管脚，因此，在 CPHA 位为 0 时无法进行正常通信。

当 CPHA 位为“1”时，如果在 SS0 输入信号为有效电平的状态下 SPI 检测到最初的 SCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。由于时钟同步运行模式下不适用 SS0 管脚，因此，在 CPHA 位为“1”时，将首个 SCK 边沿视为开始串行传送的触发信号。

如果 SPI 在 shifter 为空的状态下检测到串行传送的开始，就将 shifter 改为满状态，并且不能在串行传送过程中将数据从 TX_BUFF 传送到 shifter。如果在开始串行传送前 shifter 已满，SPI 就保持 shifter 的满状态。

如果 SPI 检测到最后采样时序的 SCK 边沿，则本次串行传输结束。在 RX_BUFF 为空的情况下，串行传送结束后 SPI 将 shifter 的接收数据复制到 RX_BUFF。可通过访问 SPI_DR 读取该数据。SPI 在串行传送结束后将 shifter 改为空状态，该状态与 RX_BUFF 的状态无关。

最后的采样时序取决于传送数据的位长，从属模式的 SPI 的数据长度取决于 SPI_CFG2.DSIZE 位的设定值。

2) 时钟同步运行模式时从机的初始化设置

- ① 设定通信配置寄存器 1 (SPI_CFG1)。
- ② 设置通信配置寄存器 2 (SPI_CFG2)，包括有传输速率，数据格式及时钟极性相位的设定等。
- ③ 如需要使用中断，请设置系统的中断寄存器。

- ④ 如需要使用 DMA，请设置 DMA 的相关寄存器。
- ⑤ 设定输入输出管脚。
- ⑥ 设定 SPI 控制寄存器 SPI_CR1，包括有模式及运行方式的设定，自诊断功能的设定，奇偶校验的设定等。
- ⑦ 确认 SPI_CR1 寄存器的设置。
- ⑧ 清除各种标志位。
- ⑨ 设置中断许可位。
- ⑩ 将控制寄存器 SPI_CR1 的 SPE 位设置成 1，动作开始。

20.6.6 几种 SPI 动作的处理流程

1) SPI 作为主机时的数据传送处理流程

- ① 等待数据发送缓冲寄存器空的中断或通过轮询方式确认数据发送缓冲寄存器处于空状态。
- ② 向数据寄存器 SPI_DR 写入要发送的数据。
- ③ 重复①②步骤直到最后一个数据发送完成。
- ④ 将发送数据寄存器空中断的允许位 TXIE 清零，同时将 SPI 闲置状态中断允许位 IDIE 设为 1
- ⑤ 发送 SPI 闲置状态中断。
- ⑥ 将 SPE 置 0，停止 SPI 动作，同时将 IDIE 清零。

2) 数据接收处理流程

- ① 等待数据接收缓冲寄存器满的中断或者通过轮询方式确认数据接收缓冲寄存器处于满状态。
- ② 通过访问 SPI_DR 从接收缓冲寄存器读取数据。
- ③ 重复①②步骤直到最后一个接收数据被读取。
- ④ 将数据接收缓冲寄存器满的中断允许位 RXIE 清零。

3) 通信错误处理流程

- ① 等待通信错误中断或者通过轮询方式确认通信错误标志位被置成 1。可轮询的错误标志位有：MODFERF/OVRERF/UDRERF/PERF。
- ② 确认 SS0 状态，排除模式故障错误。
- ③ 将 SPE 清零，停止 SPI 动作。

- ④ 将所有 SPI 中断允许位清零，屏蔽 SPI 中断。
- ⑤ 通过错误标志位确定通信错误种类，进行通讯错误处理。
- ⑥ 将错误标志位清零。
- ⑦ 启动 SPI，重新开始通信。

20.7 奇偶校验位自诊断

奇偶校验电路由传输数据的奇偶校验位和接收数据的错误检测部分构成。可按照下图所示流程使用自诊断功能对奇偶校验电路进行故障诊断。

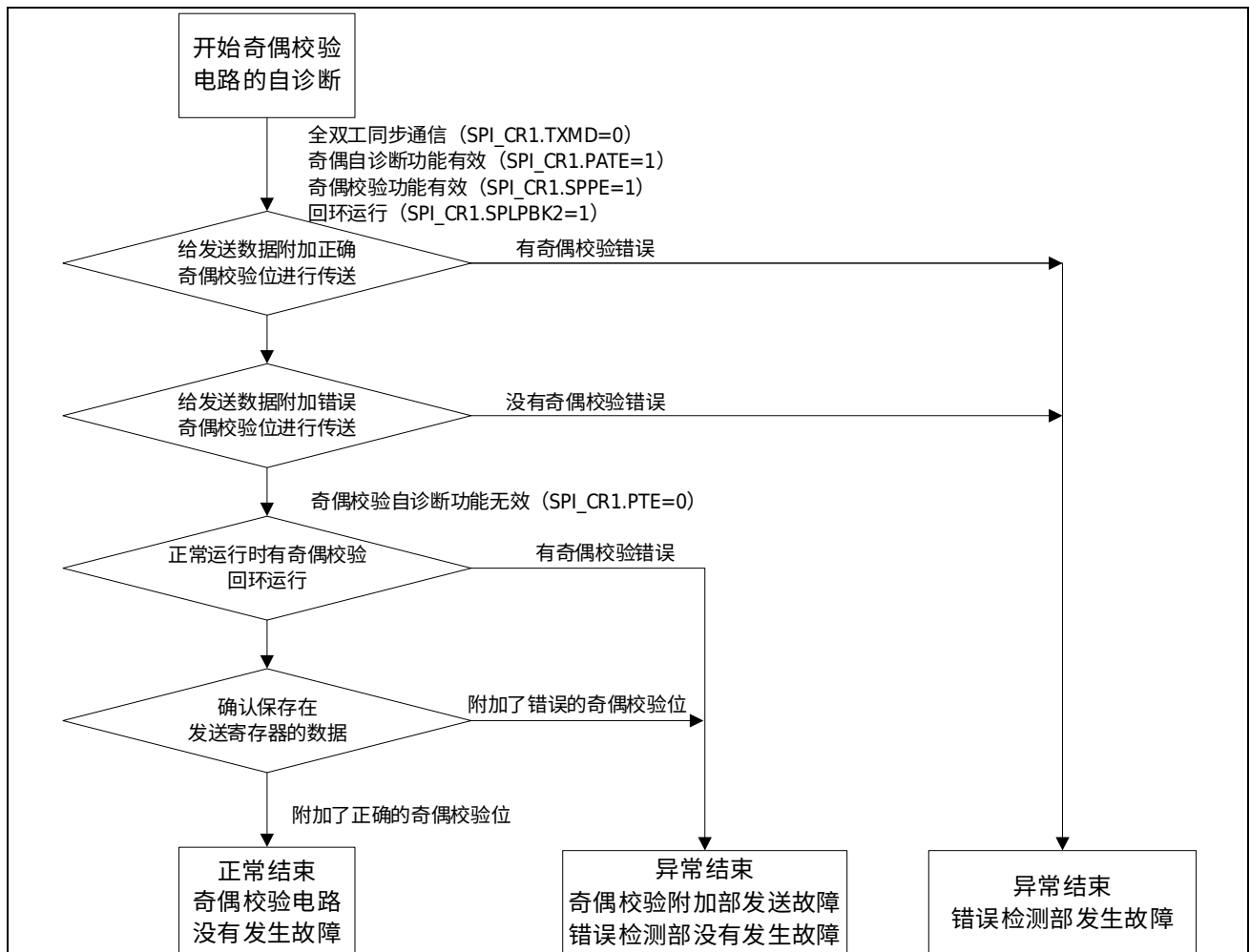


图 20-9 奇偶校验流程

20.8 错误检测

在正常的 SPI 串行传送中，系统通过对 SPI_DR 寄存器写操作进行数据串行发送，通过对 SPI_DR 寄存器的读操作获得串行接收的数据。但由于发送/接收缓冲器的状态以及串行传送开始/结束时的 SPI 的状态，某些情况下有可能会发生异常传送。当异常传送发生时，SPI 会将此次传输检测为欠载错误、过载错误、奇偶校验错误或者模式故障错误。异常传送和 SPI 错误检测的对应关系如下表 20-7 所示。

表 20-7 错误检测对应表

序号	发生条件	SPI 运行	检测错误
①	在发送缓冲器满的状态下写 SPI_DR 寄存器	保持发送缓冲器内容	无
		写数据丢失	
②	在接收缓冲器为空的状态下读 SPI_DR 寄存器	输出上次的串行接收数据	无
③	在从机模式中，发送数据未转移到移位寄存器的状态下开始串行传送	中止串行传输	欠载错误
		丢失发送和接收数据	
		停止驱动 MISO 输出信号	
		停止 SPI 功能	
④	从机模式：SS0 管脚有效电平宽度没有达到数据传输所需时间	中止传输	模式错误
		丢失发送和接收数据	
		停止 SPI 功能	
⑤	在接收缓冲器满的状态下结束串行传送	保持接收缓冲器内容	过载错误
		接收数据丢失	
⑥	在进行全双工同步串行通信，并且奇偶校验功能有效的状态下，接收到错误的奇偶校验位	奇偶校验错误标志有效	奇偶校验错误

①描述的情况，SPI 不发生检测错误。为防止在写数据到 SPI_DR 寄存器的过程中发生数据遗漏，必须通过发送缓冲器空的中断把数据写入 SPI_DR 寄存器。同理，②的情况下，SPI 也不发生检测错误。为防止无关数据被读入，SPI_DR 的数据读取必须通过接收寄存器满的中断请求来进行。

20.8.1 欠载错误

MSTR 位为 0 时，SPI 以从机状态运行，如果 SPE 置为 1 时，SS0 管脚接收到有效电平前，传送数据还未准备好，则 SPI 发生欠载错误，SPI_SR.MODFERF 和 SPI_SR.UDRERF 标志将被置为 1。

当检测到欠载错误后，SPI 将停止驱动信号输出，同时把 SPI_CR1.SPE 置为 0。

监测欠载错误可通过直接访问 SPI_SR 寄存器，或者使用 SPI 错误中断读取 SPI_SR 等方式进行。如果不使用错误中断，请使用轮询方式来监测欠载错误。

当 SPI_SR.MODFERF 为 1 时，系统禁止对 SPE 位写 1。要将 SPI_CR1.SPE 置成 1 使能 SPI 功能必须先将 MODFERF 标志清零。

20.8.2 模式错误

SPI 在主机模式时，SPI_CR.MODFE 请勿设定为 1。在从机模式时，当 SSI 有效电平宽度没有达到传输数据所需时间时，发生模式故障，SPI_SR.MODFERF 置 1，SPI_CR.SPE 置为 0。需要传送时，对 SPI_SR.MODFERF 清零后，再把 SPI_CR.SPE 置 1。

20.8.3 过载错误

如果在接收缓冲器满的状态下结束串行传送，SPI 发生过载错误，SPI_SR.OVRERF 标志被置成 1。因为在 OVRERF 标志为 1 的状态下 SPI 不会把移位寄存器的数据复制到接收缓冲器，所以接收缓冲器中保存的是发生错误前的接收数据。要将 OVRERF 标志置 0 时，需要在 OVRERF 标志为 1 的状态下读 SPI_SR 寄存器后才能给 OVRERF 标志写“0”。

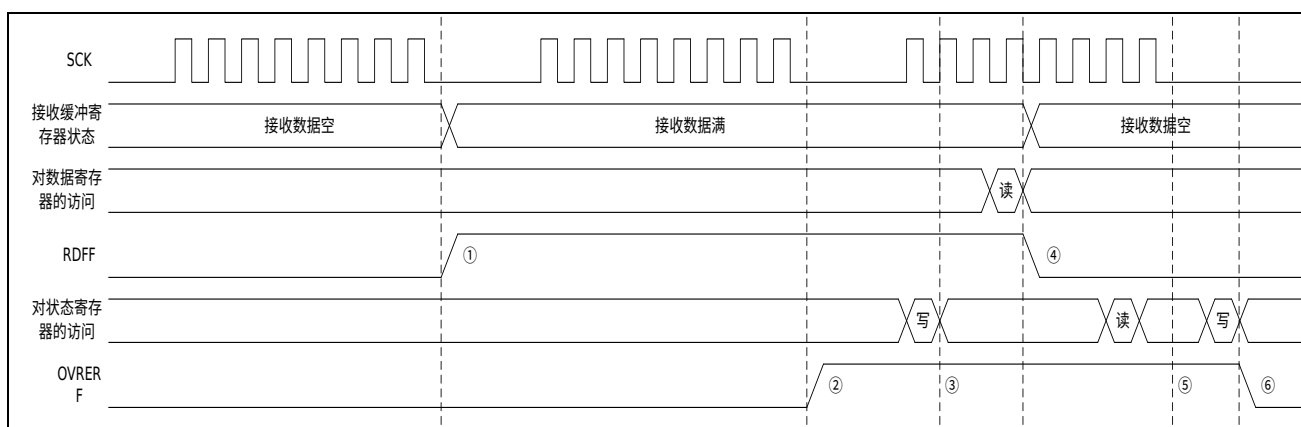


图 20-10 过载错误处理

20.8.4 奇偶校验错误

在 SPI_CR1.TXMD5 位为“0”并且 SPI_CR1.PAE 位为“1”的状态下，SPI 将在全双工同步串行通信结束时进行奇偶校验。当 SPI 检测到接收数据有奇偶校验错误时，将 SPI_SR.PERF 标志置 1。在 SPI_SR.OVRERF 位为“1”的状态下，因为 SPI 不会将移位寄存器的数据复制到接收缓冲器，所以不对接收数据进行奇偶校验错误的检测。要将 PERF 标志清零时，需要在 PERF 标志为 1 的状态下读 SPI_SR 寄存器后再给 PERF 标志写 0。

OVRERF 标志和 PERF 标志的运行实例如下图 20-11 所示。在图中的例子中，SPI 在 SPI_CR1.TXMD5 位为 0 并且 SPI_CR1.PAE 位为 1 的状态下进行全双工同步串行通信的 8 位串行传送。

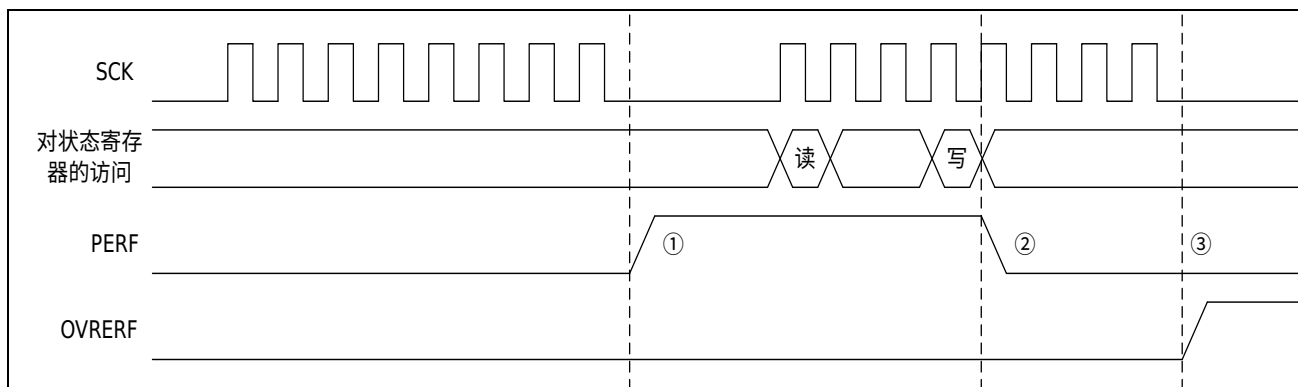


图 20-11 奇偶校验错误

以下说明在图中①~③所示的时序中标志的运行内容：

- ① SPI 未检测到过载错误，串行传送正常结束。SPI 将移位寄存器的数据复制到接收缓冲器。此时，SPI 对接收数据进行奇偶校验。如果检测到奇偶校验错误，则将 PERF 标志置 1。
- ② 在 PERF 标志为 1 的状态下读 SPI_SR 寄存器后给 PERF 标志写 0，将 PERF 标志清零。
- ③ SPI 检测到过载错误，此时 SPI 不会将移位寄存器的数据复制到接收缓冲器，SPI 不会对数据进行奇偶校验，不会发生奇偶校验错误。

可通过直接访问 SPI_SR 寄存器或者经 SPI 错误中断读取 SPI_SR 寄存器等方式监测奇偶校验错误的发生。在进行串行传送时，必须通过访问状态寄存器 SPI_SR 等方法，尽早监测到奇偶校验错误的发生。

20.9 SPI 的初始化

通过写操作或者模式故障错误检测将 SPE 位清零，可以使 SPI 功能无效并且对部分 SPI 功能进行初始化。如果发生系统复位，则对全部 SPI 功能进行初始化。

20.9.1 清除 SPE 位进行初始化

当 SPI_CR1.SPE 位为 0 时，SPI 进行以下的初始化操作：

- 中止正在进行的串行传送。
- 若处于从机状态则停止驱动输出信号（状态变为 Hi-Z）。
- 对 SPI 内部状态进行初始化。
- 清空发送缓冲寄存器，SPI_SR.TDEF 标志置为 1。

通过将 SPE 位清零来进行初始化时，不会对 SPI 的控制位进行初始化。因此，只要重新将 SPE 位置为 1，就能以和初始化前相同的传送模式启动 SPI。

清除 SPE 位不会对错误标志位和序列状态进行初始化。因此，在即使在 SPE 被清零以后，也能通过读取接收缓冲器的数据来确认 SPI 传送时的错误发生状况。

由于清除 SPE 位会清空发送缓冲寄存器，并将 SPI_SR.TDEF 标志置为 1。因此，如果在初始化后将 SPI_CR1.TXIE 位设成 1，就会产生 SPI 发送缓冲寄存器为空的中断。要避免系统发生此中断，必须在将 SPE 位清零的同时也将 TXIE 位设成 0。

20.9.2 系统复位初始化

通过系统复位进行初始化，将对 SPI 所有的控制位、状态位和数据寄存器进行初始化。

20.10 中断源

SPI 的中断源有接收缓冲器满、发送缓冲器空、模式故障、过载、欠载、奇偶校验错误和 SPI 空闲。其中接收缓冲器满和发送缓冲器空中断可用于启动 DMA 进行数据传送。

过载、欠载和奇偶校验错误的中断被集成为 SPI 错误中断 SPEI，所以需要通过标志判断实际发生的中断源。SPI 中断源的具体说明如表 20-8 所示。一旦中断条件成立，就产生相应的中断请求。

对于接收缓冲器满和发送缓冲器空的中断源，需要通过数据传送改变缓冲器状态来清除。在使用 DMA 进行发送或者接收时，必须先进行 DMA 的配置，DMA 设为动作许可的状态后再设定 SPI。

表 20-8 SPI 中断源说明

中断源	略称	中断条件	启动 DMA
接收缓冲器满	SPRI	在 SPI_CR1.RXIE 位为“1”的状态下接收缓冲器变满时	能
发送缓冲器空	SPTI	在 SPI_CR1.TXIE 位为“1”的状态下发送缓冲器变空时	能
SPI 错误（过载、欠载、奇偶校验错误）	SPEI	在 SPI_CR1.EIE 位为“1”的状态下 SPI_SR.OVRERF 和 SPI_SR.PERF 标志变为“1”时	不能
SPI 空闲中断	SPII	在 SPI_CR1.IDIE 位为“1”的状态下 IDLF 标志变为“0”时	不能

20.11 可供使用的事件触发源

SPI 产生的可供使用的事件触发源主要有以下几种：

- 数据发送缓冲寄存器空
- 数据接收缓冲寄存器满
- SPI 通信错误（包括过载，欠载，奇偶校验等错误）
- SPI 处于闲置状态
- SPI 通讯结束

用户可将上述事件触发源对应的向量写入不同的触发对象寄存器实现各种事件触发功能。

上述事件触发源对应的向量请参考【中断控制器（INTC）】。

20.12 寄存器描述

表 20-9 SPI 基地址

名称	基地址	描述
SPI	0x4000 3800	SPI 基地址

表 20-10 SPI 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
SPIDR	SPI 数据寄存器	0x00	32	0x0000 0000
SPICR	SPI 控制寄存器	0x04	32	0x0000 0000
SPICFG1	SPI 配置寄存器 1	0x0C	32	0x0000 0000
SPISR	SPI 状态寄存器	0x14	32	0x0000 0020
SPICFG2	SPI 配置寄存器 2	0x18	32	0x0000 011D

20.12.1 SPI 数据寄存器 (SPI_DR)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
SPD[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
SPD[7:0]							

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15~b0	SPD[15:0]	串行数据	SPI 数据存储	RW

20.12.2 SPI 控制寄存器 (SPI_CR1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE
b7	b6	b5	b4	b3	b2	b1	b0
Res	SPE	SPLPBK2	SPLPBK	MSTR	Res	TXMDS	SPIMDS

位/位域	标记	位名	功能	读写
b31~b16	Res	保留位	读出时为“0”,写入时写“0”	RW
b15	PAE	奇偶校验允许	0: 发送数据不附加奇偶校验位,接收数据不进行奇偶校验 1: 发送数据附加奇偶校验位,接收数据进行奇偶校验 (SPCR.TXMDS=0) 发送数据附加奇偶校验位,接收数据不进行奇偶校验 (SPCR.TXMDS=1)	RW
b14	PAOE	奇偶校验模式选择	0: 选择偶校验进行发送和接收 1: 选择奇校验进行发送和接收	RW
b13	PATE	奇偶校验自诊断	0: 奇偶校验自诊断功能无效 1: 奇偶校验自诊断功能有效	RW
b12	MODFE	模式故障错误检测允许	0: 禁止模式故障错误检测 1: 允许模式故障错误检测	RW
b11	IDIE	SPI 空闲中断允许	0: 禁止空闲中断请求产生 1: 允许空闲中断请求产生	RW
b10	RXIE	SPI 接收中断允许	0: 禁止 SPI 接收中断请求产生 1: 允许 SPI 接收中断请求产生	RW
b9	TXIE	SPI 发送中断允许	0: 禁止 SPI 发送中断请求产生 1: 允许 SPI 发送中断请求产生	RW
b8	EIE	SPI 错误中断允许	0: 禁止 SPI 错误中断请求产生 1: 允许 SPI 错误中断请求产生	RW
b7	Res	保留位	读出时为“0”,写入时写“0”	RW
b6	SPE	SPI 功能允许	0: SPI 功能无效 1: SPI 功能有效	RW
b5	SPLPBK2	SPI 回环 2 位	0: 正常模式 1: 回环模式 (发送数据=接收数据)	RW
b4	SPLPBK	SPI 回环位	0: 正常模式 1: 回环模式 (发送数据的反相=接收数据)	RW
b3	MSTR	SPI 主从模式选择	0: 从机模式 1: 主机模式	RW
b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1	TXMDS	通信模式选择	0: 全双工同步串行通信	RW

位/位域	标记	位名	功能	读写
1: 只进行发送串行通信				
b0	SPIMDS	SPI 模式选择	0: SPI 运行 (4 线式) 1: 时钟同步运行 (3 线式)	RW

20.12.3 SPI 通信配置寄存器 1 (SPI_CFG1)

复位值: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							SS0PV
b7	b6	b5	b4	b3	b2	b1	b0
Res							

位/位域	标记	位名	功能	读写
b31~b9	Res	保留位	读出时为“0”,写入时写“0”	RW
b8	SS0PV	SS0 信号极性设定	0: SS0 信号的 Low 电平有效 1: SS0 信号的 High 电平有效	RW
b7~b0	Res	保留位	读出时为“0”,写入时写“0”	RW

20.12.4 SPI 状态寄存器 (SPI_SR)

复位值: 0x0000 0020

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
RDFE	Res	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

位/位域	标记	位名	功能	读写
b31~b8	Res	保留位	读出时为“0”,写入时写“0”	RW
b7	RDFE	接收缓冲器满标志	0: SPI_DR 寄存器无数据 1: SPI_DR 寄存器有数据 硬件置位、清零, 写入时写“1”	RW
b6	Res	保留位	读出时为“0”,写入时写“0”	RW
b5	TDEF	发送缓冲器空标志	0: 发送缓冲器有数据 1: 发送缓冲器无数据 硬件置位、清零, 写入时写“1”	R
b4	UDRERF	欠载错误标志	0: 模式故障错误发生 (MODFERF=1) 1: 欠载错误发生 (MODFERF=1) 当 MODFERF=0, 此位被初始化 硬件置位后, 读取 1 写 0, 状态位清零	RW
b3	PERF	奇偶校验错误标志	0: 未发生奇偶校验错误 1: 发生奇偶校验错误 硬件置位后, 读取 1 写 0, 状态位清零	RW
b2	MODFERF	模式故障错误标志	0: 未发送模式故障错误 1: 发生模式故障错误 硬件置位后, 读取 1 写 0, 状态位清零	RW
b1	IDLNF	SPI 空闲标志	0: SPI 为空闲状态 1: SPI 为传送状态 硬件置位, 清零	R
b0	OVRERF	过载错误标志	0: 未发生过载错误 1: 发生过载错误 硬件置位后, 读取 1 写 0, 状态位清零	RW

20.12.5 SPI 通信配置寄存器 2 (SPI_CFG2)

复位值: 0x0000 011D

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res			LSBF	Res			DSIZE
b7	b6	b5	b4	b3	b2	b1	b0
Res			MBR[2:0]			CPOL	CPHA

位/位域	标记	位名	功能	读写
b31~b13	Res	保留位	读出时为“0”,写入时写“0”	RW
b12	LSBF	SPI LSB first 位	0: MSB first 1: LSB first	RW
b11~b9	Res	保留位	读出时为“0”,写入时写“0”	RW
b8	DSIZE	SPI 数据长度设定位	0: 8 bit 1: 16 bit	RW
b7~b5	Res	保留位	读出时为“0”,写入时写“0”	RW
b4~b2	MBR[2:0]	位速率分频设定位	b4~b2 0 0 0: 选择基本位速率的 2 分频 0 0 1: 选择基本位速率的 4 分频 0 1 0: 选择基本位速率的 8 分频 0 1 1: 选择基本位速率的 16 分频 1 0 0: 选择基本位速率的 32 分频 1 0 1: 选择基本位速率的 64 分频 1 1 0: 选择基本位速率的 128 分频 1 1 1: 选择基本位速率的 256 分频	RW
b1	CPOL	SCK 极性设定位	0: 空闲时的 SCK 为 Low 电平 1: 空闲时的 SCK 为 High 电平	RW
b0	CPHA	SCK 相位设定位	0: 在奇数边沿进行数据采样, 在偶数边沿数据发生变化 1: 在奇数边沿数据发生变化, 在偶数边沿进行数据采样	RW

21 CRC 运算（CRC）

21.1 概述

在许多应用中，都需要 CRC 算法来校验数据的完整性和正确性。尤其是在数据传输中，CRC 校验更是被广泛应用。本模块可采用 CRC16 和 CRC32 两种算法对数据进行运算和校验。

21.2 功能框图

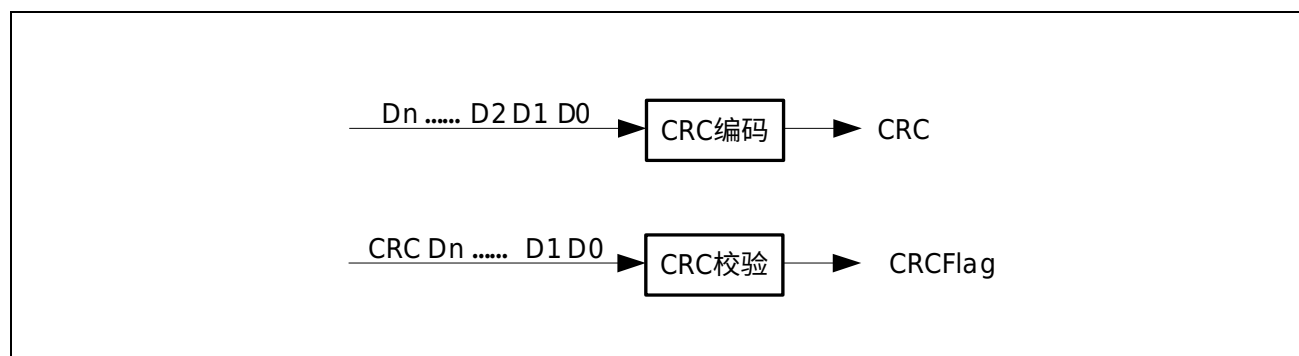


图 21-1 CRC 应用示意图

21.3 功能说明

本模块 CRC 算法遵从 ISO/IEC13239 的定义，分别采用 32 位和 16 位的 CRC。CRC32 的生成多项式为 $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ ，32 位初值为“0xFFFF FFFF”。CRC16 的生成多项式为 $X^{16}+X^{12}+X^5+1$ ，16 位初值为“0xFFFF”。

本模块功能包括：

- CRC 编码和 CRC 校验；
- 3 种位宽访问方式 8 位，16 位，32 位：
8 位位宽下输入数据示例为 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77;
16 位位宽下输入数据示例为 0x1100,0x3322,0x5544,0x7766;
32 位位宽下输入数据示例为 0x3322 1100,0x7766 5544;

21.3.1 CRC16 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC_CR.CR 写入 1'b0,选择 CRC16。
2. 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

注：

- 按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC_RESULT[15:0]，即可获得 16 位 CRC 编码值。

21.3.2 CRC16 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC_CR.CR 写入 1'b0,选择 CRC16。
2. 向 CRC_RESULT[15:0]写入 0xFFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

注：

- 按 8 位组织方式写 CRC 值，应先写低位，后写高位。将 16/32 位的 CRC 编码值写入 CRC_DAT 寄存器。
4. 读取 CRC_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

21.3.3 CRC32 编码模式

CRC 编码就是对原始数据编码以计算其 CRC 值，操作流程如下：

1. 向 CRC_CR.CR 写入 1'b1,选择 CRC32。
2. 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算。
3. 将待编码的原始数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

注：

- 按 8 位组织方式写 CRC 值，应先写低位，后写高位。
4. 读取 CRC_RESULT[31:0]，即可获得 16 位 CRC 编码值。

21.3.4 CRC32 校验模式

校验模式可以校验已编码的数据是否被篡改，操作流程如下所示：

1. 向 CRC_CR.CR 写入 1'b1,选择 CRC32。
2. 向 CRC_RESULT[31:0]写入 0xFFFF FFFF，初始化 CRC 计算。
3. 将已编码的数据按 8 位/16 位/32 位组织方式，依次写入 CRC_DAT 寄存器。

注：

- 按 8 位组织方式写 CRC 值，应先写低位，后写高位。将 16/32 位的 CRC 编码值写入 CRC_DAT 寄存器。
4. 读取 CRC_CR.FLG 寄存器，为 1 则表示校验成功，为 0 则表示校验失败。

21.4 寄存器描述

表 21-1 CRC 基地址

名称	基地址	描述
CRC	0x4001 5400	CRC 基地址

表 21-2 CRC 寄存器列表

寄存器	描述	偏移地址	位宽	复位值
CRC_CR	CRC 控制寄存器	0x00	32	0x0000 0001
CRC_RESLT	CRC 结果寄存器	0x04	32	0x0000 0000
CRC_DAT	CRC 数据寄存器	0x80~0xFF	32	0x0000 0000

21.4.1 控制寄存器 (CRC_CR)

复位值: 0x0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														FLAG	CR

位/位域	标记	位名	功能	读写
b30~b2	Res	保留位	读出时为“0”,写入时写“0”	R
b1	FLAG	校验结果位	0: 当前校验错误 1: 当前校验正确	R
b0	CR	运算控制位	0: CRC16 1: CRC32	RW

21.4.2 结果寄存器 (CRC_RESLT)

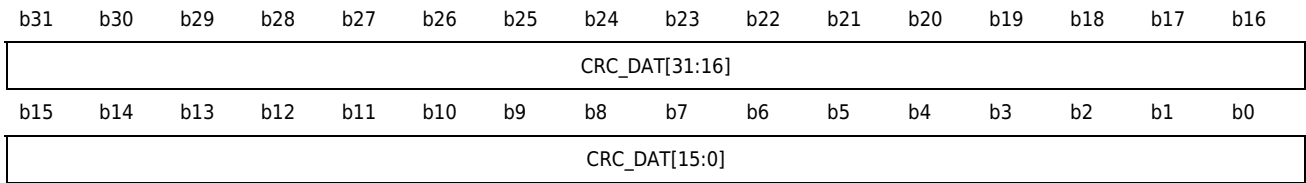
复位值: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RESULT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULT[15:0]															

位/位域	标记	位名	功能	读写
b31~b0	RESULT[31:0]	结果位	选择 CRC16 时, 取 RESULT[15:0]; 选择 CRC32 时, 取 RESULT[31:0];	RW

21.4.3 数据寄存器（CRC_DAT）

复位值：0x0000



位/位域	标记	位名	功能	读写
b31~b0	CRC_DAT[31:0]	数据寄存器	本寄存器用于输入需要运算的数据；本寄存器的地址是一个范围(0x80~0xFF)，对该范围内任何一个地址进行操作，都会认为对本寄存器进行操作。	W

22 调试控制器 (DBGC)

本产品参考如下 ARM 技术文档：

Cortex™-M0+ 技术参考手册

ARM 调试接口 V5

22.1 概述

本 MCU 的内核是 Cortex™-M0+，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，将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，将根据安全级别设定使用相应的内核硬件调试模块。

本产品提供两种调试接口：

单线串行调试接口 TOOL0

双线串行调试接口 SWD

22.2 DBGC 系统框图

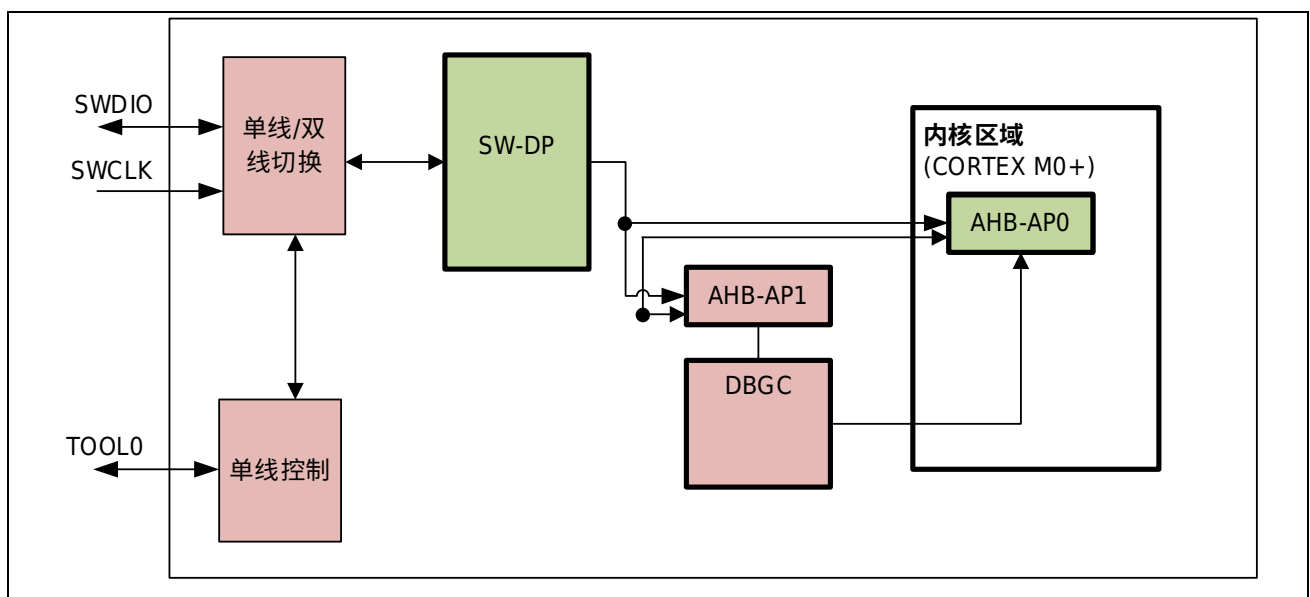


图 22-1 调试控制系统

ARM Cortex™-M0+ 内核提供集成片上调试支持。它包括：

SW-DP：串行调试端口

AHB-AP：AHB 访问端口

DWT：数据断点触发

灵活调试引脚分配

注：

- 有关 ARM Cortex™-M0+ 内核支持的调试功能的更多信息，请参见《Cortex™-M0+ 技术参考手册》。

22.3 调试端口

该 MCU 内核集成了串行调试端口(SW-DP)和单线调试接口 TOOL0。SW-DP 是 ARM 标准 CoreSight 调试端口，具有 SW-DP (2 引脚) 接口。TOOL0 是单线调试接口，只有 (1 引脚) TOOL0 接口。

默认接口为双线 SWD (时钟+数据)。

如果想要从双线 SWD 切换到单线 TOOL0 调试，可将单线调试器接上 TOOL0 端口和芯片复位脚 (NRST)，然后利用单线调试器发出的配置时序可进行单线 TOOL0 调试。

如果想要从单线 TOOL0 调试切换到双线 SWD，可将芯片重新上电复位，即可恢复为默认的双线 SWD 调试模式。

22.4 引脚排列和调试端口引脚

根据 MCU 的不同封装有不同的有效引脚数。因此，某些与引脚相关的功能可能随封装而不同。

22.4.1 调试端口引脚

MCU 的 3 个普通 I/O 口可用作 SW-DP 接口引脚。所有封装都提供这些引脚。

引脚名称	调试端口		MCU 引脚分配
	类型	调试分配	
SWCLK	I	SWD 串行时钟	P50
SWDIO	I/O	SWD 串行线数据输入/输出	P51
TOOL0	I/O	TOOL0 单线数据输入/输出	P40/MD

22.4.2 灵活的引脚分配

复位 (上电或引脚复位) 后，会将用于 SW-DP 的全部 2 个引脚指定为专用引脚，可供调试工具立即使用。但是，MCU 可以禁止部分或全部 SW-DP 端口，进而释放相关引脚以用作通用 IO (GPIO)。

复位 (上电或引脚复位) 后 TOOL0 默认为调试引脚，可供调试工具立即使用。但是在未连接调试器的状态下，MCU 可以禁止引脚调试功能，进而释放相关引脚以用作通用 IO (GPIO)。

有关如何禁止 SW-DP 端口引脚的更多详细信息，请参见：I/O 引脚复用器和映射。

22.5 引脚上的内部上拉

为避免 IO 电平浮空，器件在 SWD 引脚上内置有内部上拉电阻：

TOOL0：内部上拉

SWDIO：内部上拉

SWCLK：内部上拉

在未连接调试器状态下，用户软件可以通过设定 GPIO 控制器释放调试管脚作为普通的 I/O 口使用。

由于芯片内部上拉为<100K 欧的弱上拉，建议采用外部 10K 欧的外部上拉。

22.6 寄存器描述

表 22-1 DBG 基地址

名称	基地址	描述
DBG	0x4001 5000	DBG 基地址

表 22-2 DBG 寄存器列表

寄存器	描述	偏移地址	位宽	复位值	访问主机
MCUDBGSTAT	DBG 状态寄存器	0x0000	32	0x0000 0000	CPU/调试 IDE
MCUSTPCTL	外设调试暂停寄存器	0x0004	32	0x0000 0009	CPU/调试 IDE

22.6.1 DBG 状态寄存器（MCUDBGSTAT）

DBG 调试上电状态确认寄存器。

复位值：0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CDBG PWR UPACK	CDBG PWR UPREQ

位/位域	标记	位名	功能	读写
b31~b2	Res	保留位	读出时为“0”,写入时写“0”	RW
b1	CDBGPWRUPACK	调试器上电反馈	0: 无反馈 1: 调试上电反馈	RW
b0	CDBGPWRUPREQ	调试器上电请求	0: 无上电请求 1: 上电请求	RW

22.6.2 外设调试暂停寄存器 (MCUSTPCTL)

当 CPU 处于调试暂停状态时，外设模块暂停控制。

复位值：0x0000 0009

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMRB_8 STP	TMRB_7 STP	TMRB_6 STP	TMRB_5 STP	TMRB_4 STP	TMRB_3 STP	TMRB_2 STP	TMRB_1 STP	Res							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMR0 STP	Res										LVD STP	RTC STP	Res	SWD TSTP

位/位域	标记	位名	功能	读写
b31	TMRB8STP	TimerB_8 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b30	TMRB7STP	TimerB_7 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b29	TMRB6STP	TimerB_6 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b28	TMRB5STP	TimerB_5 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b27	TMRB4STP	TimerB_4 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b26	TMRB3STP	TimerB_3 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b25	TMRB2STP	TimerB_2 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b24	TMRB1STP	TimerB_1 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b23~b15	Res	保留位	读出时为“0”, 写入时写“0”	RW
b14	TMR0STP	Timer0 计数暂停信号	0: 即使内核停止, 计数器仍然计数 1: 内核停止时, 计数器暂停计数	RW
b13~b4	Res	保留位	读出时为“0”, 写入时写“0”	RW
b3	LVDSTP	LVD 中断/复位屏蔽	0: 即使内核停止, 仍然产生 LVD 中断申请或复位 1: 内核停止时, 屏蔽 LVD 中断申请或复位	RW
b2	RTCSTP	RTC 计数暂停信号	0: 即使内核停止, RTC 计数器仍然计数 1: 内核停止时, RTC 计数器暂停计数	RW
b1	Res	保留位	读出时为“0”, 写入时写“0”	RW
b0	SWDTSTP	SWDT 计数暂停信号	0: 即使内核停止, SWDT 计数器仍然计数 1: 内核停止时, SWDT 计数器暂停计数	RW

版本修订记录

版本号	修订日期	修订内容
Rev1.00	2024/12/12	初版发布。
Rev1.01	2026/04/16	1) 时钟控制器 (CMU): 4.5.1.2 XTALMS 位添加 “置 1” 的描述、XTALSTP 位由置 “1” 改为置 “0”; 图 4-5 清零 XTALSTDF 标志位后的等待时间由 “TDB.” 改为 “1ms”; 4.5.3 修正笔误; 4.7.1 图 4-7 “等待时钟稳定” 的上一步改为 “通过时钟配置寄存器, 使时钟源振荡”; 4.9.6 修改 CMU XTAL32 配置寄存器的功能位。 2) 电源控制 (PWC): 5.7.7 LVD 检测标志位 (b4 DETF) 的功能描述添加说明; b0 EXVCCINEN 的位名修改为 “外部输入控制位”。 3) 嵌入式 FLASH (EFM): 7.7.5 明确描述 EFM_FSR 寄存器中 ERCAL 标志位的功能; 7.8 拆分第 4 点注意事项为第 4 点和第 5 点、添加第 6 点注意事项。 4) 通用 IO (GPIO): 9.3.3 优化外部中断源的符号名称。 5) DMA 控制器 (DMA): 11.4.3 修正笔误。 6) 模数转换模块 (ADC): 表 13-4 修正笔误。 7) 实时时钟 (RTC): 16.5.4 添加 RTC 计数时钟选择 LRC 时的注意事项。 8) 调试控制器 (DBGC): 22.6.2 优化寄存器部分位的标记和位名符号。 9) 全文删除 “掉电模式” 相关的描述。